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

I. Ablation Study

A. Ablation Study

To quantify the effectiveness of DSPoint, we conduct ablation studies on ModelNet40[Wu et al.(2015)], following the same experiment setting of Shape Classification mentioned above.

a) 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 I. In point-based global processing, we use sample and grouping[Guo et al.(2021)] 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[Liu et al.(2019c)]. Results show that our modality choice with point-based local processing and voxel-based global processing has the best performance among all four combinations.

b) 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 II, we substitute our local branch point-based convolution with a different local feature operator and evaluate their performance. It shows that with PAConv[Xu et al.(2021)] consisting the local branch operator, our method has the best performance among all evaluated local operators.

c) 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 III, 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.

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

TABLE I

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

Local Operator	Accuracy
DSPoint w. MLP	91.2
DSPoint w. MLP + SG	92.4
DSPoint w. KPConv[Thomas et al.(2019)]	93.2
DSPoint w. PointConv[Wu et al.(2019)]	92.8
DSPoint w. PAConv[Xu et al.(2021)]	**93.5**

TABLE II

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[Qi et al.(2017a)]. SG stands for sample and grouping from PCT[Guo et al.(2021)].

	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 III

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.)

II. Down-stream Task Experiments

A. 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 main paper, 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.

a) Shape Part Segmentation.: We evaluate our model on ShapeNet Parts[Wu et al.(2014)] 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[Wang et al.(2019)] with PAConv[Xu et al.(2021)] 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 IV. Although our mIoU increase compared to PAConv[Xu et al.(2021)] is small, Figure 3 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[Xu et al.(2021)]. 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.

b) Indoor Scene Segmentation: We experiment on S3DIS[Armeni et al.(2016)] 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++[Qi et al.(2017b)], with PAConv[Xu et al.(2021)] 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 V, and visualized in Figure 4, demonstrating our excellent performance, while benefiting from long-range feature integrating in recognizing isolated parts.

III. 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 1, 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 lot margin instead of improving its accuracy slightly, which aligns with our model's superiority.

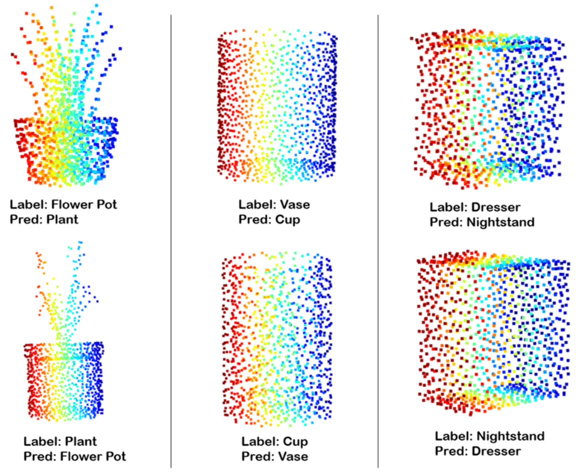

Fig. 1. Aleatoric uncertainty exhibited in test dataset.

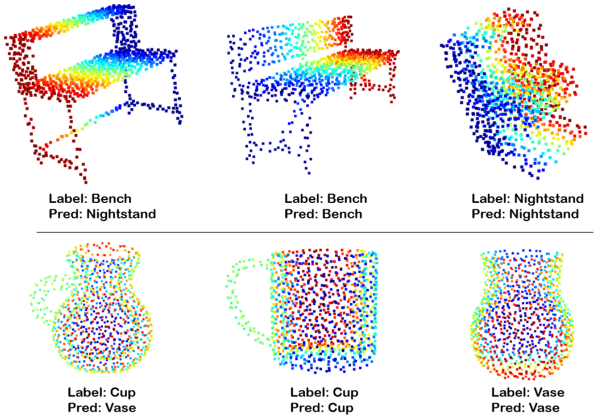

Fig. 2. Misclassification which might be solved by processing 2D projection and 3D point cloud simultaneously.

Besides, it's worth noticing that even though some test samples have significant features, they are still misclassified. As shown in Figure 2, 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 explainatino 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 nightstand, which has no back. It might be due to the bench having a carved back which is similar to the table-board of the 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.

References

[Armeni et al.(2016)] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 2016. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the*

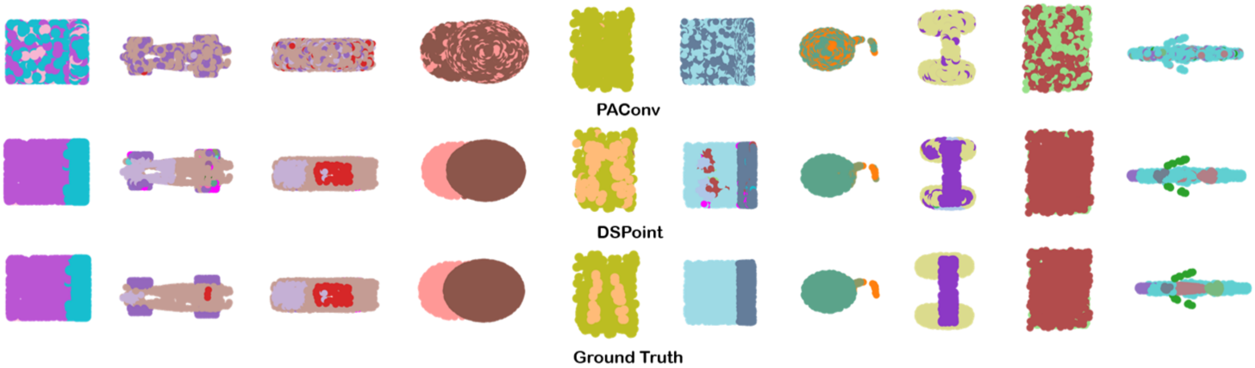

Fig. 3. Visualization Results of ShapeNet [Wu et al.(2015)]. It demonstrates our compared baseline PAConv[Xu et al.(2021)] (first row), our method DSPoint (second row), and ground truth, which indicates our excellent performance of spatial continuity on part segmentation.

Method	Cls. mIoU	Ins. mIoU	airplane	bag	cap	car	chair	earphone	guitar	knife	lamp	laptop	motorbike	mug	pistol	rocket	stakeboard	table
Local Feature																		
PointNet[Qi et al.(2017a)]	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[Li et al.(2018)]	-	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++[Qi et al.(2017b)]	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[Wang et al.(2019)]	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[Liu et al.(2019b)]	-	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[Xu et al.(2021)]	**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[Guo et al.(2021)]	-	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[Zhao et al.(2021)]	83.7	**86.6**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Global-Local Feature																		
RS-CNN[Liu et al.(2019a)]	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 IV

Results of Shape Part Segmentation on ShapeNet Parts[Wu et al.(2014)], 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.)

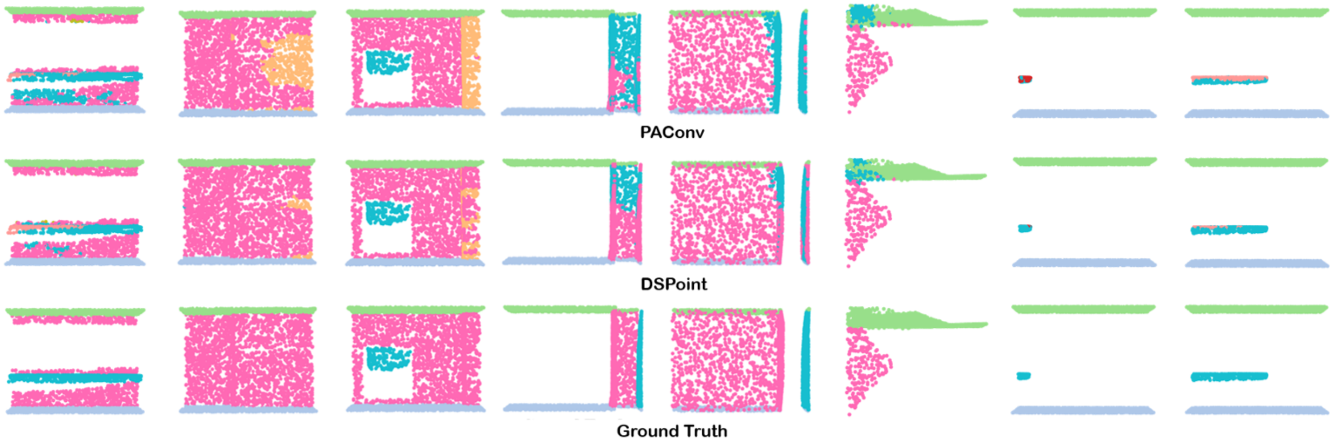

Fig. 4. Visualization of Indoor Scene Segmentation on S3DIS[Armeni et al.(2016)] Dataset. We project scenes onto a plane and visualize them in low-resolution to benefit comparison. Global attention on the 3D grid incorporates information from non-adjacent parts and helps detect spatially isolated points while maintaining local label consistency within the object parts.

IEEE Conference on Computer Vision and Pattern Recognition. 1534–1543.

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

[Li et al.(2018)] Jiaxin Li, Ben M Chen, and Gim Hee Lee. 2018. So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 9397–9406.

[Lin et al.(2020)] Yiqun Lin, Zizheng Yan, Haibin Huang, Dong Du, Ligang Liu, Shuguang Cui, and Xiaoguang Han. 2020. Fpconv: Learning local flattening for point convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4293–4302.

[Liu et al.(2019b)] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. 2019b. 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*, Vol. 33. 8778–8785.

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

Method	mAcc	mIoU	ceiling	floor	wall	beam	column	window	door	chair	table	bookcase	sofa	board	clutter
Local Feature															
PointNet[Qi et al.(2017a)]	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++[Qi et al.(2017b)]	-	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[Wang et al.(2019)]	**84.1**	56.1	-	-	-	-	-	-	-	-	-	-	-	-	-
KPConv[Thomas et al.(2019)]	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[Lin et al.(2020)]	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[Zhao et al.(2019)]	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†[Xu et al.(2021)]	(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[Guo et al.(2021)]	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[Zhao et al.(2021)]	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 V

Results of Indoor Scene Segmentation on S3DIS[Armeni et al.(2016)] 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)

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

[Qi et al.(2017a)] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017a. 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.

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

[Thomas et al.(2019)] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. 2019. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6411–6420.

[Wang et al.(2019)] 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.

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

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

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

[Xu et al.(2021)] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. 2021. PAConv: 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.

[Zhao et al.(2019)] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. 2019. Pointweb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5565–5573.

[Zhao et al.(2021)] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. 2021. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 16259–16268.