Deeper or Wider Networks of Point Clouds with Self-attention?

Haoxi Ran* Li Lu
Sichuan University
ranhaoxi@gmail.com
luli@scu.edu.cn

Abstract

*Prevalence of deeper networks driven by self-attention is in stark contrast to underexplored point-based methods. In this paper, we propose groupwise self-attention as the basic block to construct our network—**SepNet**. Our proposed module can effectively capture both local and global dependencies. This module computes the features of a group based on the summation of the weighted features of any point within the group. For convenience, we generalize groupwise operations to assemble this module. To further facilitate our networks, we deepen and widen SepNet on the tasks of segmentation and classification respectively, and verify its practicality. Specifically, SepNet achieves state-of-the-art for the tasks of classification and segmentation on most of the datasets. We show empirical evidence that SepNet can obtain extra accuracy in classification or segmentation from increased width or depth, respectively.*

1. Introduction

3D vision has attracted considerable attention for its advantages in various applications, including autonomous driving [47], robotics [35], and augmented reality [26]. Among 3D representatives, point clouds play a vital role in accessibility and flexibility. Unlike other visual elements (i.e., images, videos, and volumes), point clouds are difficult to learn due to randomness and sparsity. In this paper, we concentrate on point cloud processing for the tasks of shape classification and scene segmentation.

Deeper and wider models are broadly applied in machine learning, which typically leads to a significant revolution in a field. ResNet [10] enables deeper models for image-based tasks. Inception-v4 [32] expands the receptive field through multi-scale convolution. Multi-layer Transformers [34] and BERT [8] inspire the idea of deeper pre-training models in natural language processing. Recent work [52, 13] shows that self-attention can be a candidate for convolution to build a deeper model on image recognition. As aforementioned, deeper models make major breakthroughs in both

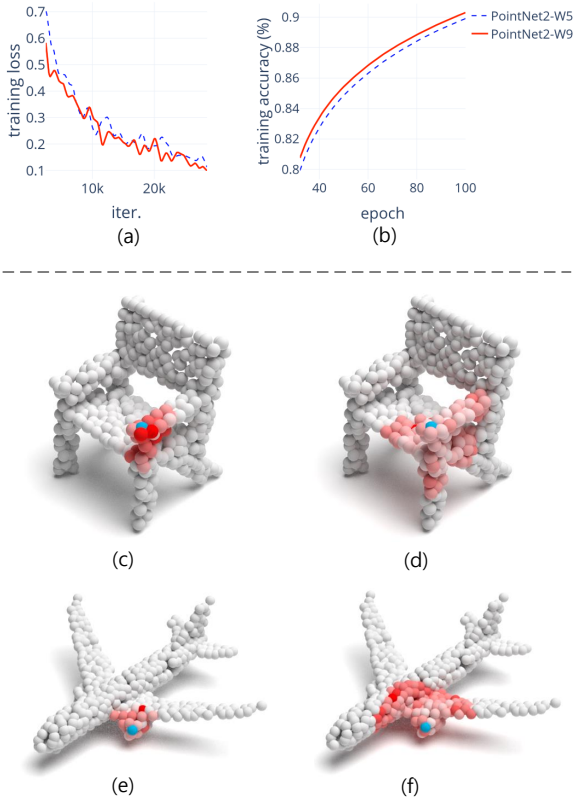

Figure 1. Performance of wider model on ModelNet40 and visualization of our groupwise self-attention. Wider model (PointNet2 with the width of 9) has lower training loss (a) and higher training accuracy (b) than its counterpart (PointNet2 with the width of 5) for shape classification. For test, wider model performs better as well. We visualize the attention weights of a chair (c)(d) and an airplane (e)(f) with different scales of grouping. Blue balls stand for the center points as well as the query points.

*This work is done when Haoxi Ran is a research assistant in Sichuan University.

accuracy and efficiency in each field. Although attractive, how to use the paradigms of deeper and wider models for point cloud understanding is still underutilized.

Previous point-based methods [28, 29, 45, 23, 33, 40, 22] focus more on local feature aggregation. They invent variations of *set abstraction* (SA) from PointNet++ [29] or convolution-like local transformations [45, 23, 33, 40, 22] to extract detailed information. Based on these approaches, neural networks can learn a better point set representation. *Feature Propagation* (FP) decodes the representation for segmentation task. However, due to the lack of attention to global dependencies, point-based methods are not fully developed.

Our Contributions. Distinguished from current methods, we propose a novel network for precise point cloud processing, called **SepNet**. We observed that more points mean a more powerful depiction of shapes. Based on this observation, we regard the self-attention module as the basic block of the network. Unlike previous local aggregation methods, our self-attention module is capable of handling relatively large point sets and learning both local and global dependencies.

Simple self-attention [34] or non-local [38] operations (applied in PointASNL [45]) are incompetent on point clouds since local aggregation requires multi-scale point sets to enhance the representation. Inspired by recent exploration of convolution-like self-attention [52] based on image, we invent **groupwise self-attention** to deal with different sizes of groups flexibly and to learn short-range and long-range dependencies.

In order to make better use of point sets, we deepen and widen our SepNet to explore the potential of the two extensive models. For a wider model, we use multi-scale groupwise self-attention to build each layer. We replace the residual link with a shortcut connection (multi-layer perceptrons followed by max-pooling). For a deeper model, we retain the original operations of a normal self-attention block. For the channelwise exploration, we also apply vector attention to our self-attention modules.

Empirically, a wider or multi-scale model is generally effective in the task of shape classification, while a deeper model works better on the segmentation datasets. With this discovery, we build wider SepNet (named **SepNet-W**) to recognize 3D shapes and deeper SepNet (named **SepNet-D**) for semantic segmentation.

We evaluate our models on the classification (i.e., ModelNet40 [41]) and segmentation datasets (i.e., ScanNet v2 [5], S3DIS [1]), which shows great development on all these datasets. Comparing models with different widths or depths, we prove the nationality of deeper and wider models on point clouds. Our bottleneck self-attention blocks further improve the efficiency of SepNet without accuracy decline.

Our contributions can be summarized as follows:

- We propose a novel network SepNet for effective point cloud processing. Our groupwise self-attention module is competitive to learn short-range and long-range dependencies.
- We design two types of basic building blocks, skip block and residual block, to construct our SepNet. We sketch the geometry through the skip connection inside the skip block, and retrieve more precise features with our self-attention module. Our designed bottleneck blocks further improve the efficiency.
- Based on our designed blocks, we build wider models (SepNet-W) and deeper models (SepNet-D) for the tasks of classification and segmentation. Experimental results show that our proposed SepNet-W and SepNet-D outperform prior point-based methods for both classification and segmentation, respectively.

2. Related Work

2.1. Learning on Point Clouds

Point-based learning methods have recently attracted great attention, with several pioneering methods proposed, including PointNet [28] and its extensive work PointNet++ [29]. PointNet learns from global information through pointwise multi-layer perceptrons and max-pooling operation. PointNet++ introduces set abstraction, a convolution-like module, to capture local features, and farthest point sampling to downsample uniformly between two set abstractions. Likewise, recent work concentrates on effective local learning approaches or various sampling manners. PointCNN [22] applies traditional convolution on point clouds after transforming neighboring points to the canonical order. PointASNL [45] effectively handles the outliers of point clouds by re-weighting the neighboring points and adjusting the sampled points, including non-local operation [38] to capture long-range dependencies of point clouds. SampleNet [19] learns a differentiable sampler for task-specific improvements. Le *et al*. [20] introduces a memory-saving module for a deeper model to focus on regional features. Point2Sequence [24] learns the information from different local regions by attention mechanism. Other recent approaches [44, 40, 11, 33, 25, 37, 23] use dynamic strategies to support the normal work of convolution on point clouds. FPConv [23] learns a weight map to softly flatten local surface for succedent convolution. PointConv [40] and PCCN [37] project the relative position of two points to a convolution weight, and PointWeb [53] enhances the regional feature by connecting every point pairs within a local region. PointGNN [31] and Grid-GCN [43] capture local geometry by graph instead of point set. On the

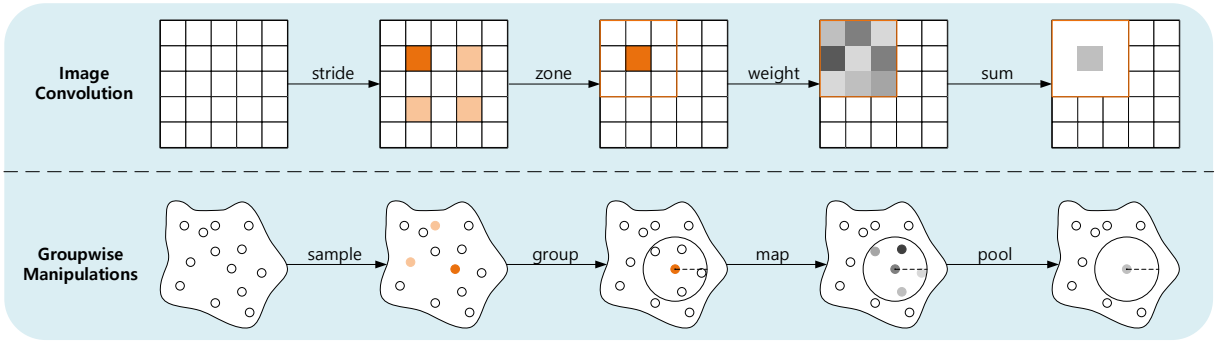

Figure 2. Analogy between image convolution and groupwise manipulations.

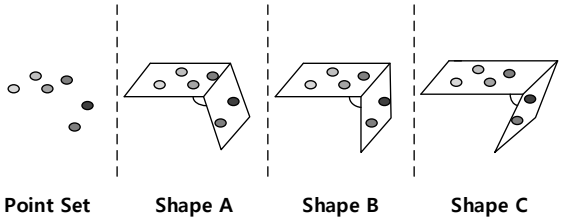

Figure 3. Example of shape ambiguity from a relatively small point set in 3D space. Given only few points, the feature extractor will learn from various shapes, which will lead to overfitting on points. In this situation, the feature extractor considers a small group of points (**the first figure**) as different shapes (**last three figures**).

other hand, unsupervised learning methods on point clouds comes up, where PointContrast [42] learns geometric information with contrast learning.

In this paper, we present deeper and wider models on point clouds. Compared with prior methods, our models lessen the problem of overfitting by multi-scale grouping for classification, and goes beyond the limitation of depth for segmentation.

2.2. Self-attention

Self-attention has generally revolutionized natural language processing [34, 39, 7, 46, 8]. This operation inspires applications in different computer vision fields, including image recognition [6, 36, 14, 15, 54, 55, 13, 2, 30], image synthesis [50, 27, 3], object detection [12, 9], and video understanding [38]. Recent work proves self-attention is a potential module to be the complement of convolution. Wang *et al*. [38] uses non-local operation in combination with convolutions to capture long-range dependencies. DETR [4] adopts transformer encoder-decoder architecture and a convolution backbone for precise detectors. Yin *et al*. [48] explores the visual clues in non-local operation and decouples pairwise terms and unary term in the operation. Channelwise attention models [36, 15, 14] re-weight activations in different channels. Very recent methods [13, 30, 52] even prove that self-attention can be an alternative to convolution. LR-Net [13] combines visual elements in a local patch with self-attention to extract image features. Zhao *et al*. [52] fully replaces convolution layers with patchwise self-attention for image classification and states higher accuracy and lower computational costs.

In this paper, we explore the effectiveness of self-attention on point clouds. The empirical evidence shows that our self-attention module can capture local geometry even with such unordered data structure.

3. Deeper or Wider Model?

In this section, we describe the notion on how to build a deeper or wider model on point clouds. Based on this, we design a groupwise self-attention module. In convolutional networks, residual learning enables networks to go deeper on image recognition. Since then, residual modules combined with convolution layers dominate computer vision. Generally, a residual block can be defined as follows:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}) + \mathbf{x}, \tag{1}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the input and output vectors of a layer. The function $\mathcal{F}(\cdot)$ represents a sequence of blocks for residual mapping in the layer.

Empirically, convolution is better for small kernels in terms of computational cost and the capabilities of local feature extraction. The most common instantiation of convolutional residual learning is ResNet [10]. The building blocks of the overall architecture are 3×3 and 1×1 convolutions, corresponding to patchwise and pixelwise enhancements respectively. To a large extent, such enhancements ignore the spatial relationship among pixels.

So can the pattern of convolutional residual learning work on point clouds? Point clouds, like other visual data

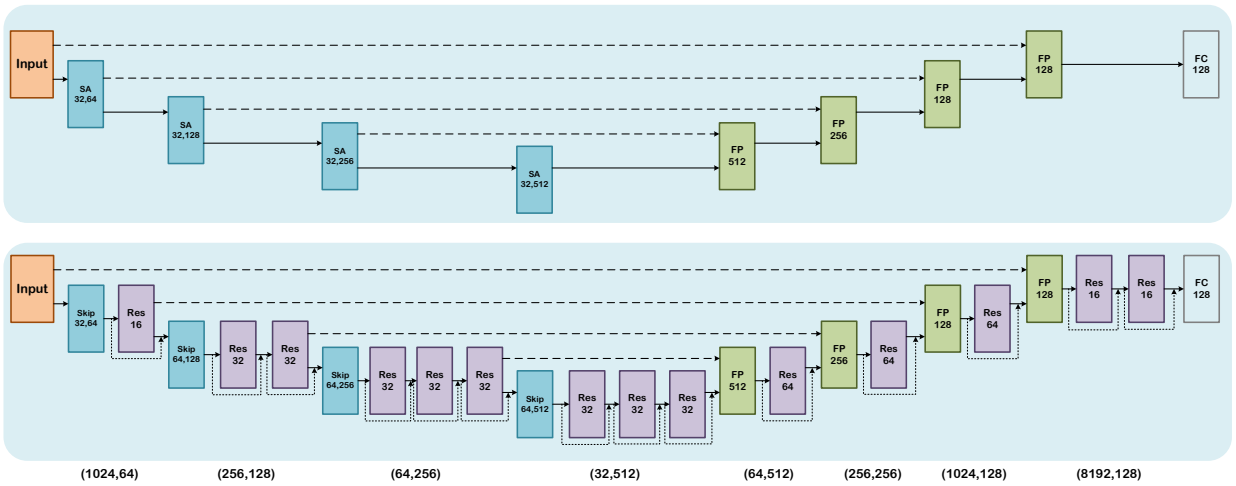

Figure 4. Comparison of architectures of PointNet++ (above) and our SepNet-D14 (below) for segmentation task. 'Res' and 'FP' represent 'Skip Block' and 'Residual Block' respectively. Each skip block groups with a specific scale and outputs the features with a fixed dimension, while each residual block enhances the features through fixed scale of groups as well. For example, the first skip block in SepNet-D14 groups 32 points corresponding to the center points and outputs 64 dimension vectors. The following residual block groups 16 points per center point.

structures (i.e., images and videos), contain spatial information. However, point clouds describe themselves through points sets or more generally through geometric surfaces. An excellent point cloud describer should focus more on shapes or sets than points. Therefore, direct application of convolutional residual learning on point clouds does not work.

Point clouds are sparse and unordered. The characteristics determine that we can hardly sketch a part of geometry with a few points (like tiny-kernel convolution), otherwise, the model will be confused about it and will learn ambiguous representations of shapes (shown in Fig. 3). Conversely, more points mean more difficult to describe the shape. PointNet++ groups a relatively large set of points (i.e., 64 or 128 out of 1024 points) to alleviate this problem.

Inspired by SA and recent convolution-like self-attention, we propose groupwise self-attention. It is efficient to capture geometry in a small or large group. Instead of scalar attention [34], we use vector attention to enables channelwise feature exploration. In this paper, we deepen and widen our model with the proposed groupwise self-attention.

Wider, or multi-scale grouping models are suitable for shape classification, while deeper, or residual models suit scene segmentation. Point or small point set plays a weaker role in shape description due to overfitting, but the combination of different sizes of grouping weakens the effect of overfitting and improves accuracy. Detailed information matters for segmentation. Therefore, deepening to enhance local information rather than extending the model works better in terms of efficiency and accuracy. We use empirical evidence to prove this hypothesis in Sec. 5.3.

4. Self-attention on Point Cloud

In this section, we first generalize the groupwise manipulations analogous to convolution operation. Next, we propose groupwise self-attention operations and the instantiations. Then we design the building block for our networks. Finally, we implement the network architectures, **P**oint Cloud **Se**lf-attention **Net**works (named **SepNet**), with the designed blocks for different tasks.

4.1. Groupwise Manipulations

Most point-based methods adopt a SA layer for feature aggregation. It achieves point downsampling as well as feature transformation. Denote the input of a layer as $\mathbf{x} \in \mathbb{R}^{C \times N}$ and $\mathbf{y} \in \mathbb{R}^{C' \times N'}$, with C, C' being the channels of input/output point features and N, N' the number of input/output points. We also denote geometric coordinates as $\mathbf{p} \in \mathbb{R}^{3 \times N}$. Specifically, this layer transforms the group of feature points $\mathbf{x}$ via pointwise multi-layer perceptrons followed by max-pooling after point sampling and grouping. To flexibly assemble feature aggregation layers of point clouds, we decompose the SA layer and generalize the manipulations on point sets analogy with image convolution layer (shown in Fig. 2). We present four types of operations on point clouds: sampling, grouping, mapping, and pooling. We present the whole formulation of a layer

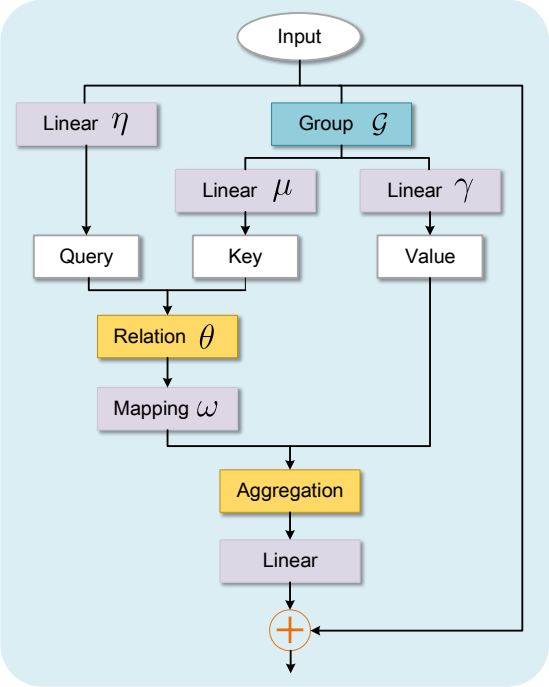

Figure 5. Our residual block with groupwise self-attention module. Skip block is similar to this except the shortcut link.

as follows:

$$\mathbf{y} = Pool\left(\mathcal{L}\left(\mathcal{K}\left(\mathbf{x}, \mathcal{S}\left(\mathbf{p}\right)\right)\right)\right). \tag{2}$$

Sampling is analogous to striding in convolution. Both operations are downsampling on point clouds and images respectively. Different from striding, point clouds can be downsampled into any number of points. We define sampling operation as follows:

$$\mathcal{S} = FPS\left(n\right), \tag{3}$$

where FPS is the algorithm of farthest point sampling [29] and n is the expected number of sampled points. For the continuity of feature learning, a relatively small factor (typically 2 or 4) is a better choice for sampling. The output of sampling $\mathcal{S}\left(\mathbf{a}\right)$ is $\mathbf{s} \in \mathbb{R}^{N'}$, meaning the indices of the sampled points. $\mathbf{a}$ denotes the coordinates for point searching.

Grouping is similar to zoning or unfolding operation in convolution layer. Typically, we have two choices to group: k-NN query and ball query from PointNet++ [29]. The former obtains more precise neighbors, while the later computes faster. We present the two of them as:

$$\mathcal{K} = kNN\left(k\right) \quad \text{or} \quad \mathcal{K} = Ball\left(k, r\right), \tag{4}$$

where k is the number of points per group and r is the radius of a ball group. The result of grouping $\mathcal{K}\left(\mathbf{b}, \mathbf{s}\right)$ is $\mathbf{g} \in \mathbb{R}^{C \times N \times K}$, which is retrieved from $\mathbf{b}$ with the indices $\mathbf{s}$.

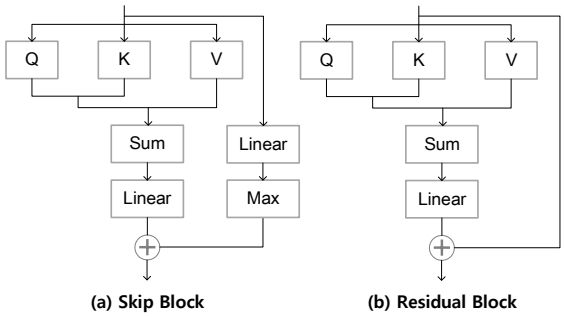

Figure 6. Our self-attention blocks. 'Q', 'K', 'V' mean 'Query', 'Key' and 'Value' corresponding to original self-attention elements from [34]. 'Sum' and 'Max' is the pooling operation for each point set. (a) is the block for point cloud downsampling. The right stream of (a) is a shortcut for sketching the features of a point set, and the left stream captures dependencies within a point set. (b) block enhances local geometric features with a residual link. We apply (a) to both classification and segmentation tasks, while (b) is only applicable to segmentation task. The number of points will not change through (b).

Mapping and weighting in convolution layer are alike, both of which transform features pointwise or patchwise. PointNet++ performs feature transformation through multi-layer perceptrons. In this paper, we define a mapping module as $\mathcal{L}$, representing a sequence of self-attention blocks.

Pooling is the final operation to summarize a group of feature vectors into one. Convolution performs the aggregation with summation, while max-pooling is better for shape description. Different pooling methods, including sum-pooling, mean-pooling, max-pooling and attentive pooling, can be applied here.

4.2. Groupwise Self-attention

We explore self-attention on point clouds. The common self-attention [34] or non-local operation concentrates on the global information. It is inadequate for local feature aggregation of point clouds. Thus we propose groupwise self-attention (shown in Fig. 5), which has the following form:

$$\mathbf{y}_i = \sum_{j \in \mathcal{G}(i)} \delta\left(\mathbf{x}_i, \mathbf{x}_j\right) \odot \gamma\left(\mathbf{x}_j\right), \tag{5}$$

where $\odot$ is the Hadamard product, i is the index of the center point of the given group, $\mathcal{G}\left(i\right)$ represents the local group of points through grouping operation $\mathcal{K}$. $\mathcal{G}\left(i\right)$ is a set of indices that specifies which feature vectors for aggregation to construct the new feature $\mathbf{y}_i$.

The function δ computes the weights $\delta\left(\mathbf{x}_i, \mathbf{x}_j\right)$ that are used to combine the transformed features $\gamma\left(\mathbf{x}_j\right)$. To dis-

Stage	Points	SepNet-W7		SepNet-W9		SepNet-W15	
GSA1	512	Ball G-SA Linear	× 3	Ball G-SA Linear	× 4	Ball G-SA Linear	× 7
GSA2	128	Ball G-SA Linear	× 3	Ball G-SA Linear	× 4	Ball G-SA Linear	× 7
GSA3	1	All G-SA Linear	× 1	All G-SA Linear	× 1	All G-SA Linear	× 1
CLS	–	512-d fc, 256-d fc, 40-d fc → softmax					

Table 1. SepNet-W Architectures for ModelNet40 Classification. We denote each groupwise self-attention layer by 'GSA'. 'Ball' and 'All' stand for ball query grouping and overall grouping strategies. 'G-SA' means our groupwise self-attention module. The scales of grouping for first two stages are limited within the fixed ranges $[16, 128]$ and $[32, 128]$.

entangle exposition of different forms of self-attention, we define δ as follows:

$$\delta\left(\mathbf{x}_i, \mathbf{x}_j\right) = \omega\left(\theta\left(\mathbf{x}_i, \mathbf{x}_j\right)\right). \tag{6}$$

The relation function θ produces a single vector that represents the features $\mathbf{x}_i$ and $\mathbf{x}_j$. The function γ then maps this vector into a vector that can be combined with $\gamma\left(\mathbf{x}_j\right)$ as shown in Eq. 6.

The function ω is a sequence of mapping operations; i.e., $\{MLP \rightarrow ReLU \rightarrow MLP\}$. ω allows us to introduce additional trainable transformations for more expressive construction of the weights $\delta\left(\mathbf{x}_i, \mathbf{x}_j\right)$. 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 vector attention.

The function γ outputs the feature vectors $\gamma\left(\mathbf{x}_j\right)$ that are aggregated by the adaptive weight vectors $\delta\left(\mathbf{x}_i, \mathbf{x}_j\right)$. We explore possible instantiations of δ, along with feature transformation elements that surround self-attention operations in our architecture:

Summation: $\theta\left(\mathbf{x}_i, \mathbf{x}_j\right) = \eta\left(\mathbf{x}_i\right) + \mu\left(\mathbf{x}_j\right)$

Concatenation: $\theta\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left[\eta\left(\mathbf{x}_i\right), \mu\left(\mathbf{x}_j\right)\right]$

Hadamard product: $\theta\left(\mathbf{x}_i, \mathbf{x}_j\right) = \eta\left(\mathbf{x}_i\right) \odot \mu\left(\mathbf{x}_j\right)$

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

4.3. Self-attention Blocks

We design two forms of blocks containing our self-attention module mentioned in Sec. 4.2: skip-connection block (Fig. 6 a) and residual block (Fig. 6 b). Skip-connection block is a transition for downsampling. The self-attention module is on the left stream, while we add a skip connection on the right for fast feature extraction.

Stage	Points	SepNet-D14		SepNet-D22		SepNet-D27	
Down1	1024	kNN G-16 L-64	× 1	kNN G-16 L-64	× 2	kNN G-16 L-64	× 3
Down2	256	kNN G-32 L-128	× 2	kNN G-32 L-128	× 3	kNN G-32 L-128	× 4
Down3	64	kNN G-64 L-256	× 3	kNN G-64 L-256	× 4	kNN G-64 L-256	× 5
Down4	32	kNN G-128 L-512	× 3	kNN G-128 L-512	× 4	kNN G-128 L-512	× 5
Skip1	64	feature propagation					
Up1	64	kNN G-128 L-512	× 1	kNN G-128 L-512	× 2	kNN G-128 L-512	× 2
Skip2	256	feature propagation					
Up2	256	kNN G-64 L-256	× 1	kNN G-64 L-256	× 2	kNN G-64 L-256	× 2
Skip3	1024	feature propagation					
Up3	1024	kNN G-32 L-128	× 1	kNN G-32 L-128	× 2	kNN G-32 L-128	× 2
Skip4	8192	feature propagation					
Up4	8192	kNN G-32 L-128	× 2	kNN G-32 L-128	× 3	kNN G-32 L-128	× 4
SEG	8192	128-d fc					

Table 2. SepNet-D Architectures for ScanNet and S3DIS Segmentation. We denote downsampling layer, upsampling layer and skip-connection layer by 'Down', 'Up' and 'Skip'. 'kNN' means kNN grouping strategy. 'G-X' is a groupwise self-attention module with the output channels 'X', while 'L-X' is a pointwise multi-layer perceptrons with the output of 'X' dimension. Before each upsampling stage, we perform feature propagation followed by skip connection.

We use residual block for the enhancement of local features. For efficiency, we design the bottleneck version of self-attention blocks. Denote the channel dimensionality of input by C. The query and the key have C/r_1 channels. The value and the output of self-attention 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 $r_2 = 4$.

4.4. SepNet

The main structures of our SepNet generally follow PointNet++ [29], which we use as baselines. Tab. 1 and Tab. 2 respectively present three architectures obtained by stacking self-attention blocks at different point cloud sizes. The number X in SepNet-WX and SepNet-DX refers to the number of our self-attention blocks. Our architectures are based mostly on self-attention.

Classification. We build our SepNet-W with skip blocks only. As shown in Tab. 1, the backbone of SepNet has three stages, each with different spatial resolution. Every stage comprises multiple self-attention blocks. In SepNet-W7, grouping of the first two stages are performed by multi-scale

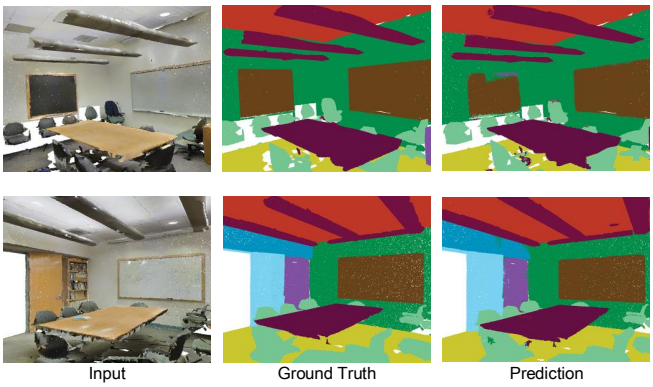

Figure 7. Examples of semantic scene labeling with SepNet-D.

Method	Modality	Accuracy(%)
PointGCN [51]	Graph	89.5
KPConv [33]	Grid	92.9
PVRNet [49]	Points+Views	93.6
SO-Net [21]	Points+Normals (5k)	93.4
RS-CNN [25]	Points	93.6
PointNet [28]	Points	89.2
FPConv [23]	Points+Normals	92.5
Grid-GCN [43]	Points+Normals	93.1
PointASNL [45]	Points	92.9
PointASNL [45]	Points+Normals	93.2
PointNet2 [29]	Points+Normals	91.9
PointNet2-W15	Points+Normals	92.5
SepNet-W7	Points	92.8
SepNet-W15	Points	**93.1**
SepNet-W7	Points+Normals	93.2
SepNet-W15	Points+Normals	**93.6**

Table 3. Performance of classification on ModelNet40 on accuracy(%).

groupers, with the sizes of $\{16, 32, 128\}$ and $\{32, 64, 128\}$ in order. SepNet-W9 and SepNet-W15 adopt more detailed scales within $[16, 128]$ and $[32, 128]$ (see supplementary material). 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. In Tab. 2, the input of SepNet-D is 8192 points containing various information (i.e. coordinates, color and normal). SepNet-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 [29]. The residual self-attention blocks are attached to the downsampling or upsampling blocks. The segmentation layer includes a linear layer and a dropout layer of 0.5. The instantiation of SepNet-D (SepNet-D14) is shown as an illustration in Fig. 4.

Method	S3DIS		ScanNet
	6-fold	Area5	
PointNet++ [29]	53.4	-	33.9
PointCNN [22]	65.4	57.3	45.8
PointWeb [53]	66.7	60.3	-
HPEIN [17]	67.8	61.9	61.8
FPConv [23]	68.7	62.8	63.9
RandLA [16]	70.0	-	-
PointASNL [45]	68.7	62.6	66.4
KPConv [33]	**70.6**	**67.1**	**68.4**
SepNet-D14	68.3	62.4	65.2
SepNet-D27	**70.1**	**64.5**	**66.7**
SepNet-D27 (16k)	**70.6**	**64.9**	**67.3**

Table 4. Mean per-class IoU(%) for the task of semantic segmentation on the datasets of ScanNet v2 and S3DIS Datasets.

5. Experiments

We evaluate our SepNet-W and SepNet-D on the classification and segmentation tasks of various datasets (including synthetic datasets and large-scale scene segmentation datasets). Ablation studies explore variants of our self-attention module and assess the effectiveness of the components used to construct the networks (see supplementary material). All the experiments are performed on a machine with four GPUs.

5.1. Evaluation on Classification

We conduct experiments on ModelNet40 [41] on classification through our SepNet-W network. The dataset contains 9843 training point clouds and 2468 test ones from 40 different categories. To compare SA and our self-attention module, we train and test using the data provided by [28].

Implementation. Our implementation mainly follows the practice in [28]. For training, we first select 1024 points as input. To prevent overfitting, we apply augmentation strategy including the following components: random anisotropic scaling in range $[0.8, 1.25]$, translation in the range $[-0.1, 0.1]$, random dropout 20% points. The initial learning rate is 0.001, and it decays by a factor of 0.7 every 20 epochs. We use Adam [18] as the optimizer with coefficients 0.9 and 0.999. For testing, similar to [28, 29], we average the predictions of randomly scaled inputs. We train our SepNet-W models according to the architectures in Tab. 1.

Results. In Tab. 3, we compare our SepNet-W with state-of-the-art classification methods on ModelNet40. Among all current methods, our method achieves state-of-the-art. Unprecedentedly, our method (1024 points input) is competitive with grid-based methods [33, 43] (6800 points input on average) and image-based methods [33, 49] (12 views with size 224×224 and 1024 points input) with lower input costs. Without a tricky voting strategy (even

wide model	PointNet2			SepNet-W(w/o bottleneck)			SepNet-W(w/ bottleneck)		
	Acc	Params	FLOPs	Acc	Params	FLOPs	Acc	Params	FLOPs
width-5	91.6	1.5M	2.7G	92.8	3.8M	11.9G	92.8	2.0M	9.9G
width-7	91.9	1.7M	4.0G	93.1	5.5M	21.6G	93.2	2.6M	17.3G
width-9	92.4	1.9M	5.7G	93.5	6.7M	28.3G	93.4	2.9M	21.8G
width-15	92.5	2.4M	11.3G	93.6	12.3M	63.5G	93.6	4.8M	44.9G

Table 5. Comparison of different widths of PointNet2 and SepNet-W on ModelNet40.

deep model	SepNet-D(w/o bottleneck)				SepNet-D(w/ bottleneck)			
	mIoU(%)	OA(%)	Params	FLOPs	mIoU(%)	OA(%)	Params	FLOPs
depth-8	63.2	84.0	4.9M	2.4G	63.1	84.1	1.0M	1.2G
depth-14	65.2	85.2	10.7M	5.2G	65.2	85.2	1.5M	2.2G
depth-22	65.9	87.3	14.0M	8.3G	66.1	87.5	2.1M	4.1G
depth-27	66.5	88.5	17.0M	10.4G	66.7	88.7	2.3M	5.1G

Table 6. Comparison of different depths of SepNet-W on ScanNet.

sampling and retrieving the best in 300 repeated tests), the result of our SepNet-W15 is the same as RS-CNN [25]. Our model without normal input is also competitive to most of the methods with normal input. Besides our SepNet-W, PointNet2 [29] gains an increase of 0.6% by expanding to a width of 15 (equivalent to 15 SA blocks). All the evidence proves the excellence of our wider model and groupwise self-attention module.

5.2. Evaluation on Segmentation

Large-scale scene segmentation is a more challenging task for outliers and noise. We evaluate our SepNet-D on Stanford 3D Large-Scale Indoor Spaces (*S3DIS*) [1] and ScanNet v2 (*ScanNet*) [5] datasets. *S3DIS* contains 271 scenes from 6 indoor areas. It provides 13 types of semantic labels for scene segmentation. *ScanNet* includes 1513 indoor training point clouds and 100 test ones. It marks points 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 areas (6-fold) and Area 5 on *S3DIS*. For training, we randomly sample 8192 (or 16384) points from the scenes. For evaluation, similar to [45], we obtain an average prediction of 5 votes by sliding a window across the room in 0.5m stride.

Results. In Tab. 4, we compare our SepNet-D with other latest methods under the same training and testing strategy (randomly chopping cubes with a fixed number of points), e.g., PointNet++ [29], PointCNN [22], PointConv [40], PointWeb [53], HPEIN [17] and PointASNL [45]. With more points input (from 8k to 16k), our SepNet-D27 achieves an extra 0.5% and 0.6% accuracy on *S3DIS* and *ScanNet*, respectively. We also list the results of another kind of methods (using non-fixed points or the entire scene as input), such as KPconv [33]. All methods only use point clouds as input without voxelization. An illustration of semantic scene labeling is shown in Fig. 7.

5.3. Analysis of Wider and Deeper Networks

We analyze the accuracy and efficiency of our SepNet-W (for classification) and SepNet-D (for segmentation) with different metrics, including accuracy (Acc) for classification precision, label accuracy (OA) and mIoU for segmentation precision, the number of parameters from a model (Params) and FLOP budgets for efficiency. Note that we also compare models with and without our bottleneck blocks.

Wider Model. As shown in Tab. 5, we compare our SepNet-W with PointNet2 multi-scale grouping (MSG) version. The accuracy of our SepNet-W is around 1% higher than that of PointNet2 with the same width, which reflects the superiority of our groupwise self-attention module. 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 SepNet-W9 and SepNet-W15 shows little accuracy advancement. We argue that this is due to the full exploitation of a grouping range.

Deeper Model. Shown in Tab. 6, deeper models are more competitive on segmentation. We compare the efficiency improvement of SepNet-D with our bottleneck blocks. To verify the effectiveness of greater depth, we gradually increase the depth of SepNet-D on *ScanNet*. Obviously, as the depth increases, the model can obtain higher mIoU and OA. We argue that our networks can deeply extract representations by stacking residual self-attention blocks.

6. Conclusion

We present groupwise self-attention as well as deeper (SepNet-D) and wider (SepNet-W) models for accurate

point cloud analysis. By obtaining the short-range and long-range dependencies within a point set, our SepNet outperforms on both classification and segmentation tasks. For higher accuracy, increasing depth or width can be helpful for segmentation or classification, respectively. We further apply bottleneck self-attention blocks for the refinement of efficiency. Experiments based on competitive datasets illustrate the effectiveness of our SepNet.

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, 8

[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] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 3

[4] 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

[5] 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, 8

[6] 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

[7] 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

[8] 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. 1, 3

[9] 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

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 3

[11] 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. 2

[12] 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

[13] 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

[14] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. In *Advances in neural information processing systems*, pages 9401–9411, 2018. 3

[15] 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

[16] 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. 7

[17] 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. 7, 8

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[19] 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

[20] 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

[21] 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. 7

[22] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in neural information processing systems*, pages 820–830, 2018. 2, 7, 8

[23] 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. 2, 7

[24] 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. 2

[25] 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. 2, 7, 8

[26] Youngmin Park, Vincent Lepetit, and Woontack Woo. Multiple 3d object tracking for augmented reality. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 117–120. IEEE, 2008. 1

[27] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. *arXiv preprint arXiv:1802.05751*, 2018. 3

[28] 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. 2, 7

[29] 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. 2, 5, 6, 7, 8

[30] 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

[31] Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1711–1719, 2020. 2

[32] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016. 1

[33] 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. 2, 7, 8

[34] 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, 4, 5

[35] Dominic Zeng Wang and Ingmar Posner. Voting for voting in online point cloud object detection. In *Robotics: Science and Systems*, volume 1, pages 10–15607, 2015. 1

[36] 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

[37] 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. 2

[38] 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. 2, 3

[39] Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. *arXiv preprint arXiv:1901.10430*, 2019. 3

[40] 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. 2, 8

[41] 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. 2, 7

[42] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas J Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. *arXiv preprint arXiv:2007.10985*, 2020. 3

[43] 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. 2, 7

[44] 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. 2

[45] 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. 2, 7, 8

[46] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763, 2019. 3

[47] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11040–11048, 2020. 1

[48] 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

[49] 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. 7

[50] 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

[51] 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. 7

[52] 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, 2, 3

[53] 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. 2, 7, 8

[54] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 267–283, 2018. 3

[55] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019. 3