University of Chinese Academy of Sciences

深度学习在3D点云处理中的探索

刘永成

中科院自动化所

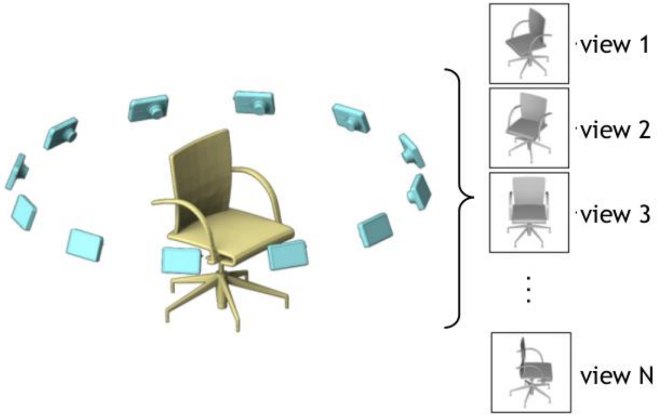
模式识别国家重点实验室

2019.08

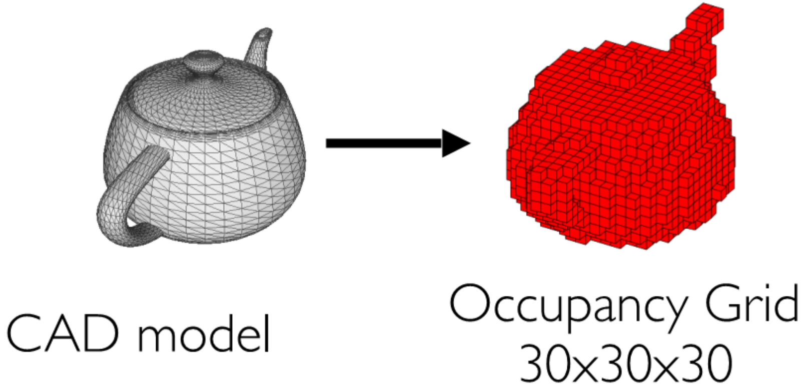
Outline

1. Introduction
2. Brief review
3. RS-CNN & DensePoint
4. Summary & Outlook

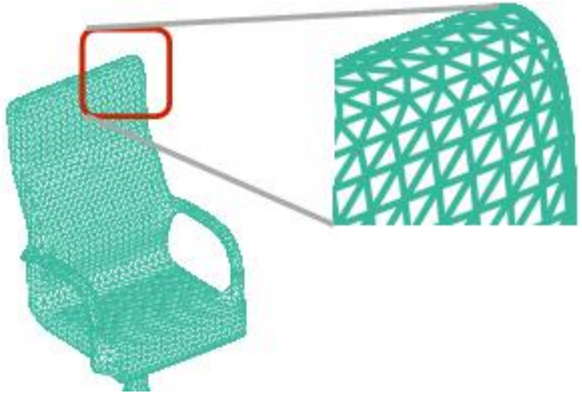
Introduction *3D representations*



multi-view images + 2D CNN



volumetric data + 3D CNN



mesh data + DL (GNN) ?

image depth + CNN

point cloud + DL (GNN & CNN) ?

Introduction *point cloud*

Advantages

- ✓ raw sensor data, e.g., Lidar
- ✓ simple representation: N * (x, y, z, color, normal…)
- ✓ better 3D shape capturing

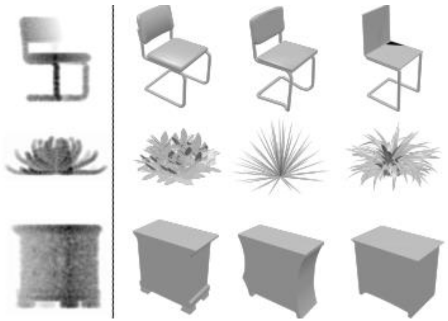
Why emerging?

- ✓ autonomous driving
- ✓ AR & VR
- ✓ robot manipulation
- ✓ Geomatics
- ✓ 3D face & medical
- ✓ AI-assisted shape design in 3D game and animation, etc.
- ✓ open problem, flexible

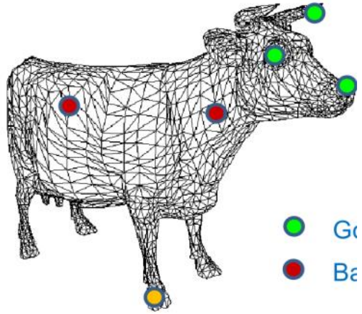
Introduction *tasks*

lamp

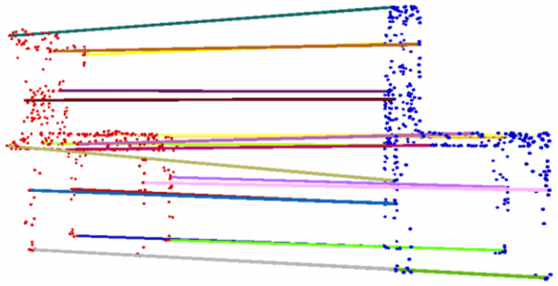
shape classification



shape retrieval

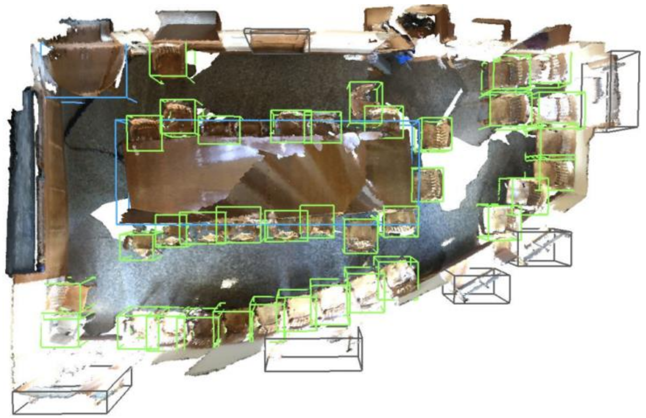


keypoint detection

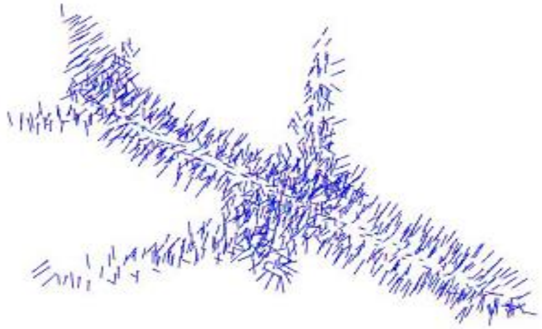


shape correspondence
& registration

semantic segmentation



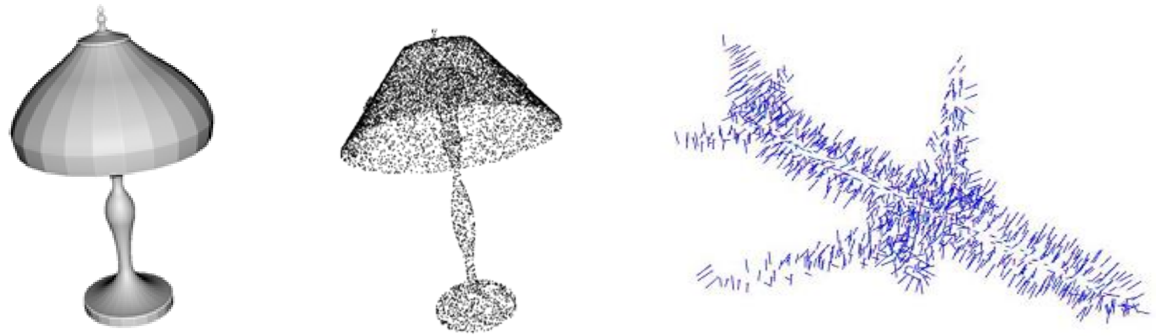
object detection

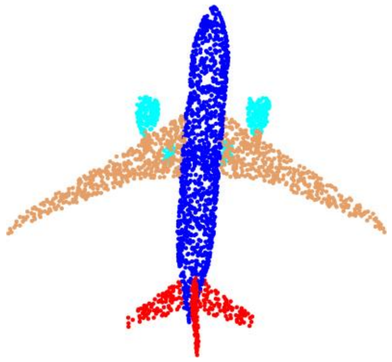


normal estimation

......

Introduction *datasets*



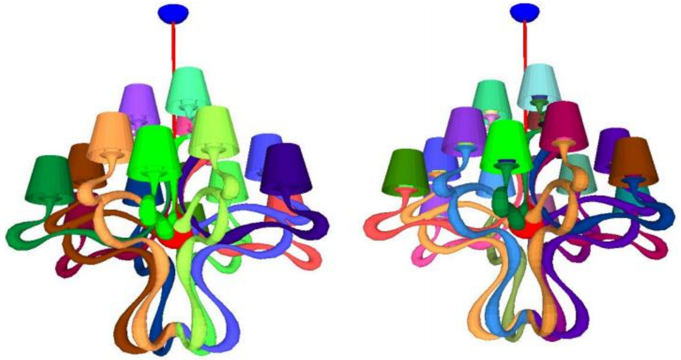


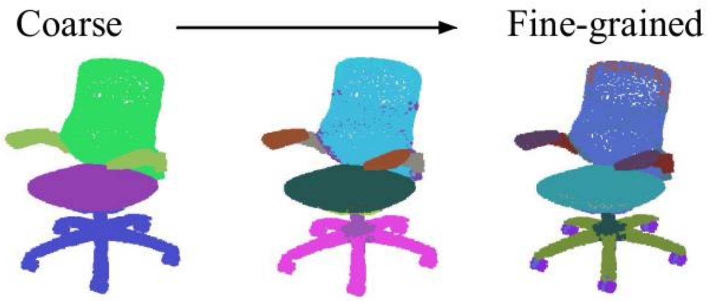
Princeton ModelNet: 1k

[1] Wu et al. CVPR 2015.

ShapeNet Part: 2k

[2] Yi et al. TOG 2016.





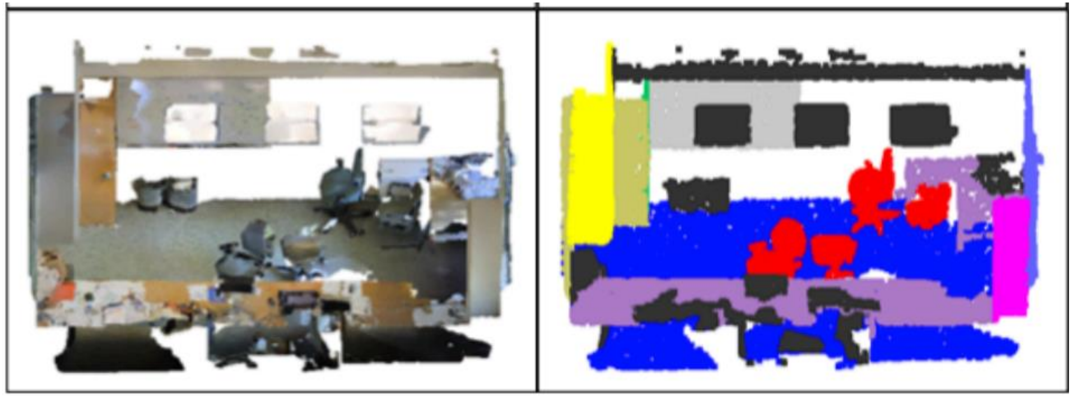
……

PartNet models

Hierarchical Semantic Segmentation

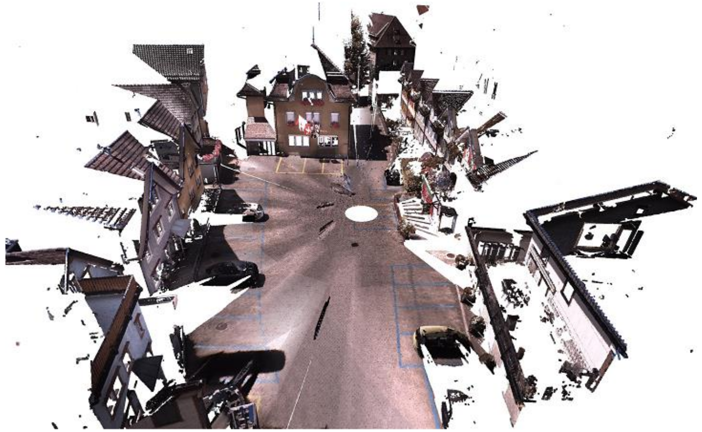
[3] Mo et al. CVPR 2019.

Introduction *datasets*



Stanford 3D indoor scene: 8k

[4] Armeni et al. CVPR 2016.



Semantic 3D: 4 billion in total

[5] Hackel et al. ISPRS 2017.

ScanNet: seg + det

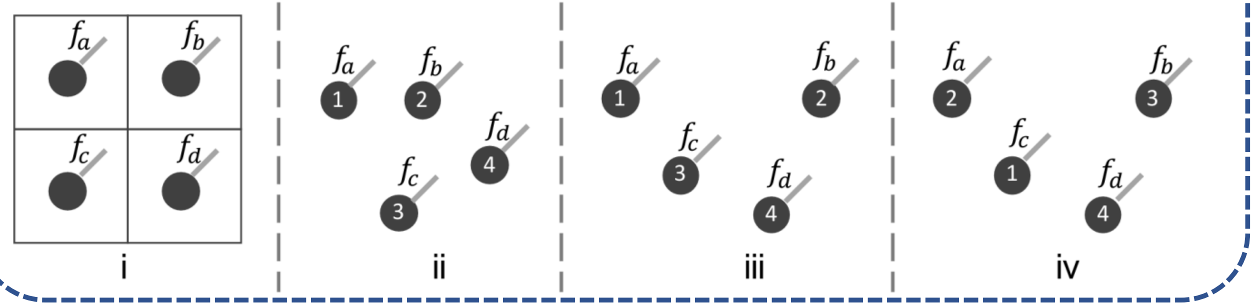
[6] Dai et al. CVPR 2017.

KITTI, nuScenes: det

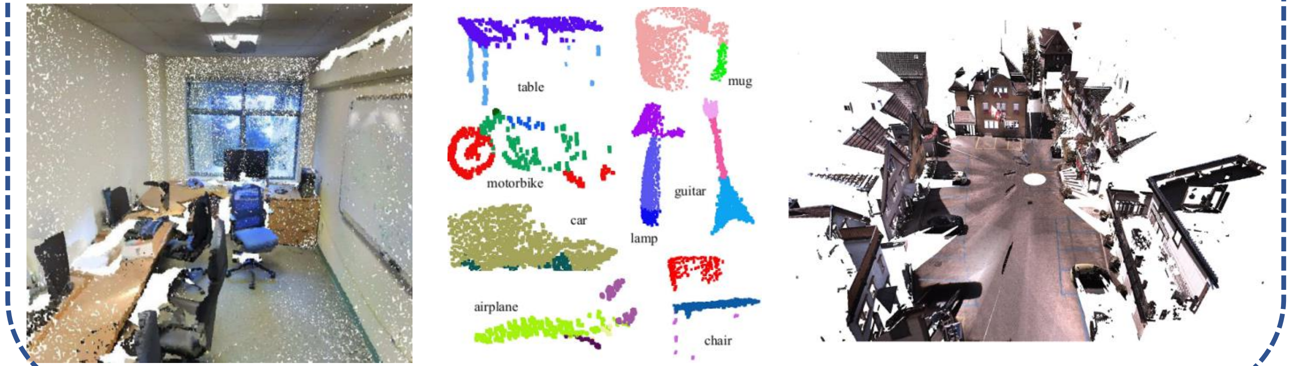
……

Introduction *some challenges*

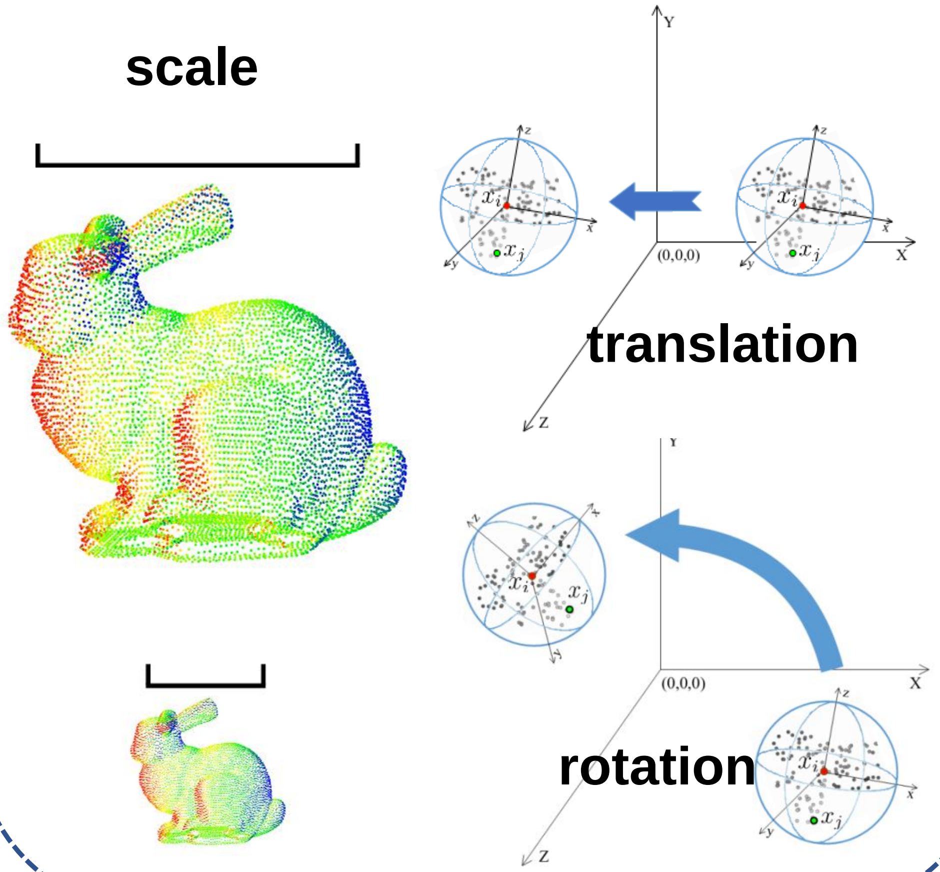
Irregular (unordered): permutation invariance



Robustness to corruption, outlier, noise; partial data; large-scale data

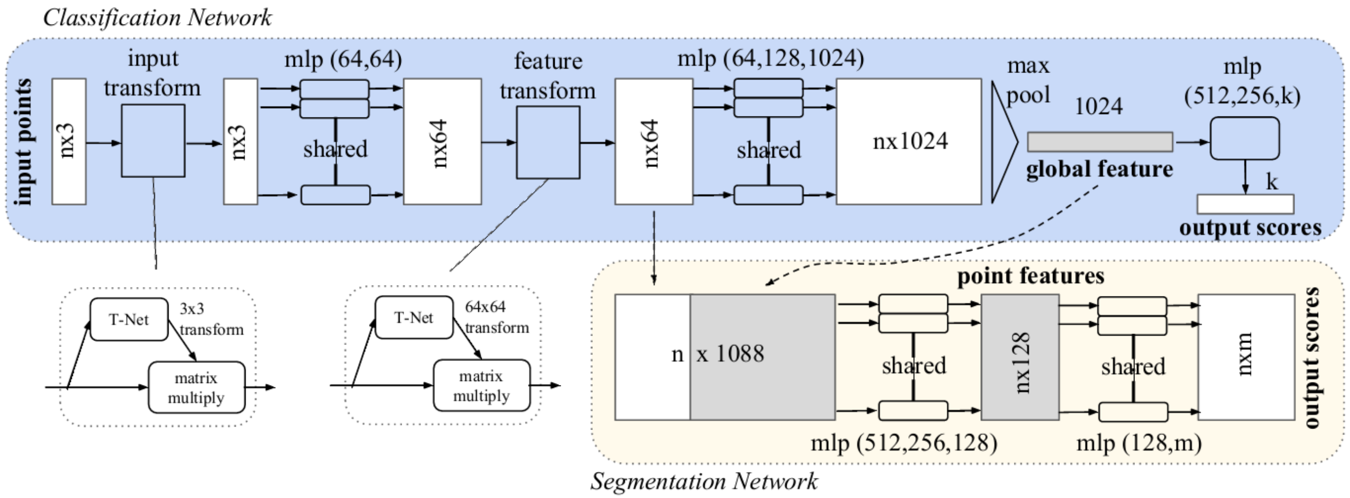


Robustness to rigid transformations



Outline

Related Work *PointNet family*

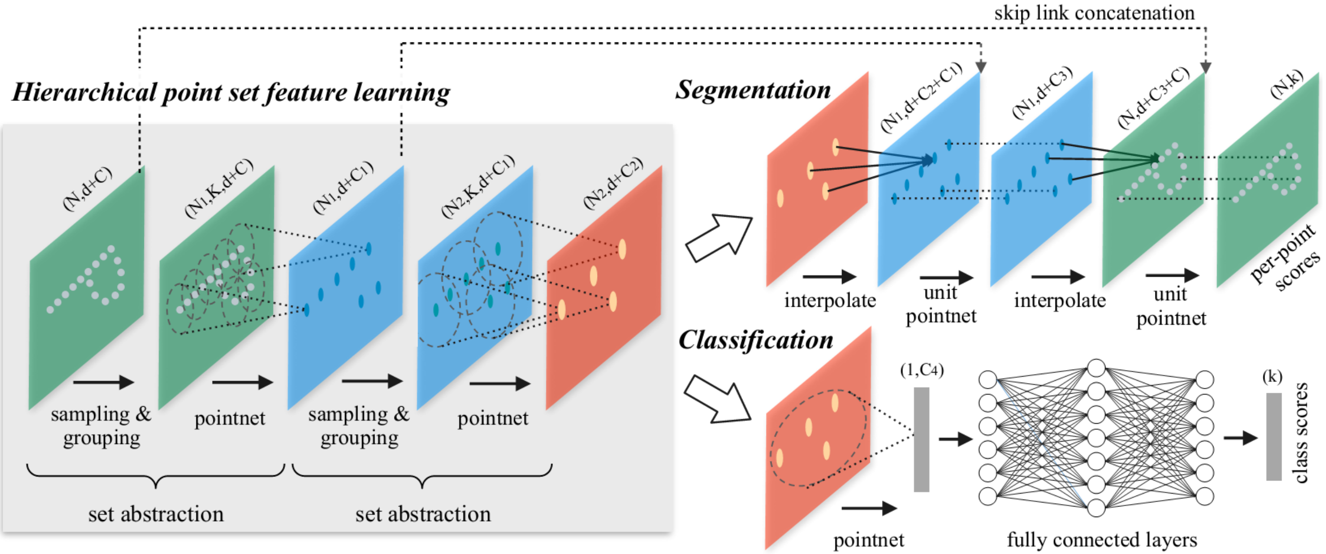


[18] Qi et al. PointNet. CVPR 2017.

Shared MLP

+

max pool (symmetric function)



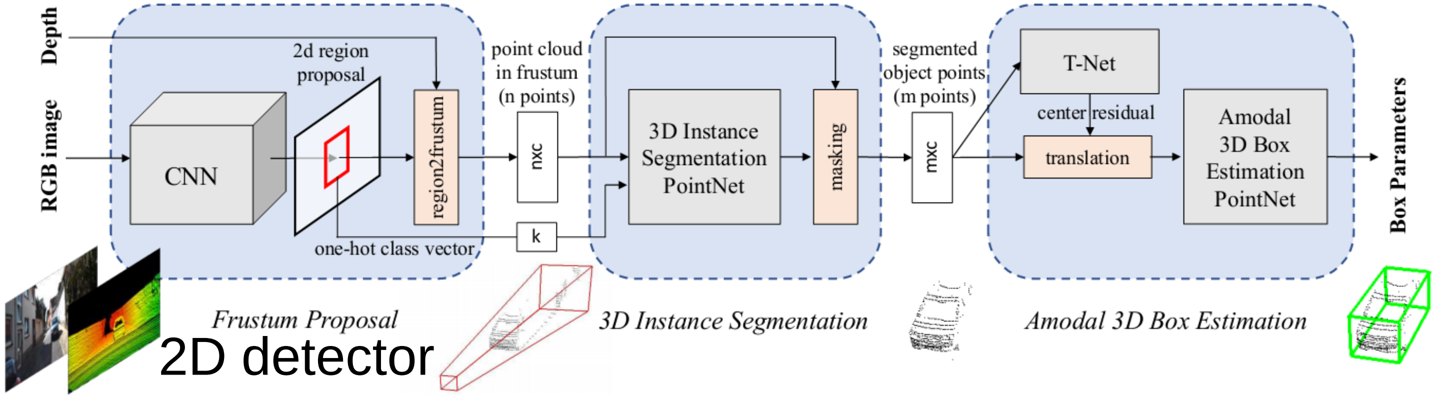
[19] Qi et al. PointNet++. NIPS 2017.

Sampling + Grouping + PointNet

capture local patterns better

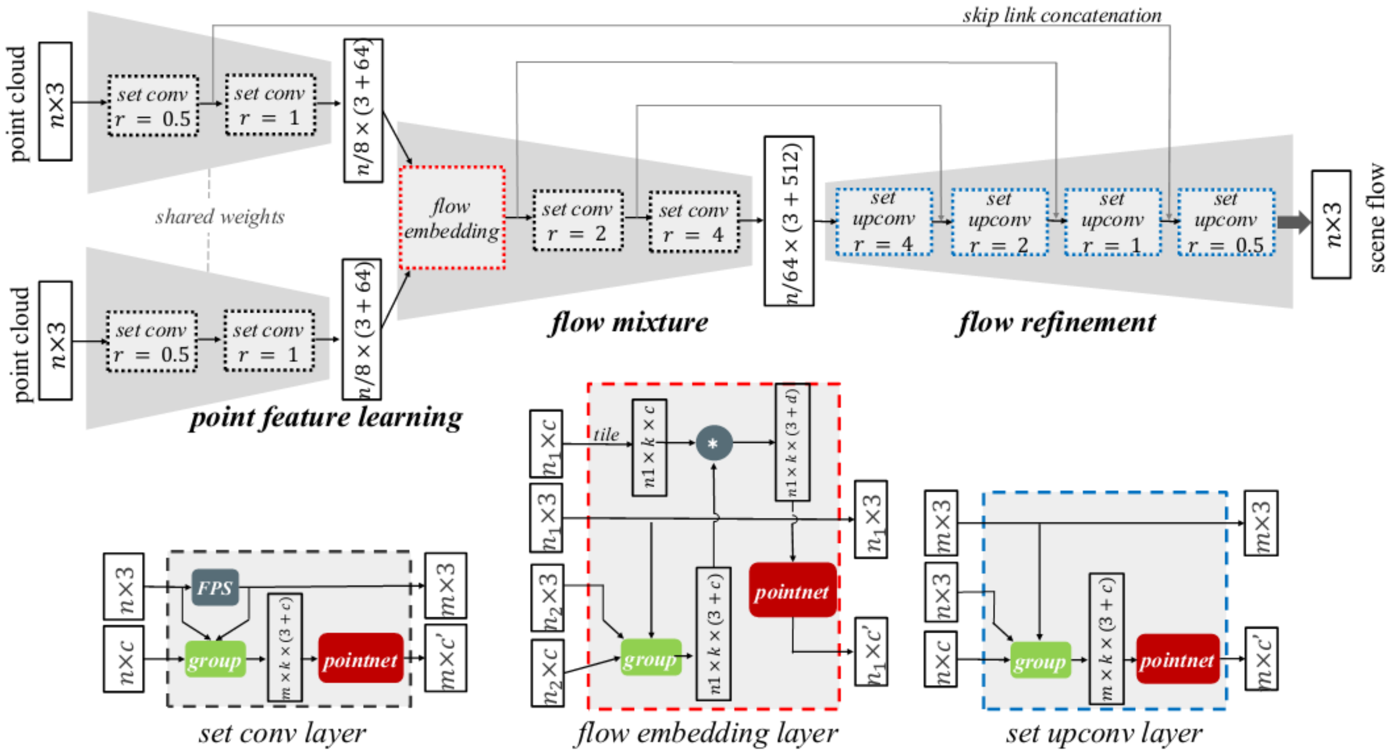
CNN like

Related Work *PointNet family*



2D detector

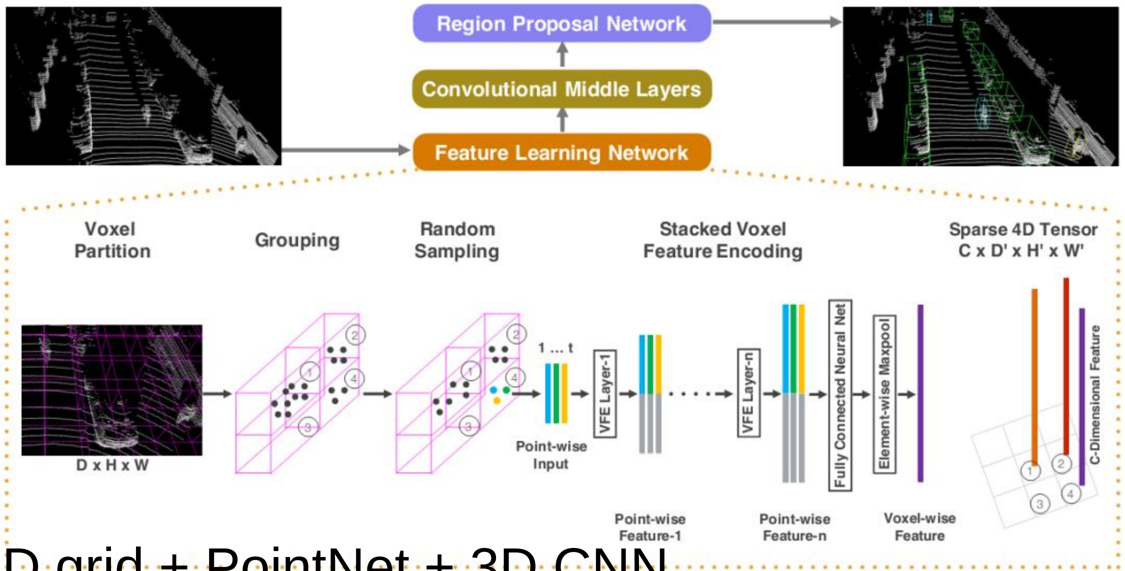
[20] Qi et al. Frustum. CVPR 2018.



[21] Liu et al. FlowNet3D. CVPR 2019.

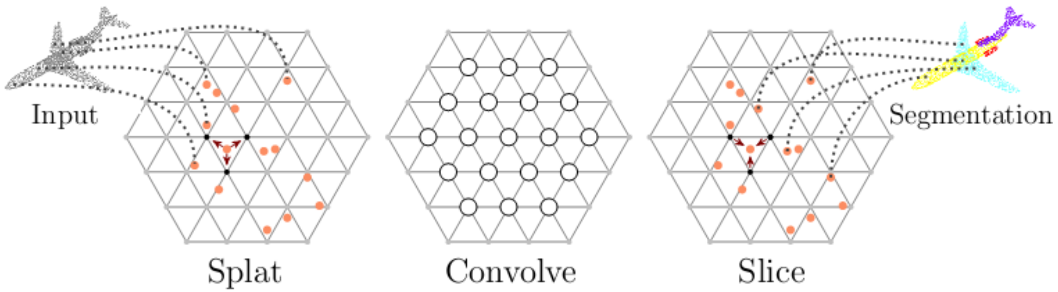
Related Work *regular processing*

[7] Zhou et al. VoxelNet. CVPR 2018.



3D grid + PointNet + 3D CNN

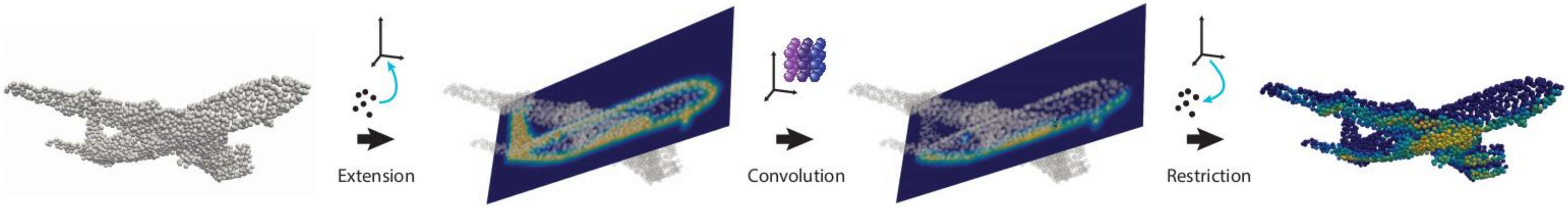
[8] Su et al. SPLATNet. CVPR 2018.



lattice + bilateral convolution + hash index

[9] Kiefel et al. Permutohedral Lattice CNNs. ICLR 2015.

[10] Jampani et al. Bilateral Neural Networks. CVPR 2016.



[11] Atzmon et al. PCNN. SIGGRAPH 2018.

“without any discretization or approximation”

Related Work *regular processing*

[12] Li et al. PointCNN. NIPS 2018.

"simultaneously weight and permute the input features"

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

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

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

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

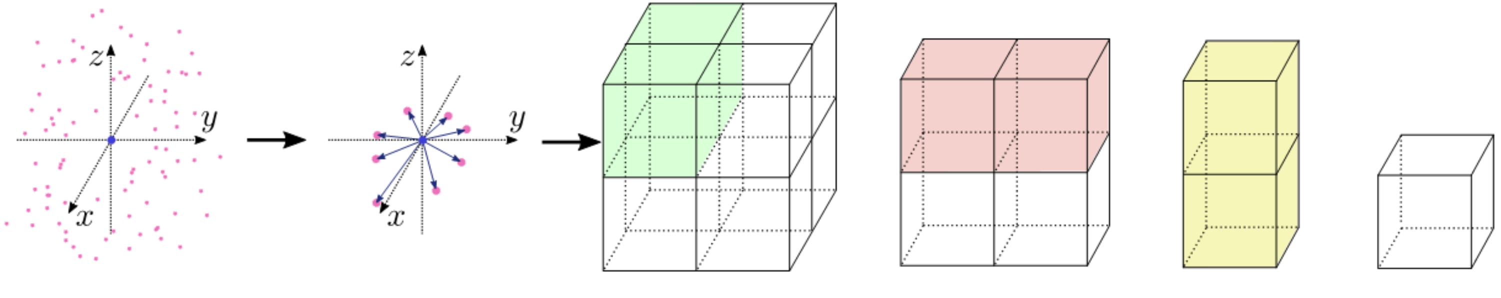
4: $\mathcal{X} \leftarrow MLP(\mathbf{P}')$ ▷ Learn the $K \times K$ $\mathcal{X}$-transformation matrix

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

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

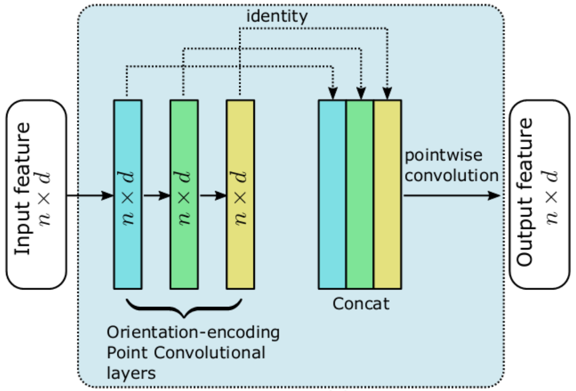
[13] Jiang et al. PointSIFT. arXiv 2018.

orientation-encoding



(a) (b) (c)

Scale-aware

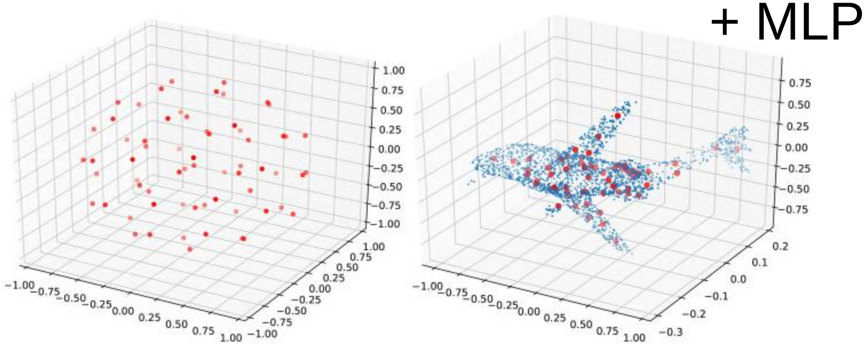


pointSIFT module

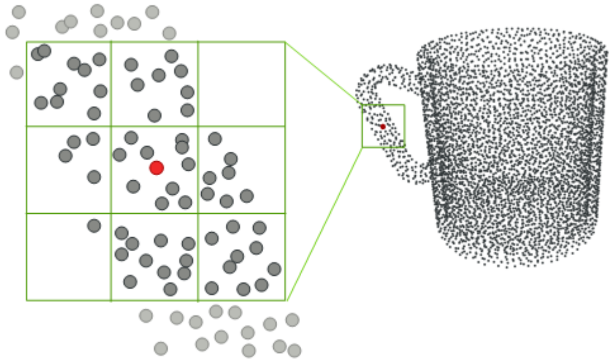
Related Work *regular processing*

[14] Li et al. SO-Net. CVPR 2018. Self-Organizing Map

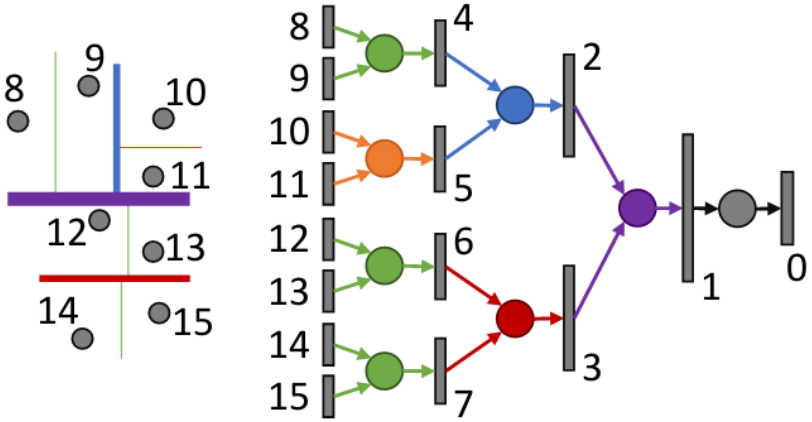
+ MLP



[15] Hua et al. Pointwise CNN. CVPR 2018.



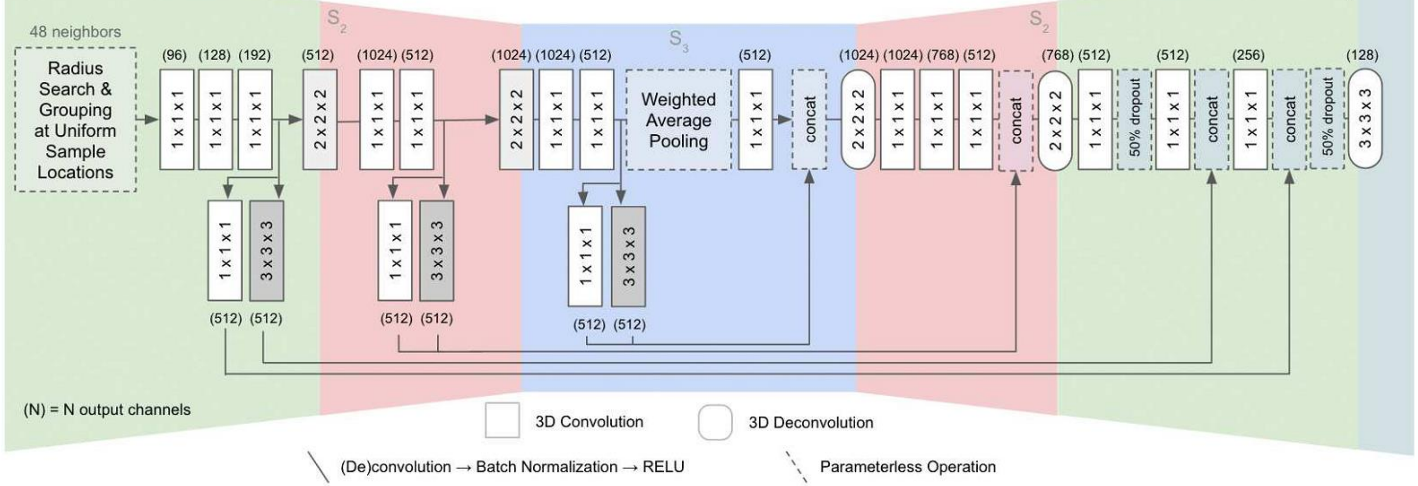
$$x_i^\ell = \sum_k w_k \frac{1}{|\Omega_i(k)|} \sum_{p_j \in \Omega_i(k)} x_j^{\ell-1}$$



Kd-Tree

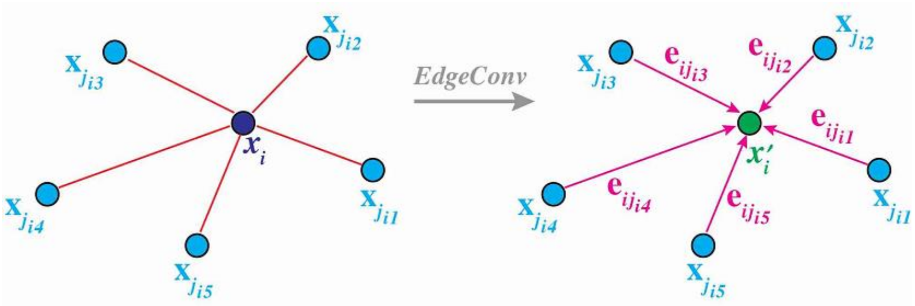
[16] Klokov et al. Kd-Net. ICCV 2017.

3D grid + PointNet + 3D CNN 200*k* points at once



[17] Rethage et al. FCPN. ECCV 2018.

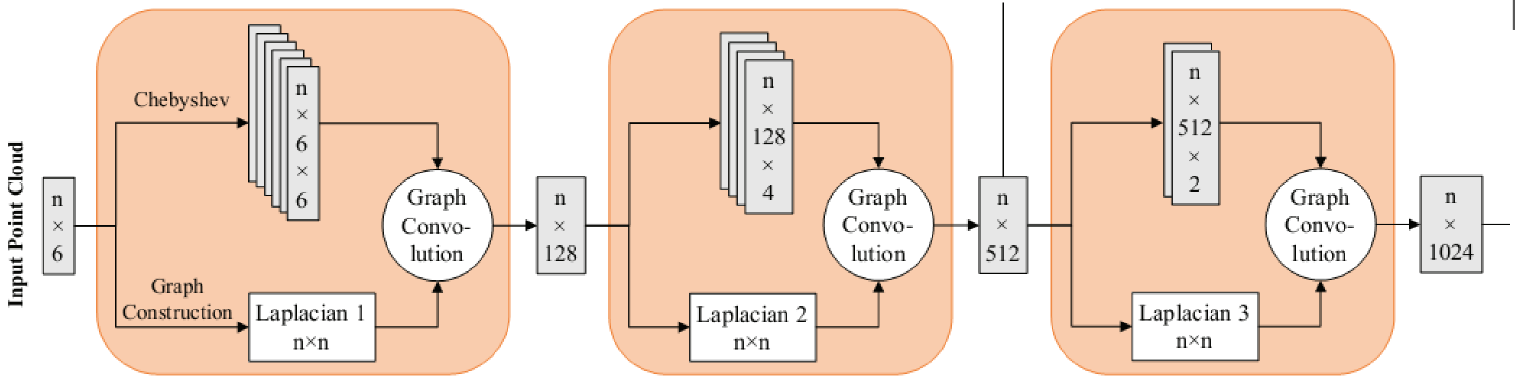
Related Work *graph-based modeling*



[29] Wang et al. DGCNN. TOG 2019.

EdgeConv kNN

$$x'_i = \mathop{\square}_{j:(i,j)\in\mathcal{E}} h_{\mathbf{\Theta}}(x_i, x_j). \qquad h_{\mathbf{\Theta}}(x_i, x_j - x_i)$$



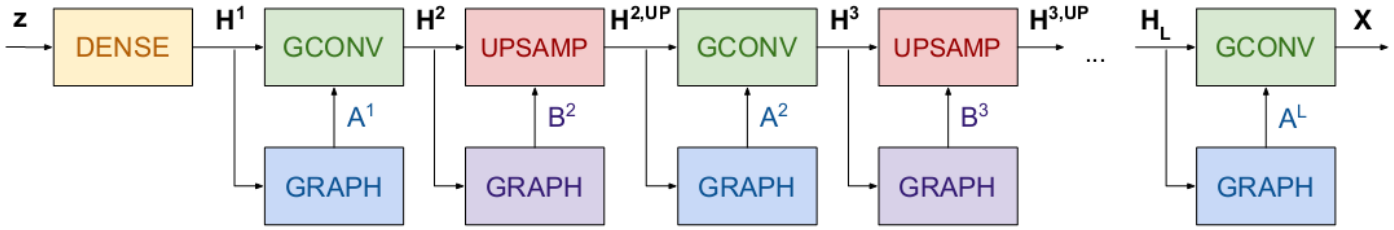
[30] Te et al. Regularized GCNN. MM 2018.

$$\mathbf{y} = g_\theta(\mathcal{L})\mathbf{x} = \sum_{k=0}^{K-1} \theta_k T_k(\mathcal{L})\mathbf{x}$$

$$a_{i,j} = \exp(-\beta \|\mathbf{p}_i - \mathbf{p}_j\|_2^2)$$

$$\sum_{i\sim j} a_{i,j}(y_i - y_j)^2$$

kNN



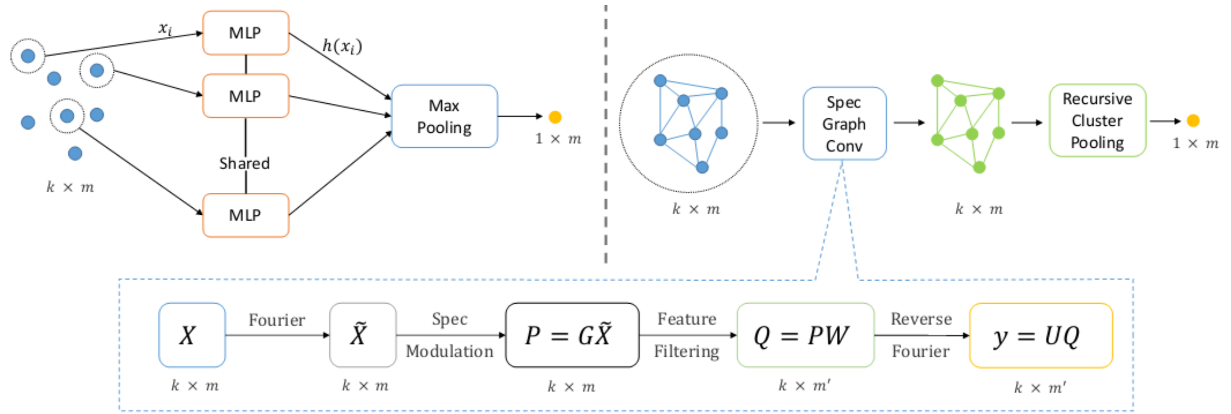
[31] Valsesia et al. GAN for Point Cloud. ICLR 2019.

$$\mathbf{h}_i^{l+1} = \sigma\left(\sum_{j\in\mathcal{N}_i^l} \frac{F_{\mathbf{w}^l}^l\left(\mathbf{h}_j^l - \mathbf{h}_i^l\right)\mathbf{h}_j^l}{|\mathcal{N}_i^l|} + \mathbf{h}_i^l\mathbf{W}^l + \mathbf{b}^l\right)$$

learn domain (the graph) and features simultaneously

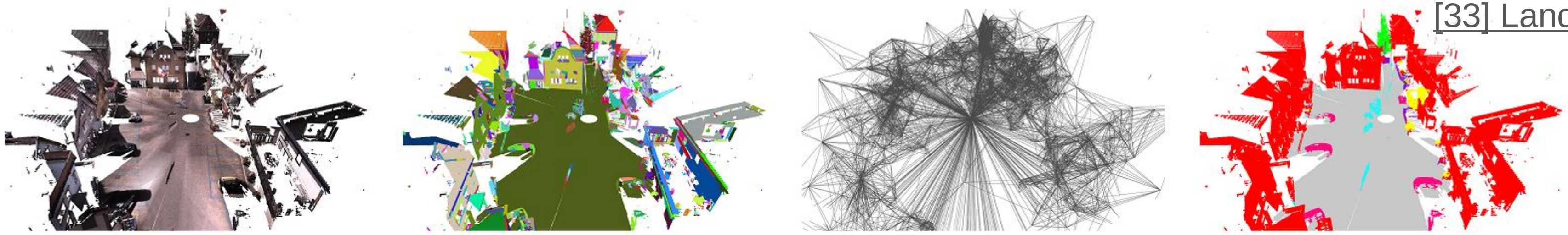
kNN

Related Work *graph-based modeling*



[32] Wang et al. Spectral Graph Convolution. ECCV 2018.

spectral graph conv +
recursive spectral cluster pooling



(a) RGB point cloud (b) Geometric partition (c) Superpoint graph (d) Semantic segmentation

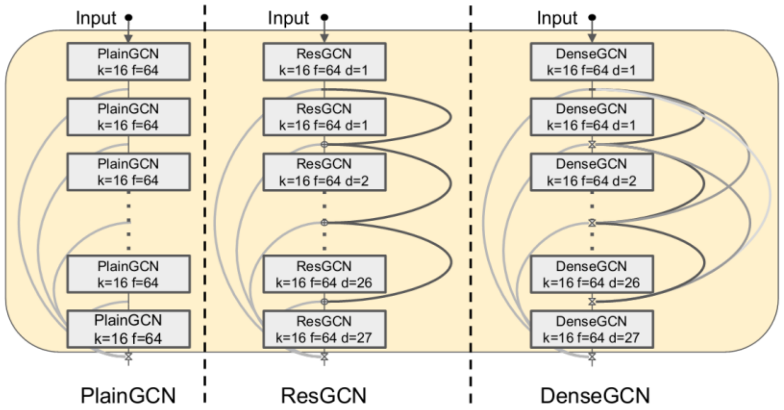
[33] Landrieu et al. Superpoint Graph. CVPR 2018.

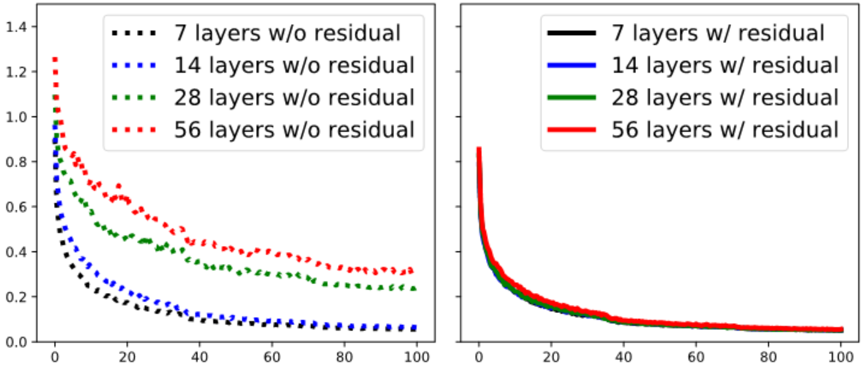
minimal partition + GCN

[34] Li et al. Gated GNN. ICLR 2016.

[35] Simonovsky et al. ECC. CVPR 2017.

[36] Landrieu et al. Oversegmentation. CVPR 2019.



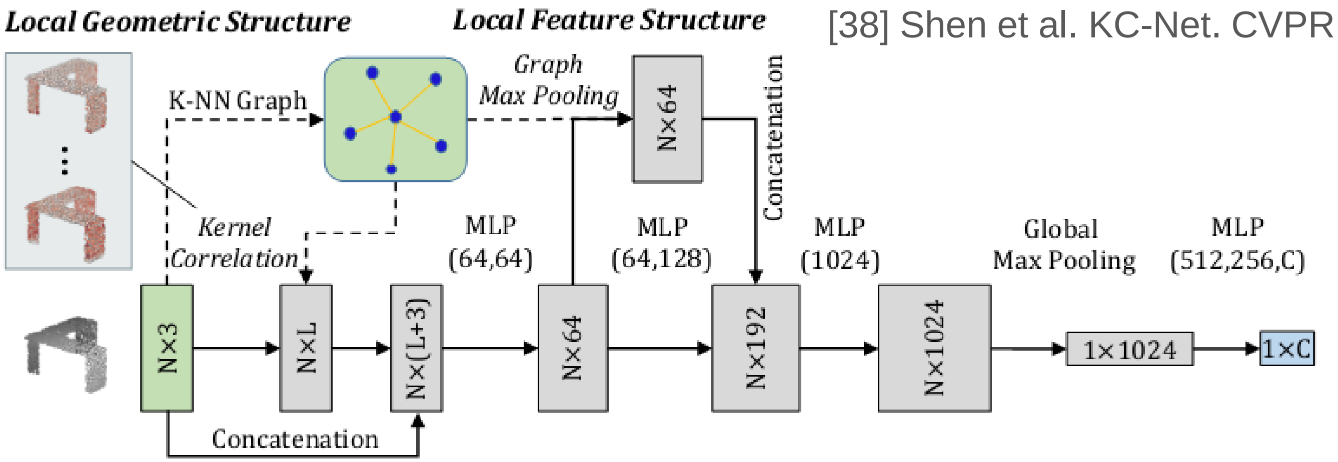


[37] Li et al. Deep GCNs. ICCV 2019.

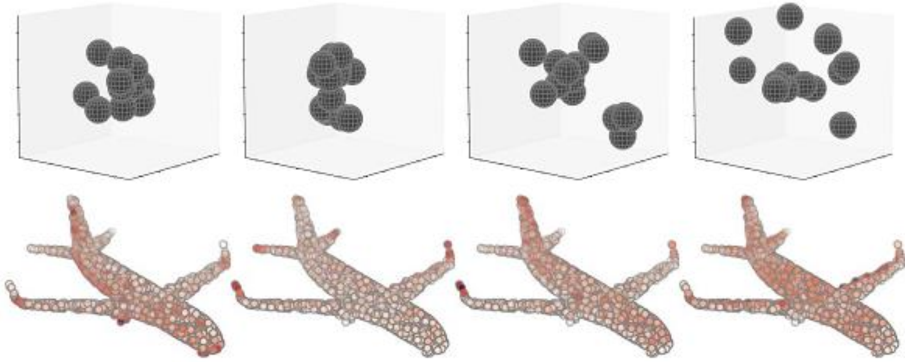
residual/dense connection
dilated conv

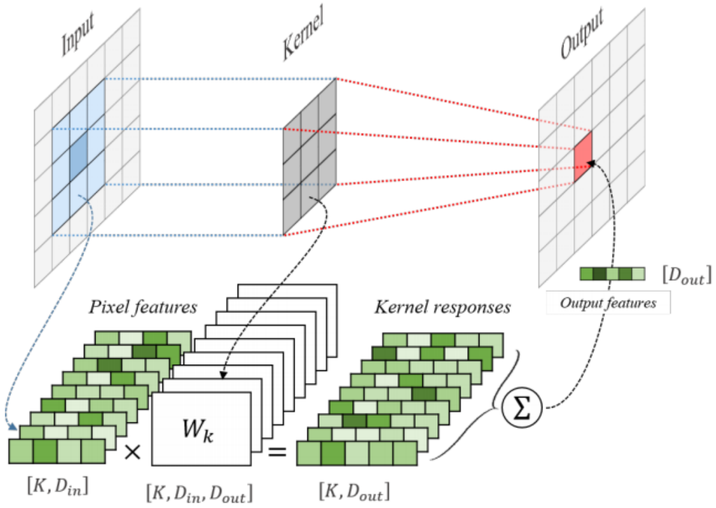
Related Work *convolution kernel*

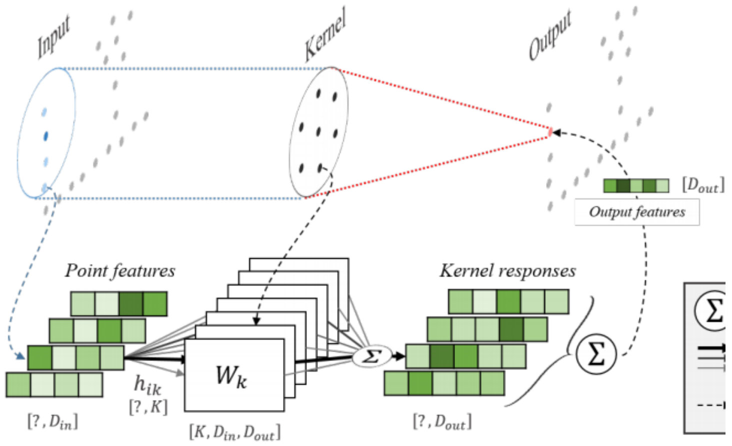
[38] Shen et al. KC-Net. CVPR 2018.



kernel correlation





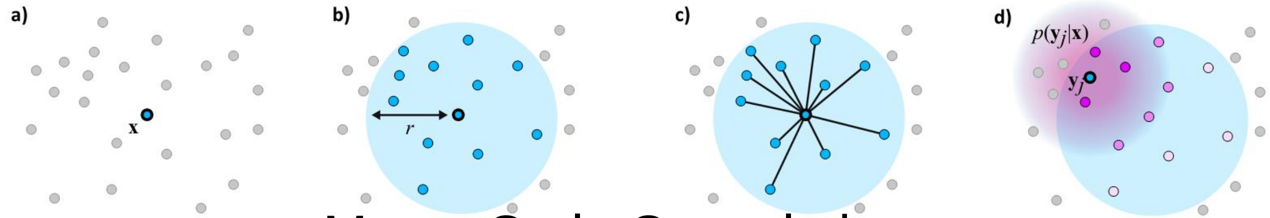


[39] Hugues et al. KPConv. arXiv 2019.

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

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

Related Work *convolution kernel*

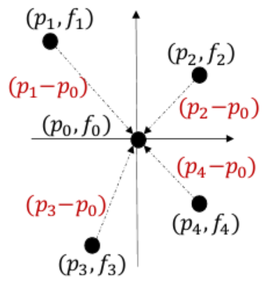


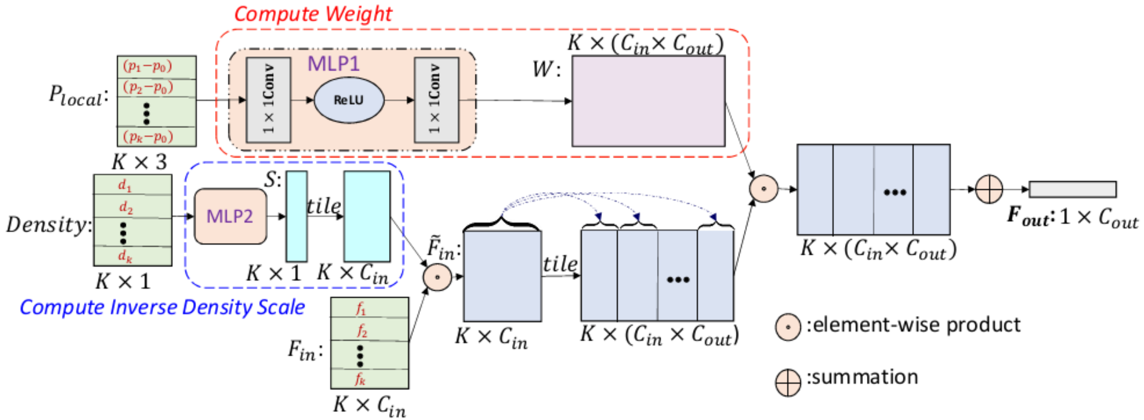
Monte Carlo Convolution

[22] HERMOSILLA et al. MCCNN. TOG 2018.

$$(f * g)(\mathbf{x}) = \int f(\mathbf{y}) g(\mathbf{x} - \mathbf{y}) d\mathbf{y}$$

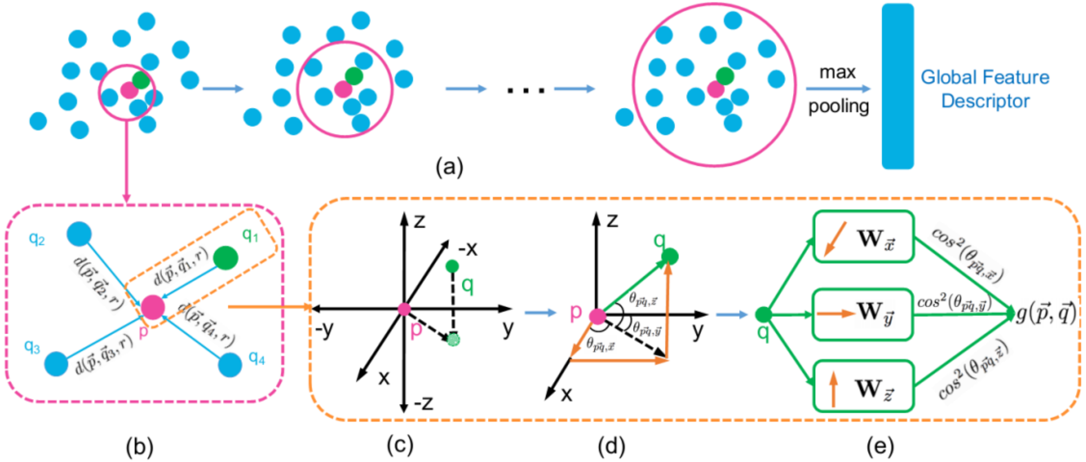
$$(f * g)(\mathbf{x}) \approx \frac{1}{|\mathcal{N}(\mathbf{x})|} \sum_{j \in \mathcal{N}(\mathbf{x})} \frac{f(\mathbf{y}_j) g\left(\frac{\mathbf{x}-\mathbf{y}_j}{r}\right)}{p(\mathbf{y}_j|\mathbf{x})}$$



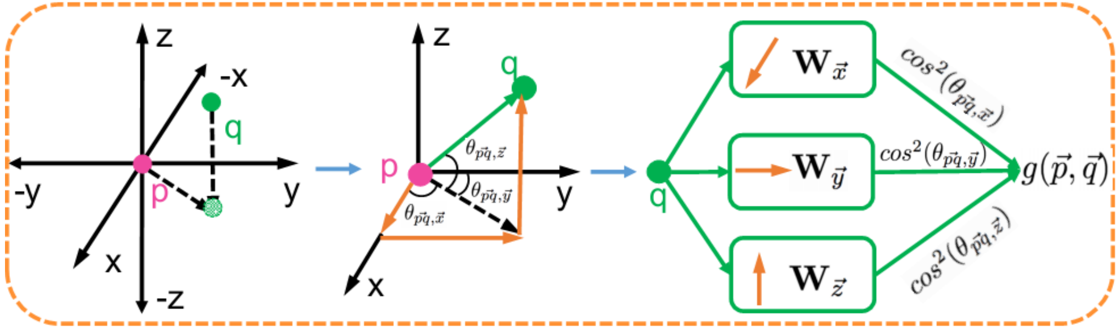


[23] Wu et al. PointConv. CVPR 2019.

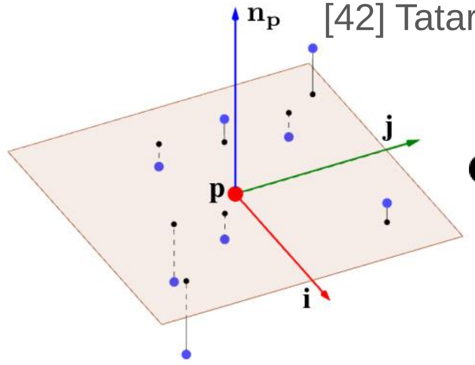
$$\mathbf{F}_{out} = \sum_{k=1}^{K} \sum_{c_{in}=1}^{C_{in}} S(k) \mathbf{W}(k, c_{in}) F_{in}(k, c_{in})$$



[41] Lan et al. Geo-CNN. CVPR 2019.



Related Work *convolution kernel*

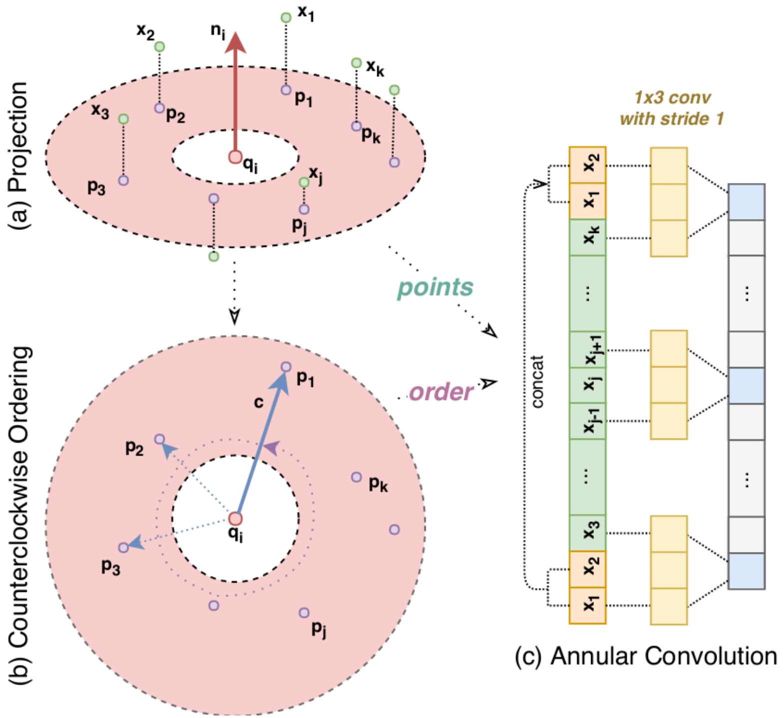


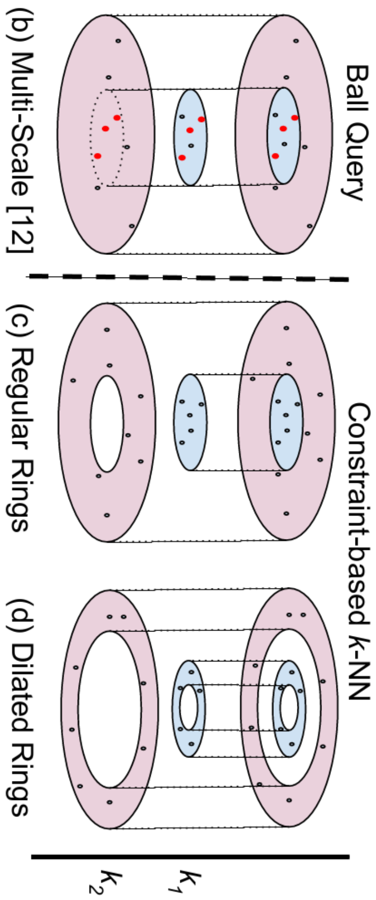
[42] Tatarchenko et al. Tangent Conv. CVPR 2018.

$$\|\mathbf{p}-\mathbf{q}\| < R \qquad \mathbf{r} = \mathbf{q} - \mathbf{p}$$

$$\mathbf{C} = \textstyle\sum_{\mathbf{q}} \mathbf{r}\mathbf{r}^{\top}$$ tangent image S

[43] Komarichev et al. A-CNN. CVPR 2019.

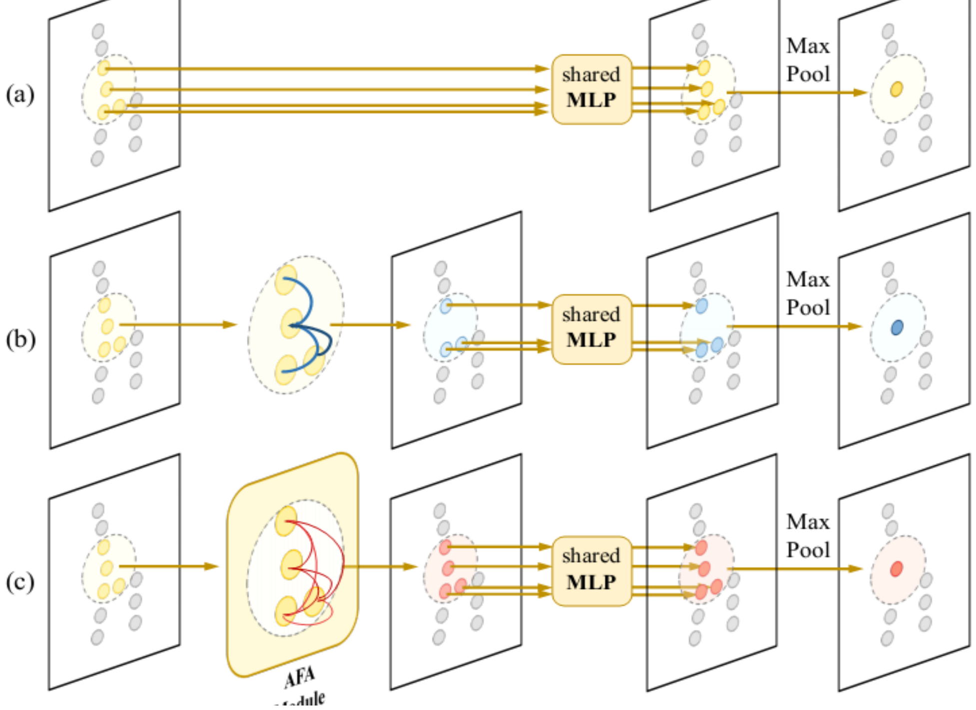




$$S(\mathbf{v}) = F(\mathbf{q}) \quad S(\mathbf{u}) = \sum_{\mathbf{v}} (w(\mathbf{u},\mathbf{v}) \cdot S(\mathbf{v}))$$

$$X(\mathbf{p}) = \int_{\pi_{\mathbf{p}}} c(\mathbf{u}) S(\mathbf{u})\, d\mathbf{u}$$

[44] Zhao et al. PointWeb. CVPR 2019.



$$f_{mod}(F_i, \mathbb{F}) = \sum_{j=1}^{M} f_{imp}(F_i, F_j) \cdot f_{rel}(F_i, F_j)$$

$$F_i' = F_i + \Delta F_i \qquad f_{imp}(F_i, F_j) = MLP(g(F_i, F_j))$$

$$\Delta F_i = f_{mod}(F_i, \mathbb{F}) \qquad f_{rel}(F_i, F_j) = F_j \qquad F_i - F_j$$

Related Work *Robustness*

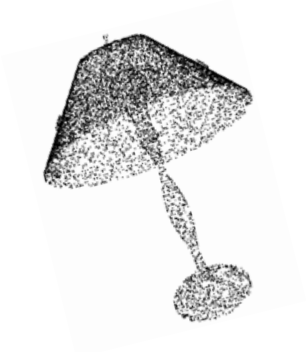
Robustness to rigid transformation

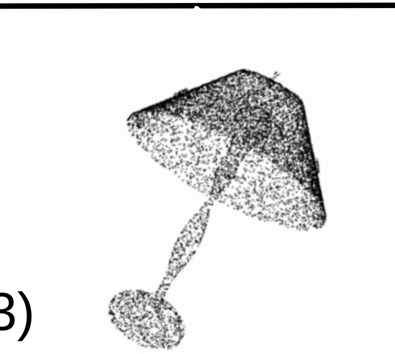
Normalization:

✓ Translation

✓ Scale

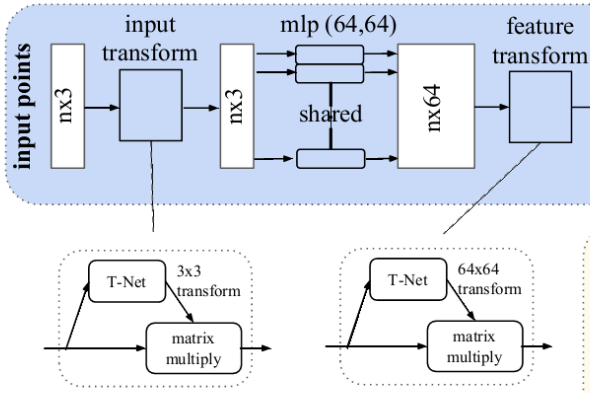
x Rotation





X · **R** = (Nx3) · (3x3)

Data augmentation or align



[18] Qi et al. PointNet. CVPR 2017.

Robustness to sampling density

Multi-scale or Input dropout

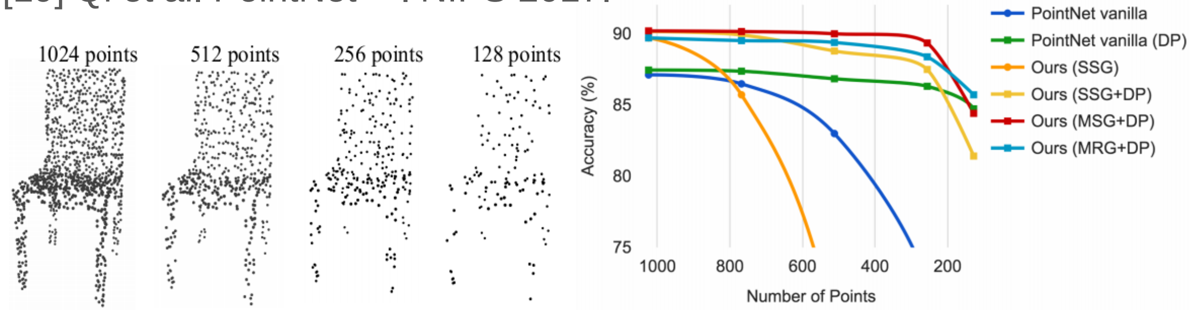
Monte Carlo integration

Embedding density info.

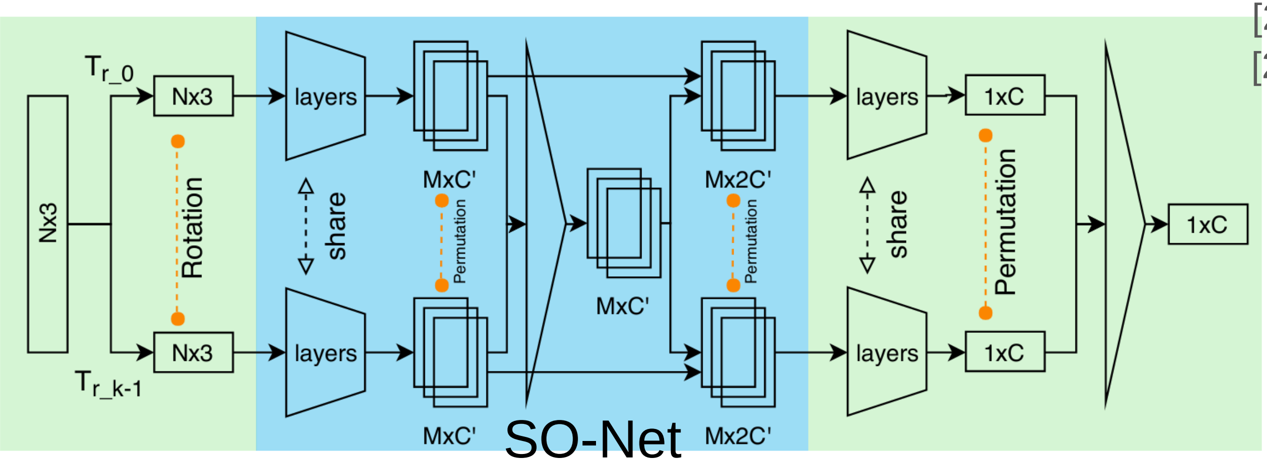
[22] HERMOSILLA et al. MCCNN. TOG 2018.

[23] Wu et al. PointConv. CVPR 2019.

[19] Qi et al. PointNet++. NIPS 2017.



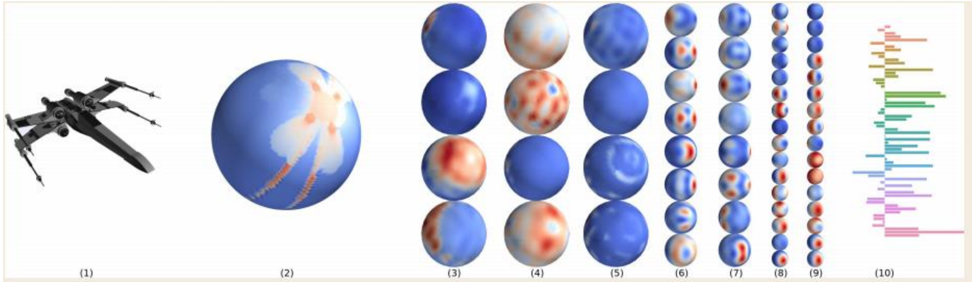
Related Work *Robustness*



[24] Li et al. Discrete Rotation Equivariance. ICRA 2019.

[25] Cohen et al. Group Equivariant CNN. ICML 2016.

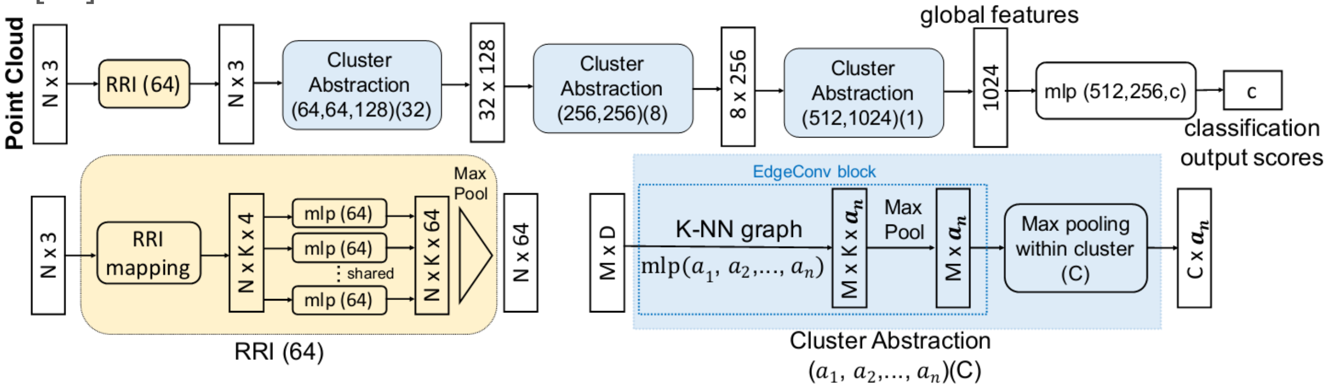
$$\Phi(T_{r_i} x) = T'_{r_i}\Phi(x)$$

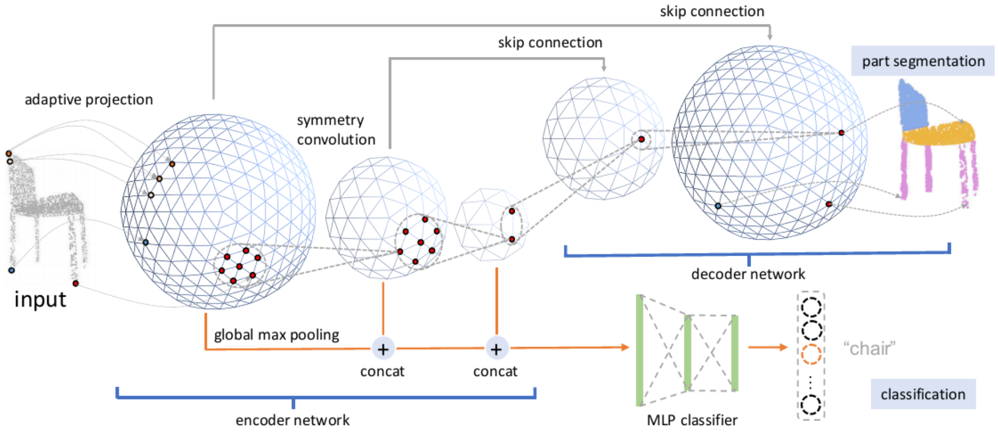


[26] Esteves et al. SO(3) Equivariant. ECCV 2018.

[27] Cohen et al. Spherical CNNs. ICLR 2018.

[40] Chen et al. ClusterNet. CVPR 2019.





[28] Rao et al. SFCNN. CVPR 2019.

Rigorously Rotation-Invariant (RRI) Representation

$$\|R\boldsymbol{x}\|_2^2 = \|\boldsymbol{x}\|_2^2 \quad \langle R\boldsymbol{x}, R\boldsymbol{y}\rangle = (R\boldsymbol{x})^{\mathsf{T}}(R\boldsymbol{y}) = \boldsymbol{x}^{\mathsf{T}}\boldsymbol{y} = \langle \boldsymbol{x}, \boldsymbol{y}\rangle$$

Github: awesome-point-cloud-analysis

CVPR, ICCV, ECCV, SIGGraph / Asia, TOG, NIPS, ICLR, AAAI, MM, ICRA, IROS, 3DV…… arXiv

awesome-point-cloud-analysis awesome

- Recent papers (from 2017)
- Datasets

Keywords

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

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

CVPR 2018, ~25

CVPR 2019, ~50

ICCV 2019, ?

2018

- [CVPR] SPLATNet: Sparse Lattice Networks for Point Cloud Processing. [caffe] [`seg.`] 🔥
- [CVPR] Attentional ShapeContextNet for Point Cloud Recognition. [`cls.` `seg.`]
- [CVPR] Mining Point Cloud Local Structures by Kernel Correlation and Graph Pooling. [code] [`cls.` `seg.`]
- [CVPR] FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformation. [code] [`cls.`]
- [CVPR] Pointwise Convolutional Neural Networks. [tensorflow] [`cls.` `seg.`]
- [CVPR] PU-Net: Point Cloud Upsampling Network. [tensorflow] [`rec.` `oth.`] 🔥
- [CVPR] SO-Net: Self-Organizing Network for Point Cloud Analysis. [pytorch] [`cls.` `seg.`] 🔥 ⭐
- [CVPR] Recurrent Slice Networks for 3D Segmentation of Point Clouds. [pytorch] [`seg.`]
- [CVPR] 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. [pytorch] [`seg.`] 🔥
- [CVPR] Deep Parametric Continuous Convolutional Neural Networks. [`seg.` `aut.`]
- [CVPR] PIXOR: Real-time 3D Object Detection from Point Clouds. [pytorch] [`det.` `aut.`]
- [CVPR] SGPN: Similarity Group Proposal Network for 3D Point Cloud Instance Segmentation. [tensorflow] [`seg.`] 🔥
- [CVPR] Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. [pytorch] [`seg.`] 🔥
- [CVPR] VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. [tensorflow] [`det.` `aut.`] 🔥 ⭐

Outline

1. Introduction
2. Brief review
3. RS-CNN & DensePoint
4. Summary & Outlook

Relation-Shape Convolutional Neural Network for Point Cloud Analysis

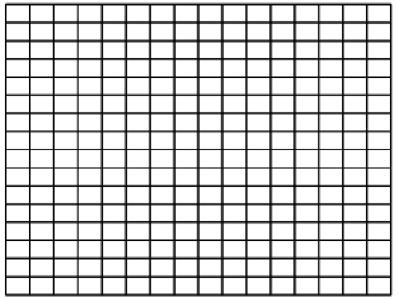
Yongcheng Liu, Bin Fan, Shiming Xiang, Chunhong Pan

CVPR 2019 Oral & Best paper finalist

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

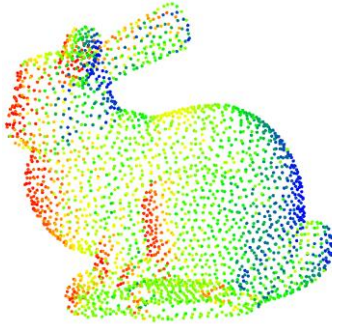
RS-CNN *Motivation*

2D image



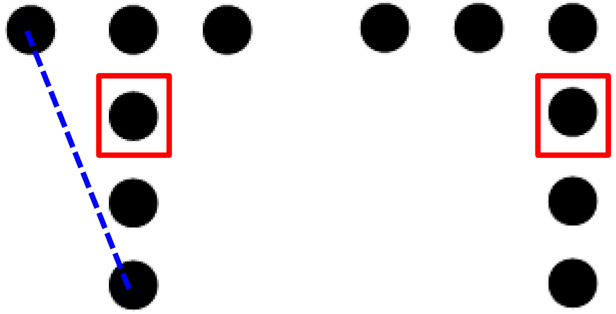
Info: RGB

3D point cloud



Info: spatial layout

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

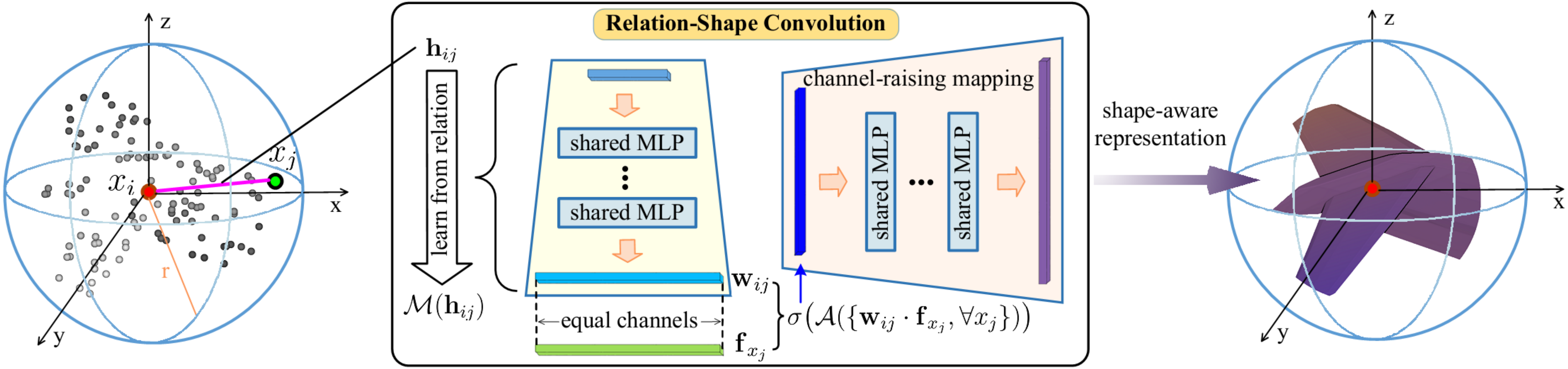
  gradient only w.r.t single point - implicit $\mathcal{T}(\mathbf{f}_{x_j}) = \mathbf{w}_j \cdot \mathbf{f}_{x_j}$
- Conversion: learn from relation $\mathcal{T}(\mathbf{f}_{x_j}) = \mathbf{w}_{ij} \cdot \mathbf{f}_{x_j} = \mathcal{M}(\mathbf{h}_{ij}) \cdot \mathbf{f}_{x_j}$

$\mathbf{h}_{ij}$: 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*



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

high-level relation encoding + channel raising mapping

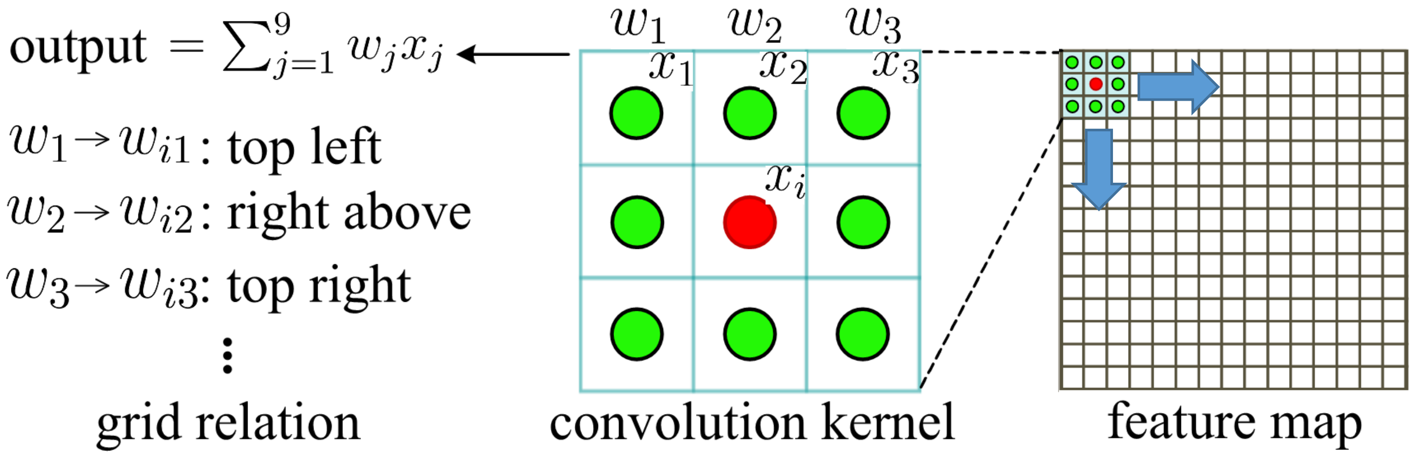
low-level relation $\mathbf{h}_{ij}$: (3D Euclidean distance, $x_i - x_j$, x_i, x_j) 10 channels

RS-CNN *RS-Conv: Properties*

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

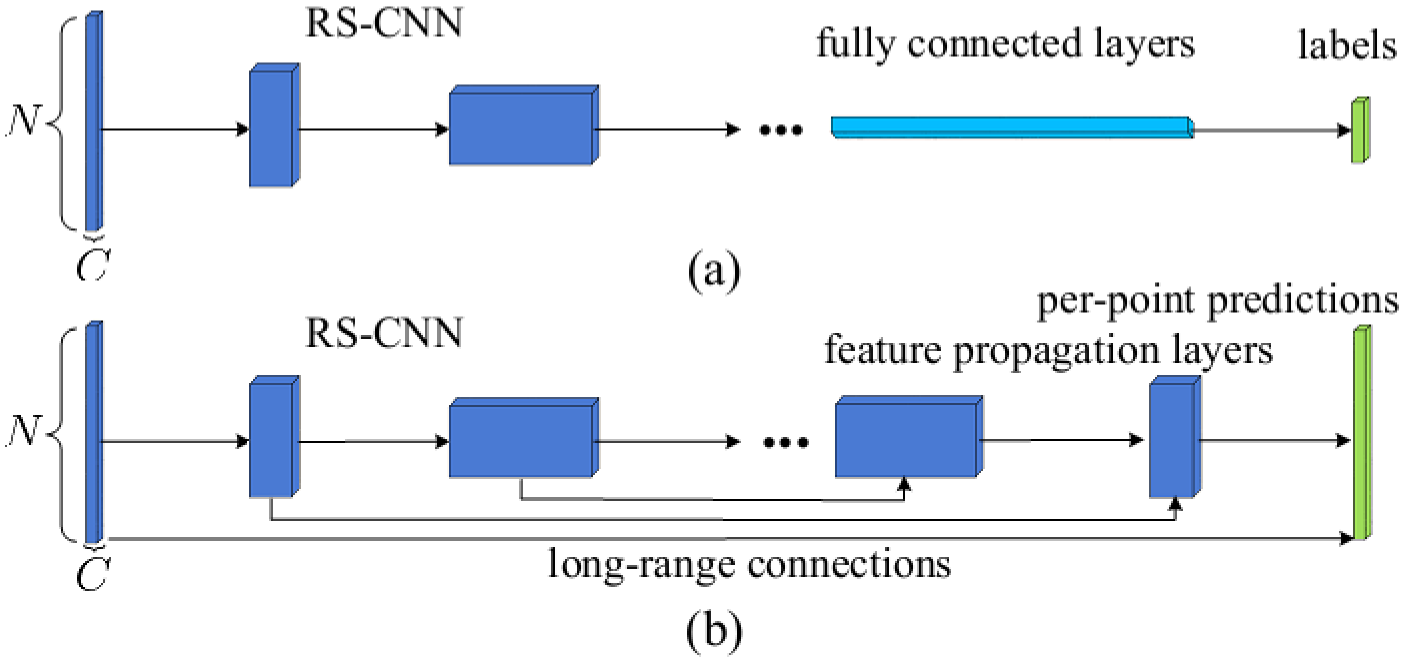
- ✓ Permutation invariance
- ✓ Robustness to rigid transformation in Relation Learning, e.g., 3D Euclidean distance
- ✓ Points' interaction
- ✓ Weight sharing

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*

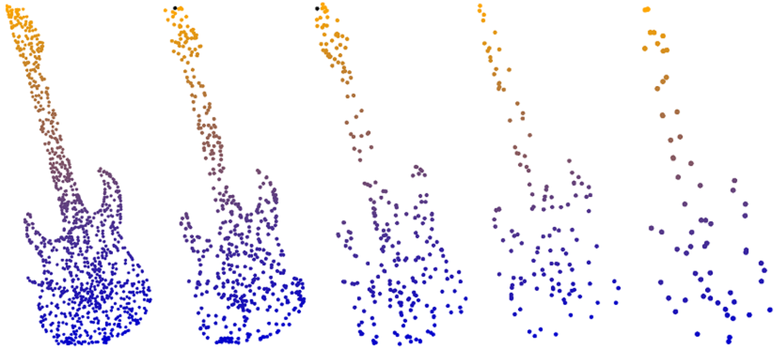


Farthest Point Sampling + Sphere Neighborhood + RS-Conv

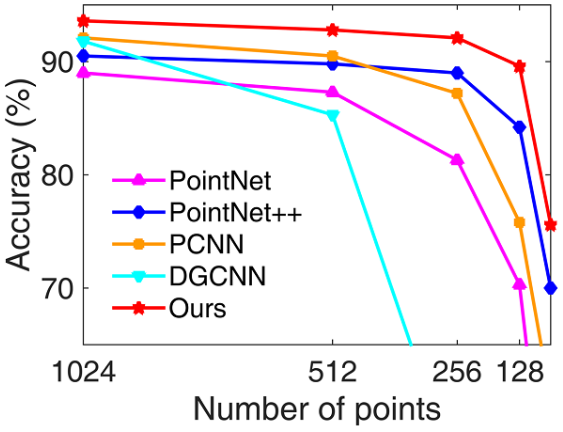
RS-CNN *Shape classification*

ModelNet40 benchmark

Robustness to sampling density



1024 512 256 128 64



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

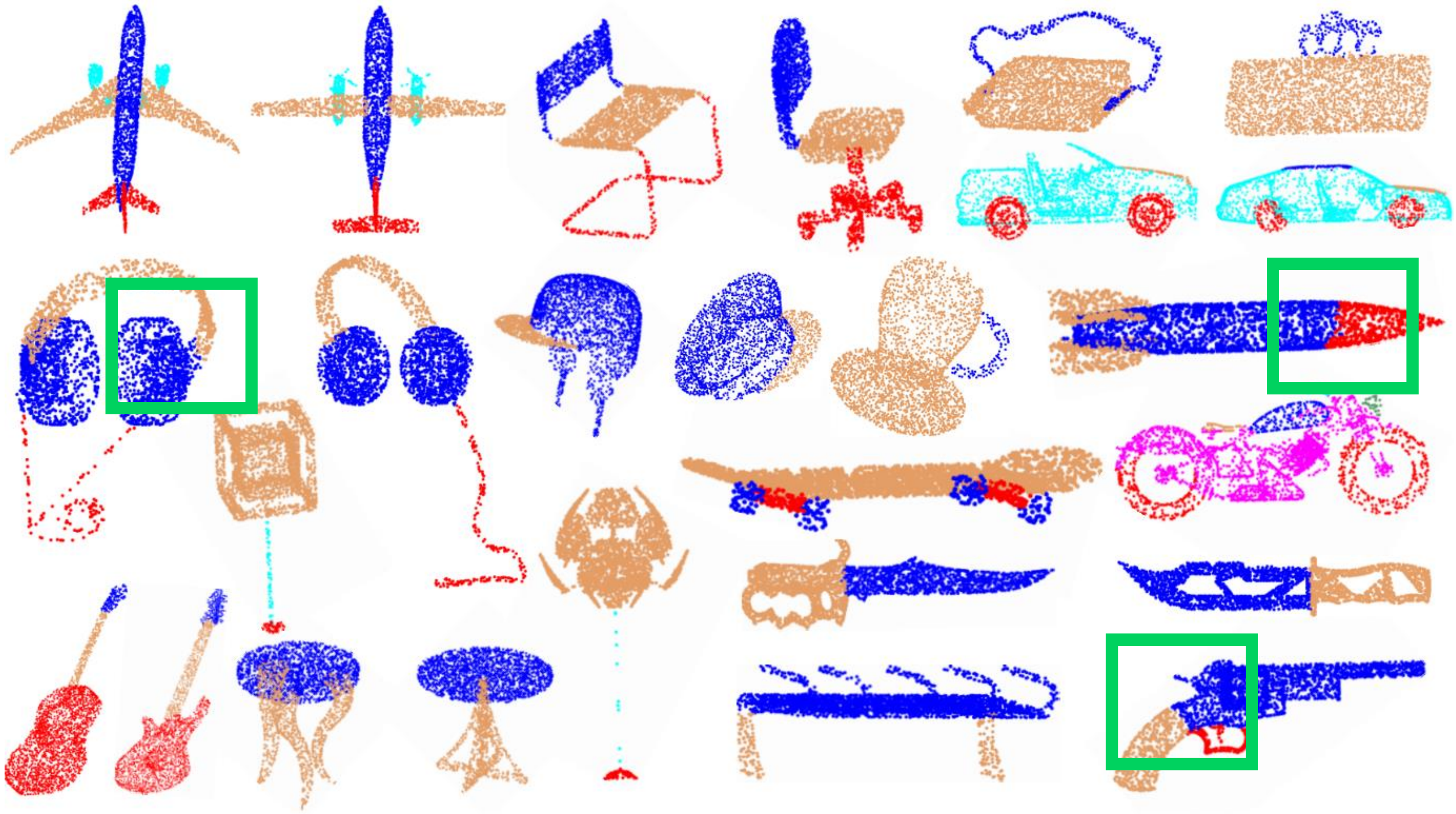
RS-CNN ShapePart Segmentation

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>



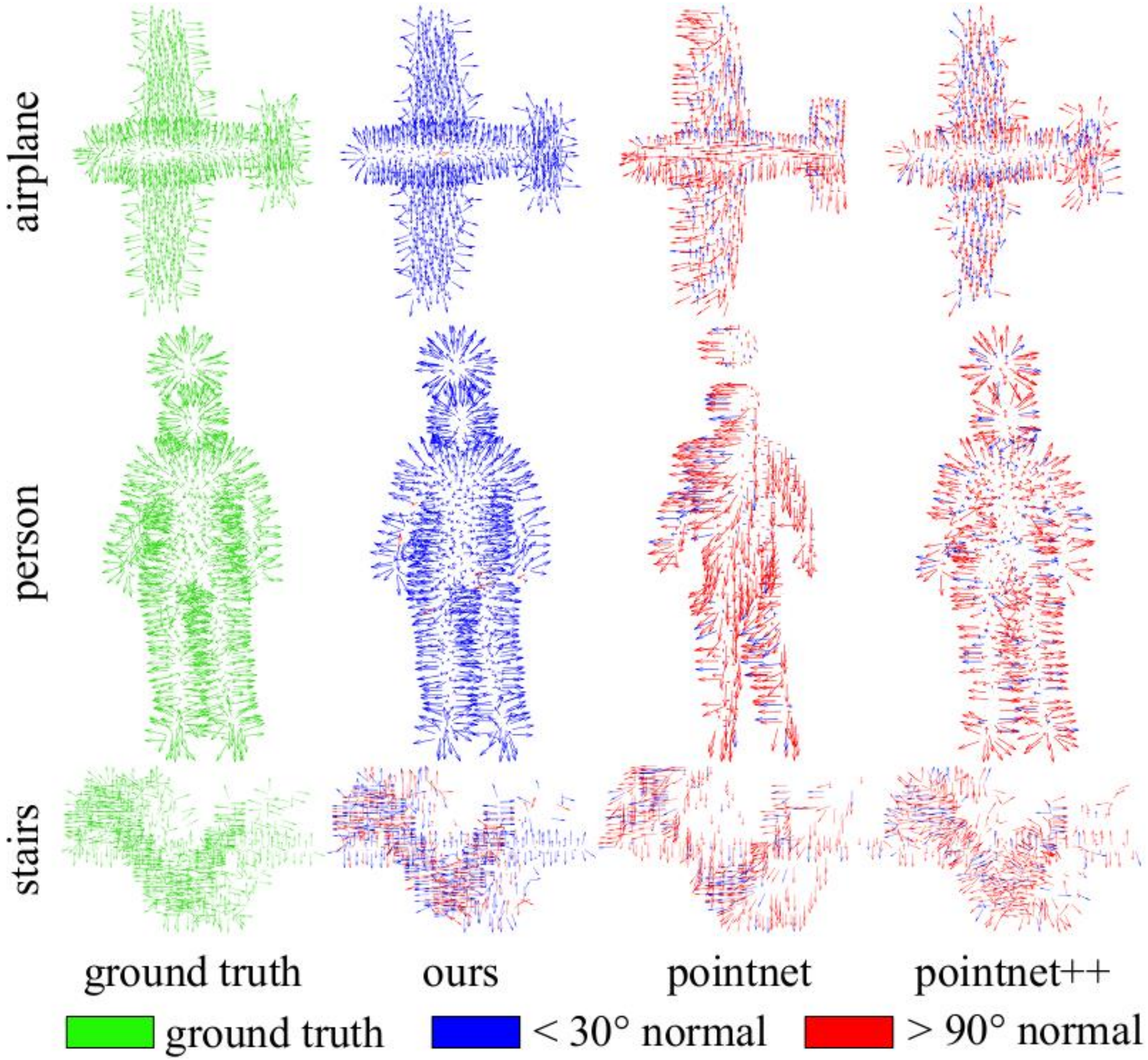
Diverse, confusing shapes

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**

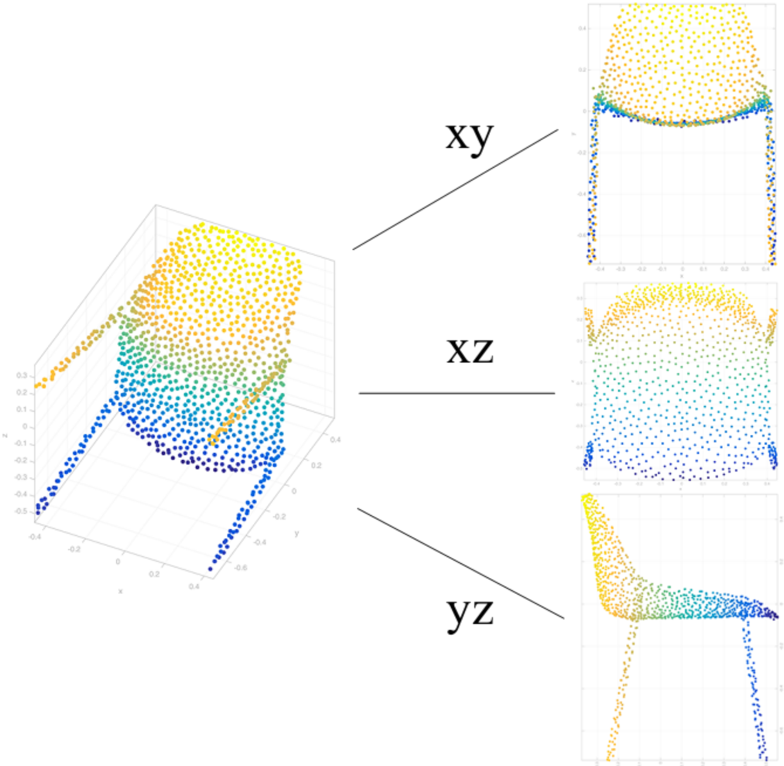
less effective for some intractable shapes, such as spiral stairs and intricate plants



RS-CNN <u>*Geometric priors*</u>

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

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

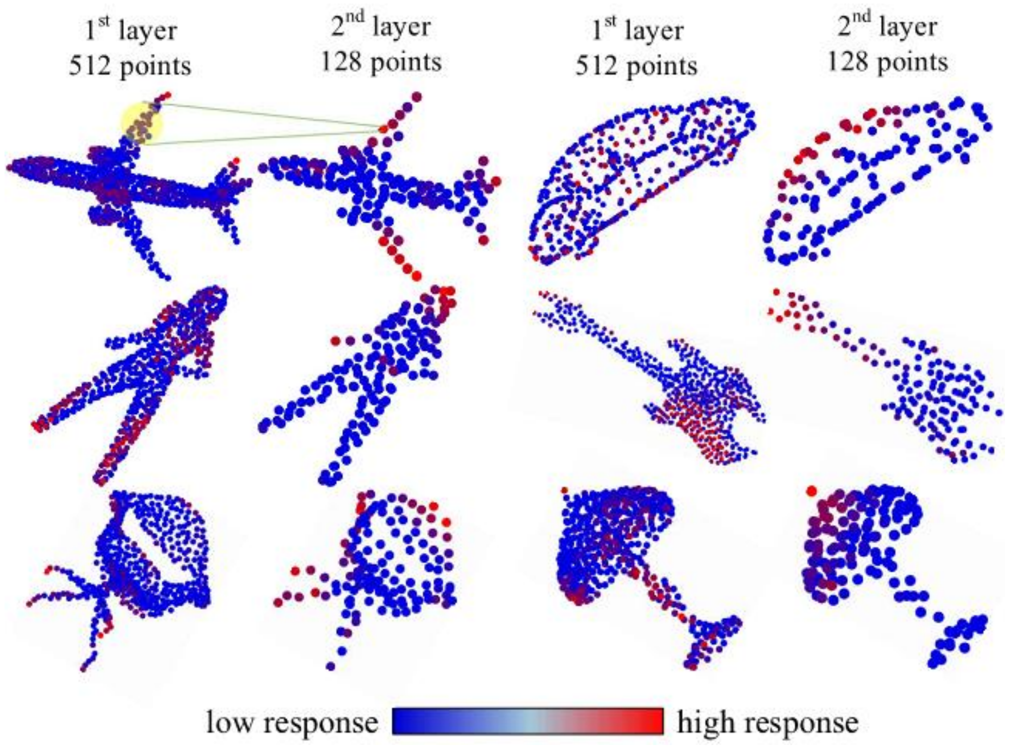
RS-CNN *Model analysis*

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

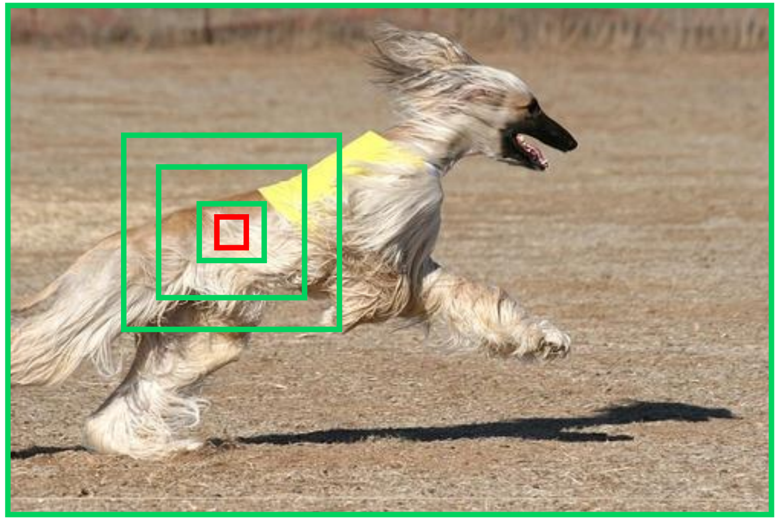
DensePoint: Learning Densely Contextual Representation for Efficient Point Cloud Processing

<u>Yongcheng Liu</u>, Bin Fan, Gaofeng Meng, Jiwen Lu, Shiming Xiang, Chunhong Pan

ICCV 2019

Code: https://github.com/Yochengliu/DensePoint

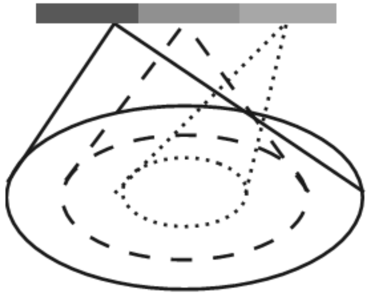
DensePoint *Motivation*

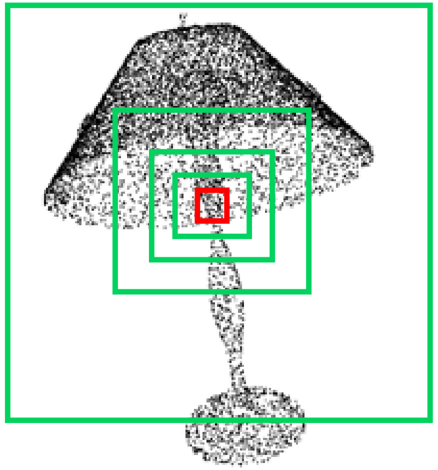


Context: potential semantic dependencies between a target pattern and its surroundings

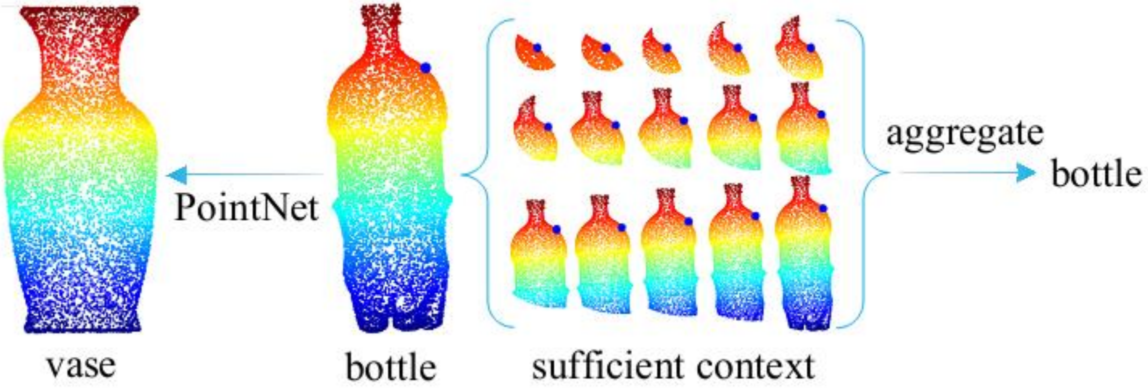
Multi-scale learning – high complexity

- parameters
- FLOPs
- scale limitation
- unintuitive (scale ←→ semantic level)





- ✓ Efficient solution using deep learning?
- ✓ Explore its performance on point cloud from various aspects.

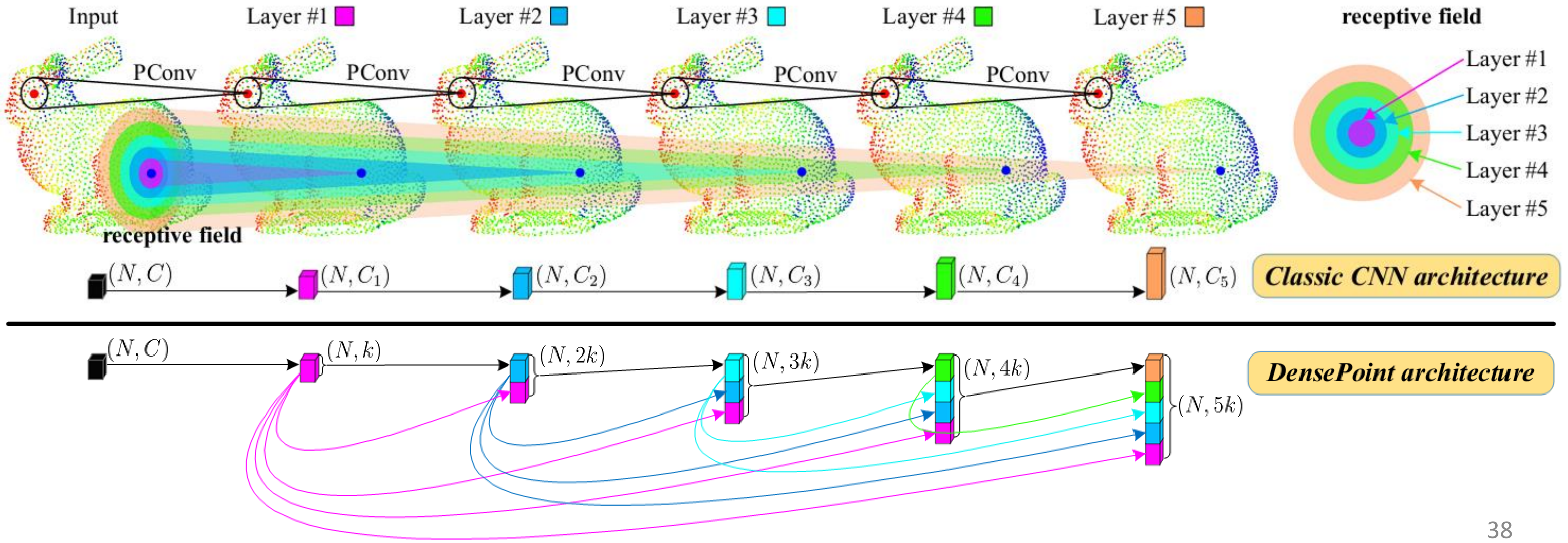


DensePoint *Method*

key idea: multi-level receptive fields + efficient conv on point cloud

dense connections + efficient point convolution

progressively aggregate multi-scale info. in an ***organic*** manner!



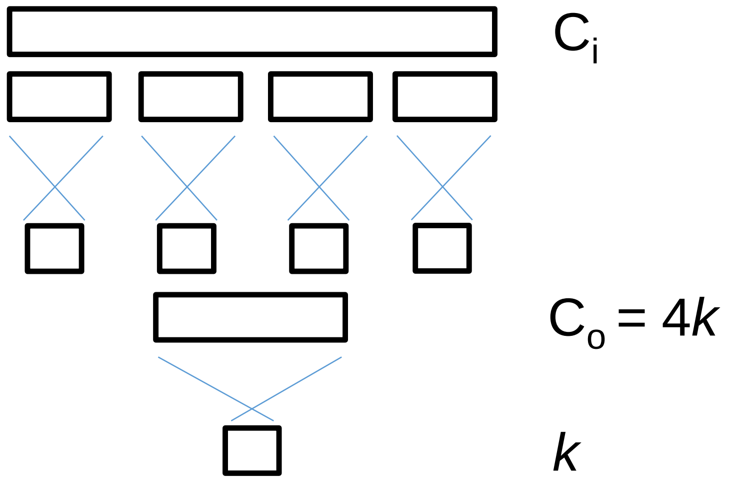
DensePoint <u>*Method: efficient PConv*</u>

$$\mathbf{f}_{\mathcal{N}(x)} = \rho(\{\phi(\mathbf{f}_{x_n}),\ \forall x_n \in \mathcal{N}(x)\})$$

ø: single-layer perceptron

enhanced PConv: filter grouping

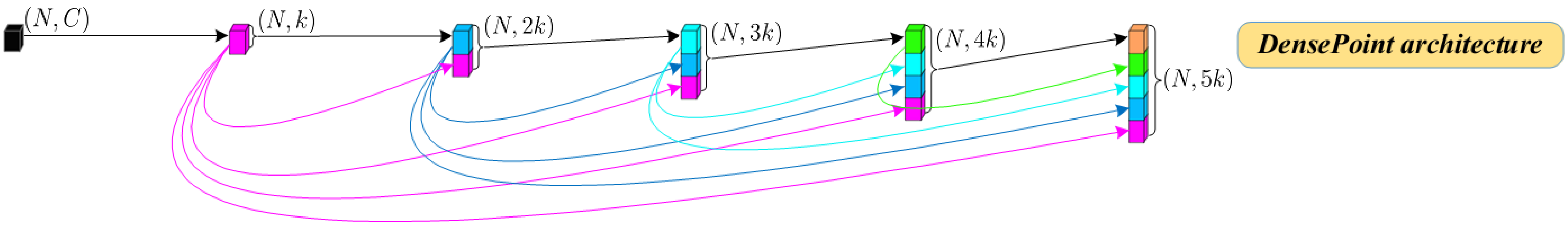
$$\mathbf{f}_{\mathcal{N}(x)} = \psi(\rho(\{\widehat{\phi}(\mathbf{f}_{x_n}),\ \forall x_n \in \mathcal{N}(x)\}))$$



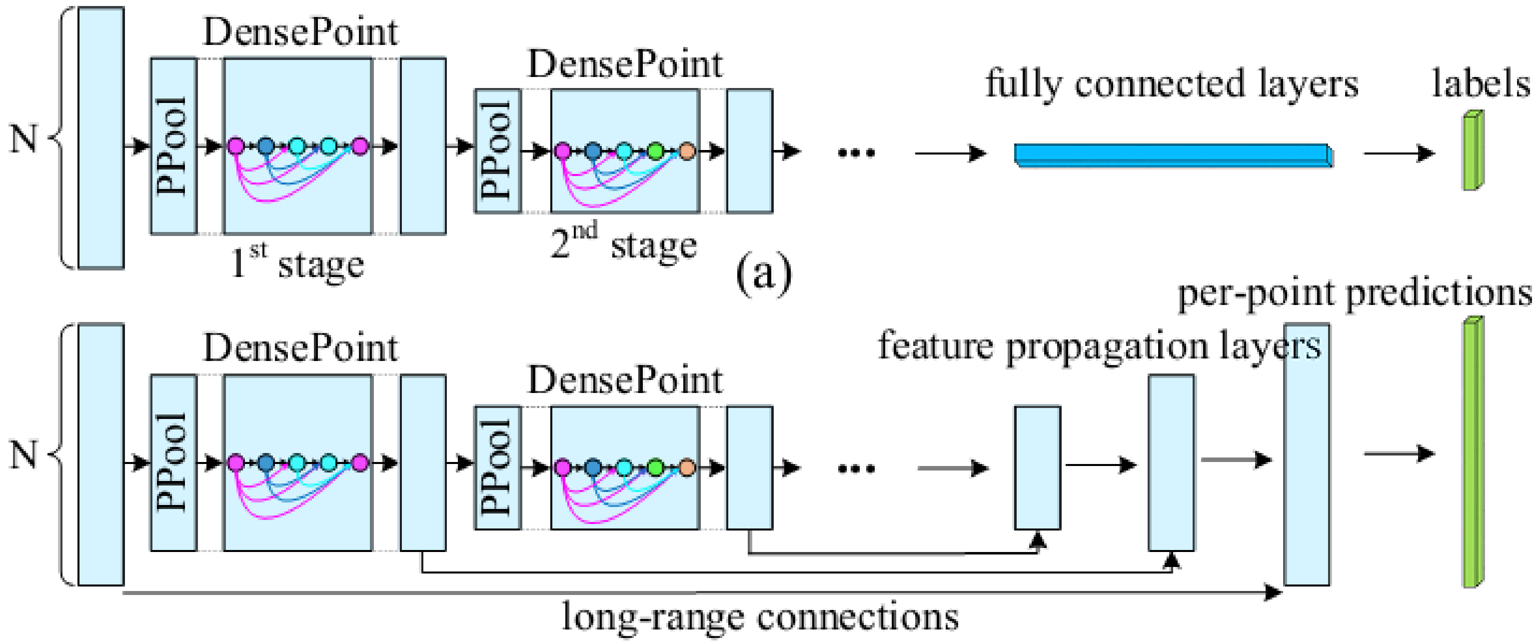
$C_i{*}k$

vs.

$C_i{*}k/4 + 4k^2$



DensePoint *Method*



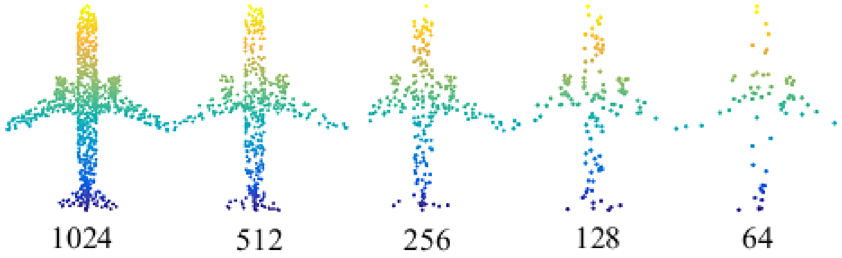
Farthest Point Sampling + Sphere Neighborhood + ePConv + PPool

+ dense connections

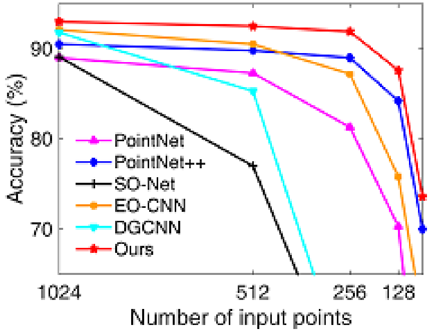
DensePoint *Shape classification*

ModelNet40 benchmark

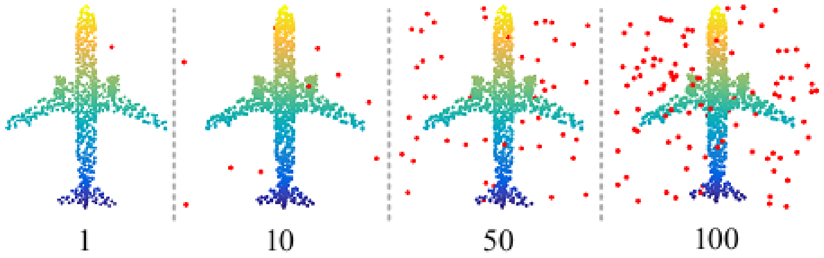
Robustness to sampling density and noise



(a)

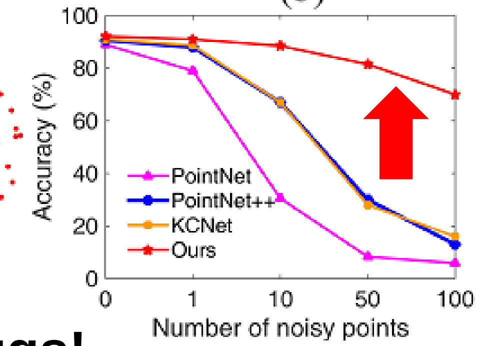


(b)



(c)

No any augs!



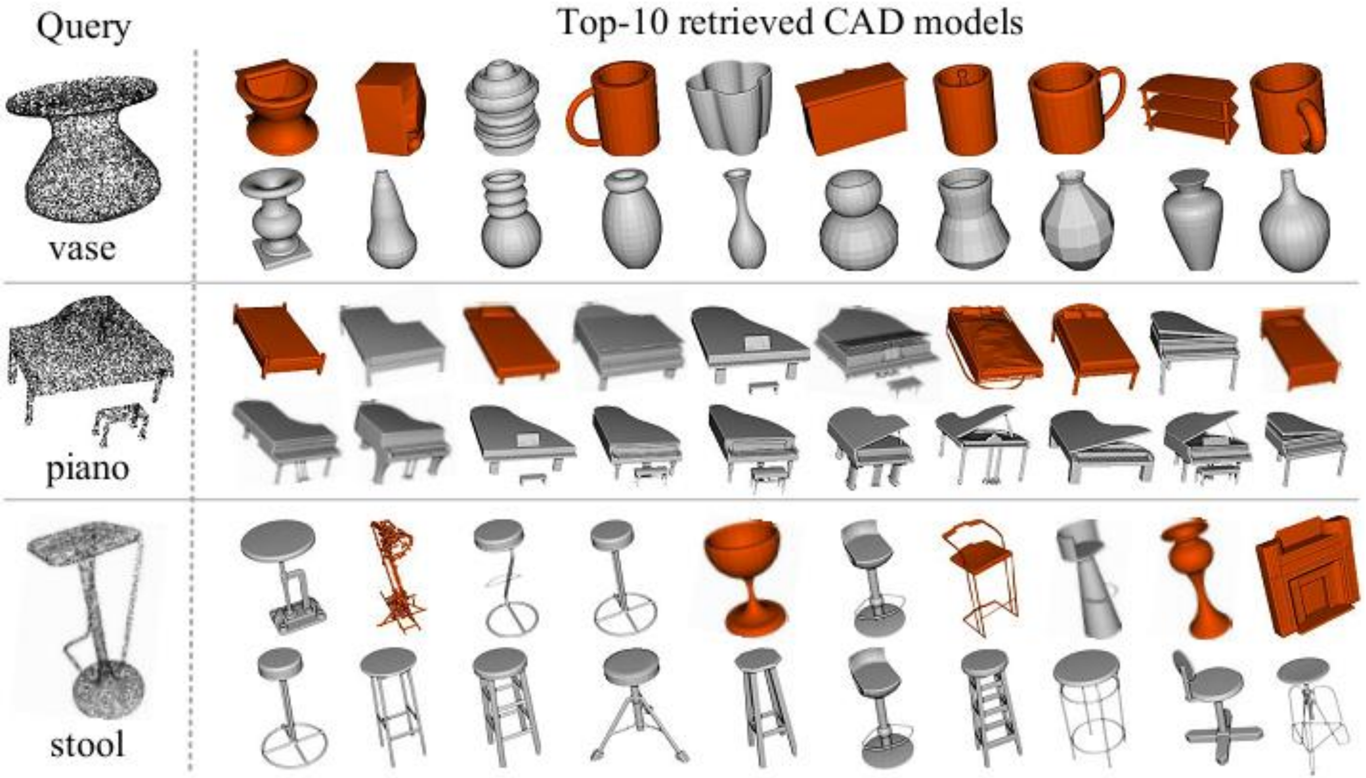
(d)

method	input	#points	M40	M10
Pointwise-CNN [12]	pnt	1k	86.1	-
Deep Sets [60]	pnt	1k	87.1	-
ECC [40]	pnt	1k	87.4	90.8
PointNet [31]	pnt	1k	89.2	-
SCN [55]	pnt	1k	90.0	-
Kd-Net(depth=10) [21]	pnt	1k	90.6	93.3
PointNet++ [33]	pnt	1k	90.7	-
MC-Conv [11]	pnt	1k	90.9	-
KCNet [39]	pnt	1k	91.0	94.4
MRTNet [4]	pnt	1k	91.2	-
SpecGCN [49]	pnt	1k	91.5	-
DGCNN [52]	pnt	1k	92.2	-
PointCNN [26]	pnt	1k	92.2	-
PCNN [1]	pnt	1k	92.3	94.9
Ours	**pnt**	**1k**	**93.2**	**96.6**
SO-Net [24]	pnt	2k	90.9	94.1
Kd-Net(depth=15) [21]	pnt	32k	91.8	94.0
O-CNN [50]	pnt, nor	-	90.6	-
Spec-GCN [49]	pnt, nor	1k	91.8	-
PointNet++ [33]	pnt, nor	5k	91.9	-
SpiderCNN [56]	pnt, nor	5k	92.4	-
SO-Net [24]	pnt, nor	5k	93.4	95.7

DensePoint *Shape retrieval*

ModelNet40 benchmark

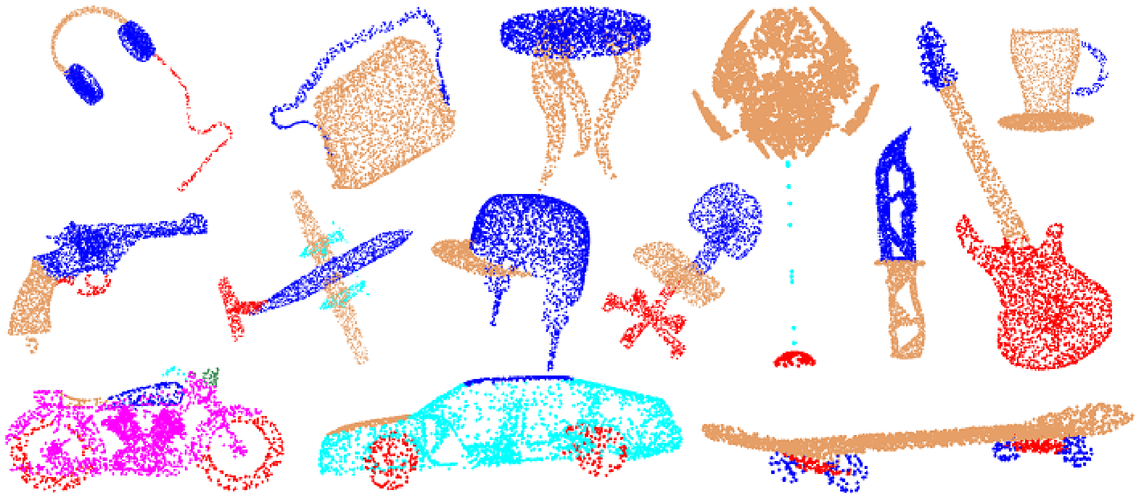
input	method	#points/views	M40	M10
Points	PointNet [10]	1k	70.5	-
	Ours	**1k**	**88.5**	**93.2**
Images	GVCNN [3]	12	85.7	-
	Triplet-center [10]	12	88.0	-
	PANORAMA-ENN [38]	-	86.3	93.3
	SeqViews [7]	12	89.1	89.5

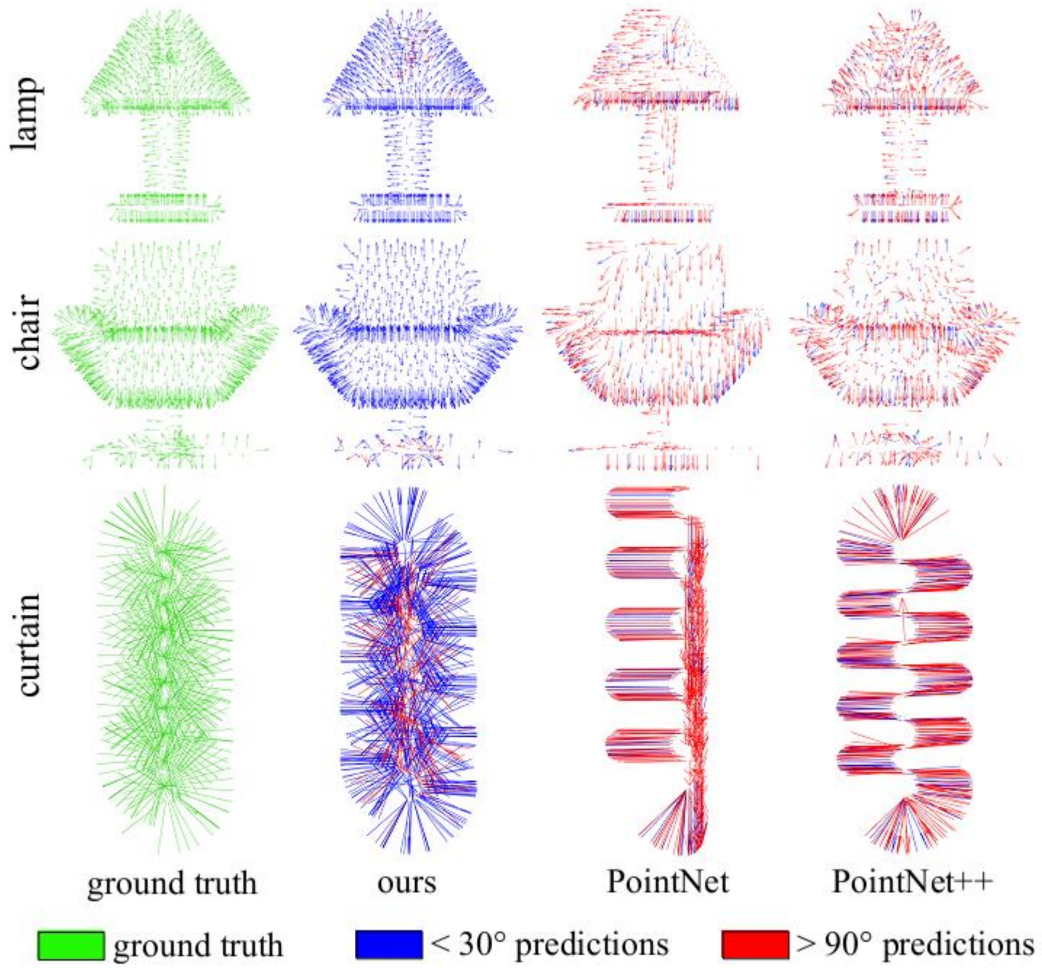


PointNet

DensePoint

DensePoint *ShapePart Segmentation & normal estimation*





DensePoint *model complexity*

method	#params	#FLOPs/sample	acc.(%)
PointNet [31]	3.50M	440M	89.2
PointNet++ [26]	1.48M	1684M	90.7
DGCNN [26]	1.84M	2767M	92.2
SpecGCN [26]	2.05M	1112M	91.5
KCNet [39]	0.90M	-	91.0
PCNN [26]	8.20M	294M	92.3
PointCNN [26]	0.60M	1581M	92.2
Ours ($k = 12$, $L = 11$)	0.56M	294M	92.1
Ours ($k = 24$, $L = 11$)	0.67M	651M	93.2
Ours ($k = 24$, $L = 6$)	**0.53M**	**148M**	92.1

1024 points

method	#points	Time (ms)		Memory (GB)	
		training	test	training	test
PointNet [31]	1024	55	22	1.318	0.469
PointNet++ [33]	1024	195	47	8.311	2.305
DGCNN [52]	1024	300	68	4.323	1.235
PointCNN [26]	1024	55	38	2.501	1.493
Ours (k=24, L=11)	1024	21	10	3.745	1.228
Ours (k=24, L=6)	1024	10	5	1.468	0.886
Ours (k=24, L=11)	4096	21	10	7.503	1.767
Ours (k=24, L=6)	4096	10	5	2.417	1.638
Ours (k=24, L=11)	8192	21	10	14.521	3.027
Ours (k=24, L=6)	8192	10	5	4.335	2.776

batchsize = 16

Titan Xp

Outline

Summary & Outlook

Brief review

- PointNet family
- regular processing
- graph-based modeling
- convolution kernel

attention/self-attention

…

RS-CNN & DensePoint

- relatoin modeling
  - geometry & deep learning
- contextual learning & efficiency
  - visual recognition & robust learning

Summary & Outlook

Advantages

- ✓ raw sensor data, e.g., Lidar
- ✓ simple representation: N * (x, y, z, color, normal…)
- ✓ better 3D shape capturing

Why emerging?

- ✓ autonomous driving
- ✓ AR & VR
- ✓ robot manipulation
- ✓ Geomatics
- ✓ 3D face & medical
- ✓ AI-assisted shape design in 3D game and animation, etc.
- ✓ <u>open problem, flexible</u>

- efficiency in large-scale point cloud
- multi-sensor/multi-modal
- reconstruction
- high-precision
- robustness

- ✓ geometric DL, segmentation (instance), detection, completion, registration…
- ✓ capsule, GAN, one-shot/zero-shot, meta-learning, NAS

Welcome to the world of 3D point cloud!

References

[1] Wu et al. 3D ShapeNets: A Deep Representation for Volumetric Shapes. CVPR 2015.

[2] Su et al. Multi-view Convolutional Neural Networks for 3D Shape Recognition. ICCV 2015.

[3] Mo et al. PartNet: A Large-scale Benchmark for Fine-grained and Hierarchical Part-level 3D Object Understanding. CVPR 2019.

[4] Armeni et al. 3d semantic parsing of large-scale indoor spaces. CVPR 2016.

[5] Hackel et al. Semantic3d. net: A new large-scale point cloud classification benchmark. ISPRS 2017.

[6] Dai et al. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. CVPR 2017.

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

[8] Su et al. SPLATNet: Sparse Lattice Networks for Point Cloud Processing. CVPR 2018.

[9] Kiefel et al. Permutohedral Lattice CNNs. ICLR 2015.

[10] Jampani et al. Learning sparse high dimensional filters: Image filtering, dense CRFs and bilateral neural networks. CVPR 2016.

[11] Atzmon et al. Point Convolutional Neural Networks by Extension Operators. SIGGRAPH 2018.

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

[13] Jiang et al. PointSIFT: A SIFT-like Network Module for 3D Point Cloud Semantic Segmentation. arXiv 2018.

References

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

[15] Hua et al. Pointwise Convolutional Neural Networks. CVPR 2018.

[16] Klokov et al. Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models. ICCV 2017.

[17] Rethage et al. Fully-Convolutional Point Networks for Large-Scale Point Clouds. ECCV 2017.

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

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

[20] Qi et al. Frustum PointNets for 3D Object Detection from RGB-D Data. CVPR 2018.

[21] Liu et al. FlowNet3D: Learning Scene Flow in 3D Point Clouds. CVPR 2019.

[22] HERMOSILLA et al. Monte Carlo Convolution for Learning on Non-Uniformly Sampled Point Clouds. TOG 2018.

[23] Wu et al. PointConv: Deep Convolutional Networks on 3D Point Clouds. CVPR 2019.

[24] Li et al. Discrete Rotation Equivariance for Point Cloud Recognition. ICRA 2019.

[25] Cohen et al. Group Equivariant Convolutional Networks. ICML 2016.

[26] Esteves et al. Learning SO(3) Equivariant Representations with Spherical CNNs. ECCV 2018.

[27] Cohen et al. Spherical CNNs. ICLR 2018.

[28] Rao et al. Spherical Fractal Convolution Neural Networks for Point Cloud Recognition. CVPR 2019.

References

[29] Wang et al. Dynamic Graph CNN for Learning on Point Clouds. TOG 2019.

[30] Te et al. RGCNN: Regularized Graph CNN for Point Cloud Segmentation. MM 2018.

[31] Valsesia et al. Learning Localized Generative Models for 3D Point Clouds via Graph Convolution. ICLR 2019.

[32] Wang et al. Local Spectral Graph Convolution for Point Set Feature Learning. ECCV 2018.

[33] Landrieu et al. Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. CVPR 2018.

[34] Li et al. Gated graph sequence neural networks. ICLR 2016.

[35] Simonovsky et al. Dynamic edge-conditioned filters in convolutional neural networks on graphs. CVPR 2017.

[36] Landrieu et al. Point Cloud Oversegmentation with Graph-Structured Deep Metric Learning. CVPR 2019.

[37] Li et al. Can GCNs Go as Deep as CNNs. ICCV 2019.

[38] Shen et al. Mining Point Cloud Local Structures by Kernel Correlation and Graph Pooling. CVPR 2018.

[39] Thomas et al. KPConv: Flexible and Deformable Convolution for Point Clouds. arXiv 2019.

[40] Chen et al. ClusterNet: Deep Hierarchical Cluster Network with Rigorously Rotation-Invariant Representation for Point Cloud Analysis. CVPR 2019.

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

[42] Tatarchenko et al. Tangent Convolutions for Dense Prediction in 3D. CVPR 2018.

References

[43] Komarichev et al. A-CNN: Annularly Convolutional Neural Networks on Point Clouds. CVPR 2019.

[44] Zhao et al. PointWeb: Enhancing Local Neighborhood Features for Point Cloud Processing. CVPR 2019.

University of Chinese Academy of Sciences

Thanks for your attention !

yongcheng.liu@nlpr.ia.ac.cn