SYDE 675
FINAL PROJECT

4/19/2022

Yikai Yao, Zhuxian Ding

Systems Design Engineering

UNIVERSITY OF WATERLOO | FACULTY OF ENGINEERING

Photo credit: @bruce.digital

Convolution in the Cloud: Learning Deformable Kernels in 3D Graph Convolution Networks for Point Cloud Analysis

Conference: CVPR 2020

Paper: https://openaccess.thecvf.com/content_CVPR_2020/papers/Lin_Convolution_in_the_Cloud_Learning_Deformable_Kernels_in_3D_Graph_CVPR_2020_paper.pdf

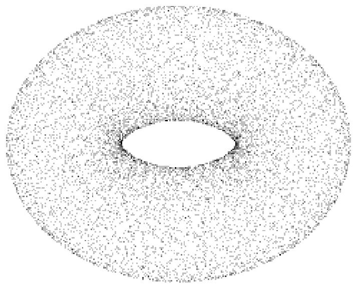
Code: https://github.com/j1a0m0e4sNTU/3dgcn

Outline

- Introduction
- Theory behind the paper
- Some preliminary results
- Other ML techniques
- Limitation and improvement

What is point cloud?

- Point cloud is a set of data points in space, representing 3D shapes or objects.
- Each point is represented by a three-dimensional coordinates (x, y, z).
- Point cloud is stored as a $N \times 3$ matrix. (N: point number, 3: coordinates)

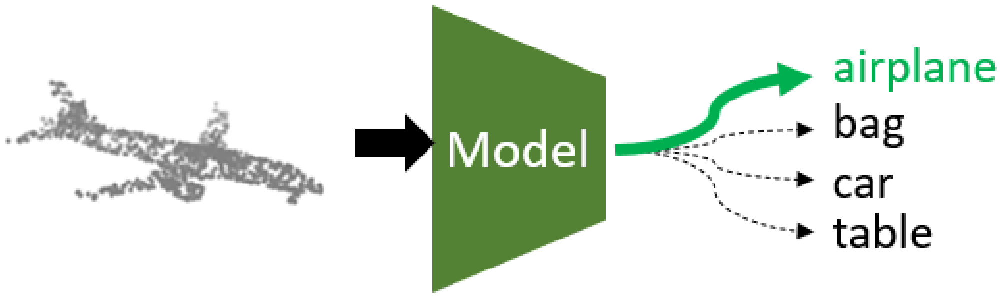


$$\begin{array}{cc} & 3 \\ N & \begin{bmatrix} (x_1, y_1, z_1) \\ (x_2, y_2, z_2) \\ \\ \\ (x_N, y_N, z_N) \end{bmatrix} \end{array}$$

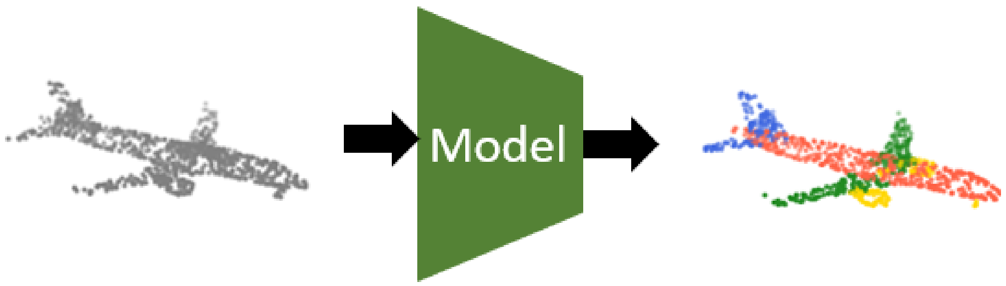
Reference: [CVPR2020] Convolution in the Cloud
https://www.youtube.com/watch?v=xfftSRFIWY0

Overview

- Learn to recognize geometric patterns of point cloud shapes.
- Propose a model for classification, semantic segmentation tasks.



Classification

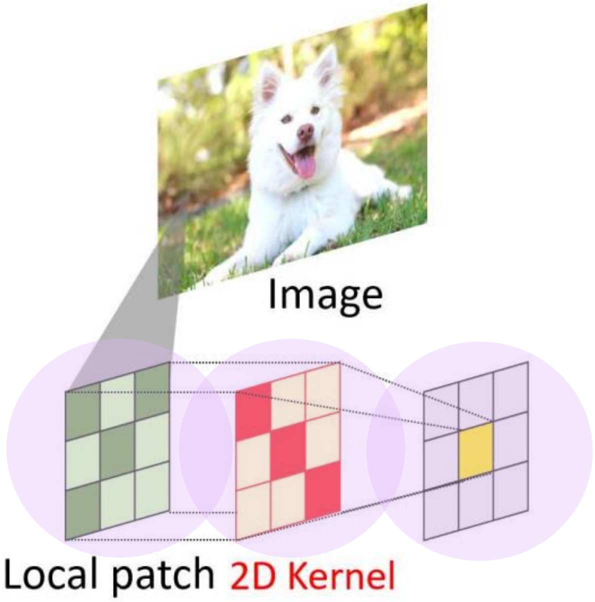


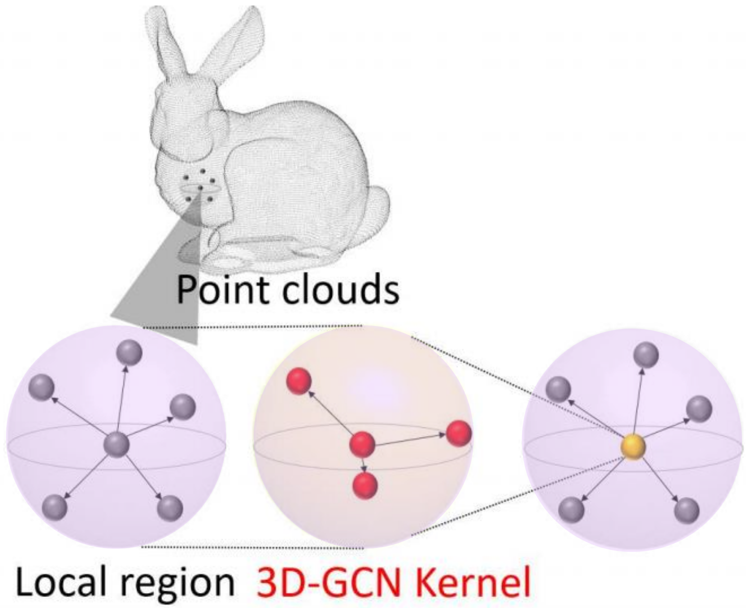
Segmentation

Method

- A robust framework for 3D point clouds analysis:

  3D Graph Convolution Networks

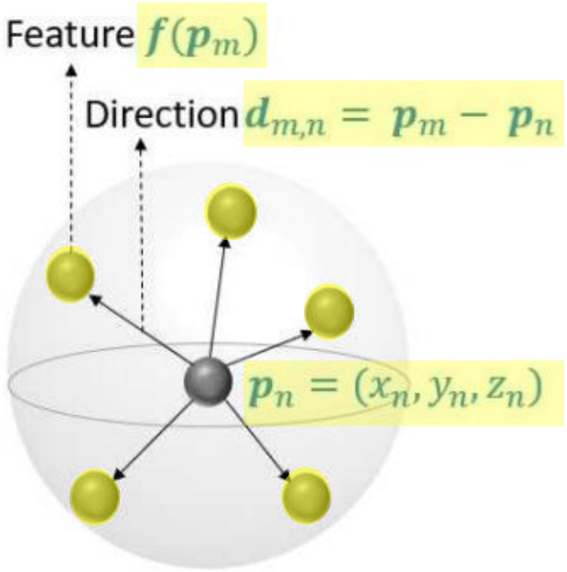




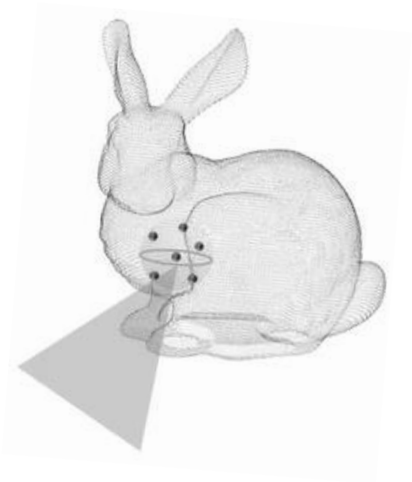
3D Graph Convolution – Receptive Fields

- Receptive field is composed of central point p_n and its M nearest neighbors.
- Each point p is represented by location $(x, y, z) \in R^3$ and feature $f(p) \in R^c$.

✓ Represent local geometry



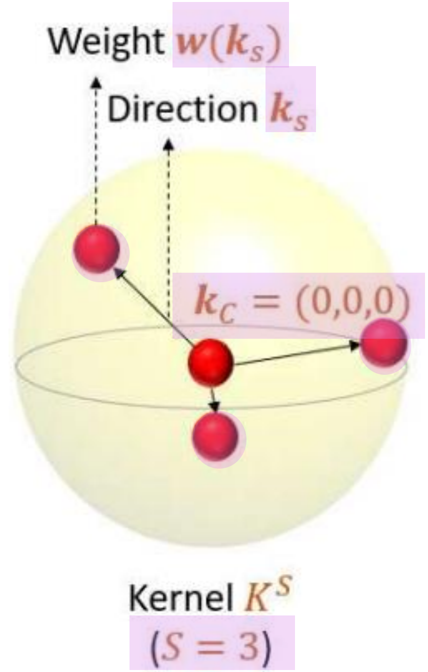
Receptive field R_n^M of point p_n $(M = 5)$



3D Graph Convolution – Kernel

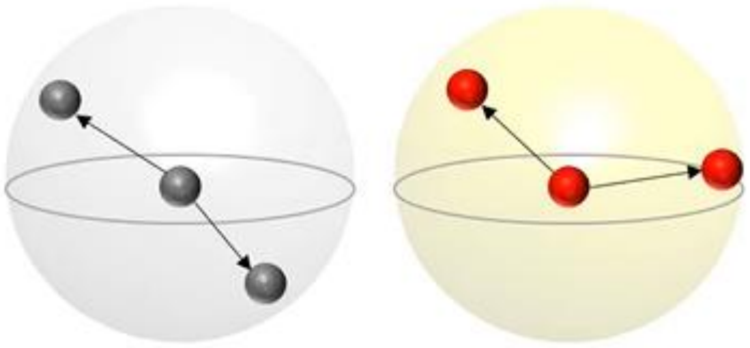
- One kernel has one central node and S support nodes,
- Each kernel point is represented by position $k \in R^3$ and weight vector $w(k) \in R^c$.
- k and $w(k)$ are learnable in training.

✓ Deformable kernels
✓ Learn to describe local geometry



3D Graph Convolution

- Similarity between R_n^M and K^S .

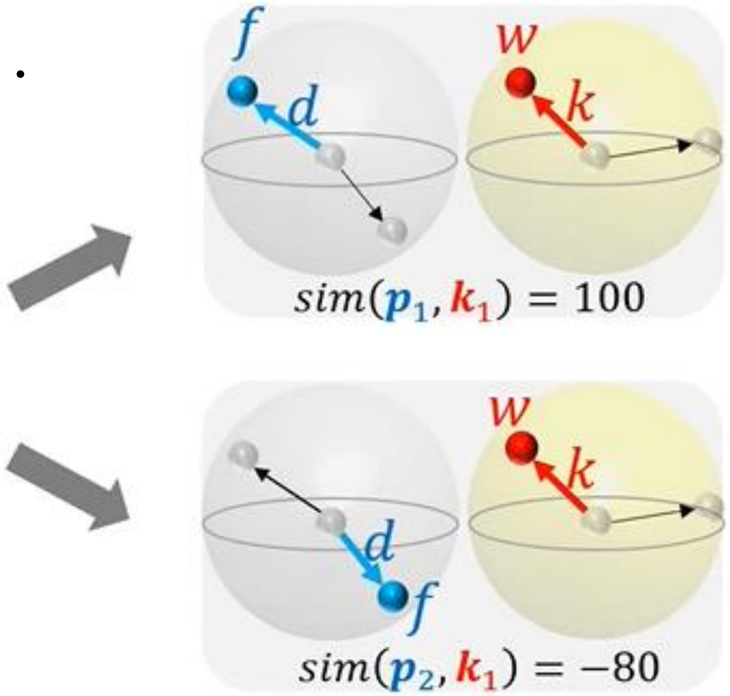


Receptive Fields

R_n^M

Kernel

K^S

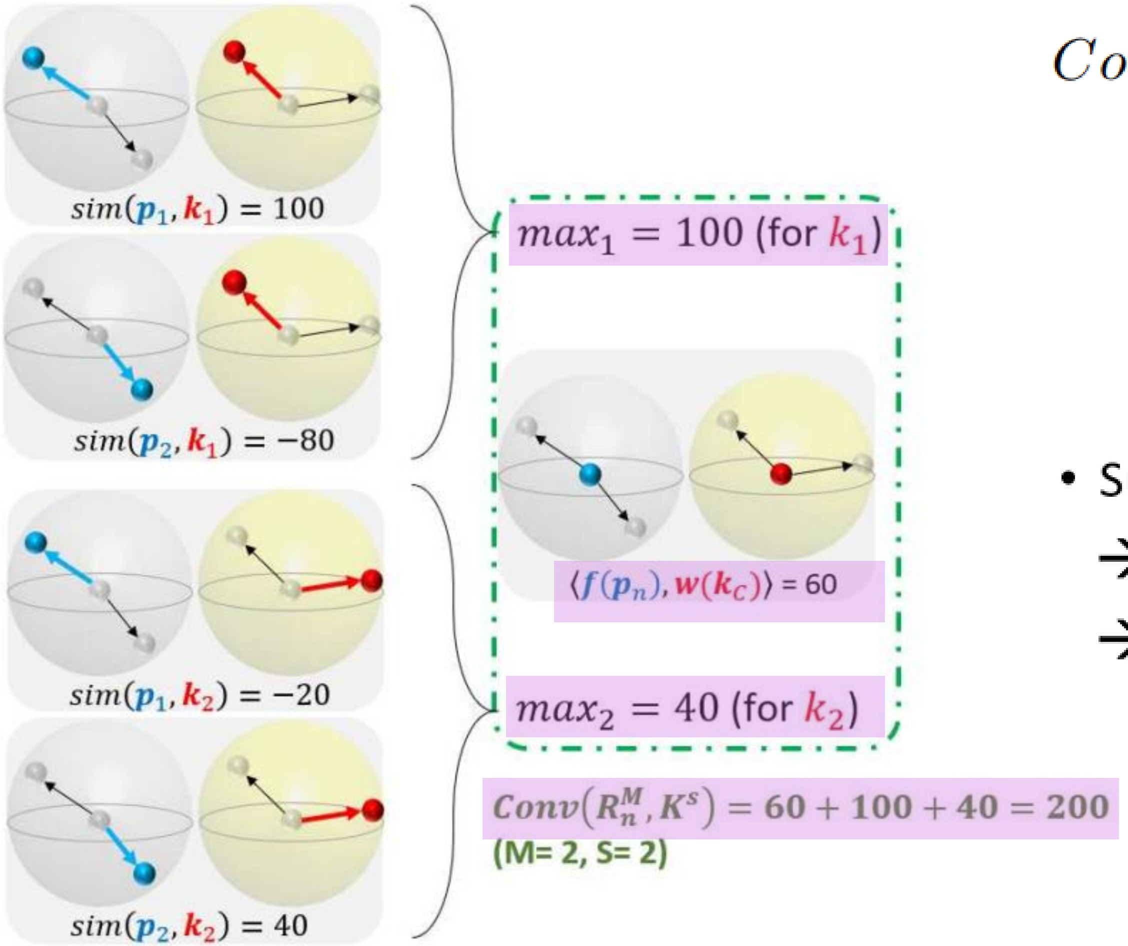


Same direction
→ similar

Opposite direction
→ Not similar

Convolve with cosine similarity: $sim(\boldsymbol{p}_m, \boldsymbol{k}_s) = \underbrace{\langle \boldsymbol{f}(\boldsymbol{p}_m), \boldsymbol{w}(\boldsymbol{k}_s)\rangle}_{\text{Inner product}} \boxed{\frac{\langle \boldsymbol{d}_{m,n}, \boldsymbol{k}_s\rangle}{\|\boldsymbol{d}_{m,n}\| \|\boldsymbol{k}_s\|}}$

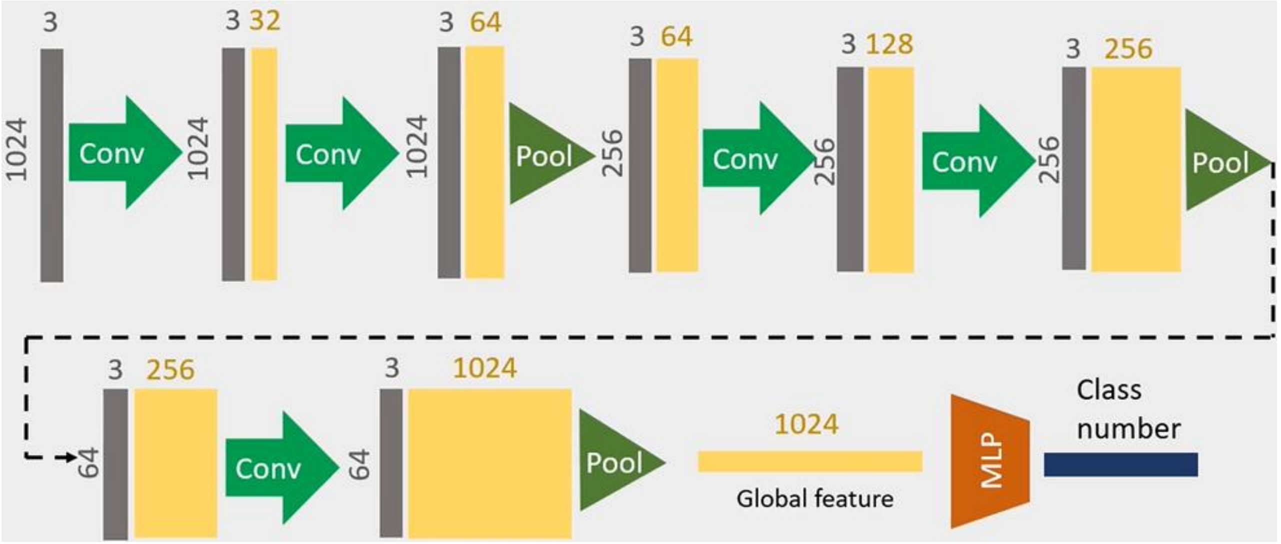
3D Graph Convolution



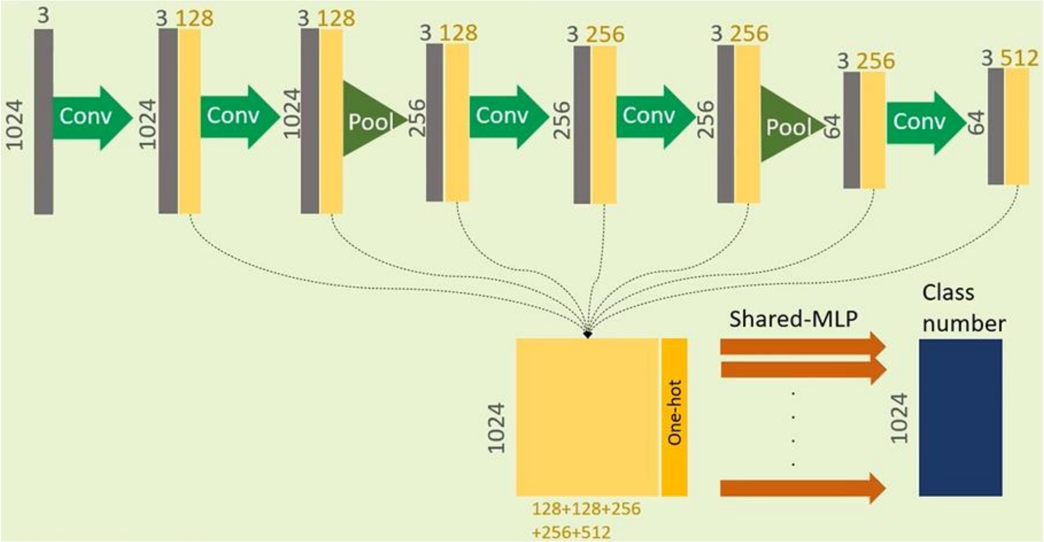
$$Conv(R_n^M, K^S) = \langle \boldsymbol{f}(\boldsymbol{p}_n), \boldsymbol{w}(\boldsymbol{k}_C) \rangle + \sum_{s=1}^{S} \max_{m \in (1,M)} \left\{ sim(\boldsymbol{p}_m, \boldsymbol{k}_s) \right\}$$

- Sums up all
  → weighted-sum
  → convolution value $Conv()$

Model structure – Classification, Segmentation



Classification



Segmentation

- Dataset: **ModelNet40**
- 40 object categories
- 1024 points from the surface for training and testing

- Dataset: **ShapeNetPart**
- 2-6 parts for 16 objects, 50 parts in total
- 1024 points are used for training and testing

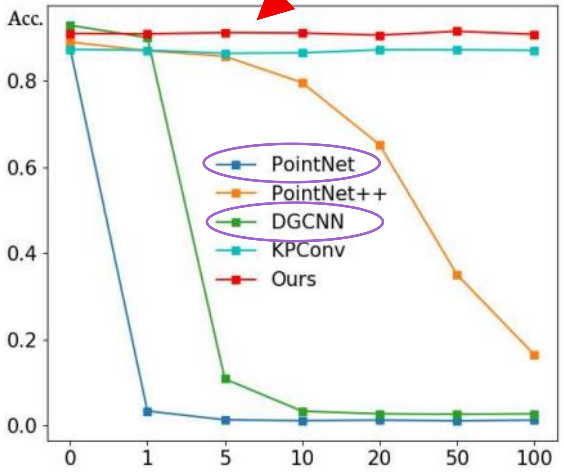
Results – Classification

- Authors' Result

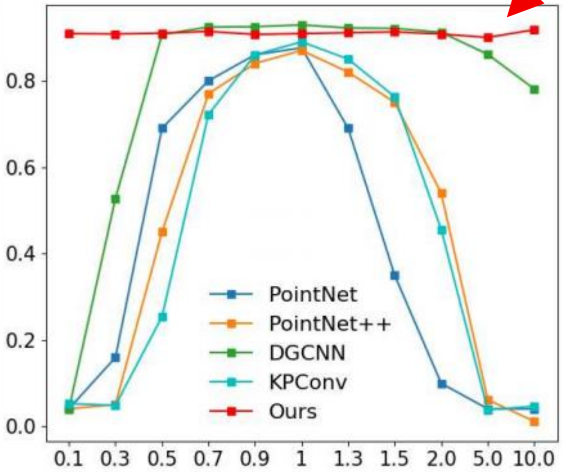
Classification results on ModelNet40

Method	input	#points	Acc.(%)
ECC [27]	xyz	1k	87.4
PointNet [21]	xyz	1k	89.2
Kd-Net (depth=10) [10]	xyz	1k	90.6
PointNet++ [23]	xyz	1k	90.7
KCNet [26]	xyz	1k	91.0
MRTNet [5]	xyz	1k	91.2
DGCNN [32]	xyz	1k	92.9
SO-Net [12]	xyz	2k	90.9
KPConv rigid [30]	xyz	6.8k	92.9
PointNet++ [23]	xyz, normal	5k	91.9
SO-Net [12]	xyz, normal	5k	93.4
Ours	xyz	1k	92.1

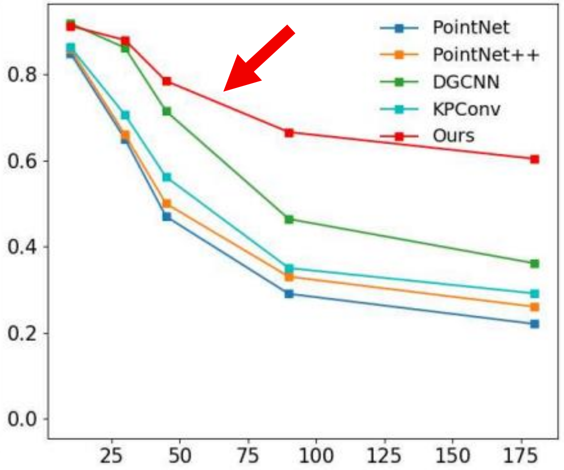
Evaluation of invariance properties on ModelNet40



Shift

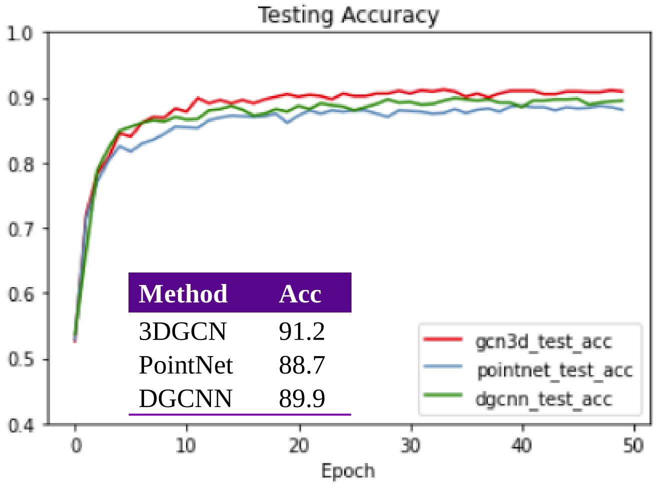


Scale

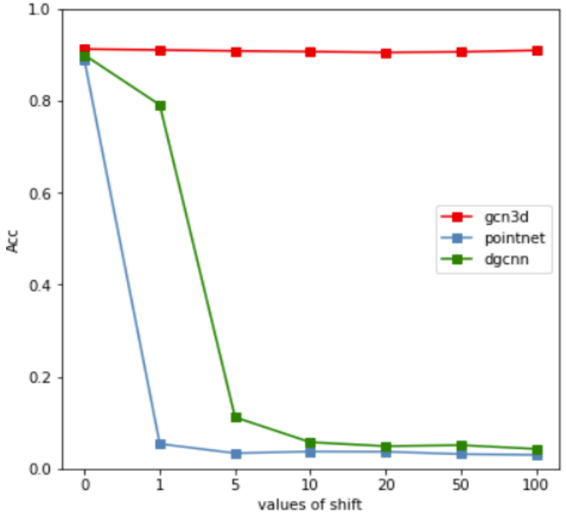


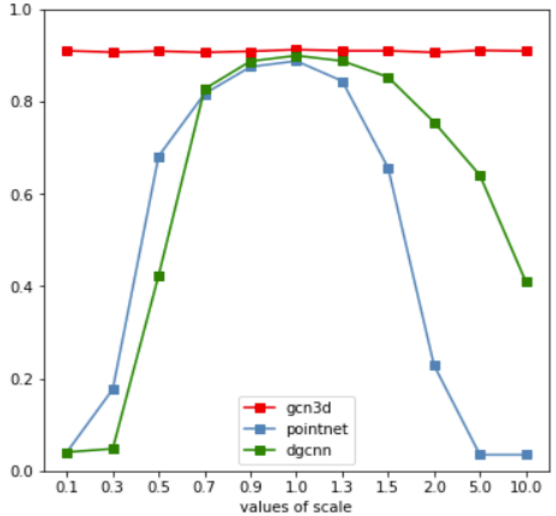
Rotation

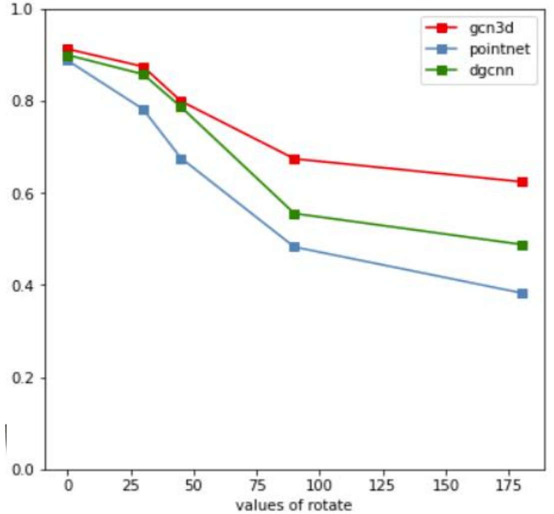
- Replicated Result



Method	Acc
3DGCN	91.2
PointNet	88.7
DGCNN	89.9



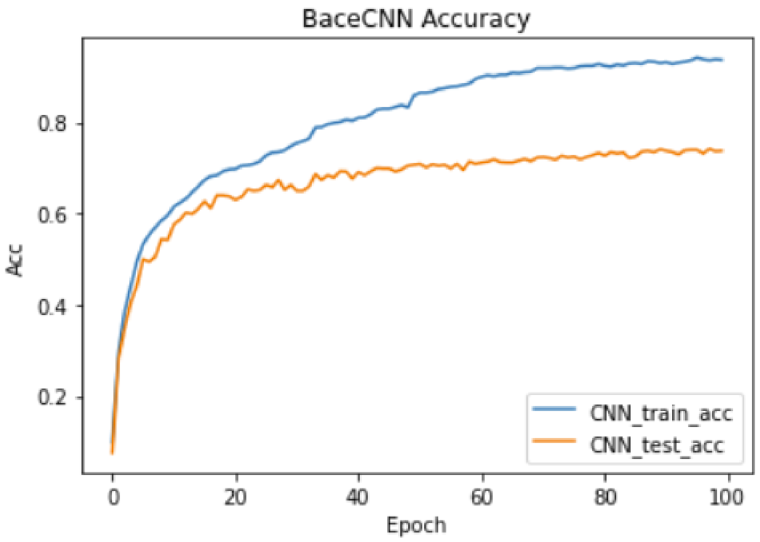




Machine Learning Algorithms – Classification

- Point clouds inherently lack topological information.
- The models used for Point cloud classification need to recover the topology.
- CNN

  - 100 epochs
  - train accuracy: 0.940
  - test accuracy: 0.742




```
Model parameter number: 626856
Model structure:
BaseCNN(
  (layer1): Sequential(
    (0): Conv2d(6, 64, kernel_size=(1, 1), stride=(1, 1))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=(2, 2), stride=(2, 2), padding=0, dilat
  )
  (layer2): Sequential(
    (0): Conv2d(128, 64, kernel_size=(1, 1), stride=(1, 1))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=(2, 2), stride=(2, 2), padding=0, dilat
  )
  (layer3): Sequential(
    (0): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=(2, 2), stride=(2, 2), padding=0, dilat
  )
  (layer4): Sequential(
    (0): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=(2, 2), stride=(2, 2), padding=0, dilat
  )
  (layer5): Sequential(
    (0): Conv1d(256, 1024, kernel_size=(1,), stride=(1,))
    (1): ReLU()
  )
  (fc1): Linear(in_features=1024, out_features=256, bias=True)
  (dropout): Dropout(p=0.2, inplace=False)
  (fc2): Linear(in_features=256, out_features=40, bias=True)
)
```

Preliminary Results – Segmentation

Segmentation Results on ShapeNetPart

method	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 [10]	77.4	82.3	80.1	74.6	74.3	70.3	88.6	73.5	90.2	87.2	81.0	84.9	87.4	86.7	78.1	51.8	69.9	80.3
MRTNet [5]	79.3	83.0	81.0	76.7	87.0	73.8	89.1	67.6	90.6	85.4	80.6	95.1	64.4	91.8	79.7	87.0	69.1	80.6
PointNet [21]	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
KCNet [26]	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
RS-Net [8]	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
SO-Net [12]	81.0	84.9	82.8	77.8	88.0	77.3	90.6	73.5	90.7	83.9	82.8	94.8	69.1	94.2	80.9	53.1	72.9	83.0
PointNet++ [23]	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
DGCNN [32]	82.3	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
KPConv deform [30]	85.1	86.4	84.6	86.3	87.2	81.1	91.1	77.8	92.6	88.4	82.7	96.2	78.1	95.8	85.4	69.0	82.0	83.6
Ours	82.1	85.1	83.1	84.0	86.6	77.5	90.3	74.1	90.9	86.4	83.8	95.6	66.8	94.8	81.3	59.6	75.7	82.8

Authors' Result

method	class mIOU	instance mIOU	air plane	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor bike	mug	pistol	rocket	skate board	table
3dgcn(ours)	0.810	0.843	0.810	0.817	0.811	0.762	0.898	0.794	0.903	0.878	0.825	0.953	0.619	0.938	0.806	0.567	0.764	0.827
pointnet	0.754	0.823	0.806	0.647	0.742	0.715	0.833	0.619	0.900	0.838	0.782	0.946	0.514	0.907	0.782	0.449	0.706	0.819

Reproduced Result

Machine Learning Algorithms – Segmentation

- KMean
- sklearn.cluster.KMeans
- SVM
- Sklearn.svm.SVC

- Machine Learning algorithms take a longer time for testing.

method	class mIOU	instance mIOU	air plane	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor bike	mug	pistol	rocket	skate board	table
3dgcn(ours)	0.810	0.843	0.810	0.817	0.811	0.762	0.898	0.794	0.903	0.878	0.825	0.953	0.619	0.938	0.806	0.567	0.764	0.827
kmeans	0.685	0.671	0.629	0.714	0.727	0.655	0.669	0.714	0.853	0.6	0.745	0.566	0.771	0.658	0.652	0.667	0.699	0.642
svm	0.840	0.857	0.826	0.786	0.909	0.763	0.915	0.857	0.945	0.812	0.769	0.94	0.948	0.947	0.864	0.611	0.699	0.847

Machine Learning Algorithms Result Comparsion

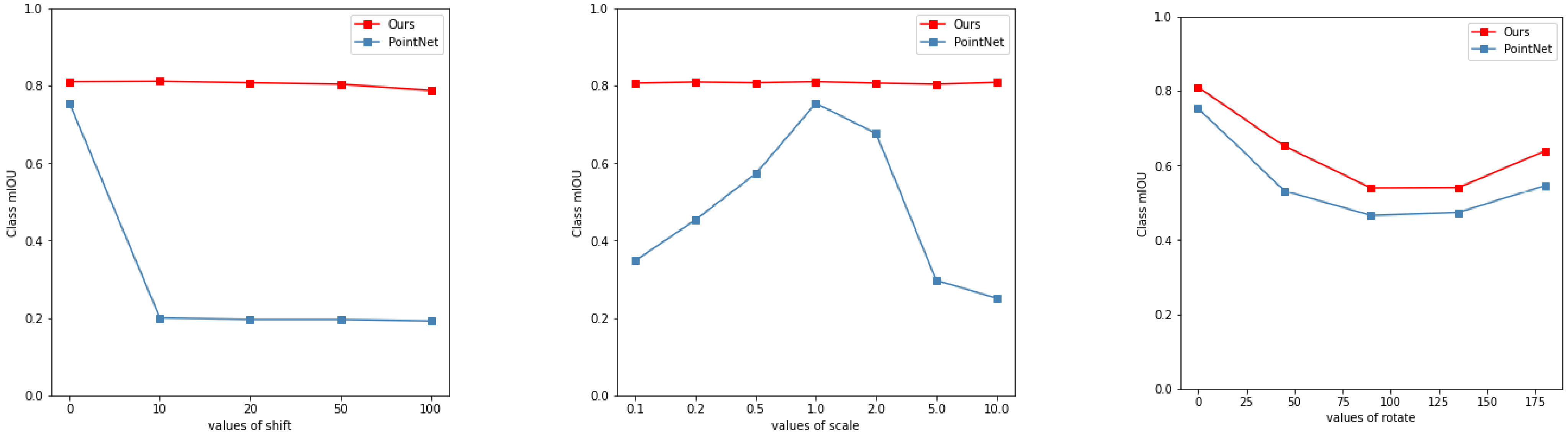
Innovations

1. Low Network Complexity with similar performance.

2. Better shift, scale and rotation invariance.

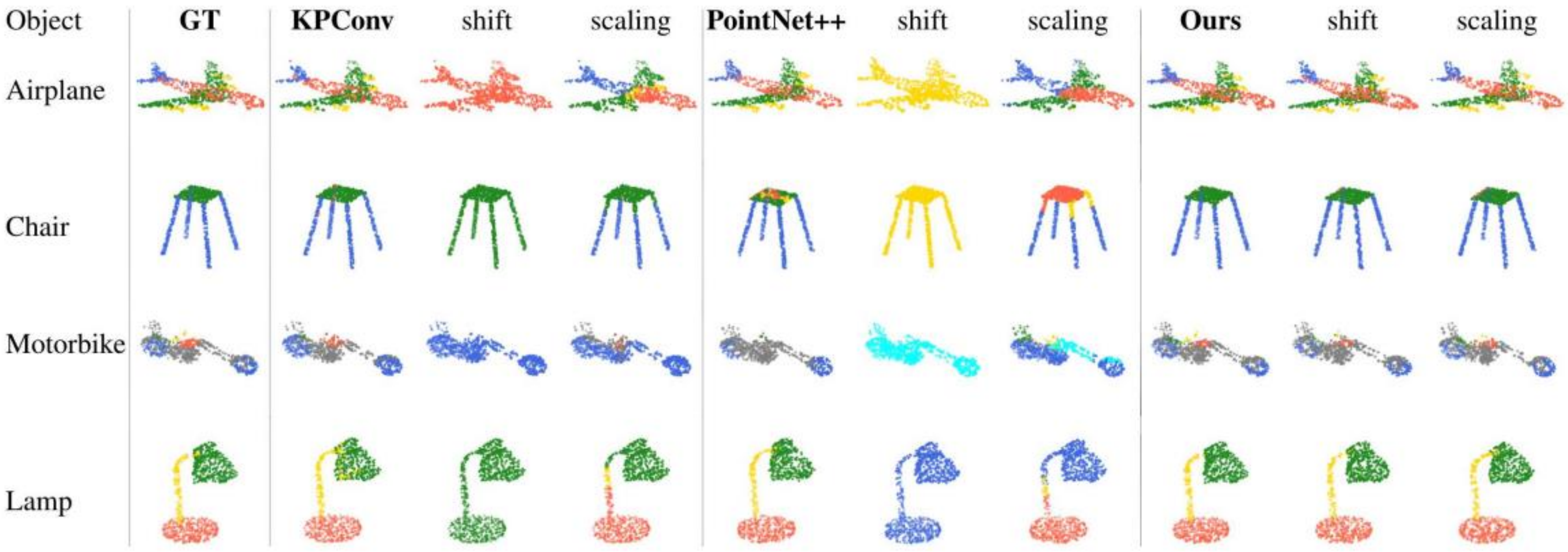
method	#params	Acc. (%)
PointNet [21]	3.5M	89.2
PointNet++ [23]	1.48M	91.9
DGCNN [32]	1.81M	92.9
KPConv [30]	14.3M	92.9
Ours	**0.89M**	92.1

Classification Accuracy and Network Parameter Numbers



Segmentation invariance evaulation testing

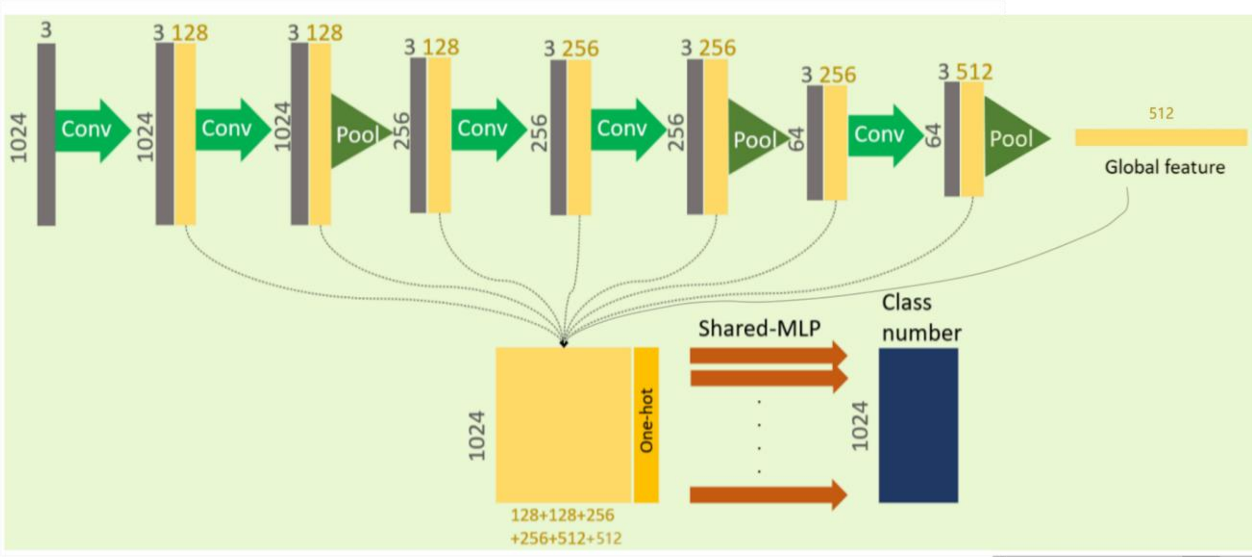
Preliminary Results – Segmentation

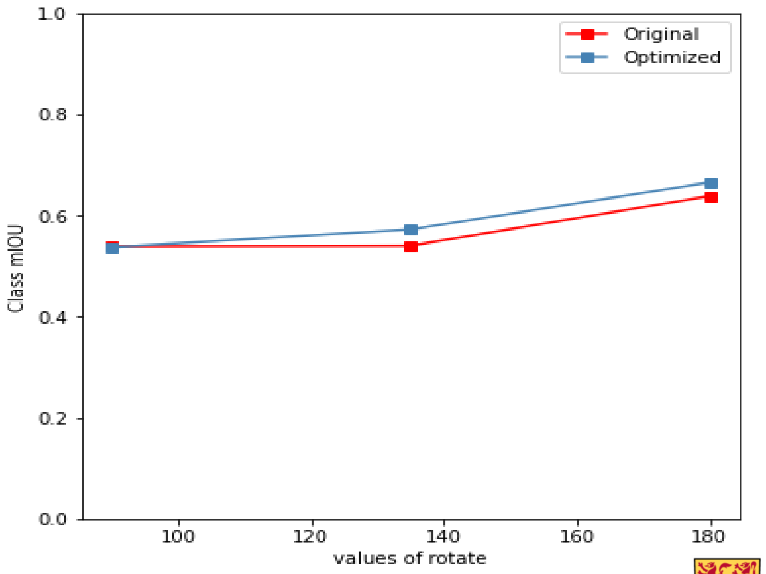


Part segmentation results on ShapeNetPart

Improvement

- The segmentation network does not take advantage of the global feature.
- So we max pool the last local feature to get the global feature and concatenate it with other local features.
- This optimization increase the networks' rotation invariance when the rotation angle is big.





References

- Lin, Zhi-Hao, Sheng-Yu Huang, and Yu-Chiang Frank Wang. Convolution in the cloud: Learning deformable kernels in 3d graph convolution networks for point cloud analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
- Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
- Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic Graph CNN for Learning on Point Clouds. ACM Transactions on Graphics (TOG), 38(5):146:1–146:12, 2019.
- https://paperswithcode.com/sota/3d-point-cloud-classification-on-modelnet40.
- [CVPR2020] Convolution in the Cloud. https://www.youtube.com/watch?v=xfftSRFlWY0.
- https://github.com/j1a0m0e4sNTU/3dgcn.

UNIVERSITY OF
WATERLOO

FACULTY OF ENGINEERING

Thank you!