Advanced Topics in Equivariance: Lecture 3
Machine Learning Summer School, Indonesia 2020

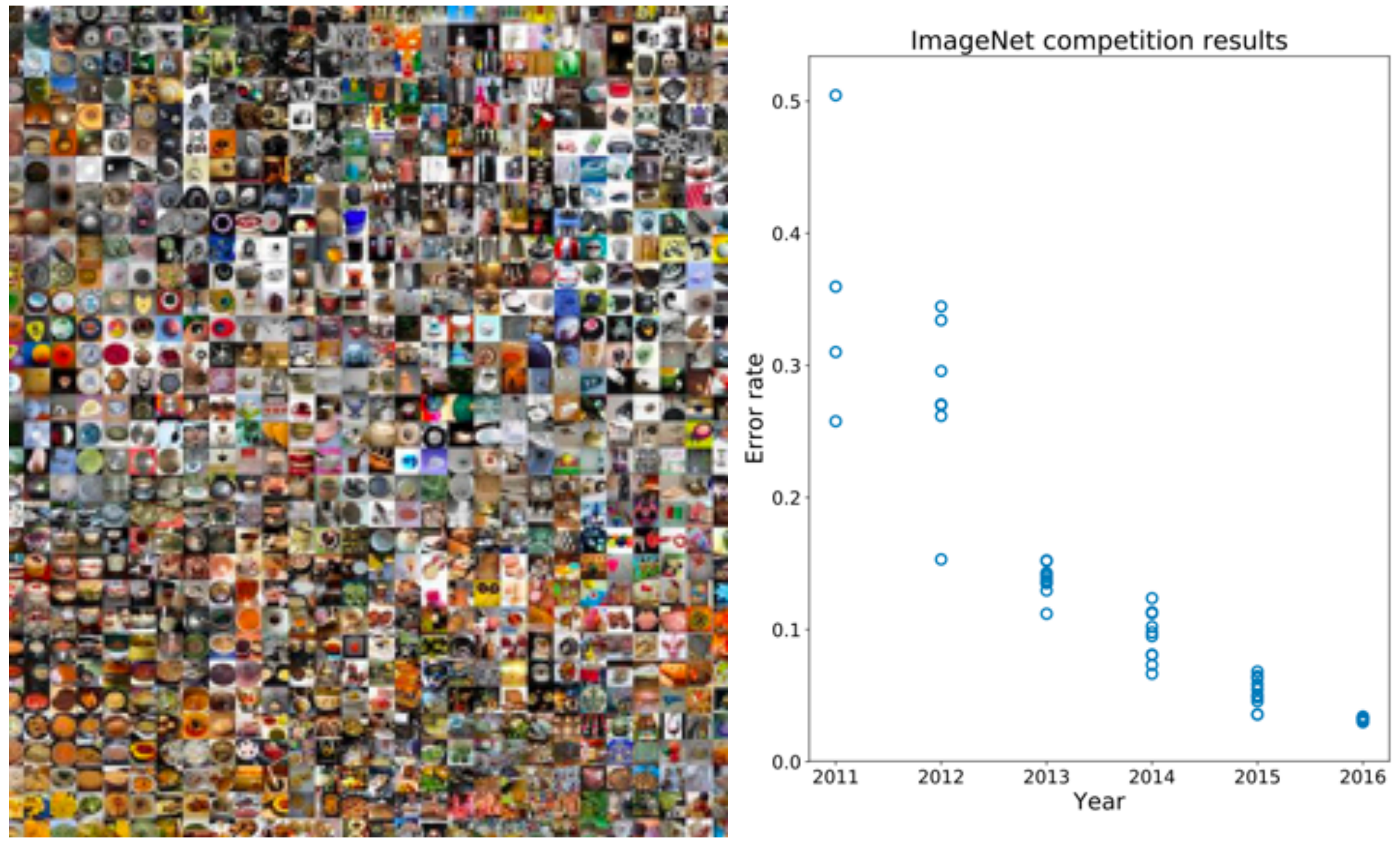
Daniel Worrall

Legong Dance

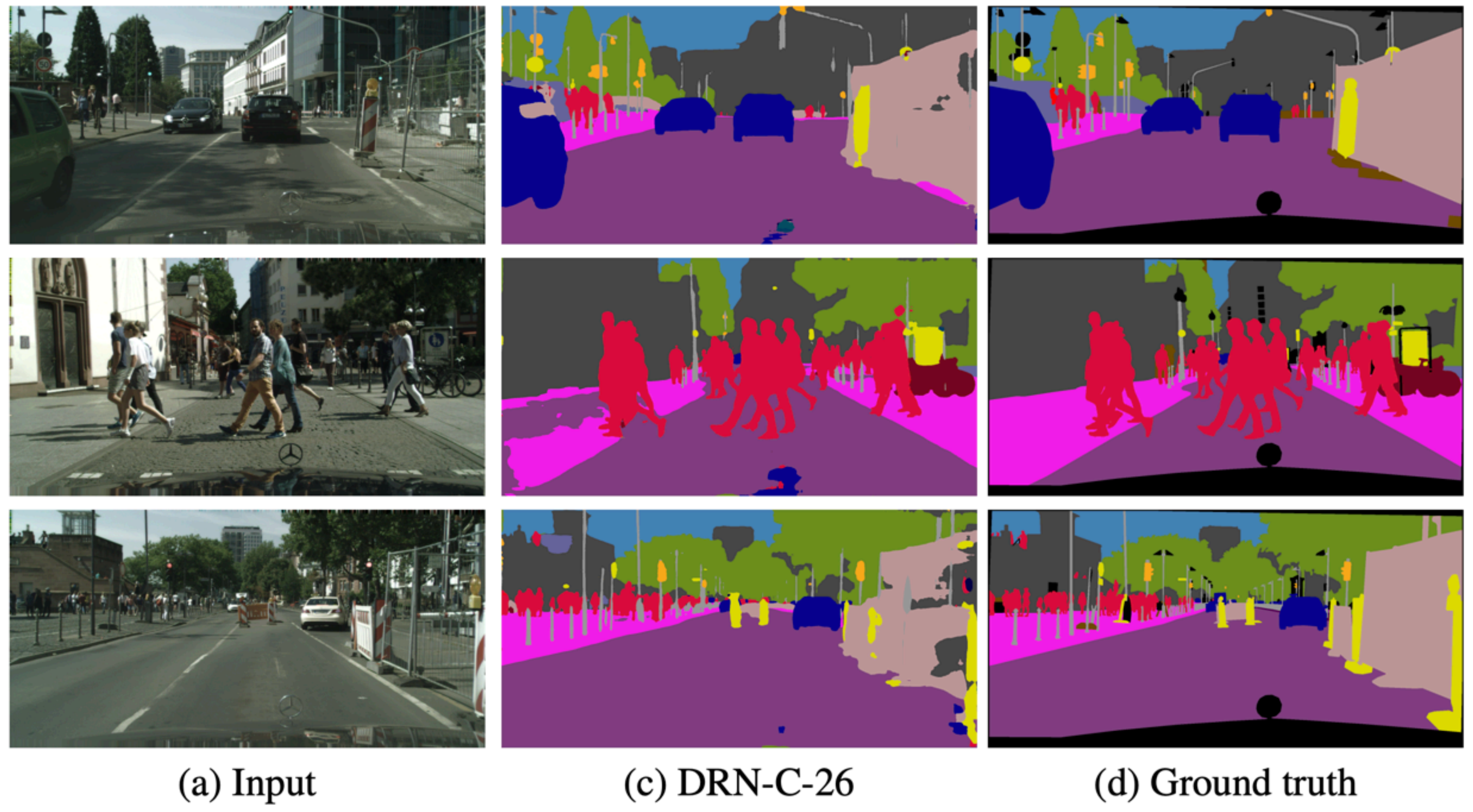
This lecture: Equivariance

- Grey Box Models
- Convolutions
- Invariance and Equivariance
- Group Theory
- Convolution Generalizations
    - Group Convolutions, Steerable Convolutions, Semigroup Correlation, Gauge Convolution

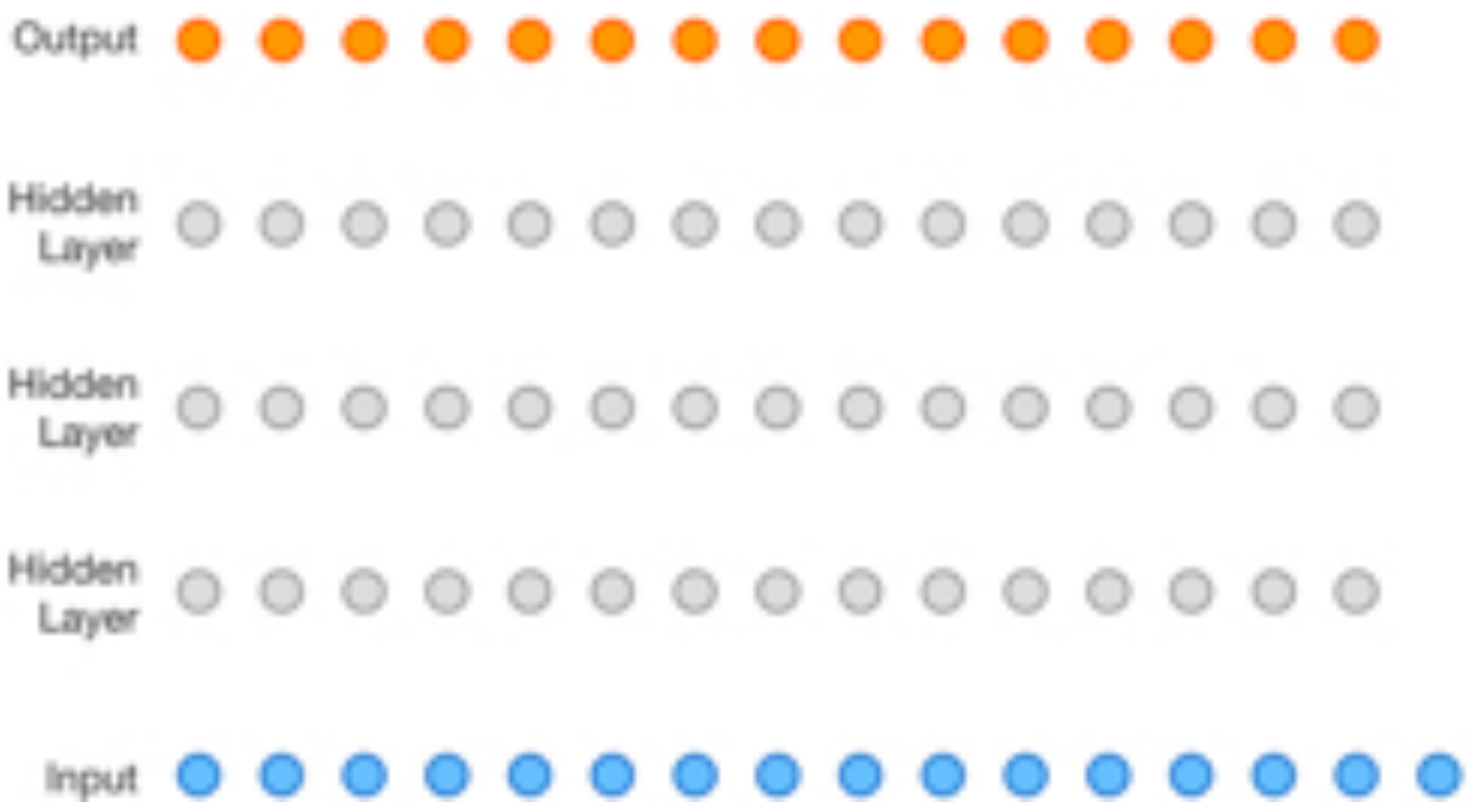
CNNs are responsible for many successes



ImageNet [figure from Wikipedia]



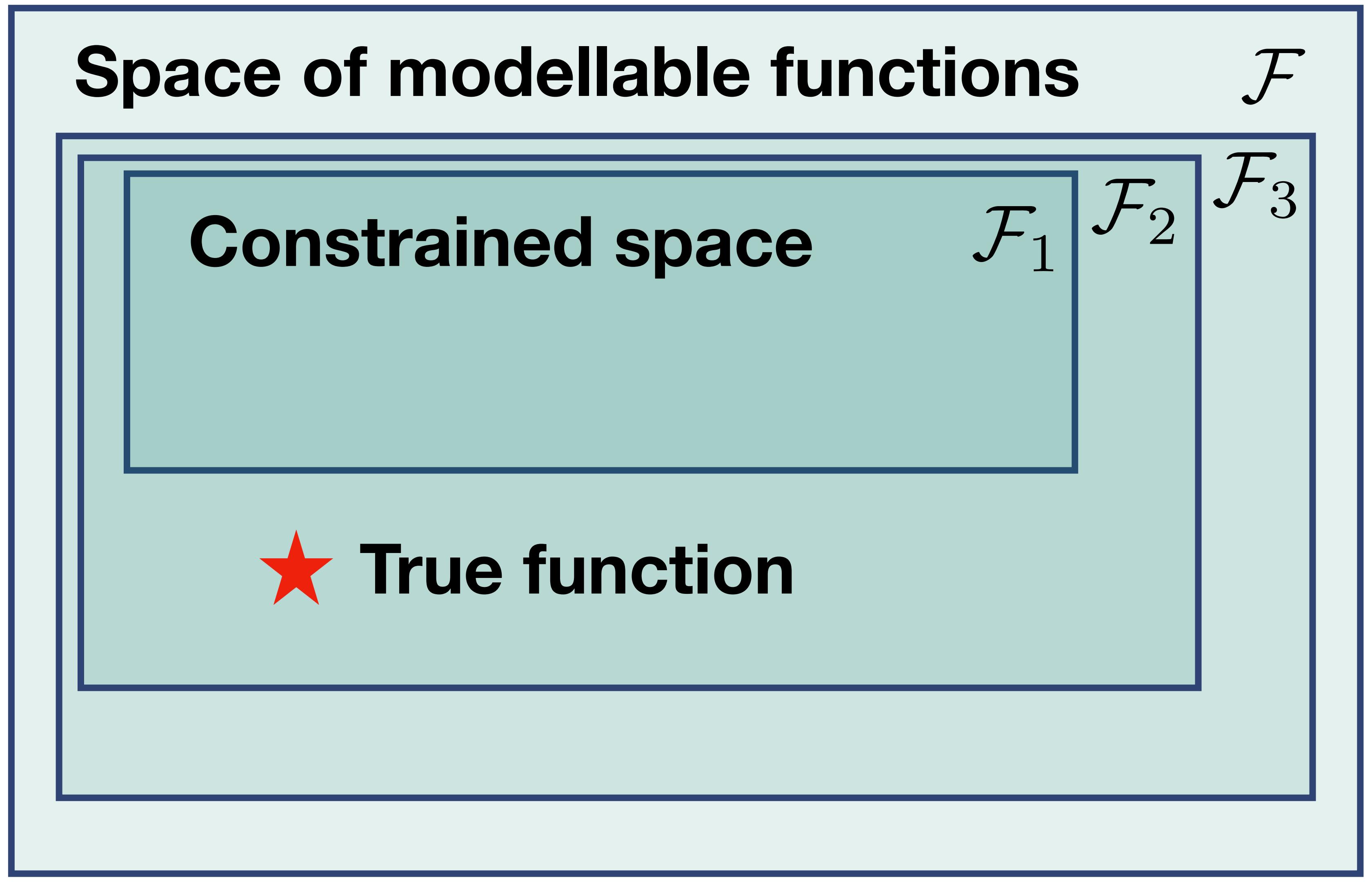
Dilated Residual Networks, Yu et al. (2017)

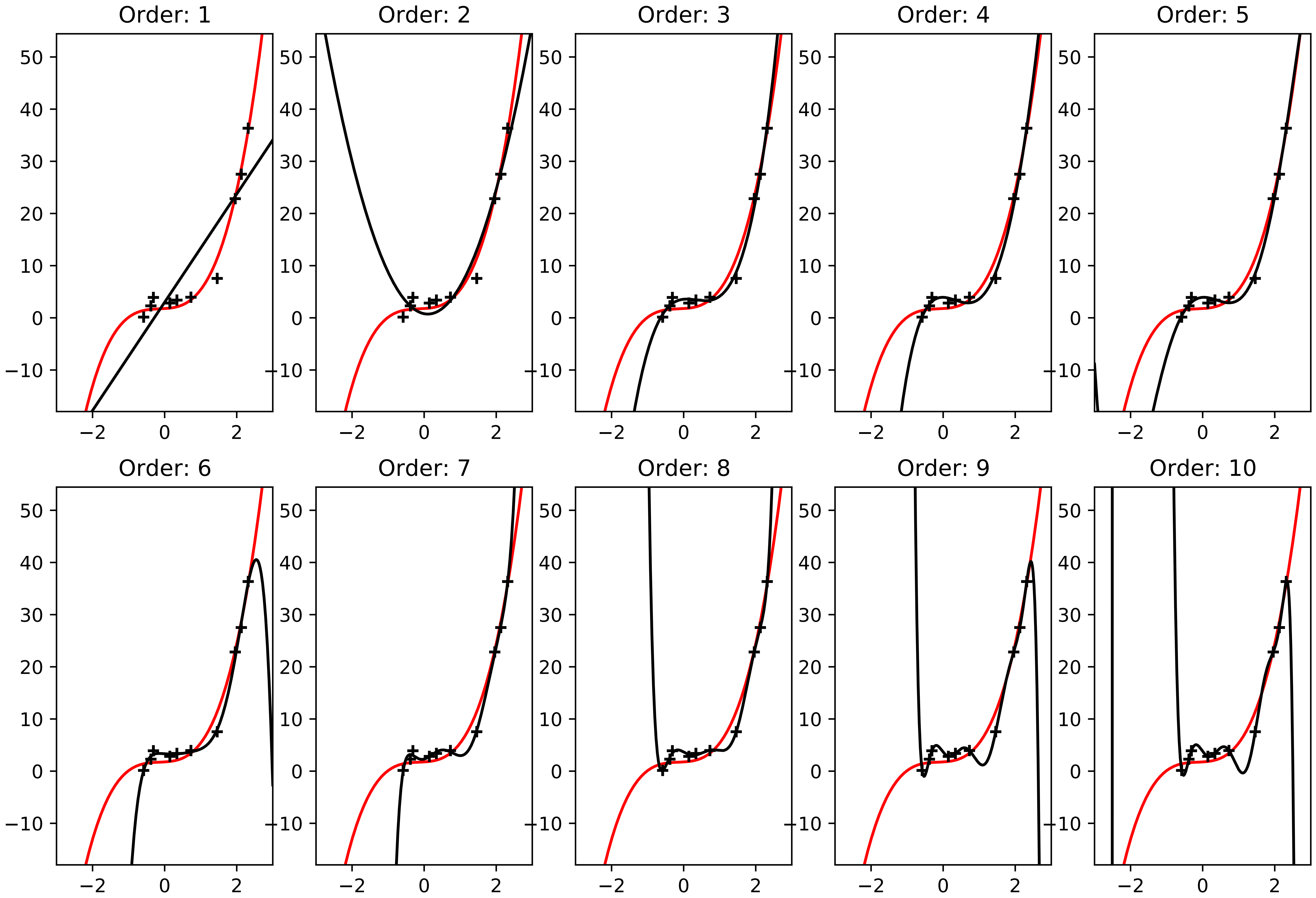


Inductive biases and overfitting

Remove *degrees of freedom* that do not aid generalization error.

- What assumptions can we make?
- How we encode these assumptions?
- What framework do we use?





Polynomial example

Polynomial fitting to a quintic. What order do we use?

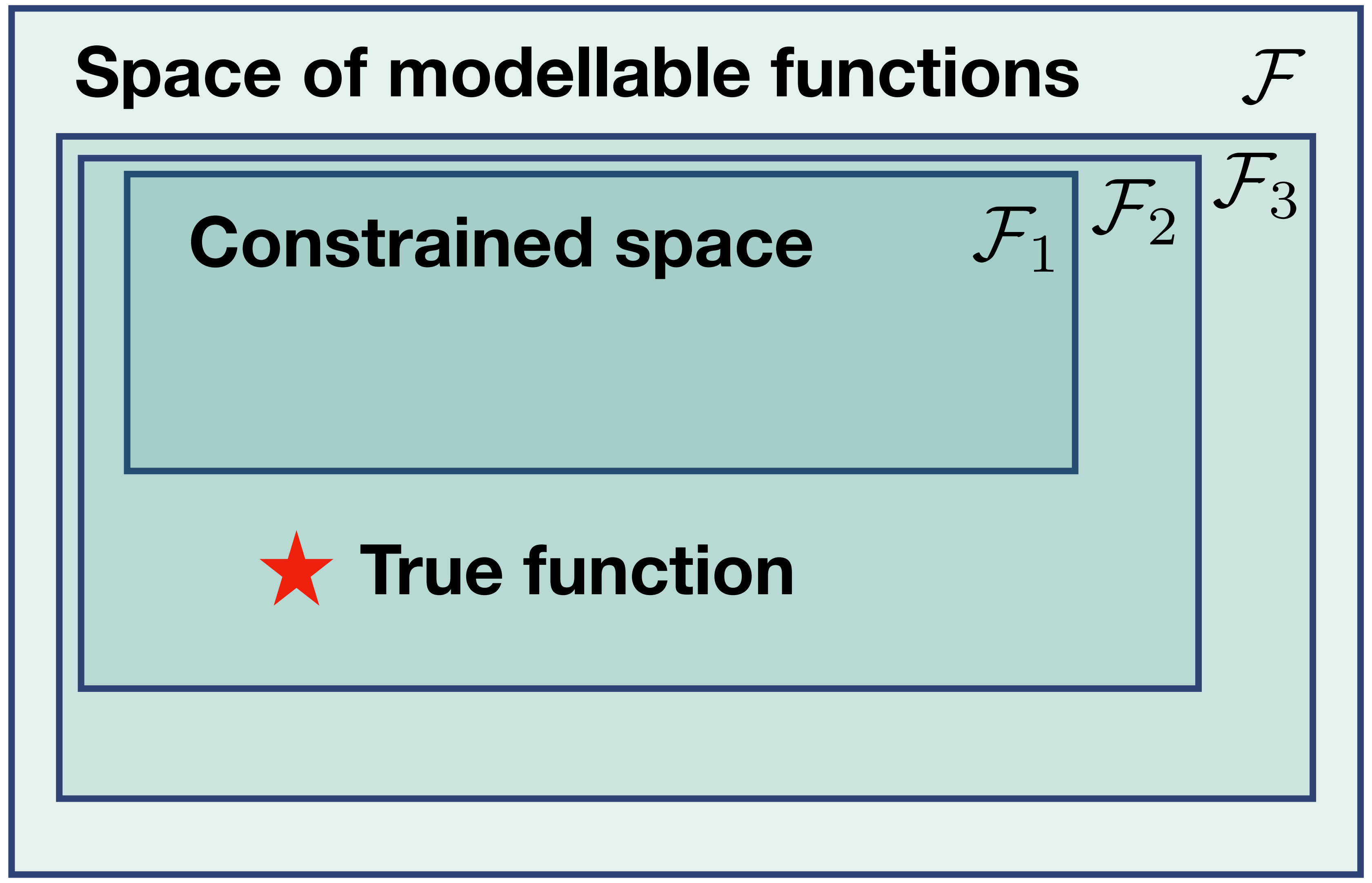
$$y = w_0 + w_1 x \ldots + w_5 x^5 + \epsilon$$

Try multiple orders

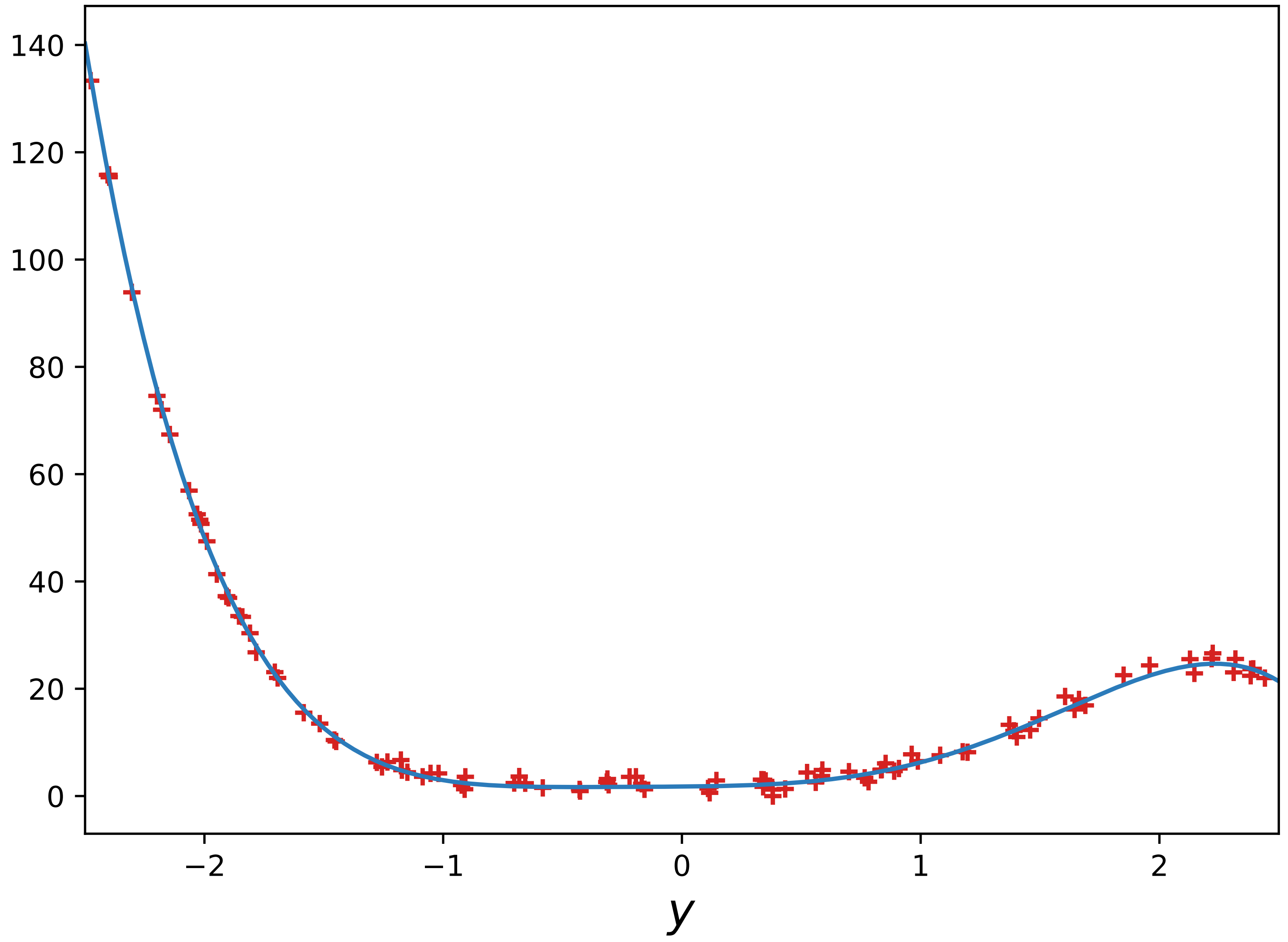
Unit Gaussian $\mathcal{N}(\mathbf{w}|\mathbf{0}, 10^2\mathbf{I})$

$$p(\mathbf{w}|\mathcal{D}, \sigma^2, \boldsymbol{\tau}) = \frac{p(\mathcal{D}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\boldsymbol{\tau})}{p(\mathcal{D}|\boldsymbol{\tau})}$$

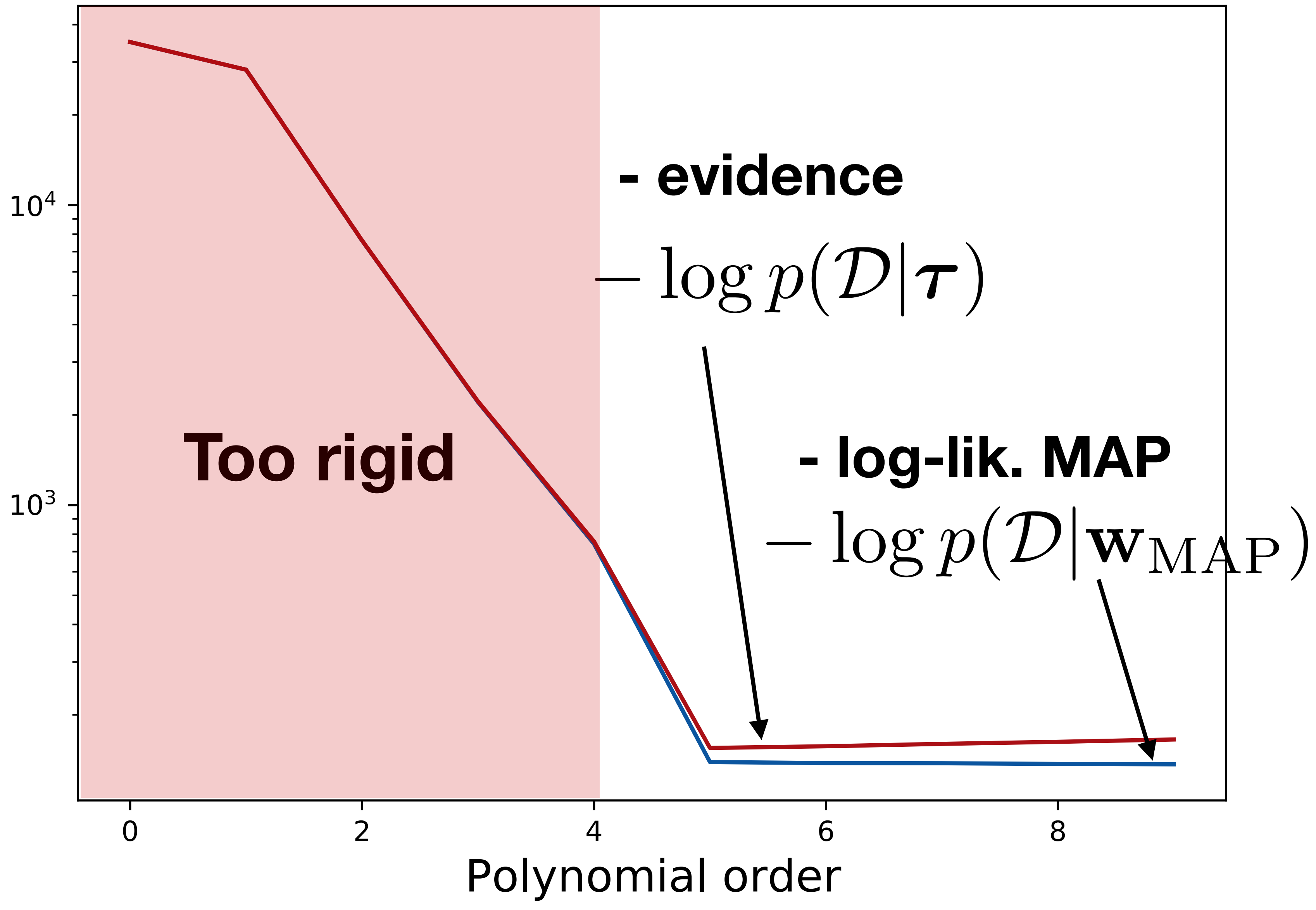
$$\mathbf{w}_{\mathrm{MAP}} = \arg\max_{\mathbf{w}} p(\mathbf{w}|\mathcal{D}, \boldsymbol{\tau})$$



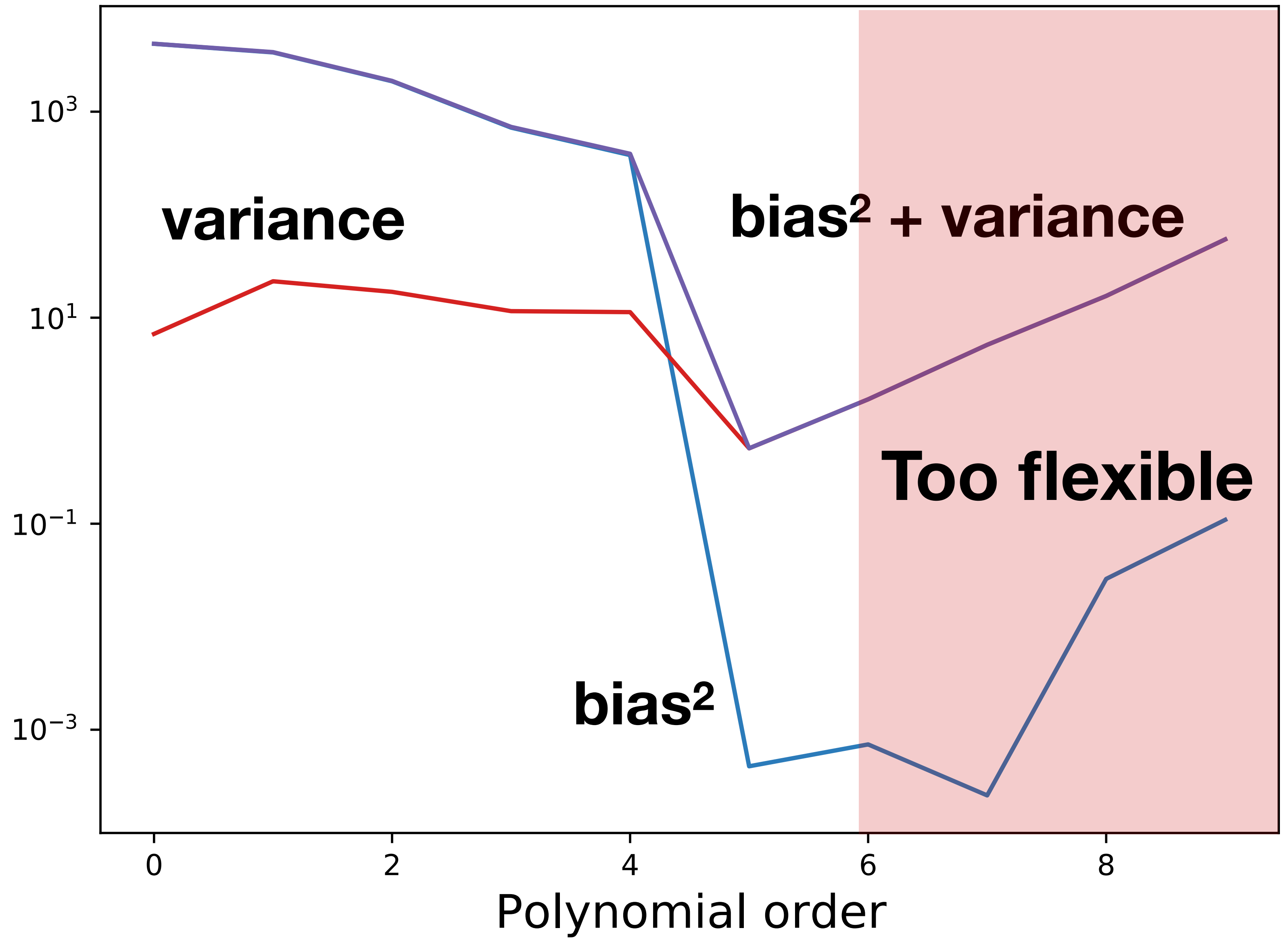
Original quintic polynomial



Log-likelihood and evidence



Bias—variance decomposition



Inductive Biases

White box models

Grey box models

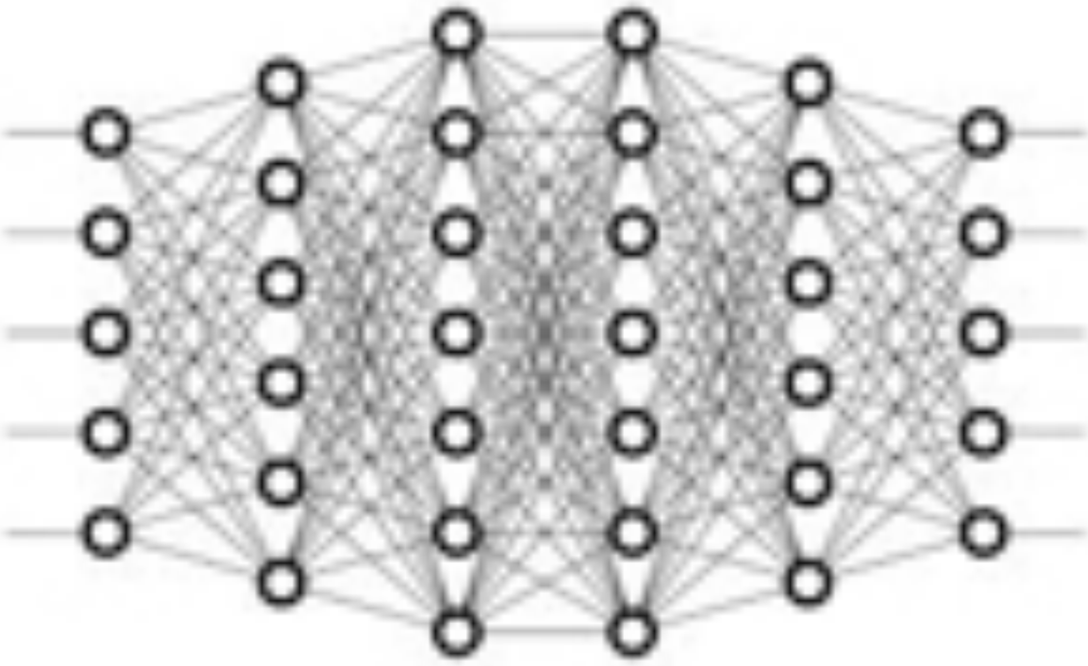
Most ML models sit here.

Regression
- Polynomial order
- Likelihood and prior distribution

Classification
- Sigmoid function
- Linear features
- ML or MAP?

Deep learning
- Number of layers
- Nonlinearities
- Optimizer
- **Structure: Convolutions**
- …

Black box models



Equivariance

Batik making

For images we use *convolutions*.

Convolutions use:

1. Translational weight-tying
2. Small reception fields

Drop-in replacement for the linear layer.

Indeed, it is just a linear layer!

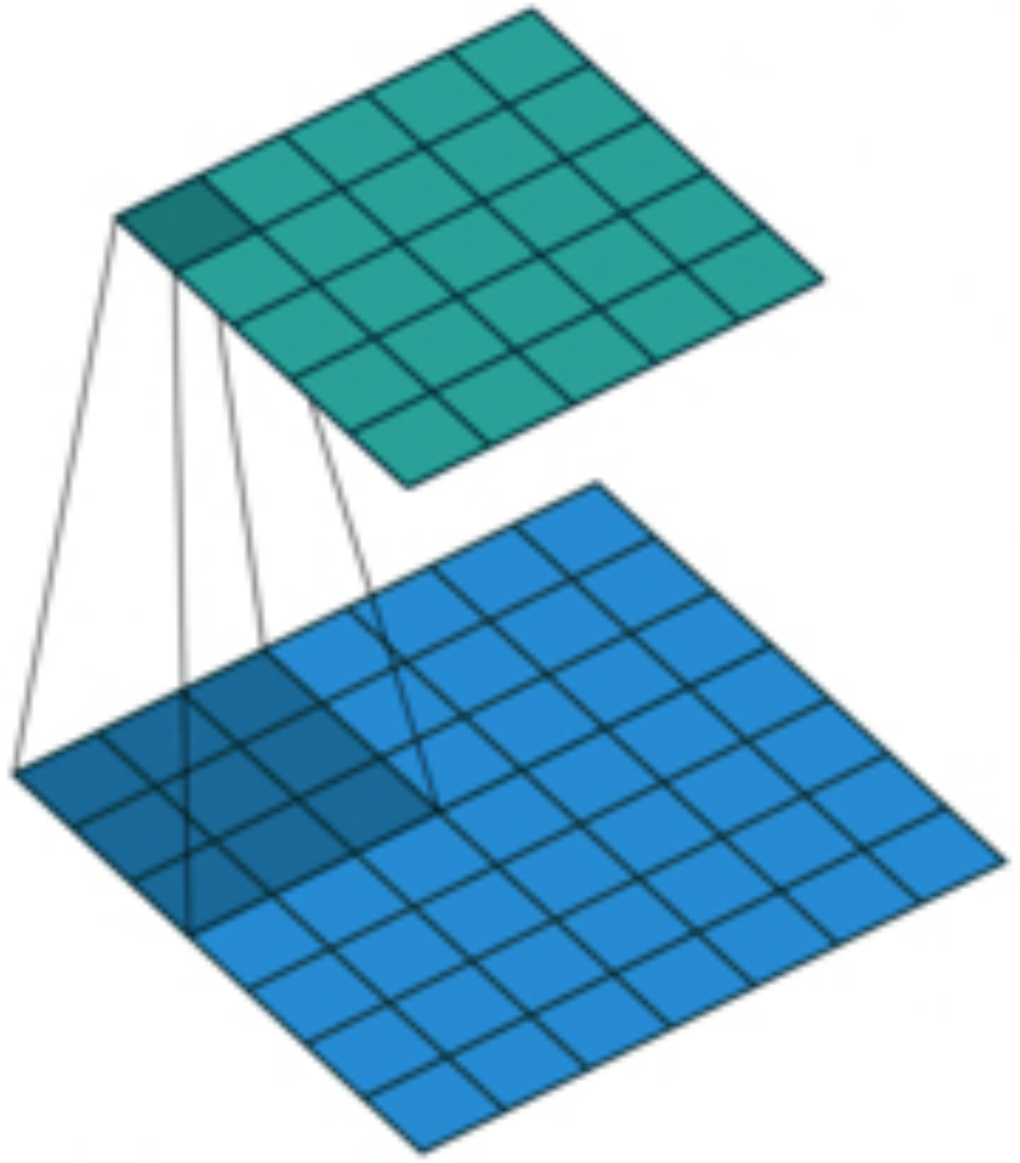
Standard convolution

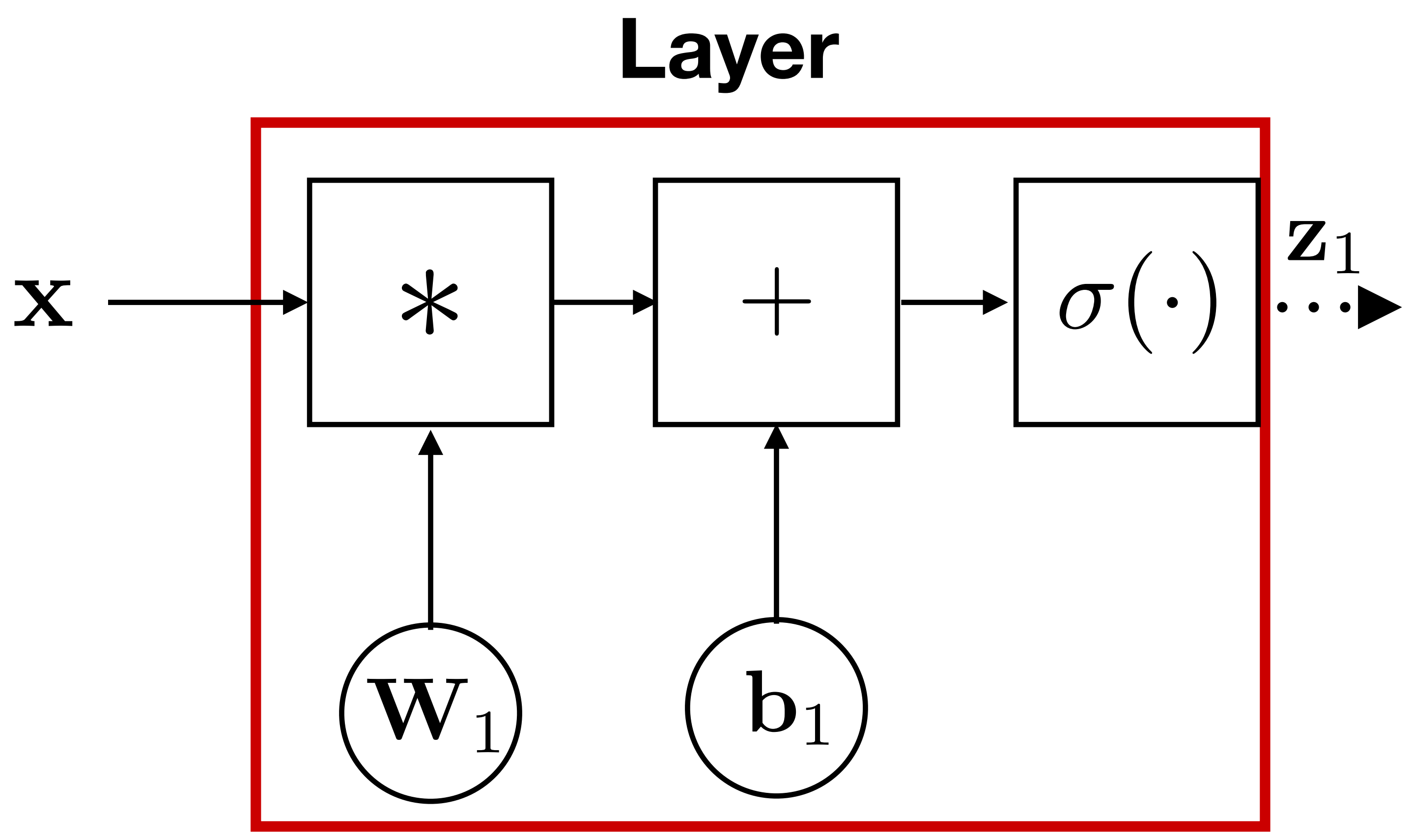
$$[\mathbf{F} * \mathbf{W}](\mathbf{x}) = \sum_{\mathbf{y} \in \mathbb{Z}^2} \mathbf{F}(\mathbf{y})\mathbf{W}(\mathbf{y} - \mathbf{x})$$

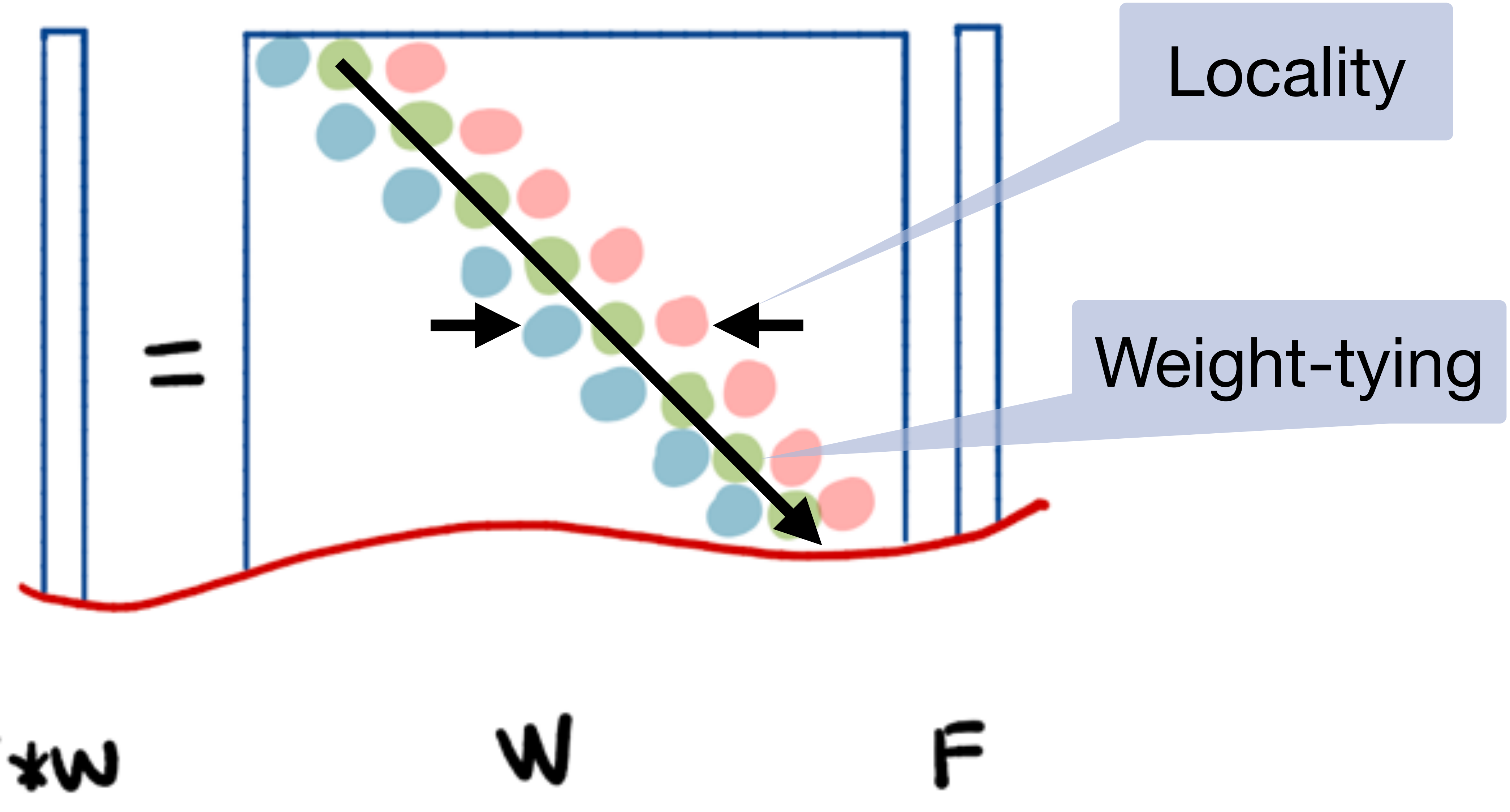
Filter

Feature map

e.g. LeCun et al. (1991)

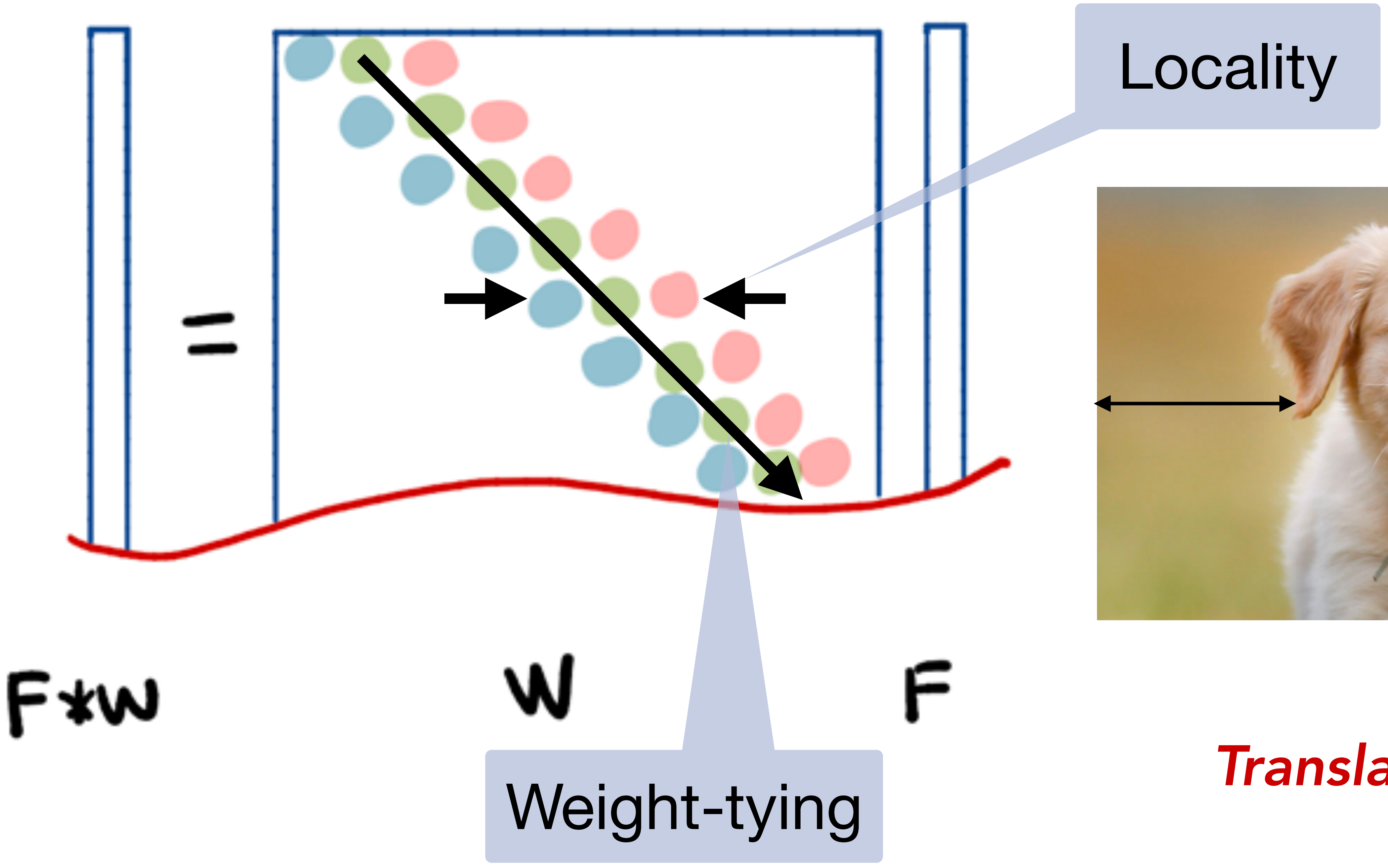


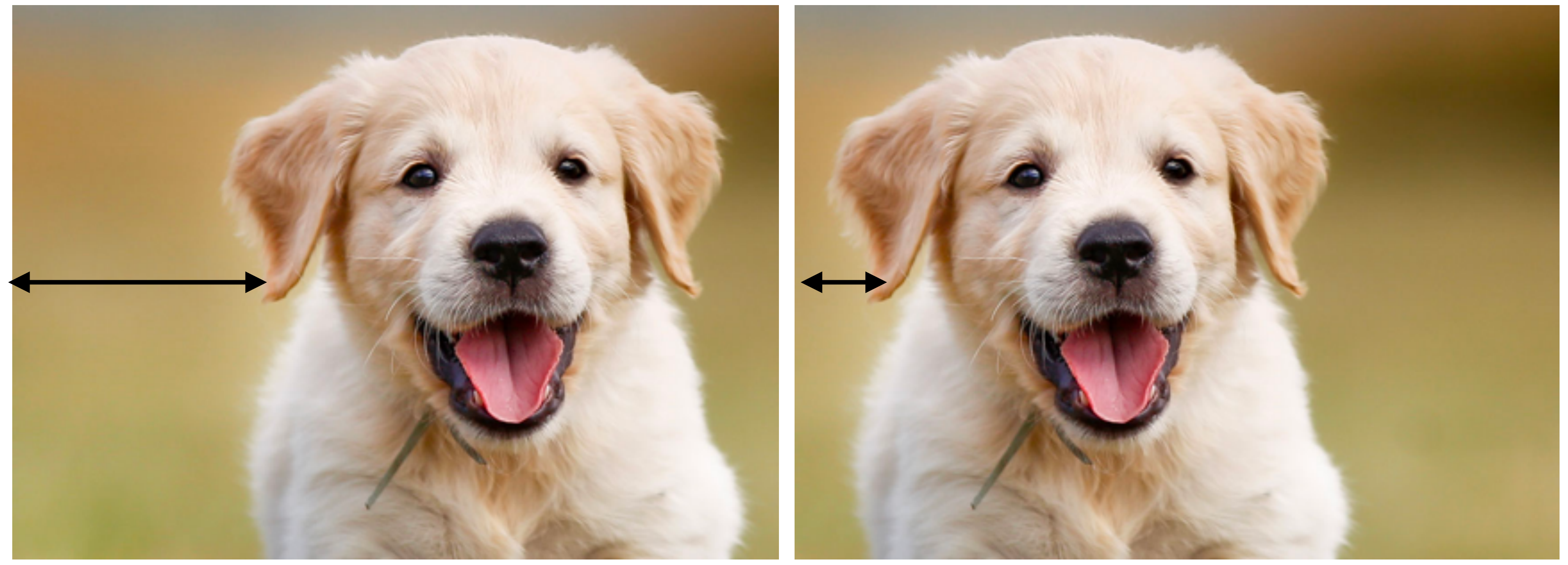




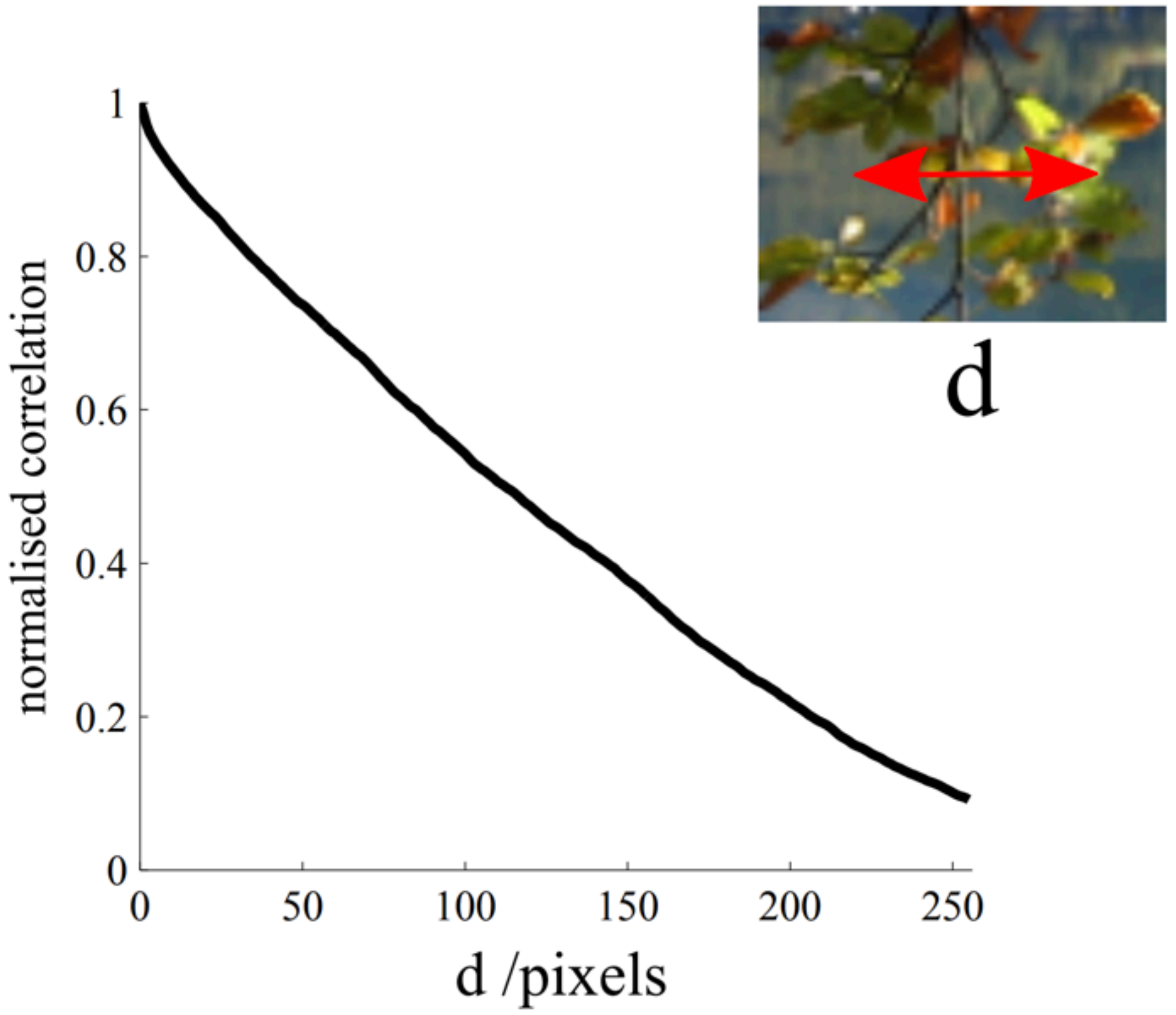
Convolutions can be “reshaped” into matrix-vector products, revealing the weight-tying and locality

Convolutions are special





Translational invariance of classification
Property of the task



Locality of *pixel statistics*
Property of the data

Fewer learnable parameters

Parameter efficient: same function — fewer parameters
Statistically efficient: need less data
Computationally efficient: need less GPU time

Why learn translational weight-tying when we can build it in?

What about rotation, scaling, warps, shears, color jitter, blur, …?

https://www.healthypawspetinsurance.com/Images/V3/DogAndPuppyInsurance/Dog_CTA_Desktop_HeroImage.jpg
http://www.gatsby.ucl.ac.uk/~turner/teaching/4g3/2013/stats-recep-field.pdf

Convolutional variants

Semigroup correlation →
- Deep Scale Space

Semigroup correlation → Group convolution

Gauge convolution → Group convolution → Steerable convolution

Steerable convolution →
- Harmonic network
- Spherical convolution
- Steerable SE(3) network
- Tensor Field Network

Gauge convolution →
- Icosahedral CNN
- Gauge Equivariant Mesh Convolution

Group convolution → Unitary convolution

Unitary convolution →
- Learning to convolve

Unitary convolution → Regular-representation conv.

Regular-representation conv. →
- **Translational convolution**
- Group CNN
- HexaConv
- CubeNet
- Scale Convolution

Regular-representation conv. → Tangent space convolutions

Tangent space convolutions → Icosahedral CNN, Gauge Equivariant Mesh Convolution

Tangent space convolutions →
- LieConv

Symmetry: Invariance

Set of input transformations leaving f invariant

Image

$$f(\mathbf{I}) = f(\mathcal{T}_\theta[\mathbf{I}])$$

Function/feature mapping

Transformation

Notation

Translation

$$\mathcal{T}_\theta[\mathbf{I}](\mathbf{x}) = \mathbf{I}(\mathbf{x} - \boldsymbol{\theta})$$

Rotation

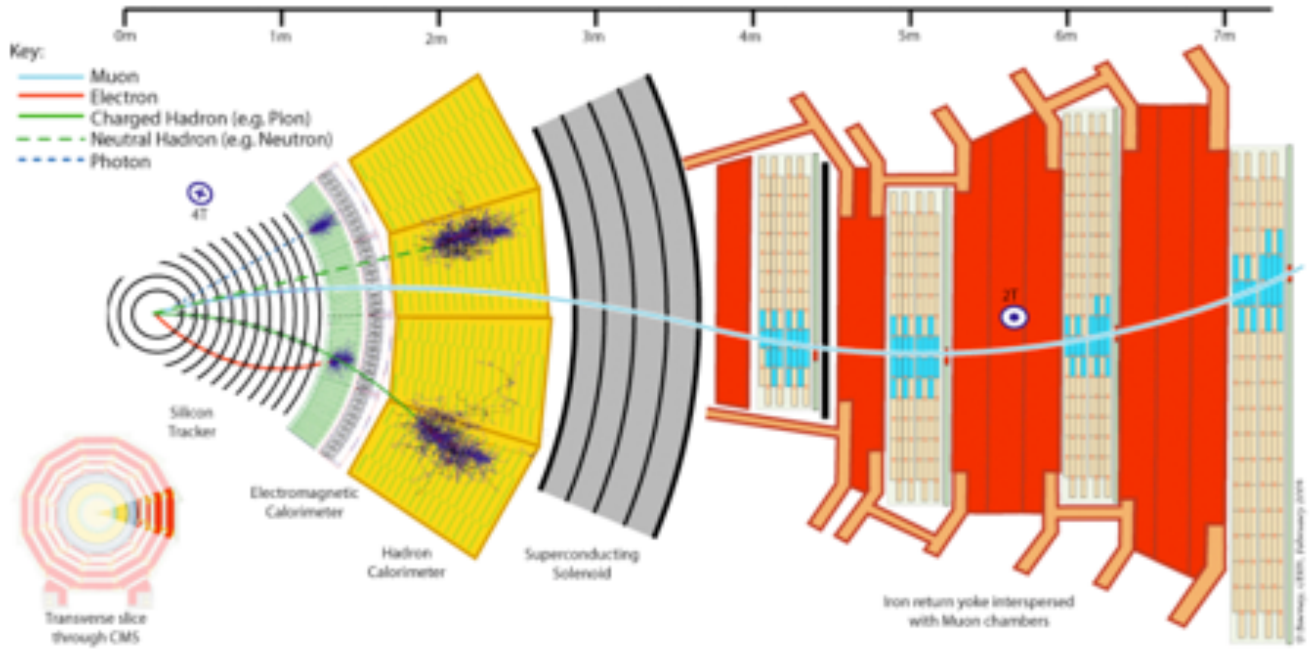
$$\mathcal{T}_\theta[\mathbf{I}](\mathbf{x}) = \mathbf{I}(\mathbf{R}_{\boldsymbol{\theta}}^{-1}\mathbf{x})$$

Data whitening

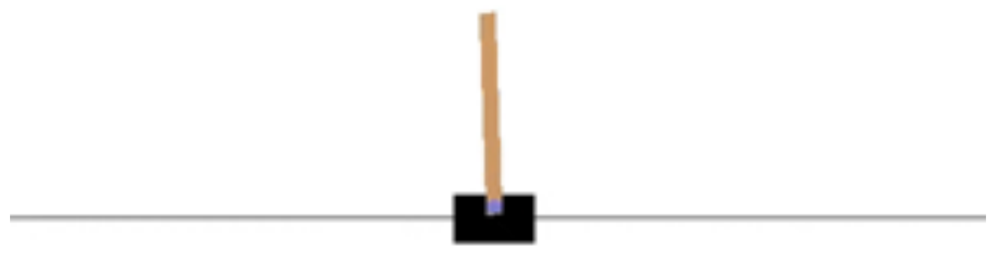
$$\mathcal{T}[\mathbf{I}] = (\mathbf{I} - \boldsymbol{\mu})/.\boldsymbol{\sigma}^{-1}$$

Symmetry is a property of tasks, not of data!

Classification



Signal discovery/detection



Dynamical Systems

Disentangling (cocktail party)

Symmetry: Equivariance

Different representations of same transformation

$$\mathcal{S}_\theta[f](\mathbf{I}) = f(\mathcal{T}_\theta[\mathbf{I}])$$

Transformation in feature space

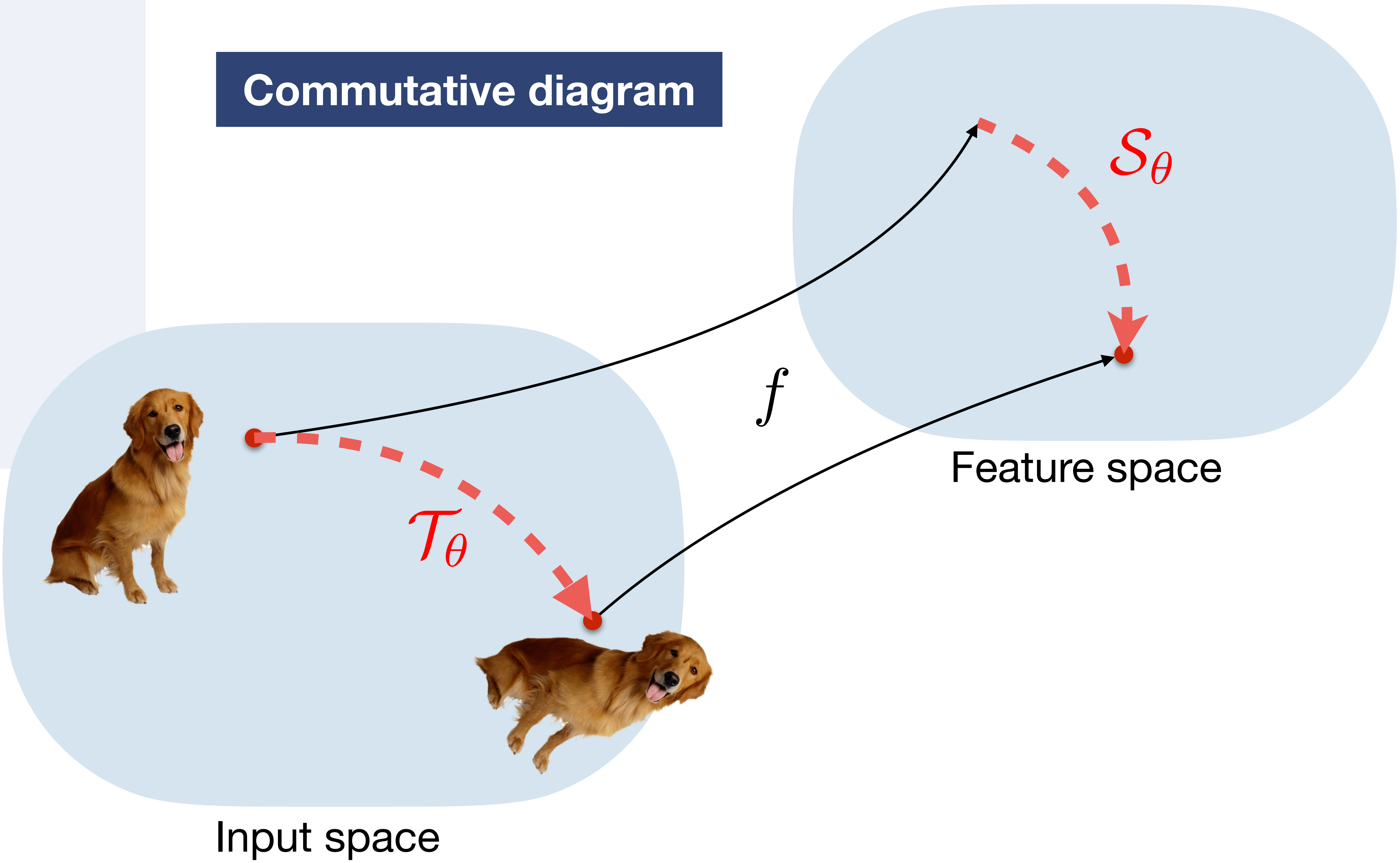
Invariance

$$\mathcal{S}_{\boldsymbol{\theta}} = \mathrm{Id}$$

Convolution (and correlation)

$$\underbrace{[\mathbf{I} * \mathbf{W}](\mathbf{x} - \boldsymbol{\theta})}_{\mathcal{S}_{\boldsymbol{\theta}}[\mathbf{I} * \mathbf{W}](\mathbf{x})} = [\mathcal{T}_{\boldsymbol{\theta}}[\mathbf{I}] * \mathbf{W}](\mathbf{x})$$

Commutative diagram



Theorem [Kondor & Trivedi (2018), Cohen et al. (2020a), Bekkers (2020)]:

Group convolution is the only linear map* equivariant to group-structured** transformations

*Linear, bounded map between measurable homogeneous spaces **Countable groups and Lie groups

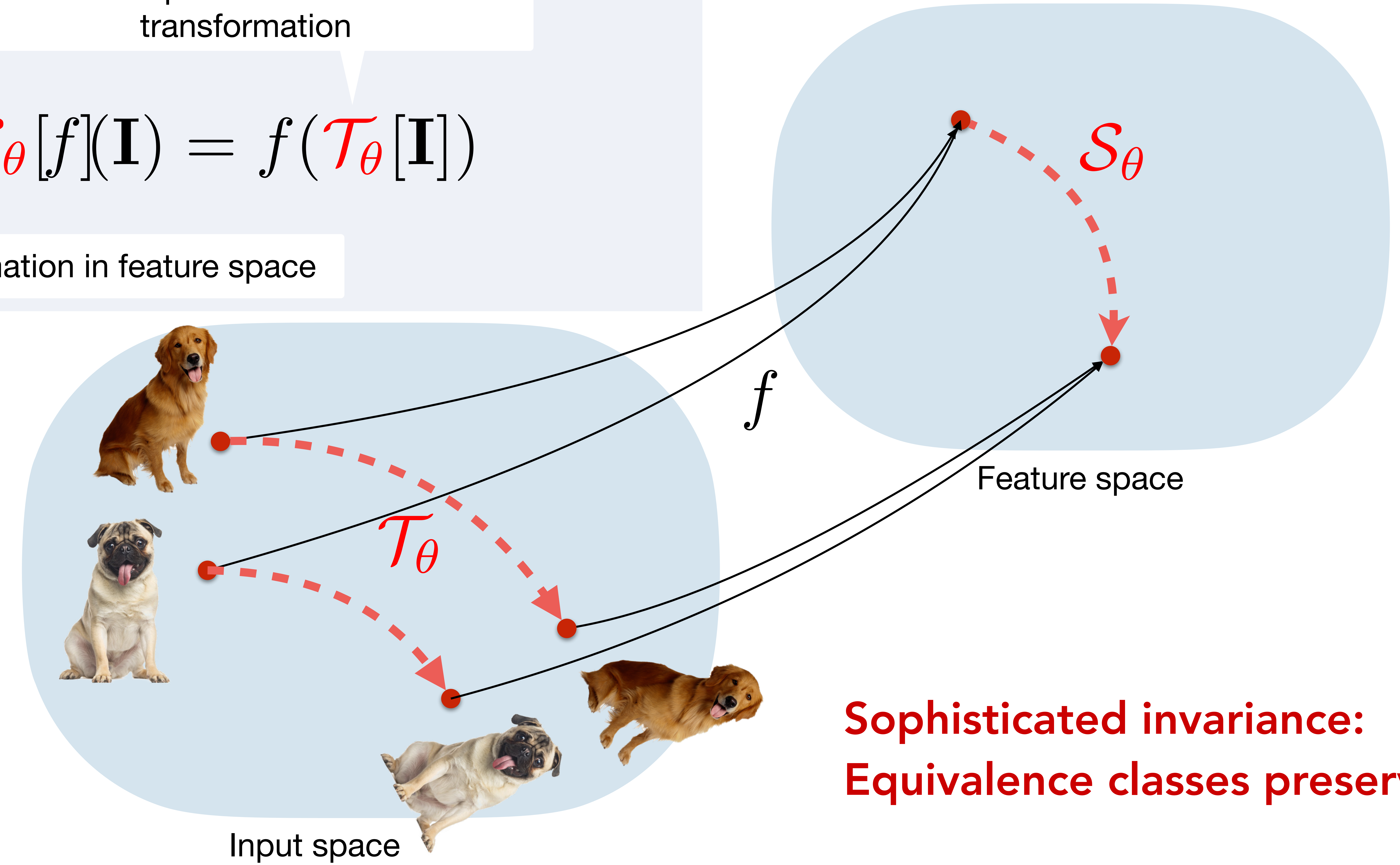
Symmetry: Equivariance

Different representations of same transformation

$$\mathcal{S}_\theta[f](\mathbf{I}) = f(\mathcal{T}_\theta[\mathbf{I}])$$

Transformation in feature space

Commutative diagram



Sophisticated invariance:
Equivalence classes preserved

Translational Equivariance Proof

$$[\mathcal{T}_{\mathbf{t}}[\mathbf{I}] * \mathbf{W}](\mathbf{x}) = \sum_{\mathbf{y} \in \mathbb{Z}^2} \mathcal{T}_{\mathbf{t}}[\mathbf{I}](\mathbf{y}) \mathbf{W}(\mathbf{y} - \mathbf{x})$$

Write out convolution in full

$$= \sum_{\mathbf{y} \in \mathbb{Z}^2} \mathbf{I}(\mathbf{y} - \mathbf{t}) \mathbf{W}(\mathbf{y} - \mathbf{x})$$

Expand translation operator

$$= \sum_{\mathbf{y}' \in \mathbb{Z}^2} \mathbf{I}(\mathbf{y}') \mathbf{W}(\mathbf{y}' - (\mathbf{x} - \mathbf{t}))$$

Change of variables
$\mathbf{y}' = \mathbf{y} - \mathbf{t}$

$$= [\mathbf{I} * \mathbf{W}](\mathbf{x} - \mathbf{t}) \quad = \mathcal{T}_{\mathbf{t}}[\mathbf{I} * \mathbf{W}](\mathbf{x})$$

Oh look, it’s a convolution!

Hints

$$[\mathbf{I} * \mathbf{W}](\mathbf{x}) = \sum_{\mathbf{y} \in \mathbb{Z}^2} \mathbf{I}(\mathbf{y}) \mathbf{W}(\mathbf{y} - \mathbf{x})$$

$$\mathcal{T}_{\mathbf{t}}[\mathbf{I}](\mathbf{x}) = \mathbf{I}(\mathbf{x} - \mathbf{t})$$

A Little Group Theory

Group theory
Mathematical abstraction modeling compositional structure.

Can be used to model transformations

Abstraction

1. **Closure**: compositions well-behaved

$$g \circ h \in G$$

Set of transformations

Composition

2. **Associativity**: brackets unnecessary

$$(g \circ h) \circ k = g \circ (h \circ k) = g \circ h \circ k$$

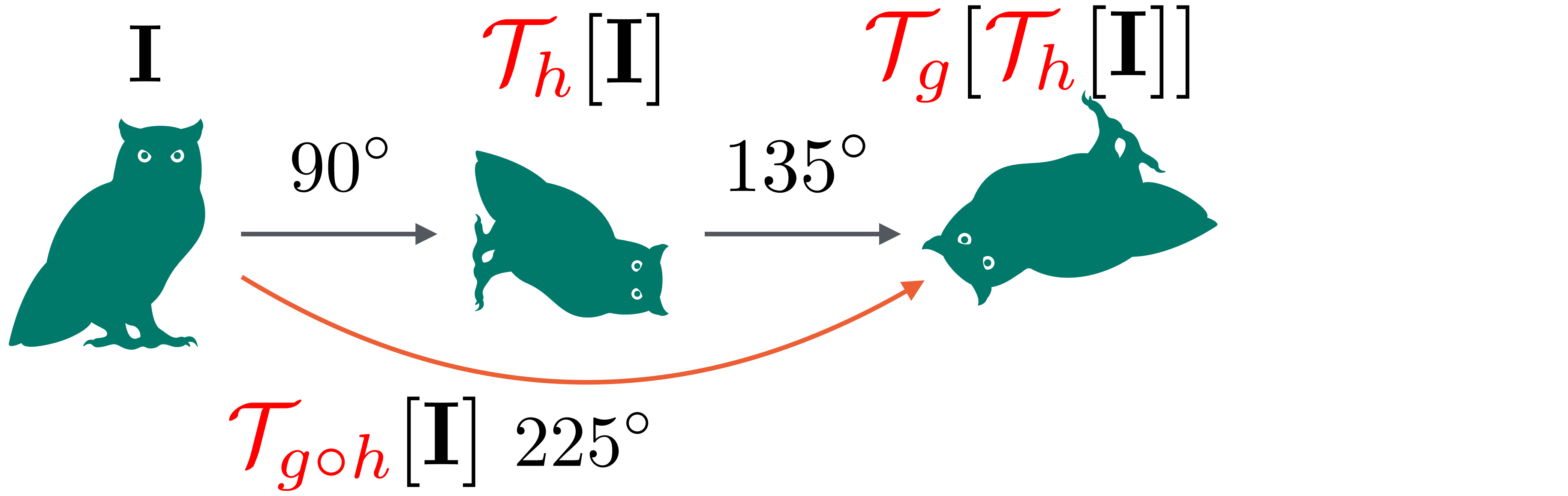
3. **Identity**: “do nothing” transformation

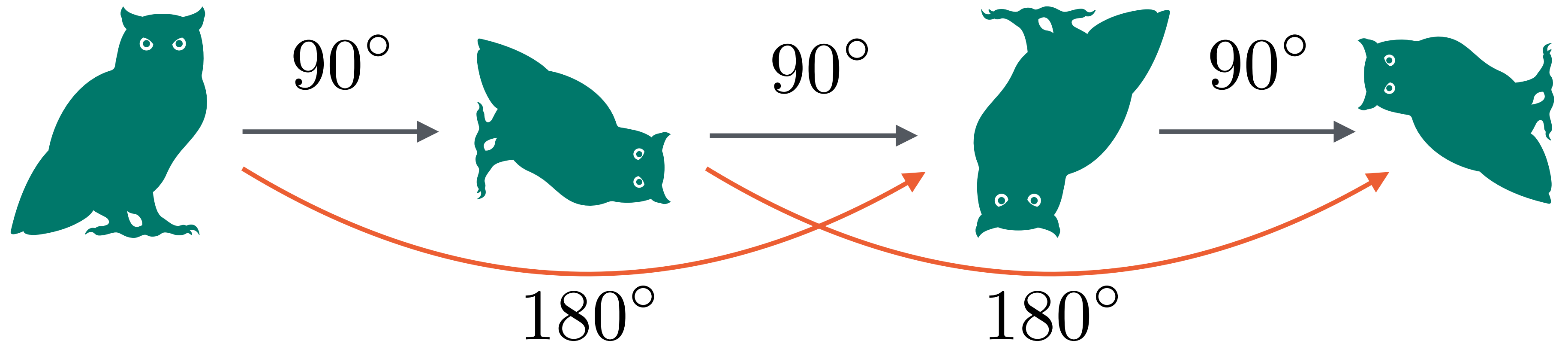
$$e \circ g = g \circ e = g$$

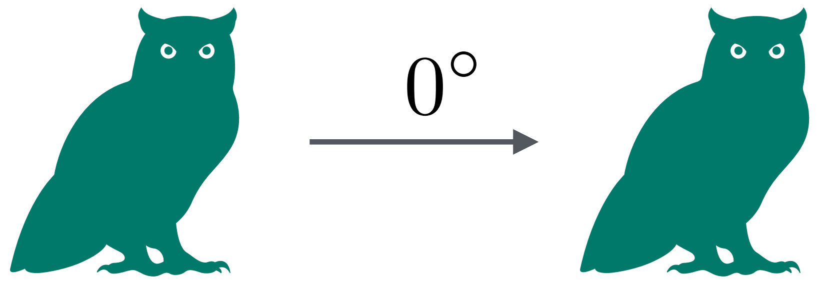
4. **Invertibility**: transformation reversible

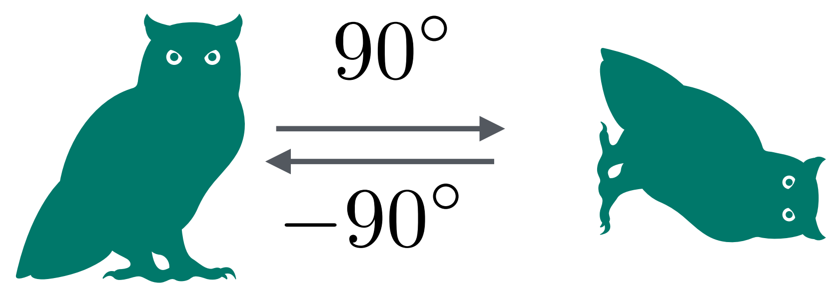
$$g \circ g^{-1} = g^{-1} \circ g = e$$

Action/Transformation









Convolutional variants: Regular-representation convolutions

Semigroup correlation

- Deep Scale Space

Gauge convolution

Group convolution

Steerable convolution

- Harmonic network
- Spherical convolution
- Steerable SE(3) network
- Tensor Field Network

- Icosahedral CNN
- Gauge Equivariant Mesh Convolution

Unitary convolution

- Learning to convolve

- **Translational convolution**
- Group CNN
- HexaConv
- CubeNet
- Scale Convolution

Tangent space convolutions

Regular-representation conv.

- LieConv

The Regular Group Convolution

Standard convolution

Inner product

$$[\mathbf{F} * \mathbf{W}](\mathbf{x}) = \sum_{\mathbf{y} \in \mathbb{Z}^2} \mathbf{F}(\mathbf{y})\mathbf{W}(\mathbf{y} - \mathbf{x})$$

Translations

Group convolution

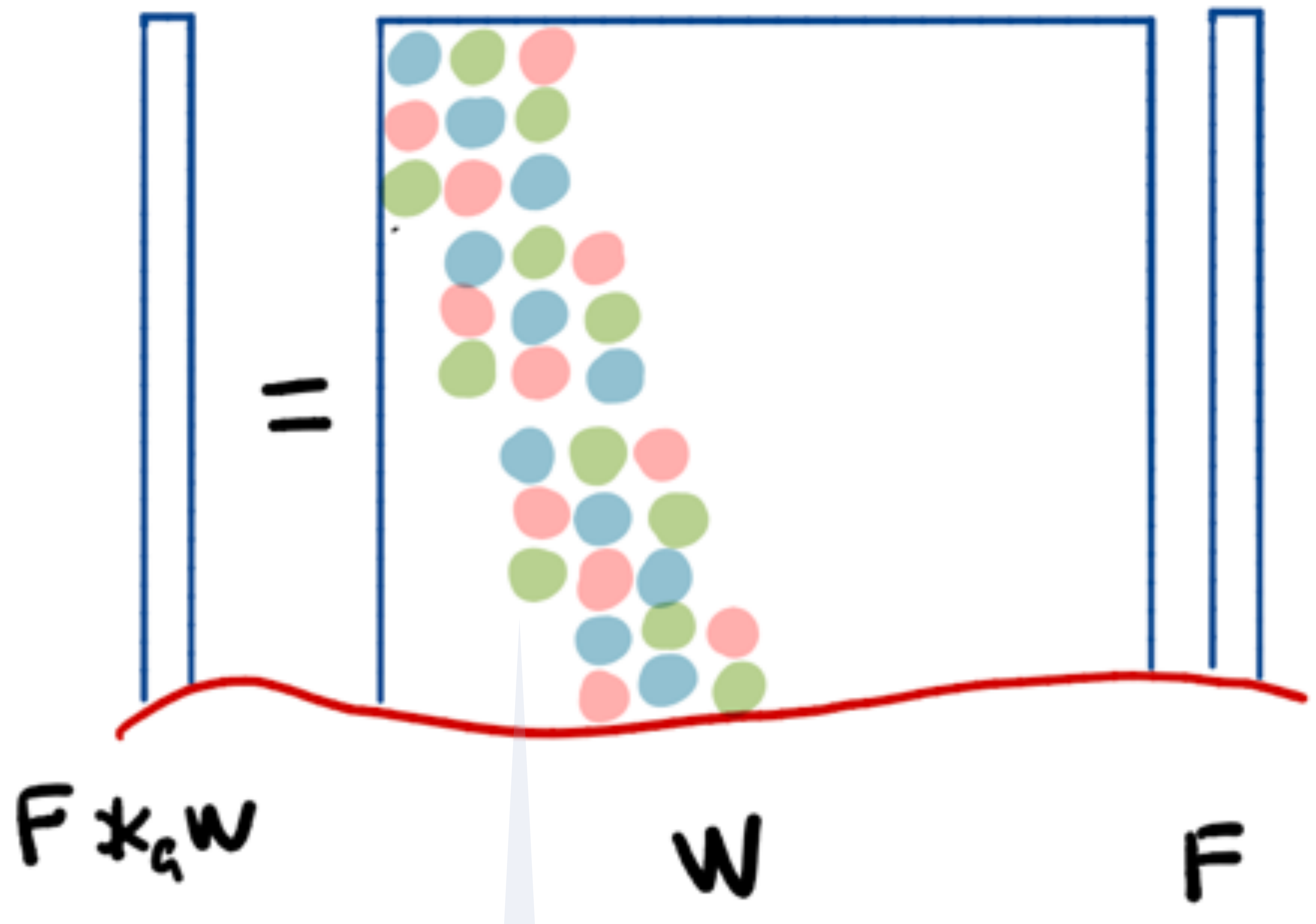
Inner product

$$[\mathbf{F} * \mathbf{W}](\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}} \mathbf{F}(\mathbf{y})\mathbf{W}(\mathcal{T}_{\mathbf{x}}^{-1}[\mathbf{y}])$$

$\mathbf{x}$ is transformation parameter

Transformations

Adjusted domain

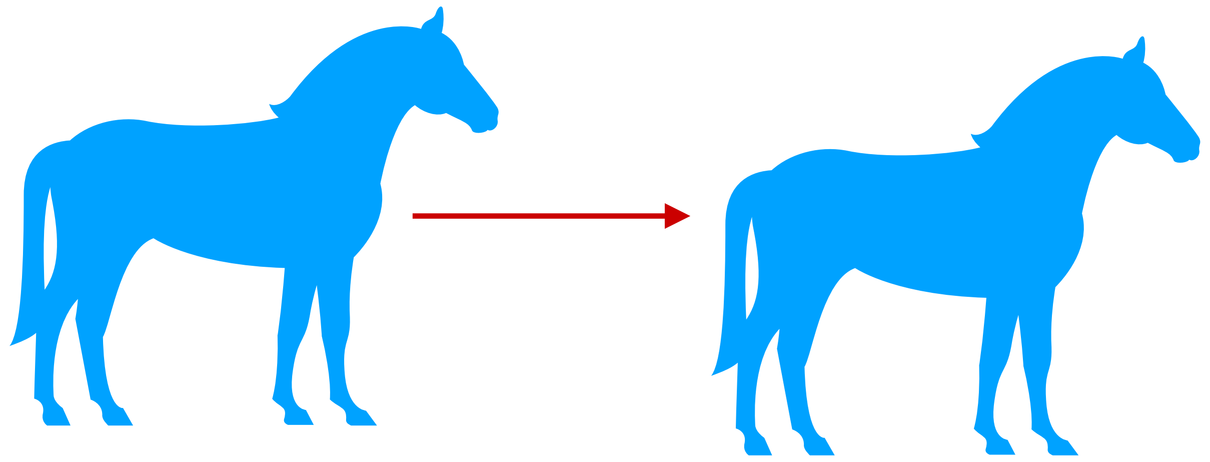


At each location multiple transformed filter copies

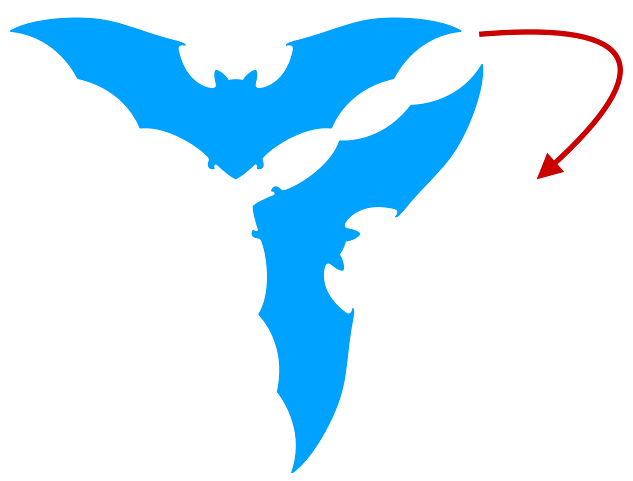
Hints

Translation	Rotation	Roto-translation
$\mathcal{T}_{\mathbf{x}}[\mathbf{y}] = \mathbf{y} + \mathbf{x}$	$\mathcal{T}_{\mathbf{x}}[\mathbf{y}] = \mathbf{R}_{\mathbf{x}}\mathbf{y}$	$\mathcal{T}_{\mathbf{x}}[\mathbf{y}] = \mathbf{R}_{\mathbf{x}}\mathbf{y} + \mathbf{t}_{\mathbf{x}}$

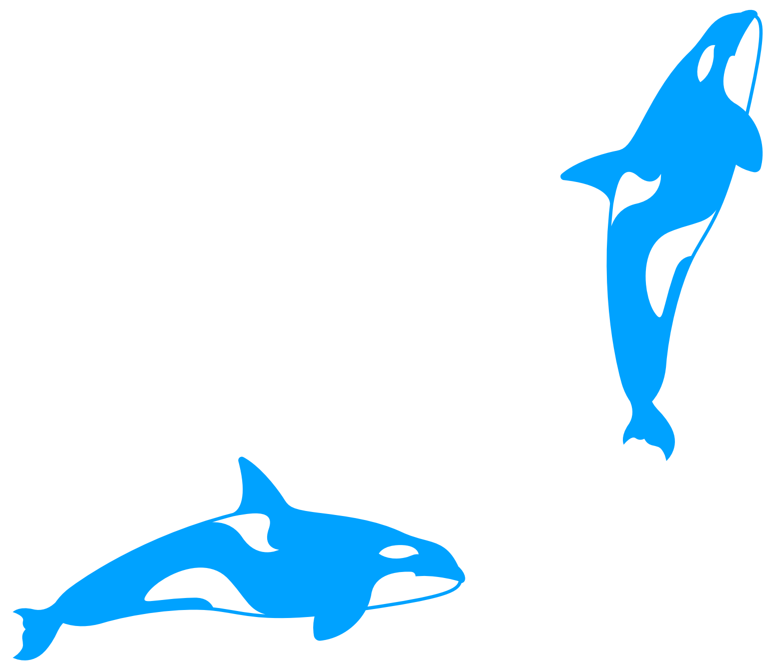
Group Transformation Examples



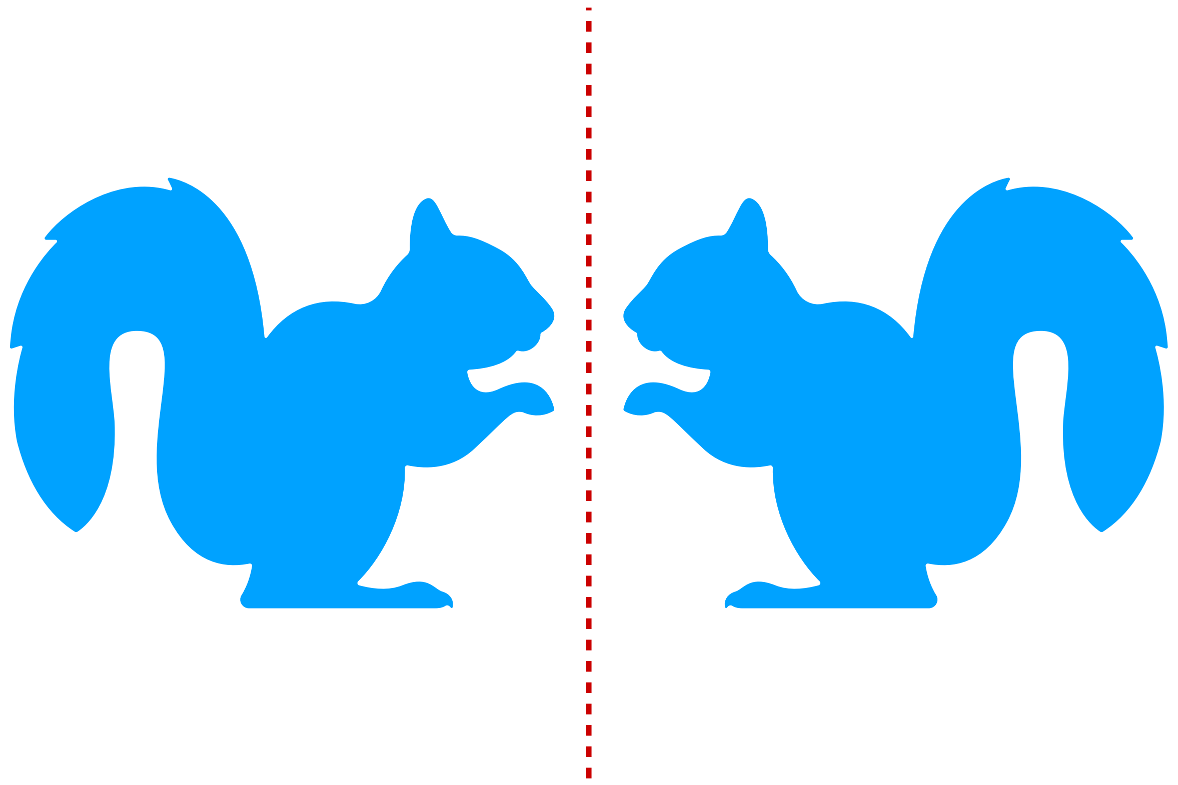
Translation



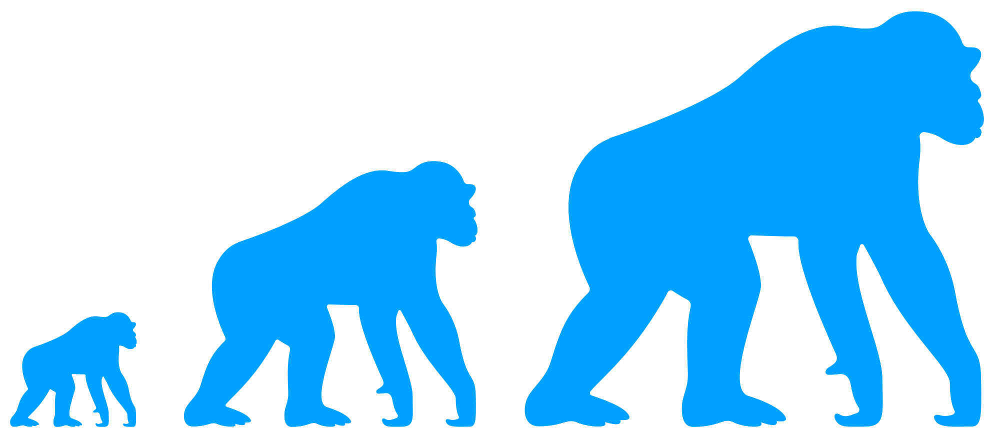
Rotation



Roto-translation

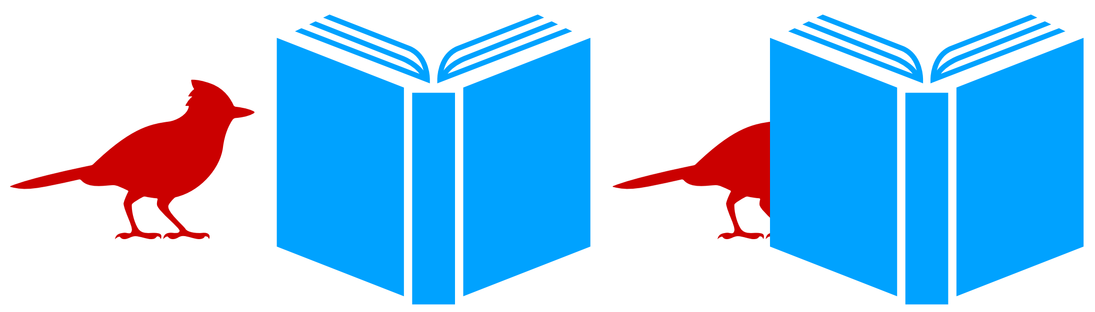


Reflections



Scalings

Non-example



Occlusions

SO(2) Rotation

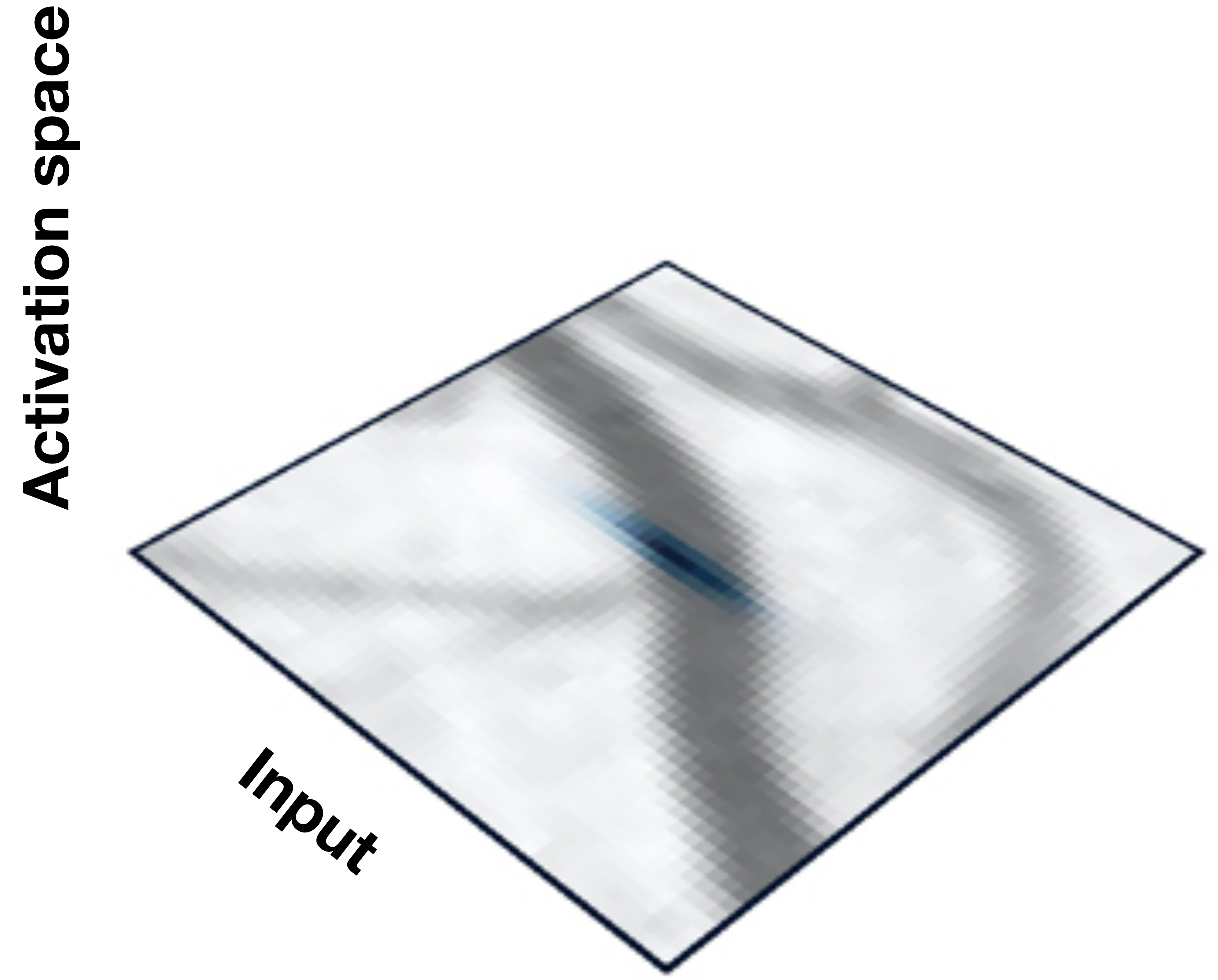
Group convolution

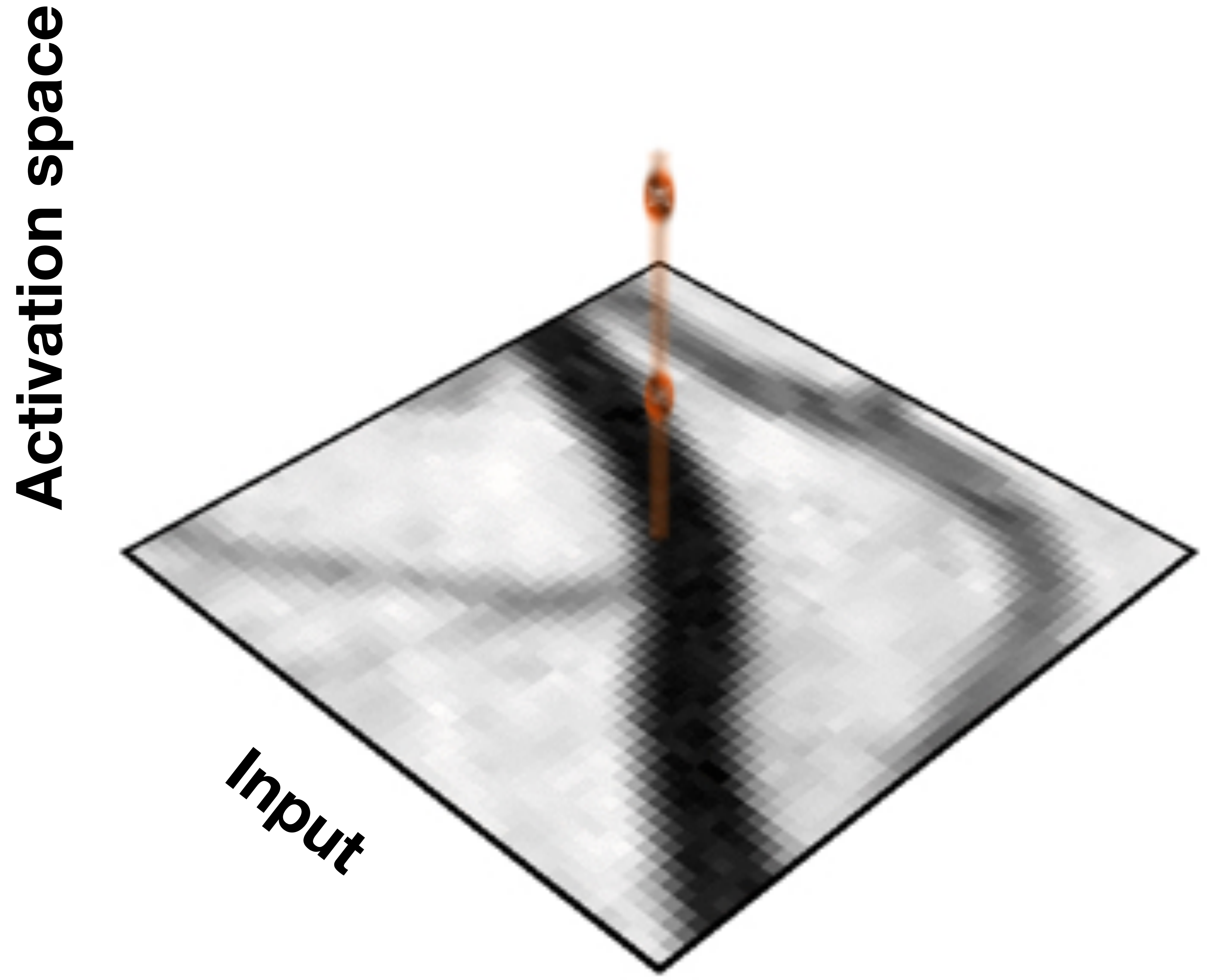
$$[\mathbf{F} * \mathbf{W}](\mathbf{x}) = \sum_{\mathbf{y}\in\mathcal{Y}} \mathbf{F}(\mathbf{y})\mathbf{W}(\mathcal{T}_{\mathbf{x}}^{-1}[\mathbf{y}])$$

Notice index of convolution

$$[\mathbf{F} * \mathbf{W}](\mathbf{R}_{\theta}) = \sum_{\mathbf{y}\in\mathcal{Y}} \mathbf{F}(\mathbf{y})\mathbf{W}(\mathcal{T}_{\mathbf{R}_{\theta}^{-1}}[\mathbf{x}])$$

Rotating kernel





$$[\mathcal{T}_{\mathbf{R}_{\psi}}[\mathbf{F}] * \mathbf{W}](\mathbf{R}_{\theta}) = [\mathbf{F} * \mathbf{W}](\mathbf{R}_{\psi}^{-1}\mathbf{R}_{\theta})$$

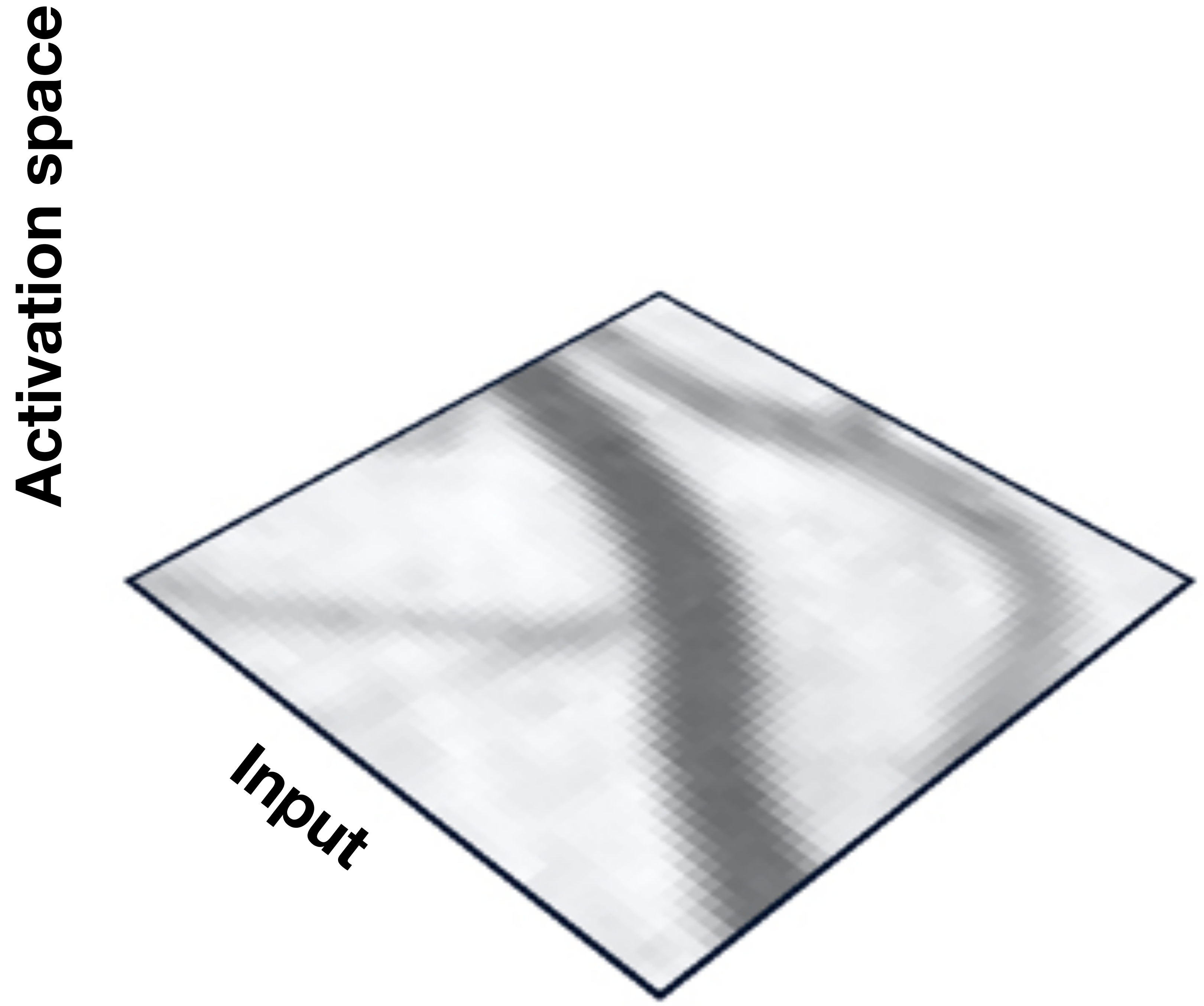
SE(2) Roto-translation

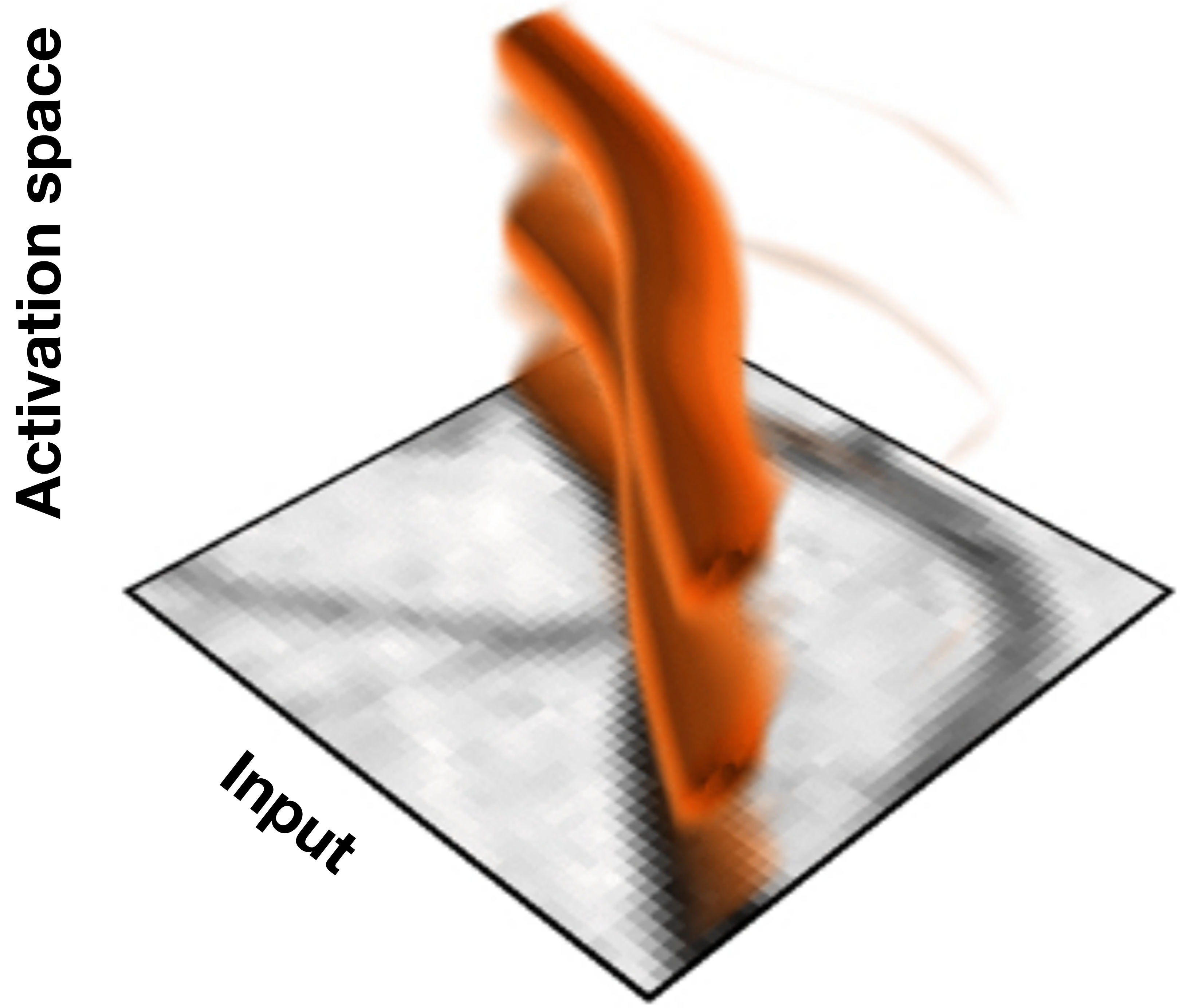
Group convolution

$$[\mathbf{F} * \mathbf{W}](\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}} \mathbf{F}(\mathbf{y}) \mathbf{W}(\mathcal{T}_{\mathbf{x}}^{-1}[\mathbf{y}])$$

$$[\mathbf{F} * \mathbf{W}](\mathbf{R}_{\theta}, \mathbf{x}) = \sum_{\mathbf{y} \in \mathbb{Z}^2} \mathbf{F}(\mathbf{y}) \mathbf{W}(\mathbf{R}_{\theta}^{-1}(\mathbf{y} - \mathbf{x}))$$

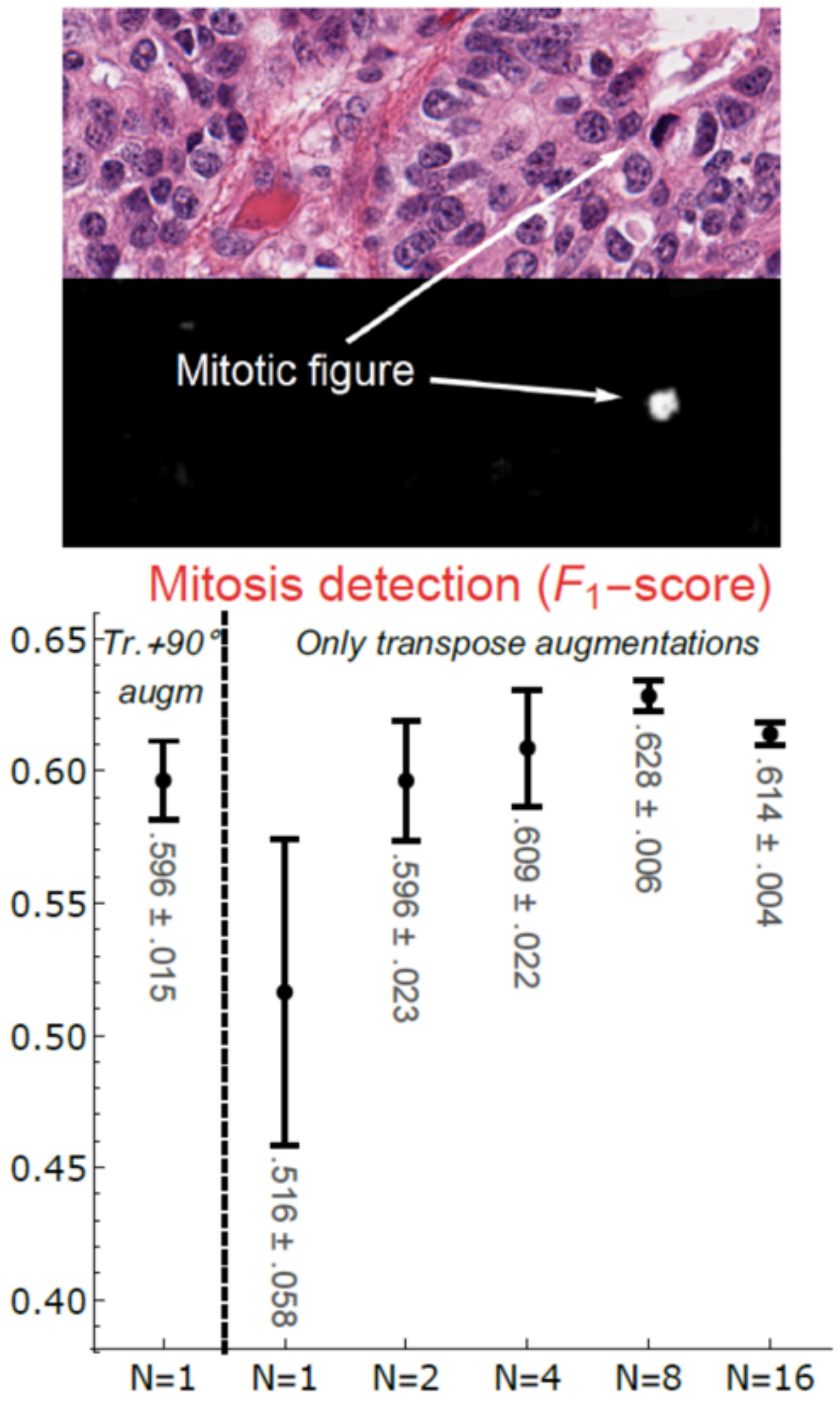
Rotating + translating kernel

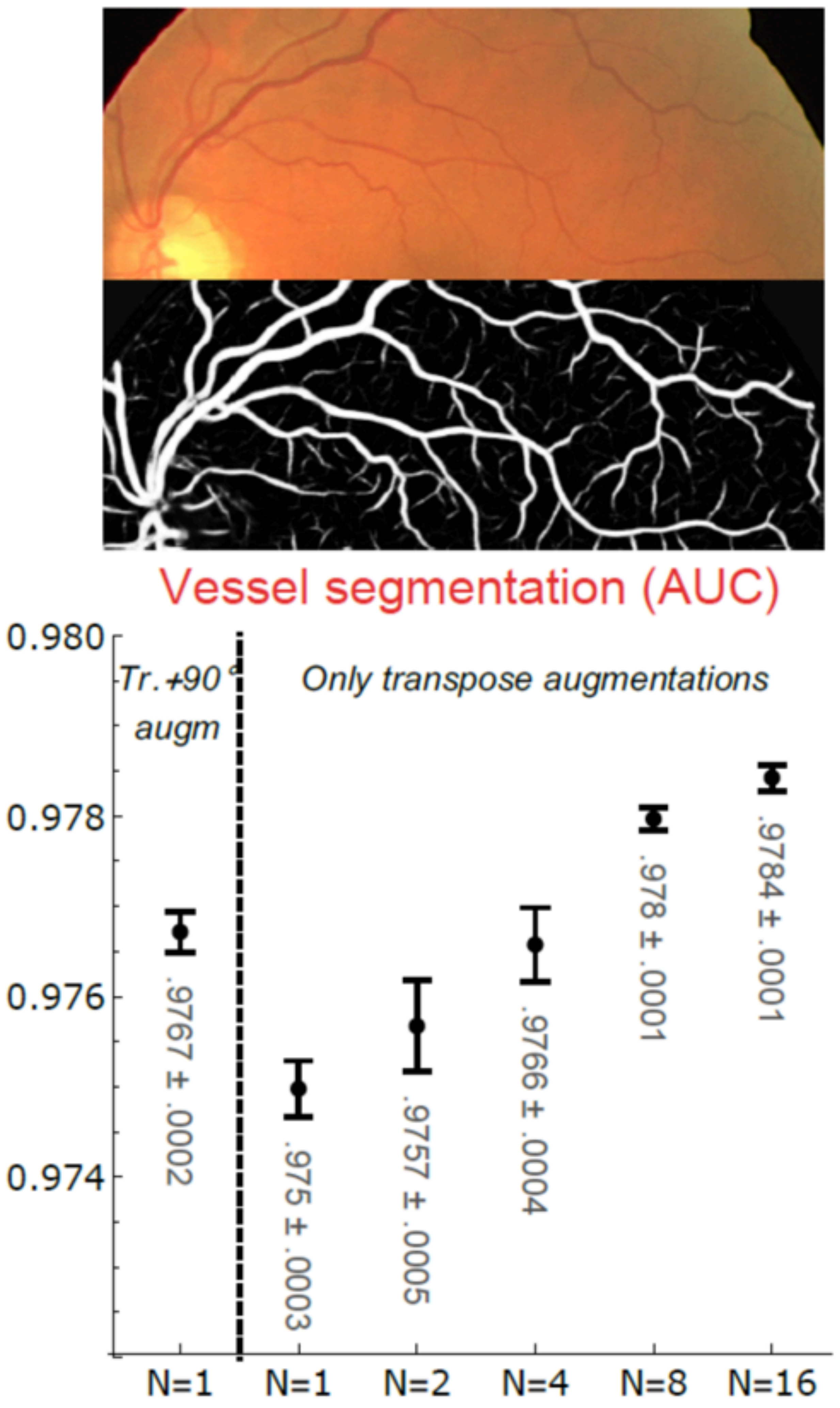


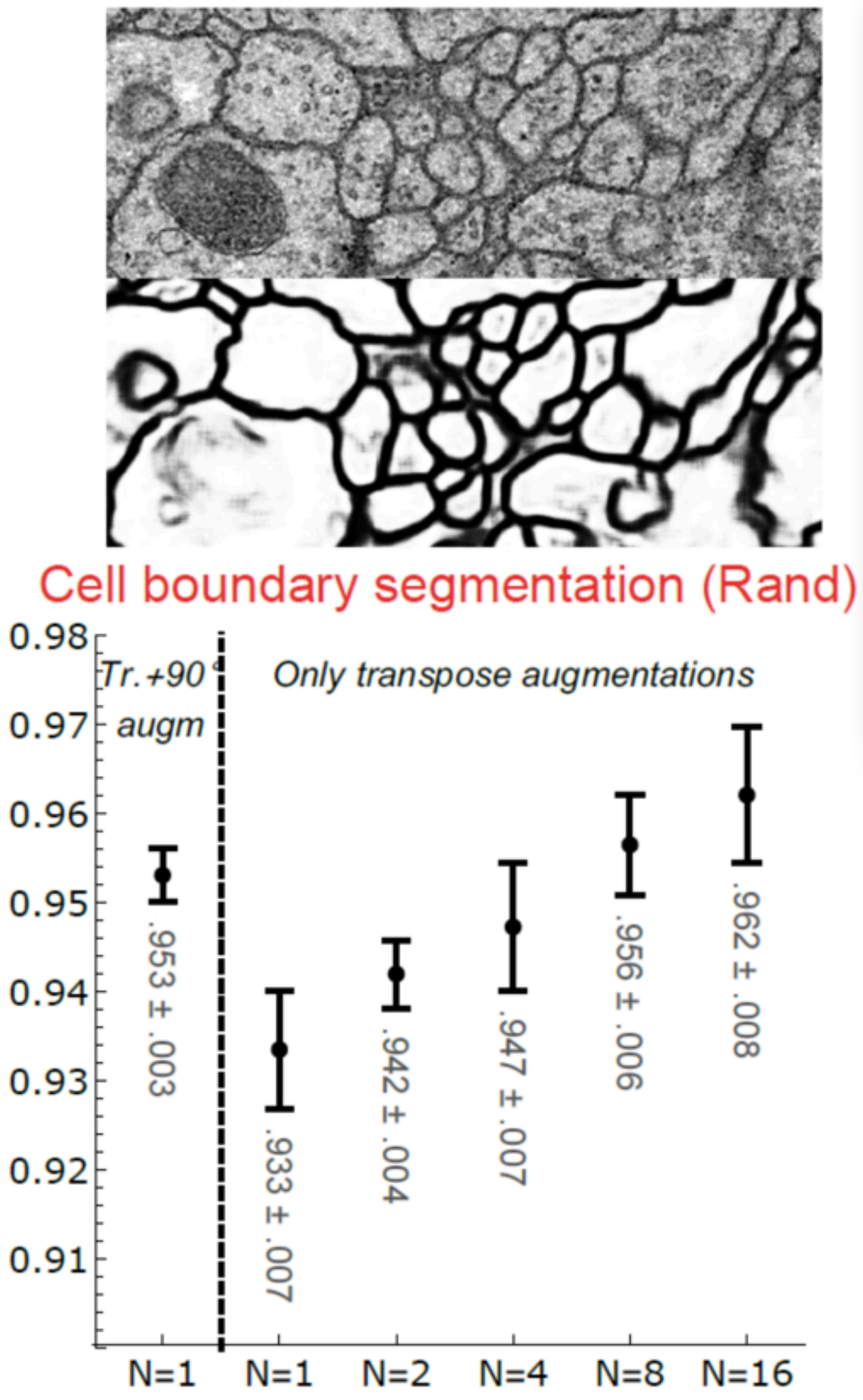


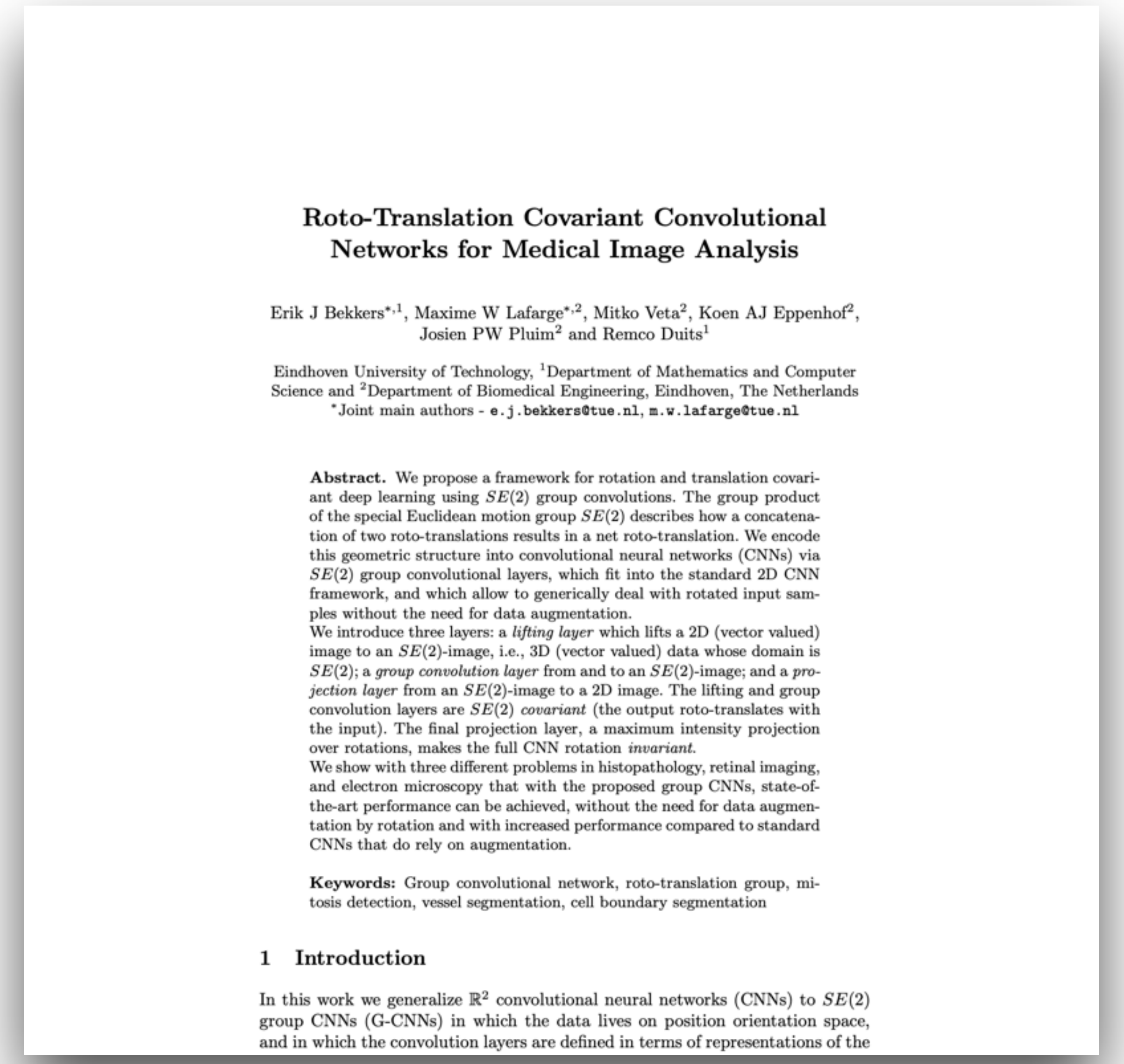
$$[\mathcal{T}_{\mathbf{R}_{\psi}, \mathbf{z}}[\mathbf{F}] * \mathbf{W}](\mathbf{R}_{\theta}, \mathbf{x}) = [\mathbf{F} * \mathbf{W}](\mathbf{R}_{\theta - \psi}, \mathbf{R}_{\psi}^{-1}(\mathbf{x} - \mathbf{z}))$$

Sample efficiency









Roto-Translation Covariant Convolutional Networks for Medical Image Analysis

Erik J Bekkers*[,1], Maxime W Lafarge*[,2], Mitko Veta[2], Koen AJ Eppenhof[2], Josien PW Pluim[2] and Remco Duits[1]

Eindhoven University of Technology, [1]Department of Mathematics and Computer Science and [2]Department of Biomedical Engineering, Eindhoven, The Netherlands
*Joint main authors - e.j.bekkers@tue.nl, m.w.lafarge@tue.nl

Abstract. We propose a framework for rotation and translation covariant deep learning using $SE(2)$ group convolutions. The group product of the special Euclidean motion group $SE(2)$ describes how a concatenation of two roto-translations results in a net roto-translation. We encode this geometric structure into convolutional neural networks (CNNs) via $SE(2)$ group convolutional layers, which fit into the standard 2D CNN framework, and which allow to generically deal with rotated input samples without the need for data augmentation.
We introduce three layers: a *lifting layer* which lifts a 2D (vector valued) image to an $SE(2)$-image, i.e., 3D (vector valued) data whose domain is $SE(2)$; a *group convolution layer* from and to an $SE(2)$-image; and a *projection layer* from an $SE(2)$-image to a 2D image. The lifting and group convolution layers are $SE(2)$ *covariant* (the output roto-translates with the input). The final projection layer, a maximum intensity projection over rotations, makes the full CNN rotation *invariant*.
We show with three different problems in histopathology, retinal imaging, and electron microscopy that with the proposed group CNNs, state-of-the-art performance can be achieved, without the need for data augmentation by rotation and with increased performance compared to standard CNNs that do rely on augmentation.

Keywords: Group convolutional network, roto-translation group, mitosis detection, vessel segmentation, cell boundary segmentation

1 Introduction

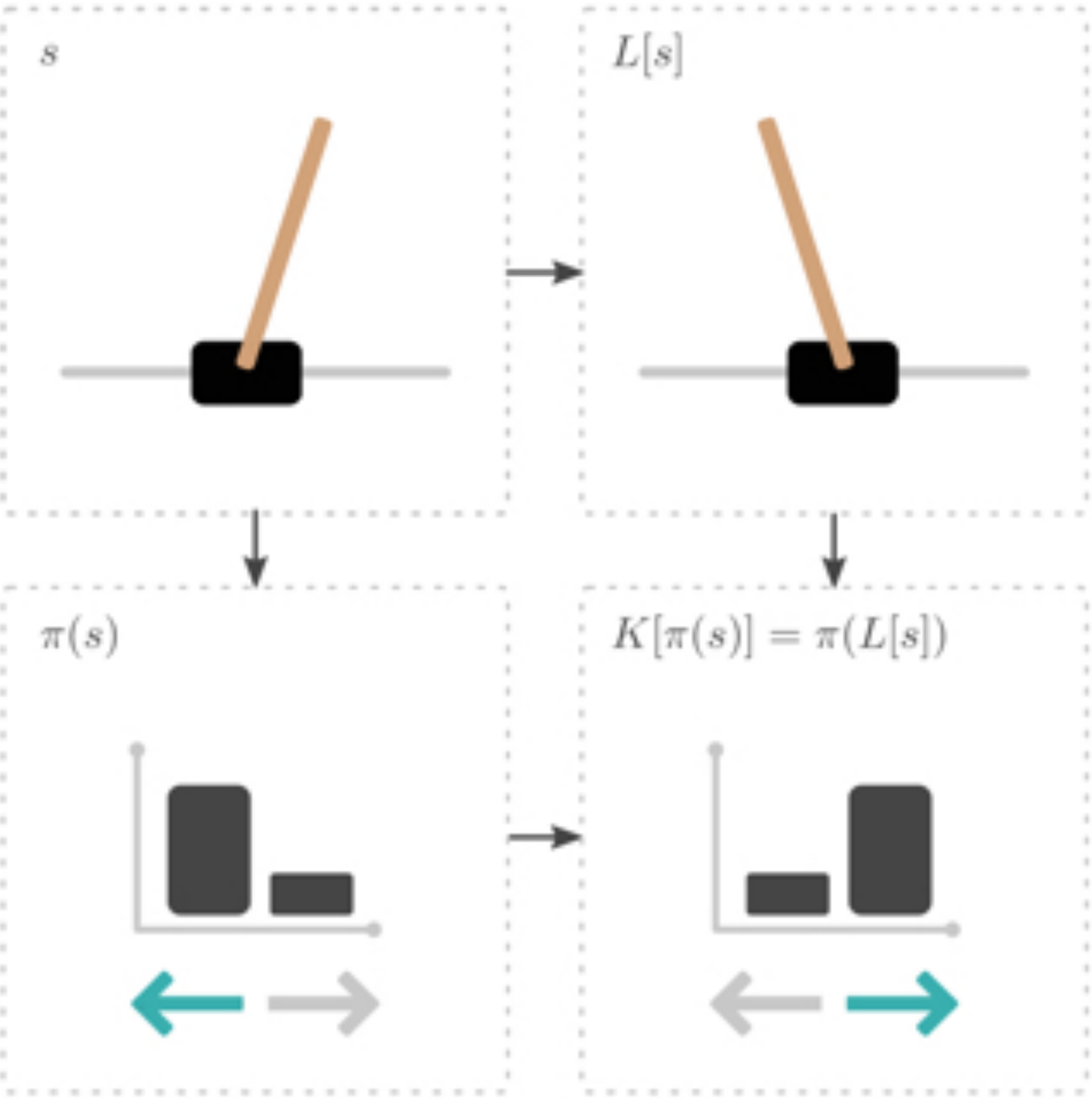
In this work we generalize $\mathbb{R}^2$ convolutional neural networks (CNNs) to $SE(2)$ group CNNs (G-CNNs) in which the data lives on position orientation space, and in which the convolution layers are defined in terms of representations of the

Bekkers et al., 2018

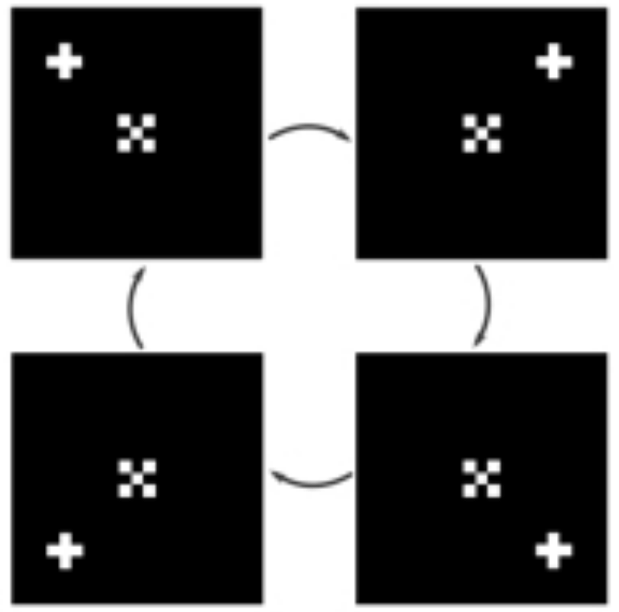
Sample efficiency

RL sample hungry: exploit symmetries

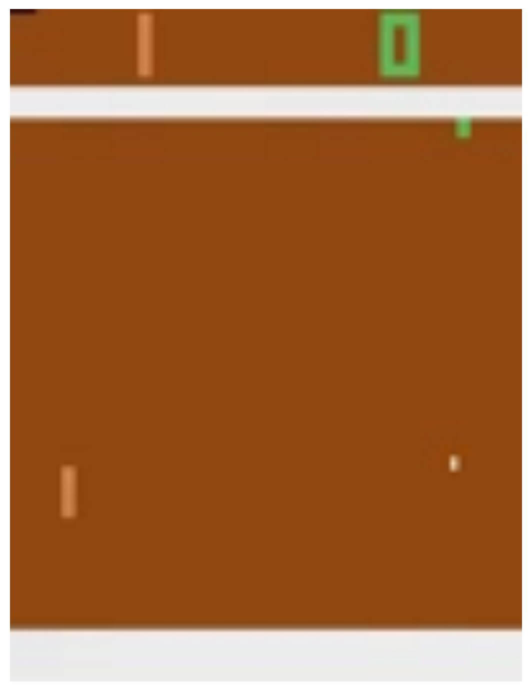
Transformation equivariant policies via MDP homomorphism theory



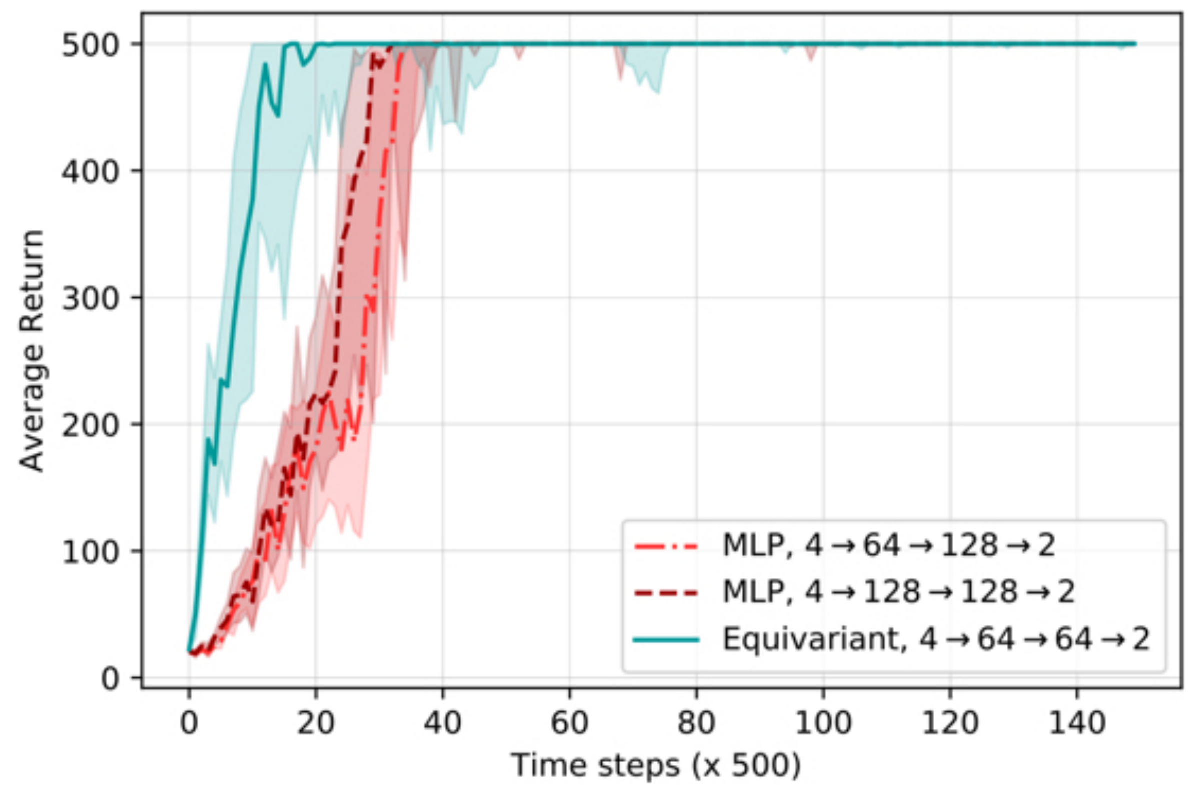
Cartpole

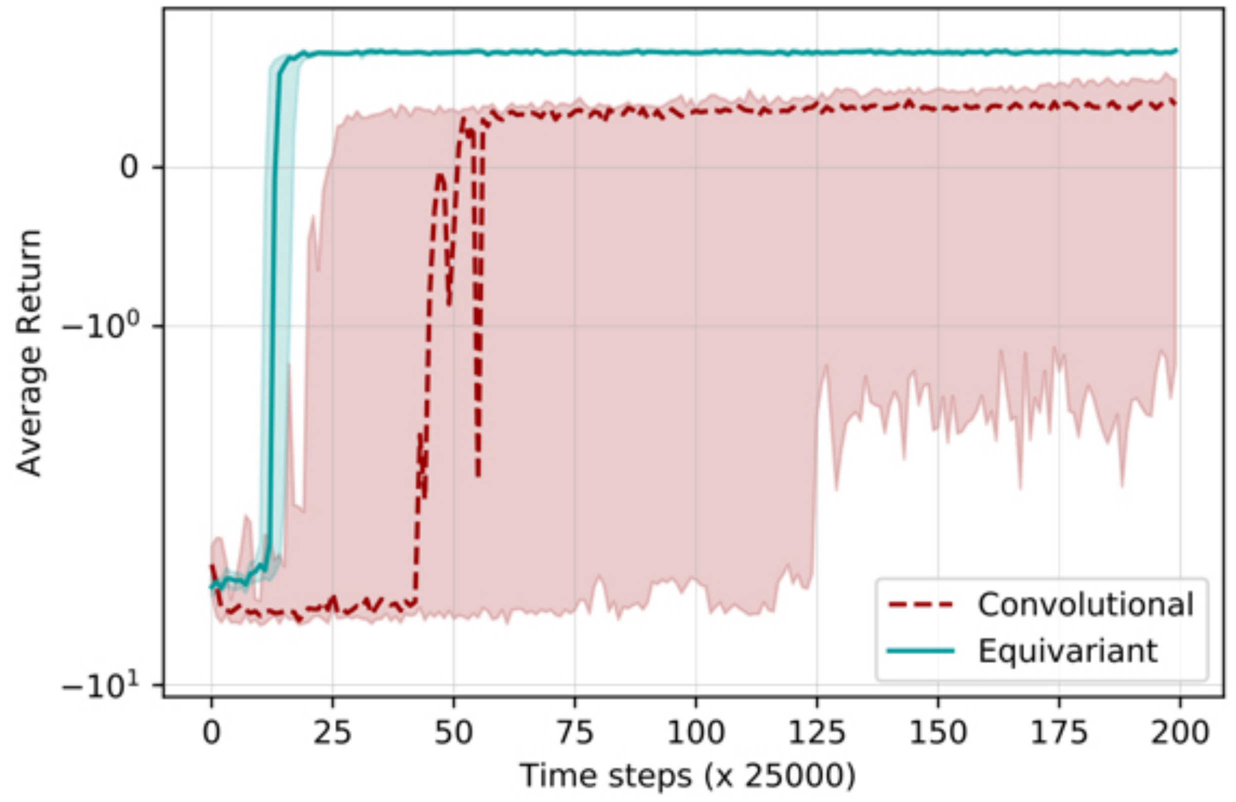


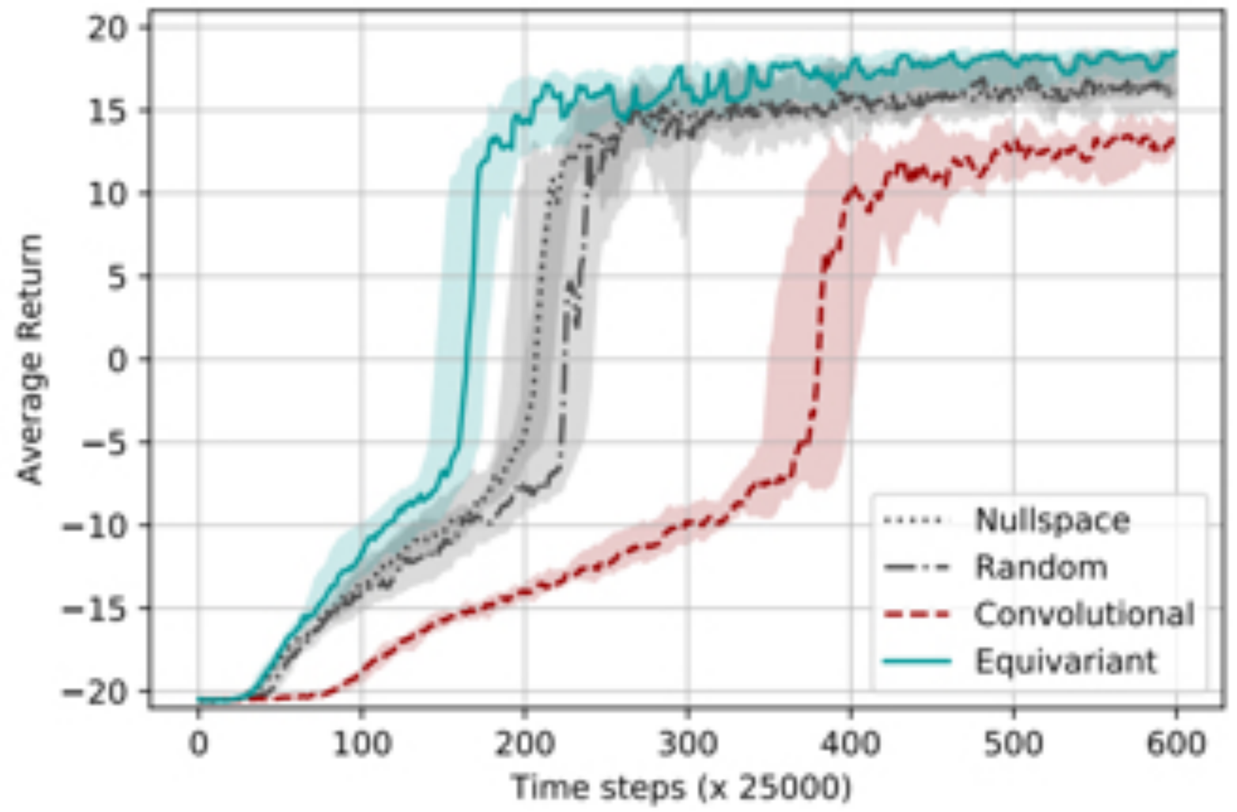
GridWorld

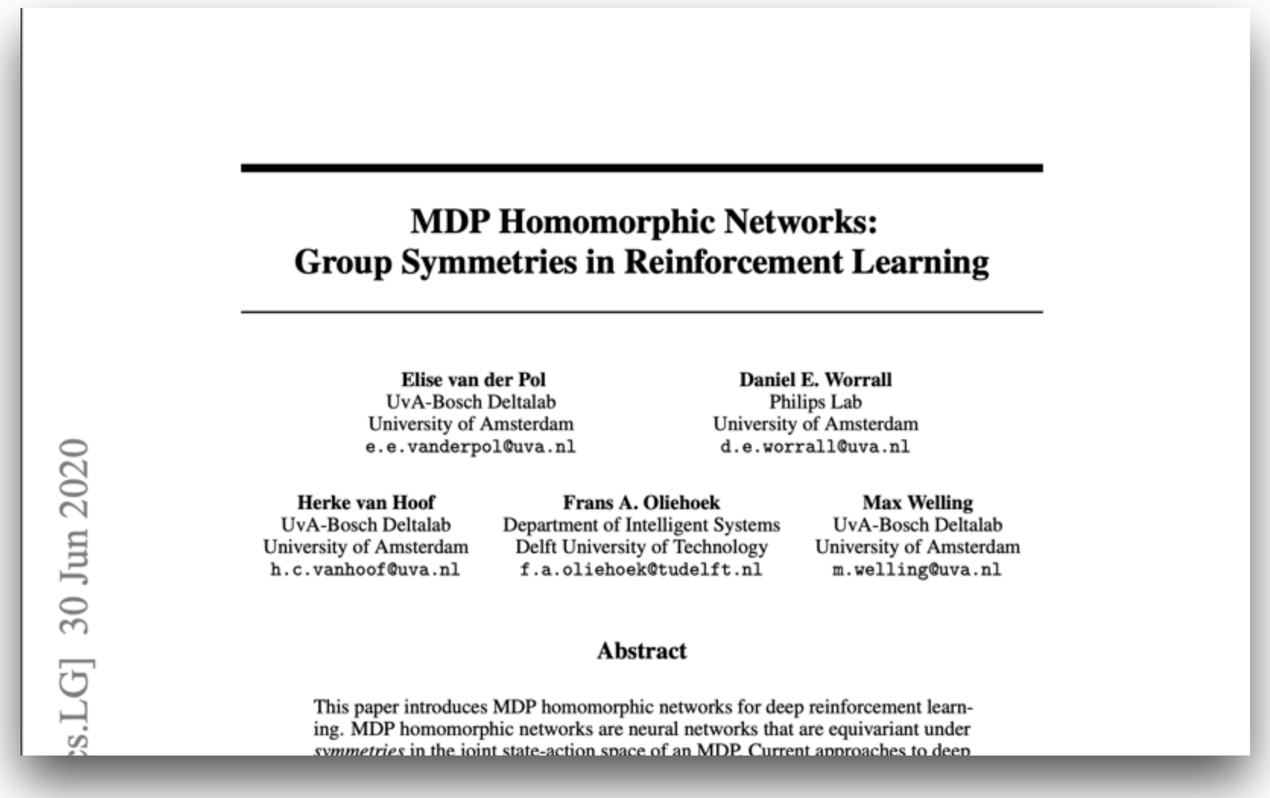


Pong









MDP Homomorphic Networks:
Group Symmetries in Reinforcement Learning

Elise van der Pol, UvA-Bosch Deltalab, University of Amsterdam, e.e.vanderpol@uva.nl
Daniel E. Worrall, Philips Lab, University of Amsterdam, d.e.worrall@uva.nl
Herke van Hoof, UvA-Bosch Deltalab, University of Amsterdam, h.c.vanhoof@uva.nl
Frans A. Oliehoek, Department of Intelligent Systems, Delft University of Technology, f.a.oliehoek@tudelft.nl
Max Welling, UvA-Bosch Deltalab, University of Amsterdam, m.welling@uva.nl

Abstract

This paper introduces MDP homomorphic networks for deep reinforcement learning. MDP homomorphic networks are neural networks that are equivariant under symmetries in the joint state-action space of an MDP. Current approaches to deep

s.LG] 30 Jun 2020

Van der Pol et al. (2020)

Convolutions are special

Semigroup correlation

- Deep Scale Space

Gauge convolution

- Icosahedral CNN
- Gauge Equivariant Mesh Convolution

Group convolution

Steerable convolution

- Harmonic network
- Spherical convolution
- Steerable SE(3) network
- Tensor Field Network

Unitary convolution

- Learning to convolve

Regular-representation conv.

- **Translational convolution**
- Group CNN
- HexaConv
- CubeNet
- Scale Convolution

Tangent space convolutions

- LieConv

Steerable Convolutions

Another equivariant mapping is the *Fourier transform*

$$\mathcal{F}\left\{F\right\}(m) = \int_{-\pi}^{\pi} F(\phi) e^{-\iota m \phi} \, \mathrm{d}\phi$$

Inner product!

Fourier Shift Theorem

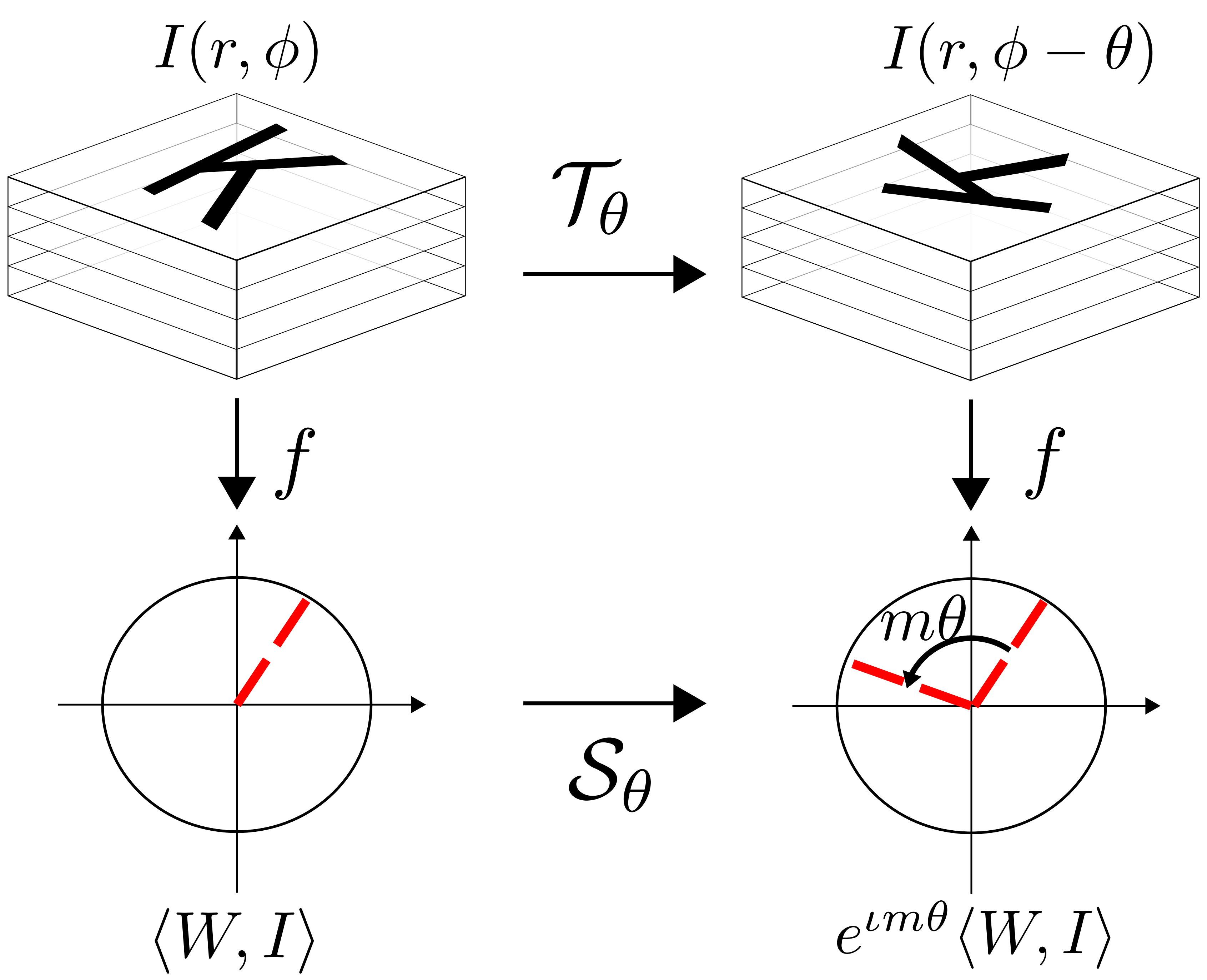
$m \in \mathbb{Z}$

$$\int_{-\pi}^{\pi} F(\phi - \theta) e^{-\iota m \phi} \, \mathrm{d}\phi = \int_{-\pi-\theta}^{\pi-\theta} F(\phi') e^{-\iota m (\phi' + \theta)} \, \mathrm{d}\phi'$$

$\iota = \sqrt{-1}$

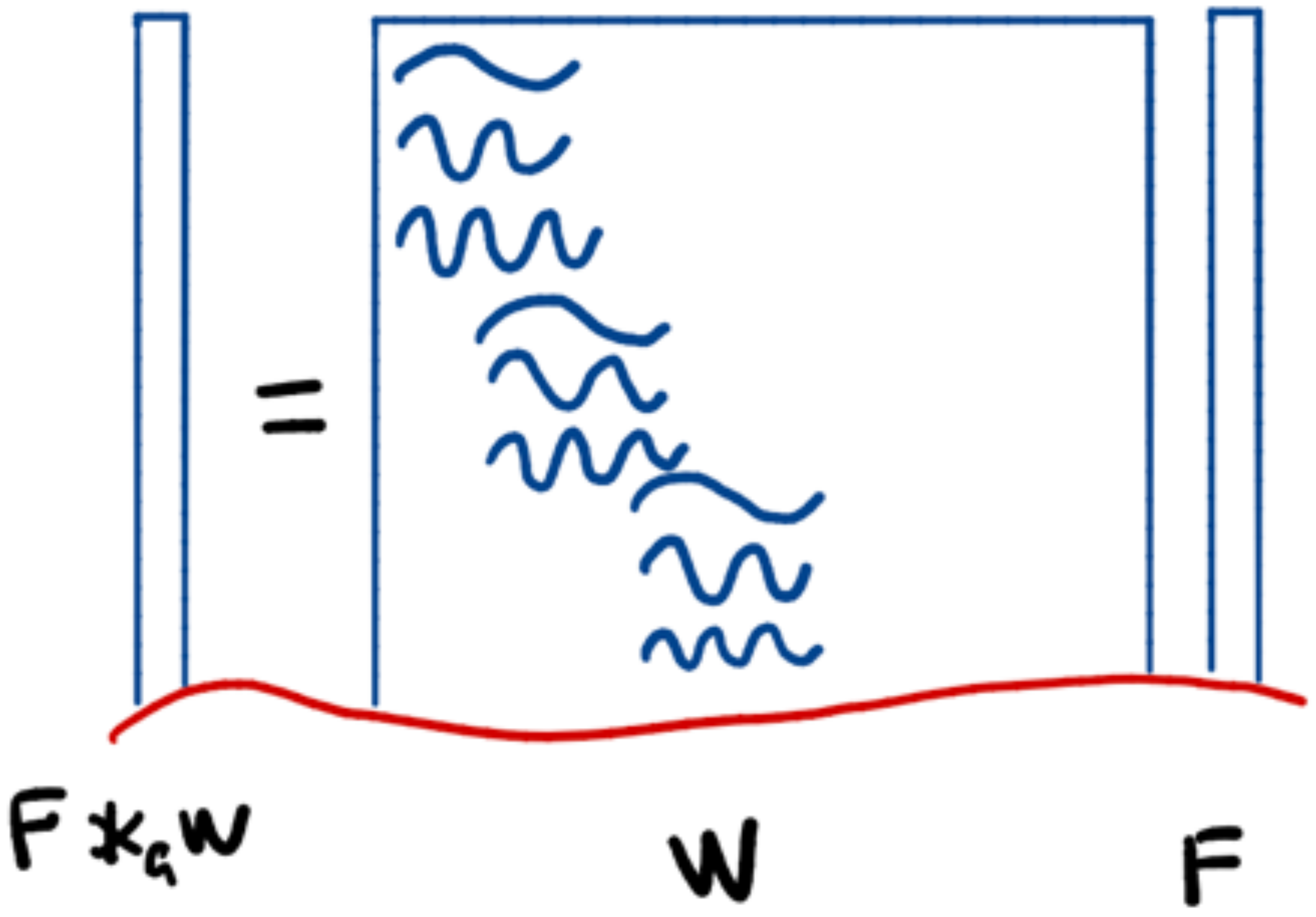
$$= \underbrace{e^{-\iota m \theta}}_{\text{Equivariant}} \underbrace{\int_{-\pi}^{\pi} F(\phi') e^{-\iota m \phi'} \, \mathrm{d}\phi'}_{\text{Invariant}}$$

$$\mathcal{F}\left\{\mathcal{T}_\theta[F]\right\}(m) = e^{-\iota m \theta} \mathcal{F}\left\{F\right\}(m) = \mathcal{S}_\theta[\mathcal{F}]\left\{F\right\}(m)$$



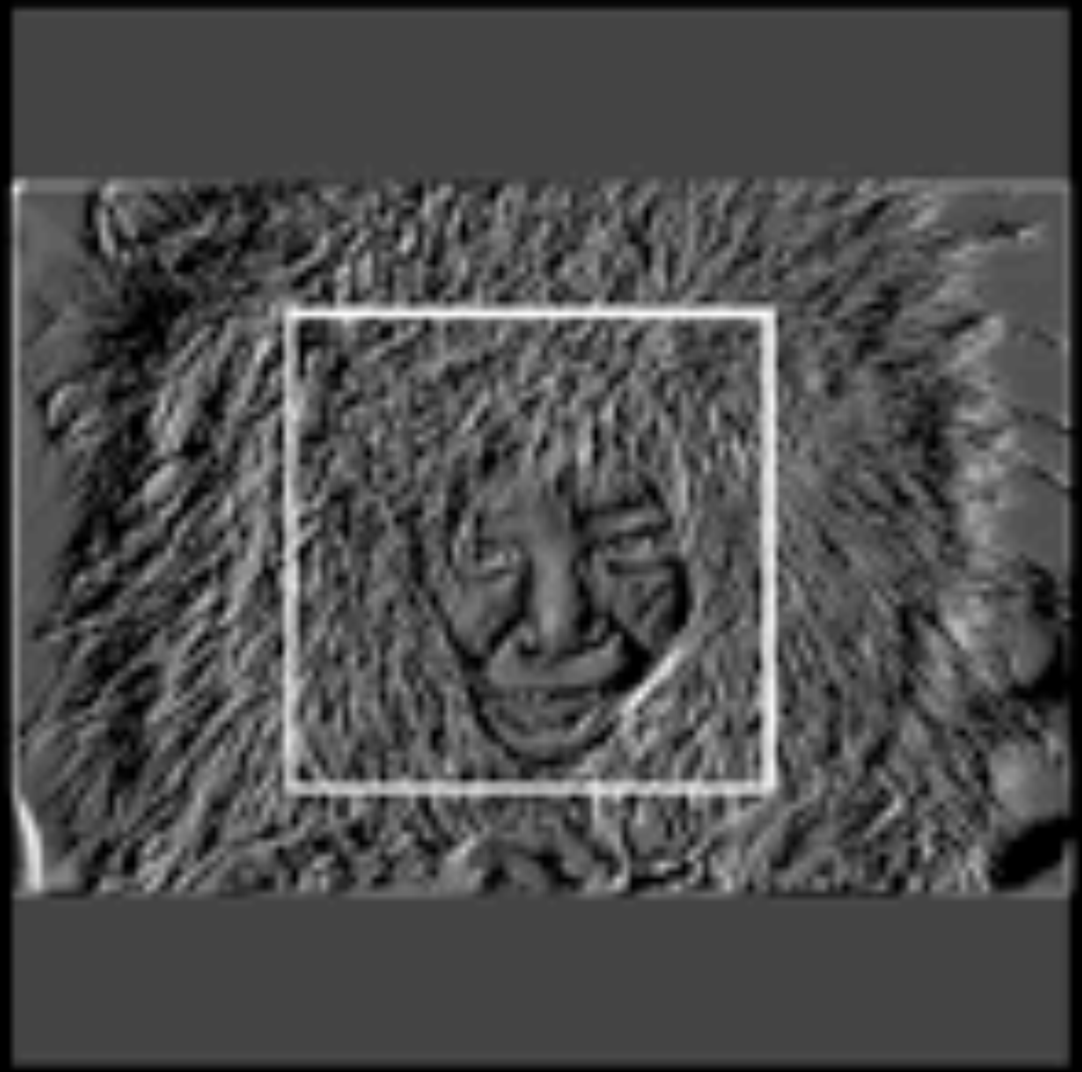
$$W_m(r; \phi) = \underbrace{R(r)}_{\text{Learnable}} e^{\iota(m\phi + \underbrace{\beta}_{\text{Learnable}})}$$

Fixed

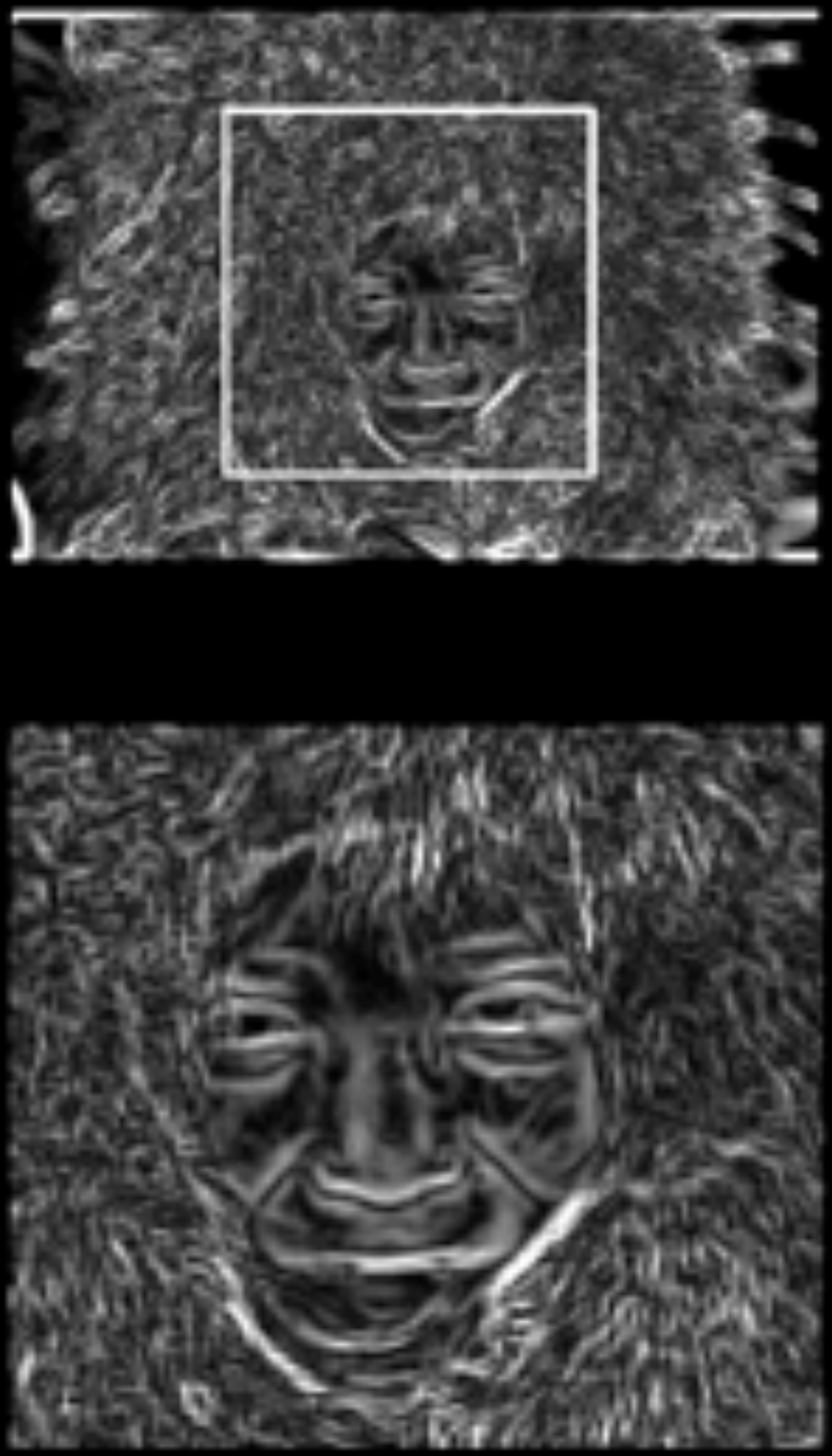


Can be generalized to other transformation groups

Lack of Rotation Equivariance



Steerable Equivariance



Comparison

Convolutional variants: Semigroup Correlations

Semigroup correlation

- Deep Scale Space

Gauge convolution

- Icosahedral CNN
- Gauge Equivariant Mesh Convolution

Group convolution

Steerable convolution

- Harmonic network
- Spherical convolution
- Steerable SE(3) network
- Tensor Field Network

Unitary convolution

- Learning to convolve

Regular-representation conv.

- **Translational convolution**
- Group CNN
- HexaConv
- CubeNet
- Scale Convolution

Tangent space convolutions

- LieConv

x2

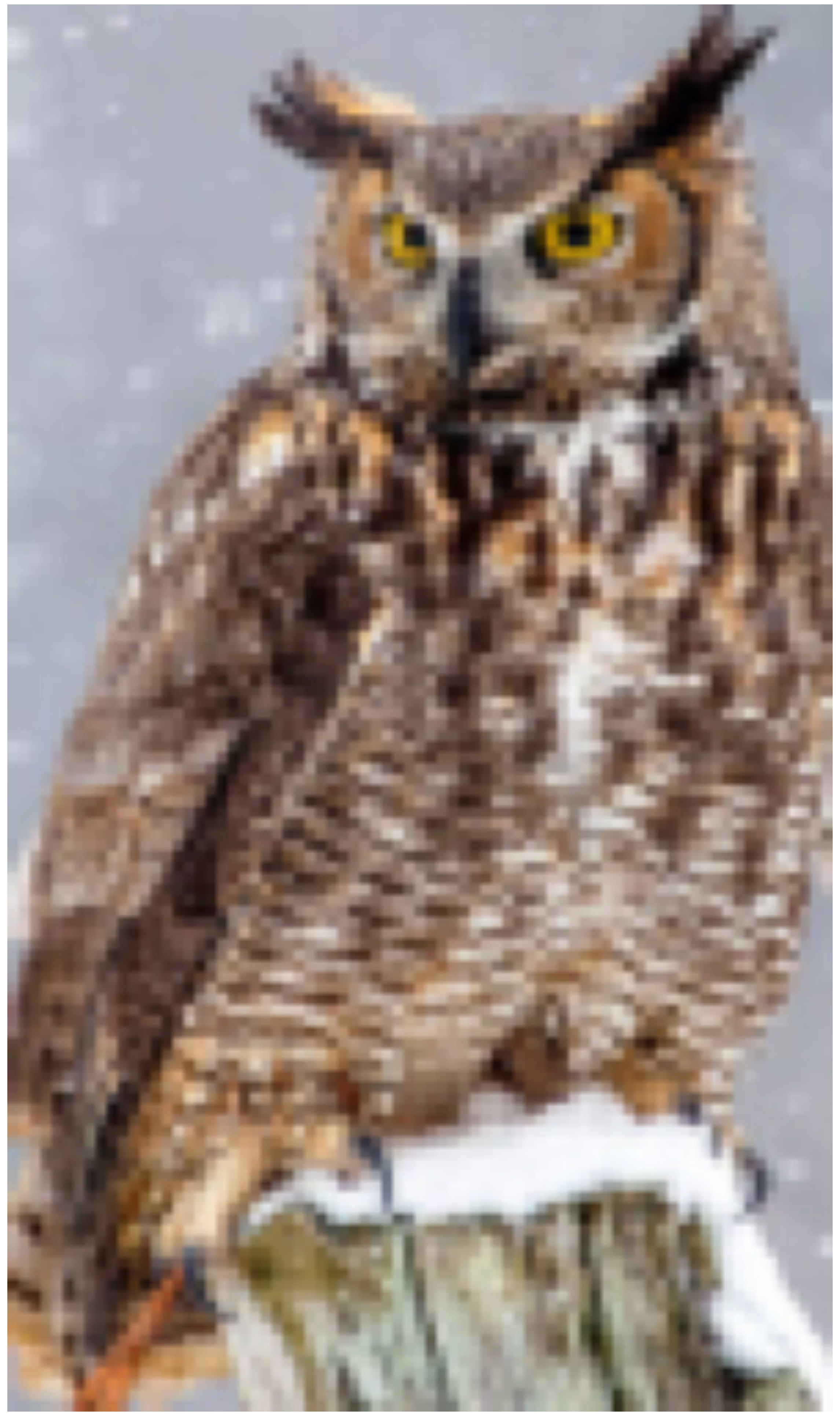
Original

Subsample

Bandlimit + subsample

Semigroups: How not to rescale

x4

Original	Subsample	Bandlimit + subsample
		

Semigroups: How not to rescale

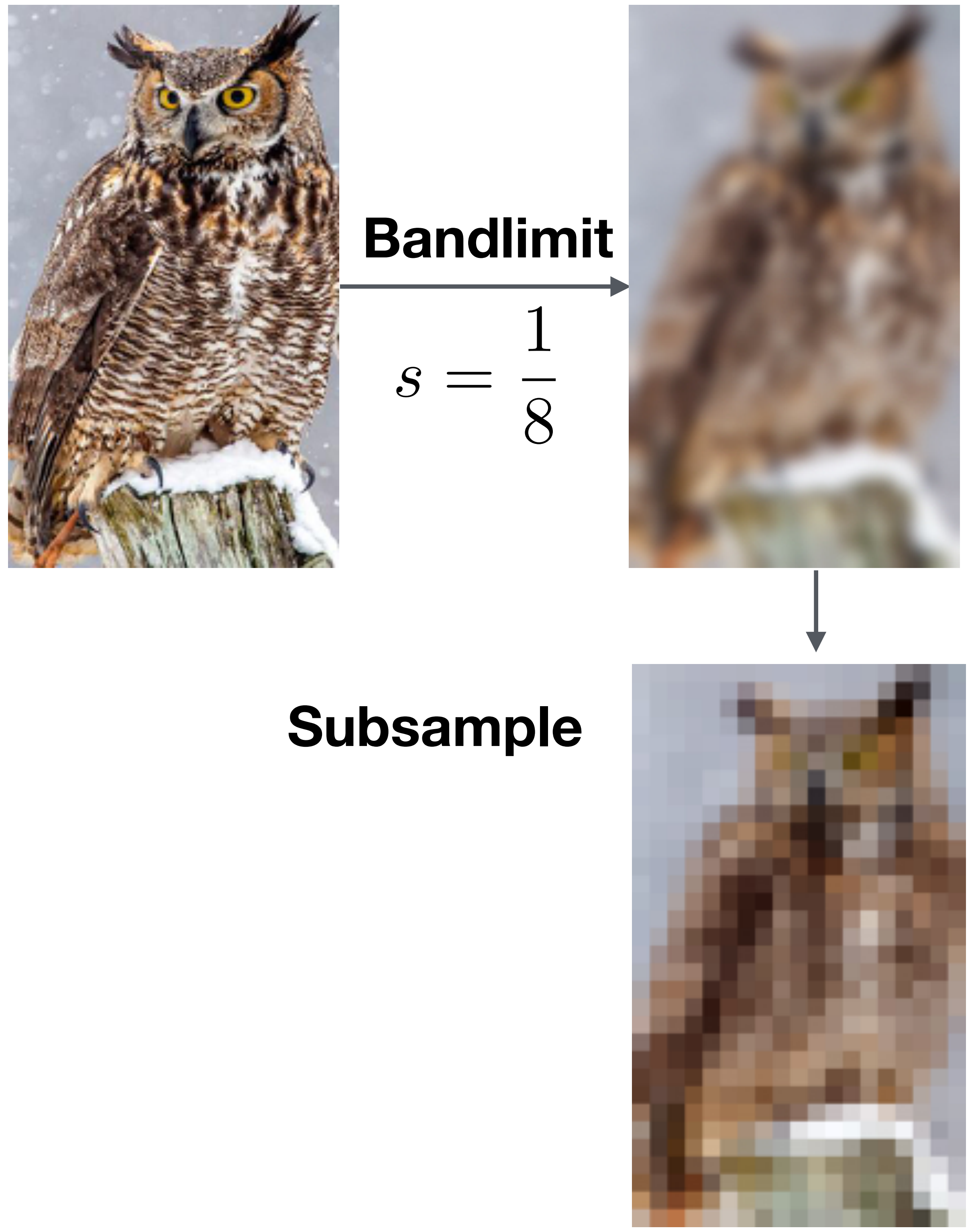
x4


Original


Subsample


Bandlimit + subsample

Semigroups: How to rescale



Blur appropriately

Subsample

$$\mathcal{T}_s[\mathbf{F}](\mathbf{x}) = \mathrm{Blur}_s[\mathbf{F}](s^{-1}\mathbf{x})$$

$\mathcal{T}_s^{-1}[\mathbf{F}](\mathbf{x})$ **does not exist!!!!**

Abstraction

1. **Closure**: compositions well-behaved

$$g \circ h \in G$$

2. **Associativity**: brackets unnecessary

$$(g \circ h) \circ k = g \circ (h \circ k) = g \circ h \circ k$$

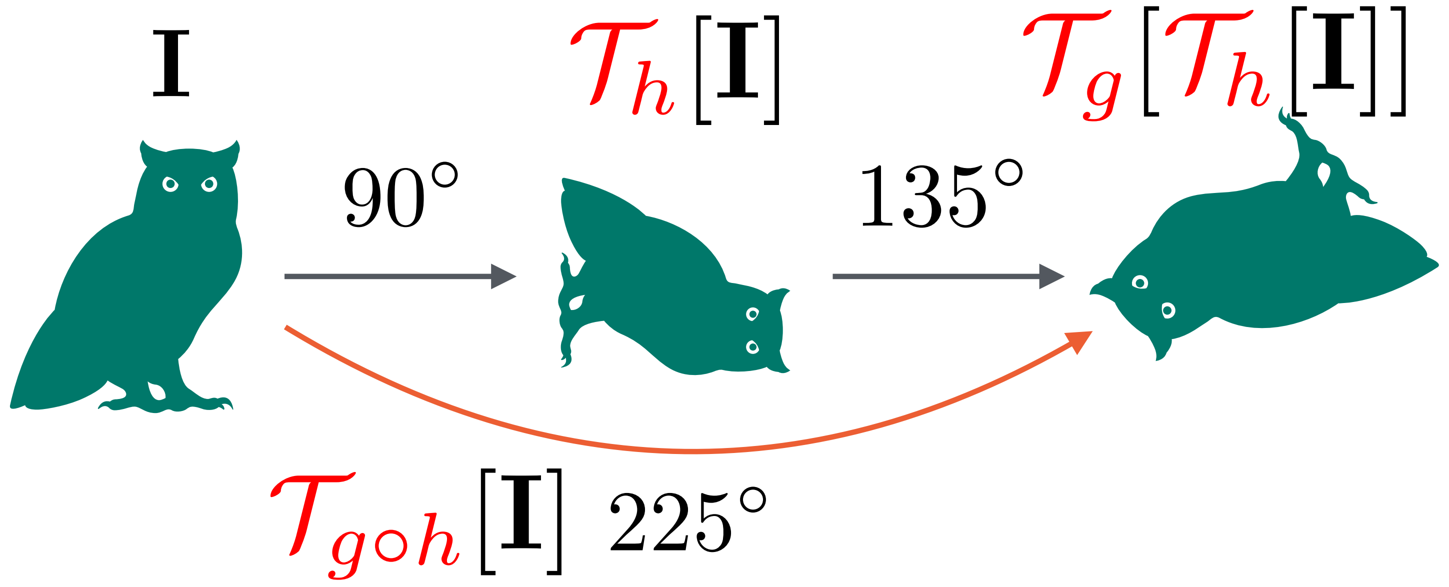
3. **Identity**: "do nothing" transformation

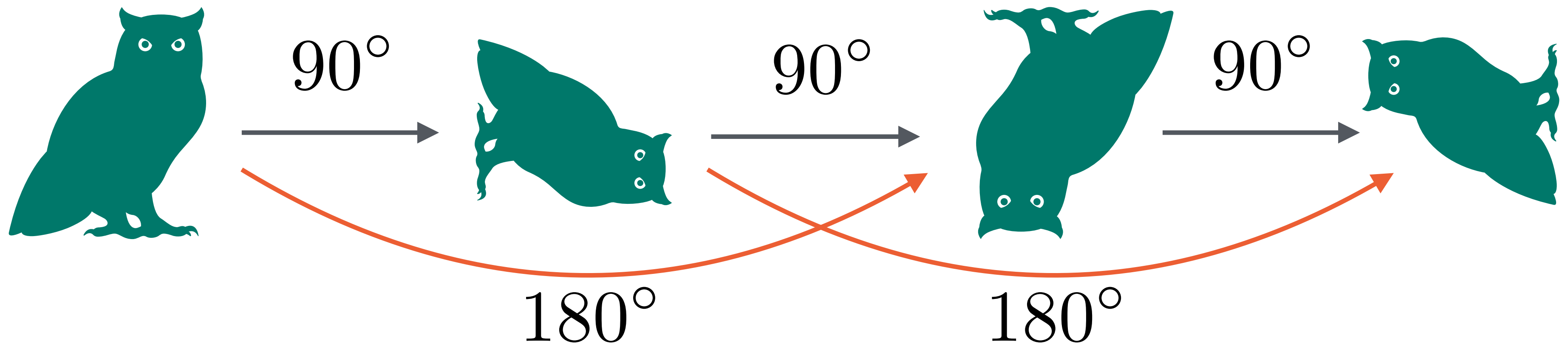
$$e \circ g = g \circ e = g$$

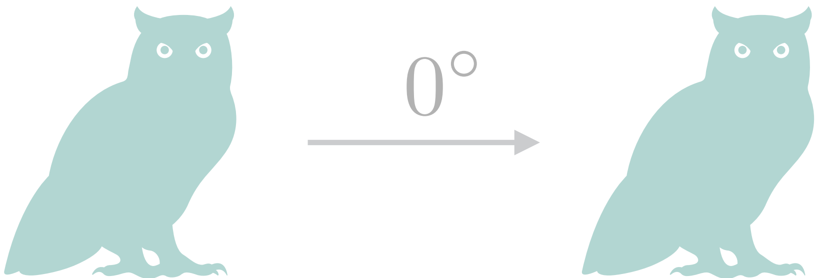
4. **Invertibility**: transformation reversible

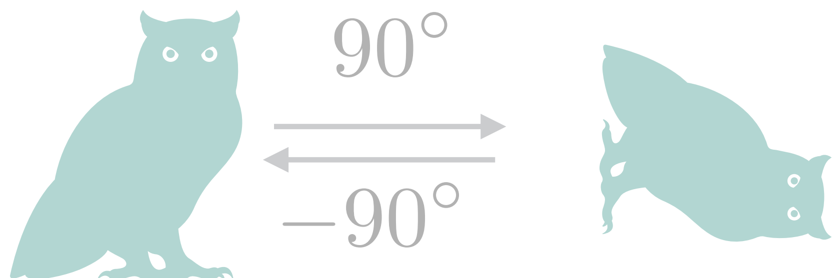
$$g \circ g^{-1} = g^{-1} \circ g = e$$

Action/Transformation









Group convolution

e.g. Cohen & Welling (2016)

$$[\mathbf{I} * \mathbf{W}](\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}} \mathbf{I}(\mathbf{y}) \mathbf{W}(\mathcal{T}_{\mathbf{x}}{}^{-1}[\mathbf{y}])$$

1) Rescaling ≠ shuffling pixel locations

Unitary Group Convolution

e.g. Diaconu & Worrall (2019)

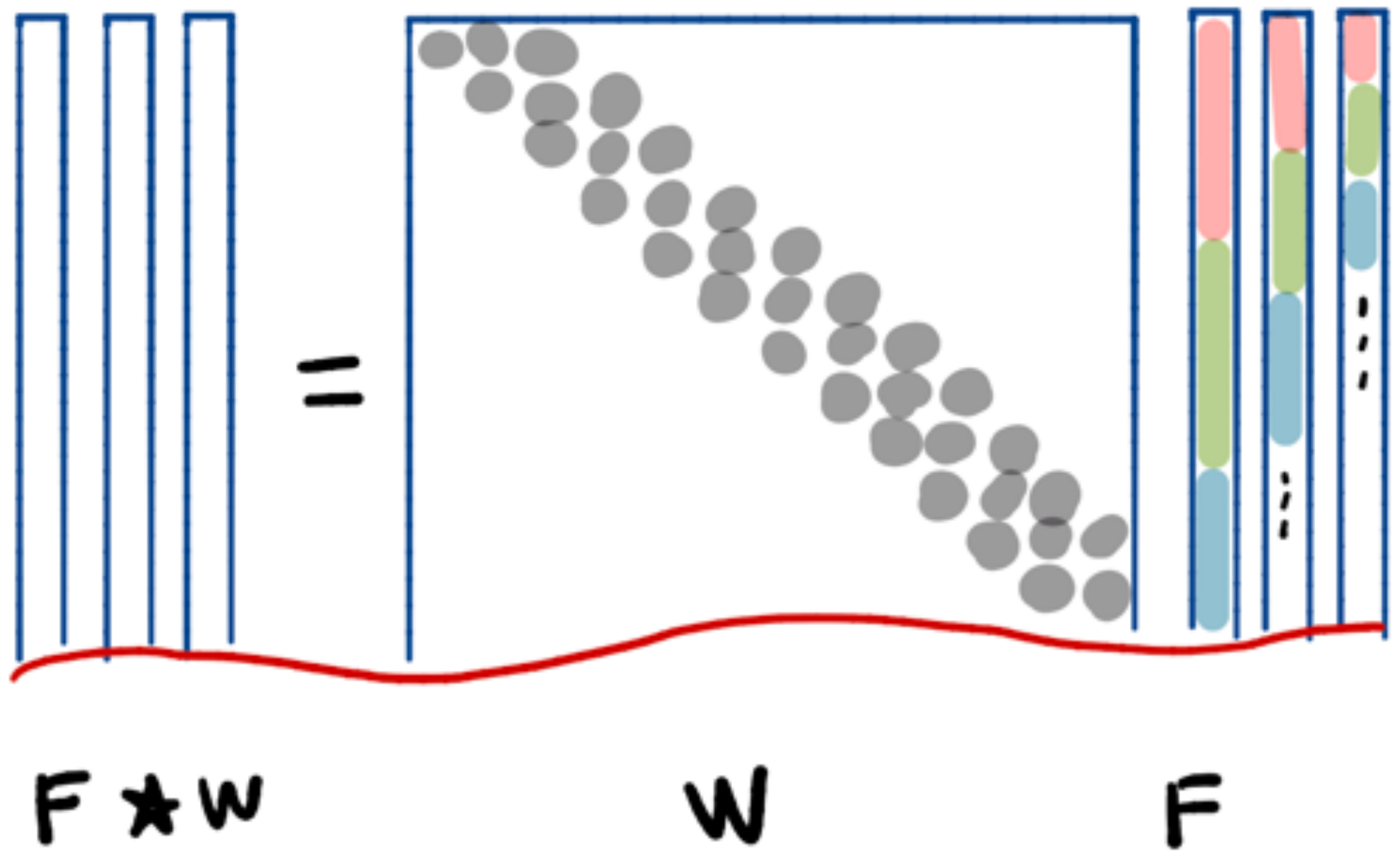
$$[\mathbf{I} * \mathbf{W}](\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}} \mathbf{I}(\mathbf{y}) \mathcal{T}_{\mathbf{x}}{}^{-1}[\mathbf{W}](\mathbf{y})$$

2) Use non-invertible transformations

Semigroup correlation

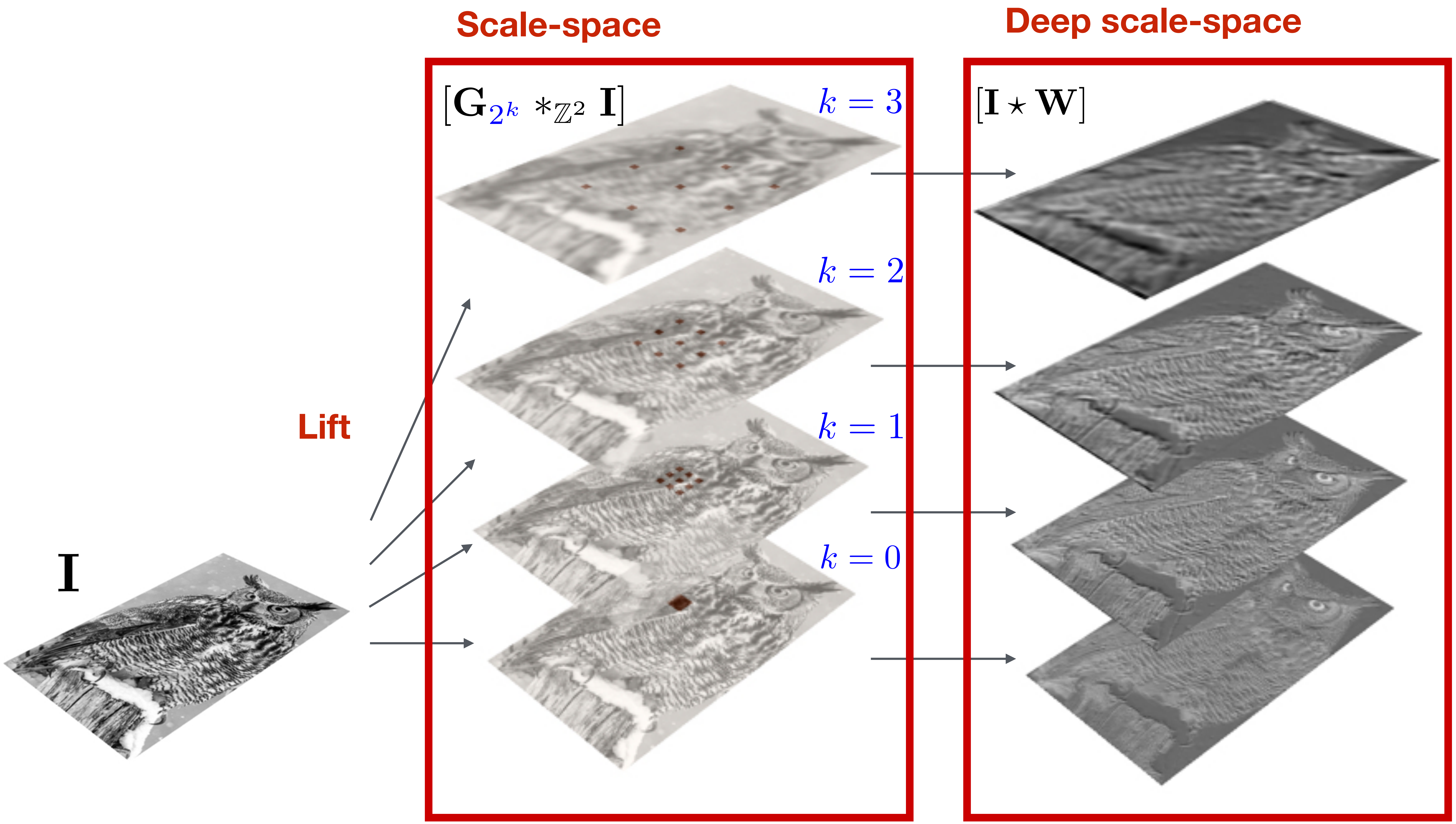
e.g. Worrall & Welling (2019)

$$[\mathbf{I} \star \mathbf{W}](\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}} \mathcal{T}_{\mathbf{x}}[\mathbf{I}](\mathbf{y}) \mathbf{W}(\mathbf{y})$$



The Semigroup Correlation

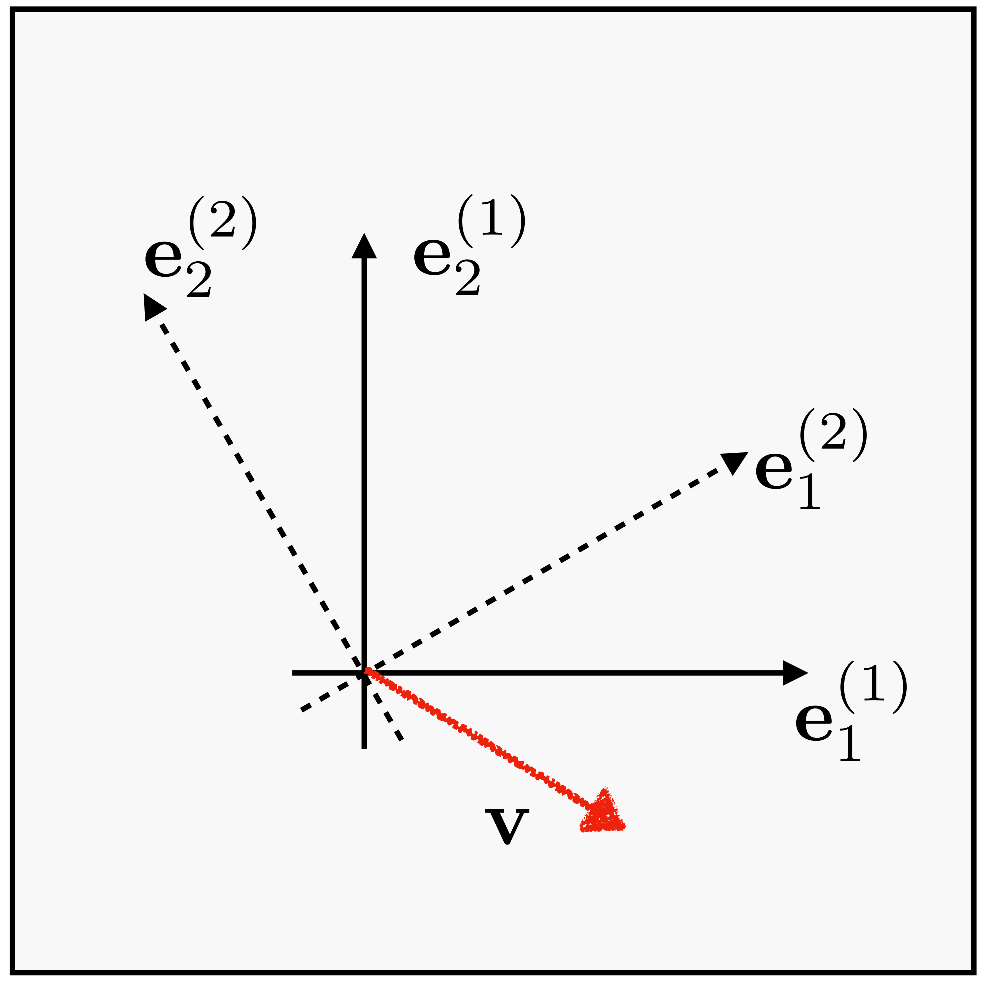
$$[\mathbf{I} \star \mathbf{W}](k, \mathbf{t}) = \sum_{\mathbf{y} \in \mathbb{Z}^2} \underbrace{[\mathbf{G}_{2^k} *_{\mathbb{Z}^2} \mathbf{I}]}_{\text{blur}} \underbrace{(2^k \mathbf{y} + \mathbf{t})}_{\text{dilated convolution}} \mathbf{W}(\mathbf{y})$$



Convolutional variants: Gauge Equivariance

Gauge Convolutions

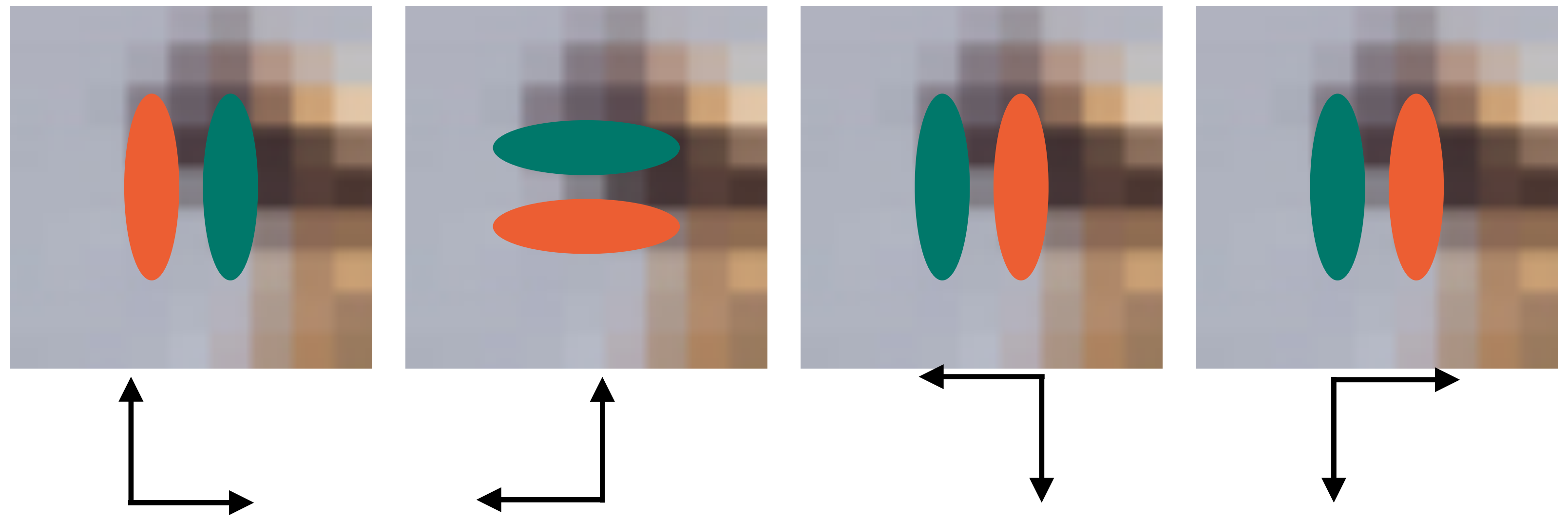
An alternative route to equivariance is to consider *local gauge transformations*



$$v_1^{(1)}\mathbf{e}_1^{(1)} + v_2^{(1)}\mathbf{e}_2^{(1)} = v_1^{(2)}\mathbf{e}_1^{(2)} + v_2^{(2)}\mathbf{e}_2^{(2)}$$

A *frame* is another name for coordinates. How to make frame independent?

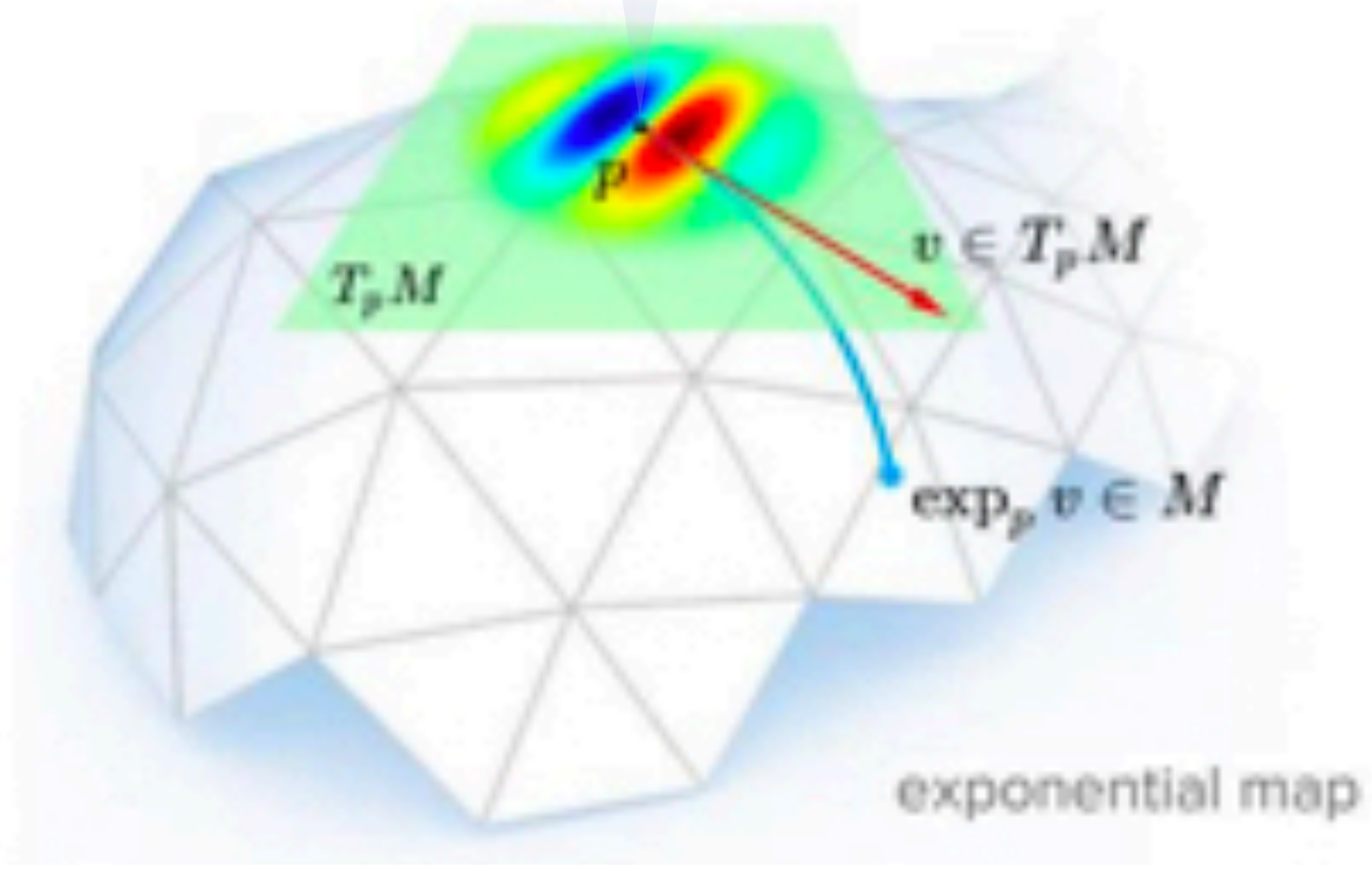
Take inner product between signal and filter represented in all frame orientations



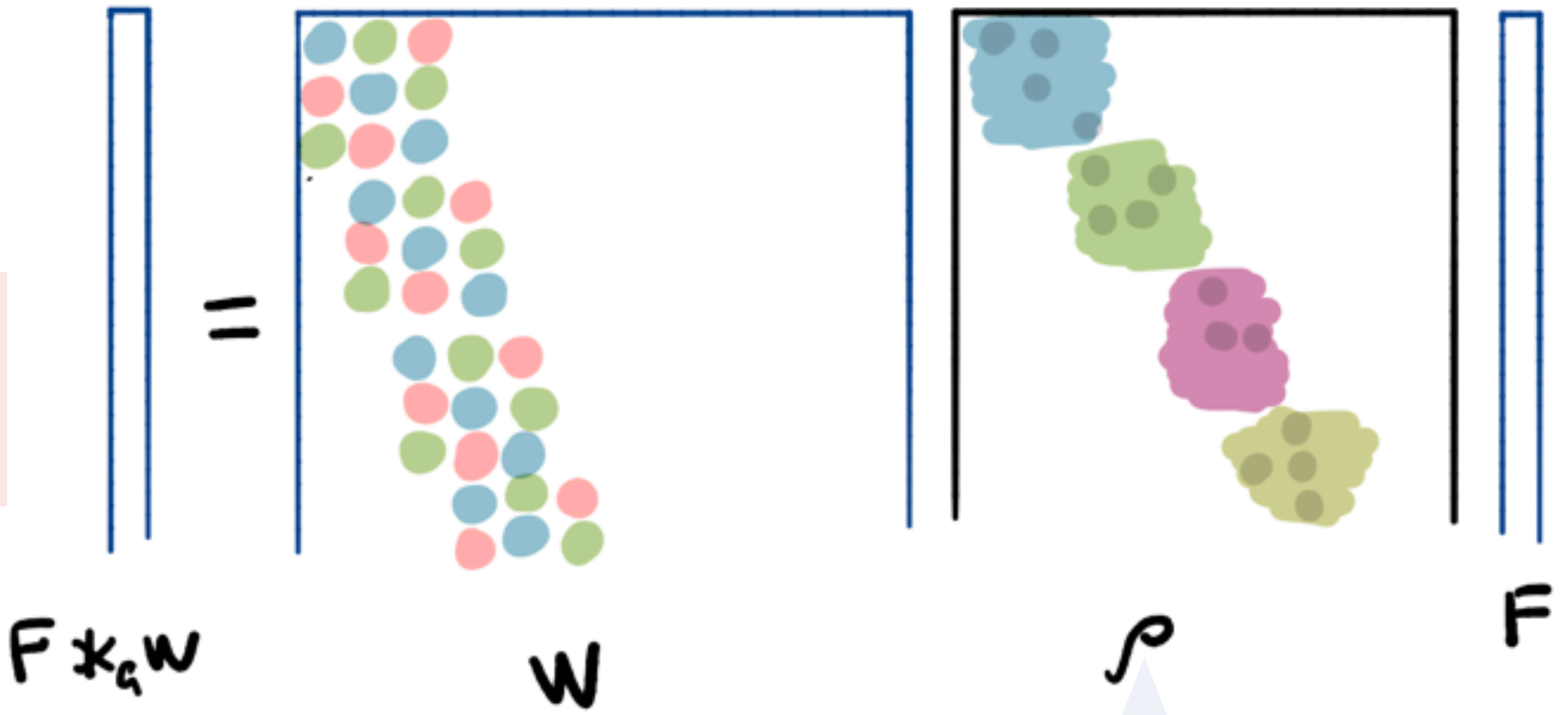
Like a group convolution, but…:
- Defined locally
- Can work on manifolds

Local group convolution, with curvature correction due to manifold

Need to map signal manifold → tangent space



Cohen et al. (2019)



manifold → tangent space

Convolutional variants

Convolutions

Powerful inductive bias

Linear layers + symmetry

Fewer learnable parameters

Many variants

This lecture: Equivariance

- Grey Box Models
- Convolutions
- Invariance and Equivariance
- Group Theory
- Convolution Generalizations
  - Group Convolutions, Steerable Convolutions, Semigroup Correlation, Gauge Convolution

References

Theory
Group CNNs - Cohen & Welling (2016)
Steerable CNNs - Cohen & Welling (2017)
On the Generalization of Equivariance and Convolution in Neural Networks to the Action of Compact Groups - Kondor & Trivedi (2018)
A General Theory of Equivariant CNNs on Homogeneous Spaces - Cohen et al. (2019)
Gauge Equivariance Convolutional Networks - Cohen*, Weiler*, Kicanaoglu* et al. (2019)
Deep Scale-spaces: Equivariance Over Scale - Worrall & Welling (2019)

Z^2 - Translation
Handwritten digit recognition with a Back-Propagation network - Lecun et al. (1990)

SE(2): Full group
Harmonic Networks: Deep translation and rotation equivariance - Worrall et al. (2017)

SE(2): All subgroups
General E(2)-Equivariant Steerable CNNs - Weiler & Cesa (2019)

E(2): Discrete subgroups
Group CNNs - Cohen & Welling (2016)
Exploiting cyclic symmetry in convolutional neural networks - Dieleman et al. (2016)
Steerable CNNs - Cohen & Welling (2017)
HexaConv - Hoogeboom*, Peters* et al. (2018)

SE(2): Discrete subgroups
Learning steerable filters for rotation equivariant CNNs - Weiler et al. (2018)
Oriented response networks - Zhou et al. (2017)
Roto-translation covariant convolutions networks for medical image analysis - Bekkers et al. (2018)
Rotation equivariant vector field networks - Marcos et al. (2017)

References

E(3): Discrete subgroups
3D G-CNNs for pulmonary nodule detection - Winkels & Cohen (2018)

SE(3): Discrete subgroups
CubeNet: Equivariance to 3D rotation and translation - Worrall & Brostow (2018)

SE(3): Full group
N-body networks: a covariant hierarchical neural network architecture for learning atomic potentials - Risi (2018)
Tensor Field networks: Rotation- and translation-equivariant neural networks for 3D point clouds - Thomas, Smidt* et al. (2018)
3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data - Weiler*, Geiger* et al. (2018)
Clebsch-Gordan Nets: A Fully Fourier Space Spherical Convolutional Neural Networks - Kondor et al. (2018)
Cormorant: Covariant molecular neural networks - Anderson et al. (2019)

SO(3) on the sphere: discrete subgroup
Spherical CNNs - Cohen*, Geiger*, Köhler et al. (2018)

SO(3) on the sphere: isotropic subgroup
Learning SO(3) Equivariant Representations with Spherical CNNs - Esteves et al. (2018)
DeepSphere: Efficient spherical Convolutions Neural Networks with HEALPix samplings for cosmological applications - Perraudin et al. (2019)

SO(3) on the sphere: full group
Spherical CNNs on unstructured grids - Jiang et al. (2019)

References

Scale semigroup
Deep Scale-spaces: Equivariance Over Scale - Worrall & Welling (2019)

Scale group
Scale equivariance in CNNs with vector fields - Marcos et al (2018)
Scale steerable filters for locally scale-invariant neural networks - Ghosh & Gupta (2019)
Scale-equivariant steerable networks - Sosnovik & Smeulders (2020)

Lie groups:
B-Spline CNNs on Lie Groups - Bekkers (2019)
Generalizing Convolutional Neural Networks for Equivariance to Lie groups on Arbitrary Continuous Data - Finzi et al. (2020)

Reverse-complement symmetry
An equivariant Bayesian convolutional network predicts recombination hotspots and accurately resolves binding motifs - Brown & Lunter (2019)

PDEs
PDE-based Group Equivariant Convolutional Neural Networks - Smets et al. (2020)
Incorporating Symmetry into Deep Dynamics Models for Improved Generalization - Wang*, Walters* et al. (2020)

Attention
Affine Self Convolution - Diaconu & Worrall, (2019)
Attentive Group Equivariant Convolution Networks - Romero et al., (2020)
SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks - Fuchs et al. (2020)

Reinforcement Learning
MDP Homomorphic Networks: Group Symmetries in Reinforcement Learning - van der Pol et al. (2020)

Appendix

Wayang Kulit Show

Group Equivariance Proof

$$[\mathcal{T}_{\mathbf{z}}[\mathbf{F}] * \mathbf{W}](\mathbf{x}) = \sum_{\mathbf{y}\in\mathcal{Y}} \mathcal{T}_{\mathbf{z}}[\mathbf{F}](\mathbf{y})\mathbf{W}(\mathcal{T}_{\mathbf{x}}^{-1}[\mathbf{y}])$$

Write out convolution in full

$$= \sum_{\mathbf{y}\in\mathcal{Y}} \mathbf{F}(\mathcal{T}_{\mathbf{z}^{-1}}[\mathbf{y}])\mathbf{W}(\mathcal{T}_{\mathbf{x}^{-1}}[\mathbf{y}])$$

Expand transformation operator

$$= \sum_{\mathbf{y}'\in\mathcal{Y}} \mathbf{F}(\mathbf{y}')\mathbf{W}(\mathcal{T}_{\mathbf{x}}^{-1}[\mathcal{T}_{\mathbf{z}}[\mathbf{y}']])$$

Change of variables

$$\mathbf{y}' = \mathcal{T}_{\mathbf{z}^{-1}}[\mathbf{y}] \iff \mathbf{y} = \mathcal{T}_{\mathbf{z}}[\mathbf{y}']$$

$$= \sum_{\mathbf{y}'\in\mathcal{Y}} \mathbf{F}(\mathbf{y}')\mathbf{W}(\mathcal{T}_{\mathbf{x}^{-1}\mathbf{z}}[\mathbf{y}'])$$

Rewriting: use the hints

$$= \sum_{\mathbf{y}'\in\mathcal{Y}} \mathbf{F}(\mathbf{y}')\mathbf{W}(\mathcal{T}_{\mathbf{z}^{-1}\mathbf{x}}^{-1}[\mathbf{y}']) = [\mathbf{F} * \mathbf{W}](\mathbf{z}^{-1}\mathbf{x}) = \mathcal{T}_{\mathbf{z}}[\mathbf{F} * \mathbf{W}](\mathbf{x})$$

It's a convolution!

Hints

Homomorphism property	Inversion	Composition
$\mathcal{T}_{\mathbf{z}^{-1}}[\mathbf{y}] = \mathcal{T}_{\mathbf{z}}^{-1}[\mathbf{y}]$	$(\mathbf{x}\mathbf{y})^{-1} = \mathbf{y}^{-1}\mathbf{x}^{-1}$	$\mathcal{T}_{\mathbf{x}}\mathcal{T}_{\mathbf{y}} = \mathcal{T}_{\mathbf{x}\mathbf{y}}$

Semigroup Equivariance Proof

$$[\mathcal{T}_{\mathbf{z}}[\mathbf{F}] \star \mathbf{W}](\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}} \mathcal{T}_{\mathbf{x}}[\mathcal{T}_{\mathbf{z}}[\mathbf{F}]](\mathbf{y})\mathbf{W}(\mathbf{y})$$

Write out convolution in full

$$= \sum_{\mathbf{y} \in \mathcal{Y}} \mathcal{T}_{\mathbf{xz}}[\mathbf{F}](\mathbf{y})\mathbf{W}(\mathbf{y})$$

Semigroup composition

$$= [\mathbf{F} \star \mathbf{W}](\mathbf{xz}) = \mathcal{T}_{\mathbf{z}}[\mathbf{F} \star \mathbf{W}](\mathbf{x})$$

Define this as left action on activations

It's a convolution!