DSPoint: Dual-scale Point Cloud Recognition with High-frequency Fusion

Renrui Zhang*[1], Ziyao Zeng*[2,3], Ziyu Guo*[4], Xinben Gao[2]
Kexue Fu[1], Jianbo Shi†[2,6]
[1]Shanghai AI Laboratory [2]UISEE [3]ShanghaiTech University
[4]Peking University [6]University of Pennsylvania
zhangrenrui@pjlab.org.cn, zengzy@shanghaitech.edu.cn
jshi@seas.upenn.edu

Abstract

*Point cloud processing is a challenging task due to its sparsity and irregularity. Prior works introduce delicate designs on either local feature aggregator or global geometric architecture, but few combine both advantages. We propose **D**ual-**S**cale Point Cloud Recognition with High-frequency Fusion (**DSPoint**) to extract local-global features by concurrently operating on voxels and points. We reverse the conventional design of applying convolution on voxels and attention to points. Specifically, we disentangle point features through channel dimension for dual-scale processing: one by point-wise convolution for fine-grained geometry parsing, the other by voxel-wise global attention for long-range structural exploration. We design a co-attention fusion module for feature alignment to blend local-global modalities, which conducts inter-scale cross-modality interaction by communicating high-frequency coordinates information. Experiments and ablations on widely-adopted ModelNet40, ShapeNet, and S3DIS demonstrate the state-of-the-art performance of our DSPoint. Our code is available at: https://github.com/Adonis-galaxy/DSPoint.*

1. Introduction

3D vision has drawn increasing attention recently with the rapid development of 3D sensing technologies. It brings out many challenging 3D tasks, such as point cloud recognition([16, 24, 46]), shape [52] and scene [27, 51, 53] segmentation, object detection based on point cloud [7, 29, 30, 38, 56] and monocular image [20, 42, 47], point cloud registration [1, 40, 50]. Unlike 2D images that consist of pixels in uniform grids, a 3D point cloud is permutation invariant, spatially irregular, and density varying, which leads to non-trivial difficulty for algorithm designs.

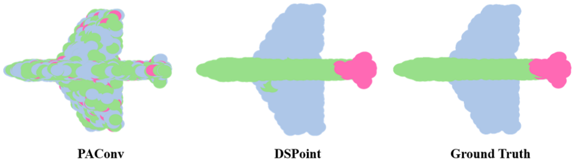

Figure 1. Visualization of our method on ShapeNet [45] Part Segmentation, compared with PAConv [48]. It shows our advantage in segmenting parts into spatial-consistent regions.

Point cloud methods can be divided into two groups: projection-based [4, 18, 22, 26, 28, 45] and point-wise [11, 14, 21, 23, 43] methods. Projection-based models convert points into a regular grid representation, such as multi-view images [4, 26, 28] or voxels [18, 22, 45], so that convolution models [18] can be used directly for recognition. However, voxelizations lose local shape details and suffer from heavy memory and computation costs. In contrast, point-wise methods require no modal transformation and thus maintain all original information, especially the fine-grained structure. PointNet [21] encodes each point with Multi-layer Perceptron (MLP) and eliminates the unordered set problem via max-pooling operation. PointNet++ [23] further introduces the hierarchical architecture for point cloud's local feature aggregation. Point-wise convolution proposed by PointCNN [11], PAConv [48] and others [8] construct permutation-invariant convolution.

Local and global features capture different aspects of the shape. The key question is: how to integrate local and global information while maintaining separate processing to prevent over smoothing between them.

From a representation perspective, projection-based representations are better suited for global part-whole structure relationships, while point representations have advantages on local parsing of shape details. Motivated by this, PVCNN [15] designs a point-voxel module to parallelly encode point clouds from dual modalities (voxels and points),

* indicates equal contributions. † indicates the corresponding author.

by leveraging 3D convolution for the voxel branch and point-wise MLP for the point branch.

The conventional wisdom of applying local convolution on voxel and global attention on points makes sense from a computational viewpoint. Still, it achieves the opposite goal of extracting global structure from voxel representation and local shape information from the point representation. Furthermore, the simple combination (by addition) of two modalities (local and global) could blur out local details.

We propose a reverse design, where we apply a global process on voxels to extract long-range structure relationships and a local one on the points to compute detail shape features. We call our **D**ual-**S**cale network for Point cloud understanding with high-frequency encoding, as **DSPoint**. Specifically, we disentangle the point representation along the channel dimension: one part encoding local feature and the other part for the global feature. In each processing block, local channels are processed by point-wise dynamic convolution [48], and the global ones are firstly converted to voxelized representation and then parsed via global attention mechanism [3].

From the representation perspective, illustrated in Figure 2, one can compute fine-grained geometry features from 3D points locally. At the same time, the voxelization process naturally aggregates neighboring points' features and is suitable for global part-whole structural relationship reasoning. From the computation view, convolution is natural for local feature aggregation, but attention is designed for long-range dependency modeling. Consequently, processing such two modalities with convolution on the local point level and attention on the voxel level could be a good choice for point cloud understanding.

After the concurrent pathways, the voxel modality is back-projected to points by assigning each voxel's feature to every point within it. Here, we obtain two part-channel representations of each point: voxel-wise global feature and point-wise local feature. We observe that the naive addition of different modalities often results in feature misalignment and blurring. To effectively exchange local-global information, we build on a recently introduced Dual-stream Net(DS-Net) [17] co-attention design. In this design, the global part configuration features serve as 'query' for the local shape feature 'keys', and vice versa. However, the direct application of DS-Net is still insufficient for removing across modality mis-alignment. This is because shape features of different points from one voxel usually are homogeneous, and directly fusing them with heterogeneous point-wise shape features would bring about ambiguities. Borrowing the high-frequency point encoding concept in NeRF [19], we encode each voxel's coordinate into high-frequency representation and integrate it with point-wise features. Aided by this inter-path coordinates communication, local-global shape features from dual modalities (voxels and points) can be highly aligned. The visual example in Figure 1 illustrates the effectiveness of our 'reverse' design of using a 3D grid for global attention and local convolution for 3D raw points.

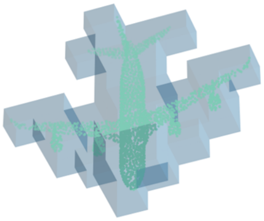

Figure 2. Dual modalities (voxels and points) processing: grey voxels aggregate points to represent the plane's global structure coarsely, while green 3D points describe subtle local shape details.

We summarize the contributions as below:

- We propose DSPoint, which concurrently processes point cloud with dual scales and modalities for robust local-global features extraction.
- A high-frequency fusion module is introduced by communicating high-dimensional coordinates information between voxel-wise and point-wise features.
- To illustrate our model's superiority, we experiment DSPoint on shape classification, shape and scene segmentation, respectively on ModelNet40, ShapeNet, and S3DIS datasets.

2. Related Work

Deep learning for Point Cloud. Projection-based models and point-wise models are two main branches of deep learning in 3D, distinguished by their data processing modality. Some of the projection-based models project raw points onto a set of image planes with pre-defined [32] or learnable [9] viewpoints and then utilize 2D convolutions for robust feature extraction. One can combine images from different views so to minimize the information loss on the original point cloud. Still, the complex and time-consuming projection process makes this approach unpractical for real-time applications. Alternatively, some approaches transform points into spatial voxels, such as VoxelNet [56] and [25, 31], which are uniform grid-based representations and thus can be applied 3D convolutions [18] or attention mechanism [3]. However, voxel-based networks confront information loss due to low-resolution quantization. It has an inpractical cubically growing running time. Point-wise networks directly process raw points with irregular distribution over 3D space. PointNet [21] leverage Multi-layer Perceptron(MLP) to extract point-wise features and integrate them with a global pooling. PointNet++ [23] proposes a hierarchical PointNet [21] architecture to capture local contexts with sampling and grouping blocks. DGCNN [41], KPConv [36] and PAConv [48] further design convolutions on spatial points for better local geometry encoding. To aggregate both advantages, our DSPoint adopts dual-path architecture to concurrently encode point features with voxel branch and point branch, respectively, for understanding global and local features.

Dual-path Networks. Constructing multiple pathways for 2D deep learning has been explored by GoogLeNet [34], EfficientNet [35], and MaX-DeepLab [39], in which different paths with varying feature resolutions and convolutional kernels are expected to extract distinct aspects of features. Recently, DS-Net [17] and LaMa [33] have designed more delicate dual-path architectures for local and global features encoding, which successively separate and fuse the two representations for sufficient cross-scale interactions. For point cloud processing, DTNet [6] proposes to apply a multi-head attention mechanism in transformer [37] for extracting both inter-channel and inter-point features. PVCNN [15] utilizes two modalities to capture point features: raw points and voxels concurrently. Therein, PVCNN encodes each raw point with MLP and each voxel with 3D convolution [15] for inter-point relation modeling. Contrary to PVCNN's design, we apply point-wise convolution on neighboring 3D points for local features and global attention on all voxels for global features extraction.

High-frequency Spatial Embedding. Deep neural networks tend to focus more on low-frequency features but neglect the higher frequency counterparts according to [19], and mapping the low dimensional data into higher ones can facilitate networks for better learning abilities. NeRF [19] introduces a high-frequency embedding function in a non-parametric manner. Combining this transformation with learnable MLP could improve the performance since the network can capture slight color and geometry variation in the 3D space. The positional encoding module also utilizes the function in transformer [37], which maps two or three channels' coordinates into higher dimensions. In DSPoint, we adopt it for encoding 3D coordinates of dual paths and implement cross-modality coordinates interaction.

3. Method

In this section, we present our dual-scale network with high-frequency fusion (DSPoint) (Figure3). In Section 3.1, we first briefly review Dual-stream Net [17] for 2D recognition. Then introduce our Dual-scale Blocks in Section 3.3. In Section 3.4, we show the high-frequency fusion module for better features alignment.

3.1. Review of Dual-stream Net

Conventional deep neural networks for 2D recognition utilize single-stream architectures to encode the image, in which shallow layers focus on extracting fine-grained features by local convolutional kernels, and deep layers aim at capturing global representations with a large receptive field. However, local and global features describe the image from two different perspectives: one texture details and the other for long-range shape structure. Dual-stream Net [17] (DS-Net) proposes to maintain separate local and global visual representations, treat them equally while concurrently exchanging information between them.

Computationally, DS-Net [17] splits the image feature along the channel dimension into two parts: f_l and f_g for dual-pathway processing. Local features of f_l remain in the high resolution to preserve the visual details and are encoded by convolution layers, Global features of f_g are downsampled to a smaller grid to filter out the low-level noise and extracted by global attention mechanism [3]. We formulate the parallel propagation as

$$f_L = \text{Convolution}(f_l); \quad f_G = \text{Attention}(f_g), \qquad (1)$$

where f_L and f_G are the specific-encoded features of f_l and f_g. Then, f_G are upsampled to the original feature resolution and conduct local-global fusion with f_L. For better blending between the two representations, DS-Net further adopts co-attention mechanism for inter-scale features alignment. During implementations of attention, supposing there are n local and m global features, f_L and f_G respectively serve as queries and extract informative features from each other by the affinity matrixes, $A_{L\to G} \in \mathbb{R}^{n\times m}$ and $A_{G\to L} \in \mathbb{R}^{m\times n}$, denoted as

$$h_L = A_{L\to G} f_G; \quad h_G = A_{G\to L} f_L, \qquad (2)$$

where h_L and h_G denote the hybrid local and global features after alignment. Finally, the two representations are concatenated together and fused by a linear layer. In this dual-stream design, DS-Net obtains robust visual representation and achieves high image classification performance.

3.2. Dual-scale Processing

Local-global Disentangling. In a point cloud, global information mainly contains overall shape properties and inter-component relationships, but local information focuses on subtle spatial geometry and density variations. Following DS-Net [17], in every processing stage, we disentangle global and local features along the channel dimension. Specifically, given a C-channel point feature, we split it into f_l with $\alpha_l C$ channels and f_g with $\alpha_g C$ channels, where α_l and α_g weighing the importance between two representations and $\alpha_l + \alpha_g = 1$. To reserve local details, we

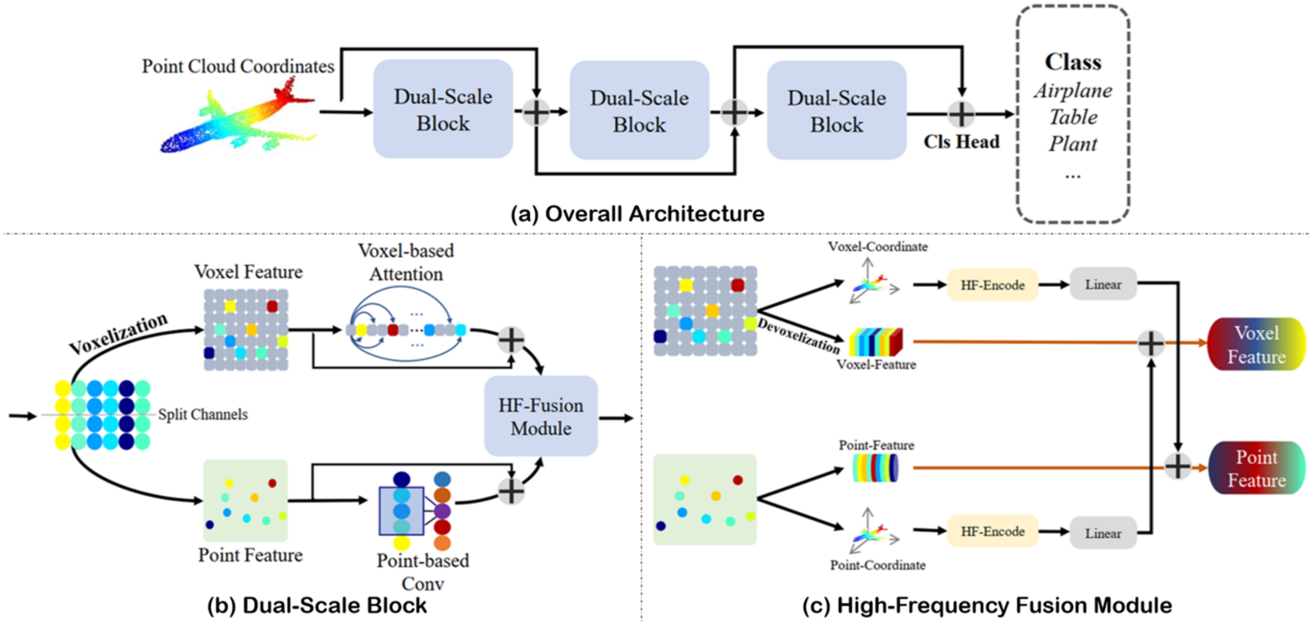

Figure 3. (a): Input coordinates will pass through three Dual-scale blocks with residual connection, then feed through a classification head to obtain shape classification prediction. (b): We split features along channels and pass in through the voxel-based global attention and point-based local convolution branches, respectively. Then, fuse two features with high-frequency module. (c): We encode coordinates of one modality with a high-frequency embedding function and inject into features of another branch to help feature alignment.

maintain the points' spatial density for f_l. As for downsampling f_g, we convert the irregular points into grid-form low-resolution voxels [15]. The voxelization averages all point features whose coordinates fall into the voxel grid. Compared to other downsampling methods in point clouds, such as farthest point sampling (FPS) [23], voxelization is more stable without any randomness and has the reversibility for devoxelization back to points. Also, representations from another modality could capture features from diverse aspects and thus leads to better feature extraction. Therefore, we select voxels for points' global representation. After the disentangling, we concurrently conduct the dual-scale propagation of global-local features encoding.

3.3. Dual-scale Block

Demonstrated in Figure 3(a), our pipeline consists of three consistent Dual-scale Blocks with residuals. Input coordinates will pass through three blocks to obtain a feature representation, then a classification head will make prediction. The structure of Dual-scale Block is shown in Figure 3(b). In each block we split features channel-wise and feed one part through the point-based convolution, and the other part through voxel-based attention branches. In point-based convolution branch, we directly apply existing 3D convolution like PAConv [48]. In voxel based attention branch, we voxelized points according to PVCNN [15], encode voxel coordinates using a linear layer and add to voxel features, employ a layer normalization, apply self-attention between all voxels, then devoxlied voxels to restore features. Two branches has residual connection inside shown in Figure 3(b). Details of two branches will be presented below. Last, we fuse two processed features with high-frequency module (Figure 3(c)), introduced in section 3.4. Implementation details are listed in section 4.

Point-scale Local Encoding. Convolution is natural for local feature extraction because it encodes translation-invariant properties and its limited receptive field makes it easier to compute and learn. We use the point-specific convolution operations from [8]. For a point, (x, y, z) with the local receptive field containing k points, the convolutional kernel is dynamically generated by their relative coordinates via a Multi-layer Perceptron (MLP). We formulate the convolution operation, which transforms $f_l(x, y, z)$ into encoded local feature $f_L(x, y, z)$ as

$$f_L(x, y, z) = \sum_{i=1}^{k} W(x_i, y_i, z_i) \odot f_l(x_i, y_i, z_i), \tag{3}$$

where $W(x_i, y_i, z_i)$ denotes the predicted kernel weight of neighboring point i, and $\odot$ denotes element-wise product. After the point-scale convolution, f_L at each point location contains local features capturing fine-grained information.

Voxel-scale Global Encoding. Attention mechanism [3] operates on the entire visual domain and conducts information interaction over long distances, which is good at summarizing overall structural shape properties. Therefore, we apply a multi-head attention mechanism over the voxel-scale branch for global features exploration. Supposing there are r voxels, We denote the encoded global feature for voxel (x, y, z) as

$$f_G(x, y, z) = A(x, y, z) f_g(x, y, z), \tag{4}$$

where $A(x, y, z) \in \mathbb{R}^{1\times r}$ denotes affinity matrix between the voxel (x, y, z) with all other voxels. Because of the coarser resolution of transformed voxel grids, attention's computation and memory costs are much manageable. In addition, without low-level spatial detail distractions, f_G can concentrate on long-range part-whole object structure relationships. Afterward, the devoxelization is conducted to project low-density voxels back to the original points, during which the voxel feature is assigned to each point within.

3.4. Fusion with High-frequency Function

We have obtained the separately encoded f_L and f_G for the point cloud and require effective fusion of the two representations. However, different from 2D images, the misalignment problem in 3D lies mainly in the mismatch of spatial locations, since f_G is regional homogeneous due to devoxelization process, but f_L is relatively point-wise heterogeneous. Besides, these modality transformations are implemented through approximation and will cause features to lose their high-frequency information. To alleviate these problems, we proposed High-frequency Fusion Module (Figure 3(c)), injecting coordinates information of another modality through high-frequency fusion to help cross-modality alignment.

We refer to high-frequency functions proposed in NeRF [19], which maps a low-frequency point coordinates into higher-dimensional vectors via a set of trigonometric functions:

$$\begin{aligned}\gamma(c) = (c, \sin(2^0\pi c), \cos(2^0\pi c), \cdots, \\ \sin(2^{L-1}\pi c), \cos(2^{L-1}\pi c))\end{aligned} \tag{5}$$

Here γ is a function mapping point coordinates c from $\mathbb{R}$ into a higher dimensional space $\mathbb{R}^{2L}$, dimension changes from 3 to 3L+3. By combining this non-parametric transformation with the learnable MLP, the issue of neglecting low-frequency information by the deep network would be largely relieved, such as subtle variations on local geometric and density.

Specifically, we denote all the coordinates of voxels from the voxel-wise branch as Coords_v, and all those of points from the point-wise branch as Coords_p. Respectively, we encode them via high-frequency function $\gamma(\cdot)$ as

$$\mathrm{hf}_v^G = \gamma(\mathrm{Coords}_v);\ \ \mathrm{hf}_p^L = \gamma(\mathrm{Coords}_p), \tag{6}$$

Method	Input	Accuracy
	Local Feature	
PointNet [21]	xyz	89.2
PointNet++ [23]	xyz	90.7
DGCNN [41]	xyz	92.9
KPConv [36]	xyz	92.9
FPConv [12]	xyz	92.5
PAConv (*PN) [48]	xyz	93.2 (92.5)
PAConv (*DGCNN) [48]	xyz	93.6 (93.4)
	Global Feature	
PCT [5]	xyz	93.2 (92.8)
PT [55]	xyz+nor	**93.7**
	Global-Local Feature	
PointASNL [49]	xyz+nor	93.2
Ours	xyz	93.5

Table 1. Results of Object Classification on ModelNet40 [45]. We only train one model instead of using multiple models ensemble. (”nor” indicates using extra normal vector information as input, *PN denotes using PointNet as the backbone, Results in brackets are our re-implementation results)

where hf_v^G and hf_p^L represent the high-frequency encoded voxels'(globle) and points'(local) coordinates, whose channels are the same as f_L and f_G. Then, we aggregate the voxel-related hf_v with point-wise features f_L, and the point-related hf_p with voxel-wise features f_G, both with simple addition. On top of that, a linear layer is applied for respectively blending and transforming dimension of the above paired voxel/point-coordinate high-frequency features with point/voxel-wise features, formulated as

$$\begin{aligned} h_L &= f_L + \mathrm{Linear}(\mathrm{hf}_v^G); \\ h_G &= f_G + \mathrm{Linear}(\mathrm{hf}_p^L), \end{aligned} \tag{7}$$

where h_L and h_G denote the hybrid features of the local and global features. This cross-scale communication of high-frequency coordinate information can mitigate the misalignment issue because hf_v brings about homogeneous alignment for local feature f_G and, while hf_p carries heterogeneous discrimination for f_L. Finally, we concatenate h_L and h_G through the channel dimension and apply a linear layer for the final fusion, which restore point features into the original C channels. After this, the dual-scale branches are combined into one, and the local-global features of point clouds can be well extracted for better 3D understanding.

4. Experiments

4.1. Shape Classification

We evaluate on ModelNet40 [45] dataset for object classification. This dataset contains 40 categories of 12,311

	Point + Global	Voxel + Gobal
Point + Local	93.2	**93.5**
Voxel + Local	93.0	93.1

Table 2. Different modalities for dual-scale processing. (Point / Voxel: point / voxel-based representation. Local / Local: local / global feature processing.)

Local Operator	Accuracy
DSPoint w. MLP	91.2
DSPoint w. MLP + SG	92.4
DSPoint w. KPConv [36]	93.2
DSPoint w. PointConv [43]	92.8
DSPoint w. PAConv [48]	**93.5**

Table 3. Ablation of local operator. We use different local feature operator in our local branch and evaluate its performance. MLP stands for shared MLP from PointNet [21]. SG stands for sample and grouping from PCT [5].

meshed CAD models, 9,843 of them are used for training and the rest 2,468 for testing. We follow the same data preprocessing in PointNet [21]: for each model, we sample the first 1,024 points and apply dropout points, random translation and shuffling all points. We only employ coordinates information as input, without extra use of normal vectors.

4.1.1 Experiment Setting

Model Architecture. Shown in Figure 3(a), our DSPoint consists of 3 consecutive Dual-scale Blocks with skip-connection, and feeds features into a classfication head to obtain prediction. The channel dimension for each block is 64, 64, 128, respectively. Voxelization resolution for each block is 8, 6, 4, respectively. We incorporate PAConv [48] as our local feature extraction operator, with 8 nearest neighbors and 8 weight matrices. The channel ratio between the local and global branches is 3:1, and L of high-frequency encoding is 10. The classification head has a Linear layer to project embedding dimension from 128 to 1024, followed by a max pooling layer to aggregate all points, then two Linear layers will project features' embedding from 1024 to 512, and 512 to 40, which is the number of classes. Then we apply a softmax to obtain classification score. Linear layers are all connected with batch normalization and ReLU activation function. To be environmental-friendly, we do not train massive amounts of models then use their ensembled score, but train only one model.

Training Setting. We use Adam optimizer, and train our model for 250 epochs and preserve the model with the best evaluation accuracy during training. During training, we set the batch size to 32, learning rate to 0.001, weight decay to 1e-4, and reduce learning rate when a metric has stopped improving at the factor of 0.5, the patience of 10, and a minimum learning rate of 0.00001.

	Global + Front	Global + Back	Global + None
Local + Front	93.2	93.4	93.0
Local + Back	93.3	**93.5**	93.1
Local + None	92.9	93.2	92.7

Table 4. Ablation of High-Frequency Fusion. (Local / Global: local or global branch. Front / Back: put High-Frequency Fusion module before/after the feature processing. None: don't use High-Frequency Fusion module.)

Method	Param.	Latency
PointNet [21]	3.47M	**13.6**
PointNet++ [23]	1.74M	35.3
DGCNN [41]	1.81M	85.8
KPConv [36]	-	120.5
FPConv [12]	-	-
PAConv (*PN) [48]	-	-
PCT [5]	2.88M	92.4
PT 2021 [55]	-	530.2
PointASNL [49]	-	923.6
Ours	**1.16M**	214.5

Table 5. Efficiency evaluation measured on ModelNet40

4.1.2 Performance

The result of classification experiments on ModelNet40 is shown in Table 1. We list previous works based on the feature representation they are using. The local feature means they only process local features around each point during inference, such as PointNet [21] or KPConv [36]. Global feature process points globally and builds long-range dependency using attention mechanisms, such as PCT [5]. There also exist other methods that process local and global features simultaneously, such as PointASNL [49]. Results show that our method outperforms or is comparable with all previous methods.

4.2. Ablation Study

To quantify the effectiveness of DSPoint, we conduct ablation studies on ModelNet40 [45], following the same experiment setting of Shape Classification mentioned above.

Dual-scale Modality Our DSPoint incorporates point-based representation to process local information and utilize voxel-based representation to handle long-range global dependency. We claim that local voxel-based representation pools neighborhood information together in low reso-

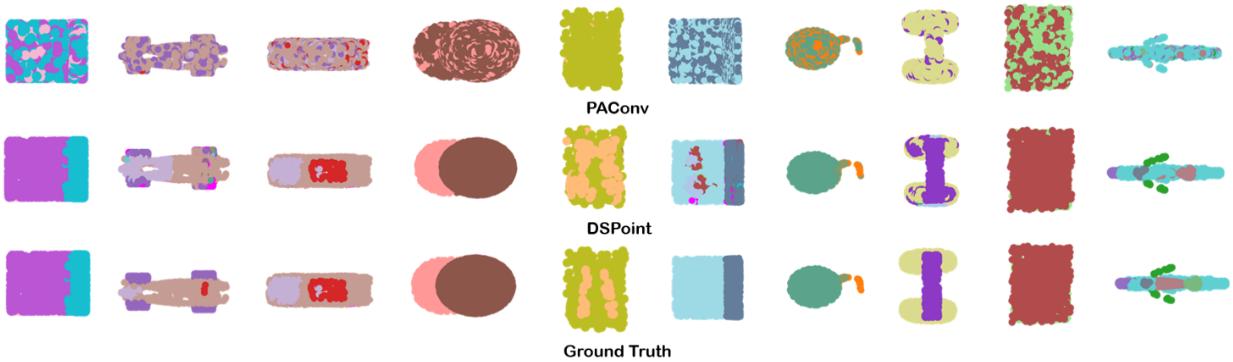

Figure 4. Visualization Results of ShapeNet [45]. It demonstrates our compared baseline PAConv [48] (first row), our method DSPoint (second row), and ground truth, which indicates our excellent performance of spatial continuity on part segmentation.

Method	Cls. mIoU	Ins. mIoU	airplane	bag	cap	car	chair	earphone	guitar	knife	lamp	laptop	motorbike	mug	pistol	rocket	stakeboard	table
Local Feature																		
PointNet [21]	80.4	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
SO-Net [10]	-	84.6	81.9	83.5	84.8	78.1	90.8	72.2	90.1	83.6	82.3	95.2	69.3	94.2	80.0	51.6	72.1	82.6
PointNet++ [23]	81.9	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
DGCNN [41]	82.3	85.2	84.0	83.4	86.7	77.8	90.6	74.7	91.2	87.5	82.8	95.7	66.3	94.9	81.1	63.5	74.5	82.6
P2Sequence [13]	-	85.2	82.6	81.8	87.5	77.3	90.8	77.1	91.1	86.9	83.9	95.7	70.8	94.6	79.3	58.1	75.2	82.8
PAConv [48]	**84.2** (83.8)	86.0 (85.8)	(83.9)	(87.4)	(88.5)	(79.0)	(90.4)	(77.1)	(91.9)	(87.8)	(81.6)	(95.9)	(73.0)	(94.7)	(84.1)	(59.9)	(81.8)	(83.8)
Global Feature																		
PCT [5]	-	86.4	85.0	82.4	**89.0**	81.2	91.9	71.5	91.3	88.1	**86.3**	**95.8**	64.6	**95.8**	83.6	62.2	77.6	73.7
PT [55]	83.7	**86.6**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Global-Local Feature																		
RS-CNN [14]	84	86.2	83.5	84.8	88.8	79.6	91.2	81.1	91.6	88.4	86.0	**96.0**	73.7	94.1	83.4	60.5	77.7	83.6
Ours	83.9	85.8	**84.1**	**84.6**	88.2	79.2	90.3	**77.9**	**91.7**	88.1	81.6	95.9	72.6	94.9	**84.4**	64.4	**80.8**	**83.9**

Table 6. Results of Shape Part Segmentation on ShapeNet Parts [44], evaluating mean class and instance IoU, and IoU within each class. We only train one model instead of using multiple models ensemble. (Result in brackets: the re-implementation result by us.)

lution, losing subtle features important for local information processing. Processing local features with point-based representation requires no pooling or grouping process, and could preserve nuanced differences among all points as much as possible, benefiting local learning.

Furthermore, global point-based representation needs to process a continuous infinite coordinate space, whose position embedding is complex for a network to learn during attention mechanism. Our global processing using voxel-based representation aligns all points with mesh grids and has a discrete finite coordinate space which is easy for position embedding.

To verify our claims, we run ablation studies demonstrated in Table 2. In point-based global processing, we use sample and grouping [5] to sample 256 points and do self-attention, and use a single Linear layer to restore point number from 256 to 1024. In voxel-based local processing, we use the same 3D voxel convolution in PVCNN [15]. Results show that our modality choice with point-based local processing and voxel-based global processing has the best performance among all four combinations.

Local Operator While efficient global feature extraction has the only option of using an attention mechanism, local feature extraction has many comparable operators. In Table 3, we substitute our local branch point-based convolution with a different local feature operator and evaluate their performance. It shows that with PAConv [48] consisting the local branch operator, our method has the best performance among all evaluated local operators.

High-Frequency Fusion We dive into the utility of High-Frequency Fusion module. We examine the influence of usage (whether use it or not) and location (before or after feature processing) of the High-Frequency Fusion module. The result are shown in Table 4, and we find that putting High-Frequency Fusion module after feature processing for both local and global branch will achieve the best performance. It shows that our high-frequency fusion module incorporates coordinates and narrows the gap between two modalities after dual processing to benefit learning.

4.3. Efficiency

We claim that our model is light-weighted and computation-efficient. It utilizes point-wise convolution as

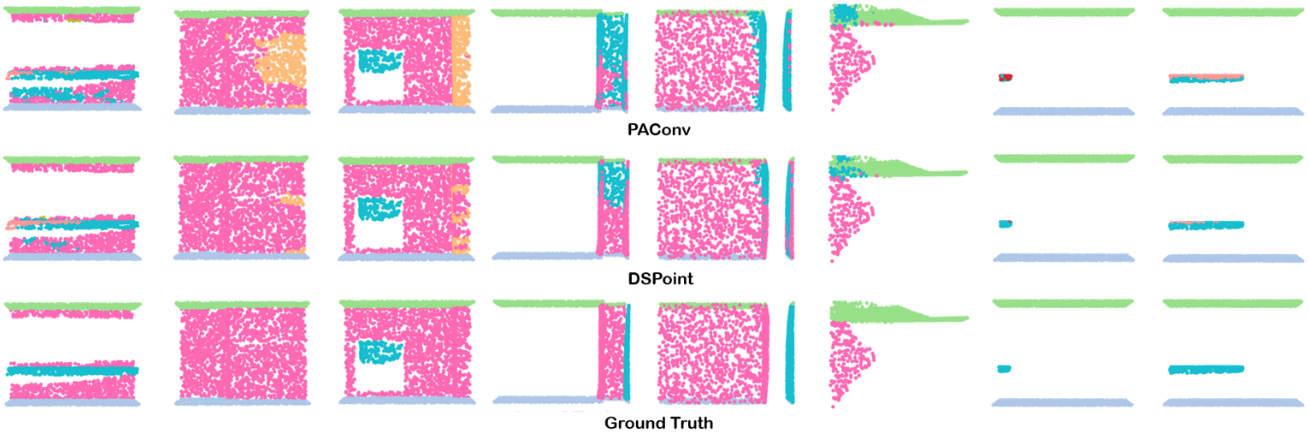

Figure 5. Visualization of Indoor Scene Segmentation on S3DIS [2] Dataset. We project scenes onto a plane and visualize them in low-resolution to benefit comparison. Global attention on the 3D grid incorporates information from non-adjacent parts and helps detect spatially isolated points while maintaining local label consistency within the object parts.

Method	mAcc	mIoU	ceiling	floor	wall	beam	column	window	door	chair	table	bookcase	sofa	board	clutter
Local Feature															
PointNet [21]	49.0	41.1	88.8	97.3	69.8	0.1	3.9	46.2	10.8	58.9	52.6	5.9	40.3	26.4	33.2
PointNet++ [23]	-	50.0	90.8	96.5	74.1	0.0	5.8	43.6	25.4	69.2	76.9	21.5	55.6	49.3	41.9
DGCNN [41]	**84.1**	56.1	-	-	-	-	-	-	-	-	-	-	-	-	-
KPConv [36]	72.8	67.1	92.8	97.3	82.4	0.0	23.9	58.0	**69.0**	**91.0**	81.5	75.3	75.4	66.7	58.9
FPConv [12]	68.9	62.8	**94.6**	98.5	80.9	0.0	19.1	60.1	48.9	88.0	80.6	68.4	53.2	68.2	54.9
PointWeb [54]	66.6	60.3	92.0	98.5	79.4	0.0	21.1	59.7	34.8	88.3	76.3	69.3	46.9	64.9	52.5
PAConv† [48]	(69.6)	66.0 (62.2)	(94.3)	(97.7)	(79.8)	(0.0)	(16.5)	(51.1)	(63.6)	(76.3)	(85.2)	(58.3)	(66.5)	(59.0)	(60.5)
Global Feature															
PCT [5]	67.7	61.3	92.5	98.4	80.6	0.0	19.4	61.6	48.0	76.6	85.2	46.2	67.7	67.9	52.3
PT [55]	76.5	**70.4**	94.0	98.5	**86.3**	0.0	**38.0**	**63.4**	74.3	82.4	89.1	**80.2**	74.3	76.0	59.3
Global-Local Feature															
Ours	70.9	63.3	94.2	98.1	82.4	0.0	19.1	49.9	66.2	78.2	85.6	59.0	67.9	62.3	**59.9**

Table 7. Results of Indoor Scene Segmentation on S3DIS [2] tested on Area 5. Evaluate mean accuracy, mean IoU, and IoU within each class. We only train one model instead of using multiple models ensemble.(Result in brackets: the re-implementation result by us. †:CUDA implementation)

the local feature extractor, which requires less parameters compared with transformer-based methods such as PCT [5]. At the same time, the dual-processing makes our latency more efficient compared with other local-global methods like PointASNL [49]. The comparison of parameter amount and latency are listed in Table 5, where all parameters and latency are measured on a single NVIDIA 2080Ti GPU.

4.4. Down-stream Task

To demonstrate the general applicability and plug-in simplicity of our method, we incorporate it into other baselines then apply to two different downstream tasks: Shape Part Segmentation and Indoor Scene Segmentation.

Shape Part Segmentation. We evaluate our model on ShapeNet Parts [44] benchmark. It comprises 16,881 shapes (14,006 for training and 2,874 for testing) with 16 categories labeled in 50 parts. For each shape, we sample 2,048 points. We incorporate our methods to the last three layers of DGCNN [41] with PAConv [48] as local operator. We use channel-wise accumulation instead of channel-wise splitting for plug-in simplicity, where weight between local and global branches is 4:1.

Results are listed in Table 6. Although our mIoU increase compared to PAConv [48] is small, Figure 4 shows clear benefits from our voxel modality, which prevents points from being fragmented into many parts. Our part segmentation is far more spatially continuous in comparison. The mIoU measurement does not reflect the fragmentation problem in PAConv [48]. In many practical applications, having a spatially coherent output, as in our method, is far more important than fragmented results. It proves our strong performance by maintaining plug-in simplicity and practical utility.

Indoor Scene Segmentation We experiment on S3DIS [2] dataset, containing 272 rooms out of six areas. For a fair comparison, we use Area-5 as the test set. Each point is labelled from 13 classes, like doors or walls. For each 1m × 1m block, we sample 4096 points. We integrate our method into all four layers of encoders of PointNet++ [23], with PAConv [48] as local operator, then use channel-wise summation instead of channel-wise dividing for plug-in succinctness, where weight between local and global branches is 4:1. The experiment results are shown in Table 7, and visualized in Figure 5, demonstrating our excellent performance, while benefiting from long-range feature integrating in recognizing isolated parts.

5. Conclusion and Limitation

We propose **D**ual-**S**cale Point Cloud Recognition with High-Frequency Fusion (**DSPoint**) to conduct dual scales and representations 3D learning. Elaborate experiments have demonstrated the effectivity of our DSPoint. Considering limitation, we use a dense sampling of the voxels for ease of implementation based on its grid structure. Some of the voxel grids are empty of the 3D points. The question remains if we should keep those 'blank' voxels or prune them. Keeping them in our naive implementation leads to memory inefficiency and penalty in running time. However, the 'blank' voxels still have position encoding and could be helpful for long-range information propagation. The geometrical relationship between empty vs. occupied voxels could provide beneficial shape information. Our future work will focus on addressing this issue.

References

[1] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7163–7172, 2019. 1

[2] 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. 8, 9

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 2, 3, 4

[4] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. *arXiv preprint arXiv:2106.05304*, 2021. 1

[5] 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, 2021. 5, 6, 7, 8

[6] Xian-Feng Han, Yi-Fei Jin, Hui-Xian Cheng, and Guo-Qiang Xiao. Dual transformer for point cloud analysis. *arXiv preprint arXiv:2104.13044*, 2021. 3

[7] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Structure aware single-stage 3d object detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11873–11882, 2020. 1

[8] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 1, 4

[9] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5010–5019, 2018. 2

[10] 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

[11] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31:820–830, 2018. 1

[12] 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. 5, 6, 8

[13] 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. 7

[14] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019. 1, 7

[15] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. *arXiv preprint arXiv:1907.03739*, 2019. 1, 3, 4, 7

[16] Mona Mahmoudi and Guillermo Sapiro. Three-dimensional point cloud recognition via distributions of geometric distances. *Graphical Models*, 71(1):22–31, 2009. 1

[17] Mingyuan Mao, Renrui Zhang, Honghui Zheng, Peng Gao, Teli Ma, Yan Peng, Errui Ding, Baochang Zhang, and Shumin Han. Dual-stream network for visual recognition. *arXiv preprint arXiv:2105.14734*, 2021. 2, 3

[18] 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. 1, 2

[19] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf:

Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020. 2, 3, 5

[20] Liang Peng, Fei Liu, Zhengxu Yu, Senbo Yan, Dan Deng, Zheng Yang, Haifeng Liu, and Deng Cai. Lidar point cloud guided monocular 3d object detection. *arXiv preprint arXiv:2104.09035*, 2021. 1

[21] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1, 2, 5, 6, 7, 8

[22] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016. 1

[23] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 1, 2, 4, 5, 6, 7, 8, 9

[24] Yongming Rao, Jiwen Lu, and Jie Zhou. Spherical fractal convolutional neural networks for point cloud recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 452–460, 2019. 1

[25] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586, 2017. 2

[26] Riccardo Roveri, Lukas Rahmann, Cengiz Oztireli, and Markus Gross. A network architecture for point cloud classification via automatic depth images generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4176–4184, 2018. 1

[27] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–6. IEEE, 2009. 1

[28] Kripasindhu Sarkar, Basavaraj Hampiholi, Kiran Varanasi, and Didier Stricker. Learning 3d shapes as multi-layered height-maps using 2d convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 71–86, 2018. 1

[29] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 1

[30] 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. 1

[31] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2530–2539, 2018. 2

[32] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015. 2

[33] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. *arXiv preprint arXiv:2109.07161*, 2021. 3

[34] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 3

[35] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 3

[36] 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/CVF International Conference on Computer Vision*, pages 6411–6420, 2019. 3, 5, 6, 8

[37] 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. 3

[38] Dominic Zeng Wang and Ingmar Posner. Voting for voting in online point cloud object detection. In *Robotics: Science and Systems*, volume 1, pages 10–15. Rome, Italy, 2015. 1

[39] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5463–5474, 2021. 3

[40] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3523–3532, 2019. 1

[41] 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. 3, 5, 6, 7, 8

[42] Xinshuo Weng and Kris Kitani. Monocular 3d object detection with pseudo-lidar point cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 1

[43] 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*, pages 9621–9630, 2019. 1, 6

[44] Zhirong Wu, Shuran Song, Aditya Khosla, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets for 2.5 d object recognition and next-best-view prediction. *arXiv preprint arXiv:1406.5670*, 2(4), 2014. 7, 8

[45] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 1, 5, 6, 7

[46] Saining Xie, Sainan Liu, Zeyu Chen, and Zhuowen Tu. Attentional shapecontextnet for point cloud recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4606–4615, 2018. 1

[47] Bin Xu and Zhenzhong Chen. Multi-level fusion based 3d object detection from monocular images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2345–2353, 2018. 1

[48] 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*, pages 3173–3182, 2021. 1, 2, 3, 4, 5, 6, 7, 8, 9

[49] 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. 5, 6, 8

[50] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2020. 1

[51] Kailun Yang, Kaiwei Wang, Ruiqi Cheng, and Xunmin Zhu. A new approach of point cloud processing and scene segmentation for guiding the visually impaired. In *2015 IET International Conference on Biomedical Image and Signal Processing (ICBISP 2015)*, pages 1–6. IET, 2015. 1

[52] Li Yi, Hao Su, Xingwen Guo, and Leonidas J Guibas. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2282–2290, 2017. 1

[53] Zhaoxuan Zhang, Kun Li, Xuefeng Yin, Xinglin Piao, Yuxin Wang, Xin Yang, and Baocai Yin. Point cloud semantic scene segmentation based on coordinate convolution. *Computer Animation and Virtual Worlds*, 31(4-5):e1948, 2020. 1

[54] 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*, pages 5565–5573, 2019. 8

[55] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 5, 6, 7, 8

[56] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018. 1, 2