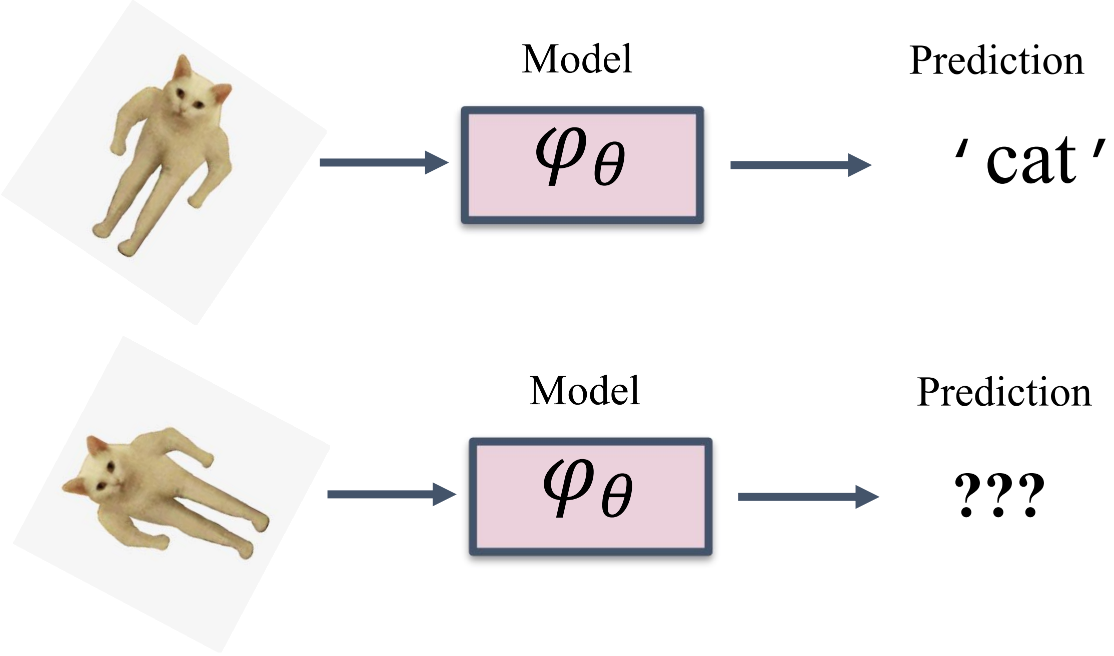
Symmetries in ML models and Group Equivariance II

Wenhan Gao

AMS, Stony Brook University

Recap: Group Equivariance in ML

Group equivariance in ML models is about **enforcing symmetries** in our architectures. In a nutshell, group equivariance means that **if the input is transformed, the output will be changed in a predictable way.**



Issues in Data Augmentation:

- ✗ **No guarantee** of having symmetries in the model
- ✗ **Wasting valuable net capacity** on learning symmetries from data
- ✗ **Redundancy** in learned feature representation

Solution:

- ✓ Building symmetries into the model by design!

Building symmetry into ML has led to major breakthroughs in deep learning:

- Imposing **translational symmetry** and parameter sharing allowed CNNs to essentially solve computer vision.
- Group Equivariance **conceptualizes CNN success (symmetry exploitation) and generalizes it to tasks with other symmetries.**

Recap: Equivariance and Invariance

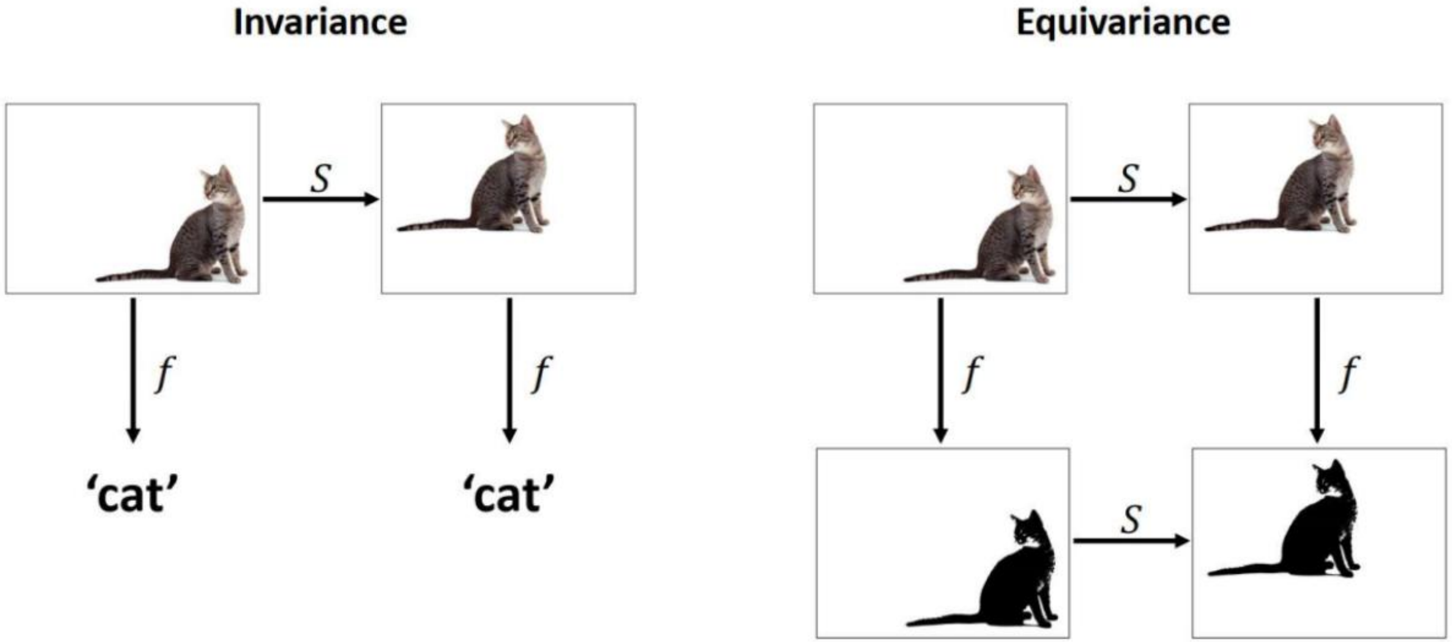
Equivariance is a property of an operator $\Phi : X \to Y$ (such as a neural network layer) by which it commutes with the group action:

$$\Phi \circ \rho^X(g) = \rho^Y(g) \circ \Phi,$$

Invariance is a property of an operator $\Phi : X \to Y$ (such as a neural network layer) by which it remains unchanged after the group action:

$$\Phi \circ \rho^X(g) = \Phi,$$

- $\rho^X(g)$: group representation action on X
- $\rho^Y(g)$: group representation action on Y
- Invariance is a special case of equivariance when $\rho^Y(g)$ is the identity.

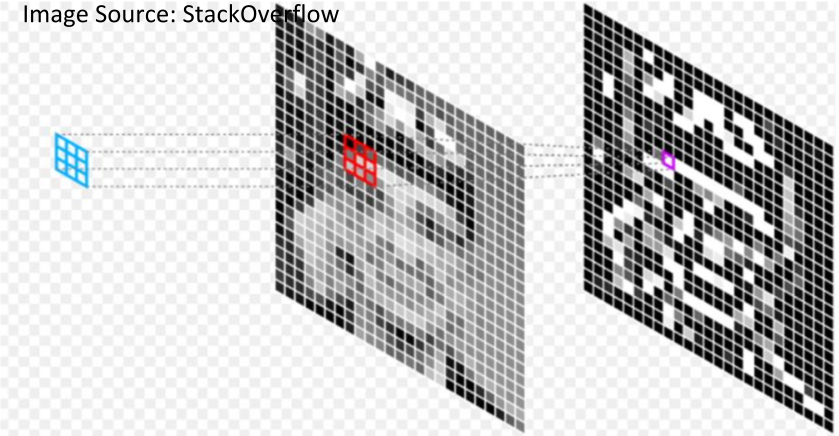


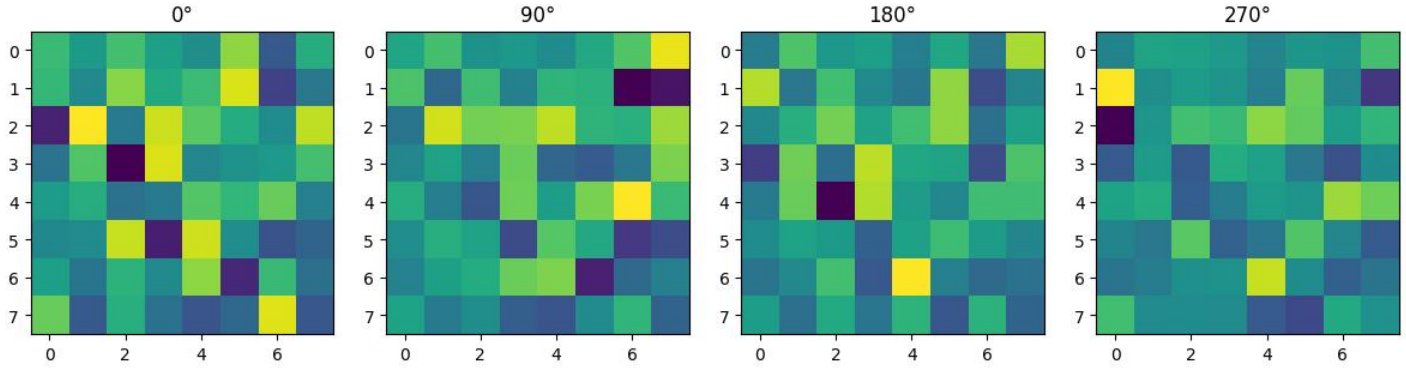
Recap: Regular Group CNN

High-level Ideas:

❖ In CNNs, we scan through many translated version of the image while creating the feature map. If we need to compute the cross-correlation for a transformed image, we can just go and look up the relevant outputs, because we have already computed them. :

Image Source: StackOverflow





CNN: One Feature Map

Group CNN: Multiple Feature Maps

Image Source: StackOverflow

❖ In Group-CNNs, we follow the same idea, we lift the image to the group space and scan through all the transformed version of the image.

However, the group size can be large or even infinite (e.g. continuous rotations). It becomes computationally intractable to lift the image to the group space.

Structure of Today's Talk

This talk will cover the following topics:

- ❖ Group Averaging and Frame Averaging
- ❖ Examples on A Few Common Symmetries

Mathematical Preliminary: Fourier Basis and Series

The set of functions

$$\{1, \cos(n\theta), \sin(n\theta)\}_{n=1}^{\infty}$$

forms a complete orthogonal basis for functions defined on the interval $[0, 2\pi]$ under suitable conditions.

WLOG, we can write the complete orthonormal basis as:

$$\{\frac{1}{\sqrt{2\pi}}, \frac{\cos(\theta)}{\sqrt{\pi}}, \frac{\sin(\theta)}{\sqrt{\pi}}, \frac{\cos(2\theta)}{\sqrt{\pi}}, \frac{\sin(2\theta)}{\sqrt{\pi}}, \frac{\cos(3\theta)}{\sqrt{\pi}}, \frac{\sin(3\theta)}{\sqrt{\pi}}, \ldots\}$$

The fourier series for functions defined on the interval

$$\begin{aligned} f(\theta) &= \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos n\theta + b_n \sin n\theta) \\ &= \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \frac{e^{in\theta} + e^{-in\theta}}{2} + b_n \frac{e^{in\theta} - e^{-in\theta}}{2i} \right) \\ &= \frac{a_0}{2} + \sum_{n=1}^{\infty} \frac{a_n - ib_n}{2} e^{in\theta} + \sum_{n=1}^{\infty} \frac{a_n + ib_n}{2} e^{-in\theta} \\ &= \sum_{n=-\infty}^{\infty} c_n e^{in\theta}, \end{aligned}$$

where $c_0 = \frac{a_0}{2}, \;\; c_n = \frac{a_n - ib_n}{2}, \;\; c_{-n} = \frac{a_n + ib_n}{2}$.

The coefficients can be obtained by

$$c_n = \frac{1}{2\pi} \int_0^{2\pi} f(\theta)\, e^{-in\theta} d\theta, \;\; n = 0, \pm 1, \pm 2, \ldots$$

Any function defined on $\mathbb{L}_2(S_1)$ can be written as a fourier series.

Mathematical Preliminary: Shift Theorem of Fourier Series

Once we have the coefficients c_n, a translation of $f(\theta)$ corresponds to a "phase shift":

$$
\begin{aligned}
c_n' &= \frac{1}{2\pi}\int_0^{2\pi} f(\theta-\phi)e^{-in\theta}d\theta \\
&= \frac{1}{2\pi}\int_0^{2\pi} f\left(\theta'\right)e^{-in\left(\theta'+\phi\right)}d\theta' \\
&= \frac{1}{2\pi}\int_0^{2\pi} f\left(\theta'\right)e^{-in\theta'}e^{-in\phi}d\theta' \\
&= e^{-in\phi}\frac{1}{2\pi}\int_0^{2\pi} f\left(\theta'\right)e^{-in\theta'}d\theta' \\
&= e^{-in\phi}c_n.
\end{aligned}
$$

What this implies is that once we have the coefficients c_n for $f(\theta)$, we can quickly get the new coefficients for a translated version, $f(\theta-\phi)$ just by a "phase shift".

The Fourier Series can be viewed as a special case of the Fourier Transform when dealing with periodic functions and finite intervals.

Recap: Different Types of Cross Correlations

- Regular Cross Correlation

$$(k \star f)(x) := \int_{\mathbb{R}^d} k(x' - x) f(x') dx' = \int_{\mathbb{R}^d} [\mathscr{L}_x k(x')] f(x') dx' = \langle \mathscr{L}_x k, f \rangle_{\mathbb{L}_2(\mathbb{R}^d)} = \langle k, \mathscr{L}_{-x} f \rangle_{\mathbb{L}_2(\mathbb{R}^d)}.$$

The domain is the translation group

Integral over the Translation Group

Translated Kernels

- Lifting Cross Correlation

$$(k \star_{SE(2)} f)(x, \theta) := \int_{\mathbb{R}^2} k\left(\mathbf{R}_\theta^{-1}(x' - x)\right) f(x') dx' = \int_{\mathbb{R}^2} [\mathscr{L}_{g=(x,\theta)} k(x')] f(x') dx' = \langle \mathscr{L}_{g=(x,\theta)} k, f \rangle_{\mathbb{L}_2(\mathbb{R}^2)}.$$

The domain of the input is R(2)

Still R(2) for lifting operation because the function is define on R(2)

The domain of the output is lifted to the SE(2) group

The kernel reflects the SE(2) group now

Regular Group CNN and SE(2) Equivariance: SE(2) Cross Correlation

$$(k \star_{SE(2)} f)(x,\theta) := \int_{SE(2)} k\left(\mathbf{R}_\theta^{-1}(x'-x), \theta'-\theta \mod 2\pi\right) f(x',\theta')\, d\theta'\, dx'$$

The domain of the input is SE(2)

The domain of the output is still the SE(2) group

Integrate over SE(2) because the function is now define on SE(2)

Planar Rotation

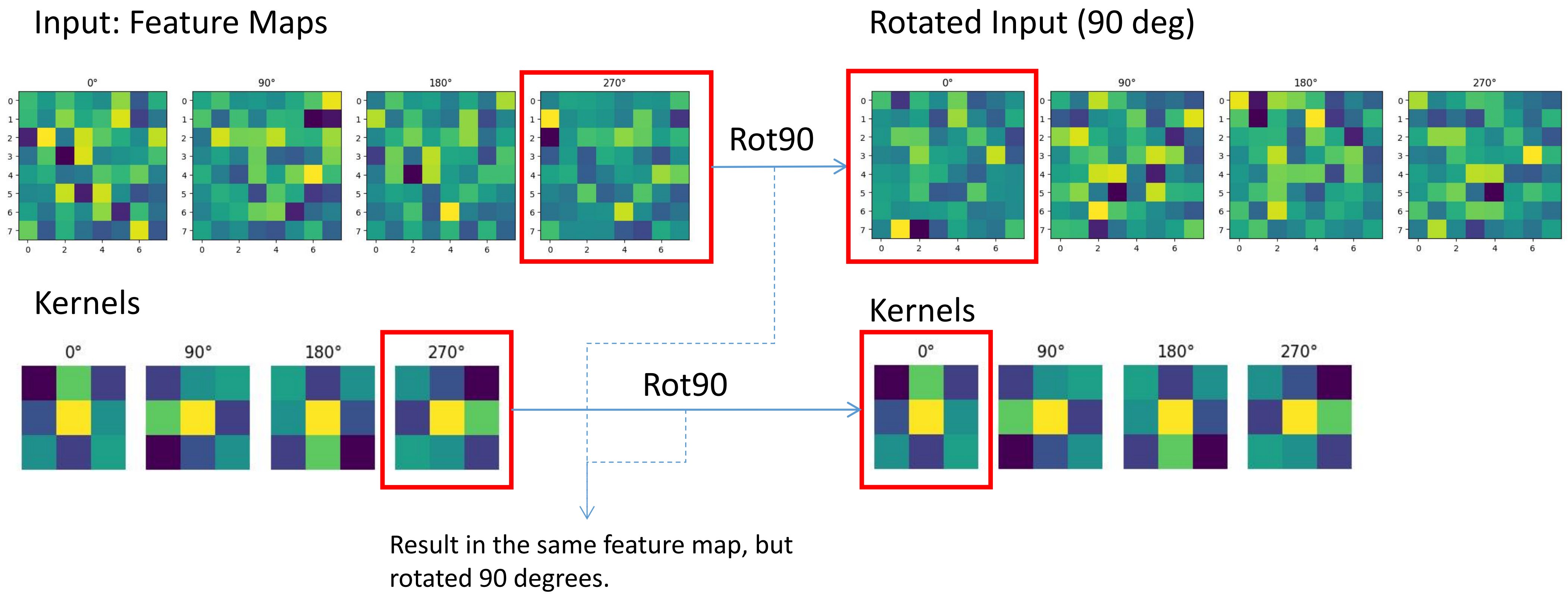
Periodic Shift

Reflect the SE(2) Group

Regular Group CNN and SE(2) Equivariance: SE(2) Cross Correlation

The goal is still

(Rotate the Input) ↺ = ⟶ + ↺ (Periodic Shift + Planar Rotation for the Output)



Thus, the resulting feature maps will still be rotated and periodically shifted. It seems that so far, we only used $\mathbf{R}_\theta^{-1}(x' - x)$, but recall that, in group correlation, we also have $\theta' - \theta$. Now, imagine when the input is rotated 180 deg, the above equivariance does not hold anymore. That's why we actually need to have convolution on the theta axis as well.

Regular Group CNN and SE(2) Equivariance: More Intuition

Although the examples are given for the group SE(2), the idea can generalize to other affine groups (semi-direct product groups).

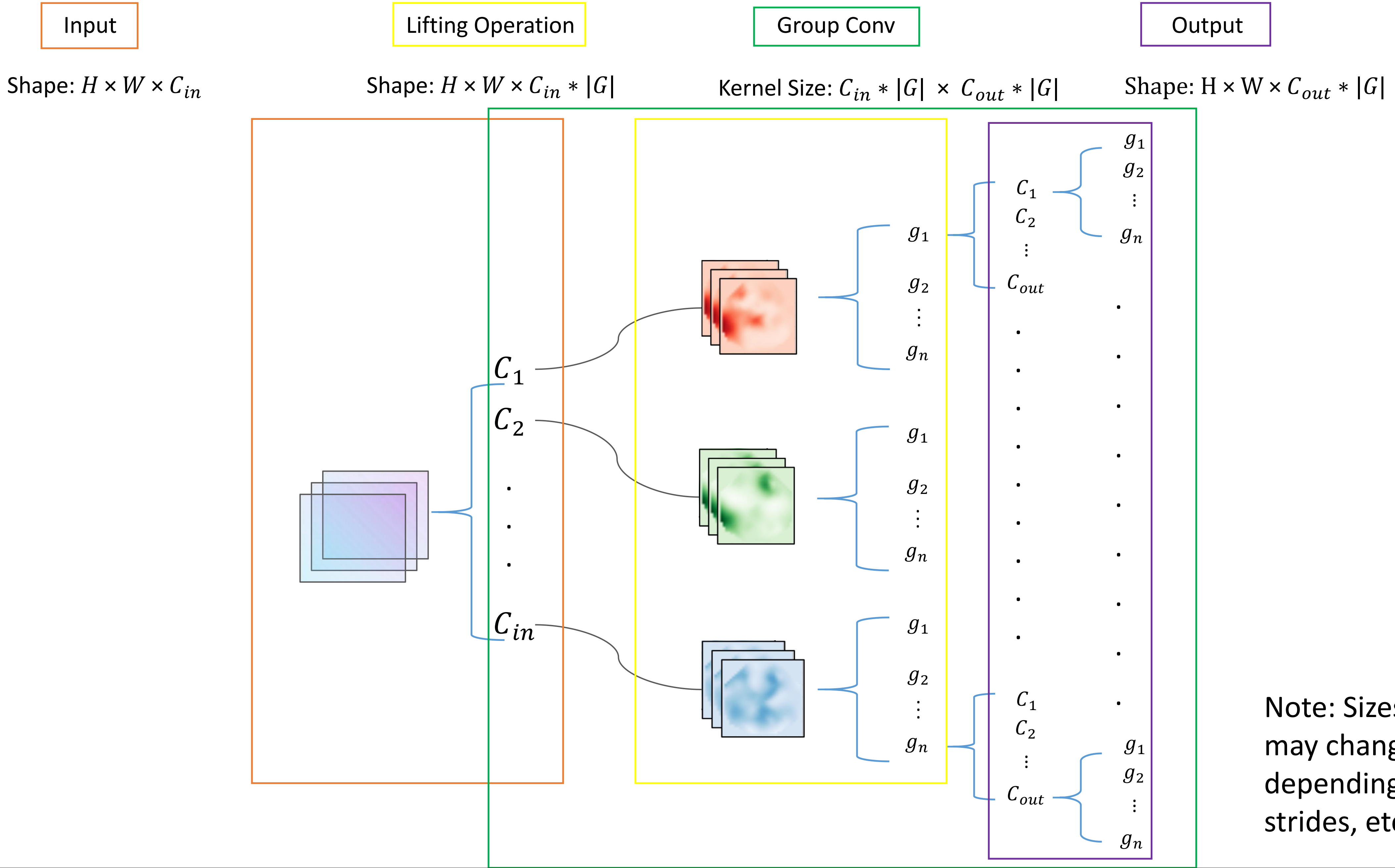
If we look carefully at how rotational equivariance is achieved, we find that it basically adds a rotation dimension represented by an axis θ, and thus, rotational equivariance problem now becomes translation equivariance problem which can be solved easily by 1D convolution/cross-correlation.

$$\text{translational weight sharing} \iff \text{translation group equivariance}$$
$$\text{affine weight sharing} \iff \text{affine group equivariance}$$

Note: Translations and H-transformations form so-called affine groups

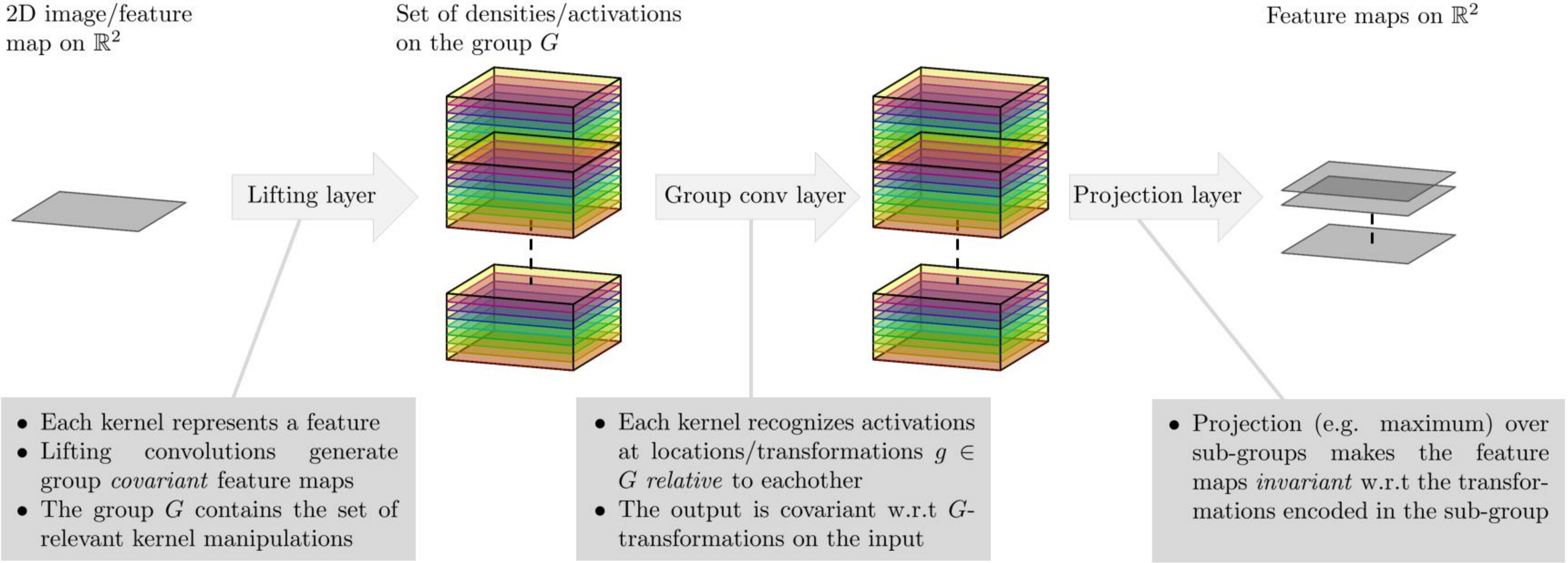
$$\text{Aff}(H) := (\mathbb{R}^d, +) \rtimes H.$$

An overview of actual implementation with nn.Conv2d()



Note: Sizes of H and W may change depending on padding, strides, etc..

Regular Group CNN and SE(2) Equivariance: Example



1. Lifting Layer (Generate group equivariant feature maps):
   - 2D input ⇒ 3D feature map with the third dimension being rotation.
2. Group Conv Layer (Group equivariant on the input):
   - 3D feature map ⇒ 3D feature map
3. Projection Layer:
   - Invariance: 3D feature map ⇒ 2D feature map by (e.g. max/avg) pooling over θ dimension. Now, it is invairant in θ dimension.
   - Equivariance: The resulting 2D feature map is rotation equivariant w.r.t. the input.

Image Source: https://uvagedl.github.io/

Results: Group Equivariant Convolutional Networks

Results on datasets with rotations: The rotated MNIST dataset contains 62000 randomly rotated handwritten digits.

Network	Test Error (%)
Larochelle et al. (2007)	10.38 ± 0.27
Sohn & Lee (2012)	4.2
Schmidt & Roth (2012)	3.98
Z2CNN	5.03 ± 0.0020
P4CNNRotationPooling	3.21 ± 0.0012
P4CNN	**2.28 ± 0.0004**

Table 1. Error rates on rotated MNIST (with standard deviation under variation of the random seed).

Z2CNN: Normal CNN

P4CNNRotation Pooling: P4CNN but impose rotation invariance in every layer

P4CNN: only rotation invariance for the output layer, equivariance for the intermediate layers.

p4: Cyclic rotation group of order 4 (0, 90, 180, 270)
p4m: p4 plus 4 horizontal and vertical flips

As expected, Group conv can ***improve model performance when (global)symmetries exists***.

Results on datasets without rotations: CIFAR10+: moderate data augmentation with horizontal flips and small translations

Network	G	CIFAR10	CIFAR10+	Param.
All-CNN	$\mathbb{Z}^2$	9.44	8.86	1.37M
	$p4$	8.84	7.67	1.37M
	$p4m$	7.59	7.04	1.22M
ResNet44	$\mathbb{Z}^2$	9.45	5.61	2.64M
	$p4m$	**6.46**	**4.94**	2.62M

Table 2. Comparison of conventional (i.e. $\mathbb{Z}^2$), $p4$ and $p4m$ CNNs on CIFAR10 and augmented CIFAR10+. Test set error rates and number of parameters are reported.

The CIFAR dataset is not actually symmetric, since objects typically appear upright. Nevertheless, we see substantial increases in accuracy on this dataset, indicating that there need not be a full symmetry for G-convolutions to be beneficial.
In the absence of global symmetries, Group Conv can still improve the performance due to its ability to capture local symmetries.

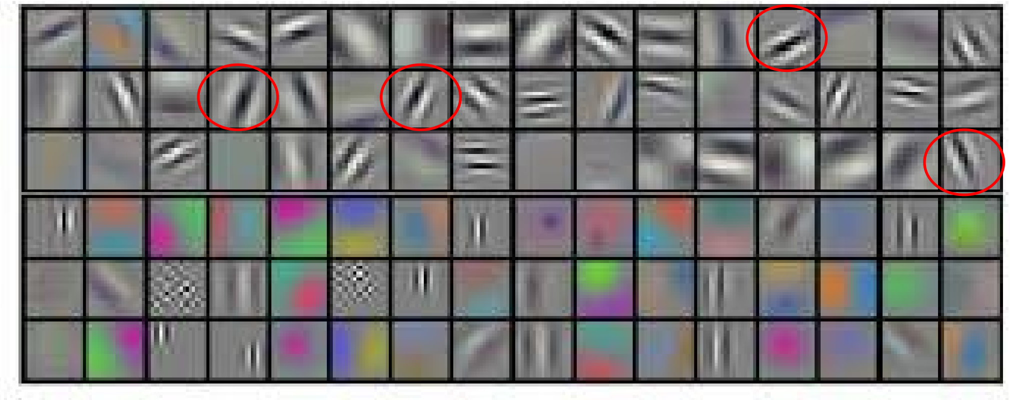
Group Equivariant Convolutional Networks; Taco S. Cohen, Max Welling ICML 2016 (https://arxiv.org/pdf/1602.07576.pdf)

Regular Group CNN: Intuition for Benefits and Advantages

The benefits of having equivariant NN architecture can be summarized as follows:

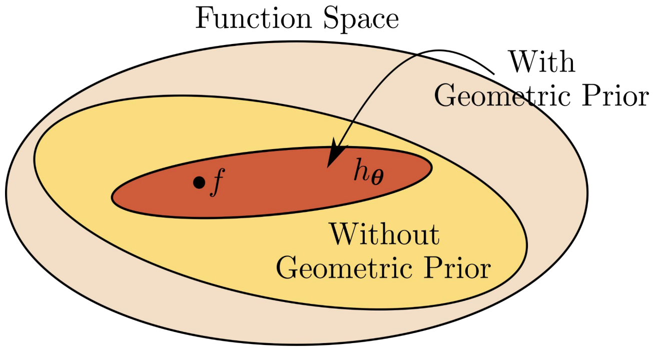
- **Equivariance**: We have geometric guarantee that the model is equivariant to certain symmetry groups.

- **Richer Feature Representations**:



Normal CNN kernel learns roto-translated features, but they are inherent in Group CNN.

- **Generalization and Efficient Learning:** Geometric priors constrain the parameter search space to smaller region → less parameters, better generalization with less data.



Conclusion

In this talk, we covered

- The issues in data augmentation to attain symmetries and the motivations of having symmetries in the model itself.
- Several basic mathematical concepts needed to understand group equivariance.
- Definitions of convolution and cross-correlation and the intuition why they are equivariant under translations.
- Generalization of the notions of translational equivariance in normal CNNs to building group equivariant CNNs.
- The mathematical formulations of group CNNs and the intuitions behind the mathematics.
- The results in the original group CNN paper.
- The intuition of the benefits of having equivariant models, which can be beyond simply achieving symmetries.