UNIVERSITY OF AMSTERDAM

Introduction to group equivariant deep learning

Erik J Bekkers - e.j.bekkers@uva.nl - 7 April 2021 - TLDR Thinking Hour

Content

Part I: Introduction to group convolutions

* Motivation
* Introduction to group theory
* Regular group convolutional neural networks
* Applications

Part II: General theory for group equivariant deep learning

* Group convolutions are all you need!
* Deeper into group theory: representation theory, homogeneous spaces
* Characterization of types of group equivariant layers

Part III: Steerable group convolutions

* Deep dive into group theory: irreducible representations, steerable operators and vector spaces
* Examples of steerable group convolutions: Spherical data and Volumetric data/3D point clouds

Lecture notes, slides and exercises available at

https://uvagedl.github.io

An Introduction to Equivariant Convolutional Neural Networks for Continuous Groups
Lecture Notes

dr.ir. Erik J. Bekkers

August 2021

Contents

1

Exercise 2.4. Show transitivity (Definition 2.8) of the action of G given in Eq. (29).

Example 2.7 (Quotient space $\mathbb{R}^d = SE(d)/SO(d)$). Let $H = (\{\mathbf{0}\} \times SO(d)$ the subgroup of rotations in $SE(d)$, with $\mathbf{0}$ the identity element of the translationg roup $(\mathbb{R}^d, +)$. The the cosets gH are given by

$$\begin{aligned} gH &= \{g \cdot (\mathbf{0}, \tilde{\mathbf{R}}) \mid \tilde{\mathbf{R}} \in SO(d)\} \\ &= \{(\mathbf{R}\mathbf{e} + \mathbf{x}, \mathbf{R}\tilde{\mathbf{R}}) \mid h \in SO(d)\} \\ &= \{(\mathbf{x}, \mathbf{R}\tilde{\mathbf{R}}) \mid \tilde{\mathbf{R}} \in SO(d)\} \\ &= \{(\mathbf{x}, \tilde{\mathbf{R}}) \mid \tilde{\mathbf{R}} \in SO(d)\}\,, \end{aligned}$$

with $g = (\mathbf{x}, \mathbf{R})$. So, the cosets are given by all possible rotations for a fixed translation vector $\mathbf{x}$, the vector $\mathbf{x}$ thus indexes these sets. We can therefore make the identification

$$\mathbb{R}^d \equiv SE(d)/SO(d)\,.$$

We already saw in Exercise 2.1 that $\mathbb{R}^d$ is a homogeneous space of $SE(d)$, this is a consequence of Lemma 2.1.

Lemma 2.1 shows that a quotient space G/H of a group G with H is a homogeneous space. We can also approach this in the other direction and state that any homogeneous space of G is equivalent to a quotient space G/H for some H. This is stated in the following Lemma, for which we first need to introduce the notion of a stabilizer.

Definition 2.13 (Stabilizer). Let G act on X via the action $\odot$. For every $x \in X$, the stabilizer subgroup of G with respect to the point x is denoted with $\mathrm{Stab}_G(x)$ is the set of all elements in G that fix x:

$$\mathrm{Stab}_G(x) = \{g \in G \mid g \odot x = x\}. \tag{30}$$

Lemma 2.2. Let X be a homogeneous space of G. Then X can be identified with G/H with $H = \mathrm{Stab}_G(x_0)$ for any $x_0 \in X$.

Affine groups Finally when it comes to types of groups and homogeneous spaces we note that often we are interested in groups that act on $\mathbb{R}^d$, as most often one deal with data on $\mathbb{R}^d$. It is therefore useful to introduce the following class of groups.

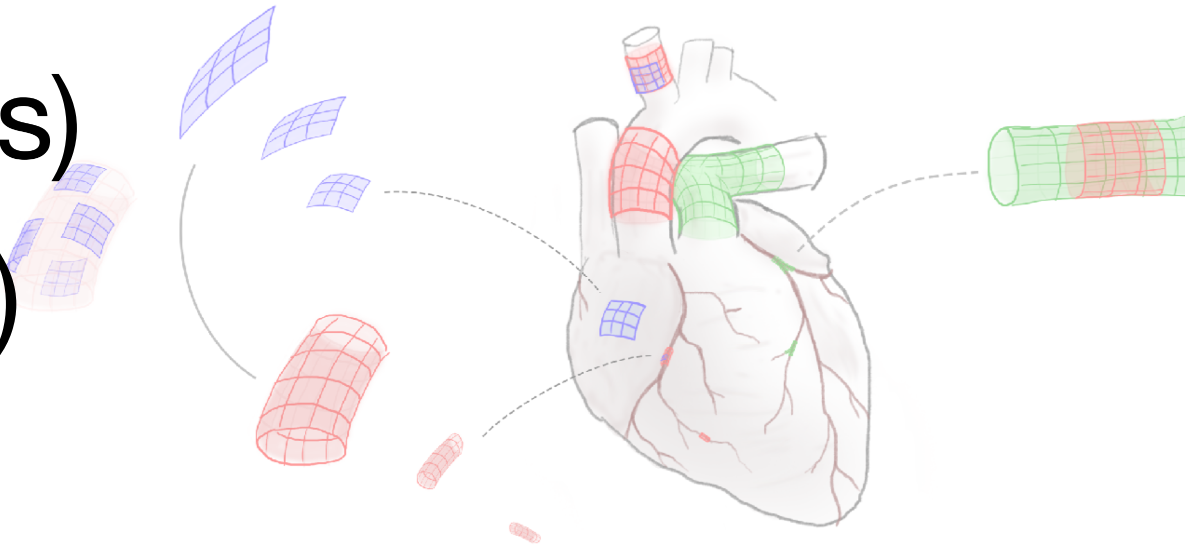
Definition 2.14 (Affine groups). Affine groups $G = \mathbb{R}^d \rtimes H$ are a class of groups that are the semidirect product of a group $H \subseteq GL(\mathbb{R}^d)$ acting on $\mathbb{R}^d$, with $GL(\mathbb{R}^d$ the group of general linear transformations acting on $\mathbb{R}^d$.

The transformations in $H \subseteq GL(\mathbb{R}^d)$ are commonly represented as invertible matrices $\mathbf{A}$ which act on $\mathbb{R}^d$ via matrix-vector multiplication, by which the group

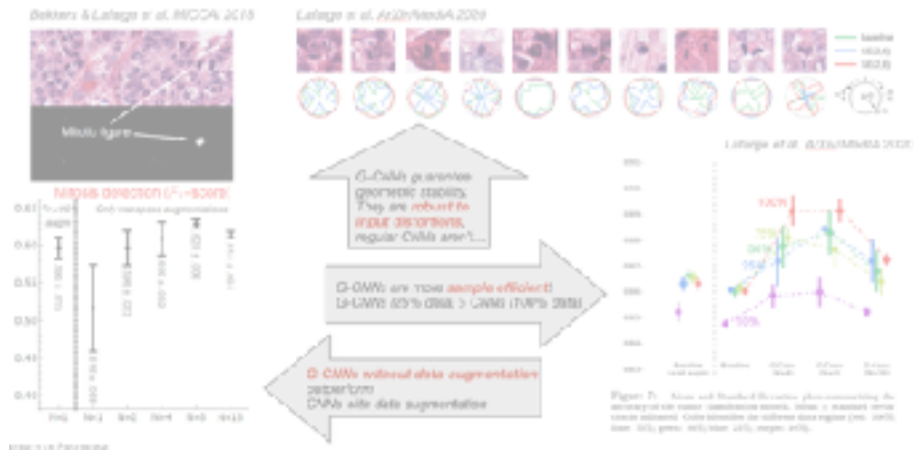
Part I

1. Why do we want equivariant learning models?
   - Geometric guarantees + weight sharing/sample efficiency

2. A group theoretical view on **recognition by components** (capsule nets)
   - Group theoretical prerequisites (**group product** and **representations**)
   - **Group convolutions** perform pattern recognition by components



3. Experimental examples



Content of this talk

1. Why do we want equivariant learning models?
   - Geometric guarantees + weight sharing/sample efficiency

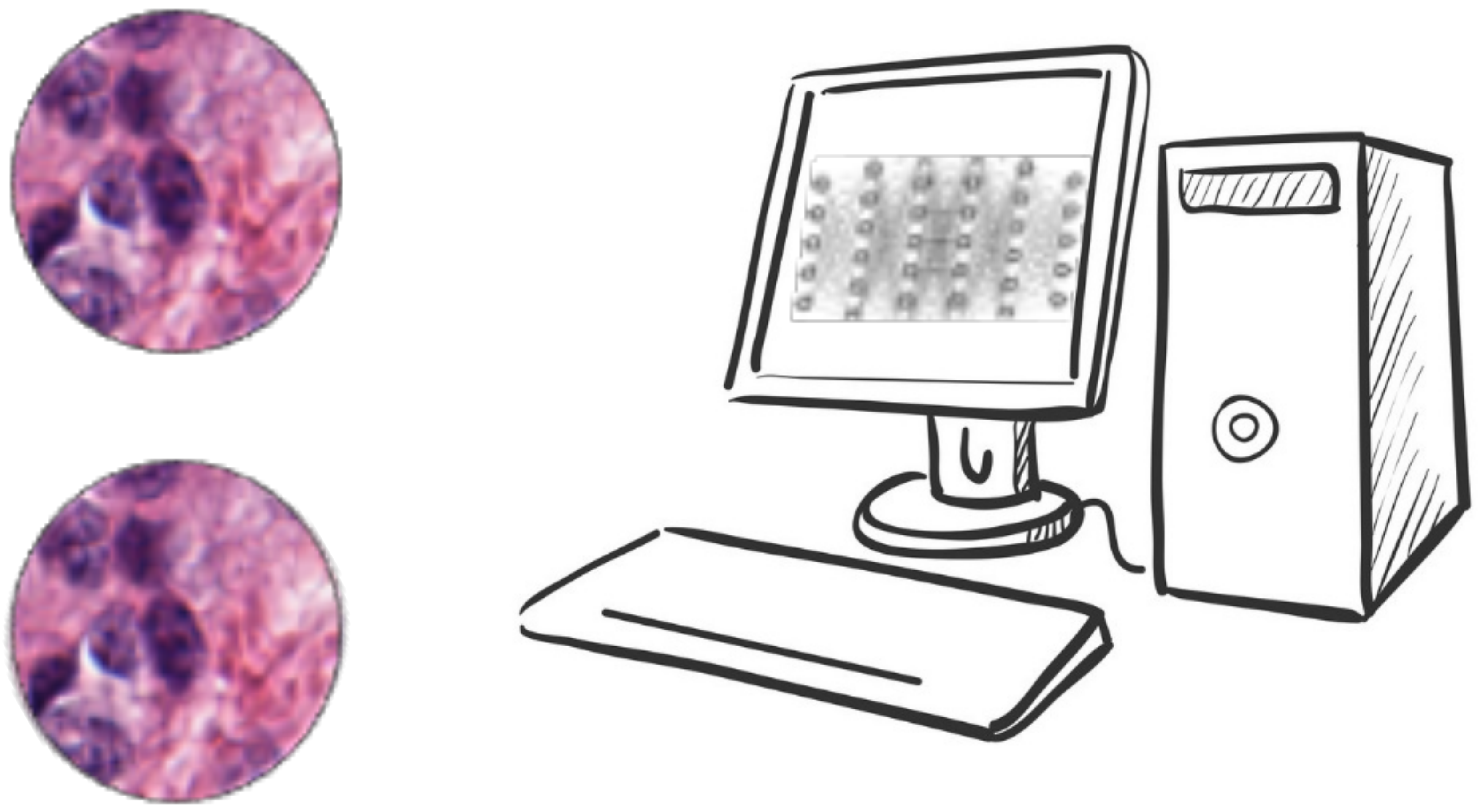
2. A group theoretical view on **recognition by components** (capsule nets)
   - Group theoretical prerequisites (**group product** and **representations**)
   - **Group convolutions** perform pattern recognition by components

3. Experimental examples

4. Theorem: Linear maps between feature maps are equivariant iff they are group convolutions

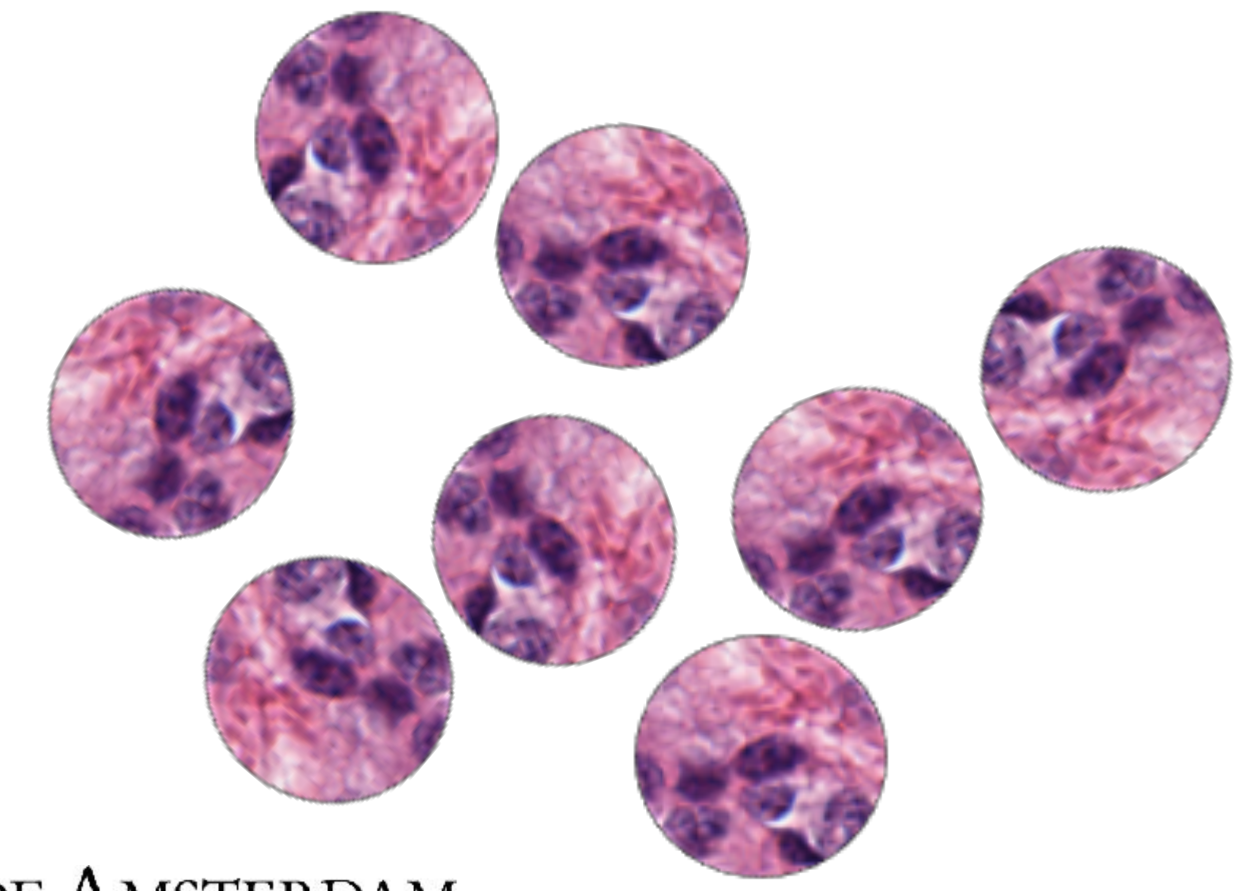
Motivation: Geometric Guarantees (invariance)

Example: Detection of pathological cells



Pathological

Common approach: data-augmentation



Issues:
- Still no guarantee of i
- Valuable net
  learning r
- Redundan

Motivation: Geometric Guarantees (equivariance)

CNNs are translation equivariant

Via convolutions →

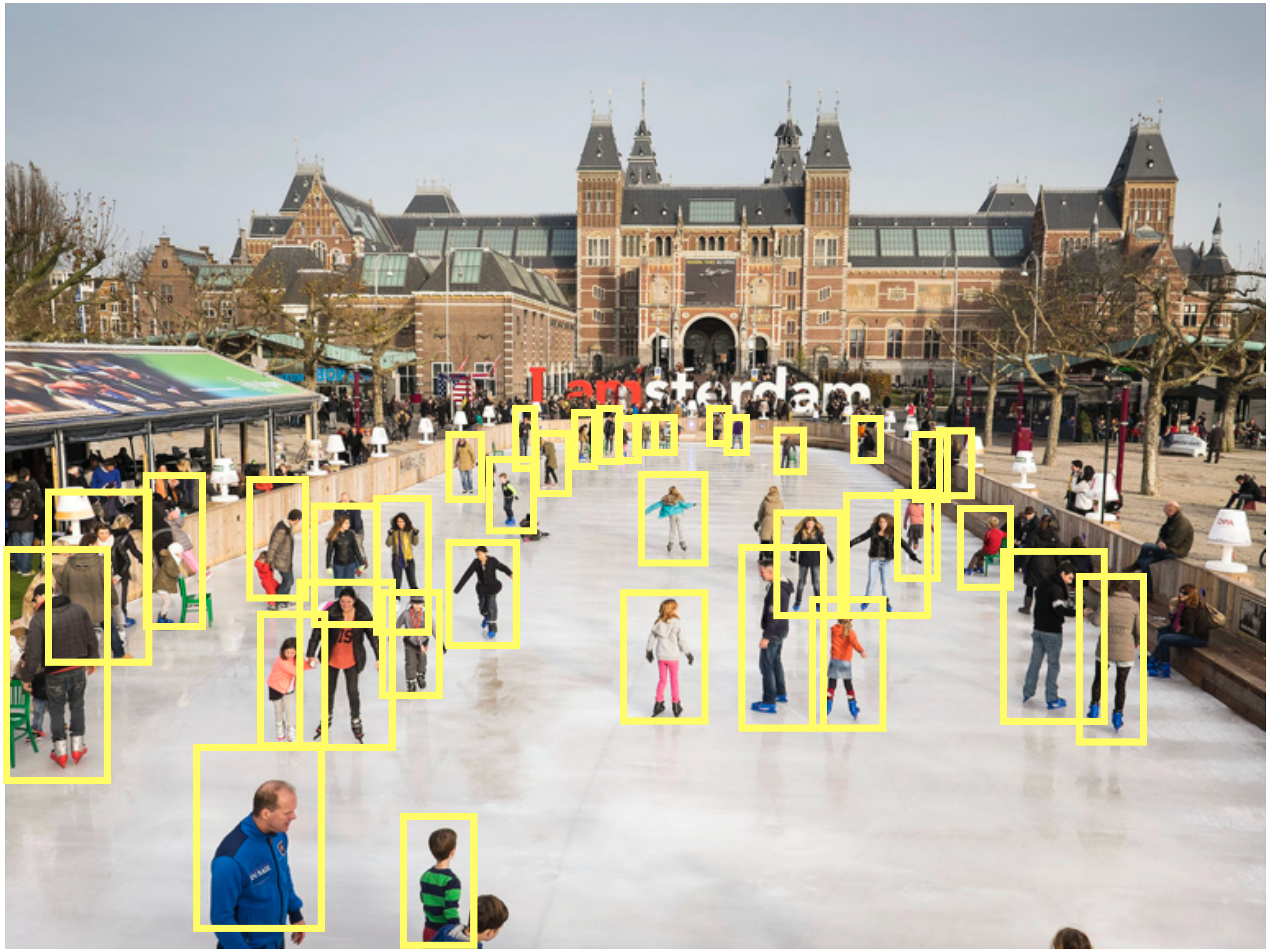
Motivation: Geometric Guarantees (equivariance)

CNNs are not rotation equivariant

Via convolutions →

Stabilized view

Motivation: Geometric Guarantees (equivariance)



Importance of equivariance:

- No information is lost when the input is transformed
- Guaranteed stability to (local + global) transformations

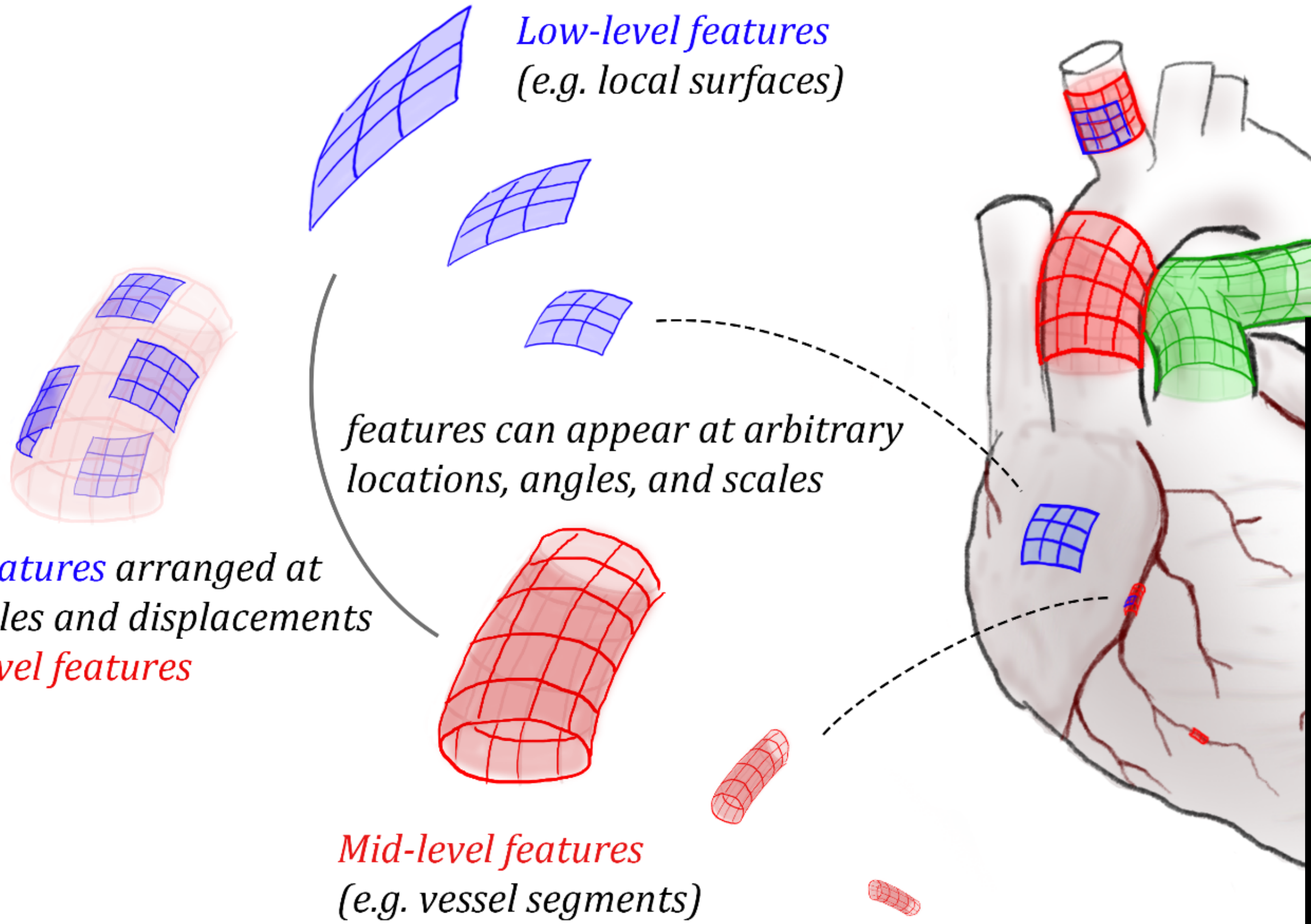
Group convolutions:

- Equivariance beyond translations
- Geometric guarantees
- Increased weight sharing

G-CNNs are not only relevant for invariant problems but for any type of structured data!

Motivation: Recognition by components

In a group theoretical setting



Rationale behind capsule networks

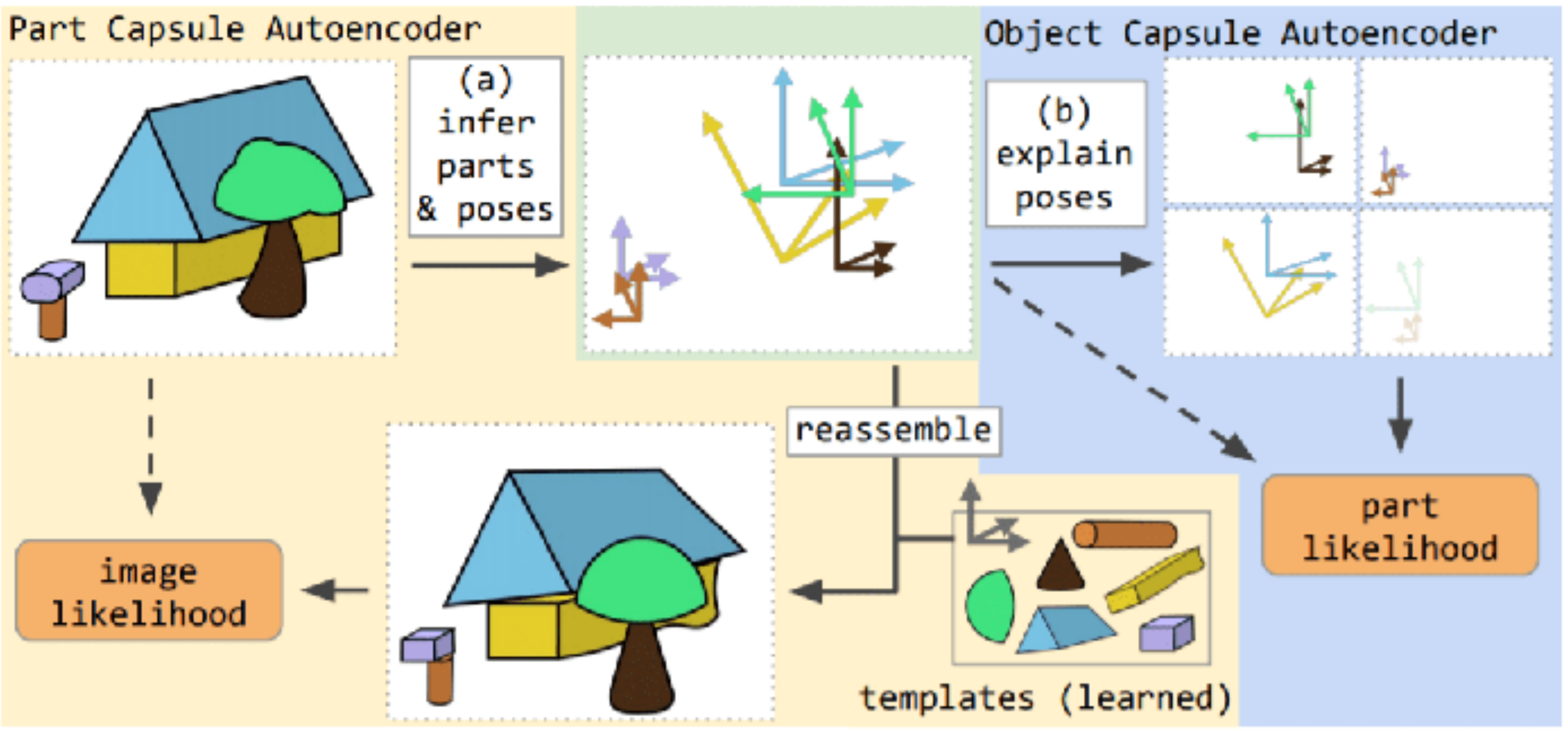


Fig from Kosiorek et al. 2019

Content of this talk

1. Why do we want equivariant learning models?
   - Geometric guarantees + weight sharing/sample efficiency

2. A group theoretical view on **recognition by components** (capsule nets)
   - Group theoretical prerequisites (**group product** and **representations**)
   - **Group convolutions** perform pattern recognition by components

3. Experimental examples

4. Theorem: Linear maps between feature maps are equivariant iff they are group convolutions

What is a group?

A group $(G, \cdot)$ is a set of elements G equipped with a group product $\cdot$, a binary operator, that satisfies the following four axioms:

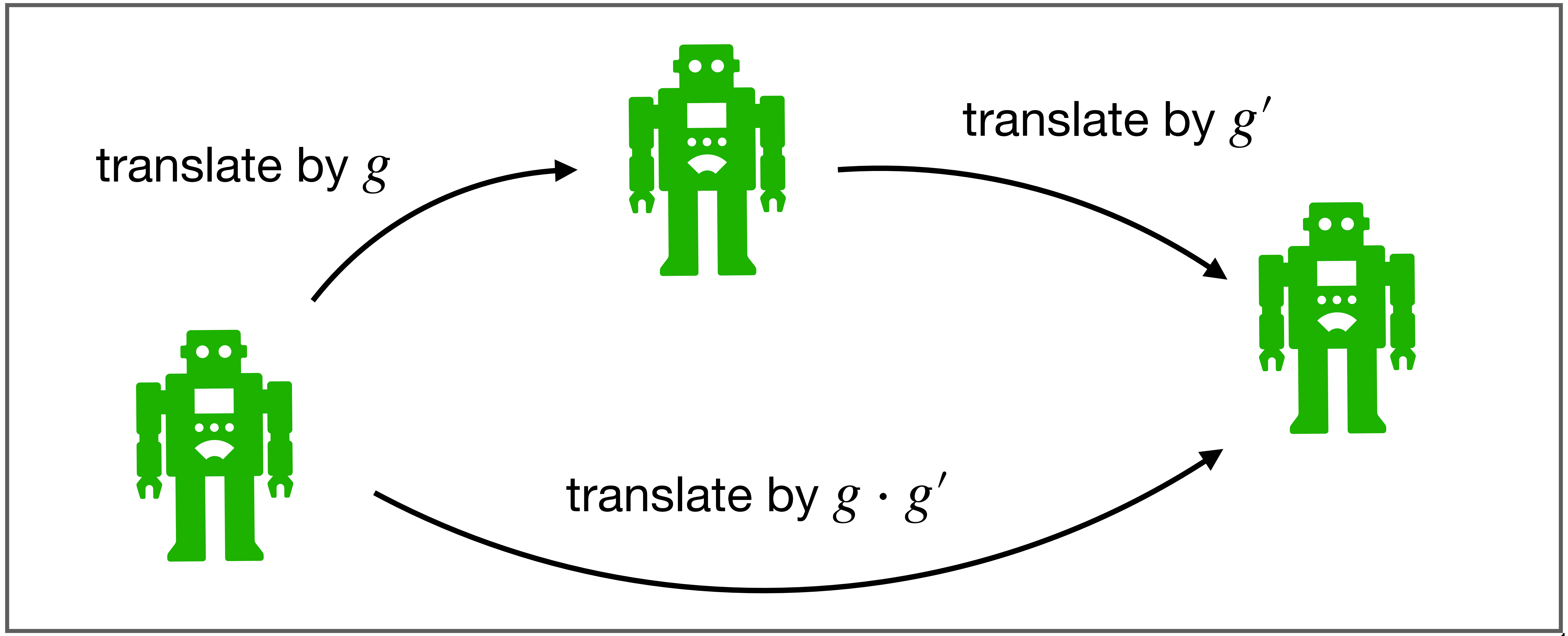
- **Closure**: Given two elements g and h of G, the product $g \cdot h$ is also in G.
- **Associativity**: For $g, h, i \in G$ the product $\cdot$ is associative, i.e., $g \cdot (h \cdot i) = (g \cdot h) \cdot i$.
- **Identity element**: There exists an identity element $e \in G$ such that $e \cdot g = g \cdot e = g$ for any $g \in G$.
- **Inverse element**: For each $g \in G$ there exists an inverse element $g^{-1} \in G$ s.t. $g^{-1} \cdot g = g \cdot g^{-1} = e$.

The translation group $(\mathbb{R}^2, +)$

The translation group consists of all possible translations in $\mathbb{R}^2$ and is equipped with the group product and group inverse:

$$g \cdot g' = (\mathbf{x} + \mathbf{x}')$$
$$g^{-1} = (-\mathbf{x})$$

with $g = (\mathbf{x}), g' = (\mathbf{x}')$ and $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^2$.



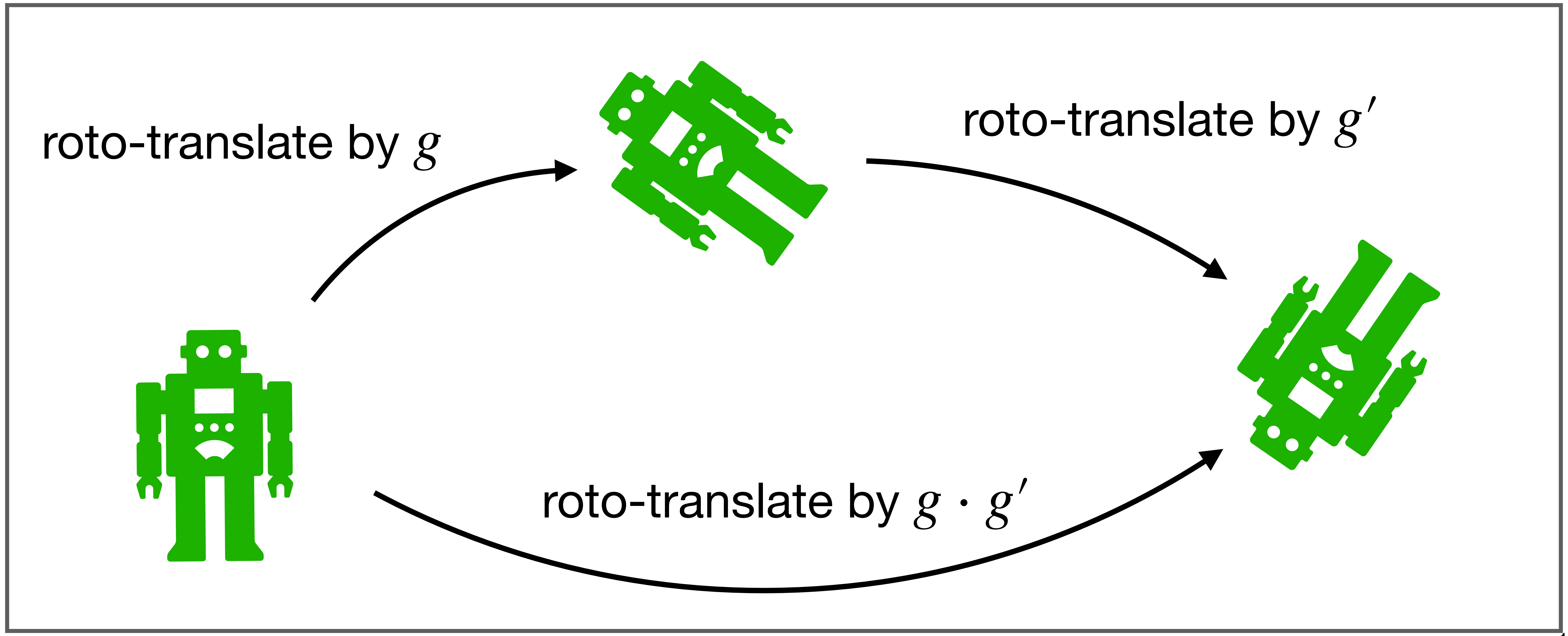
The roto-translation group $SE(2)$

*2D **S**pecial **E**uclidean motion group*

The group $SE(2) = \mathbb{R}^2 \rtimes SO(2)$ consists of the coupled space $\mathbb{R}^2 \times S^1$ of translations vectors in $\mathbb{R}^2$, and rotations in $SO(2)$ (or equivalently orientations in S^1), and is equipped with the group product and group inverse:

$$g \cdot g' = (\mathbf{x}, \mathbf{R}_\theta) \cdot (\mathbf{x}', \mathbf{R}_{\theta'}) = (\mathbf{R}_\theta \mathbf{x}' + \mathbf{x}, \mathbf{R}_{\theta+\theta'})$$
$$g^{-1} = (-\mathbf{R}_\theta^{-1}\mathbf{x}, \mathbf{R}_\theta^{-1})$$

with $g = (\mathbf{x}, \mathbf{R}_\theta)$, $g' = (\mathbf{x}', \mathbf{R}_{\theta'})$.



The scale-translation group $\mathbb{R}^2 \rtimes \mathbb{R}^+$

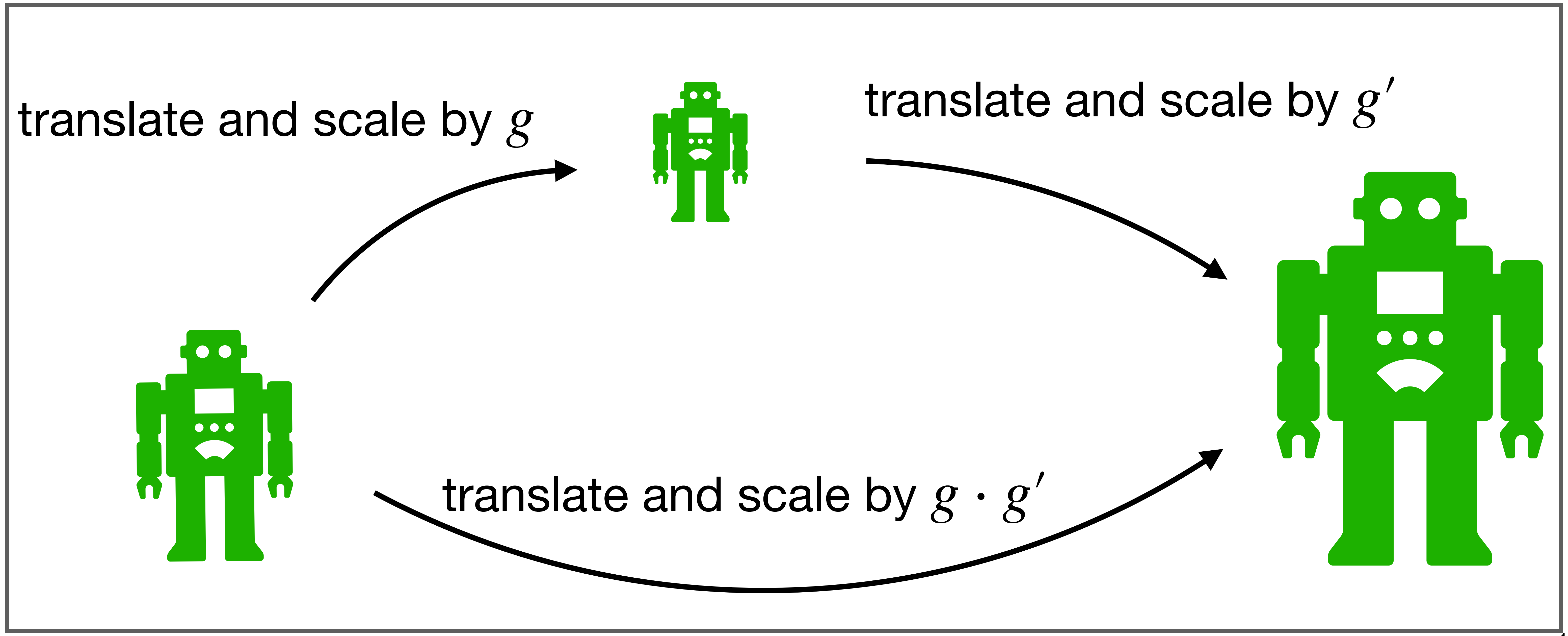
The scale-translation group of space $\mathbb{R}^2 \times \mathbb{R}^+$ of translations vectors in $\mathbb{R}^2$ and scale/dilation factors in $\mathbb{R}^+$, and is equipped with the group product and group inverse:

$$g \cdot g' = (\mathbf{x}, s) \cdot (\mathbf{x}', s') = (s\mathbf{x}' + \mathbf{x}, ss')$$
$$g^{-1} = \left(-\frac{1}{s}\mathbf{x}, \frac{1}{s}\right)$$

with $g = (\mathbf{x}, s), g' = (\mathbf{x}', s')$.

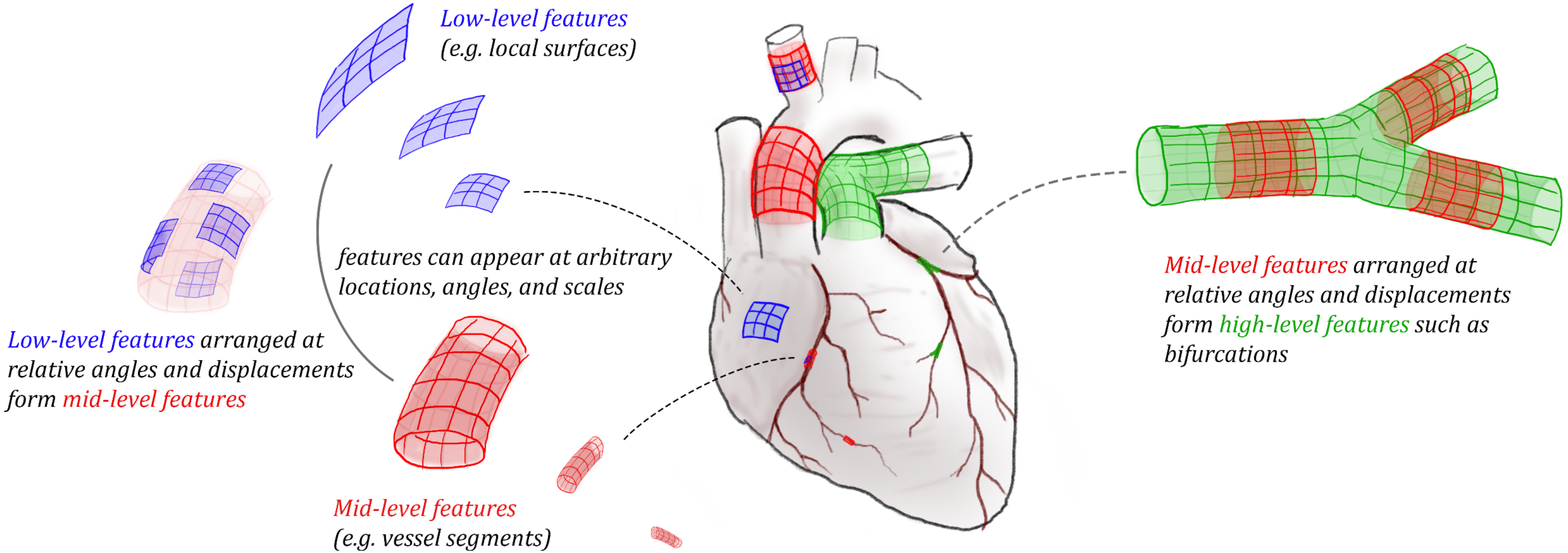
with $g \cdot g^{-1} = e = (\mathbf{0}, 1)$

matrix repr: $\mathbf{G} = \begin{pmatrix} \mathbf{I}_s & \mathbf{x} \\ \mathbf{0}^T & 1 \end{pmatrix}$



Motivation: Recognition by components

In a group theoretical setting



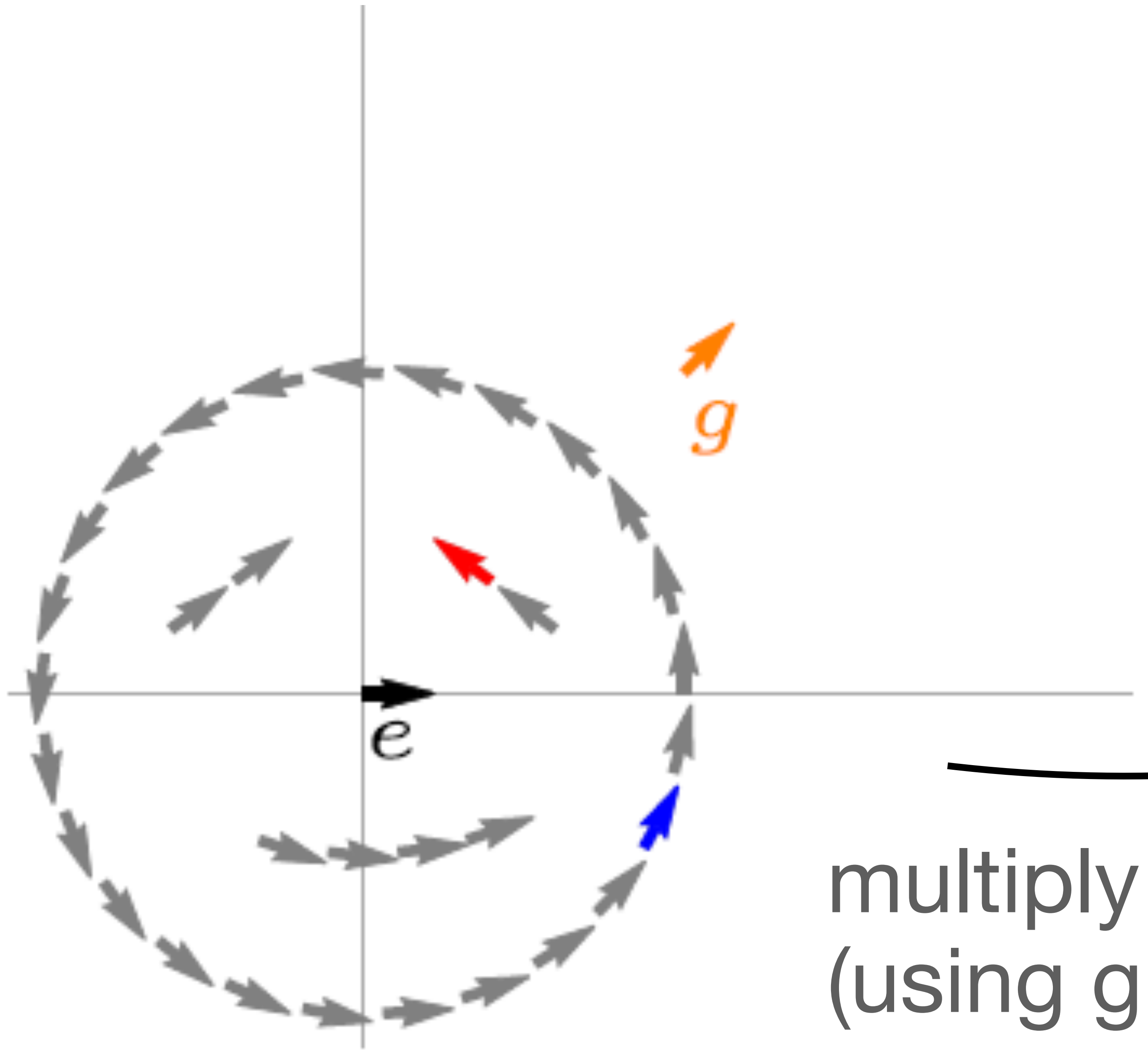
The roto-translation group $SE(2)$

*2D **S**pecial **E**uclidean motion group*

The group $SE(2) = \mathbb{R}^2 \rtimes SO(2)$ consists of the coupled space $\mathbb{R}^2 \times S^1$ of translations vectors in $\mathbb{R}^2$, and rotations in $SO(2)$ (or equivalently orientations in S^1), and is equipped with the group product and group inverse:

$$g \cdot g' = (\mathbf{x}, \mathbf{R}_\theta) \cdot (\mathbf{x}', \mathbf{R}_{\theta'}) = (\mathbf{R}_\theta \mathbf{x}' + \mathbf{x}, \mathbf{R}_{\theta+\theta'})$$
$$g^{-1} = (-\mathbf{R}_\theta^{-1}\mathbf{x}, \mathbf{R}_\theta^{-1})$$

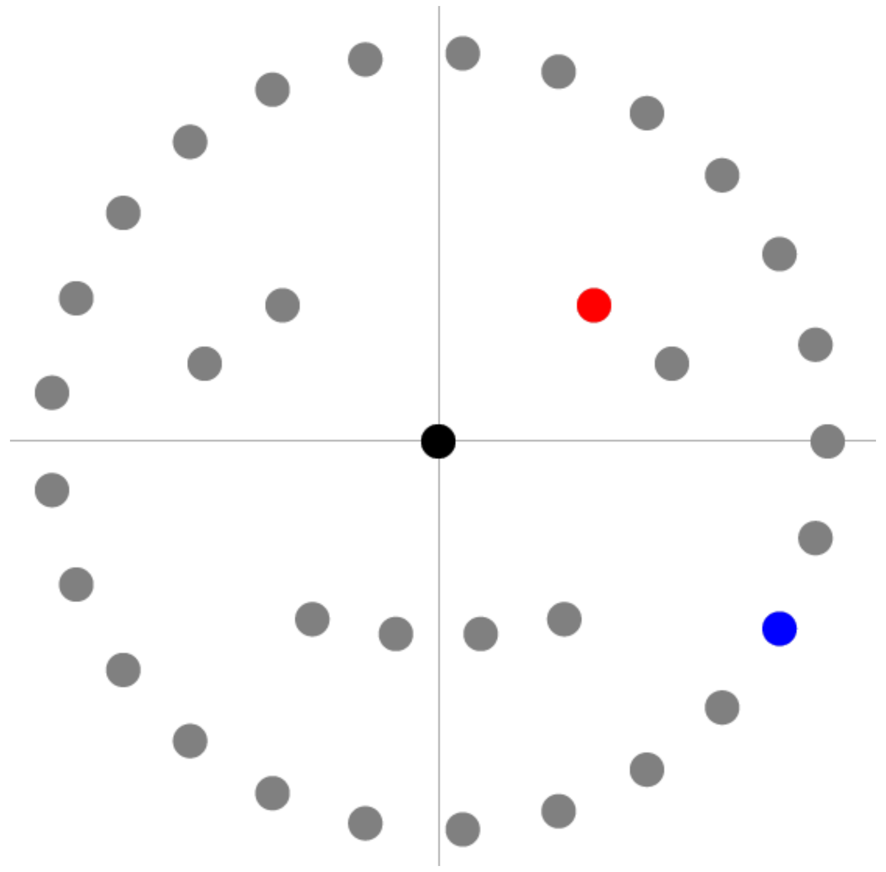
with $g = (\mathbf{x}, \mathbf{R}_\theta)$, $g' = (\mathbf{x}', \mathbf{R}_{\theta'})$.



multiply each element with g
(using group prod)

So… How to translate this to (G-)CNNs?

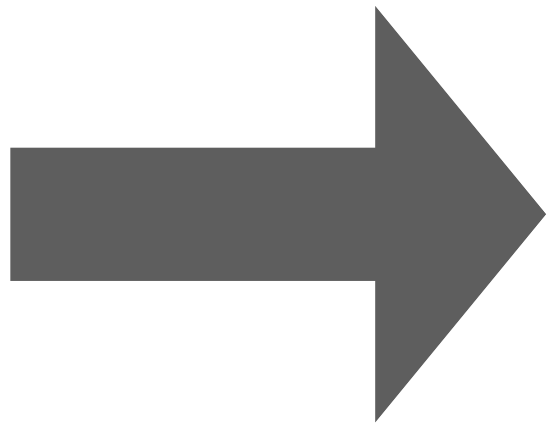
Set of points (group elements)



$$\{g_1, g_2, \ldots\} \subset G = (\mathbb{R}^2, +)$$

"A collection of parts in certain poses"

Transforms via group product



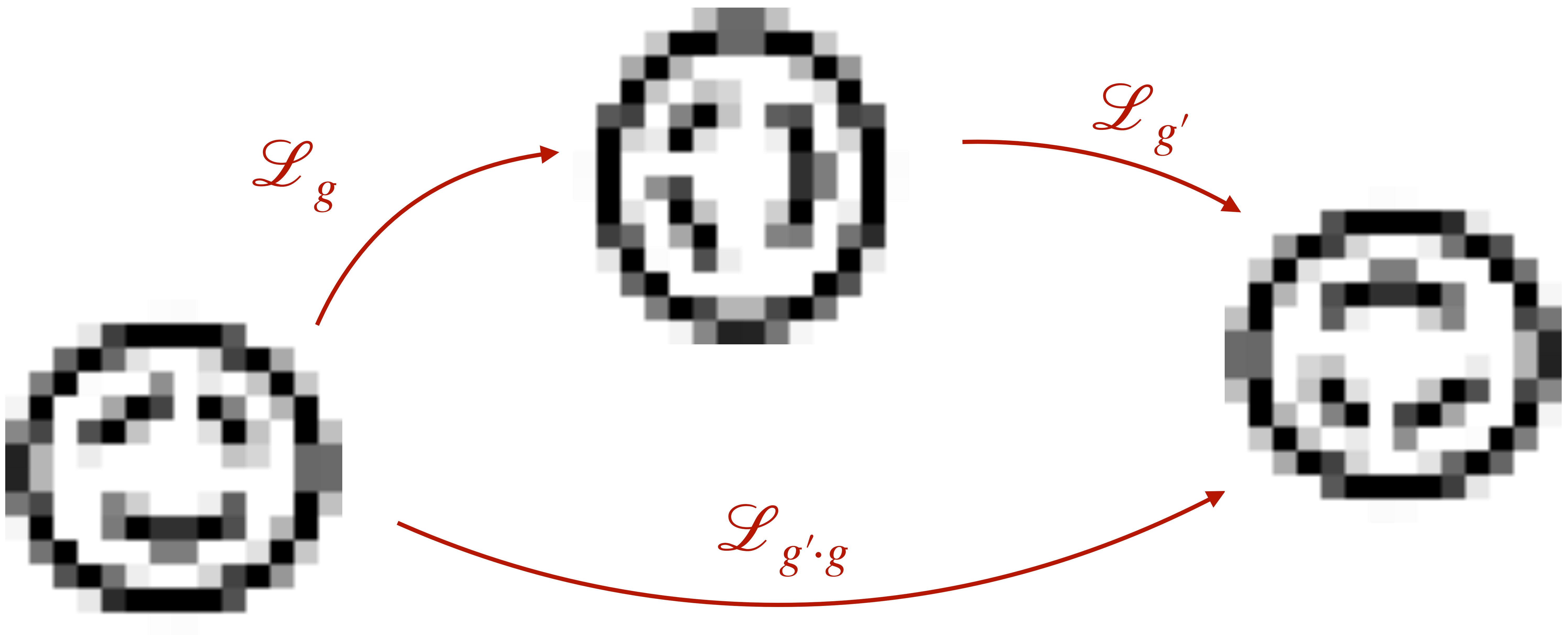
Convolution kernel

$$k \in \mathbb{L}_2(\mathbb{R}^2)$$

"Assigning weights to relative poses"

Transforms via group representations

Representations



A linear operator $\mathscr{L}_g$ that is parameterized by group elements $g \in G$ that transforms some object f (e.g. an image) is called a representation of G if it caries the group structure in the following way

$$\mathscr{L}_{g'}(\mathscr{L}_g(f)) = \mathscr{L}_{g'\cdot g}(f)$$

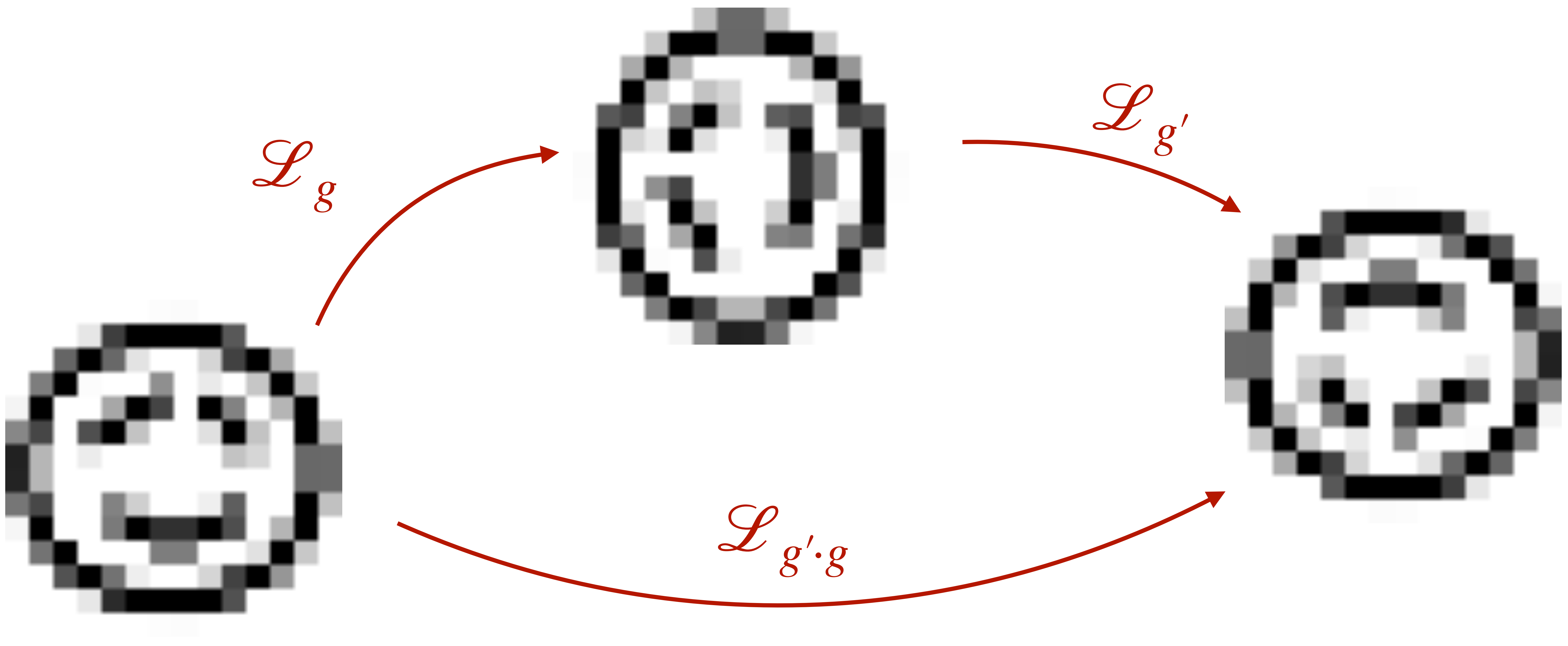
Left-regular representations

Example:

$f \in \mathbb{L}_2(\mathbb{R}^2)$
- a 2D image

$G = SE(2)$
- the roto-translation group

$\mathscr{L}_g(f)(\mathbf{y}) = f(\mathbf{R}_\theta^{-1}\mathbf{y} - \mathbf{x})$
- a roto-translation of the image



The left-regular representation of G transforms functions by acting on the domain on which they are defined via

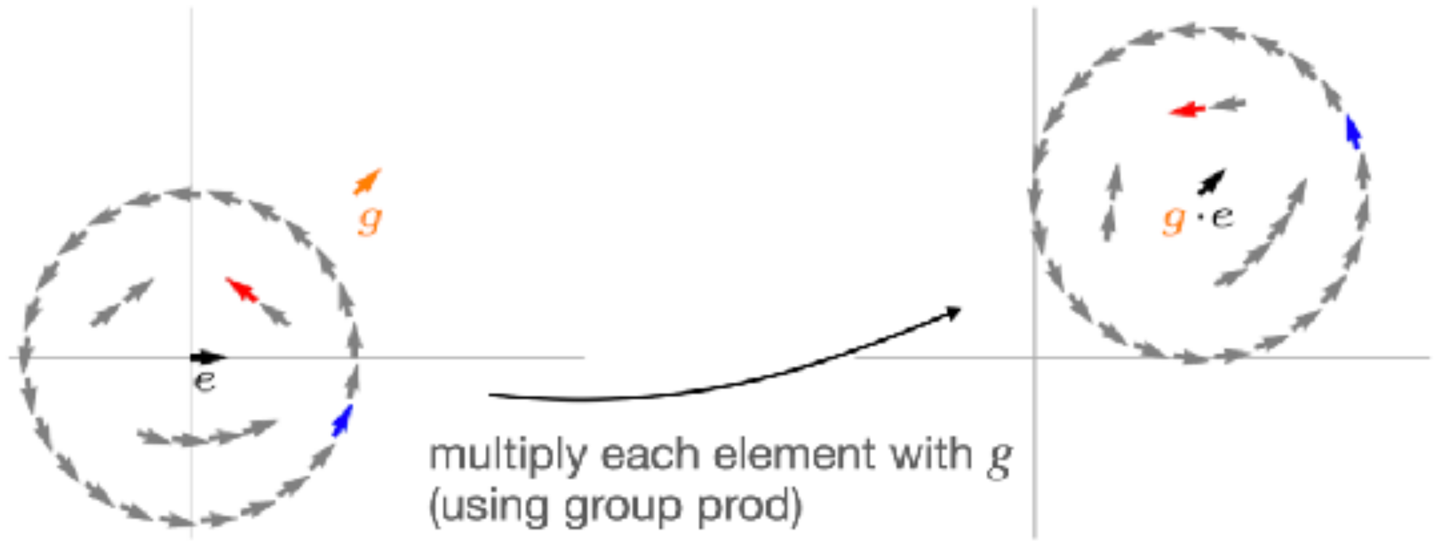
$$\mathscr{L}_g(f)(\mathbf{y}) = f(g^{-1} \odot \mathbf{y})$$

"group action" equals group product when $X = G$

Group actions

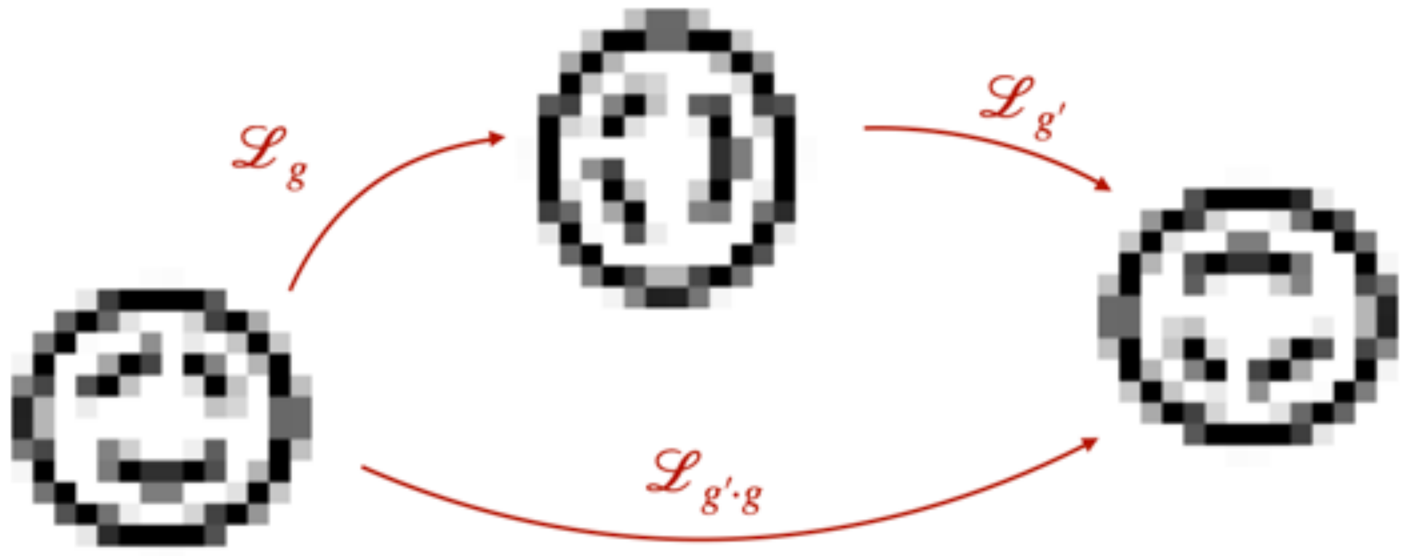
Group product (the action on G)

$g \cdot g'$ $\quad$ gg'



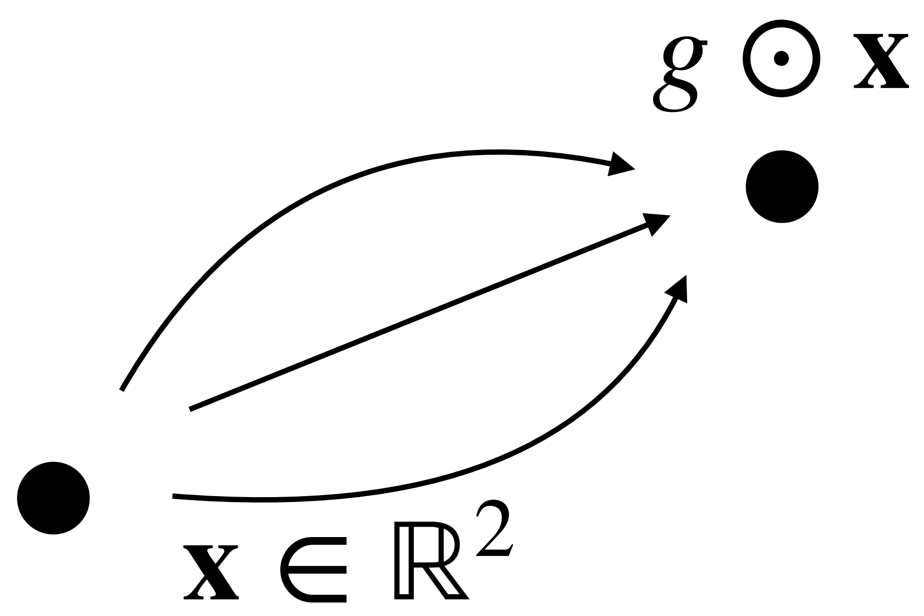
Left regular representation (the action on $\mathbb{L}_2(X)$)

$\mathscr{L}_g f$ $\quad$ gf



Group action (the action on $\mathbb{R}^d$)

$g \odot \mathbf{x}$ $\quad$ $g\mathbf{x}$

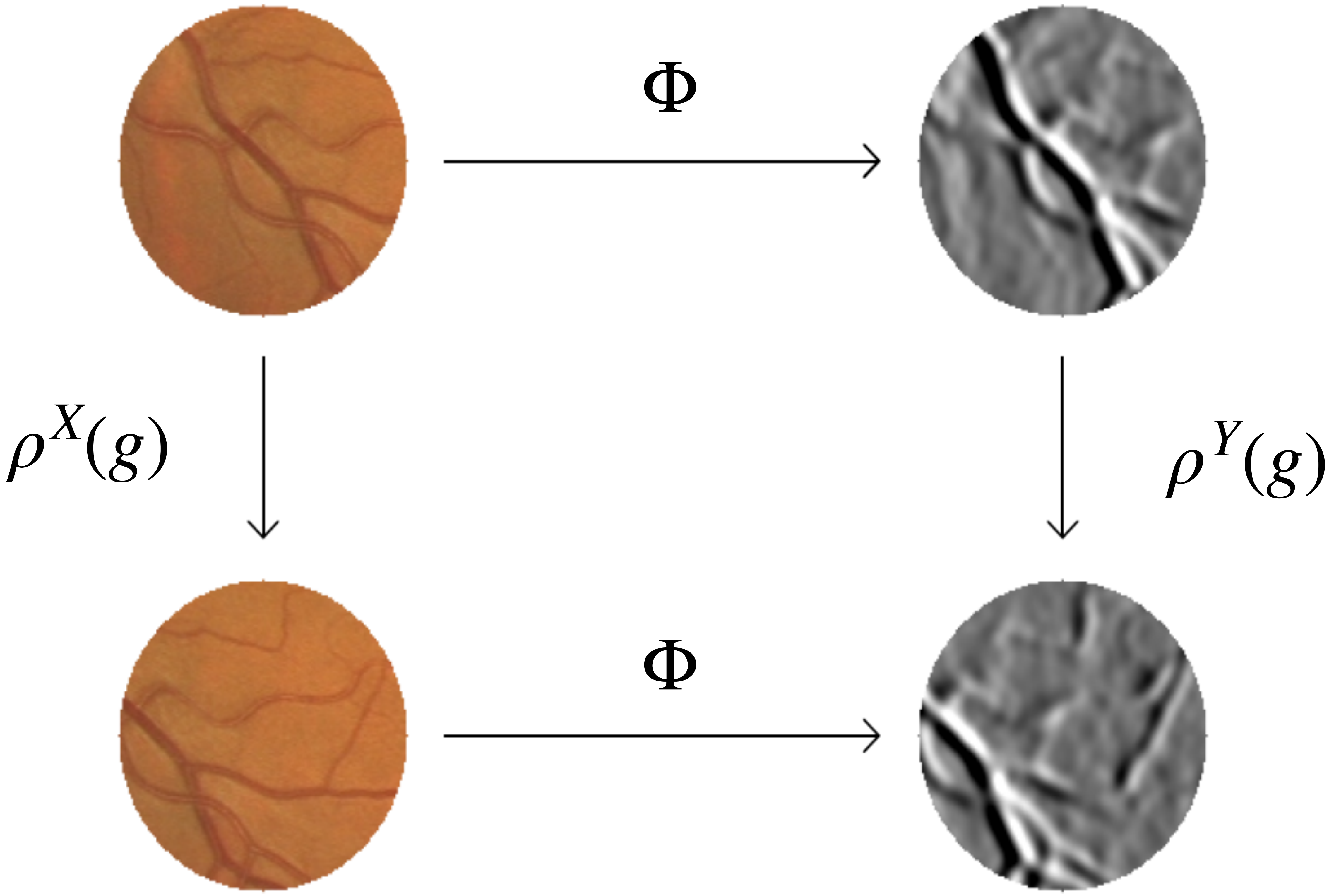


Equivariance

$$\rho^Y(g) \circ \Phi = \Phi \circ \rho^X(g)$$

$\Phi : X \to Y$

ρ^X and ρ^Y actions of G on X and Y



Content of this talk

1. Why do we want equivariant learning models?
   - Geometric guarantees + weight sharing/sample efficiency

2. A group theoretical view on **recognition by components** (capsule nets)
   - Group theoretical prerequisites (**group product** and **representations**)
   - **Group convolutions** perform pattern recognition by components

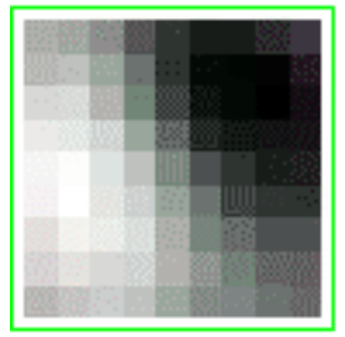
3. Experimental examples

4. Theorem: Linear maps between feature maps are equivariant iff they are group convolutions

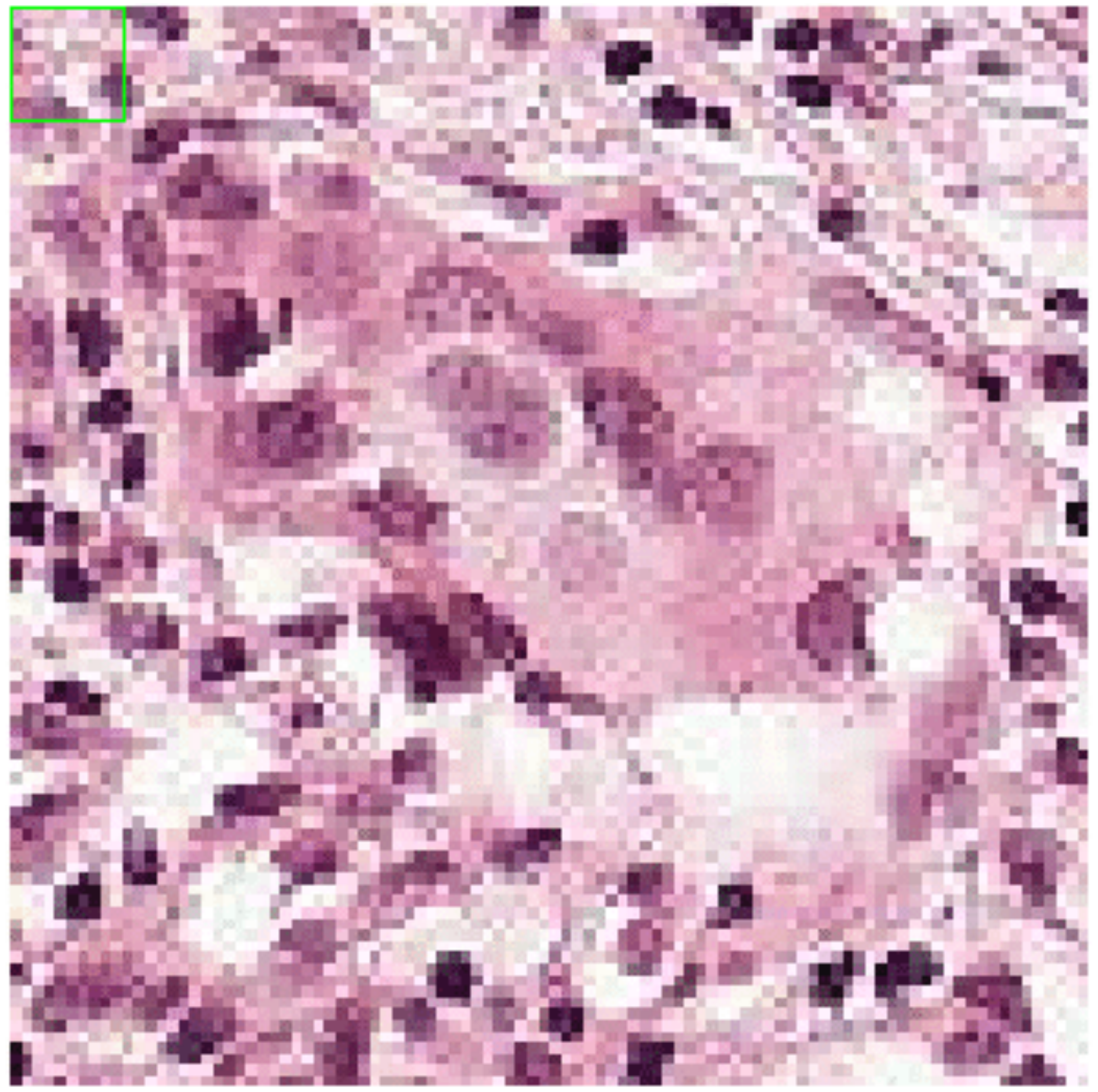
Are convolutions with reflected conv kernels (and vice versa)

Cross-correlations

$$(k \star_{\mathbb{R}^2} f)(\mathbf{x}) = \int_{\mathbb{R}^2} k(\mathbf{x}' - \mathbf{x}) f(\mathbf{x}') \mathrm{d}\mathbf{x}'$$



$\star_{\mathbb{R}^2}$



$=$

k
2D convolution kernel

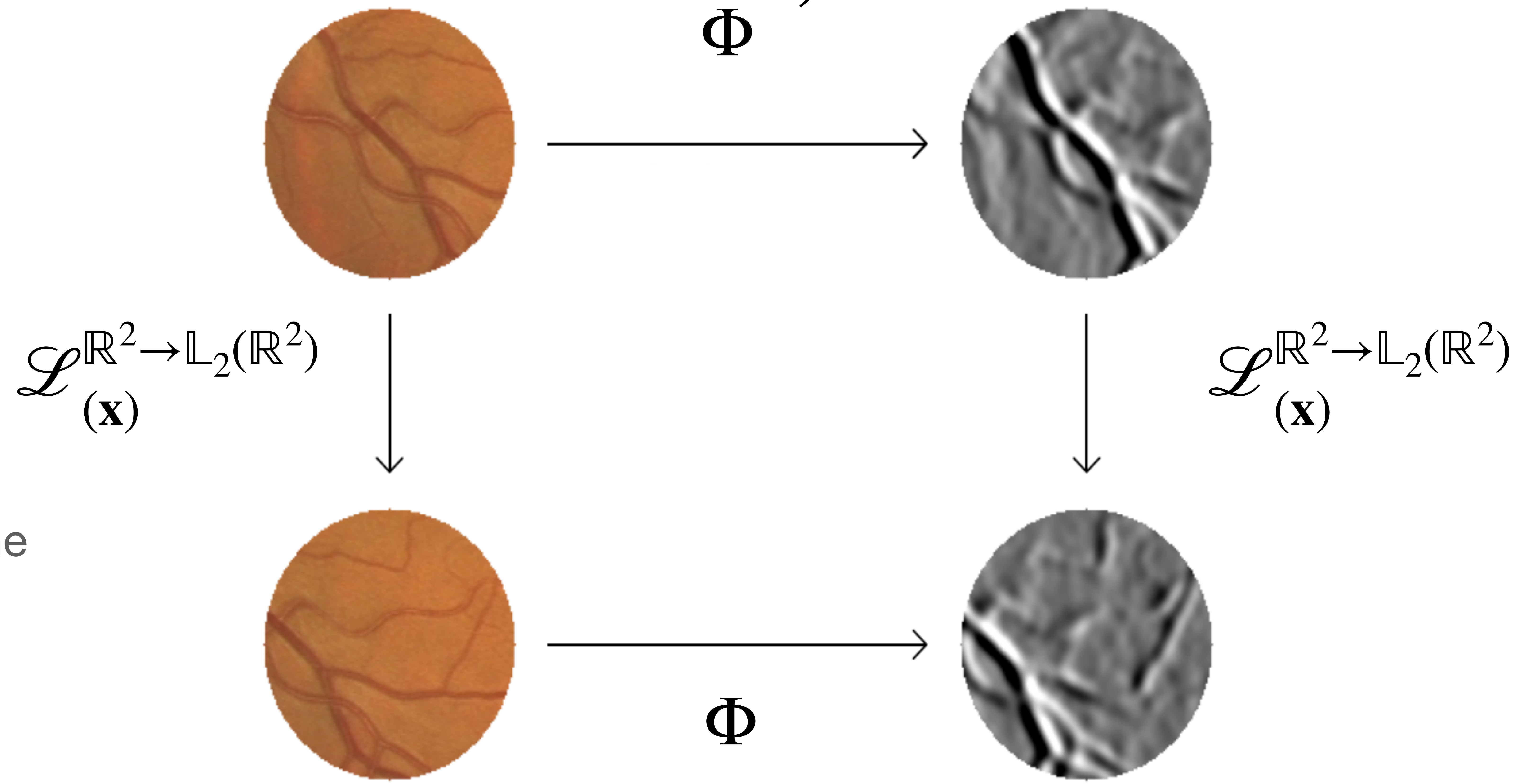
f^{in}
2D feature map

f^{out}
2D feature map (after ReLU)

Group **equivariance**

Convolutions/cross-correlations are translation equivariant

$$(k \star_{\mathbb{R}^2} f)(\mathbf{x}) = (\mathscr{L}_{(\mathbf{x})}^{\mathbb{R}^2 \rightarrow \mathbb{L}_2(\mathbb{R}^2)}\, k\, ,\, f)_{\mathbb{L}_2(\mathbb{R}^2)}$$



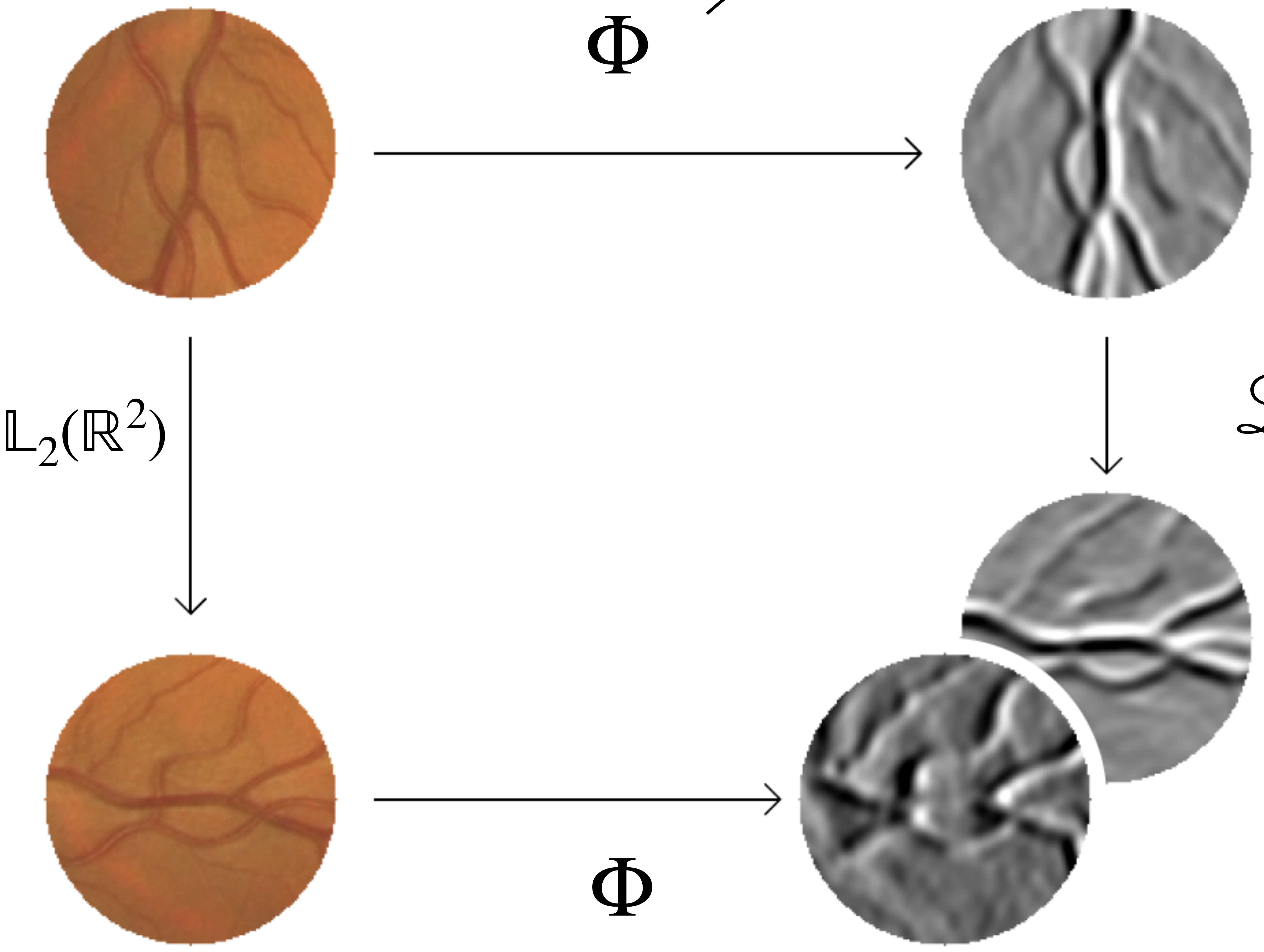
Representation of the translation group

Group **equivariance**

Convolutions are generally **not equivariant to roto-translations**

$$(k \star_{\mathbb{R}^2} f)(\mathbf{x}) = (\mathscr{L}^{\mathbb{R}^2 \to \mathbb{L}_2(\mathbb{R}^2)}_{(\mathbf{x})} k \, , f)_{\mathbb{L}_2(\mathbb{R}^2)}$$

Representation of the translation group



$$\mathscr{L}^{SO(2) \to \mathbb{L}_2(\mathbb{R}^2)}_{\theta}$$

Representation of the rotation group

$$\mathscr{L}^{SO(2) \to \mathbb{L}_2(\mathbb{R}^2)}_{g}$$

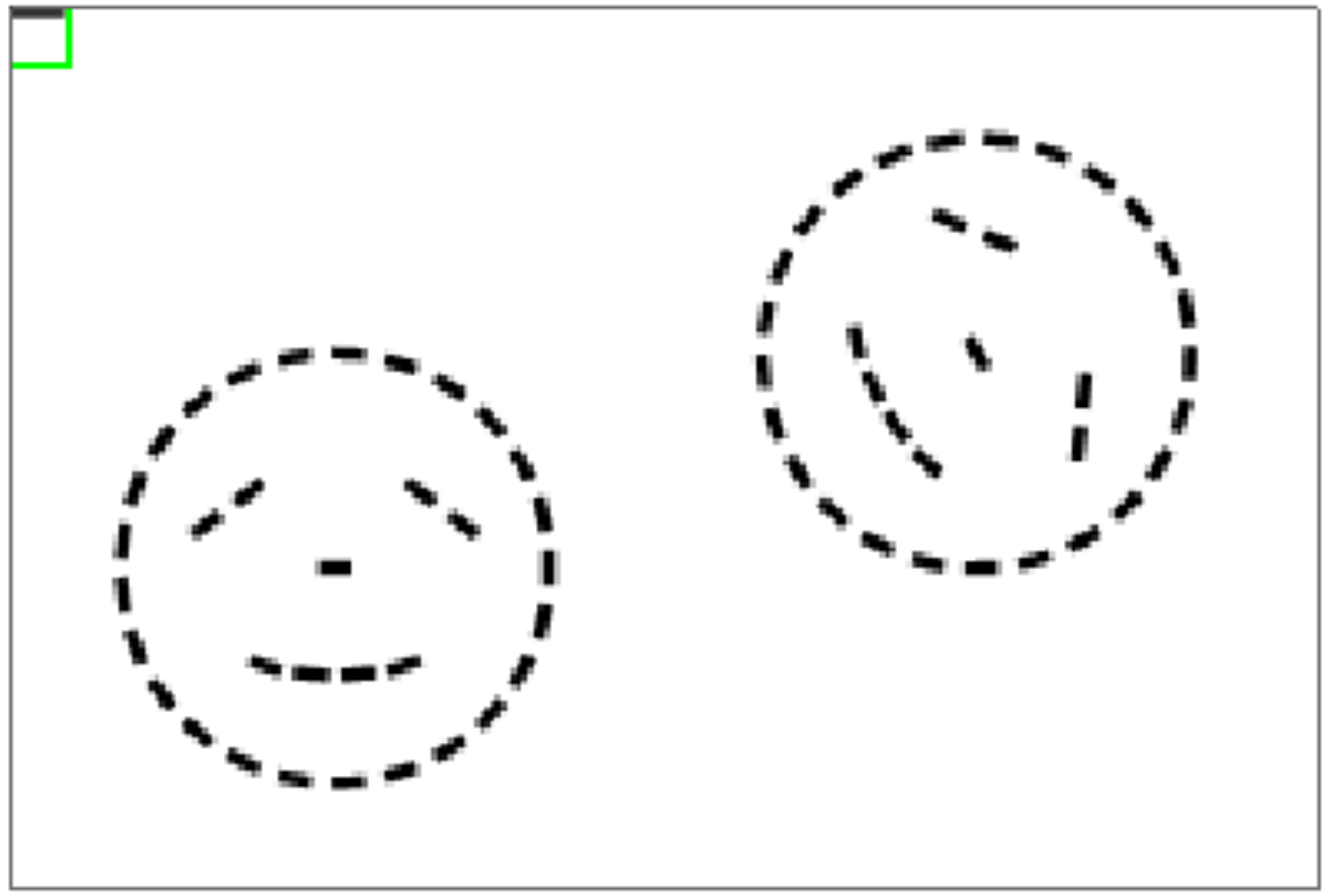
SE(2) equivariant cross-correlations

Representation of the roto-translation group!

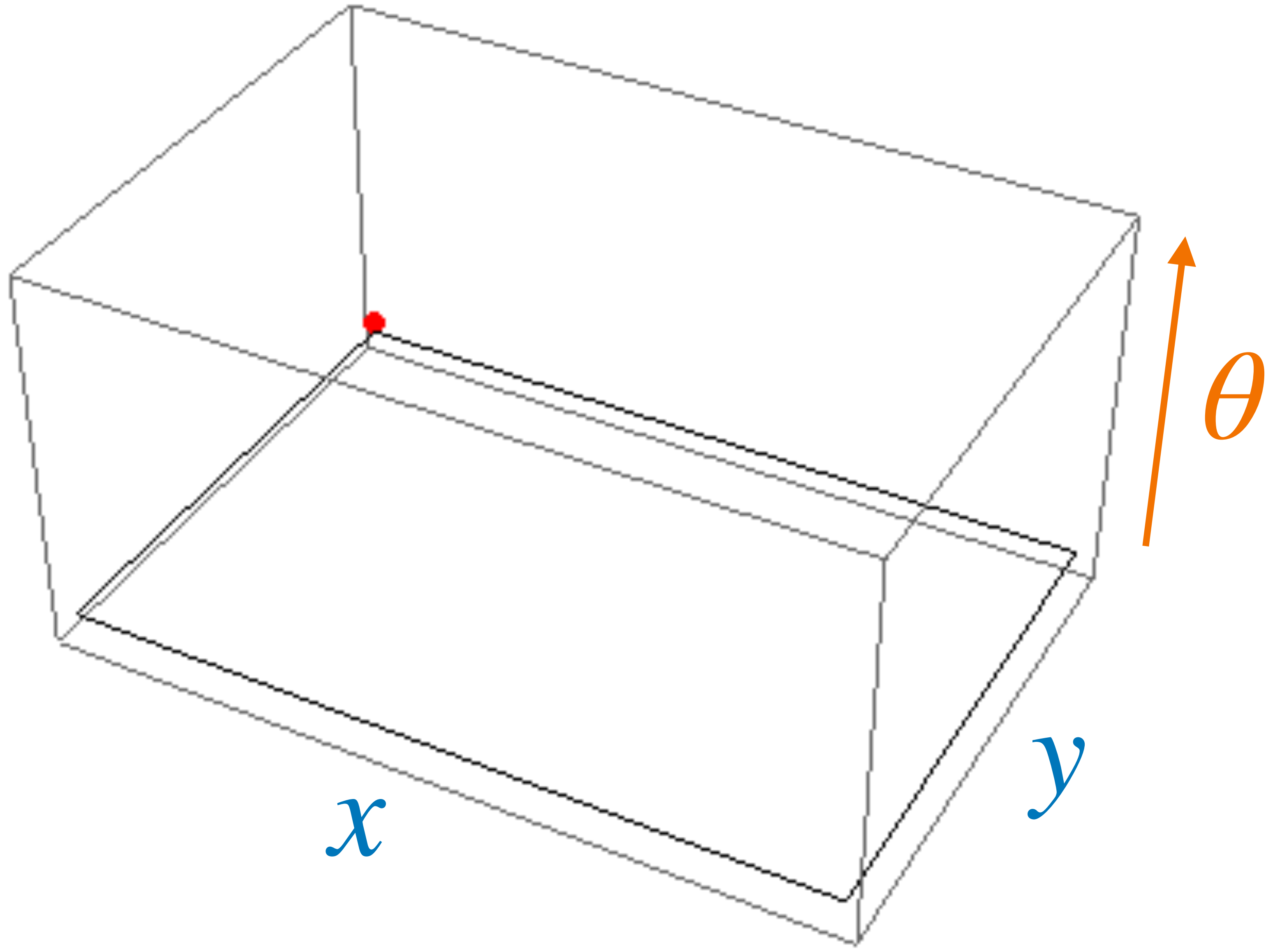
$k(\mathbf{R}_\theta^{-1}\mathbf{x}'-\mathbf{x})$

Lifting correlations: $(k \,\tilde{\star}\, f)(\mathbf{x}) = (\mathscr{L}_g^{SE(2)\to\mathbb{L}_2(\mathbb{R}^2)}\, k \,,\, f)_{\mathbb{L}_2(\mathbb{R}^2)}$

$\star_{\mathbb{R}^2}$



$=$



$\mathscr{L}_\theta^{SO(2)\to\mathbb{L}_2(\mathbb{R}^2)} k$

Rotated 2D convolution kernel

f^{in}

2D feature map

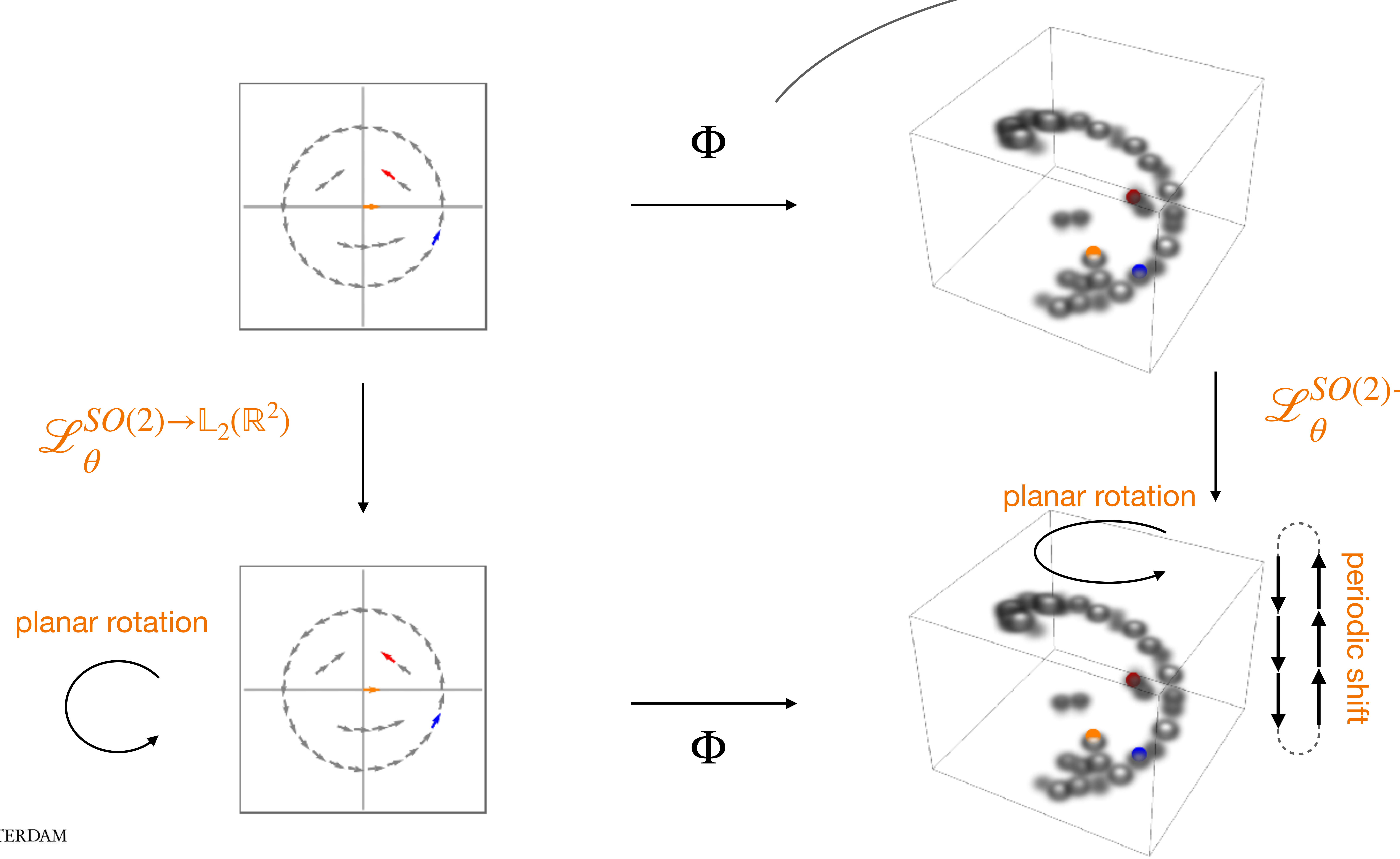
f^{out}

3D ($SE(2)$) feature map (after ReLU)

Group **equivariance**

SE(2) group **lifting convolutions** are roto-translation equivariant

$$(k \tilde{\star} f)(\mathbf{x}) = (\mathcal{L}_{\mathbf{x}}^{\mathbf{R}^2\to\mathbb{L}_2(\mathbb{R}^2)} \mathcal{L}_{\theta}^{SO(2)\to\mathbb{L}_2(\mathbb{R}^2)} k \, , f)_{\mathbb{L}_2(\mathbb{R}^2)}$$



What about subsequent layers?

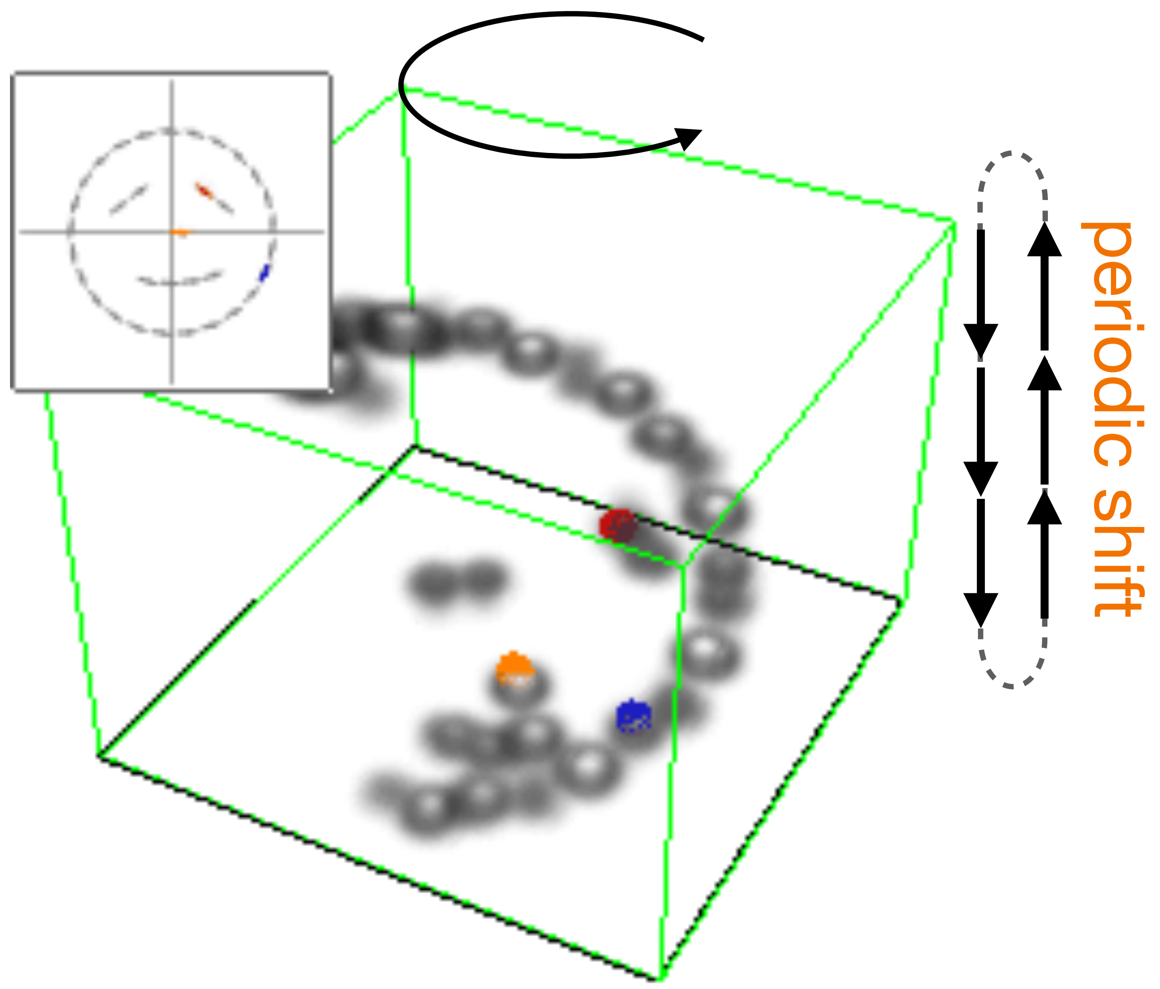
SE(2) equivariant cross-correlations

$k(\mathbf{R}_\theta^{-1}\mathbf{x}'-\mathbf{x},\mathbf{R}_{\theta'-\theta})$

Group correlations:

$$(k \star f)(\mathbf{x}) = (\mathscr{L}_g^{SE(2)\to\mathbb{L}_2(SE(2))}\, k\ ,\ f)_{\mathbb{L}_2(SE(2))} = (\overbrace{\mathscr{L}_{\mathbf{x}}^{\mathbf{R}^2\to\mathbb{L}_2(SE(2))}\mathscr{L}_\theta^{SO(2)\to\mathbb{L}_2(SE(2))}\, k}\ ,\ f)_{\mathbb{L}_2(SE(2)}$$

translation *rotation*



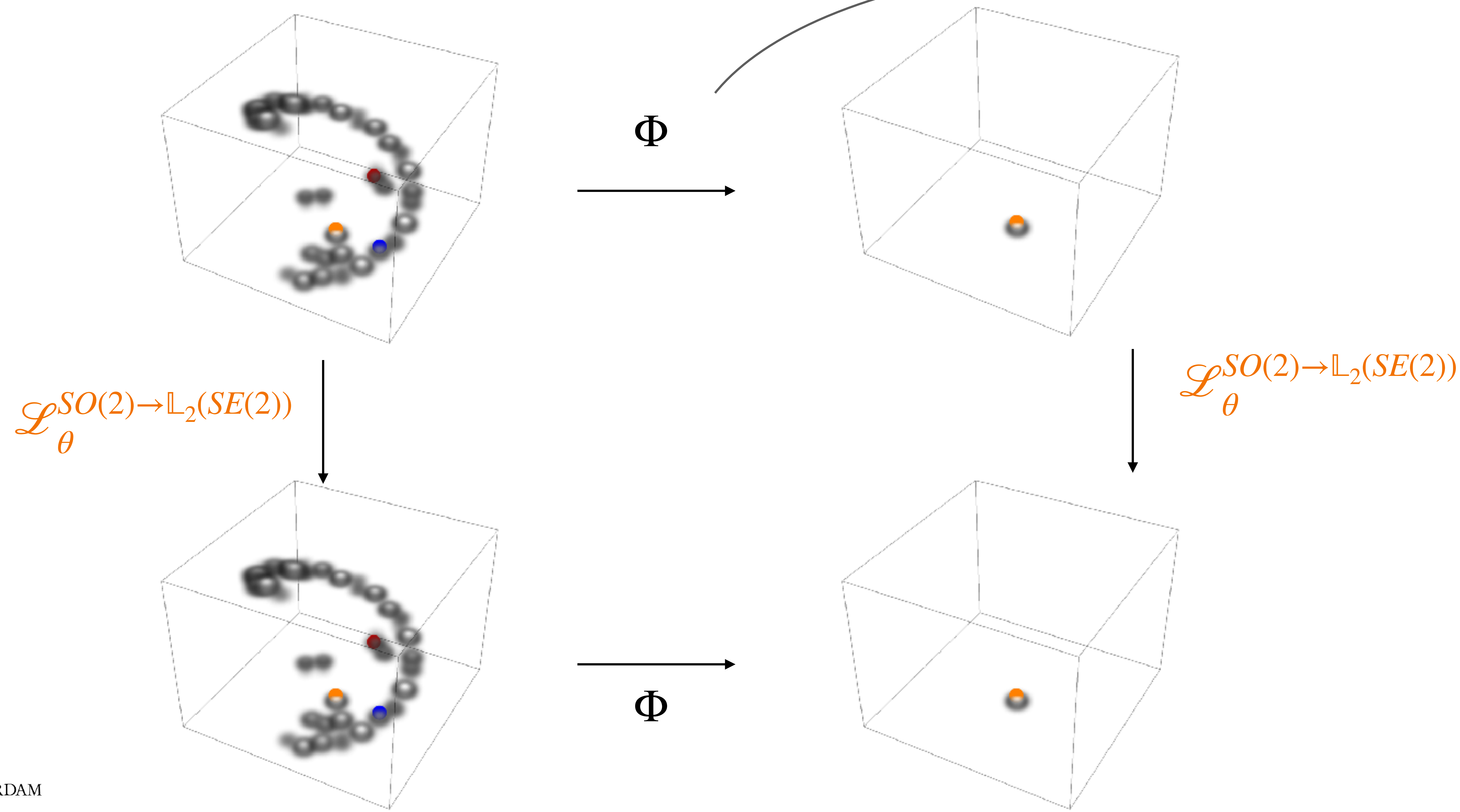
$\mathscr{L}_\theta^{SO(2)\to\mathbb{L}_2(SE(2)}k$

Rotated SE(2) convolution kernel

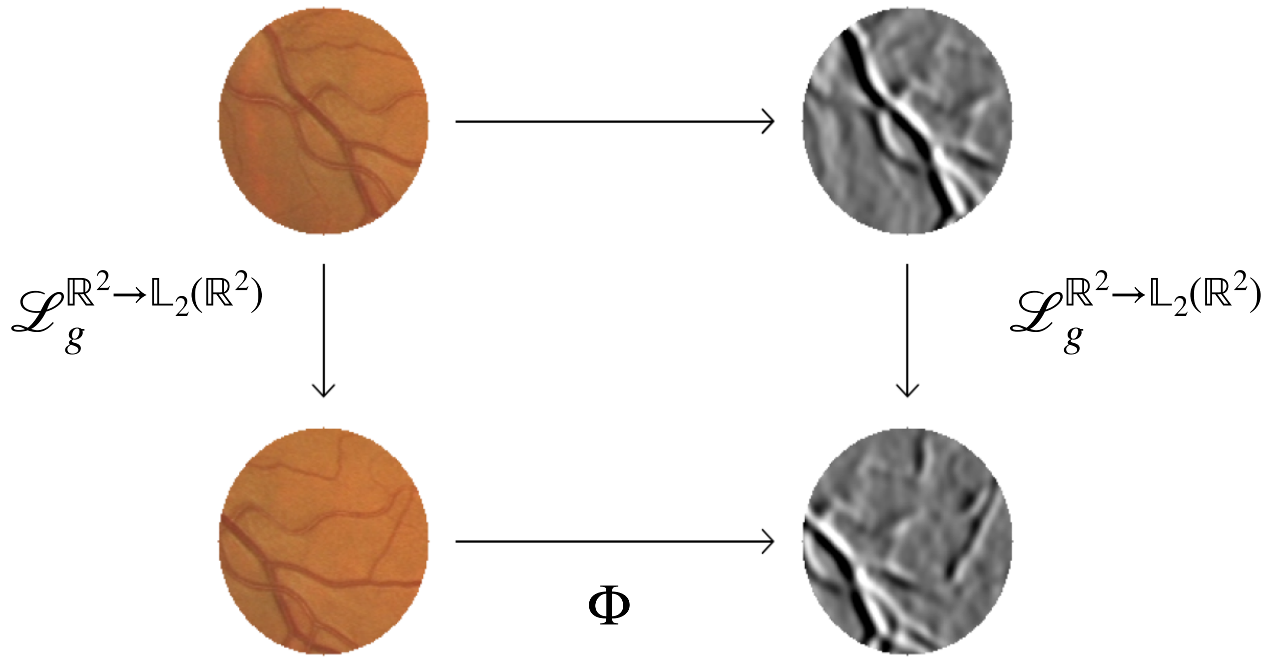
Group **equivariance**

SE(2) group convolutions are roto-translation equivariant

$$(k \star f)(\mathbf{x}) = (\mathscr{L}_{\mathbf{x}}^{\mathbb{R}^2\to\mathbb{L}_2(SE(2))}\mathscr{L}_{\theta}^{SO(2)\to\mathbb{L}_2(SE(2))}\, k \,,\, f)_{\mathbb{L}_2(SE(2))}$$



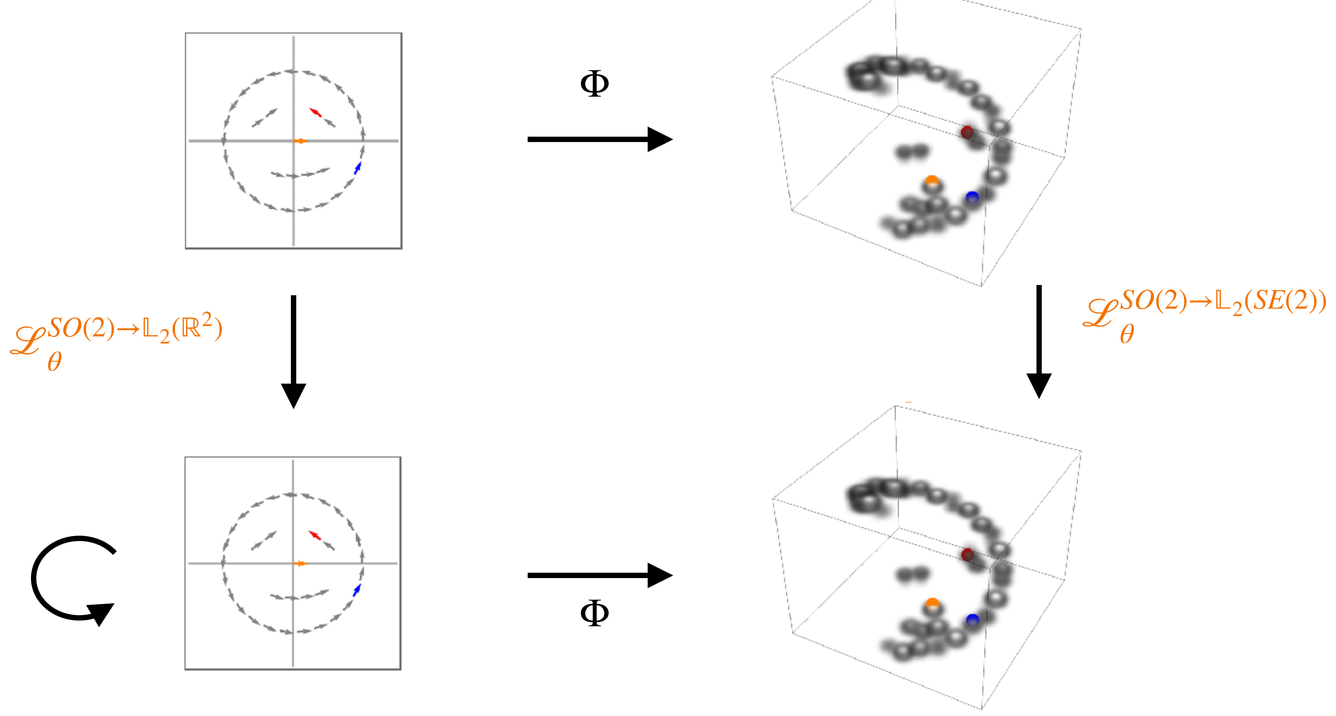
2D cross-correlation (translation equivariant)



$$(k \star_{\mathbb{R}^2} f)(\mathbf{x}) = (\mathscr{L}_{\mathbf{x}}^{\mathbb{R}^2\to\mathbb{L}_2(\mathbb{R}^2)} k, f)_{\mathbb{L}_2(\mathbb{R}^2)}$$

$$= \int_{\mathbb{R}^2} k(\mathbf{x}'-\mathbf{x}) f(\mathbf{x}') d\mathbf{x}'$$

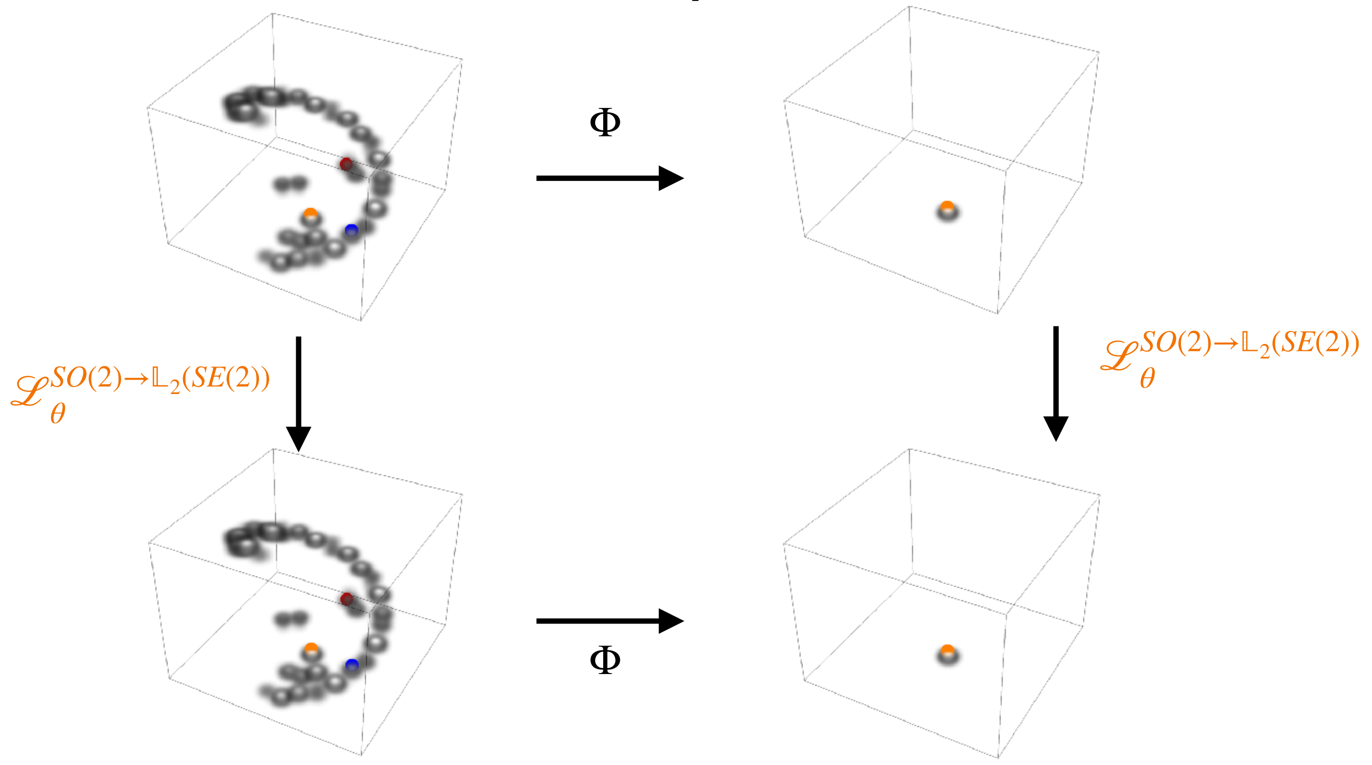
SE(2) lifting correlations (roto-translation equivariant)



$$(k \tilde{\star} f)(\mathbf{x}) = (\mathscr{L}_g^{SE(2)\to\mathbb{L}_2(\mathbb{R}^2)} k , f)_{\mathbb{L}_2(\mathbb{R}^2)}$$

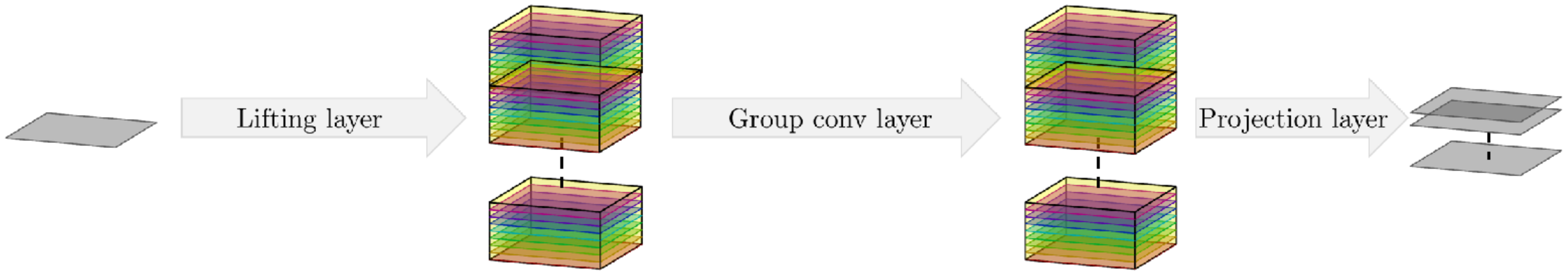
$$= \int_{\mathbb{R}^2} k(\mathbf{R}_\theta^{-1}\mathbf{x}'-\mathbf{x}) f(\mathbf{x}') d\mathbf{x}'$$

SE(2) G-correlations (roto-translation equivariant)



$$(k \tilde{\star} f)(\mathbf{x}) = (\mathscr{L}_g^{SE(2)\to\mathbb{L}_2(SE(2))} k , f)_{\mathbb{L}_2(SE(2))}$$

$$= \int_{\mathbb{R}^2}\int_{S^1} k(\mathbf{R}_\theta^{-1}\mathbf{x}'-\mathbf{x}, \theta'-\theta \mod 2\pi) f(\mathbf{x}',\theta') d\mathbf{x}'$$


Lifting layer
Group conv layer
Projection layer

Content of this talk

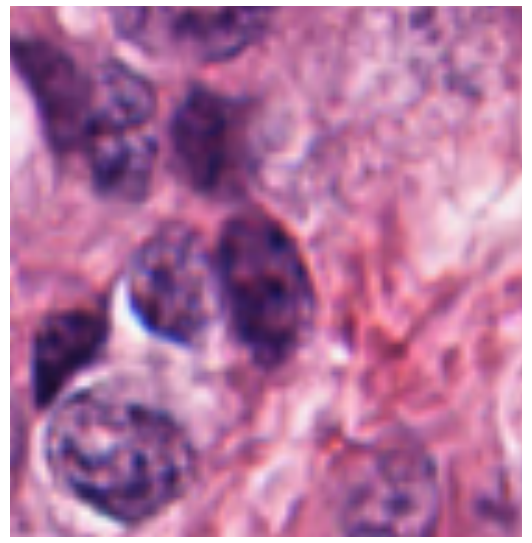
1. Why do we want equivariant learning models?
   - Geometric guarantees + weight sharing/sample efficiency

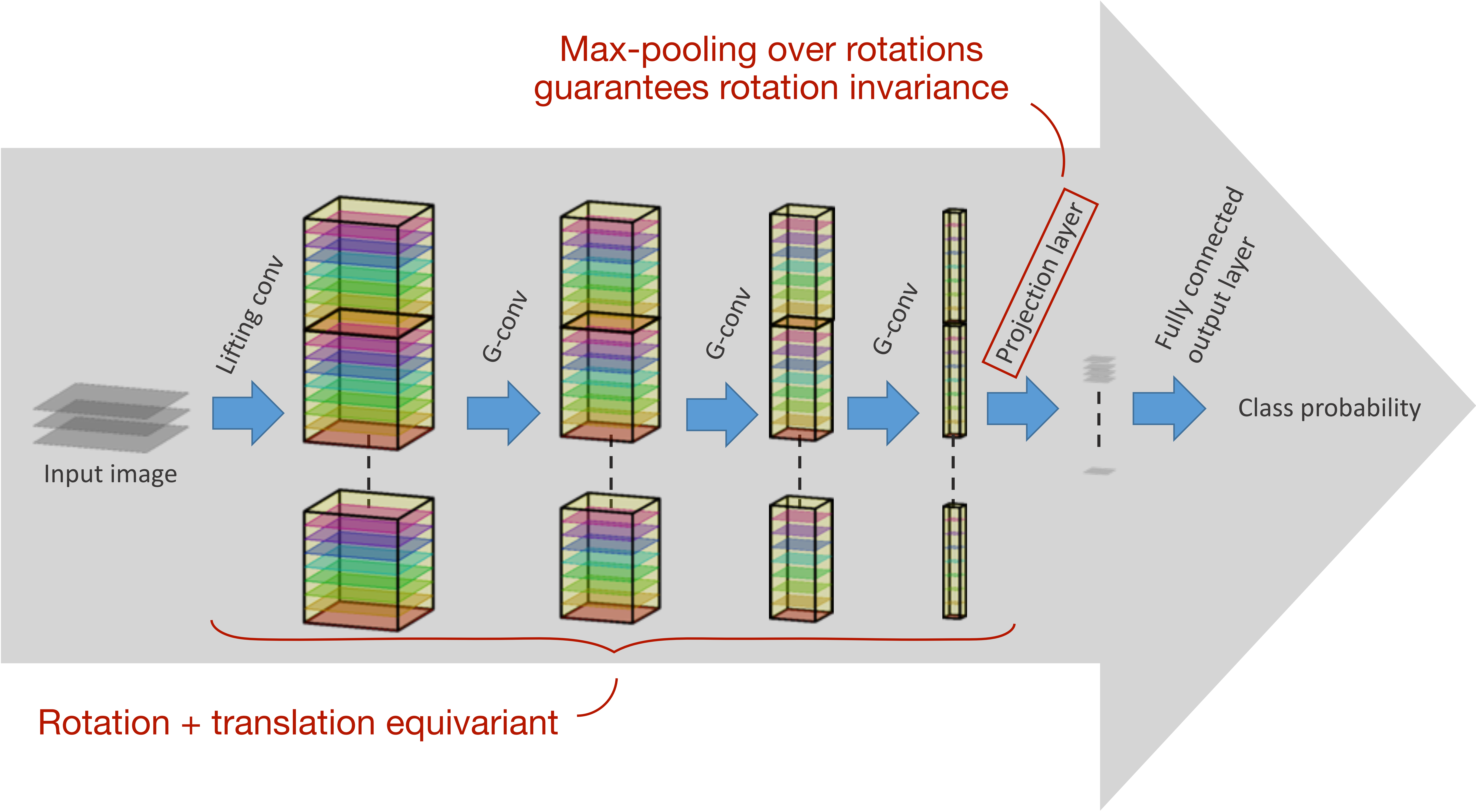
2. A group theoretical view on **recognition by components** (capsule nets)
   - Group theoretical prerequisites (**group product** and **representations**)
   - **Group convolutions** perform pattern recognition by components

3. Experimental examples

4. Theorem: Linear maps between feature maps are equivariant iff they are group convolutions

Architecture for rotation invariant mitotic cell detection

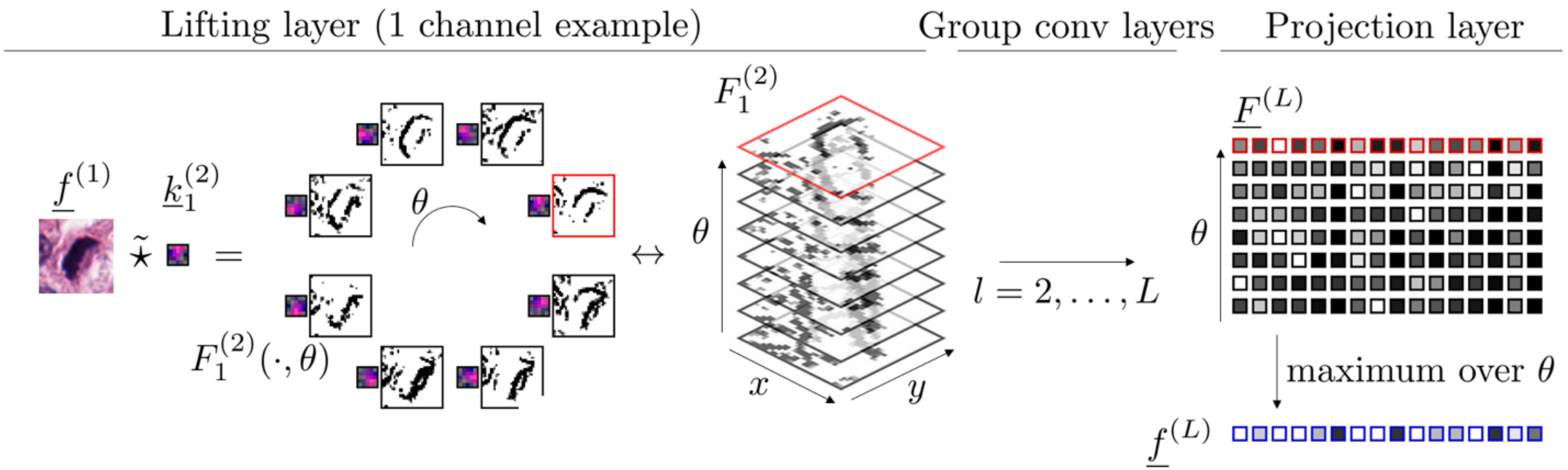




“normal” (0)
vs
“mitotic” (1)

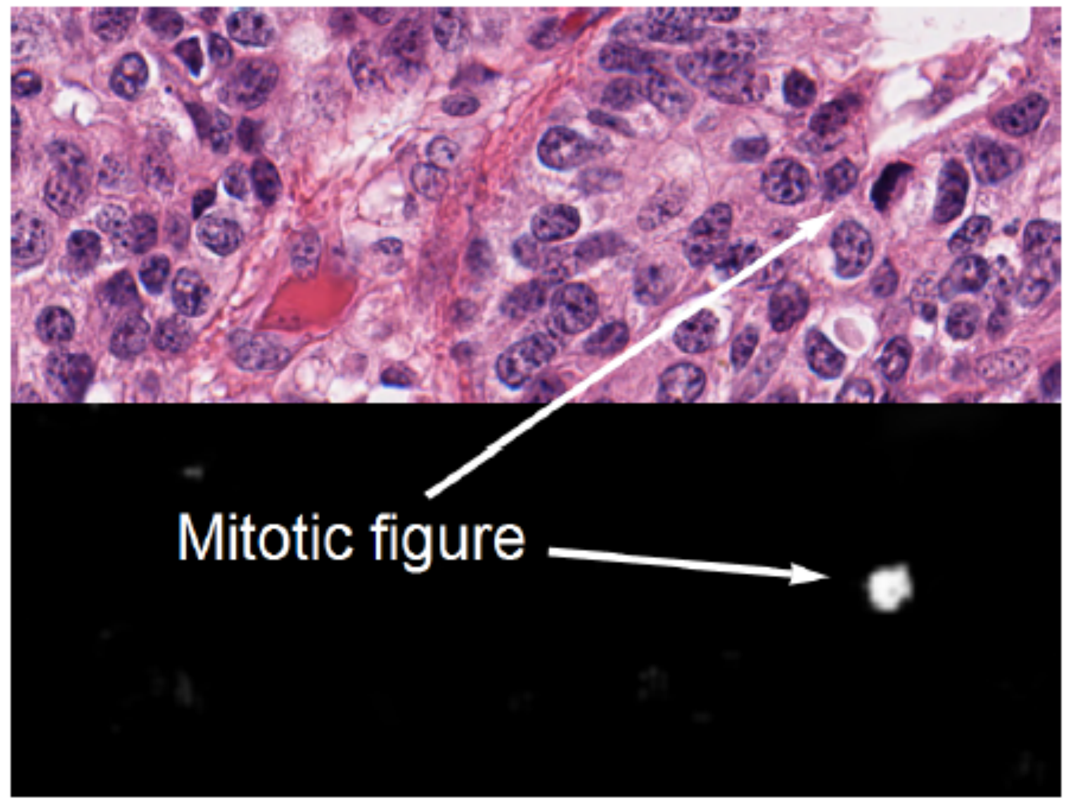
Bekkers & Lafarge et al. MICCAI 2018

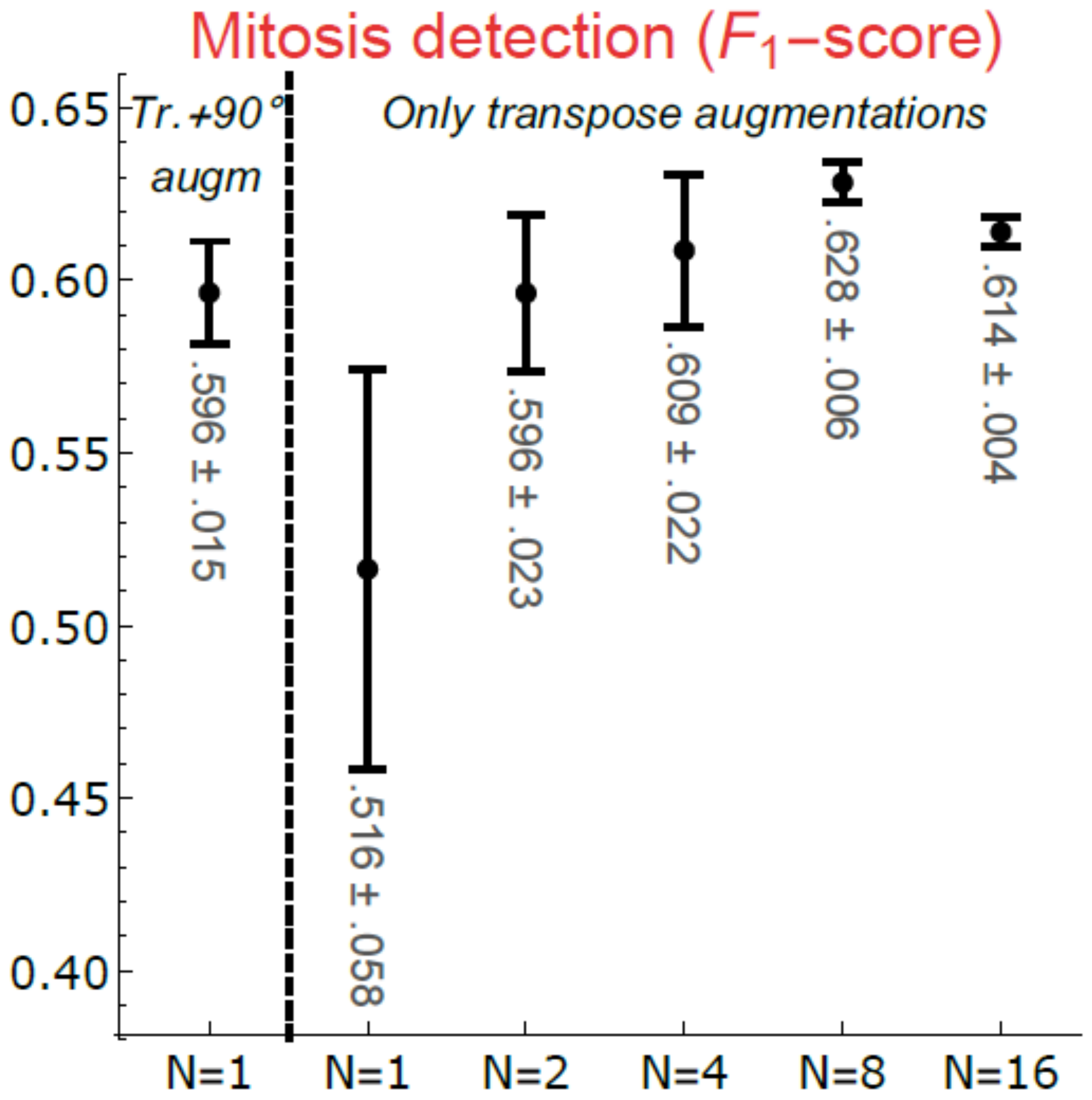
Architecture for rotation invariant mitotic cell detection



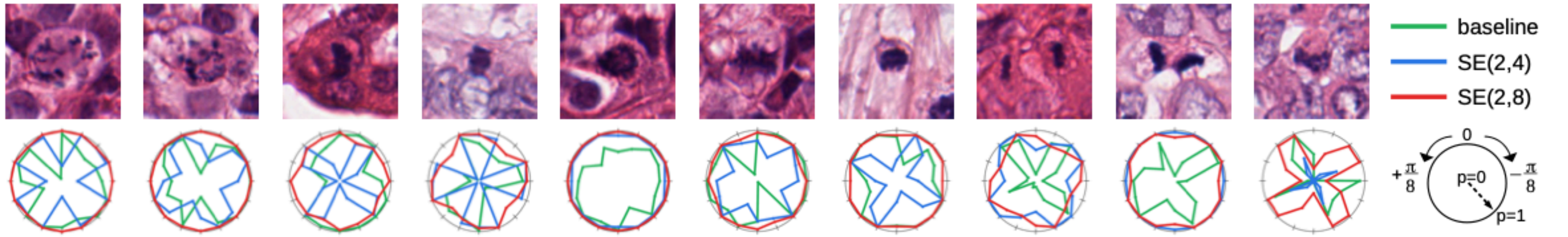
Architecture for rotation invariant mitotic cell detection

Bekkers & Lafarge et al. MICCAI 2018





Lafarge et al. MedIA 2020



G-CNNs guarantee geometric stability. They are ***robust to input distortions****, regular CNNs aren't…*

G-CNNs are more ***sample efficient****!*
G-CNNs (25% data) > CNNs (100% data)

G-CNNs without data-augmentation
outperform
CNNs with data-augmentation

Lafarge et al. ArXiv/MedIA 2020

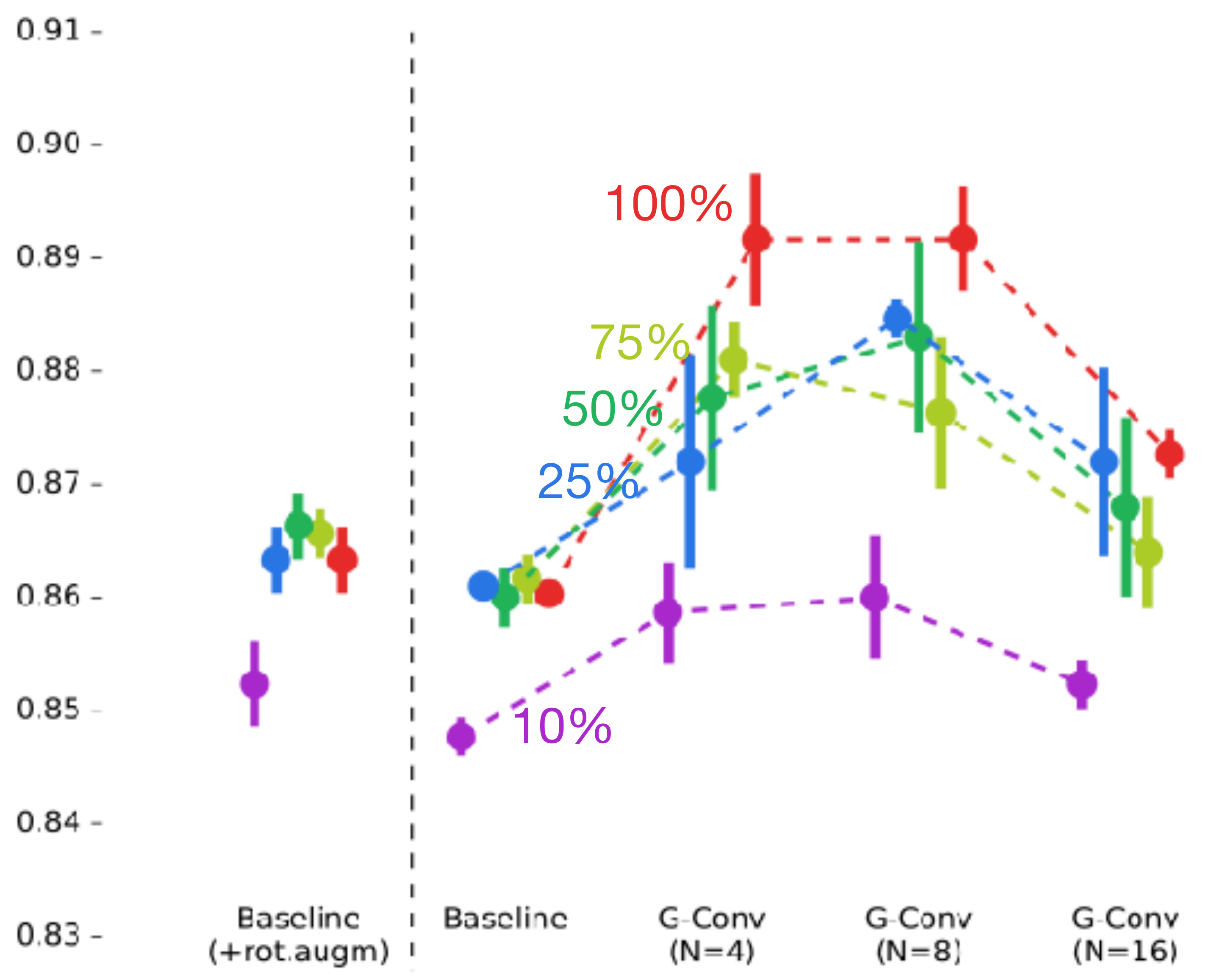
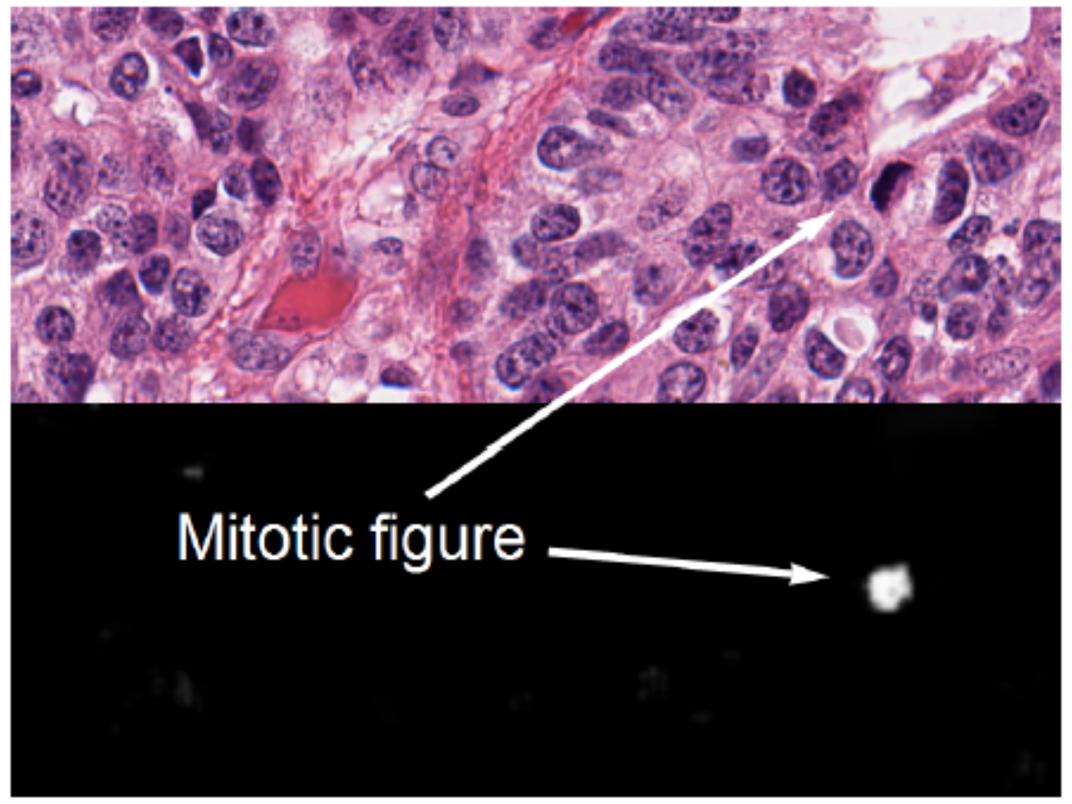
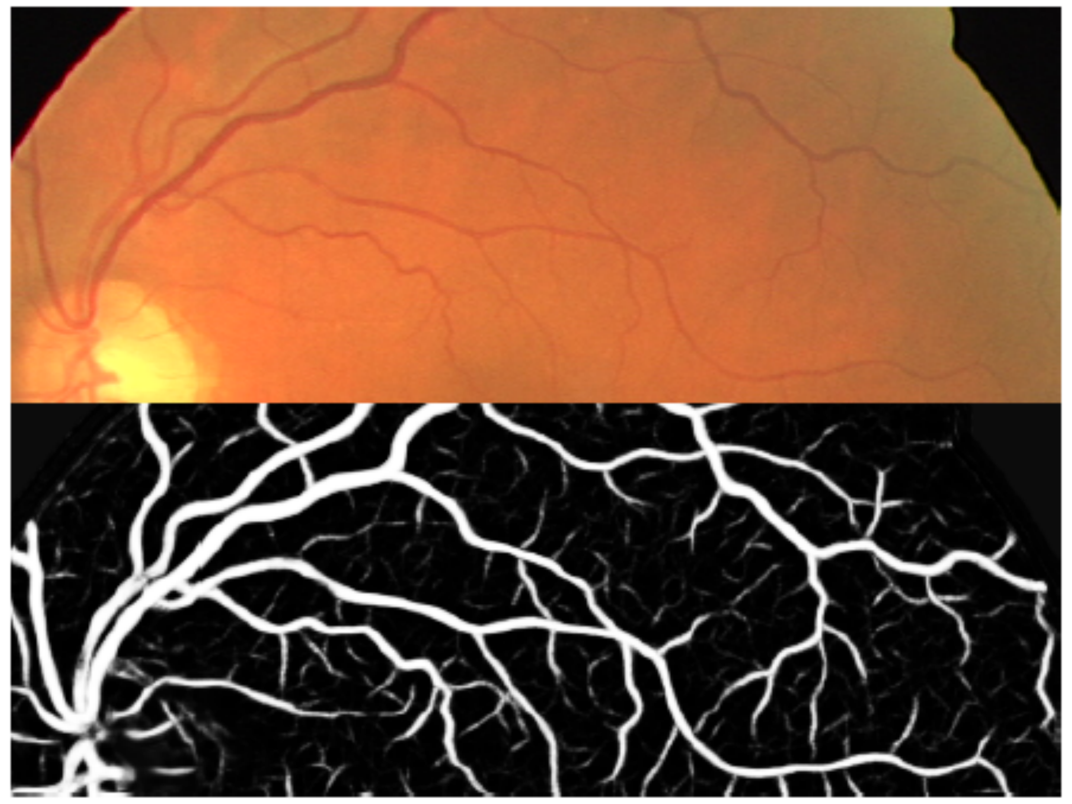


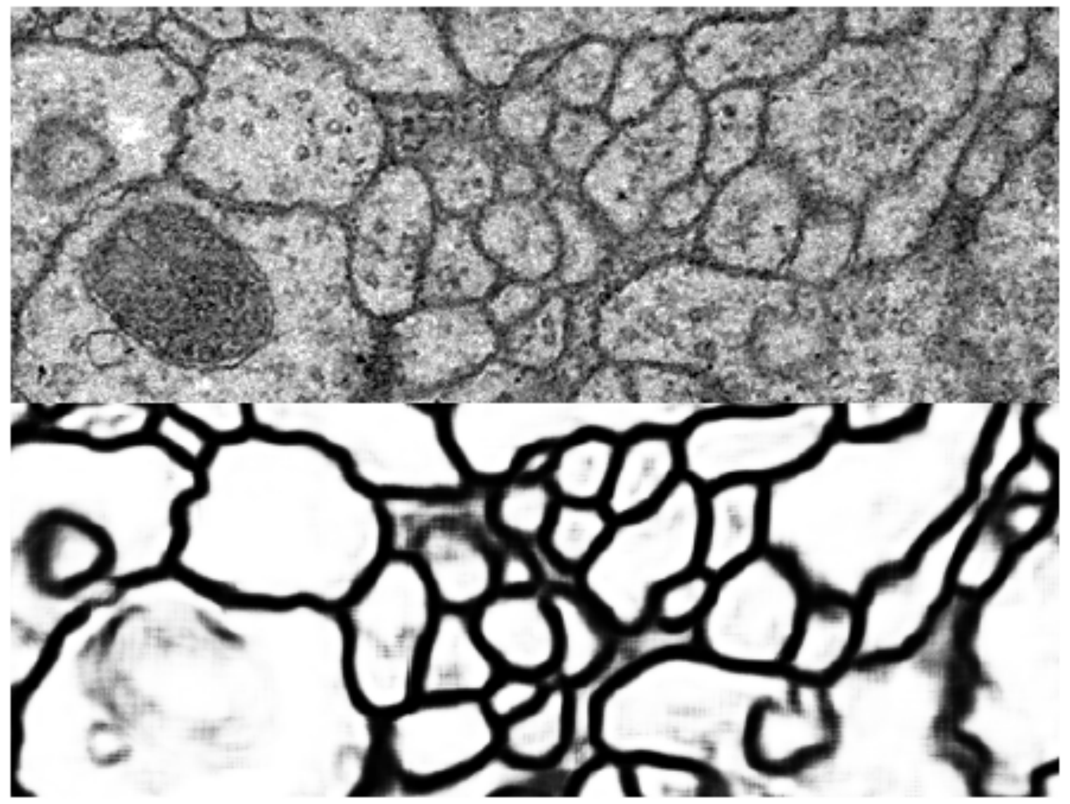
Figure 7: Mean and Standard Deviation plots summarizing the accuracy of the tumor classification models. Mean ± standard deviation is indicated. Color identifies the different data regime (red: 100%; lime: 75%; green: 50%; blue: 25%; purple: 10%).

Experiments in medical image analysis

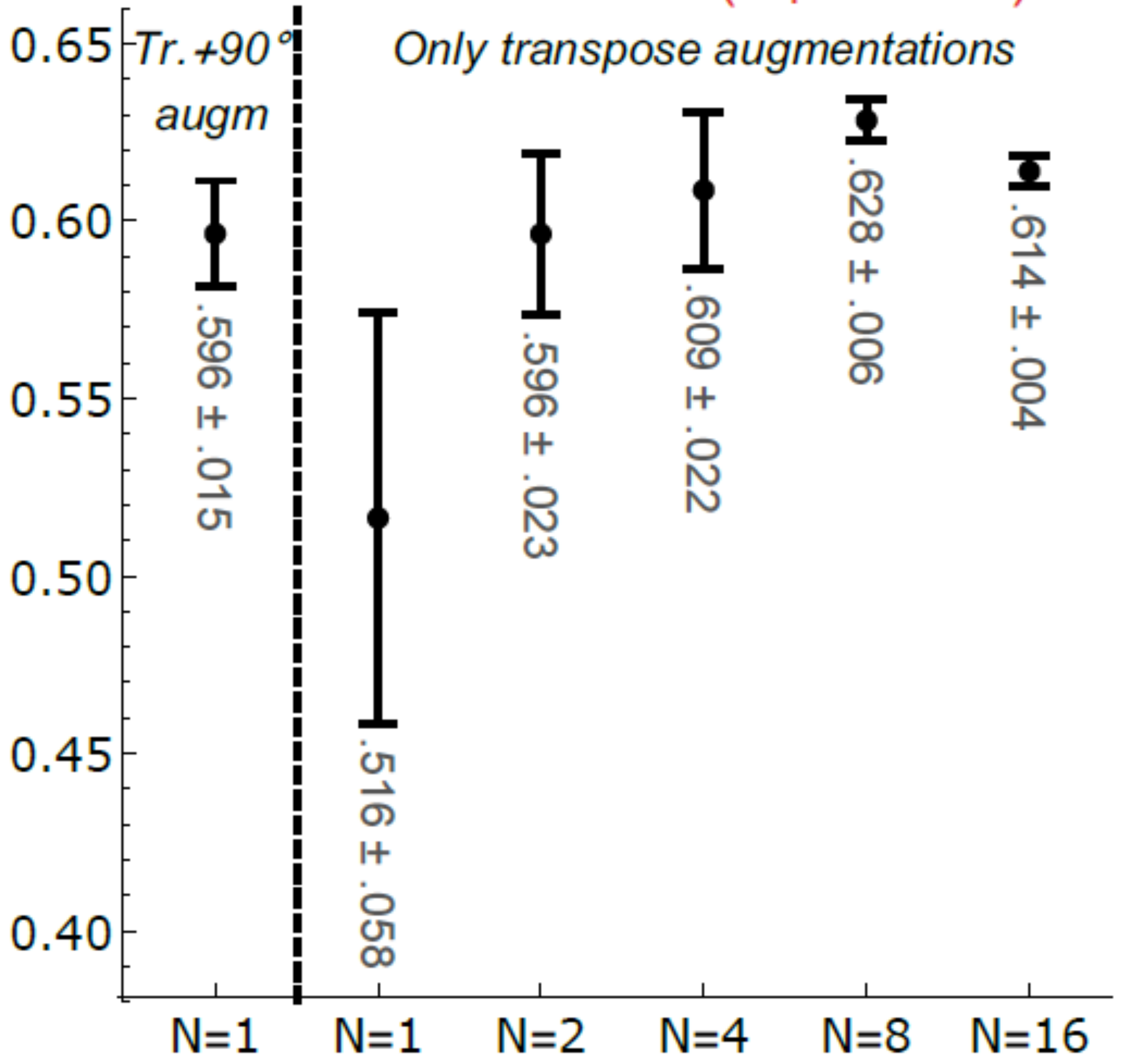
Bekkers & Lafarge et al. MICCAI 2018



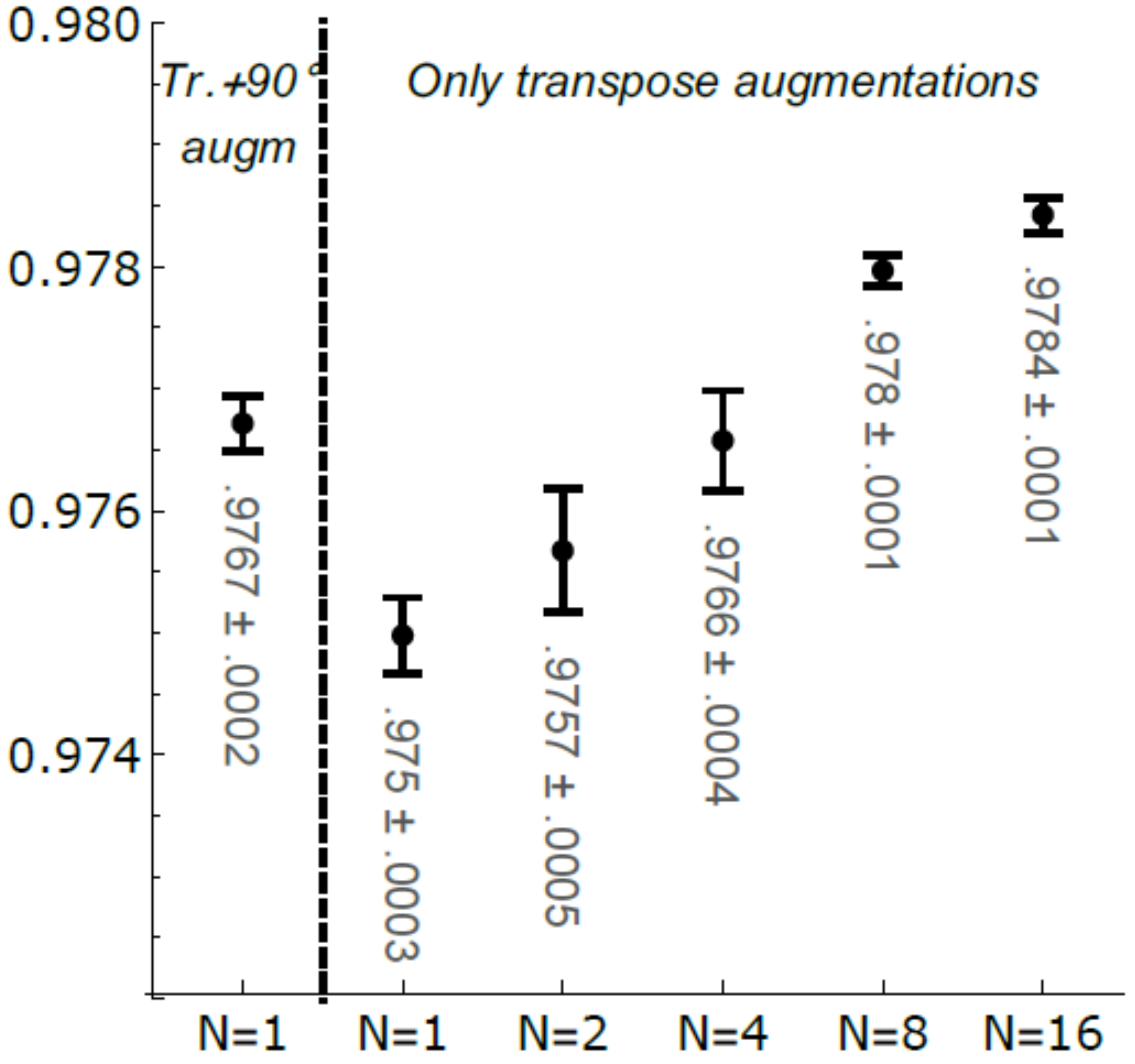




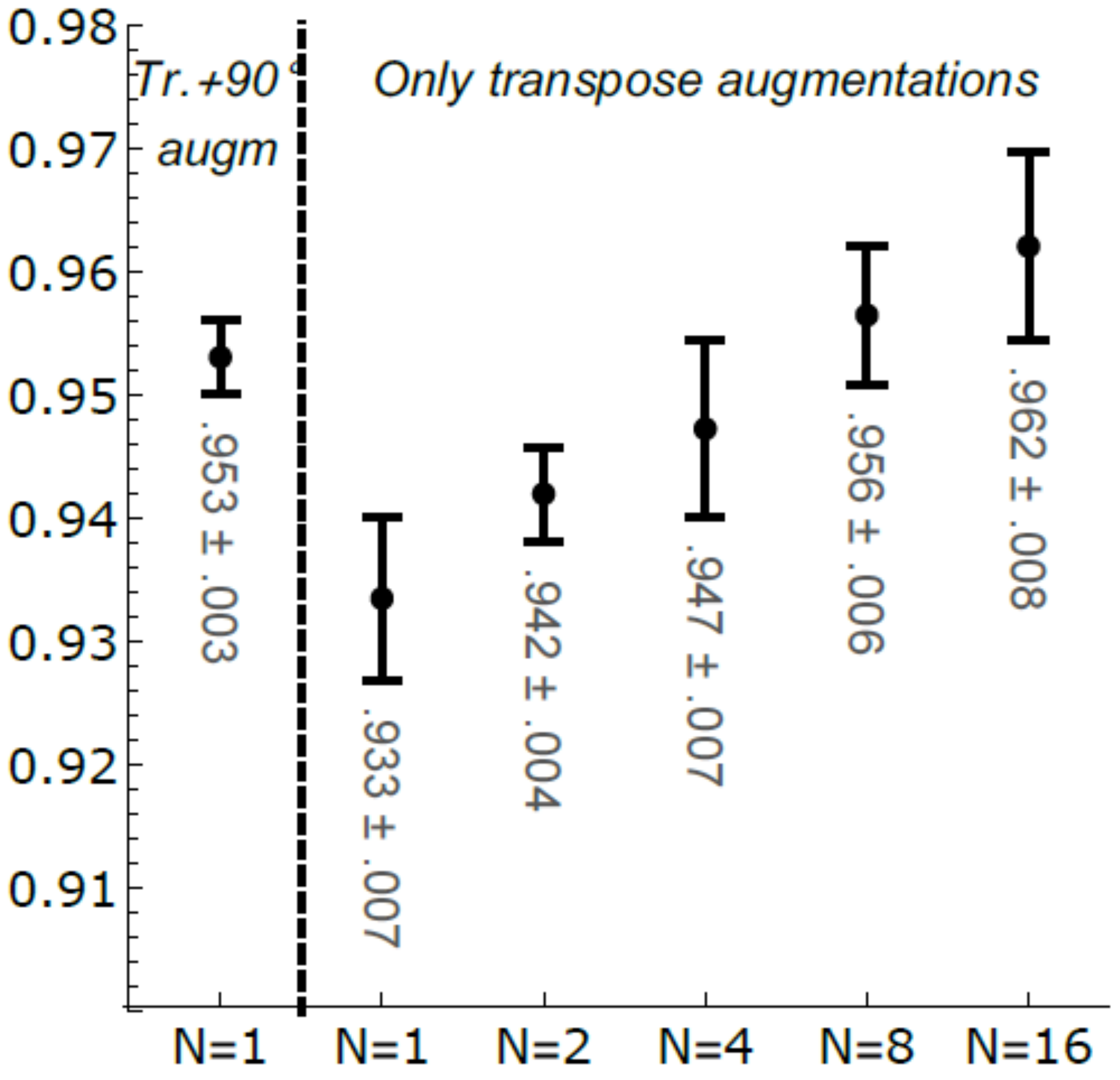
Mitosis detection (F_1–score)



Vessel segmentation (AUC)



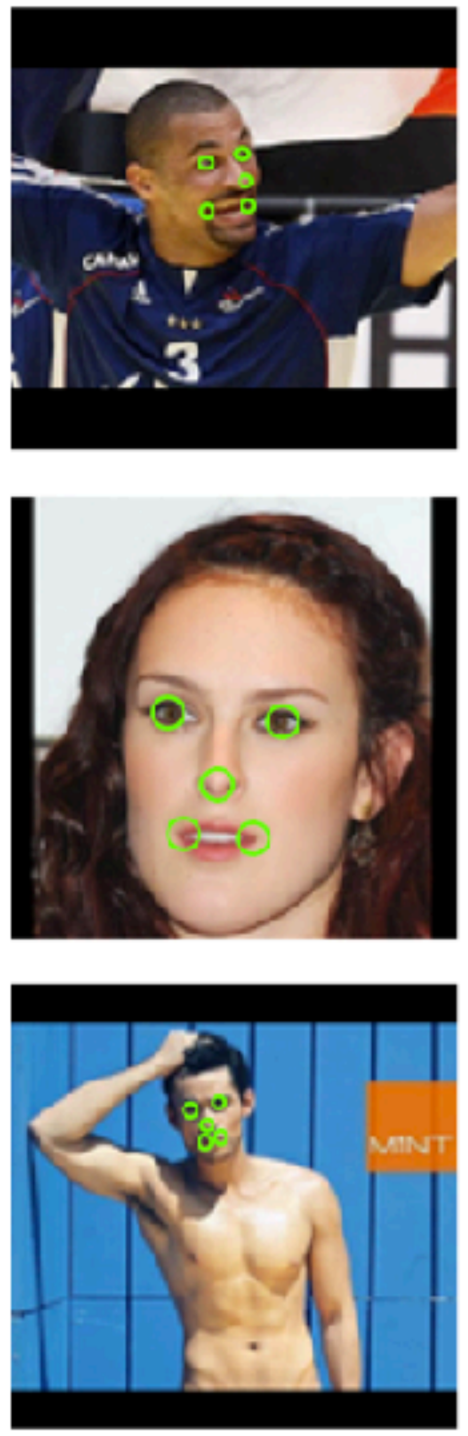
Cell boundary segmentation (Rand)

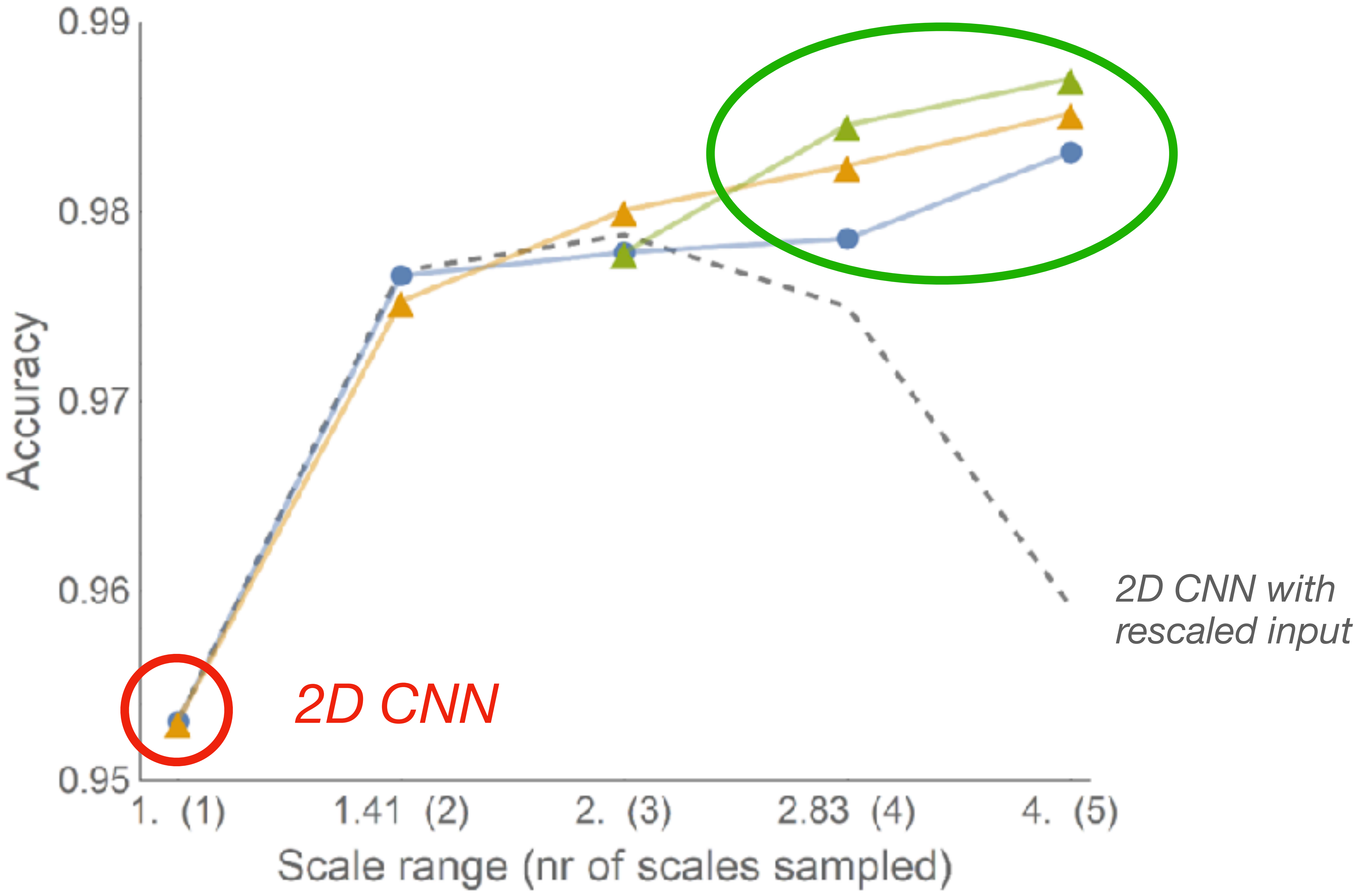


From rotation to scale equivariant CNNs

Bekkers ICLR 2020

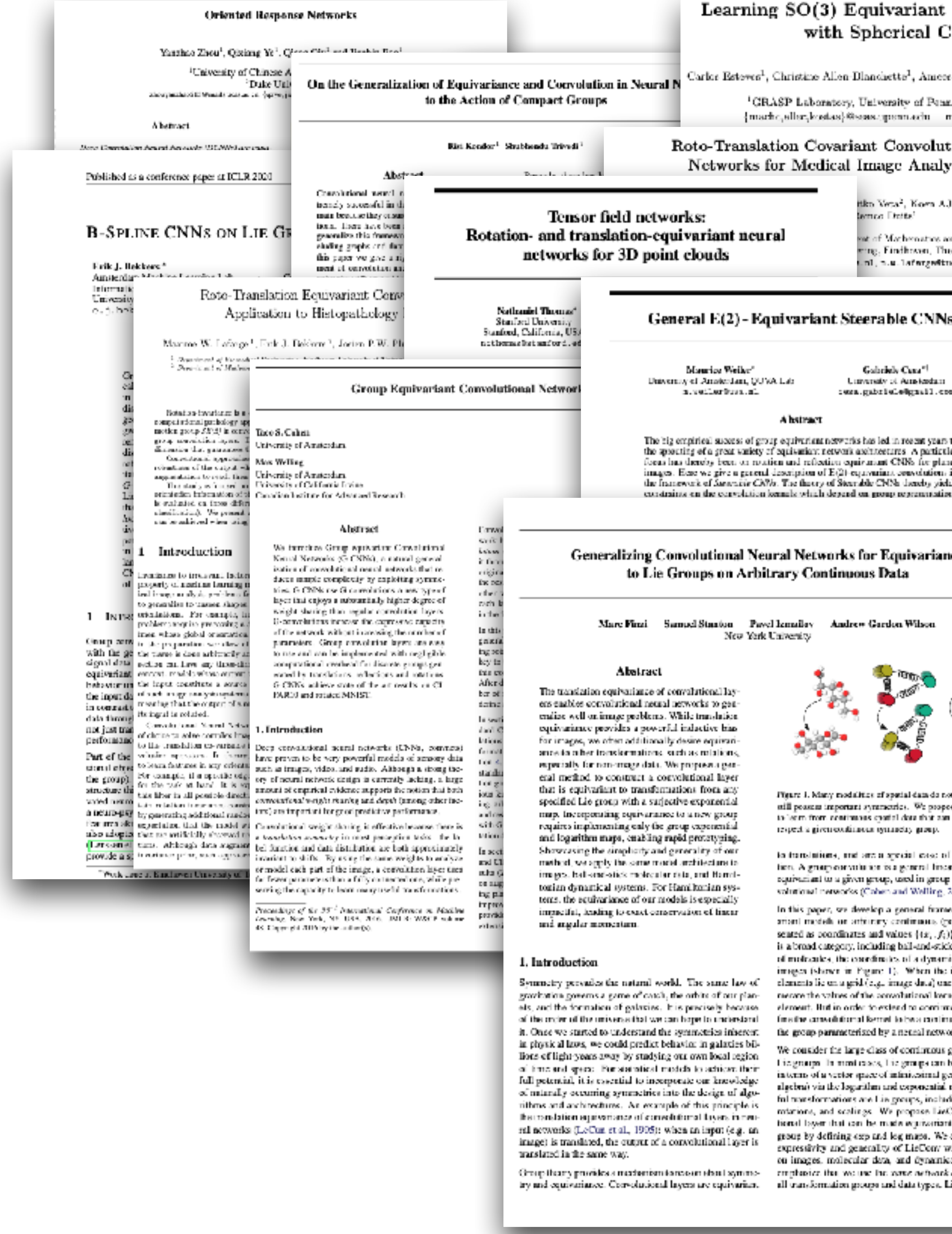
Translation + scale equivariant G-CNNs



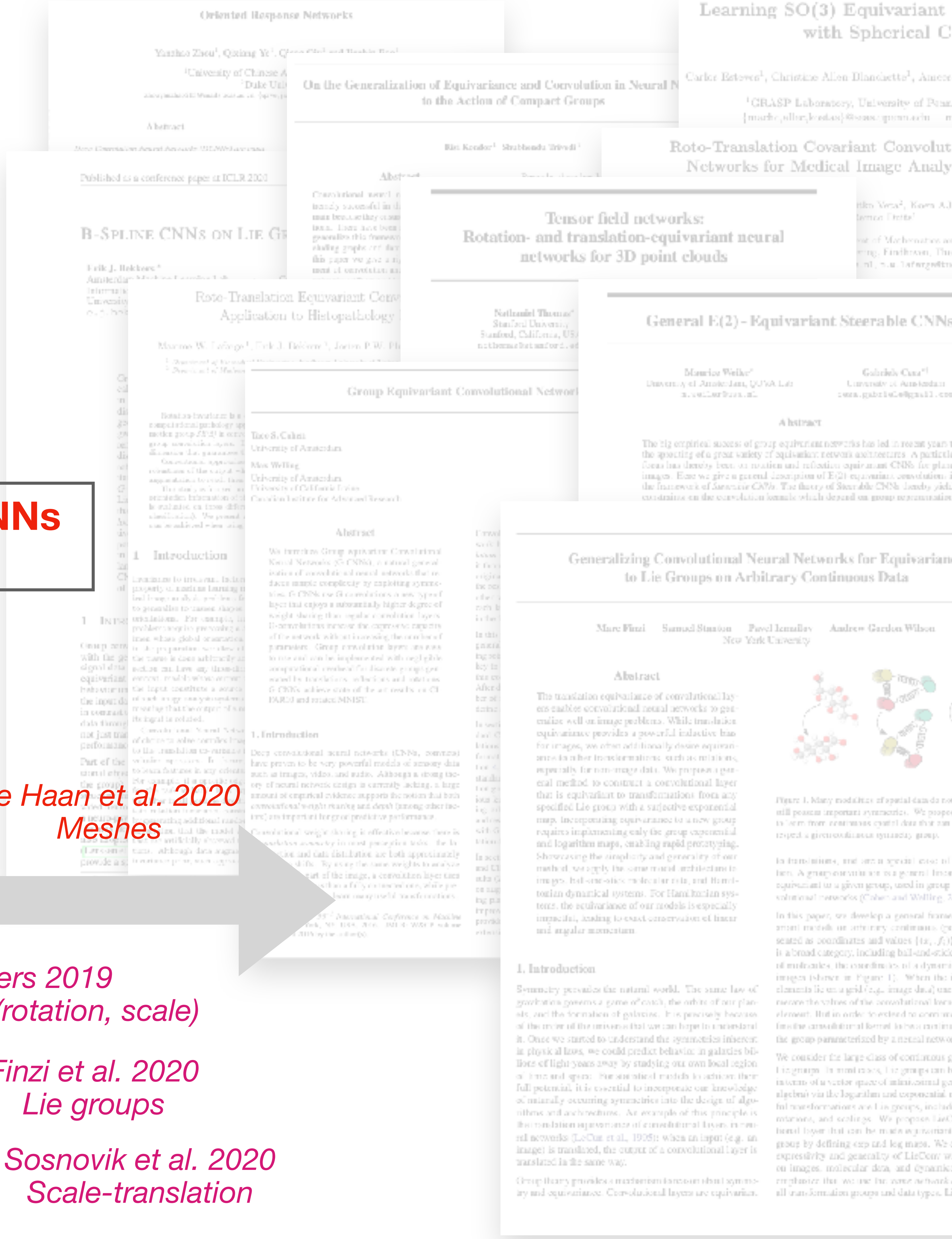


G-CNNs rule!

- The right inductive bias: guaranteed equivariance
  (no loss of information)
- Performance gains that can't be obtained by data-augmentation alone
  (both local and global equivariance/invariance)
- Increased sample efficiency
  (increased weight sharing, no geometric augmentation necessary)



A brief history of G-CNNs



Discrete G-CNNs
(square/cube symmetries)

Gauge-equivariant CNNs
(manifolds)

Cohen-Welling 2016
p4m

Winkels-Cohen 2018
3D discrete roto-transl.

Hoogeboom et al. 2018
SE(2,6)

Dieleman et al. 2016
p4m

Worrall-Brostow 2018
3D discrete roto-transl.

Cohen et al. 2019
Icosahedron

de Haan et al. 2020
Meshes

NNs → CNNs → G-CNNs

LeCun et al. 1990

Mallat et al. 2013, 2015
SE(2)

Bekkers et al. 2014-2018
SE(2)

Weiler et al. 2017
SE(2)

Zhou et al. 2017
SE(2)

Worrall et al. 2017
SE(2)

Esteves 2017
SO(3)

Kondor 2018
SE(3)

Cohen et al. 2018
SE(3)

Kondor-Trivedi 2018
SE(3)

Weiler et al. 2018
SE(3)

Thomas et al. 2018
SE(3)

Weiler-Cesa 2019
SE(2)

Bekkers 2019
Lie groups (rotation, scale)

Finzi et al. 2020
Lie groups

Sosnovik et al. 2020
Scale-translation

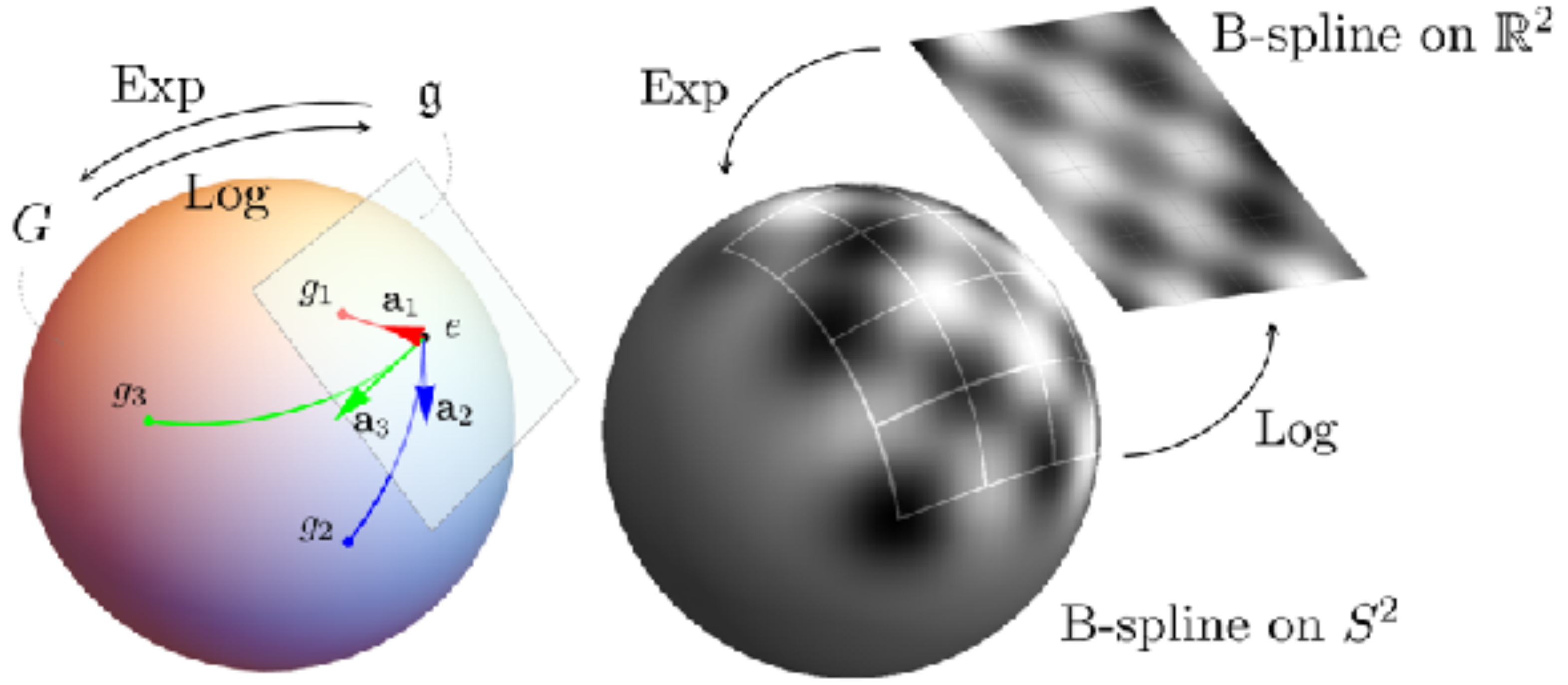
Continuous rotation G-CNNs
(regular discretizations)

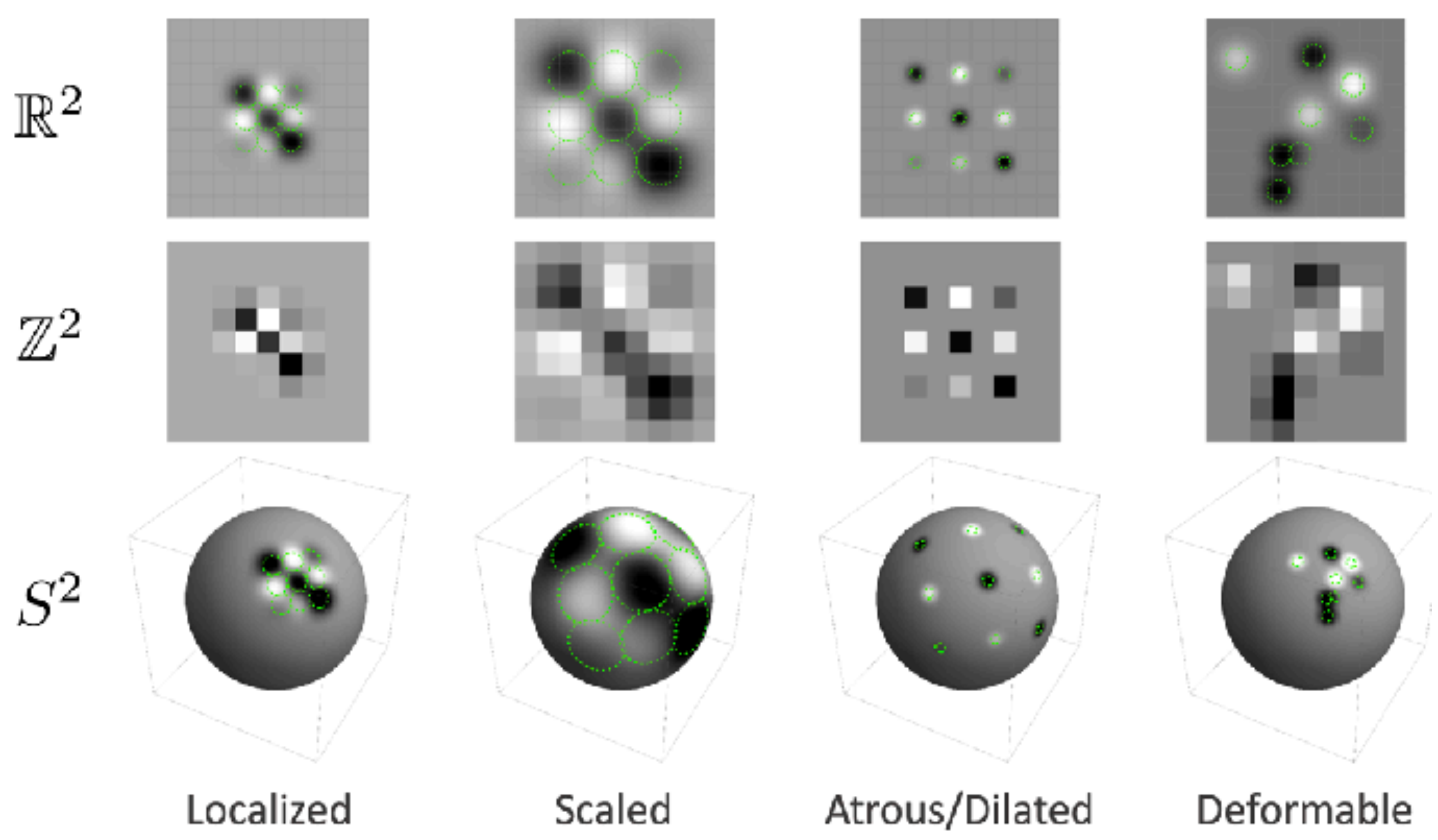
Continuous rotation G-CNNs
(steerable)

Continuous G-CNNs
(Lie groups)

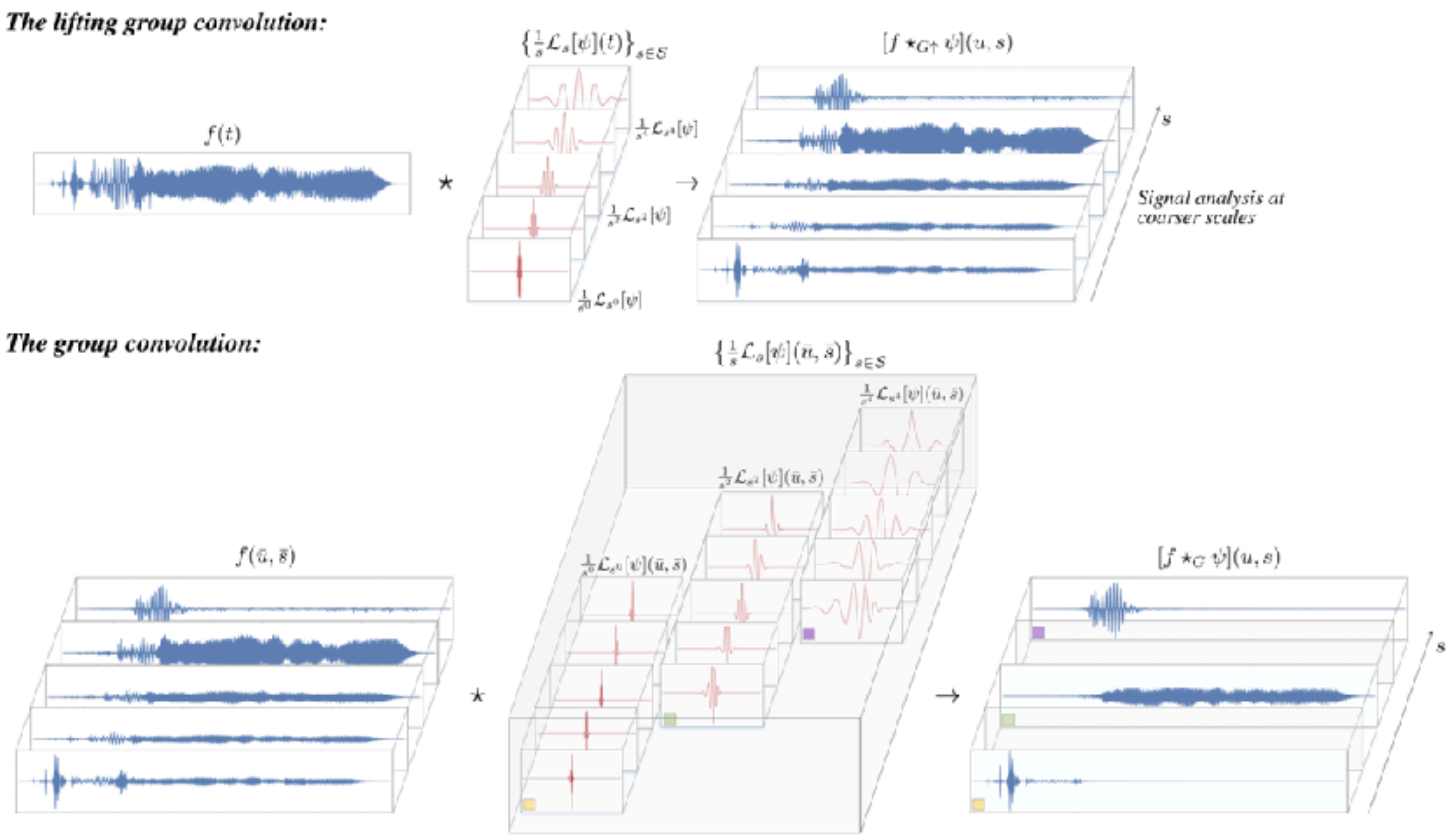
A brief history of G-CNNs

B-Spline CNNs on Lie Groups - ICLR 2020





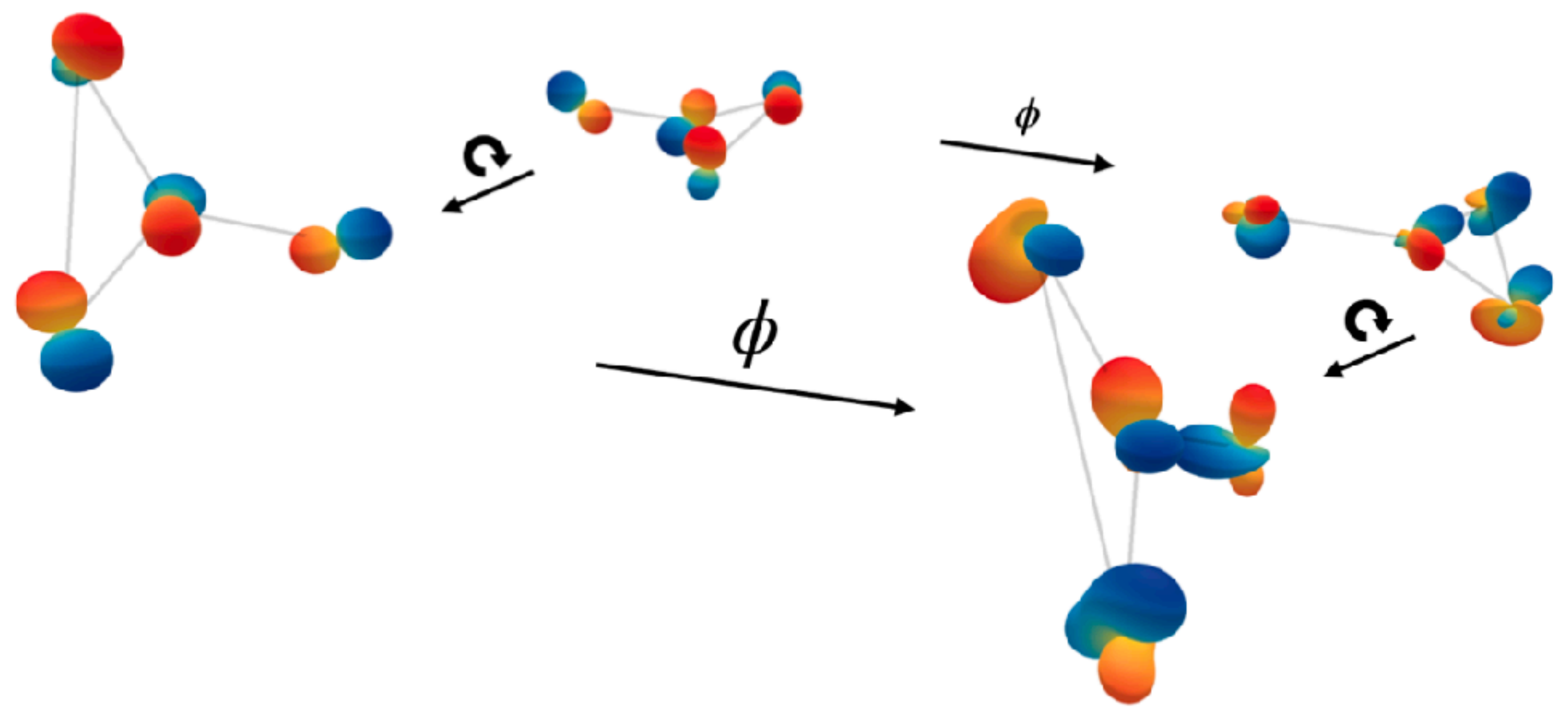
Romero, Bekkers, Tomczak, Hoogeboom
Wavelet Networks: Scale Equivariant Learning From Raw Waveforms - arXiv:2006.05259



A brief history of G-CNNs

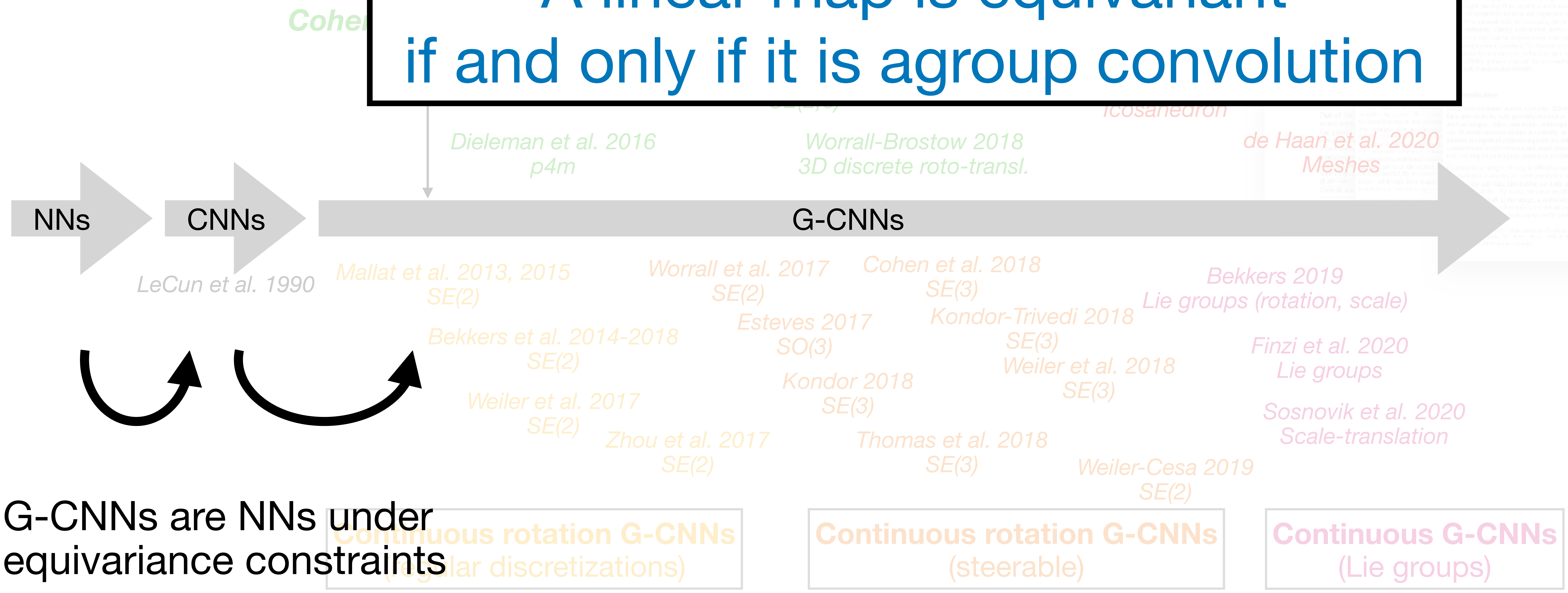
NNs

Brandstetter, Hesselink, van der Pol, Bekkers, Welling **Steerable Equivariant Message Passing on Molecular Graphs** - in review



A brief history of G-CNNs

Part II:
A linear map is equivariant
if and only if it is agroup convolution



G-CNNs are NNs under
equivariance constraints

Content

Part I: Introduction to group convolutions

* Motivation
* Introduction to group theory
* Regular group convolutional neural networks
* Applications

Part II: General theory for group equivariant deep learning

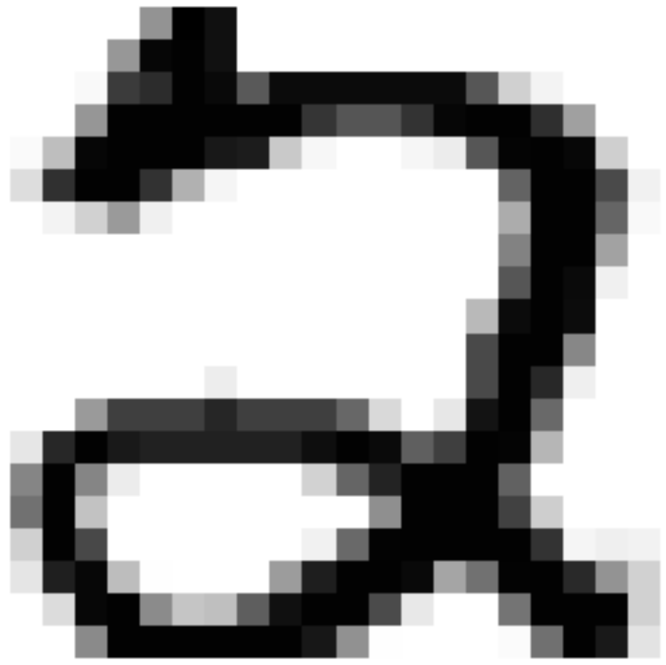
* Group convolutions are all you need!
* Deeper into group theory: representation theory, homogeneous spaces
* Characterization of types of group equivariant layers

Part III: Steerable group convolutions

* Deep dive into group theory: irreducible representations, steerable operators and vector spaces
* Examples of steerable group convolutions: Spherical data and Volumetric data/3D point clouds

Classical artificial neural networks

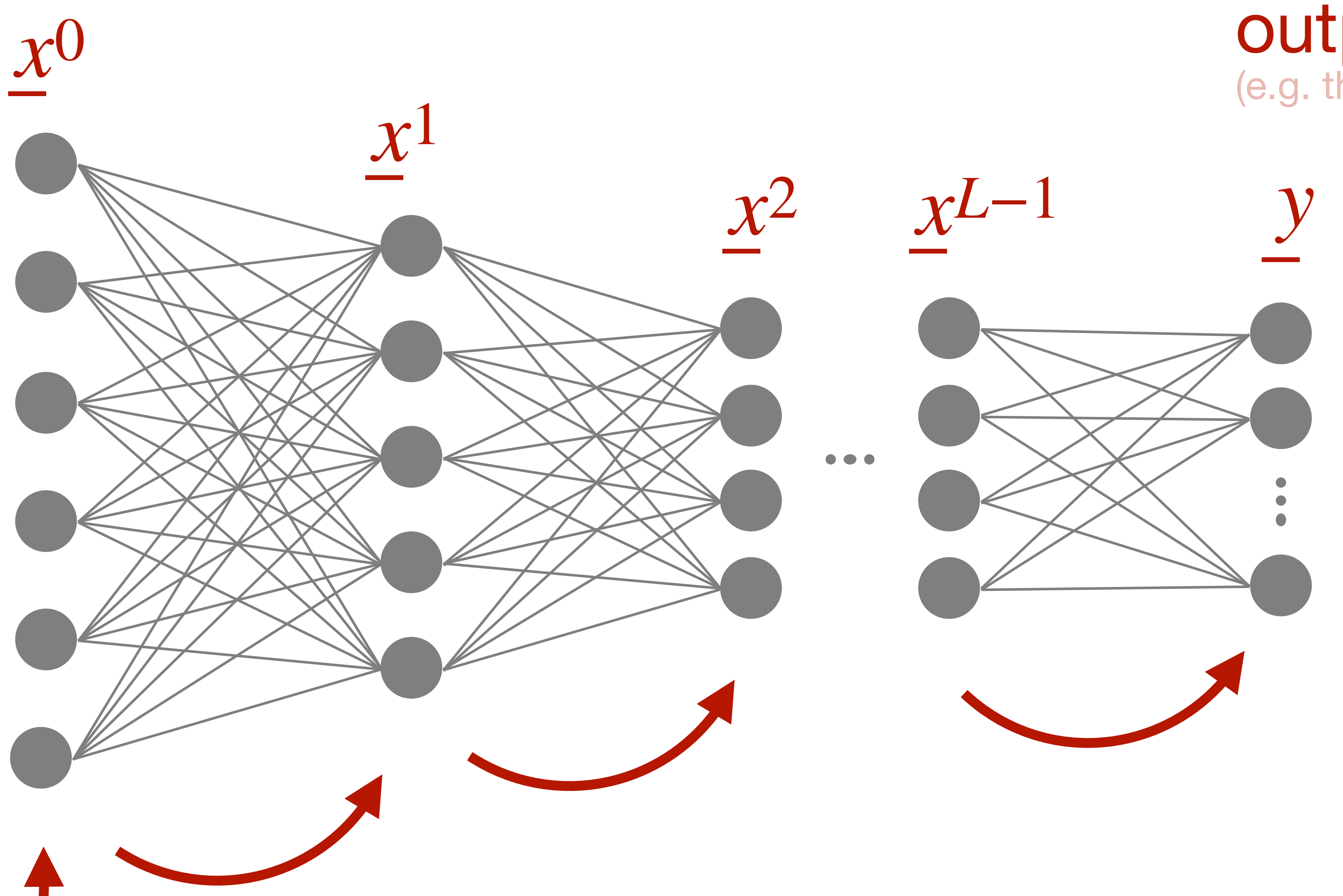
What's my input? $\underline{x}^0 \in \mathcal{X} = ?$



input vector

output probability vector

(e.g. through soft max)



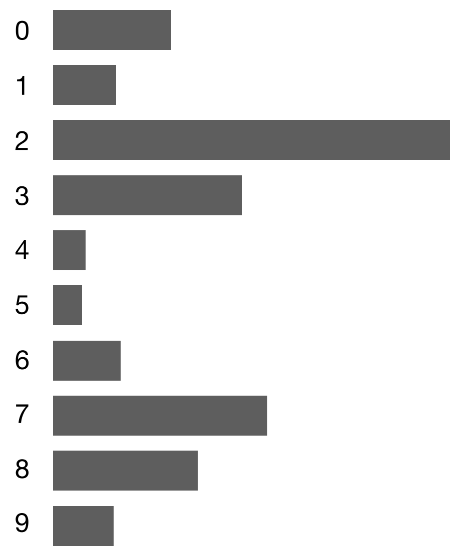


Image analyst: $\underline{x}^0 \in \mathcal{X} = \mathbb{L}_2(\mathbb{R}^2)$

Naive deep learner: $\underline{x}^0 \in \mathcal{X} = \mathbb{R}^{784}$

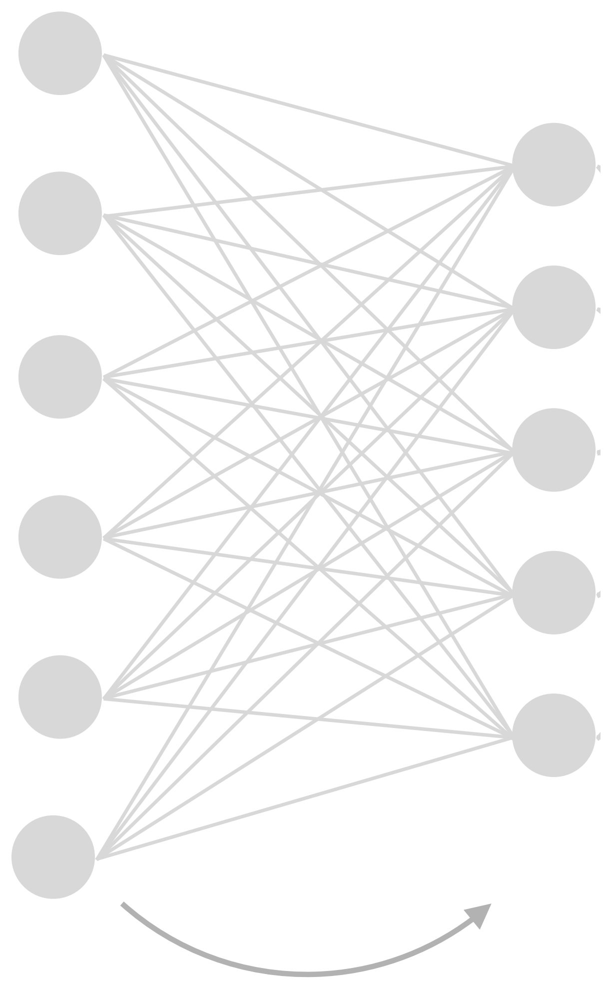
Iteratively transform the vector via

$$\underline{x}^l = \varphi(K_{\mathbf{w}_l}\underline{x}^{l-1} + \underline{b}^l)$$

Linear map: matrix-vector multiplication with $K_{\mathbf{w}_l} \in \mathbb{R}^{N^l \times N^{l-1}}$

Classical artificial NNs in the continuous world

Working with vectors $\underline{x} \in \mathcal{X} = \mathbb{R}^{N^x}$



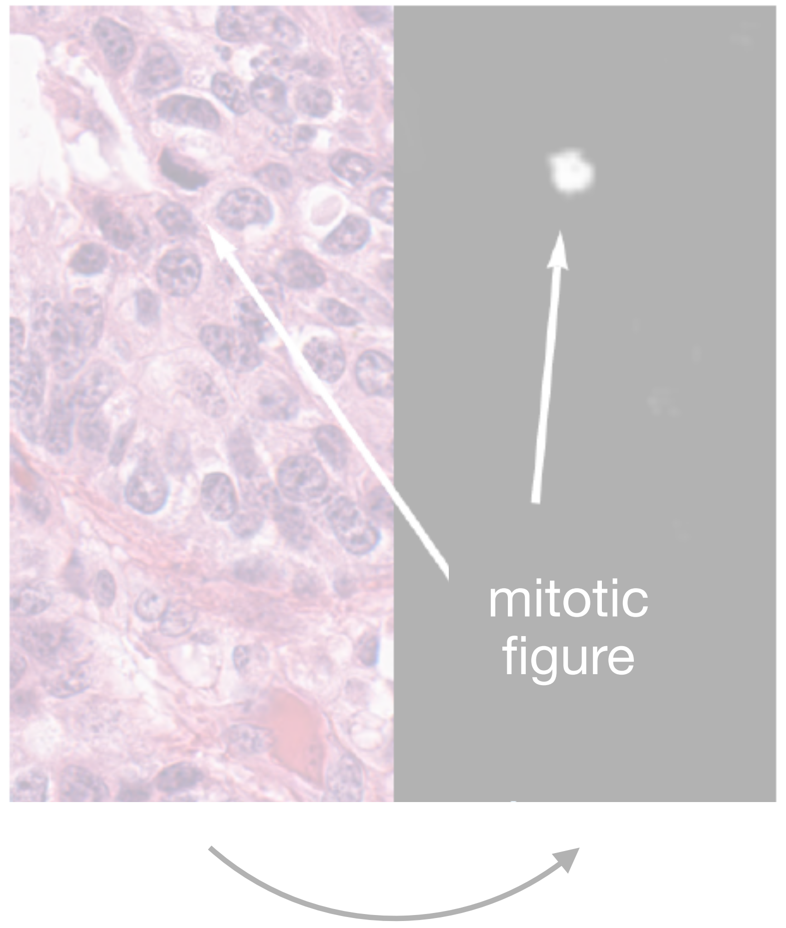
Iteratively transform the **vector** in $\mathbb{R}^{N^x}$ via

We want K to be equivariant!

Linear map: matrix-vector multiplication with $K \in \mathbb{R}^{N^y \times N^x}$

$$y_j = \sum_i K_{i,j}\, x_i$$

Working with feature maps $f \in \mathcal{X} = \mathbb{L}_2(X)$



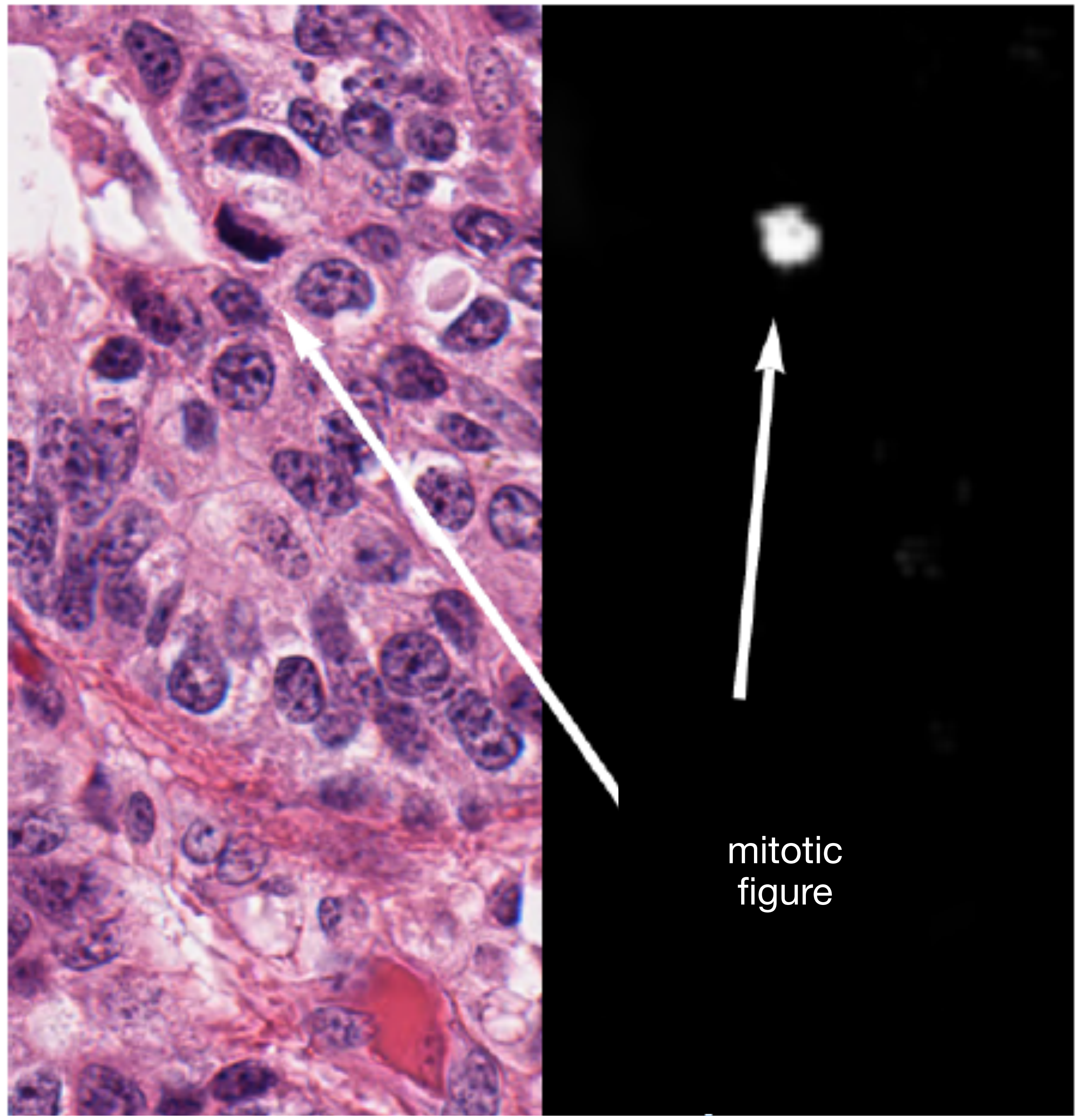
Iteratively transform the **feature map** in $\mathbb{L}_2(X)$

$$f^{out} = \varphi(K f^{in} + b^l)$$

Linear map: kernel operator with kernel in $\mathbb{L}_1(Y, X)$

$$(Kf)(y) = \int_X k(y,x) f(x) \mathrm{d}x$$

Neural Networks for Signal Data



$$\mathcal{K} : \mathbb{L}_2(X)^{N_l} \to \mathbb{L}_2(Y)^{N_{l+1}}$$

Let's build neural networks for signal data via the layers of the form:

$$\underline{f}^{l+1} = \sigma(\mathcal{K}\,\underline{f}^{l} + \mathbf{b}^{l})$$

The linear map has to be an integral transform with a two-argument kernel (Dunford-Pettis theorem)

$$(\mathcal{K}f)(y) = \int_X \mathbf{k}(y, x)\underline{f}(x)\mathrm{d}x$$

Theorem 3.2:

Let $\mathscr{K} : \mathbb{L}_2(X) \rightarrow \mathbb{L}_2(Y)$ map between signals on homogeneous spaces of G.

Let homogeneous space $Y \equiv G/H$ such that $H = \mathrm{Stab}_G(y_0)$ for some chosen origin $y_0 \in Y$ and let $g_y \in G$ such that $\forall_{y \in Y} : y = g_y y_0$.

Then $\mathscr{K}$ is equivariant to group G if and only if:

1. It is a group convolution: $[\mathscr{K} f](y) = \int_X k(g_y^{-1} x) f(x) \mathrm{d}x$
2. The kernel satisfies a symmetry constraint: $\forall_{h \in H} : \quad k(hx) = k(x)$

Theorem 3.2:

Let $\mathscr{K} : \mathbb{L}_2(X) \to \mathbb{L}_2(Y)$ map between signals on homogeneous spaces of G.

Let homogeneous space $Y \equiv G/H$ such that $H = \mathrm{Stab}_G(y_0)$ for some chosen origin $y_0 \in Y$ and let $g_y \in G$ such that $\forall_{y \in Y} : y = g_y y_0$.

Then $\mathscr{K}$ is equivariant to group G if and only if:

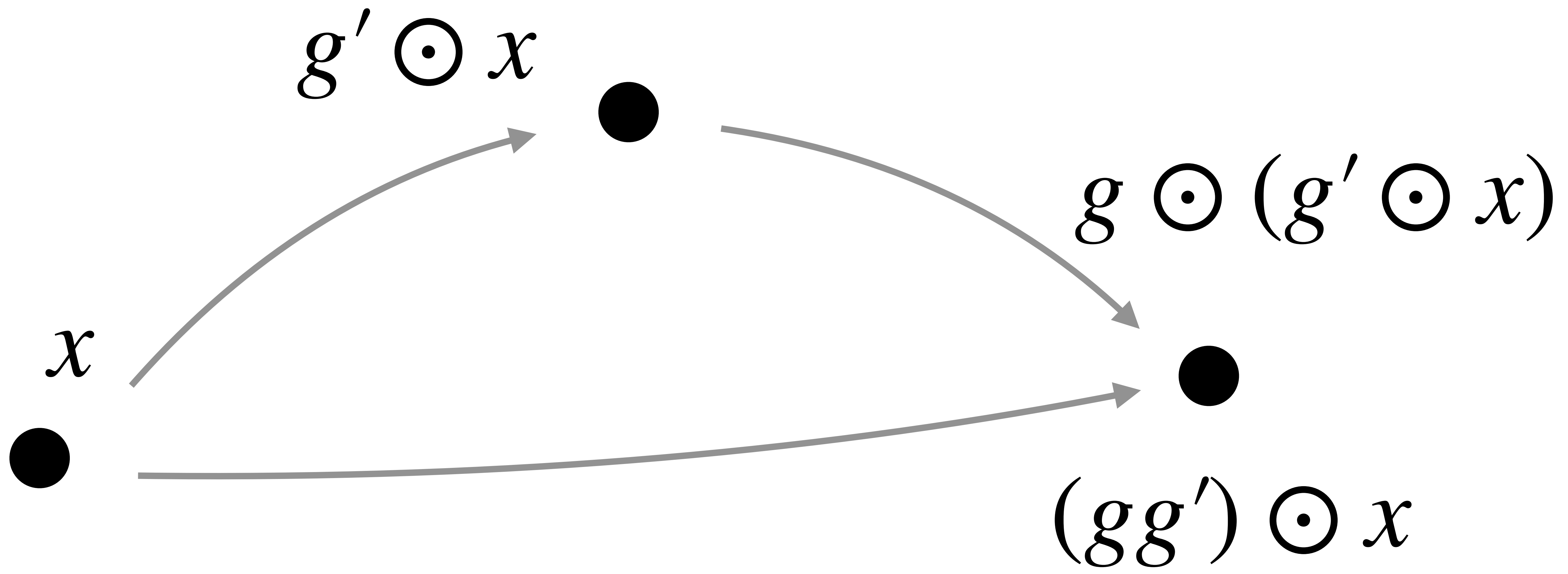
1. It is a group convolution: $[\mathscr{K}f](y) = \int_X k(g_y^{-1}x)f(x)\mathrm{d}x$

2. The kernel satisfies a symmetry constraint: $\forall_{h \in H} : \quad k(hx) = k(x)$

Group theory: Homogeneous spaces

Group action: An operator $\odot : G \times X \rightarrow X$ such that

$$\forall_{g,g' \in G, x \in X} : \; g \odot (g' \odot x) = (gg') \odot x$$

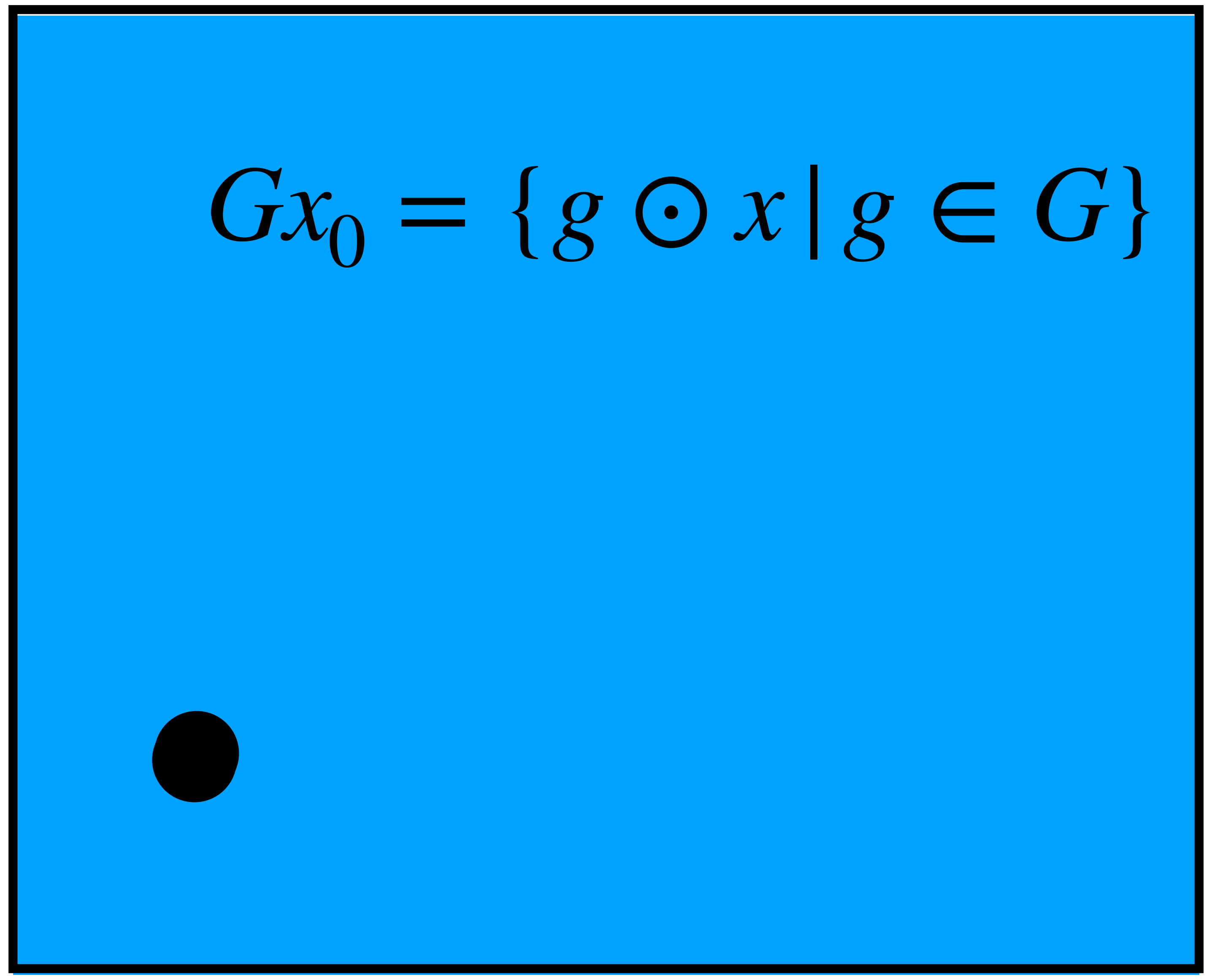


Group theory: Homogeneous spaces

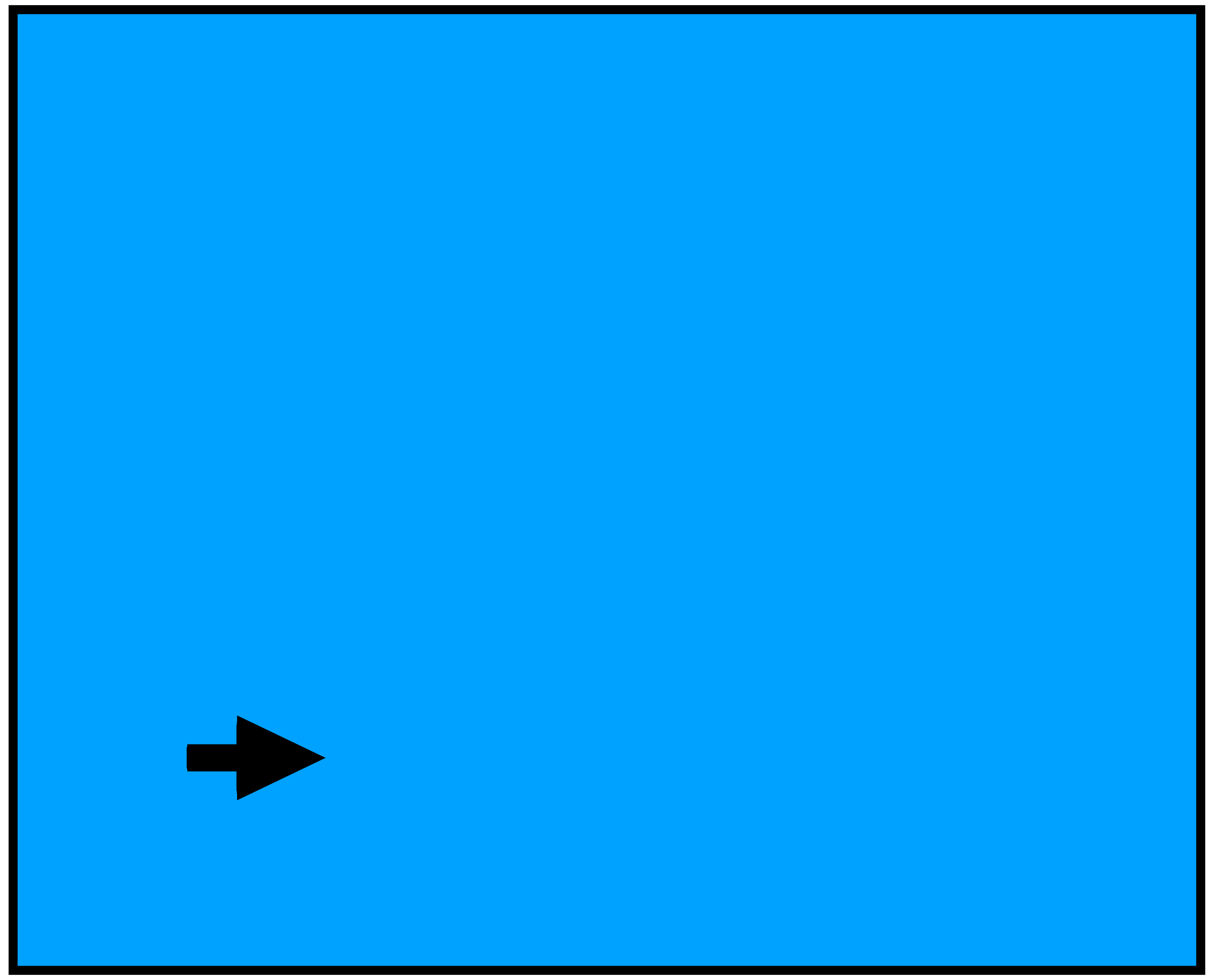
Transitive action: An action $\odot : G \times X \to X$ such that

$$\forall_{x_0, x \in X} \exists_{g \in G} : x = g \odot x_0$$

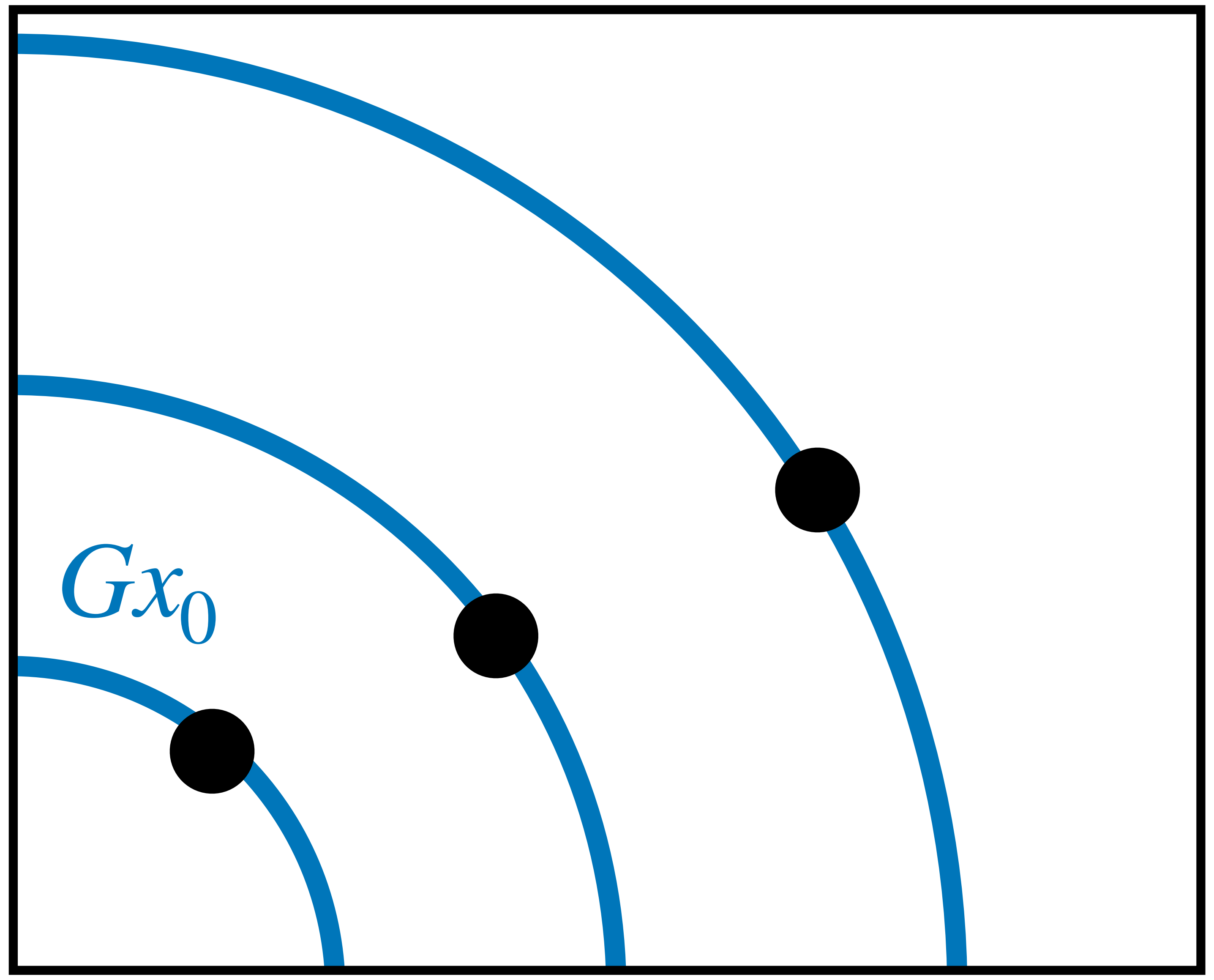
$(\mathbb{R}^2, +)$ acts transitively on $\mathbb{R}^2$



SE(2) acts transitively on $\mathbb{R}^2$



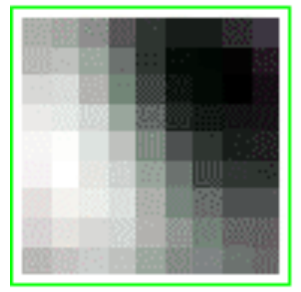
SO(2) does not …

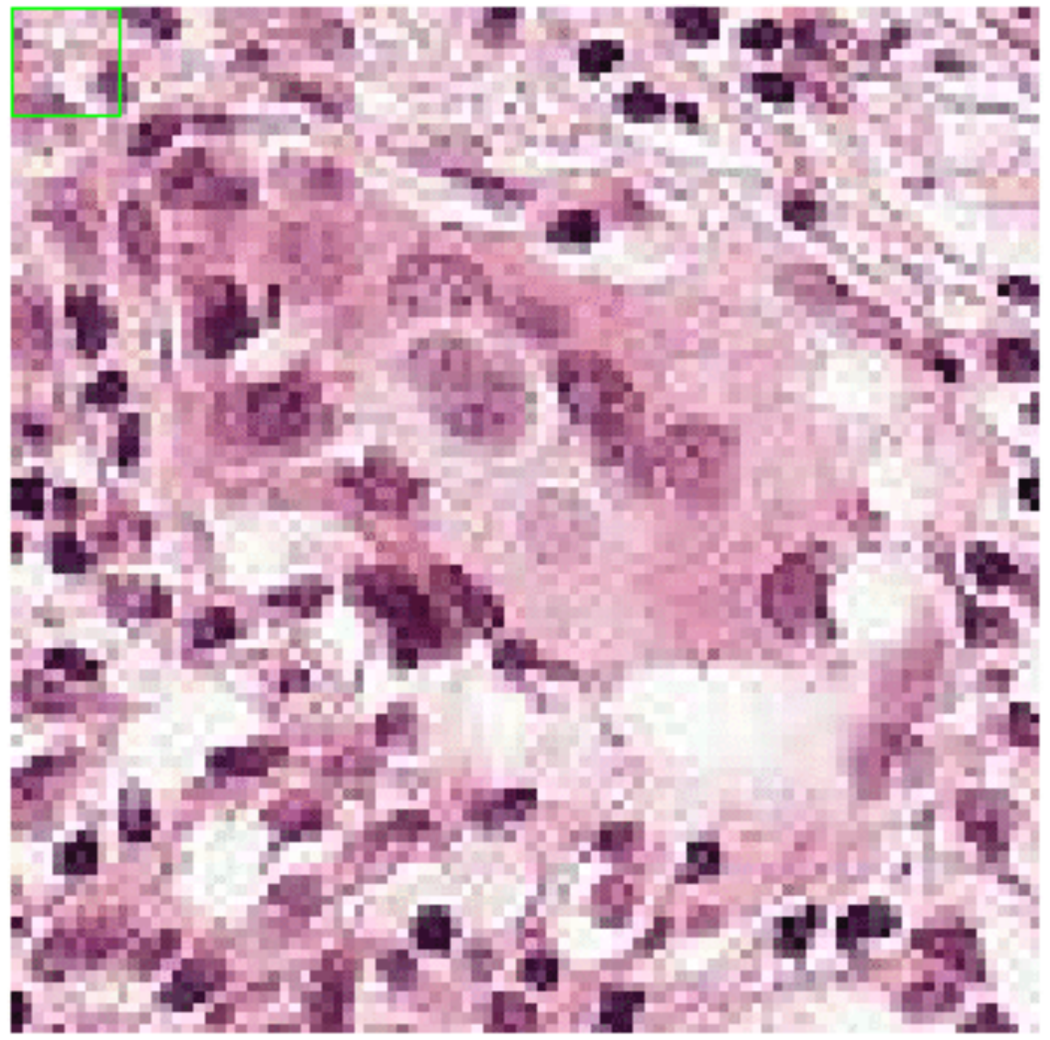


Group theory: Homogeneous spaces

Homogeneous space: A space on X on which G acts transitively.

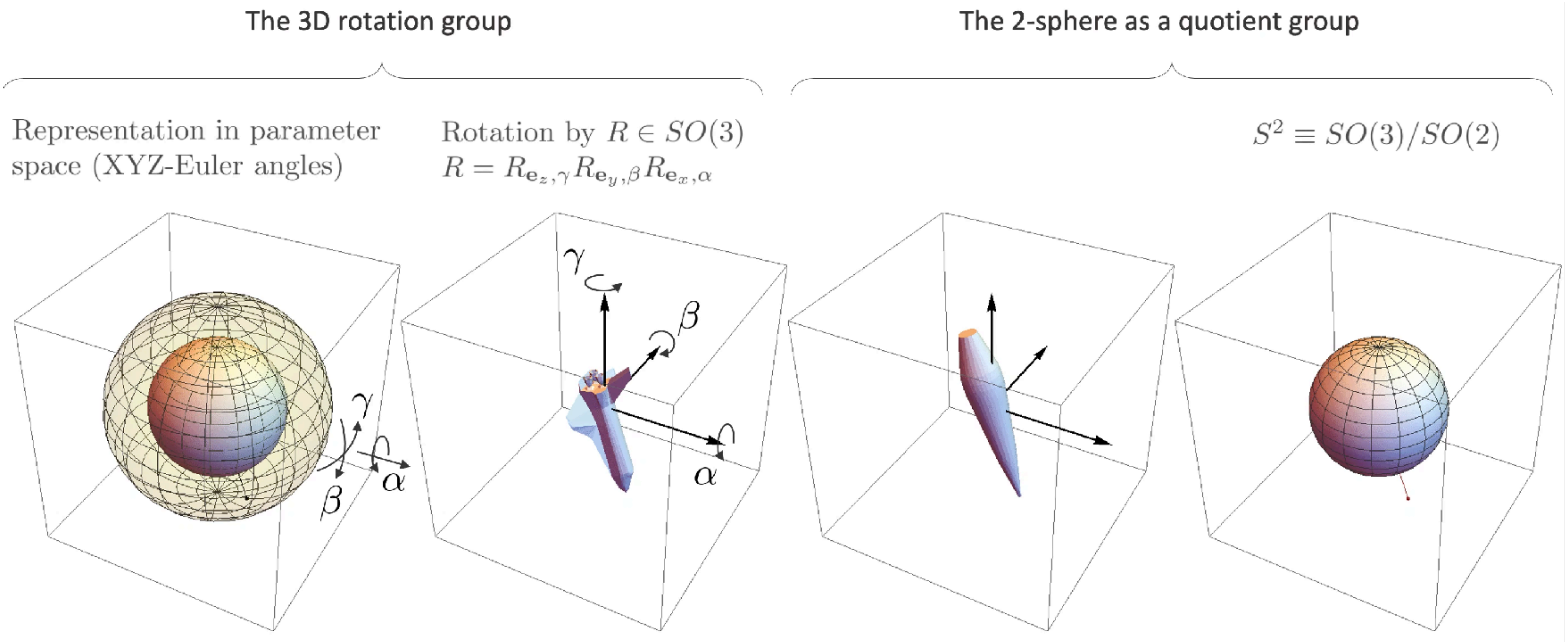
This is important as then we can guarantee that every part of the signal can be “seen” (probed by the convolution kernel)





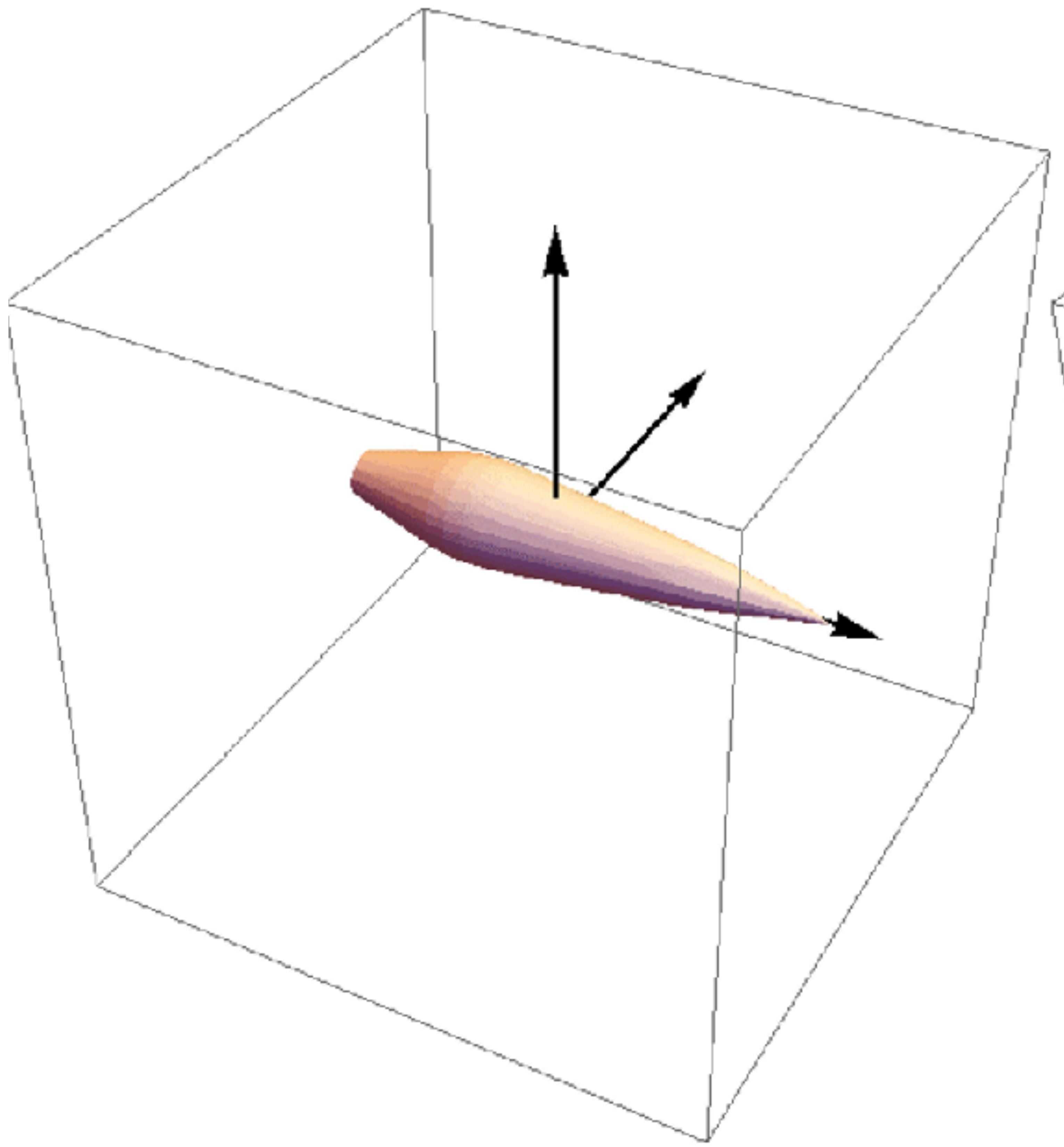
Group theory: Homogeneous spaces

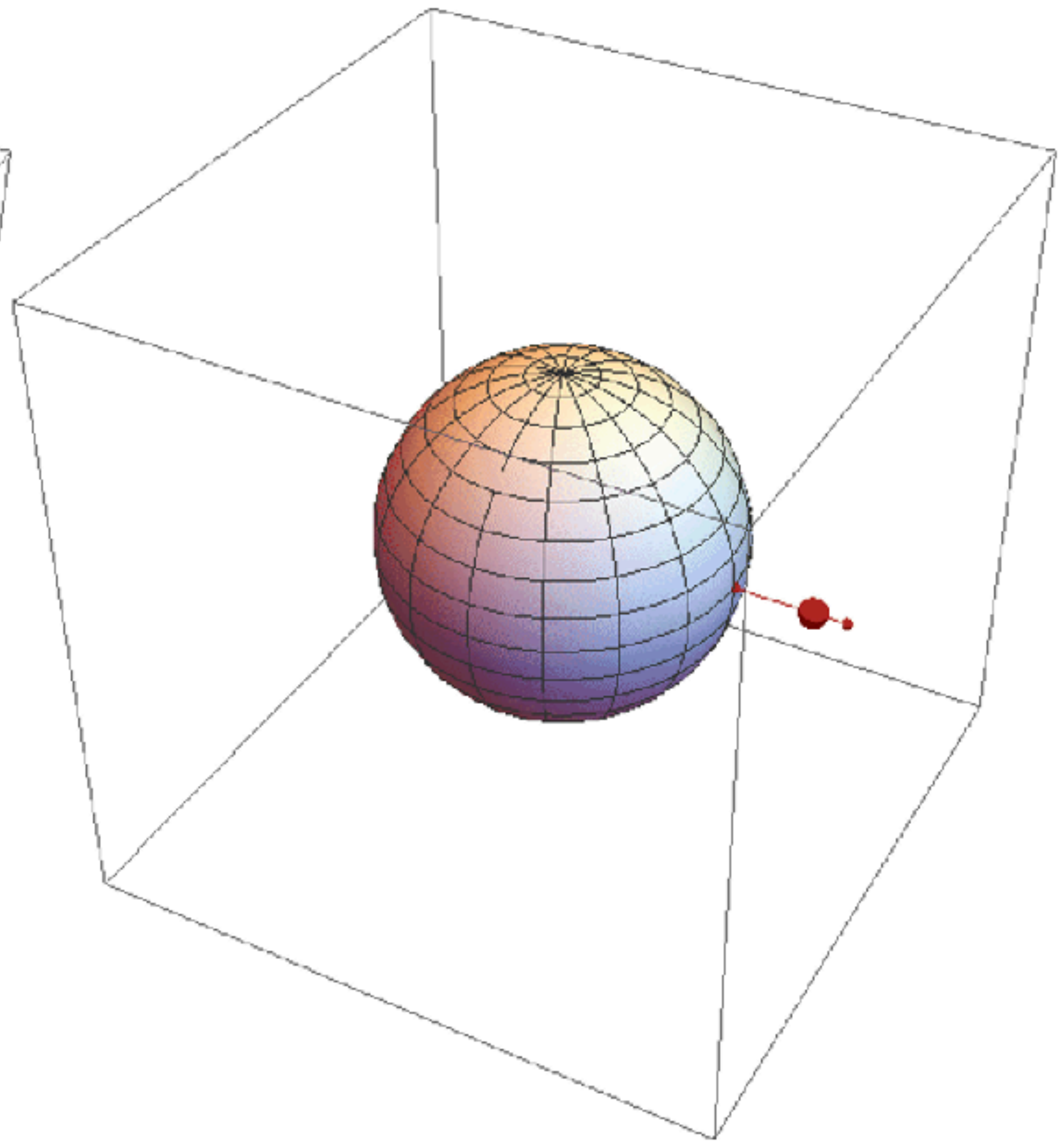
The sphere S^2 is a homogeneous space of 3D rotations $SO(3)$

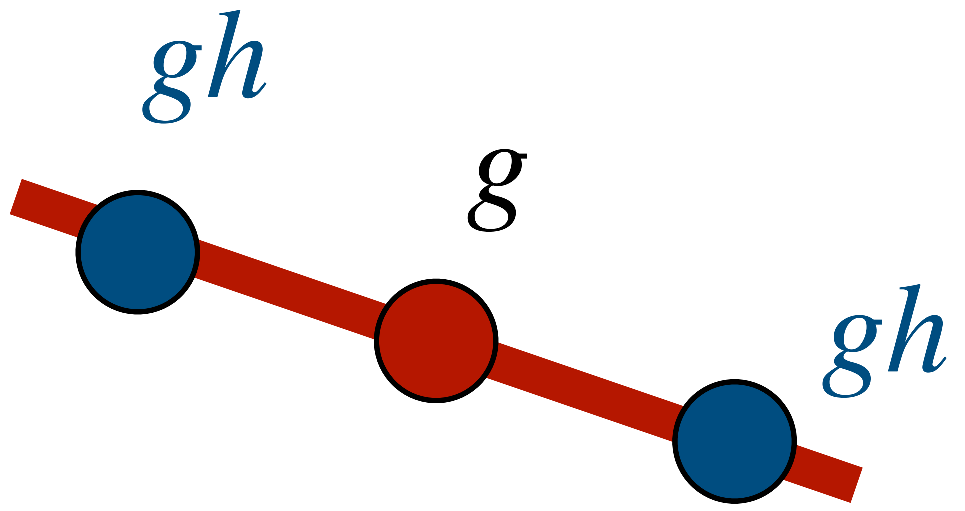


Group theory: Quotient spaces

Quotient space G/H: The space of unique cosets $gH = \{gh \,|\, h \in H\}$. Elements of the space G/H are cosets.



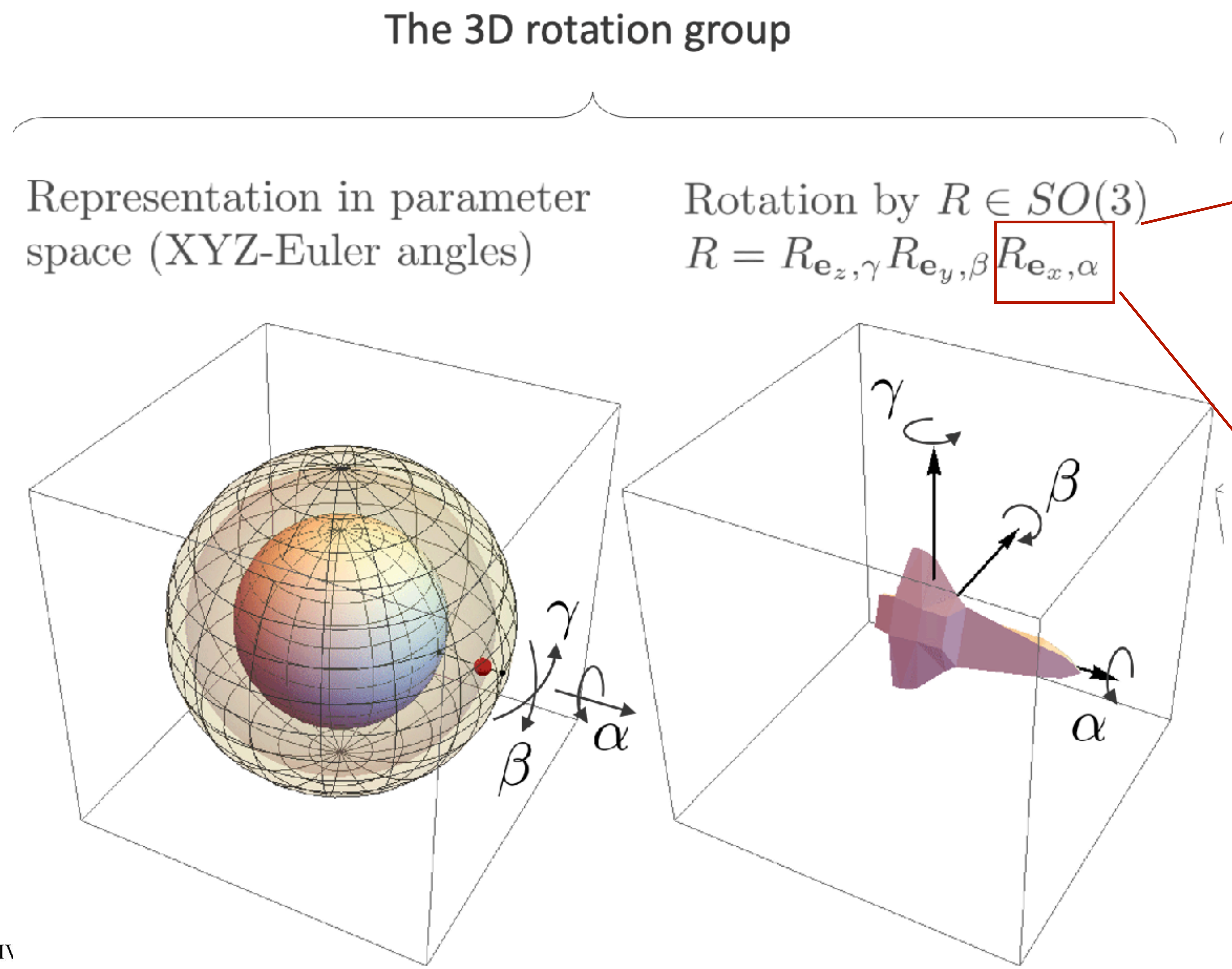




$$gH = \{gh \,|\, h \in H\}$$

Group theory: Stabilizer

Stabilizer: $\mathrm{Stab}_G(x_0)$ is a subset of G that leaves x_0 unchanged. I.e.,
$\mathrm{Stab}_G(x_0) = \{g \mid gx_0 = x_0\}$



So the sphere is a quotient space

$$S^2 \equiv SO(3)/H$$

with

$$H = \mathrm{Stab}_G(\mathbf{e}_x)$$

Group theory: Homogeneous space $\equiv$ Quotient space

Lemma 2.1: Any quotient space is a homogeneous space

Lemma 2.2: Any homogeneous space is a quotient space

Group theory: Quotient spaces

Lecture notes
Section 2.3

Example 2.7 (Quotient space $\mathbb{R}^d = SE(d)/SO(d)$). Let $H = (\{\mathbf{0}\} \times SO(d)$ the subgroup of rotations in $SE(d)$, with $\mathbf{0}$ the identity element of the translationg roup $(\mathbb{R}^d, +)$. The the cosets gH are given by

$$\begin{aligned} gH &= \{g \cdot (\mathbf{0}, \tilde{\mathbf{R}}) \mid \tilde{\mathbf{R}} \in SO(d)\} \\ &= \{(\mathbf{Re} + \mathbf{x}, \mathbf{R}\tilde{\mathbf{R}}) |\ h \in SO(d)\} \\ &= \{(\mathbf{x}, \mathbf{R}\tilde{\mathbf{R}}) |\ \tilde{\mathbf{R}} \in SO(d)\} \\ &= \{(\mathbf{x}, \tilde{\mathbf{R}}) |\ \tilde{\mathbf{R}} \in SO(d)\}\,, \end{aligned}$$

with $g = (\mathbf{x}, \mathbf{R})$. So, the cosets are given by all possible rotations for a fixed translation vector $\mathbf{x}$, the vector $\mathbf{x}$ thus indexes these sets. We can therefore make the identification

$$\mathbb{R}^d \equiv SE(d)/SO(d)\,.$$

We already saw in Exercise 2.1 that $\mathbb{R}^d$ is a homogeneous space of $SE(d)$, this is a consequence of Lemma 2.1.

Theorem 3.2:

Let $\mathscr{K} : \mathbb{L}_2(X) \rightarrow \mathbb{L}_2(Y)$ map between signals on homogeneous spaces of G.

Let homogeneous space $Y \equiv G/H$ such that $H = \mathrm{Stab}_G(y_0)$ for some chosen origin $y_0 \in Y$ and let $g_y \in G$ such that $\forall_{y \in Y} : y = g_y y_0$.

Then $\mathscr{K}$ is equivariant to group G if and only if:

1. It is a group convolution: $[\mathscr{K} f](y) = \int_X k(g_y^{-1} x) f(x) \mathrm{d}x$

2. The kernel satisfies a symmetry constraint: $\forall_{h \in H} : \quad k(hx) = k(x)$

Theorem:

Let $\mathscr{K} : \mathbb{L}_2(X) \to \mathbb{L}_2(Y)$ map between signals on homogeneous spaces of G.

Let homogeneous space $Y \equiv G/H$ such that $H = \mathrm{Stab}_G(y_0)$ for some chosen origin $y_0 \in Y$ and let $g_y \in G$ such that $\forall_{y \in Y} : y = g_y y_0$.

Then $\mathscr{K}$ is equivariant to group G if and only if:

1. It is a group convolution: $[\mathscr{K}f](y) = \int_X k(g_y^{-1}x)f(x)\mathrm{d}x$

2. The kernel satisfies a symmetry constraint: $\forall_{h \in H} : \quad k(hx) = k(x)$

Types of layers

$(X = Y = G/H)$

Isotropic/Constraint convolutions on spaces of lower dimension than G, $\forall_{h \in H} : k(hx) = k(x)$

$(X = G/H, Y = G)$

Lifting convolution. No constraints on k.

$(X = Y = G)$

Group convolutions. No constraints on k.

$(X = G, Y = G/H)$

Projection layer. Mean pooling over H.

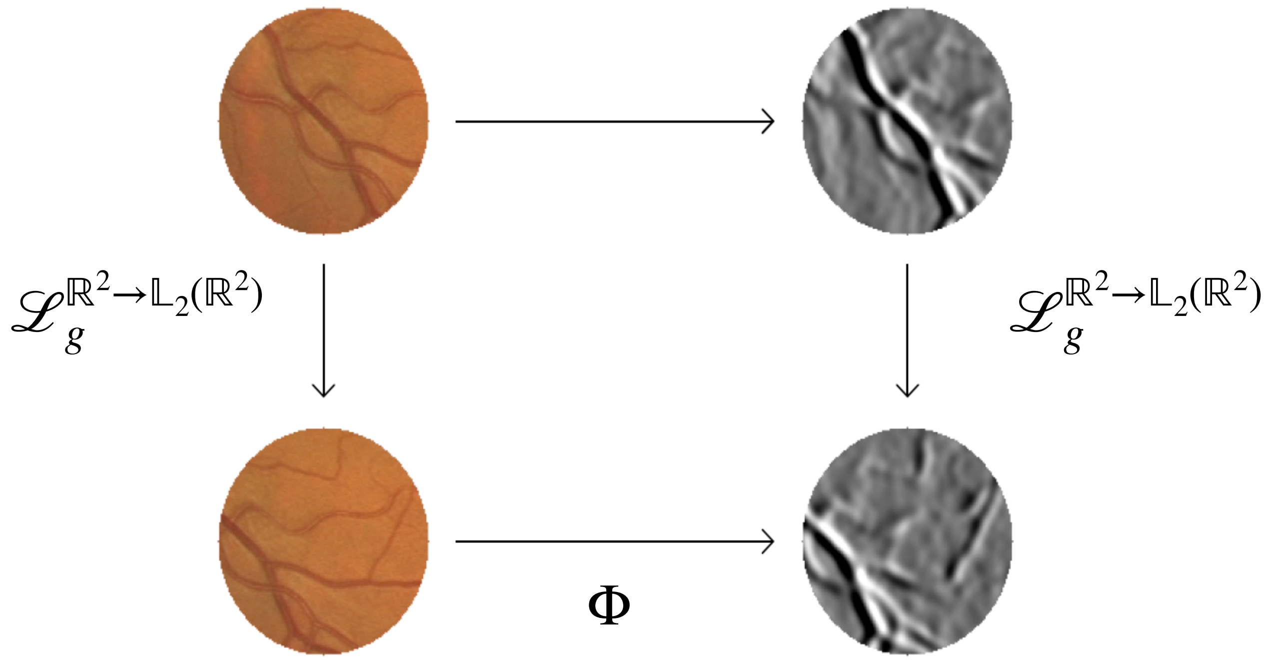
$(X = G, Y = \varnothing)$

Global pooling over G.

The case of $SE(2)$ equivariant layers

for signals on $\mathbb{R}^d \equiv SE(2)/SO(2)$

2D cross-correlation (translation equivariant) - $K : \mathbb{L}_2(\mathbb{R}^2) \to \mathbb{L}_2(\mathbb{R}^2)$



$$(k \star_{\mathbb{R}^2} f)(\mathbf{x}) = (\mathcal{L}_{\mathbf{x}}^{\mathbb{R}^2\to\mathbb{L}_2(\mathbb{R}^2)} k, f)_{\mathbb{L}_2(\mathbb{R}^2)}$$

$$= \int_{\mathbb{R}^2} k(\mathbf{x}' - \mathbf{x}) f(\mathbf{x}') \mathrm{d}\mathbf{x}'$$

SE(2) equivariance iff

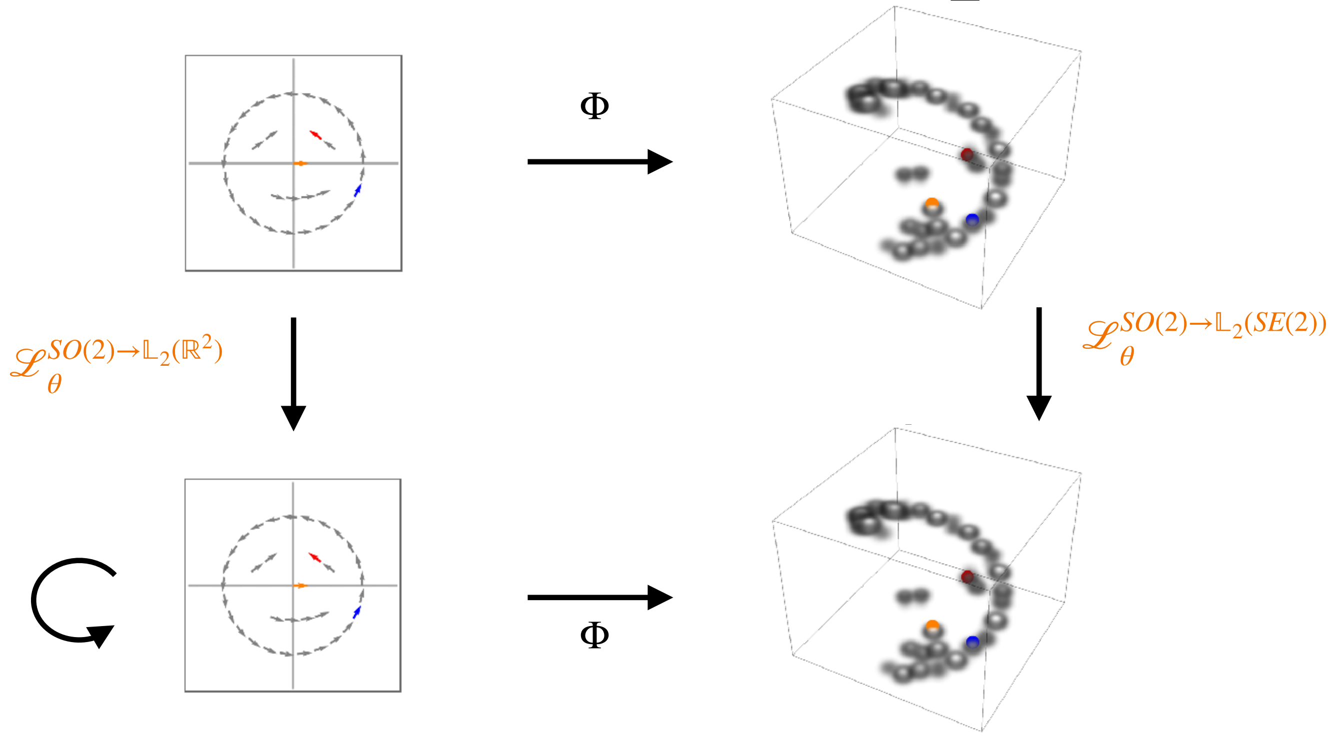
$$(\mathcal{L}_\theta^{SO(2)\to\mathbb{L}_2(\mathbb{R}^2)} k)(\mathbf{x}) = k(\mathbf{x})$$

$$\Leftrightarrow$$

$$k(\mathbf{R}_\theta^{-1}\mathbf{x}) = k(\mathbf{x})$$

since $Y = \mathbb{R}^2 \equiv SE(2)/SO(2)$

SE(2) lifting correlations - $K : \mathbb{L}_2(\mathbb{R}^2) \to \mathbb{L}_2(SE(2))$

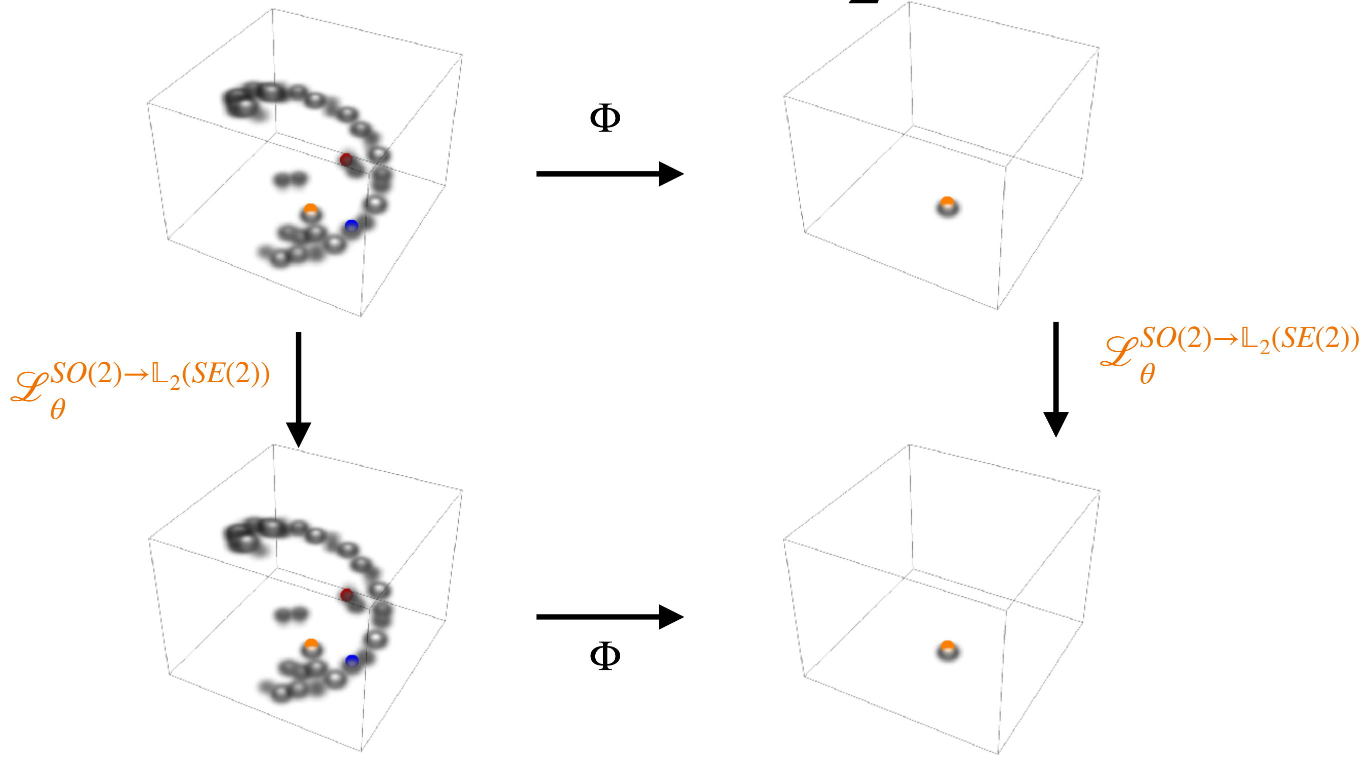


$$(k \tilde{\star} f)(\mathbf{x}) = (\mathcal{L}_g^{SE(2)\to\mathbb{L}_2(\mathbb{R}^2)} k\ , f)_{\mathbb{L}_2(\mathbb{R}^2)}$$

$$= \int_{\mathbb{R}^2} k(\mathbf{R}_\theta^{-1}\mathbf{x}' - \mathbf{x}) f(\mathbf{x}') \mathrm{d}\mathbf{x}'$$

No constraints

SE(2) G-correlations - $K : \mathbb{L}_2(SE(2)) \to \mathbb{L}_2(SE(2))$



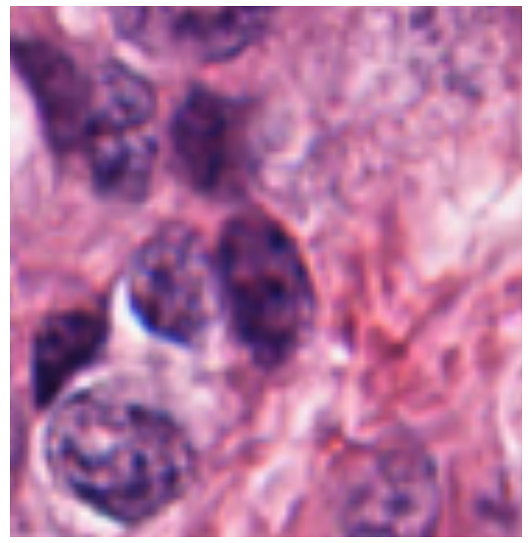
$$(k \tilde{\star} f)(\mathbf{x}) = (\mathcal{L}_g^{SE(2)\to\mathbb{L}_2(SE(2))} k\ , f)_{\mathbb{L}_2(SE(2)}$$

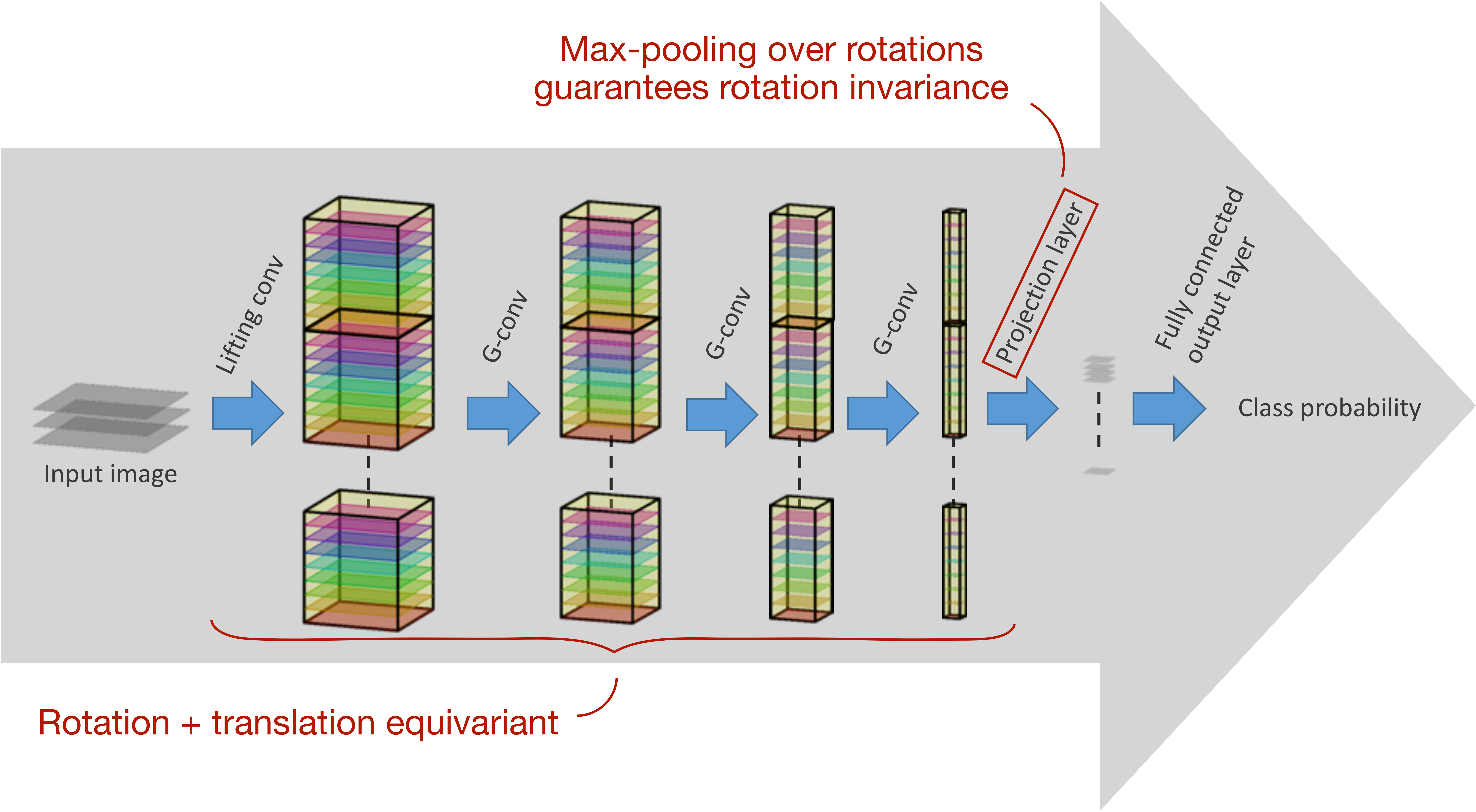
$$= \int_{\mathbb{R}^2}\int_{S^1} k(\mathbf{R}_\theta^{-1}\mathbf{x}' - \mathbf{x}, \theta' - \theta \mod 2\pi) f(\mathbf{x}', \theta') \mathrm{d}\mathbf{x}'$$

No constraints

The most expressive group equivariant architectures are obtained by lifting the feature maps to the group

General group equivariant architecture





“normal” (0)
vs
“mitotic” (1)

Content

Part I: Introduction to group convolutions

* Motivation
* Introduction to group theory
* Regular group convolutional neural networks
* Applications

Part II: General theory for group equivariant deep learning

* Group convolutions are all you need!
* Deeper into group theory: representation theory, homogeneous spaces
* Characterization of types of group equivariant layers

Part III: Steerable group convolutions

* Deep dive into group theory: irreducible representations, steerable operators and vector spaces
* Examples of steerable group convolutions: Spherical data and Volumetric data/3D point clouds

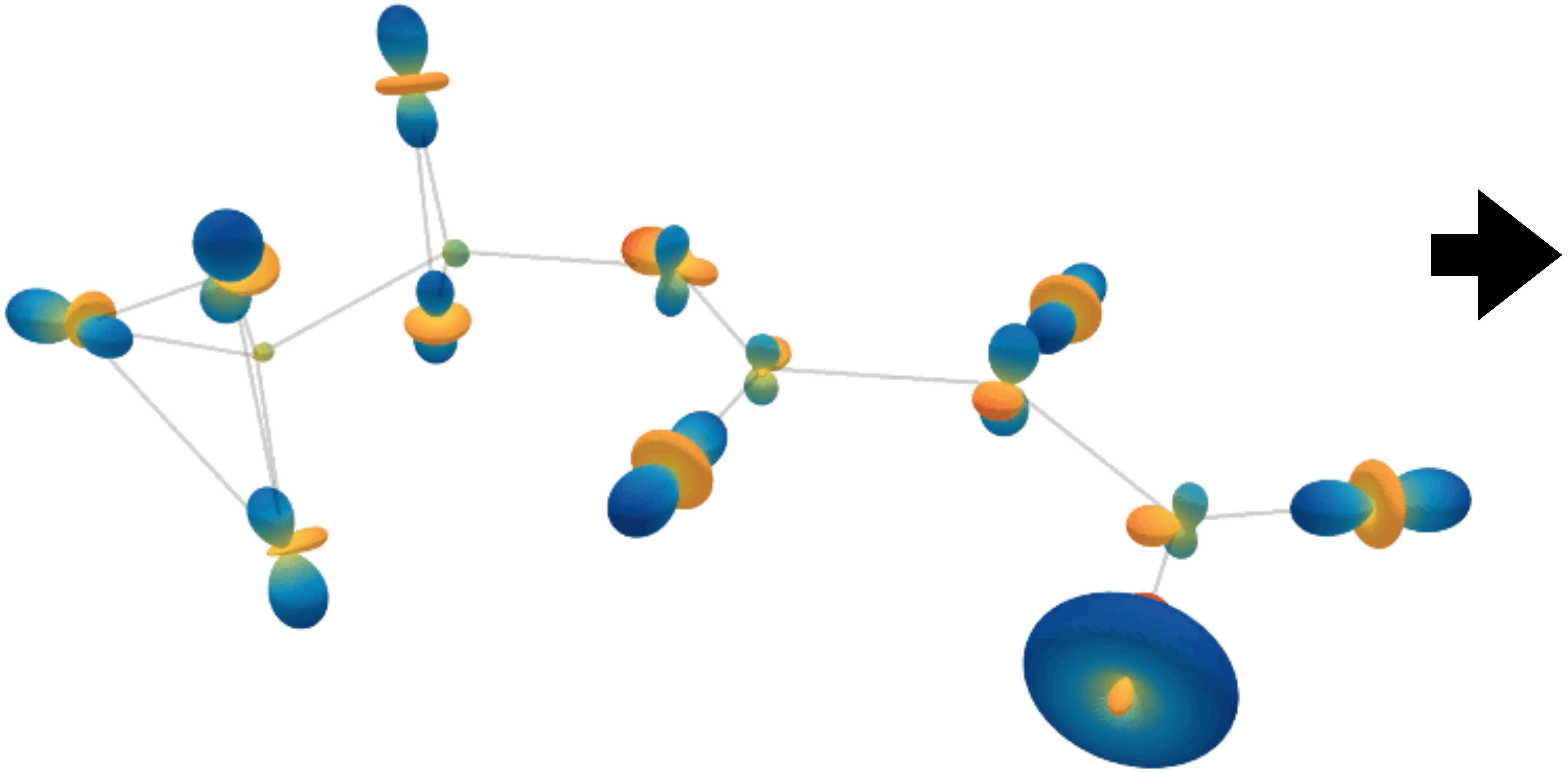
The need for steerable G-CNNs

Steerable methods are designed for groups that involve the action of $SO(d)$:

- Are based on a Fourier convolution theorem on $SO(d)$
- Avoids discretization of $SO(d)$:
  - Numerically more precise than regular group convolutions
  - Exact equivariance
  - Flexible to non-gridded data
- Provide a roadmap to local equivariance on arbitrary manifolds through Gauge theory

Steerable methods for computational chemistry

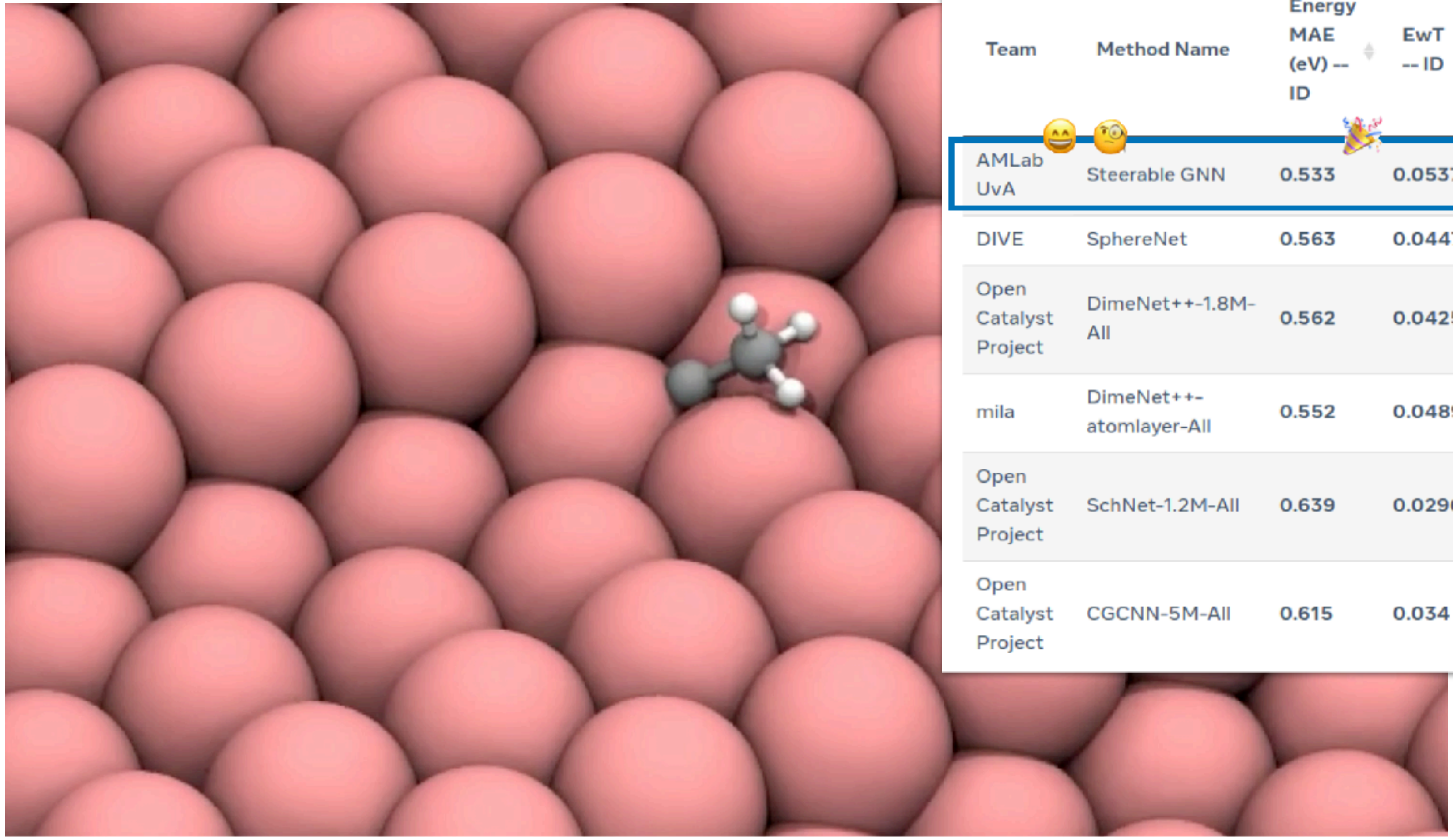
Brandstetter, Hesselink, van der Pol, Bekkers, Welling **Steerable Equivariant Message Passing on Molecular Graphs**



Molecular property prediction

Steerable methods for computational chemistry

Brandstetter, Hesselink, van der Pol, Bekkers, Welling **Steerable Equivariant Message Passing on Molecular Graphs**



Team	Method Name	Energy MAE (eV) -- ID	EwT -- ID	Energy MAE (eV) -- OOD Ads	EwT -- OOD Ads	Energy MAE (eV) -- OOD Cat	EwT -- OOD Cat	Energy MAE (eV) -- OOD Both	EwT -- OOD Both	Submitted
AMLab UvA	Steerable GNN	0.533	0.0537	0.692	0.0246	0.537	0.0492	0.679	0.0263	2021/06/12
DIVE	SphereNet	0.563	0.0447	0.703	0.0229	0.571	0.0409	0.638	0.0241	2021/03/31
Open Catalyst Project	DimeNet++-1.8M-All	0.562	0.0425	0.725	0.0207	0.576	0.041	0.661	0.0241	2021/02/16
mila	DimeNet++-atomlayer-All	0.552	0.0489	0.747	0.0259	0.557	0.0459	0.688	0.0233	2021/06/07
Open Catalyst Project	SchNet-1.2M-All	0.639	0.0296	0.734	0.0233	0.662	0.0294	0.704	0.0221	2021/02/17
Open Catalyst Project	CGCNN-5M-All	0.615	0.034	0.915	0.0193	0.622	0.031	0.851	0.02	2021/02/18

Video: Open Catalyst Project

Group theoretical background

Irreducible Representations (spherical harmonics, Wigner-D matrices)

Fourier transform on $SO(3)$

Convolution theorem

+ Clebsch-Gordan Tensor product

Steerable G-CNNs

Group theory: Irreducible Representations

Definition 4.1 (Equivalence of matrix representations). Any two matrix representations $\mathbf{D}(g)$ and $\mathbf{D}'(g)$ of a group G are *equivalent* if they relate via a similarity transform

$$\mathbf{D}'(g) = \mathbf{Q}^{-1}\mathbf{D}(g)\mathbf{Q} \; ,$$

in which $\mathbf{Q}$ carries out the change of basis.

Definition 4.2 (Reducible/irreducible matrix representation). A matrix representation is called reducible if it can be written as

$$\mathbf{D}(g) = \mathbf{Q}^{-1}(\mathbf{D}_1(g) \oplus \mathbf{D}_2(g))\mathbf{Q} = \mathbf{Q}^{-1}\begin{pmatrix}\mathbf{D}_1(g) & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2(g)\end{pmatrix}\mathbf{Q} \; ,$$

in which $\mathbf{Q}$ carries out a change of basis. If the matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ are not reducible they are called irreducible representations (irreps).

Group theory: Wigner-D Matrices

Wigner-D matrices are the irreducible matrix representations of $SO(3)$

Every representation $\mathbf{D}(g)$ of $SO(3)$ is block diagonalizable to a representation with Wigner-D matrices along the diagonal:

$$\mathbf{D}(g) = \mathbf{Q}^{-1}(\mathbf{D}^{(l_1)}(g) \oplus \mathbf{D}^{(l_2)}(g) \oplus \dots)\mathbf{Q} = \mathbf{Q}^{-1} \begin{pmatrix} \mathbf{D}^{(l_1)}(g) & & \\ & \mathbf{D}^{(l_2)}(g) & \\ & & \ddots \end{pmatrix} \mathbf{Q}$$

Group theory: Steerable vector space

Definition 4.3 (Wigner-D matrix)**.** The Wigner-D matrices of type-l are the irreducible $(2l+1)$ matrix representations of $SO(3)$. A Wigner-D matrix of type l as a function of $g \in G$ will be denoted with $\mathbf{D}^{(l)}(g)$.

Wigner-D matrices generalize the notion of a rotation matrix for the rotation of $(2l+1)$-dimensional vectors

Group theory: Steerable vector space

Definition 4.3 (Wigner-D matrix)**.** The Wigner-D matrices of type-l are the irreducible $(2l+1)$ matrix representations of $SO(3)$. A Wigner-D matrix of type l as a function of $g \in G$ will be denoted with $\mathbf{D}^{(l)}(g)$.

Definition 4.4 (Wigner-D functions)**.** The $(2l+1) \times (2l+1)$ components of the type-l Wigner-D matrices will be referred to as the type-l Wigner-D functions. The Wigner-D functions are denoted with $D^{(l)}_{mn}$ with m and n row and column index respectively.

Definition 4.5 (Steerable vector spaces and steerable vectors)**.** The $(2l+1)$-dimensional vector space on which a Wigner-D matrix of order l acts will be called *a type l steerable vector space* and is denoted with V_l. A $(2l+1)$-dimensional vector $\mathbf{v} \in V_l$ will be called a type-l vector.

Group theory: Spherical Harmonics

- Functions on the sphere
- Solutions to Laplace's equation on S^2
- The S^2 equivalent of the circular harmonics (1D Fourier basis)
- Form orthonormal basis for $\mathbb{L}_2(S^2)$
- Are Wigner-D functions:

$$Y_m^{(l)} = \mathbf{D}_{m0}^{(l)}$$

Spherical harmonics $Y_m^{(l)} : S^2 \to \mathbb{R}$

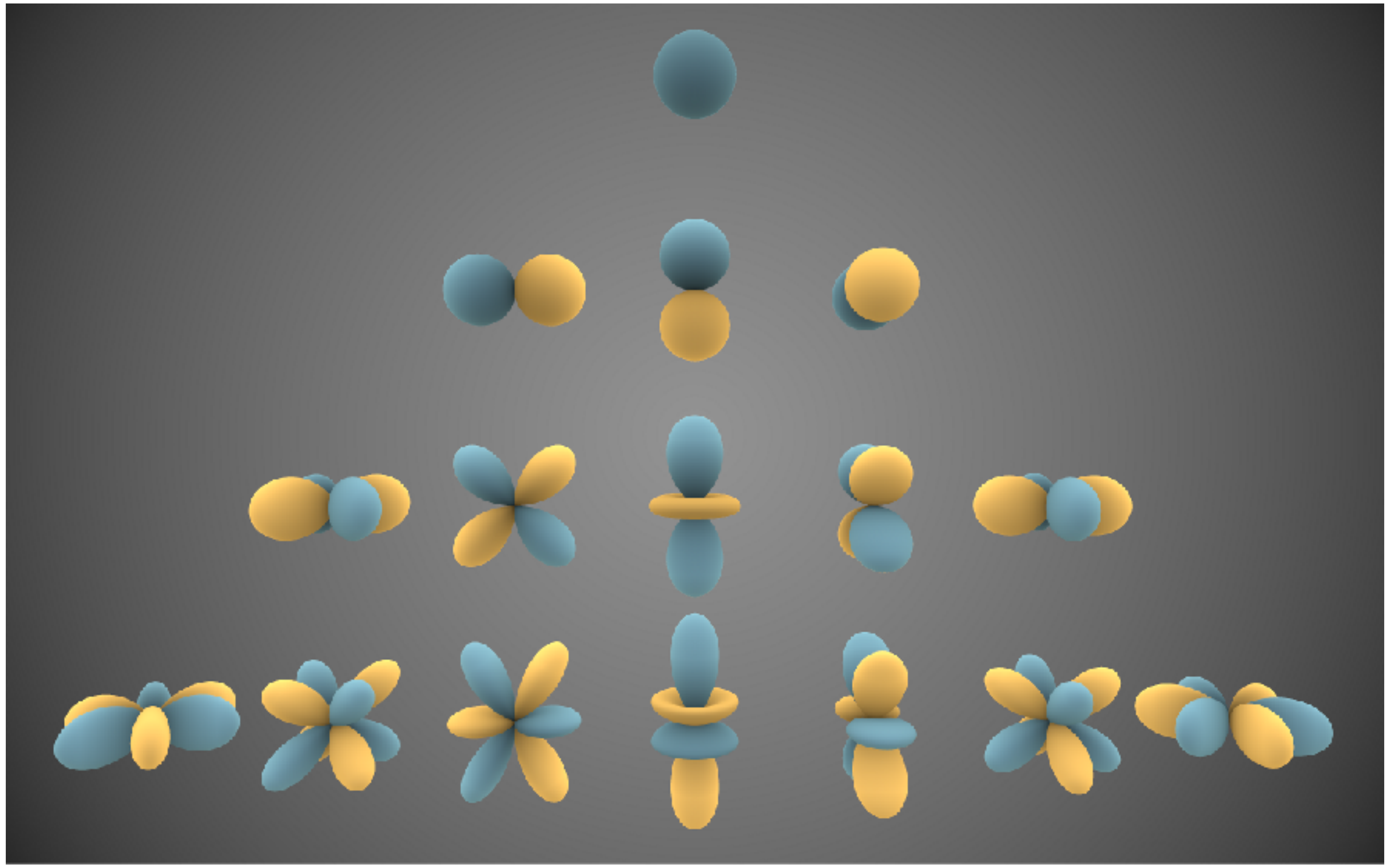
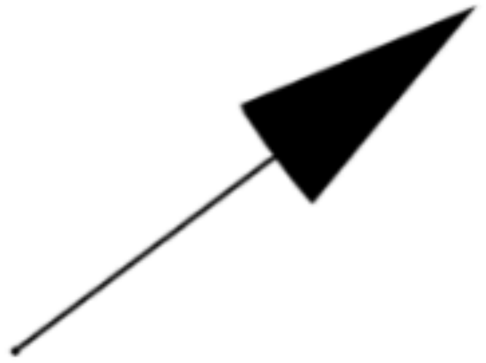


Image: wikipedia

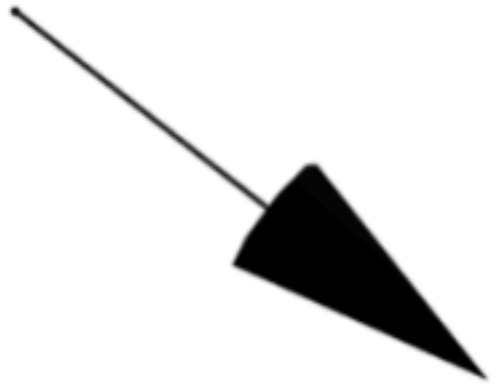
Group theory: Spherical Harmonics Form Steerable Vectors

$\mathbf{x}$ $\quad \tilde{\mathbf{h}}^{(l)} = Y^{(l)}(\mathbf{x})$ $\quad Y_m^{(l)}(\,\cdot\,)$ $\quad \sum h_m^l Y_m^{(l)}(\,\cdot\,)$



$$\begin{bmatrix} h_0^0 \end{bmatrix} \quad \begin{bmatrix} h_0^1 \\ h_1^1 \\ h_2^1 \end{bmatrix} \quad \begin{bmatrix} h_0^2 \\ h_1^2 \\ h_2^2 \\ h_3^2 \\ h_4^2 \end{bmatrix}$$

$\mathbf{R}\mathbf{x}$ $\quad \tilde{\mathbf{h}}^{(l)} = \mathbf{D}^l(g) Y^{(l)}(\mathbf{x})$ $\quad Y_m^{(l)}(\,\cdot\,)$ $\quad \sum h_m^l Y_m^{(l)}(\,\cdot\,)$



$$\mathbf{D}^0(g)\begin{bmatrix} h_0^0 \end{bmatrix} \quad \mathbf{D}^1(g)\begin{bmatrix} h_0^1 \\ h_1^1 \\ h_2^1 \end{bmatrix} \quad \mathbf{D}^2(g)\begin{bmatrix} h_0^2 \\ h_1^2 \\ h_2^2 \\ h_3^2 \\ h_4^2 \end{bmatrix}$$

Group theory: Fourier Transform on S^2

Definition 4.6 (Spherical Fourier transform). Let $f \in \mathbb{L}_2(S^2)$ be a spherical signal and let $\hat{f}(l) \in \mathbb{R}^{2l+1}$ denote the vector of Fourier coefficients of order l. We may refer to l as the frequency index. The forward and inverse Fourier transform are respectively given by

$$\hat{f}(l) = \int_{S^2} f(\mathbf{n}) \underline{Y}^{(l)}(\mathbf{n}) d\mathbf{n} \tag{64}$$

$$f(\mathbf{n}) = \sum_{l \geq 0} \hat{f}(l)^T \underline{Y}^{(l)}(\mathbf{n}), \tag{65}$$

with $\underline{Y}^{(l)} = (Y_{-l}^{(l)}, \ldots, Y_{l}^{(l)})^T \in \mathbb{L}_2(S^2)^{2l+1}$ the vector of spherical harmonics.

Group theory: Fourier Transform on S^2

Definition 4.7 ($SO(3)$ **Fourier transform).** Let $f \in \mathbb{L}_2(SO(3))$ be a spherical signal and let $\hat{f}(l) \in \mathbb{R}^{2l+1\times 2l+1}$ denote the matrix of Fourier coefficients of order l. The number l may be referred to as frequency index. The forward and inverse Fourier transform are respectively given by

$$\hat{f}(l) = \int_{SO(3)} f(g)\mathbf{D}^{(l)}(g)\mathrm{d}g \ , \tag{68}$$

$$f(g) = \sum_{l\geq 0} \mathrm{Tr}(\hat{f}(l)\mathbf{D}^{(l)}(g^{-1})) \ . \tag{69}$$

$$\sum_{l\geq 0} \sum_{m=-l}^{l} \hat{f}^{(l)}_{mn} D^{(l)}_{mn}(g^{-1})$$

Group theory: $SO(3)$ Fourier Theorems

Lemma 4.1 (Shift property)**.** Let $\mathcal{L}_g$ denote the left-regular representation of $SO(3)$ on $\mathbb{L}_2(SO(3))$ and $\hat{f}$ denote the $SO(3)$ Fourier transform $SO(3)$ (Definition 4.7). The $SO(3)$ Fourier transform is equivariant via

$$\widehat{\mathcal{L}_g f}(l) = \mathbf{D}^{(l)}(g)\hat{f}(l)\,. \tag{70}$$

Theorem 4.1 (Convolution theorem on $SO(3)$)**.** Let $k, f \in \mathbb{L}_2(SO(3))$ and let $\hat{k}(l), \hat{f}(l)$ denote their matrix valued Fourier coefficients. Then the Fourier transform of a (group) correlation of a k with f is given by

$$\widehat{k \star f}(l) = \hat{f}(l)\hat{k}(l)^T\,. \tag{71}$$

Group theory: Clebsch-Gordan Tensor Product

General tensor product between two vectors:

Group theory: Clebsch-Gordan Tensor Product

General tensor product between two vectors:

$$\mathbf{h}_1 \otimes \mathbf{h}_2 = \mathbf{h}_1 \mathbf{h}_2^T = \begin{pmatrix} h_1 h_1 & h_1 h_2 & \dots \\ h_2 h_1 & h_2 h_2 & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

Group theory: Clebsch-Gordan Tensor Product

We want the tensor product to be equivariant via

$$\mathbf{D}(g)(\tilde{\mathbf{h}}_1 \otimes \tilde{\mathbf{h}}_2) = (\mathbf{D}^{(l_1)}(g)\tilde{\mathbf{h}}_1) \otimes (\mathbf{D}^{(l_2)}(g)\tilde{\mathbf{h}}_2)$$

for some $\mathbf{D}(g)$

Group theory: Clebsch-Gordan Tensor Product

The tensor product between two steerable vectors results again in a steerable vector:

$$\mathrm{vec}\left((\mathbf{D}^{(l_1)}(g)\tilde{\mathbf{h}}_1)(\mathbf{D}^{(l_2)}(g)\tilde{\mathbf{h}}_2)^T \right) = \mathrm{vec}\left(\mathbf{D}^{(l_1)}(g)\tilde{\mathbf{h}}_1\tilde{\mathbf{h}}_2^T\mathbf{D}^{(l_2)T}(g) \right)$$
$$= \boxed{\left(\mathbf{D}^{(l_2)}(g) \otimes \mathbf{D}^{(l_1)}(g)\right)} \mathrm{vec}\left(\tilde{\mathbf{h}}_1\tilde{\mathbf{h}}_2^T \right)$$

The resulting representation is reducible.

The CG-product $\otimes_{cg}$ is defined in such a way that the output is directly obtained in direct sum of steerable vector spaces

$\tilde{\mathbf{h}}_1 \otimes_{cg} \tilde{\mathbf{h}}_2 \in V_0 \oplus V_1 \oplus \ldots$

Group theory: Clebsch-Gordan Tensor Product

Definition 4.8 (Clebsch-Gordan tensor product). Let $\tilde{\mathbf{h}}^{(l)} \in V_l = \mathbb{R}^{2l+1}$ denote a steerable vector of type l and $h_m^{(l)}$ its components with $m = -l, -l+1, \ldots, l$. Then the Clebsch-Gordan tensor product is defined is a tensor product such tat the m-th component of the type l sub-vector of the output of the tensor product between two steerable vectors of type l_1 and l_2 is given by

$$(\tilde{\mathbf{h}}^{(l_1)} \otimes_{cg} \tilde{\mathbf{h}}^{(l_2)})_m^{(l)} = \sum_{m_1=-l_1}^{l_1} \sum_{m_2=-l_2}^{l_2} C_{(l_1,m_1)(l_2,m_2)}^{(l,m)} h_{m_1}^{(l_1)} h_{m_2}^{(l_2)} \,, \tag{73}$$

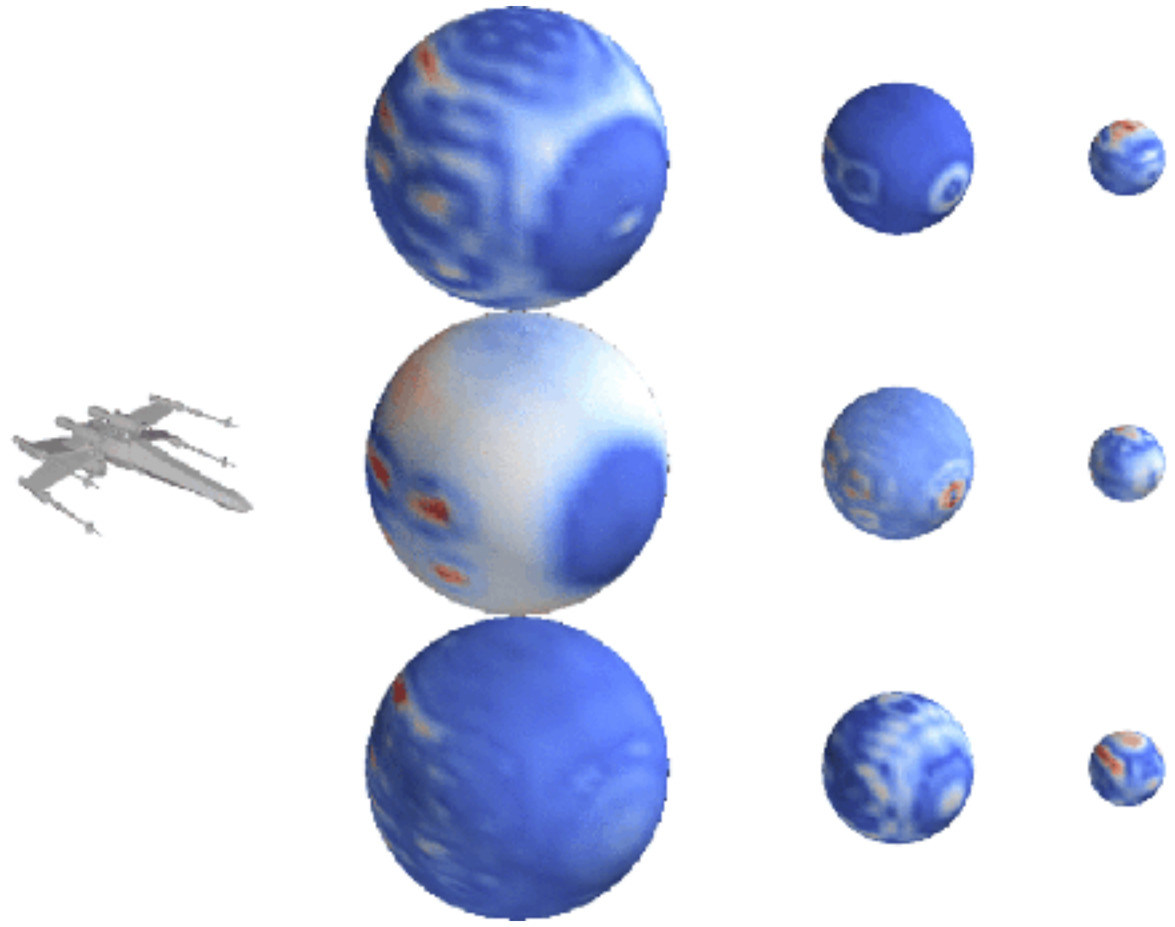
in which $C_{(l_1,m_1)(l_2,m_2)}^{(l,m)}$ are the Clebsch-Gordan coefficients. The l-th output vector $(\tilde{\mathbf{h}}^{(l_1)} \otimes_{cg} \tilde{\mathbf{h}}^{(l_2)})^{(l)} \in \mathbb{R}^{2l+1}$ is a type-l steerable vector.

Steerable Neural Networks

Steerable Neural Networks

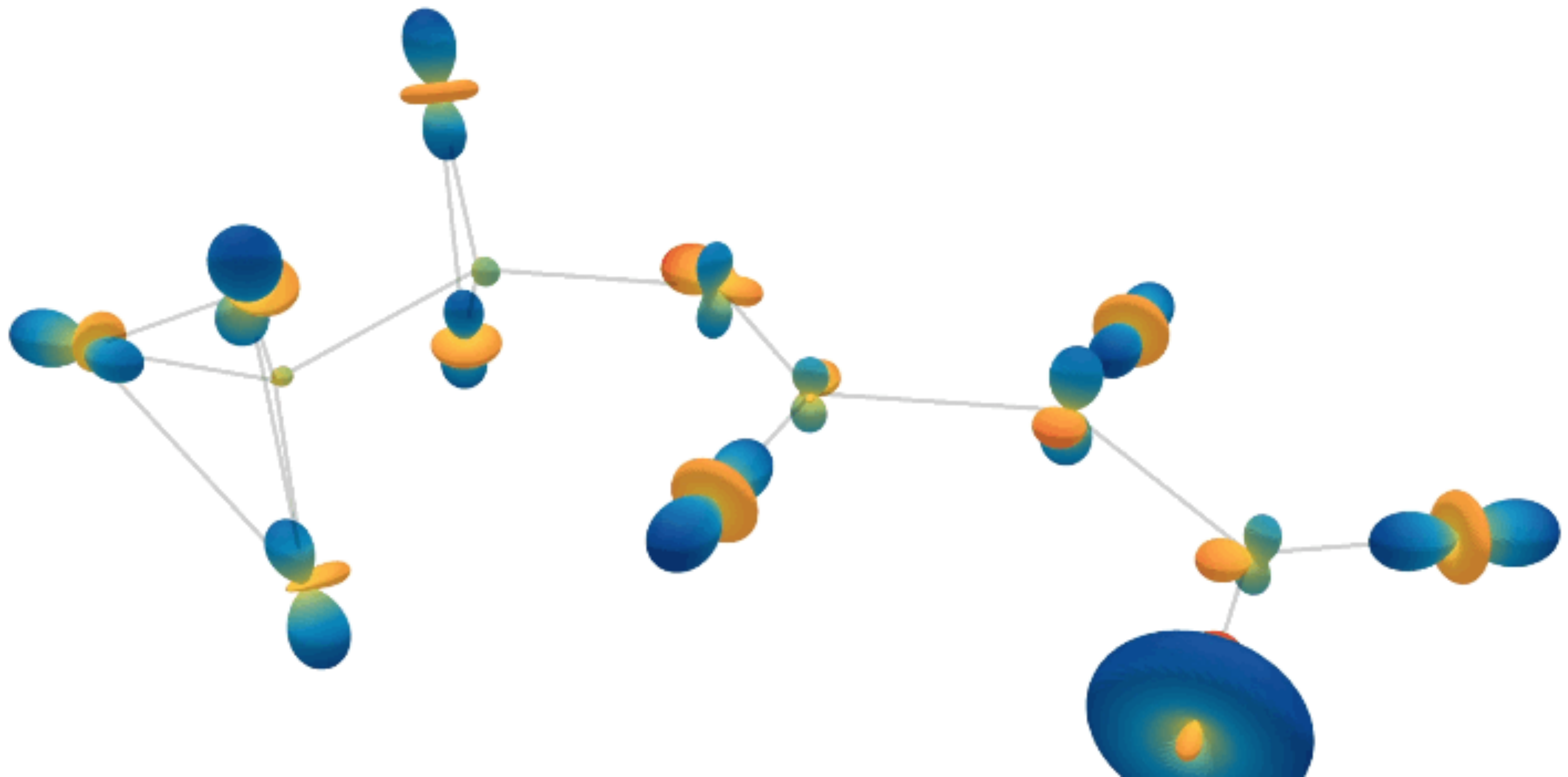
Method 1: Convolutions in the Fourier domain ($SO(3)$ gconvs)

Lecture notes Section 5.1



Method 2: Clebsch-Gordan tensor product ($SE(3)$ gconvs)

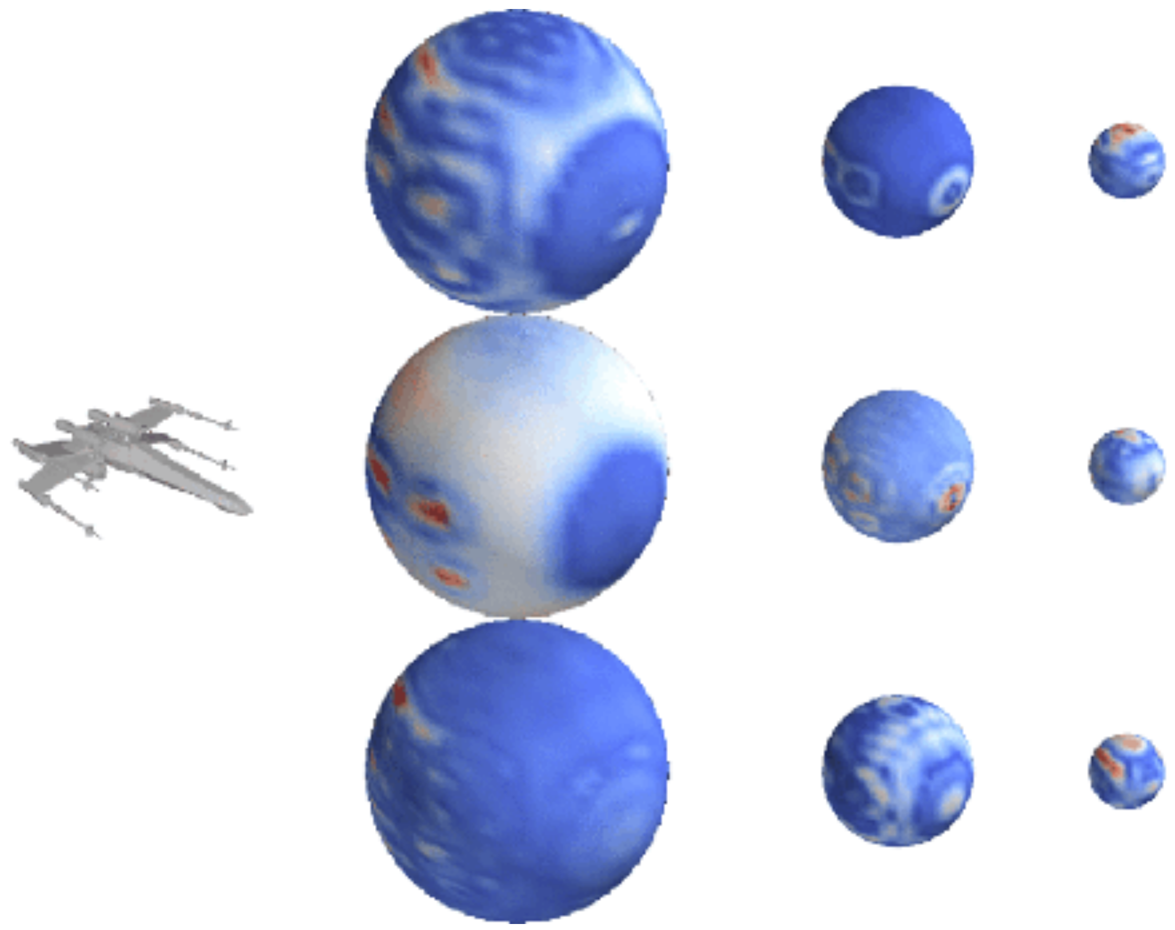
Lecture notes Section 5.2



Steerable Neural Networks

Method 1: Convolutions in the Fourier domain ($SO(3)$ gconvs)

Lecture notes Section 5.1



Method 2: Clebsch-Gordan tensor product ($SE(3)$ gconvs)

Lecture notes Section 5.2

Spherical group convolutions in the Fourier domain

Spherical CNNs are build via convolutions in the Fourier domain

$$\widehat{k \star f}(l) = \hat{f}(l)\hat{k}^T(l)$$

In vectorized form this is simply a matrix vector multiplication

$$\mathrm{vec}(\widehat{k \star f})(l) = (\hat{w}(l) \otimes I)\,\mathrm{vec}(\hat{f}(l))$$

Spherical group convolutions in the Fourier domain

Spherical CNNs are build via convolutions in the Fourier domain

$$\widehat{k \star f}(l) = \hat{f}(l)\hat{k}^T(l)$$

Let us make it explicit and consider signals with only frequency $l = 1$.
Such signals are represented with 3-dimensional vectors.

$$\mathrm{vec}(\hat{f}) = \begin{pmatrix} \hat{f}_{:,-1} \\ \hat{f}_{:,0} \\ \hat{f}_{:,+1} \end{pmatrix}$$

Spherical group convolutions in the Fourier domain

Spherical CNNs are build via convolutions in the Fourier domain

$$\widehat{k \star f}(l) = \hat{f}(l)\hat{k}^T(l)$$

Let us make it explicit and consider signals with only frequency $l = 1$.
Such signals are represented with 3-dimensional vectors.

$$\mathrm{vec}(\hat{f}) = \begin{pmatrix} \hat{f}_{:,-1} \\ \hat{f}_{:,0} \\ \hat{f}_{:,+1} \end{pmatrix}$$

For S^2 signals: red=zero

Spherical group convolutions in the Fourier domain

Spherical CNNs are build via convolutions in the Fourier domain

$$\widehat{k \star f}(l) = \hat{f}(l)\hat{k}^T(l)$$

Let us make it explicit and consider signals with only frequency $l = 1$.
Then

$$\begin{pmatrix} \widehat{k \star f}_{:,-1} \\ \widehat{k \star f}_{:,0} \\ \widehat{k \star f}_{:,+1} \end{pmatrix} = \begin{pmatrix} w_{-1,-1}I & w_{-1,0}I & w_{-1,1}I \\ w_{0,-1}I & w_{0,0}I & w_{0,1}I \\ w_{1,-1}I & w_{1,0}I & w_{1,1}I \end{pmatrix} \begin{pmatrix} \hat{f}_{:,-1} \\ \hat{f}_{:,0} \\ \hat{f}_{:,+1} \end{pmatrix}$$

Only three weights: the kernel is also a spherical harmonic!

Spherical group convolutions in the Fourier domain

Spherical CNNs are build via convolutions in the Fourier domain

$$\widehat{k \star f}(l) = \hat{f}(l)\hat{k}^T(l)$$

Let us make it explicit and consider signals with only frequency $l = 1$.
Then

$$\begin{pmatrix} \widehat{k \star f}_{:,-1} \\ \widehat{k \star f}_{:,0} \\ \widehat{k \star f}_{:,+1} \end{pmatrix} = \begin{pmatrix} w_{-1,-1}I & w_{-1,0}I & w_{-1,1}I \\ w_{0,-1}I & w_{0,0}I & w_{0,1}I \\ w_{1,-1}I & w_{1,0}I & w_{1,1}I \end{pmatrix} \begin{pmatrix} \hat{f}_{:,-1} \\ \hat{f}_{:,0} \\ \hat{f}_{:,+1} \end{pmatrix}$$

If we want the output to be a SH, only 1 weight can be non-zero!

Spherical group convolutions in the Fourier domain

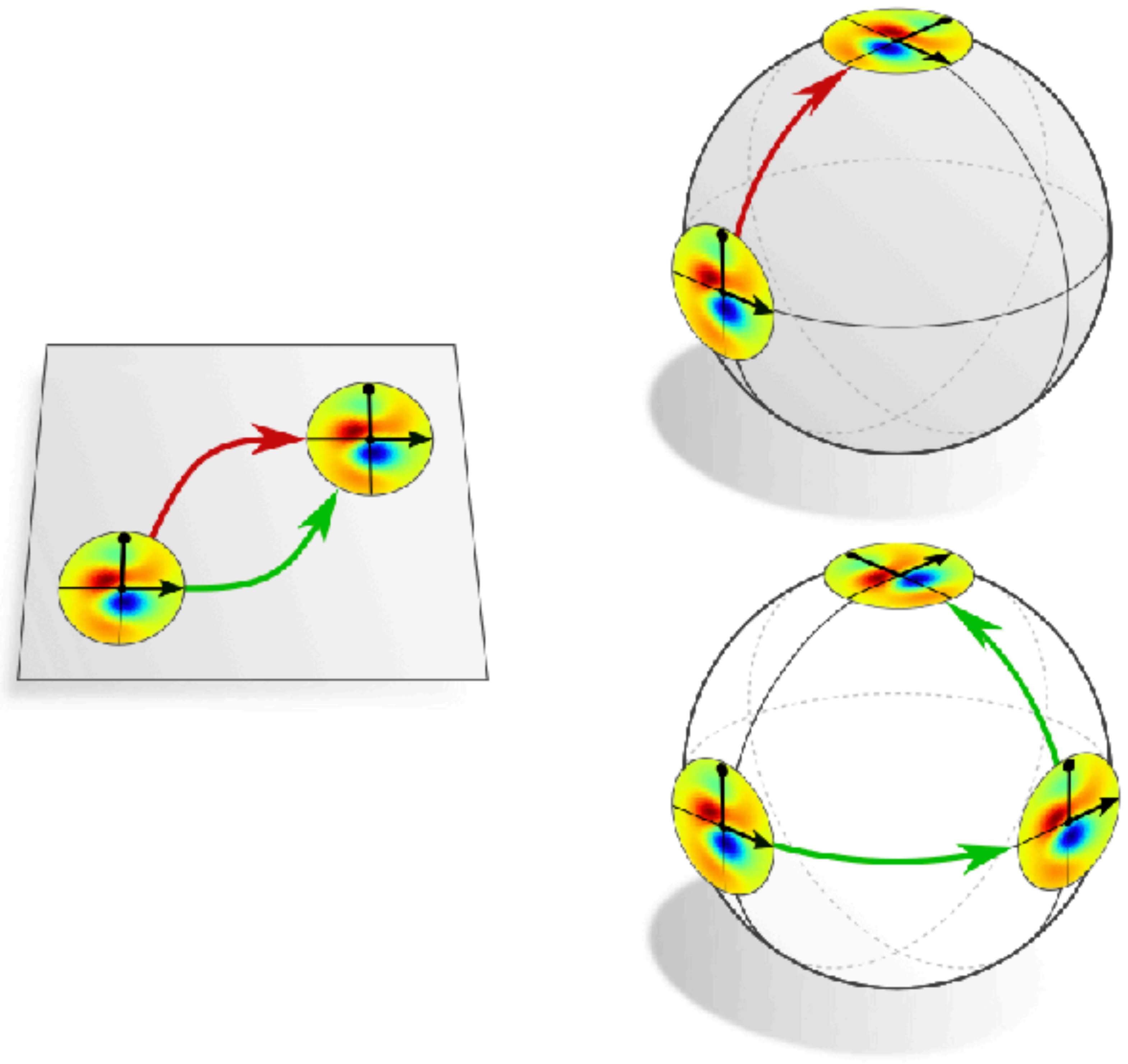


Figure from COORDINATE INDEPENDENT CONVOLUTIONAL NETWORKS, Weiler, Forré, Verlinde, Welling

Lemma 5.1. Consider the cases of $SO(3)$ equivariant linear layers for signals on $X = S^2 \equiv SO(3)/SO(2)$ or $X = SO(3)$. Such convolutions can be performed via the Fourier transform on $SO(3)$

$$\widehat{k \star f}(l) = \hat{f}(l)\hat{w}^T(l)\,,$$

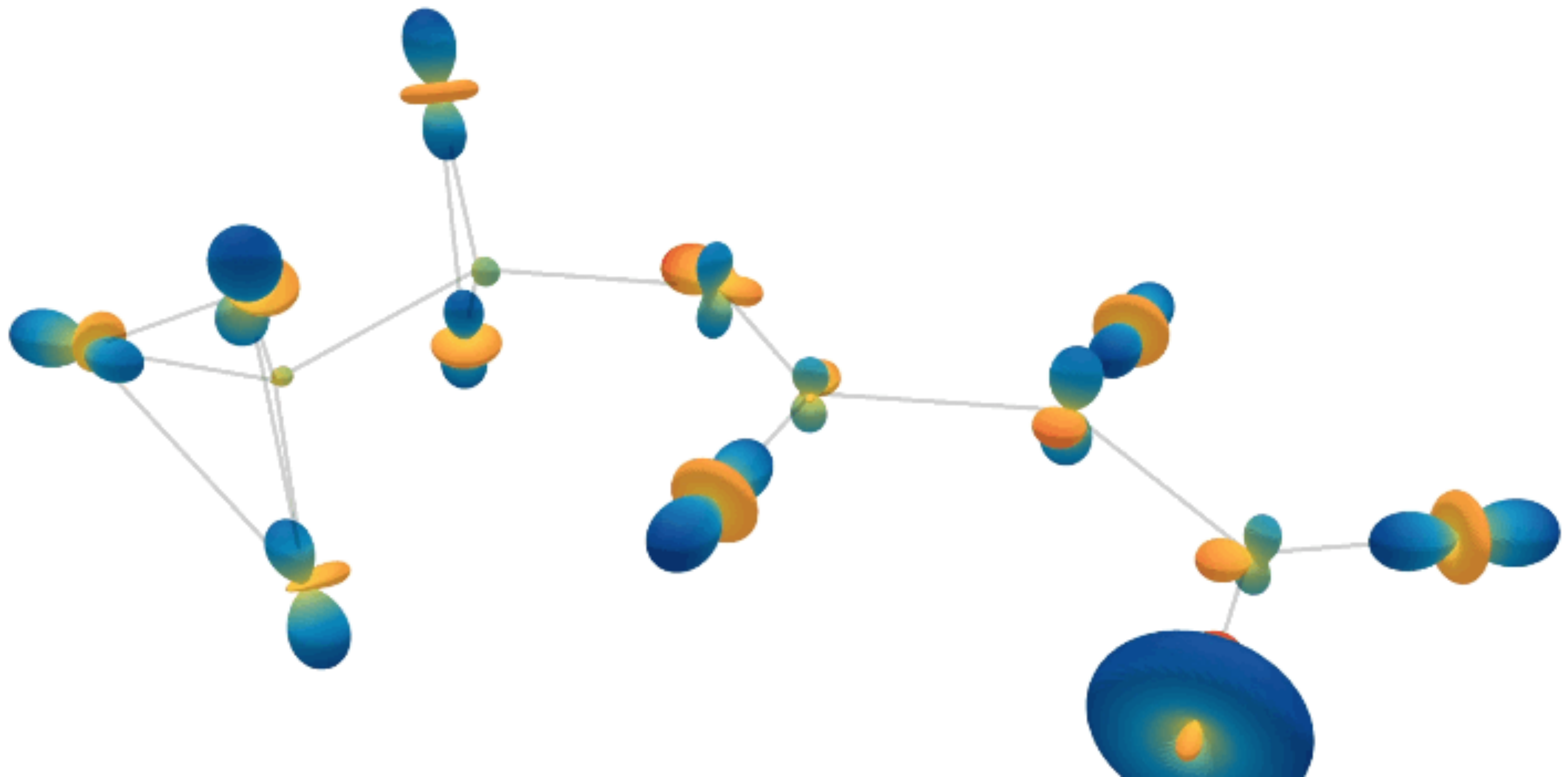
with $\hat{w}(l) \in \mathbb{R}^{2l+1 \times 2l_1}$ the learnable parameters of the convolution kernel. The following can be said about the parametrization of the kernels through $\hat{w}$.

- **Isotropic S^2 kernel convolutions** $(X = Y = S^2)$ For isotropic kernel convolutions the kernel is parametrized by a single weight per frequency l. Namely the only possibly non-zero component is $\hat{w}_{00}(l)$ and all others have to be 0. This means that the kernels represent signals on the sphere S^2 that are symmetry around the $\mathbf{n}_x$ axis, as required by Theorem 3.2.
- **Lifting convolutions** $(X = S^2, Y = SO(3))$ For lifting convolutions the kernel is parametrized by $(2l + 1)$ learnable weights per frequency l. The only possible non-zero weights are $\hat{w}_{:,0}$, i.e., the central column $(n = 0)$ of $\hat{w}(l)$. This means that the convolution kernels are unconstrained signals on the sphere S^2.
- **Group convolution** $(X = Y = SO(3))$ Group convolutions are parametrized by $(2l + 1) \times (2l + 1)$ learnable weights per frequency l. The kernels represent unconstrained functions on $SO(3)$.

Steerable Neural Networks

Method 2: Clebsch-Gordan tensor product $(SE(3)$ gconvs)

Lecture notes Section 5.2



Steerable Neural Networks

Steerable group convolutions

$$\tilde{\mathbf{f}}'(\mathbf{x}) = \int_{\mathbb{R}^3} \tilde{\mathbf{f}}(\mathbf{x}) \otimes_{cg}^{w(\|\mathbf{x}\|)} \underline{Y}^{(l)}\left(\frac{\mathbf{x}'-\mathbf{x}}{\|\mathbf{x}'-\mathbf{x}\|}\right) \mathrm{d}\mathbf{x}'$$

Angular part of k Spatial part of k

$\Leftrightarrow$

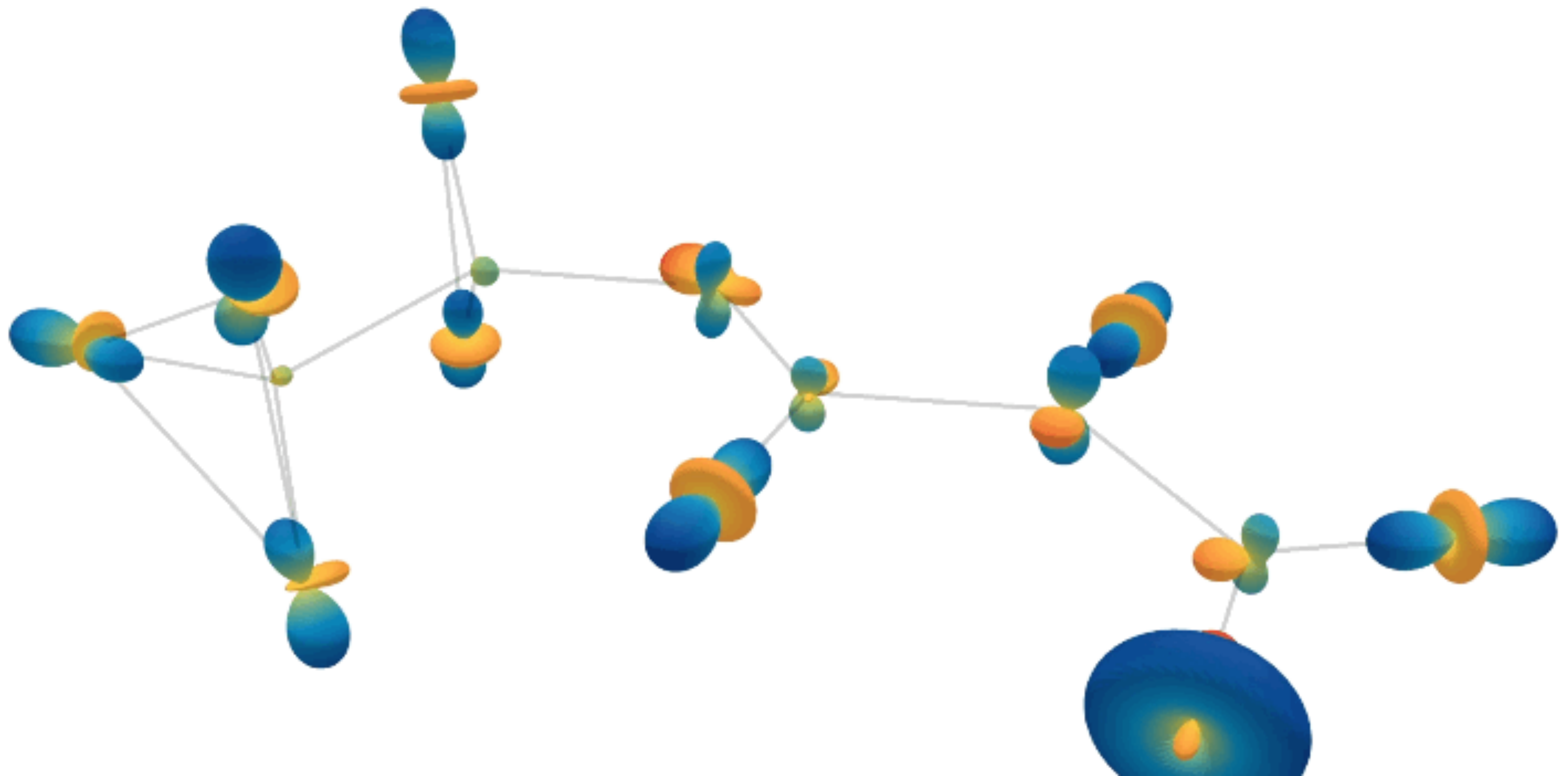
Regular group convolutions

$$f'(\mathbf{x}, \mathbf{R}) = \iint_{\mathbb{R}^3 \times SO(3)} k(\mathbf{R}^{-1}(\mathbf{x}' - \mathbf{x}), \mathbf{R}^{-1}\mathbf{R}')) f(\mathbf{x}', \mathbf{R}'))) \mathrm{d}\mathbf{x}' \mathrm{d}\mathbf{R}'$$

with $$k(\mathbf{x}) = \sum_{l}^{L} \sum_{m=-l}^{l} \tilde{\mathbf{w}}_m^{(l)}(\|\mathbf{x}\|) Y_m^{(l)}\left(\frac{\mathbf{x}}{\|\mathbf{x}\|}\right)$$

Method 2: Clebsch-Gordan tensor product ($SE(3)$ gconvs)

Lecture notes Section 5.2



Steerable Neural Networks

Steerable group convolutions

$$\tilde{\mathbf{f}}'(\mathbf{x}) = \int_{\mathbb{R}^3} \tilde{\mathbf{f}}(\mathbf{x}) \otimes_{cg}^{w(\|\mathbf{x}\|)} \underline{Y}^{(l)}\left(\frac{\mathbf{x}'-\mathbf{x}}{\|\mathbf{x}'-\mathbf{x}\|}\right) \mathrm{d}\mathbf{x}'$$

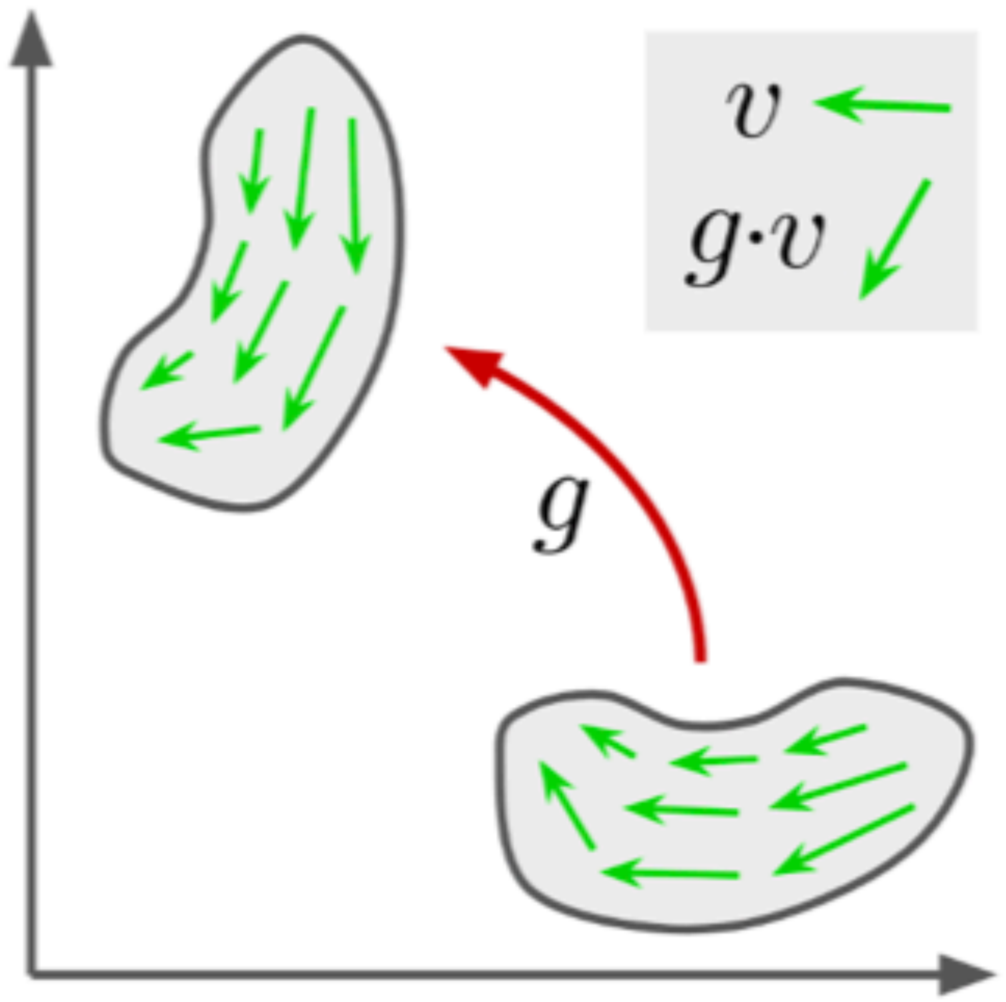
$\Leftrightarrow$

Regular group convolutions

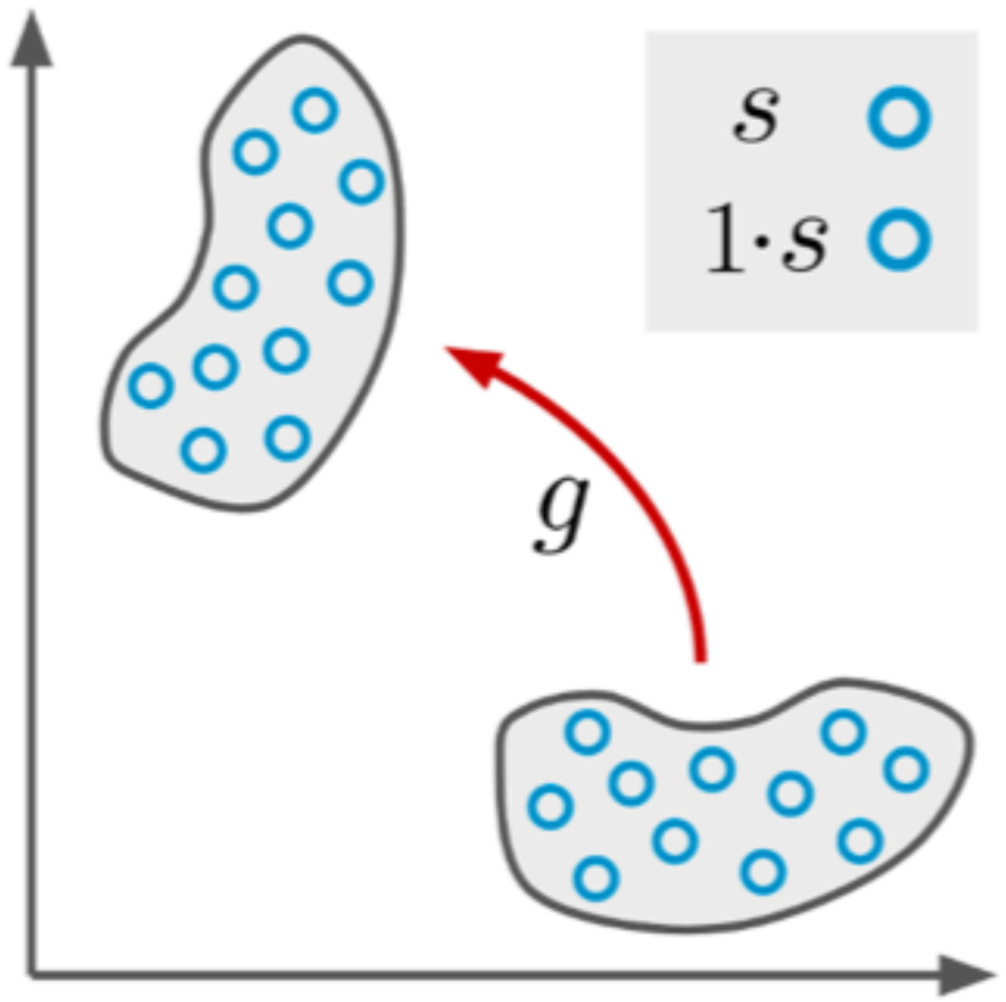
$$f'(\mathbf{x},\mathbf{R}) = \iint_{\mathbb{R}^3 \times SO(3)} k(\mathbf{R}^{-1}(\mathbf{x}'-\mathbf{x}), \mathbf{R}^{-1}\mathbf{R}'))f(\mathbf{x}',\mathbf{R}')))\mathrm{d}\mathbf{x}'\mathrm{d}\mathbf{R}'$$

with $\quad k(\mathbf{x}) = \sum_{l}^{L} \sum_{m=-l}^{l} \tilde{\mathbf{w}}_m^{(l)}(\|\mathbf{x}\|) Y_m^{(l)}\left(\frac{\mathbf{x}}{\|\mathbf{x}\|}\right)$

Feature maps: $f : \mathbb{R}^3 \to V_0 \oplus V_1 \oplus \ldots$

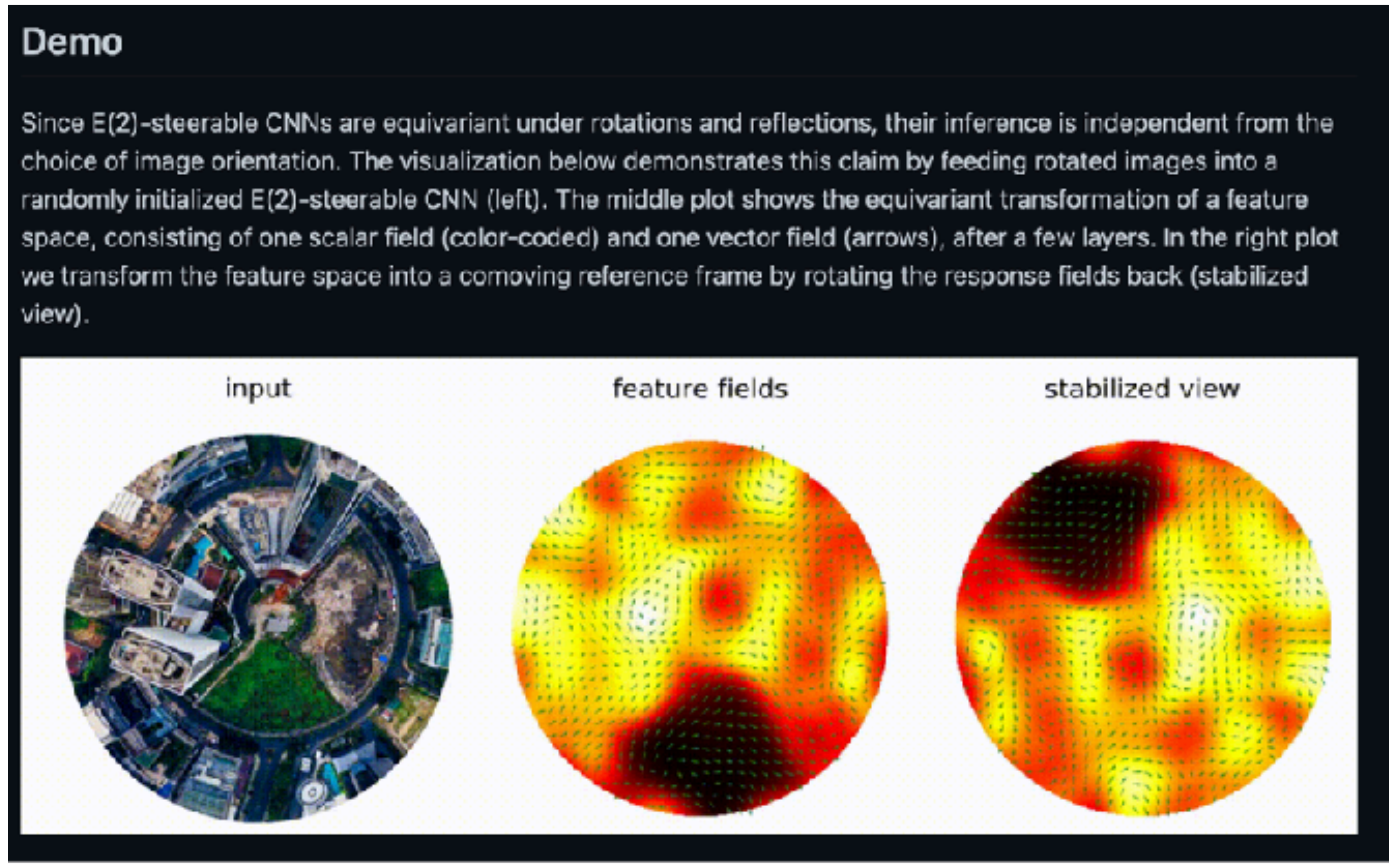


Feature maps: $f : \mathbb{R}^3 \times S^2 \to \mathbb{R}$

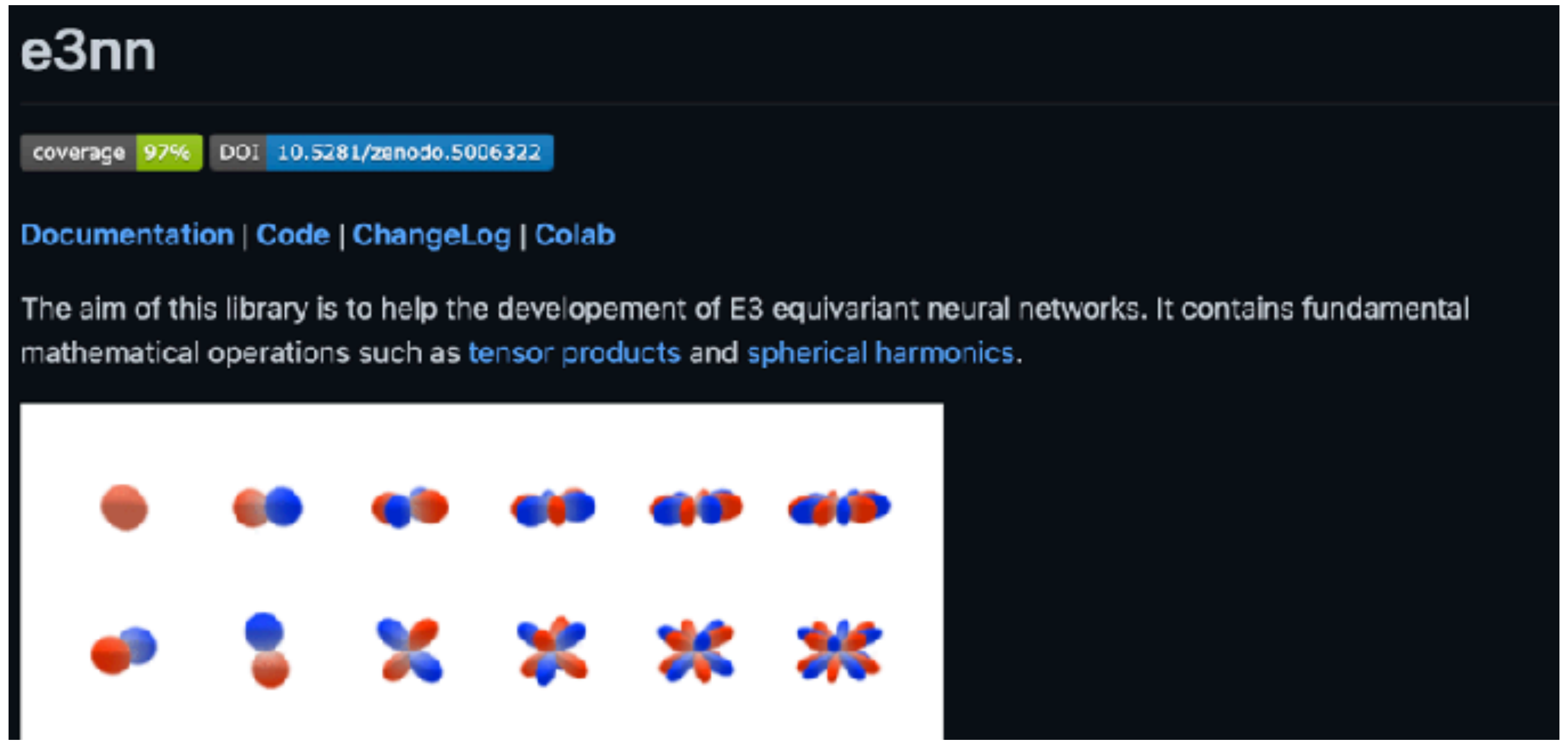


Libraries and repositories

e2cnn: https://github.com/QUVA-Lab/e2cnn



e3nn: https://github.com/e3nn/e3nn



e3cnn (released soon)

Conclusion

Conclusion

- **G-CNNs naturally arise** from NNs under equivariance constraints
- **G-CNNs improve upon classic CNNs** by
  - Making data augmentation w.r.t. the group obsolete
  - No valuable network capacity needs to be spend on dealing w geometry
  - The added geometric structure allows to deal with context (recognition by components, relative poses)
  - The added geometric structure enables to reach performances that cannot be achieved with data augmentation alone
  - Have guaranteed geometric stability
  - Can be applied to many types of signal data (not covered today: equivariance to Lie groups and gauge equivariant methods