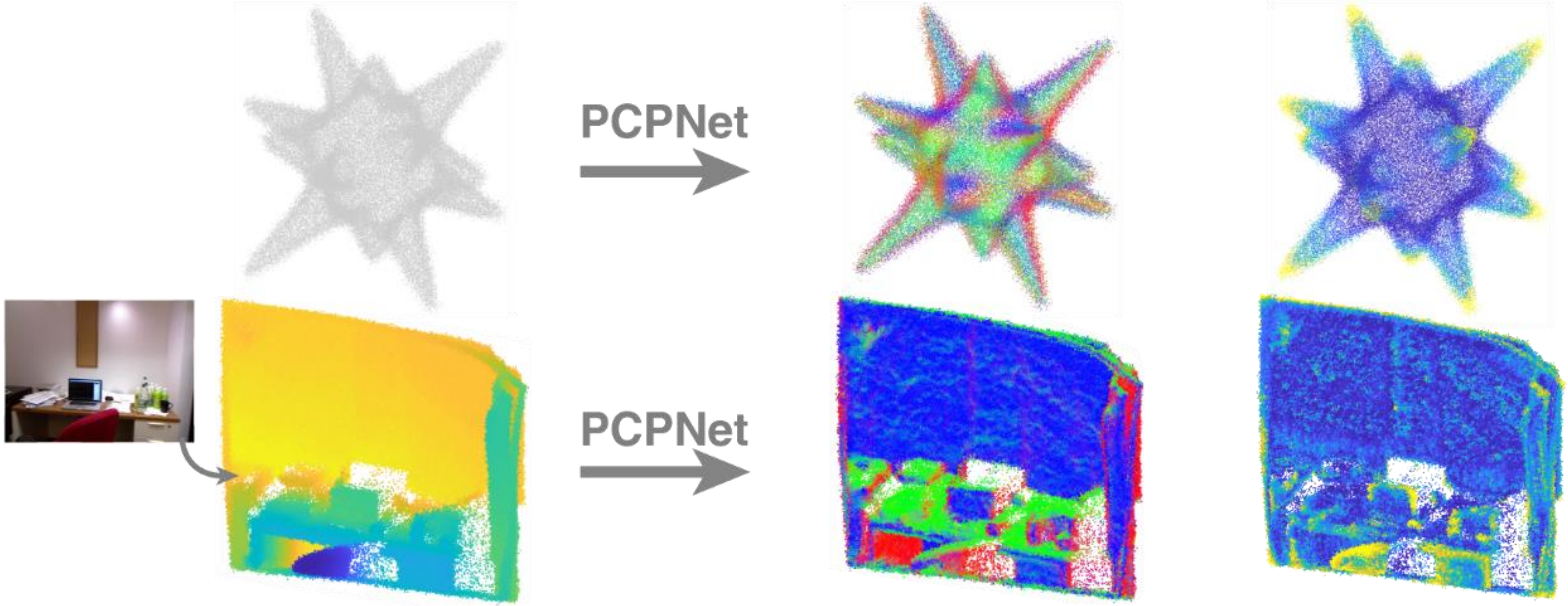
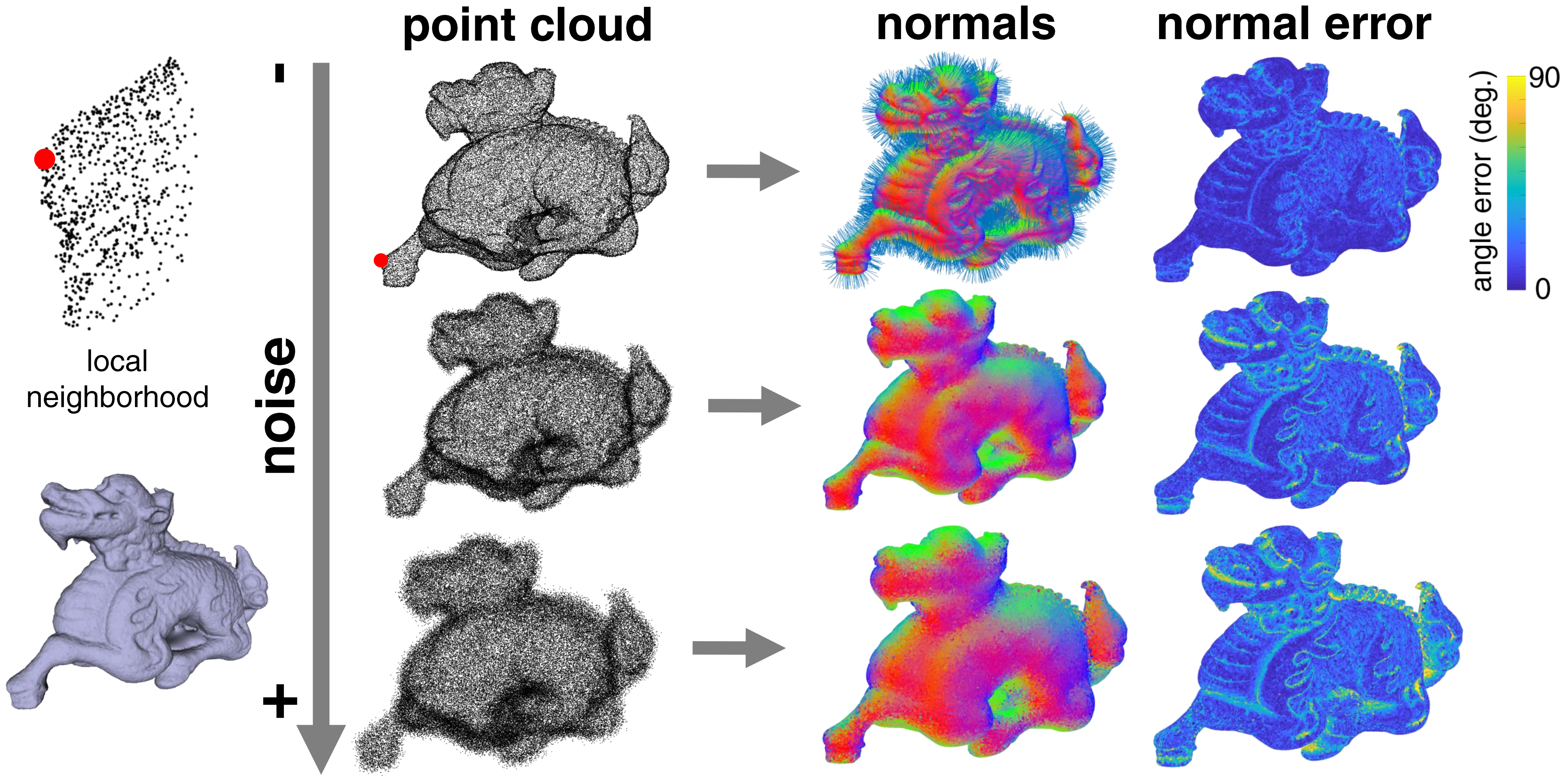


PCPNet: Learning Local Shape Properties from Raw Point Clouds

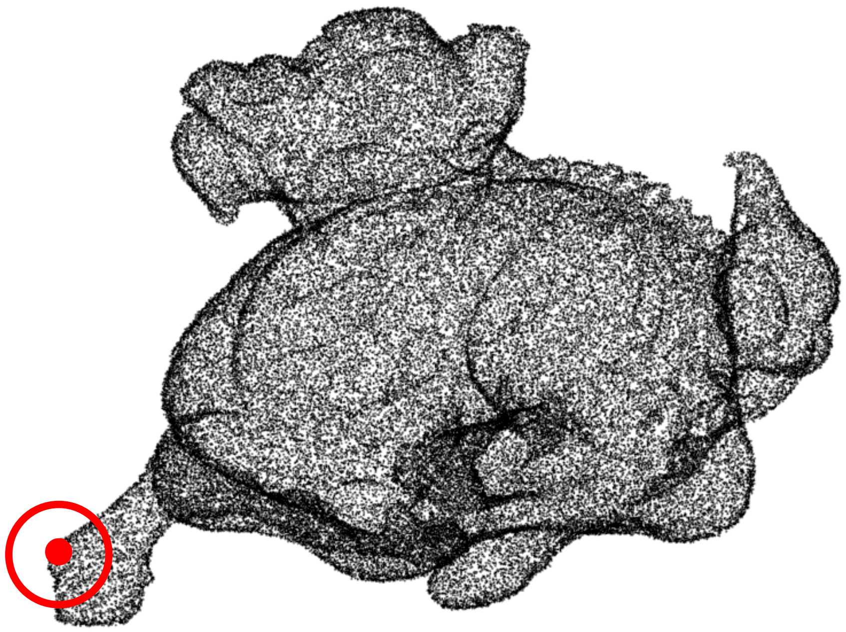
Paul Guerrero — University College London
Yanir Kleiman — LIX, École Polytechnique
Maks Ovsjanikov — LIX, École Polytechnique
Niloy J. Mitra — University College London

EUROGRAPHICS 2018

Estimating Properties of a Point Cloud

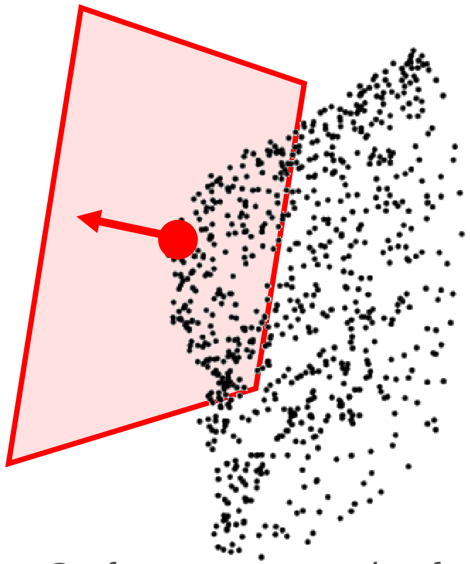


Traditional Approaches



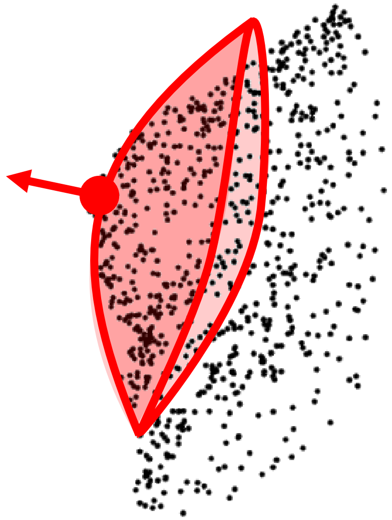
Examples:

PCA



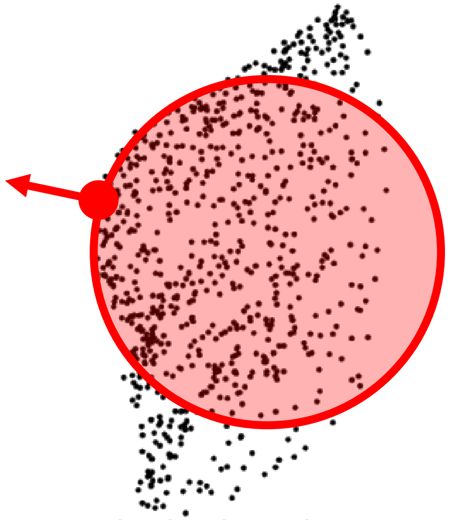
Surface reconstruction from unorganized points,
Hoppe et al., 1992

Jet fitting



Estimating differential quantities using polynomial fitting of osculating jets,
Cazals and Pouget, 2005

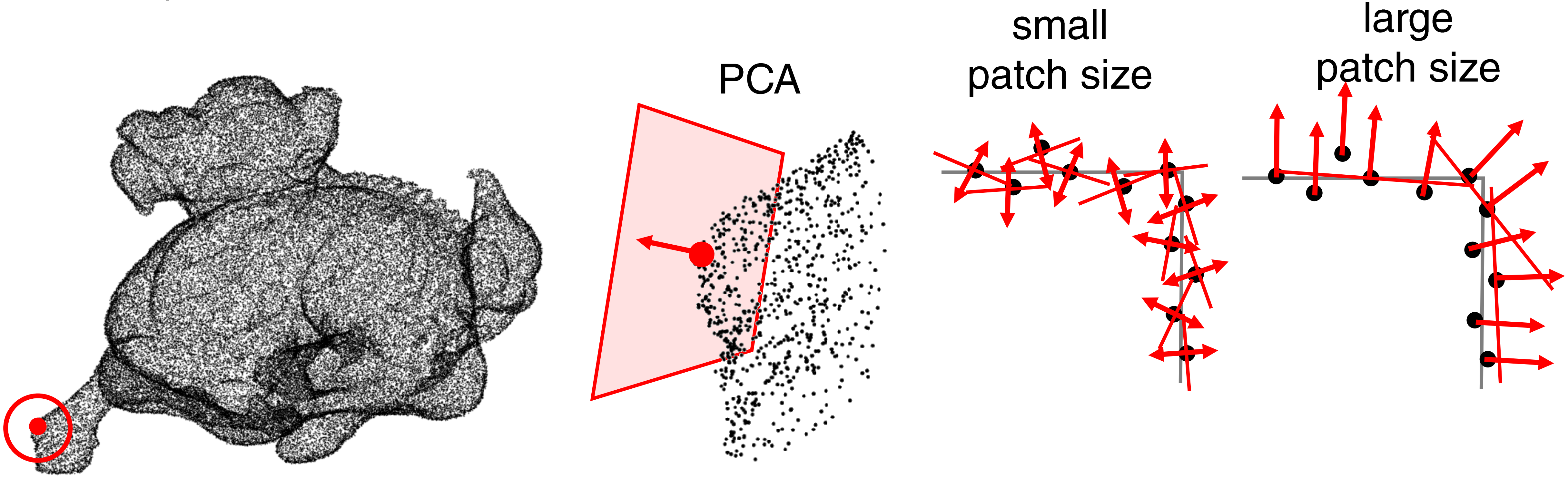
MLS Sphere Fitting



Algebraic Point Set Surfaces,
Guennebaud and Gross, 2007

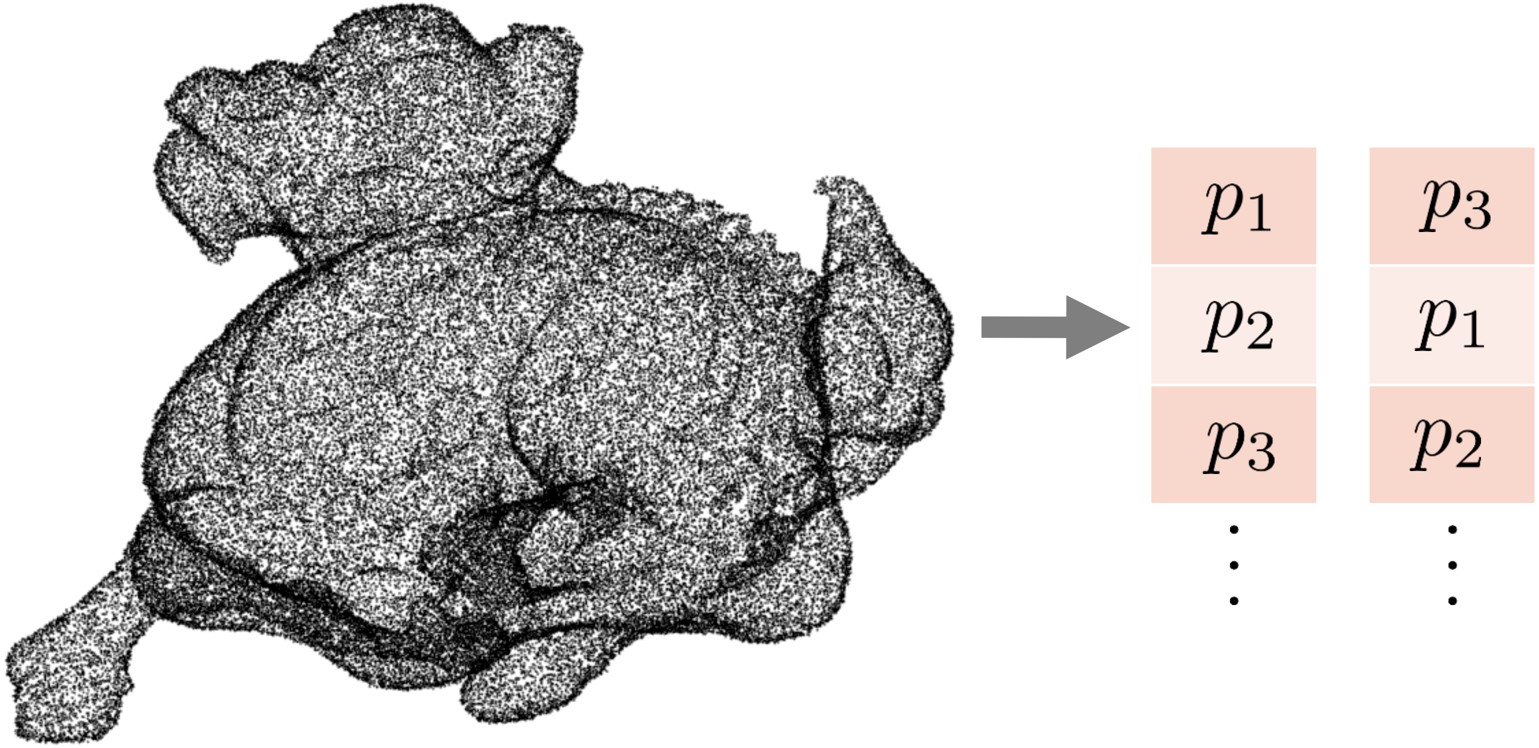
Traditional Approaches

- Sensitive to parameters like patch size
- Acceptable parameter settings depend on data conditions like noise strength, feature size, …



Deep Learning Approaches

- Robust to a large range of conditions
- Problem: invariance to the point order

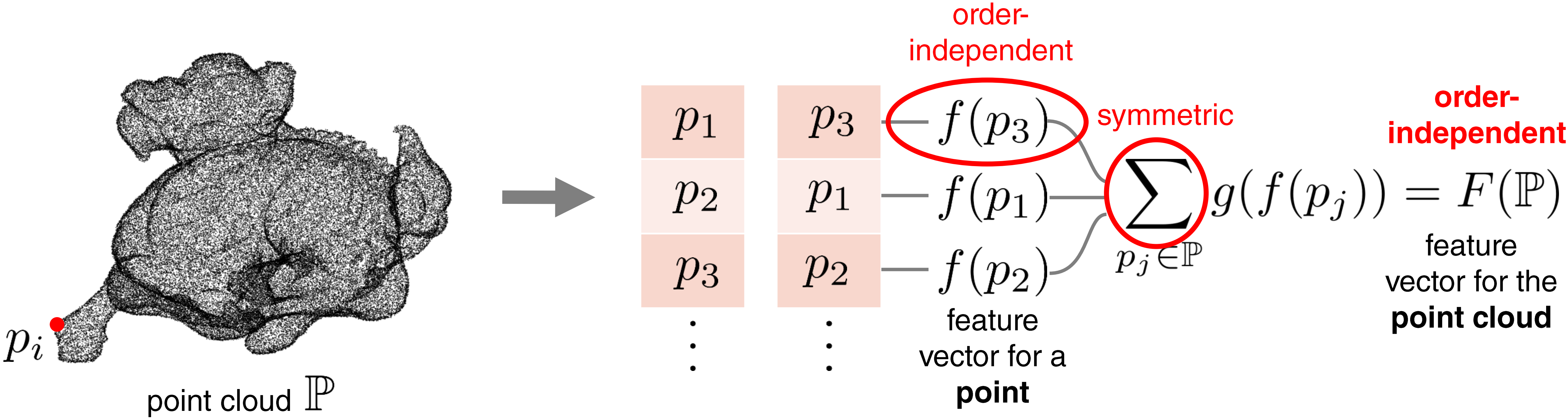


mapping to feature vector:

$$F(p_1, p_2, p_3, \ldots)$$
$$\neq F(p_3, p_1, p_2, \ldots)$$

Deep Learning Approaches: PointNet

- *PointNet: Deep Learning on Point Sets for 3D* ***Classification and Segmentation***, Qi et al., CVPR 2017

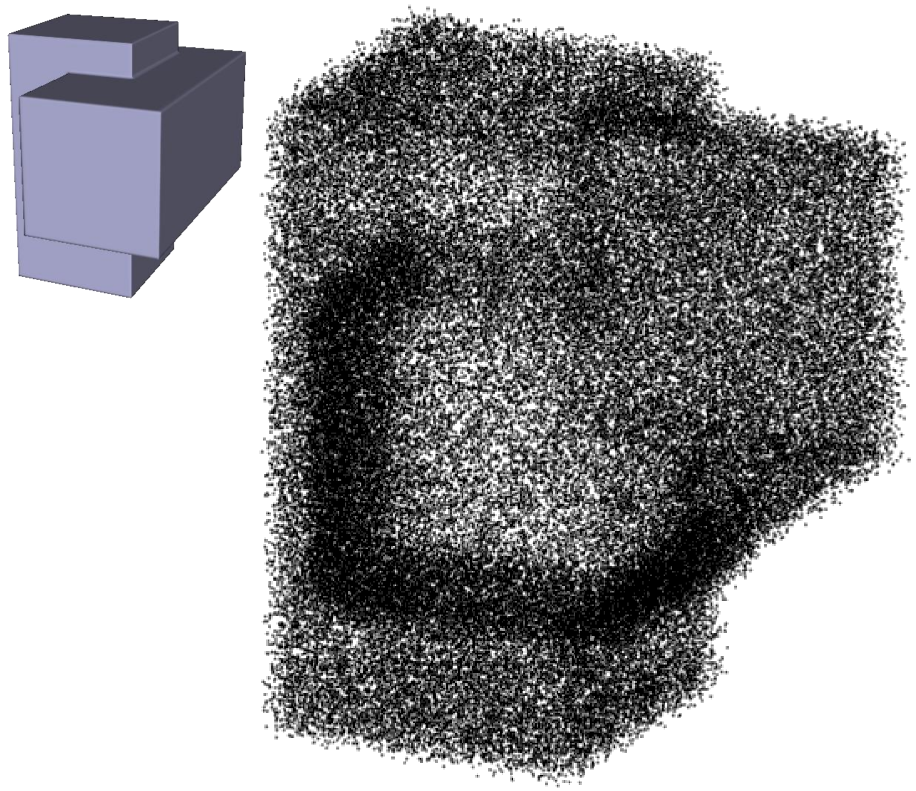


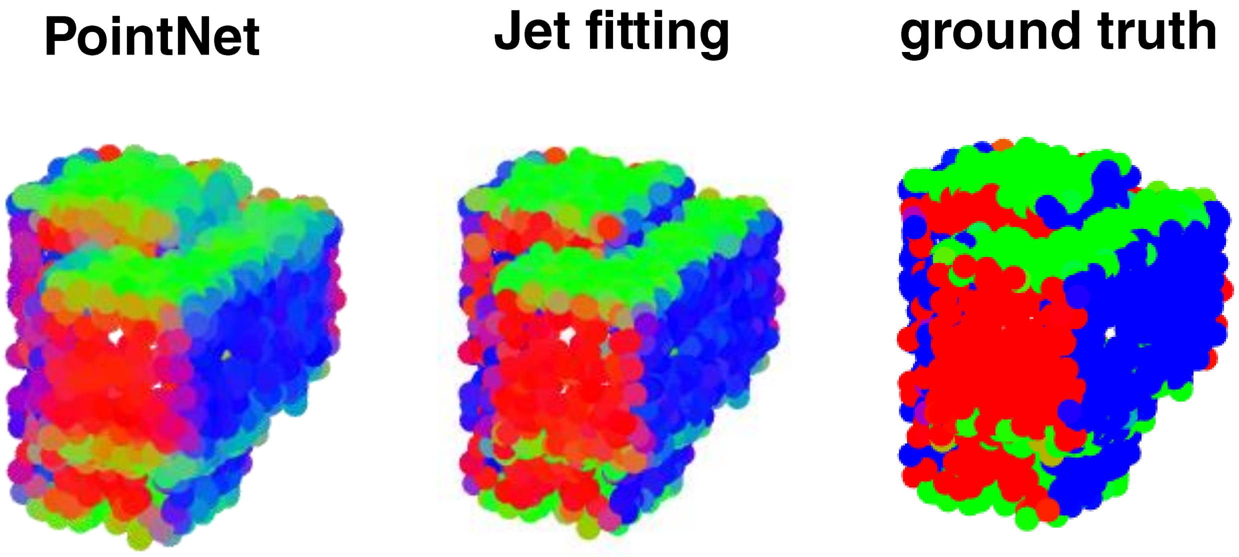
feature vector for local properties of p_i: $[f(p_i), F(\mathbb{P})]$

fully local **fully global**

Deep Learning Approaches: PointNet

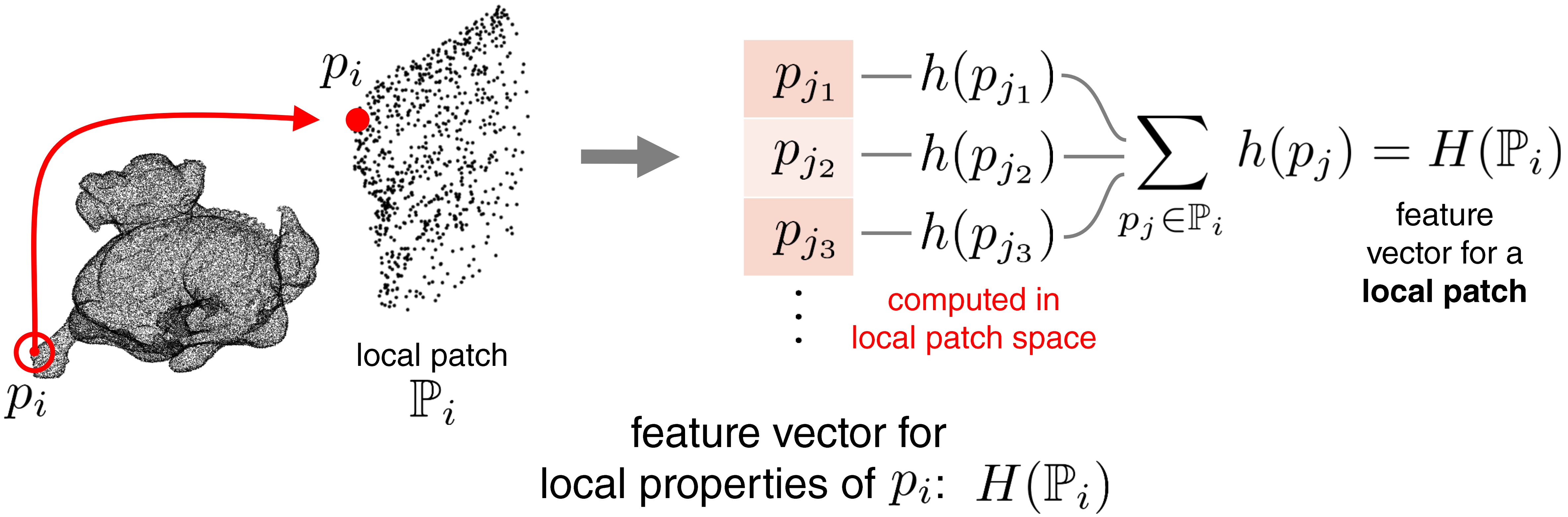
- Using only fully global features or fully local features limits accuracy
- Not well suited for normal estimation



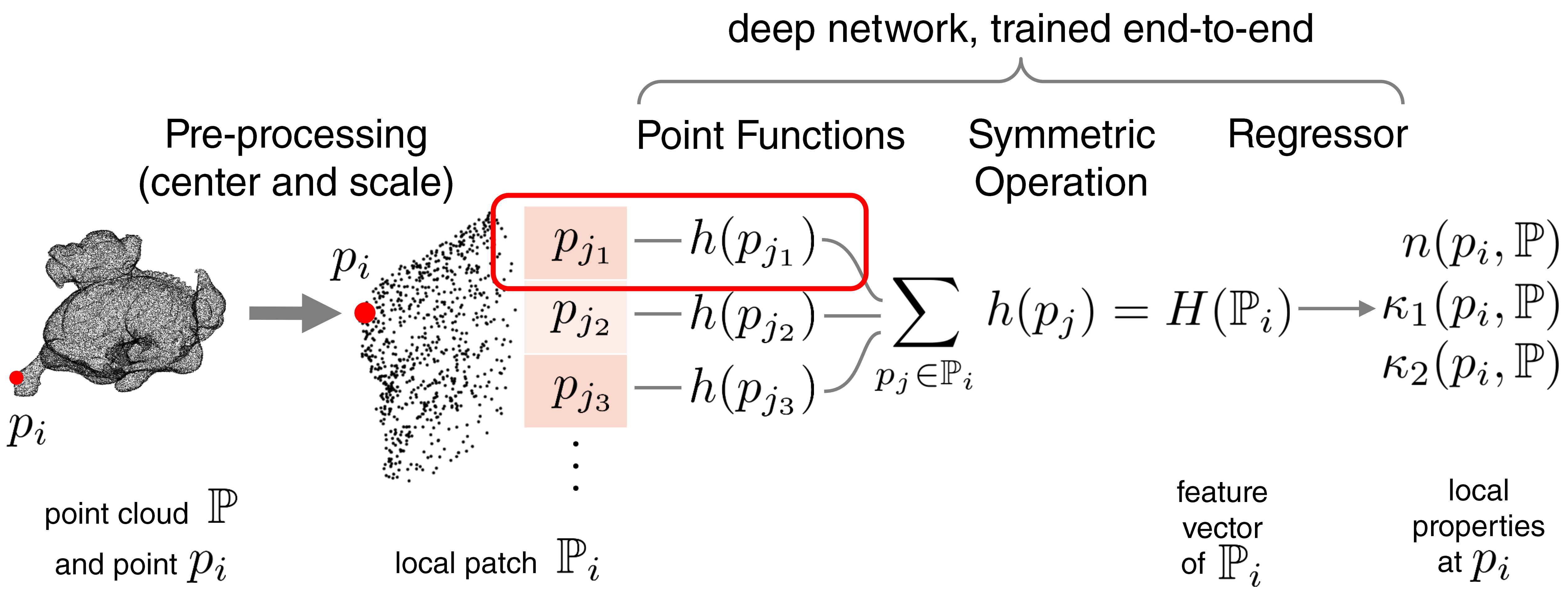


PCPNet

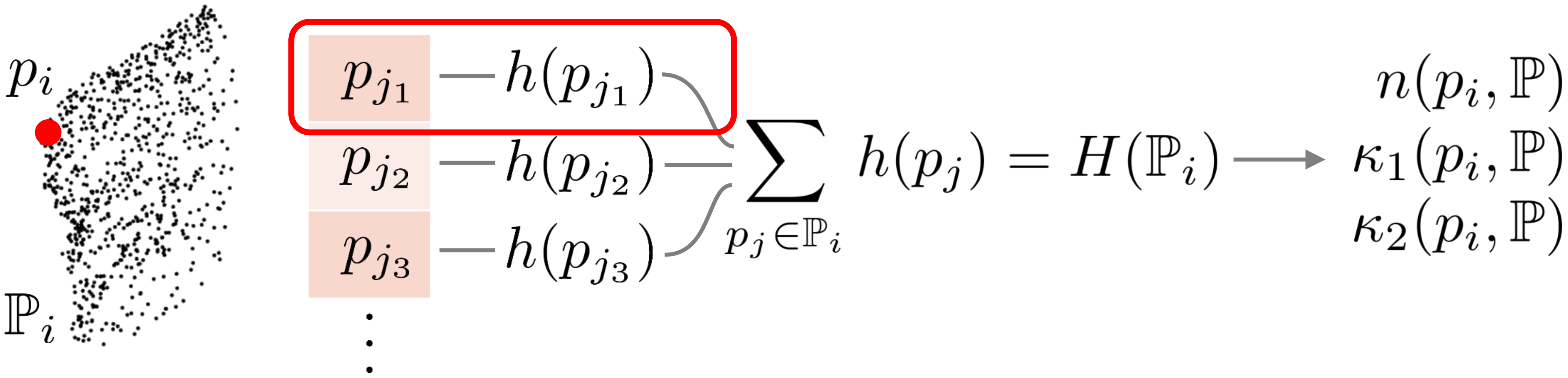
- Instead, using features of a **local patch** gives better accuracy
- State-of-the-art for normal and curvature estimation
- *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*, Qi et al., NIPS 2017



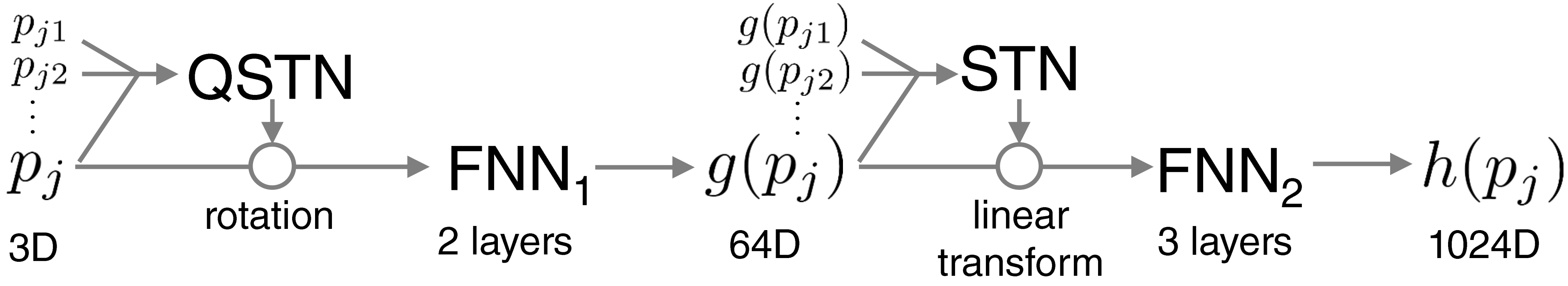
PCPNet Architecture



Point Features



1024D feature vector for each point in the patch:



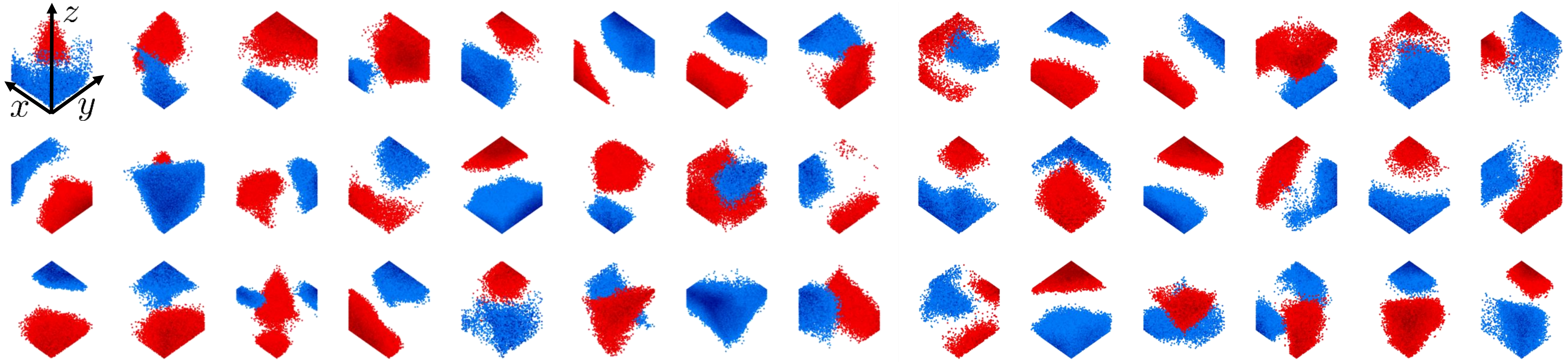
Point Functions: Two Views

$$h(p) = [h_1(p), h_2(p), \ldots, h_{1024}(p)] \text{ with } h_l : \mathbb{R}^3 \longrightarrow \mathbb{R} \qquad H_l(\mathbb{P}_i) = \sum_{p_j \in \mathbb{P}_i} h_l(p_j)$$

- Point functions $h_l(p)$ can be seen as **space probes**
- Sum over all points $H_l(\mathbb{P}_i)$ is a **density estimate**

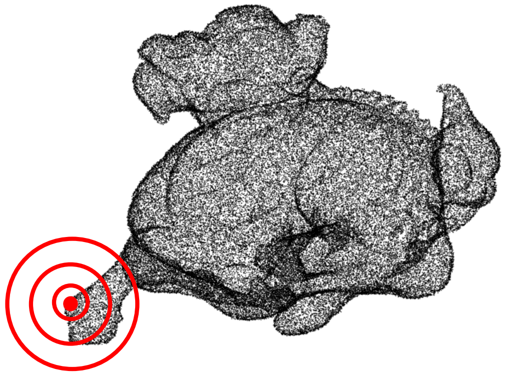
- Point functions $h_l(p)$ can be seen as **convolution kernels**
- Sum over all points $H_l(\mathbb{P}_i)$ is a **convolution**

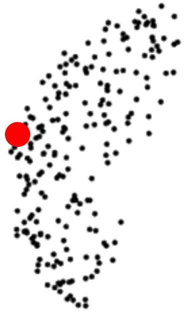
● positive ● negative ○ close to 0 (values have been 0-centered)



Multi-Scale

- Three radii, 3072 point functions, concatenate patch features



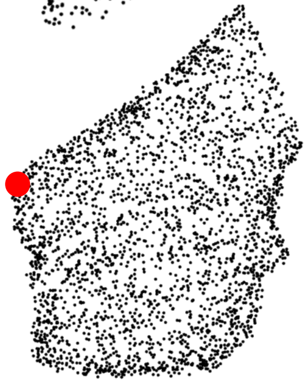


$\mathbb{P}_i^1$

$$\sum_{p_j \in \mathbb{P}_i^1} h(p_j) = H(\mathbb{P}_i^1)$$

$\mathbb{P}_i^2$

$$\sum_{p_j \in \mathbb{P}_i^2} h(p_j) = H(\mathbb{P}_i^2)$$



$\mathbb{P}_i^3$

$$\sum_{p_j \in \mathbb{P}_i^3} h(p_j) = H(\mathbb{P}_i^3)$$

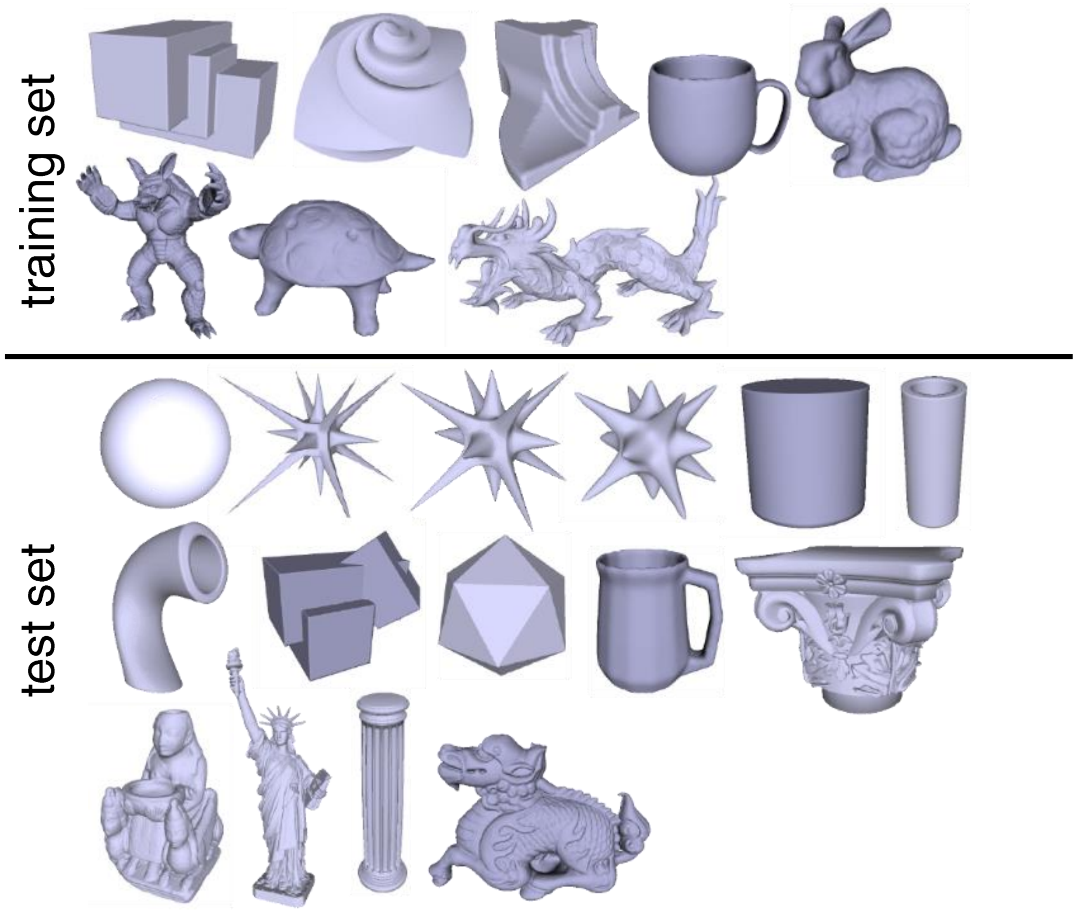
$$\begin{bmatrix} H(\mathbb{P}_i^1) \\ H(\mathbb{P}_i^2) \\ H(\mathbb{P}_i^3) \end{bmatrix} \longrightarrow \begin{matrix} n(p_i, \mathbb{P}) \\ \kappa_1(p_i, \mathbb{P}) \\ \kappa_2(p_i, \mathbb{P}) \end{matrix}$$

Results

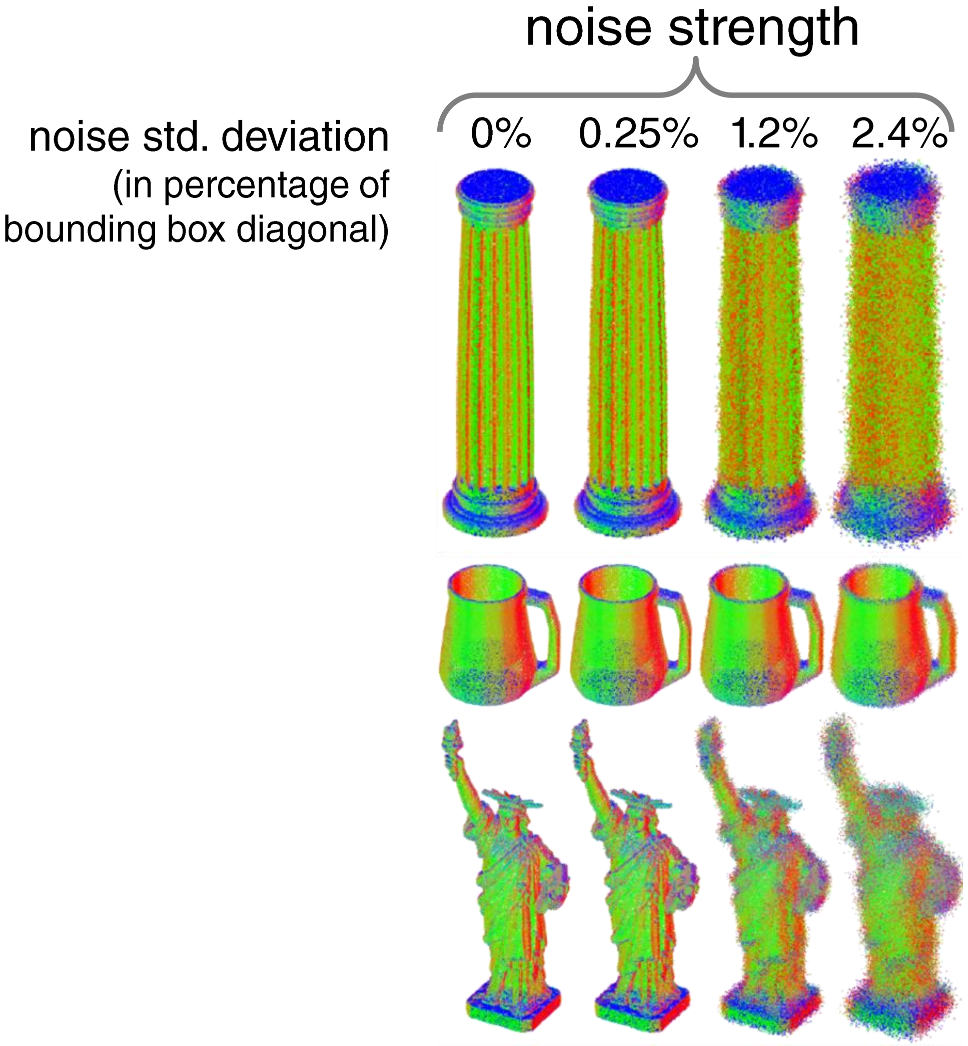
Dataset

Full shape dataset

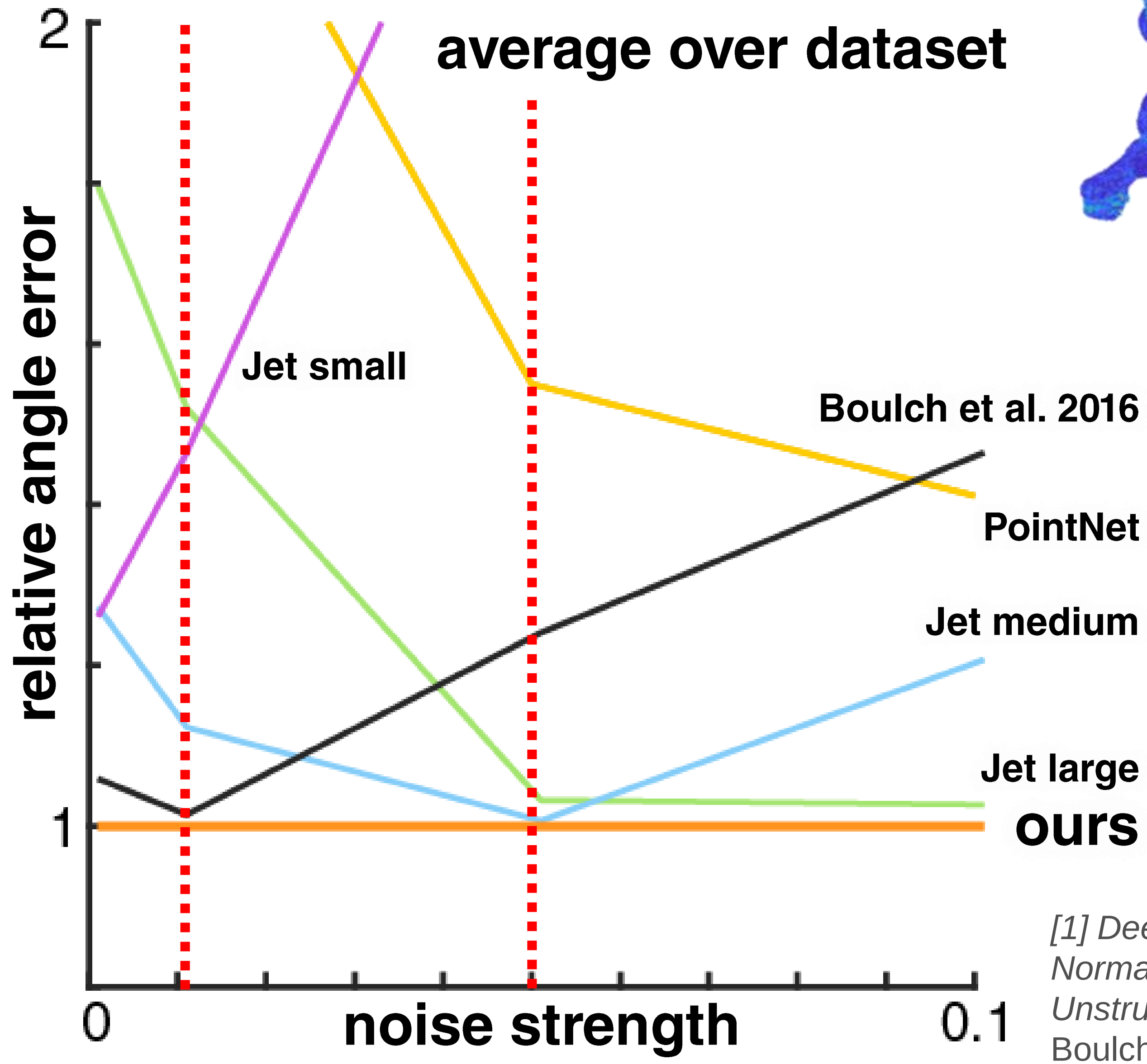
each shape sampled with 100k points
each point can be a patch center

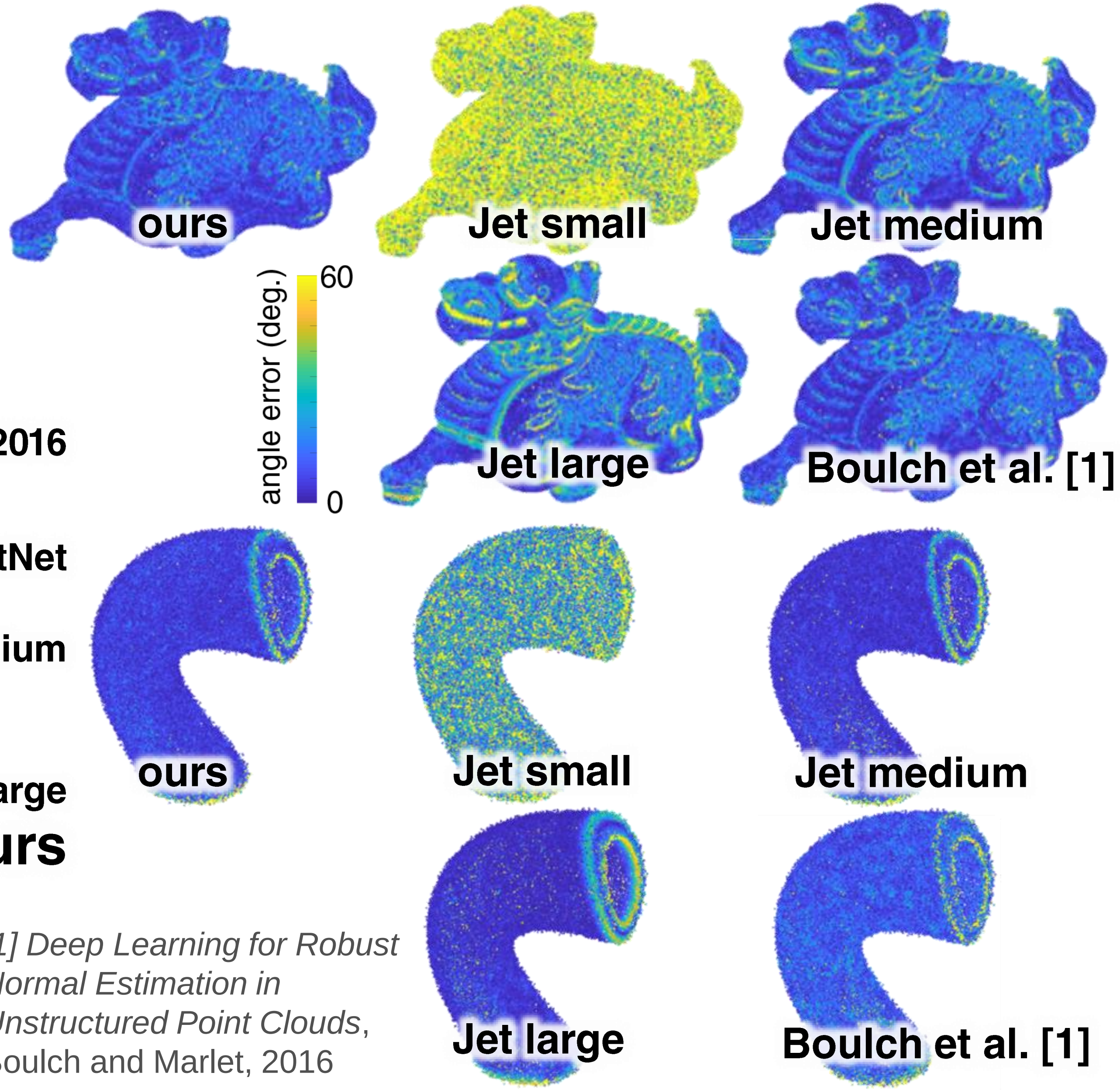


Sampling variations



Unoriented Normal Estimation

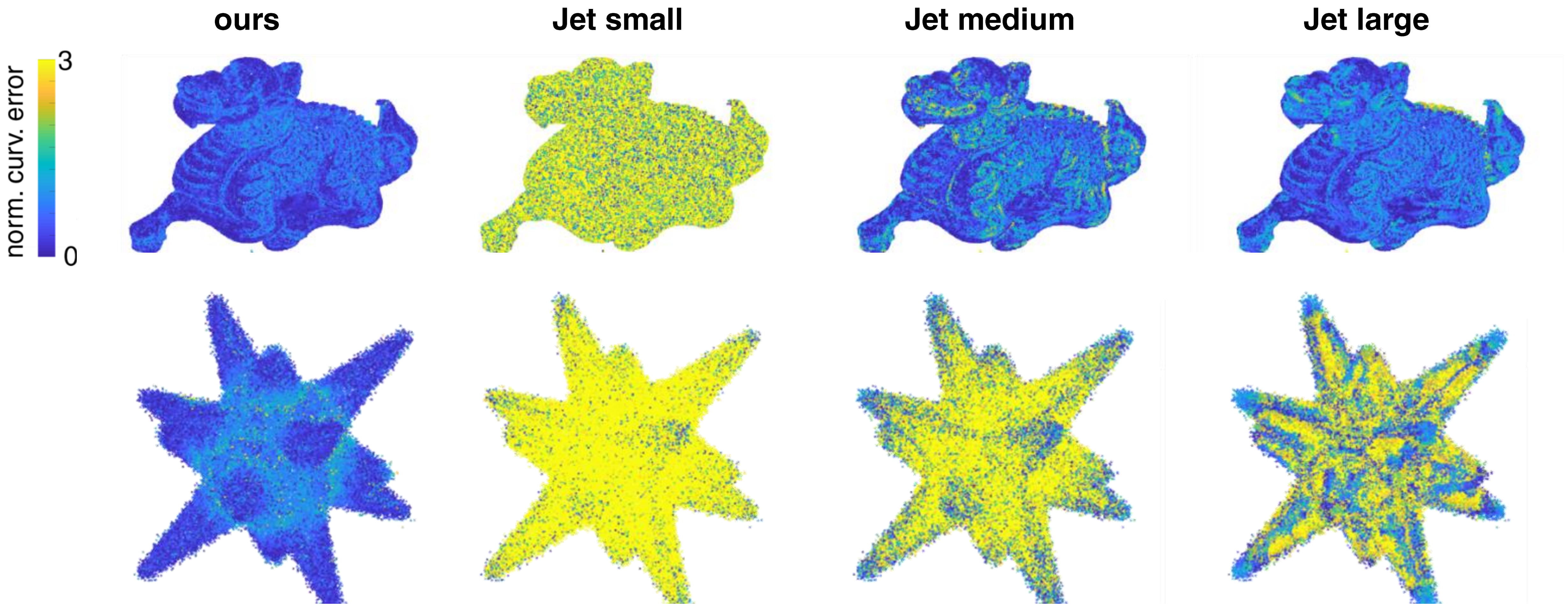




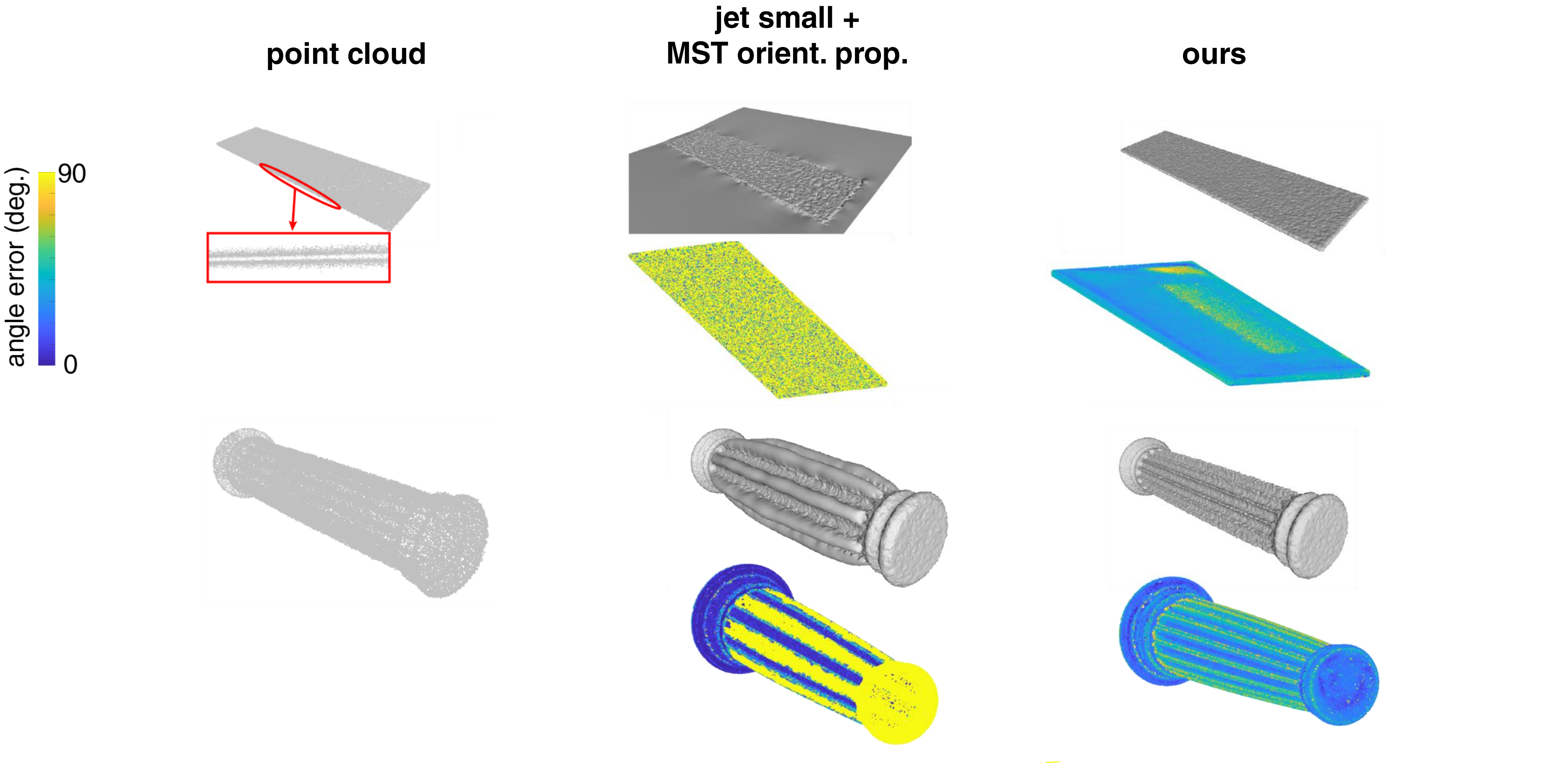
[1] Deep Learning for Robust Normal Estimation in Unstructured Point Clouds, Boulch and Marlet, 2016

Maximum Curvature Magnitude Estimation

max. curvature error

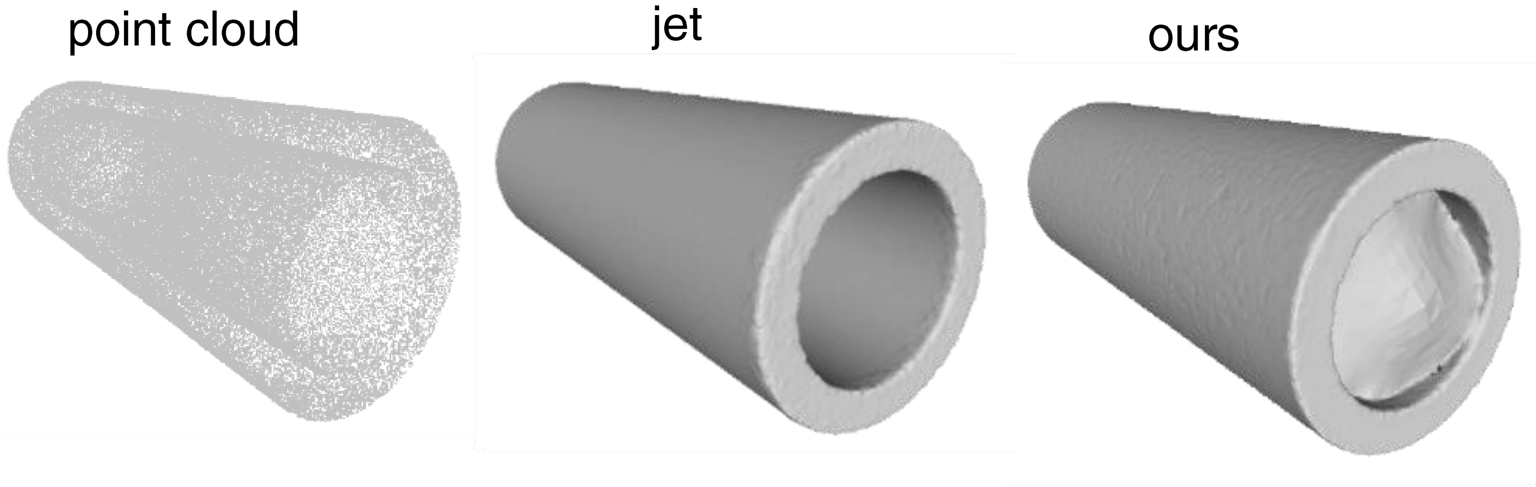


Oriented Normal Estimation & Surface Reconstr.

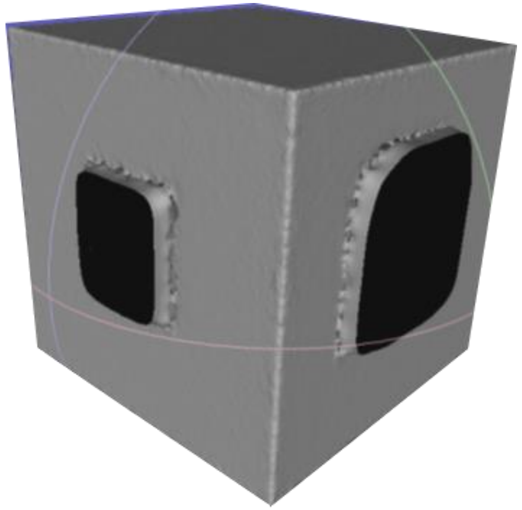


Limitations

- Usually generalizes well, but may fail on patch configurations that are very different from those given in the training set



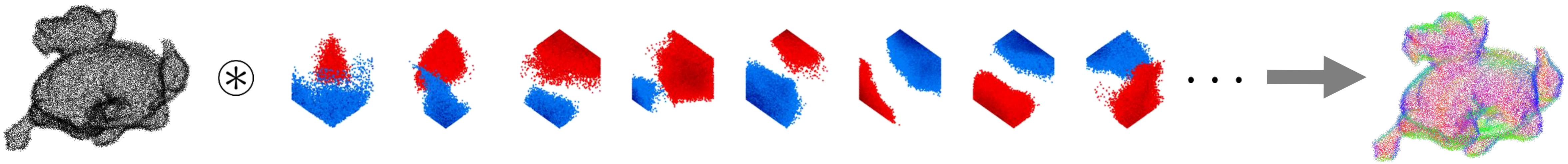
- Ambiguous orientations for flat surfaces larger than patch radius



- Slower to evaluate than PointNet, speed is ~ 200 points / second on a single Titan XP

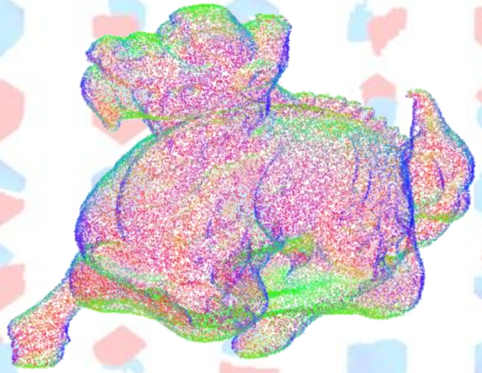
Conclusions

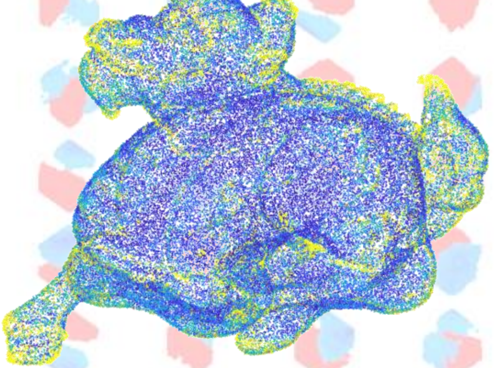
- Point Functions can be seen as learned continuous 3D kernels
- Convolving these kernels with a point cloud gives rich features that can be used for state-of-the-art normal and curvature estimation



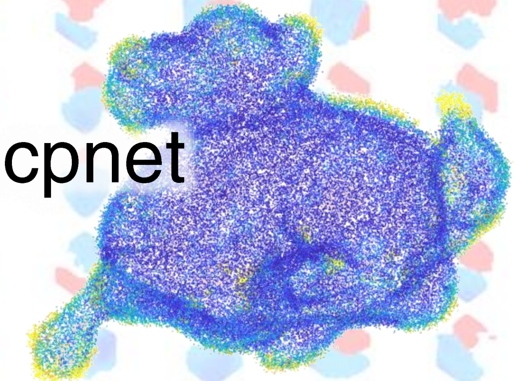
Website:
geometry.cs.ucl.ac.uk/projects/2018/pcpnet

Code:
github.com/paulguerrero/pcpnet





Thanks!

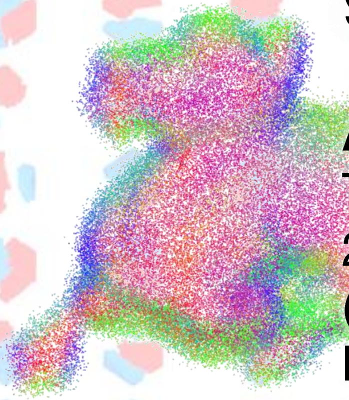


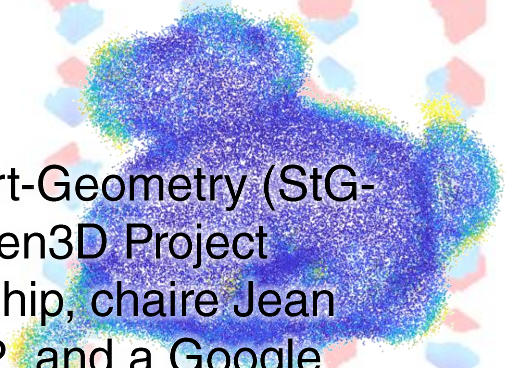
Website:

geometry.cs.ucl.ac.uk/projects/2018/pcpnet

Code:

github.com/paulguerrero/pcpnet





Acknowledgements:
This work was supported by the ERC Starting Grants Smart-Geometry (StG-2013-335373) and EXPROTEA (StG-2017-758800), the Open3D Project (EPSRC Grant EP/M013685/1), the Chateaubriand Fellowship, chaire Jean Marjoulet from Ecole Polytechnique, FUI project TANDEM 2, and a Google Focused Research Award.