INFORMATIK

augmented VISION

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Seminar
WS 2020/2021
A Survey on Hierarchical Structure-based Networks for Learning from Point Cloud

Nishan Tamang

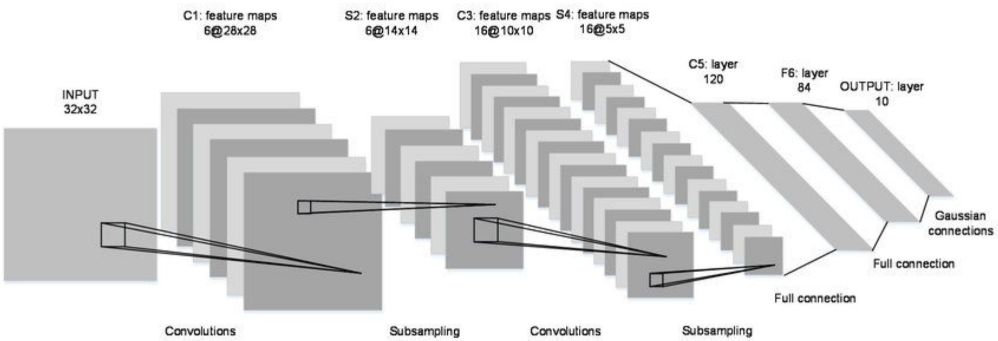
n_tamang19@cs.uni-kl.de

Supervisor: Ramy Battrawy

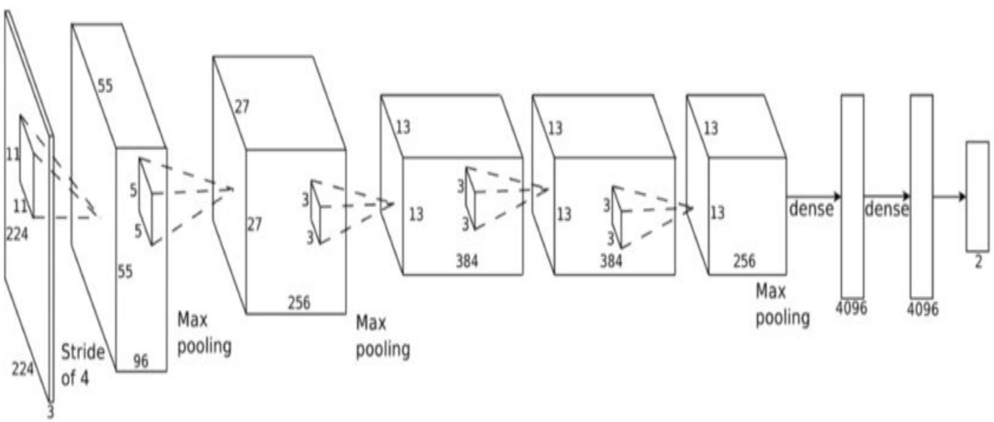
Outline:

- Motivation
- Methods:
  - Kd-networks
  - SO-Net
  - Octree guided network
  - A-CNN
- Results and Comparisons
- Conclusion

Motivation:



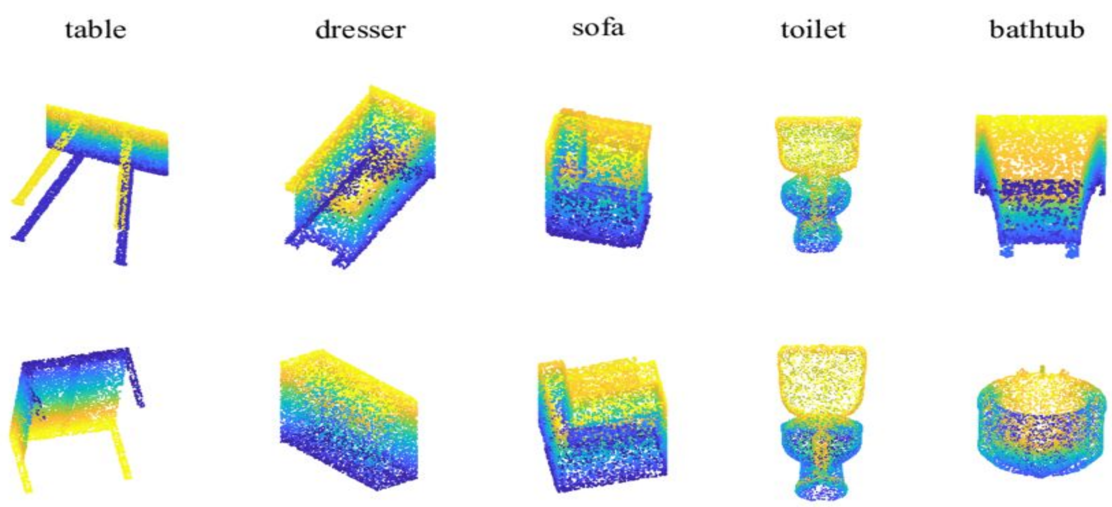
LeCun et al., 1998



Krizhevsky et al., 2012

Motivation:

- **3D data can be obtained from,**
  - **LiDAR, RGB-D cameras and stereo imaging systems.**



ModelNet10 dataset [1]

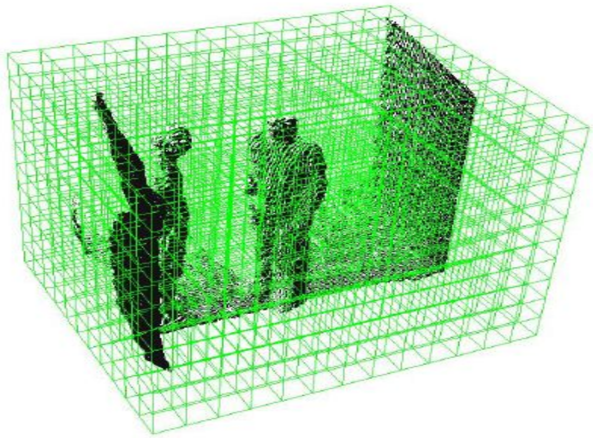
[1] https://modelnet.cs.princeton.edu/.

Motivation:

- **Problems with Point Clouds:**
  - **Invariant to point ordering (permutation invariant).**
  - **Sparse amount of data points.**
  - **Unknown number of points.**
  - **Invariant to rotation.**

Motivation:

- 3D Convolutional Networks:
  - Rasterize 3D data to voxel grids.
  - Sparsity of the 3D data.
  - Increase in processing time (computational cost).
  - Excessive memory usage.



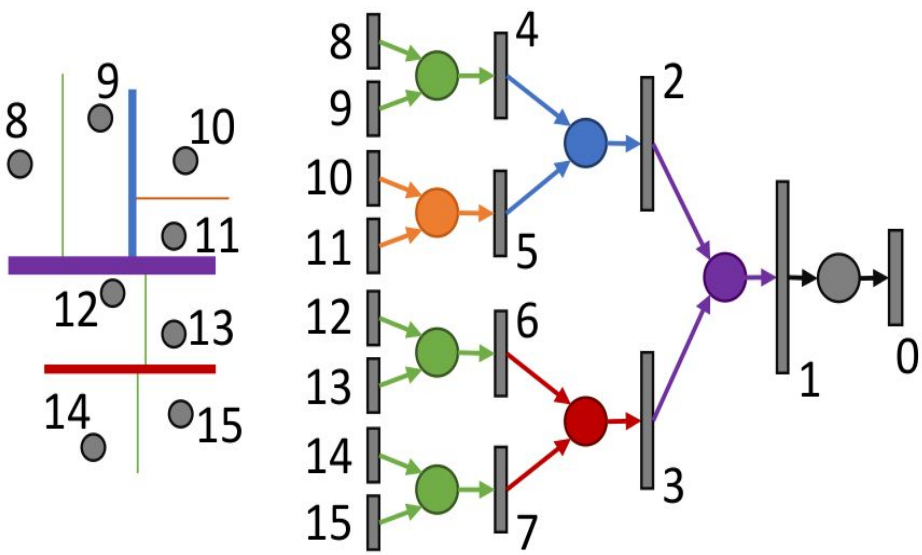
Construction of Voxel-grids on 3D point clouds [2].

- Solution:
  - Build a network based on the construction of different hierarchical structures (kd-tree, octree etc.).

[2] Liu et al., Real-Time Plane Segmentation and Obstacle Detection of 3D Point Clouds for Indoor Scenes, ECCV 2012.

Kd-networks:

- **Constructs Kd-network based on the Kd-tree structure of the input point clouds.**



$$\mathbf{v}_i = \begin{cases} \phi(W^{l_i}_{\mathrm{x}}[\mathbf{v}_{c_1(i)};\mathbf{v}_{c_2(i)}] + \mathbf{b}^{l_i}_{\mathrm{x}}), & \text{if } d_i = \mathrm{x}\,, \\ \phi(W^{l_i}_{\mathrm{y}}[\mathbf{v}_{c_1(i)};\mathbf{v}_{c_2(i)}] + \mathbf{b}^{l_i}_{\mathrm{y}}), & \text{if } d_i = \mathrm{y}\,, \\ \phi(W^{l_i}_{\mathrm{z}}[\mathbf{v}_{c_1(i)};\mathbf{v}_{c_2(i)}] + \mathbf{b}^{l_i}_{\mathrm{z}}), & \text{if } d_i = \mathrm{z}\,, \end{cases}$$

or in short form:

$$\mathbf{v}_i = \phi(W^{l_i}_{d_i}[\mathbf{v}_{c_1(i)};\mathbf{v}_{c_2(i)}] + \mathbf{b}^{l_i}_{d_i})\,.$$

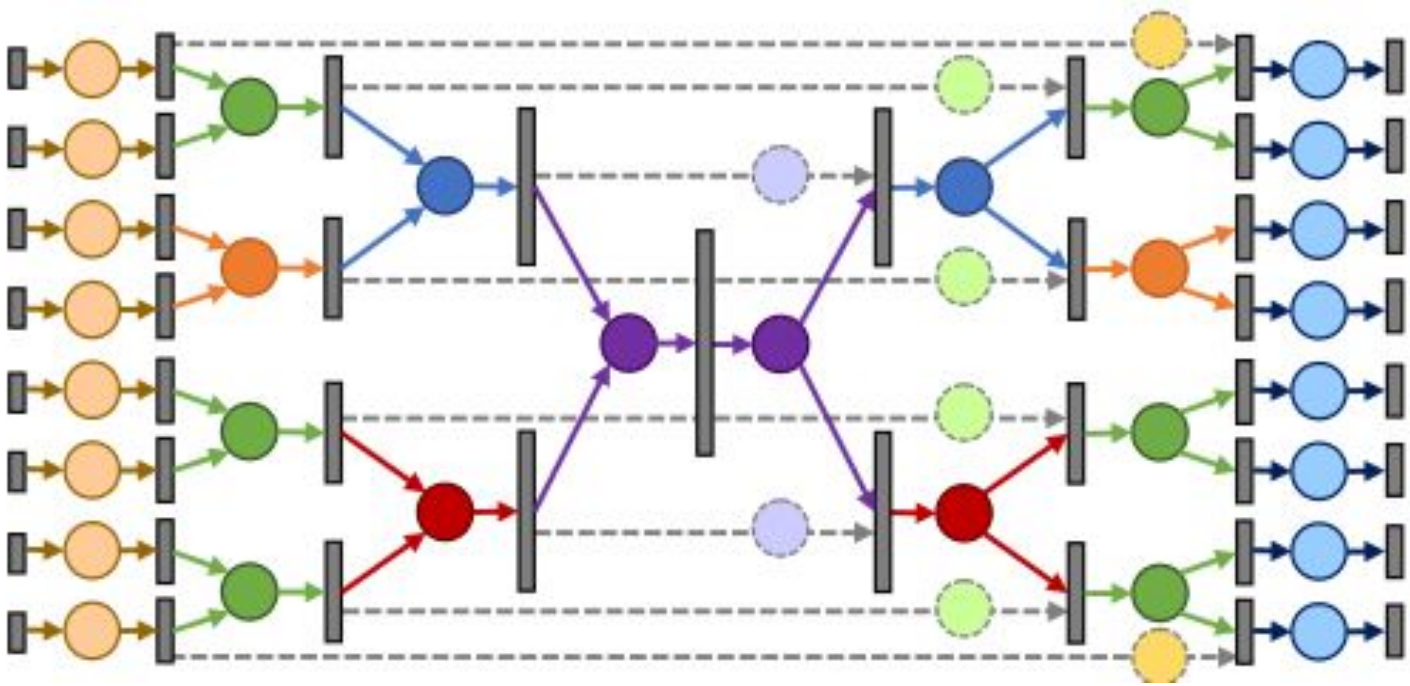
$$\mathbf{v}_0(\mathcal{T}) = W^0\mathbf{v}_1(\mathcal{T}) + \mathbf{b}^0$$

Kd-network for classification [3].

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

Kd-networks:

- **Extension for Segmentation [3] :**



$$\tilde{\mathbf{v}}_{c_1(i)} = \phi([\tilde{W}^{l_i}_{d_{c_1(i)}} \tilde{\mathbf{v}}_i + \tilde{\mathbf{b}}^{l_i}_{d_{c_1(i)}}; S^{l_i} \mathbf{v}_{c_1(i)} + \mathbf{t}^{l_i}]),$$

$$\tilde{\mathbf{v}}_{c_2(i)} = \phi([\tilde{W}^{l_i}_{d_{c_2(i)}} \tilde{\mathbf{v}}_i + \tilde{\mathbf{b}}^{l_i}_{d_{c_2(i)}}; S^{l_i} \mathbf{v}_{c_2(i)} + \mathbf{t}^{l_i}]),$$

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

Kd-networks:

Properties:

- Layerwise parameter sharing.
- Hierarchical Representation.
- Partial invariance to jitter.
- Low memory footprint for segmentation tasks.

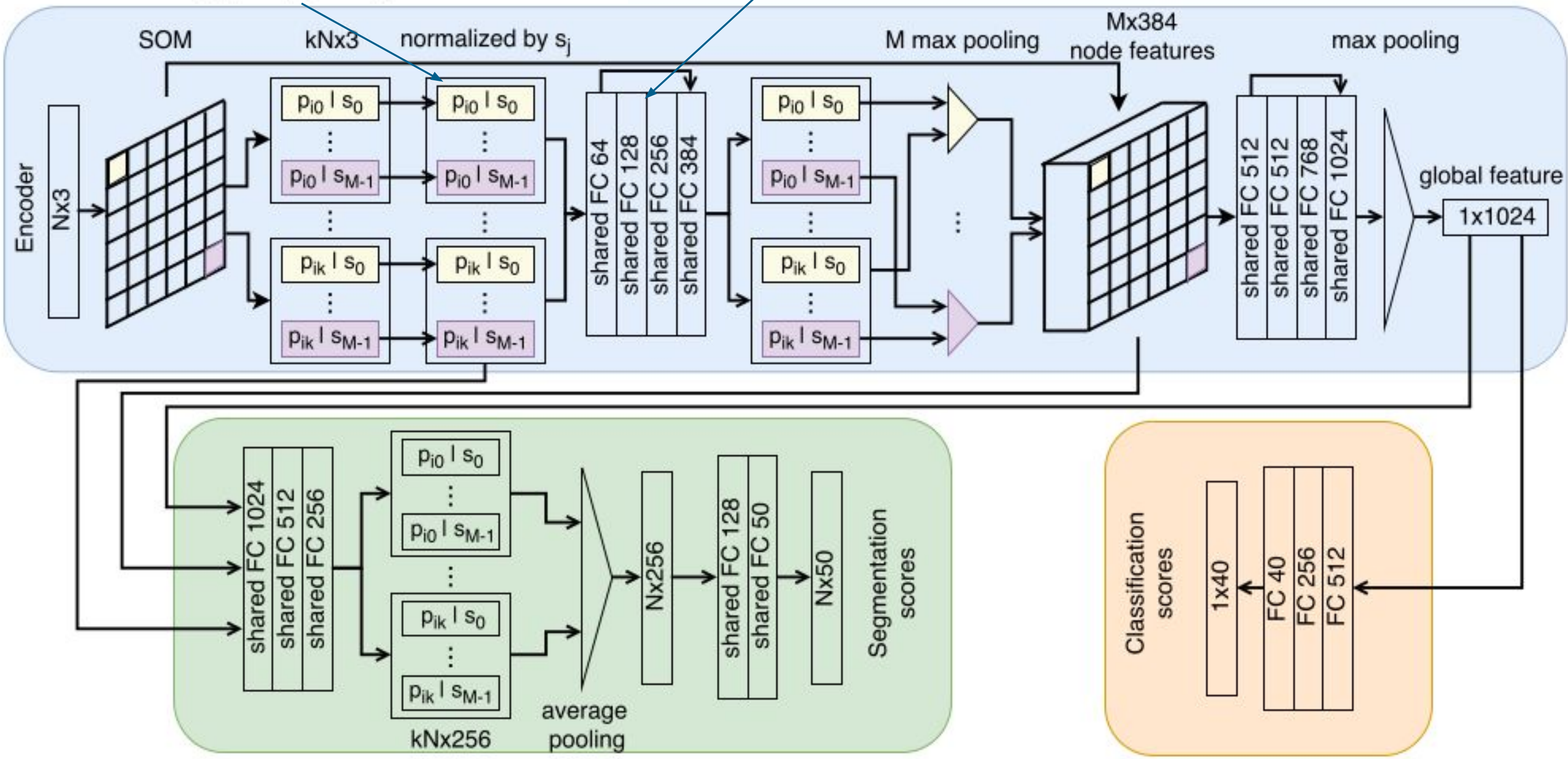
Limitations:

- Non-invariant to rotation.
- Performance depends on the tree construction.

SO-Net:

$$p_{ik}^{l+1} = \phi(W^l p_{ik}^l + b^l).$$

$$p_{ik} = p_i - s_{ik}.$$



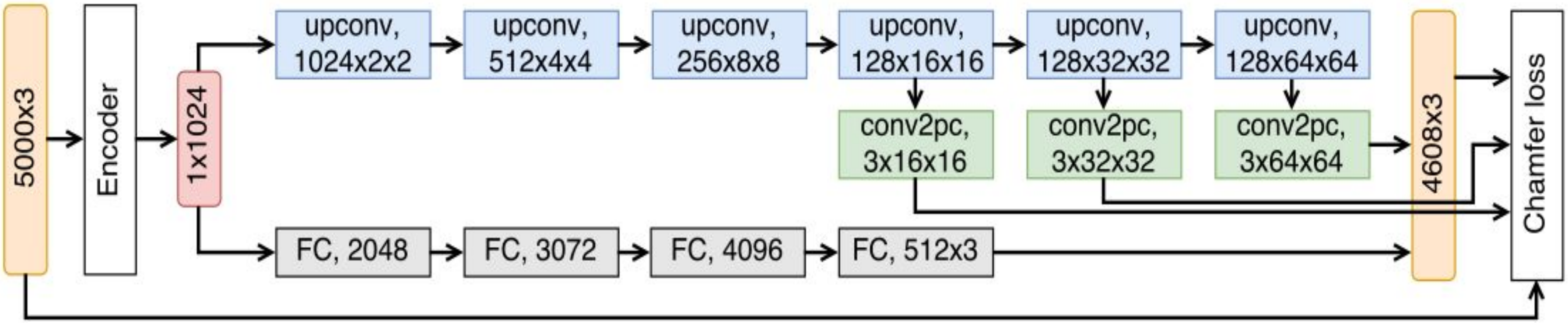
The SO-Net architecture for classification and segmentation [4].

[4] Li et al., SO-Net: Self-organizing network for point cloud analysis, CVPR 2018.

SO-Net:

Chamfer Loss:

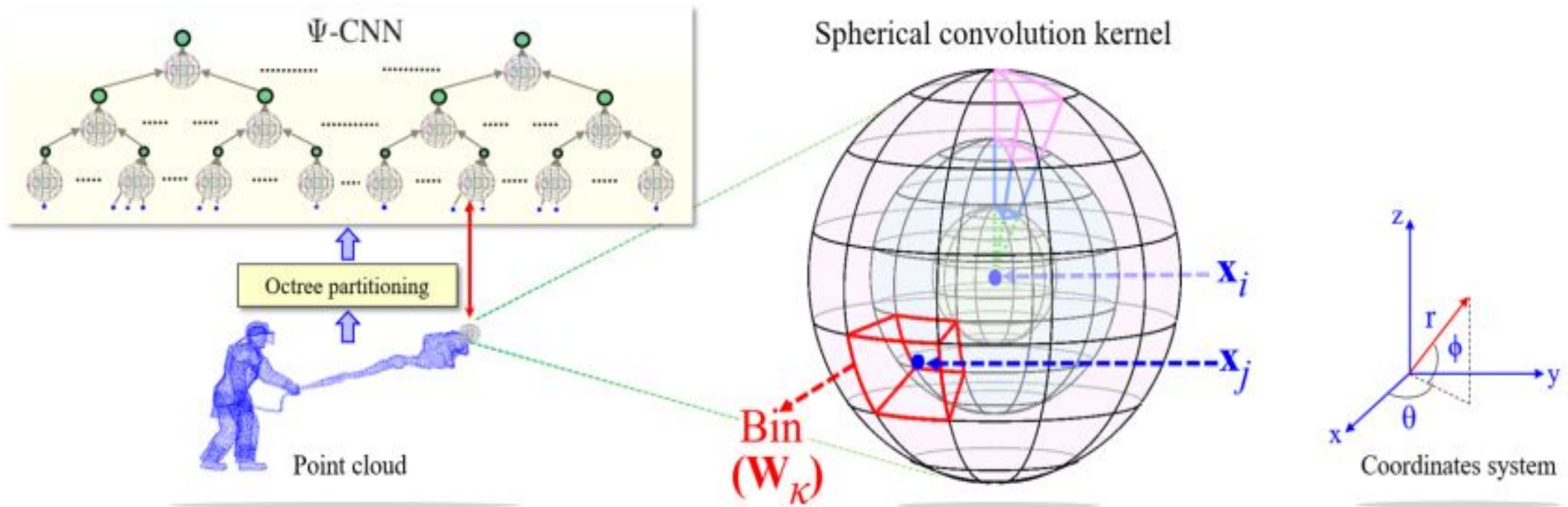
$$d(P_s, P_t) = \frac{1}{|P_s|} \sum_{x \in P_s} \min_{y \in P_t} \|x - y\|_2 + \frac{1}{|P_t|} \sum_{y \in P_t} \min_{x \in P_s} \|x - y\|_2$$



Autoencoder consisting of the decoder network to recover the input point cloud given the global feature vector [4].

[4] Li et al., SO-Net: Self-organizing network for point cloud analysis, CVPR 2018.

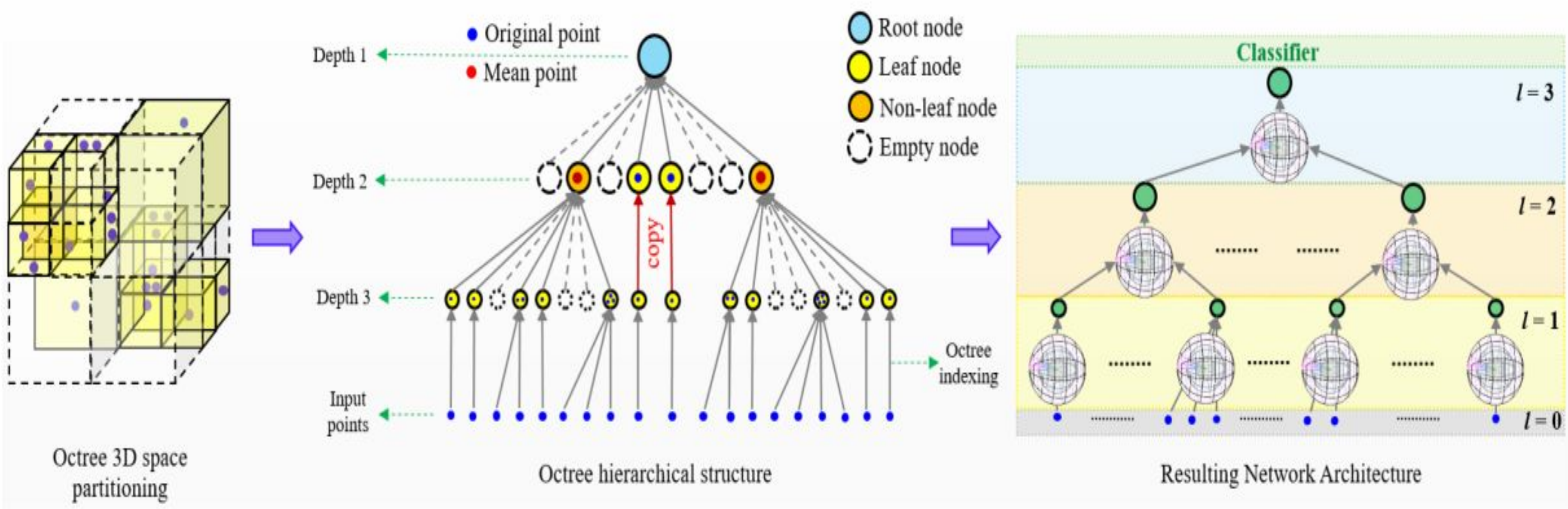
Octree guided CNN with Spherical Convolutional Kernels (Ψ-CNN):



Octree guided CNN (Ψ-CNN) where the raw input point cloud is processed using octree space partitioning. The spherical convolutional kernel is applied between the layers of the network [5].

[5] Lei et al., Octree guided CNN with Spherical Kernels for 3D Point Clouds, CVPR 2019.

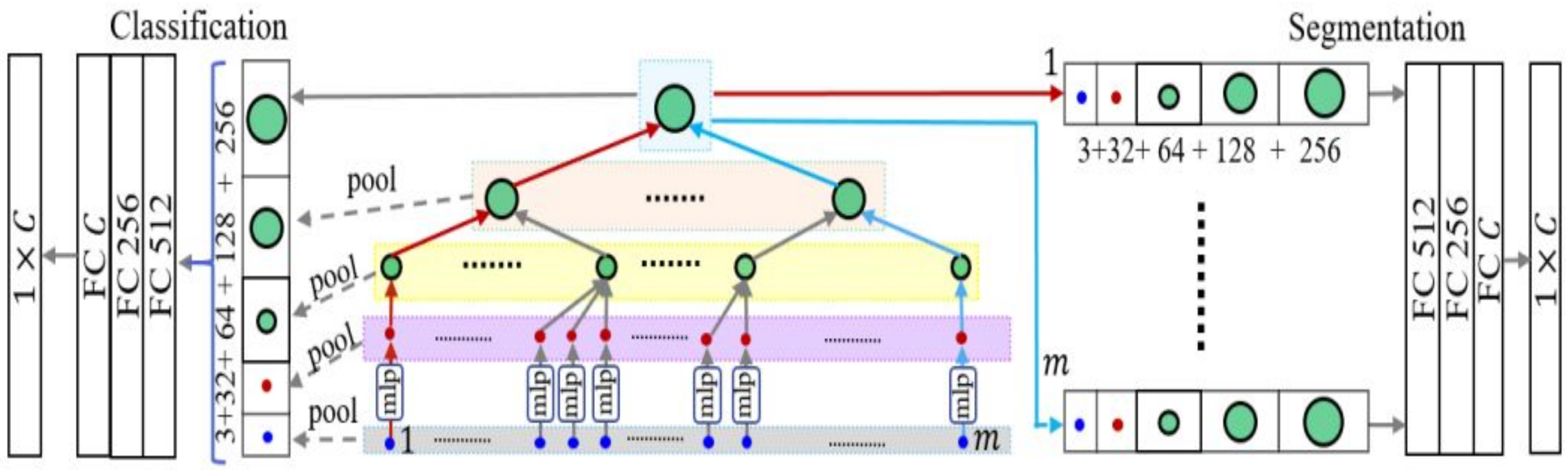
Octree guided CNN with Spherical Convolutional Kernels (Ψ-CNN):



Octree space partitioning on point clouds in 3D space. The resulting Ψ-CNN has the same depth as the corresponding partitioned tree and learns spherical convolutional kernels for feature extraction [5].

[5] Lei et al., Octree guided CNN with Spherical Kernels for 3D Point Clouds, CVPR 2019.

Octree guided CNN with Spherical Convolutional Kernels (Ψ-CNN):

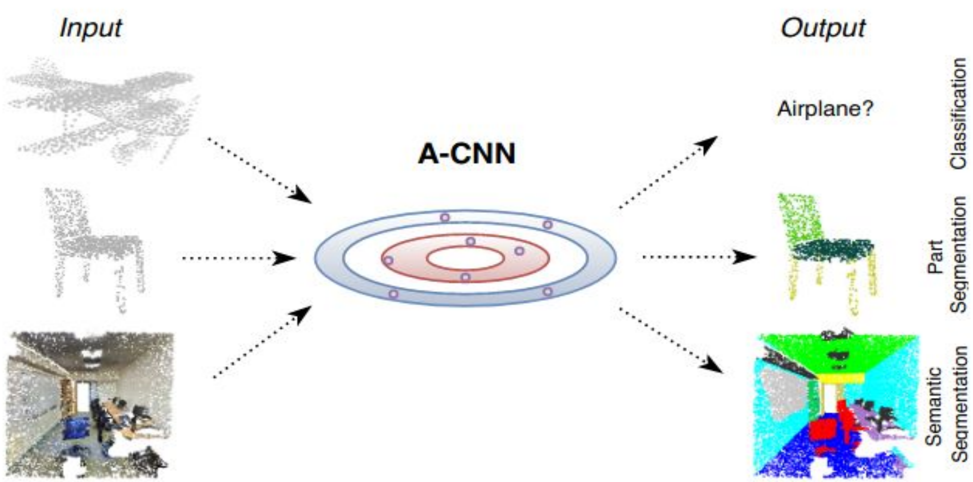


Classification and Segmentation using the root representation and features from the layers of Ψ-CNN [5].

[5] Lei et al., Octree guided CNN with Spherical Kernels for 3D Point Clouds, CVPR 2019.

A-CNN:

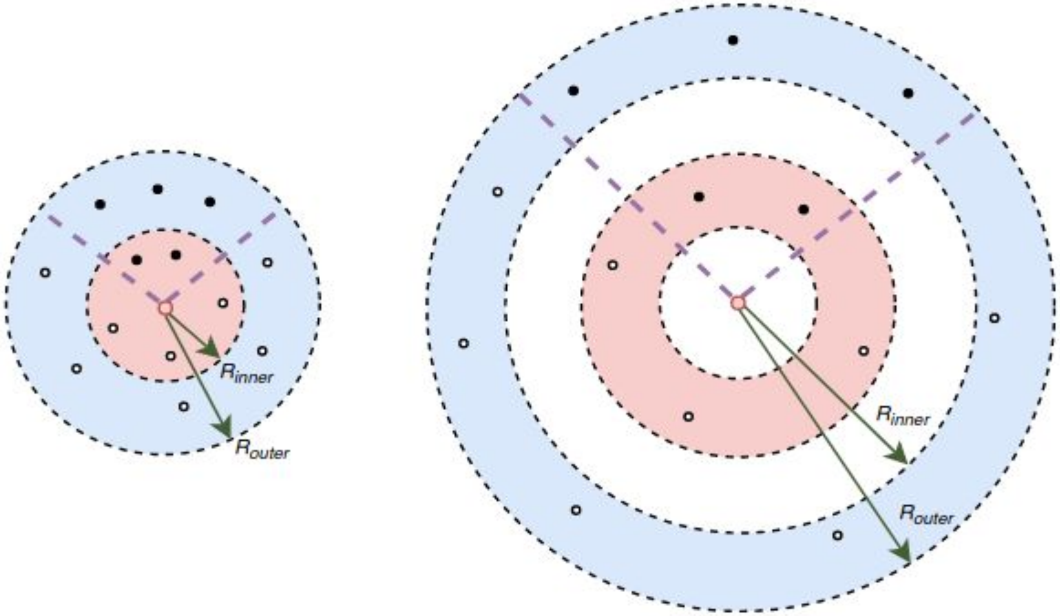
- Uses Annular Convolutions on 3D point clouds
- Captures local neighborhood geometry of each point by specifying the ring-shaped structures and directions in the computation.
- Types of rings: Regular and Dilated rings.



A-CNN on point clouds [6].

A-CNN:

- **Regular and Dilated rings:**

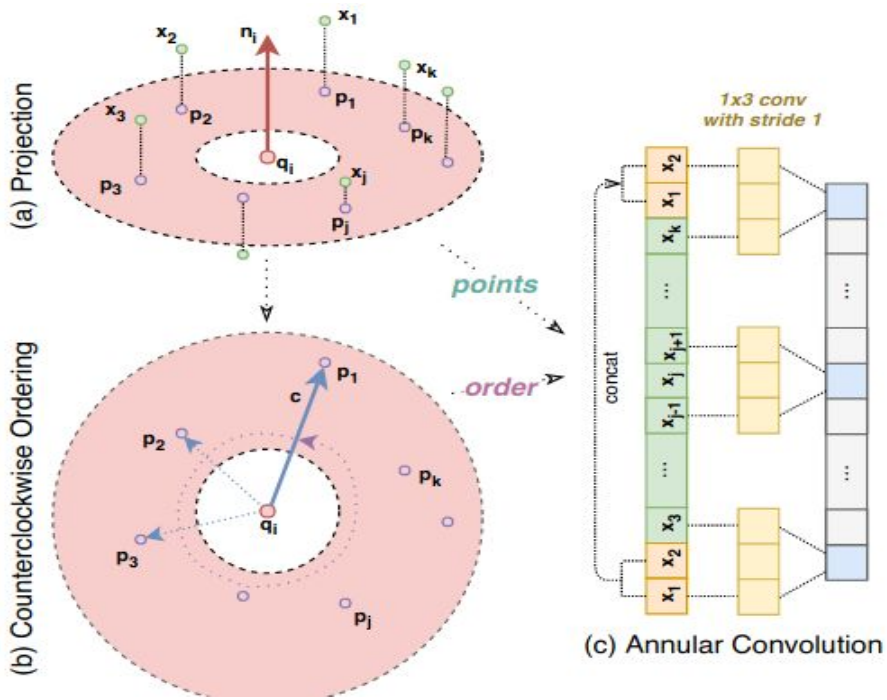


(a) Regular Rings [6] (b) Dilated Rings [6]

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

A-CNN:

- ## Ordering Neighboring Points:



Annular Convolutions on a ring. Projection of points on a tangent plane and ordering of neighboring points in counterclockwise ordering [6].

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

A-CNN:

- **Ordering Neighboring Points:**
    - **Normal estimation of point clouds:**

$$\mathbf{C} = \frac{1}{K}\sum_{j=1}^{K}(\mathbf{x}_j - \mathbf{q}_i)\cdot(\mathbf{x}_j - \mathbf{q}_i)^T,$$

$$\mathbf{C}\cdot\mathbf{v}_\gamma = \lambda_\gamma\cdot\mathbf{v}_\gamma, \gamma\in\{0,1,2\},$$

-
    - **Orthogonal Projection:**

$$\mathbf{p}_j = \mathbf{x}_j - ((\mathbf{x}_j - \mathbf{q}_i)\cdot\mathbf{n}_i)\cdot\mathbf{n}_i, \quad j\in\{1,...,K\}.$$

A-CNN:

- **Ordering Neighboring Points:**
  - **Counter-clockwise ordering**

$$cos(\theta_{\mathbf{p}_j}) = \frac{\mathbf{c} \cdot (\mathbf{p}_j - \mathbf{q}_i)}{||\mathbf{c}||||\mathbf{p}_j - \mathbf{q}_i||}.$$

  -

$$sign_{\mathbf{p}_j} = (\mathbf{c} \times (\mathbf{p}_j - \mathbf{q}_i)) \cdot \mathbf{n}_i, \tag{4}$$

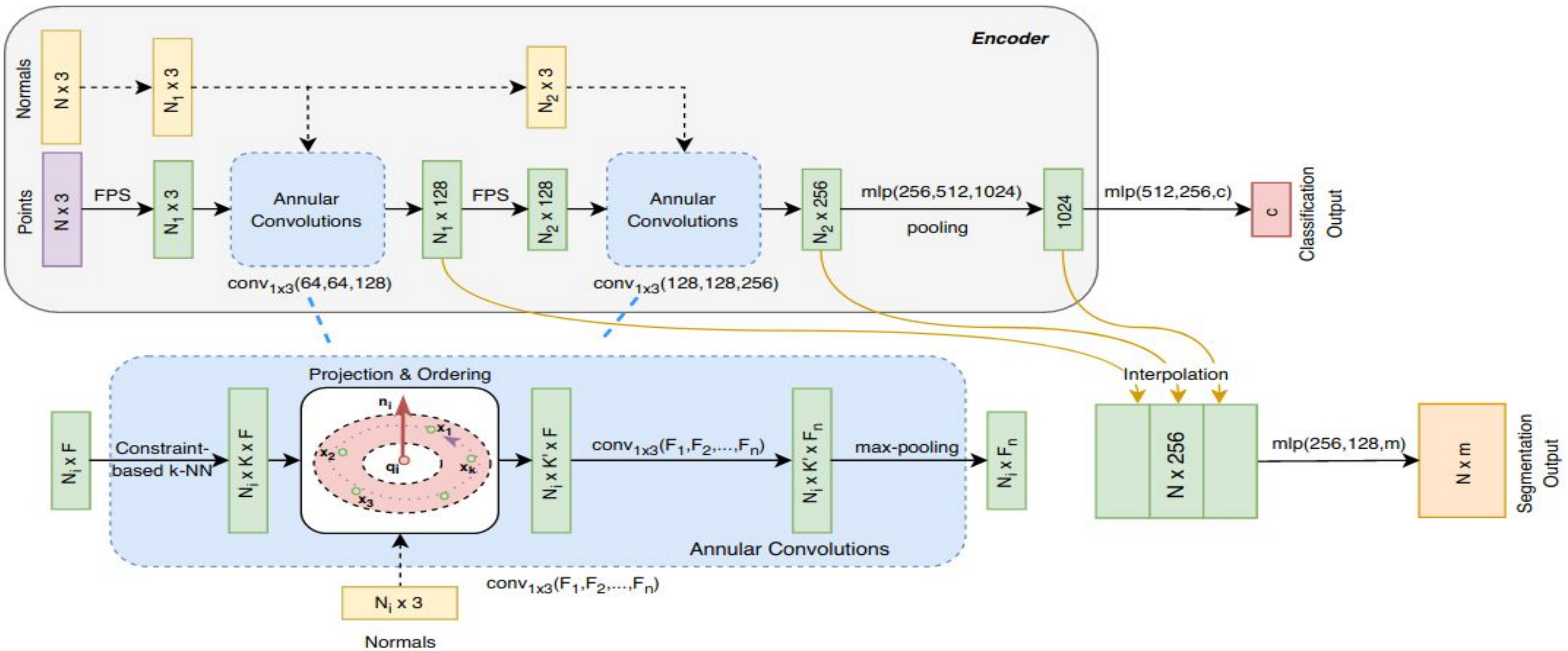
where $sign_{\mathbf{p}_j} \geq 0$ is $\theta_{\mathbf{p}_j} \in [0°, 180°]$, and $sign_{\mathbf{p}_j} < 0$ is $\theta_{\mathbf{p}_j} \in (180°, 360°)$.

  -

$$\angle_{\mathbf{p}_j} = \begin{cases} -cos(\theta_{\mathbf{p}_j}) - 2 & sign_{\mathbf{p}_j} < 0 \\ cos(\theta_{\mathbf{p}_j}) & sign_{\mathbf{p}_j} \geq 0. \end{cases}$$

A-CNN:

- **Architecture:**



A-CNN for Classification and Segmentation [6].

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

Results and Comparisons:

	Classification				Segmentation
	ModelNet10		ModelNet40		ShapeNet
Methods	**Class Accuracy**	**Instance Accuary**	**Class Accuracy**	**Instance Accuracy**	**mIoU**
Kd-Network [3]	92.8	93.8	86.3	90.6	82.3
SO-Net [4]	93.9	94.1	87.3	90.9	84.9
Ψ-CNN [5]	94.4	94.6	88.7	92.0	**86.8**
A-CNN [6]	**95.3**	**95.5**	**90.3**	**92.6**	85.9

[3] Klokov et al., Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models, ICCV 2017.
[4] Li et al. SO-Net: Self-organizing network for point cloud analysis, CVPR 2018.
[5] Lei et al. Octree guided CNN with Spherical Kernels for 3D Point Clouds, CVPR 2019.
[6] Komarichev et al., A-CNN: Annularly Convolutional Neural Networks on Point Clouds, CVPR 2019

Time and Space Complexity:

- **Kd-network (Classification) [3]:**
  - **For depth-10 model, it can take 16 hours (ModelNet40) for training and for depth-15 model, it took upto 5 days for training using an NVidia Titan Black.**
  - **For segmentation, the memory footprint of one example during training is less than 120 MB.**
- **SO-Net (For ModelNet40 classification) [4]:**
  - **3 hours on ModelNet40 with GTX1080Ti.**
- **Ψ-CNN (Test-time for classification on 1K randomly selected samples from ModelNets with point clouds of size 10K) [5]:**
  - **Total: 34.1 ms**
- **A-CNN [6]:**
  - **Training time for segmentation: 19 hours using NVIDIA Titan Xp.**
  - **Size of training model: 22 MB.**

[3] Klokov et al., Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models, ICCV 2017.
[4] Li et al. SO-Net: Self-organizing network for point cloud analysis, CVPR 2018.
[5] Lei et al. Octree guided CNN with Spherical Kernels for 3D Point Clouds, CVPR 2019.
[6] Komarichev et al., A-CNN: Annularly Convolutional Neural Networks on Point Clouds, CVPR 2019

Conclusion:

- Network architectures based on hierarchical structures more effective and practical for point clouds.
- Depth (layer) of the network is equal to the depth of the hierarchical structures.
- A-CNN [6] with ring-structures (regular and dilated) outperform methods such as Kd-network [3], SO-Net [4] and Ψ-CNN [5] on classification.
- For segmentation, Ψ-CNN [5] outperforms other methods with the mIoU of 86.8.

[3] Klokov et al., Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models, ICCV 2017.
[4] Li et al. SO-Net: Self-organizing network for point cloud analysis, CVPR 2018.
[5] Lei et al. Octree guided CNN with Spherical Kernels for 3D Point Clouds, CVPR 2019.
[6] Komarichev et al., A-CNN: Annularly Convolutional Neural Networks on Point Clouds, CVPR 2019

Thank You