Mining Point Cloud Local Structures by Kernel Correlation and Graph Pooling

Yiru Shen*[1], Chen Feng*[2], Yaoqing Yang[3], Dong Tian[2]

[1]Clemson University, [2]Mitsubishi Electric Research Laboratories (MERL), [3]Carnegie Mellon University

CVPR 2018 SALT LAKE CITY • JUNE 18-22

Motivations

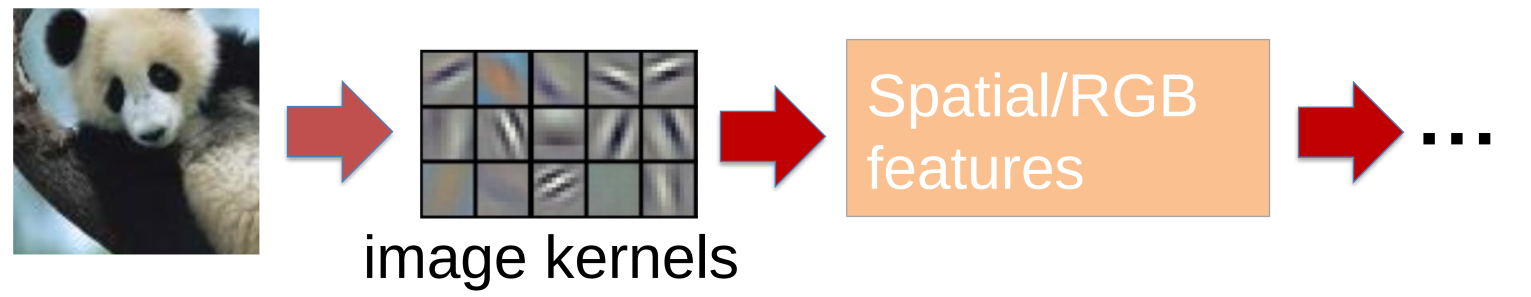
- Irregular organization in point clouds makes CNN not directly applicable.
- Point clouds are often converted to organized representations (voxel/multi-view) before deep networks.
- PointNet by Qi et al. directly consumes unordered point clouds but is difficult to capture local shape information.
- PointNet++/KdNet are time-consuming.

Goal

Efficient learning of local features in 3D point clouds.

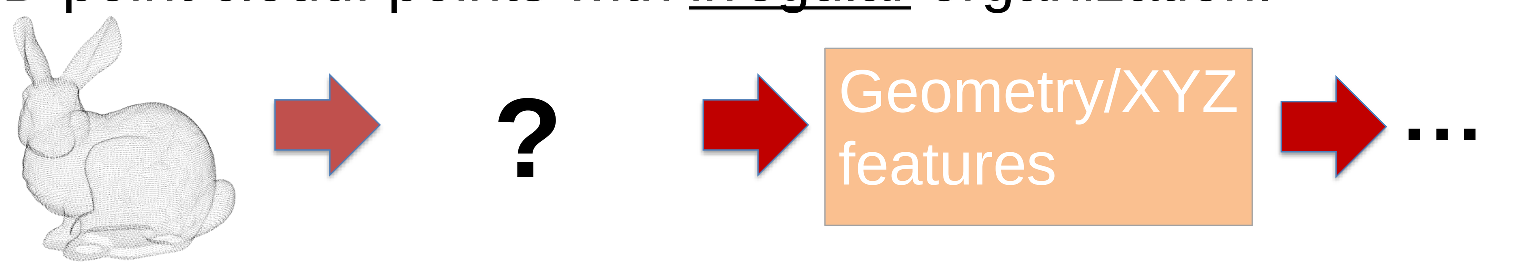
Intuition

2D image: pixels with <u>ordered</u> organization.



CNN: find out "image kernels" with a measure of affinities to a local image patch (convolution).

3D point cloud: points with <u>irregular</u> organization.



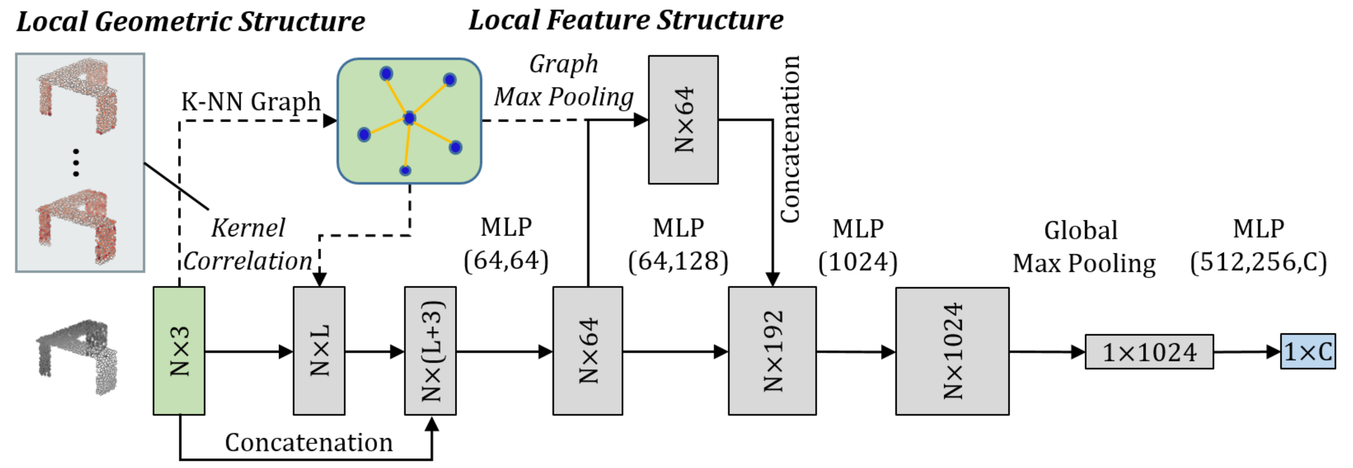
Our KCNet: find out "point kernels" with a measure of affinities to a set of local points (kernel correlation).

Key Ideas

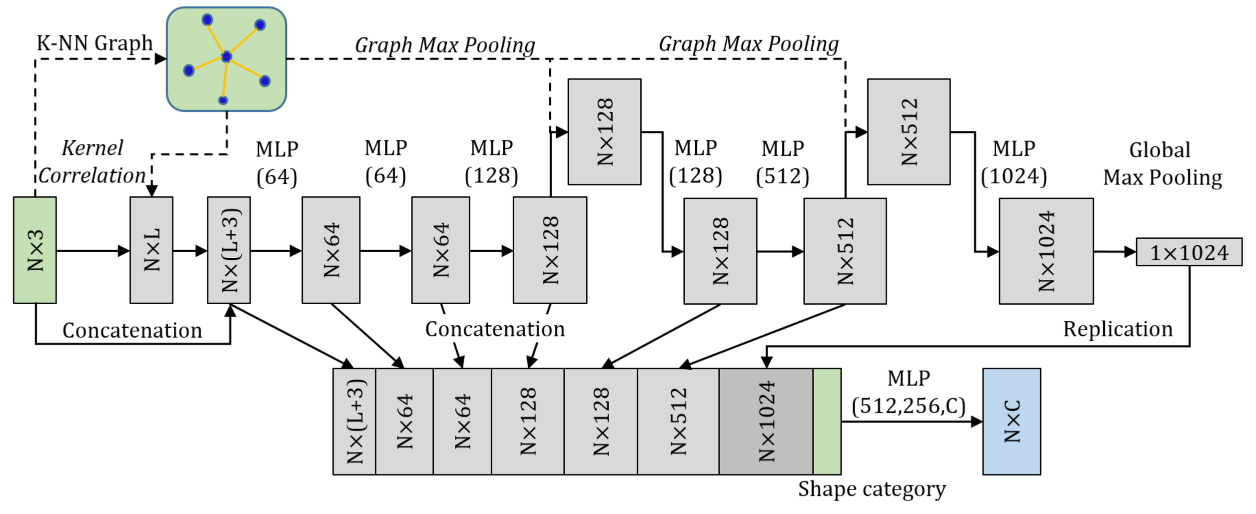
- Kernel correlation to extract local geometric structures.
- Graph max pooling to improve feature robustness.

KCNet Architecture

Classification Network



Segmentation Network



1. Extract Local Geometric Structures

We used Kernel Correlation to capture local geometric structures.

$$\mathrm{KC}(\kappa, x_i) = \frac{1}{|N(i)|}\sum_{m=1}^{M}\sum_{n\in N(i)} K_\sigma(\kappa_m, x_n - x_i),$$

$$K_\sigma(\mathrm{k}, \delta) = e^{\left(-\frac{\|k-\delta\|}{2\sigma^2}\right)},$$

where $N(i)$ indicates neighbors of x_i.

2. Robust Features by Local Aggregation

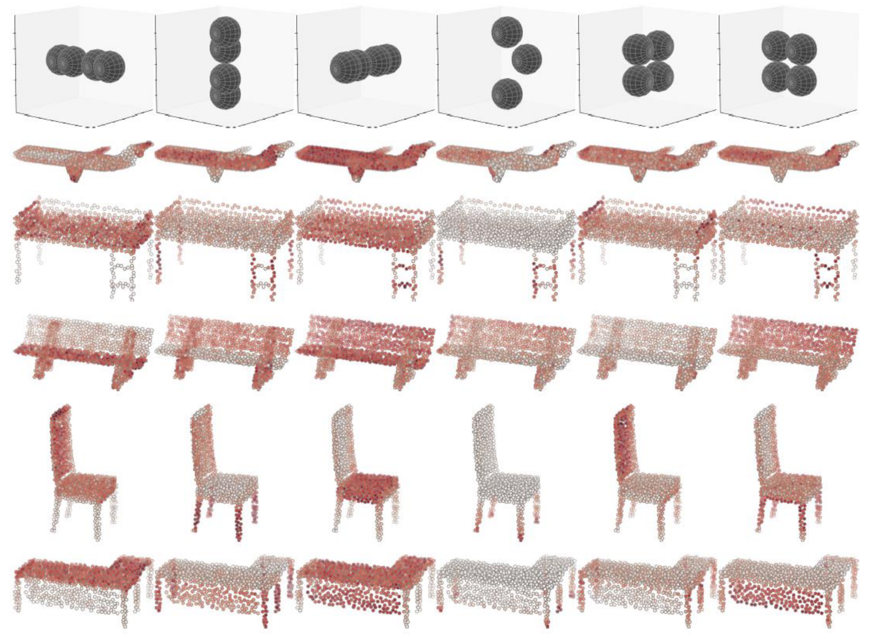
We used graph max pooling to each point within its neighborhood to improve feature robustness against noises.

$$Y(i,k) = \max_{n\in N(i)} X(n,k),$$

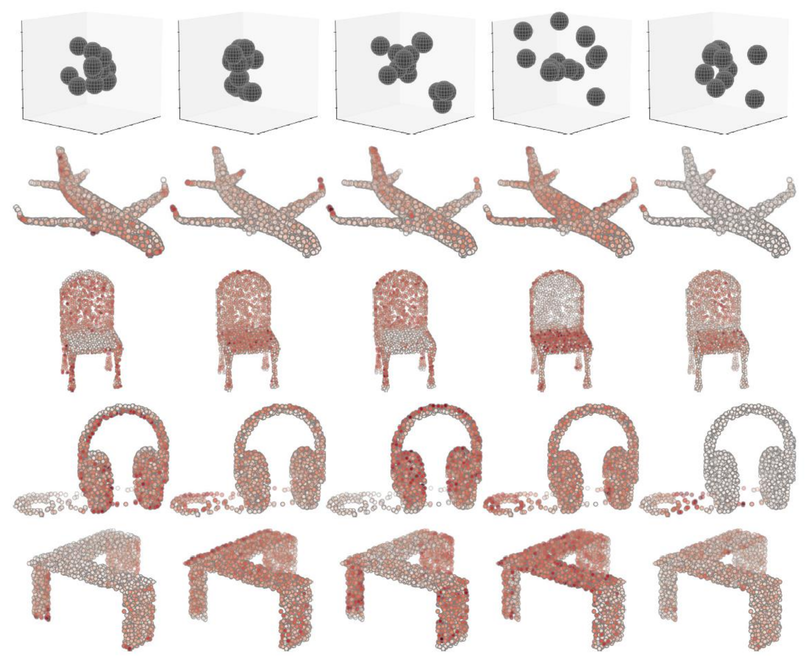
where $Y \in R^{N\times K}$ and $X \in R^{N\times K}$.

Visualization

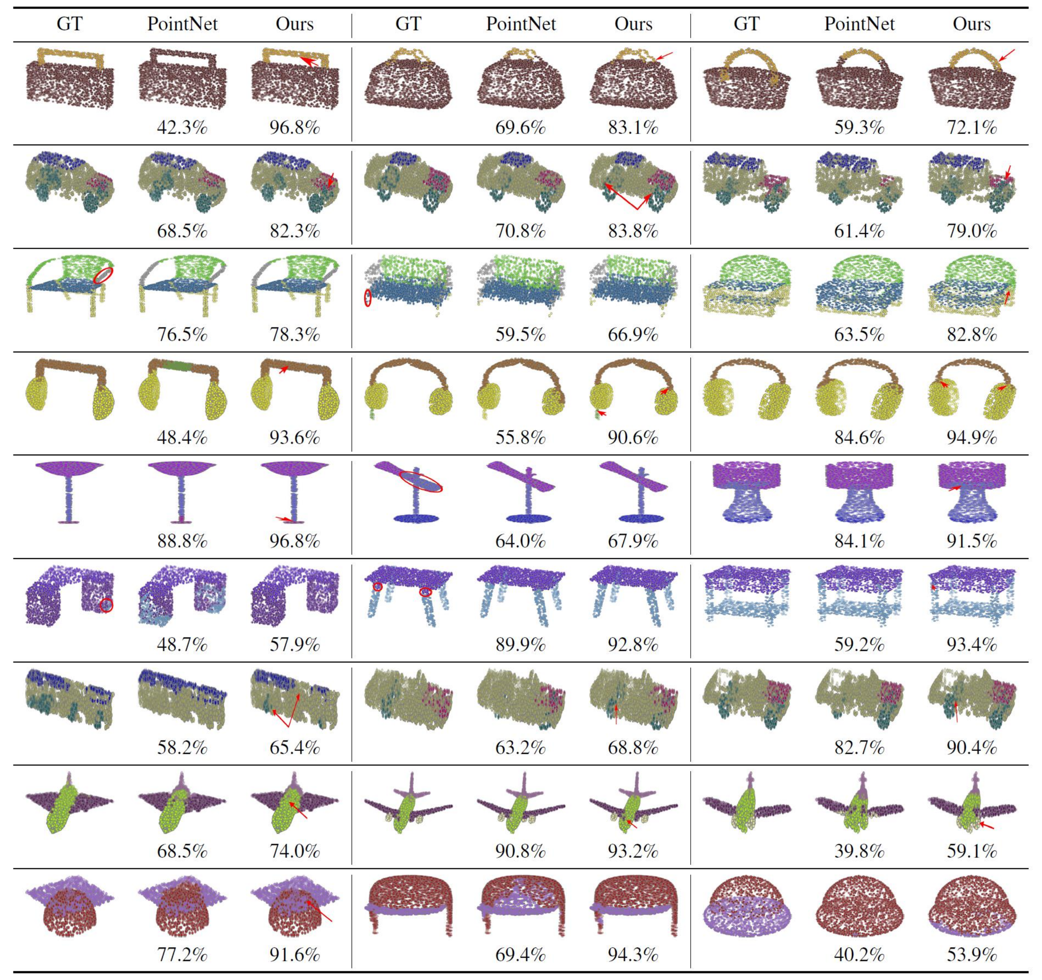
Handcrafted Kernels & Responses



Learned Kernels & Responses



ShapeNet Part Segmentation



Experimental Results

2D Digit Classification

Method	Accuracy (%)
LeNet5 [21]	99.2
PointNet (vanilla) [29]	98.7
PointNet [29]	99.2
PointNet++ [31]	**99.5**
KCNet (ours)	99.3

Table 1: Non-zero pixels in images from MNIST are converted to 2D point clouds.

3D Shape Classification

Method	MN10	MN40
MVCNN [36]	-	90.1
VRN Ensemble [2]	**97.1**	**95.5**
ECC [34]	90.0	83.2
PointNet (vanilla) [29]	-	87.2
PointNet [29]	-	89.2
PointNet++ [31]	-	90.7
Kd-Net(depth 10) [20]	93.3	90.6
Kd-Net(depth 15) [20]	94.0	91.8
KCNet (ours)	94.4	91.0

Table 2: Results on ModelNet.

3D Part Segmentation

	Cat. mIoU	Ins. mIoU	aero	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	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
PointNet++	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**
Kd-Net	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
KCNet (ours)	**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

Table 3: Part segmentation on ShapeNet. Average mIoU over instances (Ins.) and categories (Cat.) are reported.

Analysis Experiments

Robustness Test

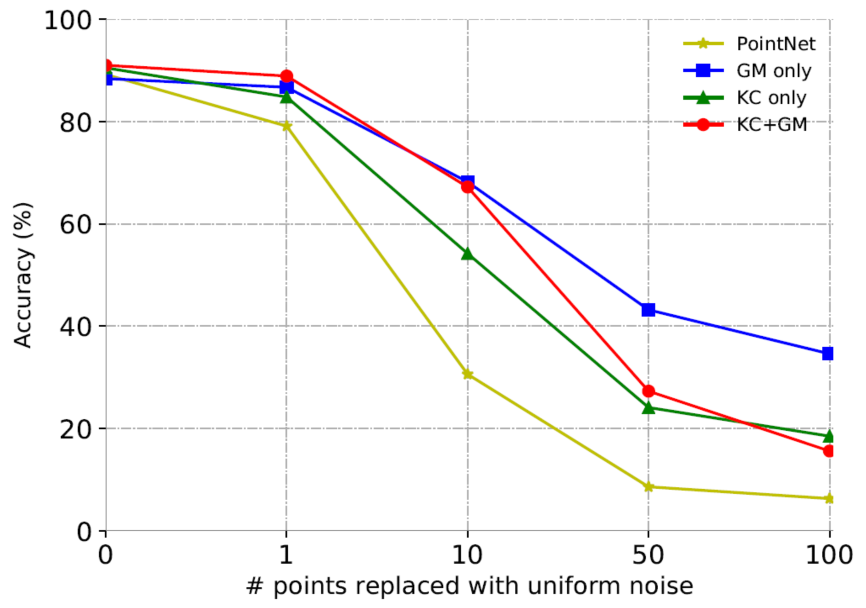


Figure 1: #points replaced with uniform noise between [-1,1]. GM only: graph max pooling only. KC only: kernel correlation only. KC+GM: both.

Ablation Study

Effectiveness of Kernel Correlation	Accuracy (%)
Normal	88.4
Kernel correlation	**90.5**
Symmetric Functions	Accuracy (%)
Graph average pooling	88.0
Graph max pooling	**88.6**
Effectiveness of Local Structures	Accuracy (%)
Baseline: PointNet (vanilla)	87.2
Kernel correlation (geometric)	90.5
Graph max pooling (feature)	88.6
Both	**91.0**

Model Size & Time

Method	#params (M)	Fwd. time (ms)
PointNet(vanilla) [31]	**0.8**	**11.6**
PointNet [31]	3.5	25.3
PointNet++(MSG) [31]	1.0	163.2
Kd-Net (depth 10)	2.0	-
KCNet ($M = 16$)	0.9	18.5
KCNet ($M = 3$)	0.9	12.0

Hyper-parameters

L	Acc. (%)	M	Acc. (%)	σ	Acc. (%)
16	90.7	3	90.9	1e−3	90.0
32	**91.0**	8	90.4	**5e−3**	**91.0**
48	91.0	**16**	**91.0**	1e−2	90.4