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

Renrui Zhang\textsuperscript{*1}, Ziyao Zeng\textsuperscript{*2,3}, Ziyu Guo\textsuperscript{*4}, Xinben Gao\textsuperscript{2}, Kexue Fu\textsuperscript{1}, Jianbo Shi\textsuperscript{1,2,6}
\textsuperscript{1}Shanghai AI Laboratory \textsuperscript{2}Uisee Research \textsuperscript{3}ShanghaiTech University \textsuperscript{4}Peking University \textsuperscript{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 Dual-Scale 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 widely-adopted ModelNet40, ShapeNet, and S3DIS demonstrate wise convolution for fine-grained geometry parsing, the other by voxel-wise global attention for long-range structural exploration.

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 widely-adopted ModelNet40, ShapeNet, and S3DIS demonstrate wise convolution for fine-grained geometry parsing, the other by voxel-wise global attention for long-range structural exploration.

CCS CONCEPTS

• Computing methodologies → Point-based models.

KEYWORDS

point cloud recognition, neural networks, dual-scale processing

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.

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), 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 Dual-Scale 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].
At the same time, the voxelization process naturally aggregates voxel-wise global feature and point-wise local feature. We observe for 3D raw points.

Design of using a 3D grid for global attention and local convolution coordinates communication, local-global shape features from dual and integrate it with point-wise features. Aided by this inter-path encode each voxel’s coordinate into high-frequency representation ing the high-frequency point encoding concept in NeRF [19], we point-wise shape features would bring about ambiguities. Borrow-
are homogeneous, and directly fusing them with heterogeneous because shape features of different points from one voxel usually insufficient for removing across modality mis-alignment. This is 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. Borrow-

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.

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 impractical 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 PConv [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) (Figure 3, 4). 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 attention mechanism [3]. We formulate the parallel propagation as

\[ f_L = \text{Convolution}(f_l); \quad f_G = \text{Attention}(f_g), \]

where \( f_L \) and \( f_G \) are the specific-encoded features of \( f_l \) and \( f_g \). Then, \( f_G \) are upscaled 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 matrices, \( A_{L\rightarrow G} \in \mathbb{R}^{n \times m} \) and \( A_{G\rightarrow L} \in \mathbb{R}^{m \times n} \), denoted as

\[ h_L = A_{L\rightarrow G} f_G; \quad h_G = A_{G\rightarrow L} f_L, \]

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 \( a_l C \) channels and \( f_g \) with \( a_g C \) channels, where \( a_l \) and \( a_g \) weighing the importance between two representations and \( a_l + a_g = 1 \). To reserve local details, we maintain the points’ spatial density for \( f_l \). As for downsampling \( f_g \), we convert the irregular

Figure 3: Input coordinates will pass through three Dual-scale blocks with residual connection, then feed through a classification head to obtain shape classification prediction. 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.
points into grid-form low-resolution voxels \cite{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) \cite{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, 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. 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\cite{48}. In voxel based attention branch, we voxelized points according to PVCNN\cite{15}, encode voxel coordinates using a linear layer and add to voxel features, employ a layer normalization, apply self-attention between all voxels, then devoxelized 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 operation from \cite{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 \(\hat{f}_l(x, y, z)\) as

\[
\hat{f}_l(x, y, z) = \sum_{i=1}^{k} W(x_i, y_i, z_i) \odot f_l(x_i, y_i, z_i),
\]

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 \cite{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

\[
f_G(x, y, z) = A(x, y, z) \hat{f}_G(x, y, z),
\]

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 distortions, \(f_G\) 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

Shown as Figure 4, 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 \cite{19}, which maps a low-frequency point coordinates into higher-dimensional vectors via a set of trigonometric functions:

\[
\gamma(c) = (c, \sin(2^0 \pi c), \cos(2^0 \pi c), \ldots, \\
\sin(2^{L-1} \pi c), \cos(2^{L-1} \pi c))
\]

Here \(\gamma\) 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 \(\text{Coords}_v\), and all those of points from the point-wise branch as \(\text{Coords}_p\). Respectively, we encode them via high-frequency function \(\gamma(\cdot)\) as

\[
\gamma^G_v = \gamma(\text{Coords}_p); \quad \gamma^L_v = \gamma(\text{Coords}_p),
\]

where \(\gamma^G_v\) and \(\gamma^L_v\) represent the high-frequency encoded voxels (global) and points (local) coordinates, whose channels are the same as \(f_l\) and \(f_G\). Then, we aggregate the voxel-related \(\gamma^G_v\) with point-wise features \(f_l\), and the point-related \(\gamma^L_v\) 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

\[
\begin{align*}
\gamma^G_L &= \gamma^G_v + \text{Linear}(hf_G^L); \\
\gamma^L_L &= \gamma^L_v + \text{Linear}(hf_G^L),
\end{align*}
\]

where \(\gamma^G_L\) and \(\gamma^L_L\) 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 \(hf_G\) brings about homogeneous alignment for local feature \(f_G\) and, while \(hf_G\) carries heterogeneous discrimination for \(f_l\). Finally, we concatenate \(\gamma^G_L\) and \(\gamma^L_L\) through the channel dimension and apply a linear layer for the final fusion, which restore point features into
Figure 4: High-frequency fusion module. We encode coordinates of one modality with a high-frequency embedding function and inject into features of another branch to help feature alignment.

### Figure 3(a)

- **HF - Fusion Module**
- **Input Fields:**
  - Point Coordinate
  - Voxel Coordinate
  - Point Features
  - Voxel Features

- **Modules:**
  - HF - Encode
  - Linear
  - Devoxelization
  - Concatenate

- **Output:**
  - Fusion Features

---

**Table 1: Results of Object Classification on ModelNet40 [45].**

| Method      | Input | Accuracy |
|-------------|-------|----------|
| 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) |
| PCT[5]       | xyz   | 93.2 (92.8) |
| PT[55]       | xyz+nor | 93.7     |

**Global-Local Feature**

| Method      | Input  | Accuracy |
|-------------|--------|----------|
| PointASNL [49] | xyz+nor | 93.2     |
| Ours        | xyz    | 93.5     |

**Table 2: Different modalities for dual-scale processing.**

|                  | Point + Global | Voxel + Global |
|------------------|----------------|----------------|
| Point + Local    | 93.2           | 93.5           |
| Voxel + Local    | 93.0           | 93.1           |

**Table 3: Ablation of local operator.**

| 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     |

---

**4 EXPERIMENTS**

**4.1 Shape Classification**

We evaluate on ModelNet40 [45] dataset for object classification. This dataset contains 40 categories of 12,311 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. Results in brackets are our re-implementation results.

---

**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 classification 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.
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]  | 1.16M  | 123.6   |

Table 5: Efficiency evaluation measured on ModelNet40

| Method          | Param. | Latency |
|-----------------|--------|---------|
| DGCNN[41]       | 1.81M  | 85.8    |
| PointNet++[23]  | 1.74M  | 35.3    |
| KPConv[36]      | -      | 120.5   |
| FPConv[12]      | -      | -       |
| PAConv (PN)[48] | -      | -       |
| PCT[5]          | 2.88M  | 92.4    |
| PT 2021[55]     | -      | 530.2   |
| PointASNL[49]   | 1.16M  | 123.6   |

activation function. To be environmental-friendly, we do not train massive amounts of models then use their ensemble 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.

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 resolution, 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 computationally-efficient. It utilizes point-wise convolution as 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. It could be noticed that our method achieved a great trade-off by improving its efficiency by a lot margin mentioned in Table 5, with a slight cost of accuracy. Such trade-off would be more worthy in the industrial field like self-driving which requires high inference speed and light model parameter amount.
Figure 5: 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  | chair | earphone | guitar | knife | lamp | laptop | motorbike | mug  | pistol | rocket | stakeboard | table |
|--------------|-----------|-----------|----------|------|------|-------|----------|--------|-------|------|--------|-----------|------|--------|--------|-------------|-------|
| 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 | 91.2   | 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     | 71.1   | 91.1  | 86.9 | 83.9   | 95.7      | 70.8 | 94.6   | 79.3   | 58.1        | 75.2  | 82.8   |
| PAConv [48]  | 84.2      | 86.0      | 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)  |

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.)

Figure 6: 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.

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
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)

| 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    |
| FConv[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** | **59.5** |
| **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    |

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 5 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 6, demonstrating our excellent performance, while benefiting from long-range feature integrating in recognizing isolated parts.

### 5 ERROR MODE ANALYSIS

Aleatoric uncertainty measures the uncertainty that how likely one sample would be misclassified as another class. Those data near the decision boundary would have high aleatoric uncertainty. As shown in Figure 7, selected misclassified samples pair are similar to the other class hence are misclassified into each other’s class. This misclassification is caused by data distribution itself which couldn’t even be told by human. Thus we could not improve our performance on those data by improving our model design. By our rough estimation, it limits the upper bound of classification accuracy of this dataset to around 94%-95%. Under such circumstances, it would be more worthy to improve the model’s efficiency by a

![Figure 7: Aleatoric uncertainty exhibited in test dataset.](image1)

![Figure 8: Misclassification which might be solved by processing 2D projection and 3D point cloud simultaneously.](image2)
Besides, it’s worth noticing that even though some test samples have significant features, they are still misclassified. As shown in Figure 8, the first sample is a cup that has a handle, like some other cups in the dataset, while all vases in the dataset have no handles. Even so, this cup is misclassified as a vase. One possible explanation could be that its small volume of handle leads to the insignificance of the feature response, while its body resembles a vase. Similarly, the bench, which has a back is mistaken as a stand, which has no back. It might be due to the bench having a curved back which is similar to the table-board of a nightstand. In the future, to better handle such a circumstance, we could project the point cloud into the 2D plane, which limits the z-axis variance and amplify the significance of the handle as well as the bench back. By processing 2D projection and 3D point cloud simultaneously, we might achieve better performance in this task.

6 CONCLUSION

We propose Dual-Scale Point Cloud Recognition with High-Frequency Fusion (DSPoint) to conduct dual scales and representations 3D learning. Elaborate experiments have demonstrated the effectiveness of our DSPoint, guiding a possible future direction.

REFERENCES

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

[2] Iro Armeni, Ozan Sener, Amir R Zamar, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 2016. 3D semantic parsing of large-scale indoor spaces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 1534–1543.

[3] Dimitry Bahdanau, KyoungHyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014).

[4] Ankur Goel, Hai Li, Bowei Liu, Alejandro Newell, and Jia Deng. 2021. Revisiting Point Cloud Shape Classification with a Simple and Effective Baseline. arXiv preprint arXiv:2106.05304 (2021).

[5] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. 2021. PCT: Point cloud transformer. Computer Vision and Image Understanding 2, 2 (2021). 187–199.

[6] Xingchen Peng, Hai Yu, Jian Lin, Hui-Xian Cheng, and Guo-Qiang Xiao. 2021. Dual Transformer for Point Cloud Analysis. arXiv preprint arXiv:2104.13044 (2021).

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

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

[9] Asako Kanazaki, Yasuuki Matsushita, and Yoshifumi Nishida. 2018. Rotationnet: Joint object categorization and pose estimation using multi-views from unsupervised viewpoints. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 5010–5019.

[10] Jiaxin Li, Ben M Chen, and Gim Hee Lee. 2018. So-net: Self-organizing network effectivity of our DSPoint, guiding a possible future direction.

[11] Yongcheng Liu, Bin Fan, Shaming Xiang, and Chunhong Fan. 2019. Relation-shape convolutional neural network for point cloud analysis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 8985–8994.

[12] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. 2019. Point-voxel cnn for efficient 3d deep learning. arXiv preprint arXiv:1907.03739 (2019).

[13] Mona Mahmoudi and Guillermo Sapiro. 2009. Three-dimensional point cloud recognition via distributions of geometric distances. Graphical Models 71, 1 (2009). 23–31.

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

[15] Daniel Maturana and Sebastian Scherer. 2015. VoxelNet: A 3d convolutional neural network for real-time object recognition. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 922–928.

[16] Ben Mildenhall, Pratul P Sinhaan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2020. Nerf: Representing scenes as neural radiance fields for view synthesis. In European conference on computer vision. Springer, 405–421.

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

[18] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. PointNet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition. 652–660.

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

[20] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. PointNet++. Deep hierarchical feature learning on point sets in a metric space. arXiv preprint arXiv:1706.02413 (2017).

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

[22] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. 2017. Octnet: Learning hierarchical feature maps on 3d point clouds. In Advances in Neural Information Processing Systems. 7163–7172.

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

[24] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, and Victor Lempitsky. 2021. Resolution-robust Large Mask Inpainting with Fourier Convolutions. arXiv preprint arXiv:2109.07161 (2021).

[25] Christian Szegedy, Wei Liu, Yangqing Jia, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition. 1–9.

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

[27] Satoru Sato, Zico Kolter, and Jason Yosinski. 2018. 3D shapes as multi-layered height-maps using 2d convolutional networks. In Proceedings of the European Conference on Computer Vision (ECCV). 71–86.

[28] Xuanjia Shi, Zhe Wang, Jingping Shi, Xiaogang Wang, and Hongsheng Li. 2020. Feature points to parts: 3d object detection from point cloud by part-aware and part-aggregation network. IEEE transactions on pattern analysis and machine intelligence (2020).

[29] Shaoshuai Shi, Zhe Wang, Jingping Shi, Xiaogang Wang, and Hongsheng Li. 2020. Feature points to parts: 3d object detection from point cloud by part-aware and part-aggregation network. IEEE transactions on pattern analysis and machine intelligence (2020).

[30] Hang Su, Varun Jampami, Desing Sun, Subhranu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. 2018. SplatNet: Sparse lattice networks for point cloud processing. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2530–2539.

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

[32] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Aude Billard. 2021. Resolution-robust Large Mask Inpainting with Fourier Convolutions. arXiv preprint arXiv:2109.07161 (2021).

[33] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning. PMLR, 6055–6114.

[34] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. 2019. Pconv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 6411–6420.
[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in neural information processing systems. 5998–6008.

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

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

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

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

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

[43] Bin Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. 2021. PACConv: Position Adaptive Convolution with Dynamic Kernel Assembling on Point Clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 3173–3182.

[44] Yu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. 2020. 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. 5589–5598.

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

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

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