IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 8, 2020, accepted June 19, 2020, date of publication June 23, 2020, date of current version July 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3004472

Spatial-Related Correlation Network for 3D Point Clouds

DAN WANG, GUOQING HU, AND CHENGZHI LYU, (Graduate Student Member, IEEE)
School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510640, China
Corresponding authors: Guoqing Hu (gqhu@scut.edu.cn) and Chengzhi Lyu (sanchilongquan@outlook.com)

ABSTRACT Due to the irregularity and inconsistency of 3D point clouds, it is difficult to extract features directly from them. Existing methods usually extract point features independently and then use the max-pooling operation to aggregate local features, which limits the feature representation capability of their models. In this work, we design a novel spatial-related correlation path, which considers both spatial information and point correlations, to preserve high dimensional features, thereby capturing fine-detail information and long-distance context of the point cloud. We further propose a new network to aggregate the spatial aware correlations with point-wise features and global features in a learnable way. The experimental results show that our method can achieve better performance than the state-of-the-art approaches on challenging datasets. We can achieve 0.934 accuracy on ModelNet40 dataset and 0.875 mean IoU (Intersection over Union) on ShapeNet dataset with only about 2.42 million parameters.

INDEX TERMS 3D point clouds, feature extraction, point correlations, neural network.

I. INTRODUCTION

Object recognition is one of the most classical and fundamental problems in computer vision. It is very useful as a preprocessing step in various computer vision applications, such as image classification and segmentation [1], [2], 3D reconstruction [3], [4], object detection [5] and pose estimation [6]. In general, the process of object recognition can be formulated in terms of labeling problems. The performance in 2D domain has been significantly improved due to the rapid development of convolutional neural networks. However, it is still difficult to directly apply classic conventional operations on 3D point clouds due to the irregularity and inconsistent of point clouds. To deal with the problem, some alternative approaches project 3D point sets into volumetric grids or multiple 2D image views to take advantage of using 3D convolution. But in this case, it will lead to high computational cost and information loss.

Recently, many researches focus on point-based networks which directly process point cloud data to improve memory efficient. The pioneering work, PointNet [2], proposed to learn point-wise features using the multi-layer perceptron (MLP) and global features using the max pooling layer. It is a simple yet computationally efficient model for extracting features from point clouds. However, unlike classification tasks, segmentation tasks require both global context understanding and local details of points. Since the features in PointNet are learned independently, it is difficult to capture the local context and the mutual interactions between points, which limits its performance. To learn richer neighboring context information, many hierarchical structure-based networks have been introduced. PointNet++ [7] proposed a hierarchical network to capture local features. They used PointNet to aggregate local neighborhood information, that is, points in a local region were treated independently. As a result, they ignored the relationships among points, which may lose rich information and limit feature representation. PointConv [8] used spatial coordinates to learn the convolutional kernel weights and density distributions for point clouds. However, their performance is limited due to a lack of geometric information. LSANet [9] constructed spatial relations in the neighborhood to capture local geometric features. They took the spatial distribution of the local region into account, thus handling geometric transform robustly. Despite considerable progress, their work lacks long-distance contextual information, which may limit their performance.

To address the above problems, we propose a spatial-related point correlation network that extract feature from point clouds effectively and efficiently. Our aim is to use point connections to extract rich contextual information, which includes local details as well as long-range contexts. Specifically, we first propose a novel spatial-related correlation path

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

(SRC path), which constructs two parallel branches to capture high dimensional detail information. Each point feature extracting from SRC path is related to dense correlation and spatial distribution. Then, we construct a simple yet efficient network based on SRC path to combine the point-wise features, fine-detail features and global features together, which can better describe features at different levels. Furthermore, we introduce an adaptive fusion block to adjust the channel weights of the combined features in a learnable way. Our method has three advantages. First, instead of generating point features from a local neighborhood, we build dense point connections with spatial awareness to capture discriminative features. Second, we do not need to explicitly select representative points and search for their neighbors. Third, our method can preserve high dimensional features and better describe the details for point clouds. The contributions of the proposed method are summarized as follows:

1. We introduce a SRC path which is capable of modeling point correlations both in feature space and in spatial space.
2. We design an efficient method to implement SRC path. It greatly reduces the parameters while still retaining high dimensional features.
3. A simple and efficient network based on SRC path is proposed, which can be used as a basic module to extract features from point clouds in other applications.
4. We have conducted experiments on two challenging tasks to verify the effectiveness of our method. The experimental results show that our network is competitive against other state-of-the-art models.

II. RELATE WORKS

PointNet [2] first proposed a network to directly processing raw point clouds. It is computationally efficient and has achieved great results in many 3D tasks, but it lacks local context modeling abilities. To integrating local features, many hierarchical structure based method have been proposed. PointNet++ [7] used a farthest sampling layer to select centroid points and employed multi-scale grouping and multi-resolution grouping to get the neighboring regions. [10] used k-means clustering and k-Nearest Neighbor to generate neighborhoods in world space and feature space, respectively. 3PNet [11] introduced a point-wise pyramid pooling module to capture multi-scale information and applied a recurrent neural network to get long-distance contexts. PointSIFT [12] introduced a novel module to implement scale awareness and orientation encoding. It generated local neighboring points from eight spatial directions to implement orientation encoding. However, these methods have not fully exploited the relationships between different points, thus rich information may be lost when aggregating local features from neighborhood points.

PointCNN [13] proposed to learn a feature transformer that can permute the point set into a canonical order and applied convolutional operations on the transformed points. Inspired of image convolution, KPConv [14] proposed a new point convolution and its deformable version. KPConv can use any number of kernel points, which gives it a lot of flexibility. The convolution weights were carried by kernel points and their range of influence is defined by the correlation function. KCNet [15] presented kernel correlations to extract neighborhood information and employed local high dimensional features through recursive feature aggregation. DGCNN [16] employed point relationship to improve the representation of local feature. It established a pair relationship between the center point and its neighboring points and dynamically updated the relationship of different layers in the network. However, it did not explicitly consider spatial information. PointWeb [17] proposed a novel module to find the interactions between points. They explored the relationships between all pairs of the points in a local area to extract the local information. Huang *et al.* [18] employed a novel hierarchical data augmentation strategy to improve the learning of deep features and reduced feature dimensions with a nonlinear manifold learning to remove redundant information. Li *et al.* [19] introduced a novel geometry-attentional network for ALS point cloud classification. They proposed GA-Conv to learn an embedding from low-level geometric information to high-level local features and construed a dense hierarchical architecture based on GA-Conv to capture multiple scale information.

Although hierarchical structure-based methods can integrate local features, it has three limitations that restrict their performance. First, the performance of hierarchical structure-based methods is sensitive to the certain points and radius selections. Second, since point clouds are unordered without explicit neighboring information, most previous methods require high extra computational cost in the selection of centroid points and search of neighboring points. It limits the efficiency of hierarchical structure-based models. Third, each local region is processed separately, thus the feature is limited to the current neighborhood. Different from existing methods, we efficiently construct high dimensional point correlations to exhaust rich contextual information. This strategy enhances the point feature extraction ability of our model. Thus, our model can better describe detail information for point clouds.

III. PROPOSED METHOD

Our method is motivated by the following facts. First, due to the lack of structural information and long-range contexts, the performance of existing methods for extracting point cloud features is limited. Second, most existing methods need neighboring feature pooling operations to enlarge receptive fields and capture wider context, resulting in a low dimension of the feature map. Third, shared MLP operations on points at different positions utilize the same weights, thus there is a lack of consideration of spatial distribution when extracting features.

To deal with these problems, we introduce a SRC path to capture long-distance contextual information for each point, which consider both spatial information and

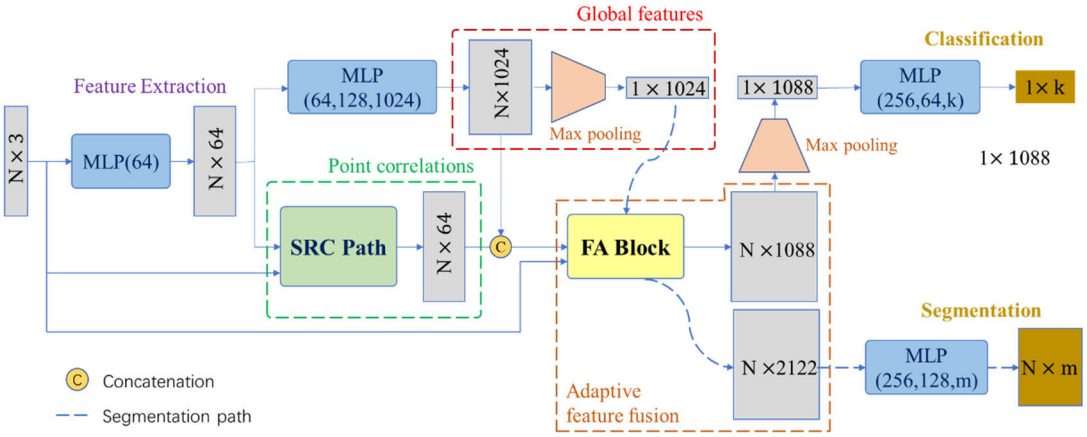

FIGURE 1. Overall framework of the proposed method.

point correlations. The spatial information is learned according to the point coordinates, while the point correlations are determined by the entire feature space, which can incorporate detail features as well as long range contexts. We further combine point-wise features, fine-detail features and global feature togethers, as shown in Fig. 1. This strategy enhances the feature representative ability of our model. Therefore, it can better describe the details of point clouds. Inspired by the T-Net in PointNet [2], we first apply a 3×3 transformation matrix to the coordinates of points to align the point cloud, and then use the adjusted points as input. In the following subsections, we will elaborate on our model. Note that the proposed method can also be served as the basic feature extraction block for other types of models.

A. SRC PATH

Most of the existing methods use adjacent points or patches to extract local features. This structure has limited expressions for long-distance connections. In order to improve the ability in modeling discriminative features, we construct dense connections to model the relationship between different points in the point set. We employ two parallel branches in SRC path. The first branch is used to extract spatial aware weights and the second branch is used to capture contextual information. The structure of SRC path is illustrated in Fig.2. Unlike previous methods that used shared MLP and symmetric functions to extract local features, we focused on modeling point correlations with long range connections, preserving detailed information. In this way, since all points are involved, the features can be learned more thoroughly. We then combine the two branch outputs so that the output of SRC path is determined by both spatial coordinates and point correlations. Let the 3D point data denote by $\mathrm{x} = \left\{(x_i^P, x_i^F) | x_i^P \in R^3, x_i^F \in R^M, i = 1, \ldots, n\right\}$, here x_i^P and x_i^F represent the coordinates and additional feature of the point, respectively. n is the number of points and M is the feature dimension. The output feature of point i is fomulated as:

$$F_i = \omega_i^P \otimes f^F \left(x_i^F, \Omega\right) \tag{1}$$

where ω_i^P denotes the spatial aware weight. f^F refers to the enhanced feature of x_i^F which involves point correlations. Ω denotes the feature set of all points and $\otimes$ is the element-wise multiplication.

1) SPATIAL WEIGHT BRANCH

The spatial aware weight can be seen as a nonlinear function of the point coordinate and it can be learned through the spatial weight branch. We use a shared MLP to learn the weight for each point, which is expressed as:

$$\omega_i^P = f^P \left(x_i^P\right) = MLP(x_i^P, W^P) \tag{2}$$

Here W^P is the parameters of the MLP. In order to preserve permutation invariance, we share the same weight W^P for all points. For a point cloud, the parameters in the spatial weight branch are the same, but the input x_i^P is different. Thus, the output ω_i^P computed from the branch is different. In this way, we can get spatially related output from this branch, which is determined by the learned parameters and the point coordinates. Then we use the spatial aware weight to re-weight the correlations learned by feature branch, as shown in Equation (1). That is, by combining spatial weights with point correlations, we can take the spatial distribution into account in the feature extracting process.

2) POINT CORRELATION BRANCH

The enhanced features for x_i^F is given by:

$$f^F \left(x_i^F, \Omega\right) = f^C \left(x_i^F, \Omega\right) + x_i^F \tag{3}$$

where f^C is the point correlation information that is learned through the second branch, as illustrated in Fig. 2. A residual connection is used to reinforce the input feature. The main challenge of f^c is the model size and computational cost. To address this problem, we first use 1×1 convolutions to reduce the feature map from M-dimensional space to 1 dimensional space. To preserve permutation invariance, we apply the same weight to all points in the point set.

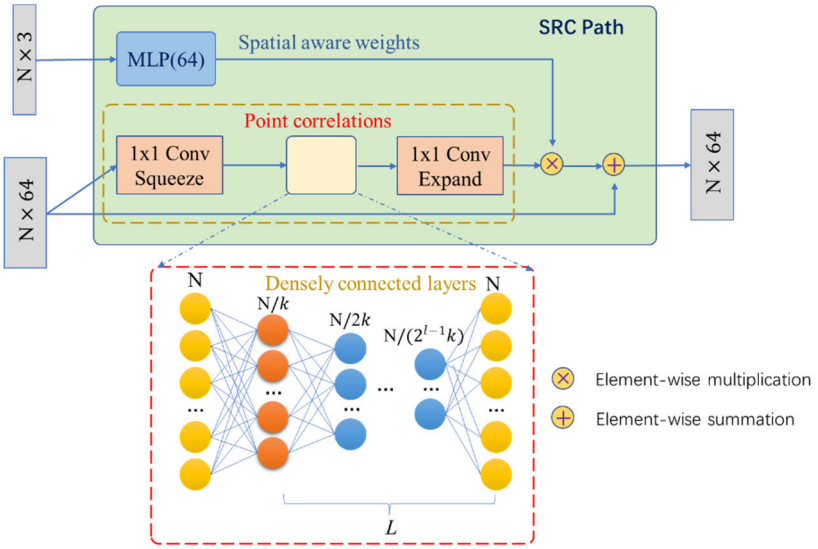

FIGURE 2. The structure of SRC path. It uses densely connected layers to model the long-rang relationships between points.

The formula can be expressed as follows:

$$\tilde{x}_i^F = Conv_{1\times1}\left(x_i^F\right) \tag{4}$$

The feature maps are then delivered to the next part which has a series of densely connected layers to learn point correlations, as shown in Fig. 2. Let a_l refers to the output of layer l, where $1 \le l \le L$ and L is the total number of layers. a_l is given by:

$$a_l = \begin{cases} \sigma\left(W_l^H \tilde{x}_i^F\right) & l = 1 \\ \sigma\left(W_l^H a_{l-1}\right) & 1 < l \le L \end{cases} \tag{5}$$

where σ denotes the activation function, $W_l^H \in R^{N_l \times N_l - 1}$ is the parameters in layer l and N_l is the number of nodes in layer l. In order to reduce the parameters of the model, we reduce the number of nodes of the middle layer with a constant k. The number of nodes in layer l is $N/(2^{l-1}k)$. However, we use N nodes in the last layer to keep the dimension of the output the same as the input. Thus, the last layer output $a_L \in R^{N\times1}$, here N is the number of input points.

Finally, we increase the channel of a_L to M and add a residual connection to the input x_i^F. Therefore, we can generate high dimensional feature maps and learn long range context through SRC path. The overall result of f^F is:

$$f^F(x_i^F) = Conv_{1\times1}(a_L) + x_i^F \tag{6}$$

The structure is designed to make us learn long range context, while drastically reducing the model parameters. It takes an input feature map $x^F \in R^{N\times M}$ and produces an output feature map $x^{out} \in R^{N\times M}$. For a standard fully connected layer for each channel, it requires $N\times N \times M$ parameters. These parameters are dependent on the number of points N and the dimension of the input point features M. To reduce model parameters, we use two strategies. First, we use bottleneck design in this branch to reduce the network parameters. Second, we decrease the number of nodes for middle layers. For the structure in the red box in Fig.2 (L= 3 and k= 4), there are $\frac{13}{32}N^2 + 2M$ paremeters, which is less than one hundredth of the standard layer parameters (N= 1024 and M = 64). We will analyze the different choices of L and k in Section IV.

Now, for point i in Ω, the overall output of the spatial correlations path is calculated as:

$$F_i = MLP(x_i^P, W^P) \otimes (Conv_{1\times1}(a_L) + x_i^F) \tag{7}$$

SRC path models point correlations in the feature space, as well as applies spatial aware weights on the correlations to compensate for the point spatial distribution. Thus, it can better describe fine-detail features for the point set. One of the advantages of our method is that we are able to produce high dimensional features. Compared with the hierarchical based approaches, our model is able to maintain the same number of points throughout the network, while still having the ability to model neighborhood information. This high-resolution information is essential for extracting fine detail features.

B. FUSION

Our network is built on SRC path and the overall framework is illustrated in Fig. 1. We first learn point-wise features from normalized point coordinates through MLP layers. These features are then given to SRC path and max pooling layers for extracting point correlations and global information. We integrate global, local and point-wise features for the final output. Previous approaches usually concatenated local features and global features to get the finally output, which considered each feature equally. However, the interaction between different scale features may differ across the channels. For example, flat regions may prefer larger scale receptive fields, while corner or edge regions may prefer more detailed features.

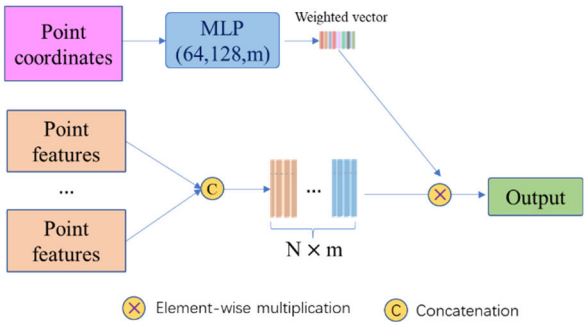

FIGURE 3. The illustration of the proposed FA block.

Motivated by this, we propose a fusion adjustment block (FA block) to combine different scale features, as shown in Fig. 3.

In FA block, we adjust the channel weights of the combined features. The filter parameters can be seen as the weight vector that identifies the importance of each channel. For each filter, the weight vector is learned by MLP from point coordinates. In this way, the fusion way for different scale features is learnable. We first apply MLP on the point coordinates to get the fusion weights. Let define $\hat{\alpha}_i$ as the fusion weight for point i and it is calculated as:

$$\hat{\alpha}_i = MLP(x_i^P) \tag{8}$$

The weight matrix is denoted as $\hat{\alpha} = [\hat{\alpha}_1, \ldots \alpha_N]^T$. We use $\hat{\alpha}$ to adjust the channel weights of the combined features, as shown in Fig. 3. Let F^G, F^P and F^{SRC} denote the point-wise features, global features and correlation features, respectively. For the segmentation path, the output of FA block is given by:

$$F = \hat{\alpha} \otimes [F^P, F^{SRC}, F^G] \tag{9}$$

where [] denotes the concatenation operation and $\otimes$ represents element-wise product. Therefore, we can make the network adaptively merge the different path features.

IV. EXPERIMENTS

To evaluate the performance of the proposed method, we have conducted experiments on both classification and segmentation tasks of point clouds.

A. IMPLEMENTATION DETAILS

During the training process, we used stochastic gradient descent (SGD) with a mini-batch size of 1. Base learning rate and weight decay were set to 0.001 and 10^{-4} for optimization. Momentum was set to 0.9. The proposed network was implemented with Tensorflow 1.13.1 and Python 3.5.

B. EVALUATION METRIC

To evaluate the performance of our method, we employed overall accuracy, average class accuracy, mean class IoU (Intersection over Union) and mean part IoU as metrics. Here the accuracy and IoU are calculated as:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

TABLE 1. Comparison results on ModelNet40 dataset, "M" denotes to million.

	Input	Overall accuracy	Avg. class accuracy	Parameters
PointNet[2]	points	0.892	0.862	3.48M
PointNet++[7]	points	0.907	-	1.48M
PointCNN[13]	points	0.922	0.881	0.6M
DGCNN[16]	points	0.922	0.902	1.84M
LSANet [9]	points	0.932	0.903	2.30M
SpiderCNN[21]	points+normal	0.924	-	5.48M
PointWeb[17]	points+normal	0.923	0.894	-
Ours	points	0.934	0.921	2.42M

$$Iou = \frac{TP}{TP + FP + FN} \tag{11}$$

where the variables TP, FP, TN, FN refer to the number of True Positive, False Positive, True Negative and False Negative respectively.

C. COMPARISON RESULTS

1) CLASSIFICATION

We chose ModelNet40 and ModelNet10 [15] for the point cloud classification task. ModelNet40 consists of 12,311 CAD models from 40 categories and is divided into two parts: 9,843 models for training and 2,468 models for testing. ModelNet10 dataset contains 4,899 CAD models from 10 categories, of which 3991 models are used for training and 908 models are used for testing. Following [2], we uniformly sampled 1,024 points from each model in ModelNet40 and ModelNet10. During the training phase, we randomly shuffled each model and rotated the input points to augment the dataset.

The classification results of different methods on ModelNet40 are summarized in Table 1. Avg. class accuracy denotes the mean accuracy across the object classes. It can be observed that our model obtains the highest score of accuracy compared with other approaches. It is 0.002 higher than LSANet [9], which is the second highest score. In term of per-class accuracy, we also achieve the best score. It is 0.018 higher than the second-best method. The reason for better performance is mainly because we have built long-distance connections to model point relationships, which can associate more points, thus providing us with more detailed information. Furthermore, the proposed method only uses point coordinates as input, while SpiderCNN [21] and PointWeb [17] utilize both point coordinates and normal.

We show the quantitative comparison results of ModelNet10 dataset in Table 2. Our method achieves 0.969 for overall accuracy, which is the highest score of all the compared methods. In terms of average class accuracy, SONet [22] gets the highest score and it is 0.002 higher than us. However, the overall accuracy of our method is better and it is 0.012 higher than SONet. It shows an effective way to directly process the point cloud and improve the capability of its feature representation.

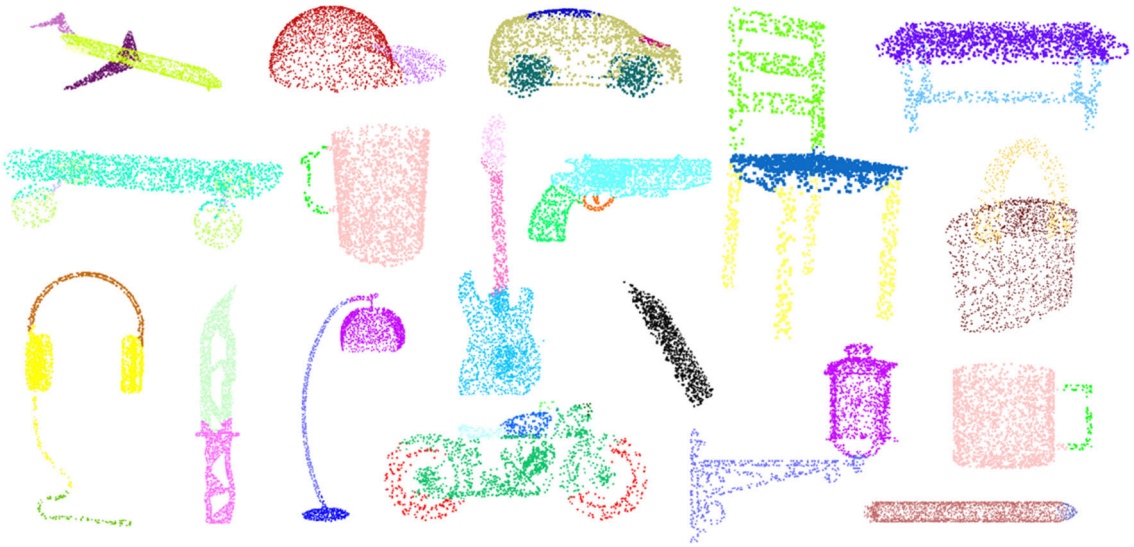

FIGURE 4. Segmentation examples on ShapeNet dataset.

TABLE 2. Comparison results on ModelNet10 dataset.

	Input	Overall accuracy	Avg. class accuracy
Kd-Net[23]	points	0.940	0.935
DensePoint[24]	points	0.966	-
OctNet[25]	octree	0.909	0.901
LP-3DCNN[26]	voxel	0.944	-
SONet[22]	points+normal	0.957	0.955
Ours	points	0.969	0.953

TABLE 3. Quantitative comparison on ShapeNet dataset.

	inputs	mean cIoU	mean pIoU
PointNet[2]	points	0.804	0.837
KCNet[15]	points	0.822	0.837
DGCNN[16]	points	0.823	0.851
DensePoint[24]	points	0.842	0.864
OG-PointNet++[28]	points	-	0.851
PointNet++[7]	points+normal	0.819	0.851
SpiderCNN[21]	points+normal	0.824	0.853
Ours	points	0.843	0.872

2) SEGMENTATION

In order to further verify the effectiveness of our model, we also evaluated our model in the part segmentation task. We used ShapeNet [27] dataset in our experiment, which is a highly competitive dataset in this field. It consists of 16,881 shapes from 16 categories. Most objects are marked as two to five parts, for a total of 50 parts. We followed [2] to split the training and testing datasets and randomly sampled 2048 points as input.

Table 3 demonstrates the comparison of the proposed model with other state-of-the-art methods. In Table 3, mean cIoU denotes the mean IoU across all classes and mean pIoU denotes the mean IoU across all parts. It can be obviously observed that our model achieves top performance. DensePoint [24] gets the second highest score of mean cIoU and it is quite close to us. However, the part average IoU of our method is better. Note that we only took point clouds as input while [7] and [21] also takes normal as input. Some examples are visualized in Fig. 4. We have noticed that our model can accurately segment part boundaries and details.

D. ABLATION STUDIES

To verify the effectiveness of the proposed SRC path and adaptive fusion block, we have conducted ablation experiments in this section. We established a simple structure as our baseline. It employed only point-wise features and global features to make final prediction. We first investigated the effectiveness of SRC path. Large receptive field and long-distance context have important effects on segmentation performance. Therefore, we propose spatial aware dense connections in SRC path to preserve long-distance contextual information. Table 4 lists the segmentation results with and without SRC path. By integrating point correlations in the network, we obtain 0.935 for accuracy, with an increment of 3.7% from the baseline. Meanwhile, mean class IoU is improved from 0.798 to 0.832, with an increase of 4.3%. This indicates that our method benefits from the SRC path. This is because SRC path effectively employ dense connections to build long-distance point correlations. With this design, our model is able to integrate local detailed features and capture discriminative information. The visualization of the results is presented in Fig. 5(a) and (b). SRC path enables our model to integrate point correlation and fine-detail information, thereby obtaining part boundaries and details more accurately than the basic network.

Further, we analyzed the proposed FA block in the network, which is designed to effectively fuse different scale features. It approximates a weight function to adjust the fusion way

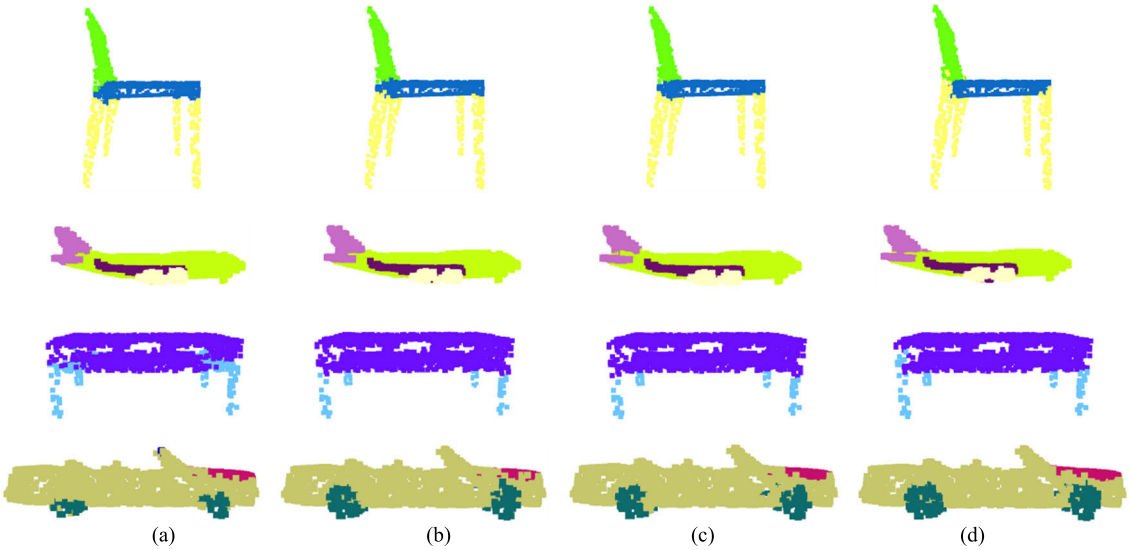

FIGURE 5. Visualization results of using different variants on ShapeNet dataset: (a) Baseline, (b) Baseline + SRC, (c) Baseline + SRC + FA, and (d) ground truth.

TABLE 4. Experimental results on ShapeNet dataset with different network variants. "C" refers to simply concatenation and "F" refers to FA block.

	Basic	SRC	FA	Accuracy	mean cIoU	mean pIoU
Baseline	✓			0.901	0.798	0.835
Baseline + SRC (c)	✓	✓		0.935	0.832	0.861
Baseline + SRC(f)(ours)	✓	✓	✓	0.947	0.843	0.872

and learns its parameters from the point coordinates through MLP layers. Thus, the pixel-wise features, global features and SRC features can be fused in a learnable way. The merged features are then transferred to the segmentation path for final prediction, as shown in Fig. 1. In order to study the effectiveness of FA blocks, we also established a simple way to directly connect the features of different scales. It considers each feature equally. The comparison results are illustrated in the second and third rows of Table 4. With FA block, the accuracy is increased from 0.901 to 0.947 and exceeds the baseline by 5.1%. In addition, it improves the accuracy by 0.012 compared with the simply concatenation. In terms of IoU, FA block achieves an increment of 5.6% and 4.4% for mean class IoU and mean part IoU from the baseline, respectively. The results show that FA block has proved to be effective for the feature fusion. It enhances the feature interaction ability. Fig. 5 shows some visualization examples for qualitative comparison. It can be clearly noticed that the model gives better performance by employing SRC path and FA block, as shown in Fig. 5 (b) and (c).

E. MODEL ANALYSIS

There are two hyper parameters in SRC path: the number of layers L and the number k, as shown in the red box of Fig. 2. We conducted experiments on ModelNet40 to choose appropriate values for L and k. Table 5 summarizes the comparison results of different values. As can be noticed from Table 5,

TABLE 5. The comparisons of different L and k on ModelNet40 dataset.

	L	Parameters	Accuracy
$k = 4$	2	3.43M	0.939
	3	3.14M	0.943
	4	3.08M	0.942
$k = 8$	2	2.68M	0.937
	3	2.42M	0.943
	4	2.40M	0.941

increasing the number of paths from 2 to 4 can reduce the model parameters. With three layers, our model achieves the best result. Note that $k = 4$ does not bring much improvement compared with $k = 8$, as shown in Table 5. With $k = 8$, our model has less parameters due to the narrow design of each layer. Thus, we choose $L = 3$ and $k = 8$ as our default structure in this work to balance the complexity and performance. It achieves 0.943 for accuracy with only 2.42M parameters.

F. ROBUSTNESS TO SAMPLING DENSITY

In order to verify the robustness of our model in sampling density, we have conducted experiments on ModelNet40 dataset. Each model in the dataset was randomly dropped some points to generate point clouds with different densities. Therefore, for each model, we used 1024, 512, 256, and 128 points, respectively. Fig.6 shows a comparison of the robustness of our model and other methods. We found that when the density decreases, our model and PointNet++ [7] perform better, the accuracy of SONet [22] takes a sharp decline. We found that when the density decreases, the performance of our model and PointNet ++ are better, and the accuracy of SONet drops sharply. When the points are reduced from 1024 to 512 and 256, the accuracy of our model

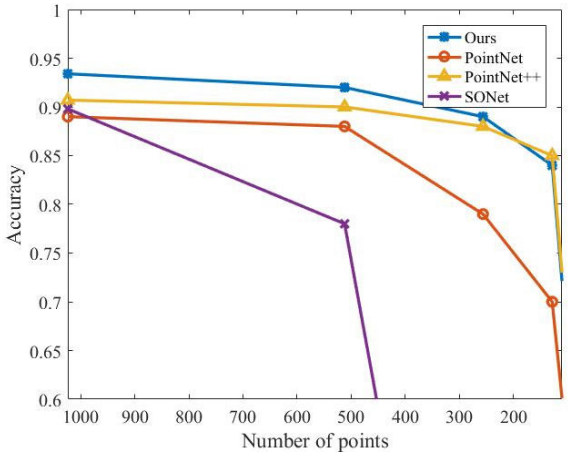

FIGURE 6. Robustness to sampling sparser points.

TABLE 6. Model complexity comparisons of different models on ModelNet40 dataset.

Methods	Parameters	FLOPs/sample	Accuracy
PointNet[2]	3.48M	440M	0.892
PointNet++[7]	1.48M	1684M	0.907
DGCNN[16]	1.84M	2767M	0.922
SpiderCNN[21]	5.84M	-	0.924
DensePoint[24]	0.67M	651M	0.932
Ours	2.42M	387M	0.943

drops by no more than 3% and 6%, respectively. The good robustness of our method to different point densities is due to two factors. First, the dense relations in SRC path has strong feature representation ability, even with spares points. Second, our fusion way can be learned from point coordinates by FA block. The weights of features can be adjusted in different regions.

G. COMPLEXITY ANALYSIS

We also performed experiments for complexity analysis. The experiments were conducted on NVIDIA GeForce GTX 1080Ti GPU. Table 6 displays the model complexity and accuracy for different methods. Following [2], we employed floating point operations/ sample (FLOPs/sample) to measure the computational cost. From Table 6, it is observed that our method can handle point clouds more efficiently. Since we do not need to explicitly select the centroid and its neighbors, our model is more efficient than other hierarchical models. We explored two strategies to reduce network complexity. First, we used bottleneck design in SRC path to reduce the parameters. Second, we approximate high dimensional point correlations with a series of densely connected layers and reduce the number of nodes in the middle layers. Thus, our FLOPs/sample are much less than those of the state-of-the-art methods. Besides that, it requires only 2.42M parameters and still get great performance. The lightweight model implies that the proposed method has low spatial and computational complexity.

V. CONCLUSION

In this work, we have presented a novel method based on dense spatial awareness point correlations for point cloud processing. This allows us generating high dimensional feature maps without losing information, thus enabling our model to learn more representative and detailed features of point cloud. We also proposed an adaptive way to combine the different scale features. Then we demonstrated the great performance of our network on both segmentation task and classification task. The proposed network is simple but effective and can be used as a basic feature extraction module for other applications. In future work, we would like to extract more different scale features to better describe point clouds.

REFERENCES

[1] C. Chen, H. Huang, C. Chen, Z. Zheng, and H. Cheng, "Multi-scale guided mask refinement for coarse-to-fine RGB-D perception," *IEEE Signal Process. Lett.*, vol. 26, no. 2, pp. 217–221, Feb. 2019.
[2] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.
[3] R. Fan, X. Ai, and N. Dahnoun, "Road surface 3D reconstruction based on dense subpixel disparity map estimation," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 3025–3035, Jun. 2018.
[4] Q. Lu, M. Xiao, Y. Lu, X. Yuan, and Y. Yu, "Attention-based dense point cloud reconstruction from a single image," *IEEE Access*, vol. 7, pp. 137420–137431, 2019.
[5] G. Wang, J. Wu, R. He, and S. Yang, "A point cloud-based robust road curb detection and tracking method," *IEEE Access*, vol. 7, pp. 24611–24625, 2019.
[6] C. Wang, D. Xu, Y. Zhu, R. Martin-Martin, C. Lu, L. Fei-Fei, and S. Savarese, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3338–3347.
[7] 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. 5104–5114.
[8] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9613–9622.
[9] L.-Z. Chen, X.-Y. Li, D.-P. Fan, M.-M. Cheng, K. Wang, and S.-P. Lu, "LSANet: Feature learning on point sets by local spatial aware layer," 2019, *arXiv:1905.05442*. [Online]. Available: https://arxiv.org/abs/1905.05442
[10] F. Engelmann, T. Kontogianni, J. Schult, and B. Leibe, "Know what your neighbors do: 3D semantic segmentation of point clouds," in *Proc. ECCVW*, 2018, pp. 395–409.
[11] X. Ye, J. Li, H. Huang, L. Du, and X. Zhang, "3D recurrent neural networks with context fusion for point cloud semantic segmentation," in *Proc. 15th Eur. Conf.*, Munich, Germany, Sep. 2018, pp. 415–430.
[12] M. Jiang, Y. Wu, T. Zhao, Z. Zhao, and C. Lu, "PointSIFT: A SIFT-like network module for 3D point cloud semantic segmentation," 2018, *arXiv:1807.00652*. [Online]. Available: https://arxiv.org/abs/1807.00652
[13] Y. Li, R. Bu, M. Sun, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Proc. Conf. Workshop Neural Inf. Process. Syst.*, 2018, pp. 828–838.
[14] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and deformable convolution for point clouds," 2019, *arXiv:1904.08889*. [Online]. Available: http://arxiv.org/abs/1904.08889
[15] Y. Shen, C. Feng, Y. Yang, and D. Tian, "Mining point cloud local structures by kernel correlation and graph pooling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4548–4557.
[16] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Nov. 2019.
[17] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, "PointWeb: Enhancing local neighborhood features for point cloud processing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5565–5573.

[18] R. Huang, Y. Xu, D. Hong, W. Yao, P. Ghamisi, and U. Stilla, "Deep point embedding for urban classification using ALS point clouds: A new perspective from local to global," *ISPRS J. Photogramm. Remote Sens.*, vol. 163, pp. 62–81, May 2020.
[19] W. Li, F.-D. Wang, and G.-S. Xia, "A geometry-attentional network for ALS point cloud classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 164, pp. 26–40, Jun. 2020.
[20] 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.
[21] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "SpiderCNN: Deep learning on point sets with parameterized convolutional filters," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 90–105.
[22] J. Li, B. M. Chen, and G. H. Lee, "SO-net: Self-organizing network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9397–9406.
[23] 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.
[24] Y. Liu, B. Fan, G. Meng, J. Lu, S. Xiang, and C. Pan, "DensePoint: Learning densely contextual representation for efficient point cloud processing," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5239–5248.
[25] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6620–6629.
[26] S. Kumawat and S. Raman, "LP-3DCNN: Unveiling local phase in 3D convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4898–4907.
[27] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas, "A scalable active framework for region annotation in 3D shape collections," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–12, Nov. 2016.
[28] X. Yao, J. Guo, J. Hu, and Q. Cao, "Using deep learning in semantic classification for point cloud data," *IEEE Access*, vol. 7, pp. 37121–37130, 2019.

DAN WANG received the master's degree in mechanical engineering from Xiamen University, Xiamen. She is currently pursuing the Ph.D. degree with the South China University of Technology, Guangzhou, Guangdong, China. Her research interests include computer vision and developing automated control systems based on image understanding and researching the machine vision-based mechanical automated processing and inspection systems.

GUOQING HU received the M.S. degree from Northwestern Polytechnical University, China, and the Ph.D. degree from Sichuan University, China.

He was a Professor with Xiamen University, China. He was an Advanced Visiting Scholar with the Chinese University of Hong Kong and the University of Nottingham. He is currently a Professor and the Ph.D. Student Supervisor with the School of Mechanical and Automotive Engineering, South China University of Technology, China. He was completed and participated more than 90 projects including the National 863 project, the National Natural Science Foundation project, the national major projects, international cooperation projects, provincial key projects, and the province fund cooperation projects. He was published 248 articles, 22 patents, and two textbooks. His research interests include amphibious flying machine, intelligent robot, industrial image processing, automation and industrial robot, electromechanical integration, and advanced sensor technology.

CHENGZHI LYU (Graduate Student Member, IEEE) received the B.E. degree in electromechanical engineering from the Hubei University of Technology, Wuhan, Hubei, in 2013. He is currently pursuing the Ph.D. degree with the South China University of Technology, Guangzhou, Guangdong, China. His research interests include artificial intelligent algorithms and machine automation.