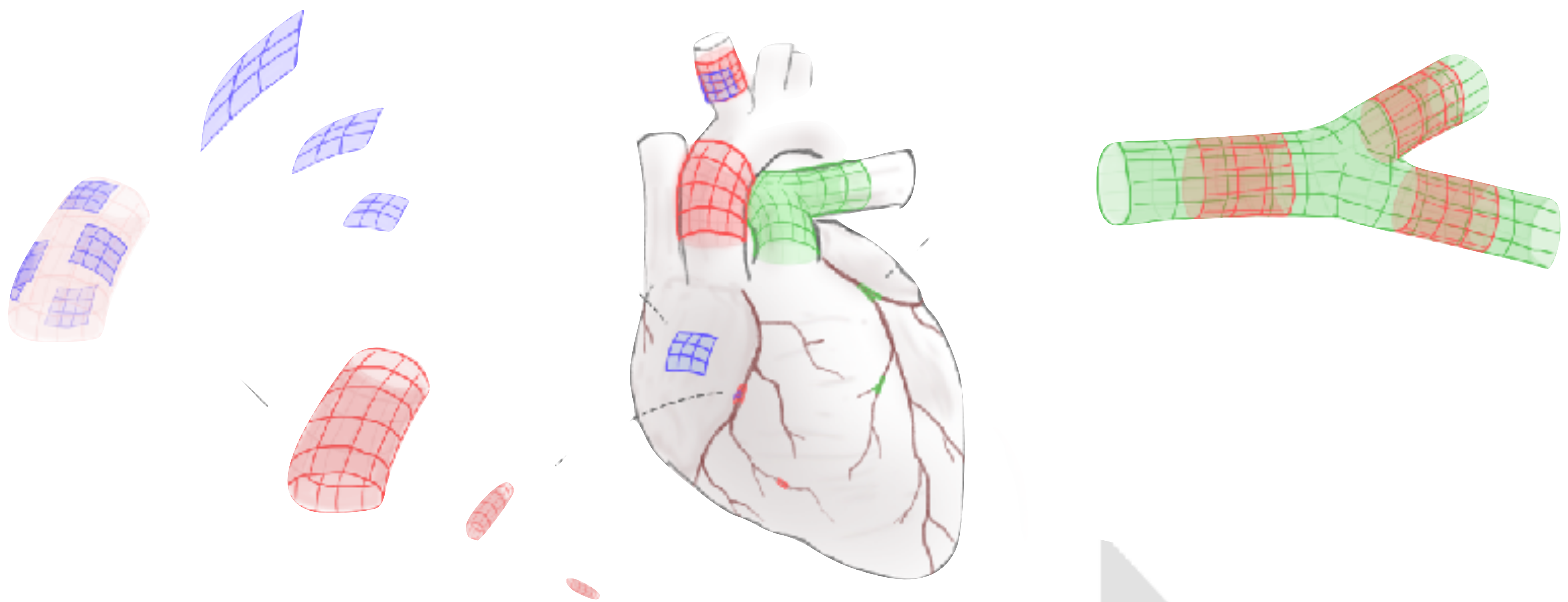
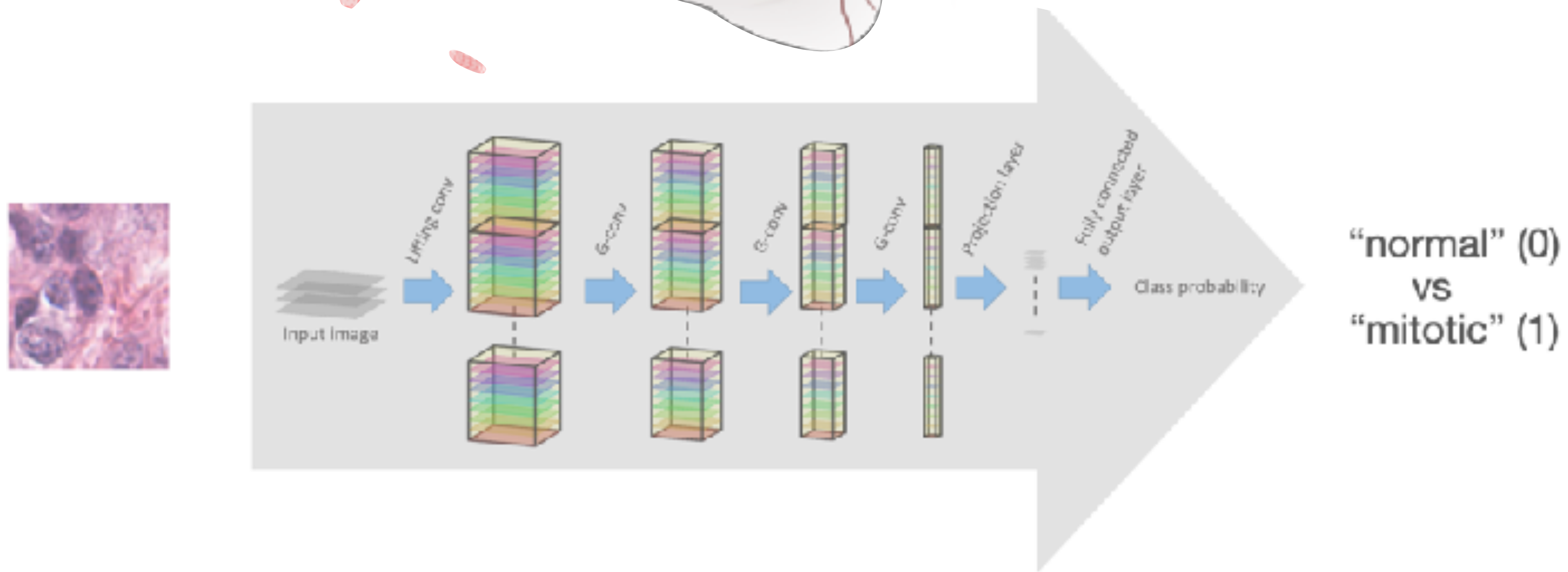
UNIVERSITY OF AMSTERDAM

Group equivariant deep learning and non-linear convolutions

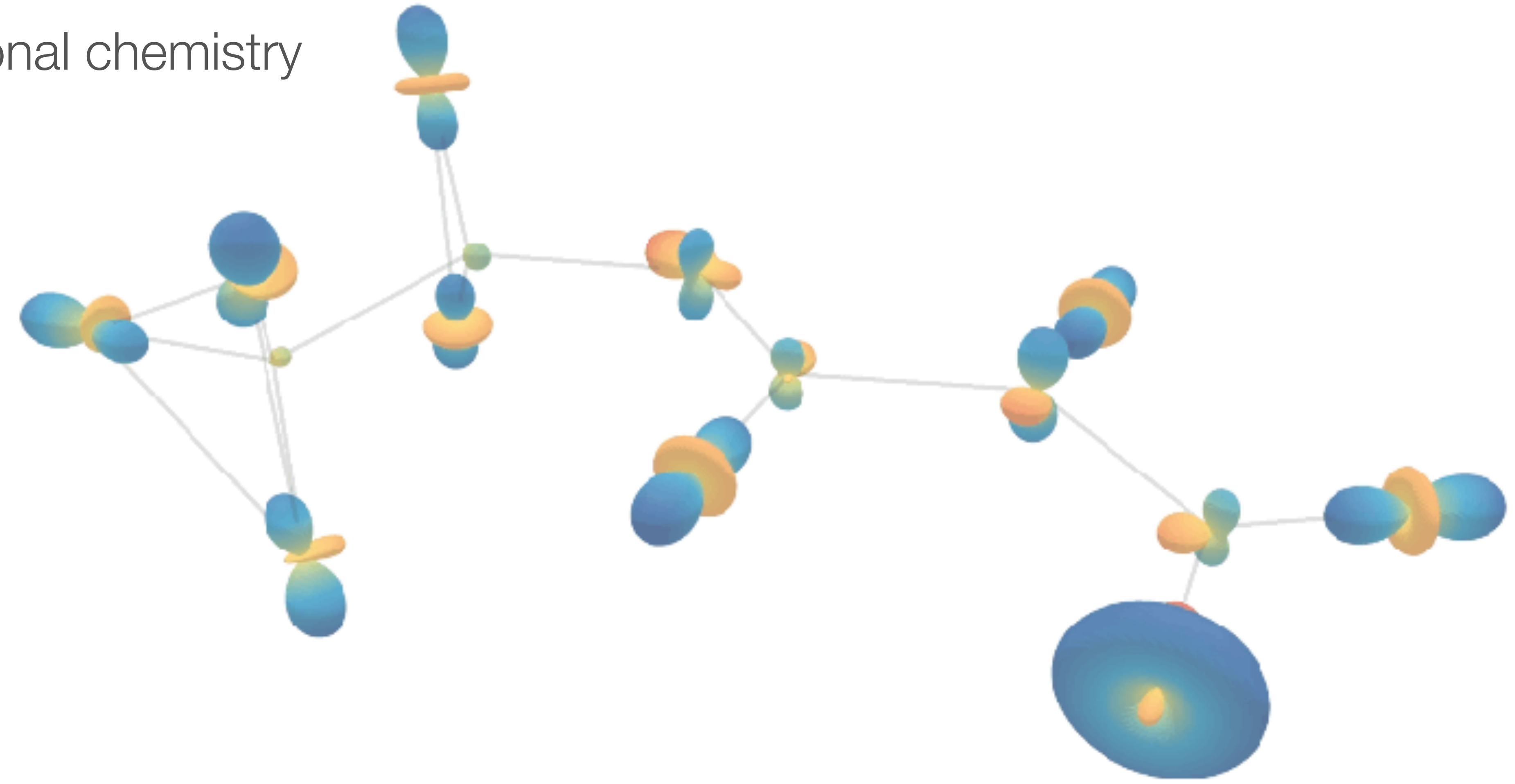
Erik J Bekkers, Amsterdam Machine Learning Lab (AMLAB), University of Amsterdam
e.j.bekkers@uva.nl

Medical image analysis





Computational chemistry



Physics

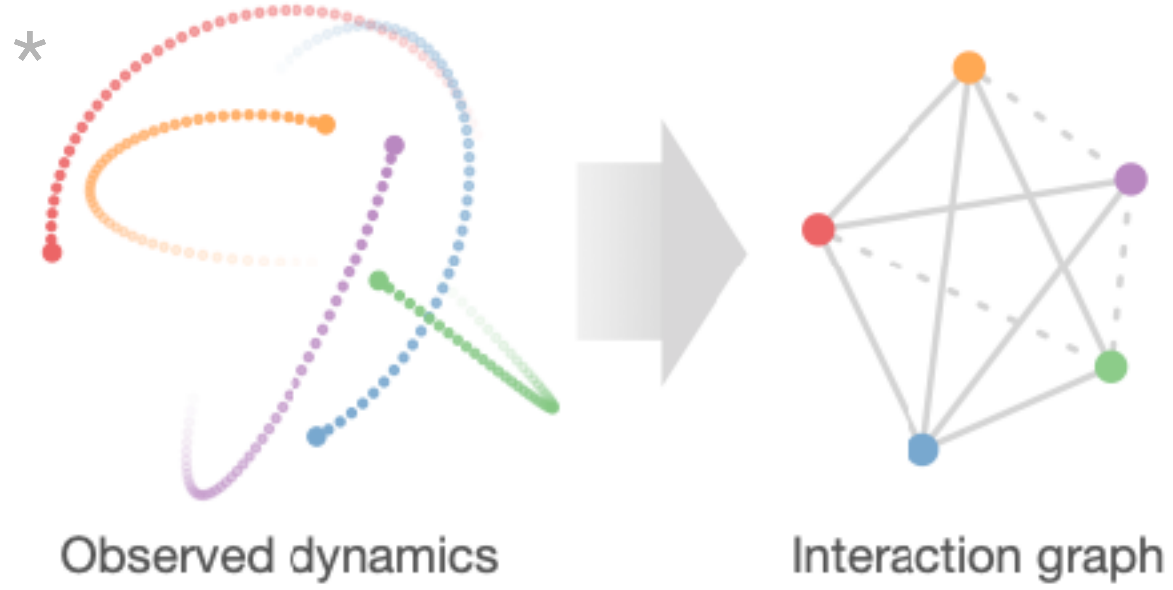


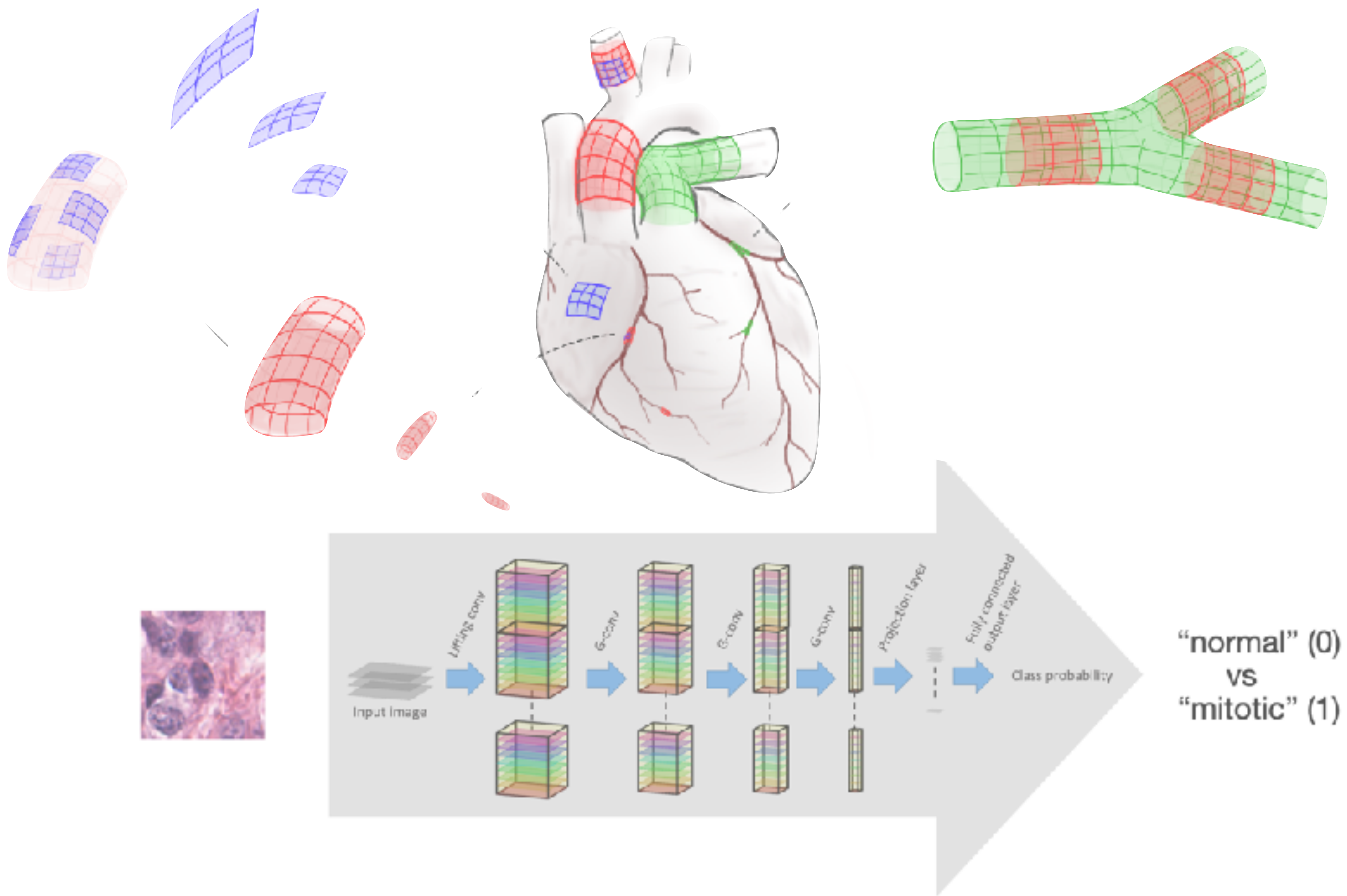
*Figure from Kipf et al 2018

UNIVERSITY OF AMSTERDAM

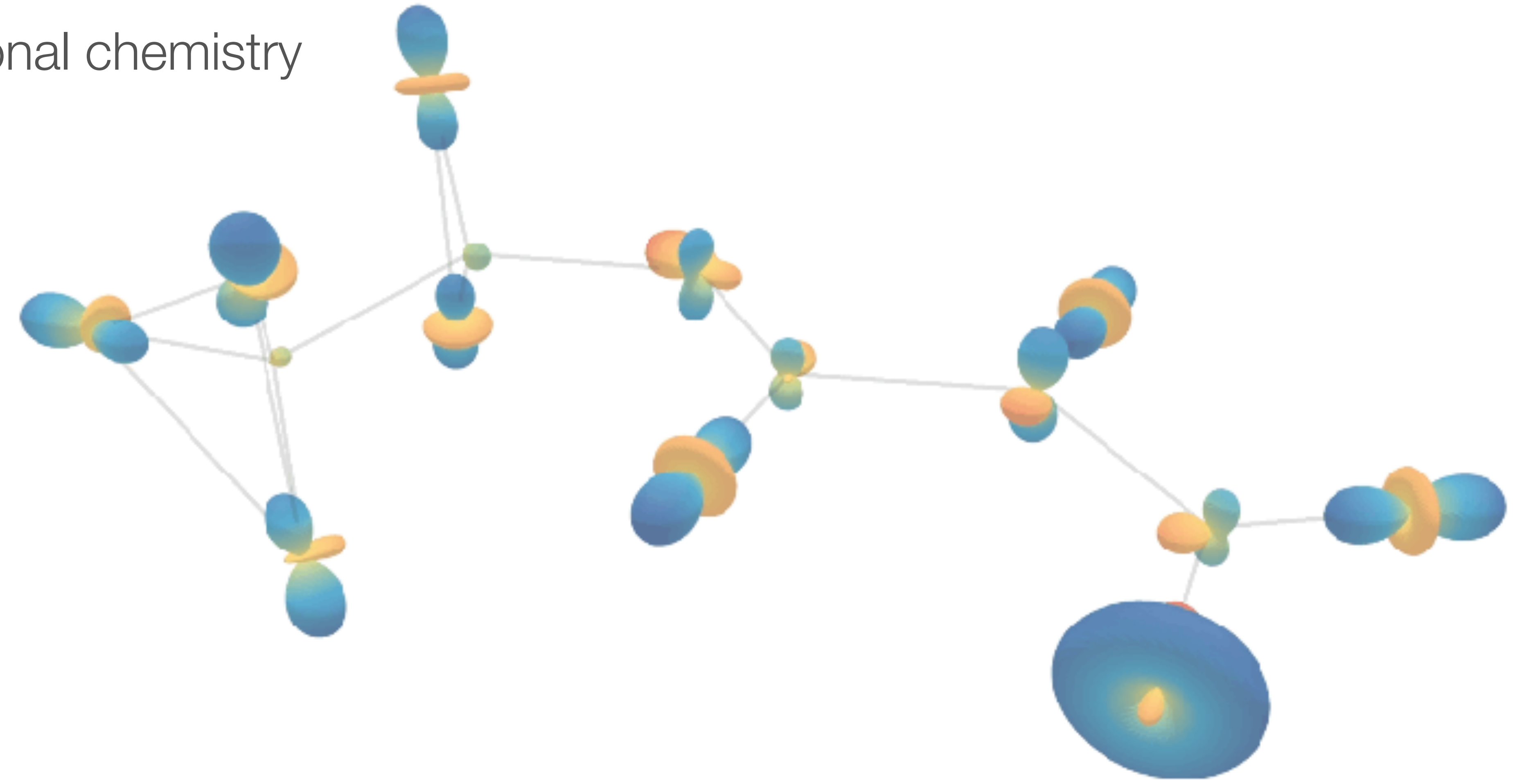
Group equivariant deep learning and non-linear convolutions

Erik J Bekkers, Amsterdam Machine Learning Lab (AMLAB), University of Amsterdam
e.j.bekkers@uva.nl

Medical image analysis



Computational chemistry



Physics

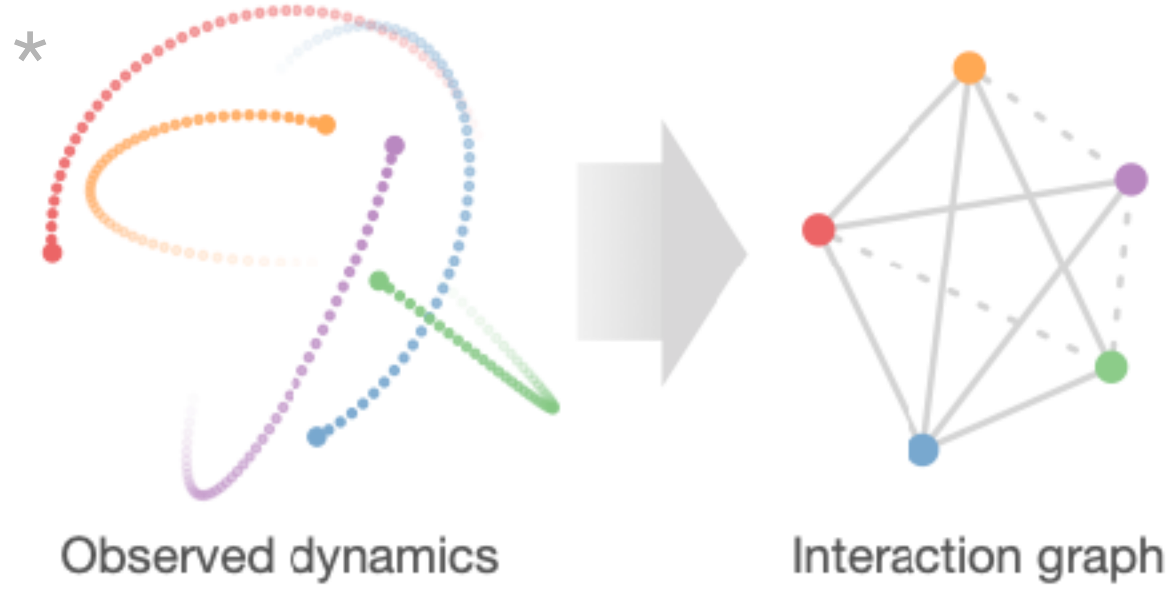
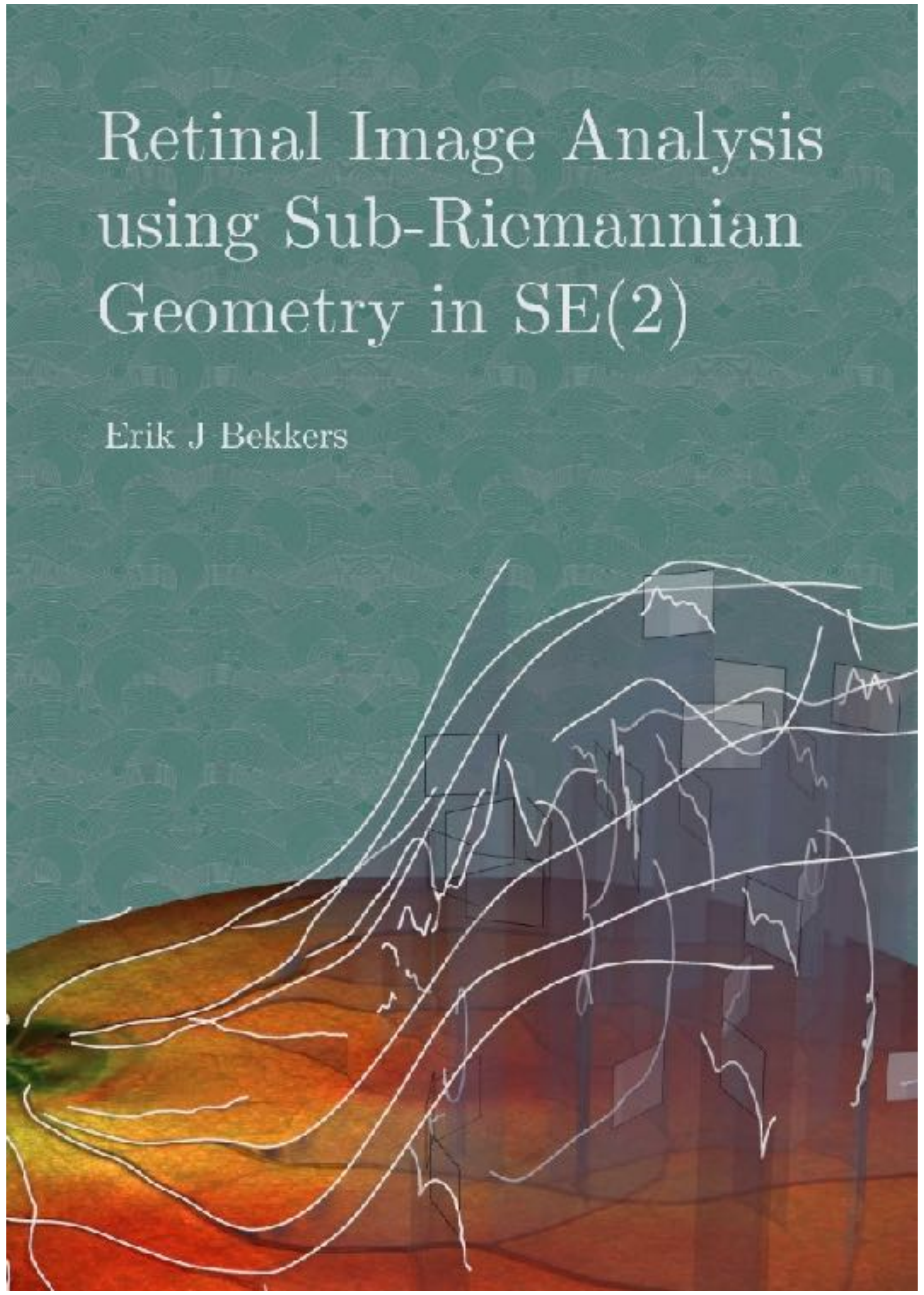


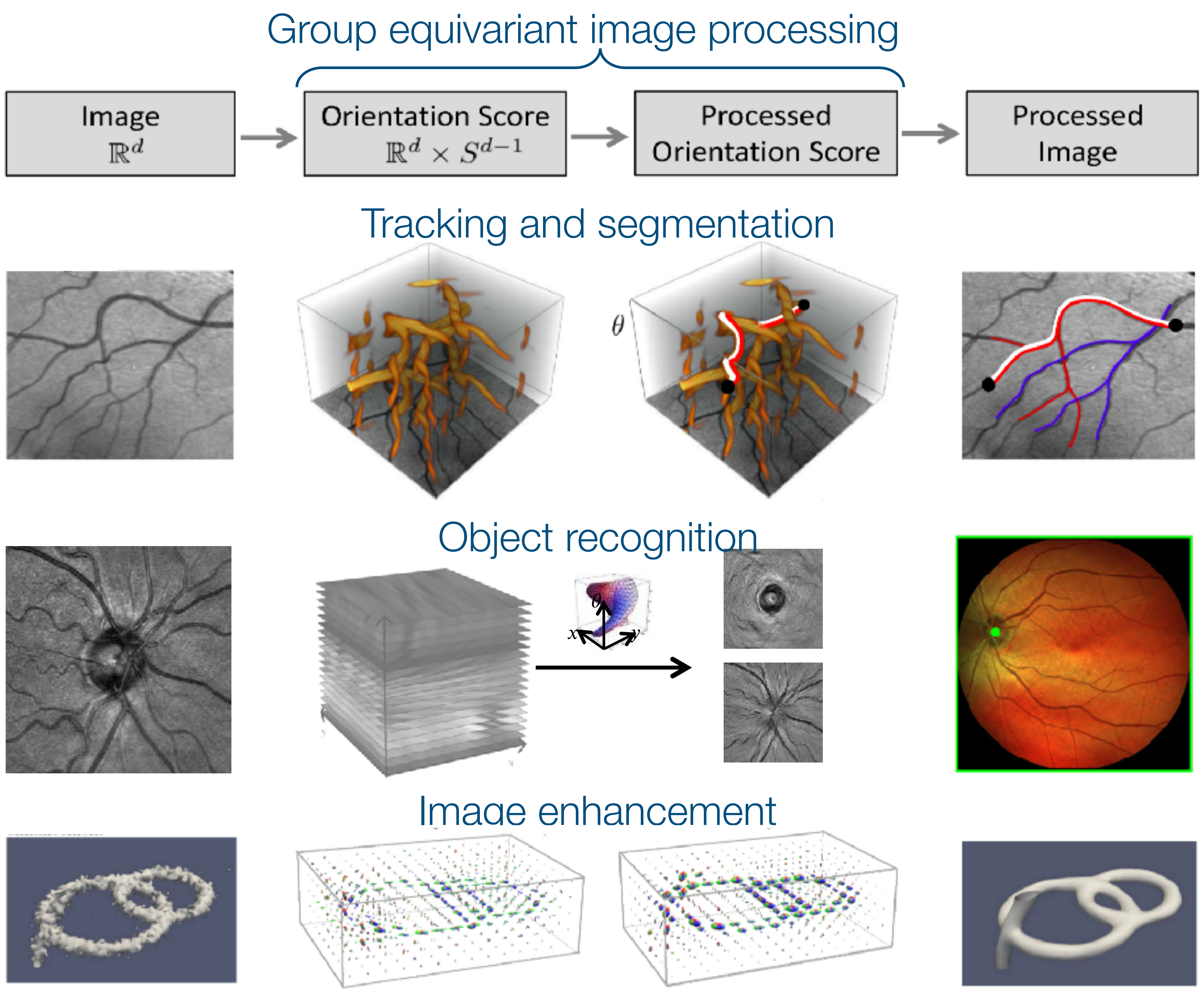
*Figure from Kipf et al 2018

Background

Previously

PhD @ TU/e (BME) w Remco Duits + Bart ter Haar Romeny
Post-doc @ TU/e (CASA)





Background

Currently

Assistant Prof @ UvA (AMLab)

EXPLOITING REDUNDANCY: SEPARABLE GROUP CONVOLUTIONAL NETWORKS ON LIE GROUPS

David M. Knigge
Universiteit van Amsterdam
The Netherlands
d.m.knigge@uva.nl

David W. Romero
Vrije Universiteit Amsterdam
The Netherlands
d.w.romeroguzman@vu.nl

Erik J. Bekkers
Universiteit van Amsterdam
The Netherlands
e.j.bekkers@uva.nl

Wavelet Networks: Scale Equivariant Learning From Raw Waveforms

David W. Romero
Vrije Universiteit Amsterdam
The Netherlands
d.w.romeroguzman@vu.nl

Erik J. Bekkers
Universiteit van Amsterdam
The Netherlands
e.j.bekkers@uva.nl

Jakub M. Tomczak
Vrije Universiteit Amsterdam
The Netherlands
j.m.tomczak@vu.nl

Mark Hoogendoorn
Vrije Universiteit Amsterdam
The Netherlands
m.hoogendoorn@vu.nl

Abstract

Inducing symmetry equivariance in deep neural architectures has resolved into improved data efficiency and generalization. In this work, we utilize the concept of scale and translation equivariance to tackle the problem of learning on time-series from raw waveforms. As a result, we obtain representations that largely resemble those of the wavelet transform at the first layer, but that evolve into much more descriptive ones as a function of depth. Our empirical results support the suitability of our *Wavelet Networks* which with a simple architecture design perform consistently better than CNNs on raw waveforms and on par with spectrogram-based methods.

B-SPLINE CNNS ON LIE GROUPS

Erik J. Bekkers
Centre for Analysis and Scientific Computing
Department of Applied Mathematics and Computer Science
Eindhoven University of Technology, Eindhoven, the Netherlands
e.j.bekkers@tue.nl

ABSTRACT

Group convolutional neural networks (G-CNNs) can be used to improve classi-

FLEXCONV: CONTINUOUS KERNEL CONVOLUTIONS WITH DIFFERENTIABLE KERNEL SIZES

David W. Romero[*,1], Robert-Jan Bruintjes[*,2], Erik J. Bekkers[3], Jakub M. Tomczak[1], Mark Hoogendoorn[1], Jan C. van Gemert[2]
[1] Vrije Universiteit Amsterdam [2] Delft University of Technology [3] University of Amsterdam
The Netherlands
d.w.romeroguzman@vu.nl, r.bruintjes@tudelft.nl

When design
of the convol
different ker
is unfeasible
during traini
bandwidth.
can describe
operation wi
size can be l
cies without
sequential d
competitive
FlexNets car
To avoid ali

CheblieNet: Invariant Spectral Graph NNs Turned Equivariant by Riemannian Geometry on Lie Groups

Hugo Aguettaz
ETHZ, Zürich, Switzerland
aguettaz@isi.ee.ethz.ch

Erik J Bekkers
UvA, Amsterdam, Netherlands
e.j.bekkers@uva.nl

Michaël Defferrard
EPFL, Lausanne, Switzerland
michael.defferrard@epfl.ch

Attentive Group Equivariant Convolutional Networks

David W. Romero[1] Erik J. Bekkers[2] Jakub M. Tomczak[1] Mark Hoogendoorn[1]

Abstract

Although group convolutional networks are able to learn powerful representations based on symmetry patterns, they lack explicit means to learn meaningful relationships among them (e.g., relative positions and poses). In this paper, we present *attentive group equivariant convolutions*, a generalization of the group convolution, in which attention is applied during the course of convo-

Figure 1. Meaningful relationships among object symme
Though every figure is composed by the same elements,
the outermost examples resemble faces. The relative posi
orientations and scales of elements in the innermost exampl

GEOMETRIC AND PHYSICAL QUANTITIES IMPROVE E(3) EQUIVARIANT MESSAGE PASSING

Johannes Brandstetter*
University of Amsterdam
Johannes Kepler University Linz
brandstetter@ml.jku.at

Rob Hesselink*
University of Amsterdam
r.d.hesselink@uva.nl

Elise van der Pol
UvA-Bosch DeltaLab
University of Amsterdam
e.e.vanderpol@uva.nl

Erik Bekkers
University of Amsterdam
e.j.bekkers@uva.nl

Max Welling
UvA-Bosch DeltaLab
University of Amsterdam
m.welling@uva.nl

ABSTRACT

Including covariant information, such as position, force, velocity or spin is important in many tasks in computational physics and chemistry. We introduce Steerable E(3) Equivariant Graph Neural Networks (SEGNNs) that generalise equivariant graph networks, such that node and edge attributes are not restricted to invariant scalars, but can contain covariant information, such as vectors or tensors. This model, composed of steerable MLPs, is able to incorporate geometric and physical information in both the message and update functions. Through the definition of steerable node attributes, the MLPs provide a new class of activation functions for general use with steerable feature fields. We discuss ours and related work through the lens of *equivariant non-linear convolutions*, which further allows us to

Background

Currently

Assistant Prof @ UvA (AMLab)

Exploiting Redundancy: Separable Group Convolutional Networks on Lie Groups

David M. Knigge Universiteit van Amsterdam The Netherlands d.m.knigge@uva.nl
David W. Romero Vrije Universiteit Amsterdam The Netherlands d.w.romeroguzman@vu.nl
Erik J. Bekkers Universiteit van Amsterdam The Netherlands e.j.bekkers@uva.nl

Wavelet Networks: Scale Equivariant Learning From Raw Waveforms

David W. Romero Vrije Universiteit Amsterdam The Netherlands d.w.romeroguzman@vu.nl
Erik J. Bekkers Universiteit van Amsterdam The Netherlands e.j.bekkers@uva.nl
Jakub M. Tomczak Vrije Universiteit Amsterdam The Netherlands j.m.tomczak@vu.nl
Mark Hoogendoorn Vrije Universiteit Amsterdam The Netherlands m.hoogendoorn@vu.nl

Abstract

Inducing symmetry equivariance in deep neural architectures has resolved into improved data efficiency and generalization. In this work, we utilize the concept of scale and translation equivariance to tackle the problem of learning on time-series from raw waveforms. As a result, we obtain representations that largely resemble those of the wavelet transform at the first layer, but that evolve into much more descriptive ones as a function of depth. Our empirical results support the suitability of our *Wavelet Networks* which with a simple architecture design perform consistently better than CNNs on raw waveforms and on par with spectrogram-based methods.

B-Spline CNNs on Lie Groups

Erik J. Bekkers
Centre for Analysis and Scientific Computing
Department of Applied Mathematics and Computer Science
Eindhoven University of Technology, Eindhoven, the Netherlands
e.j.bekkers@tue.nl

Abstract

Group convolutional neural networks (G-CNNs) can be used to improve classi-

FlexConv: Continuous Kernel Convolutions with Differentiable Kernel Sizes

David W. Romero[*,1], **Robert-Jan Bruintjes**[*,2], **Erik J. Bekkers**[3], **Jakub M. Tomczak**[1], **Mark Hoogendoorn**[1], **Jan C. van Gemert**[2]
[1] Vrije Universiteit Amsterdam [2] Delft University of Technology [3] University of Amsterdam
The Netherlands
d.w.romeroguzman@vu.nl, r.bruintjes@tudelft.nl

When design... of the convol... different ker... is unfeasible... during traini... bandwidth. ... can describe... operation wi... size can be l... cies without... sequential d... competitive... FlexNets ca... To avoid ali...

Attentive Group Equivariant Convolutional Networks

David W. Romero[1] **Erik J. Bekkers**[2] **Jakub M. Tomczak**[1] **Mark Hoogendoorn**[1]

Abstract

Although group convolutional networks are able to learn powerful representations based on symmetry patterns, they lack explicit means to learn meaningful relationships among them (e.g., relative positions and poses). In this paper, we present *attentive group equivariant convolutions*, a generalization of the group convolution, in which attention is applied during the course of convo-

Figure 1. Meaningful relationships among object symm... Though every figure is composed by the same elements, ... the outermost examples resemble faces. The relative posi... orientations and scales of elements in the innermost exampl...

ChebLieNet: Invariant Spectral Graph NNs Turned Equivariant by Riemannian Geometry on Lie Groups

Hugo Aguettaz ETHZ, Zürich, Switzerland aguettaz@isi.ee.ethz.ch
Erik J Bekkers UvA, Amsterdam, Netherlands e.j.bekkers@uva.nl
Michaël Defferrard EPFL, Lausanne, Switzerland michael.defferrard@epfl.ch

Geometric and Physical Quantities Improve E(3) Equivariant Message Passing

Johannes Brandstetter[*] University of Amsterdam Johannes Kepler University Linz brandstetter@ml.jku.at
Rob Hesselink[*] University of Amsterdam r.d.hesselink@uva.nl
Elise van der Pol UvA-Bosch DeltaLab University of Amsterdam e.e.vanderpol@uva.nl
Erik Bekkers University of Amsterdam e.j.bekkers@uva.nl
Max Welling UvA-Bosch DeltaLab University of Amsterdam m.welling@uva.nl

Abstract

Including covariant information, such as position, force, velocity or spin is important in many tasks in computational physics and chemistry. We introduce Steerable E(3) Equivariant Graph Neural Networks (SEGNNs) that generalise equivariant graph networks, such that node and edge attributes are not restricted to invariant scalars, but can contain covariant information, such as vectors or tensors. This model, composed of steerable MLPs, is able to incorporate geometric and physical information in both the message and update functions. Through the definition of steerable node attributes, the MLPs provide a new class of activation functions for general use with steerable feature fields. We discuss ours and related work through the lens of *equivariant non-linear convolutions*, which further allows us to

Background

Currently

Assistant Prof @ UvA (AMLab)

Group equivariant deep learning

Geometric deep learning

Content

Part I: Introduction to group convolutions

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

Part II: Group convolutions are all you need

* Theorem: Equivariant linear operators are group convolutions
* Characterization of types of group equivariant layers

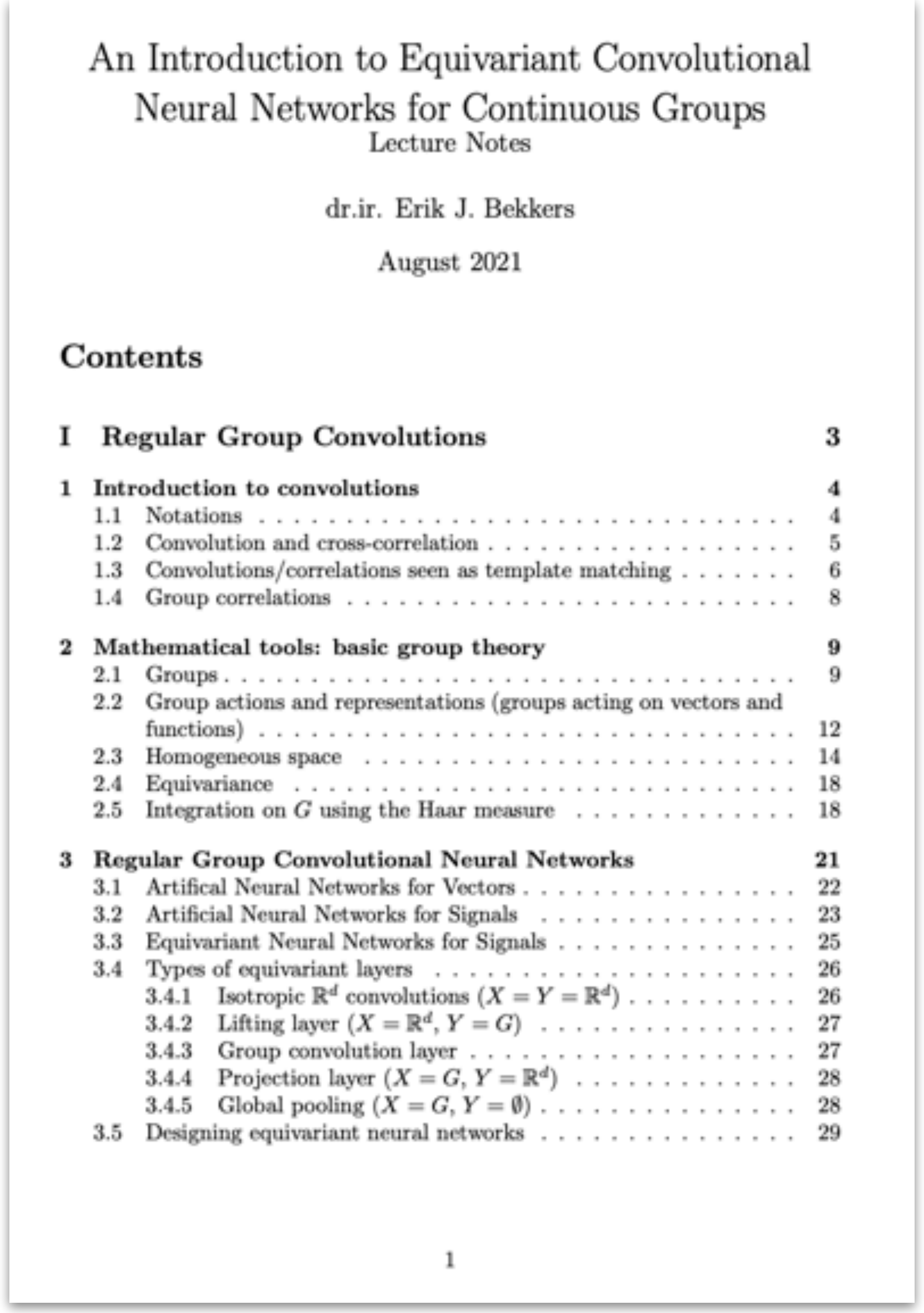
Part III: Steerable group convolutions for molecular data and the N-body problem

* Graph Neural networks
* Irreducible representations, steerable operators and vector spaces
* Steerable Graph NNs (Point Convolutions)

Lecture notes, slides and exercises available at

https://uvagedl.github.io

group **e**quivariant **d**eep **l**earning



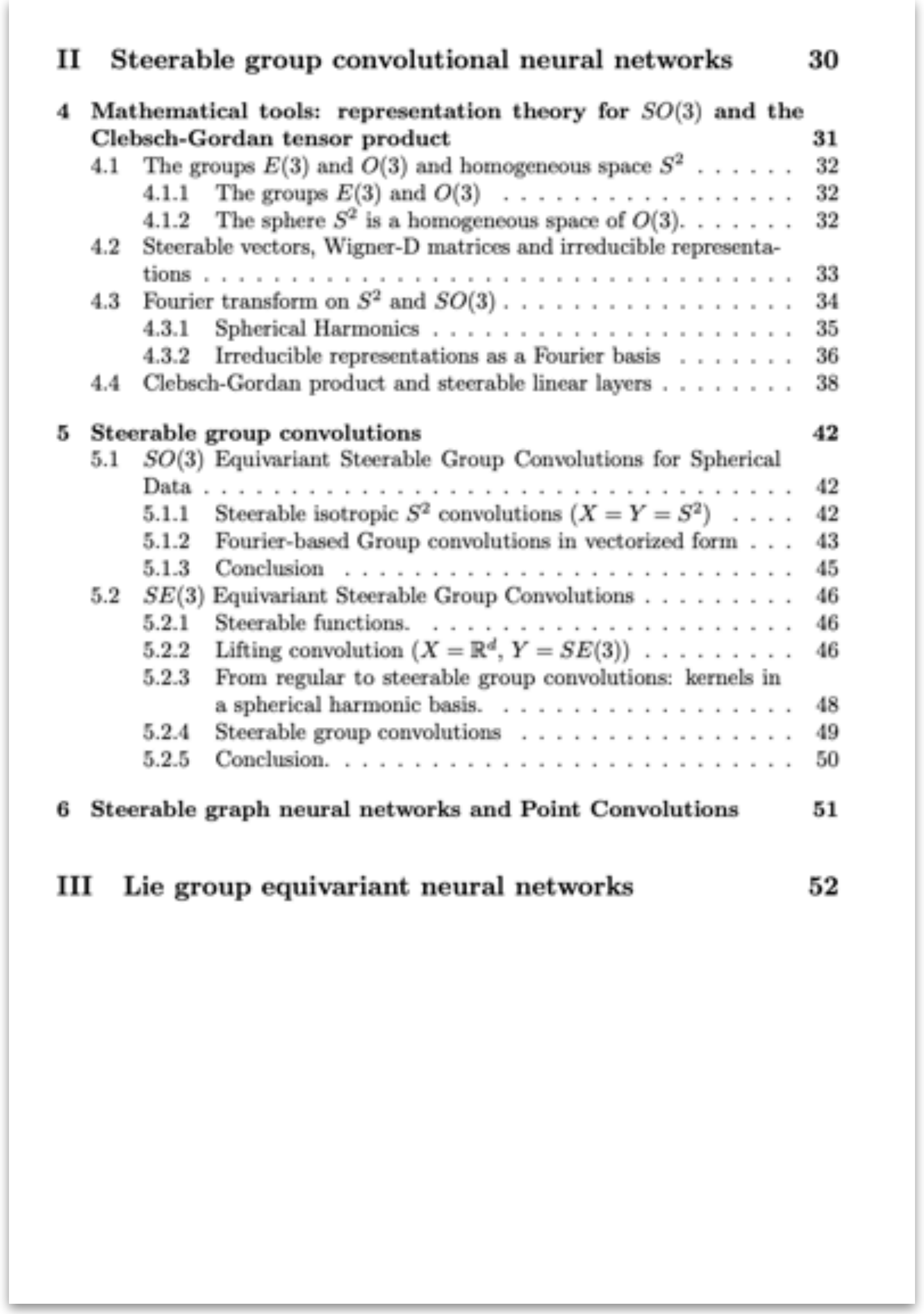
An Introduction to Equivariant Convolutional Neural Networks for Continuous Groups
Lecture Notes

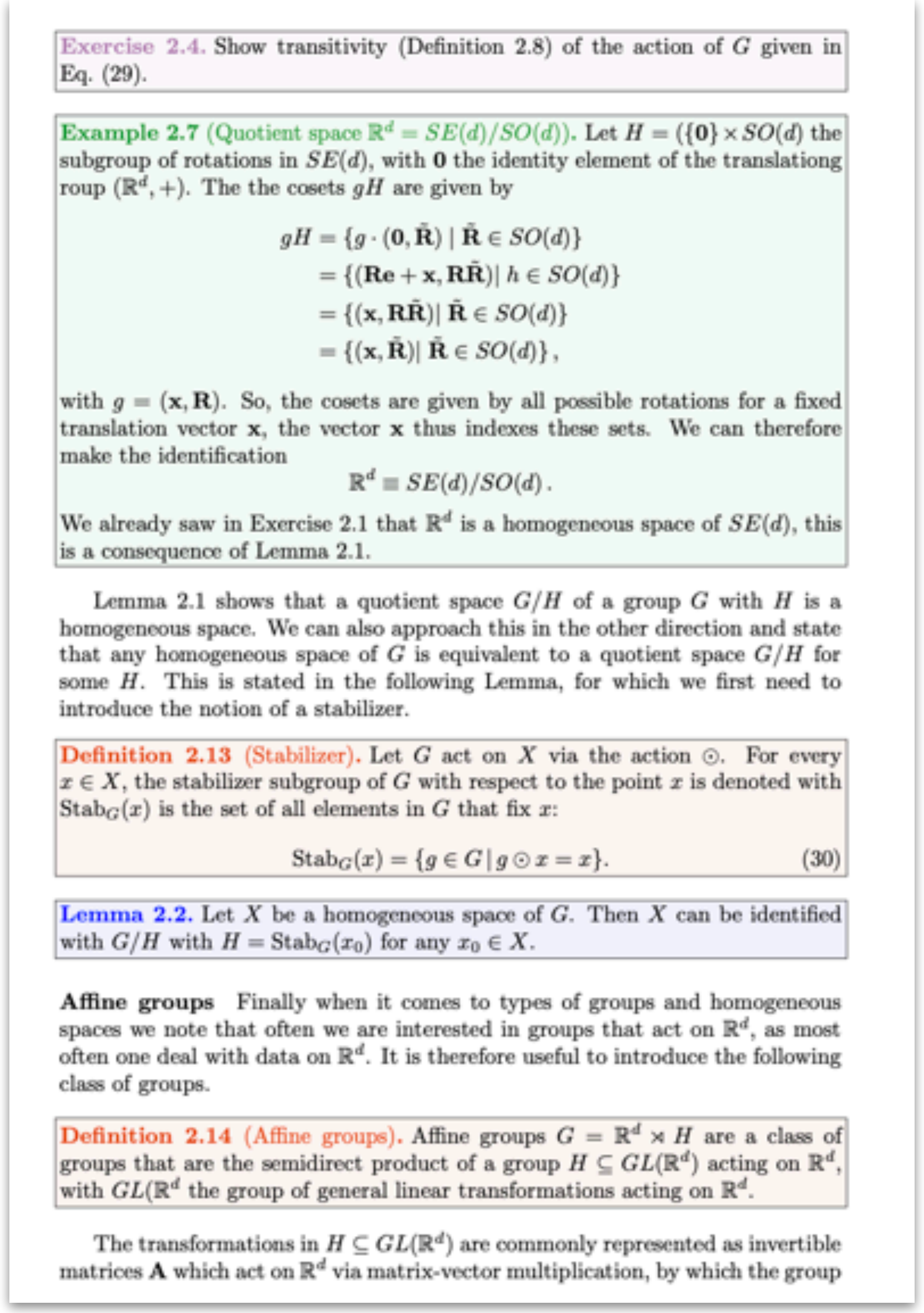
dr.ir. Erik J. Bekkers

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 translating 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

Content

Part I: Introduction to group convolutions

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

Part II: Group convolutions are all you need

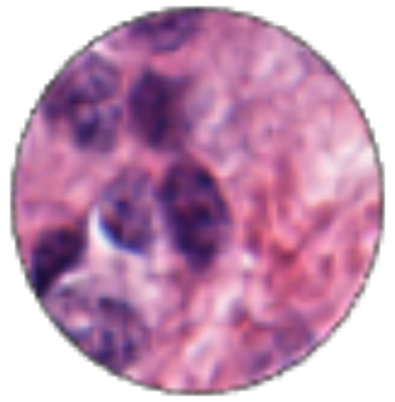
* Theorem: Equivariant linear operators are group convolutions
* Characterization of types of group equivariant layers

Part III: Steerable group convolutions for molecular data and the N-body problem

* Graph Neural networks
* Irreducible representations, steerable operators and vector spaces
* Steerable Graph NNs (Point Convolutions)

Motivation: Geometric Guarantees (invariance)

Example: Detection of pathological cells



Motivation: Geometric Guarantees (invariance)

Example: Detection of pathological cells

Healthy

?

Motivation: Geometric Guarantees (invariance)

Example: Detection of pathological cells

Healthy

?

Pathological

Motivation: Geometric Guarantees (invariance)

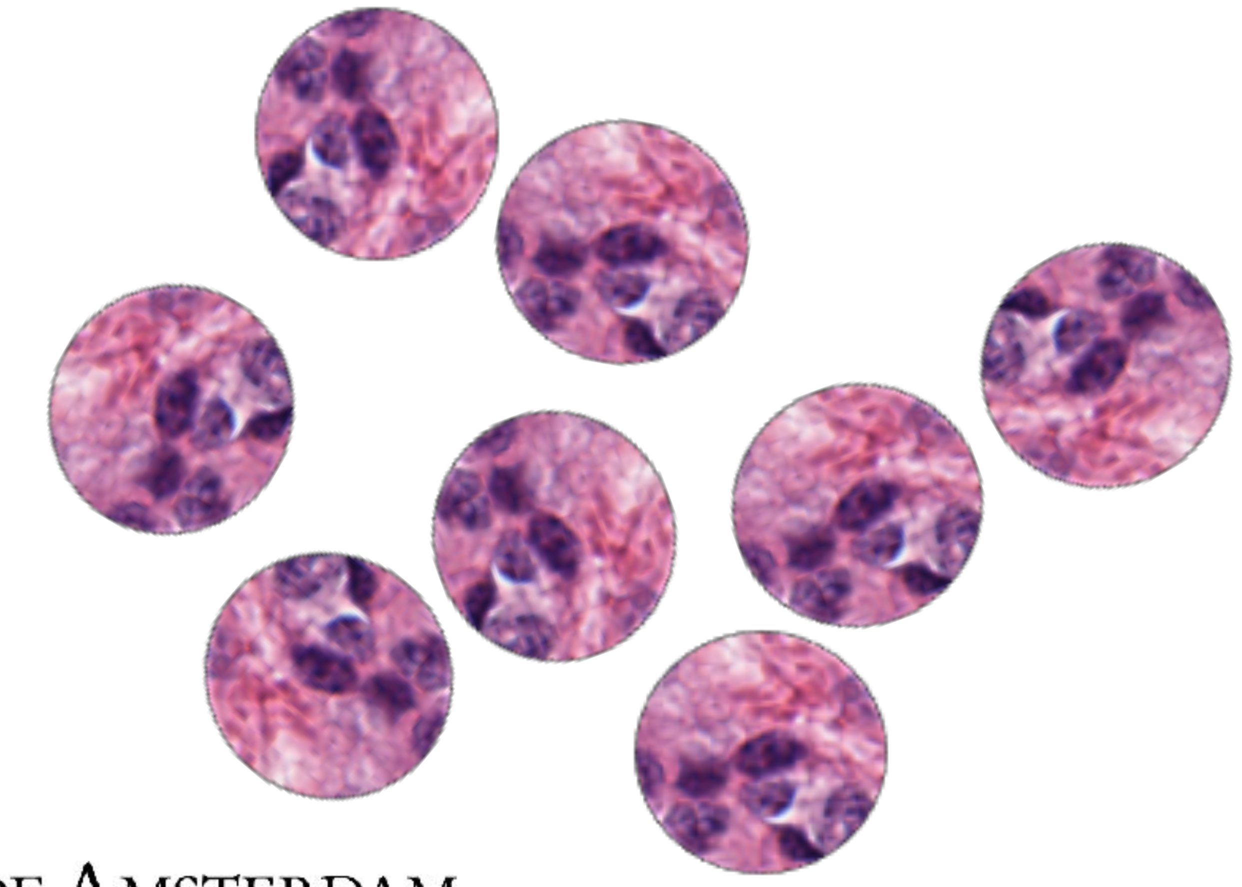
Example: Detection of pathological cells

Healthy

?

Pathological

Common approach: data-augmentation



Motivation: Geometric Guarantees (invariance)

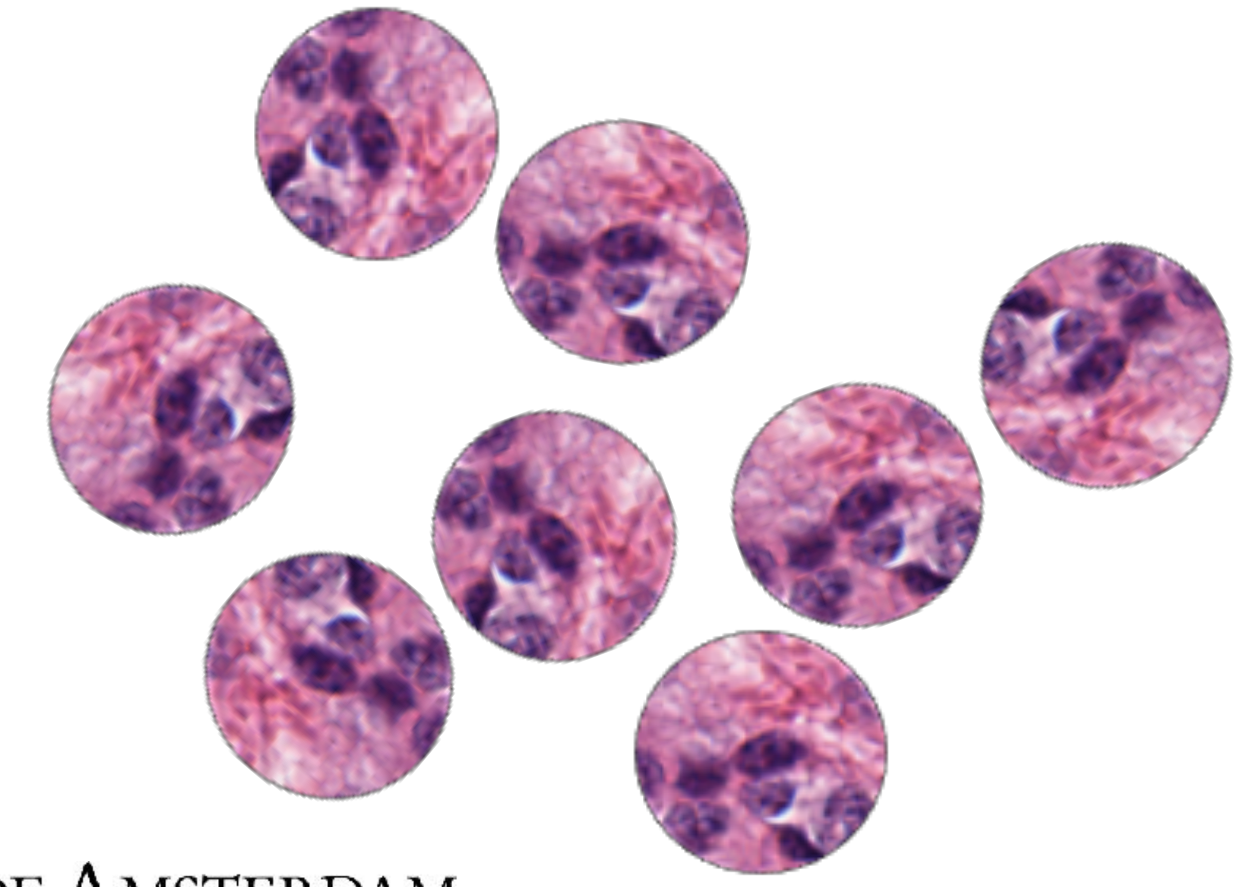
Example: Detection of pathological cells

Healthy

?

Pathological

Common approach: data-augmentation



Issues:
- Still no guarantee of invariance
- Valuable net capacity is spend on learning invariance
- Redundancy in feature repr.

Motivation: Geometric Guarantees (invariance)

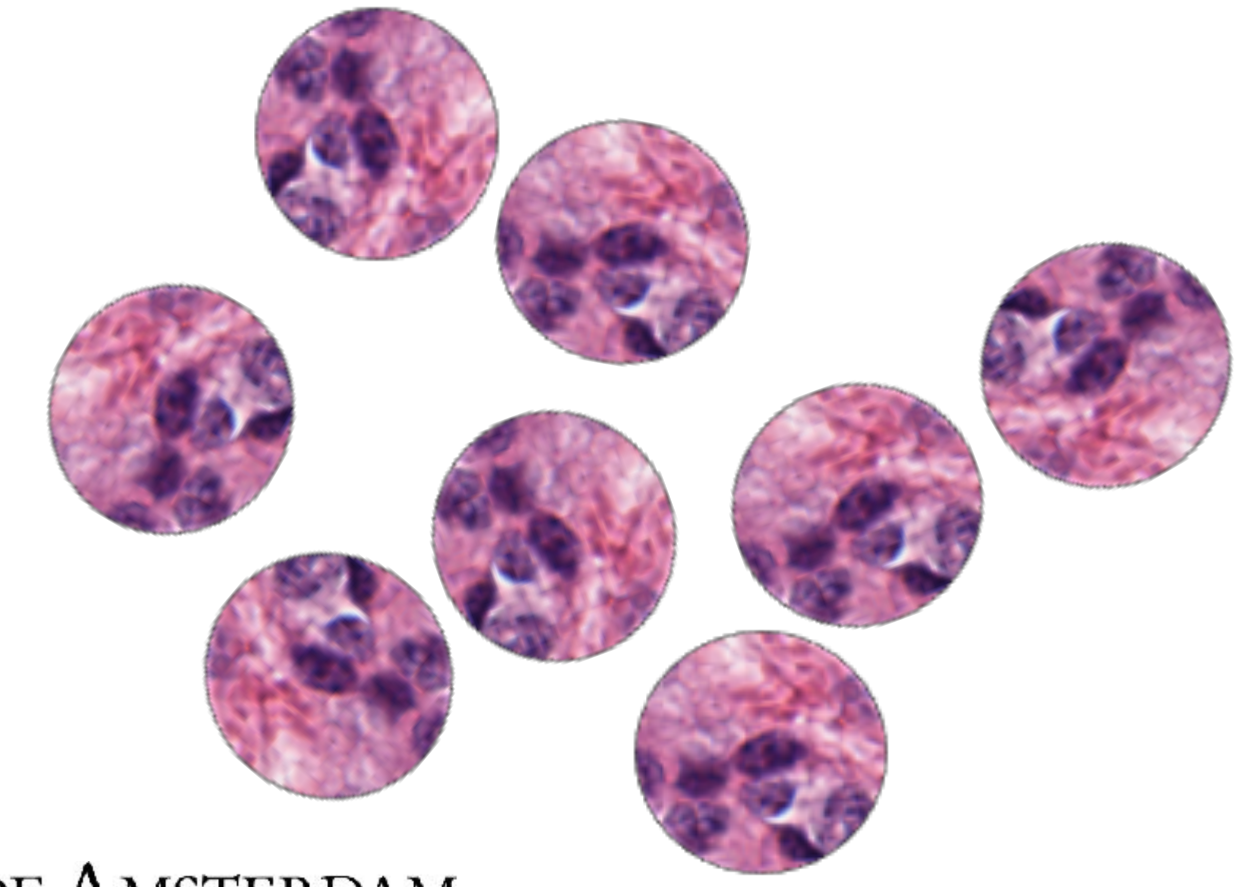
Example: Detection of pathological cells

Healthy

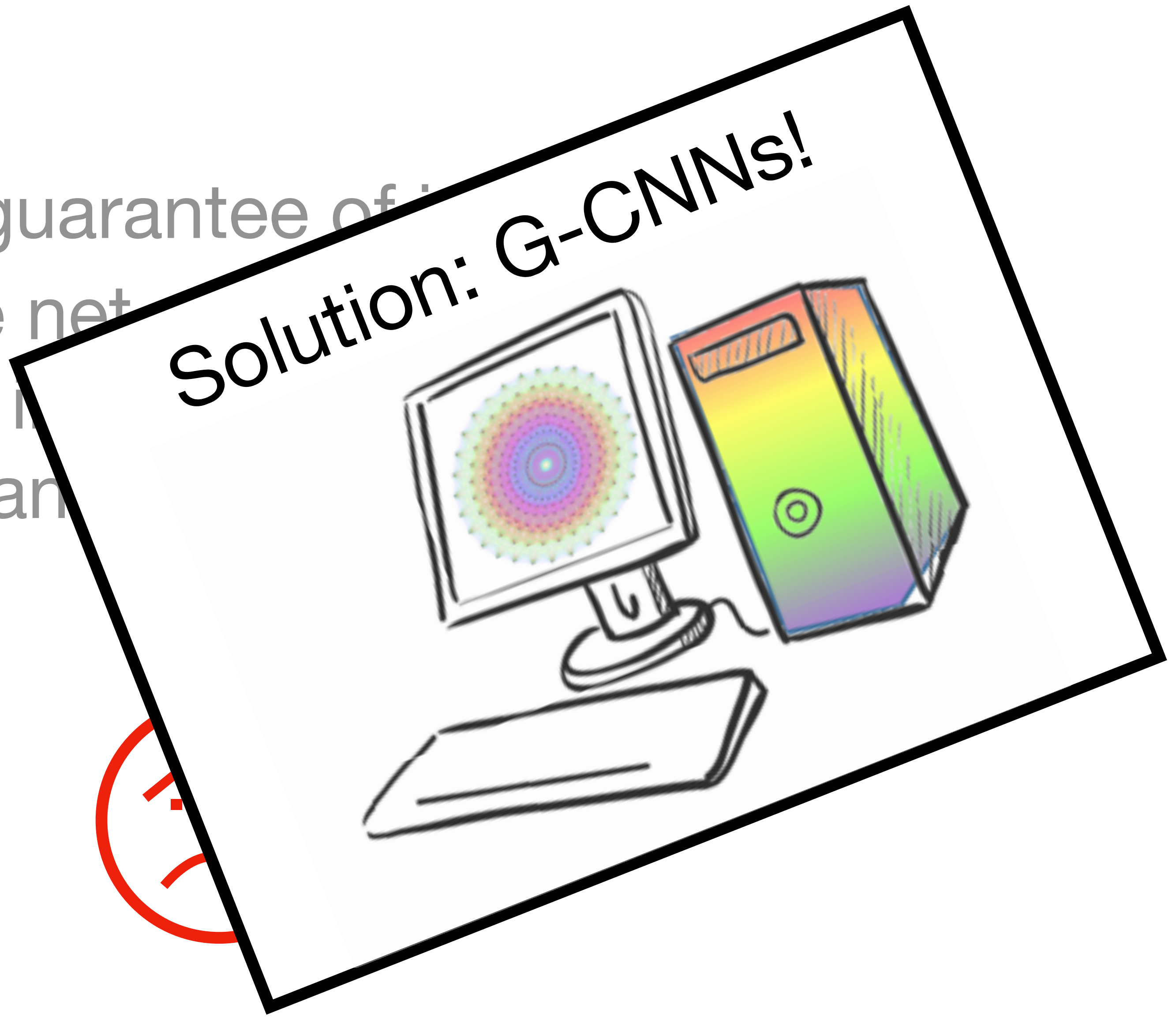
?

Pathological

Common approach: data-augmentation



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



Motivation: Geometric Guarantees (equivariance)

CNNs are translation equivariant

Via convolutions

Motivation: Geometric Guarantees (equivariance)

CNNs are translation equivariant

Via convolutions

Motivation: Geometric Guarantees (equivariance)

CNNs are not rotation equivariant

Via convolutions

Stabilized view

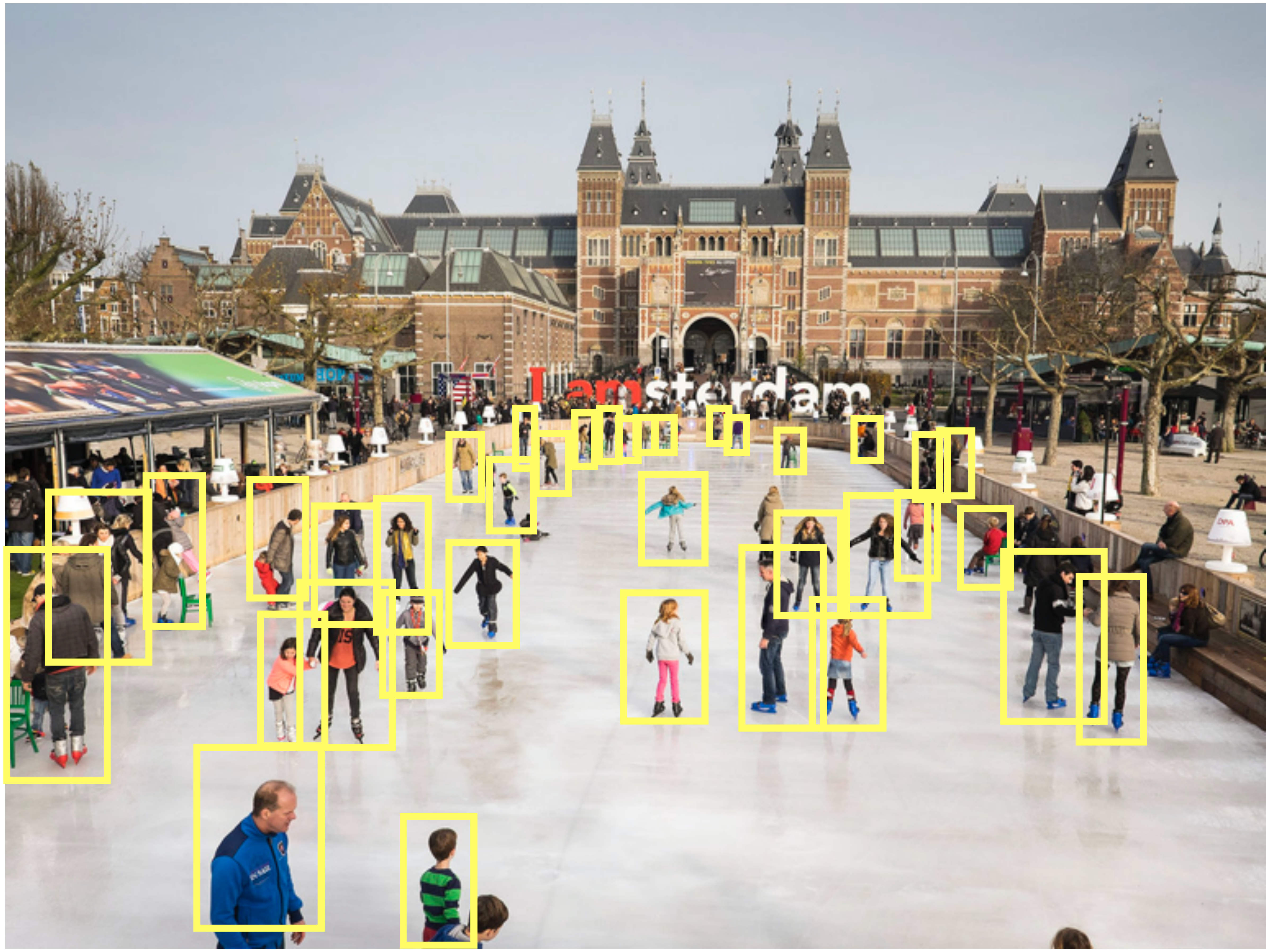
Motivation: Geometric Guarantees (equivariance)

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

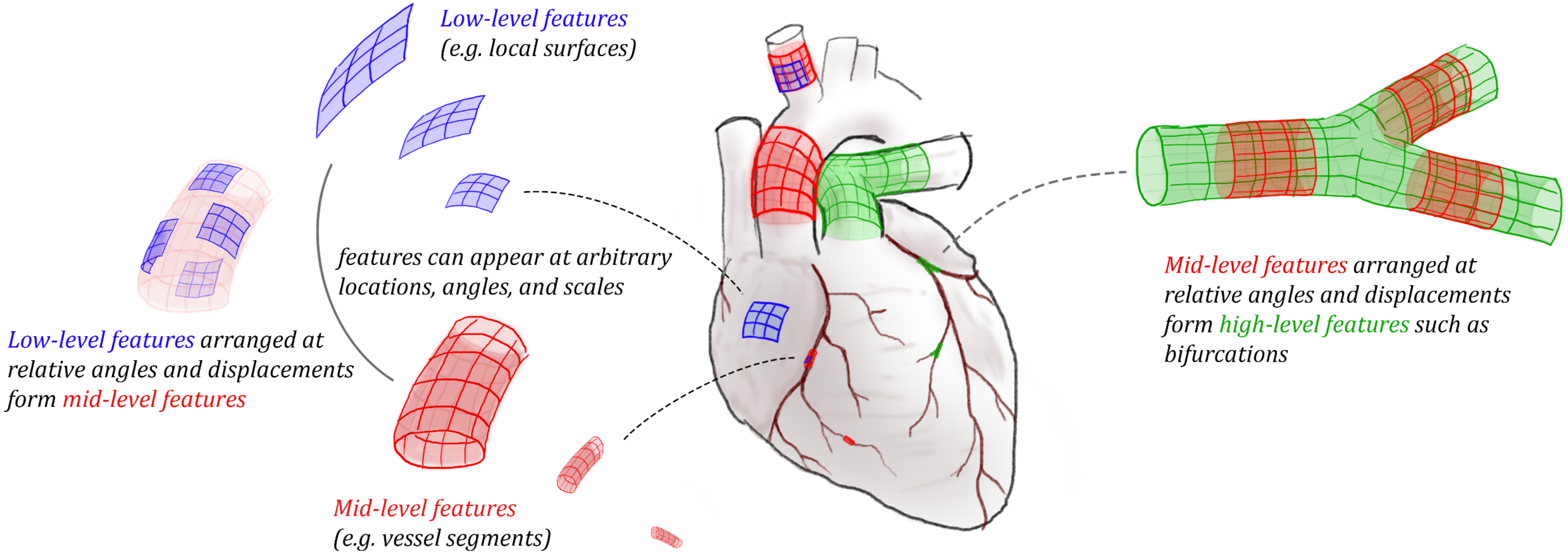
Group convolutions:

- Equivariance beyond translations
- Geometric guarantees
- Increased weight sharing

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

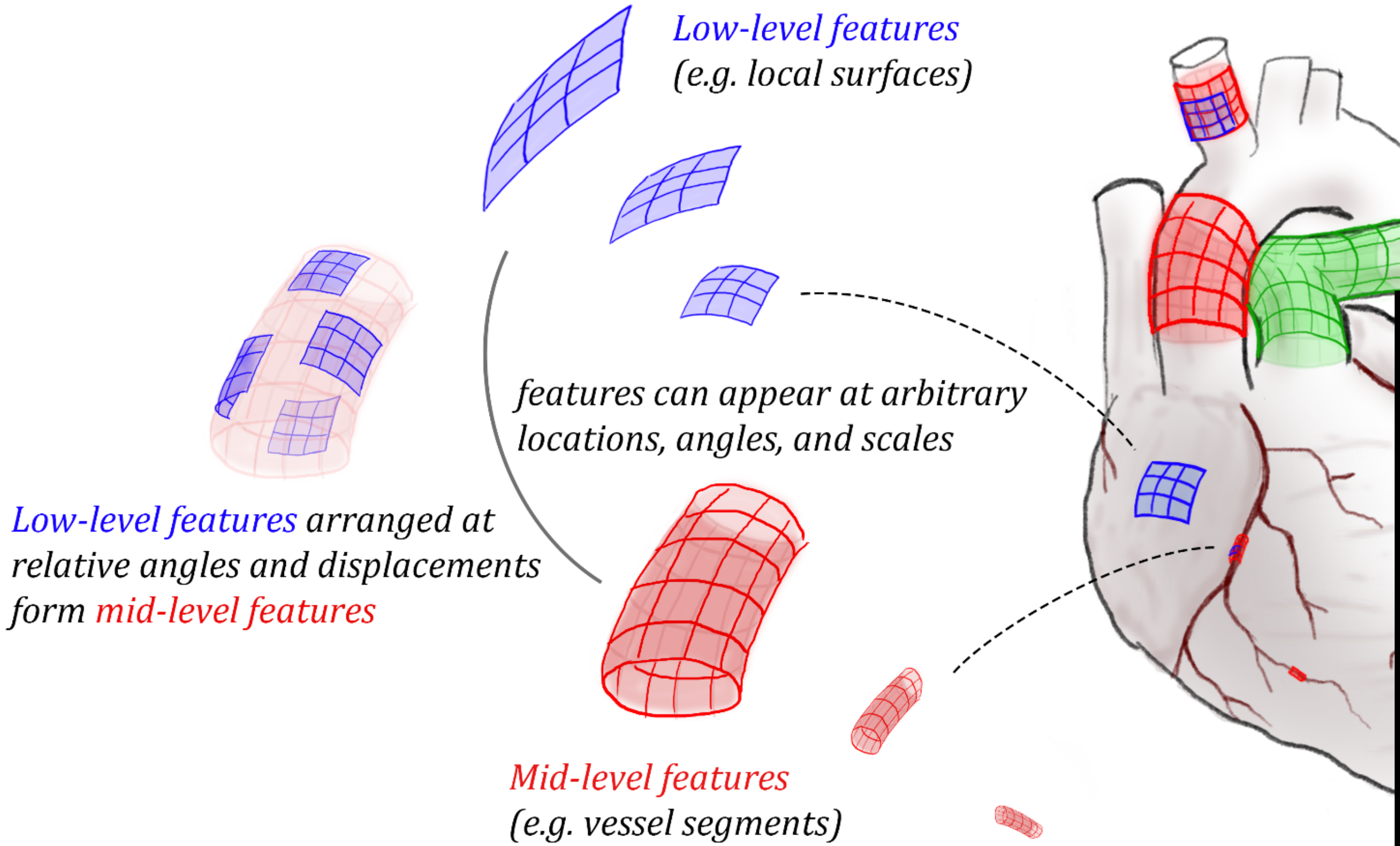
Motivation: Recognition by components

In a group theoretical setting



Motivation: Recognition by components

In a group theoretical setting



Rationale behind capsule networks

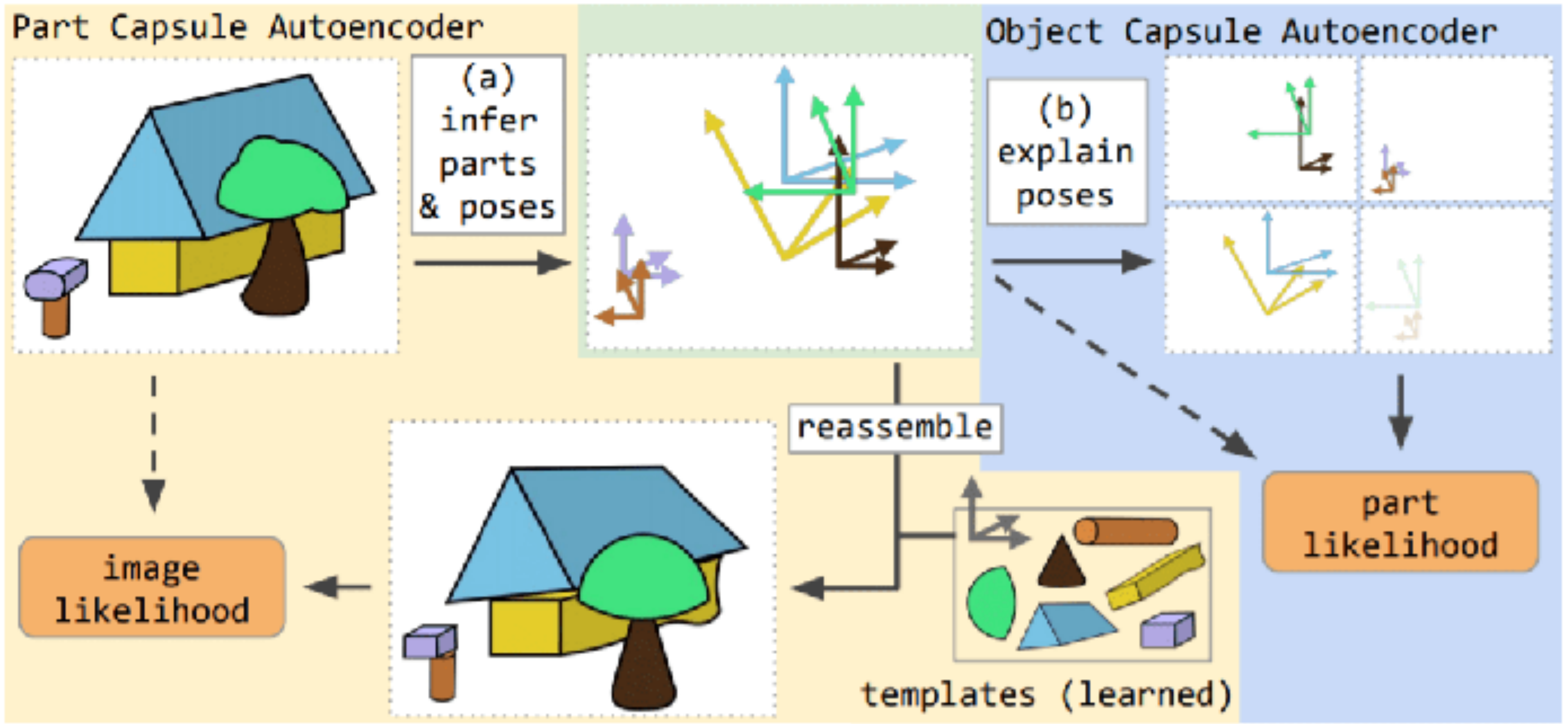


Fig from Kosiorek et al. 2019

Content

Part I: Introduction to group convolutions

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

Part II: Group convolutions are all you need

* Theorem: Equivariant linear operators are group convolutions
* Characterization of types of group equivariant layers

Part III: Steerable group convolutions for molecular data and the N-body problem

* Graph Neural networks
* Irreducible representations, steerable operators and vector spaces
* Steerable Graph NNs (Point Convolutions)

Groups

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:

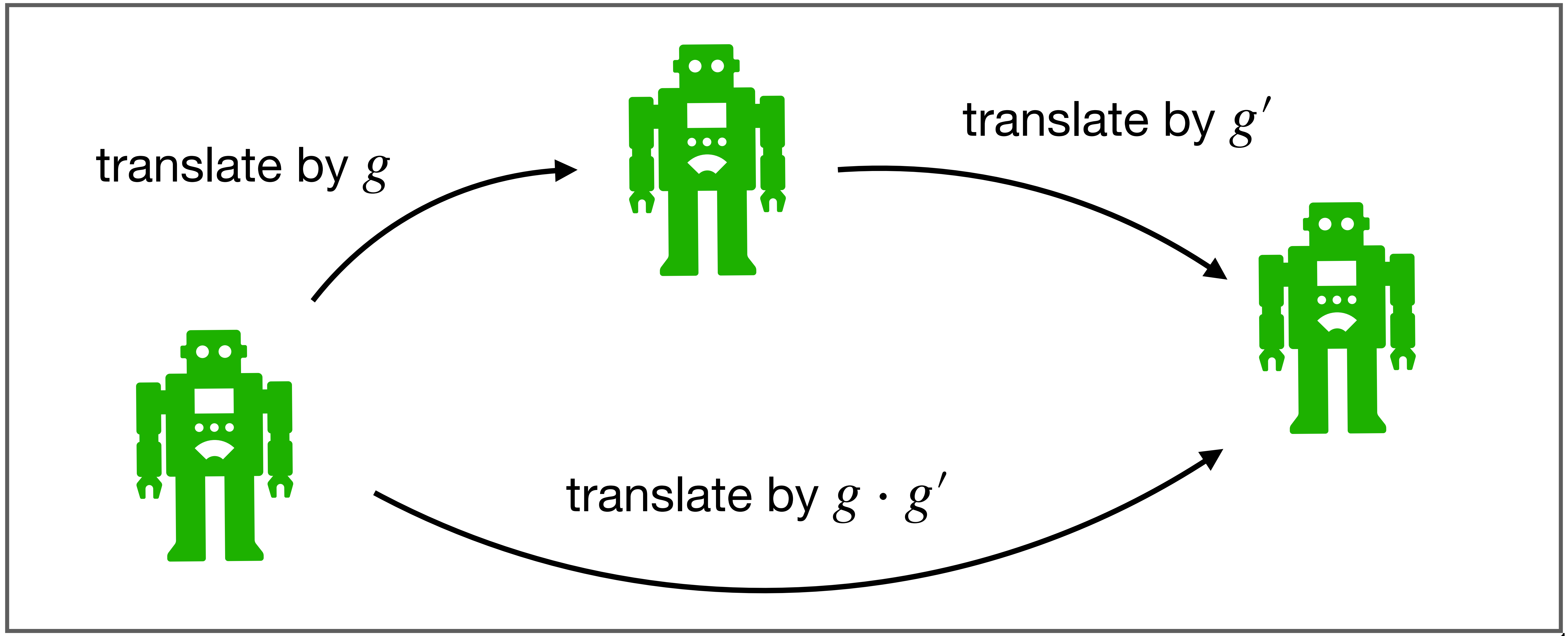
- **Closure**: Given two elements g and h of G, the product $g \cdot h$ is also in G.
- **Associativity**: For $g, h, i \in G$ the product $\cdot$ is associative, i.e., $g \cdot (h \cdot i) = (g \cdot h) \cdot i$.
- **Identity element**: There exists an identity element $e \in G$ such that $e \cdot g = g \cdot e = g$ 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 $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^2$.



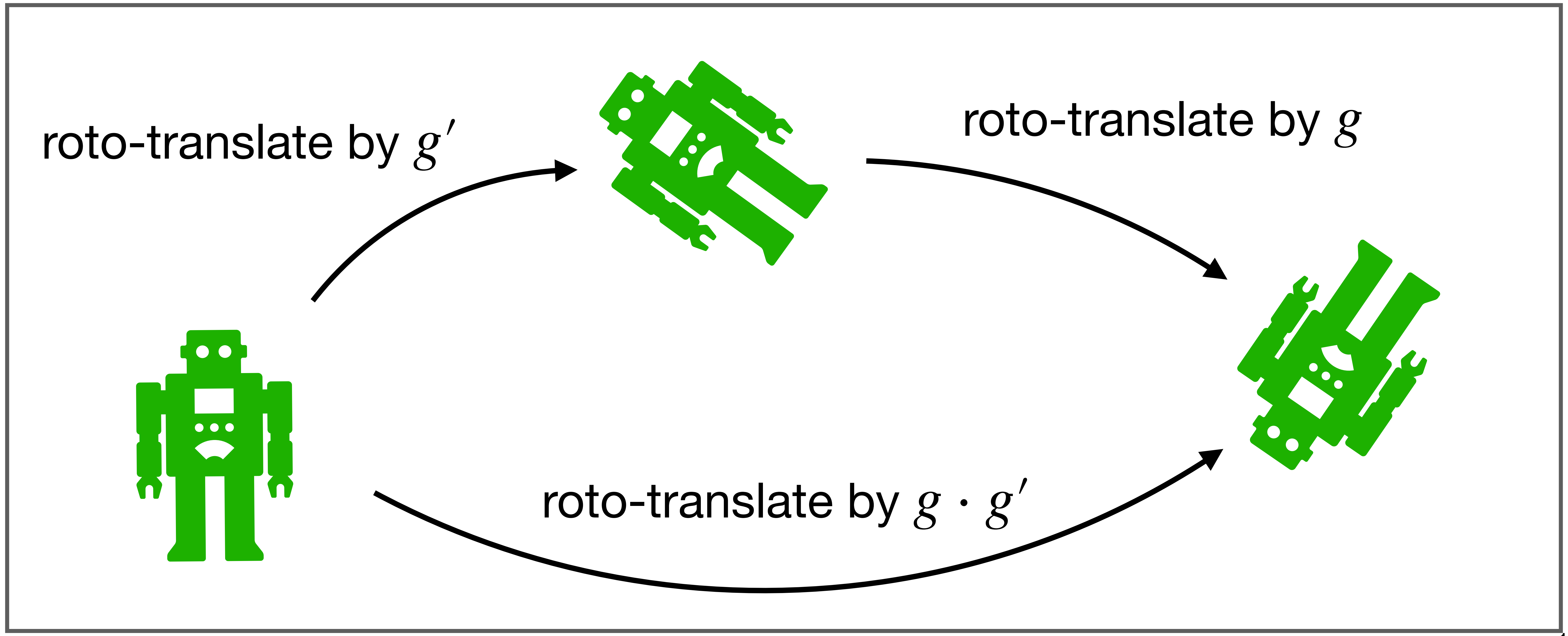
The roto-translation group $SE(2)$

*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 angles 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 $\mathbb{R}^2 \rtimes \mathbb{R}^+$

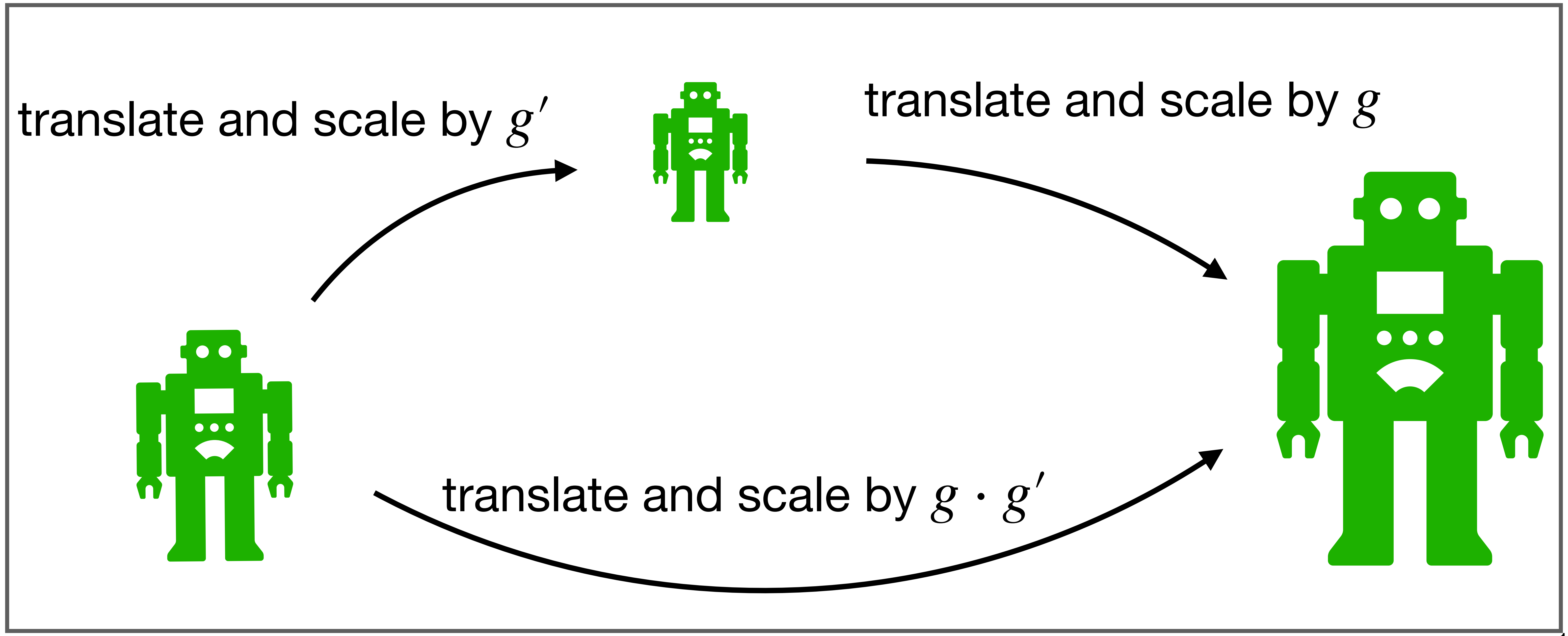
The scale-translation group of space $\mathbb{R}^2 \times \mathbb{R}^+$ of translations vectors in $\mathbb{R}^2$ and scale/dilation factors in $\mathbb{R}^+$, and is equipped with the group product and group inverse:

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

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

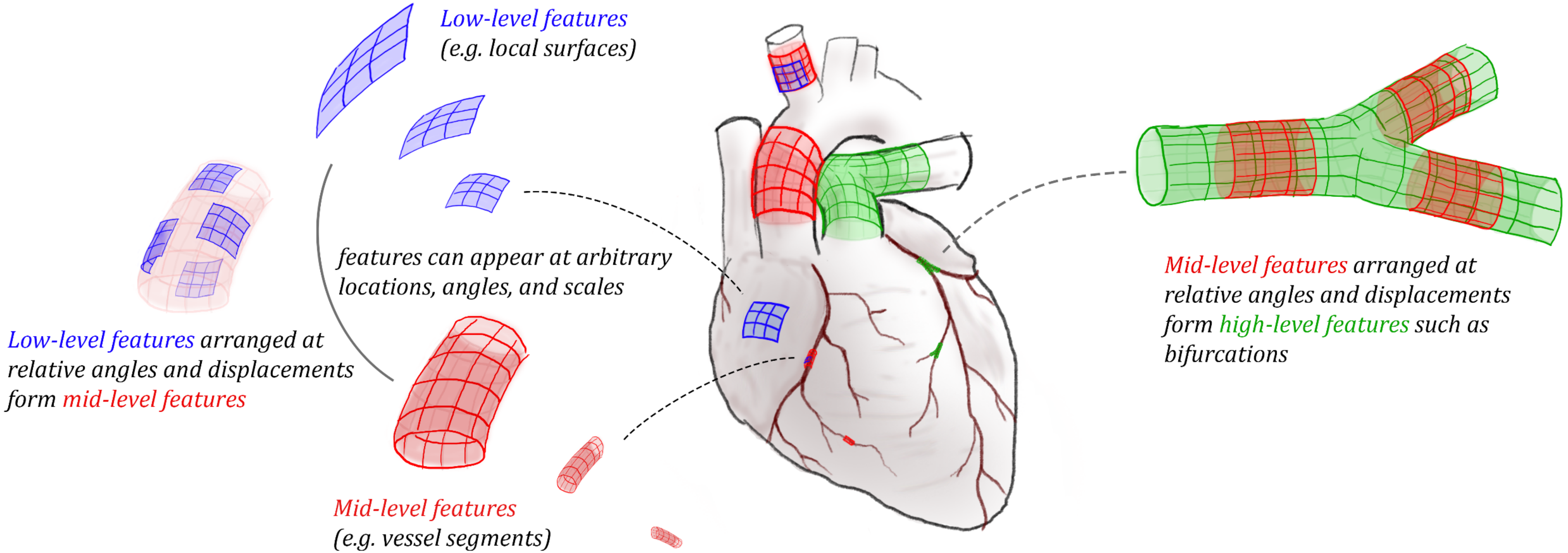
with $g \cdot g^{-1} = e = (\mathbf{0}, 1)$

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

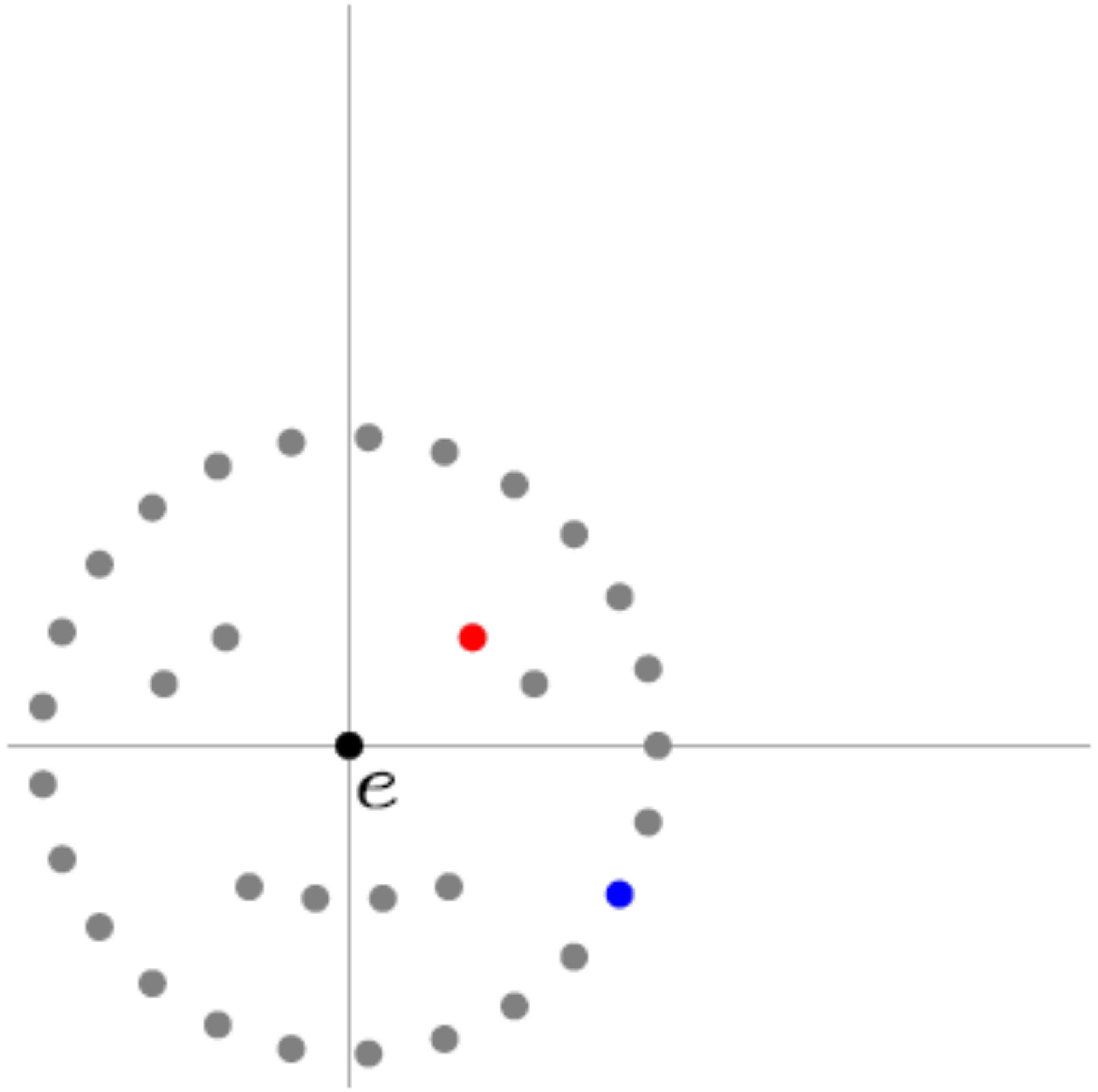


The translation group $(\mathbb{R}^2, +)$

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 $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^2$.

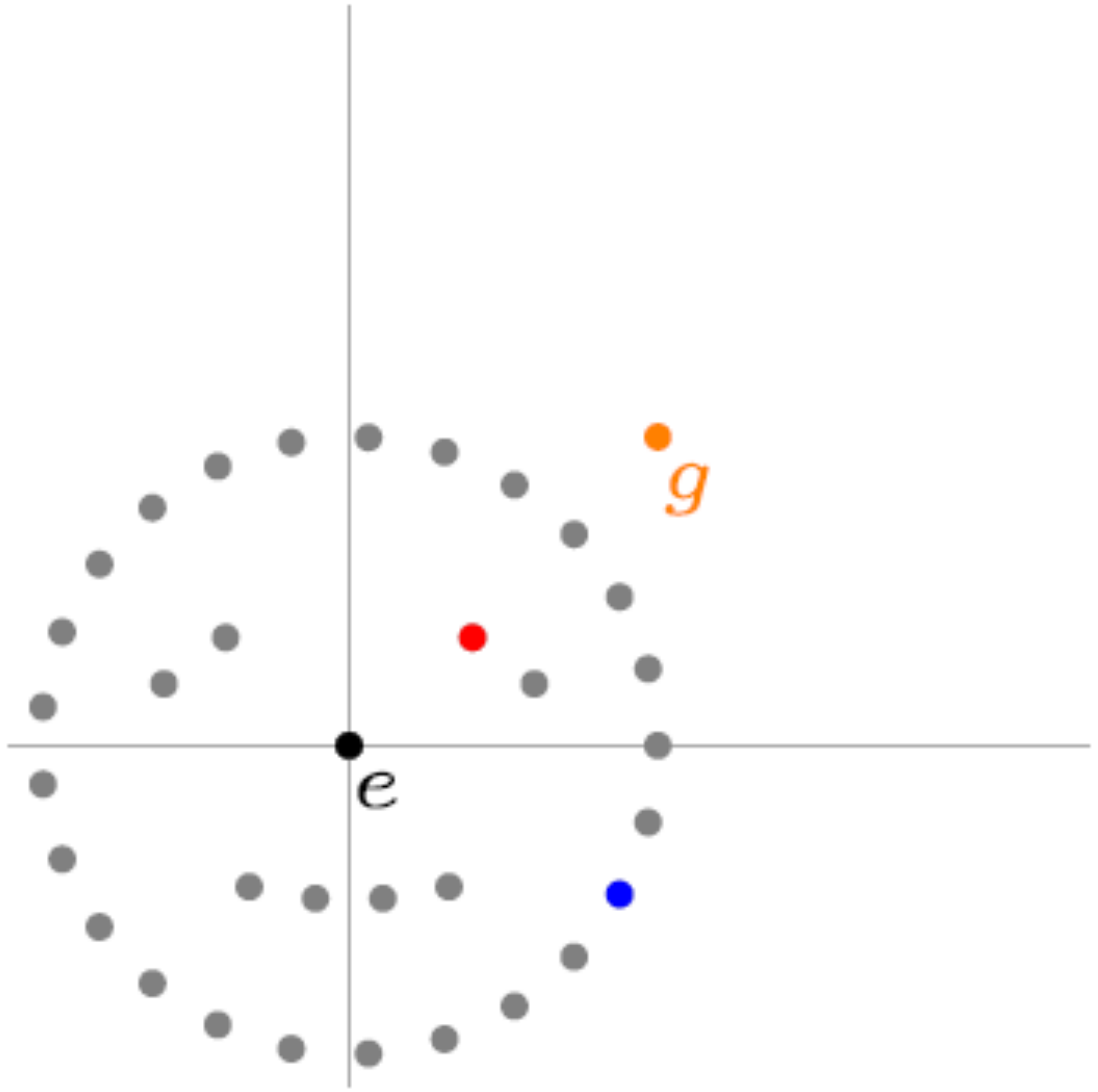


The translation group $(\mathbb{R}^2, +)$

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 $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^2$.

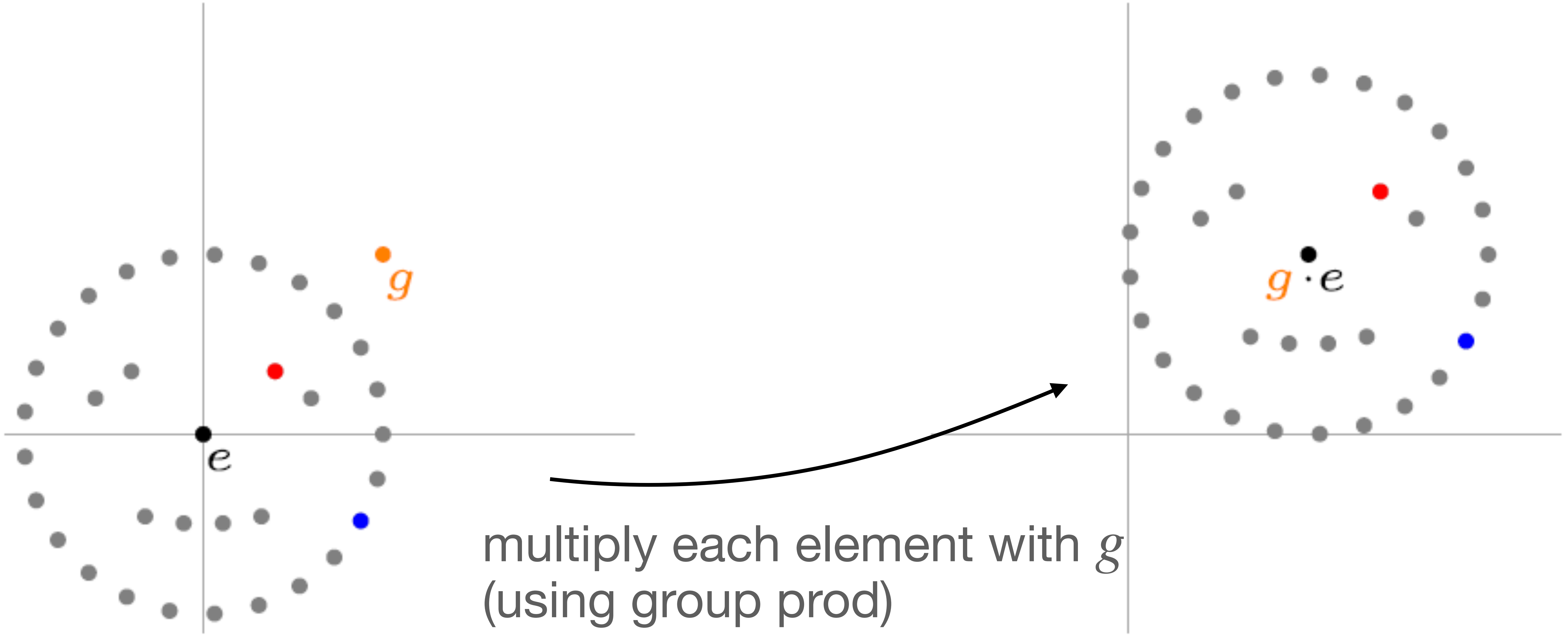


The translation group $(\mathbb{R}^2, +)$

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 $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^2$.



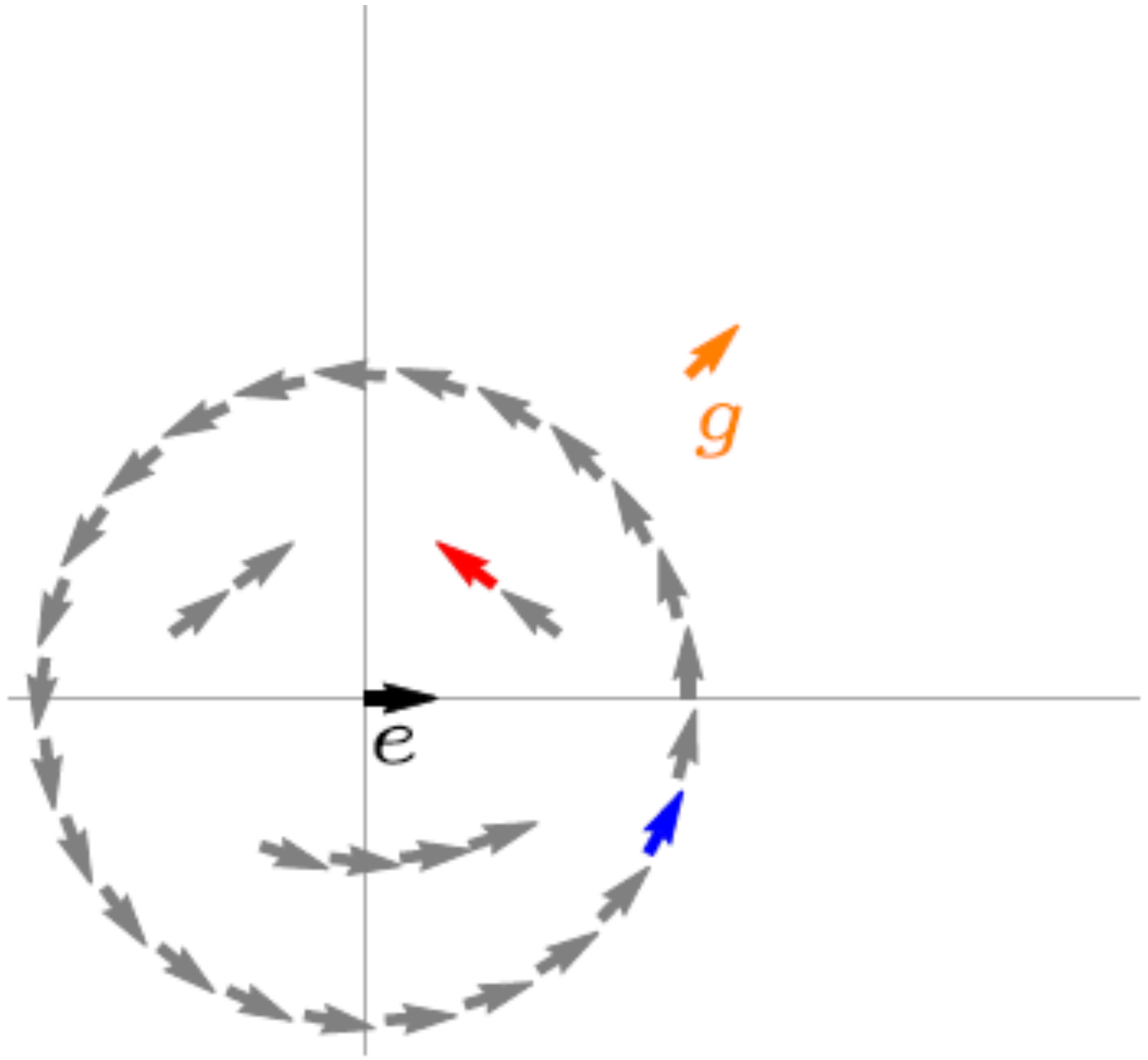
The roto-translation group $SE(2)$

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



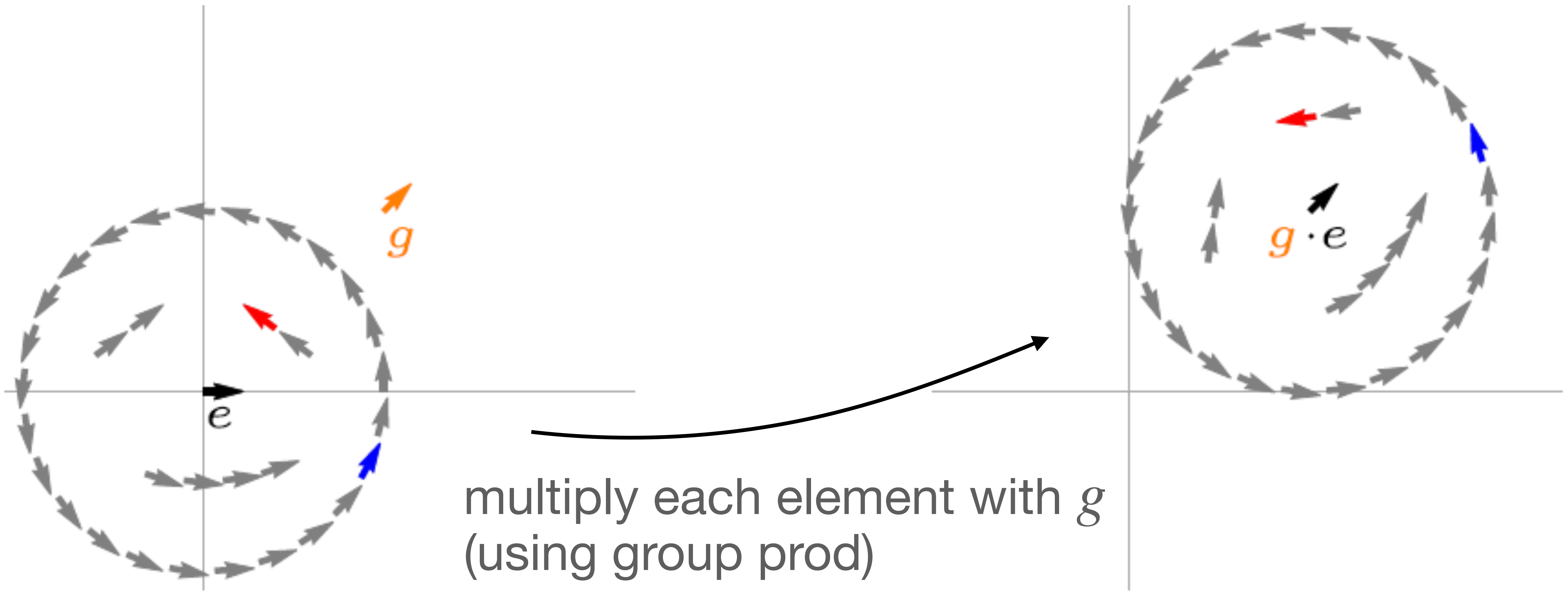
The roto-translation group $SE(2)$

*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:

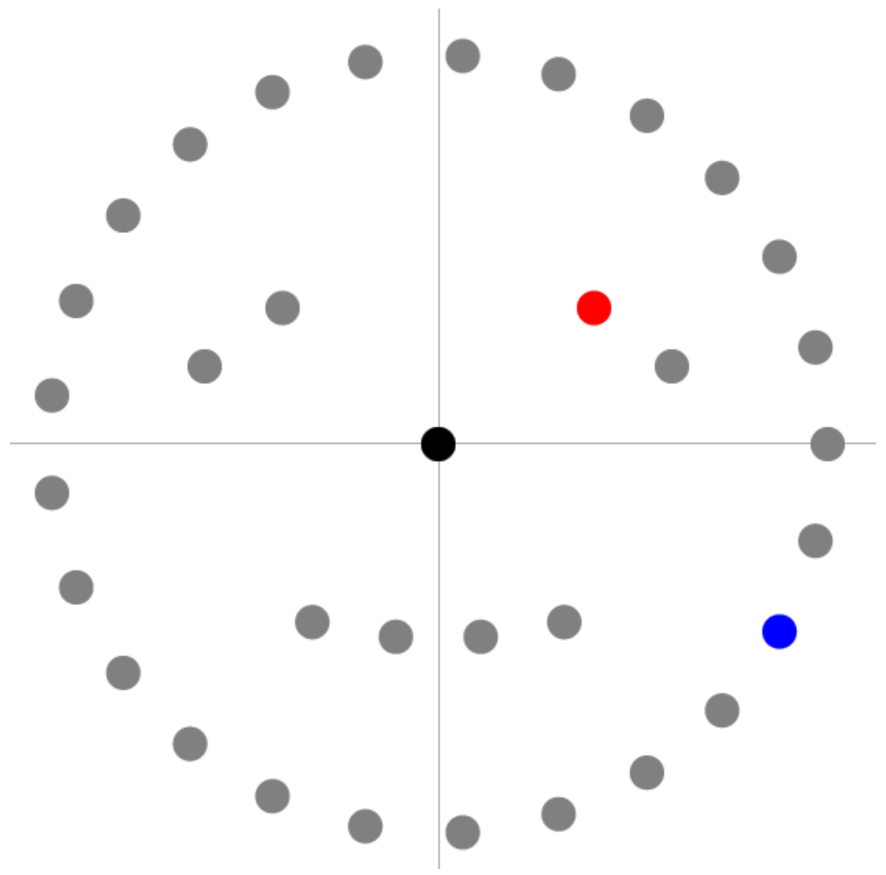
$$g \cdot g' = (\mathbf{x}, \mathbf{R}_\theta) \cdot (\mathbf{x}', \mathbf{R}_{\theta'}) = (\mathbf{R}_\theta \mathbf{x}' + \mathbf{x}, \mathbf{R}_{\theta+\theta'})$$
$$g^{-1} = (-\mathbf{R}_\theta^{-1}\mathbf{x}, \mathbf{R}_\theta^{-1})$$

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



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

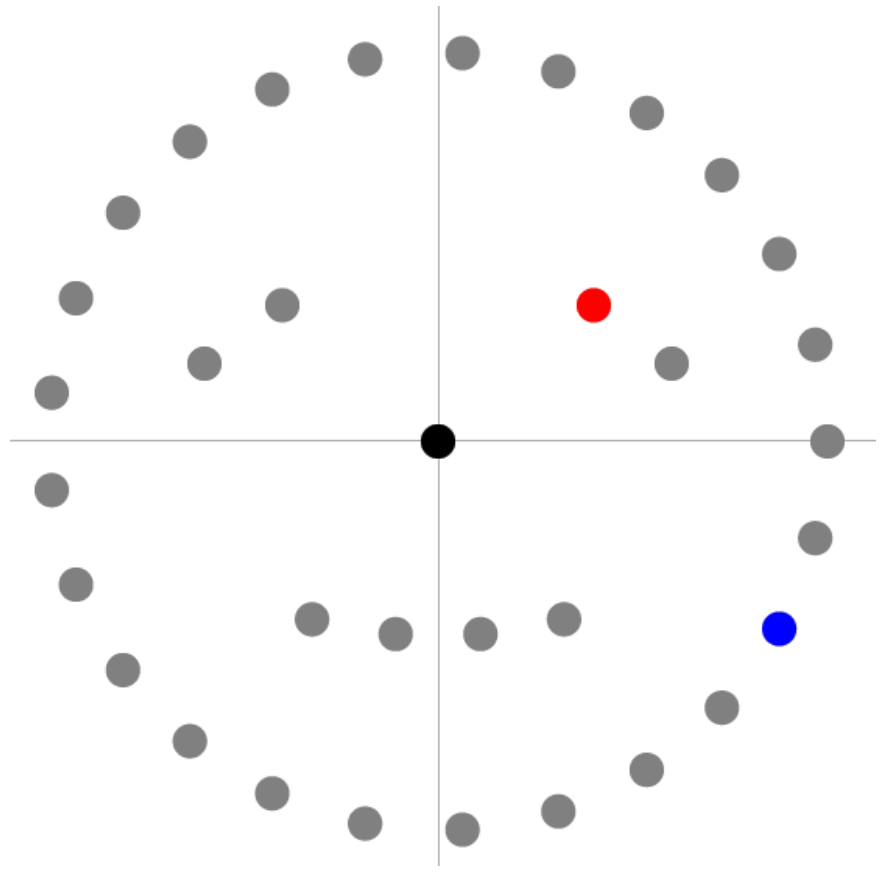
Set of points (group elements)



Convolution kernel

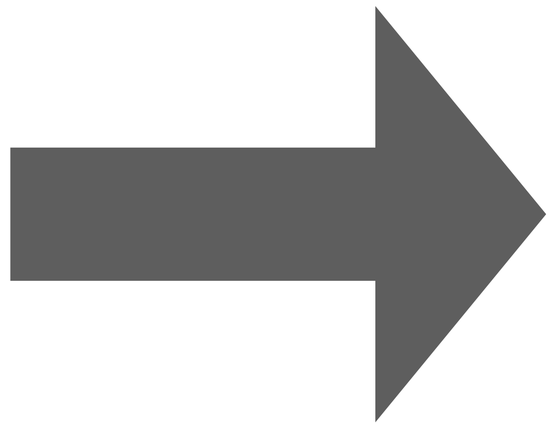
So… How to translate this to (G-)CNNs?

Set of points (group elements)



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

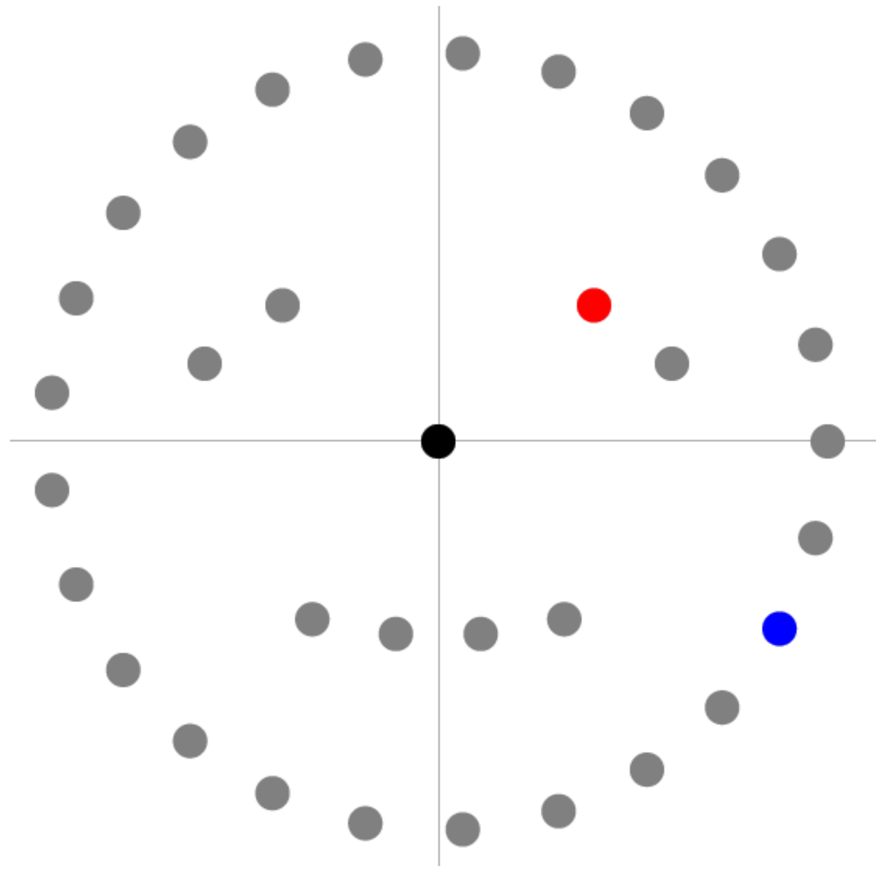
"A collection of parts in certain poses"



Convolution kernel

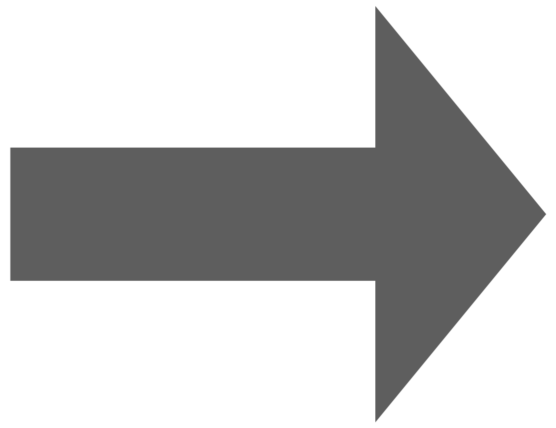
So… How to translate this to (G-)CNNs?

Set of points (group elements)



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

"A collection of parts in certain poses"



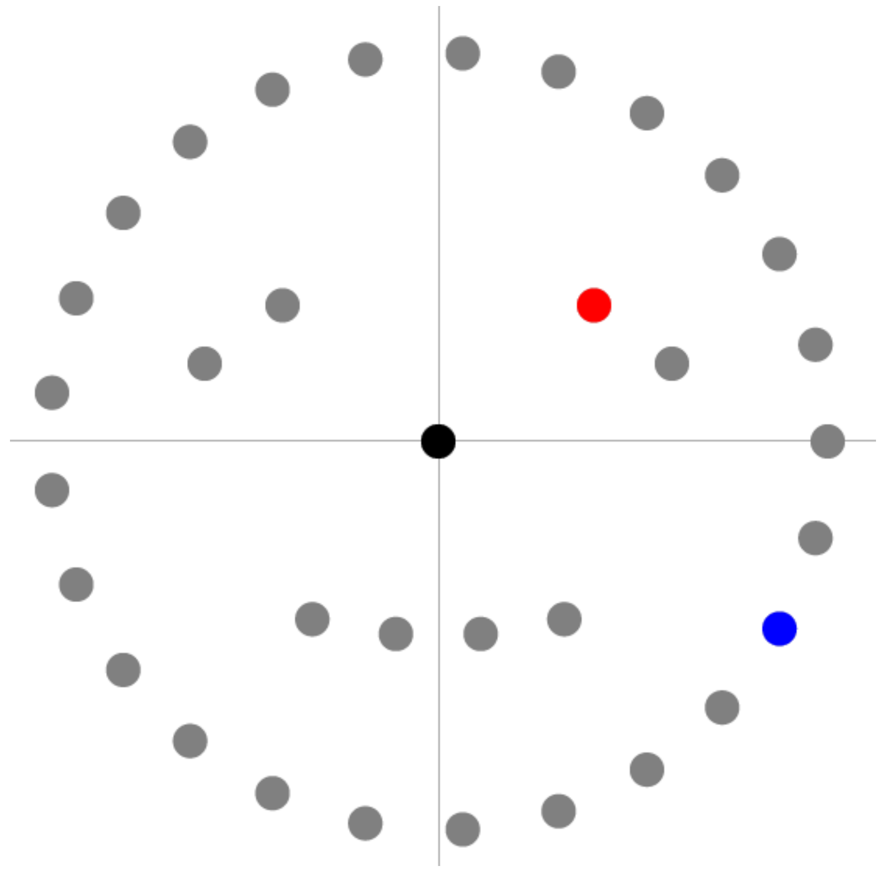
Convolution kernel

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

"Assigning weights to relative poses"

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

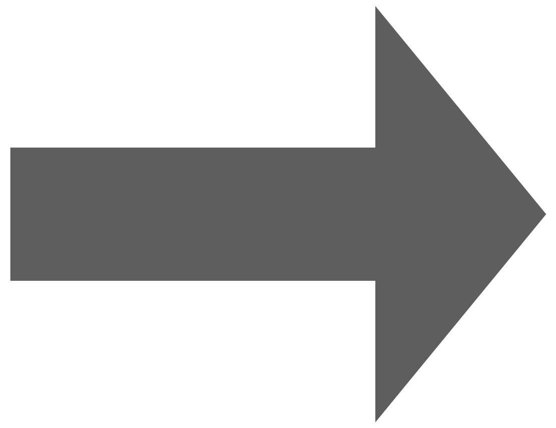
Set of points (group elements)



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

"A collection of parts in certain poses"

Transforms via group product



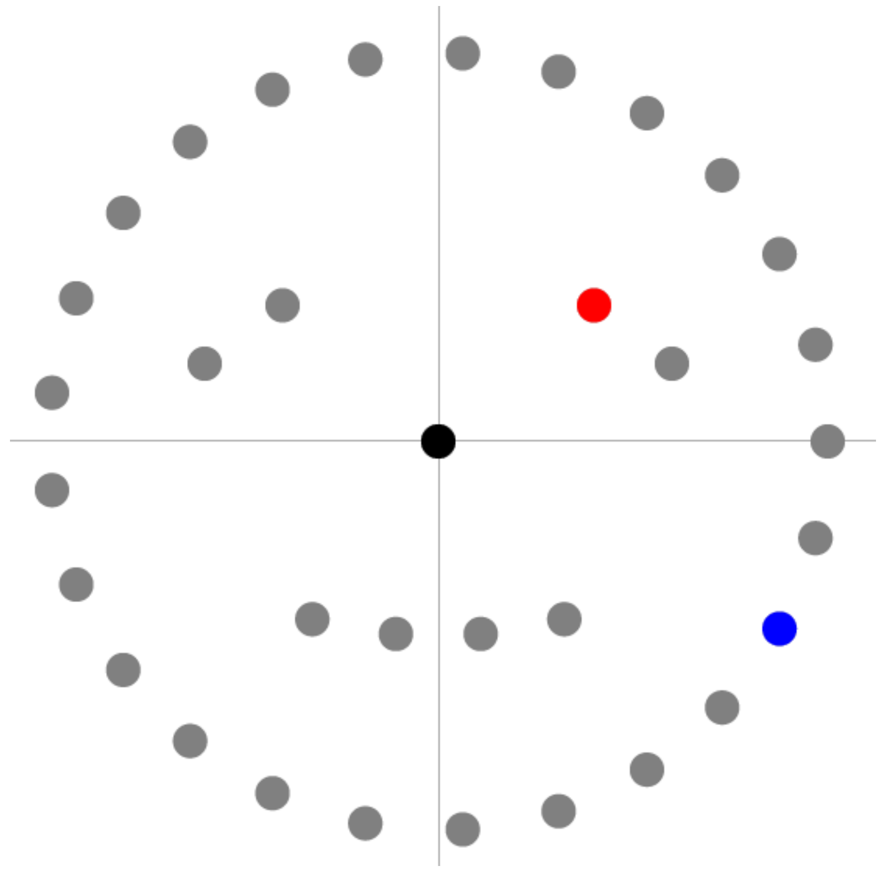
Convolution kernel

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

"Assigning weights to relative poses"

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

Set of points (group elements)



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

"A collection of parts in certain poses"

Transforms via group product

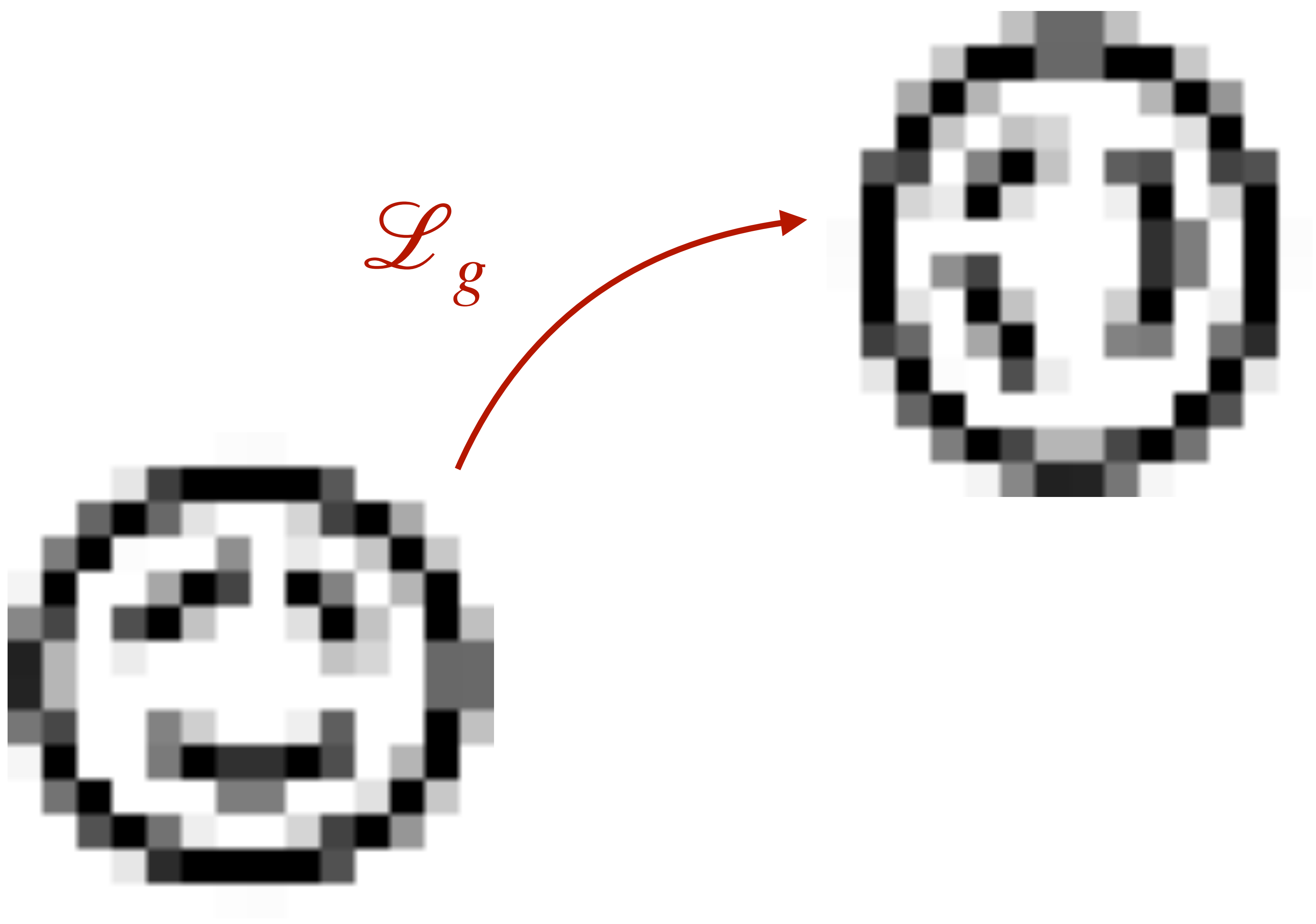
Convolution kernel

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

"Assigning weights to relative poses"

Transforms via group representations

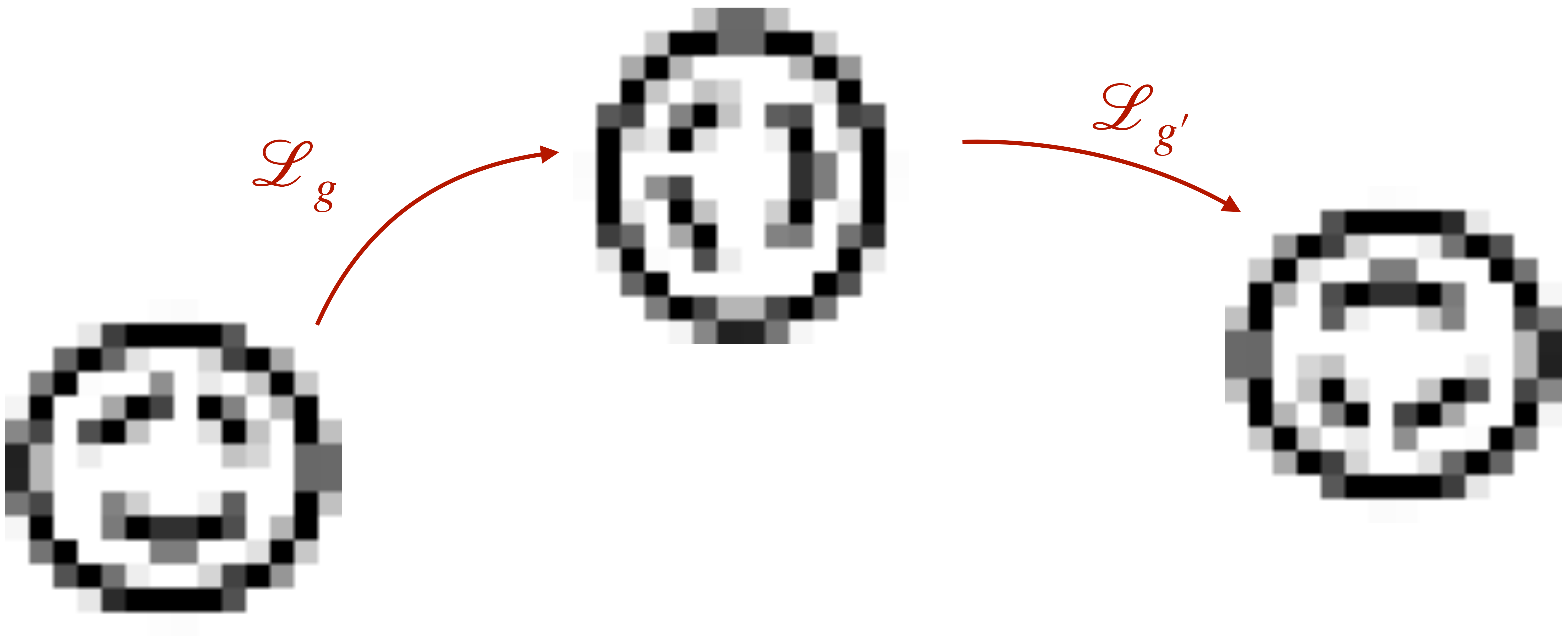
Representations



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

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

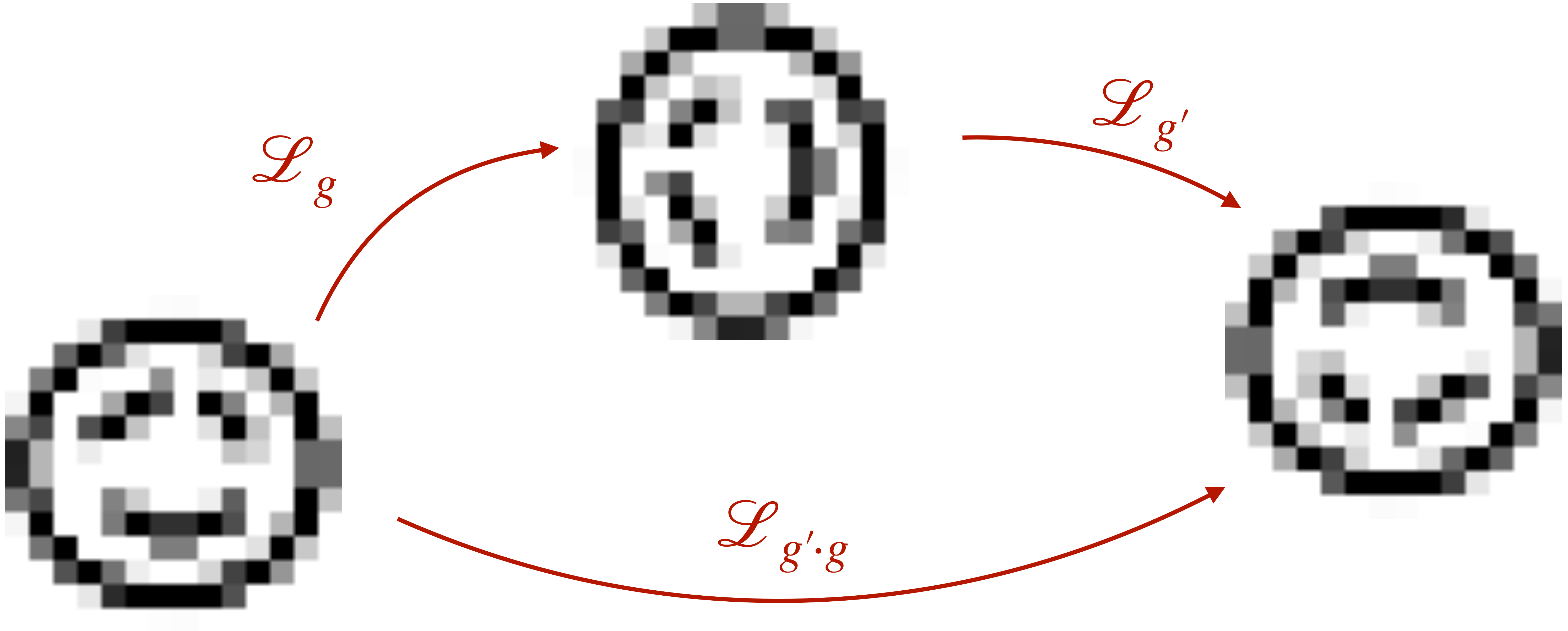
Representations



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

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

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

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

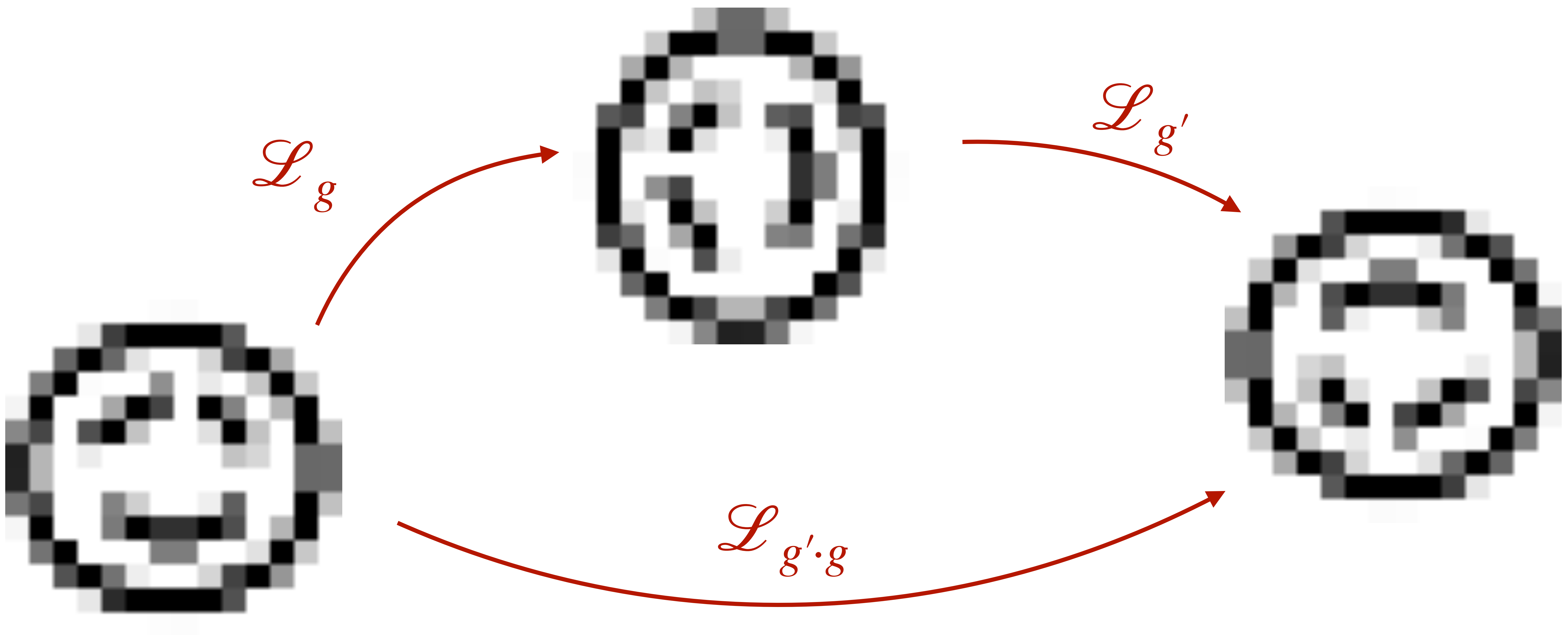
Left-regular representations

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

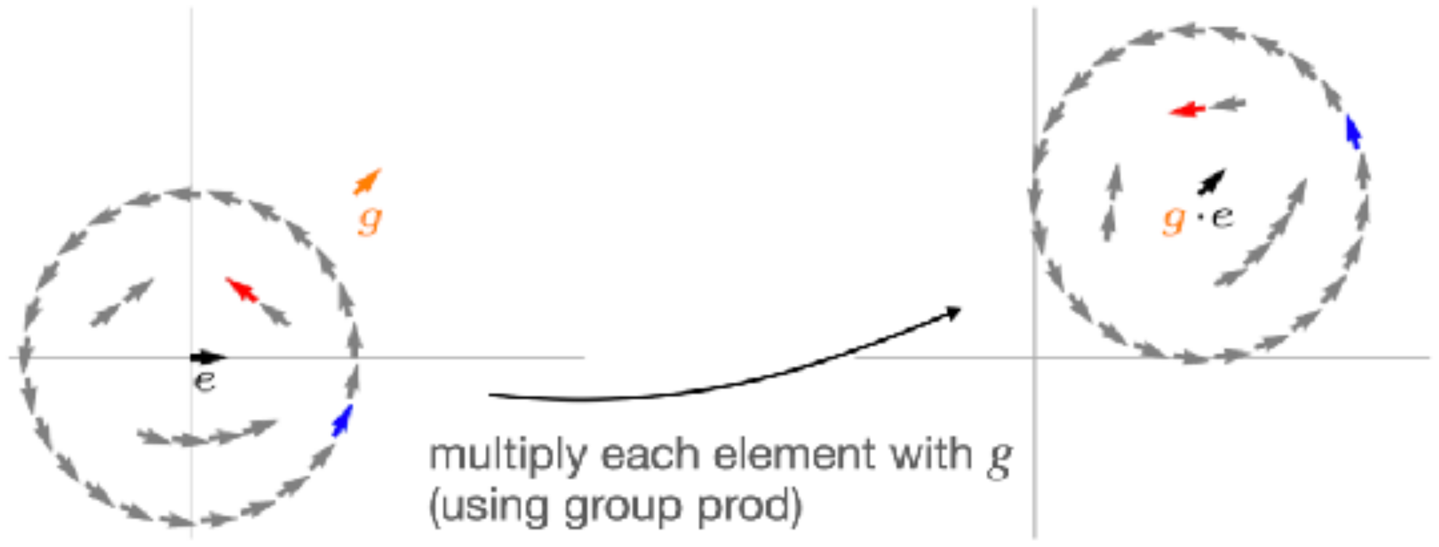
$$\mathscr{L}_g(f)(\mathbf{y}) = f(g^{-1} \odot \mathbf{y})$$

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

Group actions

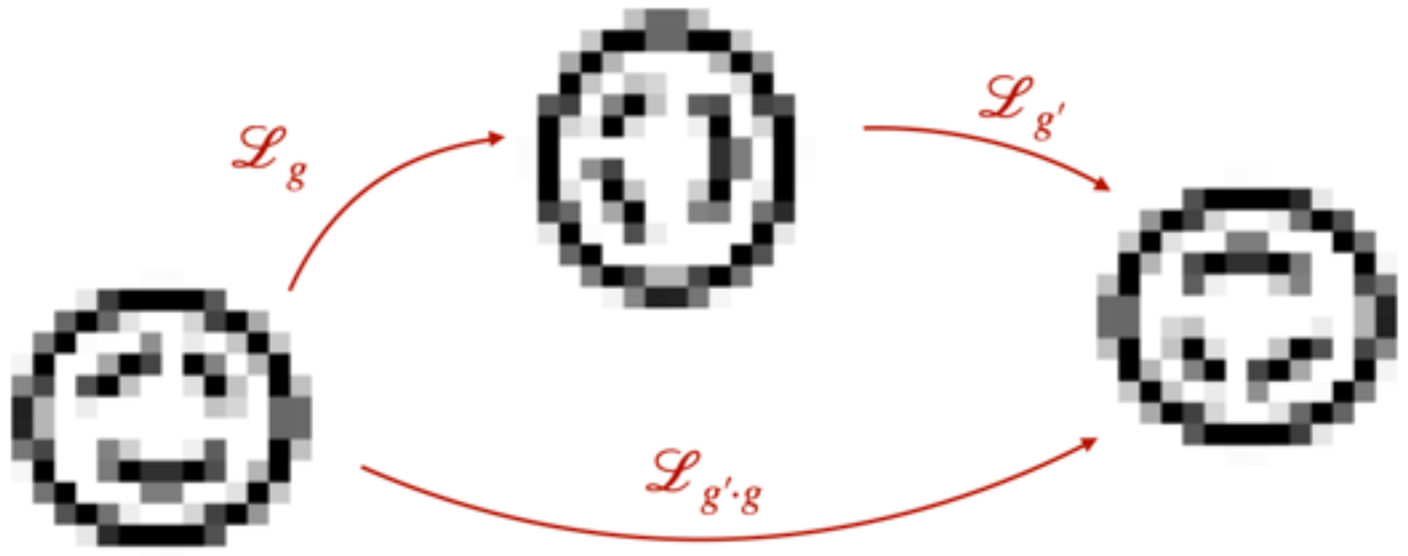
Group product (the action on G)

$$g \cdot g'$$



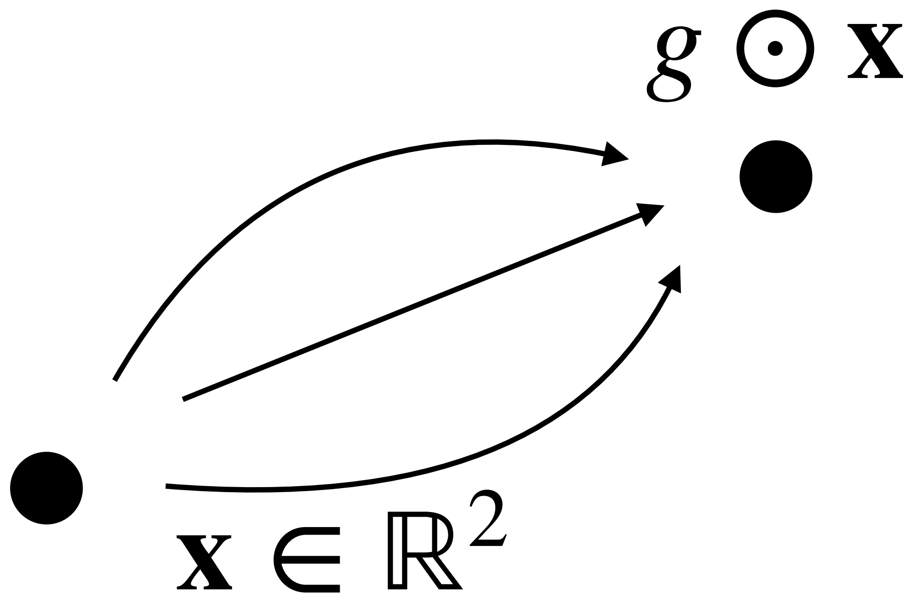
Left regular representation (the action on $\mathbb{L}_2(X)$)

$$\mathscr{L}_g f$$



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

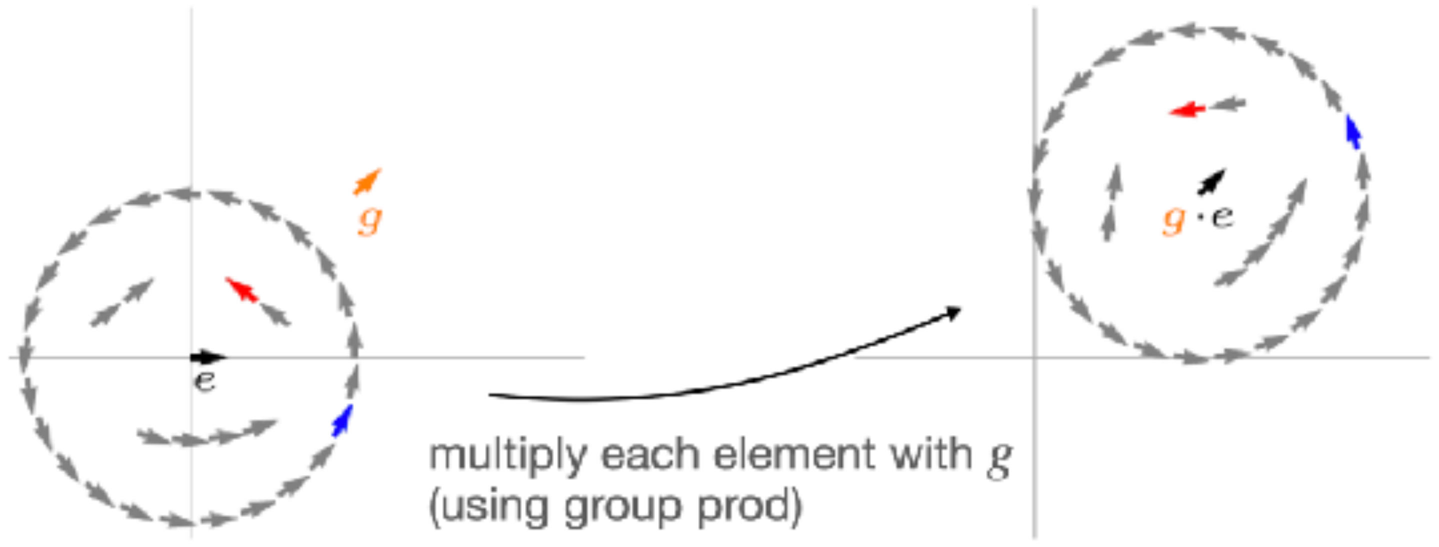
$$g \odot \mathbf{x}$$



Group actions

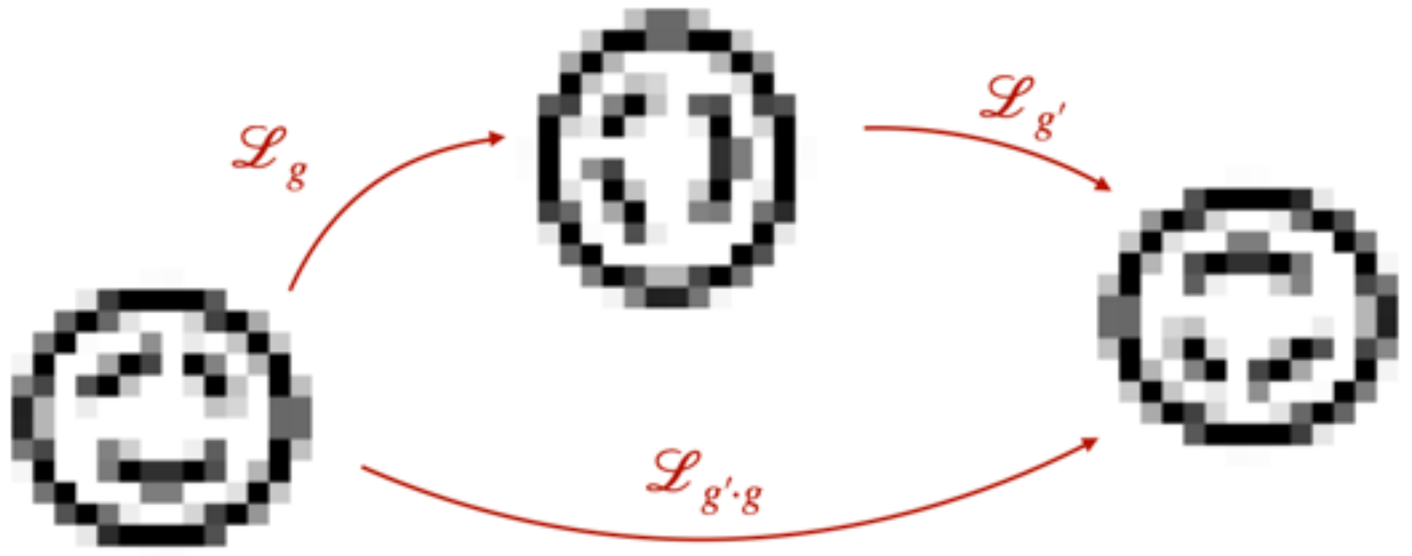
Group product (the action on G)

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



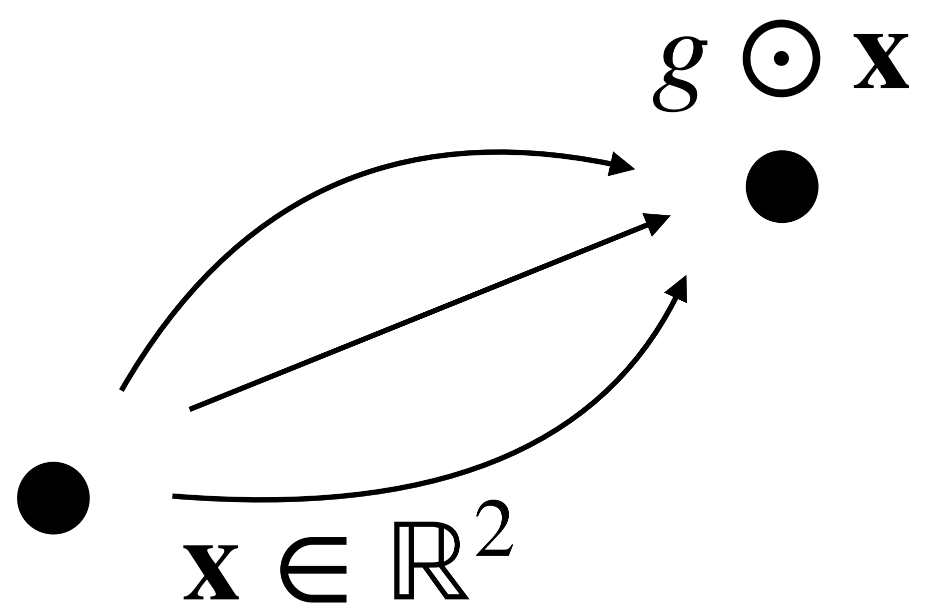
Left regular representation (the action on $\mathbb{L}_2(X)$)

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



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

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

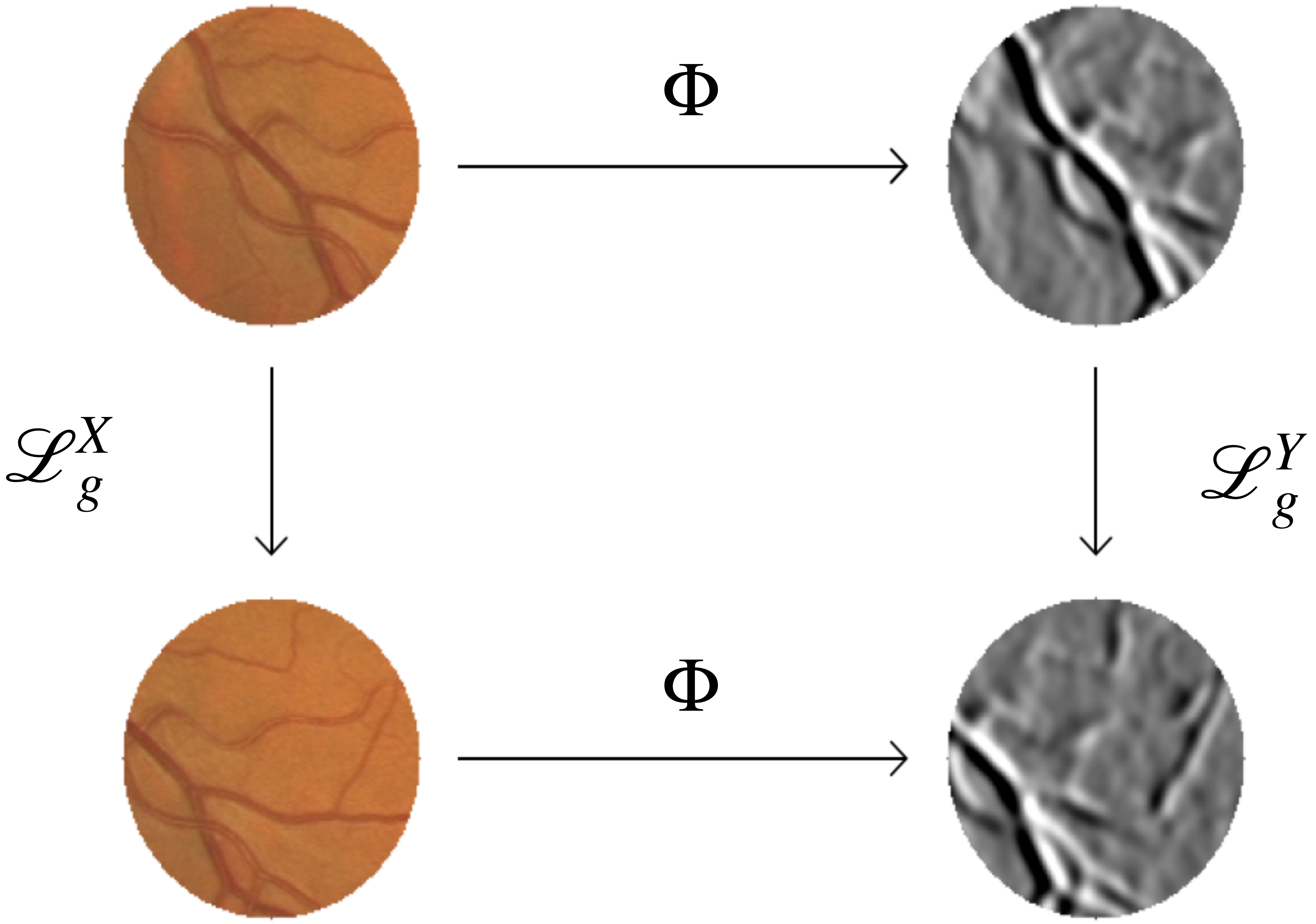


Equivariance

$$\mathscr{L}_g^Y \circ \Phi = \Phi \circ \mathscr{L}_g^X$$

$\Phi : X \to Y$

$\mathscr{L}^X$ and $\mathscr{L}^Y$ actions of G on X and Y



Content

Part I: Introduction to group convolutions

* Motivation

* **Introduction to group theory**

* **Regular group convolutional neural networks**

* Applications

“recogition-by-components”

Part II: Group convolutions are all you need

* Theorem: Equivariant linear operators are group convolutions

* Characterization of types of group equivariant layers

Part III: Steerable group convolutions for molecular data and the N-body problem

* Graph Neural networks
* Irreducible representations, steerable operators and vector spaces
* Steerable Graph NNs (Point Convolutions)

Are convolutions with reflected conv kernels (and vice versa)

Cross-correlations

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

Are convolutions with reflected conv kernels (and vice versa)

Cross-correlations

Representation of the translation group!

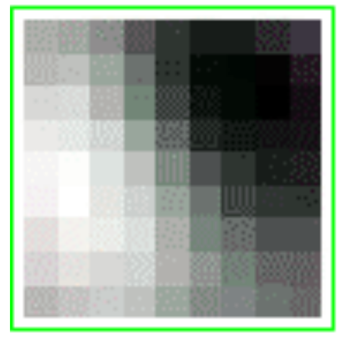
$$(k \star_{\mathbb{R}^2} f)(\mathbf{x}) = \int_{\mathbb{R}^2} k(\mathbf{x}' - \mathbf{x})f(\mathbf{x}')d\mathbf{x}' = (\mathscr{L}_g\, k\, ,\, f)_{\mathbb{L}_2(\mathbb{R}^2)}$$

Are convolutions with reflected conv kernels (and vice versa)

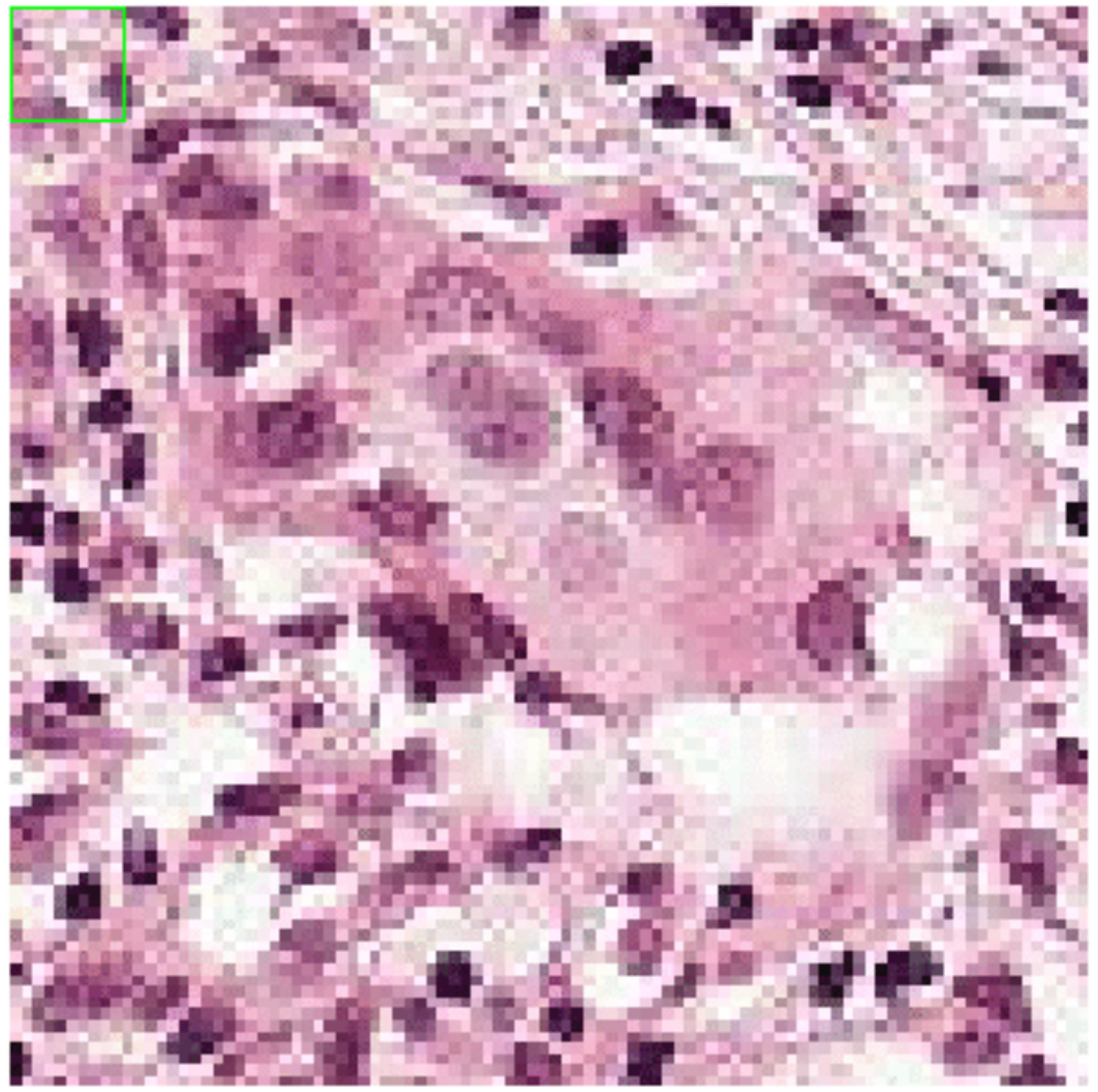
Cross-correlations

Representation of the translation group!

$$(k \star_{\mathbb{R}^2} f)(\mathbf{x}) = \int_{\mathbb{R}^2} k(\mathbf{x}' - \mathbf{x}) f(\mathbf{x}') \mathrm{d}\mathbf{x}' = (\mathscr{L}_g\, k\, ,\, f)_{\mathbb{L}_2(\mathbb{R}^2)}$$



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



$=$

k
2D convolution kernel

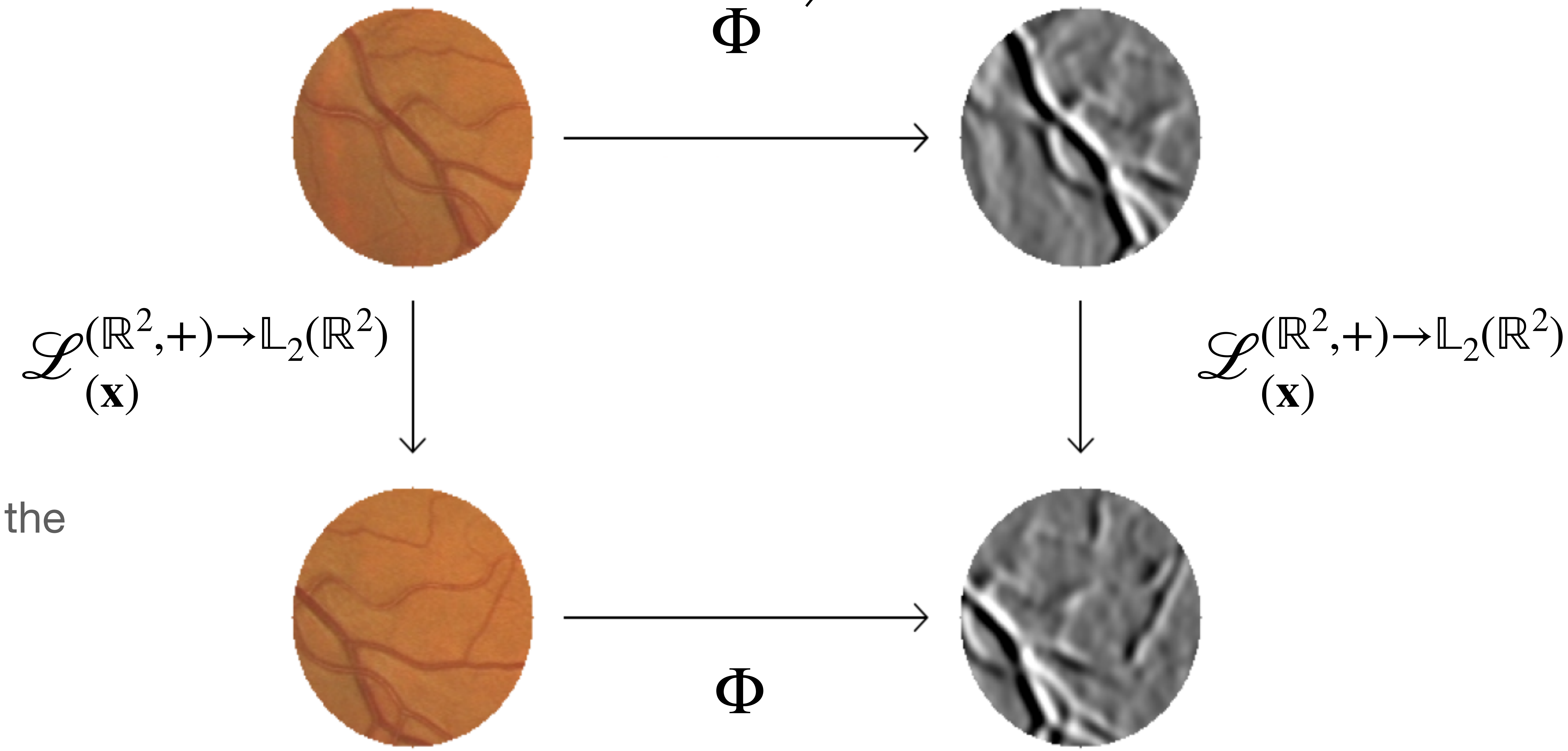
f^{in}
2D feature map

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

Group **equivariance**

Convolutions/cross-correlations are translation equivariant

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

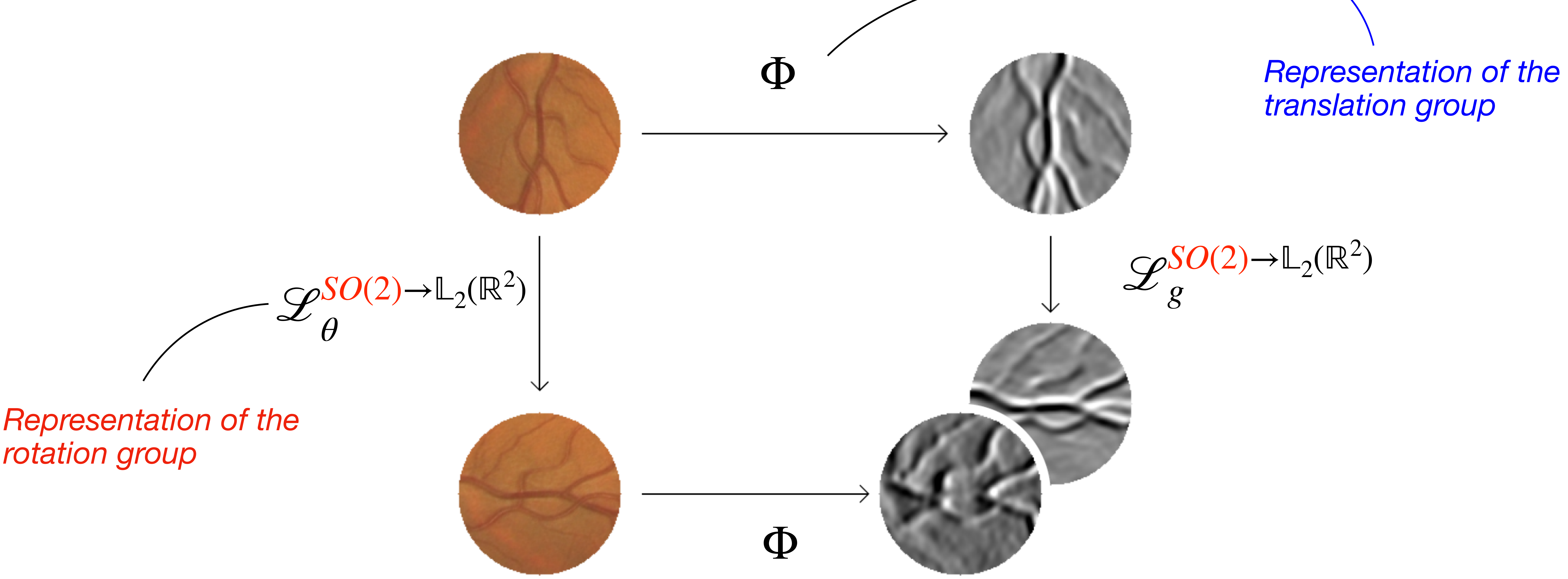


Representation of the translation group

Group **equivariance**

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

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



SE(2) equivariant cross-correlations

Representation of the roto-translation group!

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

SE(2) equivariant cross-correlations

Representation of the roto-translation group!

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

translation *rotation*

SE(2) equivariant cross-correlations

Representation of the roto-translation group!

Lifting correlations: $(k \tilde{\star} f)(\mathbf{x}) = (\mathscr{L}_g^{SE(2)\to\mathbb{L}_2(\mathbb{R}^2)} k \,, f)_{\mathbb{L}_2(\mathbb{R}^2)} = (\overbrace{\mathscr{L}_{\mathbf{x}}^{\mathbf{R}^2\to\mathbb{L}_2(\mathbb{R}^2)} \mathscr{L}_\theta^{SO(2)\to\mathbb{L}_2(\mathbb{R}^2)} k}^{k(\mathbf{R}_\theta^{-1}(\mathbf{x}'-\mathbf{x}))} \,, f)_{\mathbb{L}_2(\mathbb{R}^2)}$

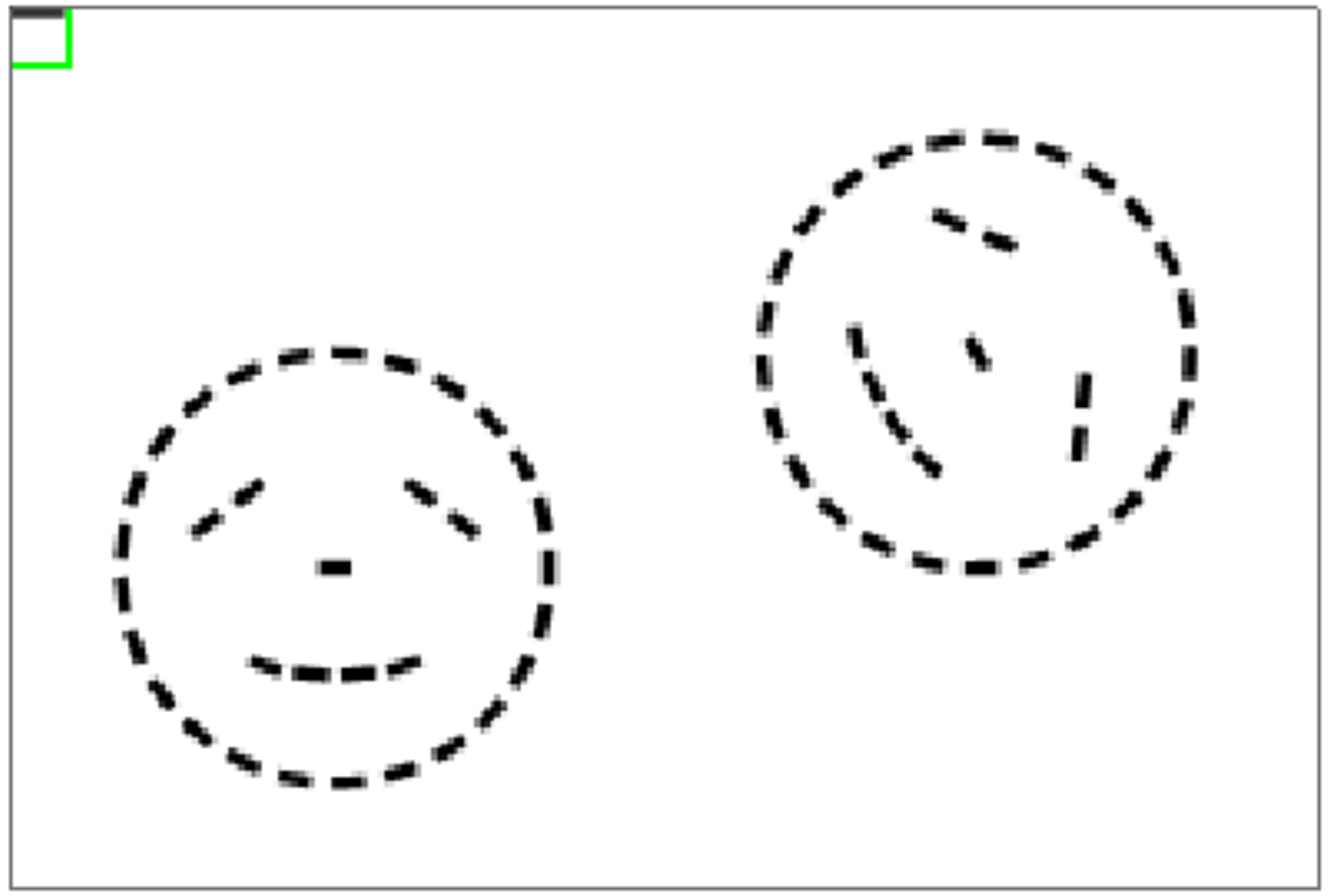
translation *rotation*

SE(2) equivariant cross-correlations

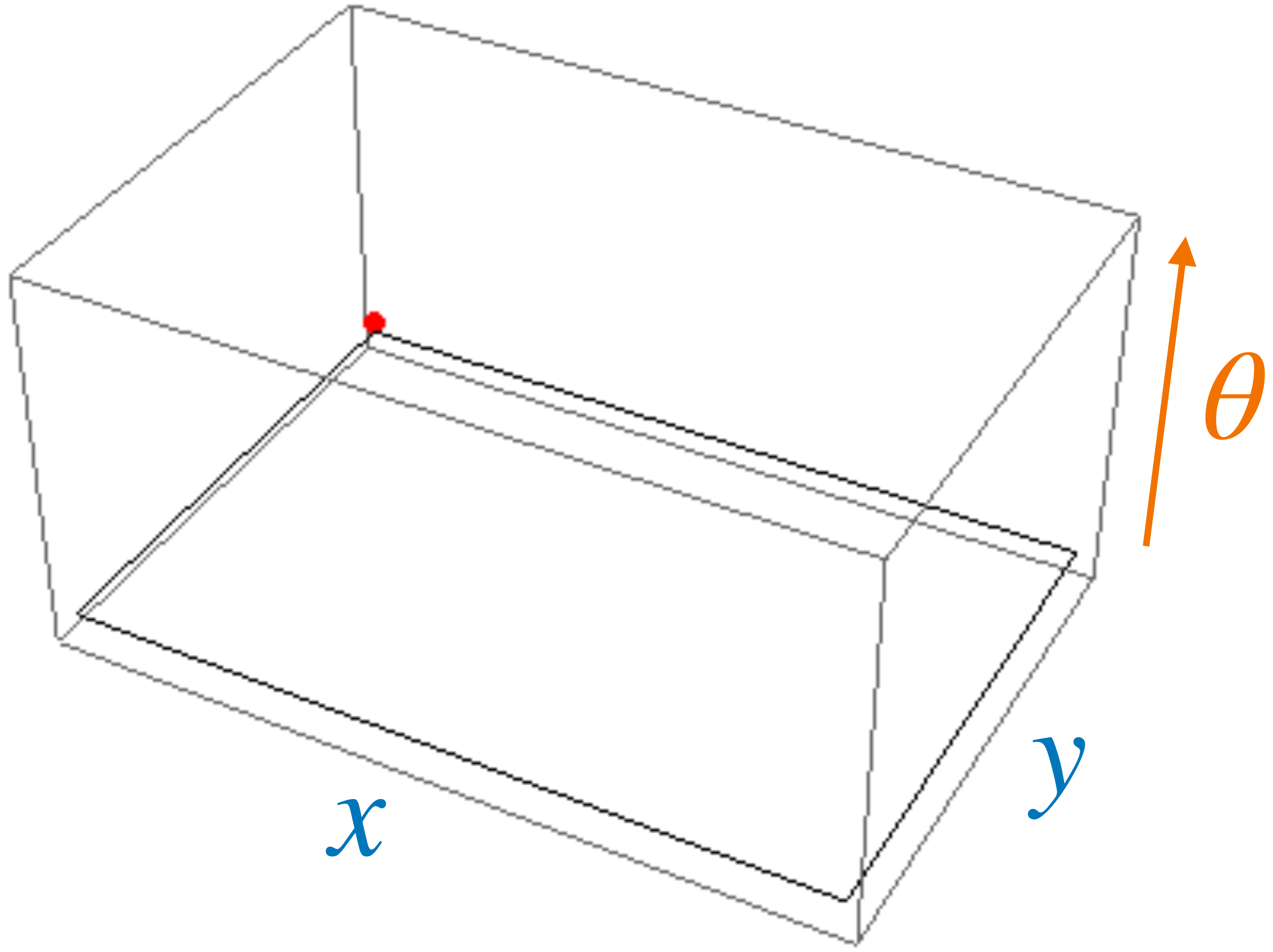
Representation of the roto-translation group!

Lifting correlations: $$(k \tilde{\star} f)(\mathbf{x}) = (\mathscr{L}_g^{SE(2)\to\mathbb{L}_2(\mathbb{R}^2)} k \,, f)_{\mathbb{L}_2(\mathbb{R}^2)} = (\overbrace{\underbrace{\mathscr{L}_{\mathbf{x}}^{\mathbf{R}^2\to\mathbb{L}_2(\mathbb{R}^2)}}_{\text{translation}} \underbrace{\mathscr{L}_\theta^{SO(2)\to\mathbb{L}_2(\mathbb{R}^2)}}_{\text{rotation}} k}^{k(\mathbf{R}_\theta^{-1}(\mathbf{x}'-\mathbf{x}))} \,, f)_{\mathbb{L}_2(\mathbb{R}^2)}$$

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



$=$



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

f^{in}
2D feature map

f^{out}
3D (*SE(2)*) feature map (after ReLU)

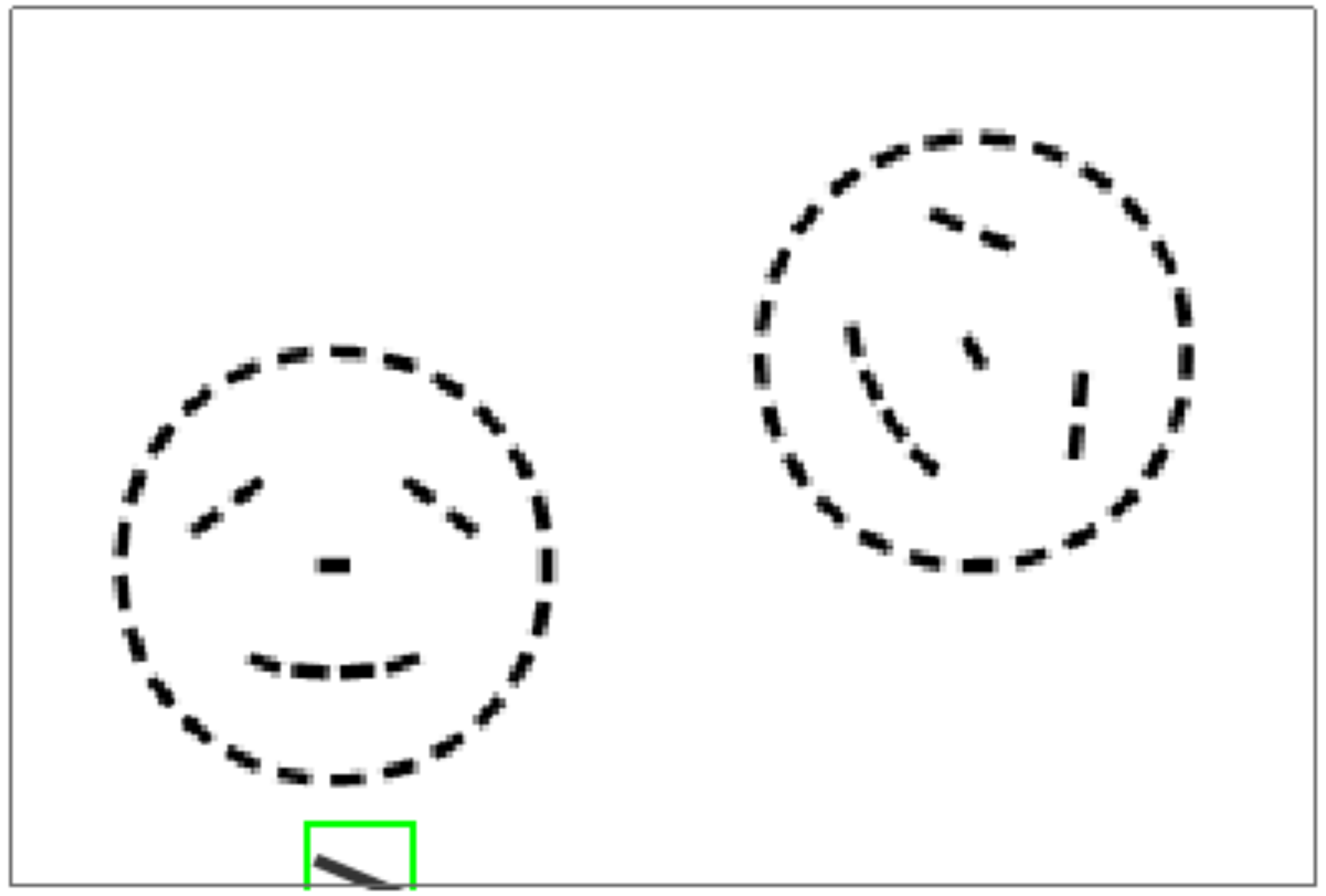
SE(2) equivariant cross-correlations

Representation of the roto-translation group!

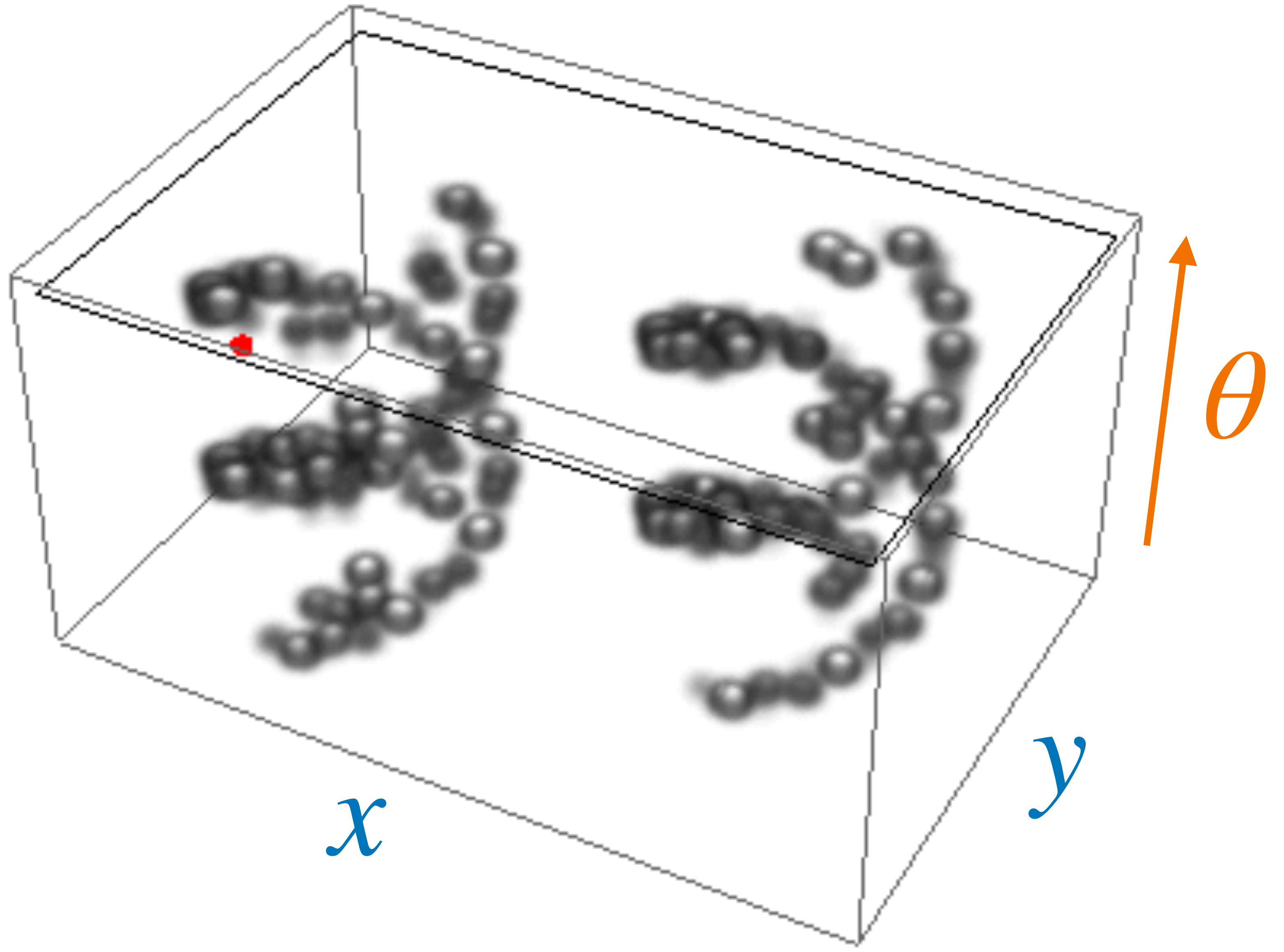
Lifting correlations: $(k \tilde{\star} f)(\mathbf{x}) = (\mathscr{L}_g^{SE(2)\to\mathbb{L}_2(\mathbb{R}^2)}\, k\,,\, f)_{\mathbb{L}_2(\mathbb{R}^2)} = (\overbrace{\mathscr{L}_{\mathbf{x}}^{\mathbf{R}^2\to\mathbb{L}_2(\mathbb{R}^2)}\, \mathscr{L}_\theta^{SO(2)\to\mathbb{L}_2(\mathbb{R}^2)}\, k}^{k(\mathbf{R}_\theta^{-1}(\mathbf{x}'-\mathbf{x}))}\,,\, f)_{\mathbb{L}_2(\mathbb{R}^2)}$

translation *rotation*

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



$=$



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

Rotated 2D convolution kernel

f^{in}

2D feature map

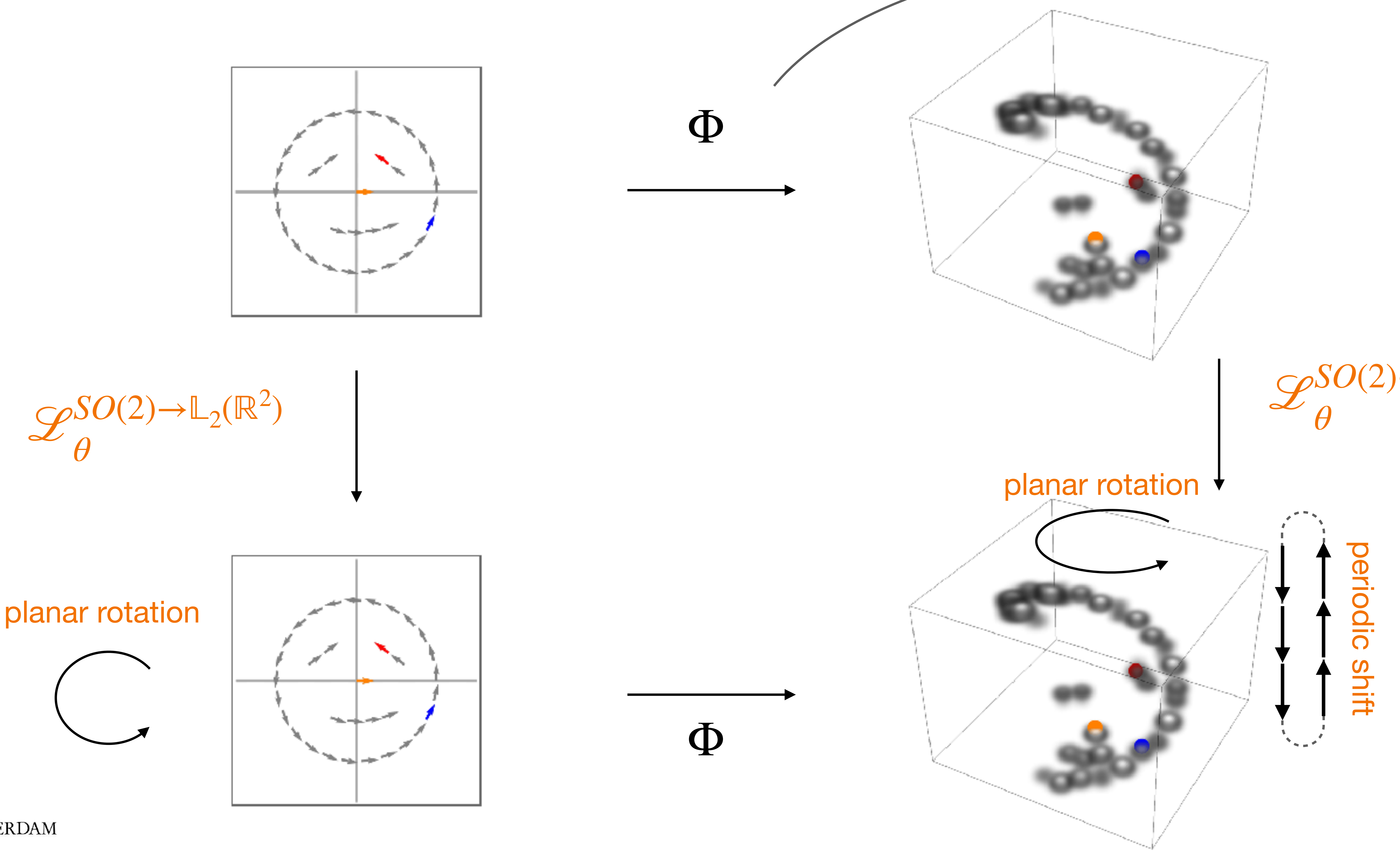
f^{out}

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

Group **equivariance**

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

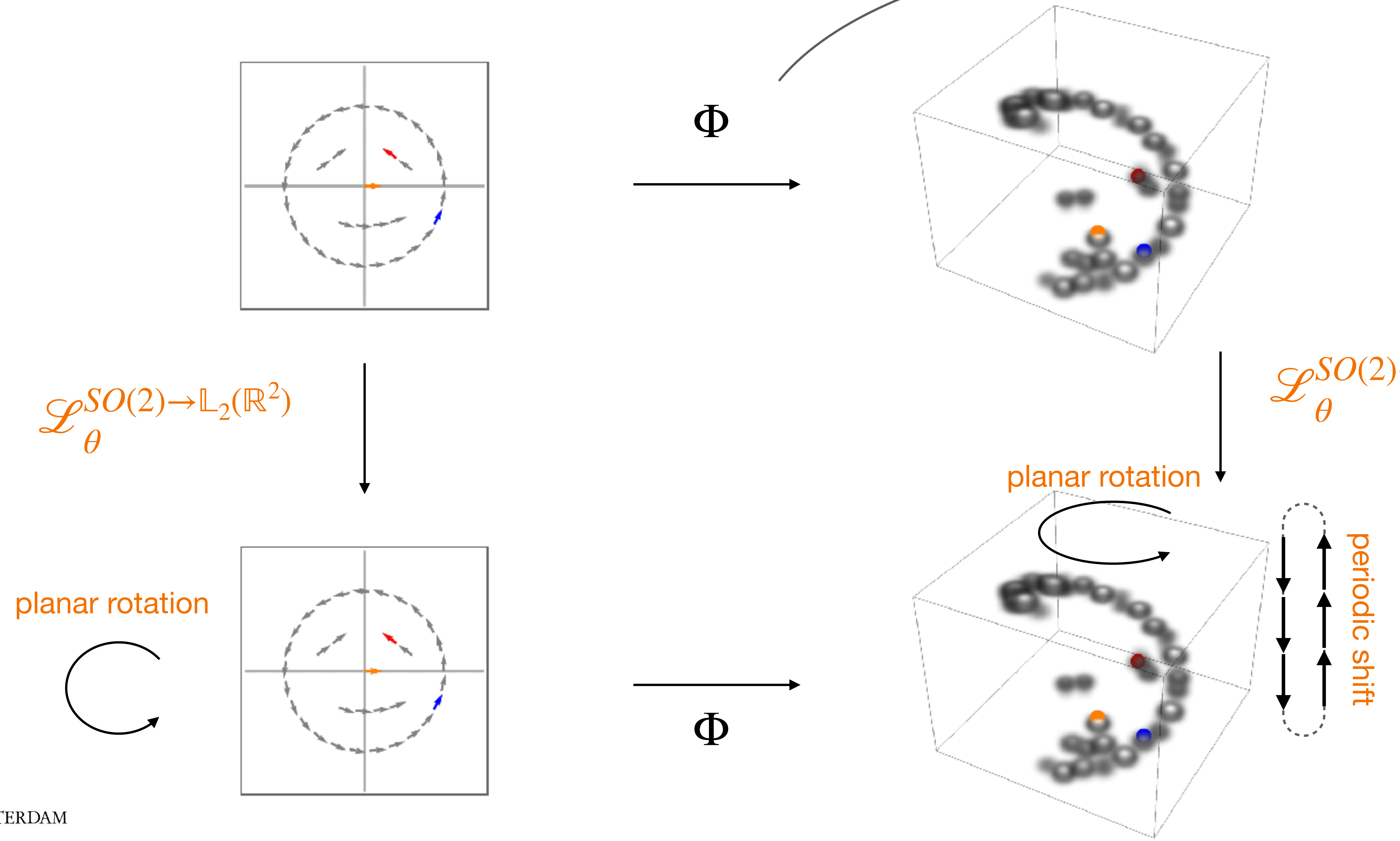
$$(k \,\tilde{\star}\, f)(\mathbf{x}) = (\mathscr{L}_{\mathbf{x}}^{\mathbf{R}^2\to\mathbb{L}_2(\mathbb{R}^2)} \mathscr{L}_{\theta}^{SO(2)\to\mathbb{L}_2(\mathbb{R}^2)} k \, , f)_{\mathbb{L}_2(\mathbb{R}^2)}$$



Group **equivariance**

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

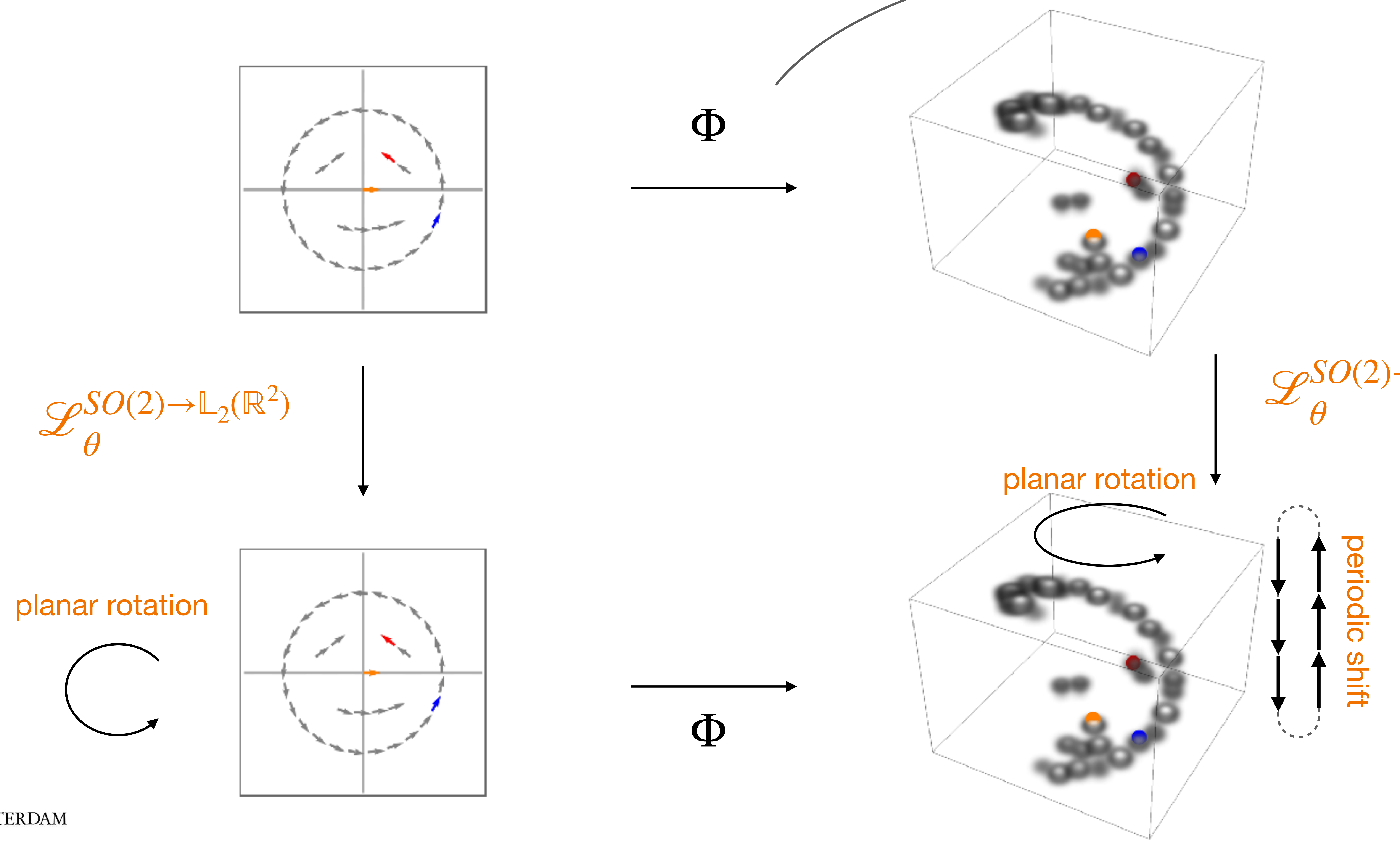
$$(k \,\tilde{\star}\, f)(\mathbf{x}) = (\mathscr{L}_{\mathbf{x}}^{\mathbf{R}^2\to\mathbb{L}_2(\mathbb{R}^2)} \mathscr{L}_{\theta}^{SO(2)\to\mathbb{L}_2(\mathbb{R}^2)} k \,,\, f)_{\mathbb{L}_2(\mathbb{R}^2)}$$



Group **equivariance**

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

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



What about subsequent layers?

SE(2) equivariant cross-correlations

Group correlations:

$$(k \star f)(\mathbf{x}) = (\mathscr{L}_g^{SE(2)\to\mathbb{L}_2(SE(2))}\, k\ ,\, f)_{\mathbb{L}_2(SE(2))} = (\overbrace{\mathscr{L}_{\mathbf{x}}^{\mathbf{R}^2\to\mathbb{L}_2(SE(2))}\,\mathscr{L}_{\theta}^{SO(2)\to\mathbb{L}_2(SE(2))}\, k}^{k(\mathbf{R}_\theta^{-1}(\mathbf{x}'-\mathbf{x}),\mathbf{R}_{\theta'-\theta})}\ ,\, f)_{\mathbb{L}_2(SE(2)}$$

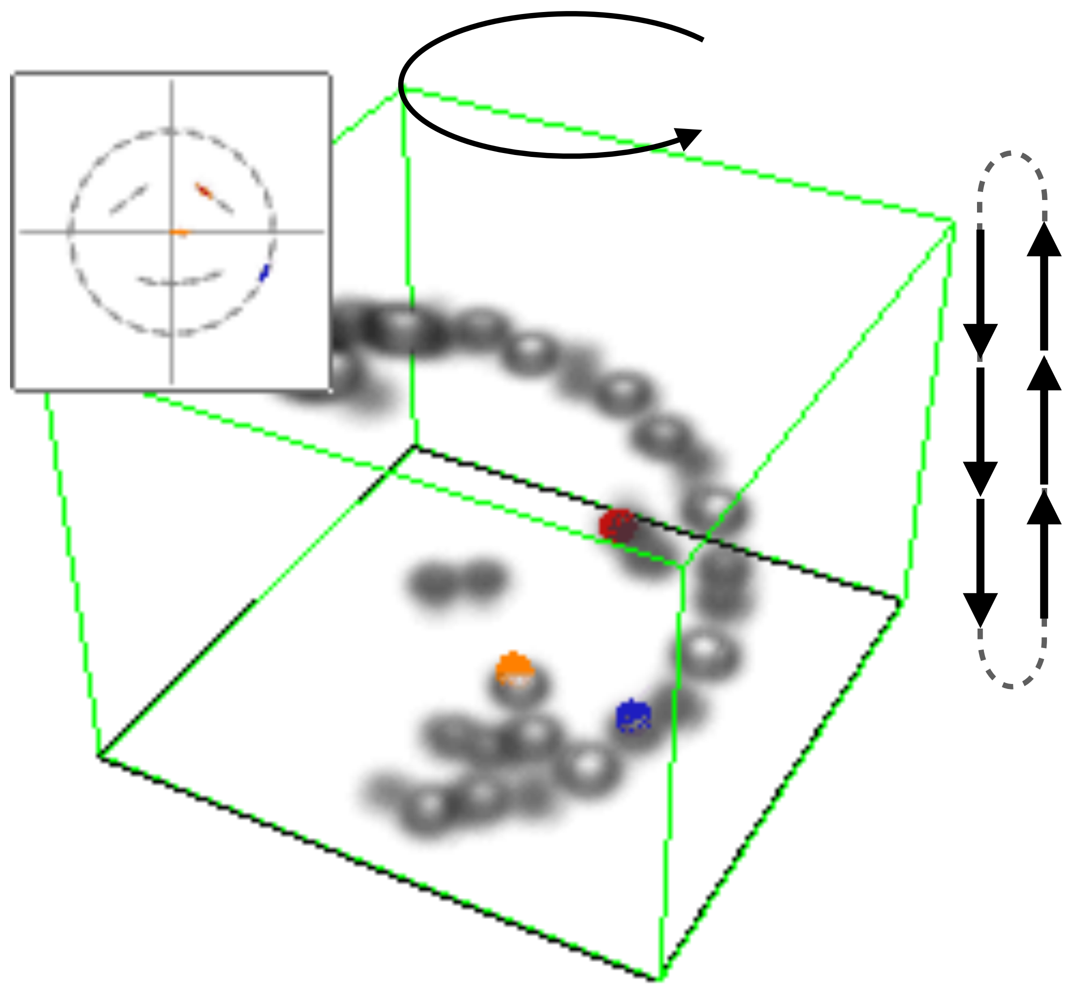
translation *rotation*

SE(2) equivariant cross-correlations

Group correlations:

$$(k \star f)(\mathbf{x}) = (\mathscr{L}_g^{SE(2)\to\mathbb{L}_2(SE(2))}\, k \,,\, f)_{\mathbb{L}_2(SE(2))} = (\overbrace{\mathscr{L}_{\mathbf{x}}^{\mathbf{R}^2\to\mathbb{L}_2(SE(2))}\,\mathscr{L}_{\theta}^{SO(2)\to\mathbb{L}_2(SE(2))}\, k}^{k(\mathbf{R}_\theta^{-1}(\mathbf{x}'-\mathbf{x}),\mathbf{R}_{\theta'-\theta}}\,,\, f)_{\mathbb{L}_2(SE(2)}$$

translation *rotation*



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

Rotated SE(2) convolution kernel

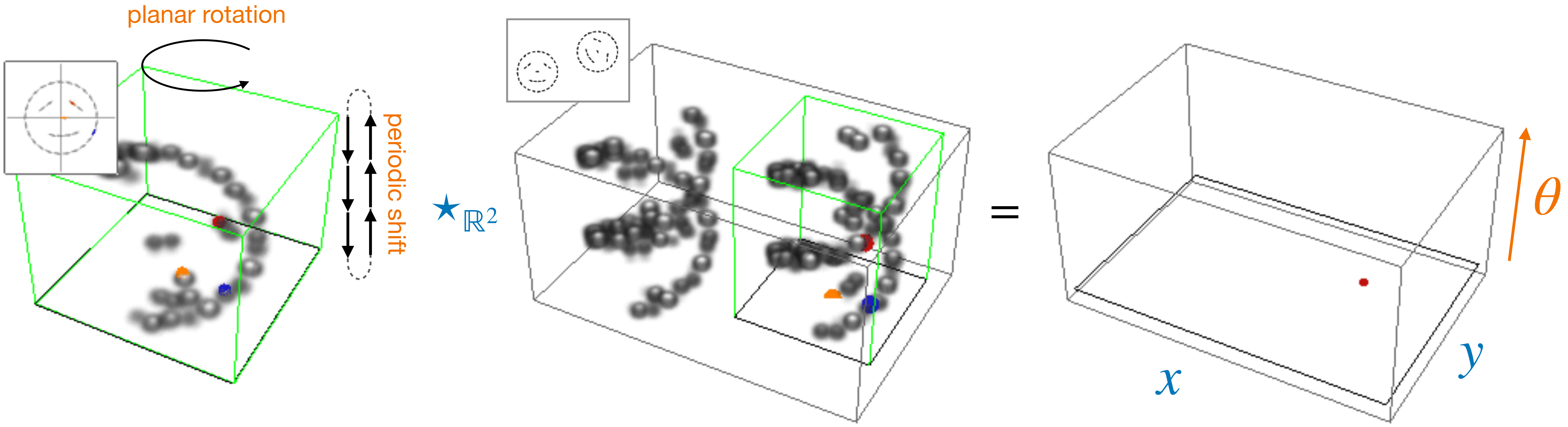
SE(2) equivariant cross-correlations

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

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

f^{in}

SE(2) feature map

f^{out}

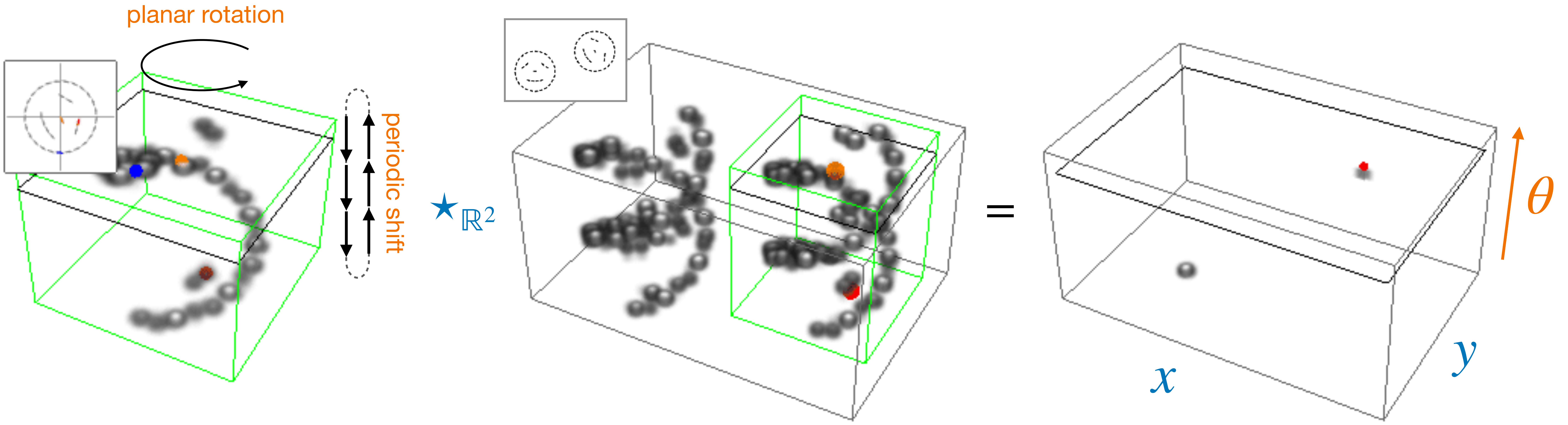
SE(2) feature map (after ReLU)

SE(2) equivariant cross-correlations

Group correlations:

$$(k \star f)(\mathbf{x}) = (\mathscr{L}_g^{SE(2)\to\mathbb{L}_2(SE(2))}\, k\,,\, f)_{\mathbb{L}_2(SE(2))} = (\underbrace{\mathscr{L}_{\mathbf{x}}^{\mathbf{R}^2\to\mathbb{L}_2(SE(2))}\,\mathscr{L}_\theta^{SO(2)\to\mathbb{L}_2(SE(2))}\, k}_{k(\mathbf{R}_\theta^{-1}(\mathbf{x}'-\mathbf{x}),\mathbf{R}_{\theta'-\theta}}\,,\, f)_{\mathbb{L}_2(SE(2)}$$

translation rotation



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

f^{in}
SE(2) feature map

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

Attentive Group Equivariant Convolutional Networks

David W. Romero [1] Erik J. Bekkers [2] Jakub M. Tomczak [1] Mark Hoogendoorn [1]

Abstract

roup convolutional networks are able
werful representations based on sym-
erns, they lack explicit means to learn
l relationships among them (e.g., rela-
ns and poses). In this paper, we present
roup equivariant convolutions, a gen-
of the group convolution, in which
s applied during the course of convo-
ccentuate meaningful symmetry com-
and suppress non-plausible, mislead-
We indicate that prior work on visual
an be described as special cases of
ed framework and show empirically
entive group equivariant convolutional
consistently outperform conventional
olutional networks on benchmark im-

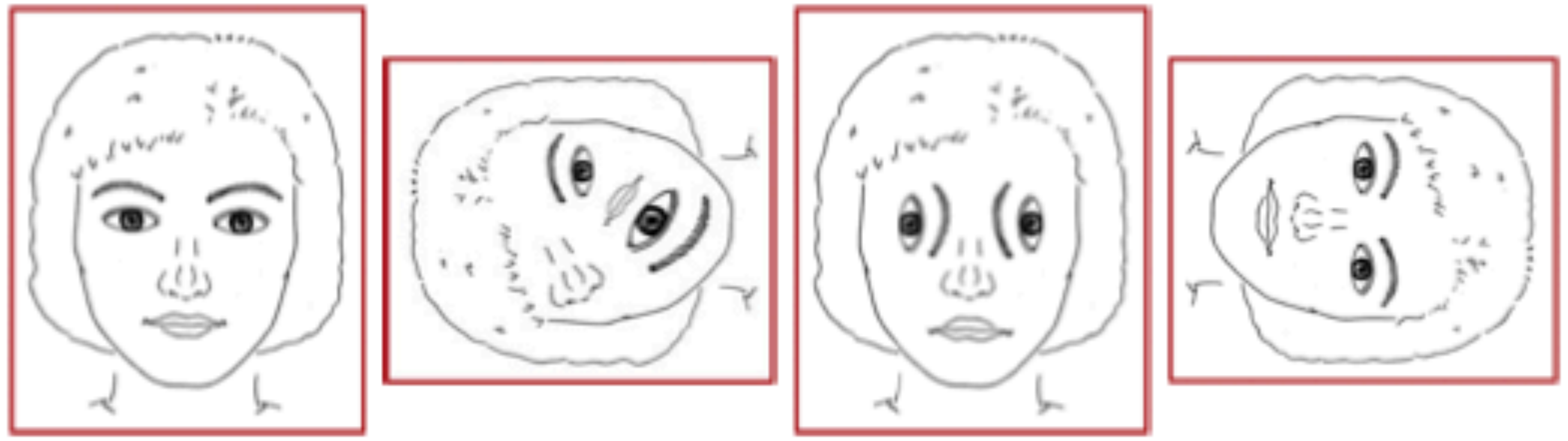


Figure 1. Meaningful relationships among object symmetries. Though every figure is composed by the same elements, only the outermost examples resemble faces. The relative positions, orientations and scales of elements in the innermost examples do not match any meaningful face composition and hence, should not be labelled as such. Built upon Fig. 1 from Schwarzer (2000).

cations in location, size, viewpoint, lighting conditions and background (Bruce & Humphreys, 1994). In addition, we do not just recognize them but are able to describe in detail the type and amount of modification applied to them as well

Attentive Group Equivariant Convolutional Networks

David W. Romero[1] Erik J. Bekkers[2] Jakub M. Tomczak[1] Mark Hoogendoorn[1]

Group convolutions
solve the
Picasso problem!

Abstract

roup convolutional networks are able
werful representations based on sym-
erns, they lack explicit means to learn
l relationships among them (e.g., rela-
ns and poses). In this paper, we present
roup equivariant convolutions, a gen-
of the group convolution, in which
s applied during the course of convo-
ccentuate meaningful symmetry com-
nd suppress non-plausible, mislead-
We indicate that prior work on visual
an be described as special cases of
ed framework and show empirically
entive group equivariant convolutional
consistently outperform conventional
olutional networks on benchmark im-

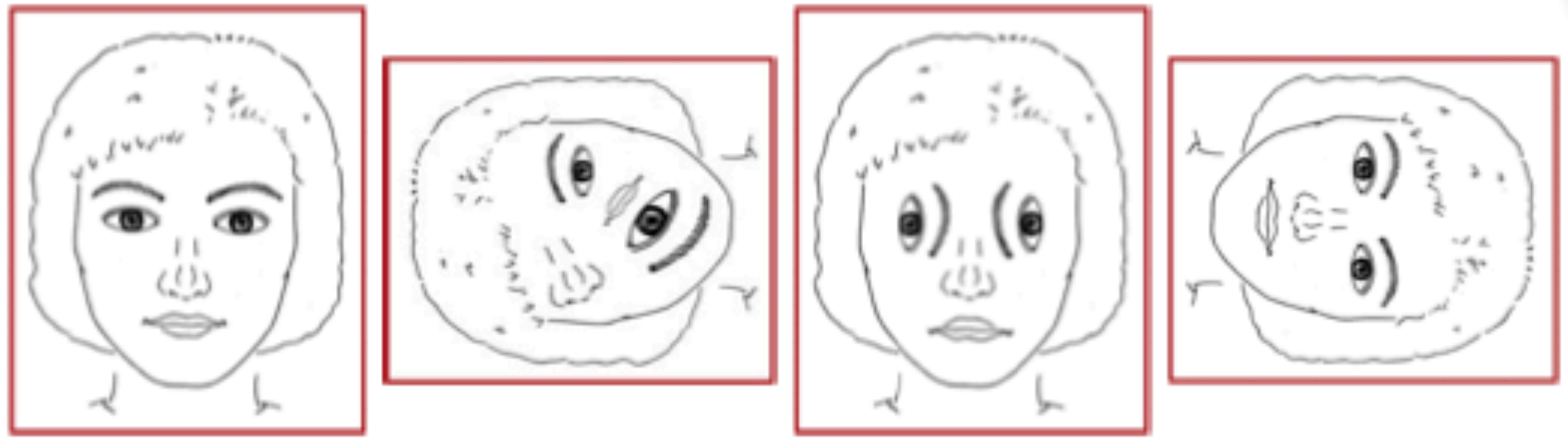


Figure 1. Meaningful relationships among object symmetries. Though every figure is composed by the same elements, only the outermost examples resemble faces. The relative positions, orientations and scales of elements in the innermost examples do not match any meaningful face composition and hence, should not be labelled as such. Built upon Fig. 1 from Schwarzer (2000).

cations in location, size, viewpoint, lighting conditions and background (Bruce & Humphreys, 1994). In addition, we do not just recognize them but are able to describe in detail the type and amount of modification applied to them as well

Attentive Group Equivariant Convolutional Networks

David W. Romero [1] Erik J. Bekkers [2] Jakub M. Tomczak [1] Mark Hoogendoorn [1]

Group convolutions solve the Picasso problem!

Abstract

roup convolutional networks are able
werful representations based on sym-
erns, they lack explicit means to learn
l relationships among them (e.g., rela-
ns and poses). In this paper, we present
roup equivariant convolutions, a gen-
of the group convolution, in which
s applied during the course of convo-
ccentuate meaningful symmetry com-
nd suppress non-plausible, mislead-
We indicate that prior work on visual
an be described as special cases of
ed framework and show empirically
entive group equivariant convolutional
onsistently outperform conventional
olutional networks on benchmark im-

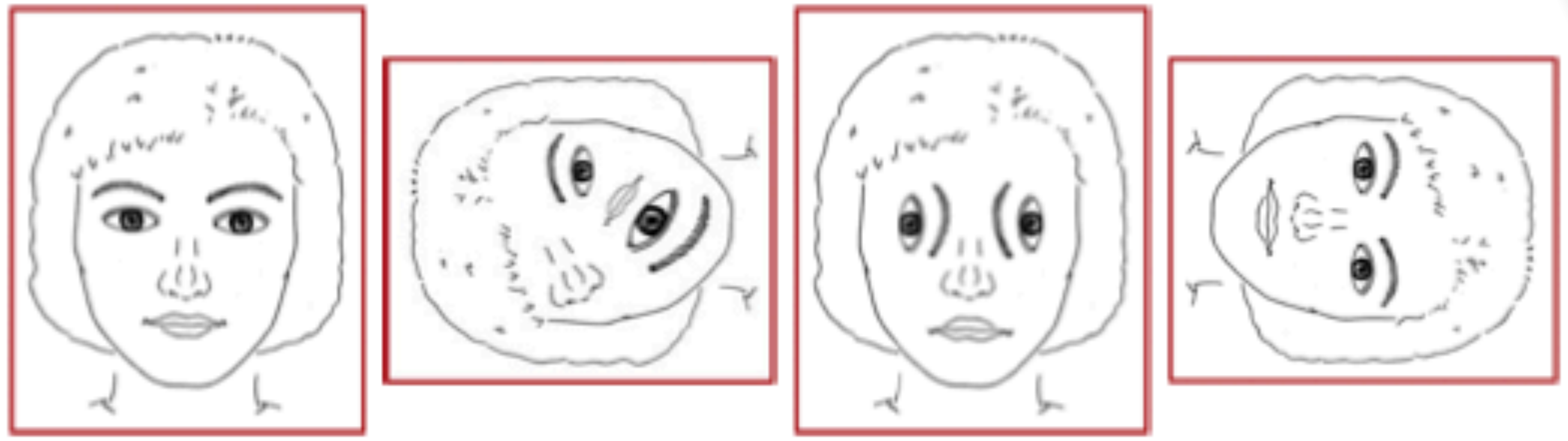


Figure 1. Meaningful relationships among object symmetries. Though every figure is composed by the same elements, only the outermost examples resemble faces. The relative positions, orientations and scales of elements in the innermost examples do not match any meaningful face composition and ... ld not be labelled as such. Built upon Fig. 1 from Sc...

Recognition by components in a group theoretical setting!

cations in location, size, viewpoint, lighting conditions and background (Bruce & Humphreys, 1994). In addition, we do not just recognize them but are able to describe in detail the type and amount of modification applied to them as well (von Helmholtz, 1868; Cassirer, 1944; Schmidt et al., 2016).

Attentive Group Equivariant Convolutional Networks

David W. Romero [1] Erik J. Bekkers [2] Jakub M. Tomczak [1] Mark Hoogendoorn [1]

Group convolutions solve the Picasso problem!

Abstract

roup convolutional networks are able
werful representations based on sym-
erns, they lack explicit means to learn
l relationships among them (e.g., rela-
ns and poses). In this paper, we present
roup equivariant convolutions, a gen-
of the group convolution, in which
s applied during the course of convo-
ccentuate meaningful symmetry com-
and suppress non-plausible, mislead-
We indicate that prior work on visual
an be described as special cases of
ed framework and show empirically
entive group equivariant convolutional
consistently outperform conventional
olutional networks on benchmark im-

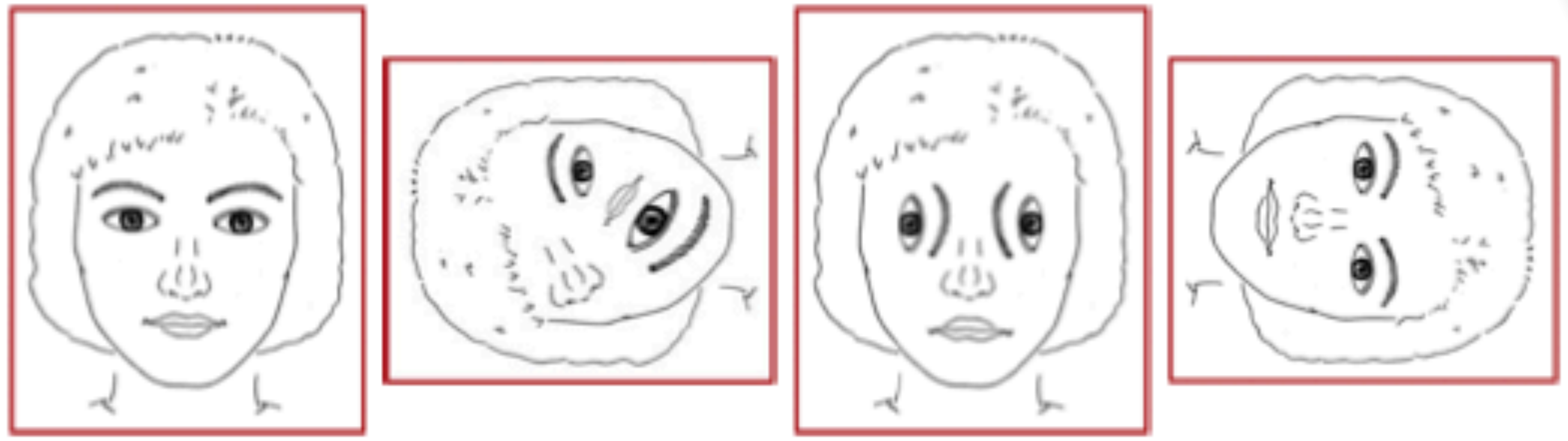


Figure 1. Meaningful relationships among object symmetries. Though every figure is composed by the same elements, only the outermost examples resemble faces. The relative positions, orientations and scales of elements in the innermost examples do not match any meaningful face composition and ... not be labelled as such. Built upon Fig. 1 from Sc...

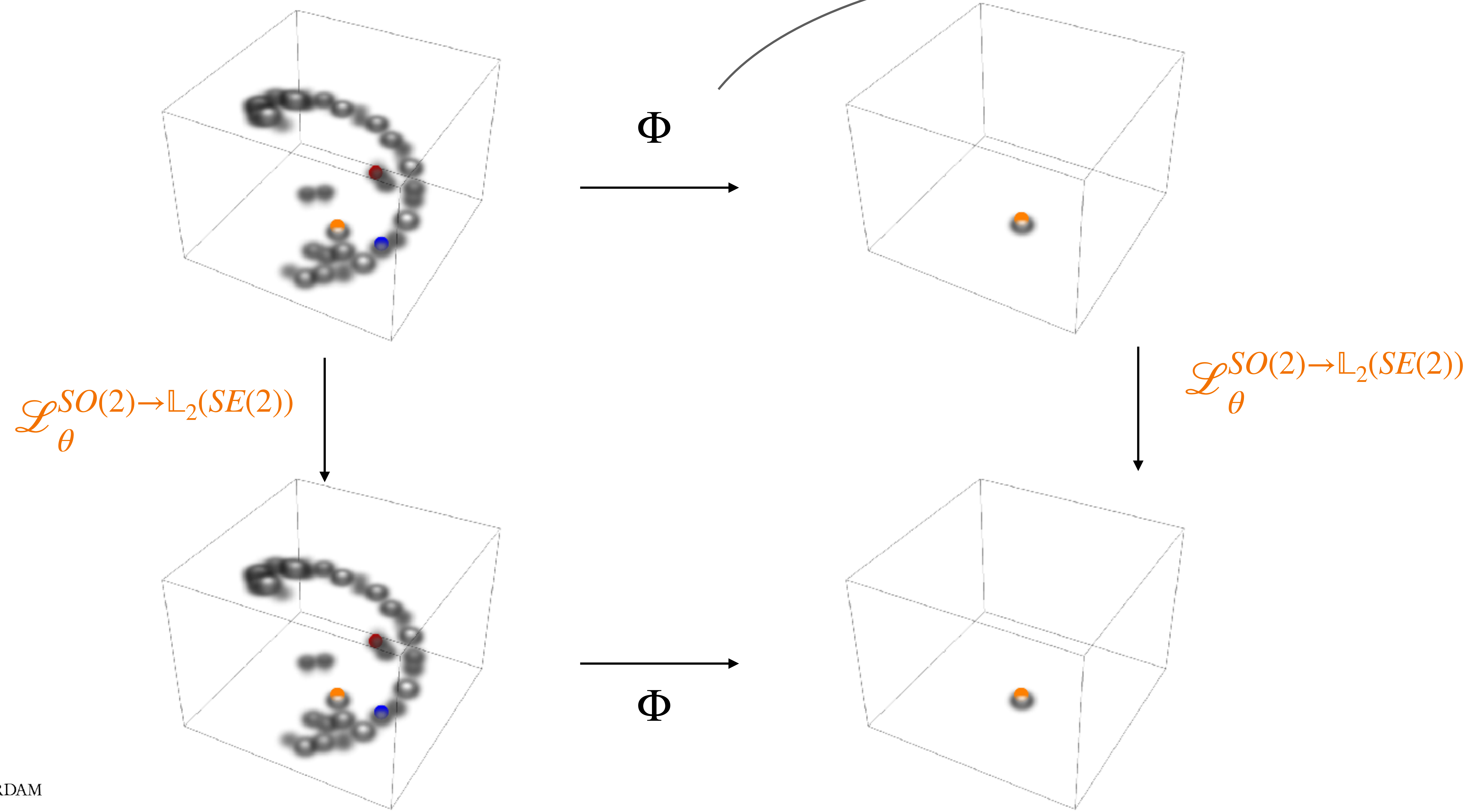
Recognition by components in a group theoretical setting!

cations in location, size, viewpoint, lighting conditions and background (Bruce & Humphreys, 1994). In addition, we do not just recognize them but are able to describe in detail the type and amount of modification applied to them as well

Group **equivariance**

SE(2) group convolutions are roto-translation equivariant

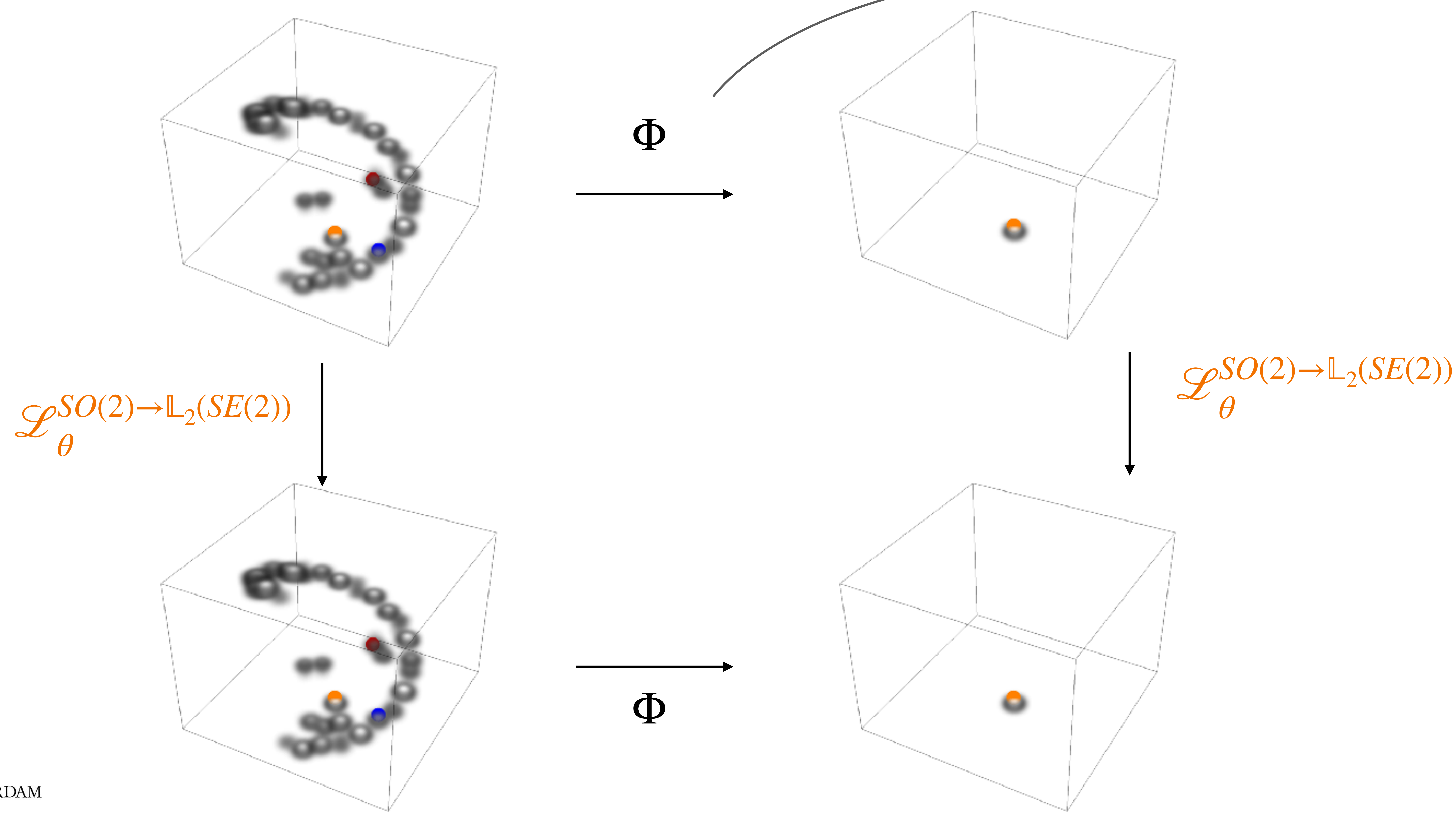
$$(k \star f)(\mathbf{x}) = (\mathscr{L}_{\mathbf{x}}^{\mathbb{R}^2 \to \mathbb{L}_2(SE(2))} \mathscr{L}_{\theta}^{SO(2) \to \mathbb{L}_2(SE(2))} k \, , f)_{\mathbb{L}_2(SE(2))}$$



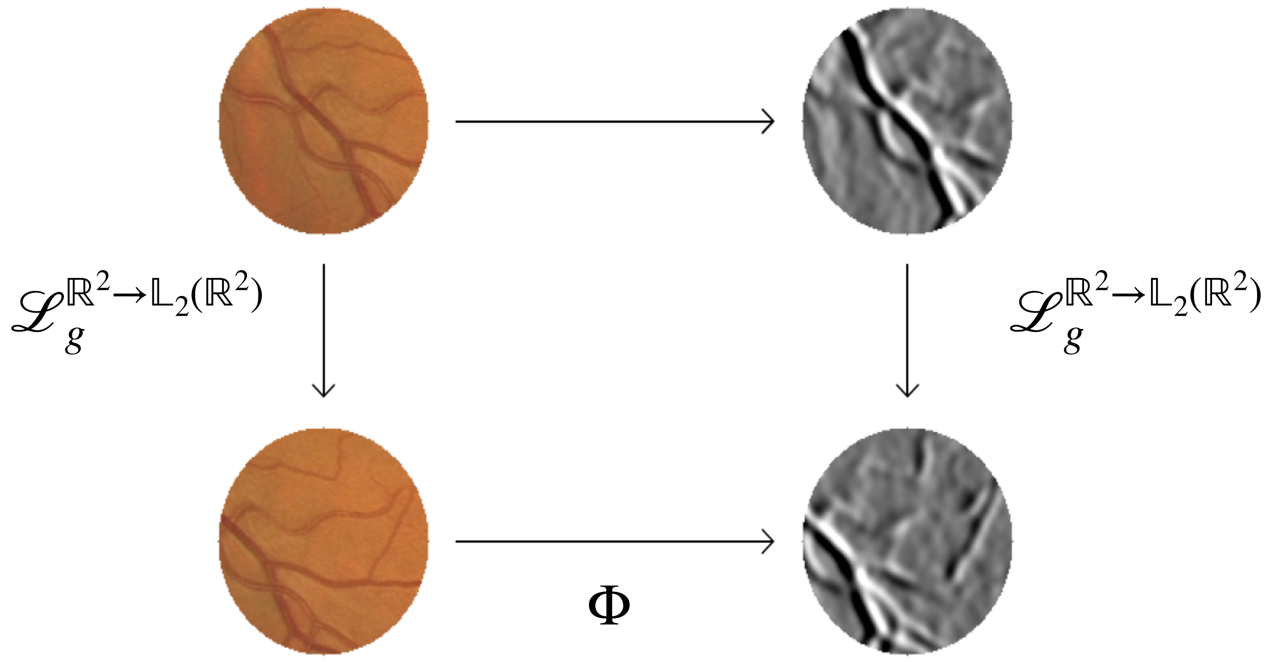
Group **equivariance**

SE(2) group convolutions are roto-translation equivariant

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



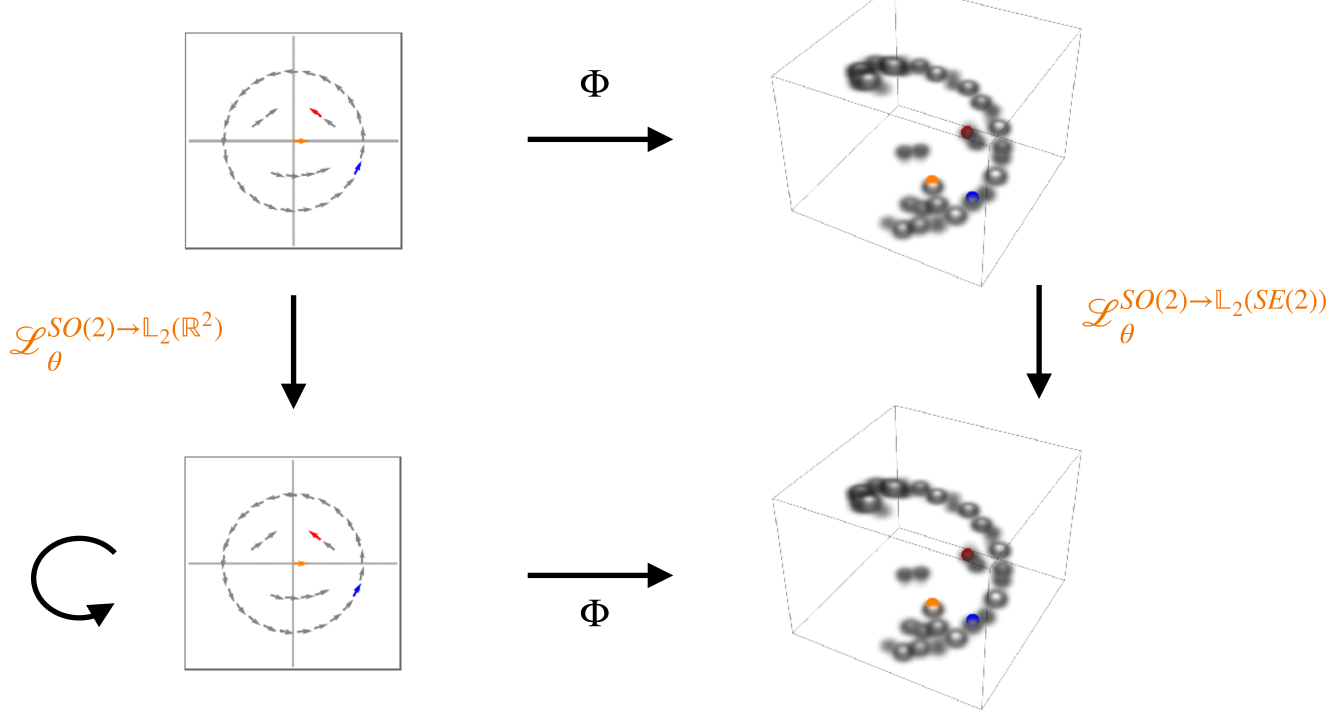
2D cross-correlation (translation equivariant)



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

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

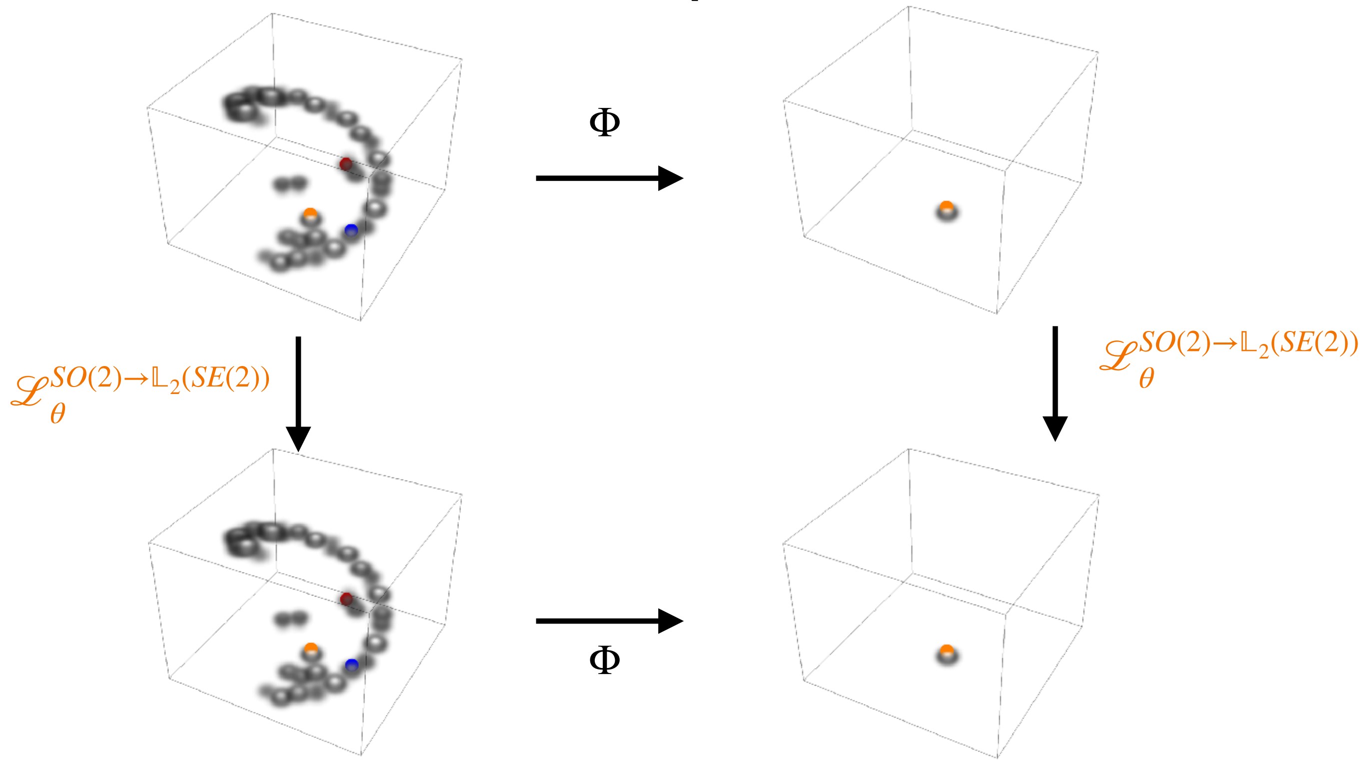
SE(2) lifting correlations (roto-translation equivariant)



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

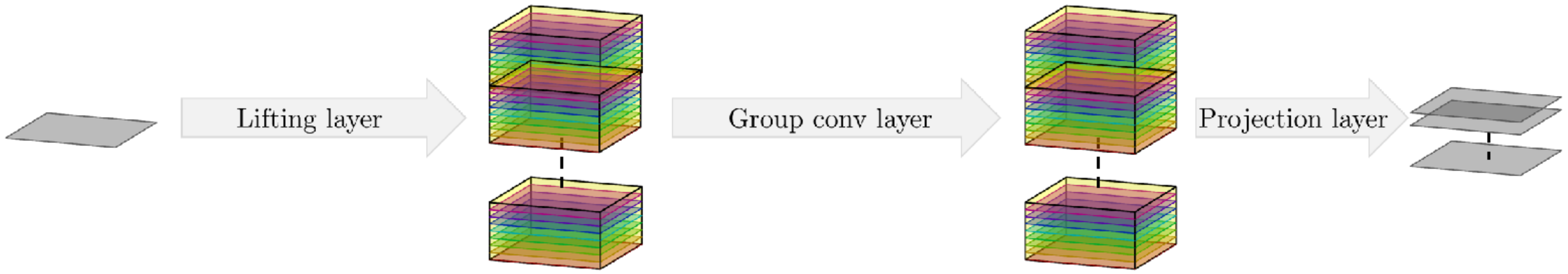
$$= \int_{\mathbb{R}^2} k(\mathbf{R}_\theta^{-1}(\mathbf{x}' - \mathbf{x})) f(\mathbf{x}') d\mathbf{x}'$$

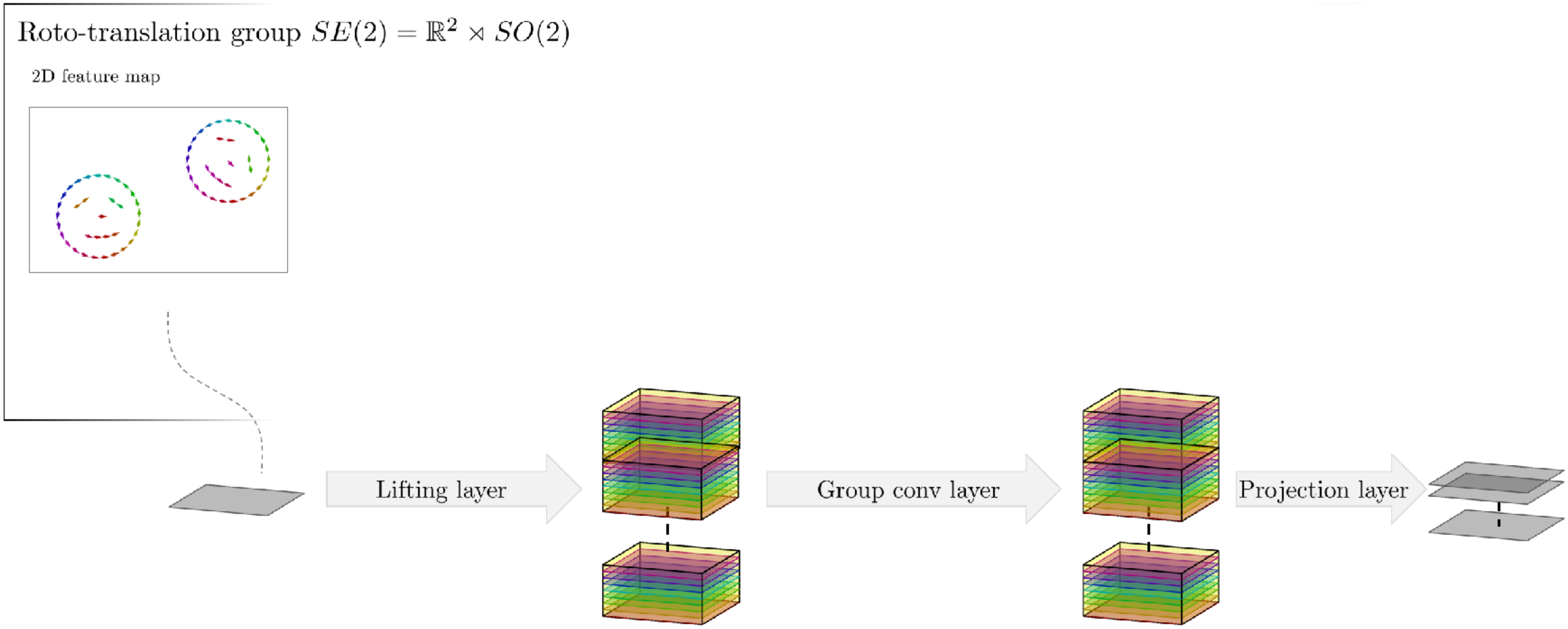
SE(2) G-correlations (roto-translation equivariant)

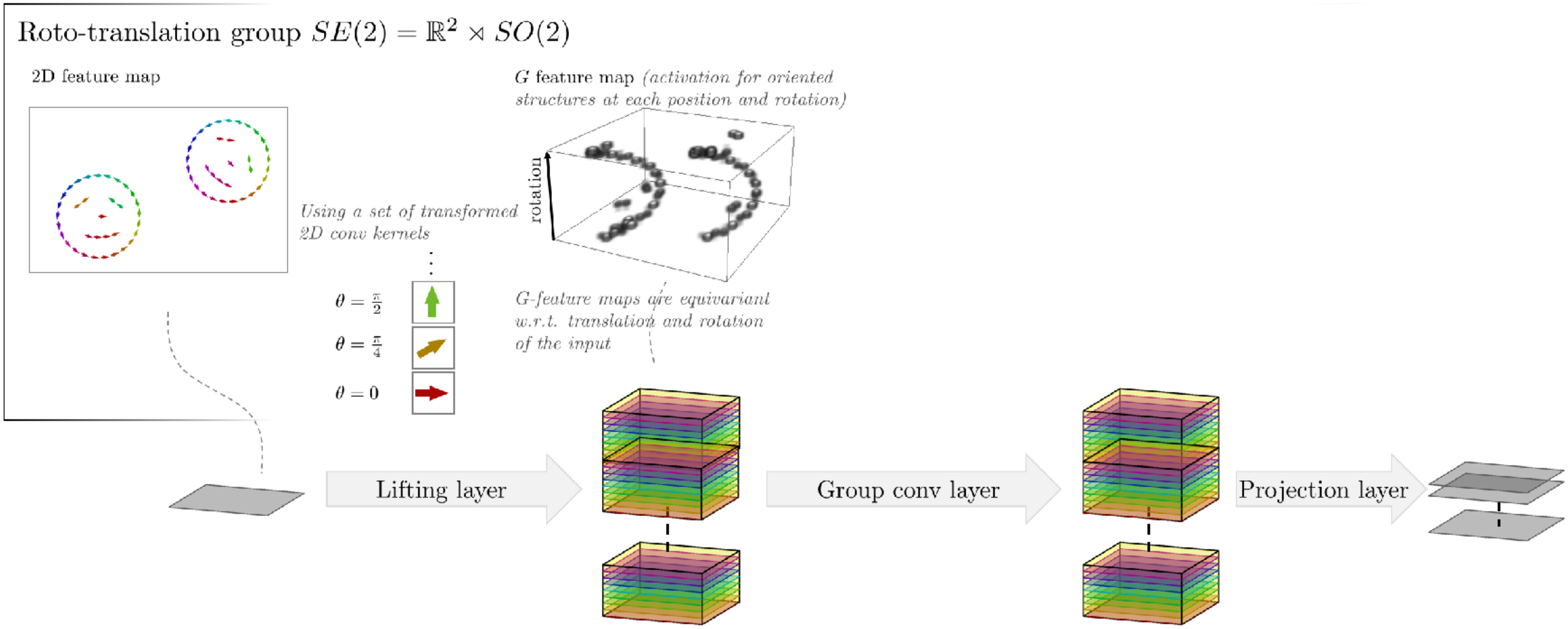


$$(k \star f)(\mathbf{x}) = (\mathscr{L}_g^{SE(2)\to\mathbb{L}_2(SE(2))} k , f)_{\mathbb{L}_2(SE(2)}$$

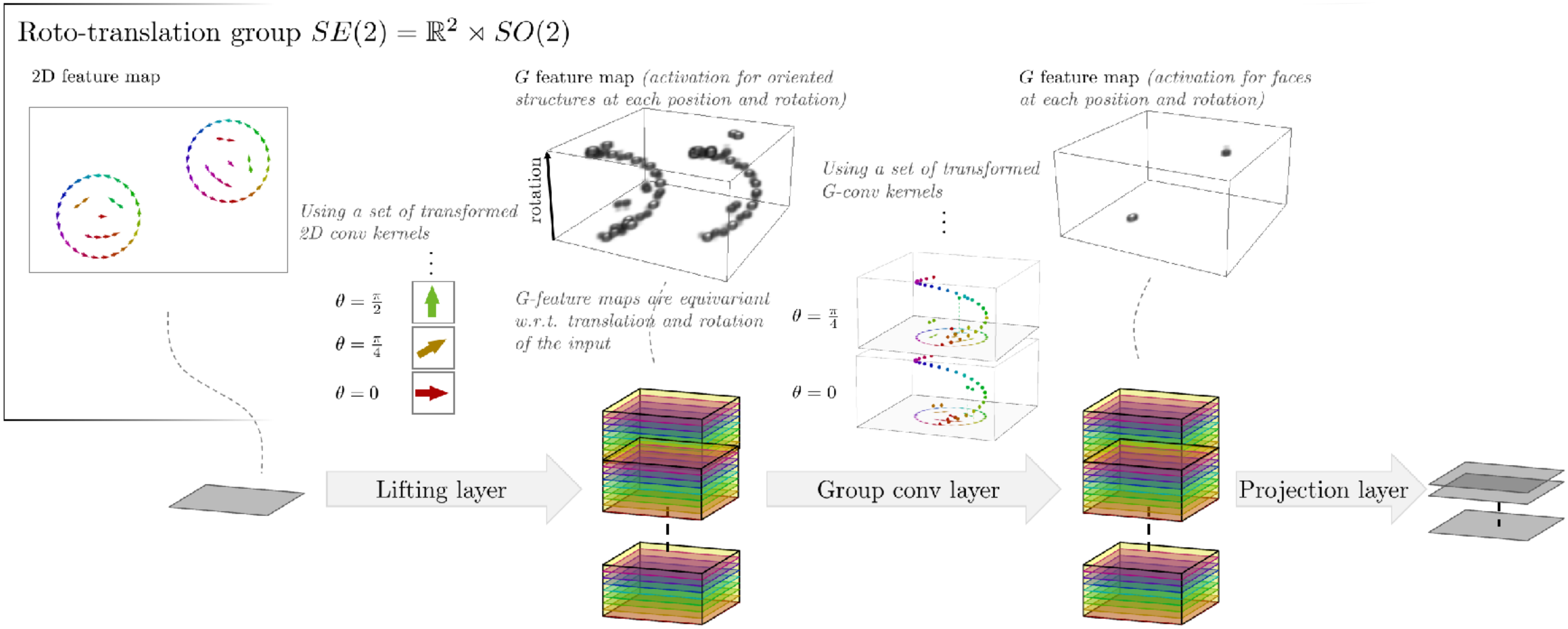
$$= \int_{\mathbb{R}^2} \int_{S^1} k(\mathbf{R}_\theta^{-1}(\mathbf{x}' - \mathbf{x}), \theta' - \theta \mod 2\pi) f(\mathbf{x}', \theta') d\mathbf{x}'$$


Lifting layer
Group conv layer
Projection layer

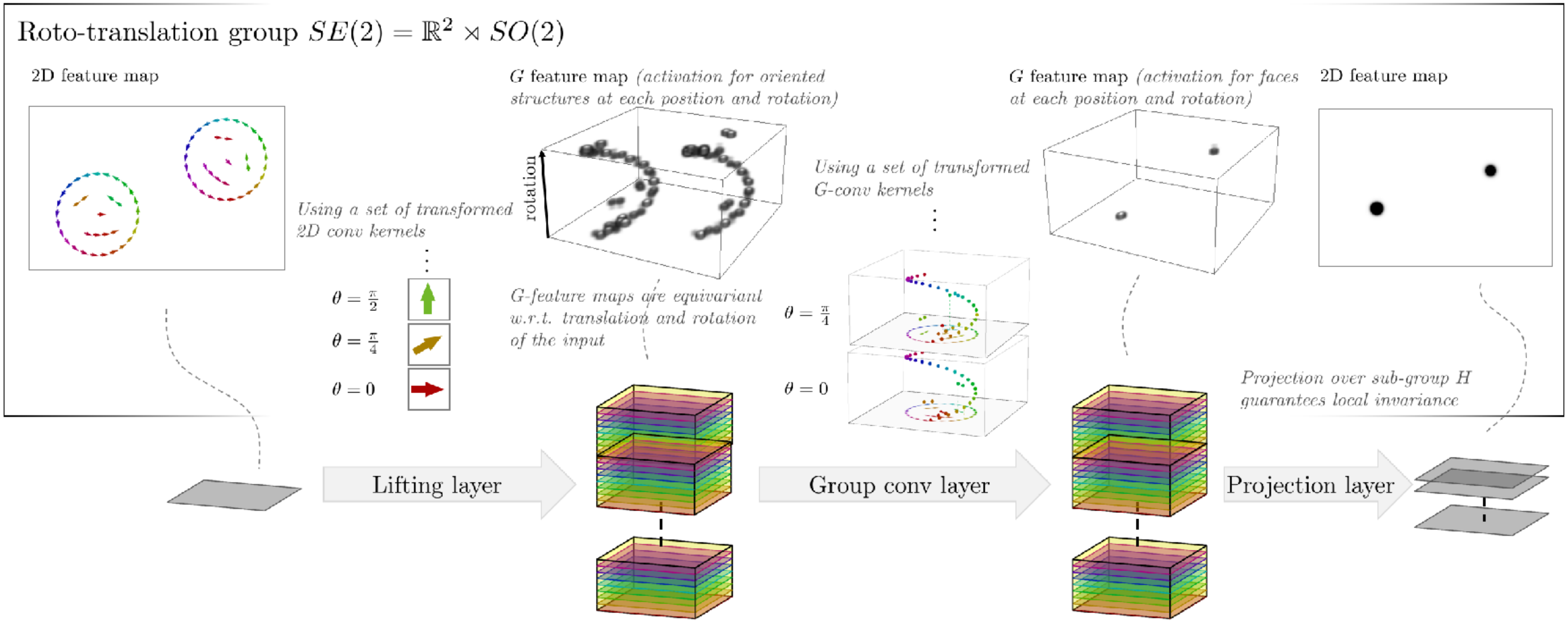

Roto-translation group $SE(2) = \mathbb{R}^2 \rtimes SO(2)$
2D feature map
Lifting layer
Group conv layer
Projection layer



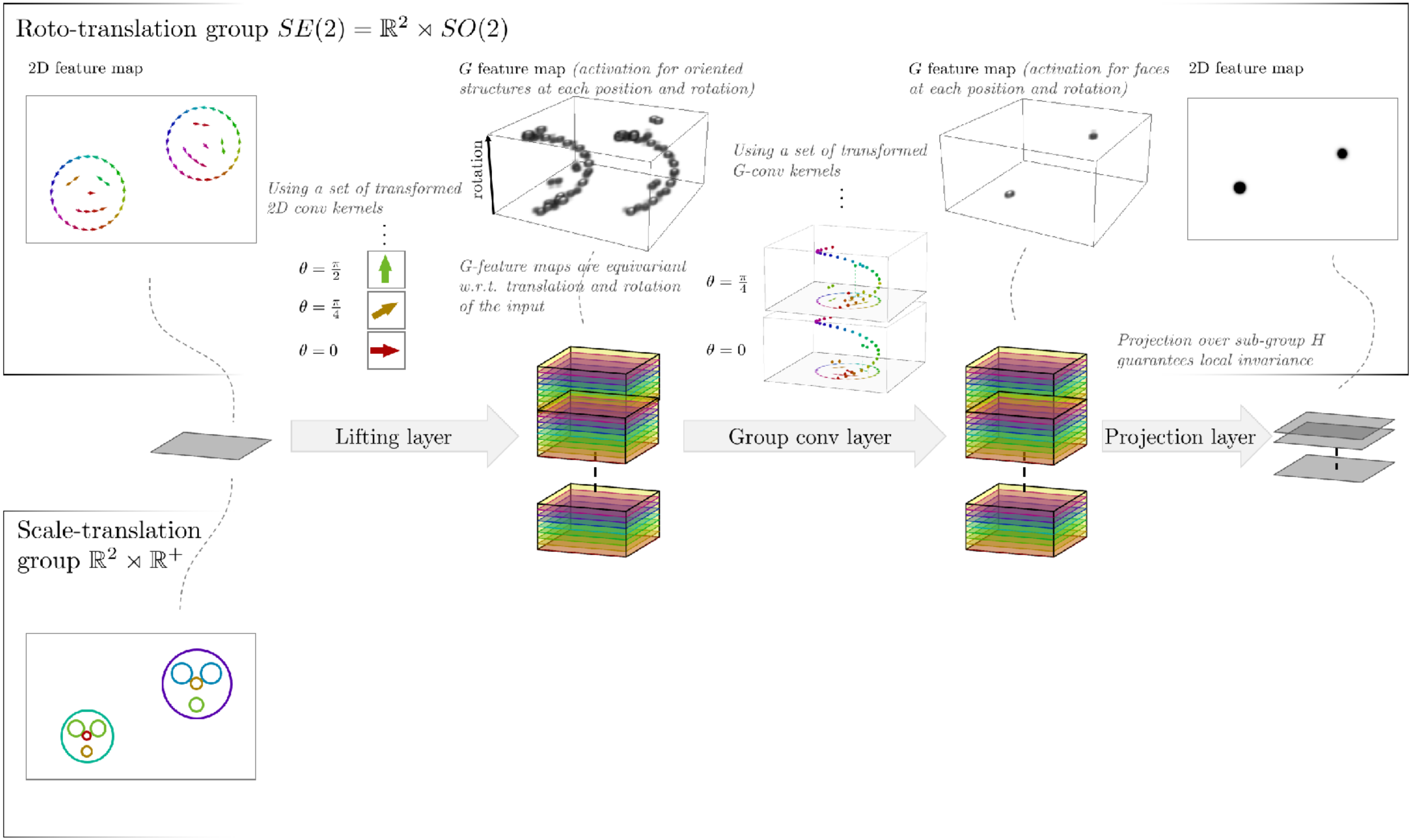
Roto-translation group $SE(2) = \mathbb{R}^2 \rtimes SO(2)$
2D feature map
G feature map *(activation for oriented structures at each position and rotation)*
rotation
Using a set of transformed 2D conv kernels
$\theta = \frac{\pi}{2}$
$\theta = \frac{\pi}{4}$
$\theta = 0$
G-feature maps are equivariant w.r.t. translation and rotation of the input
Lifting layer
Group conv layer
Projection layer



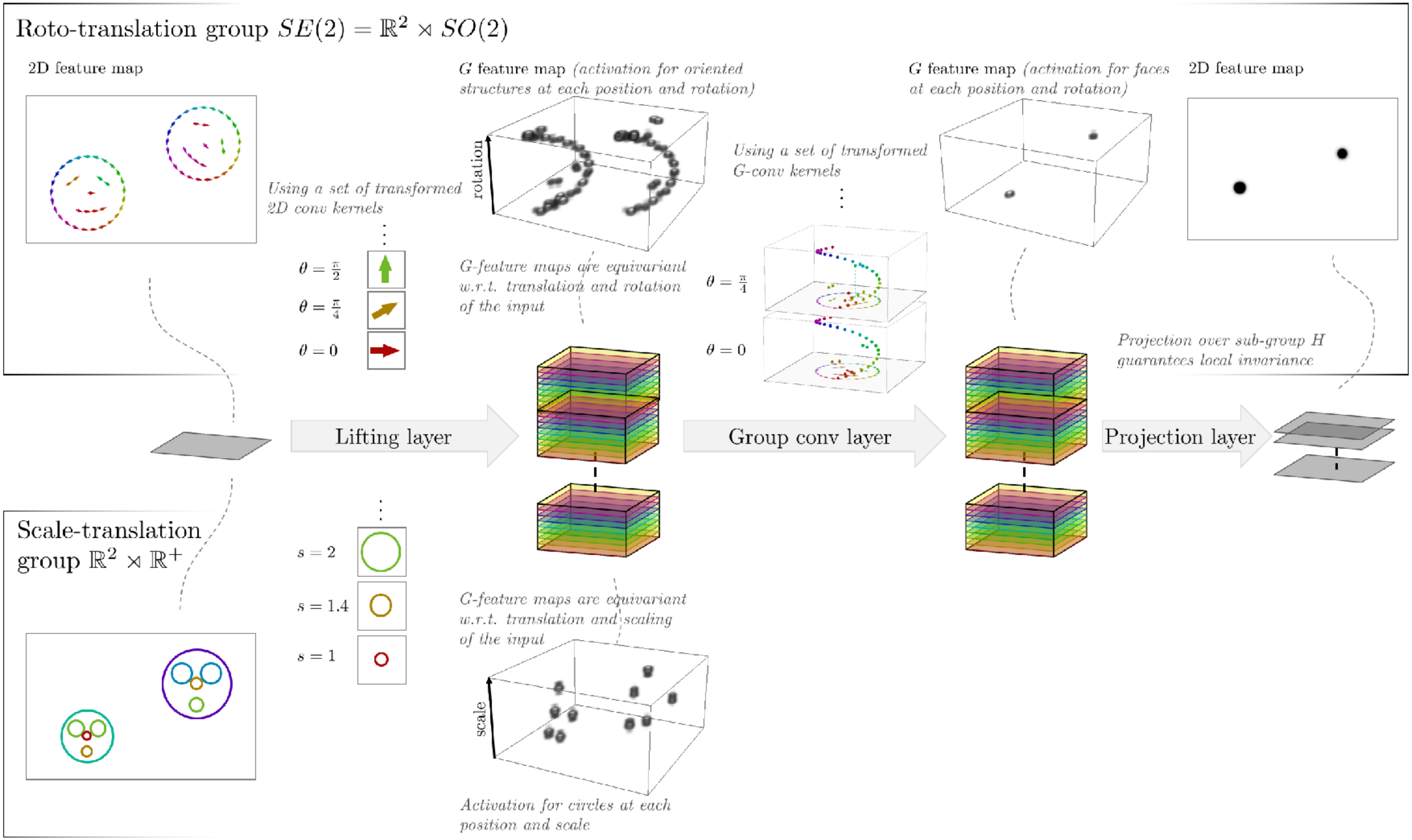
Roto-translation group $SE(2) = \mathbb{R}^2 \rtimes SO(2)$
2D feature map
Using a set of transformed 2D conv kernels
$\theta = \frac{\pi}{2}$
$\theta = \frac{\pi}{4}$
$\theta = 0$
G feature map (activation for oriented structures at each position and rotation)
rotation
G-feature maps are equivariant w.r.t. translation and rotation of the input
Using a set of transformed G-conv kernels
$\theta = \frac{\pi}{4}$
$\theta = 0$
G feature map (activation for faces at each position and rotation)
Lifting layer
Group conv layer
Projection layer



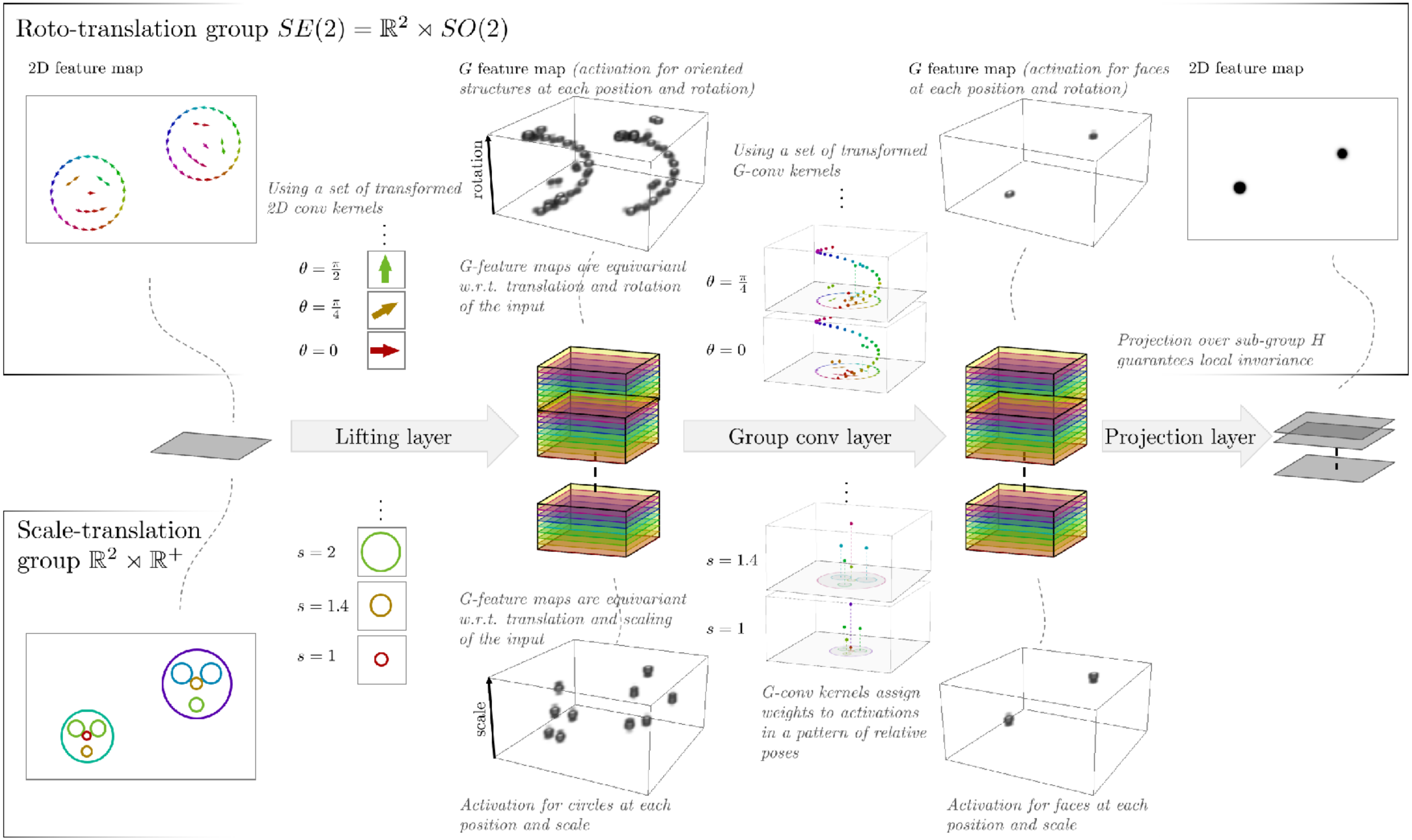
Roto-translation group $SE(2) = \mathbb{R}^2 \rtimes SO(2)$
2D feature map
Using a set of transformed 2D conv kernels
$\theta = \frac{\pi}{2}$
$\theta = \frac{\pi}{4}$
$\theta = 0$
G feature map (activation for oriented structures at each position and rotation)
rotation
G-feature maps are equivariant w.r.t. translation and rotation of the input
Using a set of transformed G-conv kernels
$\theta = \frac{\pi}{4}$
$\theta = 0$
G feature map (activation for faces at each position and rotation)
2D feature map
Projection over sub-group H guarantees local invariance
Lifting layer
Group conv layer
Projection layer



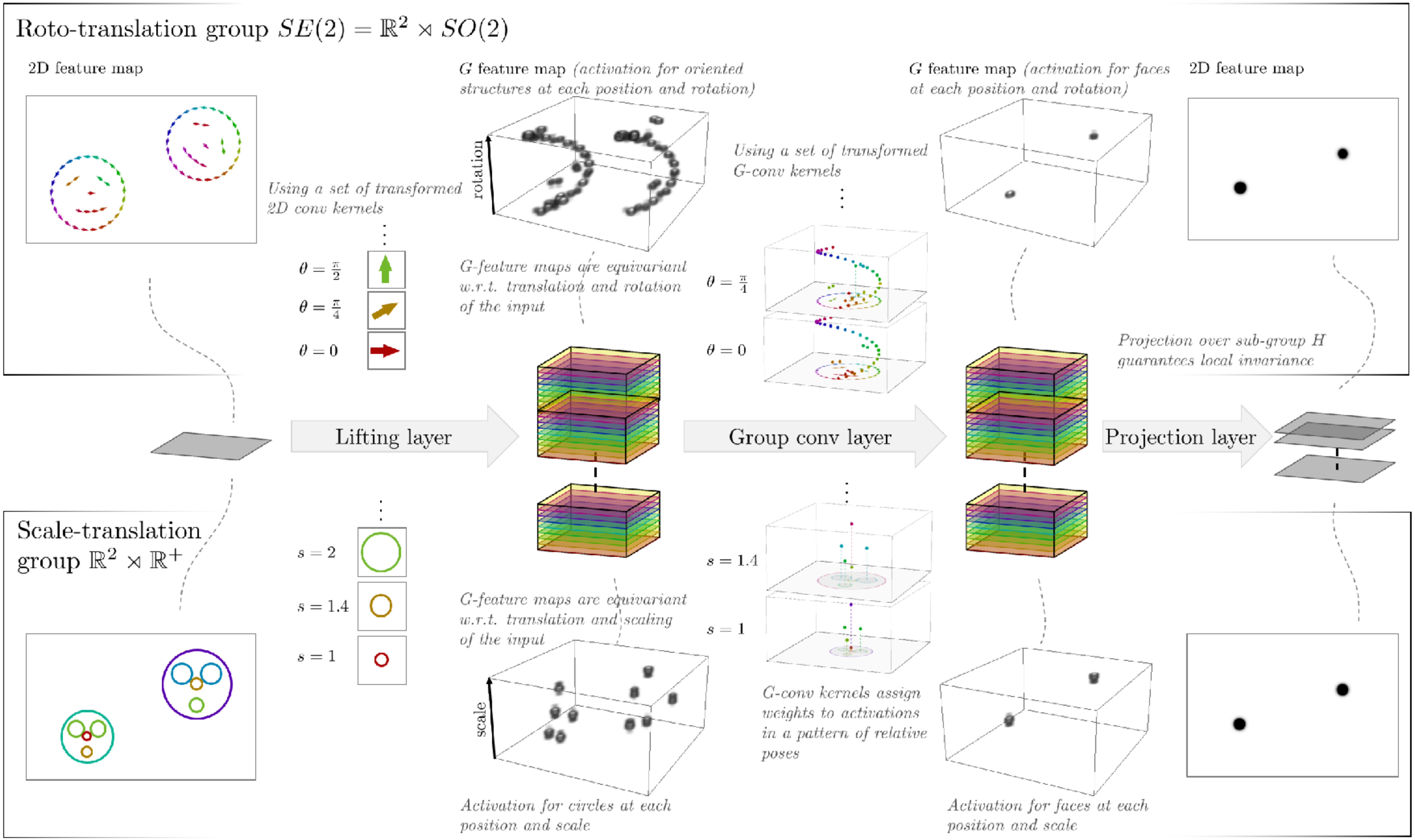
Roto-translation group $SE(2) = \mathbb{R}^2 \rtimes SO(2)$
2D feature map
Using a set of transformed 2D conv kernels
$\theta = \frac{\pi}{2}$
$\theta = \frac{\pi}{4}$
$\theta = 0$
G feature map (activation for oriented structures at each position and rotation)
rotation
G-feature maps are equivariant w.r.t. translation and rotation of the input
Using a set of transformed G-conv kernels
$\theta = \frac{\pi}{4}$
$\theta = 0$
G feature map (activation for faces at each position and rotation)
2D feature map
Projection over sub-group H guarantees local invariance
Lifting layer
Group conv layer
Projection layer
Scale-translation group $\mathbb{R}^2 \rtimes \mathbb{R}^+$



Roto-translation group $SE(2) = \mathbb{R}^2 \rtimes SO(2)$
2D feature map
Using a set of transformed 2D conv kernels
$\theta = \frac{\pi}{2}$
$\theta = \frac{\pi}{4}$
$\theta = 0$
G feature map (activation for oriented structures at each position and rotation)
rotation
G-feature maps are equivariant w.r.t. translation and rotation of the input
Using a set of transformed G-conv kernels
$\theta = \frac{\pi}{4}$
$\theta = 0$
G feature map (activation for faces at each position and rotation)
2D feature map
Projection over sub-group H guarantees local invariance
Lifting layer
Group conv layer
Projection layer
Scale-translation group $\mathbb{R}^2 \rtimes \mathbb{R}^+$
$s = 2$
$s = 1.4$
$s = 1$
G-feature maps are equivariant w.r.t. translation and scaling of the input
scale
Activation for circles at each position and scale



Roto-translation group $SE(2) = \mathbb{R}^2 \rtimes SO(2)$
2D feature map
G feature map (activation for oriented structures at each position and rotation)
G feature map (activation for faces at each position and rotation)
2D feature map
rotation
Using a set of transformed 2D conv kernels
Using a set of transformed G-conv kernels
$\theta = \frac{\pi}{2}$
$\theta = \frac{\pi}{4}$
$\theta = 0$
G-feature maps are equivariant w.r.t. translation and rotation of the input
$\theta = \frac{\pi}{4}$
$\theta = 0$
Projection over sub-group H guarantees local invariance
Lifting layer
Group conv layer
Projection layer
Scale-translation group $\mathbb{R}^2 \rtimes \mathbb{R}^+$
$s = 2$
$s = 1.4$
$s = 1$
G-feature maps are equivariant w.r.t. translation and scaling of the input
$s = 1.4$
$s = 1$
scale
G-conv kernels assign weights to activations in a pattern of relative poses
Activation for circles at each position and scale
Activation for faces at each position and scale


Roto-translation group $SE(2) = \mathbb{R}^2 \rtimes SO(2)$
2D feature map
G feature map (activation for oriented structures at each position and rotation)
rotation
Using a set of transformed 2D conv kernels
$\theta = \frac{\pi}{2}$
$\theta = \frac{\pi}{4}$
$\theta = 0$
G-feature maps are equivariant w.r.t. translation and rotation of the input
Using a set of transformed G-conv kernels
$\theta = \frac{\pi}{4}$
$\theta = 0$
G feature map (activation for faces at each position and rotation)
2D feature map
Projection over sub-group H guarantees local invariance
Lifting layer
Group conv layer
Projection layer
Scale-translation group $\mathbb{R}^2 \rtimes \mathbb{R}^+$
$s = 2$
$s = 1.4$
$s = 1$
G-feature maps are equivariant w.r.t. translation and scaling of the input
scale
Activation for circles at each position and scale
$s = 1.4$
$s = 1$
G-conv kernels assign weights to activations in a pattern of relative poses
Activation for faces at each position and scale

Content

Part I: Introduction to group convolutions

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

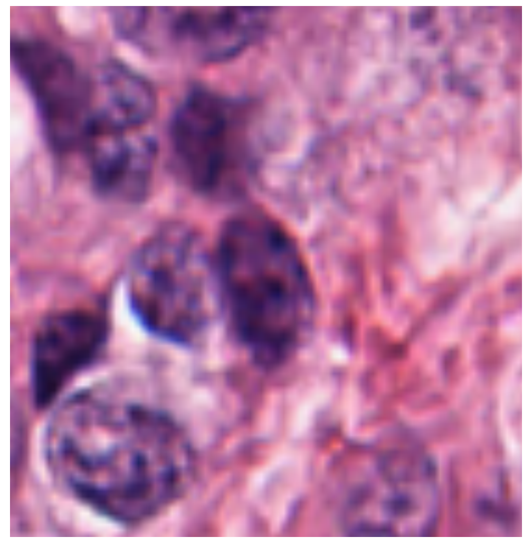
Part II: Group convolutions are all you need

* Theorem: Equivariant linear operators are group convolutions
* Characterization of types of group equivariant layers

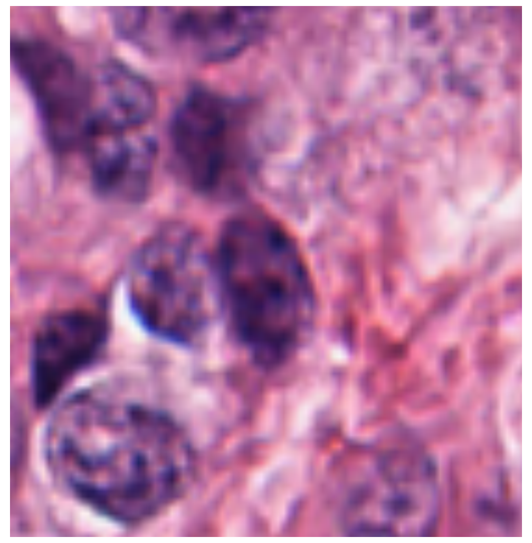
Part III: Steerable group convolutions for molecular data and the N-body problem

* Graph Neural networks
* Irreducible representations, steerable operators and vector spaces
* Steerable Graph NNs (Point Convolutions)

Architecture for rotation invariant mitotic cell detection

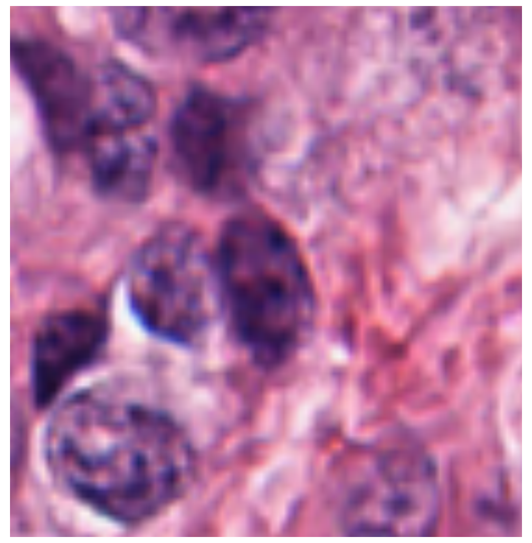


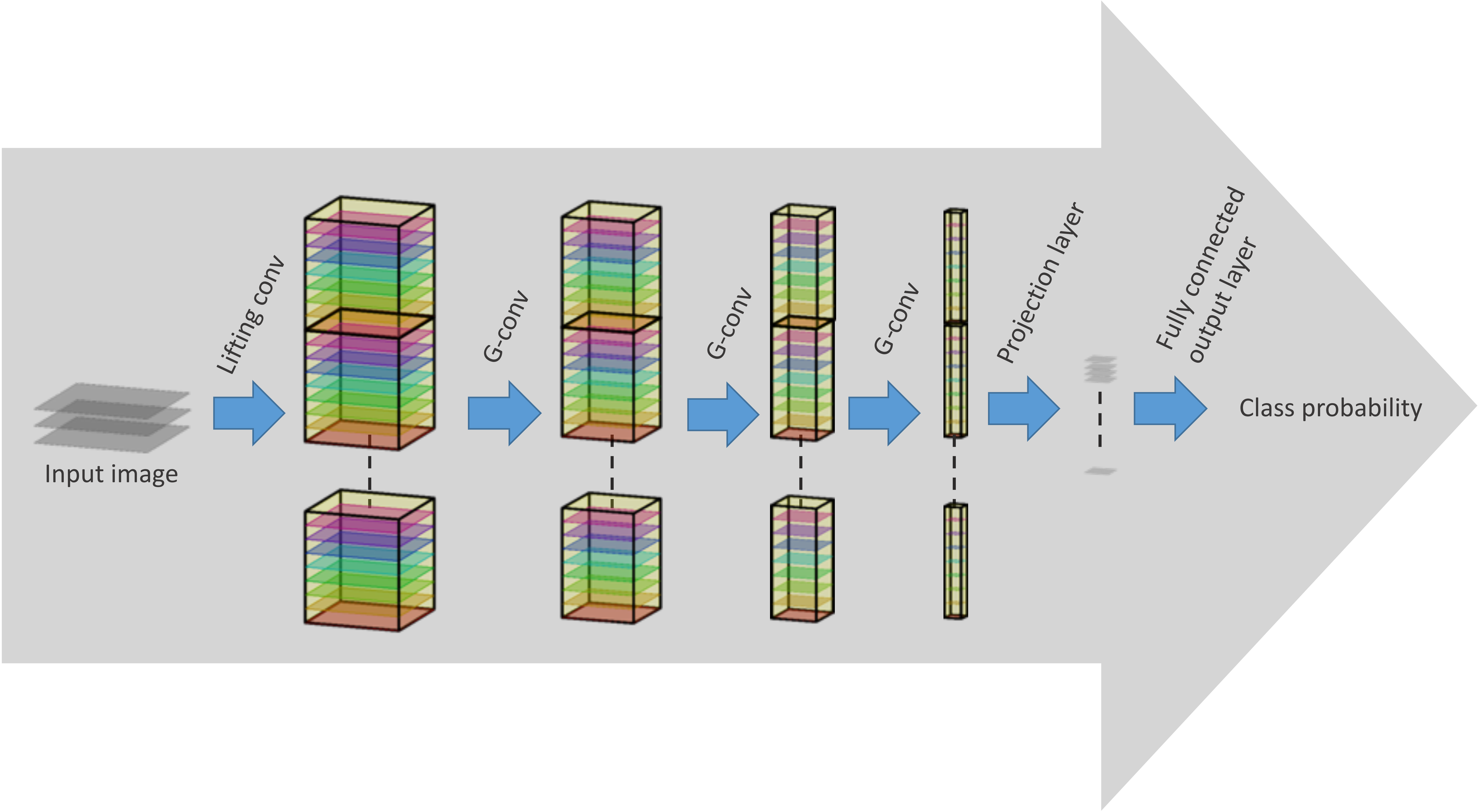
Architecture for rotation invariant mitotic cell detection



“normal” (0)
vs
“mitotic” (1)

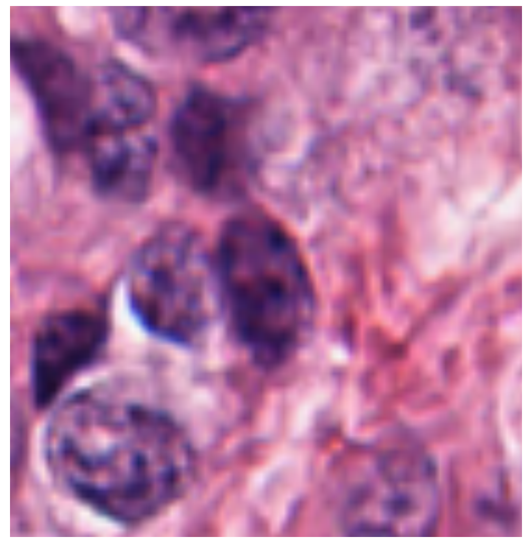
Architecture for rotation invariant mitotic cell detection

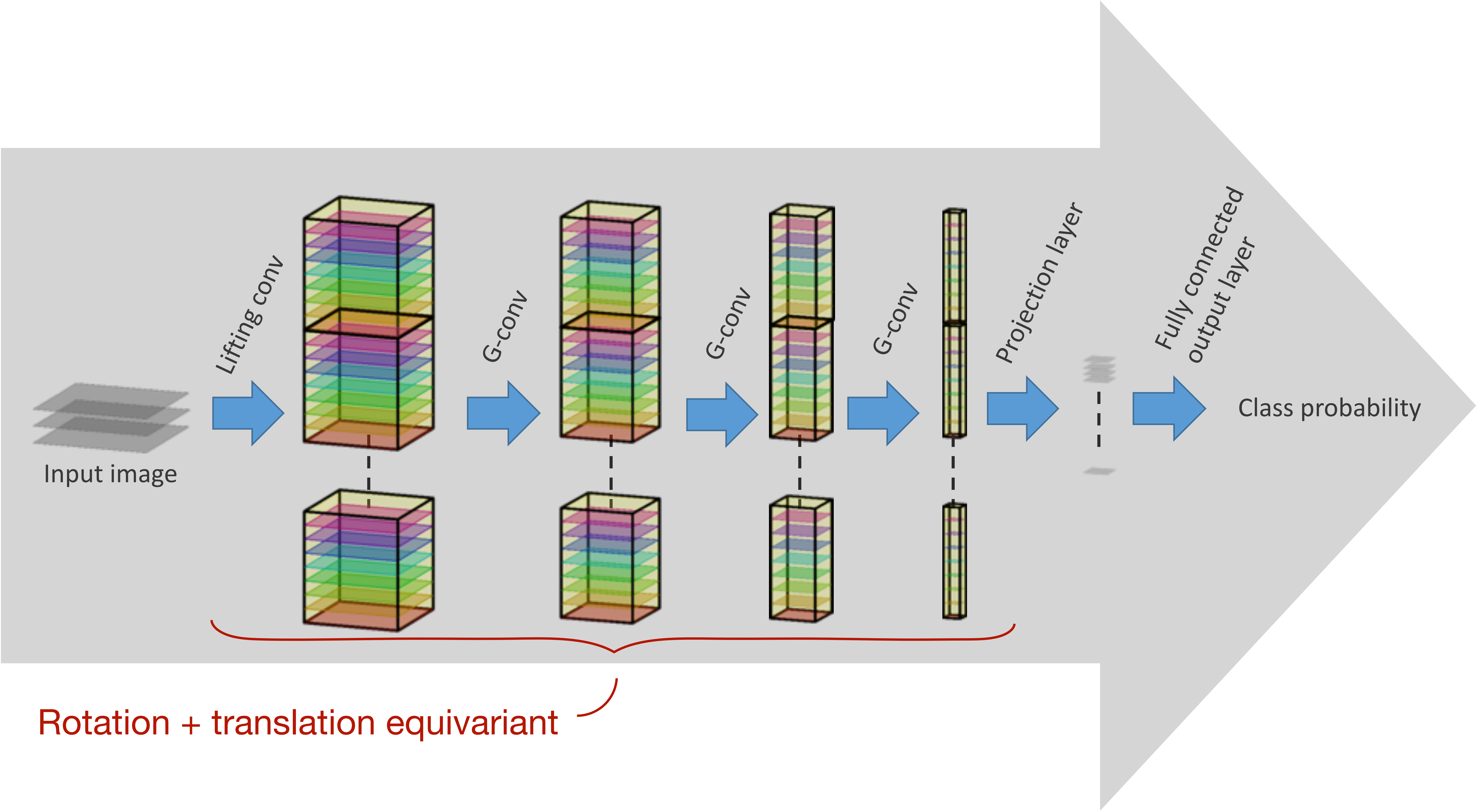




“normal” (0)
vs
“mitotic” (1)

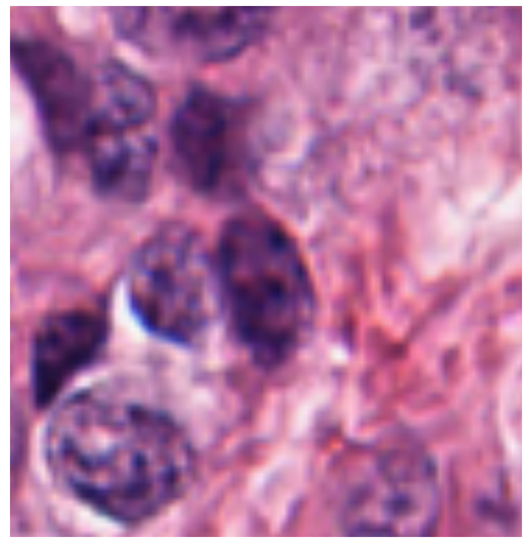
Architecture for rotation invariant mitotic cell detection

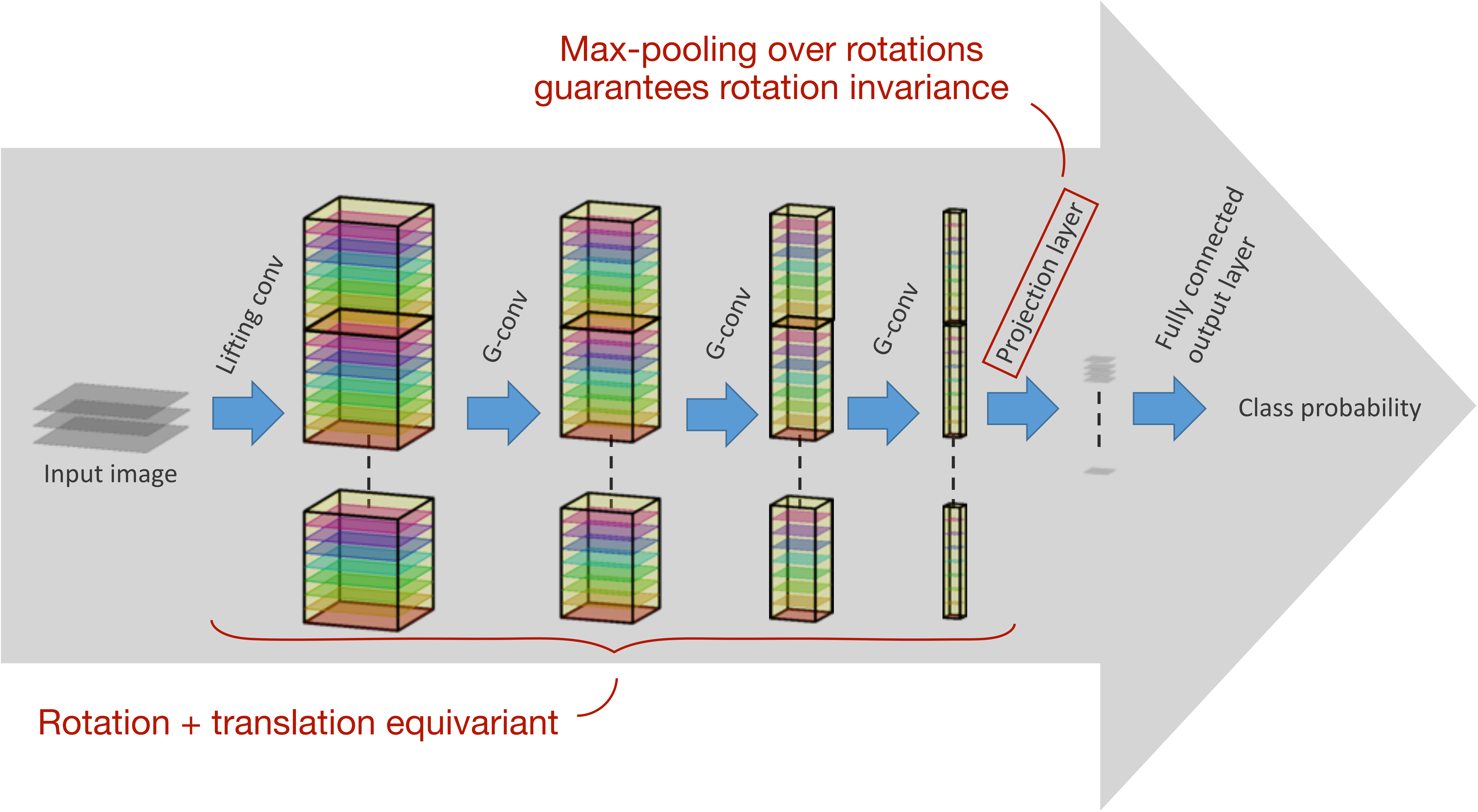




"normal" (0)
vs
"mitotic" (1)

Architecture for rotation invariant mitotic cell detection

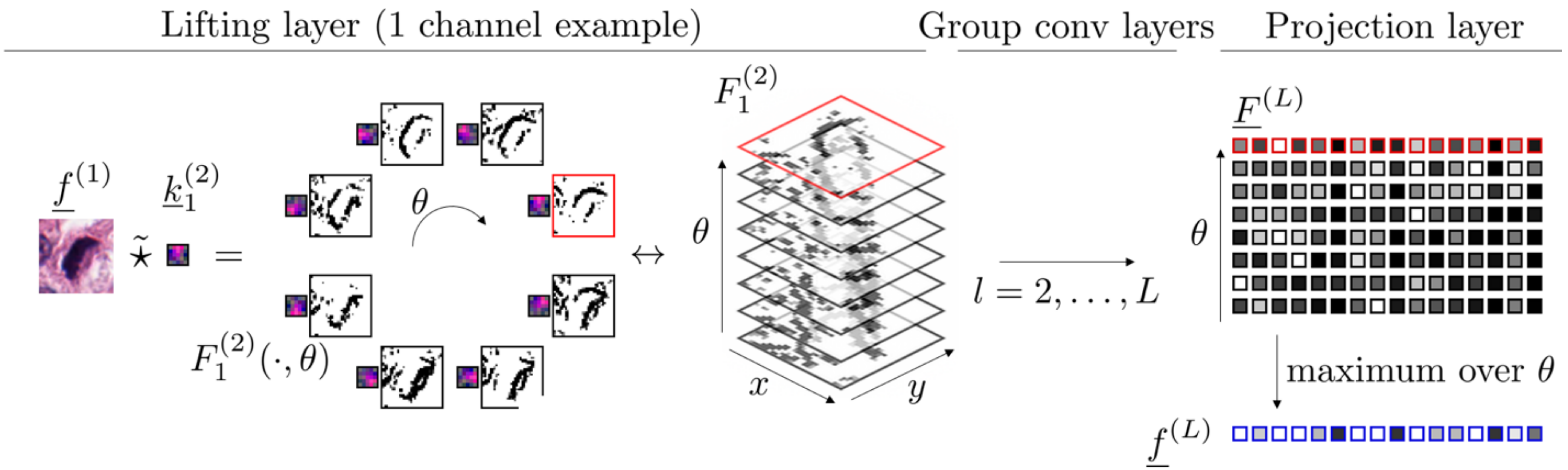




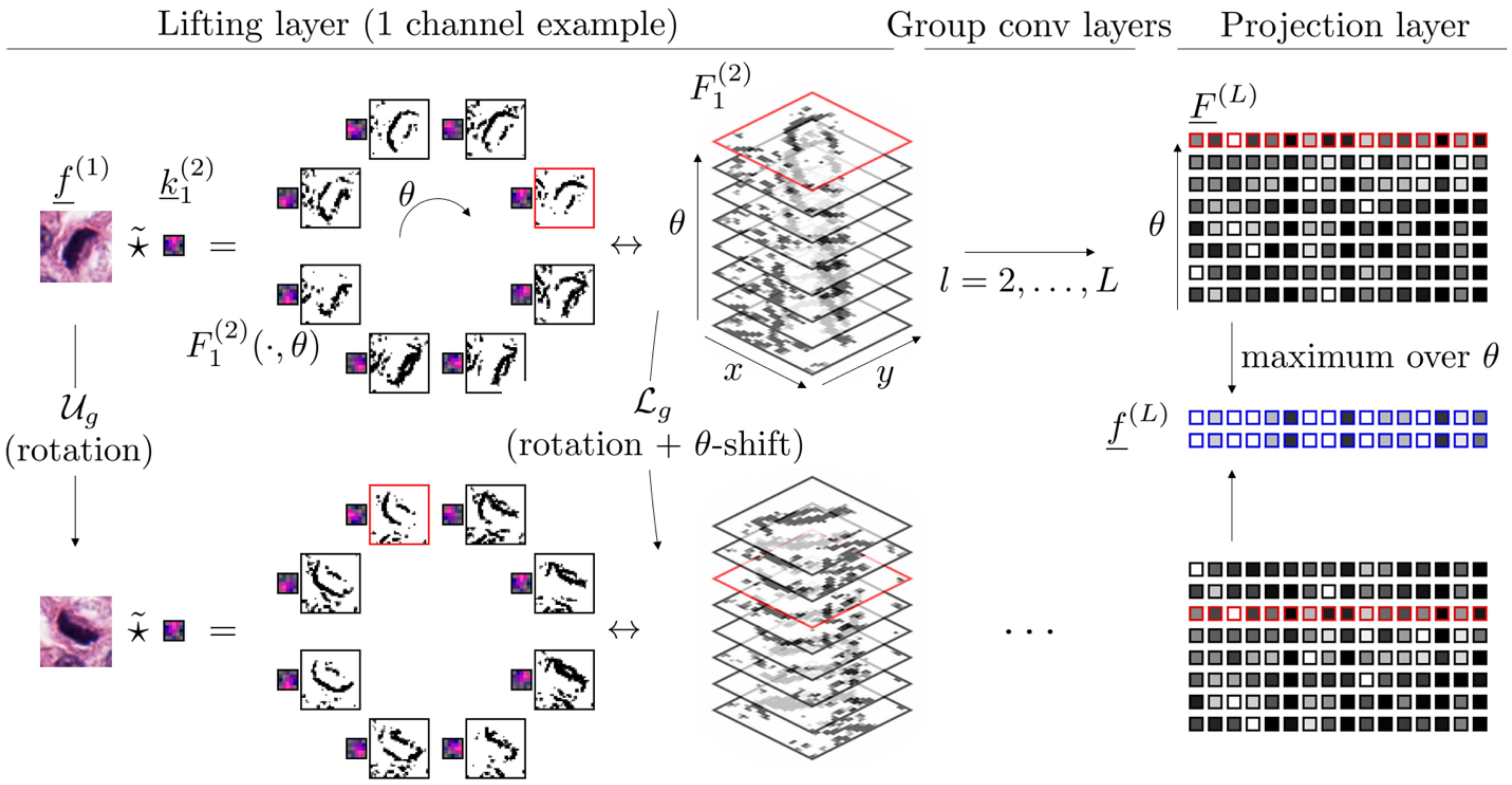
“normal” (0)
vs
“mitotic” (1)

Bekkers & Lafarge et al. MICCAI 2018

Architecture for rotation invariant mitotic cell detection

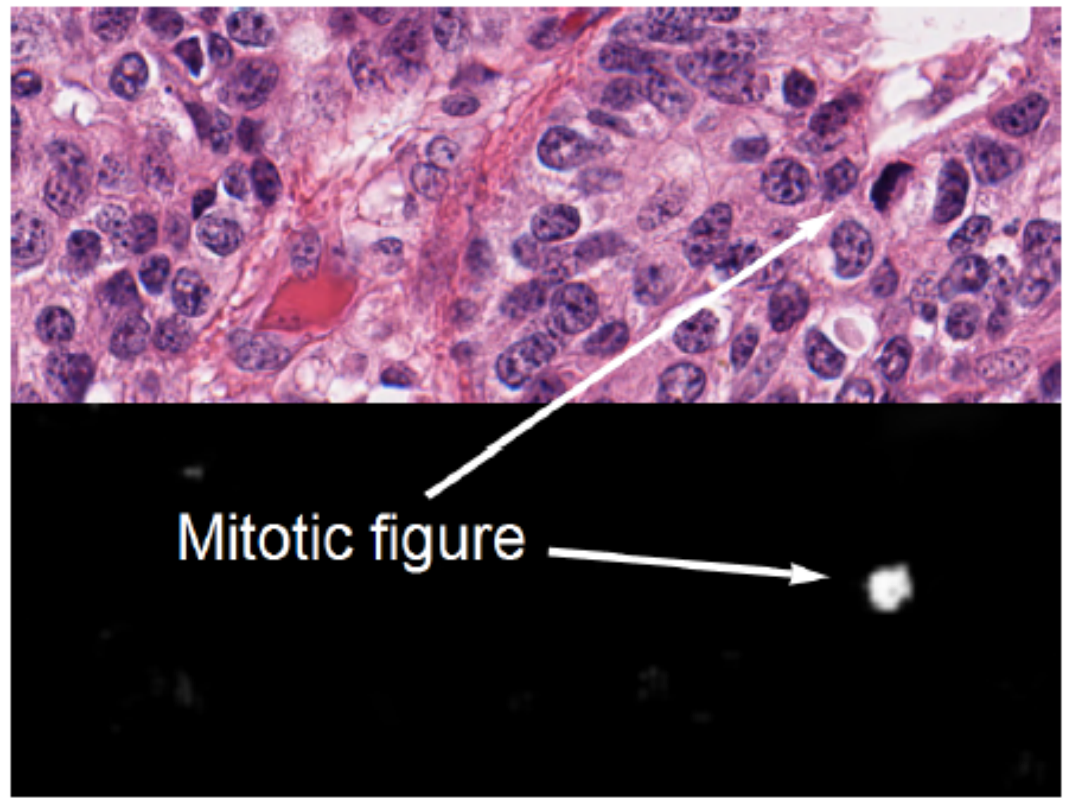


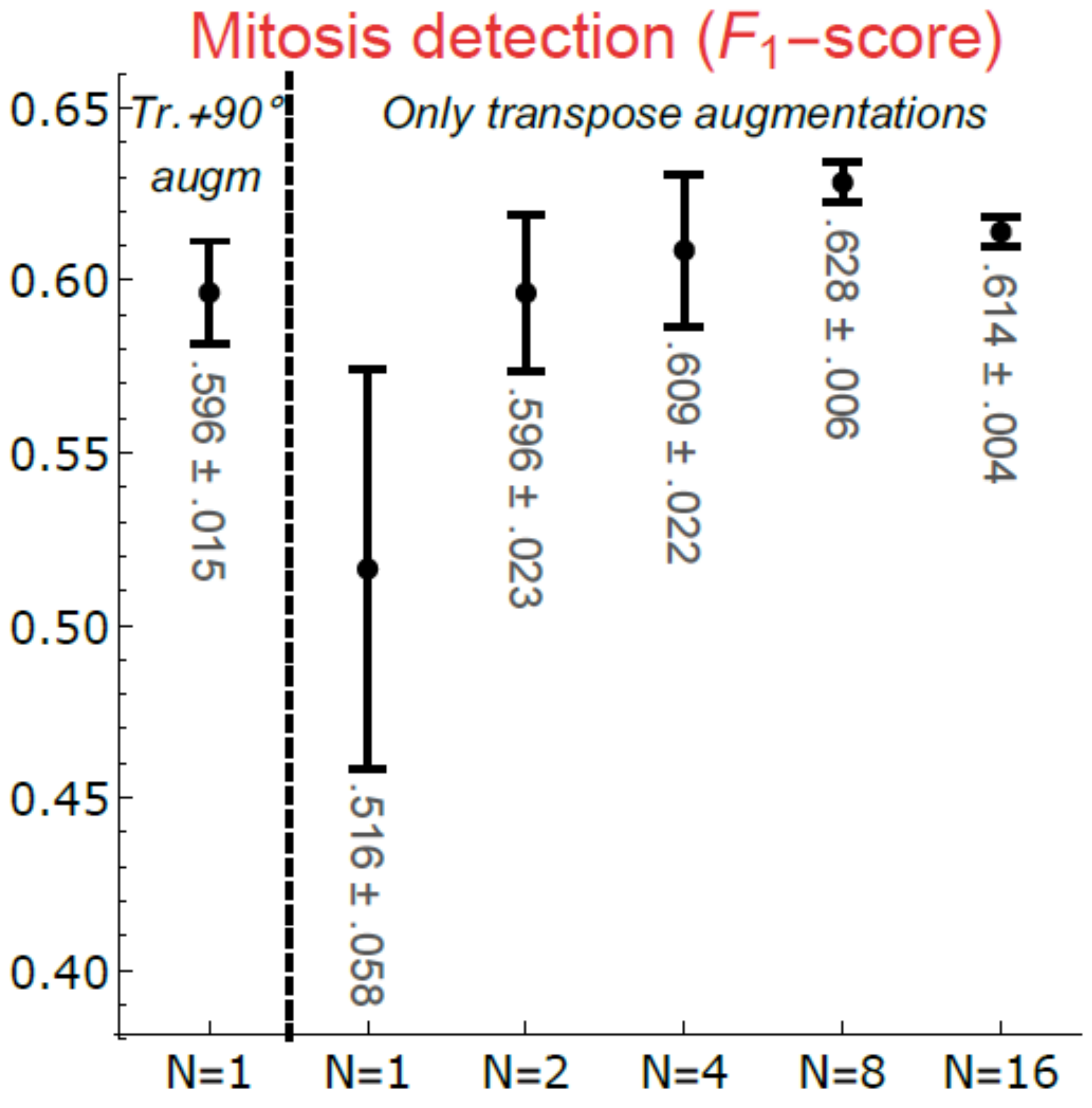
Architecture for rotation invariant mitotic cell detection



Architecture for rotation invariant mitotic cell detection

Bekkers & Lafarge et al. MICCAI 2018

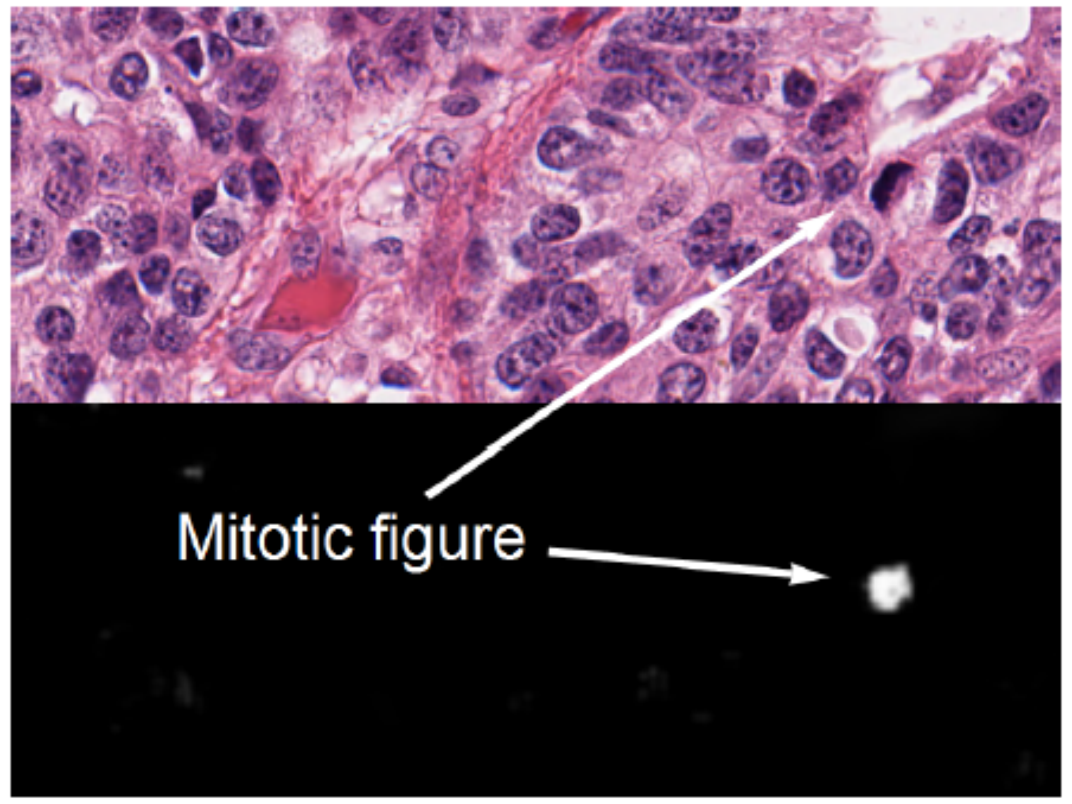


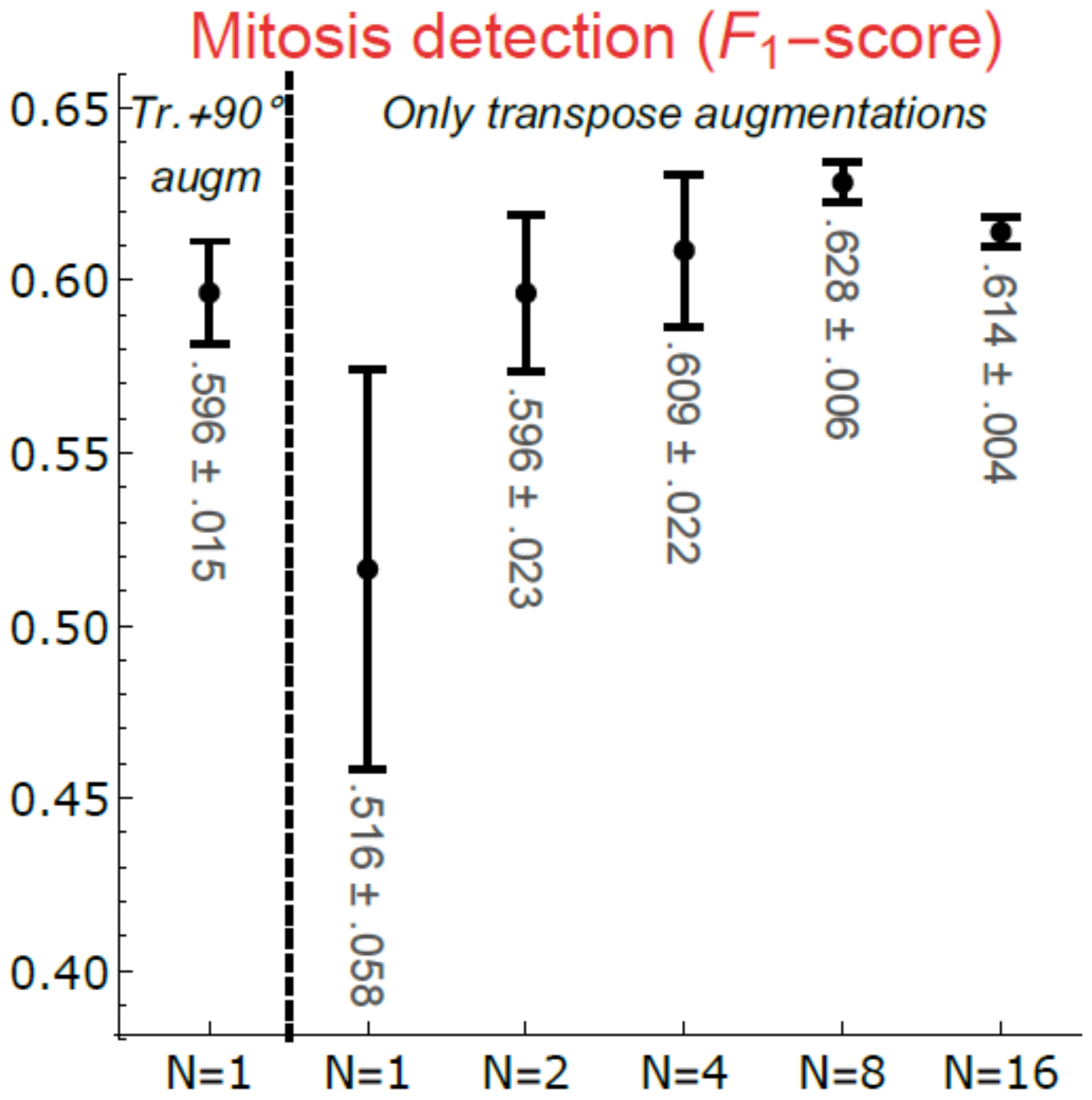


G-CNNs without data-augmentation
outperform
CNNs with data-augmentation

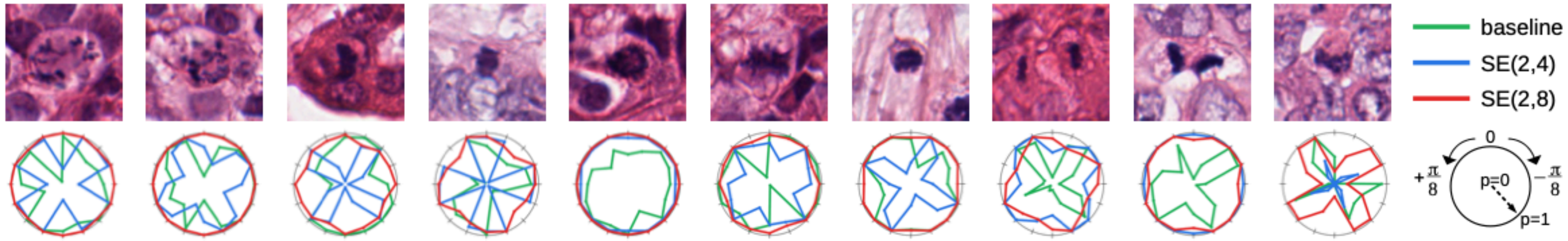
Architecture for rotation invariant mitotic cell detection

Bekkers & Lafarge et al. MICCAI 2018





Lafarge et al. MedIA 2020

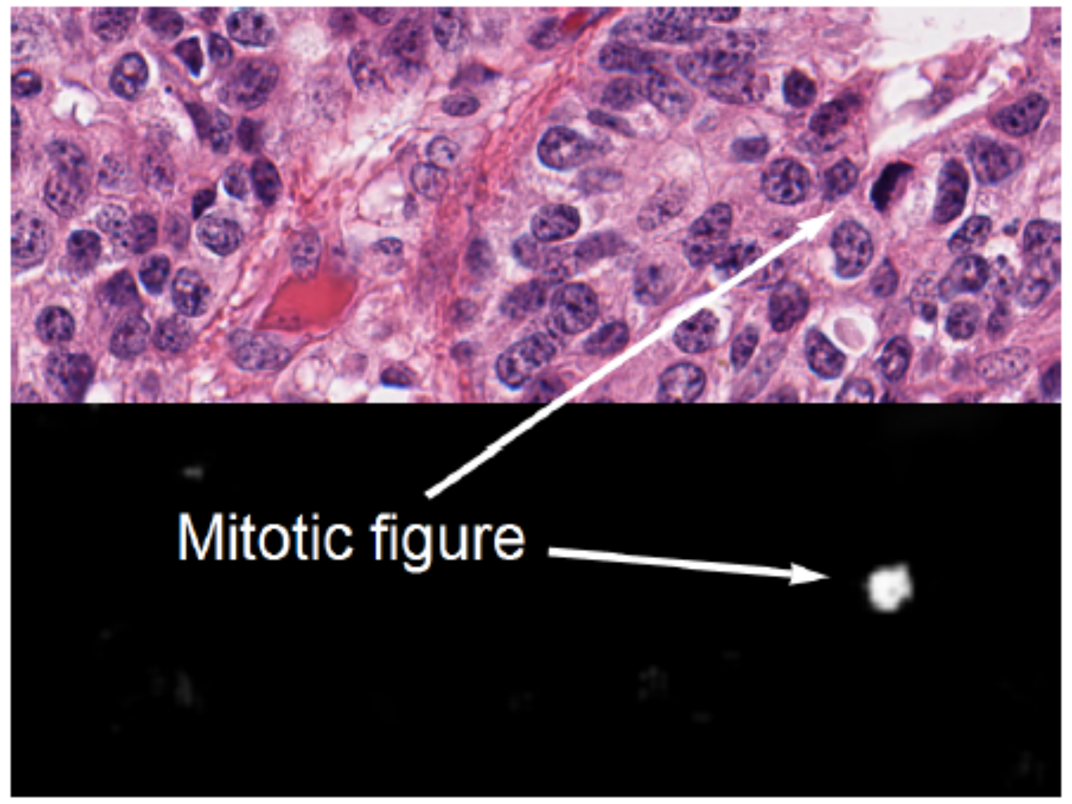


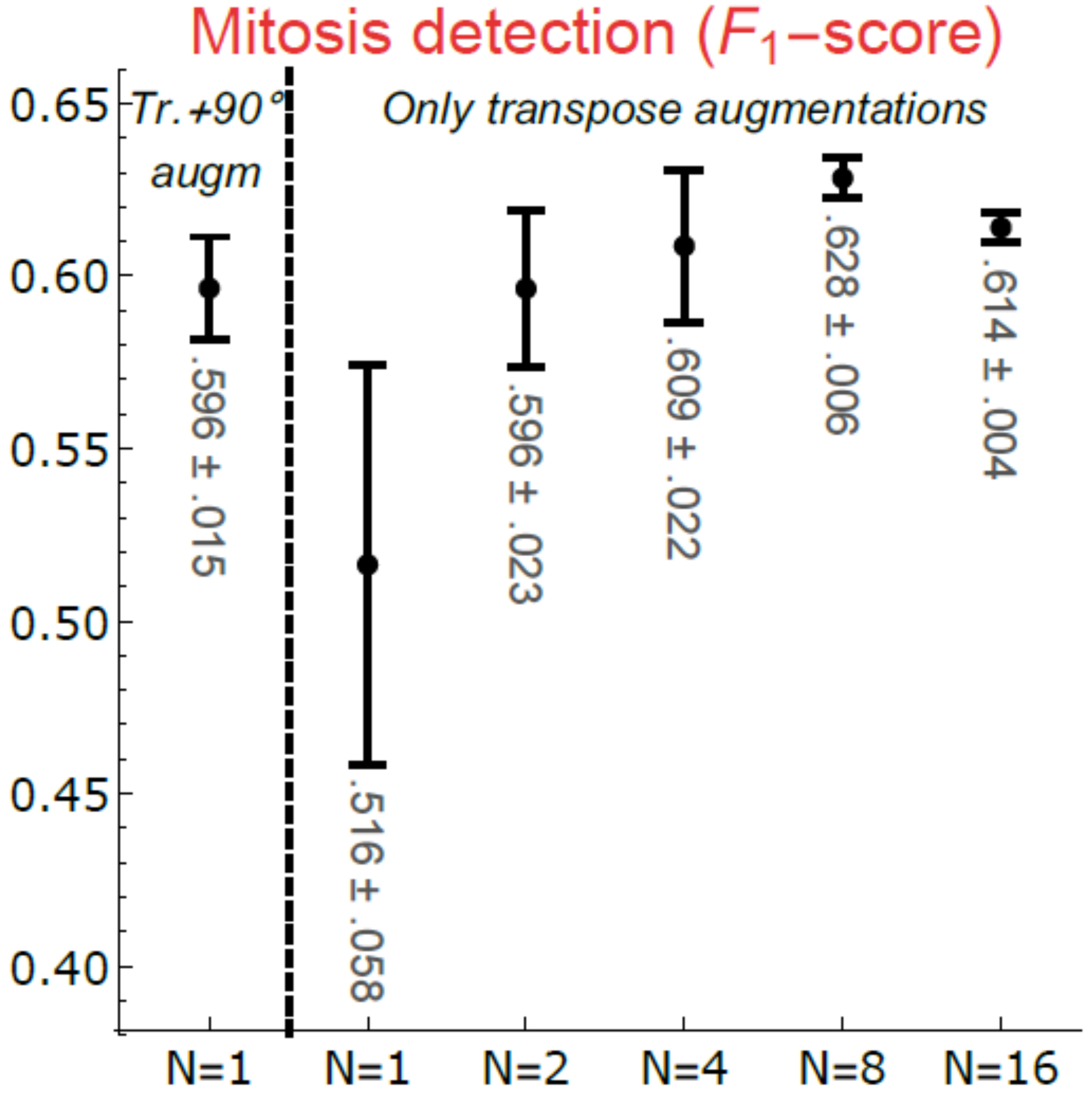
G-CNNs guarantee geometric stability. They are ***robust to input distortions****, regular CNNs aren't…*

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

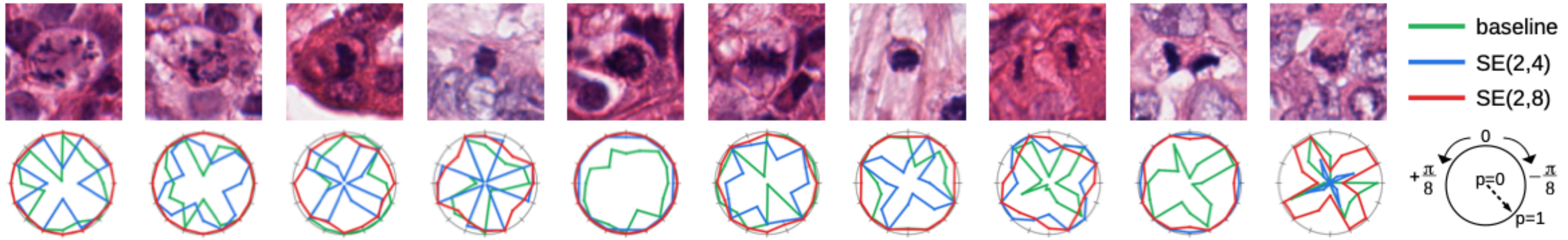
Architecture for rotation invariant mitotic cell detection

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

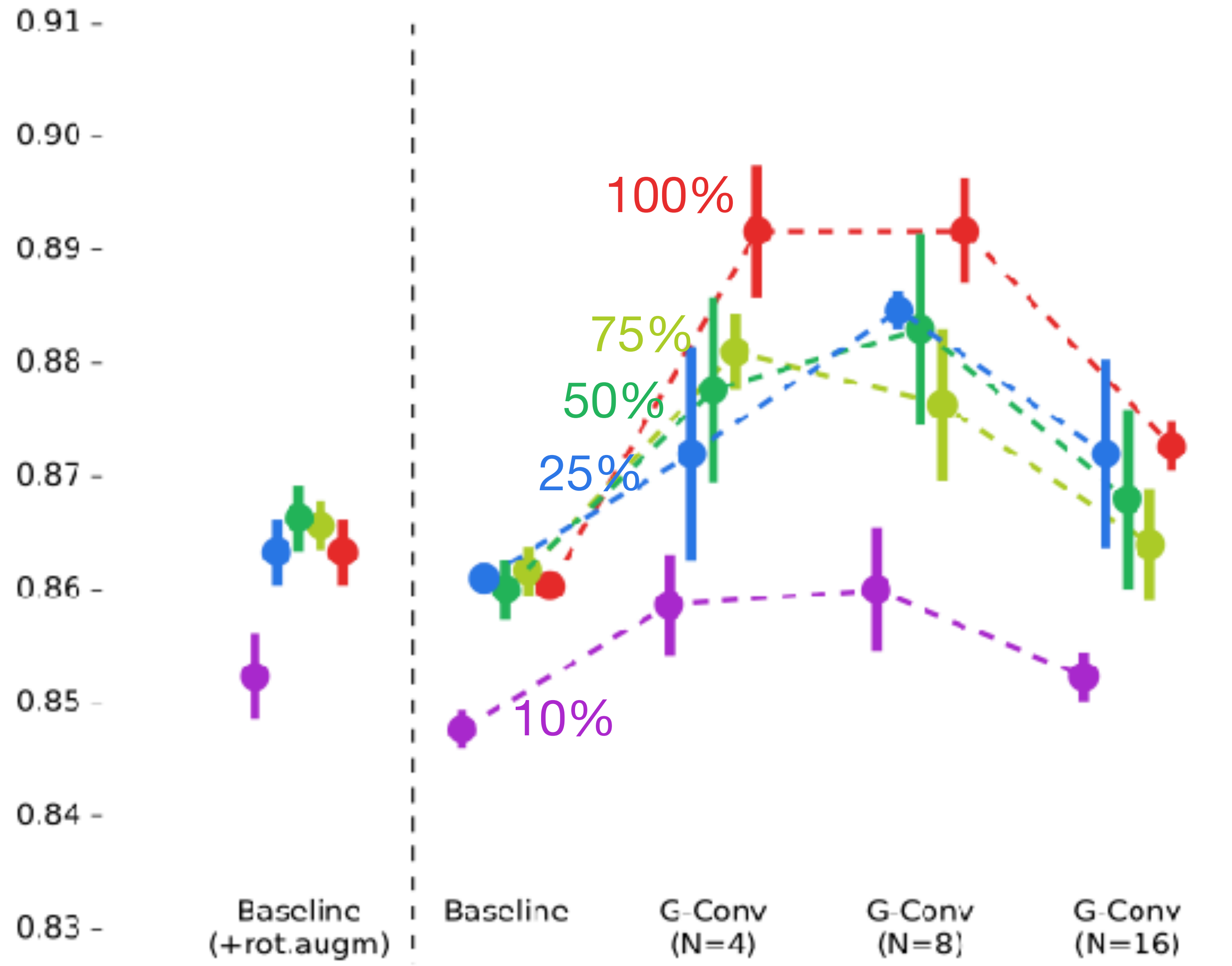
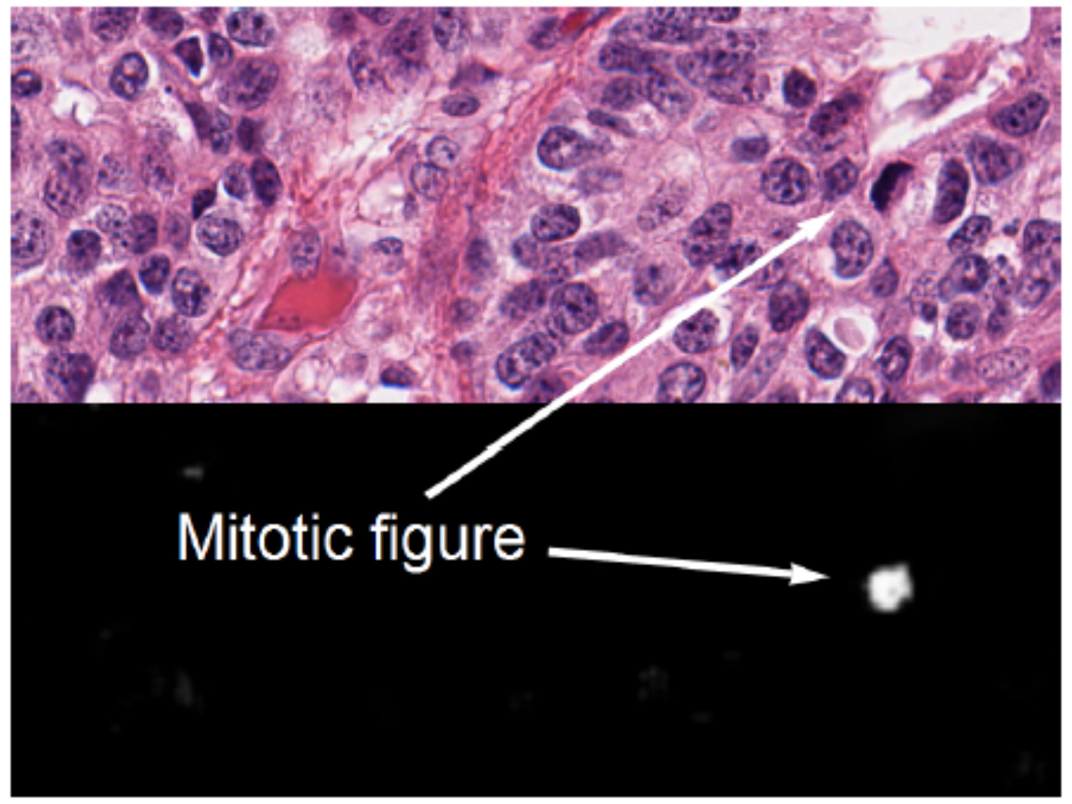
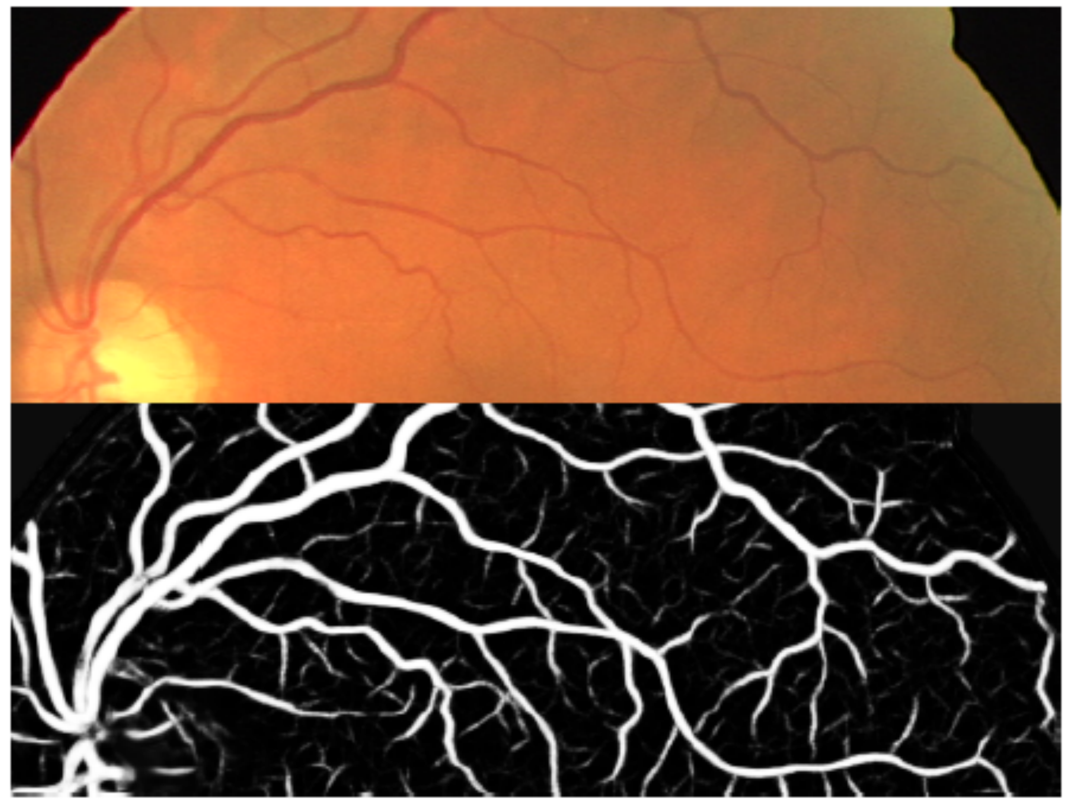


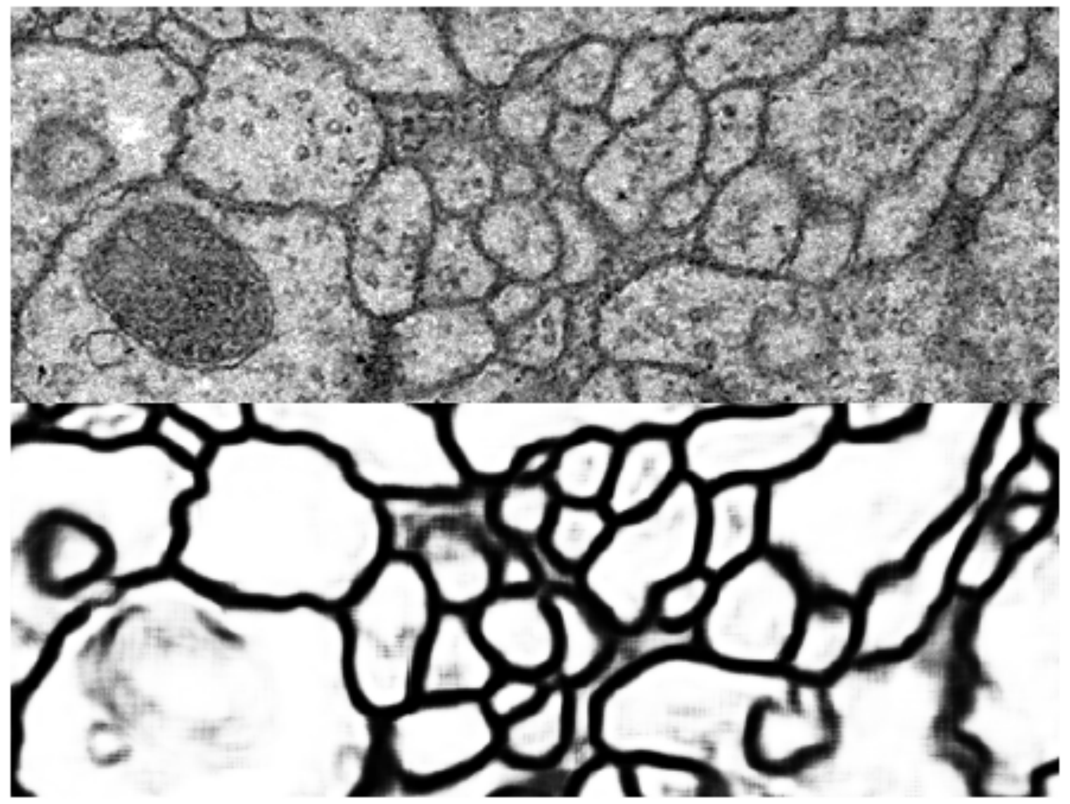
Figure 7: Mean and Standard Deviation plots summarizing the accuracy of the tumor classification models. Mean ± standard deviation is indicated. Color identifies the different data regime (red: 100%; lime: 75%; green: 50%; blue: 25%; purple: 10%).

Experiments in medical image analysis

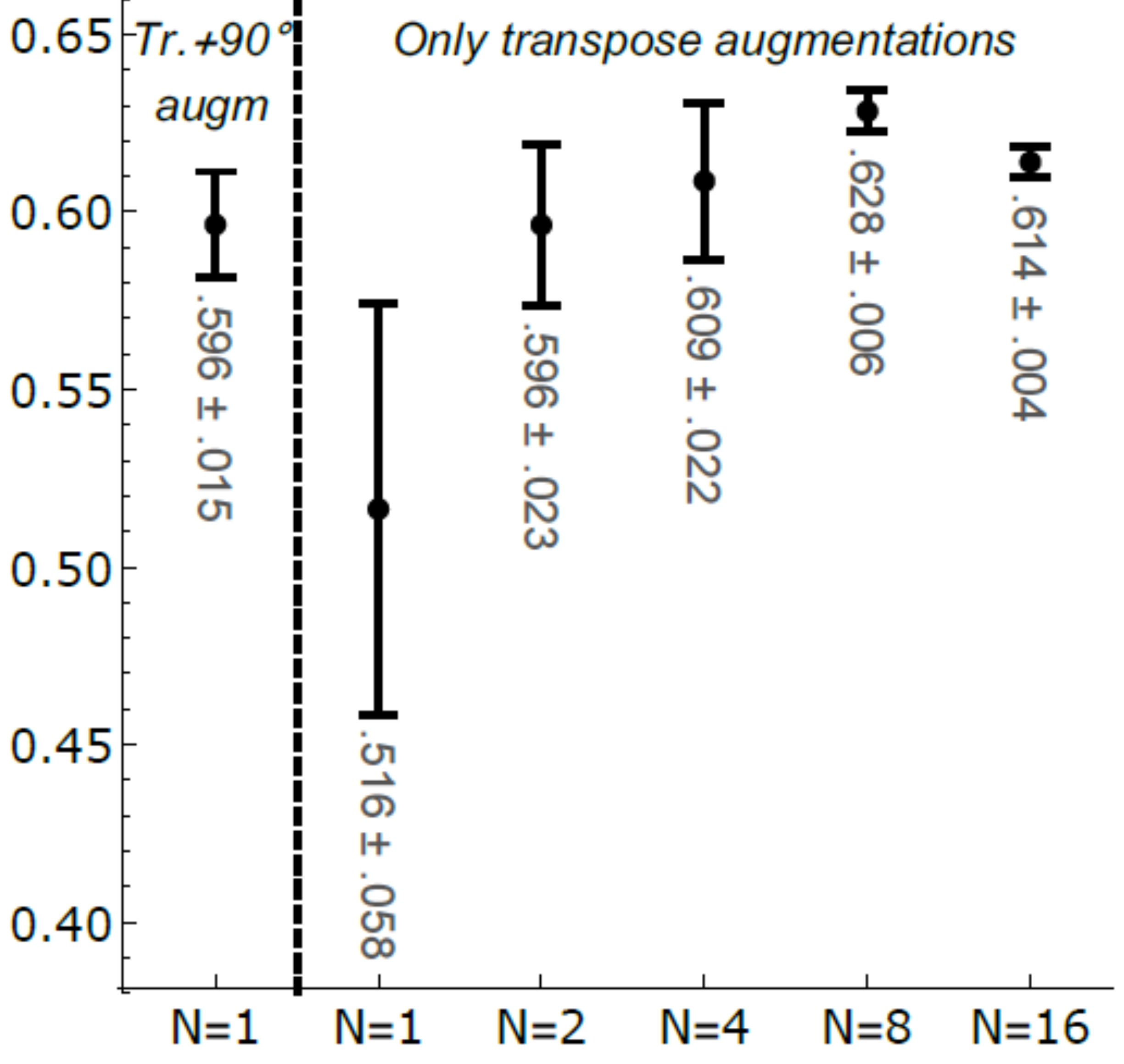
Bekkers & Lafarge et al. MICCAI 2018



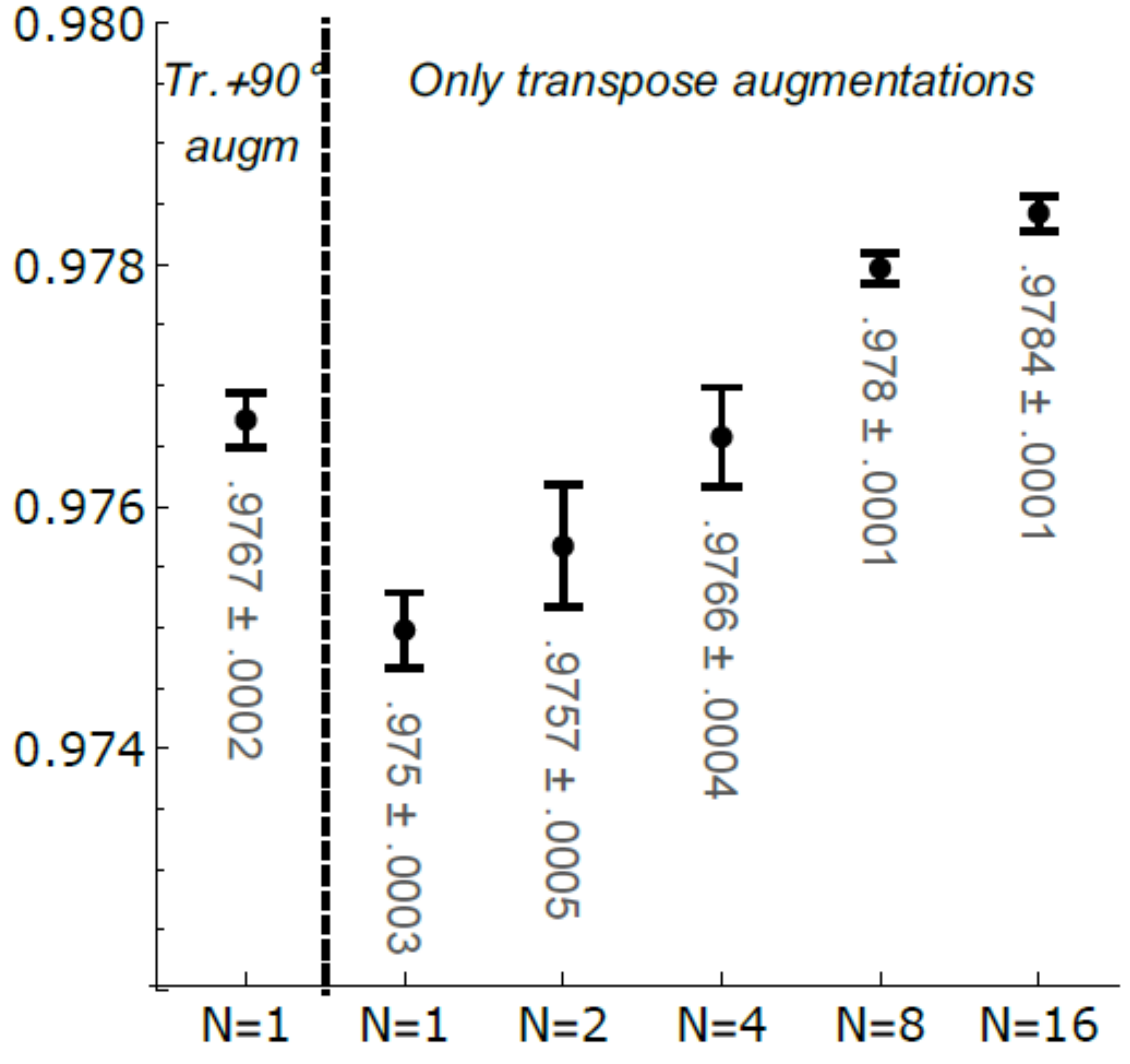




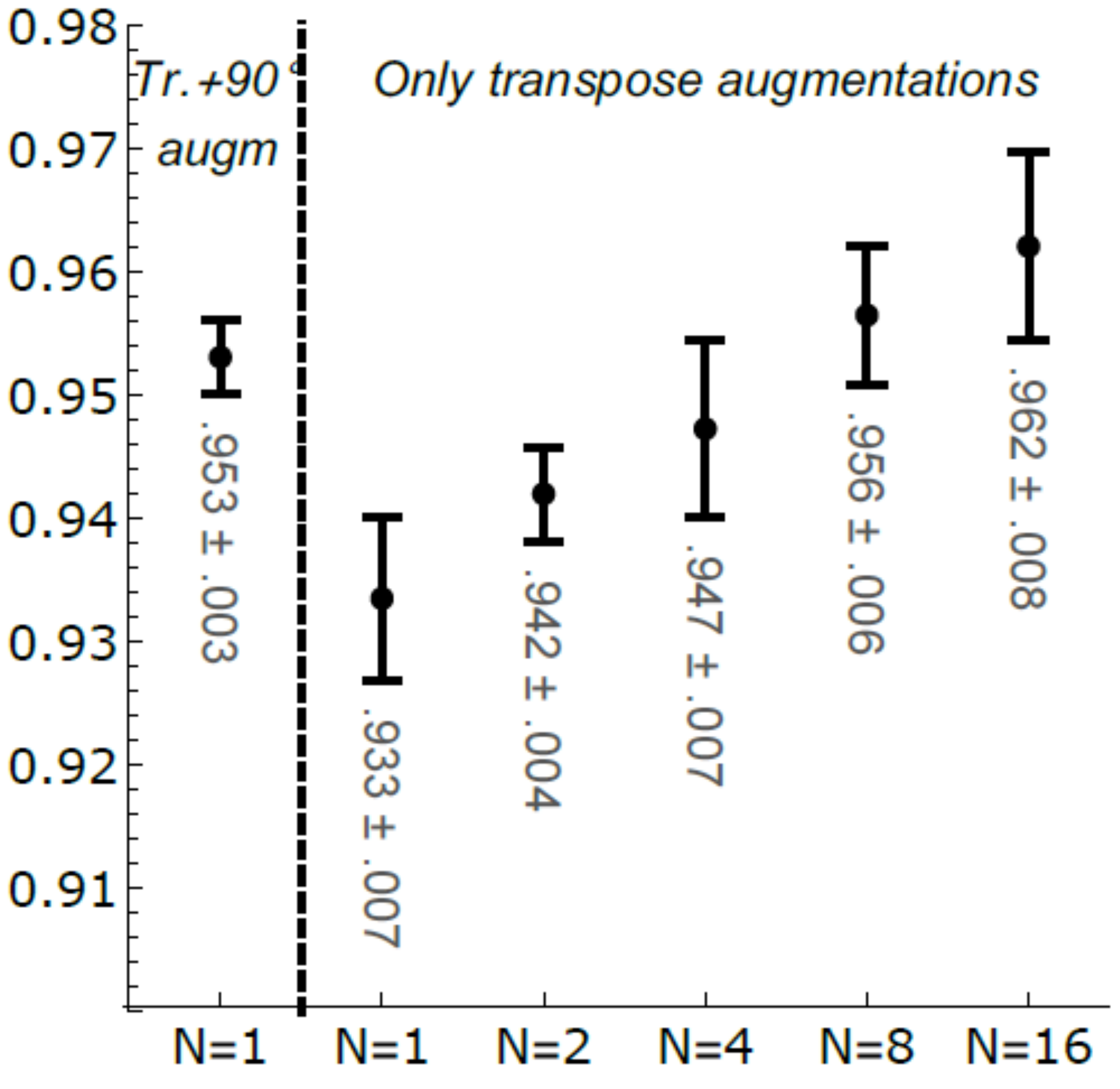
Mitosis detection (F_1–score)



Vessel segmentation (AUC)

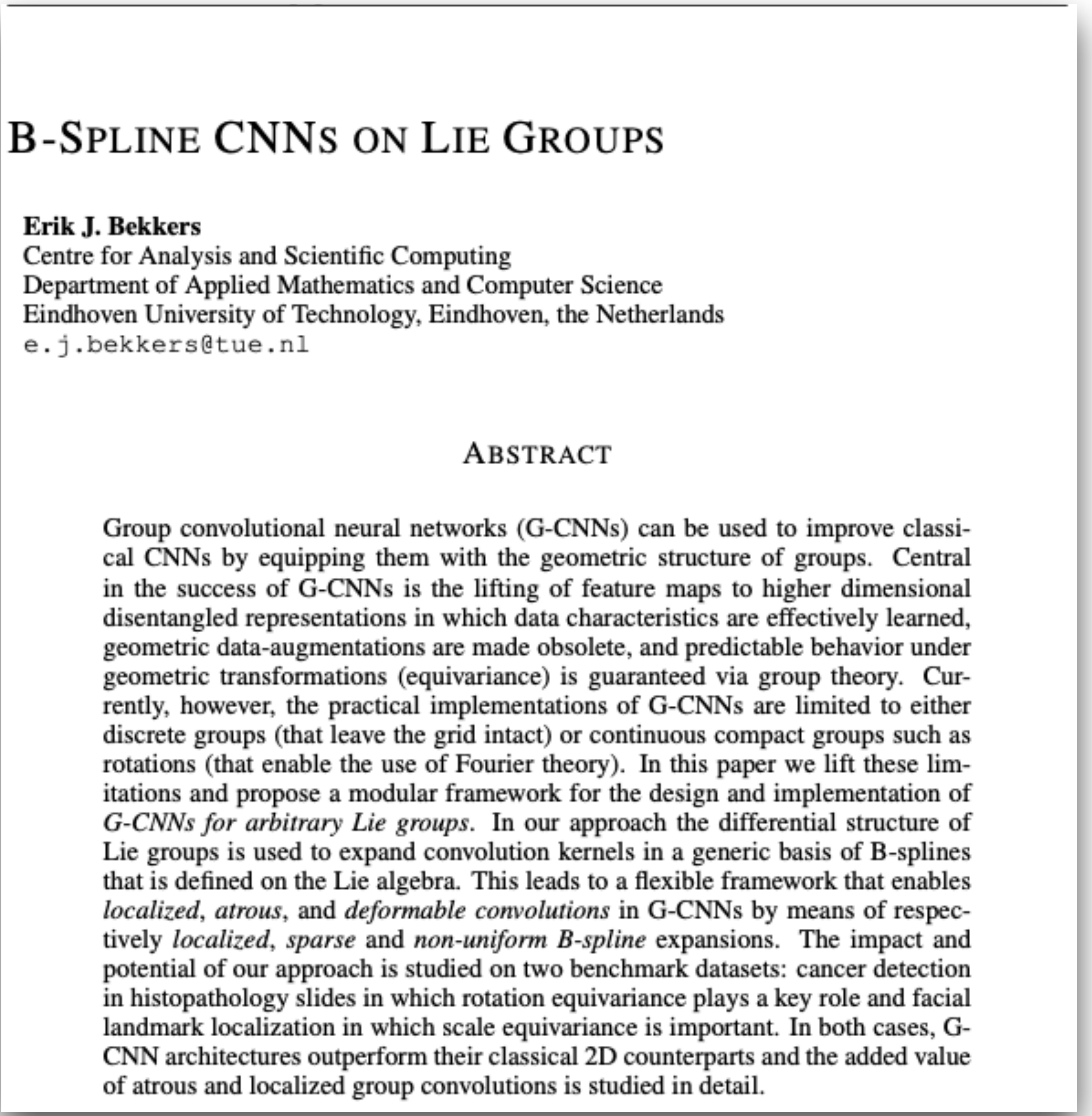


Cell boundary segmentation (Rand)



From rotation to scale equivariant CNNs

Bekkers ICLR 2020

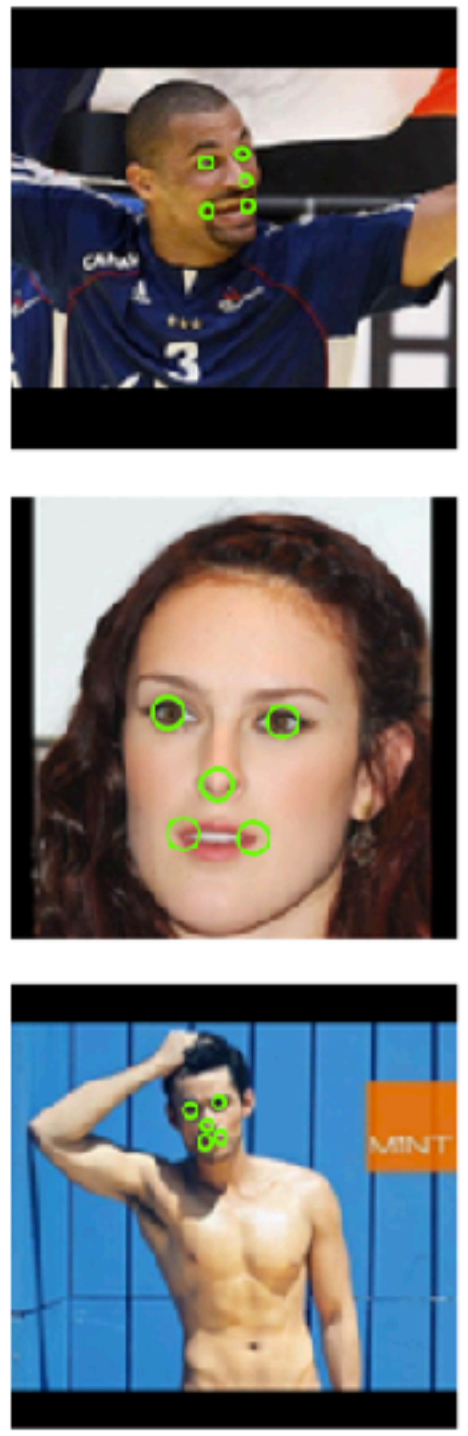


B-SPLINE CNNS ON LIE GROUPS

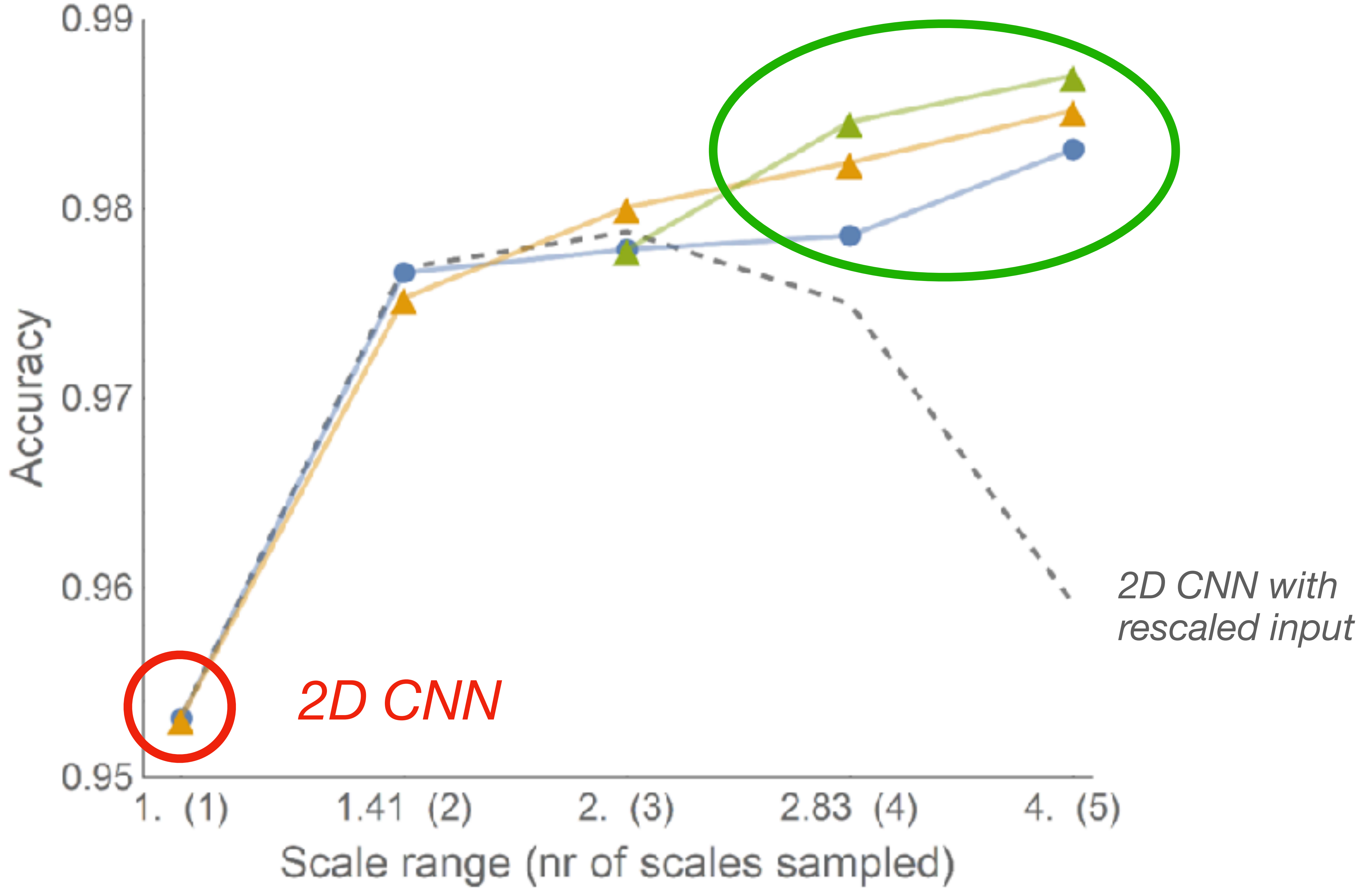
Erik J. Bekkers
Centre for Analysis and Scientific Computing
Department of Applied Mathematics and Computer Science
Eindhoven University of Technology, Eindhoven, the Netherlands
e.j.bekkers@tue.nl

ABSTRACT

Group convolutional neural networks (G-CNNs) can be used to improve classical CNNs by equipping them with the geometric structure of groups. Central in the success of G-CNNs is the lifting of feature maps to higher dimensional disentangled representations in which data characteristics are effectively learned, geometric data-augmentations are made obsolete, and predictable behavior under geometric transformations (equivariance) is guaranteed via group theory. Currently, however, the practical implementations of G-CNNs are limited to either discrete groups (that leave the grid intact) or continuous compact groups such as rotations (that enable the use of Fourier theory). In this paper we lift these limitations and propose a modular framework for the design and implementation of *G-CNNs for arbitrary Lie groups*. In our approach the differential structure of Lie groups is used to expand convolution kernels in a generic basis of B-splines that is defined on the Lie algebra. This leads to a flexible framework that enables *localized*, *atrous*, and *deformable convolutions* in G-CNNs by means of respectively *localized*, *sparse* and *non-uniform B-spline* expansions. The impact and potential of our approach is studied on two benchmark datasets: cancer detection in histopathology slides in which rotation equivariance plays a key role and facial landmark localization in which scale equivariance is important. In both cases, G-CNN architectures outperform their classical 2D counterparts and the added value of atrous and localized group convolutions is studied in detail.



Translation + *scale equivariant G-CNNs*

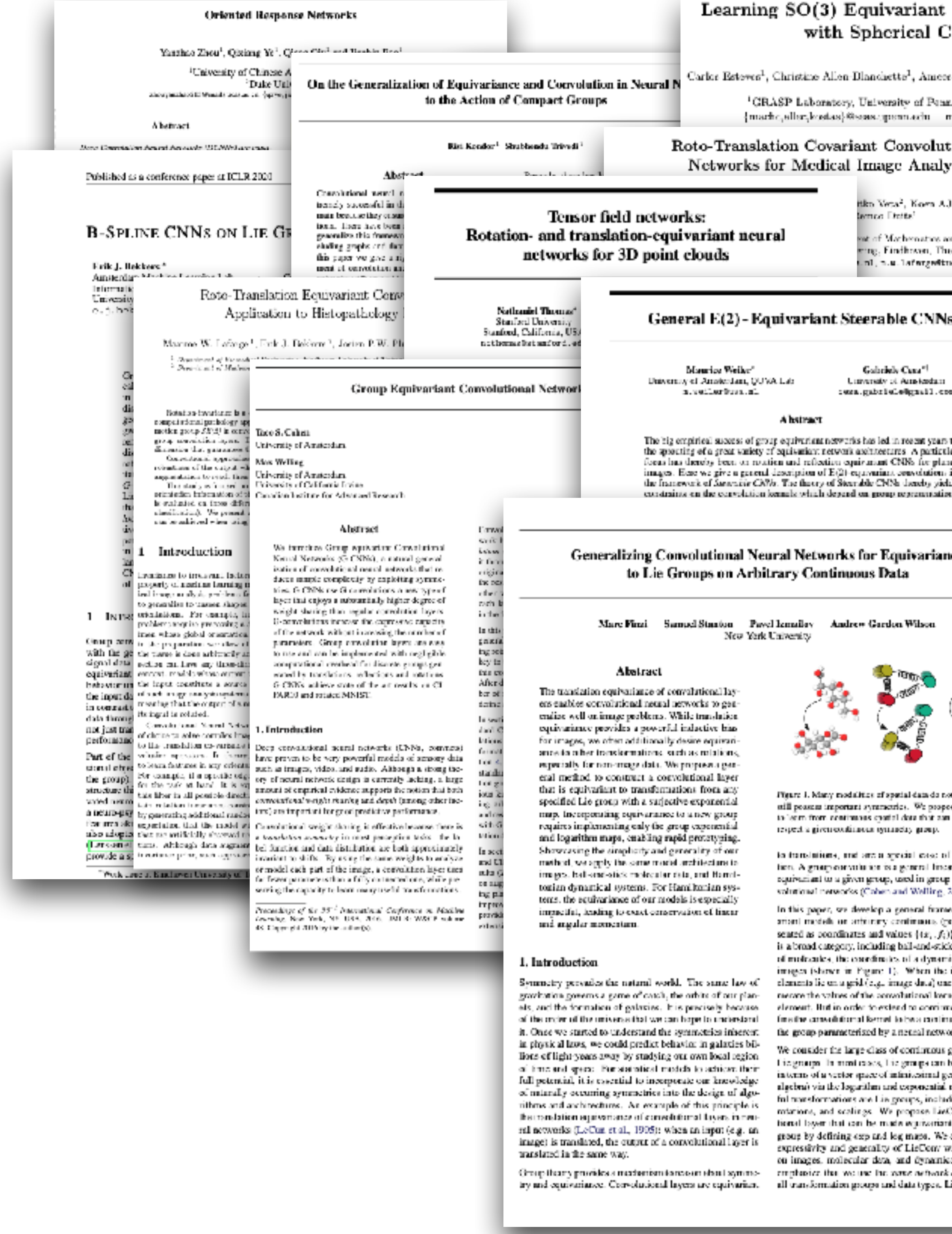


G-CNNs rule!

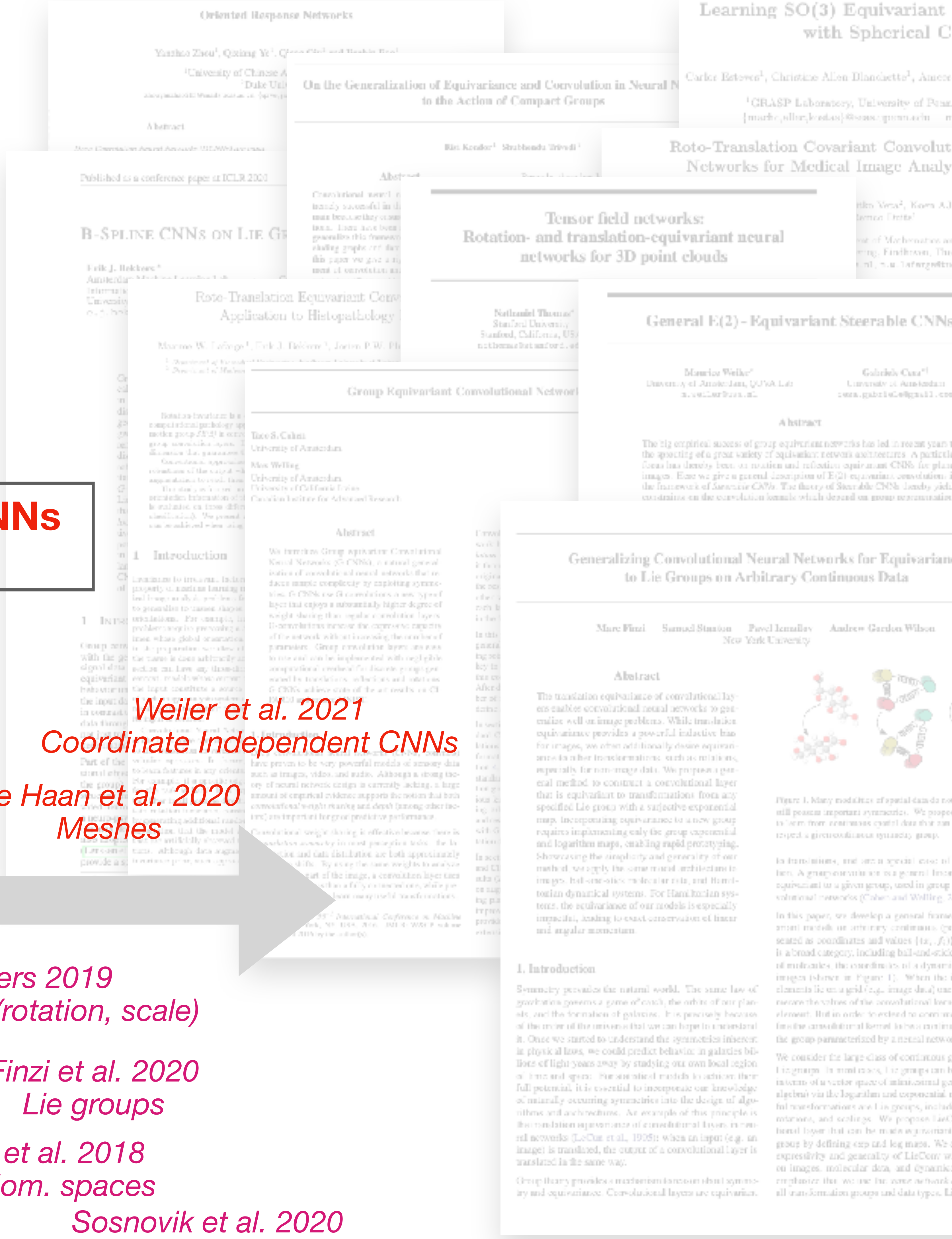
- The right inductive bias: guaranteed equivariance
  (no loss of information)
- Performance gains that can’t be obtained by data-augmentation alone
  (both local and global equivariance/invariance)
- Increased sample efficiency
  (increased weight sharing, no geometric augmentation necessary)

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



NNs → CNNs → G-CNNs

LeCun et al. 1990

Discrete G-CNNs
(square/cube symmetries)

- ***Cohen-Welling 2016*** ***p4m***
- *Dieleman et al. 2016* *p4m*
- *Winkels-Cohen 2018* *3D discrete roto-transl.*
- *Hoogeboom et al. 2018* *SE(2,6)*
- *Worrall-Brostow 2018* *3D discrete roto-transl.*

Gauge-equivariant CNNs
(manifolds)

- *Cohen et al. 2019* *Icosahedron*
- *de Haan et al. 2020* *Meshes*
- *Weiler et al. 2021* *Coordinate Independent CNNs*

Continuous rotation G-CNNs
(regular discretizations)

- *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)*
- *Esteves 2017* *SO(3)*

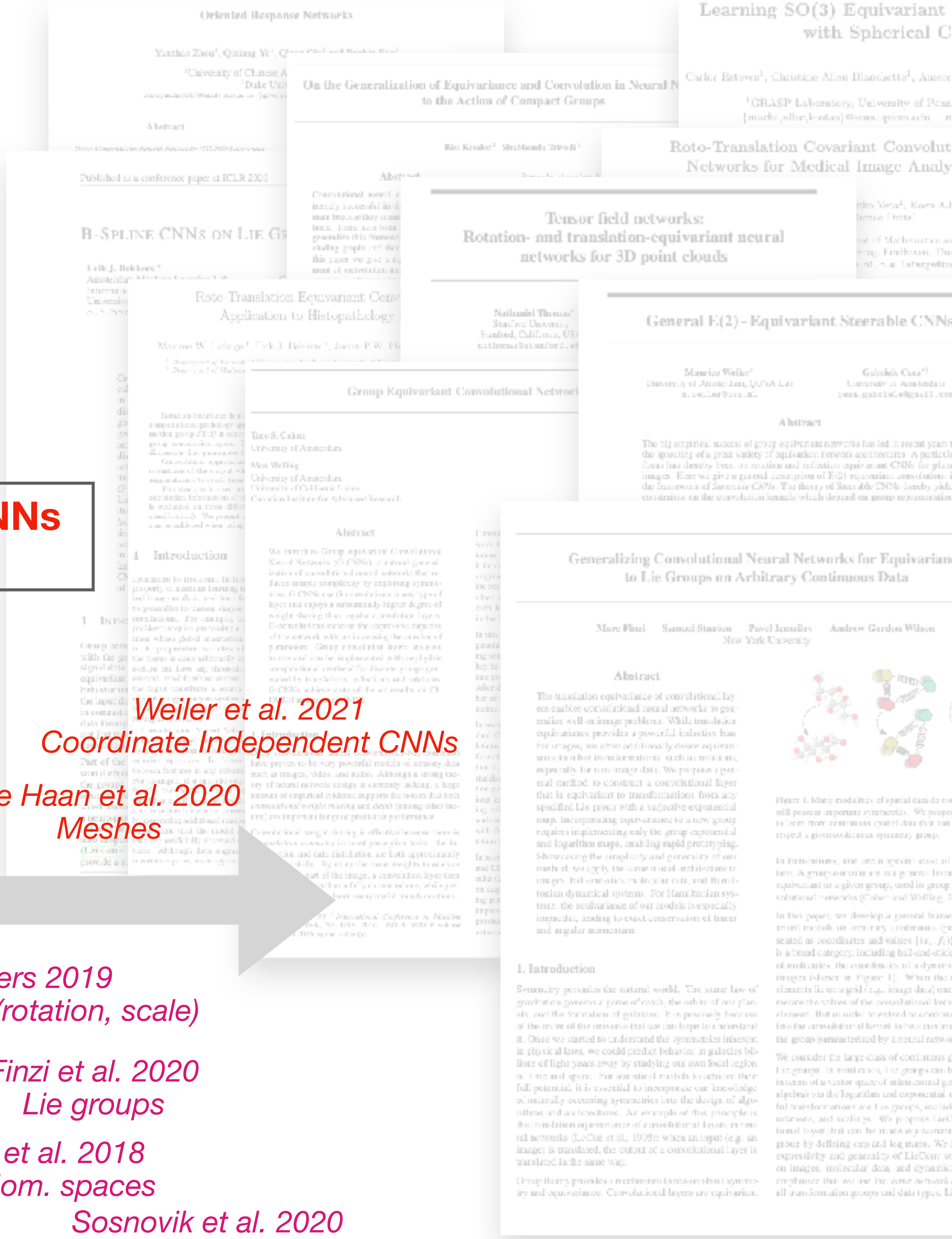
Continuous rotation G-CNNs
(steerable)

- *Worrall et al. 2017* *SE(2)*
- *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)*

Continuous G-CNNs
(Lie groups)

- *Bekkers 2019* *Lie groups (rotation, scale)*
- *Finzi et al. 2020* *Lie groups*
- *Chakraborty et al. 2018* *Riemannian Hom. spaces*
- *Sosnovik et al. 2020* *Scale-translation*

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

Weiler et al. 2021
Coordinate Independent CNNs

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)

Thomas et al. 2018
SE(3)

Cohen et al. 2018
SE(3)

Kondor-Trivedi 2018
SE(3)

Weiler et al. 2018
SE(3)

Weiler-Cesa 2019
SE(2)

Bekkers 2019
Lie groups (rotation, scale)

Finzi et al. 2020
Lie groups

Chakraborty et al. 2018
Riemannian Hom. spaces

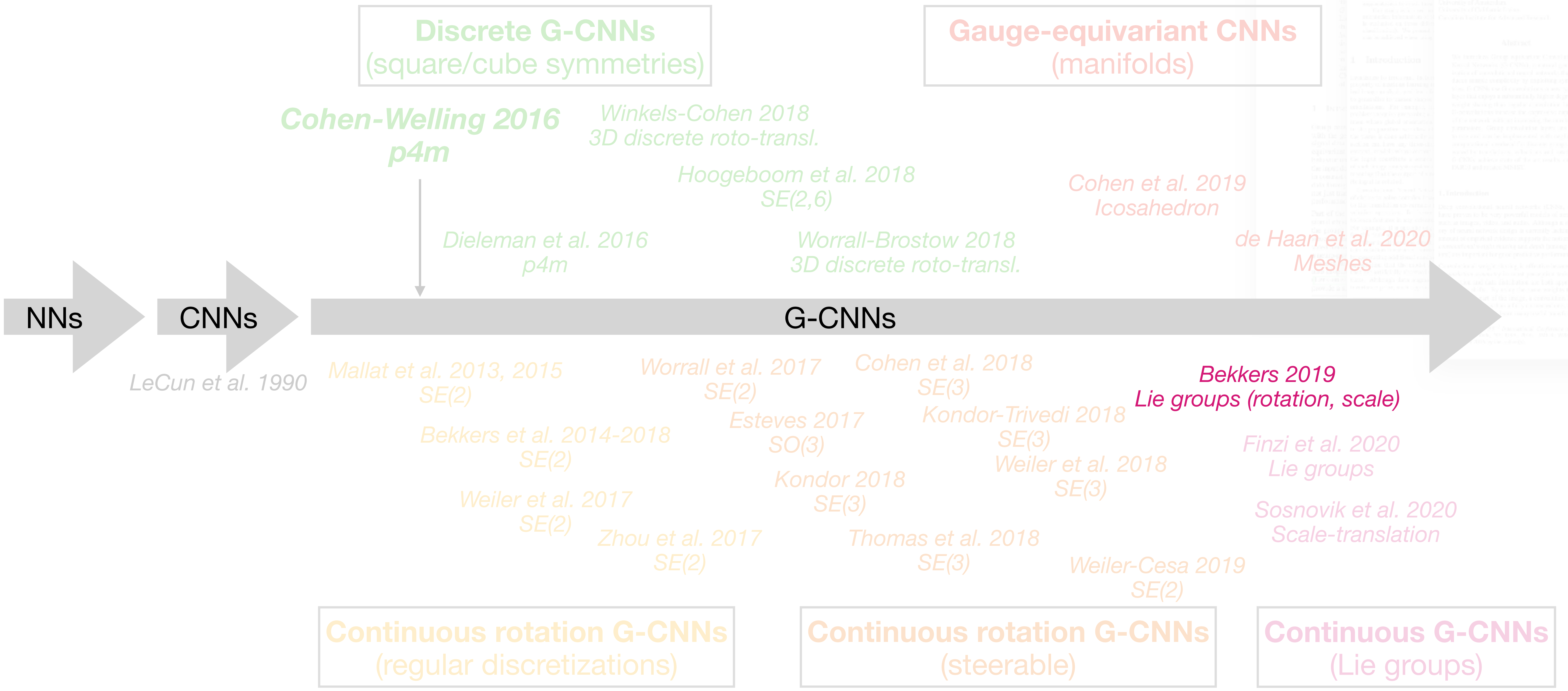
Sosnovik et al. 2020
Scale-translation

Continuous rotation G-CNNs
(regular discretizations)

Continuous rotation G-CNNs
(steerable)

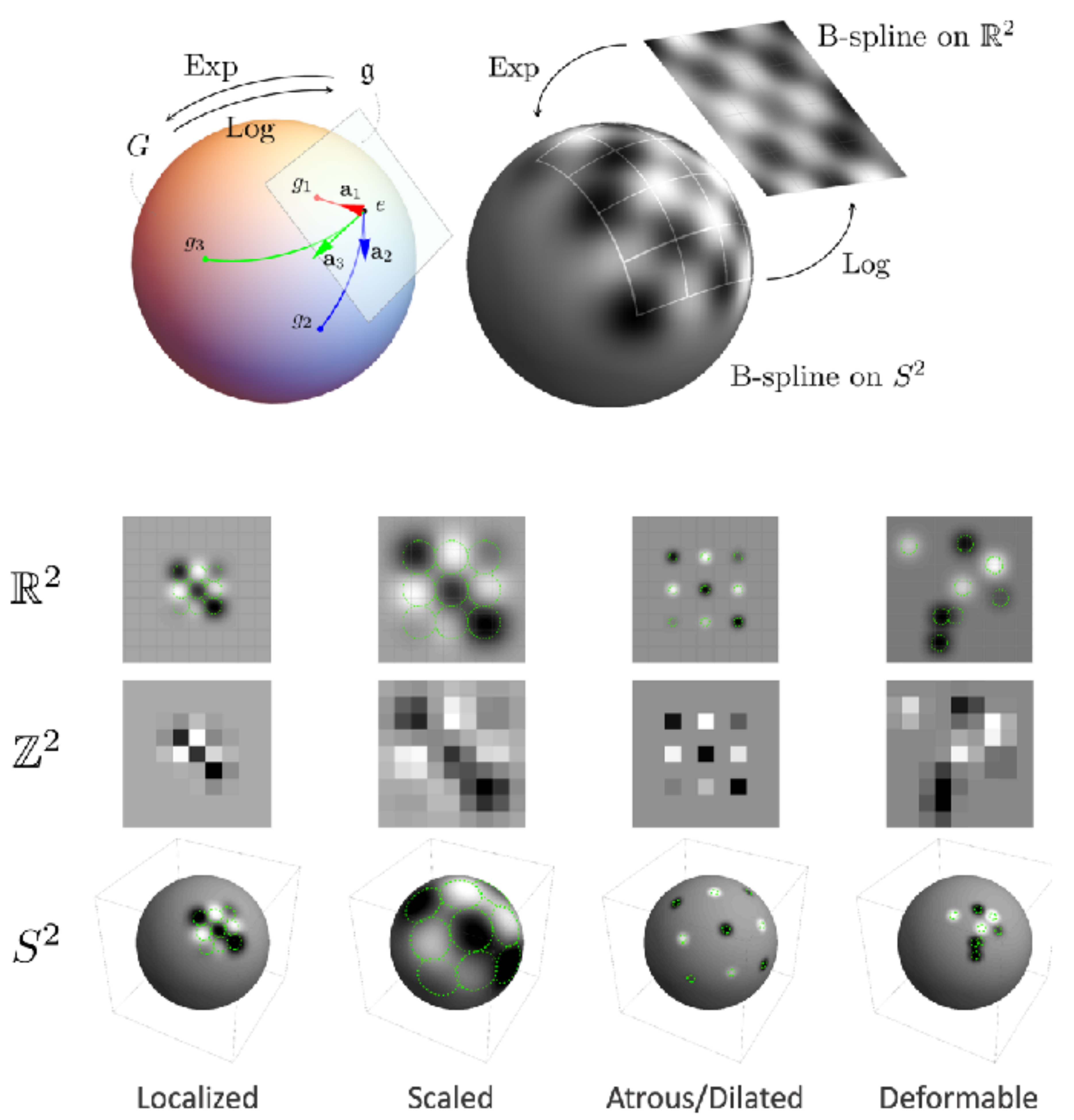
Continuous G-CNNs
(Lie groups)

A brief history of G-CNNs



A brief history of G-CNNs

B-Spline CNNs on Lie Groups - ICLR 2020



Gauge-equivariant CNNs (manifolds)

hen 2018 roto-transl.

ogeboom et al. 2018 SE(2,6)

Worrall-Brostow 2018 3D discrete roto-transl.

Cohen et al. 2019 Icosahedron

de Haan et al. 2020 Meshes

G-CNNs

et al. 2017 SE(2)

Cohen et al. 2018 SE(3)

Esteves 2017 SO(3)

Kondor-Trivedi 2018 SE(3)

Bekkers 2019 Lie groups (rotation, scale)

Finzi et al. 2020 Lie groups

Weiler et al. 2018 SE(3)

Kondor 2018 SE(3)

Sosnovik et al. 2020 Scale-translation

2017

Thomas et al. 2018 SE(3)

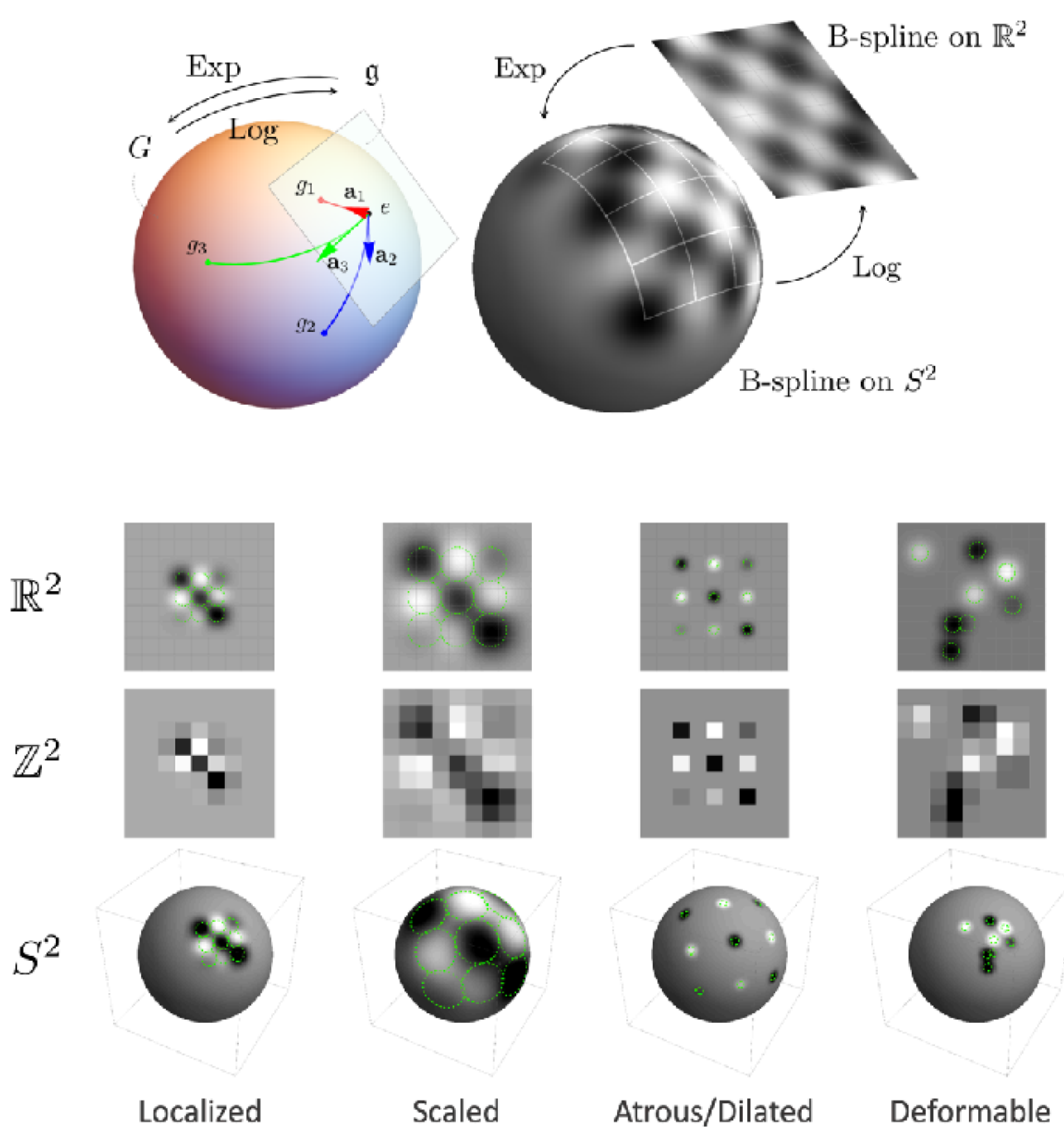
Weiler-Cesa 2019 SE(2)

Continuous rotation G-CNNs (steerable)

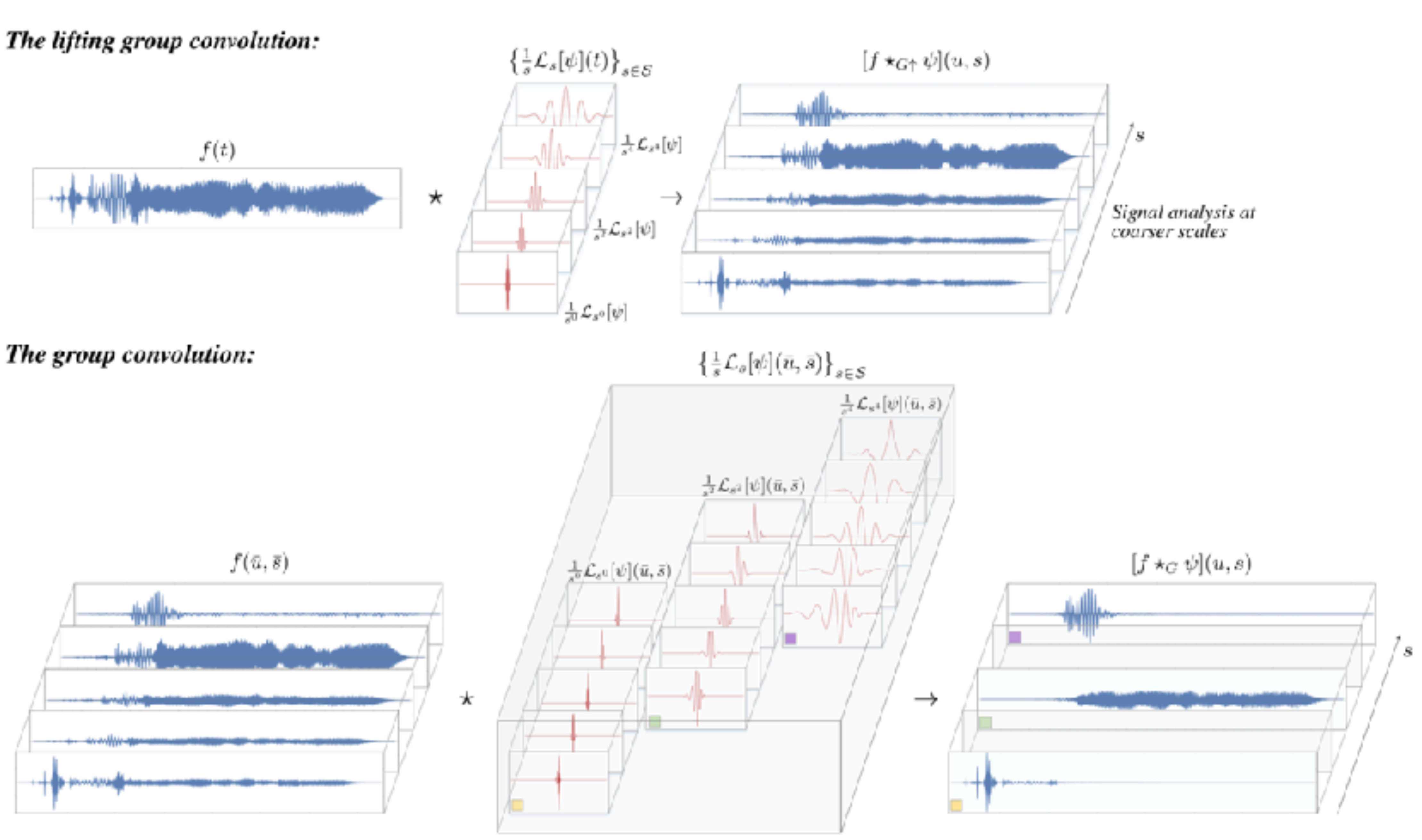
Continuous G-CNNs (Lie groups)

A brief history of G-CNNs

B-Spline CNNs on Lie Groups - ICLR 2020



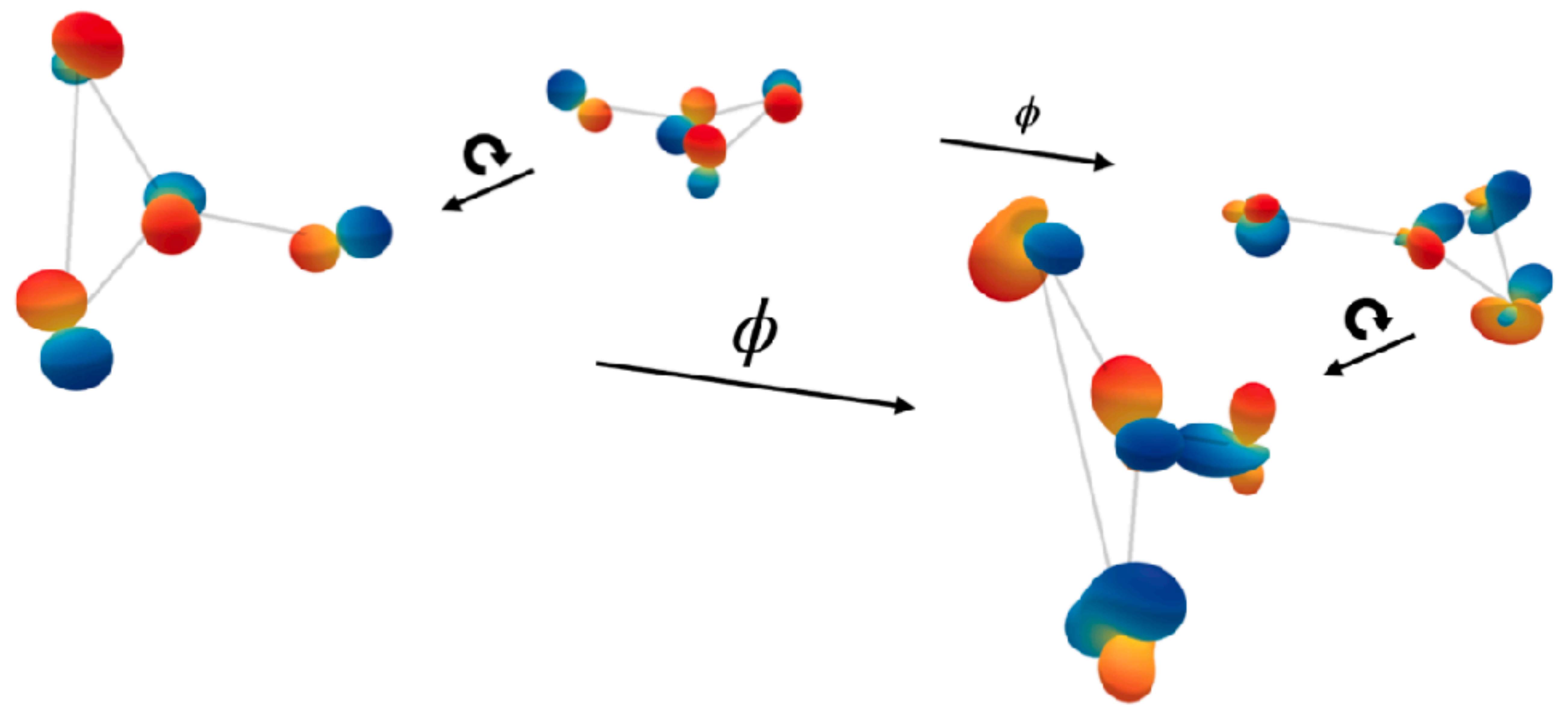
Romero, Bekkers, Tomczak, Hoogeboom
Wavelet Networks: Scale Equivariant Learning From Raw Waveforms - arXiv:2006.05259



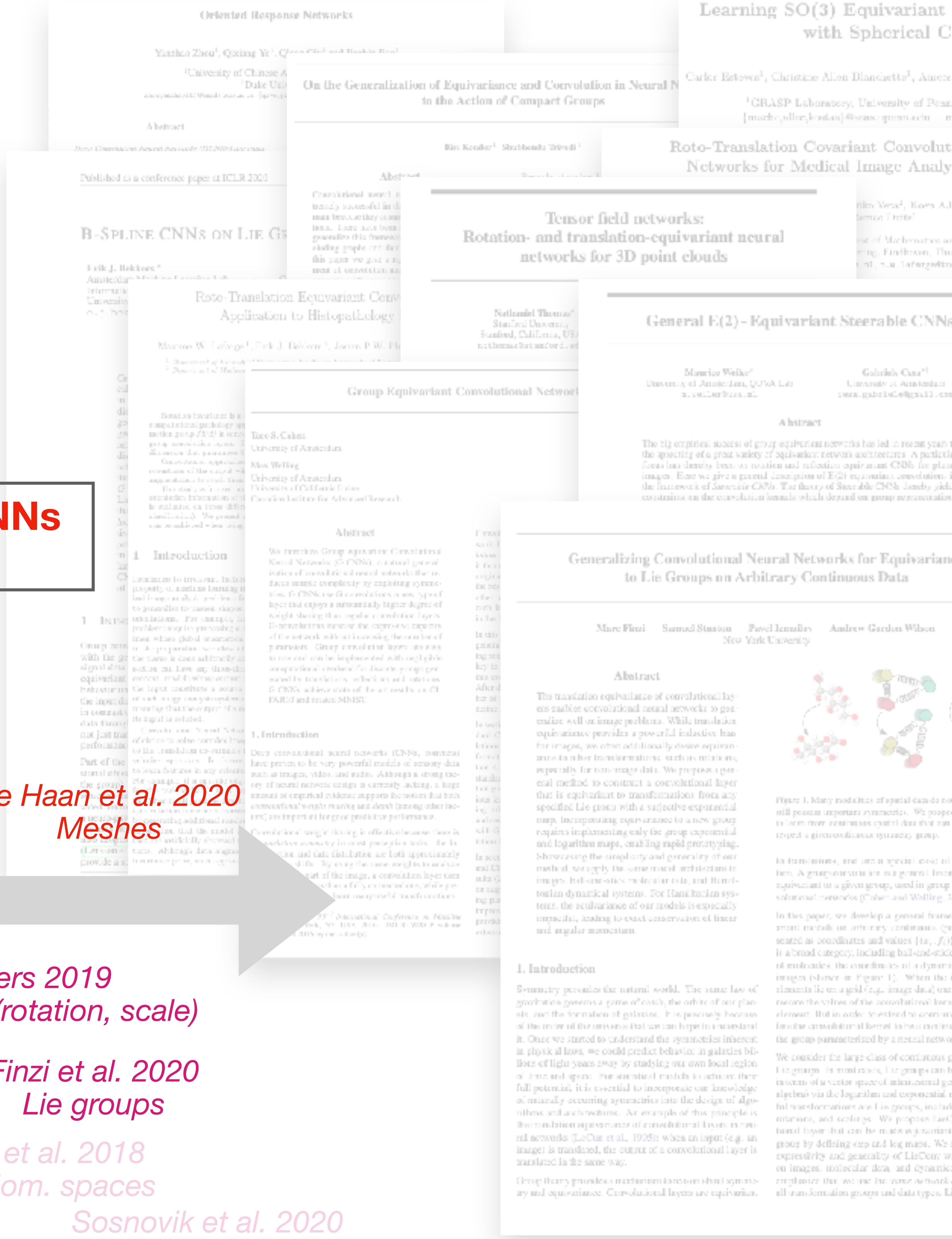
A brief history of G-CNNs

NNs

Brandstetter, Hesselink, van der Pol, Bekkers, Welling **Geometric and Physical Quantities Improve E(3) Equivariant Message Passing** - arXiv:2110.02905



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

Chakraborty et al. 2018
Riemannian Hom. spaces

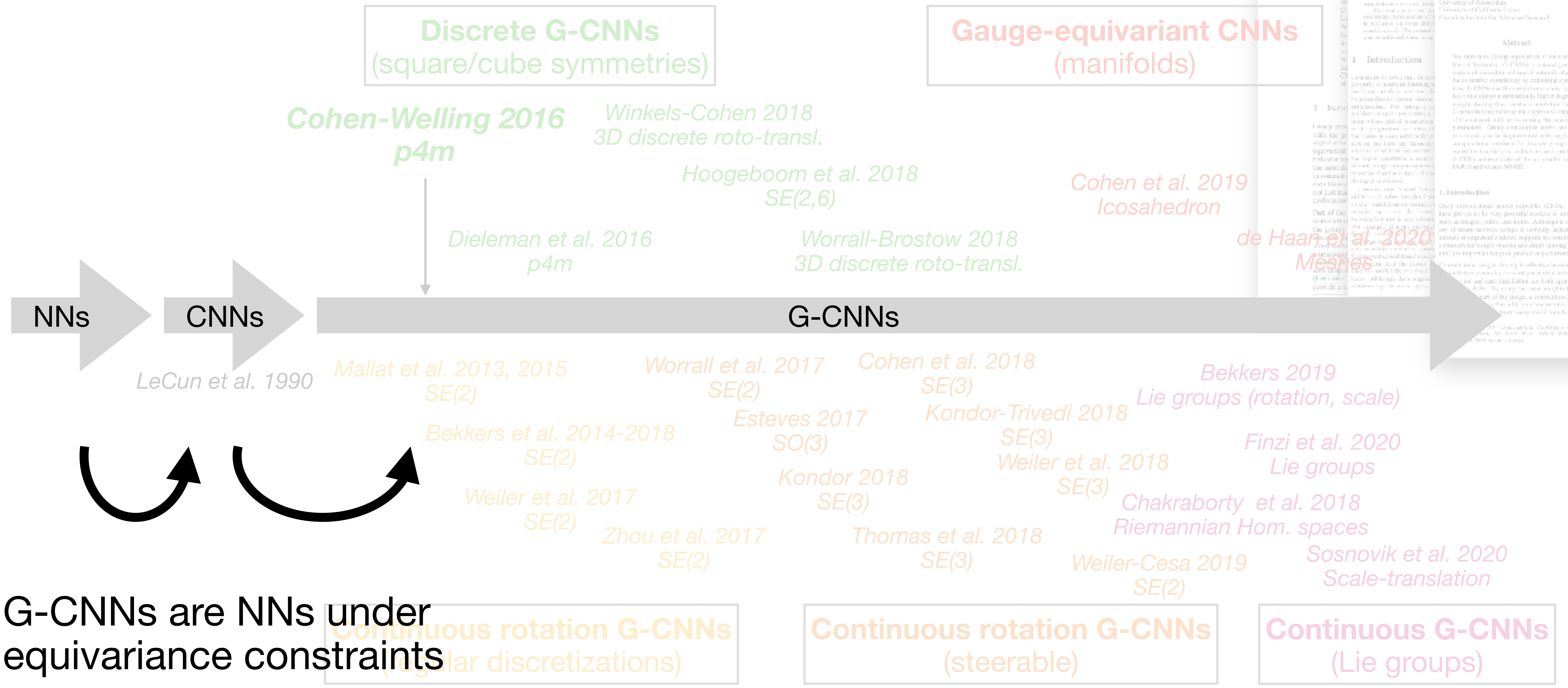
Sosnovik et al. 2020
Scale-translation

Continuous rotation G-CNNs
(regular discretizations)

Continuous rotation G-CNNs
(steerable)

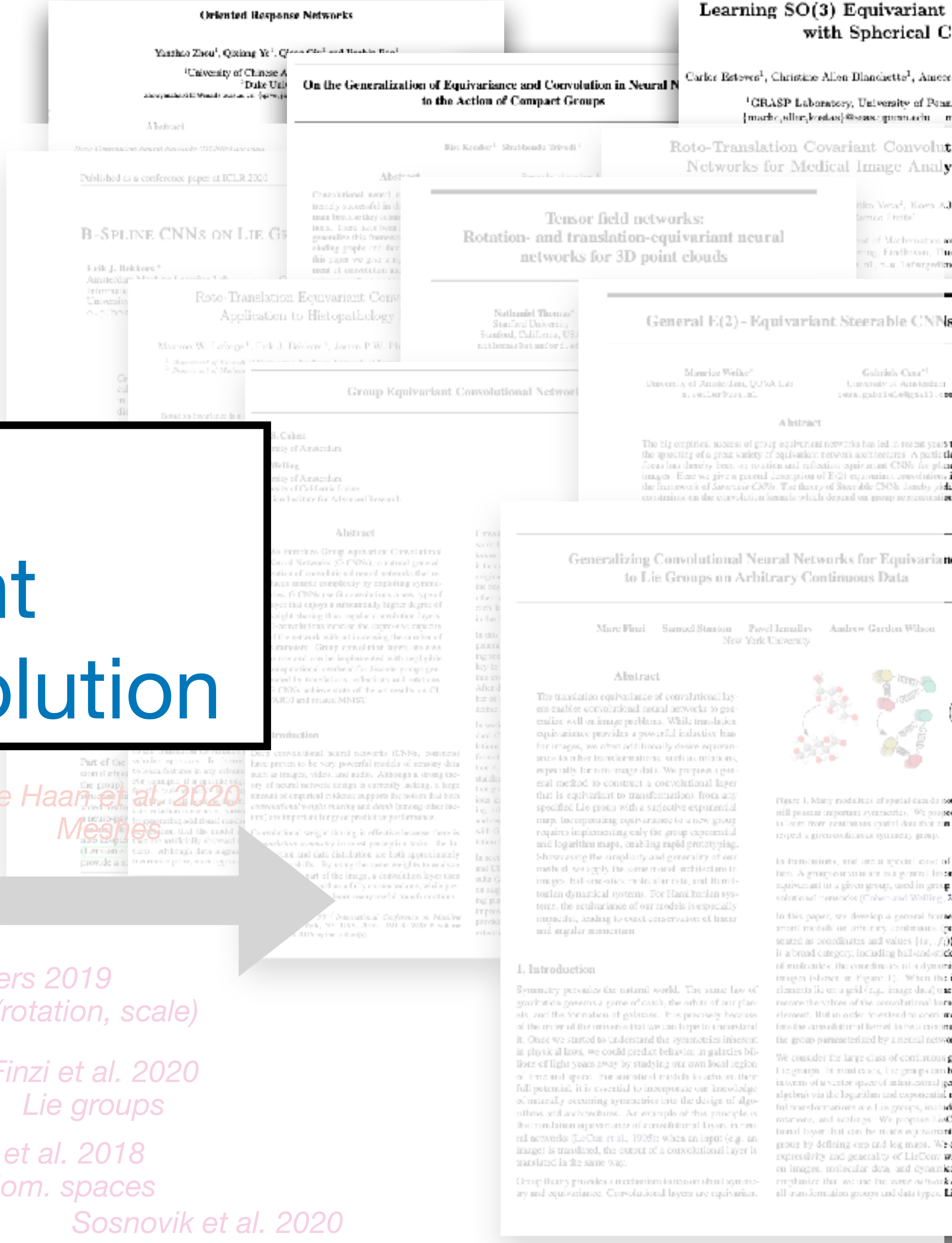
Continuous G-CNNs
(Lie groups)

A brief history of G-CNNs



G-CNNs are NNs under
equivariance constraints

A brief history of G-CNNs



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

Dieleman et al. 2016
p4m

Worrall-Brostow 2018
3D discrete roto-transl.

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)

Thomas et al. 2018
SE(3)

Cohen et al. 2018
SE(3)

Kondor-Trivedi 2018
SE(3)

Weiler et al. 2018
SE(3)

Weiler-Cesa 2019
SE(2)

Bekkers 2019
Lie groups (rotation, scale)

Finzi et al. 2020
Lie groups

Chakraborty et al. 2018
Riemannian Hom. spaces

Sosnovik et al. 2020
Scale-translation

G-CNNs are NNs under
equivariance constraints

Continuous rotation G-CNNs
(regular discretizations)

Continuous rotation G-CNNs
(steerable)

Continuous G-CNNs
(Lie groups)

Content

Part I: Introduction to group convolutions

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

Part II: Group convolutions are all you need

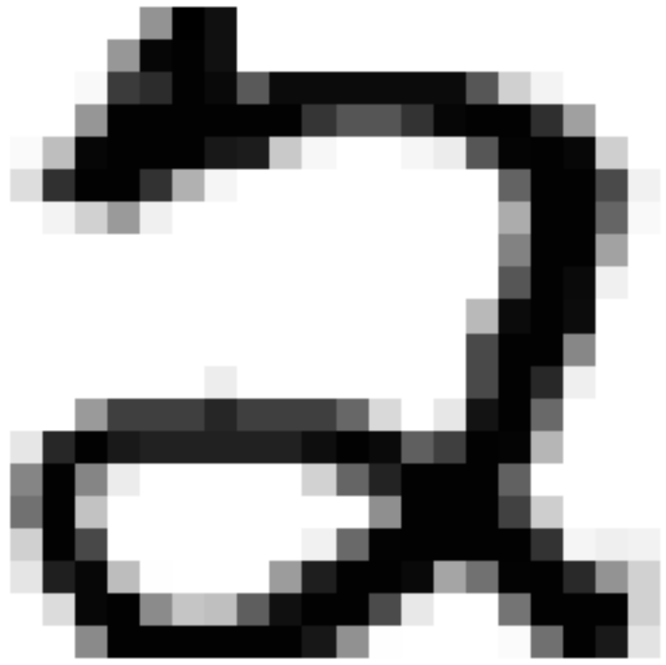
* **Theorem: Equivariant linear operators are group convolutions**
* **Characterization of types of group equivariant layers**

Part III: Steerable group convolutions for molecular data and the N-body problem

* Graph Neural networks
* Irreducible representations, steerable operators and vector spaces
* Steerable Graph NNs (Point Convolutions)

Classical artificial neural networks

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



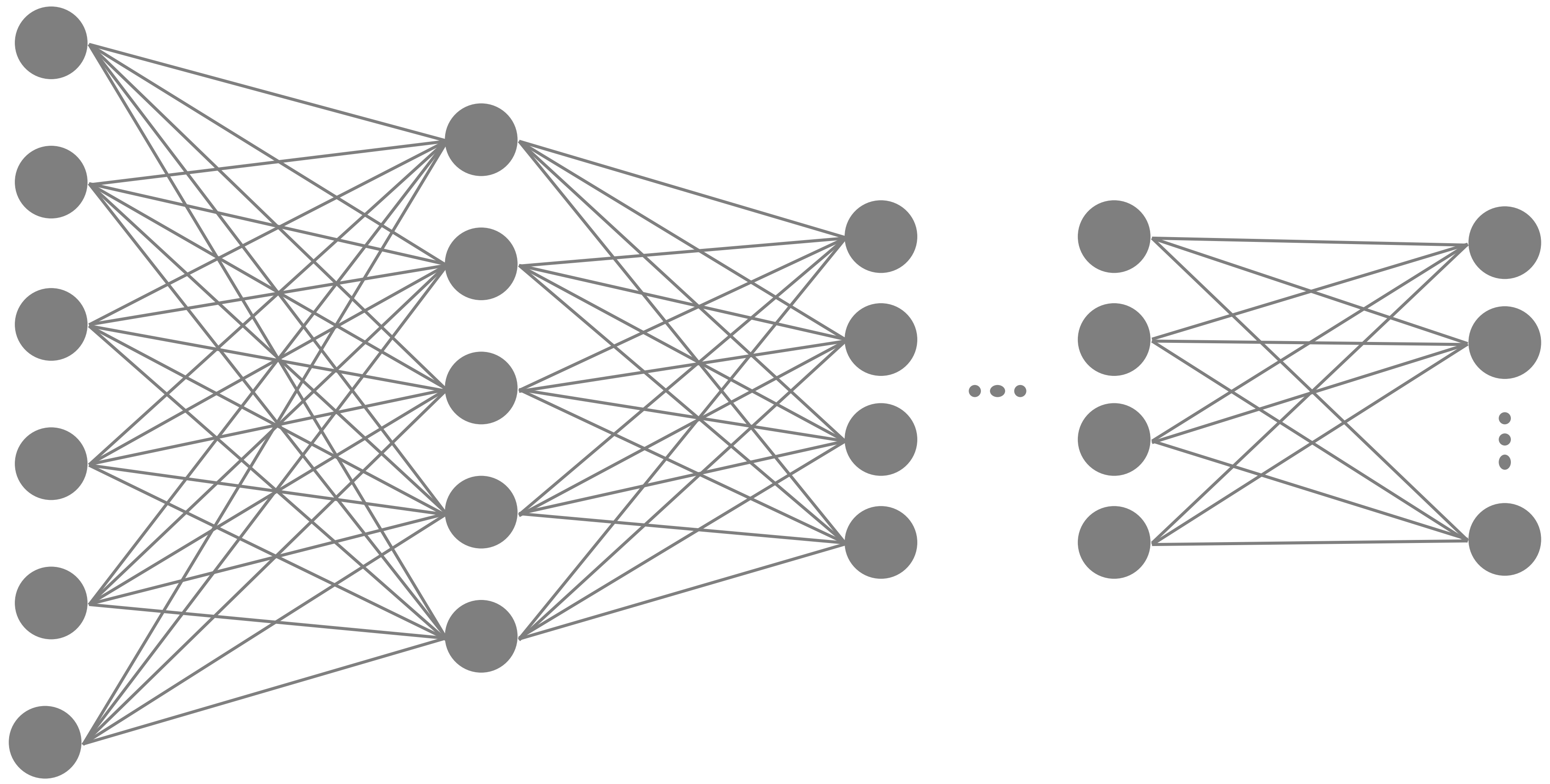
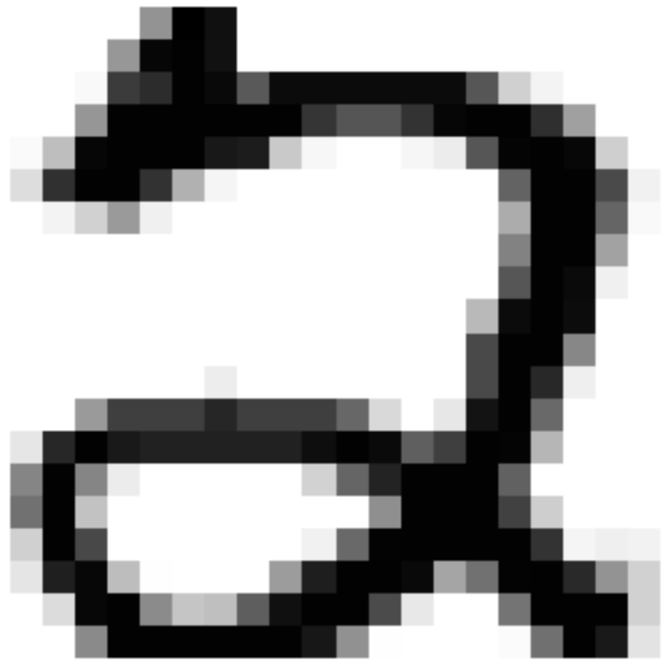


Image analyst: $\underline{x}^0 \in \mathcal{X} = \mathbb{L}_2(\mathbb{R}^2)$

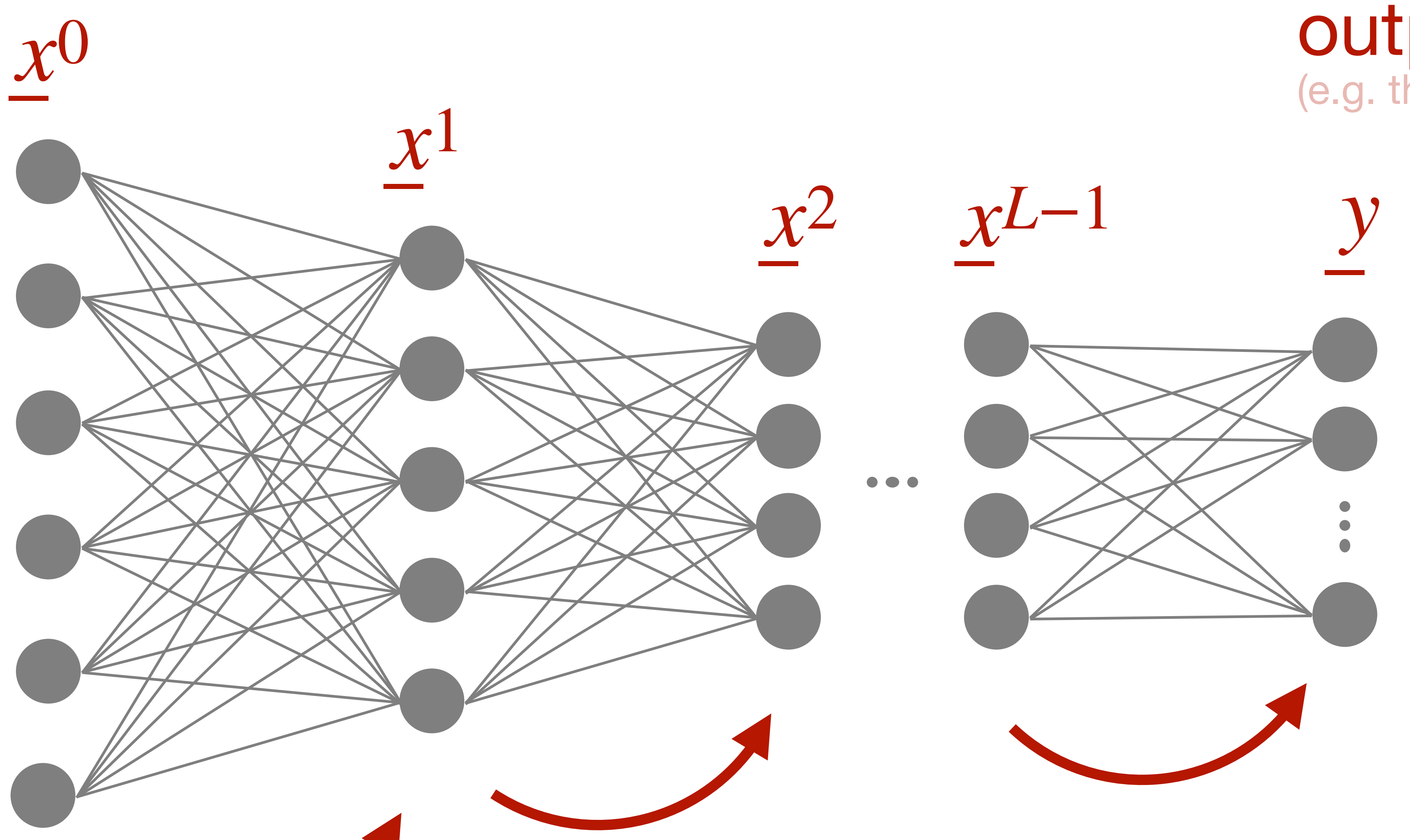
Naive deep learner: $\underline{x}^0 \in \mathcal{X} = \mathbb{R}^{784}$

Classical artificial neural networks

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



input vector



output probability vector
(e.g. through soft max)

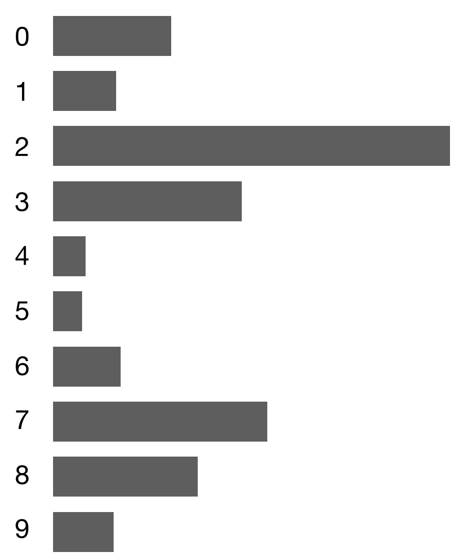
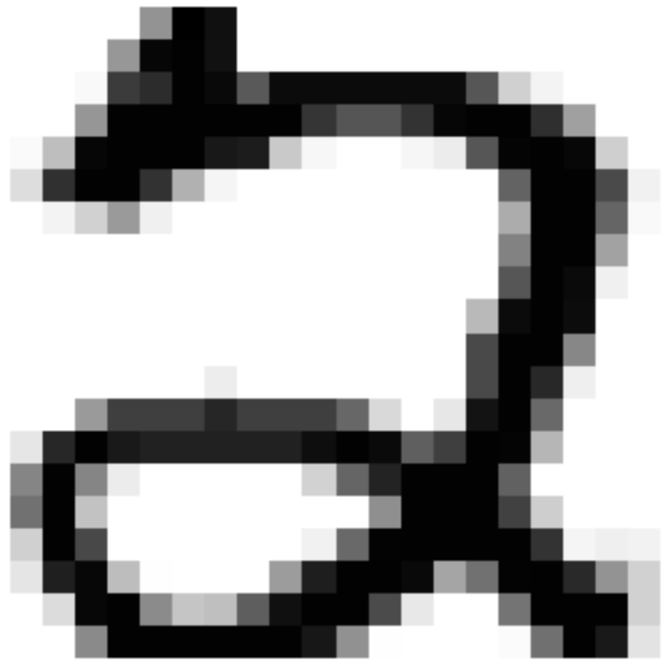


Image analyst: $\underline{x}^0 \in \mathcal{X} = \mathbb{L}_2(\mathbb{R}^2)$

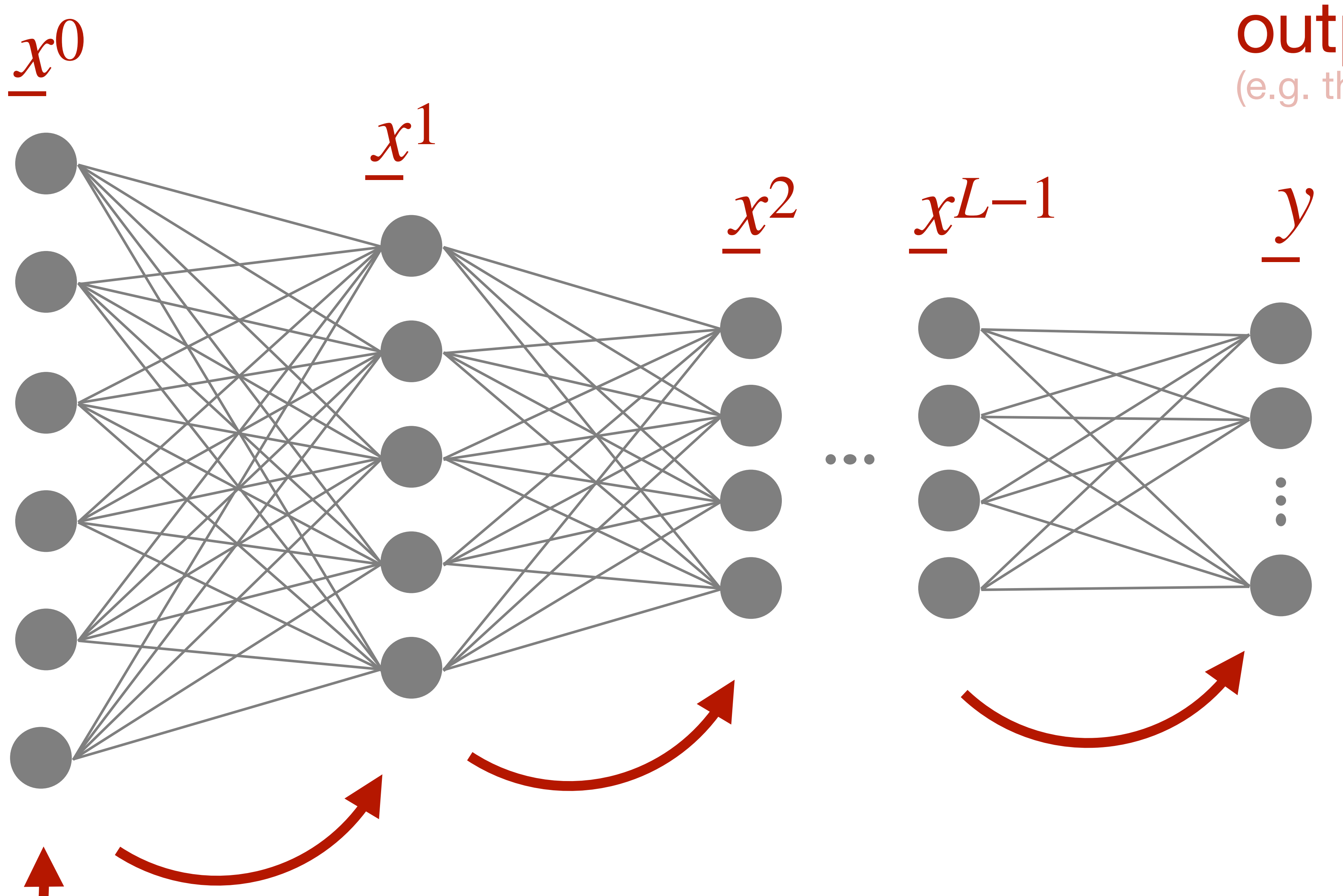
Naive deep learner: $\underline{x}^0 \in \mathcal{X} = \mathbb{R}^{784}$

Classical artificial neural networks

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



input vector



output probability vector
(e.g. through soft max)

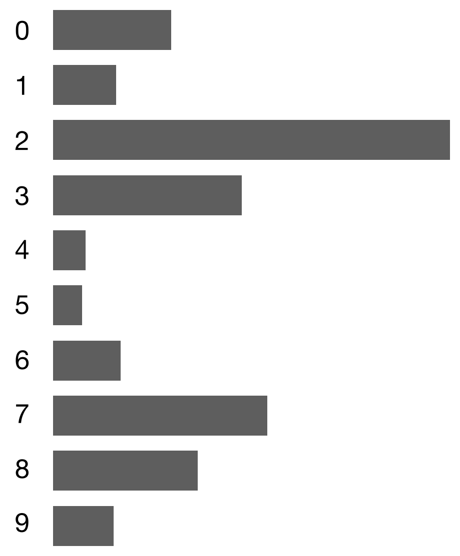


Image analyst: $\underline{x}^0 \in \mathcal{X} = \mathbb{L}_2(\mathbb{R}^2)$

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

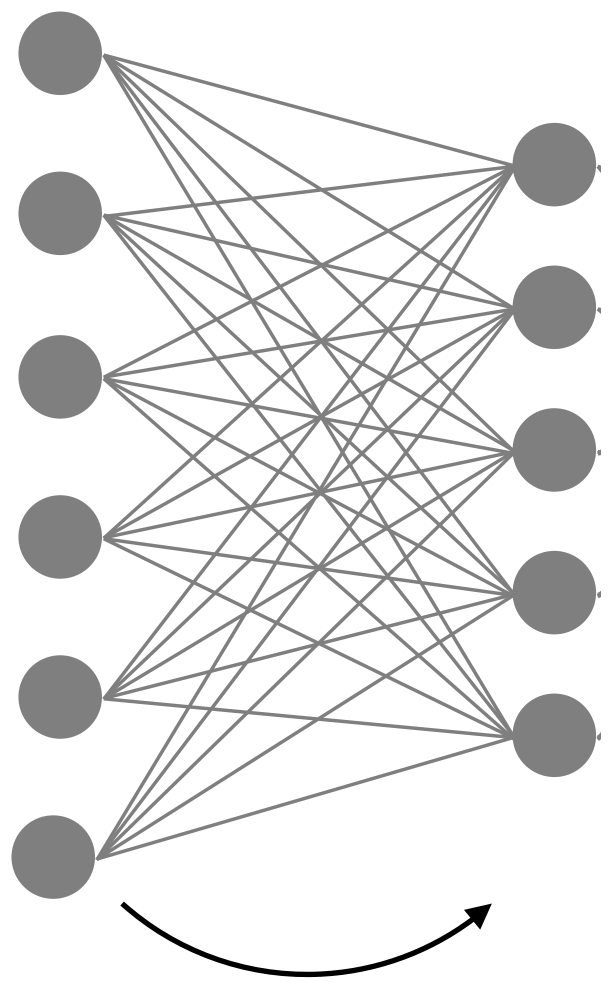
Iteratively transform the vector via

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

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



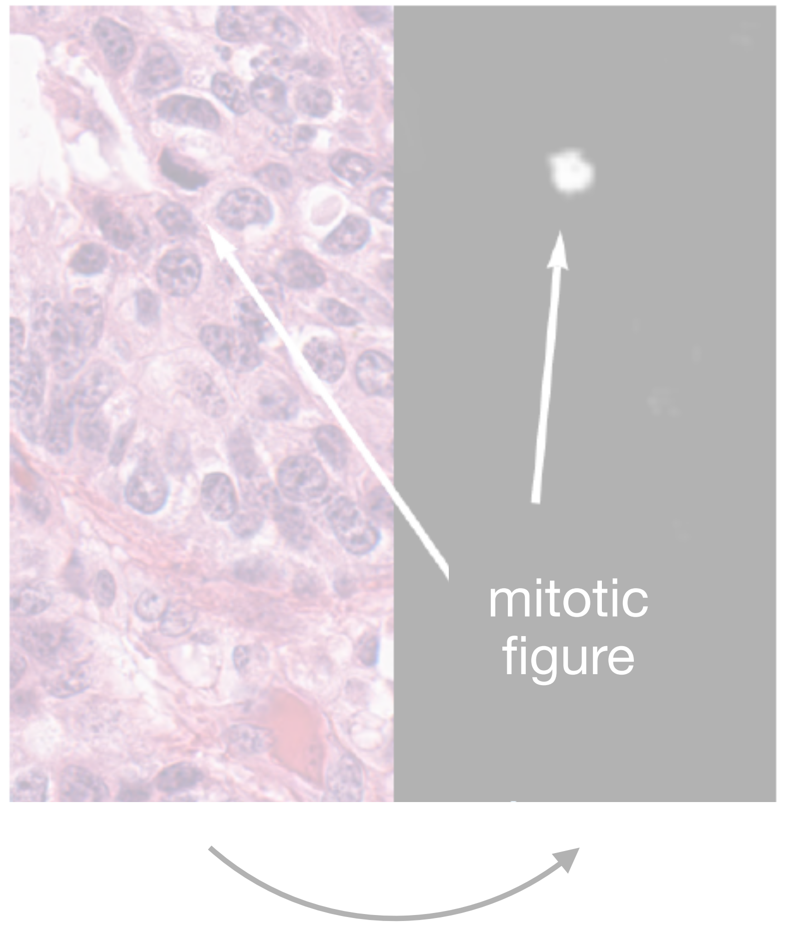
Iteratively transform the **vector** in $\mathbb{R}^{N^x}$ via

$$\underline{y} = \varphi(K\,\underline{x} + \underline{b}^l)$$

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 $\mathbb{L}_2(X)$

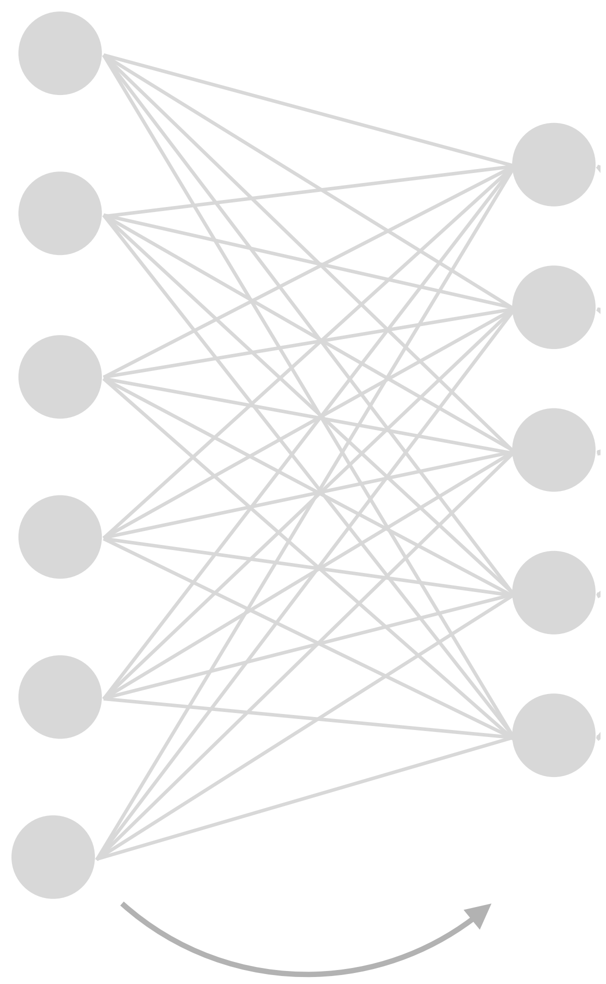
$$f^{out} = \varphi(K f^{in} + b^l)$$

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

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

Classical artificial NNs in the continuous world

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



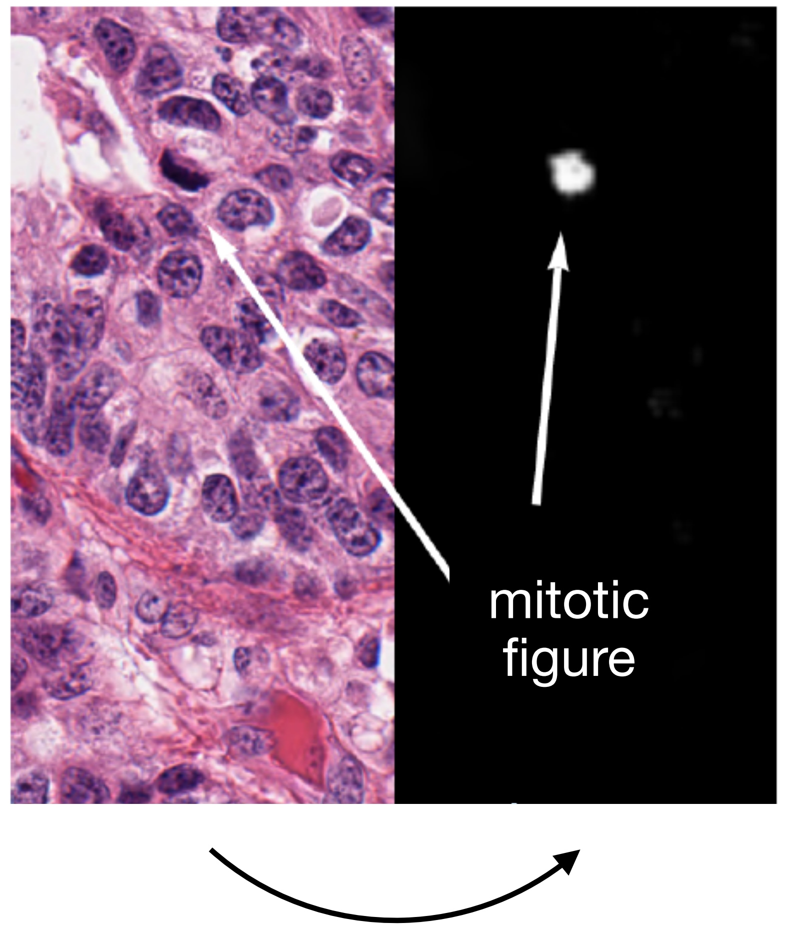
Iteratively transform the **vector** in $\mathbb{R}^{N^x}$ via

$$\underline{y} = \varphi(K\,\underline{x} + \underline{b}^l)$$

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 $\mathbb{L}_2(X)$

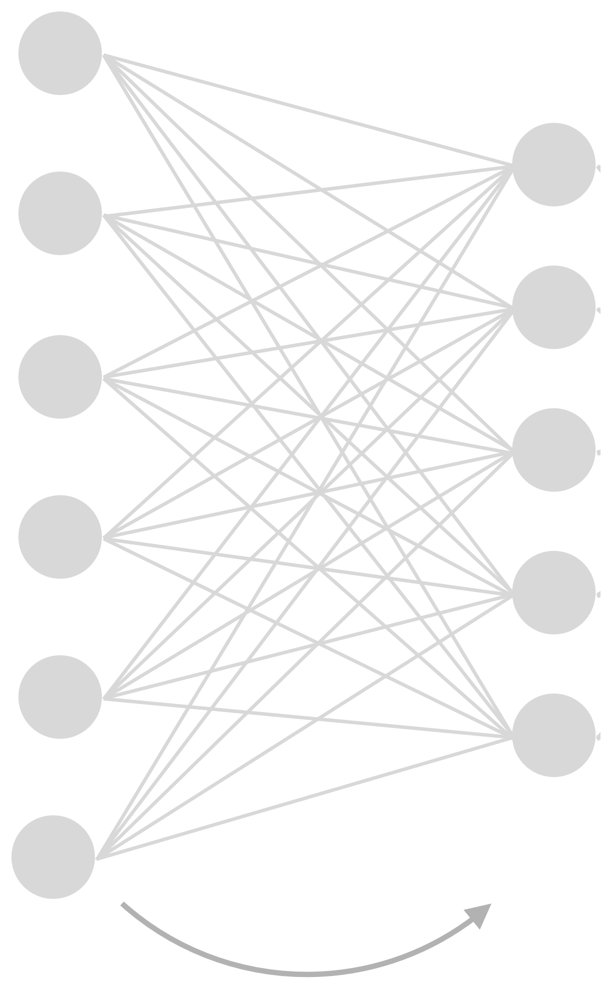
$$f^{out} = \varphi(K\,f^{in} + b^l)$$

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

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

Classical artificial NNs in the continuous world

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



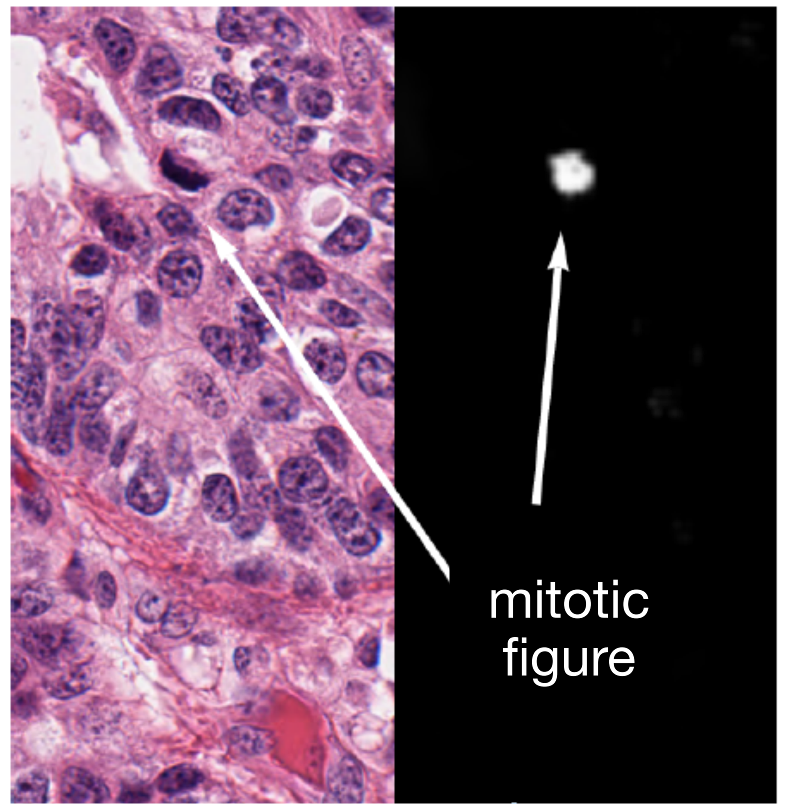
Iteratively transform the **vector** in $\mathbb{R}^{N^x}$ via

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 $f \in \mathcal{X} = \mathbb{L}_2(X)$



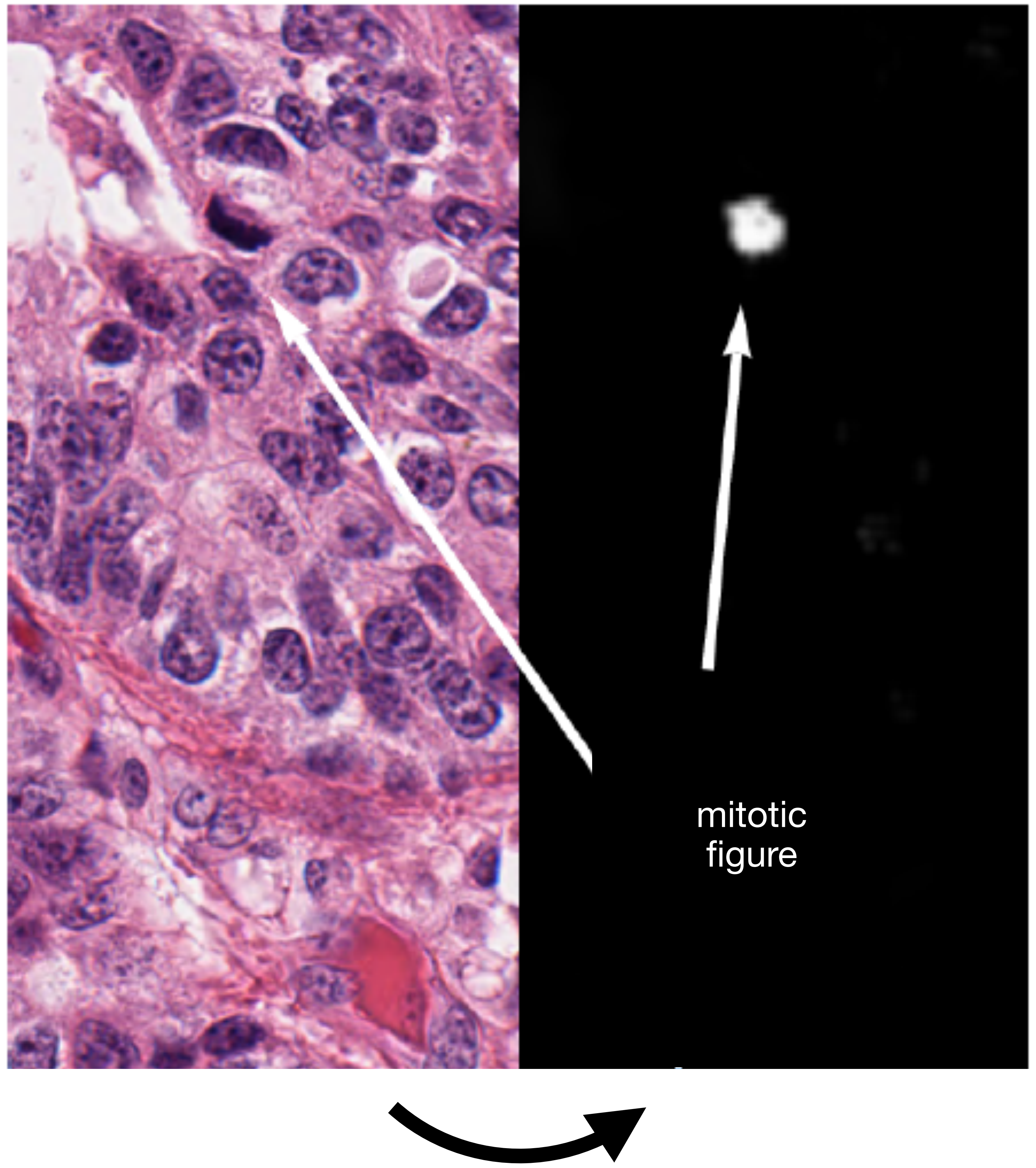
Iteratively transform the **feature map** in $\mathbb{L}_2(X)$

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

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



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

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

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

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

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

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 $\mathscr{K} : \mathbb{L}_2(X) \rightarrow \mathbb{L}_2(Y)$ 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:

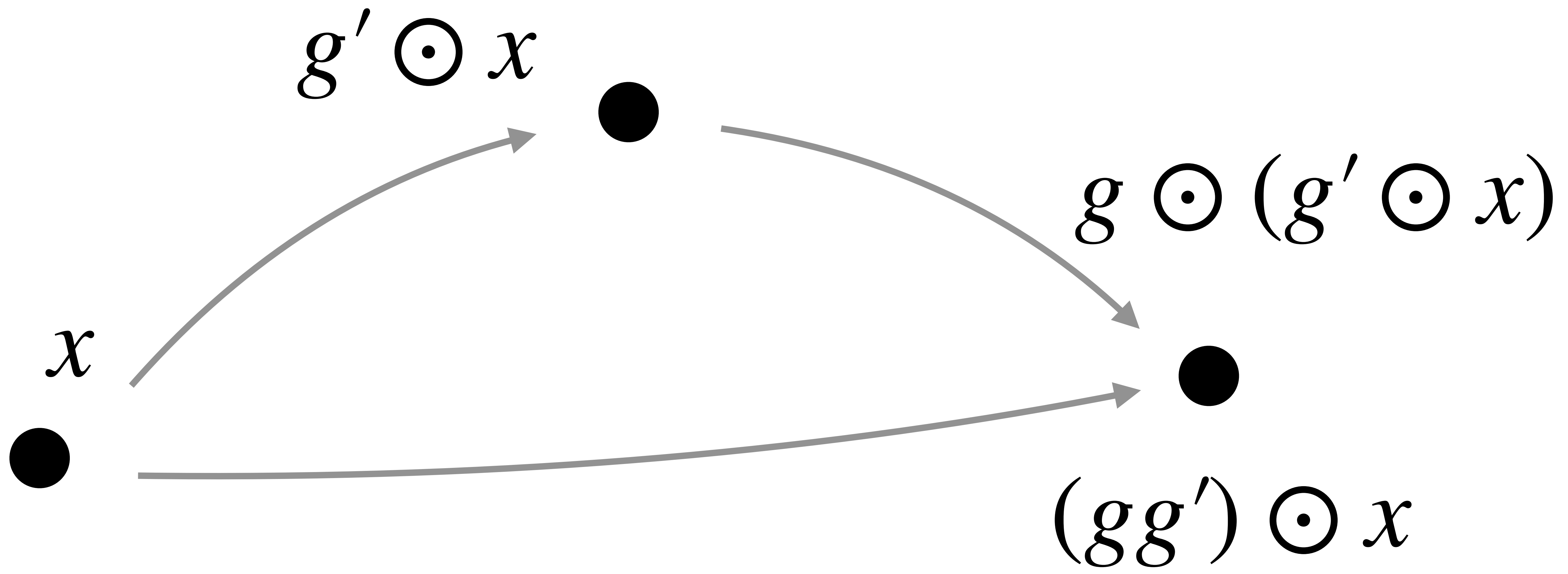
1. It is a group convolution: $[\mathscr{K} f](y) = \int_X k(g_y^{-1} x) f(x) \mathrm{d}x$

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

Group theory: Homogeneous spaces

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

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

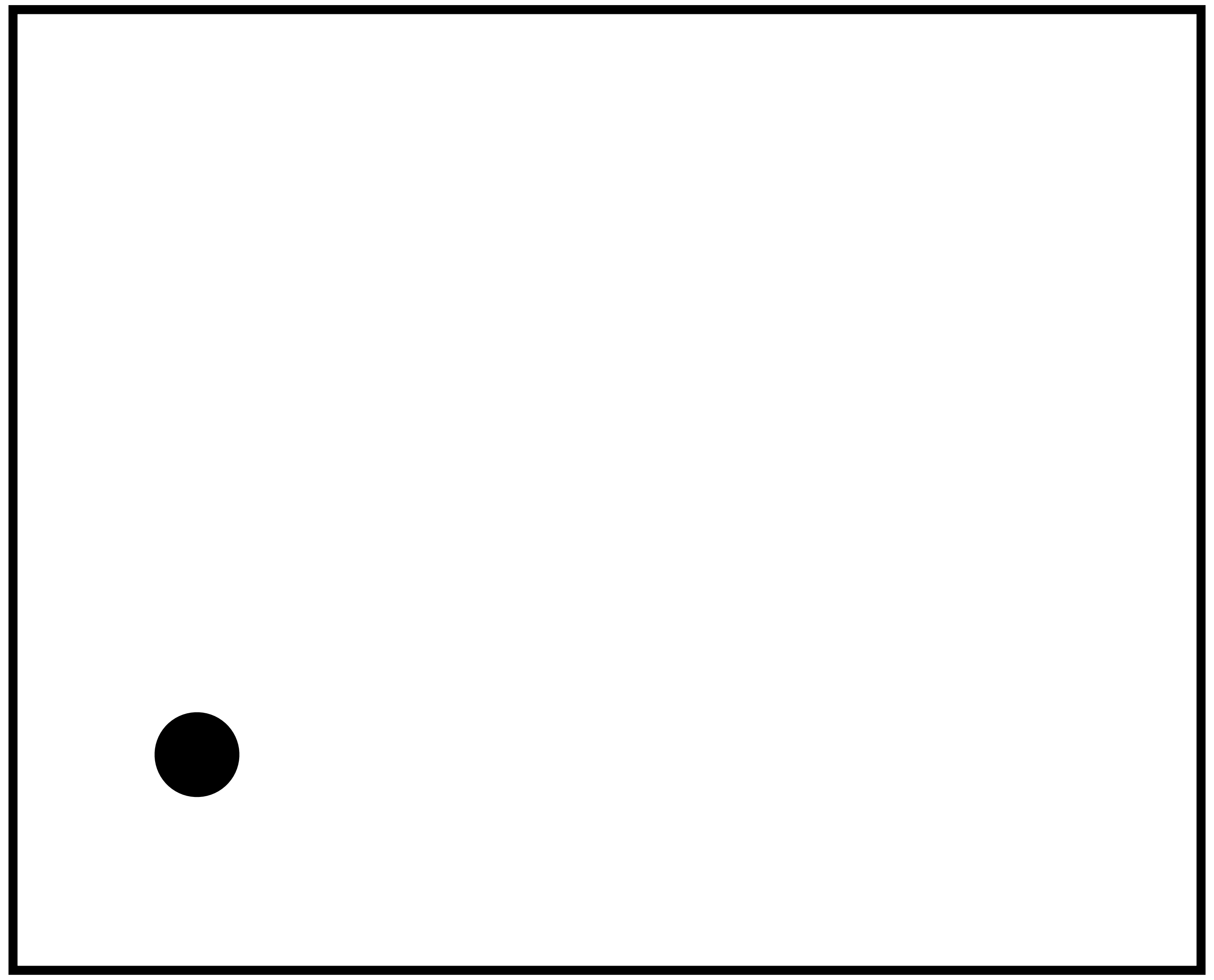


Group theory: Homogeneous spaces

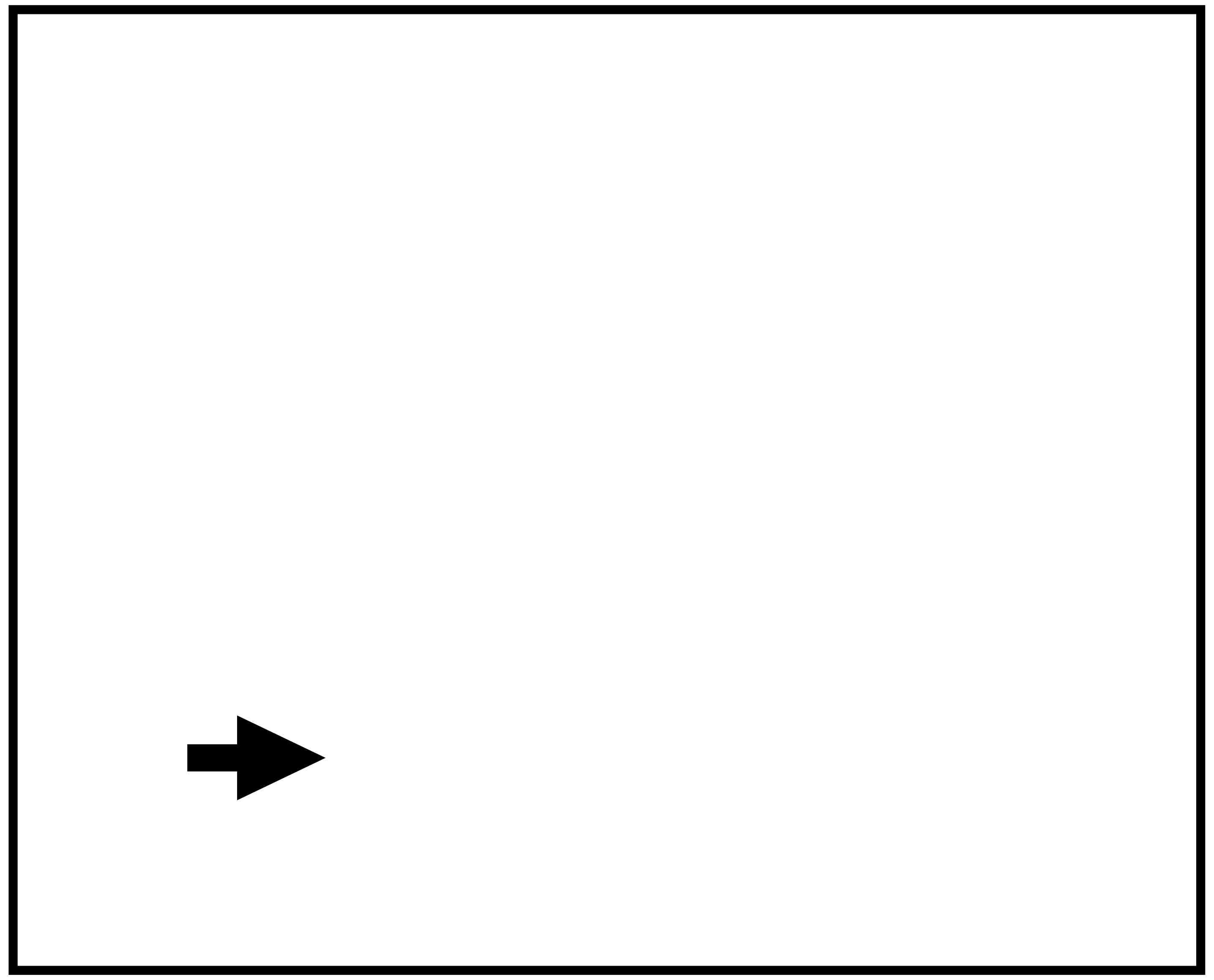
Transitive action: An action $\odot : G \times X \rightarrow X$ such that

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

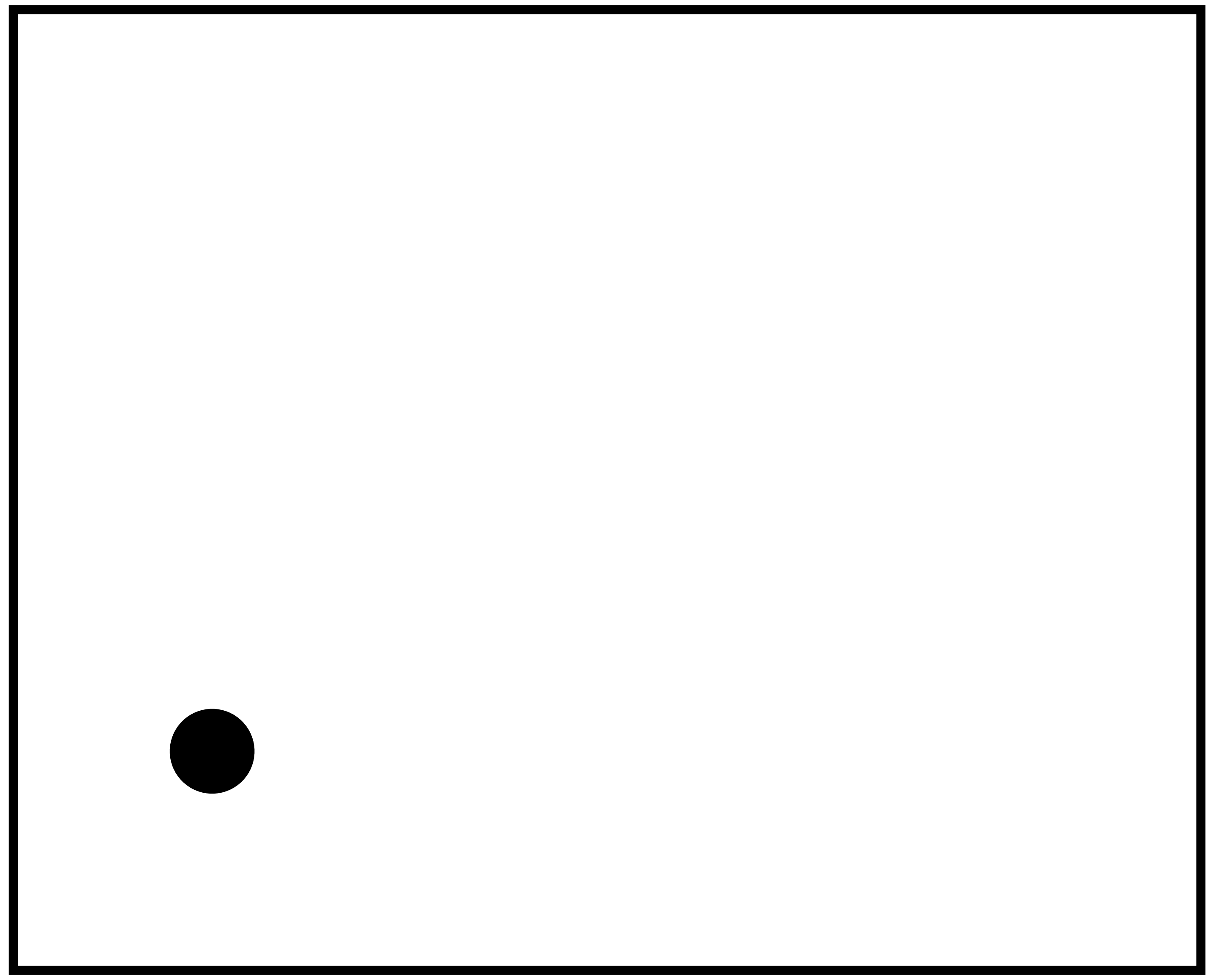
$(\mathbb{R}^2, +)$ acts transitively on $\mathbb{R}^2$



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



SO(2) does not …



Group theory: Homogeneous spaces

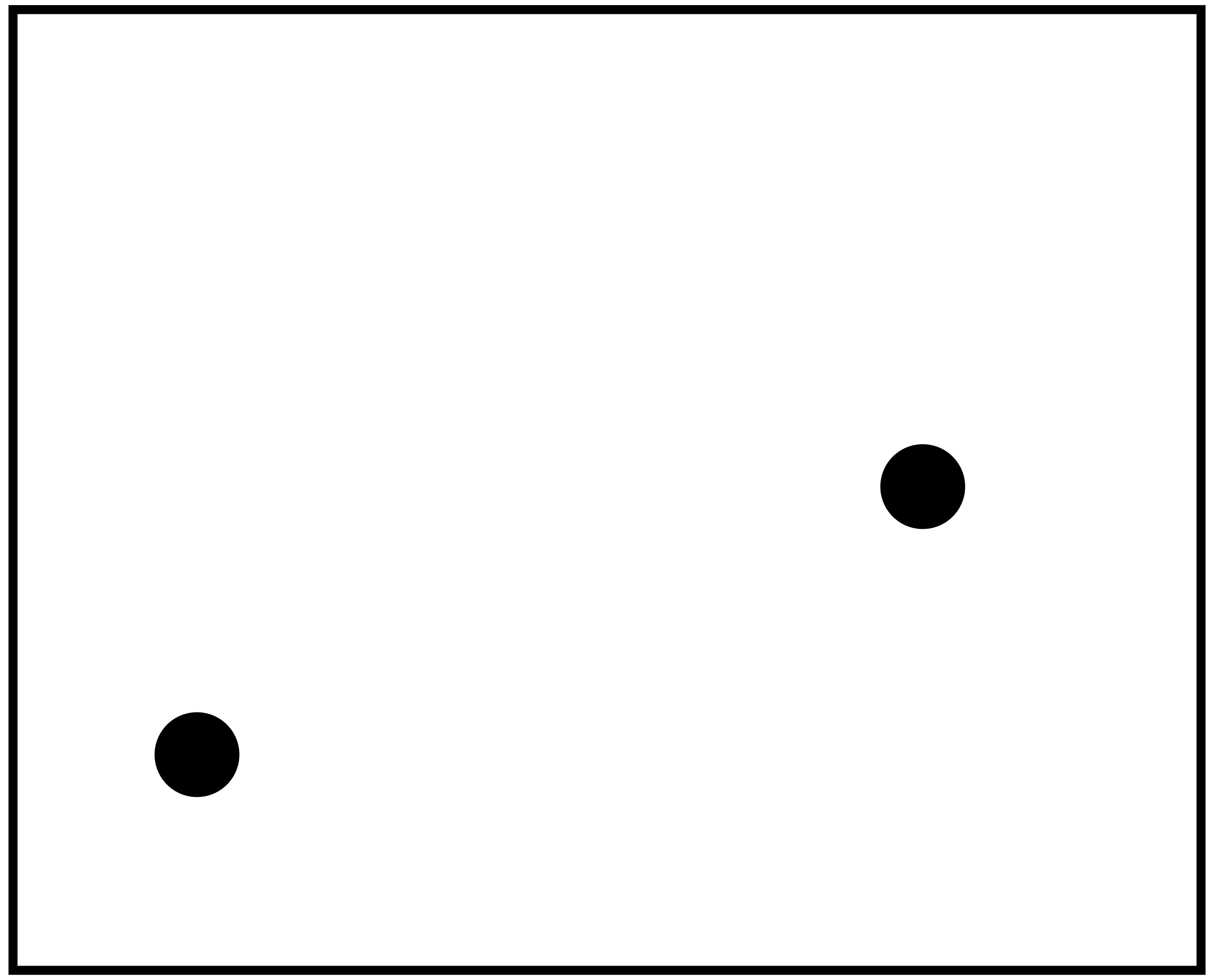
Transitive action: An action $\odot : G \times X \to X$ such that

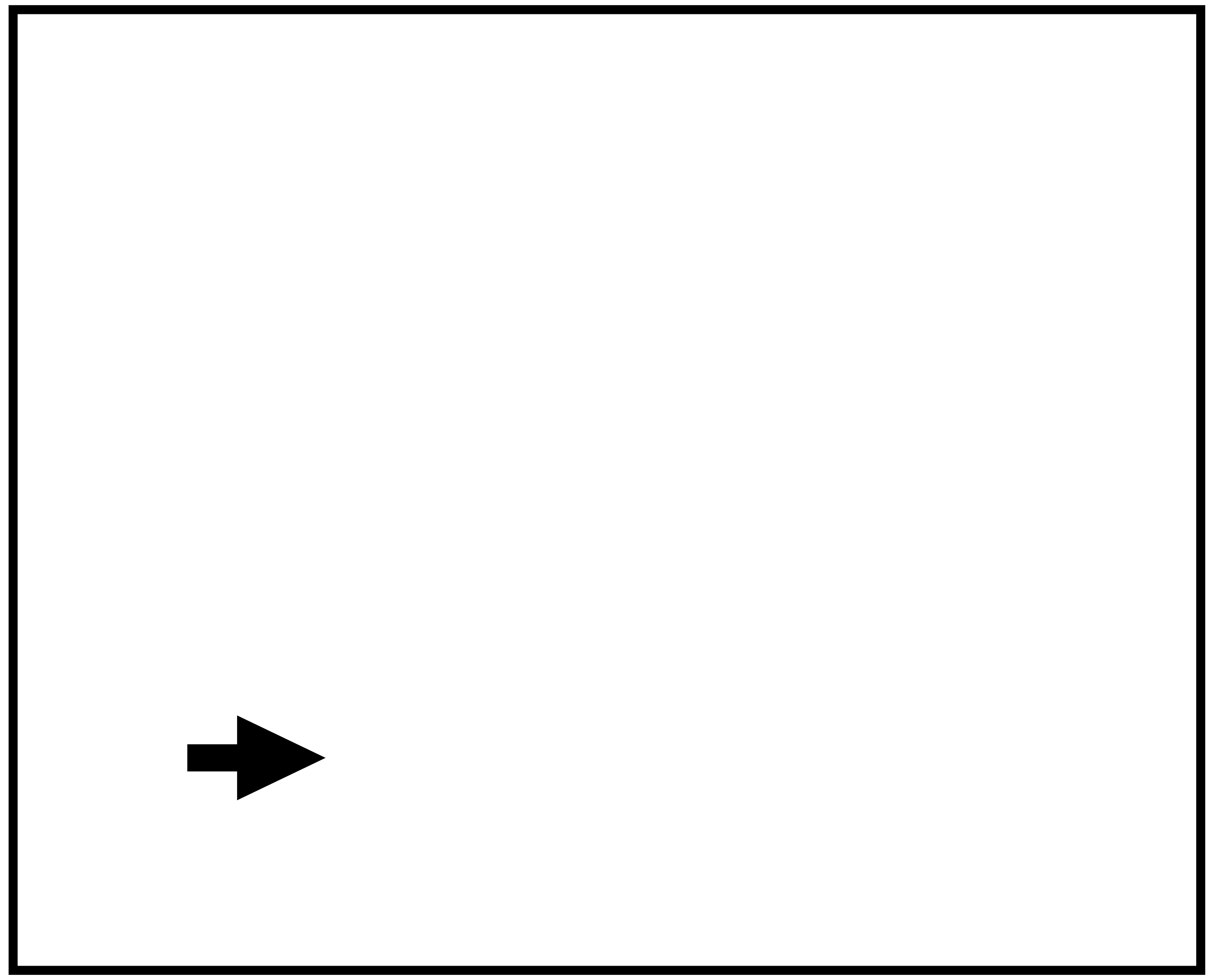
$$\forall_{x_0, x \in X} \exists_{g \in G} : x = g \odot x_0$$

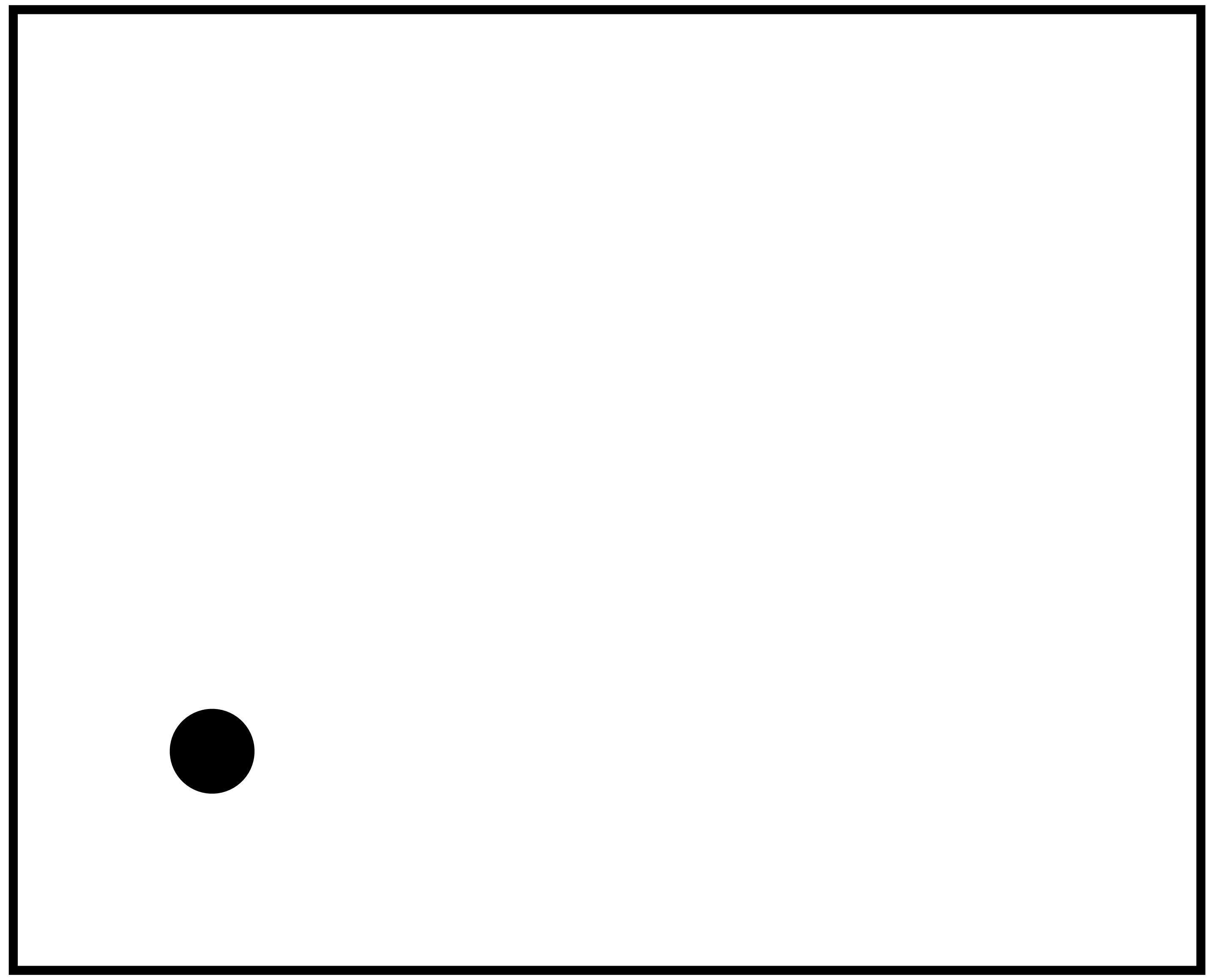
$(\mathbb{R}^2, +)$ acts transitively on $\mathbb{R}^2$

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

SO(2) does not …





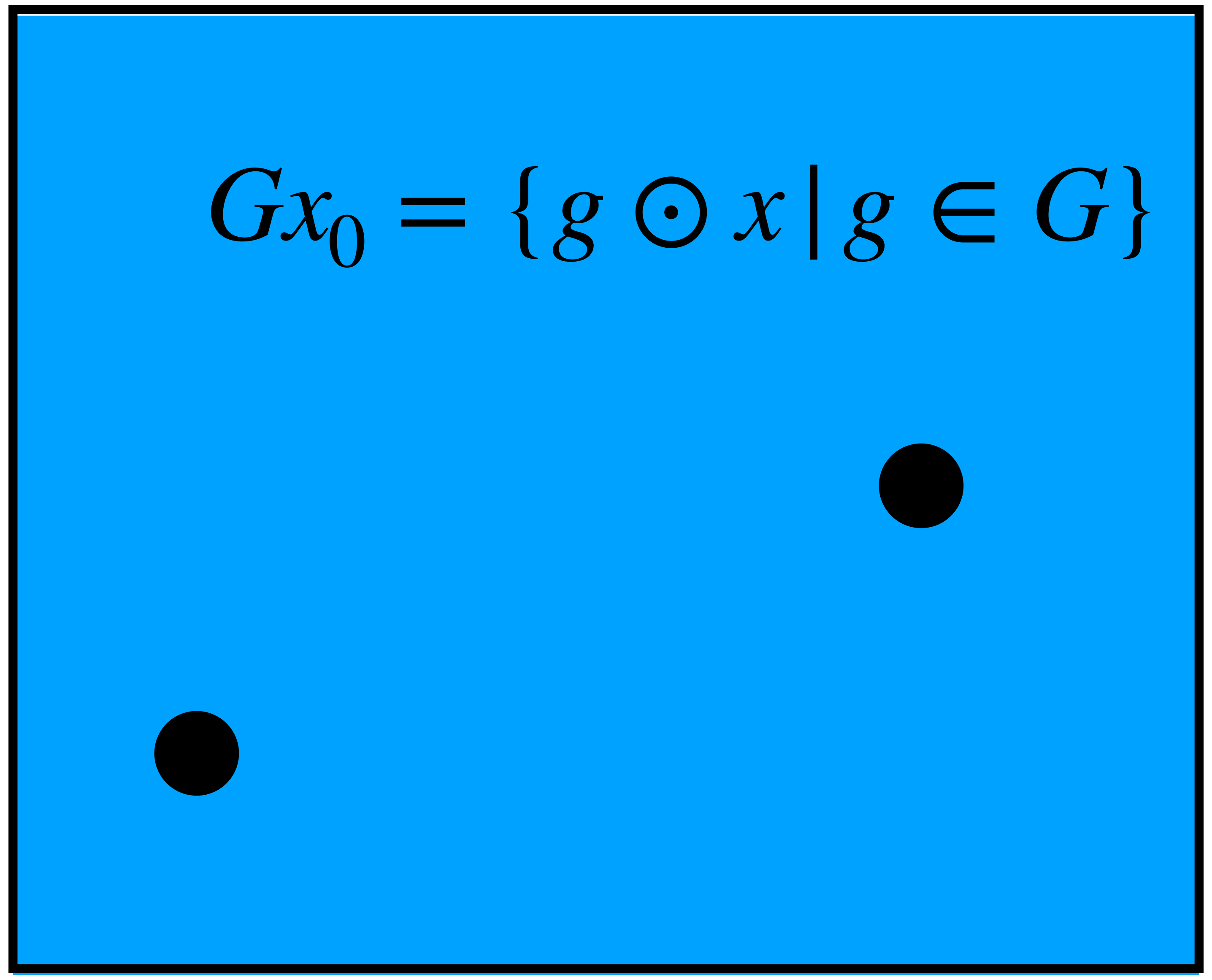


Group theory: Homogeneous spaces

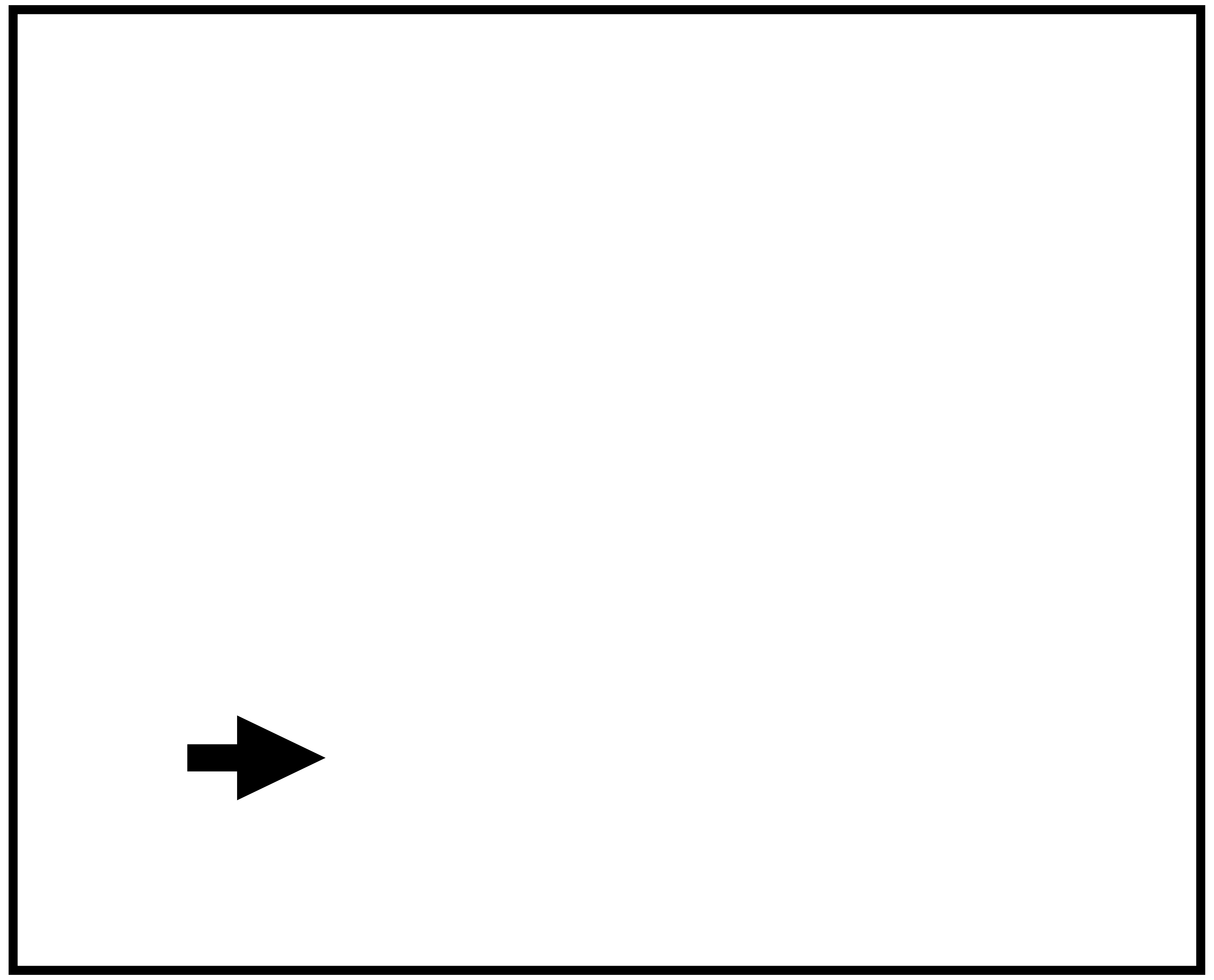
Transitive action: An action $\odot : G \times X \rightarrow X$ such that

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

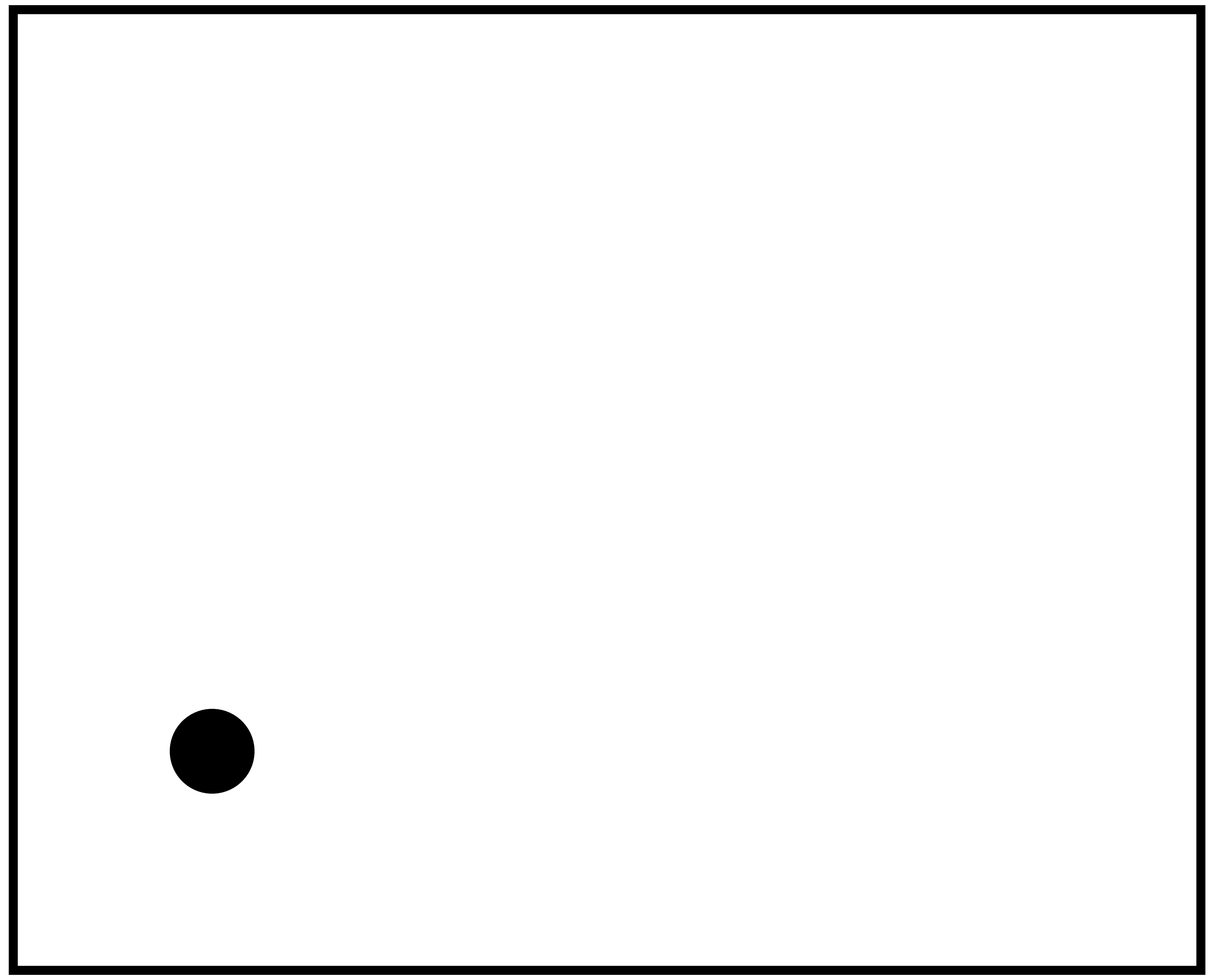
$(\mathbb{R}^2, +)$ acts transitively on $\mathbb{R}^2$



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



SO(2) does not …

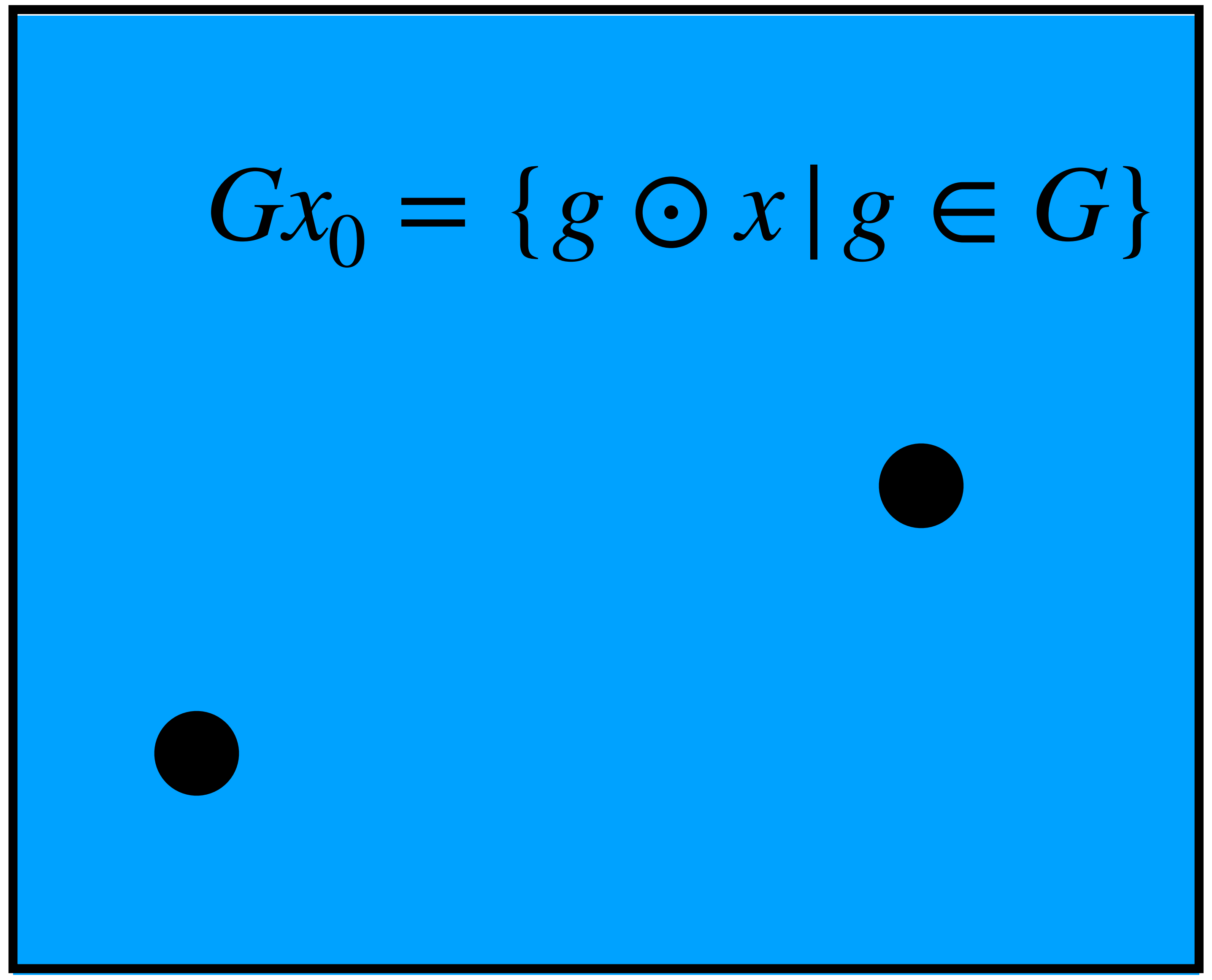


Group theory: Homogeneous spaces

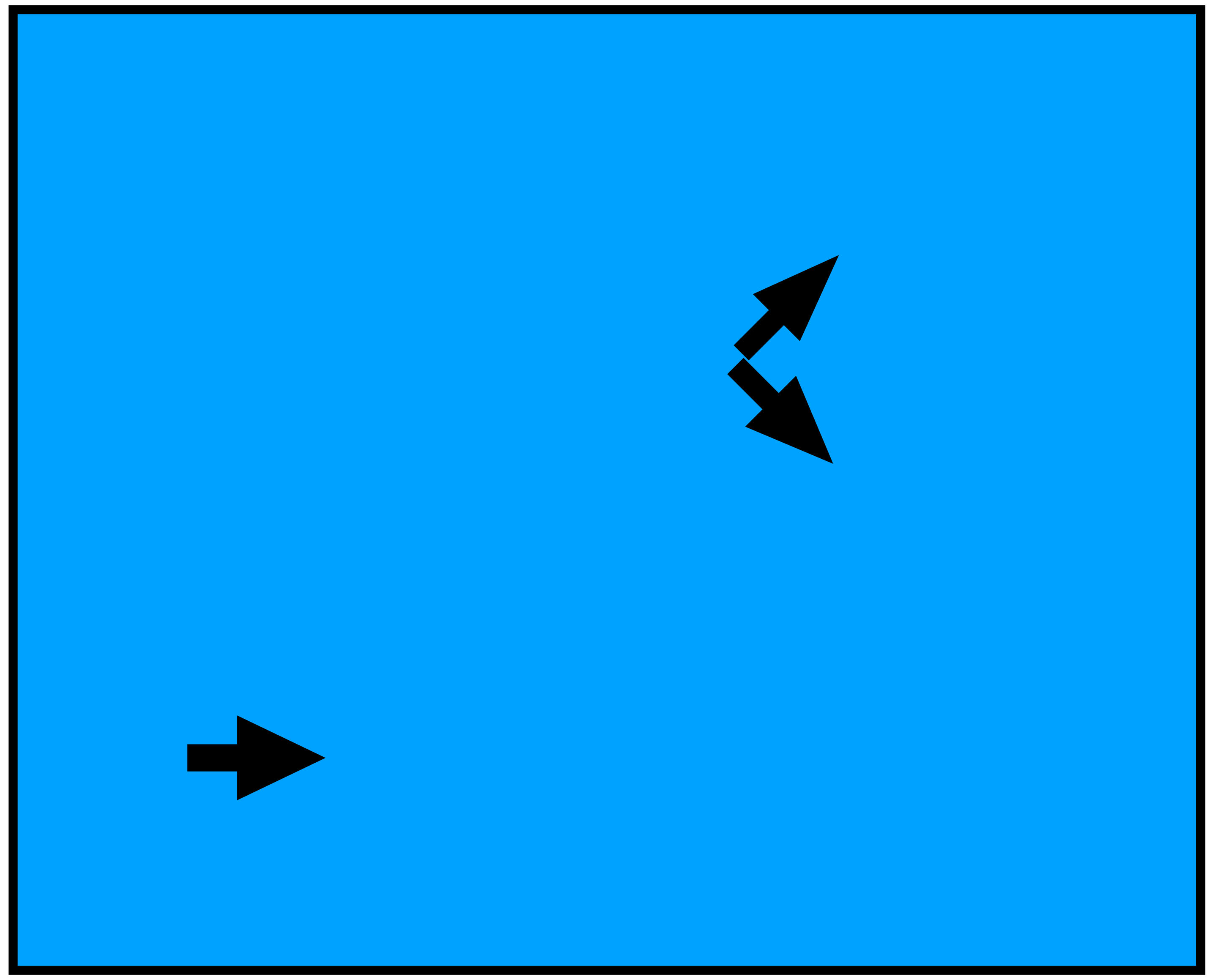
Transitive action: An action $\odot : G \times X \rightarrow X$ such that

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

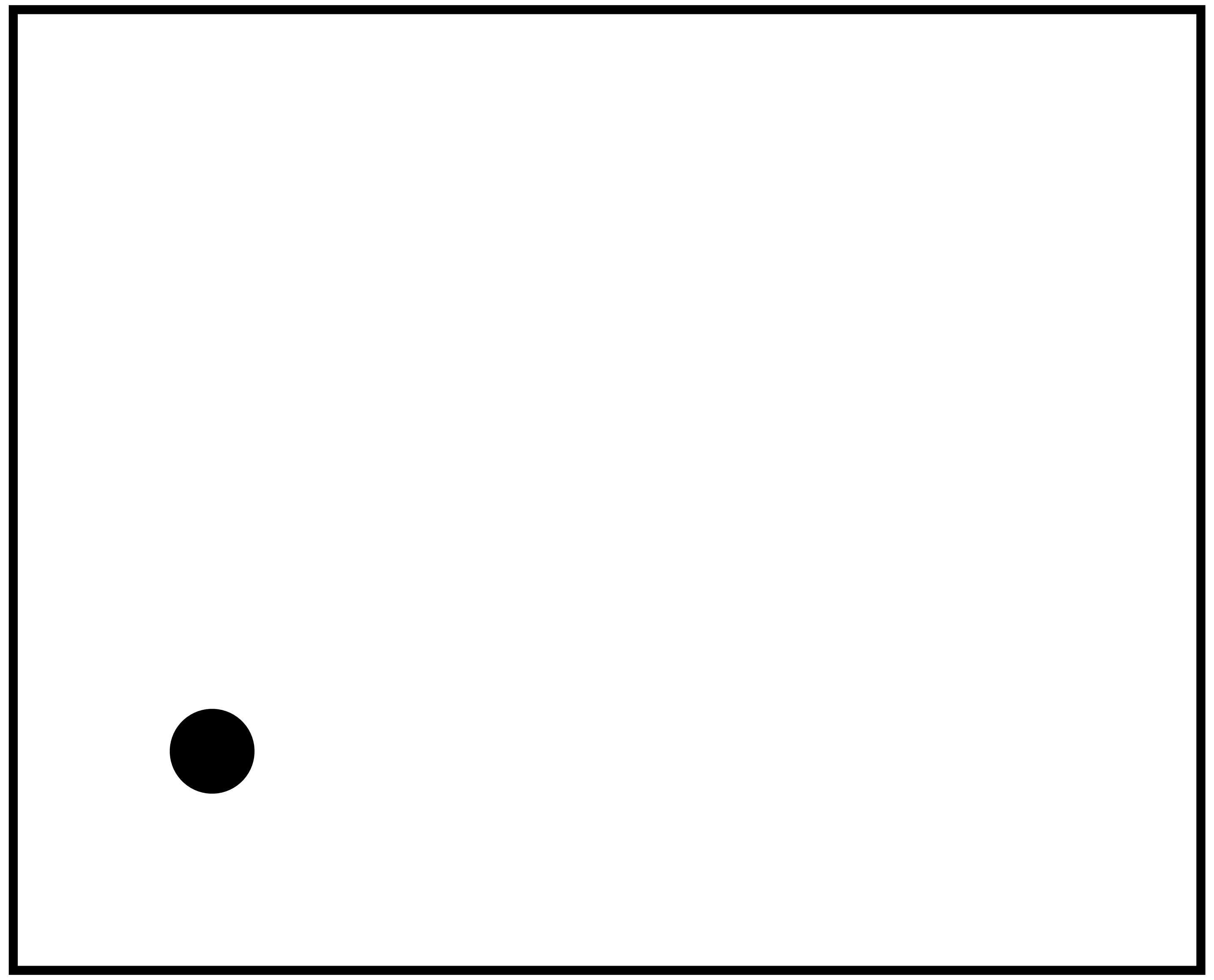
$(\mathbb{R}^2, +)$ acts transitively on $\mathbb{R}^2$



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



SO(2) does not …

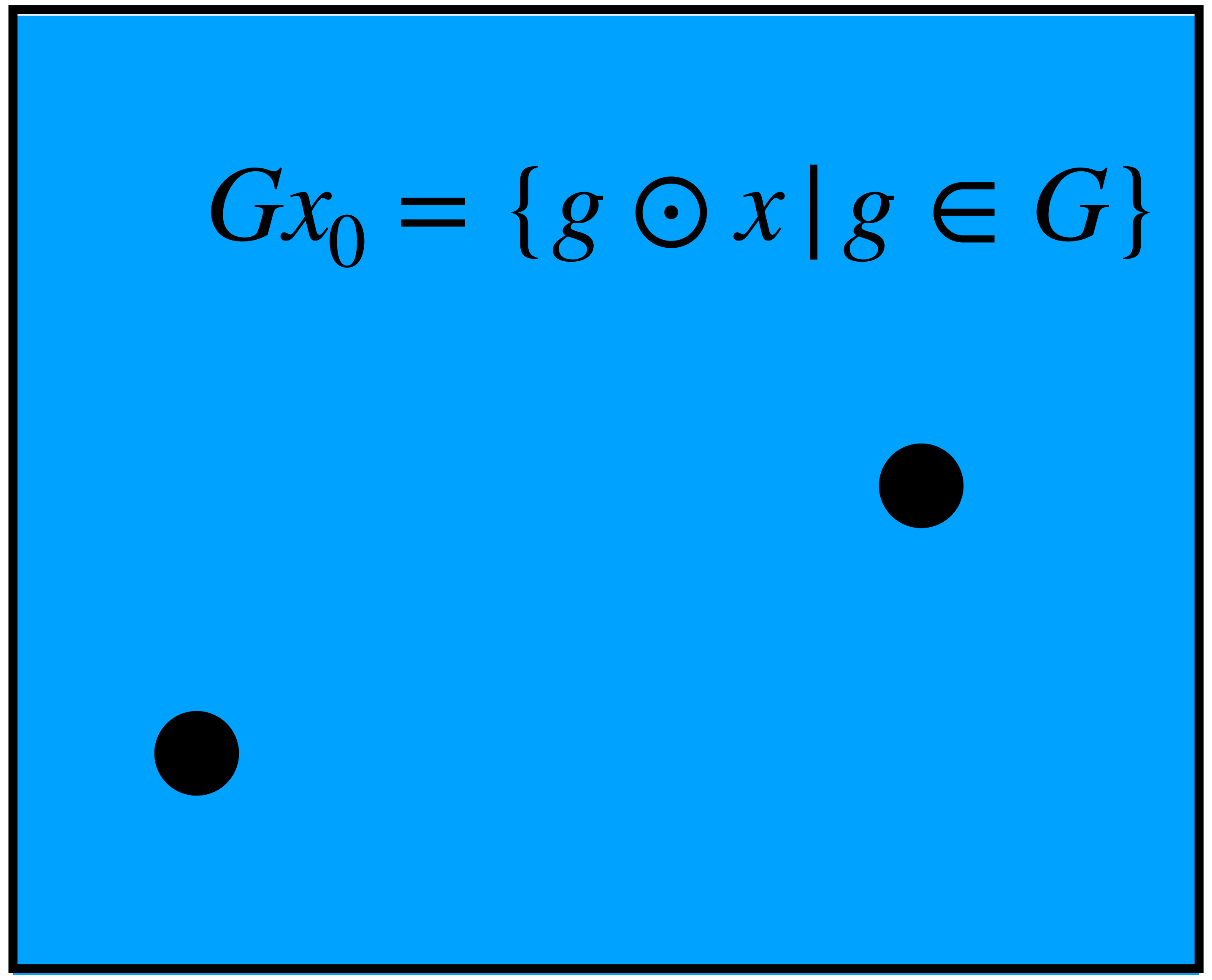


Group theory: Homogeneous spaces

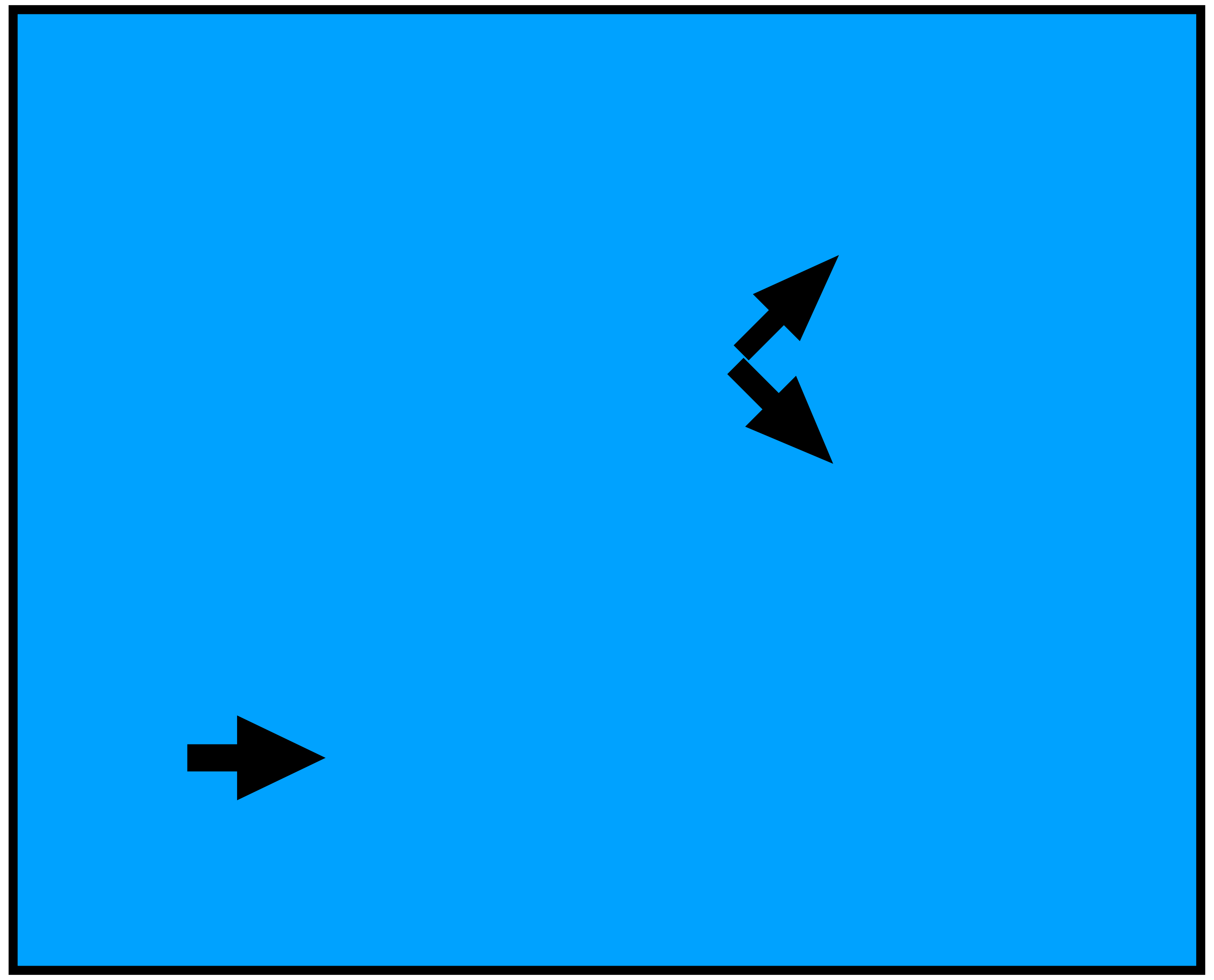
Transitive action: An action $\odot : G \times X \rightarrow X$ such that

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

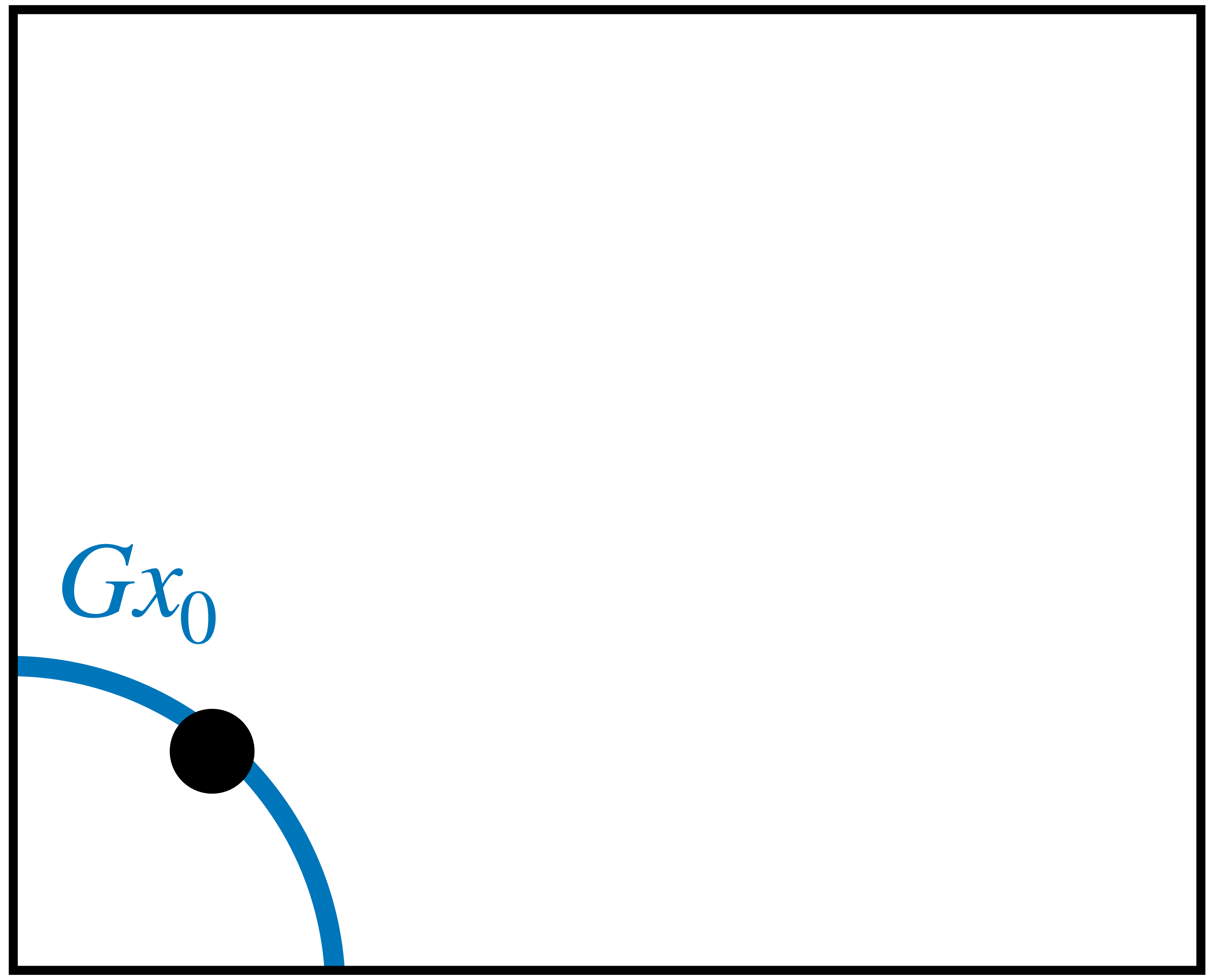
$(\mathbb{R}^2, +)$ acts transitively on $\mathbb{R}^2$



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



SO(2) does not …

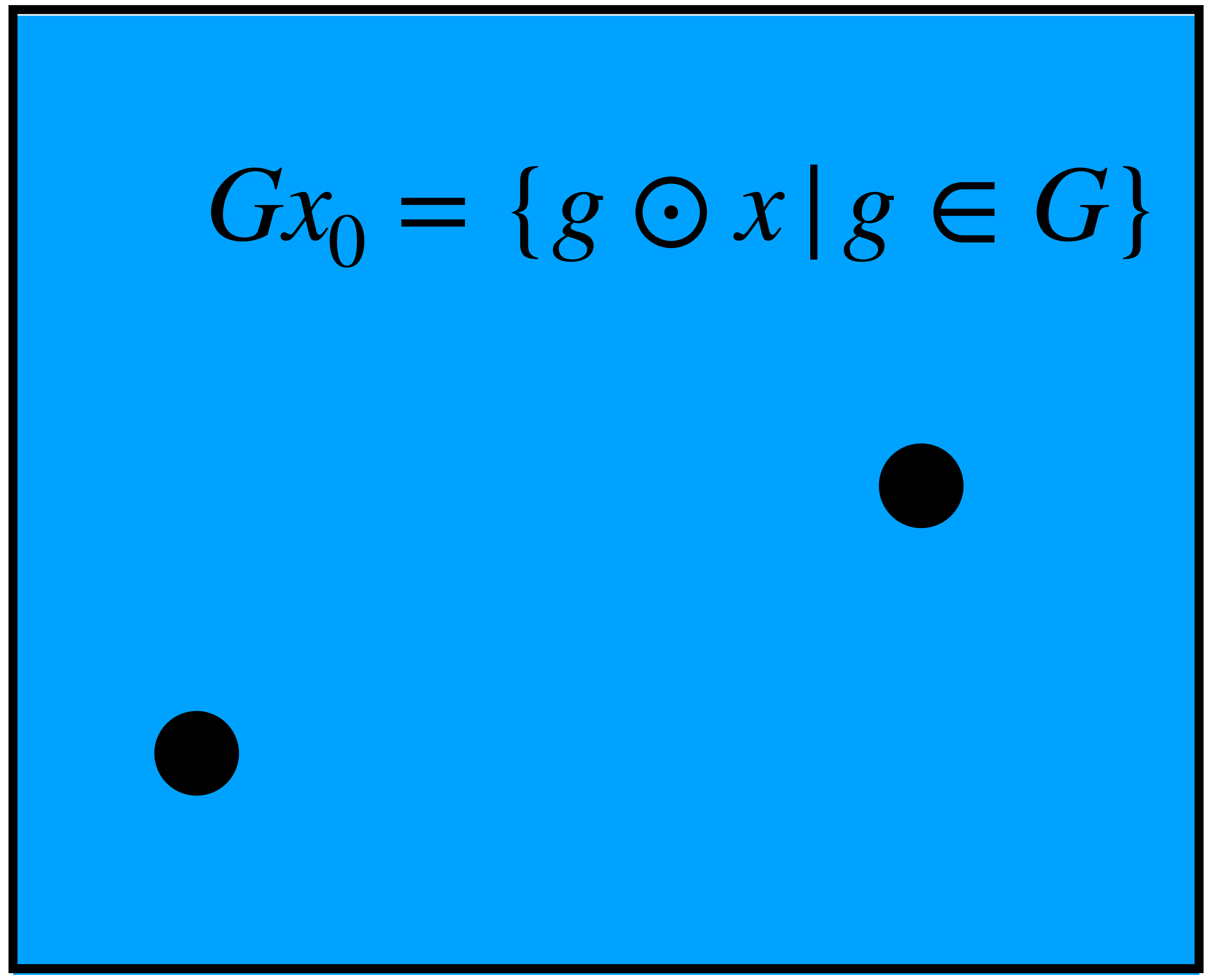


Group theory: Homogeneous spaces

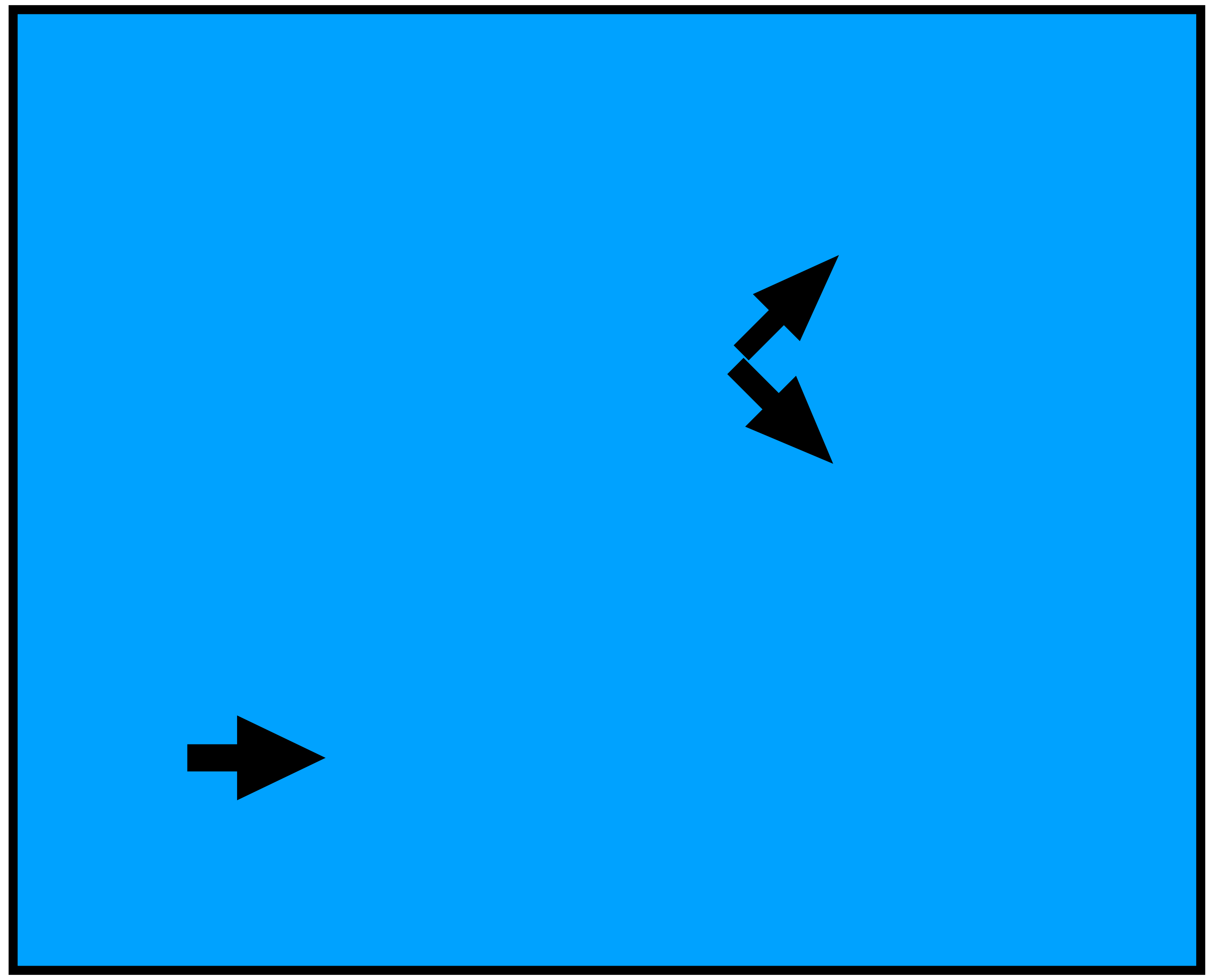
Transitive action: An action $\odot : G \times X \to X$ such that

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

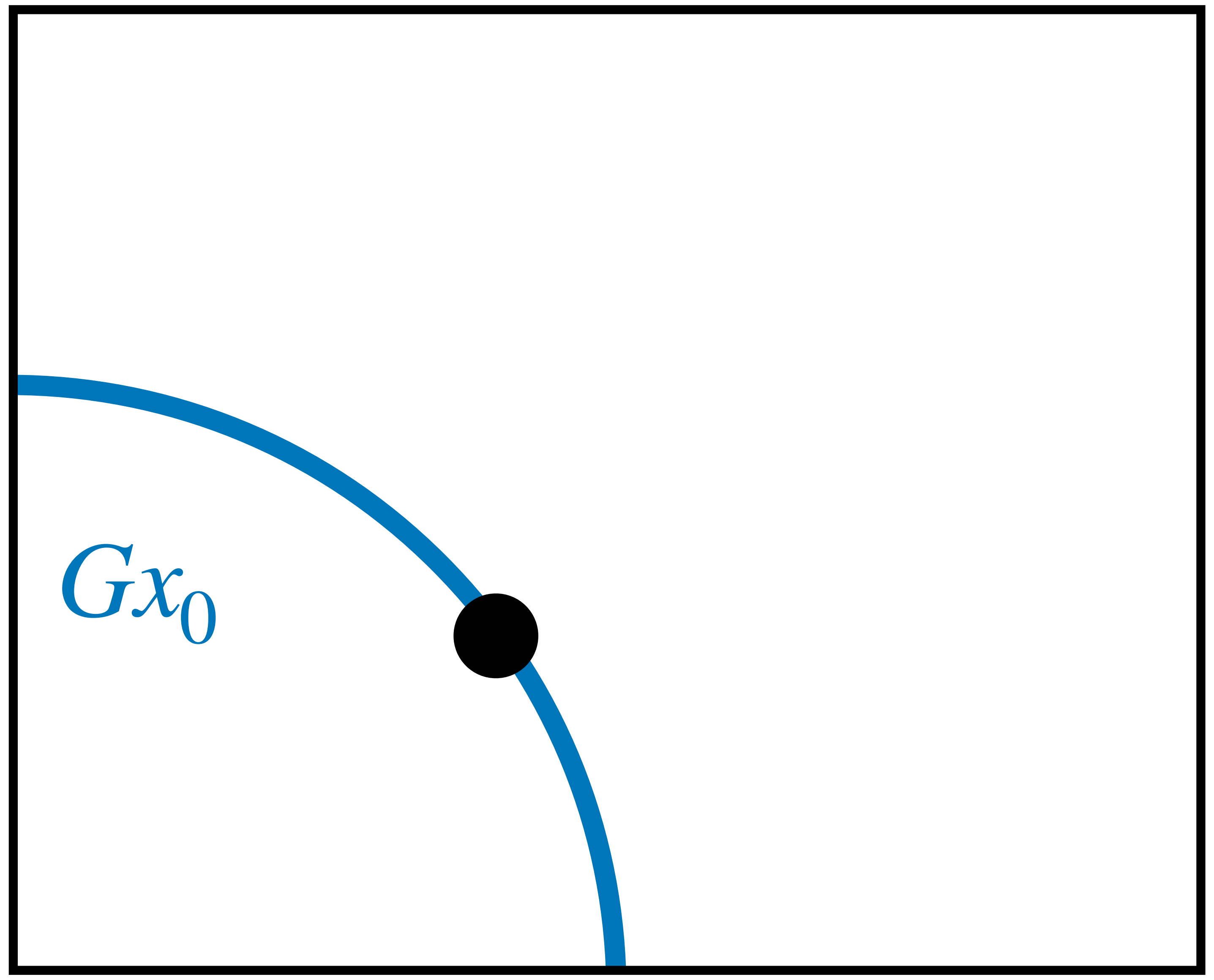
$(\mathbb{R}^2, +)$ acts transitively on $\mathbb{R}^2$



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



SO(2) does not …

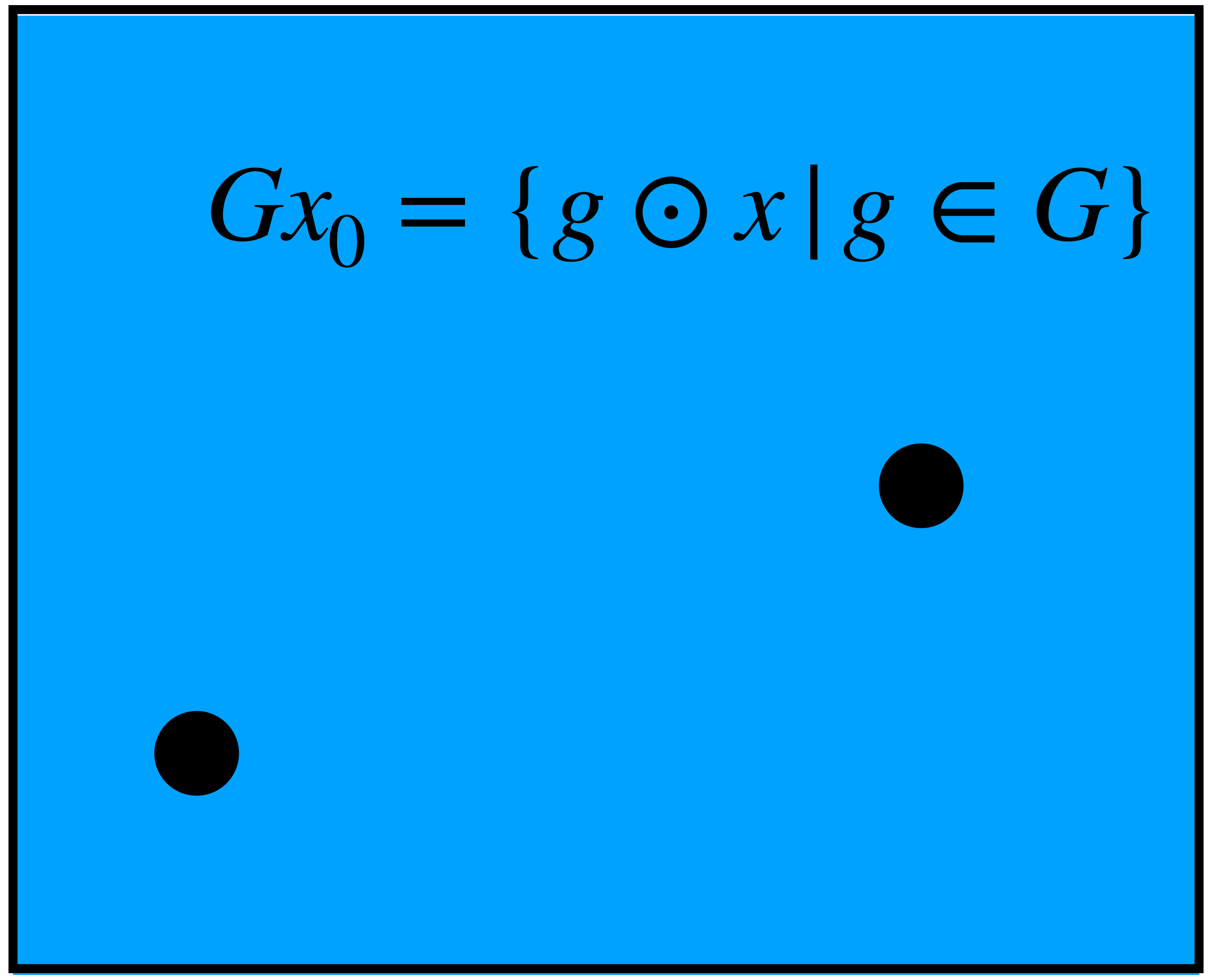


Group theory: Homogeneous spaces

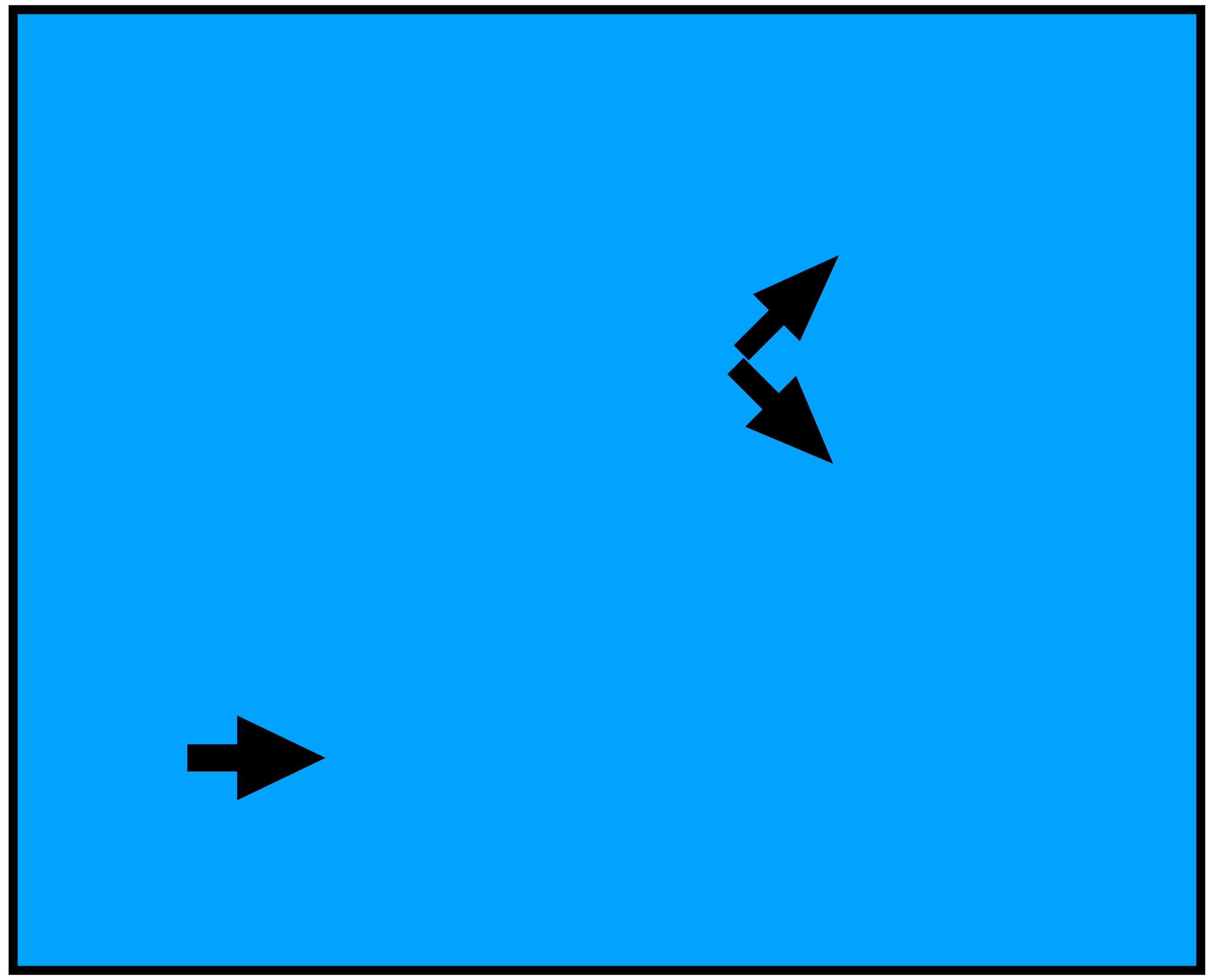
Transitive action: An action $\odot : G \times X \rightarrow X$ such that

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

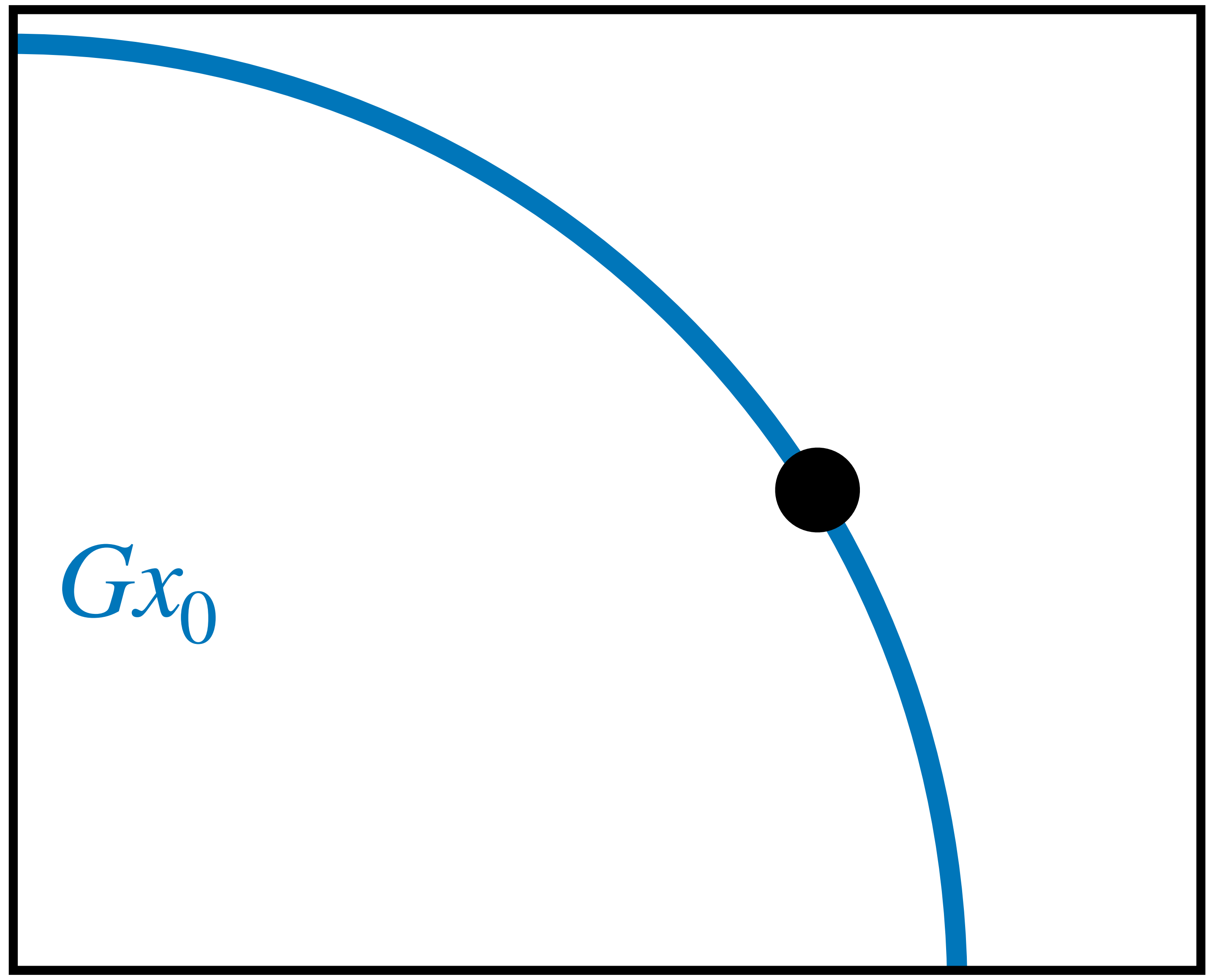
$(\mathbb{R}^2, +)$ acts transitively on $\mathbb{R}^2$



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



SO(2) does not …



Group theory: Homogeneous spaces

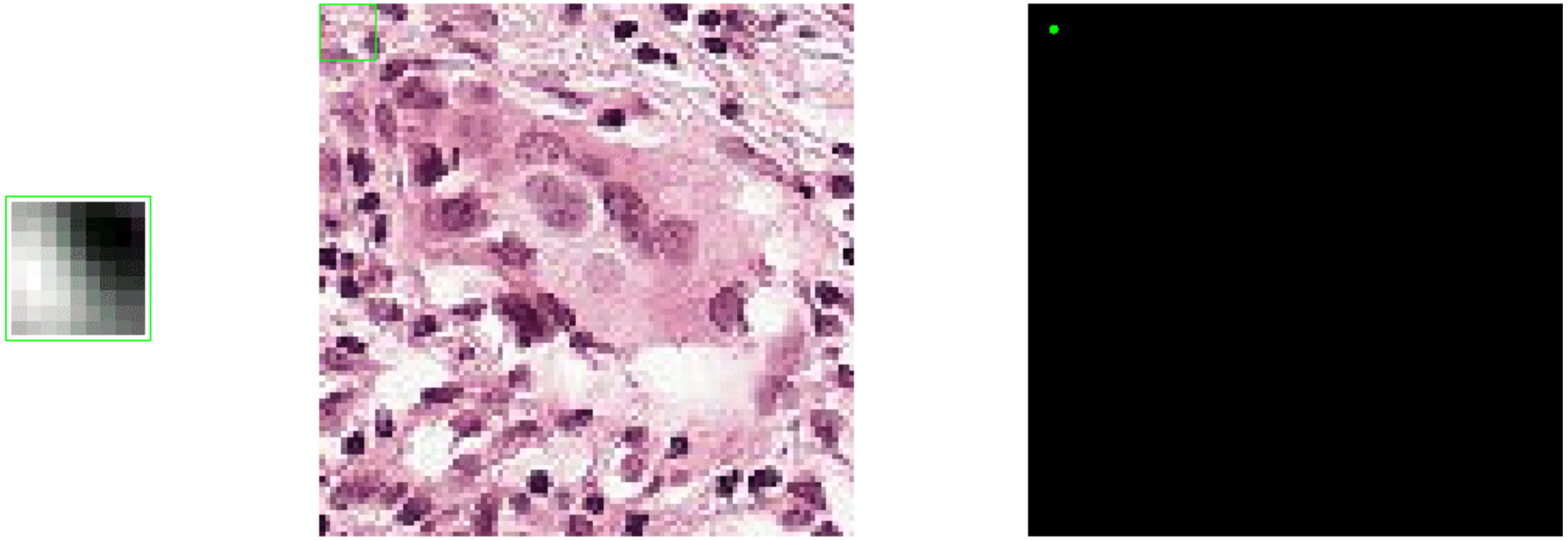
Homogeneous space: A space on X on which G acts transitively.

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

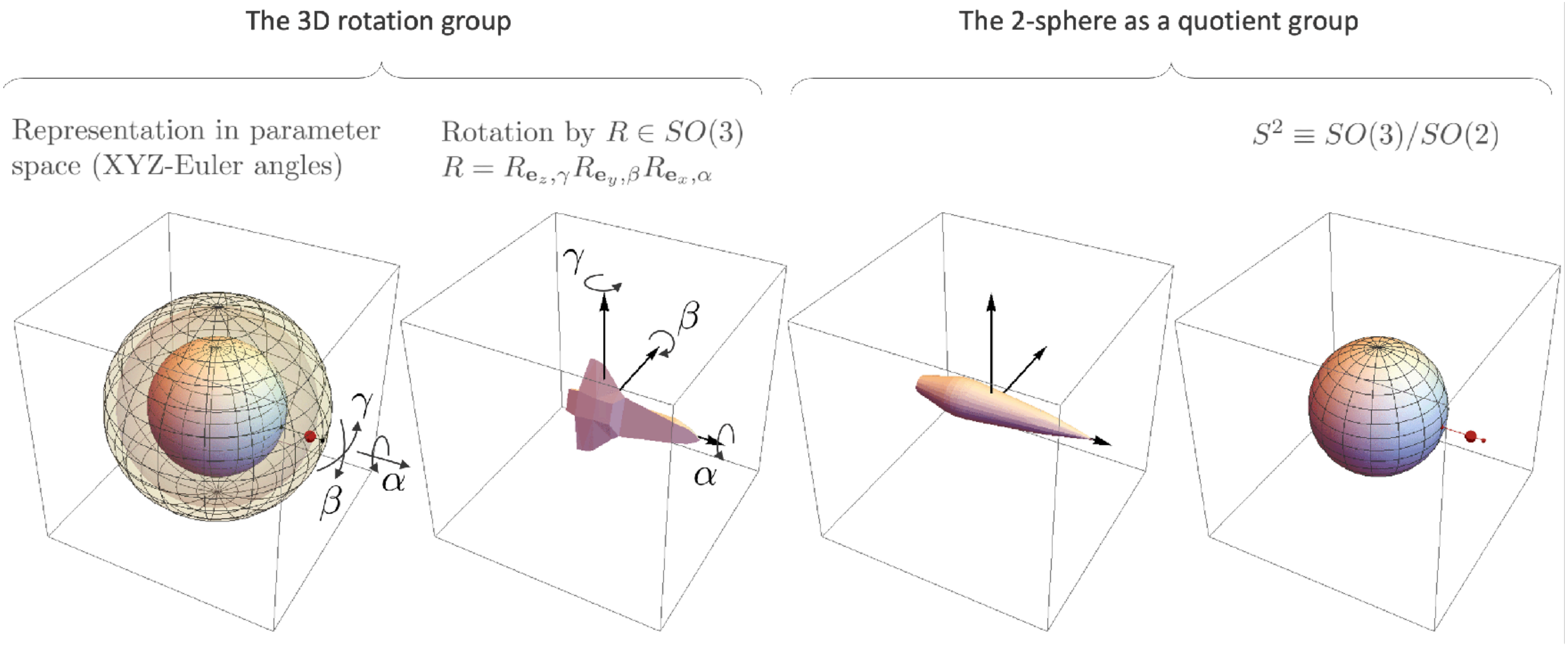
Homogeneous space: A space on X on which G acts transitively.

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



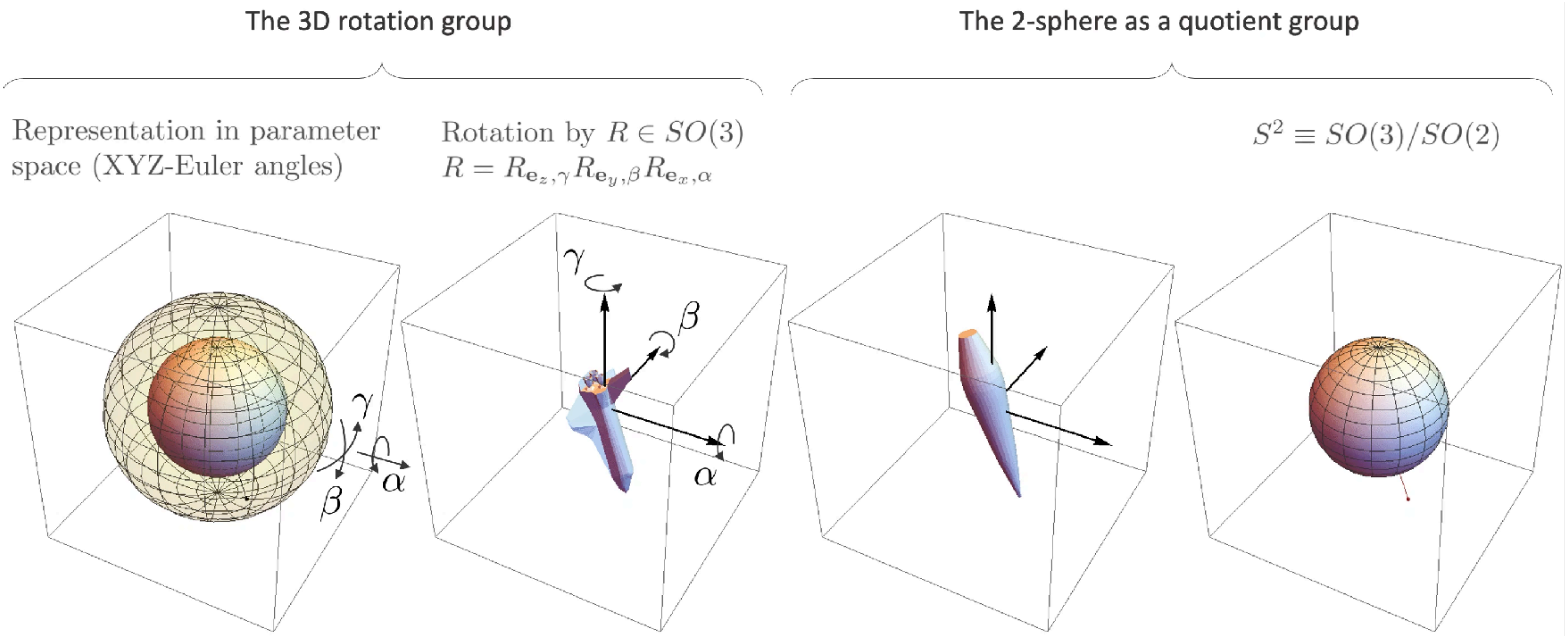
Group theory: Homogeneous spaces

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



Group theory: Homogeneous spaces

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



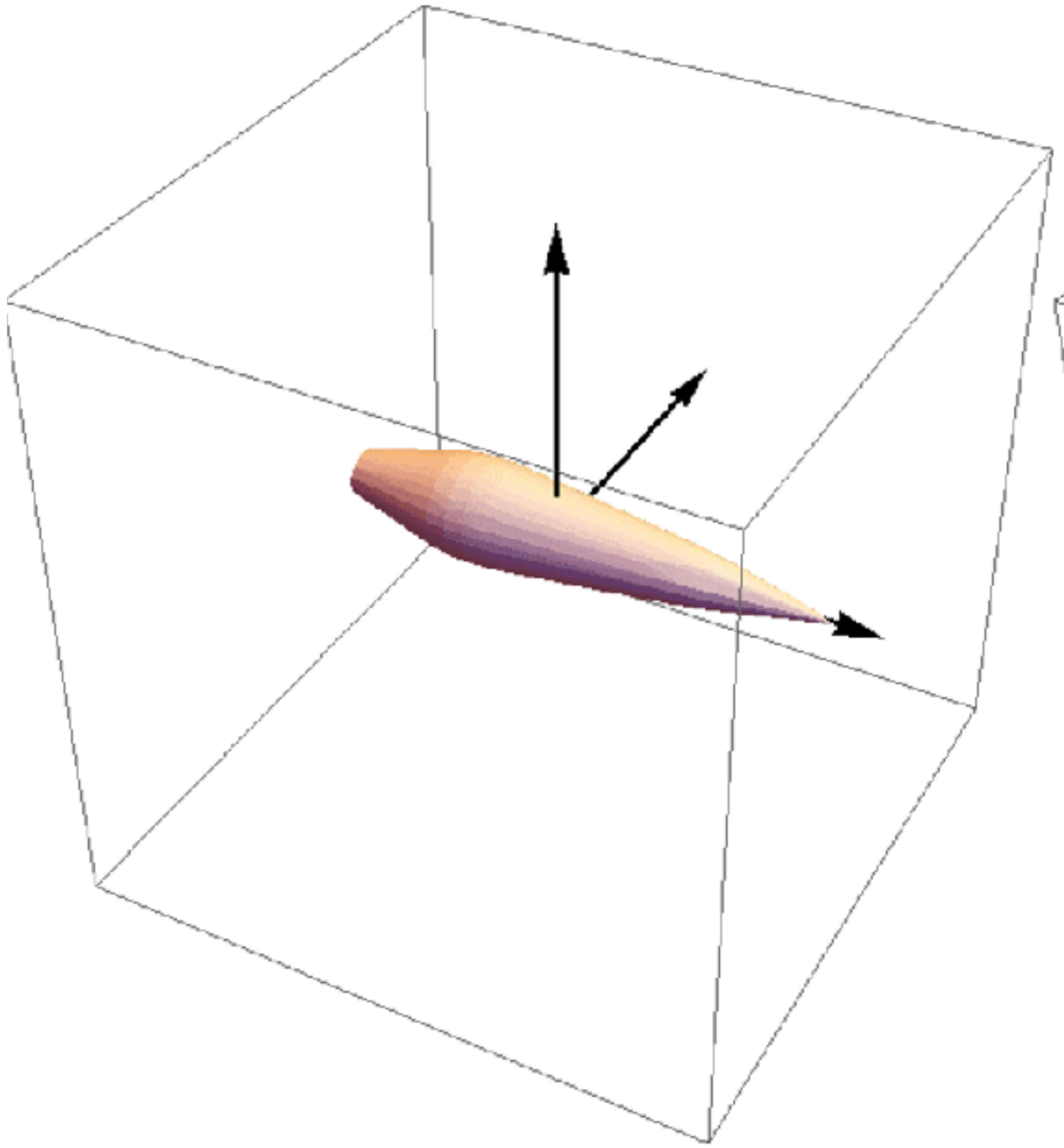
Group theory: Homogeneous space $\equiv$ Quotient space

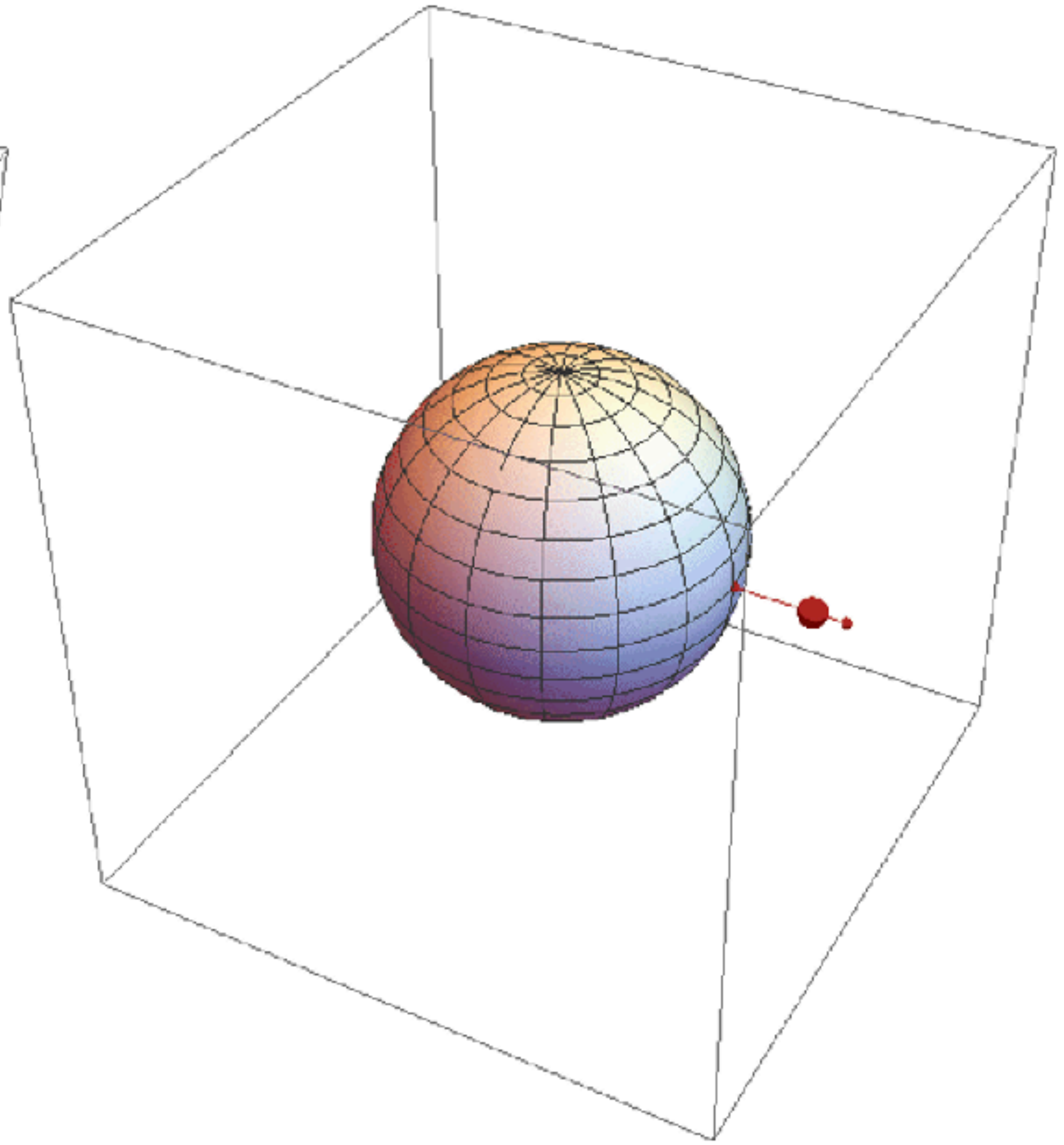
Any quotient space is a homogeneous space

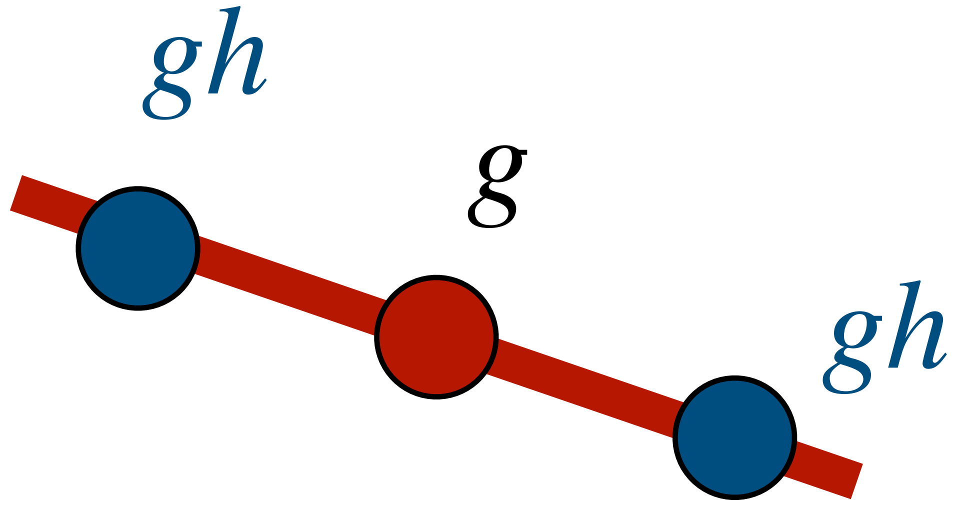
Any homogeneous space is a quotient space

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.



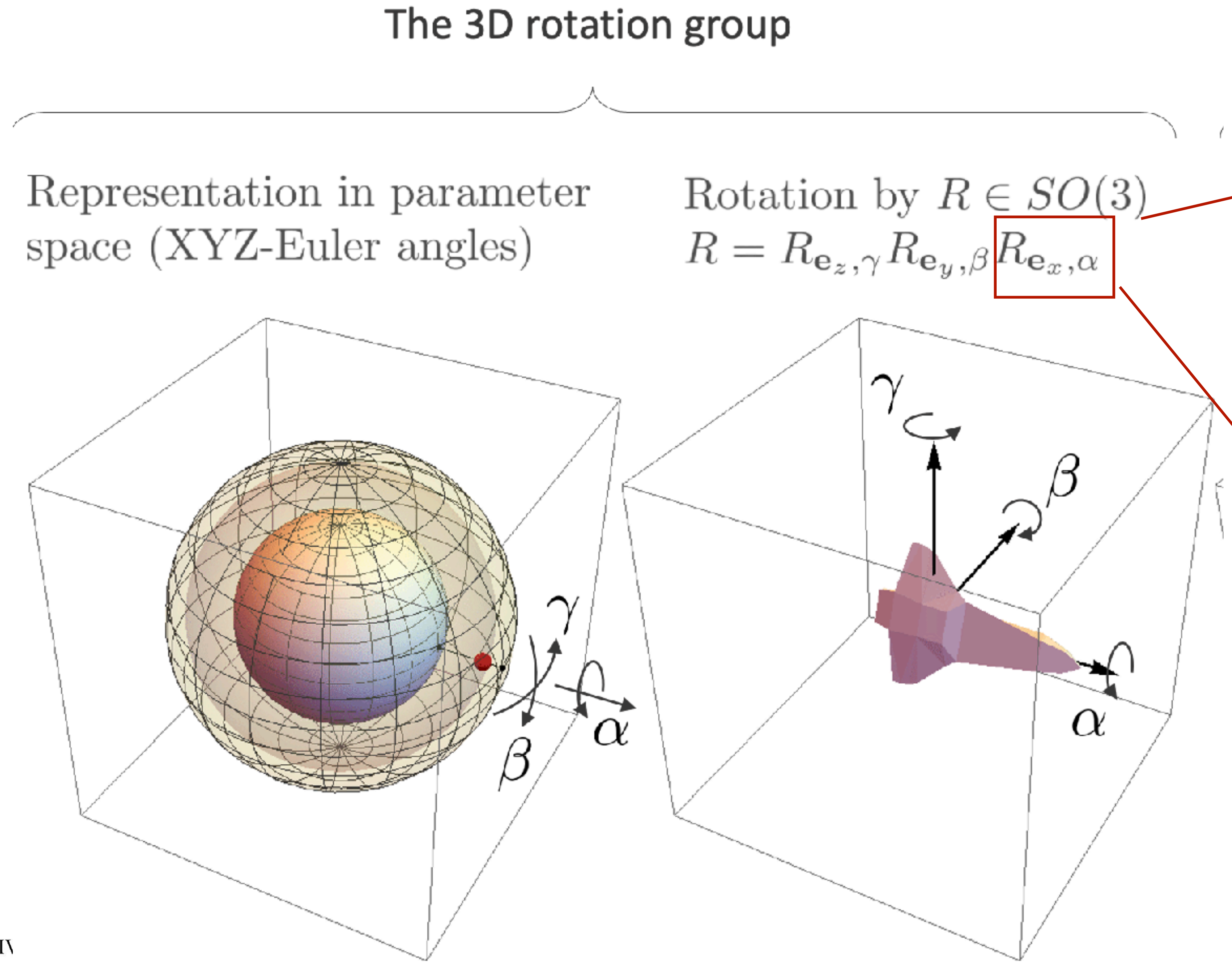




$$gH = \{gh \,|\, h \in H\}$$

Group theory: Stabilizer

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



So the sphere is a quotient space

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

with

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

Theorem 3.2:

Let $\mathcal{K} : \mathbb{L}_2(X) \rightarrow \mathbb{L}_2(Y)$ 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 $\mathcal{K}$ is equivariant to group G if and only if:

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

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.

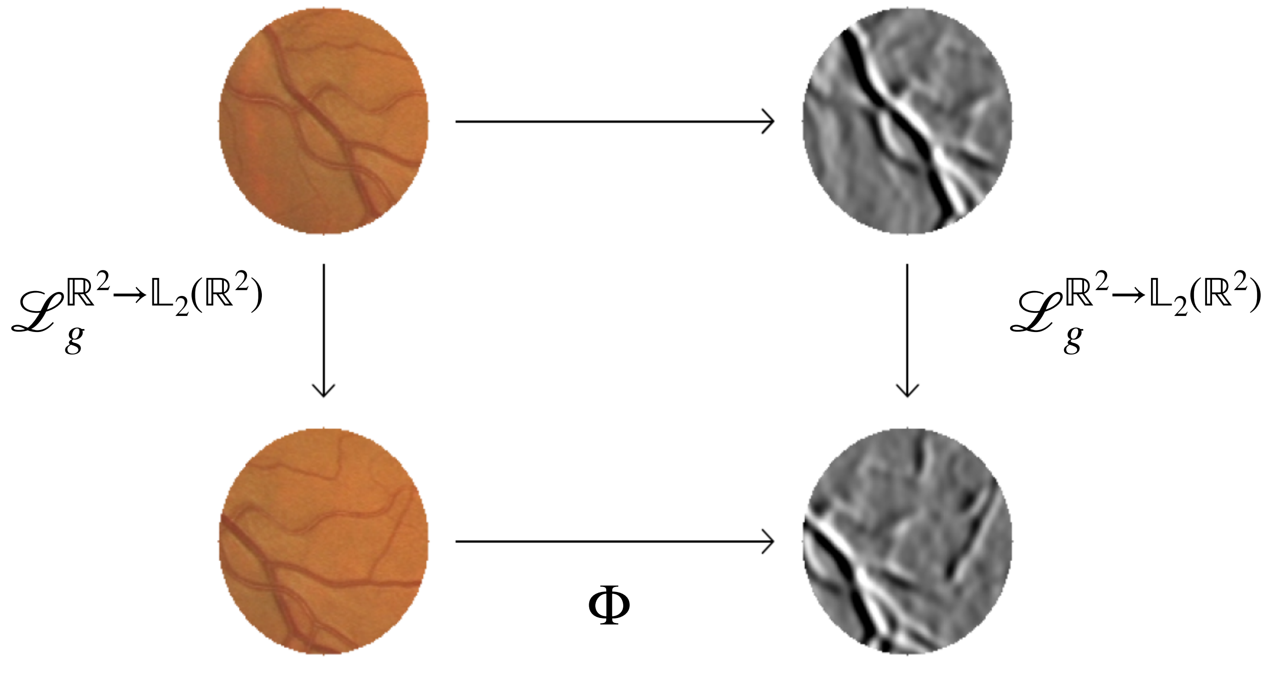
$(X = G, Y = G/H)$

Projection layer. Mean pooling over H.

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

Global pooling over G.

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



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

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

SE(2) equivariance iff

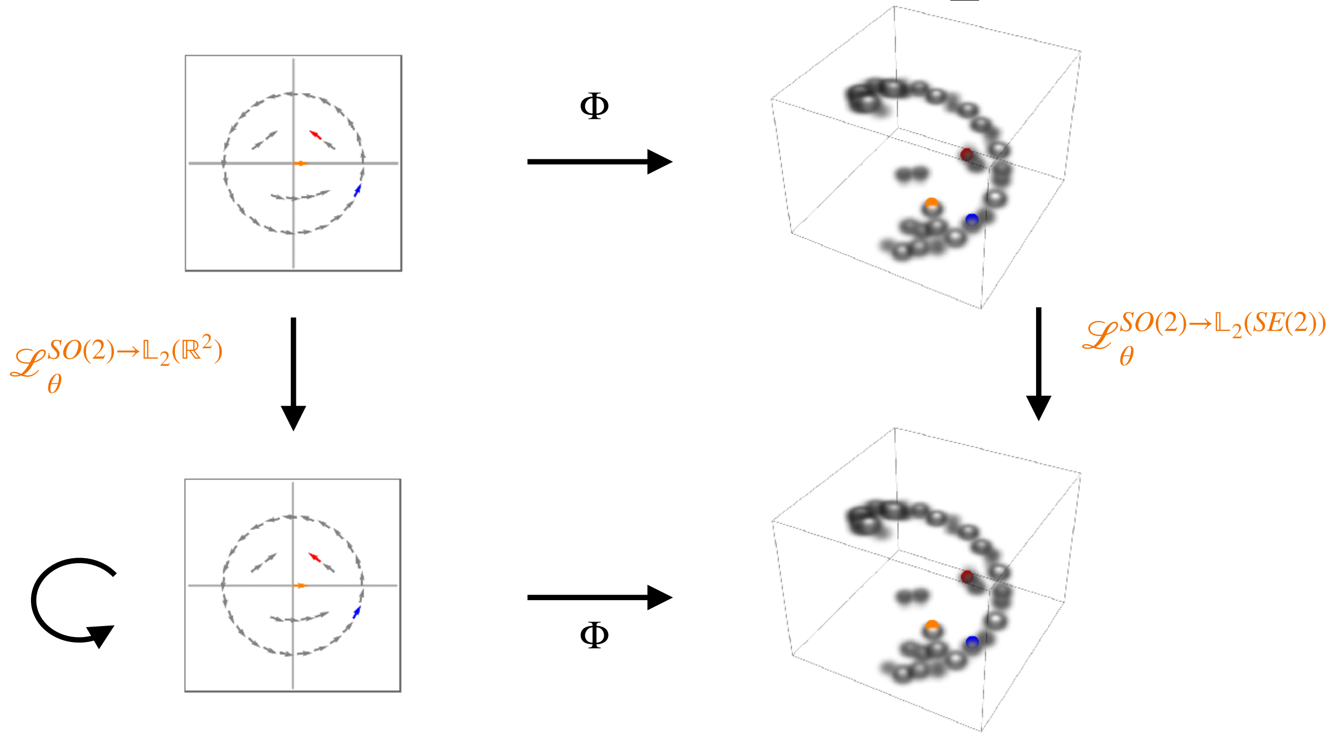
$$(\mathscr{L}_\theta^{SO(2) \to \mathbb{L}_2(\mathbb{R}^2)} k)(\mathbf{x}) = k(\mathbf{x})$$

$$\Leftrightarrow$$

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

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

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

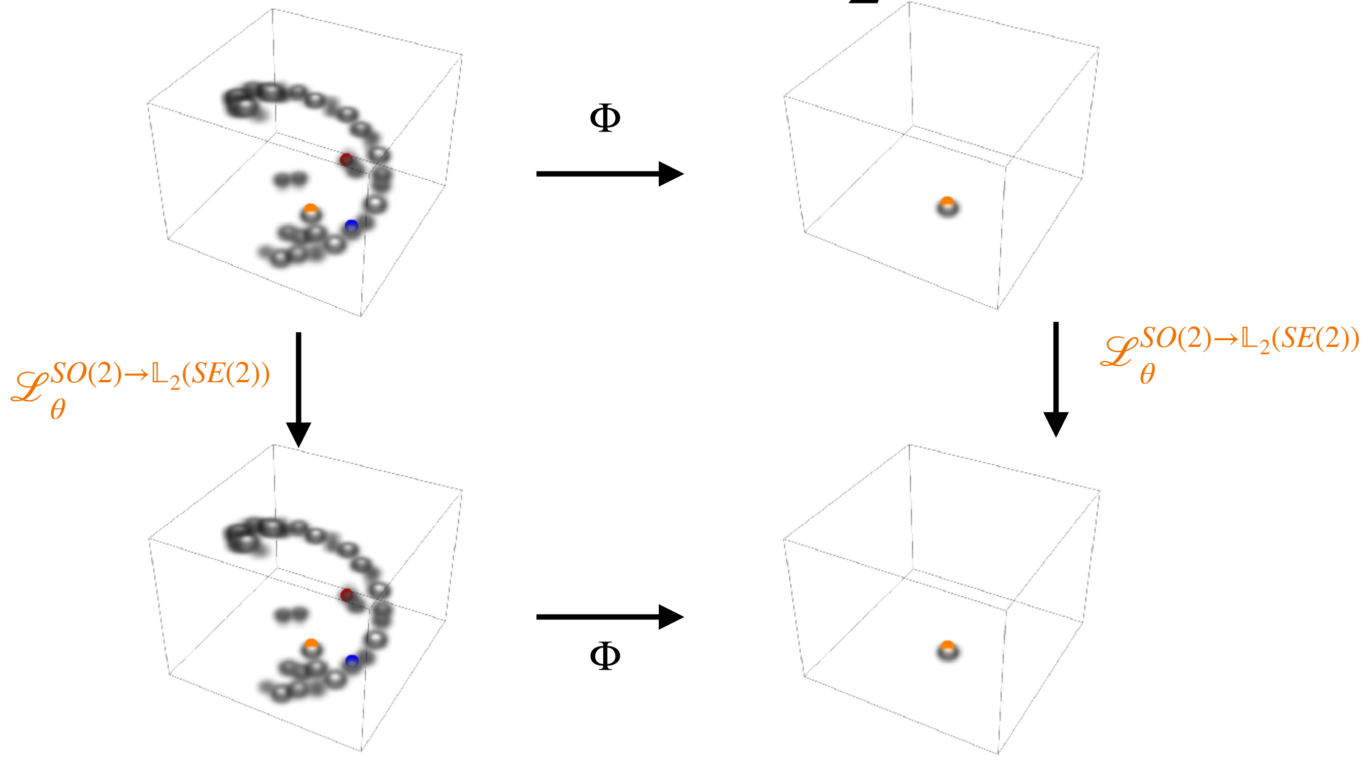


$$(k \tilde{\star} f)(\mathbf{x}) = (\mathscr{L}_g^{SE(2) \to \mathbb{L}_2(\mathbb{R}^2)} k , f)_{\mathbb{L}_2(\mathbb{R}^2)}$$

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

No constraints

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



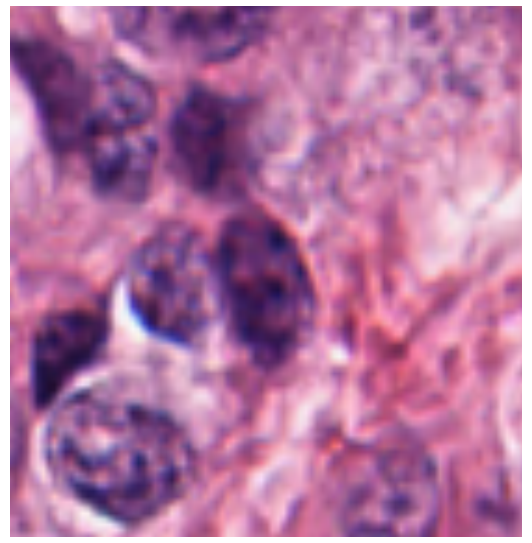
$$(k \tilde{\star} f)(\mathbf{x}) = (\mathscr{L}_g^{SE(2) \to \mathbb{L}_2(SE(2))} k , f)_{\mathbb{L}_2(SE(2)}$$

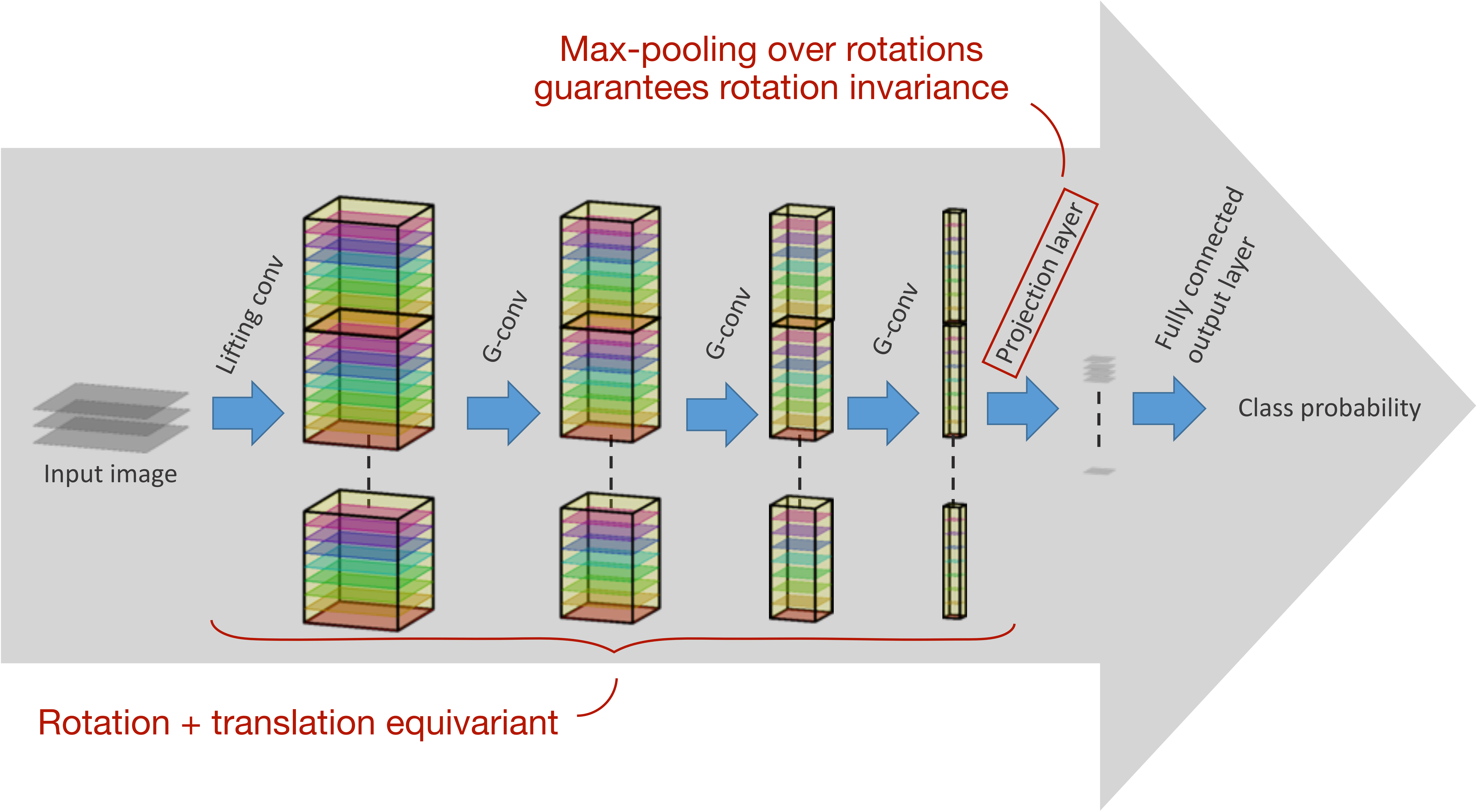
$$= \int_{\mathbb{R}^2} \int_{S^1} k(\mathbf{R}_\theta^{-1}(\mathbf{x}' - \mathbf{x}), \theta' - \theta \mod 2\pi) f(\mathbf{x}', \theta') d\mathbf{x}'$$

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

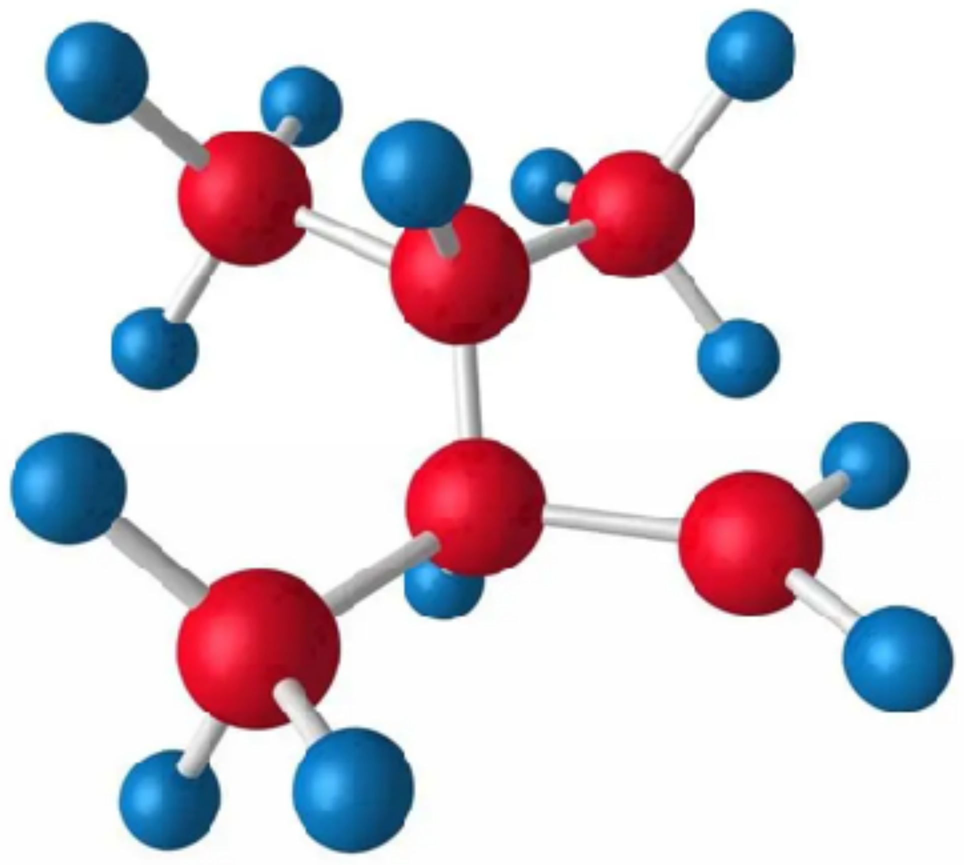
* Motivation
* Introduction to group theory
* Regular group convolutional neural networks
* Applications

Part II: Group convolutions are all you need

* Theorem: Equivariant linear operators are group convolutions
* Characterization of types of group equivariant layers

Part III: Steerable group convolutions for molecular data and the N-body problem

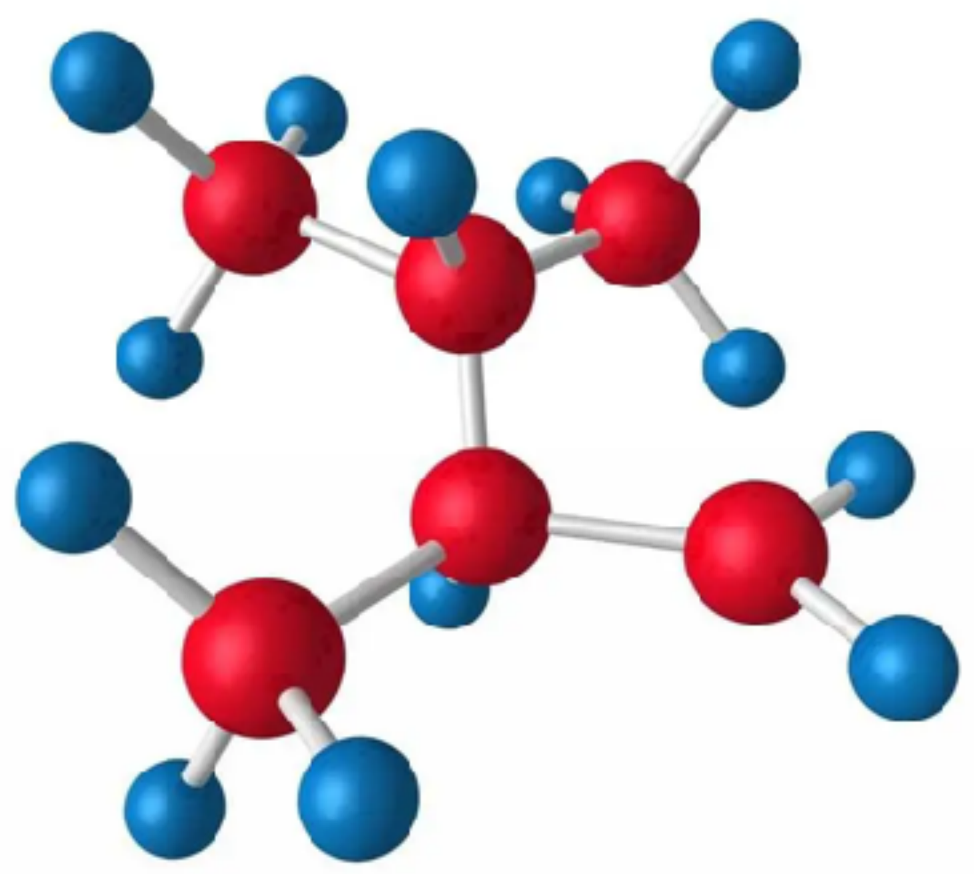
* Graph Neural networks
* Irreducible representations, steerable operators and vector spaces
* Steerable Graph NNs (Point Convolutions)

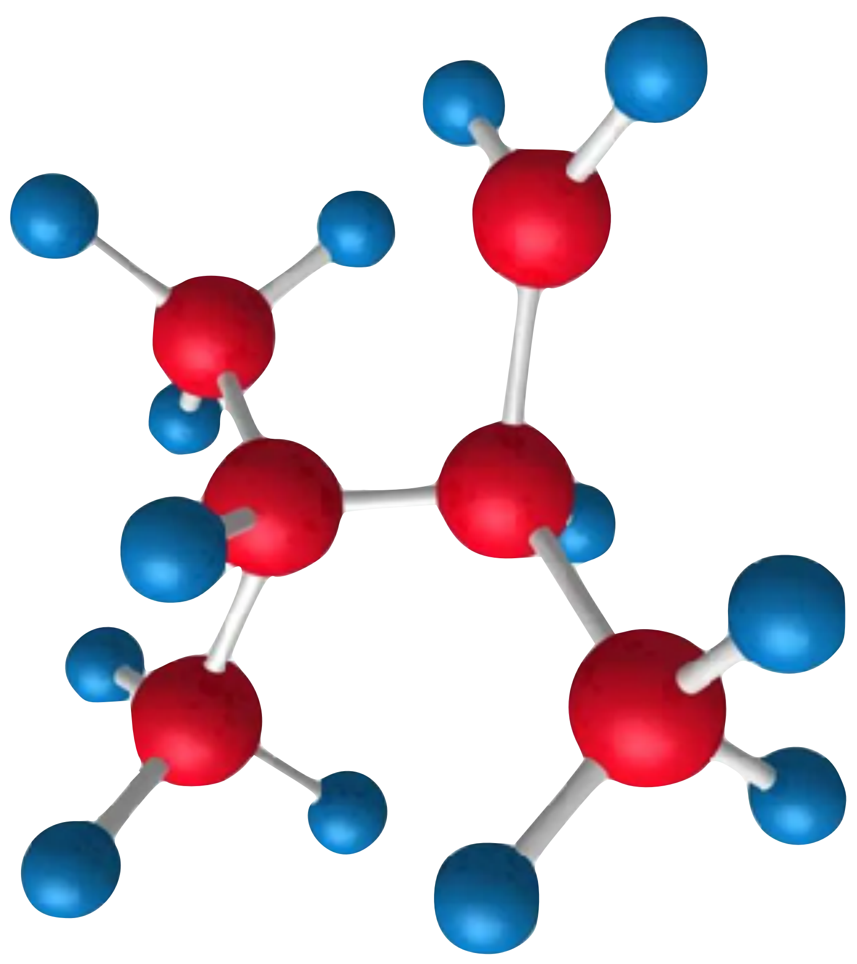


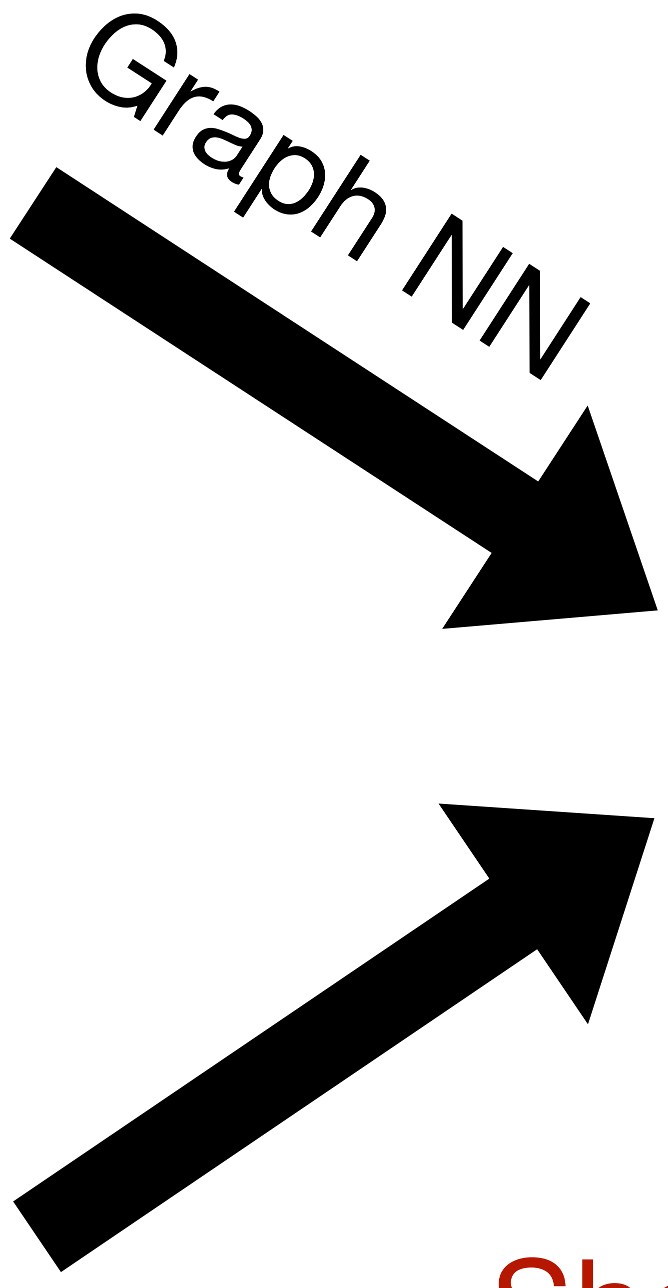
Graph NN

Property

Image from https://www.qps.com/2020/04/01/selecting-a-cro-for-chronic-disease-drug-development-using-small-molecules/





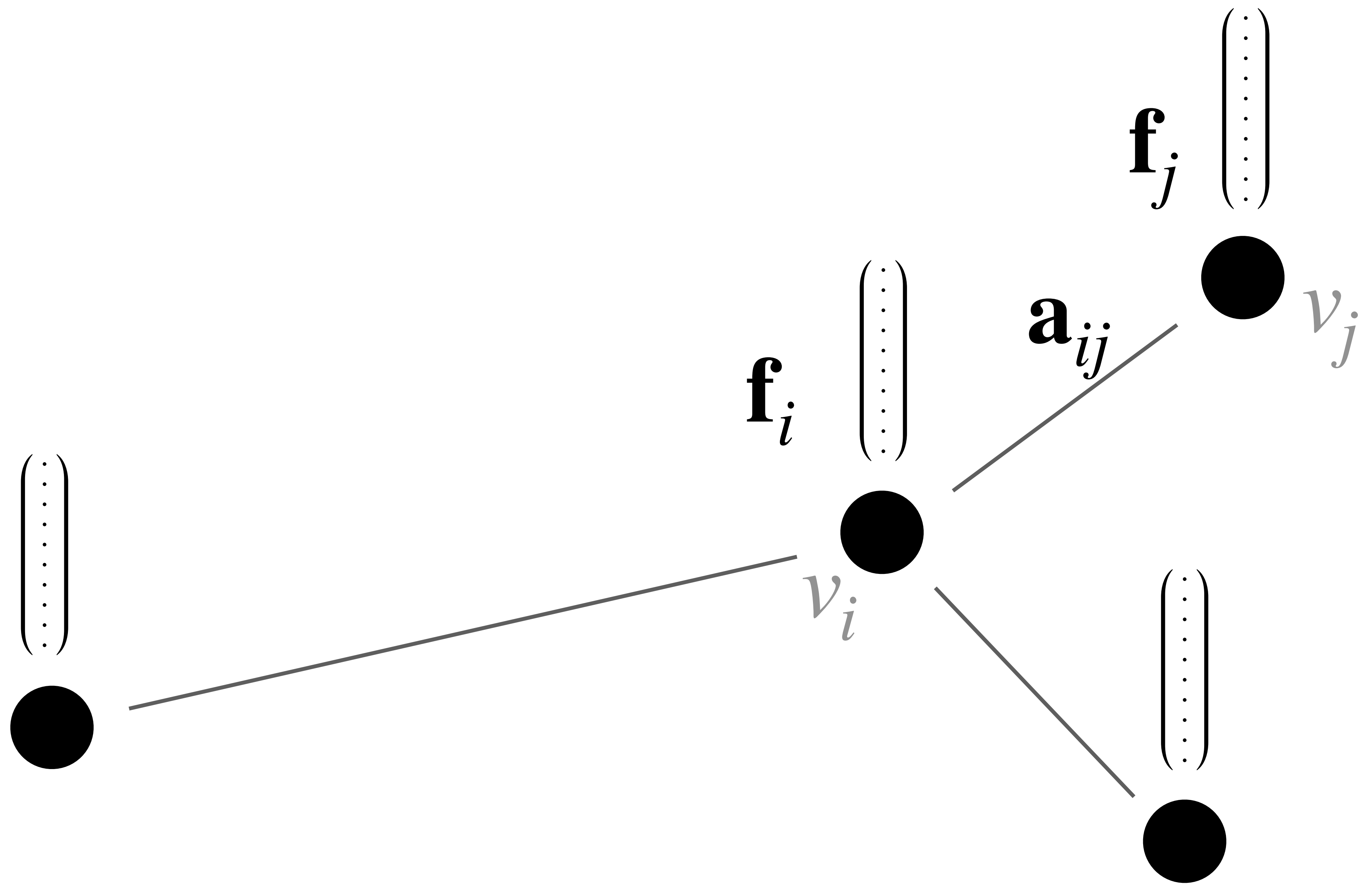


Property

Should be invariant!

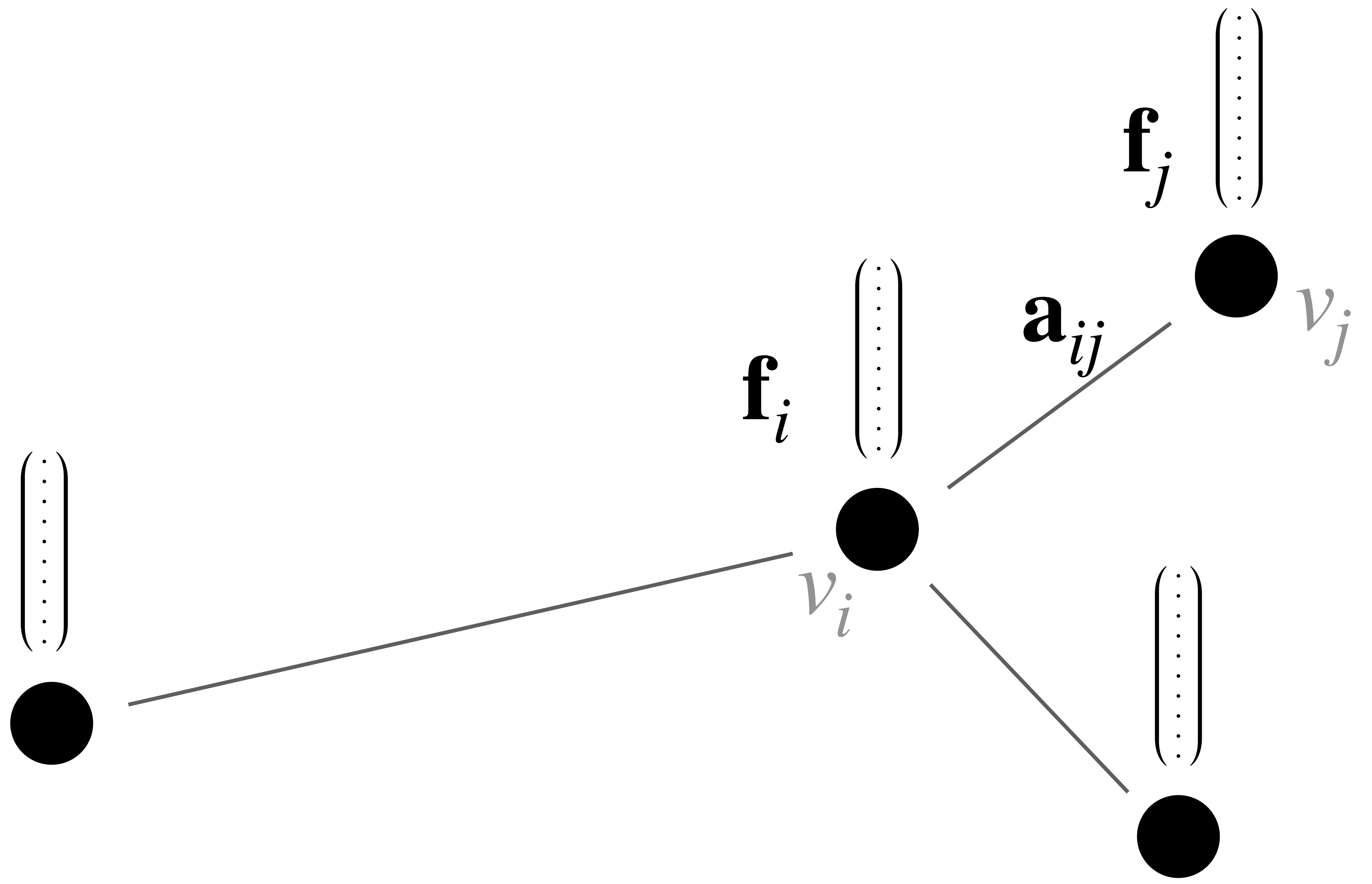
Graph Neural Networks: Message Passing

Consider graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with at each node v_i a feature vector $\mathbf{f}_j$ and possibly on each edge an edge attribute $\mathbf{a}_{ij}$.



Graph Neural Networks: Message Passing

Consider graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, with at each node v_i a feature vector $\mathbf{f}_j$ and possibly on each edge an edge attribute $\mathbf{a}_{ij}$.



Goal: iteratively update node features via message passing (Gilmer et al. 2017)

Messages

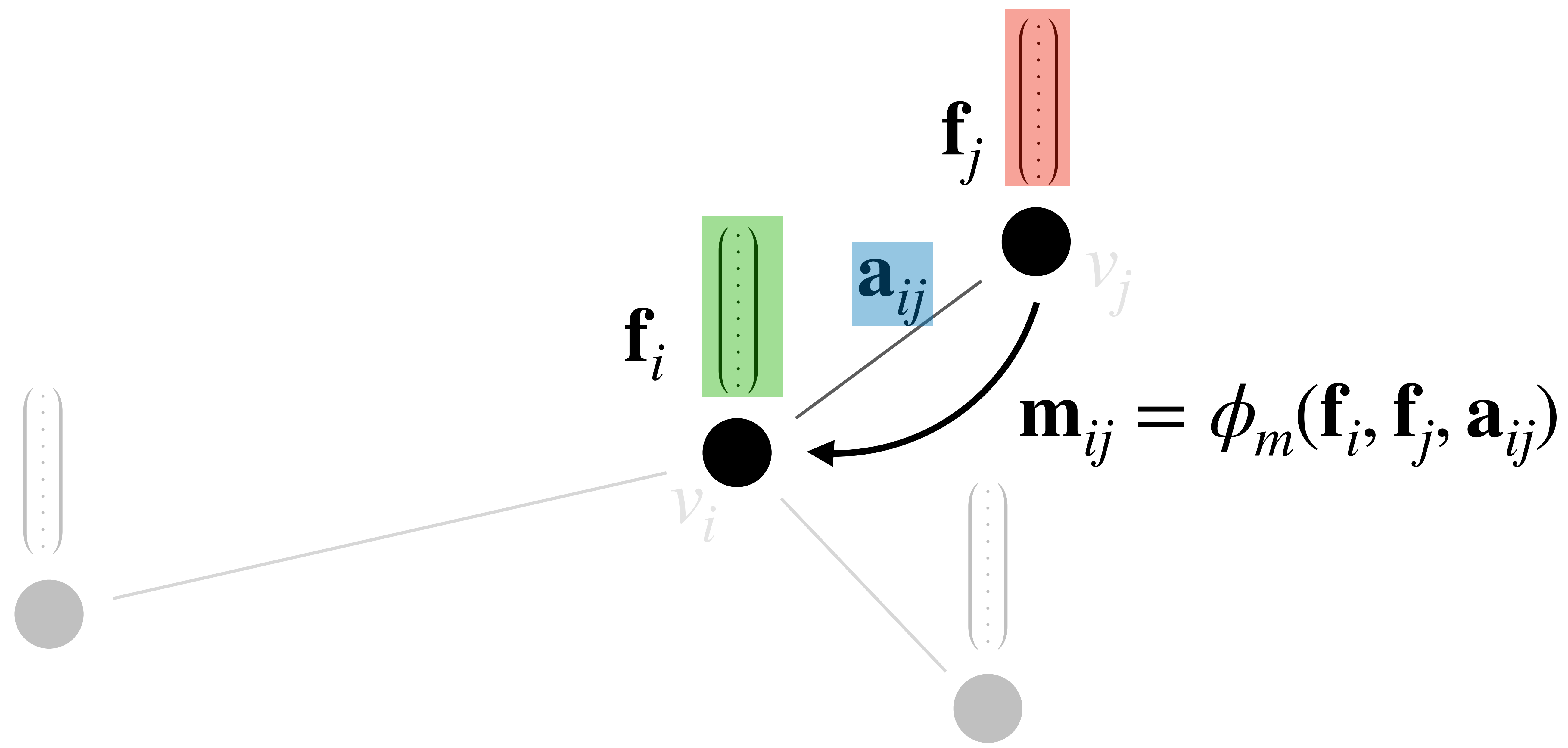
$$\mathbf{m}_{ij} = \phi_m(\mathbf{f}_i, \mathbf{f}_j, \mathbf{a}_{ij})$$

Aggregate + node updates

$$\mathbf{f}'_i = \phi_f\left(\mathbf{f}_i, \sum_{j \in \mathscr{N}(i)} \mathbf{m}_{ij}\right)$$

Graph Neural Networks: Message Passing

Consider graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with at each node v_i a feature vector $\mathbf{f}_j$ and possibly on each edge an edge attribute $\mathbf{a}_{ij}$.



Goal: iteratively update node features via message passing (Gilmer et al. 2017)

Messages

$$\mathbf{m}_{ij} = \phi_m(\mathbf{f}_i, \mathbf{f}_j, \mathbf{a}_{ij})$$

Aggregate + node updates

$$\mathbf{f}_i' = \phi_f\left(\mathbf{f}_i, \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}\right)$$

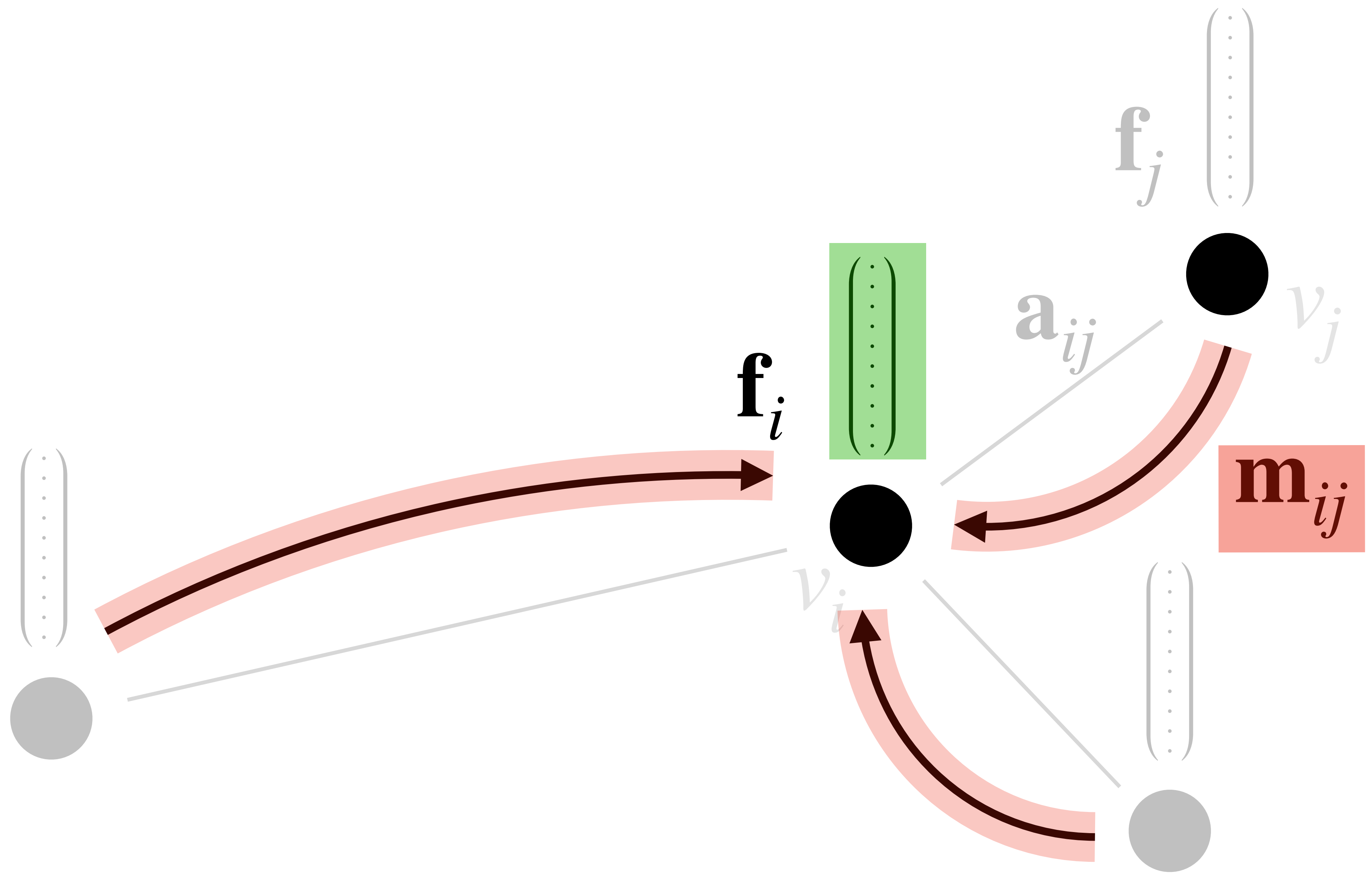
Graph Neural Networks: Message Passing

Consider graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with at each node v_i a feature vector $\mathbf{f}_j$ and possibly on each edge an edge attribute $\mathbf{a}_{ij}$.

Goal: iteratively update node features via message passing (Gilmer et al. 2017)

Messages

$$\mathbf{m}_{ij} = \phi_m(\mathbf{f}_i, \mathbf{f}_j, \mathbf{a}_{ij})$$

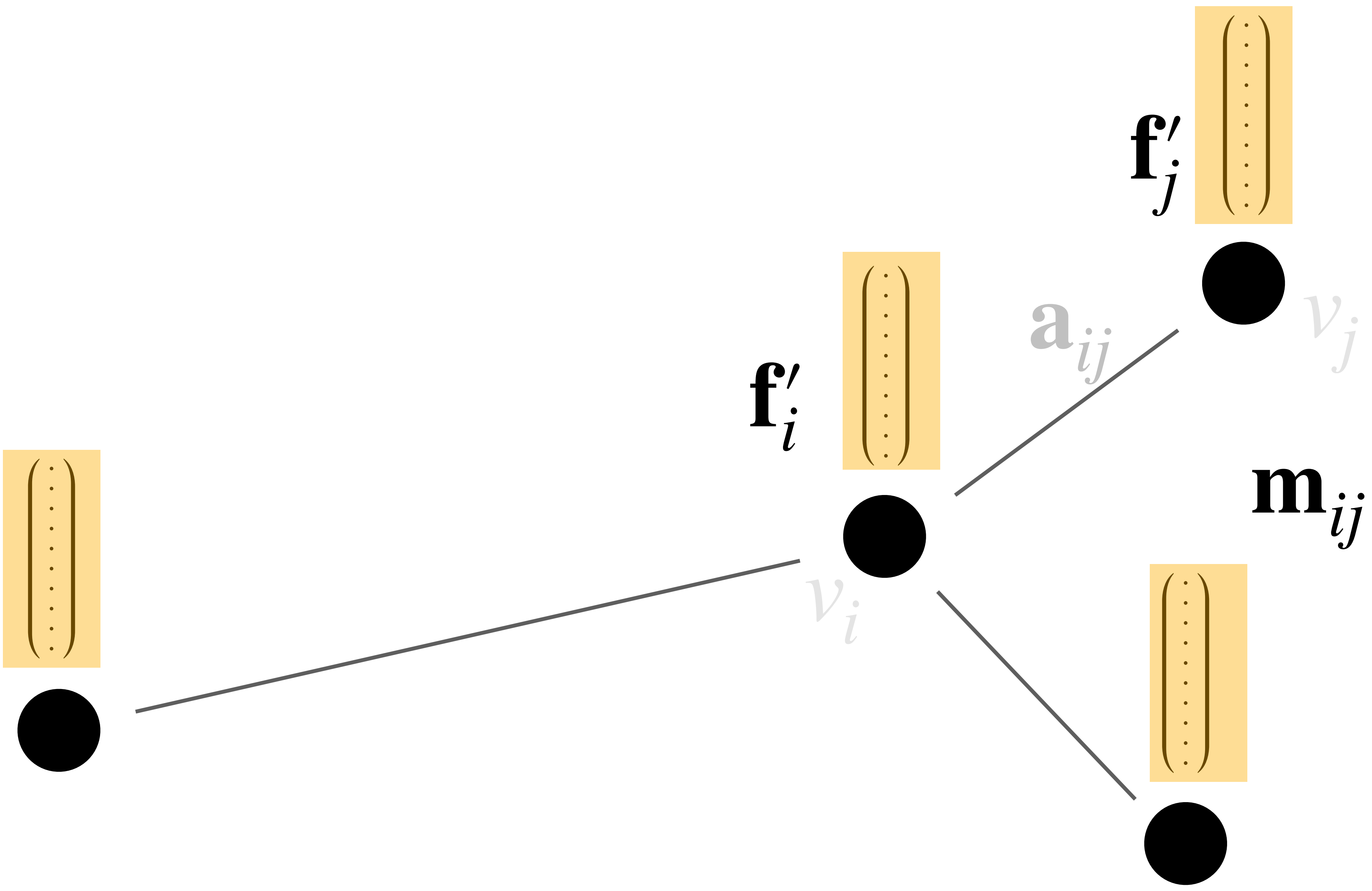


Aggregate + node updates

$$\mathbf{f}_i' = \phi_f\left(\mathbf{f}_i, \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}\right)$$

Graph Neural Networks: Message Passing

Consider graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with at each node v_i a feature vector $\mathbf{f}_j$ and possibly on each edge an edge attribute $\mathbf{a}_{ij}$.



Goal: iteratively update node features via message passing (Gilmer et al. 2017)

Messages

$$\mathbf{m}_{ij} = \phi_m(\mathbf{f}_i, \mathbf{f}_j, \mathbf{a}_{ij})$$

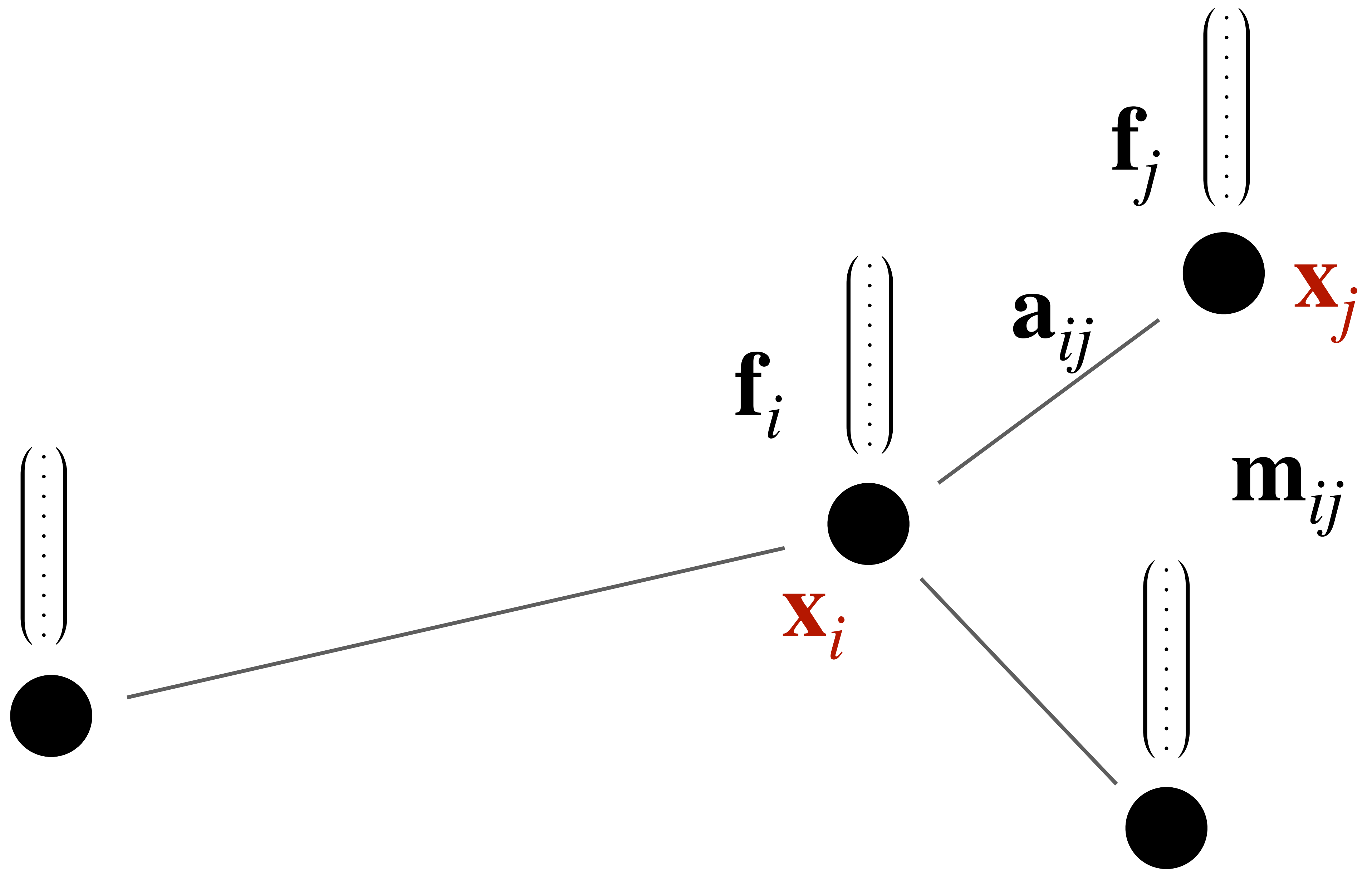
Aggregate + node updates

$$\mathbf{f}'_i = \phi_f\left(\mathbf{f}_i, \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}\right)$$

Graph Neural Networks: Message Passing

Consider graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with at each node v_i a feature vector $\mathbf{f}_j$ and possibly on each edge an edge attribute $\mathbf{a}_{ij}$.

Now with node positions $\mathbf{x}_i$ (sparse feature map/point cloud)



Special case: Point convolutions

Messages (linear transformations based on kernel)

$$\mathbf{m}_{ij} = \phi_m(\mathbf{f}_i, \mathbf{f}_j, \mathbf{x}_j - \mathbf{x}_i)$$

$$= k(\mathbf{x}_j - \mathbf{x}_i)\mathbf{f}_j \qquad (k : \mathbb{R}^d \rightarrow \mathbb{R}^{N_{out} \times N_{in}})$$

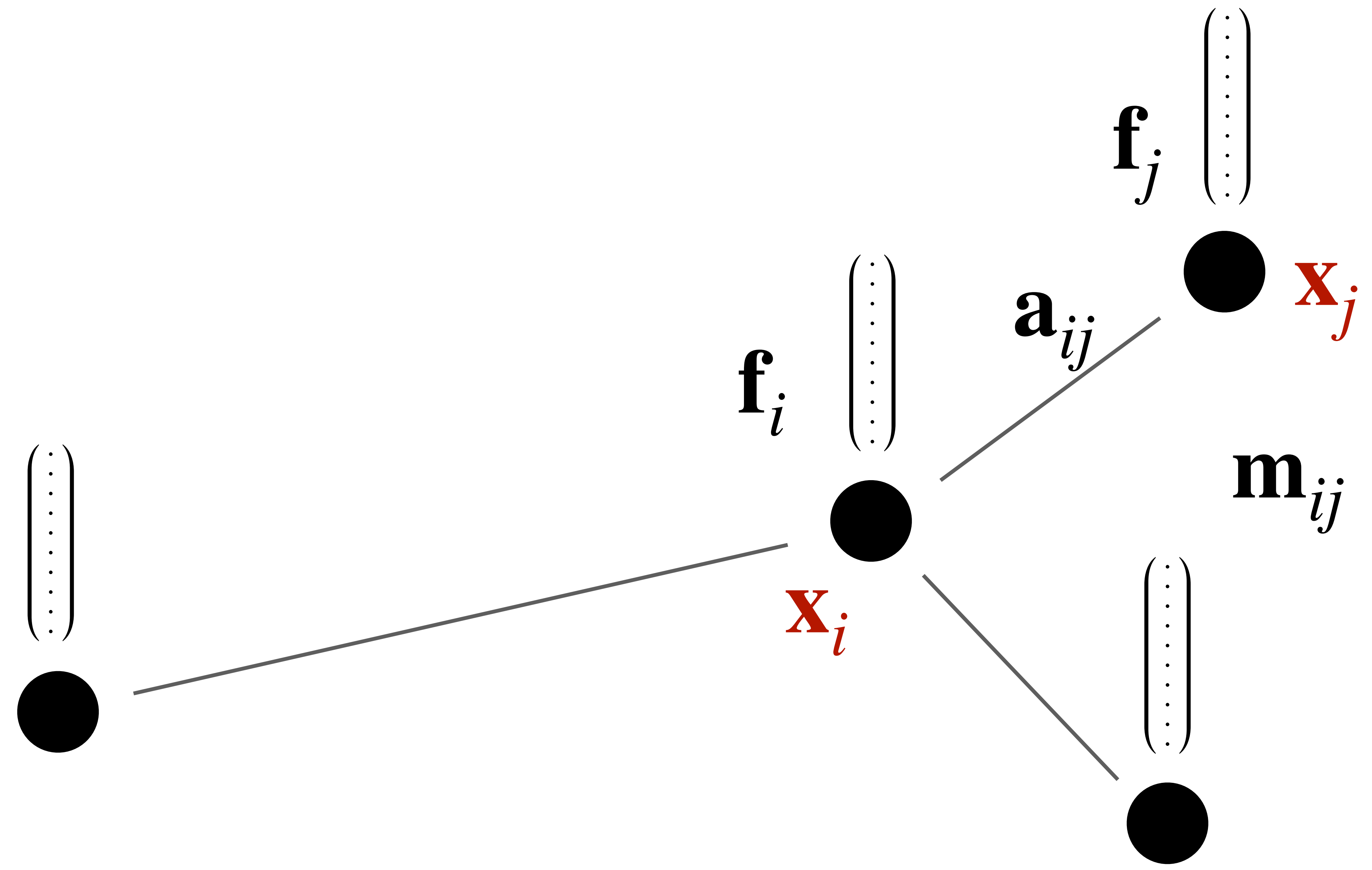
Aggregate + node updates (convolution + activation fn)

$$\mathbf{f}_i' = \phi_f(\underbrace{\sum_{j \in \mathcal{N}(i)} k(\mathbf{x}_j - \mathbf{x}_i)\mathbf{f}_j}_{(k \star f)(\mathbf{x}_i)})$$

Graph Neural Networks: Message Passing

Consider graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, with at each node v_i a feature vector $\mathbf{f}_j$ and possibly on each edge an edge attribute $\mathbf{a}_{ij}$.

Now with node positions $\mathbf{x}_i$ (sparse feature map/point cloud)



Special case: Point convolutions

Messages (linear transformations based on kernel)

$$\mathbf{m}_{ij} = \phi_m(\mathbf{f}_i, \mathbf{f}_j, \mathbf{x}_j - \mathbf{x}_i)$$

$$= k(\mathbf{x}_j - \mathbf{x}_i)\mathbf{f}_j \qquad (k : \mathbb{R}^d \rightarrow \mathbb{R}^{N_{out} \times N_{in}})$$

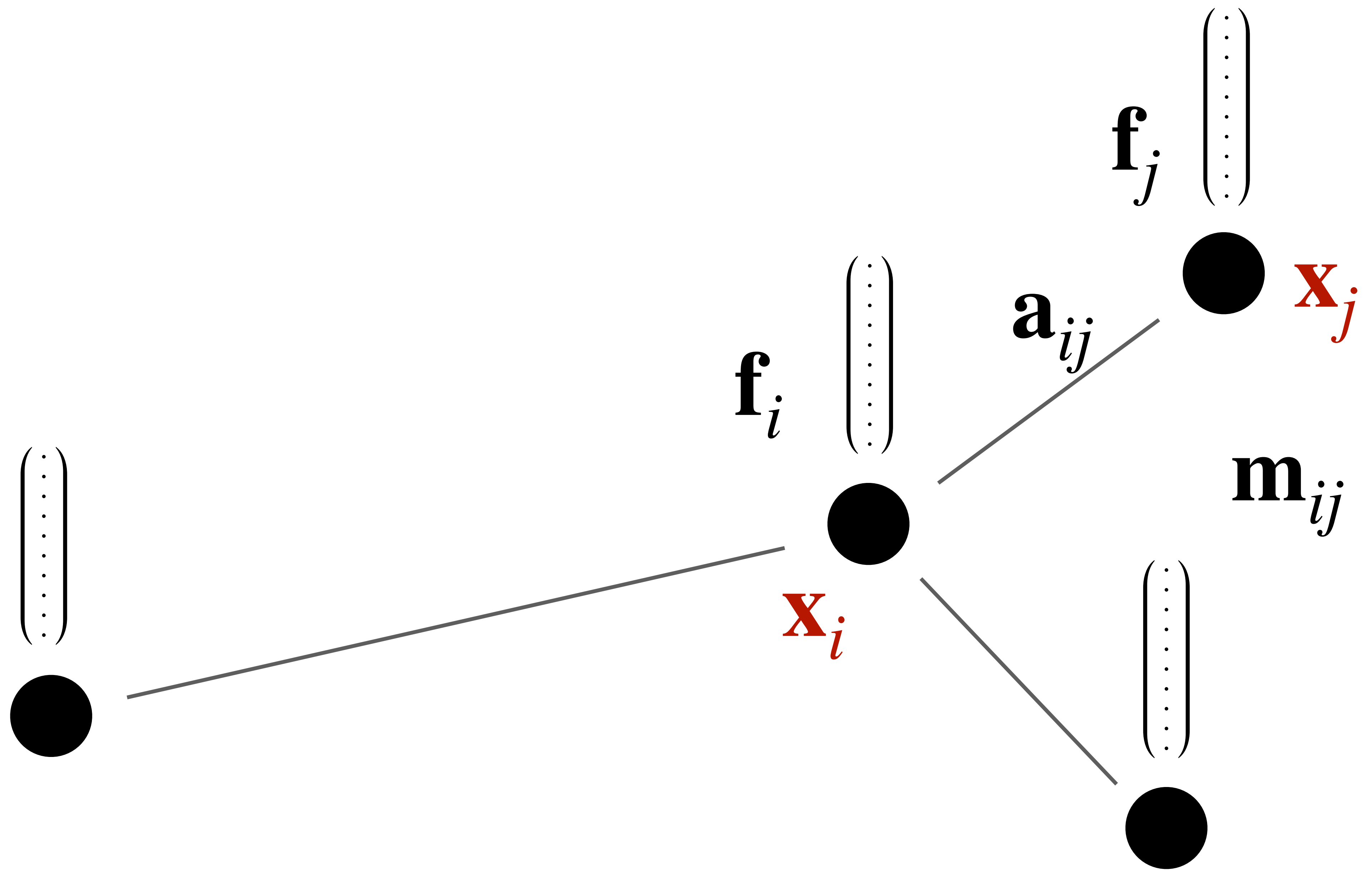
Aggregate + node updates (convolution + activation fn)

$$\mathbf{f}_i' = \phi_f(\underbrace{\sum_{j \in \mathcal{N}(i)} k(\mathbf{x}_j - \mathbf{x}_i)\mathbf{f}_j}_{(k \star f)(\mathbf{x}_i)})$$

Graph Neural Networks: Message Passing

Consider graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with at each node v_i a feature vector $\mathbf{f}_j$ and possibly on each edge an edge attribute $\mathbf{a}_{ij}$.

Now with node positions $\mathbf{x}_i$ (sparse feature map/point cloud)



Special case: Point convolutions

Messages (linear transformations based on kernel)

$$\mathbf{m}_{ij} = \phi_m(\mathbf{f}_i, \mathbf{f}_j, \mathbf{x}_j - \mathbf{x}_i)$$
$$= k(\mathbf{x}_j - \mathbf{x}_i)\mathbf{f}_j \qquad (k : \mathbb{R}^d \to \mathbb{R}^{N_{out} \times N_{in}})$$

Aggregate + node updates (convolution + activation fn)

$$\mathbf{f}'_i = \phi_f\Big(\sum_{j \in \mathcal{N}(i)} k(\mathbf{x}_j - \mathbf{x}_i)\mathbf{f}_j \Big)$$

Only SE(3) equivariant when k is isotropic

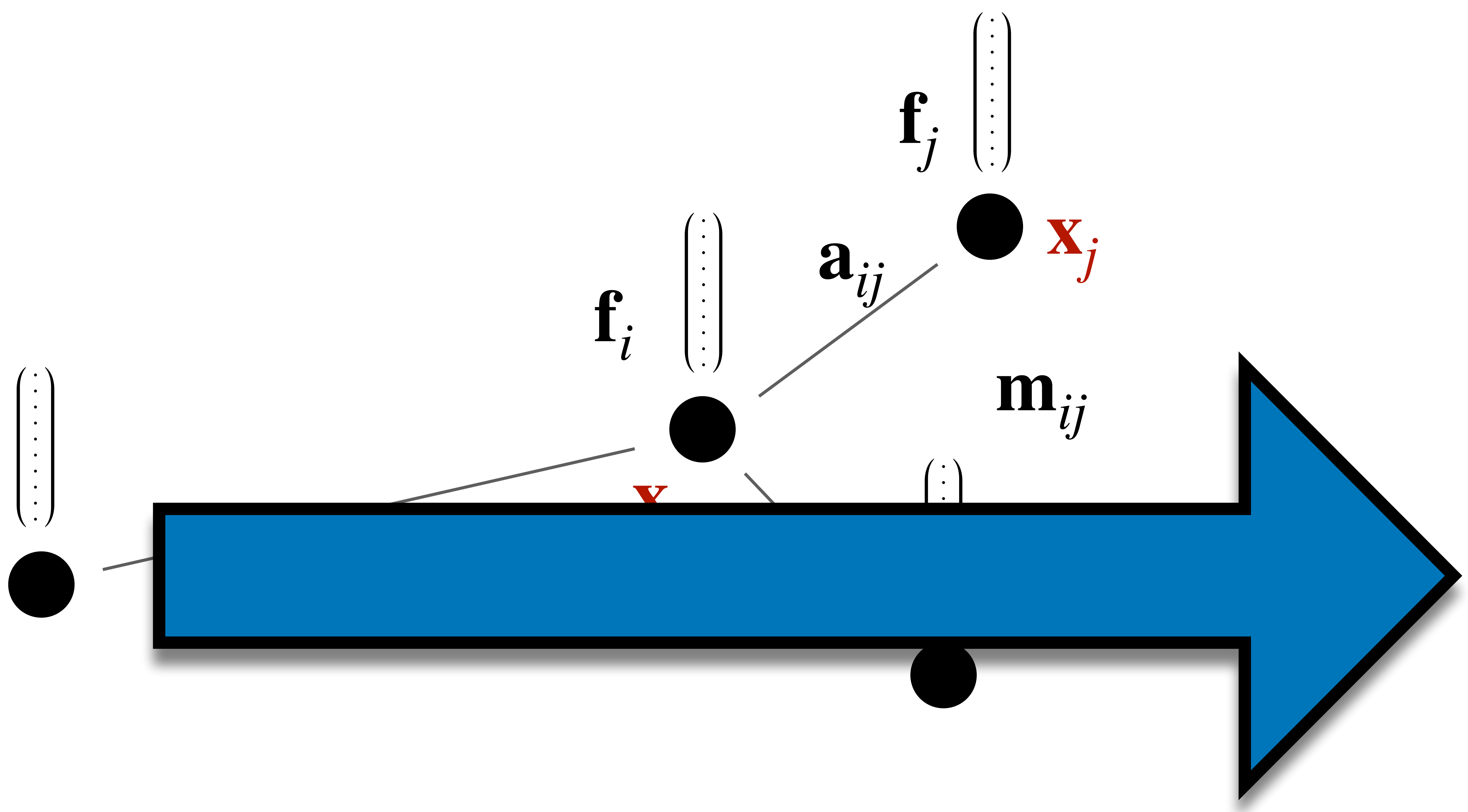
$$\mathbf{f}'_i = \phi_f\Big(\sum_{j \in \mathcal{N}(i)} k(\|\mathbf{x}_j - \mathbf{x}_i\|)\mathbf{f}_j \Big)$$

or…

Graph Neural Networks: Message Passing

Consider graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with at each node v_i a feature vector $\mathbf{f}_j$ and possibly on each edge an edge attribute $\mathbf{a}_{ij}$.

Now with node positions $\mathbf{x}_i$ (sparse feature map/point cloud)



Special case: Point convolutions

Messages (linear transformations based on kernel)

$$\mathbf{m}_{ij} = \phi_m(\mathbf{f}_i, \mathbf{f}_j, \mathbf{x}_j - \mathbf{x}_i)$$
$$= k(\mathbf{x}_j - \mathbf{x}_i)\mathbf{f}_j \qquad (k : \mathbb{R}^d \to \mathbb{R}^{N_{out} \times N_{in}})$$

Aggregate + node updates (convolution + activation fn)

$$\mathbf{f}_i' = \phi_f\Big(\sum_{j \in \mathcal{N}(i)} k(\mathbf{x}_j - \mathbf{x}_i)\mathbf{f}_j \Big)$$

Use group convolutions!

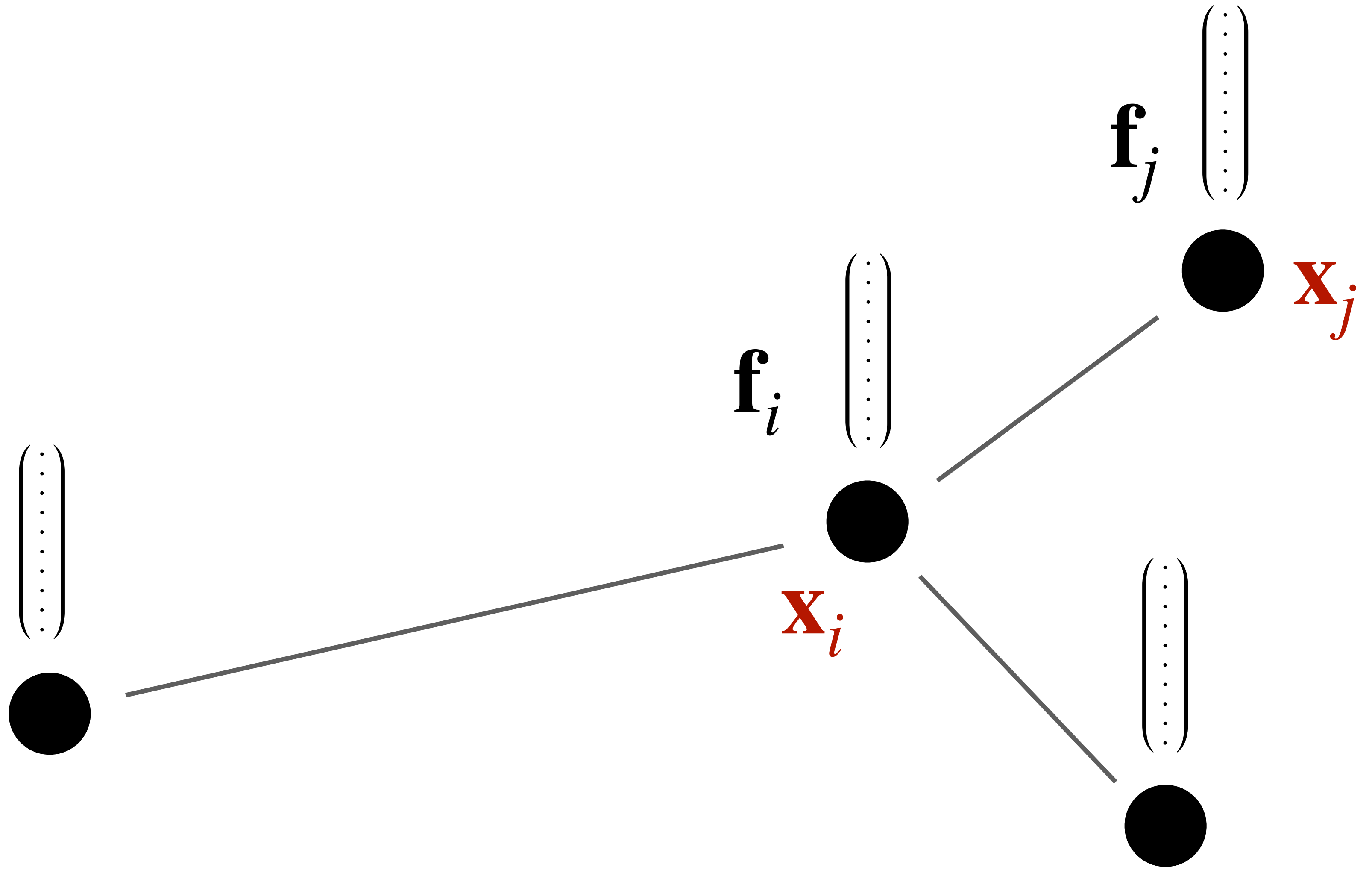
$$\mathbf{f}_i'(\mathbf{R}) = \phi_f\Big(\sum_{j \in \mathcal{N}(i)} k(\mathbf{R}^{-1}(\mathbf{x}_j - \mathbf{x}_i))\mathbf{f}_j \Big)$$

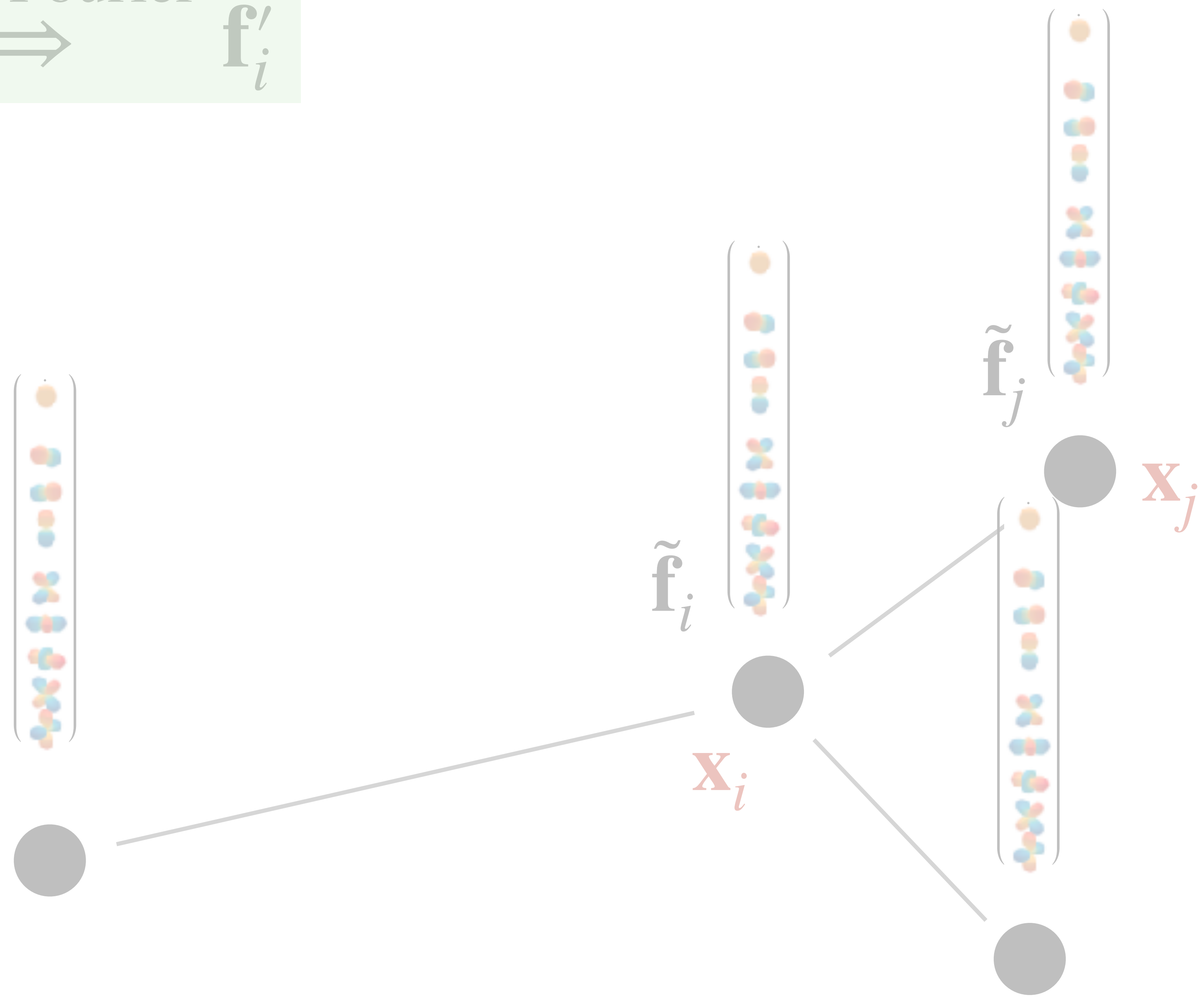
Steerable methods for computational chemistry

Instead of feature vectors $\mathbf{f}_i \in \mathbb{R}^{N_c}$

Work with steerable vectors $\tilde{\mathbf{f}}_i \in V_L^{N_c}$, whose sub-vectors represent functions on $SO(3)$ via a Fourier transform

$$\mathbf{f}'_i(\mathbf{R}) \overset{SO(3)\text{−Fourier}}{\Longleftrightarrow} \tilde{\mathbf{f}}'_i$$





Point cloud sparsely represents feature map
$\mathbf{f} : \mathbb{R}^3 \to \mathbb{R}^{Nc}$

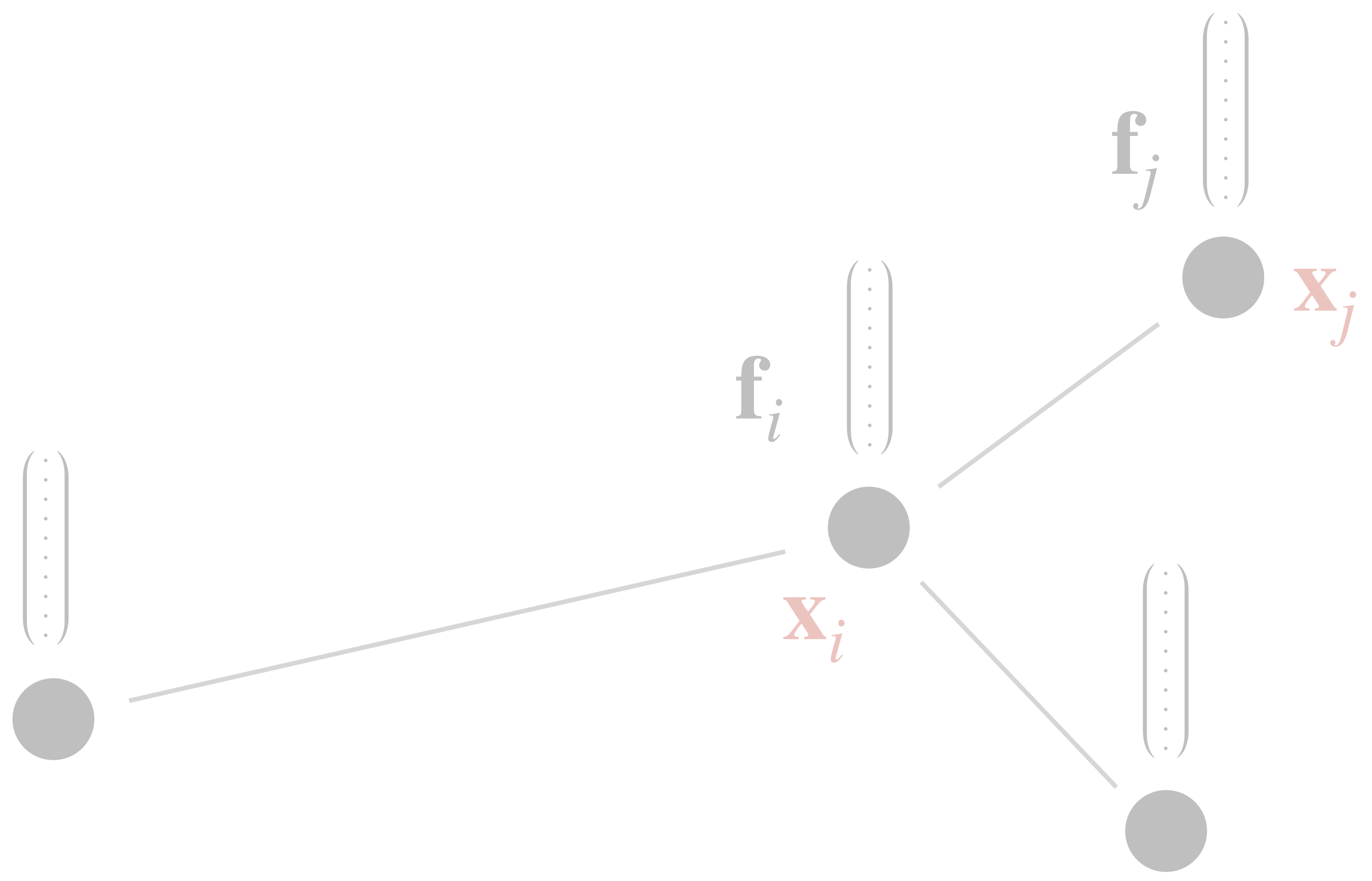
Point cloud sparsely represents steerable vector field
$\tilde{\mathbf{f}} : \mathbb{R}^3 \to V_L^{Nc}$
or a regular feature map on $\mathbb{R}^3 \times SO(3)$
$\mathbf{f} : \mathbb{R}^3 \times SO(3) \to \mathbb{R}^{Nc}$

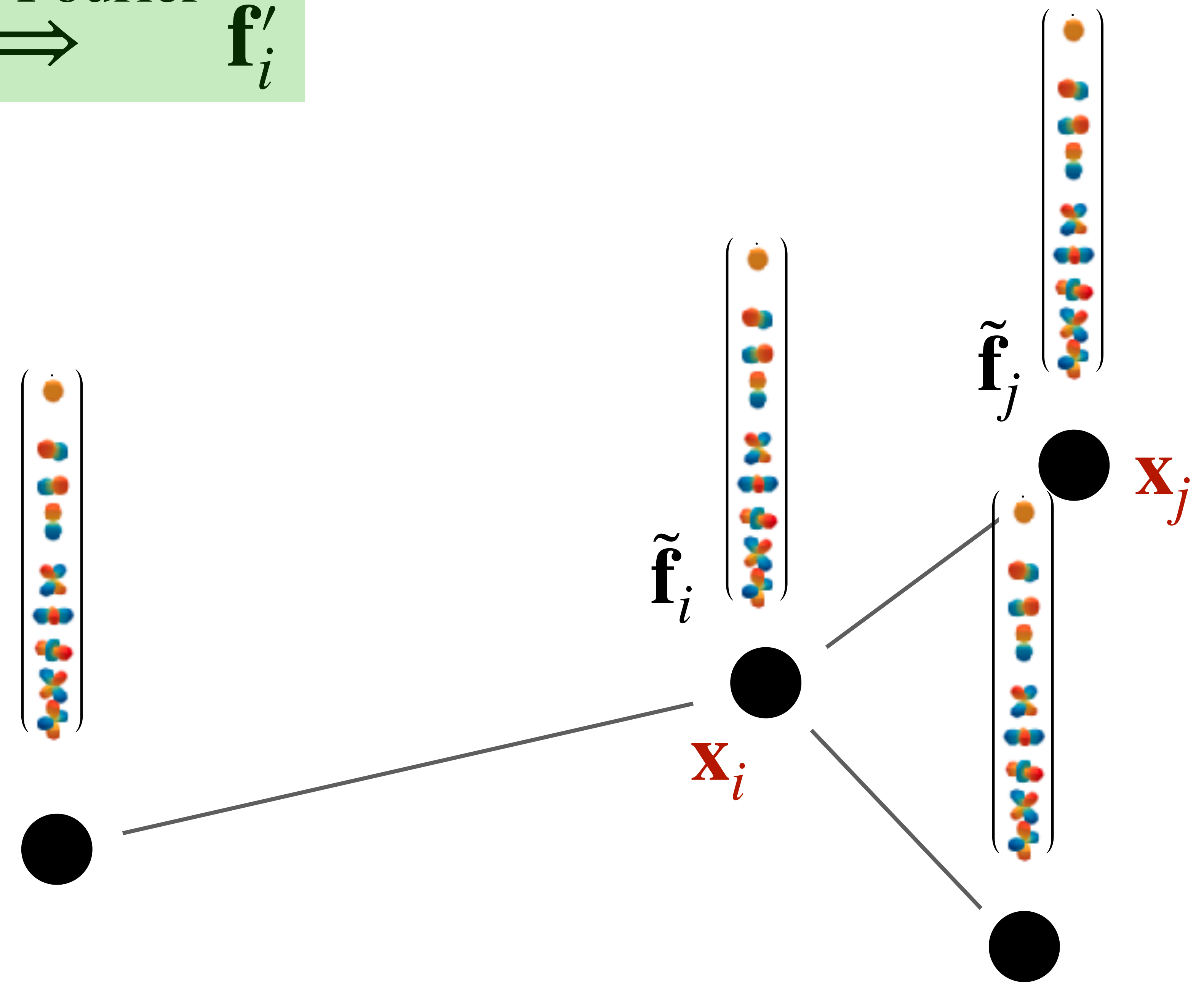
Steerable methods for computational chemistry

Instead of feature vectors $\mathbf{f}_i \in \mathbb{R}^{N_c}$

Work with steerable vectors $\tilde{\mathbf{f}}_i \in V_L^{N_c}$, whose sub-vectors represent functions on $SO(3)$ via a Fourier transform

$$\mathbf{f}'_i(\mathbf{R}) \overset{SO(3)-\text{Fourier}}{\Longleftrightarrow} \tilde{\mathbf{f}}'_i$$





Point cloud sparsely represents feature map

$$\mathbf{f} : \mathbb{R}^3 \rightarrow \mathbb{R}^{Nc}$$

Point cloud sparsely represents steerable vector field

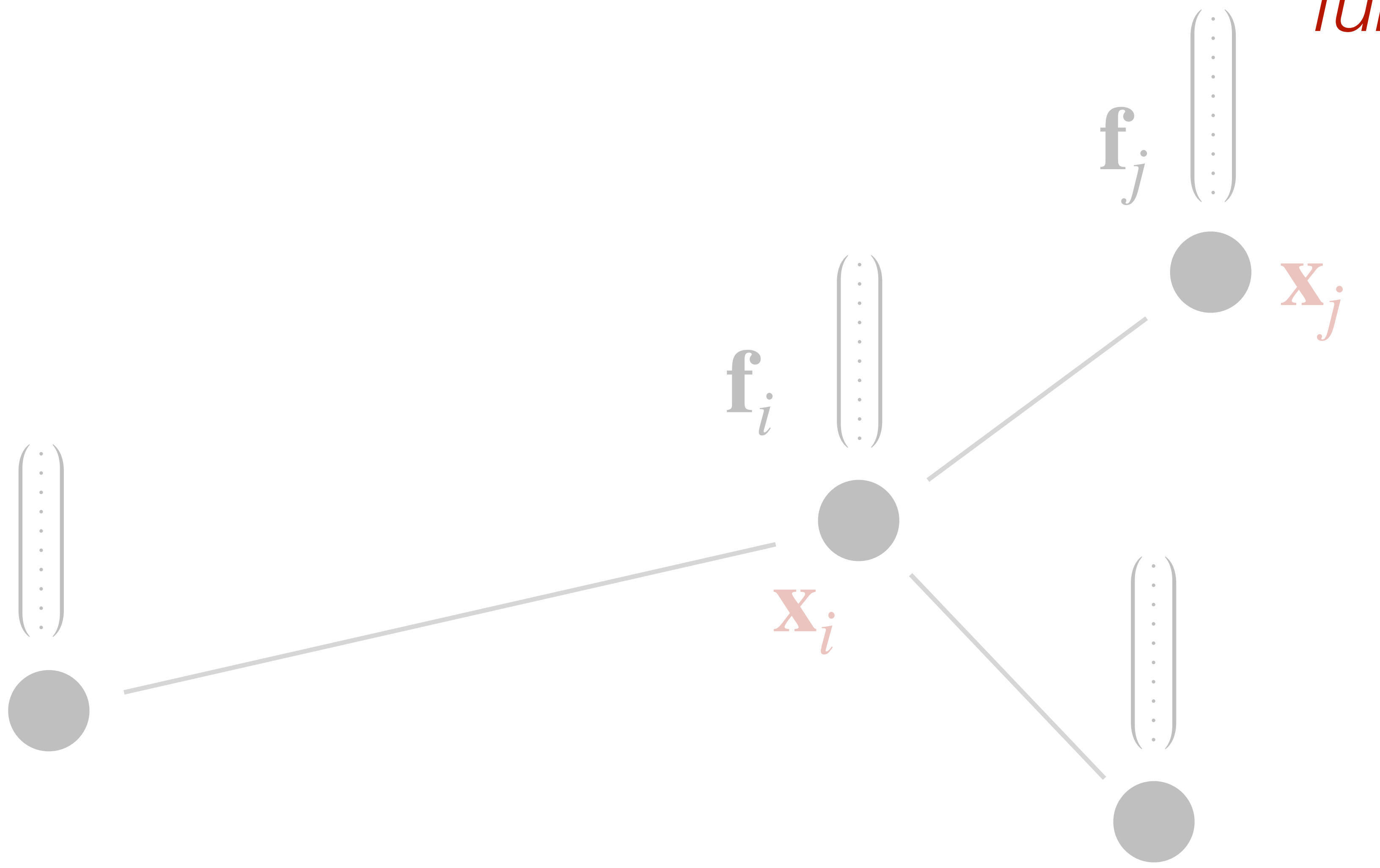
$$\tilde{\mathbf{f}} : \mathbb{R}^3 \rightarrow V_L^{Nc}$$

or a regular feature map on $\mathbb{R}^3 \times SO(3)$

$$\mathbf{f} : \mathbb{R}^3 \times SO(3) \rightarrow \mathbb{R}^{Nc}$$

Steerable methods for computational chemistry

Instead of feature vectors $\mathbf{f}_i \in \mathbb{R}^{N_c}$

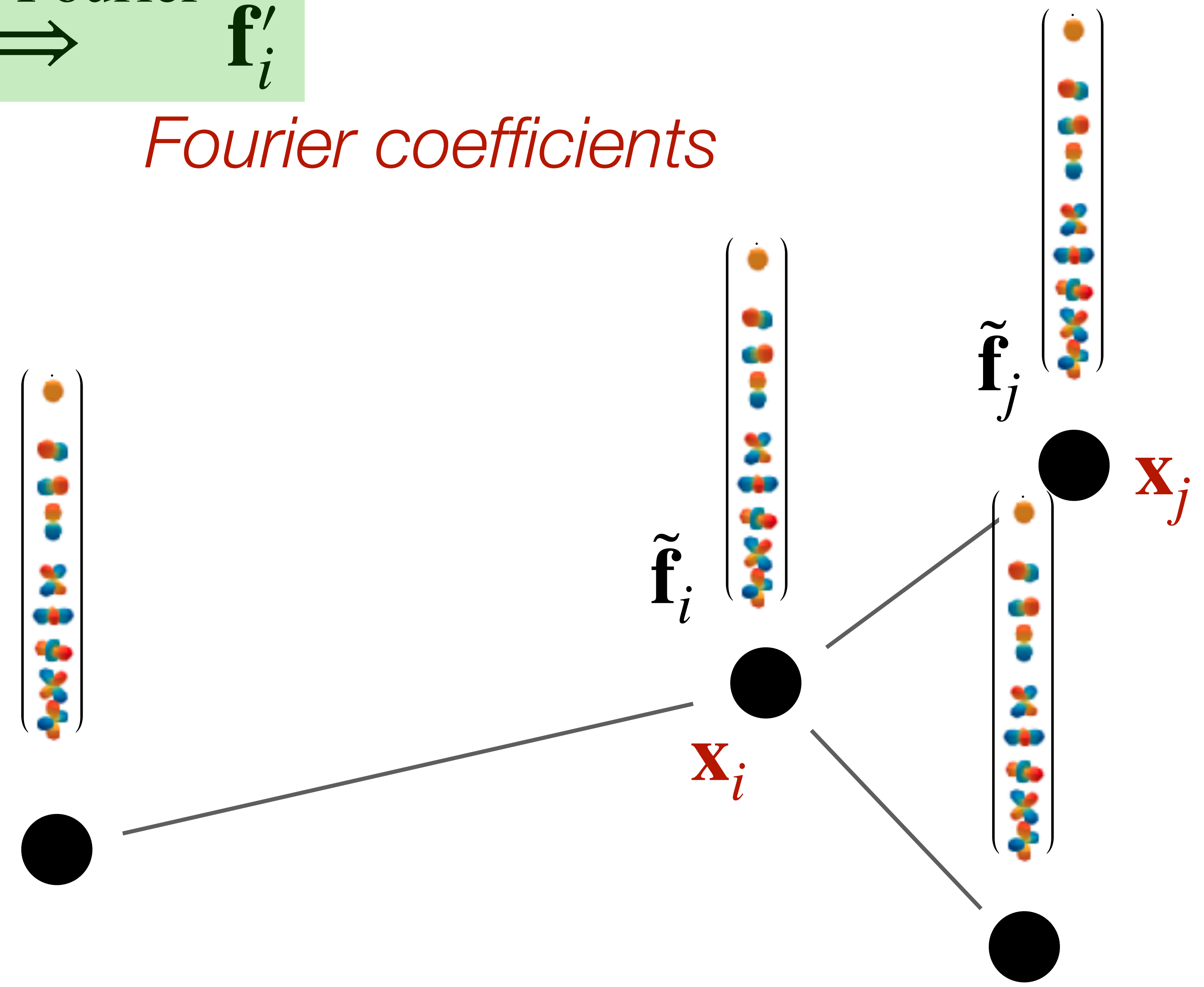


Point cloud sparsely represents feature map
$\mathbf{f} : \mathbb{R}^3 \rightarrow \mathbb{R}^{Nc}$

Work with steerable vectors $\tilde{\mathbf{f}}_i \in V_L^{N_c}$, whose sub-vectors represent functions on $SO(3)$ via a Fourier transform

$$\mathbf{f}'_i(\mathbf{R}) \overset{SO(3)-\text{Fourier}}{\Longleftrightarrow} \tilde{\mathbf{f}}'_i$$

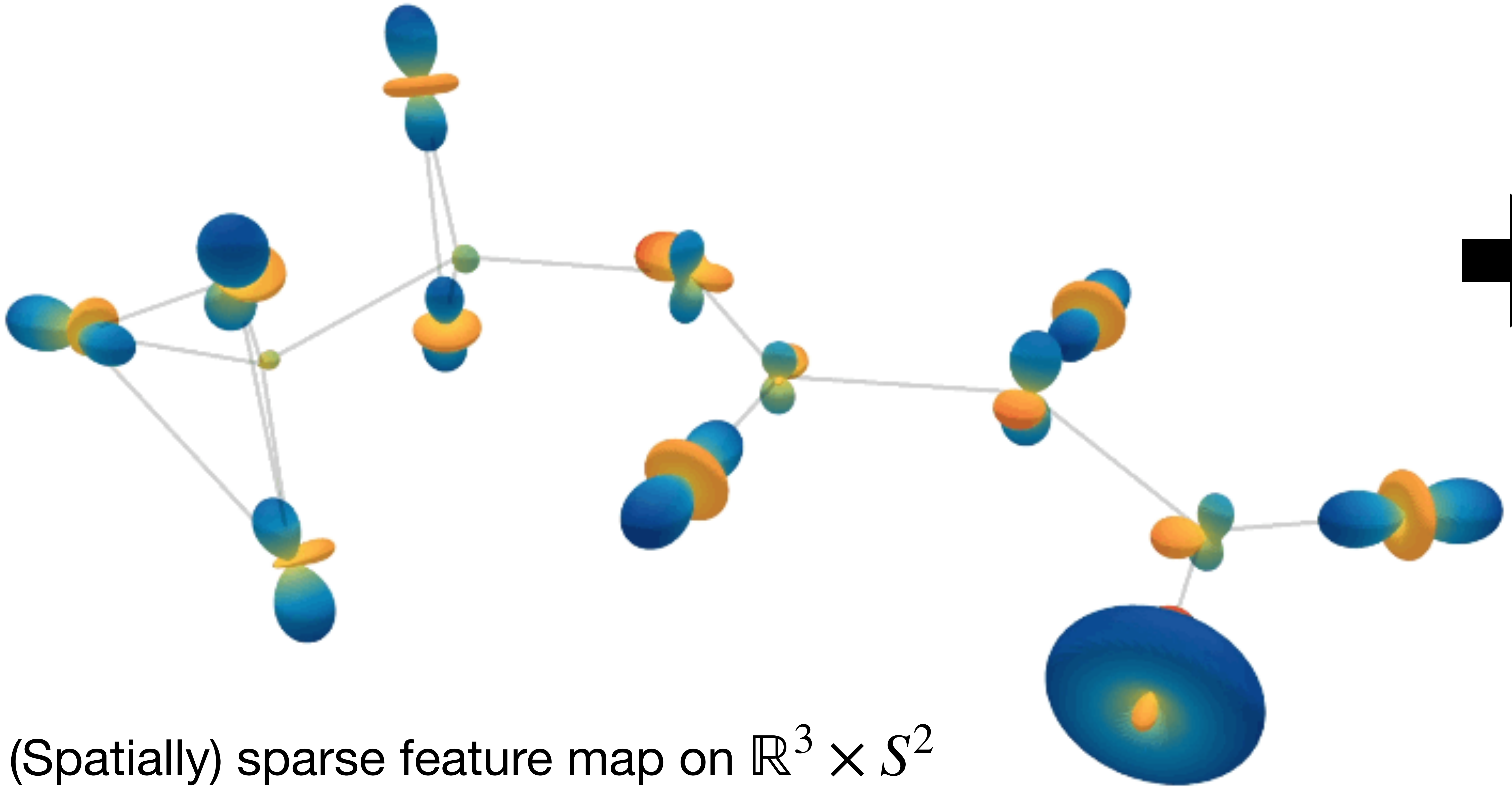
function on $SO(3)$ *Fourier coefficients*



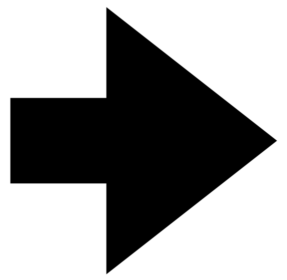
Point cloud sparsely represents steerable vector field
$\tilde{\mathbf{f}} : \mathbb{R}^3 \rightarrow V_L^{Nc}$
or a regular feature map on $\mathbb{R}^3 \times SO(3)$
$\mathbf{f} : \mathbb{R}^3 \times SO(3) \rightarrow \mathbb{R}^{Nc}$

Steerable methods for computational chemistry

Brandstetter, Hesselink, van der Pol, Bekkers, Welling **Geometric and Physical Quantities Improve E(3) Equivariant Message Passing** - arXiv:2110.02905



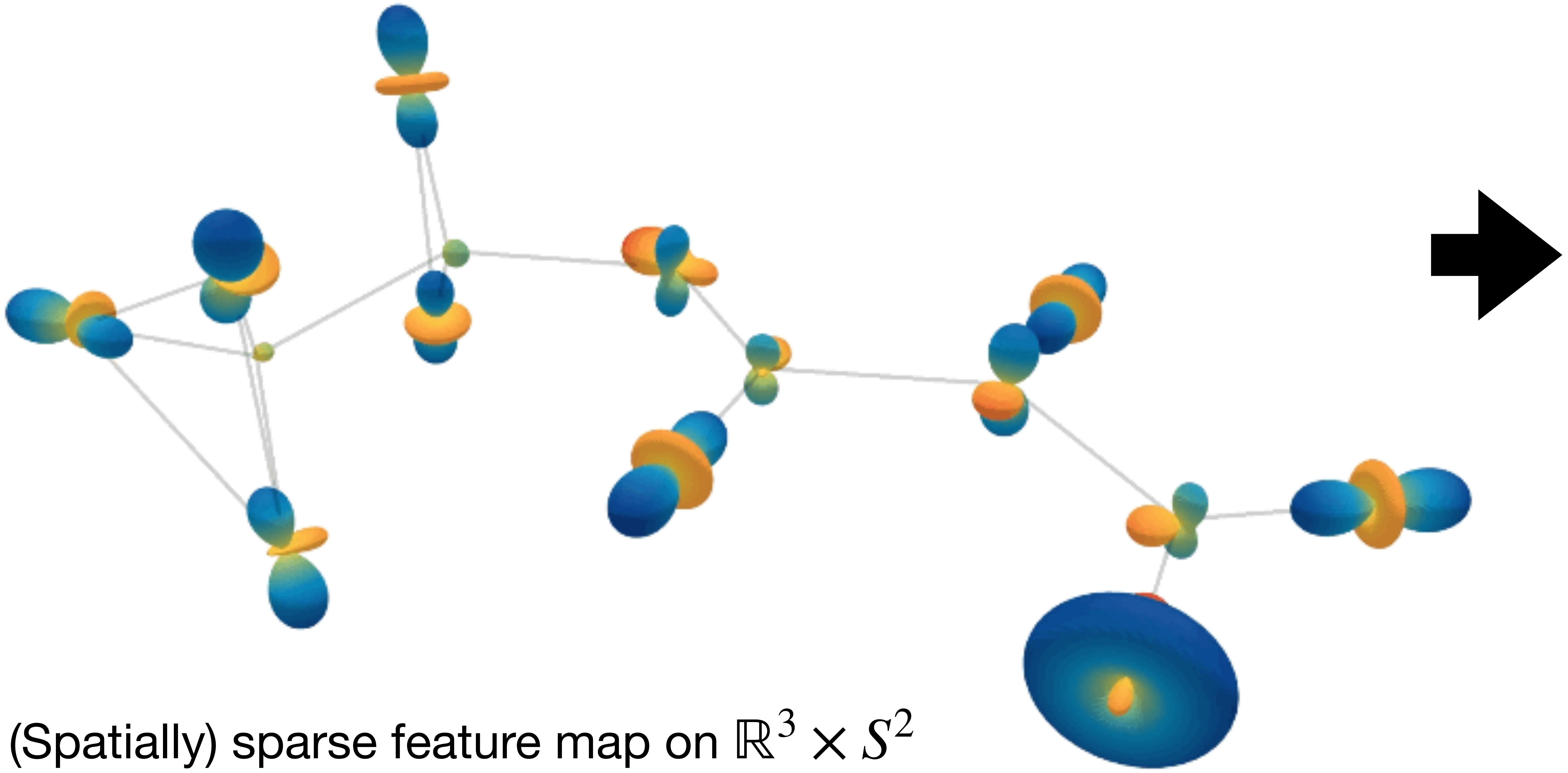
(Spatially) sparse feature map on $\mathbb{R}^3 \times S^2$



Molecular property prediction

Steerable methods for computational chemistry

Brandstetter, Hesselink, van der Pol, Bekkers, Welling **Geometric and Physical Quantities Improve E(3) Equivariant Message Passing** - arXiv:2110.02905



Molecular property prediction

(Spatially) sparse feature map on $\mathbb{R}^3 \times S^2$

Why 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

Group theory: Irreducible Representations

Matrix representations: $\mathbf{D}(g) \in \mathbb{R}^{d \times d}$ is a matrix parametrized by group elements $g \in G$ such that

$$\forall_{g,g' \in G}: \quad \mathbf{D}(g)\mathbf{D}(g') = \mathbf{D}(g \cdot g').$$

Equivalence of matrix representations: Matrix representations $\mathbf{D}(g)$, $\mathbf{D}'(g)$ are equivalent if they relate via a similarity transform ($\mathbf{Q}$ performs change of basis):

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

Reducible/irreducible matrix representations: A matrix representation is called reducible if it can be written as:

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

Group theory: Wigner-D Matrices

Wigner-D matrices of type-l are the irreducible matrix representations of $SO(3)$, are of dimension $(2l+1) \times (2l+1)$, and denoted with $\mathbf{D}^{(l)}(g)$

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

Wigner-D functions: each of the $(2l+1) \times (2l+1)$ components of the Wigner-D matrices will be refered to as type-l Wigner-D functions, denoted with $D^{(l)}_{mn}$, m and n resp. row and column index .

Group theory: Wigner-D Matrices

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

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

Steerable vectors: The $(2l + 1)$-dimensional vector space on which a Wigner-D matrix of type l acts will be called a type l steerable vector space denoted with V_l, its elements will be called steerable vectors.

Steerable methods for computational chemistry

Instead of working with feature maps:

$$\mathbb{R}^3 \to \mathbb{R}^{N_c} \qquad \text{(note } \mathbb{R}^{N_c} = V_0^{N_c}\text{)}$$

We will work with steerable feature maps:

$$\mathbb{R}^3 \to V_0^{N_0} \oplus V_1^{N_1} \oplus \ldots \oplus V_L^{N_L}$$

Instead of transforming the feature vectors via matrix vector multiplication

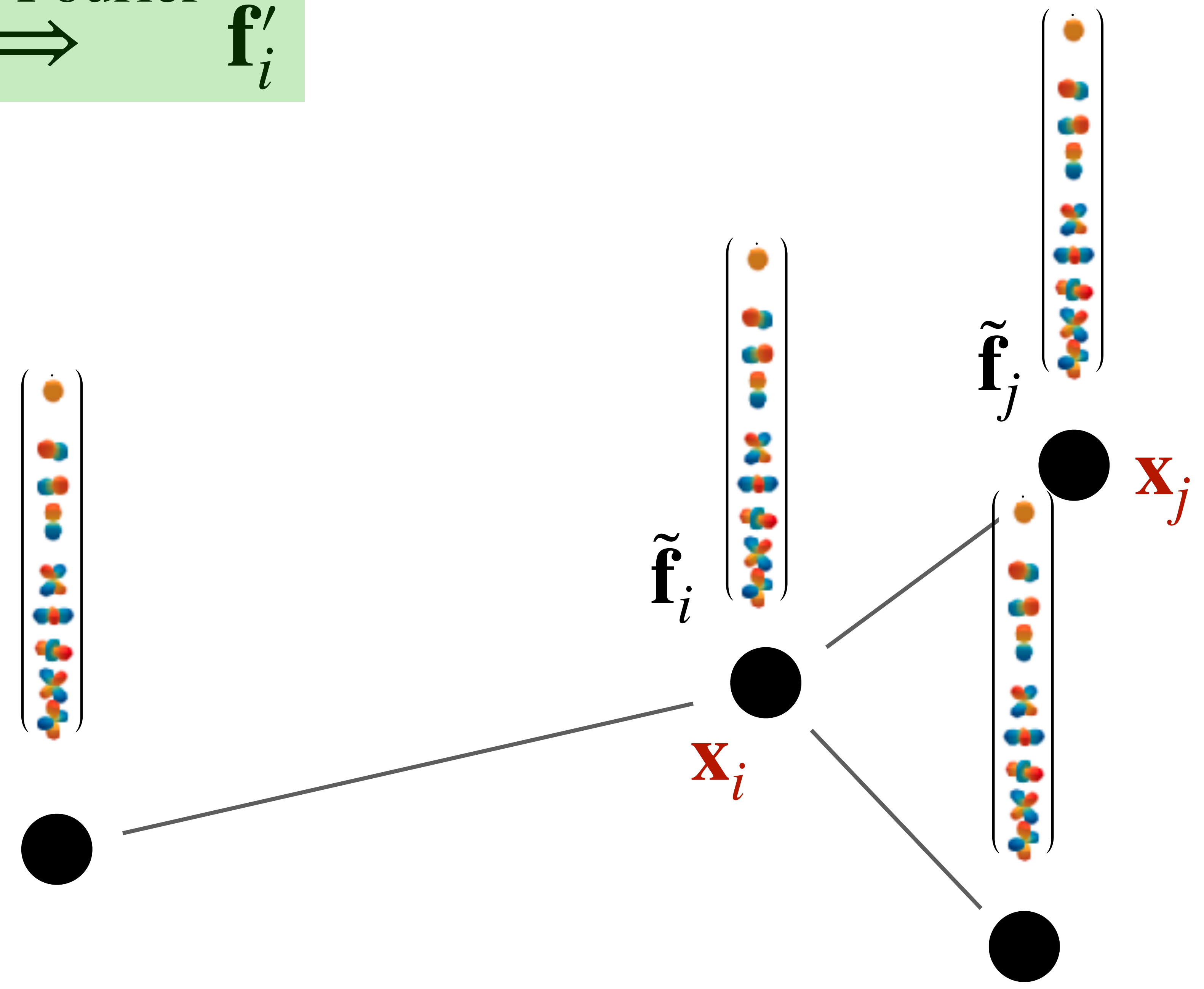
$$\mathbf{W}(\mathbf{x}_j - \mathbf{x}_i)\mathbf{f}_i$$

We transform steerable vectors via the Clebsch-Gordan tensor product

$$\mathbf{Y}(\mathbf{x}_j - \mathbf{x}_i) \otimes_{cg}^{\mathbf{W}(\|\mathbf{x}_j - \mathbf{x}_i\|)} \tilde{\mathbf{f}}_j$$

Work with steerable vectors $\tilde{\mathbf{f}}_i \in V_L^{N_c}$, whose sub-vectors represent functions on S^2 via a spherical Fourier transform

$$\mathbf{f}_i'(\mathbf{R}) \overset{SO(3)-\text{Fourier}}{\Longleftrightarrow} \tilde{\mathbf{f}}_i'$$



Point cloud sparsely represents steerable vector field

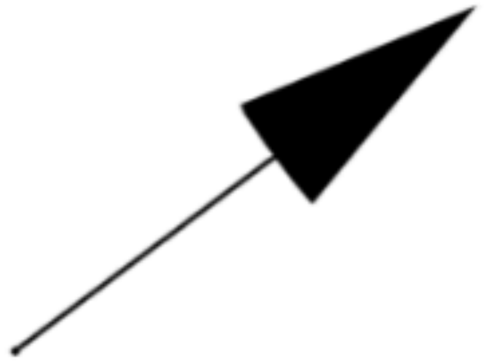
$$\tilde{\mathbf{f}} : \mathbb{R}^3 \to V_L^{Nc}$$

or a regular feature map on $\mathbb{R}^3 \times S^2$

$$\mathbf{f} : \mathbb{R}^3 \times S^2 \to \mathbb{R}^{Nc}$$

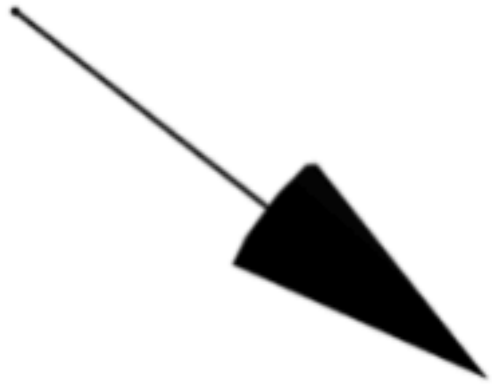
Group theory: Spherical Harmonics Form Steerable Vectors

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



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

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



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

Group theory: 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 $Y_m^{(l)} : S^2 \to \mathbb{R}$

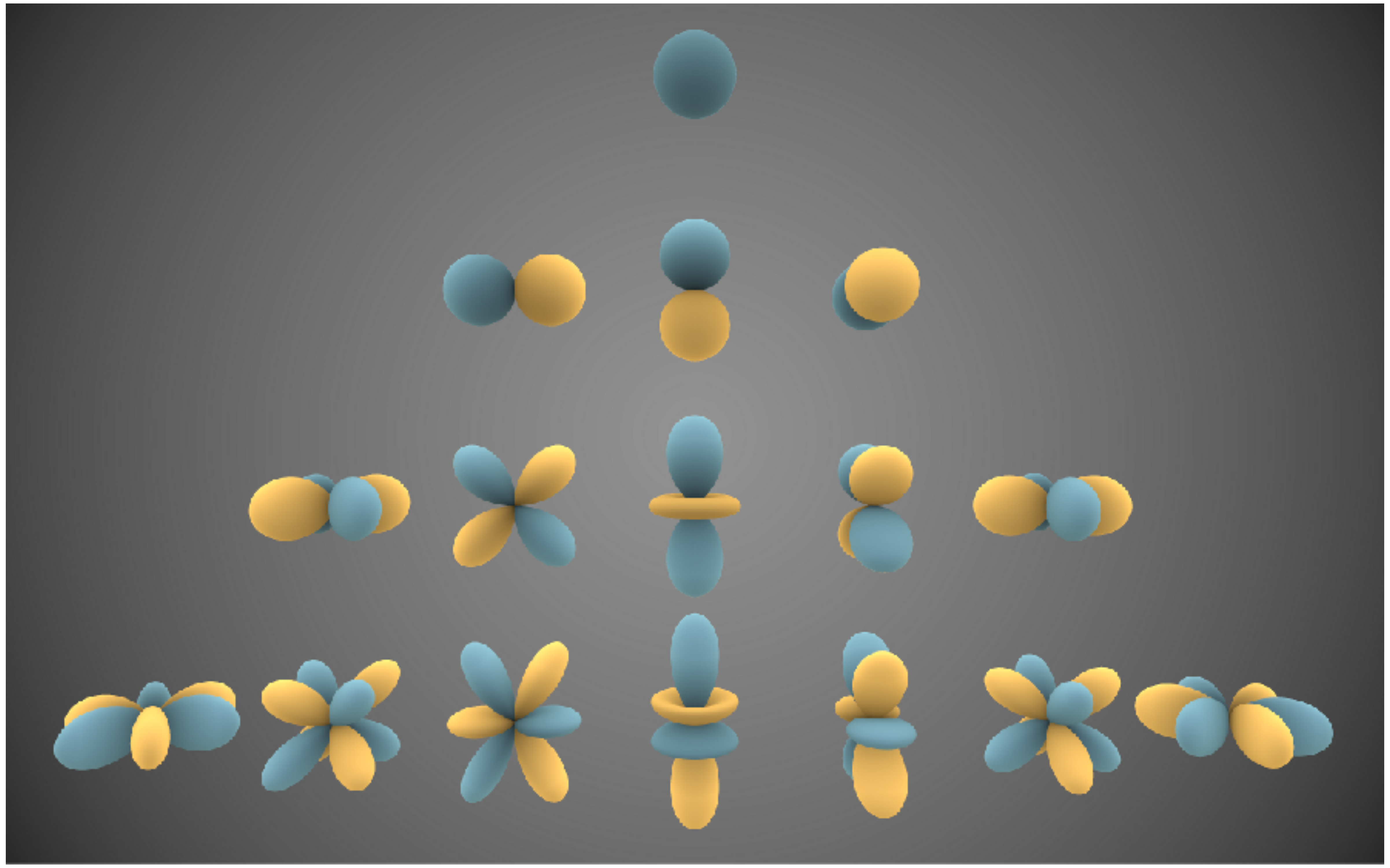


Image: wikipedia

Group theory: Clebsch-Gordan Tensor Product

A note on the CG 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

A note on the CG tensor product

When the inputs are steerable vectors, the tensor product is equivariant via

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

Group theory: Clebsch-Gordan Tensor Product

A note on the CG tensor product

When the inputs are steerable vectors, the tensor product is equivariant via

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

Rotation of the inputs

Group theory: Clebsch-Gordan Tensor Product

A note on the CG tensor product

When the inputs are steerable vectors, the tensor product is equivariant via

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

Rotation of the output via some $\mathbf{D}(g)$

Rotation of the inputs

Group theory: Clebsch-Gordan Tensor Product

A note on the CG tensor product

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

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

The resulting representation $\mathbf{D}(g) = \mathbf{D}^{(l_2)}(g) \otimes \mathbf{D}^{(l_1)}$ 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

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

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 $h^{(l)}_m$ 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

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

in which $C^{(l,m)}_{(l_1,m_1)(l_2,m_2)}$ 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 methods for computational chemistry

Instead of working with feature maps:

$\mathbb{R}^3 \to \mathbb{R}^{N_c}$ (note $\mathbb{R}^{N_c} = V_0^{N_c}$)

We will work with steerable feature maps:

$\mathbb{R}^3 \to V_0^{N_0} \oplus V_1^{N_1} \oplus \ldots \oplus V_L^{N_L}$

Instead of transforming the feature vectors via matrix vector multiplication

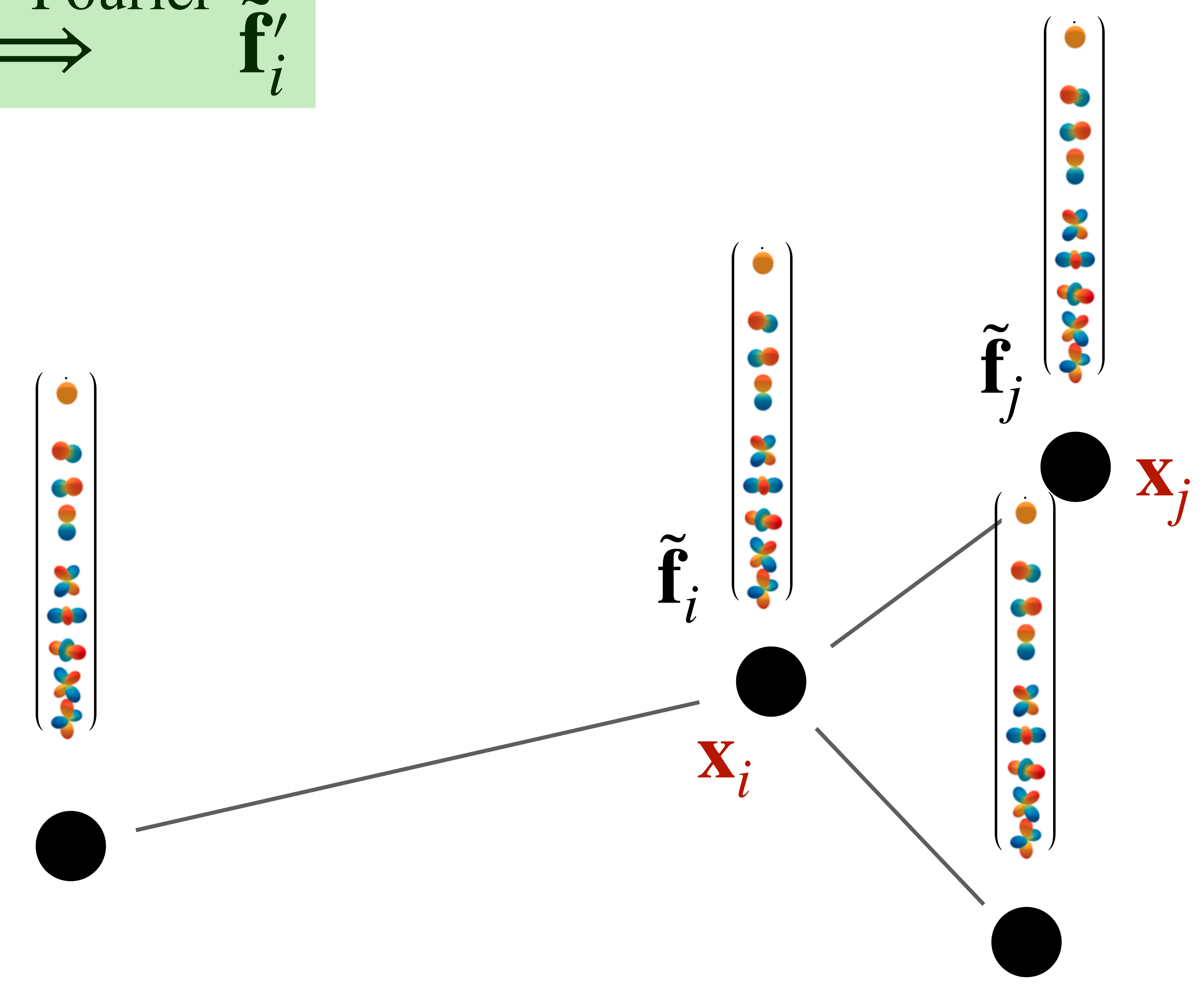
$\mathbf{W}(\mathbf{x}_j - \mathbf{x}_i)\mathbf{f}_i$

We transform steerable vectors via the Clebsch-Gordan tensor product

$\mathbf{Y}(\mathbf{x}_j - \mathbf{x}_i) \otimes_{cg}^{\mathbf{W}(\|\mathbf{x}_j - \mathbf{x}_i\|)} \tilde{\mathbf{f}}_j$

Work with steerable vectors $\tilde{\mathbf{f}}_i \in V_L^{N_c}$, whose sub-vectors represent functions on S^2 via a spherical Fourier transform

$\mathbf{f}'_i(\mathbf{R}) \overset{SO(3)-\text{Fourier}}{\Longleftrightarrow} \tilde{\mathbf{f}}'_i$



Point cloud sparsely represents steerable vector field

$\tilde{\mathbf{f}} : \mathbb{R}^3 \to V_L^{Nc}$

or a regular feature map on $\mathbb{R}^3 \times S^2$

$\mathbf{f} : \mathbb{R}^3 \times S^2 \to \mathbb{R}^{Nc}$

Steerable Neural Networks

Steerable group convolutions on $SE(3)$ from a scalar feature map to (up to) type-L feature map

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

$\Leftrightarrow$

Regular $SE(3)$ lifting convolutions:

$$\tilde{f}'(\mathbf{x}, \mathbf{R}) = \int_{\mathbb{R}^3} k(\mathbf{R}^{-1}(\mathbf{x}'-\mathbf{x}))\, f(\mathbf{x}')\, d\mathbf{x}'$$

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

Steerable Neural Networks

Steerable group convolutions on $SE(3)$ from a scalar feature map to (up to) type-L feature map

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

$\Leftrightarrow$

Regular $SE(3)$ lifting convolutions:

$$\tilde{f}'(\mathbf{x}, \mathbf{R}) = \int_{\mathbb{R}^3} k(\mathbf{R}^{-1}(\mathbf{x}'-\mathbf{x}))\, f(\mathbf{x}')\, d\mathbf{x}'$$

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

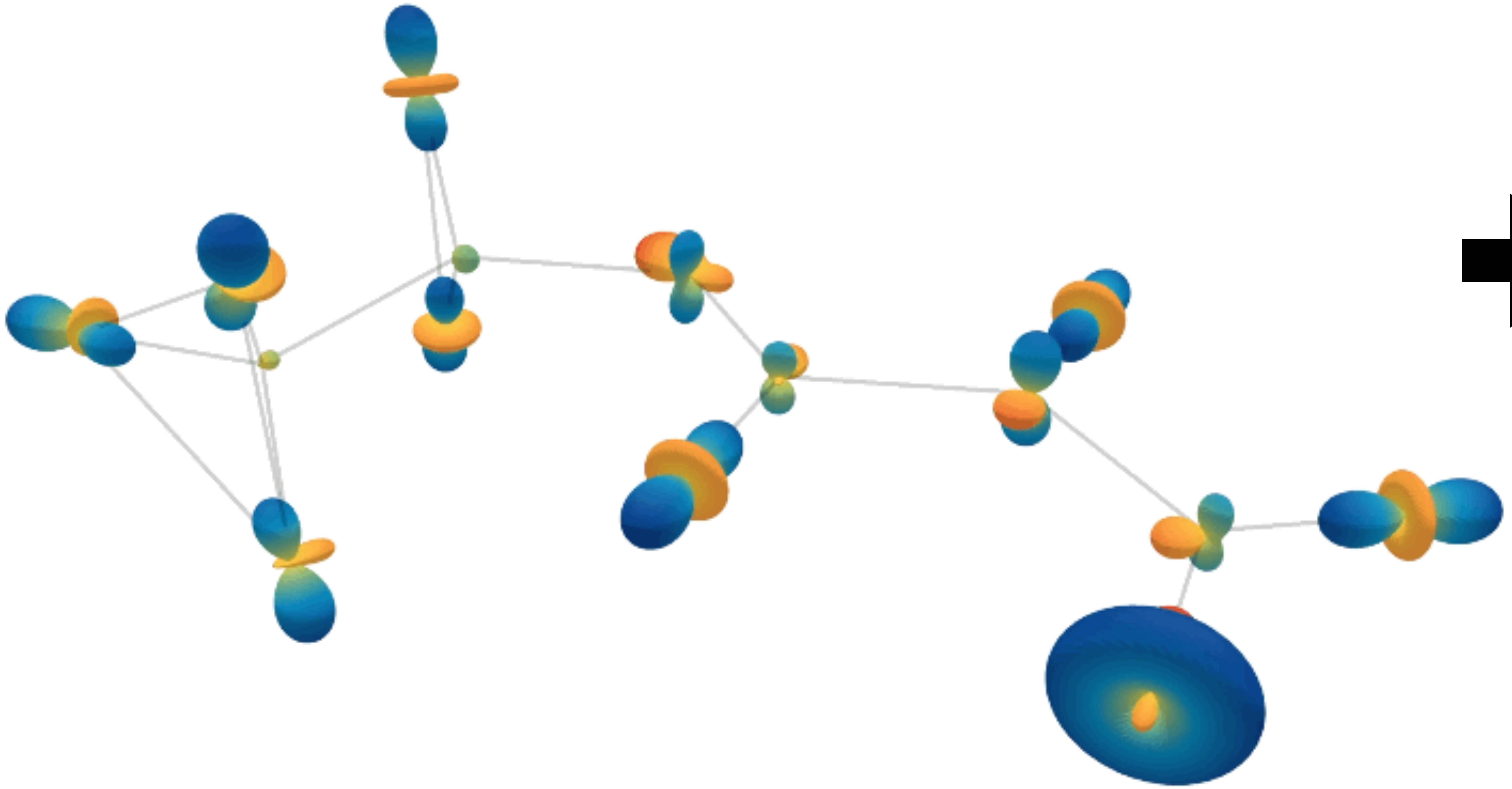
Generalized SO(3) convolutions with feature fields of arbitrary types

$$\mathbb{R}^3 \to V_0^{N_0} \oplus V_1^{N_1} \oplus \ldots \oplus V_L^{N_L}$$

Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch–gordan nets: a fully fourier space spherical convolutional neural network. Advances in Neural Information Processing Systems, 31:10117– 10126, 2018.

Steerable methods for computational chemistry

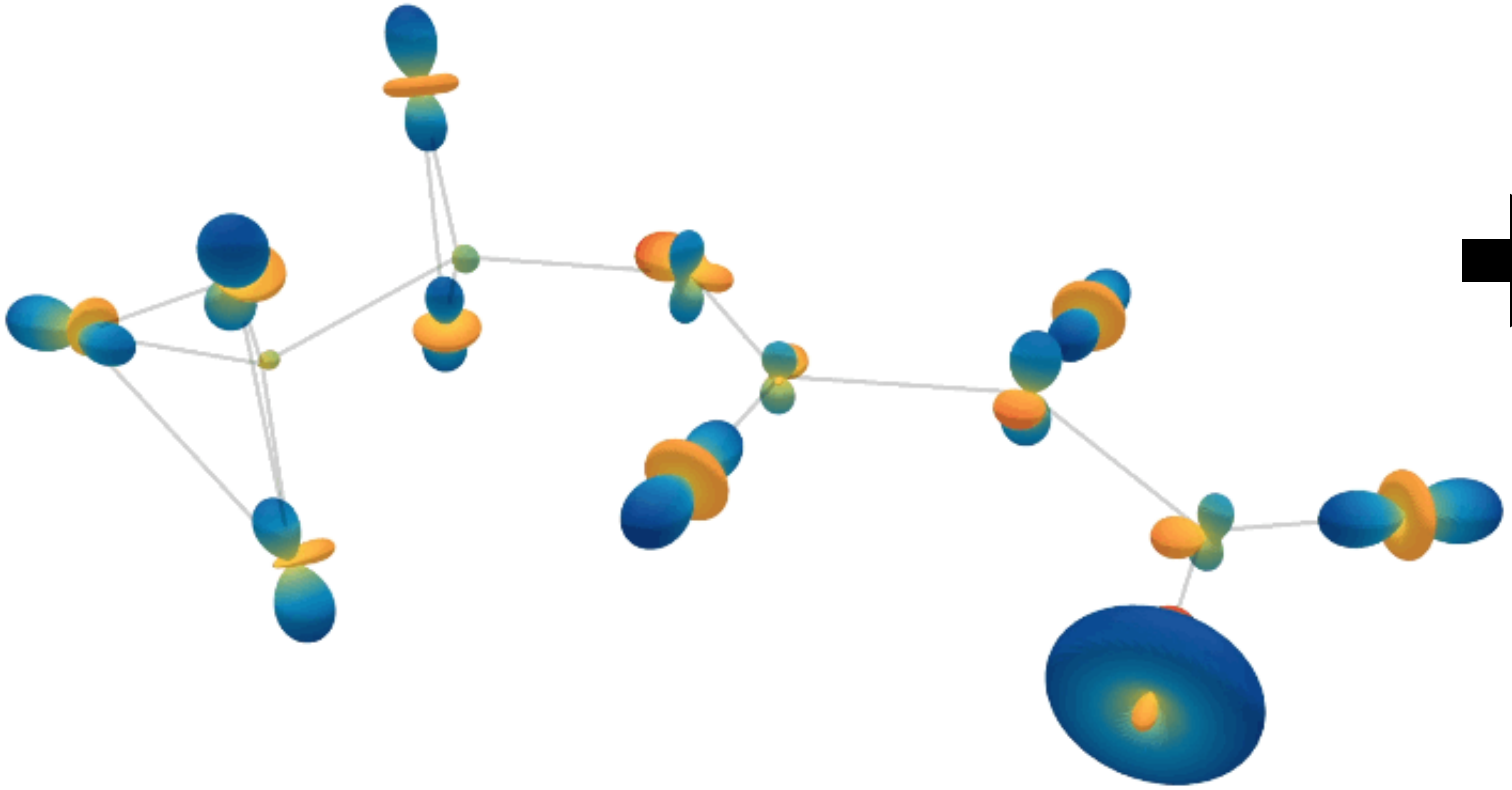
Brandstetter, Hesselink, van der Pol, Bekkers, Welling **Geometric and Physical Quantities Improve E(3) Equivariant Message Passing** - arXiv:2110.02905



Molecular property prediction

Steerable methods for computational chemistry

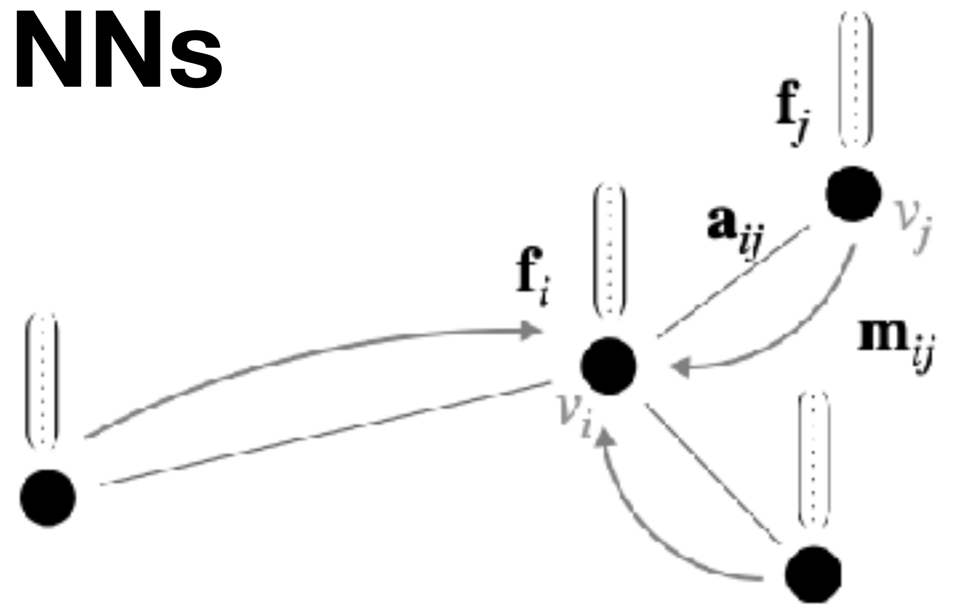
Brandstetter, Hesselink, van der Pol, Bekkers, Welling **Geometric and Physical Quantities Improve E(3) Equivariant Message Passing** - arXiv:2110.02905



Molecular property prediction

Graph Neural Networks for Point Clouds

Message passing NNs

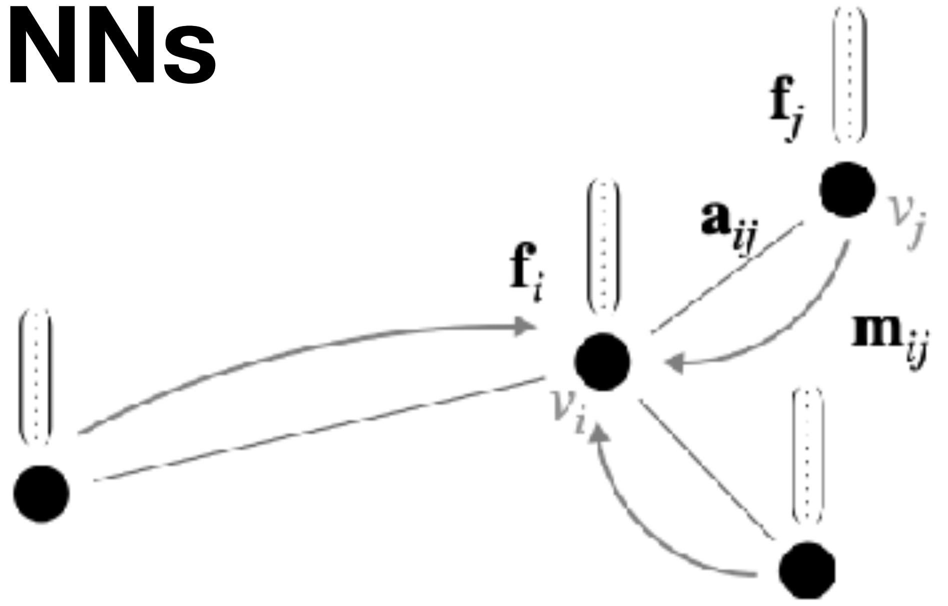


Compute messages: $\mathbf{m}_{ij} = \phi_m(\mathbf{f}_i, \mathbf{f}_j, \mathbf{a}_{ij})$

Aggregate and update: $\mathbf{f}'_i = \phi_f\left(\mathbf{f}_i, \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}\right)$

Graph Neural Networks for Point Clouds

Message passing NNs



Compute messages: $\mathbf{m}_{ij} = \phi_m(\mathbf{f}_i, \mathbf{f}_j, \mathbf{a}_{ij})$

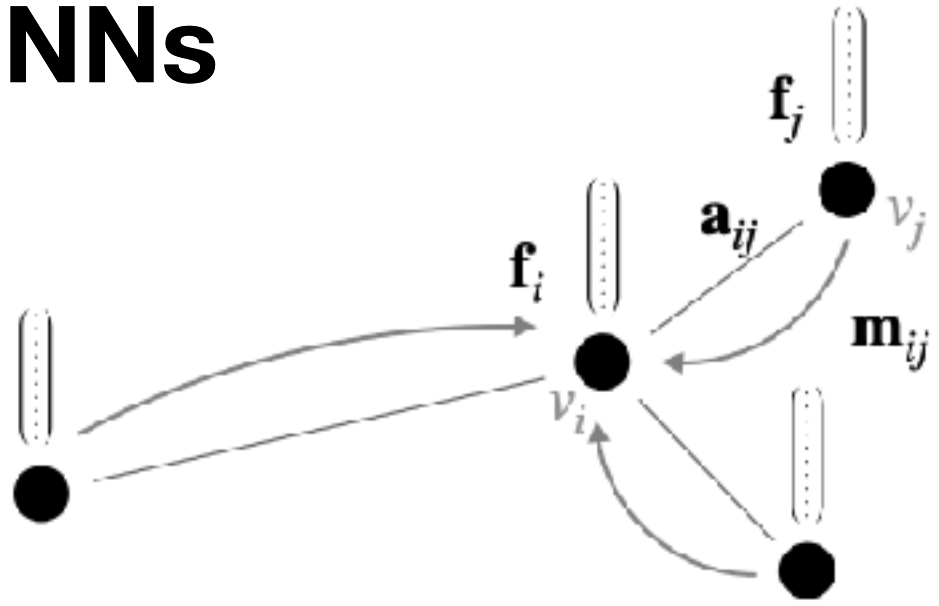
Aggregate and update: $\mathbf{f}_i' = \phi_f\left(\mathbf{f}_i, \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}\right)$

Classic (isotropic) point convolutions

$$\mathbf{m}_{ij} = \mathbf{W}(\|\mathbf{x}_j - \mathbf{x}_i\|)\mathbf{f}_j$$

Graph Neural Networks for Point Clouds

Message passing NNs



Compute messages: $\mathbf{m}_{ij} = \phi_m(\mathbf{f}_i, \mathbf{f}_j, \mathbf{a}_{ij})$

Aggregate and update: $\mathbf{f}'_i = \phi_f\left(\mathbf{f}_i, \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}\right)$

Classic (isotropic) point convolutions

$$\mathbf{m}_{ij} = \mathbf{W}(\|\mathbf{x}_j - \mathbf{x}_i\|)\mathbf{f}_j$$

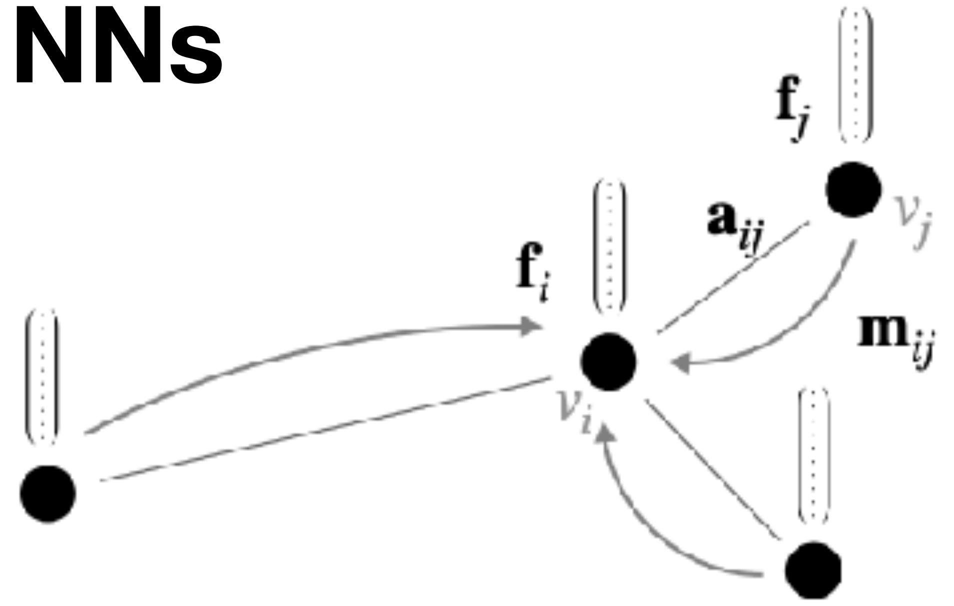
Steerable (anisotropic) G-CNNs

$$\mathbf{m}_{ij} = \mathbf{W}_{\tilde{\mathbf{a}}_{ij}}(\|\mathbf{x}_j - \mathbf{x}_i\|)\tilde{\mathbf{f}}_j$$

$$:= \tilde{\mathbf{f}}_j \otimes_{cg}^{\mathbf{W}(\|\mathbf{x}_j - \mathbf{x}_i\|)} \tilde{\mathbf{a}}_{ij}$$

Graph Neural Networks for Point Clouds

Message passing NNs



Compute messages: $\mathbf{m}_{ij} = \phi_m(\mathbf{f}_i, \mathbf{f}_j, \mathbf{a}_{ij})$

Aggregate and update: $\mathbf{f}_i' = \phi_f\left(\mathbf{f}_i, \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}\right)$

Classic (isotropic) point convolutions

$$\mathbf{m}_{ij} = \mathbf{W}(\|\mathbf{x}_j - \mathbf{x}_i\|)\mathbf{f}_j$$

$\tilde{\mathbf{a}}_{ij} = 1$

$\tilde{\mathbf{f}}_j \in V_0$

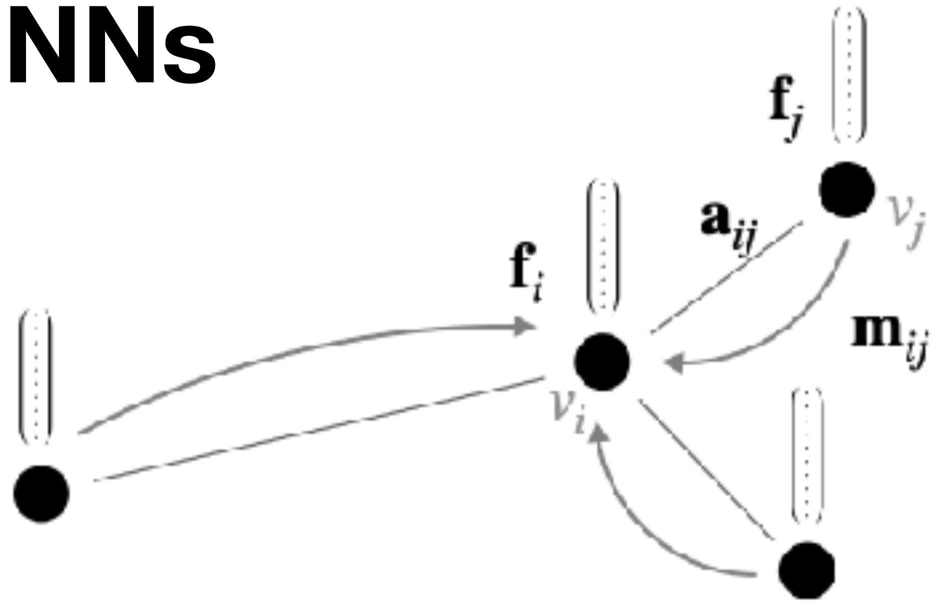
Steerable (anisotropic) G-CNNs

$$\mathbf{m}_{ij} = \mathbf{W}_{\tilde{\mathbf{a}}_{ij}}(\|\mathbf{x}_j - \mathbf{x}_i\|)\tilde{\mathbf{f}}_j$$

$$:= \tilde{\mathbf{f}}_j \otimes_{cg}^{\mathbf{W}(\|\mathbf{x}_j - \mathbf{x}_i\|)} \tilde{\mathbf{a}}_{ij}$$

Graph Neural Networks for Point Clouds

Message passing NNs



Compute messages: $\mathbf{m}_{ij} = \phi_m(\mathbf{f}_i, \mathbf{f}_j, \mathbf{a}_{ij})$

Aggregate and update: $\mathbf{f}_i' = \phi_f\left(\mathbf{f}_i, \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}\right)$

Classic (isotropic) point convolutions

$$\mathbf{m}_{ij} = \mathbf{W}(\|\mathbf{x}_j - \mathbf{x}_i\|)\mathbf{f}_j$$

Invariant Message Passing NNs

$$\mathbf{m}_{ij} = \mathrm{MLP}(\mathbf{f}_i, \mathbf{f}_j, \|\mathbf{x}_j - \mathbf{x}_i\|)$$

$\tilde{\mathbf{a}}_{ij} = 1$

$\tilde{\mathbf{f}}_j \in V_0$

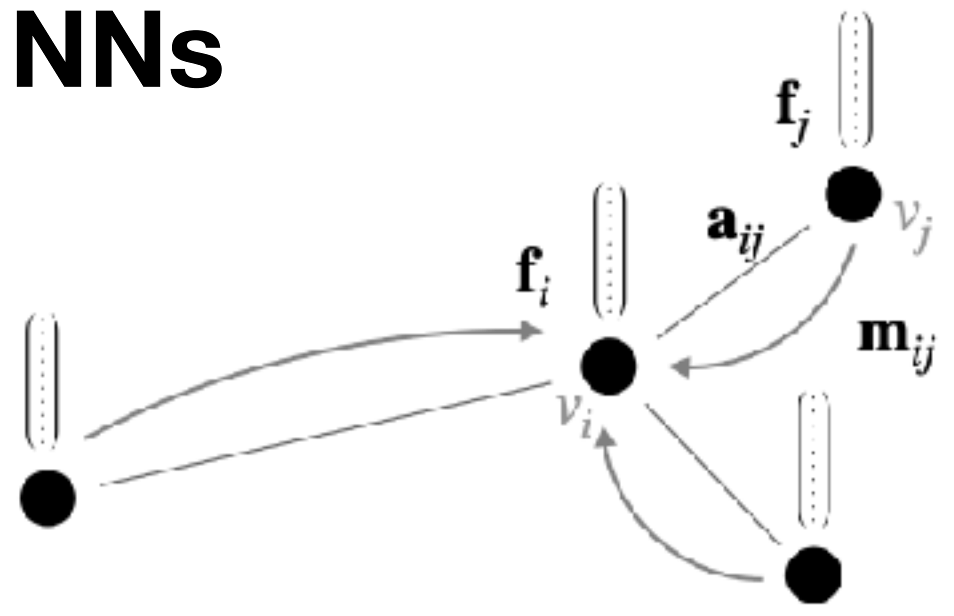
Steerable (anisotropic) G-CNNs

$$\mathbf{m}_{ij} = \mathbf{W}_{\tilde{\mathbf{a}}_{ij}}(\|\mathbf{x}_j - \mathbf{x}_i\|)\tilde{\mathbf{f}}_j$$

$$:= \tilde{\mathbf{f}}_j \otimes_{cg}^{\mathbf{W}(\|\mathbf{x}_j - \mathbf{x}_i\|)} \tilde{\mathbf{a}}_{ij}$$

Graph Neural Networks for Point Clouds

Message passing NNs



Compute messages: $\mathbf{m}_{ij} = \phi_m(\mathbf{f}_i, \mathbf{f}_j, \mathbf{a}_{ij})$

Aggregate and update: $\mathbf{f}_i' = \phi_f\left(\mathbf{f}_i, \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}\right)$

Classic (isotropic) point convolutions

$$\mathbf{m}_{ij} = \mathbf{W}(\|\mathbf{x}_j - \mathbf{x}_i\|)\mathbf{f}_j$$

Invariant Message Passing NNs

$$\mathbf{m}_{ij} = \mathrm{MLP}(\mathbf{f}_i, \mathbf{f}_j, \|\mathbf{x}_j - \mathbf{x}_i\|)$$

$\tilde{\mathbf{a}}_{ij} = 1$

$\tilde{\mathbf{f}}_j \in V_0$

Steerable (anisotropic) G-CNNs

$$\mathbf{m}_{ij} = \mathbf{W}_{\tilde{\mathbf{a}}_{ij}}(\|\mathbf{x}_j - \mathbf{x}_i\|)\tilde{\mathbf{f}}_j$$

$$:= \tilde{\mathbf{f}}_j \otimes_{cg}^{\mathbf{W}(\|\mathbf{x}_j - \mathbf{x}_i\|)} \tilde{\mathbf{a}}_{ij}$$

Steerable Message Passing NNs

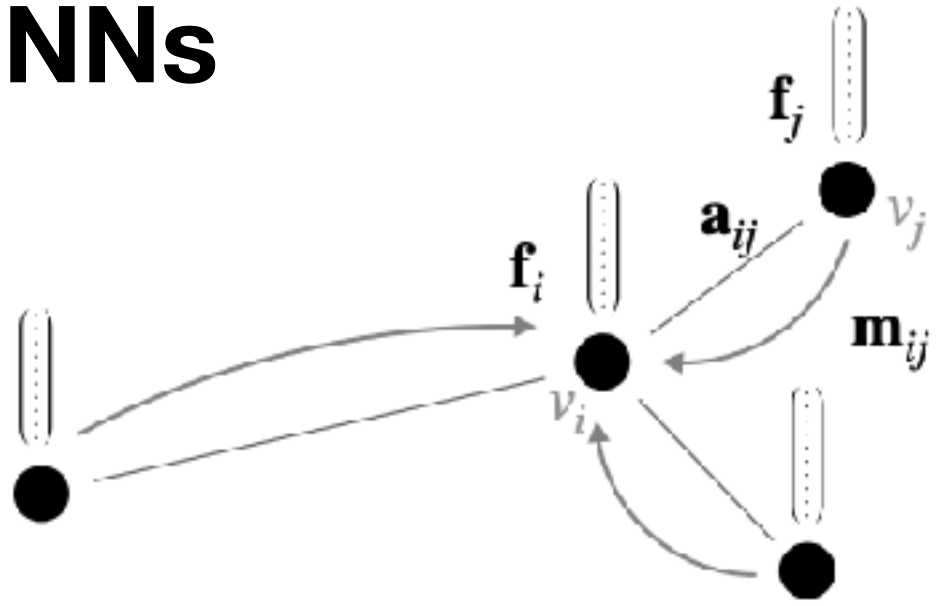
$$\mathbf{m}_{ij} = \widetilde{\mathrm{MLP}}(\mathbf{f}_i, \mathbf{f}_j, \|\mathbf{x}_j - \mathbf{x}_i\|)$$

With steerable MLP:

$$\widetilde{\mathrm{MLP}}_{\tilde{\mathbf{a}}_{ij}}(\tilde{\mathbf{f}}_i, \tilde{\mathbf{f}}_j, \|\mathbf{x}_j - \mathbf{x}_i\|^2) := \sigma(\mathbf{W}^{(n)}_{\tilde{\mathbf{a}}_{ij}}(\ldots(\sigma(\mathbf{W}^{(1)}_{\tilde{\mathbf{a}}_{ij}}\tilde{\mathbf{h}}_i))))$$

Graph Neural Networks for Point Clouds

Message passing NNs



Compute messages: $\mathbf{m}_{ij} = \phi_m(\mathbf{f}_i, \mathbf{f}_j, \mathbf{a}_{ij})$

Aggregate and update: $\mathbf{f}_i' = \phi_f\left(\mathbf{f}_i, \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}\right)$

Classic (isotropic) point convolutions

$$\mathbf{m}_{ij} = \mathbf{W}(\|\mathbf{x}_j - \mathbf{x}_i\|)\mathbf{f}_j$$

Linear convolution

Invariant Message Passing NNs

$$\mathbf{m}_{ij} = \mathrm{MLP}(\mathbf{f}_i, \mathbf{f}_j, \|\mathbf{x}_j - \mathbf{x}_i\|)$$

$\tilde{\mathbf{a}}_{ij} = 1$

$\tilde{\mathbf{f}}_j \in V_0$

Steerable (anisotropic) G-CNNs

$$\mathbf{m}_{ij} = \mathbf{W}_{\tilde{\mathbf{a}}_{ij}}(\|\mathbf{x}_j - \mathbf{x}_i\|)\tilde{\mathbf{f}}_j$$

$$:= \tilde{\mathbf{f}}_j \otimes_{cg}^{\mathbf{W}(\|\mathbf{x}_j - \mathbf{x}_i\|)} \tilde{\mathbf{a}}_{ij}$$

Steerable Message Passing NNs

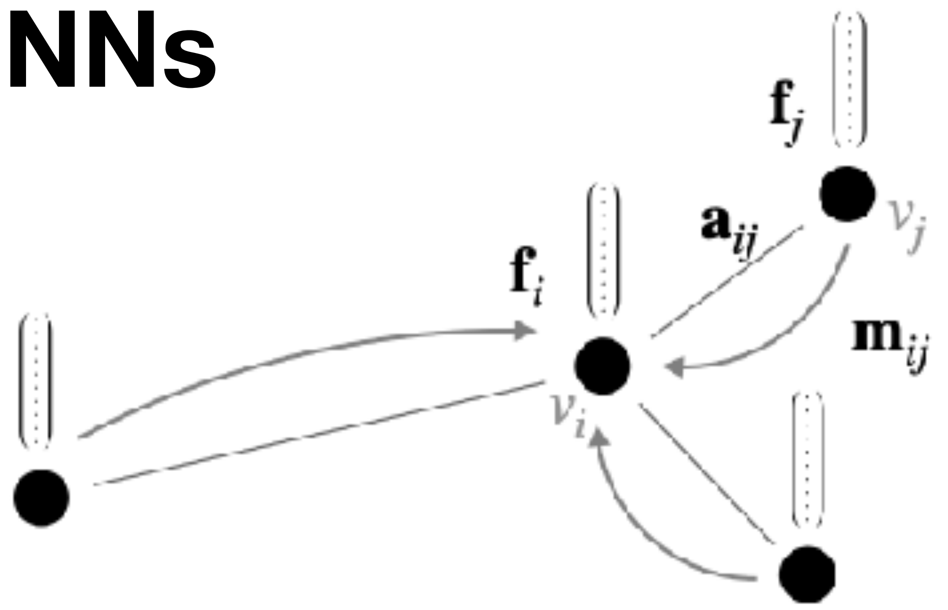
$$\mathbf{m}_{ij} = \widetilde{\mathrm{MLP}}(\mathbf{f}_i, \mathbf{f}_j, \|\mathbf{x}_j - \mathbf{x}_i\|)$$

With steerable MLP:

$$\widetilde{\mathrm{MLP}}_{\tilde{\mathbf{a}}_{ij}}(\tilde{\mathbf{f}}_i, \tilde{\mathbf{f}}_j, \|\mathbf{x}_j - \mathbf{x}_i\|^2) := \sigma(\mathbf{W}^{(n)}_{\tilde{\mathbf{a}}_{ij}}(\ldots(\sigma(\mathbf{W}^{(1)}_{\tilde{\mathbf{a}}_{ij}}\tilde{\mathbf{h}}_i))))$$

Graph Neural Networks for Point Clouds

Message passing NNs



Compute messages: $$\mathbf{m}_{ij} = \phi_m(\mathbf{f}_i, \mathbf{f}_j, \mathbf{a}_{ij})$$

Aggregate and update: $$\mathbf{f}_i' = \phi_f\left(\mathbf{f}_i, \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}\right)$$

Classic (isotropic) point convolutions

$$\mathbf{m}_{ij} = \mathbf{W}(\|\mathbf{x}_j - \mathbf{x}_i\|)\mathbf{f}_j$$

Linear convolution

Invariant Message Passing NNs

$$\mathbf{m}_{ij} = \mathrm{MLP}(\mathbf{f}_i, \mathbf{f}_j, \|\mathbf{x}_j - \mathbf{x}_i\|)$$

Non-linear “convolution”

$\tilde{\mathbf{a}}_{ij} = 1$

$\tilde{\mathbf{f}}_j \in V_0$

Steerable (anisotropic) G-CNNs

$$\mathbf{m}_{ij} = \mathbf{W}_{\tilde{\mathbf{a}}_{ij}}(\|\mathbf{x}_j - \mathbf{x}_i\|)\tilde{\mathbf{f}}_j$$

$$:= \tilde{\mathbf{f}}_j \otimes_{cg}^{\mathbf{W}(\|\mathbf{x}_j - \mathbf{x}_i\|)} \tilde{\mathbf{a}}_{ij}$$

Steerable Message Passing NNs

$$\mathbf{m}_{ij} = \widetilde{\mathrm{MLP}}(\mathbf{f}_i, \mathbf{f}_j, \|\mathbf{x}_j - \mathbf{x}_i\|)$$

With steerable MLP:

$$\widetilde{\mathrm{MLP}}_{\tilde{\mathbf{a}}_{ij}}(\tilde{\mathbf{f}}_i, \tilde{\mathbf{f}}_j, \|\mathbf{x}_j - \mathbf{x}_i\|^2) := \sigma(\mathbf{W}^{(n)}_{\tilde{\mathbf{a}}_{ij}}(\ldots(\sigma(\mathbf{W}^{(1)}_{\tilde{\mathbf{a}}_{ij}}\tilde{\mathbf{h}}_i))))$$

Task: Trajectory prediction N-body problem

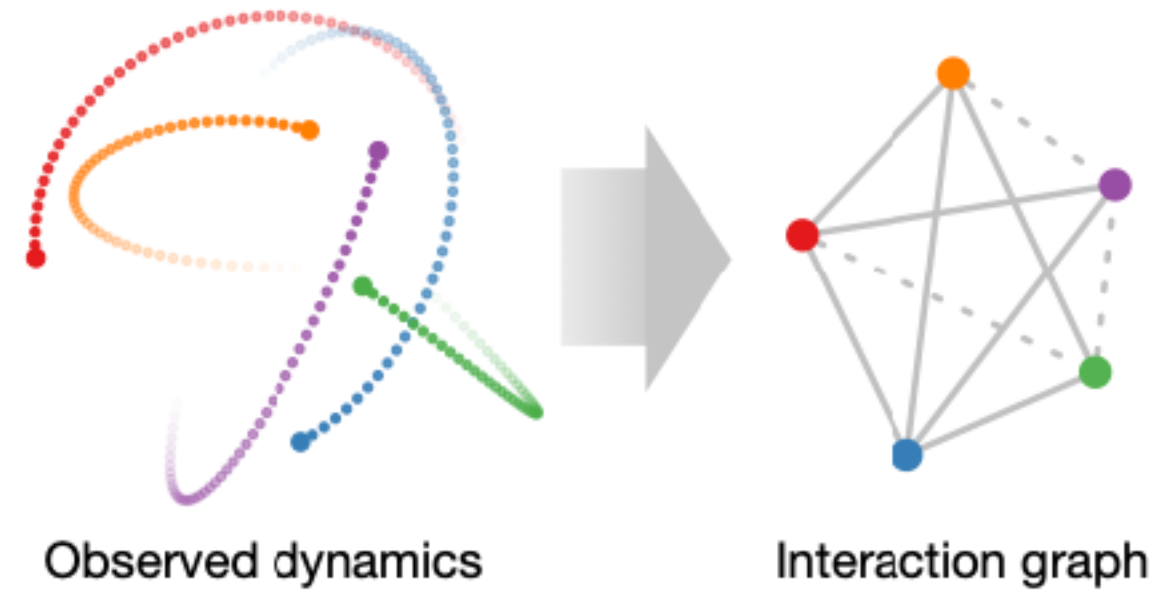


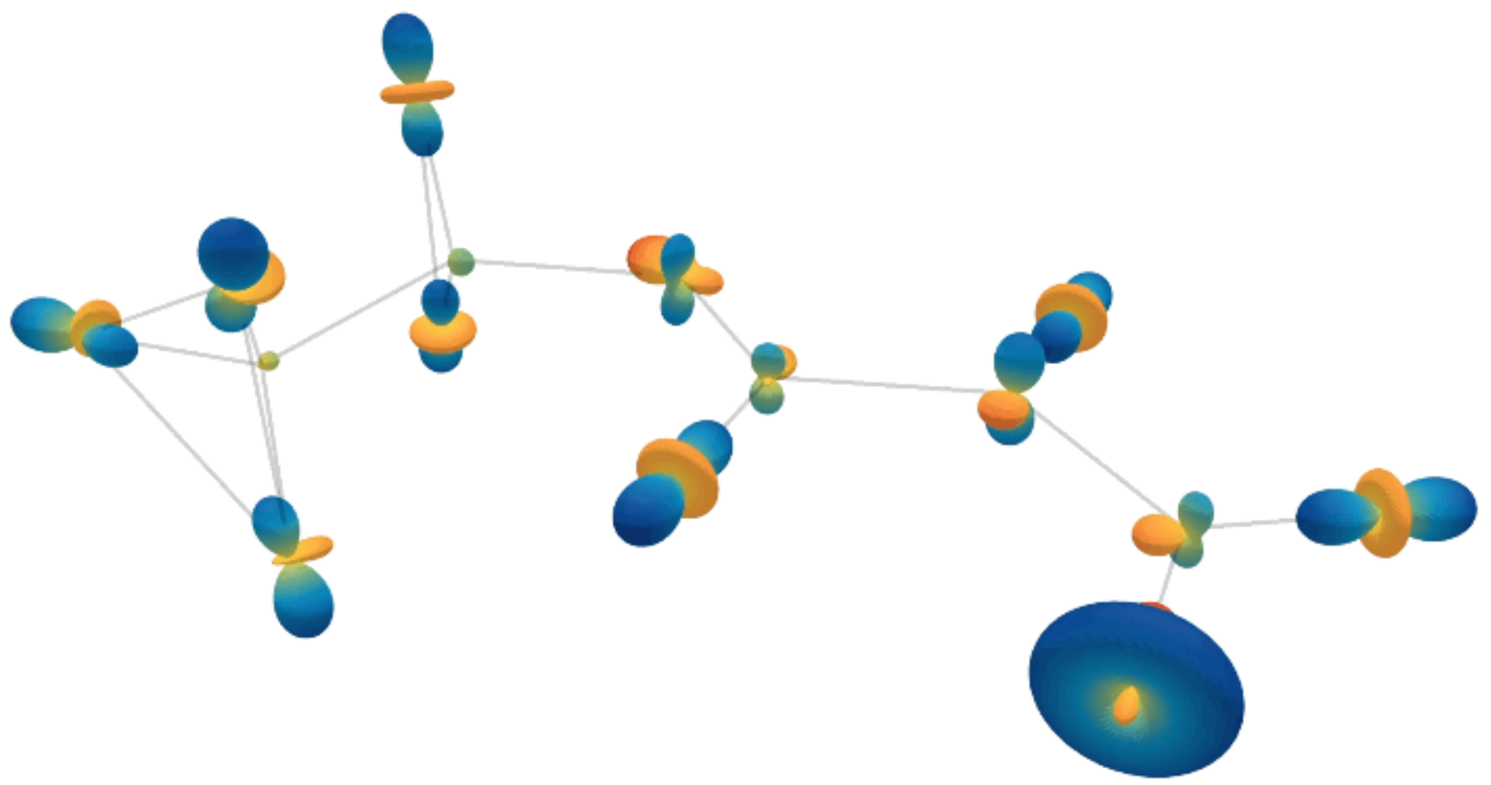
*Figure from Kipf et al 2018

	Method	MSE	Time [s]
G-CNNs	Linear	.0819	.0001
G-CNNs	SE(3)-Tr. (Fuchs et al., 2020)	.0244	.0742
G-CNNs	TFN (Thomas et al., 2018)	.0155	.0182
	NMP (Gilmer et al., 2017)	.0107	.0017
	Radial Field (Köhler et al., 2019)	.0104	.0019
Isotropic "non-linear CNNs"	EGNN (Satorras et al., 2021)	.0070± .00022	.0029
	$\text{SE}_{\text{linear}}$ $(l_f = 2, l_a = 2)$	.0116± .00021	.064
"non-linear G-CNNs"	$\text{SE}_{\text{non-linear}}$ $(l_f = 1, l_a = 1)$	.0060± .00019	.031
"non-linear G-CNNs"	SEGNN_{G} $(l_f = 1, l_a = 1)$	.0056± .00025	.025
"non-linear G-CNNs"	$\text{SEGNN}_{\text{G+P}}$ $(l_f = 1, l_a = 1)$	.0043± .00015	.026

G-CNNs outperform CNNs with isotropic kernels

"Non-linear convolutions" outperform linear convolutions

Task: Molecular property prediction



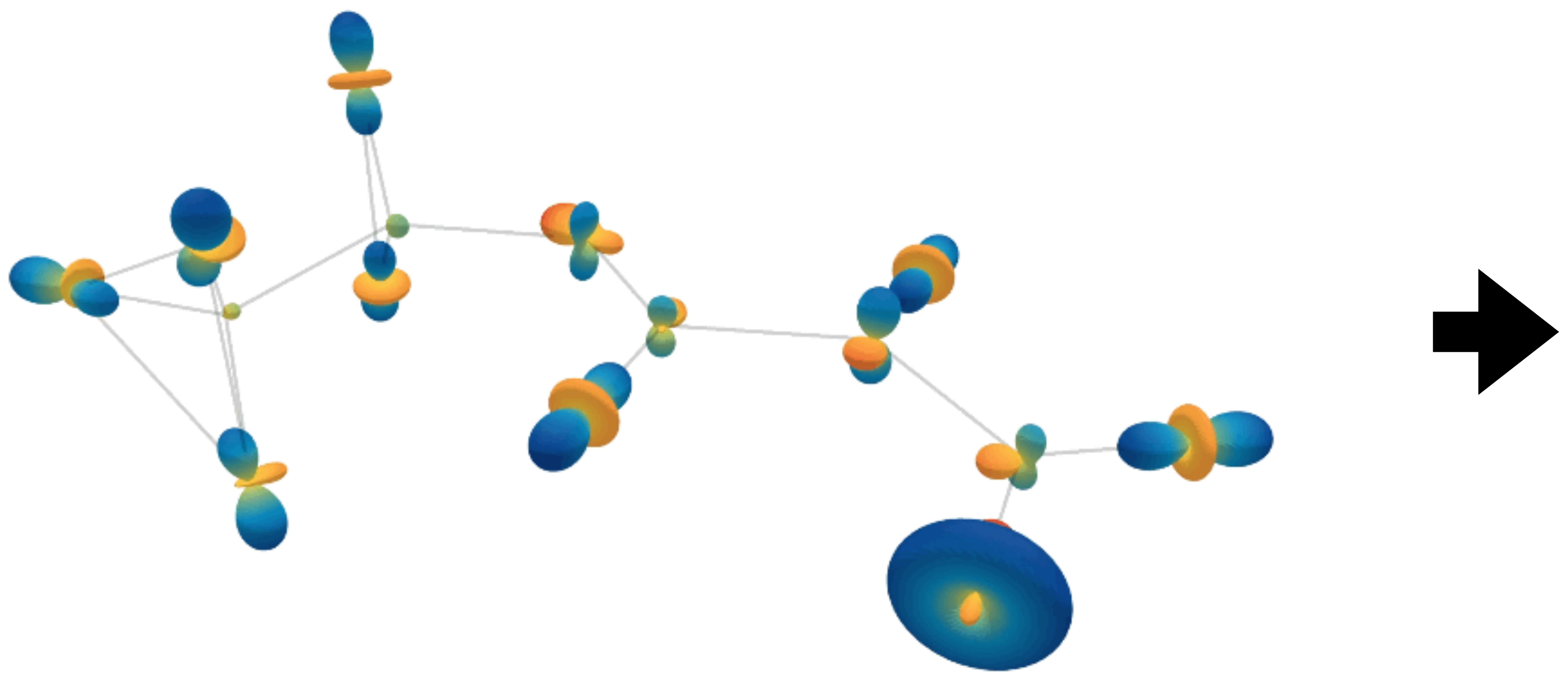
Property

	Task Units	Cutoff radius	α bohr^3	$\Delta\varepsilon$ meV	$\varepsilon_{\text{HOMO}}$ meV	$\varepsilon_{\text{LUMO}}$ meV	μ D	C_ν cal/mol K	Time [s]
Isotropic (fully connected graph)	(S)EGNN ($l_f = 0, l_a = 0$)	-	.091	53	34	28	.042	.043	0.016
Isotropic (local)	(S)EGNN ($l_f = 0, l_a = 0$)	2Å	.24	98	60	60	.34	.077	0.014
Anisotropic (local)	SEGNN ($l_f = 1, l_a = 2$)	2Å	.074	48	27	25	.031	.035	0.048
	SEGNN ($l_f = 2, l_a = 3$)	2Å	.060	42	24	21	.023	.031	0.097

(Steerable) G-CNNs allow for local connectivity (Scales to large proteins!!!)

isotropic convs require full connectivity in order to infer the geometry

Task: Molecular property prediction



Property

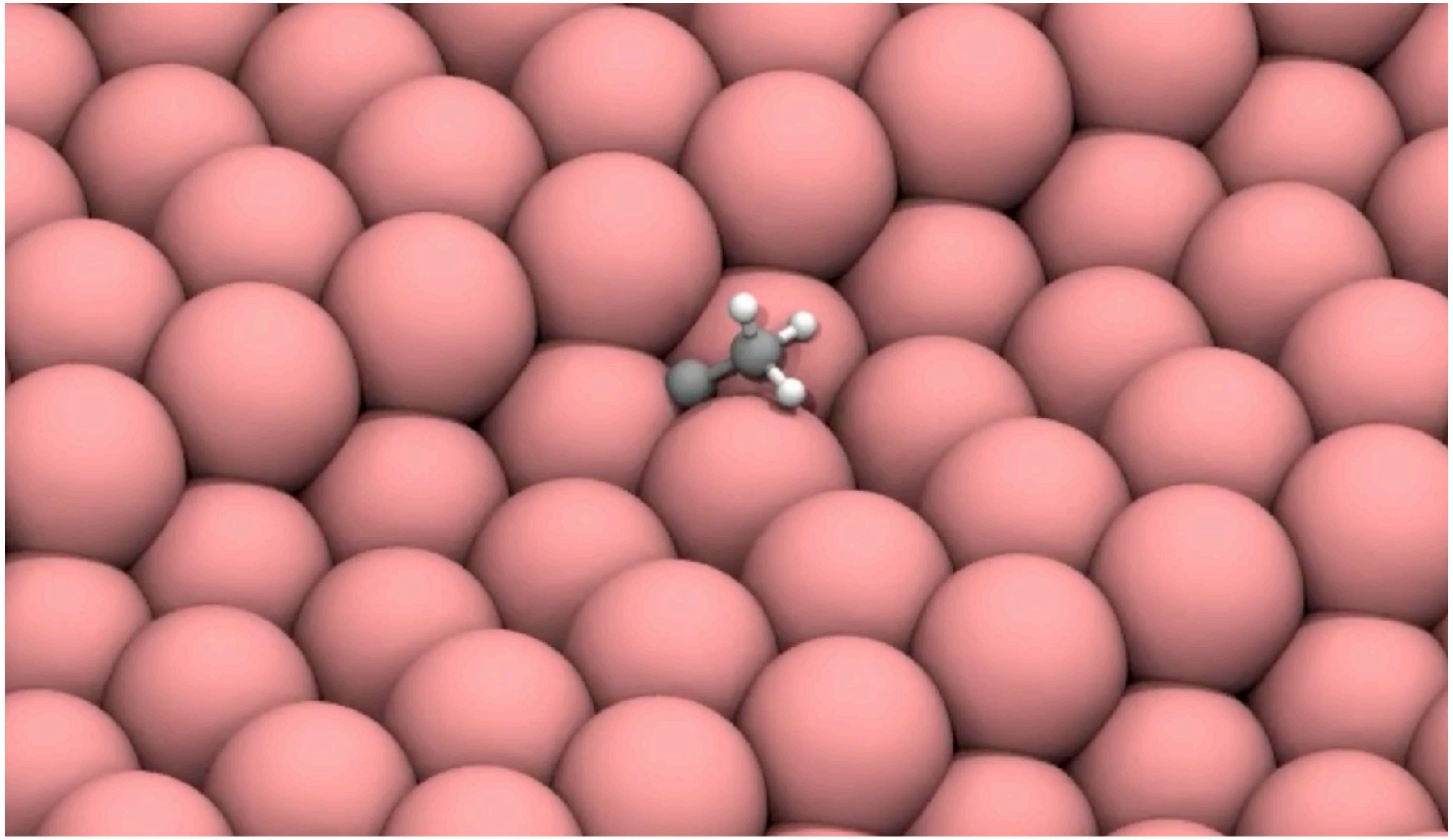
	Task Units	Cutoff radius	α bohr^3	$\Delta\varepsilon$ meV	$\varepsilon_{\text{HOMO}}$ meV	$\varepsilon_{\text{LUMO}}$ meV	μ D	C_ν cal/mol K	Time [s]
Isotropic (fully connected graph)	(S)EGNN ($l_f = 0, l_a = 0$)	-	.091	53	34	28	.042	.043	0.016
Isotropic (local)	(S)EGNN ($l_f = 0, l_a = 0$)	2Å	.24	98	60	60	.34	.077	0.014
Anisotropic (local)	SEGNN ($l_f = 1, l_a = 2$)	2Å	.074	48	27	25	.031	.035	0.048
	SEGNN ($l_f = 2, l_a = 3$)	2Å	.060	42	24	21	.023	.031	0.097

(Steerable) G-CNNs allow for local connectivity (Scales to large proteins!!!)

isotropic convs require full connectivity in order to infer the geometry

Steerable methods for computational chemistry

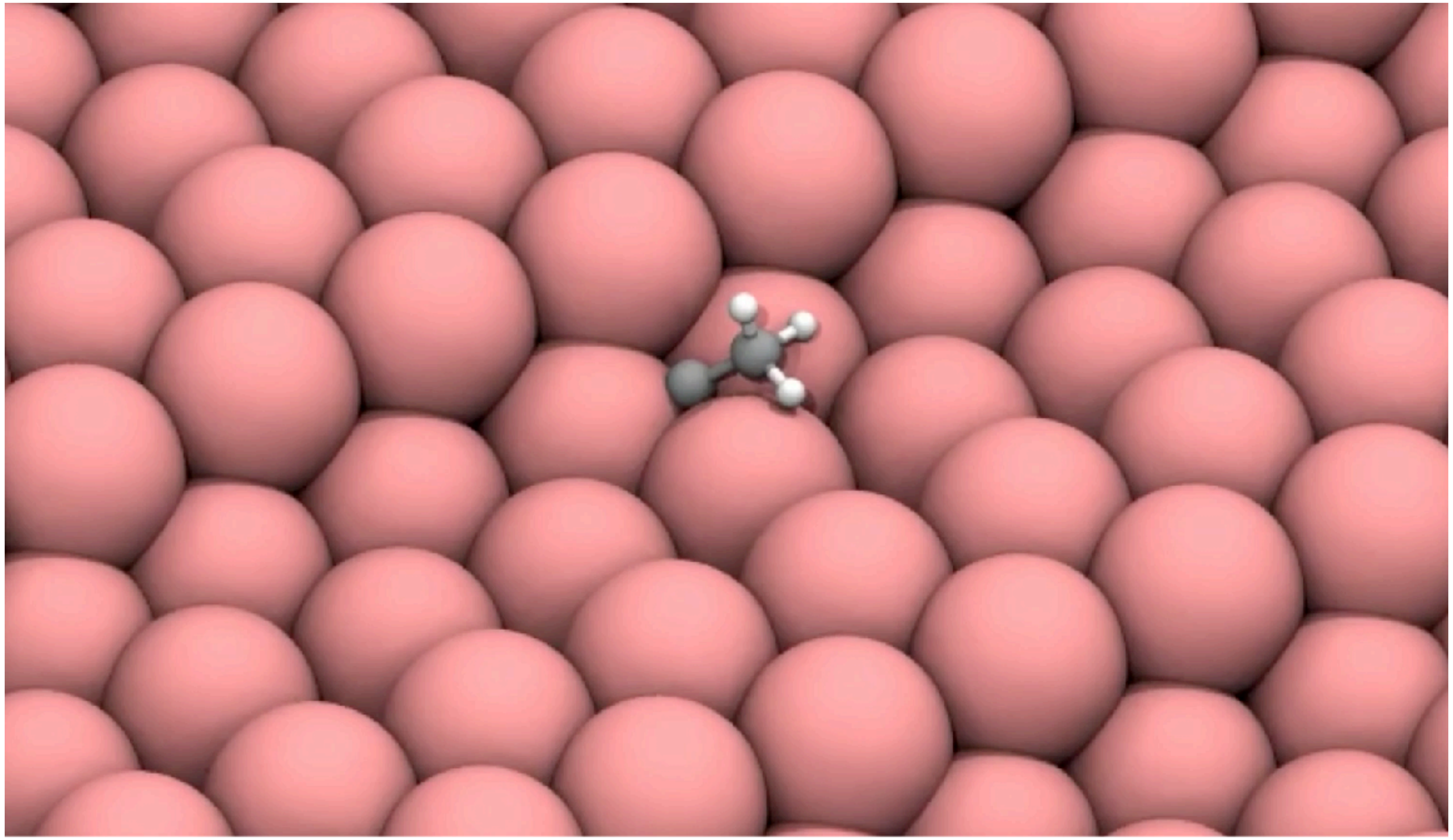
Brandstetter, Hesselink, van der Pol, Bekkers, Welling **Geometric and Physical Quantities Improve E(3) Equivariant Message Passing** - arXiv:2110.02905



Video: Open Catalyst Project

Steerable methods for computational chemistry

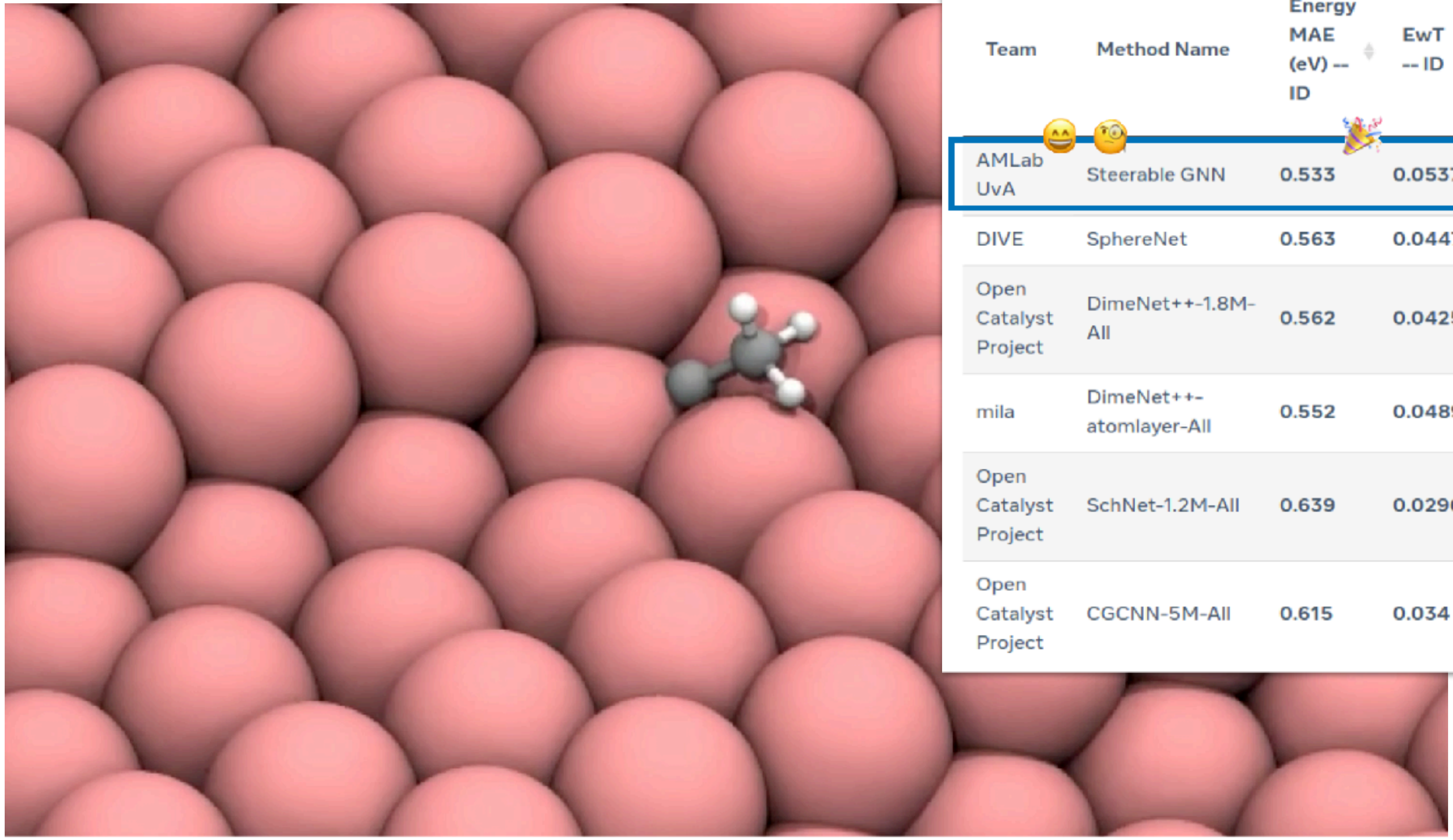
Brandstetter, Hesselink, van der Pol, Bekkers, Welling **Geometric and Physical Quantities Improve E(3) Equivariant Message Passing** - arXiv:2110.02905



Video: Open Catalyst Project

Steerable methods for computational chemistry

Brandstetter, Hesselink, van der Pol, Bekkers, Welling **Geometric and Physical Quantities Improve E(3) Equivariant Message Passing** - arXiv:2110.02905



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

Conclusion

But first… A final note on **representing functions with NN**

FlexConv: Continuous Kernel Convolutions with Differentiable Kernel Sizes

David W. Romero[*,1], **Robert-Jan Bruintjes**[*,2],
Erik J. Bekkers[3], **Jakub M. Tomczak**[1], **Mark Hoogendoorn**[1], **Jan C. van Gemert**[2]
[1] Vrije Universiteit Amsterdam [2] Delft University of Technology [3] University of Amsterdam
The Netherlands
d.w.romeroguzman@vu.nl, r.bruintjes@tudelft.nl

Abstract

When designing Convolutional Neural Networks (CNNs), one must select the size of the convolutional kernels before training. Recent works show CNNs benefit from different kernel sizes at different layers, but exploring all possible combinations is unfeasible in practice. A more efficient approach is to learn the kernel size during training. However, existing works that learn the kernel size have a limited bandwidth. These approaches scale kernels by dilation, and thus the detail they can describe is limited. In this work, we propose *FlexConv*, a novel convolutional operation with which high bandwidth convolutional kernels of learnable kernel size can be learned at a fixed parameter cost. *FlexNets* model long-term dependencies without the use of pooling, achieve state-of-the-art performance on several sequential datasets, outperform recent works with learned kernel sizes, and are competitive with much deeper ResNets on image benchmark datasets. Additionally, FlexNets can be deployed at higher resolutions than those seen during training. To avoid aliasing, we propose a novel kernel parameterization with which the frequency of the kernels can be analytically controlled. Our novel kernel parameterization shows higher descriptive power and faster convergence speed than existing parameterizations. This leads to important improvements in classification accuracy.

Multiplicative Filter Networks (MFNs). Recently, Fathony et al. (2021) proposed to construct implicit neural representations as the linear combination of exponentially many basis functions $\mathbf{g}$:

$$\mathbf{h}^{(1)} = \mathbf{g}\big([x,y];\boldsymbol{\theta}^{(1)}\big) \qquad \mathbf{g}:\mathbb{R}^2 \rightarrow \mathbb{R}^{\mathrm{N_{hid}}} \tag{2}$$

$$\mathbf{h}^{(l)} = \big(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\big)\cdot\mathbf{g}\big([x,y];\boldsymbol{\theta}^{(l)}\big) \qquad \mathbf{W}^{(l)} \in \mathbb{R}^{\mathrm{N_{hid}}\times\mathrm{N_{hid}}}, \mathbf{b}^{(l)} \in \mathbb{R}^{\mathrm{N_{hid}}} \tag{3}$$

$$\psi(x,y) = \mathbf{W}^{(\mathrm{L})}\mathbf{h}^{(\mathrm{L}-1)} + \mathbf{b}^{(\mathrm{L})} \qquad \mathbf{W}^{(\mathrm{L})} \in \mathbb{R}^{\mathrm{N}\times\mathrm{N_{hid}}}, \mathbf{b}^{(\mathrm{L})} \in \mathbb{R}^{\mathrm{N}} \tag{4}$$

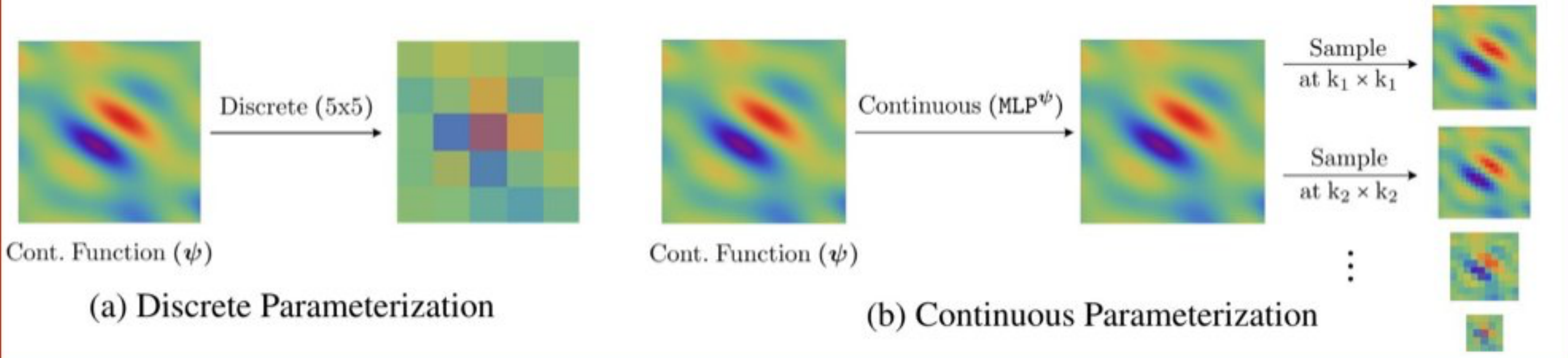
Maximum frequency of MAGNets. The maximum frequency component of a MAGNet is given by:

$$f^{+}_{\mathrm{MAGNet}} = \sum_{l=1}^{\mathrm{L}} \max_{i_l}\left(\left(\max_{j}\frac{\mathbf{W}^{(l)}_{\mathrm{g},i_l,j}}{2\pi}\right) + \frac{\sigma_{\mathrm{cut}}\min\{\gamma^{(l)}_{\mathrm{X},i_l},\gamma^{(l)}_{\mathrm{Y},i_l}\}}{2\pi}\right), \tag{21}$$

where L corresponds to the number of layers, $\mathbf{W}^{(l)}_{\mathrm{g}}, \gamma^{(l)}_{\mathrm{X}}, \gamma^{(l)}_{\mathrm{Y}}$ to the MAGNet parameters as defined in Eq. 8, and $\sigma_{\mathrm{cut}}=2\cdot\mathtt{stdev}$ to the cut-off frequency of the Gaussian envelopes in the Gabor filters. A formal treatment as well as the derivations can be found in Appx. A.1.

Aliasing regularization of FlexConv kernels. With the analytic derivation of $f^{+}_{\mathrm{FlexConv}}$ we penalize the generated kernels to have frequencies smaller or equal to their Nyquist frequency $f_{\mathrm{Nyq}}(k)$ via:

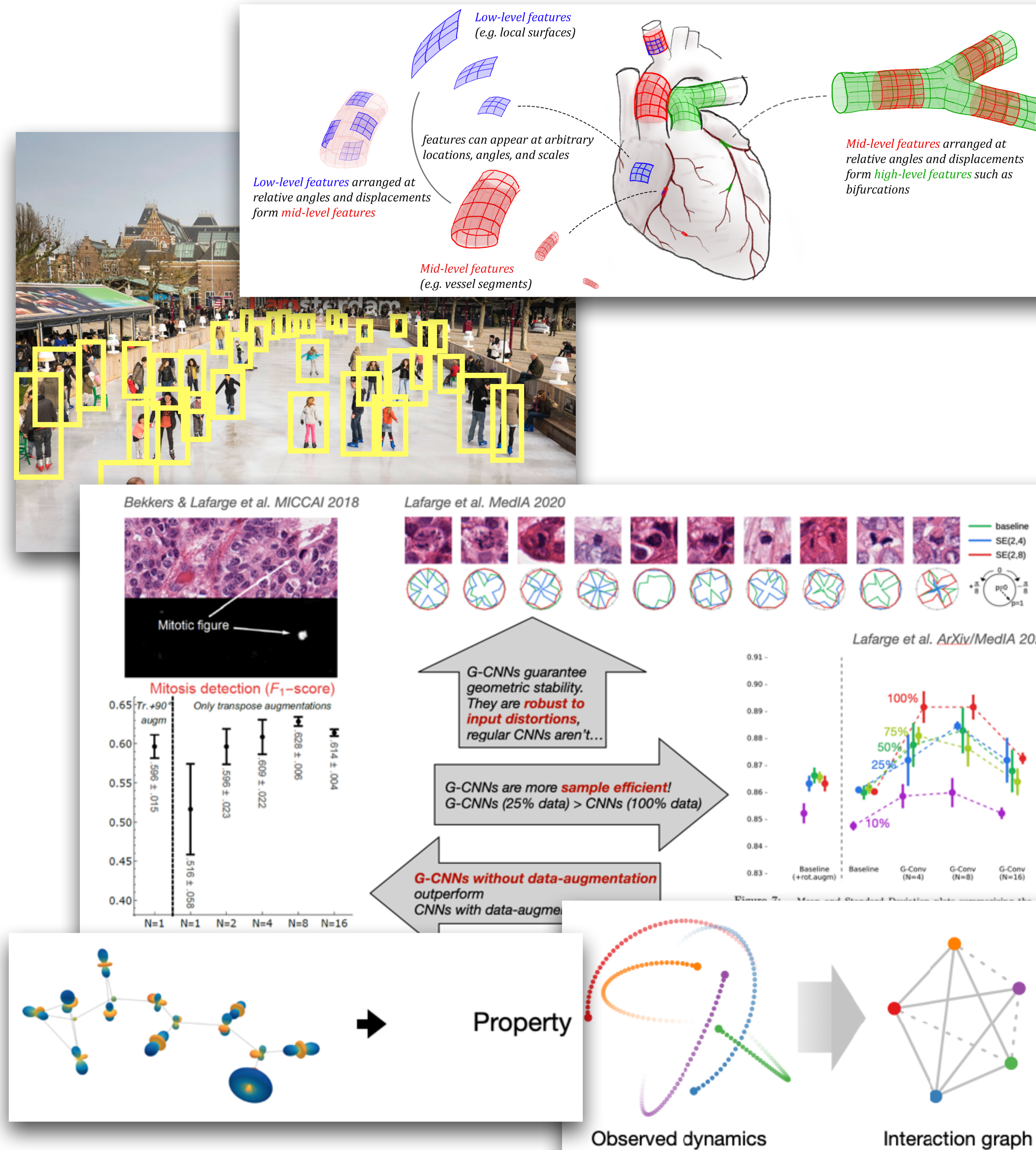
$$\mathcal{L}_{\mathrm{HF}} = \|\max\{f^{+}_{\mathrm{FlexConv}}, f_{\mathrm{Nyq}}(k)\} - f_{\mathrm{Nyq}}(k)\|^2, \quad \text{with} \quad f_{\mathrm{Nyq}}(k) = \tfrac{k-1}{4}. \tag{25}$$



(a) Discrete Parameterization (b) Continuous Parameterization

Conclusion

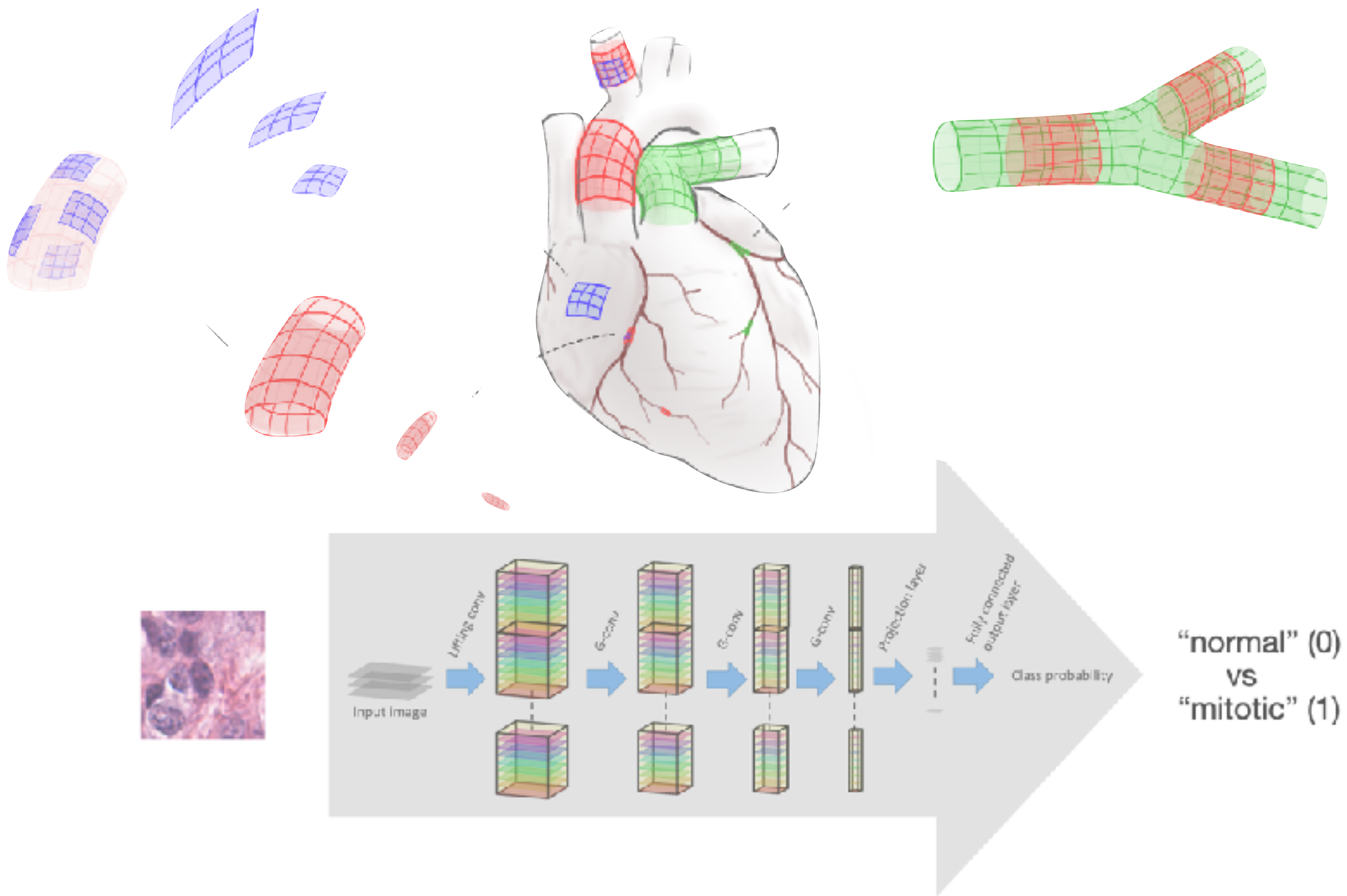
- **G-CNNs naturally arise** from NNs under equivariance constraints
- **G-CNNs improve upon classic CNNs** by
  - Making data augmentation w.r.t. the group obsolete
  - No valuable network capacity needs to be spend on dealing w geometry
  - The added geometric structure allows to deal with context (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



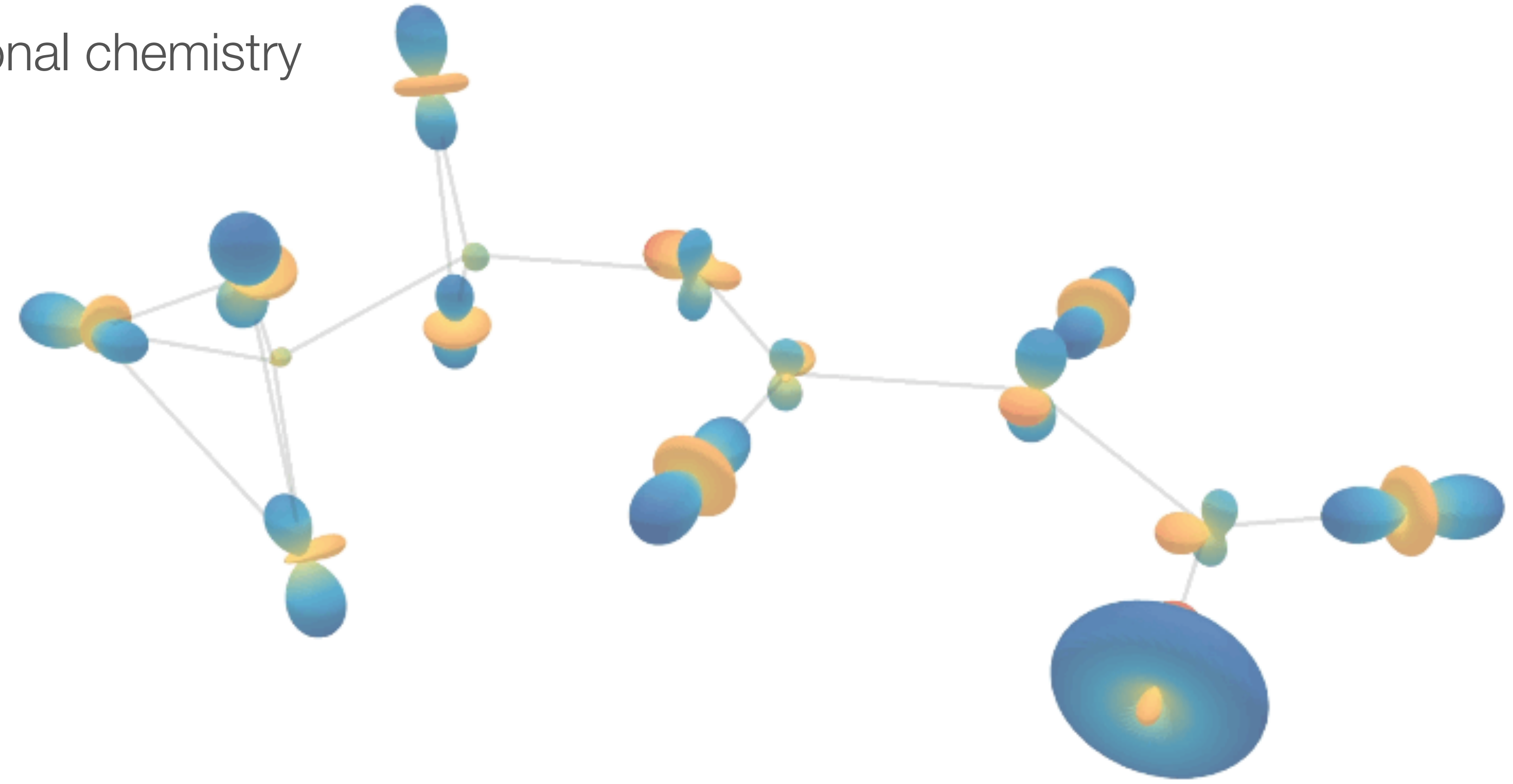
Group equivariant deep learning and non-linear convolutions

Erik J Bekkers, Amsterdam Machine Learning Lab (AMLAB Amsterdam Machine Learning Lab), University of Amsterdam
e.j.bekkers@uva.nl

Medical image analysis



Computational chemistry



Physics

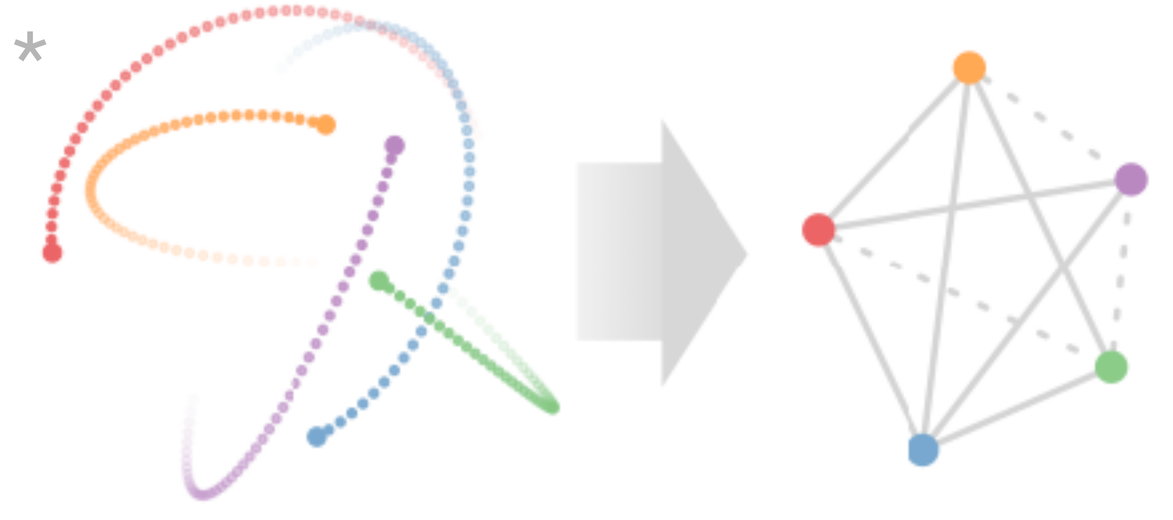


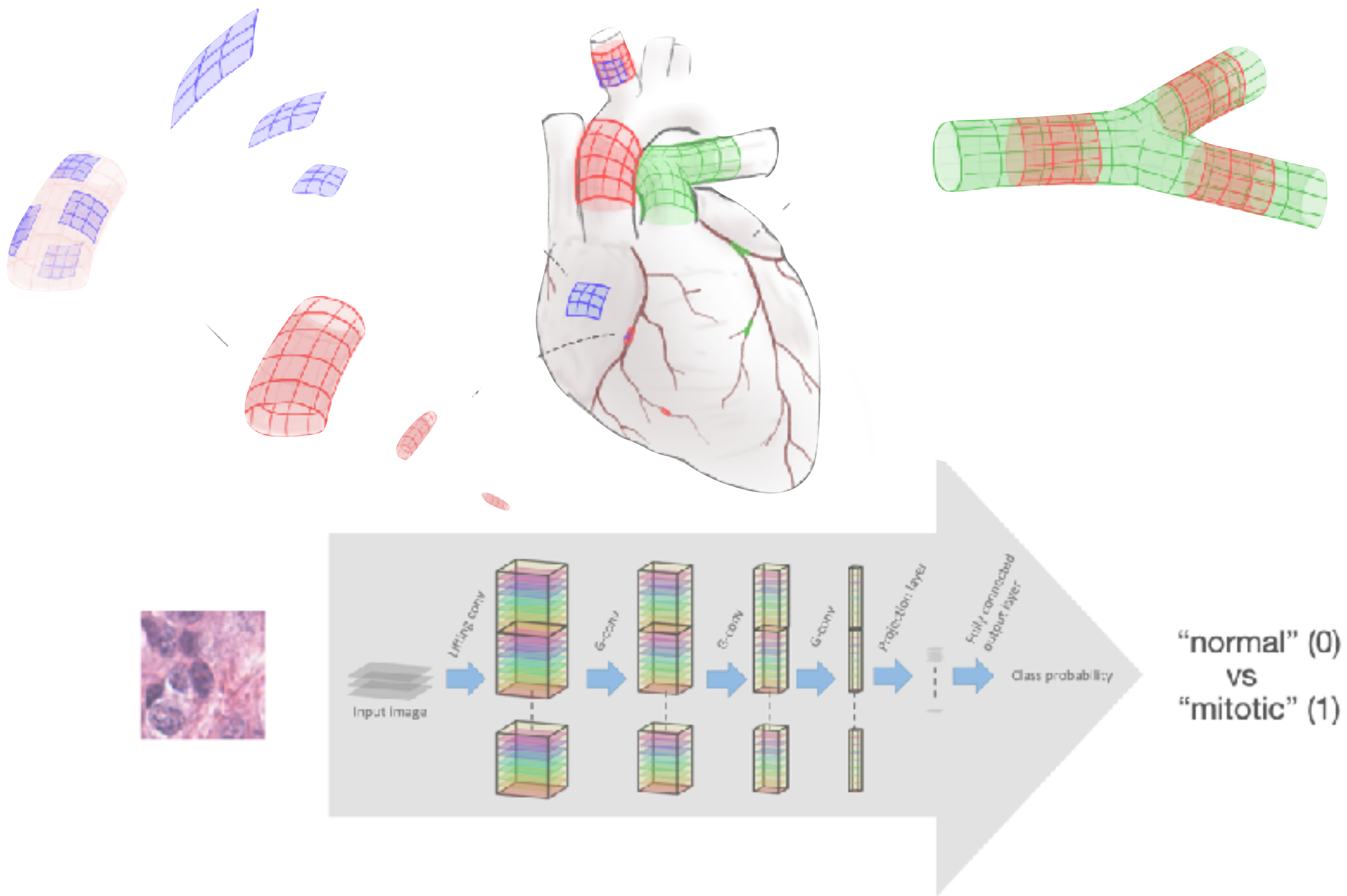
*Figure from Kipf et al 2018

UNIVERSITY OF AMSTERDAM

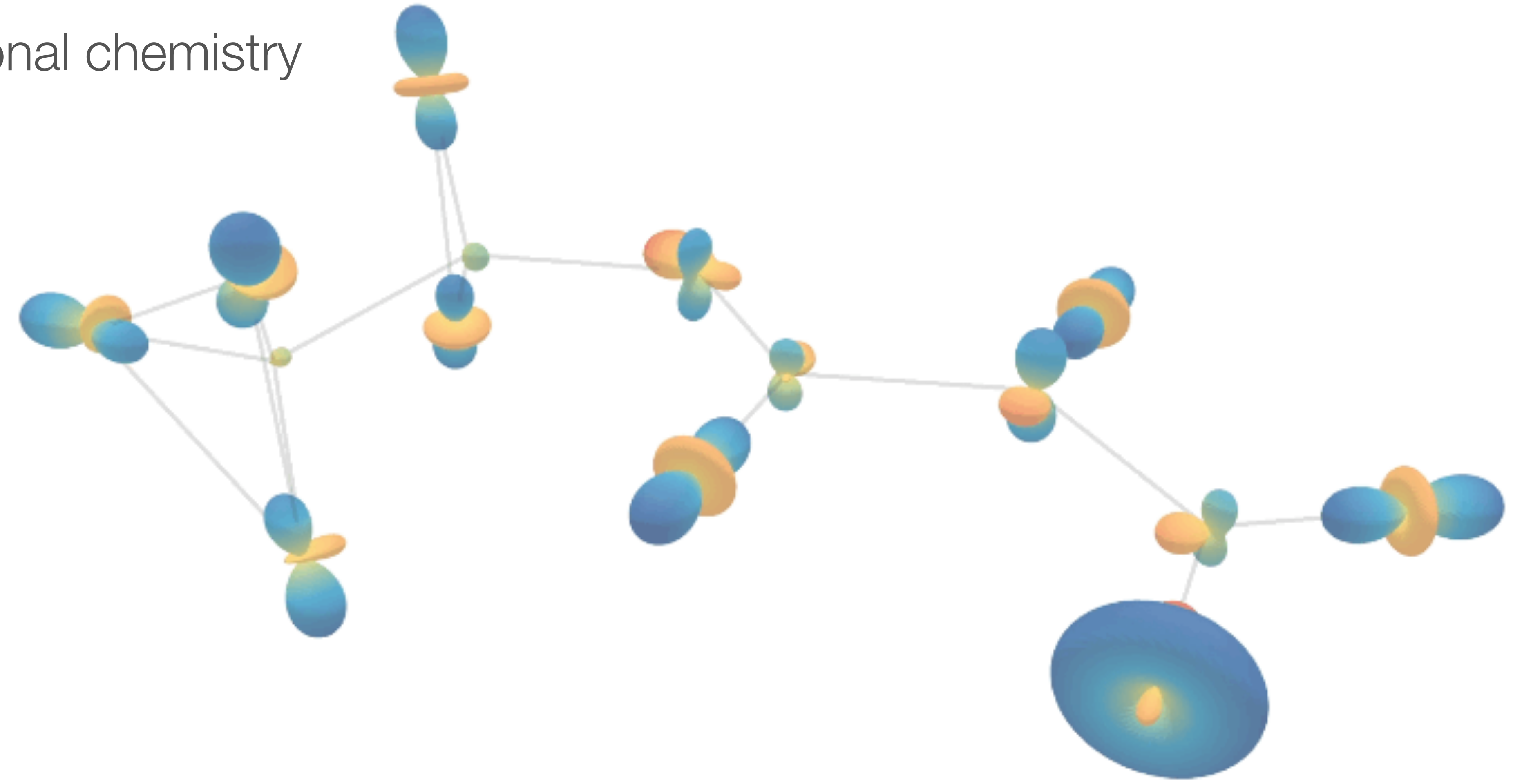
Group equivariant deep learning and non-linear convolutions

Erik J Bekkers, Amsterdam Machine Learning Lab (AMLAB), University of Amsterdam
e.j.bekkers@uva.nl

Medical image analysis



Computational chemistry



Physics

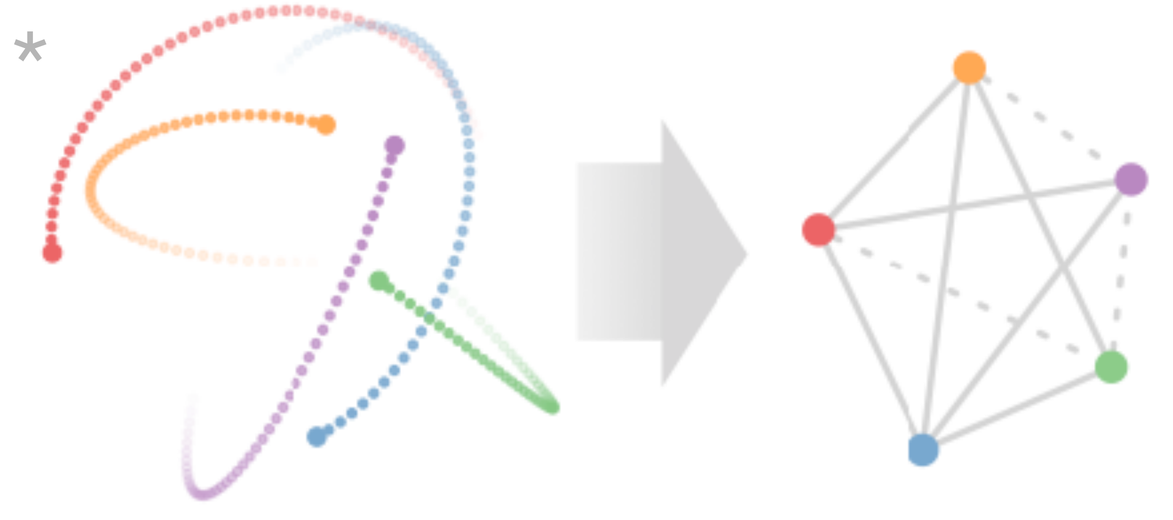


*Figure from Kipf et al 2018