IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 5, 2019, accepted November 26, 2019, date of publication December 2, 2019,
date of current version December 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2957157

MHNet: Multiscale Hierarchical Network for 3D Point Cloud Semantic Segmentation

XIAOLI LIANG[1] **AND ZHONGLIANG FU**[1,2]
[1]School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China
[2]Collaborative Innovation Center of Geospatial Technology, Wuhan University, Wuhan 430079, China

Corresponding author: Zhongliang Fu (fuzl@whu.edu.cn)

ABSTRACT Point cloud semantic segmentation is a challenging task in 3D understanding due to its disorder, unstructured and nonuniform density. Currently, most methods focus on network design and feature extraction. However, it is difficult to capture the point cloud features of complex objects comprehensively and accurately. In this paper, we propose a multiscale hierarchical network (MHNet) for 3D point cloud semantic segmentation. First, a hierarchical point cloud feature extraction structure is constructed to learn multiscale local region features. Then, these local features are subjected to feature propagation to obtain the features of the entire point set for pointwise label prediction. To take fully advantage of the correlations of propagated information between the different scale coarse layers and the original points, the local features of each scale are characterized by feature propagation to obtain the features of the original point clouds at the corresponding scale. The global features propagated from different scales are integrated to constitute the final features of the input point clouds. The concatenated multiscale hierarchical features, including both local features and global features, can better predict the segmentation probability of each point cloud. Finally, the predicted segmentation results are optimized using the conditional random filed (CRF) with a spatial consistency constraint. The efficiency of MHNet is evaluated on two 3D datasets (S3DIS and ScanNet), and the results show performance comparable or superior to the state-of-the-art on both datasets.

INDEX TERMS Point cloud, multiscale hierarchical network (MHNet), conditional random field (CRF), semantic segmentation.

I. INTRODUCTION

Point cloud semantic segmentation plays a critical role in autonomous driving, robot navigation, augmented reality and 3D reconstruction. Currently, with the development of deep learning technology, semantic segmentation has made great progress, but it also faces many difficulties. The unordered and unstructured properties of 3D point clouds make it difficult to be presented as 2D images. Therefore, it is impossible to directly apply the existing image segmentation framework to the point clouds. Moreover, its large scale and nonuniform density also present numerous challenges in 3D point cloud understanding. Previous solutions mainly transform 3D point clouds into 2D images [1]–[4] and regular voxel grids [5]–[7]. However, converting point clouds to 2D formats results in the loss of information. Voxelization assigns the points in the same voxel with the same semantic label, which tends to discard small details. Due to the sparsity of point clouds, voxelization often leads to heavy calculation and low efficiency.

PointNet [8] was the first work to directly address 3D point clouds. Without transforming to a 3D voxel grid or mesh, the network takes raw points as input and outputs the semantic label of each point. The whole network structure is simple but can effectively realize semantic segmentation, based on which many deep network structures dealing with raw point clouds are derived [9]–[15]. However, only global information is extracted in PointNet, and the relationship between neighboring point clouds is not considered. To overcome this problem, PointNet++ [16] was proposed. The improved network increased the ability to capture local features at different scales by exploiting metric space distance. In recent years, inspired by this algorithm, many networks for 3D point cloud semantic segmentation have emerged [13], [17], [18].

However, there are still some shortcomings in local feature capture of point clouds in PointNet++ [16]. In PointNet++, all of the point clouds are first downsampled randomly

The associate editor coordinating the review of this manuscript and approving it for publication was Weipeng Jing.

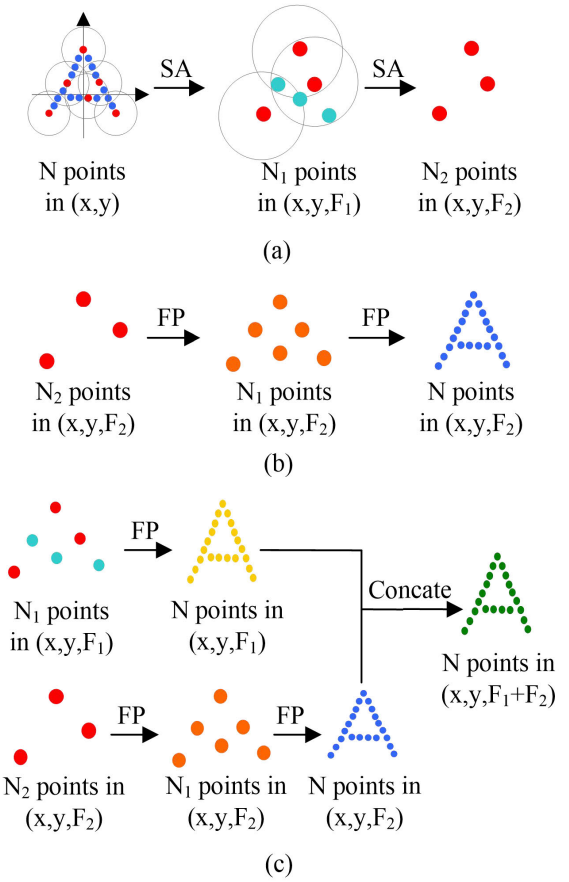


FIGURE 1. Illustration of feature extraction and propagation: (a) hierarchical point cloud feature extraction; (b) the method of original point set feature propagation in PointNet++; and (c) the improved method of original point set feature propagation in MHNet.

through iterative farthest point sampling (FPS) to obtain the centroids of local regions. Then, local region sets are constructed by searching neighborhood points around centroids. Finally, a mini PointNet is used to extract local region features of the point clouds in each local region. After several such operations, the number of points in each local region decreases, while the features extracted from each point gradually increase. This process is named the Set Abstraction (SA) module in PointNet++. For instance, in Fig. 1(a), we illustrate the process of obtaining local features of two scales through two SA operations, in which N is the number of original point clouds, N_1 and N_2 refer to the sampling points at two scales, F_1 and F_2 are corresponding local features, respectively, and each circle represents the scope of the local region searched by each centroid. Point cloud features of different scales obtained by the SA module in PointNet++ need to propagate to the original point features for per-point label prediction. The features of the original points in PointNet++ are obtained by a continuous upsampling layer by layer through several Feature Propagation (FP) modules. As shown in Fig. 1(b), to obtain the original point features, the second scale point cloud features $N_2 \times F_2$ are first upsampled to the features of the first level $N_1 \times F_2$, and then the same operation is performed on the first scale features to obtain the original point features $N \times F_2$. However, it fails to take full advantage of the correlations of propagated features between different scales and the original points. Additionally, local features and global features are not well integrated to describe point cloud information.

To address these problems, we propose a Multiscale Hierarchical Network (MHNet) for 3D point cloud semantic segmentation. In MHNet, the original point feature propagation method is improved to better capture local region features and fully fuse local features and global features of point clouds. A brief description of the improved method in MHNet is shown in Fig. 1(c). The first scale features $N_1 \times F_1$ obtain the original points features $N \times F_1$ through one FP module, and the second scale features $N_2 \times F_2$ obtain the original points features $N \times F_2$ after two FP modules. Then, the global features obtained from these two scales are concatenated to form the multiscale features $N \times (F_1 + F_2)$. The concatenated features obtained by our method not only consider the relationship between different scale features and original point features but also effectively integrate the global features obtained by different scales to express point cloud information more comprehensively.

In this paper, a hierarchical structure composed of a number of SA levels is first built for point set feature learning. Through this process, we obtain sampling points and local features of different levels. For point cloud FP, we adopt the distance based interpolation and across level skip links strategy, which is used in PointNet++ [16], to transfer features from a coarse layer to a dense layer. We propagate the local features generated by each scale in the hierarchical structure to the features of original points through several FP operations. Then, we concatenate the features of each level to obtain multiscale hierarchical features, which include the local and global information of point clouds. Finally, semantic prediction labels of input points are obtained by processing the connection features.

In many works of point cloud semantic segmentation, a Conditional Random Filed (CRF) is used as the post-processing operation to optimize the semantic prediction results [2], [5], [10]. The CRF can label the input samples sequentially by considering the spatial consistency of point clouds. In our method, CRF is used to further optimize the segmentation results of points, which can remove noise points by calculating the probability distribution.

The main contributions of our work are listed below:

- We design a new network MHNet, which can fully incorporate both local and global multiscale hierarchical features for highly accurate point cloud semantic segmentation.
- In our method, we constructed a CRF model with a spatial consistency constraint to obtain the global context for global label optimization to further improve the segmentation result.

The remainder of this paper is organized as follows. We first review the related works in Section II. The architecture details of our proposed methods are described in Section III. Section IV demonstrates the effectiveness of the proposed network in experiments using two datasets. Finally, the conclusion is given in Sections V.

II. RELATED WORKS

Due to the irregularity, disorder, and inconsistent density of point clouds, there are few 3D CNN models in comparison with 2D images. Traditional point cloud semantic segmentation algorithms mainly rely on hand-crafted features [19]–[26]. Currently, motivated by the development of deep learning technology and the large collection of scene datasets, an increasing number of works that adopt end-to-end deep learning algorithms have been proposed to address point clouds. Previous works covert point clouds to other formats, such as 2D multiview format or 3D volumetric grids. In recent years, networks that directly address raw point clouds have been endlessly emerging. Here, we mainly review the research related to our work.

A. 3D VOLUMETRIC GRIDS

The voxelization of point clouds is the first attempt in deep learning. Many works [7], [27]–[32] convert point clouds into regular volumetric grids and then apply 3D convolution networks. The 3D analysis includes 3D classification, object recognition, shape representation, point cloud labeling, shape reconstruction and scene segmentation. However, the sparse data and expensive computation of 3D convolutions constrain this type of approach. Some works have proposed easing the computational intensity. The author of [33] utilized a hybrid grid octree, which can effectively handle higher-resolution 3D voxel grids. Klokov and Lempitsky [34] proposed a Kd-networks method, which is a feedforward bottom-up computational representation. Li *et al*. [35] designed a method to reduce computations by sampling spare points before feeding into the network. In [36], [37], a spare convolution network was proposed for 3D point cloud segmentation. These works focused more on easing the computational intensity, and the loss of information in voxelization was not considered. The accuracy of the voxel-based methods is far less than that of point-level methods.

B. 2D MULTIVIEW FORMAT

Su *et al*. [38] converted 3D point clouds into several 2D images, and then used the 2D CNN for 3D shape recognition. Qi *et al*. [29] designed a 2D CNN for 3D object classification, and compared the performance of 2D multiview CNN with the 3D volumetric CNN. Pang and Neumann [39] designed a multiview CNN for object detection. Guerry *et al*. [31] achieved 3D multiview semantic labeling by directly processing RGB-D snapshots in numerous views. These studies are achieved by projecting 3D data into 2D images, which results in the loss of geometric details information, which is not conducive to target recognition and semantic segmentation. In addition, it requires extra 2D to 3D remapping.

C. POINT CLOUDS

In the seminal work of PointNet [8], the authors utilized the shared multilayer perceptions to extract features of raw point clouds directly, and then aggregated them into global features by a max pooling layer. This work failed to capture the local features of point clouds. To solve this drawback, PointNet++ [16] proposed a hierarchical network to obtain the multiscale local details of point sets. Additionally, inspired by the work of PointNet [8], Engelmann *et al*. [11] incorporated the neighborhood information by using multiscale windows or neighboring cell positions. The following works PointCNN [40], RSNet [41], 3P-RNN [42] and DGCNN [43] further focused on exploring the local context. PointCNN [40] is a generalization of typical CNNs that is capable of leveraging spatial local correlation from data represented in point clouds. RSNet [41] combined the slice pooling, RNN layer and a slice unpooling layer to equip a lightweight local dependency model. 3P-RNN [42] captured local structure information of different densities in a multiscale neighborhood by constructing the efficient pointwise pyramid pooling module. DGCNN [43] presented a novel EdgeConv operation for capturing local geometric features of point clouds while still maintaining permutation invariance. These extensions of PointNet largely focus on capturing local features while neglecting the the geometric relationships among points. Effective point cloud semantic segmentation requires a combination of local and global information. Recently, a series of effective architectures have been proposed that effectively combine the local and global features of the point cloud to improve the segmentation accuracy. For example, PC-CNNI and PC-CNNII [9] integrated local and global features comprehensively to enhance the expression ability of the model, and thus improved the semantic segmentation accuracy. Wang [10] proposed an improved PointNet structure to connect the point cloud features of different dimensions to realize the aggregation of global features and local features. In our proposed method, the local and global features of point clouds are integrated for better semantic labeling.

D. CONDITIONAL RANDOM FIELD

The CRF model has been commonly used as a postprocessing step for semantic segmentation [2]. Recently, with the rapid development of deep learning, it has become possible to embed CRF into neural networks, such as semantic segmentation [5], [10], [44], object segmentation [45], and point cloud classification [46]. The basic CRF structure is limited in building long-range connections within the model and generally results in excessive smoothing of object boundaries. Compared to the basic neighbor connected CRF model, dense CRF [47] can model long-range relationships and has become a popular tool for semantic segmentation [48], [49]. CRF is also extended with high-order potentials to further improve coherency in the label prediction. For example,

Pham *et al.* [50] integrated semantic segmentation into real-time indoor scanning by optimizing the predictions from a 2D neural network with a novel higher-order CRF model. Yang *et al.* [51] utilized a high-order CRF model to optimize 3D grid labels for fast outdoor scene segmentation, in which superpixels were used to enforce smoothness and form robust high-order potential. However, the computational complexity of high-order CRF is relatively large. The CRF model we proposed in this work is a dense CRF that can build long-range connections among the point clouds in the model to ensure clear object boundaries. Our idea is to obtain a coherent, high-quality segmentation result that will pave the way for our future contour extraction work.

III. METHODOLOGY

We design an architecture for 3D point cloud semantic segmentation, which directly uses raw point clouds as input data. In our method, a multiscale hierarchical network is constructed to fully capture features of different scales for semantic label prediction. Through the established MHNet, we obtain multiscale hierarchical features with local features and global features, which can better predict the label probability of a point cloud. To eliminate the influence of noise points in the segmentation process, the input points are regarded as a nondirectional probability graph, based on which a CRF model is constituted. The predicted label probability of the point cloud obtained from MHNet is optimized globally using the CRF model with a spatial consistency constraint. The overall flowchart of our proposed method is shown in Fig. 2.

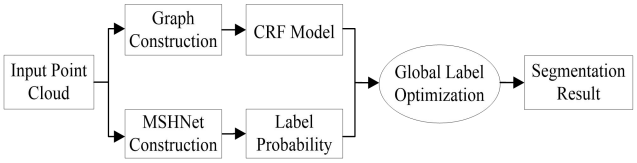


FIGURE 2. The flowchart of our proposed method.

A. HIERARCHICAL POINT CLOUD FEATURE EXTRACTION

To aggregate the features of the whole point set, we build a hierarchical structure that can gradually abstract increasingly local regions. The hierarchical structure is composed of a number of feature encoding modules, which is similar to the SA module introduced in PointNet++ [16]. The SA module is a downsampling stage. At each SA level, the input points $N_l \times (d + C_l)$ (that is N_l points with d dimensional coordinates and C_l dimensional features) are processed through the sampling layer, grouping layer and PointNet layer to obtain the sampled points and their local region features. In the sampling layer, N_l centroids are fund by the FPS. Then, k points in the neighborhood of each centroid are queried to form the local region. In this process, a k Nearest Neighbor (kNN) search or radius based ball query is used to obtain the groups of point sets $N_{l-1} \times k \times (d + C_{l-1})$. The ball query searches all points within a radius to the query point, and the kNN search finds a fixed number of neighboring points. Compared with kNN, the local neighborhood of the radius based ball query provides a fixed region scale. Therefore, local region features are more generalized in space, which is helpful for point cloud semantic labeling. Finally, local region features of centroids $N_{l-1} \times (d + C_{l-1})$ are fed through the PointNet layer. The obtained downsampled points in each level with d dimensional coordinates and C_{l-1} dimensional local region features.

In our network, four consecutive downsampling operations (namely, SA module) decrease the size of the point set to N_1, N_2, N_3, N_4. The corresponding local region features of each SA level are C_1, C_2, C_3, and C_4, respectively. Through the hierarchical structure, we obtain different scales of sampled points and local region features. The architecture of the hierarchical structure can be seen in Fig. 3, and the details are listed in TABLE 1.

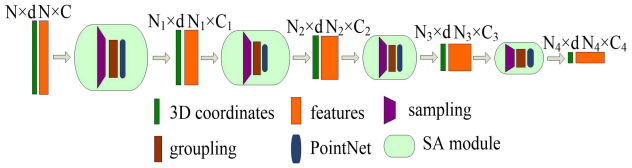


FIGURE 3. Hierarchical point cloud feature extraction.

B. FEATURE PROPAGATION FOR PER-POINT LABEL PREDICTION

In the semantic segmentation task, we need to predict point-wise labels. A method to propagate features from a subsampled point to a denser point is needed to obtain features for the original input points. PointNet++ [16] adopts a hierarchical propagation strategy to propagate local features to the original point set through several FP modules. The purpose of the FP module is to propagate features from sparse points to dense points. The inverse distance weighted average based on k-nearest neighbor linear interpolation is used to upsample the point clouds, as seen in Eq. (1). It propagates points from size N_l to N_{l-1}, and the feature dimension is not changed in this stage. The interpolated features on N_{l-1} points are concatenated with skip linked point features from the SA level, and then passed through a PointNet unit. This process is repeated until the features of the original points are obtained.

$$f^{(j)}(x) = \frac{\sum_{i=1}^{k} w_i(x) f_i^j}{\sum_{i=1}^{k} w_i(x)}, \; w_i(x) = \frac{1}{d(x, x_i)^2}, \quad j = 1, 2, \ldots, C \tag{1}$$

As mentioned above, in PointNet++ [16], the features of original points are obtained through the level by level propagation, which dose not fully consider the correlations of propagated features from the different levels and the features of original points. Point cloud semantic segmentation requires a combination of local and global knowledge. We can

TABLE 1. Architecture details of MHNet.

Process	Module	Layer	Input						Output
Hierarchical feature extraction	SA	1	(N, 3)	None	1024	0.1	31	[32, 32, 64]	(1024, 3), (1024, 64)
		2	(1024, 3)	(1024, 64)	256	0.2	31	[64, 64, 128]	(256, 3), (256, 128)
		3	(256, 3)	(256, 128)	64	0.4	31	[128, 128, 256]	(64, 3), (64, 256)
		4	(64, 3)	(64, 256)	16	0.8	31	[256, 256, 512]	(16, 3), (16, 512)
Feature propagation	FP	1	(N, 3)	(1024, 3)	None	(1024, 64)		[64, 64]	(N, 64)
			(1024, 3)	(256, 3)	(1024, 64)	(256, 128)		[128, 128]	(1024, 128)
		2	(N, 3)	(1024, 3)	None	(1024, 128)		[128, 128]	(N, 128)
			(256, 3)	(64, 3)	(256, 128)	(64, 256)		[256, 256]	(256, 256)
		3	(1024, 3)	(256, 3)	(1024, 64)	(256, 256)		[256, 256]	(1024, 256)
			(N, 3)	(1024, 3)	None	(1024, 256)		[256, 256]	(N, 256)
			(64, 3)	(16, 3)	(64, 256)	(16, 512)		[512, 512]	(64, 512)
			(256, 3)	(64, 3)	(256, 128)	(64, 512)		[512, 512]	(256, 512)
		4	(1024, 3)	(256, 3)	(1024, 64)	(256, 512)		[512, 512]	(1024, 512)
			(N, 3)	(1024, 3)	None	(1024, 512)		[512, 512]	(N, 512)
	Concat	1	(N, 64), (N, 128), (N, 256), (N, 512)						(N, 960)
Per-point label prediction	Convolution	1	(N, 960)						(N, 512)
		2	(N, 512)						(N, 256)
		3	(N, 256)						(N, 128)
	DP	1	(N, 128)						(N, 128)
	Convolution	1	(N, 128)						(N, num_class)

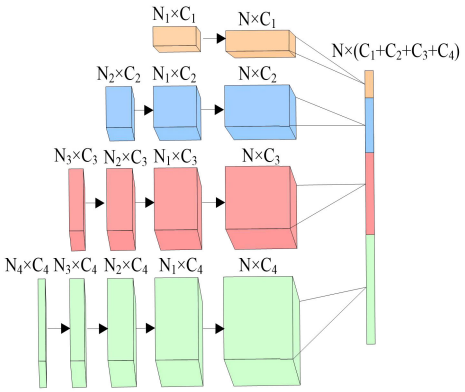


FIGURE 4. Point cloud feature propagation.

achieve this by a simple yet highly effective method. Unlike PointNet++ [16], we conduct the FP operation several times on local region features of each level in the hierarchical structure until we have the propagated features to the original set of points. As shown in Fig. 4, local features of the four scales obtained through the hierarchical structure, $N_1 \times C_1$, $N_2 \times C_2$, $N_3 \times C_3$, and $N_4 \times C_4$, are propagated level by level through the FP module to obtain the corresponding scale original set features $N \times C_1$, $N \times C_2$, $N \times C_3$, and $N \times C_4$. After capturing the original point cloud features from different levels, we concatenate the features propagated from different SA levels to obtain the multiscale hierarchical features $N \times (C_1 + C_2 + C_3 + C_4)$.

The FP module consists of three main parts: interpolation, skip links and PointNet unit. The details of the FP process are shown in Fig. 5. The point features $N_l \times C_l$ output from SA level l are first interpolated to obtain the features $N_{l-1} \times C_l$ in the denser layer. The interpolated features on N_{l-1} points are concatenated with skip linked point features $N_{l-1} \times C_{l-1}$ from the SA level $l-1$. Then, the concatenated features $N_{l-1} \times (C_l + C_{l-1})$ are conducted on a PointNet unit to obtain the features $N_{l-1} \times C_{l-1}$ in a FP module. At each level, the FP module is repeated to gradually transfer features from a coarse level to a finer level until we obtain the features of the original points $N \times C_{l-1}$. By combining the original point cloud features obtained at different scales, we obtain the multiscale point cloud features. With this modification, our network can predict per-point quantities that rely on both local and global semantics.

To collect key features from the integrated features for efficient semantic prediction of the point set, we replaced the fully connected layer in PointNet++ with three convolution layers. The concatenated features are input into the last three convolution layers, which are followed by a dropout (DP) layer with drop ration 0.5. Finally, through the last score prediction layer convolution for per-point label prediction. The overall architecture of MHNet is shown in Fig. 5. The details of MHNet are shown in TABLE 1.

To avoid the disappearance of the gradient and shorten the training time, in our model, batch normalization (BN) and ReLU processing are performed in each input layer except for the final score prediction layer, the specific principle is shown in Fig. 6. Let $B = \{x_{1,2,\ldots,m}\}$ be the value of a batch, the definitions of mean value μ_B, variance σ_B^2, standardization $\hat{x}_i$ and linear transformation of all point clouds in each batch are shown in Eq. (2)-(5). When the activation input of each neuron forms a normal distribution, that is, with a mean value of 0 and variance of 1, the expression ability of the network will decline, and then the usage of Eq. (5) can avoid this situation. The two newly added parameters γ and β in Eq. (5) are obtained through training, and they are used to invert the input value of the transformed activation, which can effectively enhance the network expression ability.

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{2}$$

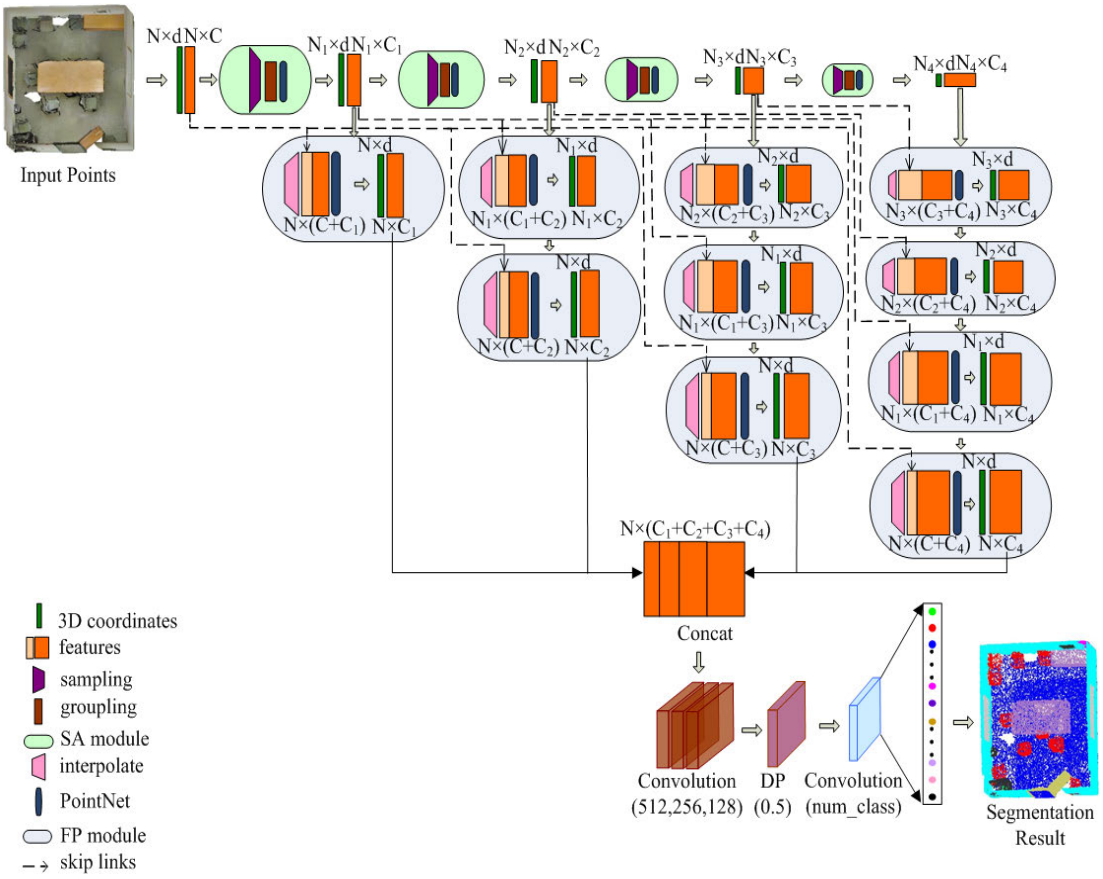


FIGURE 5. Illustration of our MHNet architecture.

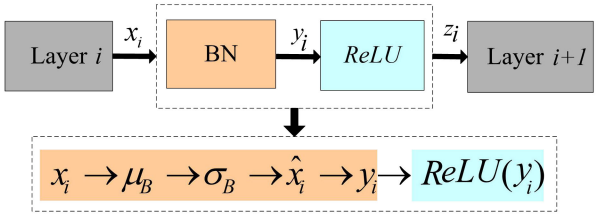


FIGURE 6. BN and ReLU processing in each input layer.

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B^2) \tag{3}$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \tag{4}$$

$$y_i = \gamma\hat{x}_i + \beta = BN_{\gamma,\beta}(x_i) \tag{5}$$

C. CRF OPTIMIZATION SEMANTIC LABEL

CRF is a nondirectional probability graph model, which is usually used to label and analyze sequential data. Each point must be assigned a unique label in point cloud semantic segmentation to indicate its class, so it also belongs to the category of labeling sequential data. To optimize the point cloud segmentation results, we construct a CRF model with space consistency to optimize the category label of each point in the point clouds.

The segmentation results of our network are inevitably influenced by noise, which are isolated points with different categories from the surrounding points. The CRF can remove noise points by calculating the probability distribution. We construct a graph $G = (V, E)$ based on the point clouds, in which each node V corresponds to each point in the point clouds, and the edges E are added between the point and its k-nearest points in the scenario of point cloud.

Let $X = \{X_1, X_2, \ldots, X_N\}$ be a set of random variables corresponding to the 3D points $i \in \{1, 2, \ldots, N\}$.Each random variable X_i takes a label from $L = \{l_1, l_2, \ldots, l_k\}$ when considering k different classes. We construct a CRF model (P, X) based on the graph $G = (V, E)$ of the point clouds, where $P$is the global observation of $G = (V, E)$. Based on the CRF, the probability distribution for a possible labeling $X \in L^N$ given a point cloud p is defined by:

$$p(X = l|P) = \frac{1}{Z(p)}\exp(-E(l|P)) \tag{6}$$

where $E(l|P)$ is the Gibbs energy defined on the CRF graph and $Z(P)$ is the normalized index. $P = \{\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_N\}$ is the predicted class probability of the point clouds, which is

obtained from our MHNet. The Gibbs energy of label l is defined on the unary and pairwise cliques in the graph given by:

$$E(l|P) = \sum_{i \in V} \phi(\hat{p}_i, l_i) + \sum_{(i,j) \in E} \psi(l_i, l_j) \tag{7}$$

where $\phi(\hat{p}_i, l_i)$ is the unary potential, which is the predicted class probability of the point cloud obtained from our MHNet. $\psi(l_i, l_j)$ is the pairwise potentials describing the relationship between points, which encourages assigning the same label to adjacent points. In our work, this distance is defined by calculating the Euclidean distance between two points. The pairwise energies make up for the shortcomings of MHNet segmentation, which can be calculated by the following equation:

$$\psi(l_i, l_j) = \mu(l_i, l_j) \sum_{m=1}^{M} w^m K_G^m(f_i, f_j) \tag{8}$$

where each K_G^m for $m = 1, 2, \ldots, M$is a Gaussian kernel used to measure the similarity of feature vectors of points i and j. The feature vector of points i and j, denoted by f_i and f_j, is represented by point features, such as coordinate location and RGB values [47]. The function $\mu(l_i, l_j)$ represents a compatibility measure between different pairs of labels, w^m represents the linear combination weights.

The optimal point cloud segmentation result can be obtained by minimizing the energy function $E(l|P)$ with the mean-field iteration algorithm [47]. The output after the CRF energy minimization provides us predictions for each 3D point that takes smoothness and consistency into account.

IV. EXPERIMENTS AND ANALYSIS

Two commonly used datasets and three evaluation criteria are chosen to verify the effectiveness of MHNet in our experiments. We compared our segmentation results with state-of-the-art methods. The experimental results of the two datasets are shown in Section IV. B and Section IV. C. We also validated the effects of different architecture choices and testing schemes in ablation experiments. Finally, we analyzed the inference speed and GPU memory consumption in our method.

A. EXPERIMENTAL SETTINGS

1) DATASETS

We benchmark our network on two commonly used large-scale realistic 3D segmentation datasets: the Standford Large-Scale 3D Indoor Spaces (S3DIS) dataset [52] and the ScanNet dataset [27]. The S3DIS dataset contains six large-scale indoor areas of RGB-D point clouds from different buildings, which in total includes 272 rooms. Each point is annotated with semantic labels from 13 categories. The ScanNet dataset is also used to evaluate our method. This dataset contains 1,513 scanned 3D indoor scenes and 21 categories. Our method follows the experimental settings in ScanNet [27]: 1,201 scenes for training and 312 scenes for testing.

TABLE 2. Segmentation result comparisons on the S3DIS dataset (6-fold cross validation).

Method	mAcc (%)	mIoU (%)	OA (%)
PoineNet [8]	66.20	47.60	78.50
MS+CU [11]	59.70	47.80	79.20
SEGCloud^C [5]	57.35	48.92	80.80
G+RCU [11]	66.40	49.70	81.10
SGPN [13]	-	50.37	80.78
ASIS [53]	62.30	51.10	81.70
PC-CNNI [9]	-	52.53	81.95
PointNet++ [16]	67.05	54.49	81.03
SPG [12]	64.40	54.10	82.90
DGCNN [43]	-	56.10	84.10
RSNet [41]	66.45	56.47	-
[54]	67.77	58.27	83.95
MHNet	69.37	58.49	84.92
MHNet^C	**70.85**	**60.89**	**85.64**

2) EVALUATION METRICS

In the validation of S3DIS dataset experiments, k-fold cross validation strategy was used for train and test, and we also used area 6 validation to evaluate our model, as has been used in [9], [10]. Three widely used metrics, mean per-class accuracy (mAcc), mean intersection over union over all classes (mIoU) and overall accuracy (OA), are used to measure the segmentation performance. Per-class accuracy and mAcc are defined as:

$$acc_i = \frac{TP_i}{T_i}, \quad mAcc = \frac{1}{N} \sum_{i=1}^{N} acc_i \tag{9}$$

where TP_i is the number of true positives, T_i is the number of ground truth positive points, P_i is the number of predicted positives, N is the number of classes. We defined per-class IoU and mIoU as:

$$IoU_i = \frac{TP_i}{(T_i + P_i - TP_i)}, \quad mIoU = \frac{1}{N} \sum_{i=1}^{N} IoU_i \tag{10}$$

The overall accuracy of semantic segmentation is defined as:

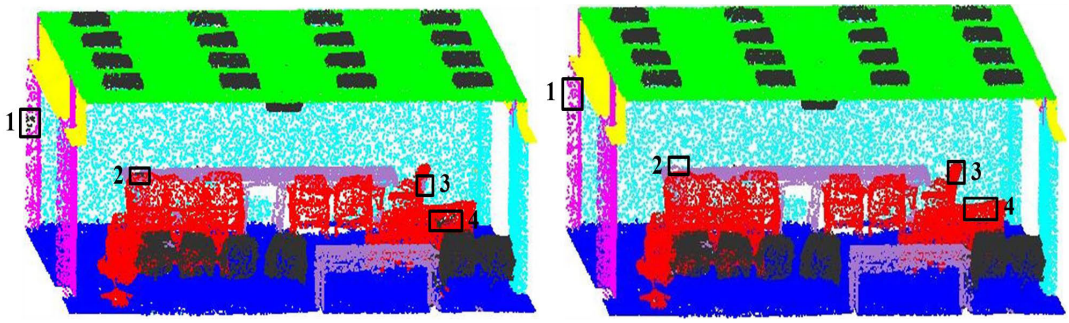
$$OA = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} P_i} \tag{11}$$

3) EXPERIMENTAL DETAILS

In the experiments of the S3DIS dataset, we use the method in PointNet [8] to process the training dataset. For the S3DIS dataset, each input point is represented by a 9-dim vector, namely, XYZ, RGB and normalized coordinate information X'Y'Z'. Since in the PointNet++ [16] ScanNet dataset experiment, only XYZ information of point cloud was used, to make a fair comparison, each point is represented by a 3-dim vector (XYZ) in our ScanNet dataset experiments. In our experiments, we do not use any data augmentations. We set the input data size N to 4,096 and 8,192 for the S3DIS

TABLE 3. IoU for all categories on the S3DIS dataset.

Method	PointNet	MS+CU	SEGCloud	G+RCU	ASIS	PC-CNNI	PointNet	SPG	RSNet	[54]	MHNet	MHNet^C
Ceiling	88.00	88.60	90.06	90.30	91.30	90.58	76.46	92.20	**92.48**	92.10	88.64	88.42
Floor	88.70	95.80	**96.05**	92.10	89.70	90.11	77.15	95.00	92.83	90.40	92.89	93.05
Wall	69.30	67.30	69.86	67.90	69.80	74.17	68.94	71.9	**78.56**	78.50	76.85	78.09
Beam	42.40	36.90	0.00	44.70	**45.80**	31.34	28.88	33.50	32.75	37.80	28.72	30.49
Column	23.10	24.90	18.37	24.20	27.00	29.02	29.69	15.00	34.37	**35.70**	32.63	34.78
Window	47.50	48.60	38.35	**52.30**	51.90	46.64	48.87	46.50	51.62	51.20	49.16	51.48
Door	51.60	52.30	23.12	51.20	55.10	61.92	61.27	60.90	**68.11**	65.40	61.73	62.57
Table	54.10	51.90	**70.40**	58.10	61.00	56.64	60.06	69.40	60.13	64.00	64.17	66.55
Chair	42.00	45.10	**75.89**	47.40	49.30	54.94	67.04	65.00	59.72	61.60	67.18	69.18
Sofa	9.60	10.60	40.88	6.90	9.10	16.62	37.18	38.20	**50.22**	25.60	39.71	48.18
Bookcase	38.20	36.80	**58.42**	39.00	40.20	45.38	54.11	56.80	16.42	51.60	52.68	57.21
Board	29.40	24.70	12.96	30.00	33.50	34.87	47.02	6.86	44.85	49.90	50.41	**53.41**
Clutter	35.20	37.50	41.60	41.90	40.70	46.81	51.68	51.30	52.03	53.70	55.60	**58.21**



(a) (b)

FIGURE 7. The benefits of CRF in the segmentation results on the S3DIS dataset: (a) the segmentation result of base MHNet without CRF; and (b) the segmentation result with CRF global optimization.

dataset and ScanNet dataset, respectively. The network is trained for 50 epochs and 201 epochs, respectively, with batch size 8, learning rate 0.001, and momentum 0.9. The learning rate decay is set to 0.5 in two datasets. The Adam solver is adopted to optimize the network on a single GPU.

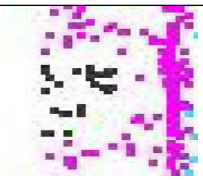
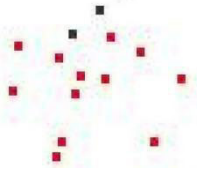
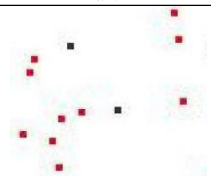
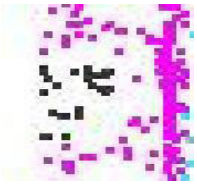
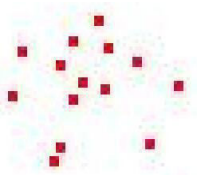
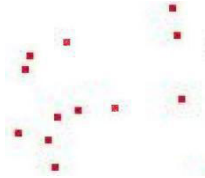
B. SEGMENTATION ON THE S3DIS DATASET

The segmentation results on the S3DIS dataset are shown in TABLE 2, in which the superscript C denotes using the CRF optimization. TABLE 2 shows that our method achieves the best segmentation results on the S3DIS dataset. In particular, even without CRF, MHNet improves PointNet by 3.17% in mAcc, 10.89% in mIoU, and 6.42% in OA. Compared with PointNet++ [16] (single scale grouping, SSG), our base method (without CRF) improves the mAcc, mIoU and OA by 2.32%, 4.00% and 3.89%, respectively. Per-class IoUs are shown in TABLE 3, in which the best IoUs of classes are shown in bold. Compared with all methods, our method is not the best in most of the categories in this dataset, but it has a strong adaptability to the segmentation of all categories. Compared with any method, our method achieves superior results in most categories. The semantic segmentation effect of each class achieves almost the level of the most advanced methods, with mAcc of 69.37% and mIoU of 58.49%.

From the above analysis, we know that our base MHNet achieves state-of-the-art performance. In this section, we analyze the influence of CRF on the segmentation results. The CRF offers a relative improvement of 1.48% mAcc, 2.40% mIOU, and 0.72% OA for this dataset. An example of an openspace in area 6 shows the benefits of the CRF in segmentation results. Fig. 7(a) is the MHNet segmentation result which contains some noise on the chair and column. The combination with the CRF in the MHNet can remove the noise and provide a cleaner segmentation of the point cloud, as shown in the Fig. 7(b). To show the results before and after CRF optimization more clearly, we select four most obvious parts, as shown in the black bounding boxes in Fig. 7. Then, we select some points from the four parts and enlarge them to illustrate the influence of CRF. TABLE 4 clearly shows the comparison of the change in noise points in the point clouds before and after the use of CRF in these four parts. After the CRF global optimization, many noise points are correctly reclassified and fused with the surrounding points.

The comparisons between PointNet++ and our method are visualized in Fig. 8, in which the ceiling, part of the walls and beams are hidden for clarity. From left to right are the input scenes, the results produced by the PointNet++, MHNet, MHNet with CRF (MHNet^C), and the ground truth. These visualizations show that our network achieves a better segmentation effect and can achieve more accurate results in classes, such as door, table, board, column and bookcase. The CRF global optimization also shows its advantages in point

TABLE 4. Segmentation result comparisons between with CRF and without CRF.

Method	Part 1	Part 2	Part 3	Part 4
Without CRF				
With CRF				

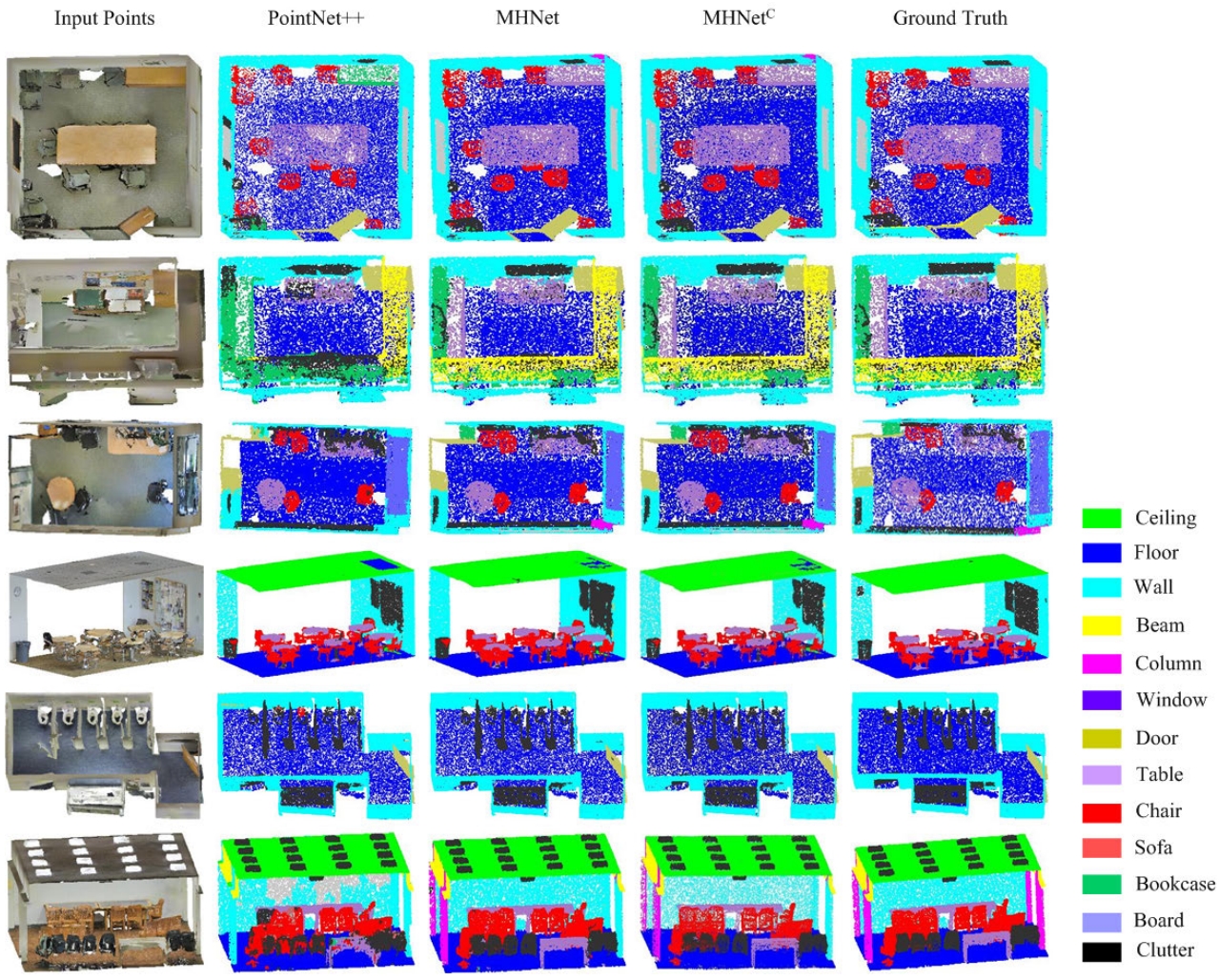


FIGURE 8. Visualization of results on the S3DIS dataset.

cloud noise removal, which can be seen in categories such as ceiling, door, wall, chair, etc.

The segmentation results on the area 6 fold are shown in TABLE 5, and the best results are shown in bold. Compared with other methods, our method performs better in this test area, with 83.77% mean per-class accuracy, mIoU of 73.45%, and overall accuracy of 89.92%. The semantic segmentation performance of our model is better than that of PointNet [8]

TABLE 5. Segmentation result comparisons on the S3DIS dataset (Area 6 validation).

Method	mAcc (%)	mIoU(%)	OA(%)
PointNet [8]	74.98	64.70	86.52
PointNet++ [16]	82.78	72.19	89.41
PC-CNNI [9]	72.43	61.32	84.81
PC-CNNII [9]	75.02	64.09	85.81
Wang [10]	-	63.96	86.42
MHNet	**83.77**	**73.45**	**89.92**

TABLE 6. Neighborhood query: kNN vs. ball query.

Method	mAcc (%)	mIoU(%)	OA(%)
kNN	73.25	61.80	84.09
Ball query	**77.60**	**68.27**	**87.23**

TABLE 7. Segmentation accuracy of different k in the kNN query.

k	mAcc (%)	mIoU(%)	OA(%)
16	69.32	57.94	81.97
32	**73.25**	**61.80**	**84.09**
64	71.29	60.47	83.77

TABLE 8. The segmentation results of different input information on the S3DIS dataset.

Input	mAcc(%)	mIoU(%)	OA(%)
XYZ	64.06	52.91	82.28
XYZ-RGB	68.81	58.26	**85.30**
XYZ-RGB-X'Y'Z'	**69.37**	**58.49**	84.92

TABLE 9. Segmentation result comparisons on the ScanNet dataset.

Method	mAcc (%)	mIoU(%)	OA(%)
G+RCU [11]	-	-	63.40
PointNet [8]	19.90	14.69	73.90
3DCNN [27]	-	-	73.00
PointNet++ [16]	43.77	34.26	74.30
[54]	25.39	-	75.53
3P-RNN [42]	-	-	76.50
RSNet [41]	48.37	39.35	-
PointConv [18]	-	55.60	-
MHNet	73.07	58.19	82.01
MHNet^C	**73.86**	**59.06**	**82.91**

and PointNet++ [16], and the mean per-class accuracy, mIoU, and overall accuracy are improved by 0.99%∼8.79%, 1.26%∼8.75%, and 0.51%∼3.40%.

In our method, we compare two options to select a local neighborhood on the Area1∼Area5 training set, including a radius based ball query and a *kNN* based neighborhood search. The segmentation result comparisons are shown in TABLE 6. In the *kNN* based neighborhood search, we also consider different k values to choose the best result, as seen in TABLE 7. We can see that even in the best result in the *kNN* query (when k is 32), the ball query method still performs slightly well. Thus, we use a radius based ball query in our method. To reduce the computational burden, the input point is represented by 3-dim coordinate information in this comparison experiment.

To verify the influence of different input vectors of the point cloud on the segmentation results, we conduct several groups of comparative experiments on the base MHNet. In our experiments, we use 3-dim XYZ, 6-dim XYZ-RGB, and 9-dim XYZ-RGB-X'Y'Z' information to represent the input point cloud, and analyze the semantic segmentation results of different dimension input vectors. The results are shown in TABLE 8. We can see that the increased RGB information of the point cloud improves the segmentation accuracy by 4.75%, 5.35%, 3.02%, respectively. The normalized coordinate information X'Y'Z' can also slightly contribute to increasing the segmentation effect of our method.

C. SEGMENTATION ON THE SCANNET DATASET

The performance of MHNet on the ScanNet dataset is reported in TABLE 9. Here, we also present the results of other state-of-the-art methods, including G+RCU [11], PointNet [8], 3DCNN [27], PointNet++ (SSG) [16], Engelmann [54], 3P-RNN [42], RSNet [41] and PointConv [18]. TABLE 9 shows that our base MHNet achieves the best segmentation results on the ScanNet dataset with mAcc of 73.07%, mIoU of 58.19%, and OA of 82.01%. Compared with PointNet++ [16], our method improves the mAcc, mIoU and OA by 29.30%, 23.93% and 7.71%, respectively. The CRF global optimization slightly improves the mAcc by 0.79%, mIoU by 0.87%, and OA by 0.90%. IoUs for all categories of ScanNet dataset are shown in TABLE 10. The results show that our method outperforms all other methods in most of the categories, and show a good advantage even in the categories with poor segmentation results in other methods, such as desk, bathtub, door, refrigerator, and cabinet. The comparisons between PointNet++ and our method are visualized in Fig. 9. Our method performs better segmentation results in the categories of door, table, refrigerator, desk than other methods, which is consistent with the results in TABLE 10.

Furthermore, considering that only XYZ information was used in the ScanNet dataset experiments, the effectiveness of RGB information is not verified. In this part, we compare the segmentation results of two groups of experiments, in which the input information is XYZ and XYZ-RGB. The results in TABLE 11 show that when the input information is XYZ-RGB, the segmentation accuracy is 60.28% for mAcc, 43.63% for mIoU, and 74.68% for OA. Compared with the XYZ experiment results, we can see that the RGB information does not improve the segmentation results on the ScanNet dataset. Thus, in our ScanNet experiment, we choose the coordinate information to represent the input points.

D. ABLATION EXPERIMENTS

We perform more ablation experiments for our method using the S3DIS dataset. In this section, we validate the effects of various architecture choices and testing schemes. In our

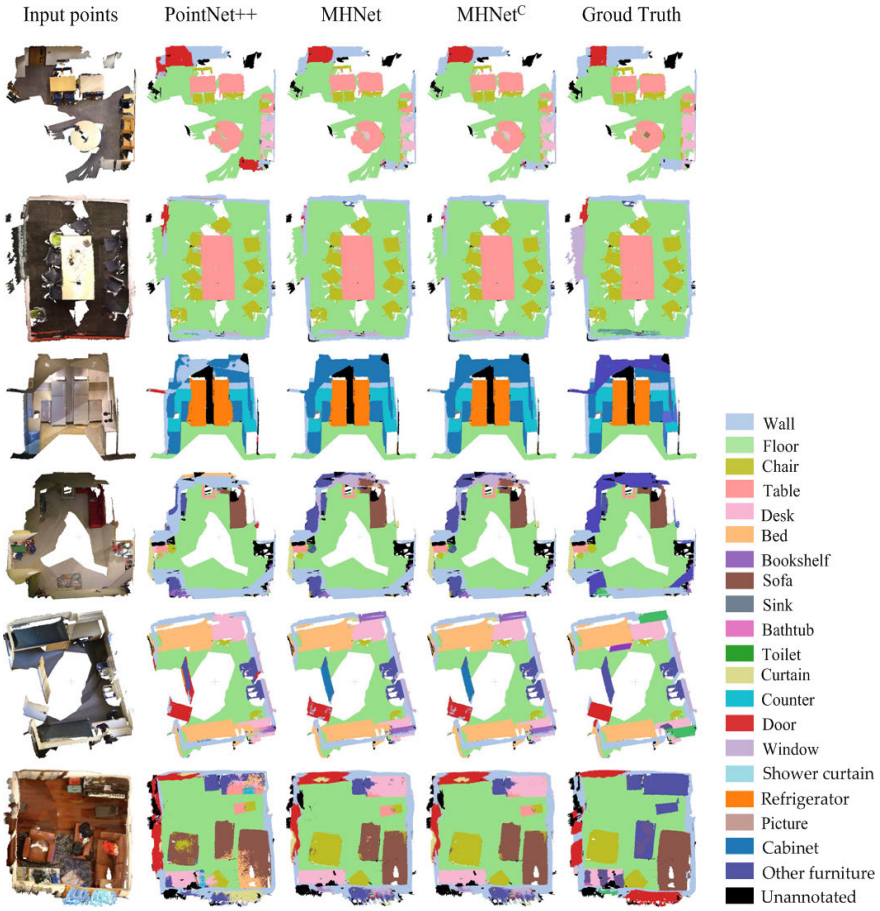


FIGURE 9. Visualization of results on the ScanNet dataset.

TABLE 10. IoU for all categories of the ScanNet dataset.

Method	PointNet [8]	PointNet++ [16]	RSNet [41]	MHNet	MHNet^C
Wall	69.44	77.48	**79.23**	72.90	73.70
Floor	88.59	92.50	94.10	94.12	**94.79**
Chair	35.93	64.55	**64.99**	44.09	49.61
Table	32.78	46.60	51.04	**70.77**	69.77
Desk	2.63	12.69	34.53	82.44	**82.98**
Bed	17.96	51.32	55.95	68.99	**73.08**
Book-shelf	3.18	52.93	53.02	61.58	**64.73**
Sofa	32.97	52.27	**55.41**	34.33	35.86
Sink	0.00	30.23	34.84	**47.71**	45.14
Bathtub	0.17	42.72	49.38	69.76	**71.21**
Toilet	0.00	31.37	**54.16**	13.67	11.02
Curtain	0.00	32.97	6.78	56.27	**57.17**
Counter	0.00	20.04	22.72	48.97	**52.85**
Door	5.09	2.02	3.00	**58.03**	53.87
Window	0.00	3.56	8.75	31.16	**37.50**
Shower curtain	0.00	27.43	29.92	**57.33**	51.92
Refrigerator	0.00	18.51	37.90	82.02	**82.45**
Picture	0.00	0.00	0.95	**56.49**	55.97
Cabinet	4.99	23.81	31.92	77.87	**80.13**
Other furniture	0.13	2.20	18.98	35.32	**37.35**

ablation experiments, several key parameters are considered: 1) the size of the striding block; 2) the stride size of the sliding window; 3) the impact of the input order; and 4) the effect of data augmentation.

TABLE 11. The segmentation results of different input information on the ScanNet dataset.

Input	mAcc (%)	mIoU (%)	OA (%)
XYZ	**73.07**	**58.19**	**82.01**
XYZ-RGB	60.28	43.63	74.68

1) SIZE OF THE SLIDING BLOCK

We report the results of three different block sizes, 1 m, 2 m, and 3 m in TABLE 12. The segmentation accuracy tends to decrease with increasing block size. This is because a larger block size produces more slices, which makes it hard to build the model. Among various settings, the optimal block size for the S3DIS dataset is 1 m. Thus, to prepare training data, we first split points by room and then sample rooms into blocks with an area of 1 m by 1 m.

2) STRIDE SIZE OF THE SLIDING WINDOW

In the segmentation experiment of the S3DIS dataset, we sample 1.5 m$\times$1.5 m$\times$3 m cubes from the initial scene to generate the training data. We train our model with 4,096 points randomly sampled from a cube and evaluate the model exhaustively to choose all points in the cube in a

TABLE 12. The accuracy of different block sizes on the S3DIS dataset.

Block Size (m)	mAcc (%)	mIoU (%)	OA (%)
1.0	**69.37**	**58.49**	**84.92**
2.0	65.07	54.29	84.05
3.0	59.18	47.61	79.57

TABLE 13. The accuracy of different stride sizes of sliding windows on the S3DIS dataset.

Stride Size (m)	mAcc (%)	mIoU (%)	OA (%)
0.2	70.86	59.79	**87.47**
0.5	**70.92**	**60.15**	86.78
1.0	69.37	58.49	84.92

TABLE 14. The segmentation results of different input orders on the S3DIS dataset.

Method	mAcc (%)	mIoU (%)	OA (%)
Unsorted	69.37	58.49	84.92
Sorted-XYZ	**69.67**	**58.70**	**85.33**

sliding window fashion through the XY-plane with different stride sizes. Here, we set the sliding stride into three values: 0.2 m, 0.5 m, and 1.0 m. The first two options require splitting the scenes into overlapping cubes during testing, which is also used in PointNet++ [16]. Like PointNet [8], the last option produces nonoverlapping cubes. The experimental results are reported in TABLE 13. Experimental results show that using overlapped division can slightly increase the performance with 1.49%∼1.55% in mean per-class accuracy, 1.30%∼1.66% in mean IoU, and 1.86%∼2.55% in overall accuracy on the S3DIS dataset. However, testing using overlapped division requires more computations as there are more cubes to process. Thus, we set the stride size to 1.0 m; namely, there is nonoverlap sliding in our MHNet.

3) THE IMPACT OF THE INPUT ORDER

To verify the impact of the order of input point sets on the segmentation performance of the network model, we compare the effect of semantic segmentation under two conditions: unsorted, and sorted by 3D coordinate (namely, Sorted-XYZ). In the comparison experiments, we show the segmentation results of two sorting modes in TABLE 14. The results show that the order of input point sets can affect the segmentation performance of the MHNet. Compared with the unsorted input points, the XYZ sorted method is slightly improved the segmentation result, in which the mean per-class accuracy and mIoU are increased by 0.3% and 0.21%, respectively, and the overall accuracy is improved by 0.41%. Since the model trained by the XYZ sorted point cloud has no obvious effect on improving the segmentation effect, the unsorted point cloud is used in our network.

TABLE 15. The effect of data augmentation on the S3DIS dataset.

Method	mAcc (%)	mIoU (%)	OA (%)
PointNet [8]	65.47	54.54	80.91
PointNet^{A1} [8]	74.98	64.70	86.52
PointNet++ [16]	82.67	71.66	88.87
PointNet++^{A2} [16]	82.78	72.19	89.41
MHNet	83.77	73.45	89.92
MHNet^{A1}	78.72	68.30	86.77
MHNet^{A2}	**84.27**	**73.61**	**90.81**

4) THE EFFECT OF DATA AUGMENTATION

In PointNet [8], different data augmentation methods, including random rotation point clouds and jittering of XYZ coordinates, are used to augment point clouds. PointNet++ [16] adopt rotation along the Z-axis to augment data. We want to determine the role of data augmentation methods on the performance of our MHNet architecture. We, therefore, train the MHNet on Area1∼Area5 training set with two types of data augmentation methods used in PointNet [8] and PointNet++ [16] and report their performances in TABLE 15. The superscript denotes the data augmentation methods in PointNet (A1) and PointNet++ (A2) are used in experiments, respectively. We observe that the data augmentation method used in PointNet [8] does play a significant role in the final performance of PointNet. However, it does not improve the segmentation accuracy of our MHNet. The data augmentation method used in PointNet++ [16] can slightly improve the segmentation accuracy both in PointNet++ and in our MHNet. However, even without any geometric data augmentation, our base MHNet outperforms the PointNet++ [16] by 1.26 mIOU points.

E. COMPUTATION ANALYSIS

In this section, we demonstrate the effectiveness of our base MHNet in terms of inference speed and GPU memory consumption in a batch of points with a size of 4,096×9. We report the computation time measured on a single GTX 1080Ti GPU in TensorFlow. The computation time and memory consumption are estimated on the area 6 test data of the S3DIS dataset. We take the first conference room as an example, which contains 18 batches. The first batch is neglected since there is some preparation for the GPU. Then, the mean inference time of the remaining 17 batches is chosen as the final judgment basis. The speed and memory measurements are reported in TABLE 16.

The results show that the memory consumption of our method is similar to the single-scale version of PointNet++ [16] and even lower than PointNet [8]. Moreover, the inference speed of our MHNet is approximately equal to PointNet++ [16]. These prove the effectiveness of our model, which does not consume too much memory or time while the complexity of the model increases, and the segmentation accuracy is improved.

TABLE 16. Computation analysis between PointNet, PointNet++, and MHNet.

Method	PointNet [8]	PointNet++ [16]	MHNet
Inference time	9 ms	42 ms	44 ms
Memory	805 MB	753 MB	756 MB

V. CONCLUSION

In this paper, we propose a deep learning framework MHNet for 3D point cloud semantic segmentation. The framework consists of two important components. In the first part, a MHNet structure is constructed for feature learning to predict point label probability, in which the set abstraction module to learn local features at each level and the feature propagation module to integrate multiscale hierarchical features. The proposed method allows us to create effective and simple neural networks for learning local and global features of point clouds. To address the MHNet segmentation inconsistency of one object category, a CRF model is constructed to optimize the predicted class labels. We demonstrate that MHNet has a significant improvement over state-of-the-art for semantic segmentation tasks on the S3DIS dataset and ScanNet dataset. We also conduct comprehensive experiments to justify the effectiveness of our proposed method.

However, there are also many problems that need to be solved in our network. Although our method improves the accuracy of semantic segmentation, the computational burden is large due to the iterate farthest point sampling and ball query, and it consumes more computations in the higher local context resolutions. In the future, it is worthwhile to consider how to accelerate the run time of our proposed network. Additionally, a robust solution to handle large-scale point clouds for scene understanding would be an interesting work. In addition, the CRF only acts as a postprocessing module to optimize the segmentation results in our method. The CRF global optimization only improves the overall accuracy by less than 1.00%. This is because the CRF reclassifies the noise in point clouds rather than segmenting point clouds again. Although there are considerable noise points in the segmentation result of MHNet, the number of noise points only accounts for a small part of the total point clouds. Even if all the noise points are corrected, the overall accuracy of the point clouds will not be greatly improved. In the next step, embedding it into the network to construct an end-to-end system should be considered. We expect wide application of the proposed method in 3D semantic segmentation and hope the design provides a basis for our future work on indoor modeling research.

REFERENCES

[1] A. Boulch, B. Le Saux, and N. Audebert, "Unstructured point cloud semantic labeling using deep segmentation networks," in *Proc. Eurographics Workshop 3D Object Retr.*, 2017, pp. 17–24.

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[3] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[4] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," Apr. 2017, *arXiv:1704.06857*. [Online]. Available: https://arxiv.org/abs/1704.06857

[5] L. P. Tchapmi, C. B. Choy, I. Armeni, J. Gwak, and S. Savarese, "SEGCloud: Semantic segmentation of 3D point clouds," Oct. 2017, *arXiv:1710.07563*. [Online]. Available: https://arxiv.org/abs/1710.07563

[6] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep./Oct. 2015, pp. 922–928.

[7] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

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

[9] R. Zhang, "Research on polymorphic object semantic segmentation of complex 3D scenes based on laser point clouds," Ph.D. dissertation, Dept. Survey Map, PLA Strategic Support Force Inf. Eng. Univ., Zhengzhou, China, 2018.

[10] L. Wang, "Intelligent segmentation of point cloud based on PointNet network," M.S. thesis, Dept. RS Inf. Eng., Wuhan Univ., Wuhan, China, 2018.

[11] F. Engelmann, T. Kontogianni, A. Hermans, and B. Leibe, "Exploring spatial context for 3D semantic segmentation of point clouds," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2017, pp. 716–724.

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

[13] W. Wang, R. Yu, Q. Huang, and U. Neumann, "SGPN: Similarity group proposal network for 3D point cloud instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 2569–2578.

[14] Q.-H. Pham, D. T. Nguyen, B.-S. Hua, G. Roig, and S.-K. Yeung, "JSIS3D: Joint semantic-instance segmentation of 3D point clouds with multi-task pointwise networks and multi-value conditional random fields," Apr. 2019, *arXiv:1904.00699*. [Online]. Available: https://arxiv.org/abs/1904.00699

[15] Y. Mohammed, J. K. David, J. I. Emmett, and S. Carl, "A multi-scale fully convolutional network for semantic labeling of 3D point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 143, pp. 191–204, Sep. 2018.

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

[17] M. Jiang, Y. Wu, T. Zhao, Z. Zhao, and C. Lu, "PointSIFT: A SIFT-like network module for 3d point cloud semantic segmentation," Jul. 2018, *arXiv:1807.00652*. [Online]. Available: https://arxiv.org/abs/1807.00652

[18] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," Nov. 2018, *arXiv:1811.07246*. [Online]. Available: https://arxiv.org/abs/1811.07246

[19] W. Guan, S. You, and G. Pang, "Estimation of camera pose with respect to terrestrial LiDAR data," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Jan. 2013, pp. 391–398.

[20] J. Huang and S. You, "Point cloud matching based on 3D self-similarity," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2012, pp. 41–48.

[21] J. Huang and S. You, "Detecting objects in scene point cloud: A combinational approach," in *Proc. Int. Conf. 3D Vis. (3DV)*, Jun./Jul. 2013, pp. 175–182.

[22] J. Huang and S. You, "Pole-like object detection and classification from urban point clouds," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2015, pp. 3032–3038.

[23] R. Qiu and U. Neumann, "Exemplar-based 3D shape segmentation in point clouds," in *Proc. IEEE 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 203–211.

[24] R. Qiu and U. Neumann, "IPDC: Iterative part-based dense correspondence between point clouds," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–9.

[25] R. Qiu, Q.-Y. Zhou, and U. Neumann, "Pipe-run extraction and reconstruction from point clouds," in *Proc. Eur. Conf. Comput. Vis.*, Zürich, Switzerland, Sep. 2014, pp. 17–30.

[26] D. Bazazian, J. R. Casas, and J. Ruiz-Hidalgo, "Segmentation-based multi-scale edge extraction to measure the persistence of features in unorganized point clouds," in *Proc. 12th Int. Conf. Comput. Vis. Theory Appl. (VISAPP)*, vol. 4, 2017, pp. 317–325.
[27] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Niener, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, 2017, pp. 5828–5839.
[28] J. Huang and S. You, "Vehicle detection in urban point clouds with orthogonal-view convolutional neural network," in *Proc. IEEE Int. Conf. Imag. (ICIP)*, Sep. 2016, pp. 2593–2597.
[29] C. R. Qi, H. Su, M. Niessner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5648–5656.
[30] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner, "Vote3Deep: Fast object detection in 3D point clouds using efficient convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May/Jun. 2017, pp. 1355–1361.
[31] J. Guerry, A. Boulch, B. Le Saux, J. Moras, A. Plyer, and D. Filliat, "SnapNet-R: Consistent 3D multi-view semantic labeling for robotics," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 669–678.
[32] L. Yi *et al.*, "Large-scale 3D shape reconstruction and segmentation from ShapeNet Core55," Oct. 2017, *arXiv:1710.06104*. [Online]. Available: https://arxiv.org/abs/1710.06104
[33] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6620–6629.
[34] R. Klokov and V. Lempitsky, "Escape from cells: Deep Kd-networks for the recognition of 3D point cloud models," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 863–872.
[35] Y. Li, S. Pirk, H. Su, C. R. Qi, and L. J. Guibas, "FPNN: Field probing neural networks for 3D data," in *Proc. NIPS*, Dec. 2016, pp. 307–315.
[36] B. Graham, "Spatially-sparse convolutional neural networks," Sep. 2014, *arXiv:1409.6070*. [Online]. Available: https://arxiv.org/abs/1409.6070
[37] B. Graham, "Sparse 3D convolutional neural networks," May 2015, *arXiv:1505.02890*. [Online]. Available: https://arxiv.org/abs/1505.02890
[38] 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. (ICCV)*, Dec. 2015, pp. 945–953.
[39] G. Pang and U. Neumann, "3D point cloud object detection with multi-view convolutional neural network," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 585–590.
[40] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on $\mathcal{X}$-transformed points," Jan. 2018, *arXiv:1801.07791*. [Online]. Available: https://arxiv.org/abs/1801.07791
[41] Q. Huang, W. Wang, and U. Neumann, "Recurrent slice networks for 3D segmentation of point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 2626–2635.
[42] X. Ye, J. Li, H. Huang, D. Liang, and X. Zhang, "3D recurrent neural networks with context fusion for point cloud semantic segmentation," in *Proc. 15th Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 415–430.
[43] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," Jan. 2018, *arXiv:1801.07829*. [Online]. Available: https://arxiv.org/abs/1801.07829
[44] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," Feb. 2015, *arXiv:1502.03240*. [Online]. Available: https://arxiv.org/abs/1502.03240
[45] 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," Oct. 2017, *arXiv:1710.07368*. [Online]. Available: https://arxiv.org/abs/1710.07368
[46] L. Wang, Y. Huang, J. Shan, and L. He, "MSNet: Multi-scale convolutional network for point cloud classification," *Remote Sens.*, vol. 10, no. 4, p. 612, Apr. 2018.
[47] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2011, pp. 109–117.
[48] A. Hermans, G. Floros, and B. Leibe, "Dense 3D semantic mapping of indoor scenes from RGB-D images," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May/Jun. 2014, pp. 2631–2638.
[49] D. Wolf, J. Prankl, and M. Vincze, "Fast semantic segmentation of 3D point clouds using a dense CRF with learned parameters," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2015, pp. 4867–4873.
[50] Q.-H. Pham, B.-S. Hua, D. T. Nguyen, and S.-K. Yeung, "Real-time progressive 3D semantic segmentation for indoor scene," Apr. 2018, *arXiv:1804.00257*. [Online]. Available: https://arxiv.org/abs/1804.00257
[51] S. Yang, Y. Huang, and S. Scherer, "Semantic 3D occupancy mapping through efficient high order CRFs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 590–597.
[52] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3D semantic parsing of large-scale indoor spaces," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1534–1543.
[53] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia, "Associatively segmenting instances and semantics in point clouds," Feb. 2019, *arXiv:1902.09852*. [Online]. Available: https://arxiv.org/abs/1902.09852
[54] F. Engelmann, T. Kontogianni, J. Schult, and B. Leibe, "Know what your neighbors do: 3D semantic segmentation of point clouds," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, Sep. 2018, pp. 395–409.

XIAOLI LIANG is currently pursuing the Ph.D. degree in photogrammetry and remote sensing with the School of Remote Sensing and Information Engineering, Wuhan University, China. Her research interests include remote sensing image information extraction, point cloud processing, and indoor 3D reconstruction.

ZHONGLIANG FU received the B.S., M.S., and Ph.D. degrees in photogrammetry and remote sensing from the Wuhan Technical University of Survey and Mapping, Wuhan, China, in 1985, 1988, and 1996, respectively. He is currently a Professor and a Ph.D. Advisor with the School of Remote Sensing and Information Engineering, Wuhan University. He is also the Director of the Geographic Information System Department. His interests include spatial data management and update, remote sensing image processing and analysis, map scanning image recognition, vehicle license plate recognition, and geographic information engineering technology.