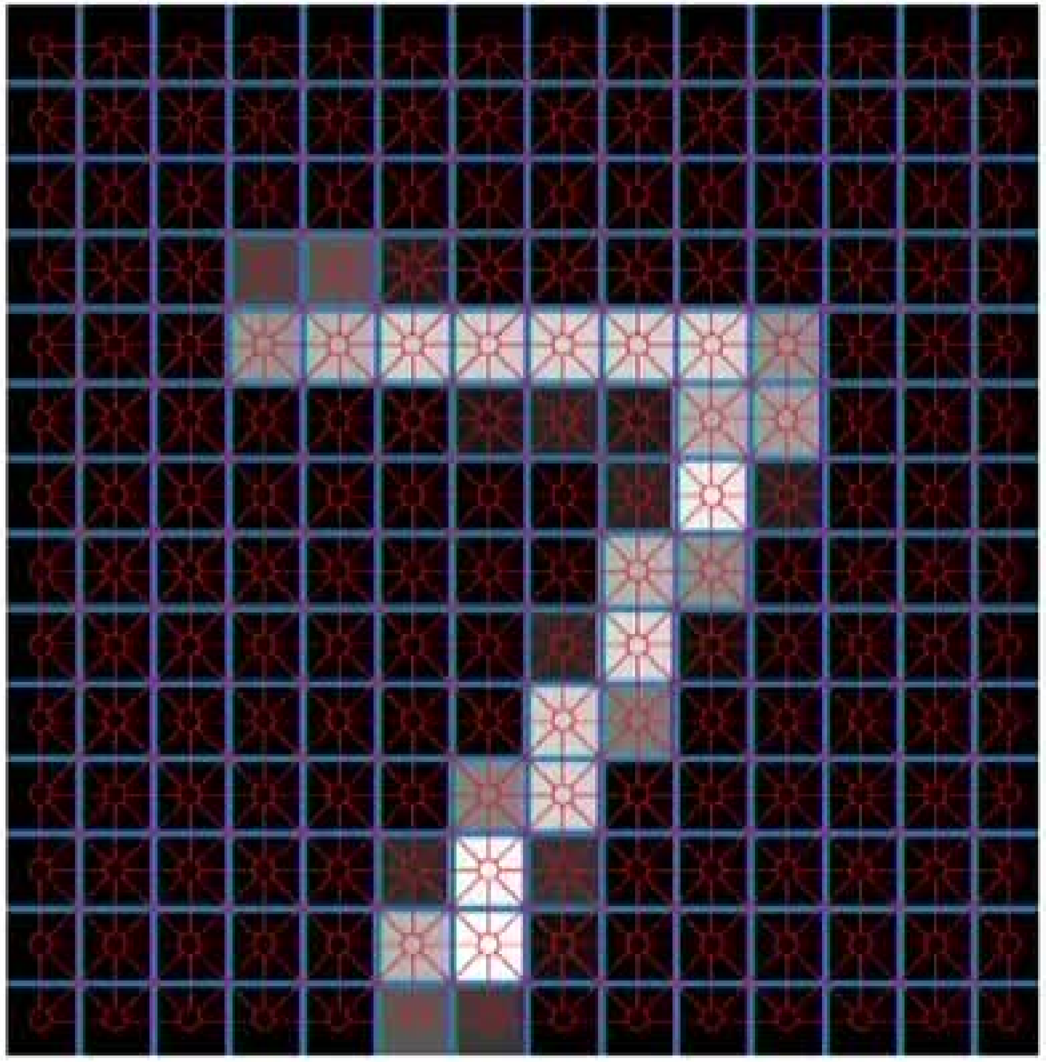
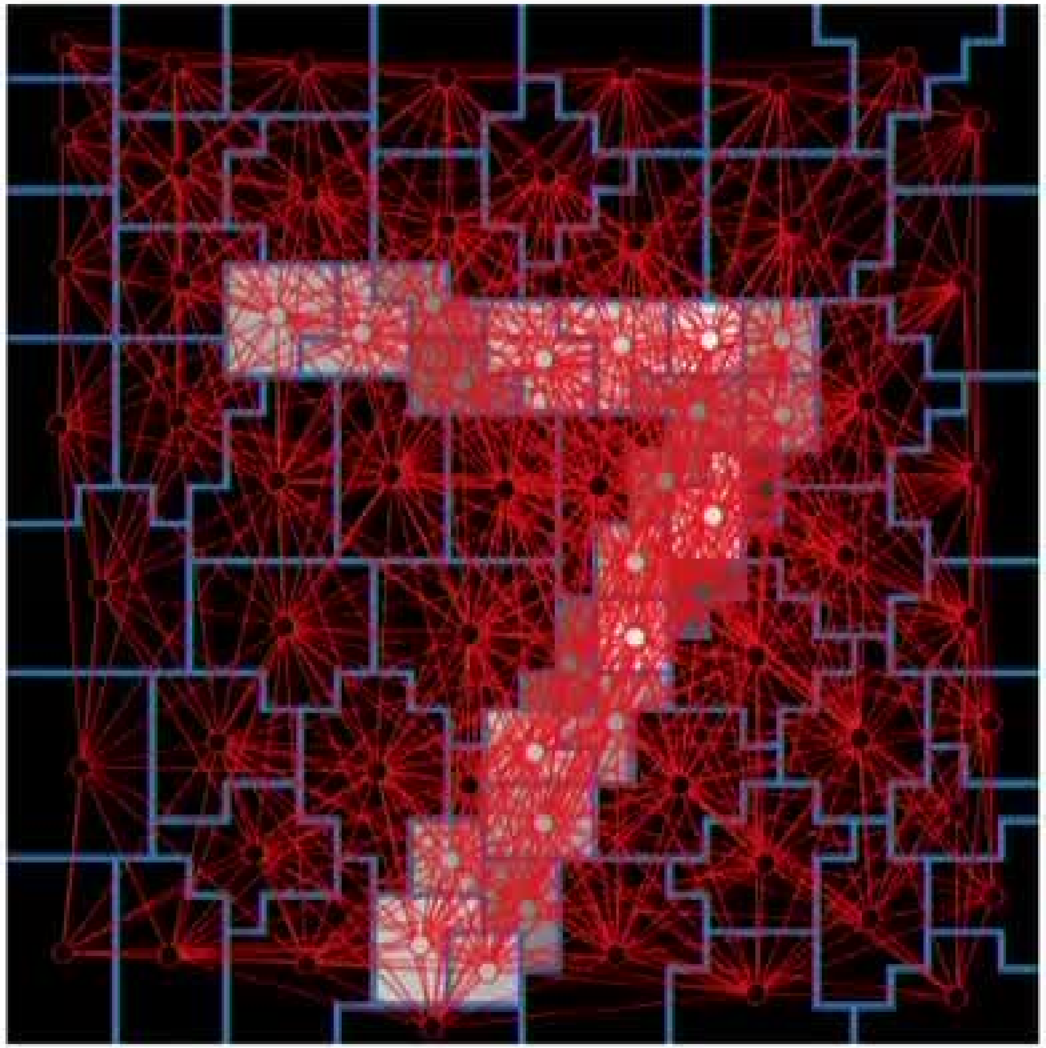
Example: MNIST digits classification



Regular grid
(fixed graph, different data)



Superpixels
(different graph and data)

Monti et al. 2017

Example: MNIST digits classification

Dataset	LeNet-5[1]	ChebNet[2]
*Full grid	99.33%	99.14%
* $\frac{1}{4}$ grid	98.59%	97.51%
300 Superpixels	-	**88.05%**
150 Superpixels	-	**80.94%**
75 Superpixels	-	**75.62%**

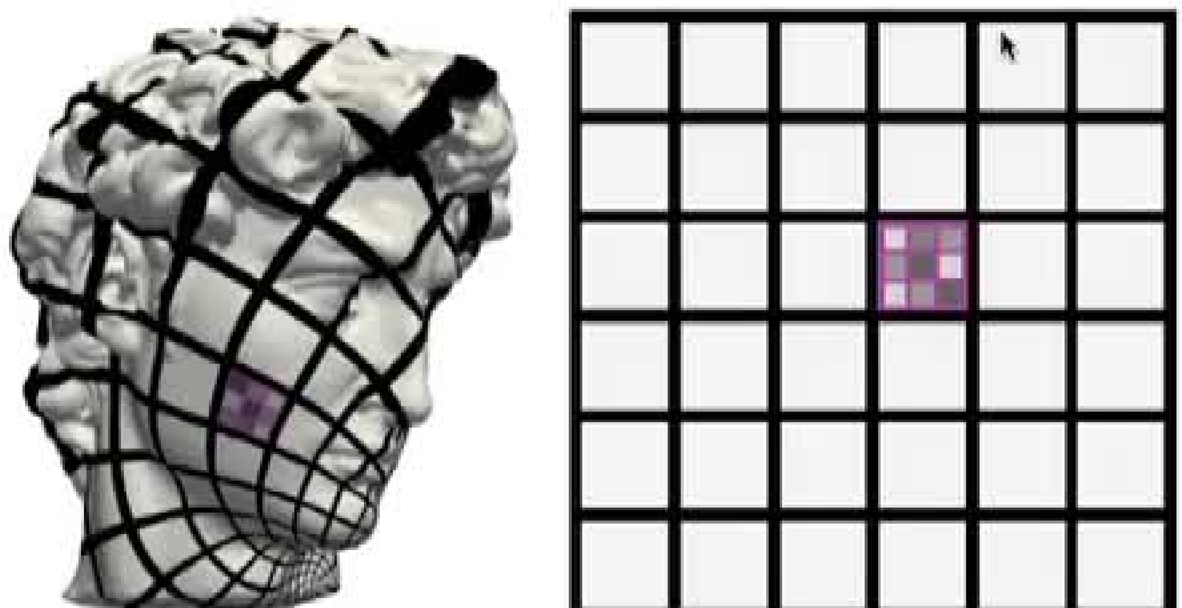
Classification accuracy of different methods on MNIST dataset
*All images have the same graph

Monti et al. 2017; [1]LeCun et al. 1998; [2]Defferrard, Bresson, Vandergheynst 2016

Different formulations of non-Euclidean CNNs

Spectral domain

Spatial domain



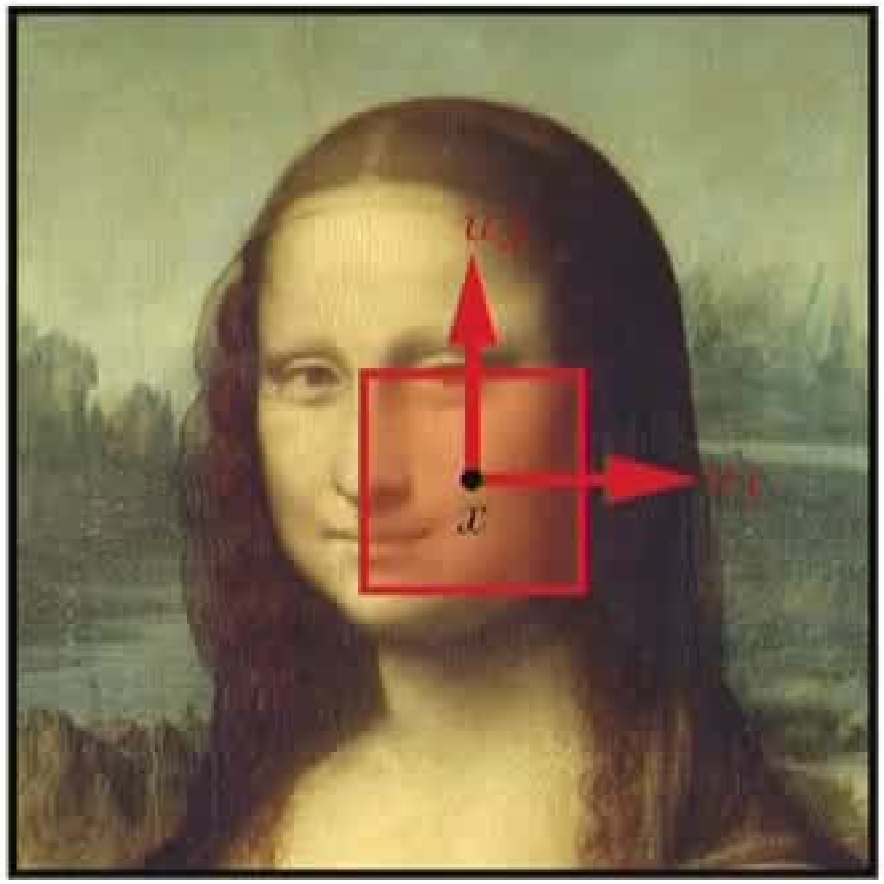
Parametric domain

Spatial domain (charting-based) geometric deep learning methods

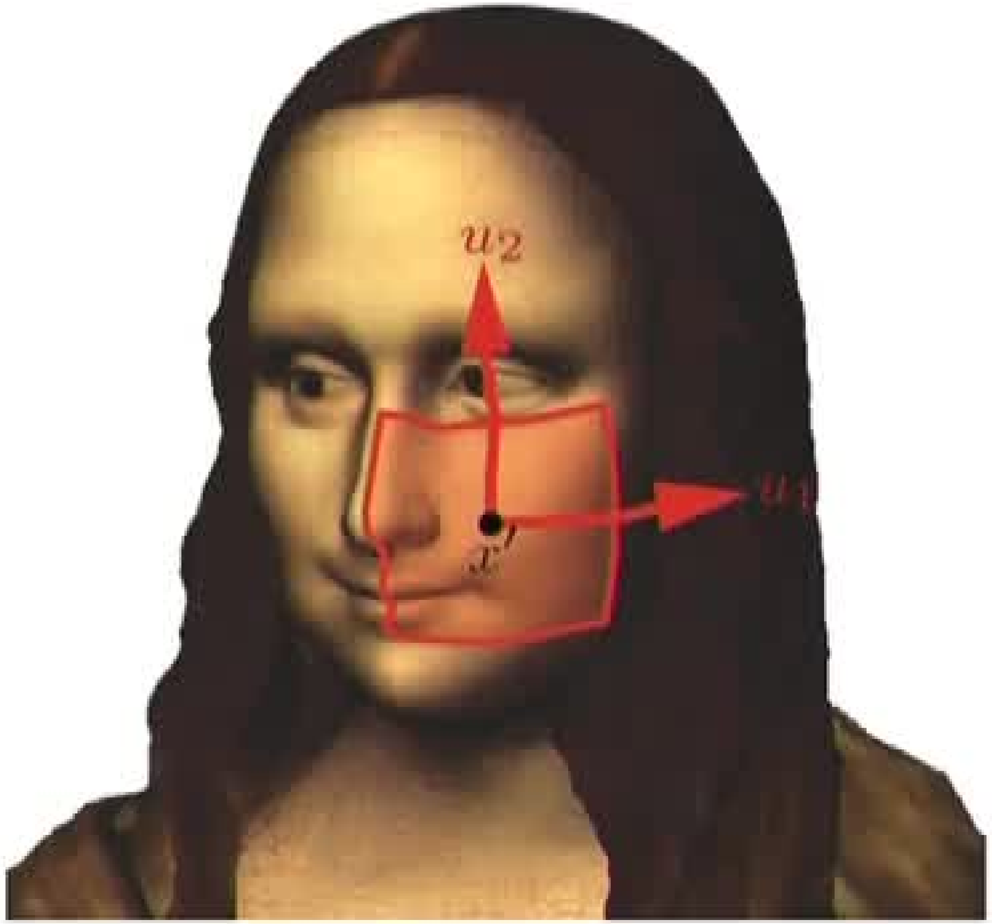
Convolution

Euclidean	**Non-Euclidean**
Spatial domain	
$(f \star g)(x) = \int_{-\pi}^{\pi} f(x')g(x-x')dx'$	?
Spectral domain	
$\widehat{(f \star g)}(\omega) = \hat{f}(\omega) \cdot \hat{g}(\omega)$	$\widehat{(f \star g)}_k = \langle f, \phi_k \rangle_{L^2(\mathcal{X})} \langle g, \phi_k \rangle_{L^2(\mathcal{X})}$
'Convolution Theorem'	

Patch operators



Image



Manifold

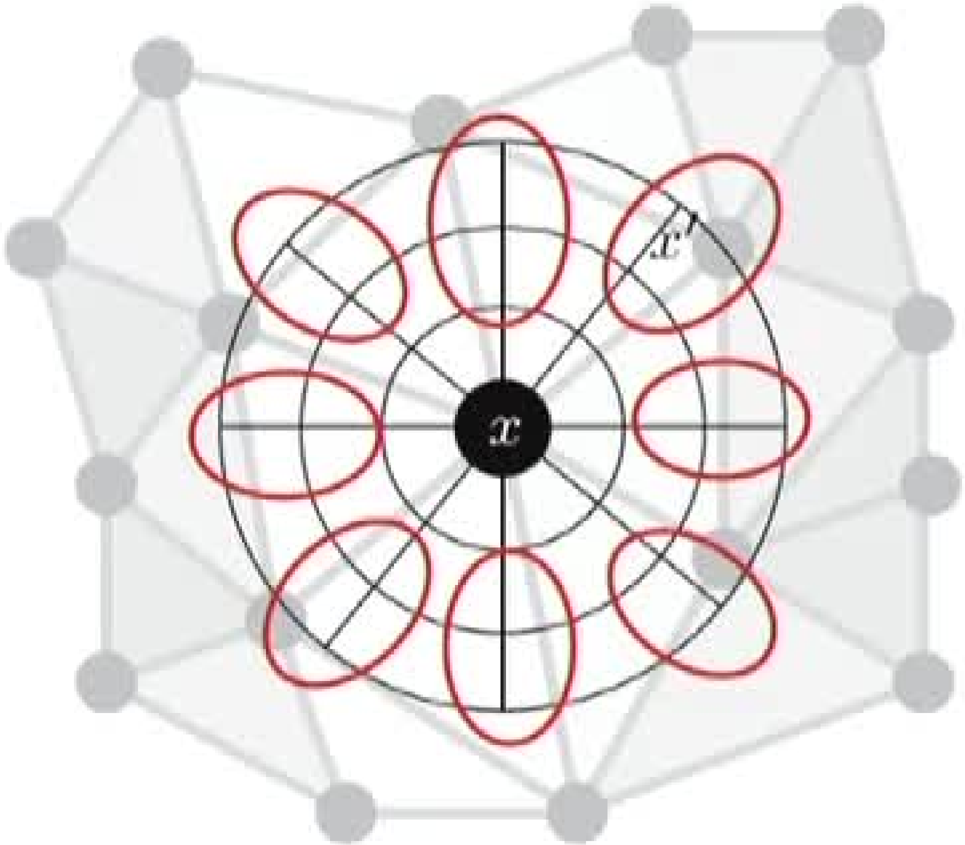
Boscaini et al. 2015

Convolution on manifolds

- Local system of coordinates $\mathbf{u}(x, x')$ around x' (e.g. geodesic polar)

- Local weights $w_1(\mathbf{u}), \ldots, w_L(\mathbf{u})$ w.r.t. $\mathbf{u}$, e.g. Gaussians

$$w_\ell = \exp\left(-(\mathbf{u} - \boldsymbol{\mu}_\ell)^\top \boldsymbol{\Sigma}_\ell^{-1} (\mathbf{u} - \boldsymbol{\mu}_\ell)\right)$$

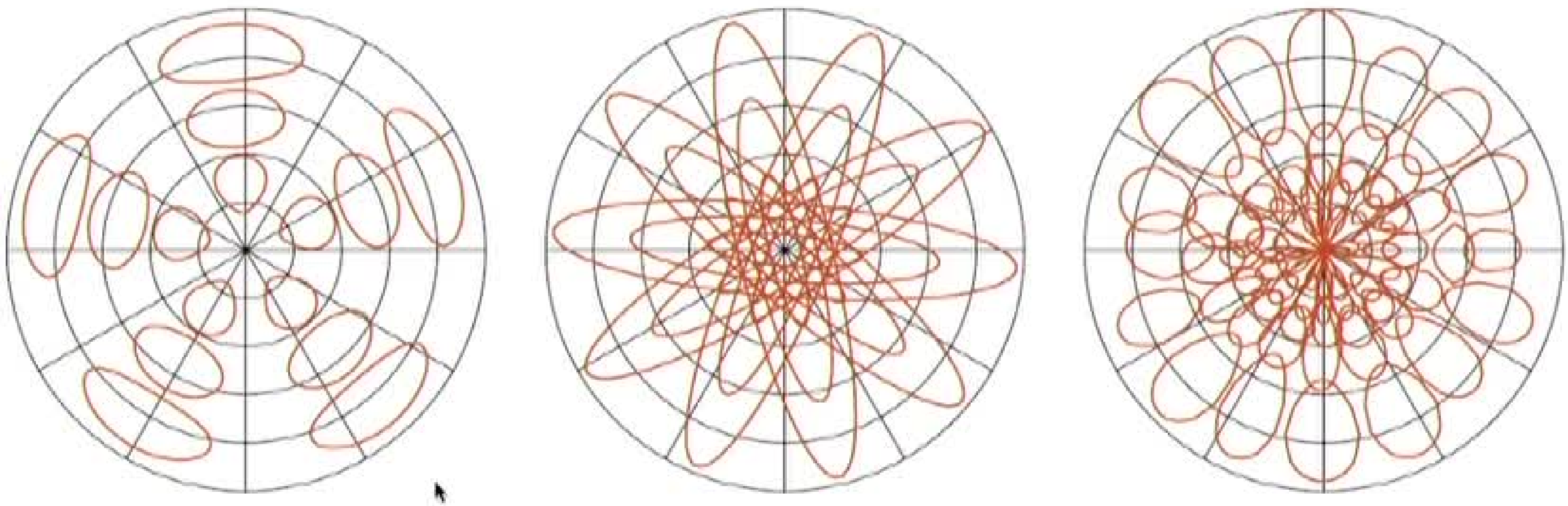


- Spatial convolution with filter g

$$(f \star g)(x) \propto \sum_{\ell=1}^{L} g_\ell \underbrace{\int_{\mathcal{X}} w_\ell(\mathbf{u}(x, x')) f(x')\, dx'}_{\text{patch operator}}$$

Masci et al. 2015; Boscaini et al. 2016; Monti et al. 2017

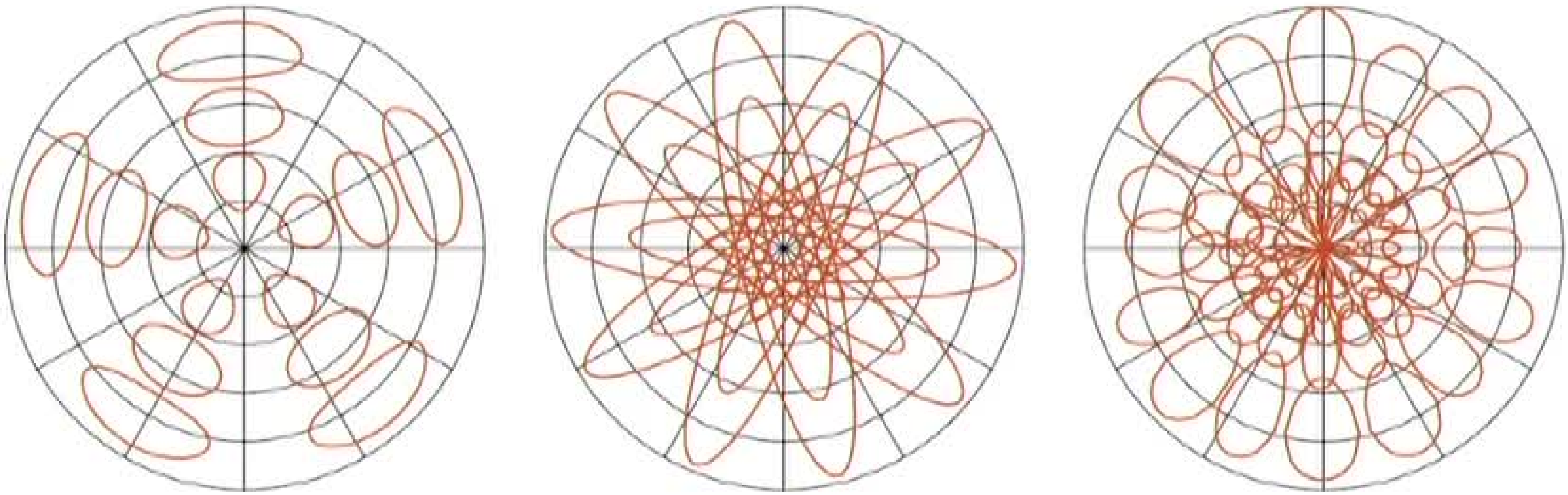
Patch operator weight functions



GCNN ACNN MoNet

Masci et al. 2015 (GCNN); Boscaini et al. 2016 (ACNN); Monti et al. 2017 (MoNet)

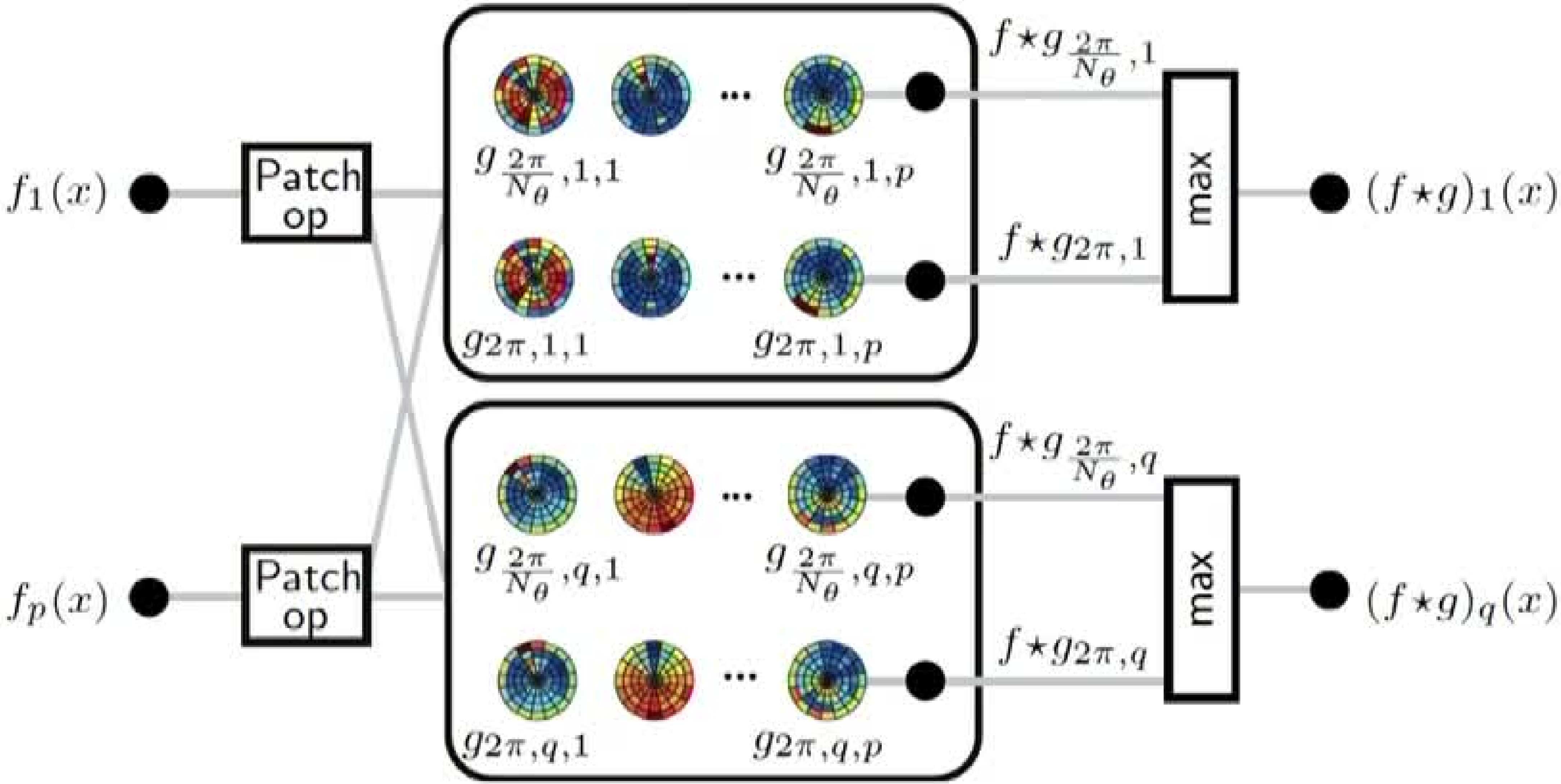
Patch operator weight functions



GCNN ACNN MoNet

Masci et al. 2015 (GCNN); Boscaini et al. 2016 (ACNN); Monti et al. 2017 (MoNet)

Geodesic convolution layer

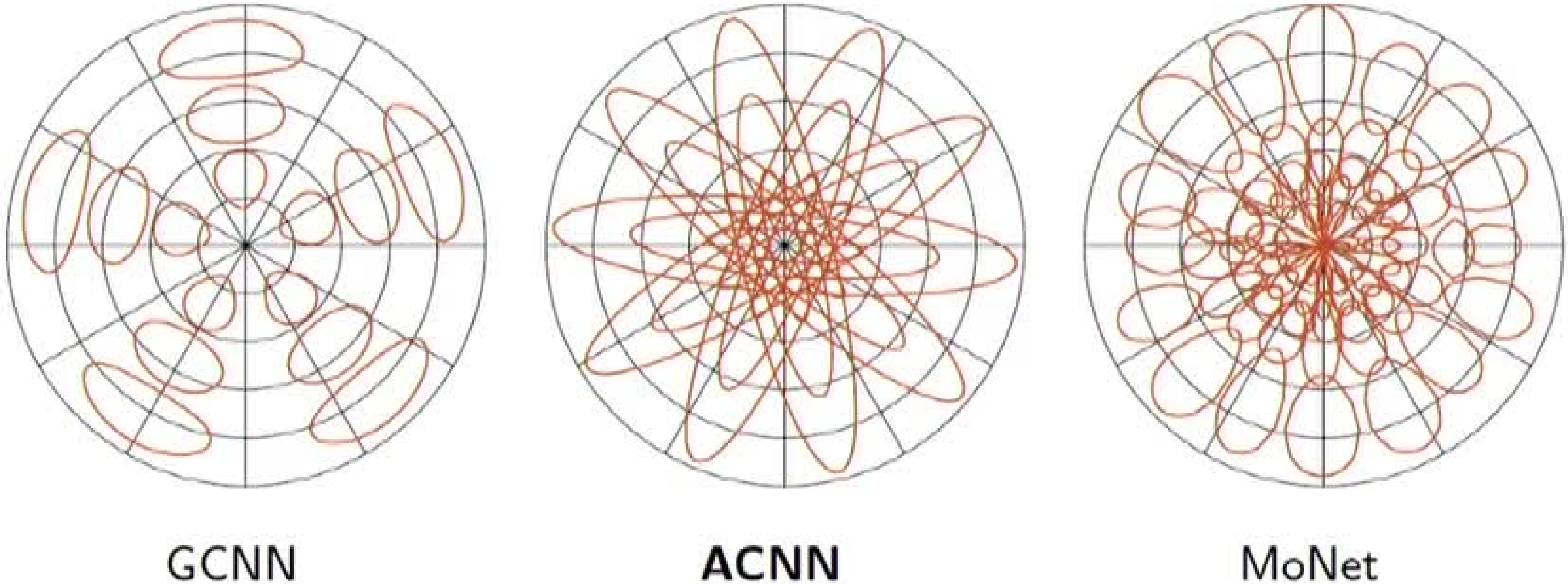


Conv. layer $$(f_l \star g)_{\Delta\theta,l}(x) = \xi\left(\sum_{\ell=1}^{p}(f_\ell \star g_{\Delta\theta,l,\ell})(x)\right)$$

Angular max pooling $$(f \star g)_l(x) = \max_{\Delta\theta}(f \star g)_{\Delta\theta,l}(x)$$

Masci et al. 2015

Patch operator weight functions



Masci et al. 2015 (GCNN); Boscaini et al. 2016 (ACNN); Monti et al. 2017 (MoNet)

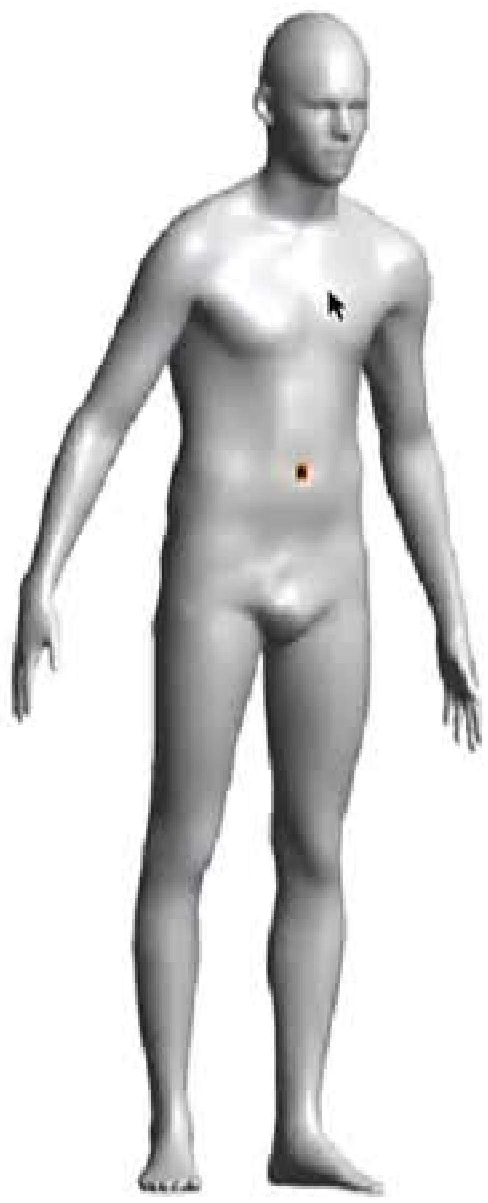
Anisotropic diffusion

$$f_t(x) = -\mathrm{div}(\mathbf{A}(x)\nabla f(x))$$

$\mathbf{A}(x)$ = heat conductivity tensor describing heat conduction properties of the material (diffusion speed is position + direction dependent)

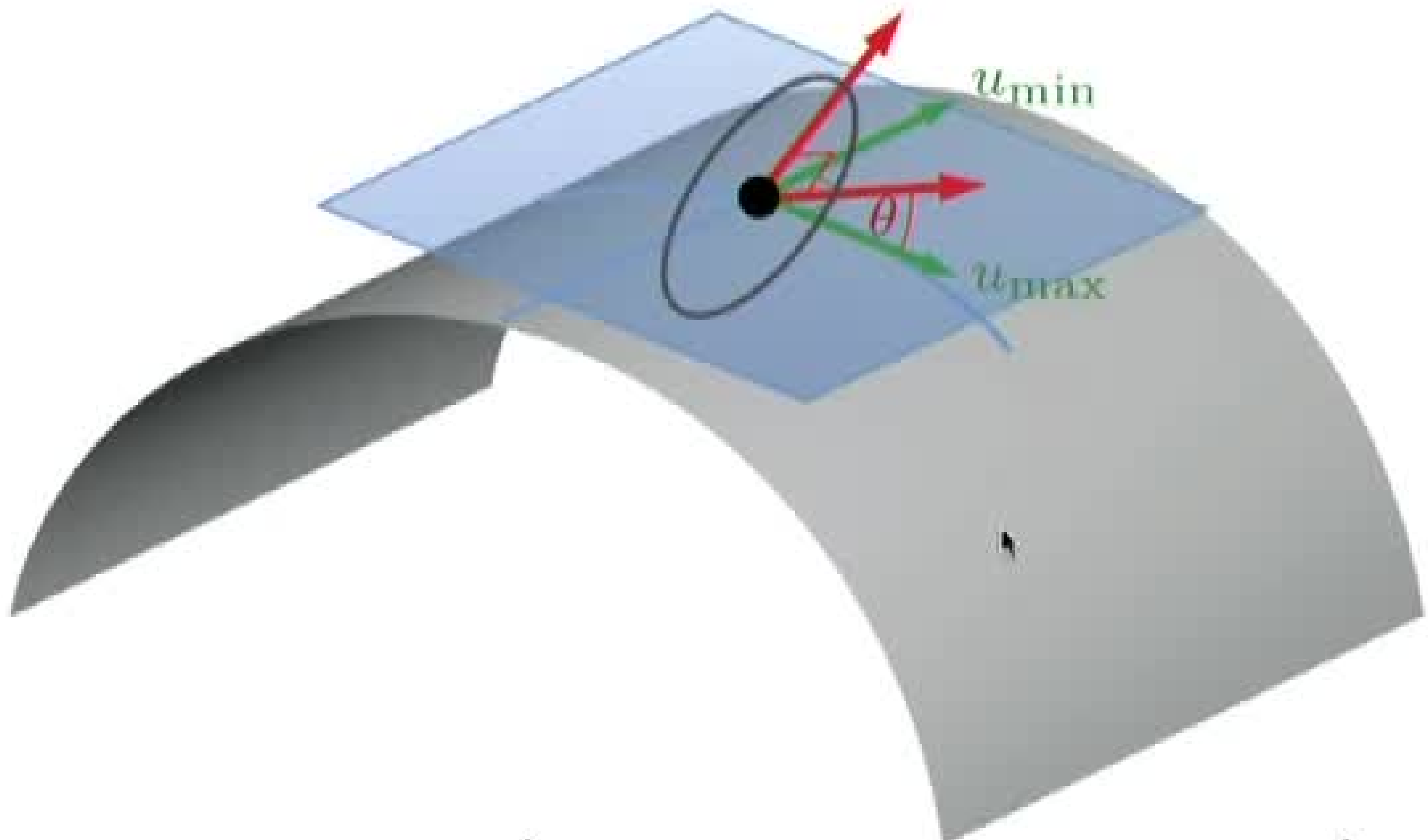
Anisotropic diffusion

Isotropic



Anisotropic

Anisotropic diffusion on manifolds

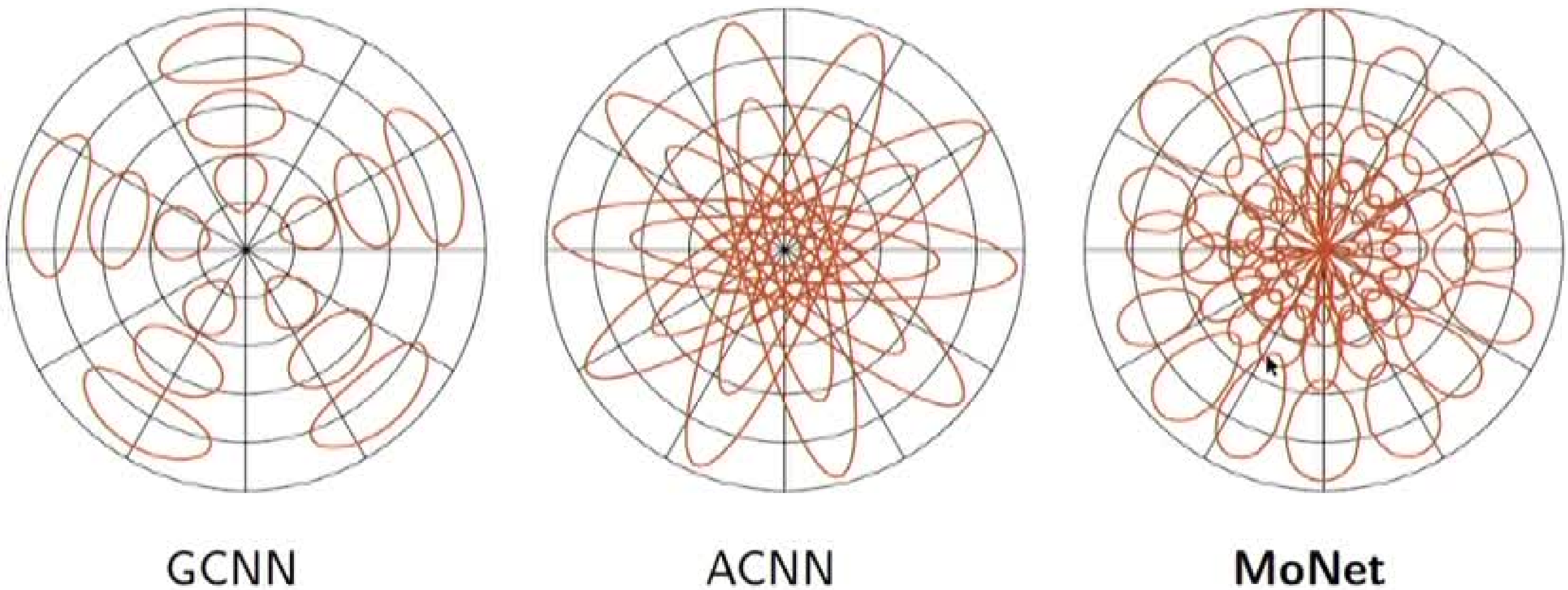


$$f_t(x) = -\mathrm{div}\left(\underbrace{\mathbf{R}_\theta \begin{pmatrix} \alpha & \\ & 1 \end{pmatrix} \mathbf{R}_\theta^\top}_{\mathbf{D}_{\alpha\theta}(x)} \nabla f(x) \right)$$

- Anisotropic Laplacian $\Delta_{\alpha\theta} f(x) = \mathrm{div}\left(D_{\alpha\theta}(x) \nabla f(x)\right)$
- $\theta =$ orientation w.r.t. max curvature direction
- $\alpha =$ 'elongation'

Andreux et al. 2014; Boscaini et al. 2016

Patch operator weight functions



Masci et al. 2015 (GCNN); Boscaini et al. 2016 (ACNN); Monti et al. 2016 (MoNet)

Mixture Model Network (MoNet)

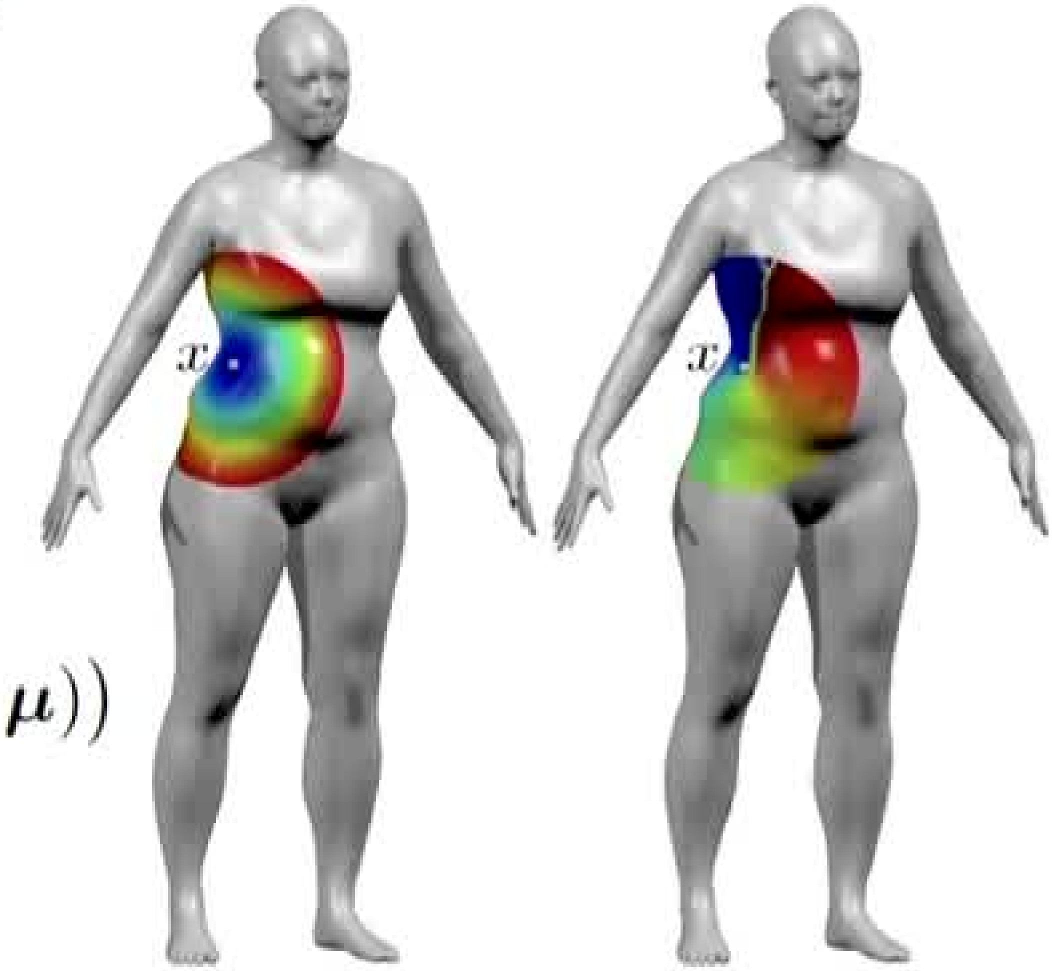
- Geodesic polar coordinates

$$\mathbf{u}(x,y) = (\rho(x,x'), \theta(x,x'))$$

- Gaussian weighting functions

$$w_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\mathbf{u}) = \exp\left(-\tfrac{1}{2}(\mathbf{u}-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\mathbf{u}-\boldsymbol{\mu})\right)$$

with learnable covariance $\boldsymbol{\Sigma}$ and mean $\boldsymbol{\mu}$



$u_1 = \rho$ $u_2 = \theta$

Spatial convolution

$$(f \star g)(x) \propto \int_{\mathcal{X}} \underbrace{\sum_{\ell=1}^{L} g_\ell w_{\boldsymbol{\mu}_\ell,\boldsymbol{\Sigma}_\ell}(\mathbf{u}(x,x'))}_{\text{Gaussian mixture}} f(x')\, dx'$$

where $g_1,\ldots,g_L$ are the spatial filter coefficients and $\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_L$ and $\boldsymbol{\Sigma}_1,\ldots,\boldsymbol{\Sigma}_L$ are patch operator parameters

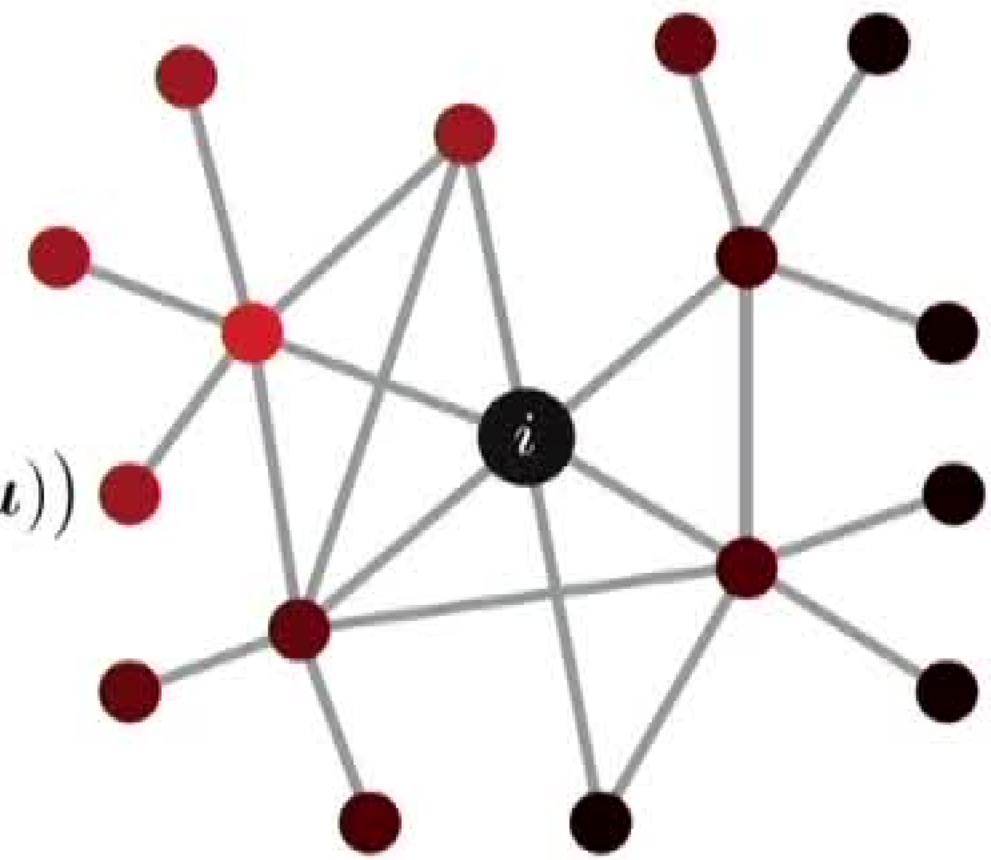
Monti et al. 2017

Mixture Model Network on graphs

- Local coordinates $\mathbf{u}_{ij}$, e.g. vertex degree, geodesic distance,...
- Gaussian weighting functions

  $$w_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\mathbf{u}) = \exp\left(-\tfrac{1}{2}(\mathbf{u}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{u}-\boldsymbol{\mu})\right)$$

  with learnable covariance $\boldsymbol{\Sigma}$ and mean $\boldsymbol{\mu}$



Local coordinates on graph

Spatial convolution

$$(f \star g)_i \propto \sum_{\ell=1}^{L} g_\ell \sum_{j=1}^{n} w_{\boldsymbol{\mu}_\ell,\boldsymbol{\Sigma}_\ell}(\mathbf{u}_{i,j}) f_j$$

where $g_1, \ldots, g_L$ are the spatial filter coefficients and $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_L$ and $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_L$ are patch operator parameters

Monti et al. 2017

Spectral vs Spatial methods

ChebNet filter

$$h_i = \sum_{\ell=0}^{r} \alpha_\ell (\mathbf{\Delta}^\ell \mathbf{f})_i$$

Spatial filter

$$h_i = \sum_{\ell=1}^{L} g_\ell (\mathbf{W}_\ell \mathbf{f})_i$$

ChebNet is a particular setting of spatial convolution with local weighting functions given by the powers of the Laplacian $\mathbf{W}_\ell = \mathbf{\Delta}^\ell$

Defferrard, Bresson, Vandergheynst 2016; Masci et al. 2015; Monti et al. 2017

Graph Attention Networks (GAT)

Main idea: neighborhood average

$$\mathbf{f}_i' = \sum_{j:(i,j)\in\mathcal{E}} \alpha_{ij}\mathbf{f}_j$$

weighted by attention score

$$\alpha_{ij} = \frac{e^{\xi([\mathbf{f}_i\mathbf{W},\,\mathbf{f}_j\mathbf{W}]\mathbf{a})}}{\sum_{k:(i,k)\in\mathcal{E}} e^{\xi([\mathbf{f}_i\mathbf{W},\,\mathbf{f}_k\mathbf{W}]\mathbf{a})}}$$

which is a learnable transformation of the local features with learnable parameters $\mathbf{W}$, $\mathbf{a}$

Particular case of MoNet-type architectures!

Veličković et al. 2018

Dual/Primal Graph CNN (DPGCNN)

Alternate GAT-type convolutions applied on primal and dual graphs

- Dual convolution on $\tilde{\mathcal{G}}$:

$$\tilde{\mathbf{f}}'_{ij} = \xi\left(\sum_{r\in\mathcal{N}_i}\tilde{\alpha}_{ij,ir}\tilde{\mathbf{f}}_{ir}\tilde{\mathbf{W}} + \sum_{t\in\mathcal{N}_j}\tilde{\alpha}_{ij,tj}\tilde{\mathbf{f}}_{tj}\tilde{\mathbf{W}}\right)$$

$$\tilde{\alpha}_{ij,ik} = \frac{e^{\xi([\tilde{\mathbf{f}}_{ij}\tilde{\mathbf{W}},\,\tilde{\mathbf{f}}_{ik}\tilde{\mathbf{W}}]\tilde{\mathbf{a}})}}{\sum_{r\in\mathcal{N}_i} e^{\xi([\tilde{\mathbf{f}}_{ij}\tilde{\mathbf{W}},\,\tilde{\mathbf{f}}_{ir}\tilde{\mathbf{W}}]\tilde{\mathbf{a}})} + \sum_{t\in\mathcal{N}_j} e^{\xi([\tilde{\mathbf{f}}_{ij}\tilde{\mathbf{W}},\,\tilde{\mathbf{f}}_{tj}\tilde{\mathbf{W}}]\tilde{\mathbf{a}})}}$$

- Primal convolution on $\mathcal{G}$:

$$\mathbf{f}'_i = \xi\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}\mathbf{f}_j\mathbf{W}\right) \qquad \alpha_{ij} = \frac{e^{\xi(\tilde{\mathbf{f}}'_{ij}\mathbf{a})}}{\sum_{k\in\mathcal{N}_i} e^{\xi(\tilde{\mathbf{f}}'_{ik}\mathbf{a})}}$$

Monti et al. 2018

Dual/Primal Graph CNN (DPGCNN)

Alternate GAT-type convolutions applied on primal and dual graphs

- Dual convolution on $\tilde{\mathcal{G}}$:

$$\tilde{\mathbf{f}}'_{ij} = \xi\left(\sum_{r\in\mathcal{N}_i}\tilde{\alpha}_{ij,ir}\tilde{\mathbf{f}}_{ir}\tilde{\mathbf{W}} + \sum_{t\in\mathcal{N}_j}\tilde{\alpha}_{ij,tj}\tilde{\mathbf{f}}_{tj}\tilde{\mathbf{W}}\right)$$

$$\tilde{\alpha}_{ij,ik} = \frac{e^{\xi([\tilde{\mathbf{f}}_{ij}\tilde{\mathbf{W}},\tilde{\mathbf{f}}_{ik}\tilde{\mathbf{W}}]\tilde{\mathbf{a}})}}{\sum_{r\in\mathcal{N}_i} e^{\xi([\tilde{\mathbf{f}}_{ij}\tilde{\mathbf{W}},\tilde{\mathbf{f}}_{ir}\tilde{\mathbf{W}}]\tilde{\mathbf{a}})} + \sum_{t\in\mathcal{N}_j} e^{\xi([\tilde{\mathbf{f}}_{ij}\tilde{\mathbf{W}},\tilde{\mathbf{f}}_{tj}\tilde{\mathbf{W}}]\tilde{\mathbf{a}})}}$$

- Primal convolution on $\mathcal{G}$:

$$\mathbf{f}'_i = \xi\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}\mathbf{f}_j\mathbf{W}\right) \qquad \alpha_{ij} = \frac{e^{\xi(\tilde{\mathbf{f}}'_{ij}\mathbf{a})}}{\sum_{k\in\mathcal{N}_i} e^{\xi(\tilde{\mathbf{f}}'_{ik}\mathbf{a})}}$$

Monti et al. 2018

Example: citation networks

Method	Cora[1]	CiteSeer[2]
Manifold Regularization[3]	59.5%	60.1%
Semidefinite Embedding[4]	59.0%	59.6%
Label Propagation[5]	68.0%	45.3%
DeepWalk[6]	67.2%	43.2%
Planetoid[7]	75.7%	64.7%
GCN[8]	81.6%	70.3%
MoNet[9]	81.7%	–
GAT[10]	83.0%	72.5%
DPGCN[11]	**83.3%**	**72.6%**

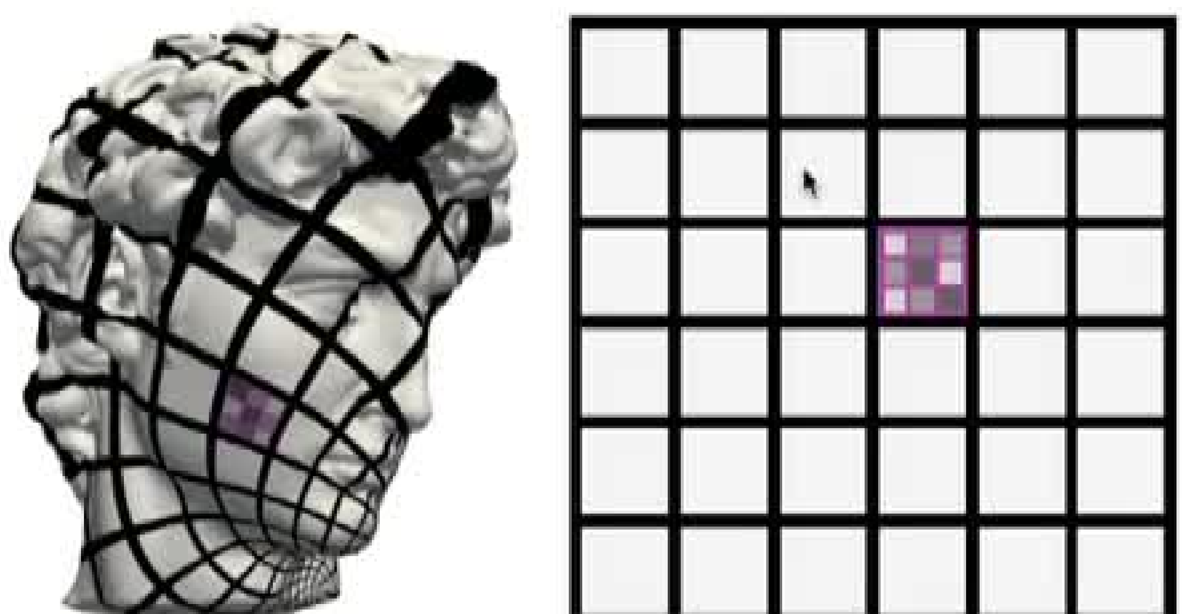
Classification accuracy of different methods on citation network datasets

Data: [1,2]Sen et al. 2008; methods: [3]Belkin et al. 2006; [4]Weston et al. 2012; [5]Zhu et al. 2003; [6]Perozzi et al. 2014; [7]Yang et al. 2016; [8]Kipf, Welling 2016 ; [9]Monti et al. 2017; [10]Veličković et al. 2018; [11]Monti et al. 2018

Different formulations of non-Euclidean CNNs

Spectral domain

Spatial domain

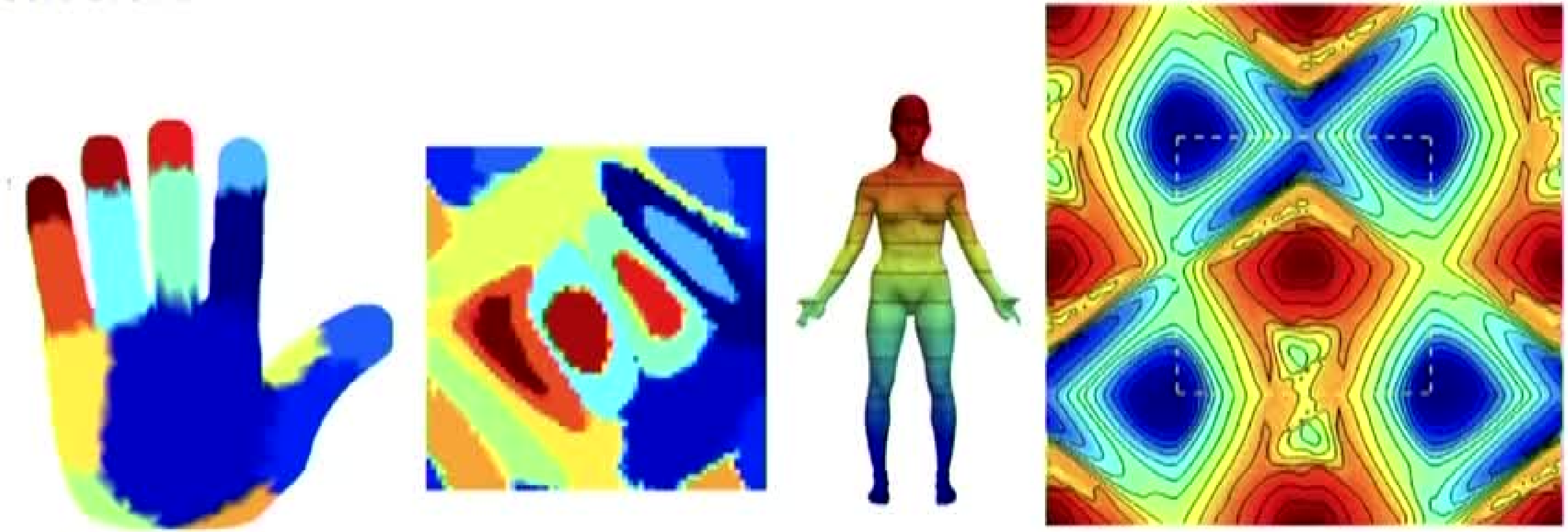


Parametric domain

Parametric domain geometric deep learning methods

Global parametrization

Map the input surface to some parametric domain with shift-invariant structure



- ☺ Allows to use standard CNNs (pull back convolution from the parametric space)
- ☺ Guaranteed invariance to some classes of transformations
- ☹ Parametrization may not be unique
- ☹ Embedding may introduce distortion

Sinha et al. 2016; Maron et al. 2017; Ezuz et al. 2017

Translation invariance on manifolds

Translation on manifold = locally Euclidean translation = flow along a non-vanishing vector field

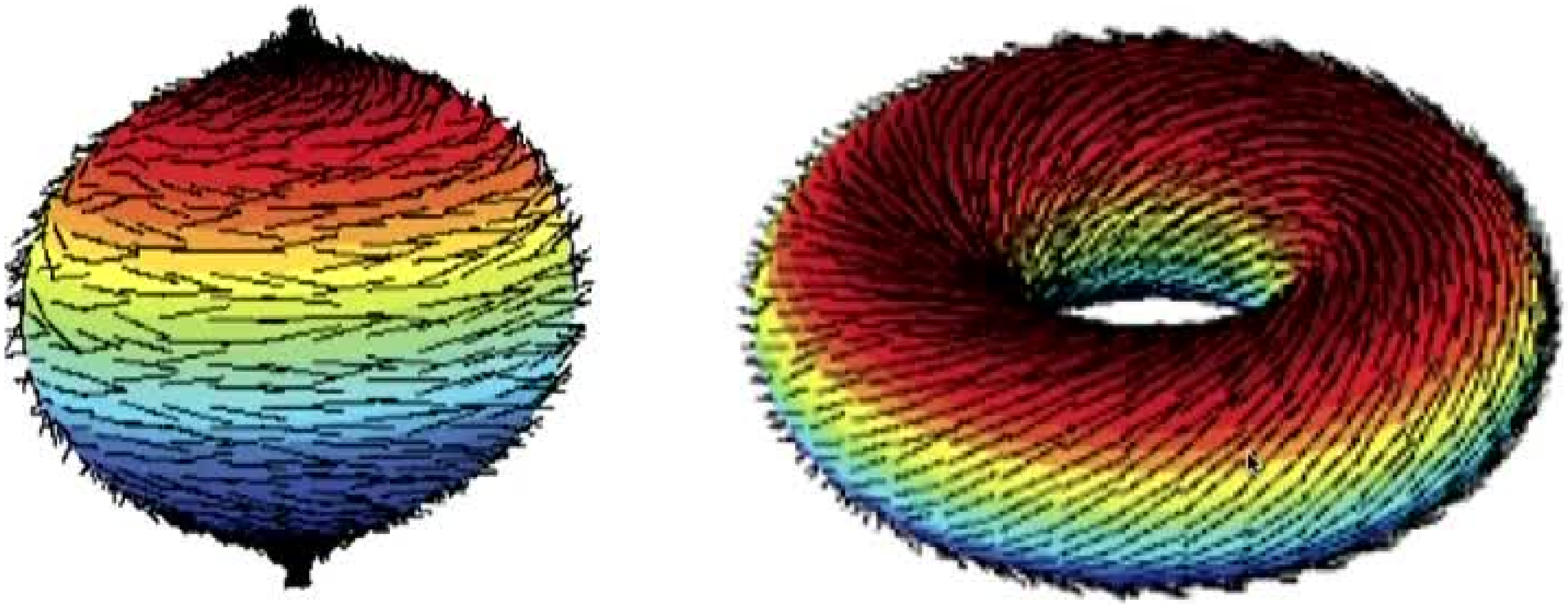
Poincaré-Hopf Theorem Non-vanishing vector field on a closed orientable compact 2-manifold implies manifold of genus 1 (torus)

Poincaré 1881; Hopf 1926

Translation invariance on manifolds

Translation on manifold = locally Euclidean translation = flow along a non-vanishing vector field

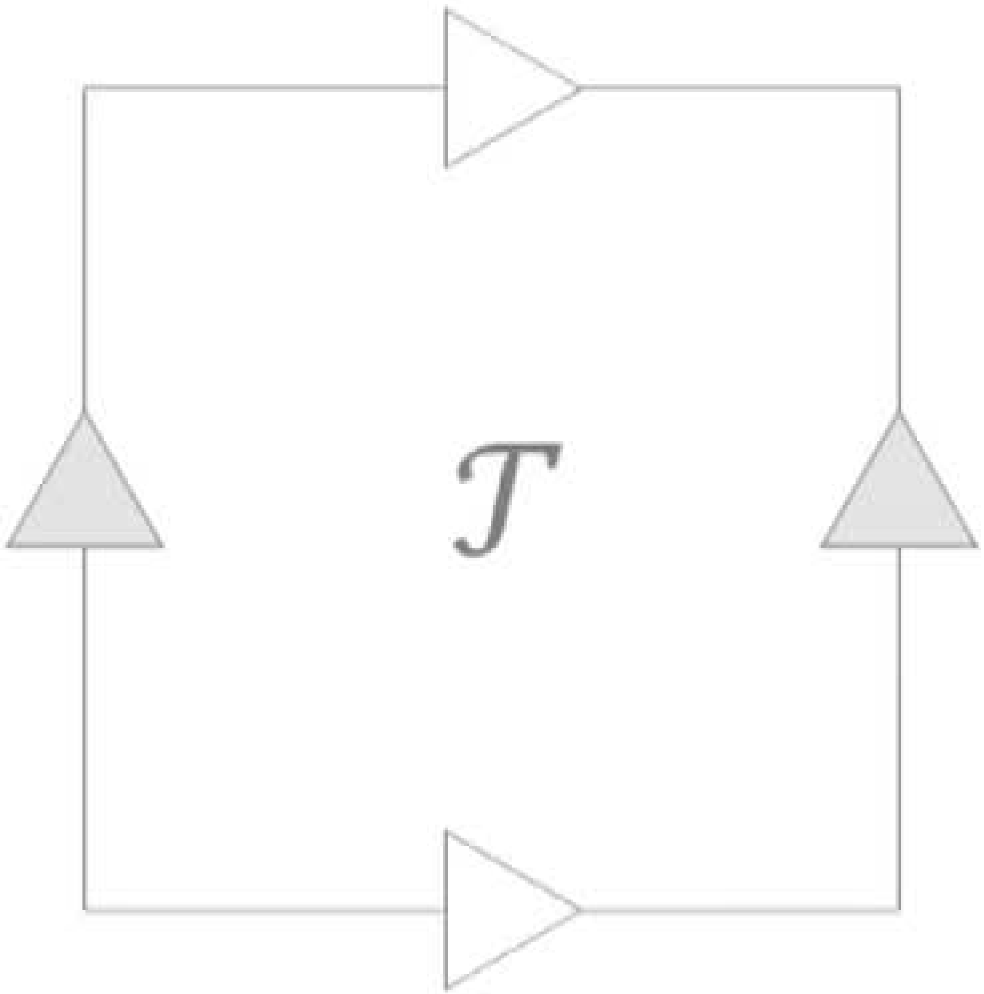
> **Poincaré-Hopf Theorem** Non-vanishing vector field on a closed orientable compact 2-manifold implies manifold of genus 1 (torus)

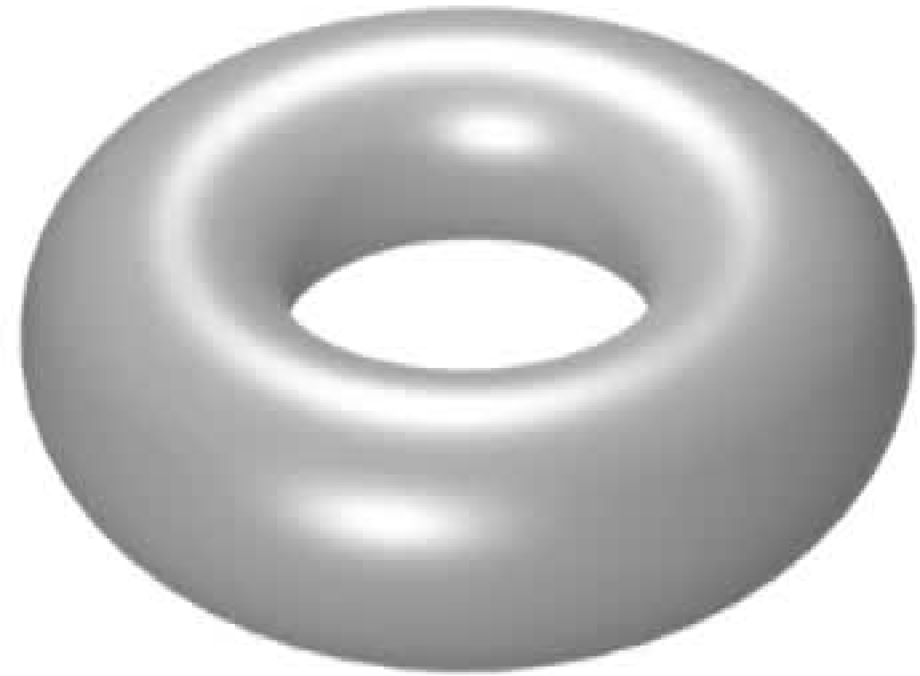


'Hairy Ball Theorem' states that a sphere cannot be combed

Poincaré 1881; Hopf 1926

Translation invariance on the torus

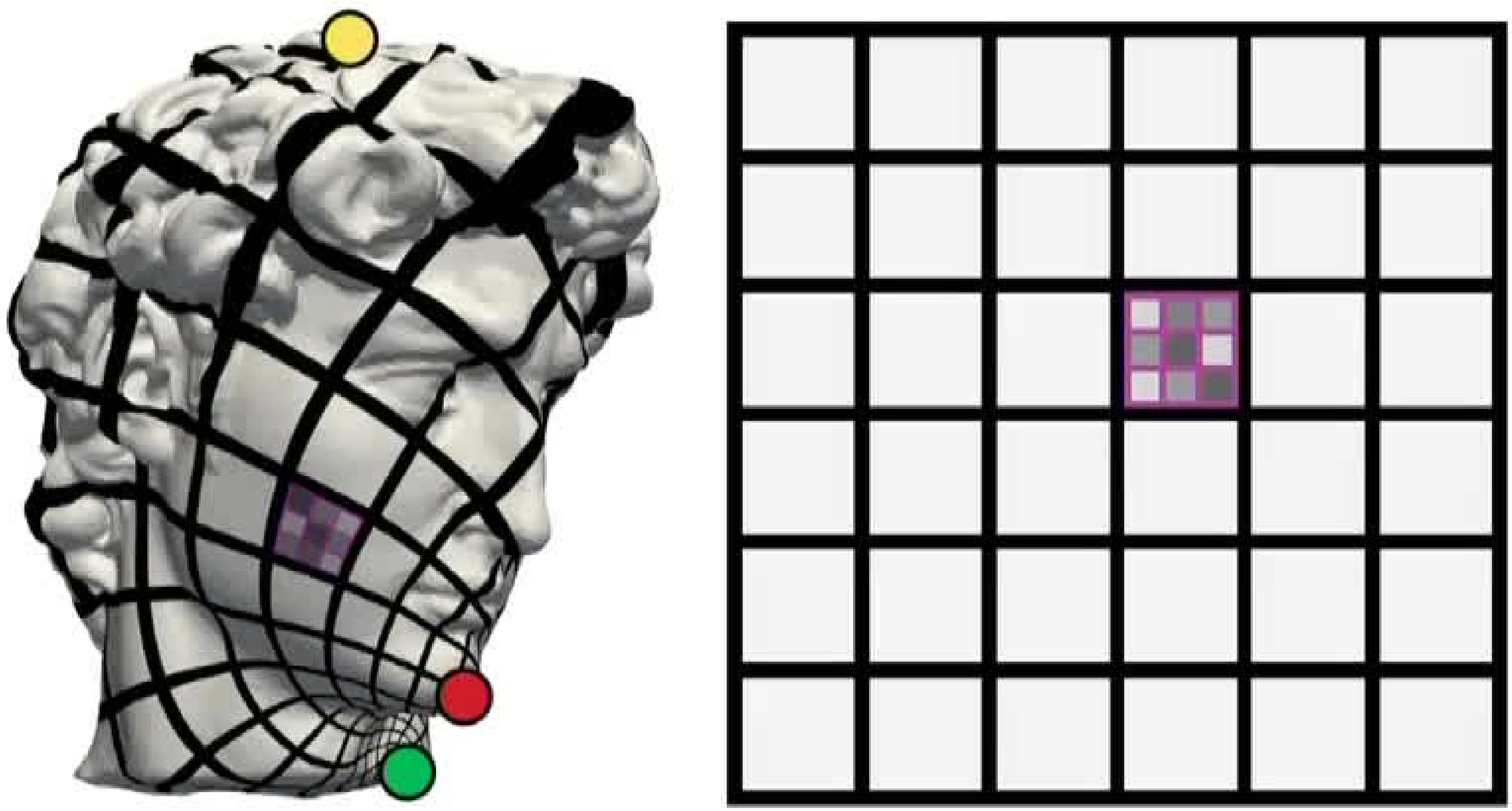




Torus is the only closed orientable surface admitting a translation group

Convolution on torus

For any triplet of points on $\mathcal{X}$, construct conformal homeomorphism from the 4-cover $\mathcal{X}^4$ to $\mathcal{T}$ using orbifold-Tutte method



Maron et al. 2017

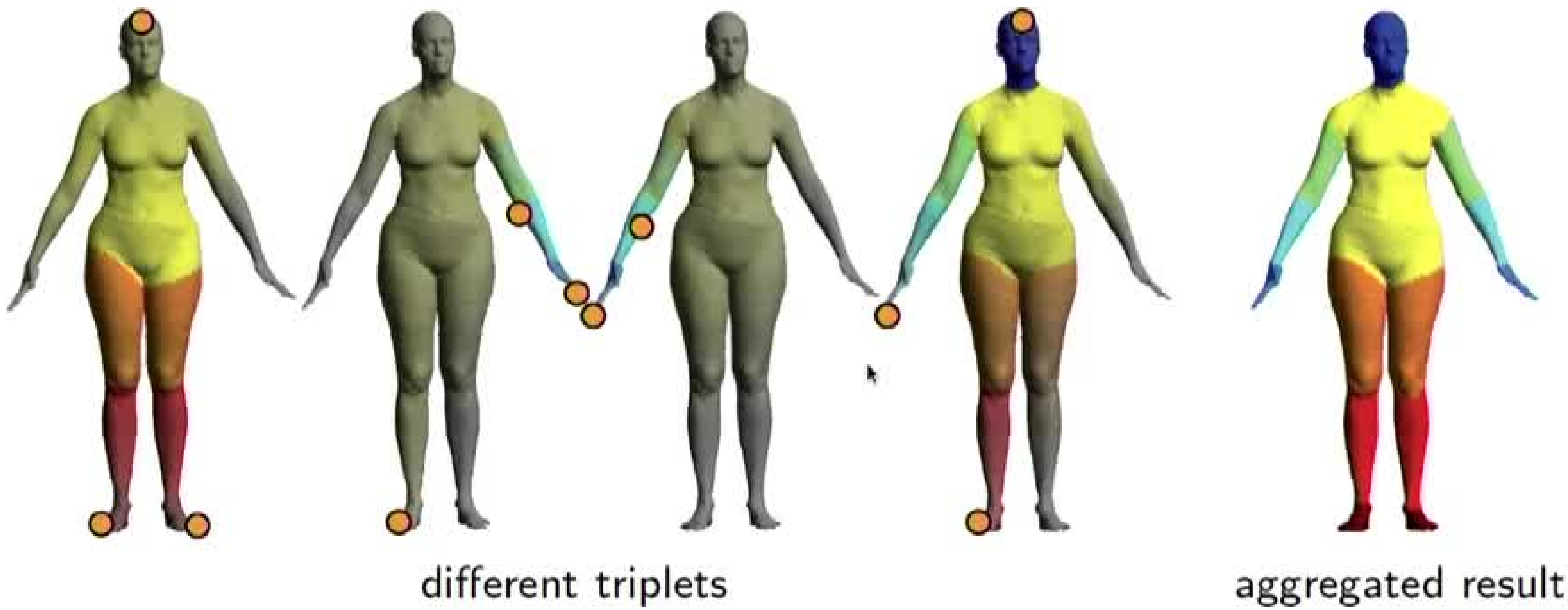
Conformal zoom

- Embedding depends on the choice of the triplets of points
- 'Conformal zoom' effect

Maron et al. 2017

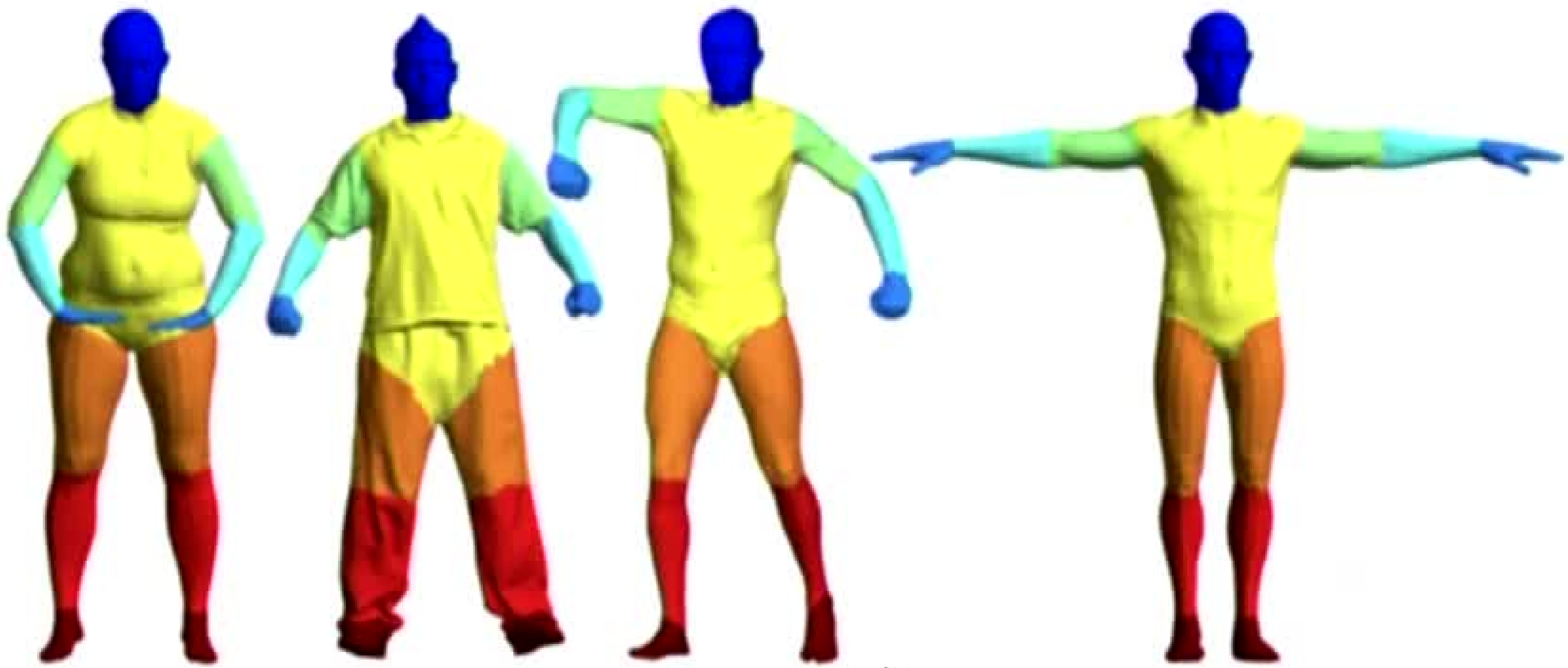
Conformal zoom

- Embedding depends on the choice of the triplets of points
- 'Conformal zoom' effect
- Choose multiple triples and aggregate results in training / test phase



Maron et al. 2017

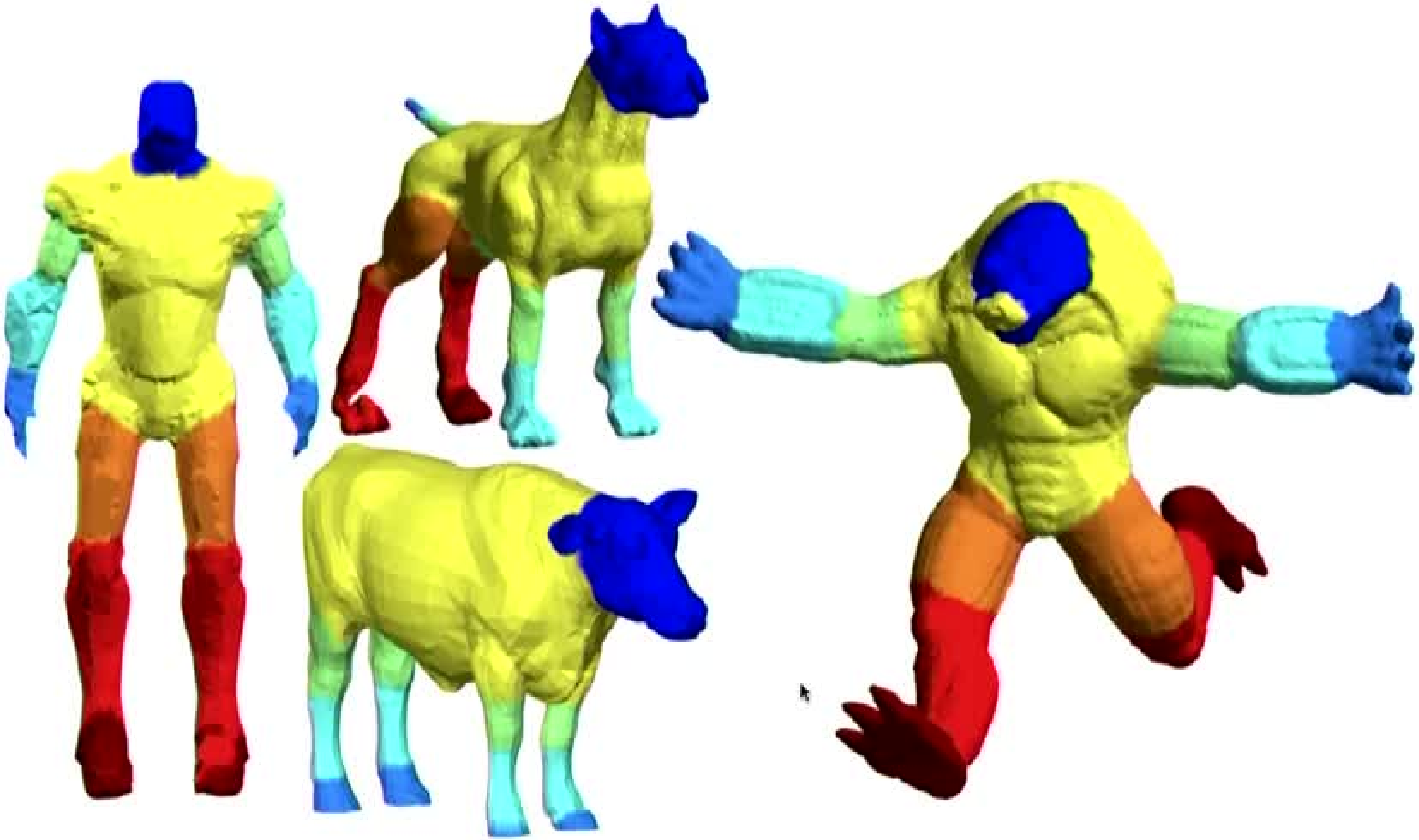
Example: shape segmentation with Toric CNN



Examples of shape segmentation obtained with Toric CNN

Maron et al. 2017

Example: shape segmentation with Toric CNN



Examples of shape segmentation obtained with Toric CNN

Maron et al. 2017

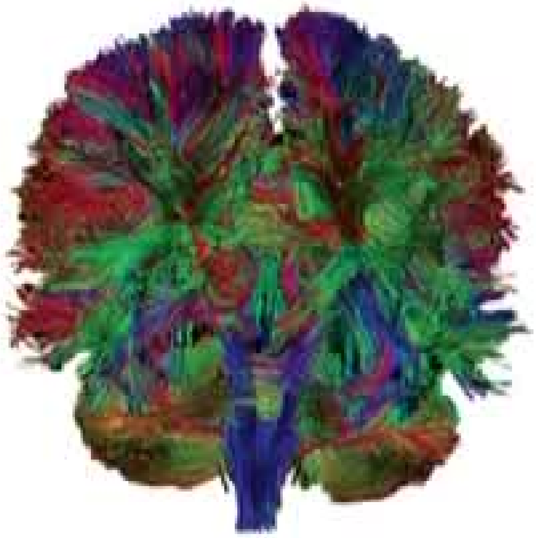
Application dealing with 3D data

Computer graphics

Virtual/augmented reality

Robotics

Autonomous driving



Medicine

Drug design

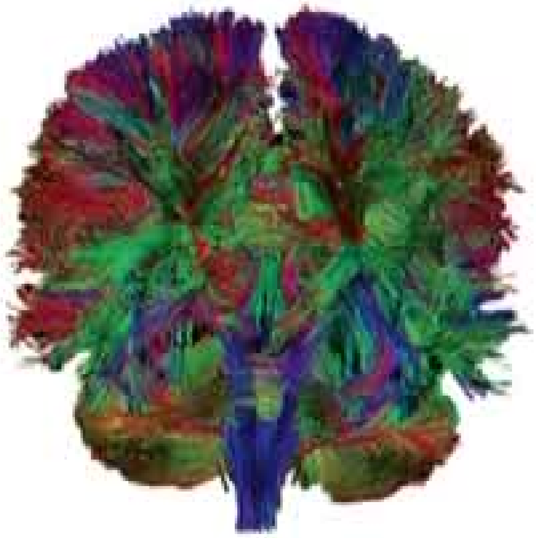
Application dealing with 3D data

Computer graphics

Virtual/augmented reality

Robotics

Autonomous driving

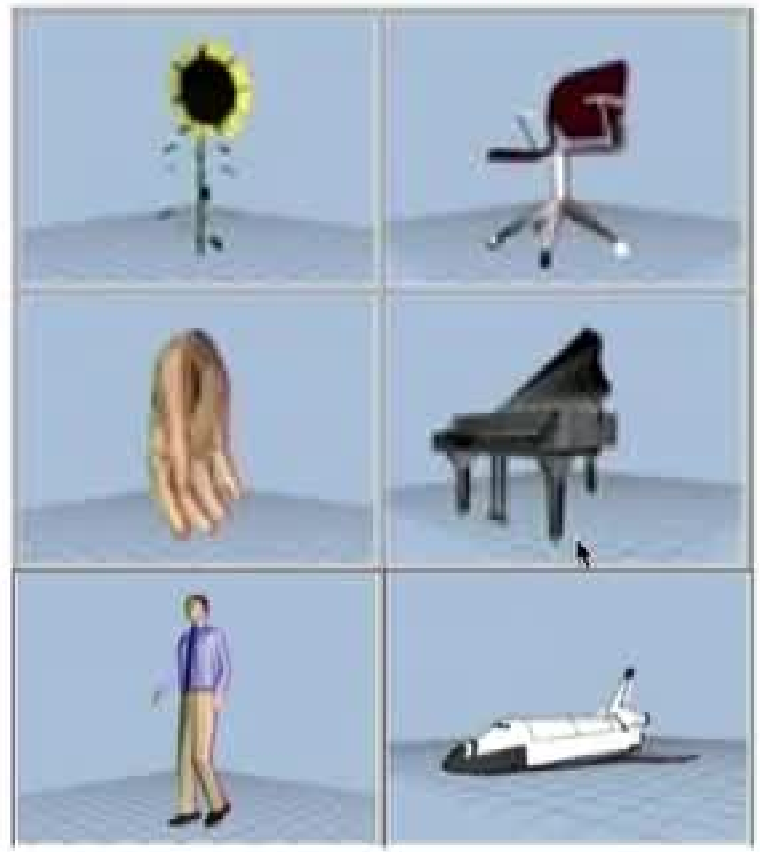


Medicine

Drug design

3D data

1998



2008

2018

>$10K

2005

$100

2010

$20

2014

???

2017

>$10K

2005

$100

2010

$20

2014

???

2017

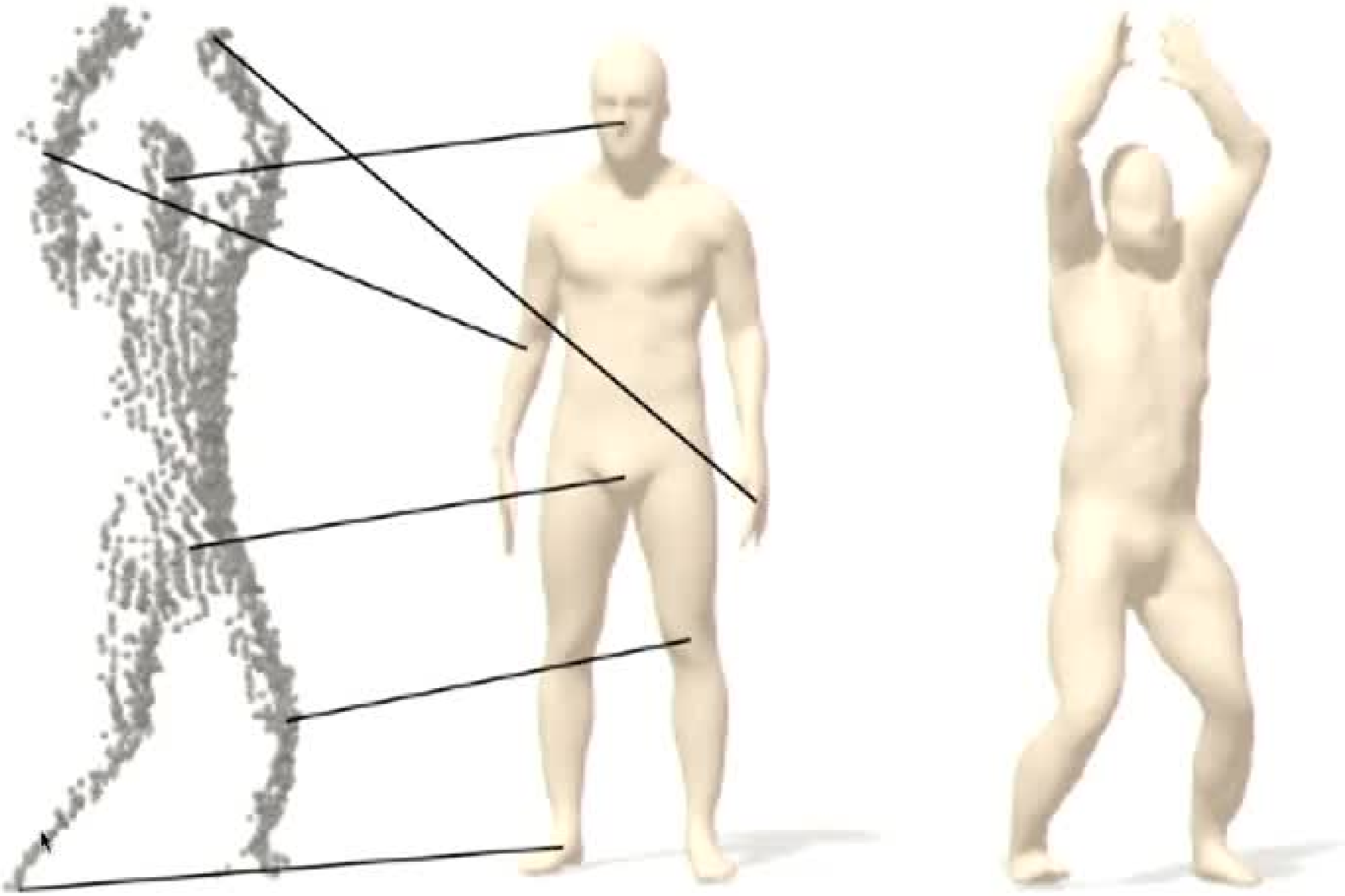
Faceshift (acquired by Apple in 2015)

Faceshift (acquired by Apple in 2015)

Faceshift (acquired by Apple in 2015)

Faceshift (acquired by Apple in 2015)

Analysis and synthesis

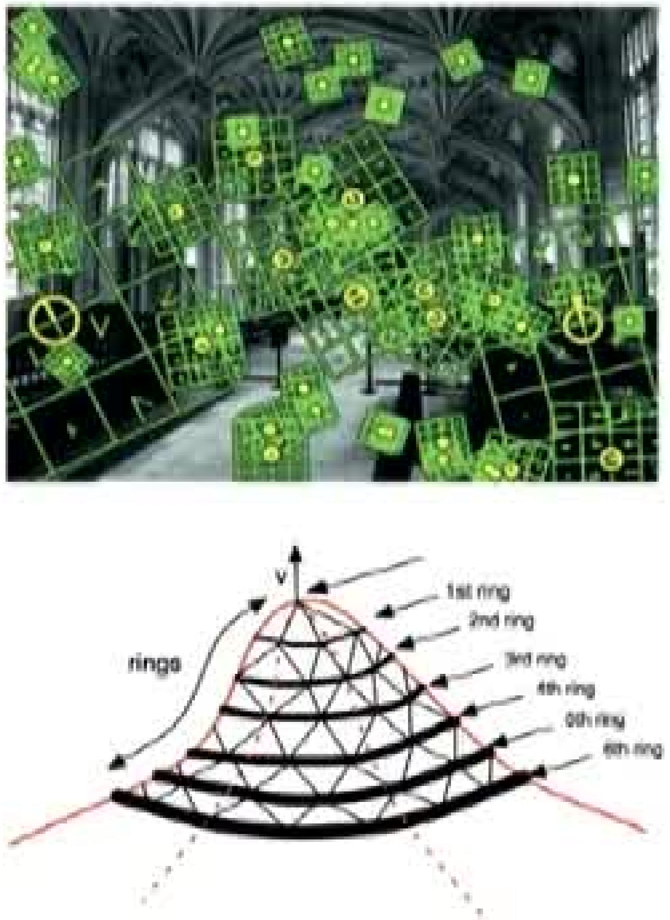


Input
3D scan

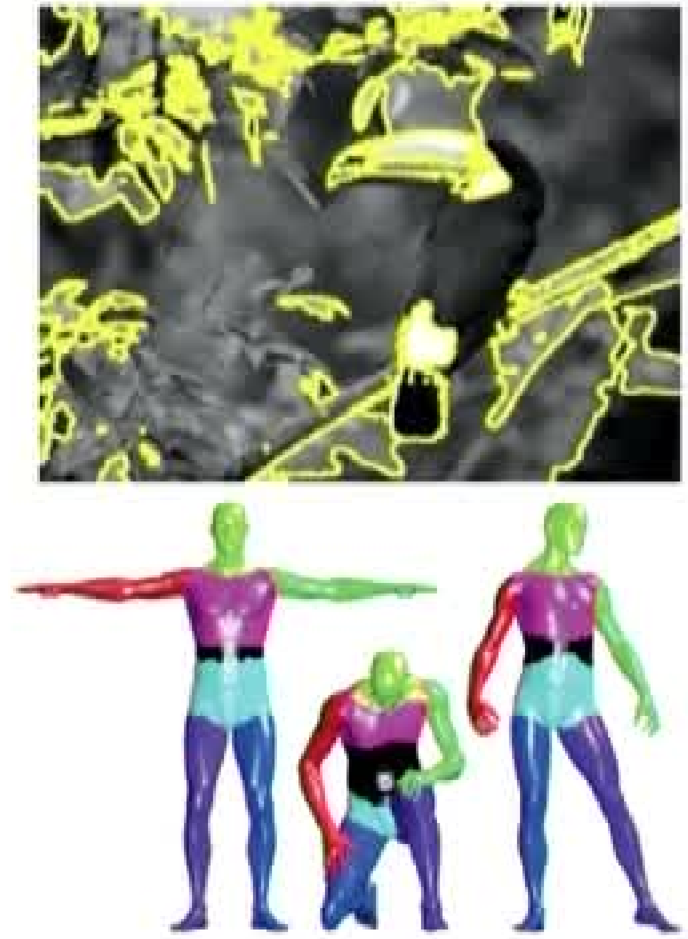
Correspondence to
Reference shape

Deformation of
reference shape

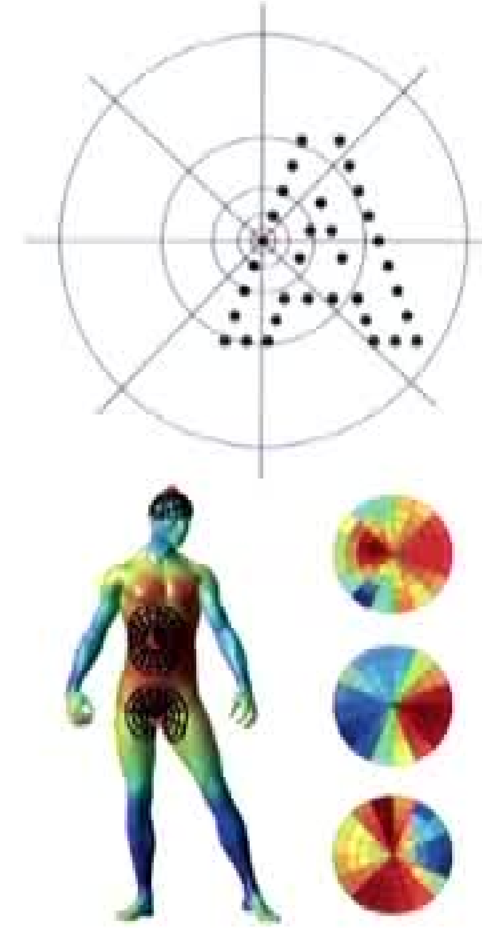
3D feature descriptors



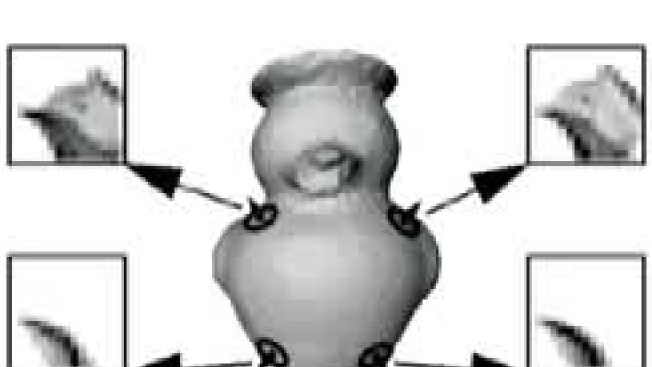
SIFT[1] / MeshHOG[2]



MSER[3] / ShapeMSER[4]



(Intrinsic[6]) Shape context[5]



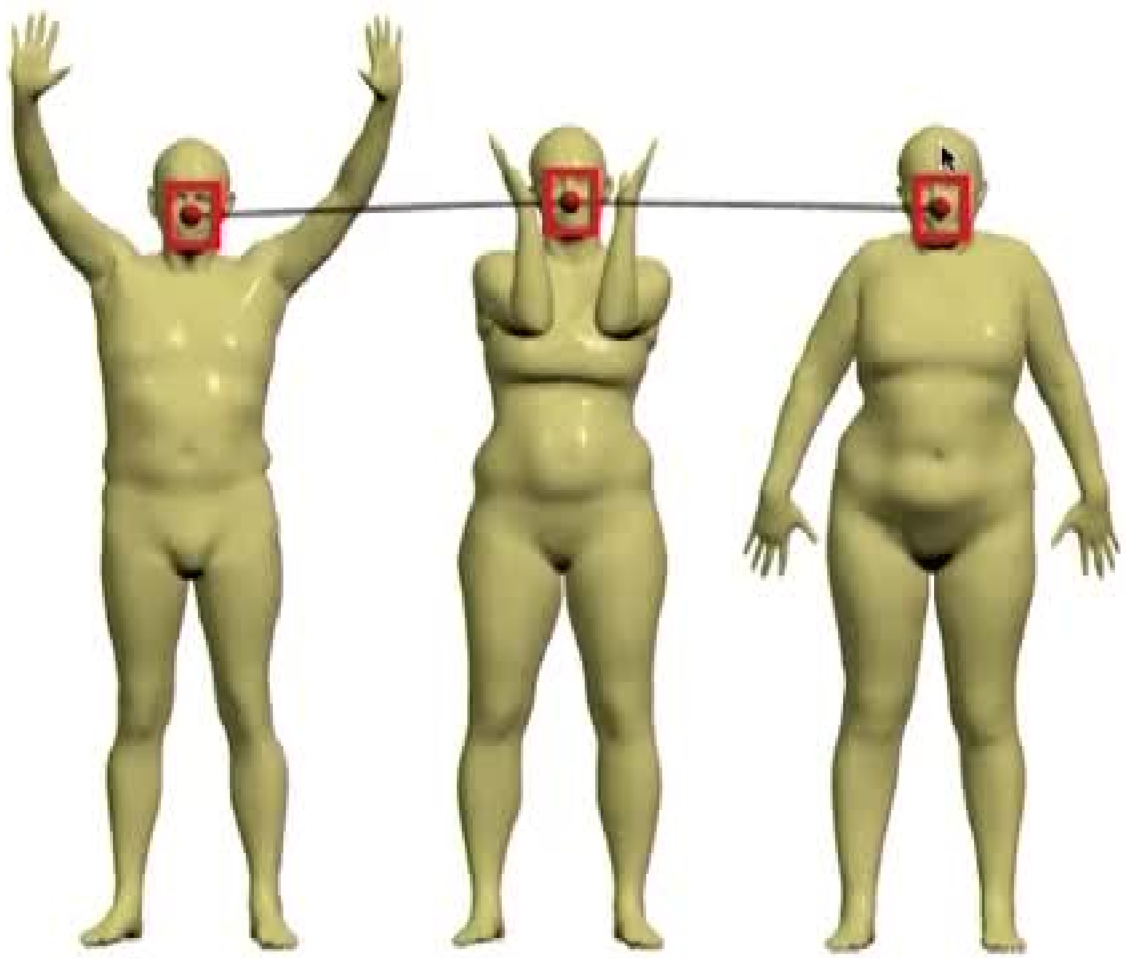
Spin image[7]

Heat kernel signature[8]

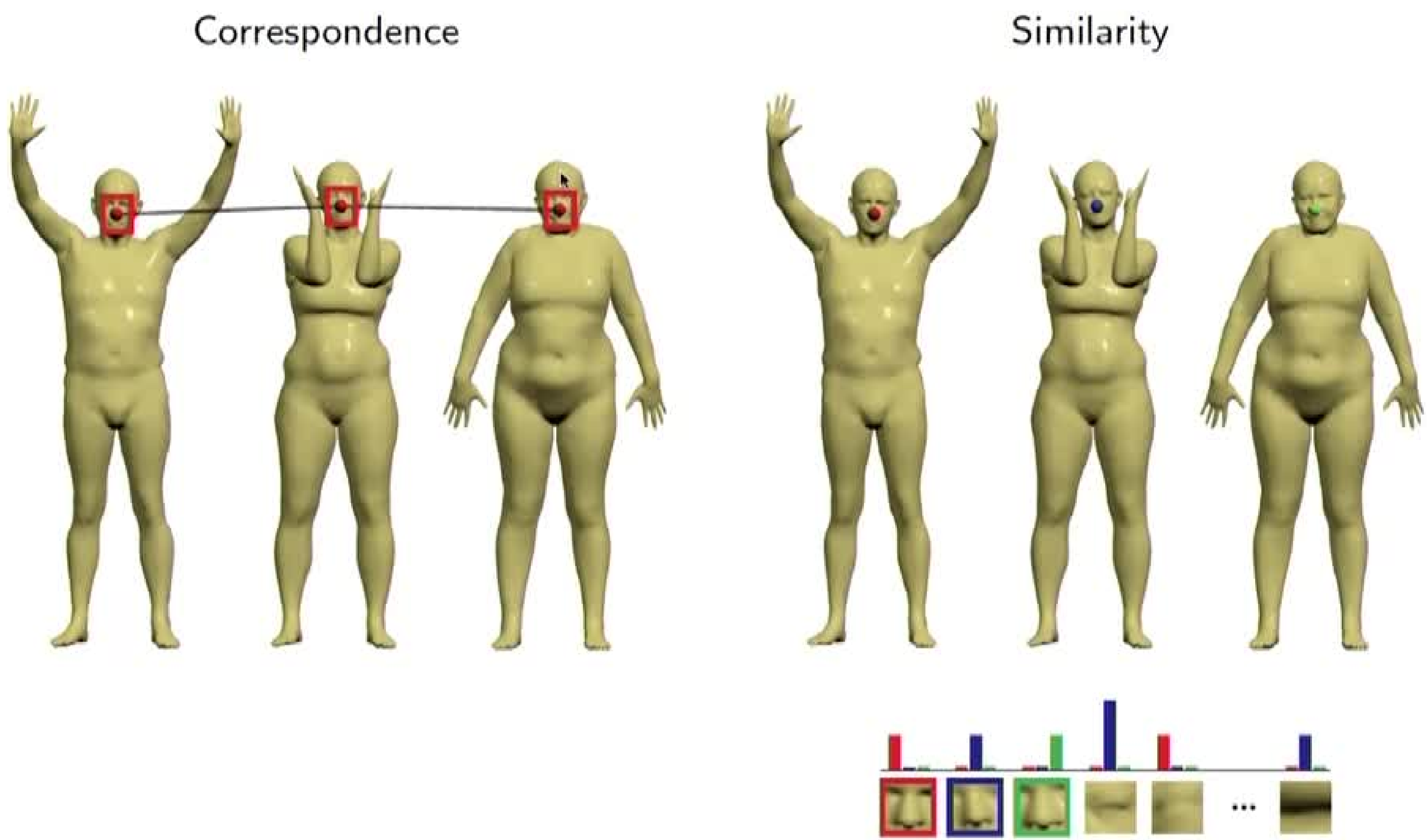
[1] Lowe 2004; [2] Zaharescu et al. 2009; [3] Matas et al. 2002; [4] Litman et al. 2010; [5] Belongie et al. 2000; [6] Kokkinos et al. 2012; [7] Johnson et al. 1999; [8] Sun et al. 2009

Task-specific features

Correspondence



Task-specific features



Shape representation

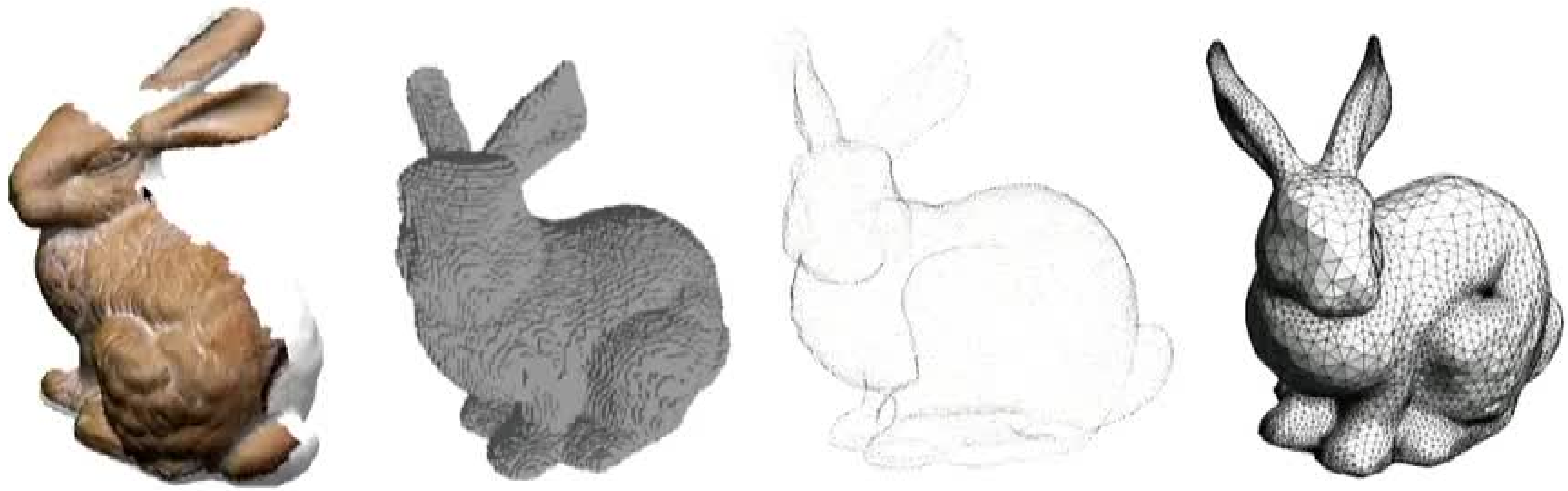
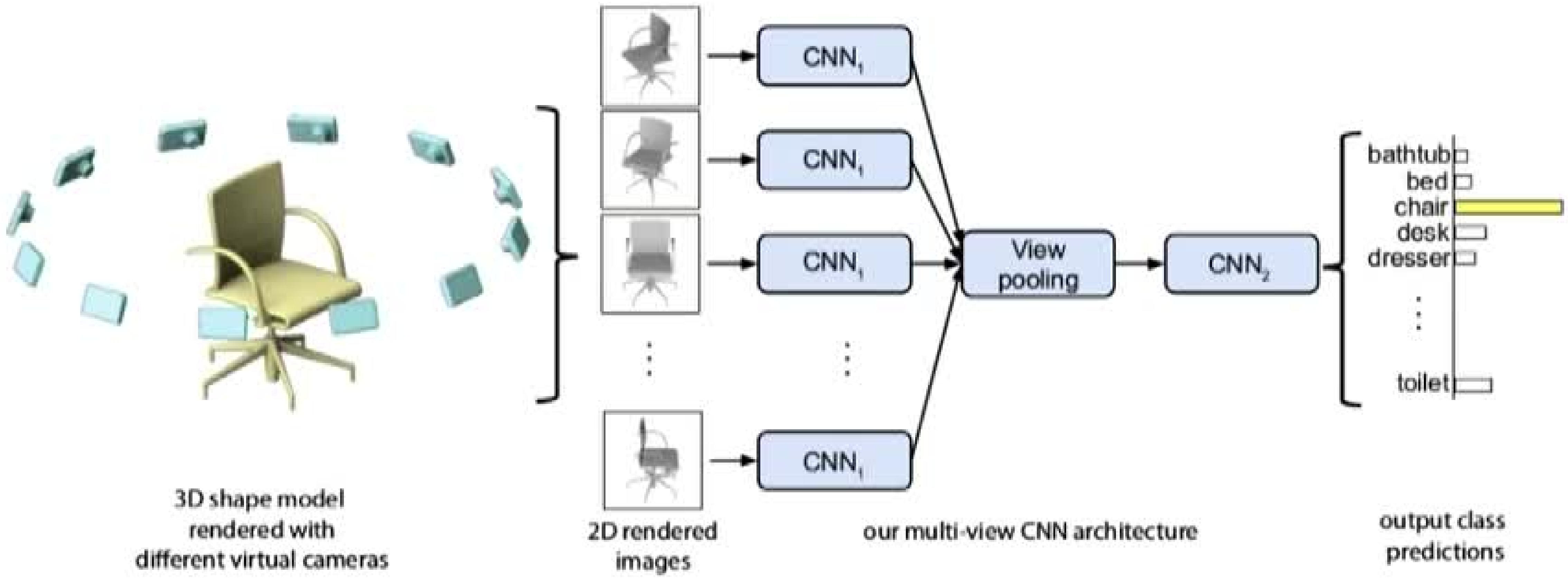


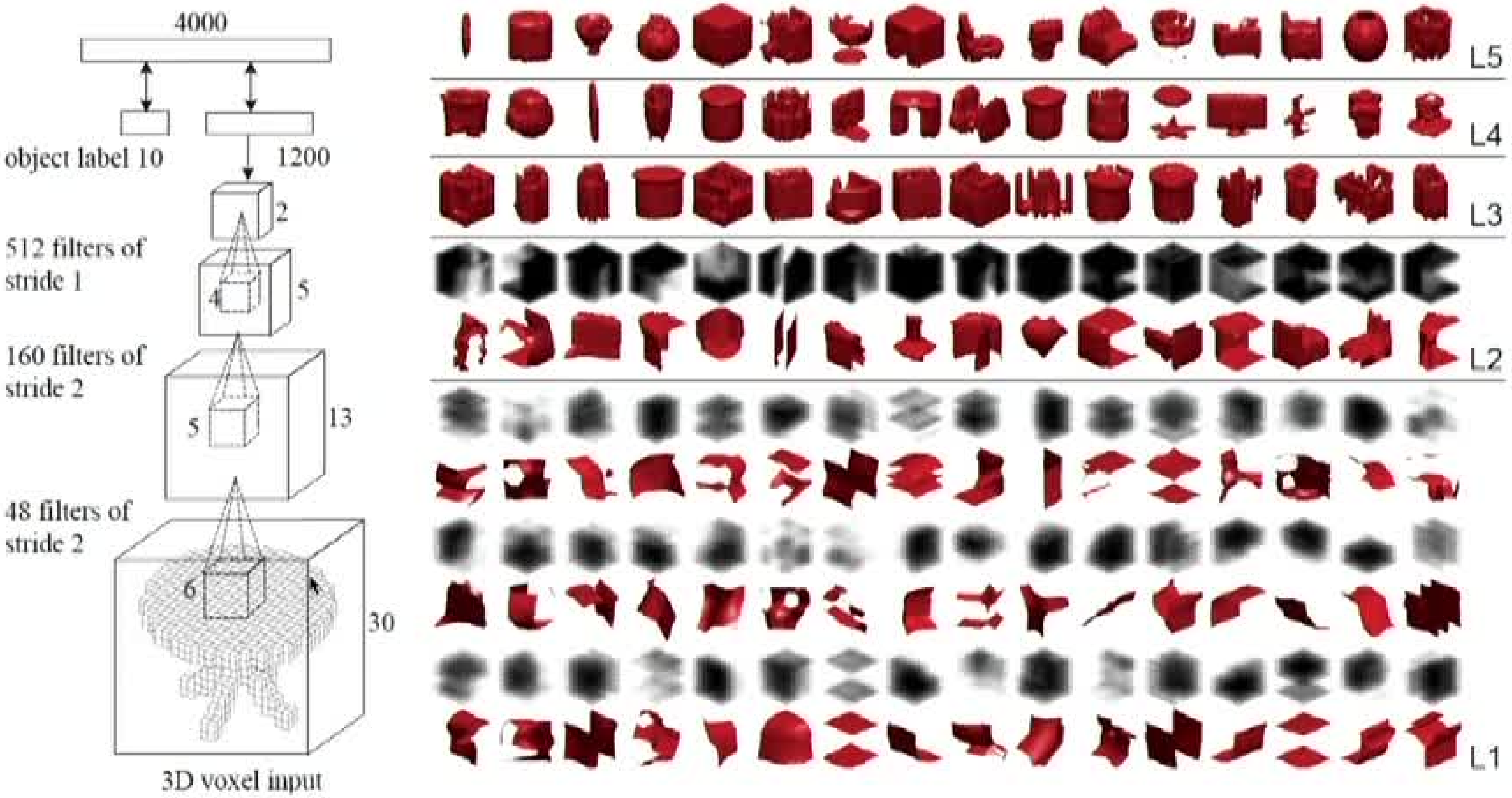
Image-based Volumetric Point-based Surface-based

Multi-view CNN



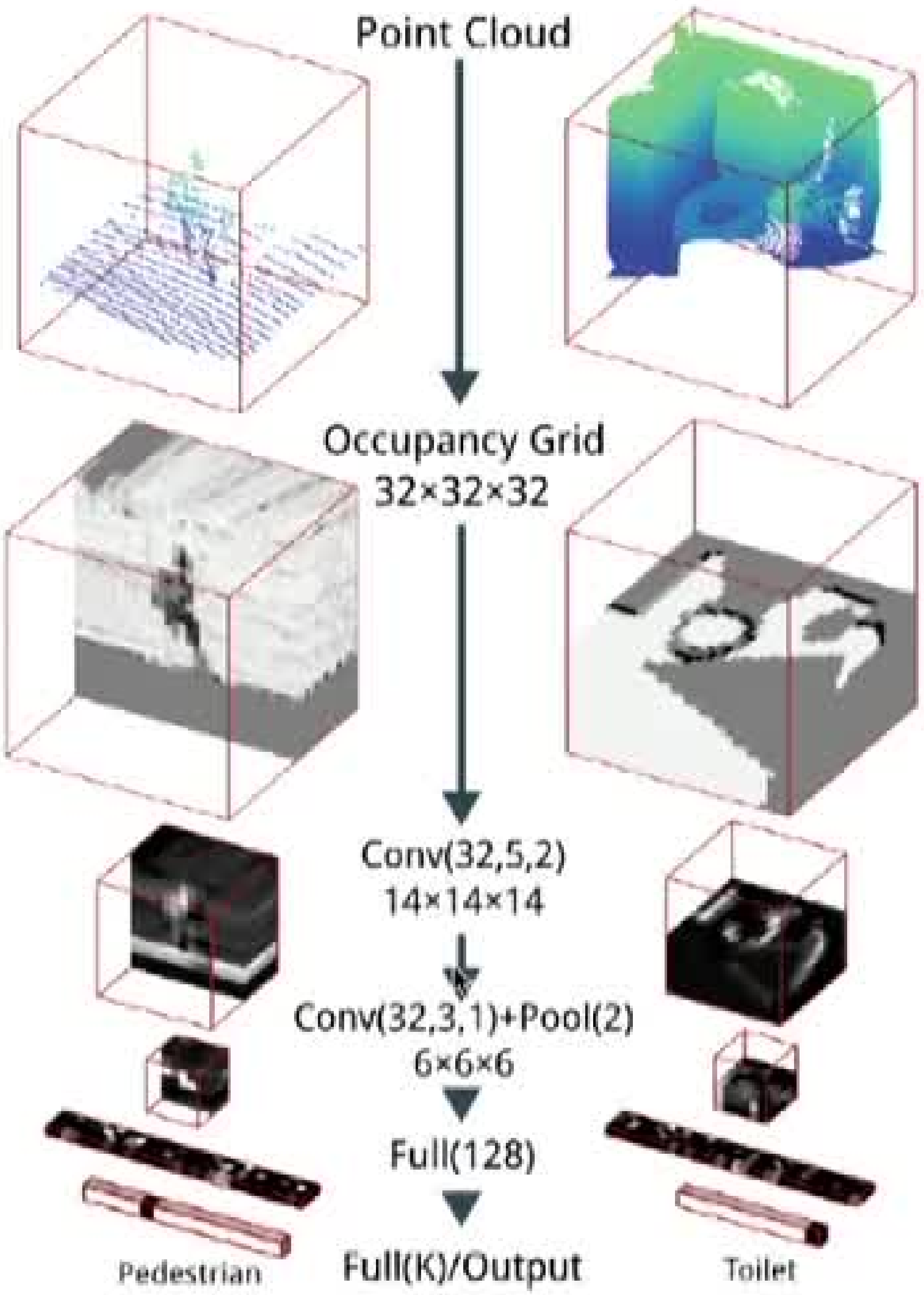
Su, Maji et al. 2015

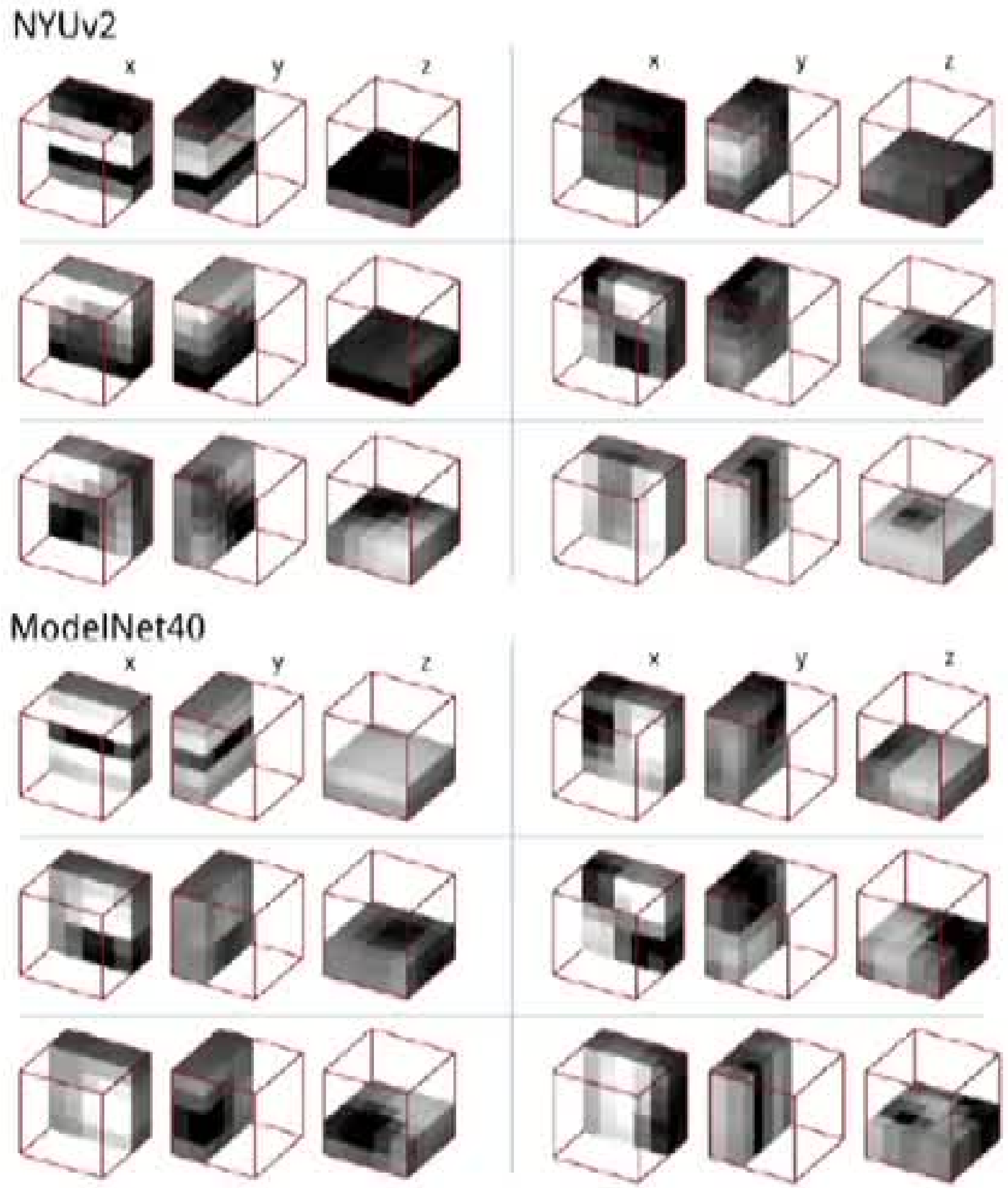
Volumetric methods: 3D ShapeNet



Wu et al. 2015

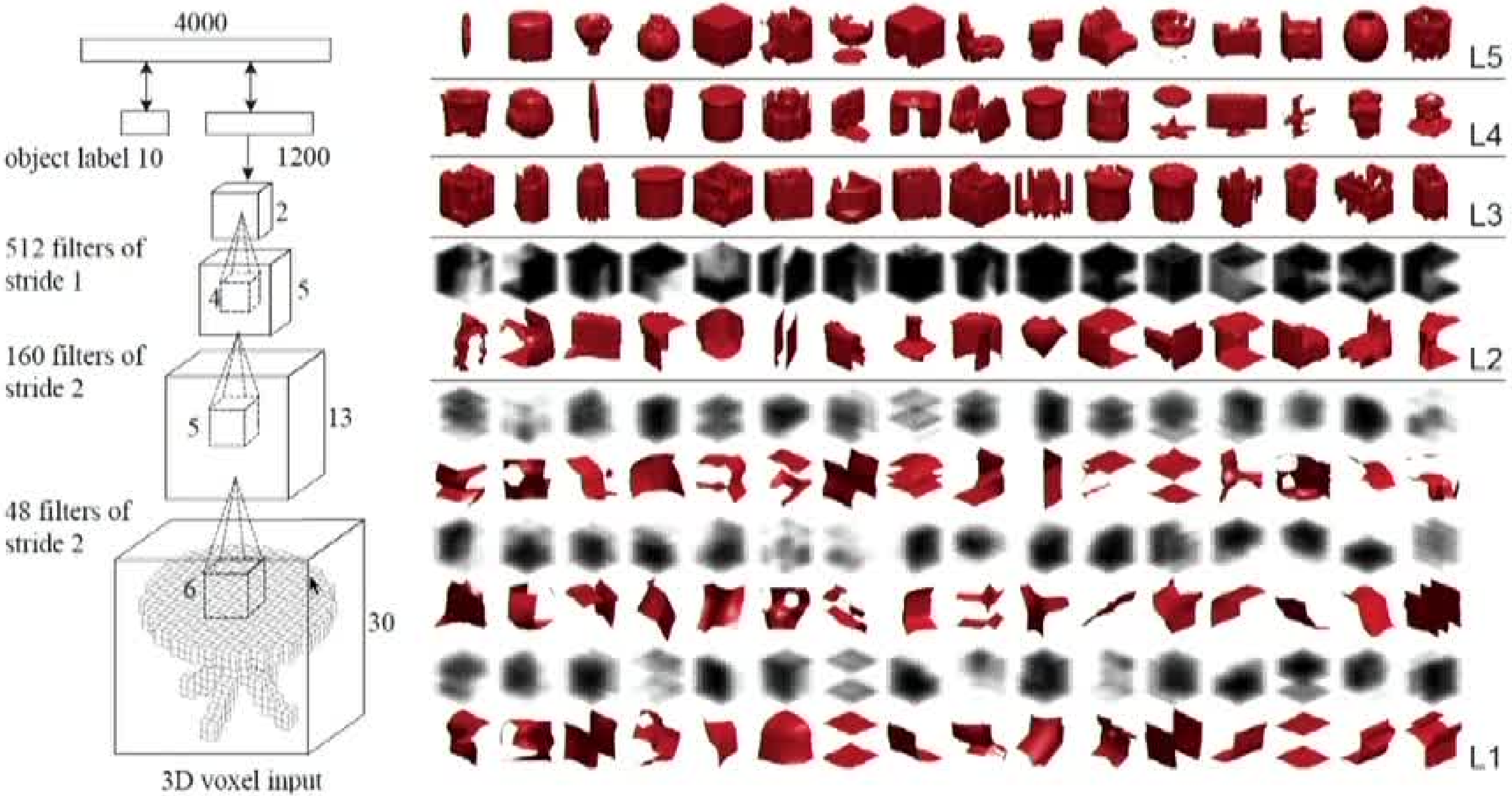
Volumetric methods: VoxNet





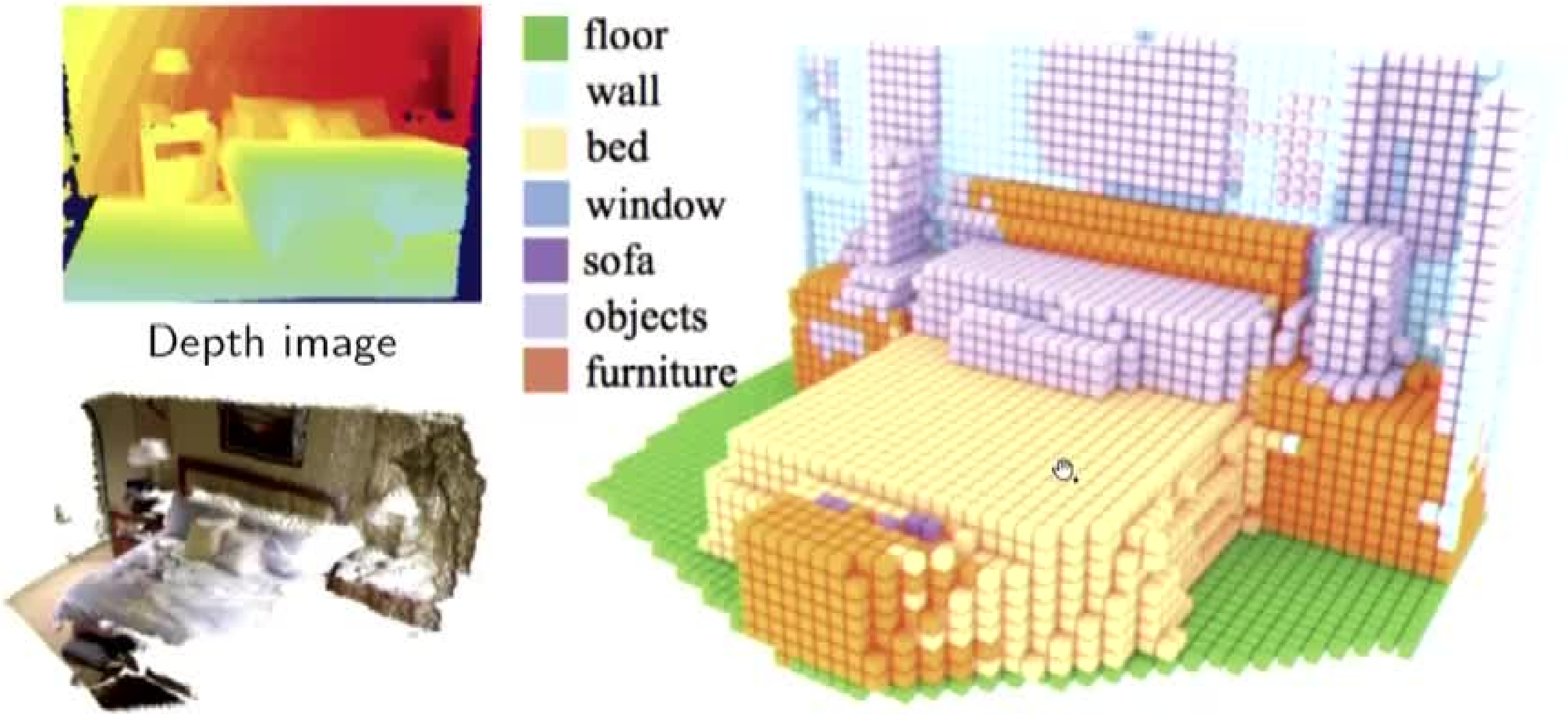
Maturana, Scherer 2015

Volumetric methods: 3D ShapeNet



Wu et al. 2015

Volumetric semantic segmentation



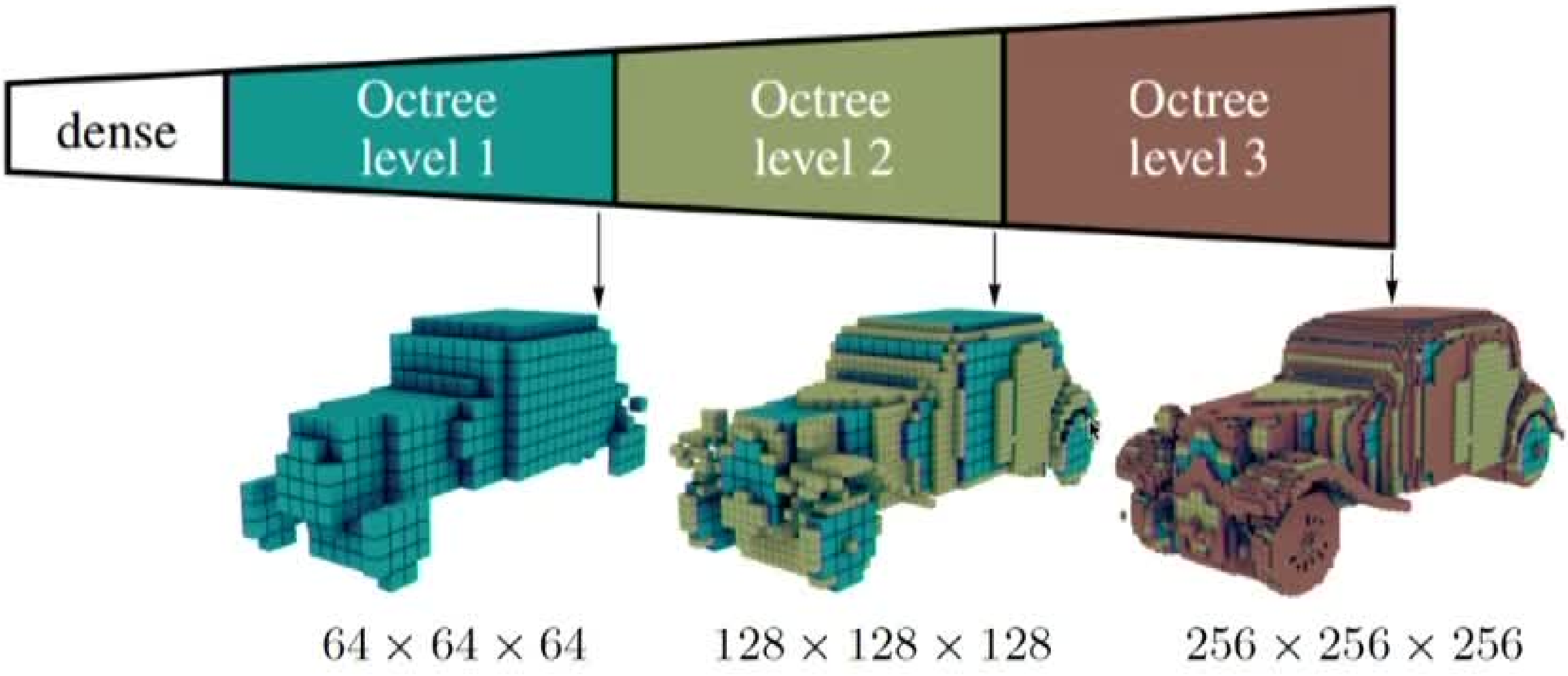
Depth image

Visible surface

Semantic segmentation

Song et al. 2017 (slide credit: Niloy Mitra)

Efficient volumetric data structures



Tatarchenko, Dosovitskiy, Brox 2017

PointNet: learning on sets

- Permutation-invariant function

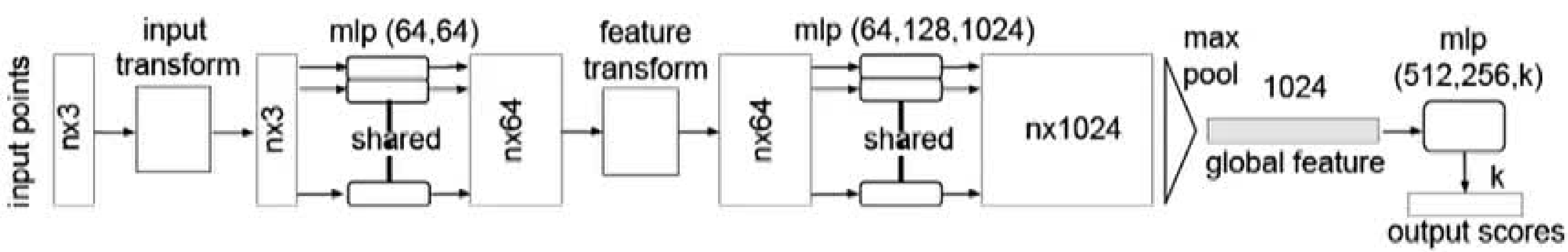
  $$f(\mathbf{x}_1, \ldots, \mathbf{x}_n) = f(\mathbf{x}_{\pi_1}, \ldots, \mathbf{x}_{\pi_n})$$

  where $\mathbf{x}_i \in \mathbb{R}^d$ is feature at vertex i

- Shared function $h_\Theta(\cdot)$ applied to each point + permutation-invariant aggregation ($\max$ or $\sum$)

- Spatial transformer units

- Local grouping (PointNet++, PCPNet)



(Vinyals, Bengio, Kudlur 2015); Qi et al. 2017; Qi, Yi et al. 2017; Guerrero et al. 2018

PointNet applications

Object recognition   Part segmentation   Semantic segmentation

Curvature   Normals   Point cloud generation

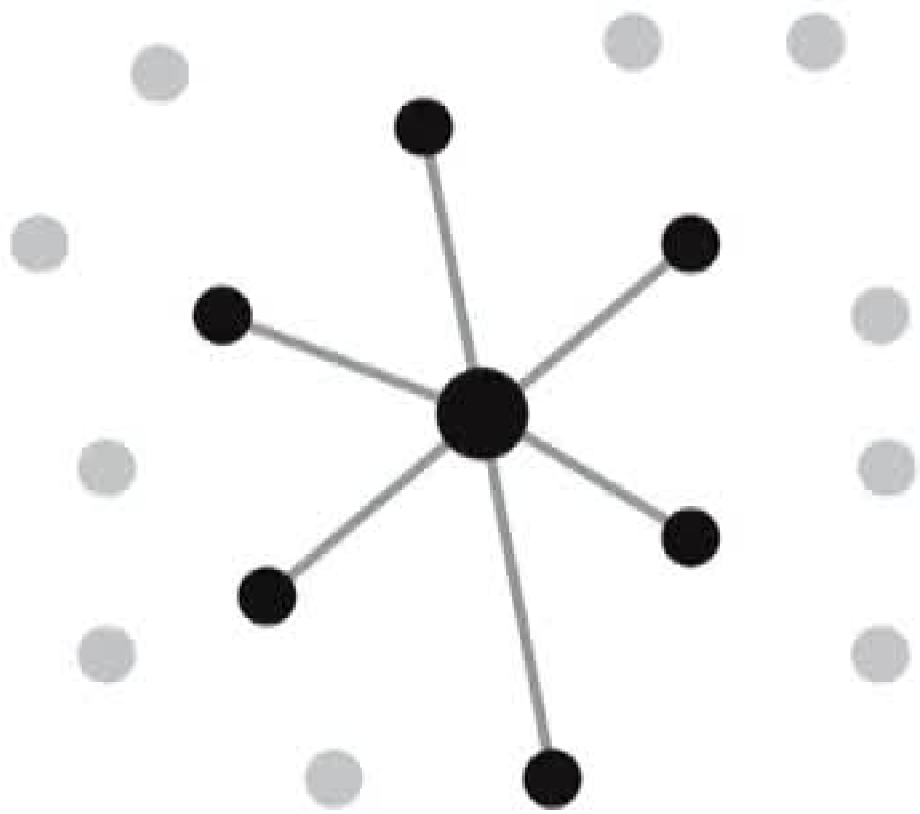
Particular cases

Method	**Aggregation** $\square$	**Edge feature** $h(\mathbf{x}_i, \mathbf{x}_j)$
Laplacian	$\sum$	$w_{ij}(\mathbf{x}_j - \mathbf{x}_i)$
PointNet[1]	–	$h(\mathbf{x}_i)$
PointNet++[2]	max	$h(\mathbf{x}_i)$
MoNet[3]	$\sum$	$\sum_\ell g_\ell w_\ell(\mathbf{u}_{ij})\mathbf{x}_j$
PCNN[4]	$\sum$	$\sum_{\ell m} c(\mathbf{x}_i \cdot \mathbf{k}_{\ell m}) w_i q_{\mathbf{\Theta}_\ell}(\mathbf{x}_i, \mathbf{x}_j)$

Wang et al. 2018; [1]Qi et al. 2017; [2]Qi, Su et al. 2017; [3]Monti et al. 2017; [4]Atzmon et al. 2018

Dynamic Graph CNN (DynGCNN)

Construct k-NN graph in feature space and update it after each layer

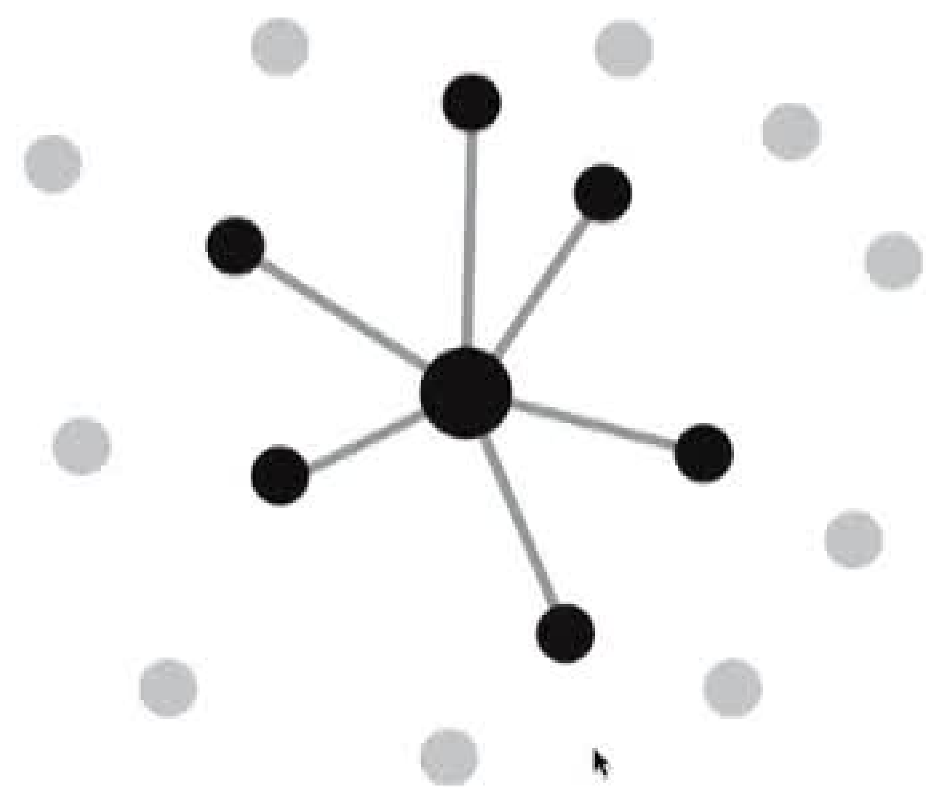


Layer l

Features $\mathbf{x}_1^{(l)}, \ldots, \mathbf{x}_n^{(l)} \in \mathbb{R}^{d_l}$

k-NN graph $\mathcal{G}^{(l)}$

$h^{(l)} : \mathbb{R}^{d_l} \times \mathbb{R}^{d_l} \to \mathbb{R}^{d_{l+1}}$



Layer $l+1$

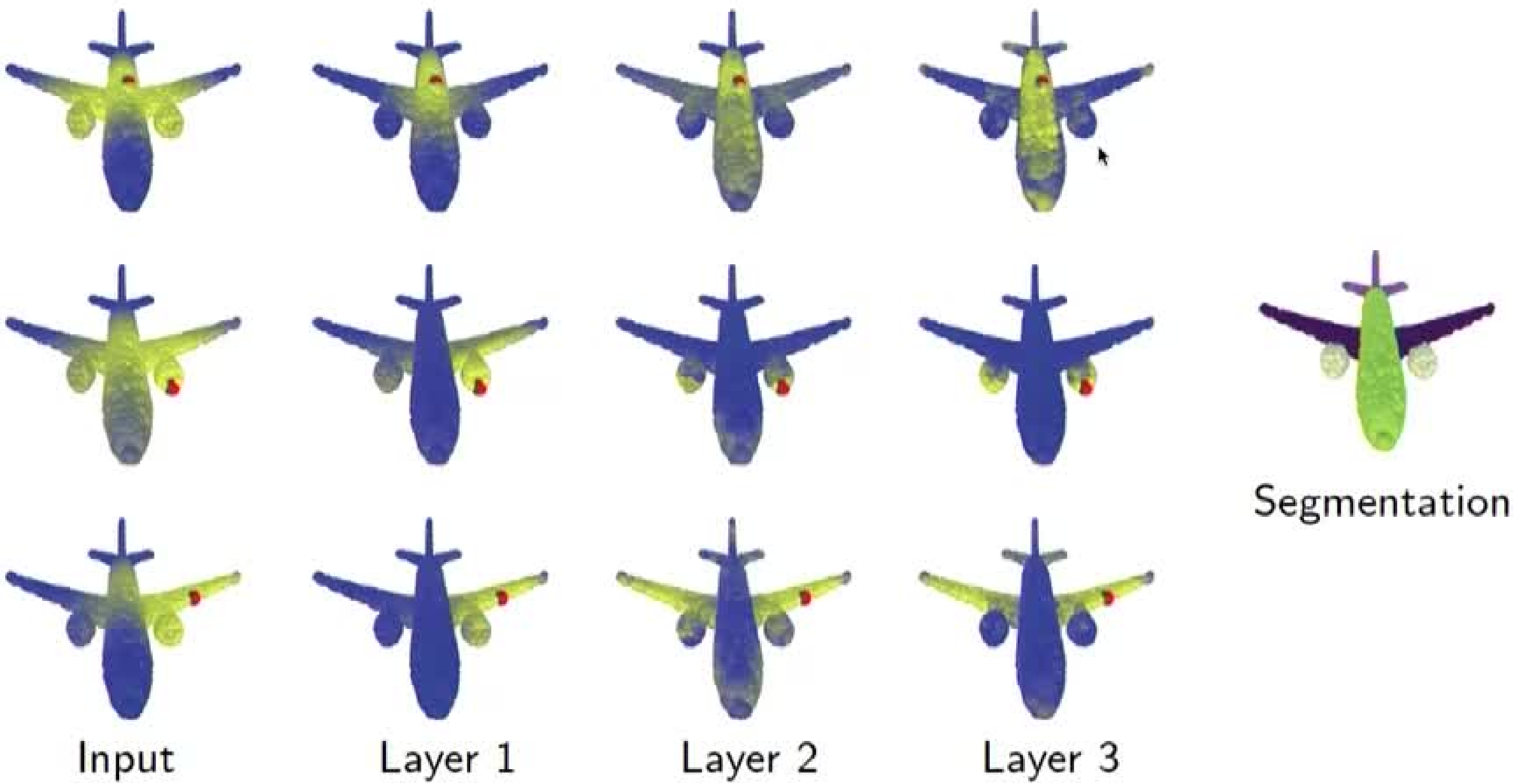
Features $\mathbf{x}_1^{(l+1)}, \ldots, \mathbf{x}_n^{(l+1)} \in \mathbb{R}^{d_{l+1}}$

k-NN graph $\mathcal{G}^{(l+1)}$

$h^{(l+1)} : \mathbb{R}^{d_{l+1}} \times \mathbb{R}^{d_{l+1}} \to \mathbb{R}^{d_{l+2}}$

Wang et al. 2018

Learning semantic features



Left: Distance from red point in the feature space of different DynGCNN layers
Right: semantic segmentation results

Wang et al. 2018

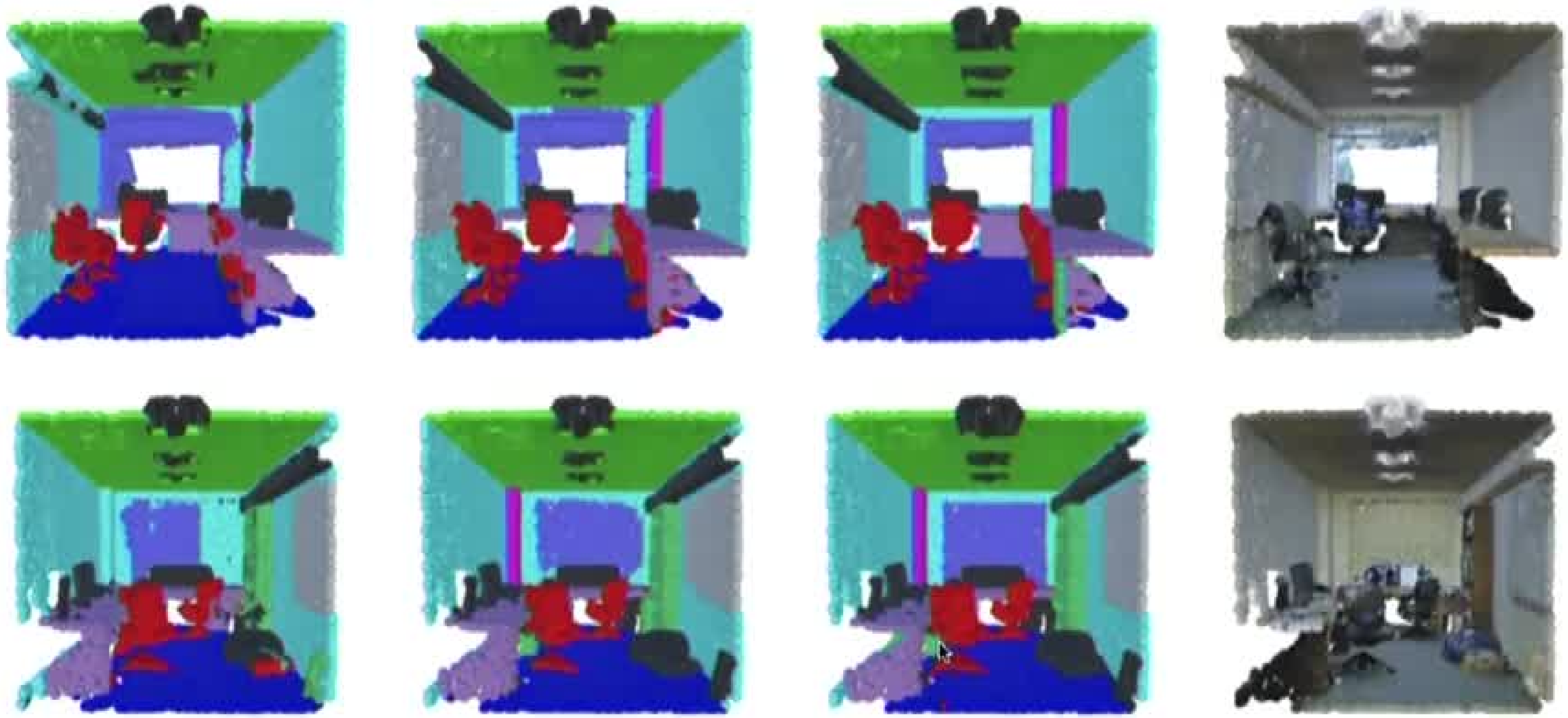
Shape classification (ModelNet40)

Method	Mean class accuracy	Overall accuracy
3DShapeNet[1]	77.3%	84.7%
VoxNet[2]	83.0%	85.9%
Subvolume[3]	86.0%	89.2%
ECC[4]	83.2%	87.4%
PointNet[5]	86.0%	89.2%
PointNet++[6]	–	90.7%
Kd-Net[7]	–	91.8%
DynGCNN (baseline) [8]	88.8%	91.2%
DynGCNN[8]	**90.2%**	**92.2%**

Classification accuracy of different methods on ModelNet40

Methods: [1]Wu et al. 2015; [2]Maturana et al. 2015; Qi et al. 2016; [4]Simonovsky, Komodakis 2017; [5]Qi et al. 2017; [6]Qi, Su et al. 2017; [7]Klokov, Lempitsky 2017; [8]Wang et al. 2018; data: Wu et al. 2015 (ModelNet)

Semantic segmentation: indoor scans (S3DIS)

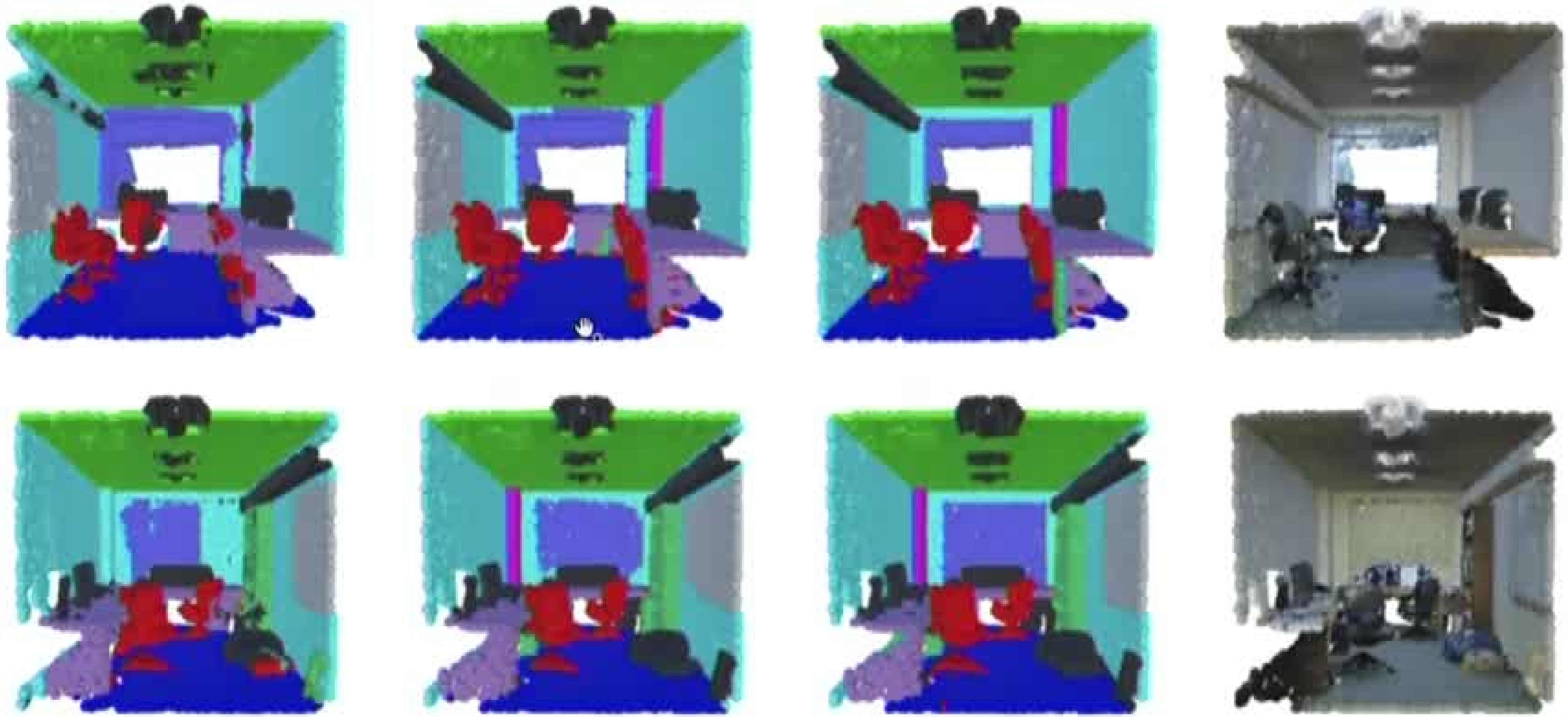


PointNet **DynGCNN** Groundtruth Scene

Results of semantic segmentation of point cloud+RGB data using different architectures

Methods: Qi et al. 2017 (PointNet); Wang et al. 2018 (DynGCNN); data: Armeni et al. 2016 (S3DIS)

Semantic segmentation: indoor scans (S3DIS)



PointNet | **DynGCNN** | Groundtruth | Scene

Results of semantic segmentation of point cloud+RGB data using different architectures

Methods: Qi et al. 2017 (PointNet); Wang et al. 2018 (DynGCNN); data: Armeni et al. 2016 (S3DIS)

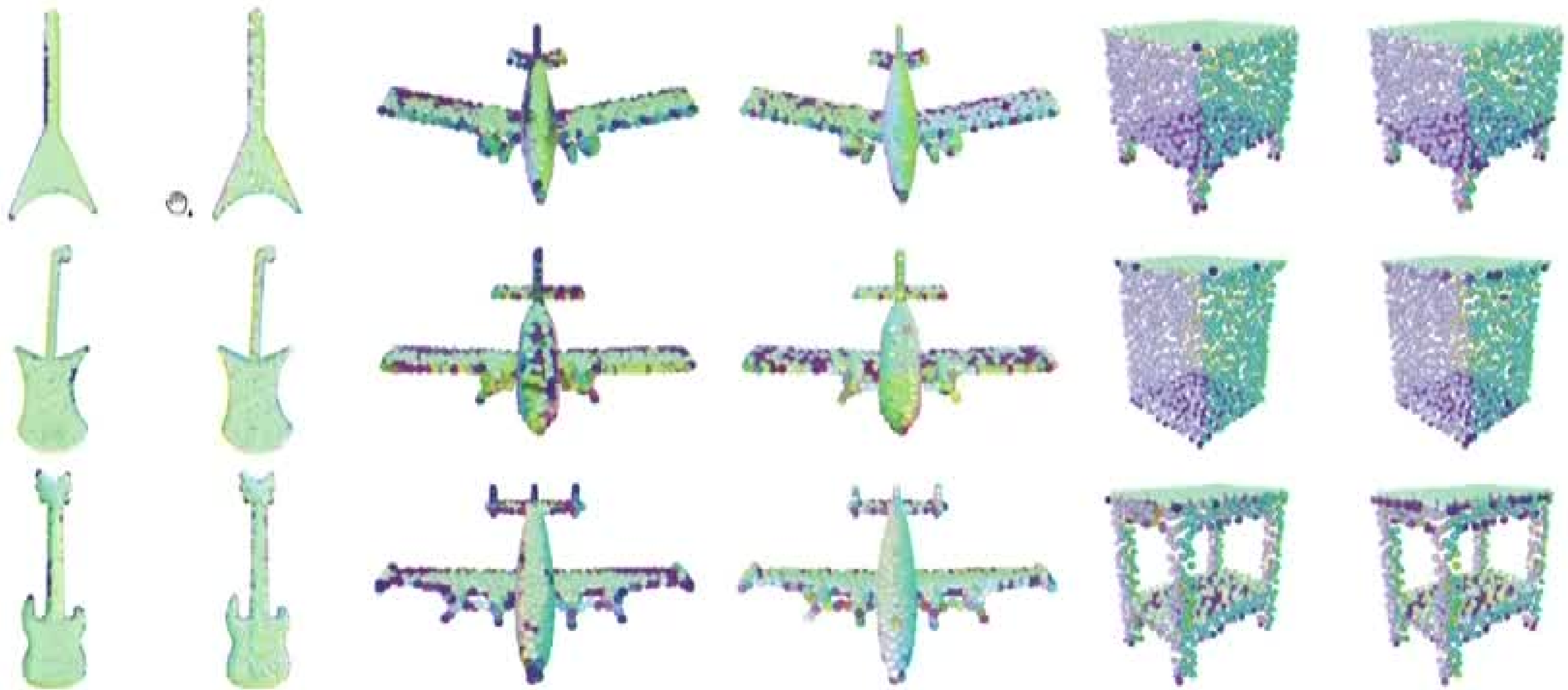
Shape segmentation: indoor scans (S3DIS)

Method	Mean IoU	Overall accuracy
PointNet (Baseline)[1]	20.1%	53.2%
PointNet[1]	47.6%	78.5%
MS + CU(2)[2]	47.8%	79.2%
G + RCU[2]	49.7%	81.1%
DynGCNN[3]	**56.1%**	**84.1%**

S3DIS indoor scene semantic segmentation accuracy

Methods: [1]Qi et al. 2017; [2]Engelmann et al. 2017 [3]Wang et al. 2018; data: Armeni et al. 2016 (S3DIS)

Surface normal prediction



Surface normal predicted using DynGCNN (odd columns) and groundtruth (even columns). Normal direction is color-coded

Wang et al. 2018; data: Wu et al. 2015 (ModelNet)

Shape representation

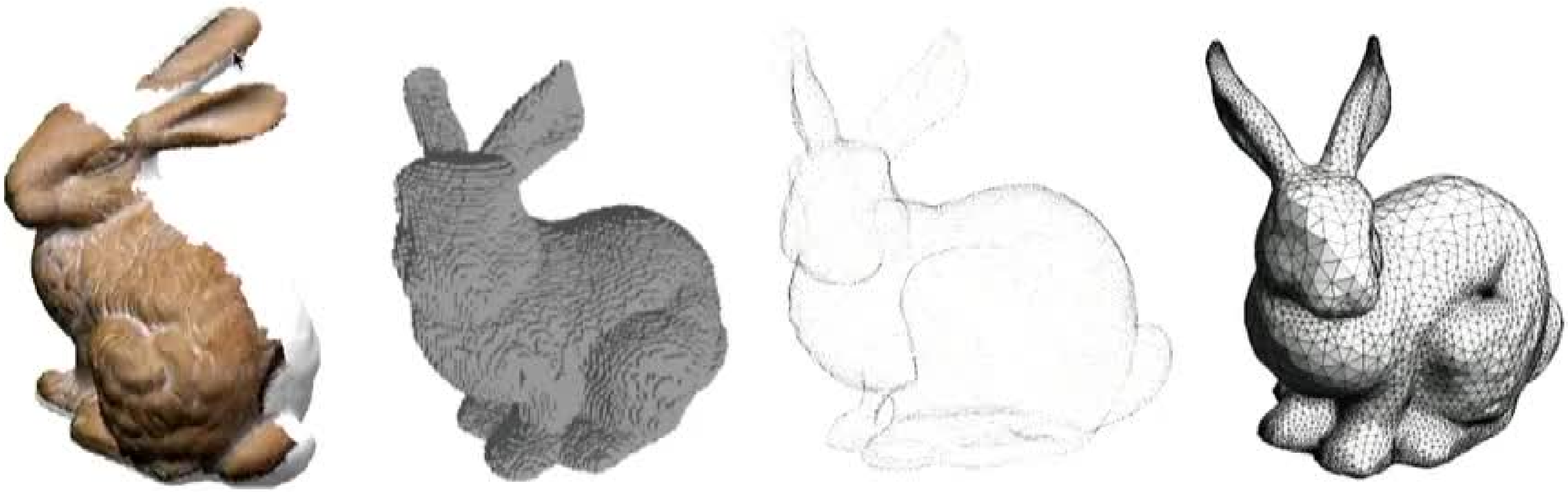
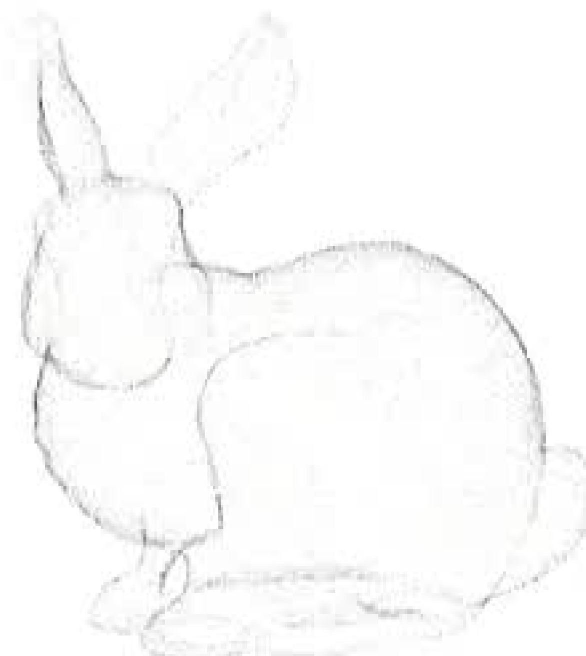


Image-based　　Volumetric　　Point-based　　Surface-based

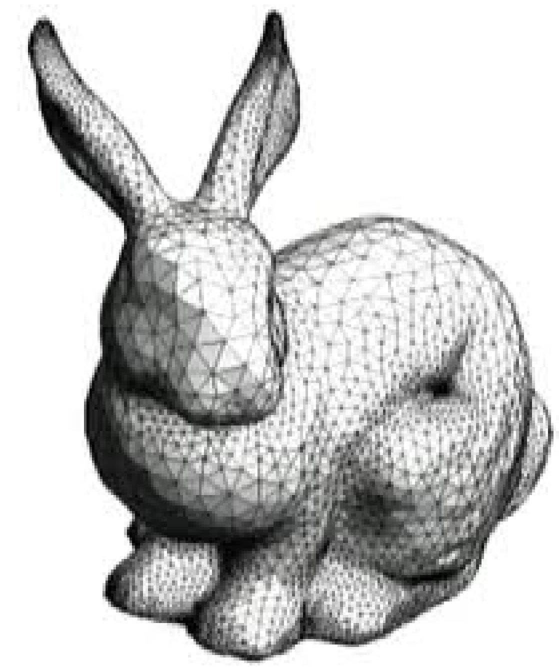
Shape representation

Image-based

Volumetric

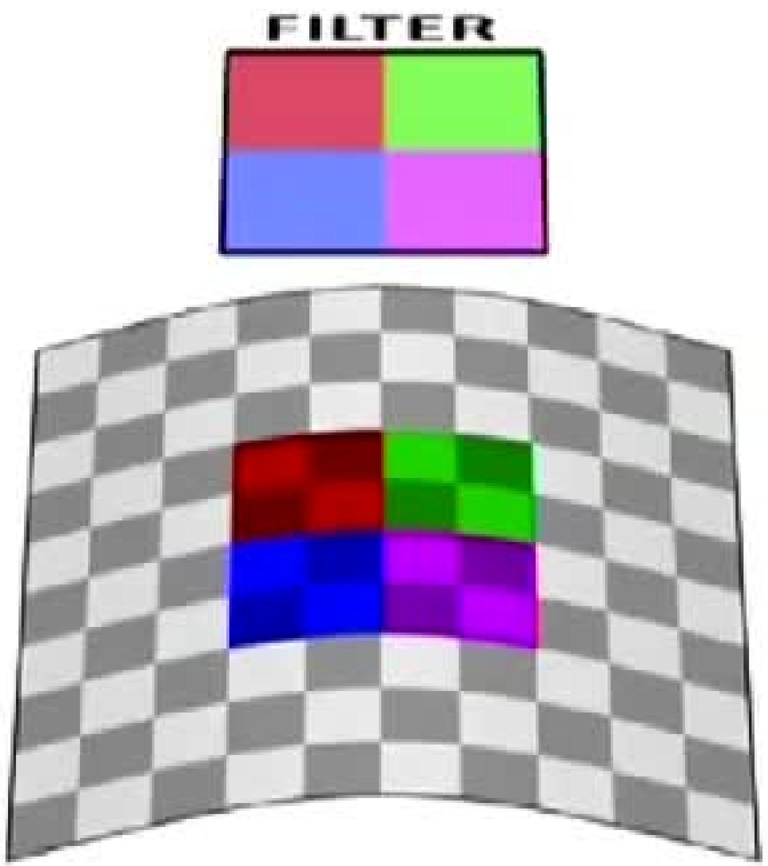


Point-based

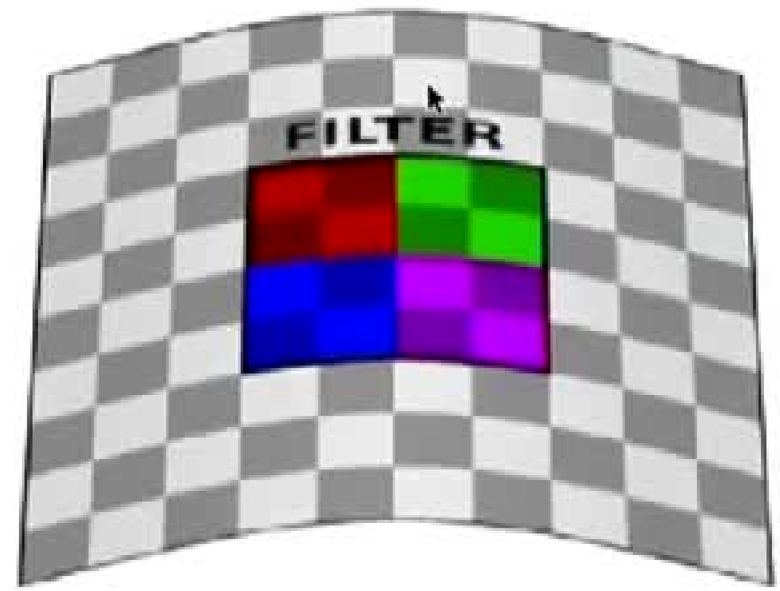


Surface-based

Extrinsic vs Intrinsic CNNs



Extrinsic

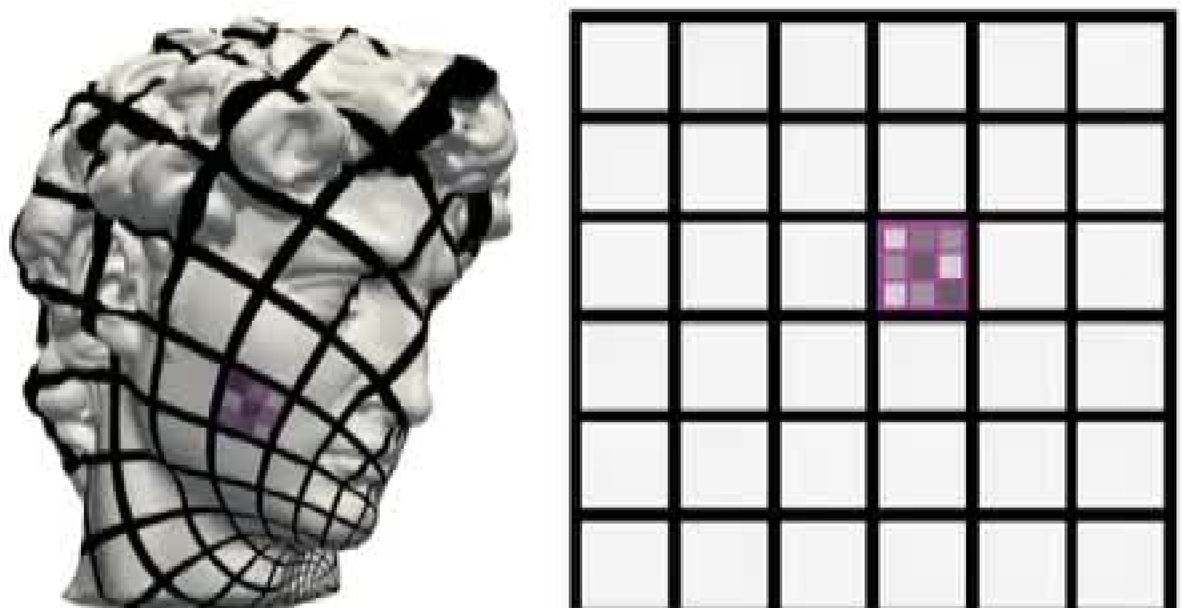


Intrinsic

Different formulations of non-Euclidean CNNs

Spectral domain

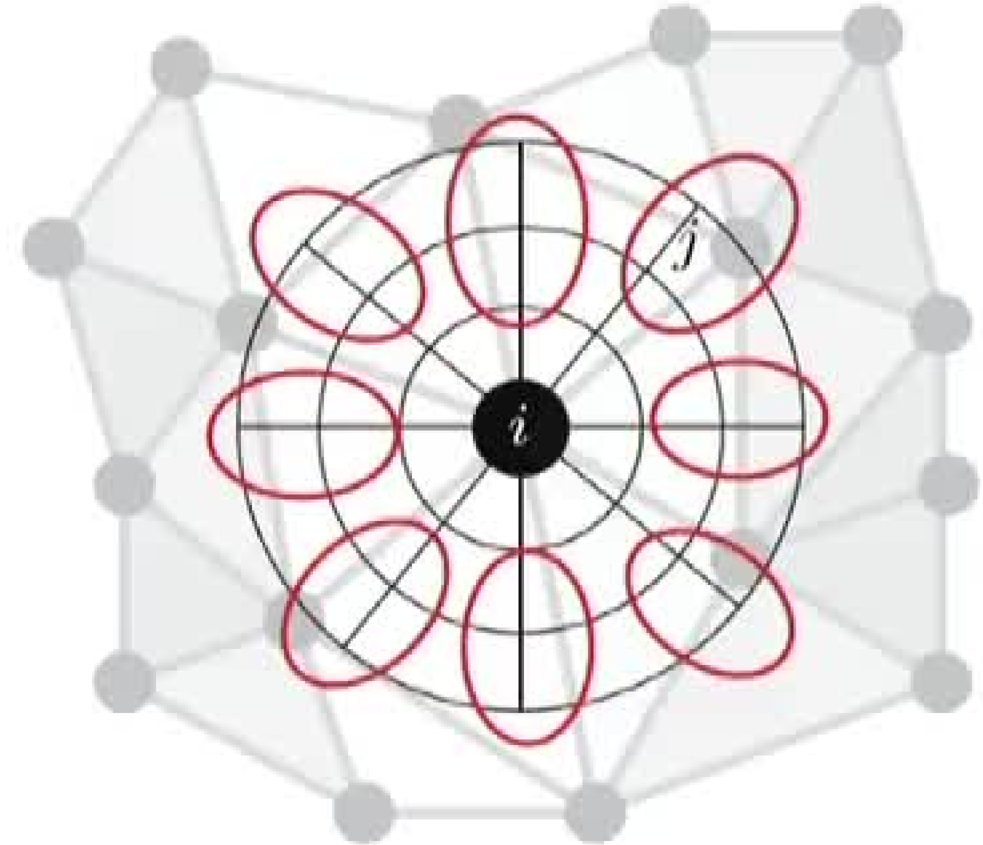
Spatial domain



Parametric domain

Convolution on meshes

- Local system of coordinates $\mathbf{u}_{ij}$ around i (e.g. geodesic polar)
- Local weights $w_1(\mathbf{u}), \ldots, w_L(\mathbf{u})$ w.r.t. $\mathbf{u}$, e.g. Gaussians
  $$w_\ell = \exp\left(-(\mathbf{u} - \boldsymbol{\mu}_\ell)^\top \boldsymbol{\Sigma}_\ell^{-1}(\mathbf{u} - \boldsymbol{\mu}_\ell)\right)$$

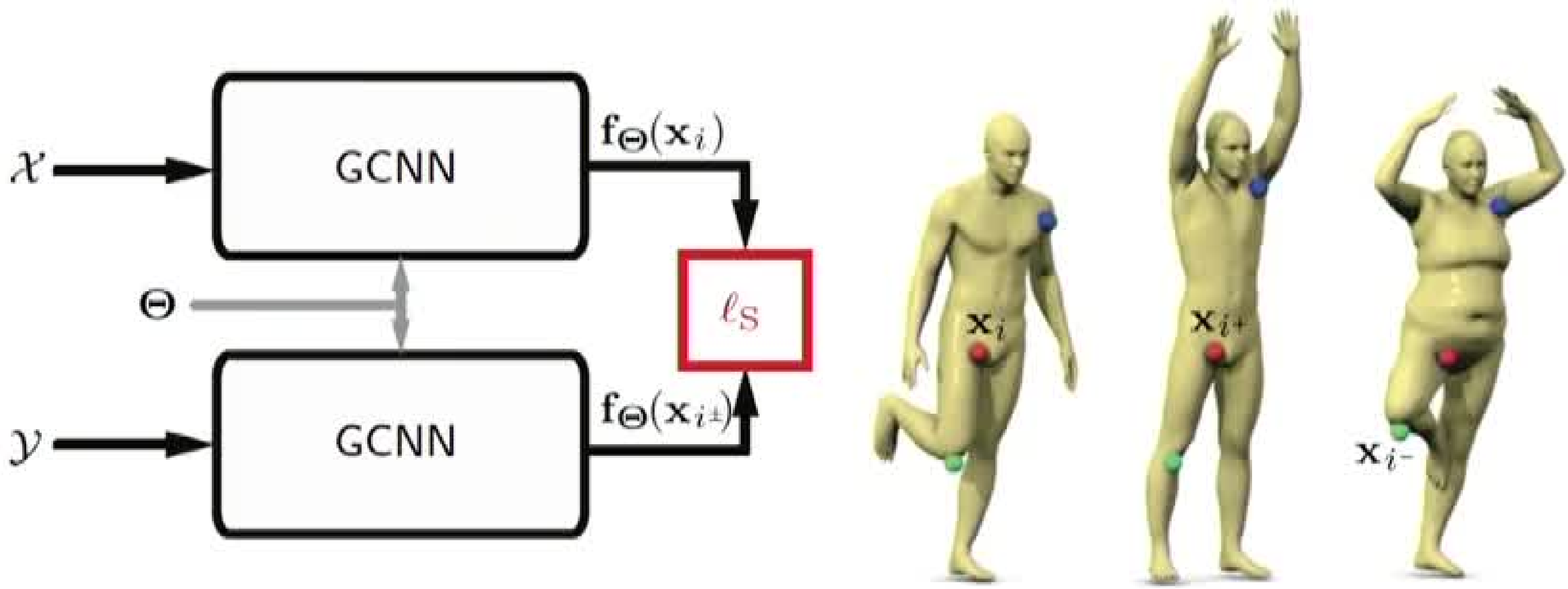


- Spatial convolution with filter g
  $$\mathbf{x}'_i \propto \sum_{\ell=1}^{L} g_\ell \underbrace{\sum_{j=1}^{n} w_\ell(\mathbf{u}_{ij})\mathbf{x}_j}_{\text{patch operator}}$$

  where $\mathbf{x}_i \in \mathbb{R}^d$ is feature at vertex i

Masci et al. 2015; Boscaini et al. 2016; Monti et al. 2017

Learning local descriptors with intrinsic CNNa



Training set	positive (i, i^+) and negative (i, i^-) pairs of points
Siamese net	two net instances with shared parameters $\boldsymbol{\Theta}$
Poitwise feature cost	$\ell_{\mathrm{S}}(\boldsymbol{\Theta}) = \gamma \sum_{i,i^+} \|\mathbf{f}_{\boldsymbol{\Theta}}(\mathbf{x}_i) - \mathbf{f}_{\boldsymbol{\Theta}}(\mathbf{x}_{i^+})\|_2^2 + (1-\gamma) \sum_{i,i^-} [\mu - \|\mathbf{f}_{\boldsymbol{\Theta}}(\mathbf{x}_i) - \mathbf{f}_{\boldsymbol{\Theta}}(\mathbf{x}_{i^-})\|_2^2]_+$

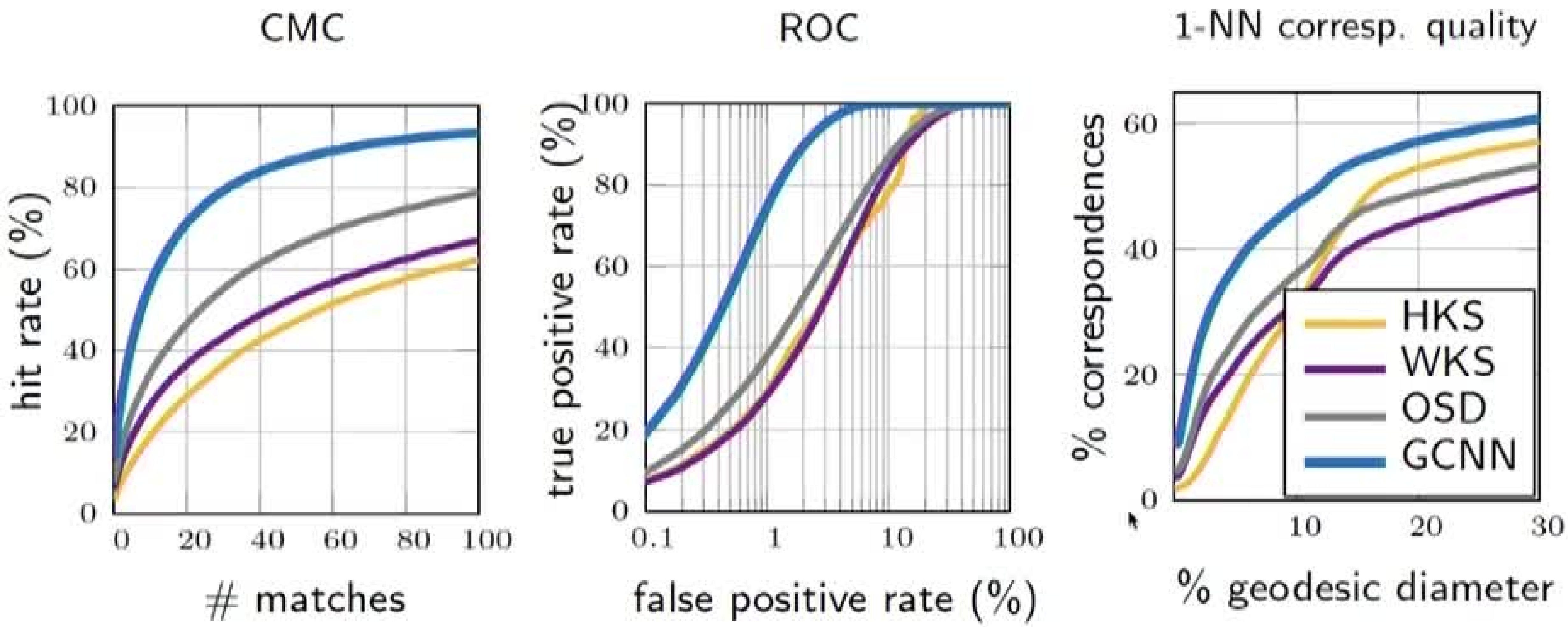
Masci et al. 2015

Descriptor learned with GCNN

Distance in the space of local GCNN features
(shown is distance from a point on the shoulder marked in white)

Descriptor: Masci et al. 2015 (GCNN); data: Bronstein, Bronstein, Kimmel 2008 (TOSCA); Anguelov et al. 2005 (SCAPE); Bogo et al. 2014 (FAUST)

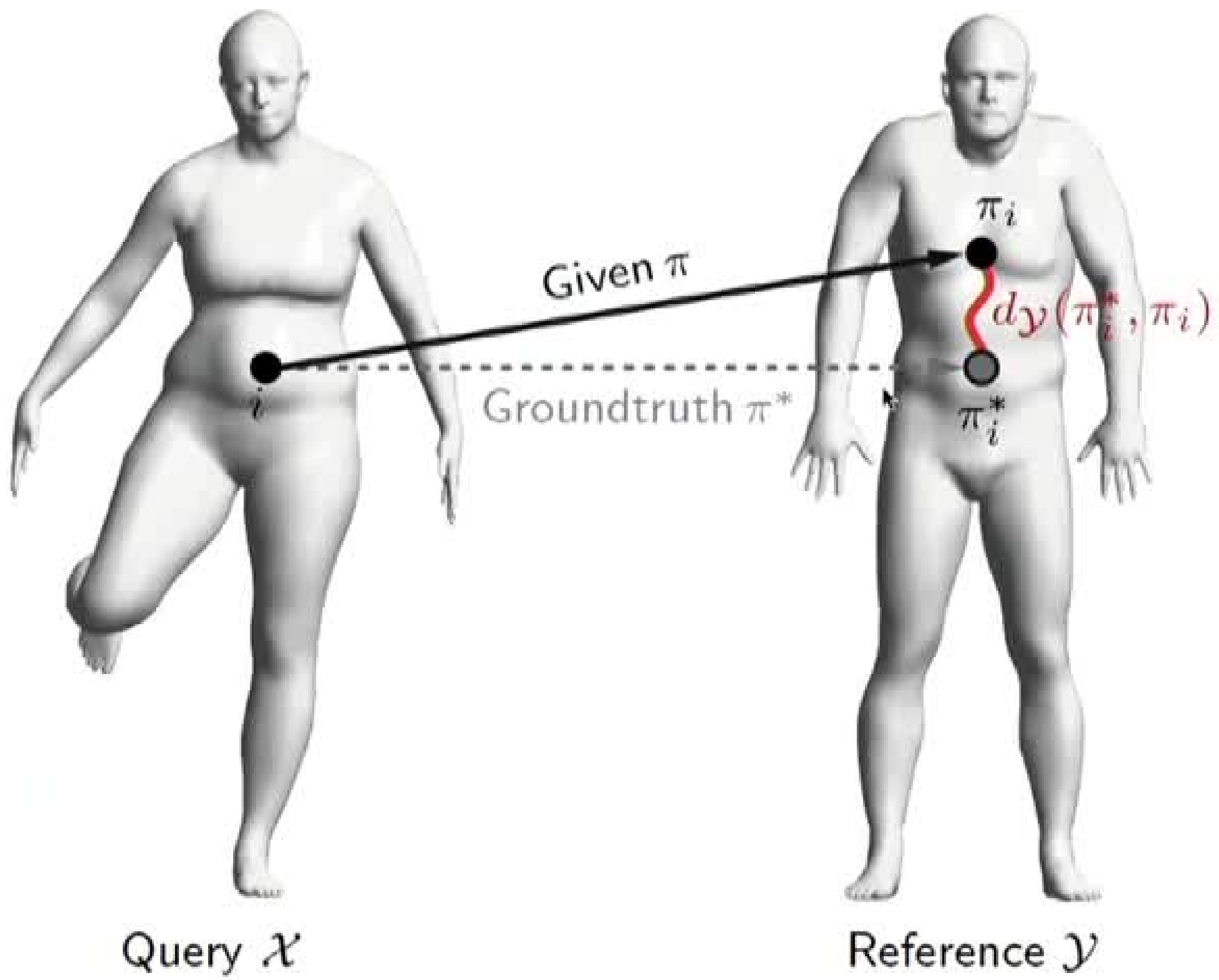
Descriptor quality comparison



Descriptor performance using symmetric Princeton benchmark
(training and testing: disjoint subsets of FAUST)

Methods: Sun et al. 2009 (HKS); Aubry et al. 2011 (WKS); Litman, B 2014 (OSD); Masci et al. 2015 (GCNN); data: Bogo et al. 2014 (FAUST); benchmark: Kim et al. 2011

Correspondence evaluation: Princeton benchmark

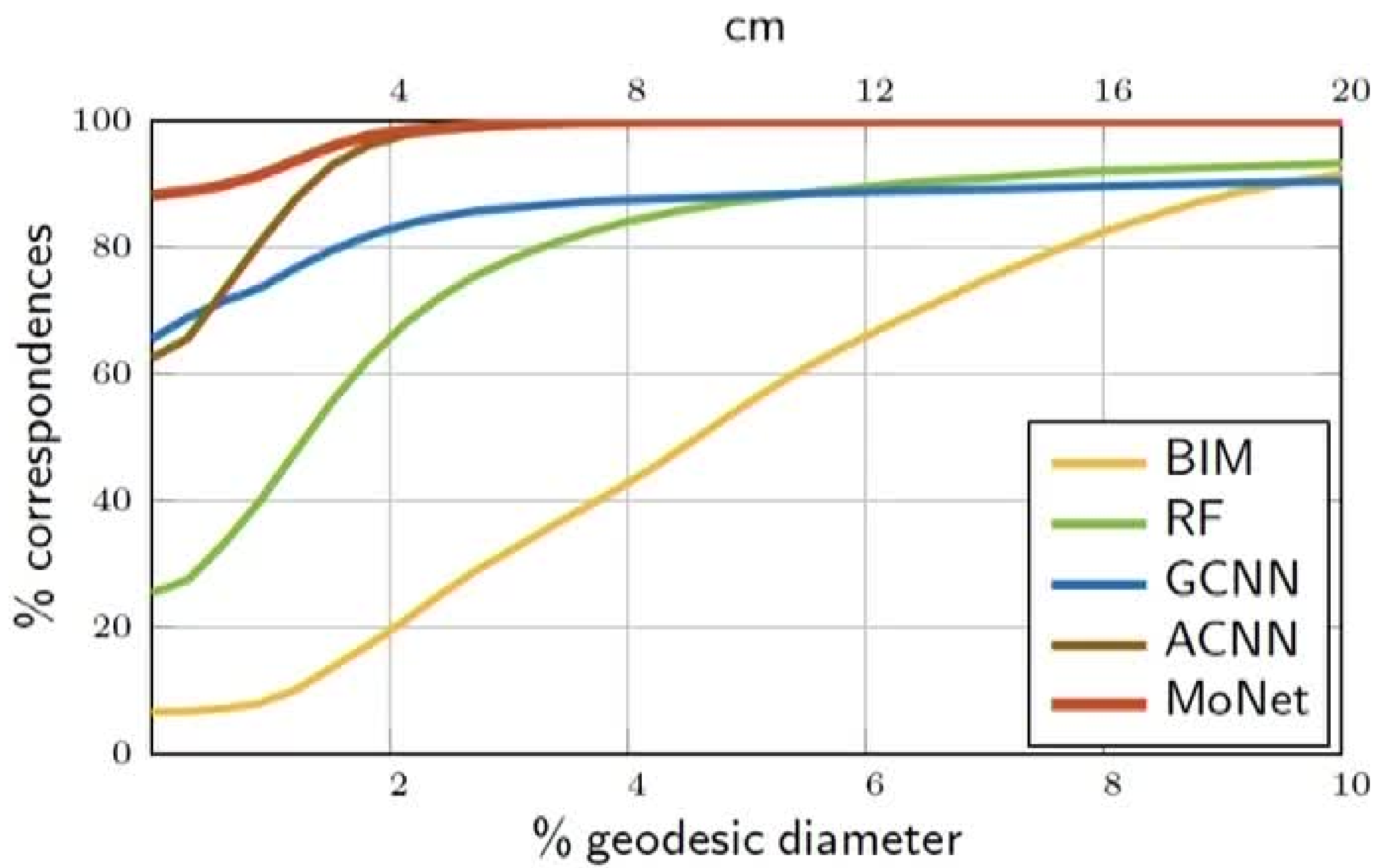


Pointwise correspondence error = geodesic distance from the groundtruth

$$\epsilon_i = d_{\mathcal{Y}}(\pi_i^*, \pi_i)$$

Kim et al. 2011

Correspondence quality comparison



Correspondence evaluated using asymmetric Princeton benchmark
(training and testing: disjoint subsets of FAUST)

Methods: Kim et al. 2011 (BIM); Rodolà et al. 2014 (RF); Boscaini et al. 2015 (ADD); Masci et al. 2015 (GCNN); Boscaini et al. 2016 (ACNN); Monti et al. 2016 (MoNet); data: Bogo et al. 2014 (FAUST); benchmark: Kim et al. 2011

Shape correspondence error: MoNet

Pointwise correspondence error (geodesic distance from groundtruth)

Monti et al. 2016

Shape correspondence visualization: MoNet

Reference

Texture transferred from reference to query shapes

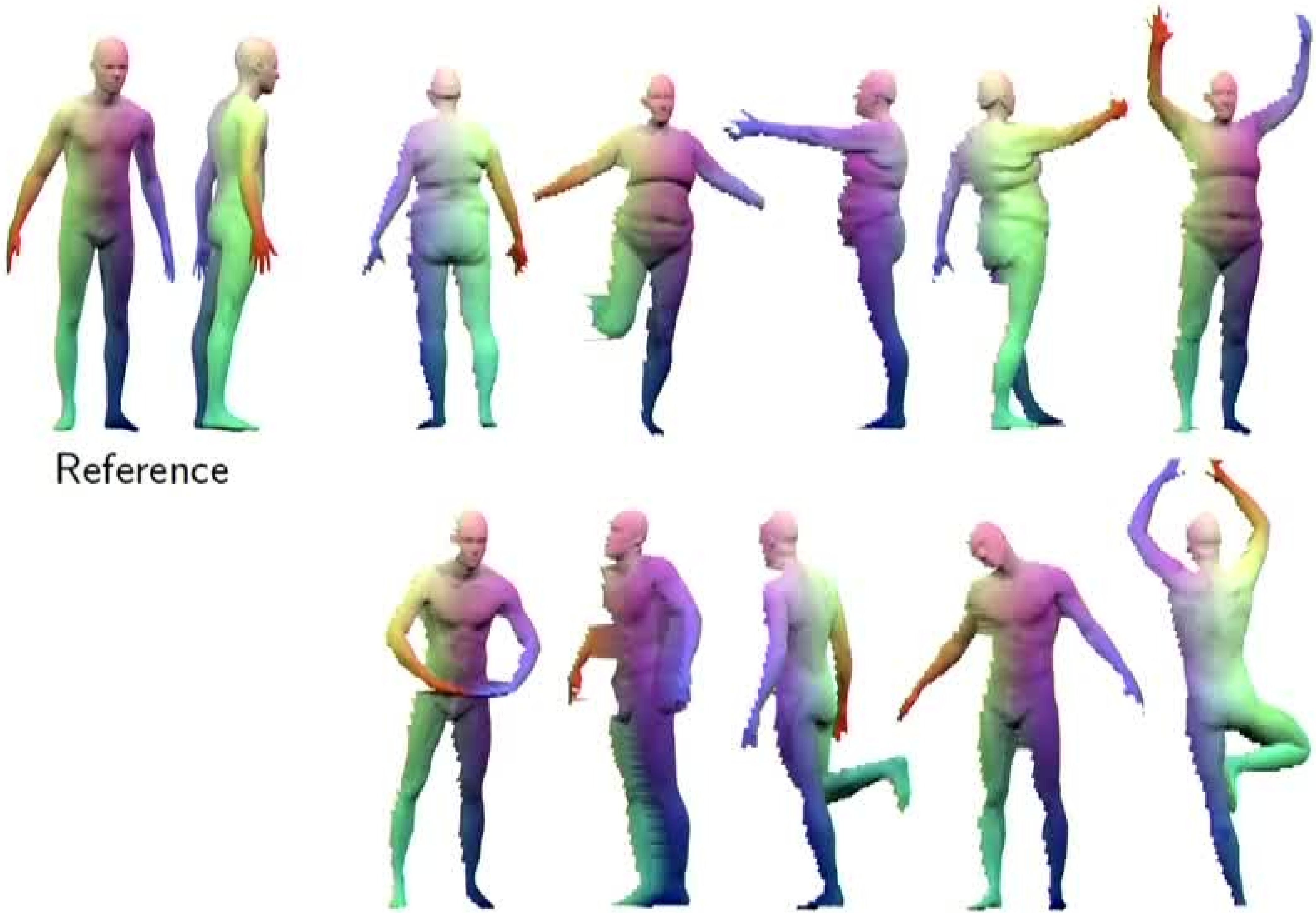
Monti et al. 2016

Correspondence on range images: MoNet

Pointwise correspondence error (geodesic distance from groundtruth)

Monti et al. 2016

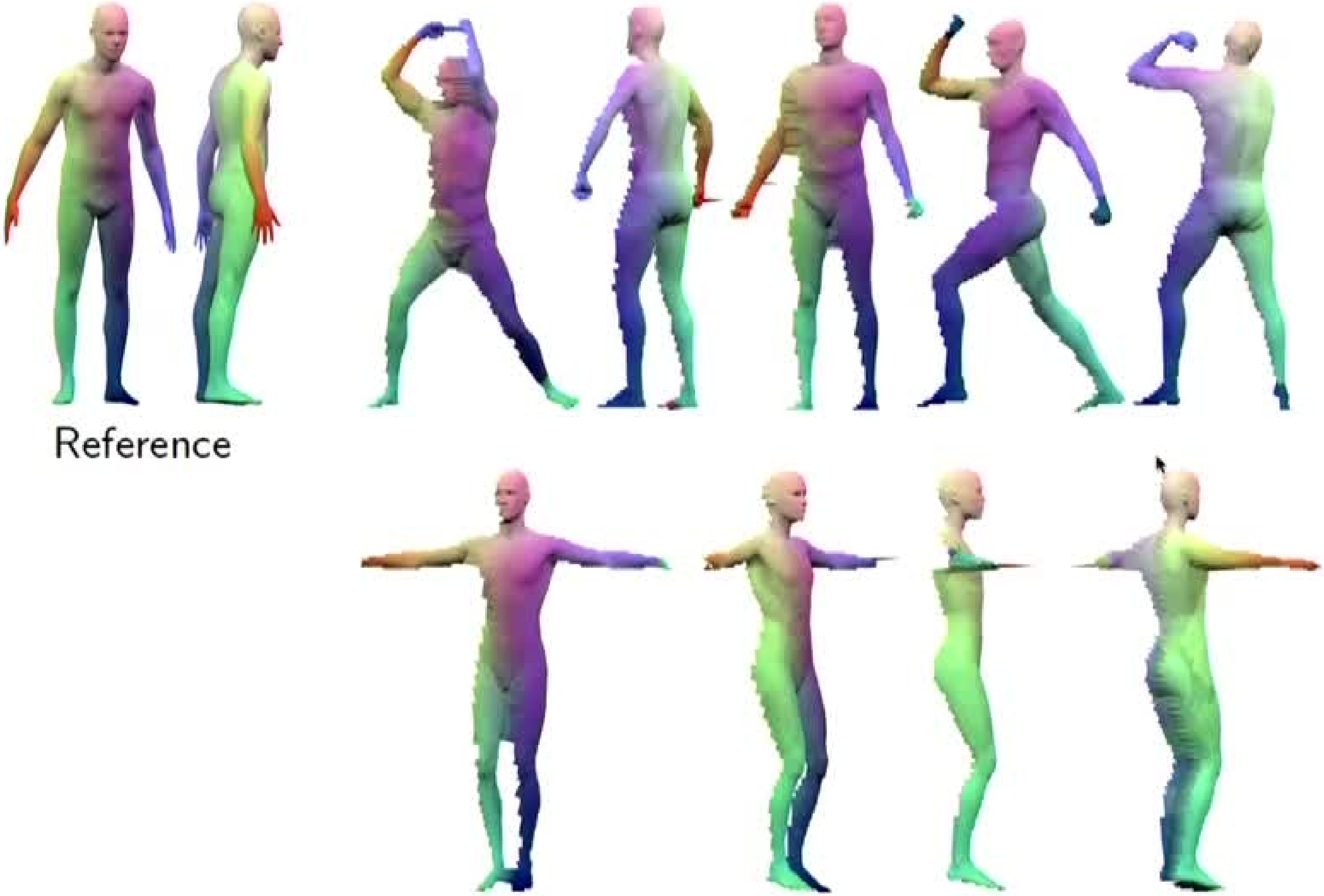
Correspondence with MoNet: Range images



Correspondence visualization (similar colors encode corresponding points)

Monti et al. 2016

Correspondence with MoNet: Range images

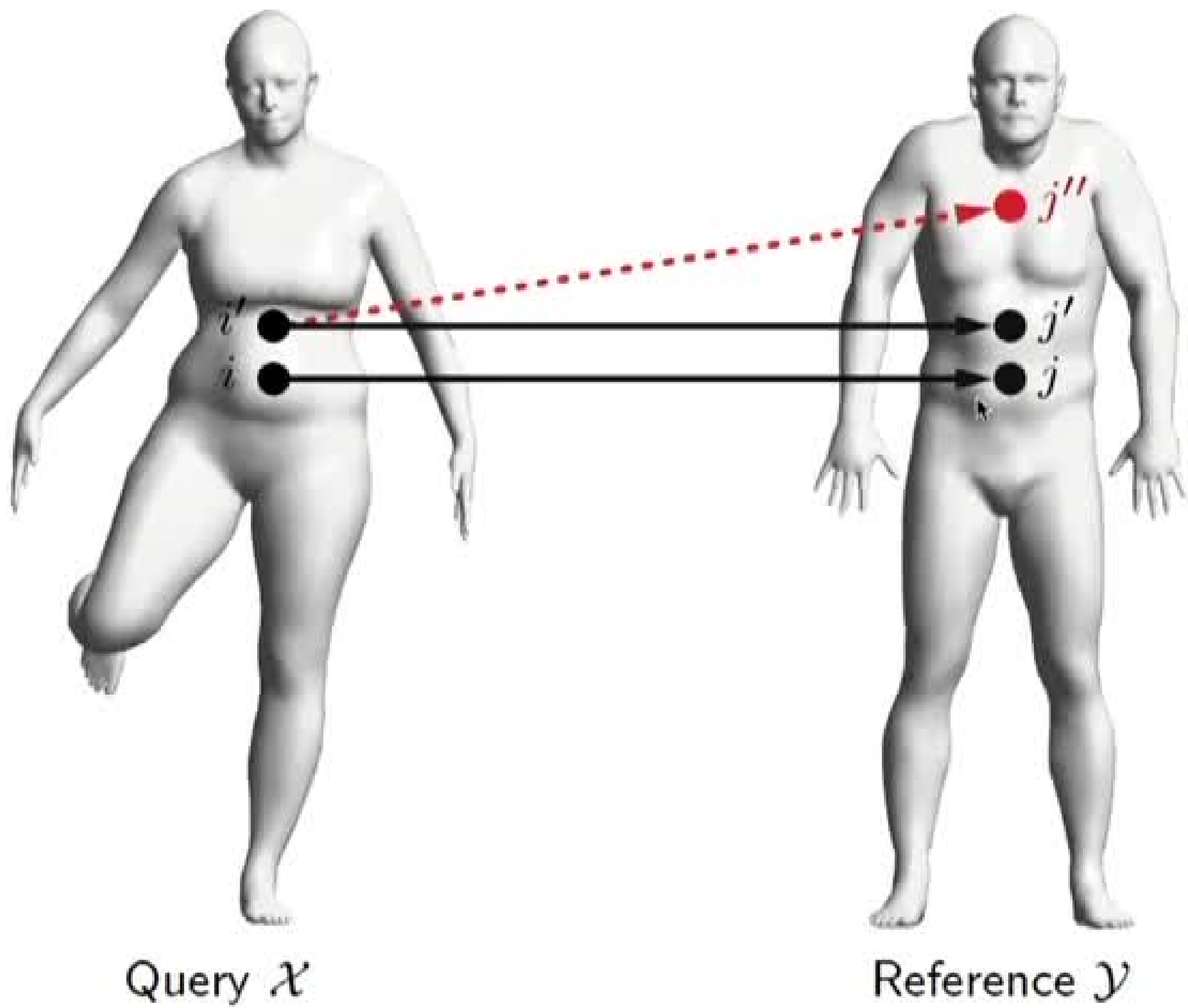


Reference

Correspondence visualization (similar colors encode corresponding points)

Monti et al. 2016

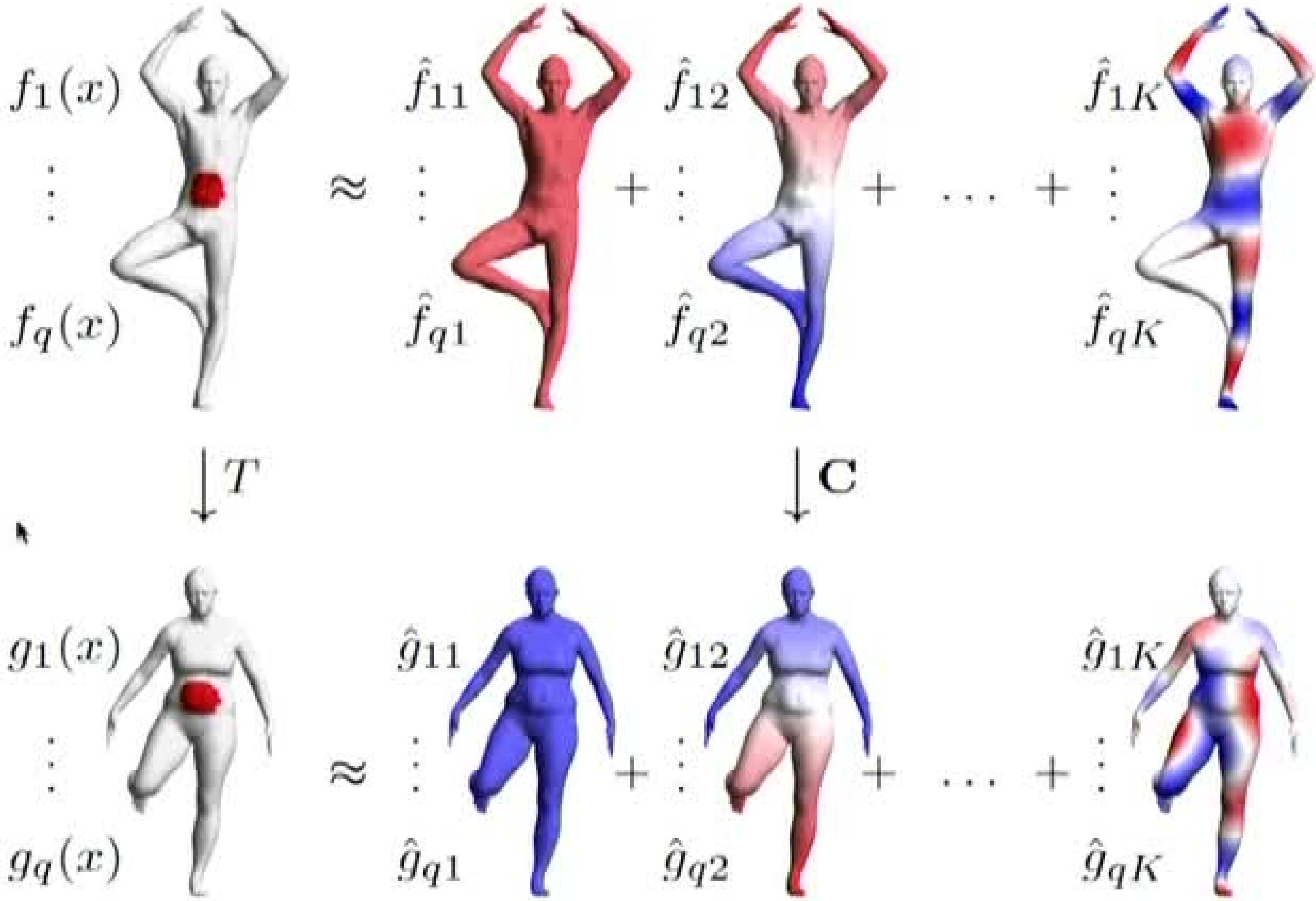
Pointwise vs Structured learning



Nearby points i, i' on query shape are **not guaranteed** to map to nearby points j, j' on reference shape at **test time**

Litany et al. 2017

Functional maps: spectral domain

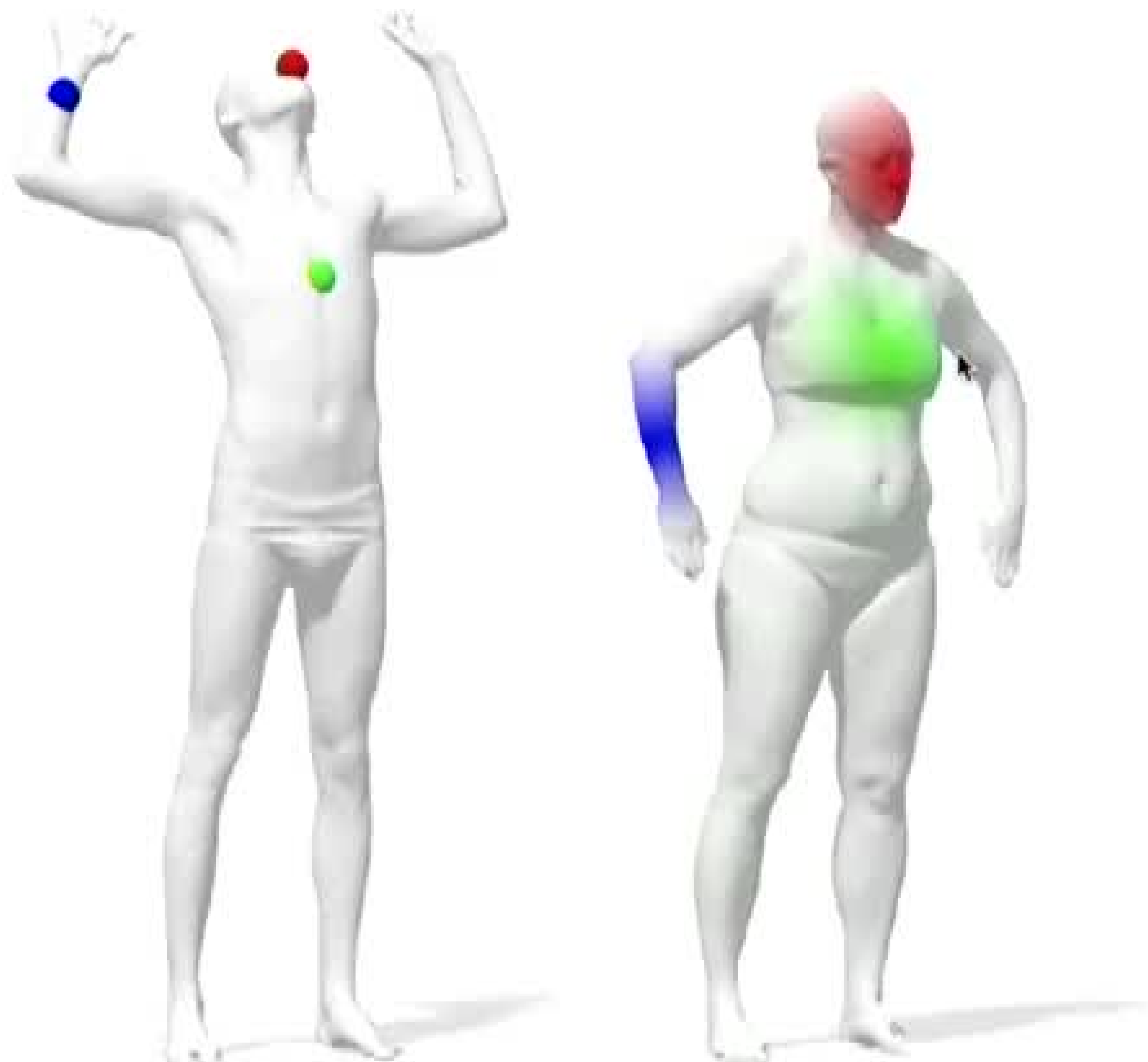


Recover correspondence from $q \geq k$ dimensional pointwise features

$$\mathbf{C}^* = \underset{\mathbf{C}}{\operatorname{argmin}} \|\hat{\mathbf{F}}\mathbf{C} - \hat{\mathbf{G}}\|_{\mathrm{F}}^2$$

Ovsjanikov et al. 2012

Functional maps: spatial domain

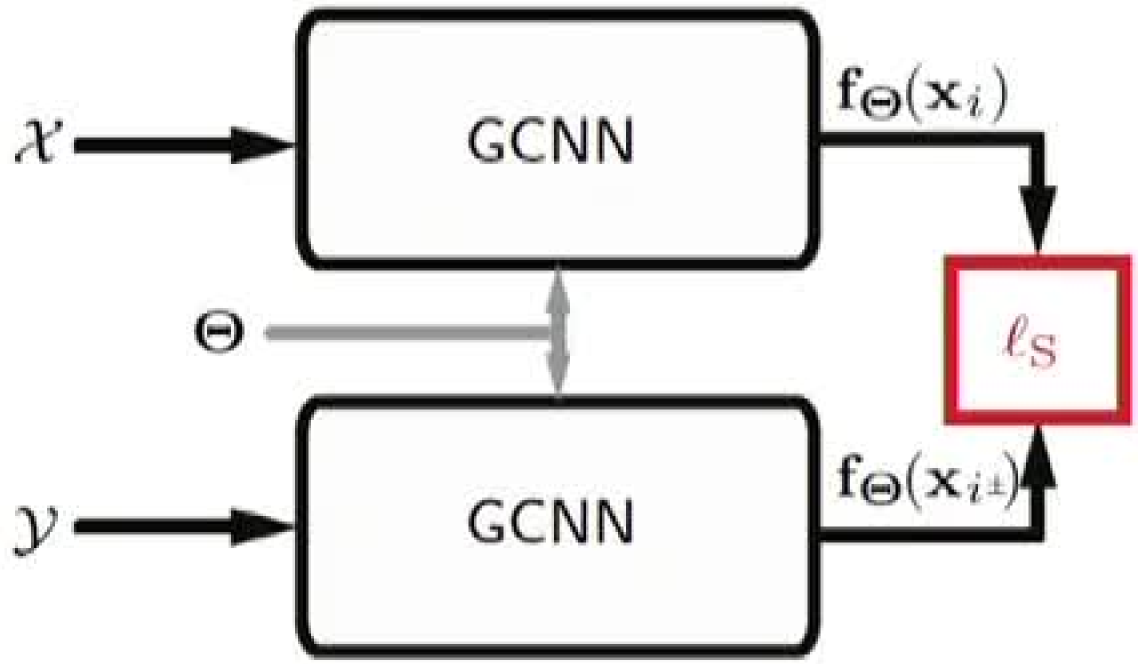


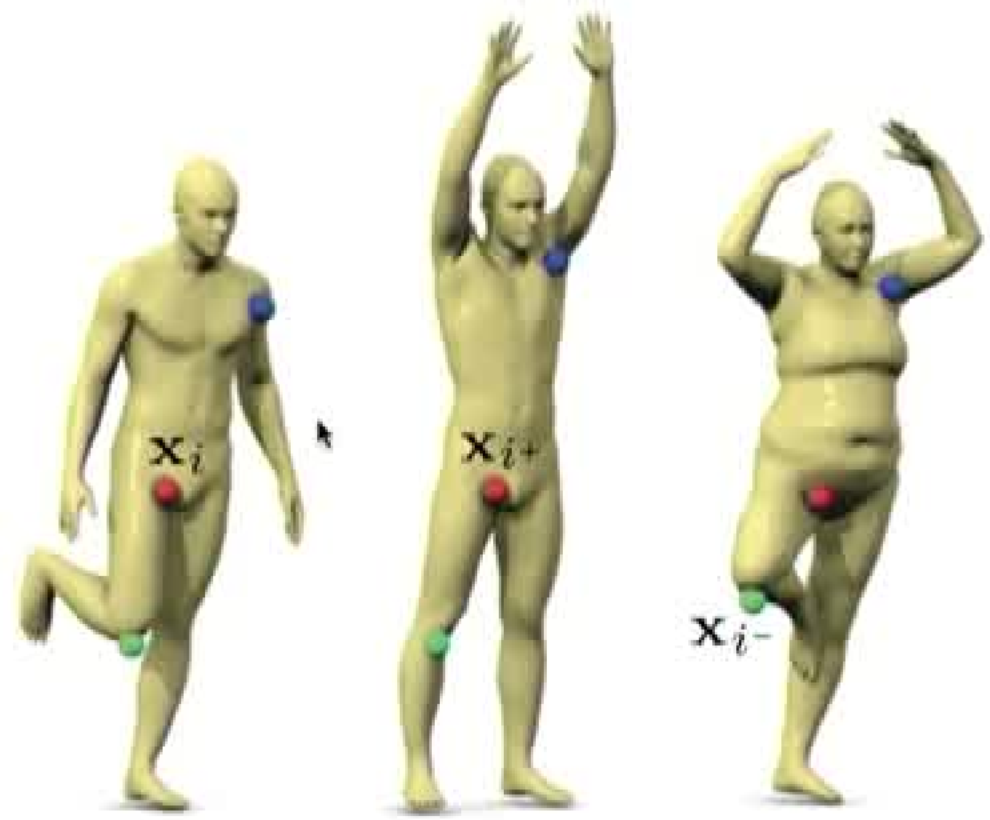
Probability p_{ij} of point j mapping to i

$$\mathbf{P} \approx |\mathbf{\Psi C \Phi}^\top|_{\|\cdot\|}$$

Ovsjanikov et al. 2012

Siamese metric learning

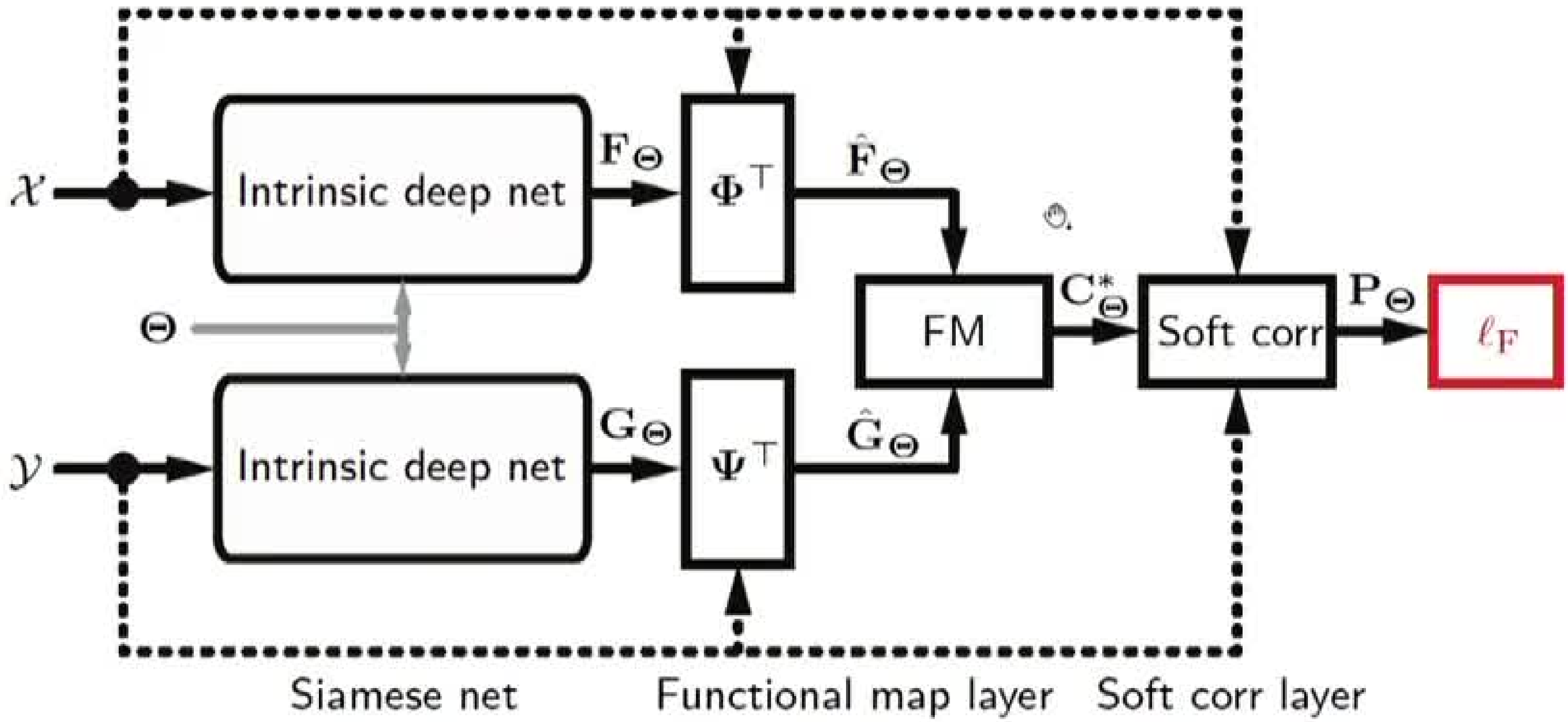




Training set	positive (i, i^+) and negative (i, i^-) pairs of points
Siamese net	two net instances with shared parameters $\boldsymbol{\Theta}$
Poitwise feature cost	$\ell_{\mathrm{S}}(\boldsymbol{\Theta}) = \gamma \sum_{i,i^+} \lVert \mathbf{f}_{\Theta}(\mathbf{x}_i) - \mathbf{f}_{\Theta}(\mathbf{x}_{i^+}) \rVert_2^2 + (1-\gamma) \sum_{i,i^-} \left[\mu - \lVert \mathbf{f}_{\Theta}(\mathbf{x}_i) - \mathbf{f}_{\Theta}(\mathbf{x}_{i^-}) \rVert_2^2\right]_+$

Masci et al. 2015

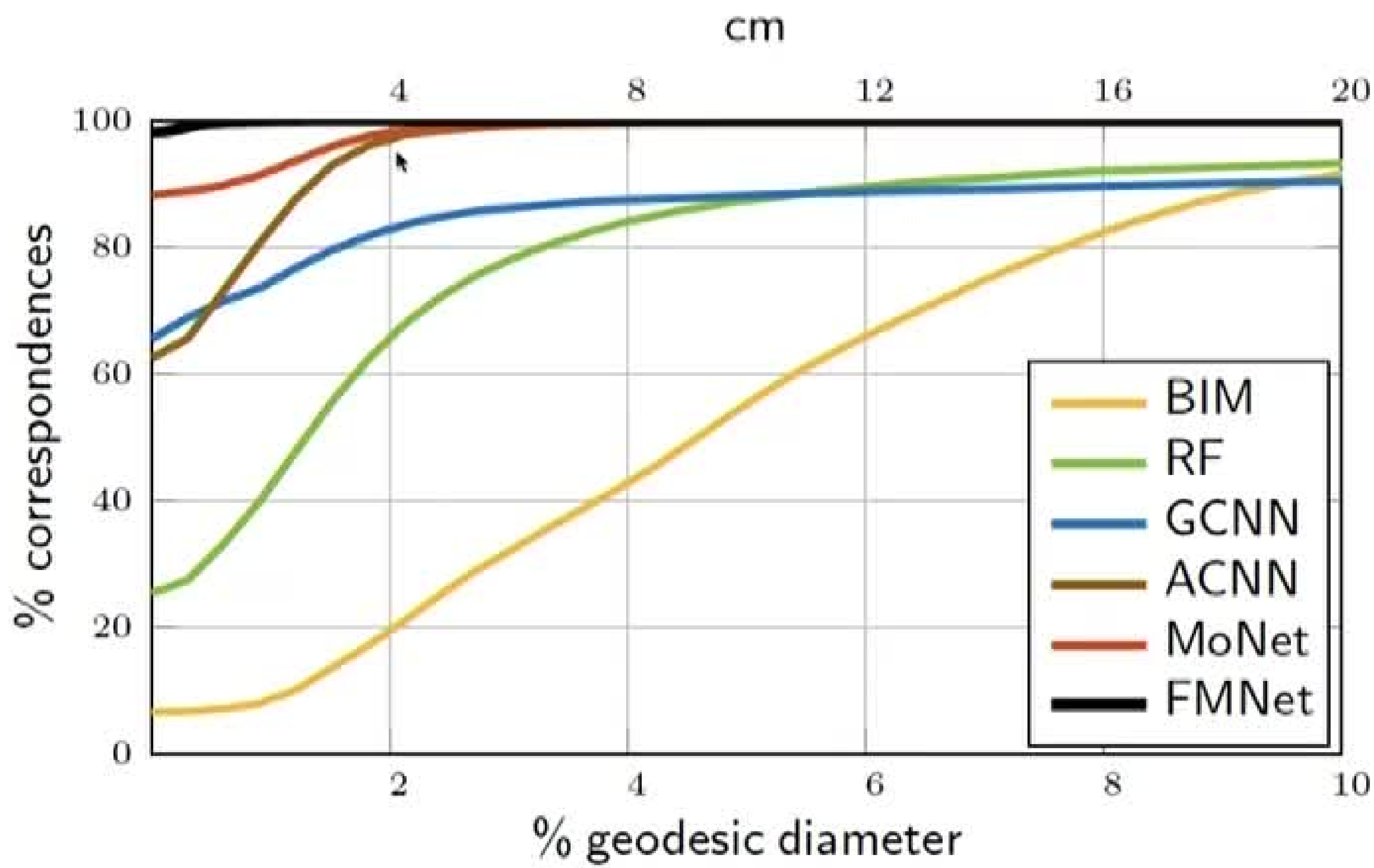
Structured correspondence with FMNet



Siamese net	two net instances with shared parameters $\mathbf{\Theta}$
Functional map layer	$\mathbf{C}^*_\Theta = \hat{\mathbf{F}}^\dagger_\Theta \hat{\mathbf{G}}_\Theta$
Soft correspondence layer	$\mathbf{P}_\Theta = \lvert \mathbf{\Psi} \mathbf{C}_\Theta \mathbf{\Phi}^\top \rvert_{\lVert\cdot\rVert}$
Soft error cost	$\ell_\mathrm{F}(\mathbf{\Theta}) = \lVert \mathbf{P}_\Theta \circ \mathbf{D}_\mathcal{Y} \rVert$

Litany et al. 2017

Correspondence quality comparison



Correspondence evaluated using asymmetric Princeton benchmark
(training and testing: disjoint subsets of FAUST)

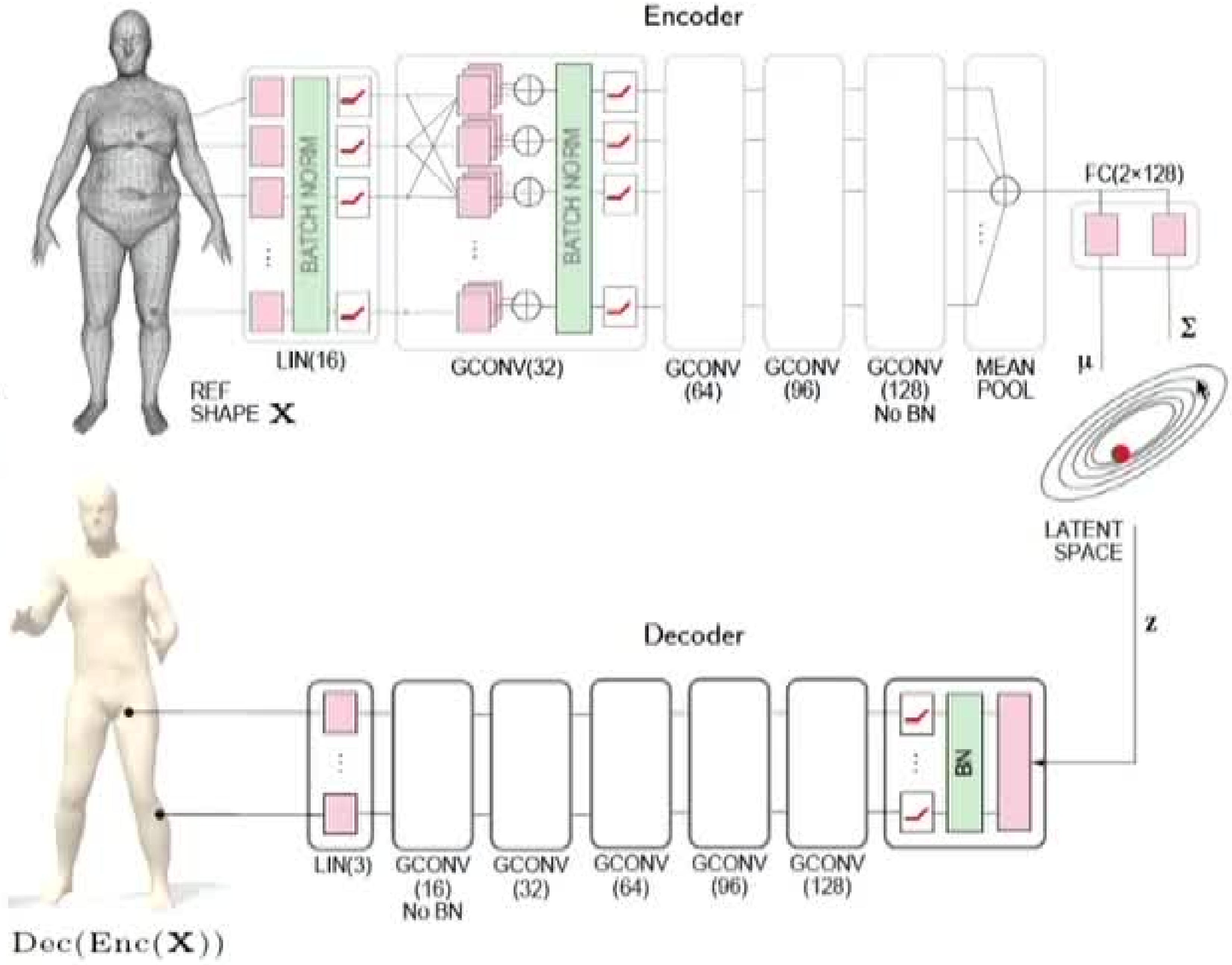
Methods: Kim et al. 2011 (BIM); Rodolà et al. 2014 (RF); Boscaini et al. 2015 (ADD); Masci et al. 2015 (GCNN); Boscaini et al. 2016 (ACNN); Monti et al. 2016 (MoNet); Litany et al. 2017 (FMNet); data: Bogo et al. 2014 (FAUST); benchmark: Kim et al. 2011

Intrinsic Variational Autoencoder (VAE)

Minimize

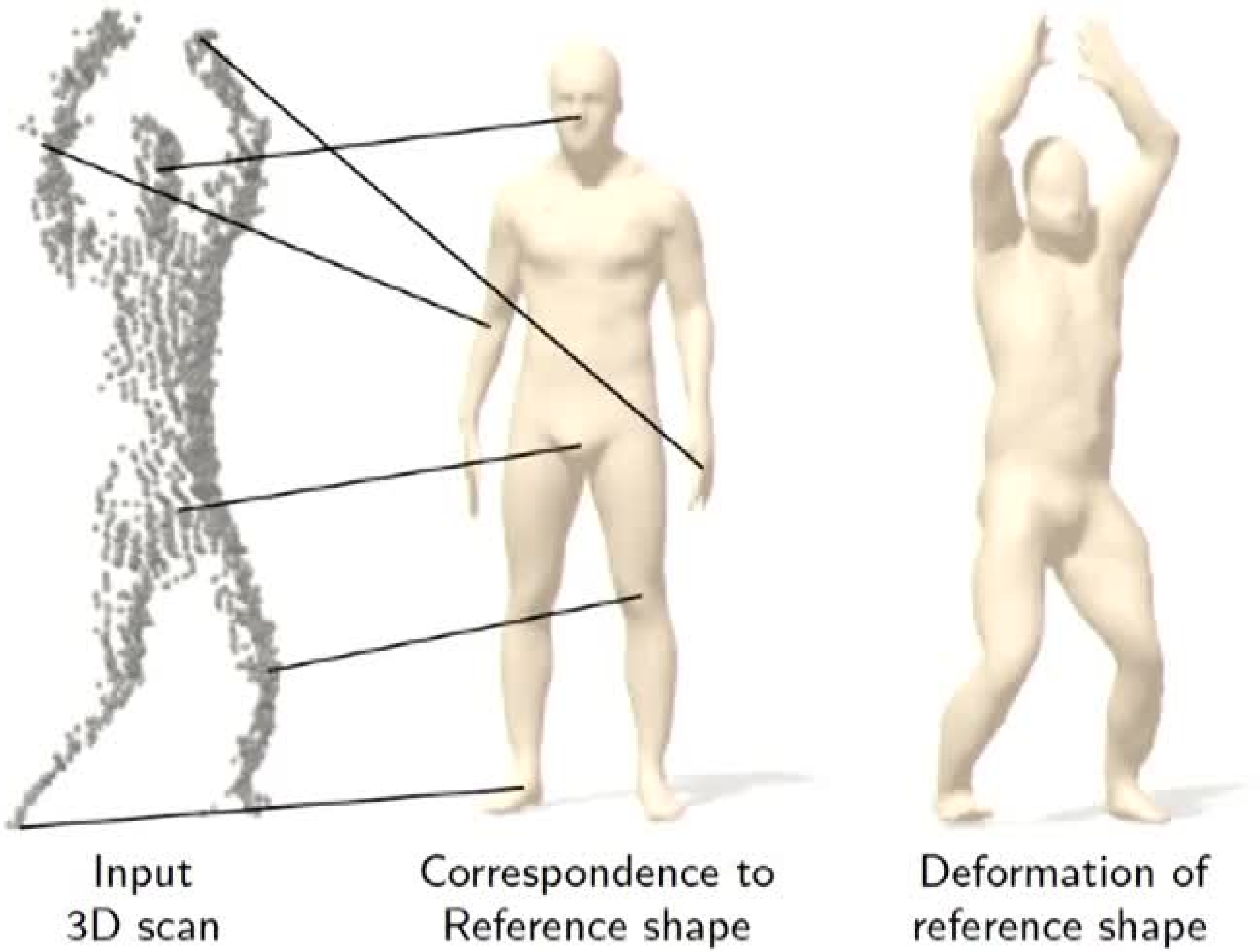
$$\|\mathrm{Dec}(\mathrm{Enc}(\mathbf{X})) - \mathbf{X}\|_{\mathrm{F}} + \lambda D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{X})\|p(\mathbf{z}))$$

w.r.t. net parameters

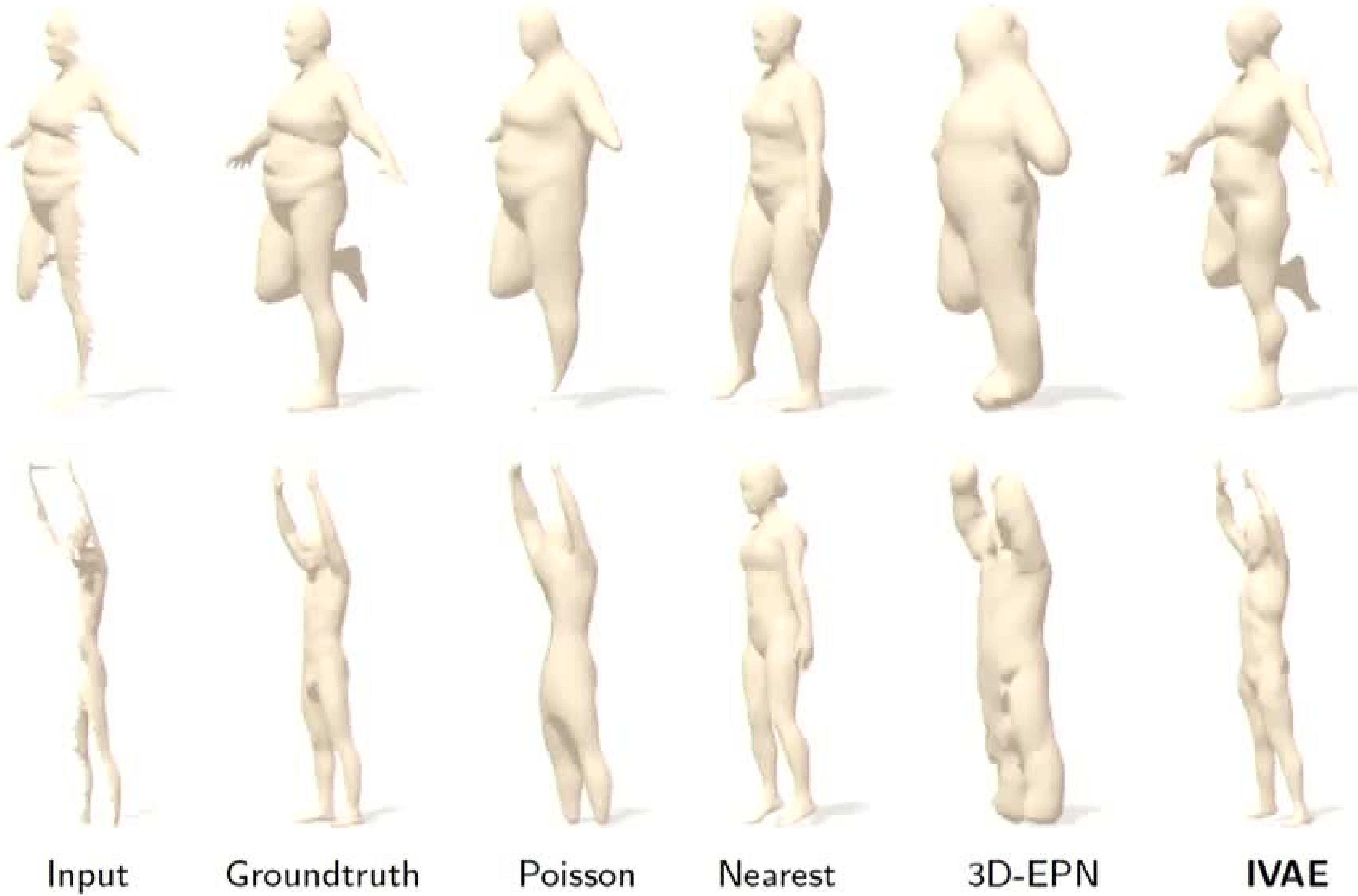


Litany et al. 2017; training on Dynamic FAUST (Bogo et al. 2017)

3D shape analysis and synthesis

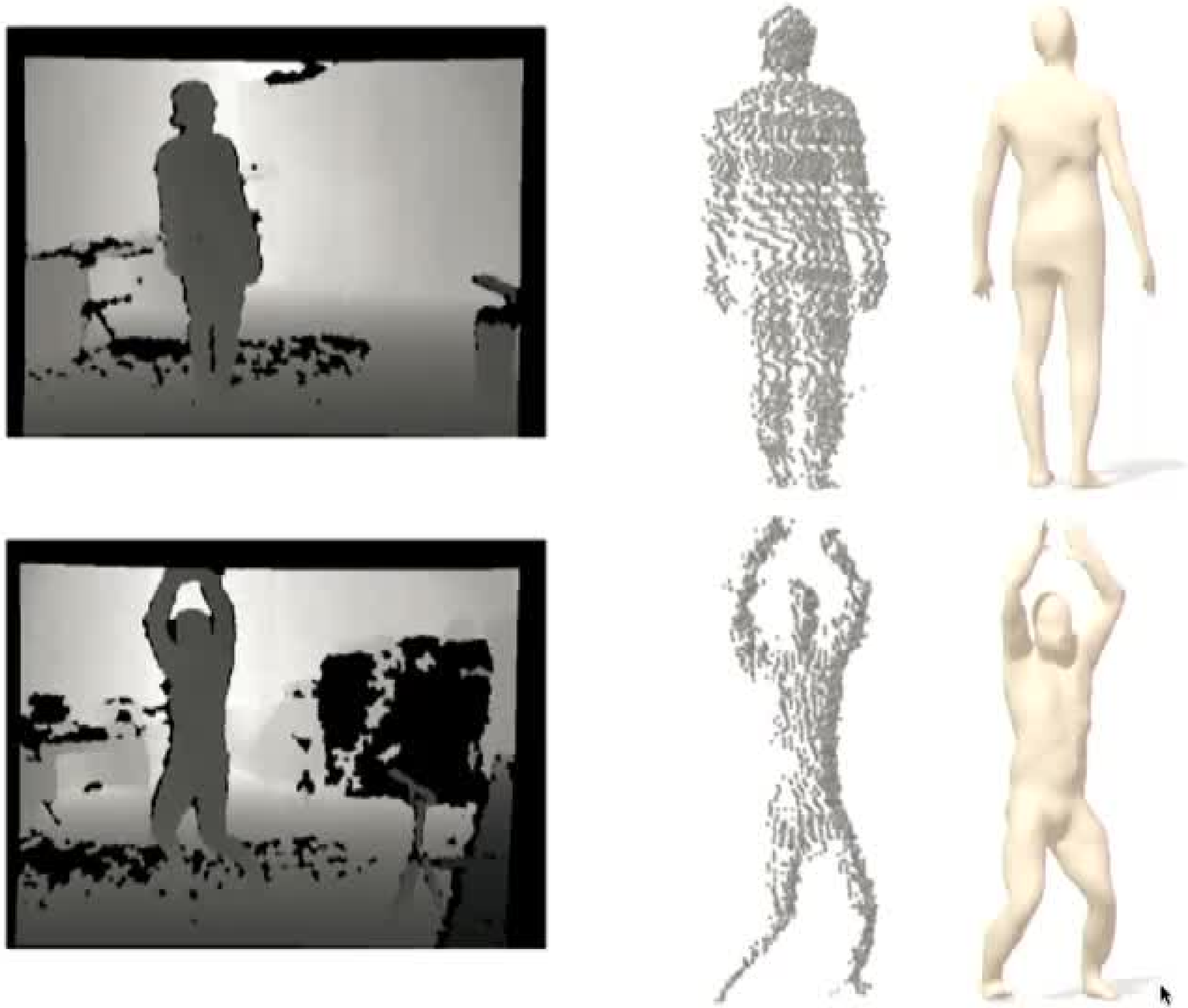


Shape completion comparison



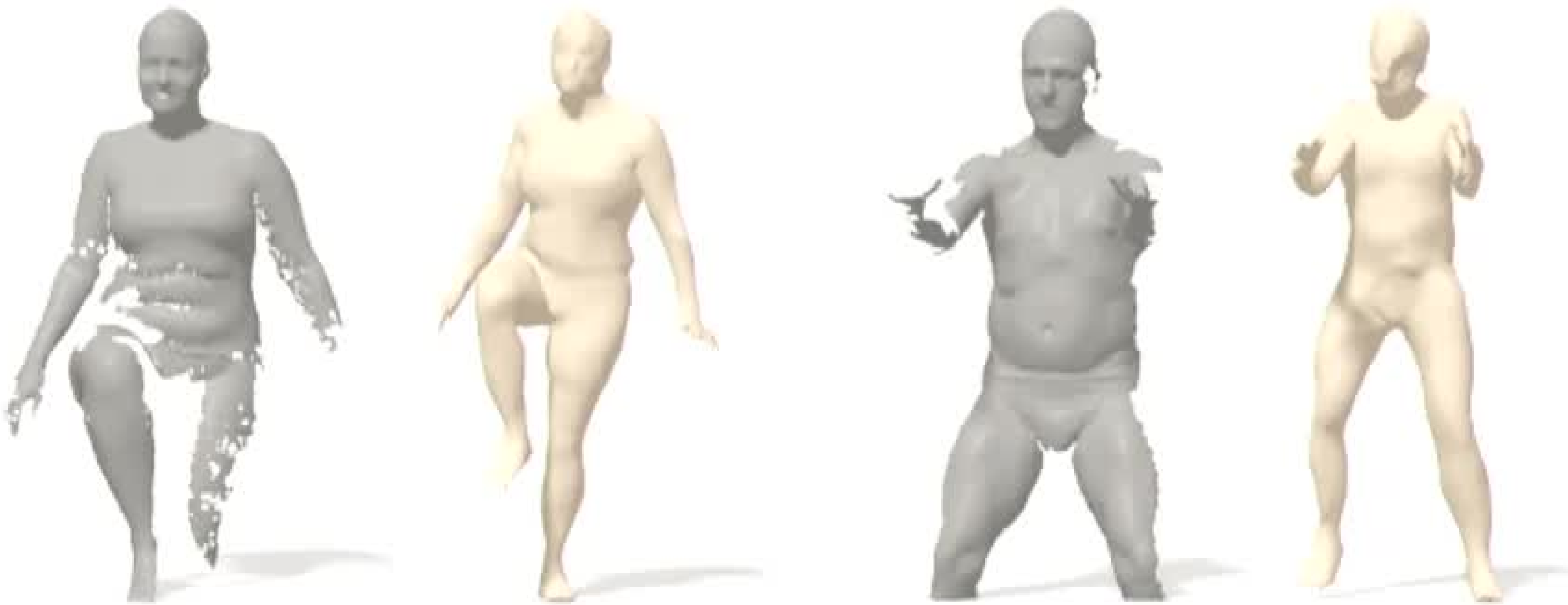
Methods: Litany et al. 2017; Dai et al. 2016 (3D-EPN); Kazhdan et al. 2013 (Poisson)

Shape completion examples



Litany et al. 2017; data: Ofli et al. 2014 (MHAD)

Shape completion examples

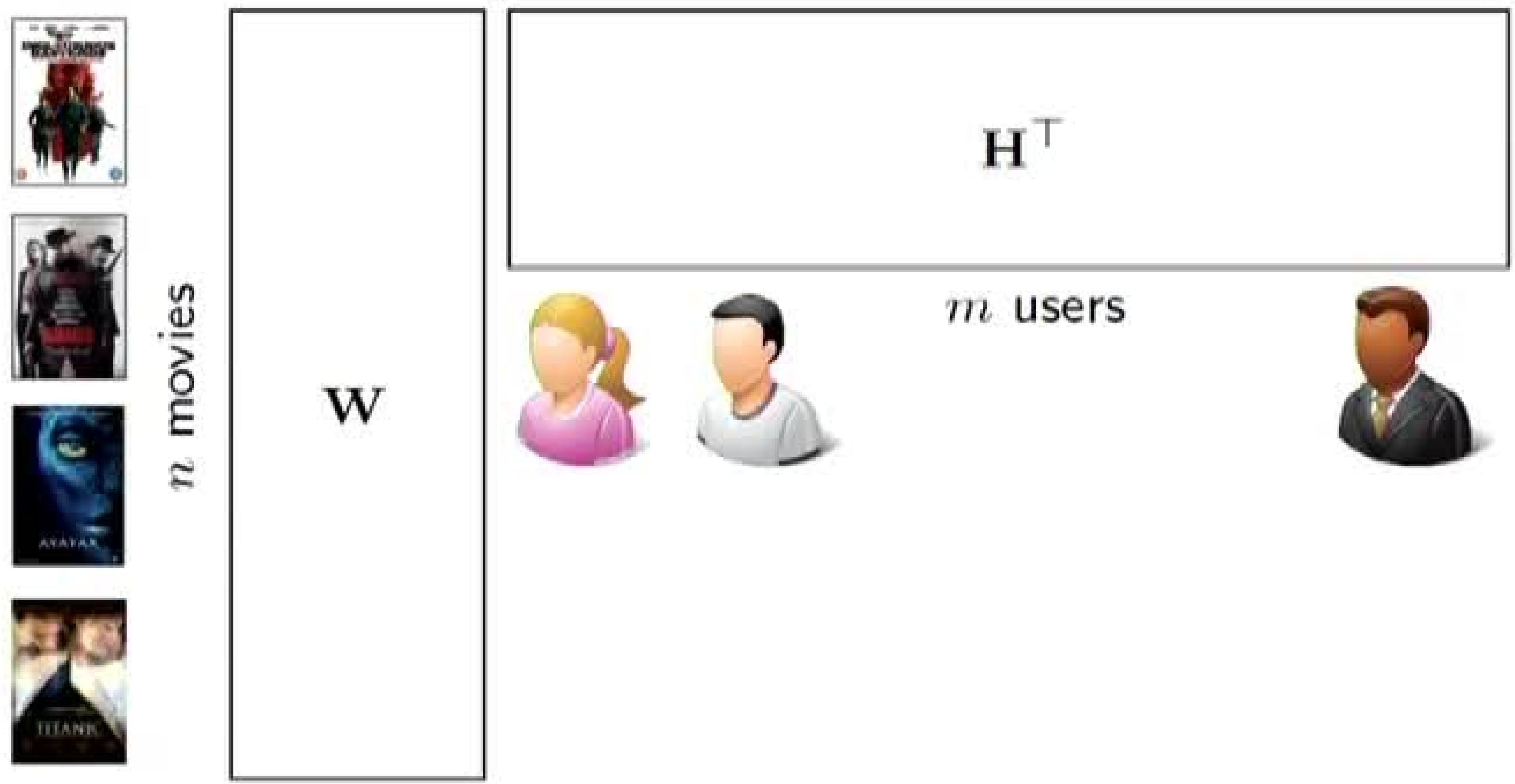


Litany et al. 2017; data: Bogo et al. 2014 (FAUST)


Delivery Man
12 Years a Slave
Gravity
Frozen
American Hustle
Thor: The Dar...
The Hunger Ga...
Captain Phillips
Dallas Buyers Club
Escape Plan
Saving Mr. Banks
Reasonable Doubt
Last Vegas
Ender's Game
Jackass Prese...
Rush
The Wolverine...
About Time
The Bag Man
Nebraska
Despicable Me 2
World War Z UNR...
The Counselor
We're the Mil...
Man of Steel
Now You See M...
Cloudy with a...
This Is the End
The Returned
Homefront
August Osage C...
Elysium
Pacific Rim
The Great Gatsby
The Lone Ranger
Blue Jasmine
2 Guns
Romeo and Juliet
Monsters Univer...
The Hobbit: A...


Delivery Man
12 Years a Slave
Gravity
Frozen
American Hustle
Thor: The Dar...
The Hunger Ga...
Captain Phillips
Dallas Buyers Club
Escape Plan
Saving Mr. Banks
Reasonable Doubt
Last Vegas
Ender's Game
Jackass Prese...
Rush
The Wolverine...
About Time
The Bag Man
Nebraska
Despicable Me 2
World War Z UNR...
The Counselor
We're the Mil...
Man of Steel
Now You See M...
Cloudy with a...
This Is the End
The Returned
Homefront
August Osage C...
Elysium
Pacific Rim
The Great Gatsby
The Lone Ranger
Blue Jasmine
2 Guns
Romeo and Juliet
Monsters Univer...
The Hobbit: A...

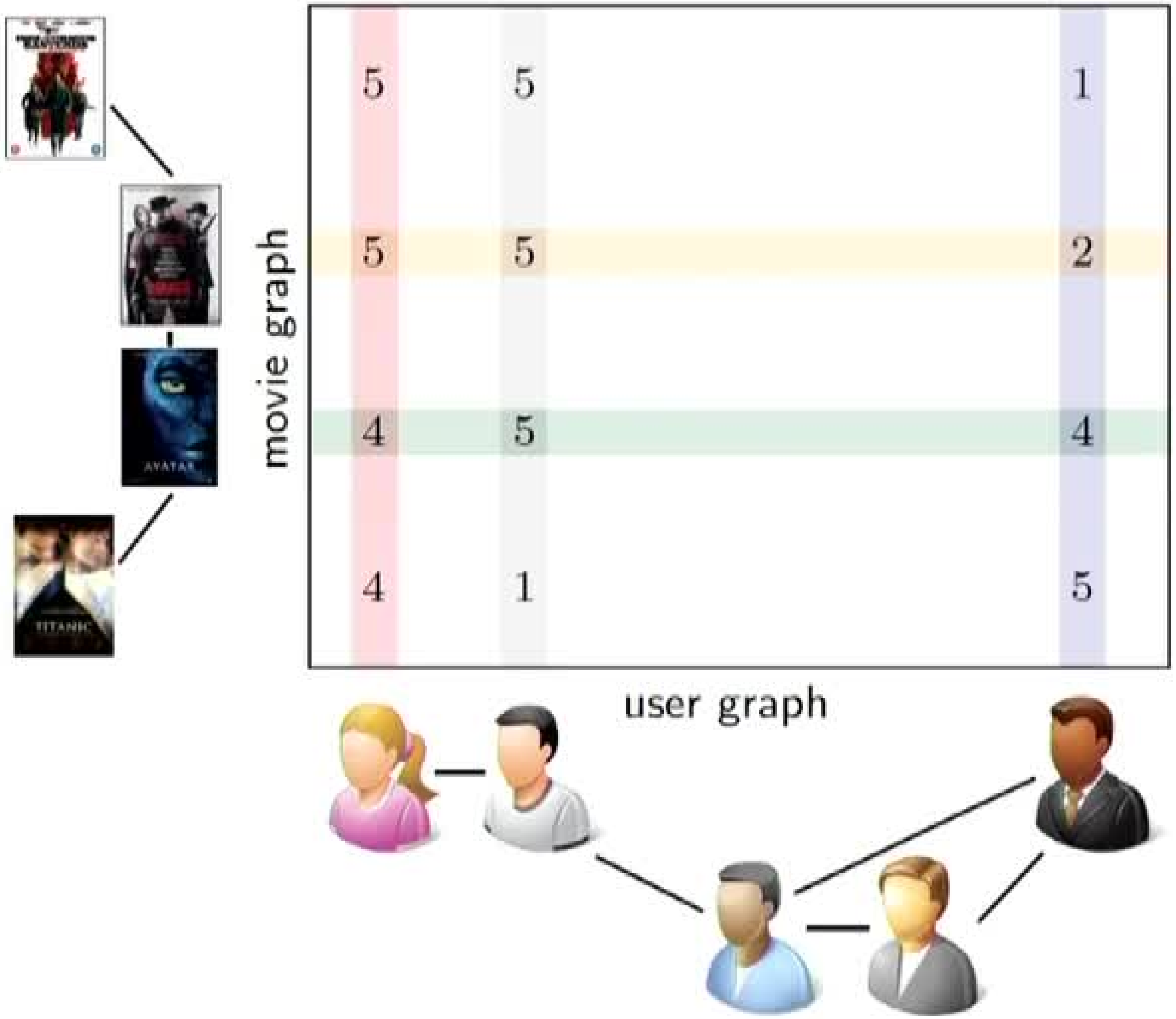
Factorized matrix completion models



$$\min_{\substack{\mathbf{W}\in\mathbb{R}^{m\times s}\\ \mathbf{H}\in\mathbb{R}^{n\times s}}} \quad \mu\|\mathbf{\Omega}\circ(\mathbf{X}-\mathbf{A})\|_{\mathrm{F}}^2 + \mu_{\mathrm{c}}\|\mathbf{W}\|_{\mathrm{F}}^2 + \mu_{\mathrm{r}}\|\mathbf{H}\|_{\mathrm{F}}^2$$

Srebro, Rennie, Jaakkola 2004

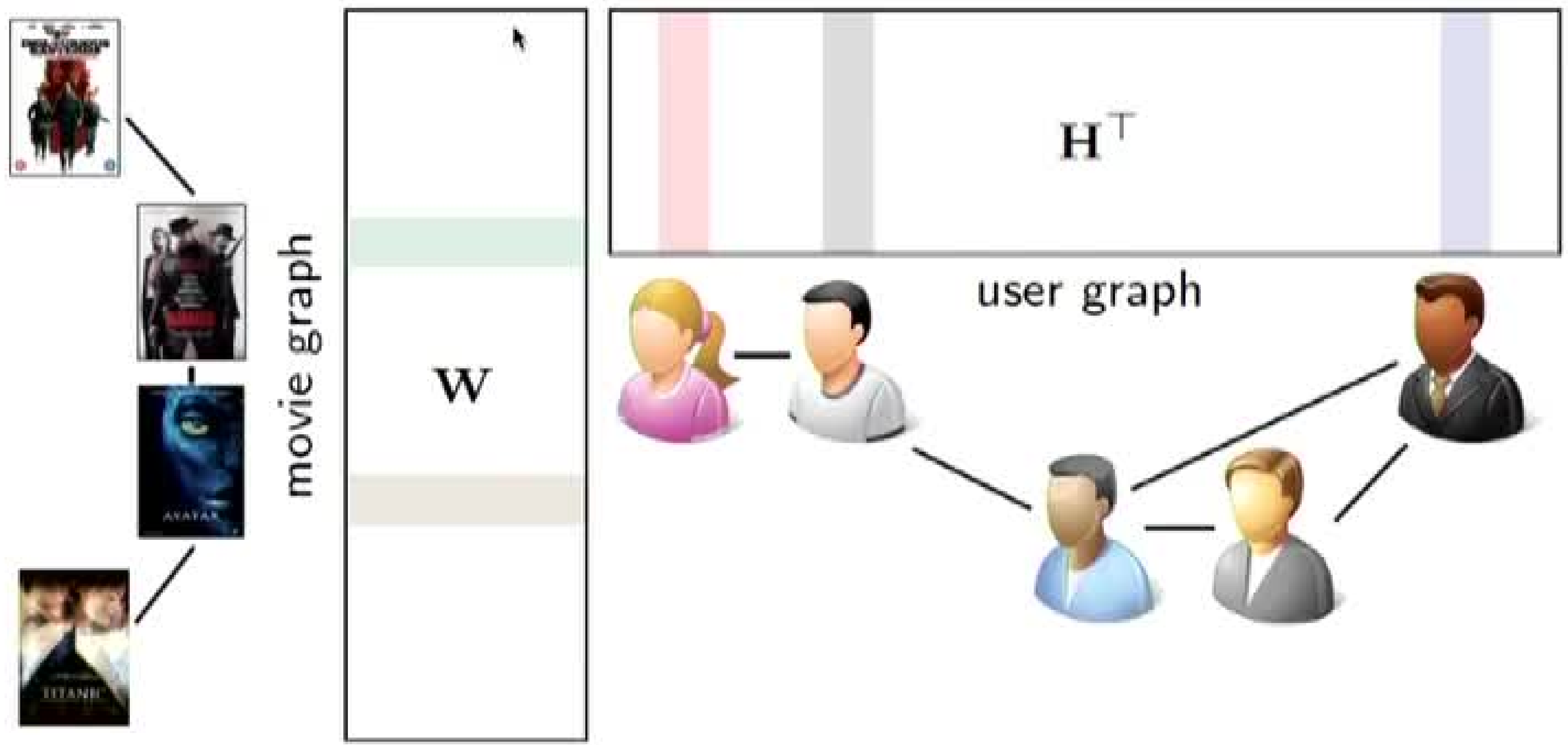
Geometric matrix completion



$$\min_{\mathbf{X}\in\mathbb{R}^{m\times n}} \quad \mu\|\mathbf{\Omega}\circ(\mathbf{X}-\mathbf{A})\|_{\mathrm{F}}^2 + \mu_{\mathrm{c}}\underbrace{\mathrm{tr}(\mathbf{X}\mathbf{\Delta}_{\mathrm{c}}\mathbf{X}^\top)}_{\|\mathbf{X}\|_{\mathcal{G}_{\mathrm{c}}}^2} + \mu_{\mathrm{r}}\underbrace{\mathrm{tr}(\mathbf{X}^\top\mathbf{\Delta}_{\mathrm{r}}\mathbf{X})}_{\|\mathbf{X}\|_{\mathcal{G}_{\mathrm{r}}}^2}$$

Coifman, Gavish 2011; Kalofolias et al. 2014

Factorized geometric matrix completion models



$$\min_{\substack{\mathbf{W}\in\mathbb{R}^{m\times s}\\ \mathbf{H}\in\mathbb{R}^{n\times s}}} \quad \mu\|\mathbf{\Omega}\circ(\mathbf{X}-\mathbf{A})\|_{\mathrm{F}}^2 + \mu_{\mathrm{c}}\mathrm{tr}(\mathbf{H}^\top\mathbf{\Delta}_{\mathrm{c}}\mathbf{H}) + \mu_{\mathrm{r}}\mathrm{tr}(\mathbf{W}^\top\mathbf{\Delta}_{\mathrm{r}}\mathbf{W})$$

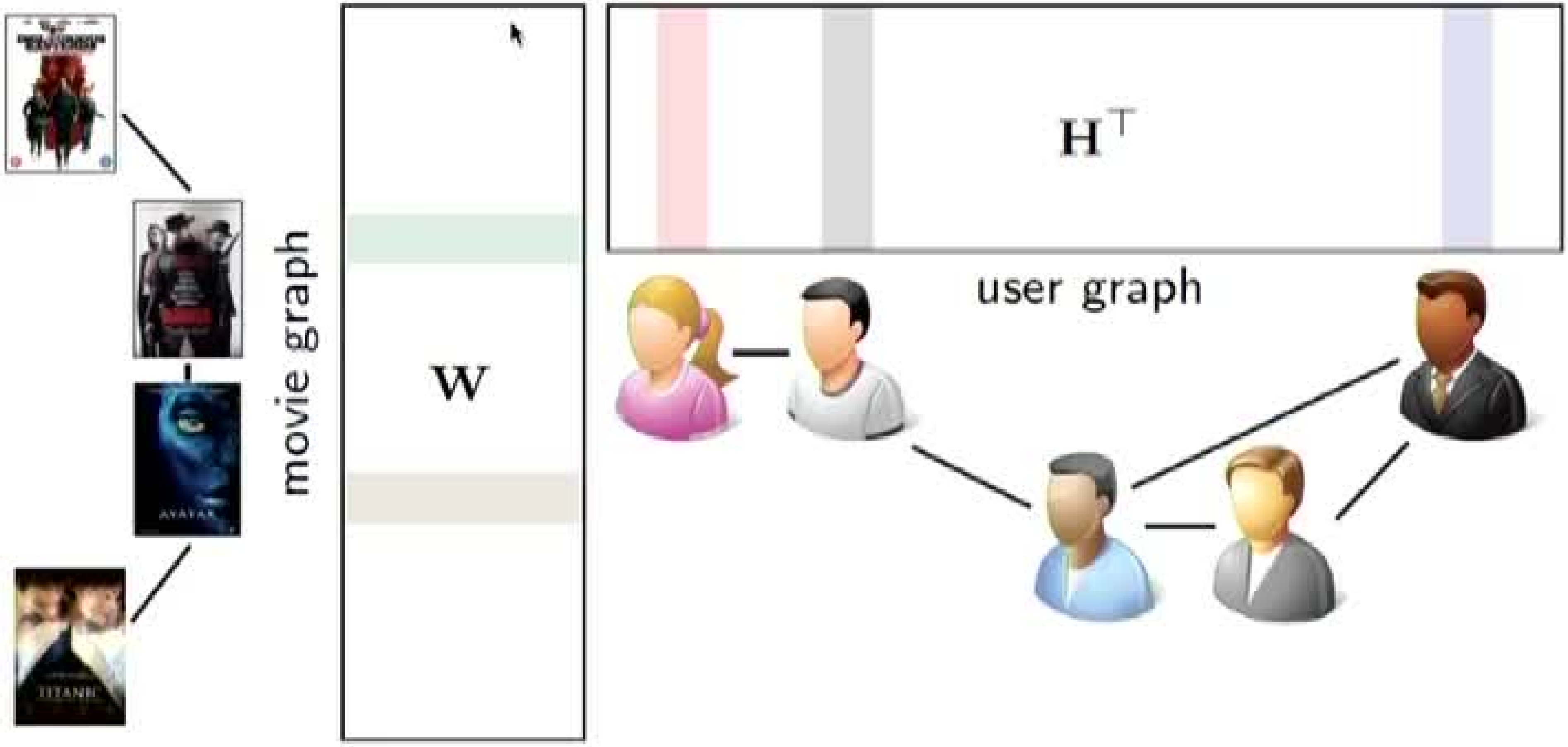
☺ $\mathcal{O}(n+m)$ variables instead of $\mathcal{O}(nm)$

☺ $\mathrm{rank}(\mathbf{X}) = \mathrm{rank}(\mathbf{W}\mathbf{H}^\top) \leq s$ by construction

Rao et al. 2015

How to learn filters on multiple graphs?

Factorized geometric matrix completion models



$$\min_{\substack{\mathbf{W}\in\mathbb{R}^{m\times s}\\ \mathbf{H}\in\mathbb{R}^{n\times s}}} \quad \mu\|\boldsymbol{\Omega}\circ(\mathbf{X}-\mathbf{A})\|_{\mathrm{F}}^2 + \mu_{\mathrm{c}}\mathrm{tr}(\mathbf{H}^\top\boldsymbol{\Delta}_{\mathrm{c}}\mathbf{H}) + \mu_{\mathrm{r}}\mathrm{tr}(\mathbf{W}^\top\boldsymbol{\Delta}_{\mathrm{r}}\mathbf{W})$$

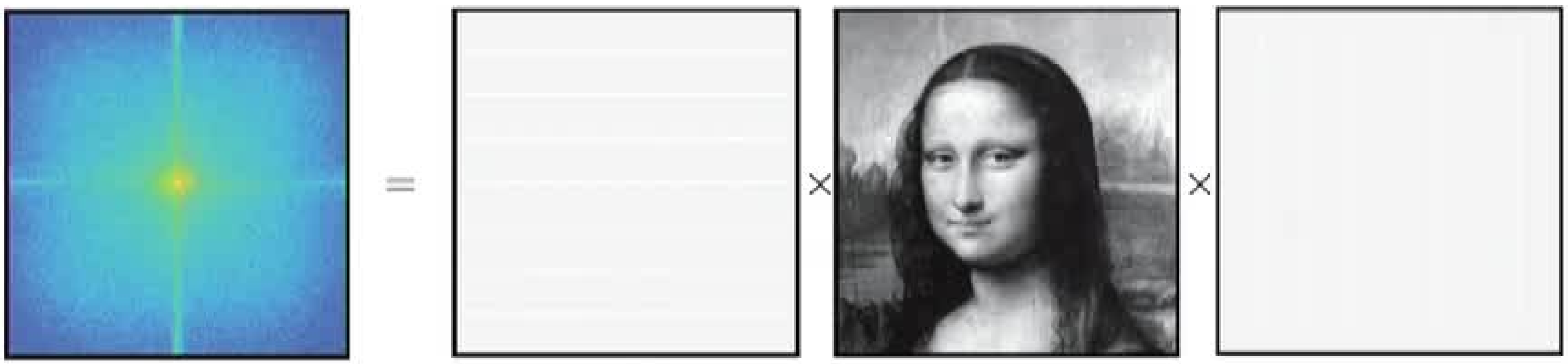
- ☺ $\mathcal{O}(n+m)$ variables instead of $\mathcal{O}(nm)$
- ☺ $\mathrm{rank}(\mathbf{X}) = \mathrm{rank}(\mathbf{W}\mathbf{H}^\top) \leq s$ by construction

Rao et al. 2015

2D Fourier transform

x

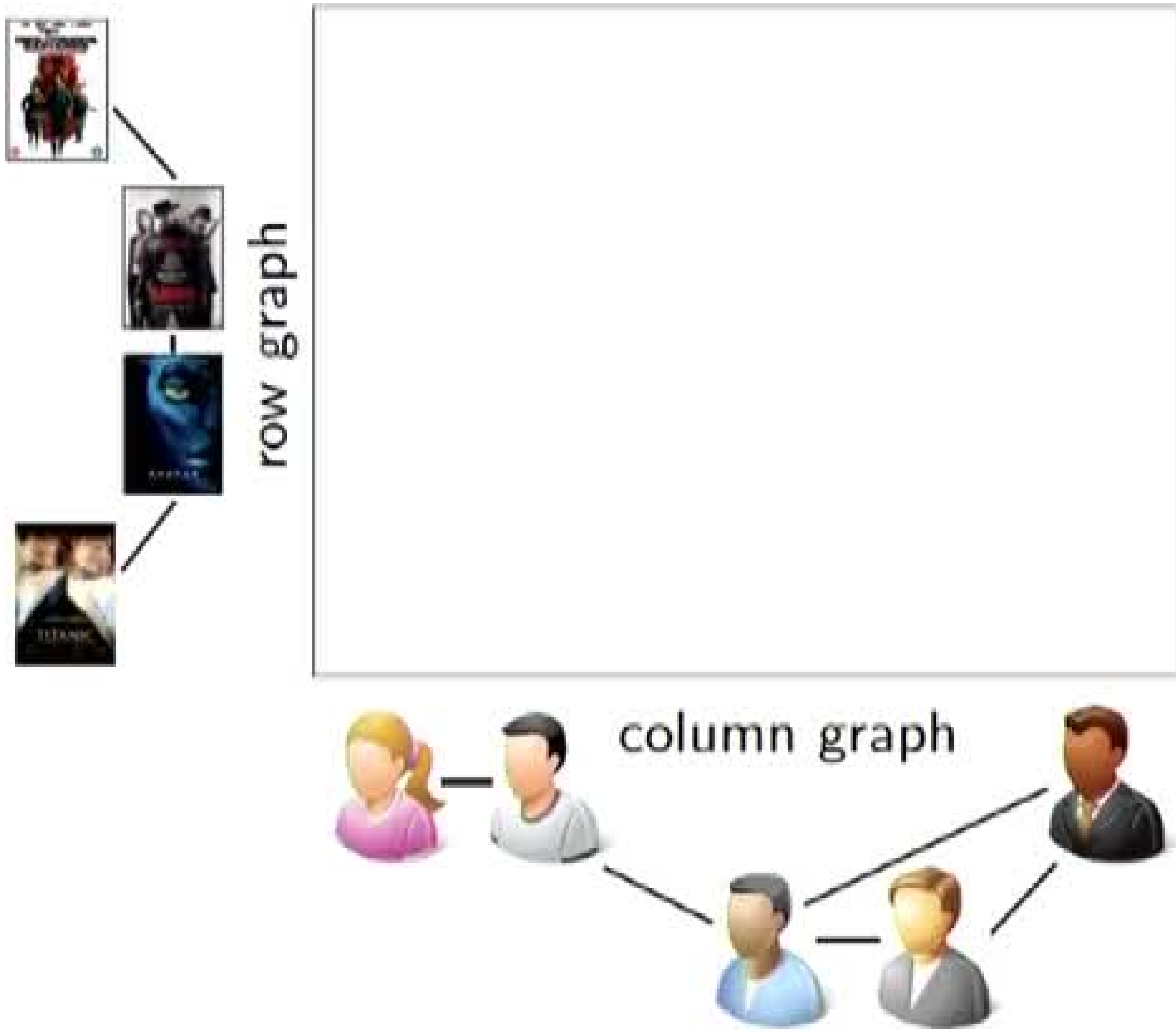
2D Fourier transform



$$\hat{\mathbf{X}} = \boldsymbol{\Phi}^\top \times \mathbf{X} \times \boldsymbol{\Phi}$$

Column-wise trasform + Row-wise transform = 2D transform

Multi-graph spectral filters



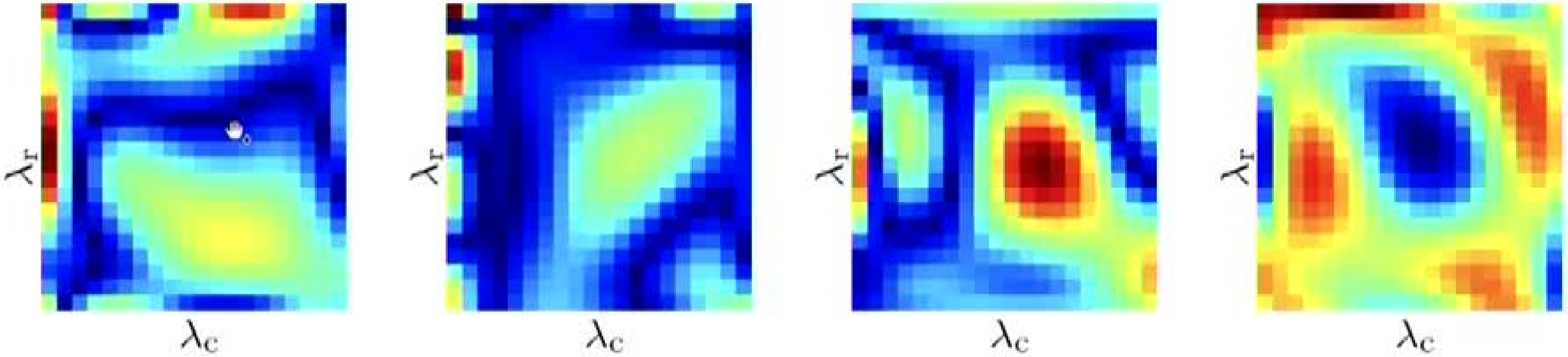
Multi-graph bi-variate polynomial filter

$$\mathbf{Y} = \tau_{\mathbf{\Theta}}(\mathbf{X}) = \sum_{j,j'=1}^{r} \theta_{jj'} \mathbf{\Delta}_{\mathrm{r}}^{j} \mathbf{X} \mathbf{\Delta}_{\mathrm{c}}^{j'},$$

where $\mathbf{\Theta} = (\theta_{jj'})$ is the $r \times r$ matrix of filter parameters

Sandryhaila, Moura 2014; Monti, Bresson, Bronstein 2017

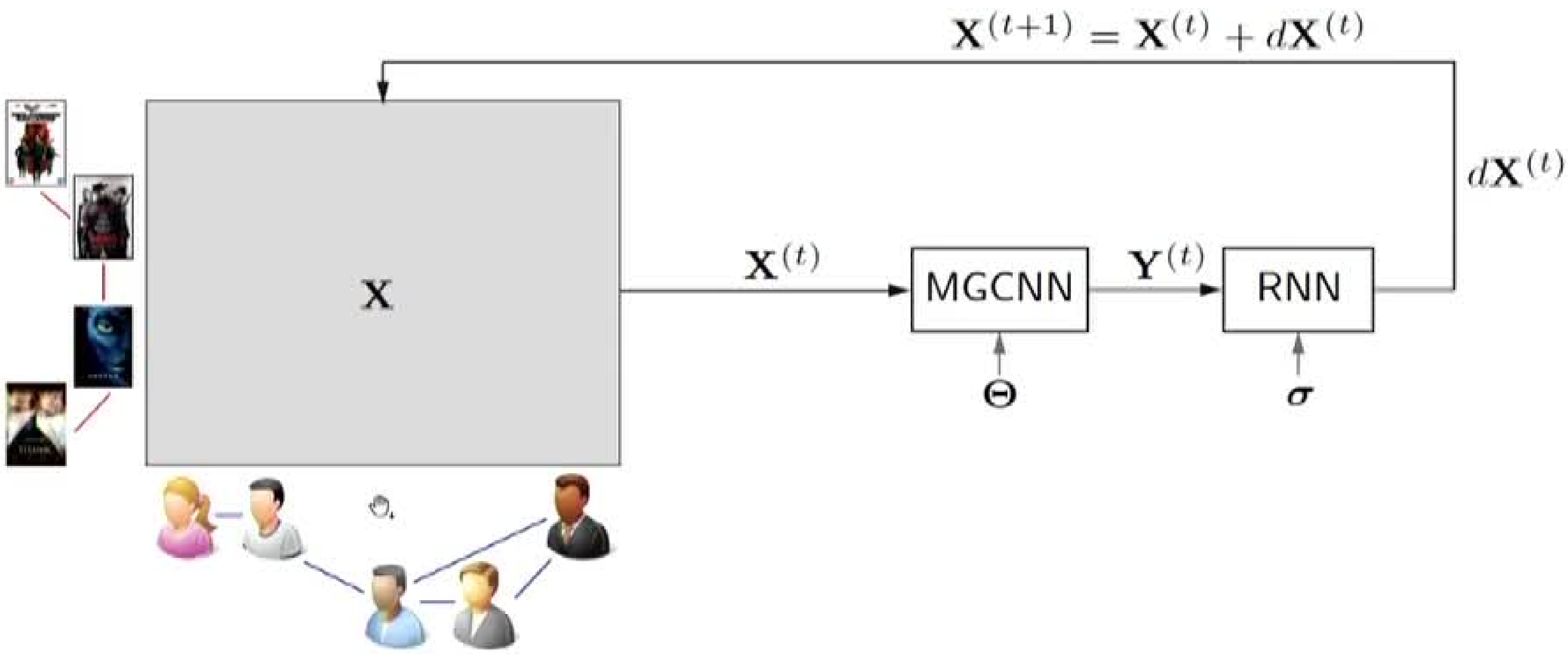
Multi-graph spectral filters



Examples of multi-graph spectral filters (shown is $|\tau(\lambda_c, \lambda_r)|$)

Monti, Bresson, Bronstein 2017

Matrix completion with Recurrent Multi-Graph CNN

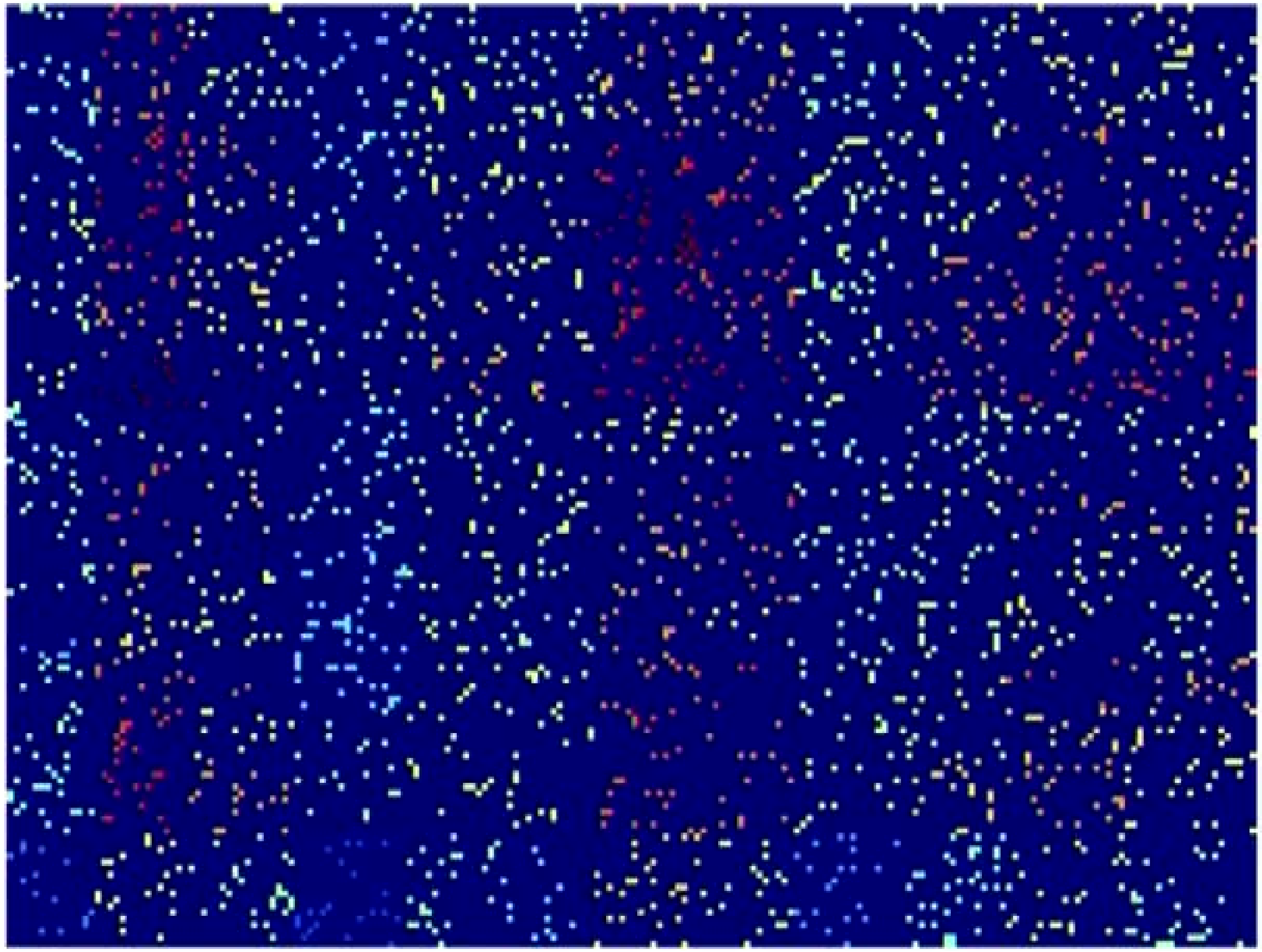


Recurrent multigraph CNN (RMCNN) architecture
for matrix completion

$$\min_{\mathbf{\Theta},\boldsymbol{\sigma}} \|\mathbf{X}^{(T)}_{\mathbf{\Theta},\boldsymbol{\sigma}}\|^2_{\mathcal{G}_\mathrm{r}} + \|\mathbf{X}^{(T)}_{\mathbf{\Theta},\boldsymbol{\sigma}}\|^2_{\mathcal{G}_\mathrm{c}} + \frac{\mu}{2}\|\mathbf{\Omega} \circ (\mathbf{X}^{(T)}_{\mathbf{\Theta},\boldsymbol{\sigma}} - \mathbf{A})\|^2_\mathrm{F}$$

Monti, Bresson, Bronstein 2017

Incremental updates with RNN

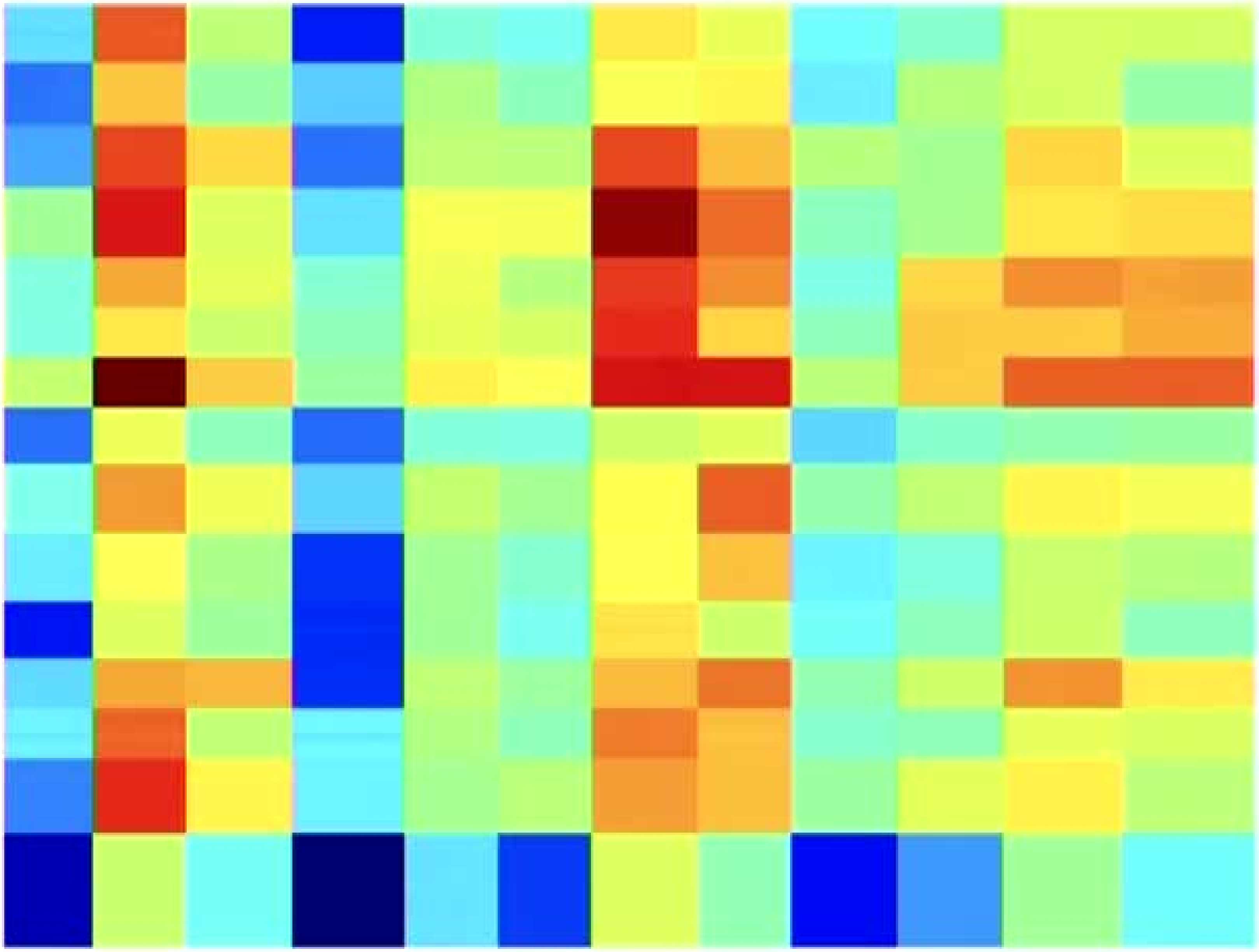


RMSE=2.26

Matrix completion results on a synthetic dataset (initialization)

Monti, Bresson, Bronstein 2017

Incremental updates with RNN



RMSE=0.01

Matrix completion results on a synthetic dataset after $t = 10$ instances

Monti, Bresson, Bronstein 2017

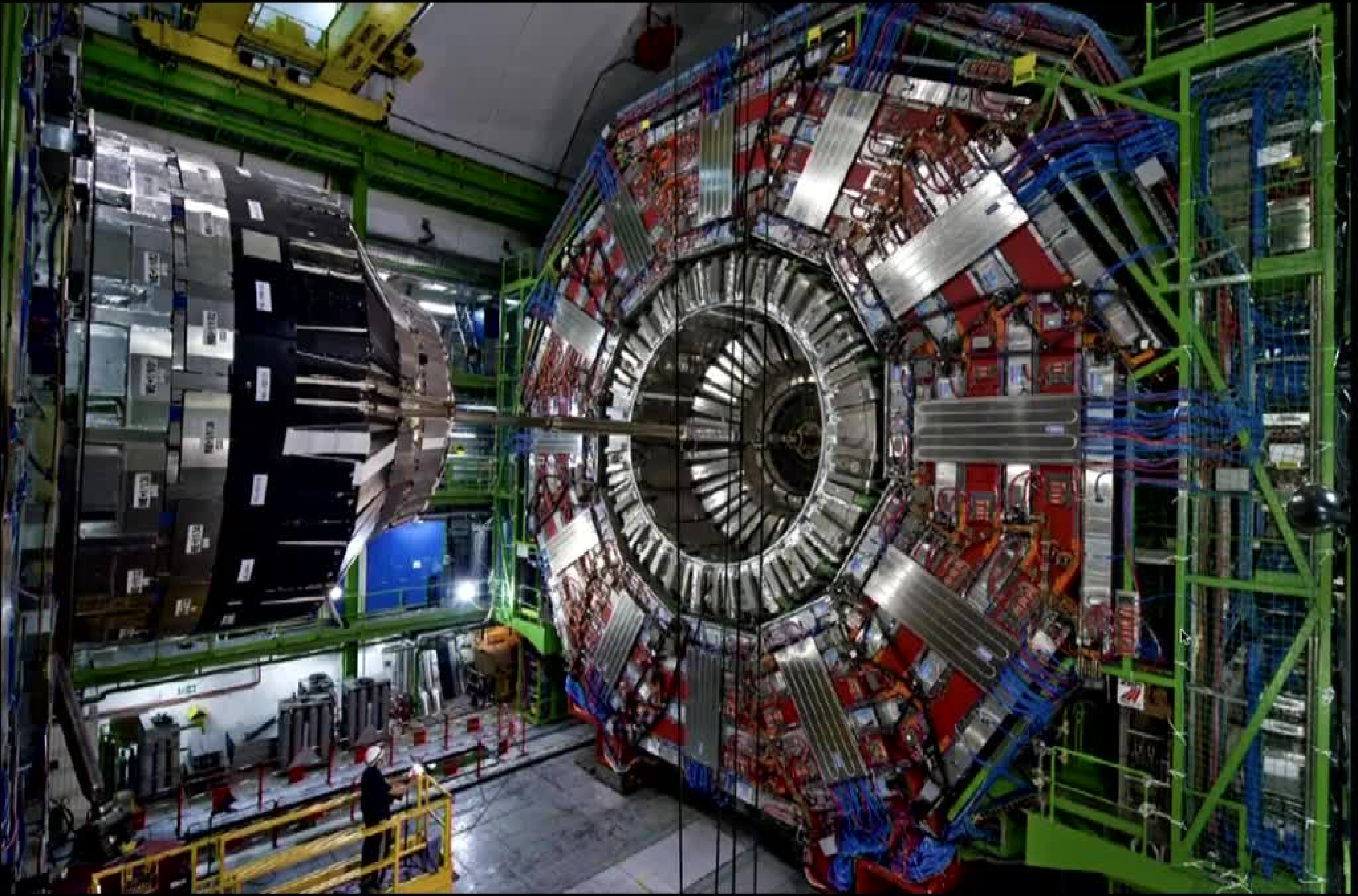
Matrix completion methods comparison

Method	MovieLens[1]	Flixster[2]	Douban[3]	Yahoo[4]
IMC[5]	1.653	–	–	–
GMC[6]	0.996	–	–	–
MC[7]	0.973	–	–	–
GRALS[8]	0.945	1.245	0.833	38.042
sRGCNN (Cheb)[9]	**0.929**	**0.926**	**0.801**	**22.415**
sRGCNN (Cayley)[10]	**0.922**	–	–	–
sRGCNN (Primal/Dual)[11]	**0.915**	**0.902**	**0.789**	**21.970**

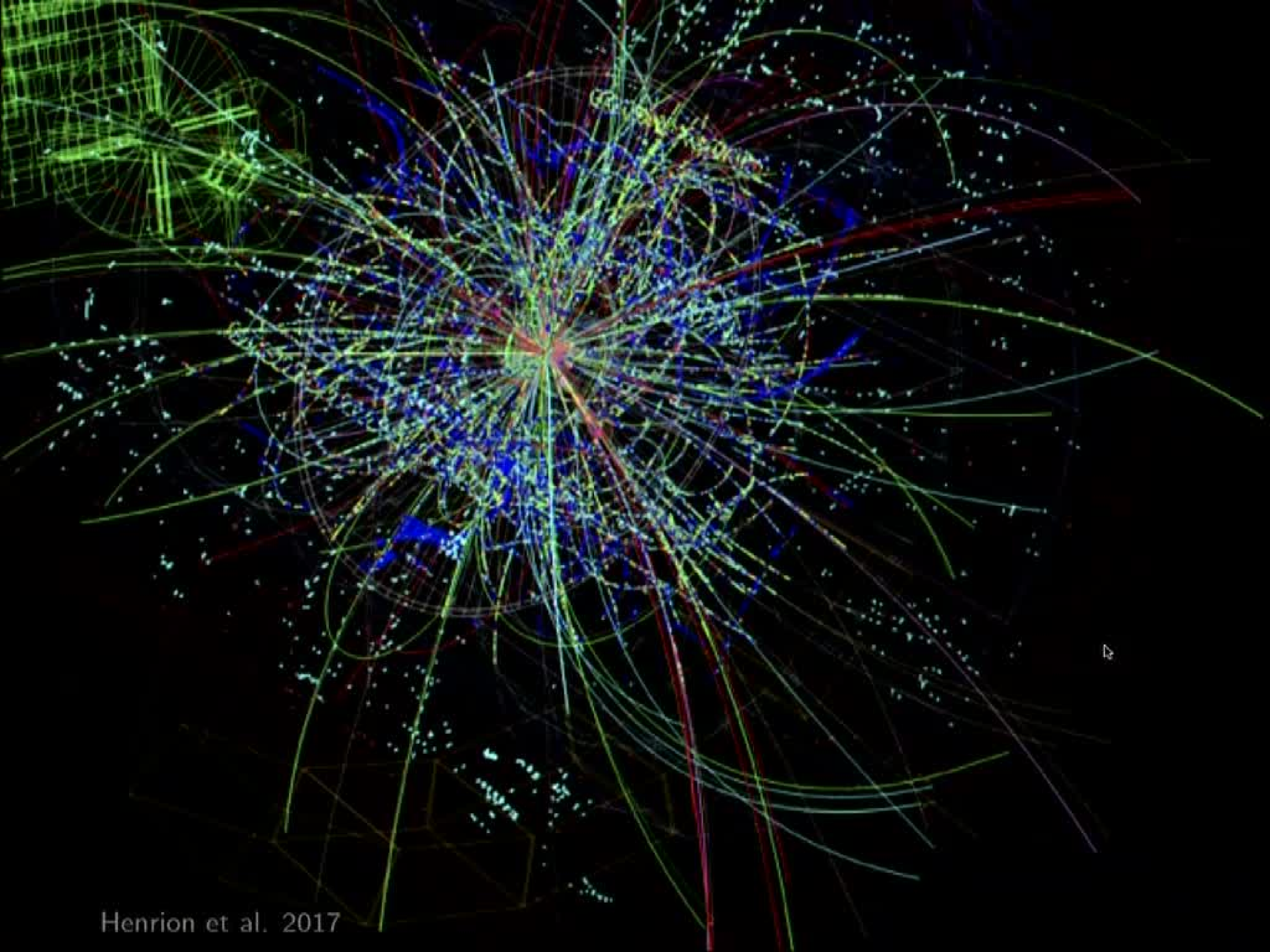
Accuracy (RMS error) of matrix completion methods on real data

Data: [1]Miller et al. 2003; [2]Jamali, Ester 2010; [3]Ma et al. 2011; [4]Dror et al. 2012
Methods: [5]Jain, Dhillon 2013; [6]Kalofolias et al. 2014; [7]Candès, Recht 2012; [8]Rao et al. 2015; [9]Monti, Bresson, Bronstein 2017; [10]Levie et al. 2017; [11]Monti et al. 2018

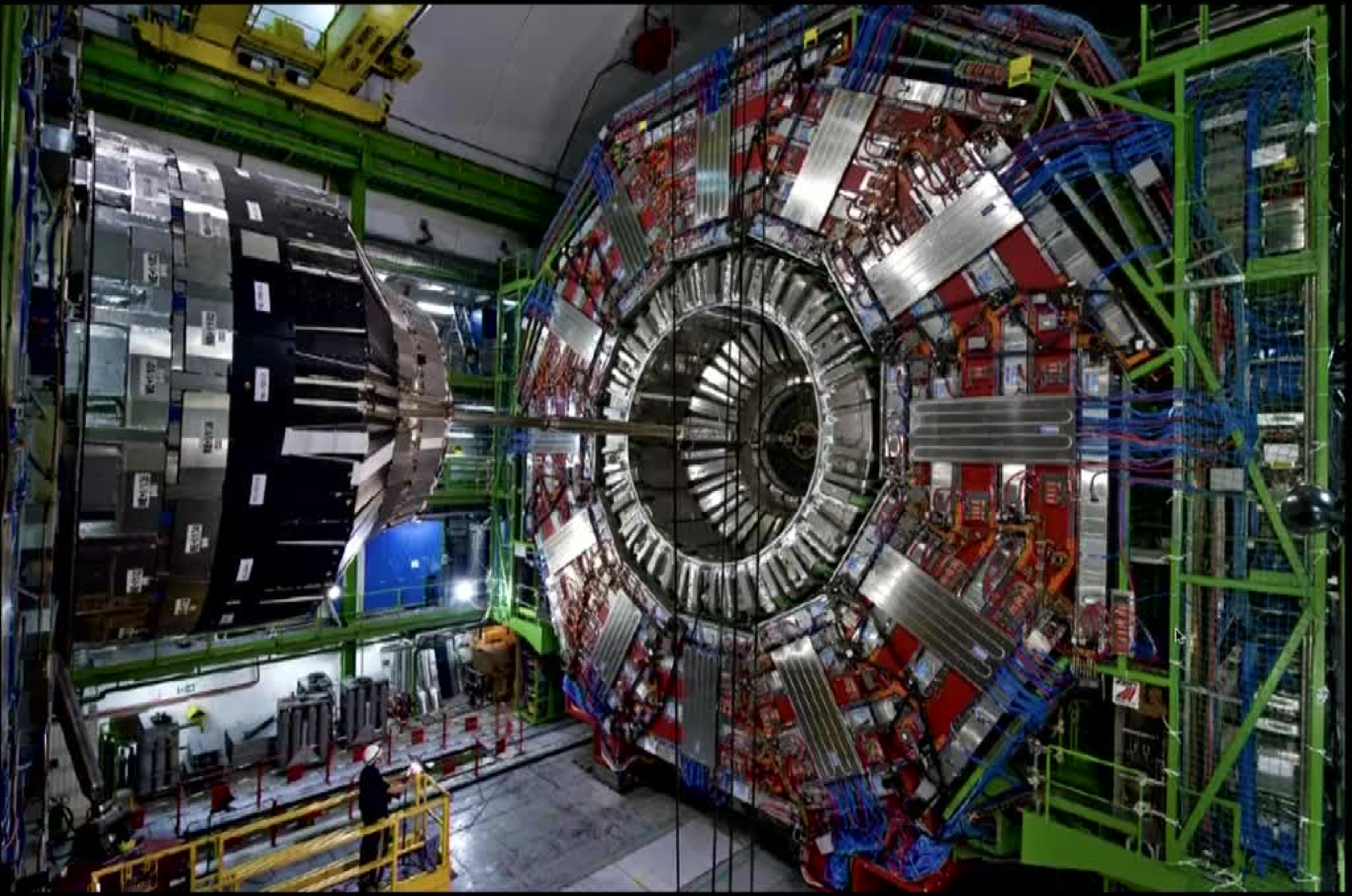
Application in
High-Energy Physics



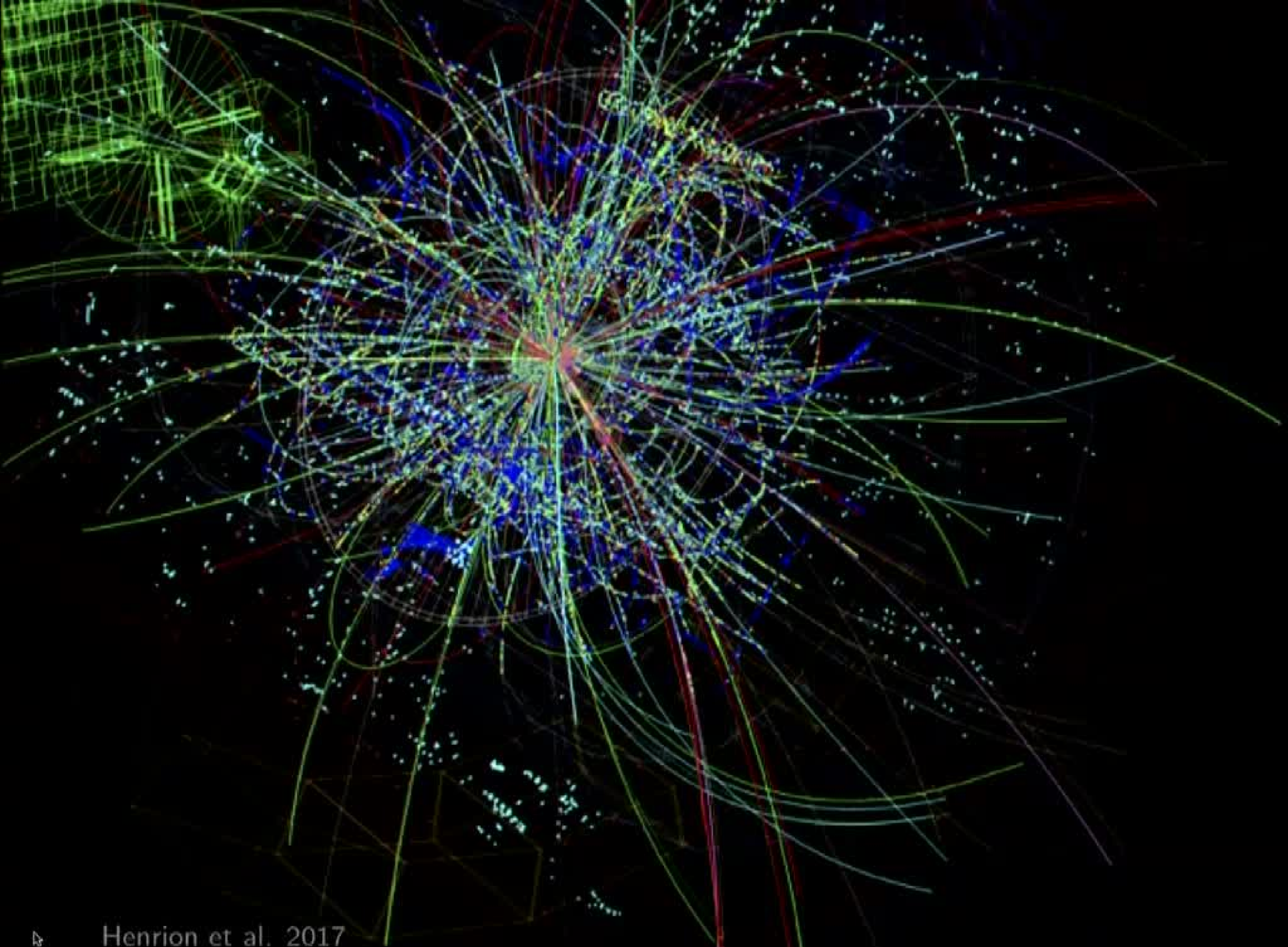
ATLAS detector in the Large Hadron Collider (CERN)



Henrion et al. 2017

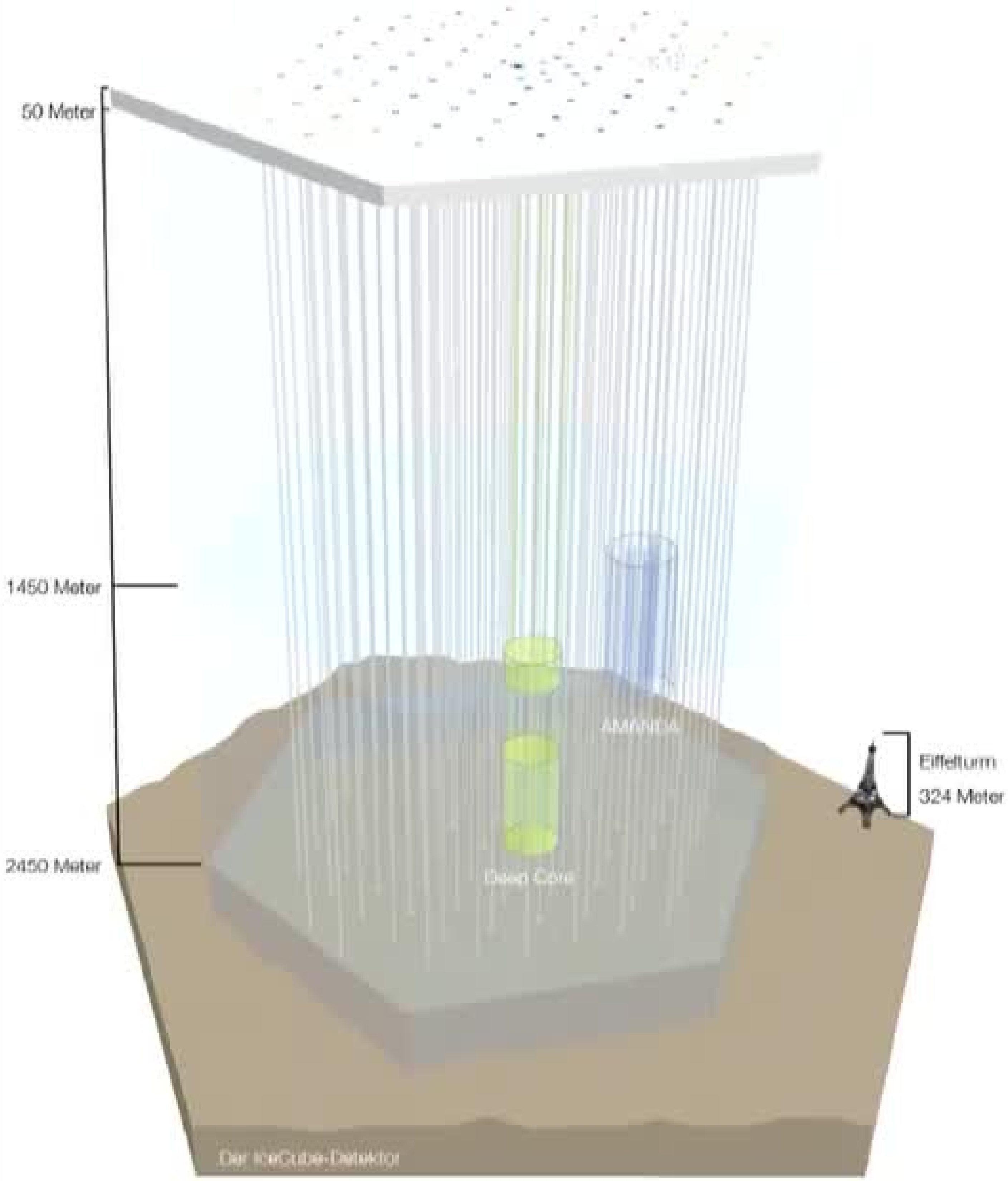


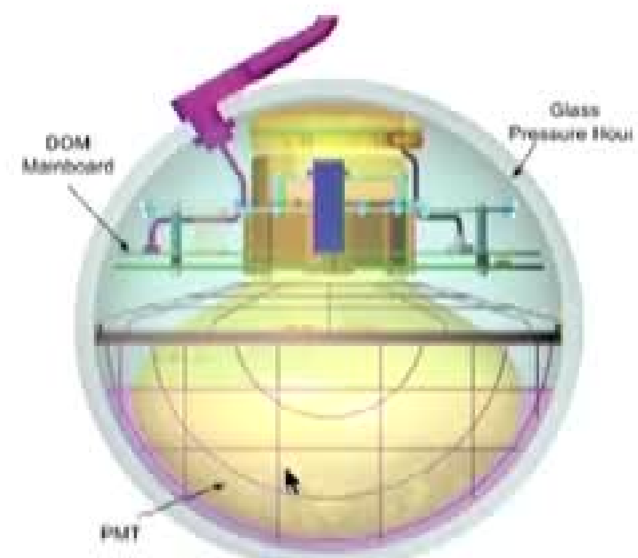
ATLAS detector in the Large Hadron Collider (CERN)



Henrion et al. 2017

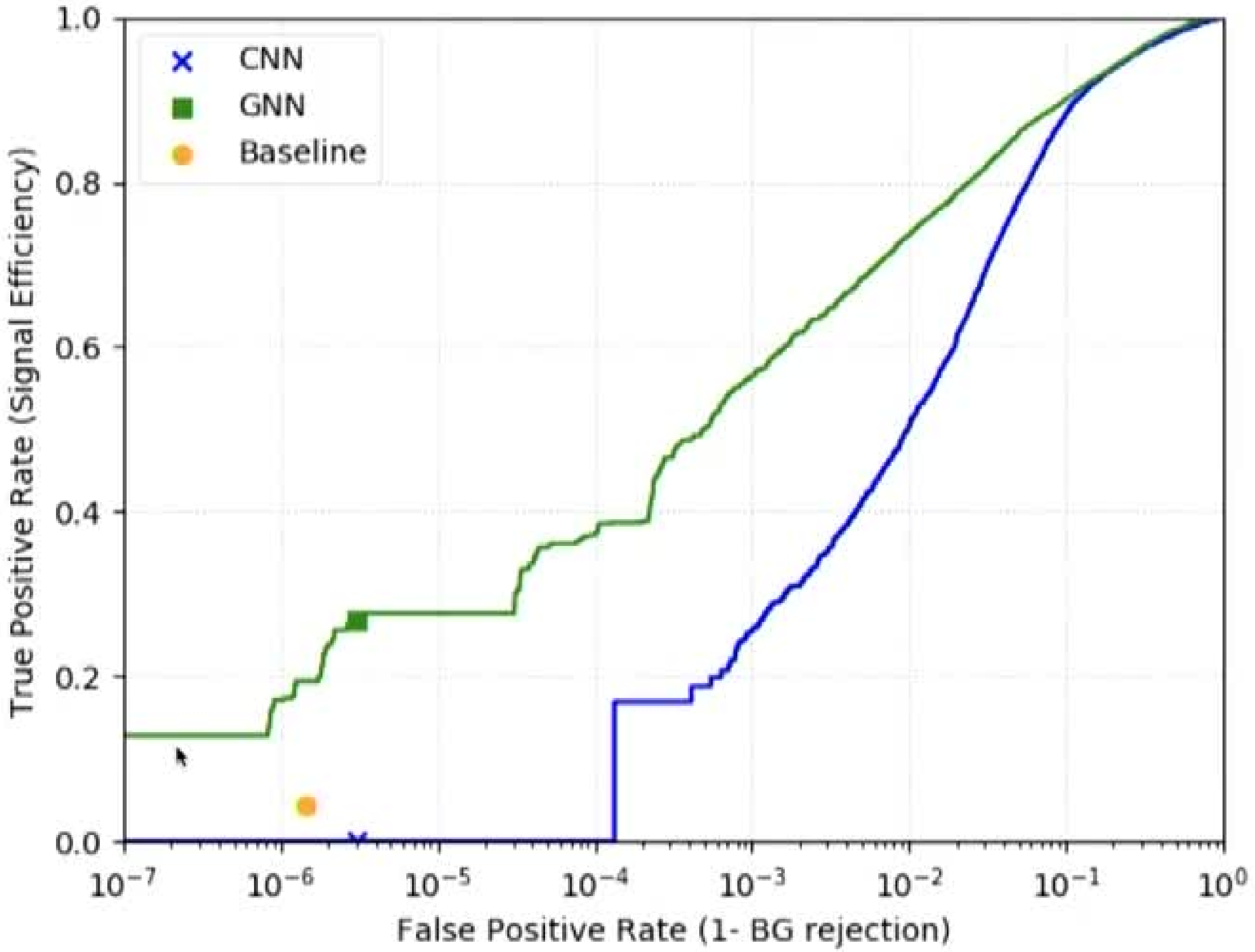
IceCube neutrino observatory





IceCube neutrino observatory

Neutrino detection in IceCube

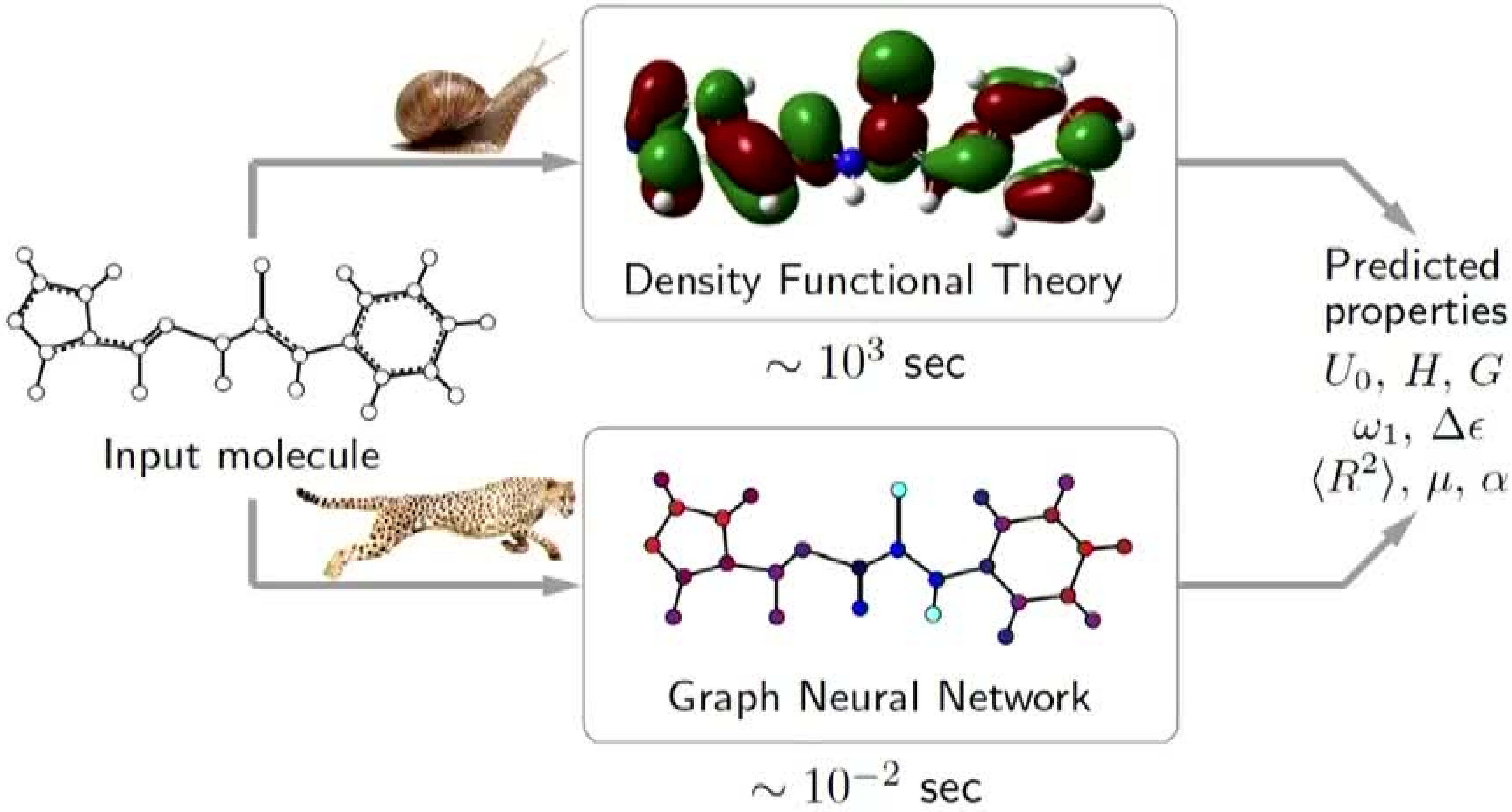


ROC curve comparing different methods for neutrino detection

Choma, Monti et al. 2018 for the IceCube collaboration
(joint work with NERSC+LBL, Berkeley)

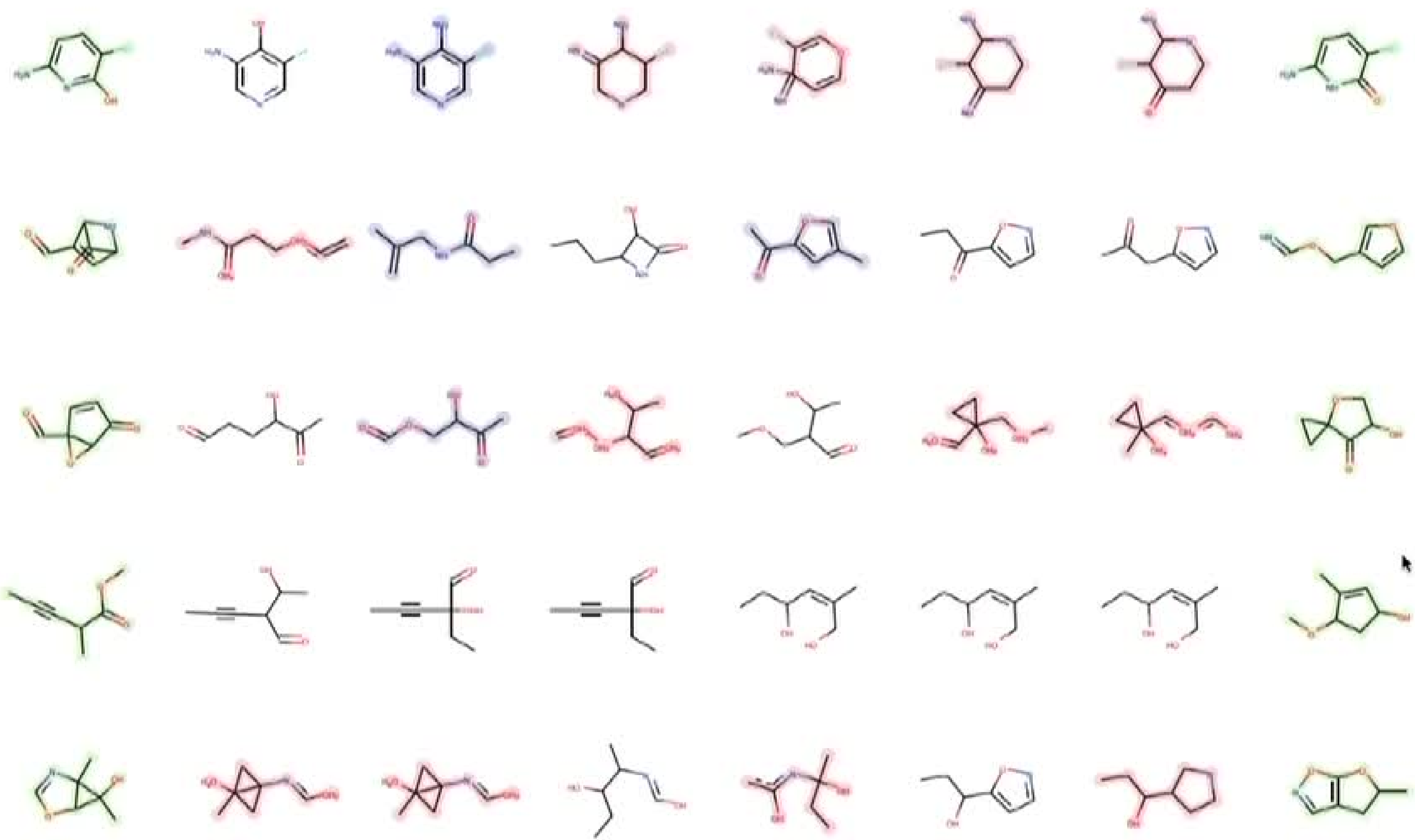
Applications in Chemistry and Drug Design

Molecule property prediction



Duvenaud et al. 2015; Gomez-Bombarelli et al. 2016; Gilmer et al. 2017

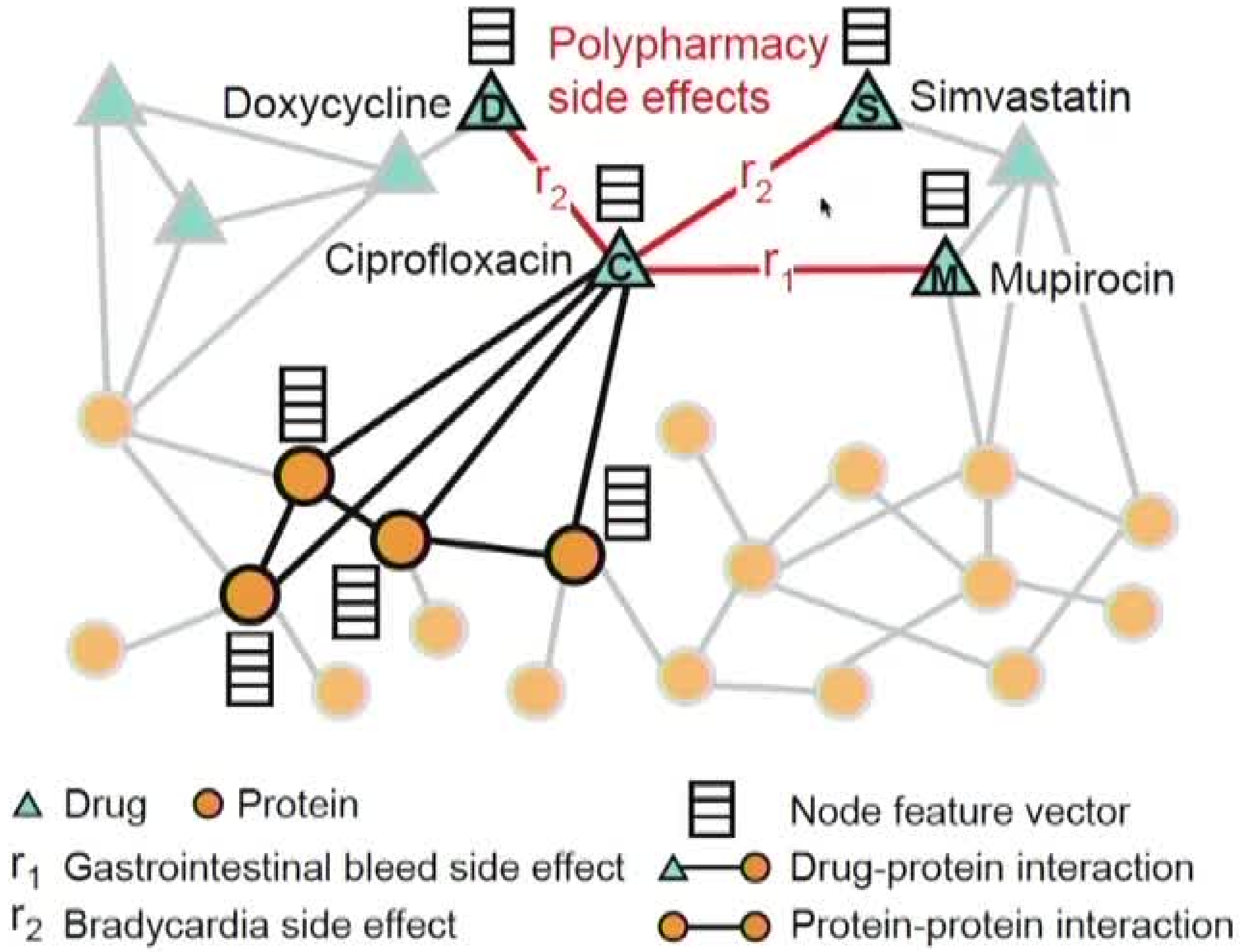
Molecule generation



Molecules generated with a graph VAE

Simonovsky, Komodakis 2017; You et al. 2018; De Cao, Kipf 2018 (MolGAN)

Polypharmacy and Drug repurposing

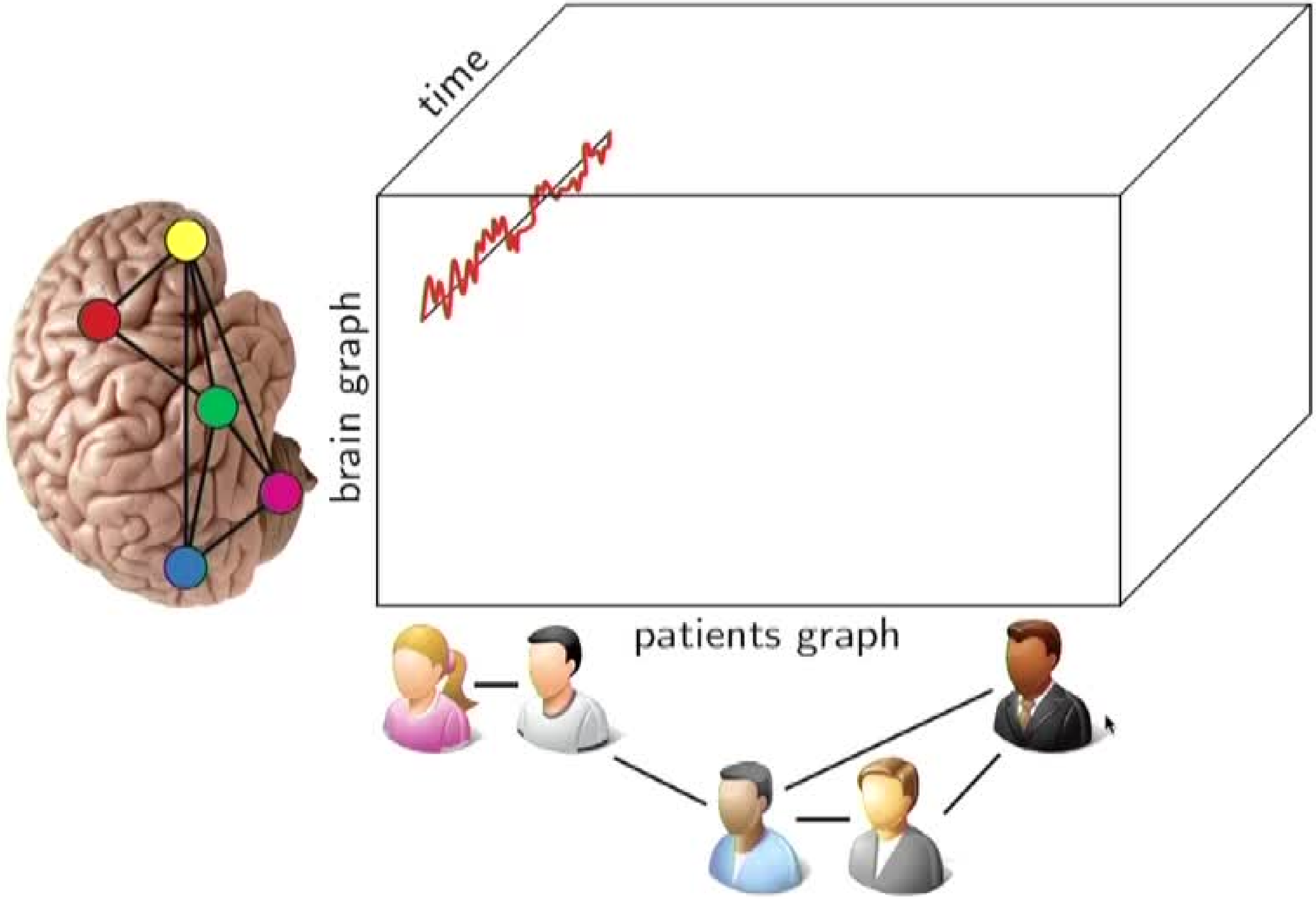


Prediction of side effects of drug combinations using deep learning on multimodal graph

Zitnik, Agrawal, Leskovec 2018

Other Applications

Brain imaging



Parisot et al. 2017

Adversarial noise

Targeted

stop speed limit 45

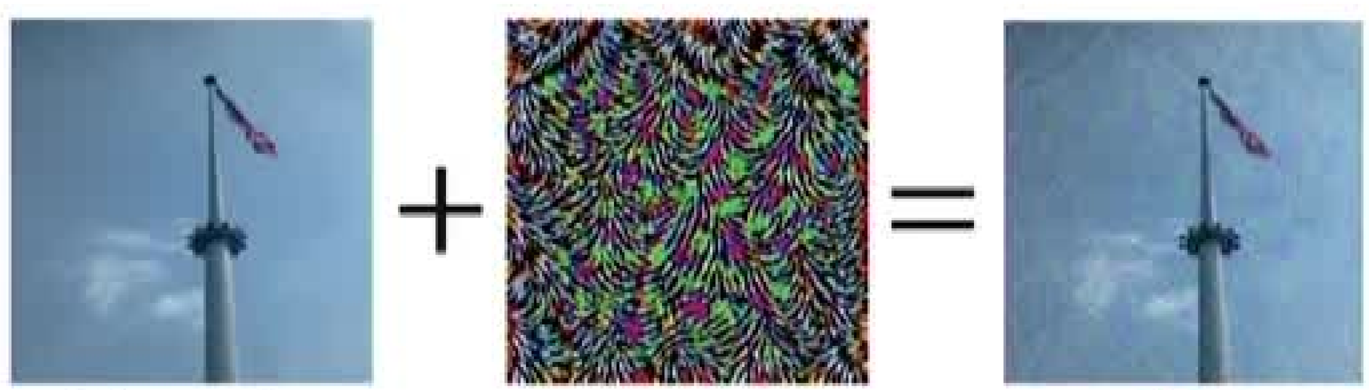
Eykholt et al. 2017

One pixel

cat bird

Su, Vargas, Sakurai 2018

Universal



flagpole dog

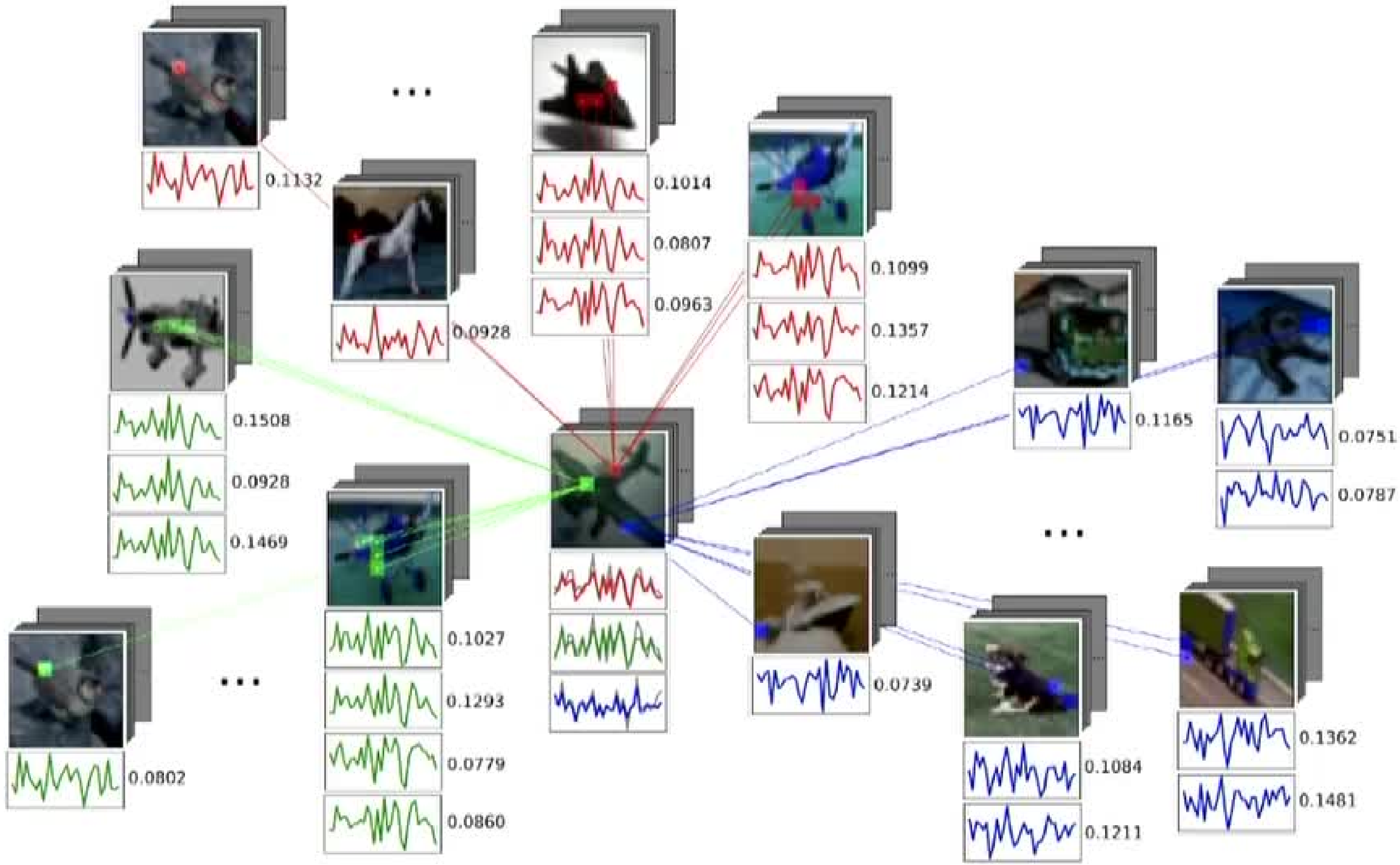
Moosavi-Dezfooli et al. 2017

Adversarial noise

Adversarial attack on a face recognition system makes it mistake faces with specially crafted glasses (top) for other people (bottom)

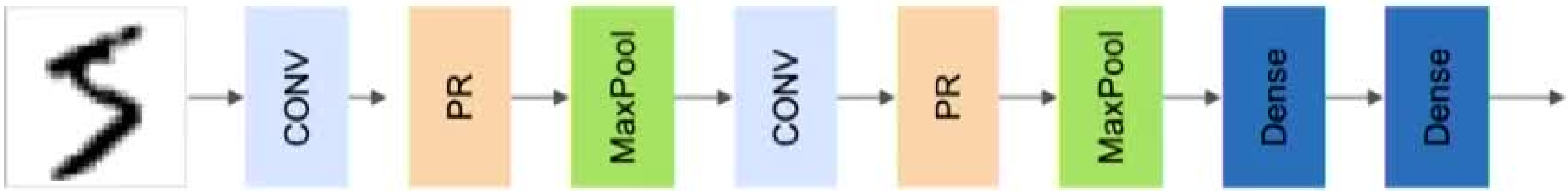
Sharif et al. 2016

Peer regularization

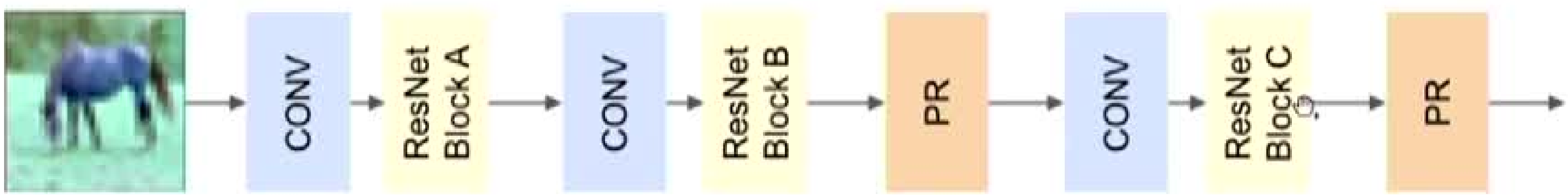


Svoboda et al. 2018

Peer regularization



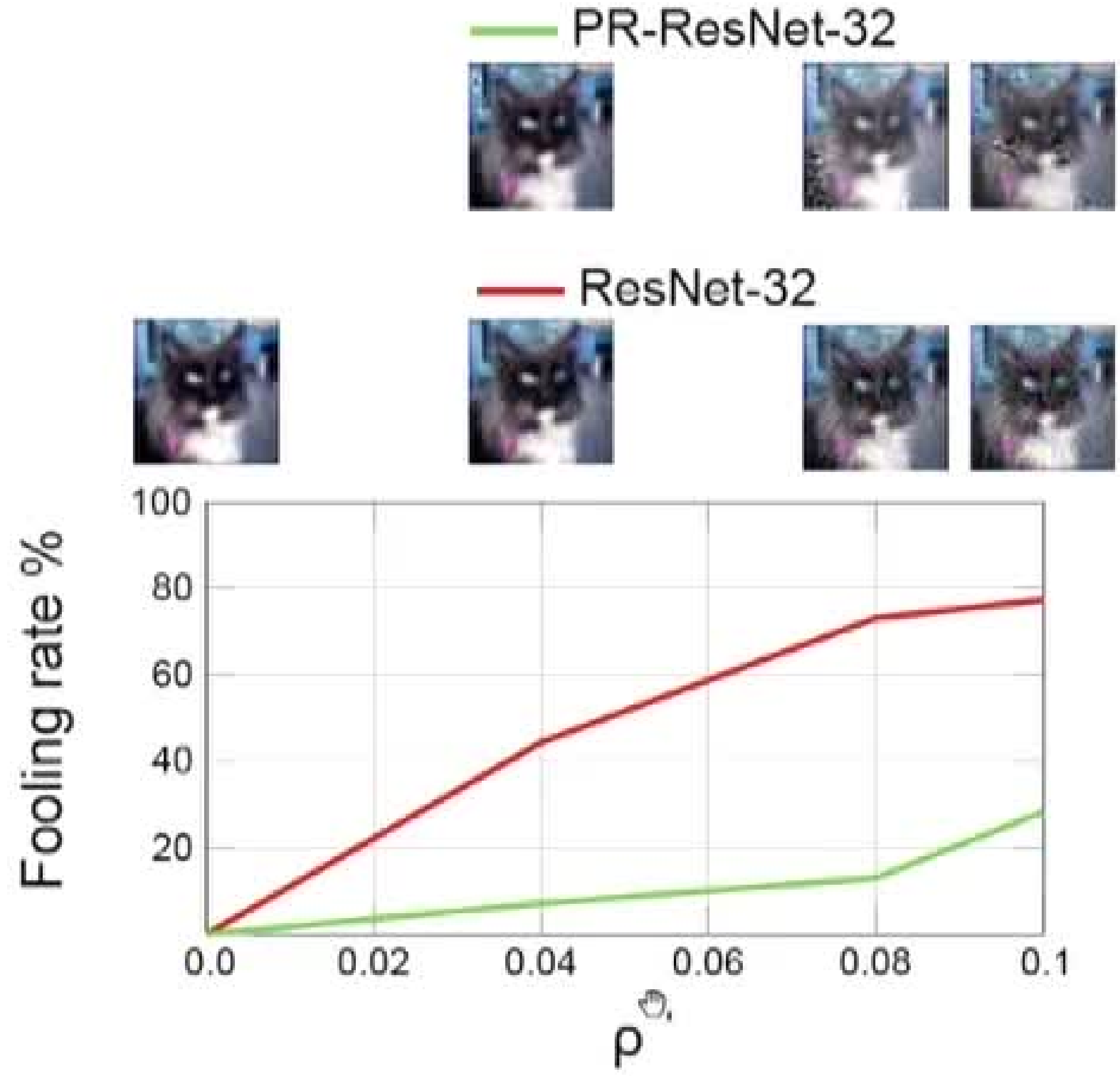
Peer-Regularized LeNet-5



Peer-Regularized ResNet-32

Svoboda et al. 2018

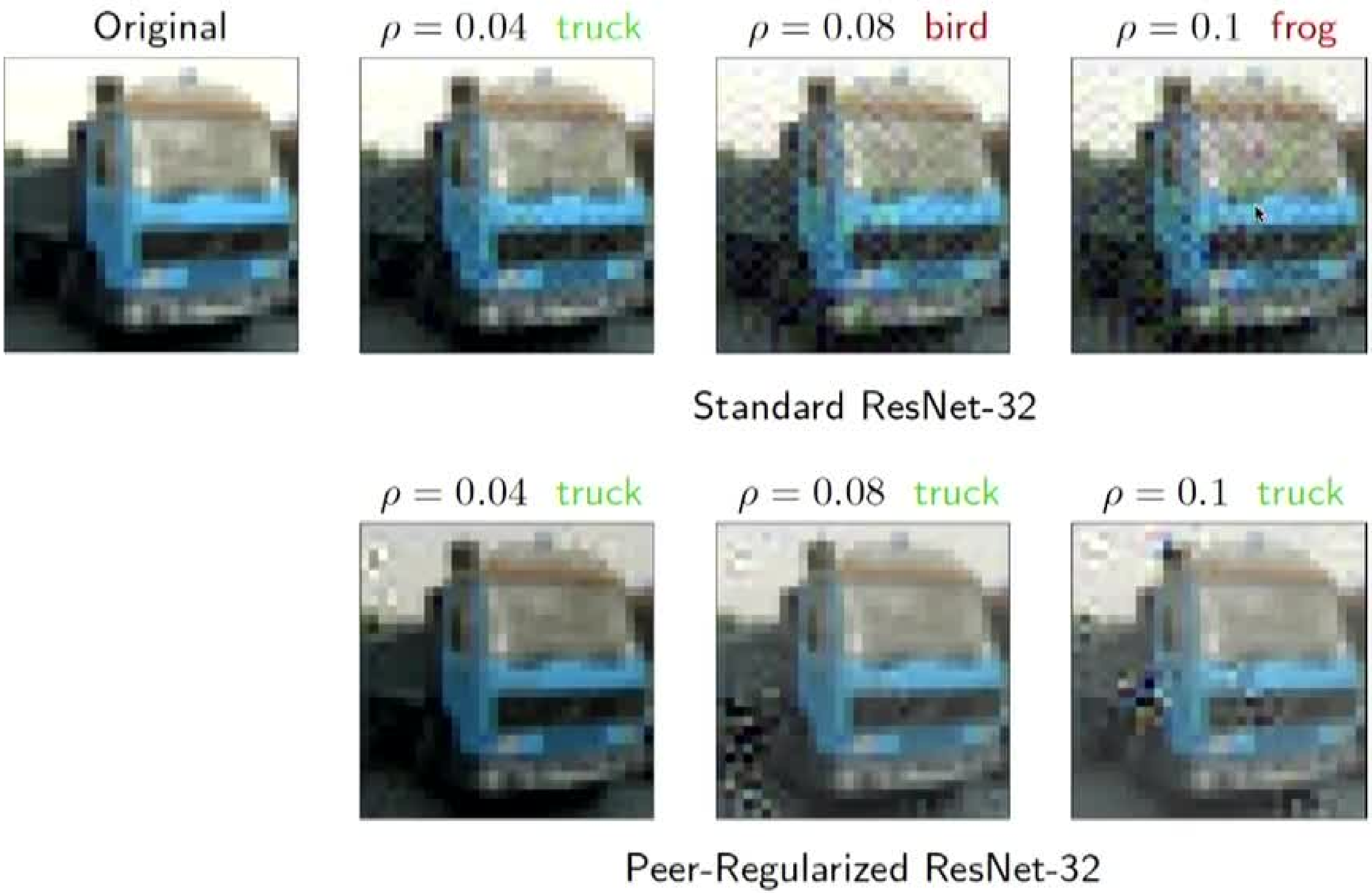
Universal perturbation



Fooling rate on CIFAR-10 for universal adversarial perturbation of different strength ρ

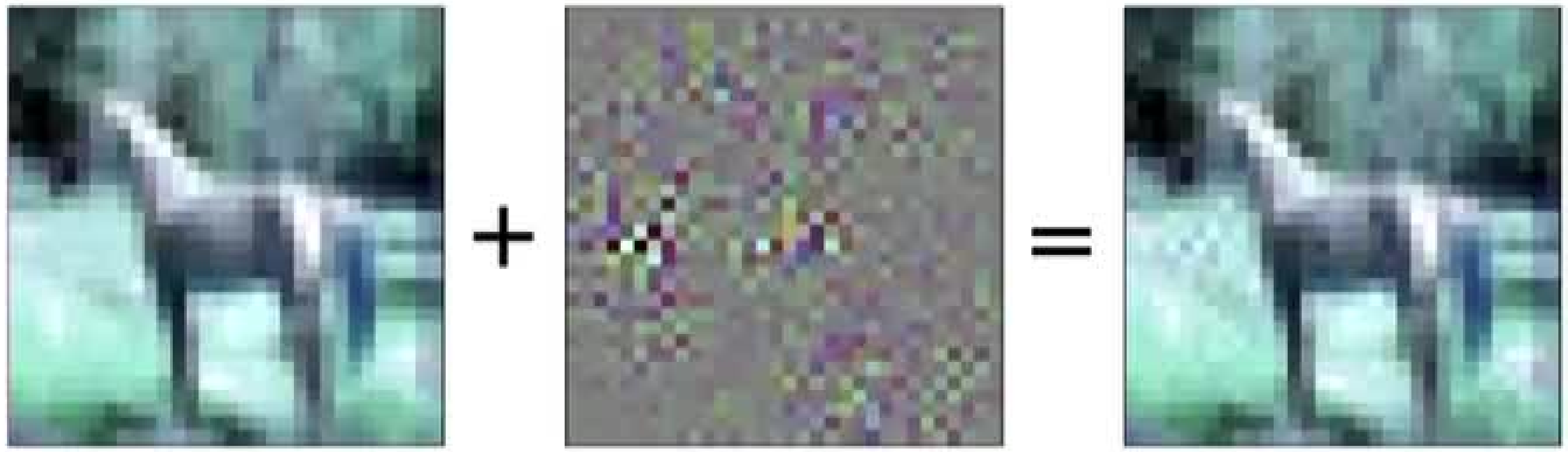
Svoboda et al. 2018; data: Krizhevsky 2009 (CIFAR-10)

Universal perturbation

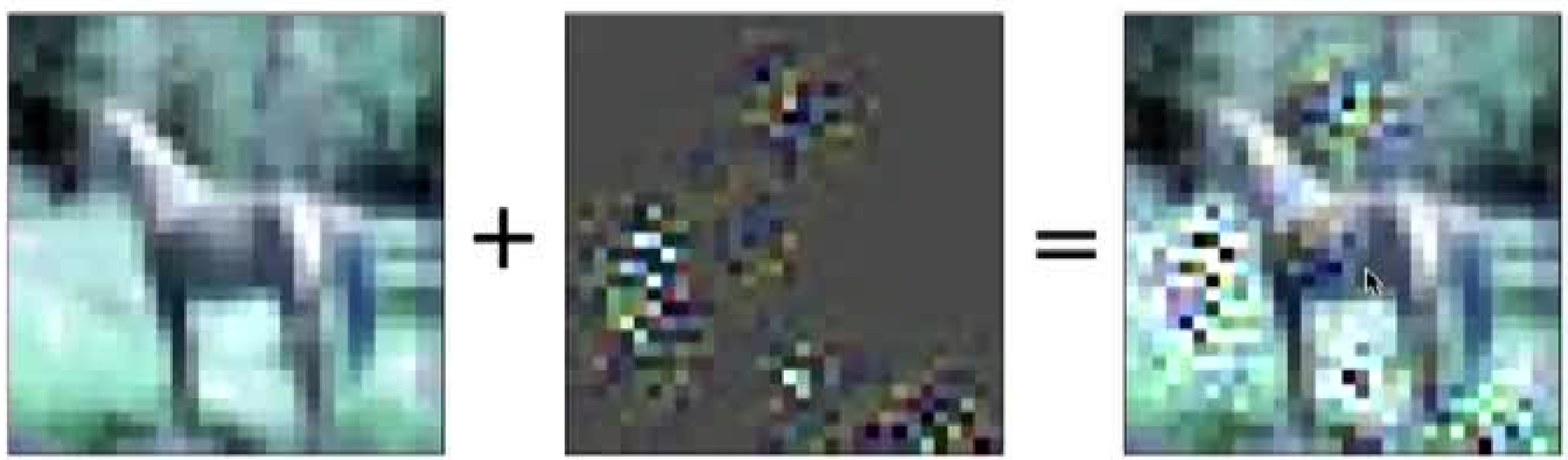


Svoboda et al. 2018; Method: Moosavi-Dezfooli et al. 2016

Non-targeted perturbation (gradient descent)



Standard ResNet-32



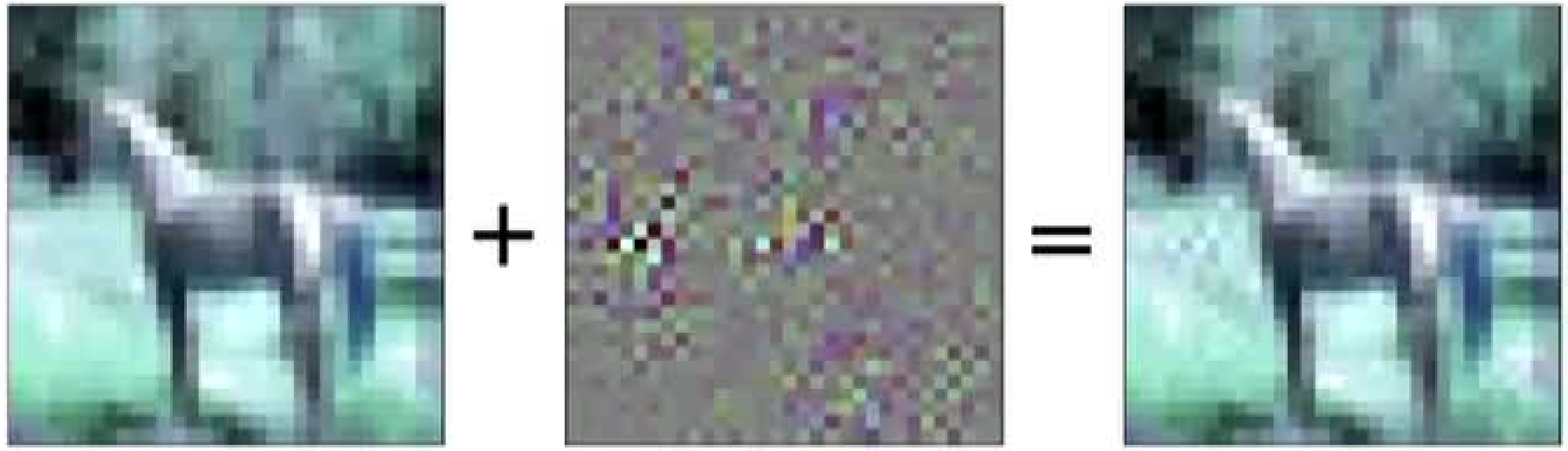
Peer-Regularized ResNet-32

Much stronger noise is needed to fool PeerNet!

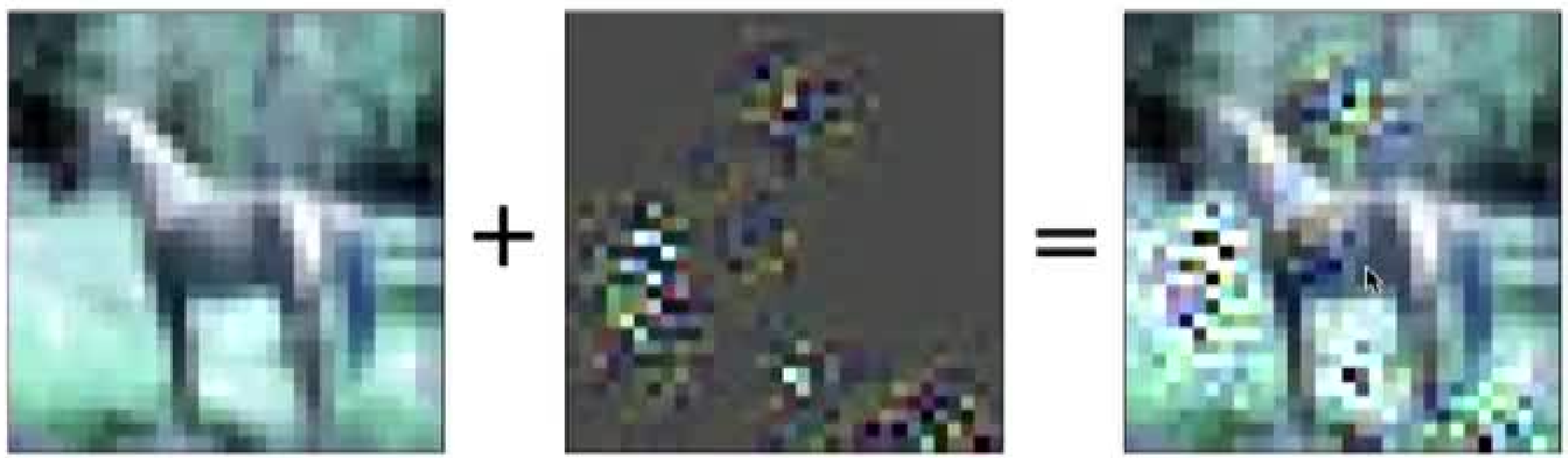
Svoboda et al. 2018; Method: Rauber et al. 2017

Vosoghi et al. 2018

Non-targeted perturbation (gradient descent)



Standard ResNet-32



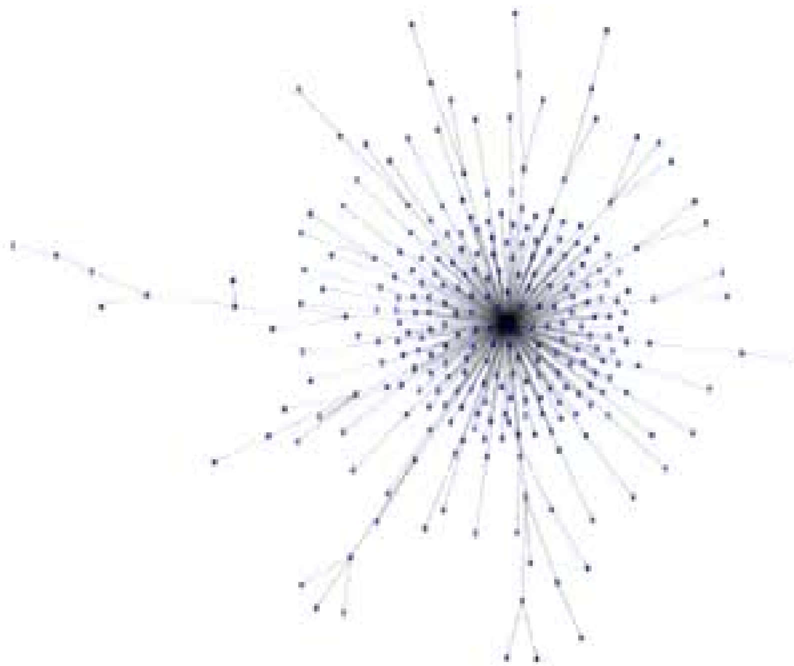
Peer-Regularized ResNet-32

Much stronger noise is needed to fool PeerNet!

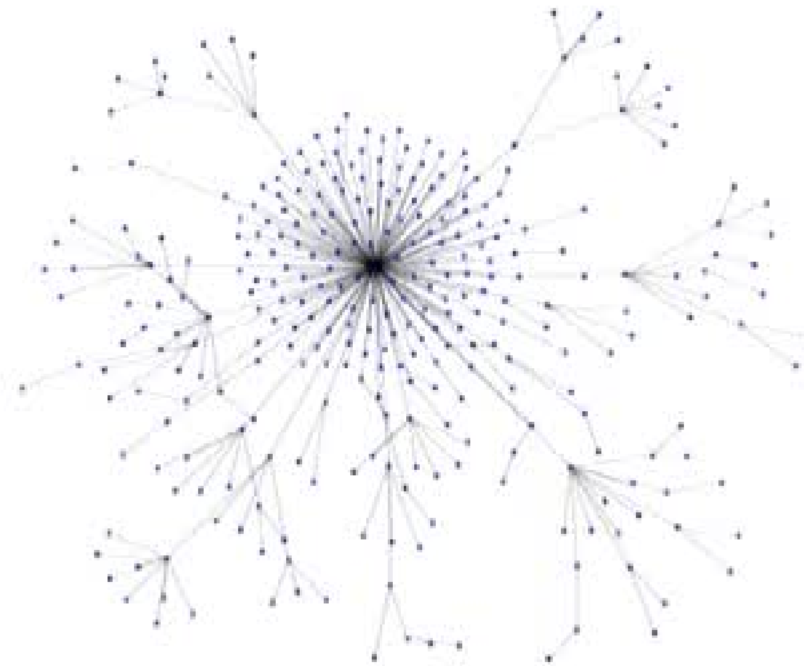
Svoboda et al. 2018; Method: Rauber et al. 2017

Vosoghi et al. 2018

Fake news detection



Fake news



True news

We are hiring in Switzerland and UK!

Fabula AI proprietary technology. Multiple patents pending

Conclusions

- Geometric Priors exist beyond Euclidean domains
- They can be exploited by modeling locality and weight sharing
- Spectral and Spatial models obtain locality and sharing through different routes
- Resulting neural models related to attention mechanisms
- Challenge: Scaling up models to millions of nodes
- Large areas of application: physical sciences, machine translation, graphics, relational learning