Lecture 6:

PointNet

Simone Melzi, Marco tarini

Milano, 22/09/2021

PhD School – Learning on 3D geometries

LA STATALE Università degli Studi di Milano
SAPIENZA Università di Roma

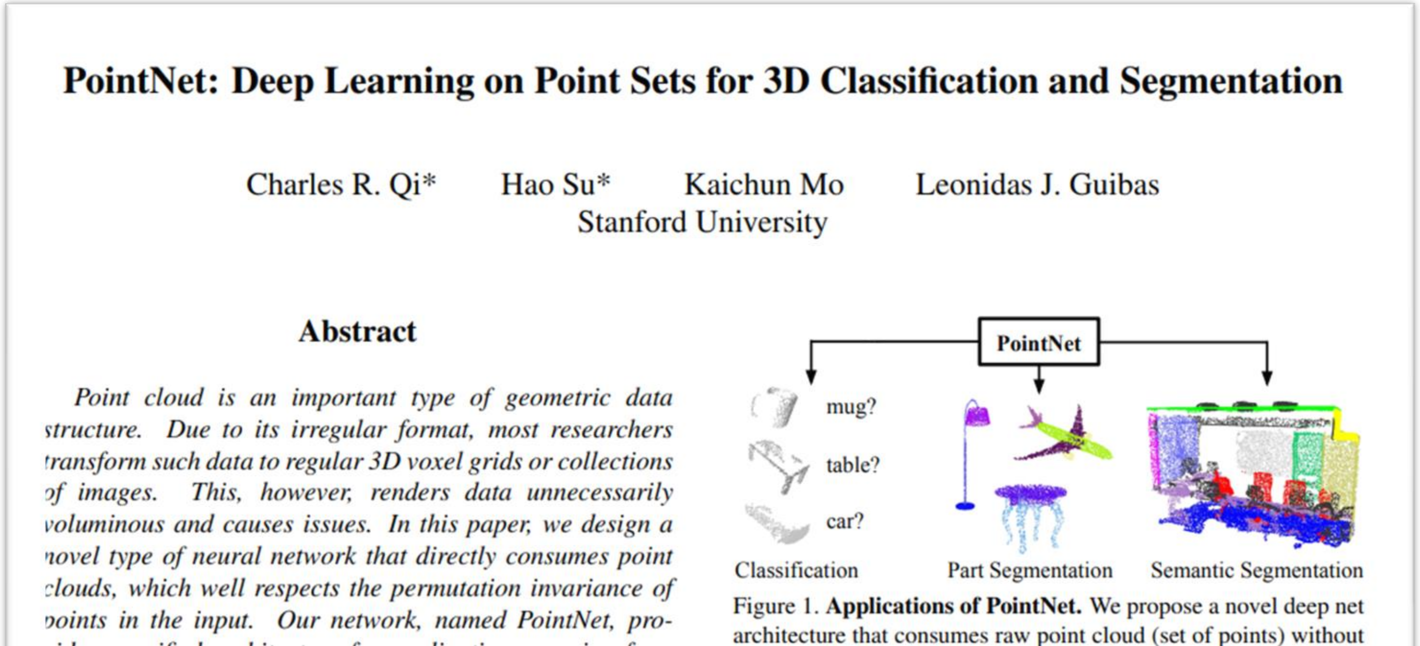
Point clouds:

This is for meshes, what could be done on pointclouds?

Spatial convolution w.r.t. Euclidean distances

Spectral convolution w.r.t. LBO for graph or point clouds

Pointnet



PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

Charles R. Qi* Hao Su* Kaichun Mo Leonidas J. Guibas
Stanford University

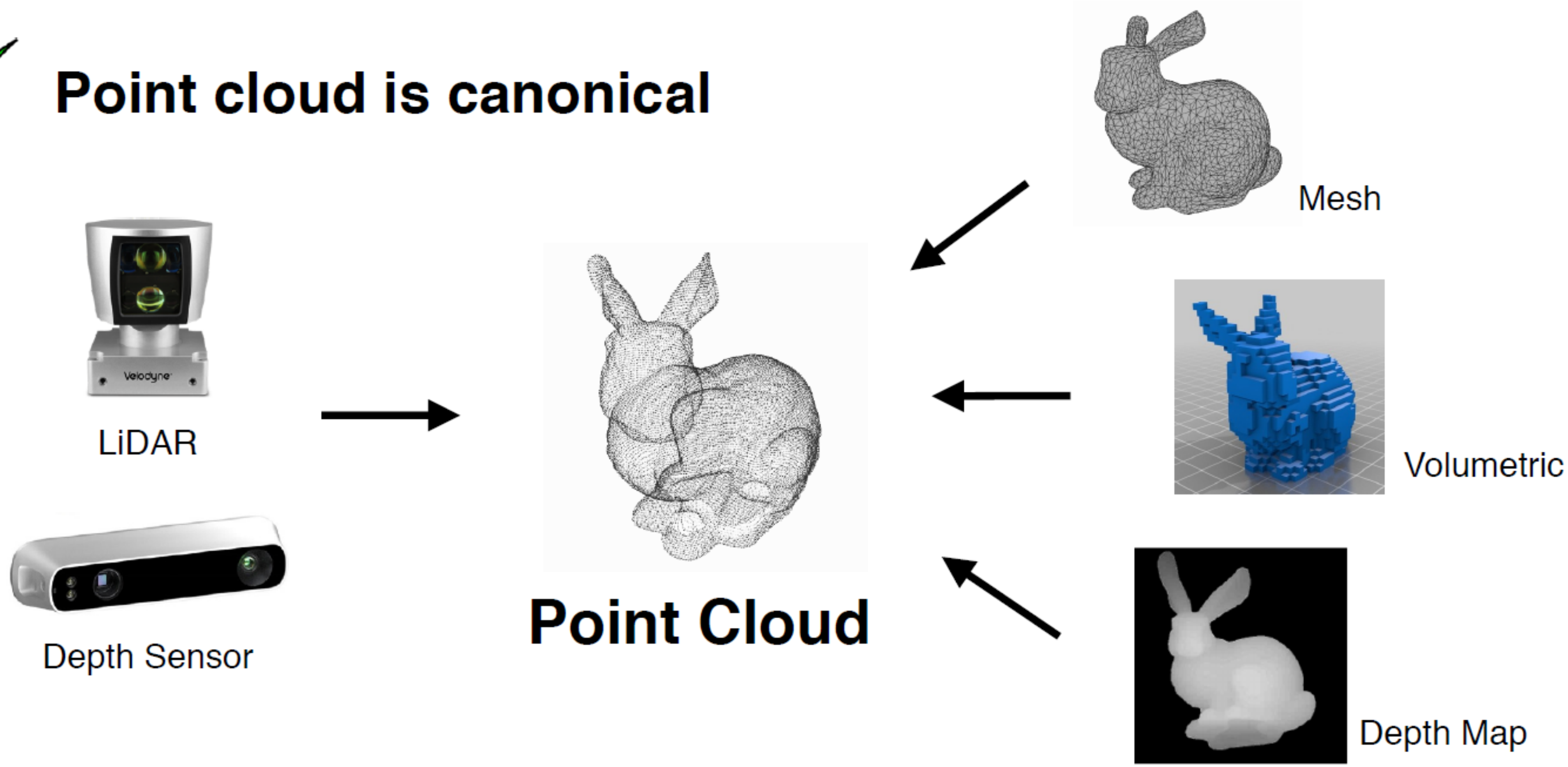
Abstract

Point cloud is an important type of geometric data structure. Due to its irregular format, most researchers transform such data to regular 3D voxel grids or collections of images. This, however, renders data unnecessarily voluminous and causes issues. In this paper, we design a novel type of neural network that directly consumes point clouds, which well respects the permutation invariance of points in the input. Our network, named PointNet, pro-

Figure 1. **Applications of PointNet.** We propose a novel deep net architecture that consumes raw point cloud (set of points) without

PointNET why pointclouds?

- ✓ **Point cloud is close to raw sensor data**
- ✓ **Point cloud is canonical**



[“PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*]

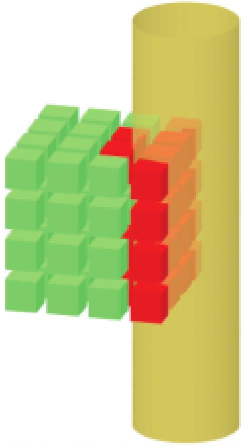
Previous alternatives:

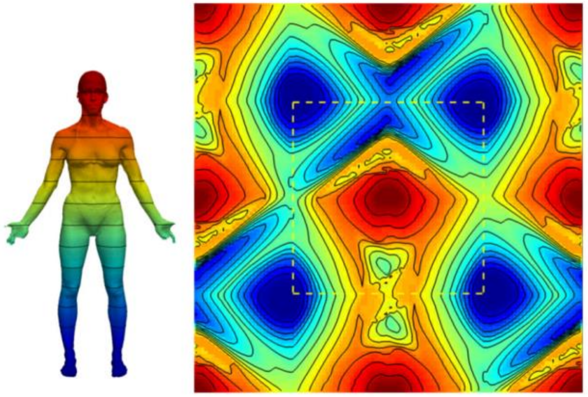
Voxelization → 3D CNN

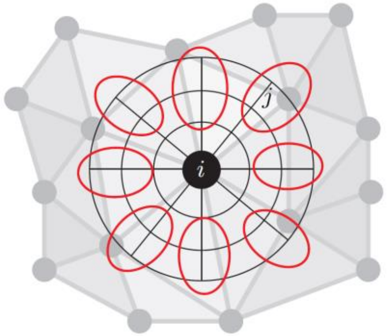
Rendering → 2D CNN

2D projection → 2D CNN

Connectivity → Geometric convolution

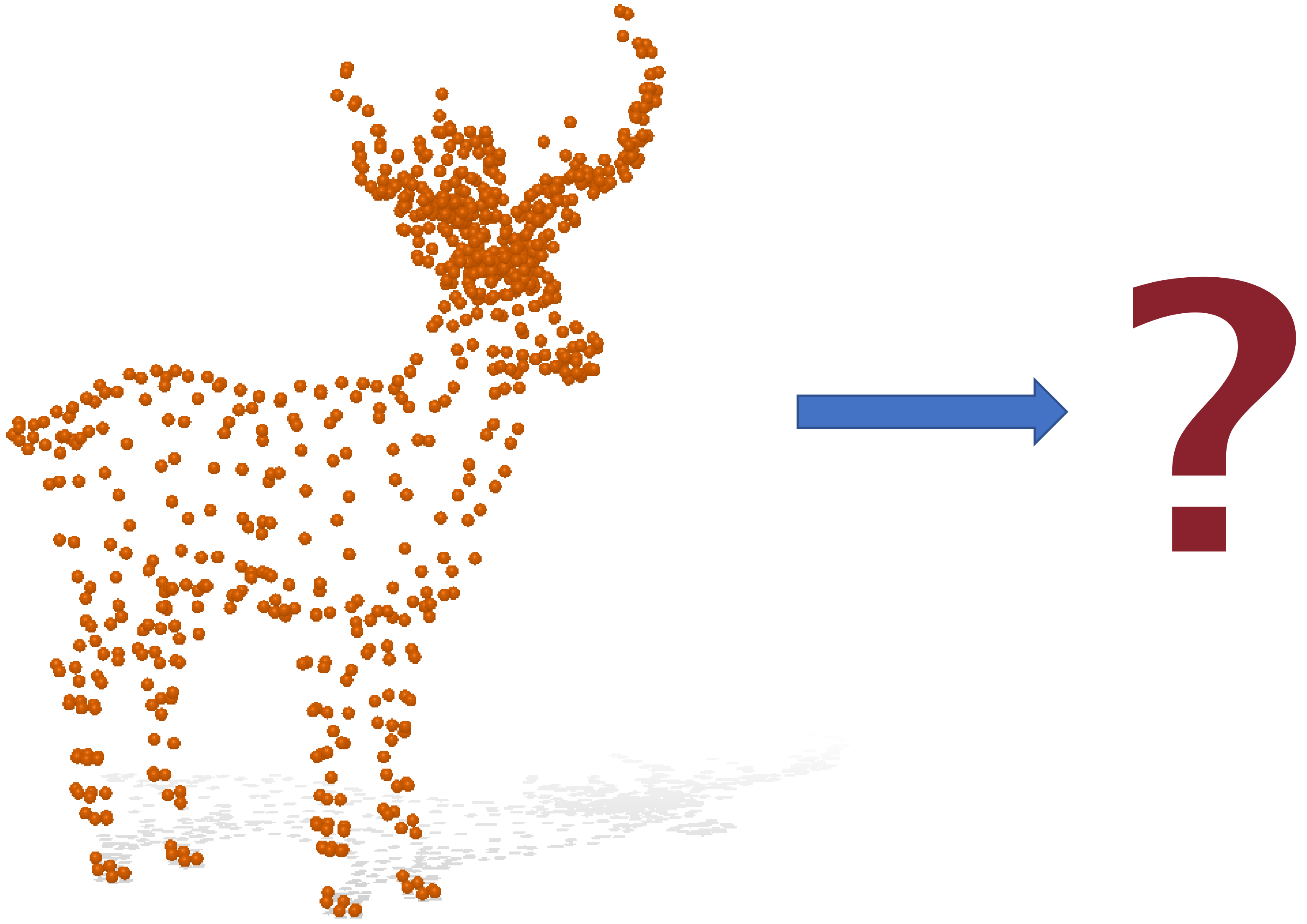






"PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*

Can we directly work on pointclouds?

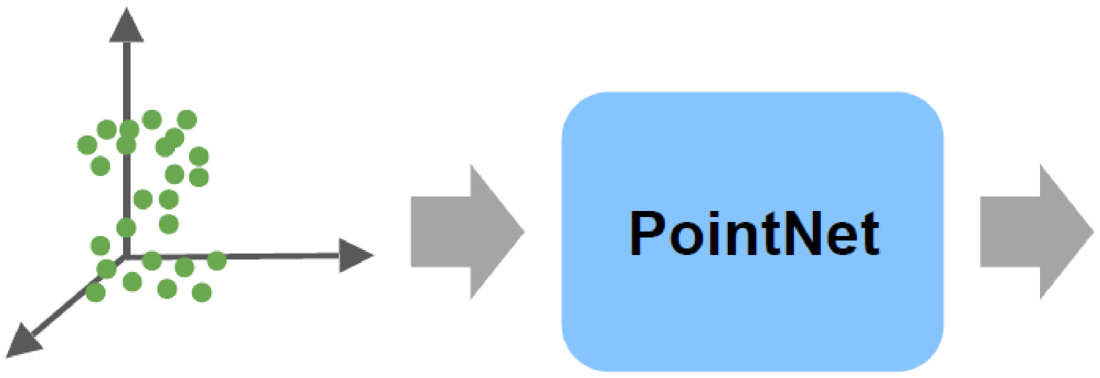


[“PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*]

PointNET pipeline:

End-to-end learning for unorganized data = point clouds

A unique framework for multiple task:



Object Classification

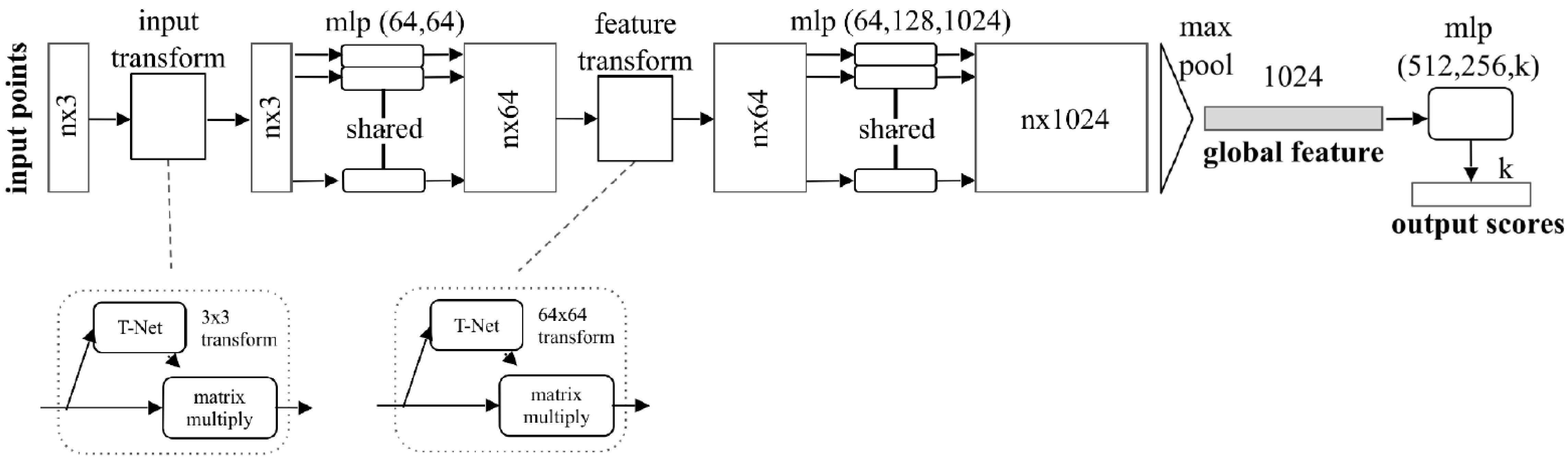
Object Part Segmentation

Semantic Scene Parsing

...

[“PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*]

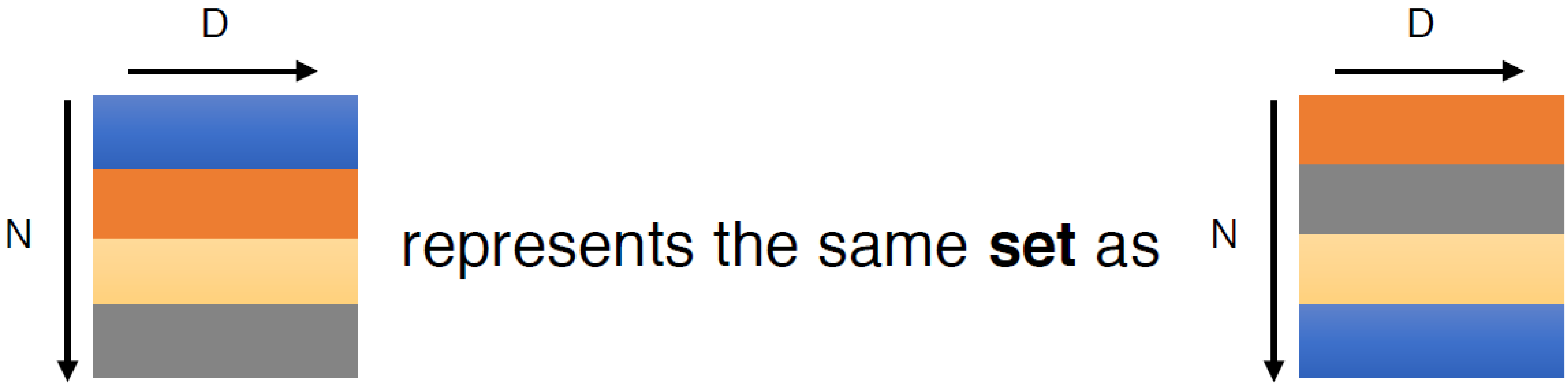
PointNET global feature:



"PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*

PointNET properties:

unorganized data = should be invariant to permutations and to different possible sampling



[“PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*]

Permutation invariance

Examples of symmetric functions: **?**

$f(x_1, x_2, \cdots, x_n) = f\left(x_{\pi_1}, x_{\pi_2}, \cdots, x_{\pi_n}\right), x_i \in \mathbb{R}^D, \pi$ permutazione

$f(x_1, x_2, \cdots, x_n) = \max\{x_1, x_2, \cdots, x_n\}$

$f(x_1, x_2, \cdots, x_n) = \mathrm{mean}\{x_1, x_2, \cdots, x_n\}$

$f(x_1, x_2, \cdots, x_n) = x_1 + x_2 + \cdots + x_n$

…

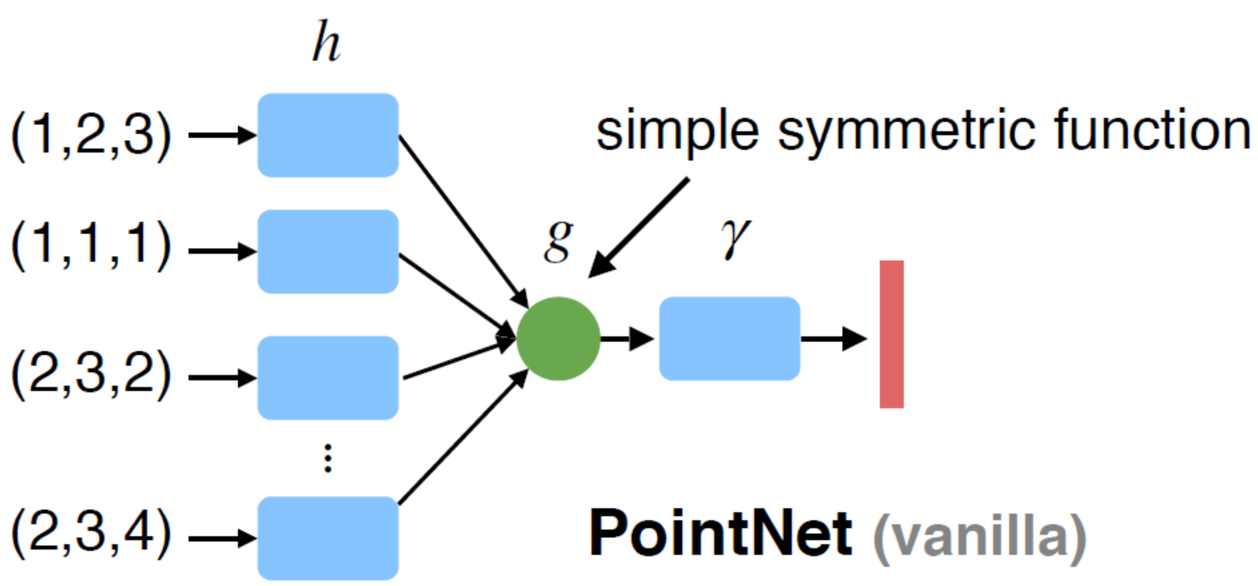
Can we construct a family of symmetric function by Neural Networks?

[“PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*]

Observation

$$f(x_1, x_2, \cdots, x_n) = \gamma \circ g(h(x_1), h(x_2), \cdots, h(x_n)),$$

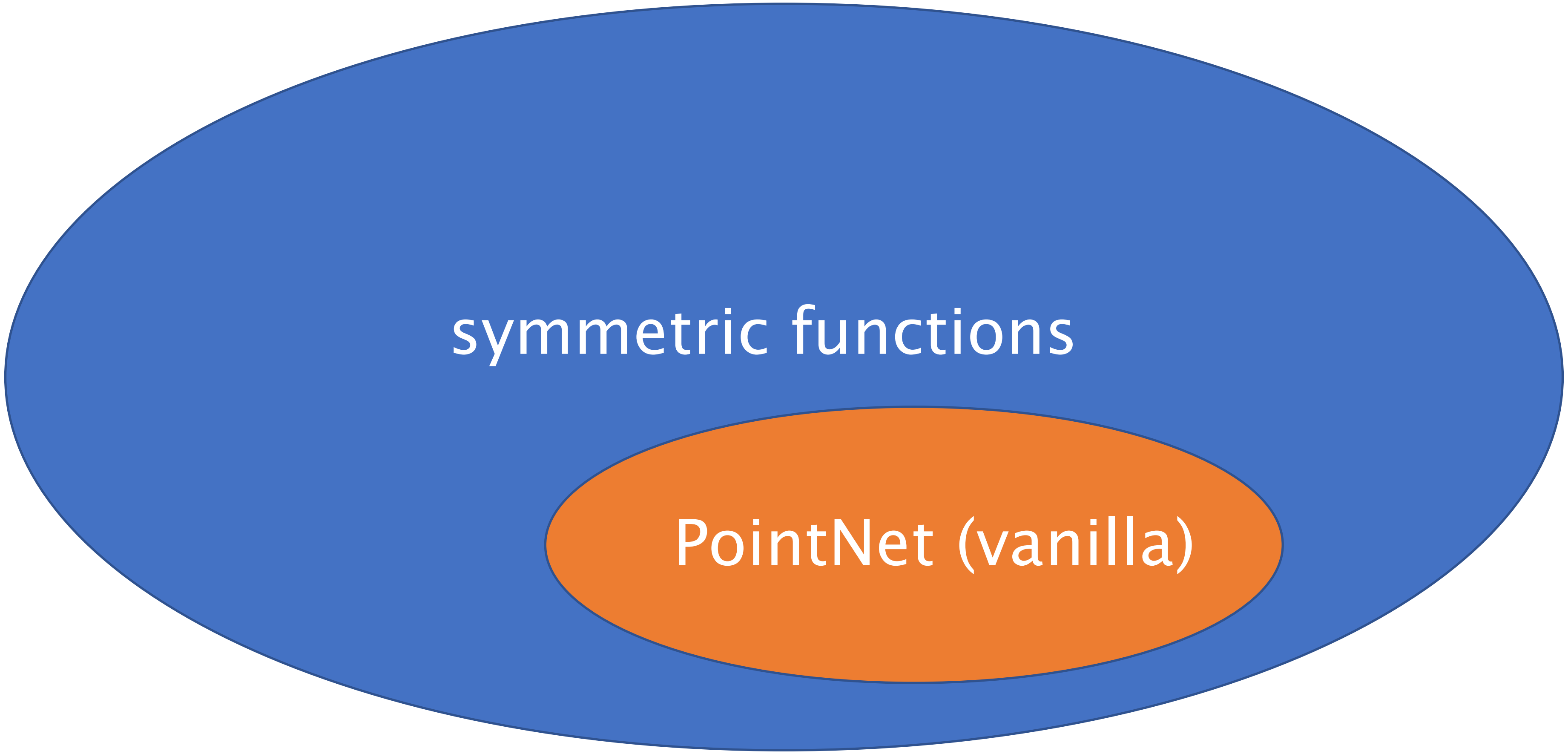
Is symmetric if g is symmetric



$g = max$

"PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*

PointNet (vanilla) functions



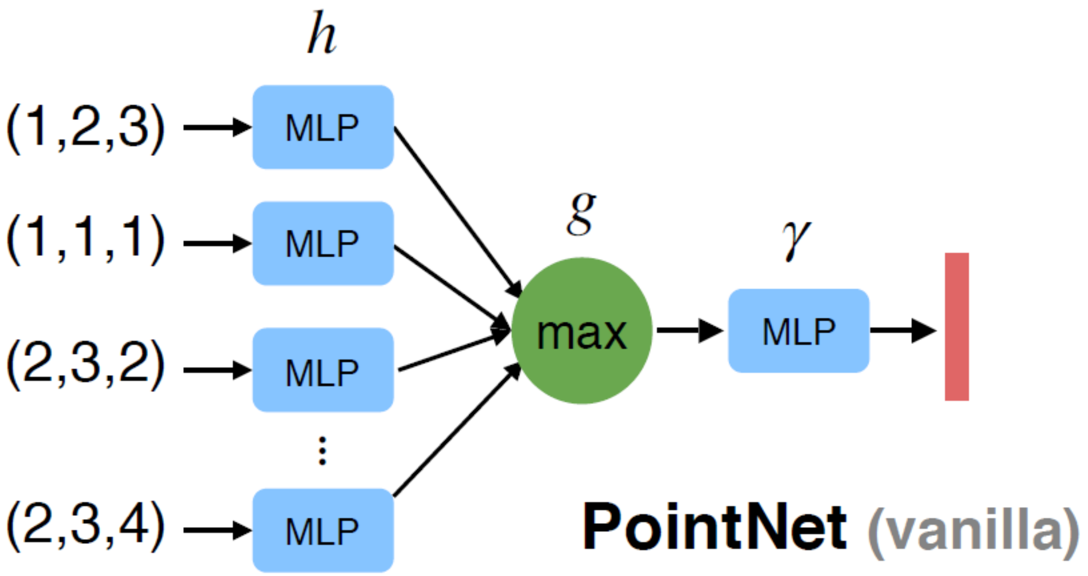
In the paper a theoretical results about which symmetric functions can be approximated by PointNet is reported

[“PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*]

Basic PintNet

g = max pooling

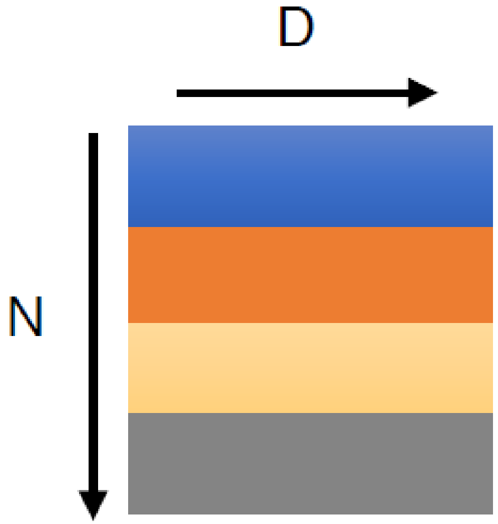
h, γ = Multi-Layer percepton



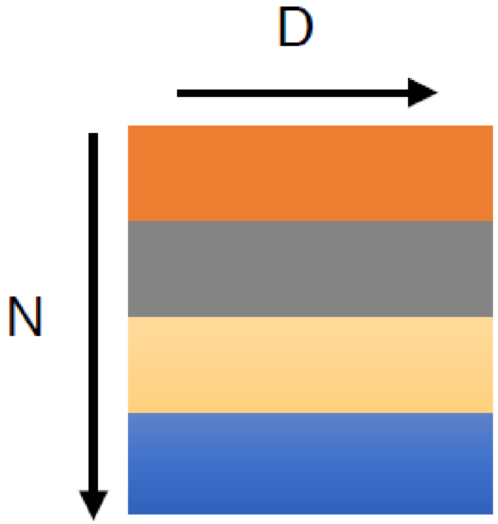
[“PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*]

PointNET pros (1)

With these choices PointNet is permutation invariant!



represents the same **set** as



[“PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*]

PointNET properties:

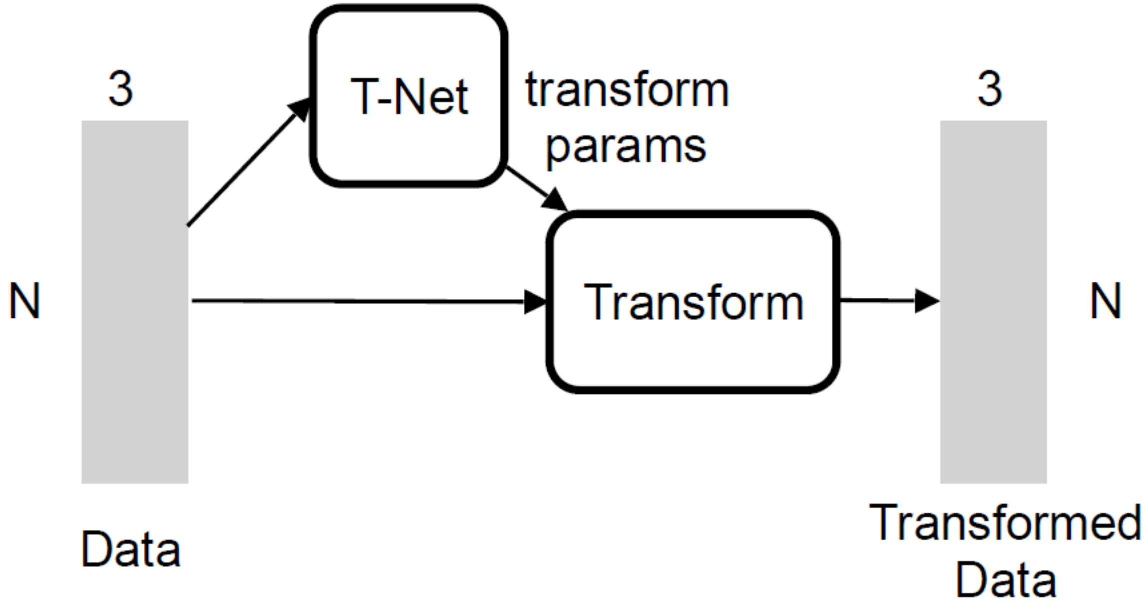
Invariant to geometric transformations …
… at least rigid ones!

Should represent the same set!

[“PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*]

Transformation Network:

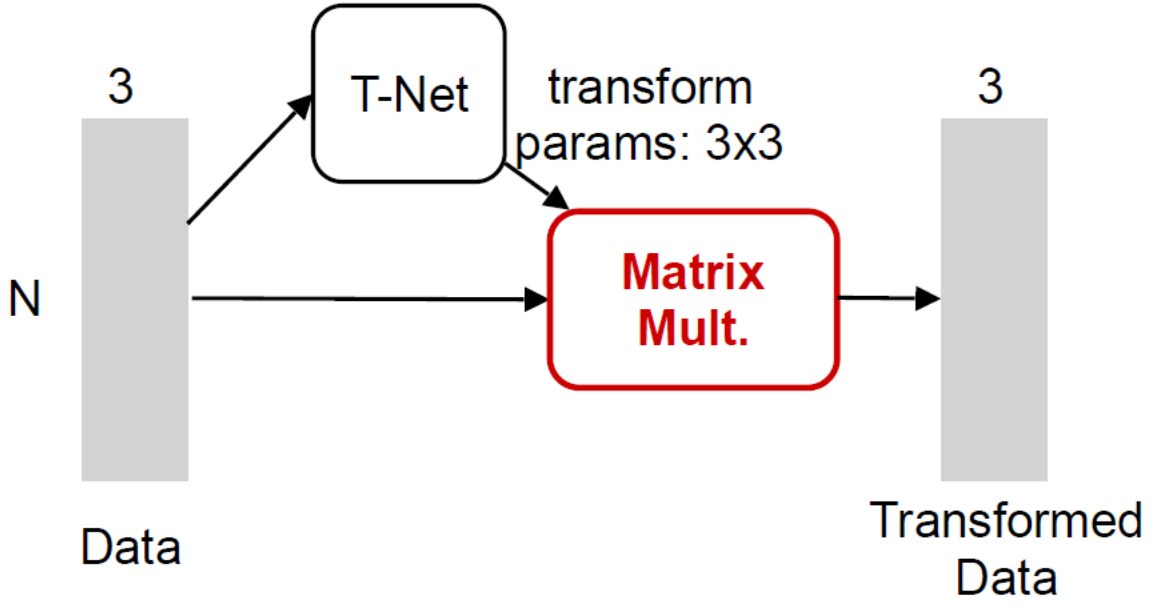
The idea is to learn a data dependent transofrmation for rigid alignment



[“PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*]

Simple Transformation Network:

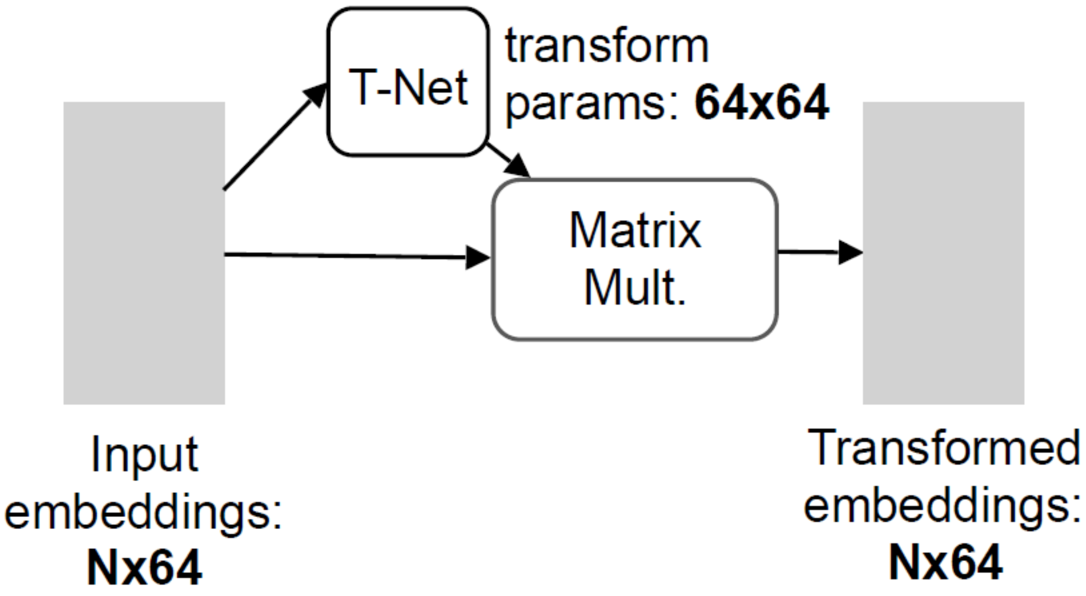
The idea is to learn a data dependent transofrmation for rigid alignment (Very simple = linear)



"PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*

Not only on the input embedding:

The same idea applies to the learned embedding



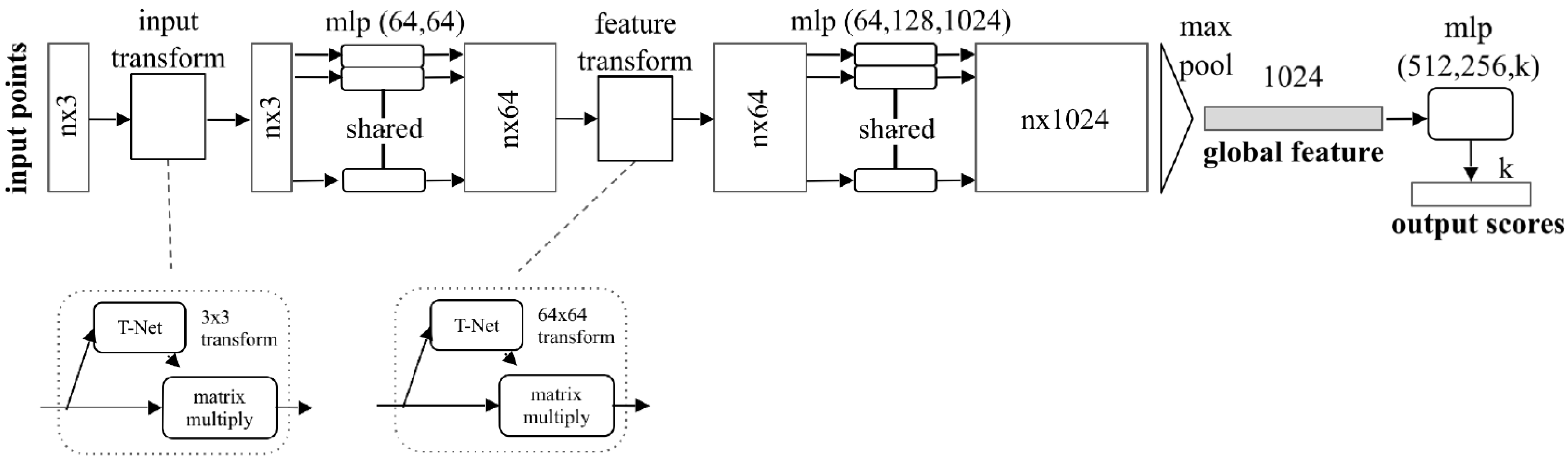
Regularization:

Transform matrix A 64x64 close to orthogonal:

$$L_{reg} = \|I - AA^T\|_F^2$$

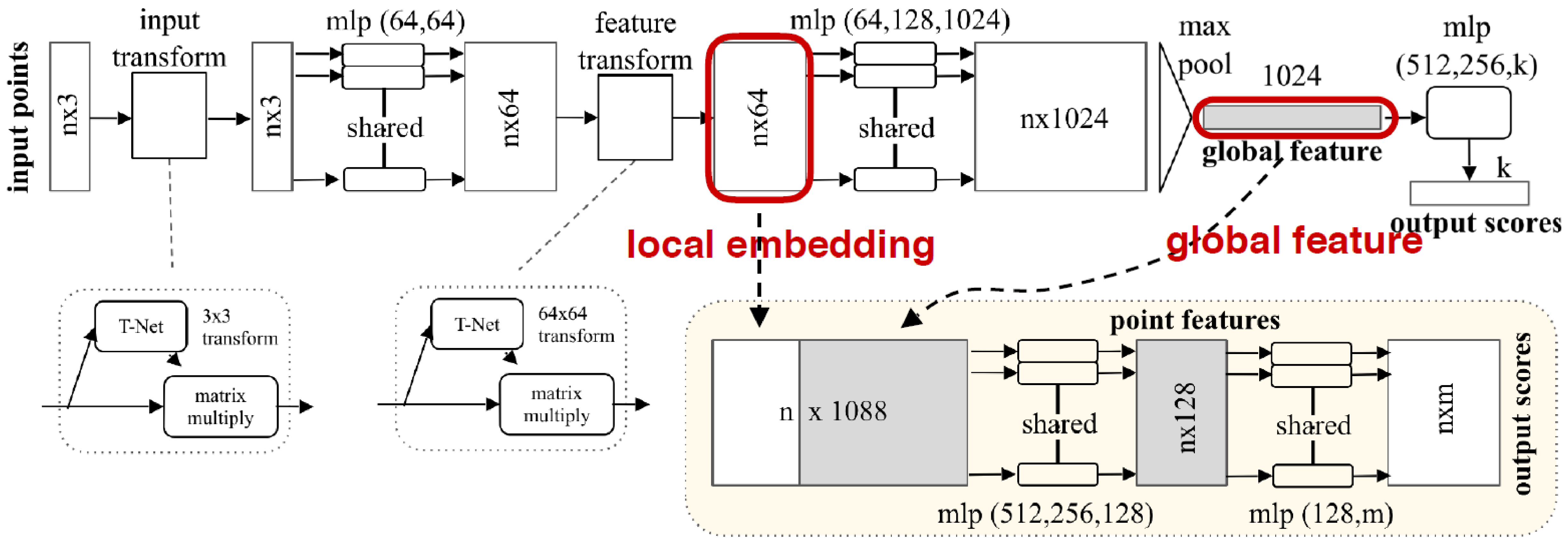
[“PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*]()

PointNET global feature:



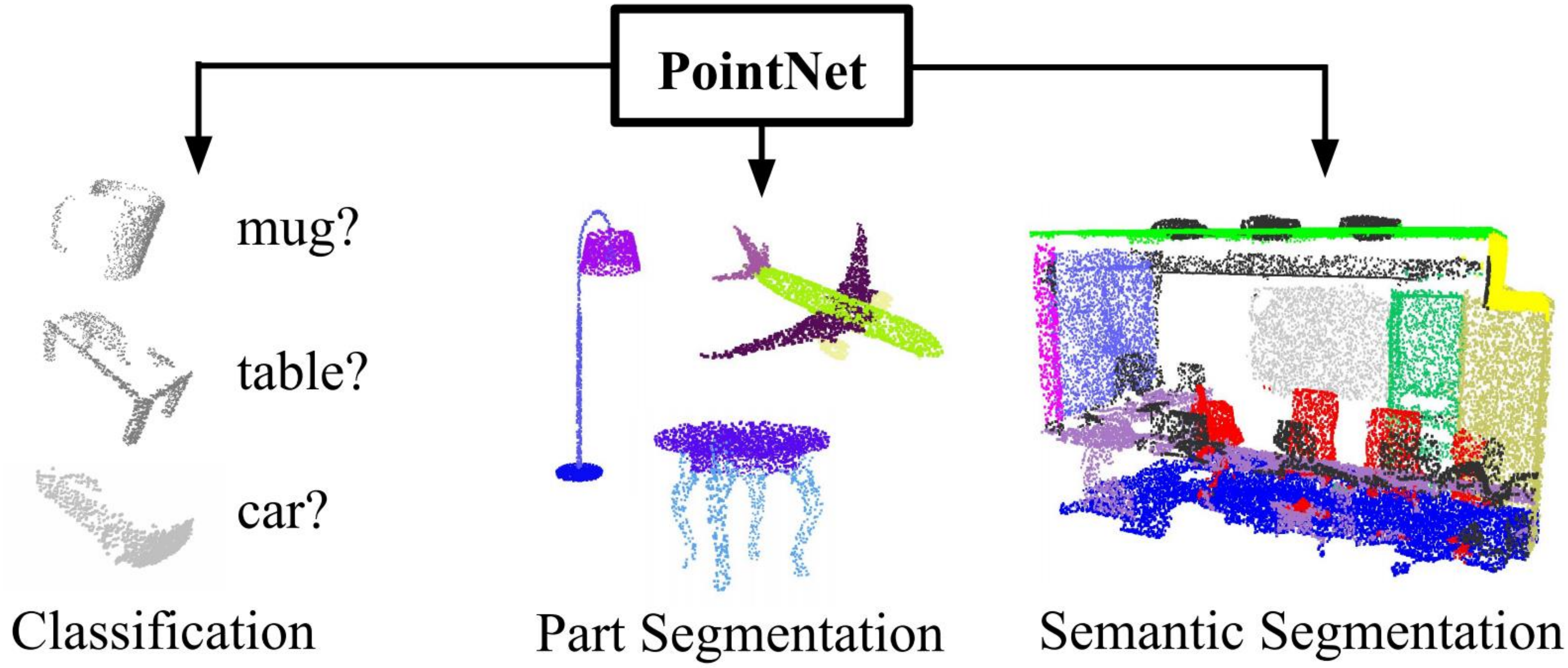
"PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*

PointNET pointwise feature:



[“PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*]

PointNET applications:



"PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*

Quantitative Classification Results:

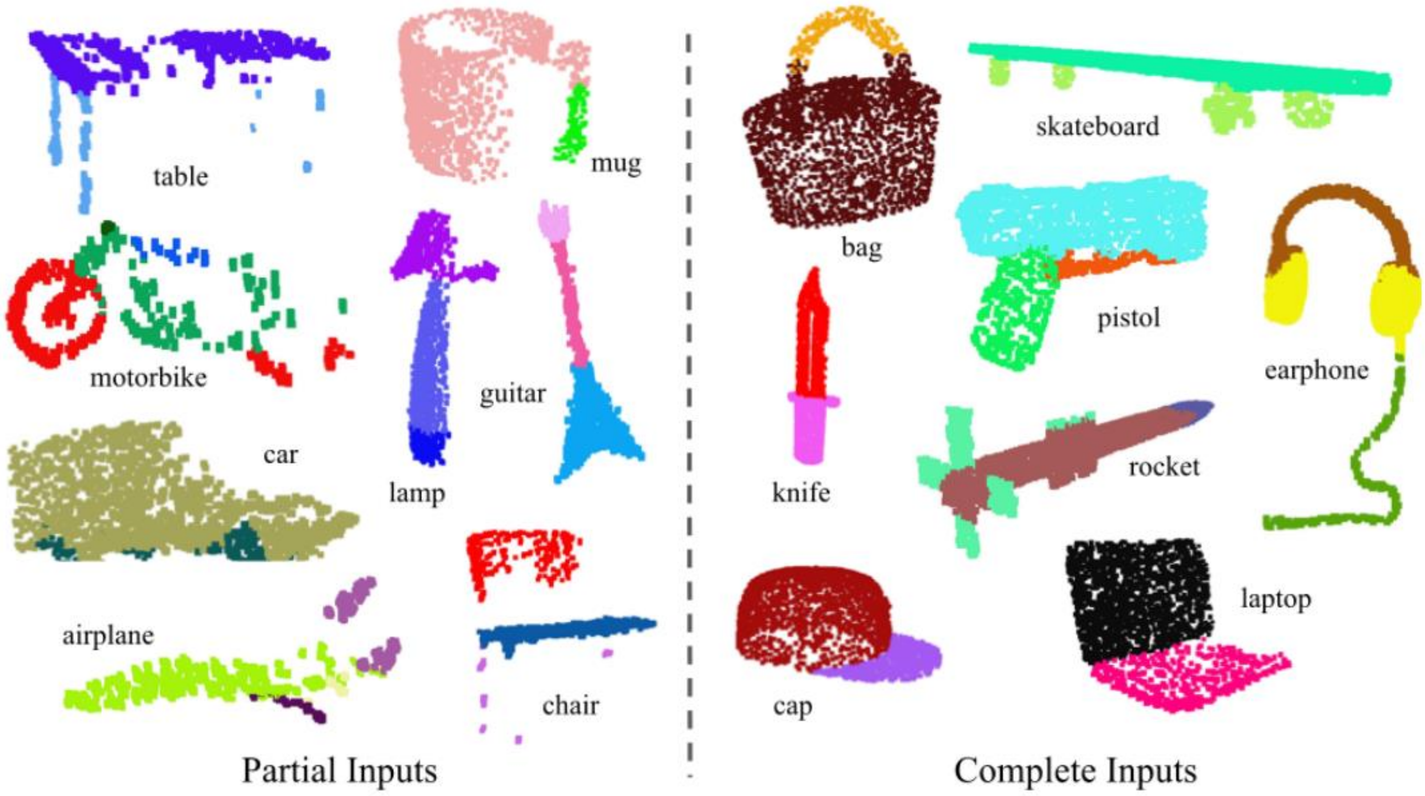
	input	#views	accuracy avg. class	accuracy overall
SPH [12]	mesh	-	68.2	
3DShapeNets [29]	volume	1	77.3	84.7
VoxNet [18]	volume	12	83.0	85.9
Subvolume [19]	volume	20	86.0	**89.2**
LFD [29]	image	10	75.5	-
MVCNN [24]	image	80	**90.1**	-
Ours baseline	point	-	72.6	77.4
Ours PointNet	point	1	86.2	**89.2**

3D CNNs

Classification on ModelNet40 dataset

[“PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*]

Qualitative Classification Results:



[“PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*]

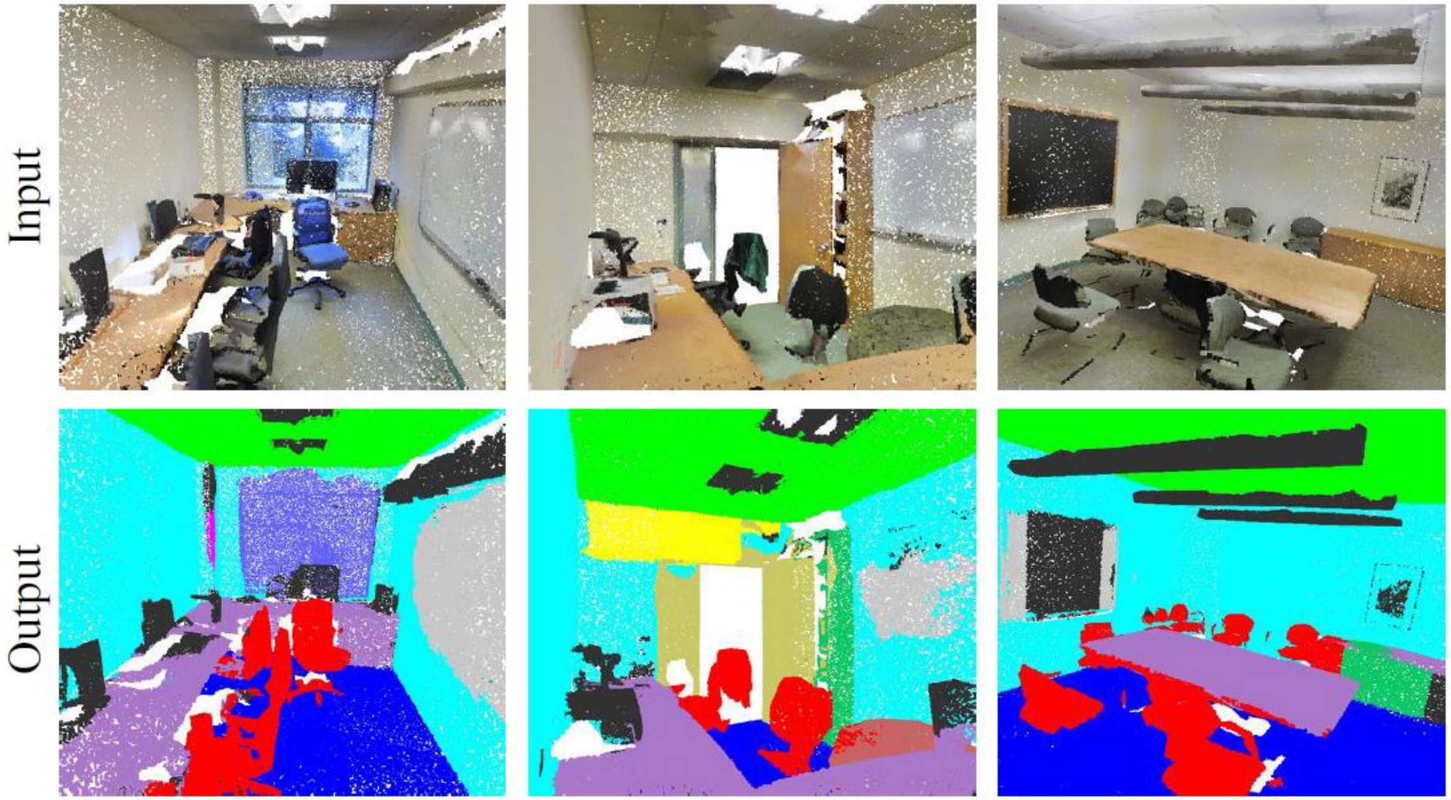
Quantitative Segmentation Results:

	mean	aero	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate board	table
# shapes		2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271
Wu [28]	-	63.2	-	-	-	73.5	-	-	-	74.4	-	-	-	-	-	-	74.8
Yi [30]	81.4	81.0	78.4	77.7	**75.7**	87.6	61.9	**92.0**	85.4	**82.5**	**95.7**	**70.6**	91.9	**85.9**	53.1	69.8	75.3
3DCNN	79.4	75.1	72.8	73.3	70.0	87.2	63.5	88.4	79.6	74.4	93.9	58.7	91.8	76.4	51.2	65.3	77.1
Ours	**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**

Segmentation on ShapeNetPart dataset

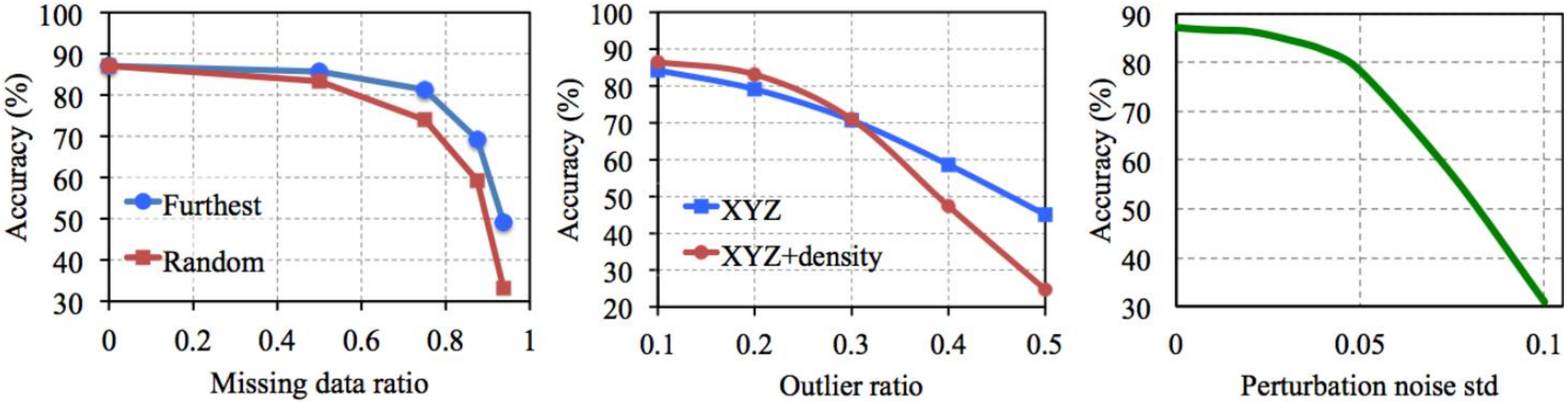
["PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*]

Qualitative Segmentation Results:



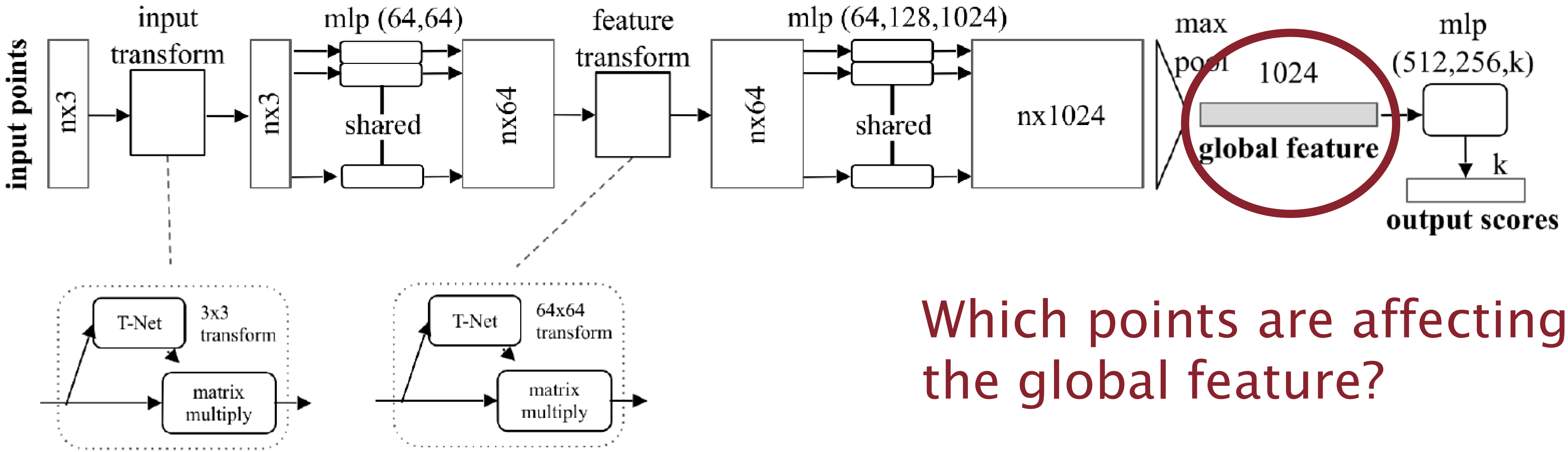
"PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*

Robustness:



[“PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*]

Global features:



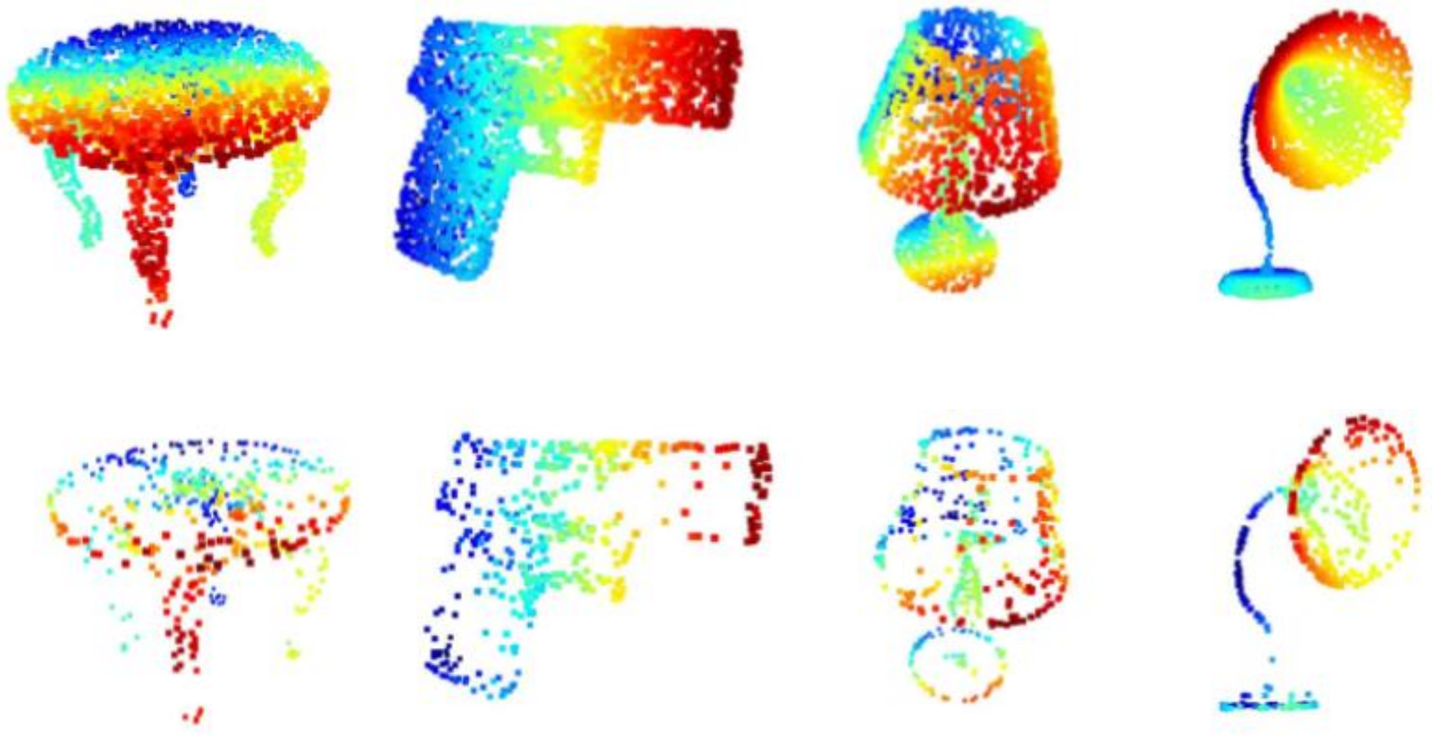
Which points are affecting the global feature?

[“PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*]

Important points for global features:

Original sets of points

Points represented in the global feature

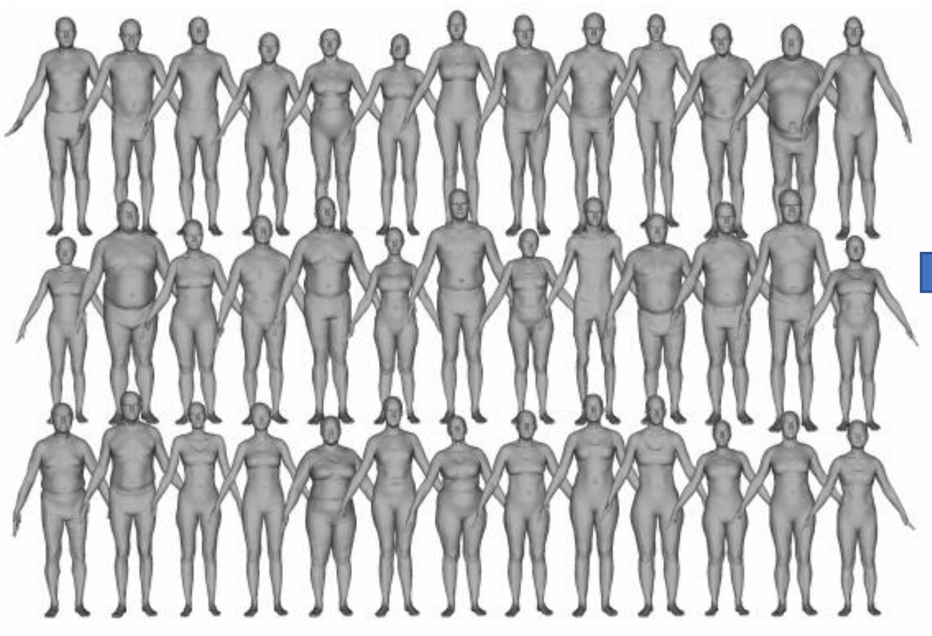


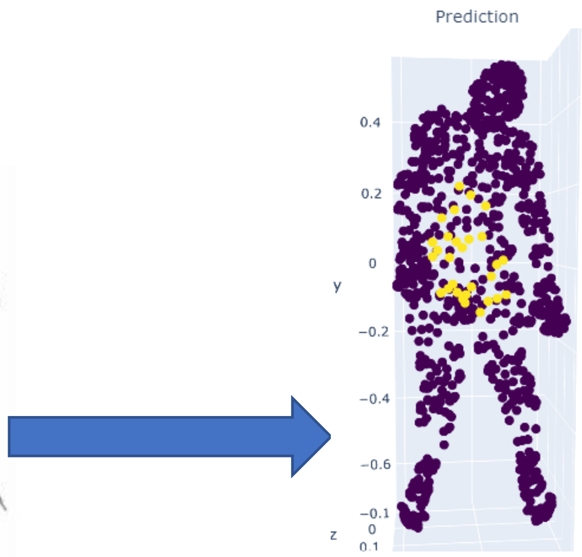
Which ones are these points?

[“PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" *Qi et al., 2016*]

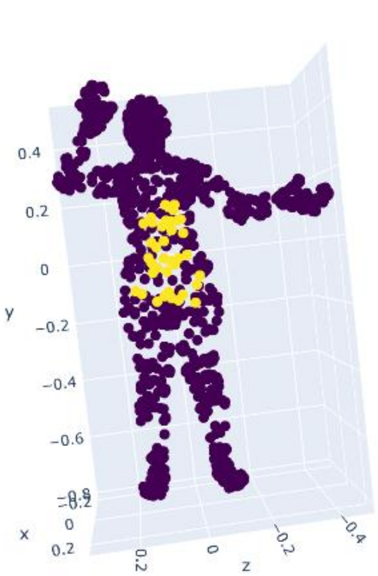
PointNet Demo

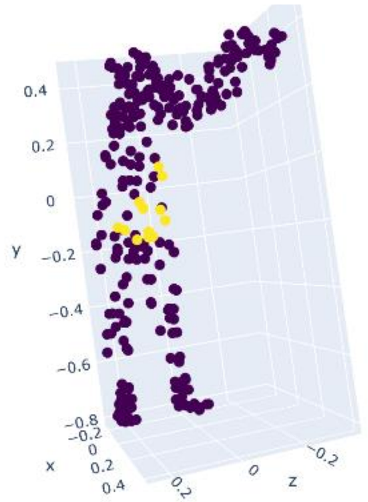
https://github.com/melzismn/Digital-Design-2020-2021/blob/master/PointNet_Demo.ipynb



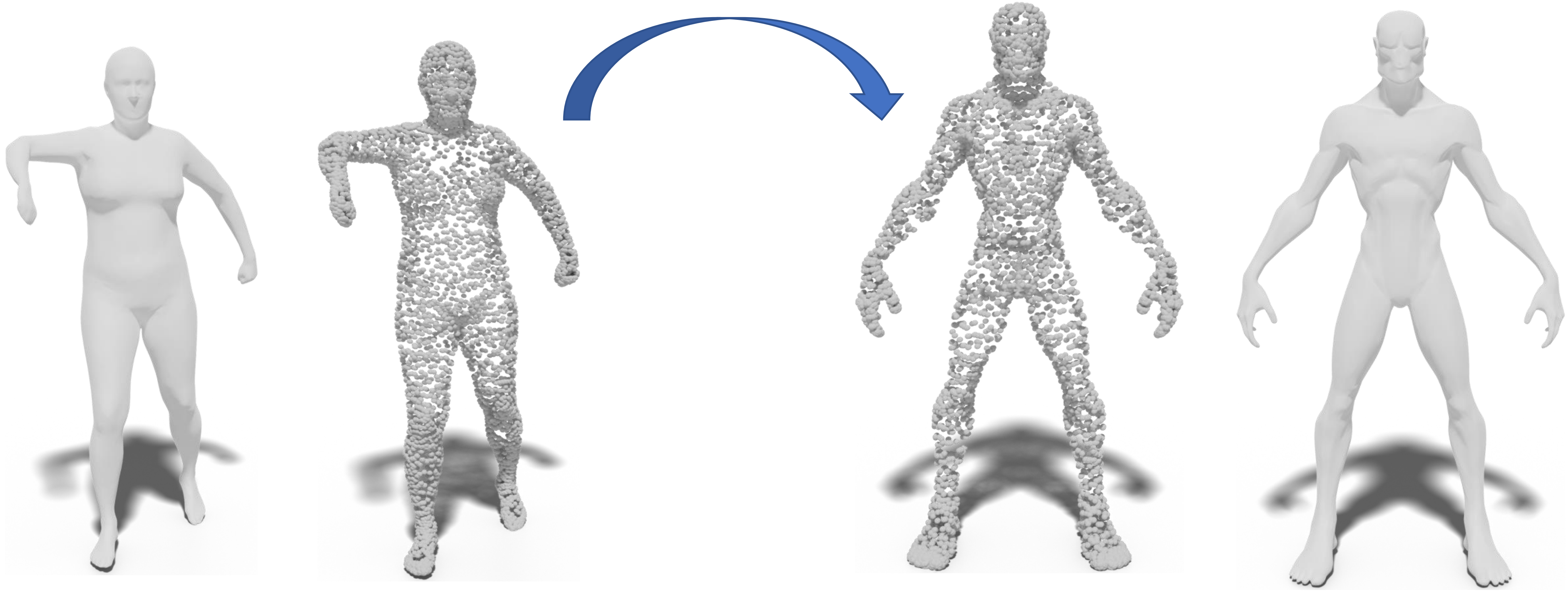


Prediction

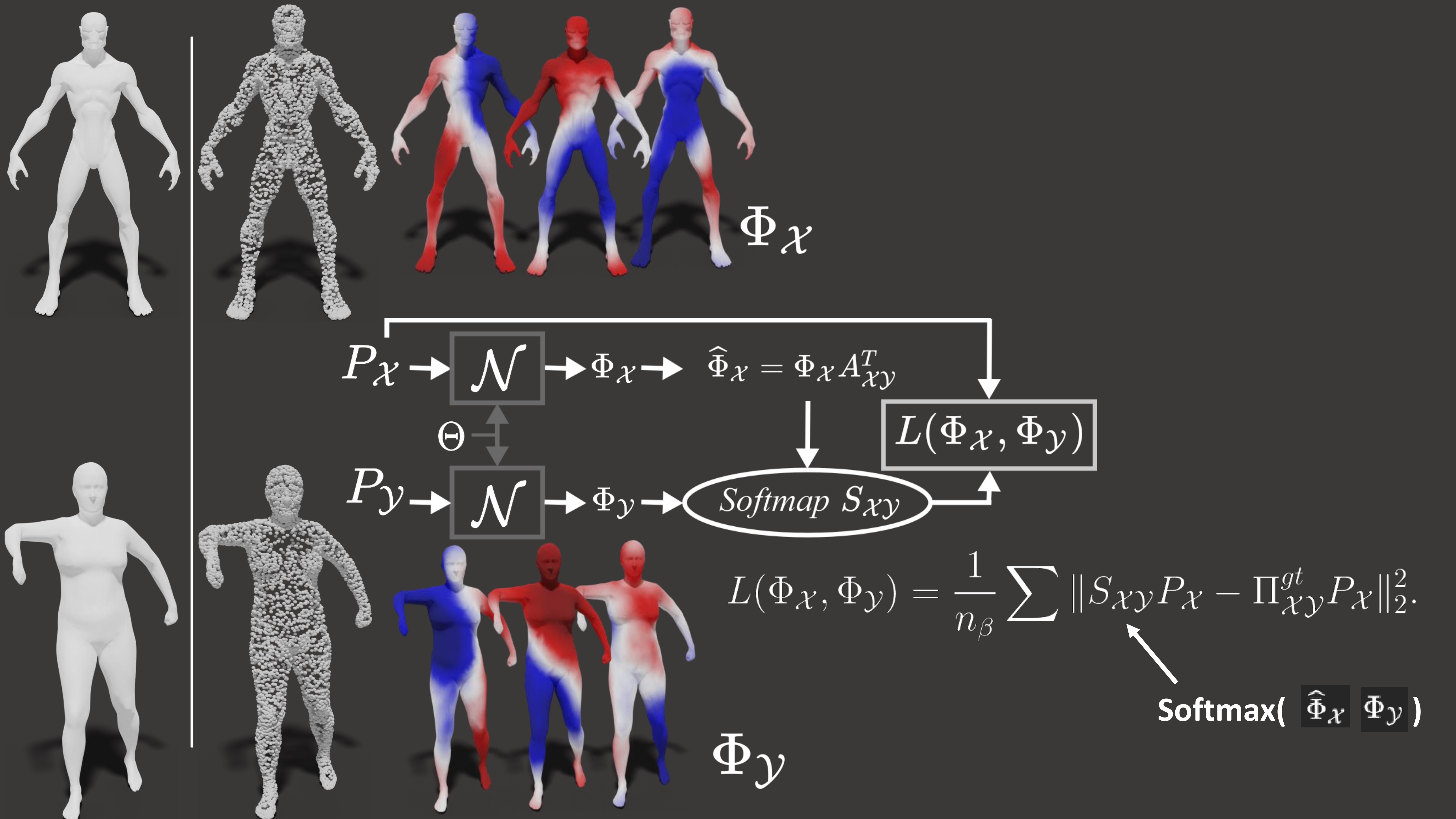




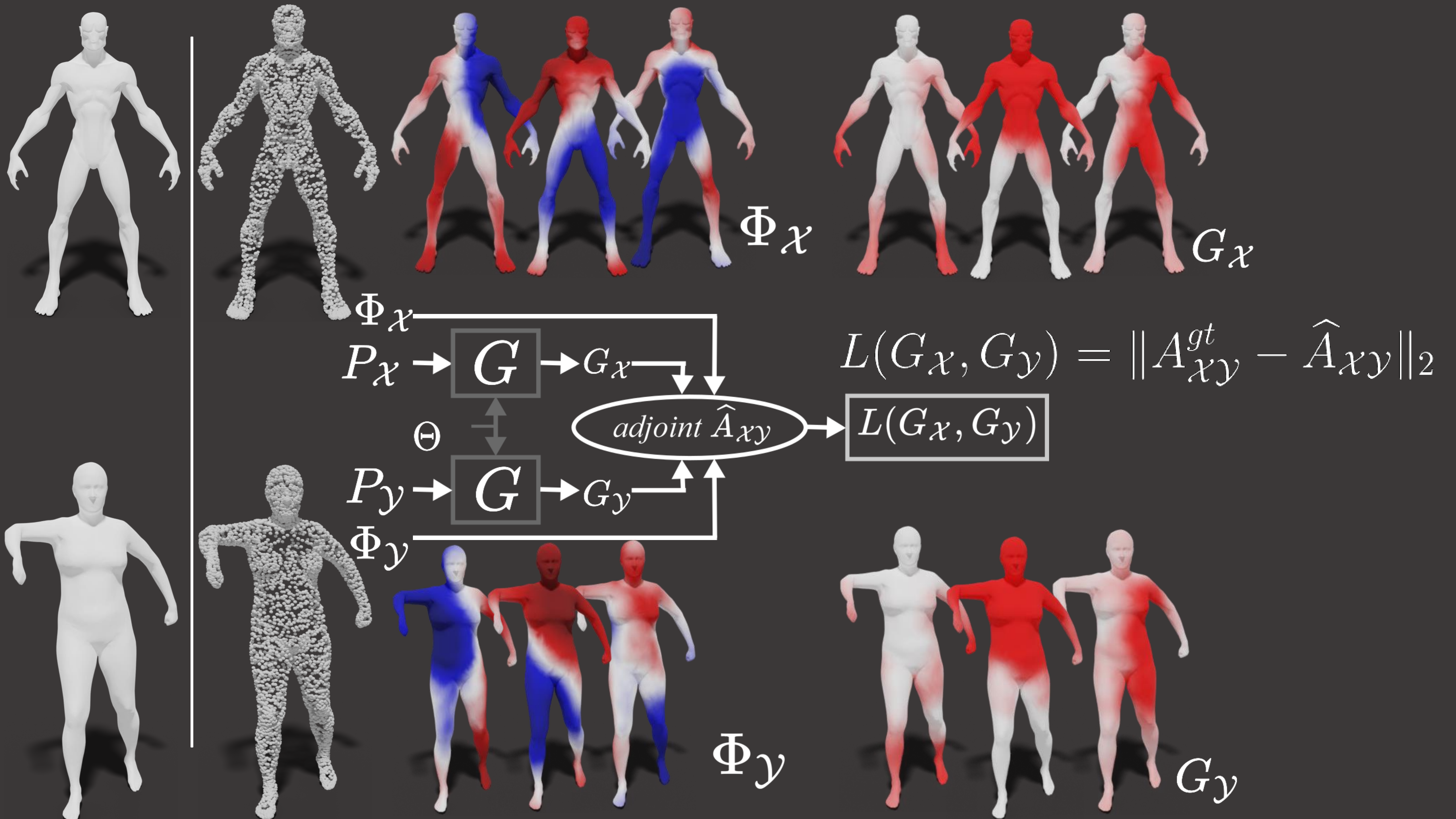
PointNet And FMAP



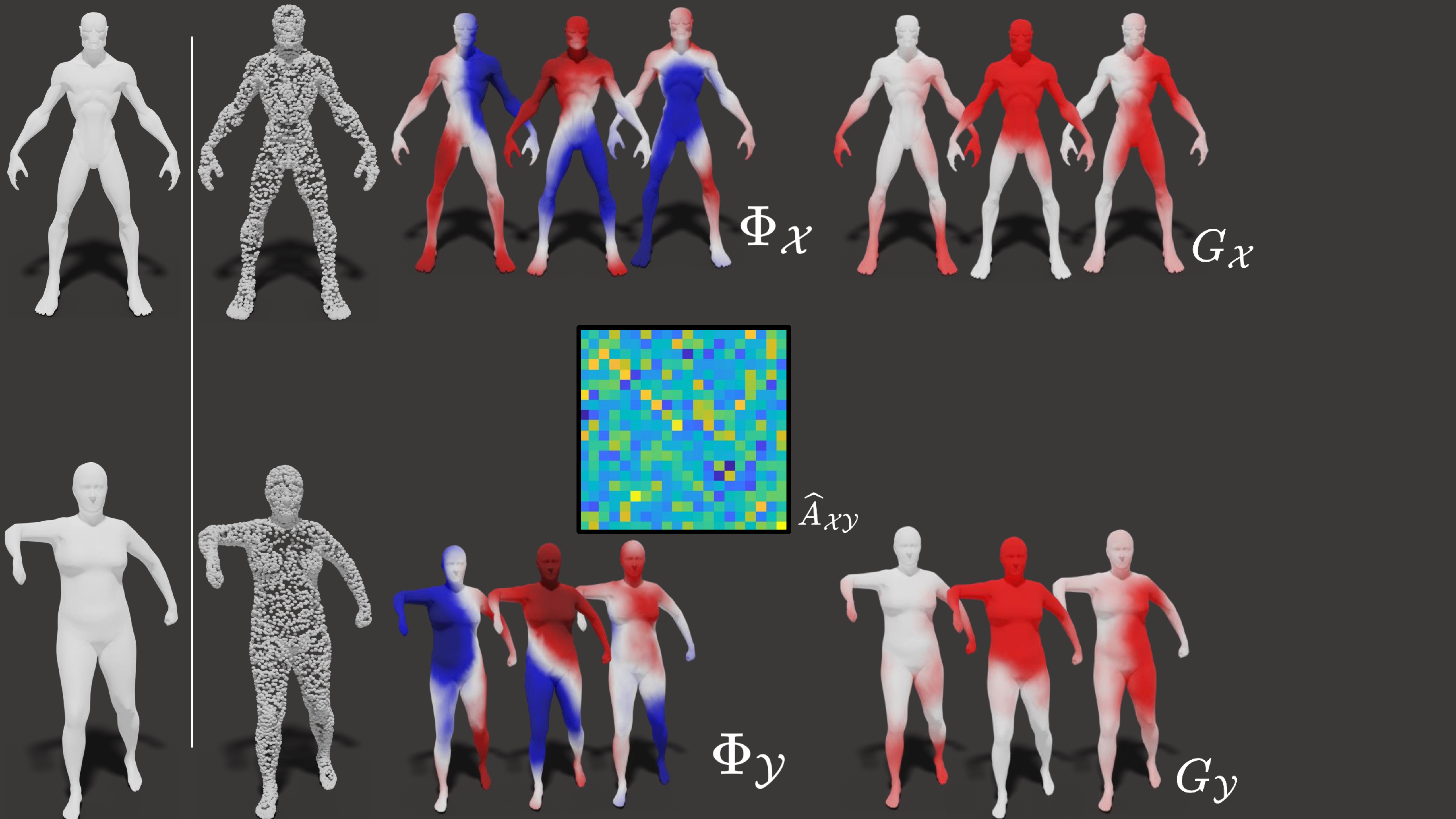
Can we use FMAPS to match these 2 point clouds?

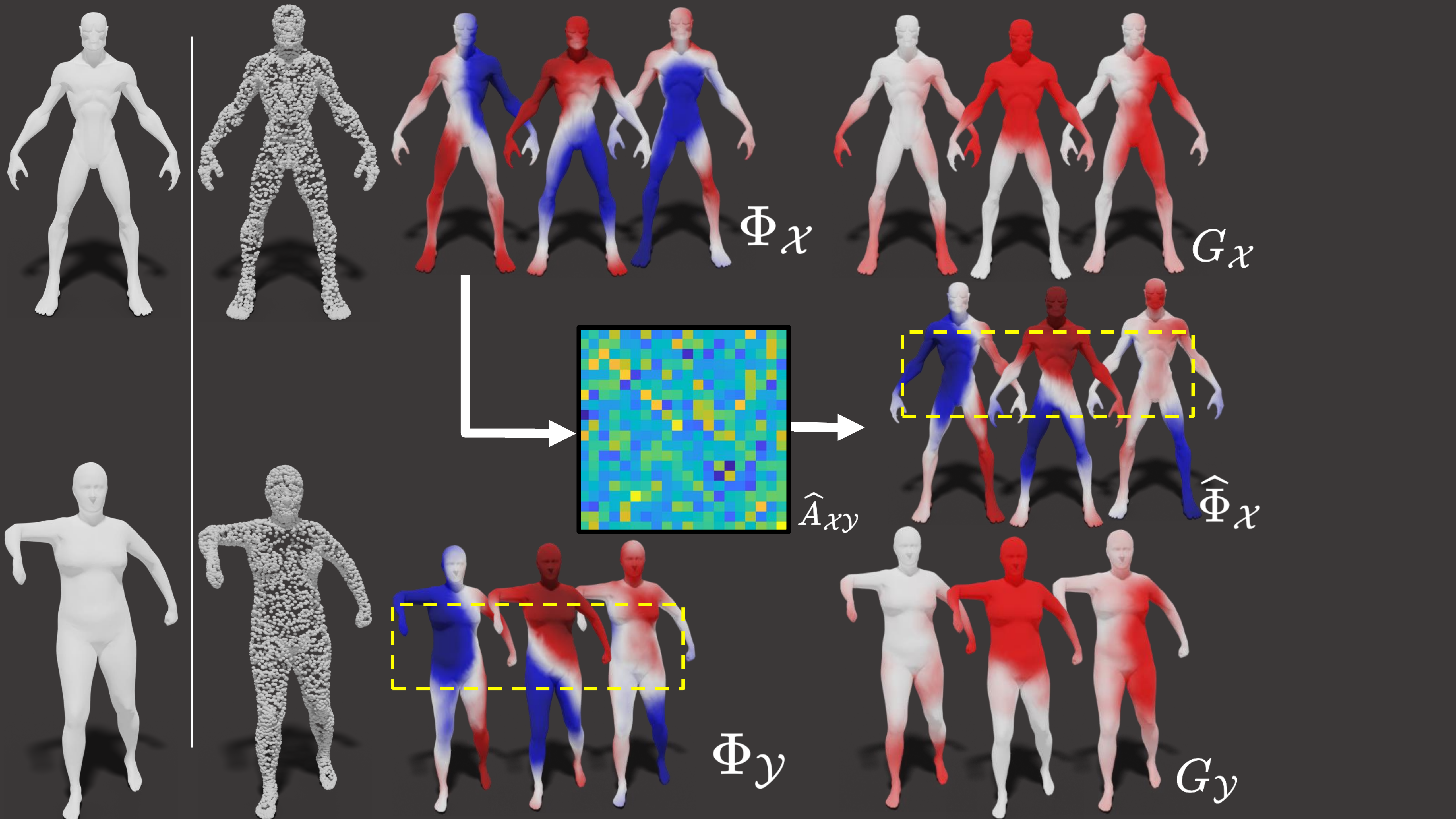


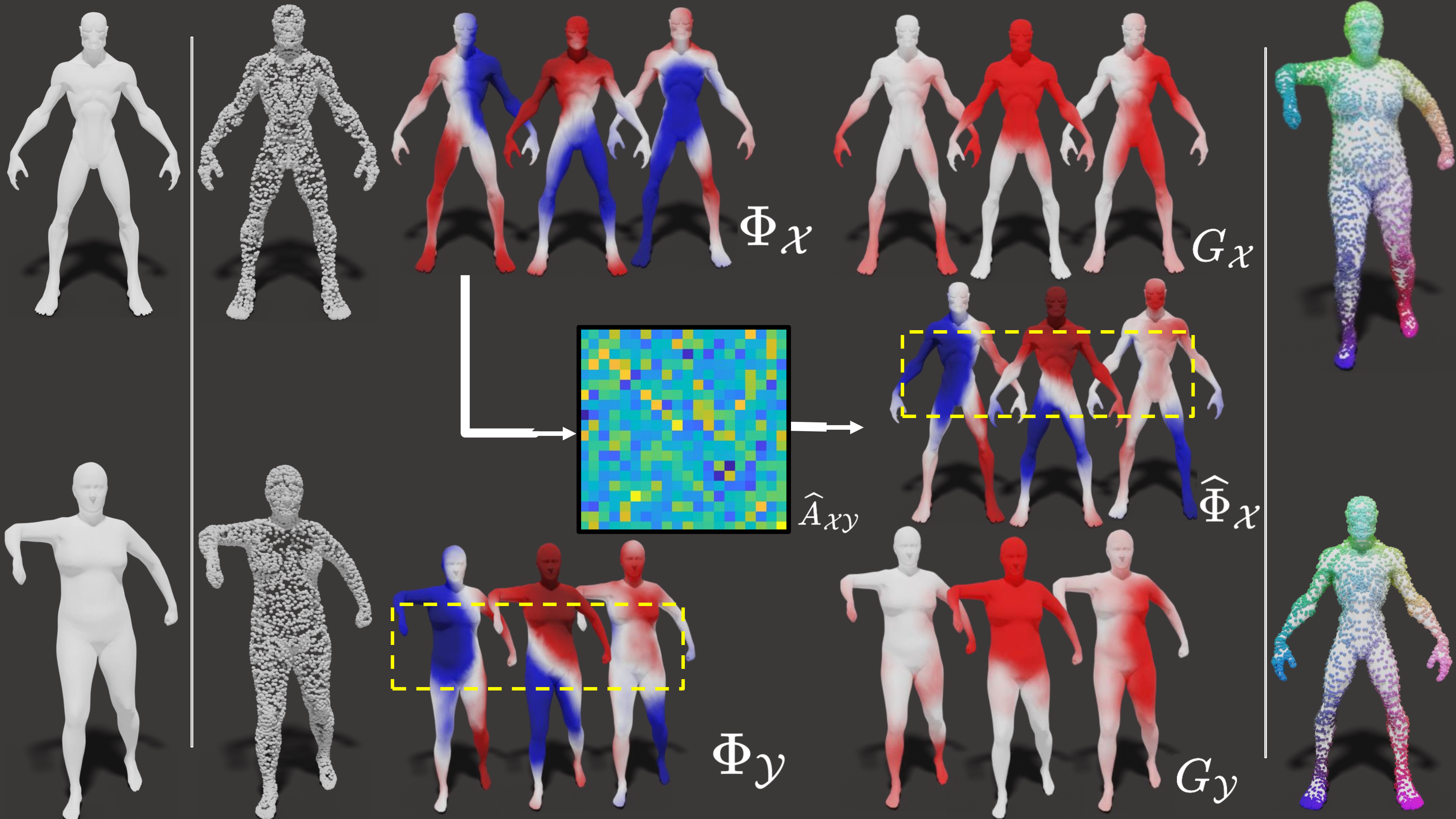
$\Phi_{\mathcal{X}}$
$P_{\mathcal{X}}$
$\mathcal{N}$
$\Phi_{\mathcal{X}}$
$\widehat{\Phi}_{\mathcal{X}} = \Phi_{\mathcal{X}} A^T_{\mathcal{X}\mathcal{Y}}$
Θ
$L(\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}})$
$P_{\mathcal{Y}}$
$\mathcal{N}$
$\Phi_{\mathcal{Y}}$
Softmap $S_{\mathcal{X}\mathcal{Y}}$
$L(\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}}) = \frac{1}{n_\beta} \sum \| S_{\mathcal{X}\mathcal{Y}} P_{\mathcal{X}} - \Pi^{gt}_{\mathcal{X}\mathcal{Y}} P_{\mathcal{X}} \|_2^2.$
Softmax($\widehat{\Phi}_{\mathcal{X}}$ $\Phi_{\mathcal{Y}}$)
$\Phi_{\mathcal{Y}}$

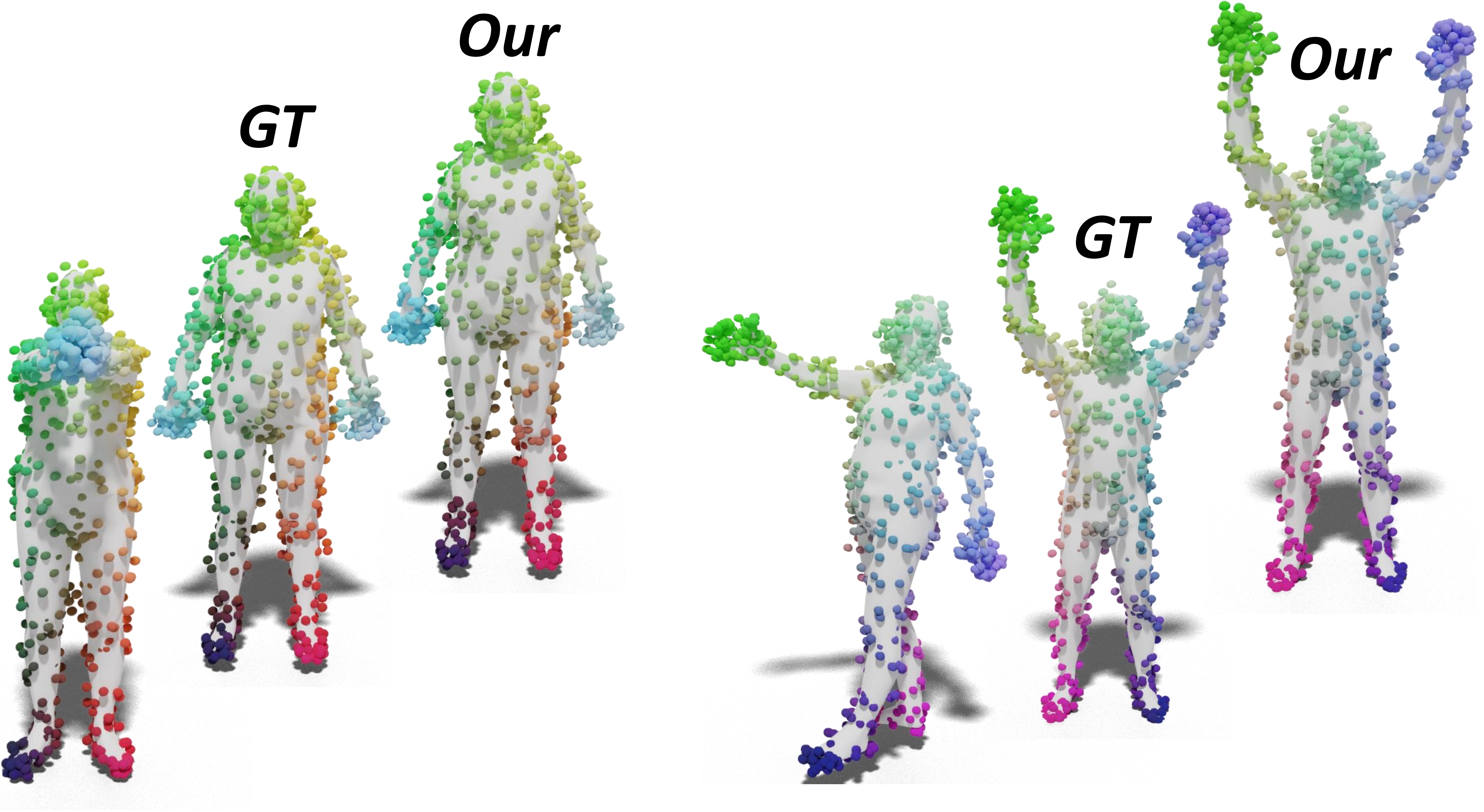


$\Phi_{\mathcal{X}}$
$G_{\mathcal{X}}$
$\Phi_{\mathcal{X}}$
$P_{\mathcal{X}}$
G
$G_{\mathcal{X}}$
$L(G_{\mathcal{X}}, G_{\mathcal{Y}}) = \|A^{gt}_{\mathcal{X}\mathcal{Y}} - \widehat{A}_{\mathcal{X}\mathcal{Y}}\|_2$
Θ
adjoint $\widehat{A}_{\mathcal{X}\mathcal{Y}}$
$L(G_{\mathcal{X}}, G_{\mathcal{Y}})$
$P_{\mathcal{Y}}$
G
$G_{\mathcal{Y}}$
$\Phi_{\mathcal{Y}}$
$\Phi_{\mathcal{Y}}$
$G_{\mathcal{Y}}$


$\Phi_{\mathcal{X}}$
$G_{\mathcal{X}}$
$\hat{A}_{\mathcal{X}\mathcal{Y}}$
$\Phi_{\mathcal{Y}}$
$G_{\mathcal{Y}}$


$\Phi_{\mathcal{X}}$
$G_{\mathcal{X}}$
$\widehat{A}_{\mathcal{XY}}$
$\widehat{\Phi}_{\mathcal{X}}$
$\Phi_{\mathcal{Y}}$
$G_{\mathcal{Y}}$


$\Phi_{\mathcal{X}}$
$G_{\mathcal{X}}$
$\hat{A}_{\mathcal{X}\mathcal{Y}}$
$\widehat{\Phi}_{\mathcal{X}}$
$\Phi_{\mathcal{Y}}$
$G_{\mathcal{Y}}$


GT
Our
GT
Our

[Correspondence Learning via Linearly-invariant Embedding, Marin et al. NeurIPS 2020]

Code is available: https://github.com/riccardomarin/Diff-FMaps