University of Amsterdam

AMLAB Amsterdam Machine Learning Lab

Group Equivariant Deep Learning

Lecture 2 - Steerable group convolutions

Lecture 2.7 - Derivation of Harmonic[1] nets from regular g-convs

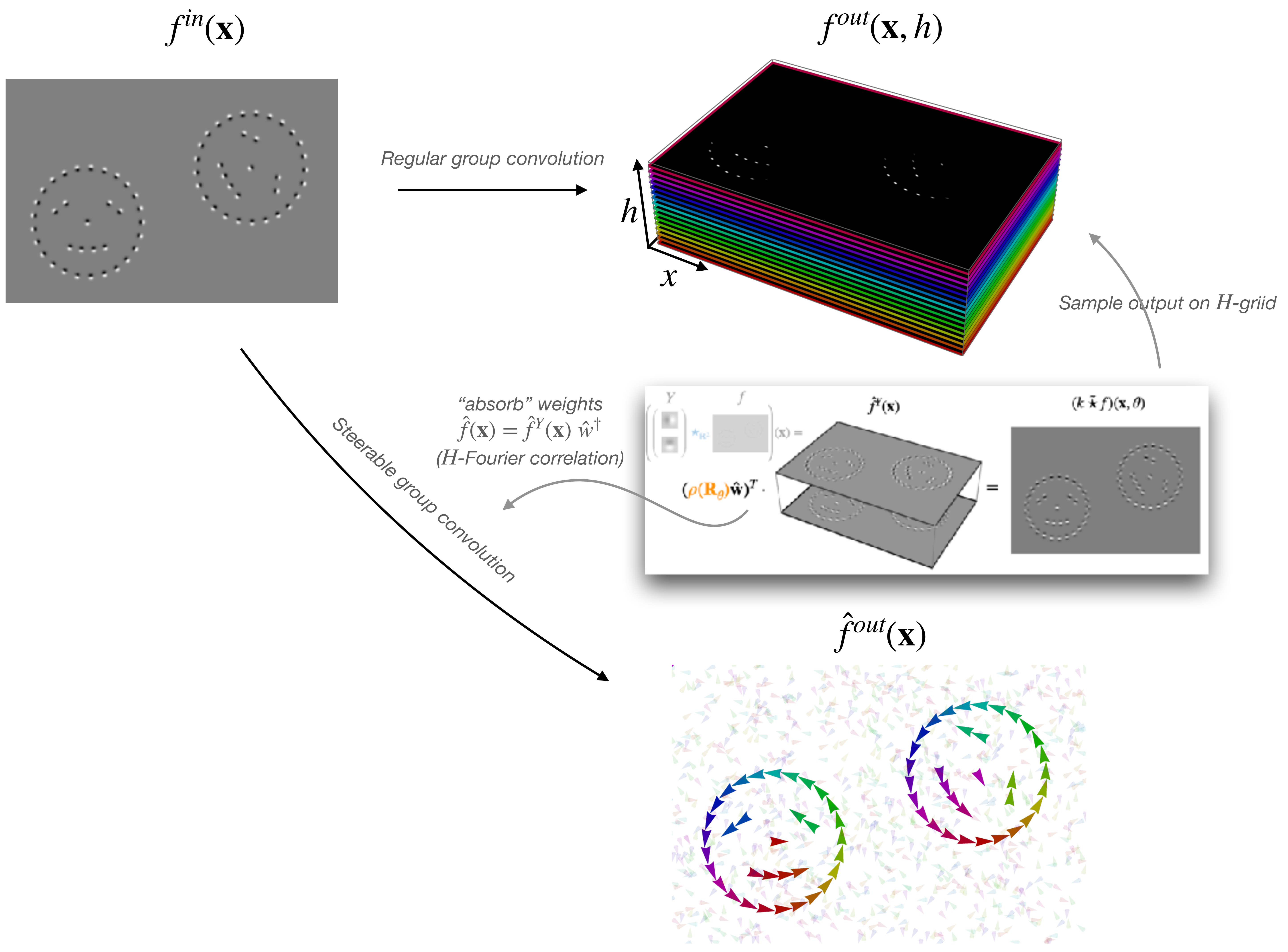
Using complex irreps of $SO(2)$

[1] Worrall, D. E., Garbin, S. J., Turmukhambetov, D., & Brostow, G. J. Harmonic networks: Deep translation and rotation equivariance. CVPR 2017

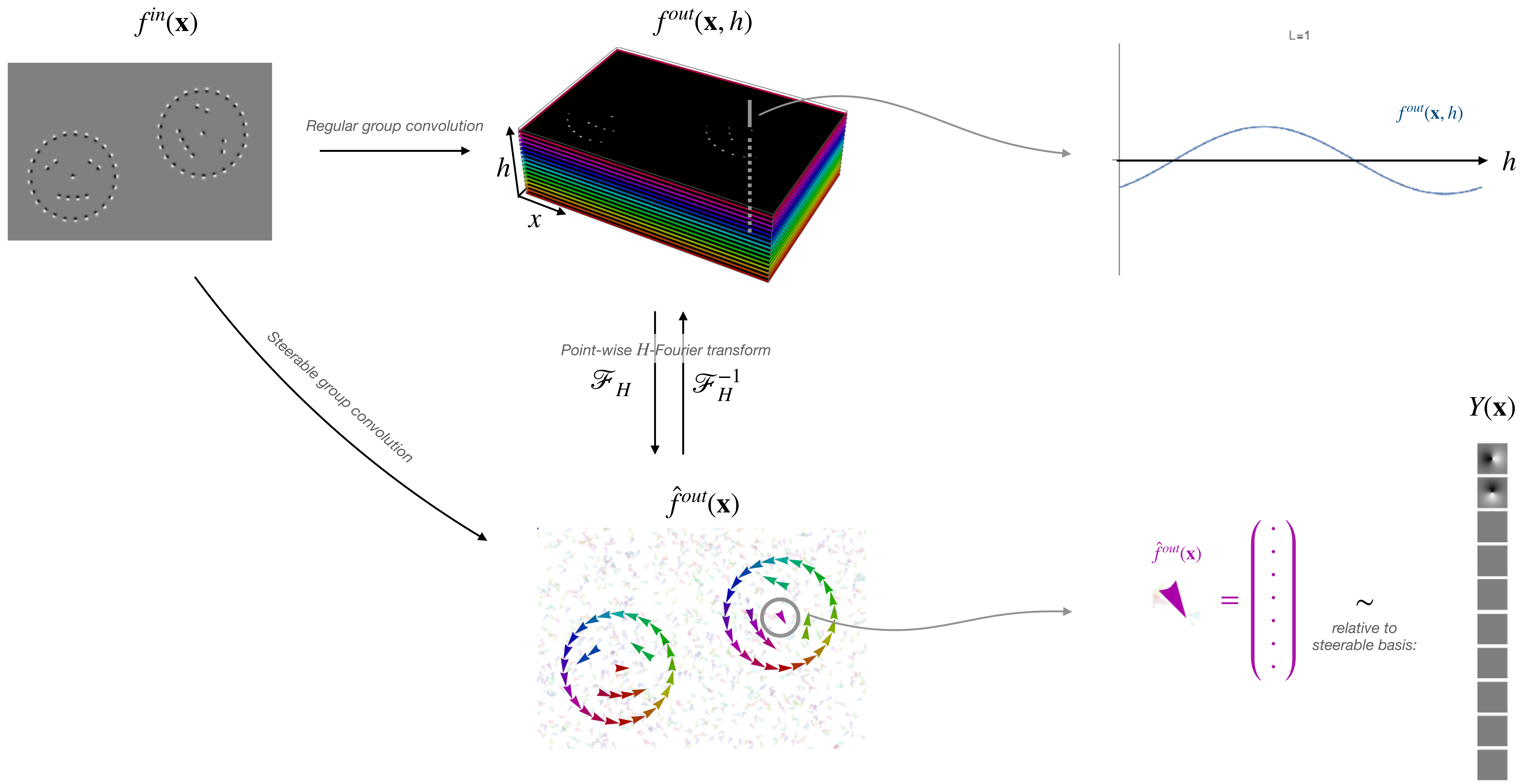
Erik Bekkers, Amsterdam Machine Learning Lab, University of Amsterdam

This mini-course serves as a module with the UvA Master AI course Deep Learning 2 https://uvadl2c.github.io/

From regular to steerable via a Fourier transform



From regular to steerable via a Fourier transform



From regular to steerable via a Fourier transform

Regular group convolutions:
Domain expanded feature maps

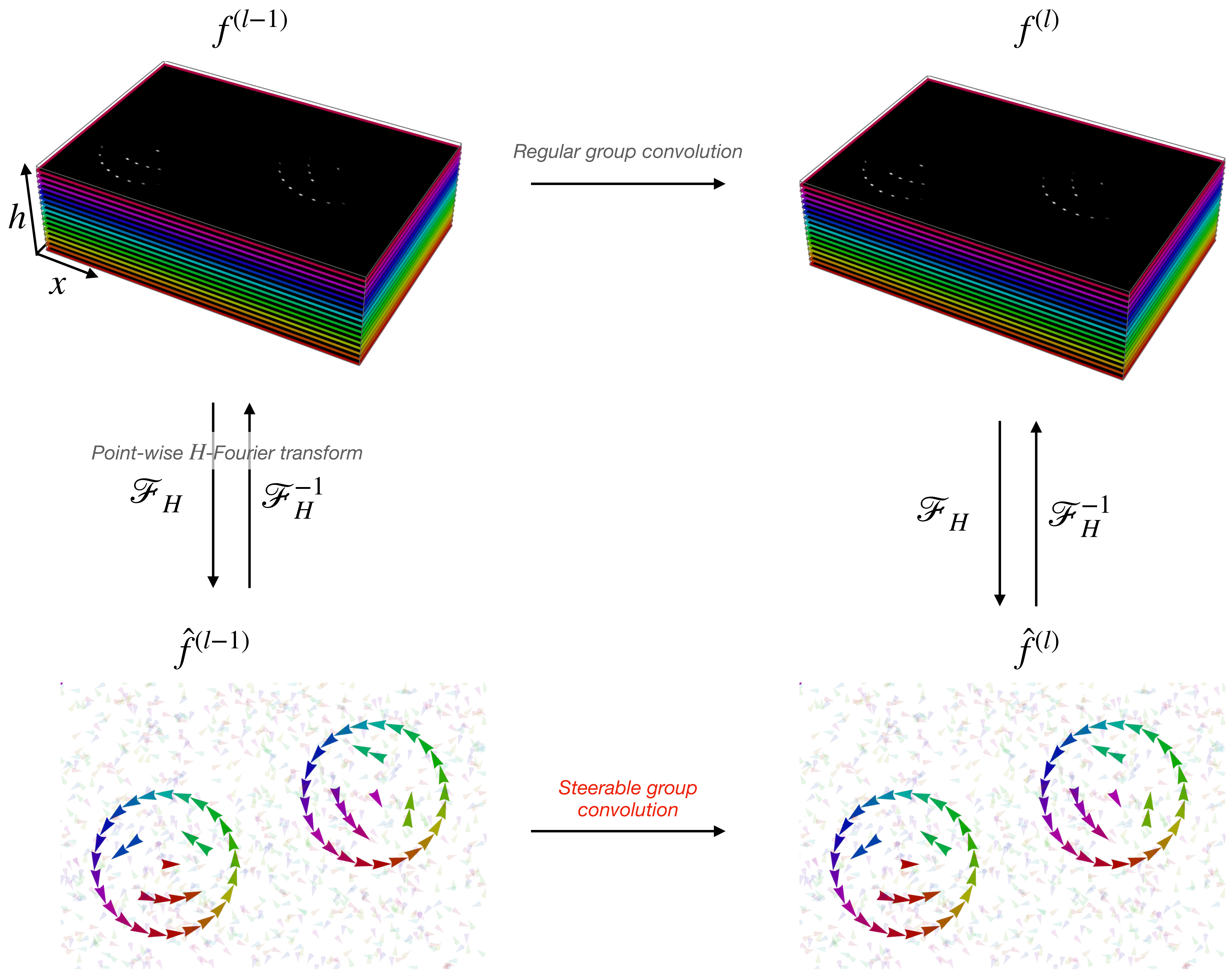
$$f^{(l)} : \mathbb{R}^d \times H \to \mathbb{R}$$

added axis

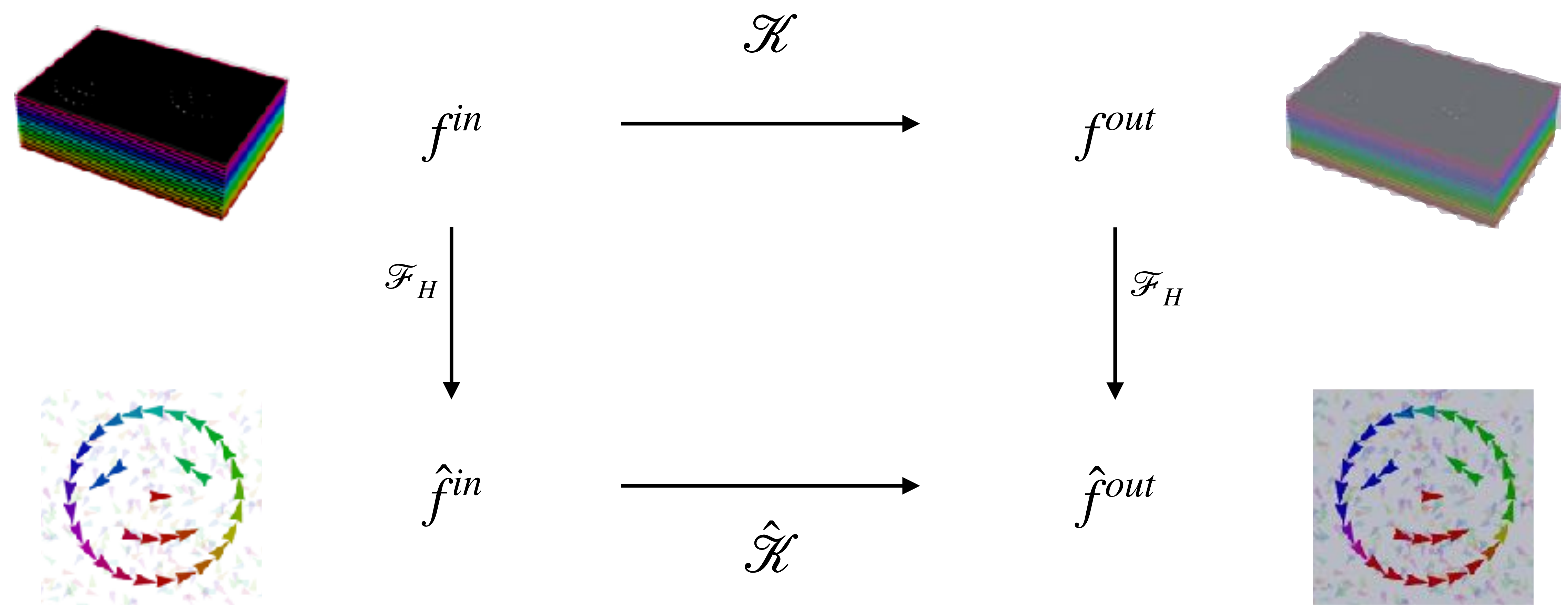
Steerable group convolutions:
Co-domain expanded feature maps (feature fields)

$$\hat{f}^{(l)} : \mathbb{R}^d \to V_H$$

vector field instead of scalar field
(vectors in V_H transform via group H representations)



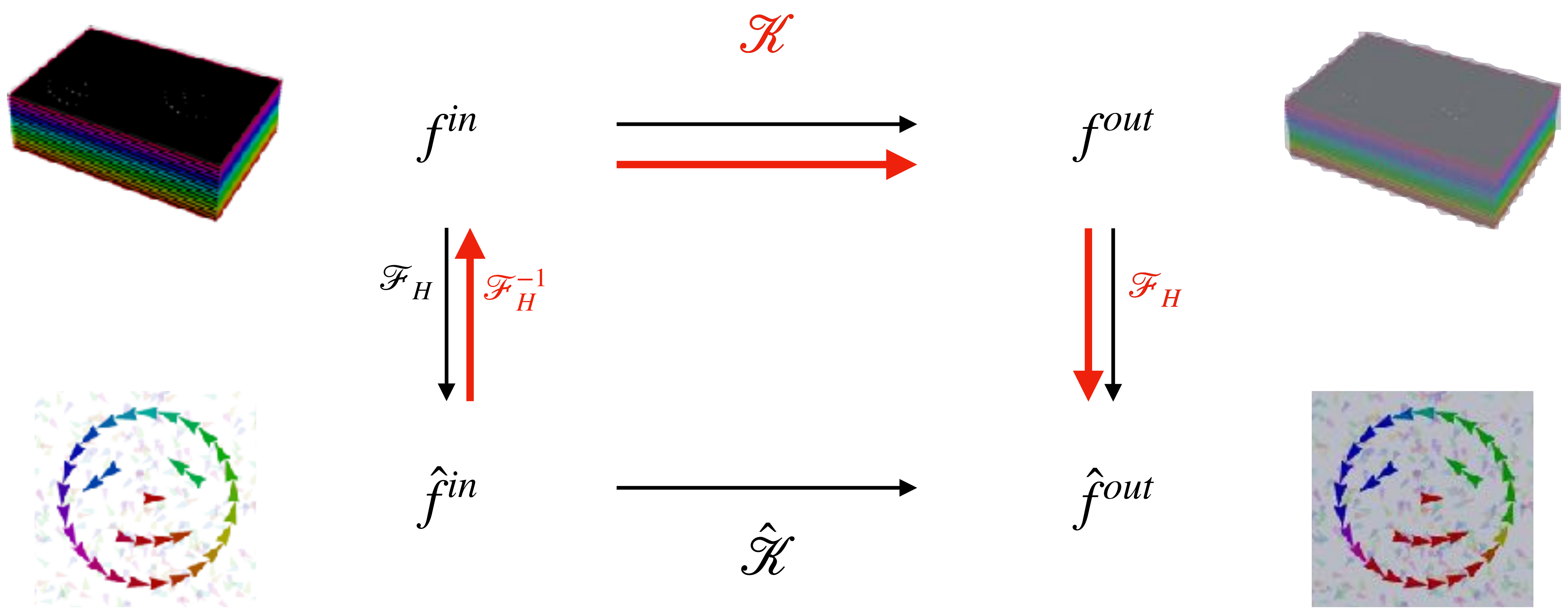
Deriving $\hat{\mathscr{K}}$ from the knowns $(\mathscr{K}, \mathscr{F}_H)$



Instead of solving the kernel constraint, let's compute

$$\hat{\mathscr{K}}(\hat{f}^{in}) = [\mathscr{F}_H \circ \mathscr{K} \circ \mathscr{F}_H^{-1}](\hat{f}^{in})$$

Deriving $\hat{\mathscr{K}}$ from the knowns $(\mathscr{K}, \mathscr{F}_H)$



Instead of solving the kernel constraint, let's compute

$$\hat{\mathscr{K}}(\hat{f}^{in}) = [\mathscr{F}_H \circ \mathscr{K} \circ \mathscr{F}_H^{-1}](\hat{f}^{in})$$

Deriving $\hat{\mathscr{K}}$ from the knowns $(\mathscr{K}, \mathscr{F}_H)$

$$\hat{\mathscr{K}}(\hat{f}) = \mathscr{F}_H(\mathscr{K}(\mathscr{F}_H^{-1}(\hat{f})))$$

Deriving $\hat{\mathscr{K}}$ from the knowns $(\mathscr{K}, \mathscr{F}_H)$

$$\hat{\mathscr{K}}(\hat{f}) = \mathscr{F}_H(\mathscr{K}(\mathscr{F}_H^{-1}(\hat{f})))$$

Deriving $\hat{\mathscr{K}}$ from the knowns ($\mathscr{K}$, $\mathscr{F}_H$)

$$\hat{\mathscr{K}}(\hat{f}) = \mathscr{F}_H(\mathscr{K}(\mathscr{F}_H^{-1}(\hat{f})))$$

$$\hat{\mathscr{K}}(\hat{f})_j(\mathbf{x}) = \int_{S^1} \mathscr{K}(\mathscr{F}_H^{-1}(\hat{f}))(\mathbf{x}, \theta)\, e^{ij\theta} \mathrm{d}\theta$$

Fourier transform: $\mathscr{F}_H(f(\cdot))_j = \int_{S^1} f(\theta) e^{ij\theta} \mathrm{d}\theta$

Deriving $\hat{\mathscr{K}}$ from the knowns $(\mathscr{K}, \mathscr{F}_H)$

$$\hat{\mathscr{K}}(\hat{f}) = \mathscr{F}_H(\mathscr{K}(\mathscr{F}_H^{-1}(\hat{f})))$$

$$\hat{\mathscr{K}}(\hat{f})_j(\mathbf{x}) = \int_{S^1} \mathscr{K}(\mathscr{F}_H^{-1}(\hat{f}))(\mathbf{x}, \theta)\, e^{ij\theta} \mathrm{d}\theta$$

$$= \int_{S^1} \int_{\mathbb{R}^2} \int_{S^1} k(\mathbf{R}_\theta(\mathbf{x}' - \mathbf{x}), \theta' - \theta) \sum_l \hat{f}_l(\mathbf{x}') e^{-il\theta'} \mathrm{d}\mathbf{x}' \mathrm{d}\theta'\, e^{ij\theta} \mathrm{d}\theta$$

Fourier transform: $\mathscr{F}_H(f(\cdot))_j = \int_{S^1} f(\theta) e^{ij\theta} \mathrm{d}\theta$

Regular group conv: $\mathscr{K}(f)(g) = \int_G k(g^{-1}g') f(g') \mathrm{d}g'$

Inverse Fourier trafo: $\mathscr{F}_H^{-1}(\hat{f})(h) = \sum_l \hat{f}_l e^{-il\theta'}$

Deriving $\hat{\mathscr{K}}$ from the knowns $(\mathscr{K}, \mathscr{F}_H)$

$$\hat{\mathscr{K}}(\hat{f}) = \mathscr{F}_H(\mathscr{K}(\mathscr{F}_H^{-1}(\hat{f})))$$

$$\hat{\mathscr{K}}(\hat{f})_j(\mathbf{x}) = \int_{S^1} \mathscr{K}(\mathscr{F}_H^{-1}(\hat{f}))(\mathbf{x}, \theta)\, e^{ij\theta} \mathrm{d}\theta$$

$$= \int_{S^1} \int_{\mathbb{R}^2} \int_{S^1} k(\mathbf{R}_\theta(\mathbf{x}' - \mathbf{x}), \theta' - \theta) \sum_l \hat{f}_l(\mathbf{x}') e^{-il\theta'}\, \mathrm{d}\mathbf{x}' \mathrm{d}\theta'\, e^{ij\theta} \mathrm{d}\theta$$

$$= \int_{\mathbb{R}^2} \sum_l \int_{S^1} \int_{S^1} k(\mathbf{R}_\theta(\mathbf{x}' - \mathbf{x}), \theta' - \theta)\, e^{-il\theta'}\, e^{ij\theta} \mathrm{d}\theta\, \mathrm{d}\theta'\, \hat{f}_l(\mathbf{x}')\, \mathrm{d}\mathbf{x}'$$

Fourier transform: $\mathscr{F}_H(f(\cdot))_j = \int_{S^1} f(\theta) e^{ij\theta} \mathrm{d}\theta$

Regular group conv: $\mathscr{K}(f)(g) = \int_G k(g^{-1}g') f(g') \mathrm{d}g'$

Inverse Fourier trafo: $\mathscr{F}_H^{-1}(\hat{f})(h) = \sum_l \hat{f}_l e^{-il\theta'}$

Deriving $\hat{\mathscr{K}}$ from the knowns $(\mathscr{K}, \mathscr{F}_H)$

$$\hat{\mathscr{K}}(\hat{f}) = \mathscr{F}_H(\mathscr{K}(\mathscr{F}_H^{-1}(\hat{f})))$$

$$\hat{\mathscr{K}}(\hat{f})_j(\mathbf{x}) = \int_{S^1} \mathscr{K}(\mathscr{F}_H^{-1}(\hat{f}))(\mathbf{x}, \theta)\, e^{ij\theta} \mathrm{d}\theta$$

$$= \int_{S^1} \int_{\mathbb{R}^2} \int_{S^1} k(\mathbf{R}_\theta(\mathbf{x}' - \mathbf{x}), \theta' - \theta) \sum_l \hat{f}_l(\mathbf{x}') e^{-il\theta'} \mathrm{d}\mathbf{x}' \mathrm{d}\theta'\, e^{ij\theta} \mathrm{d}\theta$$

$$= \int_{\mathbb{R}^2} \sum_l \int_{S^1} \int_{S^1} k(\mathbf{R}_\theta(\mathbf{x}' - \mathbf{x}), \theta' - \theta)\, e^{-il\theta'}\, e^{ij\theta} \mathrm{d}\theta\, \mathrm{d}\theta'\, \hat{f}_l(\mathbf{x}')\, \mathrm{d}\mathbf{x}'$$

Fourier transform: $\mathscr{F}_H(f(\cdot))_j = \int_{S^1} f(\theta) e^{ij\theta} \mathrm{d}\theta$

Regular group conv: $\mathscr{K}(f)(g) = \int_G k(g^{-1}g') f(g') \mathrm{d}g'$

Inverse Fourier trafo: $\mathscr{F}_H^{-1}(\hat{f})(h) = \sum_l \hat{f}_l e^{-il\theta'}$

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

A normal conv with a kernel $\hat{k} : \mathbb{R}^d \to \mathbb{R}^{d_{out} \times d_{in}}$!

(recall lecture 2.5)

Deriving $\hat{\mathcal{K}}$ from the knowns $(\mathcal{K}, \mathcal{F}_H)$

So we "just" need to compute

$$\hat{k}_{jl}(\mathbf{x}' - \mathbf{x}) = \int_{S^1}\int_{S^1} k(\mathbf{R}_\theta(\mathbf{x}'-\mathbf{x}), \theta'-\theta)\; e^{-il\theta'}\; e^{ij\theta}\mathrm{d}\theta\;\mathrm{d}\theta'$$

Deriving $\hat{\mathscr{K}}$ from the knowns $(\mathscr{K}, \mathscr{F}_H)$

So we “just” need to compute

$$\hat{k}_{jl}(\mathbf{x}' - \mathbf{x}) = \int_{S^1} \int_{S^1} k(\mathbf{R}_\theta(\mathbf{x}' - \mathbf{x}), \theta' - \theta)\, e^{-il\theta'}\, e^{ij\theta} \mathrm{d}\theta\, \mathrm{d}\theta'$$

Recall that we given enough frequencies we can expand any spatial kernel in circular harmonics (lecture 2.1)

$$k(\mathbf{x}, \theta) = \sum_J w_J(r, \theta) e^{iJ\alpha}$$

Deriving $\hat{\mathcal{K}}$ from the knowns ($\mathcal{K}, \mathcal{F}_H$)

So we "just" need to compute

$$\hat{k}_{jl}(\mathbf{x}' - \mathbf{x}) = \int_{S^1}\int_{S^1} k(\mathbf{R}_\theta(\mathbf{x}'-\mathbf{x}), \theta'-\theta)\; e^{-il\theta'}\; e^{ij\theta}\mathrm{d}\theta\; \mathrm{d}\theta'$$

Recall that we given enough frequencies we can expand any spatial kernel in circular harmonics (lecture 2.1)

$$k(\mathbf{x}, \theta) = \sum_J w_J(r, \theta)e^{iJ\alpha}$$

Such kernels are spatially rotation steerable via

$$\begin{aligned} k(\mathbf{R}_\theta^{-1}\mathbf{x}, \theta' - \theta) &= \sum_J w_J(r, \theta' - \theta)e^{iJ(\alpha-\theta)} \\ &= e^{-iJ\theta}\sum_J w_J(r, \theta' - \theta)e^{iJ\alpha} \\ &= e^{-iJ\theta}k(\mathbf{x}, \theta' - \theta) \end{aligned}$$

Deriving $\hat{\mathscr{K}}$ from the knowns ($\mathscr{K}, \mathscr{F}_H$)

So we “just” need to compute

$$\hat{k}_{jl}(\mathbf{x}' - \mathbf{x}) = \int_{S^1}\int_{S^1} k(\mathbf{R}_\theta(\mathbf{x}'-\mathbf{x}), \theta'-\theta)\, e^{-il\theta'}\, e^{ij\theta} \mathrm{d}\theta\, \mathrm{d}\theta'$$

Recall that we given enough frequencies we can expand any spatial kernel in circular harmonics (lecture 2.1)

$$k(\mathbf{x}, \theta) = \sum_J w_J(r, \theta) e^{iJ\alpha}$$

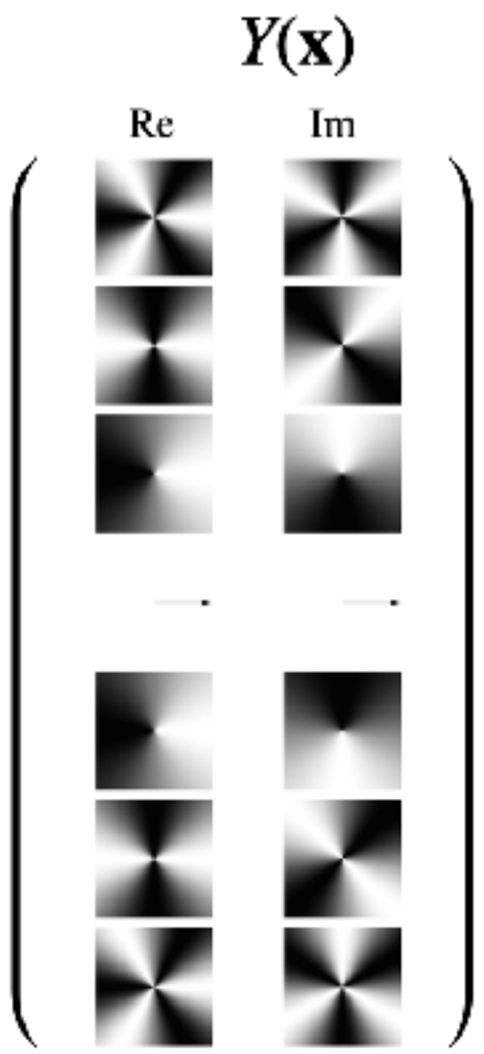
Such kernels are spatially rotation steerable via

$$\begin{aligned} k(\mathbf{R}_\theta^{-1}\mathbf{x}, \theta' - \theta) &= \sum_J w_J(r, \theta' - \theta) e^{iJ(\alpha - \theta)} \\ &= e^{-iJ\theta} \sum_J w_J(r, \theta' - \theta) e^{iJ\alpha} \\ &= e^{-iJ\theta} k(\mathbf{x}, \theta' - \theta) \end{aligned}$$

Deriving $\hat{\mathscr{K}}$ from the knowns $(\mathscr{K}, \mathscr{F}_H)$

So we "just" need to compute

$$\hat{k}_{jl}(\mathbf{x}' - \mathbf{x}) = \int_{S^1}\int_{S^1} e^{-iJ\theta} \sum_J w_J(\|\mathbf{x}' - \mathbf{x}\|, \theta' - \theta) e^{iJ\alpha_{\mathbf{x}'-\mathbf{x}}} e^{-il\theta'} e^{ij\theta} \mathrm{d}\theta \, \mathrm{d}\theta'$$



Deriving $\hat{\mathscr{K}}$ from the knowns $(\mathscr{K}, \mathscr{F}_H)$

So we "just" need to compute

$$\hat{k}_{jl}(\mathbf{x}' - \mathbf{x}) = \int_{S^1}\int_{S^1}\sum_J w_J(\|\mathbf{x}' - \mathbf{x}\|, \theta' - \theta)\, e^{iJ\alpha_{\mathbf{x}'-\mathbf{x}}}\, e^{-il\theta'}\, e^{-i(J-j)\theta}\mathrm{d}\theta\, \mathrm{d}\theta'$$

Deriving $\hat{\mathcal{K}}$ from the knowns $(\mathcal{K}, \mathcal{F}_H)$

So we "just" need to compute

$$\hat{k}_{jl}(\mathbf{x}' - \mathbf{x}) = \int_{S^1} \int_{S^1} \sum_J w_J(\|\mathbf{x}' - \mathbf{x}\|, \theta' - \theta)\, e^{iJ\alpha_{\mathbf{x}'-\mathbf{x}}}\, e^{-il\theta'}\, e^{-i(J-j)\theta} \mathrm{d}\theta\, \mathrm{d}\theta'$$

$$= \int_{S^1} \sum_J \overline{\hat{w}_J}(\|\mathbf{x}' - \mathbf{x}\|)\, e^{iJ\alpha_{\mathbf{x}'-\mathbf{x}}}\, e^{-il\theta}\, e^{-i(J-j)\theta} \mathrm{d}\theta$$

Fourier trafo + shift theorem
(reflection ↔ conjugate)

Deriving $\hat{\mathscr{K}}$ from the knowns $(\mathscr{K}, \mathscr{F}_H)$

So we "just" need to compute

$$\hat{k}_{jl}(\mathbf{x}' - \mathbf{x}) = \int_{S^1} \int_{S^1} \sum_J w_J(\|\mathbf{x}' - \mathbf{x}\|, \theta' - \theta)\, e^{iJ\alpha_{\mathbf{x}'-\mathbf{x}}}\, e^{-il\theta'}\, e^{-i(J-j)\theta} \mathrm{d}\theta\, \mathrm{d}\theta'$$

$$= \int_{S^1} \sum_J \overline{\hat{w}_J}(\|\mathbf{x}' - \mathbf{x}\|)\, e^{iJ\alpha_{\mathbf{x}'-\mathbf{x}}}\, e^{-il\theta}\, e^{-i(J-j)\theta} \mathrm{d}\theta$$

Fourier trafo + shift theorem
(reflection ↔ conjugate)

$$= \sum_J \overline{\hat{w}_J}(\|\mathbf{x}' - \mathbf{x}\|)\, e^{iJ\alpha_{\mathbf{x}'-\mathbf{x}}} \int_{S^1} e^{-i(l+J-j)\theta} \mathrm{d}\theta$$

$$\int_{S^1} e^{-i(l+J-j)\theta} \mathrm{d}\theta = \begin{cases} 2\pi & \text{if } \; l + J - j = 0 \\ 0 & \text{otherwise} \end{cases}$$

Deriving $\hat{\mathscr{K}}$ from the knowns $(\mathscr{K}, \mathscr{F}_H)$

So we "just" need to compute

$$\hat{k}_{jl}(\mathbf{x}' - \mathbf{x}) = \int_{S^1}\int_{S^1}\sum_J w_J(\|\mathbf{x}' - \mathbf{x}\|, \theta' - \theta)\, e^{iJ\alpha_{\mathbf{x}'-\mathbf{x}}}\, e^{-il\theta'}\, e^{-i(J-j)\theta} \mathrm{d}\theta\, \mathrm{d}\theta'$$

$$= \int_{S^1}\sum_J \overline{\hat{w}_J}(\|\mathbf{x}' - \mathbf{x}\|)\, e^{iJ\alpha_{\mathbf{x}'-\mathbf{x}}}\, e^{-il\theta}\, e^{-i(J-j)\theta} \mathrm{d}\theta$$

Fourier trafo + shift theorem

(reflection ↔ conjugate)

$$= \sum_J \overline{\hat{w}_J}(\|\mathbf{x}' - \mathbf{x}\|)\, e^{iJ\alpha_{\mathbf{x}'-\mathbf{x}}} \int_{S^1} e^{-i(l+J-j)\theta} \mathrm{d}\theta$$

$$\int_{S^1} e^{-i(l+J-j)\theta} \mathrm{d}\theta = \begin{cases} 2\pi & \text{if } l + J - j = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$= \sum_{J=j-l} \overline{\hat{w}_J}(\|\mathbf{x}' - \mathbf{x}\|)\, Y_J(\alpha_{\mathbf{x}'-\mathbf{x}})$$

Deriving $\hat{\mathscr{K}}$ from the knowns $(\mathscr{K}, \mathscr{F}_H)$

So we "just" need to compute

$$\hat{k}_{jl}(\mathbf{x}'-\mathbf{x}) = \int_{S^1}\int_{S^1}\sum_J w_J(\|\mathbf{x}'-\mathbf{x}\|, \theta'-\theta)\, e^{iJ\alpha_{\mathbf{x}'-\mathbf{x}}}\, e^{-il\theta'}\, e^{-i(J-j)\theta}\mathrm{d}\theta\, \mathrm{d}\theta'$$

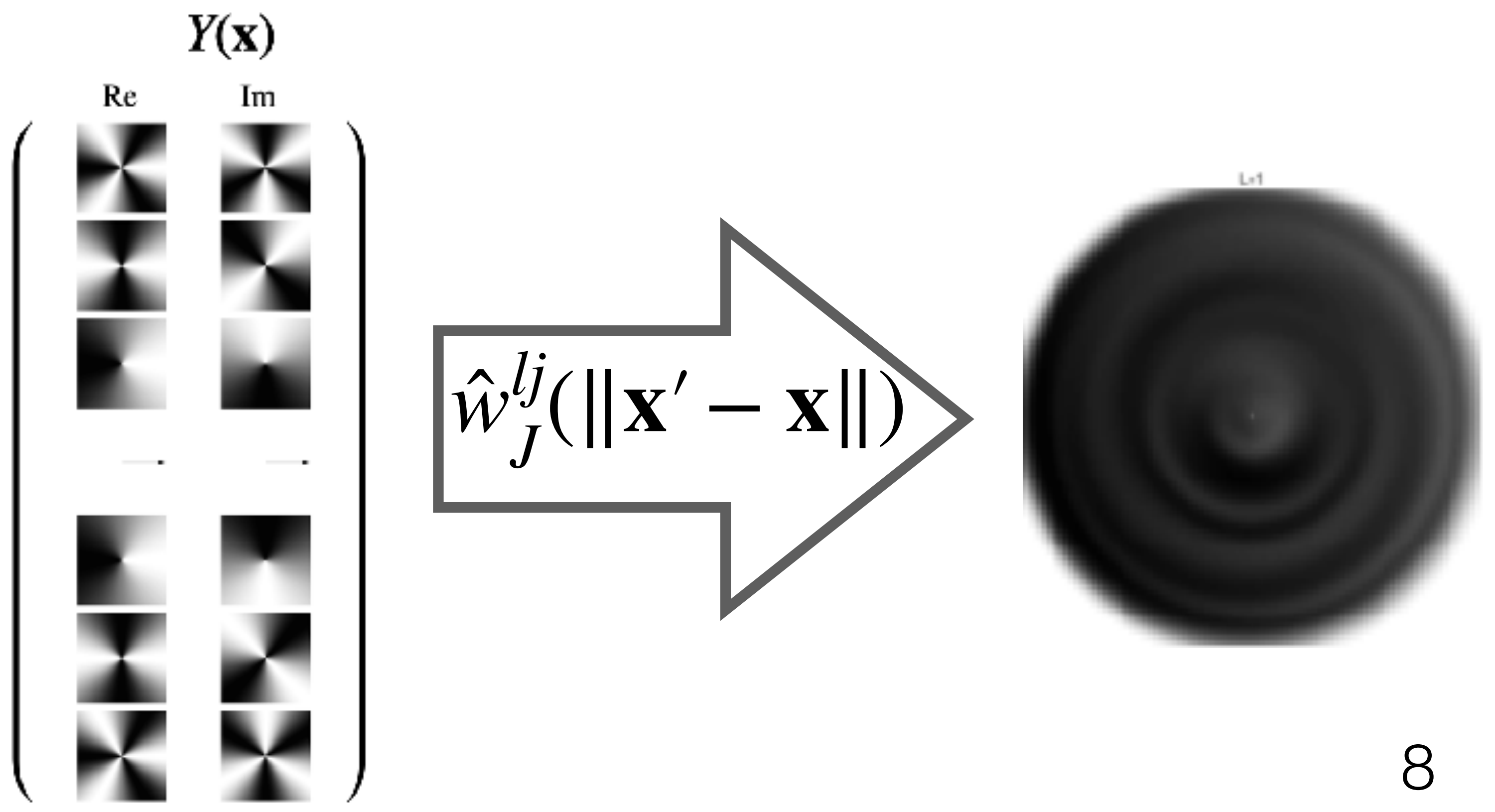
$$= \int_{S^1}\sum_J \overline{\hat{w}}_J(\|\mathbf{x}'-\mathbf{x}\|)\, e^{iJ\alpha_{\mathbf{x}'-\mathbf{x}}}\, e^{-il\theta}\, e^{-i(J-j)\theta}\mathrm{d}\theta$$

Fourier trafo + shift theorem
(reflection ↔ conjugate)

$$= \sum_J \overline{\hat{w}}_J(\|\mathbf{x}'-\mathbf{x}\|)\, e^{iJ\alpha_{\mathbf{x}'-\mathbf{x}}} \int_{S^1} e^{-i(l+J-j)\theta}\mathrm{d}\theta$$

$$\int_{S^1} e^{-i(l+J-j)\theta}\mathrm{d}\theta = \begin{cases} 2\pi & \text{if } \; l+J-j=0 \\ 0 & \text{otherwise} \end{cases}$$

$$= \sum_{J=j-l} \overline{\hat{w}}_J(\|\mathbf{x}'-\mathbf{x}\|)\, Y_J(\alpha_{\mathbf{x}'-\mathbf{x}})$$

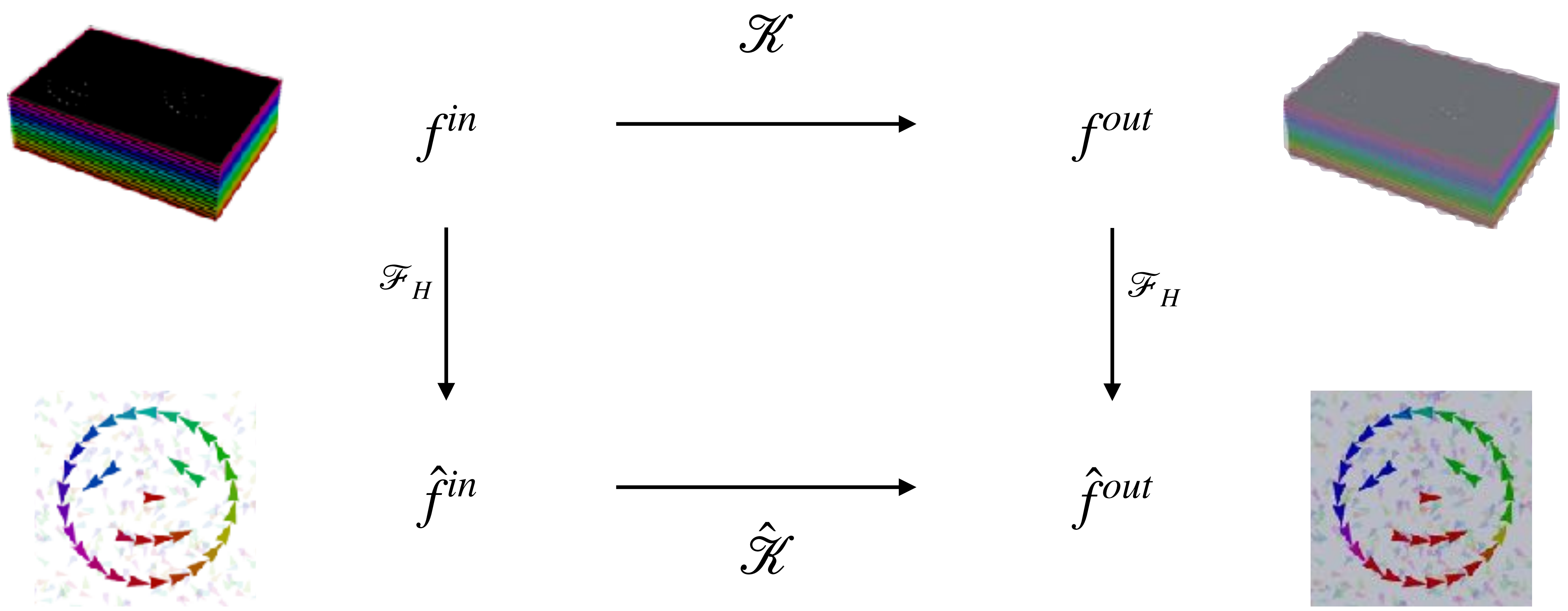


Deriving $\hat{\mathscr{K}}$ from the knowns $(\mathscr{K}, \mathscr{F}_H)$

$$\mathscr{K}(f)(g) = \int_G k(g^{-1}g')f(g')\mathrm{d}g'$$

with kernel $k(\mathbf{x}, \theta) = \sum_l \sum_{J=j-l} \overline{\hat{w}}_J(\|\mathbf{x}' - \mathbf{x}\|)\, Y_J(\alpha_{\mathbf{x}'-\mathbf{x}})Y_l(\theta)$

Regular group convolutions



Steerable group convolutions

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

with kernel $\hat{k}_{jl}(\mathbf{x}) = \sum_{J=j-l} \hat{w}_J(\|\mathbf{x}' - \mathbf{x}\|)\, Y_J(\alpha_{\mathbf{x}'-\mathbf{x}})$

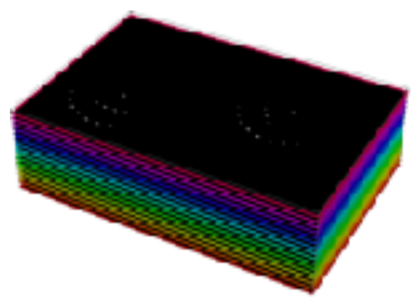
[1] Worrall, D. E., Garbin, S. J., Turmukhambetov, D., & Brostow, G. J. Harmonic networks: Deep translation and rotation equivariance. CVPR 2017

Deriving $\hat{\mathscr{K}}$ from the knowns ($\mathscr{K}, \mathscr{F}_H$)

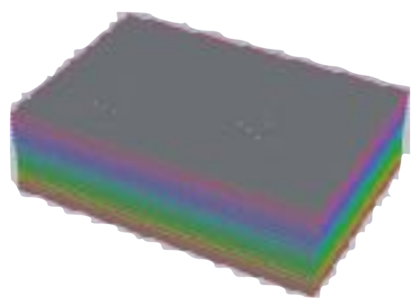
$$\mathscr{K}(f)(g) = \int_G k(g^{-1}g')f(g')\mathrm{d}g'$$

with kernel $k(\mathbf{x}, \theta) = \sum_{l} \sum_{J=j-l} \overline{\hat{w}}_J(\|\mathbf{x}' - \mathbf{x}\|)\, Y_J(\alpha_{\mathbf{x}'-\mathbf{x}}) Y_l(\theta)$

Regular group convolutions

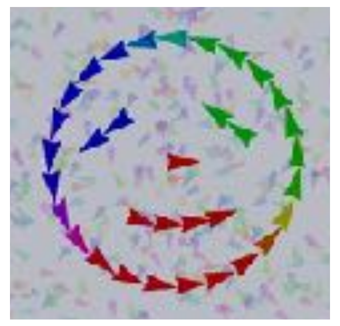


$f^{in} \xrightarrow{\mathscr{K}} f^{out}$



$f^{in} \downarrow \mathscr{F}_H$ $\qquad$ $f^{out} \downarrow \mathscr{F}_H$

$\hat{f}^{in} \xrightarrow{\hat{\mathscr{K}}} \hat{f}^{out}$



$\uparrow \mathscr{F}_H^{-1}$

Given specified:
- input band-limit $l \leq l_{max}$
- output band-limit $j \leq j_{max}$
- steerable basis $Y_l(\theta) = e^{il\theta}$

Steerable group convolutions

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

with kernel $\hat{k}_{jl}(\mathbf{x}) = \sum_{J=j-l} \hat{w}_J(\|\mathbf{x}' - \mathbf{x}\|)\, Y_J(\alpha_{\mathbf{x}'-\mathbf{x}})$

[1] Worrall, D. E., Garbin, S. J., Turmukhambetov, D., & Brostow, G. J. Harmonic networks: Deep translation and rotation equivariance. CVPR 2017

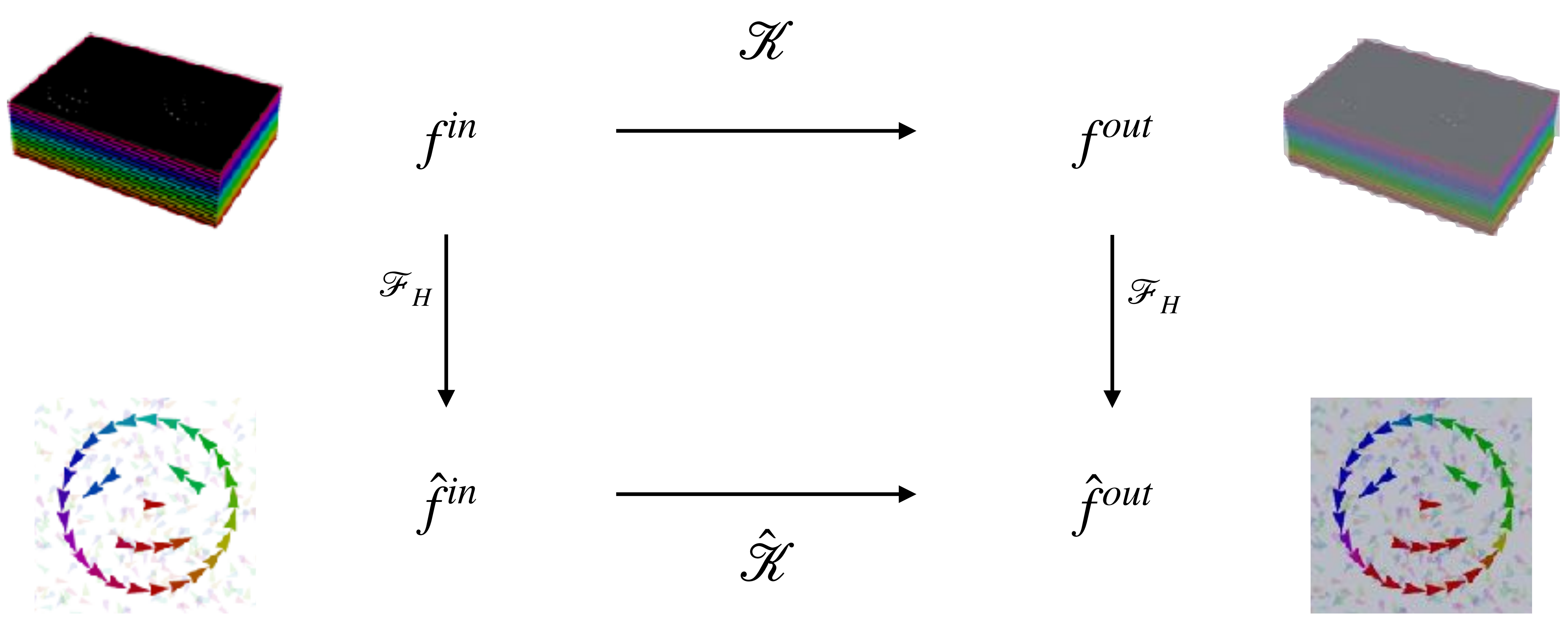
Deriving $\hat{\mathcal{K}}$ from the knowns $(\mathcal{K}, \mathcal{F}_H)$

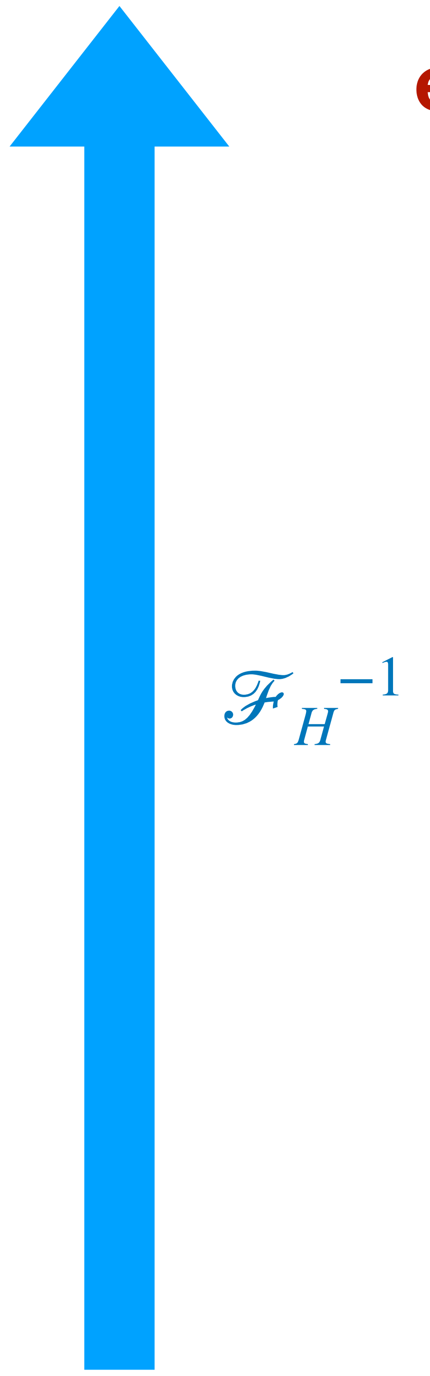
$$\mathcal{K}(f)(g) = \int_G k(g^{-1}g')f(g')\mathrm{d}g'$$

with kernel $k(\mathbf{x}, \theta) = \sum_l \sum_{J=j-l} \overline{\hat{w}}_J(\|\mathbf{x}' - \mathbf{x}\|)\, Y_J(\alpha_{\mathbf{x}'-\mathbf{x}}) Y_l(\theta)$

equivariance condition of harmonic nets!

Regular group convolutions





Given specified:

- input band-limit $l \leq l_{max}$
- output band-limit $j \leq j_{max}$
- steerable basis $Y_l(\theta) = e^{il\theta}$

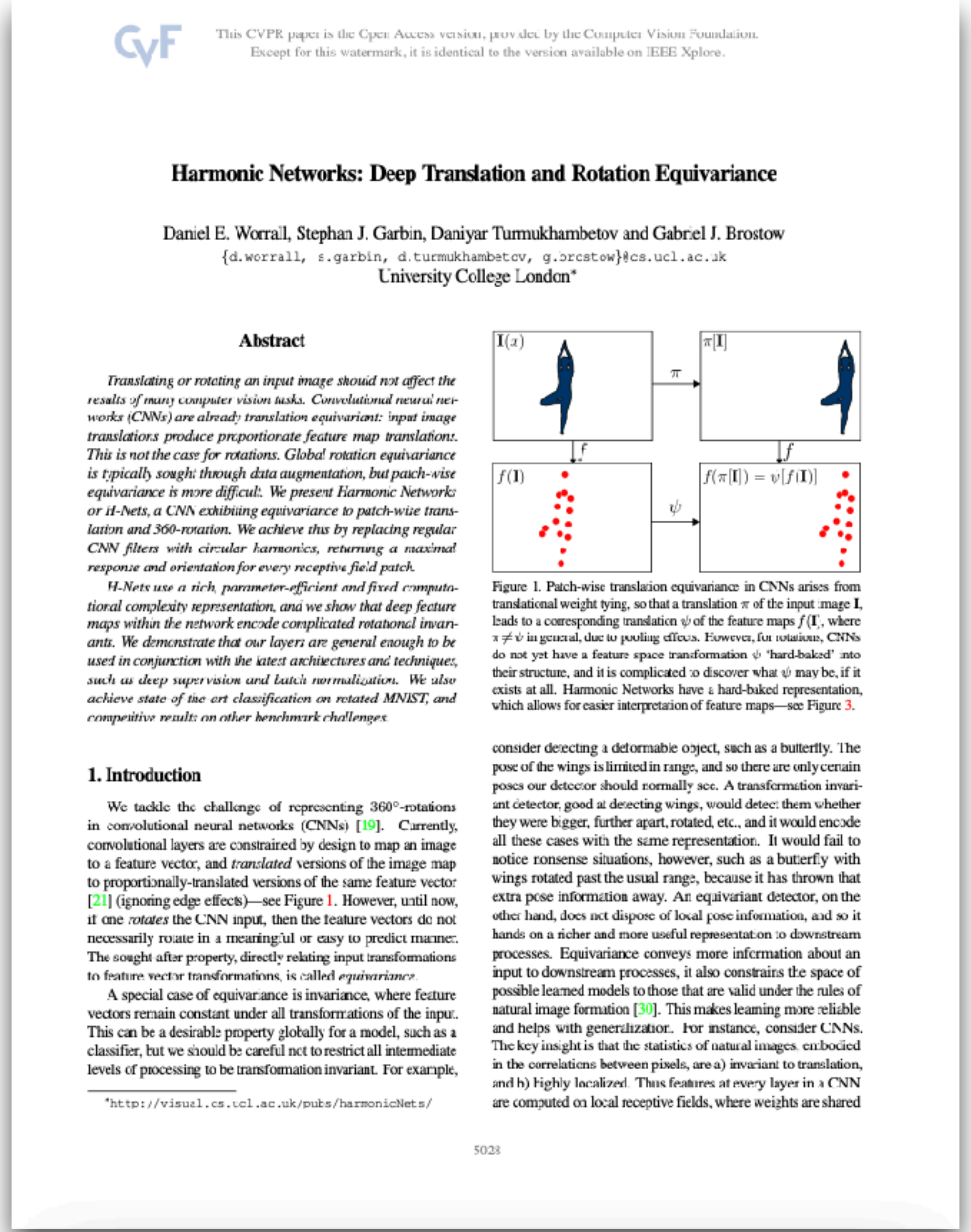
Steerable group convolutions

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

with kernel $\hat{k}_{jl}(\mathbf{x}) = \sum_{J=j-l} \hat{w}_J(\|\mathbf{x}' - \mathbf{x}\|)\, Y_J(\alpha_{\mathbf{x}'-\mathbf{x}})$

[1] Worrall, D. E., Garbin, S. J., Turmukhambetov, D., & Brostow, G. J. Harmonic networks: Deep translation and rotation equivariance. CVPR 2017

Harmonic networks



Harmonic Networks: Deep Translation and Rotation Equivariance

Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov and Gabriel J. Brostow
{d.worrall, s.garbin, d.turmukhambetov, g.brostow}@cs.ucl.ac.uk
University College London*

Abstract

Translating or rotating an input image should not affect the results of many computer vision tasks. Convolutional neural networks (CNNs) are already translation equivariant: input image translations produce proportionate feature map translations. This is not the case for rotations. Global rotation equivariance is typically sought through data augmentation, but patch-wise equivariance is more difficult. We present Harmonic Networks or H-Nets, a CNN exhibiting equivariance to patch-wise translation and 360-rotation. We achieve this by replacing regular CNN filters with circular harmonics, returning a maximal response and orientation for every receptive field patch.

H-Nets use a rich, parameter-efficient and fixed computational complexity representation, and we show that deep feature maps within the network encode complicated rotational invariants. We demonstrate that our layers are general enough to be used in conjunction with the latest architectures and techniques, such as deep supervision and batch normalization. We also achieve state of the art classification on rotated MNIST, and competitive results on other benchmark challenges.

Figure 1. Patch-wise translation equivariance in CNNs arises from translational weight tying, so that a translation π of the input image $\mathbf{I}$, leads to a corresponding translation ψ of the feature maps $f(\mathbf{I})$, where $\pi \neq \psi$ in general, due to pooling effects. However, for rotations, CNNs do not yet have a feature space transformation ψ 'hard-baked' into their structure, and it is complicated to discover what ψ may be, if it exists at all. Harmonic Networks have a hard-baked representation, which allows for easier interpretation of feature maps—see Figure 3.

1. Introduction

We tackle the challenge of representing 360°-rotations in convolutional neural networks (CNNs) [19]. Currently, convolutional layers are constrained by design to map an image to a feature vector, and *translated* versions of the image map to proportionally-translated versions of the same feature vector [21] (ignoring edge effects)—see Figure 1. However, until now, if one *rotates* the CNN input, then the feature vectors do not necessarily rotate in a meaningful or easy to predict manner. The sought after property, directly relating input transformations to feature vector transformations, is called *equivariance*.

A special case of equivariance is invariance, where feature vectors remain constant under all transformations of the input. This can be a desirable property globally for a model, such as a classifier, but we should be careful not to restrict all intermediate levels of processing to be transformation invariant. For example, consider detecting a deformable object, such as a butterfly. The pose of the wings is limited in range, and so there are only certain poses our detector should normally see. A transformation invariant detector, good at detecting wings, would detect them whether they were bigger, further apart, rotated, etc., and it would encode all these cases with the same representation. It would fail to notice nonsense situations, however, such as a butterfly with wings rotated past the usual range, because it has thrown that extra pose information away. An equivariant detector, on the other hand, does not dispose of local pose information, and so it hands on a richer and more useful representation to downstream processes. Equivariance conveys more information about an input to downstream processes, it also constrains the space of possible learned models to those that are valid under the rules of natural image formation [30]. This makes learning more reliable and helps with generalization. For instance, consider CNNs. The key insight is that the statistics of natural images, embodied in the correlations between pixels, are a) invariant to translation, and b) highly localized. Thus features at every layer in a CNN are computed on local receptive fields, where weights are shared

*http://visual.cs.ucl.ac.uk/pubs/harmonicNets/

Steerable G-CNNs as Clebsch-Gordan networks

$$\hat{\mathcal{K}}(\hat{f})(\mathbf{x}) = \int_{\mathbb{R}^d} \sum_{l} \sum_{J=j-l} \hat{w}_J(\|\mathbf{x}' - \mathbf{x}\|)\, Y_J(\alpha_{\mathbf{x}'-\mathbf{x}})\hat{f}_l(\mathbf{x}')\mathrm{d}\mathbf{x}'$$

Steerable G-CNNs as Clebsch-Gordan networks

$$\hat{\mathcal{K}}(\hat{f})(\mathbf{x}) = \int_{\mathbb{R}^d} \sum_{l} \sum_{J} \hat{w}^{j}_{Jl}(\|\mathbf{x}' - \mathbf{x}\|)\, Y_J(\alpha_{\mathbf{x}'-\mathbf{x}})\hat{f}_l(\mathbf{x}')\mathrm{d}\mathbf{x}'$$

Steerable G-CNNs as Clebsch-Gordan networks

$$\hat{\mathscr{K}}(\hat{f})(\mathbf{x}) = \int_{\mathbb{R}^d} \hat{f} \otimes_{CG}^{\hat{w}(\|\mathbf{x}'-\mathbf{x}\|)} Y(\alpha_{\mathbf{x}'-\mathbf{x}}) \, \mathrm{d}\mathbf{x}'$$