Supplementary Material for
Convolution in the Cloud: Learning Deformable Kernels in 3D Graph Convolution Networks for Point Cloud Analysis

Zhi-Hao Lin[1]   Sheng-Yu Huang[1]   Yu-Chiang Frank Wang[1,2]
[1]Graduate Institute of Communication Engineering, National Taiwan University, Taiwan
[2]ASUS Intelligent Cloud Services, Taiwan
{r08942062, r08942095, ycwang}@ntu.edu.tw

Support Number	#params	Acc.(%)
1	0.89M	93.3
3	1.51M	93.9
5	2.13M	93.6

Table 1: **Performances of shape classification on ModelNet10 with varying support number S**

A. Ablation study of support number

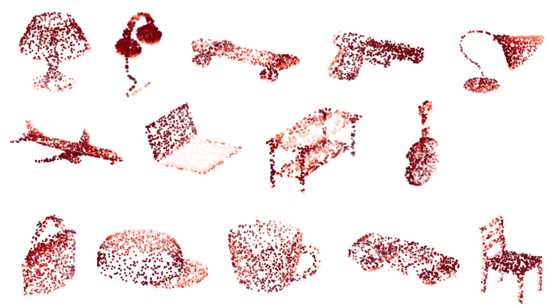
We conduct experiments on 3D-GCN by varying the support number S of the proposed learnable kernels. The results are shown in Table 1. From this table, we see that while kernels with more support numbers (e.g., $S = 3$ and 5) generally showed slightly better classification accuracy, the performance enhancement was marginal. More importantly, the model complexity would grow significantly with more support numbers, leading to large memory and computation loads. As a result, $S = 1$ in our work is a reasonable choice.

B. More visualization of kernel response

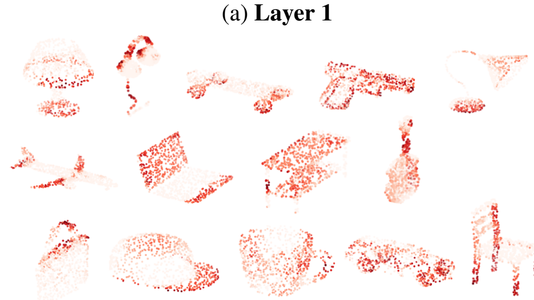
In Figure 1, we visualize more results of the points which have large response values at each layer of our 3D-GCN. From low to high-level layers, we can see that responses were shifted from point to part levels.

C. More Visualization of Part Segmentation

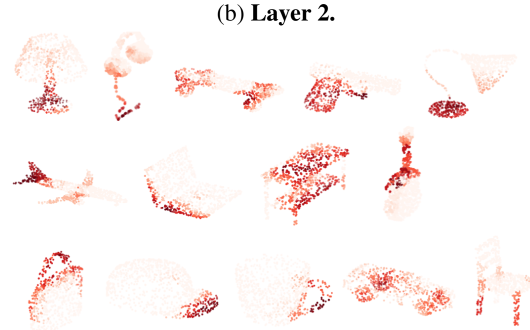
In this section, we show one pair of part segmentation result for each category in ShapeNetPart dataset [3], in comparison with KPConv [2] and PointNet++ [1]. To have a fair comparison, we retrain all three models with training data sampled with 1024 points and normalized to a unit sphere, without any other data augmentation. As shown in Figure 2, both KPConv and PointNet++ failed to properly segment the corresponding parts in both cases even when the object contains only two parts (*e.g.*, laptop and mug). In contrast, our 3D-GCN remains stable and shows impressive performance.



(a) **Layer 1**



(b) **Layer 2.**



(c) **Layer 4**

Figure 1: **Kernel responses in different layers.** Note that points with larger responses are colored in darker red.

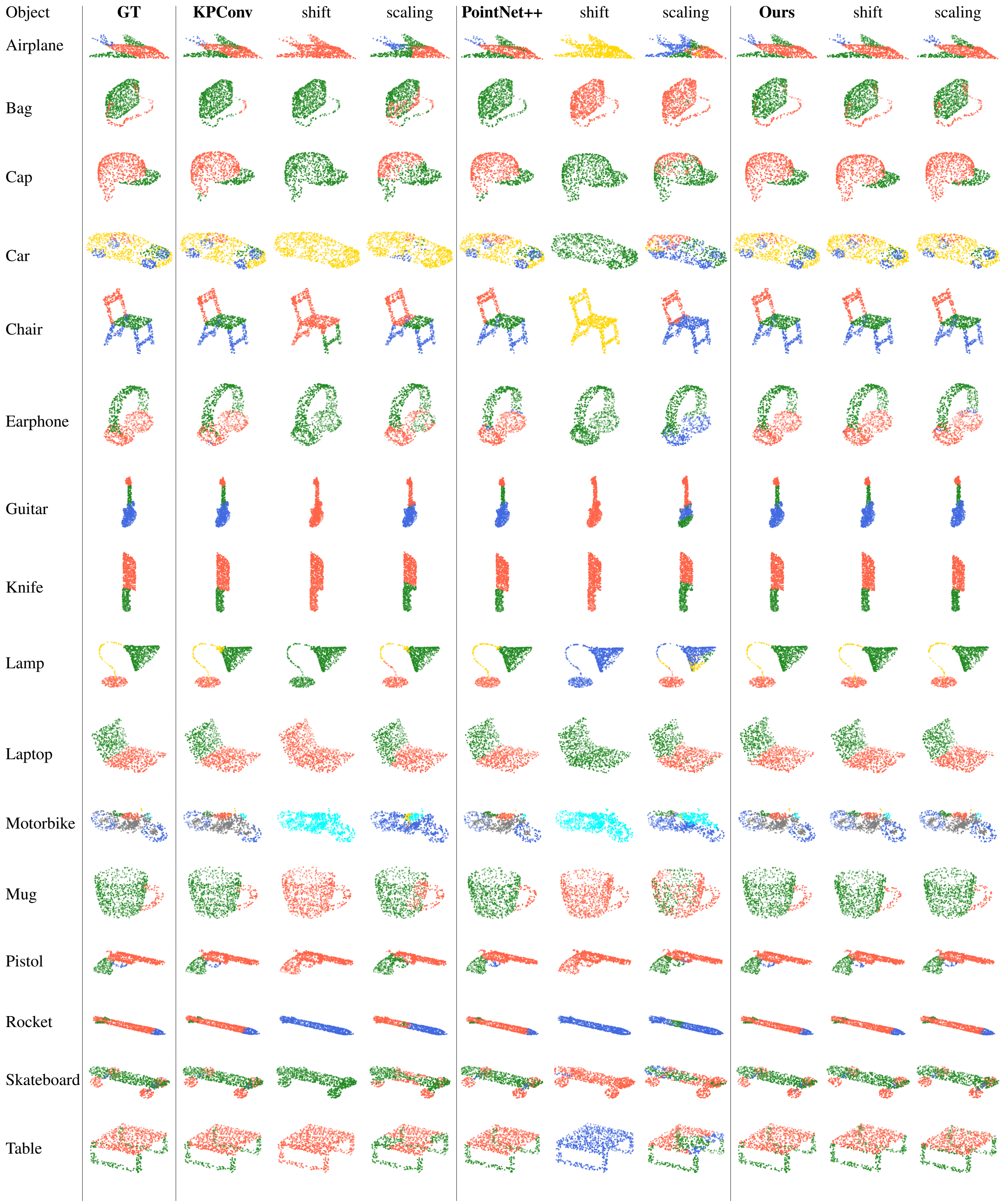


Figure 2: **Visualization of part segmentation on ShapeNetPart.** We show more example segmentation results along with those produced by KPConv [2] and PointNet++ [1]. Note that we shift the center/coordinates of each object by 100, and enlarge object size by 10 times for testing as we did in the main paper.

References

[1] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 1, 2

[2] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, Francois Goulette, and Leonidas J. Guibas. KPConv: Flexible and Deformable Convolution for Point Clouds. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 2

[3] Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A Scalable Active Framework for Region Annotation in 3D Shape Collections. *SIGGRAPH Asia*, 2016. 1