PointNest: Learning Deep Multiscale Nested Feature Propagation for Semantic Segmentation of 3-D Point Clouds

Jie Wan, Ziyin Zeng, Qinjun Qiu, Zhong Xie, and Yongyang Xu

***Abstract*—3-D point cloud semantic segmentation is a fundamental task for scene understanding, but this task remains challenging due to the diverse scene classes, data defects, and occlusions. Most existing deep learning-based methods focus on new designs of feature extraction operators but neglect the importance of exploiting multiscale point information in the network, which is crucial for identifying objects under complex scenes. To tackle this limitation, we propose an innovative network called PointNest that efficiently learns multiscale point feature propagation for accurate point segmentation. PointNest employs a deep nested U-shape encoder–decoder architecture, where the encoder learns multiscale point features through nested feature aggregation units at different network depths and propagates local geometric contextual information with skip connections along horizontal and vertical directions. The decoder then receives multiscale nested features from the encoder to progressively recover geometric details of the abstracted decoding point features for pointwise semantic prediction. In addition, we introduce a deep supervision strategy to further promote multiscale information propagation in the network for efficient training and performance improvement. Experiments on three public benchmarks demonstrate that PointNest outperforms existing mainstream methods with the mean intersection over union scores of 68.8%, 74.7%, and 62.7% in S3DIS, Toronto-3D, and WHU-MLS datasets, respectively.**

***Index Terms*—3-D point cloud, deep supervision (DS), multiscale feature propagation, semantic segmentation.**

Manuscript received 24 May 2023; revised 16 August 2023 and 8 September 2023; accepted 10 September 2023. Date of publication 14 September 2023; date of current version 5 October 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB3904200, in part by the Natural Science Foundation of Hubei Province of China under Grant 2022CFB640, and in part by the Open Fund of Hubei Key Laboratory of Intelligent Vision Based Monitoring for Hydroelectric Engineering under Grant 2022SDSJ04. *(Corresponding author: Qinjun Qiu.)*

Jie Wan is with the Key Laboratory of Geological and Evaluation of Ministry of Education, China University of Geosciences, Wuhan 430074, China (e-mail: wanjie@cug.edu.cn).

Ziyin Zeng is with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China (e-mail: zengziyin@cug.edu.cn).

Qinjun Qiu and Yongyang Xu are with the School of Computer Science, Wuhan 430074, China (e-mail: qiuqinjun@cug.edu.cn; yongyangxu@cug.edu.cn).

Zhong Xie is with the School of Computer Science, China University of Geosciences, Wuhan 430074, China, and also with the Key Laboratory of Geological and Evaluation of Ministry of Education, China University of Geosciences, Wuhan 430074, China (e-mail: xiezhong@cug.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2023.3315557

I. Introduction

With the rise of smart cities, more and more emerging applications, such as robotics [1], autonomous driving [2], and 3-D urban modeling [3], pose an increasing demand for accurate 3-D semantic information to perform precise 3-D scene analysis and interpretation. Due to recent advancements in sensor technology, 3-D point cloud data collected by the laser scanning equipment can provide remarkable details when describing large-scale urban scenes, making it increasingly valuable for urban studies. As a basic research topic for 3-D scene understanding, 3-D point cloud semantic segmentation can deliver significant semantic insights of scene objects by categorizing every point within the point cloud. However, unlike 2-D image pixels organized in the ordered grids, 3-D point cloud is disordered and unequally distributed across a large-scale 3-D space. Besides, scene objects in the point cloud display variations in structure and size, and they may encounter severe occlusions, and diverse overlaps or data defects. The aforementioned objective factors bring a great challenge for 3-D point cloud semantic segmentation.

In the last few years, deep learning network models have gained popularity in 3-D point cloud semantic segmentation due to their powerful capacity of feature learning and parameter sharing [4]. Many novel or enhanced methods have been put forward successively, including projection-based methods [5], [6], [7], voxel-based methods [8], [9], [10], and point-based methods [11], [12], [13], [14], [15]. In contrast to projection-based methods and voxel-based methods, point-based methods circumvent the need for data conversion procedures, such as point cloud projection and voxelization, which allows deep learning networks to be applied directly on the raw point cloud. PointNet [11] directly learned per-point features on the irregular 3-D point cloud data, achieving end-to-end point cloud semantic segmentation for the first time. However, PointNet lacks consideration of local feature extraction on the point cloud, thus limiting its ability to handle details and generalize to complex scenes. To this end, most of the follow-up methods begin to focus on designing the advanced feature extraction operators to capture the local geometric structures and details from the point cloud to improve performance. For example, PointNet++ [12] designed multiple set abstraction modules to hierarchically aggregate discriminative point features from different subregions. RandLA-Net [15] proposed a residual aggregation block to capture local

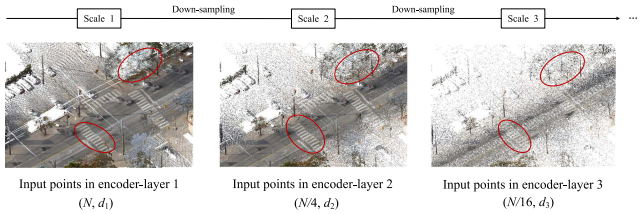

Fig. 1. Schematic illustration of down-sampling operations in the encoder of network. (N, d) denotes the number of input points and feature channels, respectively.

distinctive features of every point by increasing its receptive fields. Moreover, to improve the processing efficiency of large-scale scene 3-D point clouds, many existing methods typically adopt multiple down-sampling operations in the encoder of network, as depicted in Fig. 1, to progressively reduce the number of input points. Although these methods have shown promising overall segmentation results, some small or occluded objects that consists of few points are still hard to be accurately segmented. The main reasons for this may lie in two aspects: 1) some small or occluded objects (i.e., the road marking and the telegraph pole denoted by red circles in Fig. 1) are usually attached to or overlapped with the adjacent objects that share similar colors or structures, which can easily make the network confuse them in the lack of necessary detailed and contextual information and 2) as shown in Fig. 1, some details of object boundaries and structures are lost during the point sampling process, causing the networks more difficult to distinguish small or occluded objects with limited sampled point information. To address the above issues, we not only construct a fundamental point cloud feature extraction operator but also introduce a novel network architecture to effectively exploit multiscale feature propagation and achieve highly accurate semantic segmentation of 3-D point clouds.

In fact, multiscale feature propagation is already extensively utilized in most of 2-D segmentation networks, typically with the image feature pyramid [16], [17] and spatial pyramid pooling [18], [19]. It is noteworthy that most existing 3-D segmentation networks for large-scale point clouds [12], [15], [20], [21], [22], generally employ a U-shape encoder–decoder network architecture, which is derived from the classical 2-D segmentation network of U-Net [23], to exploit multiscale point features gathered from the encoder to enrich the geometric details of decoding features with skip connections. However, the normal U-shape encoder–decoder network architecture only allows the integration of the same-scale encoded features with the corresponding decoding features at each layer through the simple skip connection, it fails to achieve effective cross-scale information interaction to fully leverage multiscale geometric point features that are essential for distinguishing objects with diverse sizes in complex scenes. Motivated by the advances in 2-D image segmentation [24], we refine the normal encoder–decoder architecture to a nested U-shape architecture (NUA) and ingeniously apply it in 3-D semantic segmentation for large-scale point clouds.

In this article, a nested U-shape encoder–decoder deep neural network, namely PointNest, is proposed to perform multiscale point feature propagation for 3-D point cloud semantic segmentation. The encoder of PointNest learns and propagates point features with nested feature aggregation blocks, each of which captures local complex geometric structures of each point in a graph-like local region. The multiscale geometric features learned from the nested blocks between encoder and decoder are then horizontally and vertically concatenated by compact connections to enhance cross-scale information interaction, which helps to handle occlusions and boost robustness of network to the complex scene variances. Our primary contributions are outlined below.

1) A refined local feature aggregation module of RandLA-Net is employed as the basic feature aggregation unit (FAU) to capture local significant geometric features of every point by expanding the receptive field through two stacked graph convolution operations.
2) A nested U-shape network architecture is constructed to enable the extracted multiscale point features to be fused and propagated across different depths to collect more geometric details and contextual information for the accurate pointwise semantic prediction.
3) A deep supervision (DS) strategy is introduced to supervise multiscale output predictions to accelerate multiscale point feature propagation in the whole nested network architecture for the efficient network training and further performance improvement.

II. Related Works

A. Point Cloud Semantic Segmentation Based on Deep Learning

Recent deep learning-based methods have achieved outstanding performance in point cloud semantic segmentation. The pioneering work of PointNet [11] directly learned per-point global features for semantic segmentation by utilizing shared multilayer perceptrons (MLPs) and the symmetrical function. However, PointNet handled every point independently without considering the local structures of point clouds, which makes it sensitive to noise and limits its applicability in complex scenes. To overcome it, the subsequent PointNet++ [12] adopted a hierarchical feature learning framework to extract multilevel local features with multiple sampling and grouping operators, but it is still hard to recognize fine-gained local patterns without considering relations between each point and its neighbors. To this end, some works [25], [26], [27] built a local graph-like structure for every point and simultaneously applied an attention mechanism [28] on its neighboring points, which promotes the extraction of local geometric features. Despite the above methods have shown satisfying segmentation performance on simple 3-D shapes and small-scale point clouds, most of them are unable to be applied on large-scale point clouds due to the intensive computational complexity of their network architecture designs. To achieve the efficient segmentation of large-scale point clouds, SPG [29] converted the raw point clouds into super graphs before utilizing deep neural network to conduct pointwise

prediction. PCT [30] took a different approach by preprocessing the large point clouds into regular voxels, while Boulch et al. [7] applied CNNs on multiple projected 2-D view images derived from the large-scale point cloud. To circumvent the need for computational preprocessing, an end-to-end U-shape network architecture that is generally utilized in 2-D image segmentation [31] has been introduced. This U-shape point cloud network architecture allows for sampling input points layer by layer to perform feature learning and propagation for pointwise semantic prediction.

B. Multiscale Feature Learning Based on U-shape Network Architecture

In the U-shape 2-D image segmentation network [23], [24], [32], the encoder reduces the size of input images with multiple pooling operators to learn and propagate multiscale pixel features along the encoder path. The decoder path then restores the feature resolution of encoded images for pixelwise semantic prediction. The middle skip connections are added to allow the decoder to access high-resolution features from the encoder. Based on the U-shape network architecture, many works are devoted to utilizing multiscale features for 2-D image semantic segmentation. PANet [32] constructed a bottom–top connection path to efficiently propagate multiscale features with better localization from shallow to deep layers. U-Net++ [24] rebuilt the network architecture of U-Net [23] by embedding nested convolution blocks in different layers and using densely connected skip connections to propagate multiscale features along horizontal direction. As far as 3-D semantic segmentation for large-scale point clouds, most existing methods [12], [15], [20], [21], [22] adopt a normal U-shape network architecture that utilizes simple skip connections to fuse same-scale point features between the corresponding encoding and decoding layers. They are limited in their ability to integrate cross-scale information from encoding and decoding layers at different network depths. This limitation hinders the capturing of contextual geometric details necessary for accurate segmentation under complex scenes. To enhance the ability cross-scale information interaction of the normal U-shape network, GADH-Net [33] introduced a dense hierarchical network architecture to achieve cross-scale feature fusion. Nie et al. [34] constructed a pyramid architecture to allow multiscale point information to propagate freely through upward and downward links between layers. RFFS-Net [35] employed a multilevel decoder in the network architecture to fuse cross-scale point features from the encoder with directed skip connections. Besides, some researchers combined different designed feature extractors to gather multiscale information to improve the network performance. Most feature extractors obtain multiscale point features by expanding receptive fields. PointWeb [36] proposed an adaptive feature adjustment module to adaptively weight the neighbors of each point and expand its receptive field according to the density of points. PointSIFT [37] introduced an orientation-encoding unit to learn and fuse multiscale features from the neighbors of eight directions. In addition, some works constructed multiscale graph structures and leveraged graph convolutional operations to obtain multiscale features. MSGCNN [38] developed a multiscale graph convolution operator to realize multiscale feature extraction over the lattice structure of point clouds. DenseKPNET [39] proposed a multilevel feature extraction module, which consists of dense connection-based graph convolution operators with multiscale kernel points to acquire discriminative contextual semantic information. Different from above multiscale feature learning methods, we perform efficient multiscale feature learning based on a novel nested U-shape network architecture in this study.

III. Methodology

In this section, the overview of the network architecture that learns and fuses deep multiscale point features with nested FAUs is first presented. Then, the details of the basic FAU are described. Finally, a DS strategy for the network training is illustrated.

A. Nested U-Shape Network Architecture

The proposed PointNest adopts a nested U-shaped network architecture with nested FAUs that are successively connected between the encoder and decoder paths at different depths. As illustrated in Fig. 2, the raw point cloud is first input into a shared fully connected layer to learn per-point feature. Then, the learned point features are further processed by several encoding and decoding layers to obtain abstract semantic features along a U-shape encoder–decoder path.

Along the encoder path, each layer combines an FAU to gather rich local geometric detailed information of every point, and a random sampling operation to decrease the spatial resolution of point features. After each encoding layer, a part of points is preserved (i.e., ($N \rightarrow N/4 \rightarrow N/16 \rightarrow N/64 \rightarrow N/256 \rightarrow N/512$), whereas the point feature channel is increased to ($8 \rightarrow 16 \rightarrow 64 \rightarrow 128 \rightarrow 256 \rightarrow 512$)). Symmetrically, the decoder path restores the spatial resolution of encoded features through the nearest neighbor interpolation and supervises multiscale output predictions in a bottom-to-top manner. Finally, two shared fully connected layers are introduced to generate semantic prediction labels of input points (N, n_{class}), each of which belongs to a specific class.

In particularly, as illustrated in Fig. 3, each FAU propagates and receives multiscale features from its preceding blocks across different feature resolutions. Define the output features of a block in PointNest as $F_{i,j,}$ where i and j represent the depth and width of current layer, respectively, so the learned features of the each block at second layer can be denoted as (1) shown at the bottom of the next page, where $\Psi[,]$ is the feature concatenation, FA() is feature aggregation in the block, and DS() and US() are the down sampling and up sampling, respectively.

Compared with the normal U-shape network architecture, as described in Fig. 4(a), our proposed network architecture further inserts nested blocks at different depths, as shown in Fig. 4(b). In our nested U-shape network architecture, the hierarchical point information collected along the encoder path can be horizontally propagated to the corresponding decoding layers through skip connections between nested blocks, so that

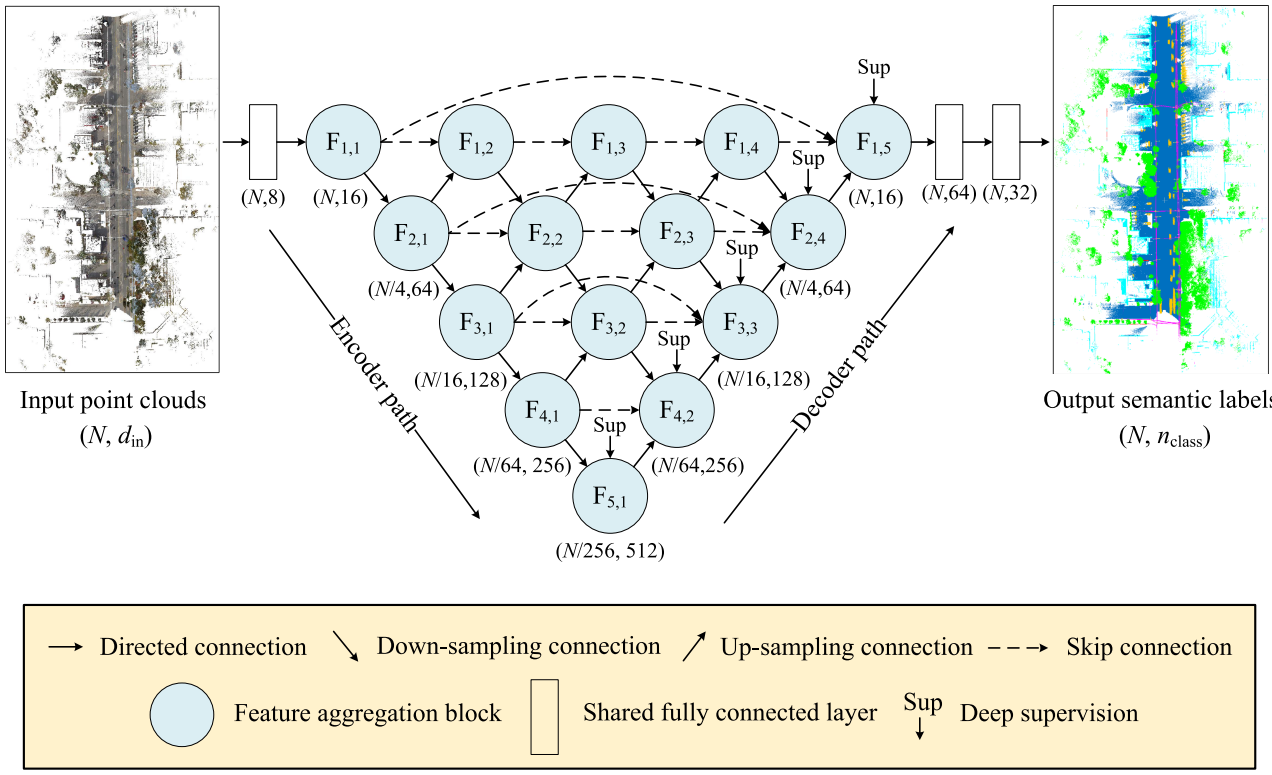

Fig. 2. Diagram of the nested U-shape network architecture. (N, d) below the block denotes the number of input points and feature channels, respectively.

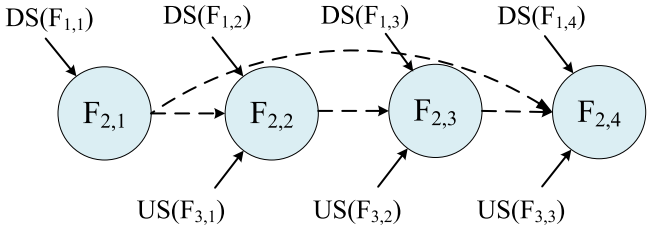

Fig. 3. Diagram of second layer nested blocks in PointNest.

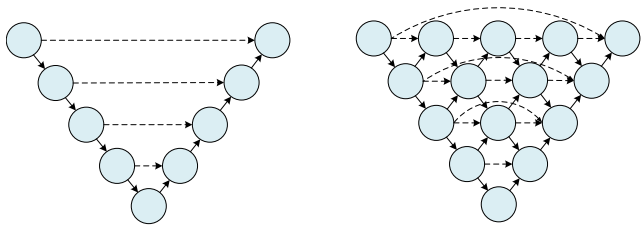

Fig. 4. Comparison with the normal U-shape network architecture. (a) Normal U-shape network architecture. (b) Our nested U-shape network architecture.

multiscale geometric details can be fully reused and fused for point cloud semantic segmentation. Besides, the vertical connections denoted by down-sampling and up-sampling arrows are added to enhance cross-scale information interaction. Notably, in our NUA, small-scale point features (i.e., $F_{1,1}$ and $F_{2,1}$) generated in the lower network depths generally contain rich geometric details (e.g., shape and boundary information) of objects in the point cloud, whereas large-scale point features (i.e., $F_{4,1}$ and $F_{5,1}$) contain more abstracted semantic information. Therefore, the cross-scale information interaction is crucial for integrating small-scale geometric details and large-scale semantic information during feature encoding, which helps to achieve accurate semantic prediction of objects at point level in complex scenes.

B. Feature Aggregation Unit

Although the down-sampling operations used along the encoder path can progressively reduce the size of input points to improve the efficiency of feature extraction, detailed geometric information can be lost as the spatial resolution of point features decreases, which is particularly problematic for incomplete and small objects with sparse points in the scene. To alleviate

$$F_{i,j} = \begin{cases} \psi[\mathrm{DS}(F_{i-1,j})] & i = 2,\ j = 1 \\ \psi\left[\mathrm{FA}(F_{i-1,j}), \mathrm{DS}(F_{i-1,j}), \mathrm{US}(F_{i+1,j-1})\right] & i = 2, j = 2 \text{ or } 3 \\ \psi\left[\mathrm{FA}(F_{i-1,j}), \mathrm{DS}(F_{i-1,j}), \mathrm{US}(F_{i+1,j-1}), F_{i,1}\right] & i = 2, j = 4 \end{cases} \tag{1}$$

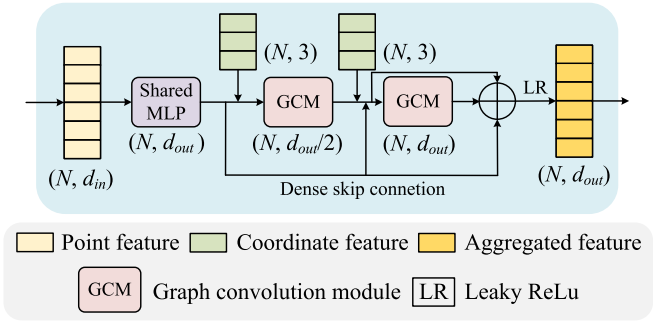

Fig. 5. Diagram of the FAU.

the negative impact caused by the down-sampling operation, a refined local feature aggregation module of RandLA-Net [15] is employed as the basic FAU of our proposed PointNest to capture local rich geometric features over multiscale sampled points. As illustration in Fig. 5, the FAU mainly consists of two stacked graph convolution modules (GCMs). In each GCM, 3-D coordinates (x, y, z) are repetitively introduced and processed together with the input point features for the local feature aggregation. Moreover, inspired by the DenseNet [40], we replace the residual connection utilized in the original module of RandLA-Net with a dense skip connection (DSC) positioned behind a shared MLP layer. This modification enables the FAU to concatenate all the outputs from two stacked GCMs, thereby further enhancing point feature propagation and improving the representation of local complex structures.

The details of GCM are illustrated in Fig. 6, which contains two key steps: local graph construction and graph attention pooling (GAP). Given an input point cloud set $H = \{h_1, h_2, \ldots, h_N\}$ and its corresponding point features $S_H = \{s_1, s_2, \ldots, s_N\} \in R^{N\times d}$. In the step of local graph construction, for each input point h_i, the GCM first uses a k nearest neighbor algorithm to search its k-top adjacent points $h_K = \{ h_{ij} \in \mathrm{R}^{N\times 3} \mid j = 1,2,\ldots,k\}$ and its corresponding neighboring point features $s_K = \{s_{ij} \in R^{N\times d} \mid j = 1,2,\ldots,k\}$ in a 3-D coordinate space. Then, for every point h_i and its k nearest neighboring points h_K, the local graph G_i is implicitly built in feature space as follows:

$$\begin{aligned} G_i &= (V_i, E_i) \\ V_i &= \left\{s_{ij} \cup s_i \in R^{N\times d} | j = 1, 2, \ldots, k\right\} \\ E_i &= \left\{e_{ij} = s_{ij} - s_i \in R^{N\times d} | j = 1, 2, \ldots, k\right\} \end{aligned} \tag{2}$$

where V_i and E_i are the node set and edge set, respectively, in the local graph G_i, - is the element subtraction, and e_{ij} is the edge feature calculated by the directed feature distance between the reference point s_i and its jth neighboring point feature s_{ij}.

Simultaneously, the relative position encoding (RPE) followed by a shared MLP is introduced to get local relative features $s^r{}_{ij}$ of the reference point h_i by attending its neighboring point h_{ij}, which can be denoted as follows:

$$s_i^r = \mathrm{MLP}\left([h_i, h_j, (h_{ij} - h_i), ||h_{ij} - h_i||]\right) \tag{3}$$

where h_i and h_{ij} are the 3-D coordinate points with x-y-z feature channels, [,] is the feature concatenation, $\|\cdot\|$ is the calculation of Euclidean distance, and - is the same as described in (2). It is noteworthy that the local relative features contain rich coordinate information, which can be integrated to assist the network to improve the performance. Then, each edge feature e_{ij} of the reference point h_i in the local graph is further combined with the corresponding relative feature $s^r{}_{ij}$ to obtain the enhanced edge feature $e*_{ij}$ as follows:

$$e*_{ij} = \left[e_{ij}, s_{ij}^r\right] \tag{4}$$

where [,] is the same as described in (3).

The step of GAP is utilized to aggregate each enhanced edge feature $e*_{ij}$ of the reference point h_i to extract its local discriminative geometric feature from the local graph. To this end, the powerful attention mechanism [28] is incorporated in the graph pooling to enable the network concentrate on the most important part of each enhanced edge feature and capture more significant neighboring features. To formulate the step of the GAP, given the set of enhanced edge features $E*_{ij} = \{e*_{i1}, e*_{i1}, \ldots, e*_{ij}\}$, the shared activation function $\sigma(\cdot)$ followed by a Softmax classifier is first introduced to the learn attention score α_{ij} of each enhanced edge feature as follows:

$$\alpha_{ij} = \mathrm{Softmax}\left(\frac{\exp\left(\sigma\left(e_{ij}^*, W\right)\right)}{\sum_{j=1}^{k}\exp\left(\sigma\left(e_{ij}^*, W\right)\right)}\right) \tag{5}$$

where $W \in R^{1\times d}$ is the learnable matrix. Then, each enhanced edge feature $e*_{ij}$ is multiplied with the corresponding learned α_{ij} and further aggregated as follows:

$$s_i^* = \sum_{j=1}^{k}\left(e_{ij}^* \cdot \alpha_{ij}\right) \tag{6}$$

where $s*_i$ is the aggregated features of the reference central point h_i in a local graph and $\cdot$ is the dot product. Through a weighted summation process on edge features in the local graph, the most important neighboring features are selected and fused into the reference central point, which enhances its distinguishability.

To further illustrate the capability of the proposed feature aggregation block, we visualize its process of local feature extraction on input point clouds in Fig. 7. In Fig. 7, the red point denotes the reference point feature, the dotted arrow denotes the direction of information flow, and dotted circles denote the receptive field of each input point. As seen from Fig. 7, the red central point receives local geometric details and contextual information from its k neighboring points after the first GCM and then its information perception range can be expended to k^2 neighboring points after the second GCM. Therefore, by stacking two GCMs to process each input point in a local graph, more useful neighboring information can be collected in a larger receptive field through the point feature propagation and aggregation.

C. Deep Supervision

To introduce the DS strategy [41] for the network training, the proposed PointNest adds 1×1 convolution layer followed

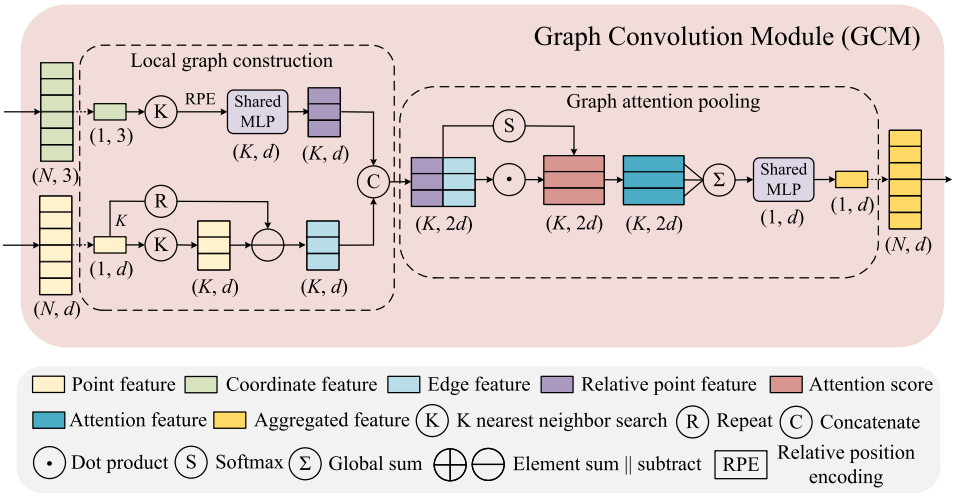

Fig. 6. Details of GCM in the basic FAU.

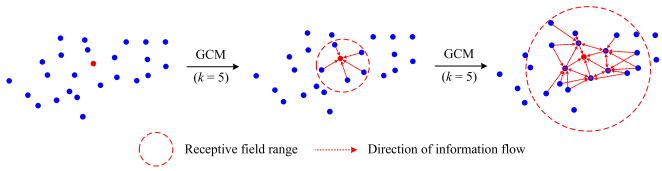

Fig. 7. Graphic illustration of local feature extraction in the basic FAU.

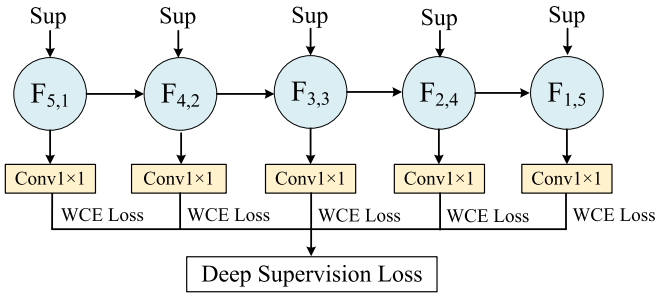

Fig. 8. Diagram of the DS in PointNest.

by a weighted cross entropy loss function (L_{WCE}) behind each block along the decoder path, as shown in Fig. 8. The final loss is calculated by using the output predictions from all depths so that each branch can be concurrently optimized through back propagation during the network training. In order to promote multiscale information propagation and obtain abstracted semantic information for pointwise prediction, a DS loss function (L_{DS}) is constructed to efficiently supervise the network training at different depths. The reference L_{WCE} is denoted as follows:

$$L_{\mathrm{WCE}} = -\sum_{i}^{N_s} w_i y_i \log(x_i) \tag{7}$$

$$w_i = \frac{\sum_{m=1}^{n_{\mathrm{class}}} N_m}{N_m + \varepsilon} \tag{8}$$

where N_s is the number of total sample points, x_i and y_i are the per-point prediction vector and ground truth vector respectively, w_i is the class balance coefficient of the ith sample point, n_{class} is the number of categories, N_m is the number of sample points belonging to the mth category, and ε is a constant coefficient used to avoid minimal denominator and set to 0.02 in this study.

Based on the L_{WCE}, the proposed L_{DS} can be denoted as follows:

$$L_{\mathrm{DS}} = L^{1}_{\mathrm{WCE}} + \mu \cdot \sum_{d=2}^{D} L^{d}_{\mathrm{WCE}} \tag{9}$$

where μ is an adjustment factor introduced to balance the two loss items, D is the number of the network layers and set to 5 in this study, ${L^1}_{\mathrm{WCE}}$ is the output loss after the FAU ($F_{1,5}$) at network layer 1, and ${L^d}_{\mathrm{WCE}}$ is the output loss from the units (i.e., $F_{1,5}$) at deeper network layer d. Different from the normal cross entropy loss function L_{CE} used only at network layer 1 in most existing methods [11], [14], [15], the proposed L_{DS} leverages an L_{WCE} to handle class imbalance and further applies it in other intermediate network depths to incorporate and backpropagate the auxiliary multiscale supervision information for the efficient network training.

IV. Experiments

In order to comprehensively verify the effectiveness of our proposed method, extensive experiments are performed on two large-scale urban point cloud datasets. Besides, ablation studies and analysis are also conducted to investigate the impact of different network designs and hyperparameter settings on the network performance.

A. Description of Datasets

S3DIS dataset [42]: This dataset comprises six indoor areas, totaling 271 rooms, and covering a combined area of 6020 m^2. It contains approximately 215 million labeled points, each with 3-D coordinates (*XYZ*), 3-D spectral information (RGB), and 3-D normalized location information, divided into 13 categories. During the experiment, the sixfold cross validation strategy [15] is used for network training and testing. The network input

TABLE I
QUANTITATIVE RESULTS WITH DIFFERENT METHODS ON S3DIS DATASET (AREA 5) (%)

Method	OA	mIoU	ceil.	floor	wall	beam	col.	wind.	door	table	chair	sofa	book.	board	clutter
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
DGCNN [45]	59.8	51.5	93.0	97.4	77.7	0.0	12.0	47.8	39.8	67.4	72.4	23.2	52.3	39.8	46.6
PointCNN [12]	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
PointWeb [36]	87.0	60.3	92.0	98.5	79.4	0.0	21.1	59.7	34.8	76.3	88.3	46.9	69.3	64.9	52.5
GACNet [25]	87.8	62.9	92.3	98.3	81.9	0.0	20.4	59.1	40.9	78.5	85.8	61.7	70.8	74.7	52.8
TG-Net [46]	88.5	57.8	93.3	97.6	78.0	0.0	9.3	57.0	39.4	83.4	76.4	60.6	41.8	58.7	55.3
RandLA-Net [15]	87.2	62.4	91.1	95.6	80.2	0.0	24.7	62.3	47.7	76.2	83.7	60.2	71.2	70.1	53.9
BAF-LAC [20]	87.6	63.3	91.2	97.3	81.6	0.0	27.9	59.3	49.5	78.0	87.2	63.4	66.5	69.6	51.1
BAAF-Net [47]	88.9	65.4	92.9	97.9	82.3	0.0	23.1	**65.5**	**64.9**	78.5	87.5	61.4	70.7	68.7	57.2
PAConv [48]	-	66.6	94.6	**98.6**	82.3	0.0	26.4	58.0	60.0	**89.7**	80.4	74.3	69.8	73.5	57.7
LGGCM [49]	88.8	63.3	**94.8**	98.3	81.5	0.0	**35.9**	63.3	43.5	80.2	88.4	68.8	55.7	64.6	47.8
CGGC-Net [50]	87.2	62.4	93.5	96.8	79.6	0.0	24.2	62.2	32.6	74.5	86.5	71.1	69.9	67.6	53.3
NeiEA-Net [51]	88.5	66.1	92.9	97.4	83.3	0.0	34.9	61.8	55.3	78.8	86.7	77.1	69.5	67.9	54.2
PointNest (Ours)	**90.2**	**68.8**	93.7	98.2	**84.5**	0.0	34.3	63.8	59.9	81.4	**90.5**	**78.9**	**71.7**	**77.3**	**59.4**

The bold numbers denote the highest performance values.

consists of both the 3-D spatial coordinates (*XYZ*) and the 3-D spectral information (RGB) information from the point cloud data.

Toronto-3D dataset [43]: This dataset is comprised of approximately 80 million points and is divided into four distinct areas, covering nearly one kilometer of outdoor environments. Each point in the dataset is labeled into eight categories and contains 3-D spatial coordinates (*XYZ*) and 3-D spectral information (RGB). During the experiment, the area 2 of the dataset is used for testing while others for training. Only the 3-D spatial coordinates (*XYZ*) information of point cloud is used as the network input.

WHU-MLS dataset [44]. This dataset is collected by the mobile LiDAR scanner in the campus of Wuhan University, China. It consists of approximately 7.3 million points. Each point with 3-D spatial coordinates (*XYZ*) is manually labeled into seven categories. During the experiment, the dataset partitioning method for network training and testing refers to [44]. Only the 3-D spatial coordinates (*XYZ*) information of point cloud is used as the network input.

B. Implementation Details and Metrics

The proposed deep neural network in this study was developed on the open-source Tensorflow platform. During the stage of network training, the learning rate was established as 0.01 and the number of epochs as 100. To address the GPU memory limitations in different ablation experiments, the batch size was adjusted accordingly. During the stage of network testing, the trained network with best performance was used for the per-point semantic prediction on the testing data. To enable parallel training, a fixed set of 40 960 sampled points was fed into the network. All experiments were carried out on a single NVIDIA RTX3080Ti GPU, and the neighbor search range k of every point was set to 16.

To measure the performance of network in a quantitative manner, three standard semantic segmentation evaluation metrics [43] are adopted in this study, including per-class intersection over union (IoU), mean intersection over union (mIoU) for all classes, and overall accuracy (OA), which are calculated as follows:

$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{10}$$

$$\mathrm{mIoU} = \frac{1}{n_{\mathrm{class}}} \sum_{i=1}^{n_{\mathrm{class}}} \mathrm{IoU}_i \tag{11}$$

$$\mathrm{OA} = \frac{\mathrm{TP}}{n_{\mathrm{total}}} \tag{12}$$

where n_{class} is the number of categories, N_{total} is the total number of samples, and TP, FN, and FP are the number of true positives, false negative samples, and false positive samples, respectively.

C. Experiment Results and Analysis

Results on S3DIS dataset: Tables I and II give the quantitative comparison of PointNest with other mainstream methods. In terms of the area 5, the proposed PointNest has achieved the highest value for mIoU (68.8%), which are 6.4% and 1.9% higher than RandLA-Net and NeiEA-Net, respectively. Meanwhile, our network also outperforms others in per-class IoU for most segmented objects, such as walls, chairs, and boards. In the test of sixfold cross validation result, the pyramid architecture proposed by Nie et al. [34] yields the highest value of mIoU, and the proposed PointNest with the NUA also achieves a considerable overall segmentation accuracy, demonstrating the effectiveness of our proposed method.

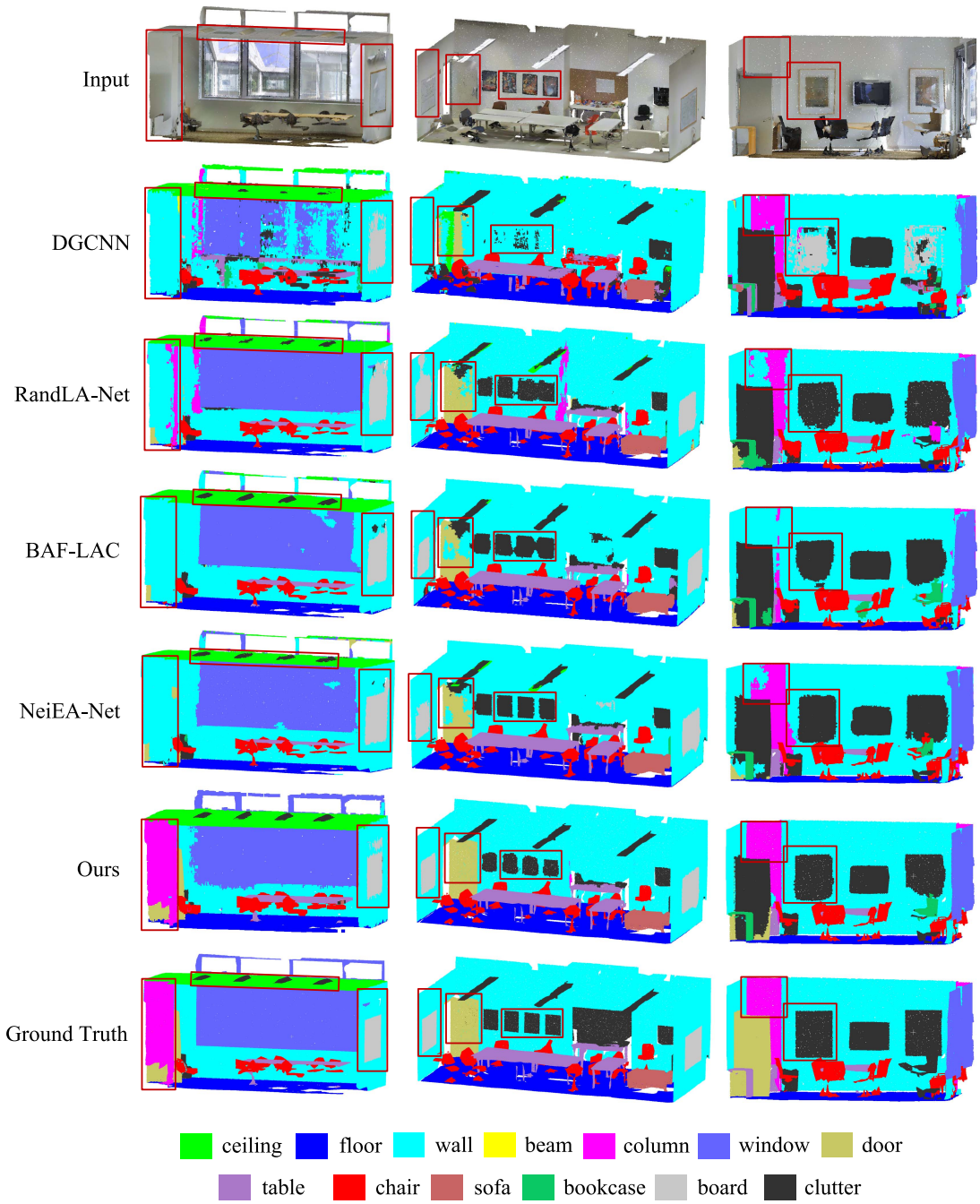

Fig. 9. Visual comparison with different methods on the S3DIS dataset.

Fig. 9 presents a visual comparison of results achieved by DGCNN, RandLA-Net, BAF-LAC, NeiEA-Net, and our proposed PointNest. As shown in Fig. 9, our PointNest obtains more accurate semantic segmentation results that are more closely aligned with the ground truth, and the majority of scene objects are well segmented. Whereas the compared DGCNN yields incomplete and discontinuous segmentation especially in the regions marked by red boxes. It struggles with accurately distinguishing objects with angular structures, such as chairs and tables, and fully extracting plane structures, such as windows, doors, and columns. This can be attributed to the limited network architecture of DGCNN, which hinders its ability to extract multiscale point features, resulting in suboptimal segmentation performance. Although RandLA-Net and BAF-LAC employ the standard U-shaped network architecture to hierarchically learn multiscale point features for efficient point segmentation and achieve relatively satisfactory performance, they still lack rich geometric details to produce accurate and complete results. NeiEA-Net showcases proficient segmentation outcomes at object boundaries by integrating local discriminative features and entire neighboring points. However, it falls to fully extracting plane structures, which include elements, such as doors and

TABLE II
QUANTITATIVE RESULTS WITH DIFFERENT METHODS ON S3DIS DATASET (SIXFOLD CROSS VALIDATION) (%)

Test area	Method	OA	mIoU
6-Fold	PointNet [11]	78.6	47.6
	DGCNN [45]	84.1	56.1
	PointCNN [13]	88.1	65.4
	RandLA-Net [15]	88.0	70.0
	BAF-LAC [20]	88.2	71.7
	SCF-Net [52]	88.4	71.6
	DenseKPNET [39]	**89.3**	71.9
	CGGC-Net [50]	88.0	70.2
	NeiEA-Net [51]	87.4	68.7
	Nie et al. [34]	-	**73.0**
	PointNest (Ours)	89.1	71.8

The bold numbers denote the highest performance values.

columns, attributed to its deficiency in incorporating overarching structural information. By contrast, our PointNest stands out from the other methods by introducing a novel NUA that effectively gathers multiscale geometric information from input point clouds. This allows the network to capture complete structure features of scene objects in cluttered indoor scenes, and thus resulting in more accurate segmentation results.

Results on Toronto-3D dataset: Table III gives the quantitative comparison between the proposed PointNest and other representative methods. We can observe that our network exhibits superior performance on OA (97.0%) and mIoU (74.7%), surpassing RandLA-Net by 2.6% and 3.3%, respectively. Moreover, our proposed PointNest outperforms most of the other methods in terms of per-class IoU, particularly in some small-scale categories, such as road marking, utility line, and pole.

Fig. 10 presents a visual comparison of DGCNN, RandLA-Net, BAF-LAC, NeiEA-Net, and our proposed PointNest. As seen in Fig. 10, the proposed PointNest performs significantly better alignment with the ground truth and produces fewer classification errors than the other two methods particularly in the subregions indicated by the red boxes. It becomes apparent that DGCNN struggles with segmenting small or incomplete objects with sparse points, including telegraph pole, road markings, and broken-down cars. This is potentially due to the fact that these smaller objects are often attached to or overlapped by other categorical objects that share similar geographical distributions and topological features and they lack distinct boundaries. Although RandLA-Net, BAF-LAC, and NeiEA-Net achieve satisfying segmentation performance on most objects, the multiple down-sampling operation leads to a shortage of rich geometric details, which makes them difficult to differentiate between road markings and the road. By contrast, upon observing the qualitative results, our PointNest outperforms the compared methods in segmenting small or incomplete objects and showcases remarkable resistance to nearby interference under complex scenes.

Results on WUH-MLS dataset: Table IV gives the quantitative comparison between the proposed PointNest and other advanced methods, including DGCNN, RandLA-Net, BAF-LAC, and NeiEA-Net. As given in Table IV, the OA (93.0%) and mIoU (62.7%) obtained by the proposed method are higher than those other comparative methods.

Fig. 11 presents a qualitative comparison of DGCNN, RandLA-Net, BAF-LAC, NeiEA-Net, and the proposed PointNest. As marked in the black boxes in Fig. 11, some of the smaller objects, such as pedestrians, are incorrectly classified by the comparative models. In addition, these models struggle to precisely and completely segment larger objects, such as buildings. By contrast, the scene objects are correctly classified by our PointNest, despite they vary in size and shape. Owe to the local discriminative feature extraction and multiscale feature propagation capability of the proposed PointNest, the spatial context of the scene objects is effectively characterized. Objects depicted in Fig. 11, including lamps within vegetation, cars with incomplete shapes, and fences under trees, are accurately classified by our method despite they are partially occluded. This robustly showcases the effectiveness of the proposed method.

D. Ablation Studies

In this section, we conduct extensive ablation studies to investigate the impact of different network design choices on the performance of our proposed PointNest. All the ablated experiments are conducted on the area 5 of the S3DIS dataset.

Ablation study for the main component of network: To verify the effectiveness of the PointNest, several ablation experiments are carried out on its three main components including FAU, NUA, and DS. In the ablation experiments, the FAU can be replaced by the local feature aggregation module of RandLA-Net, whereas the NUA can be replaced by the normal U-shape architecture, and the DS can be replaced by the normal training manner with cross entropy loss function. Note that the DS is based on the use of the NUA, and removing the NUA will also remove the DS.

Table V quantitatively illustrates the results of the ablation study for main components of PointNest. As given in Table V, the removal of the basic FAU in model a_2 resulted in a small decline in semantic segmentation performance, with the mIoU score dropping by 3.8%. When compared with the model a_3, the mIoU score of model a_4 delivered an increase of 2.8%, this is mainly because the nested connections between the basic blocks in the NUA promote the multiscale information propagation for the performance improvement. In contrast, the proposed model a_5 with full components achieved best performance with the mIoU score of 68.8%, thereby demonstrating the effectiveness of its main components.

Fig. 12 presents the visual comparison of PointNest with different components. The segmented areas marked by the red circles indicate that both models a_1 and a_2 failed to obtain

TABLE III
QUANTITATIVE RESULTS WITH DIFFERENT METHODS ON TORONTO-3D DATASET (AREA 2) (%)

Method	OA	mIoU	road	rdmk	nature	building	utline	pole	car	fence
PointNet [11]	81.9	31.2	88.6	0.0	63.2	44.8	13.4	0.0	59.8	3.3
PointNet ++ [12]	92.6	59.5	92.9	0.0	86.1	82.2	60.9	62.8	76.4	14.4
TGNet [41]	94.1	61.3	93.5	0.0	90.8	81.6	65.3	62.9	88.7	7.9
DGCNN [45]	94.2	61.7	93.9	0.0	91.3	80.4	62.4	62.3	88.3	15.8
MS-PCNN [46]	90.0	65.9	93.8	3.8	93.5	82.6	67.8	71.9	91.1	22.5
GAANet [53]	94.6	67.5	**94.9**	**29.3**	94.9	85.8	77.8	71.2	71.3	14.4
KPConv [54]	95.4	69.1	94.6	0.1	96.1	91.5	87.7	**81.6**	85.7	15.7
MS-TGNet [38]	95.7	70.5	94.4	17.2	95.7	88.8	76.0	73.9	**94.2**	23.6
RandLA-Net [15]	95.4	71.4	94.8	0.0	95.3	**92.6**	86.4	71.4	90.7	**40.4**
BAF-LAC [20]	95.4	70.2	94.8	0.0	95.8	92.3	80.2	73.8	90.5	33.8
NeiEA-Net [51]	95.3	68.5	94.7	0.0	95.2	89.1	79.6	76.5	93.3	19.6
PointNest (Ours)	**97.0**	**74.7**	91.0	27.9	**96.2**	89.5	**88.3**	78.6	91.1	35.1

The bold numbers denote the highest performance values.

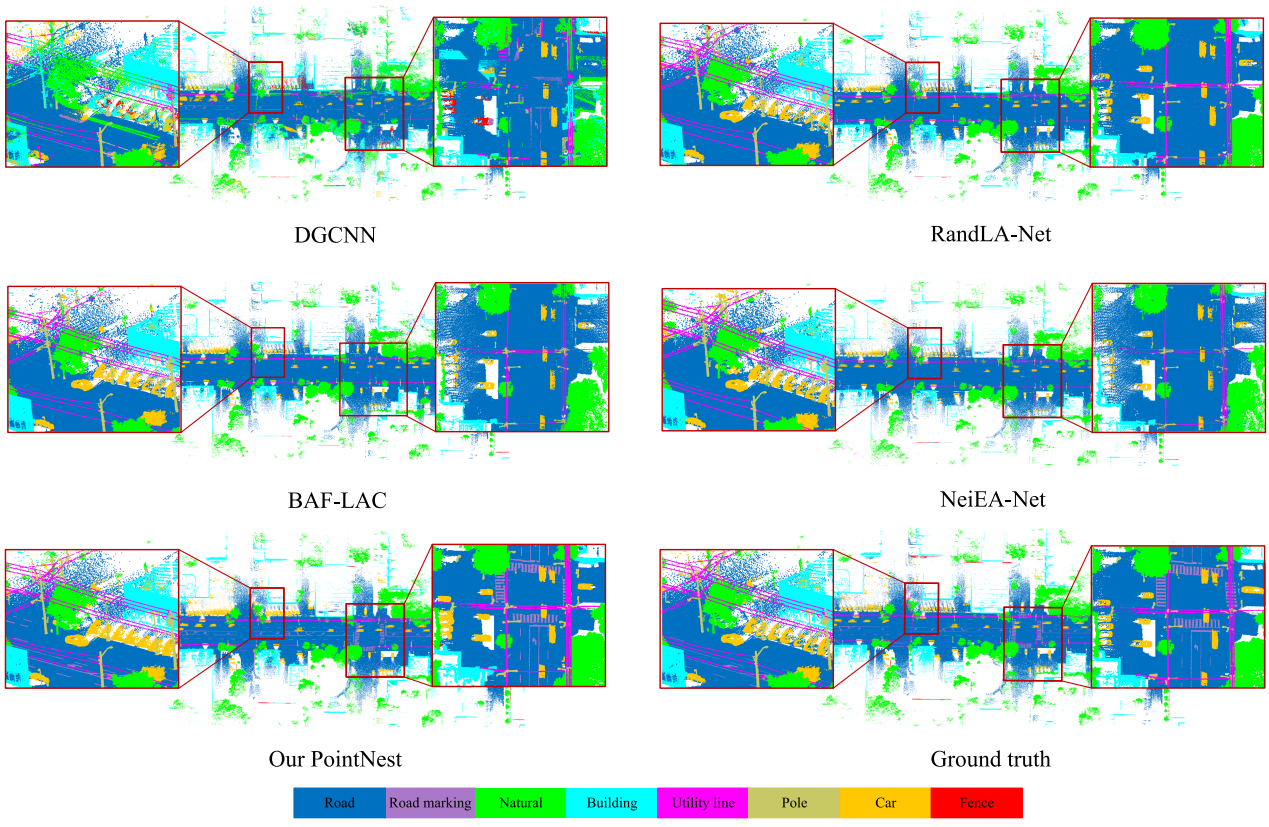

Fig. 10. Visual comparison with different methods on the Toronto-3D dataset.

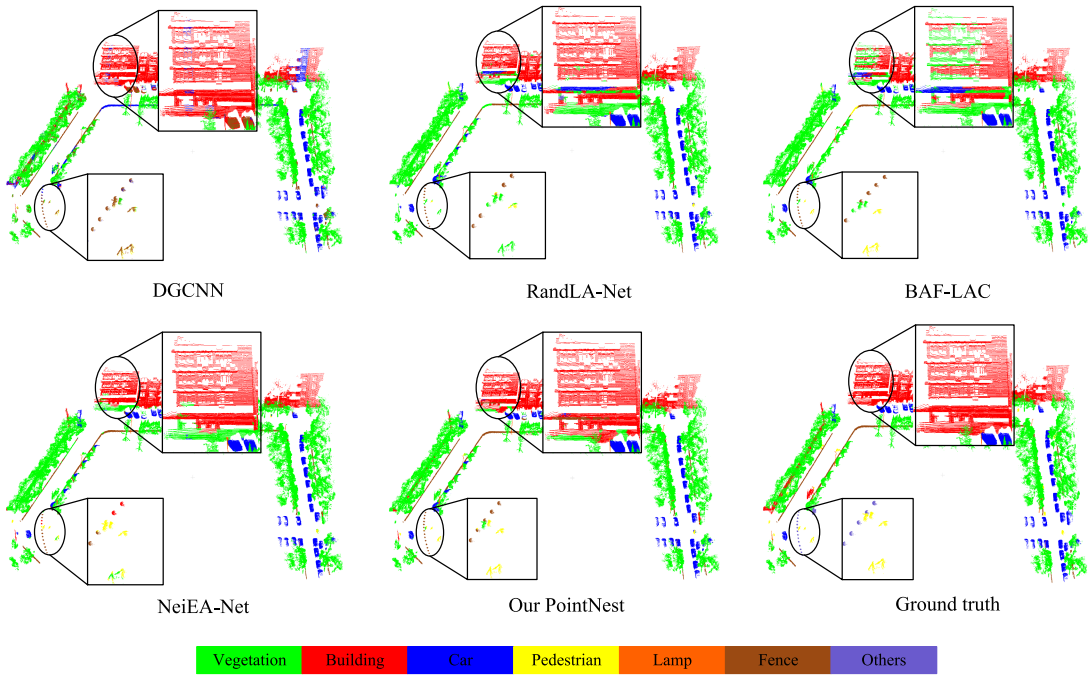

Fig. 11. Visual comparison with different methods on the WHU-MLS dataset.

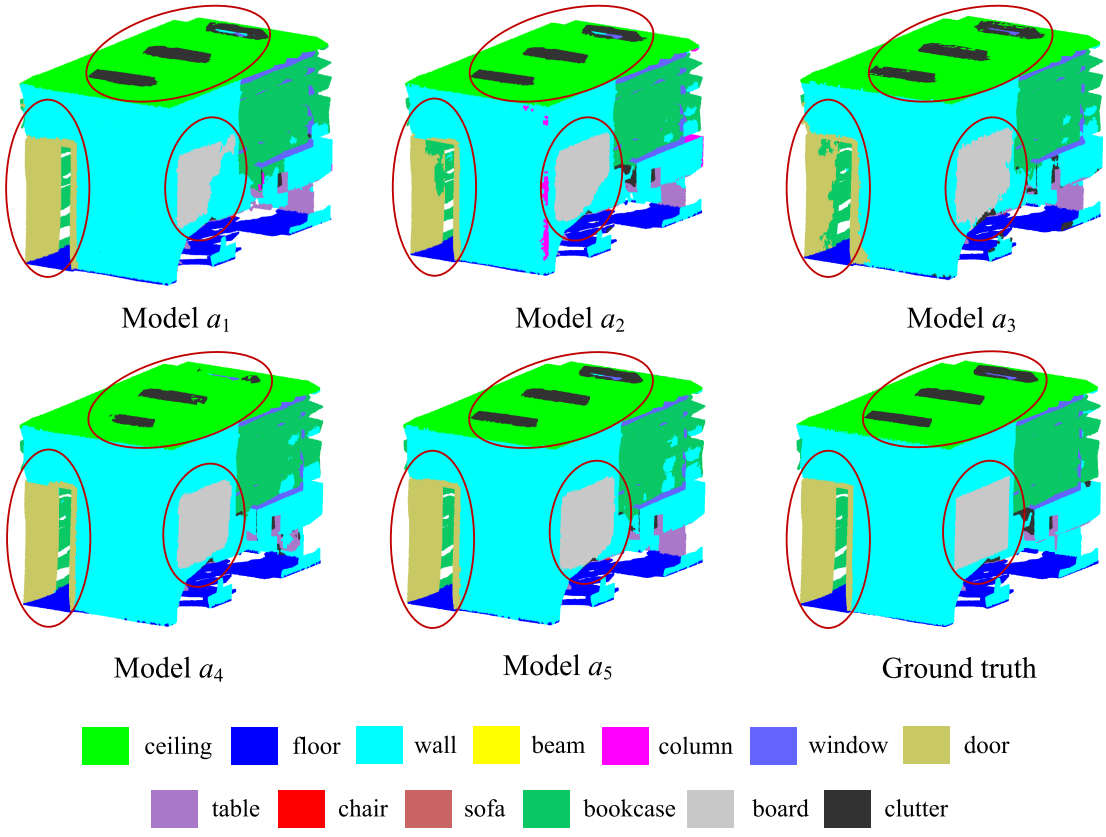

Fig. 12. Visual comparison of PointNest with different components.

TABLE IV
QUANTITATIVE RESULTS WITH DIFFERENT METHODS ON WHU-MLS DATASET (%)

Method	OA	mIoU	veg.	buil.	car	ped.	lamp	fence	others
PointNet [11]	-	27.8	65.4	49.5	43.9	3.9	0.3	29.4	2.2
PointNet ++ [12]	-	52.8	72.3	56.0	57.2	**25.4**	26.8	58.7	**3.4**
DGCNN [45]	86.8	45.5	85.1	80.4	66.6	6.1	43.3	37.0	0.0
RandLA-Net [15]	84.1	52.2	71.5	70.0	**90.0**	7.4	53.0	73.7	0.0
BAF-LAC [20]	90.7	56.5	85.3	88.7	69.9	15.6	47.0	67.0	0.2
NeiEA-Net [51]	88.8	54.8	80.3	85.1	88.7	22.4	36.7	70.5	0.0
PointNest (Ours)	**93.0**	**62.7**	**90.1**	**92.4**	89.2	23.5	**61.8**	**82.0**	0.0

The bold numbers denote the highest performance values.

TABLE V
ABLATION STUDY FOR MAIN COMPONENTS OF POINTNEST

Model	FAB	NUA	DS	mIoU (%)
a_1		✓		64.2
a_2		✓	✓	65.0
a_3	✓			62.0
a_4	✓	✓		65.2
a_5	✓	✓	✓	**68.8**

The bold number denote the highest performance value.

TABLE VI
ABLATION STUDY FOR DIFFERENT DESIGN STRATEGIES OF FAUS

Model	Design strategy in FAU	mIoU (%)
b_1	with only RPE	64.3
b_2	with only DSC	58.6
b_3	with only GAP	57.3
b_4	with DSC and GAP	59.8
b_5	with DSC and RPE	64.2
B_6	with RPE and GAP	64.5
B_7	with full of strategies (Ours)	**68.8**

The bold number denote the highest performance value.

TABLE VII
ABLATION STUDY FOR DIFFERENT CONNECTION MODES

Model	Connection mode	mIoU (%)
c_1	with only horizontal links	38.5
c_2	with horizontal and downward links	42.0
c_3	with horizontal and upward links	66.7
c_4	with all type of links (Ours)	**68.8**

The bold number denote the highest performance value.

smooth boundaries for plane structures, such as the door and board, due to the lack of considering local contextual relationship between neighboring points in the local feature aggregation module of RandLA-Net. However, it should be noted that even model a_3 was unable to achieve a complete segmentation of the door, board, and clutter, and its segmentation boundaries were relatively rough. Due to the adoption of NUA, the model a_4 is able to gather more intricate geometric details, thereby enhancing the boundary segmentation. In contrast, the proposed model a_5 achieved high-quality segmentation results by leveraging both NUA and DS.

Ablation study for the design strategy of the basic unit: To further validate the effectiveness of the designed FAU in PointNest, several ablation experiments are conducted on its three design strategies, including RPE, DSC, and GAP.

Table VI gives the quantitative results of the ablation study for different design strategies of FAU. As seen from Table VI, models b_{2-4} suffered a significant decline in segmentation performance due to the lack of RPE. The primary reason for this is that the RPE provides important relative position information that significantly enhances the local geometric feature representation of every input point. When compared with model b_1, model b_5 performs significantly better, owing to its effective utilization and propagation of point features under the effect of DSC. Furthermore, model b_6 also enhances performance by leveraging GAP to perform reliable and robust neighboring feature aggregation on each input point. In contrast, the proposed model b_7 yields the best overall performance with full of design strategies in FAU.

Ablation study for the connection mode of network architecture: To further showcase the effectiveness of the proposed NUA of PointNest, several ablation experiments are performed on its different connection modes. The ablated networks with different connection modes are presented in Fig. 13.

As given in Table VII, due to the absence of information interaction between the cross layers, the use of only horizontal links in the nested blocks of model c_1 results in the gradual loss of significant geometric detail information during the multiscale feature propagation process. Hence, model c_1 delivered the

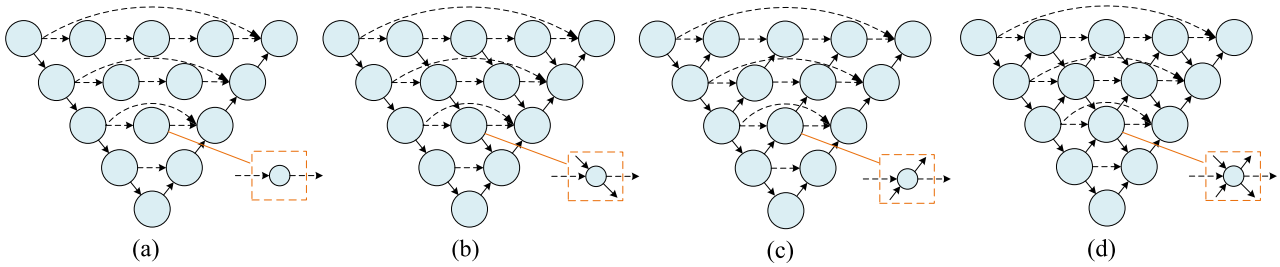

Fig. 13. PointNest with different connection modes. (a) PointNest c_1. (b) PointNest c_2. (c) PointNest c_3. (d) PointNest c_4.

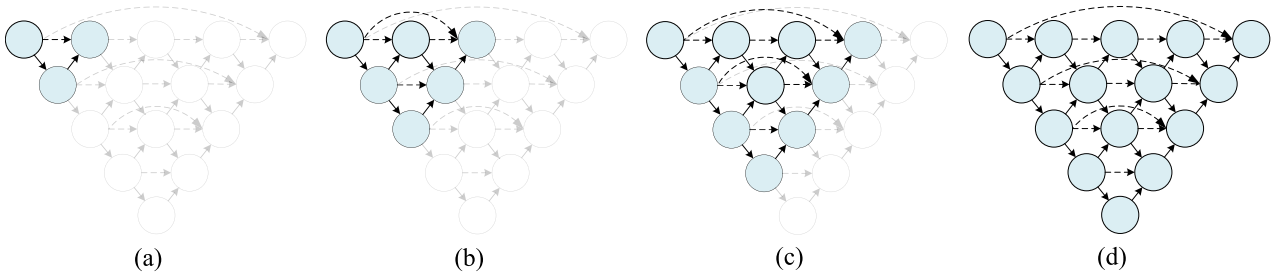

Fig. 14. PointNest with different connection modes. (a) PointNest d_1. (b) PointNest d_2. (c) PointNest d_3. (d) PointNest d_4.

poorest performance with a low mIoU score of 38.5%. Despite the inclusion of downward links, model c_2 was still unable to achieve satisfying results due to the limited point information increment caused by the successive down-sampling operations. By incorporating both horizontal and upward links, the model c_3 can propagate and receive more multiscale point information as the resolution of point features gradually increases during the successive up-sampling operations. Therefore, it obtains a relative considerable result. In contrast, our proposed model c_4 comprehensively incorporates all types of links to sufficiently capture geometric details through cross-scale up-sampling and down-sampling operations, thereby enhancing the recognition ability of complex objects and obtaining the best segmentation performance.

Ablation study for the network depth: In order to explore the impact of network depth on the performance of PointNest, several ablation experiments are performed on its different network depths. All ablated networks are set as: PointNest with depth 2 (d_1), PointNest with depth 3 (d_2), PointNest with depth 4 (d_3), and PointNest with depth 5 (d_4). The ablated networks with different network depths are shown in Fig. 14.

Table VIII gives quantitative results of ablation study for different network depths. As can be seen in Table VIII, the score of mIoU shows an increasing trend as the network depth increases. This is mainly because the feature learning ability of the network is enhanced with an increase in depth. Furthermore, model d_4 still achieves a satisfying performance when the network depth is set to 4. It is worth noting that the number of parameters in model d_3 is only about a quarter of that in model d_4. Our proposed model d_4 achieves the highest mIoU score of 68.8% when the network depth is set to 5. Due to the computational memory limitation, we are unable to perform ablation experiments on the network with deeper depths.

TABLE VIII
ABLATION STUDY FOR NETWORK DEPTH OF POINTNEST

Model	Network depth	Param (M)	mIoU (%)
d_1	depth 2	0.20	33.2
d_2	depth 3	1.07	37.6
d_3	depth 4	1.52	62.0
d_4	depth 5	6.16	**68.8**

The bold number denote the highest performance value.

TABLE IX
RESULTS OF POINTNEST WITH DIFFERENT LOSS FUNCTIONS

Method	mIoU (%)
PointNest with L_{CE}	62.6
PointNest with L_{WCE}	63.1
PointNest with L_{CE+DS}	65.5
PointNest with L_{WCE+DS} (L_{DE})	**68.8**

The bold number denote the highest performance value.

E. Loss Function Analysis

To further verify the effectiveness of the proposed deep supervision loss function L_{DS}, several representative loss functions are selected to perform comparative experiments on the S3DIS dataset, including the L_{CE} loss, the L_{WCE} loss, and $L_{\mathrm{CE+DS}}$ loss.

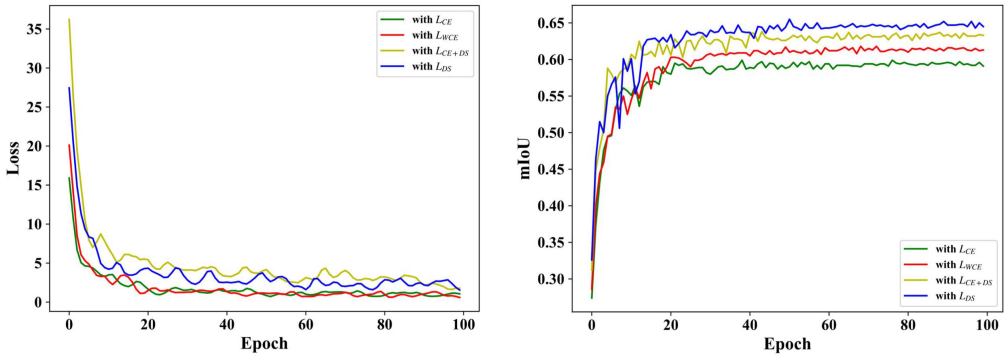

Fig. 15. Training process of PointNest with different loss functions. (a) Training loss value curve. (b) Training mIoU value curve.

TABLE X
DIFFERENT MODEL SIZE AND EFFICIENCY COMPARISON

Method	Param (Million)	FLOPs (10^7)	Inference time (S)	mIoU (%)
DGCNN [44]	**1.84**	67.9	-	51.5
RandLA-Net [15]	4.99	2.64	**216.8**	62.4
BAAF-Net [46]	4.97	2.97	252.4	65.4
BAF-LAC [20]	11.64	6.62	287.1	63.3
NeiEA-Net [50]	4.87	3.40	266.9	66.1
PointNest (Ours)	6.16	**1.86**	313.7	**68.8**

The bold numbers denote the highest performance values.

Table IX gives the quantitative comparison of the proposed method with different loss functions. As given in Table IX, PointNest with our proposed L_{DS} loss outperforms the second-best method by 3.3% in the mIoU score. PointNest utilizing either L_{CE} or L_{WCE}, solely oversees the output prediction signal from the nested blocks in the first layer. This method, however, falls short of fully harnessing the output predictions from intermediate layers to effectively enhance the ultimate prediction outcomes throughout the network training process. Consequently, this limitation leads to a decline in the overall performance.

Fig. 15 further presents the visual comparison of the training process of PointNest with different loss functions. As seen from Fig. 15(a), when the network training reaches approximately 20 epochs, the use of $L_{\mathrm{CE+DS}}$ and our proposed L_{DS} loss with DS strategy results in a faster convergence speed compared with the other two loss functions. However, their final loss values are unable to reach lower levels as they calculate multiscale output predictions, and their initial loss values are already high. Simultaneously, as depicted in Fig. 15(b), our proposed L_{DS} loss yields a more rapid increase in mIoU value when the network training progresses to approximately 20 epochs, which further demonstrates the effectiveness of the proposed L_{DS} loss function.

F. Computation Efficiency Analysis

To analyze the computation requirements and efficiency of the PointNest, we quantified the number of trainable network parameters and calculated the floating-point operations (FLOPs) that measures the complexity of network. Besides, we counted total inference time of the testing samples on the S3DIS dataset.

As given in Table X, although DGCNN consumes the lowest memory requirement of only 1.84M parameters, it demands a peak processing load of 6.79×10^8 FLOPs and yields poorest performance. In contrast, the proposed PointNest consumes modest computational resources but achieves best overall performance. Despite having the lowest model complexity, the inference time of the network testing using our proposed model is the longest, as it involves frequent addition and concatenation operations in the nested block, leading to increased memory access time. Furthermore, due to the variations in network architecture implementation, it is not possible to compare the inference time of DGCNN with other methods that have been tested consistently.

V. CONCLUSION

In this study, we propose a novel nested U-shape deep network, PointNest, for semantic segmentation of 3-D point clouds in both indoor and outdoor environments. The proposed PointNest utilizes horizontal and vertical connections to link up the basic FAUs at different layers. The nested U-shape network architecture facilitates the propagation and fusion of supervised multiscale information from the encoder path to the decoder path, which enables the network to comprehensively exploit hierarchical geometric features and further enhance the cross-scale

information interaction for the accurate pointwise prediction. Both quantitative and qualitative experimental results reveal that the proposed PointNest can effectively leverage multiscale information to distinguish complex objects within point clouds. Besides, the ablation studies and analysis on different network designs and choices show substantial improvements in performance. Our proposed method can directly process large-scale point clouds and has the potential to provide valuable 3-D semantic information for various real-world applications, such as urban planning, 3-D semantic map construction, and environmental monitoring.

In the future, we will continue to work on the refinement of our proposed nested U-shape network architecture and try to reduce the computational resource consume with the model pruning technology. By using fewer critical nested units and relatively lower network layers to achieve a more lightweight network for the efficient point cloud semantic segmentation. Moreover, the weakly supervised semantic segmentation on point cloud is another research direction, which can realize the semantic prediction of large-scale point clouds with a small number of manually labeled point samples and contributes to the network model migration.

ACKNOWLEDGMENT

We gratefully acknowledge the Stanford University, the University of Waterloo, and the Wuhan University for providing the experimental datasets. We also sincerely acknowledge the Editor, Associate Editor, and Reviewers for their valuable comments and suggestions on this work.

REFERENCES

[1] X. Wang, Y. Mizukami, M. Tada, and F. Matsuno, "Navigation of a mobile robot in a dynamic environment using a point cloud map," *Artif. Life Robot.*, vol. 26, no. 1, pp. 10–20, 2021.

[2] X. Yue, B. Wu, S. A. Seshia, K. Keutzer, and A. L. Sangiovanni-Vincentelli, "A LiDAR point cloud generator: From a virtual world to autonomous driving," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2018, pp. 458–464.

[3] C. Mallet, F. Bretar, M. Roux, U. Soergel, and C. Heipke, "Relevance assessment of full-waveform LiDAR data for urban area classification," *ISPRS J. Photogrammetry Remote Sens.*, vol. 66, no. 6, Supplement, pp. S71–S84, 2011.

[4] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep Learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4338–4364, Dec. 2021.

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

[6] 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.

[7] A. Boulch, J. Guerry, B. Le Saux, and N. Audebert, "SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks," *Comput. Graph.*, vol. 71, pp. 189–198, 2018.

[8] A. X. Chang et al., "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.

[9] 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.*, 2015, pp. 922–928.

[10] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, "SEGCloud: Semantic segmentation of 3D point clouds," in *Proc. Int. Conf. 3D Vis.*, 2017, pp. 537–547.

[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/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85.

[12] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 5099–5108, 2017.

[13] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," 2018, *arXiv:1801.07791*.

[14] 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. 9613–9622.

[15] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, and Z. Wang, "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11108–11117.

[16] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical multi-scale attention for semantic segmentation," 2020, *arXiv:2005.10821*.

[17] O. Ostapenko, M. Puscas, T. Klein, P. Jahnichen, and M. Nabi, "Learning to remember: A synaptic plasticity driven framework for continual Learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11321–11329.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[19] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2881–2890.

[20] 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, Apr. 2021.

[21] Z. Zeng, Y. Xu, Z. Xie, W. Tang, J. Wan, and W. Wu, "LEARD-Net: Semantic segmentation for large-scale point cloud scene," *Int. J. Appl. Earth Observation Geoinf.*, vol. 112, 2022, Art. no. 102953.

[22] 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. 14504–14513.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, 2015, pp. 234–241.

[24] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: Redesigning skip connections to exploit multiscale features in image segmentation," *IEEE Trans. Med. Imag.*, vol. 39, no. 6, pp. 1856–1867, Jun. 2020.

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

[26] T. Jiang, J. Sun, S. Liu, X. Zhang, Q. Wu, and Y. Wang, "Hierarchical semantic segmentation of urban scene point clouds via group proposal and graph attention network," *Int. J. Appl. Earth Observation Geoinf.*, vol. 105, 2021, Art. no. 102626.

[27] H. Zhou, Y. Feng, M. Fang, M. Wei, J. Qin, and T. Lu, "Adaptive graph convolution for point cloud analysis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 4965–4974.

[28] A. Vaswani et al., "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 5998–6008, 2017.

[29] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4558–4567.

[30] S. Chen, S. Niu, T. Lan, and B. Liu, "PCT: Large-scale 3D point cloud representations via graph inception networks with applications to autonomous driving," in *Proc. IEEE Int. Conf. Inf. Process.*, 2019, pp. 4395–4399.

[31] G. Du et al., "Medical image segmentation based on U-net: A review," *J. Imag. Sci. Technol.*, vol. 64, pp. 1–12, 2020.

[32] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation Network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8759–8768.

[33] W. Li, F. D. Wang, and G. S. Xia, "A geometry-attentional network for ALS point cloud classification," *ISPRS J. Photogrammetry Remote Sens.*, vol. 164, pp. 26–40, Jun. 2020.

[34] D. Nie, R. Lan, L. Wang, and X. Ren, "Pyramid architecture for multi-scale processing in point cloud segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17263–17273.

[35] Y. Mao et al., "Beyond single receptive field: A receptive field fusion-and-stratification network for airborne laser scanning point cloud classification," *ISPRS J. Photogrammetry Remote Sens.*, vol. 188, pp. 45–61, Jun. 2022.
[36] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, "PointWeb: Enhancing local neighborhood features for point cloud processing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5565–5573.
[37] 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*.
[38] M. Xu, W. Dai, Y. Shen, and H. Xiong, "MSGCNN: Multi-scale graph convolutional neural network for point cloud segmentation," in *Proc. IEEE 5th Int. Conf. Multimedia Big Data*, 2019, pp. 118–127.
[39] 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, Apr. 2022, Art. no. 5702913.
[40] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer, "DenseNet: Implementing efficient ConvNet descriptor pyramids," 2014, *arXiv: 1404.1869*.
[41] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Proc. 18th Int. Conf. Artif. Intell. Statist.*, 2015, pp. 562–570.
[42] I. Armeni et al., "3D semantic parsing of large-scale indoor spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1534–1543.
[43] 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. 797–806.
[44] L. Wang, Y. Huang, J. Shan, and L. He, "MSNet: Multi-scale convolutional network for point cloud classification," *Remote Sens.*, vol. 10, no. 4, Apr. 2018, Art. no. 4.
[45] 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.
[46] Y. Li, L. Ma, Z. Zhong, D. Cao, and J. Li, "TGNet: Geometric graph CNN on 3-D point cloud segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3588–3600, May 2020.
[47] 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.
[48] M. Xu, R. Ding, H. Zhao, and X. Qi, "PAConv: Position adaptive convolution with dynamic kernel assembling on point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3173–3182.
[49] 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.*, early access, Mar. 14, 2022, doi: 10.1109/TNNLS.2022.3155282.
[50] X. Wang, J. Yang, Z. Kang, J. Du, Z. Tao, and D. Qiao, "A category-contrastive guided-graph convolutional network approach for the semantic segmentation of point clouds," *Int. J. Appl. Earth Observation Geoinf.*, vol. 16, pp. 3715–3729, 2023.
[51] Y. Xu et al., "NeiEA-NET: Semantic segmentation of large-scale point cloud scene via neighbor enhancement and aggregation," *Int. J. Appl. Earth Observation Geoinf.*, vol. 119, May 2023, Art. no. 103285.
[52] 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.
[53] J. Wan, Y. Xu, Q. Qiu, and Z. Xie, "A geometry-aware attention network for semantic segmentation of MLS point clouds," *Int. J. Geographical Inf. Sci.*, vol. 37, no. 1, pp. 138–161, 2023.
[54] 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.

Jie Wan received the M.S. degree in surveying and mapping engineering in 2020 from the China University of Geosciences, Wuhan, China, where he is currently working toward the Ph.D. degree in geomatics with the Key Laboratory of Geological and Evaluation of Ministry of Education.

His research interests include deep learning, 3-D scene understanding, 3-D semantic segmentation, and point cloud analysis and process.

Ziyin Zeng received the M.S. degree in electronic information from the China University of Geosciences, Wuhan, China, in 2023. He is currently working toward the doctorate degree in photogrammetry and remote sensing with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan.

His research interests include LiDAR remote sensing, computer vision, and point cloud processing, such as point cloud semantic segmentation, 3-D reconstruction from multisource remote sensing data, and graph-structured data applications.

Qinjun Qiu received the B.S. degree in computer science and technology and the M.S. degree in computer applicated technology from the China Three Gorges University, Yichang, China, in 2011 and 2014, respectively, and the Ph.D. degree in geographic information engineering from the China University of Geosciences, Wuhan, China, in 2020.

He is currently an Associate Professor with the School of Geography and Information Engineering, China University of Geosciences. His research interests include deep learning, text mining, and knowledge graph.

Zhong Xie received the B.S., M.S., and Ph.D. degrees in cartography and geographic information engineering from the China University of Geosciences, Wuhan, China, in 1990, 1998, and 2002, respectively.

He is currently a Professor with the School of Geography and Information Engineering, China University of Geosciences. His research interests include deep learning, 3-D rebuilding and spatial analysis, and image processing.

Yongyang Xu received the B.S. degree in computer science and technology and the Ph.D. degree in geographic information engineering from the China University of Geosciences, Wuhan, China, in 2014 and 2019, respectively.

He is currently an Associate Professor with the School of Geography and Information Engineering, China University of Geosciences. His research interests include deep learning and vector data rendering and processing.