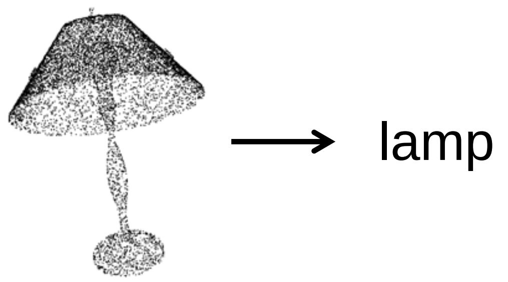
Deep Learning in 3D Point Cloud Processing

Yongcheng Liu

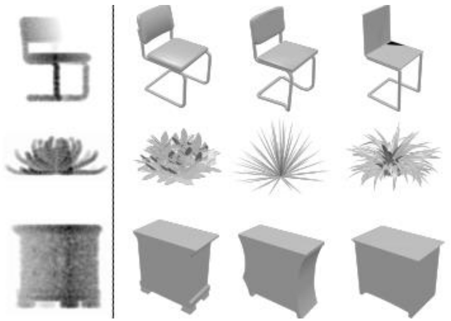
2019.05

Introduction

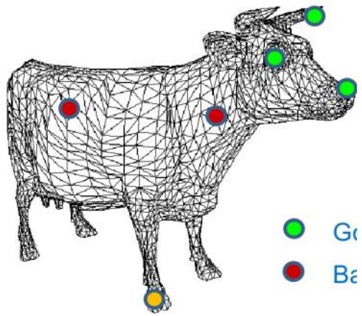
Introduction *tasks*



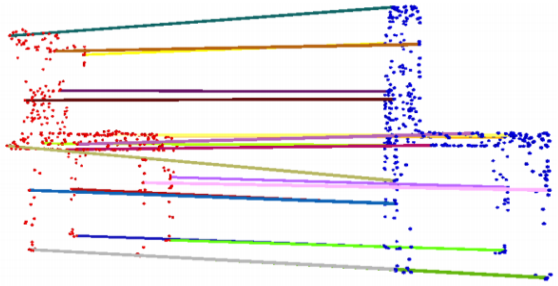
shape classification



shape retrieval

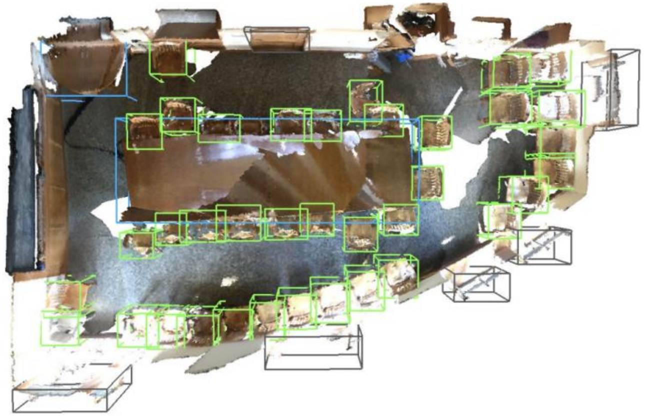


keypoint detection

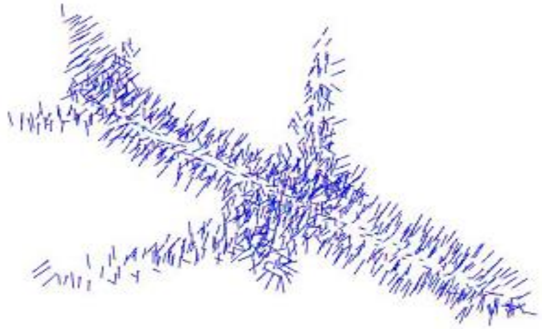


shape correspondence

semantic segmentation



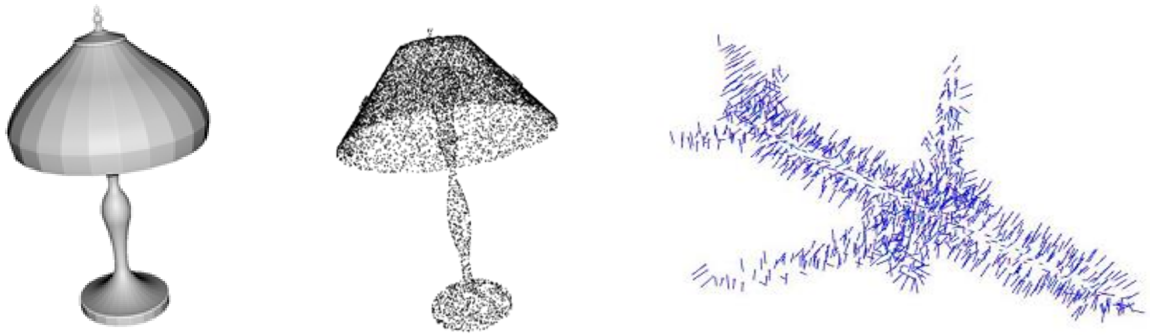
object detection

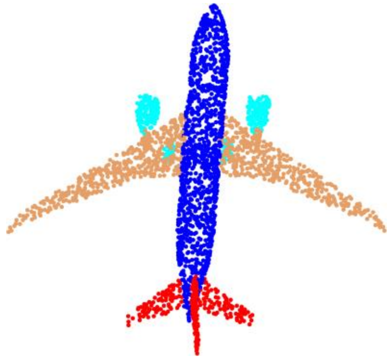


normal estimation

......

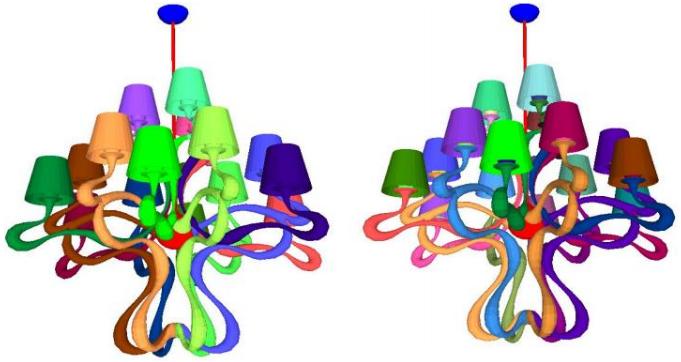
Introduction *datasets*

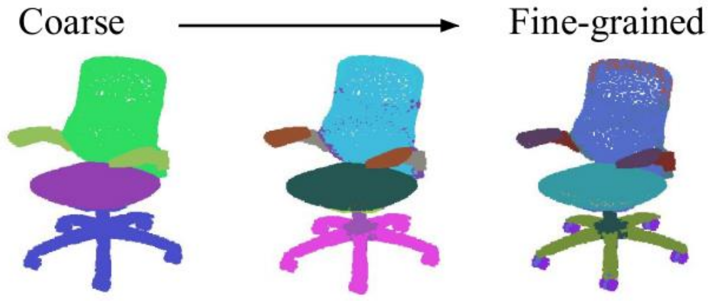




Princeton ModelNet: 1k

ShapeNet Part: 2k





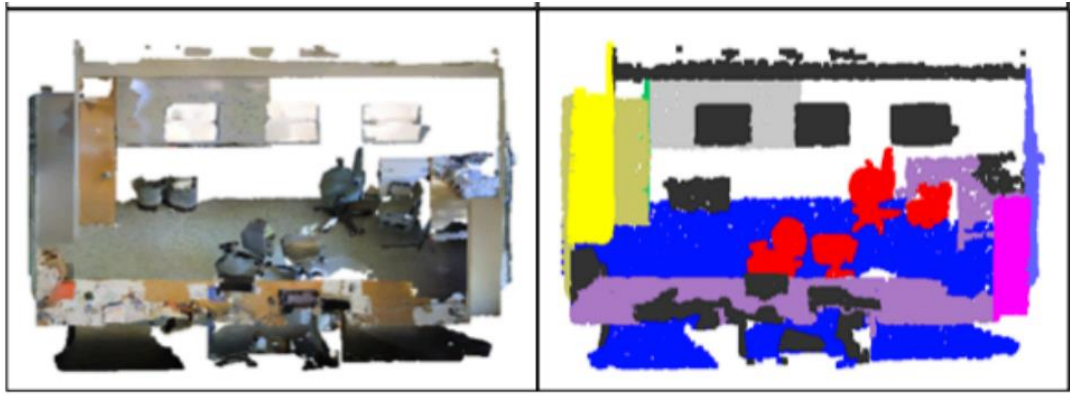
......

PartNet models

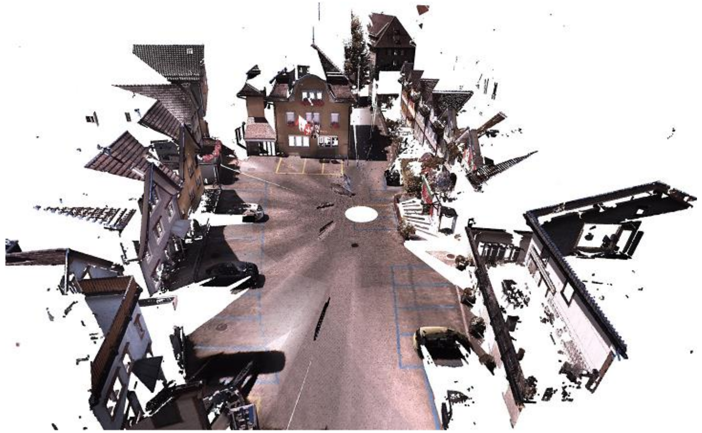
Hierarchical Semantic Segmentation

Mo et al. PartNet: A Large-scale Benchmark for Fine-grained and Hierarchical Part-level 3D Object Understanding. CVPR 2019.
Yi et al. A scalable active framework for region annotation in 3D shape collections. TOG 2016.
Wu et al. 3D ShapeNets: A Deep Representation for Volumetric Shapes. CVPR 2015.

Introduction *datasets*



Stanford 3D indoor scene: 8k



Semantic 3D: 4 billion in total

ScanNet: seg + det

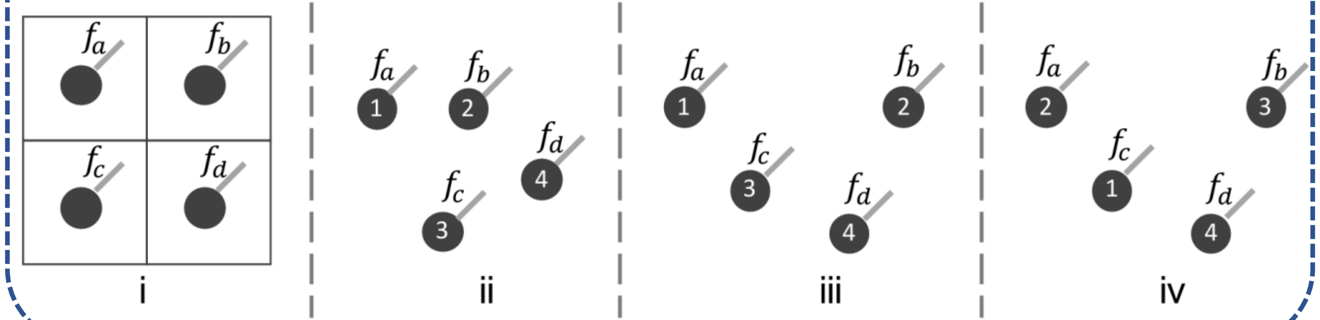
KITTI: det

......

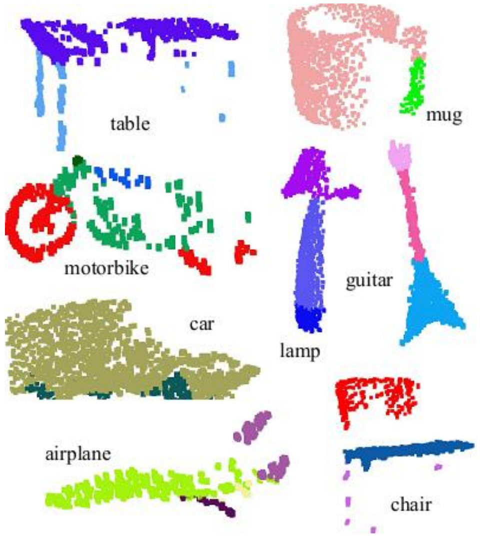
Dai et al. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. CVPR 2017.
Armeni et al. 3d semantic parsing of large-scale indoor spaces. CVPR 2016.
Hackel et al. Semantic3d. net: A new large-scale point cloud classification benchmark. ISPRS 2017.

Introduction *some challenges*

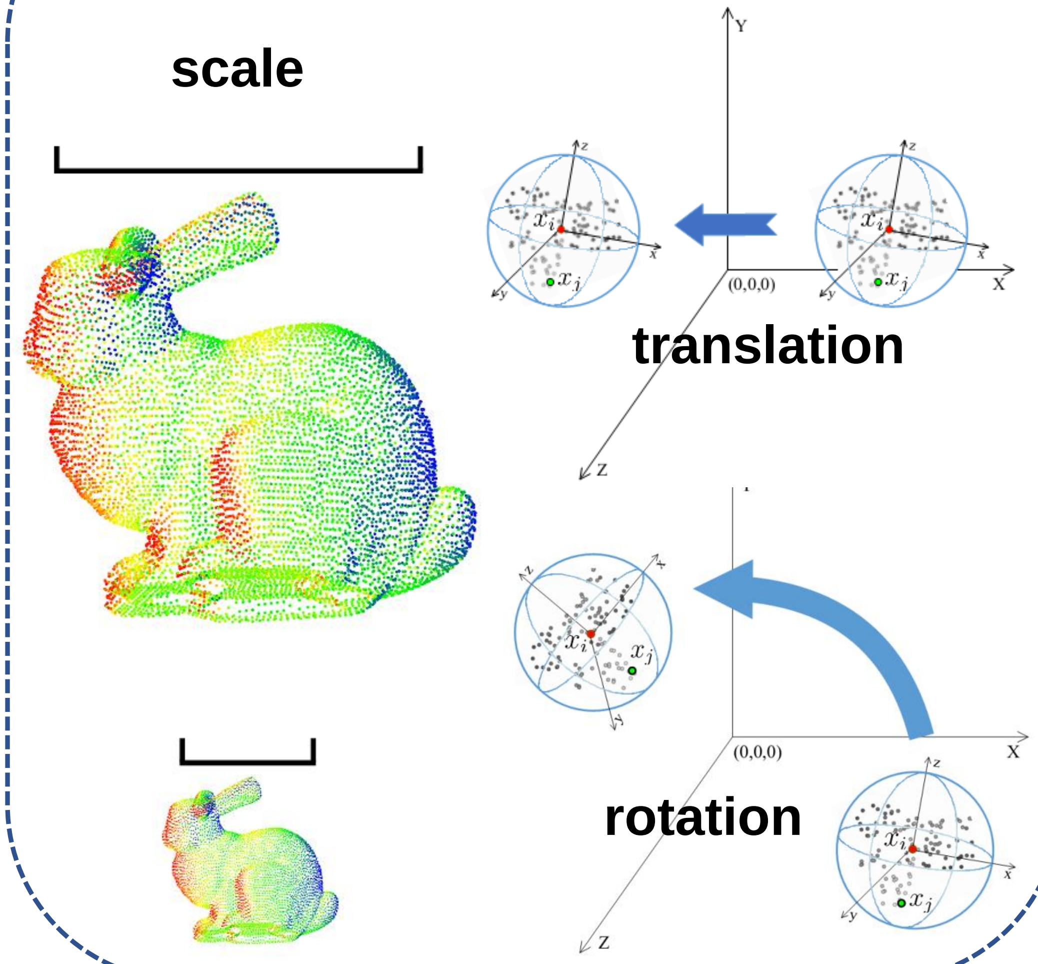
Irregular (unordered): permutation invariance



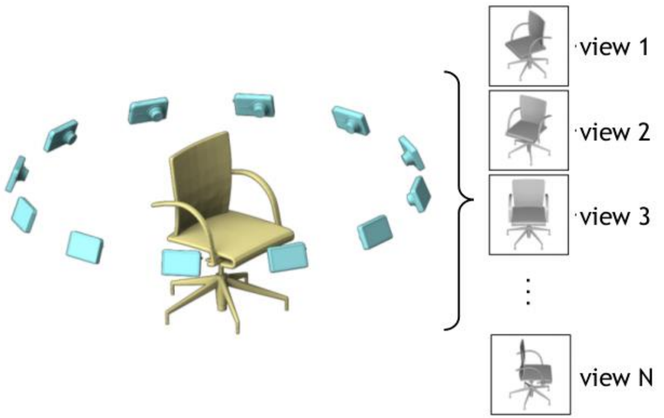
Robustness to corruption, outlier, noise; partial data



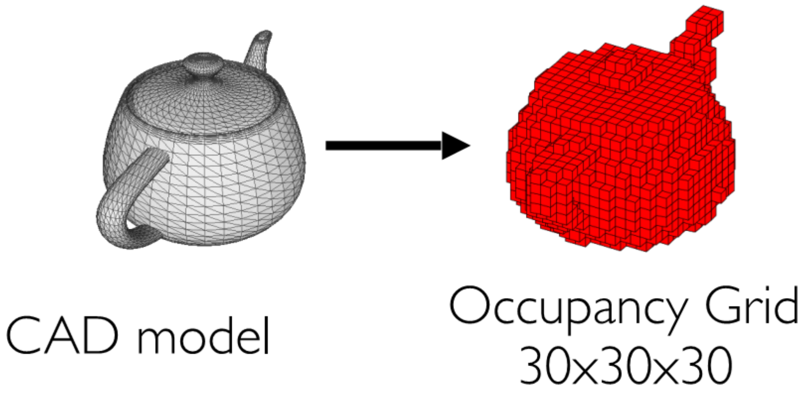
Robustness to rigid transformations



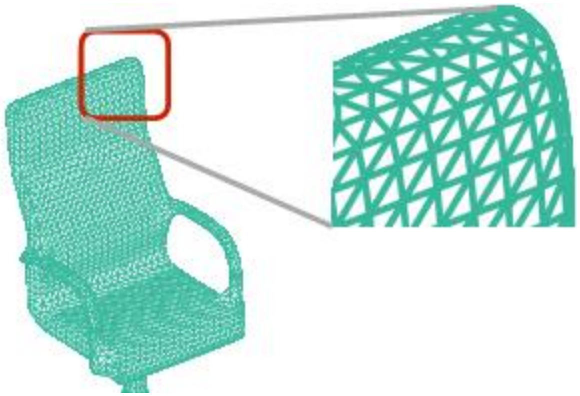
Introduction *3D representations*



multi-view images + 2D CNN



volumetric data + 3D CNN



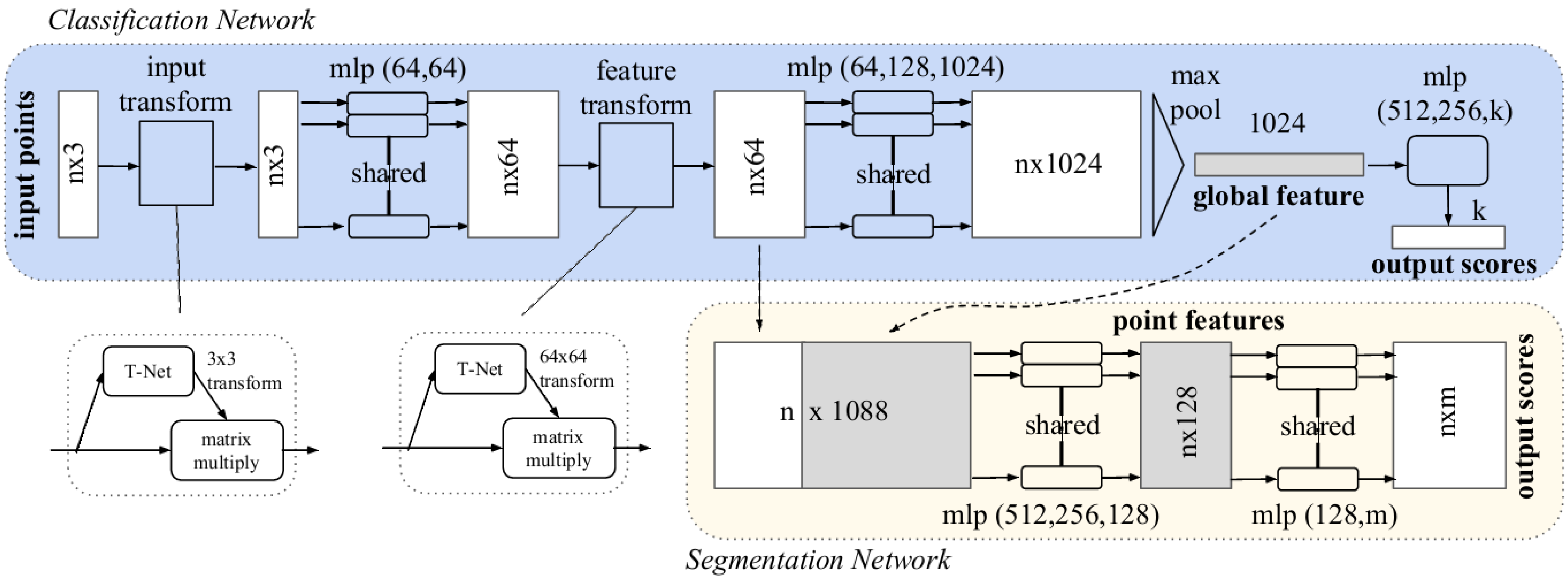
mesh data + DL (GNN) ?

image depth + CNN

point cloud + DL (CNN) ?

Related work – PointNet family

Related Work <u>*PointNet: permutation invariance*</u>

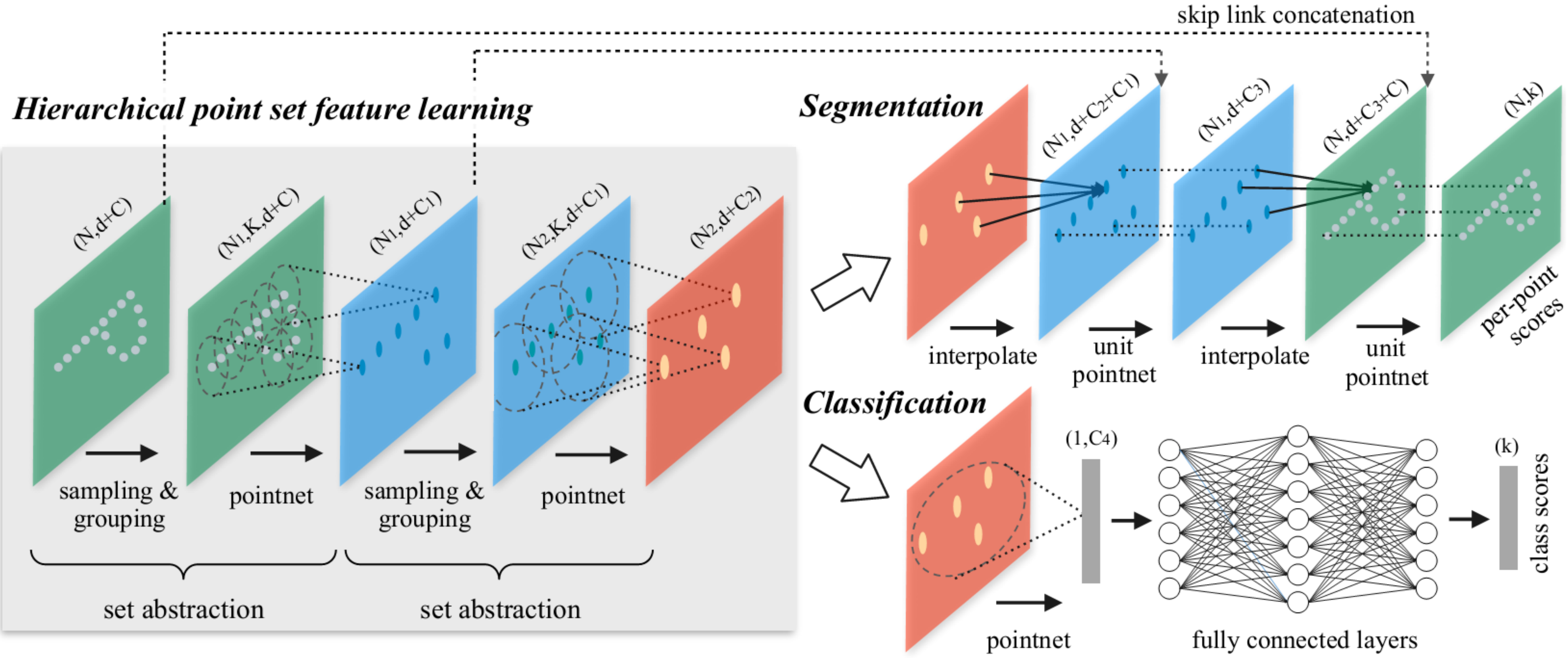


Shared MLP + max pool (symmetric function)

No local patterns capturing

Qi et al. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. CVPR 2017.

Related Work *PointNet++: local to global*



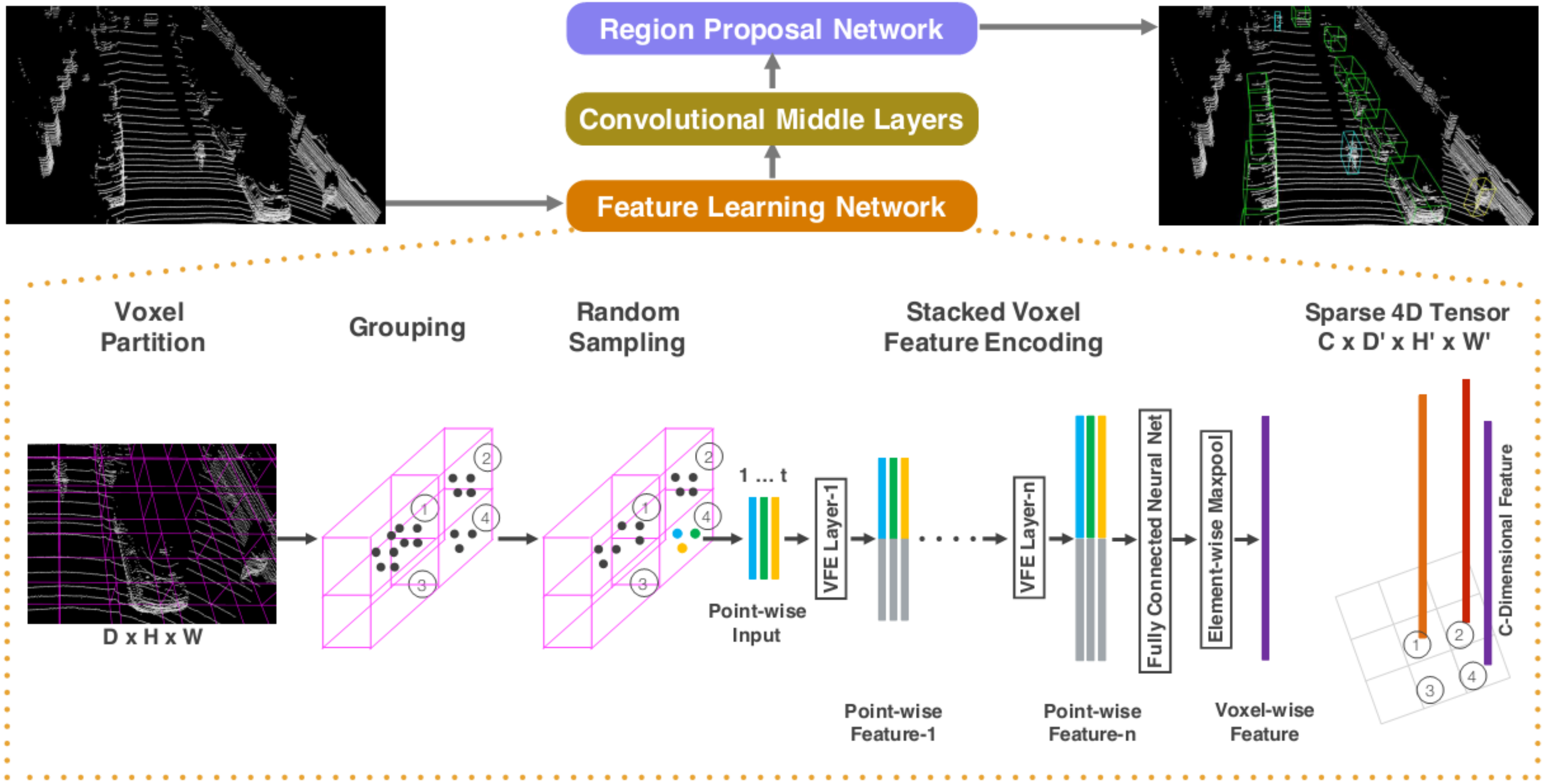
Sampling + Grouping + PointNet

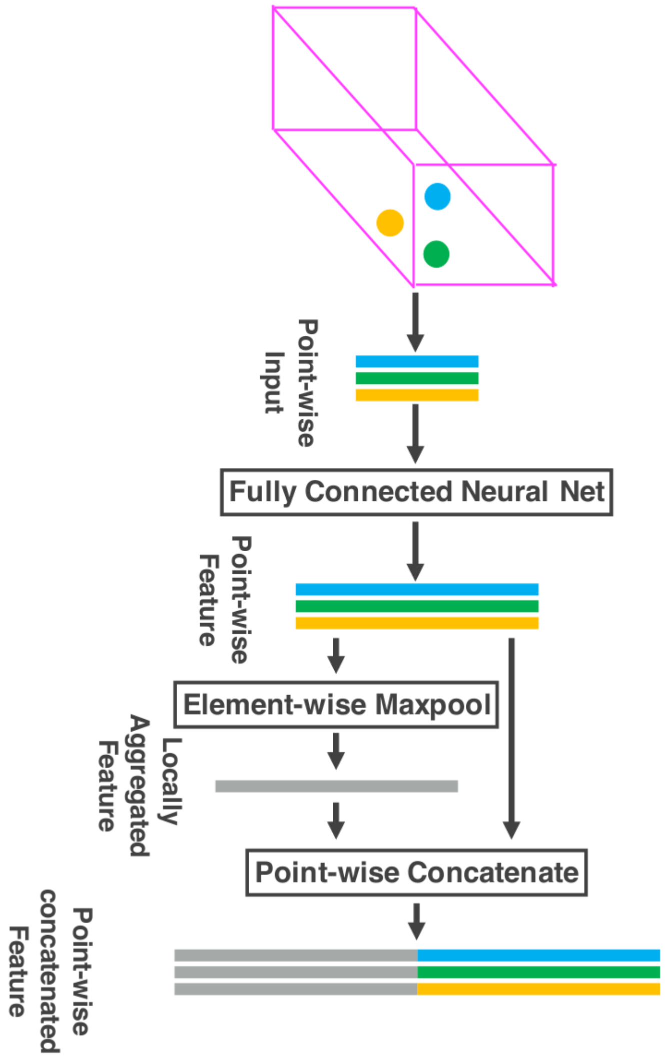
Only similar to CNN in framework

Qi et al. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. NIPS 2017.

Related work – regularization

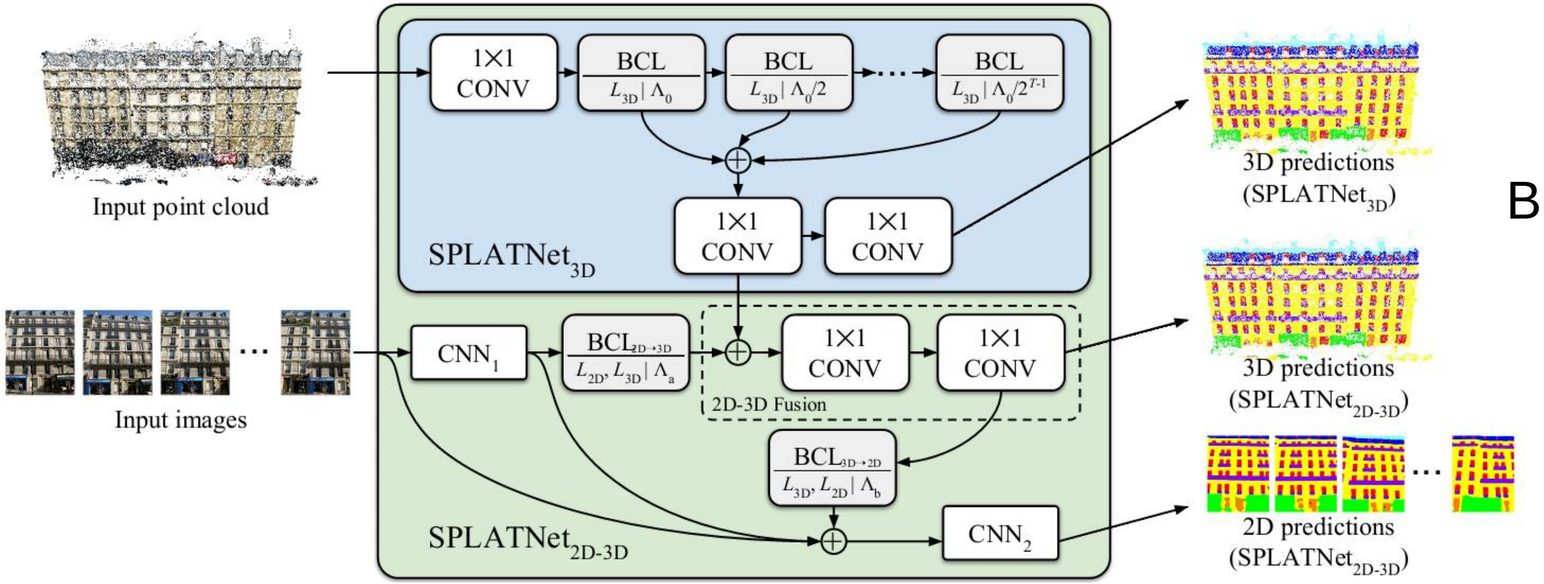
Related Work *VoxelNet: voxelization*



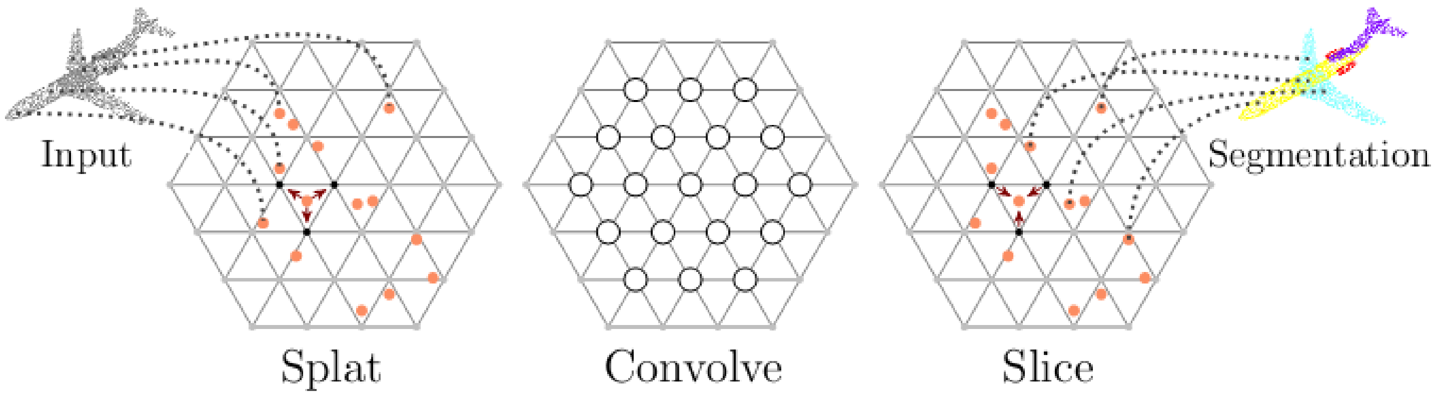


Zhou et al. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. CVPR 2018.

Related Work *SPLATNet: high-dimensional lattice*



BCL: Bilateral convolution layer



Kiefel et al. Permutohedral Lattice CNNs. ICLR 2015.
Jampani et al. Learning sparse high dimensional filters: Image filtering, dense CRFs and bilateral neural networks. CVPR 2016.
Su et al. SPLATNet: Sparse Lattice Networks for Point Cloud Processing. CVPR 2018.

Related Work *PointCNN: X-transformation*

In this paper, we propose to learn a $K \times K$ $\mathcal{X}$-transformation for the coordinates of K input points $(p_1, p_2, ..., p_K)$, with a multilayer perceptron [39], i.e., $\mathcal{X} = MLP(p_1, p_2, ..., p_K)$. Our aim is to use it to simultaneously weight and permute the input features, and subsequently apply a typical convolution on the transformed features. We refer to this process as $\mathcal{X}$-Conv, and it is the basic

ALGORITHM 1: $\mathcal{X}$-Conv Operator

Input : $\mathbf{K}$, p, $\mathbf{P}$, $\mathbf{F}$

Output : $\mathbf{F}_p$ ▷ Features "projected", or "aggregated", into representative point p

$\mathbf{P}' \leftarrow \mathbf{P} - p$ ▷ Move $\mathbf{P}$ to local coordinate system of p

$\mathbf{F}_\delta \leftarrow MLP_\delta(\mathbf{P}')$ ▷ **Individually** lift each point into C_δ dimensional space

$\mathbf{F}_* \leftarrow [\mathbf{F}_\delta, \mathbf{F}]$ ▷ Concatenate $\mathbf{F}_\delta$ and $\mathbf{F}$, $\mathbf{F}_*$ is a $K \times (C_\delta + C_1)$ matrix

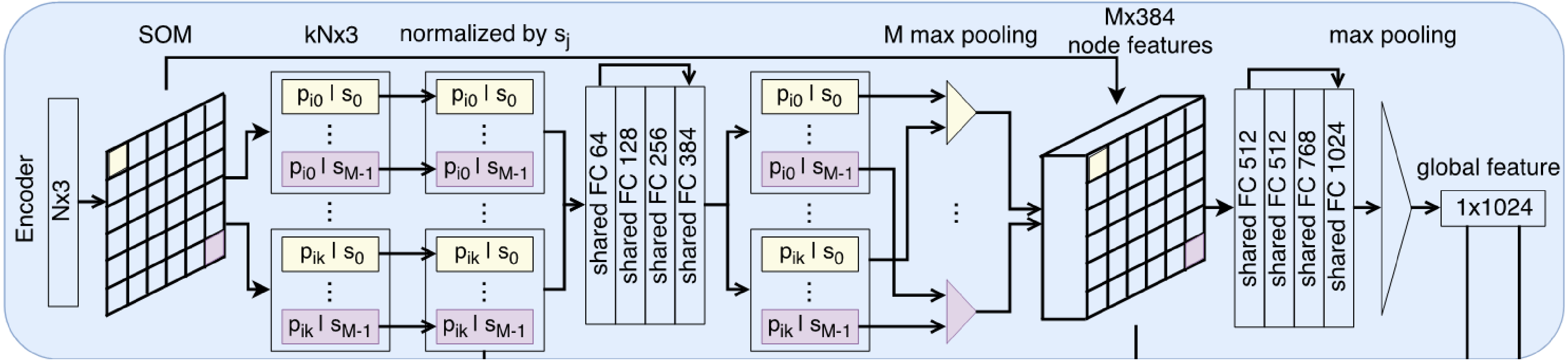
$\mathcal{X} \leftarrow MLP(\mathbf{P}')$ ▷ Learn the $K \times K$ $\mathcal{X}$-transformation matrix

$\mathbf{F}_\mathcal{X} \leftarrow \mathcal{X} \times \mathbf{F}_*$ ▷ Weight and permute $\mathbf{F}_*$ with the learnt $\mathcal{X}$

$\mathbf{F}_p \leftarrow \mathrm{Conv}(\mathbf{K}, \mathbf{F}_\mathcal{X})$ ▷ Finally, typical convolution between $\mathbf{K}$ and $\mathbf{F}_\mathcal{X}$

Li et al. PointCNN: Convolution On X-Transformed Points. NIPS 2018.

Related Work SO-Net: Self-Organizing Map (SOM)

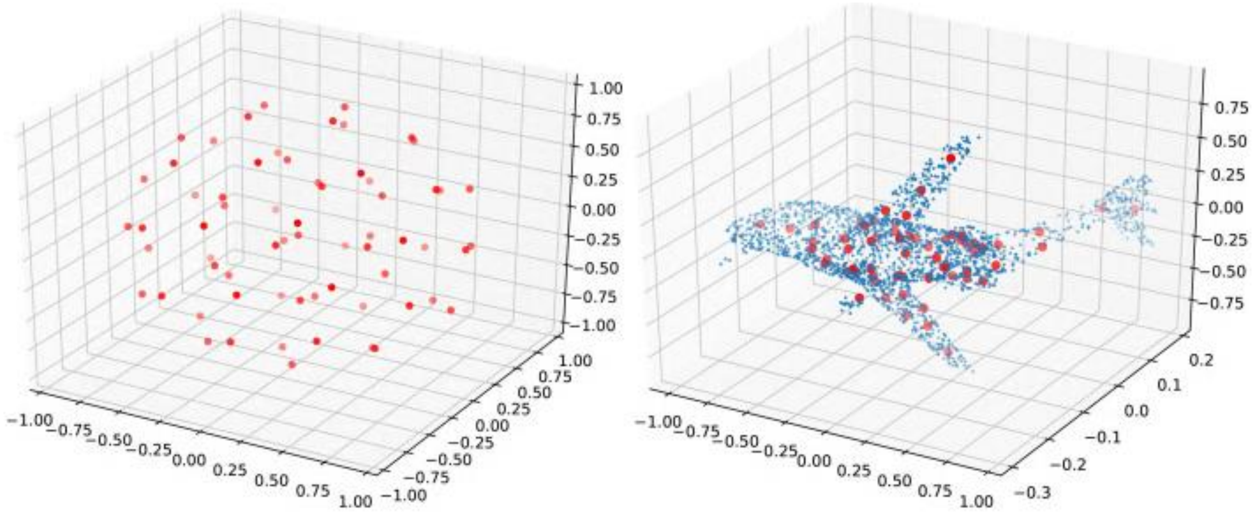


$$s_{ik} = \mathrm{kNN}(p_i \mid s_j,\ j = 0, \cdots, M-1).$$

$$p_{ik} = p_i - s_{ik}.$$

$$p_{ik}^{l+1} = \phi(W^l p_{ik}^l + b^l).$$

$$s_j^0 = \max(\{p_{ik}^l, \forall s_{ik} = s_j\}).$$



Li et al. SO-Net: Self-Organizing Network for Point Cloud Analysis. CVPR 2018.

Related work – robustness to rigid transformation

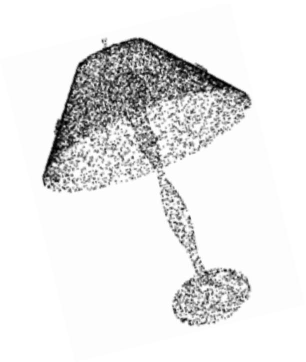
Related Work

Normalization:

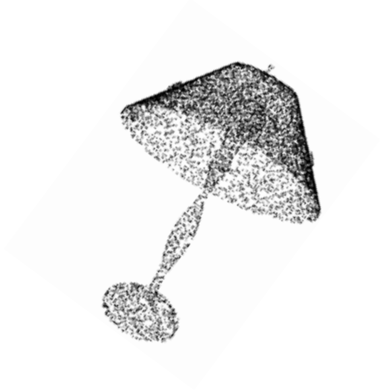
✓ Translation

✓ Scale

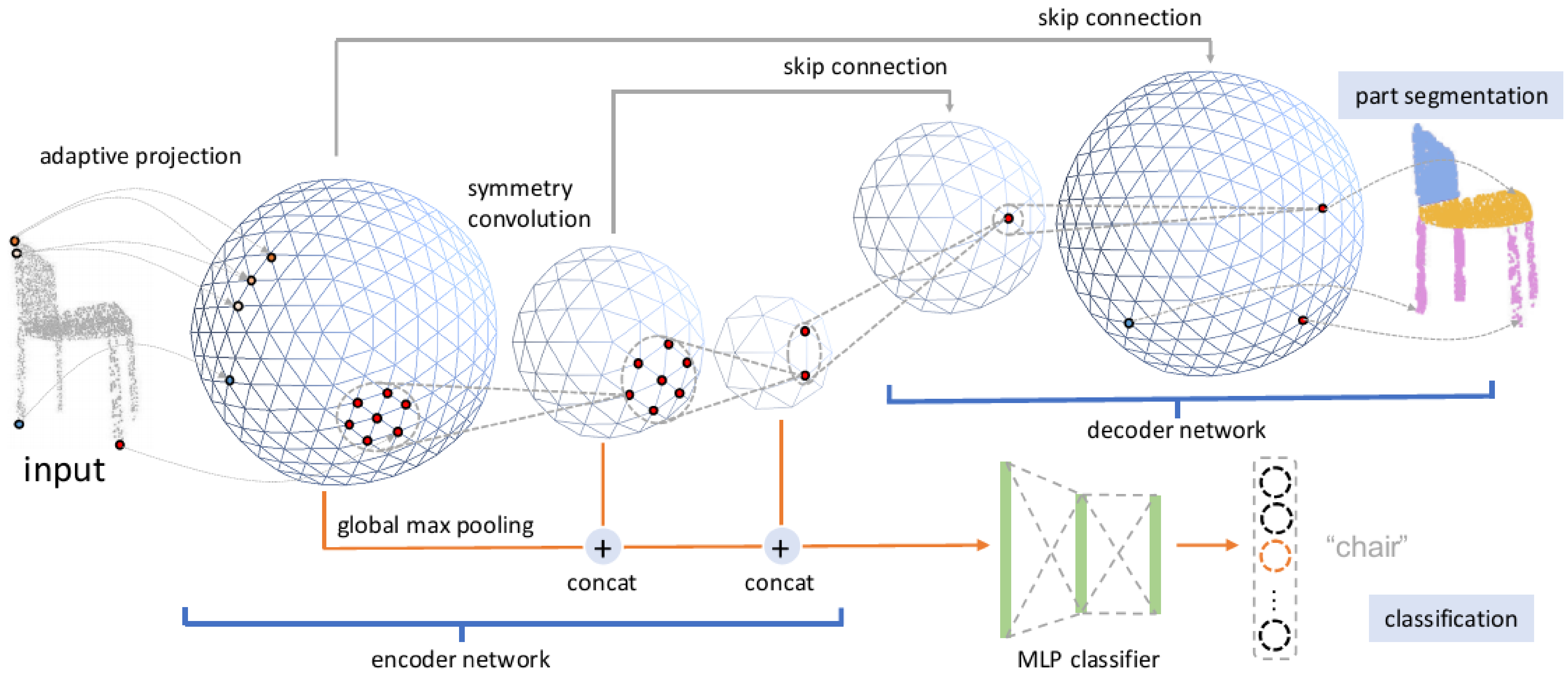
x Rotation



$\mathbf{X} \cdot \mathbf{R}$, (Nx3) · (3x3)



Related Work *SFCNN: Spherical Fractal CNN*



Cohen et al. Spherical CNNs. ICLR 2018.
Esteves et al. Learning so (3) equivariant repre-sentations with spherical cnns. ECCV 2018.
Rao et al. Spherical Fractal Convolution Neural Networks for Point Cloud Recognition. CVPR 2019.

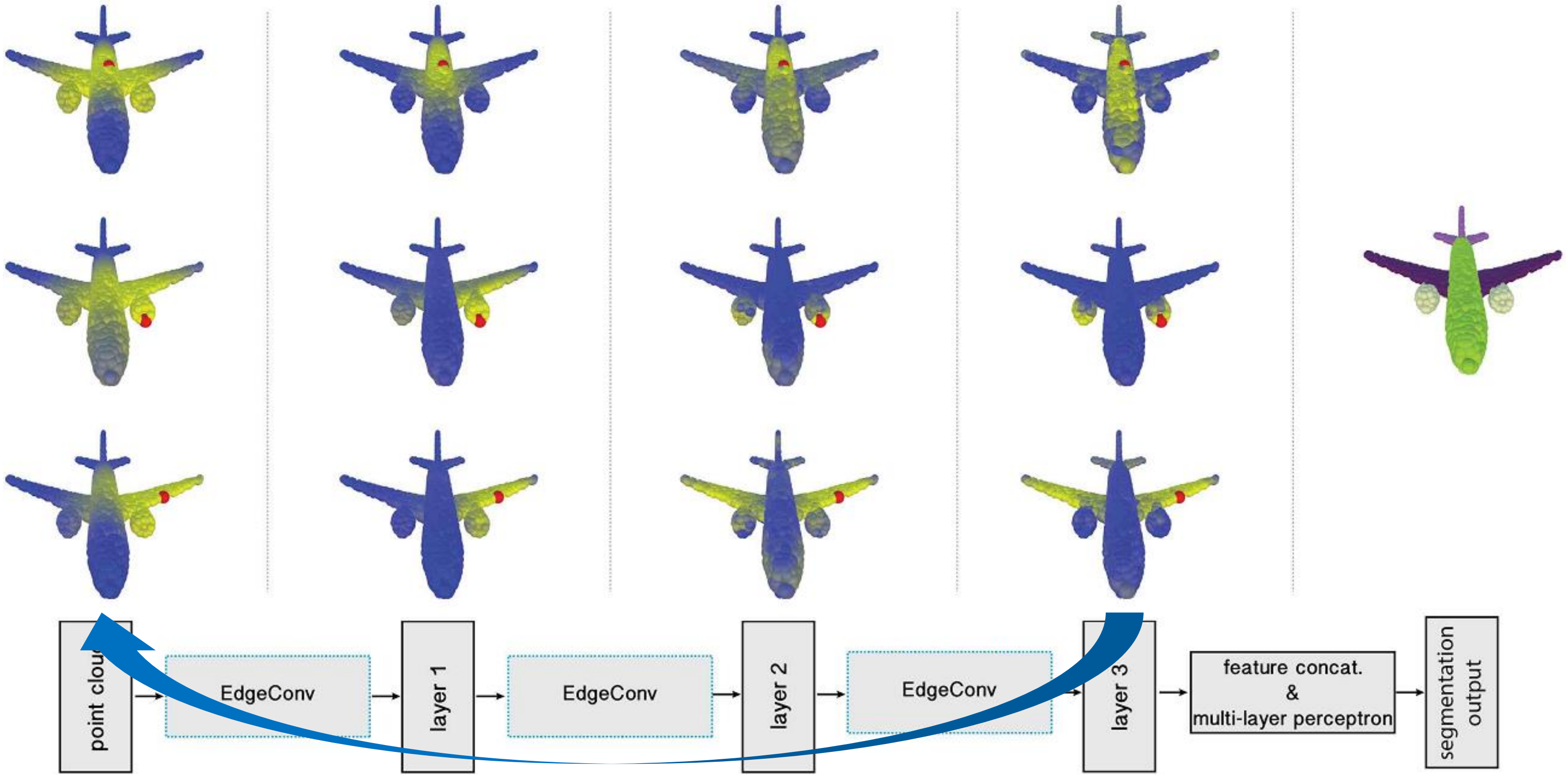
Related work – relation modeling

Related Work <u>*DGCNN*</u>

Dynamic Graph CNN (DGCNN)

Points in high-level feature space captures semantically similar structures.

Despite a large distance between them in the original 3D space.



Wang et al. Dynamic Graph CNN for Learning on Point Clouds. 2018.

Related Work *DGCNN*

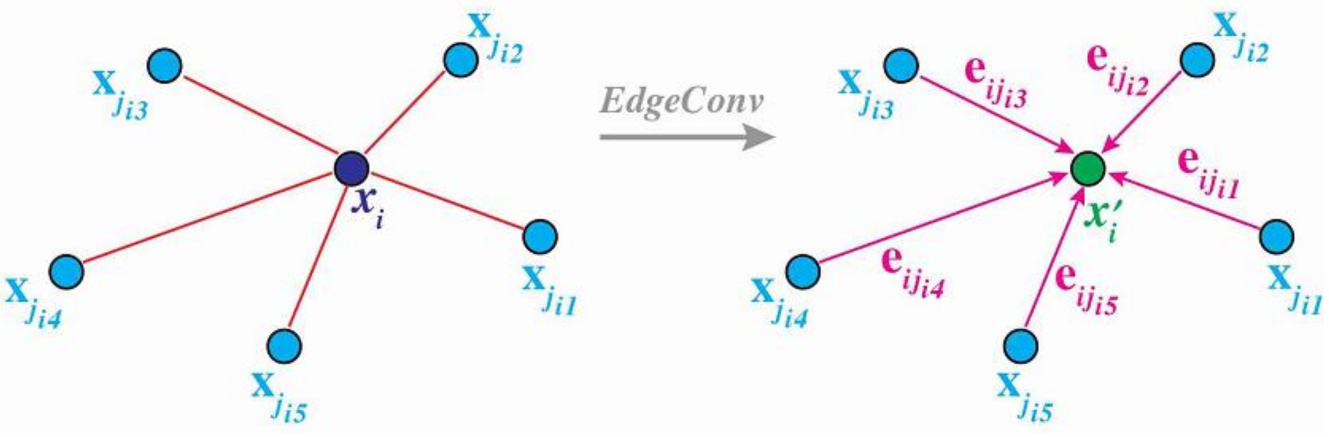
global info. local info.

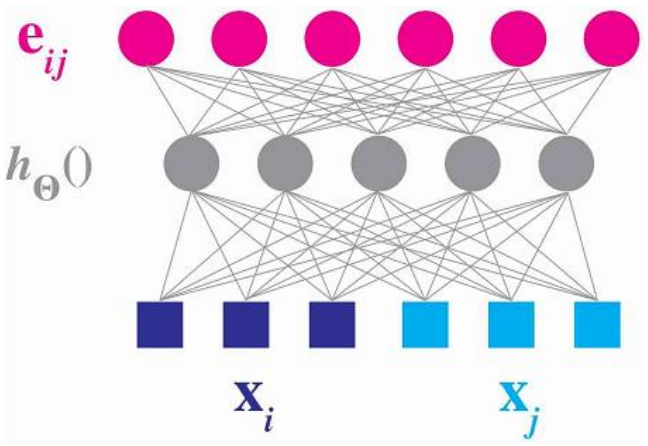
$$h_\Theta(\boxed{x_i}, \boxed{x_j - x_i})$$

$$x_i' = \mathop{\square}\limits^{\text{max}}_{j:(i,j)\in\mathcal{E}} h_\Theta(x_i, x_j).$$

- Neighbors are found in feature space
- Learn from semantically similar structures

DGCNN —— EdgeConv





Wang et al. Dynamic Graph CNN for Learning on Point Clouds. 2018.

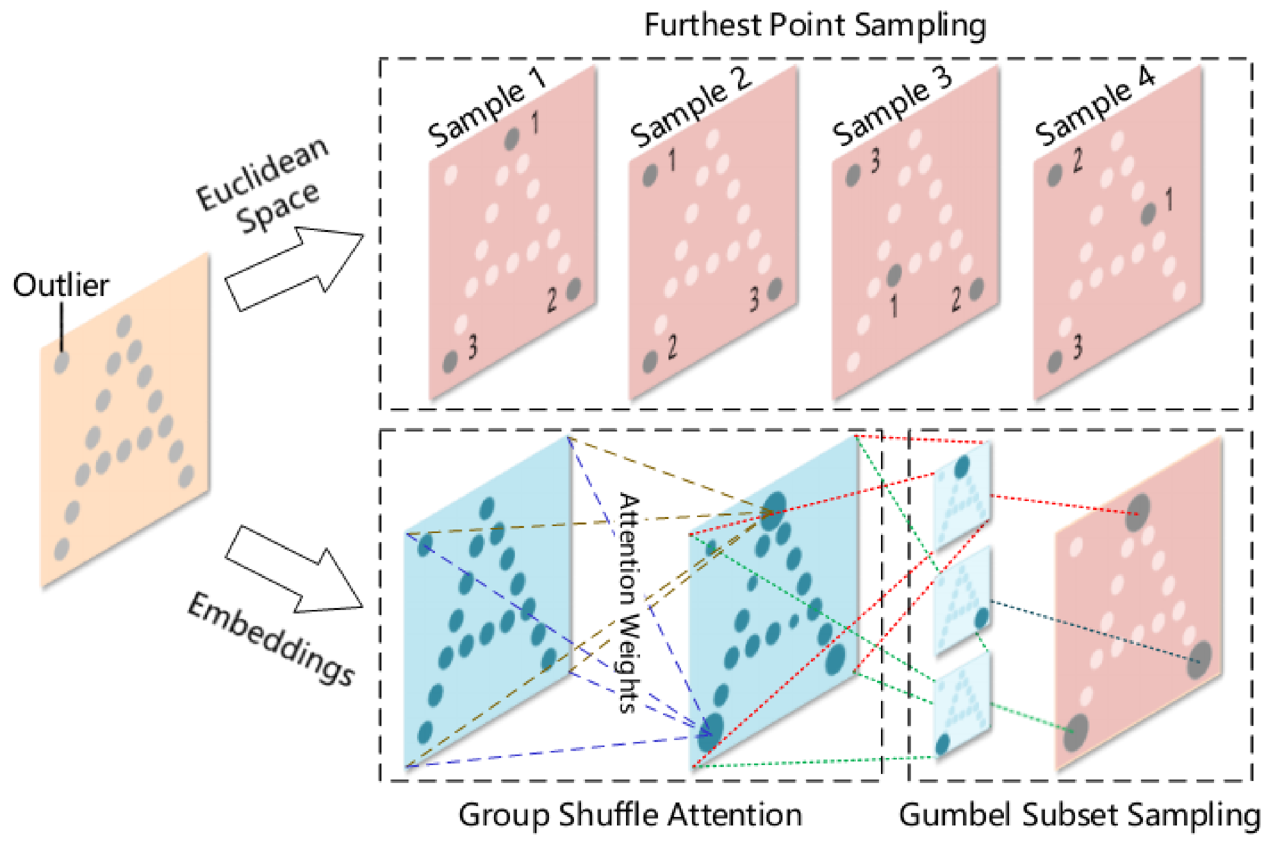
Related Work *self-attention*

- Relation modeling: self-attention
- Gumbel Subset Sampling VS. Farthest Point Sampling

  — permutation-invariant

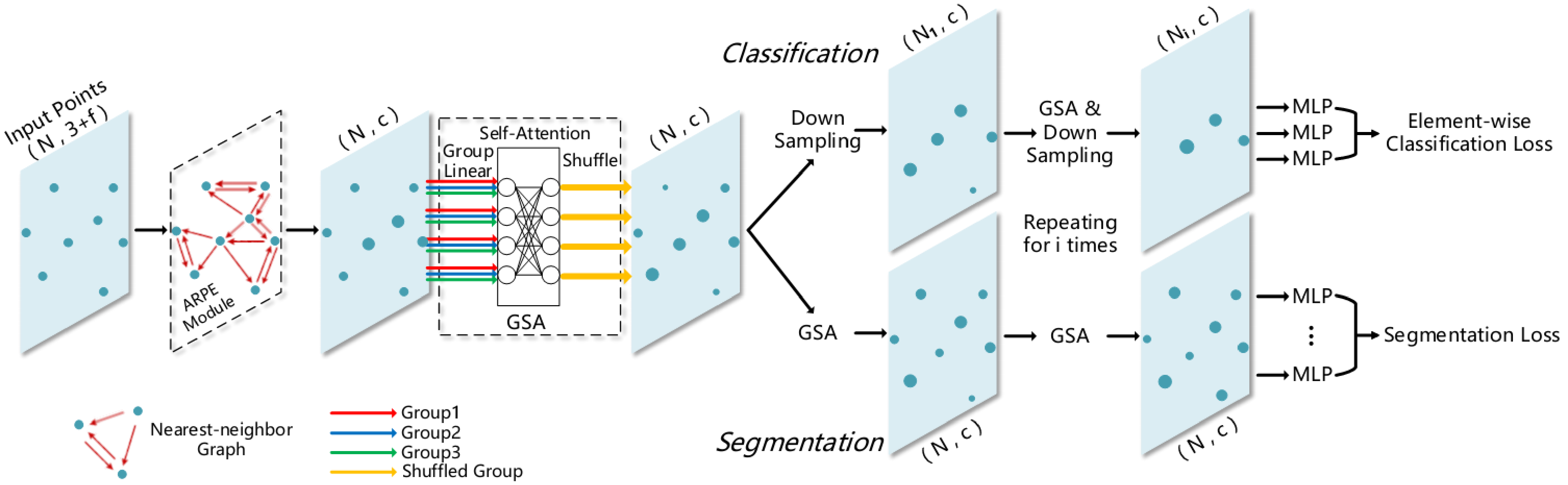
  — high-dimension embedding space

  — differentiable



Yang et al. Modeling Point Clouds with Self-Attention and Gumbel Subset Sampling. CVPR 2019.

Related Work <u>*self-attention*</u>



Embedding：PointNet

$$X_p' = \{(x_p, x_i - x_p)\,|\, i \neq p\}.$$

Self-attention：

group convolution + channel shuffle + pre-activation

Yang et al. Modeling Point Clouds with Self-Attention and Gumbel Subset Sampling. CVPR 2019.

Related Work *self-attention*

$$X_i \in \mathbb{R}^{N_i \times c}$$

$$X_{i+1} \in \mathbb{R}^{N_{i+1} \times c} \subseteq X_i$$

Gumbel Subset Sampling:

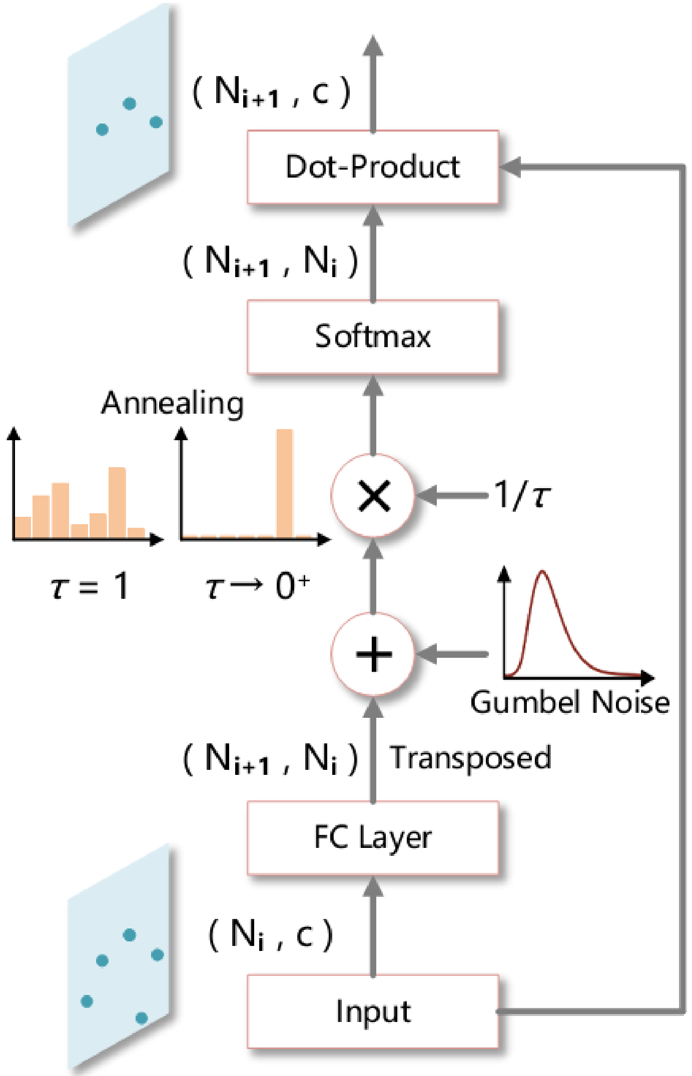
$$y = softmax(wX_i^T) \cdot X_i, \quad w \in \mathbb{R}^c.$$

discrete reparameterization trick

$$y_{gumbel} = gumbel\_softmax(wX_i^T) \cdot X_i, \quad w \in \mathbb{R}^c.$$

multiple point version

$$GSS(X_i) = gumbel\_softmax(WX_i^T) \cdot X_i, \quad W \in \mathbb{R}^{N_{i+1} \times c}.$$



Maximilian et al. Attention-based deep multiple instance learning. ICML 2018.
Yang et al. Modeling Point Clouds with Self-Attention and Gumbel Subset Sampling. CVPR 2019.

Related work – convolution on point cloud

Related Work <u>*Kernel Point Convolution*</u>

$$(\mathcal{F} * g)(x) = \sum_{x_i \in \mathcal{N}_x} g(x_i - x) f_i$$

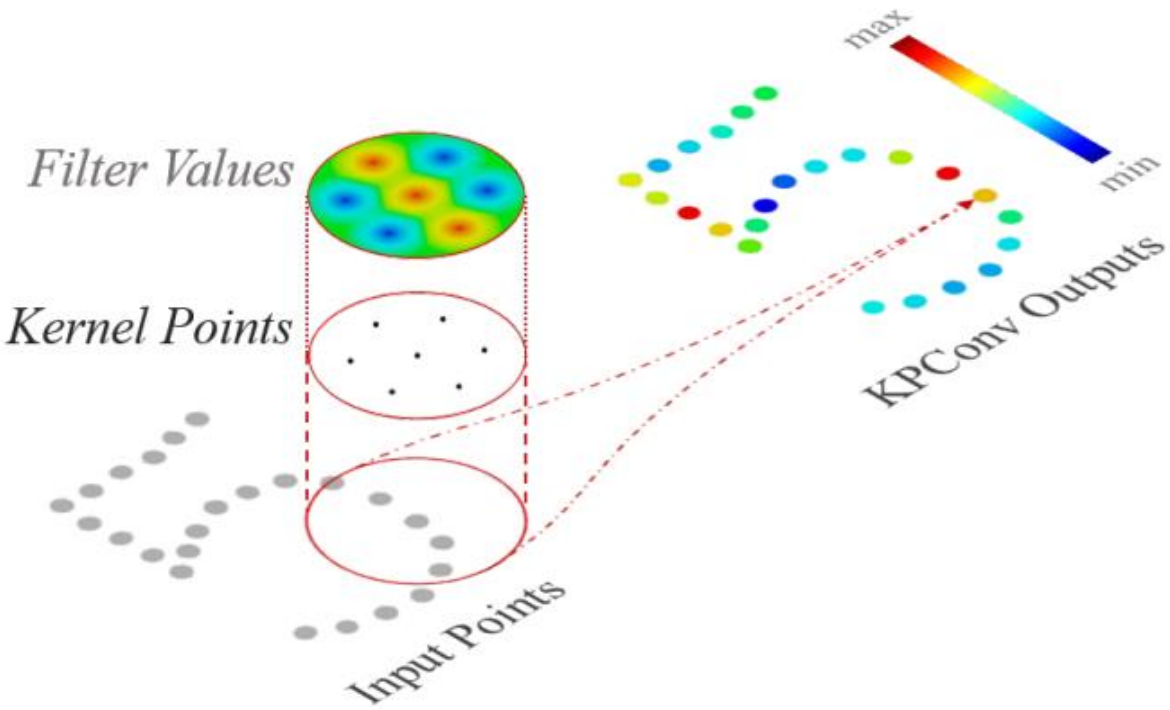
$$\begin{aligned} y_i &= x_i - x \\ \mathcal{B}_r^3 &= \left\{ y \in \mathbb{R}^3 \mid \|y\| \leqslant r \right\} \end{aligned}$$

$$g(y_i) = \sum_{k<K} h\left(y_i, \widetilde{x}_k\right) W_k$$

kernel points: $\{\widetilde{x}_k \mid k < K\} \subset \mathcal{B}_r^3$

$\{W_k \mid k < K\} \subset \mathbb{R}^{D_{in} \times D_{out}}$

$$h\left(y_i, \widetilde{x}_k\right) = \max\left(0, 1 - \frac{\|y_i - \widetilde{x}_k\|}{\sigma}\right)$$



Hugues et al. KPConv: Flexible and Deformable Convolution for Point Clouds. arXiv 2019.

Related Work *Kernel Point Convolution*

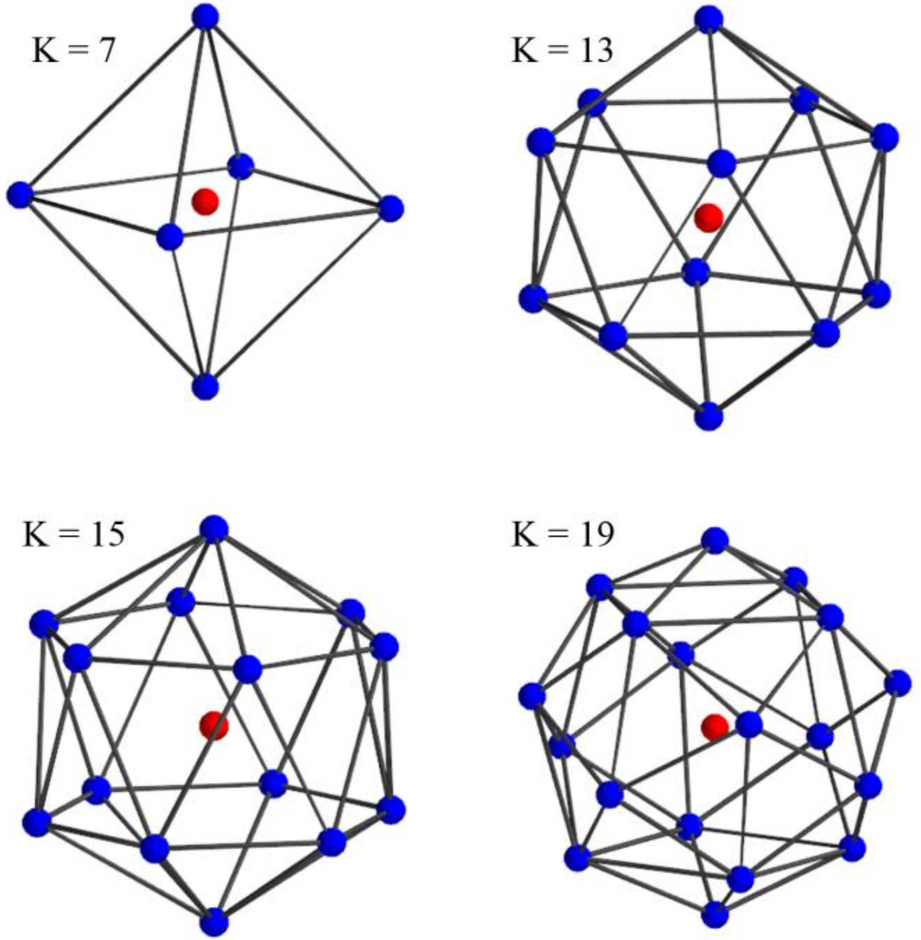
repulsive potential:

$$\forall x \in \mathbb{R}^3, \quad E_k^{rep}(x) = \frac{1}{\|x - \widetilde{x}_k\|}$$

attractive potential:

$$\forall x \in \mathbb{R}^3, \quad E^{att}(x) = \|x\|^2$$

$$E^{tot} = \sum_{k<K} \left(E^{att}(\widetilde{x}_k) + \sum_{l \neq k} E_k^{rep}(\widetilde{x}_l) \right)$$



Hugues et al. KPConv: Flexible and Deformable Convolution for Point Clouds. arXiv 2019.

Related Work *Geometric Deep Learning*

Bronstein et al. Geometric deep learning: going beyond euclidean data. IEEE SPM, 2017.

Li et al. Supervised Fitting of Geometric Primitives to 3D Point Clouds. CVPR 2019.

Lan et al. Modeling Local Geometric Structure of 3D Point Clouds using Geo-CNN. CVPR 2019.

He et al. GeoNet: Deep Geodesic Networks for Point Cloud Analysis. CVPR 2019.

http://geometricdeeplearning.com/

Github: awesome-point-cloud-analysis

awesome-point-cloud-analysis 🕶 awesome

for anyone who wants to do research about 3D point cloud.

If you find the awesome paper/code/dataset or have some suggestions, please contact linhua2017@ia.ac.cn. Thanks for your valuable contribution to the research community 😃

```
- Recent papers (from 2017)
```

Keywords

`dat.` : dataset | `cls.` : classification | `rel.` : retrieval | `seg.` : segmentation
`det.` : detection | `tra.` : tracking | `pos.` : pose | `dep.` : depth
`reg.` : registration | `rec.` : reconstruction | `aut.` : autonomous driving
`oth.` : other, including normal-related, correspondence, mapping, matching, alignment, compression, generative model...

Statistics: 🔥 code is available & stars >= 100 | ⭐ citation >= 50

2017

- [CVPR] PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. [tensorflow][pytorch] [`cls.` `seg.` `det.`] 🔥 ⭐
- [CVPR] Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs. [`cls.`] ⭐
- [CVPR] SyncSpecCNN: Synchronized Spectral CNN for 3D Shape Segmentation. [torch] [`seg.` `oth.`] ⭐
- [CVPR] ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. [project][git] [`dat.` `cls.` `rel.` `seg.` `oth.`] 🔥 ⭐

Relation-Shape Convolutional Neural Network for Point Cloud Analysis

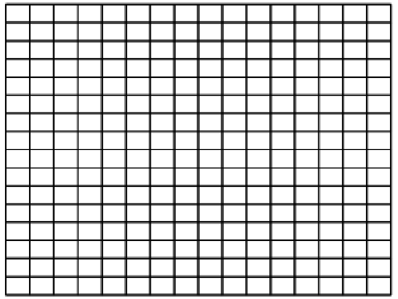
Yongcheng Liu, Bin Fan, Shiming Xiang, Chunhong Pan

CVPR 2019 **Oral Presentation**

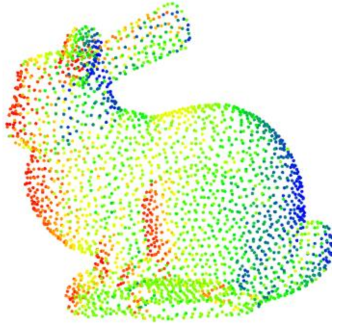
Project Page: https://yochengliu.github.io/Relation-Shape-CNN/

RS-CNN *Motivation*

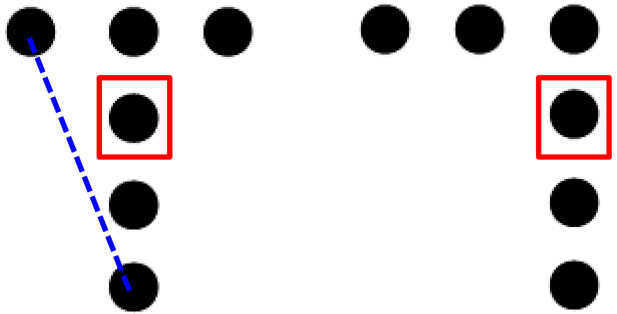
2D image



3D point cloud



3D **Shape** Learning



Relation Learning

Deep Learning (CNN)

RS-CNN *Method: Relation-Shape Conv*

local point subset $P_{\text{sub}} \subset \mathbb{R}^3$ → spherical neighborhood: $x_i + x_j \in \mathcal{N}(x_i)$

$$\mathbf{f}_{P_{\text{sub}}} = \sigma\big(\mathcal{A}(\{\mathcal{T}(\mathbf{f}_{x_j}),\ \forall x_j\})\big)^{1},\ d_{ij} < r\ \forall x_j \in \mathcal{N}(x_i) \qquad y = \sigma(\sum \mathbf{W} * \mathbf{X})$$

$\mathcal{T}$: feature transformation $\mathcal{A}$: feature aggregation

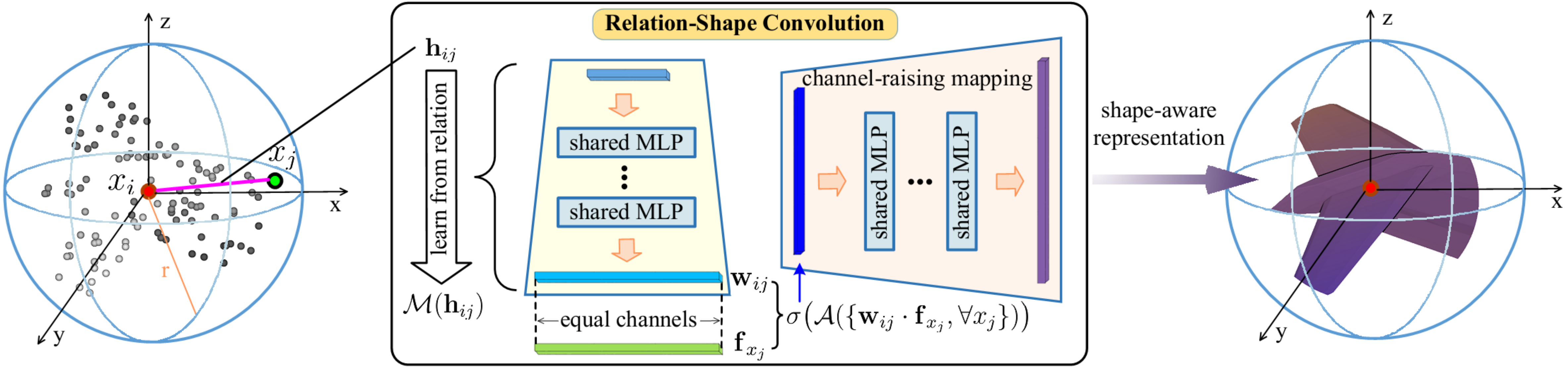
- Permutation invariance: only when A is symmetric and T is shared over each point
- Limitations of CNN: weight is not shared
  gradient only w.r.t single point - implicit $\mathcal{T}(\mathbf{f}_{x_j}) = \mathbf{w}_j \cdot \mathbf{f}_{x_j}$
- Conversion: learn from relation $\mathcal{T}(\mathbf{f}_{x_j}) = \mathbf{w}_{ij} \cdot \mathbf{f}_{x_j} = \mathcal{M}(\mathbf{h}_{ij}) \cdot \mathbf{f}_{x_j}$

$\mathbf{h}_{ij}$: predefined geometric priors → low-level relation

$$\mathbf{f}_{P_{\text{sub}}} = \sigma\big(\mathcal{A}(\{\mathcal{M}(\mathbf{h}_{ij}) \cdot \mathbf{f}_{x_j},\ \forall x_j\})\big)$$

$\mathcal{M}$: mapping function(shared MLP) → high-level relation

RS-CNN *Method*



high-level relation encoding + channel raising mapping

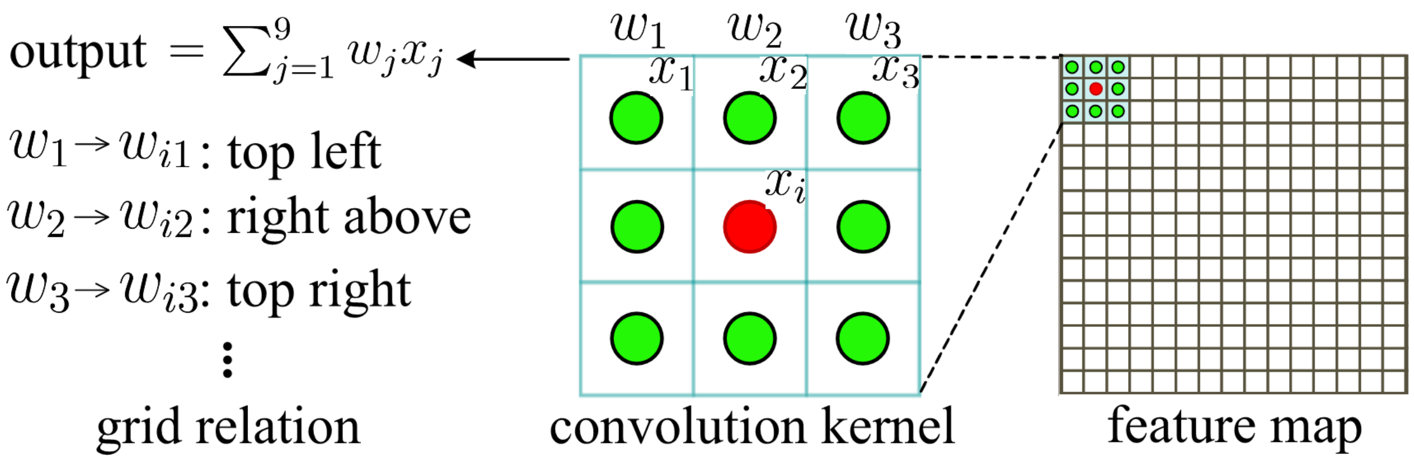
low-level relation $\mathbf{h}_{ij}$: (3D Euclidean distance, $x_i - x_j$, x_i, x_j) 10 channels

RS-CNN *RS-Conv: Properties*

$$\mathbf{f}_{P_{\text{sub}}} = \sigma\Big(\mathcal{A}(\{\mathcal{M}(\mathbf{h}_{ij}) \cdot \mathbf{f}_{x_j}, \forall x_j\})\Big)$$

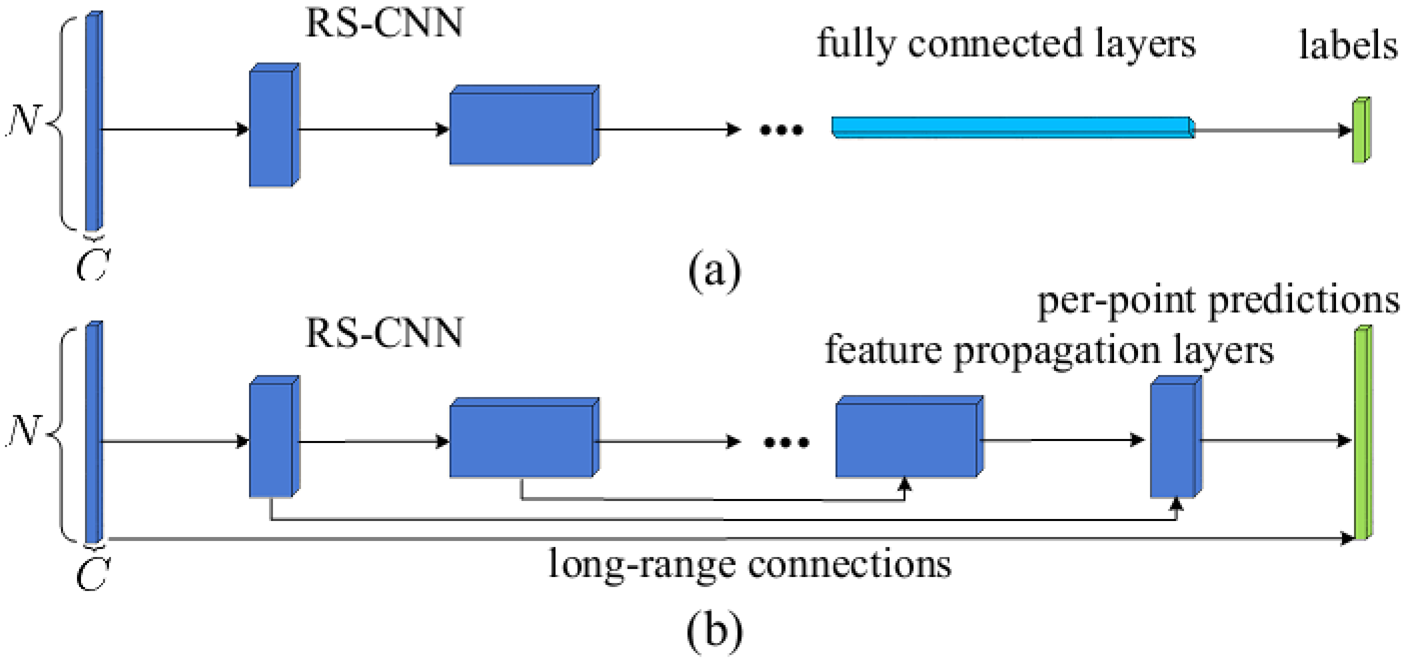
- ✓ Permutation invariance
- ✓ Robustness to rigid transformation in Relation Learning, e.g., 3D Euclidean distance
- ✓ Points' interaction
- ✓ Weight sharing

Revisiting 2D Conv:



RS-Conv with relation learning is more general and can be applied to model 2D grid spatial relationship.

RS-CNN *RS-CNN*

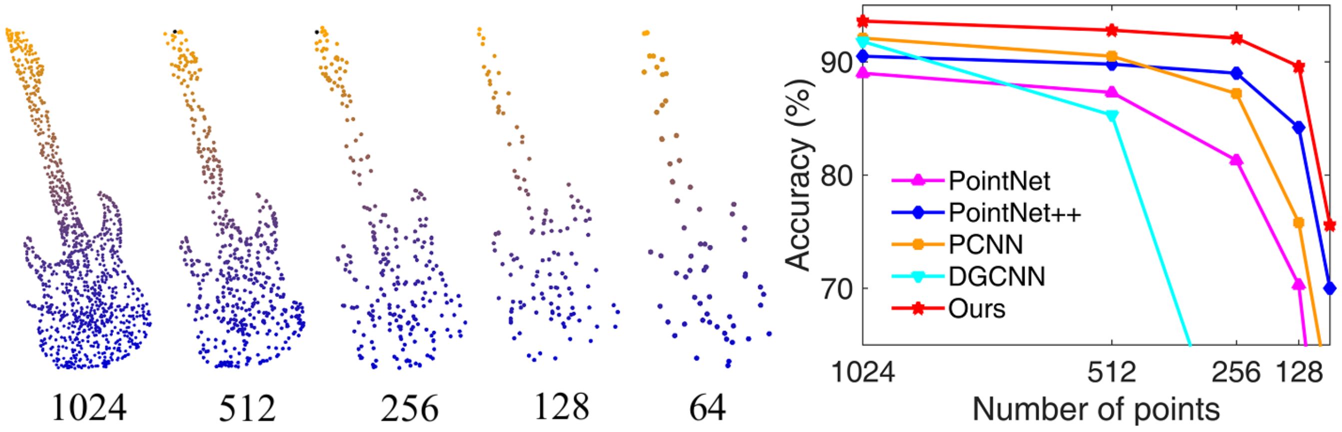


Farthest Point Sampling + Sphere Neighborhood + RS-Conv

RS-CNN *Shape classification*

ModelNet40 benchmark

Robustness to sampling density



method	input	#points	acc.
Pointwise-CNN [10]	xyz	1k	86.1
Deep Sets [48]	xyz	1k	87.1
ECC [31]	xyz	1k	87.4
PointNet [24]	xyz	1k	89.2
SCN [44]	xyz	1k	90.0
Kd-Net(depth=10) [16]	xyz	1k	90.6
PointNet++ [26]	xyz	1k	90.7
KCNet [30]	xyz	1k	91.0
MRTNet [3]	xyz	1k	91.2
Spec-GCN [38]	xyz	1k	91.5
PointCNN [21]	xyz	1k	91.7
DGCNN [41]	xyz	1k	92.2
PCNN [1]	xyz	1k	92.3
Ours	**xyz**	**1k**	**93.6**
SO-Net [19]	xyz	2k	90.9
Kd-Net(depth=15) [16]	xyz	32k	91.8
O-CNN [39]	xyz, nor	-	90.6
Spec-GCN [38]	xyz, nor	1k	91.8
PointNet++ [26]	xyz, nor	5k	91.9
SpiderCNN [45]	xyz, nor	5k	92.4
SO-Net [19]	xyz, nor	5k	93.4

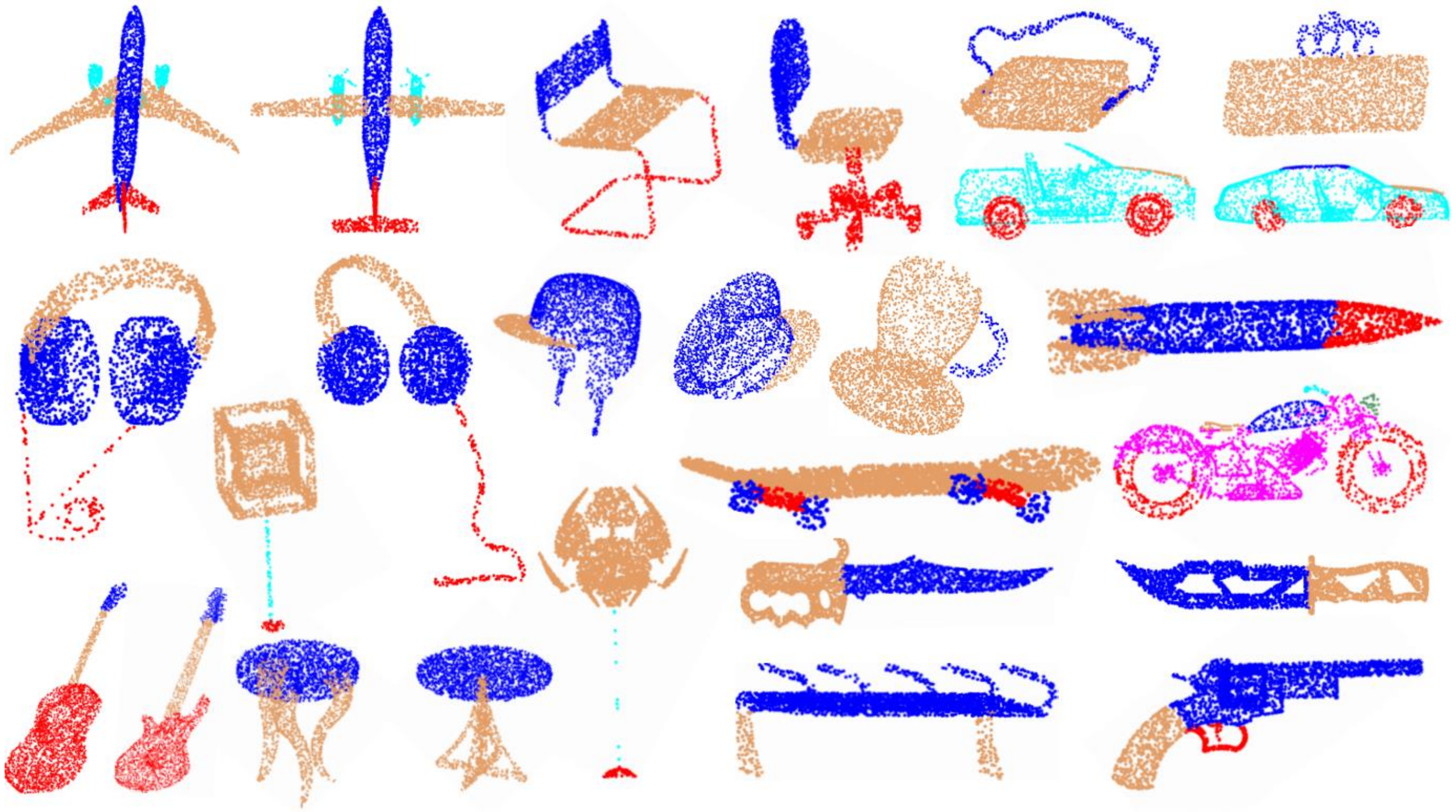
RS-CNN *ShapePart Segmentation*

method	input	class mIoU	instance mIoU	air plane	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor bike	mug	pistol	rocket	skate board	table
Kd-Net [16]	4k	77.4	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
PointNet [24]	2k	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
RS-Net [11]	-	81.4	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
SCN [44]	1k	81.8	84.6	83.8	80.8	83.5	79.3	90.5	69.8	**91.7**	86.5	82.9	96.0	69.2	93.8	82.5	**62.9**	74.4	80.8
PCNN [1]	2k	81.8	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
SPLATNet [34]	-	82.0	84.6	81.9	83.9	88.6	79.5	90.1	73.5	91.3	84.7	84.5	**96.3**	69.7	**95.0**	81.7	59.2	70.4	81.3
KCNet [30]	2k	82.2	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
DGCNN [41]	2k	82.3	85.1	**84.2**	83.7	84.4	77.1	90.9	78.5	91.5	87.3	82.9	96.0	67.8	93.3	82.6	59.7	75.5	82.0
Ours	**2k**	**84.0**	**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**
PointNet++ [26]	2k,nor	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
SyncCNN [47]	mesh	82.0	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 [19]	1k,nor	80.8	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
SpiderCNN [45]	2k,nor	82.4	85.3	83.5	81.0	87.2	77.5	90.7	76.8	91.1	87.3	83.3	95.8	70.2	93.5	82.7	59.7	75.8	82.8

class mIoU 1.7↑ instance mIoU 1.1↑

Best results over 10 categories

RS-CNN <u>*ShapePart Segmentation*</u>



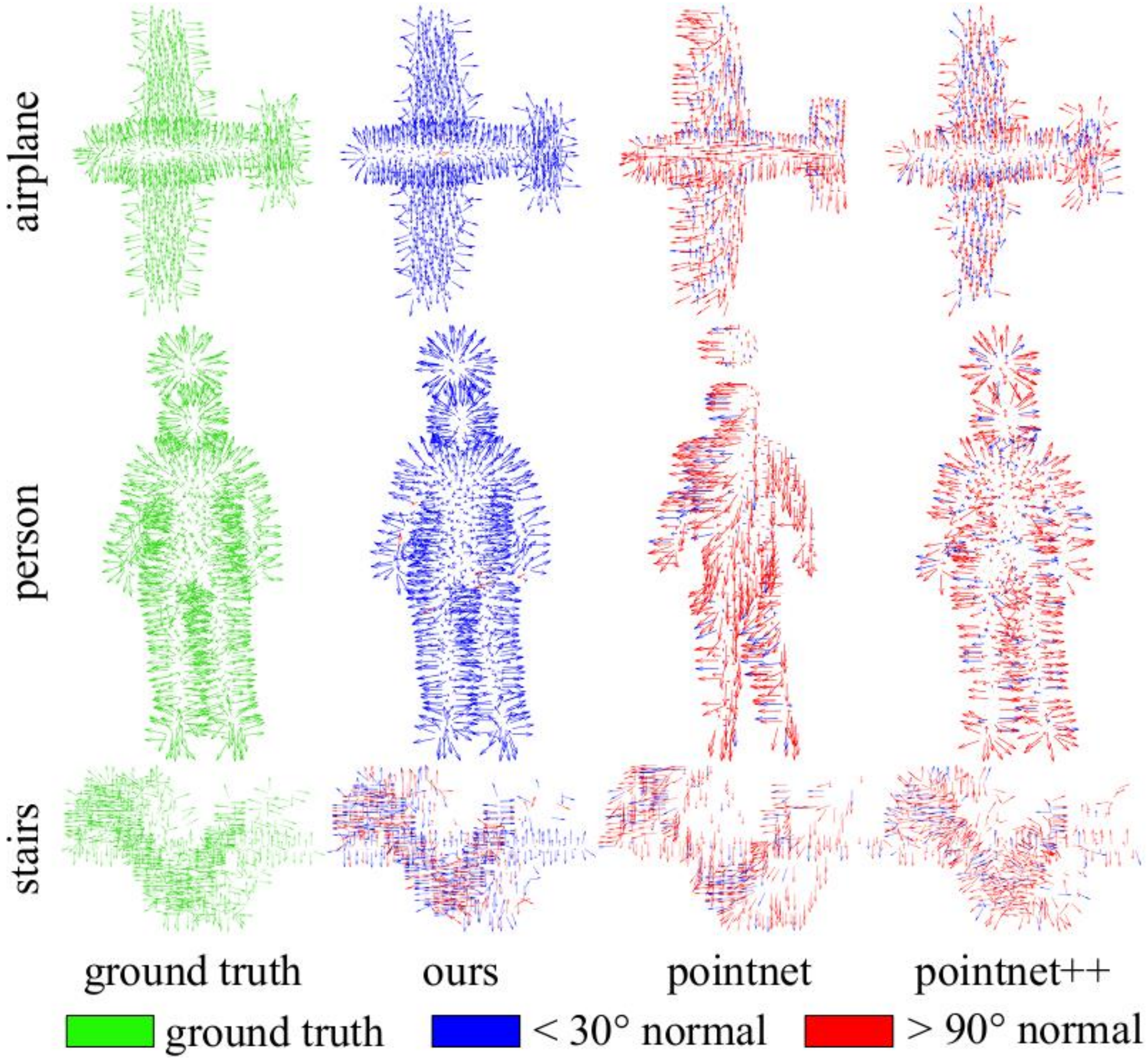
Diverse, confusing shapes

RS-CNN *Normal estimation*

Table 3. Normal estimation error on ModelNet40 dataset.

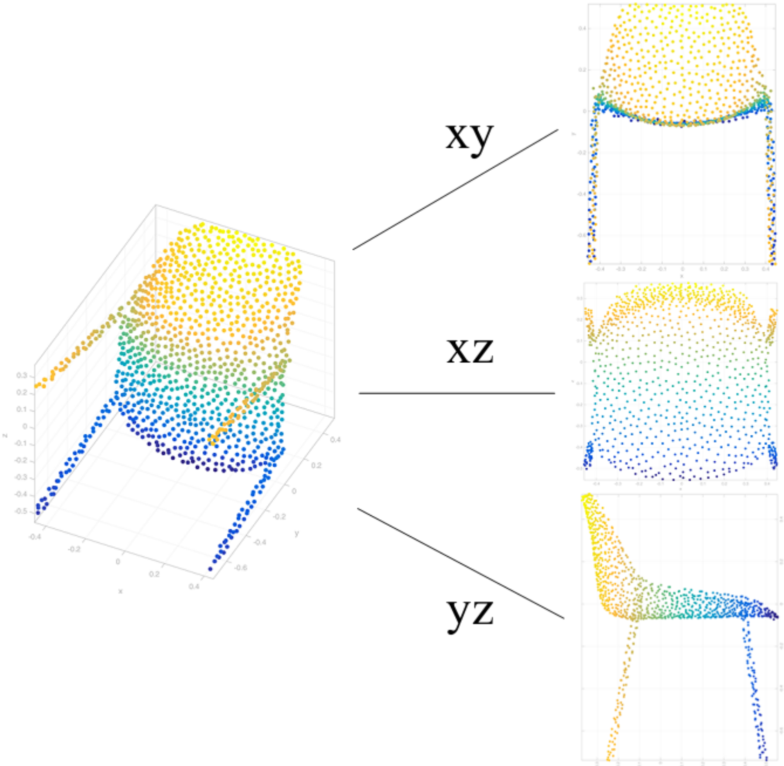
dataset	method	#points	error
ModelNet40	PointNet [1]	1k	0.47
	PointNet++ [1]	1k	0.29
	PCNN [1]	1k	0.19
	Ours	**1k**	**0.15**

less effective for some intractable shapes, such as spiral stairs and intricate plants



RS-CNN *Geometric priors*

$$\mathbf{f}_{P_{\text{sub}}} = \sigma\big(\mathcal{A}(\{\mathcal{M}(\mathbf{h}_{ij}) \cdot \mathbf{f}_{x_j},\ \forall x_j\})\big)$$



model	low-level relation $\mathbf{h}$	channels	acc.
A	(3D-Ed)	1	92.5
B	(3D-Ed, $x_i - x_j$)	4	93.0
C	(3D-Ed, $x_i - x_j$, x_i, x_j)	10	**93.6**
D	(3D-cosd, x_i^{nor}, x_j^{nor})	7	92.8
E	(2D-Ed, $x'_i - x'_j$, x'_i, x'_j)	10	$\approx$ 92.2

low-level relation $\mathbf{h}$	channels	acc.
(XY-Ed, $x_i^{\text{xy}} - x_j^{\text{xy}}$, x_i^{xy}, x_j^{xy})	10	92.1
(XZ-Ed, $x_i^{\text{xz}} - x_j^{\text{xz}}$, x_i^{xz}, x_j^{xz})	10	92.1
(YZ-Ed, $x_i^{\text{yz}} - x_j^{\text{yz}}$, x_i^{yz}, x_j^{yz})	10	92.2
fusion of above three views		92.5

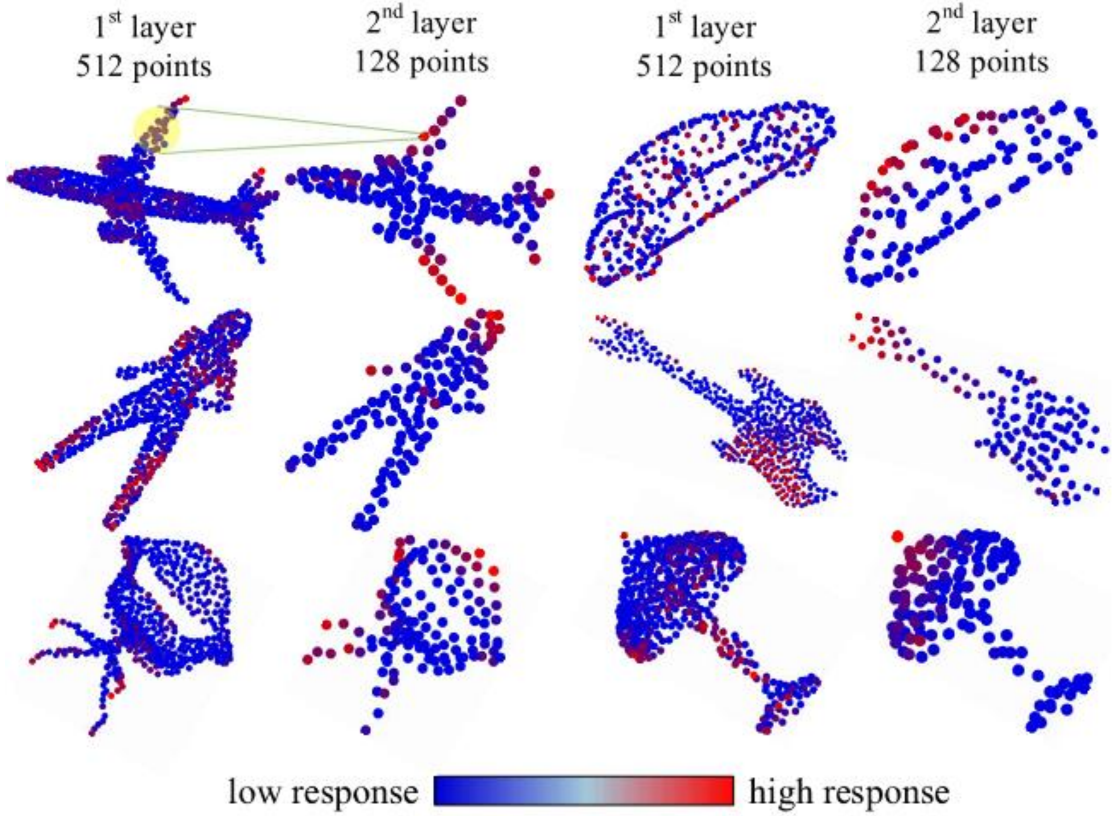
RS-CNN *Model analysis*

Robustness to point permutation and rigid transformation

relation: 3D Euclidean distance

method	acc.	perm.	+0.2	-0.2	90°	180°
PointNet [24]	88.7	88.7	70.8	70.6	42.5	38.6
PointNet++ [26]	88.2†	88.2	88.2	88.2	47.9	39.7
Ours	**90.3†**	**90.3**	**90.3**	**90.3**	**90.3**	**90.3**

$$\mathbf{f}_{P_{\text{sub}}} = \sigma\big(\mathcal{A}(\{\mathcal{M}(\mathbf{h}_{ij}) \cdot \mathbf{f}_{x_j}, \ \forall x_j\})\big)$$



Model complexity

method	#params	#FLOPs/sample
PointNet [24]	3.50M	440M
PointNet++ [21]	1.48M	1684M
PCNN [21]	8.20M	**294M**
Ours	**1.41M**	295M

Thanks for your attention !