SCF-Net: Learning Spatial Contextual Features for Large-Scale Point Cloud Segmentation

Siqi Fan[1 2], Qiulei Dong[†2 3 4], Fenghua Zhu[1], Yisheng Lv[†1], Peijun Ye[1], Fei-Yue Wang[1]
[1]State Key Laboratory for Management and Control of Complex Systems, CASIA
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]National Laboratory of Pattern Recognition, CASIA
[4]Center for Excellence in Brain Science and Intelligence Technology, CAS
{fansiqi2019, fenghua.zhu, yisheng.lv, peijun.ye, feiyue.wang} @ia.ac.cn, qldong@nlpr.ia.ac.cn

Abstract

*How to learn effective features from large-scale point clouds for semantic segmentation has attracted increasing attention in recent years. Addressing this problem, we propose a learnable module that learns Spatial Contextual Features from large-scale point clouds, called **SCF** in this paper. The proposed module mainly consists of three blocks, including the local polar representation block, the dual-distance attentive pooling block, and the global contextual feature block. For each 3D point, the local polar representation block is firstly explored to construct a spatial representation that is invariant to the z-axis rotation, then the dual-distance attentive pooling block is designed to utilize the representations of its neighbors for learning more discriminative local features according to both the geometric and feature distances among them, and finally, the global contextual feature block is designed to learn a global context for each 3D point by utilizing its spatial location and the volume ratio of the neighborhood to the global point cloud. The proposed module could be easily embedded into various network architectures for point cloud segmentation, naturally resulting in a new 3D semantic segmentation network with an encoder-decoder architecture, called SCF-Net in this work. Extensive experimental results on two public datasets demonstrate that the proposed SCF-Net performs better than several state-of-the-art methods in most cases.*

1. Introduction

With the rapid development of 3D sensors, semantic segmentation of 3D point clouds has attracted more and more attention in the computer vision field. Compared with 2D images, 3D point clouds could provide richer geometric information of scenes. However, semantic segmentation of 3D point clouds, particularly segmentation of large-scale point clouds, is still a challenging task due to the fact that 3D point clouds are generally unstructured and unordered.

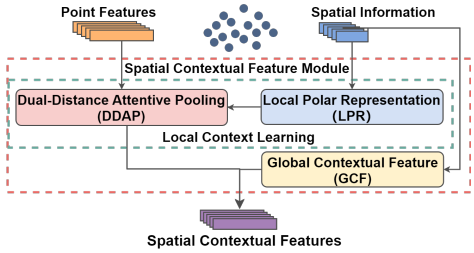

Figure 1. Diagram of the spatial contextual feature (SCF) module.

In recent years, a lot of DNN (Deep Neural Network)-based methods have been proposed for segmenting 3D point clouds [29, 30, 40, 22, 46, 10]. These methods could be roughly divided into 3 categories [11]: projection-based methods [21, 2], discretization-based methods [10, 33, 27, 15], and point-based methods [14, 39, 35, 7, 46, 44, 3, 29, 30, 45]. Both the projection-based and discretization-based methods are computationally expensive to handle large-scale point clouds, which need extra procedures to transform point clouds to a regular representation and project the intermediate segmentation results back to the point clouds. Different from those methods, point-based methods directly work on 3D point clouds. Although some existing point-based methods have achieved promising performances on small-sized point clouds, they could not deal with the large-scale point clouds. Recently, some methods designed for large-scale point clouds have been proposed, such as SPG [20], PCT [4] and RandLA-Net [13]. However, most of them still have to confront with the following problem: **how to learn more effective features from large-scale point clouds for semantic segmentation?**

[†]Corresponding author.

Inspired by the success of contextual information in many visual tasks [43, 5, 24, 19, 28], we investigate how to learn spatial contextual features from large-scale point clouds for semantic segmentation here. We decompose the aforementioned problem into three sub-problems as:

1) how to represent the local context of a 3D point?

2) how to learn local contextual features?

3) how to learn global contextual features?

Addressing the three subproblems, we propose a learnable module, called SCF in this paper, consisting of 3 blocks, including the local polar representation block, the dual-distance attentive pooling block, and the global contextual feature block. The diagram of SCF is shown in Figure 1. For each 3D point, the local polar representation block is firstly explored to construct a z-axis rotation-invariant representation in a polar coordinate system for representing the local context. Then the representations of its neighbors are integrated to learn effective local features by utilizing the weights learnt by the dual-distance attentive pooling block. Finally, the global contextual feature block learns global context of each 3D point by utilizing both the location and the volume ratio of the neighborhood. Various network architectures could utilize the proposed module SCF for point cloud segmentation, and under an encoder-decoder architecture, a new 3D semantic segmentation network is presented, called SCF-Net in this work. In sum, the main contributions are listed as follows:

- We propose the Local Polar Representation (LPR) block, which could learn locally z-axis rotation-invariant representation for each 3D point.
- We propose the Dual-Distance Attentive Pooling (DDAP) block, which could automatically learn effective local features based on both the geometric and feature distances.
- We propose the Global Contextual Feature (GCF) block, which could learn the global context of each 3D point from the point cloud.
- We propose the SCF module, which could be applied to various architectures for exploring new point cloud segmentation networks. Extensive experimental results in Section 4 demonstrate that the proposed SCF-Net by embedding the SCF module into a standard encoder-decoder architecture achieves state-of-the-art performances.

2. Related Work

In this section, we introduce the three mentioned categories of point cloud segmentation methods in Section 1, including the projection-based methods, the discretization-based methods, and the point-based methods in detail.

2.1. Projection-based Methods

To leverage the 2D segmentation methods, many existing works aim to project 3D point clouds into 2D images and then process 2D semantic segmentation. For example, the point clouds were transformed to multi-view representations in [21, 2]. However, the projection inevitably causes the information loss of the details. Besides, these methods need to project back the intermediate segmentation results to the point clouds, which is computationally expensive.

2.2. Discretization-based Methods

The discretization-based methods convert the point cloud into a discrete representation, such as voxel. The point cloud was voxelized into 3D grids and fed to a fully-3D CNN for voxel-wise segmentation [15]. Many works [10, 33, 27] achieved point clouds semantic segmentation based on discretization. In particular, Fully-Convolutional Point Network (FCPN) [31] can process massive point clouds. However, the performance of these methods is sensitive to the granularity of the voxels, and the voxelization inherently introduces discretization artifacts.

2.3. Point-based Methods

Different from the projection-based and discretization-based methods, point-based methods directly work on the point clouds. These methods can be generally classified as point convolution and pointwise MLP (Multi-Layer Perceptron) methods. Inspired by the successful application of convolution operators for images, many works [14, 39, 35, 7] tended to propose convolution methods for point clouds. The pointwise MLP methods use shared MLP as the basic unit. The pioneering work of these methods, PointNet [29], learnt per-point features. However, per-point features cannot capture the local geometric patterns, and the contextual features among points are lost. To deal with that, many methods have been explored recently, which mainly utilize two techniques, including neighboring feature pooling and attention-based aggregation.

Neighboring feature pooling: The information from local neighbors are aggregated for each point in these methods [30, 8, 46, 17, 44]. PointNet++ [30] improved the performance of PointNet by grouping points hierarchically and learning local features with increasing contextual scale. Different from that, two neighborhoods were generated in world and feature space leveraging K-means clustering and KNN [8]. PointWeb [46] was proposed to extract contextual features from local neighborhood by densely constructing a locally fully-linked web. Inspired by the 2D descriptor SIFT [25], Jiang et al. [17] proposed PointSIFT module. The orientation-encoding was achieved by encoding the information from eight crucial orientations.

Attention-based aggregation: These methods introduce attention mechanism [37] to further improve the per-

formance. Yang et al. [41] developed Point Attention Transformers to model the interactions between points. A Local Spatial Aware layer was proposed by Chen et al. [3] to learn Spatial Distribution Weights and capture the local geometric structure.

To capture contextual features and geometric structures, several works tried to achieve segmentation resorting to graph networks [20, 18, 38, 26] and RNN (Recurrent Neural Networks) [6, 42, 47, 23].

RandLA-Net [13] utilized random sampling to achieve high efficiency and leveraged local feature aggregation module to learn and preserve geometric patterns.

3. Methodology

In this section, we firstly propose the SCF module for large-scale point cloud segmentation, consisting of three blocks, LPR, DDAP and GCF. Then we present the SCF-Net, which has an encoder-decoder with the SCF module.

3.1. SCF Module

The SCF module is proposed to learn spatial contextual features. We introduce the three proposed blocks in detail, and describe the architecture of the SCF module in this subsection.

3.1.1 Local Polar Representation

It is noted that in many real scenes, the orientations of the objects belonging to a same class are generally different, such as chairs in a conference room, indicating that the features directly learnt from the input 3D points are orientation-sensitive. Such an orientation-sensitive case could hamper the segmentation performance to some extent. Addressing this issue, we propose the LPR for learning a z-axis rotation-invariant representation, which represents the local context of a 3D point in a polar coordinate system instead of a Cartesian coordinate system. Different from the design of the 3D shape descriptor [9], the architecture of the LPR is shown in Figure 2.

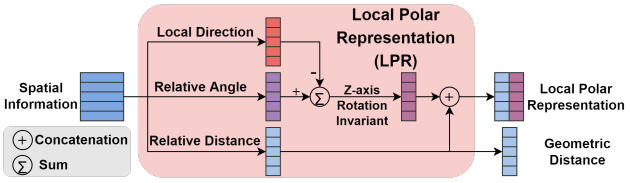

Figure 2. Architecture of the local polar representation block.

As seen from Figure 2, the local spatial information is fed into the LPR block, and the output is the polar representation and the geometric distance.

LPR includes the following steps:

Constructing initial local representation: Firstly, calculate the relative coordinates of neighboring points in the polar coordinate system. For a point p_i, its K-nearest neighbors $\{p_i^1, p_i^2, \ldots, p_i^k, \ldots, p_i^K\}$ are gathered by the KNN (K nearest neighbors) algorithm based on Euclidean distances. The local representation is expressed as $(dis_i^k, \phi_i^k, \theta_i^k)$.

$$dis_i^k = \sqrt{x_i^{k^2} + y_i^{k^2} + z_i^{k^2}} \tag{1}$$

$$\phi_i^k = \arctan(\frac{y_i^k}{x_i^k}) \tag{2}$$

$$\theta_i^k = \arctan(\frac{z_i^k}{\sqrt{x_i^{k^2} + y_i^{k^2}}}) \tag{3}$$

where (x_i^k, y_i^k, z_i^k) is the relative coordinate in the Cartesian coordinate system.

Calculating the local direction: We then calculate the center-of-mass point p_i^m of the local neighborhood. The local direction is defined as the direction from p_i to p_i^m, which has the following two advantages:

a) The center-of-mass point can reflect the general picture of the local neighborhood;

b) The randomness introduced by down sampling can be effectively reduced by using the mean value in the calculation of p_i^m.

Updating the ϕ_i^k and θ_i^k: The ϕ_i^k and θ_i^k are updated to $\phi_i^{k'}$ and $\theta_i^{k'}$, respectively as:

$$\phi_i^{k'} = \phi_i^k - \alpha_i \tag{4}$$

$$\theta_i^{k'} = \theta_i^k - \beta_i \tag{5}$$

where α_i and β_i are the relative angle of p_i^m. As noted in (4) and (5), $\phi_i^{k'}$ and $\theta_i^{k'}$ remains unchanged when point clouds rotate around z axis.

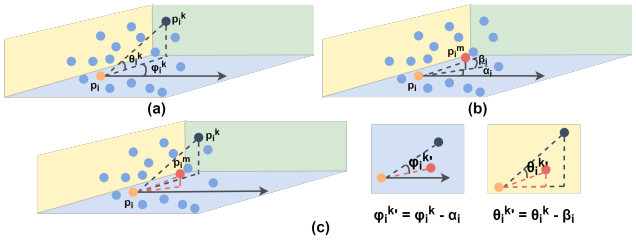

Figure 3. Illustration of updating ϕ_i^k and θ_i^k. (a) the original relative angles ϕ_i^k and θ_i^k; (b) the relative angles α_i and β_i of the local direction; (c) the updated $\phi_i^{k'}$ and $\theta_i^{k'}$.

The update algorithm is shown in Figure 3. The relative angle of p_i^k (dark blue) is ϕ_i^k and θ_i^k. The local direction is from point p_i (orange) to the barycenter point p_i^m (red). The relative angle is updated to $\phi_i^{k'}$ and $\theta_i^{k'}$, respectively.

After the LPR block, the local representation is invariant to the z-axis rotation.

3.1.2 Dual-Distance Attentive Pooling

Given the local representation, the next problem to be faced is how to learn local contextual features utilizing the neighboring point features. Heuristically, distance is an important variable to measure the correlation among points. The smaller the distance is, the more relevant they are. Therefore, we propose the dual-distance attentive pooling block to automatically learn effective local contextual features by integrating the neighboring point features $\{\mathbf{f_i^1}, \mathbf{f_i^2}, \ldots, \mathbf{f_i^k}, \ldots, \mathbf{f_i^K}\}$. The architecture of the DDAP is shown in Figure 4.

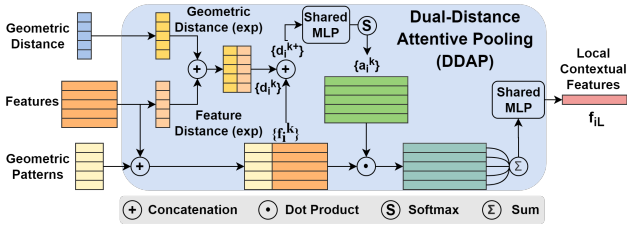

Figure 4. Architecture of the dual-distance attentive pooling block.

As seen from Figure 4, DDAP has three inputs, including geometric distance, point features and geometric patterns. To generate geometric patterns, the local representation output by the LPR are concatenated with the absolute coordinates, and further processed via a shared MLP.

Particularly, we focus on two representative distances, the geometric distance d_{ig}^k in the world space and the feature distance d_{if}^k in the feature space. Without loss of generality, let $g(i)$ and $g(k)$ denote the input feature vectors of the i-th point and its k-th $(k = 1, 2, ..., K)$ neighbor to the DDAP block respectively. The feature distance d_{if}^k between $g(i)$ and $g(k)$ is defined as:

$$d_{if}^k = \text{mean}(|g(i) - g(k)|) \tag{6}$$

where '$|\cdot|$' is the L_1 norm and $\text{mean}(\cdot)$ is the mean function. The negative exponential of both are used to learn the attentive pooling weights. Besides, we use λ to tune d_{if}^k to handle its instability, because features are automatically learnt by the network.

$$\mathbf{d_i^k} = \exp(-d_{ig}^k) \oplus \lambda \exp(-d_{if}^k) \tag{7}$$

where '$\oplus$' is the concatenation operator.

In addition, the dual-distance $\mathbf{d_i^k}$ and the feature $\mathbf{f_i^k}$ is merged via concatenation.

$$\mathbf{d_i^{k+}} = \mathbf{d_i^k} \oplus \mathbf{f_i^k} \tag{8}$$

Then, a shared MLP followed by softmax is applied to $\mathbf{d_i^{k+}}$, and the attentive pooling weight $\mathbf{a_i^k}$ is learnt automatically as:

$$\mathbf{a_i^k} = softmax(MLP(\mathbf{d_i^{k+}})) \tag{9}$$

Finally, the local contextual features are obtained by calculating the weighted-sum of the neighboring point features with the learnt weights $\mathbf{a_i^k}$.

$$\mathbf{f_{iL}} = \sum_{k=1}^{K} (\mathbf{a_i^k} \cdot \mathbf{f_i^k}) \tag{10}$$

3.1.3 Global Contextual Feature

Local contextual feature describes the context among points in the neighborhood, but it is not discriminative enough for semantic segmentation. To obtain more effective features, we propose the global contextual feature block to learn the global context from 3D points. The illustration of the GCF is displayed in Figure 5.

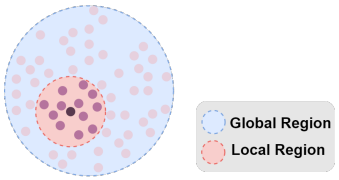

Figure 5. Illustration of the global contextual feature block.

As seen from Figure 5, both the local and global spatial information are used. The region is shown as a circular area, which is actually a 3D spherical space.

We utilize the location and volume ratio r_i in the global context representation. It is noted that a same category of objects (e.g. chairs) in different scenes usually have various styles, and their geometric architectures are generally similar, but not exactly the same. Hence, considering that the volume ratio is not sensitive to the positions of the inner points within the local and global bounding spheres, we use it so that the representation could tolerate slight geometric deformations of the objects of a same category.

$$r_i = \frac{v_i}{v_g} \tag{11}$$

where v_i is the volume of the neighborhood's bounding sphere corresponding to p_i, and v_g is the volume of the bounding sphere of the point cloud.

The x-y-z coordinate of p_i is used to represent the location of the local neighborhood. Therefore, the global contextual features are defined as $\mathbf{f_{iG}}$.

$$\mathbf{f_{iG}} = MLP((x_i, y_i, z_i) \oplus r_i) \tag{12}$$

where (x_i, y_i, z_i) is the coordinate of p_i, and '$\oplus$' is the concatenation operator.

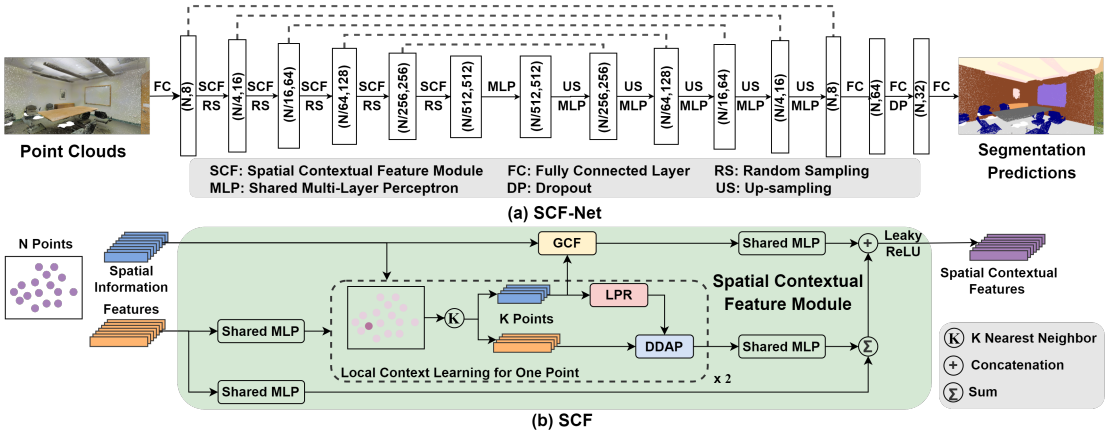

Figure 6. Architecture of the SCF-Net and the SCF module. The local context learning for one point is shown.

3.1.4 Architecture of SCF

The architecture of the SCF module is shown in Figure 6(b). Its inputs are the spatial information and the features learnt previously. The spatial information is utilized to learn both the local and global contextual features, while the learnt features are only used for local feature learning. The local contextual features are learnt by the LPR and DDAP blocks. Figure 6(b) shows the local contextual feature learning for one point, which is applied to each point in parallel. The local context representations constructed by the LPR are automatically integrated by the DDAP. We learn local contextual features twice to increase contextual information. Then, the features are further added with another feature map, resulting in the local features. The global contextual features are learnt from the spatial information by the GCF block. The output of the module is the learnt spatial contextual feature, which is the concatenation of the local and global contextual features.

3.2. Architecture of SCF-Net

In this subsection, we embedded the proposed SCF module into a standard encoder-decoder architecture, resulting the new segmentation network, SCF-Net. The complete architecture of SCF-Net is shown in Figure 6(a).

As seen from Figure 6(a), the input of the network is a point cloud of size $N \times d$, where N is the number of the points and d is the input feature dimension. The per-point features are firstly extracted by a fully connected layer, and the dimension is unified to 8. Five encoder layers are utilized progressively to encode the features. Among them, random sampling is used to down sample the point cloud, and the SCF module is embedded to learn spatial contextual features. The number of points is gradually decreased from N to $\frac{N}{512}$, while the feature dimension is increased from 8 to 512. Next, five decoder layers are used to decode the features. The encoded features are up sampled through the nearest-neighbor interpolation, which simply utilizes the value at the nearest neighbor as the interpolated value, and further concatenated with the intermediate feature map through skip connection. At last, three consecutive fully-connected layers are used to predict the semantic labels. The output is the segmentation predictions of size $N \times c$, where c is the number of classes. Besides, the cross entropy loss is used for training.

4. Experiments

In this section, we evaluate our SCF-Net on two typical large-scale point cloud benchmarks, S3DIS [1] and Semantic3D [12]. The experiments* are implemented in the Tensorflow on a server with NVIDIA Titan Xp GPUs, CUDA 9.0 and cuDNN v7.

In addition, we also report the corresponding results of 9 methods [29, 16, 42, 20, 22, 46, 44, 35, 13] on the S3DIS and the results of 10 methods [2, 33, 34, 32, 44, 38, 20, 35, 36, 13] on the Semantic3D for comparison, including SPG, KPConv, and RandLA-Net.

4.1. Implementation Detail and Dataset

We use the Adam optimizer with an initial learning rate of 10^{-2}. The batch size is set as 4 and 3 when training with S3DIS and Semantic3D, respectively. The network is trained for 100 epochs, with learning rate dropped by 5% after each epoch. The number of neighbors is set to be 16 (K=16) for efficiency. A fixed number of points ($\approx 10^5$) are sampled from each training point cloud for network training, while the whole raw test point clouds are used for testing. Each point is represented by 3D coordinates and color information in the experiments.

*The code is available at https://github.com/leofansq/SCF-Net

Methods	mIoU (%)	mAcc (%)	OA (%)	Ceil.	Floor	Wall	Beam	Col.	Wind.	Door	Table	Chair	Sofa	Book.	Board	Clut.
PointNet [29]	47.6	66.2	78.6	88.0	88.7	69.3	42.4	23.1	47.5	51.6	54.1	42.0	9.6	38.2	29.4	35.2
RSNet [16]	56.5	66.5	-	92.5	92.8	78.6	32.8	34.4	51.6	68.1	59.7	60.1	16.4	50.2	44.9	52.0
3P-RNN [42]	56.3	-	86.9	92.9	93.8	73.1	42.5	25.9	47.6	59.2	60.4	66.7	24.8	57.0	36.7	51.6
SPG [20]	62.1	73.0	86.4	89.9	95.1	76.4	62.8	47.1	55.3	68.4	**73.5**	69.2	63.2	45.9	8.7	52.9
PointCNN [22]	65.4	75.6	88.1	**94.8**	**97.3**	75.8	63.3	51.7	58.4	57.2	71.6	69.1	39.1	61.2	52.2	58.6
PointWeb [46]	66.7	76.2	87.3	93.5	94.2	80.8	52.4	41.3	64.9	68.1	71.4	67.1	50.3	62.7	62.2	58.5
ShellNet [44]	66.8	-	87.1	90.2	93.6	79.9	60.4	44.1	64.9	52.9	71.6	**84.7**	53.8	64.6	48.6	59.4
KPConv [35]	70.6	79.1	-	93.6	92.4	**83.1**	63.9	**54.3**	**66.1**	**76.6**	57.8	64.0	**69.3**	**74.9**	61.3	60.3
RandLA-Net [13]	70.0	82.0	88.0	93.1	96.1	80.6	62.4	48.0	64.4	69.4	69.4	76.4	60.0	64.2	65.9	60.1
SCF-Net (Ours)	**71.6**	**82.7**	**88.4**	93.3	96.4	80.9	**64.9**	47.4	64.5	70.1	71.4	81.6	67.2	64.4	**67.5**	**60.9**

Table 1. Quantitative results of different methods on S3DIS. The classwise metric is IoU(%).

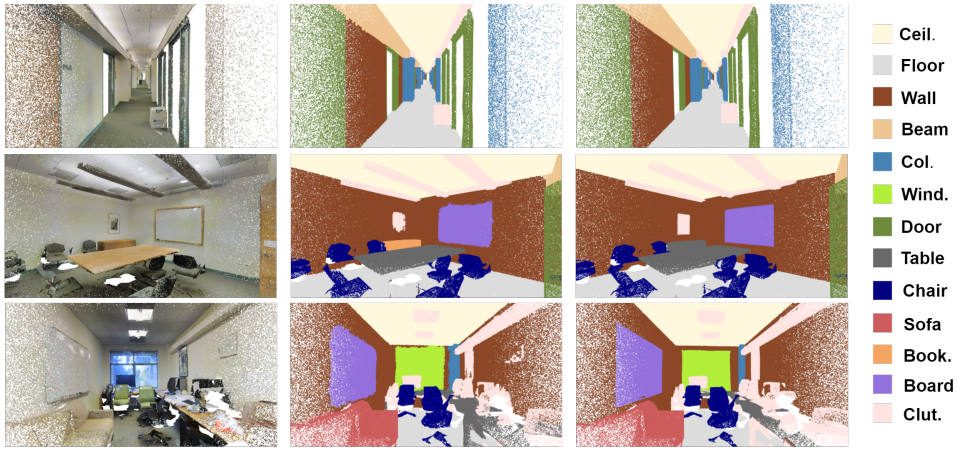

Figure 7. Visualization examples of three typical indoor scenes (hallway, conference room and office) on S3DIS. Left: RGB colored input point clouds; Middle: Predictions obtained via the proposed SCF-Net; Right: Ground truths.

S3DIS is a large-scale indoor point cloud dataset, which consists of point clouds of 6 areas including 271 rooms. Each point cloud is a medium-sized room, and each point is annotated with one of the semantic labels from 13 classes.

Semantic3D is a large-scale outdoor point cloud dataset with over 3 billion points from real-world, including urban and rural scenes. It consists of 15 training point clouds and 15 online testing point clouds. In addition to coordinates and color information, each point also has intensity values, but we do not use them. Each point is annotated with one of the semantic labels from 8 classes.

4.2. Evaluation on S3DIS

As done in [29], we perform 6-fold cross validation to evaluate our methods. The mean Intersection-over-Union (mIoU), mean class Accuracy (mAcc) and Overall Accuracy (OA) are used as standard metrics.

The quantitative results of all the referred methods are reported in Table 1. As seen from this table, our method performs better than others on all the three metrics (mIoU, mAcc and OA), and achieves the best performance on 3 categories, including beam, board, and clutter.

The visualization examples of three typical indoor scenes are shown in Figure 7, including hallway, conference room and office. In general, semantic segmentation of indoor scenes is difficult, because some categories are hard to distinguish, such as white boards on white walls. Our method performs well on the board class, which can be seen from both the quantitative and qualitative results. Nevertheless, the misclassification is inevitable. As shown in the middle row of Figure 7, a table (center area) in the conference room is misclassified to bookcase.

4.3. Evaluation on Semantic3D

We submit our results to the sever and evaluate on the reduced set of 4 subsampled point clouds. The mIoU and OA of the test data are compared.

We report the quantitative results of all the referred methods in Table 2. As seen from this table, our method has the best mIoU among all these methods. As for OA, it is slightly lower than for RandLA-Net, but it is better than all the other compared methods. Besides, SCF-Net also achieves the best performance on car segmentation.

The visualization results are shown in Figure 8. Note that

Methods	mIoU (%)	OA (%)	Time (s)	Man-made.	Natural.	High Veg.	Low Veg.	Buildings	Hardscape	Artefacts	Cars
SnapNet_ [2]	59.1	88.6	3600.0	82.0	77.3	79.7	22.9	91.1	18.4	37.3	64.4
SEGCloud [33]	61.3	88.1	1881.0	83.9	66.0	86.0	40.5	91.1	30.9	27.5	64.3
RF_MSSF [34]	62.7	90.3	1643.8	87.6	80.3	81.8	36.4	92.2	24.1	42.6	56.6
MSDeepVoxNet [32]	65.3	88.4	115000.0	83.0	67.2	83.9	36.7	92.4	31.3	50.0	78.2
ShellNet [44]	69.3	93.2	3000.0	96.3	90.4	83.9	41.0	94.2	34.7	43.9	70.2
GACNet [38]	70.8	91.9	1380.0	86.4	77.7	**88.5**	**60.6**	94.2	37.3	43.5	77.8
SPG [20]	73.2	94.0	3000.0	97.4	92.6	87.9	44.0	93.2	31.0	63.5	76.2
KPConv [35]	74.6	92.9	600.0	90.9	82.2	84.2	47.9	94.9	40.0	**77.3**	79.7
RGNet [36]	74.7	94.5	-	**97.5**	**93.0**	88.1	48.1	94.6	36.2	72.0	68.0
RandLA-Net [13]	77.4	**94.8**	-	95.6	91.4	86.6	51.5	**95.7**	**51.5**	69.8	76.8
SCF-Net (Ours)	**77.6**	94.7	563.6	97.1	91.8	86.3	51.2	95.3	50.5	67.9	**80.7**

Table 2. Quantitative results of different methods on the reduced-8 split of Semantic3D. The runtime of the compared methods is obtained from the benchmark. The classwise metric is IoU(%).

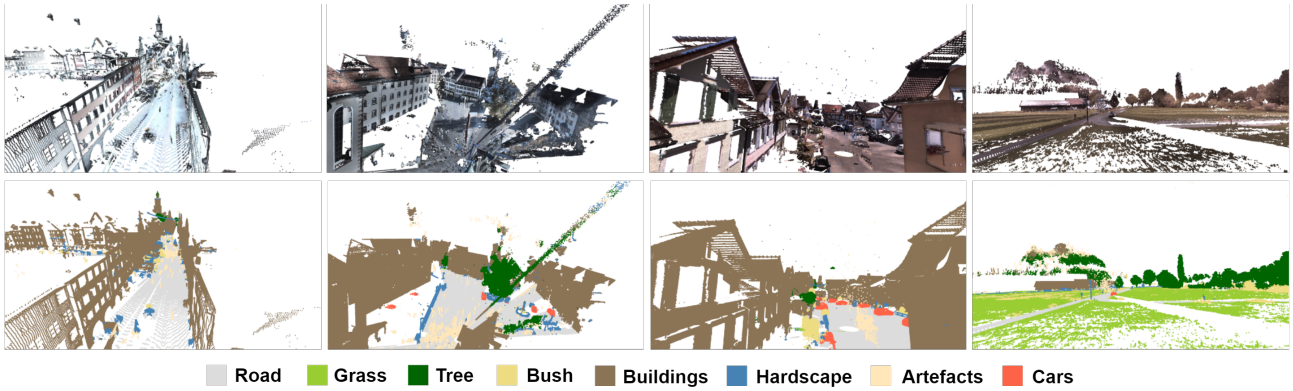

Figure 8. Visualization results on the reduced-8 split of Semantic3D. Top: RGB colored input point clouds; Bottom: Predictions obtained via the proposed SCF-Net. Note that the ground truth of the test set is not publicly available.

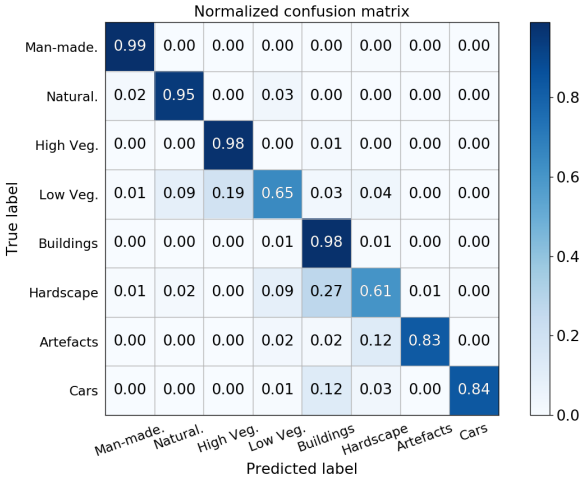

Figure 9. Confusion matrix on Semantic3D.

the ground truth of the test set is not publicly available, so only RGB colored point clouds and predictions are shown.

The confusion matrix shown in Figure 9 provides a detailed look at the error sources. Most of the errors are caused by the hard scape and the low vegetation classes. Probably because the hard scape class includes several kinds of objects, and the appearances of the three natural classes are similar.

4.4. Ablation Study

The effectiveness of our approach is verified by the experimental results on S3DIS and Semantic3D. To better understand the network, we further evaluate it and conduct the following two groups of experiments. The experiments are conducted on S3DIS due to the lack of public ground truth of Semantic3D test set.

4.4.1 Ablation Study on SCF

The following ablation studies are conducted to study the impacts of the three proposed blocks. We use the standard 6-fold cross validation to evaluate the ablated networks, and show the comparison in Table 3.

First of all, we remove the GCF block. The improvement from the second row to the first row demonstrates that

	mIoU(%)
SCF-Net	**71.6**
removing GCF	70.5
removing GCF & replacing DDAP with SAP	69.3
removing GCF & replacing DDAP with SAP & replacing LPR with LSR	67.9

Table 3. Results of ablated networks on S3DIS. SAP is attentive pooling only based on features themselves. In LSR, the representation is $p_i \oplus p_i^k \oplus (p_i - p_i^k) \oplus ||p_i - p_i^k||$.

the introduction of global contextual features can effectively improve the understanding of the scene. Secondly, we additionally replace the DDAP block with a normal self attentive pooling (SAP). Only neighboring point features are taken into consideration while learning the pooling weights. The effectiveness of the DDAP is verified in the third row. The mIoU is improved by 1.2%. Finally, we replace the LPR block with a normal local spatial representation (LSR). The representation is $p_i \oplus p_i^k \oplus (p_i - p_i^k) \oplus ||p_i - p_i^k||$ in LSR. The replacing of LPR diminishes segmentation performance by 1.4%, which shows the impact of it. The distance information is explicitly encoded in both LPR and LSR. The main difference between them is the representing method of the relative position. In LSR, it is represented in Cartesian coordinates, while it is represented as the relative angle to the local direction in LPR, which is not sensitive to the rotation around the z-axis. The improvement confirms the importance of the local representation.

4.4.2 Ablation Study on DDAP

The following ablation studies are conducted to understand the impacts of various design choices made in DDAP.

First of all, we study the influences of the distances. All ablated networks in this study are evaluated on area 2 of S3DIS, which is the most difficult area according to the mIoU results. We remove the feature and the geometric distance in turn, and report the comparison in Table 4. The removal of the feature distance diminishes segmentation performance by 0.4%, while 1.5% decline is caused by the removal of the geometric distance. From that, the effectiveness of the dual-distance is demonstrated. In addition, it can be seen that the improvement benefited from the geometric distance is greater, which also shows the importance of focusing on the spatial contextual features.

	mIoU(%)
dual-distance	**59.7**
removing feature distance	59.3
removing both geometric and feature distance	57.8

Table 4. Distances influences on DDAP.

Secondly, we explore different fusion methods of the dual-distance $\mathbf{d_i^k}$ and the feature $\mathbf{f_i^k}$. The experiments are also conducted on area 2. We evaluate two typical fusion methods, concatenation and weighted summation, and report the comparison in Table 5. For weighted summation, three weight ratios are taken into consideration. The experimental results show that concatenation is better than weighted summation. The weighted summation with 5:5 ratio is the best among the three, which demonstrates the effectiveness of both $\mathbf{d_i^k}$ and $\mathbf{f_i^k}$.

	mIoU(%)
concatenation	**59.7**
weighted summation (5:5)	58.9
weighted summation (9:1)	55.2
weighted summation (1:9)	58.7

Table 5. Fusion methods of the dual-distance $\mathbf{d_i^k}$ and the feature $\mathbf{f_i^k}$. The ratio is expressed as $(\mathbf{d_i^k} : \mathbf{f_i^k})$

Finally, we evaluate three values of λ. The experiments are conducted on 6 areas to investigate the general effect, and the comparison is reported in Table 6. It can be seen that 0.1 is a better choice, which achieves the best performance on five of the six areas.

	Area 1	Area 2	Area 3	Area 4	Area 5	Area 6
1	74.4	59.4	76.7	58.0	62.0	79.0
0.1	**75.1**	**59.7**	**78.4**	60.2	**63.4**	**80.2**
0.01	75.1	59.5	77.2	**61.7**	61.6	80.0

Table 6. Comparison of mIoU(%) with different values of λ.

5. Conclusion

In this paper, we propose the learnable module SCF to learn effective features from large-scale point clouds for semantic segmentation. The proposed module mainly consists of three blocks, including the local polar representation block, the dual-distance attentive pooling block, and the global contextual feature block. The LPR and DDAP blocks are used for learning discriminative local contextual features, while the GCF block is proposed to learn the global contextual features. SCF could be easily embedded into various network architectures for point cloud segmentation, and we embed it into an encoder-decoder architecture, resulting in the SCF-Net in this work. Extensive experimental results on S3DIS and Semantic3D demonstrate that the proposed method achieves state-of-the-art performances on both indoor and outdoor scenes.

Acknowledgments This work was supported by the National Key R&D Program of China 2018YFB1004803, the National Natural Science Foundation of China (61991423, U1811463, U1805264, 61876011), the Strategic Priority Research Program of the Chinese Academy of Sciences XDB32050100, and Chinese Guangdong's project (2020B0909050001, 2019B1515120030).

References

[1] 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 *CVPR*, pages 1534–1543, 2016. 5

[2] A. Boulch, B. Le Saux, and N. Audebert. Unstructured point cloud semantic labeling using deep segmentation networks. In *3DOR*, pages 17–24, 2017. 1, 2, 5, 7

[3] L. Chen, X. Li, D. Fan, M. Cheng, K. Wang, and S. Lu. LSANet: Feature learning on point sets by local spatial attention. *arXiv preprint arXiv:1905.05442*, 2019. 1, 3

[4] S. Chen, S. Niu, T. Lan, and B. Liu. Large-scale 3d point cloud representations via graph inception networks with applications to autonomous driving. In *ICIP*, pages 4395–4399, 2019. 1

[5] Y. Ding and J. Xiao. Contextual boost for pedestrian detection. In *CVPR*, pages 2895–2902, 2012. 2

[6] F. Engelmann, T. Kontogianni, A. Hermans, and B. Leibe. Exploring spatial context for 3d semantic segmentation of point clouds. In *ICCVW*, pages 716–724, 2017. 3

[7] F. Engelmann, T. Kontogianni, and B. Leibe. Dilated Point Convolutions: On the receptive field size of point convolutions on 3d point clouds. In *ICRA*, pages 9463–9469, 2020. 1, 2

[8] F. Engelmann, T. Kontogianni, J. Schult, and B. Leibe. Know what your neighbors do: 3d semantic segmentation of point clouds. In *ECCV*, pages 395–409, 2018. 2

[9] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *ECCV*, pages 224–237, 2004. 3

[10] B. Graham, M. Engelcke, and L. Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, pages 9224–9232, 2018. 1, 2

[11] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun. Deep learning for 3d point clouds: A Survey. *arXiv preprint arXiv:1912.12033*, 2019. 1

[12] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys. SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume IV-1-W1, pages 91–98, 2017. 5

[13] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham. RandLA-Net: Efficient semantic segmentation of large-scale point clouds. In *CVPR*, pages 11105–11114, 2020. 1, 3, 5, 6, 7

[14] B. Hua, M. Tran, and S. Yeung. Pointwise convolutional neural networks. In *CVPR*, pages 984–993, 2018. 1, 2

[15] J. Huang and S. You. Point cloud labeling using 3d convolutional neural network. In *ICPR*, pages 2670–2675, 2016. 1, 2

[16] Q. Huang, W. Wang, and U. Neumann. Recurrent slice networks for 3d segmentation of point clouds. In *CVPR*, pages 2626–2635, 2018. 5, 6

[17] M. Jiang, Y. Wu, T. Zhao, Z. Zhao, and C. Lu. PointSIFT: A sift-like network module for 3d point cloud semantic segmentation. *arXiv preprint arXiv:1807.00652*, 2018. 2

[18] L. Landrieu and M. Boussaha. Point cloud oversegmentation with graph-structured deep metric learning. In *CVPR*, pages 7432–7441, 2019. 3

[19] L. Landrieu, H. Raguet, B. Vallet, C. Mallet, and M. Weinmann. A structured regularization framework for spatially smoothing semantic labelings of 3d point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 132:102–118, 2017. 2

[20] L. Landrieu and M. Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, pages 4558–4567, 2018. 1, 3, 5, 6, 7

[21] F. J. Lawin, M. Danelljan, P. Tosteberg, G. Bhat, F. S. Khan, and M. Felsberg. Deep projective 3d semantic segmentation. In *CAIP*, pages 95–107, 2017. 1, 2

[22] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. PointCNN: Convolution on x-transformed points. In *NIPS*, pages 820–830, 2018. 1, 5, 6

[23] F. Liu, S. Li, L. Zhang, C. Zhou, R. Ye, Y. Wang, and J. Lu. 3DCNN-DQN-RNN: A deep reinforcement learning framework for semantic parsing of large-scale 3d point clouds. In *ICCV*, pages 5679–5688, 2017. 3

[24] N. Liu, J. Han, and M. Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *CVPR*, pages 3089–3098, 2018. 2

[25] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 2

[26] Y. Ma, Y. Guo, H. Liu, Y. Lei, and G. Wen. Global context reasoning for semantic segmentation of 3d point clouds. In *WACV*, pages 2920–2929, 2020. 3

[27] H. Meng, L. Gao, Y. Lai, and D. Manocha. VV-Net: Voxel vae net with group convolutions for point cloud segmentation. In *ICCV*, pages 8499–8507, 2019. 1, 2

[28] J. Niemeyer, F. Rottensteiner, and U. Soergel. Contextual classification of lidar data and building object detection in urban areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87:152–165, 2014. 2

[29] C. R. Qi, H. Su, M. Kaichun, and L. J. Guibas. PointNet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 77–85, 2017. 1, 2, 5, 6

[30] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, pages 5100–5109, 2017. 1, 2

[31] D. Rethage, J. Wald, J. Sturm, N. Navab, and F. Tombari. Fully-convolutional point networks for large-scale point clouds. In *ECCV*, pages 625–640, 2018. 2

[32] X. Roynard, J. Deschaud, and F. Goulette. Classification of point cloud scenes with multiscale voxel deep network. *arXiv preprint arXiv:1804.03583*, 2018. 5, 7

[33] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese. SEGCloud: Semantic segmentation of 3d point clouds. In *3DV*, pages 537–547, 2017. 1, 2, 5, 7

[34] H. Thomas, F. Goulette, J. Deschaud, B. Marcotegui, and Y. L. Gall. Semantic classification of 3d point clouds with multiscale spherical neighborhoods. In *3DV*, pages 390–398, 2018. 5, 7

[35] H. Thomas, C. R. Qi, J. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas. KPConv: Flexible and deformable

convolution for point clouds. In *ICCV*, pages 6410–6419, 2019. 1, 2, 5, 6, 7

[36] G. Truong, S. Z. Gilani, S. M. S. Islam, and D. Suter. Fast point cloud registration using semantic segmentation. In *DICTA*, pages 1–8, 2019. 5, 7

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, U. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, pages 5999–6009, 2017. 2

[38] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan. Graph attention convolution for point cloud semantic segmentation. In *CVPR*, pages 10288–10297, 2019. 3, 5, 7

[39] S. Wang, S. Suo, W. Ma, A. Pokrovsky, and R. Urtasun. Deep parametric continuous convolutional neural networks. In *CVPR*, pages 2589–2597, 2018. 1, 2

[40] W. Wu, Z. Qi, and L. Fuxin. PointConv: Deep convolutional networks on 3d point clouds. In *CVPR*, pages 9613–9622, 2019. 1

[41] J. Yang, Q. Zhang, B. Ni, L. Li, J. Liu, M. Zhou, and Q. Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *CVPR*, pages 3318–3327, 2019. 3

[42] X. Ye, J. Li, H. Huang, L. Du, and X. Zhang. 3d recurrent neural networks with context fusion for point cloud semantic segmentation. In *ECCV*, pages 415–430, 2018. 3, 5, 6

[43] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal. Context encoding for semantic segmentation. In *CVPR*, pages 7151–7160, 2018. 2

[44] Z. Zhang, B. S. Hua, and S. K. Yeung. ShellNet: Efficient point cloud convolutional neural networks using concentric shells statistics. In *ICCV*, pages 1607–1616, 2019. 1, 2, 5, 6, 7

[45] C. Zhao, W. Zhou, L. Lu, and Q. Zhao. Pooling scores of neighboring points for improved 3d point cloud segmentation. In *ICIP*, pages 1475–1479, 2019. 1

[46] H. Zhao, L. Jiang, C. Fu, and J. Jia. PointWeb: Enhancing local neighborhood features for point cloud processing. In *CVPR*, pages 5560–5568, 2019. 1, 2, 5, 6

[47] Z. Zhao, M. Liu, and K. Ramani. DAR-Net: Dynamic aggregation network for semantic scene segmentation. *arXiv preprint arXiv:1907.12022*, 2019. 3