Supplementary Material:
A-CNN: Annularly Convolutional Neural Networks on Point Clouds

1. Ball Query *vs* Ring-based Scheme

The comparison of multi-scale method proposed in [12] and our ring-based scheme is depicted in Fig. 1. It is noted that comparing to multi-scale regions, the ring-based structure does not have overlaps (no neighboring point duplication) at the query point's neighborhood. It means that each ring contains its own unique points.

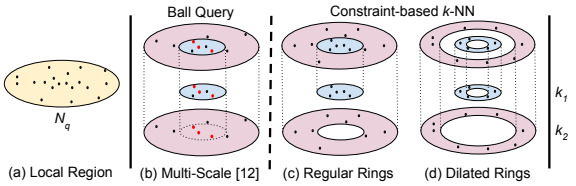


Figure 1: A schematic comparison for searching neighbors in a local region with N_q points between multi-scale approach from [12] and our proposed approaches with regular and dilated rings. The number of neighboring points per region (e.g., k_1 and k_2) is the same between different methods. Regions in multi-scale architecture have neighboring overlaps (red points belong to different regions near the same query point $\mathbf{q}$), while regular and dilated rings have the unique neighbors.

Table 1: Experiments on redundancy on *ModelNet40* dataset. AAC is accuracy average class, OA is overall accuracy.

	AAC	OA
PointNet++ (multi-scale / with overlap)	86.5	90.2
PointNet++ (multi-ring / without overlap)	87.3	90.6
A-CNN (with all components)	**90.3**	**92.6**

We have discovered that reducing redundancy can improve the existing multi-scale approach in [12]. We test redundancy issue on original PointNet++ model [12] with and without overlap / redundancy. We compare the original PointNet++ multi-scale model with ball queries (with redundant points) against PointNet++ with our proposed regular rings (without redundant points). Our experiments show that the proposed multi-ring (i.e., without redundant points) outperforms the multi-scale scheme (i.e., with redundant points) on *ModelNet40* according to Tab. 1.

2. Training Details

We use *A-CNN-3L* network configuration in Tab. 2 for all experiments on point cloud classification tasks and *A-CNN-4L* network configuration in Tab. 2 for both part segmentation and semantic segmentation tasks. We use regular rings in L_1 and dilated rings in L_2 in our *A-CNN-3L* architecture. Similarly, we use regular rings in L_1 and dilated rings in L_2 and L_3 in our *A-CNN-4L* architecture.

We use Adam optimization method with learning rate 0.001 and decay rate 0.7 in classification and decay 0.5 in segmentation tasks. We have trained our classification model for 250 epochs, our part segmentation model for 200 epochs, and our large-scale semantic segmentation models for 50 epochs on each area of *S3DIS* and for 200 epochs on *ScanNet*. The training time of our model is faster than that of PointNet++ model, since we use ring-based neighboring search, which is more efficient and effective than ball query in PointNet++ model. For instance, the training time on the segmentation model for 200 epochs is about 19 hours on a single NVIDIA Titan Xp GPU with 12 GB GDDR5X, and PointNet++ model needs about 32 hours for the same task. The size of our trained model is 22.3 MB and the size of PointNet++ model is 22.1 MB.

3. Feature Visualization

Local Feature Visualization. Fig. 2 and Fig. 3 visualize the magnitude of the gradient per point in the classification task on *ModelNet10* and *ModelNet40* datasets. Blue color represents low magnitude of the gradients and red color represents high magnitude of the gradients. The points with higher magnitudes get greater updates during training and the learning contribution of them is higher. Therefore, this feature visualization could be thought as the object saliency. For example, in *ModelNet40* dataset our model considers wings and tails as important regions to classify an object as an airplane; bottle neck is important for a bottle; the flowers and leaves are important for a plant; tube or middle part (usually narrow parts) is important for a lamp; legs are important to classify an object as a stool.

Global Feature Visualization. Fig. 4 and Fig. 5 shows the t-SNE clustering visualization [10] of the learned global shape features from the proposed A-CNN model for the shape classification tasks in *ModelNet10* and *ModelNet40* test splits. We reduce 1024-dim feature vectors to 2-dim features. We can see that similar shapes are well clustered together according to their semantic categories. For example, in *ModelNet10* dataset the clusters of desk, dresser, night stand, and table classes are closer and even intersect with each other, because the objects from these classes look similar. The perplexity parameters for *ModelNet10* and *ModelNet40* datasets are set as 15 and 50, respectively, to

Table 2: Network configurations.

		L_1	L_2	L_3	L_4
A-CNN-3L (classification)	C	512	128	1	-
	rings	[[0.0, 0.1], [0.1, 0.2]]	[[0.1, 0.2], [0.3, 0.4]]	-	-
	k	[16, 48]	[16, 48]	128	-
	F	[[32,32,64], [64,64,128]]	[[64,64,128], [128,128,256]]	[256,512,1024]	-
A-CNN-4L (segmentation)	C	512	128	32	1
	rings	[[0.0, 0.1], [0.1, 0.2]]	[[0.1, 0.2], [0.3, 0.4]]	[[0.2, 0.4], [0.6, 0.8]]	-
	k	[16, 48]	[16, 48]	[16, 48]	32
	F	[[32,32,64], [64,64,128]]	[[64,64,128], [128,128,256]]	[[128,128,256], [256,256,512]]	[512,768,1024]

Note: Both of the models represent encoder part. *A-CNN-3L* model consists of three layers. A-CNN-4L model consists of four layers. For each layer, C is the number of centroids, $rings$ is the inner and outer radiuses of a ring: $[R_{inner}, R_{outer}]$, k is number of neighbors, F is feature map size. For example, our *A-CNN-4L* model at the first layer L_1 has 512 centroids; two regular rings where first ring constrained by radiuses of 0.0 and 0.1 and the second ring has radiuses of 0.1 and 0.2; k-NN search returns 16 points in the first ring, and 48 points in the second ring; the feature map size in the first ring is equal to $[32, 32, 64]$ and in the second ring is $[64, 64, 128]$. Convolutional kernel size across different rings and layers is the same and equal to 1×3. Also, we have to double the number of centroids in each layer in model *A-CNN-4L* on *ScanNet* as the number of points in each block is twice more than that in *S3DIS*.

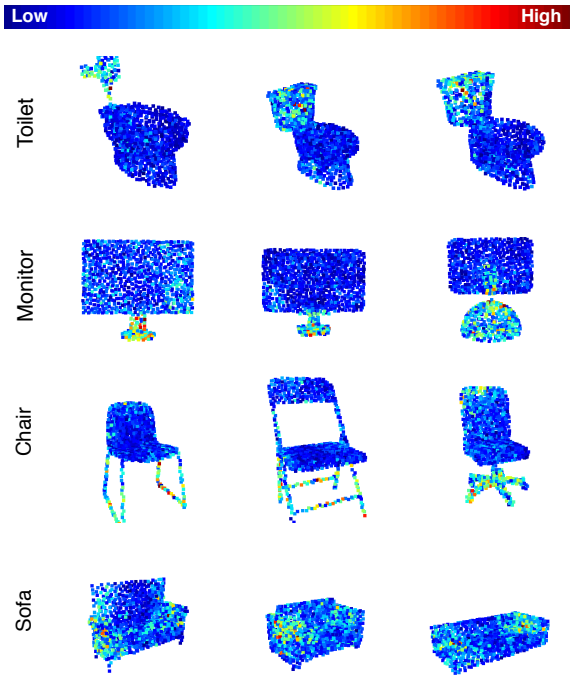


Figure 2: The magnitude of the gradient per point in the classification task on *ModelNet10* dataset.

reduce spare space between clusters.

4. Data Preparation Details

***S3DIS* data preparation.** To prepare training and testing datasets, we divide every room into blocks with a size of 1 m $\times$ 1 m $\times$ 2 m and with a stride of 0.5 m. We has sampled 4096 points from each block. The height of each block is scaled to 2 m to ensure that our constraint-based k-NN search works optimally with the provided radiuses. In total, the prepared dataset contains 23,585 blocks across all six areas. Each point is represented as a 6D vector (XYZ: normalized global point coordinates and centered at origin, RGB: colors). We do not use the relative position of the block in the room scaled between 0 and 1 as used in [11], because our model already achieves better results without

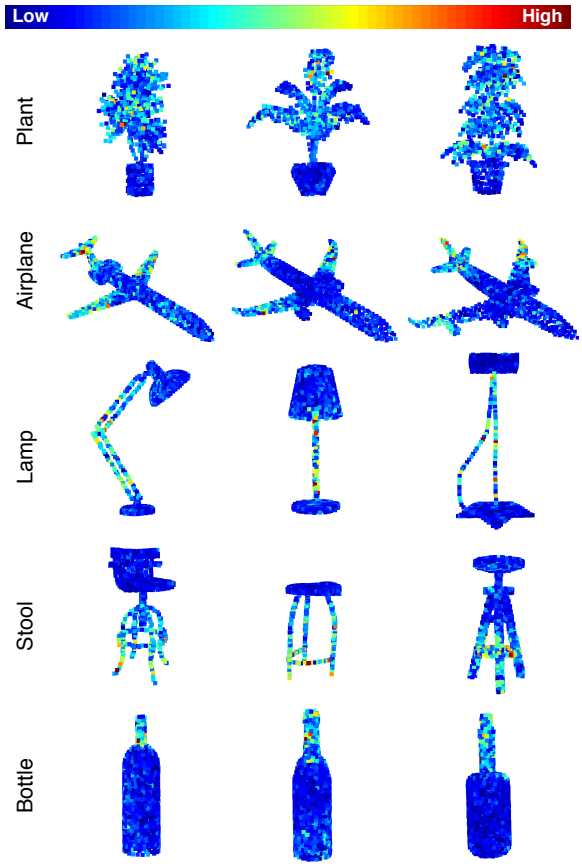


Figure 3: The magnitude of the gradient per point in the classification task on *ModelNet40* dataset.

using this additional information. We calculate point normals for each room by using the Point Cloud Library (PCL) library [13]. The calculated normals are only used to order points in the local region. For data augmentation, we use the same data augmentation strategy as used in the point cloud segmentation on *ShapeNet-part* dataset which is point perturbation with point shuffling.

***ScanNet* data preparation**. ScanNet divides original 1513 scanned scenes in 1201 and 312 for training and testing, respectively. We sample blocks from the scenes fol-

Table 3: Segmentation results on *ShapeNet-part* dataset (input is *XYZ* only). Per-category and mean IoUs (%) are reported.

	mean	areo	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate board	table
# shapes		2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271
PointNet [11]	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
Kd-Net [4]	82.3	80.1	74.6	74.3	70.3	88.6	73.5	90.2	87.2	81.0	94.9	57.4	86.7	78.1	51.8	69.9	80.3
KCNet [14]	84.7	82.8	81.5	**86.4**	77.6	90.3	76.8	91.0	87.2	84.5	95.5	69.2	94.4	81.6	60.1	75.2	81.3
PCNN [1]	85.1	82.4	80.1	85.5	79.5	90.8	73.2	91.3	86.0	85.0	95.7	73.2	94.8	83.3	51.0	75.0	81.8
PointGrid [6]	**86.4**	**85.7**	82.5	81.8	77.9	**92.1**	**82.4**	**92.7**	85.8	84.2	95.3	65.2	93.4	81.7	56.9	73.5	**84.6**
PointCNN [8]	86.1	84.1	86.5	86.0	**80.8**	90.6	79.7	92.3	**88.4**	**85.3**	**96.1**	**77.2**	**95.2**	**84.2**	**64.2**	**80.0**	83.0
A-CNN (our)	85.9	83.9	**86.7**	83.5	79.5	91.3	77.0	91.5	86.0	85.0	95.5	72.6	94.9	83.8	57.8	76.6	83.0

Table 4: Segmentation results on *ShapeNet-part* dataset (input is *XYZ* + *normals*). Per-category and mean IoUs (%) are reported.

	mean	areo	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate board	table
# shapes		2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271
PointNet++ [12]	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
SyncSpecCNN [19]	84.7	81.6	81.7	81.9	75.2	90.2	74.9	**93.0**	86.1	84.7	95.6	66.7	92.7	81.6	60.6	82.9	82.1
SO-Net [7]	84.9	82.8	77.8	**88.0**	77.3	90.6	73.5	90.7	83.9	82.8	94.8	69.1	94.2	80.9	53.1	72.9	83.0
SGPN [17]	85.8	80.4	78.6	78.8	71.5	88.6	78.0	90.9	83.0	78.8	**95.8**	**77.8**	93.8	**87.4**	60.1	**92.3**	**89.4**
RSNet [3]	84.9	82.7	86.4	84.1	78.2	90.4	69.3	91.4	87.0	83.5	95.4	66.0	92.6	81.8	56.1	75.8	82.2
O-CNN (+ CRF) [16]	85.9	**85.5**	**87.1**	84.7	77.0	91.1	**85.1**	91.9	**87.4**	83.3	95.4	56.9	**96.2**	81.6	53.5	74.1	84.4
Point2Sequence [9]	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
A-CNN (our)	**86.1**	84.2	84.0	**88.0**	**79.6**	**91.3**	75.2	91.6	87.1	**85.5**	95.4	75.3	94.9	82.5	**67.8**	77.5	83.3

Note: "CRF" stands for conditional random field method for final result refinement in O-CNN method.

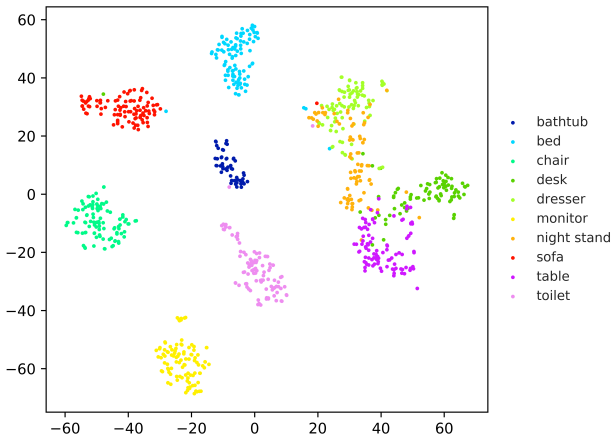


Figure 4: The t-SNE clustering visualization of the learned global shape features from the proposed A-CNN model for the shapes in *ModelNet10* test split.

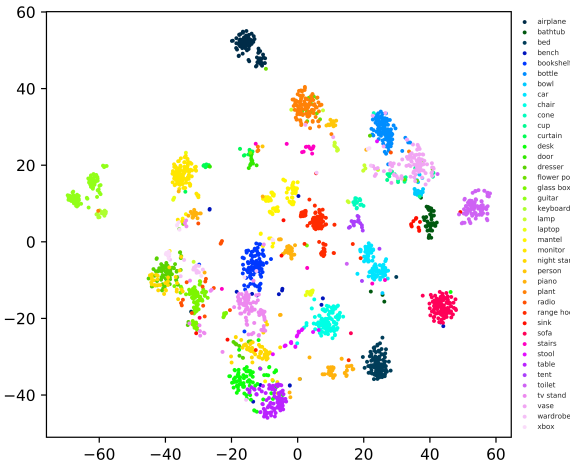


Figure 5: The t-SNE clustering visualization of the learned global shape features from the proposed A-CNN model for the shapes in *ModelNet40* test split.

lowing the same procedure as in [12], where every block has a size of 1.5 m × 1.5 m with 8192 points. We estimate point normals using the PCL library [13]. Each point is represented as a 6D vector (XYZ: coordinates of the block centered at origin, $N_xN_yN_z$: normals) without RGB information. For data augmentation, we use the point perturbation with point shuffling.

5. More Experimental Results

Point Cloud Segmentation. Tab. 3 and Tab. 4 show the quantitative results of part segmentation on *ShapeNet-part* dataset with two different inputs. Tab. 3 reports results when the input is point position only. Tab. 4 reports results when the input is point position with its normals.

For *ShapeNet-part* dataset, we visualize more results (besides the segmentation results shown in the paper) in Fig. 6. We compare our results with PointNet++ [12], and our A-CNN model can produce better segmentation results than PointNet++ model.

Semantic Segmentation in Scenes. For *S3DIS* dataset, we pick rooms from all six areas: area 1 (*row 1*), area 2 (*row 2*), area 3 (*row 3*), area 4 (*row 4*), area 5 (*row 5*), and area 6 (*row 6*); and compare them with PointNet [11] results and ground truth. The results are shown in Fig. 7. The detailed quantitative evaluation results for each shape class are reported in Tab. 5. Our model demonstrates good semantic segmentation results and achieves the state-of-the-art per-

Table 5: Segmentation results on *S3DIS* dataset. "acc" is overall accuracy and "mean" is average IoU over 13 classes.

	acc	mean	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
PointNet [11]	78.5	47.6	88.0	88.7	69.3	42.4	23.1	47.5	51.6	54.1	42.0	9.6	38.2	29.4	35.2
MS+CU (2) [2]	79.2	47.8	88.6	95.8	67.3	36.9	24.9	48.6	52.3	51.9	45.1	10.6	36.8	24.7	37.5
G+RCU [2]	81.1	49.7	90.3	92.1	67.9	44.7	24.2	52.3	51.2	58.1	47.4	6.9	39.0	30.0	41.9
RSNet [3]	-	56.5	92.5	92.8	78.6	32.8	34.4	51.6	68.1	59.7	60.1	16.4	50.2	44.9	52.0
3P-RNN [18]	86.9	56.3	92.9	93.8	73.1	42.5	25.9	47.6	59.2	60.4	66.7	24.8	57.0	36.7	51.6
SPGraph [5]	85.5	62.1	89.9	95.1	76.4	62.8	47.1	55.3	**68.4**	**73.5**	69.2	**63.2**	45.9	8.7	52.9
PointCNN [8]	**88.1**	**65.4**	**94.8**	**97.3**	75.8	**63.3**	**51.7**	**58.4**	57.2	71.6	69.1	39.1	**61.2**	**52.2**	**58.6**
A-CNN (our)	87.3	62.9	92.4	96.4	**79.2**	59.5	34.2	56.3	65.0	66.5	**78.0**	28.5	56.9	48.0	56.8

formance on segmenting *walls* and *chairs*. Meanwhile, our model performs slightly worse than PointCNN [8] on other categories due to their non-overlapping block sampling strategy with paddings which we do not use. Supplementary Video is included for dynamically visualizing each area in detail.

For *ScanNet* dataset, we pick six challenging scenes and visualize the results of our A-CNN model, PointNet++ [12], and ground truth side by side. The visualization results are provided in Fig. 8. Our approach outperforms PointNet++ [12] and other baseline methods, such as PointNet [11], TangentConv [15], and PointCNN [8] according to Tab. 2 in the main paper.

References

[1] M. Atzmon, H. Maron, and Y. Lipman. Point convolutional neural networks by extension operators. *ACM Transactions on Graphics*, 37(4):71:1–71:12, 2018.

[2] F. Engelmann, T. Kontogianni, A. Hermans, and B. Leibe. Exploring spatial context for 3D semantic segmentation of point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–724, 2017.

[3] Q. Huang, W. Wang, and U. Neumann. Recurrent slice networks for 3D segmentation of point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2626–2635, 2018.

[4] R. Klokov and V. Lempitsky. Escape from cells: Deep Kd-networks for the recognition of 3D point cloud models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 863–872, 2017.

[5] L. Landrieu and M. Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4558–4567, 2018.

[6] T. Le and Y. Duan. PointGrid: A deep network for 3D shape understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9204–9214, 2018.

[7] J. Li, B. M. Chen, and G. H. 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.

[8] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. PointCNN: Convolution on X-transformed points. In *Advances in Neural Information Processing Systems*, pages 828–838, 2018.

[9] X. Liu, Z. Han, Y.-S. Liu, and M. Zwicker. Point2Sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network. In *Association for the Advancement of Artificial Intelligence*, 2019.

[10] L. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[11] C. Qi, H. Su, K. Mo, and L. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[12] C. Qi, L. Yi, H. Su, and L. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5105–5114, 2017.

[13] R. Rusu and S. Cousins. 3D is here: Point cloud library (PCL). In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1–4, 2011.

[14] Y. Shen, C. Feng, Y. Yang, and D. Tian. Mining point cloud local structures by kernel correlation and graph pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4548–4557, 2018.

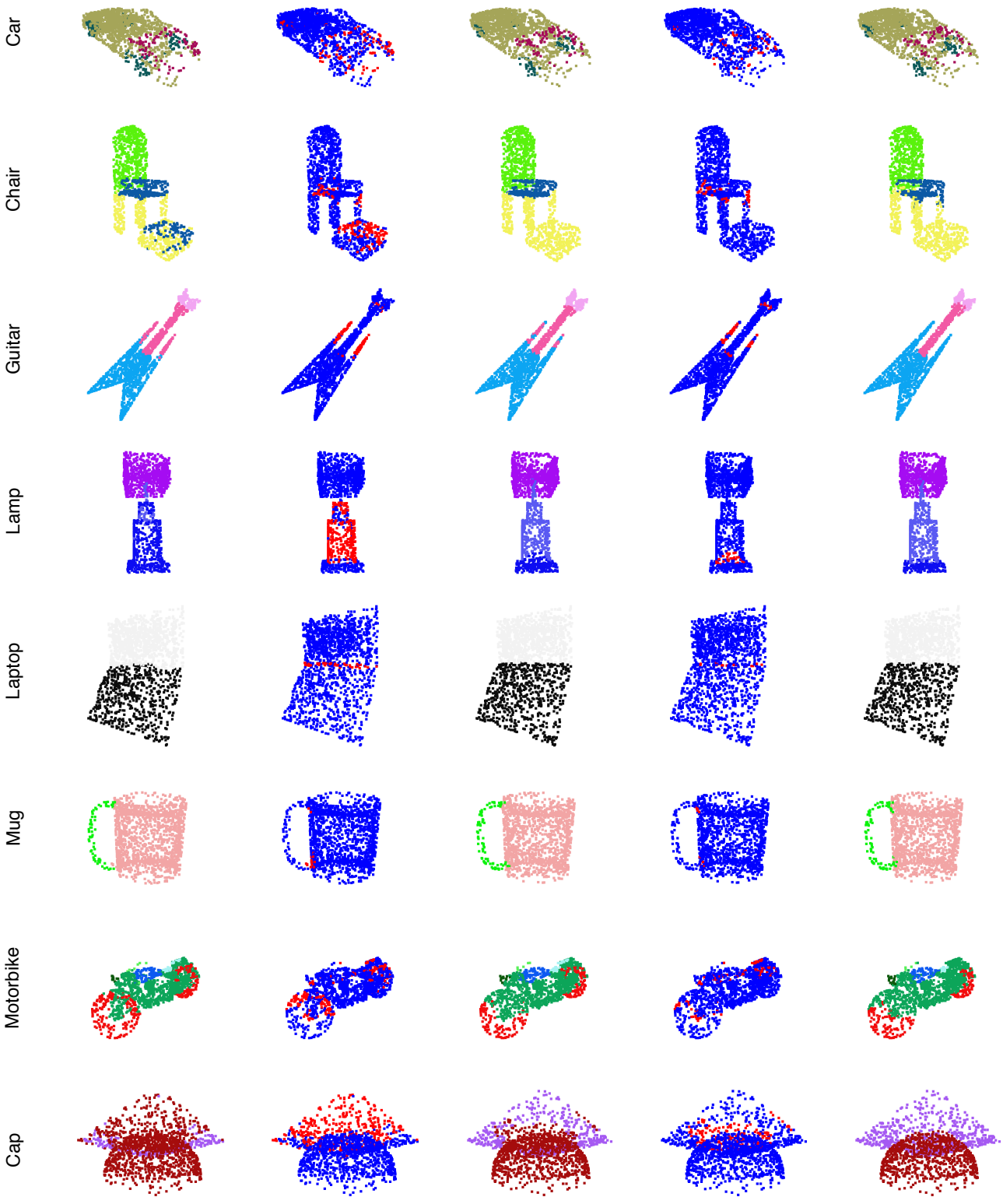
[15] M. Tatarchenko, J. Park, V. Koltun, and Q.-Y. Zhou. Tangent convolutions for dense prediction in 3D. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3887–3896, 2018.

[16] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong. O-CNN: Octree-based convolutional neural networks for 3D shape analysis. *ACM Transactions on Graphics*, 36(4):72, 2017.

[17] W. Wang, R. Yu, Q. Huang, and U. Neumann. SGPN: Similarity group proposal network for 3D point cloud instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2569–2578, 2018.

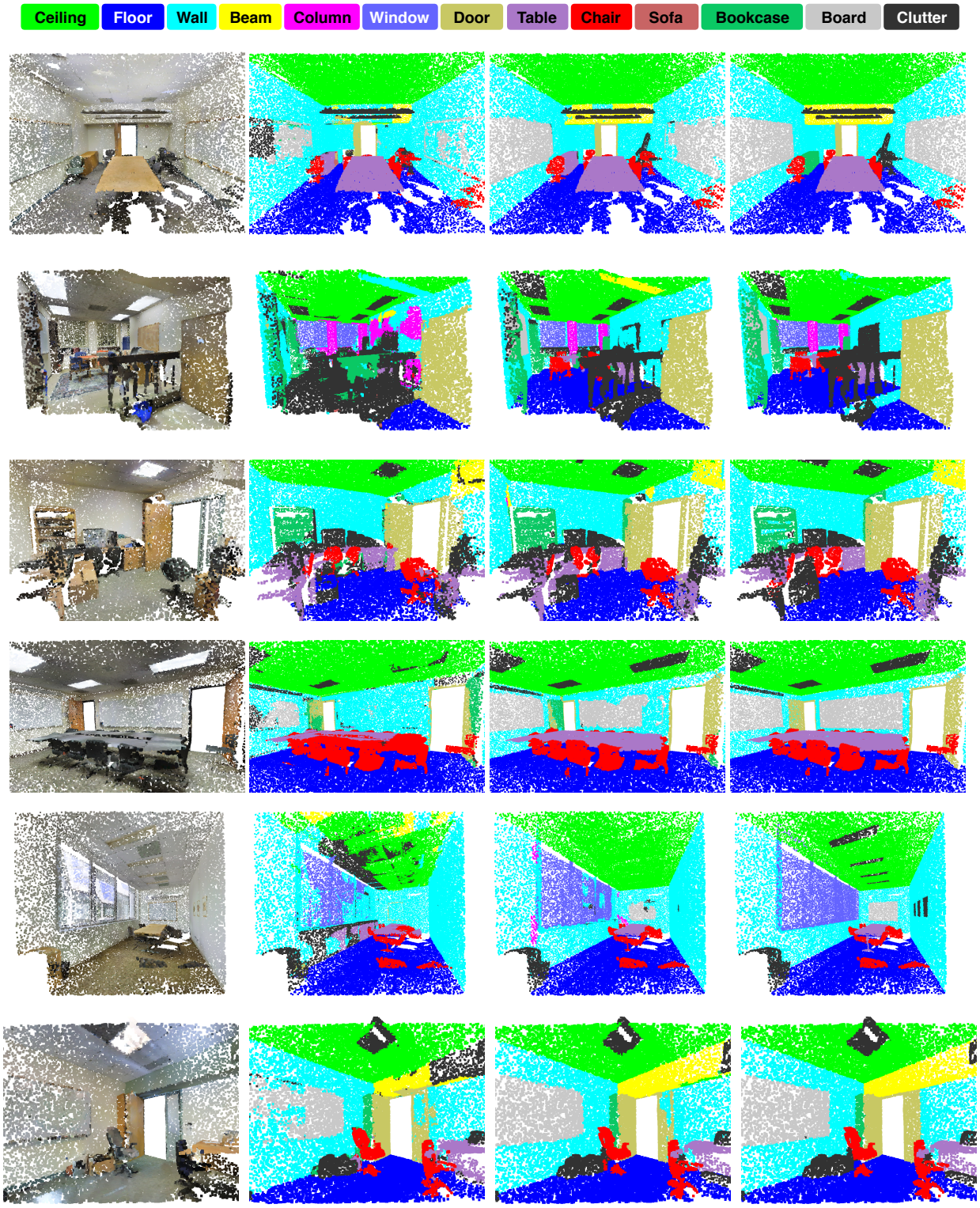
[18] X. Ye, J. Li, H. Huang, L. Du, and X. Zhang. 3D recurrent neural networks with context fusion for point cloud semantic segmentation. In *Proceedings of The European Conference on Computer Vision*, September 2018.

[19] L. Yi, H. Su, X. Guo, and L. Guibas. SyncSpecCNN: Synchronized spectral CNN for 3D shape segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6584–6592, 2017.



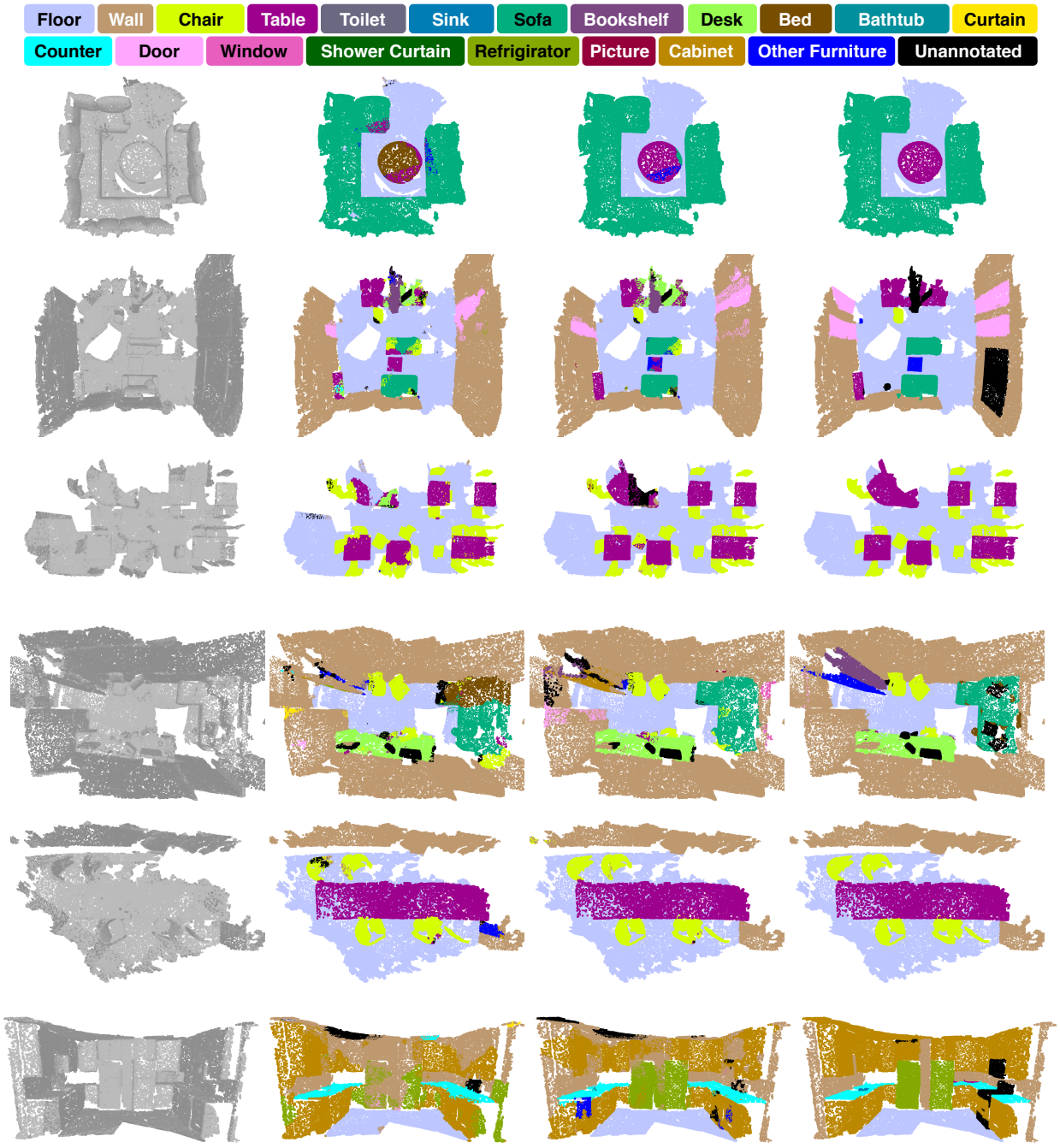
(a) **PointNet++ [12]** (b) **PointNet++ (diff)** (c) **Our** (d) **Our (diff)** (e) **Ground Truth**

Figure 6: More segmentation results on *ShapeNet-part* dataset. Second and fourth columns show the differences between ground truth and prediction (red points are mislabeled points) of PointNet++ and our method.



(a) **Input** (b) **PointNet [11]** (c) **Our** (d) **Ground Truth**

Figure 7: The visualization results on *S3DIS* dataset. We compare our model with PointNet [11] and the ground truth. The challenging sample rooms have been picked from the all six areas: area 1 (*row 1*), area 2 (*row 2*) area 3 (*row 3*), area 4 (*row 4*), area 5 (*row 5*), and area 6 (*row 6*).



(a) **Input** (b) **PointNet++ [12]** (c) **Our** (d) **Ground Truth**

Figure 8: The visualization results on *ScanNet* dataset. We compare our model with PointNet++ [12] and the ground truth. The challenging sample rooms have been picked from the *ScanNet* dataset.