Group Equivariant Deep Learning

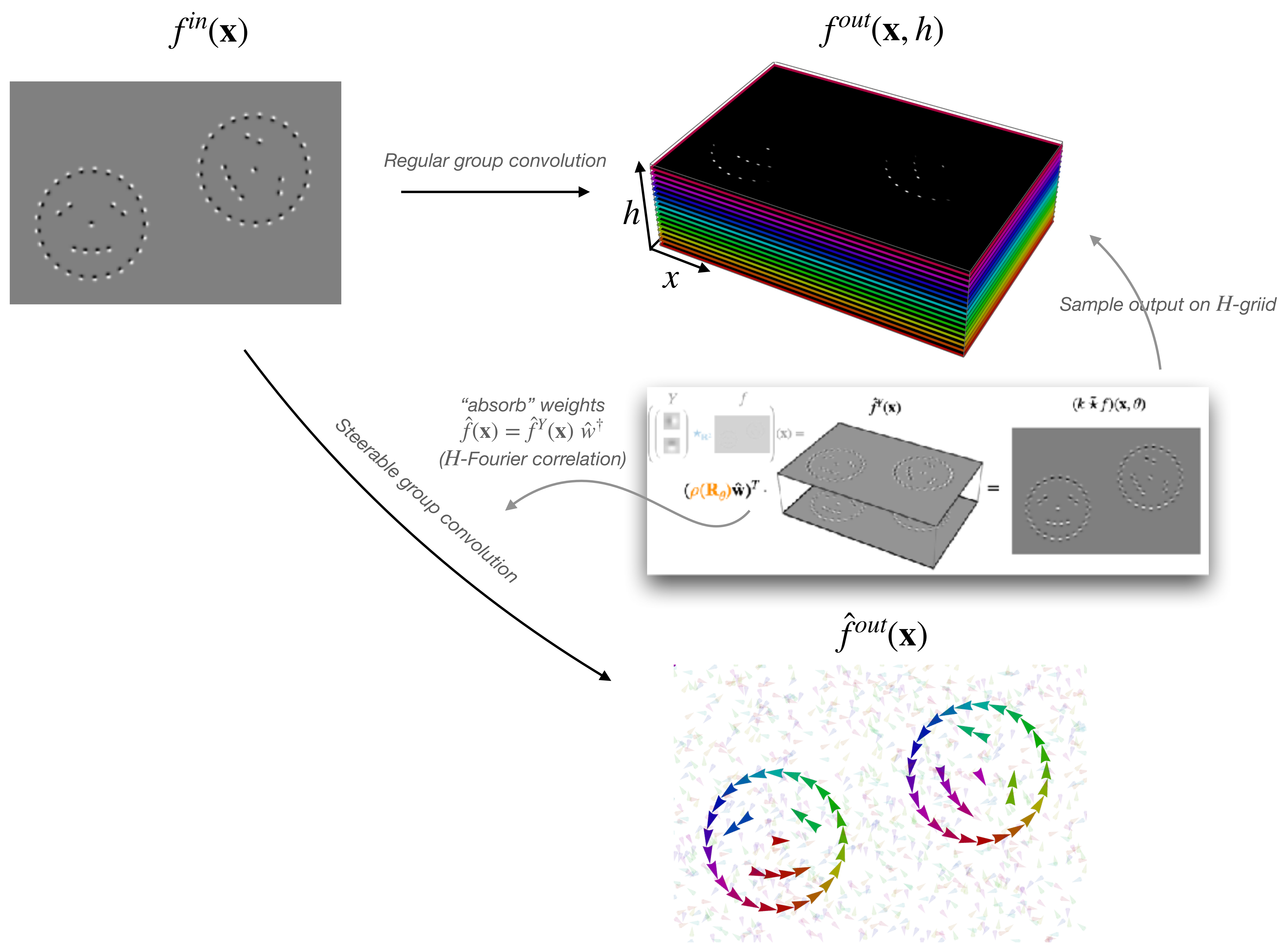
Lecture 2 - Steerable group convolutions

Lecture 2.5 - Steerable group convolutions

Erik Bekkers, Amsterdam Machine Learning Lab, University of Amsterdam

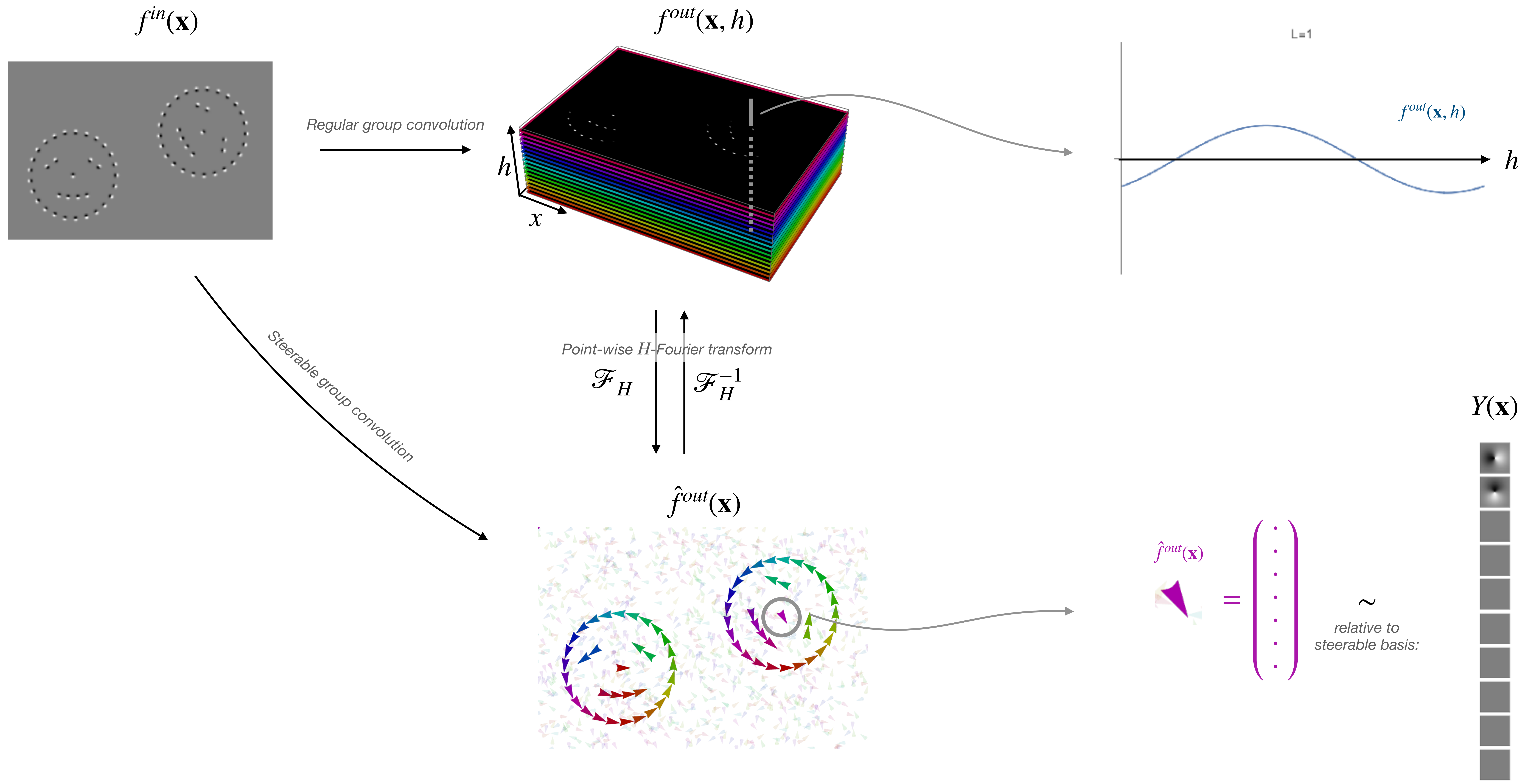
This mini-course serves as a module with the UvA Master AI course Deep Learning 2 https://uvadl2c.github.io/

From regular to steerable via a Fourier transform



$f^{in}(\mathbf{x})$
Regular group convolution
$f^{out}(\mathbf{x}, h)$
h
x
Sample output on H-griid
"absorb" weights
$\hat{f}(\mathbf{x}) = \hat{f}^{Y}(\mathbf{x})\ \hat{w}^{\dagger}$
(H-Fourier correlation)
$\hat{f}^{Y}(\mathbf{x})$
$(k \tilde{\star} f)(\mathbf{x}, \theta)$
$(\rho(\mathbf{R}_\theta)\hat{\mathbf{w}})^T \cdot$
Steerable group convolution
$\hat{f}^{out}(\mathbf{x})$

From regular to steerable via a Fourier transform


$f^{in}(\mathbf{x})$
Regular group convolution
$f^{out}(\mathbf{x}, h)$
h
x
L=1
$f^{out}(\mathbf{x}, h)$
h
Steerable group convolution
Point-wise H-Fourier transform
$\mathscr{F}_H$
$\mathscr{F}_H^{-1}$
$\hat{f}^{out}(\mathbf{x})$
$Y(\mathbf{x})$
$\hat{f}^{out}(\mathbf{x})$
=
~
relative to steerable basis:

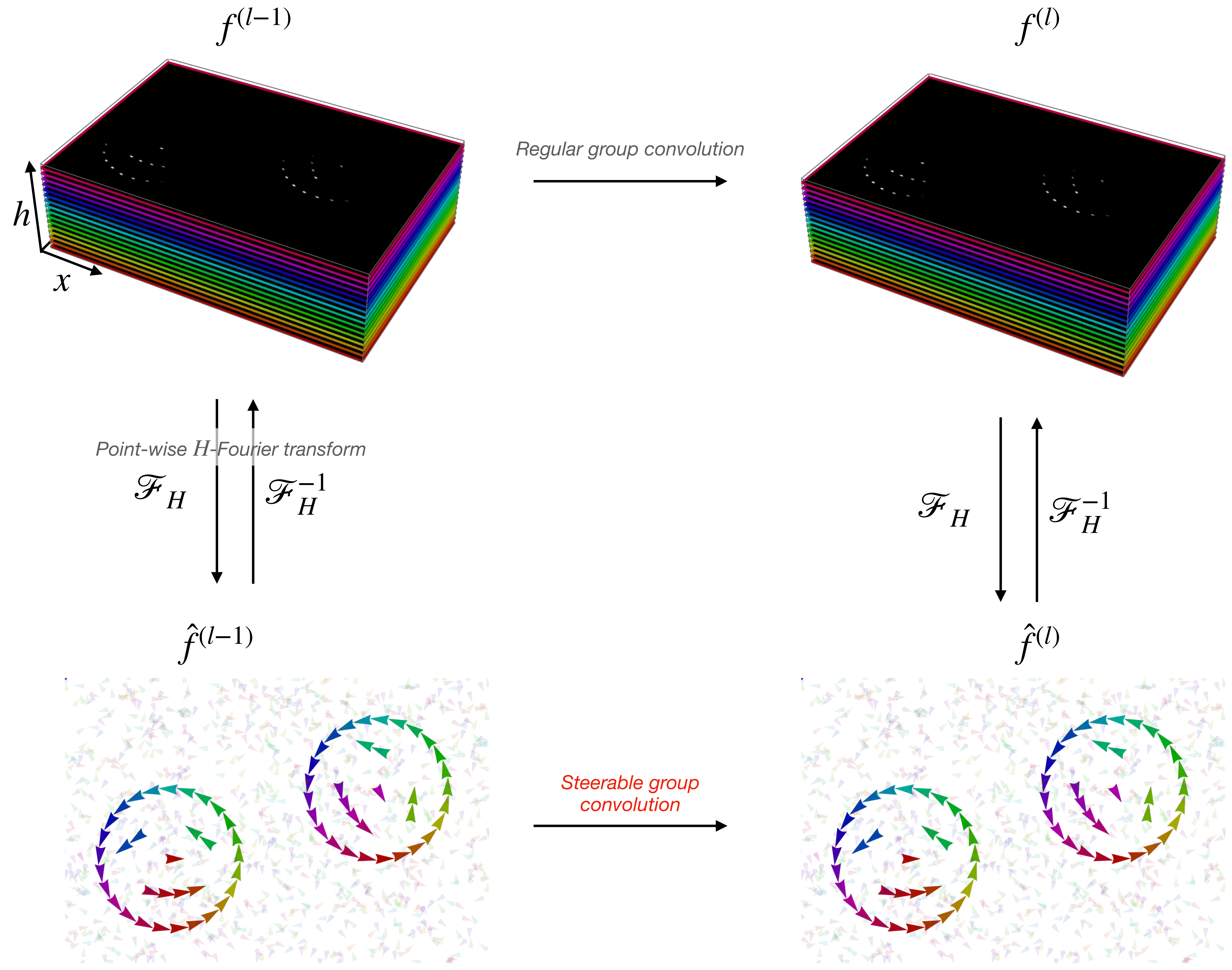
From regular to steerable via a Fourier transform

Regular group convolutions:
Domain expanded feature maps

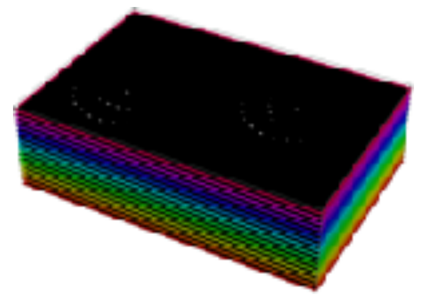
$$f^{(l)} : \mathbb{R}^d \times \underbrace{H}_{\text{added axis}} \to \mathbb{R}$$

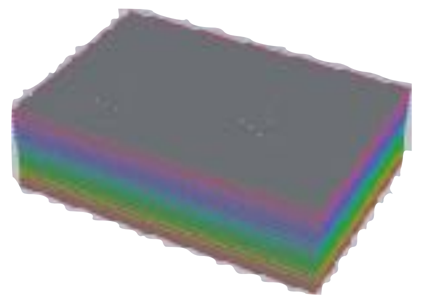
Steerable group convolutions:
Co-domain expanded feature maps (feature fields)

$$\hat{f}^{(l)} : \mathbb{R}^d \to \underbrace{V_H}_{\substack{\text{vector field instead of scalar field} \\ \text{(vectors in } V_H \text{ transform via group } H \text{ representations)}}}$$

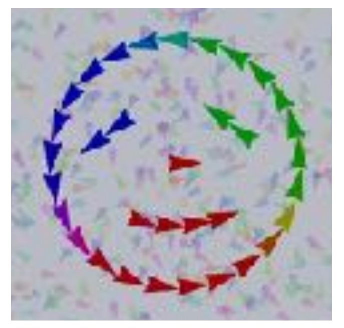


Steerable group convolutions

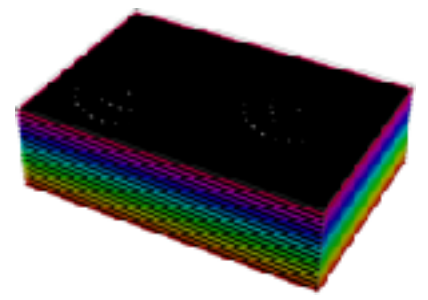
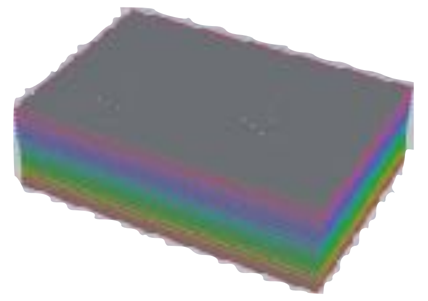
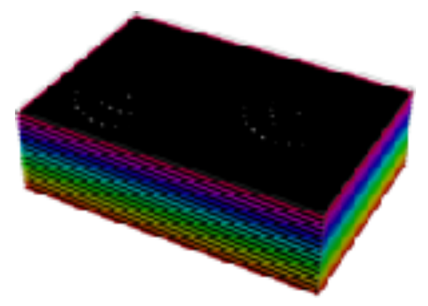
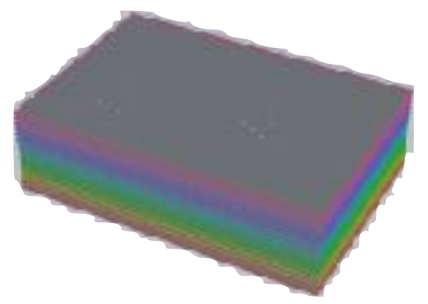




$$\begin{tikzcd} f^{in} \arrow[r, "\mathscr{K}"] \arrow[d, "\mathscr{F}_H"'] & f^{out} \arrow[d, "\mathscr{F}_H"] \\ \hat{f}^{in} \arrow[r, "\hat{\mathscr{K}}"'] & \hat{f}^{out} \end{tikzcd}$$



Steerable group convolutions

$$\begin{tikzcd}
\mathscr{L}_g f^{in} \arrow[r, "\mathscr{K}"] & \mathscr{L}_g f^{out} \arrow[d, "\mathscr{L}_{g^{-1}}"] \\
f^{in} \arrow[u, "\mathscr{L}_g"] \arrow[r, "\mathscr{K}"] \arrow[d, "\mathscr{F}_H"] & f^{out} \arrow[d, "\mathscr{F}_H"] \\
\hat{f}^{in} \arrow[r, "\hat{\mathscr{K}}"'] & \hat{f}^{out}
\end{tikzcd}$$

If $\mathscr{K}$ is linear

$$\mathscr{K}[f](g) = \int_G k(g, g')f(g)\mathrm{d}g$$

and equivariant

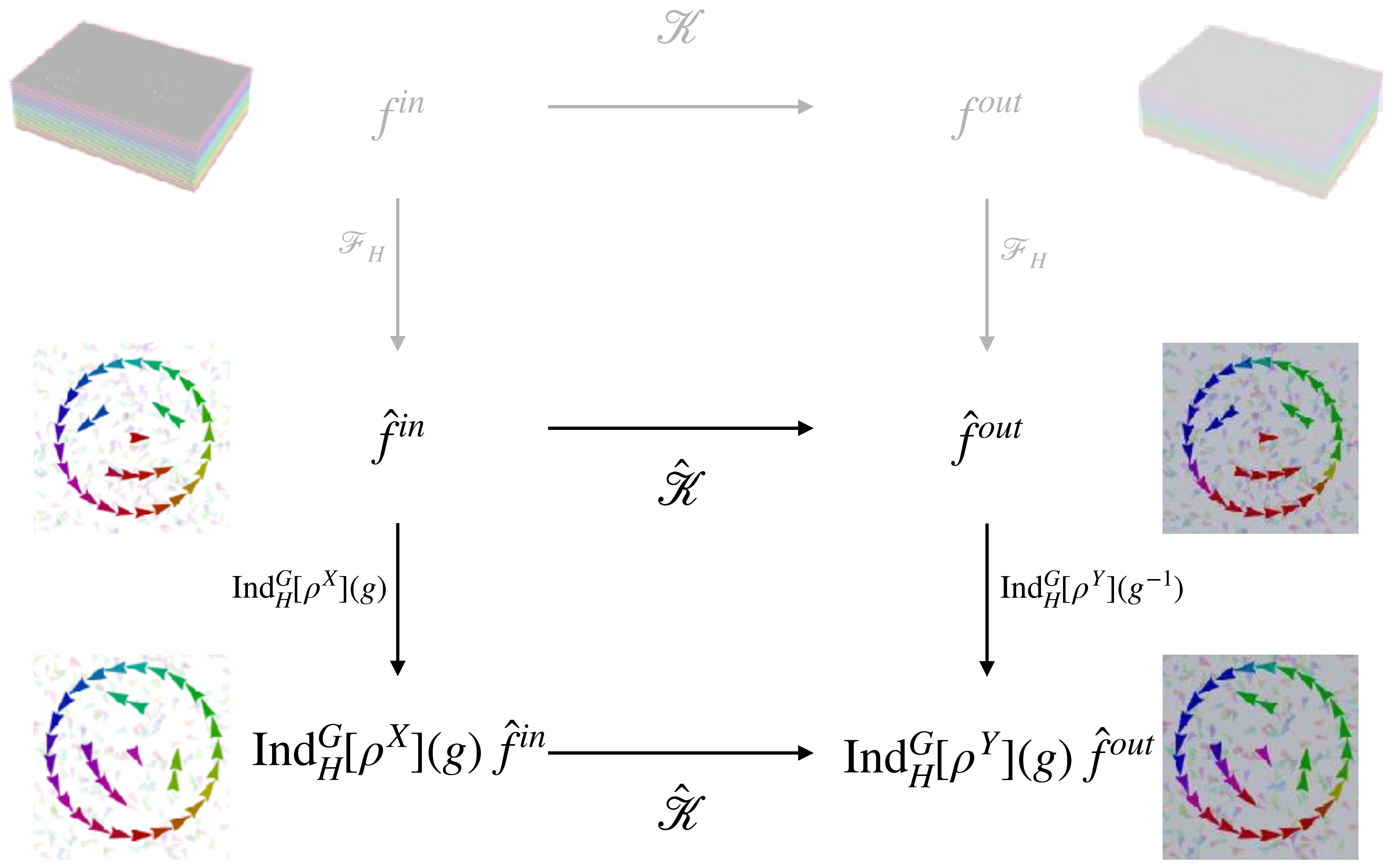
$$\mathscr{K}[\mathscr{L}_g f] = \mathscr{L}_g \mathscr{K}[f]$$

Then it is a group convolution

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

Recall lecture 1.7 (Group convolutions are all you need)

Steerable group convolutions



$$\begin{tikzcd}
f^{in} \arrow[r, "\mathscr{K}"] \arrow[d, "\mathscr{F}_H"'] & f^{out} \arrow[d, "\mathscr{F}_H"] \\
\hat{f}^{in} \arrow[r, "\hat{\mathscr{K}}"'] \arrow[d, "\mathrm{Ind}_H^G[\rho^X](g)"'] & \hat{f}^{out} \arrow[d, "\mathrm{Ind}_H^G[\rho^Y](g^{-1})"] \\
\mathrm{Ind}_H^G[\rho^X](g)\,\hat{f}^{in} \arrow[r, "\hat{\mathscr{K}}"'] & \mathrm{Ind}_H^G[\rho^Y](g)\,\hat{f}^{out}
\end{tikzcd}$$

If $\mathscr{K}$ is linear

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

and equivariant

$$\mathscr{K}[\mathrm{Ind}_H^G[\rho^X](g)\,\hat{f}] = \mathrm{Ind}_H^G[\rho^Y](g)\,\mathscr{K}[\hat{f}]$$

$$\Downarrow$$

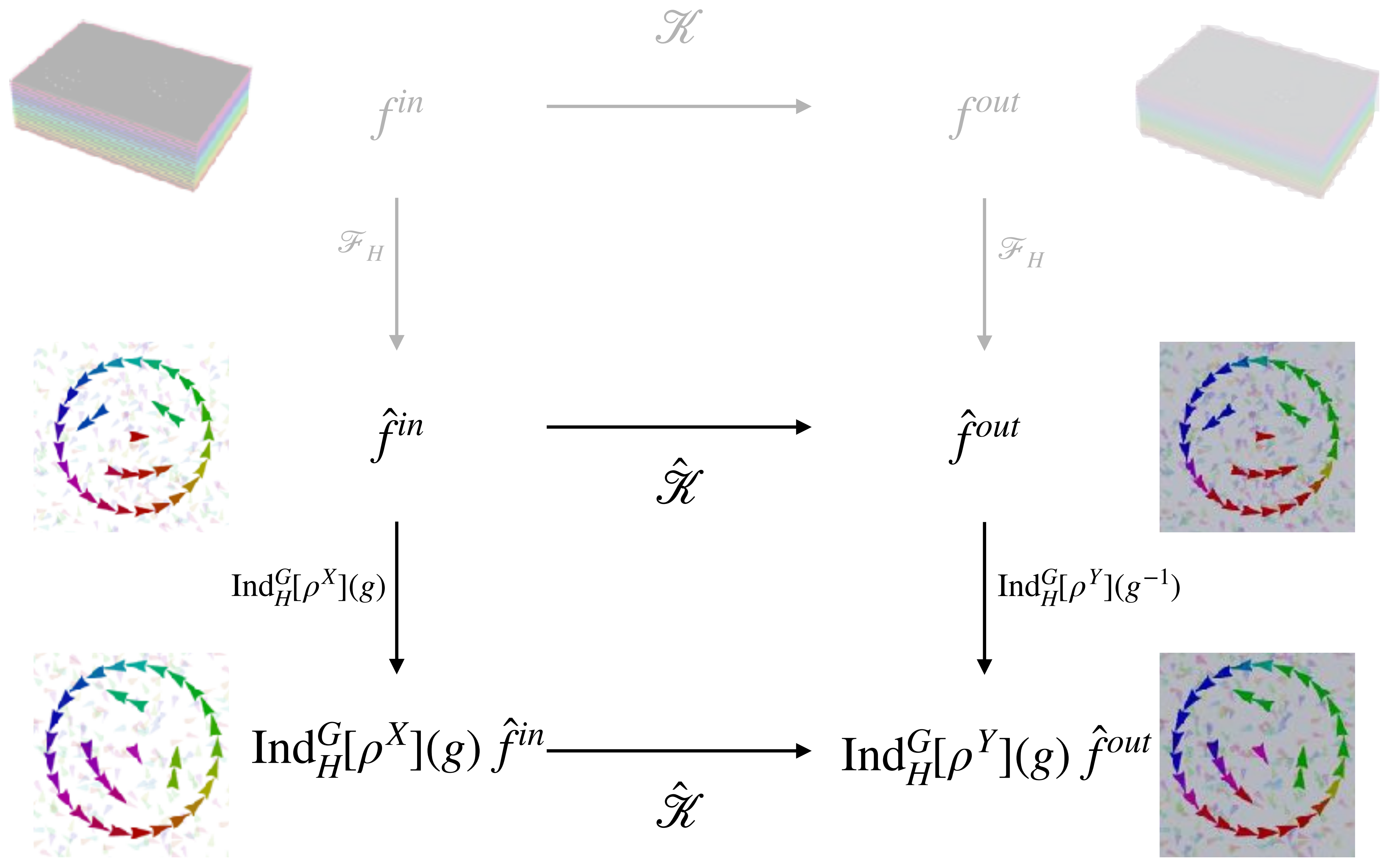
Then it is a normal convolution

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

but with kernel $k : \mathbb{R}^d \rightarrow \mathbb{R}^{d_Y \times d_X}$ satisfying **constraint**

$$\forall_{h \in H}\ \forall_{\mathbf{x} \in \mathbb{R}^d} : \qquad k(h\,\mathbf{x}) = \rho_Y(h)k(\mathbf{x})\rho_X(h^{-1})$$

Steerable group convolutions



$$\begin{tikzcd}
f^{in} \arrow[r, "\mathscr{K}"] \arrow[d, "\mathscr{F}_H"'] & f^{out} \arrow[d, "\mathscr{F}_H"] \\
\hat{f}^{in} \arrow[r, "\hat{\mathscr{K}}"'] \arrow[d, "\mathrm{Ind}_H^G[\rho^X](g)"'] & \hat{f}^{out} \arrow[d, "\mathrm{Ind}_H^G[\rho^Y](g^{-1})"] \\
\mathrm{Ind}_H^G[\rho^X](g)\, \hat{f}^{in} \arrow[r, "\hat{\mathscr{K}}"'] & \mathrm{Ind}_H^G[\rho^Y](g)\, \hat{f}^{out}
\end{tikzcd}$$

If $\mathscr{K}$ is linear

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

and equivariant

$$\mathscr{K}[\mathrm{Ind}_H^G[\rho^X](g)\, \hat{f}] = \mathrm{Ind}_H^G[\rho^Y](g)\, \mathscr{K}[\hat{f}]$$

$$\Downarrow$$

Then it is a normal convolution

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

but with kernel $k : \mathbb{R}^d \rightarrow \mathbb{R}^{d_Y \times d_X}$ satisfying **constraint**

$$\forall_{h \in H}\ \forall_{\mathbf{x} \in \mathbb{R}^d} : \qquad k(h\,\mathbf{x}) = \rho_Y(h)k(\mathbf{x})\rho_X(h^{-1})$$

Problem: The G-steerability constraint! [1,2]

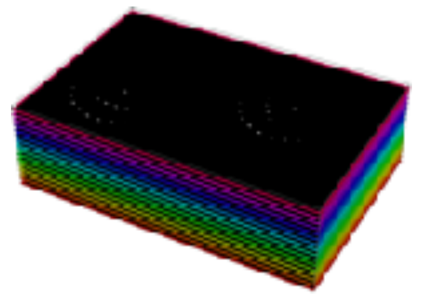
Solution: Expand kernel in steerable basis

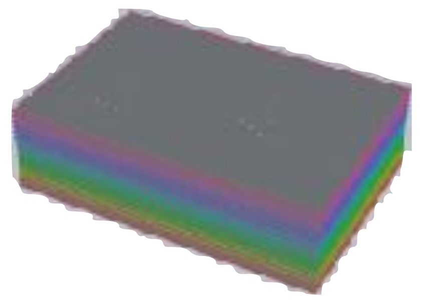
[1] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S. Cohen. 3D steerable CNNs: Learning rotationally equivariant features in volumetric data. In Conference on Neural Information Processing Systems (NeurIPS), 2018a.

[2] Cesa, G., Lang, L., & Weiler, M. (2021, September). A Program to Build E (N)-Equivariant Steerable CNNs. In International Conference on Learning Representations.

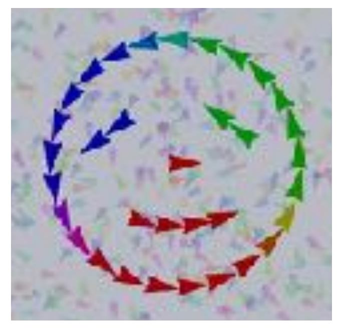
Steerable group convolutions

Group convolution $\mathscr{K}[f](g) = \int_G k(g^{-1}g')f(g)\mathrm{d}\mathbf{x}'\mathrm{d}g$





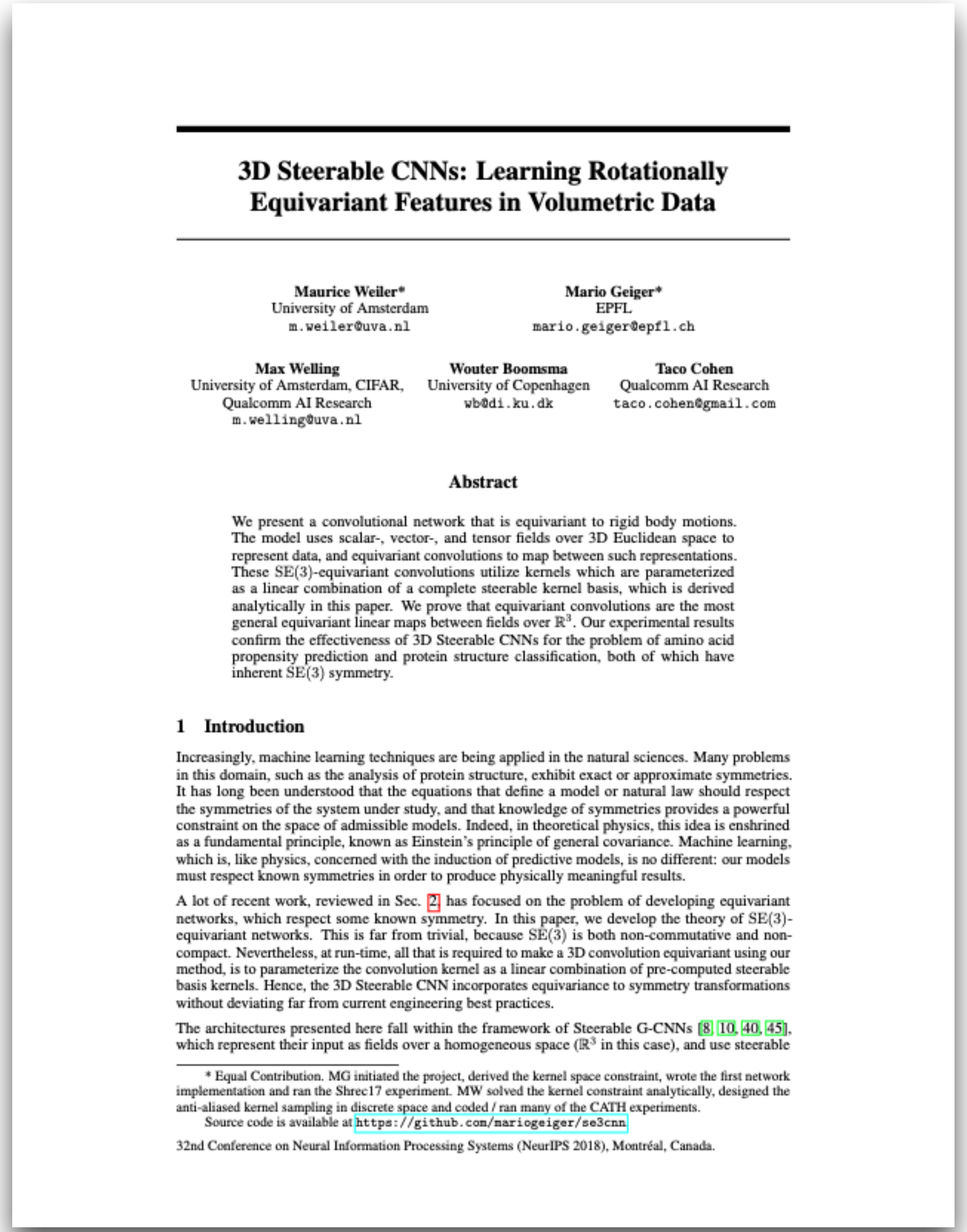
$$\begin{tikzcd}
f^{in} \arrow[r, "\mathscr{K}"] \arrow[d, "\mathscr{F}_H"'] & f^{out} \arrow[d, "\mathscr{F}_H"] \\
\hat{f}^{in} \arrow[r, "\hat{\mathscr{K}}"'] & \hat{f}^{out}
\end{tikzcd}$$



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

but with kernel $k : \mathbb{R}^d \to \mathbb{R}^{d_Y \times d_X}$ satisfying **constraint**

$\forall_{h \in H}\ \forall_{\mathbf{x} \in \mathbb{R}^d} :\qquad k(g\,\mathbf{x}) = \rho_Y(h)k(\mathbf{x})\rho_X(h^{-1})$



3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data

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

Mario Geiger*
EPFL
mario.geiger@epfl.ch

Max Welling
University of Amsterdam, CIFAR,
Qualcomm AI Research
m.welling@uva.nl

Wouter Boomsma
University of Copenhagen
wb@di.ku.dk

Taco Cohen
Qualcomm AI Research
taco.cohen@gmail.com

Abstract

We present a convolutional network that is equivariant to rigid body motions. The model uses scalar-, vector-, and tensor fields over 3D Euclidean space to represent data, and equivariant convolutions to map between such representations. These SE(3)-equivariant convolutions utilize kernels which are parameterized as a linear combination of a complete steerable kernel basis, which is derived analytically in this paper. We prove that equivariant convolutions are the most general equivariant linear maps between fields over $\mathbb{R}^3$. Our experimental results confirm the effectiveness of 3D Steerable CNNs for the problem of amino acid propensity prediction and protein structure classification, both of which have inherent SE(3) symmetry.

1 Introduction

Increasingly, machine learning techniques are being applied in the natural sciences. Many problems in this domain, such as the analysis of protein structure, exhibit exact or approximate symmetries. It has long been understood that the equations that define a model or natural law should respect the symmetries of the system under study, and that knowledge of symmetries provides a powerful constraint on the space of admissible models. Indeed, in theoretical physics, this idea is enshrined as a fundamental principle, known as Einstein's principle of general covariance. Machine learning, which is, like physics, concerned with the induction of predictive models, is no different: our models must respect known symmetries in order to produce physically meaningful results.

A lot of recent work, reviewed in Sec. 2, has focused on the problem of developing equivariant networks, which respect some known symmetry. In this paper, we develop the theory of SE(3)-equivariant networks. This is far from trivial, because SE(3) is both non-commutative and non-compact. Nevertheless, at run-time, all that is required to make a 3D convolution equivariant using our method, is to parameterize the convolution kernel as a linear combination of pre-computed steerable basis kernels. Hence, the 3D Steerable CNN incorporates equivariance to symmetry transformations without deviating far from current engineering best practices.

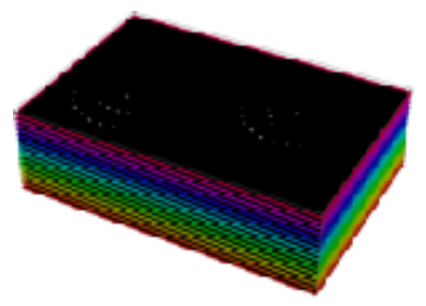
The architectures presented here fall within the framework of Steerable G-CNNs [8, 10, 40, 45], which represent their input as fields over a homogeneous space ($\mathbb{R}^3$ in this case), and use steerable

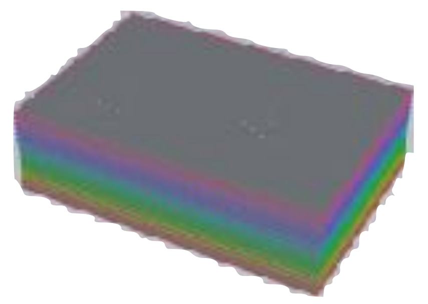
* Equal Contribution. MG initiated the project, derived the kernel space constraint, wrote the first network implementation and ran the Shrec17 experiment. MW solved the kernel constraint analytically, designed the anti-aliased kernel sampling in discrete space and coded / ran many of the CATH experiments.
Source code is available at https://github.com/mariogeiger/se3cnn

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

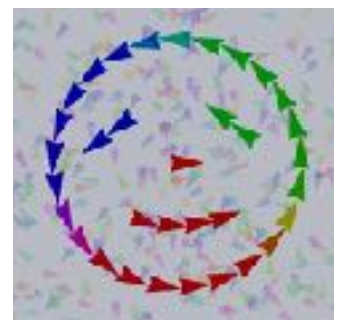
Steerable group convolutions

Group convolution $\mathcal{K}[f](g) = \int_G k(g^{-1}g')f(g)\mathrm{d}\mathbf{x}'\mathrm{d}g$





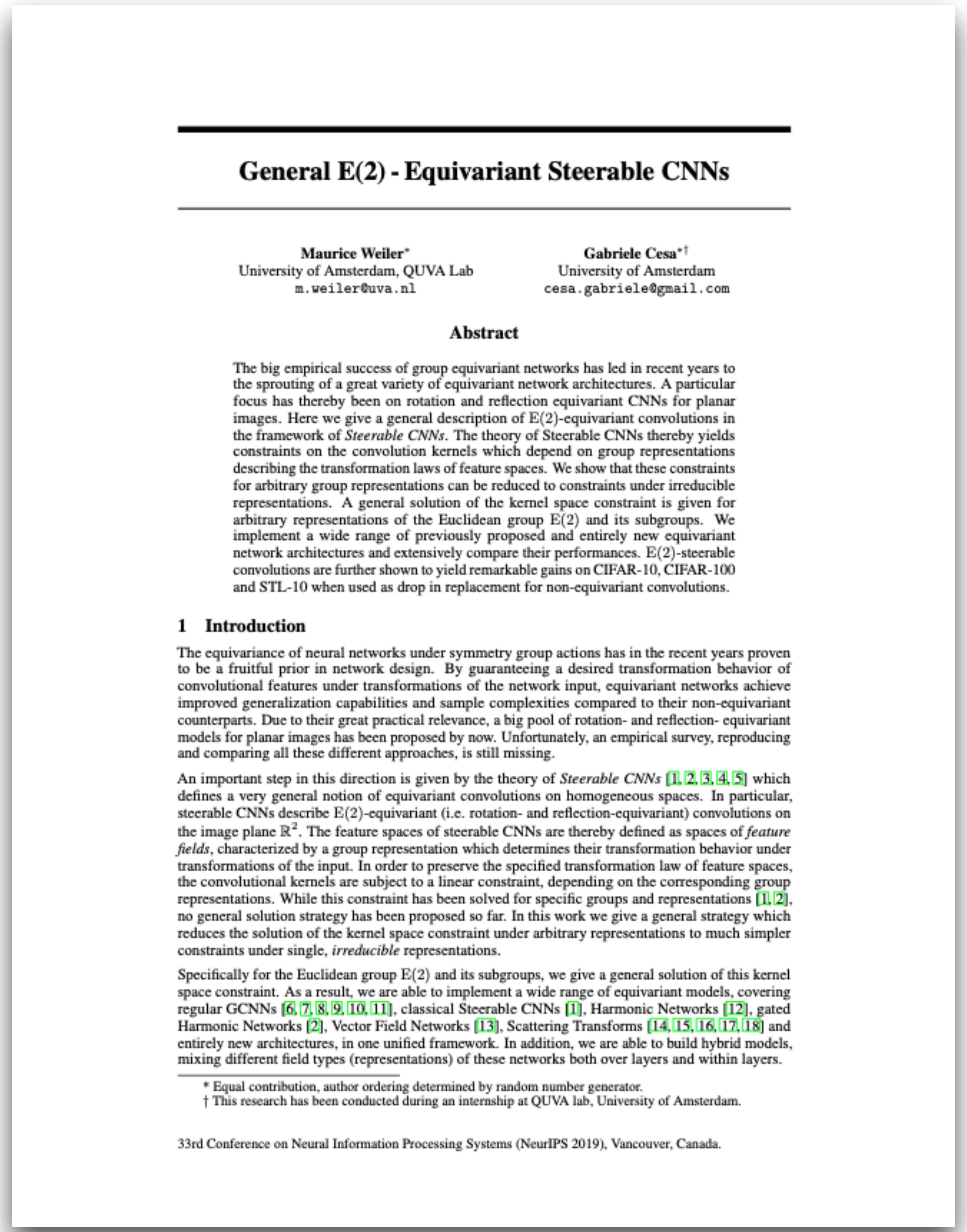
$$\begin{tikzcd}
f^{in} \arrow[r, "\mathcal{K}"] \arrow[d, "\mathcal{F}_H"'] & f^{out} \arrow[d, "\mathcal{F}_H"] \\
\hat{f}^{in} \arrow[r, "\hat{\mathcal{K}}"'] & \hat{f}^{out}
\end{tikzcd}$$



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

but with kernel $k : \mathbb{R}^d \to \mathbb{R}^{d_Y \times d_X}$ satisfying **constraint**

$\forall_{h \in H} \forall_{\mathbf{x} \in \mathbb{R}^d}$: $k(g\,\mathbf{x}) = \rho_Y(h)k(\mathbf{x})\rho_X(h^{-1})$



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

Abstract

The big empirical success of group equivariant networks has led in recent years to the sprouting of a great variety of equivariant network architectures. A particular focus has thereby been on rotation and reflection equivariant CNNs for planar images. Here we give a general description of E(2)-equivariant convolutions in the framework of *Steerable CNNs*. The theory of Steerable CNNs thereby yields constraints on the convolution kernels which depend on group representations describing the transformation laws of feature spaces. We show that these constraints for arbitrary group representations can be reduced to constraints under irreducible representations. A general solution of the kernel space constraint is given for arbitrary representations of the Euclidean group E(2) and its subgroups. We implement a wide range of previously proposed and entirely new equivariant network architectures and extensively compare their performances. E(2)-steerable convolutions are further shown to yield remarkable gains on CIFAR-10, CIFAR-100 and STL-10 when used as drop in replacement for non-equivariant convolutions.

1 Introduction

The equivariance of neural networks under symmetry group actions has in the recent years proven to be a fruitful prior in network design. By guaranteeing a desired transformation behavior of convolutional features under transformations of the network input, equivariant networks achieve improved generalization capabilities and sample complexities compared to their non-equivariant counterparts. Due to their great practical relevance, a big pool of rotation- and reflection- equivariant models for planar images has been proposed by now. Unfortunately, an empirical survey, reproducing and comparing all these different approaches, is still missing.

An important step in this direction is given by the theory of *Steerable CNNs* [1, 2, 3, 4, 5] which defines a very general notion of equivariant convolutions on homogeneous spaces. In particular, steerable CNNs describe E(2)-equivariant (i.e. rotation- and reflection-equivariant) convolutions on the image plane $\mathbb{R}^2$. The feature spaces of steerable CNNs are thereby defined as spaces of *feature fields*, characterized by a group representation which determines their transformation behavior under transformations of the input. In order to preserve the specified transformation law of feature spaces, the convolutional kernels are subject to a linear constraint, depending on the corresponding group representations. While this constraint has been solved for specific groups and representations [1, 2], no general solution strategy has been proposed so far. In this work we give a general strategy which reduces the solution of the kernel space constraint under arbitrary representations to much simpler constraints under single, *irreducible* representations.

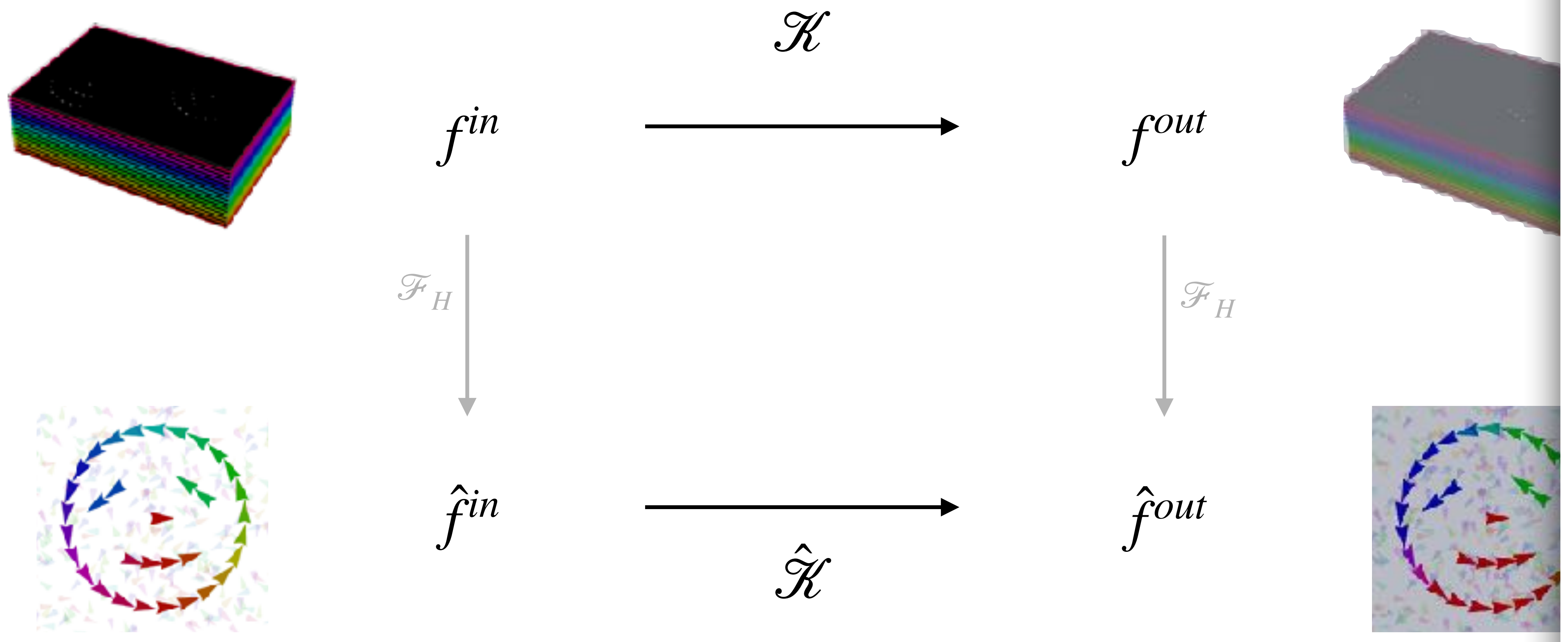
Specifically for the Euclidean group E(2) and its subgroups, we give a general solution of this kernel space constraint. As a result, we are able to implement a wide range of equivariant models, covering regular GCNNs [6, 7, 8, 9, 10, 11], classical Steerable CNNs [1], Harmonic Networks [12], gated Harmonic Networks [2], Vector Field Networks [13], Scattering Transforms [14, 15, 16, 17, 18] and entirely new architectures, in one unified framework. In addition, we are able to build hybrid models, mixing different field types (representations) of these networks both over layers and within layers.

[*] Equal contribution, author ordering determined by random number generator.
[†] This research has been conducted during an internship at QUVA lab, University of Amsterdam.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Steerable group convolutions

Group convolution $\mathscr{K}[f](g) = \int_G k(g^{-1}g')f(g)\mathrm{d}\mathbf{x}'\mathrm{d}g$



$$\begin{tikzcd}
f^{in} \arrow[r, "\mathscr{K}"] \arrow[d, "\mathscr{F}_H"'] & f^{out} \arrow[d, "\mathscr{F}_H"] \\
\hat{f}^{in} \arrow[r, "\hat{\mathscr{K}}"'] & \hat{f}^{out}
\end{tikzcd}$$

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

but with kernel $k : \mathbb{R}^d \to \mathbb{R}^{d_Y \times d_X}$ satisfying **constraint**

$\forall_{h \in H}\ \forall_{\mathbf{x} \in \mathbb{R}^d} :$ $\quad k(g\,\mathbf{x}) = \rho_Y(h)k(\mathbf{x})\rho_X(h^{-1})$

Published as a conference paper at ICLR 2022

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

Abstract

Equivariance is becoming an increasingly popular design choice to build data efficient neural networks by exploiting prior knowledge about the symmetries of the problem at hand. Euclidean steerable CNNs are one of the most common classes of equivariant networks. While the constraints these architectures need to satisfy are understood, existing approaches are tailored to specific (classes of) groups. No generally applicable method that is *practical* for implementation has been described so far. In this work, we generalize the Wigner-Eckart theorem proposed in Lang & Weiler (2020), which characterizes general G-steerable kernel spaces for compact groups G over their homogeneous spaces, to arbitrary G-spaces. This enables us to directly parameterize filters in terms of a band-limited basis on the whole space rather than on G's orbits, but also to easily implement steerable CNNs equivariant to a large number of groups. To demonstrate its generality, we instantiate our method on a variety of isometry groups acting on the Euclidean space $\mathbb{R}^3$. Our framework allows us to build E(3) and SE(3)-steerable CNNs like previous works, but also CNNs with arbitrary $G \leq \mathrm{O}(3)$-steerable kernels. For example, we build 3D CNNs equivariant to the symmetries of platonic solids or choose $G = \mathrm{SO}(2)$ when working with 3D data having only azimuthal symmetries. We compare these models on 3D shapes and molecular datasets, observing improved performance by matching the model's symmetries to the ones of the data.

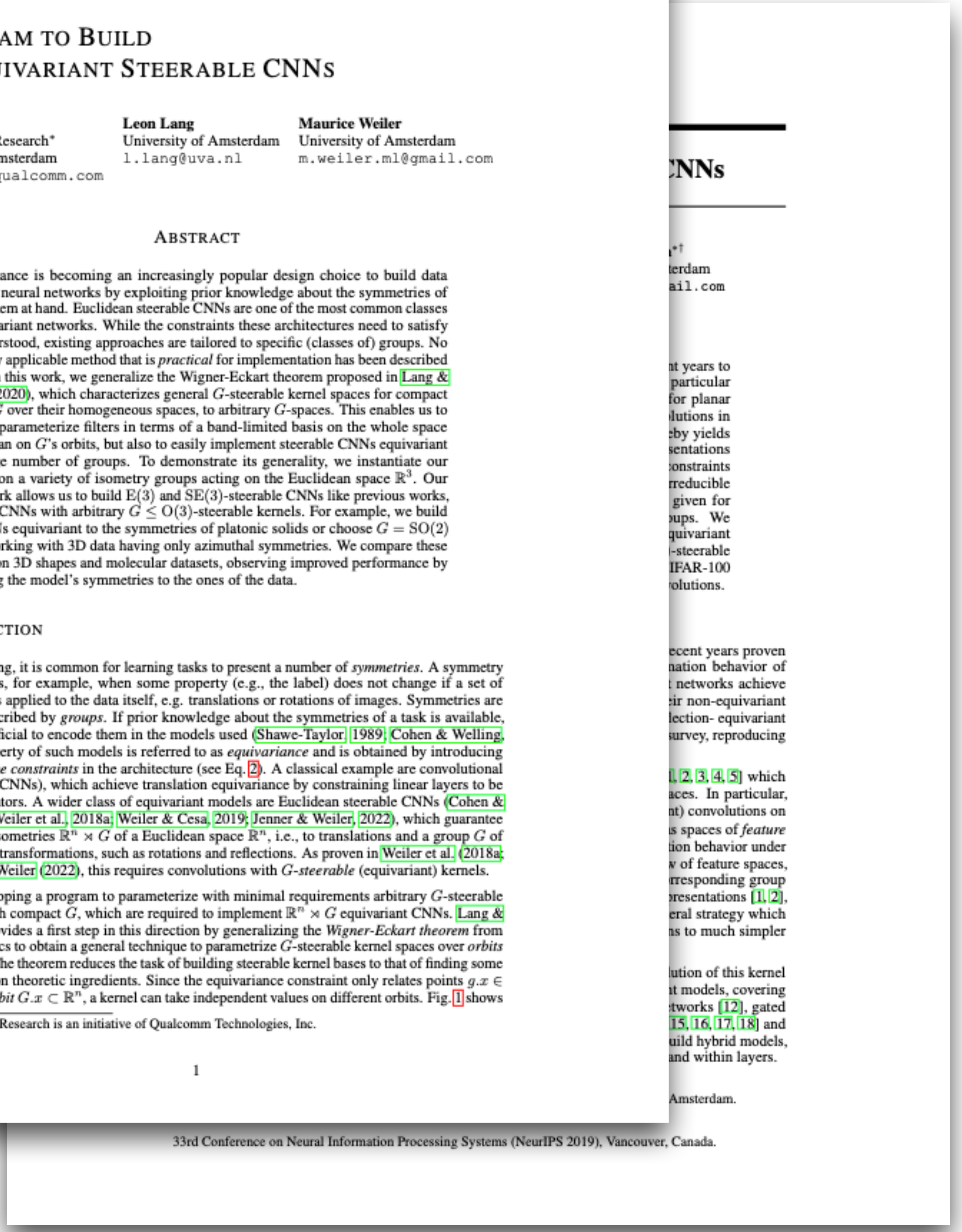
1 Introduction

In machine learning, it is common for learning tasks to present a number of *symmetries*. A symmetry in the data occurs, for example, when some property (e.g., the label) does not change if a set of transformations is applied to the data itself, e.g. translations or rotations of images. Symmetries are algebraically described by *groups*. If prior knowledge about the symmetries of a task is available, it is usually beneficial to encode them in the models used (Shawe-Taylor, 1989; Cohen & Welling, 2016a). The property of such models is referred to as *equivariance* and is obtained by introducing some *equivariance constraints* in the architecture (see Eq. 2). A classical example are convolutional neural networks (CNNs), which achieve translation equivariance by constraining linear layers to be convolution operators. A wider class of equivariant models are Euclidean steerable CNNs (Cohen & Welling, 2016b; Weiler et al., 2018a; Weiler & Cesa, 2019; Jenner & Weiler, 2022), which guarantee equivariance to isometries $\mathbb{R}^n \rtimes G$ of a Euclidean space $\mathbb{R}^n$, i.e., to translations and a group G of origin-preserving transformations, such as rotations and reflections. As proven in Weiler et al. (2018a; 2021); Jenner & Weiler (2022), this requires convolutions with G-*steerable* (equivariant) kernels.

Our goal is developing a program to parameterize with minimal requirements arbitrary G-steerable kernel spaces, with compact G, which are required to implement $\mathbb{R}^n \rtimes G$ equivariant CNNs. Lang & Weiler (2020) provides a first step in this direction by generalizing the *Wigner-Eckart theorem* from quantum mechanics to obtain a general technique to parametrize G-steerable kernel spaces over *orbits* of a compact G. The theorem reduces the task of building steerable kernel bases to that of finding some pure representation theoretic ingredients. Since the equivariance constraint only relates points $g.x \in \mathbb{R}^n$ in the same *orbit* $G.x \subset \mathbb{R}^n$, a kernel can take independent values on different orbits. Fig. 1 shows

*Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

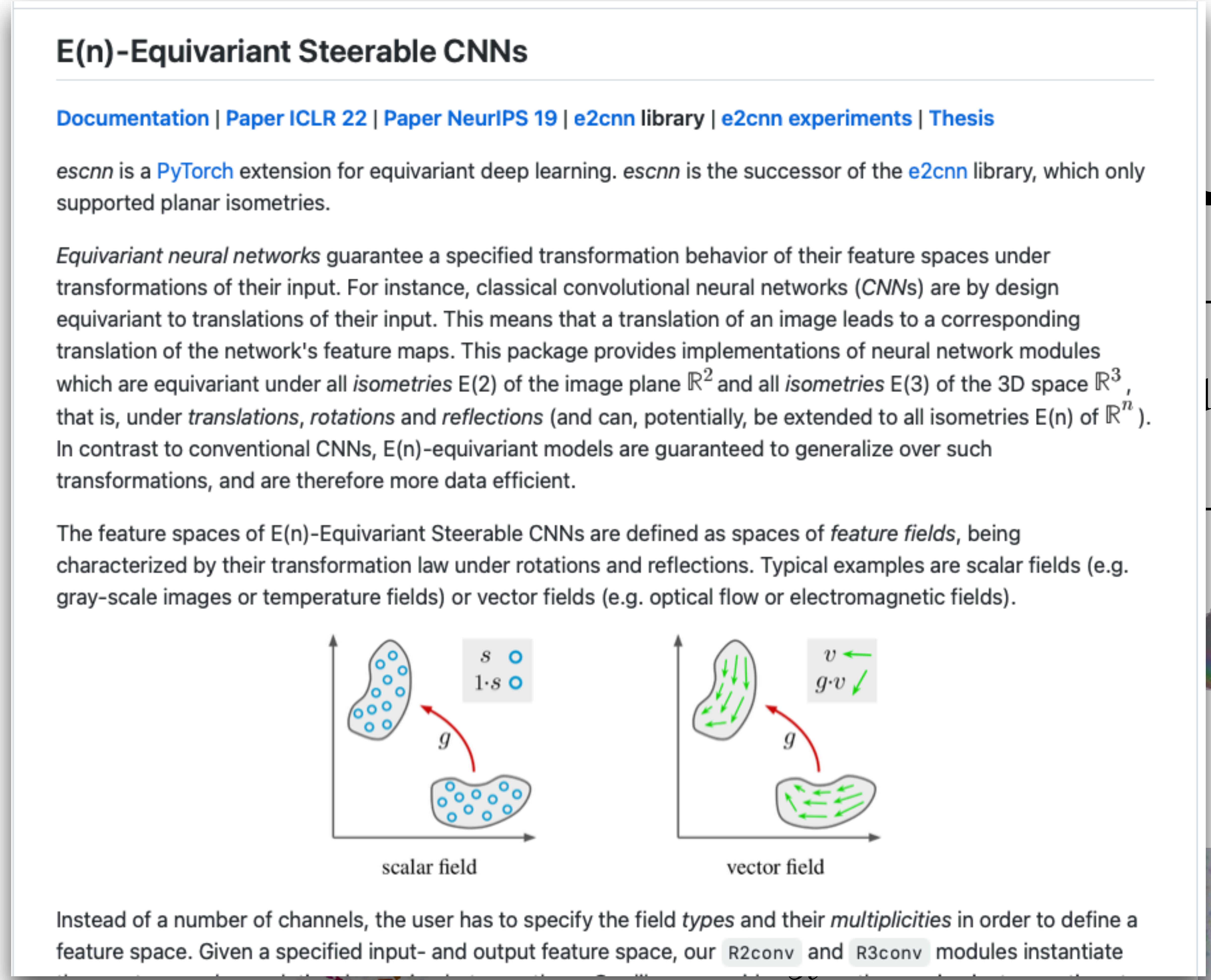
1



33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

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

up convolutions



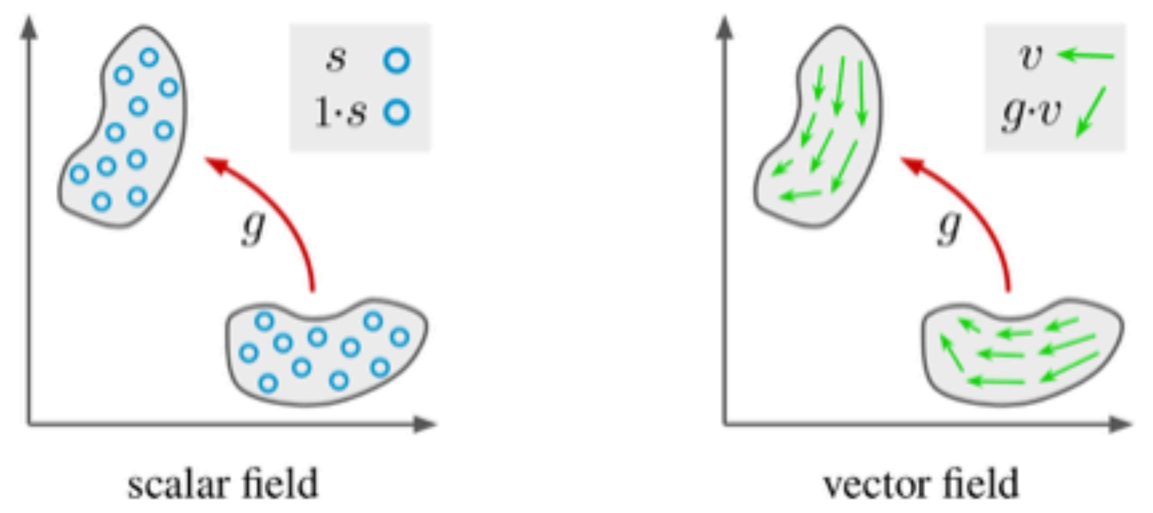
E(n)-Equivariant Steerable CNNs

Documentation | Paper ICLR 22 | Paper NeurIPS 19 | e2cnn library | e2cnn experiments | Thesis

escnn is a PyTorch extension for equivariant deep learning. *escnn* is the successor of the e2cnn library, which only supported planar isometries.

Equivariant neural networks guarantee a specified transformation behavior of their feature spaces under transformations of their input. For instance, classical convolutional neural networks (*CNN*s) are by design equivariant to translations of their input. This means that a translation of an image leads to a corresponding translation of the network's feature maps. This package provides implementations of neural network modules which are equivariant under all *isometries* E(2) of the image plane $\mathbb{R}^2$ and all *isometries* E(3) of the 3D space $\mathbb{R}^3$, that is, under *translations*, *rotations* and *reflections* (and can, potentially, be extended to all isometries E(n) of $\mathbb{R}^n$). In contrast to conventional CNNs, E(n)-equivariant models are guaranteed to generalize over such transformations, and are therefore more data efficient.

The feature spaces of E(n)-Equivariant Steerable CNNs are defined as spaces of *feature fields*, being characterized by their transformation law under rotations and reflections. Typical examples are scalar fields (e.g. gray-scale images or temperature fields) or vector fields (e.g. optical flow or electromagnetic fields).



Instead of a number of channels, the user has to specify the field *types* and their *multiplicities* in order to define a feature space. Given a specified input- and output feature space, our `R2conv` and `R3conv` modules instantiate

Published as a conference paper at ICLR 2022

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

Abstract

Equivariance is becoming an increasingly popular design choice to build data efficient neural networks by exploiting prior knowledge about the symmetries of the problem at hand. Euclidean steerable CNNs are one of the most common classes of equivariant networks. While the constraints these architectures need to satisfy are understood, existing approaches are tailored to specific (classes of) groups. No generally applicable method that is *practical* for implementation has been described so far. In this work, we generalize the Wigner-Eckart theorem proposed in Lang & Weiler (2020), which characterizes general G-steerable kernel spaces for compact groups G over their homogeneous spaces, to arbitrary G-spaces. This enables us to directly parameterize filters in terms of a band-limited basis on the whole space rather than on G's orbits, but also to easily implement steerable CNNs equivariant to a large number of groups. To demonstrate its generality, we instantiate our method on a variety of isometry groups acting on the Euclidean space $\mathbb{R}^3$. Our framework allows us to build E(3) and SE(3)-steerable CNNs like previous works, but also CNNs with arbitrary $G \leq \mathrm{O}(3)$-steerable kernels. For example, we build 3D CNNs equivariant to the symmetries of platonic solids or choose $G = \mathrm{SO}(2)$ when working with 3D data having only azimuthal symmetries. We compare these models on 3D shapes and molecular datasets, observing improved performance by matching the model's symmetries to the ones of the data.

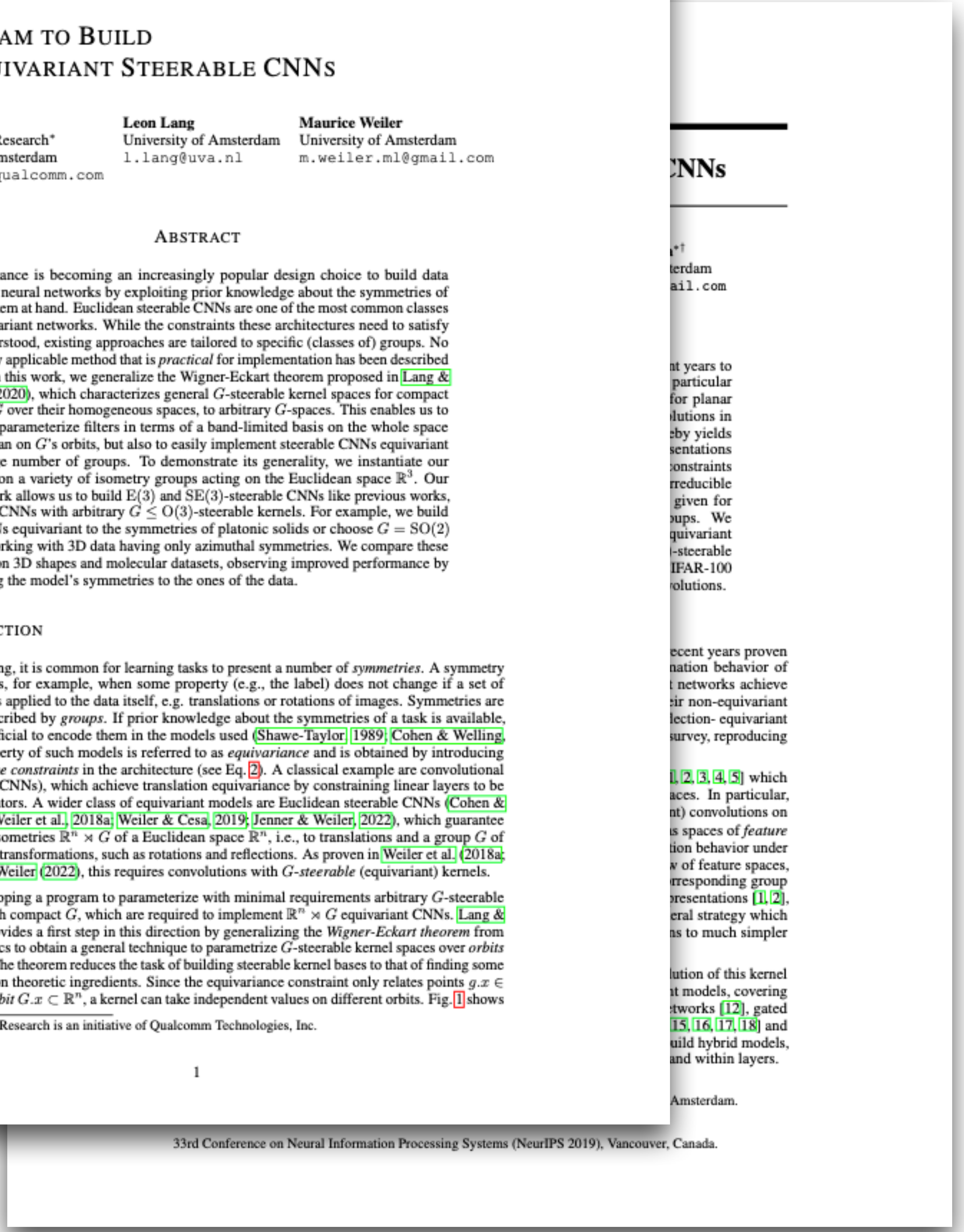
1 Introduction

In machine learning, it is common for learning tasks to present a number of *symmetries*. A symmetry in the data occurs, for example, when some property (e.g., the label) does not change if a set of transformations is applied to the data itself, e.g. translations or rotations of images. Symmetries are algebraically described by *groups*. If prior knowledge about the symmetries of a task is available, it is usually beneficial to encode them in the models used (Shawe-Taylor, 1989; Cohen & Welling, 2016a). The property of such models is referred to as *equivariance* and is obtained by introducing some *equivariance constraints* in the architecture (see Eq. 2). A classical example are convolutional neural networks (CNNs), which achieve translation equivariance by constraining linear layers to be convolution operators. A wider class of equivariant models are Euclidean steerable CNNs (Cohen & Welling, 2016b; Weiler et al., 2018a; Weiler & Cesa, 2019; Jenner & Weiler, 2022), which guarantee equivariance to isometries $\mathbb{R}^n \rtimes G$ of a Euclidean space $\mathbb{R}^n$, i.e., to translations and a group G of origin-preserving transformations, such as rotations and reflections. As proven in Weiler et al. (2018a; 2021); Jenner & Weiler (2022), this requires convolutions with G-*steerable* (equivariant) kernels.

Our goal is developing a program to parameterize with minimal requirements arbitrary G-steerable kernel spaces, with compact G, which are required to implement $\mathbb{R}^n \rtimes G$ equivariant CNNs. Lang & Weiler (2020) provides a first step in this direction by generalizing the *Wigner-Eckart theorem* from quantum mechanics to obtain a general technique to parametrize G-steerable kernel spaces over *orbits* of a compact G. The theorem reduces the task of building steerable kernel bases to that of finding some pure representation theoretic ingredients. Since the equivariance constraint only relates points $g.x \in \mathbb{R}^n$ in the same *orbit* $G.x \subset \mathbb{R}^n$, a kernel can take independent values on different orbits. Fig. 1 shows

*Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

1



CNNs

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

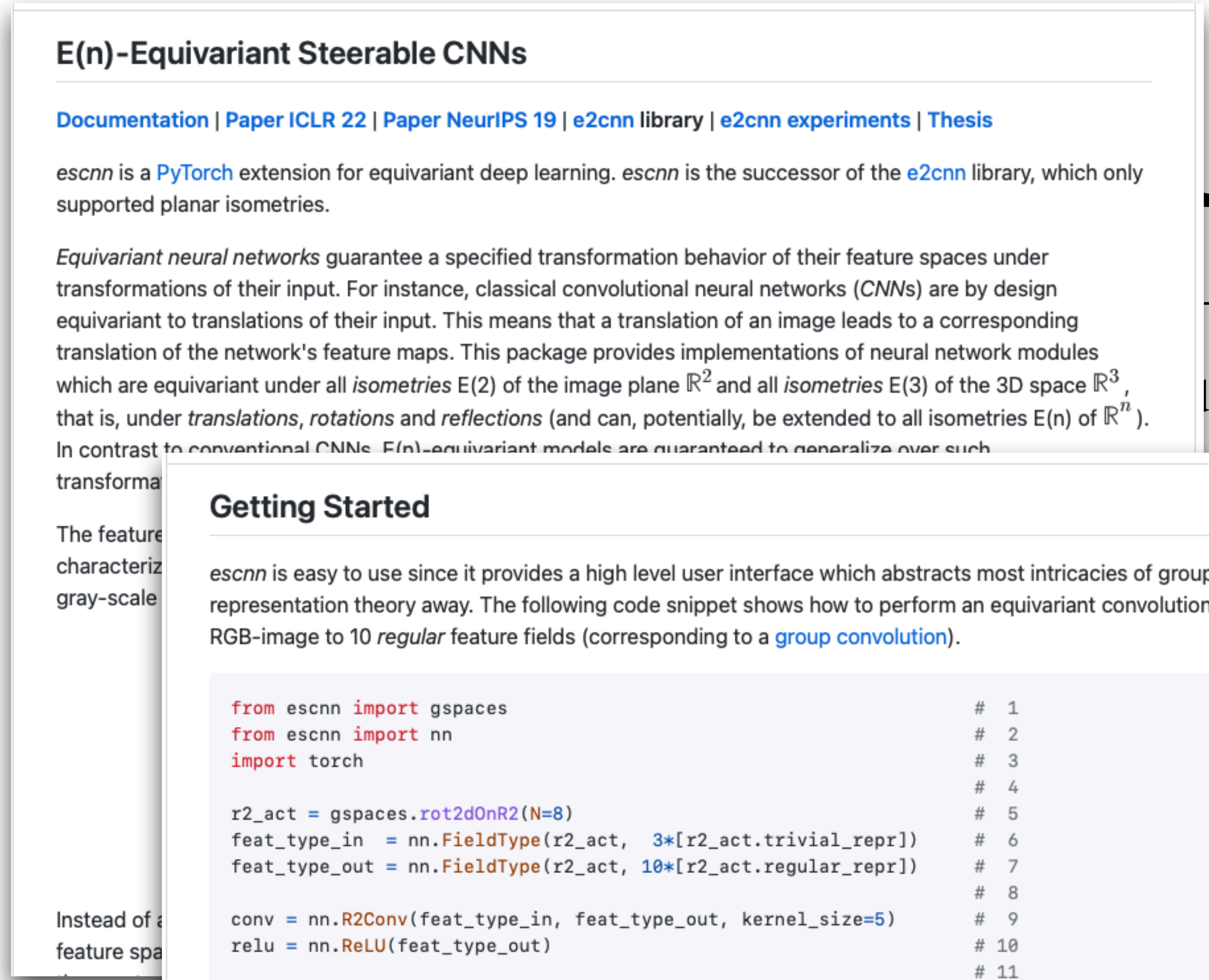
Normal convolution $\mathscr{K}[\hat{f}](\mathbf{x}) = \int_{\mathbb{R}^d} k(\mathbf{x}' - \mathbf{x}) f(\mathbf{x}')\mathrm{d}\mathbf{x}'$

but with kernel $k : \mathbb{R}^d \to \mathbb{R}^{d_Y \times d_X}$ satisfying **constraint**

$\forall_{h \in H}\ \forall_{\mathbf{x} \in \mathbb{R}^d}: \quad k(g\,\mathbf{x}) = \rho_Y(h) k(\mathbf{x}) \rho_X(h^{-1})$

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

up convolutions



E(n)-Equivariant Steerable CNNs

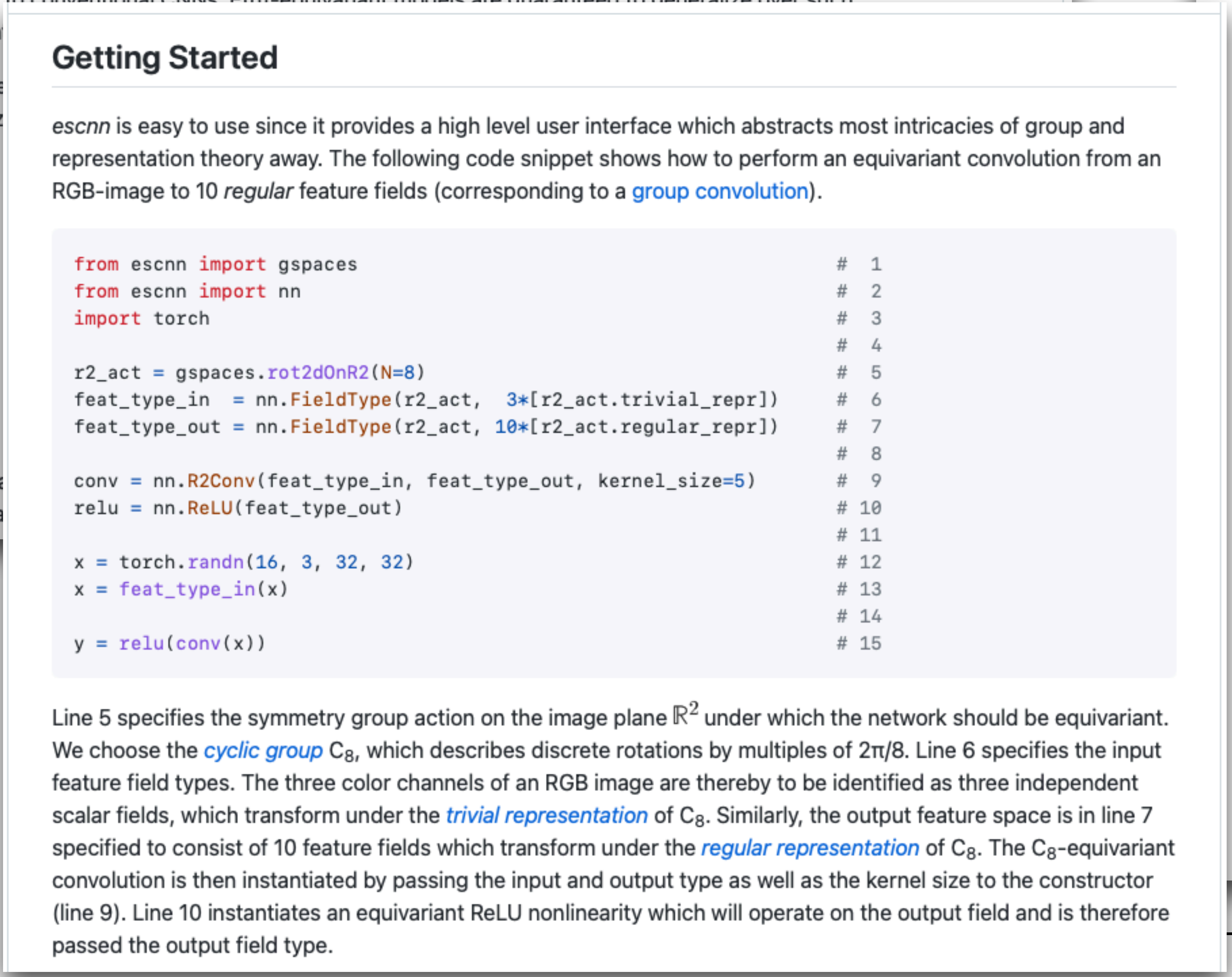
Documentation | Paper ICLR 22 | Paper NeurIPS 19 | e2cnn library | e2cnn experiments | Thesis

escnn is a PyTorch extension for equivariant deep learning. *escnn* is the successor of the e2cnn library, which only supported planar isometries.

Equivariant neural networks guarantee a specified transformation behavior of their feature spaces under transformations of their input. For instance, classical convolutional neural networks (*CNN*s) are by design equivariant to translations of their input. This means that a translation of an image leads to a corresponding translation of the network's feature maps. This package provides implementations of neural network modules which are equivariant under all *isometries* E(2) of the image plane $\mathbb{R}^2$ and all *isometries* E(3) of the 3D space $\mathbb{R}^3$, that is, under *translations*, *rotations* and *reflections* (and can, potentially, be extended to all isometries E(n) of $\mathbb{R}^n$).
In contrast to conventional CNNs, E(n)-equivariant models are guaranteed to generalize over such
transforma

The feature
characteriz
gray-scale

Instead of a
feature spa



Getting Started

escnn is easy to use since it provides a high level user interface which abstracts most intricacies of group and representation theory away. The following code snippet shows how to perform an equivariant convolution from an RGB-image to 10 *regular* feature fields (corresponding to a group convolution).

```python
from escnn import gspaces                                              #  1
from escnn import nn                                                   #  2
import torch                                                           #  3
                                                                       #  4
r2_act = gspaces.rot2dOnR2(N=8)                                        #  5
feat_type_in  = nn.FieldType(r2_act,  3*[r2_act.trivial_repr])         #  6
feat_type_out = nn.FieldType(r2_act, 10*[r2_act.regular_repr])         #  7
                                                                       #  8
conv = nn.R2Conv(feat_type_in, feat_type_out, kernel_size=5)           #  9
relu = nn.ReLU(feat_type_out)                                          # 10
                                                                       # 11
x = torch.randn(16, 3, 32, 32)                                         # 12
x = feat_type_in(x)                                                    # 13
                                                                       # 14
y = relu(conv(x))                                                      # 15
```

Line 5 specifies the symmetry group action on the image plane $\mathbb{R}^2$ under which the network should be equivariant. We choose the *cyclic group* C_8, which describes discrete rotations by multiples of $2\pi/8$. Line 6 specifies the input feature field types. The three color channels of an RGB image are thereby to be identified as three independent scalar fields, which transform under the *trivial representation* of C_8. Similarly, the output feature space is in line 7 specified to consist of 10 feature fields which transform under the *regular representation* of C_8. The C_8-equivariant convolution is then instantiated by passing the input and output type as well as the kernel size to the constructor (line 9). Line 10 instantiates an equivariant ReLU nonlinearity which will operate on the output field and is therefore passed the output field type.

Published as a conference paper at ICLR 2022

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

Abstract

Equivariance is becoming an increasingly popular design choice to build data efficient neural networks by exploiting prior knowledge about the symmetries of the problem at hand. Euclidean steerable CNNs are one of the most common classes of equivariant networks. While the constraints these architectures need to satisfy are understood, existing approaches are tailored to specific (classes of) groups. No generally applicable method that is *practical* for implementation has been described so far. In this work, we generalize the Wigner-Eckart theorem proposed in Lang & Weiler (2020), which characterizes general G-steerable kernel spaces for compact groups G over their homogeneous spaces, to arbitrary G-spaces. This enables us to directly parameterize filters in terms of a band-limited basis on the whole space rather than on G's orbits, but also to easily implement steerable CNNs equivariant to a large number of groups. To demonstrate its generality, we instantiate our method on a variety of isometry groups acting on the Euclidean space $\mathbb{R}^3$. Our framework allows us to build E(3) and SE(3)-steerable CNNs like previous works, but also CNNs with arbitrary $G \leq O(3)$-steerable kernels. For example, we build 3D CNNs equivariant to the symmetries of platonic solids or choose $G = SO(2)$ when working with 3D data having only azimuthal symmetries. We compare these models on 3D shapes and molecular datasets, observing improved performance by matching the model's symmetries to the ones of the data.

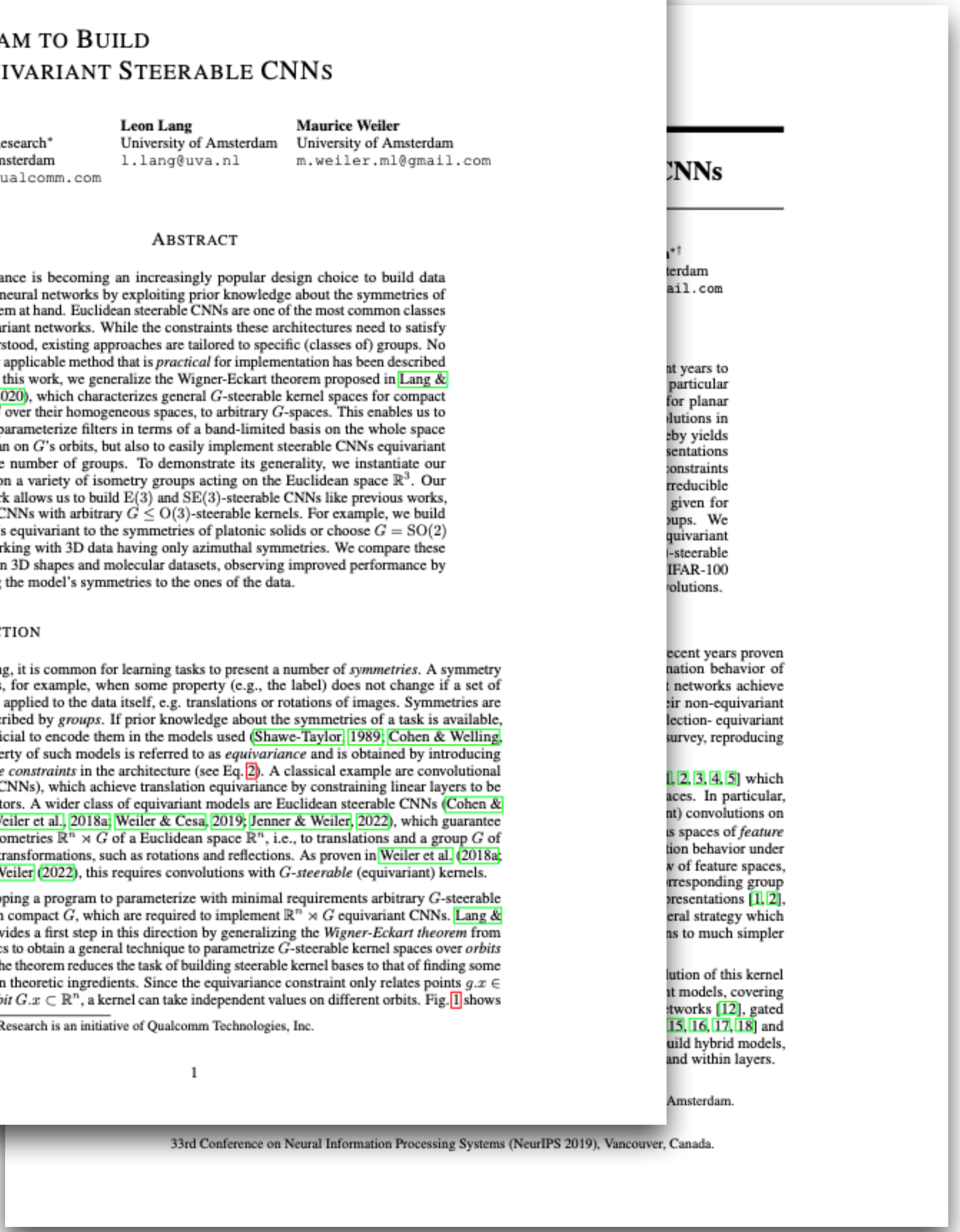
1 Introduction

In machine learning, it is common for learning tasks to present a number of *symmetries*. A symmetry in the data occurs, for example, when some property (e.g., the label) does not change if a set of transformations is applied to the data itself, e.g. translations or rotations of images. Symmetries are algebraically described by *groups*. If prior knowledge about the symmetries of a task is available, it is usually beneficial to encode them in the models used (Shawe-Taylor, 1989; Cohen & Welling, 2016a). The property of such models is referred to as *equivariance* and is obtained by introducing some *equivariance constraints* in the architecture (see Eq. 2). A classical example are convolutional neural networks (CNNs), which achieve translation equivariance by constraining linear layers to be convolution operators. A wider class of equivariant models are Euclidean steerable CNNs (Cohen & Welling, 2016b; Weiler et al., 2018a; Weiler & Cesa, 2019; Jenner & Weiler, 2022), which guarantee equivariance to isometries $\mathbb{R}^n \rtimes G$ of a Euclidean space $\mathbb{R}^n$, i.e., to translations and a group G of origin-preserving transformations, such as rotations and reflections. As proven in Weiler et al. (2018a; 2021); Jenner & Weiler (2022), this requires convolutions with G-*steerable* (equivariant) kernels.

Our goal is developing a program to parameterize with minimal requirements arbitrary G-steerable kernel spaces, with compact G, which are required to implement $\mathbb{R}^n \rtimes G$ equivariant CNNs. Lang & Weiler (2020) provides a first step in this direction by generalizing the *Wigner-Eckart theorem* from quantum mechanics to obtain a general technique to parametrize G-steerable kernel spaces over *orbits* of a compact G. The theorem reduces the task of building steerable kernel bases to that of finding some pure representation theoretic ingredients. Since the equivariance constraint only relates points $g.x \in \mathbb{R}^n$ in the same *orbit* $G.x \subset \mathbb{R}^n$, a kernel can take independent values on different orbits. Fig. 1 shows

*Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

1



CNNs

*†
terdam
ail.com

nt years to
particular
for planar
lutions in
eby yields
sentations
constraints
rreducible
given for
oups. We
quivariant
-steerable
IFAR-100
volutions.

ecent years proven
nation behavior of
t networks achieve
eir non-equivariant
lection- equivariant
survey, reproducing

1, 2, 3, 4, 5] which
aces. In particular,
nt) convolutions on
s spaces of *feature*
tion behavior under
v of feature spaces,
rresponding group
presentations [1, 2],
eral strategy which
ns to much simpler

lution of this kernel
nt models, covering
tworks [12], gated
15, 16, 17, 18] and
uild hybrid models,
and within layers.

Amsterdam.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

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

E(n)-Equivariant Steerable CNNs

Documentation | Paper ICLR 22 | Paper NeurIPS 19 | e2cnn library | e2cnn experiments | Thesis

escnn is a PyTorch extension for equivariant deep learning. *escnn* is the s
supported planar isometries.

Equivariant neural networks guarantee a specified transformation behavi
transformations of their input. For instance, classical convolutional neura
equivariant to translations of their input. This means that a translation of
translation of the network's feature maps. This package provides implem
which are equivariant under all *isometries* E(2) of the image plane $\mathbb{R}^2$ and
that is, under *translations*, *rotations* and *reflections* (and can, potentially,
In con
transf

The fe
chara
gray-s

Instea
featur

Getting Started

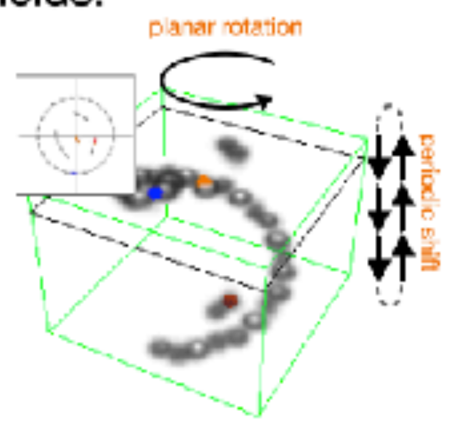
escnn is easy to use since it provides a high level user interfac
representation theory away. The following code snippet shows
RGB-image to 10 *regular* feature fields (corresponding to a gr

```
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.tri
feat_type_out = nn.FieldType(r2_act, 10*[r2_act.reg

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

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

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

Line 5 specifies the symmetry group action on the image plane $\mathbb{R}^2$ under which the network should be equivariant. We choose the *cyclic group* C_8, which describes discrete rotations by multiples of $2\pi/8$. Line 6 specifies the input feature field types. The three color channels of an RGB image are thereby to be identified as three independent scalar fields, which transform under the *trivial representation* of C_8. Similarly, the output feature space is in line 7 specified to consist of 10 feature fields which transform under the *regular representation* of C_8. The C_8-equivariant convolution is then instantiated by passing the input and output type as well as the kernel size to the constructor (line 9). Line 10 instantiates an equivariant ReLU nonlinearity which will operate on the output field and is therefore passed the output field type.

Feature field and induced representation

Regular G feature maps: $f(\mathbf{x}, h)$ considered so far can be considered feature fields.



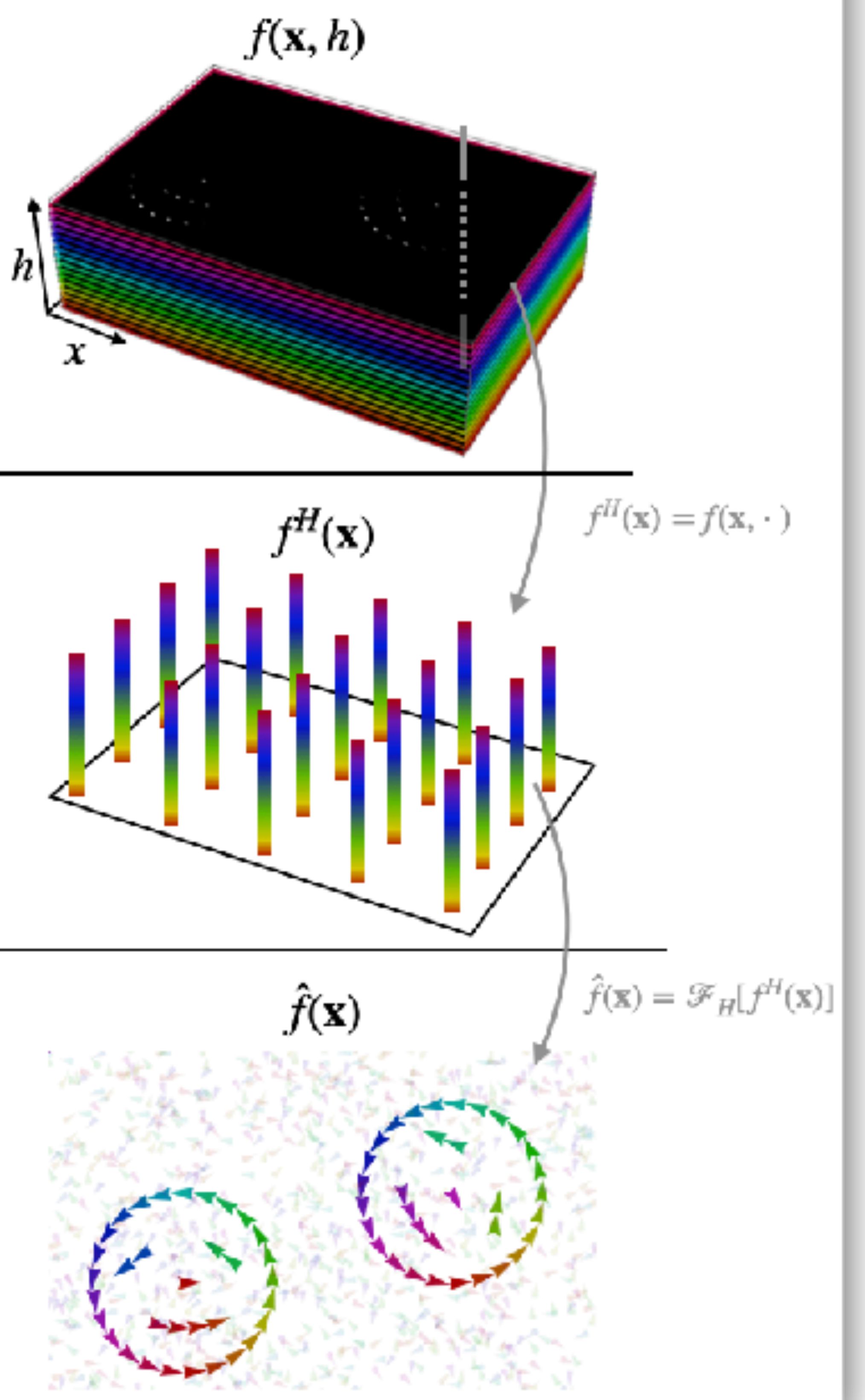
$$(\mathscr{L}_g f)(\mathbf{x}', h') = f(h^{-1}(\mathbf{x}' - \mathbf{x}), h^{-1}h')$$

Regular H feature fields: Let $f^H(\mathbf{x}) = f(\mathbf{x}, \cdot\,)$ be the field of functions $f^H(\mathbf{x}) : H \to \mathbb{R}$ on the subgroup H, then the functions (**fibers**) transform via the regular representation $\mathscr{L}^H_h$ (recall. $\mathscr{L}^H_h f(h') = f(h^{-1}h')$)

$$(\mathscr{L}_g f)(\mathbf{x}', h') \iff \left(\mathrm{Ind}^G_H[\ \mathscr{L}^H_h\ ](\mathbf{x}, h) f^H\right)(\mathbf{x}')$$

Steerable H feature fields: Since the fibers $f^H(\mathbf{x})$ are functions on H we can represent them via their Fourier coefficients $\hat{f}(\mathbf{x}) = \mathscr{F}_H[f^H(\mathbf{x})]$. These vectors of coefficients transform via irreps $\rho(h) = \bigoplus_l \rho_l(h)$

$$(\mathscr{L}_g f)(\mathbf{x}', h') \iff \left(\mathrm{Ind}^G_H[\ \mathscr{L}^H_h\ ](\mathbf{x}, h)\hat{f}\right)(\mathbf{x}') \iff \left(\mathrm{Ind}^G_H[\ \rho(h)\ ](\mathbf{x}, h)\hat{f}\right)(\mathbf{x}')$$



$f(\mathbf{x}, h)$
h
x
$f^H(\mathbf{x})$
$f^H(\mathbf{x}) = f(\mathbf{x}, \cdot\,)$
$\hat{f}(\mathbf{x})$
$\hat{f}(\mathbf{x}) = \mathscr{F}_H[f^H(\mathbf{x})]$

kernel spaces, with compact G, which are required to implement $\mathbb{R}^n \rtimes G$ equivariant CNNs. Lang & Weiler (2020) provides a first step in this direction by generalizing the *Wigner-Eckart theorem* from quantum mechanics to obtain a general technique to parametrize G-steerable kernel spaces over *orbits* of a compact G. The theorem reduces the task of building steerable kernel bases to that of finding some pure representation theoretic ingredients. Since the equivariance constraint only relates points $g.x \in \mathbb{R}^n$ in the same *orbit* $G.x \subset \mathbb{R}^n$, a kernel can take independent values on different orbits. Fig. 1 shows

*Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

1

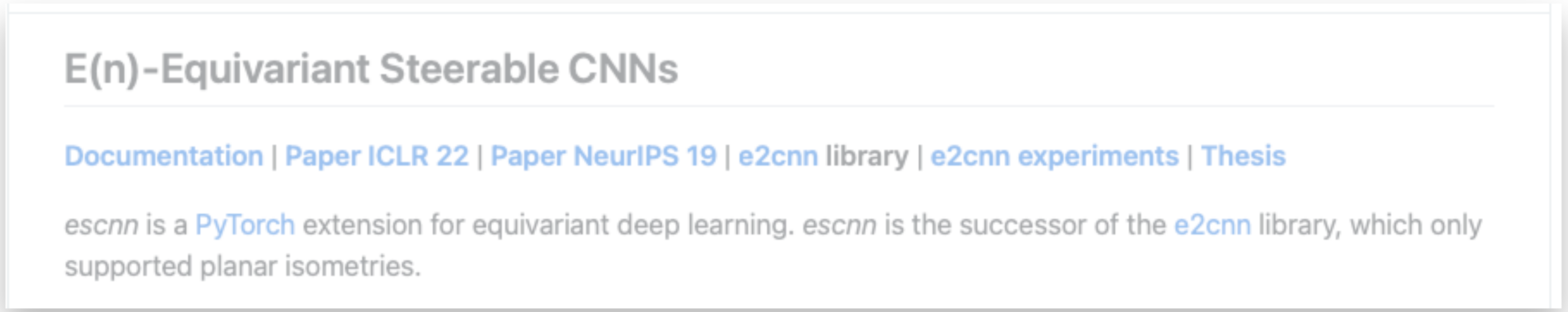
eral strategy which
ns to much simpler

lution of this kernel
nt models, covering
etworks [12], gated
15, 16, 17, 18] and
uild hybrid models,
and within layers.

Amsterdam.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Steerable group convolutions

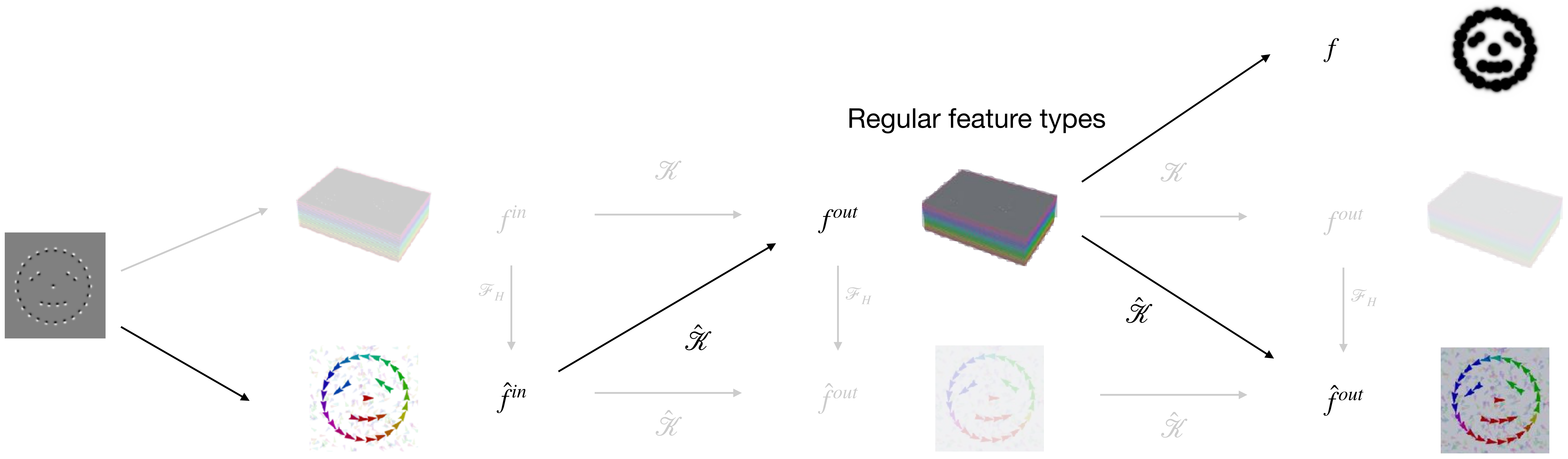


E(n)-Equivariant Steerable CNNs

Documentation | Paper ICLR 22 | Paper NeurIPS 19 | e2cnn library | e2cnn experiments | Thesis

escnn is a PyTorch extension for equivariant deep learning. *escnn* is the successor of the e2cnn library, which only supported planar isometries.

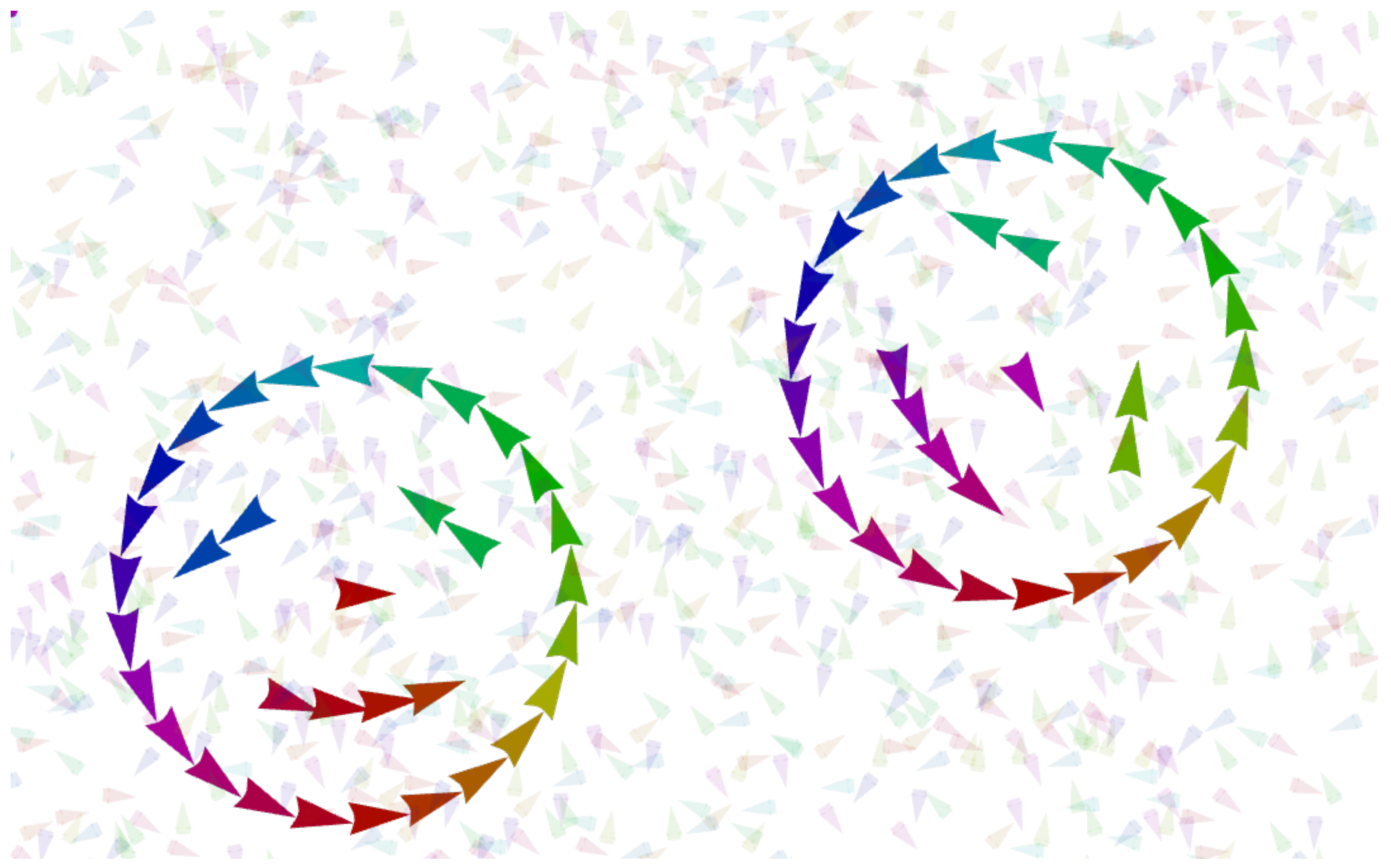
https://github.com/QUVA-Lab/escnn



Type-0 field usual 2D feature map (e.g. for segmentation)
Regular feature types
Steerable (irrep) feature types
Type-1 vector fields (e.g. force/velocity vectors)
f
f^{in}
f^{out}
$\hat{f}^{in}$
$\hat{f}^{out}$
$\mathcal{K}$
$\hat{\mathcal{K}}$
$\mathcal{F}_H$

Feature field types

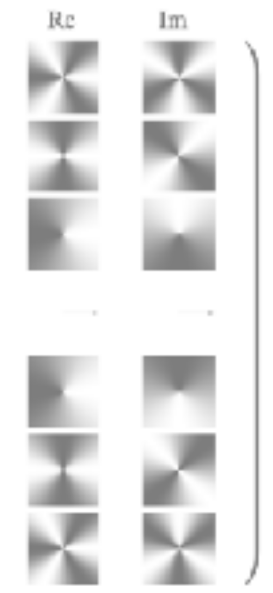
- A feature field is defined by its type ρ
- Feature fields, each of their own type ρ_l, can be stacked:
  - should be thought of as the channels in standard CNNs
- The sub-vectors/channels in these fields:
  - live in their own sub-vector spaces V_l
  - transform by their own representations ρ_l



Different types ρ

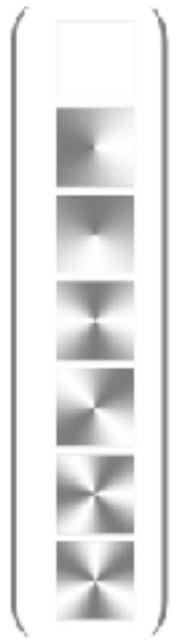
Complex irreps

$$\begin{pmatrix} e^{3i\theta} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{2i\theta} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e^{1i\theta} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{-1i\theta} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{-2i\theta} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{-3i\theta} \end{pmatrix}$$



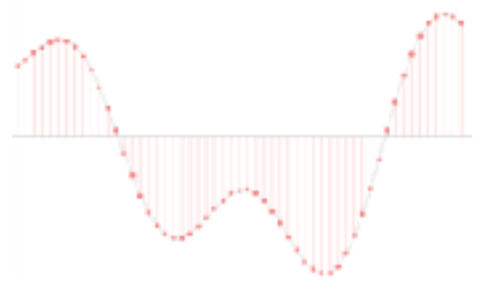
Real irreps

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 & 0 & 0 & 0 \\ 0 & -\sin\theta & \cos\theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos 2\theta & \sin 2\theta & 0 & 0 \\ 0 & 0 & 0 & -\sin 2\theta & \cos 2\theta & 0 & 0 \\ & 0 & 0 & 0 & 0 & \cos 3\theta & \sin 3\theta \\ & 0 & 0 & 0 & 0 & -\sin 3\theta & \cos 3\theta \end{pmatrix}$$



Regular reps

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}^n \begin{pmatrix} 0.21 \\ 0.20 \\ -0.29 \\ -0.16 \\ -0.39 \\ 0.02 \\ 0.34 \end{pmatrix}$$



Feature field types

- A feature field is defined by its type ρ
- Feature fields, each of their own type ρ_l, can be stacked:
  - should be thought of as the channels in standard CNNs
- The sub-vectors/channels in these fields:
  - live in their own sub-vector spaces V_l
  - transform by their own representations ρ_l
- Example notations (ρ_l denote irreps)

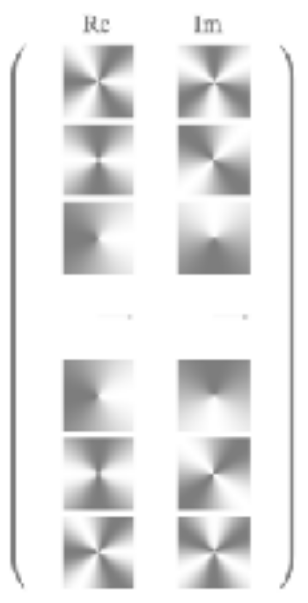
$$\rho = \rho_1 \oplus \rho_2 \oplus \rho_3 \oplus \dots$$ (Steerable G-CNNs | Fourier)

$$\rho = n\rho_0$$ (Normal CNNs with isotropic kernels)

$$\rho = n\mathscr{L}^H$$ (Regular G-CNNs)

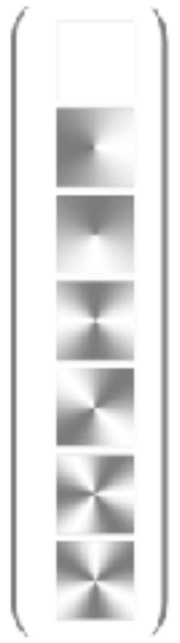
Complex irreps

$$\begin{pmatrix} e^{3i\theta} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{2i\theta} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e^{1i\theta} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{-1i\theta} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{-2i\theta} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{-3i\theta} \end{pmatrix}$$



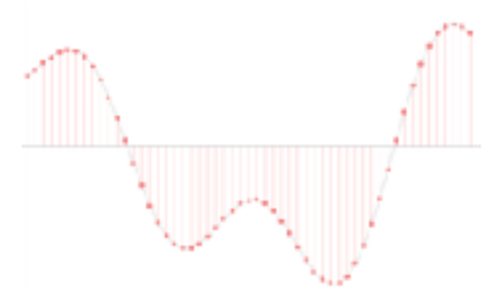
Real irreps

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 & 0 & 0 & 0 \\ 0 & -\sin\theta & \cos\theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos 2\theta & \sin 2\theta & 0 & 0 \\ 0 & 0 & 0 & -\sin 2\theta & \cos 2\theta & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cos 3\theta & \sin 3\theta \\ 0 & 0 & 0 & 0 & 0 & -\sin 3\theta & \cos 3\theta \end{pmatrix}$$



Regular reps

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}^n \begin{pmatrix} 0.21 \\ 0.20 \\ -0.29 \\ -0.16 \\ -0.39 \\ 0.02 \\ 0.34 \end{pmatrix}$$



Feature field t

- A feature field is defined by its
- Feature fields, each of their ov
  - should be thought of as th
- The sub-vectors/channels in t
  - live in their own sub-vecto
  - transform by their own rep
- Example notations (ρ_l denote irreps)

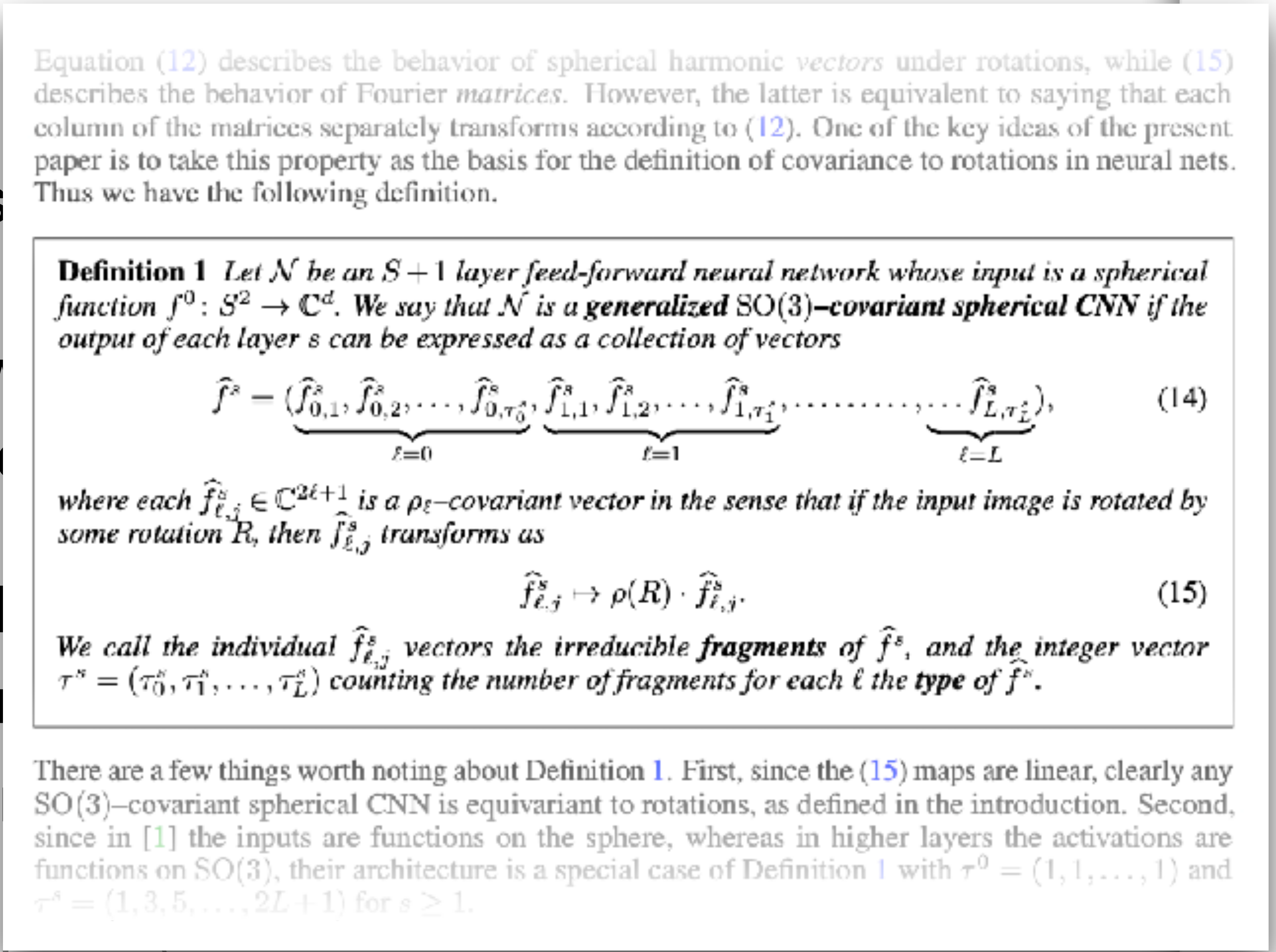
$$\rho = \rho_1 \oplus \rho_2 \oplus \rho_3 \oplus \dots$$ (Steerable G-CNNs | Fourier)

$$\rho = 4\rho_0 \oplus 9\rho_1 \oplus \dots$$ (Steerable G-CNNs | General)

define $$n\rho_l = \underbrace{\rho_l \oplus \rho_l \oplus \dots \oplus \rho_l}_{\text{n times}}$$

$$\rho = n\rho_0$$ (Normal CNNs with isotropic kernels)

$$\rho = n\mathscr{L}^H$$ (Regular G-CNNs)



Equation (12) describes the behavior of spherical harmonic *vectors* under rotations, while (15) describes the behavior of Fourier *matrices*. However, the latter is equivalent to saying that each column of the matrices separately transforms according to (12). One of the key ideas of the present paper is to take this property as the basis for the definition of covariance to rotations in neural nets. Thus we have the following definition.

Definition 1 *Let $\mathcal{N}$ be an $S+1$ layer feed-forward neural network whose input is a spherical function $f^0 : S^2 \to \mathbb{C}^d$. We say that $\mathcal{N}$ is a* ***generalized*** *SO(3)–**covariant spherical CNN** if the output of each layer s can be expressed as a collection of vectors*

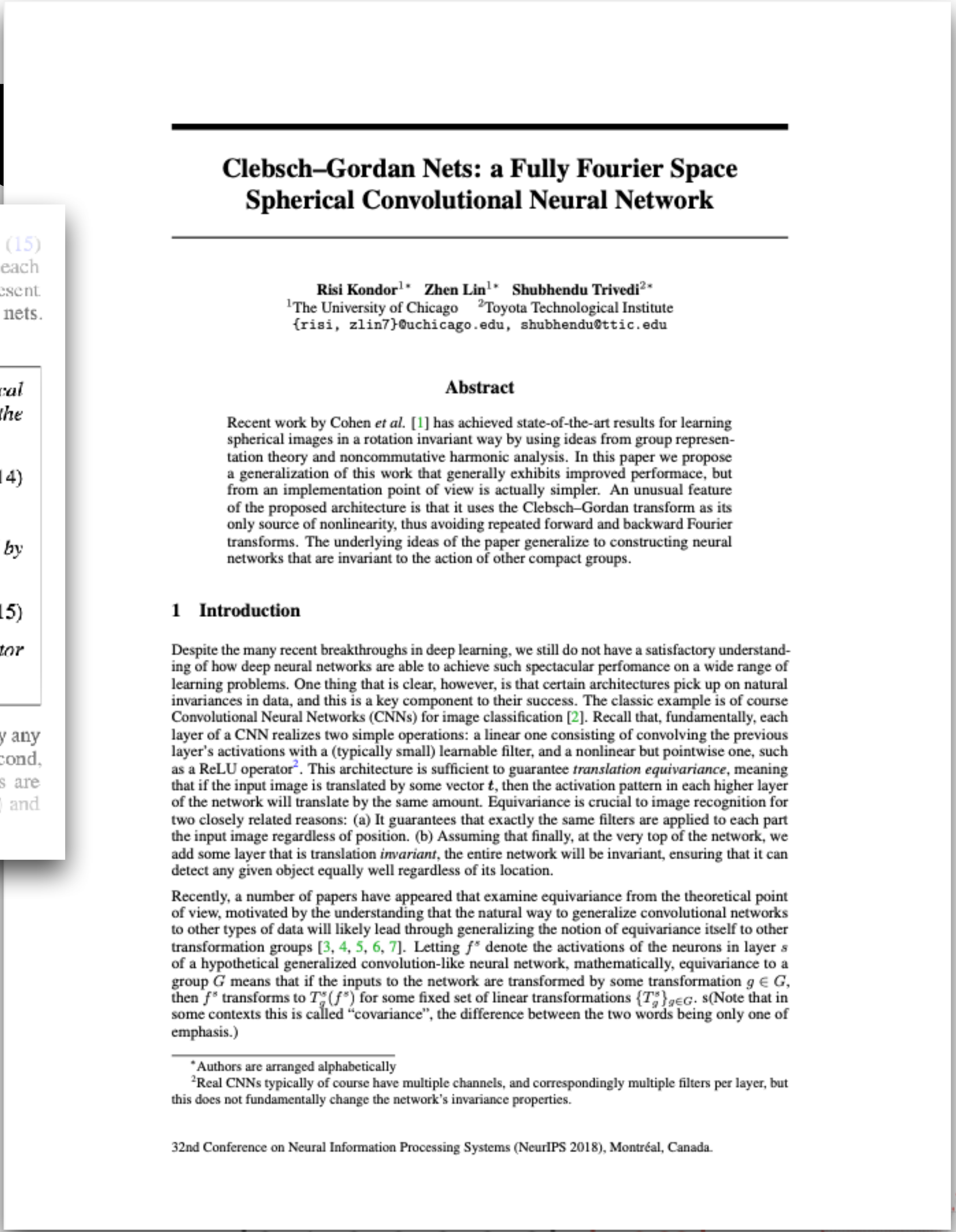
$$\widehat{f}^s = (\underbrace{\widehat{f}^s_{0,1}, \widehat{f}^s_{0,2}, \dots, \widehat{f}^s_{0,\tau^s_0}}_{\ell=0}, \underbrace{\widehat{f}^s_{1,1}, \widehat{f}^s_{1,2}, \dots, \widehat{f}^s_{1,\tau^s_1}}_{\ell=1}, \dots\dots\dots, \underbrace{\dots \widehat{f}^s_{L,\tau^s_L}}_{\ell=L}), \tag{14}$$

where each $\widehat{f}^s_{\ell,j} \in \mathbb{C}^{2\ell+1}$ is a ρ_ℓ–covariant vector in the sense that if the input image is rotated by some rotation R, then $\widehat{f}^s_{\ell,j}$ transforms as

$$\widehat{f}^s_{\ell,j} \mapsto \rho(R) \cdot \widehat{f}^s_{\ell,j}. \tag{15}$$

We call the individual $\widehat{f}^s_{\ell,j}$ vectors the irreducible ***fragments*** *of $\widehat{f}^s$, and the integer vector $\tau^s = (\tau^s_0, \tau^s_1, \dots, \tau^s_L)$ counting the number of fragments for each ℓ the* ***type*** *of $\widehat{f}^s$.*

There are a few things worth noting about Definition 1. First, since the (15) maps are linear, clearly any SO(3)–covariant spherical CNN is equivariant to rotations, as defined in the introduction. Second, since in [1] the inputs are functions on the sphere, whereas in higher layers the activations are functions on SO(3), their architecture is a special case of Definition 1 with $\tau^0 = (1, 1, \dots, 1)$ and $\tau^s = (1, 3, 5, \dots, 2L+1)$ for $s \geq 1$.



Clebsch–Gordan Nets: a Fully Fourier Space Spherical Convolutional Neural Network

Risi Kondor[1*] **Zhen Lin**[1*] **Shubhendu Trivedi**[2*]
[1]The University of Chicago [2]Toyota Technological Institute
{risi, zlin7}@uchicago.edu, shubhendu@ttic.edu

Abstract

Recent work by Cohen *et al.* [1] has achieved state-of-the-art results for learning spherical images in a rotation invariant way by using ideas from group representation theory and noncommutative harmonic analysis. In this paper we propose a generalization of this work that generally exhibits improved performace, but from an implementation point of view is actually simpler. An unusual feature of the proposed architecture is that it uses the Clebsch–Gordan transform as its only source of nonlinearity, thus avoiding repeated forward and backward Fourier transforms. The underlying ideas of the paper generalize to constructing neural networks that are invariant to the action of other compact groups.

1 Introduction

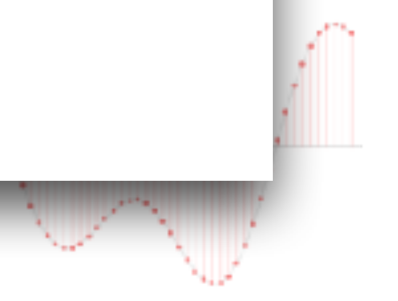
Despite the many recent breakthroughs in deep learning, we still do not have a satisfactory understanding of how deep neural networks are able to achieve such spectacular perfomance on a wide range of learning problems. One thing that is clear, however, is that certain architectures pick up on natural invariances in data, and this is a key component to their success. The classic example is of course Convolutional Neural Networks (CNNs) for image classification [2]. Recall that, fundamentally, each layer of a CNN realizes two simple operations: a linear one consisting of convolving the previous layer's activations with a (typically small) learnable filter, and a nonlinear but pointwise one, such as a ReLU operator[2]. This architecture is sufficient to guarantee *translation equivariance*, meaning that if the input image is translated by some vector t, then the activation pattern in each higher layer of the network will translate by the same amount. Equivariance is crucial to image recognition for two closely related reasons: (a) It guarantees that exactly the same filters are applied to each part the input image regardless of position. (b) Assuming that finally, at the very top of the network, we add some layer that is translation *invariant*, the entire network will be invariant, ensuring that it can detect any given object equally well regardless of its location.

Recently, a number of papers have appeared that examine equivariance from the theoretical point of view, motivated by the understanding that the natural way to generalize convolutional networks to other types of data will likely lead through generalizing the notion of equivariance itself to other transformation groups [3, 4, 5, 6, 7]. Letting f^s denote the activations of the neurons in layer s of a hypothetical generalized convolution-like neural network, mathematically, equivariance to a group G means that if the inputs to the network are transformed by some transformation $g \in G$, then f^s transforms to $T^s_g(f^s)$ for some fixed set of linear transformations $\{T^s_g\}_{g \in G}$. s(Note that in some contexts this is called "covariance", the difference between the two words being only one of emphasis.)

*Authors are arranged alphabetically
[2]Real CNNs typically of course have multiple channels, and correspondingly multiple filters per layer, but this does not fundamentally change the network's invariance properties.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} -0.29 \\ -0.16 \\ -0.39 \\ 0.02 \\ 0.34 \end{pmatrix}$$



Feature field types

- A feature field is defined by its type ρ
- Feature fields, each of their own type ρ_l, can be stacked:
  - should be thought of as the channels in standard CNNs
- The sub-vectors/channels in these fields:
  - live in their own sub-vector spaces V_l
  - transform by their own representations ρ_l
- Example notations (ρ_l denote irreps)

$\rho = \rho_1 \oplus \rho_2 \oplus \rho_3 \oplus \dots$ (Steerable G-CNNs | Fourier)

$\rho = 4\rho_0 \oplus 9\rho_1 \oplus \dots$ (Steerable G-CNNs | General)

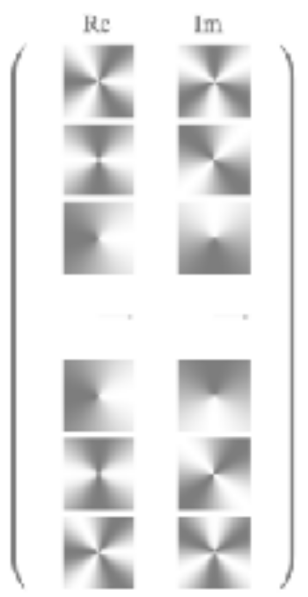
define $n\rho_l = \underbrace{\rho_l \oplus \rho_l \oplus \dots \oplus \rho_l}_{\text{n times}}$

$\rho = n\rho_0$ (Normal CNNs with isotropic kernels)

$\rho = n\mathscr{L}^H$ (Regular G-CNNs)

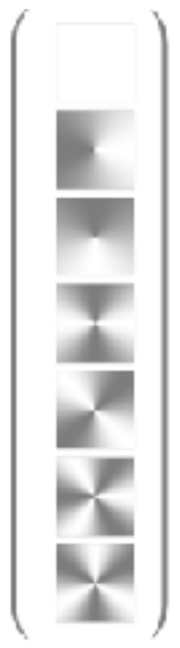
Complex irreps

$$\begin{pmatrix} e^{3i\theta} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{2i\theta} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e^{1i\theta} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{-1i\theta} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{-2i\theta} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{-3i\theta} \end{pmatrix}$$



Real irreps

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 & 0 & 0 & 0 \\ 0 & -\sin\theta & \cos\theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos 2\theta & \sin 2\theta & 0 & 0 \\ 0 & 0 & 0 & -\sin 2\theta & \cos 2\theta & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cos 3\theta & \sin 3\theta \\ 0 & 0 & 0 & 0 & 0 & -\sin 3\theta & \cos 3\theta \end{pmatrix}$$



Regular reps

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}^n \begin{pmatrix} 0.21 \\ 0.20 \\ -0.29 \\ -0.16 \\ -0.39 \\ 0.02 \\ 0.34 \end{pmatrix}$$