

Learning Inner-Group Relations on Point Clouds

Haoxi Ran^{*†} Wei Zhuo[†] Jun Liu[†] Li Lu^{*}
^{*}Sichuan University [†]Tencent
 {ranhaoxi, junsenselee}@gmail.com
 weizhuo@tencent.com luli@scu.edu.cn

Abstract

The prevalence of relation networks in computer vision is in stark contrast to underexplored point-based methods. In this paper, we explore the possibilities of local relation operators and survey their feasibility. We propose a scalable and efficient module, called group relation aggregator. The module computes a feature of a group based on the aggregation of the features of the inner-group points weighted by geometric relations and semantic relations. We adopt this module to design our **RPNet**. We further verify the expandability of RPNet, in terms of both depth and width, on the tasks of classification and segmentation. Surprisingly, empirical results show that wider RPNet fits for classification, while deeper RPNet works better on segmentation. RPNet achieves state-of-the-art for classification and segmentation on challenging benchmarks. We also compare our local aggregator with PointNet++, with around 30% parameters and 50% computation saving. Finally, we conduct experiments to reveal the robustness of RPNet with regard to rigid transformation and noises.

1. Introduction

Point cloud processing has attracted considerable attention for its advantages in various applications, including autonomous driving, augmented reality, and robotics. Though easily accessible, unlike other visual elements (i.e., images), point clouds can be difficult to learn due to irregularity.

To duplicate the success of convolutional networks on regular grids [24, 43], some prior works change point clouds into multi-view images [12, 10] or regular volumes [53, 12] before convolution. However, image-based projection and voxelization reduce the resolution of point clouds and result in the damage of internal geometric information. These explicit transformations also lead to complex preprocessing and significant computations.

PointNet [38] diverts the attention to the methods of processing raw point clouds. To handle irregular points, it adopts point-wise multi-layer perceptrons (MLP) to learn

on points independently and utilize a symmetric function to obtain the global information. For the ignorance of local structures, PointNet++ [40] further introduces *set abstraction* (SA) (shown in Fig. 1 left) as the local aggregator to build the hierarchical networks. However, this aggregator keeps learning on points independently, losing the sight of *shape awareness*.

When a local aggregator independently learns on points, the *shape ambiguity* problem has been exposed: since no points inside the set react with others, the aggregator will be sensitive to the coordinates $\mathcal{S} \in \mathbb{R}^{N \times 3}$ and be confused about the outline and the geometric information of the set. Here N is the number of points inside the set. The shape ambiguity problem causes the damage to the robustness and generalization of an aggregator.

In general, an excellent aggregator is underdeveloped for two reasons: it should discriminatively describe the *underlying shape* of point sets, and it should be robust to *rigid transformation* (i.e., translation, rotation) as well as noises. For a preliminary exploration, RS-CNN [33] computes a point feature from the aggregation of features weighted by predefined geometric relations (*low-level relation*) between the point \mathcal{S}_i and its neighbors $\mathcal{N}(\mathcal{S}_i)$ (shown in Fig. 1 middle). Based on the low-level relations instead of coordinates only, RS-CNN is insensitive to coordinates and robust to rigid transformation. However, RS-CNN is insufficient to learn semantic relations (*high-level relations*) for the lack of interaction between features.

In this case, self-attention [47] (shown in Fig. 1 right) may be a good instance as the supplement for high-level relations. Self-attention achieves great success on natural language processing. Recent work [64, 19] has shown that self-attention can be a viable alternative for convolution on images. However, self-attention can be depressing for *significant computations* as well as *a large number of parameters*. The goal of this work is to extend grid-based self-attention to irregular points with a high-efficiency strategy.

To this end, we propose *group relation aggregator* (GRA) to learn from both low-level and high-level relations. Compared with self-attention and SA, our designed bottle-

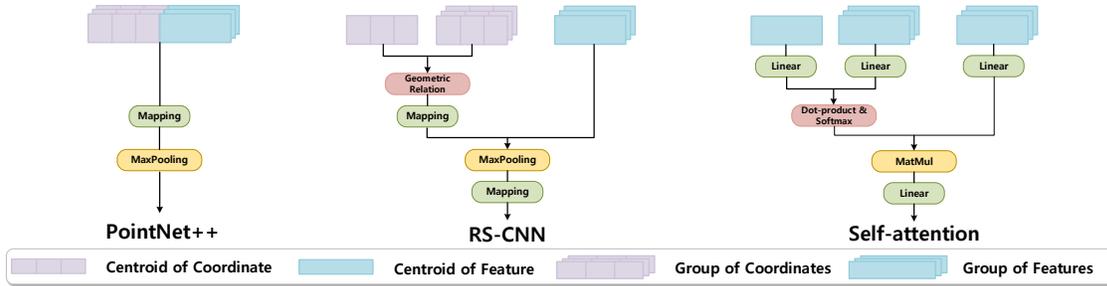


Figure 1. Comparison of some prior works. We input each module with one centroid (query) and its neighbors (keys) of features or coordinates. PointNet++ [40] (left) learns on points independently and lacks the interaction between points. RS-CNN [33] (middle) focuses on low-level relation learning, while self-attention (right) in Transformer [47] learns from high-level relations between features. Here the input group in self-attention is in the global scope.

neck version of GRA is obviously efficient in terms of computation and the number of parameters. With bottleneck GRA, we construct the efficient point-based networks **RPNet**.

Specifically, we construct each point set by taking a sampled point \mathcal{S}_i as the centroid and its neighbors $\mathcal{N}(\mathcal{S}_i)$, corresponding to features \mathcal{F}_i and $\mathcal{N}(\mathcal{F}_i)$ respectively. To exploit a point set, we force GRA to learn attention of the set from both predefined geometric priors (i.e., Euclidean distance between \mathcal{S}_i and $\mathcal{N}(\mathcal{S}_i)$) and feature-level interaction (i.e., mapping through a linear layer followed by matrix multiplication and scaled dot-product on \mathcal{S}_i and $\mathcal{N}(\mathcal{F}_i)$). By applying the attention to the transformed features $\mathcal{N}(\mathcal{F}_i)$ (i.e. mapping through a linear layer), the weighted features can reflect the geometric shape as well as semantic information of a point set. This proposed module benefits from the geometric priors in terms of shape awareness and robustness to rigid transformation, while the feature interaction enables the adaptation to content and the robustness to noises.

Considering the efficiency of GRA, we introduce the bottleneck concept to this module. The output of the first linear layer has the identical channels to the input. We cut down its output channels with a specific factor. Though this behavior harms the model quality (discussed in [47]), we add a mapping after the relation operation. Cross-channel attention also helps the module to explore channel-wise. All the changes turn our module into a high-efficiency version.

With our proposed module, we construct RPNet with respect to width (RPNet-W) and depth (RPNet-D). We then evaluate these two types of models on the datasets of classification (i.e., ModelNet40 [53]) and segmentation (i.e., ScanNet v2 [6], S3DIS [1]). The results show that our method outperforms point-based methods by a large margin, and even achieves comparable performance with all convolution-based methods in a state of high-efficiency. Interestingly, our model may obtain extra accuracy on classification by increasing the width, while deep model works

better than wide model on segmentation in terms of efficiency and accuracy.

Our key contributions are manifold:

- A novel scalable local aggregator for point clouds is proposed. It encodes the geometric and semantic relations between points;
- An expandable and high-efficiency hierarchy RPNet is proposed. Equipped with the bottleneck version of our aggregator, extensive RPNet keeps efficient;
- Experiments on the challenging benchmarks of classification and segmentation, indicate that RPNet achieves state of the arts.

2. Related Work

2.1. Learning on Point Clouds

Multi-view methods [10, 15, 54, 16, 39] describe a 3D object with multiple views from different viewpoints. Recent works have been proposed to recognize 3D shapes through convolutional neural networks, i.e., converting 3D shapes to 2D images [45] or lattice space [44]. However, this transformation results in the loss of shape information for self-occlusion, and a great number of views are required for decent performance. **Voxel-based methods** [53, 35, 12, 23, 5, 49, 42] apply volumetric CNN to recognize the 3D grids transformed from 3D shapes, i.e., the efficient submanifold sparse convolution [13]. The input 3D grid is limited to low resolution considering computational cost, leading to the loss of structural information. Different from multi-view and voxel-based methods, the goal of our work is to process point clouds directly.

Point-based methods [25, 11, 26, 34, 36, 22] have attracted great attention for processing on raw point clouds recently. PointNet [38] learns from global information through pointwise multi-layer perceptrons and max-pooling operation. PointNet++ [40] introduces set abstraction to

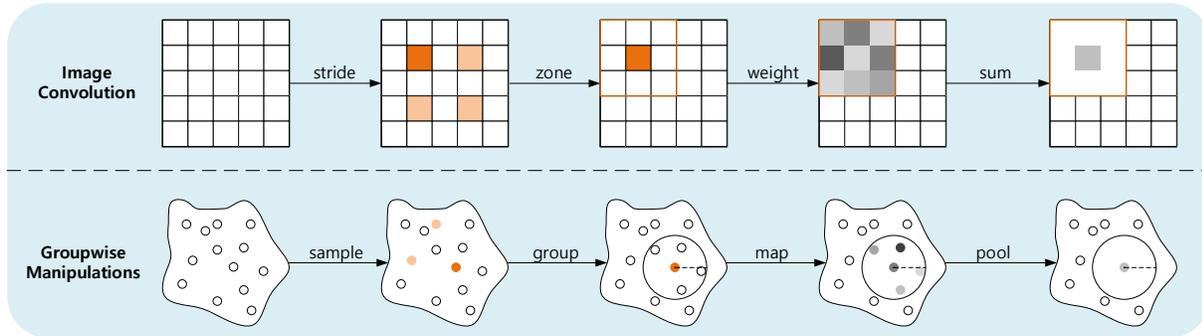


Figure 2. A simple analogy between image convolution and set abstraction.

capture local features, and farthest point sampling to down-sample uniformly between two set abstractions. Likewise, recent works concentrate on effective local learning approaches or various sampling manners. Point2Sequence [31] learns the information from different local regions by attention mechanism. ShufflePointNet [4] goes wider in an efficient manner via group convolution and channel shuffling. RandLA-Net [21] aggregates the local region with spatial encoding followed by attentive pooling. **Convolution-based methods** [57, 52, 17, 46, 33, 32, 63, 30, 29, 65, 50, 52] are another branch for local aggregation, using dynamic strategies of transformation to support the normal work of convolution on point clouds. PointCNN [28] applies traditional convolution on point clouds after transforming neighboring points to the canonical order. GridGCN [56] captures local geometry by graph instead of point set. However, these intuitively predefined transformations for the followed convolution operation may also cause a loss of structural information of the original point clouds, and the model may be sensitive to rigid transformation for convolution. In this paper, we focus on the relation learning based on MLP for a robust learning.

2.2. Relation Learning

Relation learning (i.e., self-attention [47]) has generally revolutionized natural language processing [8, 9]. It inspires applications in different computer vision fields, including image recognition [7, 48, 20, 19, 2, 41, 59, 64], image synthesis [61, 37], object detection [18, 14], and video understanding [51]. Wang *et al.* [51] uses non-local operation to model the relation between two pixels in an image, capturing long-range dependencies. DETR [3] adopts transformer encoder-decoder architecture for a competitive end-to-end detector.

Recent work proves the practicability of relation learning on point clouds. PointASNL [58] adopts the non-local operation to capture long-range dependencies of point clouds. ShapeContextNet [55] applies self-attention-like operator to

learn the global feature of a point cloud. Though effective, these methods focus on global relation learning, leading to loss of local information. RS-CNN [33] learns the relations within a local region by a predefined geometric priors, but the low-level relation cannot fully represent the relation between two points. In this paper, we aim to design a local aggregator to model the relation between two points on both geometric level and semantic level.

3. Relation Learning on Point Clouds

In this section, we first review PointNet++ [40], and the relation-based modules RS-CNN [33] and self-attention [47]. Next, we propose a general operator to learn inner-group relations and its instances. Then we design its bottleneck version as the building block, and finally, we implement the network architectures, Inner-Group Relation Point-based Networks (named **RPNet**) with this block.

3.1. Background

Most point-based blocks come from SA (shown in Fig. 1 left) in PointNet++ [40] to aggregate local features. It achieves point downsampling as well as feature transformation. Denote one input point as $x_i \in \mathbb{R}^{3 \times N}$, its neighbors as $x_{i.}$, its feature as $\mathbf{f}(x_i) \in \mathbb{R}^{C \times N}$ and its coordinate as $\mathbf{p}(x_i) \in \mathbb{R}^{3 \times N}$, with C being the channels of input point features and N the number of input points. Specifically, this layer transforms the group of feature points $\mathbf{f}(x_{i.})$ via pointwise multi-layer perceptrons followed by max-pooling after point sampling and grouping. For an intuitive presentation, we show an analogy between the operations inside image convolution and SA in Fig. 2. SA without sampling can be formulated as follows:

$$\mathbf{y}(x_i) = \mathcal{A}(\{\mathcal{M}(x_{ij}), \forall x_{ij} \in \mathcal{G}(x_i)\}), \quad (1)$$

where \mathcal{A} is the aggregation function (i.e., max-pooling), \mathcal{M} is the mapping function, and \mathcal{G} is the grouping method (i.e.,

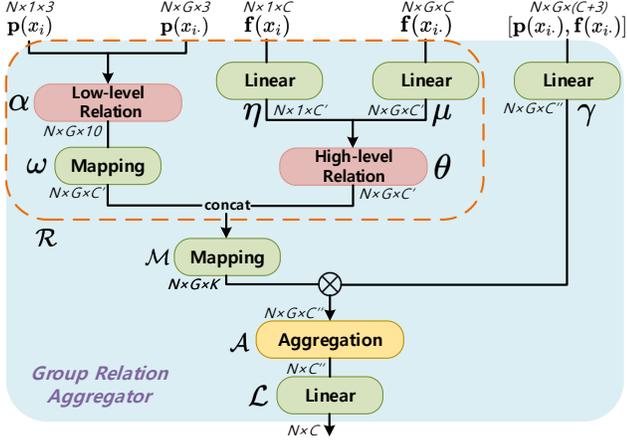


Figure 3. Our group relation aggregator. After sampling and grouping operations on one point cloud, we input features and coordinates of the centroids $\mathbf{f}(x_i)$, $\mathbf{p}(x_i)$ and their neighbors $\mathbf{f}(x_i)$, $\mathbf{p}(x_i)$. There are two parts in the relation function \mathcal{R} , α for the geometric relation as well as θ for the semantic relation. We apply cross-channel attention (denoted as \otimes in the above figure) to the output of \mathcal{M} and the γ . Here we introduce the cross-channel attention, which segments the output of γ to K groups with size $N \times G \times (C''/K)$, and we apply one of K weight map with size $N \times G$ to each of the (C''/K) channels of the corresponding group via element-wise product. The outputs of cross-channel attention and γ have the same shape. Finally, we aggregate the output followed by a linear function.

kNN, Ball Query [40]). Here we adopt farthest point sampling as the default sampling method.

SA is popular for its simplicity, but it suffers from shape ambiguity as it learns from features or coordinates independently. In this way, RS-CNN [33] tries to solve this problem. Relation-shape convolution is of shape-awareness for the predefined geometric relation. As shown in Fig. 1 middle, the relation-shape convolution adaptively aggregates the key contents according to the weight from the predefined function \mathcal{R}_l . For a more powerful shape-aware representation, it further applies a channel-raising mapping \mathcal{L}_{cr} after the weighted features. Relation-shape convolution can be formulated as:

$$\mathbf{y}(x_i) = \mathcal{L}_{cr}(\mathcal{A}(\{\mathcal{R}_l(x_i, x_{ij}) \cdot \mathbf{f}(x_{ij}), \forall x_{ij} \in \mathcal{G}(x_i)\})). \quad (2)$$

Relation-shape convolution is good practice to learn geometric relations on point clouds, but semantic level relations may be ignored. In this case, self-attention operation can be an inspiration to complement the semantic level relation learning on point clouds. Recently, self-attention is proved to be effective in computer vision. Similar to relation-shape convolution, self-attention (shown in Fig. 1 right) also learns to aggregate the key contents according to the weight (or compatibility) of query-key pairs. However,

differently, the weight is implicitly obtained by the interaction of one query and its keys in a high-level space instead of 3D space. The relation function is defined as \mathcal{R}_h (i.e., scaled dot-product). \mathcal{L} is a linear function after aggregation. Self-attention operator can be formulated as:

$$\mathbf{y}(x_i) = \mathcal{L}(\mathcal{A}(\{\mathcal{R}_h(x_i, x_{ij}) \cdot \beta(x_{ij}), \forall x_{ij} \in \mathcal{G}(x_i)\})). \quad (3)$$

3.2. Inner-Group Relation Learning

We explore a local aggregator to learn from both geometric relations and semantic relations inside a point set. Thus we propose group relation aggregator (shown in Fig. 3), which has the following form:

$$\mathbf{y}(x_i) = \mathcal{L}(\mathcal{A}(\{\mathcal{H}(x_i, x_{ij}), \forall x_{ij} \in \mathcal{G}(x_i)\})), \quad (4)$$

where the inner-group relation function \mathcal{H} equipped with cross-channel attention is defined as:

$$\mathcal{H}(x_i, x_{ij}) = \mathcal{M}(\mathcal{R}(x_i, x_{ij})) \otimes \gamma(x_{ij}), \quad (5)$$

where \otimes means cross-channel attention. Denote the number of attention maps as K to enable the operation of cross-channel attention, the total number of centroids as N , each centroid with G neighbors, the feature dimension as C . $C' = C/r_1$ and $C'' = C/r_2$ (details in Sec. 3.4). We found our introduction of cross-channel attention beneficial by allowing the model to attend to information from different representation subspaces. Cross-channel attention first segments the output of γ to K groups with size $N \times G \times (C''/K)$. For each of K weight map obtained from \mathcal{M} , it then applies the map with size $N \times G$ to every channel (totally C''/K channels) of the corresponding group by element-wise product. The number of channels in the output of γ is consistent after the operation of cross-channel attention. Cross-channel attention will degenerate into vanilla attention when $K = 1$. The operation of cross-channel attention can further allow the design of bottleneck, significantly reducing the number of parameters and the computations. \mathcal{M} a combination of both linear functions and non-linearity functions, i.e., $\{MLP \rightarrow ReLU \rightarrow MLP\}$. It allows us to introduce additional trainable transformations for more expressive construction of the weights. The output dimensionality of ω does not need to match that of γ as the attention weights are shared across a group of channels for cross-channel attention. The function γ is a linear function here. The relation function \mathcal{R} contains a geometric function $\alpha(\cdot, \cdot)$ followed by a mapping function ω as well as a semantic function $\theta(\cdot, \cdot)$:

$$\mathcal{R}(x_i, x_{ij}) = [\omega(\alpha(x_i, x_{ij})), \theta(x_i, x_{ij})], \quad (6)$$

where ω is a sequence of mapping operations, and α can be defined like this:

$$\alpha(x_i, x_{ij}) = [||\mathbf{p}(x_i) - \mathbf{p}(x_{ij})||, \mathbf{p}(x_i), \mathbf{p}(x_{ij}), \mathbf{p}(x_i) - \mathbf{p}(x_{ij})]. \quad (7)$$

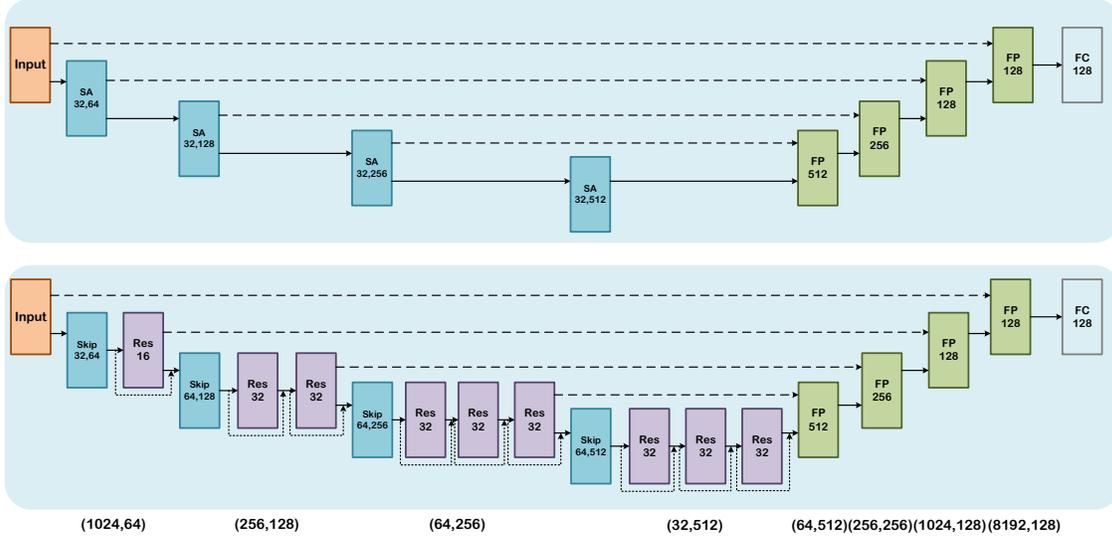


Figure 4. Comparison of architectures of PointNet++ (above) and our RPNNet-D14 (below) for segmentation task. ‘Skip’ and ‘Res’ represent ‘Skip Block’ and ‘Residual Block’ respectively. The two blocks are based on our GRA. Skip block is a GRA combined with down-sampling, while residual block has a branch of residual link. Each skip block groups with a specific scale and outputs the features with a fixed dimension, while each residual block enhances the features through a fixed scale of groups. For example, the first skip block in RPNNet-D14 groups 32 points corresponding to the center points and outputs 64 dimension vectors. The following residual block groups 16 points per center point.

We explore possible instantiations of θ , along with feature transformation elements that surround self-attention operations in our architecture:

Concatenation: $\theta(x_i, x_{ij}) = [\eta(\mathbf{f}(x_i)), \mu(\mathbf{f}(x_{ij}))]$

Summation: $\theta(x_i, x_{ij}) = \eta(\mathbf{f}(x_i)) + \mu(\mathbf{f}(x_{ij}))$

Subtraction: $\theta(x_i, x_{ij}) = \eta(\mathbf{f}(x_i)) - \mu(\mathbf{f}(x_{ij}))$

Hadamard product: $\theta(x_i, x_{ij}) = \eta(\mathbf{f}(x_i)) \odot \mu(\mathbf{f}(x_{ij}))$

Here η and μ are trainable transformations such as linear mappings, and have matching output dimensionality.

3.3. Bottleneck Improves Efficiency

Matrix multiplication in self-attention brings about significant computations. The complexity of GRA is of $\mathcal{O}(C^2)$. To design an efficient aggregator, we introduce the philosophy of bottleneck to GRA. Denote the channel dimensionality of input features by C . The output of η and μ have C/r_1 channels. The output of γ and \mathcal{H} have the same dimension C/r_2 . The output of the block is subsequently expanded back to C through a linear mapping. In our architectures, we set $r_1 = 16$ and $r_2 = 4$.

3.4. RPNNet

The main structures of our RPNNet generally follow PointNet++ [40], which we use as our baseline. The number X in RPNNet-W X and RPNNet-D X refers to the number of our GRA blocks.

Classification. RPNNet-W consists of GRA only. The backbone of RPNNet-W has three stages, each with different spatial resolution. Every stage comprises multiple self-attention blocks. In RPNNet-W7, grouping of the first two stages are performed by multi-scale groupers, with the sizes of $\{16, 32, 128\}$ and $\{32, 64, 128\}$ in order. RPNNet-W9 and RPNNet-W15 adopt more detailed scales within $[16, 128]$ and $[32, 128]$. The third stage groups all the rest points for aggregation. The output of the third stage is processed by a classification layer that comprises three linear layers and dropout with a ratio of 0.5 between two of the layers, followed by a softmax activation.

Segmentation. We build RPNNet-D with skip block (GRA with down-sampling) and residual block (GRA with a residual link). The input of RPNNet-D is 8k or 16k points containing various information (i.e. coordinates, color and normal). RPNNet-D first encodes a point cloud by downsampling points, i.e. $\{8192 \rightarrow 1024 \rightarrow 256 \rightarrow 64 \rightarrow 32\}$, and then decodes by upsampling through feature propagation [40]. The residual GRA blocks are attached to the down-sampling or upsampling blocks. The segmentation layer includes a linear layer and a dropout layer of 0.5. An illustration (Fig. 4) shows the architecture of RPNNet-D14.

4. Experiments

First, we evaluate our RPNNet-W and RPNNet-D on the classification and segmentation tasks of various datasets, in-

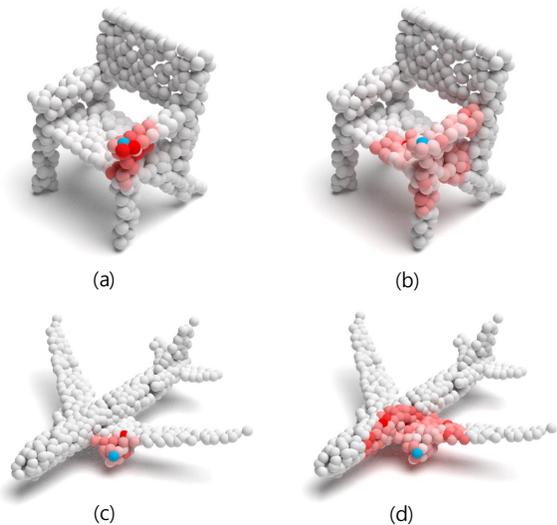


Figure 5. We visualize the attention weights of a chair (a)(b) and an airplane (c)(d) from our GRA in different scales. Blue balls stand for the center (query) points. By visualizing, we can find that the bounding points obtain higher importance. We argue that GRA describes a shape representation by the bounding points to a large extent. Intuitively, it works on human eyes as well. We discriminate a shape by its outline most of the time.

cluding synthetic datasets and scene segmentation datasets. Furthermore, we discuss the expandability of our RPNNet. We then analyze the robustness of our models in terms of rigid transformation and noises. The following ablation studies explore the variants of our module and assess the effectiveness of the components used to construct the networks. All the experiments are performed on a machine with four V100 GPUs.

4.1. Evaluation on Classification

We conduct experiments on ModelNet40 [53] on classification through our RPNNet-W network. The dataset contains 9843 training point clouds and 2468 test ones from 40 different categories.

Implementation. Our implementation mainly follows the practice in [38]. For training, we first select 1024 points as input. To prevent overfitting, we apply augmentation strategy including the following components: random scaling in the range [0.8, 1.25], random shift in the range $[-0.1, 0.1]$, random dropout points with the ratio of the range [0, 87.5%]. The initial learning rate is 0.001, and it decays by a factor of 0.7 every 20 epochs. For testing, similar to [38, 40], we average the predictions of randomly scaled inputs.

Results. In Tab. 1, we compare our RPNNet-W with state-of-the-art classification methods on ModelNet40. Among all current methods, our method achieves state-of-the-art

Method	Modality	Accuracy(%)
PointGCN [62]	Graph	89.5
KPConv [46]	Grid	92.9
SO-Net [27]	Points+Normals (5k)	93.4
PVRNet [60]	Points+Views	93.6
RS-CNN [33] w/ vot.	Points	93.6
PointNet [38]	Points	89.2
RS-CNN [33] w/o vot.	Points	92.4
PointASNL [58]	Points	92.9
RPNNet-W7 (ours)	Points	93.8 $\uparrow 0.9$
RPNNet-W9 (ours)	Points	93.9 $\uparrow 1.0$
PointNet++ [40]	Points+Normals	91.9
FPCov [29]	Points+Normals	92.5
Grid-GCN [56]	Points+Normals	93.1
PointASNL [58]	Points+Normals	93.2
RPNNet-W7 (ours)	Points+Normals	93.9 $\uparrow 0.7$
RPNNet-W9 (ours)	Points+Normals	94.1 $\uparrow 0.9$

Table 1. Performance of classification on ModelNet40 on accuracy(%).

Method	S3DIS-6	ScanNet
<i>Convolution-based Methods</i>		
PointCNN [28]	65.4	45.8
FPCov [29]	68.7	63.9
KPConv [46]	70.6	68.4
<i>MLP-based Methods</i>		
RandLA [21] (10^5)	70.0	-
PointNet++ [40]	53.4	33.9
PointWeb [65]	66.7	-
PointASNL [58]	68.7	63.0
RPNNet-D8 (ours)	69.1	67.1
RPNNet-D14 (ours)	70.0	67.7
RPNNet-D27 (ours)	70.8 $\uparrow 2.1$	68.2 $\uparrow 5.2$

Table 2. Mean per-class IoU(%) for the task of semantic segmentation on the datasets of ScanNet v2 and S3DIS (6-fold cross validation). “-” means unknown.

with a promotion of 0.9%. We also compare our RPNNet-W7 with RS-CNN and PointASNL with and without normal input. Besides, we visualize the attention maps in Fig. 5.

4.2. Evaluation on Segmentation

Large-scale scene segmentation is a more challenging task due to outliers and noises. We evaluate our RPNNet-D on Stanford 3D Large-Scale Indoor Spaces (*S3DIS*) [1] and ScanNet v2 (*ScanNet*) [6] datasets. *S3DIS* contains 271 scenes from six zones. It provides 13 types of semantic labels for scene segmentation. *ScanNet* includes 1513 training point clouds and 100 test ones. It marks each point from 21 categories.

Implementation. On both datasets, we verify each method with mean per-class IoU (mIoU), and use point position and RGB information as input. In particular, we evaluate models with 6-fold cross-validation over all six zones (6-fold) on *S3DIS*. For training, we randomly sample 16384

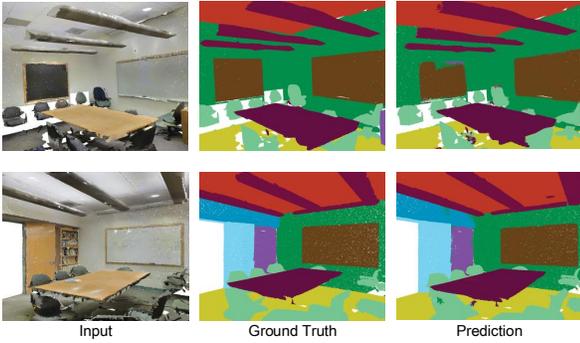


Figure 6. Examples of semantic scene labeling with RPNNet.

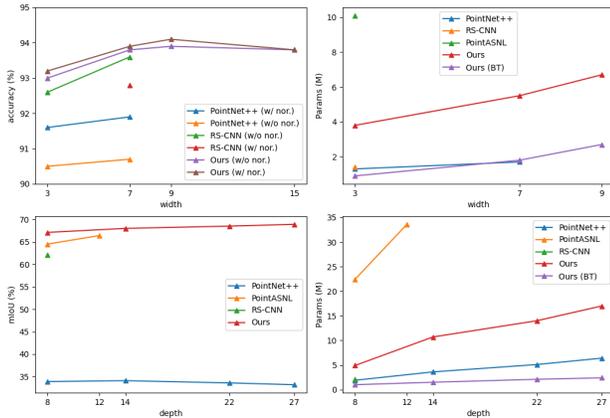


Figure 7. Model comparison in terms of width on classification (above) and depth on segmentation (below). We compare models by both accuracy or mIoU (left) and the number of parameters (right). Here “nor.” means normals, “BT” means bottleneck version.

points from the scenes. For evaluation, similar to [58], we obtain an average prediction of 5 votes by sliding a window across the room in 0.5m stride.

Results. In Tab. 2, we list the latest methods to compare with our RPNNet-D, i.e., PointASNL [58]. We adopt the same training and test approach by randomly chopping cubes with a fixed number of points. We show the results of convolution-based methods as well, i.e., FPconv [29], KPconv [46]. All of these methods utilize the raw point clouds as the input. An illustration of semantic scene labeling is shown in Fig. 6.

4.3. Discussion of Expandability

As show in Fig. 7, we evaluate various models with respect to accuracy and the number of parameters. With the same depth or width, our model outperforms SOTAs by an obvious margin. Besides, the efficiency of our model is also competitive compared with prior works.

Shown in Fig. 7 above, we compare our RPNNet-W with

PointNet++ multi-scale grouping (MSG) version. To improve efficiency, we use a bottleneck architecture to build blocks. These blocks significantly reduce the computational cost while maintaining high accuracy. By increasing the width, our model can obtain more geometric information from a point cloud, and thus higher accuracy. The comparison of RPNNet-W9 and RPNNet-W15 shows a overfitting problem. We argue that the width of 9 is enough for the full exploitation. Wider model would bring much redundant information and cause the decrease of performance.

Shown in Fig. 7 below, similar to [26], we deepen our models on segmentation. We compare the efficiency improvement of RPNNet-D with our bottleneck blocks. To verify the effectiveness of greater depth, we gradually increase the depth of RPNNet-D on ScanNet. Obviously, as the depth increases, the model can obtain higher mIoU. We argue that relation learning helps to deeply extract representations by stacking residual blocks. We also obtain a similar conclusion from [26] that going deeper can help improve the accuracy on segmentation.

To explore which kind of model works better on the two tasks, we conduct simple experiments on deeper models for classification and wider models for segmentation. First, we test deeper model using RPNNet-W1 on classification:

Depth	3	5	7
acc.	92.9	92.5 ↓0.4	92.1 ↓0.8

Shown in the table, the results perform worse with the depth increasing. We discuss that the task of classification concentrates on the global view instead of point-wise recognition. Increasing the depth could not be beneficial to obtain such global view. Besides, there would be a great number of redundant features inside each layer. The errors of a global view would increase significantly if the depth is over the threshold value.

Also, we test wider RPNNet-D4 (RPNNet-D without residual block) on segmentation:

Width	1	2	3
acc.	66.8	67.0 ↑0.2	67.1 ↑0.3

Shown in the table, increasing the width leads to almost no improvement, but with nearly X -fold increasing on the number of parameters and the computation. Through the empirical results as well as our discussion, we conclude that RPNNet-W fits for classification, while RPNNet-D works better on segmentation.

4.4. Ablation Study

We conduct ablation studies to evaluate the effectiveness of our design. We mainly discuss some key components of our GRA, including inner-group relation function, aggregation function and cross-channel attention. We perform all ablation studies on S3DIS with 6-fold validation.

Inner-group relation function \mathcal{H} . Shown in Tab. 3, we

Model	Geometric Relation α				Semantic Relation θ				mIoU (%)
	ℓ_2	ℓ_1	$x_i - x_{ij}$	$[x_i, x_{ij}]$	sum	sub	cat	had	
A	✓								63.7
B	✓	✓							63.8
C	✓	✓	✓	✓					64.2
D	✓	✓	✓	✓	✓				67.8
E	✓	✓	✓	✓		✓			67.8
F	✓	✓	✓	✓			✓		67.5
G	✓	✓	✓	✓				✓	67.6

Table 3. The results of different designs on inner-group relation function \mathcal{H} (sum: summation, sub: subtraction, cat: concatenation, had: Hadamard product). The experiments are on S3DIS with 6-fold validation.

ablate the designs of geometric and semantic relation functions α and θ in details. We define the geometric priors as four possible components: ℓ_2 , ℓ_1 , $x_i - x_{ij}$ and $[x_i, x_{ij}]$. ℓ_2 and ℓ_1 can directly describe the distances between two points, while $x_i - x_{ij}$ and $[x_i, x_{ij}]$ show the relative and global positions of two points. Model C outperforms model A and B, which shows that four components of geometric priors boost the performance of our RPNets at the same time. Furthermore, we survey the designs of semantic relation function θ with four possibilities: summation, subtraction, concatenation and Hadamard product. The results prove that summation or subtraction works better in RPNets. Summation or subtraction would be better in terms of computation as well.

Aggregation function A. We adopt three types of symmetric function to aggregate in our GRA: max-pooling, avg-pooling and sum-pooling. Here is the results:

RPNets-D8	max-pooling	avg-pooling	sum-pooling
acc.	67.8	67.5	67.7

The table shows that max-pooling performs better than the others. We argue that max-pooling can filter the redundant information in our GRA and select the expressive features.

Cross-channel attention. We test the design of our cross-channel attention below:

RPNets-D8	vanilla att.	cross-channel att.
acc.	67.8	69.1 $\uparrow 1.3$

As shown in the table, cross-channel attention greatly boosts our GRA by 1.3% on S3DIS. Vanilla attention does not enable the channelwise exploration. However, different channels may play different roles to influence the final weights. One channel attention map cannot fully utilize the feature space, especially when the space has a great number of dimensions. To handle this problem, cross-channel attention applies multiple attention maps to different groups of channels, allowing the channelwise exploration in GRA.

4.5. Analysis of Robustness

Robustness to rigid transformation. We evaluate the robustness to rigid transformation for the comparison of our RPNets with PointNet [38], PointNet++ [40], RS-CNN

model	origin	perm	± 0.2	90°	180°	270°
PointNet [38]	88.7	88.7	70.8	42.5	38.6	40.7
PointNet++ [40]	88.2	88.2	88.2	88.2	47.9	39.7
RS-CNN [33]	90.3	90.3	90.3	90.3	90.3	90.3
RPNets-W7 (ours)	90.9	90.9	90.9	90.9	90.9	90.9

Table 4. Robustness to point permutation and rigid transformation (%). We perform the operations of random permutation (perm), translation of ± 0.2 , and clockwise rotation around Y axis.

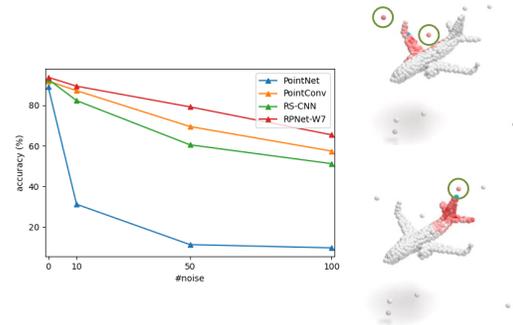


Figure 8. (Left) Classification results of different models with noises. (Right) By learning semantic relation, the local aggregator is relatively insensitive to noises, concentrating more on the shape instead of independent points.

[33]. We follow the experimental setup of RS-CNN for the evaluation. As shown in Tab. 4, all these coordinate-based methods are insensitive to permutation thanks to the design of symmetric function. However, PointNet is sensitive to translation and rotation, while PointNet++ is vulnerable to rotation. RS-CNN and our RPNets perform robust due to the usage of relation learning. Our RPNets outperforms RS-CNN for the learning on high-level relation.

Robustness to noises. We also evaluate our networks on robustness to noises. Shown in Fig. 8, our RPNets outperforms other competitive methods with the same noises input. Note that RS-CNN is sensitive to noise. We argue that the predefined relation may affect the output of attention map, causing the wrong relation representation. However, our RPNets uses semantic level relation, which is skilled at denoising and focusing on the content.

5. Conclusion

We present group relation aggregator as well as deeper (RPNets-D) and wider (RPNets-W) models for efficient point cloud analysis. By learning from both geometric and semantic relations inside a point set, our RPNets achieves state-of-the-art on both classification and segmentation tasks. We further introduce bottleneck philosophy to our module for high efficiency. Experiments based on challenging benchmarks illustrate the effectiveness of our RPNets.

References

- [1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543, 2016. 2, 6
- [2] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3286–3295, 2019. 3
- [3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020. 3
- [4] Can Chen, Luca Zanotti Fragonara, and Antonios Tsourdos. Go wider: An efficient neural network for point cloud analysis via group convolutions. *Applied Sciences*, 10(7):2391, 2020. 3
- [5] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 2
- [6] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 2, 6
- [7] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 3
- [8] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019. 3
- [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3
- [10] Yifan Feng, Zizhao Zhang, Xibin Zhao, Rongrong Ji, and Yue Gao. Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 264–272, 2018. 1, 2
- [11] Kent Fujiwara and Taiichi Hashimoto. Neural implicit embedding for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11734–11743, 2020. 2
- [12] Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3d point cloud processing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–118, 2018. 1, 2
- [13] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232, 2018. 2
- [14] Jiayuan Gu, Han Hu, Liwei Wang, Yichen Wei, and Jifeng Dai. Learning region features for object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 381–395, 2018. 3
- [15] Haiyun Guo, Jinqiao Wang, Yue Gao, Jianqiang Li, and Hanqing Lu. Multi-view 3d object retrieval with deep embedding network. *IEEE Transactions on Image Processing*, 25(12):5526–5537, 2016. 2
- [16] Zhizhong Han, Mingyang Shang, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, Junwei Han, and CL Philip Chen. Seqviews2seqlabels: Learning 3d global features via aggregating sequential views by rnn with attention. *IEEE Transactions on Image Processing*, 28(2):658–672, 2018. 2
- [17] Pedro Hermosilla, Tobias Ritschel, Pere-Pau Vázquez, Àlvar Vinacua, and Timo Ropinski. Monte carlo convolution for learning on non-uniformly sampled point clouds. *ACM Transactions on Graphics (TOG)*, 37(6):1–12, 2018. 3
- [18] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018. 3
- [19] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3464–3473, 2019. 1, 3
- [20] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 3
- [21] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11108–11117, 2020. 3, 6
- [22] Li Jiang, Hengshuang Zhao, Shu Liu, Xiaoyong Shen, Chi-Wing Fu, and Jiaya Jia. Hierarchical point-edge interaction network for point cloud semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10433–10441, 2019. 2
- [23] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 863–872, 2017. 2
- [24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 1
- [25] Itai Lang, Asaf Manor, and Shai Avidan. Samplenet: Differentiable point cloud sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7578–7588, 2020. 2
- [26] Eric-Tuan Le, Iasonas Kokkinos, and Niloy J Mitra. Going deeper with lean point networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9503–9512, 2020. 2, 7

- [27] Jiaxin Li, Ben M Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9397–9406, 2018. 6
- [28] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointnet: Convolution on x-transformed points. In *Advances in neural information processing systems*, pages 820–830, 2018. 3, 6
- [29] Yiqun Lin, Zizheng Yan, Haibin Huang, Dong Du, Ligang Liu, Shuguang Cui, and Xiaoguang Han. Fpconv: Learning local flattening for point convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4293–4302, 2020. 3, 6, 7
- [30] Zhi-Hao Lin, Sheng-Yu Huang, and Yu-Chiang Frank Wang. Convolution in the cloud: Learning deformable kernels in 3d graph convolution networks for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1800–1809, 2020. 3
- [31] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. 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, pages 8778–8785, 2019. 3
- [32] Yongcheng Liu, Bin Fan, Gaofeng Meng, Jiwen Lu, Shiming Xiang, and Chunhong Pan. Densepoint: Learning densely contextual representation for efficient point cloud processing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5239–5248, 2019. 3
- [33] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019. 1, 2, 3, 4, 6, 8
- [34] Ze Liu, Han Hu, Yue Cao, Zheng Zhang, and Xin Tong. A closer look at local aggregation operators in point cloud analysis. In *European Conference on Computer Vision*, pages 326–342. Springer, 2020. 2
- [35] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE, 2015. 2
- [36] Ehsan Nezhadarya, Ehsan Taghavi, Ryan Razani, Bingbing Liu, and Jun Luo. Adaptive hierarchical down-sampling for point cloud classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12956–12964, 2020. 2
- [37] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. *arXiv preprint arXiv:1802.05751*, 2018. 3
- [38] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1, 2, 6, 8
- [39] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016. 2
- [40] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 1, 2, 3, 4, 5, 6, 8
- [41] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019. 3
- [42] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586, 2017. 2
- [43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1
- [44] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2530–2539, 2018. 2
- [45] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015. 2
- [46] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6411–6420, 2019. 3, 6, 7
- [47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1, 2, 3
- [48] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2017. 3
- [49] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics (TOG)*, 36(4):1–11, 2017. 2
- [50] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2589–2597, 2018. 3
- [51] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 3

- [52] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 3
- [53] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 1, 2, 6
- [54] Jin Xie, Guoxian Dai, Fan Zhu, Edward K Wong, and Yi Fang. Deepshape: Deep-learned shape descriptor for 3d shape retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1335–1345, 2016. 2
- [55] Saining Xie, Sainan Liu, Zeyu Chen, and Zhuowen Tu. Attentional shapecontextnet for point cloud recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4606–4615, 2018. 3
- [56] Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang, and Ulrich Neumann. Grid-gcn for fast and scalable point cloud learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5661–5670, 2020. 3, 6
- [57] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 87–102, 2018. 3
- [58] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5589–5598, 2020. 3, 6, 7
- [59] Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. Disentangled non-local neural networks. *arXiv preprint arXiv:2006.06668*, 2020. 3
- [60] Haoxuan You, Yifan Feng, Xibin Zhao, Changqing Zou, Rongrong Ji, and Yue Gao. Pvrnet: Point-view relation neural network for 3d shape recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9119–9126, 2019. 6
- [61] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363. PMLR, 2019. 3
- [62] Yingxue Zhang and Michael Rabbat. A graph-cnn for 3d point cloud classification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6279–6283. IEEE, 2018. 6
- [63] Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1607–1616, 2019. 3
- [64] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10076–10085, 2020. 1, 3
- [65] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. Pointweb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5565–5573, 2019. 3, 6