Dynamic Graph CNN for Learning on Point Clouds

Presenters: Long Jiajun, Zhang Ximan, Zheng Yikai, Luo Jie and Sinrithy Vong

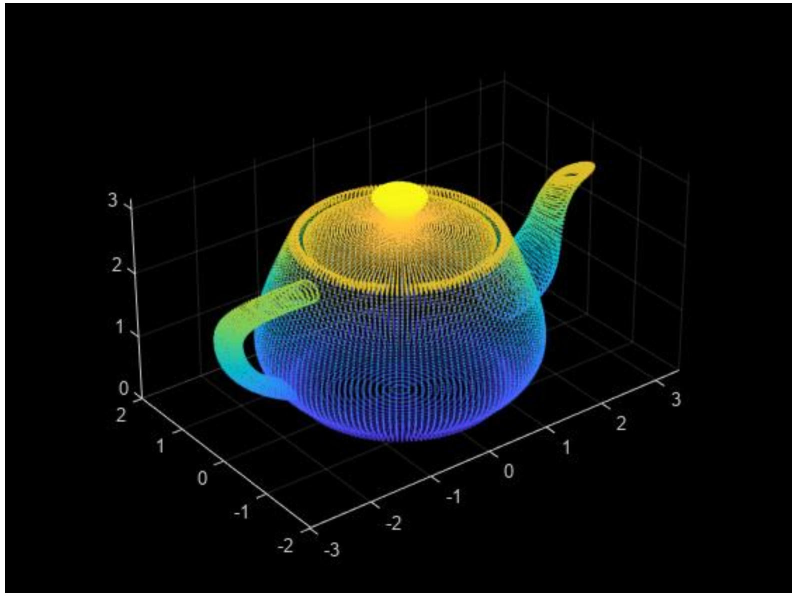
Date: March 30, 2023

AncoraSIR.com

Motivation and Main Problem

Point Cloud:

- Definition: a discrete set of data points in space to represent a 3D shape or object
- Problem: inherently lack topological information
- Application: Indoor navigation, self-driving vehicles, robotics, and shape synthesis and modeling



CNN: a class of artificial neural network most commonly applied to analyze visual imagery

AncoraSIR.com

Problem Setting

EdgeConv:

- Suitable for CNN-based high-level tasks on point clouds including classification and segmentation
- Captures local geometric structure while maintaining permutation invariance

Key contributions of their work in the paper:

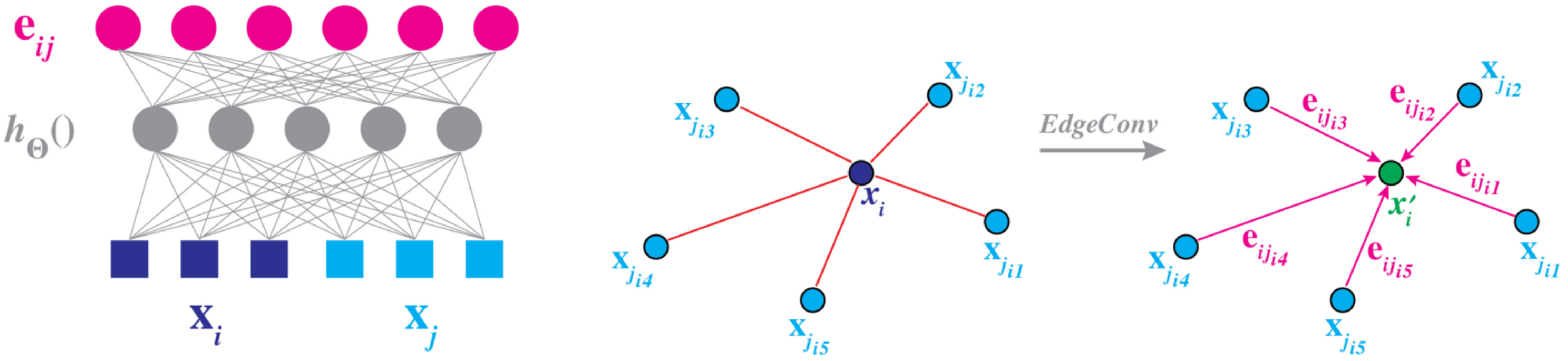
- Introduce EdgeConv
- Show that this model can learn to semantically group points by dynamically updating the graph from layer to layer
- Demonstrate that EdgeConv can be integrated into multiple existing pipelines for point cloud processing
- Do extensive analysis and testing on EdgeConv and show its performance on benchmark datasets
- Release codes to the public

Related Work and Limitations of Prior Work

1. Hand-Crafted Features: use various algorithms on the information present in the image itself (edges and corners detection)
2. Deep Learning on Geometry: use emerging techniques that generalize neural networks to Euclidean and non-Euclidean domains
3. Geometric Generative Models: attempt to generalize models such as autoencoders, variational autoencoders (VAE) and generative adversarial networks (GAN) to the non-Euclidean setting

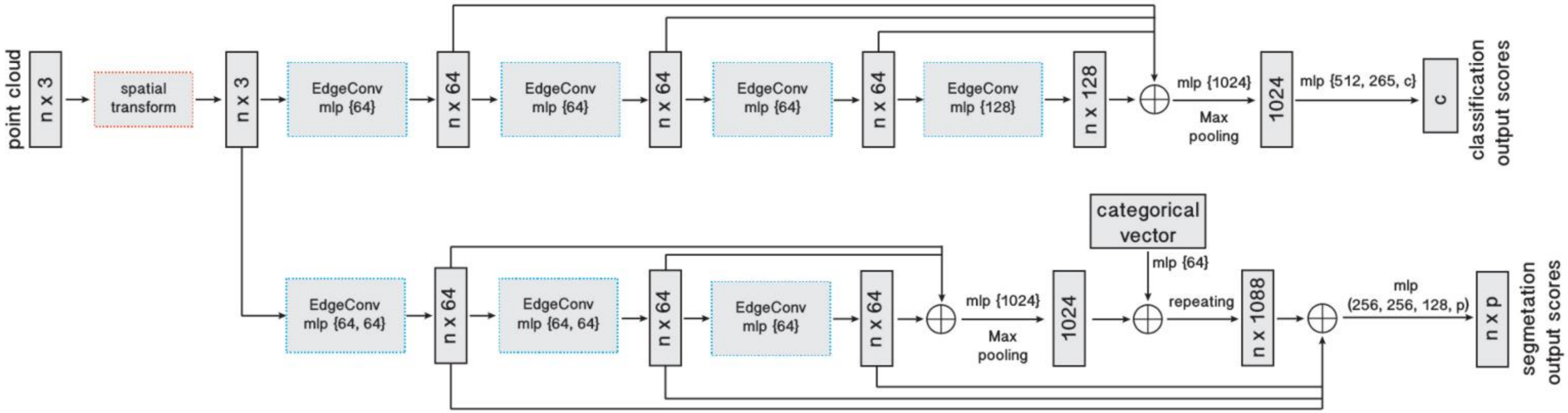
Key limitation of prior work: treat each point independently, thereby ignoring local geometric information

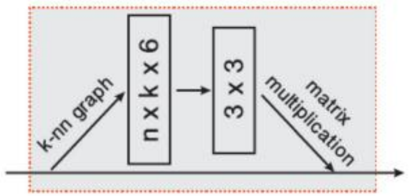
Edge Convolution



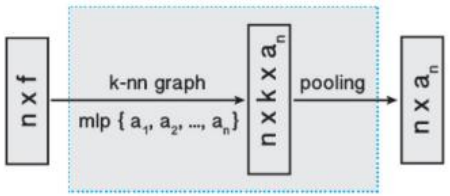
AncoraSIR.com

Edge Convolution





spatial transform



EdgeConv
mlp { a_1, a_2, ..., a_n }

AncoraSIR.com

Edge Convolution

Standard convolution	$x'_{im} = \sum_{j:(i,j)\in\mathcal{E}} \theta_m \cdot \mathbf{x}_j,$
used in PointNet	$h_\Theta(\mathbf{x}_i, \mathbf{x}_j) = h_\Theta(\mathbf{x}_i),$
adopted by Atzmon et al.	$h_\Theta(\mathbf{x}_i, \mathbf{x}_j) = h_\Theta(\mathbf{x}_j)$ $\quad x'_{im} = \sum_{j\in\mathcal{V}} (h_{\theta(\mathbf{x}_j)}) g(u(\mathbf{x}_i, \mathbf{x}_j)),$
only local information	$h_\Theta(\mathbf{x}_i, \mathbf{x}_j) = h_\Theta(\mathbf{x}_j - \mathbf{x}_i).$

an asymmetric edge function

$$h_\Theta(\mathbf{x}_i, \mathbf{x}_j) = \bar{h}_\Theta(\mathbf{x}_i, \mathbf{x}_j - \mathbf{x}_i).$$

$$e'_{ijm} = \mathrm{ReLU}(\theta_m \cdot (\mathbf{x}_j - \mathbf{x}_i) + \phi_m \cdot \mathbf{x}_i),$$

$$x'_{im} = \max_{j:(i,j)\in\mathcal{E}} e'_{ijm},$$

AncoraSIR.com

Dynamic graph update

1. With dynamic graph updates, the receptive field is as large as the diameter of the point cloud, while being sparse.

2. Compute a pairwise distance matrix in feature space and then take the closest k points for each single point

Properties

$$\mathbf{x}'_i = \max_{j:(i,j)\in\mathcal{E}} h_{\Theta}(\mathbf{x}_i, \mathbf{x}_j)$$

Permutation Invariance

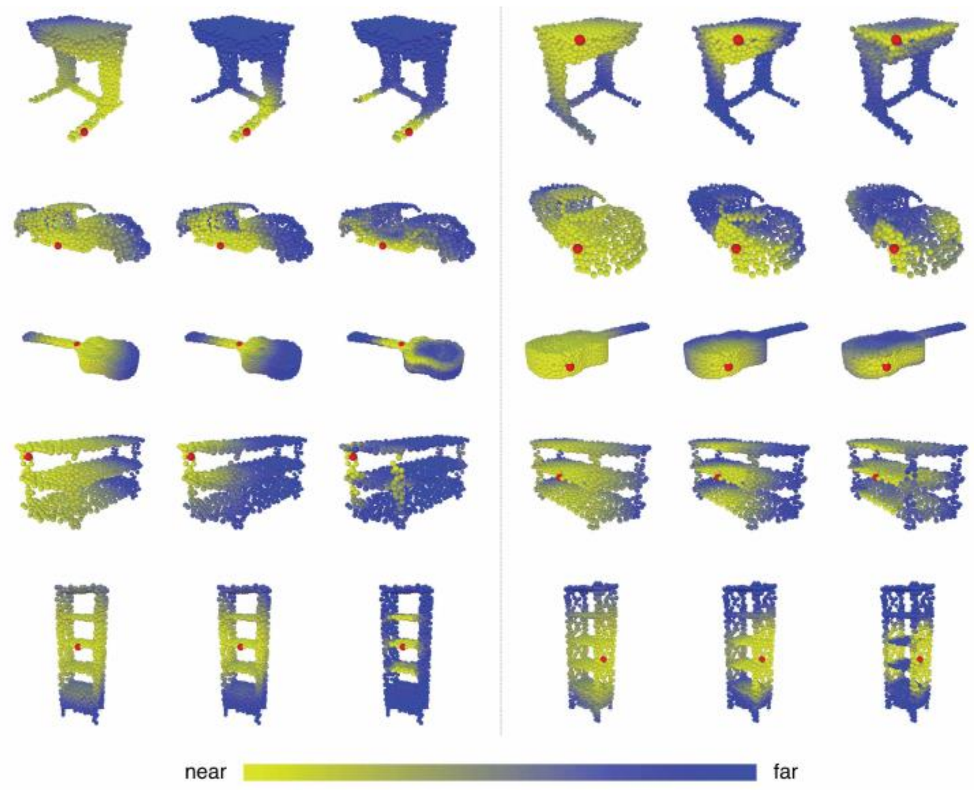
Translation Invariance

$$\begin{aligned} e'_{ijm} &= \theta_m \cdot (\mathbf{x}_j + T - (\mathbf{x}_i + T)) + \phi_m \cdot (\mathbf{x}_i + T) \\ &= \theta_m \cdot (\mathbf{x}_j - \mathbf{x}_i) + \phi_m \cdot (\mathbf{x}_i + T). \end{aligned}$$

AncoraSIR.com

Comparison to existing methods

	Aggregation	Edge Function	Learnable parameters
PointNet [Qi et al. 2017b]	—	$h_\Theta(\mathbf{x}_i, \mathbf{x}_j) = h_\Theta(\mathbf{x}_i)$	Θ
PointNet++ [Qi et al. 2017c]	max	$h_\Theta(\mathbf{x}_i, \mathbf{x}_j) = h_\Theta(\mathbf{x}_j)$	Θ
MoNet [Monti et al. 2017a]	$\sum$	$h_{\theta_m, w_n}(\mathbf{x}_i, \mathbf{x}_j) = \theta_m \cdot (\mathbf{x}_j \odot g_{w_n}(u(\mathbf{x}_i, \mathbf{x}_j)))$	w_n, θ_m
PCNN [Atzmon et al. 2018]	$\sum$	$h_{\theta_m}(\mathbf{x}_i, \mathbf{x}_j) = (\theta_m \cdot \mathbf{x}_j) g(u(\mathbf{x}_i, \mathbf{x}_j))$	θ_m



AncoraSIR.com

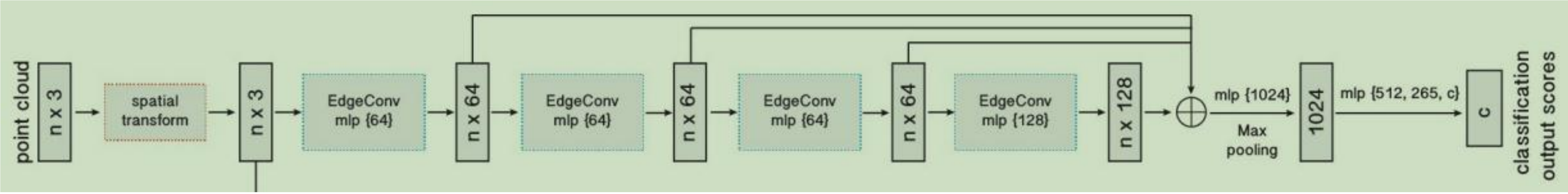
Evaluation

1.Classification

Data source: ModelNet40 [Wu et al. 2015]

Augment the data by randomly scaling objects and perturbing the object and point locations.

Architecture:



The model architectures used for classification
nx3,nx64......: 边缘特征响应/边缘特征
Max pooling：取局部接受域中值最大的点

AncoraSIR.com

Evaluation

1.Classification

Training: momentum for batch normalization : 0.9 batch size : 32 momentum: 0.9.

Result:

	Mean Class Accuracy	Overall Accuracy
3DShapeNets [Wu et al. 2015]	77.3	84.7
VoxNet [Maturana and Scherer 2015]	83.0	85.9
Subvolume [Qi et al. 2016]	86.0	89.2
VRN (single view) [Brock et al. 2016]	88.98	-
VRN (multiple views) [Brock et al. 2016]	91.33	-
ECC [Simonovsky and Komodakis 2017]	83.2	87.4
PointNet [Qi et al. 2017b]	86.0	89.2
PointNet++ [Qi et al. 2017c]	-	90.7
Kd-net [Klokov and Lempitsky 2017]	-	90.6
PointCNN [Li et al. 2018a]	88.1	92.2
PCNN [Atzmon et al. 2018]	-	92.3
Ours (baseline)	88.9	91.7
Ours	**90.2**	**92.9**
Ours (2048 points)	**90.7**	**93.5**

Table 2. Classification results on ModelNet40.

AncoraSIR.com

Evaluation

2.Model Complexity

Using the ModelNet40 [Wu et al. 2015] classification experiment
to compare the complexity of our model to previous state-of-the-art.

	Model size(MB)	Time(ms)	Accuracy(%)
PointNet (Baseline) [Qi et al. 2017b]	9.4	6.8	87.1
PointNet [Qi et al. 2017b]	40	16.6	89.2
PointNet++ [Qi et al. 2017c]	12	163.2	90.7
PCNN [Atzmon et al. 2018]	94	117.0	92.3
Ours (Baseline)	11	19.7	91.7
Ours	21	27.2	92.9

Table 3. Complexity, forward time, and accuracy of different models

Model complexity: number of parameters
Baseline: fixed k-NN graph

Computational complexity:time consumed

AncoraSIR.com

Evaluation

3.More experiments on dataset ModelNet40

Analyzing the effectiveness of different distance metric.

CENT	DYN	MPOINTS	MEAN CLASS ACCURACY(%)	OVERALL ACCURACY(%)
			88.9	91.7
x			89.3	92.2
x	x		90.2	92.9
x	x	x	90.7	93.5

CENT: using concatenation of xi and xi − xj as the edge features rather than concatenating xi and xj
DYN:Dynamic graph recomputation instead of a fixed graph
MPOINTS:denoteing experiments with 2048 points

Evaluation

3.More experiments on dataset ModelNet40

Experimenting with different numbers k of nearest neighbors

Number of nearest neighbors (k)	Mean Class Accuracy(%)	Overall Accuracy(%)
5	88.0	90.5
10	88.9	91.4
20	90.2	92.9
40	89.4	92.4

Table 5. Results of our model with different numbers of nearest neighbors.

AncoraSIR.com

Evaluation

3.More experiments on dataset ModelNet40
The robustness of the model

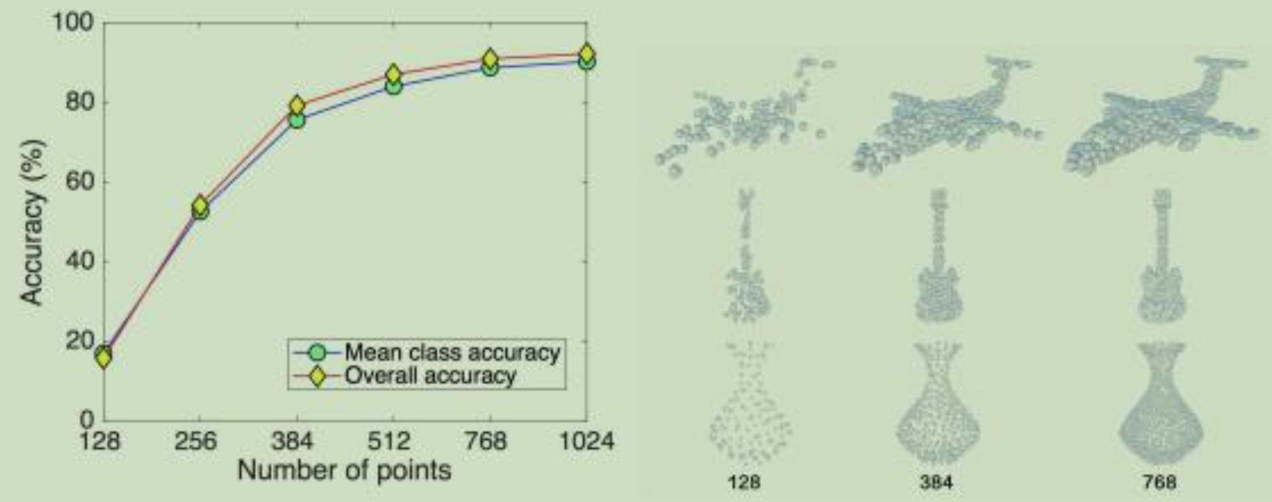


Fig. 5. **Left:** Results of our model tested with random input dropout. The model is trained with number of points being 1024 and k being 20. **Right:** Point clouds with different number of points. The numbers of points are shown below the bottom row.

Model trained with 1024 points and k=20.   Simulate the environment that random input points drops out during testing.

AncoraSIR.com

Part Segmentation

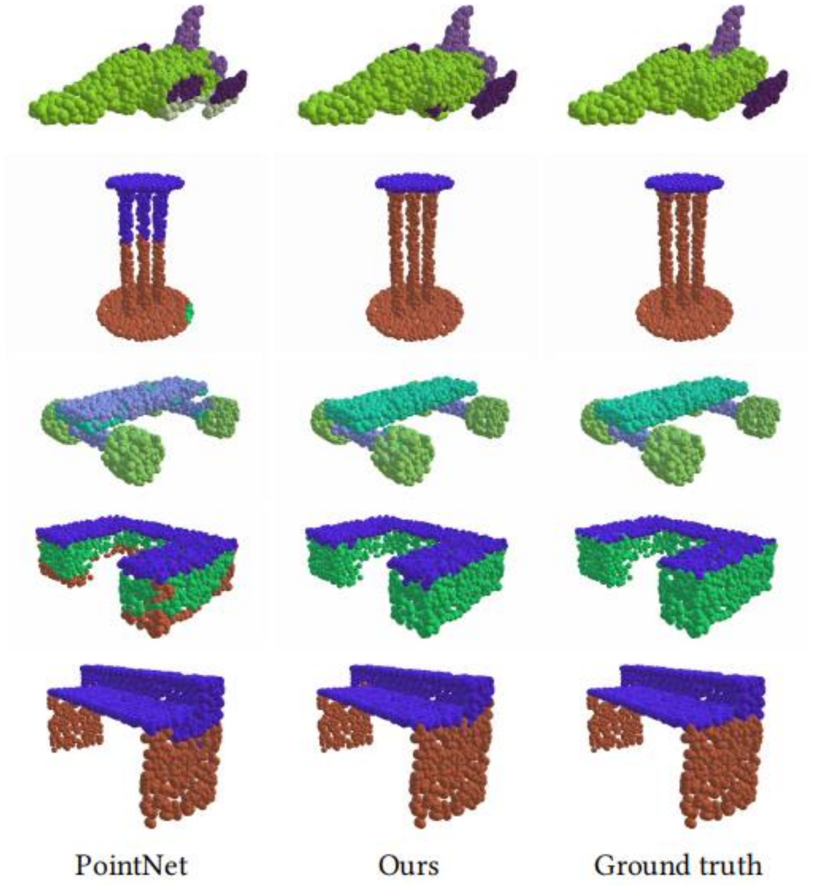
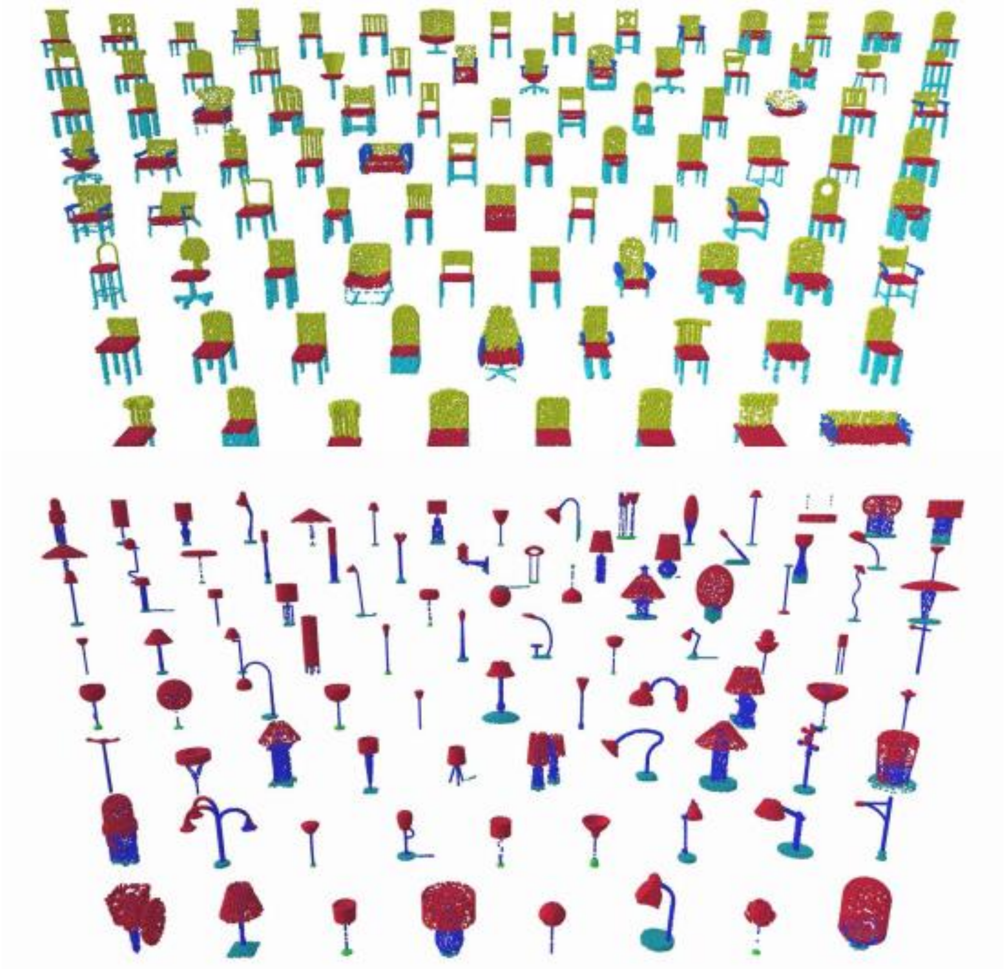


Fig. 7. Compare part segmentation results. For each set, from left to right: PointNet, ours and ground truth.

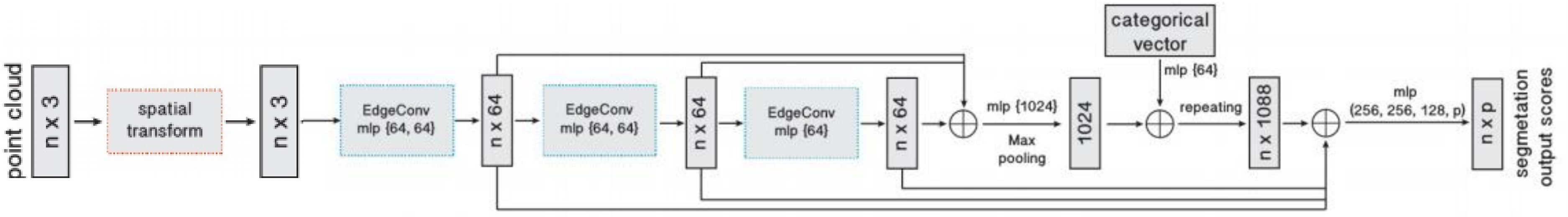


AncoraSIR.com

Data

dataset:ShapeNet part dataset scheme:train-validation-test split

Architecture



Training

the same training setup as for our classification task was used.

AncoraSIR.com

Evaluation

Intersection-over-Union on points

The IoU of a shape is computed by averaging the IoUs of different parts occurring in that shape.

The IoU of a category is obtained by averaging the IoUs of all the shapes belonging to that category.

The mean IoU (mIoU) is finally calculated by averaging the IoUs of all the testing shapes.

Results

	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	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++	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	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
LocalFeatureNet	84.3	**86.1**	73.0	54.9	77.4	88.8	55.0	90.6	86.5	75.2	**96.1**	57.3	91.7	83.1	53.9	72.5	**83.8**
PCNN	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
PointCNN	**86.1**	84.1	**86.45**	86.0	**80.8**	90.6	**79.7**	**92.3**	**88.4**	**85.3**	**96.1**	**77.2**	**95.3**	**84.2**	**64.2**	**80.0**	83.0
Ours	85.2	84.0	83.4	86.7	77.8	90.6	74.7	91.2	87.5	82.8	95.7	66.3	94.9	81.1	63.5	74.5	82.6

Table 6. Part segmentation results on ShapeNet part dataset. Metric is mIoU(%) on points.

SUSTech
Southern University of Science and Technology

Future Work for Paper / Reading

Paper mentioned:

Incorporating fast data structures rather than pairwise to calculate the distance

Design non-shared transformer network

To more abstract point clouds – document retrieval/ image processing

AncoraSIR.com

SUSTech
Southern University
of Science and Technology

Future Work for Paper / Reading

Future work:

- Our ideas：

a. Using this catching method in more specific situation(Ex, in sweeping robot, etc)

b. Revise the details to improve efficiency or scalability

- Others' ideas:

a. A Plug-and-Play for 3D Point Clouds

b. Unsupervised Structural Representation Learning of 3D Point Clouds.

c. Semi-supervised 3D shape segmentation with multilevel consistency and part substitution.

AncoraSIR.com

Extended Readings

Extended paper 1:

- Point cloud completion via structured feature maps using a feedback network

- Tackle the challenging problem of point cloud completion from the perspective of feature learning

- Link: https://link.springer.com/article/10.1007/s41095-022-0276-6

From: Web of Science

AncoraSIR.com

Extended Readings

Extended paper 2:

- MVGCN: Multi-View Graph Convolutional Neural Network for Surface Defect Identification Using Three-Dimensional Point Cloud

- detect and classify surface defects using a 3D point cloud

- The proposed approach consists of an unsupervised method for defect detection and a multi-view deep learning model for defect classification, which can keep track of the features from both defective and non-defective regions. We prove that the proposed approach is invariant to different permutations and transformations.

- Link:https://asmedigitalcollection.asme.org/manufacturingscience/article/145/3/031004/1148268/MVGCN-Multi-View-Graph-Convolutional-Neural

AncoraSIR.com

From: Web of Science

Summary

Why?

inherently lack topological information

Limit:

(1) treat each point independently

(2) ignoring local geometric information

Innovative point:

(1) Suitable for CNN-based high-level tasks on point clouds including classification and segmentation

(2) Captures local geometric structure while maintaining permutation invariance

Proved:

(1) Demonstrate that EdgeConv can be integrated into multiple existing pipelines for point cloud processing

(2) Do extensive analysis and testing on EdgeConv and show its performance on benchmark datasets