Efficient architectures

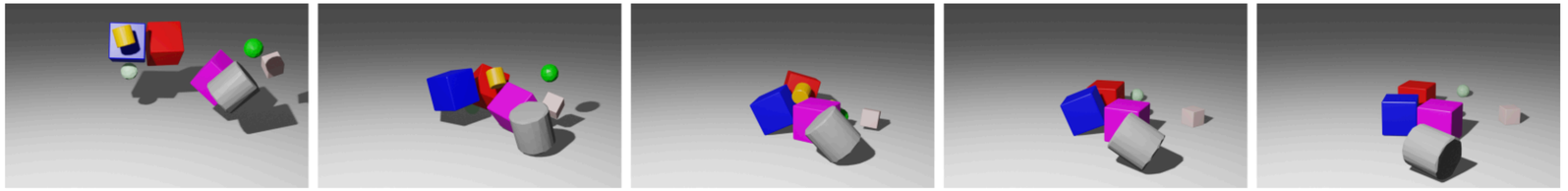
with geometric deep learning

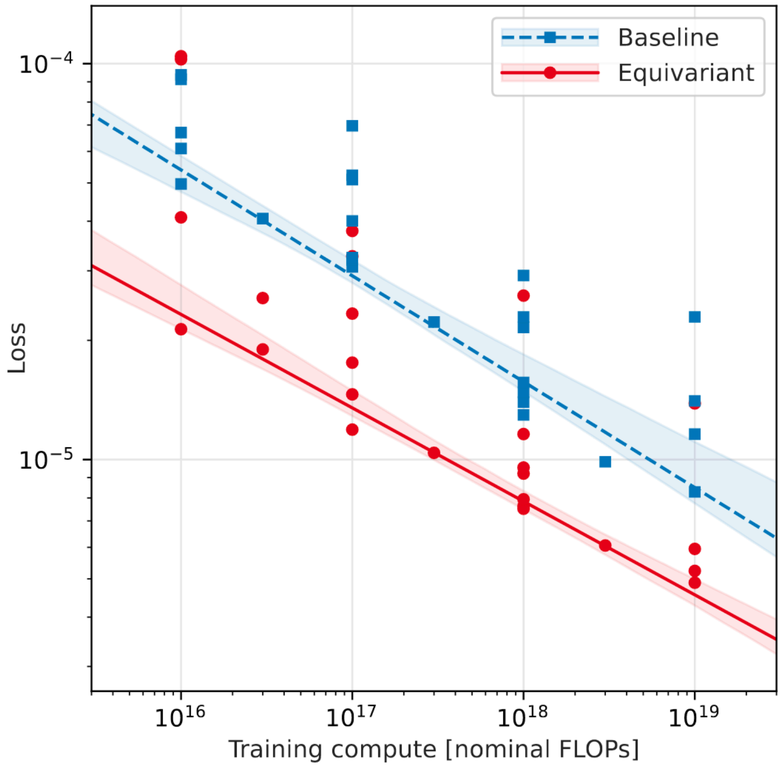
Hampus Linander 2025-04-03 IPA Colloquium 2025 ETH Zurich

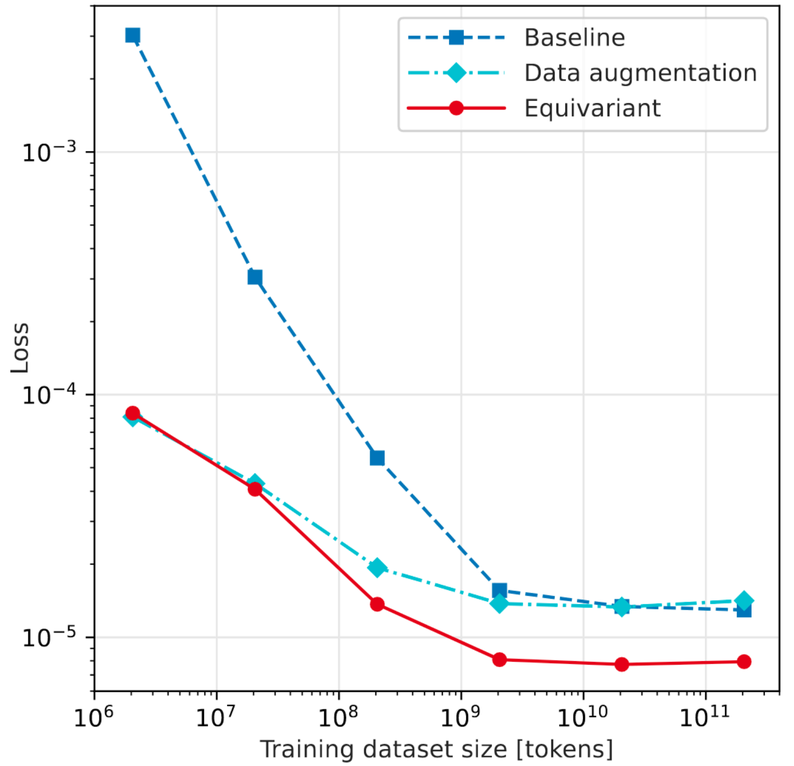
VERSES AI, Chalmers university of technology, University of Gothenburg

Geometric deep learning

A peek at recent results



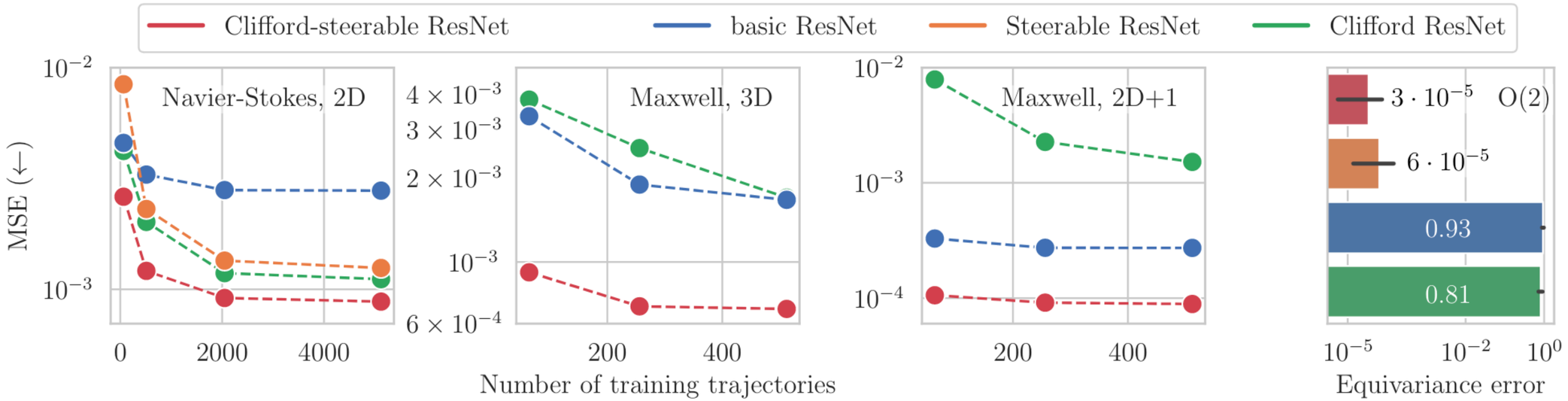


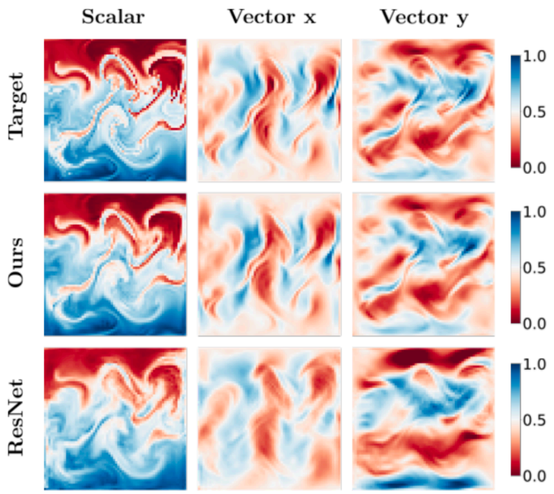


Brehmer, Johann, Sönke Behrends, Pim de Haan, and Taco Cohen. “Does Equivariance Matter at Scale?” 2024, arXiv:2410.23179

Geometric deep learning

A peek at recent results





"In the NavierStokes experiment, [Clifford-steerable ResNet] require only 64 trajectories to outperform the basic ResNet trained on 80× more data."

Zhdanov, Maksim, David Ruhe, Maurice Weiler, Ana Lucic, Johannes Brandstetter, and Patrick Forré. "Clifford-steerable convolutional neural networks." *arXiv:2402.14730*

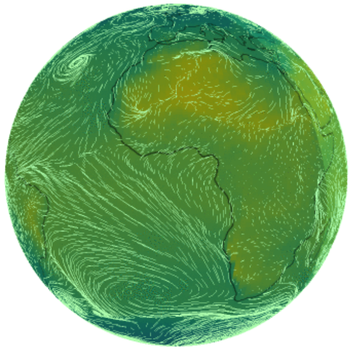
Deep learning

Learning functions

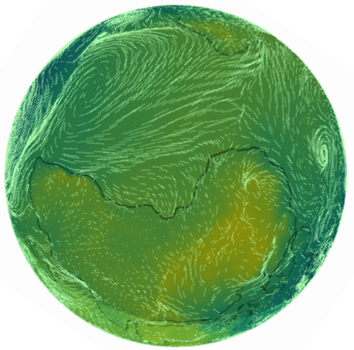
Classify

"cat"

Segment



Forecast



Esteves, Carlos, Jean-Jacques Slotine, and Ameesh Makadia. "Scaling spherical cnns." *arXiv:2306.05420* (2023)

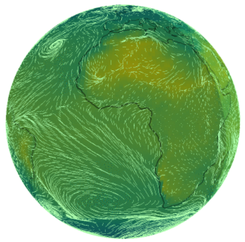
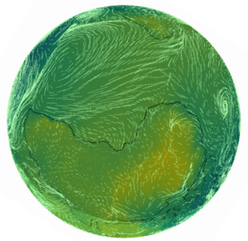
Deep learning

Learning functions

$$X \xrightarrow{f} Y$$

$\mathbb{R}^{W \times H \times 3}$  → "cat" $\{\text{cat}, \text{dog}, \ldots\}$

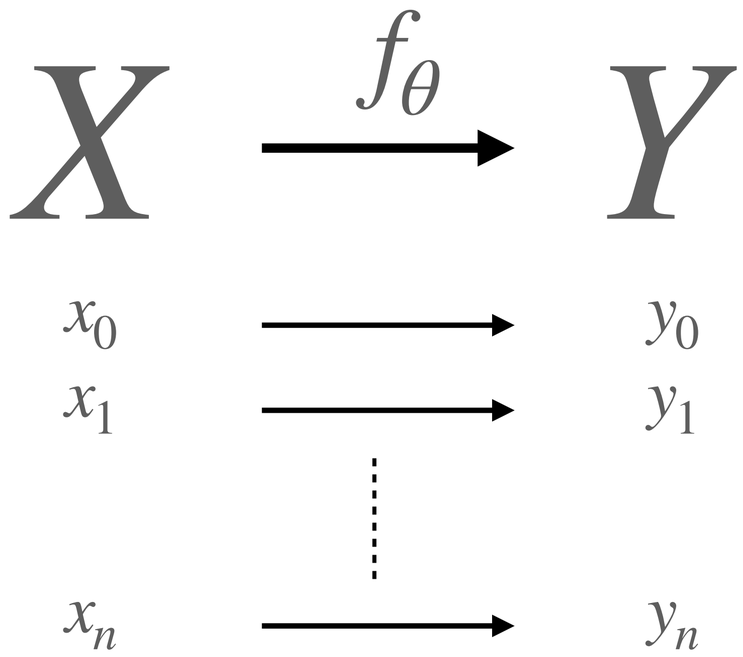
$\mathbb{R}^{W \times H \times 3}$  →  $\mathbb{R}^{W \times H \times N_{\text{classes}}}$

$\mathbb{R}^{Lon \times Lat \times Vars}$  →  $\mathbb{R}^{Lon \times Lat \times Vars}$

Esteves, Carlos, Jean-Jacques Slotine, and Ameesh Makadia. "Scaling spherical cnns." *arXiv:2306.05420* (2023)

Deep learning

Learning functions

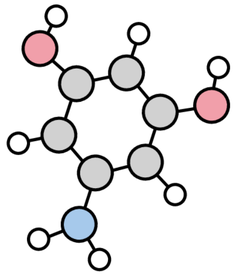


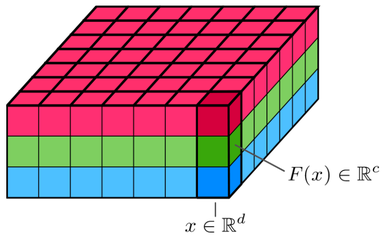
$$\theta^\star = \min_\theta \sum_{i=0}^{n} \text{error}(f_\theta(x_i), y_i)$$

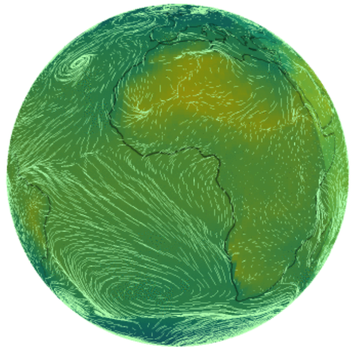
Esteves, Carlos, Jean-Jacques Slotine, and Ameesh Makadia. "Scaling spherical cnns." *arXiv:2306.05420* (2023)

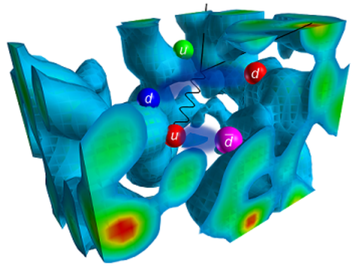
Geometric deep learning

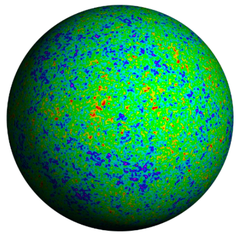
Geometric, topological, and algebraic structure can inform model architecture.











Sanborn, et.al. “Beyond Euclid: An Illustrated Guide to Modern Machine Learning with Geometric, Topological, and Algebraic Structures.”, arXiv 2024
Bronstein, et.al. “Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges.”, arXiv 2021
Weiler, et.al. “Equivariant and coordinate independent convolutional networks”, 2024

Gerken, Aronsson, Carlsson, Linander, Ohlsson, Petersson, Persson. “Geometric Deep Learning and Equivariant Neural Networks.”
Artificial Intelligence Review 2023

Image credits

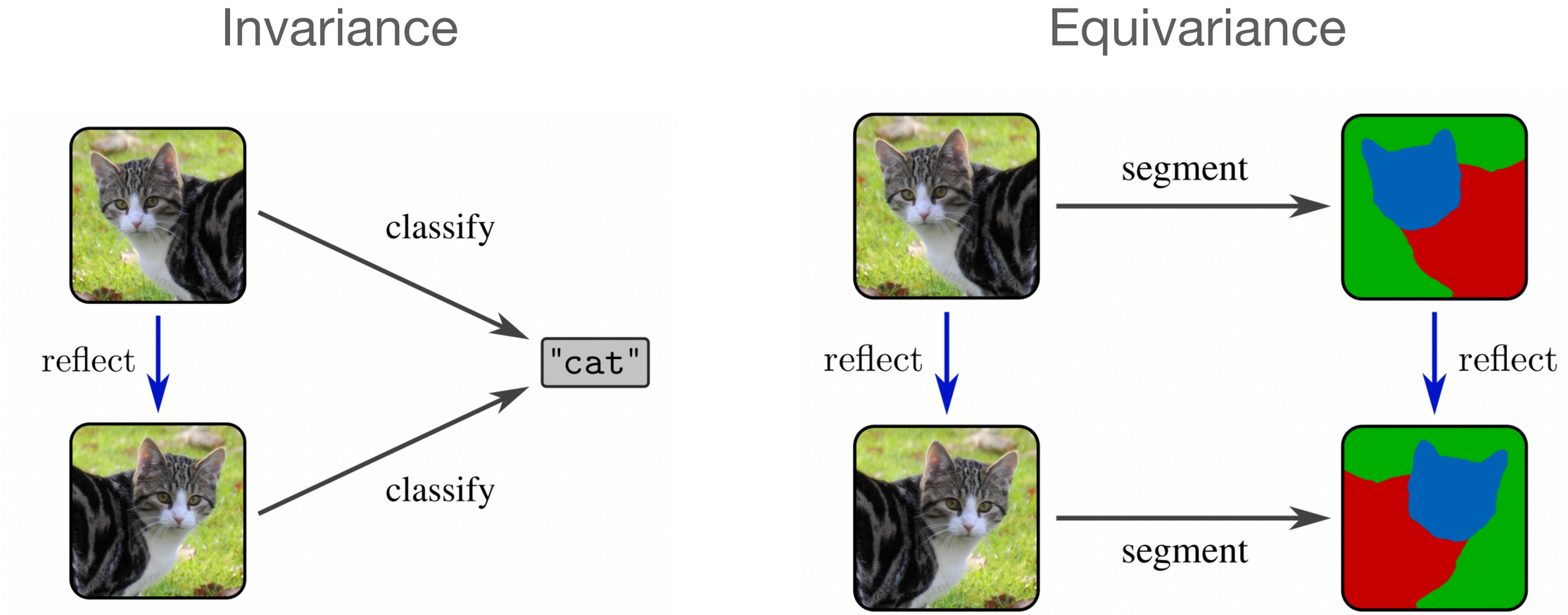
Wang, Ziming, and Rebecka Jörnsten. “SE(3)-Bi-Equivariant Transformers for Point Cloud Assembly.” arXiv 2407.09167.

Hoogeboom, Emiel, Victor Garcia Satorras, Clement Vignac, and Max Welling. “Equivariant Diffusion for Molecule Generation in 3D.” ICML2022

Esteves, Carlos, Jean-Jacques Slotine, and Ameesh Makadia. "Scaling spherical cnns." *arXiv:2306.05420* (2023)

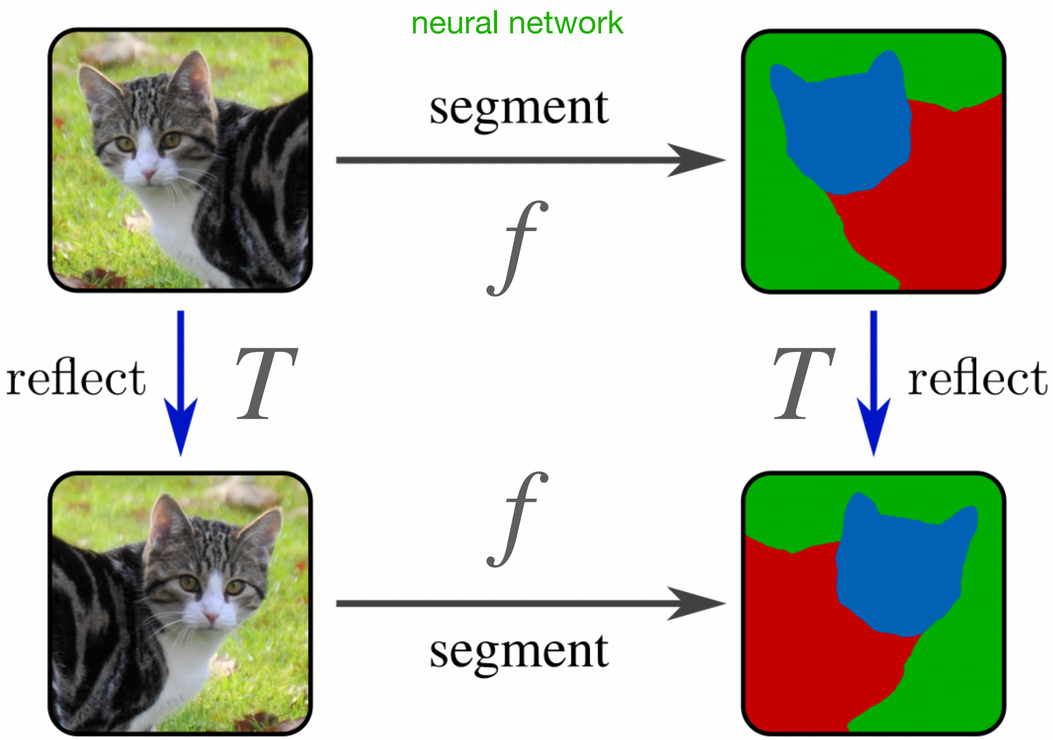
“Equivariant and coordinate independent convolutional networks” M. Weiler 2024, Thesis

Invariance and equivariance



[Equivariant and coordinate independent convolutional networks, M. Weiler 2024]

Equivariance



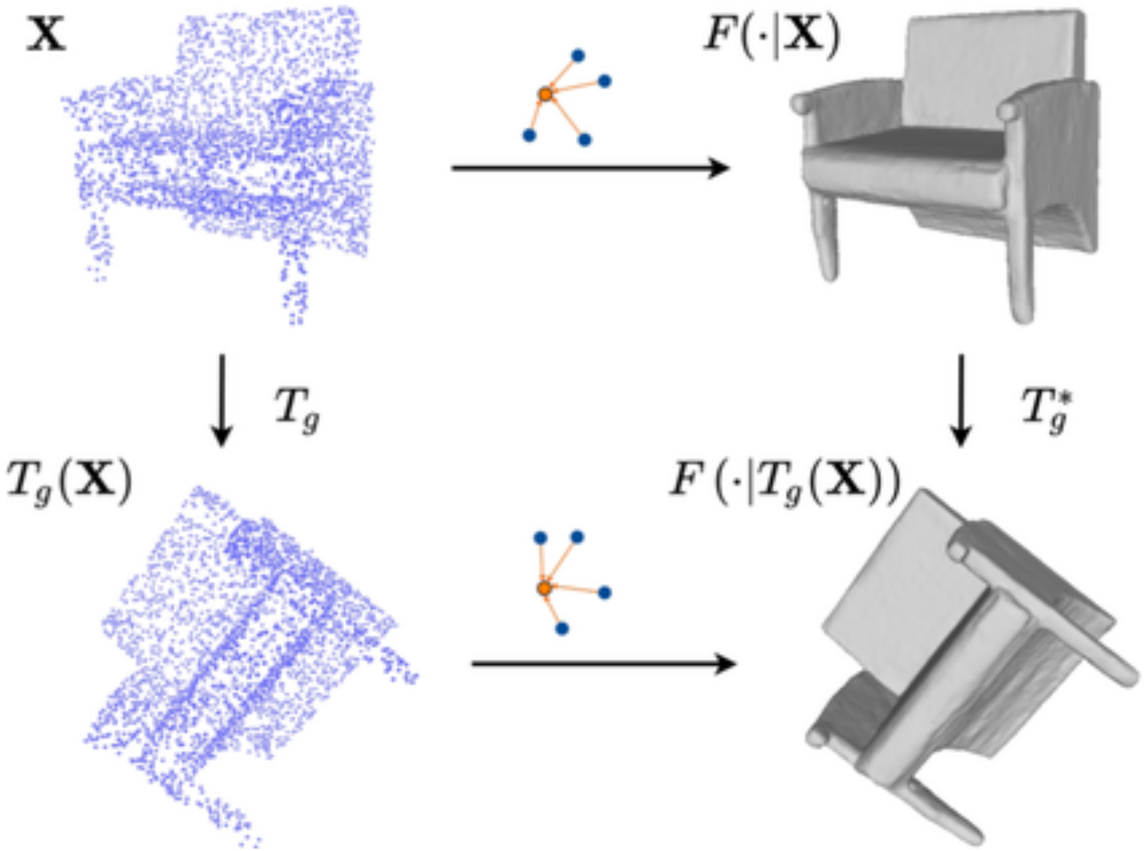
f is equivariant under transformation T if

$$f(Tx) = Tf(x)$$

where T might act in different representations on the input and output space.

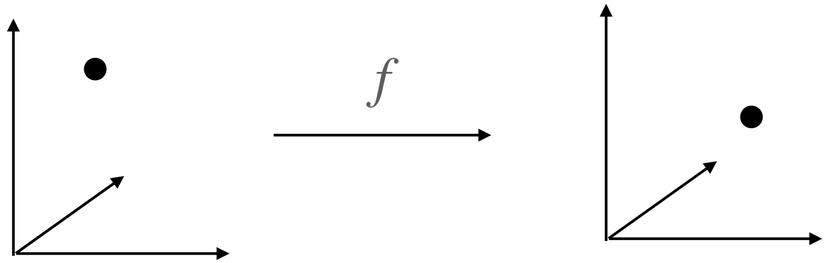
[Equivariant and coordinate independent convolutional networks, M. Weiler 2024]

Equivariance

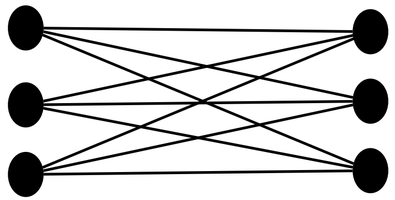


Chen, Yunlu, et al. "3d equivariant graph implicit functions." *ECCV*, 2022

MLP - non-equivariant



$x \in \mathbb{R}^3 \qquad y \in \mathbb{R}^3$



$$y = Wx = W^{ij}x_j\hat{e}_i$$

Linear transform of input

$$M \in \mathbb{R}^{3\times3}$$
$$x = Mx_0$$

How does the output transform?

Could it be that

$$y(Mx_0) \stackrel{?}{=} My(x_0)$$

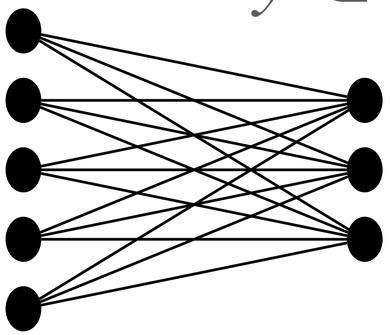
$$y = Wx = WMx_0 \neq MWx_0$$

Vector neuron

Non-equivariant

$x \in \mathbb{R}^m$

$y \in \mathbb{R}^n$



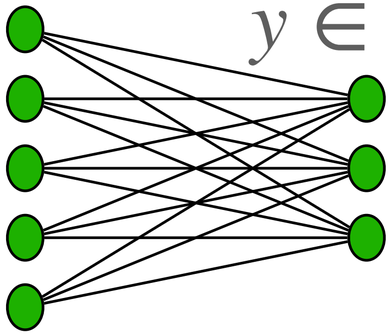
$$y(x) = Wx = W^{ij} x_j \hat{e}_i$$

Equivariant

Promote each neuron to vector

$x \in \mathbb{R}^m \times \mathbb{R}^3$

$y \in \mathbb{R}^n \times \mathbb{R}^3$



Linear transformation W mixes vectors from different neurons, but does not act in R^3

$$y(x) = Wx = W^{ij} x_j^{\alpha} \, \hat{e}_i \, \hat{f}_{\alpha}$$

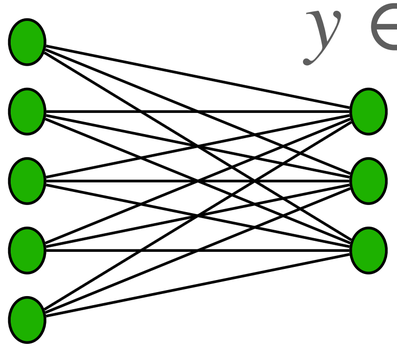
Deng, Congyue, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J. Guibas. "Vector neurons: A general framework for so (3)-equivariant networks.", ICCV 2021

Vector neurons

are equivariant

$$x \in \mathbb{R}^m \times \mathbb{R}^3$$

$$y \in \mathbb{R}^n \times \mathbb{R}^3$$



$$y(x) = Wx = W^{ij} x_j^{\alpha} \, \hat{e}_i \, \hat{f}_{\alpha}$$

$$\begin{aligned} y(Tx) &= WTx \\ &= W^{ij} T^{\alpha\beta} x_{j\beta} \, \hat{e}_i \, \hat{f}_{\alpha} \\ &= TWx \end{aligned}$$

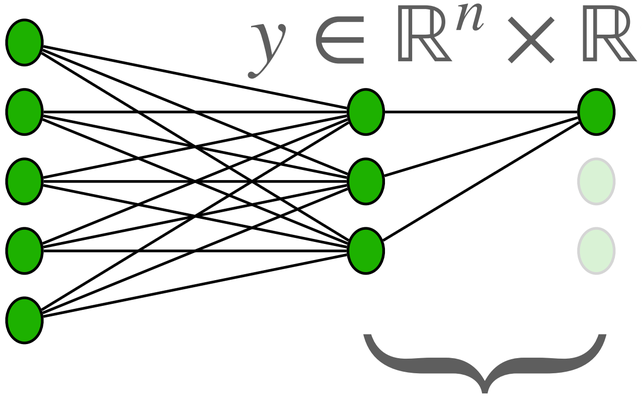
Deng, Congyue, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J. Guibas. "Vector neurons: A general framework for so (3)-equivariant networks.", ICCV 2021

Vector neuron

ReLU

$$x \in \mathbb{R}^m \times \mathbb{R}^3$$

$$y \in \mathbb{R}^n \times \mathbb{R}^3$$

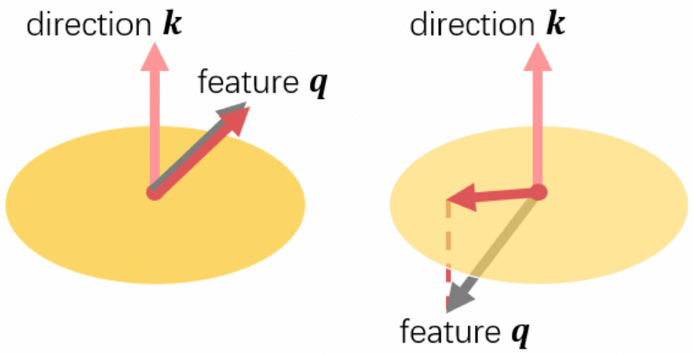


activation

$$q = \tilde{W}y$$

$$k = Uy$$

$$y' = \begin{cases} q & \text{if } \langle q, k \rangle \geq 0 \\ q - \left\langle q, \frac{k}{|k|} \right\rangle \frac{k}{|k|} & \text{otherwise} \end{cases}$$



Deng, Congyue, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J. Guibas. "Vector neurons: A general framework for so (3)-equivariant networks.", ICCV 2021

Vector neuron

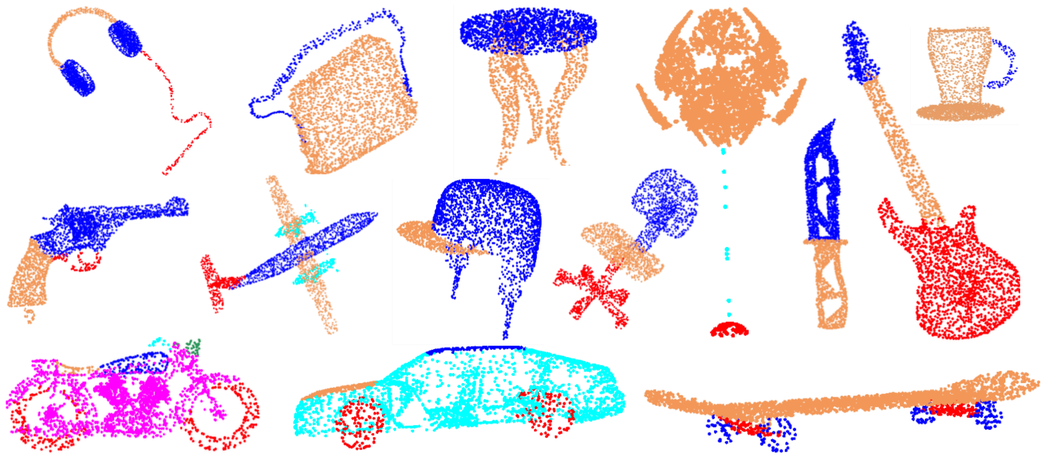
Classification on ModelNet40

Methods	z/z	$z/\mathrm{SO}(3)$	$\mathrm{SO}(3)/\mathrm{SO}(3)$
Point / mesh inputs			
PointNet [25]	85.9	19.6	74.7
DGCNN [35]	90.3	33.8	88.6
VN-PointNet	77.5	77.5	77.2
VN-DGCNN	89.5	**89.5**	**90.2**
PCNN [2]	92.3	11.9	85.1
ShellNet [40]	**93.1**	19.9	87.8
PointNet++ [26]	91.8	28.4	85.0
PointCNN [20]	92.5	41.2	84.5
Spherical-CNN [11]	88.9	76.7	86.9
a^3S-CNN [21]	89.6	87.9	88.7
SFCNN [27]	91.4	84.8	90.1
TFN [32]	88.5	85.3	87.6
RI-Conv [39]	86.5	86.4	86.4
SPHNet [24]	87.7	86.6	87.6
ClusterNet [6]	87.1	87.1	87.1
GC-Conv [41]	89.0	89.1	89.2
RI-Framework [18]	89.4	89.4	89.3

Deng, Congyue, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J. Guibas. "Vector neurons: A general framework for so (3)-equivariant networks.", ICCV 2021

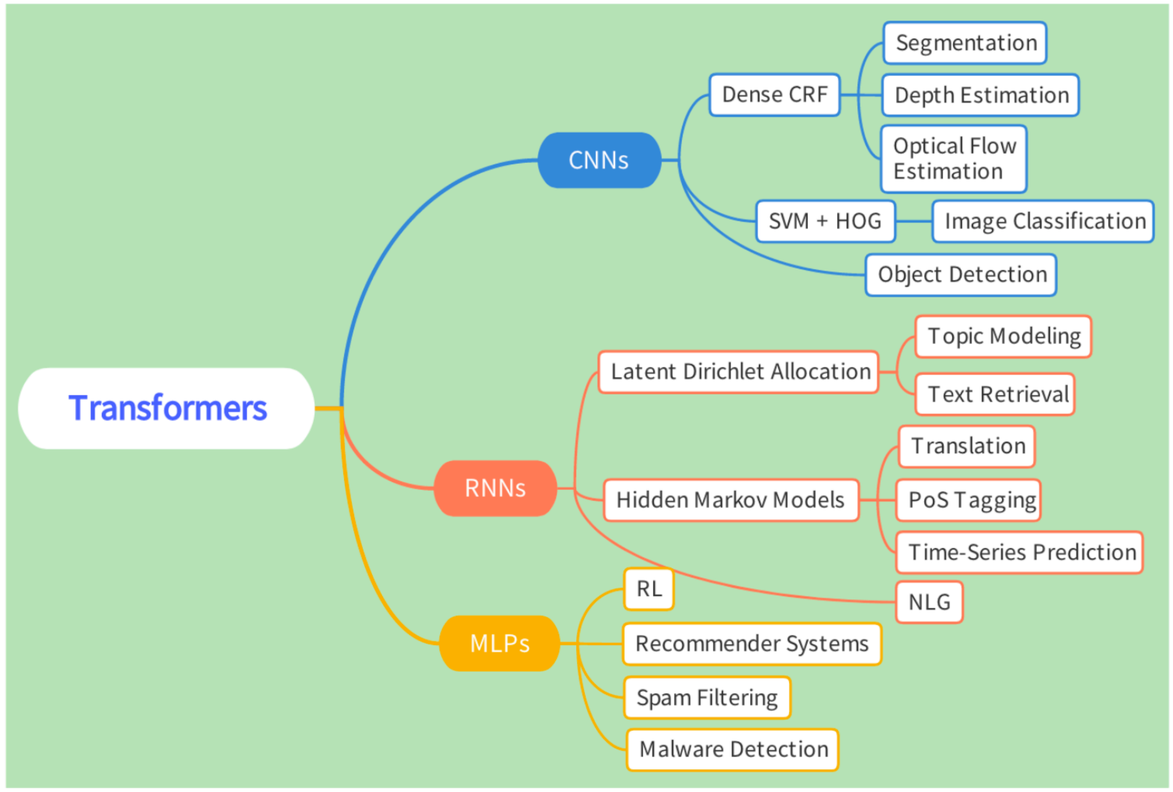
Vector neuron

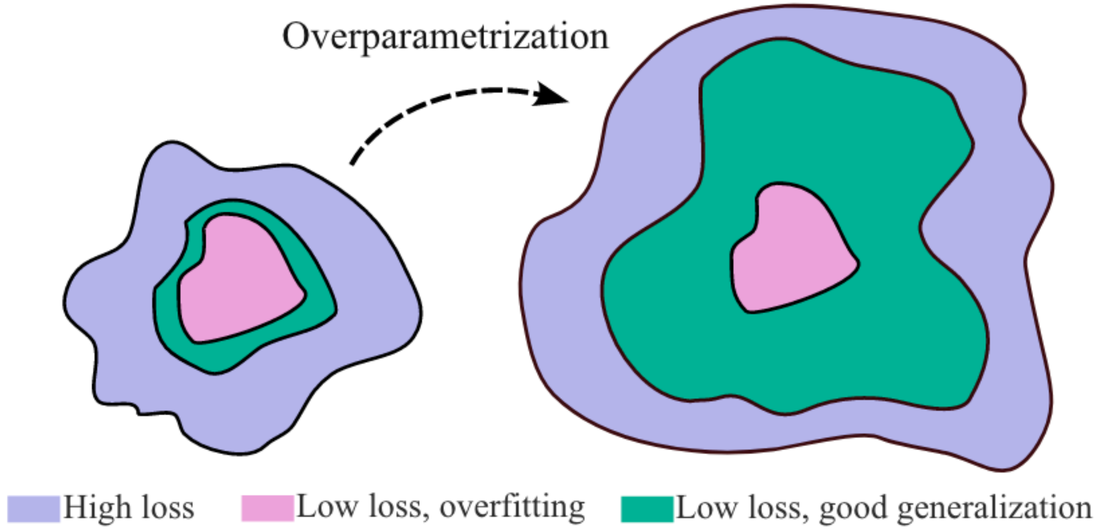
Methods	z/SO(3)	SO(3)/SO(3)
Point / mesh inputs		
PointNet [25]	38.0	62.3
DGCNN [35]	49.3	78.6
VN-PointNet	72.4	72.8
VN-DGCNN	**81.4**	**81.4**
PointCNN [20]	34.7	71.4
PointNet++ [26]	48.3	76.7
ShellNet [40]	47.2	77.1
RI-Conv [39]	75.3	75.3
TFN [32]	76.8	76.2
GC-Conv [41]	77.2	77.3
RI-Framework [18]	79.2	79.4



Deng, Congyue, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J. Guibas. "Vector neurons: A general framework for so (3)-equivariant networks.", ICCV 2021

Architecture convergence





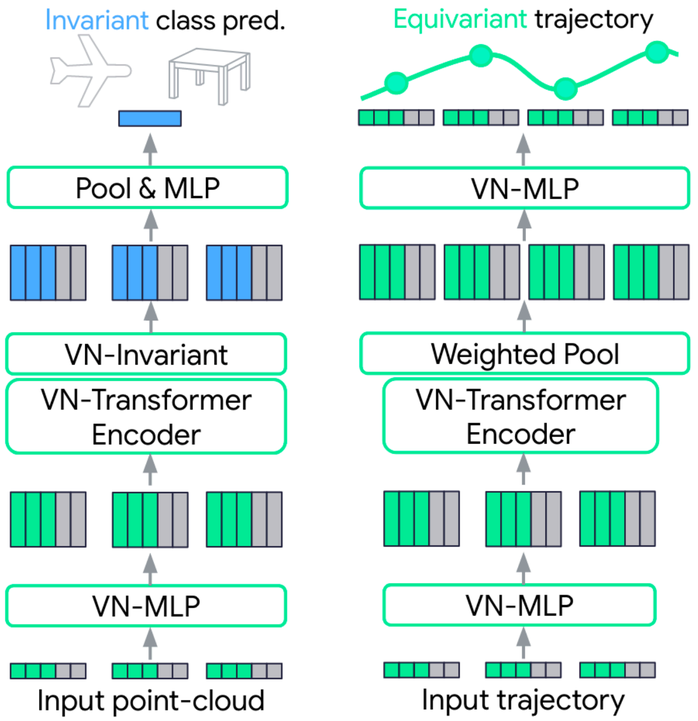
$$\overbrace{R(h)}^{\text{expected risk}} \leq \overbrace{\hat{R}(h)}^{\text{empirical risk}} + \overbrace{\Delta\sqrt{\frac{K(h|A)\log 2 + \log\frac{1}{\delta}}{2n}}}^{\text{compression}}$$

Goldblum et. al., “The No Free Lunch Theorem, Kolmogorov Complexity, and the Role of Inductive Biases in Machine Learning.”, 2024. arXiv.2304.05366.
Wilson, “Deep Learning Is Not So Mysterious or Different.” 2025. arXiv.2503.02113.

Vector neuron

Transformers

VN-Transformer: Rotation-Equivariant Attention for Vector Neurons



(a) Rotation-invariant classification model.

(b) Rotation-equivariant trajectory forecasting model.

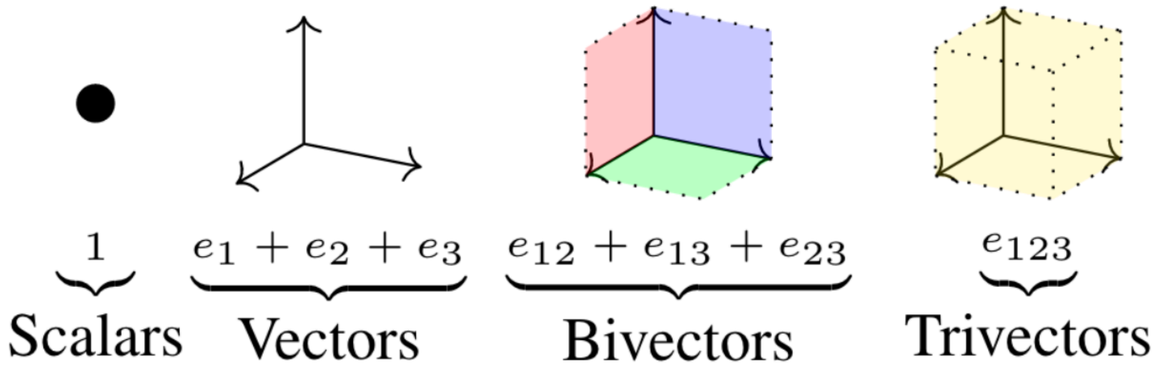
Table 1: ModelNet40 test accuracy. Top block shows SO(3)-invariant baselines taken from Deng et al (2021), included here for convenience.

Model	Acc.	# Params
TFN (Thomas et al., 2018)	88.5%	–
RI-Conv (Zhang et al., 2019)	86.5%	–
GC-Conv (Zhang et al., 2020)	89.0%	–
VN-PointNet (Deng et al., 2021)	77.2%	2.20M
VN-DGCNN (Deng et al., 2021)	90.0%	2.00M
VN-Transformer (ours)	90.8%	0.04M

Assaad, Serge, Carlton Downey, Rami Al-Rfou, Nigamaa Nayakanti, and Ben Sapp. "VN-Transformer: Rotation-Equivariant Attention for Vector Neurons." TMLR 2023

Geometric algebra

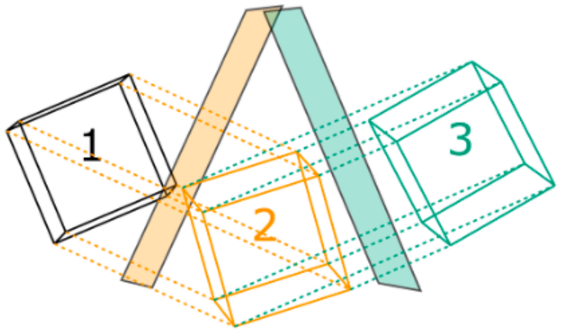
Clifford algebra

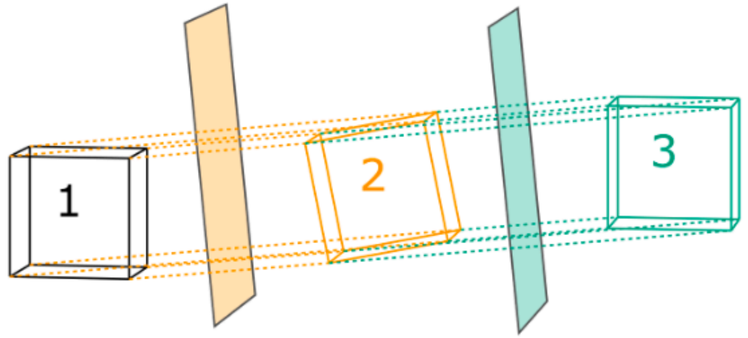


$$e_i e_i \in \{+1, -1, 0\}, \quad e_i e_j = -e_j e_i \quad (i \neq j)$$

$$x = x_s + x_1 e_1 + x_2 e_2 + x_3 e_3 + x_{12} e_1 e_2 + x_{13} e_1 e_3 + x_{23} e_2 e_3 + x_{123} e_1 e_2 e_3$$

$$(x_s, x_1, \ldots, x_{123}) \in \mathbb{R}^8$$



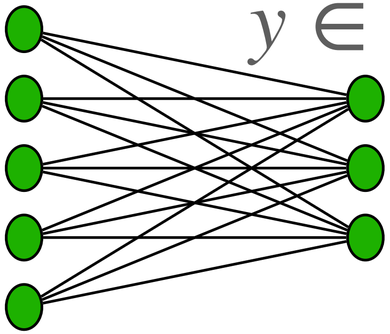


Ruhe, David, Jayesh K. Gupta, Steven De Keninck, Max Welling, and Johannes Brandstetter. "Geometric clifford algebra networks." ICML, 2023.

Geometric algebra MLP

$$x \in \mathbb{R}^m \times \mathbb{R}^8$$

$$y \in \mathbb{R}^n \times \mathbb{R}^8$$



$$y^i(x) = W^{ij} x_j = (W^{ijk}\hat{e}_k)(x_j{}^l \hat{e}_l)$$

$$= W^{ijk} x^k{}_j \hat{e}_k \hat{e}_l$$

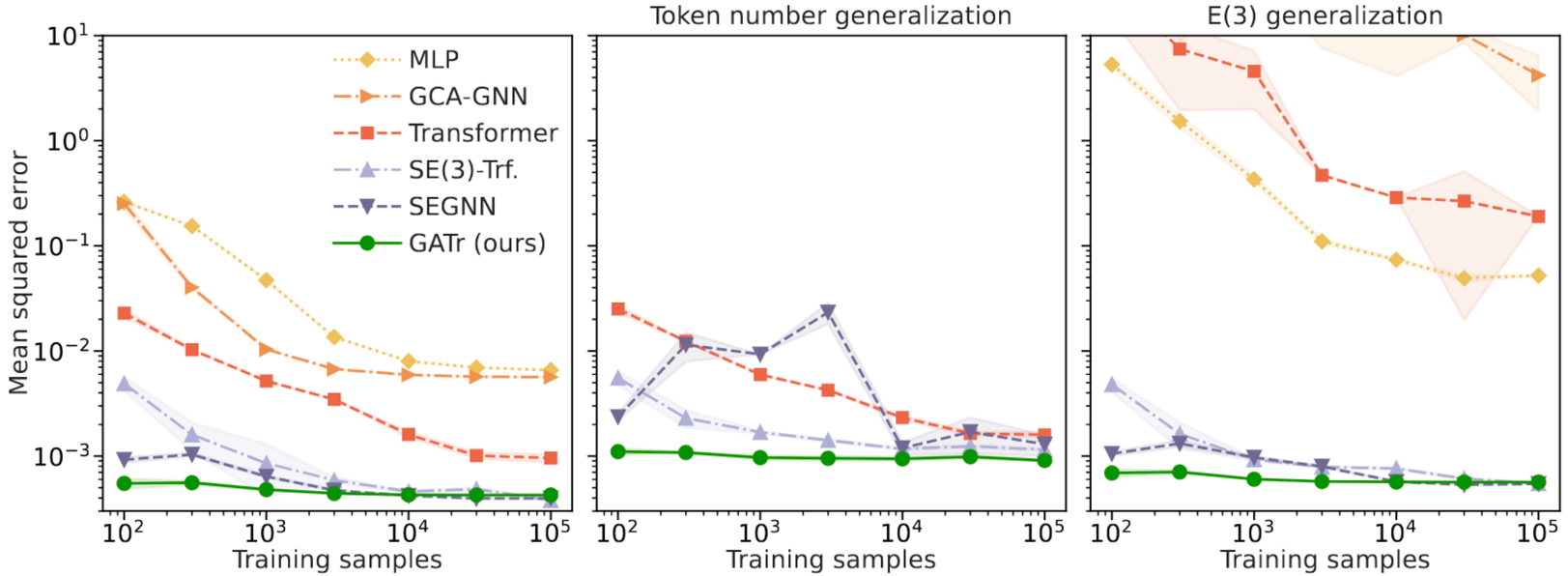
$$= W^{ijk} x^k{}_j f_{klm} \hat{e}^m$$

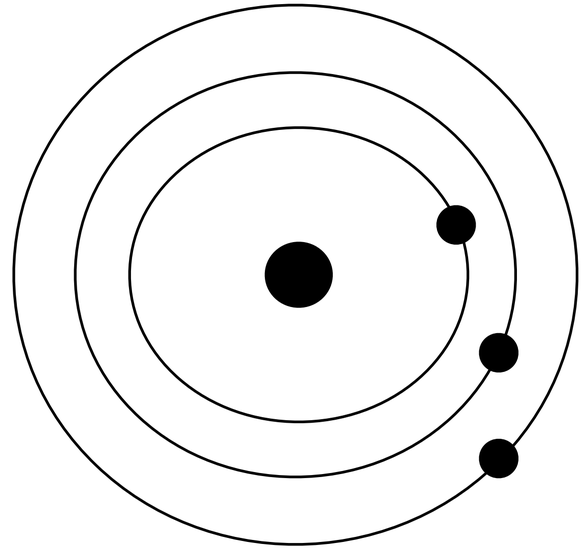
```
                    [m, n, 8]    [8, 8, 8]
einsum("ijk,kj,klm", W, x, f)
                          [m, 8]
```

Geometric algebra transformer

$$\text{Attention}(q, k, v)_{i'c'} = \sum_i \text{Softmax}_i \left(\frac{\sum_c \langle q_{i'c}, k_{ic} \rangle}{\sqrt{8n_c}} \right) v_{ic'}$$

$$\text{GatedGELU}(x) = \text{GELU}(x_1)x$$

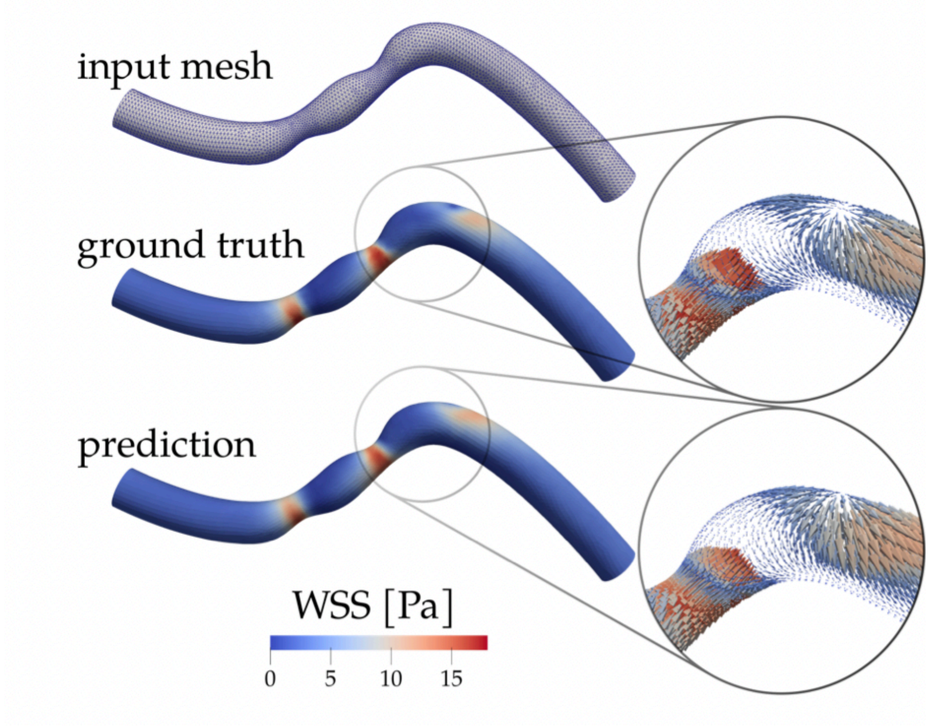


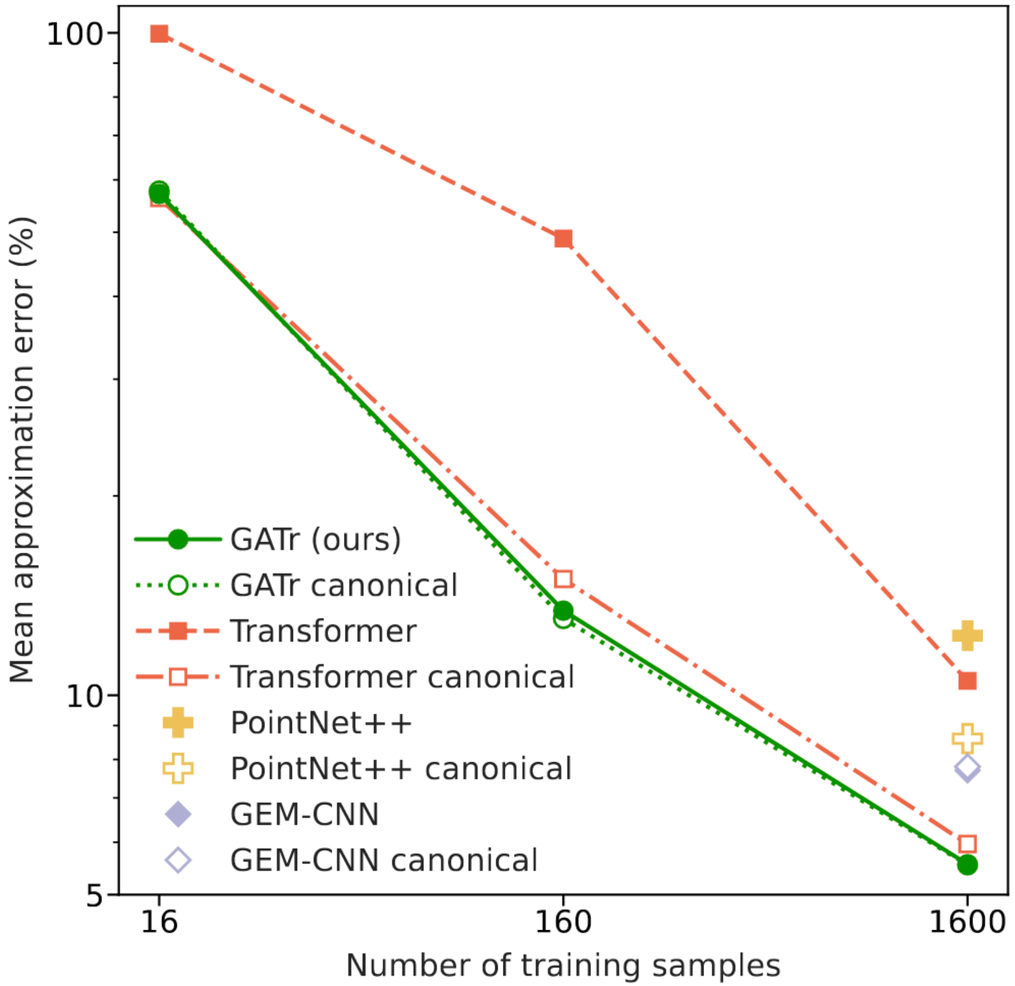


Brehmer, Johann, Pim de Haan, Sönke Behrends, and Taco Cohen. “Geometric Algebra Transformer,” NeurIPS 2023

Geometric algebra transformer

Arterial wall shear stress

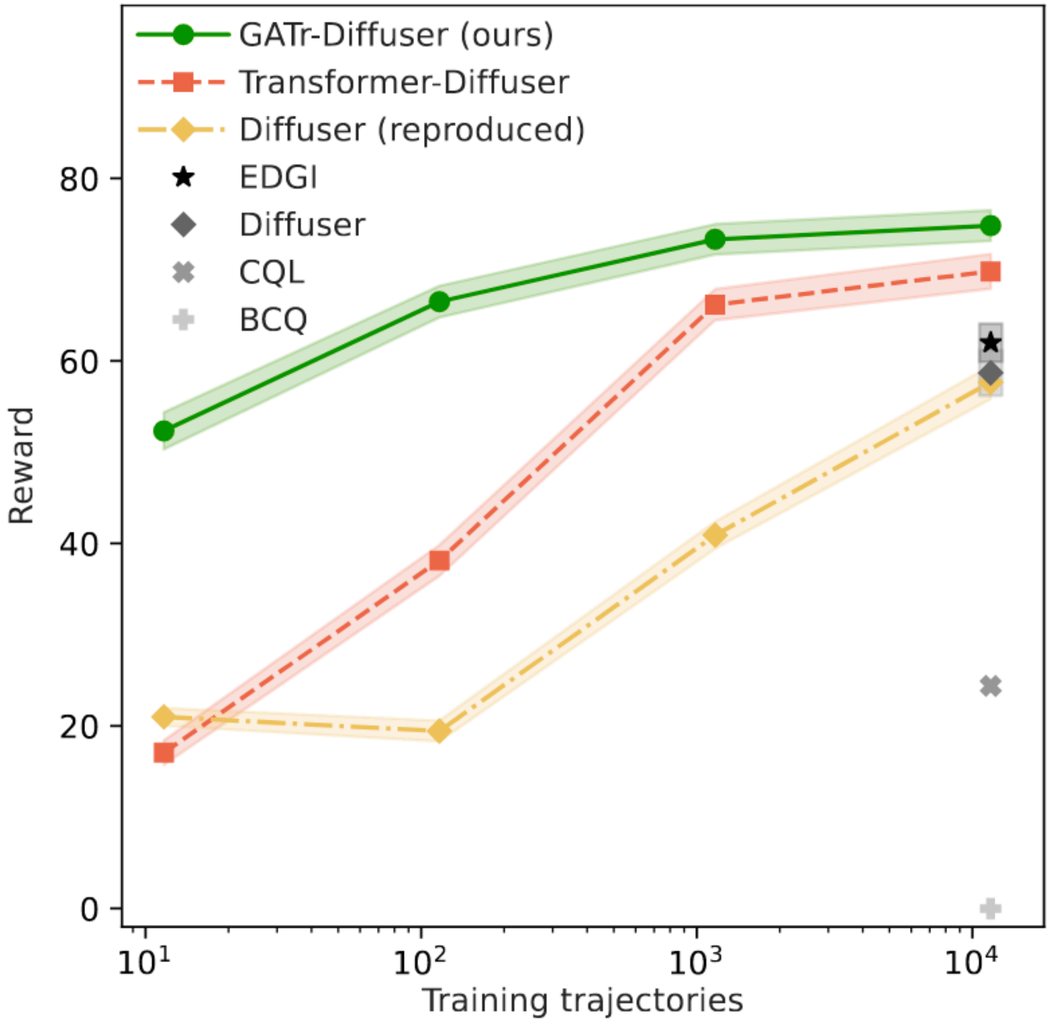


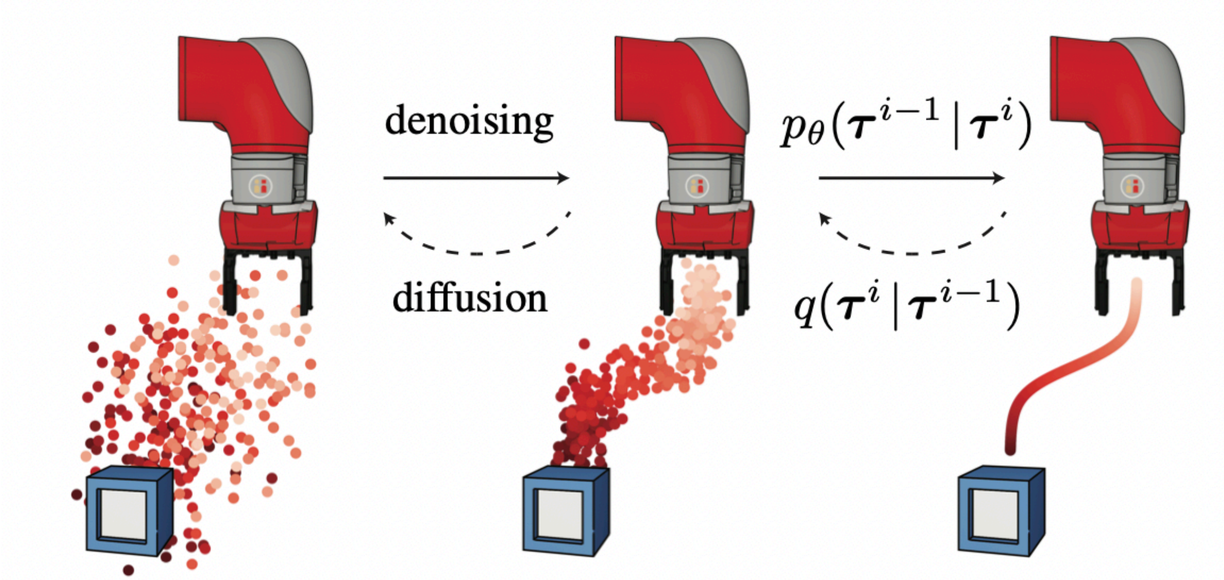


Brehmer, Johann, Pim de Haan, Sönke Behrends, and Taco Cohen. “Geometric Algebra Transformer,” NeurIPS 2023

Suk, Julian, et al. "Mesh neural networks for SE (3)-equivariant hemodynamics estimation on the artery wall." *Computers in Biology and Medicine*

Geometric algebra transformer



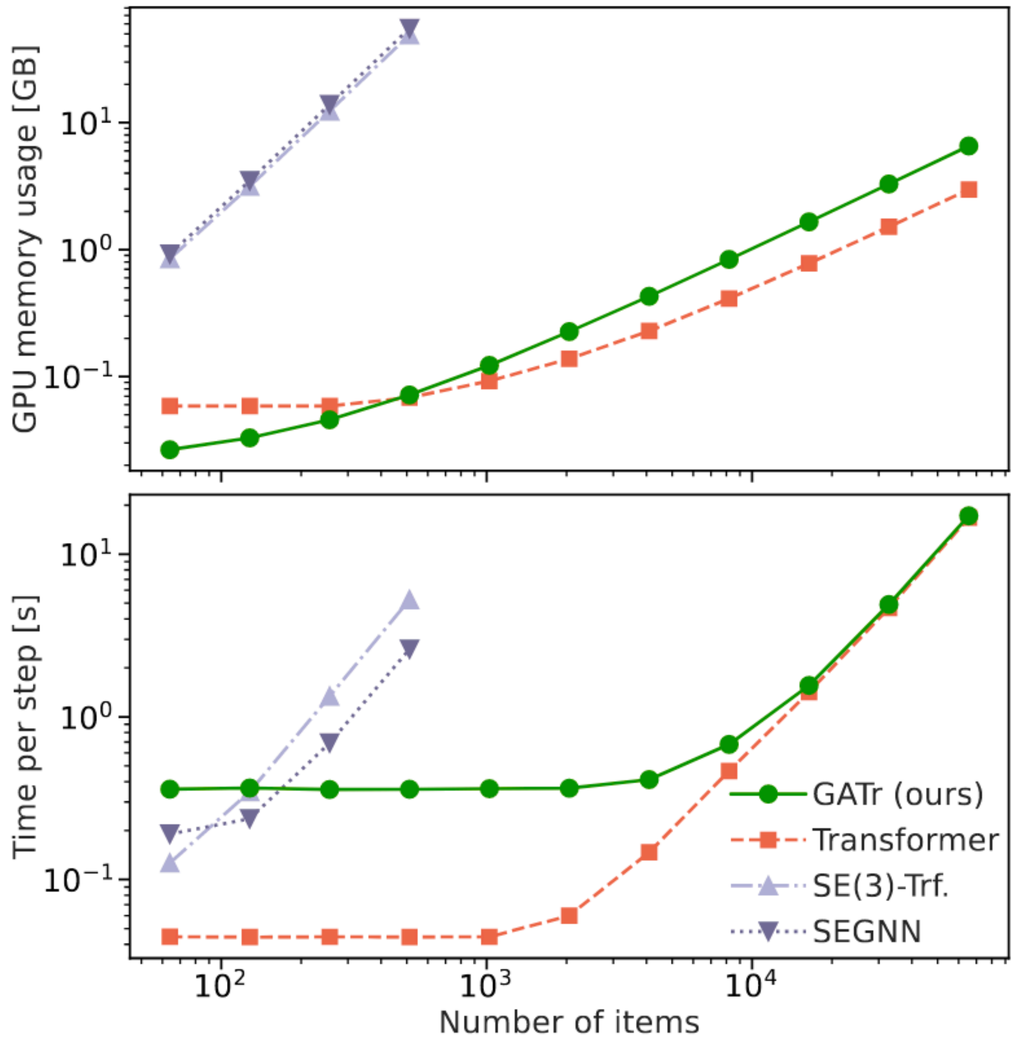


Janner, Michael, et al. "Planning with diffusion for flexible behavior synthesis." *arXiv:2205.09991*

Brehmer, Johann, Pim de Haan, Sönke Behrends, and Taco Cohen. “Geometric Algebra Transformer,” NeurIPS 2023

Geometric algebra transformer

Compute



Brehmer, Johann, Pim de Haan, Sönke Behrends, and Taco Cohen. “Geometric Algebra Transformer,” NeurIPS 2023

Molecule generation

EGNN + Diffusion

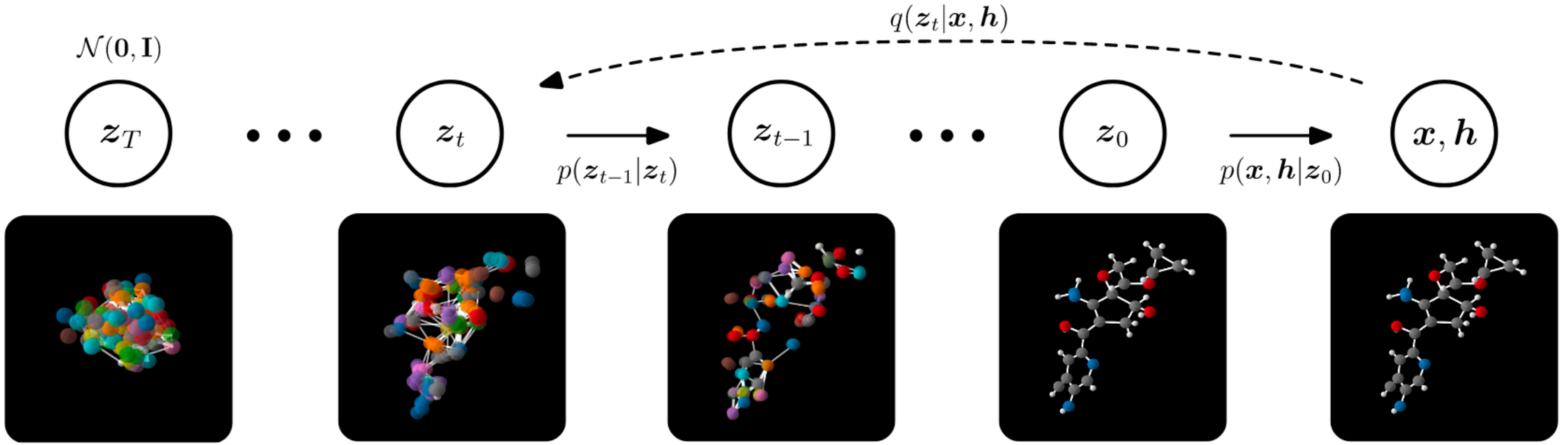


Figure 2. Overview of the Equivariant Diffusion Model. To generate molecules, coordinates $\boldsymbol{x}$ and features $\boldsymbol{h}$ are generated by denoising variables $\boldsymbol{z}_t$ starting from standard normal noise $\boldsymbol{z}_T$. This is achieved by sampling from the distributions $p(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t)$ iteratively. To train the model, noise is added to a datapoint $\boldsymbol{x}, \boldsymbol{h}$ using $q(\boldsymbol{z}_t|\boldsymbol{x},\boldsymbol{h})$ for the step t of interest, which the network then learns to denoise.

Hoogeboom, Emiel, Victor Garcia Satorras, Clement Vignac, and Max Welling. “Equivariant Diffusion for Molecule Generation in 3D.” ICML, 2022

Molecule generation

EGNN + Diffusion

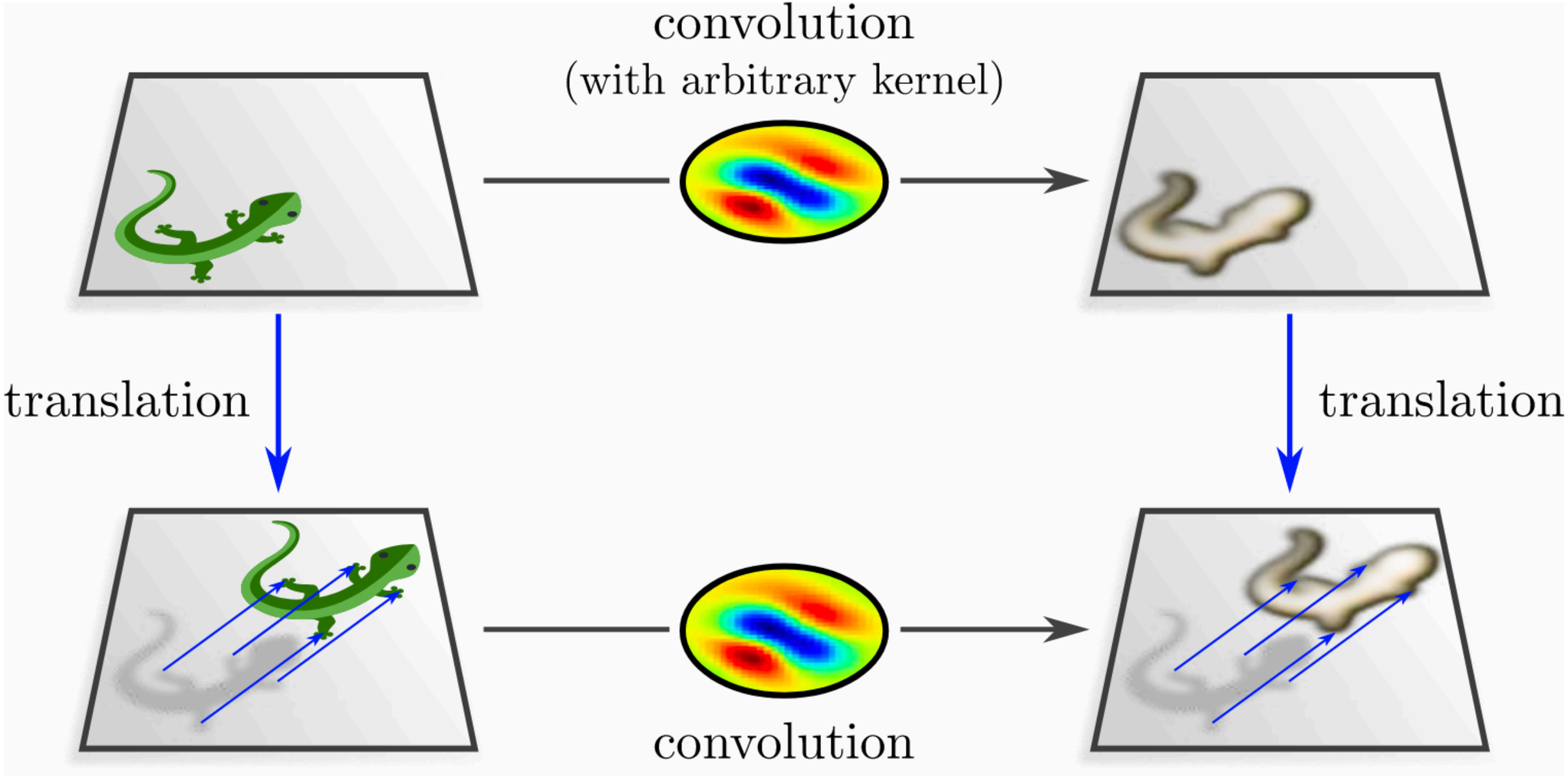
$$\hat{\boldsymbol{\epsilon}}_t^{(x)}, \hat{\boldsymbol{\epsilon}}_t^{(h)} = \mathrm{EGNN}(\boldsymbol{z}_t^{(x)}, [\boldsymbol{z}_t^{(h)}, t/T]) - [\boldsymbol{z}_t^{(x)}, \mathbf{0}]$$

Table 1. Neg. log-likelihood $-\log p(\mathbf{x}, \mathbf{h}, M)$, atom stability and molecule stability with standard deviations across 3 runs on QM9, each drawing 10000 samples from the model.

# Metrics	NLL	Atom stable (%)	Mol stable (%)
E-NF	-59.7	85.0	4.9
G-Schnet	N.A	95.7	68.1
GDM	-94.7	97.0	63.2
GDM-aug	-92.5	97.6	71.6
EDM (ours)	**-110.7**$\pm$1.5	**98.7**$\pm$0.1	**82.0**$\pm$0.4
Data		99.0	95.2

Hoogeboom, Emiel, Victor Garcia Satorras, Clement Vignac, and Max Welling. "Equivariant Diffusion for Molecule Generation in 3D." ICML, 2022

Convolutional neural networks



Equivariant and coordinate independent convolutional networks, M. Weiler 2024

Convolutional neural networks

Equivariance under other group actions?

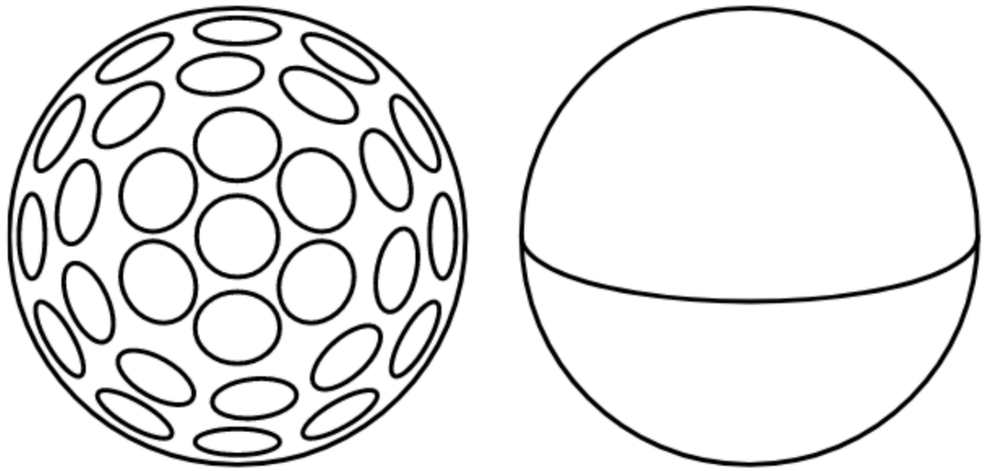
Theorem 1. *A feed forward neural network $\mathcal{N}$ is equivariant to the action of a compact group G on its inputs if and only if each layer of $\mathcal{N}$ implements a generalized form of convolution derived from (1).*

$$(f * g)(u) = \int_G f(uv^{-1})\, g(v)\, d\mu(v). \qquad (1)$$

Kondor, Risi, and Shubhendu Trivedi. “On the Generalization of Equivariance and Convolution in Neural Networks to the Action of Compact Groups.”, 2018
Weiler, Maurice, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. “3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data.”, 2018
Aronsson, J. Homogeneous vector bundles and G-equivariant convolutional neural networks. *Sampling Theory, Signal Processing, and Data Analysis, 2022*

Group convolutions on homogeneous spaces

Equivariant neural networks on the sphere



$\mathrm{SO}(3)$ $\mathrm{SO}(3)/\mathrm{SO}(2)$

$$\psi : \mathrm{SO}(3) \to \mathbb{R}$$
$$f : \mathrm{SO}(3) \to \mathbb{R}$$

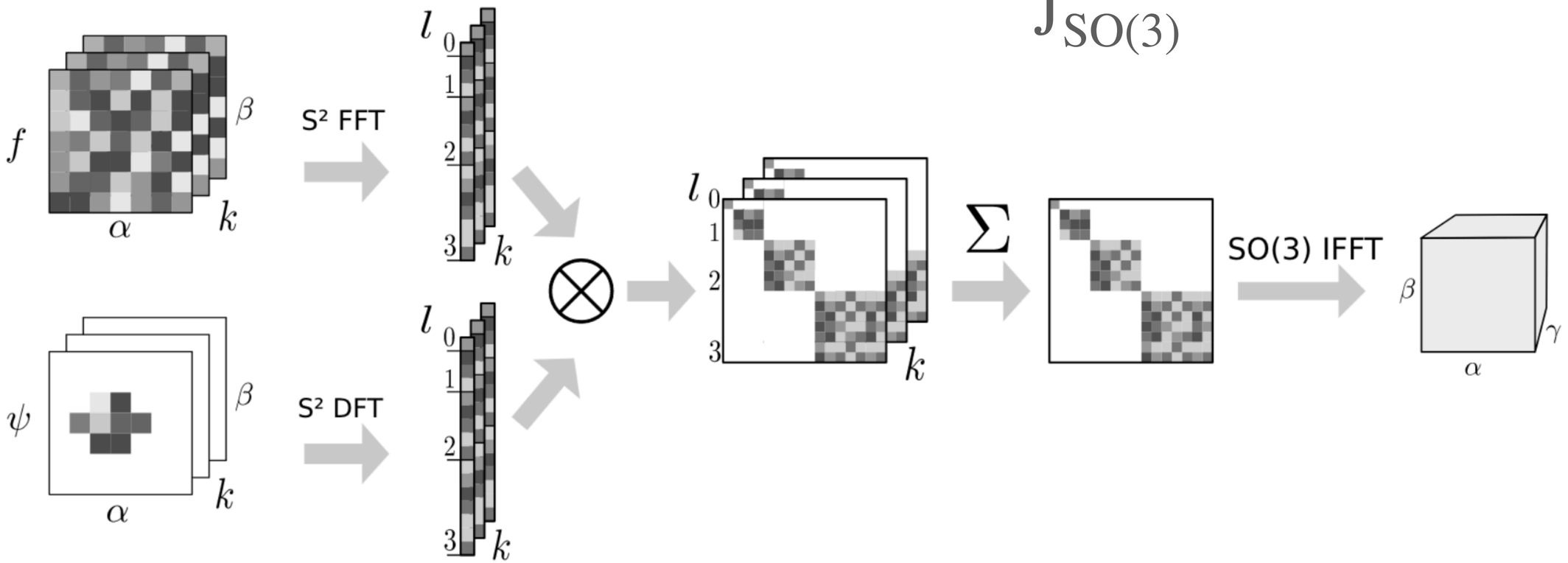
$$(\psi \star f)(u) = \int_{\mathrm{SO}(3)} \psi(u^{-1}g) f(g) \mathrm{d}g$$

Cohen, Taco S., Mario Geiger, Jonas Koehler, and Max Welling. “Spherical CNNs.” *ICLR 2018* http://arxiv.org/abs/1801.10130.

Group convolutions on homogeneous spaces

Equivariant neural networks on the sphere

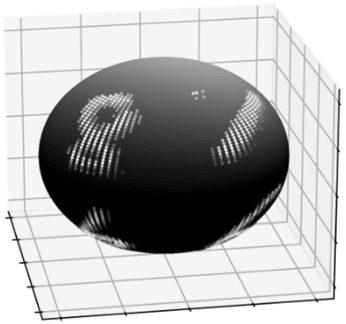
$$(\psi \star f)(u) = \int_{\mathrm{SO}(3)} \psi(u^{-1}g) f(g) \mathrm{d}g$$

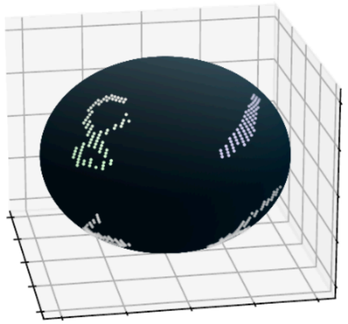


Cohen, Taco S., Mario Geiger, Jonas Koehler, and Max Welling. “Spherical CNNs.” *ICLR 2018* http://arxiv.org/abs/1801.10130.

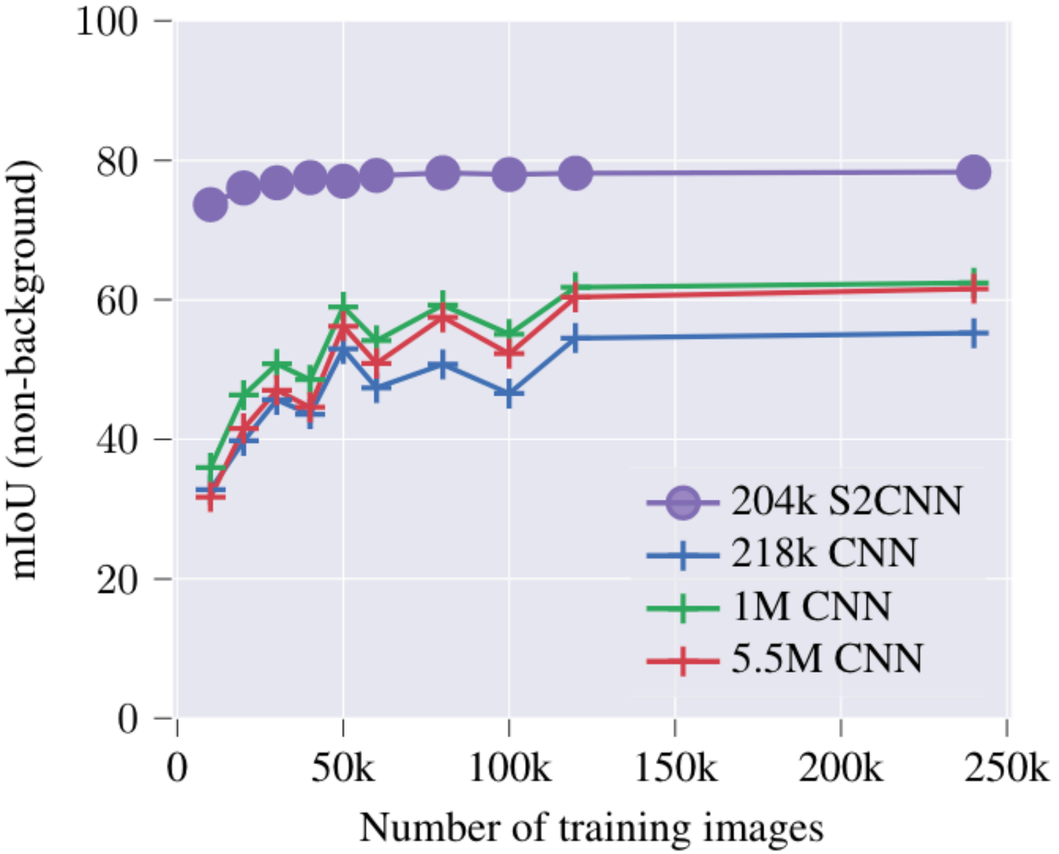
Geometric deep learning

Equivariance vs augmentation





60 x 60 Driscoll-Healy grid

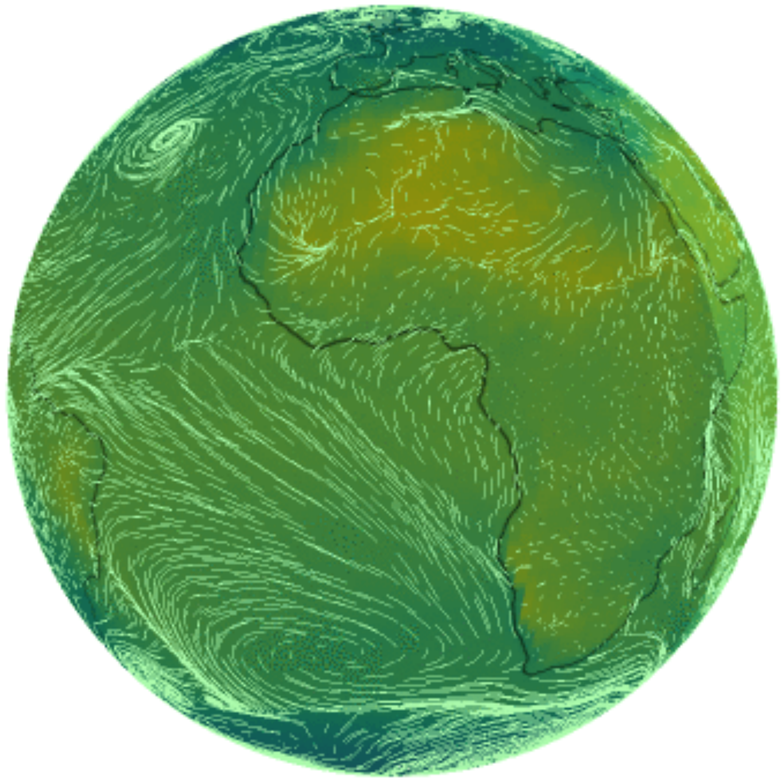


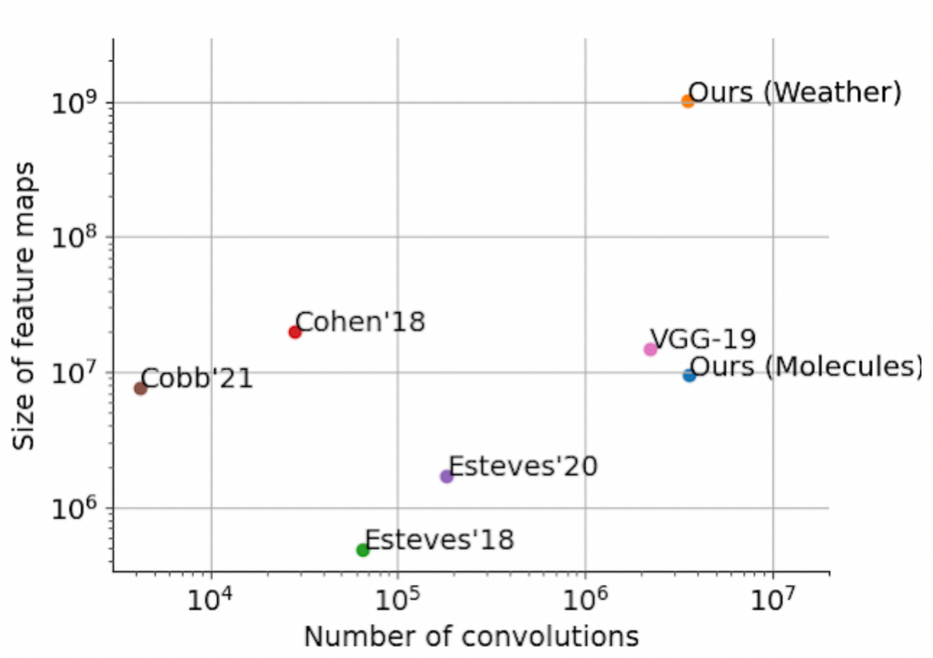
	Batch size	Latency (ms)	Throughput (N/s)
S2CNN	1	111 ± 0.6	9.0 ± 0.04
CNN	1	5.93 ± 0.24	169 ± 5.8

Nvidia T4 16GB GPU

Gerken J, Carlsson O, Linander H, Ohlsson F, Petersson C, Persson D. "Equivariance versus augmentation for spherical images." ICML, 2022

Scaling group convolutions

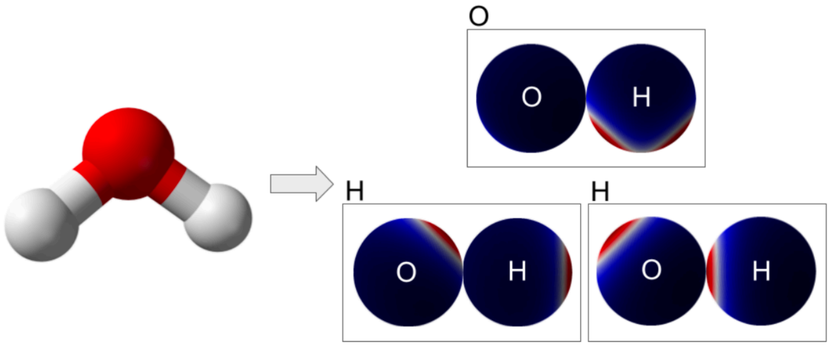




Esteves, Carlos, Ameesh Makadia, and Kostas Daniilidis. "Spin-weighted spherical cnns." *NeurIPS* (2020)
Esteves, Carlos, Jean-Jacques Slotine, and Ameesh Makadia. "Scaling spherical cnns." *arXiv:2306.05420* (2023)

Group convolutions on homogeneous spaces

Equivariant neural networks on the sphere



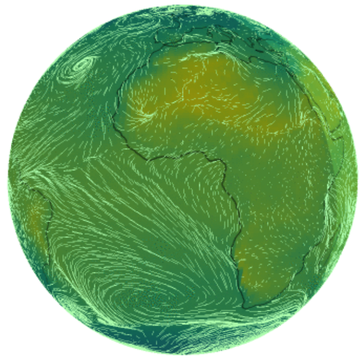


Table 2. QM9 mean average errors (MAE). We scale spherical CNNs for QM9 for the first time, and show they are competitive with the previously dominant equivariant graph neural networks and transformers. We compare on two splits found in the literature, where "Split 1" has a larger training set. Our model outperforms the baselines on 8 out of 12 targets in "Split 1" and 9 out of 12 targets in "Split 2".

		μ [D]	α [a_0^3]	ϵ_{HOMO} [meV]	ϵ_{LUMO} [meV]	ϵ_{gap} [meV]	$< R^2 >$ [a_0^2]	zpve [meV]	U_0 [meV]	U [meV]	H [meV]	G [meV]	C_v [$\frac{\text{cal}}{\text{mol K}}$]
Split 1	DimeNet++ (2020)	0.030	0.044	24.6	19.5	32.6	0.331	1.21	6.32	6.28	6.53	7.56	0.023
	PaiNN (2021)	0.012	0.045	27.6	20.4	45.7	0.066	1.28	5.85	5.83	5.98	7.35	0.024
	TorchMD-Net (2022)	0.011	0.059	20.3	17.5	36.1	0.033	1.84	6.15	6.38	6.16	7.62	0.026
	Ours	0.016	0.049	21.6	18.0	28.8	0.027	1.15	5.65	5.72	5.69	6.54	0.022
Split 2	EGNN (2021)	0.029	0.071	29.0	25.0	48.0	0.106	1.55	11.00	12.00	12.00	12.00	0.031
	SEGNN (2022)	0.023	0.060	24.0	21.0	42.0	0.660	1.62	15.00	13.00	16.00	15.00	0.031
	Equiformer (2022)	0.014	0.056	17.0	16.0	33.0	0.227	1.32	10.00	11.00	10.00	10.00	0.025
	Ours	0.017	0.049	22.3	19.1	29.8	0.028	1.19	5.96	5.98	5.97	6.97	0.023

	3 days			5 days		
	Z500 [m^2/s^2]	T850 [K]	T2M [K]	Z500 [m^2/s^2]	T850 [K]	T2M [K]
2 predictors						
Rasp et al. (2020)	626	2.87	-	757	3.37	-
Ours	531	2.38	-	717	3.03	-
117 predictors						
Rasp & Thuerey$^{\text{cont}}$	331	1.87	1.60	545	2.57	2.06
Rasp & Thuerey	314	1.79	1.53	561	2.82	2.32
Ours	329	1.62	1.29	601	2.57	1.89
Pretrained						
Rasp & Thuerey$^{\text{pre}}$	268	1.65	1.42	523	2.52	2.03
Rasp & Thuerey$^{\text{pre,cont}}$	284	1.72	1.48	499	2.41	1.92

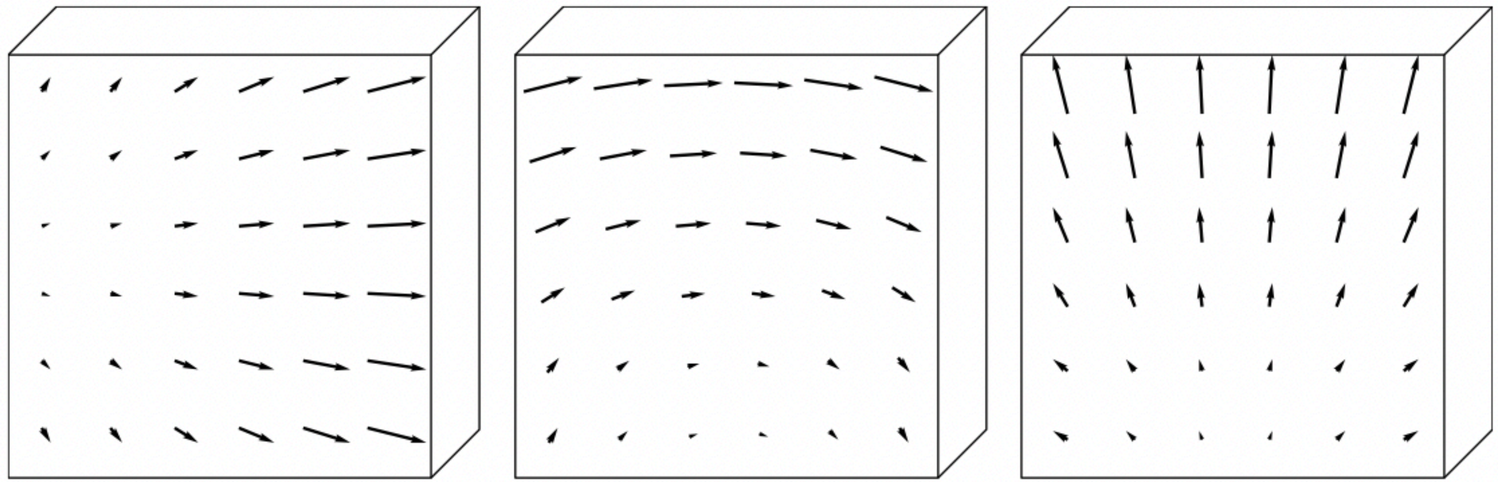
JAX implementation
https://github.com/google-research/spherical-cnn

Esteves, Carlos, Jean-Jacques Slotine, and Ameesh Makadia. "Scaling spherical cnns." *arXiv:2306.05420* (2023)

Steerable convolution

Equivariant convolution on non-scalar features

$$f(x) \in \mathbb{R}^n \qquad g \in G \qquad \rho : G \to GL(V)$$



$f(x)$ $\qquad$ $f(g^{-1}x)$ $\qquad$ $\rho(g)f(g^{-1}x)$

Weiler, Maurice, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. “3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data.” 2018. http://arxiv.org/abs/1807.02547.

Steerable convolution

Equivariant convolution on non-scalar features

$$(k \star f)^i(y) = \int_{\mathbb{R}^n} k^{ij}(y) f_j(x - y) dx$$

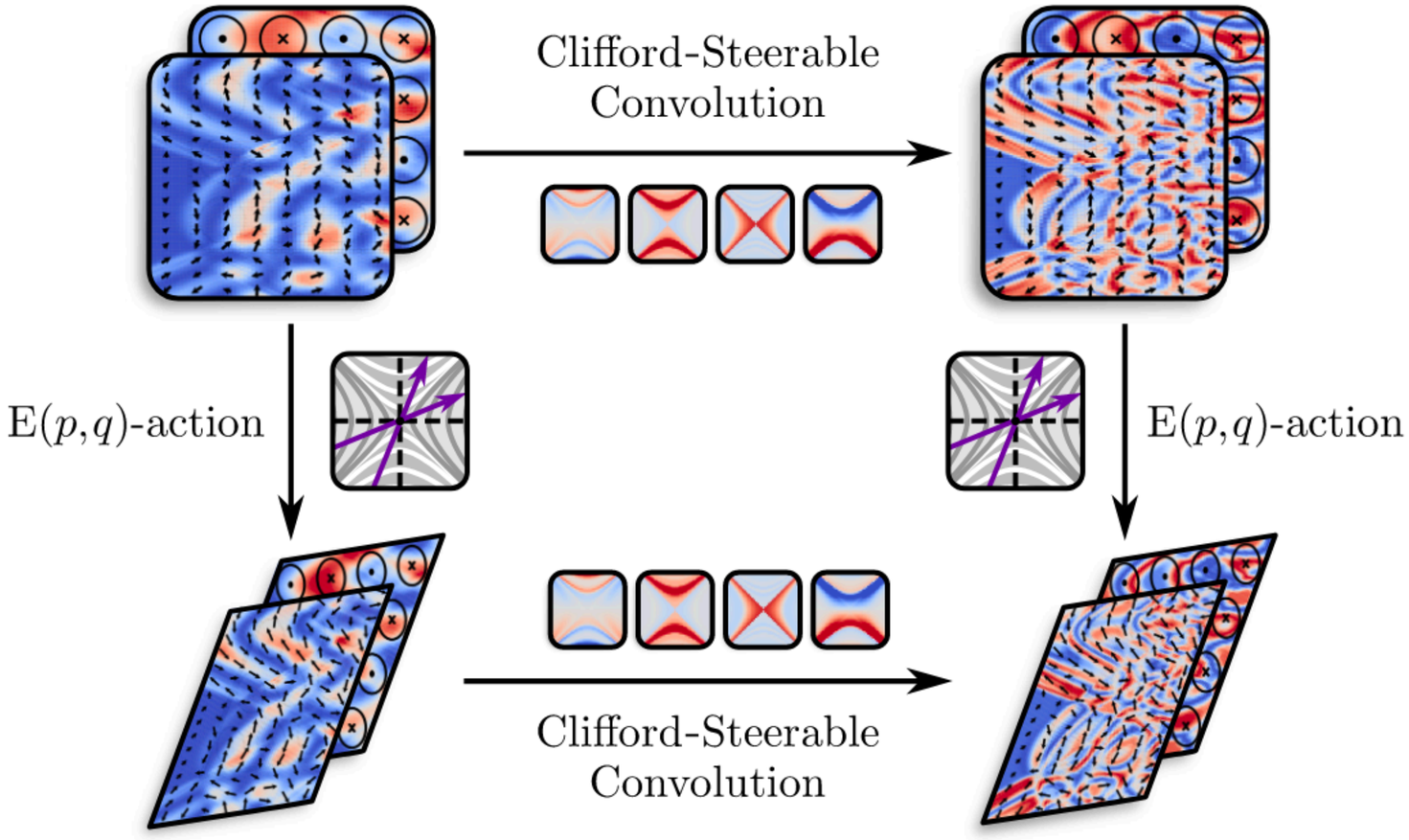
Convolution equivariant iff k satisfies constraint

$$k(gx) = \rho_{out}(g) k(x) \rho_{in}(g^{-1})$$

Such a kernel is called “steerable”, in close analogy with steerable filters.

Weiler, Maurice, and Gabriele Cesa. “General E(2)-Equivariant Steerable CNNs.” NeurIPS, 2019
Freeman, W.T., and E.H. Adelson. “The Design and Use of Steerable Filters.” *IEEE Pattern Analysis and Machine Intelligence,* 1991

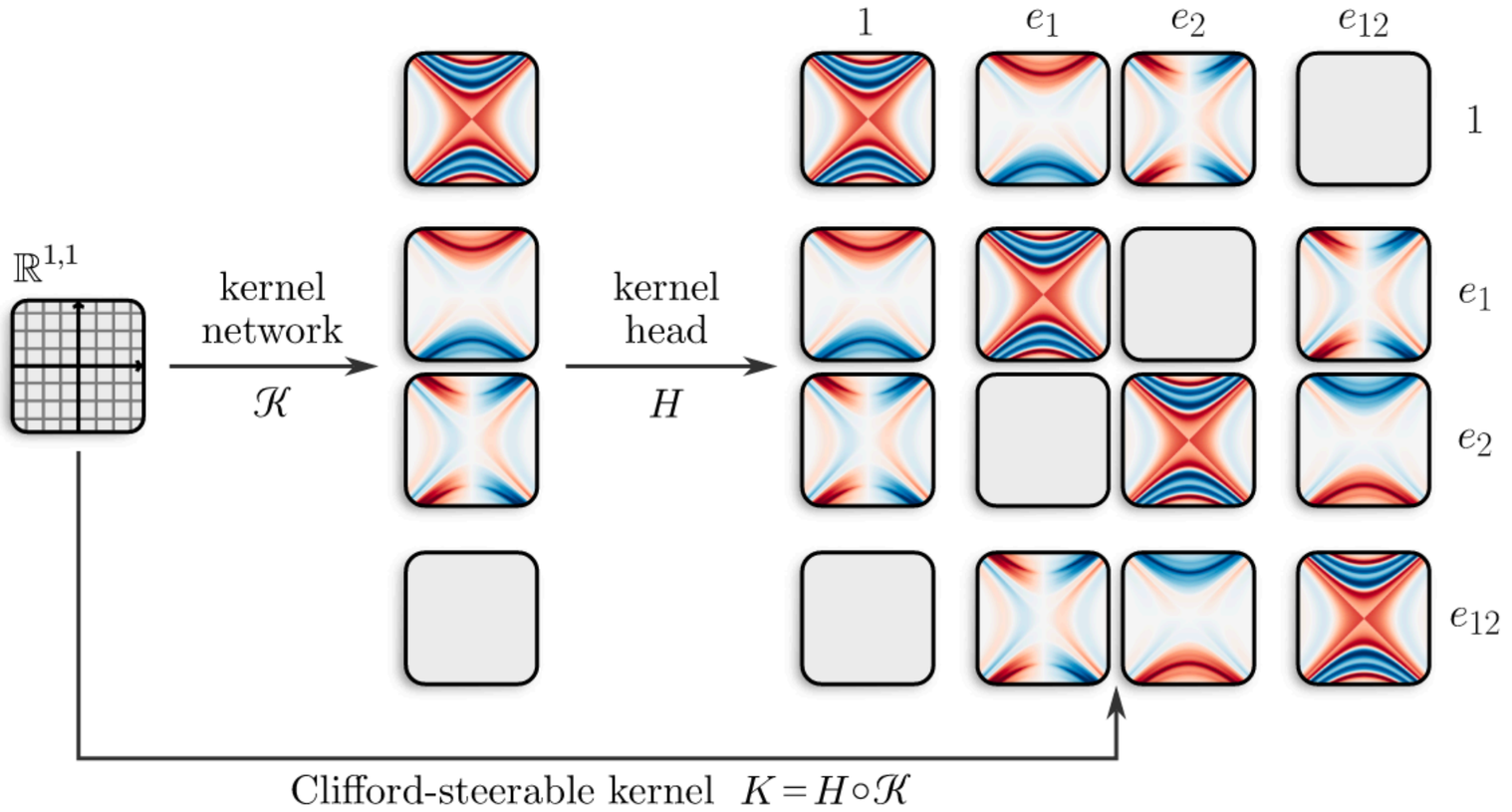
Clifford-Steerable convolution



Zhdanov, Maksim, David Ruhe, Maurice Weiler, Ana Lucic, Johannes Brandstetter, and Patrick Forré. "Clifford-steerable convolutional neural networks." *arXiv:2402.14730*

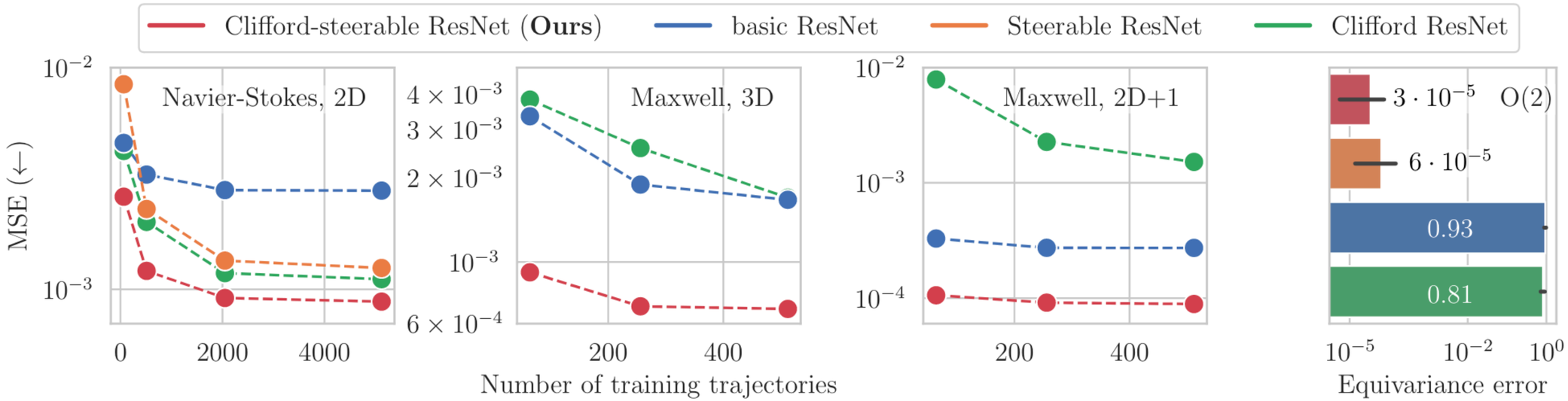
Clifford-Steerable convolution

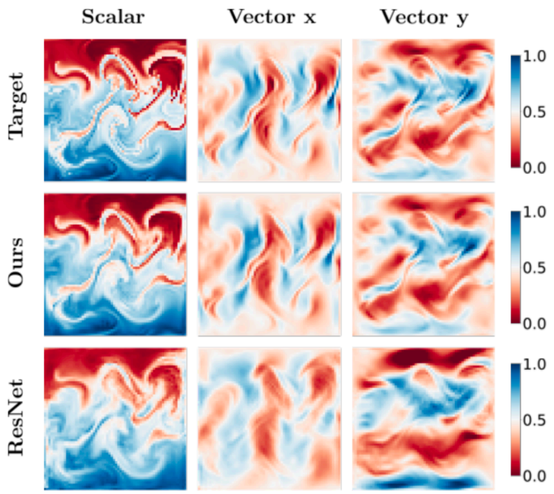
Solving the steerable kernel constraint



Zhdanov, Maksim, David Ruhe, Maurice Weiler, Ana Lucic, Johannes Brandstetter, and Patrick Forré. "Clifford-steerable convolutional neural networks." *arXiv:2402.14730*

Geometric deep learning





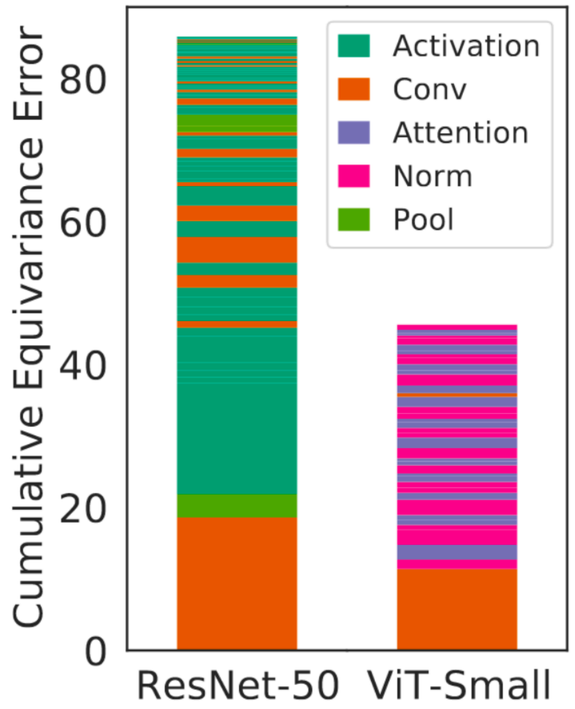
"In the NavierStokes experiment, [Clifford-steerable ResNet] require only 64 trajectories to outperform the basic ResNet trained on 80× more data."

Zhdanov, Maksim, David Ruhe, Maurice Weiler, Ana Lucic, Johannes Brandstetter, and Patrick Forré. "Clifford-steerable convolutional neural networks." *arXiv:2402.14730*

Soft inductive bias

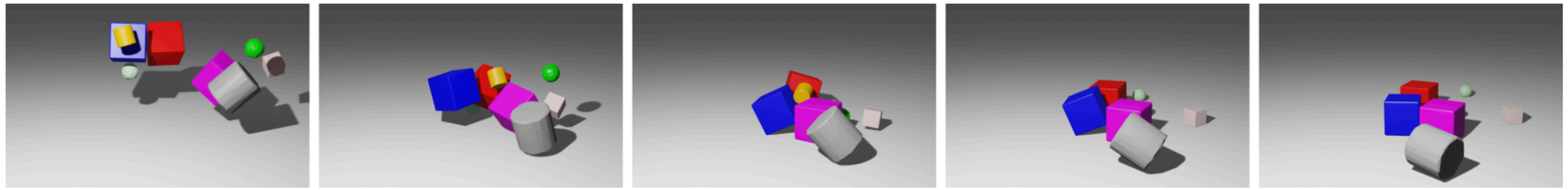
"Similarly to some recent work, we find that no invariance or equivariance with respect to global rotations and translation of the molecule are required in the architecture and we therefore omit them to simplify the machine learning architecture."

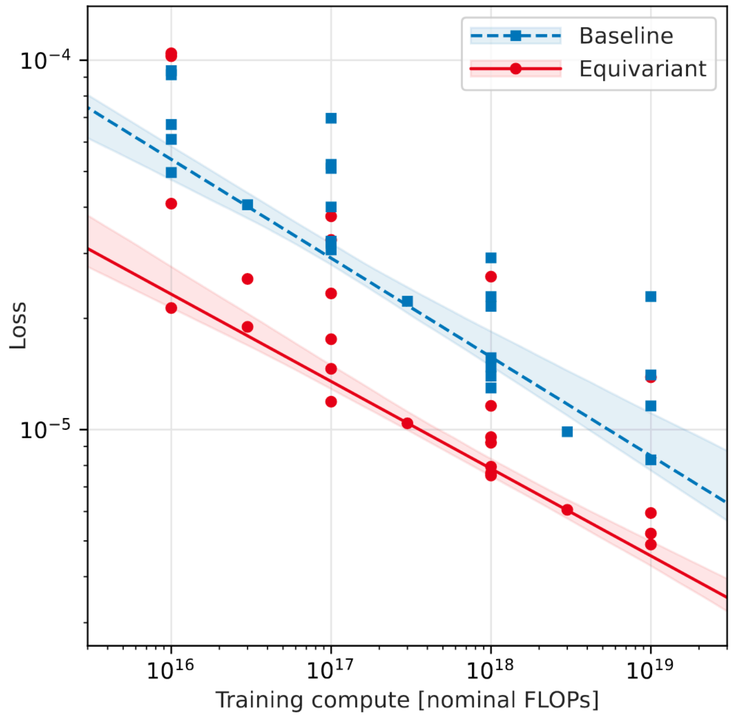
Abramson, Josh, et al. "Accurate structure prediction of biomolecular interactions with AlphaFold 3." *Nature, 2024*

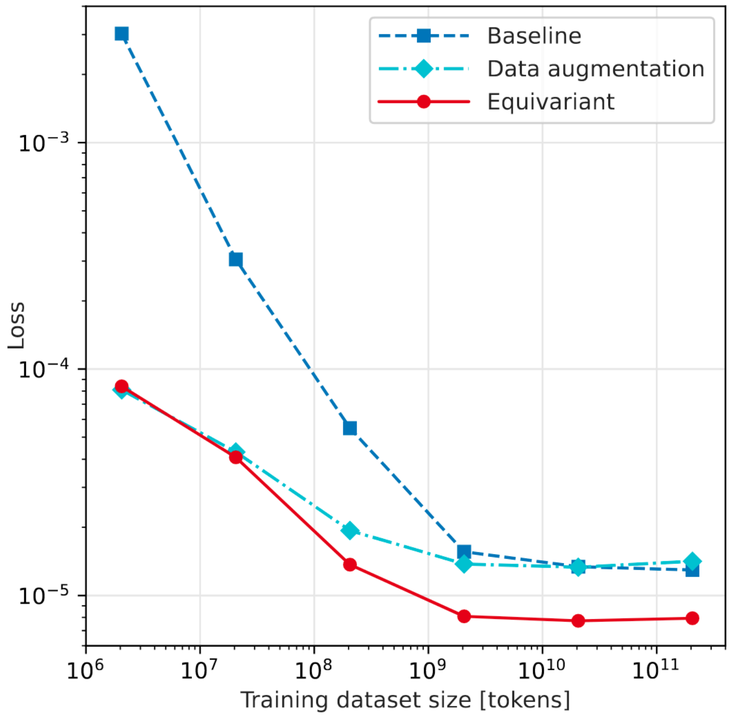


Gruver, Nate, Marc Finzi, Micah Goldblum, and Andrew Gordon Wilson. "The Lie Derivative for Measuring Learned Equivariance.", ICLR 2023. arXiv.2210.02984.

Geometric deep learning







Brehmer, Johann, Sönke Behrends, Pim de Haan, and Taco Cohen. “Does Equivariance Matter at Scale?” 2024, arXiv:2410.23179

hampus.linander@pm.me

gapindnns.github.io