# Improving Graph Representation for Point Cloud Segmentation via Attentive Filtering

Nan Zhang, Zhiyi Pan, Thomas H. Li, Wei Gao✉, Ge Li
School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University
{zhangnan, panzhiyi}@stu.pku.edu.cn
{thomas, gaowei262}@pku.edu.cn, geli@ece.pku.edu.cn

## Abstract

*Recently, self-attention networks achieve impressive performance in point cloud segmentation due to their superiority in modeling long-range dependencies. However, compared to self-attention mechanism, we find graph convolutions show a stronger ability in capturing local geometry information with less computational cost. In this paper, we employ a hybrid architecture design to construct our Graph Convolution Network with Attentive Filtering (**AF-GCN**), which takes advantage of both graph convolution and self-attention mechanism. We adopt graph convolutions to aggregate local features in the shallow encoder stages, while in the deeper stages, we propose a self-attention-like module named Graph Attentive Filter (GAF) to better model long-range contexts from distant neighbors. Besides, to further improve graph representation for point cloud segmentation, we employ a Spatial Feature Projection (SFP) module for graph convolutions which helps to handle spatial variations of unstructured point clouds. Finally, a graph-shared down-sampling and up-sampling strategy is introduced to make full use of the graph structures in point cloud processing. We conduct extensive experiments on multiple datasets including S3DIS, ScanNetV2, Toronto-3D, and ShapeNetPart. Experimental results show our **AF-GCN** obtains competitive performance.*

## 1. Introduction

With the rapid development of 3D sensing technologies (such as LiDARs and RGB-D cameras), 3D point clouds have demonstrated great potential in many applications such as robotics, autonomous driving, virtual reality and augmented reality [10]. Consequently, point cloud segmentation has attracted more and more attention. Unlike regular pixel grids in 2D images, 3D points in point clouds are irregular and unstructured, thereby posing significant challenges for point cloud segmentation.

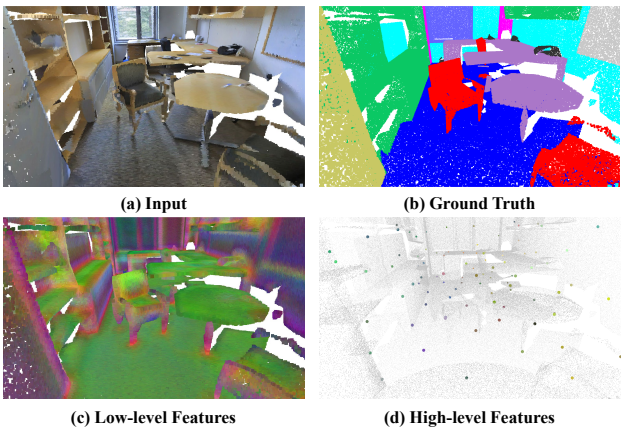


Figure 1. (a) and (b) are the input point cloud and corresponding semantic labels, respectively. (c) and (d) are the visualization of the low-level and high-level point features after the first and last down-sampling, respectively. Differences in color indicate differences in features. As shown in (c), neighbors in the same object may have low feature correlations due to the differences in RGB attributes or geometry structures. As shown in (d), points after several down-sampling are sparse and the distant neighbors in high-level feature aggregation should be filtered because of containing possible irrelevant information.

Several researches [18, 21, 46] adopt graph convolution networks to utilize the topological structure of point cloud for segmentation. Graph convolution networks learn features from points and their neighbors for better capturing local geometric features while maintaining permutation invariance, which have intrinsic advantages for handling non-Euclidean point cloud data. Furthermore, many works [17, 44, 51, 59] improve the graph convolution networks by proposing well-designed convolution kernels and get promising performance in point cloud segmentation.

Recently inspired by the great success of vision transformers [8, 11, 23, 34, 56], several works [9, 15, 36, 52, 55, 58]

*Corresponding author: Wei Gao.*

introduce self-attention mechanism into point cloud analysis for its superiority in modeling long-range dependencies and high-level relations, which obtain significant performance improvement, especially in point cloud segmentation. However, self-attention mechanism exhibits certain limitations in capturing local geometry information. Compared with graph convolutions, self-attention mechanisms require additional computation for feature correlations, and assign large weights to neighbors which have high feature correlations. As illustrated in Figure 1, points in low-level feature learning phases are dense and low-level features are mainly extracted from the colors and geometry structures (like edges, corners and surfaces). Therefore, self-attention mechanisms are inefficient in low-level feature aggregation and may neglect information about neighbors which have considerable differences in colors or geometry structures.

To exploit the advantages of graph convolution in capturing local geometry information and self-attention mechanism in modeling long-range contexts simultaneously, we design a hybrid network, namely *Graph Convolution Network with Attentive Filtering* (AF-GCN). In the shallow stages of the encoder, we adopt graph convolutions to aggregate local geometry information. While in the deeper stages, we propose a self-attention-like module in the graph convolution form called *Graph Attentive Filter* (GAF) to improve graph representation for point cloud segmentation. Different from previous studies [9, 15, 36, 50, 58], our proposed *Graph Attentive Filter* estimates the correlation between the points from both features and spatial structures information, then suppresses irrelevant information from the distant neighbors to better capture high-level relations.

To further improve our graph convolution networks for point cloud segmentation, we adopt a *Spatial Feature Projection* (SFP) module for graph convolutions. The spatial feature projection module projects the spatial information of points into the feature space, which helps graph convolutions with isotropic kernels to model spatial variations effectively. Moreover, we design a graph-shared down-sampling and up-sampling strategy to better utilize the graph structures in the decoder. In general, our key contributions are summarized as follows:

- We construct a hierarchical graph convolution network AF-GCN with a hybrid architecture design for point cloud segmentation, which takes advantage of graph convolution and self-attention mechanism.
- We propose a novel Graph Attentive Filters module to suppress irrelevant information from distant neighbors by estimating the correlation between the points from both features and spatial structure information.
- We employ a Spatial Feature Projection module for graph convolutions to handle the spatial variation of irregular point clouds. To better exploit the graph structures, we design a graph-shared down-sampling and up-sampling strategy.
- Experimental results demonstrate our model achieves state-of-the-art performance on multiple point cloud segmentation datasets. Ablation studies also verify the effectiveness of each proposed component.

## 2. Related Work

**3D representation learning.** How to overcome the disorder, irregularity and geometric transformation invariance of point clouds is a hot topic of early research in the field of 3D deep learning. Multi-view-based approaches [4, 16, 20, 40, 43] and voxel-based approaches [2, 6, 26, 27, 37] process point cloud into a regularly arranged compact data by geometric computations such as projection or quantization, and then learn network embedding using 2D or 3D convolution. Although this processing overcomes the above challenges, it reintroduces quantification error and heavy calculation. Therefore, point-based methods are received increasing attention from researchers, which takes the point cloud directly as the networks' input and alleviates the obstructions of the point cloud through a sophisticated network design.

**Point-based methods.** For different network structure designs, point-based methods can be broadly subdivided into three categories, i.e., MLPs-based methods [31, 32], convolution-based methods [14, 19, 47], and graph convolution-based methods [17, 21, 46]. The MLPs-based methods perform global or hierarchical local feature extraction and aggregation of point cloud inputs through permutation-invariance MLPs and pooling operators. Convolution-based methods obtain convolution kernels applicable to irregularly arranged and unordered data in predefined or learning ways and obtain representations of the point cloud by multiple convolutions in local regions. The graph convolution-based approach treats the points as the vertices of the graph and implements graph convolution by considering features on both vertices and edges, thus maintaining the topological relationship of the graph over the features. Along with the considerable research on the network structure of point-based methods, some works [12, 18, 24, 33, 35, 42, 44, 51, 57, 59] further explore how to improve existing frameworks to capture more solid semantic representations. Owing to the success of attention on 2D visual tasks, alternative approaches [9, 58] attempt to introduce it into point cloud learning. We collectively refer to such works as attention-based methods.

**Attention-based methods.** Attention mechanism is proposed in the field of natural language processing to capture long dependencies in textual contexts [45]. In recent years, self-attention [11, 34, 56], as well as Transformer [8, 23],

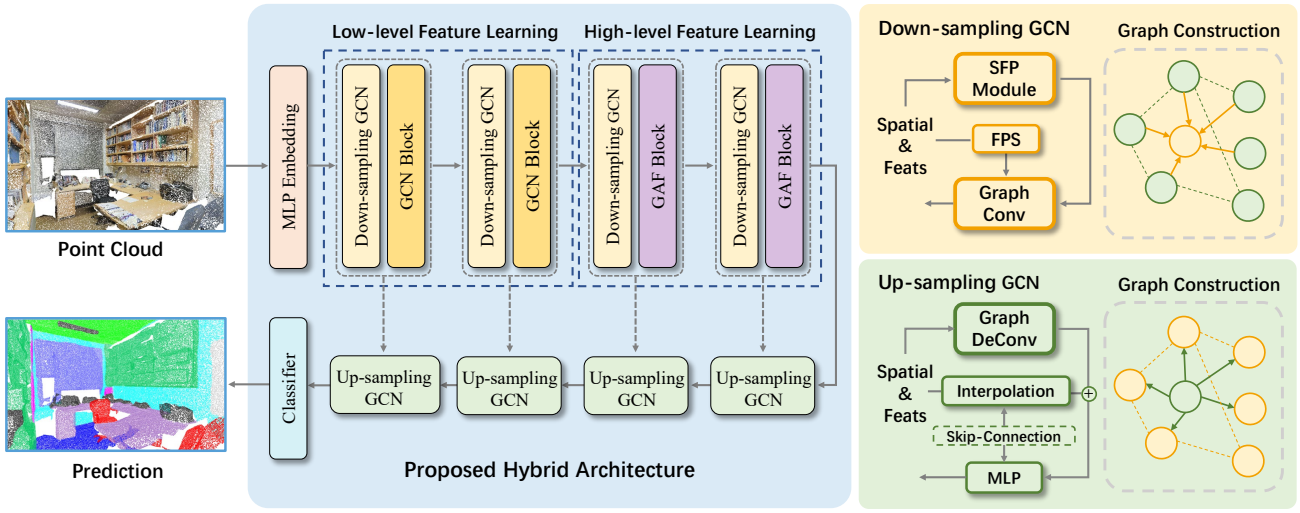


Figure 2. The framework of our proposed model. A hybrid architecture design is employed: In the shallow stages of the encoder, we adopt GCN blocks to aggregate local geometry information. In the deeper stages, we adopt GAF blocks to suppress irrelevant information from distant neighbors. Graph-shared down-sampling and up-sampling layers are utilized for better exploiting the multi-scale graph structures. GAF: Graph Attentive Filter. SFP: Spatial Feature Projection. FPS: Farthest Point Sampling.

have been introduced to 2D vision tasks, enabling features to be more focused on the regions of interest in the task. As a result, many works attempt to utilize attention mechanism for feature aggregation on 3D vision tasks such as semantic segmentation [9, 52], object detection [3, 25, 53], target tracking [39], etc., while demonstrating the generalizability and capability of attention mechanism on 3D vision tasks. PointTransformer [58] utilizes vector self-attention to achieve point features aggregation within a local region on a U-Net structure network with skip connections. An almost parameter-free cross-scale attention mechanism is designed and introduced into the model for fusing point representations at multiple scales [29]. In order to enlarge the range of effective receptive fields, Stratified Transformer [15] proposes a stratified strategy for sampling keys in self-attention module. RPNet [36] explores an attention-like group relation aggregator to consider both geometric relations and semantic relations within a local region. FPT [30] and PatchFormer [55] further investigate how to design a lightweight Transformer for 3D scene understanding that reduces the time for model training and inference.

Although these methods demonstrate that the attention mechanisms have great potential in 3D representation learning, they indiscriminately use the attention module at all stages of the networks, thus incurring extra computational costs and weakening the local information. Differently, our method design a hybrid architecture to take advantage of both graph convolutions and self-attention mechanisms.

## 3. Methods

In this section, we introduce our proposed **G**raph **C**onvolution **N**etwork with **A**ttentive **F**iltering termed AFGCN. We first give preliminaries of the proposed model, then we give elaborations of the components that build our model. Finally, we describe the specific architectures.

### 3.1. Preliminaries

**Graph construction.** We construct a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to represent the structures of point clouds, where $\mathcal{V} = \{1, \ldots, n\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ are the vertices and edges respectively. In our model, we construct the graph $\mathcal{G}$ by applying a ball query to each point in point clouds. We formulate the process as follows:

$$\mathcal{E} = \{(i, j) \mid \|p_i - p_j\| \leq r, \forall i, j \in \mathcal{V}\}, \tag{1}$$

where $p_i$ and $p_j$ are the corresponding coordinates, $r$ is the ball query radius. For computational efficiency, we set the maximum number of edges per point to $K$. We can also represent the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, where $\mathbf{A}_{ij} = \{1, \text{if } (i, j) \in \mathcal{E}; 0, \text{if } (i, j) \notin \mathcal{E}\}$. The graph $\mathcal{G}$ is reconstructed after every down-sampling and up-sampling and the radius changes accordingly.

**Graph convolutions.** The standard graph convolution is formulated as follows:

$$\mathbf{F}^{l+1} = \mathbf{A}\mathbf{F}^{l}\mathbf{W}^{l}, \tag{2}$$

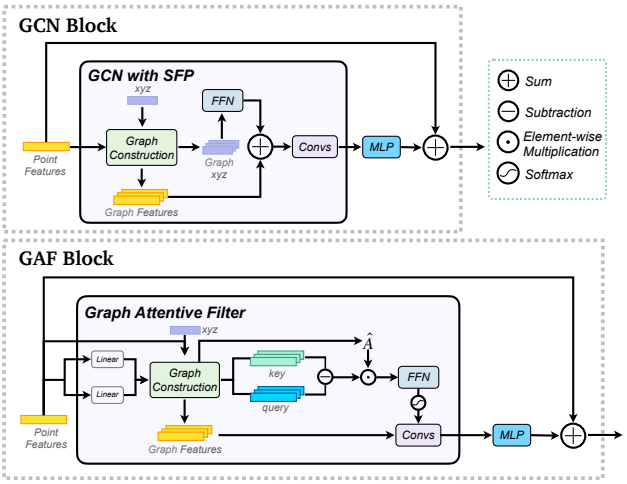


Figure 3. Architectures of the proposed GCN Block and GAF Block. Zoom in for a better view.

where $\mathbf{F}^l$ is the input features of the $l$-th layer, $\mathbf{F}^{l+1}$ is the output features. $\mathbf{W}^l$ is the weight matrix of the $l$-th graph convolution. Equation 2 describe the graph convolutions which use sum pooling to aggregate edge features and contain linear feature transformation only. In practice, non-linear feature transformation is important, and max pooling is also suitable for feature aggregation adopted in many previous works. We give a more general formulation of graph convolutions for a better understanding of our method:

$$\mathbf{F}^{l+1} = \mathbf{A} \diamond \Phi(\mathbf{F}^l), \tag{3}$$

where $\diamond$ is the graph aggregation operation to generate graph features $\mathbf{F}^{l+1}$ and $\mathbf{F}_{ij}^{l+1} = \text{Pooling}(\mathbf{A}_{ik}\,\Phi(\mathbf{F}^l)_{kj})$. $\Phi$ is the non-linear feature transform function.

### 3.2. Hybrid Architecture Design

The proposed AF-GCN model is a hierarchical graph convolution network including an encoder and a decoder. During gradual down-sampling and feature aggregation by the encoder, the receptive field of each point continues to expand. We design a hybrid architecture for the encoder to fit the variance of the receptive field in each encoder block. As shown in Figure 2, the encoder of our model is split into two phases. We consider that the encoder layers in low-level feature learning phase capture the local patterns, and the encoder layers in high-level feature learning phase capture the more abstract features and long-range contexts. We adopt graph convolutions in the low-level features learning phase. In the high-level feature learning phase, we propose a feature aggregation module named *Graph Attentive Filter* to empower the graph convolutions with the superiority of self-attention mechanism in modeling long-range contexts.

**Graph Attentive Filter** We design a novel graph convolution module called Graph Attentive Filter as shown in Figure 3. Compared with regular graph convolutions, our proposed Graph Attentive Filter estimates the correlations between points to their neighbors before feature aggregation. By applying the estimated edge correlations, our module could filter the irrelevant information from distant neighbors for better modeling long-range contexts. Inspired by the previous self-attention mechanism study in image recognition and point cloud analysis [8, 56, 58], we estimate the preliminary point-to-point feature correlation by subtraction for computational efficiency. We formulate the correlation matrix $\mathcal{R} \in \mathbb{R}^{N\times N\times C}$ as follows:

$$\mathcal{R}_{ij} = \phi(\mathbf{f}_i) - \psi(\mathbf{f}_j), \tag{4}$$

where $\mathbf{f}_i$ and $\mathbf{f}_j$ are the corresponding point features, $\phi$ and $\psi$ are two separated linear functions to project the point features into metrics space. However, the correlation matrix $\mathcal{R}$ only takes into account of feature-level correlations. We consider that spatial and structural correlations are also significant in graph learning. Intuitively, we use the point-to-point distance to update the ball query based adjacency matrix $\mathbf{A}$. We formulated the updated adjacency matrix $\tilde{\mathbf{A}}$ as follows:

$$\tilde{\mathbf{A}}_{ij} = \mathbf{A}_{ij} \cdot \mathrm{e}^{-\|\mathbf{p}_i - \mathbf{p}_j\|_2^2}, \tag{5}$$

where $\mathbf{p}_i$ and $\mathbf{p}_j$ are the corresponding point positions. Point pairs that have smaller distances have bigger edge weights in the updated adjacency matrix $\tilde{\mathbf{A}}$. Then we normalize $\tilde{\mathbf{A}}$ by the diagonal degree matrix $\mathbf{D}$, which is formulated as:

$$\begin{aligned} \hat{\mathbf{A}} &= \mathbf{D}^{-\frac{1}{2}}\tilde{\mathbf{A}}\mathbf{D}^{-\frac{1}{2}}, \\ \mathbf{D}_{ii} &= \Sigma_j \tilde{\mathbf{A}}_{ij}. \end{aligned} \tag{6}$$

As Equation 6, the symmetric adjacency matrix $\hat{\mathbf{A}}$ is calculated based on the topological relations of the graph and the point-to-point distances, which could embody both spatial and structural correlations. Then we integrate the correlation matrix $\mathcal{R}$ into the graph convolution form:

$$\mathbf{F}^{l+1} = \sigma(\text{FFN}(\mathcal{R} \cdot \hat{\mathbf{A}})) \diamond \Phi(\mathbf{F}^l), \tag{7}$$

where FFN is a feed-forward network to further learn the correlations, $\cdot$ represents element-wise multiplication for each channel. $\diamond$ is the graph aggregation operation. $\Phi$ represents the non-linear feature transformation and $\sigma$ is a normalization operator. In practice, we use the softmax function as normalization. As Equation 7, our graph attentive filter suppresses irrelevant information by estimating the correlation from both features and spatial structures. We give a detailed description in the supplementary material.

**Graph-Conv with Spatial Feature Projection** We adopt graph convolutions in down-sampling layers and feature extractors in our GCN blocks to aggregate the point features.

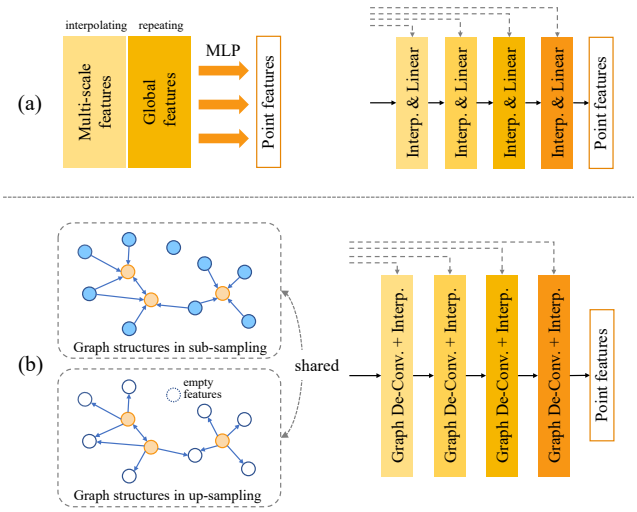


Figure 4. (a) shows two decoder illustrations of previous works. (b) shows the graph structures (left) and the architecture of the decoder (right) in our AF-GCN. The yellow points are the down-sampled points and the hollow points are the up-sampled points.

Previous GCNs works in point cloud analysis [51, 59] have pointed out that isotropic kernels in the standard graph convolution neglect space and feature correspondences. These works give various designs of dynamic graph convolution kernels that aim to model the relationship between point-to-point distance and features. Unlike these dynamic kernel generation methods, we adopt a simpler way to model the feature with spatial information in our graph convolutions. We consider spatial information is essential in point cloud analysis and should be emphasized during down-sampling and feature aggregation. Simply concatenating spatial information and features will reduce the validity of spatial information when the feature dimension increases. Thus, we employ a spatial feature projection module to project the relative spatial information into the feature space and add it to point features as shown in Figure 3. Then we formulate graph convolutions in our models the as:

$$\mathbf{F}_i^{l+1} = \underset{(i,j)\in\mathcal{E}}{\text{MaxPooling}}\ \Phi(\mathbf{F}_j^l + \delta(\mathbf{p}_j - \mathbf{p}_i)), \tag{8}$$

where $\mathbf{F}_i^{l+1}$, $\mathbf{F}_j^l$ are the output features and input features of corresponding points. $\mathbf{p}_i$ and $\mathbf{p}_j$ are the corresponding spatial information. $\Phi$ is a non-linear feature transformation function, and $\delta$ is the Spatial Feature Projection Module implemented by a feed-forward network.

### 3.3. Graph-shared Down-sampling & Up-sampling

Previous GCNs works in point cloud analysis [49, 51, 58, 59] mostly use two types of decoder illustrated in Figure 4 (a) to generate the final point feature at the same resolution as input, which neglecting the graph structure in the encoder. Inspired by [5], we propose the graph-shared down-sampling and up-sampling GCN layers to fully utilize the multi-scale graph structure information.

As Figure 2, in the down-sampling GCN layers, we first use Furthest Point Sampling (FPS) to select down-sampled points. For the down-sampled points, we use ball query in the original point sets to generate graph $G = \mathbf{A}$. Then we use a graph convolution with spatial feature projection to aggregate the point features. In the up-sampling GCN layers, we define the shared graph $\hat{G} = \mathbf{A}^\top$, then we perform the graph de-convolution based on the shared graph, formulated as follows:

$$\mathbf{F}_{\text{d}}^l = \mathbf{A}^\top \diamond \Phi(\mathbf{F}_{\text{up}}^{l+1}), \tag{9}$$

where $\mathbf{F}_{\text{d}}^l$ is the de-convolution output. For clarity, we denote output features of the $l$-th decoder stage as $\mathbf{F}_{\text{up}}^l$. For a model including $n$ stages, $\mathbf{F}_{\text{up}}^{n+1} = \mathbf{F}^{n+1}$. In practice, we find that not every point feature is aggregated in the down-sampling GCN layers. It means the graph de-convolution mentioned above will generate empty features in some up-sampled points, which affects the predictions. We illustrated the phenomenon in Figure 4 (b). To supplement the empty points, we add the trilinear interpolation to the de-convolution output, and the final up-sampling can be formulated as follows:

$$\mathbf{F}_{\text{up}}^l = \text{MLP}([\mathbf{F}_{\text{d}}^{l+1} + \text{interp}(\mathbf{F}_{\text{up}}^{l+1}), \mathbf{F}^l]), \tag{10}$$

where $\mathbf{F}_{\text{up}}^{l+1}$ is the input low-resolution features and $\mathbf{F}_{\text{d}}^{l+1}$ is the de-convolution output. $\mathbf{F}_{\text{up}}^l$ is the final up-sampled features. We concatenate the high-resolution features $\mathbf{F}^l$ from the encoder to implement skip connections.

### 3.4. Network Architecture

In this section, we give instructions on our network architecture for object part segmentation task and scene segmentation task. As shown in Figure 2, our model consists of a U-Net-like [38] feature extractor and a classifier as the segmentation head. We stacked GCN blocks and GAF blocks in the encoder of our model. The feature dimension in the blocks doubles after each down-sampling layer and halves after each up-sampling layer.

For scene segmentation tasks, four stages are constructed in our model. Following [33], the number of blocks in each stage is set to $[2, 4, 2, 2]$. We split the first two stages into the low-level feature learning phase (using GCN blocks) and the last two stages into the high-level feature learning phase (using GAF blocks). The initial feature dimension is set to 32. The segmentation head is a simple MLP with BatchNorm, ReLu and Dropout layer. To explore the potential of performance in big parameters, we also construct a large version of our model. The initial feature dimension is set to 64 and the number of blocks is set to $[3, 6, 3, 3]$.

| Method | Cat. mIoU | Ins. mIoU |
|---|---|---|
| PointNet [31] | 80.4 | 83.7 |
| PointNet++ [32] | 81.9 | 85.1 |
| DGCNN [46] | 82.3 | 85.2 |
| PointCNN [19] | 84.6 | 86.1 |
| PVCNN [24] | - | 86.2 |
| KPConv [44] | 85.0 | 86.2 |
| 3D-GCN [21] | 82.7 | 85.3 |
| PAConv [51] | 84.6 | 86.1 |
| AdaptConv [59] | 83.4 | 86.4 |
| CurveNet [49] | - | 86.8 |
| PointTrans. [58] | 83.7 | 86.6 |
| Strantified Trans. [15] | 85.1 | 86.6 |
| ScatterNet [22] | - | 86.7 |
| **Ours** | **85.3** | **87.0** |

Table 1. Results on ShapeNetPart for part segmentation.

For part segmentation tasks, four stages are constructed in our model. Due to the relatively small input point cloud, no GCN blocks are adopted, and only one GAF block is adopted in the third stage. The part segmentation head is constructed following CurveNet [49]. The initial feature dimension is set to 64 and the sampling rate in each stage is set to 2. A detailed illustration is included in the supplementary material.

## 4. Experiments

We evaluate our proposed AF-GCN on the object part segmentation task and 3D semantic segmentation task. For part segmentation, we use ShapeNetPart [54]. For semantic segmentation, we use three indoor/outdoor datasets, including S3DIS [1], ScanNetV2 [7] and Toronto-3D [41].

### 4.1. Part Segmentation

**Datasets.** We perform part segmentation on ShapeNetPart [54]. The ShapeNetPart dataset consists of 16,881 3D shapes from 16 categories, with 14,006 shapes for training and 2,874 for testing. There are 50 part categories in total and each 3D shape contains 2-6 parts.

**Implementation.** Following the previous works, we sample 2048 points from each shape in ShapeNetPart. The initial radius $r$ is set to 0.1m and is enlarged by $2.5\times$ after each down-sampling. The maximum number of edges $K$ is set to 32. We train our model for 300 epochs in four NVIDIA GeForce RTX3090 GPUs.

**Comparison result.** We compare our model with recent state-of-the-art methods and use category mean IoU and instance mean IoU for evaluation. As shown in Table 1, our method outperforms others both in category mIoU and instance mIoU in ShapeNetPart. We also give a visualization of our method in Figure 5.

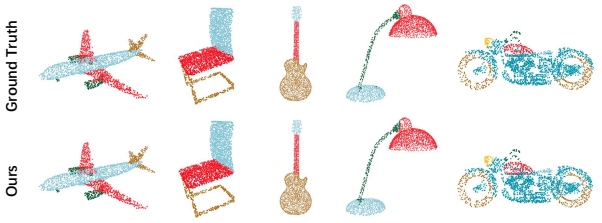


Figure 5. Visualization on ShapeNetPart. From left to right are aeroplane, chair, guitar, lamp and motorbike.

| Method | Input | Val mIoU | Test mIoU |
|---|---|---|---|
| PointNet++ [32] | point | 53.5 | 55.7 |
| PointCNN [19] | point | - | 45.8 |
| PointConv [47] | point | 61.0 | 66.6 |
| KPConv [44] | point | 69.2 | 68.6 |
| MinkowskiNet [6] | voxel | 72.2 | 73.6 |
| PointASNL [52] | point | 63.5 | 66.6 |
| SegGCN [17] | point | - | 58.9 |
| RandLA-Net [12] | point | - | 64.5 |
| JSENet [13] | point | - | 69.9 |
| Mix3d [28] | voxel | 73.6 | **78.1** |
| PointTrans. [58] | point | 70.6 | - |
| CBL [42] | point | - | 70.5 |
| RepSurf-U [35] | point | - | 70.0 |
| FastPointTrans. [30] | point | 72.4 | - |
| PTV2 [48] | point | **75.4** | 75.2 |
| **Ours**$^{\dagger}$ | point | 73.4 | 71.8 |

Table 2. Quantitative results on ScanNetV2 for semantic segmentation. More results are included in the supplementary material.

### 4.2. Semantic Segmentation

**Datasets.** We evaluate our method on S3DIS [1], ScanNetV2 [7] and Toronto-3D [41] for semantic segmentation. S3DIS is a challenging large-scale dataset for indoor scene segmentation, which includes 271 rooms in 6 areas. 273 million points are scanned in total, in which each point is annotated with one semantic label from 13 categories. The ScanNetV2 dataset consists of 1513 indoor point clouds for training and 100 point clouds for testing. It annotates each point with 21 categories. 242 million points are scanned by RGB-D cameras. We report implementation details and result for Toronto-3D in the supplementary material.

**Implementation.** For S3DIS, input points are grid sampled with the grid size set to 0.04m following previous work [58]. The sampling rate in each stage is set to 4. The initial radius $r$ is set to 0.1m, and it doubles after each down-sampling. For ScanNetV2, input points are grid sampled with the grid size set to 0.02m. The initial radius $r$ is set to 0.05m. For all datasets mentioned above, the maximum number of edges $K$ is set to 32. Our models are trained for 100 epochs in four NVIDIA GeForce RTX3090 GPUs.

| Method | Input | S3DIS 6-fold | | | S3DIS Area-5 | | |
|---|---|---|---|---|---|---|---|
| | | mIoU | mAcc | OA | mIoU | mAcc | OA |
| PointNet [31] | point | 47.6 | 66.2 | 78.5 | 41.1 | 48.9 | - |
| PointNet++ [32] | point | 59.9 | 66.1 | 87.5 | 56.0 | 61.2 | 86.4 |
| PointWeb [57] | point | 66.7 | 76.2 | 87.3 | 60.2 | 66.6 | 86.9 |
| KPConv [44] | point | 70.6 | 79.1 | - | 67.1 | 72.8 | - |
| MinkowskiNet [6] | voxel | - | - | - | 65.4 | 71.7 | - |
| PointASNL [52] | point | 68.7 | 79.0 | 88.8 | 62.6 | 68.5 | 87.7 |
| RandLA-Net [12] | point | 70.0 | 82.0 | 88.0 | 62.4 | 71.4 | 87.2 |
| PAConv [51] | point | 69.3 | 78.6 | - | 66.5 | 73.0 | - |
| PointTrans. [58] | point | 73.5 | 81.9 | 90.2 | 70.4 | 76.5 | 90.8 |
| CBL [42] | point | 73.1 | 79.4 | 89.6 | 69.4 | 75.2 | 90.6 |
| RepSurf-U [35] | point | 74.3 | 82.6 | 90.8 | 68.9 | 76.0 | 90.2 |
| PointNeXt-XL [33] | point | 74.9 | 83.0 | 90.3 | 71.1 | 77.2 | 91.0 |
| **Ours** | point | **77.7** | **85.1** | **91.7** | **72.3** | **77.9** | **91.1** |
| **Ours**$^{\dagger}$ | point | **78.4** | **86.2** | **91.8** | **73.3** | **79.3** | **91.5** |

Table 3. Quantitative results of semantic segmentation on S3DIS datasets (6-fold cross-validation and evaluation on Area 5). We compare with different methods in terms of mean per-class IoU (mIoU), mean per-class accuracy (mAcc), and overall accuracy (OA). Ours$^{\dagger}$ represent the large version of our model. We reported the best performance for comparison, and more detailed results are included in the supplementary material.

**Comparison result.** For S3DIS, we evaluate our method by 6-fold cross-validation and on Area-5. In Table 3, our proposed method outperforms previous methods in both Area-5 and 6-fold cross-validation. Performance achieved by the large version also shows the potential of our method in big parameters. As the qualitative results visualized in Figure 6, our method obtains a better performance, especially in the edges and corners. To further demonstrate the generalization of our proposed AF-GCN, we conduct extensive experiments on ScanNetV2. For ScanNetV2, our model achieves reasonable results and outperforms most point-based methods in the official validation set and online test set as shown in Table 2. PTV2 [48] achieves better performance for their partition-based pooling and well-designed transformer networks but requires more training time and larger training input.

### 4.3. Ablation Study

In Table 4, we conduct comprehensive ablation studies on S3DIS dataset to verify the effectiveness of each component in our method.

**Graph convolutions with Spatial Feature Projection.** In Table 4, we compare experiment I and II and find our graph convolutions with Spatial Feature Projection improve the baseline with 1.1% mIoU and 0.9% mAcc. Comparing experiment IV and V, we notice our method without SFP losses 1.9% mIoU and 1.4% mAcc in performance.

**Graph-shared down-sampling & up-sampling.** Comparing experiment II and III, we find adopting graph-shared down-sampling and up-sampling could provide a performance gain of 0.7% mIoU and 0.4% mAcc. Comparing

| ID | SFP | Graph-shared | GAF | mIoU | mAcc |
|---|---|---|---|---|---|
| I | | | | 69.3 | 75.7 |
| II | ✓ | | | 70.4 | 76.6 |
| III | ✓ | ✓ | | 71.1 | 77.0 |
| IV | ✓ | ✓ | ✓ | **72.3** | **77.9** |
| V | | ✓ | ✓ | 70.4 | 76.5 |
| VI | ✓ | | ✓ | 71.6 | 77.3 |

Table 4. Ablation studies conducted on S3DIS Area-5. SFP: Graph convolutions with Spatial Feature Projection. Graph-shared: Graph-shared down-sampling and up-sampling. GAF: Graph Attentive Filter Block. Metric: mIoU, mAcc.

| ID\stage | 1 | 2 | 3 | 4 | mIoU | mAcc | OA |
|---|---|---|---|---|---|---|---|
| I | ● | ● | ● | ● | 71.1 | 77.0 | 91.0 |
| II | ● | ● | ● | ☆ | 71.5 | 77.1 | **91.1** |
| III | ● | ● | ☆ | ☆ | **72.3** | **77.9** | **91.1** |
| IV | ● | ☆ | ☆ | ☆ | 69.3 | 75.2 | 90.5 |
| V | ☆ | ☆ | ☆ | ☆ | 70.1 | 77.1 | 89.8 |

Table 5. Ablation study on feature learning phases division. ● denotes GCN Blocks. ☆ denotes GAF Blocks.

experiment IV and VI, our method without graph-shared down-sampling and up-sampling losses 0.7% mIoU and 0.6% mAcc in performance. Note that the parameters of our model are only increased by around 4% parameters (0.35M) using the graph-shared down-sampling and up-sampling.

**Graph Attentive Filter.** Comparing experiment III and IV, using Graph Attentive Filter blocks, the model achieves higher mIoU and mAcc performance. Further, we conduct ablation experiments on the division of the feature learning phases in Table 5. We achieve the best performance when

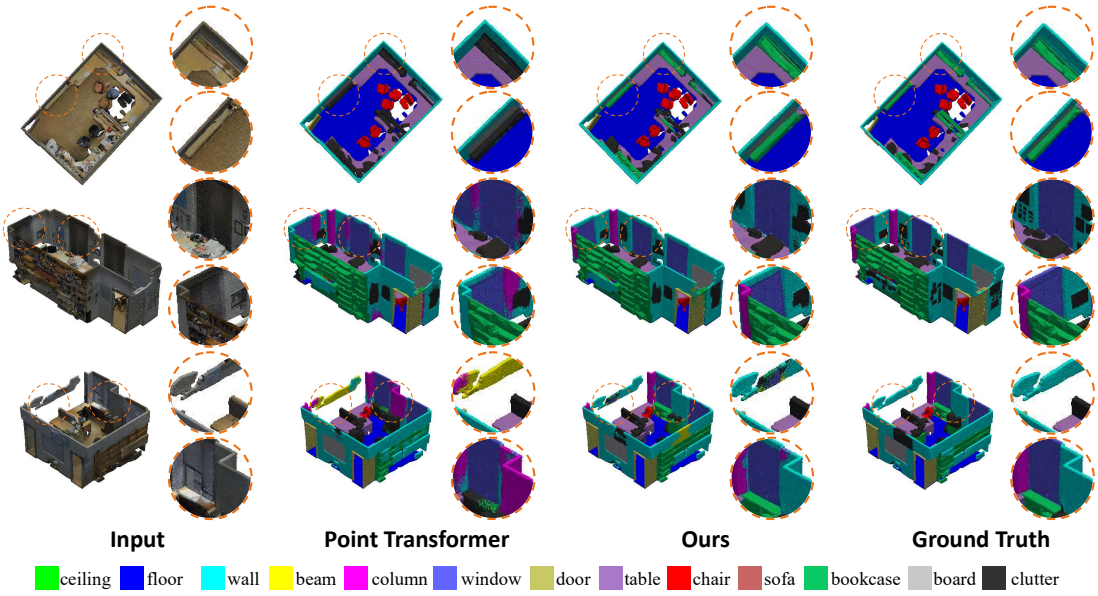


Figure 6. Visualization of semantic segmentation on S3DIS. We compare our method with the previous self-attention-based SOTA point transformer [58]. Our method achieves better performance especially in the corners and edges due to better preservation of local geometries.

| Method | None | Jitter. | 90° | 180° | ×0.8 | ×1.2 | +0.2 |
|---|---|---|---|---|---|---|---|
| PointNet++ [32] | 59.75 | 59.05 | 58.15 | 57.18 | 56.24 | 59.74 | 22.33 |
| PAConv [51] | 65.63 | 65.12 | 61.66 | 63.48 | 64.20 | 63.94 | 55.81 |
| PointTrans [58] | 70.36 | 59.67 | 65.94 | 67.78 | 65.73 | 66.15 | 70.44 |
| Ours | **72.34** | **72.40** | **72.27** | **72.30** | **72.37** | **67.98** | **72.90** |

Table 6. Robustness study on S3DIS Area-5.

adopting GCN blocks in the first two stages and adopting GAF blocks in the last two stages. The empirical results also validate graph convolution is more suitable for capturing local geometry information compared with self-attention mechanisms.

### 4.4. Robustness Study

We conduct evaluations in jitter, rotation, scale and shift scenarios on S3DIS to verify our model's robustness to various perturbations. As shown in Table 6, our proposed model is robust to various perturbations compared with previous methods such as PAConv and PointTransformer.

### 4.5. Efficiency

We demonstrate the training speed and inference speed comparison with previous methods such as PAConv [51], PointTransformer [58] and PointNext-XL [33] in Table 7. We take 16×15,000 points to evaluate the inference speed in an NVIDIA GeForce RTX3090 GPU. Note that our method is 28.4% faster than Point Transformer in inference and about 5× faster in training, while the large version has a comparable inference speed to PointNext-XL.

| Method | Train. Speed (ins./sec.) | Infer. Speed (ins./sec.) | mIoU | mAcc |
|---|---|---|---|---|
| PAConv [51] | - | 59.5 | 66.5 | 73.0 |
| PointTrans. [58] | 7.0 | 66.8 | 70.4 | 76.5 |
| PointNext-XL [33] | 32.0 | 33.4 | 71.1 | 77.2 |
| Ours | **38.3** | **85.8** | **72.3** | **77.9** |
| Ours† | 21.8 | 37.2 | **73.3** | **79.3** |

Table 7. Efficiency study evaluated on S3DIS Area-5.

## 5. Conclusion

In this paper, we propose a hybrid network framework AF-GCN. By using GCN blocks and well-designed GAF blocks in a phased manner, our framework is able to significantly preserve the geometric information and efficiently aggregate the semantic information. In addition, the spatial feature projection module explicitly boosts the validity of spatial information in feature aggregation, and the graph-shared down-sampling and up-sampling modules are able to align the topological information between up-sampling and down-sampling, thus maintaining a consistent feature representation. The experimental results under several generic segmentation datasets validate the performance and generality of our proposed AF-GCN. Robustness and efficiency experiments also verify the superiority of our AF-GCN.

## Acknowledgements

This work was supported by the National Key R&D Program of China (No. 2020AAA0103501).

## 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 (CVPR)*, June 2016. 6

[2] Yizhak Ben-Shabat, Michael Lindenbaum, and Anath Fischer. 3dmfv: Three-dimensional point cloud classification in real-time using convolutional neural networks. *IEEE Robotics and Automation Letters*, 3(4):3145–3152, 2018. 2

[3] Prarthana Bhattacharyya, Chengjie Huang, and Krzysztof Czarnecki. Sa-det3d: Self-attention based context-aware 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3022–3031, 2021. 3

[4] Alexandre Boulch, Bertrand Le Saux, and Nicolas Audebert. Unstructured point cloud semantic labeling using deep segmentation networks. *3dor@ eurographics*, 3:1–8, 2017. 2

[5] Jaesung Choe, Chunghyun Park, Francois Rameau, Jaesik Park, and In So Kweon. Pointmixer: Mlp-mixer for point cloud understanding. In *European Conference on Computer Vision*. Springer, 2022. 5

[6] 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 (CVPR)*, June 2019. 2, 6, 7

[7] 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 (CVPR)*, July 2017. 6

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 1, 2, 4

[9] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, Apr 2021. 1, 2, 3

[10] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4338–4364, 2020. 1

[11] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3464–3473, 2019. 1, 2

[12] 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 (CVPR)*, June 2020. 2, 6, 7

[13] Zeyu Hu, Mingmin Zhen, Xuyang Bai, Hongbo Fu, and Chiew-lan Tai. Jsenet: Joint semantic segmentation and edge detection network for 3d point clouds. In *European Conference on Computer Vision*, pages 222–239. Springer, 2020. 6

[14] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 984–993, 2018. 2

[15] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8500–8509, June 2022. 1, 2, 3, 6

[16] Felix Järemo Lawin, Martin Danelljan, Patrik Tosteberg, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Deep projective 3d semantic segmentation. In *International Conference on Computer Analysis of Images and Patterns*, pages 95–107. Springer, 2017. 2

[17] Huan Lei, Naveed Akhtar, and Ajmal Mian. Seggcn: Efficient 3d point cloud segmentation with fuzzy spherical kernel. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 6

[18] Guohao Li, Matthias Müller, Guocheng Qian, Itzel Carolina Delgadillo Perez, Abdulellah Abualshour, Ali Kassem Thabet, and Bernard Ghanem. Deepgcns: Making gcns go as deep as cnns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1, 2

[19] 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*, volume 31, 2018. 2, 6

[20] 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 (CVPR)*, pages 4293–4302, 2020. 2

[21] Zhi-Hao Lin, Sheng-Yu Huang, and Yu-Chiang Frank Wang. Learning of 3d graph convolution networks for point cloud analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4212–4224, 2021. 1, 2, 6

[22] Qi Liu, Nianjuan Jiang, Jiangbo Lu, Mingang Chen, Ran Yi, and Lizhuang Ma. Scatternet: Point cloud learning via scatters. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 5611–5619, New York, NY, USA, 2022. Association for Computing Machinery. 6

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021. 1, 2

[24] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 2, 6

[25] Zhe Liu, Xin Zhao, Tengteng Huang, Ruolan Hu, Yu Zhou, and Xiang Bai. Tanet: Robust 3d object detection from point

clouds with triple attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11677–11684, 2020. 3

[26] 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

[27] Hsien-Yu Meng, Lin Gao, Yu-Kun Lai, and Dinesh Manocha. Vv-net: Voxel vae net with group convolutions for point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8500–8508, 2019. 2

[28] Alexey Nekrasov, Jonas Schult, Or Litany, Bastian Leibe, and Francis Engelmann. Mix3D: Out-of-Context Data Augmentation for 3D Scenes. In *International Conference on 3D Vision (3DV)*, 2021. 6

[29] Dong Nie, Rui Lan, Ling Wang, and Xiaofeng Ren. Pyramid architecture for multi-scale processing in point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17284–17294, June 2022. 3

[30] Chunghyun Park, Yoonwoo Jeong, Minsu Cho, and Jaesik Park. Fast point transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16949–16958, June 2022. 3, 6

[31] 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 (CVPR)*, pages 652–660, July 2017. 2, 6, 7

[32] 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*, volume 30, 2017. 2, 6, 7, 8

[33] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In *Advances in Neural Information Processing Systems*, 2022. 2, 5, 7, 8

[34] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 1, 2

[35] Haoxi Ran, Jun Liu, and Chengjie Wang. Surface representation for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18942–18952, June 2022. 2, 6, 7

[36] Haoxi Ran, Wei Zhuo, Jun Liu, and Li Lu. Learning inner-group relations on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15477–15487, October 2021. 1, 2, 3

[37] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3577–3586, 2017. 2

[38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 5

[39] Jiayao Shan, Sifan Zhou, Zheng Fang, and Yubo Cui. Ptt: Point-track-transformer module for 3d single object tracking in point clouds. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1310–1316. IEEE, 2021. 3

[40] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 945–953, 2015. 2

[41] Weikai Tan, Nannan Qin, Lingfei Ma, Ying Li, Jing Du, Guorong Cai, Ke Yang, and Jonathan Li. Toronto-3d: A large-scale mobile lidar dataset for semantic segmentation of urban roadways. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 6

[42] Liyao Tang, Yibing Zhan, Zhe Chen, Baosheng Yu, and Dacheng Tao. Contrastive boundary learning for point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8489–8499, June 2022. 2, 6, 7

[43] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3887–3896, 2018. 2

[44] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, Francois Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1, 2, 6, 7

[45] 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*, volume 30, 2017. 2

[46] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 1, 2, 6

[47] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9621–9630, 2019. 2, 6

[48] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. In *Neural Information Processing Systems*, 2022. 6, 7

[49] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 915–924, October 2021. 5, 6

[50] 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 (CVPR)*, June 2018. 2

[51] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3173–3182, June 2021. 1, 2, 5, 6, 7, 8

[52] 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 (CVPR)*, June 2020. 1, 3, 6, 7

[53] Zetong Yang, Yin Zhou, Zhifeng Chen, and Jiquan Ngiam. 3d-man: 3d multi-frame attention network for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1863–1872, 2021. 3

[54] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016. 6

[55] Cheng Zhang, Haocheng Wan, Xinyi Shen, and Zizhao Wu. Patchformer: An efficient point transformer with patch attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11799–11808, June 2022. 1, 3

[56] 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 (CVPR)*, pages 10076–10085, 2020. 1, 2, 4

[57] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. Pointweb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 7

[58] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H.S. Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16259–16268, October 2021. 1, 2, 3, 4, 5, 6, 7, 8

[59] Haoran Zhou, Yidan Feng, Mingsheng Fang, Mingqiang Wei, Jing Qin, and Tong Lu. Adaptive graph convolution for point cloud analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4965–4974, October 2021. 1, 2, 5, 6