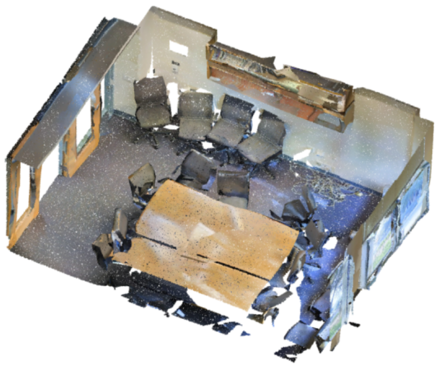
PS^2-Net: A Locally and Globally Aware Network for Point-Based Semantic Segmentation


Na Zhao


Tat-Seng Chua

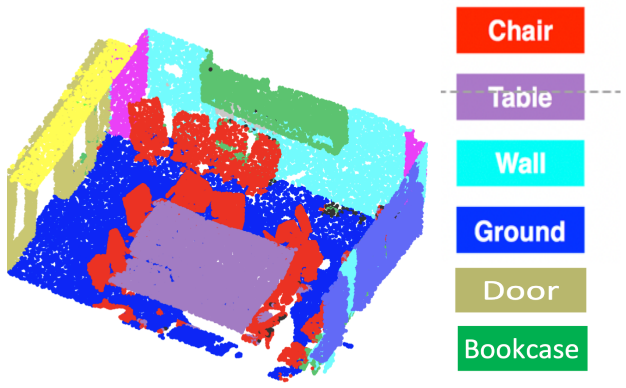

Gim Hee Lee

3D Semantic Segmentation



Deep Model ?

+ permutation-invariant



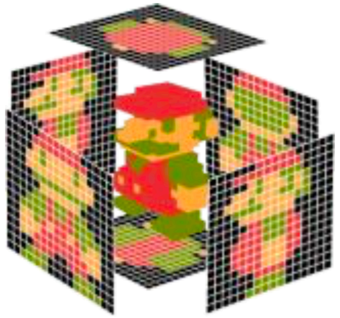
Input: 3D point cloud

- unordered
- irregular

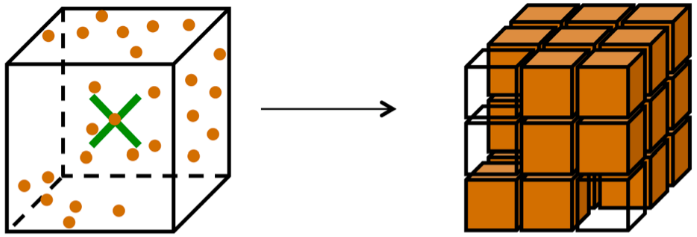
Output: segmented point cloud

Indirect methods

- To circumvent the permutation invariance problem, indirect approaches convert point clouds into ordered and regular representations before feeding them into deep models.



Multi-view Projection
(2D CNN)

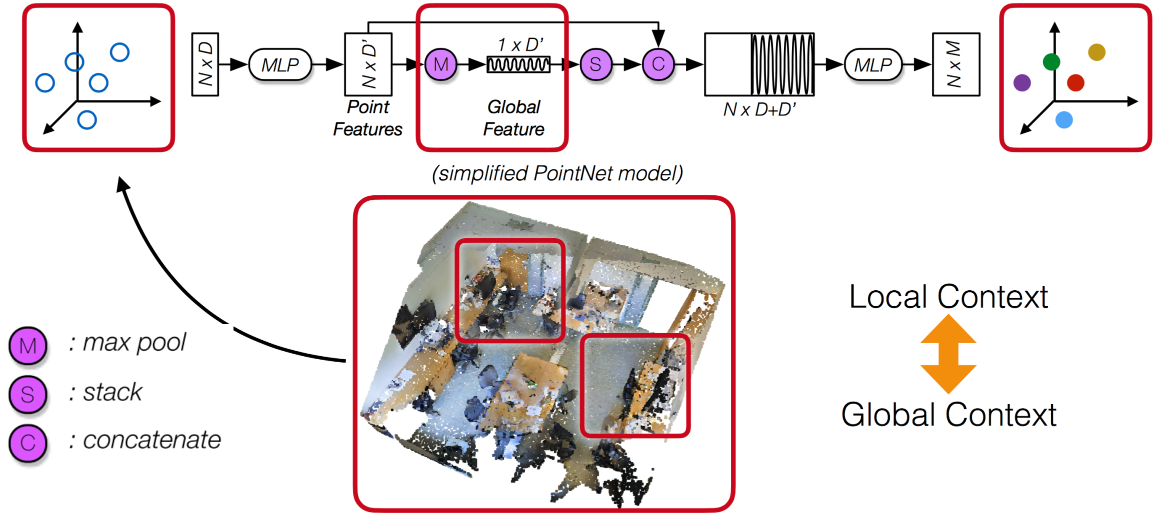


Voxel-grid Transformation
(3D CNN)

However, both approaches result in severe loss of information.

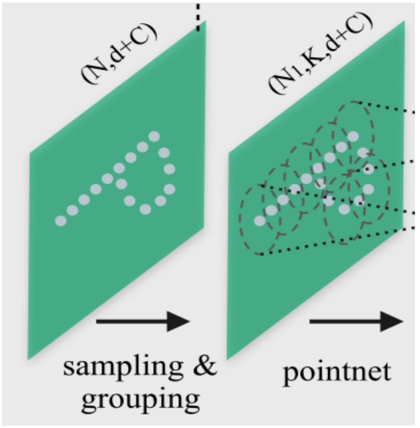
- Multi-view projection suffers information loss from occlusions and 3D to 2D projections.
- Voxel-grid transformation suffers from quantization error and the high computational cost of 3D CNN limits its scalability.

Direct methods

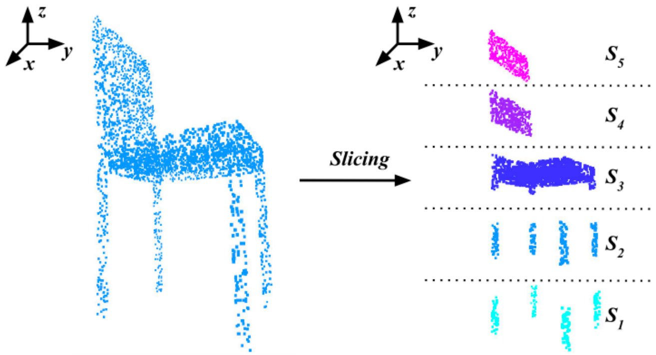


PointNet [CVPR 2017]

can not capture the local information



PointNet ++ [NeurIPS 2017]

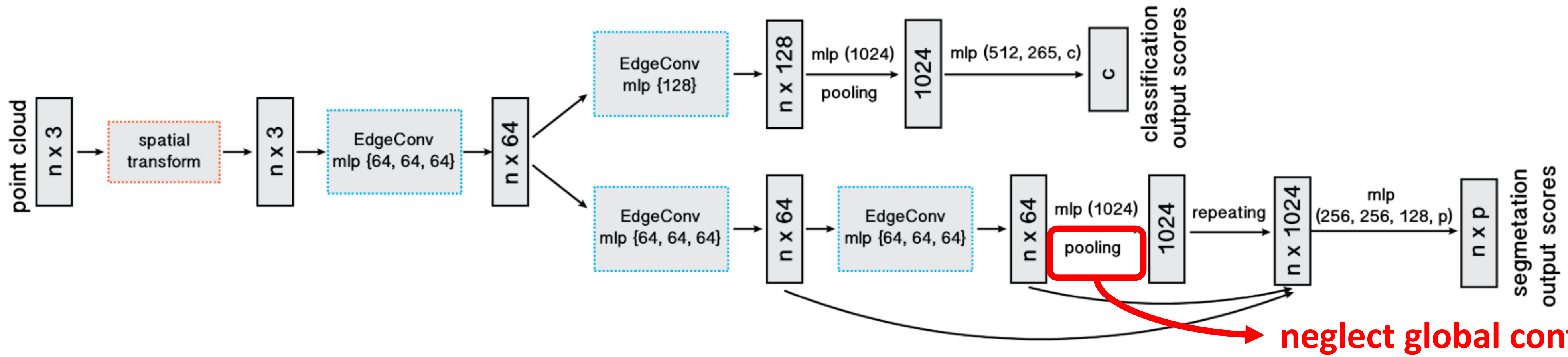


RSNet [CVPR 2018]

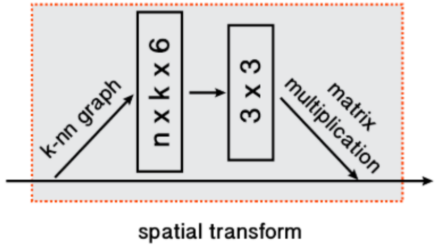
fail to model geometric relationships among neighbor points

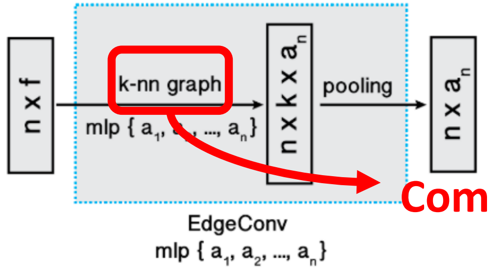
C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CVPR* 2017
C.R.Qi, L.Yi, H.Su, and L.J.Guibas. Pointnet++:Deep hierarchical feature learning on point sets in a metric space. NeurIPS 2017
Q. Huang, W. Wang, and U. Neumann. Recurrent slice net- works for 3d segmentation of point clouds. CVPR 2018

Direct method - DGCNN

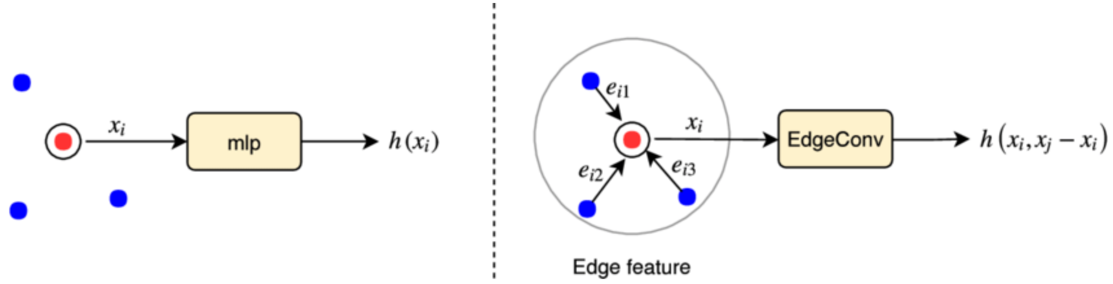


neglect global context





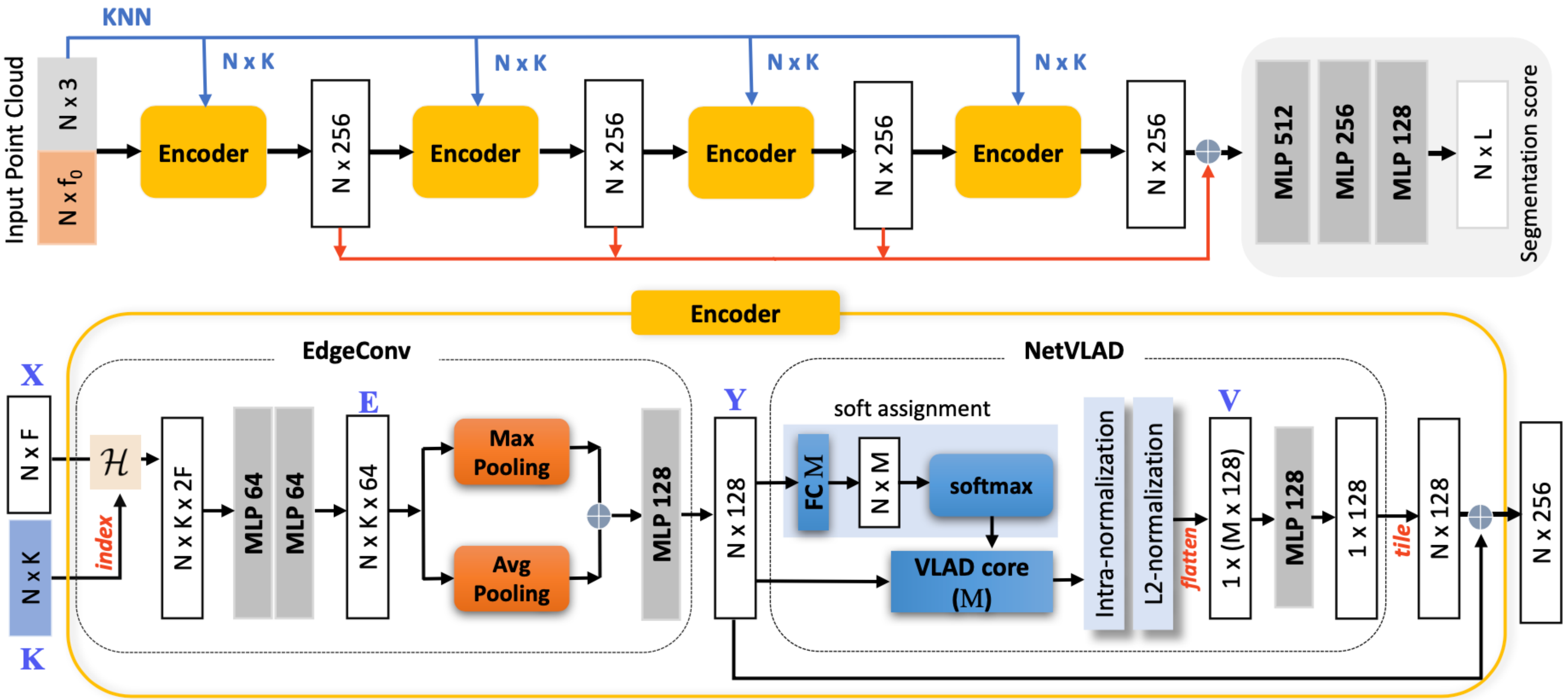
Computationally expensive



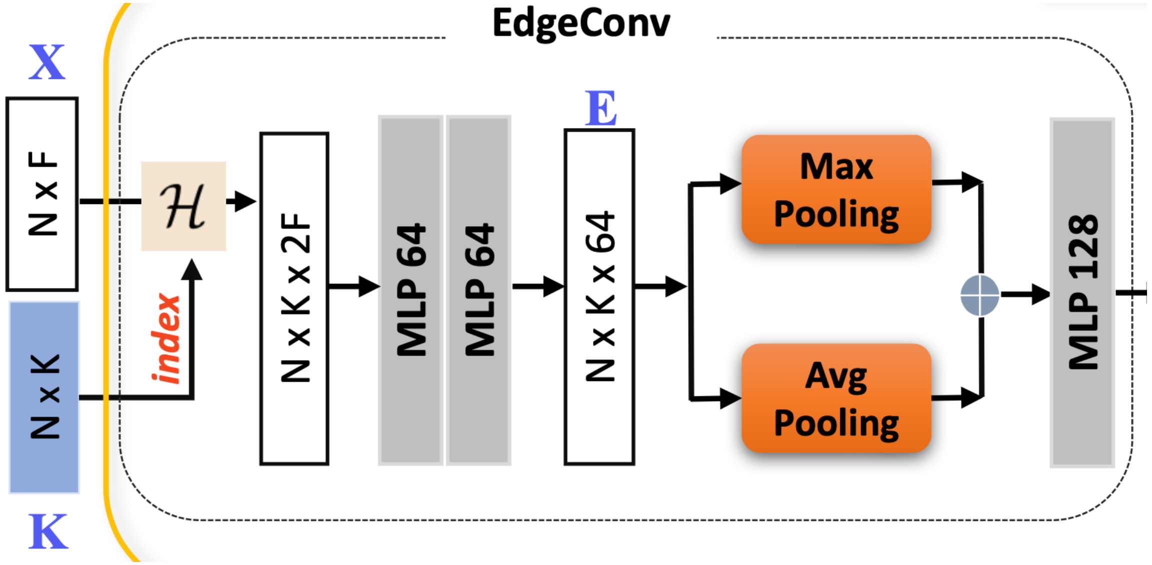
EdgeConv: embed the relationship between a point and its neighbors, then use a max-pooling over local neighborhood to capture *local geometric features.*

Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*, 2019

Our Proposed Permutation Invariant PS²-Net



Exploitation of Local Structure



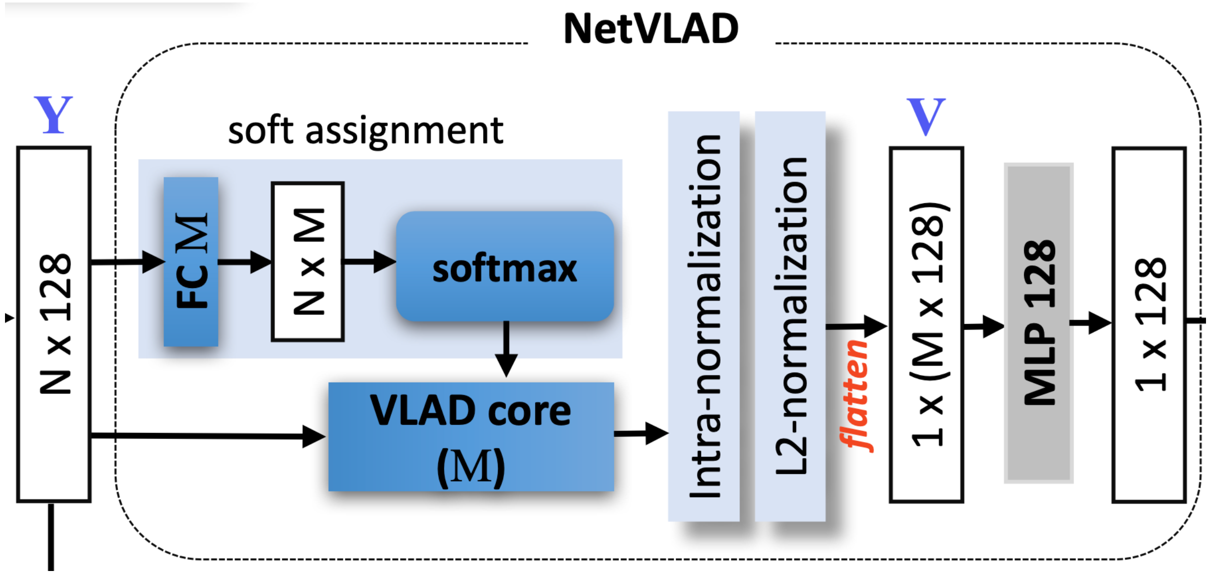
$$\mathbf{h}_i(j) = [\mathbf{x}_i, \mathbf{x}_j - \mathbf{x}_i], \quad j \in \mathbf{k}_i$$

$$\mathbf{e}_i = \sigma(\mathcal{F}(\sigma(\mathcal{F}(\mathbf{h}_i; \Theta_1)); \Theta_2))$$

$$\mathbf{y}_i = \sigma(\mathcal{F}([\max_{k \in \mathbf{k}_i}(\mathbf{e}_k), \operatorname{avg}_{k \in \mathbf{k}_i}(\mathbf{e}_k)]; \ \Theta_3))$$

- Use static KNN graphs
  - supervise local structure learning with spatial constraint
  - analogous to the structure of images, where the neighborhoods of pixels remain fixed during convolution.
- Apply two symmetric operations
  - Max-pooling: emphasize the strong responses in the local regions
  - Avg-pooling: preserve fine-grained local information

Aggregation of Global Context



$$\mathbf{v}_m = \sum_{i=1}^{N} \frac{e^{\mathbf{w}_m^T \mathbf{y}_i + b_m}}{\sum_{m'=1}^{M} e^{\mathbf{w}_{m'}^T \mathbf{y}_i + b_{m'}}} (\mathbf{y}_i - \mathbf{c}_m).$$

- Global context provides scene-level semantic information, which can alleviate local confusions.
- NetVLAD reveals fine-grained global contexts due to the large receptive field and aggregation of the learned relationships with all points.

Results on S3DIS

Table 1. Comparison of performances on S3DIS dataset using 6-fold cross validation. Results in the upper and lower table are obtained with data preparation setups in [18] (denoted as *P1*) and [15] (denoted as *P2*), respectively. Class-wise IoU is also given.

Method	**OA**	**mIoU**	**ceiling**	**floor**	**wall**	**beam**	**column**	**window**	**door**	**table**	**chair**	**sofa**	**bookcase**	**board**	**clutter**
PointNet [18]	78.5	47.6	88	88.7	69.3	42.4	23.1	47.5	51.6	54.1	42	9.6	38.2	29.4	35.2
G+RCU [8]	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	30	41.9
DGCNN [26]	84.1	56.1	—	—	—	—	—	—	—	—	—	—	—	—	—
RNNCF [28]	**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**	36.7	51.6
RSNet [10]	—	56.47	92.48	92.83	78.56	32.75	34.37	51.62	**68.11**	60.13	59.72	**50.22**	16.42	44.85	52.03
PS^2-Net—P1	86.69	**61.56**	**93.40**	**95.64**	**79.94**	37.17	**40.93**	**59.83**	66.65	**63.65**	65.71	37.16	49.83	**54.56**	**55.66**
PointCNN [15]	88.14	65.39	**94.78**	**97.3**	75.82	**63.25**	51.71	58.38	57.18	**71.63**	69.12	39.08	**61.15**	52.19	58.59
PS^2-Net—P2	**88.22**	**66.60**	93.04	96.26	**83.22**	41.61	**54.05**	**60.08**	**70.40**	67.37	**73.13**	**48.75**	58.73	**58.68**	**60.48**

Results on ScanNet

Table 2. Comparison of performances on ScanNet dataset using XYZ information as input. Results in the upper and lower table are obtained with data preparation setups in [19] (denoted as *P3*) and [15] (denoted as *P2*), respectively. Class-wise IoU is also given.

Method	**OA**	**mIoU**	**wall**	**floor**	**chair**	**table**	**desk**	**bed**	**bookshelf**	**sofa**	**sink**	**bathtub**
PointNet [18]	—	14.69	69.44	88.59	35.93	32.78	2.63	17.96	3.18	32.79	0	0.17
PointNet++ [19]	—	34.26	77.48	92.5	64.55	46.6	12.69	51.32	52.93	52.27	30.23	42.72
RSNet [10]	—	39.35	**79.23**	**94.1**	64.99	**51.04**	**34.53**	**55.95**	**53.02**	**55.41**	**34.84**	49.38
PS^2-Net—P3	—	**40.17**	72.44	91.51	**65.08**	45.61	26.27	48.90	39.96	53.94	24.58	**64.78**
PointCNN [15]	85.1	—	—	—	—	—	—	—	—	—	—	—
PS^2-Net—P2	**87.21**	44.90	77.02	91.22	68.36	56.66	31.62	53.55	36.32	58.75	43.07	70.11

Method	**toilet**	**curtain**	**counter**	**door**	**window**	**shower curtain**	**refridgerator**	**picture**	**cabinet**	**other furniture**
PointNet [18]	0	0	5.09	0	0	0	0	0	4.99	0.13
PointNet++ [19]	31.37	32.97	20.04	2.02	3.56	27.43	18.51	0	23.81	2.2
RSNet [10]	54.16	6.78	22.72	3	8.75	29.92	**37.9**	0.95	**31.29**	**18.98**
PS^2-Net—P3	**60.87**	**41.20**	**24.62**	**8.37**	**21.55**	**47.24**	19.68	**2.63**	28.02	16.22
PointCNN [15]	—	—	—	—	—	—	—	—	—	—
PS^2-Net—P2	66.28	41.94	23.73	10.94	17.82	51.02	44.19	3.17	32.82	19.33

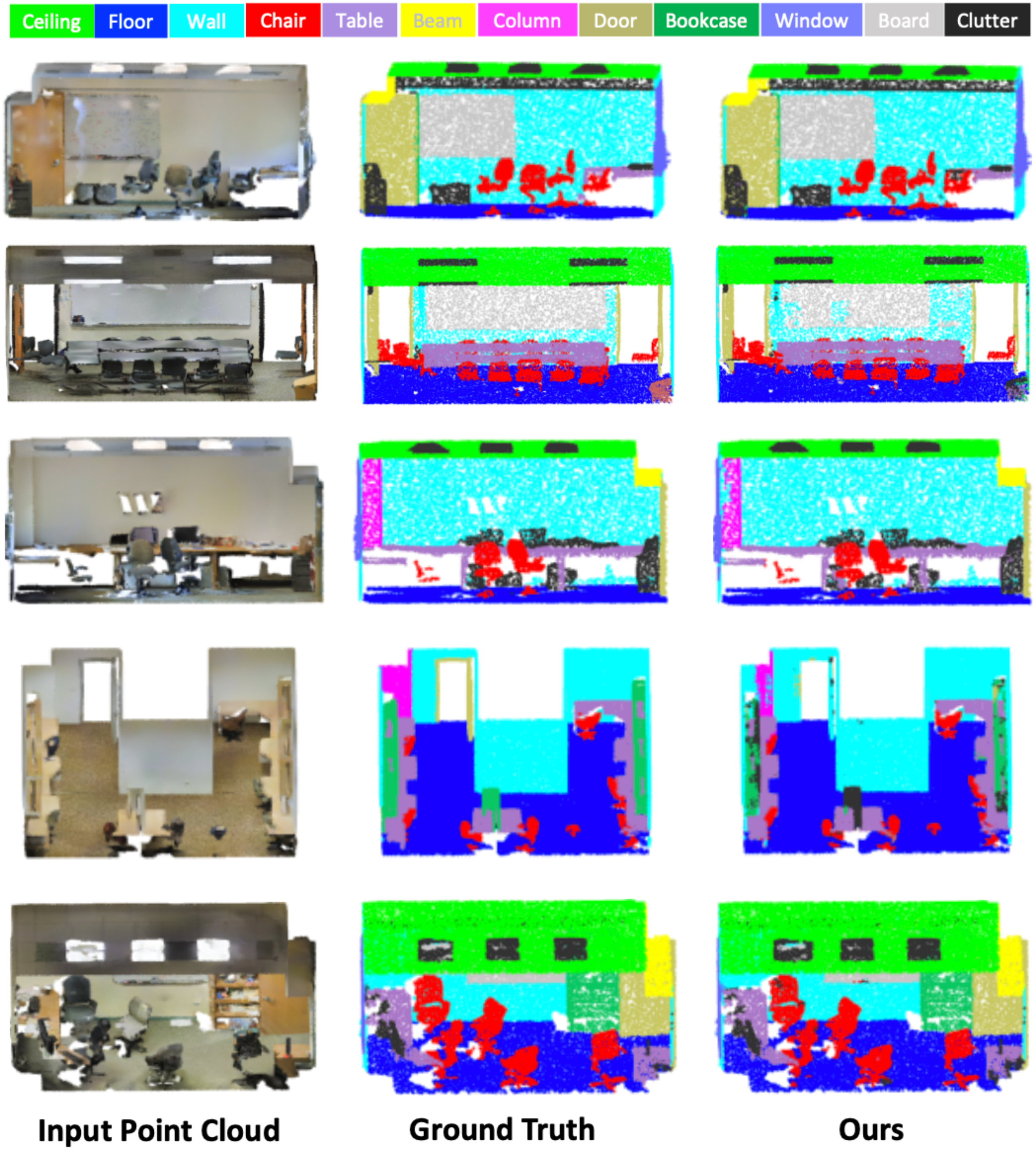


Figure 3. PS^2-Net semantic segmentation results on S3DIS.

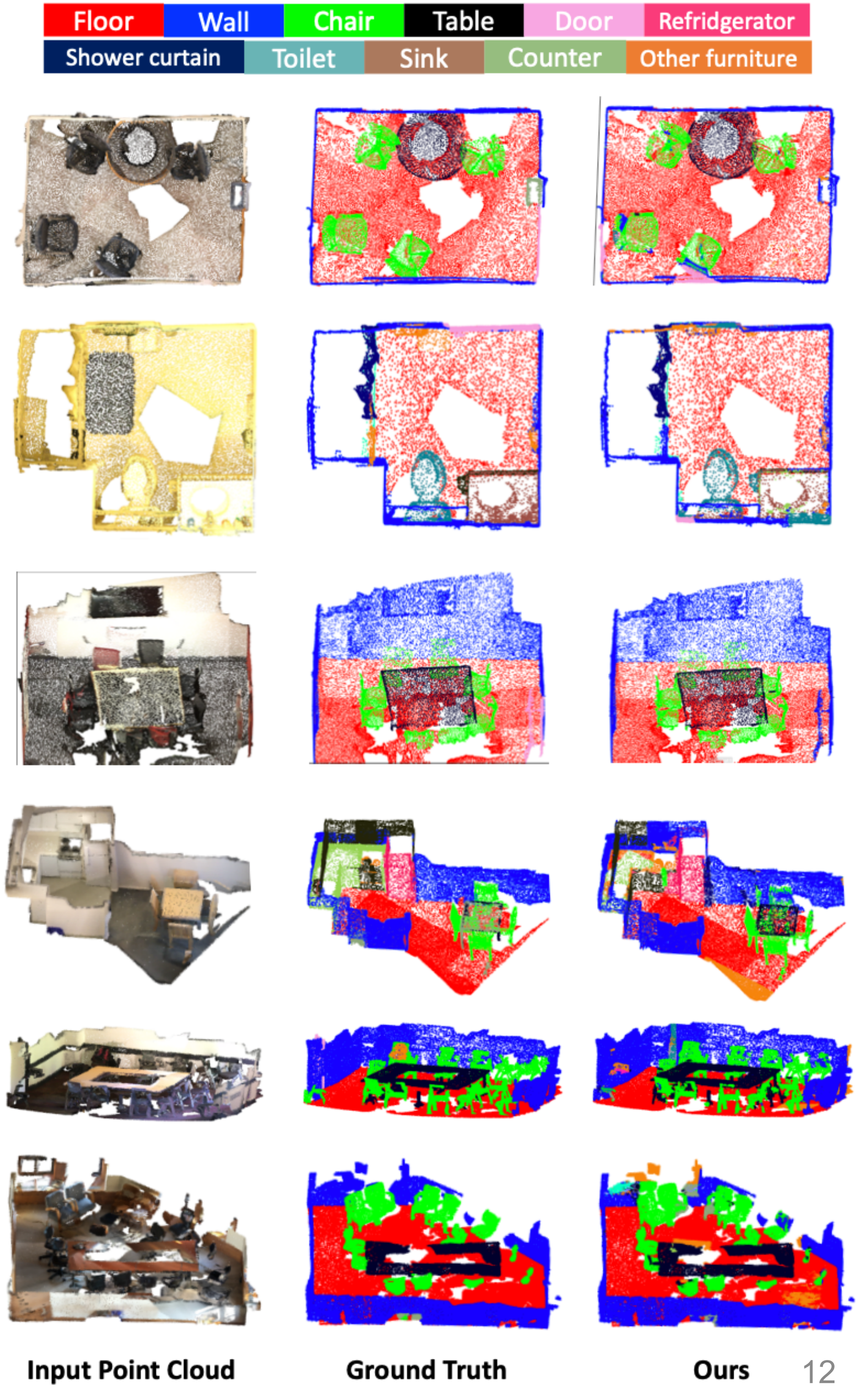
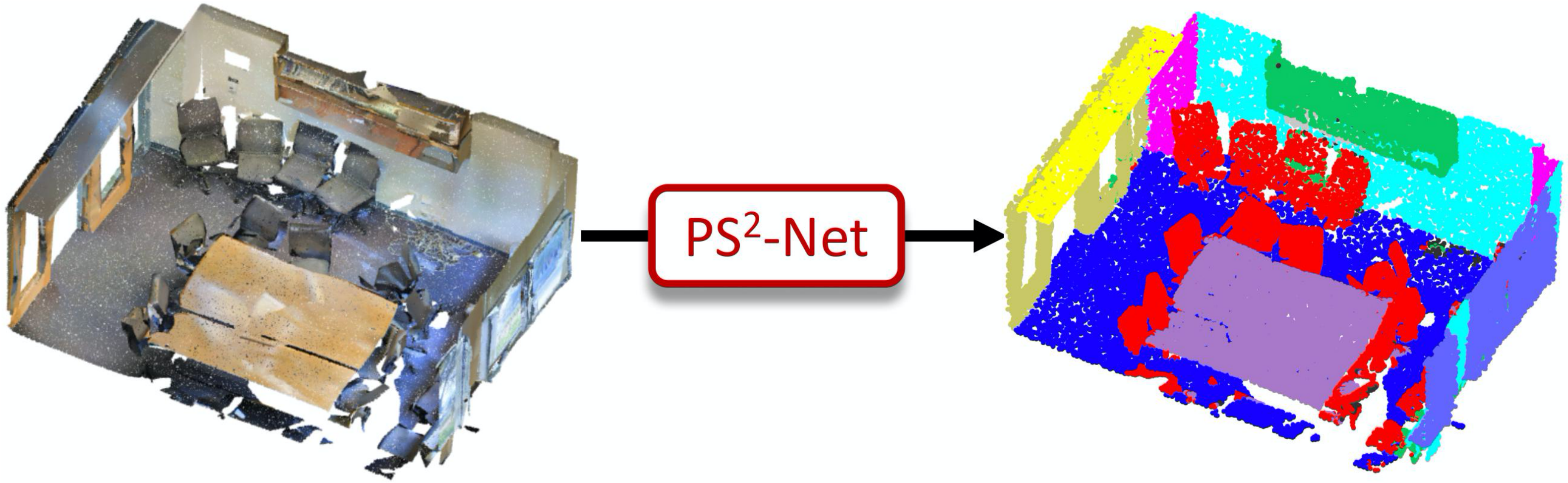


Figure 4. PS^2-Net semantic segmentation results on ScanNet.

THANK YOU

For more information, please visit our project website:
https://github.com/Na-Z/PS-2Net



Input: 3D point cloud

Output: Segmented Result