Point Convolutional Neural Networks by Extension Operators

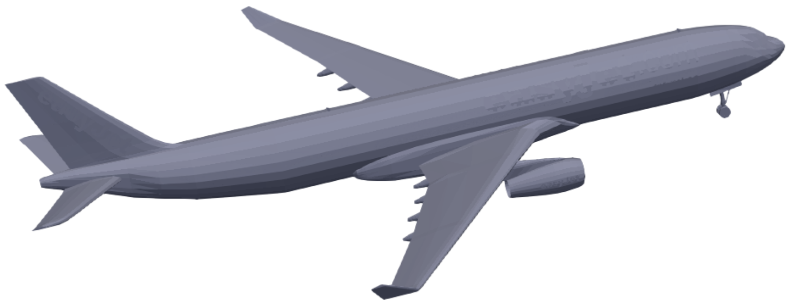
Matan Atzmon

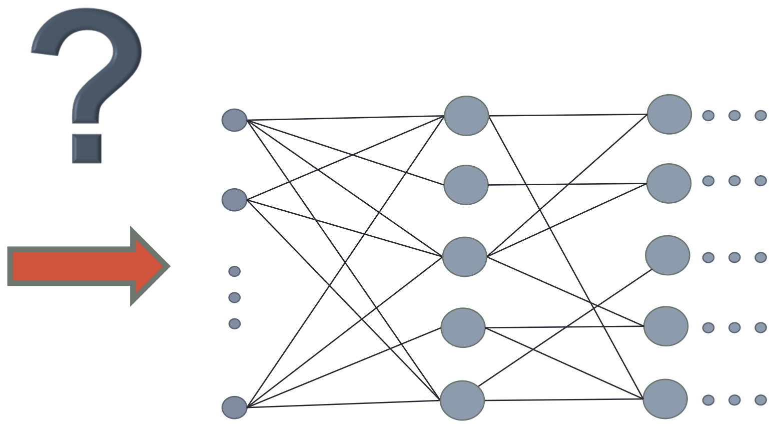
Joint work with **Haggai Maron** and **Yaron Lipman**

Weizmann Institute of Science

SIGGRAPH, August 2018

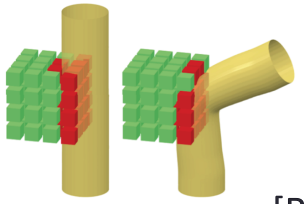
Introduction





Introduction

- Volumetric Methods

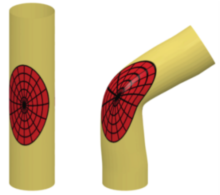


[Boscaini et al. 2016]

- View Based Methods

[Su et al. 2015]

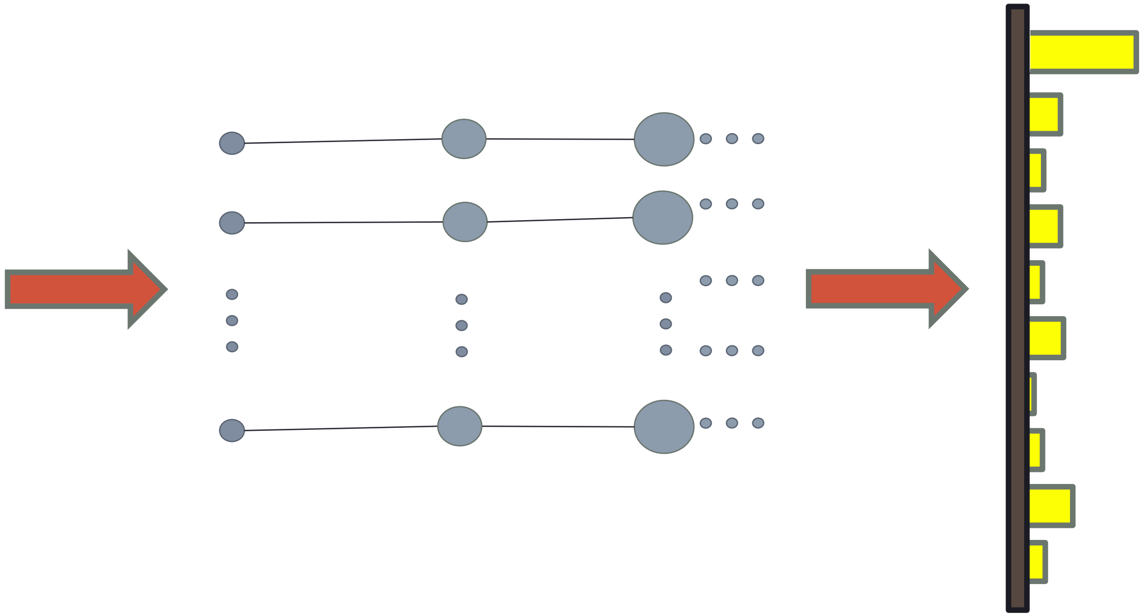
- Graph Spectral Methods
- Surface Representation Methods



[Boscaini et al. 2016]

Deep Learning on Point Clouds

Point Cloud



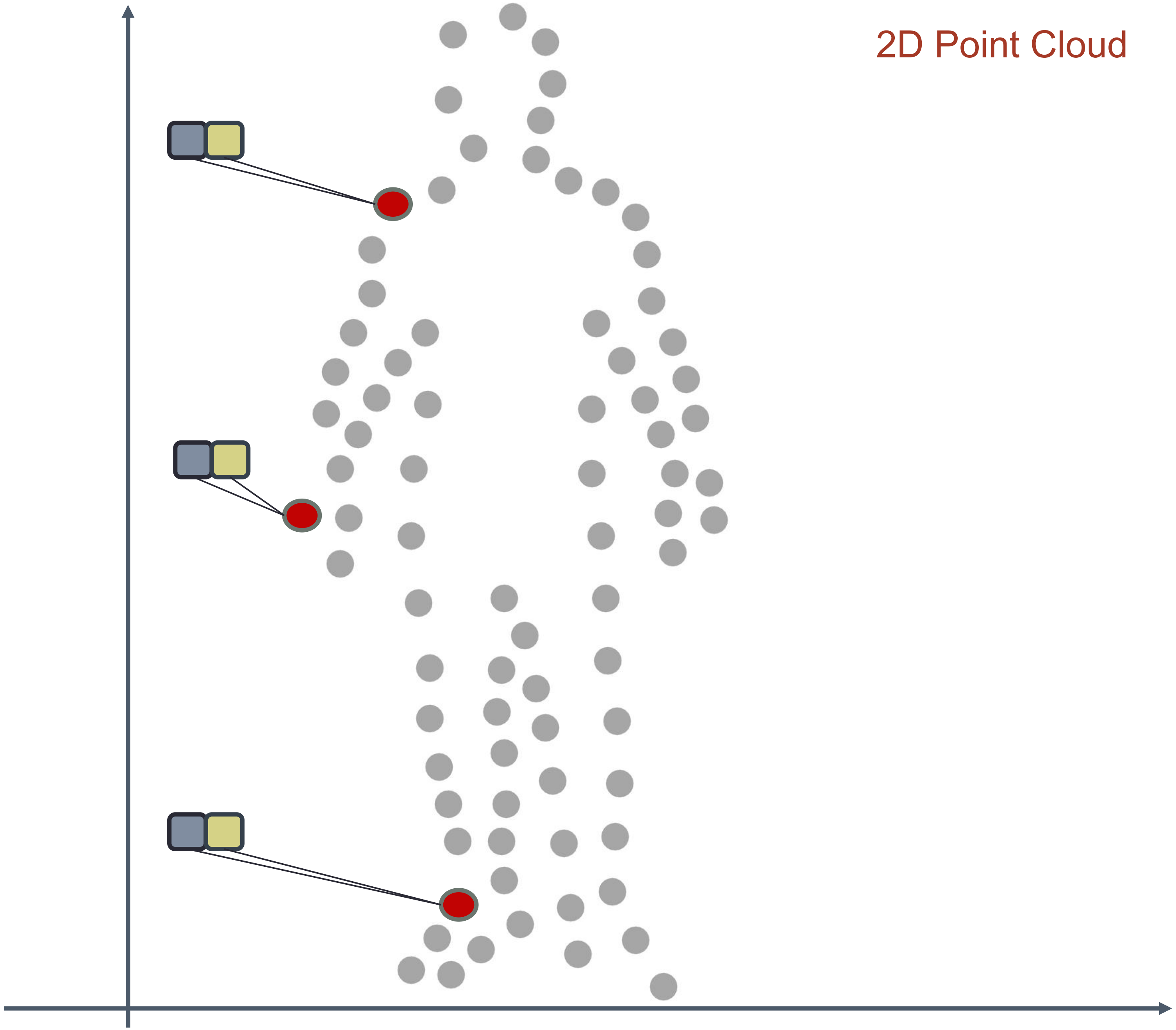
PointNet [Qi 2017]

Question:

How to generalize **convolution** to **point clouds?**

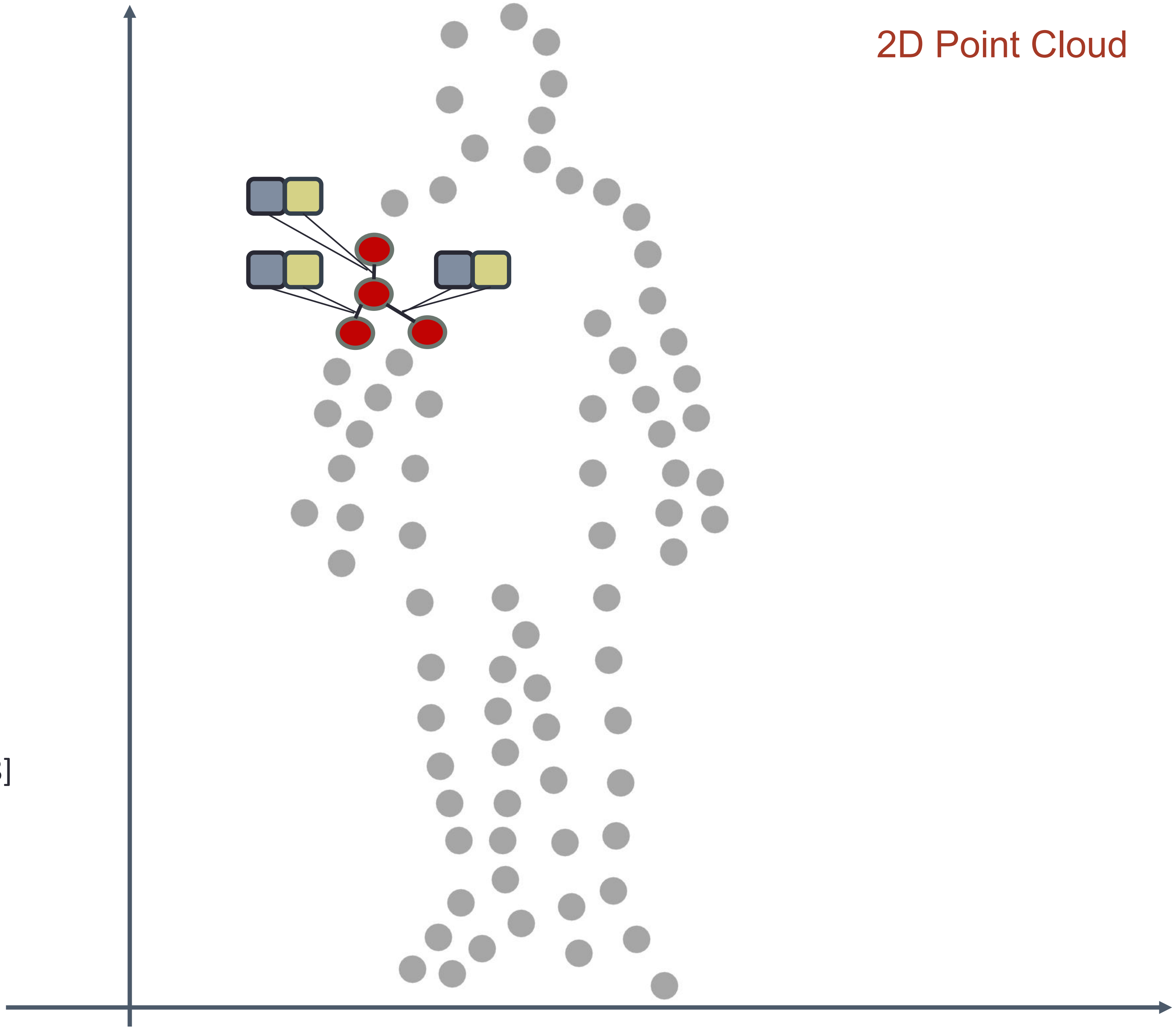
Related Work

- 1×1 Convolution [PointNet, Qi 2017]
  - Order Equivariance
  - No neighborhood information



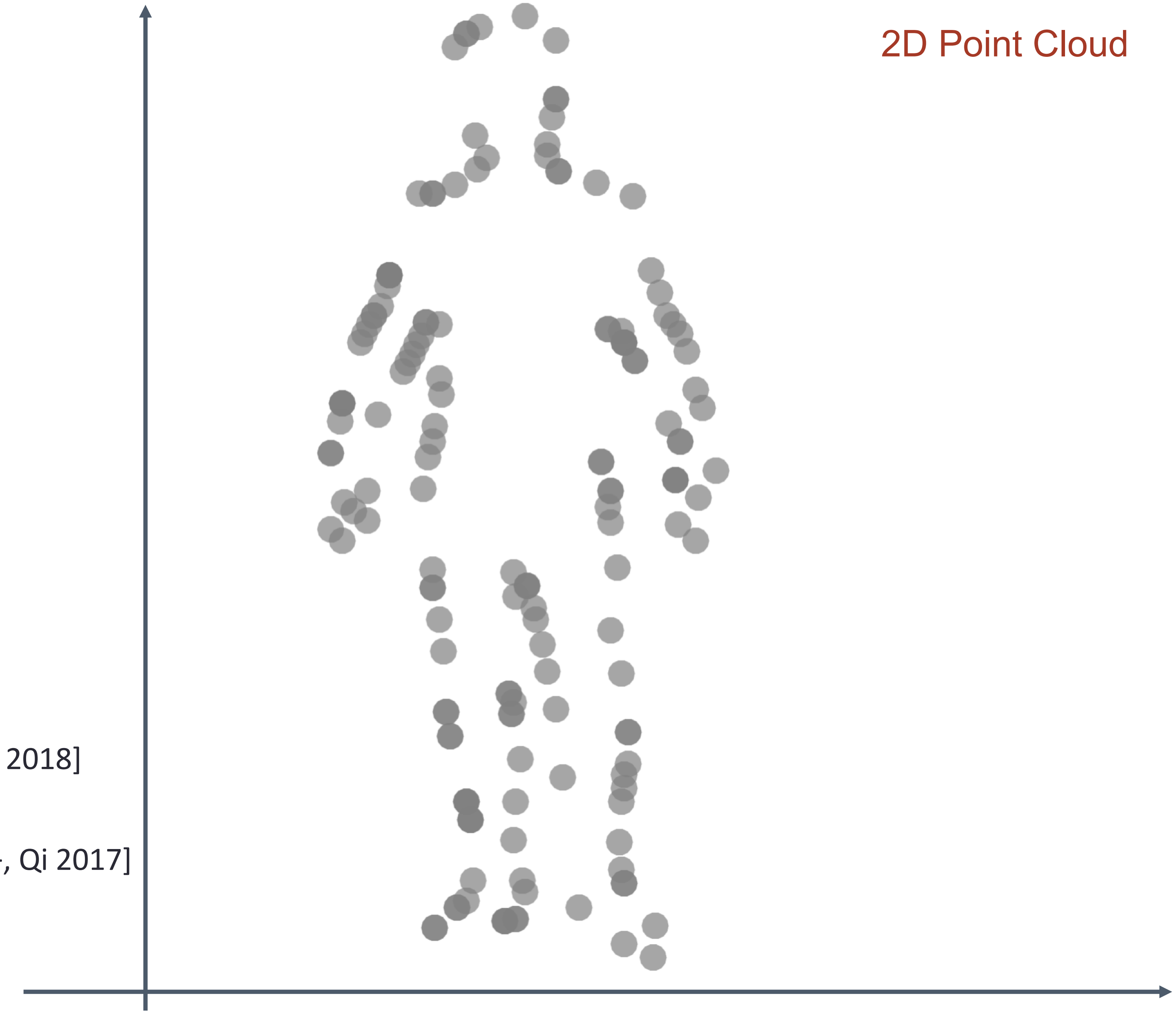
Related Work

- 1×1 Convolution [PointNet, Qi 2017]
    - Order Equivariance
    - No neighborhood information
- Graph Convolution
  [Velickovic, 2017] , [Simonovsky, 2017], [Wang, 2018]



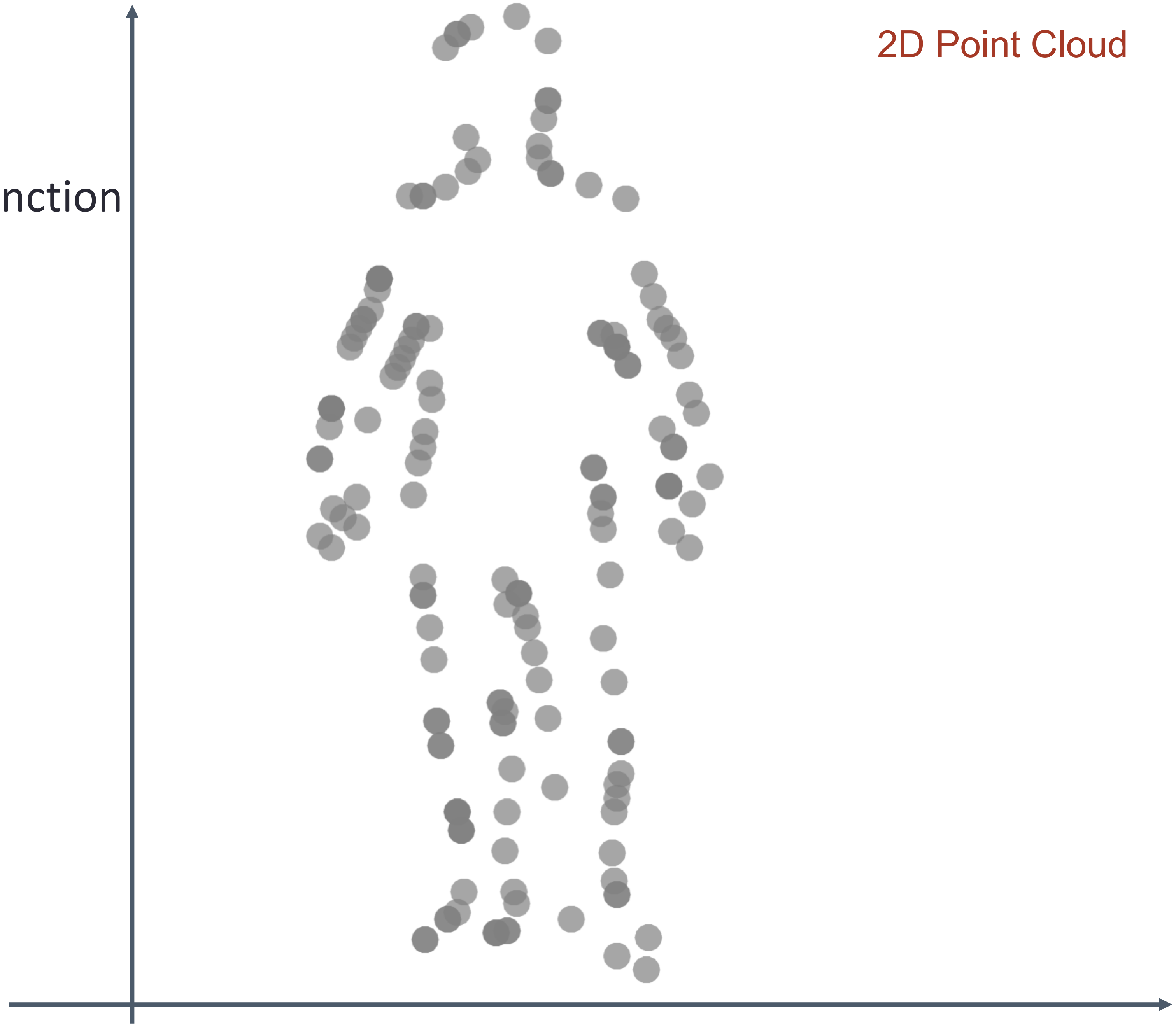
Related Work

- 1×1 Convolution [PointNet, Qi 2017]
  - Order Equivariance
  - No neighborhood information
- Graph Convolution
  [Velickovic, 2017] , [Simonovsky, 2017], [Wang, 2018]
- Varying sampling densities [PointNet++, Qi 2017]



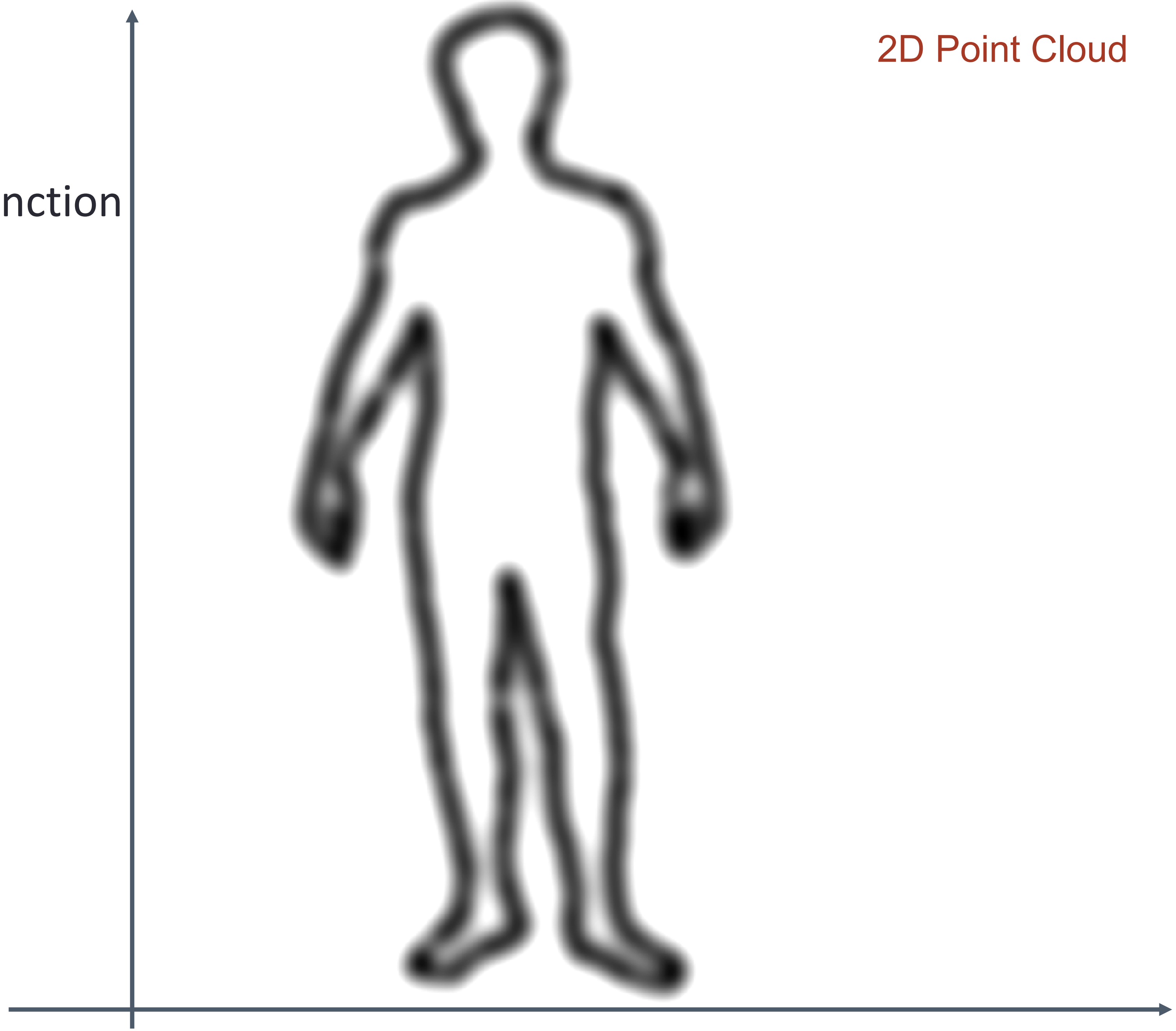
Our Approach

- Extend the sample to a continuous function



Our Approach

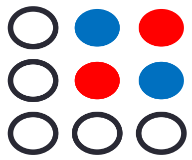
- Extend the sample to a continuous function



Our Approach

- Extend the sample to a continuous function

- Kernel definition

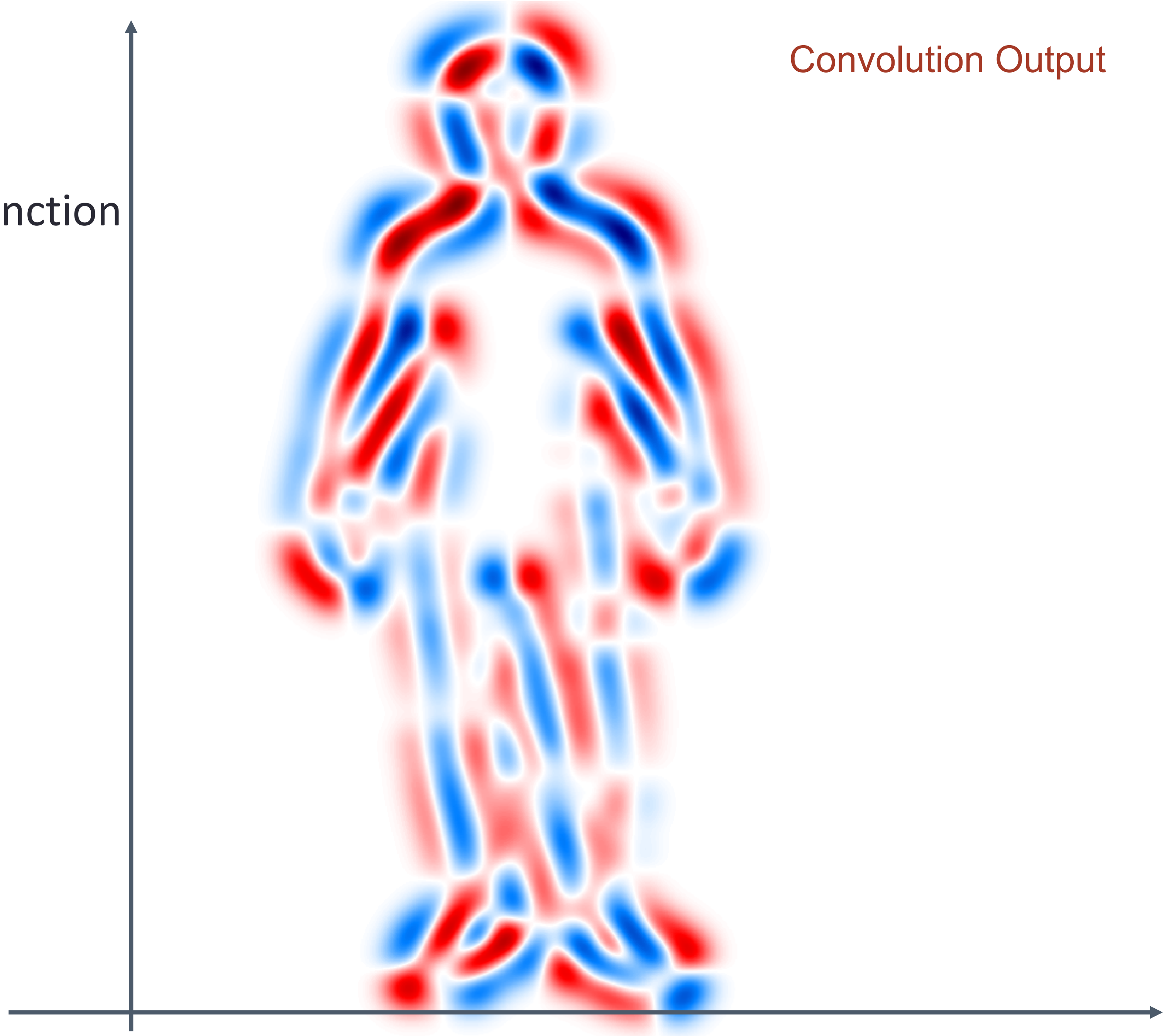


Kernel Point Cloud

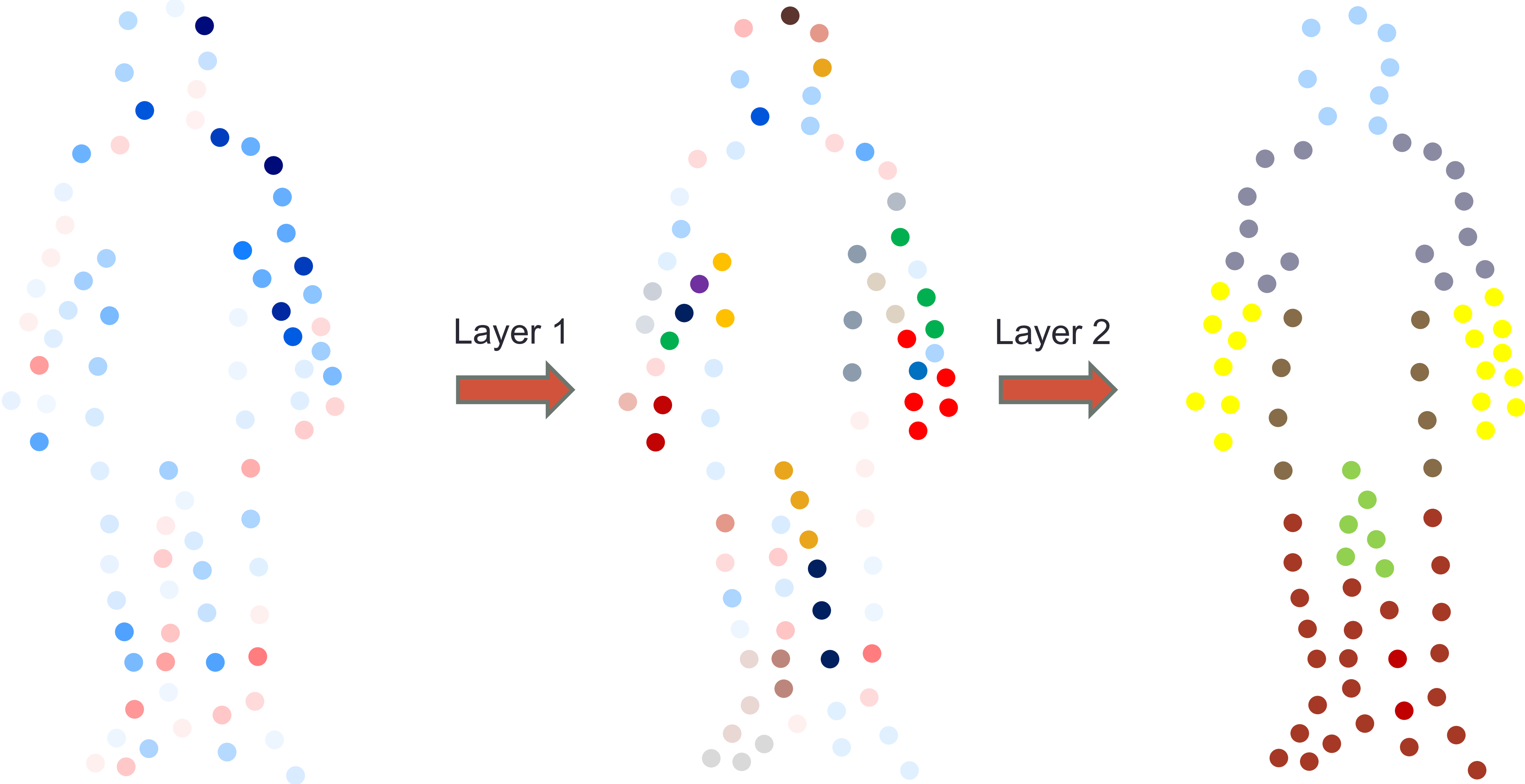
Kernel Extension

- Convolution in the ambient space

$$f * k\,(x) = \int_{\mathbb{R}^2} f(y)k(x-y)dy$$



Our Approach

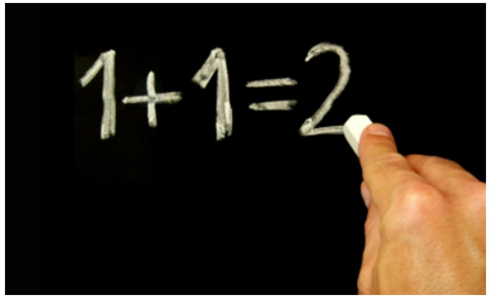


Question:

What **properties** should **point cloud** convolutional layer satisfy?

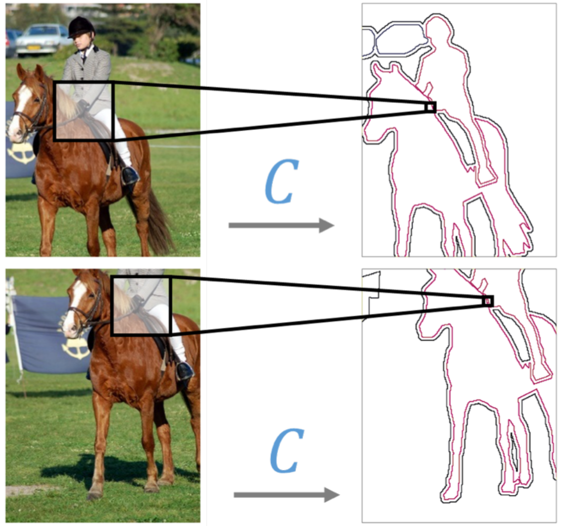
Image Convolution Properties

- Simplicity (Sparse+Linear)



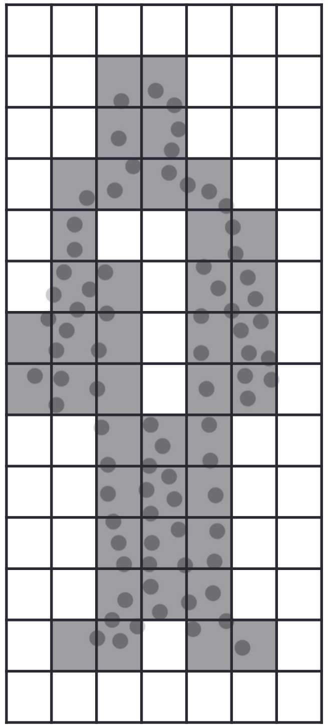
69

- Translation equivariance and locality



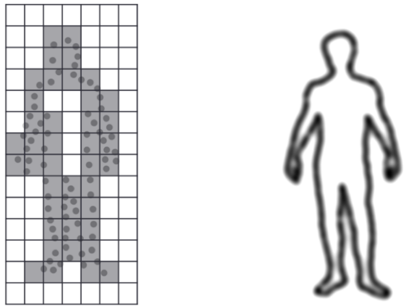
Point Cloud Convolution Properties

- Work in correct dimensionality



Point Cloud Convolution Properties

- Work in correct dimensionality

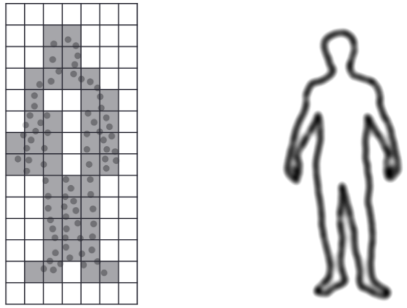


- Order equivariance

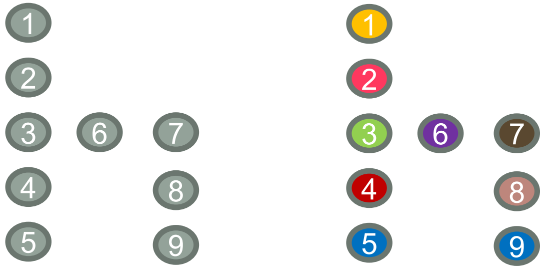
Input Output

Point Cloud Convolution Properties

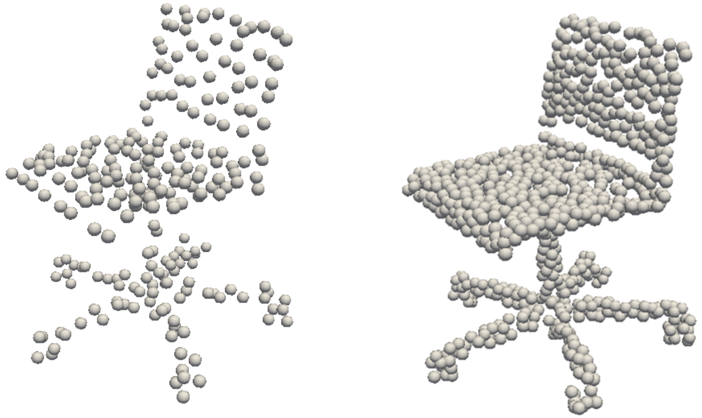
- Work in correct dimensionality



- Order equivariance

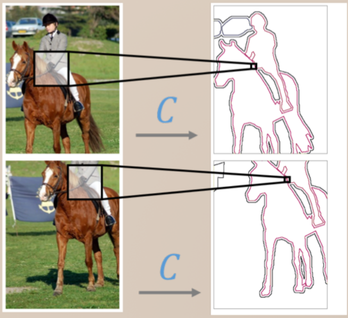


- Robustness to sampling

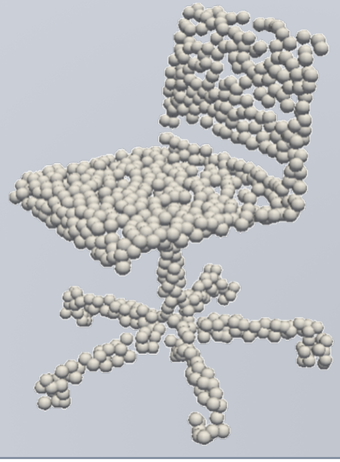


Point Cloud Convolution Properties

- Simplicity (Sparse+Linear)
- Translation equivariance and locality



- Work in correct dimensionality
- Order equivariance
- Robustness to sampling



Extension Operator

- $\mathcal{R}_X \circ O \circ \mathcal{E}_X$
- Given a point cloud function $\{f_i\}$
- Define

Gaussian (RBF)

$$\mathcal{E}_X[f](x) = \sum \alpha_\mathrm{i} \Phi(x - x_i)$$

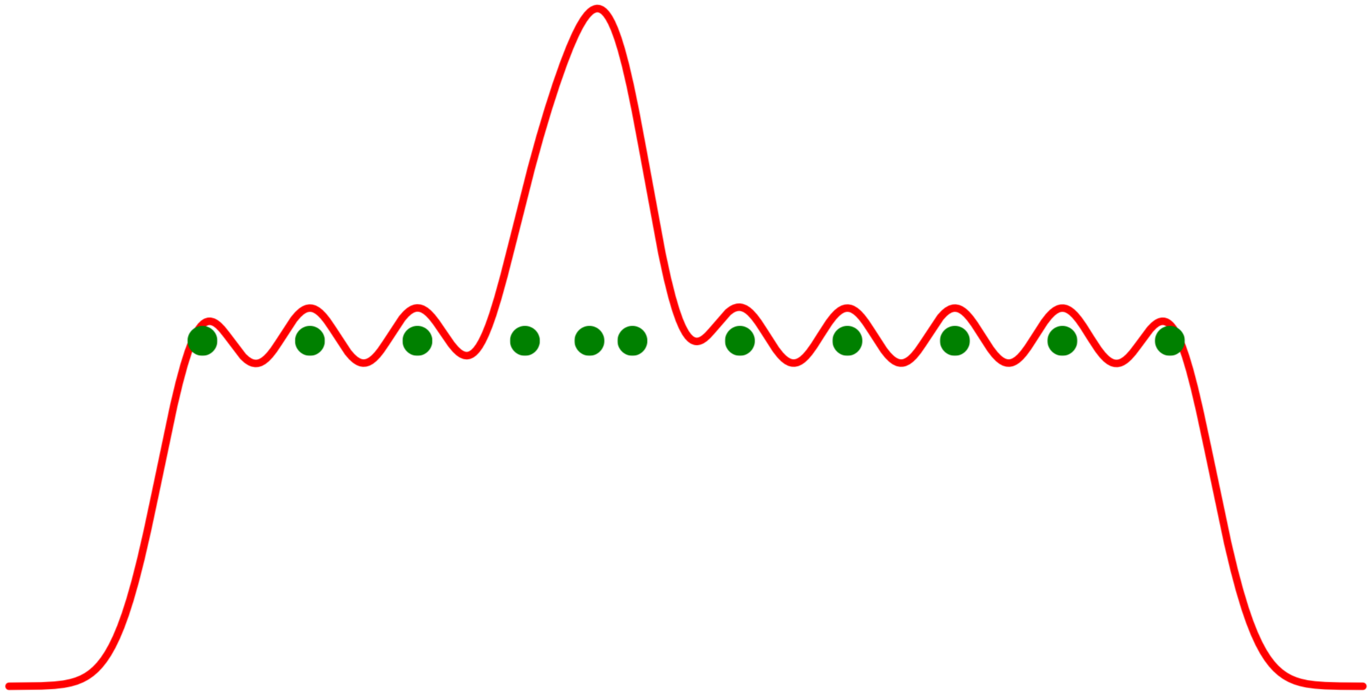
?

$$\alpha_i = f_i w_i$$

Choice of weights

$$\mathcal{E}_X[f](x) = \sum \alpha_i \Phi(x - x_i),\ \alpha_i = f_i w_i$$

- $w_i = 1$
  - Will work poorly for varying densities

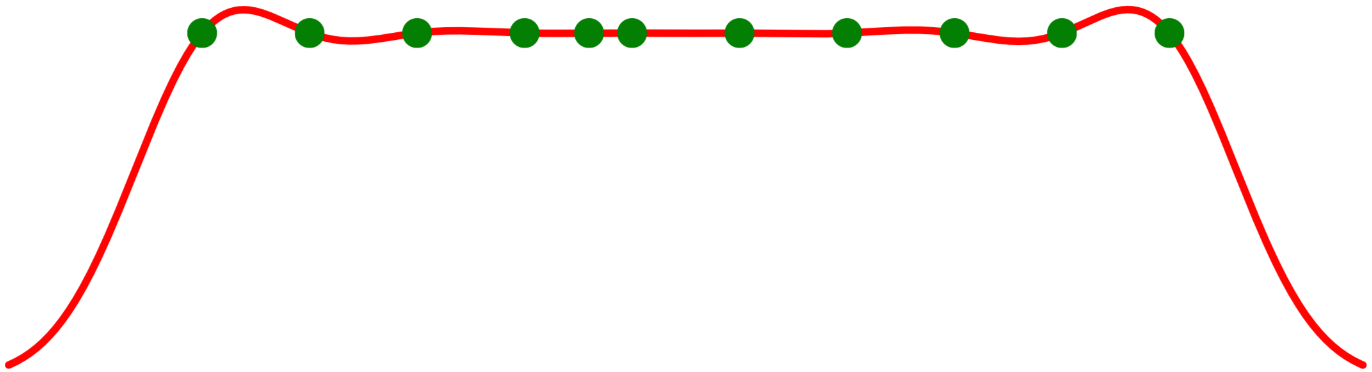


Choice of weights

$$\mathcal{E}_X[f](x) = \sum \alpha_i \Phi(x - x_i),\ \alpha_i = f_i w_i$$

- $w_i = 1$
  - Will work poorly for varying densities

- Interpolation: $\alpha = A^{-1} f$ where $A_{ij} = \Phi\left(x_i - x_j\right)$
  - Requires inverting a dense matrix
  - Numerical issues



Choice of weights

$$\mathcal{E}_X[f](x) = \sum \alpha_i \Phi(x - x_i),\ \alpha_i = f_i w_i$$

- $w_i = 1$
  - Will work poorly for varying densities
- Interpolation: $\alpha = A^{-1} f$ where $A_{ij} = \Phi(x_i - x_j)$
  - Requires inverting a dense matrix
  - Numerical issues
- Approximation: $w_i = c\Omega_i$
  - Converges to the original function
  - Efficient computation by density estimation

Voronoi Diagram

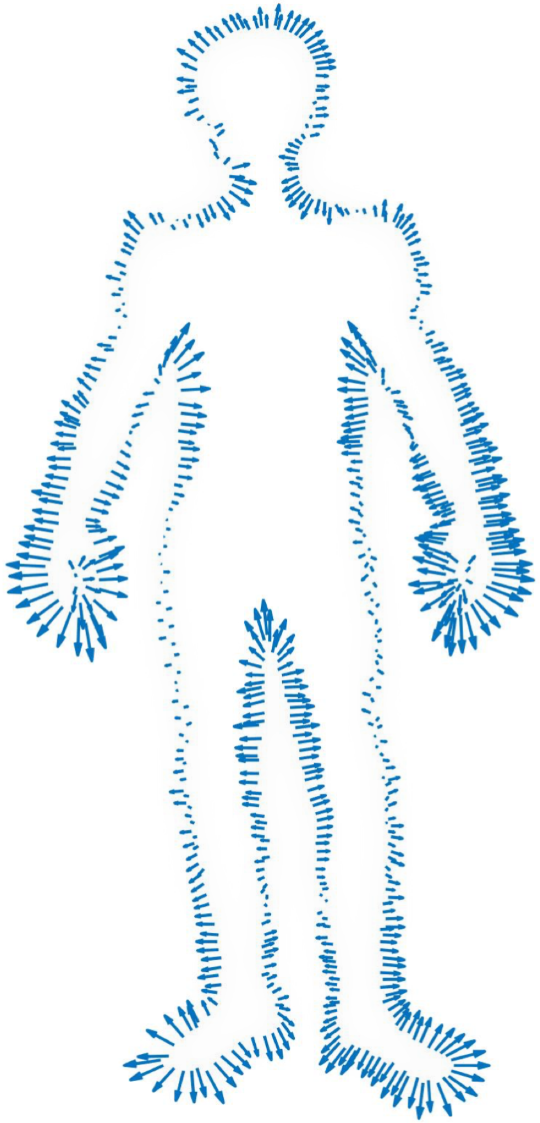
Extension operator

$$\mathcal{E}_X[f](x) = \sum \alpha_i \Phi(x - x_i)$$

- Extension of the constant function $f_i \equiv 1$

- $\nabla \mathcal{E}_S[\mathbf{1}] \rightarrow -H \cdot n$

- Input to first layer of the network



Continuous Kernel Model

- $\mathcal{O}_X = \mathcal{R}_X \circ \mathcal{O} \circ \mathcal{E}_X$

- The kernel is a weighted sum of RBFs

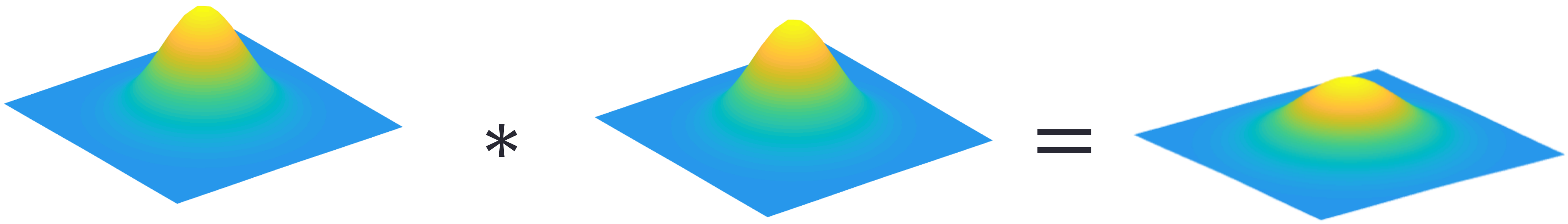
$$k(x) = \sum_{l=1}^{L} k_l \, \Phi(x - T_l)$$

Learnable parameters

Kernel layout
(can also be learnt)

Convolution Calculation

- Φ is a Gaussian
  - Convolution of two gaussians is another gaussian



- Closed form convolution

$$\mathcal{E}_X * k(x) = \sum_{i,l} \alpha_i k_l \, \Phi(x - x_i - T_l)$$

Point Cloud Convolution Operator

$$\mathcal{R}_X \circ \ O \circ \mathcal{E}_X(x_{i'}) = \sum_{i,l} \alpha_i k_l \, \Phi(x_{i'} - x_i - T_l)$$

Properties

$$\mathcal{R}_X \circ O \circ \mathcal{E}_X\boxed{(x_{i'})} = \boxed{\sum_{i,l} \alpha_i k_l \, \Phi(x_{i'} - x_i - T_l)}$$

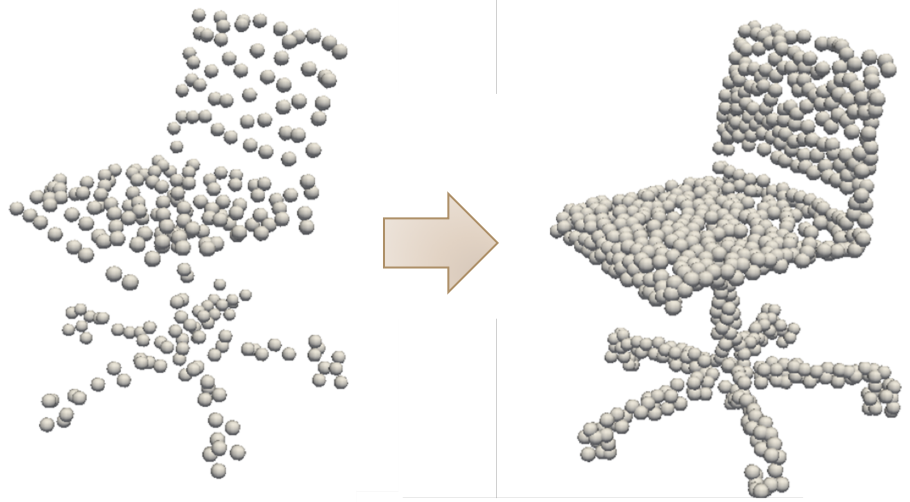
- Translation equivariance
- Locality

By construction

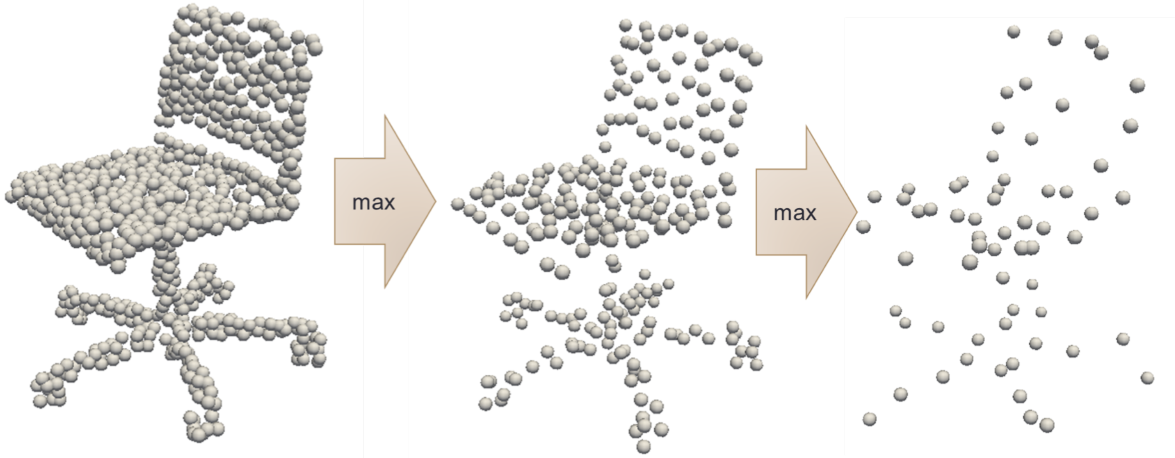
- Linearity $\alpha_i = w_i f_i$
- Operate intrinsically
- Order equivariance
- Robustness to sampling $w_i = c\Omega_i$

Other Operators

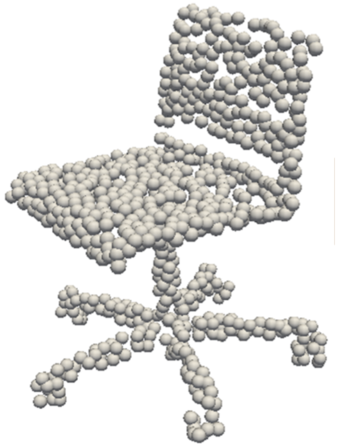
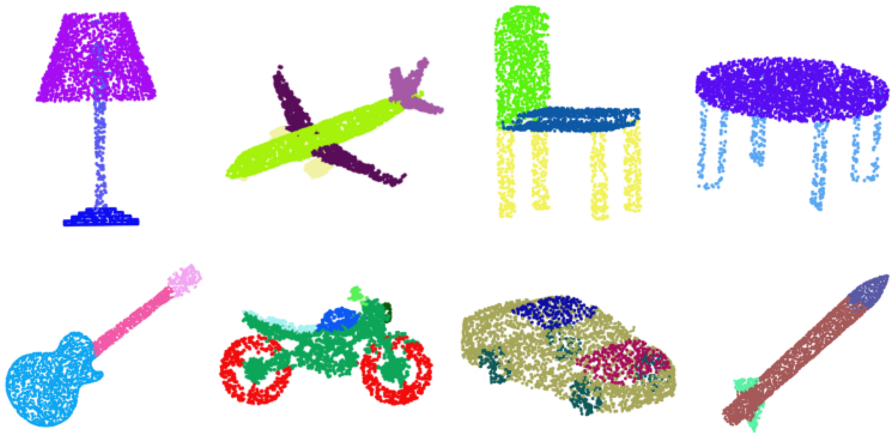
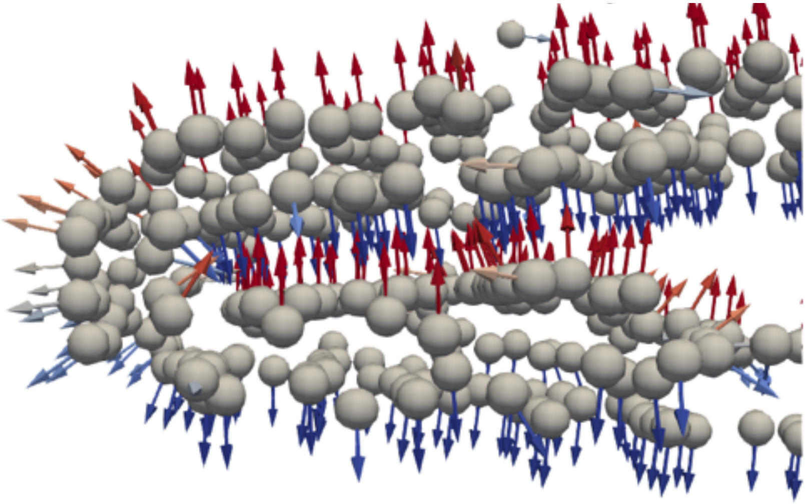
- If O is an operator on volumetric functions: $O_X = \mathcal{R}_X \circ O \circ \mathcal{E}_X$
- Upsampling



- Pooling

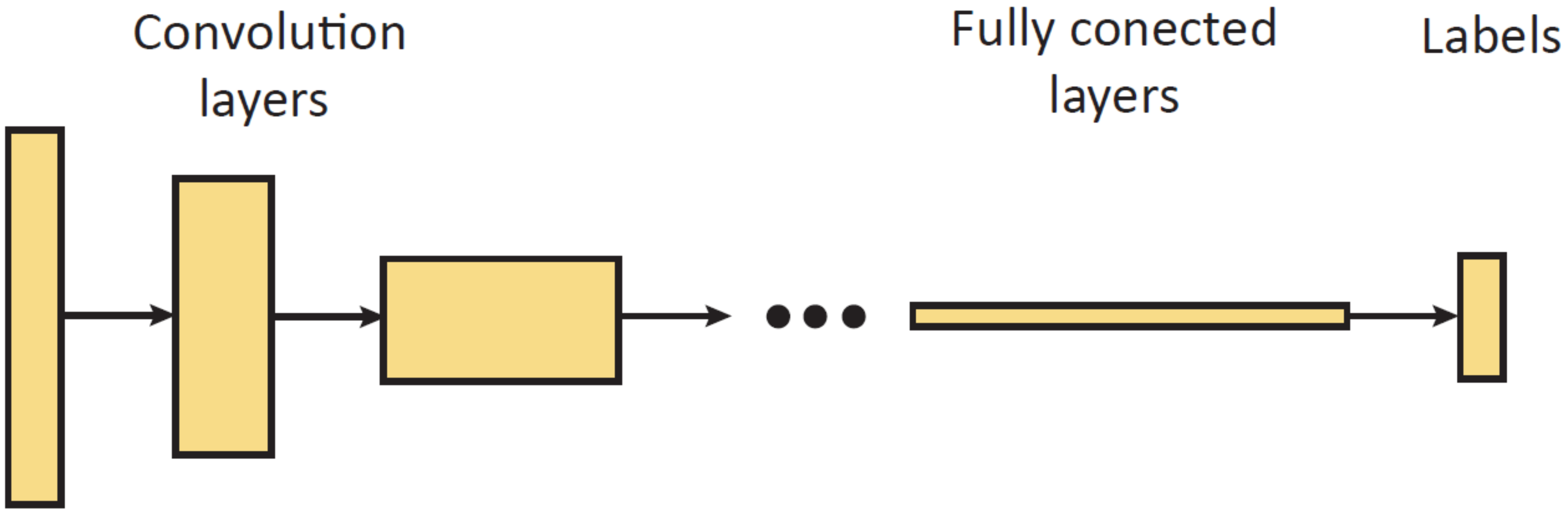


Applications

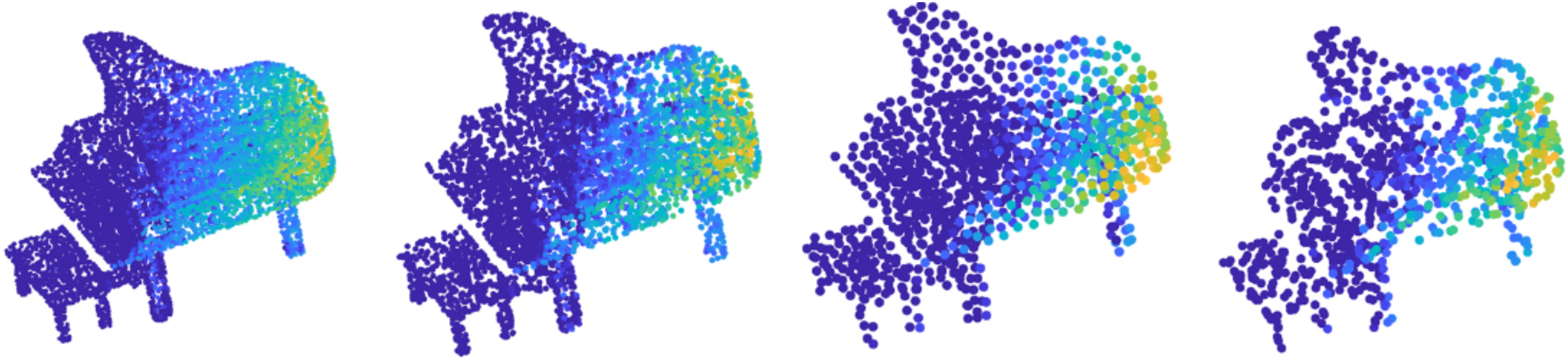




Classification – standard architecture

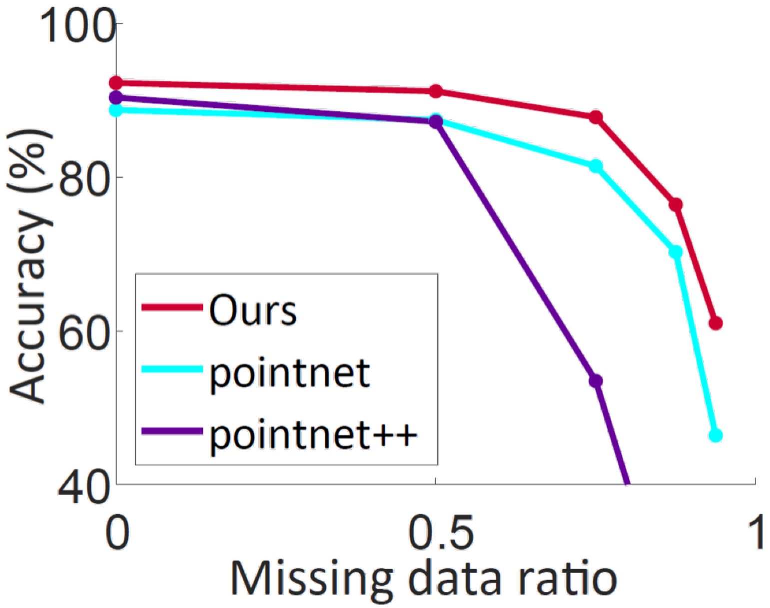
- Concatenation of convolution and pooling as in images



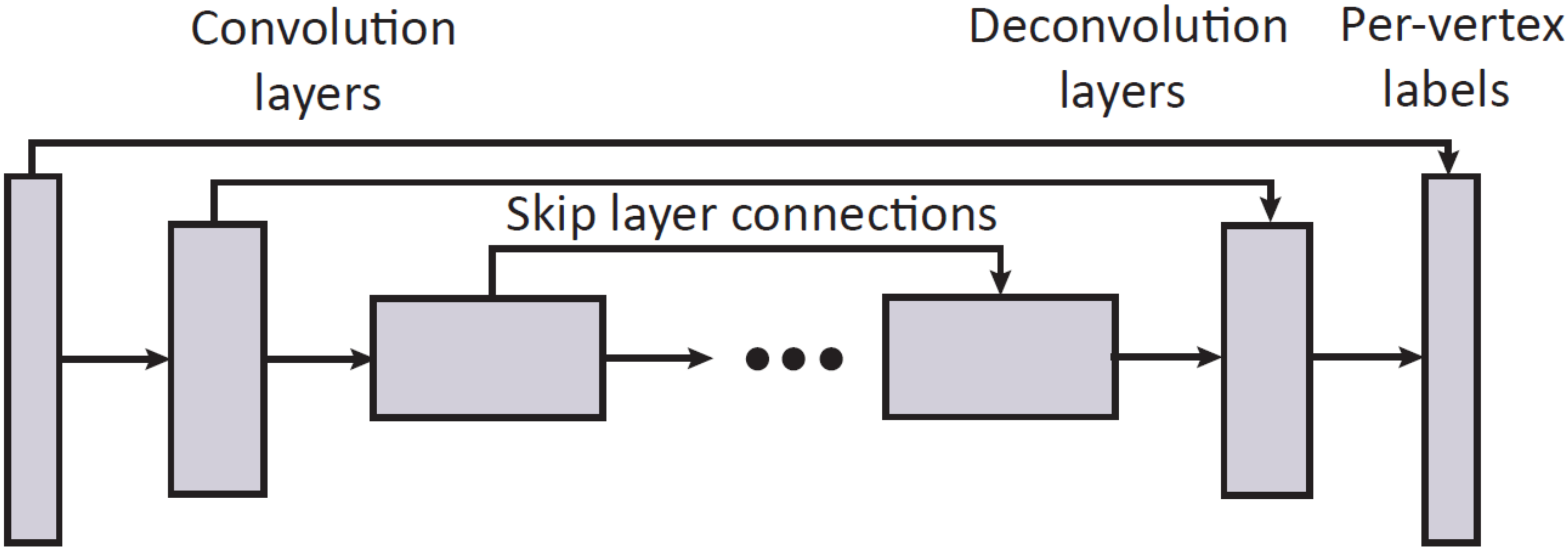
Classification - Robustness to sampling



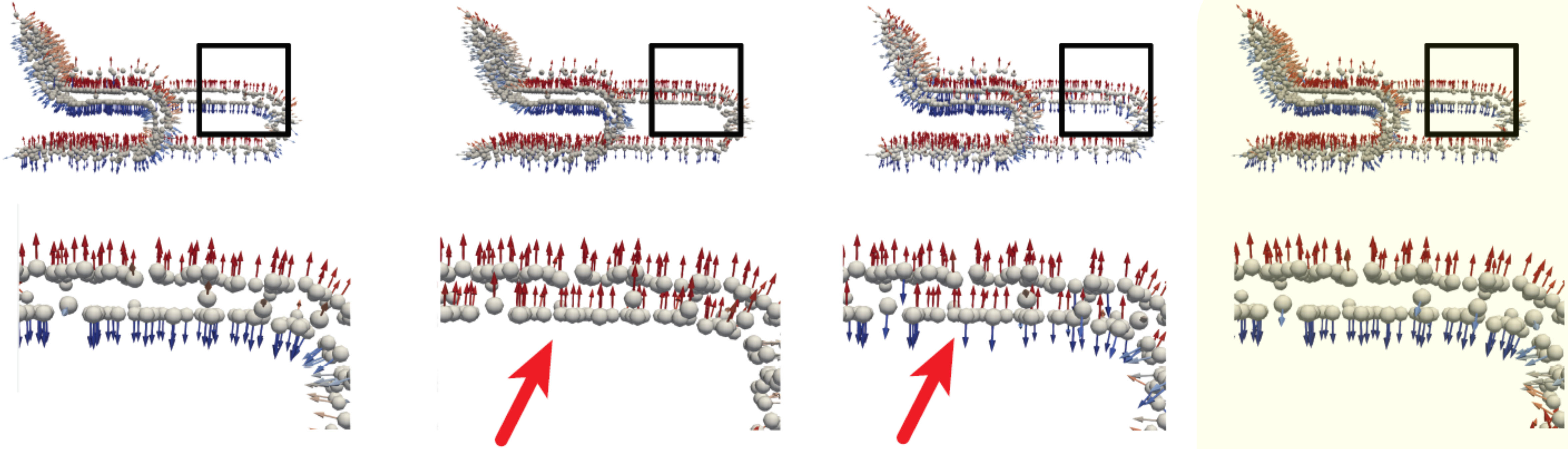
10K points 5K points (random sampling) 1K points (uniform sampling) 1K points (random sampling)



Normal Estimation



Normal Estimation



Ground Truth Pointnet Pointnet++ Ours

Summary

- New framework for learning on point clouds
- Generalizes regular CNNs and PointNet
- State of the art results
- Limitations and future work directions
  - Sparse efficient implementation is needed
  - Different RBF for better reconstruction
  - Apply to irregular sampling of images

The End

- Code is online

  http://www.wisdom.weizmann.ac.il/~haggaim

- Support
  - ERC Starting Grant (SurfComp)
  - Israel Science Foundation
  - I-CORE