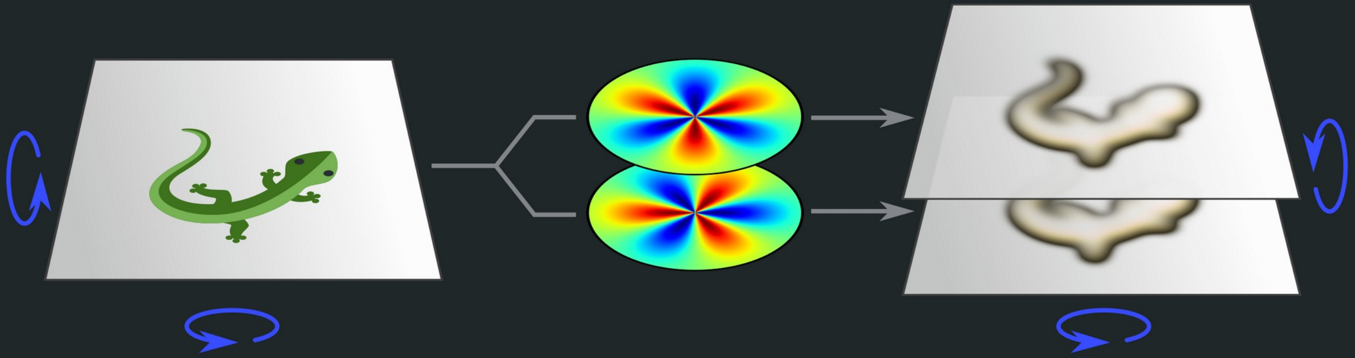
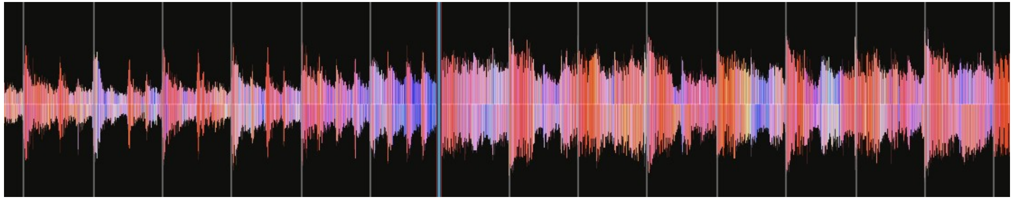
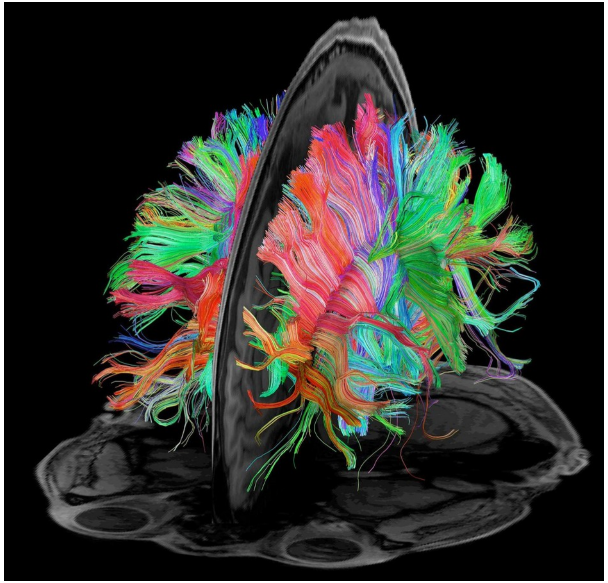
Group Equivariant Convolutional Networks on Euclidean spaces

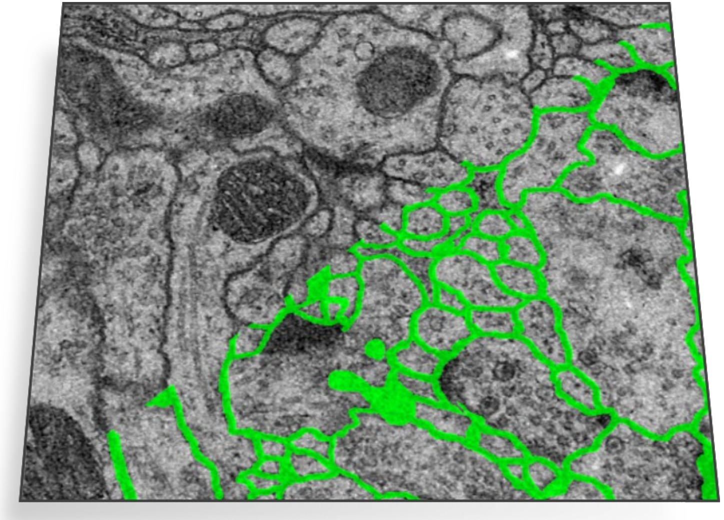
Maurice Weiler
AMLab, QUVA Lab
University of Amsterdam
@maurice_weiler



signals on Euclidean spaces







Outline

MLPs for image processing?

Translation equivariant CNNs on Euclidean spaces

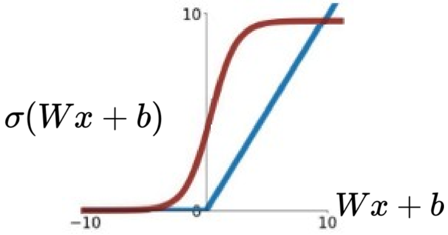
Affine equivariant steerable CNNs on Euclidean spaces

Group convolutions & convolutions on homogeneous spaces

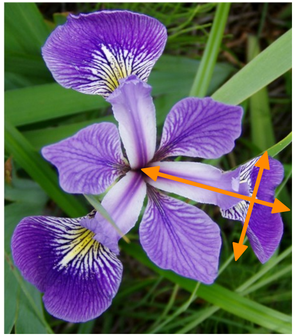
Multilayer Perceptrons (MLPs)

universal function approximators $f : \mathbb{R}^N \to \mathbb{R}^M$

composed of affine maps + nonlinearities: $x_{i+1} = \sigma(Wx_i + b)$



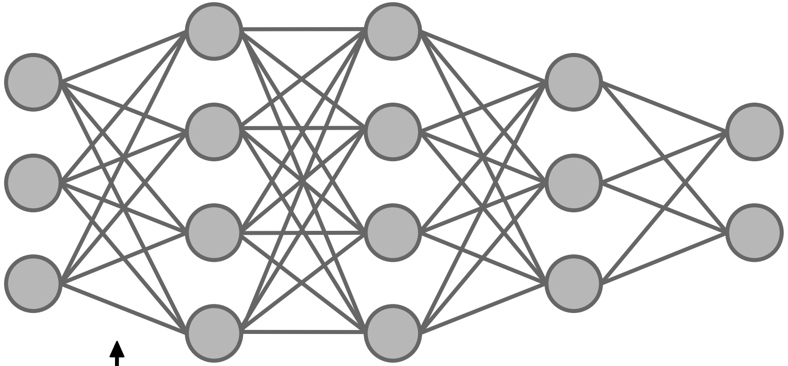
$$\mathbb{R}^3 \xrightarrow{f_1} \mathbb{R}^4 \xrightarrow{f_2} \mathbb{R}^4 \xrightarrow{f_3} \mathbb{R}^3 \xrightarrow{f_4} \mathbb{R}^2$$



length

width

thickness



p(species A | )

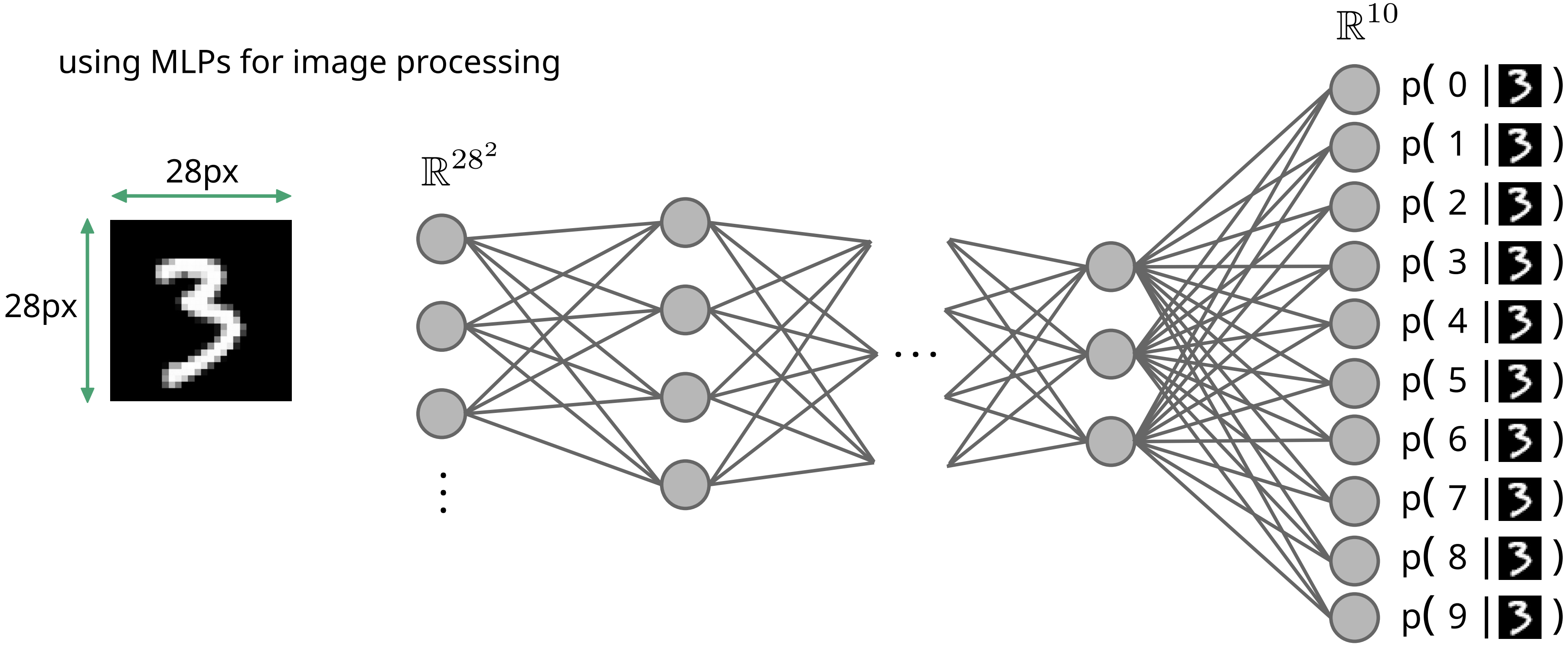
p(species B | )

$W \in \mathbb{R}^{4\times 3}$

$b \in \mathbb{R}^4$

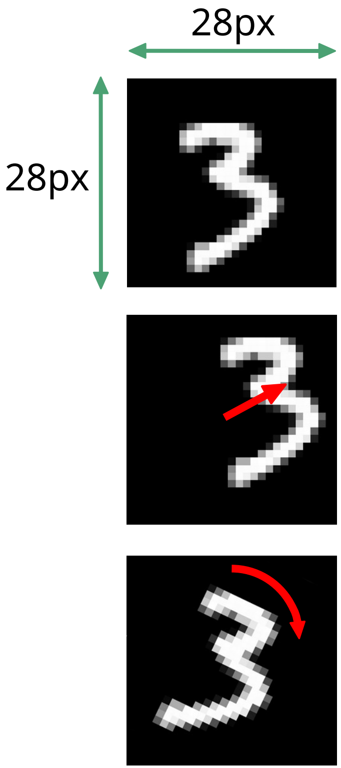
Multilayer Perceptrons (MLPs)

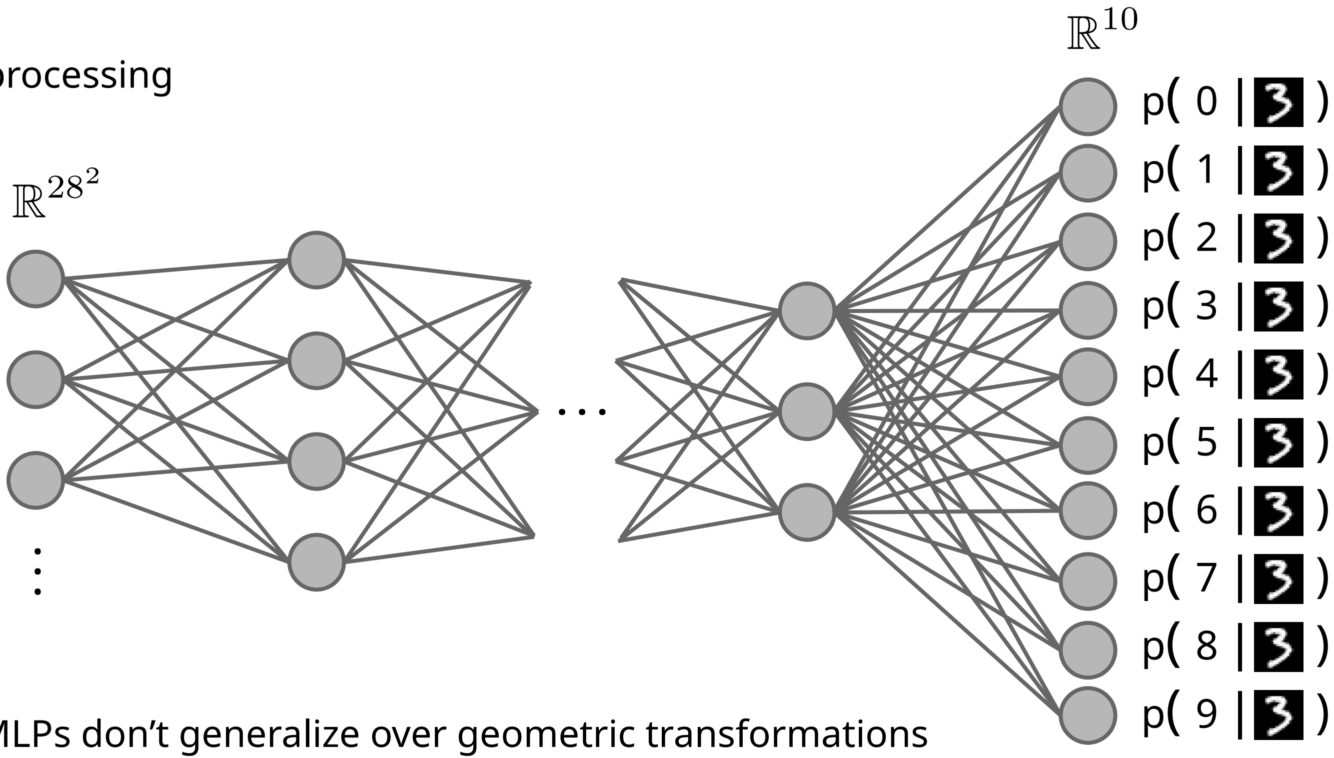
using MLPs for image processing



Multilayer Perceptrons (MLPs)

using MLPs for image processing

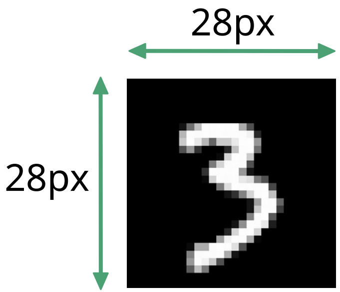


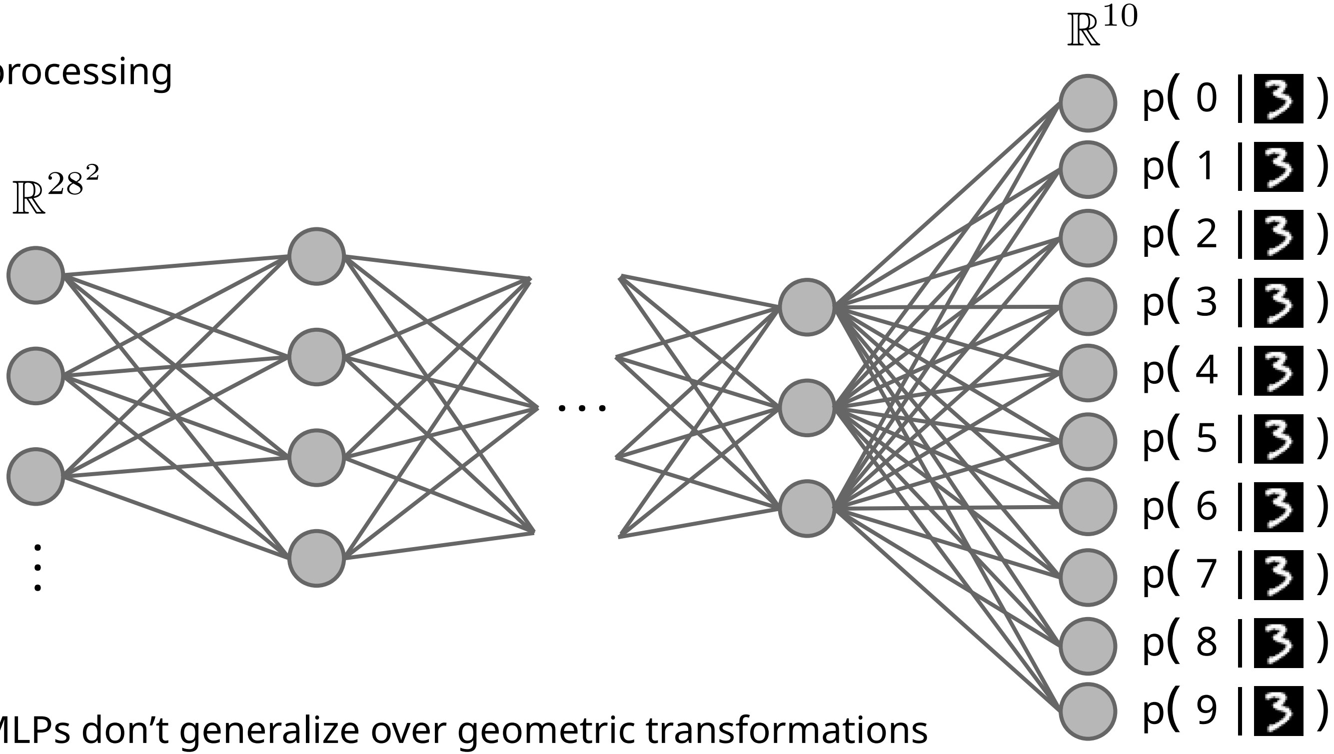


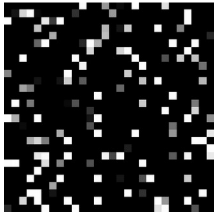
MLPs don't generalize over geometric transformations

Multilayer Perceptrons (MLPs)

using MLPs for image processing



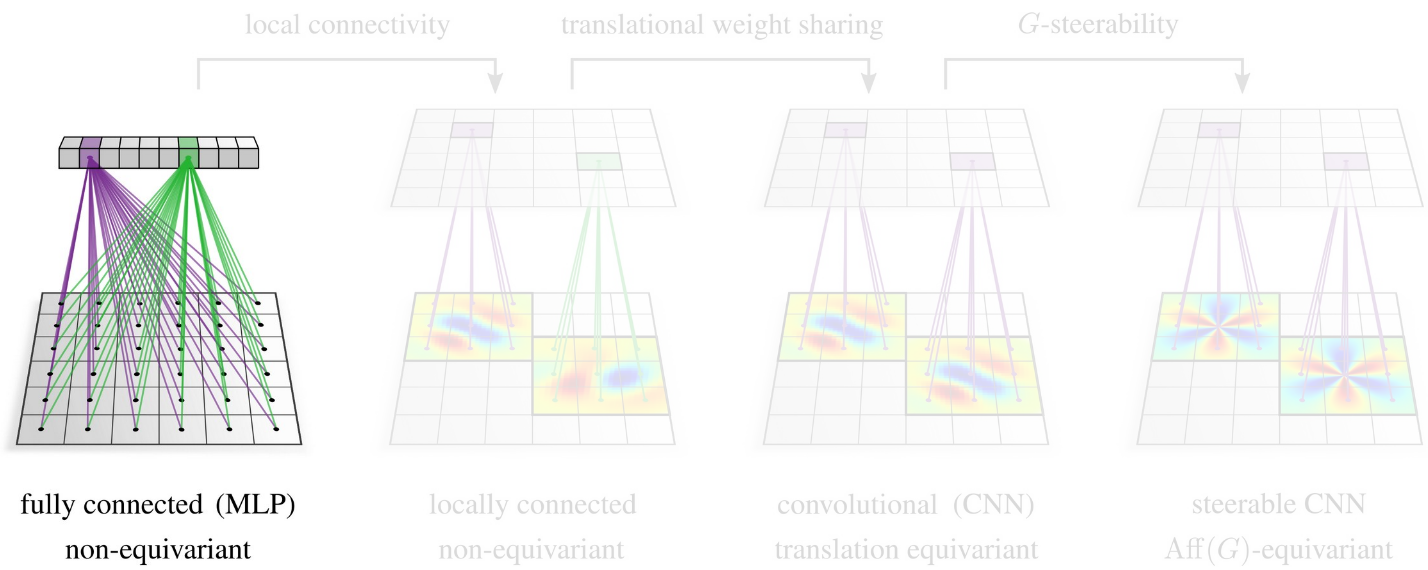




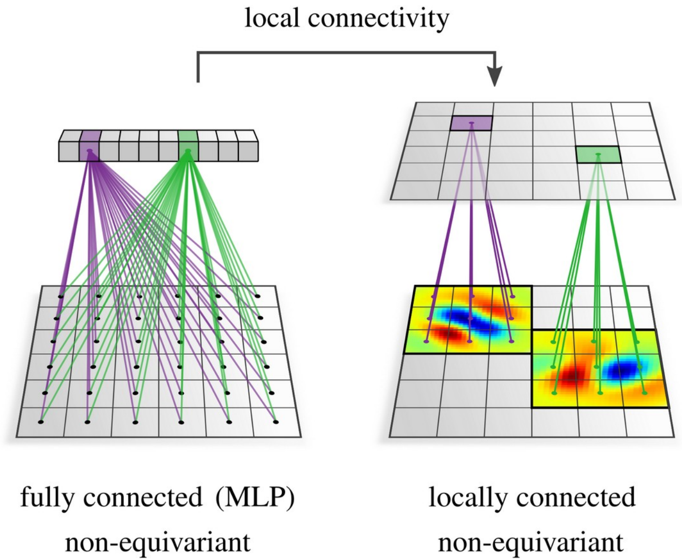
MLPs don't generalize over geometric transformations

MLPs are ignorant of the geometric arrangement of pixels
(any permutation of pixels would be equivalent)

convolutional networks == MLPs + equivariance

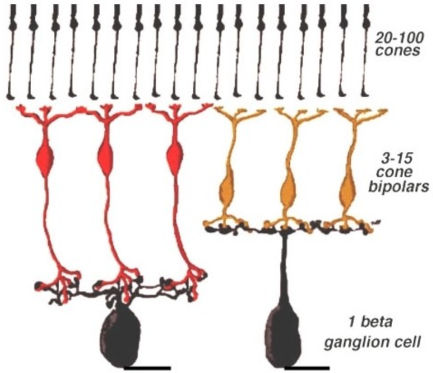


convolutional networks == MLPs + equivariance

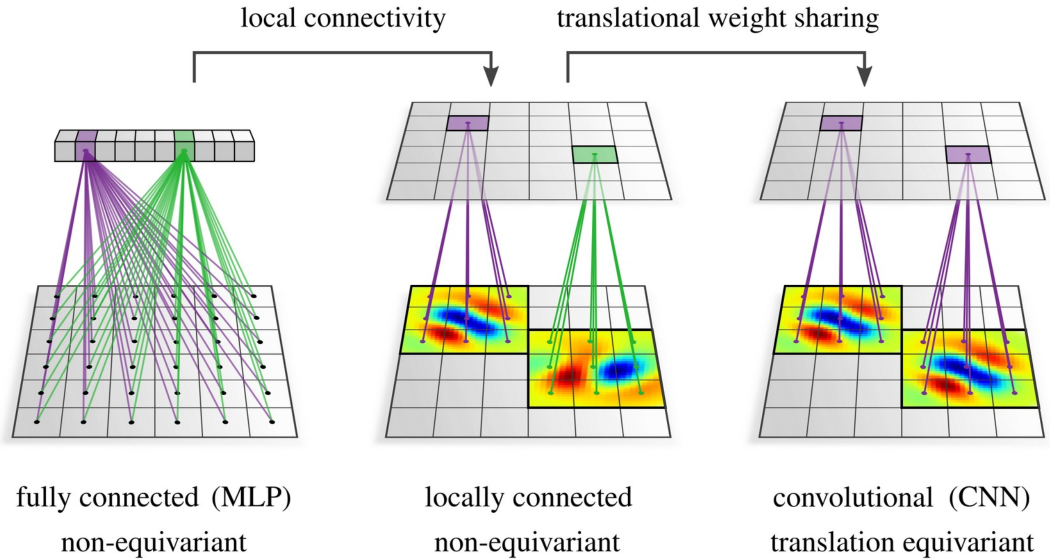


spatial encoding
in *feature maps*

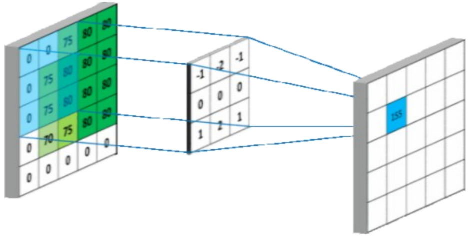
biological retina



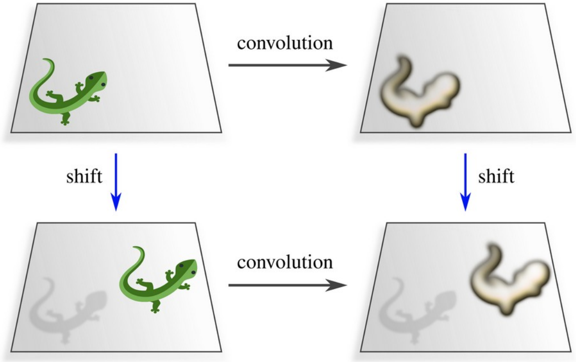
convolutional networks == MLPs + equivariance



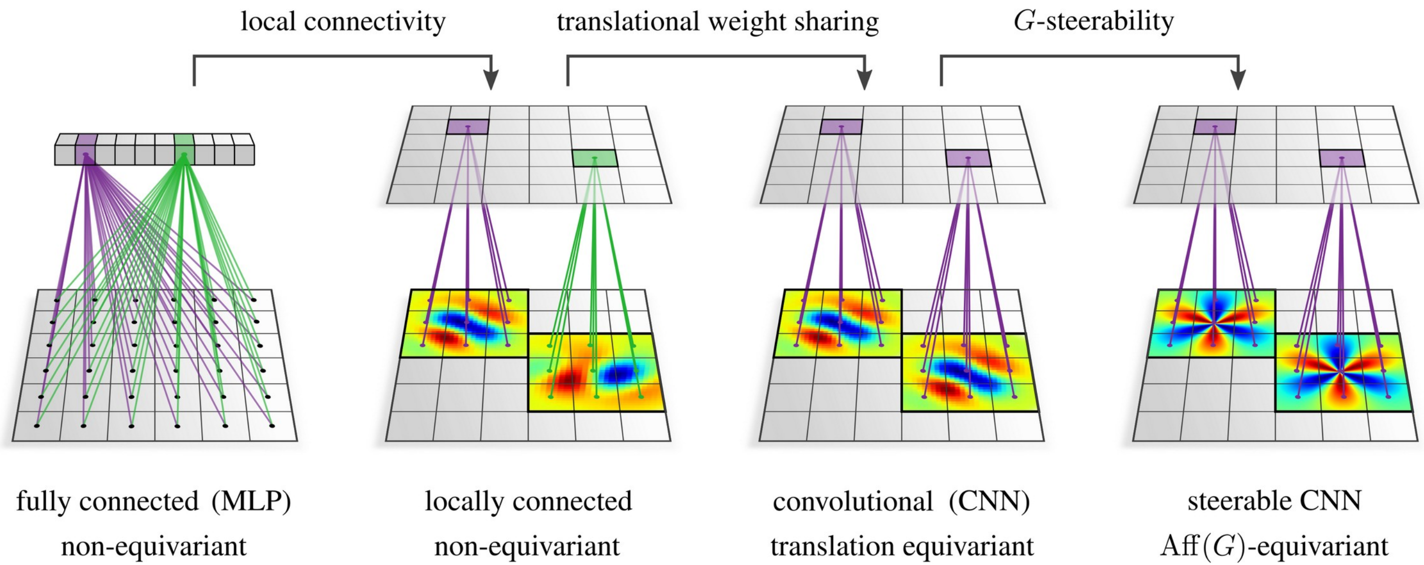
convolution



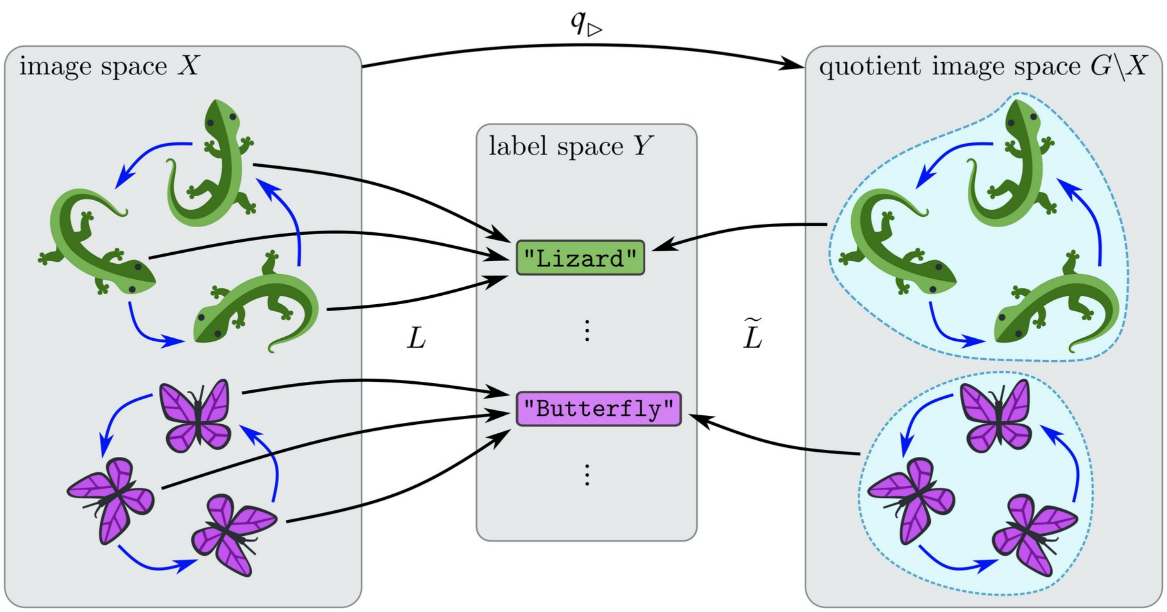
translation equivariance



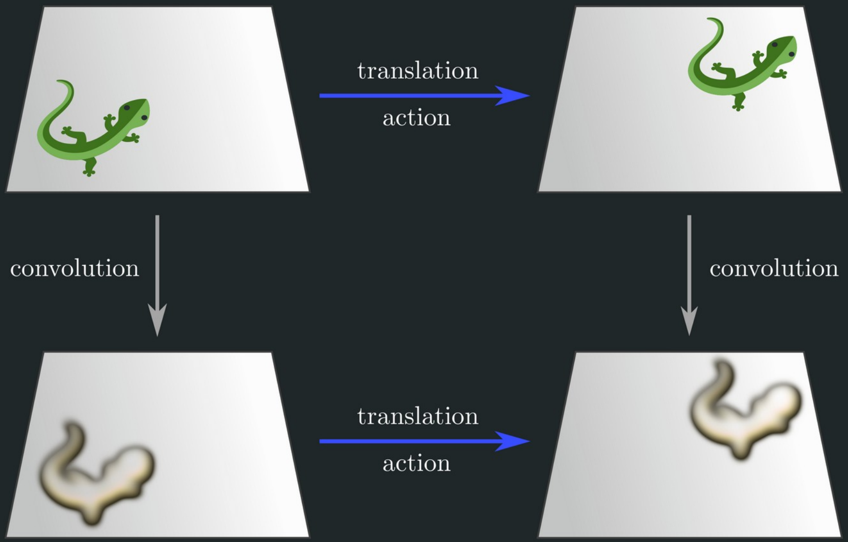
convolutional networks == MLPs + equivariance



G-equivariant models generalize over *G-orbits*



Translation equivariant CNNs on Euclidean spaces



Equivariant Neural Networks

(feed forward) neural networks are sequences of layers:

$$\mathcal{F}_0 \xrightarrow{\ L_1\ } \mathcal{F}_1 \xrightarrow{\ L_2\ } \mathcal{F}_2 \xrightarrow{\ L_3\ } \dots \xrightarrow{\ L_{N-1}\ } \mathcal{F}_{N-1} \xrightarrow{\ L_N\ } \mathcal{F}_N$$

equivariant NNs are sequences of equivariant layers:

$$\begin{tikzcd}
\mathcal{F}_0 \arrow[r, "L_1"] \arrow[d, "g \rhd_0"'] & \mathcal{F}_1 \arrow[r, "L_2"] \arrow[d, "g \rhd_1"'] & \mathcal{F}_2 \arrow[r, "L_3"] \arrow[d, "g \rhd_2"'] & \dots \arrow[r, "L_{N-1}"] & \mathcal{F}_{N-1} \arrow[r, "L_N"] \arrow[d, "g \rhd_{N-1}"'] & \mathcal{F}_N \arrow[d, "g \rhd_N"] \\
\mathcal{F}_0 \arrow[r, "L_1"'] & \mathcal{F}_1 \arrow[r, "L_2"'] & \mathcal{F}_2 \arrow[r, "L_3"'] & \dots \arrow[r, "L_{N-1}"'] & \mathcal{F}_{N-1} \arrow[r, "L_N"'] & \mathcal{F}_N
\end{tikzcd}$$

to design an equivariant network, we need to ...

... specify the *feature spaces* and *group actions* on them $\longrightarrow$ feature maps with translation action

... design *equivariant layers,* which commute with the group actions $\longrightarrow$ convolutions, bias summation, nonlinearities, etc.

Feature maps

discretized feature maps on $\mathbb{R}^d$ are implemented as "tensors" of shape $(\overbrace{X_1, \dots, X_d}^{\text{spatial / pixel dimensions}},\ \overset{\text{feature channels}}{C})$

continuous feature maps are functions $f : \mathbb{R}^d \to \mathbb{R}^c$ that assign feature vectors $f(x) \in \mathbb{R}^c$ to points $x \in \mathbb{R}^d$

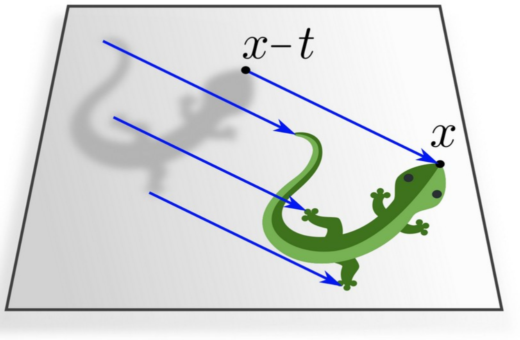
$\rotatebox[origin=c]{90}{$\in$}$

$L^2(\mathbb{R}^d, \mathbb{R}^c)$ = *vector space* of feature maps

linear

feature maps carry a translation group action $\triangleright : (\mathbb{R}^d, +) \times L^2(\mathbb{R}^d, \mathbb{R}^c) \to L^2(\mathbb{R}^d, \mathbb{R}^c)$

defined by $\big[t \triangleright f\big](x) := f(x - t)$



feature maps form a $(\mathbb{R}^d, +)$-representation,

known as *regular representation* $\big(\triangleright,\ L^2(\mathbb{R}^d, \mathbb{R}^c)\big)$

Translation equivariant NNs

translation equivariant networks consist of layers $\mathcal{L} : L^2(\mathbb{R}^d, \mathbb{R}^{c_{\text{in}}}) \to L^2(\mathbb{R}^d, \mathbb{R}^{c_{\text{out}}})$ that ...

... map between c_{in} and c_{out}-dimensional input and output feature maps

... commute with the group action: $\mathcal{L}\big[t \rhd f\big](x) = \big[t \rhd \mathcal{L}[f]\big](x) \quad \forall\ t \in (\mathbb{R}^d, +),\ x \in \mathbb{R}^d$

$$\begin{array}{ccc}
L^2\left(\mathbb{R}^d, \mathbb{R}^{c_{\text{in}}}\right) & \xrightarrow{\mathcal{L}} & L^2\left(\mathbb{R}^d, \mathbb{R}^{c_{\text{out}}}\right) \\
{\scriptstyle t \rhd_{\text{in}}} \downarrow & & \downarrow {\scriptstyle t \rhd_{\text{out}}} \\
L^2\left(\mathbb{R}^d, \mathbb{R}^{c_{\text{in}}}\right) & \xrightarrow[\mathcal{L}]{} & L^2\left(\mathbb{R}^d, \mathbb{R}^{c_{\text{out}}}\right)
\end{array}$$

Linear equivariant maps $\Leftrightarrow$ convolutions

ansatz for linear map:

generic integral transform $\quad \mathtt{I}_\kappa : L^2(\mathbb{R}^d, \mathbb{R}^{c_{\text{in}}}) \to L^2(\mathbb{R}^d, \mathbb{R}^{c_{\text{out}}})$

parameterized by 2-point correlator $\quad \kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{c_{\text{out}} \times c_{\text{in}}}$

and defined by $\quad \mathtt{I}_\kappa\big[f\big](x) := \int_{\mathbb{R}^d} dy\ \kappa(x,y)\, f(y)$

intuition:
matrix multiplication

$$(Mv)_x = \sum_y M_{xy}\, v_y$$

Linear equivariant maps $\Leftrightarrow$ convolutions

ansatz for linear map:

generic integral transform $\mathtt{I}_\kappa : L^2(\mathbb{R}^d, \mathbb{R}^{c_\text{in}}) \to L^2(\mathbb{R}^d, \mathbb{R}^{c_\text{out}})$

parameterized by 2-point correlator $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{c_\text{out} \times c_\text{in}}$

and defined by $\mathtt{I}_\kappa \big[f\big](x) := \int_{\mathbb{R}^d} dy\ \kappa(x,y)\, f(y)$

now demand equivariance:

Theorem (Regular translation intertwiners are convolutions). *The integral transform $\mathtt{I}_\kappa$ is translation equivariant iff the correlator depends only on* relative distances, *i.e. satisfies*

$$\kappa(x+t,\, y+t) = \kappa(x,y) \quad \text{for any } \ x,y,t \in \mathbb{R}^d$$

One can always choose $t=-y$ to obtain $\kappa(x,y) = \kappa(x-y,0) =: K(x-y)$, where we defined the matrix valued kernel

$$K : \mathbb{R}^d \to \mathbb{R}^{c_\text{out} \times c_\text{in}}, \quad x \mapsto K(x) := \kappa(x,0)\,.$$

The integral transform is therefore equivalent to a convolution *integral*

$$\mathtt{I}_\kappa \big[f\big](x) \ = \ \big[K * f\big](x) \ = \ \int_{\mathbb{R}^d} dy\, K(x-y)\, f(y)\,.$$

Linear equivariant maps $\Leftrightarrow$ convolutions

ansatz for linea

key takeaways:

translation equivariant linear maps = convolutions (assuming regular reps)

convolutional weight sharing = translation relativity

now demand equivariance:

Theorem (Regular translation intertwiners are convolutions). *The integral transform* $\mathtt{I}_\kappa$ *is translation equivariant iff the correlator depends only on* relative distances*, i.e. satisfies*

$$\kappa(x+t,\, y+t) = \kappa(x,y) \quad \text{for any } \; x,y,t \in \mathbb{R}^d$$

One can always choose $t=-y$ *to obtain* $\kappa(x,y) = \kappa(x-y,0) =: K(x-y)$*, where we defined the* matrix valued kernel

$$K:\mathbb{R}^d \to \mathbb{R}^{c_{\text{out}}\times c_{\text{in}}}, \quad x \mapsto K(x) := \kappa(x,0)\,.$$

The integral transform is therefore equivalent to a convolution *integral*

$$\mathtt{I}_\kappa\big[f\big](x) \;=\; \big[K*f\big](x) \;=\; \int_{\mathbb{R}^d} dy\, K(x-y)\, f(y)\,.$$

Translation equivariant bias summation

consider a general bias summation operation $B_{\mathfrak{b}}:\ L^2(\mathbb{R}^d,\mathbb{R}^c) \to L^2(\mathbb{R}^d,\mathbb{R}^c), \quad f \mapsto f + \mathfrak{b}$

parameterized by a **bias field** $\mathfrak{b}:\mathbb{R}^d \to \mathbb{R}^c$ $\implies$ allows to sum a *different bias* $\mathfrak{b}(x) \in \mathbb{R}^c$ at each $x \in \mathbb{R}^d$

Theorem (Translation equivariant bias summation). *The bias field summation is translation equivariant iff the bias field is* spatially constant*:*

$$\mathfrak{b}(x) = b \quad \text{for some} \quad b \in \mathbb{R}^c$$

similar *spatial invariance* results hold for other operations like nonlinearities, pooling, ...

Translation equivariant CNNs - Summary

we defined **feature vector spaces** as spaces of feature maps

we defined a (linear) **translation group action** on feature maps

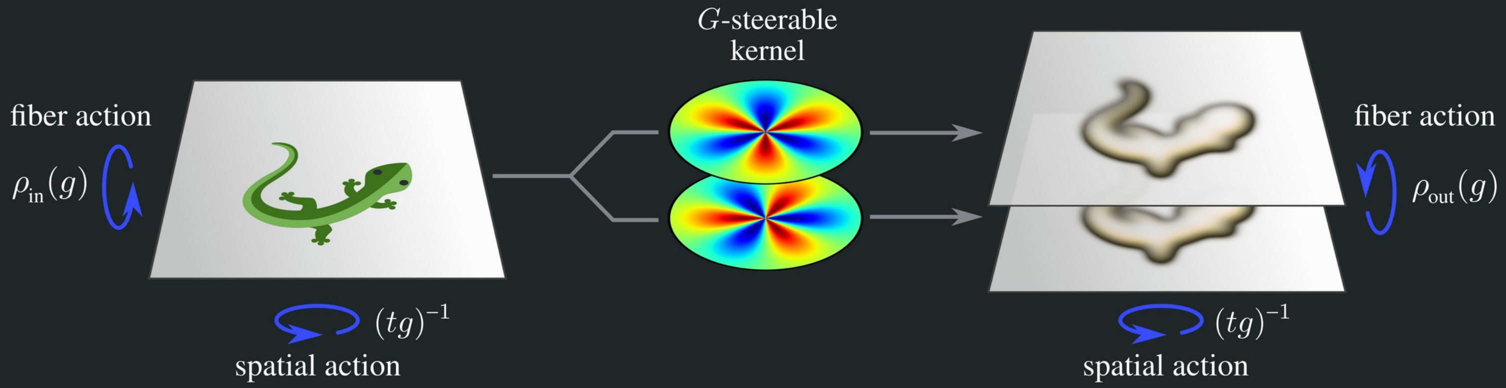
(regular) **translation group representation**

we derived **CNN operations** like convolutions / bias summation / etc by:

1) asuming a flexible **ansatz** (linear map, bias field summation)
2) demanding **translation equivariance** $\longrightarrow$ resulting in **spatial invariance / relativity / weight sharing**

next we do the same with more general symmetries of Euclidean space

Affine equivariant steerable CNNs on Euclidean spaces



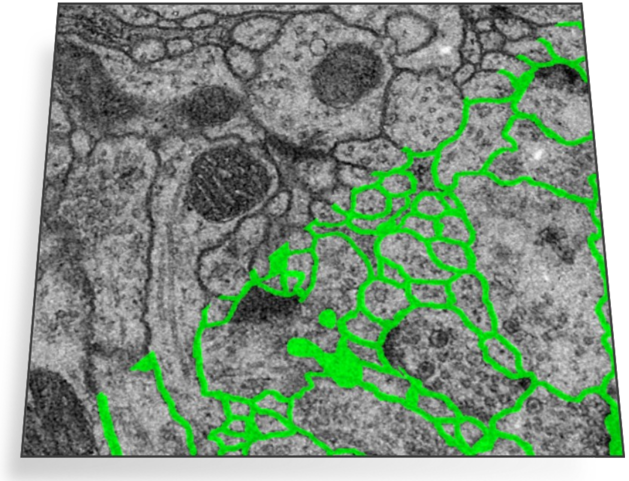
Affine group equivariant CNNs

translations

stabilizer / local symmetries (rotations / reflections / scaling / shearing / ...)

affine groups: $\mathrm{Aff}(G) := (\mathbb{R}^d, +) \rtimes G \qquad G \leq \mathrm{GL}(d)$

action on $\mathbb{R}^d$: $(tg)x := gx + t$



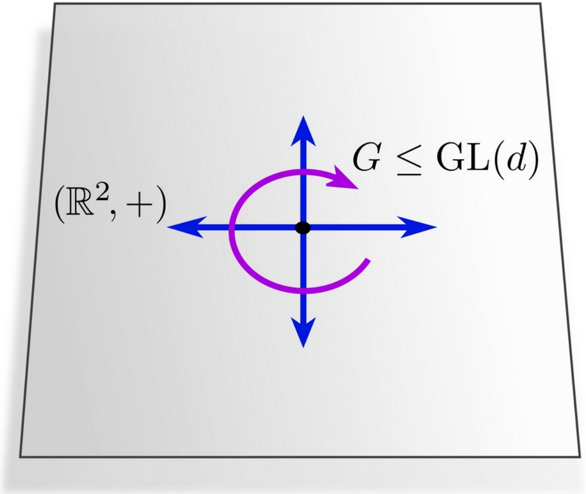


Image from ISBI 2012 EM segmentation challenge

Affine group equivariant CNNs

translations

stabilizer / local symmetries (rotations / reflections / scaling / shearing / ...)

affine groups: $\mathrm{Aff}(G) := (\mathbb{R}^d, +) \rtimes G \qquad G \leq \mathrm{GL}(d)$

action on $\mathbb{R}^d$: $(tg)x := gx + t$

action on feature spaces ?

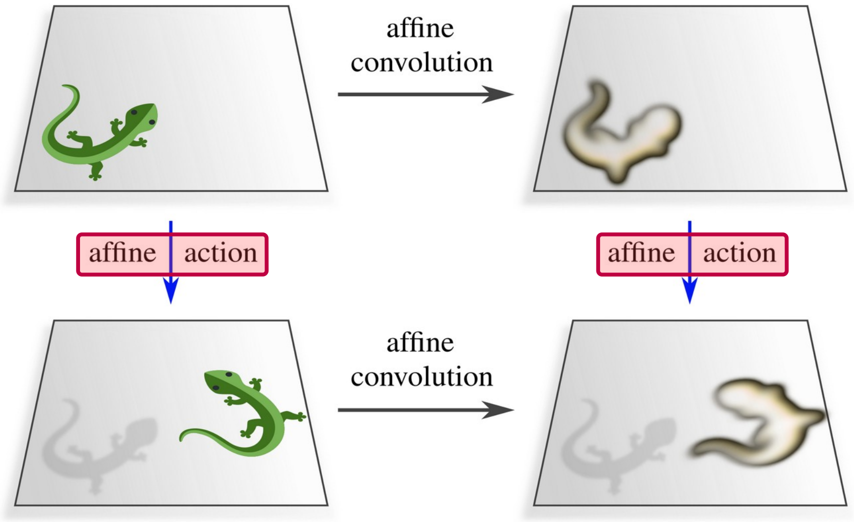


Image from ISBI 2012 EM segmentation challenge

Feature vector fields

feature vector fields on Euclidean spaces ...

... are functions $f : \mathbb{R}^d \to \mathbb{R}^c$ that assign feature vectors $f(x) \in \mathbb{R}^c$ to points $x \in \mathbb{R}^d$ (like feature maps)

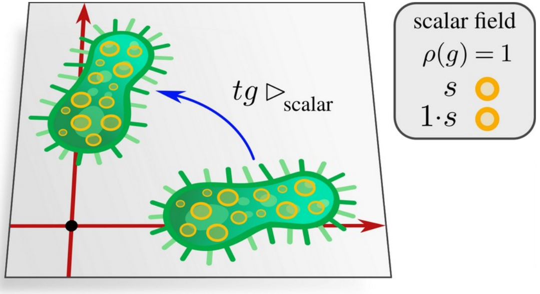
... carry an $\mathrm{Aff}(G)$-action (the details depend on their *field type* ρ)

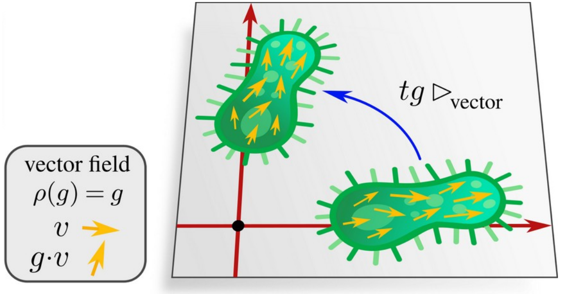
examples: scalar fields $s : \mathbb{R}^d \to \mathbb{R}^1$ transform like: $\big[(tg) \rhd s\big](x) \;=\; 1 \cdot s\big((tg)^{-1}x\big)$

tangent vector fields $v : \mathbb{R}^d \to \mathbb{R}^d$ transform like: $\big[(tg) \rhd v\big](x) \;=\; g \cdot v\big((tg)^{-1}x\big)$

$\mathrm{Aff}(G)$ acts here by... 1) moving feature vectors on $\mathbb{R}^d$

2) G-transforming feature vectors in $\mathbb{R}^c$





Feature vector fields

feature vector fields on Euclidean spaces ...

... are functions $f: \mathbb{R}^d \to \mathbb{R}^c$ that assign feature vectors $f(x) \in \mathbb{R}^c$ to points $x \in \mathbb{R}^d$ (like feature maps)

... carry an $\mathrm{Aff}(G)$-action (the details depend on their *field type* ρ)

examples: scalar fields $s: \mathbb{R}^d \to \mathbb{R}^1$ transform like: $\big[(tg) \rhd s\big](x) \;=\; \boxed{1 \cdot} \, s \boxed{\big((tg)^{-1}x\big)}$

tangent vector fields $v: \mathbb{R}^d \to \mathbb{R}^d$ transform like: $\big[(tg) \rhd v\big](x) \;=\; \boxed{g \cdot} \, v \boxed{\big((tg)^{-1}x\big)}$

$\mathrm{Aff}(G)$ acts here by... 1) moving feature vectors on $\mathbb{R}^d$

2) G-transforming feature vectors in $\mathbb{R}^c$

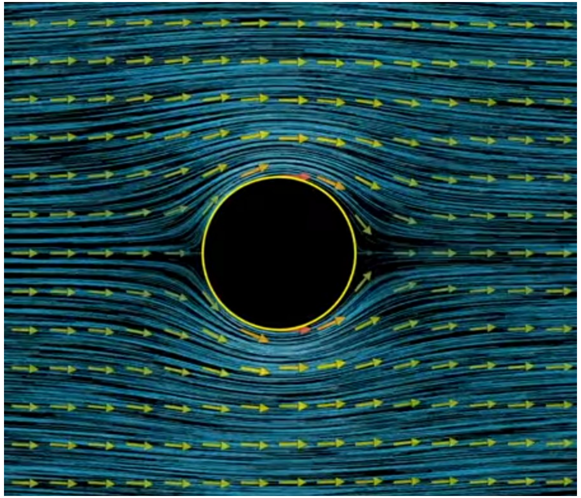
ρ-feature fields $f: \mathbb{R}^d \to \mathbb{R}^c$ transform like: $\big[(tg) \rhd f\big](x) \;=\rho(g)\, f\,\big((tg)^{-1}x\big)$

where $\rho: G \to \mathrm{GL}(c)$ is a G-representation acting on individual feature vectors in $\mathbb{R}^c$

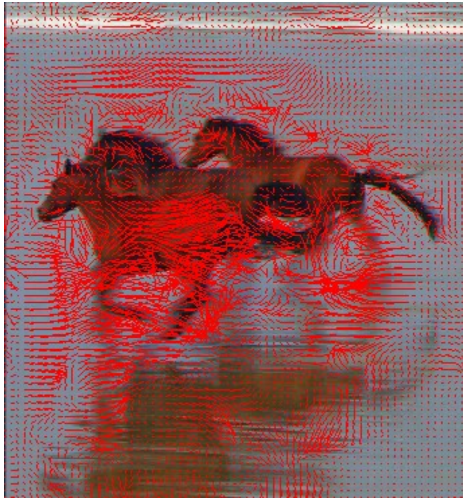
ρ-feature fields form an $\mathrm{Aff}(G)$-representation, denoted as **induced representation** $\mathrm{Ind}_G^{\mathrm{Aff}(G)}\,\rho$

Feature vector fields - examples

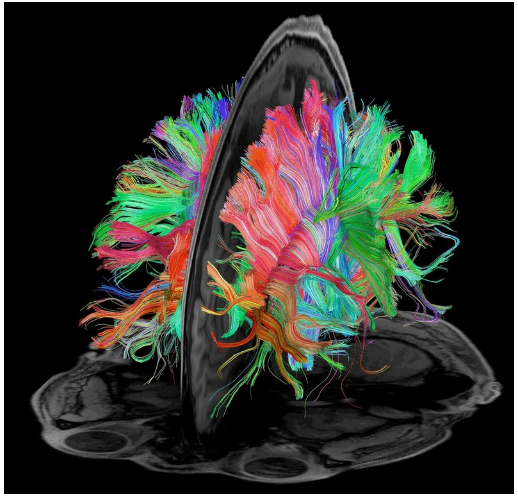
fluid flow



optical flow



diffusion tensor image



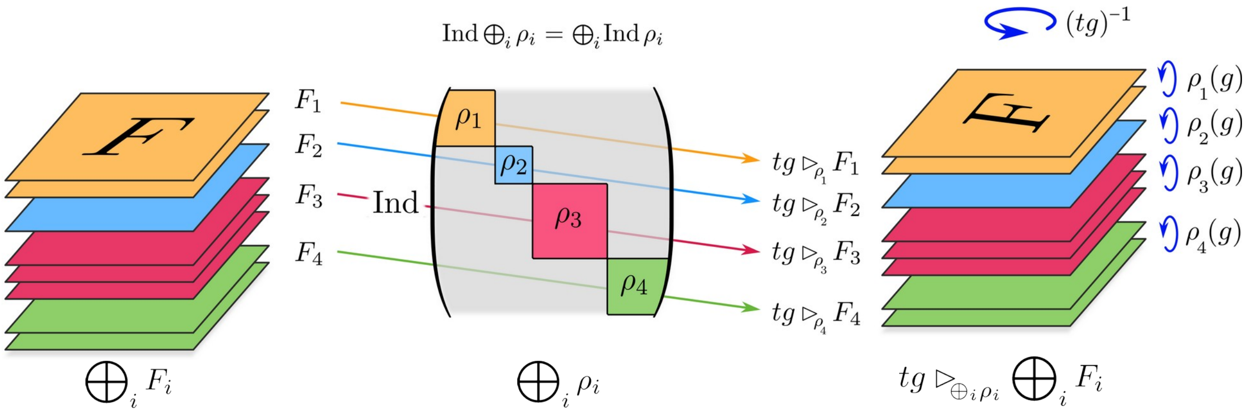
Feature vector fields – direct sum

conventional CNNs operate on a “stack” of multiple independent feature map channels

$\Rightarrow$ *#channels* as hyperparameter

steerable CNNs operate on ”stacks” $\bigoplus_i f_i$ of multiple independent feature fields

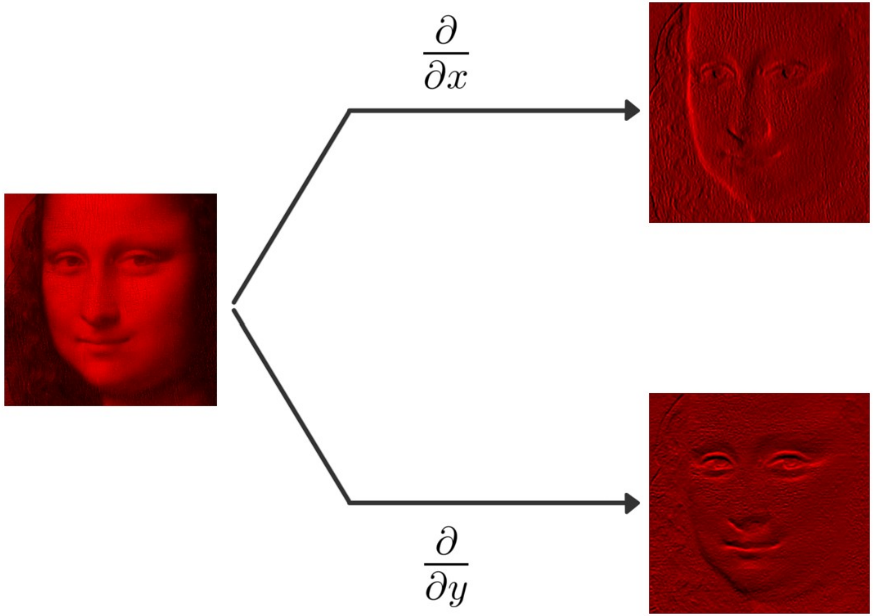
$\Rightarrow$ field *types* ρ_i and *multiplicities* as hyperparameters



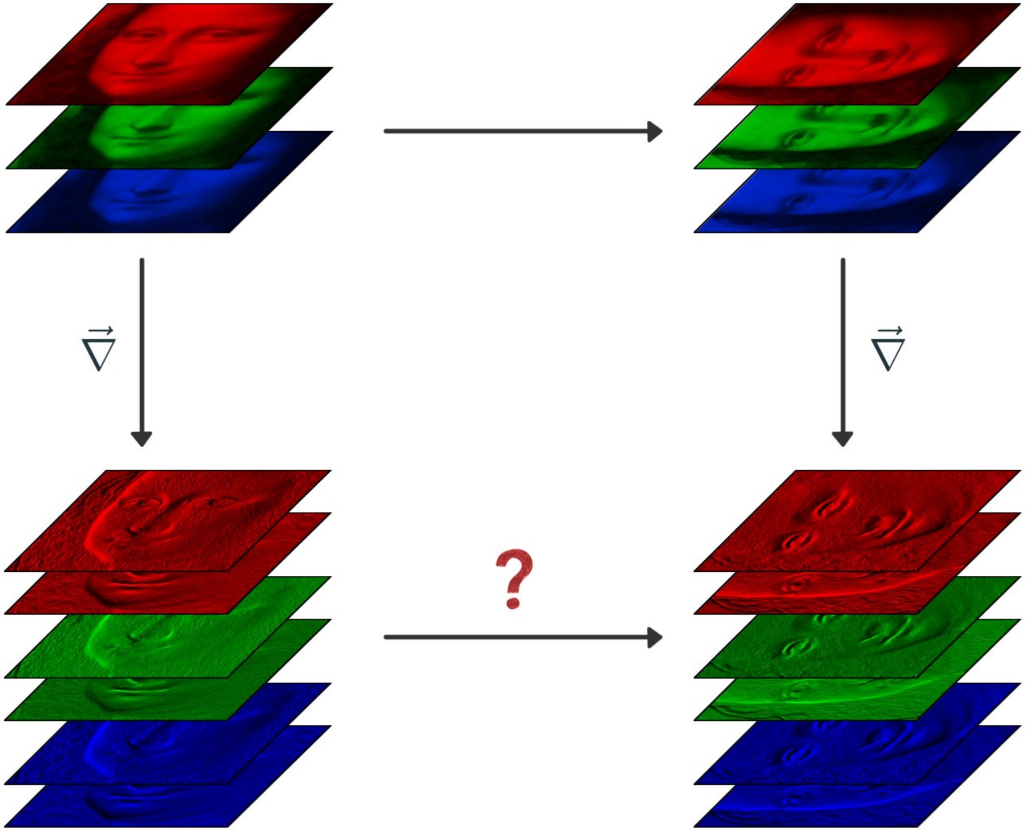
Feature vector fields - examples

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix} \longrightarrow \underbrace{\begin{pmatrix} 1 & & \\ & 1 & \\ & & 1 \end{pmatrix}}_{\rho(r) = \bigoplus_{i=1}^{3} \mathrm{id}} \cdot \begin{pmatrix} r \\ g \\ b \end{pmatrix}$$

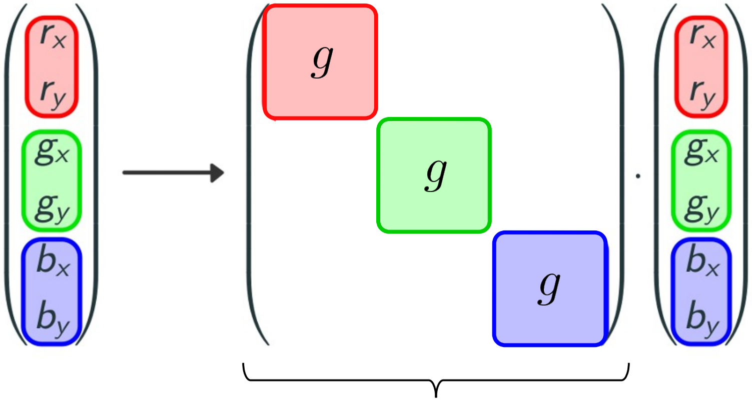
Feature vector fields - examples



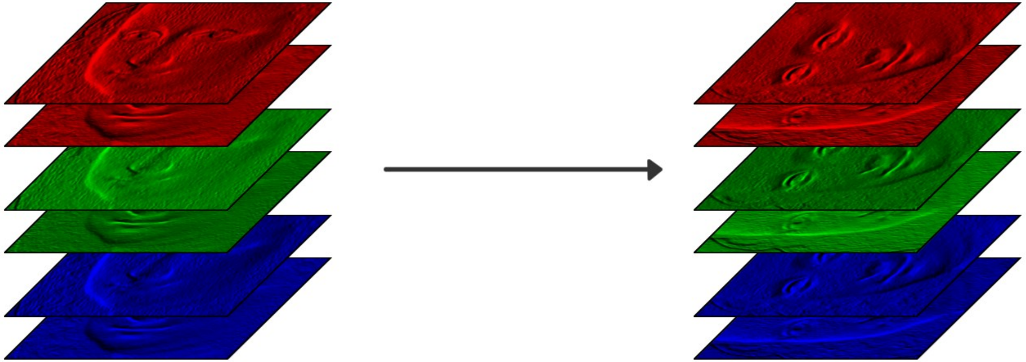
Feature vector fields - examples



Feature vector fields - examples



$$\rho(g) = \bigoplus_{i=1}^{3} g$$



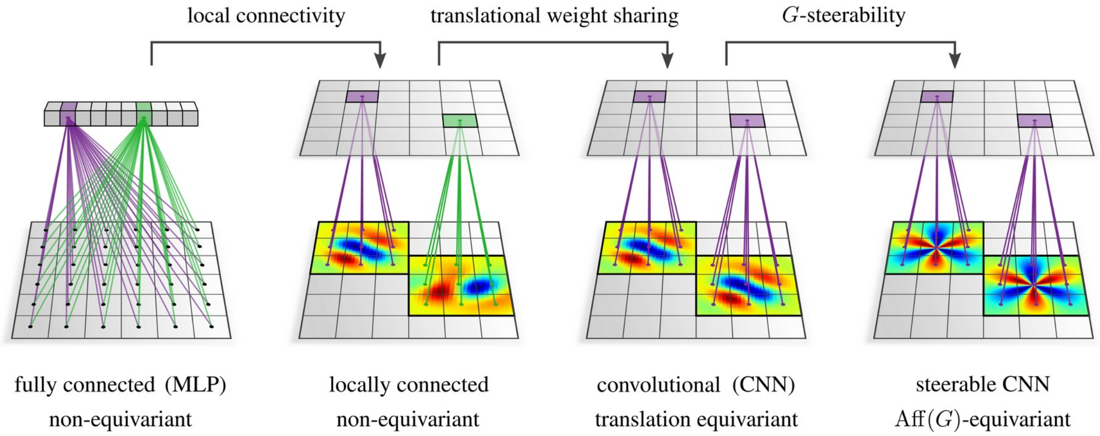
Steerable CNNs

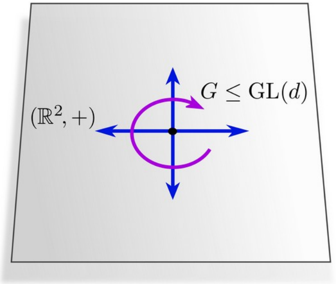
Steerable CNN layers map between feature fields of types ρ_{in} and ρ_{out}

$$
\begin{tikzcd}
\mathcal{L}^2\left(\mathbb{R}^d, \mathbb{R}^{c_{\text{in}}}\right) \arrow[r, "L"] \arrow[d, "\operatorname{Ind}\rho_{\text{in}}(tg)"'] & \mathcal{L}^2\left(\mathbb{R}^d, \mathbb{R}^{c_{\text{out}}}\right) \arrow[d, "\operatorname{Ind}\rho_{\text{out}}(tg)"] \\
\mathcal{L}^2\left(\mathbb{R}^d, \mathbb{R}^{c_{\text{in}}}\right) \arrow[r, "L"'] & \mathcal{L}^2\left(\mathbb{R}^d, \mathbb{R}^{c_{\text{out}}}\right)
\end{tikzcd}
$$

Steerable CNNs

Steerable CNN layers map between feature fields of types ρ_{in} and ρ_{out}





approach: - start with flexible ansatz for layers

- demand $\mathrm{Aff}(G)$-equivariance, resulting in...

1) spatial weight sharing —— $(\mathbb{R}^d, +) \rtimes G =: \mathrm{Aff}(G)$

2) *G*-steerability

Linear equivariant maps $\Leftrightarrow$ *G-steerable* convolutions

ansatz for linear map:

generic integral transform $\mathtt{I}_\kappa : L^2(\mathbb{R}^d, \mathbb{R}^{c_{\text{in}}}) \to L^2(\mathbb{R}^d, \mathbb{R}^{c_{\text{out}}})$

parameterized by 2-point correlator $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{c_{\text{out}} \times c_{\text{in}}}$

and defined by $\mathtt{I}_\kappa\big[f\big](x) := \int_{\mathbb{R}^d} dy\ \kappa(x,y)\, f(y)$

now demand equivariance:

Theorem. *The integral transform $\mathtt{I}_\kappa$ is $\mathrm{Aff}(G)$ equivariant iff:*

1) *it is a* convolution integral

$$\mathtt{I}_\kappa\big[f\big](x) \;=\; \big[K * f\big](x) \;=\; \int_{\mathbb{R}^d} dy\, K(x-y)\, f(y)\,.$$

with a matrix valued kernel $K : \mathbb{R}^d \to \mathbb{R}^{c_{\text{out}} \times c_{\text{in}}}$ defined by translation relativity $\kappa(x,y) = K(x-y)$

2) *the kernel is G-*steerable*:* $\quad K(gx) \;=\; \frac{1}{|\det g|} \rho_{\text{out}}(g)\, K(x)\, \rho_{\text{in}}(g)^{-1} \quad \forall\, g \in G,\ x \in \mathbb{R}^d$

Linear equivariant maps $\Leftrightarrow$ *G-steerable* convolutions

ansatz for linear map: generic integral transform $\mathtt{I}_\kappa : L^2(\mathbb{R}^d, \mathbb{R}^{c_\text{in}}) \to L^2(\mathbb{R}^d, \mathbb{R}^{c_\text{out}})$

given access to *G*-steerable kernels,

this is just a standard convolution as you know it

(our PyTorch library **escnn** implements them for compact G)

$) \, f(y)$

now demand equivaria

Theorem. *The integral transform* $\mathtt{I}_\kappa$ *is* $\mathrm{Aff}(G)$ *equivariant iff:*

1) *it is a* $\boxed{\text{convolution integral}}$

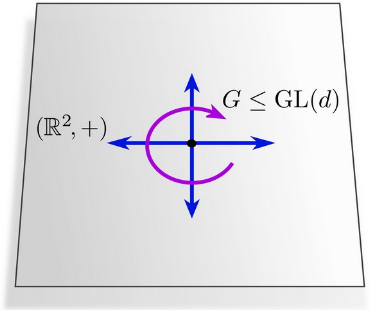
$$\mathtt{I}_\kappa\big[f\big](x) \;=\; \big[K * f\big](x) \;=\; \int_{\mathbb{R}^d} dy\, K(x-y)\, f(y)\,.$$

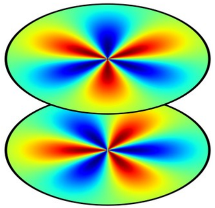
with a matrix valued kernel $K : \mathbb{R}^d \to \mathbb{R}^{c_\text{out} \times c_\text{in}}$ *defined by translation relativity* $\kappa(x,y) = K(x-y)$

2) $\boxed{\textit{the kernel is } G\text{-steerable:}}$ $\quad K(gx) \;=\; \frac{1}{|\det g|}\rho_\text{out}(g)\, K(x)\, \rho_\text{in}(g)^{-1} \quad \forall\, g \in G,\; x \in \mathbb{R}^d$

G-*steerable* kernels

G-steerable (or G-equivariant) kernels account for *local stabilizer subgroup transformations* of their field of view





more precisely, if 1) the field of view transforms spatially via g

2) its feature vectors transform according to $\rho_{\mathrm{in}}(g)$

$$\left.\begin{array}{l} \text{1) the field of view transforms spatially via } g \\ \text{2) its feature vectors transform according to } \rho_{\mathrm{in}}(g) \end{array}\right\} \mathrm{Ind}_G^{\mathrm{Aff}(G)} \rho_{\mathrm{in}}(g)$$

then it is guaranteed by G-steerability that the output feature transforms according to $\rho_{\mathrm{out}}(g)$

G-*steerable* kernels – reflection group examples

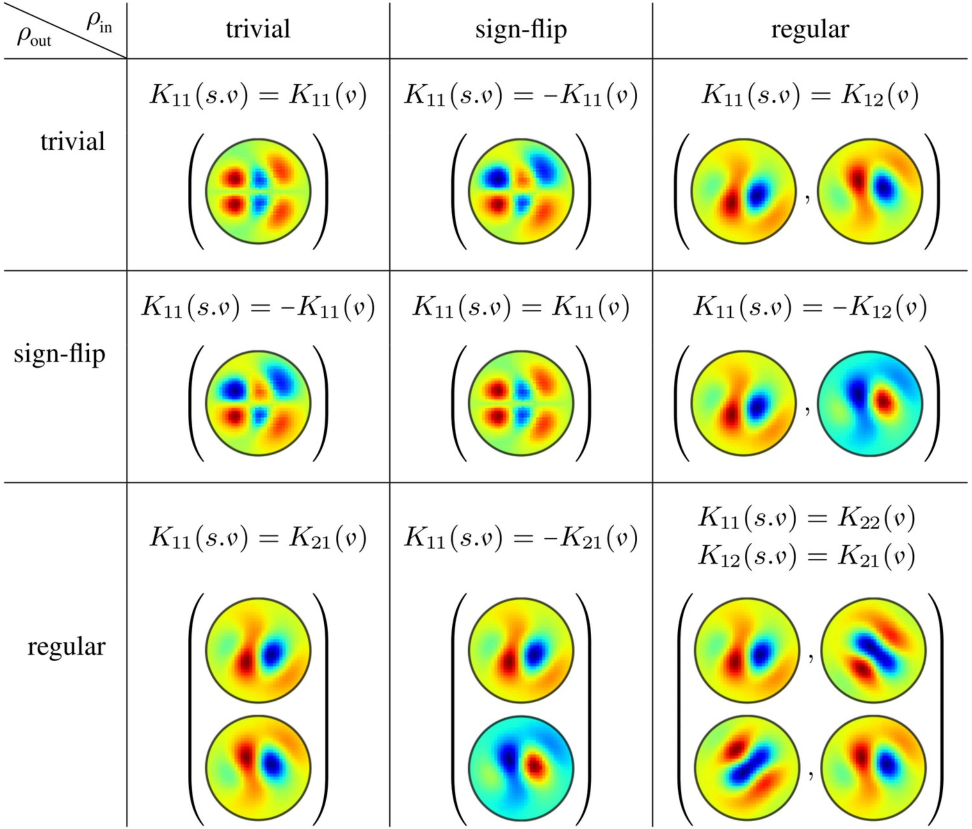
example: *reflection* steerable kernels

$$G = \{e, s\}, \quad s^2 = e$$

$$K(gx) = \boxed{\cancel{\frac{1}{|\det g|}}}\, \rho_{\text{out}}(g) K(x) \rho_{\text{in}}(g)^{\boxed{\cancel{-1}}}$$

$$\forall\, \boxed{g \in G,}\ \ x \in \mathbb{R}^2$$

representation ρ	group elements identity e	 reflection s
trivial / scalar	(1)	(1)
sign-flip / pseudo-scalar	(1)	(-1)
regular	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$



full derivation of these examples @ Weiler et al. 2021, *Coordinate Independent Convolutional Networks*, Section 5.3.3

G-*steerable* kernels – reflection group examples

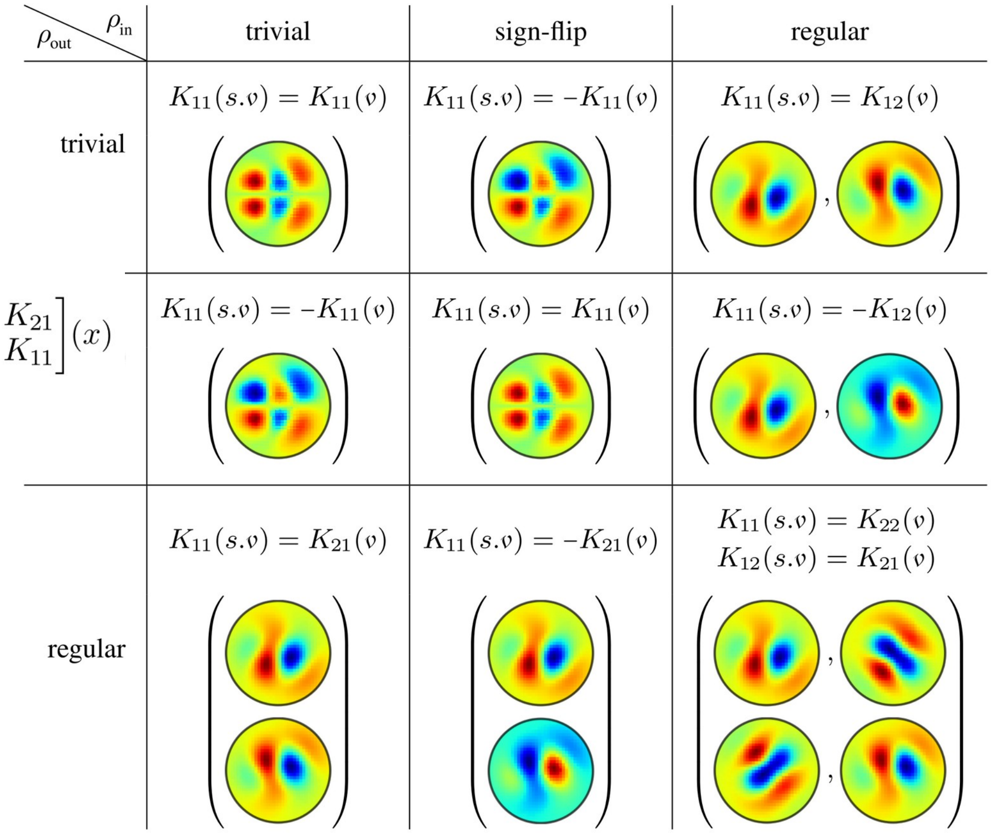
example: *reflection* steerable kernels

$$G = \{e, s\}, \quad s^2 = e$$

- **regular → regular:** $K : \mathbb{R}^2 \to \mathbb{R}^{2\times 2}$

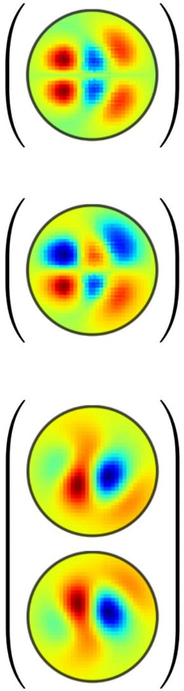
$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}(s.x) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}(x) \cdot \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} K_{22} & K_{21} \\ K_{12} & K_{11} \end{bmatrix}(x)$$

representation ρ	group elements: identity e	reflection s
trivial / scalar	(1)	(1)
sign-flip / pseudo-scalar	(1)	(-1)
regular	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$



full derivation of these examples @ Weiler et al. 2021, *Coordinate Independent Convolutional Networks*, Section 5.3.3

G-*steerable* kernels – expansion in linear basis



to solve the G-steerability kernel constraint in general, observe that:

- the set $\{K : \mathbb{R}^d \to \mathbb{R}^{c_\text{out} \times c_\text{in}}\}$ of *unconstrained* convolution kernels forms a *vector space*

- the constraint $K(gx) \;=\; \frac{1}{|\det g|} \rho_\text{out}(g)\, K(x)\, \rho_\text{in}(g)^{-1} \quad \forall\, g \in G,\; x \in \mathbb{R}^d$ is *linear*

$\Longrightarrow$ G-steerable kernels form a *linear (vector) subspace* !

to parameterize steerable convolutions:

1) solve for a *basis* $\{K_1, \dots, K_N\}$ of G-steerable kernels (precomputation step)

2) expand kernel in this basis with trainable weights: $K = \sum_{i=1}^{N} w_i K_i$ (during forward pass)

G-*steerable* kernels – Wigner-Eckart theorem

the kernel constraint is for compact G analytically solved (including in particular any $G \leq \mathrm{O}(d)$)

A WIGNER-ECKART THEOREM FOR GROUP EQUIVARIANT CONVOLUTION KERNELS

Leon Lang*
AMLab, CSL
University of Amsterdam
l.lang@uva.nl

Maurice Weiler
AMLab, QUVA Lab
University of Amsterdam
m.weiler.ml@gmail.com

A PROGRAM TO BUILD E(n)-EQUIVARIANT STEERABLE CNNS

Gabriele Cesa
Qualcomm AI Research*
University of Amsterdam
gcesa@qti.qualcomm.com

Leon Lang
University of Amsterdam
l.lang@uva.nl

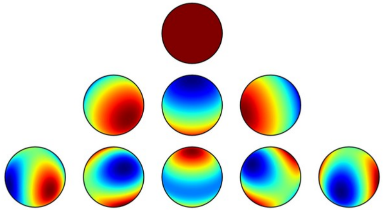
Maurice Weiler
University of Amsterdam
m.weiler.ml@gmail.com

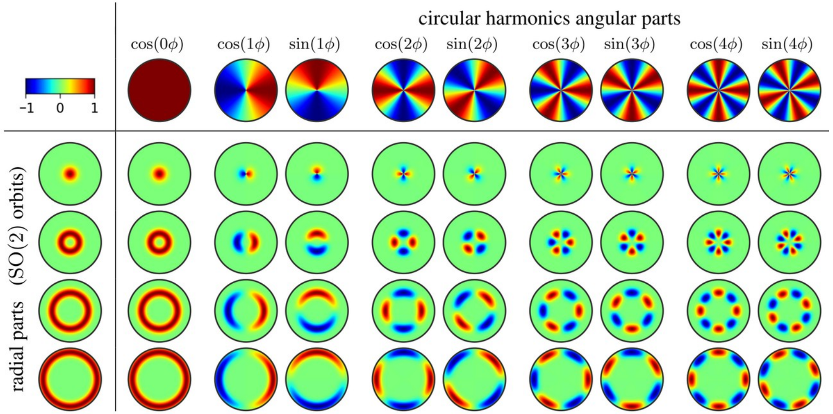

Leon Lang


Gabriele Cesa

the solution decomposes steerable kernels into:

- harmonics on *G*-orbits (Peter-Weyl)
- *Clebsch-Gordan coefficients*
- *irrep endomorphisms*





G-*steerable* kernels – Wigner-Eckart theorem

the kernel constraint is for compact G analytically solved (including in particular any $G \leq \mathrm{O}(d)$)

A WIGNER-ECKART THEOREM FOR GROUP EQUIVARIANT CONVOLUTION KERNELS

Leon Lang*
AMLab, CSL
University of Amsterdam
l.lang@uva.nl

Maurice Weiler
AMLab, QUVA Lab
University of Amsterdam
m.weiler.ml@gmail.com

A PROGRAM TO BUILD E(n)-EQUIVARIANT STEERABLE CNNS

Gabriele Cesa
Qualcomm AI Research*
University of Amsterdam
gcesa@qti.qualcomm.com

Leon Lang
University of Amsterdam
l.lang@uva.nl

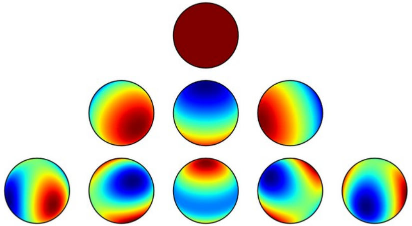
Maurice Weiler
University of Amsterdam
m.weiler.ml@gmail.com


Leon Lang

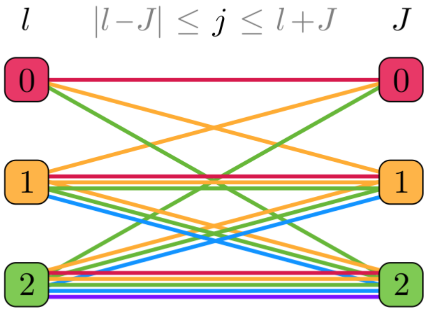

Gabriele Cesa

the solution decomposes steerable kernels into:

- harmonics on *G*-orbits (Peter-Weyl)
- *Clebsch-Gordan coefficients*
- *irrep endomorphisms*

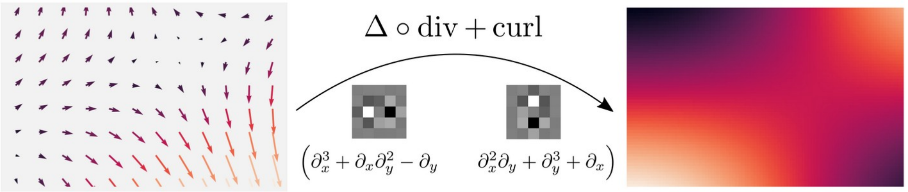


we get *transition rules* between irrep-fields (as in quantum mechanics)



transition rules for $\mathrm{SO}(3)$

Linear equivariant maps $\iff$ G-*steerable* convolutions



STEERABLE PARTIAL DIFFERENTIAL OPERATORS FOR EQUIVARIANT NEURAL NETWORKS

Erik Jenner*
University of Amsterdam
erik@ejenner.com

Maurice Weiler
University of Amsterdam
m.weiler.ml@gmail.com

Erik Jenner

linear maps revisited:

our integral transform ansatz $\mathtt{I}_\kappa[f](x) := \int_{\mathbb{R}^d} dy\ \kappa(x,y)\, f(y)$ does not cover all possible linear maps

a stronger version of the theorem proves:

continuous, $\mathrm{Aff}(G)$-equivariant linear maps $\iff$ convolutions with G-steerable Schwartz distributions

the distributional setting covers in particular *equivariant partial differential operators*

$\mathrm{Aff}(G)$-equivariant bias summation

as before:

consider a general bias summation operation $\quad B_{\mathfrak{b}} :\ L^2(\mathbb{R}^d, \mathbb{R}^c) \to L^2(\mathbb{R}^d, \mathbb{R}^c), \quad f \mapsto f + \mathfrak{b}$

parameterized by a **bias field** $\mathfrak{b} : \mathbb{R}^d \to \mathbb{R}^c \quad \Longrightarrow$ allows to sum a *different bias* $\mathfrak{b}(x) \in \mathbb{R}^c$ at each $x \in \mathbb{R}^d$

demanding equivariance, we get:

Theorem. *The bias field summation* $\mathrm{Aff}(G)$*-equivariant iff the bias field is* $\mathrm{Aff}(G)$-invariant. *This requires in particular*

1) *a spatially constant bias field, i.e.* $\mathfrak{b}(x) = b$ *for some shared bias* $b \in \mathbb{R}^c$, *and*

2) *this shared bias needs to be* G*-invariant, that is,* $b = \rho(g)\, b \quad \forall\, g \in G$.

$\Longrightarrow$ one may only sum biases to the *trivial irrep subspaces* of ρ (e.g. not to tangent vector fields)

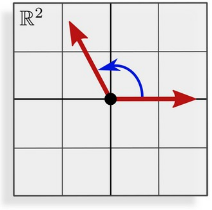
$\mathrm{Aff}(G)$-equivariant nonlinearities

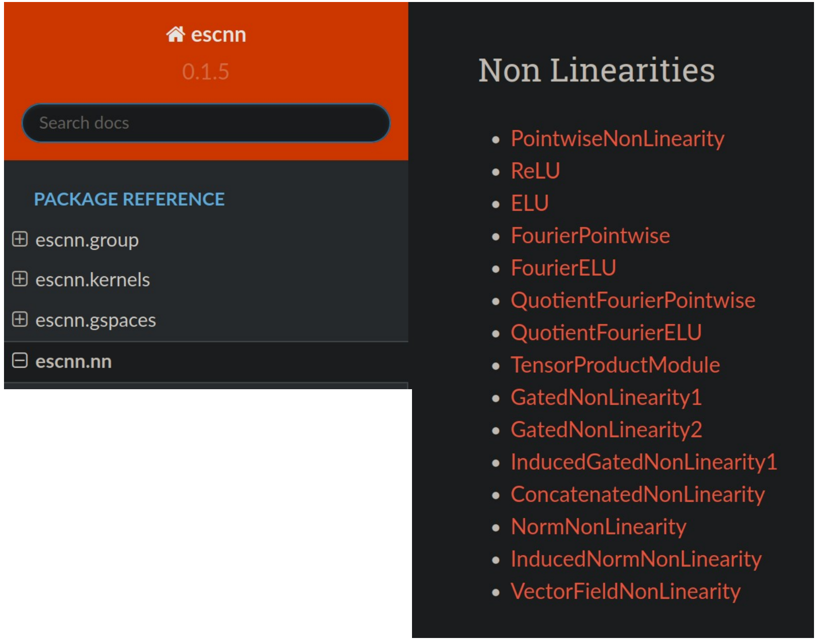
(local) nonlinearities also need to be

1) spatially shared
2) *G*-equivariant

the admissible choices depend on the field type ρ

e.g. we can't use channel-wise ReLUs on tangent vectors





more details in Pim's talk

Implementation & empirical results

```python
from escnn import gspaces
from escnn import nn
import torch

r2_act = gspaces.rot2dOnR2(N=8)
feat_type_in  = nn.FieldType(r2_act,  3*[r2_act.trivial_repr])
feat_type_out = nn.FieldType(r2_act, 10*[r2_act.regular_repr])

conv = nn.R2Conv(feat_type_in, feat_type_out, kernel_size=5)
relu = nn.ReLU(feat_type_out)

x = torch.randn(16, 3, 32, 32)
x = feat_type_in(x)

y = relu(conv(x))
```

e2cnn / escnn library

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

PyTorch extension for $\mathrm{Aff}(G)$-steerable CNNs (for compact *G*)

General E(2) - Equivariant Steerable CNNs

Maurice Weiler*
University of Amsterdam, QUVA Lab
m.weiler@uva.nl

Gabriele Cesa*†
University of Amsterdam
cesa.gabriele@gmail.com

A PROGRAM TO BUILD
E(n)-EQUIVARIANT STEERABLE CNNS

Gabriele Cesa
Qualcomm AI Research*
University of Amsterdam
gcesa@qti.qualcomm.com

Leon Lang
University of Amsterdam
l.lang@uva.nl

Maurice Weiler
University of Amsterdam
m.weiler.ml@gmail.com

convolution in native PyTorch:

```
conv = nn.Conv2d(in_channels=3, out_channels=64, kernel_size=5)
```

convolution in e2cnn / escnn:

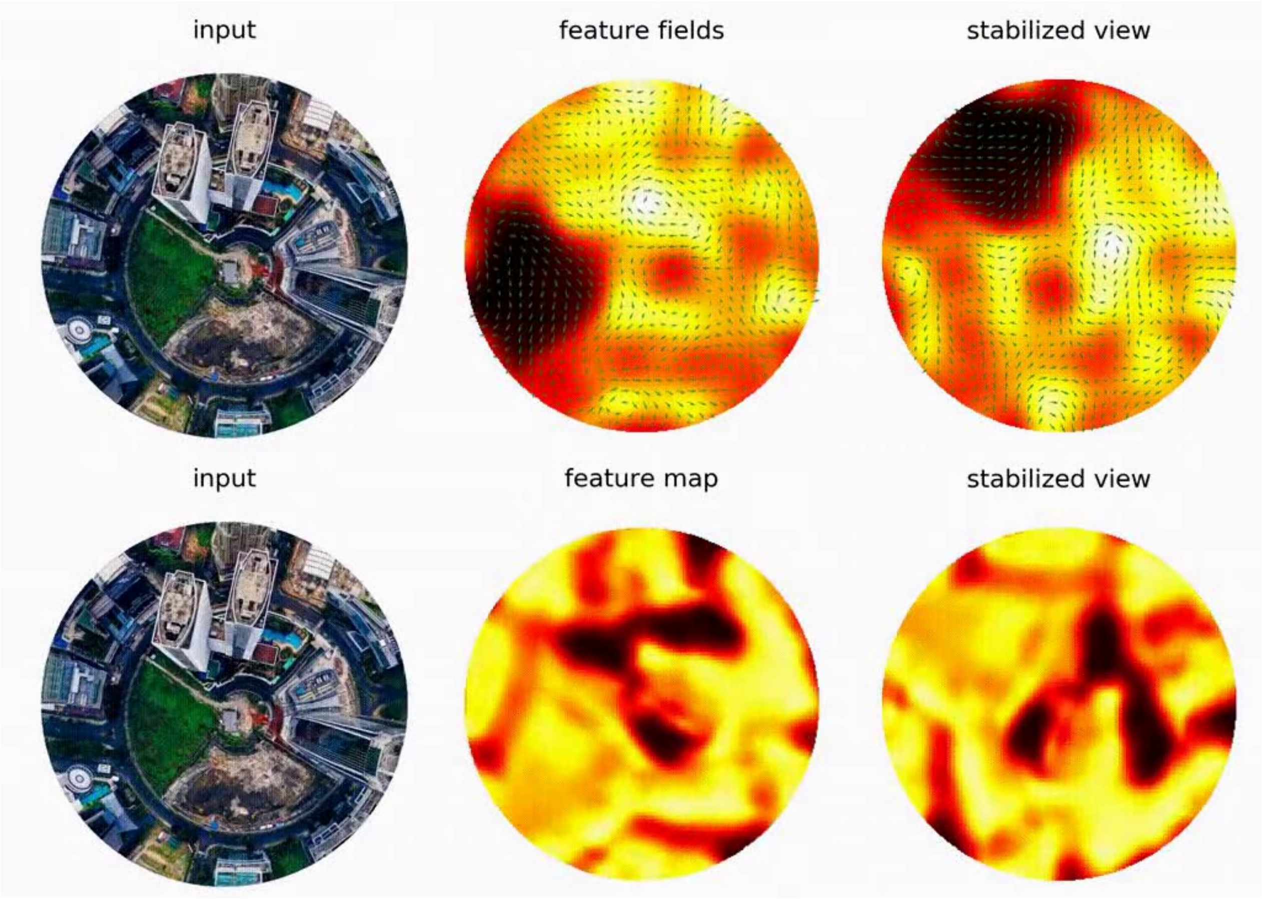
- fix symmetry group $G = \mathbb{Z}_8$ + action on $\mathbb{R}^2$
- fix types + multiplicities of feature fields
- construct Aff(*G*)-equivariant convolution

```
r2_act = gspaces.Rot2dOnR2(N=8)

feat_type_in  = nn.FieldType(r2_act,  3*[r2_act.trivial_repr])
feat_type_out = nn.FieldType(r2_act, 10*[r2_act.regular_repr])

conv = nn.R2Conv(feat_type_in, feat_type_out, kernel_size=5)
```

Equivariance demonstration

SE(2)-steerable CNN:

conventional CNN:



Empirical results – group size

consider: cyclic groups $G = \mathrm{C}_N$

dihedral groups $G = \mathrm{D}_N$ for different N

G-augmented MNIST variants for $G = \mathrm{O}(2)$ or $\mathrm{SO}(2)$

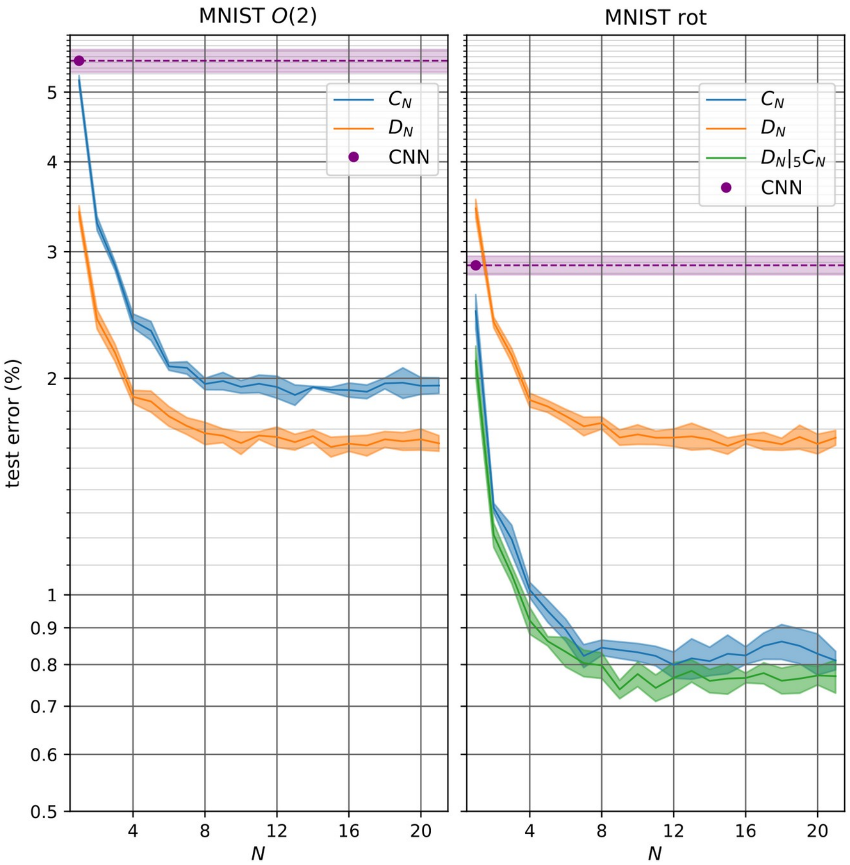
$\dots$

apply G-*invariant* (!) map at final layer

results:

decreasing classification error for larger groups

too much invariance hurts,
but can be solved via *group restriction* $\mathrm{Res}_{\mathrm{C}_N}^{\mathrm{D}_N}$



Emperical results

extensive benchmark of:

- groups $G \leq \mathrm{O}(2)$
- *G*-representations / field types
- *G*-equivariant nonlinearities
- invariant maps

covering a wide range of related work and new models

	group	representation		nonlinearity	invariant map	citation	MNIST O(2)	MNIST rot	MNIST 12k
1	$\{e\}$	(conventional CNN)		ELU	-	-	5.53 ± 0.20	2.87 ± 0.09	0.91 ± 0.06
2	C_1					[7, 9]	5.19 ± 0.08	2.48 ± 0.13	0.82 ± 0.01
3	C_2					[7, 9]	3.29 ± 0.07	1.32 ± 0.02	0.87 ± 0.04
4	C_3					-	2.87 ± 0.04	1.19 ± 0.06	0.80 ± 0.03
5	C_4					[6, 1, 7, 9, 10]	2.40 ± 0.05	1.02 ± 0.03	0.99 ± 0.03
6	C_6	regular	ρ_{reg}	ELU	G-pooling	[8]	2.08 ± 0.03	0.89 ± 0.03	0.84 ± 0.02
7	C_8					[7, 9]	1.96 ± 0.04	0.84 ± 0.02	0.89 ± 0.03
8	C_{12}					[7]	1.95 ± 0.07	0.80 ± 0.03	0.89 ± 0.03
9	C_{16}					[7, 9]	1.93 ± 0.04	0.82 ± 0.02	0.95 ± 0.04
10	C_{20}					[7]	1.95 ± 0.05	0.83 ± 0.05	0.94 ± 0.06
11	C_4		$5\rho_{\text{reg}} \oplus 2\rho_{\text{quot}}^{\mathrm{C}_4/\mathrm{C}_2} \oplus 2\psi_0$			[1]	2.43 ± 0.05	1.03 ± 0.05	1.01 ± 0.03
12	C_8		$5\rho_{\text{reg}} \oplus 2\rho_{\text{quot}}^{\mathrm{C}_8/\mathrm{C}_2} \oplus 2\rho_{\text{quot}}^{\mathrm{C}_8/\mathrm{C}_4} \oplus 2\psi_0$			-	2.03 ± 0.05	0.84 ± 0.05	0.91 ± 0.02
13	C_{12}	quotient	$5\rho_{\text{reg}} \oplus 2\rho_{\text{quot}}^{\mathrm{C}_{12}/\mathrm{C}_2} \oplus 2\rho_{\text{quot}}^{\mathrm{C}_{12}/\mathrm{C}_4} \oplus 3\psi_0$			-	2.04 ± 0.04	0.81 ± 0.02	0.95 ± 0.02
14	C_{16}		$5\rho_{\text{reg}} \oplus 2\rho_{\text{quot}}^{\mathrm{C}_{16}/\mathrm{C}_2} \oplus 2\rho_{\text{quot}}^{\mathrm{C}_{16}/\mathrm{C}_4} \oplus 4\psi_0$			-	2.00 ± 0.01	0.86 ± 0.04	0.98 ± 0.04
15	C_{20}		$5\rho_{\text{reg}} \oplus 2\rho_{\text{quot}}^{\mathrm{C}_{20}/\mathrm{C}_2} \oplus 2\rho_{\text{quot}}^{\mathrm{C}_{20}/\mathrm{C}_4} \oplus 5\psi_0$			-	2.01 ± 0.05	0.83 ± 0.03	0.96 ± 0.04
16		regular/scalar	$\psi_0 \xrightarrow{\text{conv}} \rho_{\text{reg}} \xrightarrow{G\text{-pool}} \psi_0$	ELU, G-pooling		[6, 36]	2.02 ± 0.02	0.90 ± 0.03	0.93 ± 0.04
17	C_{16}	regular/vector	$\psi_1 \xrightarrow{\text{conv}} \rho_{\text{reg}} \xrightarrow{\text{vector pool}} \psi_1$	vector field		[13, 37]	2.12 ± 0.02	1.07 ± 0.03	0.78 ± 0.03
18		mixed vector	$\rho_{\text{reg}} \oplus \psi_1 \xrightarrow{\text{conv}} 2\rho_{\text{reg}} \xrightarrow[\text{pool}]{\text{vector}} \rho_{\text{reg}} \oplus \psi_1$	ELU, vector field		-	1.87 ± 0.03	0.83 ± 0.02	0.63 ± 0.02
19	D_1					-	3.40 ± 0.07	3.44 ± 0.10	0.98 ± 0.03
20	D_2					-	2.42 ± 0.07	2.39 ± 0.04	1.05 ± 0.03
21	D_3					-	2.17 ± 0.06	2.15 ± 0.05	0.94 ± 0.02
22	D_4					[6, 1, 38]	1.88 ± 0.04	1.87 ± 0.04	1.69 ± 0.03
23	D_6	regular	ρ_{reg}	ELU	G-pooling	[8]	1.77 ± 0.06	1.77 ± 0.04	1.00 ± 0.03
24	D_8					-	1.68 ± 0.06	1.73 ± 0.03	1.64 ± 0.02
25	D_{12}					-	1.66 ± 0.05	1.65 ± 0.05	1.67 ± 0.01
26	D_{16}					-	1.62 ± 0.04	1.65 ± 0.02	1.68 ± 0.04
27	D_{20}					-	1.64 ± 0.06	1.62 ± 0.05	1.69 ± 0.03
28	D_{16}	regular/scalar	$\psi_{0,0} \xrightarrow{\text{conv}} \rho_{\text{reg}} \xrightarrow{G\text{-pool}} \psi_{0,0}$	ELU, G-pooling		-	1.92 ± 0.03	1.88 ± 0.07	1.74 ± 0.04
29		irreps ≤ 1	$\bigoplus_{i=0}^{1} \psi_i$			-	2.98 ± 0.04	1.38 ± 0.09	1.29 ± 0.05
30		irreps ≤ 3	$\bigoplus_{i=0}^{3} \psi_i$			-	3.02 ± 0.18	1.38 ± 0.09	1.27 ± 0.03
31		irreps ≤ 5	$\bigoplus_{i=0}^{5} \psi_i$			-	3.24 ± 0.05	1.44 ± 0.10	1.36 ± 0.04
32		irreps ≤ 7	$\bigoplus_{i=0}^{7} \psi_i$			-	3.30 ± 0.11	1.51 ± 0.10	1.40 ± 0.07
33		$\mathbb{C}$-irreps ≤ 1	$\bigoplus_{i=0}^{1} \psi_i^{\mathbb{C}}$	ELU, norm-ReLU	conv2triv	[12]	3.39 ± 0.10	1.47 ± 0.06	1.42 ± 0.04
34		$\mathbb{C}$-irreps ≤ 3	$\bigoplus_{i=0}^{3} \psi_i^{\mathbb{C}}$			[12]	3.48 ± 0.16	1.51 ± 0.05	1.53 ± 0.07
35		$\mathbb{C}$-irreps ≤ 5	$\bigoplus_{i=0}^{5} \psi_i^{\mathbb{C}}$			-	3.59 ± 0.08	1.59 ± 0.05	1.55 ± 0.06
36		$\mathbb{C}$-irreps ≤ 7	$\bigoplus_{i=0}^{7} \psi_i^{\mathbb{C}}$			-	3.64 ± 0.12	1.61 ± 0.06	1.62 ± 0.03
37	SO(2)			ELU, squash		-	3.10 ± 0.09	1.41 ± 0.04	1.46 ± 0.05
38				ELU, norm-ReLU		-	3.23 ± 0.08	1.38 ± 0.08	1.33 ± 0.03
39				ELU, shared norm-ReLU	norm	-	2.88 ± 0.11	1.15 ± 0.06	1.18 ± 0.03
40		irreps ≤ 3	$\bigoplus_{i=0}^{3} \psi_i$	shared norm-ReLU		-	3.61 ± 0.09	1.57 ± 0.05	1.88 ± 0.05
41				ELU, gate	conv2triv	-	2.37 ± 0.06	1.09 ± 0.03	1.10 ± 0.02
42				ELU, shared gate		-	2.33 ± 0.06	1.11 ± 0.03	1.12 ± 0.04
43				ELU, gate	norm	-	2.23 ± 0.09	1.04 ± 0.04	1.05 ± 0.06
44				ELU, shared gate		-	2.20 ± 0.06	1.01 ± 0.03	1.03 ± 0.03
45		irreps $= 0$	$\psi_{0,0}$	ELU	-	-	5.46 ± 0.46	5.21 ± 0.29	3.98 ± 0.04
46		irreps ≤ 1	$\psi_{0,0} \oplus \psi_{1,0} \oplus 2\psi_{1,1}$			-	3.31 ± 0.17	3.37 ± 0.18	3.05 ± 0.09
47		irreps ≤ 3	$\psi_{0,0} \oplus \psi_{1,0} \bigoplus_{i=1}^{3} 2\psi_{1,i}$	ELU, norm-ReLU	O(2)-conv2triv	-	3.42 ± 0.03	3.41 ± 0.10	3.86 ± 0.09
48		irreps ≤ 5	$\psi_{0,0} \oplus \psi_{1,0} \bigoplus_{i=1}^{5} 2\psi_{1,i}$			-	3.59 ± 0.13	3.78 ± 0.31	4.17 ± 0.15
49		irreps ≤ 7	$\psi_{0,0} \oplus \psi_{1,0} \bigoplus_{i=1}^{7} 2\psi_{1,i}$			-	3.84 ± 0.25	3.90 ± 0.18	4.57 ± 0.27
50		Ind-irreps ≤ 1	$\mathrm{Ind}\,\psi_0^{\mathrm{SO}(2)} \oplus \mathrm{Ind}\,\psi_1^{\mathrm{SO}(2)}$			-	2.72 ± 0.05	2.70 ± 0.11	2.39 ± 0.07
51	O(2)	Ind-irreps ≤ 3	$\mathrm{Ind}\,\psi_0^{\mathrm{SO}(2)} \bigoplus_{i=1}^{3} \mathrm{Ind}\,\psi_i^{\mathrm{SO}(2)}$	ELU, Ind norm-ReLU	Ind-conv2triv	-	2.66 ± 0.07	2.65 ± 0.12	2.25 ± 0.06
52		Ind-irreps ≤ 5	$\mathrm{Ind}\,\psi_0^{\mathrm{SO}(2)} \bigoplus_{i=1}^{5} \mathrm{Ind}\,\psi_i^{\mathrm{SO}(2)}$			-	2.71 ± 0.11	2.84 ± 0.10	2.39 ± 0.09
53		Ind-irreps ≤ 7	$\mathrm{Ind}\,\psi_0^{\mathrm{SO}(2)} \bigoplus_{i=1}^{7} \mathrm{Ind}\,\psi_i^{\mathrm{SO}(2)}$			-	2.80 ± 0.12	2.85 ± 0.06	2.25 ± 0.08
54		irreps ≤ 3	$\psi_{0,0} \oplus \psi_{1,0} \bigoplus_{i=1}^{3} 2\psi_{1,i}$	ELU, gate	O(2)-conv2triv	-	2.39 ± 0.05	2.38 ± 0.07	2.28 ± 0.07
55					norm	-	2.21 ± 0.09	2.24 ± 0.06	2.15 ± 0.03
56		Ind-irreps ≤ 3	$\mathrm{Ind}\,\psi_0^{\mathrm{SO}(2)} \bigoplus_{i=1}^{3} \mathrm{Ind}\,\psi_i^{\mathrm{SO}(2)}$	ELU, Ind gate	Ind-conv2triv	-	2.13 ± 0.04	2.09 ± 0.05	2.05 ± 0.05
57					Ind-norm	-	1.96 ± 0.06	1.95 ± 0.05	1.85 ± 0.07

Emperical results

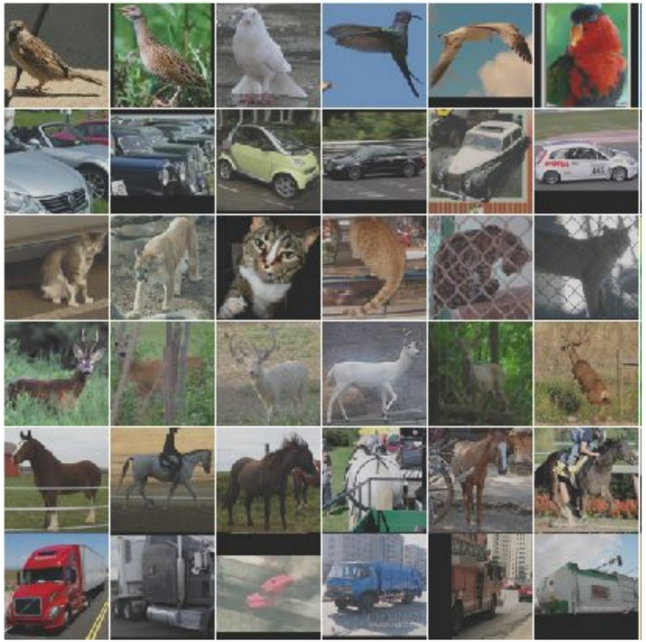
Dihedral point group D_N

group convolutions as drop in replacement

- same number of parameters
- same training setup
- no hyperparameter tuning

model	CIFAR-10	CIFAR-100	STL-10
CNN baseline	2.6 ± 0.1	17.1 ± 0.3	12.74 ± 0.23
GCNN	2.05 ± 0.03	14.30 ± 0.09	9.80 ± 0.40

Test errors on natural image datasets

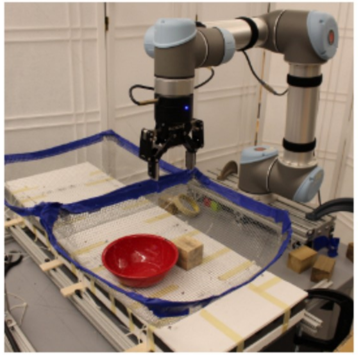


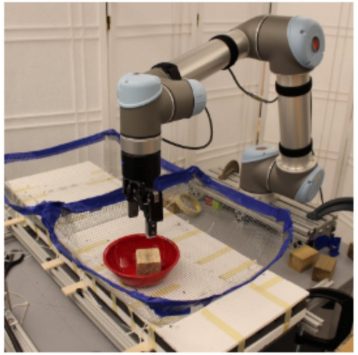
exploiting local symmetries

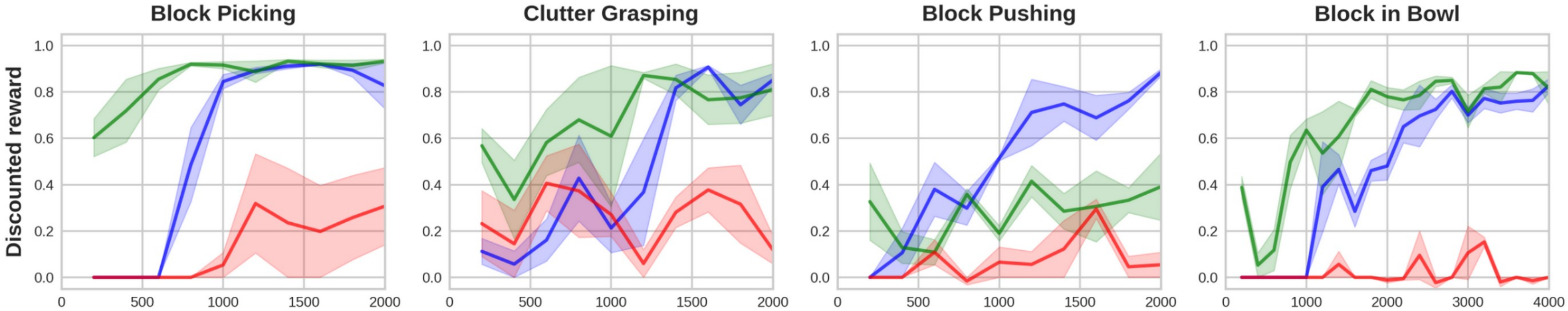
Emperical results – reinforcement learning

On-Robot Learning With Equivariant Models

Dian Wang **Mingxi Jia** **Xupeng Zhu** **Robin Walters** **Robert Platt**
Khoury College of Computer Sciences
Northeastern University
Boston, MA 02115, USA







Emperical results – equivariant convolutional Gaussian processes

Equivariant Learning of Stochastic Fields:
Gaussian Processes and Steerable Conditional Neural Processes

Peter Holderrieth [*1] **Michael Hutchinson** [*1] **Yee Whye Teh** [1 2]

convolutional GP with:

- mean = tangent vector field
- covariance = symmetric tensor field

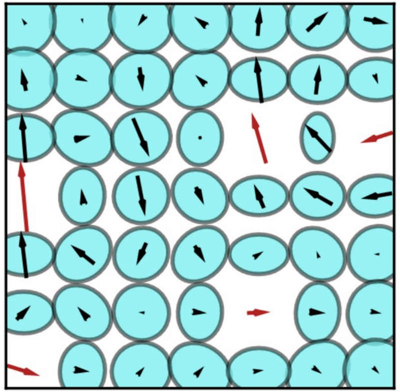


Table 3. Results on ERA5 weather experiment trained on US data. Mean log-likelihood $\pm$ 1 standard deviation over 5 random seeds reported. Left: tested on US data. Right: tested on China data.

Model	US	China
GP	0.386$\pm$0.005	-0.755$\pm$0.001
CNP	0.001$\pm$0.017	-2.456$\pm$0.365
ConvCNP	0.898$\pm$0.045	-0.890$\pm$0.059
SteerCNP (C_4)	**1.255**$\pm$0.019	-0.578$\pm$0.173
SteerCNP (C_8)	1.038$\pm$0.026	-0.582$\pm$0.104
SteerCNP (C_{16})	1.094$\pm$0.015	-0.550$\pm$0.073
SteerCNP (D_4)	1.037$\pm$0.037	**-0.429**$\pm$0.067
SteerCNP (D_8)	1.032$\pm$0.011	-0.539$\pm$0.129

Relation to group convolutions (correlations) & convolutions on homogeneous spaces

$$\left[K \star_G f\right](g) = \int_G K(g^{-1}h)\, f(h)\; d\lambda(h)$$

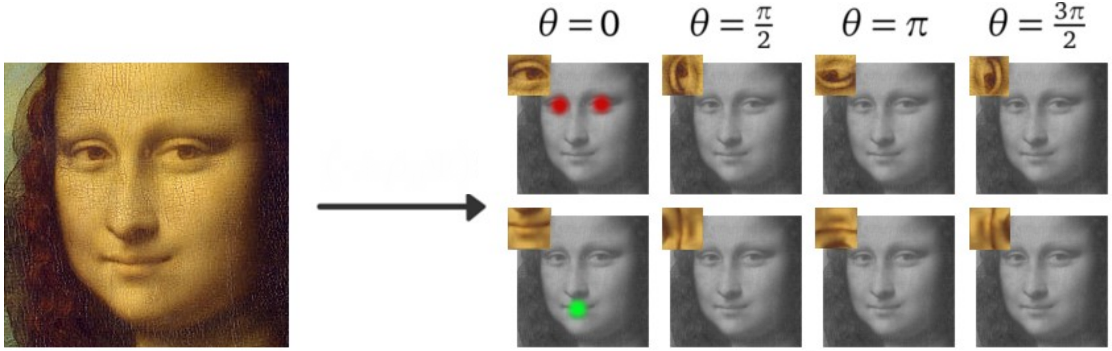
Group correlations

G-correlation can be thought of as patern matching with *G*-transformed templates (kernel)

conventional correlation on $\mathbb{R}^d$: $\langle t \blacktriangleright K \mid f \rangle_{\mathbb{R}^d} : \quad \mathbb{R}^d \to \mathbb{R}$ (feature map on $\mathbb{R}^d$)

$\mathbb{R}^d$ to $\mathrm{Aff}(G)$ *lifting* correlation: $\langle tg \blacktriangleright K \mid f \rangle_{\mathbb{R}^d} : \quad \mathrm{Aff}(G) \to \mathbb{R}$ (feature map on $\mathrm{Aff}(G)$)

$G = \mathrm{C}_4$



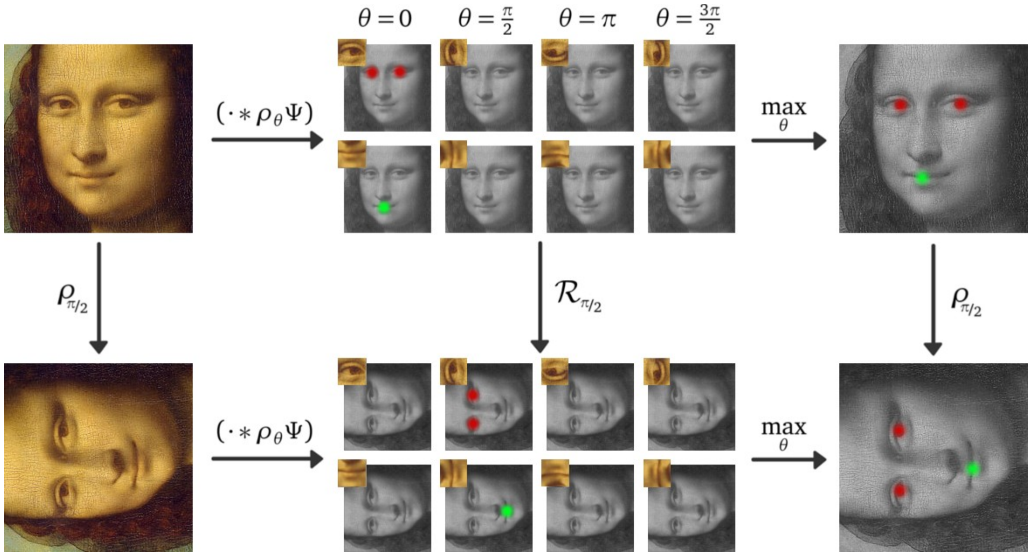
Group correlations

how to process lifted feature maps on $\mathrm{Aff}(G)$ further?

G-pooling: - pool over *G*-axis, i.e. the responses from kernels at the same position, but different *G*-pose

- maps back to *scalar fields* $\mathbb{R}^d \to \mathbb{R}$

- *G-invariant* responses, can not encode *G*-pose of features



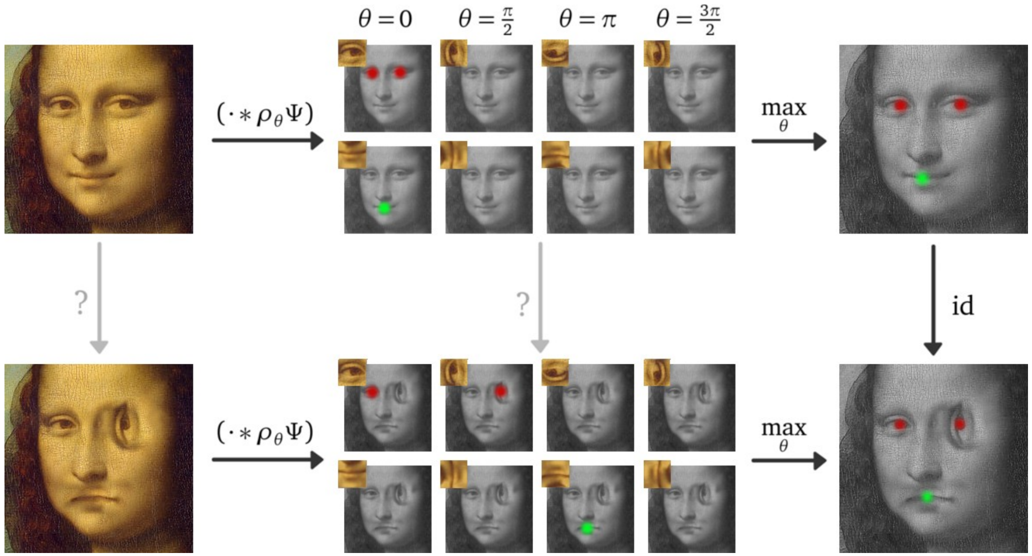
Group correlations

how to process lifted feature maps on $\mathrm{Aff}(G)$ further?

G-pooling: - pool over *G*-axis, i.e. the responses from kernels at the same position, but different *G*-pose

- maps back to *scalar fields* $\mathbb{R}^d \to \mathbb{R}$

- *G-invariant* responses, can not encode *G*-pose of features



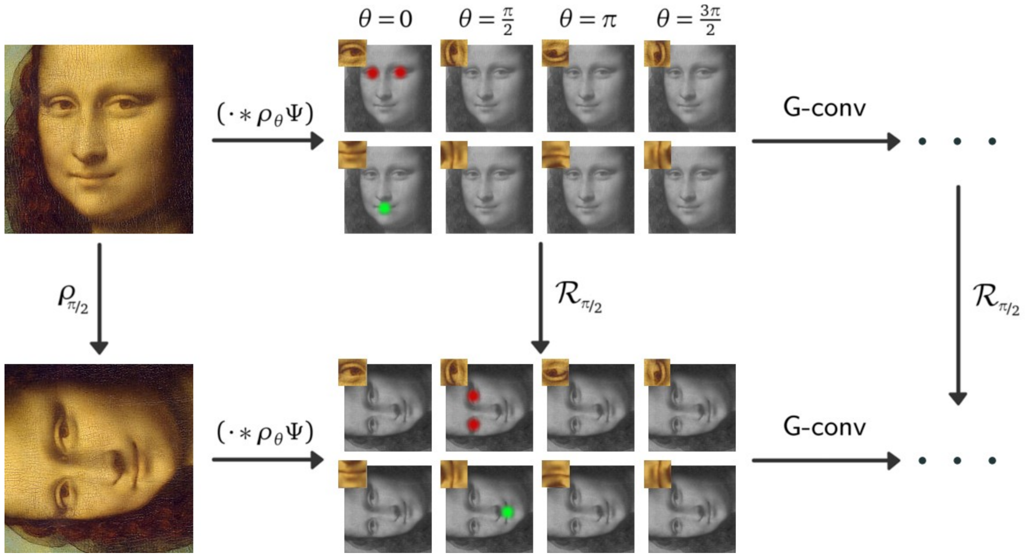
Group correlations

how to process lifted feature maps on $\mathrm{Aff}(G)$ further?

G-correlation: $\langle tg \blacktriangleright K \,|\, f \rangle_{\mathrm{Aff}(G)} \;:\; \mathrm{Aff}(G) \to \mathbb{R}$ with kernel on $\mathrm{Aff}(G)$

non-trivially equivariant

can be chained



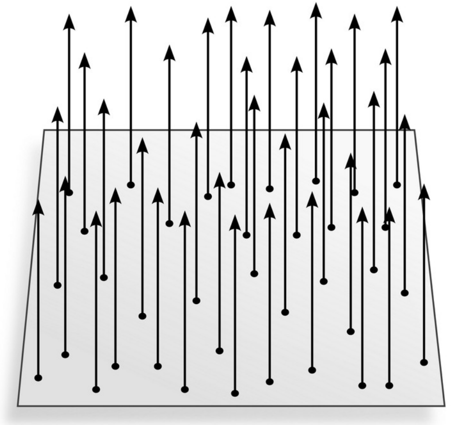
Relation to steerable CNNs

group-correlation feature maps are scalar functions on $\mathrm{Aff}(G) := (\mathbb{R}^d, +) \rtimes G \cong_{\text{top}} \mathbb{R}^d \times G$

equivalent to functions $\underbrace{\mathbb{R}^d \to \underbrace{\{\phi : G \to \mathbb{R}\}}_{\textit{regular rep} \text{ of } G}}_{\text{steerable feature field of type } \rho_{\text{reg}}}$

(the transformation laws match as well)

$\mathrm{Aff}(G) \cong_{\text{top}} \mathbb{R}^2 \times G$

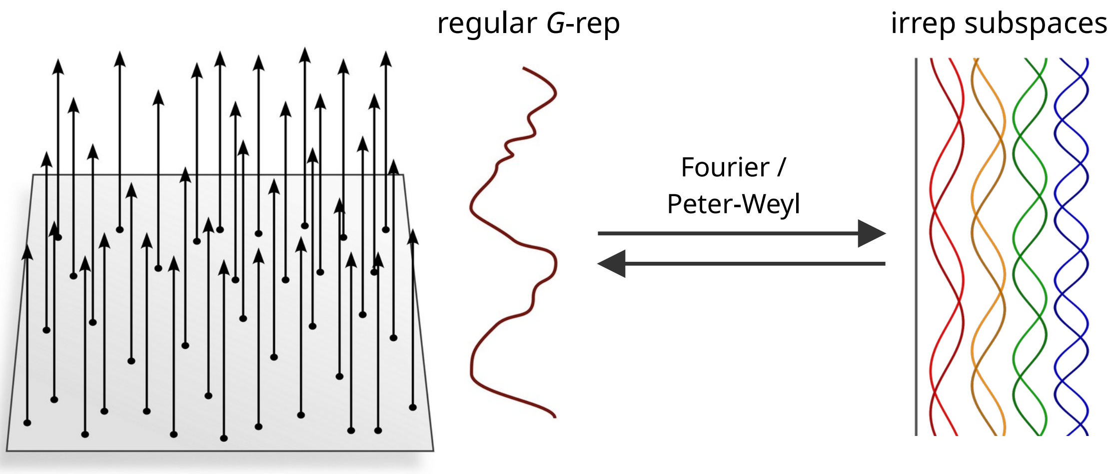


Relation to steerable CNNs

steerable CNNs allow for regular representation feature fields, but also other representations

they can in particular directly address *irrep subspaces*, which is not possible with group correlations

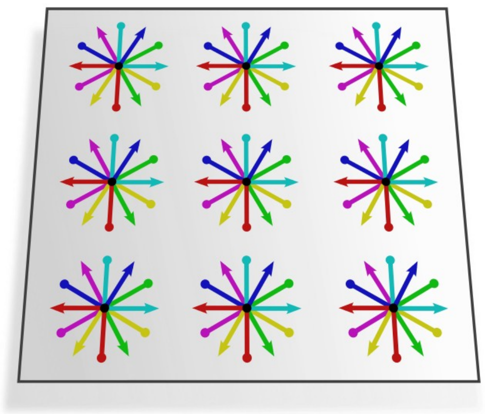
$$\mathrm{Aff}(G) \cong_{\mathrm{top}} \mathbb{R}^2 \times G$$



Equivariant convolutions on homogeneous spaces

group correlations and steerable CNNs generalize to arbitrary homogeneous spaces

$\mathrm{Aff}(\mathrm{SO}(2)) = \mathrm{SE}(2)$ as $\mathrm{SO}(2)$-bundle over $\mathbb{R}^2$



$\mathrm{SO}(3)$ as $\mathrm{SO}(2)$-bundle over S^2