D-Net: Learning for distinctive point clouds by self-attentive point searching and learnable feature fusion ☆

Xinhai Liu [a], Zhizhong Han [b], Sanghuk Lee [a], Yan-Pei Cao [c], Yu-Shen Liu [a,*]

[a] *School of Software, Tsinghua University, Beijing, China*
[b] *Department of Computer Science, Wayne State University, USA*
[c] *Y-tech, Kuaishou Technology, China*

ARTICLE INFO

ABSTRACT

Learning and selecting important points on a point cloud is crucial for point cloud understanding in various applications. Most of early methods selected the important points on 3D shapes by analyzing the intrinsic geometric properties of every single shape, which fails to capture the importance of points that distinguishes a shape from objects of other classes, i.e., the distinction of points. To address this problem, we propose D-Net (Distinctive Network) to learn for distinctive point clouds based on a self-attentive point searching and a learnable feature fusion. Specifically, in the self-attentive point searching, we first learn the distinction score for each point to reveal the distinction distribution of the point cloud. After ranking the learned distinction scores, we group a point cloud into a high distinctive point set and a low distinctive one to enrich the fine-grained point cloud structure. To generate a compact feature representation for each distinctive point set, a stacked self-gated convolution is proposed to extract the distinctive features. Finally, we further introduce a learnable feature fusion mechanism to aggregate multiple distinctive features into a global point cloud representation in a channel-wise aggregation manner. The results also show that the learned distinction distribution of a point cloud is highly consistent with objects of the same class and different from objects of other classes. Extensive experiments on public datasets, including ModelNet and ShapeNet part dataset, demonstrate the ability to learn for distinctive point clouds, which helps to achieve the state-of-the-art performance in some shape understanding applications.

© 2023 Elsevier B.V. All rights reserved.

1. Introduction

Point cloud captured by 3D scanners has attracted extensive research interest in various point cloud understanding applications, such as 3D object classification Qi et al. (2017a,b); Wang et al. (2019b); Liu et al. (2019c), detection Shi et al. (2019); Chen et al. (2019); Lang et al. (2019) and segmentation Hu et al. (2020); Yang et al. (2019); Wang et al. (2019a). Unlike regular 2D images and 3D voxels with fixed local spatial distribution, the point cloud is irregular and unordered, consisting of 3D coordinates and some additional attributes, e.g., color, normal, reflectance, etc. To avoid processing unstruc-
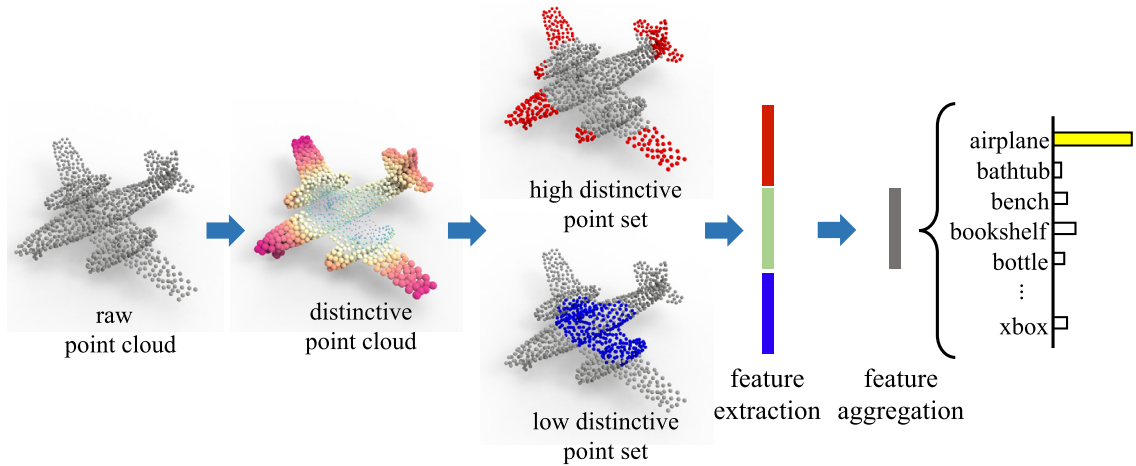

Fig. 1. The learning for distinctive point clouds. Specifically, for a raw point cloud, we first build the distinctive point cloud by learning a distinction score for each point, where the red points with a large radius indicate a high distinction and the blue points with a small radius indicate a low distinction. By ranking the distinction scores, we respectively select the high distinctive point set and low distinctive point set to enrich fine-grained details. Then, we extract the corresponding feature representation of each distinctive point set, as well as the raw point cloud. The extracted features of three point sets are aggregated into a global point cloud representation, which distinguishes a 3D shape from others of different classes.

tured point sets, it is intuitive to first convert point clouds into regular views Chen et al. (2017); Roveri et al. (2018) or voxels Zhou and Tuzel (2018); Meng et al. (2019) and then apply traditional convolutional neural networks to complete the recognition process. However, due to lacking depth information, 2D view-based methods are limited in multiple applications like semantic segmentation, and 3D voxel-based methods require high memory and computational cost.

Many subsequent improvements Li et al. (2018b); Wang et al. (2019b); Liu et al. (2020, 2019c) mainly focus on capturing local or global structures within point clouds and have achieved better performances. They usually treat all points of a point cloud equally. However, the important regional points on point clouds and their correlation are not well measured and detected for learning the global shape representation for distinguishing shapes from others in recognition tasks. In this paper, we found that learning and selecting distinctive points from point clouds are important to distinguish each shape from shapes of other classes. We follow the definition of the distinction of points on a 3D object, as proposed by Shilane and Funkhouser Shilane and Funkhouser (2007), for learning and selecting the important points, where the distinction of every point in an object is defined as how useful it is for distinguishing the object from others of different classes. Most prior methods Lee et al. (2005); Shilane and Funkhouser (2006) have selected the important points on shapes by analyzing the intrinsic geometric properties of every single shape, which lacks to capture the importance of points that distinguish each shape from objects of other types. This common strategy is first to extract the hand-crafted shape features from the local neighborhood and then obtain the distinctiveness of regional points based on the difference between their features to others. To alleviate the limitation of hand-crafted shape features, a few approaches Shu et al. (2018); Nezhadarya et al. (2020); Zheng et al. (2019) have been proposed to learn to detect the points of interest or salient points, which often consider how unique and visible a point is relative to other points within the same object.

In contrast, the distinction of a point considers how common and unique the point is relative to objects of other types in a dataset. In addition, the above-mentioned methods also lack to explore the spatial correlation among multiple distinctive point sets of a point cloud. To address this problem, we propose D-Net (Distinctive Network) to explore both the local structures within points and the spatial correlation among multiple distinctive point sets of a point cloud. As shown in Fig. 1, we first learn the distinction score for each point to reveal the point distinction distribution. By ranking the learned distinction scores, we group each point cloud into a high distinctive point sets and a low distinctive point sets. Then, to extract the feature representation of each distinctive point set, a stacked self-gated convolution is proposed for extracting the distinctive features. Finally, a feature aggregation operation is designed for fusing multiple distinctive point set features. In feature fusion, the pooling operation has been widely adopted by existing works to combine multiple features with deep neural networks, but simply pooling disregards a lot of content information and the spatial correlation among different distinctive point set features, which limits the discriminability of learned global shape representation. To address this issue, we further introduce a learnable feature fusion mechanism to aggregate the distinctive point set features into a global point cloud representation in a channel-wise aggregation manner. Our main contributions are summarized as follows.

- We design a self-attentive point searching module to capturing a distinction score for each point, which provides a primary basis of the ranking of points though the back-propagation of gradients.
- To effectively capture the features of point sets, we propose a self-gated convolution module, which enables the information forgetting in different abstraction layers by introducing the gate structure.

- To aggregate the multiple features of distinctive point sets in a channel-wise way, we introduce a learnable gate fusion strategy to adaptively aggregate multiple distinctive features.

2. Related work

Learning on Point Clouds. Recently, deep networks have achieved promising performances in point cloud analysis. However, due to the irregular and unordered data representation, there are still some challenges learning from raw point clouds. PointNet Qi et al. (2017a) is the prior work that applies deep model directly on point clouds. Specifically, a point-wise feature is first extracted for each point independently, and all features are then aggregated into the global representation with a max-pooling operation. However, PointNet missed the important local region structures of point clouds. PointNet++ Qi et al. (2017b) employed a hierarchical strategy to capture the local structures inside small point clusters. To mitigate the impact of point cloud irregularities, some studies utilize scalable indexing techniques to build regular structures on points. OctNet Riegler et al. (2017) hierarchically divided points into regular grids with unbalanced octrees, and extracted the corresponding feature for each leaf node. Kd-Net Klokov and Lempitsky (2017) performed multiplicative transformations according to the subdivisions of point clouds based on kd-trees.

Influenced by traditional convolutional neural networks, some other methods focus on building graphs in local regions and applying CNN-like operations to capture local structures. PointCNN Li et al. (2018b) used the canonical order of local region points and applied a typical convolution operator on transformed features. SPGraph Landrieu and Simonovsky (2018) partitioned large scale points into geometrically homogeneous elements with building a super-point graph. RS-CNN Liu et al. (2019c) also extended typical CNN to irregular configuration, which encoded the geometric relation of points to achieve contextual shape-aware learning of point cloud. DGCNN Wang et al. (2019b) applied an EdgeConv operation to capture the local region structures with dynamic graph updating. These approaches mainly focus on capturing local or global structures of point clouds, which ignores the difference of point importance in distinguishing shape classes. To address this problem, we propose a self-attentive point searching module to capture the point importance and explore fine-grained structures within multiple distinctive point sets.

Distinction Detection on Point Clouds. The distinction, or distinctiveness, was first introduced by Shilane and Funkhouser Shilane and Funkhouser (2006, 2007). These methods rely on extracting local shape descriptors and obtaining the distinctiveness of each local region by comparing the difference between pairs of shape descriptors. One hand-crafted descriptor is usually designed for a specific task, which cannot generalize well to another tasks. And it is difficult to find the best combination of existing hand-crafted descriptors for the current task. To alleviate the limitation of hand-crafted shape descriptors, several methods Shu et al. (2018); Li et al. (2020); Song et al. (2018) employed learning-based strategies to capture shape distinction by back-propagation optimization. Besides 3D shapes, several approaches have been developed to extract the discriminative regions from images. Similar to distinction, saliency has also been explored by Gal and Cohen-Or (2006); Wang et al. (2018b), which considered how unique and visible of regions within the same object. More recently, CP-Net Nezhadarya et al. (2020) performed adaptive down-sampling with considering point importance by making channel-wise statistics on point features. In Zheng et al. (2019), a way of characterizing critical points and segments to build point-cloud saliency maps was proposed. However, existing approaches only focus on extracting high distinctive points, which ignores the spatial correlation information among different distinctive regions. The distinctive regions and their spatial correlations usually provide high-level information for distinguishing every object from others of different classes Shilane and Funkhouser (2007).

Inspired by the region distinction Shilane and Funkhouser (2007), we propose an end-to-end network, called D-Net, to learn point distinction directly from raw point clouds. However, different from the original distinction region, we conduct the distinction detection in a learning-based manner. So, the learned distinctive or important points are relative to the training task, such as shape classification and shape part segmentation. In D-Net, we pay attention to different distinctive regional point sets that contain fine-grained detail information of point clouds and distinguish a shape from objects of a different class. We rely on self-attentive point searching to learn different distinctive point sets and capture the corresponding feature representations with stacked self-gated convolution. Moreover, the final global representation of the point cloud is aggregated with a learnable feature fusion mechanism by leveraging the correlation of distinctive point sets.

3. The D-Net method

The architecture of our D-Net is illustrated in Fig. 2, which consists of three components: the point importance capturing, the distinctive feature extracting, and distinctive feature aggregating. We will illustrate the network details of our D-Net as follows.

Point Importance Capturing. Each point in a point cloud has different importance. To capture structures within the irregular point cloud, we should consider not only the raw point cloud itself but also some fine-grained details within distinctive point sets. We propose a novel self-attentive point searching layer to enrich point cloud information by taking point distinction into consideration. The high distinctive point set and the low distinctive point set are selected out to enhance the point
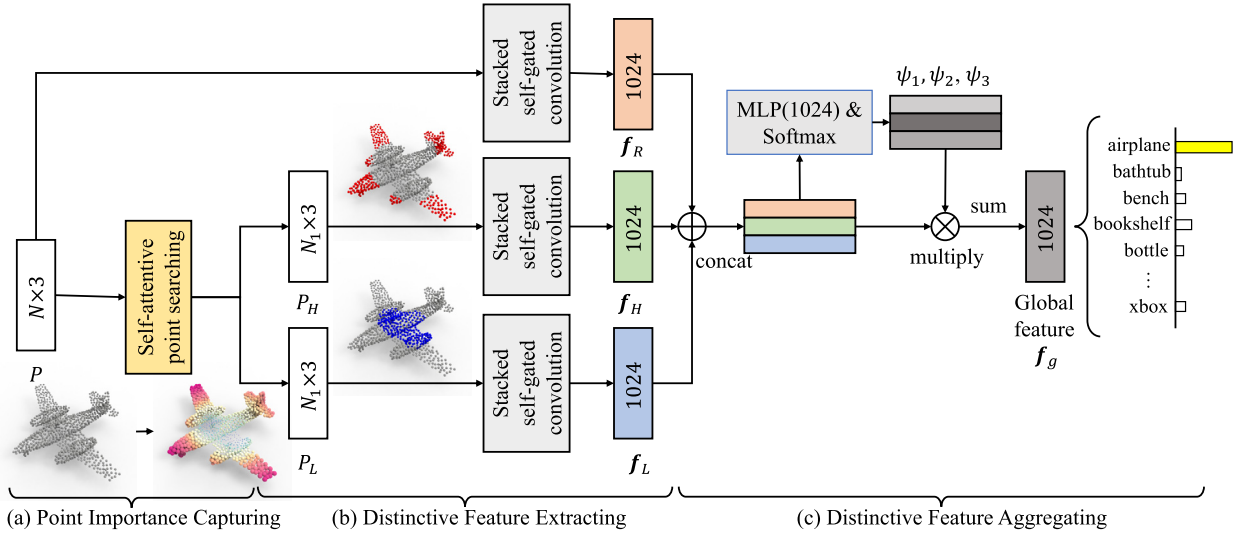
Fig. 2. Our D-Net architecture. Our D-Net has three components, including the point importance capturing, the distinctive feature extracting, and the distinctive feature aggregating. Specifically, the point importance capturing relies on self-attentive point searching to learn a distinction score for each point. By ranking the distinction scores of points, we select out the high distinctive point set P_H and the low distinctive point set P_L to enrich the fine-grained structures of point clouds. With stacked self-gated convolution, we extract the feature representation for each distinctive point set. We resort to a learnable gate for feature aggregation to effectively aggregate point set features f_R, f_H, and f_L into a compact global point cloud representation f_g. The final global representation can be applied to point cloud analysis tasks, including shape classification and shape part segmentation.

cloud representation learning. The enriched details from distinctive point sets can help to boost point cloud analysis tasks, such as shape classification performances.

Distinctive Feature Extracting. With multiple distinctive point sets, we resort to a CNN-like operation with gated aggregation to capture the local region contexts within each distinctive point set. To further exploit the local structures of the point set, the SGC (Self-Gated Convolution) is employed in the point set representation learning, which relies on building graphs that dynamically compose and update the feature representation of each point within a local region. The feature representation of each distinctive point set is extracted by aggregating the different point features with stacked self-gated convolution.

Distinctive Feature Aggregating. Based on the obtained feature representations of distinctive point sets, we introduce a learnable feature fusion mechanism to integrate these features into a compact global point cloud representation. We calculate channel-wise gate weights for each corresponding feature to utilize the spatial correlation among different distinctive features. Afterward, the multiple distinctive set features are aggregated into the final global point cloud representation.

3.1. Self-attentive point searching

The existing point cloud based learning approaches usually treat all points of a point cloud equally, without distinguishing the point distinction that plays an essential role in distinguishing a shape from objects of a different class Shilane and Funkhouser (2006, 2007); Li et al. (2020). As shown in Fig. 1, the point sets in the point cloud can be roughly divided into the high distinctive and the low distinctive point sets. In this paper, besides the raw point cloud, we incorporate the high distinctive point set and the low distinctive point set to enrich the discriminative context of point clouds. With such incorporated contextual information, D-Net can capture the detailed point cloud structure information.

As illustrated in Fig. 3 (a), given a point cloud $P = \{p_i^N_{i=1}\}$ with N points, where $p_i \in \mathbb{R}^3$ denotes the 3D coordinates (x_i, y_i, z_i) of the i-th point. To obtain the distinction score of points, we first transform the point cloud P into two feature space $\boldsymbol{g}, \boldsymbol{h} \in \mathbb{R}^D$ to calculate the bilinear similarity between points, where $\boldsymbol{g}(P) = \boldsymbol{W_g}P$, $\boldsymbol{h}(P) = \boldsymbol{W_h}P$,

$$\beta_{j,i} = \frac{exp(s_{ij})}{\sum_{i=1}^{D} s_{ij}}, \text{ where } s_{ij} = \boldsymbol{g}(p_i)^T \boldsymbol{h}(p_j), \quad \alpha_i = \sum_{j=1}^{N} \beta_{j,i},$$
$$idx_{high} = \underset{N_1}{top}(sort(\{\alpha_i\})),$$
$$idx_{low} = \underset{N_1}{bottom}(sort(\{\alpha_i\}))$$

(1)

(a) Self-attentive point searching
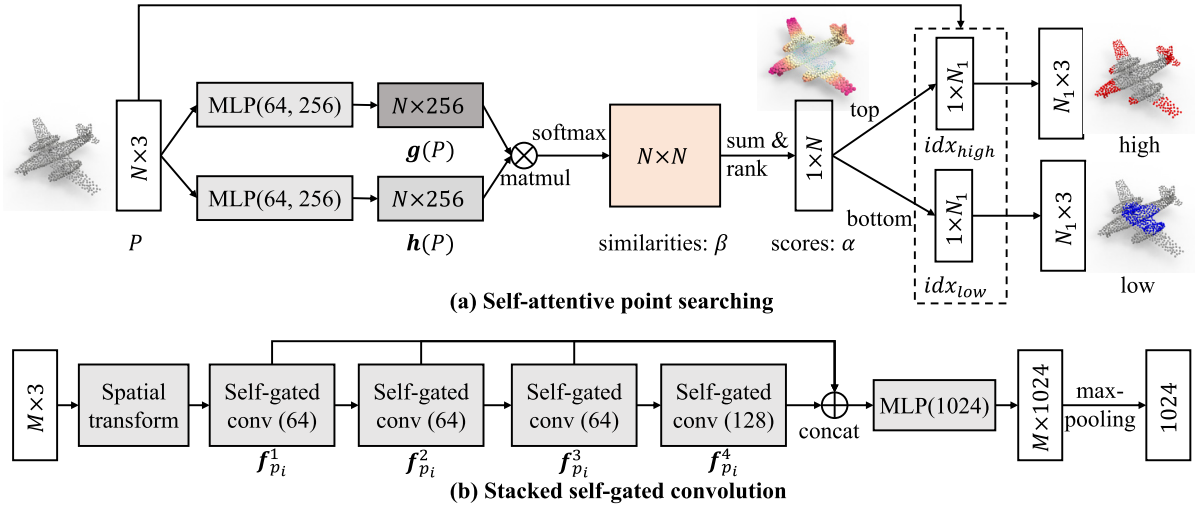

(b) Stacked self-gated convolution

Fig. 3. The proposed modules, including (a) self-attentive point searching module and (b) stacked self-gated convolution module.

Specifically, $\boldsymbol{W_g}$ and $\boldsymbol{W_h}$ are learnable parameters of Multi-Layer Perceptrons (MLPs). $\beta_{j,i}$ evaluates the bilinear similarity between the i-th point p_i and the j-th point p_j in the feature space and α_i denotes the self-attentive score of the i-th point p_i. According to the ranking of self-attentive scores, we search out the index idx_{high} of top N_1 points as the high distinctive point set and idx_{low} of bottom N_1 points as the low distinctive point set, respectively. With the point index idx_{high} and idx_{low}, we pick out the corresponding points to establish the high distinctive point set P_H and low distinctive point set P_L. As such, the contextual information of the raw point cloud is enriched with two selected distinctive point sets. Note that selecting the low distinctive point set P_L in this step aims to make up for the neglecting structural information of point clouds.

3.2. Stacked self-gated convolution

Based on the enriched distinctive point sets, we rely on a distinctive feature extracting (DFE) module with the self-gated CNN to extract the feature representation for each distinctive point set. The architecture of DFE module is illustrated in Fig. 3 (b). Following previous approaches Qi et al. (2017a); Wang et al. (2019b), we first employ a transform layer to align an input point set to a canonical space by applying an estimated 3×3 matrix. The k neighboring points are used in the estimation of 3×3 matrix. We will extract the local structures based on the transformed point set with CNN-like operations.

Self-gated Convolution. Local structure information has been proven to be important in learning the point cloud representation. In order to capture local structures, we propose one novel self-gated CNN to capture the local region context around each point. Similar to Qi et al. (2017b); Wang et al. (2019b), we first use k-nearest neighbor (k-NN) to build a local region $P_r \subset \mathbb{R}^3$, with k surrounding points as its neighbors $p_j \in \mathcal{N}(p_i)$ of p_i. Our goal is to learn an inductive representation $\boldsymbol{f}_{P_r}$ of this local region, which should discriminatively encode the underlying shape information.

To this end, we formulate a general convolutional operation as

$$\boldsymbol{f}_{P_r} = \sigma(\mathcal{A}(\{\mathcal{T}(\boldsymbol{h}_{ij}), \forall p_j\})), \forall p_j \in \mathcal{N}(p_i),$$
$$\boldsymbol{h}_{ij} = \boldsymbol{f}_{p_i} \oplus (\boldsymbol{f}_{p_j} - \boldsymbol{f}_{p_i}), \tag{2}$$

where $\boldsymbol{f}$ is a feature vector, $\oplus$ is the concatenation operation and $\boldsymbol{h}_{ij}$ is the relation description of feature vectors between two points. Here $\boldsymbol{f}_{P_r}$ is obtained by first transforming the features of all the points in $\mathcal{N}(p_i)$ with function $\mathcal{T}$, and then aggregating them with function $\mathcal{A}$ followed by a nonlinear activator σ. In this formulation, the two functions $\mathcal{A}$ and $\mathcal{T}$ are the key to $\boldsymbol{f}_{P_r}$. That is, the permutation invariance of point set can be achieved only when $\mathcal{A}$ is symmetric (e.g., max-pooling) and $\mathcal{T}$ is shared over each point in $\mathcal{N}(p_i)$. As illustrated in Fig. 4, $\mathcal{T}$ is a MLP function, $\mathcal{A}$ is a max-pooling operation and σ is a ReLU layer. Through this convolution operation, we update the point feature $\boldsymbol{f}_{P_i}$ with the regional feature $\boldsymbol{f}_{P_r}$, which contains structural information of local regions.

By stacking T self-gated convolution modules, we can obtain multiple point features $\{\boldsymbol{f}_{p_i}^1, \cdots, \boldsymbol{f}_{p_i}^t, \cdots, \boldsymbol{f}_{p_i}^T\}$ of p_i in different depth of network as shown in Fig. 3 (b). In particular, we also dynamically update the graph of local regions as in Wang et al. (2019b). To obtain the feature representation of each distinctive point set, we first integrate these features into one consistent point representation. A novel self-gate is proposed to integrate these multiple point features. We formulate the self-gated aggregation process as
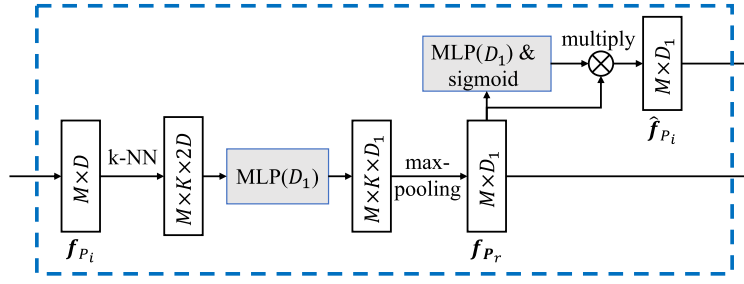
Fig. 4. Self-gated convolution.

$$\theta_{P_i}^t = sigmoid(MLP(\boldsymbol{f}_{p_i}^t)),$$
$$\hat{\boldsymbol{f}}_{p_i}^t = \theta_{P_i}^t \otimes \boldsymbol{f}_{p_i}^t,$$
$$\hat{\boldsymbol{f}}_{p_i} = \hat{\boldsymbol{f}}_{p_i}^1 \oplus \cdots \oplus \hat{\boldsymbol{f}}_{p_i}^t \oplus \cdots \oplus \hat{\boldsymbol{f}}_{p_i}^T,$$
$$\boldsymbol{f} = max(\{\hat{\boldsymbol{f}}_{p_i}\}),$$

$$(3)$$

where $\theta_{p_i}^t$ is the product factor of the self-gate and $\boldsymbol{f}_{p_i}^t$ is the gated point feature at t-th depth of the network. $\otimes$ denotes multiply operation and $\oplus$ denotes the concatenation operation. $\hat{\boldsymbol{f}}_{p_i}$ is the aggregated point feature from multiple depth features. With a max-pooling, the corresponding feature of each distinctive point set is obtained, including the point cloud feature $\boldsymbol{f}_P$, the high distinctive point set feature $\boldsymbol{f}_{P_H}$ and the low distinctive point set feature $\boldsymbol{f}_{P_L}$.

3.3. Learnable feature fusion

Our goal is to aggregate the obtained point cloud feature $\boldsymbol{f}_P$, the high distinctive point set feature $\boldsymbol{f}_{P_H}$ and the low distinctive point set feature $\boldsymbol{f}_{P_L}$ into a global point cloud representation. To fuse information of different distinctive point sets, we propose a non-linear and data-adaptive mechanism to explore the correlation among distinctive point sets. By applying a *softmax* on newly mapped descriptors, we formulate the fusion process as

$$\omega_1 = MLP(\boldsymbol{f}_P), \omega_2 = MLP(\boldsymbol{f}_{P_H}), \omega_3 = MLP(\boldsymbol{f}_{P_L}),$$
$$\psi_{1,c} = \frac{exp(\omega_{1,c})}{exp(\omega_{1,c}) + exp(\omega_{2,c}) + exp(\omega_{3,c})},$$
$$\psi_{2,c} = \frac{exp(\omega_{2,c})}{exp(\omega_{1,c}) + exp(\omega_{2,c}) + exp(\omega_{3,c})},$$
$$\psi_{3,c} = \frac{exp(\omega_{3,c})}{exp(\omega_{1,c}) + exp(\omega_{2,c}) + exp(\omega_{3,c})},$$

$$(4)$$

where $\omega_1, \omega_2, \omega_3 \in \mathbb{R}^{1024}$ are new mapped descriptors that are transformed from channel-wise descriptors $\boldsymbol{f}_P$, $\boldsymbol{f}_{P_H}$ and $\boldsymbol{f}_{P_L}$ with different MLPs. $\psi_{1,c}$, $\psi_{2,c}$ and $\psi_{3,c}$ are the fusion weights of point set features, where $\psi_{1,c} + \psi_{2,c} + \psi_{3,c} = 1$. In particular, the symbol c indicates that the learnable feature fusion is a channel-wise operation, which enables the exploration of general patterns between channels.

The ultimate global point cloud feature $\boldsymbol{f}_g$ can be fused by a weighted sum operation as

$$\boldsymbol{f}_{g:c} = \psi_{1,c} \cdot \boldsymbol{f}_{P:1,c} + \psi_{2,c} \cdot \boldsymbol{f}_{P_H:2,c} + \psi_{1,c} \cdot \boldsymbol{f}_{P_L:1,c}, \qquad (5)$$

where $\boldsymbol{f}_{g:c}$ denotes the c-th channel value of $\boldsymbol{f}_g$. The learned global shape representation $\boldsymbol{f}_g$ can be applied to various point cloud analysis applications, such as shape classification and shape part segmentation.

3.4. Training loss

In the representation learning of point clouds, we focus on capturing the discriminative point cloud features, which can be applied to multiple point analysis applications, including point cloud classification and shape part segmentation. Following the strategy of previous point cloud recognition methods, we train our approach in an end-to-end manner with the cross-entropy loss. For example, in the point cloud classification task, we classify the extract global feature $\boldsymbol{f}_g$ into one of C shape classes by a softmax function layer. The softmax function outputs the classification probabilities $\boldsymbol{p}$, such that each value $\{\boldsymbol{p}(c), c \in [1, C]\}$ indicates the probability under the current class. The objective function of $\mathcal{L}_{ce}$ is the cross entropy between $\boldsymbol{p}$ and the ground-truth probability $\boldsymbol{p}'$,

$$\mathcal{L}_{ce}(\boldsymbol{p}, \boldsymbol{p}') = -\sum_{c \in [1,C]} \boldsymbol{p}'(c) log \, \boldsymbol{p}(c). \tag{6}$$

Similar to point cloud classification, the shape part segmentation can be regarded as a low-level point-wise classification task. Therefore, we also employ the cross entropy loss in the shape part segmentation task.

4. Experiments

In this section, we arrange comprehensive experiments to validate the proposed D-Net. First, we investigate how some key parameters affect the performance of D-Net. And then, we evaluate D-Net in shape classification and shape part segmentation, respectively. We also conducted an ablation study to evaluate the effectiveness of each proposed module under the shape classification benchmark. In addition, we visualize some experimental results to show the performance of our approach intuitively. Finally, we do a model complexity analysis to show the effectiveness of our D-Net.

Implementation Details. The number of neighboring points k in each local region is set as 20. And the number of high distinctive points and low distinctive points N_1 is set as 320. For feature extraction, a stacked self-gated CNN at depth four is used, where the output of each self-gated CNN is set as 64, 64, 64, and 128, respectively. During the training, we use an ADAM optimizer with an initial learning rate of 0.001, a batch size of 16, and a batch normalization rate of 0.5. In addition, ReLU is used after each fully connected layer followed by a dropout layer with a drop ratio 0.5 in the fully connected layers.

Parameters. All the experiments in this section were evaluated on ModelNet40, which contains 40 categories and 12,311 CAD shapes with 9,843 shapes for training and 2,468 shapes for testing, respectively. For each 3D shape, we adopt the point cloud with 1,024 points Qi et al. (2017a,b); Wang et al. (2019b) which are uniformly sampled from the corresponding mesh as input.

In the module of distinctive feature extracting, the number of neighbors k in each local region is an important parameter, which influences the capturing of local structures around each point. The results of several settings of k are shown in Table 3. The best instance accuracy 93.15% is reached at $k = 20$, which can better cover the context information inside local regions. From the results, we can see that capturing local structures plays an important role in enhancing the global representation learning of point clouds.

Finally, we explore the effect of the distinctive point set size N_1 under ModelNet40. N_1 is also an important parameter that influences the capturing of the structure information within distinctive point sets. Due to the variance of point clouds, the frequency of distinctive points and non-distinctive points are likely to be class-dependent or even shape-dependent. To simplify the parameter setting, we set the number of distinctive and non-distinctive points as a same value N_1. And in order to further explore the influence of N_1, we have conducted parameter comparison in Table 4. As shown in Table 4, the best install accuracy is achieved at $N_1 = 320$, which better covers the entire training data.

Shape Classification. We evaluate D-Net on ModelNet40 classification benchmark Wu et al. (2015), which is composed of 9,843 train models and 2,468 test models in 40 classes. Same as Qi et al. (2017a,b), we sample 1,024 points and normalize them to a unit sphere for each point cloud.

The quantitative comparisons with the state-of-the-art point-based methods are summarized in Table 1, where D-Net outperforms all the xyz-input methods. On the ModelNet40 dataset, our approach performs best among the compared methods, reaching 93.2% in instance accuracy. We further test our method with point normals. For a fair comparison, we report the results of RS-CNN without voting strategy according to their paper Liu et al. (2019c). In addition, we also show the results of D-Net in unsupervised shape classification under the ModelNet40 benchmark. Following PointNet++ Qi et al. (2017b) and LGAN Achlioptas et al. (2017), we integrate feature interpolation for point upsampling and earth mover's distance (EMD) as the training loss to formulate D-Net++. And we employ multi-class SVM to be the classifier as in Liu et al. (2019b). D-Net also achieves state-of-the-art performances in the unsupervised shape classification application.

To compare the point distinction detection performances, we revisit the public distinction detection methods and corresponding source code. We compare our D-Net with existing methods, including Farthest point sampling (FPS), ISS Zhong (2009) and UDDR Li et al. (2020) as shown in Fig. 6. Specifically, FPS is a widely applied algorithm for uniform sampling of point clouds, which selects points according to the point-to-point distance and ignores the point distinction. Here, we select 32 points from each raw point cloud (1024 points). The ISS algorithm is a traditional key point detection method, which calculates the point distinction via the geometry distribution of points. Due to the predefined threshold value, the number of selected key points in ISS algorithm is unstable. And the UDDR is a learning-based distinctive region detection method, which learns the point distinction with a clustering-based nonparametric softmax classifier in an iterative re-clustering manner. Similarly, both UDDR and D-Net predict a distinction score for each point, where the warmth of the point color indicates the magnitude of the distinction score. From the visualization results, we can see that both UDDR and D-Net tend to assign higher distinction scores to edge points or sharp points. And the distribution of the distinction point is largely determined by the design of different strategies, such as the distinction metric and the task optimization. Overall, different from existing methods, our D-Net focuses more on extracting high-frequency information, such as corners or edges, to learn

Table 1
Shape classification results (%) under ModelNet40.

Methods	Supervised	Input	#Points	Acc.
PointNet Qi et al. (2017a)	yes	xyz	1k	89.2
PointNet++ Qi et al. (2017b)	yes	xyz	1k	90.7
Kd-Net(depth=10) Klokov and Lempitsky (2017)	yes	xyz	1k	90.6
KC-Net Shen et al. (2018)	yes	xyz	1k	91.0
ShapeContextNet Xie et al. (2018)	yes	xyz	1k	90.0
LRC-Net Liu et al. (2020)	yes	xyz	1k	93.1
PointCNN Li et al. (2018b)	yes	xyz	1k	91.7
PCNN Atzmon et al. (2018)	yes	xyz	1k	92.3
DGCNN Wang et al. (2019b)	yes	xyz	1k	92.9
Point2Sequence Liu et al. (2019a)	yes	xyz	1k	92.6
A-CNN Komarichev et al. (2019)	yes	xyz	1k	92.6
RS-CNN Liu et al. (2019c)	yes	xyz	1k	92.9
SO-Net Li et al. (2018a)	yes	xyz	2k	90.9
Kd-Net(depth=15) Klokov and Lempitsky (2017)	yes	xyz	32k	91.8
CP-Net Nezhadarya et al. (2020)	yes	xyz	1k	92.4
Grid-GCN Xu et al. (2020)	yes	xyz	16k	93.1
PACov Xu et al. (2021)	yes	xyz	1k	93.6
PointASNL Yan et al. (2020)	yes	xyz	1k	93.2
PointTransformer Zhao et al. (2021)	yes	xyz	1k	**93.7**
O-CNN Wang et al. (2017)	yes	xyz+nor.	-	90.6
Spec-GCN Wang et al. (2018a)	yes	xyz+nor.	10k	91.8
PointNet++ Qi et al. (2017b)	yes	xyz+nor.	5k	91.9
SpiderCNN Xu et al. (2018)	yes	xyz+nor.	5k	92.4
D-Net(Ours)	yes	xyz	1k	93.2
D-Net(Ours)	yes	xyz+nor.	1k	93.3
LGAN Achlioptas et al. (2017)	no	xyz	2k	85.7
FoldingNet Yang et al. (2018)	no	xyz	2k	88.4
MAP-VAE Han et al. (2019)	no	xyz	2k	90.2
UDDR Li et al. (2020)	no	xyz	1k	88.1
L2G-AE Liu et al. (2019b)	no	xyz	1k	90.6
D-Net++(Ours)	no	xyz	1k	**90.9**

discriminative point cloud representations. And the experimental results have proved the effectiveness of our strategy to explore the point distinction in the feature space.

Shape Part Segmentation. Part segmentation is a challenging point cloud analysis task, which predicts a semantic part label for each input point. We evaluate D-Net in shape part segmentation on ShapeNet part benchmark Yi et al. (2016). This dataset consists of 16,881 models from 16 shape categories and is labeled in 50 part classes in total. We follow Qi et al. (2017a) to split shapes into the training set and test set with 2,048 points for each point cloud. For comparison, we report the mean instance IoU (Intersection-over-Union) that is averaged across all instance point clouds.

Table 2 shows the quantitative comparisons with the state-of-the-art approaches, where D-Net achieves the best performance with an instance mIoU of 86.2%. We resort to FPS to select points from the raw point cloud, where Farthest Point Sampling (FPS) can better capture the underlying shape for point clouds in part segmentation. The results of different settings are shown in Table 2, including input with xyz only (xyz), with xyz and normal (xyz,nor.), and using self-attentive point searching (sps). To qualitatively show the performance of D-Net, we also visualize some examples of prediction results, where our results are high consistency with the ground-truth as shown in Fig. 5.

Ablation Study. To quantitatively evaluate the effect of proposed modules, we show the performances of D-Net under three settings: without self-attentive point searching module (NoSPS), without self-gate in the convolution module (NoSG), and without learnable feature fusion module (NoLFF). Specifically, in NoSPS, we utilize FPS (FPS) and random sampling (RS) to replace the self-attentive point searching. And in the NoLFF, we replace the learnable gate with max-pooling (Max), mean pooling (Mean), or simple concatenation (SC). In addition, we also show the result of D-Net with all proposed modules (ALL). As shown in Table 5, we report the results of D-Net with the above settings. Each module effectively learns the discriminative point cloud representation, which distinguishes a shape from other classes. And to explore the geodesic distance in searching the distinctive points, we also show the results of extracting local features with geodesic distances (Geo). To calculate the geodesic distance between points, we directly apply the Isomap Tenenbaum et al. (2000) with the Python implementation for approximating the geodesic distance between points. Thus, we adopt point geodesic distances to search neighbor points for extracting point features from local regions. Due to the limitation of approximation accuracy and large computational complexity, the geodesic distance is not performing well in the point cloud classification. In addition, we also report the distinction detection results under the usage of point geodesic distance.

To explore the effect of the distinctive point set, we adjust the used distinctive point set as shown in Table 6. Specifically, we report several results, including raw point cloud only (P_R), raw point cloud and high distinctive point set ($P_R + P_H$),

Table 2

The shape segmentation results (%) on ShapeNet part segmentation dataset.

Methods	Mean IoU	Intersection over Union (IoU)															
		air.	bag	cap	car	cha.	ear.	gui.	kni.	lam.	lap.	mot.	mug	pis.	roc.	ska.	tab.
# SHAPES		2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271
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
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
ShapeContextNet Xie et al. (2018)	84.6	83.8	80.8	83.5	79.3	90.5	69.8	91.7	86.5	82.9	96.0	69.2	93.8	82.5	62.9	74.4	80.8
KCNet 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 Wang et al. (2019b)	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
SO-Net Li et al. (2018a)	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
A-CNN Komarichev et al. (2019)	86.1	84.2	84.0	88.0	79.6	**91.3**	75.2	91.6	87.1	85.5	95.4	75.3	94.9	82.5	**67.8**	77.5	83.3
RS-CNN Liu et al. (2019c)	**86.2**	83.5	84.8	88.8	79.6	91.2	81.1	91.6	**88.4**	**86.0**	96.0	73.7	94.1	83.4	60.5	77.7	**83.6**
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
PointCNN Li et al. (2018b)	86.1	84.1	**86.5**	86.0	**80.8**	90.6	79.7	**92.3**	88.4	85.3	**96.1**	**77.2**	95.3	**84.2**	64.2	**80.0**	83.0
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
LRC-Net Liu et al. (2020)	85.3	82.6	85.2	87.4	79.0	90.7	80.2	91.3	86.9	84.5	95.5	71.4	93.8	79.4	51.7	75.5	82.6
D-Net(sps)	85.7	83.8	81.0	86.9	78.4	91.1	74.7	91.6	87.1	84.4	95.7	71.6	95.1	80.9	58.9	75.3	83.4
D-Net(xyz)	86.0	**84.5**	83.1	85.2	78.9	91.0	**81.9**	91.5	86.7	85.0	95.9	72.7	**95.9**	82.4	58.2	76.6	83.4
D-Net(xyz+nor.)	**86.2**	84.4	83.7	**90.4**	79.4	**91.3**	80.5	91.8	86.9	85.6	96.0	74.9	95.7	83.5	59.4	78.8	83.2

Table 3
The effect of the number of neighbors k under ModelNet40.

k	5	10	15	20	25	30
Instance acc. (%)	91.94	92.54	92.75	**93.15**	93.07	92.75

Table 4
The effect of distinctive point set size N_1 under ModelNet40.

N_1	384	320	256	192	128
Instance acc. (%)	92.83	**93.15**	92.63	92.79	92.34

Table 5
The ablation study under ModelNet40.

Metric	Geo	FPS	RS	NoSG	Max	Mean	SC	ALL
Acc.	92.78	92.71	92.79	91.08	92.34	92.63	92.63	**93.15**
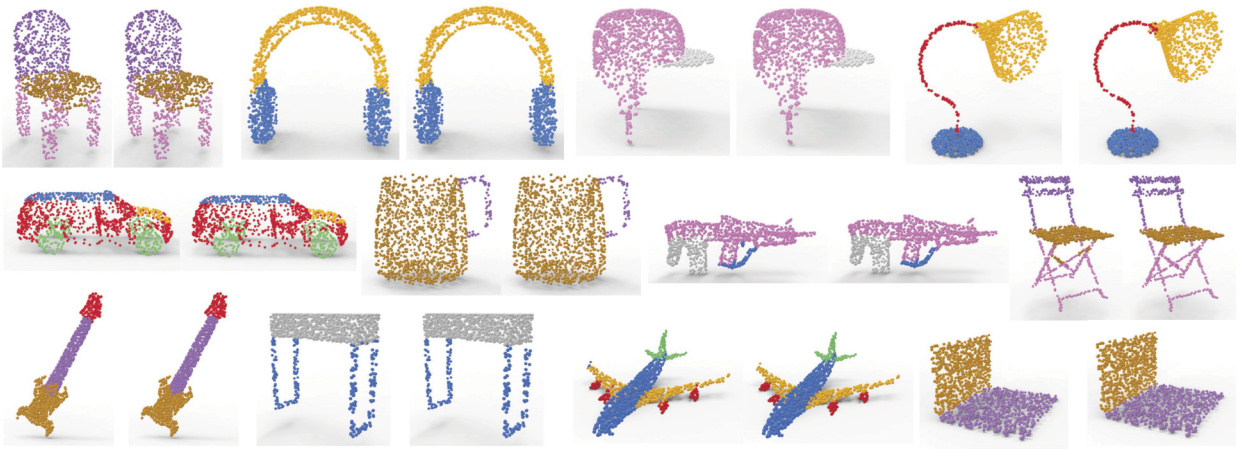
Fig. 5. The visualization of shape part segmentation. For each pair of 3D shapes, the left one indicates the ground-truth shape segmentation, and the right one is the predicted shape segmentation. In each shape, each color denotes a specific semantic part label.

Table 6
The effect of used point sets under ModelNet40.

Metric	P_R	$P_R + P_H$	P_H	$P_H + P_L$	ALL
Instance acc. (%)	92.34	92.54	90.7	92.1	**93.15**

high distinctive point set only (P_H), high distinctive point set, low distinctive point set ($P_H + P_L$), and all distinctive point sets ($P_R + P_H + P_L$, ALL).

Visualization Analysis. In D-Net, there are some important visualization results that should be shown, including distinctive point clouds and learnable feature fusion weights. In Fig. 7, we show some examples of the distinctive point clouds, where the self-attentive point searching captures a distinction score for each point. In addition, we also show the learnable feature fusion weights in Fig. 8, where the first row is the raw point cloud and corresponding fusion weights. In particular, we resize each 1024-dimensional weight vector into 32×32 matrix for visualization. Similar to the first row, the second row represents the high distinctive point sets, and the third row represents low distinctive point sets. And we have marked several similar areas with red boxes in the weight matrixes. For two airplanes from the same class in ModelNet40, the weight matrix shows the consistency between the same class and the difference between different classes.

To further evaluate our D-Net in terms of effectively learning distinctive points, we show the visualization of some distinctive point clouds from ISDB Gal et al. (2007). ISDB is a database of different articulated models of animals and humans containing about 104 models. We sample 1,024 points from each mesh shape with Poisson Disk Sampling Corsini et al. (2012) and directly process these points with a network trained under ModelNet40. In Fig. 9, we visualize the distinctive point clouds of horses, dogs, and lions. The visualization results suggest that our method can learn shapes with different poses with high consistency within the same class, such as horse and dog. In addition, the distinction distribution of shapes under ModelNet40 also shows high consistency, as shown in Fig. 10.
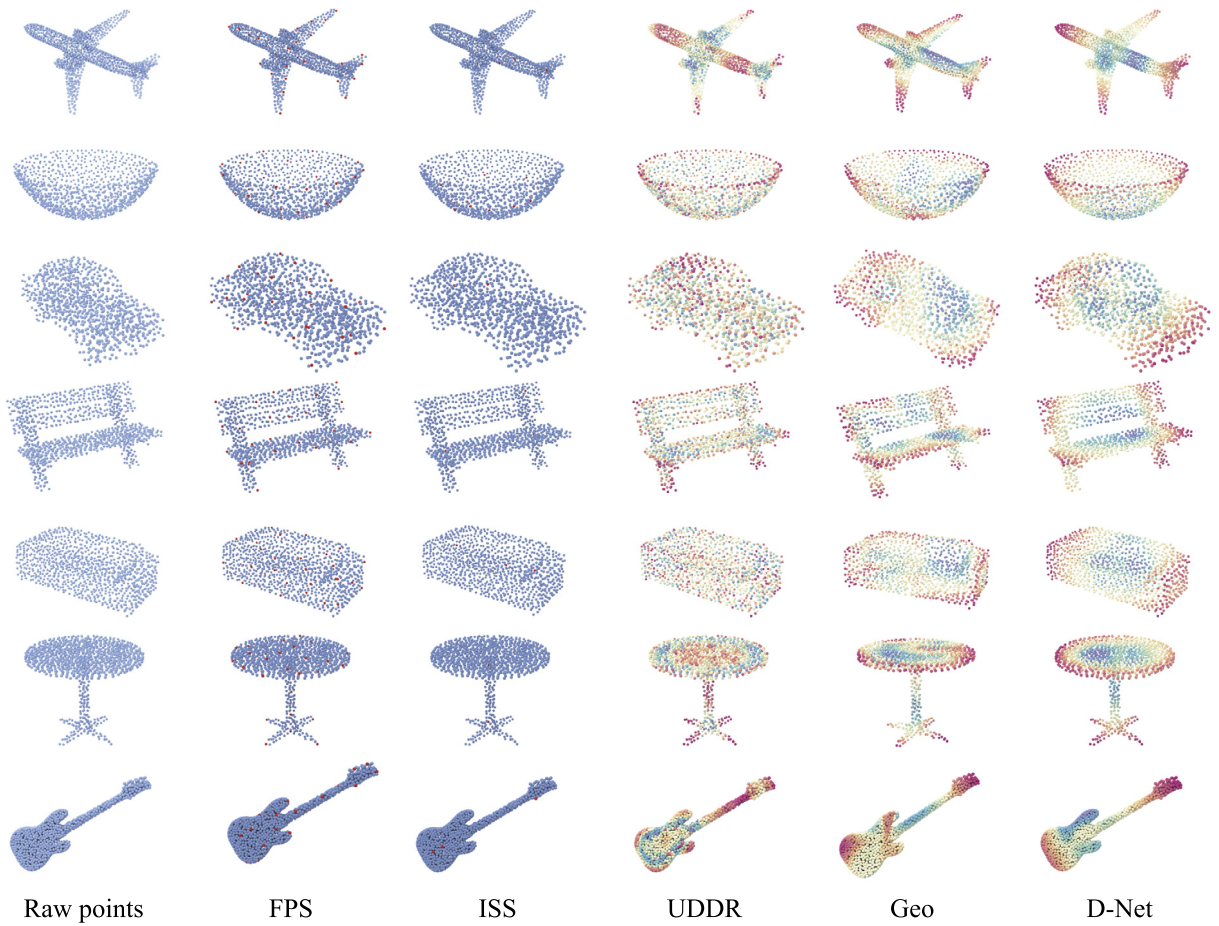
Fig. 6. The comparisons with existing methods on distinction detection.

Fig. 7. The visualization of point distinction under ModelNet40.

In D-Net, we propose a self-attentive point searching module to capture an important score for each point. As shown in the experiments, the distinction distribution of points tends to be consistent within a class. To further reveal the consistency of the learned features, we draw the feature distribution as shown in Fig. 11. The horizontal axis represents the feature channel (1,024-dimensional), and the vertical axis represents the feature value of each channel, such as airplanes and beds. From the visualization, we can know that the learned features from the same class have a close feature distribution and high feature consistency. Otherwise, the learned features from different classes have low feature consistency.
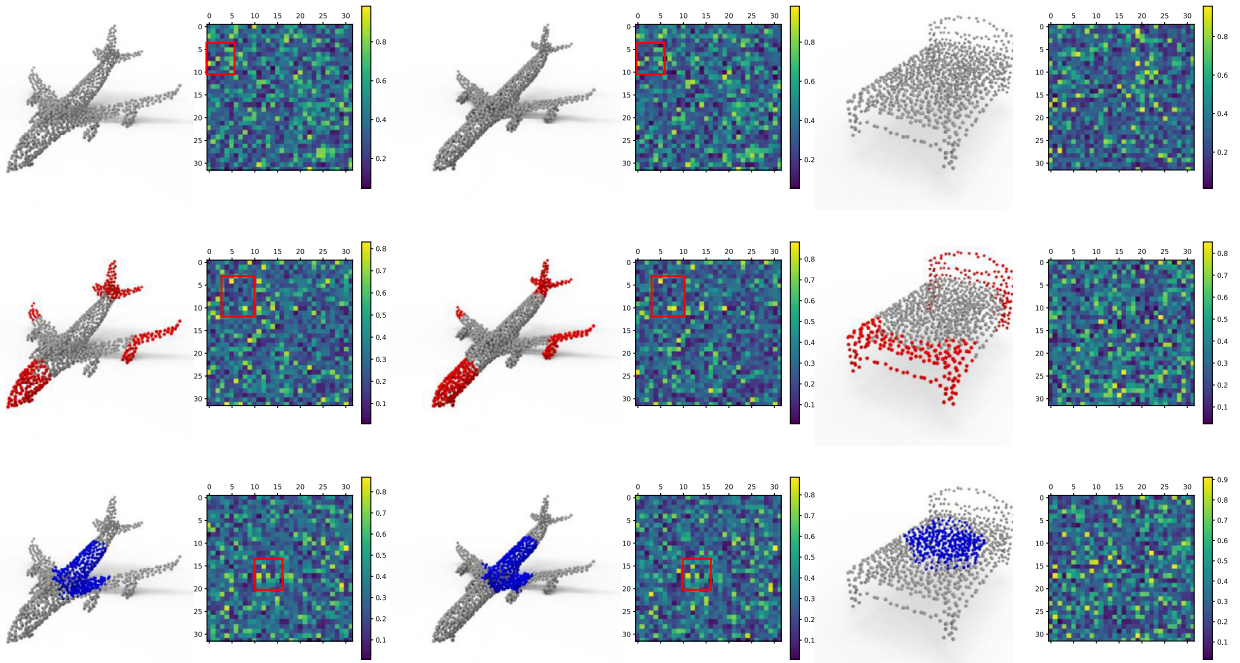
Fig. 8. The visualization of feature fusion weights.

Fig. 9. Consistent distinction distribution under ISDB Gal et al. (2007).

5. Model complexity

In addition, to show the network complexity of D-Net intuitively, we make statistics of model size and computational cost of some point cloud based methods. We follow PointNet++ to evaluate the time and space cost of several point cloud based methods as shown in Table 7. We record forward time under the same conditions with a batch size 8 using TensorFlow 1.0 using a single GTX 1080 Ti. Table 7 shows D-Net can achieve a trade-off between the model complexity (number of parameters) and computational complexity (forward pass time). For a fair comparison, we report the model complexity of PointNet++ Qi et al. (2017b) and DGCNN Wang et al. (2019b), which are the basis of our D-Net. In addition, we also show the complexity of some recent works including PointCNN Li et al. (2018b) and LRC-Net Liu et al. (2020).

6. Conclusion

In this paper, D-Net (Distinctive Network) is proposed to capture the distinction of points for global feature learning in point cloud analysis tasks. The core of D-Net is point importance capturing, which successfully learns a distinction score
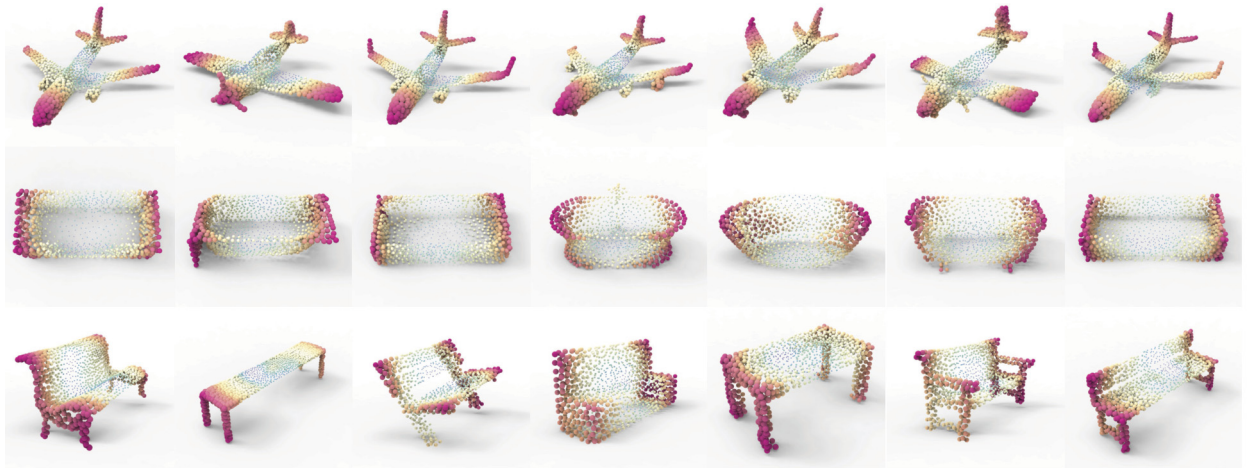
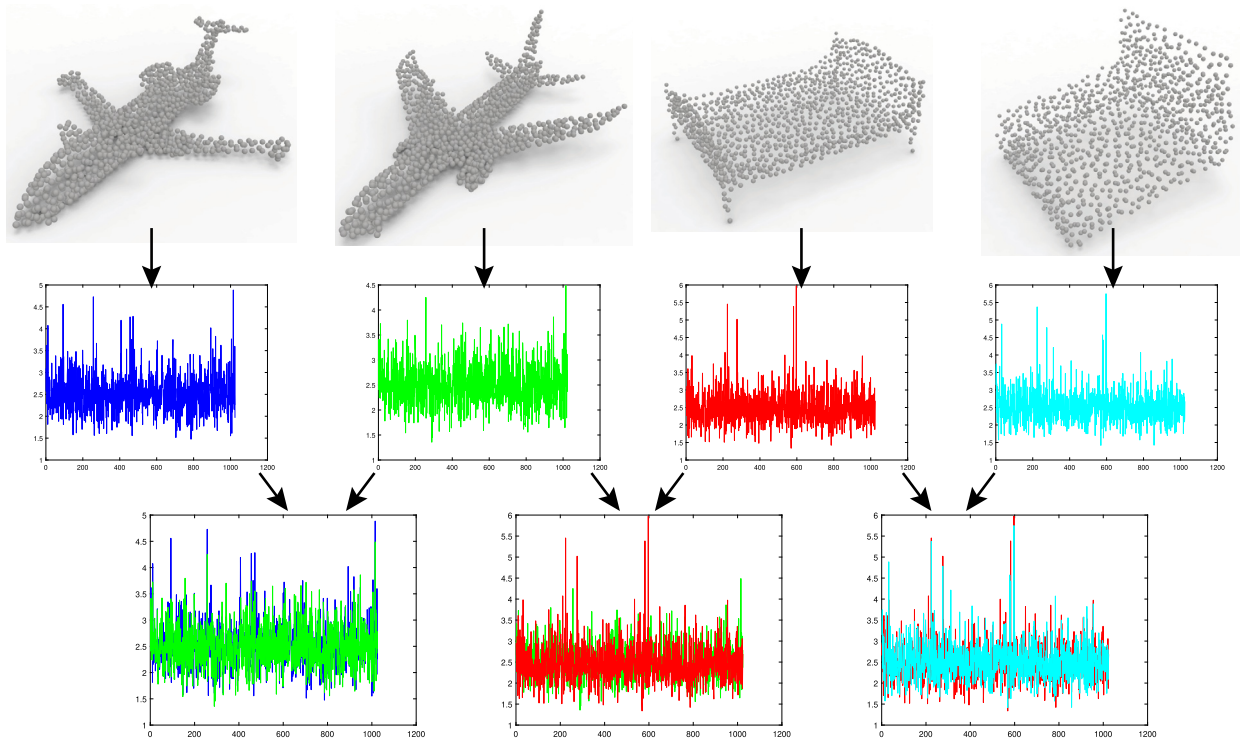**Fig. 10.** Shape consistency under ModelNet40.

Fig. 11. The visualization of feature consistency. The first row contains four 3D point clouds, including two airplanes and two beds. And the second row indicates the corresponding learned feature distribution of corresponding point clouds, where each feature is extracted by D-Net under ModelNet40. To show the consistency between the learned features, we show the overlapping of different feature distributions in the last row, where different features are mixed as shown by arrows.

for each point with self-attentive point searching. With multiple distinctive point sets, the proposed stacked self-gated convolution module can effectively extract distinctive set features. In addition, our learnable feature fusion mechanism effectively aggregates distinctive point set features into a global point cloud representation. Our superior over other methods in experiments demonstrates that D-Net is effective in the learning of distinction for points, which can be further leveraged for better point cloud analysis.

CRediT authorship contribution statement

Xinhai Liu: Conceptualization, Methodology, Writing and Editing.

Table 7
Complexity, forward time, and accuracy of different models under ModelNet40.

Method	Model size (MB)	Time (ms)	Accuracy (%)
PointNet (vanilla) Qi et al. (2017a)	9.4	6.8	87.1
PointNet Qi et al. (2017a)	40	16.6	89.2
PointNet++ (SSG) Qi et al. (2017b)	8.7	82.4	-
PointNet++ (MSG) Qi et al. (2017b)	12	163.2	90.7
LRC-Net Liu et al. (2020)	18	115.8	93.1
DGCNN Wang et al. (2019b)	21	27.2	92.9
PointCNN Li et al. (2018b)	94	117.0	92.3
D-Net (ours)	28.8	85.8	93.2

Sanghuk Lee: Experiment, Visualization and Investigation.
Yan-Pei Cao: Methodology and Editing.
Yu-Shen Liu: Conceptualization, Methodology, and Editing.
Zhizhong Han: Conceptualization, Methodology, and Editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgements

References

Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L., 2017. Learning representations and generative models for 3D point clouds. In: International Conference on Machine Learning.

Atzmon, M., Maron, H., Lipman, Y., 2018. Point convolutional neural networks by extension operators. ACM Trans. Graph.

Chen, C., Li, G., Xu, R., Chen, T., Wang, M., Lin, L., 2019. ClusterNet: deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4994–5002.

Chen, X., Ma, H., Wan, J., Li, B., Xia, T., 2017. Multi-view 3D object detection network for autonomous driving. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1907–1915.

Corsini, M., Cignoni, P., Scopigno, R., 2012. Efficient and flexible sampling with blue noise properties of triangular meshes. IEEE Trans. Vis. Comput. Graph., 914–924.

Gal, R., Cohen-Or, D., 2006. Salient geometric features for partial shape matching and similarity. ACM Trans. Graph., 130–150.

Gal, R., Shamir, A., Cohen-Or, D., 2007. Pose-oblivious shape signature. IEEE Trans. Vis. Comput. Graph., 261–271.

Han, Z., Wang, X., Liu, Y.S., Zwicker, M., 2019. Multi-angle point cloud-VAE: unsupervised feature learning for 3D point clouds from multiple angles by joint self-reconstruction and half-to-half prediction. In: IEEE International Conference on Computer Vision.

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A., 2020. RandLA-Net: efficient semantic segmentation of large-scale point clouds. In: IEEE Conference on Computer Vision and Pattern Recognition.

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

Komarichev, A., Zhong, Z., Hua, J., 2019. A-CNN: annularly convolutional neural networks on point clouds. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 7421–7430.

Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4558–4567.

Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O., 2019. PointPillars: fast encoders for object detection from point clouds. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 12697–12705.

Lee, C.H., Varshney, A., Jacobs, D.W., 2005. Mesh saliency. In: ACM SIGGRAPH, pp. 659–666.

Li, J., Chen, B.M., Hee Lee, G., 2018a. SO-Net: self-organizing network for point cloud analysis. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 9397–9406.

Li, X., Yu, L., Fu, C.W., Cohen-Or, D., Heng, P.A., 2020. Unsupervised detection of distinctive regions on 3D shapes. ACM Trans. Graph. 39, 1–14.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B., 2018b. PointCNN: convolution on X-transformed points. In: Advances in Neural Information Processing Systems, pp. 820–830.

Liu, X., Han, Z., Hong, F., Liu, Y.S., Zwicker, M., 2020. LRC-Net: learning discriminative features on point clouds by encoding local region contexts. Comput. Aided Geom. Des. 101859.

Liu, X., Han, Z., Liu, Y.S., Zwicker, M., 2019a. Point2Sequence: learning the shape representation of 3D point clouds with an attention-based sequence to sequence network. In: AAAI Conference on Artificial Intelligence, pp. 8778–8785.

Liu, X., Han, Z., Wen, X., Liu, Y.S., Zwicker, M., 2019b. L2G auto-encoder: understanding point clouds by local-to-global reconstruction with hierarchical self-attention. In: ACM International Conference on Multimedia, pp. 989–997.

Liu, Y., Fan, B., Xiang, S., Pan, C., 2019c. Relation-shape convolutional neural network for point cloud analysis. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 8895–8904.

Meng, H.Y., Gao, L., Lai, Y.K., Manocha, D., 2019. VV-Net: voxel VAE net with group convolutions for point cloud segmentation. In: IEEE International Conference on Computer Vision, pp. 8500–8508.

Nezhadarya, E., Taghavi, E., Liu, B., Luo, J., 2020. Adaptive Hierarchical Down-Sampling for Point Cloud Classification, pp. 12956–12964.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. PointNet: deep learning on point sets for 3D classification and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 652–660.

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

Riegler, G., Osman Ulusoy, A., Geiger, A., 2017. OctNet: learning deep 3D representations at high resolutions. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3577–3586.

Roveri, R., Rahmann, L., Oztireli, C., Gross, M., 2018. A network architecture for point cloud classification via automatic depth images generation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4176–4184.

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

Shi, S., Wang, X., Li, H., 2019. PointRCNN: 3D object proposal generation and detection from point cloud. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–779.

Shilane, P., Funkhouser, T., 2006. Selecting distinctive 3D shape descriptors for similarity retrieval. In: International Conference on Shape Modeling and Applications, p. 18.

Shilane, P., Funkhouser, T., 2007. Distinctive regions of 3D surfaces. ACM Trans. Graph.

Shu, Z., Xin, S., Xu, X., Liu, L., Kavan, L., 2018. Detecting 3D points of interest using multiple features and stacked auto-encoder. IEEE Trans. Vis. Comput. Graph., 2583–2596.

Song, R., Liu, Y., Rosin, P.L., 2018. Distinction of 3D objects and scenes via classification network and Markov random field. IEEE Trans. Vis. Comput. Graph.

Tenenbaum, J.B., Silva, V.d., Langford, J.C., 2000. A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323.

Wang, C., Samari, B., Siddiqi, K., 2018a. Local spectral graph convolution for point set feature learning. In: European Conference on Computer Vision, pp. 52–66.

Wang, P.S., Liu, Y., Guo, Y.X., Sun, C.Y., Tong, X., 2017. O-CNN: octree-based convolutional neural networks for 3D shape analysis. ACM Trans. Graph., 1–11.

Wang, X., He, J., Ma, L., 2019a. Exploiting local and global structure for point cloud semantic segmentation with contextual point representations. In: Advances in Neural Information Processing Systems, pp. 4573–4583.

Wang, X., Koch, S., Holmqvist, K., Alexa, M., 2018b. Tracking the gaze on objects in 3D: how do people really look at the bunny? ACM Trans. Graph., 1–18.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019b. Dynamic graph CNN for learning on point clouds. ACM Trans. Graph. 38, 1–12.

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

Xie, S., Liu, S., Chen, Z., Tu, Z., 2018. Attentional shapecontextnet for point cloud recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4606–4615.

Xu, M., Ding, R., Zhao, H., Qi, X., 2021. PAConv: position adaptive convolution with dynamic kernel assembling on point clouds. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3173–3182.

Xu, Q., Sun, X., Wu, C.Y., Wang, P., Neumann, U., 2020. Grid-GCN for fast and scalable point cloud learning. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 5661–5670.

Xu, Y., Fan, T., Xu, M., Zeng, L., Qiao, Y., 2018. SpiderCNN: deep learning on point sets with parameterized convolutional filters. In: European Conference on Computer Vision, pp. 87–102.

Yan, X., Zheng, C., Li, Z., Wang, S., Cui, S., 2020. PointASNL: robust point clouds processing using nonlocal neural networks with adaptive sampling. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 5589–5598.

Yang, B., Wang, J., Clark, R., Hu, Q., Wang, S., Markham, A., Trigoni, N., 2019. Learning object bounding boxes for 3D instance segmentation on point clouds. In: Advances in Neural Information Processing Systems, pp. 6737–6746.

Yang, Y., Feng, C., Shen, Y., Tian, D., 2018. FoldingNet: point cloud auto-encoder via deep grid deformation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 206–215.

Yi, L., Kim, V.G., Ceylan, D., Shen, I.C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., Guibas, L., 2016. A scalable active framework for region annotation in 3D shape collections. ACM Trans. Graph., 1–12.

Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V., 2021. Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 16259–16268.

Zheng, T., Chen, C., Yuan, J., Li, B., Ren, K., 2019. PointCloud saliency maps. In: IEEE International Conference on Computer Vision, pp. 1598–1606.

Zhong, Y., 2009. Intrinsic shape signatures: a shape descriptor for 3D object recognition. In: IEEE International Conference on Computer Vision Workshops, pp. 689–696.

Zhou, Y., Tuzel, O., 2018. VoxelNet: end-to-end learning for point cloud based 3D object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4490–4499.