Equivariance and Symmetries in CNNs

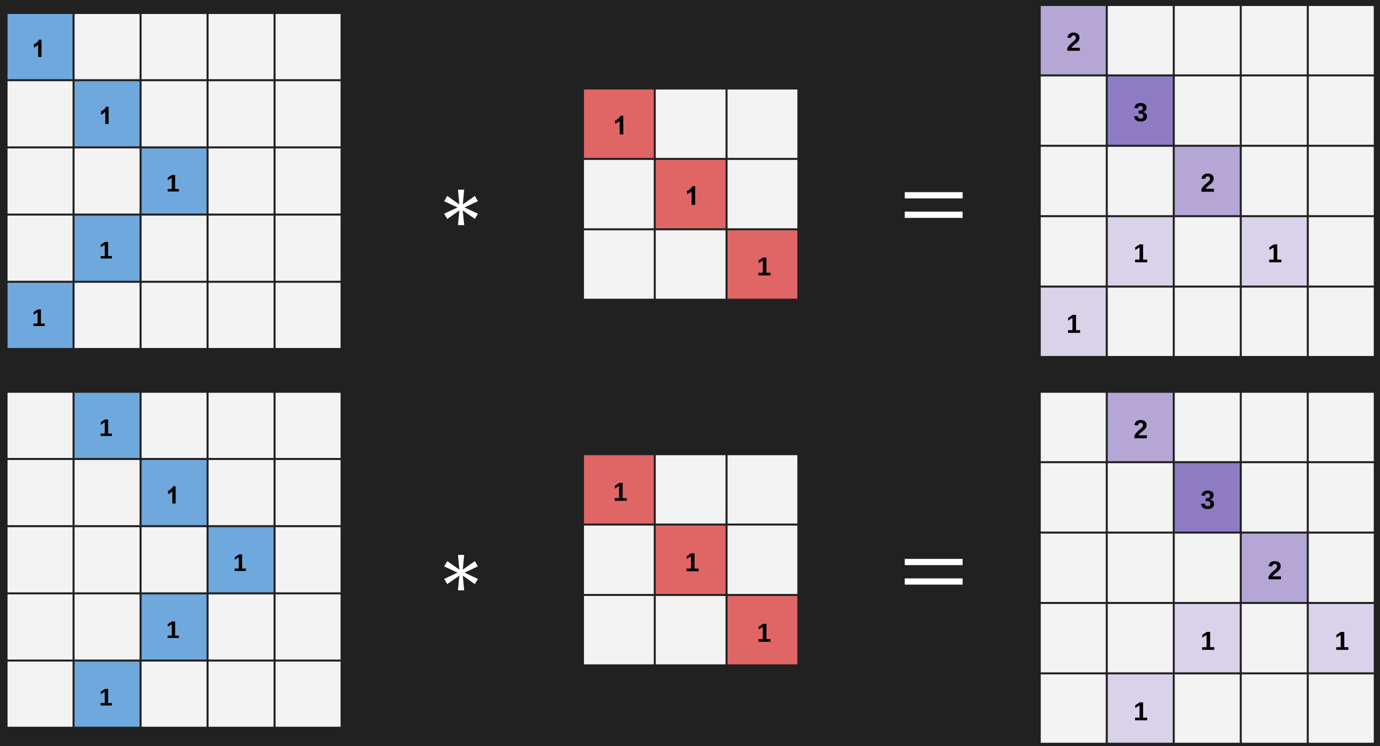
(Stuff that Taco Cohen did)

James Allingham and Omer Sella

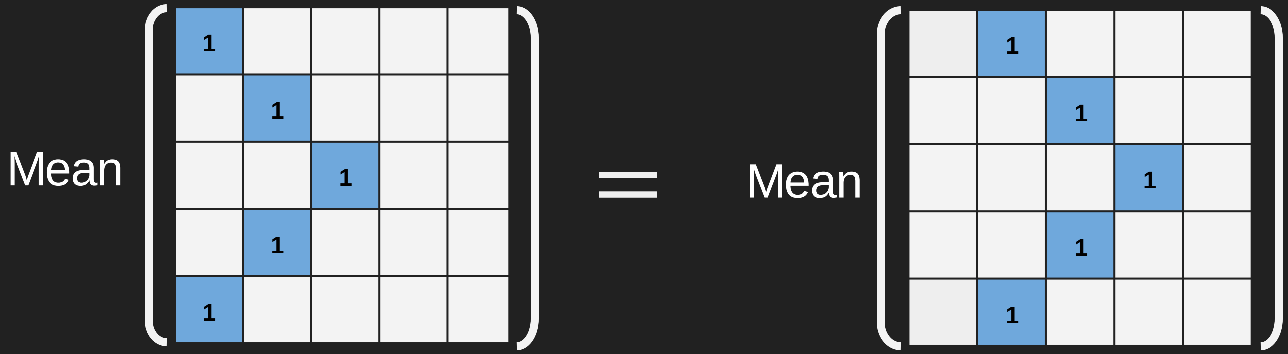
Equivariance

(VS invariance)

Equivariance Visualised



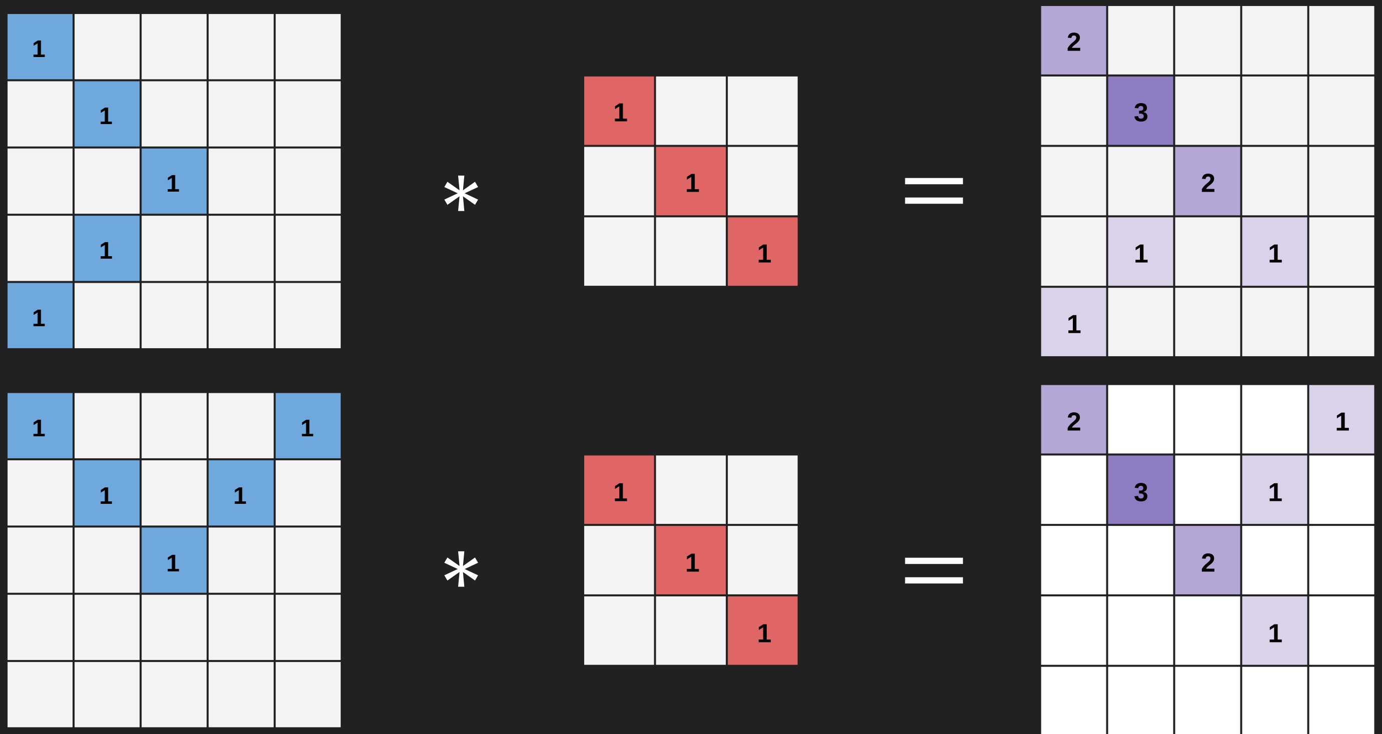
Invariance Visualised



Group Equivariant CNNs[1]

[1] Cohen, Taco S., and Welling, Max. "Group equivariant convolutional networks." *International conference on machine learning*. 2016.

Rotations in CNNs



A *little* bit of group theory!

- A ***symmetry*** of an object is a transformation that leaves the object invariant.
- A ***symmetry group*** is a set of transformations such that for two symmetry transformations **g** and **h**:
  - $g.h$ is also a symmetry.
  - g^{-1} is also a symmetry.
  - $g^{-1}.g$ is the identity transformation e.
- An example is 2D integer translations ($\mathbb{Z}^2$):
  - The group operation (.) is addition (+).
  - $(n, m) + (p, q) = (n + p, m + q)$.
  - This is the group for standard (translation invariant) convolutions!

A couple more groups

The group **p4**:

$$g(r,u,v) = \begin{bmatrix} \cos(\frac{r\pi}{2}) & -\sin(\frac{r\pi}{2}) & u \\ \sin(\frac{r\pi}{2}) & \cos(\frac{r\pi}{2}) & v \\ 0 & 0 & 1 \end{bmatrix}$$

The group **p4m**:

$$g(m,r,u,v) = \begin{bmatrix} (-1)^m\cos(\frac{r\pi}{2}) & -(-1)^m\sin(\frac{r\pi}{2}) & u \\ \sin(\frac{r\pi}{2}) & \cos(\frac{r\pi}{2}) & v \\ 0 & 0 & 1 \end{bmatrix}$$

To act on a pixel (a point in $\mathbb{Z}^2$) with coordinates (p,q) we multiply the matrix g with the coordinate vector $\boldsymbol{x} = (p,q,1)$ of the point: gx.

What is an image? What is a filter?

$$f: \mathbb{Z}^2 \to \mathbb{R}^K$$

How do we transform a filter?

$$[L_g f](x) = [f \circ g^{-1}](x) = f(g^{-1}x)$$

$$L_g L_h = L_{hg}$$

For example, if g is a translation by $t = (u, v)$ then we get

$$g^{-1}x = x - t$$

Correlation in CNNs

$$[f \star \psi](x) = \sum_{y \in \mathbb{Z}^2} \sum_{k=1}^{K} f_k(y)\, \psi_k(y - x)$$

$$[[L_t f] \star \psi](x) = [L_t[f \star \psi]](x)$$ 👍

$$[[L_r f] \star \psi](x) = [L_{r^{-1}}[f \star \psi]](x)$$ 👎

Correlation in **G**-CNNs

$$[f \star \psi](g) = \sum_{y \in \mathbb{Z}^2} \sum_{k=1}^{K} f_k(y)\, \psi_k(g^{-1}y)$$

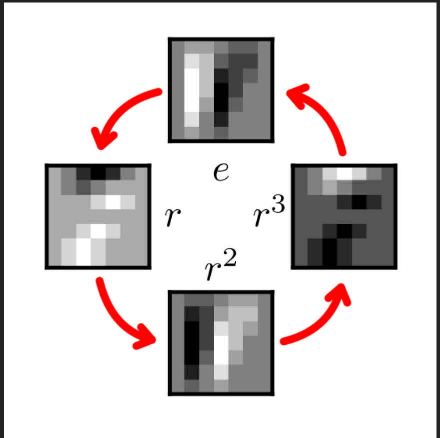
$$f \star \psi : G \to \mathbb{R}^K$$

$$[f \star \psi](g) = \sum_{h \in G} \sum_{k=1}^{K} f_k(h)\, \psi_k(g^{-1}h)$$

$$[L_u f] \star \psi = L_u[f \star \psi] \longleftarrow$$

Practical considerations

- What about biases?
- What about other layers?
  - Pooling
  - Elementwise non-linearities
  - Batch-norm
  - Skip connections
- Efficient implementation (https://github.com/tscohen/GrouPy)



Implementation

$$[f \star \psi](ts) = \sum_{h \in X} \sum_{k=1}^{K} f_k(h) L_t [L_s \psi_k(h)]$$

Standard convolution

Filter transformation

Filter transformation:

$$F = K^l \times K^{l-1} \times S^{l-1} \times n \times n$$

$$F^+ = K^l \times S^l \times K^{l-1} \times S^{l-1} \times n \times n$$

$$F^+[i, s, j, s, u, v] = F[i, j, \overline{s}, \overline{u}, \overline{v}]$$

$$\overline{s}, \overline{u}, \overline{v} = g^{-1}(g(s', 0, 0)^{-1} g(s, u, v))$$

Implementation

Standard convolution

$$[f \star \psi](ts) = \sum_{h \in X} \sum_{k=1}^{K} f_k(h) L_t [L_s \psi_k(h)]$$

Filter transformation

Standard convolution:

$$K^l \times S^l \times K^{l-1} \times S^{l-1} \times n \times n \longrightarrow K^l S^l \times K^{l-1} S^{l-1} \times n \times n$$

Results

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).

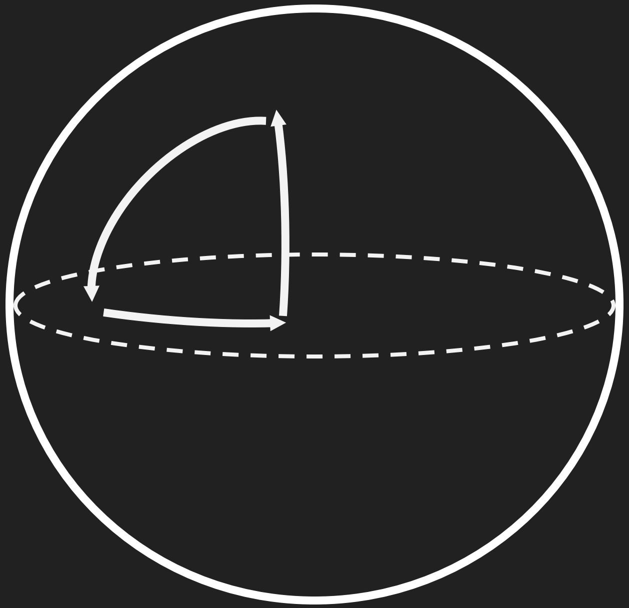
Network	G	CIFAR10	CIFAR10+	Param.
All-CNN	$\mathbb{Z}^2$	9.44	8.86	1.37M
	$p4$	8.84	7.67	1.37M
	$p4m$	7.59	7.04	1.22M
ResNet44	$\mathbb{Z}^2$	9.45	5.61	2.64M
	$p4m$	**6.46**	**4.94**	2.62M

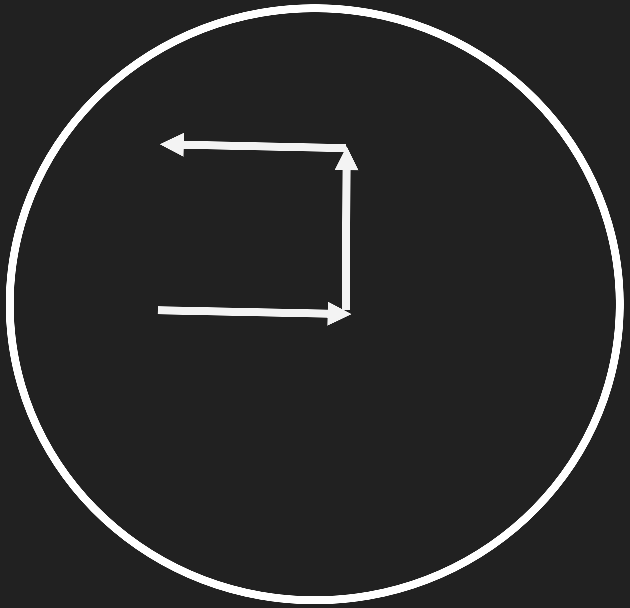
Table 2. Comparison of conventional (i.e. $\mathbb{Z}^2$), $p4$ and $p4m$ CNNs on CIFAR10 and augmented CIFAR10+. Test set error rates and number of parameters are reported.

Spherical CNNs[2]

[2] Cohen, Taco S., et al. "Spherical CNNs." (2018).

Flat earth?



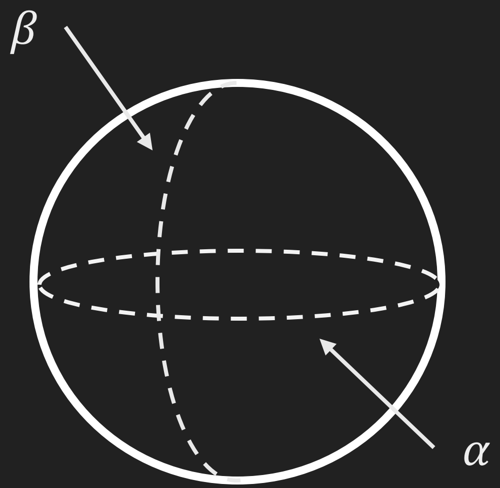


If you still aren’t convinced...

The unit sphere

- The unit sphere (S^2) is the points $\mathbf{x} = (x, y, z)$ such that $\sqrt{x^2 + y^2 + z^2} = 1$.
- Because of the constraint it is possible to parameterize the unit sphere with two angles $\alpha \in [0, 2\pi]$ and $\beta \in [0, \pi]$.
- Then we have:

$$x = \sin\beta \, \cos\alpha$$
$$y = \sin\beta \, \sin\alpha$$
$$z = \cos\beta$$



A spherical image/filter is then a function $f: S^2 \to \mathbb{R}^K$

Rotations in 3D - SO(3)

- SO(3) can be represented by 3×3 matrices R.
- To rotate a point in 3D we simply compute Rx.
- These rotations preserve distance ($\|Rx\| = \|x\|$) and orientation ($|R| = +1$).
- One parameterization of SO(3) is the ZYZ Euler angles: $\alpha \in [0, 2\pi], \beta \in [0, \pi]$, and $\gamma \in [0, 2\pi]$, giving the following rotation matrix:

$$\begin{bmatrix} \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & -\cos\gamma\sin\gamma - \cos\beta\cos\gamma\sin\alpha & \cos\alpha\sin\beta \\ \cos\alpha\sin\gamma + \cos\beta\cos\gamma\sin\alpha & \cos\alpha\cos\gamma - \cos\beta\sin\alpha\sin\gamma & \sin\alpha\sin\beta \\ -\cos\gamma\sin\beta & \sin\beta\sin\gamma & \cos\beta \end{bmatrix}$$

Spherical correlation

$$[f \star \psi](R) = \langle f, L_R \psi \rangle = \int_{S^2} \sum_{k=1}^{K} f_k(x)\, \psi_k(R^{-1}x) dx$$

$$[L_R f](x) = f(R^{-1}x)$$

$$dx = \frac{d\alpha \sin\beta\, d\beta}{4\pi}$$

$$\int_{S^2} f(Rx) dx = \int_{S^2} f(x) dx$$

$$\langle f, L_R \psi \rangle = \langle L_{R^{-1}} f, \psi \rangle$$

SO(3) correlation

$$[f \star \psi](R) = \langle f, L_R \psi \rangle = \int_{SO(3)} \sum_{k=1}^{K} f_k(Q)\, \psi_k(R^{-1}Q) dQ$$

$$[L_R f](Q) = f(R^{-1}Q)$$

$$dQ = \frac{d\alpha \sin\beta d\beta d\gamma}{8\,\pi^2}$$

$$\langle f, L_R \psi \rangle = \langle L_{R^{-1}} f, \psi \rangle$$

$$\big[[L_Q f] \star \psi\big](R) = \big\langle L_Q f, L_R \psi \big\rangle = \big\langle f, L_{Q^{-1}R} \psi \big\rangle = [f \star \psi](Q^{-1}R) = [L_Q[f \star \psi]](R)$$

Practical Considerations & Implementation

- Theory presented is for continuous data not discrete.
- Implemented using a generalized FFT for spherical and SO(3) signals.

$$\mathcal{F}(f \star \psi) = \mathcal{F}(f)\mathcal{F}(\psi)$$

- https://github.com/jonas-koehler/s2cnn

Results

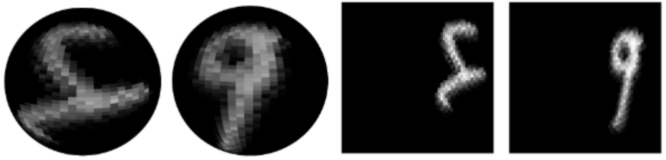
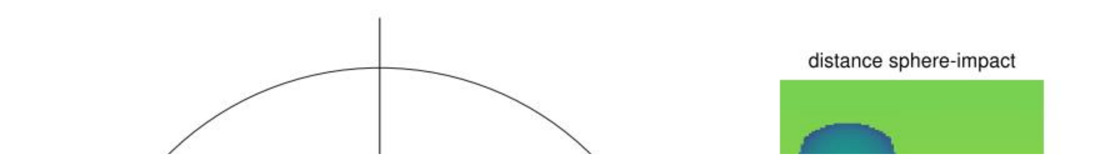


Figure 4: Two MNIST digits projected onto the sphere using stereographic projection. Mapping back to the plane results in non-linear distortions.

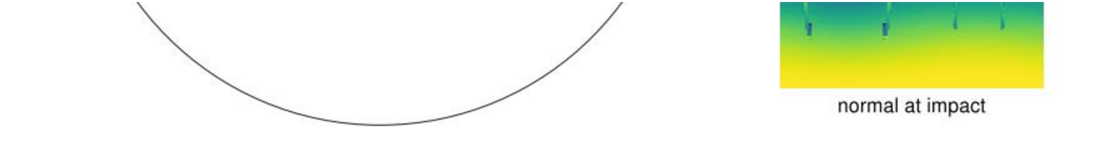
	NR / NR	R / R	NR / R
planar	0.98	0.23	0.11
spherical	0.96	0.95	0.94

Results



Method	P@N	R@N	F1@N	mAP	NDCG
Tatsuma_ReVGG	0.705	0.769	0.719	0.696	0.783
Furuya_DLAN	0.814	0.683	0.706	0.656	0.754
SHREC16-Bai_GIFT	0.678	0.667	0.661	0.607	0.735
Deng_CM-VGG5-6DB	0.412	0.706	0.472	0.524	0.624
Ours	0.701 (3rd)	0.711 (2nd)	0.699 (3rd)	0.676 (2nd)	0.756 (2nd)

Table 2: Results and best competing methods for the SHREC17 competition.

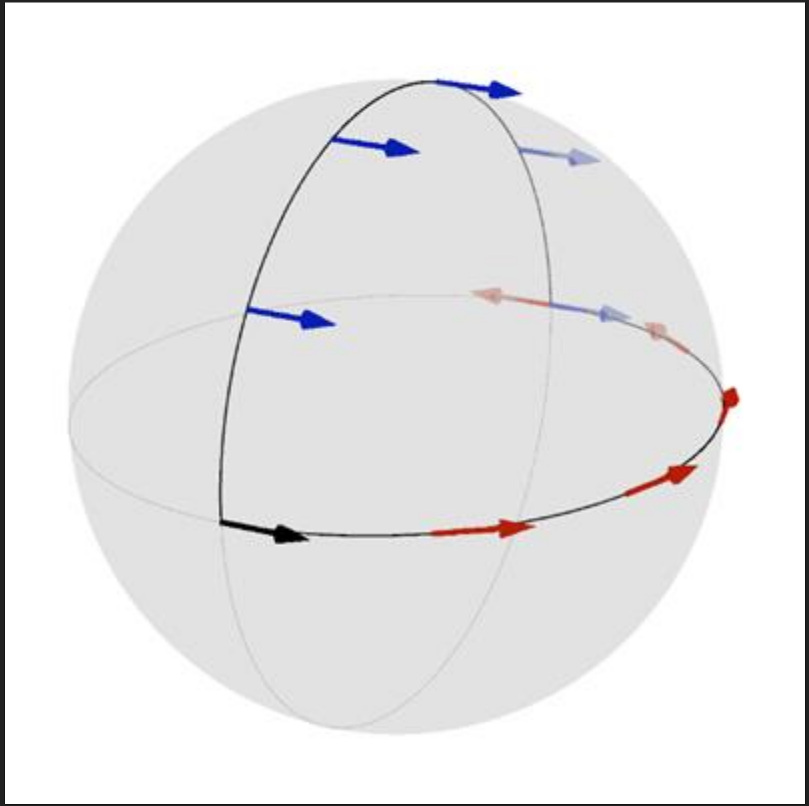


Gauge Equivariant CNNs

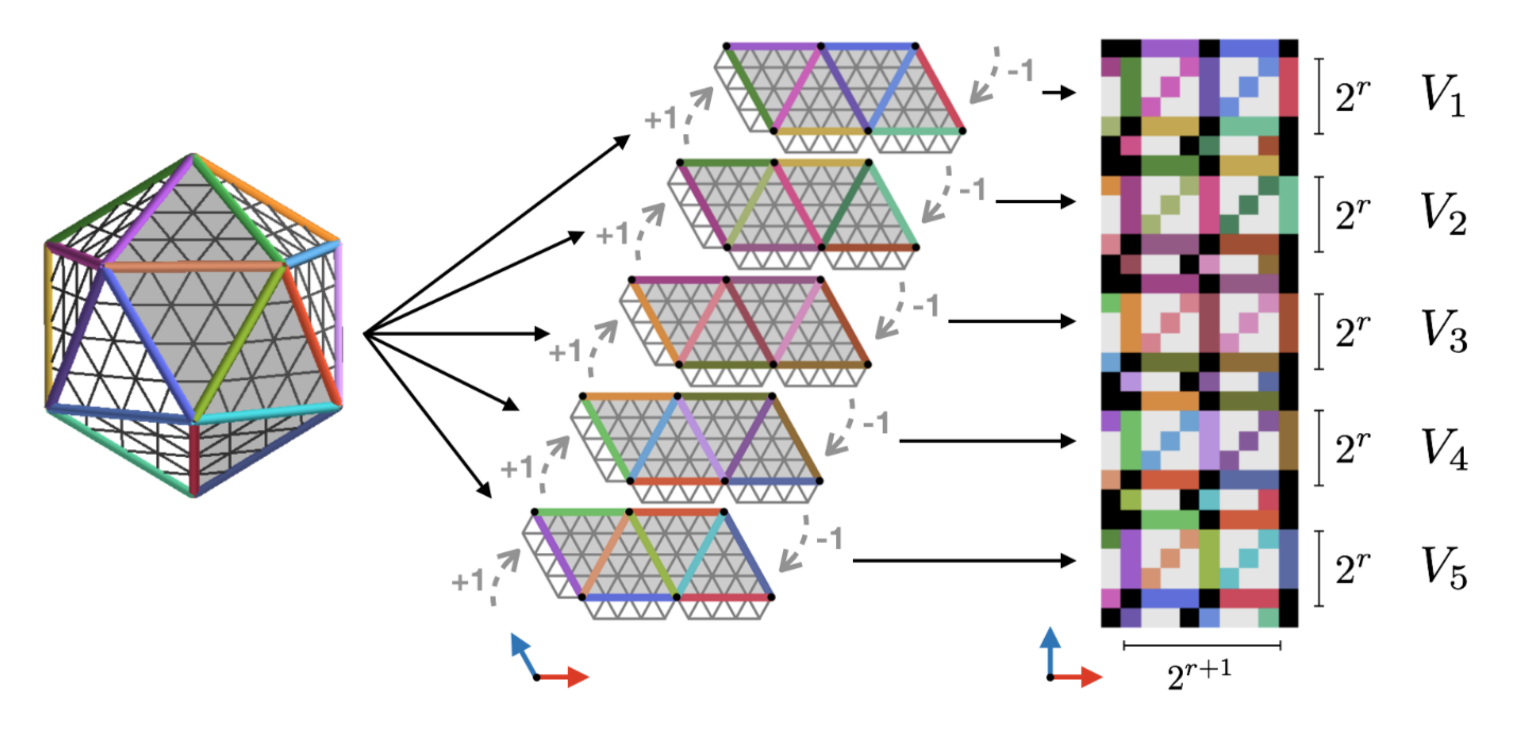
and the Icosahedral CNN[3]

[3] Cohen, Taco S., et al. "Gauge equivariant convolutional networks and the icosahedral cnn." *arXiv preprint arXiv:1902.04615* (2019).

Time for some Gauge Theory!

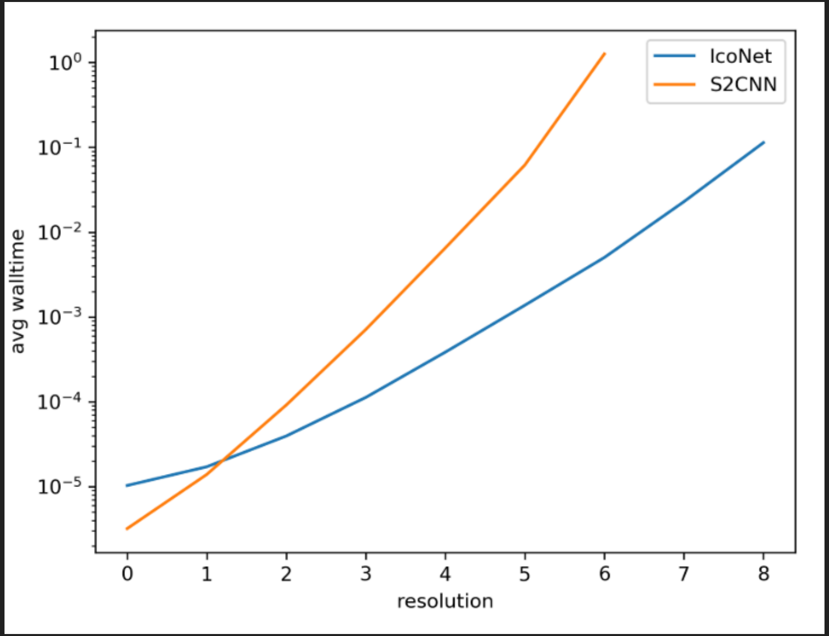


Icosahedral CNN



Implementation

GConv(f,w) = conv2d(GPad(f),expand(w))



Results

Arch.	N/N	N/I	N/R	I/ I	I / R	R / R
S2CNN	99.38	99.38	99.38	99.12	99.13	99.12
NP+NE	99.29	25.50	16.20	98.52	47.77	94.19
NE	99.42	25.41	17.85	98.67	60.74	96.83
NP	99.27	36.76	21.4	98.99	61.62	97.87
S2S	97.81	97.81	55.64	97.72	58.37	89.92
S2R	98.99	98.99	59.76	98.62	55.57	98.74
R2R	**99.43**	**99.43**	**69.99**	**99.38**	**66.26**	**99.31**

Key takeaways

- If you believe your predictions should be equivariant to some symmetries in the data you need to build it into your model!
- For rotations and flips on the plane Taco Cohen has some fairly easy to use code available so you might as well try it out.
- Similarly for rotations on the sphere.

References

[1] Cohen, Taco S., and Welling, Max. "Group equivariant convolutional networks." *International conference on machine learning*. 2016.

[2] Cohen, Taco S., et al. "Spherical CNNs." (2018).

[3] Cohen, Taco S., et al. "Gauge equivariant convolutional networks and the icosahedral cnn." *arXiv preprint arXiv:1902.04615* (2019).