Dynamic Graph CNN for Learning on Point Clouds

Credit: Yue Wang\textsuperscript{1}, Yongbin Sun\textsuperscript{1}, Ziwei Liu\textsuperscript{2}, Sanjay E. Sarma\textsuperscript{1}, Michael M. Bronstein\textsuperscript{3}, Justin M. Solomon\textsuperscript{1}

\textsuperscript{1}MIT
\textsuperscript{2}UC Berkeley
\textsuperscript{3}USI/TAU/Intel

Presenter: Fuwen Tan
https://qdata.github.io/deep2Read
Point Cloud Representation of 3D Shape

\[ \mathbf{X} = \{ \mathbf{x}_1, \cdots, \mathbf{x}_n \} \subseteq \mathbb{R}^F \]  

Figure: Point Cloud representation of a *plane*. Each point vector may encode multiple attributes, e.g. 3D coordinate, surface normal, color, etc.
**Figure**: Class-specific part segmentation

Classification  | Part Segmentation  | Semantic Segmentation

Credit: Yue Wang, Yongbin Sun, Ziwei Liu, SaDynamic Graph CNN for Learning on Point Clouds. Presenter: Fuwen Tan
**EdgeConv**

\[ e_{ij} = h_{\theta}(x_i, x_j - x_i) = W_c([x_i; x_j - x_i]) = \max_{j: (i, j) \in E} \{e_{ij}\} \]

**Figure:** Edge Convolution: a symmetry function for the two vertices.
How to define E (the edge set)?

- k-nn in the **feature** space ($x_i \in \mathbb{R}^F$)
- the main distinction from previous works
- each layer has a different graph, which will change after each training iteration
Dynamic Graph CNNs

Figure: Overview

Credit: Yue Wang, Yongbin Sun, Ziwei Liu, SaDynamic Graph CNN for Learning on Point Clouds
Presenter: Fuwen Tan
https://qdata.github.io/deep2Read
Point Cloud Transformation

- Proposed in PointNet [6]
- Align the local neighborhood of a point to a canonical space by applying an estimated 3x3 matrix
- Similar with the spatial transformer network in 2D
Shape recognition: implementation

- K=20
- Each EdgeConv block has a shared edge function $h_\theta$
- Short-cut connections for multi-scale feature aggregations (not clear)
- ReLU+BatchNorm after each layer
- 0.5 Dropout rate for the last two fc layers
- A variant version (Baseline): no point cloud transformer and using fixed graph

Credit: Yue Wang, Yongbin Sun, Ziwei Liu, SaDynamic Graph CNN for Learning on Point Cl
Presenter: Fuwen Tan
Shape recognition: experiment

- Dataset: ModelNet40 [12]
  - 9843/2468 CAD shapes
  - 40 categories
  - 1024 points sampled for each shape and normalized to the unit sphere
# Shape recognition: results

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean Class Accuracy</th>
<th>Overall Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>3DShapeNets [12]</td>
<td>77.3</td>
<td>84.7</td>
</tr>
<tr>
<td>VoxNet [5]</td>
<td>83.0</td>
<td>85.9</td>
</tr>
<tr>
<td>Subvolume [7]</td>
<td>86.0</td>
<td>89.2</td>
</tr>
<tr>
<td>ECC [10]</td>
<td>83.2</td>
<td>87.4</td>
</tr>
<tr>
<td>PointNet [6]</td>
<td>86.0</td>
<td>89.2</td>
</tr>
<tr>
<td>PointNet++ [8]</td>
<td>-</td>
<td>90.7</td>
</tr>
<tr>
<td>Kd-Net (depth 10) [4]</td>
<td>-</td>
<td>90.6</td>
</tr>
<tr>
<td>Kd-Net (depth 15) [4]</td>
<td>-</td>
<td>91.8</td>
</tr>
<tr>
<td>Ours (baseline)</td>
<td>88.8</td>
<td>91.2</td>
</tr>
<tr>
<td>Ours</td>
<td>90.2</td>
<td>92.2</td>
</tr>
</tbody>
</table>

**Table:** Classification results on ModelNet40.

Credit: Yue Wang, Yongbin Sun, Ziwei Liu, SaDynamic Graph CNN for Learning on Point Clouds

Presenter: Fuwen Tan

[https://qdata.github.io/deep2Read](https://qdata.github.io/deep2Read)
Shape recognition: model complexity

<table>
<thead>
<tr>
<th>Model</th>
<th>Model size (MB)</th>
<th>Forward time (ms)</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>POINTNet (Baseline)</td>
<td>9.4</td>
<td>11.6</td>
<td>87.1</td>
</tr>
<tr>
<td>POINTNet</td>
<td>40</td>
<td>25.3</td>
<td>89.2</td>
</tr>
<tr>
<td>POINTNet++</td>
<td>12</td>
<td>163.2</td>
<td>90.7</td>
</tr>
<tr>
<td>Ours (Baseline)</td>
<td>11</td>
<td>29.7</td>
<td>91.2</td>
</tr>
<tr>
<td>Ours</td>
<td>21</td>
<td>94.6</td>
<td>92.2</td>
</tr>
</tbody>
</table>

**Table:** Complexity, forward time and accuracy of different models
Centralization: $h_\theta(x_i, x_j - x_i)$ vs $h_\theta(x_i, x_j)$

<table>
<thead>
<tr>
<th>CENT</th>
<th>DYN</th>
<th>XFORM</th>
<th>MEAN CLASS ACCURACY(%)</th>
<th>OVERALL ACCURACY(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>x</td>
<td></td>
<td>88.8</td>
<td>91.2</td>
</tr>
<tr>
<td>x</td>
<td></td>
<td>x</td>
<td>88.8</td>
<td>91.5</td>
</tr>
<tr>
<td>x</td>
<td></td>
<td></td>
<td>89.6</td>
<td>91.9</td>
</tr>
<tr>
<td></td>
<td>x</td>
<td>x</td>
<td>89.8</td>
<td>91.9</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>90.2</td>
<td>92.2</td>
</tr>
</tbody>
</table>

**Table:** Effectiveness of different components. CENT denotes centralization, DYN denotes dynamical graph recomputation, and XFORM denotes the use of a spatial transformer.

Credit: Yue Wang, Yongbin Sun, Ziwei Liu, SaDynamic Graph CNN for Learning on Point Cli
Presenter: Fuwen Tan [https://qdata.github.io/deep2Read](https://qdata.github.io/deep2Read)
Shape recognition: ablation study

Figure:
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.

Credit: Yue Wang, Yongbin Sun, Ziwei Liu, SaDynamic Graph CNN for Learning on Point Clouds
Presenter: Fuwen Tan
https://qdata.github.io/deep2Read
### Shape recognition: ablation study

<table>
<thead>
<tr>
<th>Number of Nearest Neighbors (K)</th>
<th>Mean Class Accuracy (%)</th>
<th>Overall Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>88.0</td>
<td>90.5</td>
</tr>
<tr>
<td>10</td>
<td>88.8</td>
<td>91.4</td>
</tr>
<tr>
<td>20</td>
<td>90.2</td>
<td>92.2</td>
</tr>
<tr>
<td>40</td>
<td>89.2</td>
<td>91.7</td>
</tr>
</tbody>
</table>

**Table**: Results of our model with different numbers of nearest neighbors.
- \( K=30 \)
- similar with the shape recognition model
Part segmentation: experiment

- Dataset: ShapeNet part dataset [11]
  - 16881 3D shapes
  - splits defined in [2]
  - 16 categories
  - 50 parts in total
  - 2048 points sampled for each shape
  - evalutation metric: IoU on points
### Part segmentation: results

<table>
<thead>
<tr>
<th></th>
<th>MEAN</th>
<th>AERO</th>
<th>BAG</th>
<th>CAP</th>
<th>CAR</th>
<th>CHAIR</th>
<th>EAR</th>
<th>GUITAR</th>
<th>KNIFE</th>
<th>LAMP</th>
<th>LAPTOP</th>
<th>MOTOR</th>
<th>MUG</th>
<th>PISTOL</th>
<th>ROCKET</th>
<th>SKATE</th>
<th>BOARD</th>
<th>TABLE</th>
<th>WINNING</th>
</tr>
</thead>
<tbody>
<tr>
<td># SHAPES</td>
<td>2690</td>
<td>76</td>
<td>55</td>
<td>898</td>
<td>3758</td>
<td>69</td>
<td>787</td>
<td>392</td>
<td>1547</td>
<td>451</td>
<td>202</td>
<td>184</td>
<td>283</td>
<td>66</td>
<td>152</td>
<td>5271</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POINTNet [6]</td>
<td>83.7</td>
<td>83.4</td>
<td>78.7</td>
<td>82.5</td>
<td>74.9</td>
<td>89.6</td>
<td>73.0</td>
<td>91.5</td>
<td>85.9</td>
<td>80.8</td>
<td>95.3</td>
<td>65.2</td>
<td>93.0</td>
<td>81.2</td>
<td>57.9</td>
<td>72.8</td>
<td>80.6</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>POINTNet++ [8]</td>
<td><strong>85.1</strong></td>
<td>82.4</td>
<td>79.0</td>
<td><strong>87.7</strong></td>
<td>77.3</td>
<td>90.8</td>
<td>71.8</td>
<td>91.0</td>
<td>85.9</td>
<td><strong>83.7</strong></td>
<td>95.3</td>
<td><strong>71.6</strong></td>
<td><strong>94.1</strong></td>
<td>81.3</td>
<td>58.7</td>
<td><strong>76.4</strong></td>
<td>82.6</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>KD-Net [4]</td>
<td>82.3</td>
<td>80.1</td>
<td>74.6</td>
<td>74.3</td>
<td>70.3</td>
<td>88.6</td>
<td>73.5</td>
<td>90.2</td>
<td>87.2</td>
<td>81.0</td>
<td>94.9</td>
<td>57.4</td>
<td>86.7</td>
<td>78.1</td>
<td>51.8</td>
<td>69.9</td>
<td>80.3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>LOCALFeatureNet [9]</td>
<td>84.3</td>
<td><strong>86.1</strong></td>
<td>73.0</td>
<td>54.9</td>
<td><strong>77.4</strong></td>
<td>88.8</td>
<td>55.0</td>
<td>90.6</td>
<td>86.5</td>
<td>75.2</td>
<td><strong>96.1</strong></td>
<td>57.3</td>
<td>91.7</td>
<td><strong>83.1</strong></td>
<td>53.9</td>
<td>72.5</td>
<td><strong>83.8</strong></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td><strong>85.1</strong></td>
<td>84.2</td>
<td><strong>83.7</strong></td>
<td>84.4</td>
<td>77.1</td>
<td><strong>90.9</strong></td>
<td><strong>78.5</strong></td>
<td><strong>91.5</strong></td>
<td>87.3</td>
<td>82.9</td>
<td>96.0</td>
<td>67.8</td>
<td>93.3</td>
<td>82.6</td>
<td><strong>59.7</strong></td>
<td>75.5</td>
<td>82.0</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

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

Credit: Yue Wang, Yongbin Sun, Ziwei Liu, SaDynamic Graph CNN for Learning on Point Clouds. Presenter: Fuwen Tan. [https://qdata.github.io/deep2Read](https://qdata.github.io/deep2Read)
Figure: **Left:** The mean IoU (%) improves when the ratio of kept points increases. Points are dropped from one of six sides (top, bottom, left, right, front and back) randomly during evaluation process. **Right:** Part segmentation results on partial data. Points on each row are dropped from the same side. The keep ratio is shown below the bottom row. Note that the segmentation results of turbines are improved when more points are included.
Indoor scene segmentation: experiment

- Dataset: S3DIS [1]
  - 6 indoor areas
  - 272 rooms in total
  - 16 semantic categories
  - 9D feature vector: XYZ, normalized XYZ, color
  - 4096 points sampled for each shape during training, all points are used during testing
  - evaluation metric: IoU on points
### Table: 3D semantic segmentation results on S3DIS

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean IoU</th>
<th>Overall Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>PointNet (Baseline) [6]</td>
<td>20.1</td>
<td>53.2</td>
</tr>
<tr>
<td>PointNet [6]</td>
<td>47.6</td>
<td>78.5</td>
</tr>
<tr>
<td>MS + CU(2) [3]</td>
<td>47.8</td>
<td>79.2</td>
</tr>
<tr>
<td>G + RCU [3]</td>
<td>49.7</td>
<td>81.1</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>56.1</strong></td>
<td><strong>84.1</strong></td>
</tr>
</tbody>
</table>

Table: 3D semantic segmentation results on S3DIS. MS+CU for multi-scale block features with consolidation units; G+RCU for the grid-blocks with recurrent consolidation Units.
Conclusion

- Simple, effective, maybe not very efficient
- The performance looked good at the submitted time (Jan. 2018)
- Not in good shape yet

3d semantic parsing of large-scale indoor spaces.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu.

ShapeNet: An Information-Rich 3D Model Repository.

Francis Engelmann, Theodora Kontogianni, Alexander Hermans, and Bastian Leibe.

Exploring spatial context for 3d semantic segmentation of point clouds.

Roman Klokov and Victor Lempitsky.
Escape from cells: Deep kd-networks for the recognition of 3d point cloud models.

D. Maturana and S. Scherer.
In *IROS*, 2015.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas.
Pointnet: Deep learning on point sets for 3d classification and segmentation.

Charles R Qi, Hao Su, Matthias Niessner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas.
Volumetric and multi-view cnns for object classification on 3d data.
Pointnet++: Deep hierarchical feature learning on point sets in a metric space.


Yiru Shen, Chen Feng, Yaoqing Yang, and Dong Tian.
Neighbors do help: Deeply exploiting local structures of point clouds.

*CoRR, abs/1712.06760, 2017.*

Martin Simonovsky and Nikos Komodakis.
Dynamic edge-conditioned filters in convolutional neural networks on graphs.

*In CVPR, 2017.*

Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas.
A scalable active framework for region annotation in 3d shape collections.

*ACM Transactions on Graphics (SIGGRAPH ASIA), 2016.*

3d shapenets: A deep representation for volumetric shapes.
In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.