SK-Net: Deep Learning on Point Cloud via End-to-End Discovery of Spatial Keypoints

Weikun Wu,[1*] **Yan Zhang,**[1,2*] **David Wang,**[3] **Yunqi Lei**[1†]
[1]Computer Science Department, Xiamen University, China
[2]School of Mathematics Science, Guizhou Normal University, China
[3]Department of Electrical and Computer Engineering, The Ohio State University, USA
wuweikun@stu.xmu.edu.cn, zy@gznu.edu.cn, Wang.8552@buckeyemail.osu.edu, yqlei@xmu.edu.cn

Abstract

Since the PointNet was proposed, deep learning on point cloud has been the concentration of intense 3D research. However, existing point-based methods usually are not adequate to extract the local features and the spatial pattern of a point cloud for further shape understanding. This paper presents an end-to-end framework, *SK-Net*, to jointly optimize the inference of spatial keypoint with the learning of feature representation of a point cloud for a specific point cloud task. One key process of SK-Net is the generation of spatial keypoints (Skeypoints). It is jointly conducted by two proposed regulating losses and a task objective function without knowledge of Skeypoint location annotations and proposals. Specifically, our Skeypoints are not sensitive to the location consistency but are acutely aware of shape. Another key process of SK-Net is the extraction of the local structure of Skeypoints (detail feature) and the local spatial pattern of normalized Skeypoints (pattern feature). This process generates a comprehensive representation, pattern-detail (PD) feature, which comprises the local detail information of a point cloud and reveals its spatial pattern through the part district reconstruction on normalized Skeypoints. Consequently, our network is prompted to effectively understand the correlation between different regions of a point cloud and integrate contextual information of the point cloud. In point cloud tasks, such as classification and segmentation, our proposed method performs better than or comparable with the state-of-the-art approaches. We also present an ablation study to demonstrate the advantages of SK-Net.

1 Introduction

Rapid development in stereo sensing technology has made 3D data ubiquitous. The naive structure of most 3D data obtained by the stereo sensor is a point cloud. It makes those methods directly consuming points the most straightforward approach to handle the 3D tasks. Most of the existing point-based methods work on the following aspects: (1) Utilizing the multi-layer perceptron (MLP) to extract the point features and using the symmetry function to guarantee the permutation invariance. (2) Capturing the local structure through explicit or other local feature extraction approaches, and/or modeling the spatial pattern of a point cloud by discovering a set of keypoint-like points. (3) Aggregating features to deal with a specific point cloud task.

Despite being a pioneer in this area, PointNet(Qi et al. 2017a) only extracts the point features to acquire the global representation of a point cloud. For boosting point-based performance, the study advancing on above (2) has recently fostered the proposal of many techniques. One of these proposals is downsampling a point cloud to choose a set of spatial locations of a point cloud as local feature extraction regions(Qi et al. 2017b; Jiang et al. 2018; Liu et al. 2019a). However, in this method, those locations are mostly obtained by artificial definition, e.g., FPS algorithm(Eldar et al. 1997), and do not reckon with the spatial pattern and the correlation between different regions of a point cloud. Similarly, (Xie et al. 2018; Hua, Tran, and Yeung 2018) achieve pointwise local feature extraction by corresponding each point to a region with respect to modeling the local structure of a point cloud. A special way is that (Le and Duan 2018; Li et al. 2018) combine the regular grids of 3D space with points, leveraging spatially-local correlation of regular grids to represent local information. However, performing pointwise local feature extraction and incorporating regular grids often require high computational costs, which is infeasible with large point injections and higher grid resolutions. Furthermore, SO-Net(Li, Chen, and Hee Lee 2018) introduces the Self-Organizing Map (SOM)(Kakuda et al. 1998) to produce a set of keypoint-like points for modeling the spatial pattern of a point cloud. Even though SO-Net takes the regional correlation of a point cloud into account, it trains SOM separately. This leaves the spatial modeling of SOM and a specific point cloud task disconnected.

To address these issues, this paper explores a method that can benefit from an end-to-end framework while the inference of Skeypoints is jointly optimized with the learning of the local details and the spatial pattern of a point cloud. By this means, those end-to-end learned Skeypoints are in strong connection with the local feature extraction and the spatial modeling of a point cloud. It is worth mentioning that related approaches of end-to-end learning of 3D key-

*Contributed equally.
†Corresponding author.

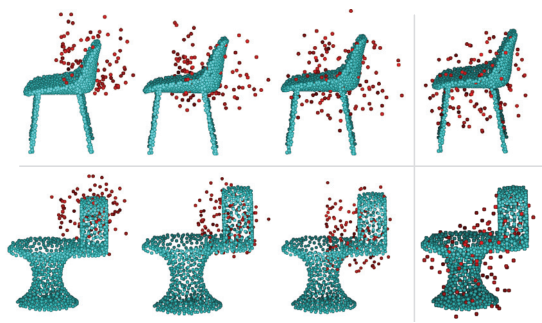


Figure 1: The generation of Skeypoints and the preliminary spatial exploration of the point cloud. From left to right, the figures show the distributions of generated Skeypoints (red balls) with training epochs increasing; the rightmost figures show the resulting Skeypoints.

points usually require the keypoint location proposals(Georgakis et al. 2018; 2019) and expect the location consistency of 3D keypoints. Besides, most of the existing methods handle the tasks of 3D matching or 3D pose estimation. But those methods are not extended to the point cloud recognition tasks(Georgakis et al. 2018; Suwajanakorn et al. 2018; Zhou et al. 2018; Georgakis et al. 2019). Our work focuses on point cloud recognition tasks. Before producing the PD feature, all local features perform max-pooling to ensure that the local spatial pattern is of permutation invariance and fine shape-description. Thus the generated Skeypoints are not sensitive to location consistency. With further training under the adjustment of two regulating losses, these Skeypoints gradually spread to the entire space of a point cloud and distribute around the discriminative regions of the point cloud. The generation of Skeypoints and the preliminary spatial exploration of the point cloud is shown in Figure 1. The **key contributions** of this paper are as follows:

- We propose a novel network named *SK-Net*. It is an end-to-end framework that jointly optimizes the inference of spatial keypoints (*Skeypoints*) and the learning of feature representation in the process of solving a specific point cloud task, e.g., classification and segmentation. The generated Skeypoints do not require location consistency but are acutely aware of shape. It benefits the local detail extraction and the spatial modeling of a point cloud.
- We design a pattern and detail extraction module (*PDE module*) where the key detail features and pattern features are extracted and then aggregated to form the *pattern-detail (PD) feature*. It promotes correlation between different regions of a point cloud and integrates the contextual information of the point cloud.
- We propose two *regulating losses* to conduct the generation of Skeypoints without any location annotations and proposals. These two regulating losses are mutually reinforcing and neither of them can be omitted.
- We conduct extensive experiments to evaluate the performance of our method, which is better than or comparable with the state-of-the-art approaches. In addition, we present an ablation test to demonstrate the advantages of the SK-Net.

2 Related Work

In this section, we briefly review existing works of various regional feature extraction on point cloud and end-to-end learning of 3D keypoints.

2.1 Extraction of regional feature on point cloud

PointNet++(Qi et al. 2017b) is the earliest proposed method that addresses the problem of local information extraction. In practice, this method partitions the set of points into overlapping local regions corresponding to the spatial locations chosen by the FPS algorithm through the distance metric of the underlying space. It enables the learning of local features of a local region with increasing contextual scales. Besides, (Jiang et al. 2018; Liu et al. 2019a) also perform the similar downsampling of a point cloud to obtain the spatial locations regarding local feature extraction. PointSIFT(Jiang et al. 2018) extends the traditional feature SIFT(Lowe 2004) to develop a local region sampling approach for capturing the local information of different orientations around a spatial location. It is similar to (Xie et al. 2018) expanding from ShapeContext(Belongie, Malik, and Puzicha 2001). Point2Sequence(Liu et al. 2019a) explores the attention-based aggregation of multi-scale areas of each local region to achieve local feature extraction. And that, (Hua, Tran, and Yeung 2018) defines a pointwise convolution operator to query each point's neighbors and bin them into kernel cells for extracting pointwise local features. (Shen et al. 2018) proposes a point-set kernel operator as a set of learnable 3D points that jointly respond to a set of neighboring data points according to their geometric affinities. Other techniques also exploit extension operators to extract local structure. For example, (Klokov and Lempitsky 2017; Atzmon, Maron, and Lipman 2018) map the point cloud to different representation space, and (Simonovsky and Komodakis 2017; Wu et al. 2018) dynamically construct a graph comprising edge features for each specific local sampling of a point cloud. In particular, (Le and Duan 2018; Li et al. 2018) combine the regular grids of 3D space with points and leverage the spatially-local correlation of regular grids to achieve local information capture. Among the latest methods, the study of local geometric relations plays an important role. (Lan et al. 2019) models geometric local structure through decomposing the edge features along three orthogonal directions. (Liu et al. 2019b) learns the geometric topology constraint among points to acquire an inductive local representation. By contrast, our method not only performs the relevant regional feature extraction in the PDE module, but also learns a set of Skeypoints corresponding to geometrically and semantically meaningful regions of the point cloud. In this way, our network can effectively model the spatial pattern of a point cloud and promotes the correlation between different regions of the point cloud.

Most related to our approach is the recent work, SO-Net(Li, Chen, and Hee Lee 2018). This work suggests utilizing the Self-Organizing Map (SOM) to build the spatial pattern of a point cloud. However, SO-Net is not proved to be generic to large-scale semantic segmentation. This is intuitively because its network is not end-to-end and the SOM is as an early stage with respect to the overall pipeline, so the SOM does not establish a connection with a specific point cloud task. In contrast, our method is capable of extending to large-scale point cloud analysis and achieving the end-to-end joint learning of Skeypoints and feature representation towards a specific point cloud task.

2.2 End-to-end learning of 3D keypoints

Approaches for end-to-end learning 3D keypoints have been investigated. Most of them focus on handling 3D matching or 3D pose estimation tasks. They do not extend to point cloud recognition tasks, especially classification and segmentation(Georgakis et al. 2018; Suwajanakorn et al. 2018; Zhou et al. 2018; Georgakis et al. 2019; Feng et al. 2019; Lu et al. 2019). Some of these methods usually require keypoint proposals. For example, (Georgakis et al. 2018) proposes that using a Region Proposal Network (RPN) to obtain several Region of Interests (ROIs) then determines the keypoint locations by the centroids of those ROIs. Similarly, (Georgakis et al. 2019) casts a keypoint proposal network (KPN) comprising two convolutional layers to acquire the keypoint confidence score in evaluating whether the candidate location is a keypoint or not. In addition, KeypointNet(Suwajanakorn et al. 2018) presents an end-to-end geometric reasoning framework to learn an ordered set of 3D keypoints, whose discovery is guided by the carefully constructed consistency and relative pose objective functions. These approaches tend to depend on the location consistency of keypoints of different views(Georgakis et al. 2018; Suwajanakorn et al. 2018; Zhou et al. 2018; Georgakis et al. 2019). Moreover, (Feng et al. 2019) presents an end-to-end deep network architecture to jointly learn the descriptors for 2D and 3D keypoints from an image and point cloud, establishing 2D-3D correspondence. (Lu et al. 2019) end-to-end trains its keypoint detector to enable the system to avoid the inference of dynamic objects and leverage the help of sufficiently salient features on stationary objects. By contrast, our SK-Net focuses on point cloud classification and segmentation. In addition, our Skeypoints not only are inferred without any location annotations or proposals but also are insensitive to location consistency. This greatly benefits the spatial region relation exploration of a point cloud.

3 End-to-end optimization of Skeypoints and feature representation

In this section, we present the end-to-end framework of our SK-Net. First, the architecture of SK-Net is introduced. Then its important process, PDE module, is discussed in detail. Finally, the significant properties of Skeypoints are analyzed.

3.1 SK-Net architecture

The overall architecture of SK-Net is shown in Figure 2. The backbone of our network consists of the following three

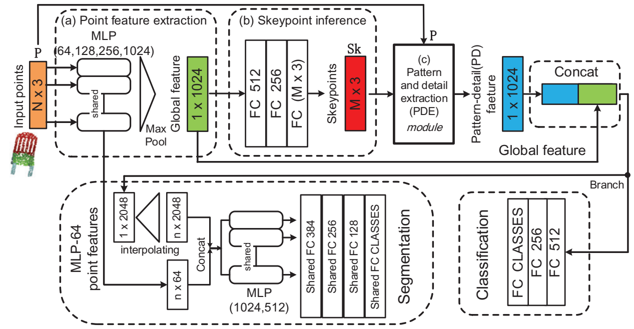


Figure 2: The overall architecture of the SK-Net.

components: (a) point feature extraction, (b) skeypoint inference and (c) pattern and detail extraction.

(a) Point feature extraction. Given an input point cloud P with size of N, i.e. $P = \{p_i \subseteq \mathbb{R}^3, i = 0, 1, \ldots, N-1\}$, where each point p_i is a vector of its (x, y, z) coordinates. P goes through a series of multi-layer perceptrons (MLPs) to extract its point features. Next, we use the symmetry function of max-pooling to acquire the global feature.

(b) Skeypoint inference. M Skeypoints, $Sk = \{sk_j \subseteq \mathbb{R}^3, j = 0, 1, \ldots, M-1\}$, are regressed by stacking three fully connected layers on the top of the global feature. sk_j represents the j-th Skeypoint. Note that the MLP-64 point features are used to handle point cloud segmentation tasks. Meanwhile, the interpolating operation proposed by (Qi et al. 2017b) is utilized.

(c) Pattern and detail extraction. The generated Skeypoints are forwarded into the PDE module to get PD features. More details are presented in the next subsection.

Finally, our network combines the PD feature aggregated in the PDE module with the global feature obtained in (a), for a specific point cloud task e.g., classification and segmentation.

3.2 Pattern and detail extraction module

The PDE module is shown in Figure 3. The PDE module consumes M generated Skeypoints and N input points. It contains three parts: (1) extraction of local details, (2) capture of local spatial patterns, (3) integration of features.

Extraction of local details In this part, we apply a simple kNN search to sample the local region. The number of local regions to extract local information from is identical with the number of generated Skeypoints. Consequently, M local neighborhoods captured by the Skeypoints are grouped to form a $M \times H \times 3$ tensor named local detail tensor. We use the (x, y, z) coordinate as our tensor's channels. The captured points are the points live in the local neighborhood captured by a generated Skeypoint and are denoted by $Cps_j, j = 0, 1, \ldots, M-1$. H is the number of captured points of each generated Skeypoint. Each captured point of a local neighborhood is denoted by $cp_h, h = 0, 1, \ldots, H-1$. The M set of captured points can be formulated by:

$$Cps_j = kNN(cp_h | sk_j, h = 0, 1, \ldots, H-1) \quad (1)$$

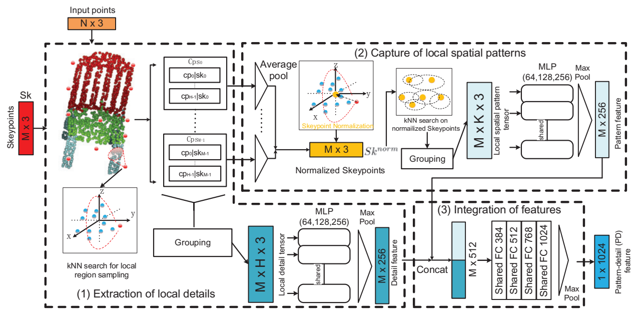


Figure 3: The pivotal PDE module of our SK-Net. The red balls around the point cloud object in (1) are the Skeypoints, and the orange balls in (2) represent the normalized Skeypoints.

Subsequently, PDE module applies a series of MLPs on the local detail tensor to extract the $M \times 256$ detail feature. It is worth mentioning that there are other ways to sample local regions. We will present relevant experiments in the next section to illustrate the superiority of using the kNN search in our scheme.

Capture of local spatial patterns To effectively capture the local spatial patterns of a point cloud, we define a concept named normalized Skeypoint. We average the location coordinates of points which live in the local neighborhood captured by each generated Skeypoint for acquiring the M normalized Skeypoints, $Sk^{norm} = \{sk_j^{norm} \subseteq \mathbb{R}^3, j = 0, 1, \ldots, M-1\}$, where sk_j^{norm} is defined by:

$$sk_j^{norm} = average(Cps_j) \tag{2}$$

The normalized Skeypoints are distributed into the discriminative regions of a point cloud surface, as illustrated in Figure 4. It is beneficial to perform the entire spatial modeling of the point cloud. Next, we use kNN search on normalized Skeypoints themselves to capture the local spatial patterns of the point cloud. After that, we group the search results to get the $M \times K \times 3$ local spatial pattern tensor, where K is the number of normalized Skeypoints in each local spatial region. The $M \times 256$ pattern feature is extracted by the same approach as obtaining the detail feature.

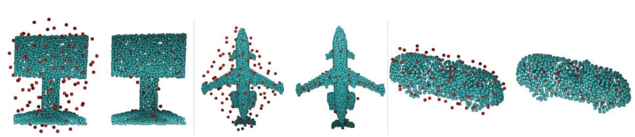


Figure 4: Each pair contains a point cloud (the left) generated by Skeypoints and corresponding one generated by normalized Skeypoints.

Integration of features In this part, the PDE module concatenates the pattern feature with the detail feature to form $M \times 512$ intermediate result. This result is further aggregated through a stack of shared FC layers to acquire the pattern-detail (PD) feature. The PD feature builds the connections between different regions of a point cloud and integrates the contextual information of the point cloud.

3.3 Properties of Skeypoints

The fact that a point set is permutation invariant requires certain symmetrizations in the net computation. It is the same in our Skeypoint inference and relevant feature formation in order to perform the permutation invariance. Moreover, for our scheme and point cloud tasks, it requires that our Skeypoints are distinct from each other (distinctiveness), adapted to the spatial modeling of a point cloud (adaptation) and close to geometrically and semantically meaningful regions (closeness). However, due to the lack of keypoint annotations, the Skeypoints might either tend to be extremely close and even collapse to the same 3D location, or may stray away from the point cloud object. Because of these issues, we propose two regulating losses to conduct the generation of Skeypoints. They guarantee the distinctiveness, adaptation, and closeness of Skeypoints.

Permutation invariance We employ a modified PointNet framework to acquire the global feature of a point cloud. It is basis of the inferring of Skeypoints. After that, the activated outputs of the Skeypoint inference component are directly reshaped to obtain Skeypoints. Each coordinate of a Skeypoint is regressed independently. These procedures make Skeypoint generation satisfy permutation invariance. Note that the nonlinear activation function PReLU(He et al. 2015) is adopted to guarantee that each layer activation matches the location distribution of the point cloud and the regressive consequence is well-adapted and robust. Moreover, all local features use max-pooling to ensure their permutation invariance before producing the PD feature.

Distinctiveness and adaptation In order to achieve these characteristics, we take the separation distance between Skeypoints as a hyperparameter δ and propose the separation loss L_{sep}. Hyperparameter δ provides a prior knowledge of the location distribution of the point cloud. It makes the generated Skeypoints adapted to the spatial modeling of a point cloud and boosts the convergence of our network. The separation loss L_{sep} prompts the M Skeypoints to be distinct from each other and penalizes two Skeypoints if their distance is less than δ. The L_{sep} is formulated by:

$$L_{sep} = \frac{1}{M^2} \sum_{i=1}^{M} \sum_{i \neq j}^{M} max(0, \delta - ||sk_i - sk_j||^2) \tag{3}$$

Closeness In order to make Skeypoints correspond to geometrically and semantically meaningful regions of a point cloud, the network should encourage Skeypoints to close to the discriminative regions of the point cloud surface. A regulating loss named close loss L_{close} is proposed to penalize a Skeypoint and its captured points live in the local structure of this Skeypoint if they are farther than the value of hyperparameter θ in 3D Euclidean space. The L_{close} is represented by:

$$L_{close} = \frac{1}{MH} \sum_{i=1}^{M} \sum_{h=1}^{H} max(0, ||sk_i - cp_h||^2 - \theta) \quad (4)$$

These two loss terms allow our network to discover satisfactory Skeypoints as far as possible. Besides, they prompt our network to extract more compact local features with respect to the regions corresponding to the learned Skeypoints and to capture the effective spatial pattern of a point cloud by utilizing the normalized Skeypoints.

4 Experiments

Datasets We validate on four datasets to demonstrate the effectiveness of our SK-Net. Object classification on ModelNet(Wu et al. 2015) is evaluated by accuracy, part segmentation on ShapeNetPart(Yi et al. 2016) is evaluated by mean Intersection over Union (mIoU) on points and semantic scene labeling on ScanNet(Dai et al. 2017) is evaluated by per-point accuracy.

- ModelNet. This includes two datasets which respectively contain CAD models of 10 and 40 categories. ModelNet10 consists of $4,899$ object instances which are split into $2,468$ training samples and 909 testing samples. ModelNet40 consists of $12,311$ object instances among which $9,843$ objects belong to the training set and the other $3,991$ samples for testing.
- ShapeNetPart. $16,881$ shapes from 16 categories, labeled with 50 parts in total. A large proportion of shape categories are annotated with two to five parts.
- ScanNet. $1,200$ shapes from 50 categories. Each category contains 24 shapes which are mostly organic ones with various poses such as horses, cats, etc.

Implementation details Sk-Net is implemented by Tensorflow in CUDA. We run our model on GeForce GTX Titan X for training. In general, we set the number of the Skeypoints to 192, and K to 16, H to 32 in the PDE module. In most experiments, the hyperparameters δ , θ of our two regulating losses are both 0.05, and the weights of all loss terms are identical. In addition, we use Adam(Kingma and Ba 2014) optimizer with an initial learning rate of 0.001 and the learning rate is decreased by staircase exponential decay. Batch size is 16. All layers are implemented with batch normalization. PReLU activation is applied to the layers of the point feature extraction and Skeypoint inference components, while ReLU activation is applied to every layer of the subsequent network.

4.1 Classification on ModelNet

For a fair comparison, the ModelNet10/40 datasets for our experiments are preprocessed by (Qi et al. 2017b). By default, 1024 input points are used. Moreover, we attempt to use more points and surface normals as additional features for improving performance. Table 1 shows the classification accuracy of state-of-the-art methods on point cloud representation. In ModelNet10, our network outperforms state-of-the-art methods by 95.8% with 1024 points and achieves a better accuracy of 96.2% with 5000 points and normal information. In ModelNet40, the network performs a comparable result of 92.7% through training with 5000 points and surface normal vectors as additional features. Although SO-Net presents the best result in ModelNet40, its network is trained separately and not generic to large-scale semantic segmentation, while our proposed model is trained end-to-end and does extend to large-scale point cloud analysis.

4.2 Part segmentation on ShapeNetPart

Part segmentation is more challenging than object classification and can be formulated as a per-point classification problem. We use the segmentation network to do this. Fairly, ShapeNetPart dataset is prepared by (Qi et al. 2017b) and we also follow the metric protocol from (Qi et al. 2017b). The results of the related approaches are illustrated in Table 2. Although the best mIoU of all shapes is performed by (Liu et al. 2019a), ours outperforms state-of-the-art methods in six categories and achieves comparable results in the remaining categories. We provide ShapeNetPart segmentation visualization in the supplementary material.

4.3 Semantic scene labeling on ScanNet

We conduct experiments on ScanNet's Semantic Scene Labeling task to validate that our method is suitable for large-scale point cloud analysis. We use our segmentation network to do this. To perform this task, we set the point cloud to be normalized into $[-1, 1]$ because the L_{sep} is sensitive to the distribution of point clouds. All experiments use 8192 points. We compare the per-point accuracy and the number of the local region features in each method, as shown in Table 3. Our approach is better than PointNet and slightly behind PointNet++. However, the number of our local region features is only 192 which is much smaller than that in PointNet++. These results show that our method is extendable to large-scale semantic segmentation and the local region features obtained by our method are compact and efficient.

4.4 Ablation study

In this subsection, we further conduct several ablation experiments to investigate various setup variations and to demonstrate the advantages of SK-Net.

Effects of features extracted in PDE module We present an ablation test on ModelNet10 classification to show the effect of the features extracted by PDE module i.e. detail features and pattern features, as shown in Figure 3. The accuracy of the experiments as follows: 93.9% (using only detail features), 93.3% (using only pattern features), and 95.8% (using their concatenated result). It demonstrates that our proposed PD feature promotes correlation between different regions of a point cloud and is more effective in modeling the whole spatial distribution of the point cloud.

Table 1: Object classification accuracy (%) on ModelNet.

Method	Representation	Input	ModelNet10		ModelNet40	
			Class	Instance	Class	Instance
PointNet(Qi et al. 2017a)	points	1024×3	–	–	86.2	89.2
PointNet++(Qi et al. 2017b)	points+normal	5000×6	–	–	–	91.9
Kd-Net(Klokov and Lempitsky 2017)	points	$2^{15} \times 3$	93.5	94.0	88.5	91.8
OctNet(Riegler, Osman Ulusoy, and Geiger 2017)	points	128^3	90.1	90.9	83.8	86.5
SCN(Xie et al. 2018)	points	1024×3	–	–	87.6	90.0
ECC(Simonovsky and Komodakis 2017)	points	1000×3	90.0	90.8	83.2	87.4
KC-Net(Klokov and Lempitsky 2017)	points	2048×3	–	94.4	–	91.0
DGCNN(Wu et al. 2018)	points	1024×3	–	–	90.2	92.2
PointGrid(Le and Duan 2018)	points	1024×3	–	–	88.9	92.0
PointCNN(Li et al. 2018)	points	1024×3	–	–	–	91.7
PCNN(Atzmon, Maron, and Lipman 2018)	points	1024×3	–	94.9	–	92.3
Point2Sequence(Liu et al. 2019a)	points	1024×3	95.1	95.3	90.4	92.6
SO-Net(Li, Chen, and Hee Lee 2018)	points+normal	5000×6	95.5	95.7	**90.8**	**93.4**
Ours	points	1024×3	95.6	95.8	89.0	91.5
Ours	points+normal	5000×6	**96.2**	**96.2**	90.3	92.7

Table 2: Object part segmentation results on ShapeNetPart dataset.

	Intersection over Union (IoU)																
	mean	air	bag	cap	car	chair	ear.	gui.	knife	lamp	lap.	motor	mug	pistol	rocket	skate	table
PointNet(Qi et al. 2017a)	83.7	83.4	78.7	82.5	74.9	89.6	73.0	**91.5**	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
PointNet++(Qi et al. 2017b)	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	**71.6**	94.1	81.3	58.7	**76.4**	82.6
Kd-Net(Klokov and Lempitsky 2017)	82.3	80.1	74.6	74.3	70.3	88.6	73.5	90.2	87.2	81.0	94.9	57.4	86.7	78.1	51.8	69.9	80.3
KC-Net(Shen et al. 2018)	84.7	82.8	81.5	86.4	77.6	90.3	76.8	91.0	87.2	**84.5**	95.5	69.2	94.4	81.6	60.1	75.2	81.3
DGCNN (Wu et al. 2018)	85.1	**84.2**	**83.7**	84.4	77.1	**90.9**	78.5	**91.5**	87.3	82.9	96.0	67.8	93.3	**82.6**	59.7	75.5	82.0
point2sequence (Liu et al. 2019a)	**85.2**	82.6	81.8	87.5	77.3	90.8	77.1	91.1	86.9	83.9	**95.7**	70.8	**94.6**	79.3	58.1	75.2	82.8
SO-Net (Li, Chen, and Hee Lee 2018)	84.9	82.8	77.8	**88.0**	77.3	90.6	73.5	90.7	83.9	82.8	94.8	69.1	94.2	80.9	53.1	72.9	**83.0**
Ours	85.0	82.9	80.7	87.6	**77.8**	90.5	**79.9**	91.0	**88.1**	84.0	**95.7**	69.9	94.0	81.1	**60.8**	**76.4**	81.9

Table 3: Comparisons of per-point classification on ScanNet.

Method	Local regions	Local selection operator	accuracy
PointNet	–	–	77.7%
PointNet++(ssg)	[1024, 256, 64, 16]	FPS	82.6%
PointNet++(msg)	[512, 128]	FPS	83.2%
Ours	192	End-to-end learning	81.4%

Complementarity of regulating losses Ablation experiments of the losses are presented to validate the effect of our loss functions (classification loss: L_{cls}, separation loss: L_{sep}, close loss: L_{close}). As shown in Table 4, EXP.1 is the baseline and only the classification loss is used. The results show that our method is effective. EXP.2,3,4 show that the two regulating losses proposed by our scheme are mutually reinforcing and neither of them can be omitted. The effect of combining these two losses can refine the spatial model and improve the performance by 1.5%.

Table 4: Ablation test of the losses on ModelNet10 classification.

EXP.	L_{cls}	L_{sep}	L_{close}	accuracy
1	✓	–	–	94.3%
2	✓	✓	–	93.8%
3	✓	–	✓	93.2%
4	✓	✓	✓	**95.8%**

Effects of local region samplings In this part, we experiment using other techniques (ball query and sift query proposed by (Jiang et al. 2018)) to sample local regions and also play with the general search radii: $0.1, 0.2$. For all experiments, the number of sample points is 32 and the number of input points is 1024. As shown in Table 5, the sift query is the least effective method and the ball query is slightly worse than kNN. This shows kNN is the most beneficial to our scheme.

Table 5: Effects of local sample choices on ModelNet10 classification.

kNN	ball query		sift query	
	r=0.1	r=0.2	r=0.1	r=0.2
95.8%	94.3%	94.7%	92.5%	92.7%

Effectiveness of Skeypoints In order to exhibit the effectiveness of our Skeypoints, we eliminate the Skeypoints. And then the network is trained respectively with random dropout, farthest point sampling (FPS) algorithm and the SOM. We use these methods to discover a set of keypoint-like points for building the spatial pattern of a point cloud. Note that the keypoints are jittered slightly to avoid turning into naive downsampling in the random dropout and FPS methods. In addition, we use 11×11 SOM nodes processed by SO-Net(Li, Chen, and Hee Lee 2018) and likewise the number of keypoints is 121 for all comparative experiments. Subsequently, we evaluate the accuracy of different spatial modeling methods on ModelNet10. Visualization examples and the accuracy results are illustrated in Figure 5 and Figure 6. It is interesting that the keypoints respectively selected by the FPS algorithm and learned by SOM, are more well-distributed than our method but have approximately 2-3 percent lower accuracy than ours. Moreover, it is more sensitive to the density perturbation of points with the unlearned random dropout and FPS algorithm. By contrast, our Skeypoints are jointly optimized with the feature learning for a specific point cloud task, promoting the robustness of the variability of points density and performing better performance.

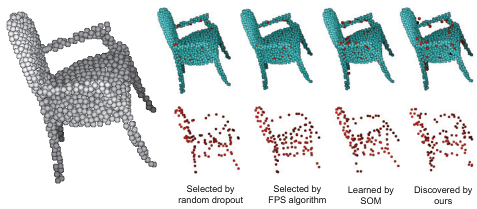


Figure 5: There is a point cloud on the left-most side and the corresponding normalized keypoints(red balls) discovered by different methods on the other side.

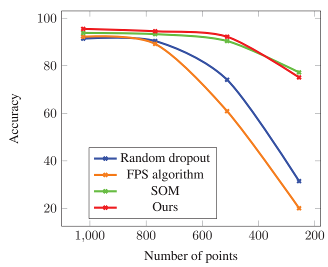
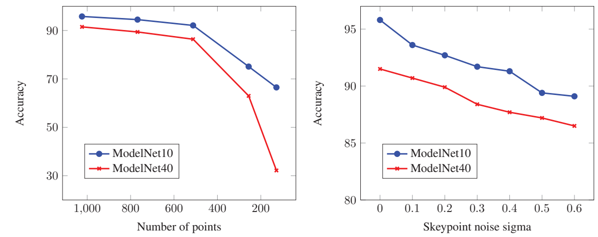


Figure 6: Curve shows the performance of the spatial modeling approaches on various points densities.

Robustness to corruption The network is trained with input points of 1024 and evaluated with point dropout randomly to demonstrate the robustness of the point density variability. In ModelNet10/40, our accuracy drops by 3.4% and 4.5% respectively with 50% points missing (1024 to 512), and remains 75.1% and 63.0% respectively with 256 points, as shown in Figure 7 (a). Moreover, our SK-Net is robust to the noise or corruption of the Skeypoints, as exhibited in Figure 7 (b). When the noise sigma is the maximum of 0.6 in our experiments, the accuracy on ModelNet10/40 is respectively 89.1 and 86.5.

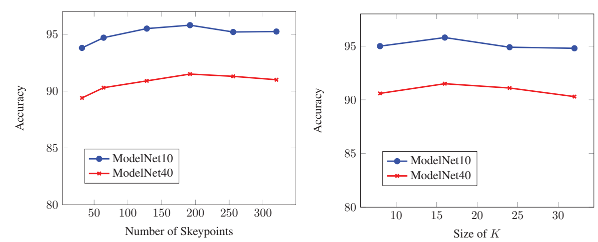
Effects of preferences Our proposed method is more than simply using the end-to-end learned Skeypoints to choose the local regions for extracting the local features, as an existing method(Qi et al. 2017b) does. The Skeypoints also play an important role in the spatial modeling of a point cloud, enhancing the regional associations and prompting our model to get more compact regional features, as illustrated in Table 3. Therefore, the number of generated Skeypoints should be adapted to urge the Skeypoints to embrace the point cloud adequately, and the size K of the normalized Skeypoints of a local spatial region should properly reinforce the region correlation of the point cloud. As shown



(a) (b)

Figure 7: Robustness to the corruption. (a) There is point dropout randomly during testing, and the size H decrease by 8 when the number of input points is less than 512. (b) Gaussian noise $\mathcal{N}(0, \sigma)$ is added to the generated Skeypoints during testing.

in Figure 8 (a), our network gradually performs better accuracy with the Skeypoints increasing from 32 to 192, while the accuracy decreases slightly with 256 Skeypoints. And then the effect of the size K is also shown in Figure 8 (b) on condition that the number of generated Skeypoints remains at 128. The best accuracies are achieved when K is 16.



(a) (b)

Figure 8: Effect of preferences. (a) The effect of the number of Skeypoints on the condition that K is 16. (b) The effect of various K when fixing 128 Skeypoints.

5 Conclusions

In this work, we propose an end-to-end framework named SK-Net to jointly optimize the inference of Skeypoints with the feature learning of a point cloud. These Skeypoints are generated by two complementary regulating losses. Specifically, their generation requires neither location labels and proposals nor location consistency. Furthermore, we design the PDE module to extract and integrate the detail feature and pattern feature, so that local region feature extraction and the spatial modeling of a point cloud can be achieved efficiently. As a result, the correlation between different regions of a point cloud is enhanced, and the learning of the contextual information of the point cloud is promoted. In addition, we conduct experiments to show that for both point cloud classification and segmentation, our method achieves better or similar performance when compared with other existing methods. The advantages of our SK-Net is also demonstrated by several ablation experiments.

Acknowledgement

Yunqi Lei is the corresponding author. This work was supported by the National Natural Science Foundation of China (61671397). We thank all anonymous reviewers for their constructive comments.

References

Atzmon, M.; Maron, H.; and Lipman, Y. 2018. Point convolutional neural networks by extension operators. *arXiv preprint arXiv:1803.10091*.

Belongie, S.; Malik, J.; and Puzicha, J. 2001. Shape context: A new descriptor for shape matching and object recognition. In *Advances in Neural Information Processing Systems*, 831–837.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5828–5839.

Eldar, Y.; Lindenbaum, M.; Porat, M.; and Zeevi, Y. Y. 1997. The farthest point strategy for progressive image sampling. In *IEEE Transactions on Image Processing*, 1305–1315.

Feng, M.; Hu, S.; Ang, M.; and Lee, G. H. 2019. 2D3D-matchnet: Learning to match keypoints across 2D image and 3D point cloud. *arXiv preprint arXiv:1904.09742*.

Georgakis, G.; Karanam, S.; Wu, Z.; Ernst, J.; and Košecká, J. 2018. End-to-end learning of keypoint detector and descriptor for pose invariant 3D matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1965–1973.

Georgakis, G.; Karanam, S.; Wu, Z.; and Košecká, J. 2019. Learning local rgb-to-cad correspondences for object pose estimation. *arXiv preprint arXiv:1811.07249*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

Hua, B.; Tran, M.; and Yeung, S. a. 2018. Pointwise convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 984–993.

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

Kakuda, N.; Miwa, T.; Nagaoka, M.; and Kohonen, T. 1998. The self-organizing map. *Neurocomputing* 21(1):1–6.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klokov, R., and Lempitsky, V. 2017. Escape from cells: Deep kd-networks for the recognition of 3D point cloud models. In *Proceedings of the IEEE International Conference on Computer Vision*, 863–872.

Lan, S.; Yu, R.; Yu, G.; and Davis, L. S. 2019. Modeling local geometric structure of 3D point clouds using geo-cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 998–1008.

Le, T., and Duan, Y. 2018. Pointgrid: A deep network for 3D shape understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9204–9214.

Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; and Chen, B. 2018. Pointcnn: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems*, 820–830.

Li, J.; Chen, B. M.; and Hee Lee, G. 2018. So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9397–9406.

Liu, X.; Han, Z.; Liu, Y.-S.; and Zwicker, M. 2019a. Point2sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 8778–8785.

Liu, Y.; Fan, B.; Xiang, S.; and Pan, C. 2019b. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8895–8904.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, 91–110.

Lu, W.; Wan, G.; Zhou, Y.; Fu, X.; Yuan, P.; and Song, S. 2019. Deepicp: An end-to-end deep neural network for 3D point cloud registration. *arXiv preprint arXiv:1905.04153*.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, 5099–5108.

Riegler, G.; Osman Ulusoy, A.; and Geiger, A. 2017. Octnet: Learning deep 3D representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3577–3586.

Shen, Y.; Feng, C.; Yang, Y.; and Tian, D. 2018. Mining point cloud local structures by kernel correlation and graph pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4548–4557.

Simonovsky, M., and Komodakis, N. 2017. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3693–3702.

Suwajanakorn, S.; Snavely, N.; Tompson, J. J.; and Norouzi, M. 2018. Discovery of latent 3D keypoints via end-to-end geometric reasoning. In *Advances in Neural Information Processing Systems*, 2059–2070.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3D shapenets: A deep representation for volumetric shapes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1912–1920.

Wu, B.; Liu, Y.; Lang, B.; and Huang, L. 2018. Dgcnn: Disordered graph convolutional neural network based on the gaussian mixture model. *Neurocomputing* 321:346–356.

Xie, S.; Liu, S.; Chen, Z.; and Tu, Z. 2018. Attentional shapecontextnet for point cloud recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4606–4615.

Yi, L.; Kim, V. G.; Ceylan, D.; Shen, I.; Yan, M.; Su, H.; Lu, C.; Huang, Q.; Sheffer, A.; Guibas, L.; et al. 2016. A scalable active framework for region annotation in 3D shape collections. *ACM Transactions on Graphics* 35(6):210.

Zhou, X.; Karpur, A.; Gan, C.; Luo, L.; and Huang, Q. 2018. Unsupervised domain adaptation for 3D keypoint estimation via view consistency. In *European Conference on Computer Vision*, 137–153.