ICML
International Conference
On Machine Learning



{pavlo.melnyk, michael.felsberg, marten.wadenback}@liu.se

Steerable 3D Spherical Neurons

Pavlo Melnyk, Michael Felsberg, Mårten Wadenbäck

Computer Vision Laboratory, Linköping University, Sweden

LINKÖPING UNIVERSITY

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

Motivation

- 3D point cloud processing with NNs
  - Classification: given a point cloud $\boldsymbol{X}$, predict its class

Motivation

- 3D point cloud processing with NNs
  - Classification: given a point cloud $\boldsymbol{X}$, predict its class

- Rotation equivariant (steerable) models: $f(\boldsymbol{RX}) = \rho(\boldsymbol{R})\, f(\boldsymbol{X})$

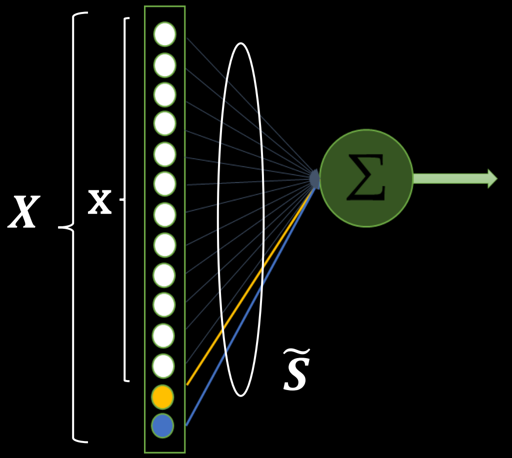
- Rotation invariant predictions: $f(\boldsymbol{RX}) = f(\boldsymbol{X})$

Motivation

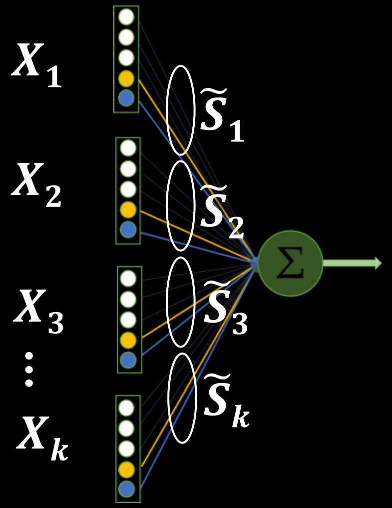
- 3D point cloud processing with NNs
  - Classification: given a point cloud $\boldsymbol{X}$, predict its class
- Rotation equivariant (steerable) models: $f(\boldsymbol{RX}) = \rho(\boldsymbol{R})\, f(\boldsymbol{X})$
- Rotation invariant predictions: $f(\boldsymbol{RX}) = f(\boldsymbol{X})$
- How to achieve this with models comprised of spherical neurons?

Spherical neurons

- A neuron with a *spherical* decision surface
  - the *hypersphere neuron* (Banarer et al. 2003) or the *geometric neuron* (Melnyk et al. 2021)



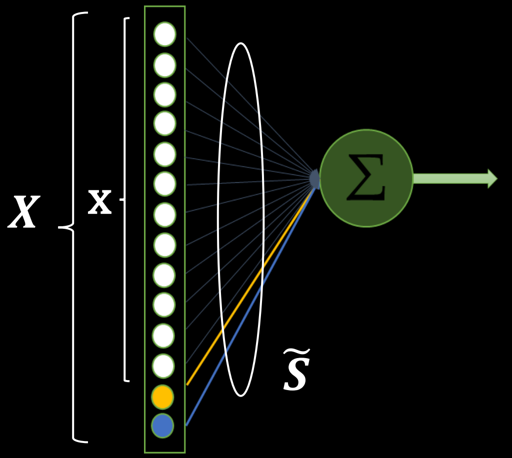
$$z = X^{\mathrm{T}}\tilde{S}$$



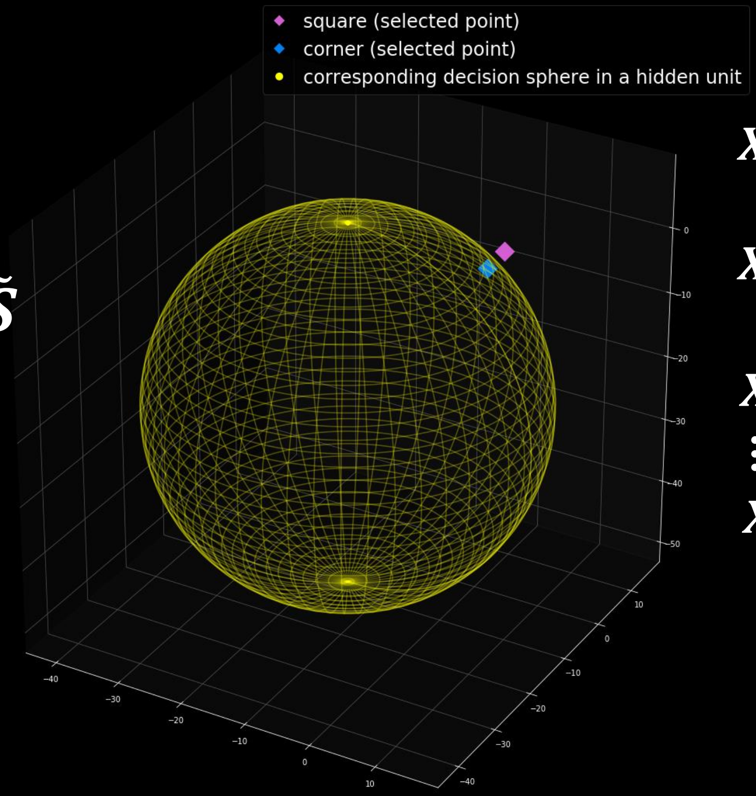
$$z = \sum_{i=1}^{K} X_i^{\mathrm{T}} \tilde{S}_i$$

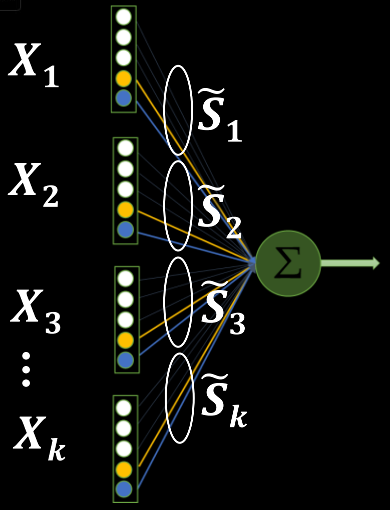
Spherical neurons

- A neuron with a *spherical* decision surface
  - the *hypersphere neuron* (Banarer et al. 2003) or the *geometric neuron* (Melnyk et al. 2021)



$$z = X^{\mathrm{T}}\tilde{S}$$





$$z = \sum_{i=1}^{K} X_i^{\mathrm{T}}\tilde{S}_i$$

How to make a spherical neuron steer?

- In 3D, $f(x, y, z)$ steers if

$$f^{\boldsymbol{R}}(x, y, z) = \sum_{j=1}^{M} v_j(\boldsymbol{R}) f^{\boldsymbol{R}_j}(x, y, z)$$

Freeman et al. (1991), Knutsson et al. (1992)

How to make a spherical neuron steer?

- In 3D, *f*(*x, y, z*) steers if

$$f^{\boldsymbol{R}}(x, y, z) = \sum_{j=1}^{M} v_j(\boldsymbol{R}) f^{\boldsymbol{R}_j}(x, y, z)$$

Freeman et al. (1991), Knutsson et al. (1992)

- Derive for a spherical neuron $f(\boldsymbol{X}) = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{S}$

How to make a spherical neuron steer?

- In 3D, $f(x, y, z)$ steers if

1. Number of basis functions

$$f^{\boldsymbol{R}}(x, y, z) = \sum_{j=1}^{M} v_j(\boldsymbol{R}) f^{\boldsymbol{R}_j}(x, y, z)$$

original function in orientation $\boldsymbol{R}$

Freeman et al. (1991), Knutsson et al. (1992)

- Derive for a spherical neuron $f(\boldsymbol{X}) = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{S}$

How to make a spherical neuron steer?

- In 3D, $f(x, y, z)$ steers if

1. Number of basis functions

basis function orientations

$$f^{\boldsymbol{R}}(x, y, z) = \sum_{j=1}^{M} v_j(\boldsymbol{R}) f^{\boldsymbol{R}_j}(x, y, z)$$

original function in orientation $\boldsymbol{R}$

2. Basis functions

Freeman et al. (1991), Knutsson et al. (1992)

- Derive for a spherical neuron $f(\boldsymbol{X}) = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{S}$

How to make a spherical neuron steer?

- In 3D, $f(x, y, z)$ steers if

3. Interpolation coefficients

1. Number of basis functions

basis function orientations

$$f^{\boldsymbol{R}}(x, y, z) = \sum_{j=1}^{M} v_j(\boldsymbol{R})\, f^{\boldsymbol{R}_j}(x, y, z)$$

original function in orientation $\boldsymbol{R}$

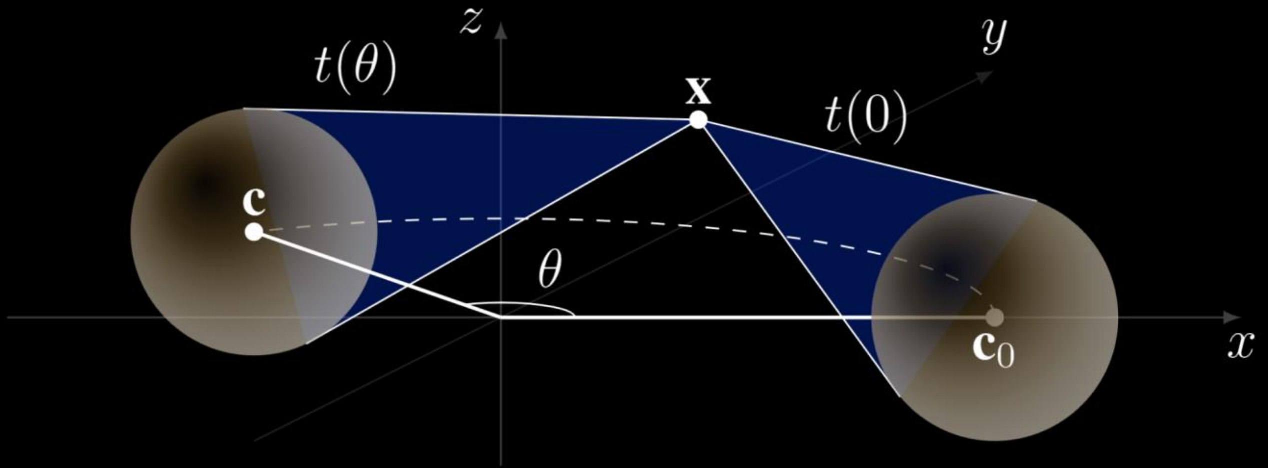
2. Basis functions

Freeman et al. (1991), Knutsson et al. (1992)

- Derive for a spherical neuron $f(\boldsymbol{X}) = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{S}$

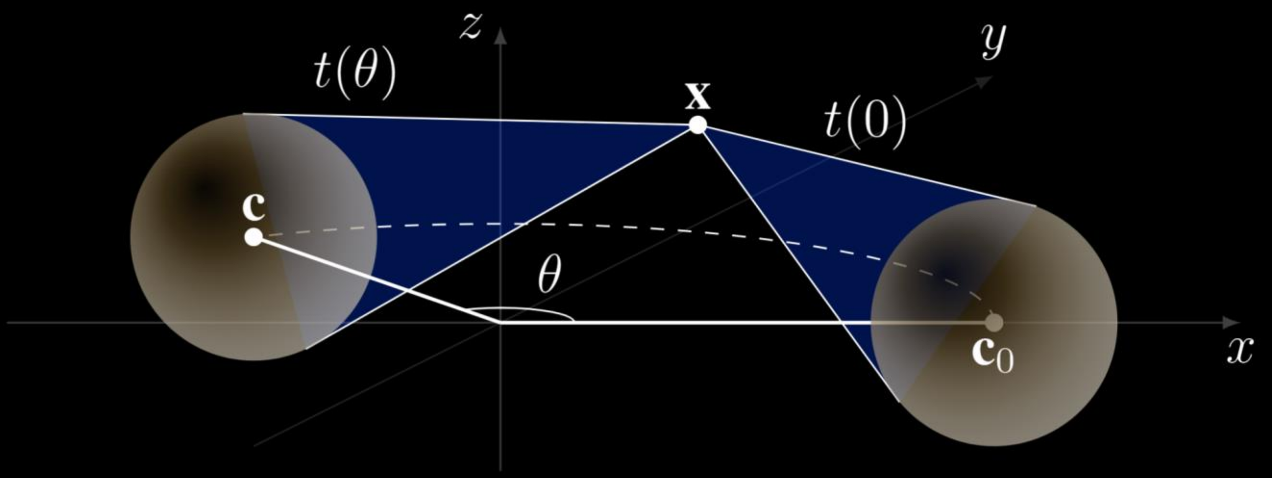
1. Min # basis functions, *M*

- *Spherical neuron* activation, $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{S}$ vs. Its rotated version activation, $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{S}'$



1. Min # basis functions, *M*

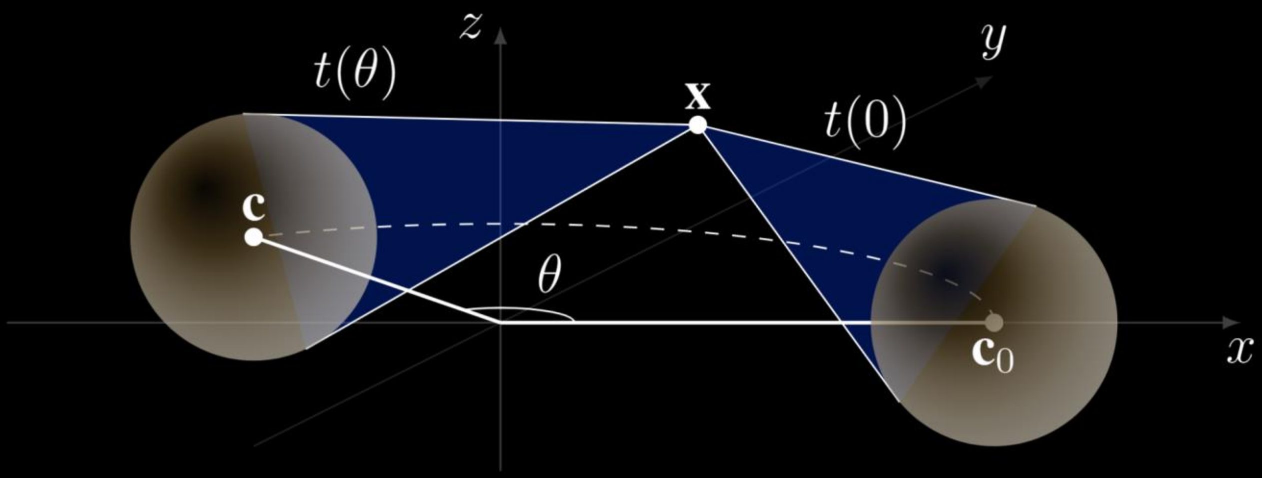
- *Spherical neuron* activation, $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{S}$ vs. Its rotated version activation, $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{S}'$



- Vary by (up to) first-degree (N=1) spherical harmonics in the rotation angle θ

1. Min # basis functions, *M*

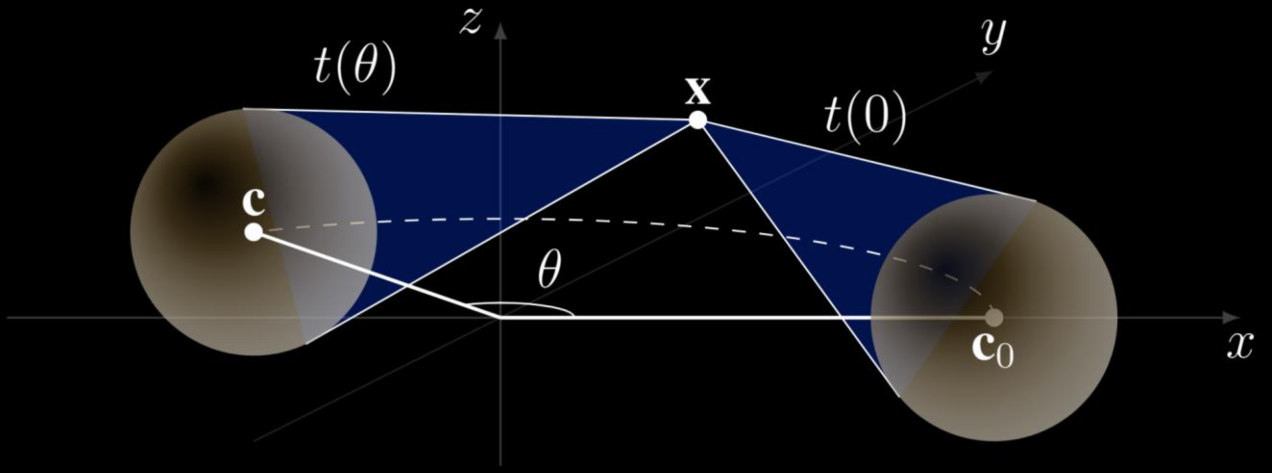
- *Spherical neuron* activation, $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{S}$ vs. Its rotated version activation, $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{S}'$



- Vary by (up to) first-degree (*N*=1) spherical harmonics in the rotation angle θ
- In any dimension! (Theorem 4.1)

1. Min # basis functions, *M*

- *Spherical neuron* activation, $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{S}$ vs. Its rotated version activation, $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{S}'$



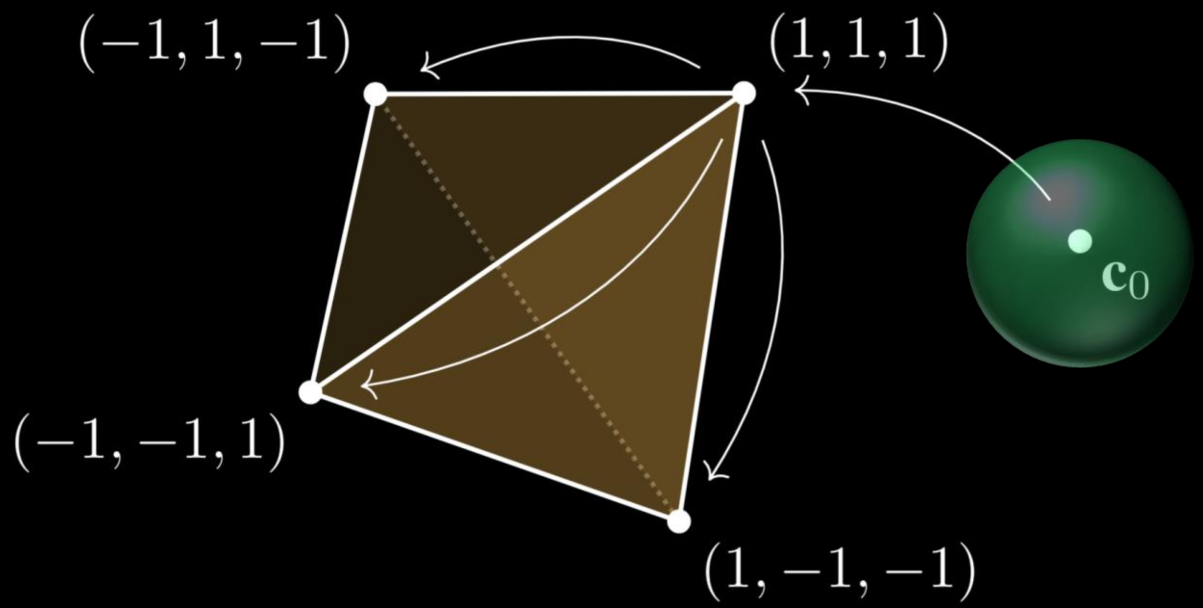
- Vary by (up to) first-degree (N=1) spherical harmonics in the rotation angle θ
- In any dimension! (Theorem 4.1)
- In 3D, we need only $M = (N+1)^2 = 4$ basis functions (Theorem 4 by Freeman et al. (1991))

2. The basis functions

- The original sphere with center $\mathbf{c_0}$
- The four spheres must be spaced in 3D *equally*

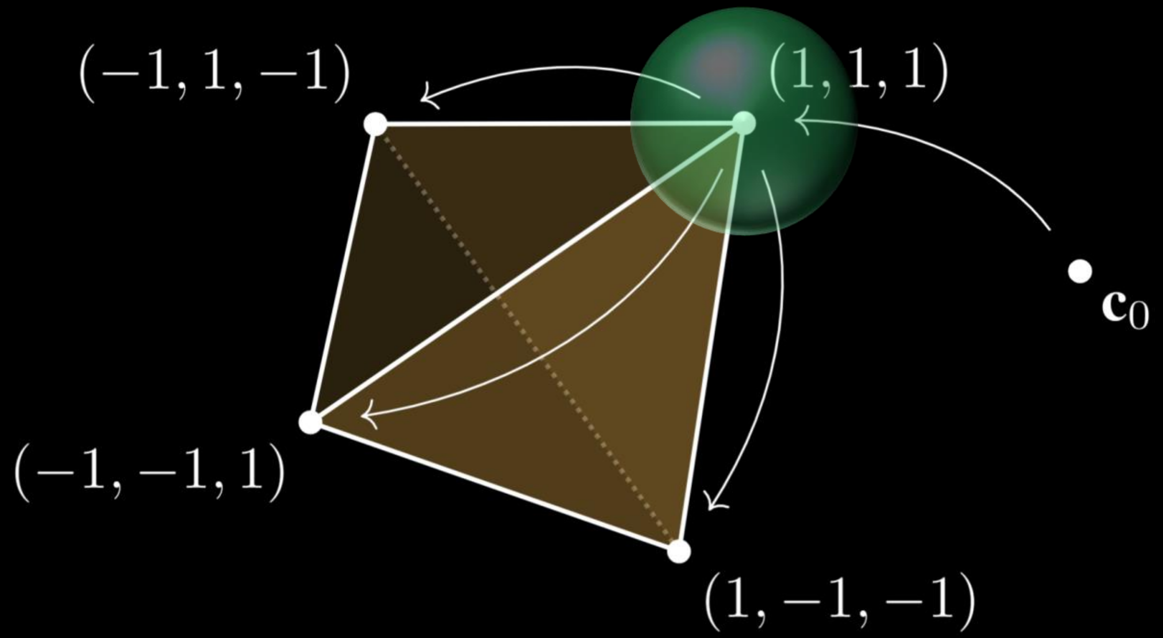
2. The basis functions

- The original sphere with center $\mathbf{c_0}$
- The four spheres must be spaced in 3D *equally*
- Form a *regular tetrahedron* – *spherical filter bank*:



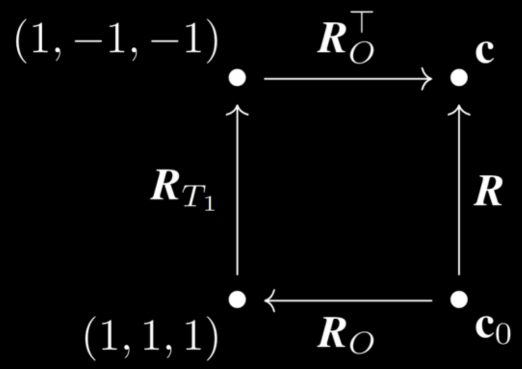
2. The basis functions

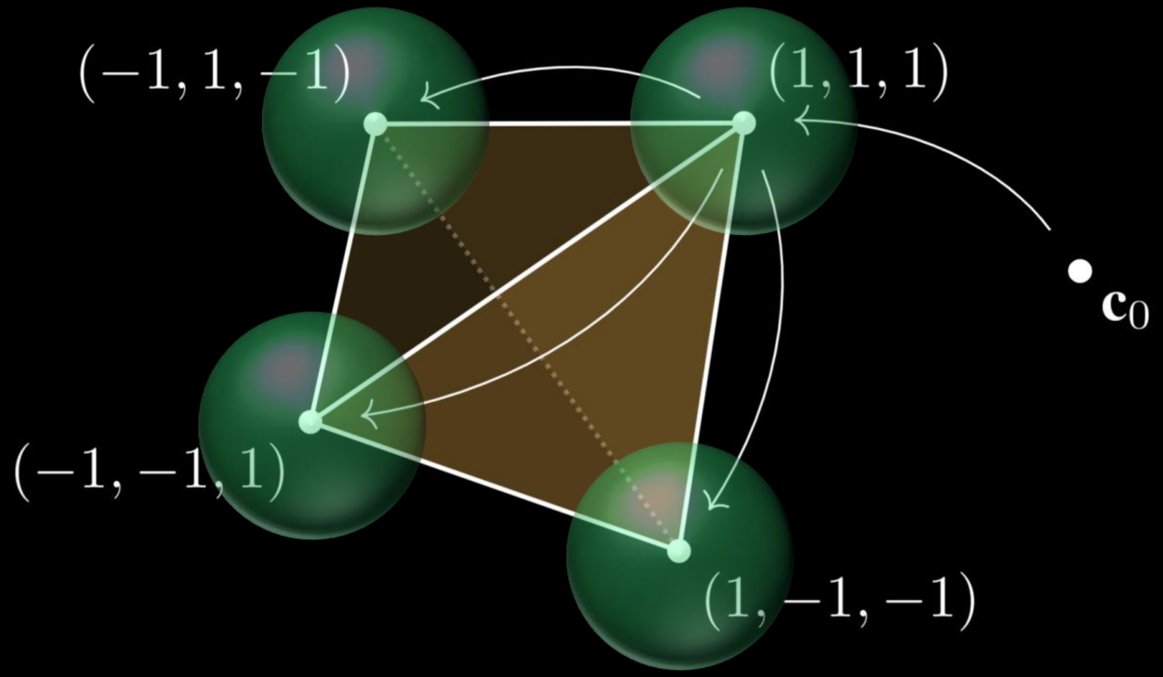
- The original sphere with center $\mathbf{c}_0$
- The four spheres must be spaced in 3D *equally*
- Form a *regular tetrahedron* – *spherical filter bank*:



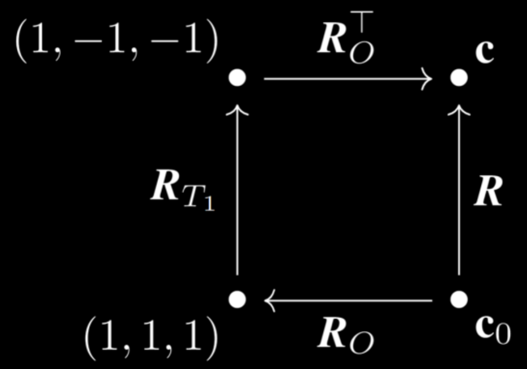
2. The basis functions

- The original sphere with center $\mathbf{c}_0$
- The four spheres must be spaced in 3D *equally*
- Form a *regular tetrahedron* – *spherical filter bank*:



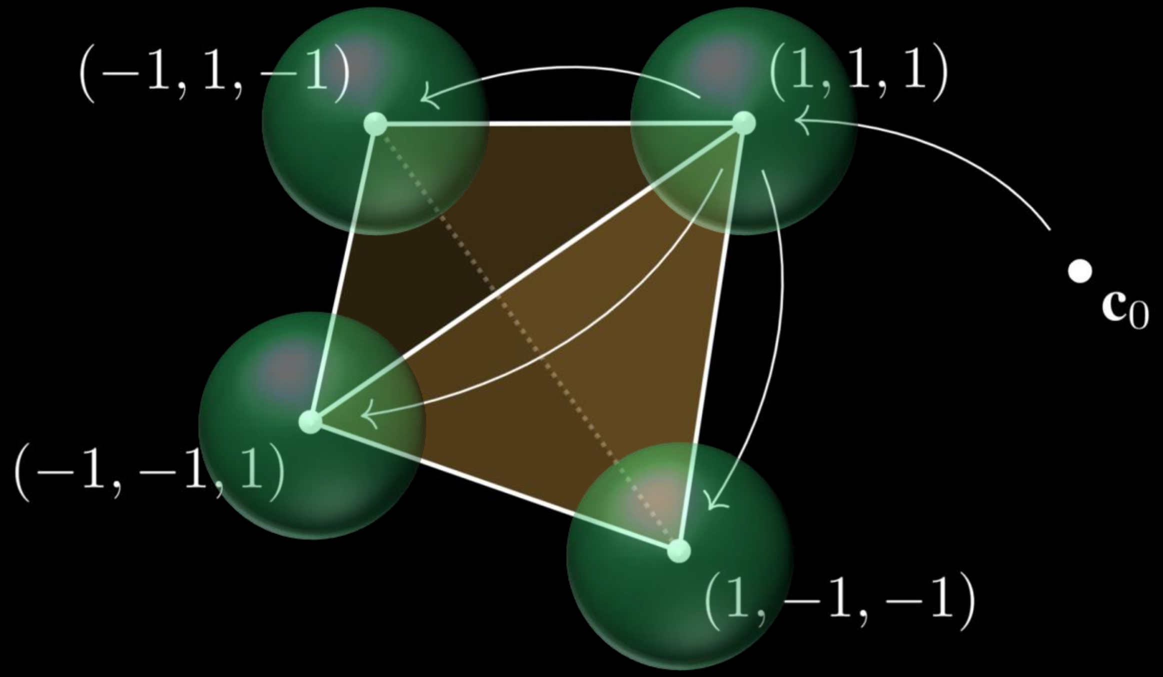


2. The basis functions

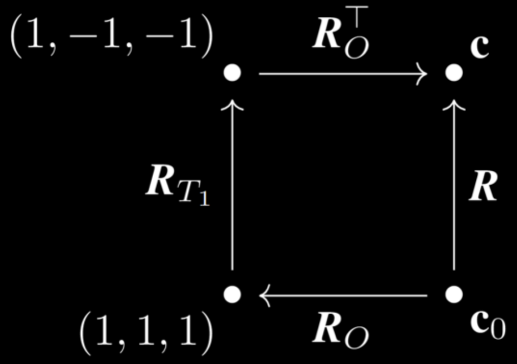


- The original sphere with center $\mathbf{c_0}$
- The four spheres must be spaced in 3D *equally*
- Form a *regular tetrahedron* – *spherical filter bank*:

$$B(\boldsymbol{S}) = \left[\boldsymbol{R}_O^\top \boldsymbol{R}_{Ti} \boldsymbol{R}_O \boldsymbol{S}\right]$$



2. The basis functions



- The original sphere with center $\mathbf{c}_0$
- The four spheres must be spaced in 3D *equally*
- Form a *regular tetrahedron* – *spherical filter bank*:

$$B(\boldsymbol{S}) = \left[\boldsymbol{R}_O^\top \boldsymbol{R}_{Ti} \boldsymbol{R}_O \boldsymbol{S}\right]$$

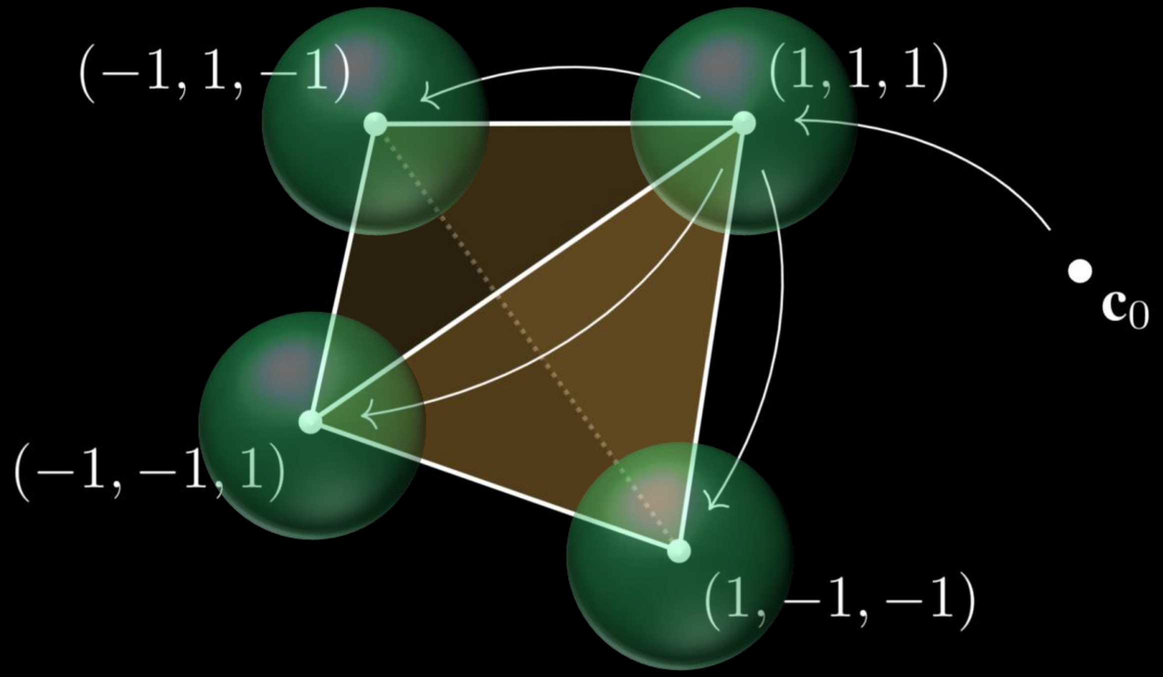
4×5

back to the original CS

original sphere (5×1)

initial rotation (5×5) – from the orig. center into $\|\mathbf{c}_0\|\cdot(1, 1, 1)$

tetrahedron rotation (5×5) – from (1, 1, 1) into one of the other three vertices



3. Interpolation coefficients

- Spherical filter banks are *SO(3)-equivariant* (Theorem 4.2)

$$V_{\boldsymbol{R}}\, B(\boldsymbol{S})\, \boldsymbol{X} = B(\boldsymbol{S})\, \boldsymbol{R}\, \boldsymbol{X}$$

3. Interpolation coefficients

- Spherical filter banks are *SO(3)-equivariant* (Theorem 4.2)

$$V_{\boldsymbol{R}}\, B(\boldsymbol{S})\, \boldsymbol{X} = B(\boldsymbol{S})\, \boldsymbol{R}\, \boldsymbol{X}$$

representation of $\boldsymbol{R}$ in the filter bank output space

$$V_{\boldsymbol{R}} = \mathbf{M}^{\top} \boldsymbol{R}_O\, \boldsymbol{R}\, \boldsymbol{R}_O{}^{\top} \mathbf{M} \quad \in \mathbb{R}^{4\times 4}$$

3. Interpolation coefficients

- Spherical filter banks are *SO(3)-equivariant* (Theorem 4.2)

$$V_{\boldsymbol{R}}\, B(\boldsymbol{S})\, \boldsymbol{X} = B(\boldsymbol{S})\, \boldsymbol{R}\, \boldsymbol{X}$$

representation of $\boldsymbol{R}$ in the filter bank output space

$$V_{\boldsymbol{R}} = \mathbf{M}^\top \boldsymbol{R}_O\, \boldsymbol{R}\, \boldsymbol{R}_O{}^\top \mathbf{M} \quad \in \mathbb{R}^{4\times 4}$$

the change-of-basis matrix

$$\mathbf{M} = \begin{bmatrix}\mathbf{m}_1 & \mathbf{m}_2 & \mathbf{m}_3 & \mathbf{m}_4\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1 & 1 & -1 & -1\\ 1 & -1 & 1 & -1\\ 1 & -1 & -1 & 1\\ 1 & 1 & 1 & 1\end{bmatrix}$$

3. Interpolation coefficients

- Spherical filter banks are *SO(3)-equivariant* (Theorem 4.2)

$$V_{\boldsymbol{R}}\, B(\boldsymbol{S})\, \boldsymbol{X} = B(\boldsymbol{S})\, \boldsymbol{R}\, \boldsymbol{X}$$

representation of $\boldsymbol{R}$ in the filter bank output space
the 1st column contains interpolation coefficients $v(\boldsymbol{R})$

$$V_{\boldsymbol{R}} = \mathbf{M}^\top \boldsymbol{R}_O \boldsymbol{R} \boldsymbol{R}_O{}^\top \mathbf{M} \quad \in \mathbb{R}^{4\times 4}$$

the change-of-basis matrix

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}_1 & \mathbf{m}_2 & \mathbf{m}_3 & \mathbf{m}_4 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

The steerability constraint

$$f(\boldsymbol{X}) = f^{\boldsymbol{R}}(\boldsymbol{R}\boldsymbol{X}) = \sum_{j=1}^{M} v_j(\boldsymbol{R})\, f^{\boldsymbol{R}_j}(\boldsymbol{R}\boldsymbol{X}) = v(\boldsymbol{R})^\top B(\boldsymbol{S})\, \boldsymbol{R}\boldsymbol{X}$$

The steerability constraint

$$f(\boldsymbol{X}) = f^{\boldsymbol{R}}(\boldsymbol{R}\boldsymbol{X}) = \sum_{j=1}^{M} v_j(\boldsymbol{R})\, f^{\boldsymbol{R}_j}(\boldsymbol{R}\boldsymbol{X}) = v(\boldsymbol{R})^\top B(\boldsymbol{S})\, \boldsymbol{R}\boldsymbol{X}$$

scalar

$$v(\boldsymbol{R}) = \mathbf{M}^\top (\boldsymbol{R}_O\, \boldsymbol{R}\, \boldsymbol{R}_O{}^\top \mathbf{m}_1)$$

4×1

The steerability constraint

$$f(\boldsymbol{X}) = f^{\boldsymbol{R}}(\boldsymbol{R}\boldsymbol{X}) = \sum_{j=1}^{M} v_j(\boldsymbol{R})\, f^{\boldsymbol{R}_j}(\boldsymbol{R}\boldsymbol{X}) = v(\boldsymbol{R})^\top B(\boldsymbol{S})\,\boldsymbol{R}\boldsymbol{X}$$

1×4

scalar

4×5

5×1

as per Melnyk et al. (2021)

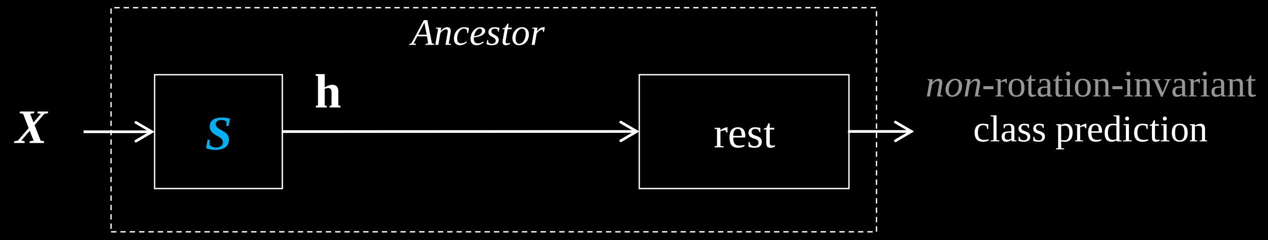
$$v(\boldsymbol{R}) = \mathbf{M}^\top \left(\boldsymbol{R}_O\, \boldsymbol{R}\, \boldsymbol{R}_O{}^\top \mathbf{m}_1\right)$$

4×1

scalar output

Model outline

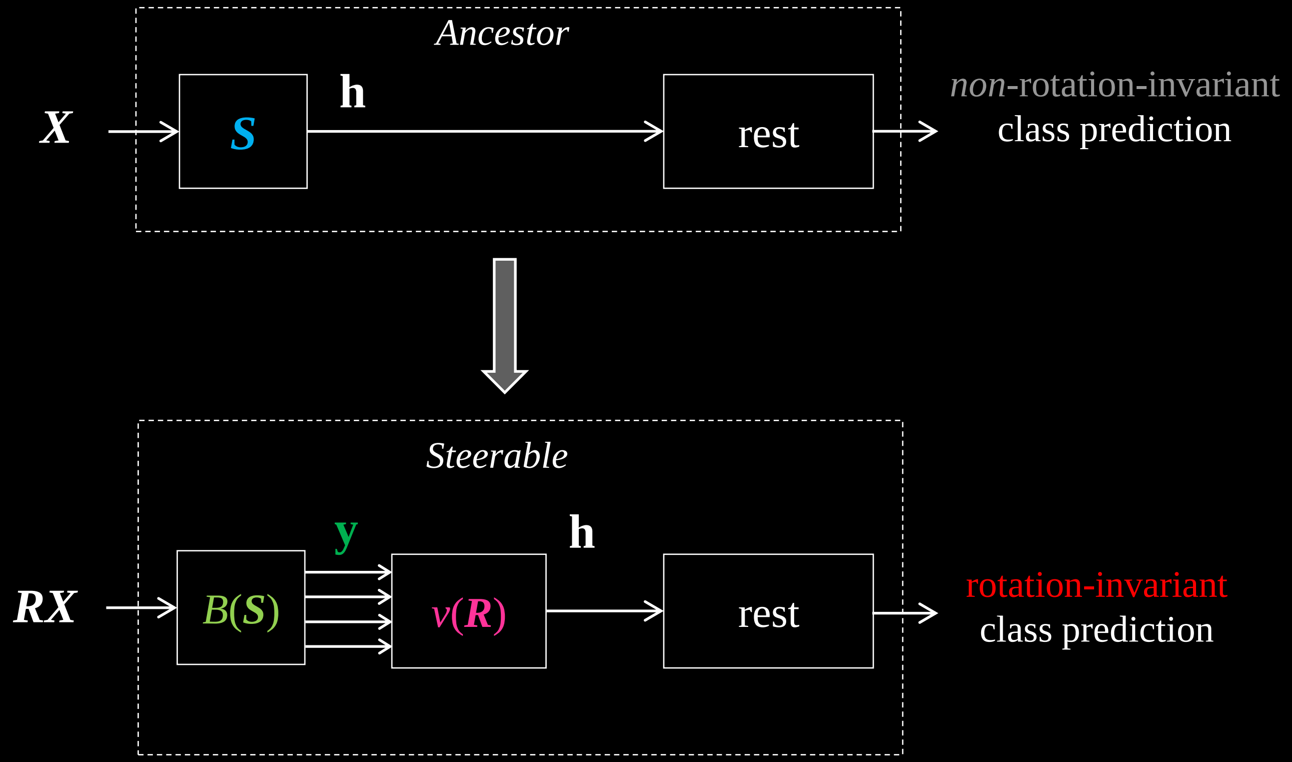
1) Train Ancestor



Model outline

1) Train Ancestor

2) Steer the spherical neurons

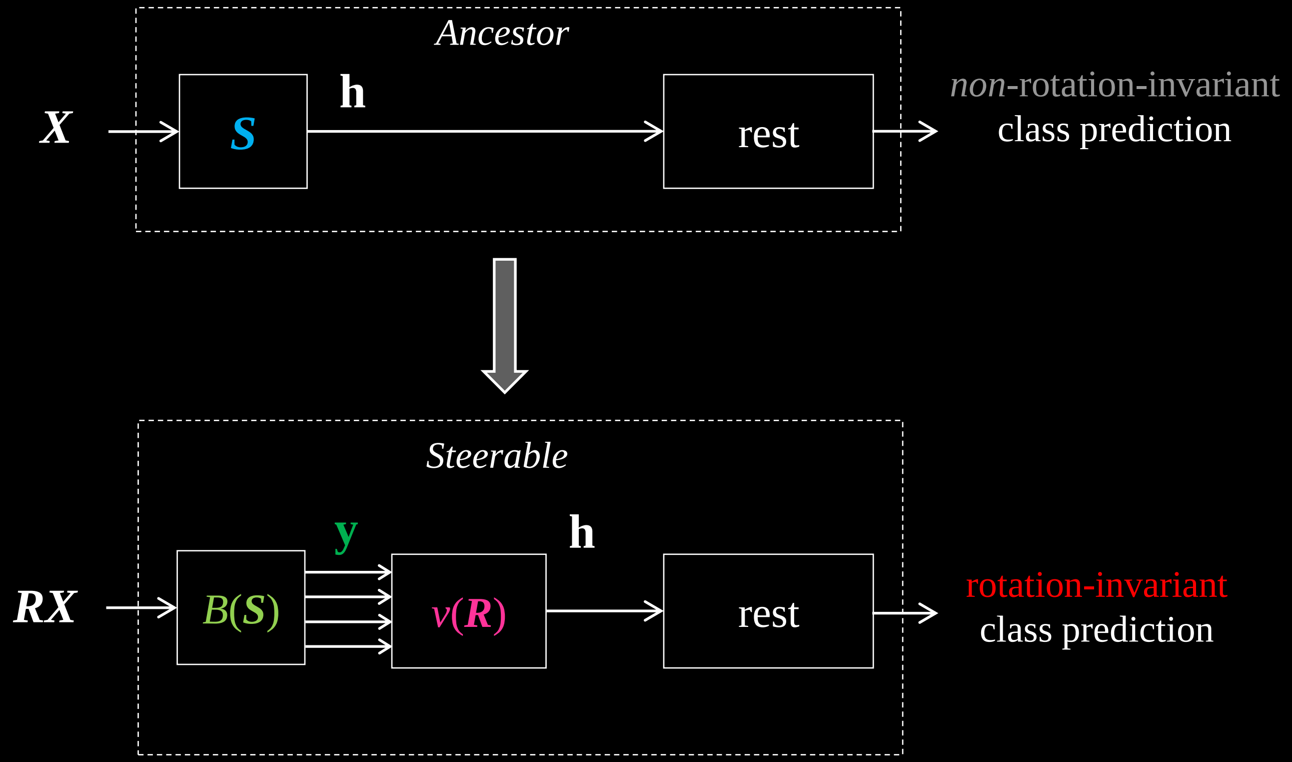


Model outline

1) Train Ancestor

2) Steer the spherical neurons

- construct filter banks
- with SO(3)-equivariant outputs **y**



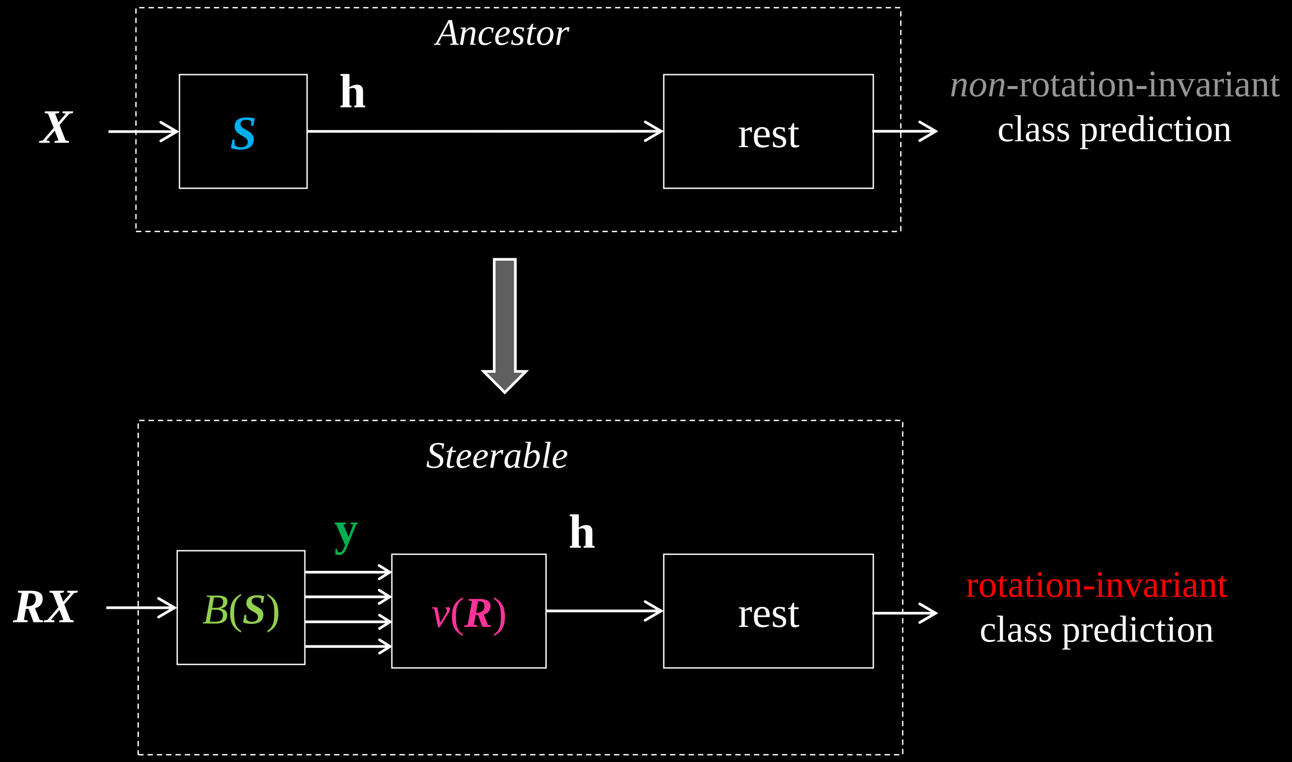
Model outline

1) Train Ancestor

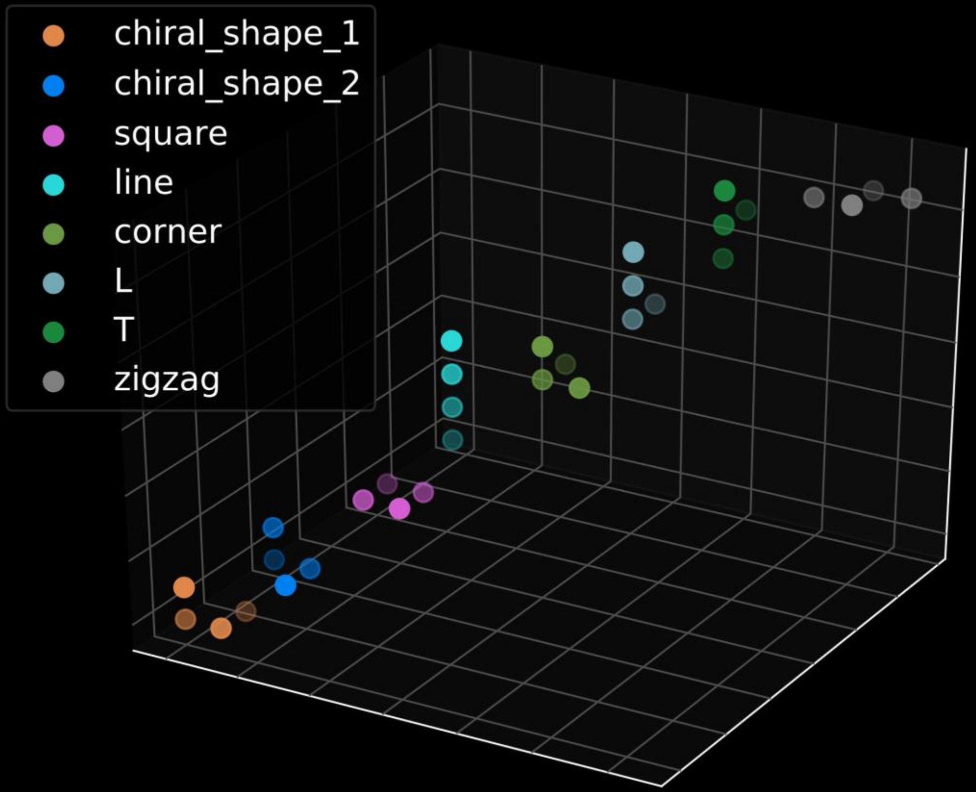
2) Steer the spherical neurons

- construct filter banks
- with SO(3)-equivariant outputs $\mathbf{y}$

- add interpolation coefficients as *free* parameters
- computed correctly → SO(3)-invariant predictions

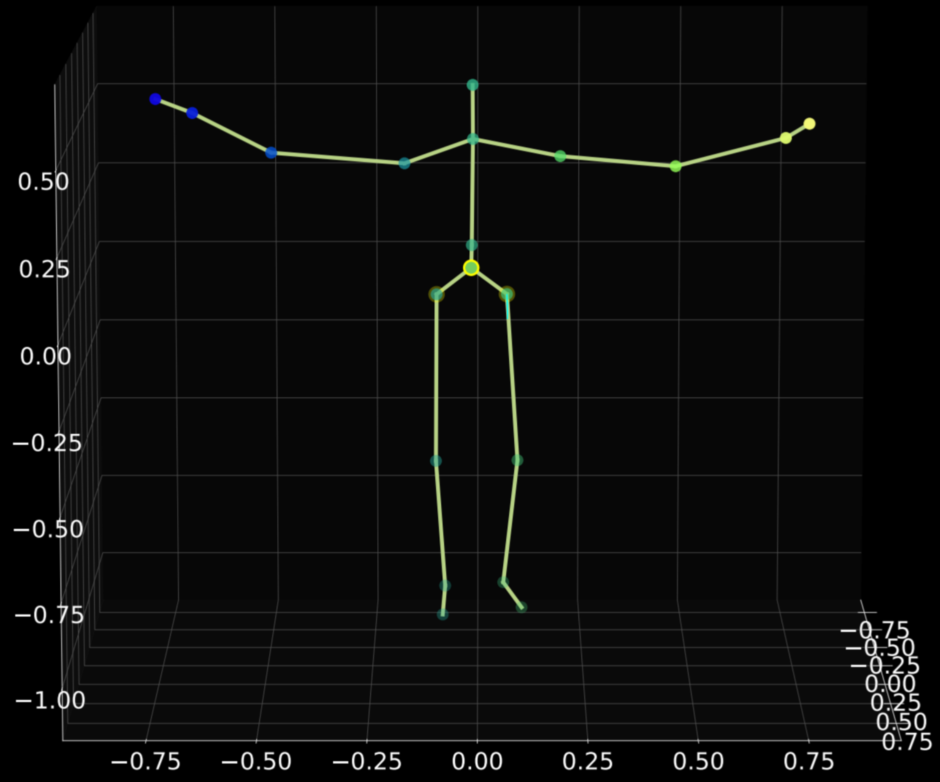


Experimental validation



3D Tetris data (Thomas et al. 2018)

Image: Melnyk et. al (2021)



3D skeleton data (Xia et al. 2012)

Image: Melnyk et. al (2022)

Experimental validation

Table 1. The steerable model classification accuracy for the distorted (the noise units are specified in the square brackets) rotated shapes and the ancestor accuracy for the distorted shapes in their canonical orientation (mean and std over 1000 runs, %).

3D Tetris			3D skeleton data (*test* set)		
Noise (a), [1]	*Steerable*	*Ancestor*	Noise (a), [m]	*Steerable*	*Ancestor*
0.00	100.0 ± 0.0	100.0 ± 0.0	0.000	92.9 ± 0.0	92.9 ± 0.0
0.05	100.0 ± 0.0	100.0 ± 0.0	0.005	92.4 ± 0.2	92.4 ± 0.2
0.10	100.0 ± 0.0	100.0 ± 0.0	0.010	91.1 ± 0.3	91.1 ± 0.3
0.20	100.0 ± 0.4	100.0 ± 0.0	0.020	87.1 ± 0.5	87.1 ± 0.5
0.30	99.7 ± 1.9	99.8 ± 1.6	0.030	82.3 ± 0.6	82.2 ± 0.6
0.50	94.9 ± 7.7	95.0 ± 7.9	0.050	72.0 ± 0.7	71.9 ± 0.7

Experimental validation

Table 1. The steerable model classification accuracy for the distorted (the noise units are specified in the square brackets) rotated shapes and the ancestor accuracy for the distorted shapes in their canonical orientation (mean and std over 1000 runs, %).

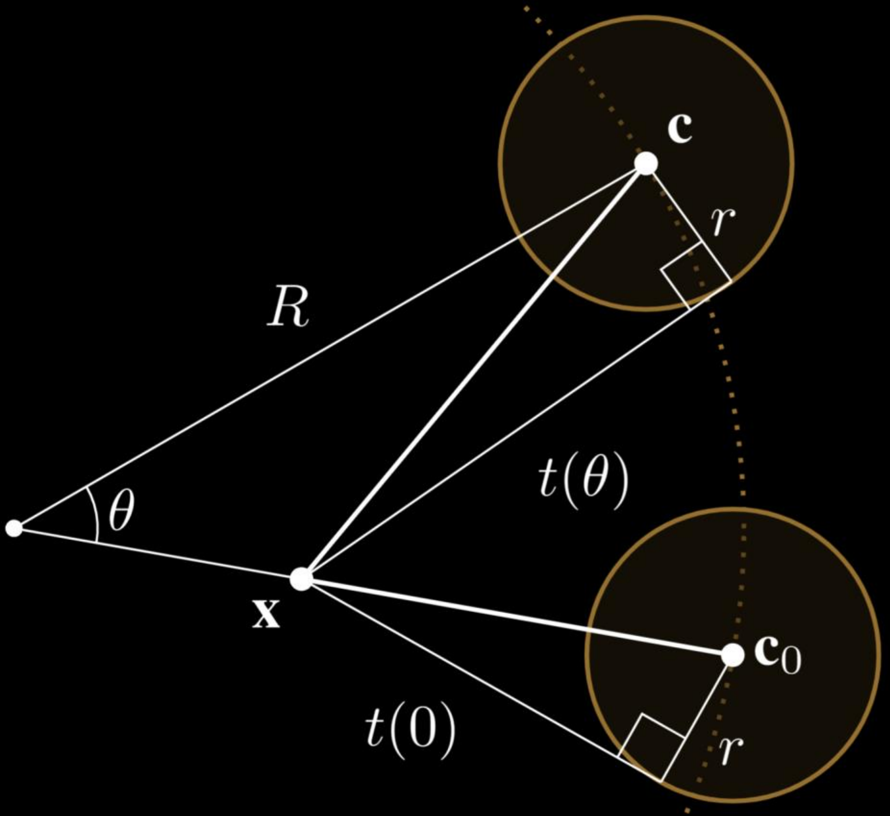
3D Tetris			3D skeleton data (*test* set)		
Noise (a), [1]	*Steerable*	*Ancestor*	Noise (a), [m]	*Steerable*	*Ancestor*
0.00	100.0 ± 0.0	100.0 ± 0.0	0.000	92.9 ± 0.0	92.9 ± 0.0
0.05	100.0 ± 0.0	100.0 ± 0.0	0.005	92.4 ± 0.2	92.4 ± 0.2
0.10	100.0 ± 0.0	100.0 ± 0.0	0.010	91.1 ± 0.3	91.1 ± 0.3
0.20	100.0 ± 0.4	100.0 ± 0.0	0.020	87.1 ± 0.5	87.1 ± 0.5
0.30	99.7 ± 1.9	99.8 ± 1.6	0.030	82.3 ± 0.6	82.2 ± 0.6
0.50	94.9 ± 7.7	95.0 ± 7.9	0.050	72.0 ± 0.7	71.9 ± 0.7

Steerable 3D Spherical Neurons

pavlo.melnyk@liu.se

michael.felsberg@liu.se

marten.wadenback@liu.se



paper

code

LINKÖPING UNIVERSITY