Efficient architectures

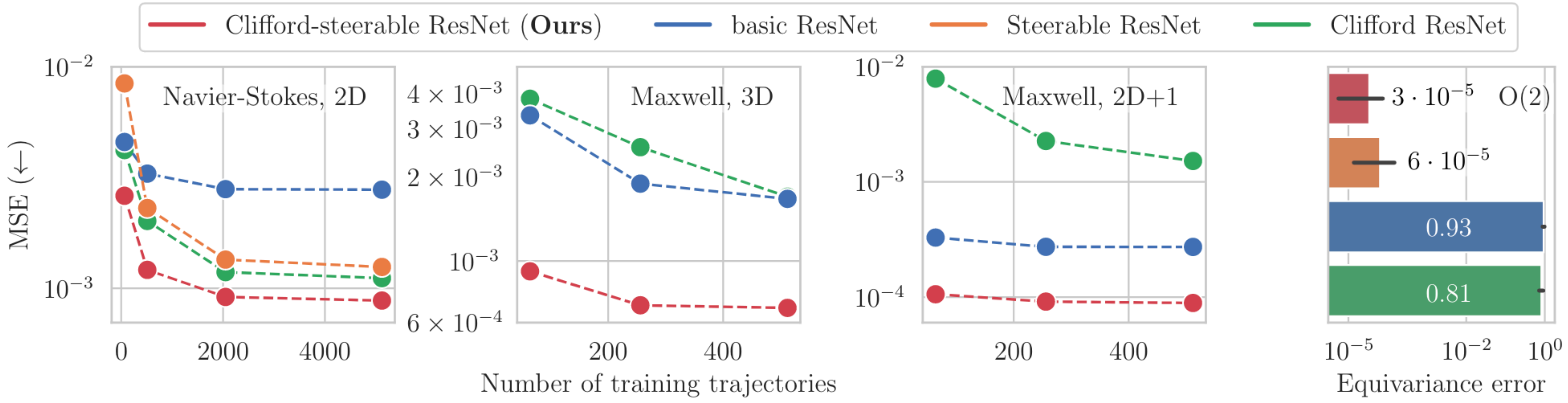
with geometric deep learning

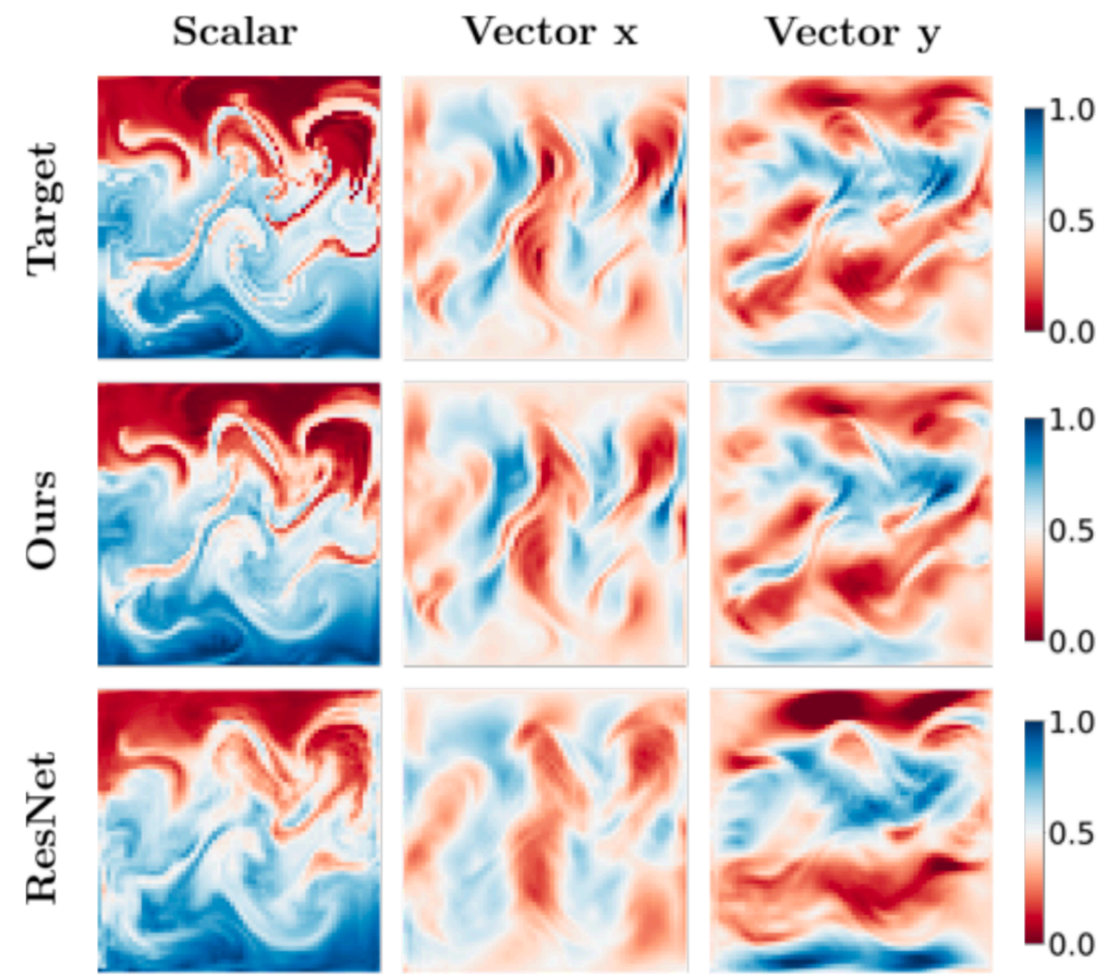
Hampus Linander 2024-09-25 EdgeML school

VERSES AI & Chalmers university of technology

Geometric deep learning

A peek at recent results



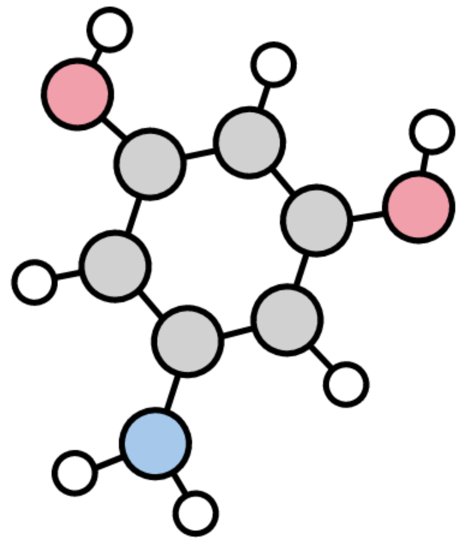


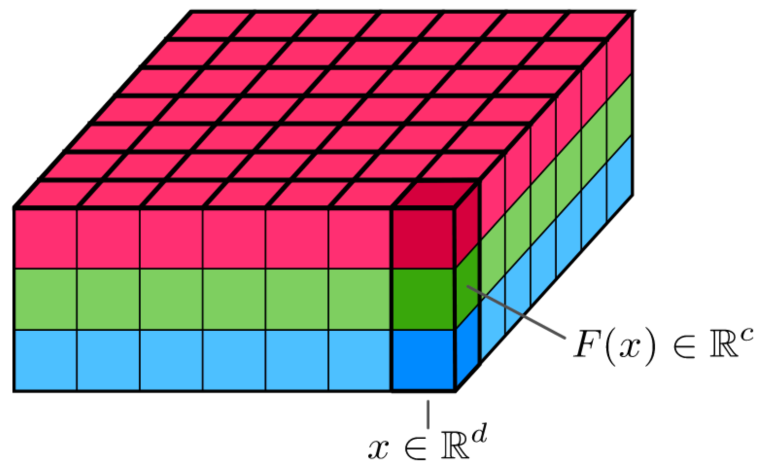
"In the NavierStokes experiment, [Clifford-steerable ResNet] require only 64 trajectories to outperform the basic ResNet trained on 80× more data."

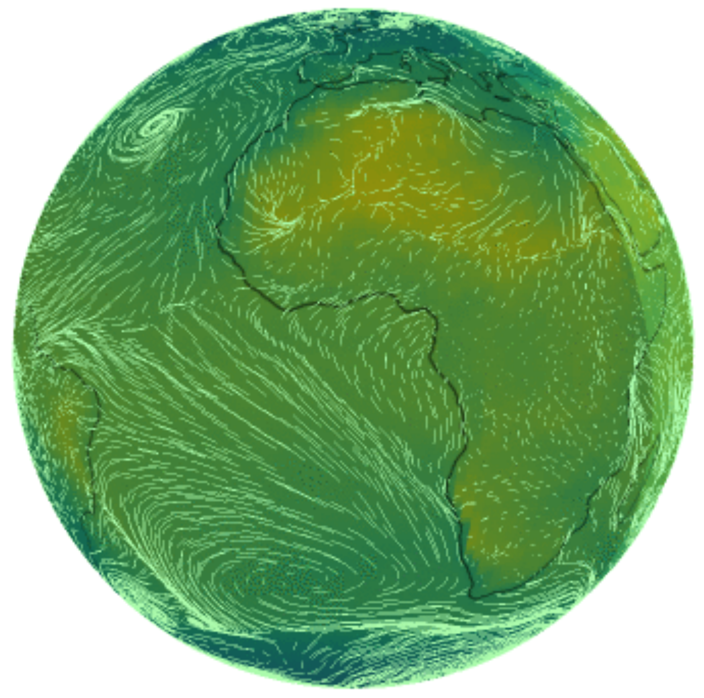
Zhdanov, Maksim, David Ruhe, Maurice Weiler, Ana Lucic, Johannes Brandstetter, and Patrick Forré. "Clifford-steerable convolutional neural networks." *arXiv:2402.14730*

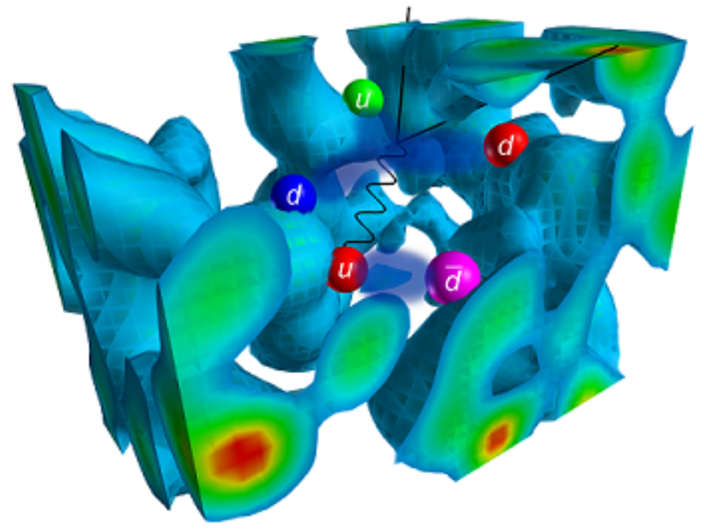
Geometric deep learning

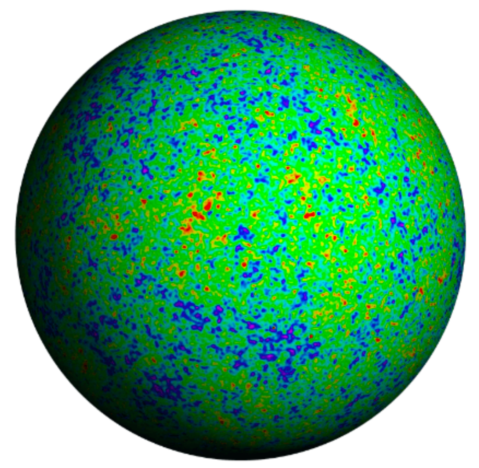
Geometric, topological, and algebraic structure in data can inform model architecture.











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.

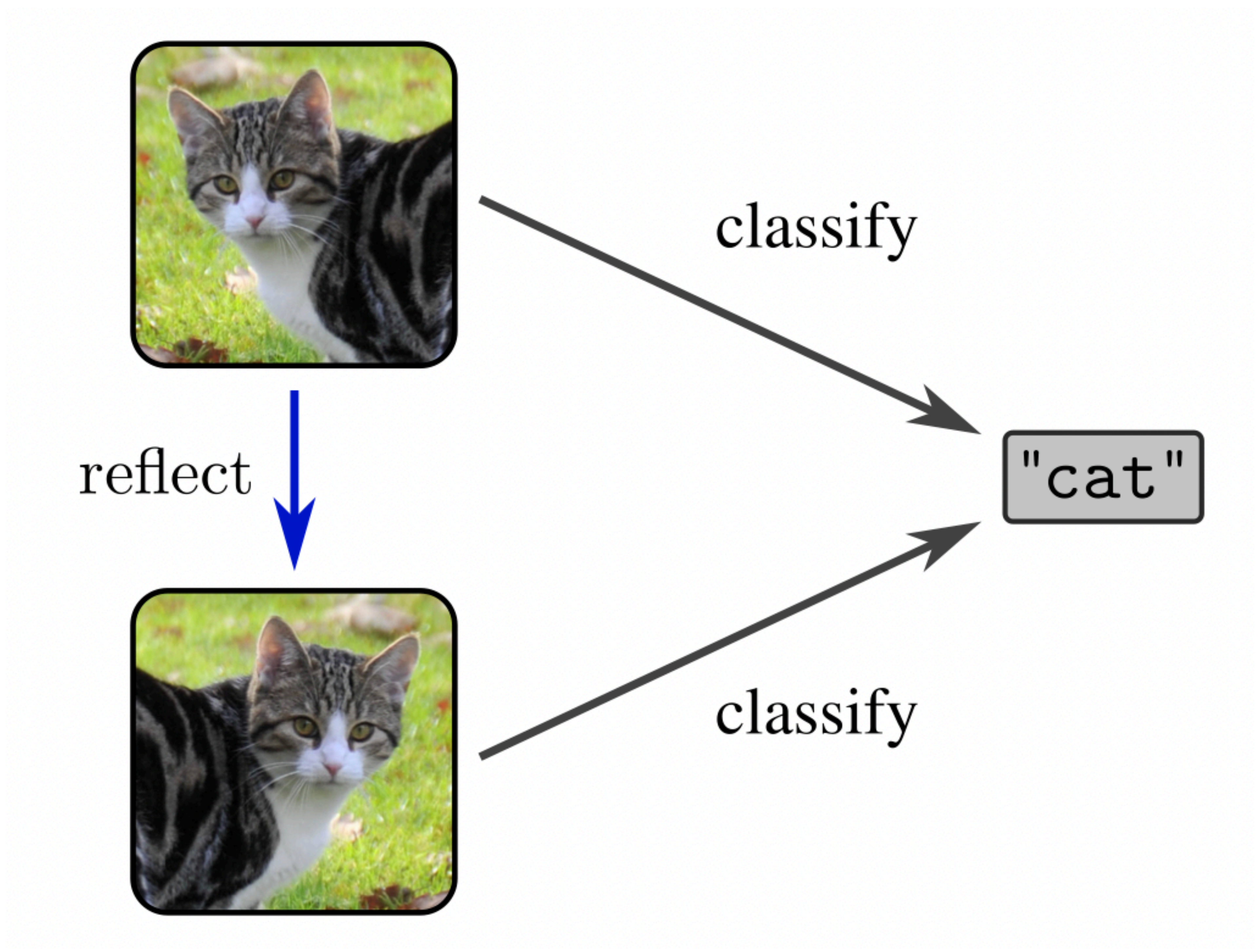
Hoogeboom, Emiel, Victor Garcia Satorras, Clement Vignac, and Max Welling. “Equivariant Diffusion for Molecule Generation in 3D.” ICML2022

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

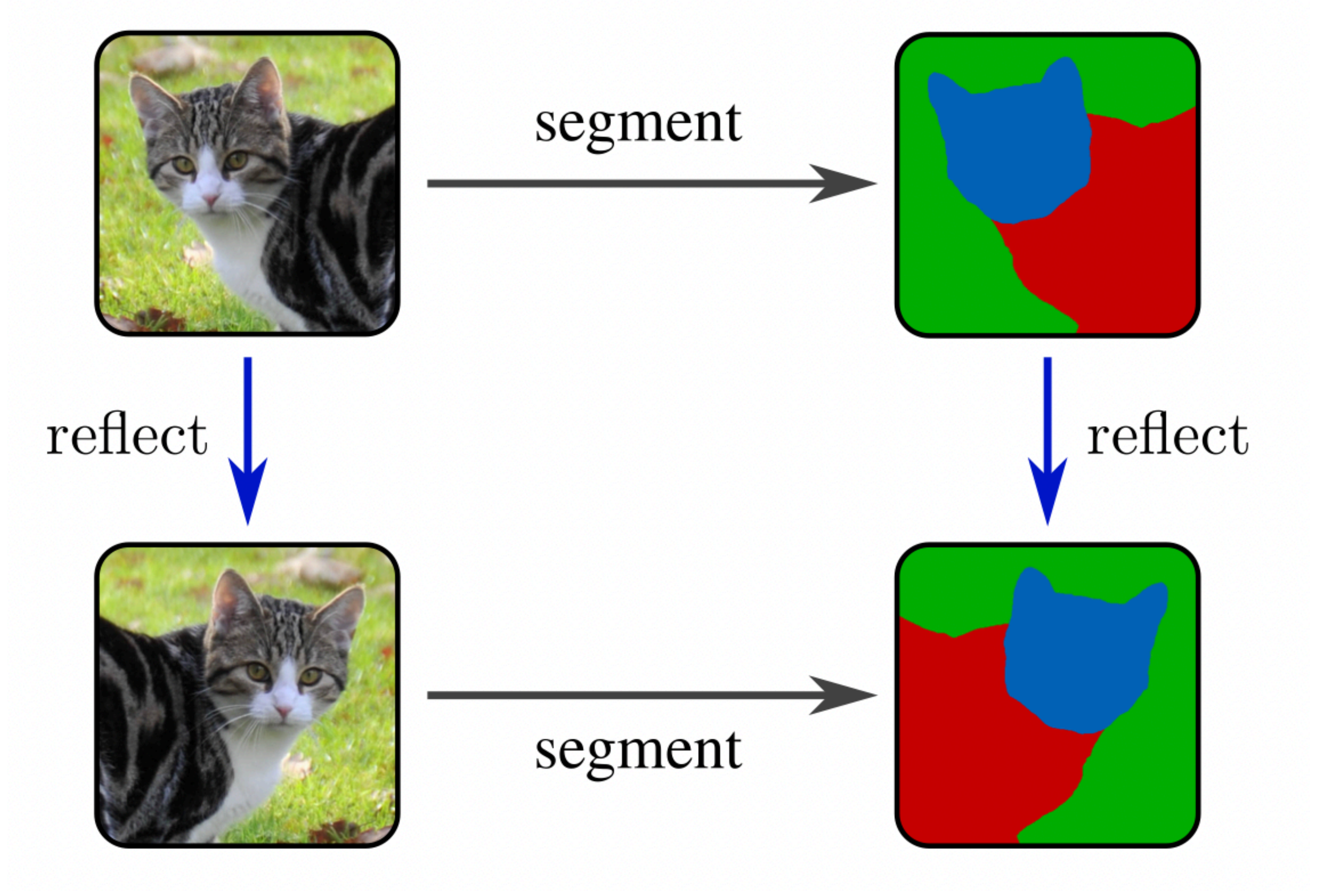
“Equivariant and coordinate independent convolutional networks” M. Weiler 2024, Thesis

Invariance and equivariance

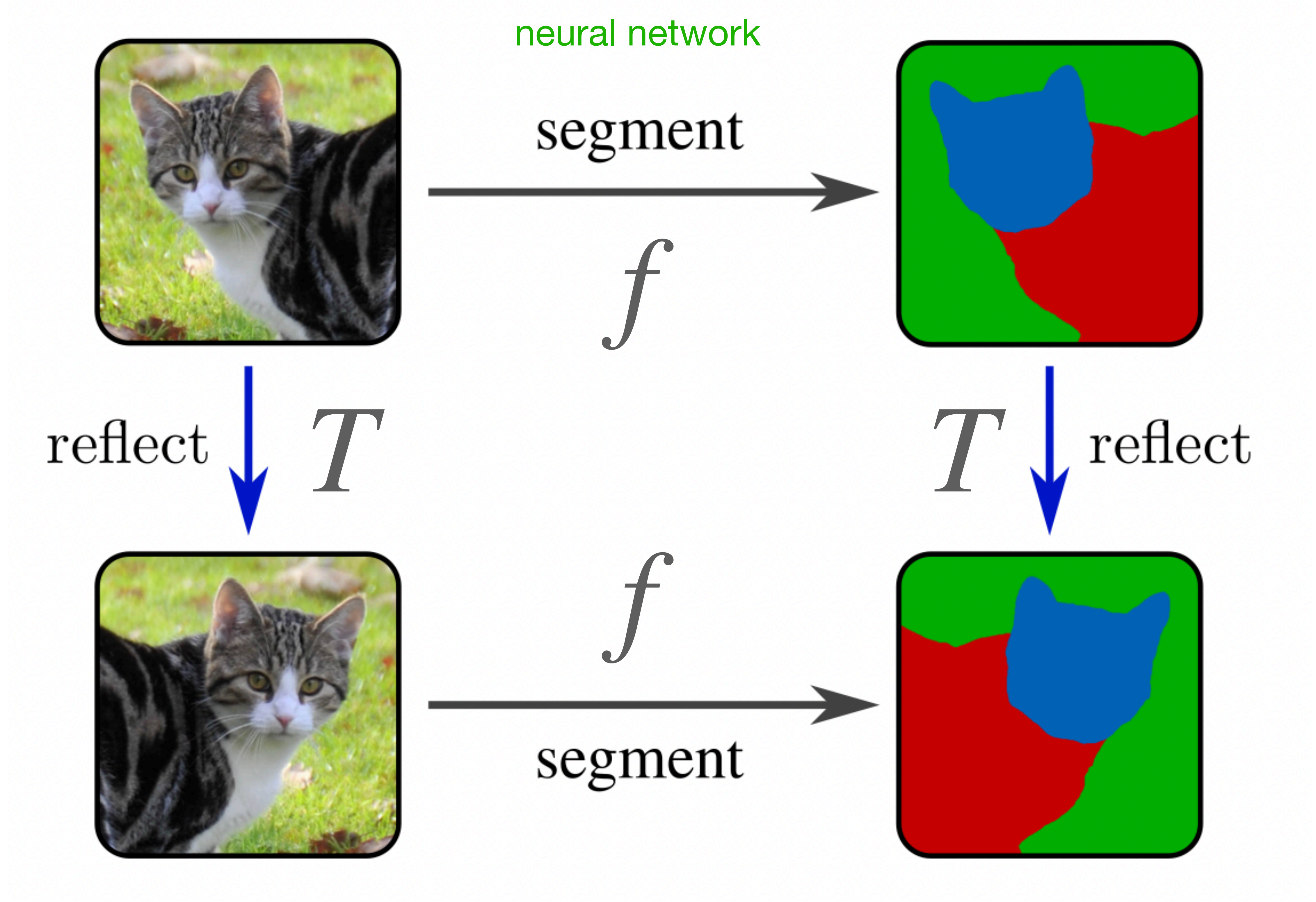
Invariance



Equivariance



Equivariance



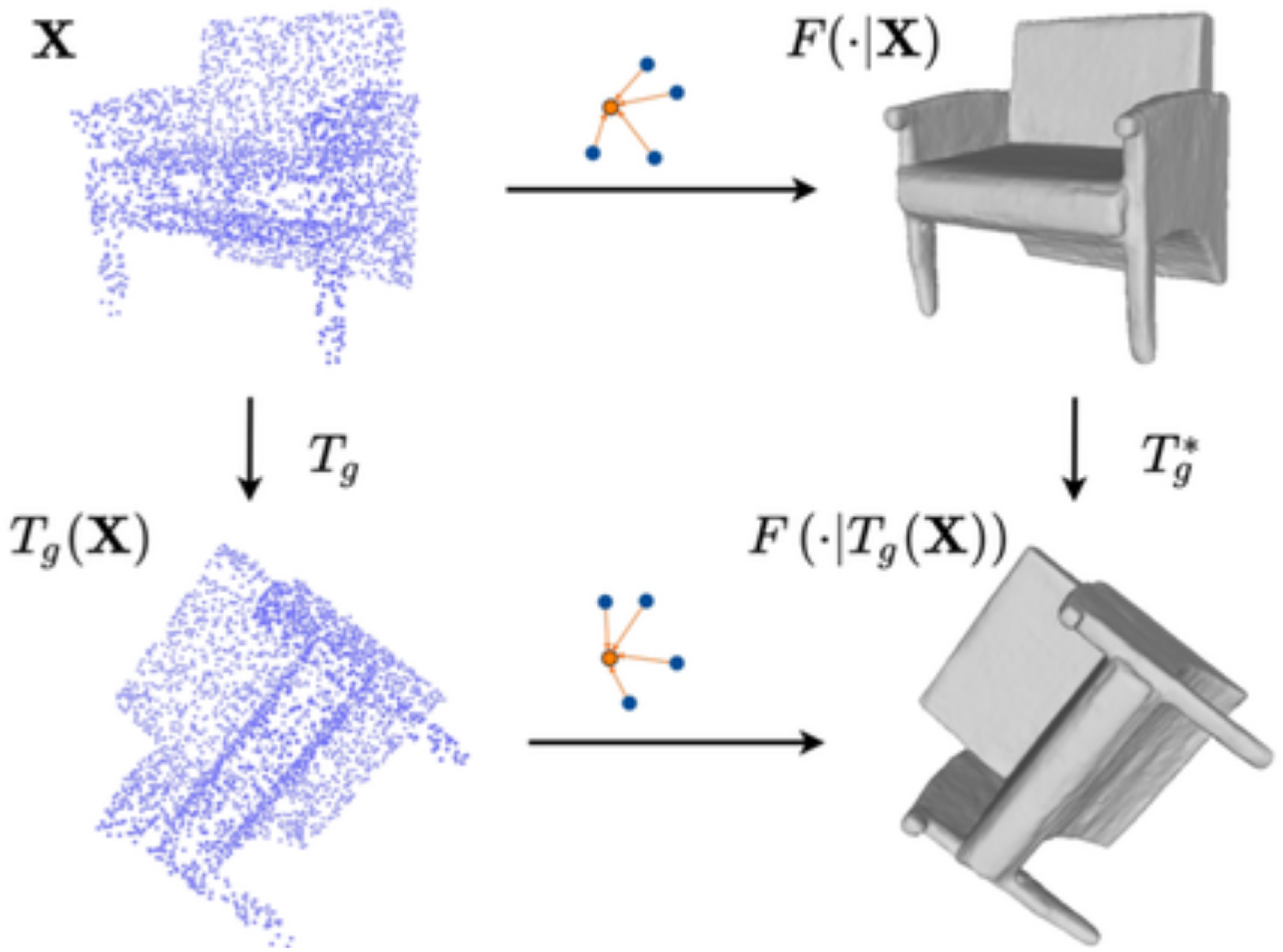
f is equivariant under transformation T if

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

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

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

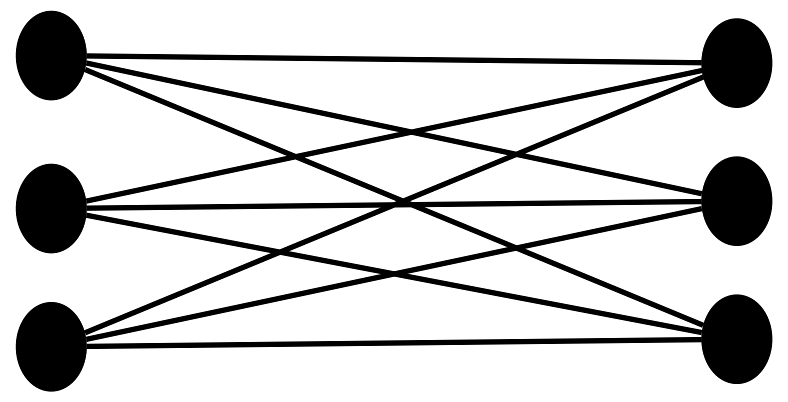
Equivariance



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

MLP - non-equivariant

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



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

Linear transform of input

$$M \in \mathbb{R}^{m\times m}$$
$$x = Mx_0$$

How does the output transform?
Could it be that

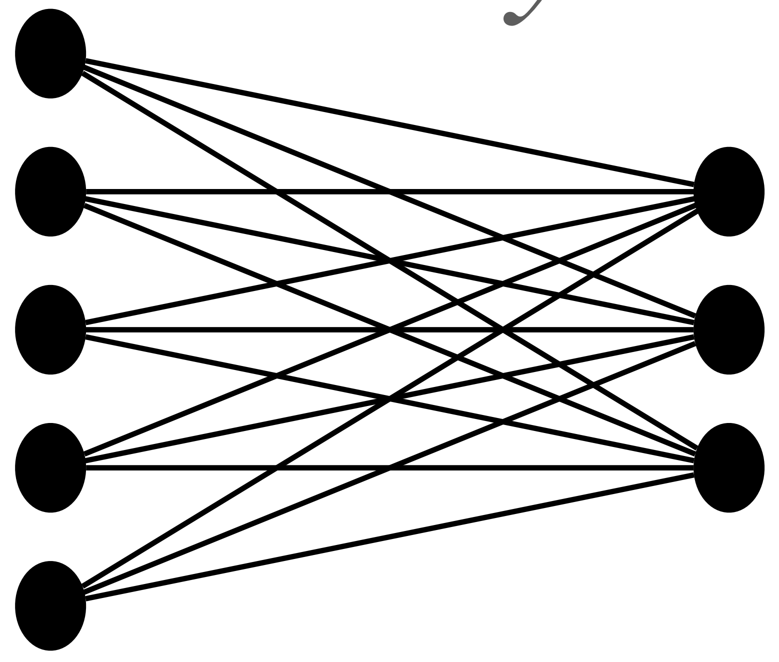
$$y(Mx_0) \overset{?}{=} My(x_0)$$

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

Vector neuron

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

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

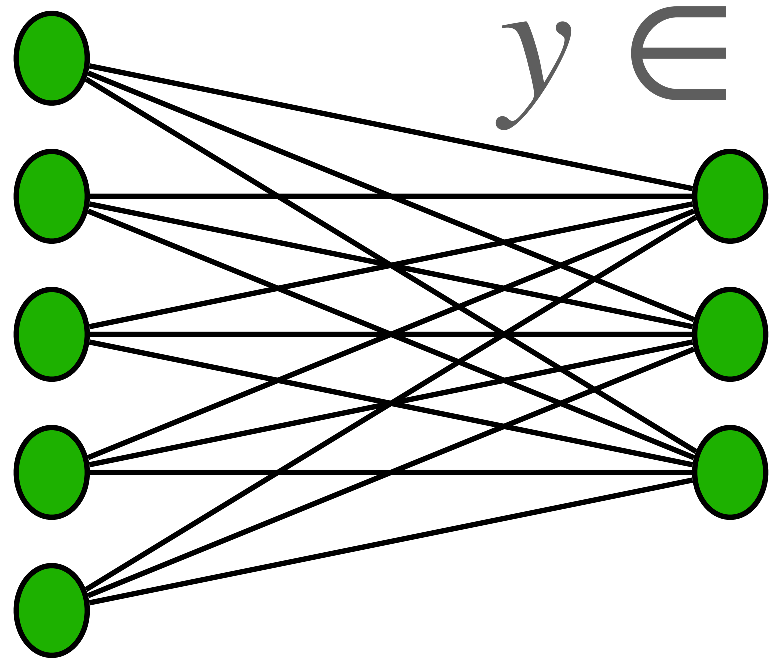


$$y(x) = Wx = W^{ij} x_j \hat{e}_i$$

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 R³

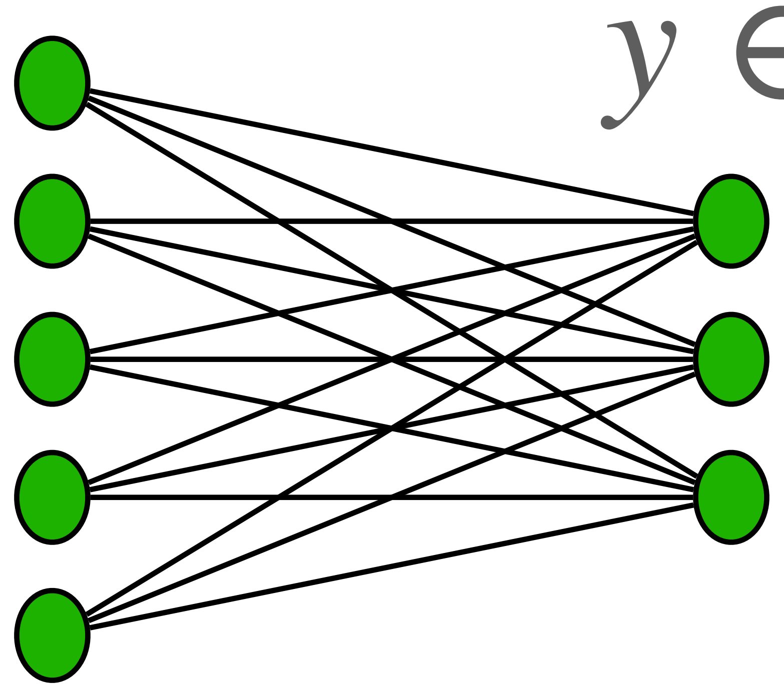
$$y(x) = Wx = W^{ij} x_j^{\alpha} \, \hat{e}_i \, \hat{f}_\alpha$$

Vector neurons

are equivariant

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

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



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

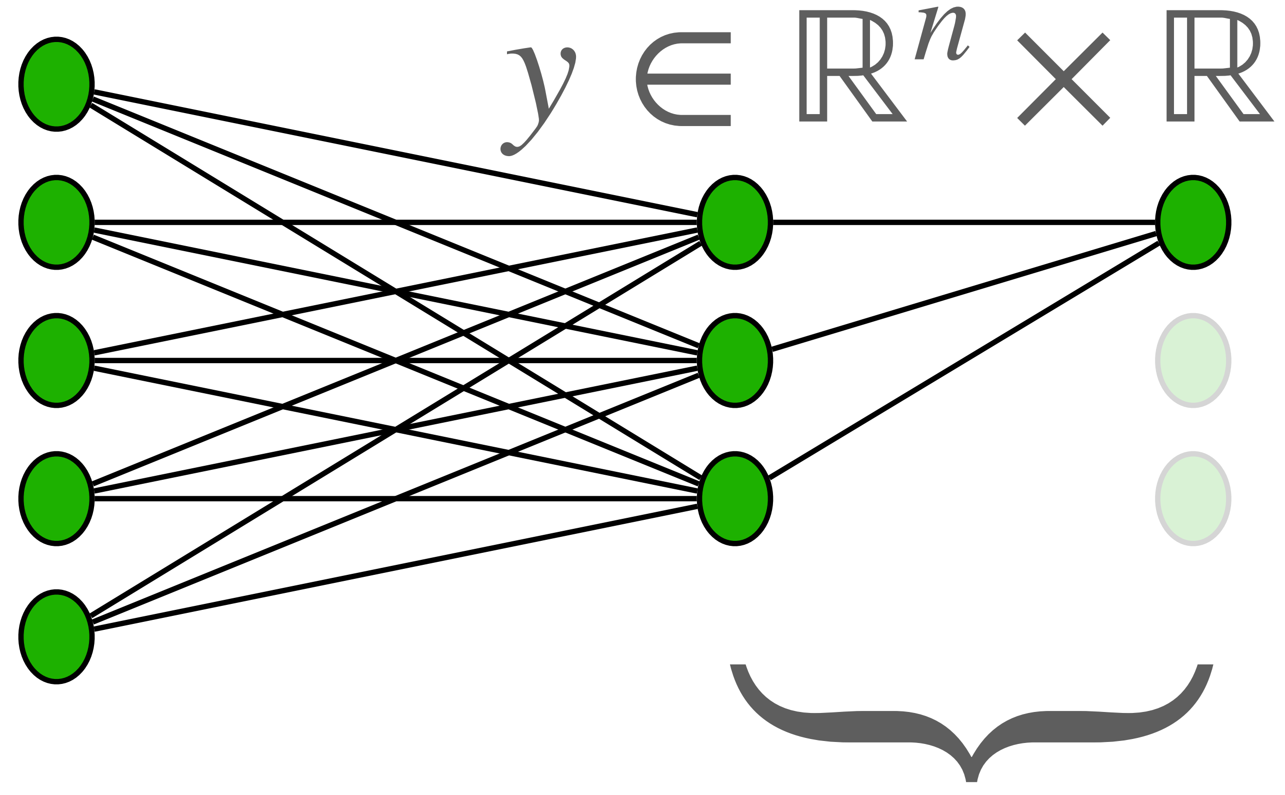
$$\begin{aligned} y(Tx) &= WTx \\ &= W^{ij} T^{\alpha\beta} x_{j\beta}\, \hat{e}_i\, \hat{f}_\alpha \\ &= TWx \end{aligned}$$

Vector neuron

ReLU

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

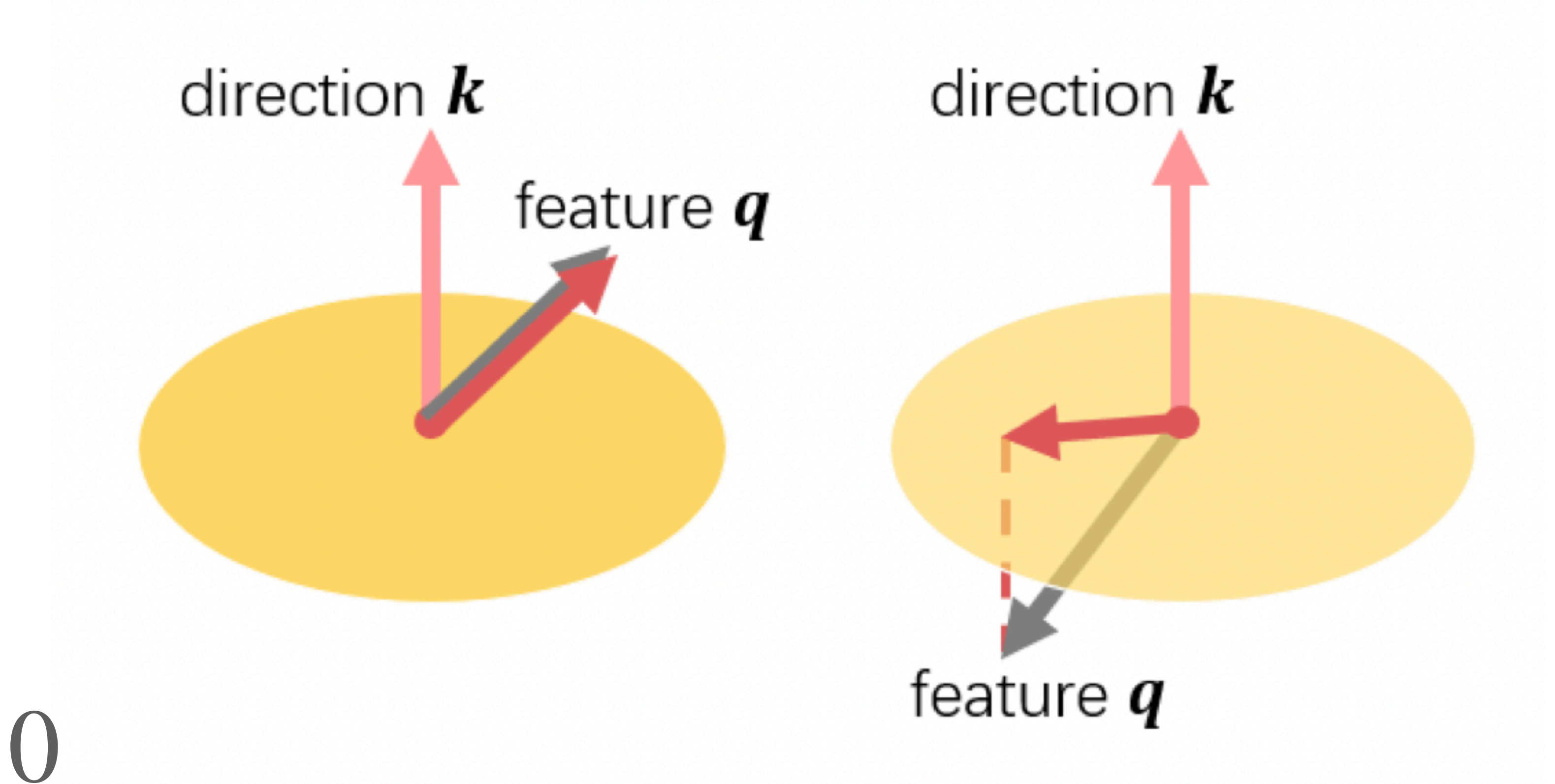
$$y \in \mathbb{R}^n \times \mathbb{R}^3$$



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

$$k = Uy$$

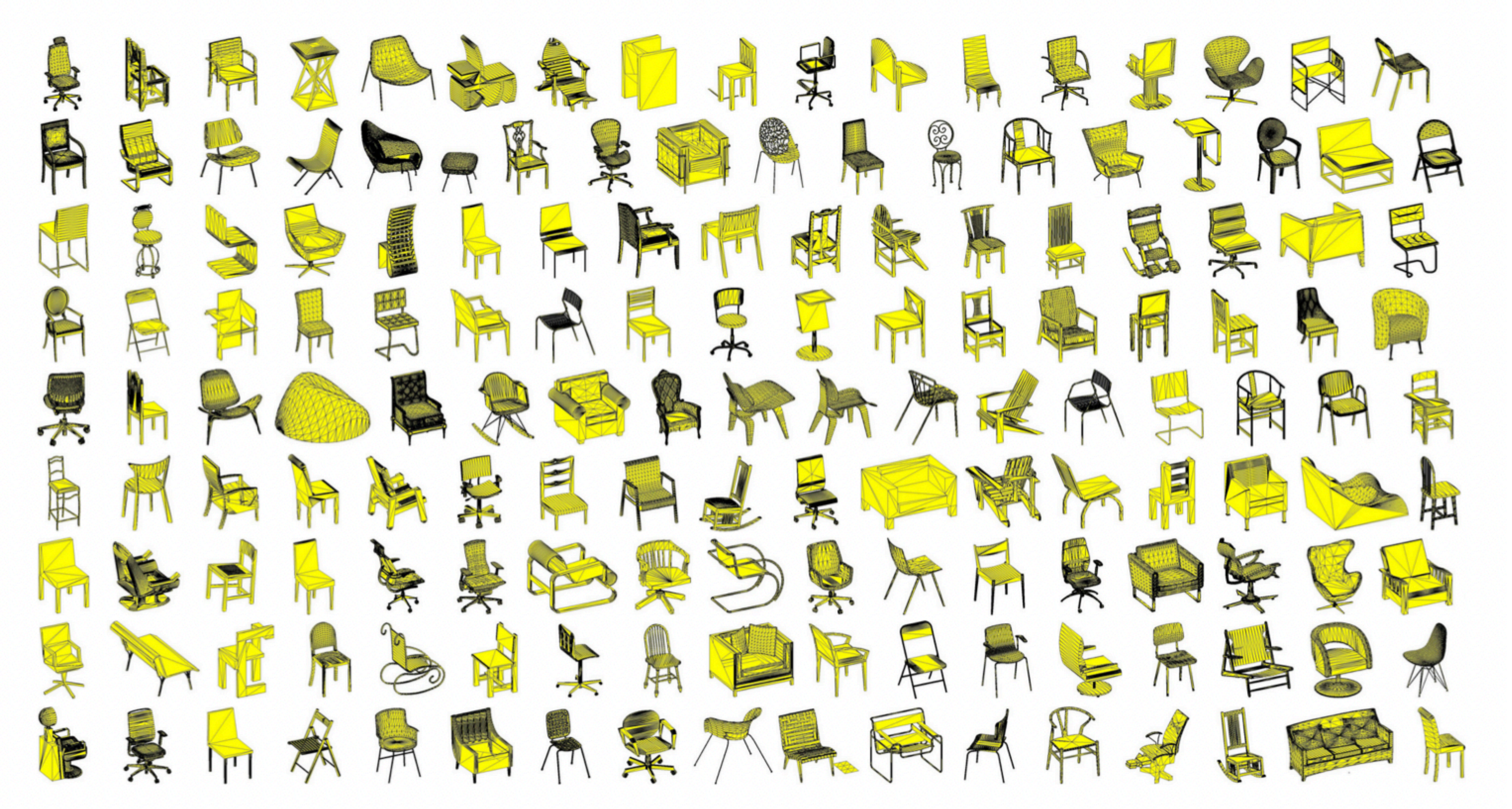
$$y' = \begin{cases} q & \text{if } \langle q, k \rangle \geq 0 \\ q - \left\langle q, \frac{k}{|k|} \right\rangle \frac{k}{|k|} & \text{otherwise} \end{cases}$$



Vector neuron

Classification on ModelNet40

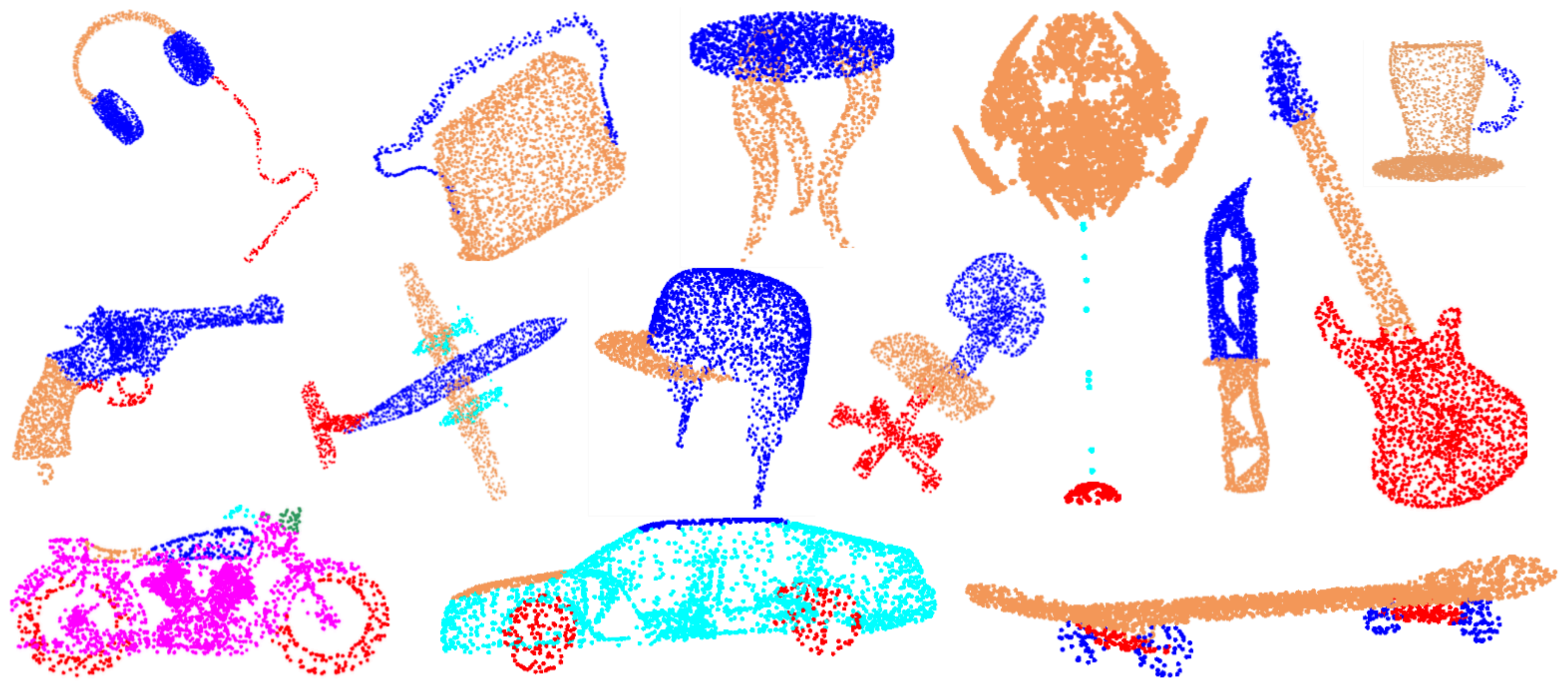
Methods	z/z	$z/\mathrm{SO}(3)$	$\mathrm{SO}(3)/\mathrm{SO}(3)$
Point / mesh inputs			
PointNet [25]	85.9	19.6	74.7
DGCNN [35]	90.3	33.8	88.6
VN-PointNet	77.5	77.5	77.2
VN-DGCNN	89.5	**89.5**	**90.2**
PCNN [2]	92.3	11.9	85.1
ShellNet [40]	**93.1**	19.9	87.8
PointNet++ [26]	91.8	28.4	85.0
PointCNN [20]	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

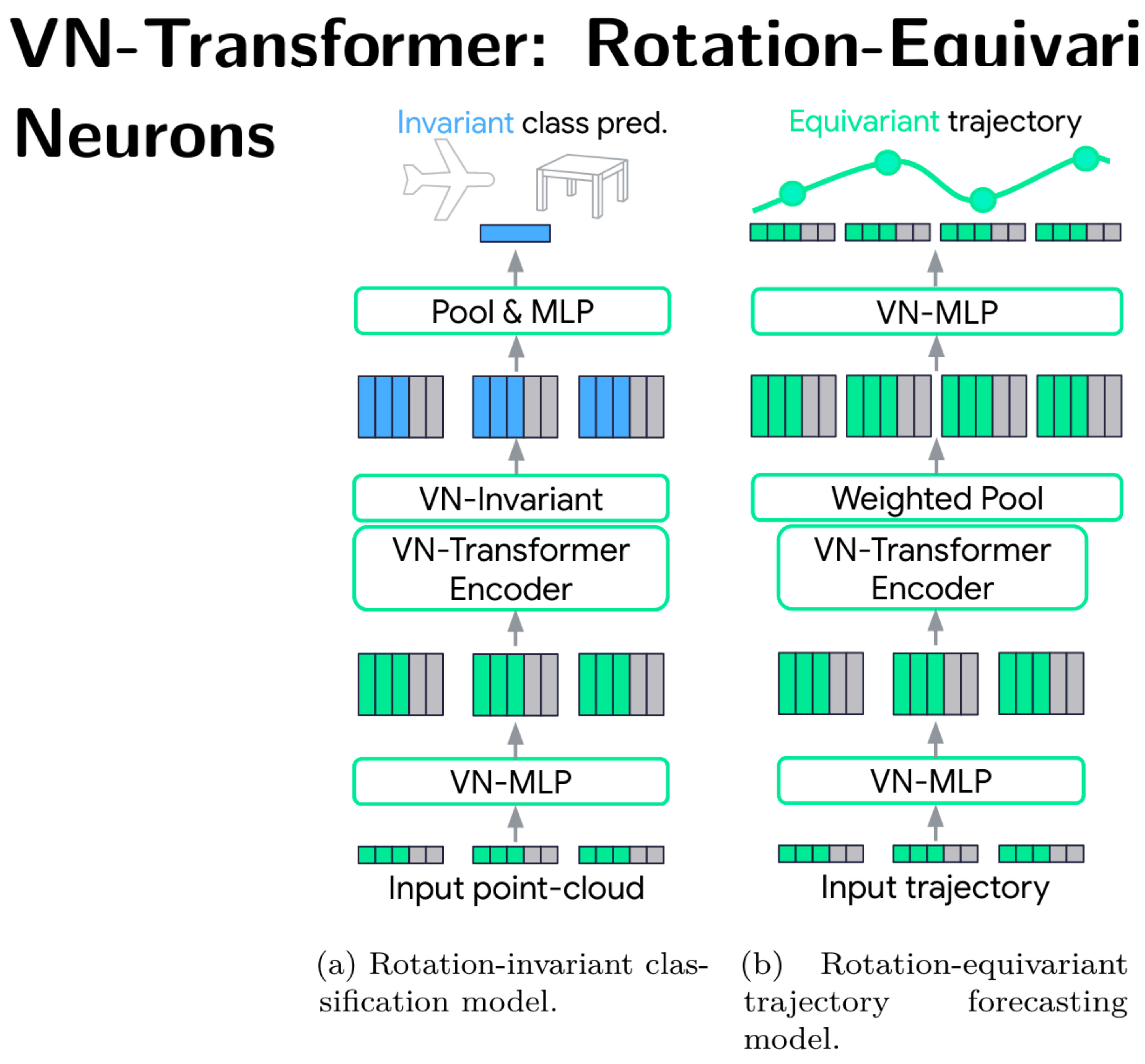
Methods	z/SO(3)	SO(3)/SO(3)
Point / mesh inputs		
PointNet [25]	38.0	62.3
DGCNN [35]	49.3	78.6
VN-PointNet	72.4	72.8
VN-DGCNN	**81.4**	**81.4**
PointCNN [20]	34.7	71.4
PointNet++ [26]	48.3	76.7
ShellNet [40]	47.2	77.1
RI-Conv [39]	75.3	75.3
TFN [32]	76.8	76.2
GC-Conv [41]	77.2	77.3
RI-Framework [18]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

Vector neuron

Transformers



VN-Transformer: Rotation-Equivariant Attention for Vector Neurons

(a) Rotation-invariant classification model.

(b) Rotation-equivariant trajectory forecasting model.

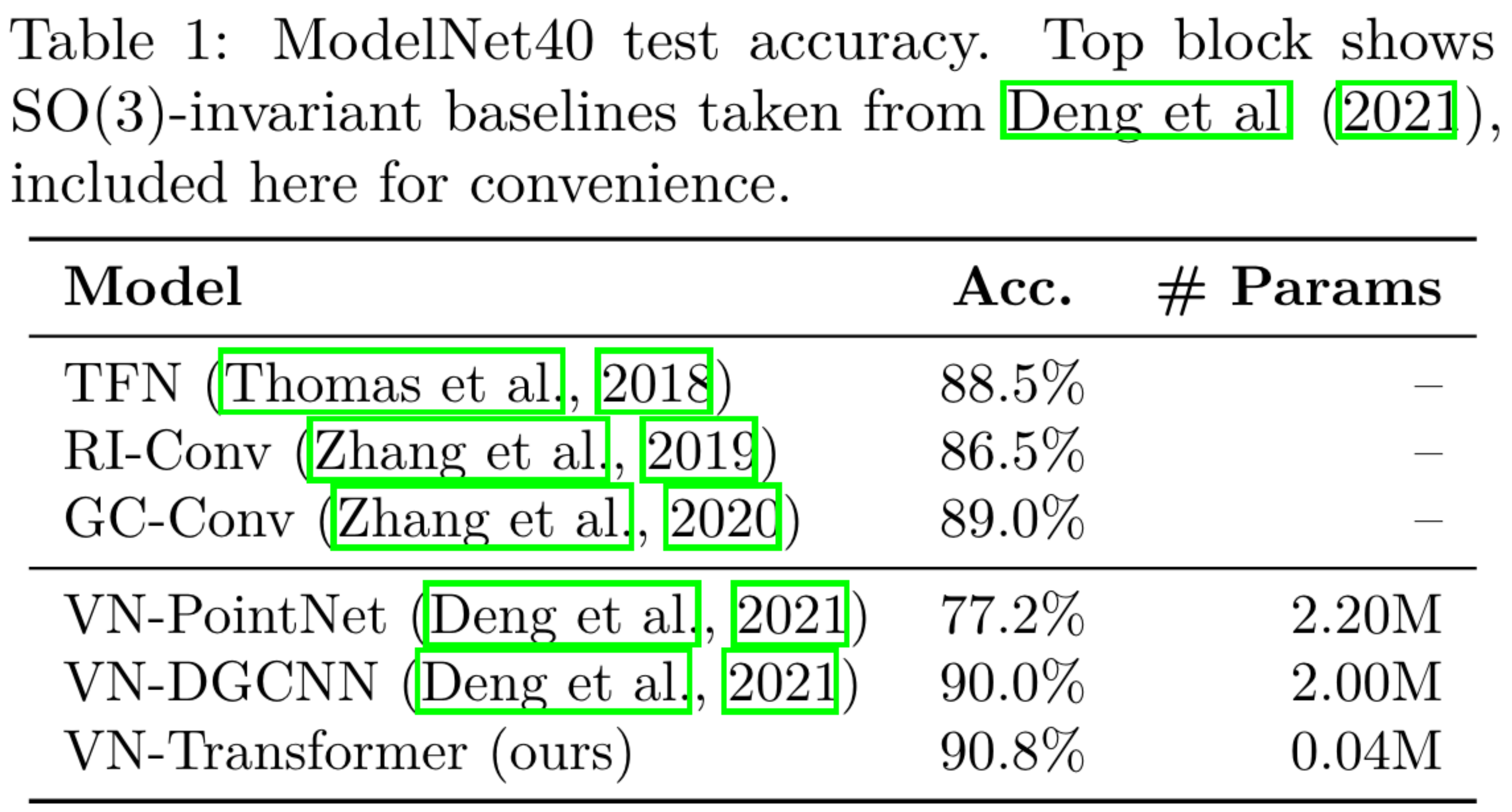
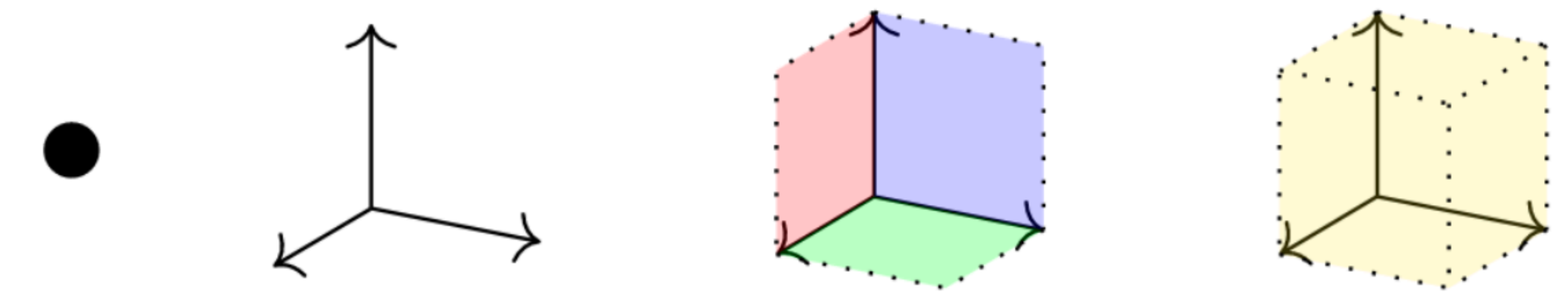


Table 1: ModelNet40 test accuracy. Top block shows SO(3)-invariant baselines taken from Deng et al (2021), included here for convenience.

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

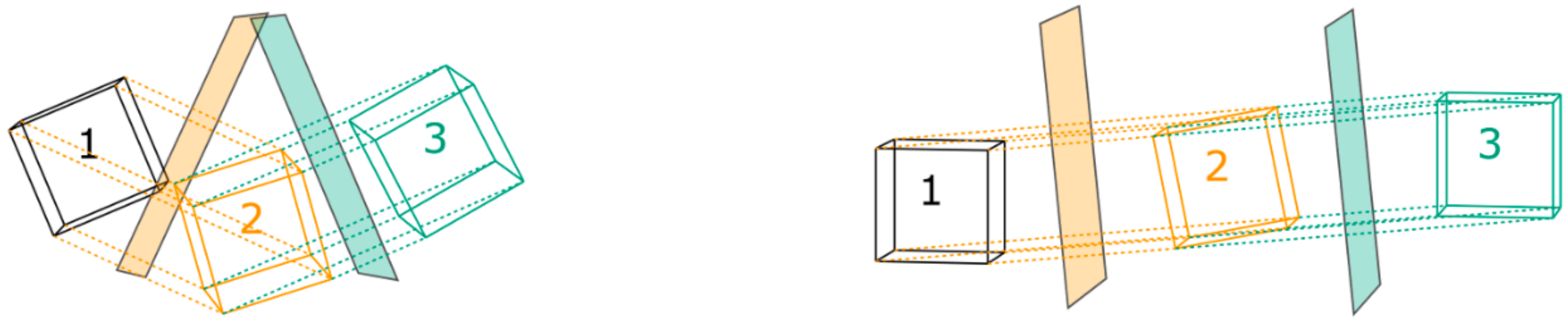


$$\underbrace{1}_{\text{Scalars}} \quad \underbrace{e_1 + e_2 + e_3}_{\text{Vectors}} \quad \underbrace{e_{12} + e_{13} + e_{23}}_{\text{Bivectors}} \quad \underbrace{e_{123}}_{\text{Trivectors}}$$

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

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

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

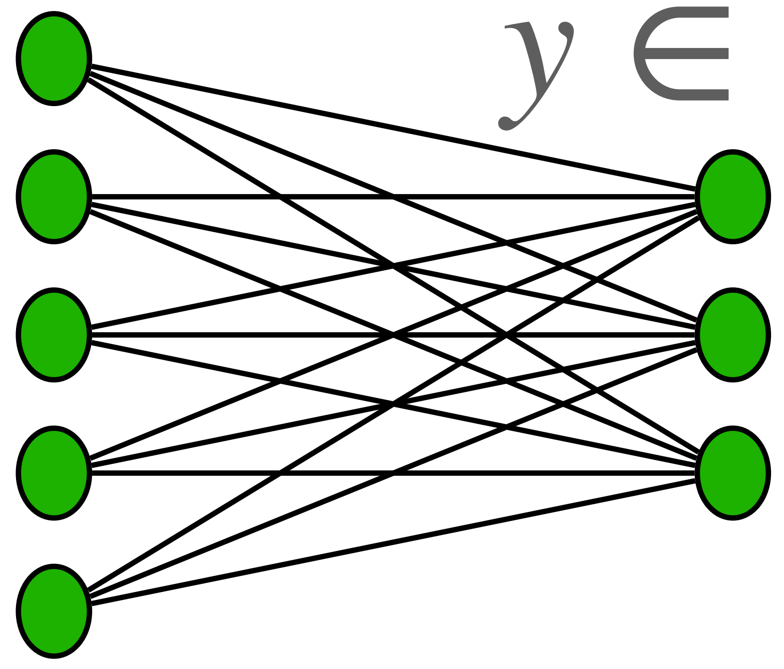


Ruhe, David, Jayesh K. Gupta, Steven De Keninck, Max Welling, and Johannes Brandstetter. "Geometric clifford algebra networks." ICML, 2023.

Geometric algebra MLP

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

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



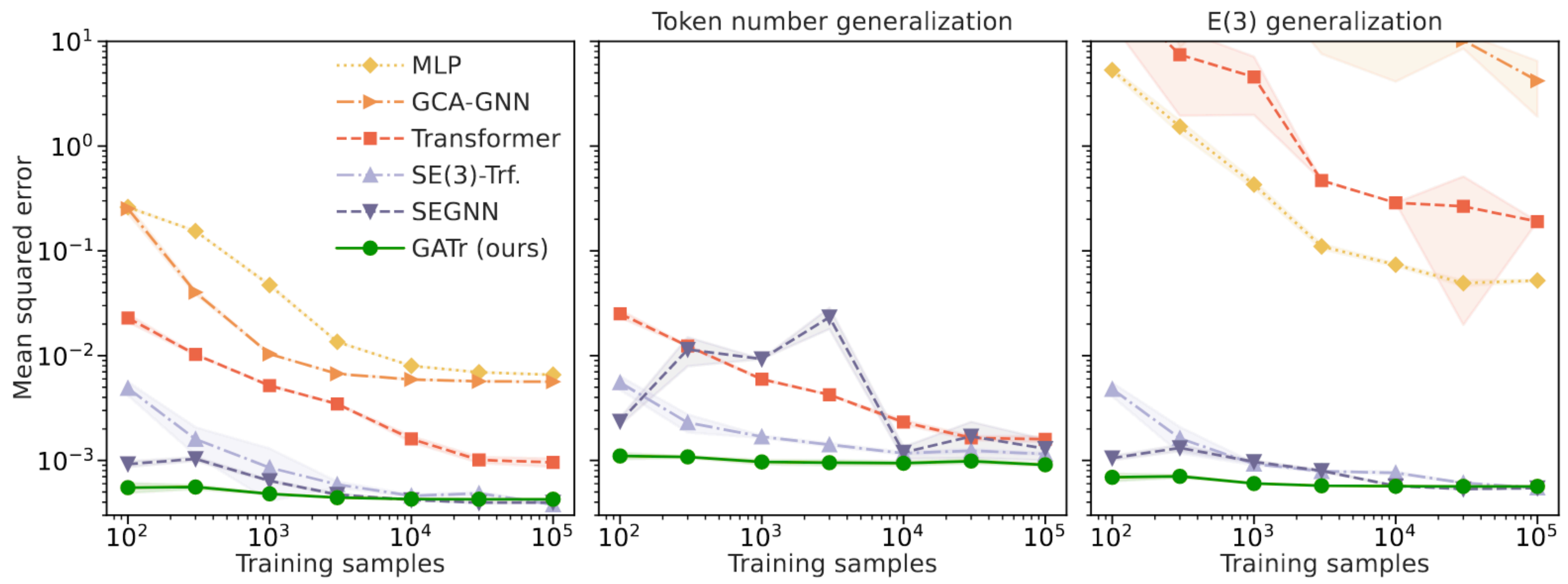
$$y^i(x) = W^{ij}x_j = (W^{ijk}\hat{e}_k)(x_j{}^l\hat{e}_l)$$
$$= W^{ijk}x^k{}_j\hat{e}_k\hat{e}_l$$
$$= W^{ijk}x^k{}_j f_{klm}\hat{e}^m$$

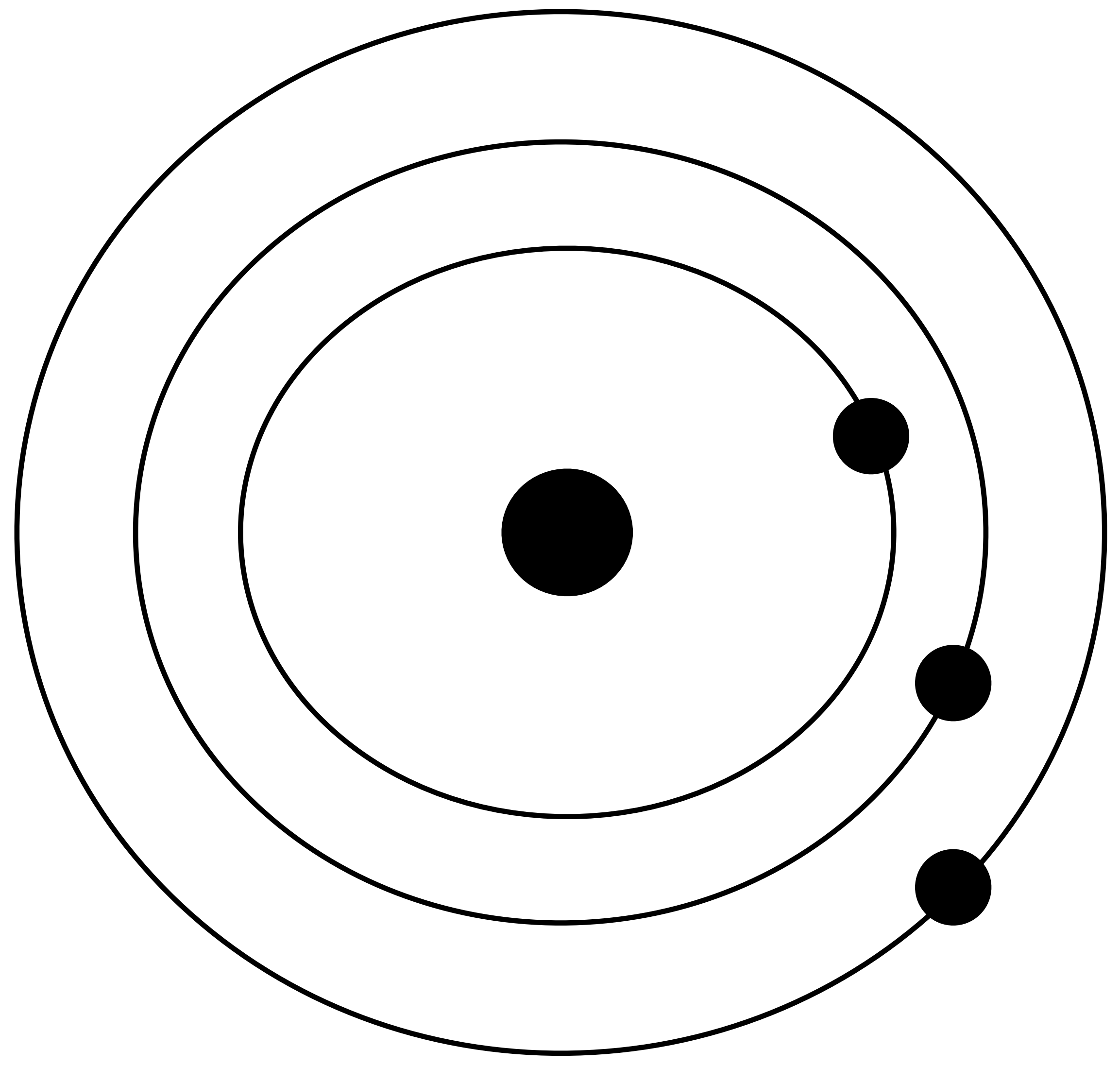
```
                         [m, n, 8]   [8, 8, 8]
einsum(“ijk,kj,klm”, W, x, f)
                              [m, 8]
```

Geometric algebra transformer

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

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

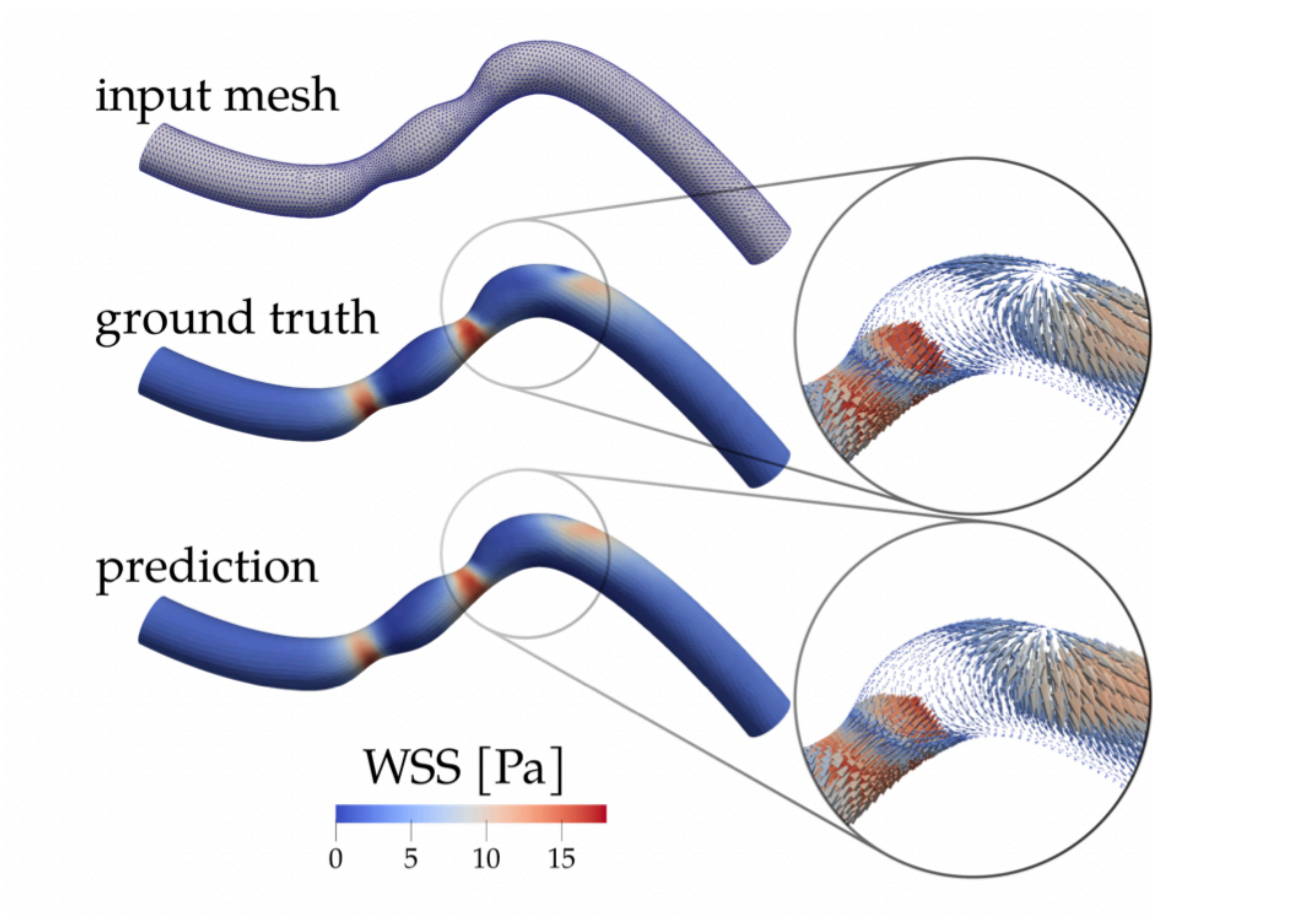


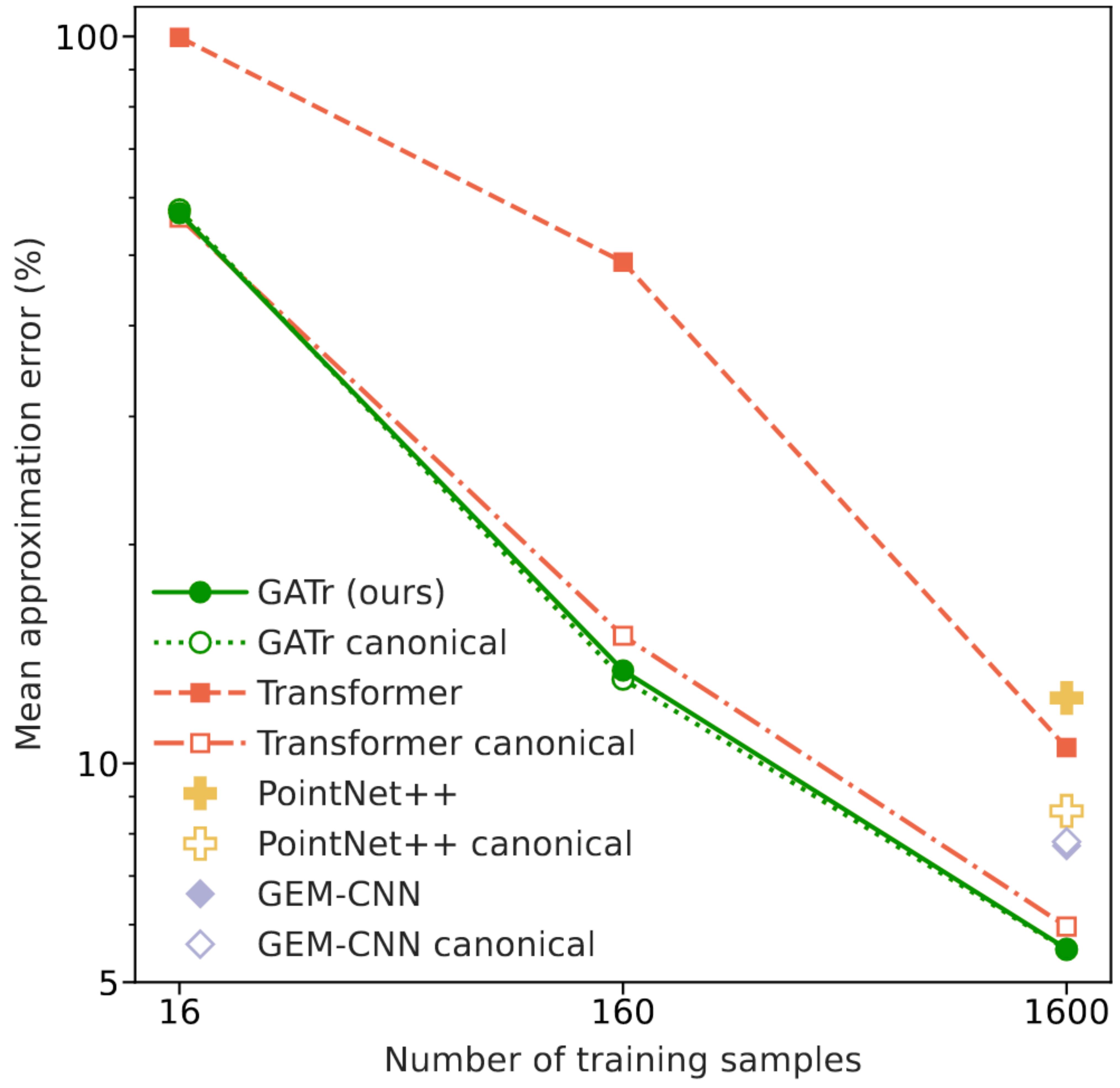


Brehmer, Johann, Pim de Haan, Sönke Behrends, and Taco Cohen. “Geometric Algebra Transformer,” NeurIPS 2023

Geometric algebra transformer

Arterial wall shear stress

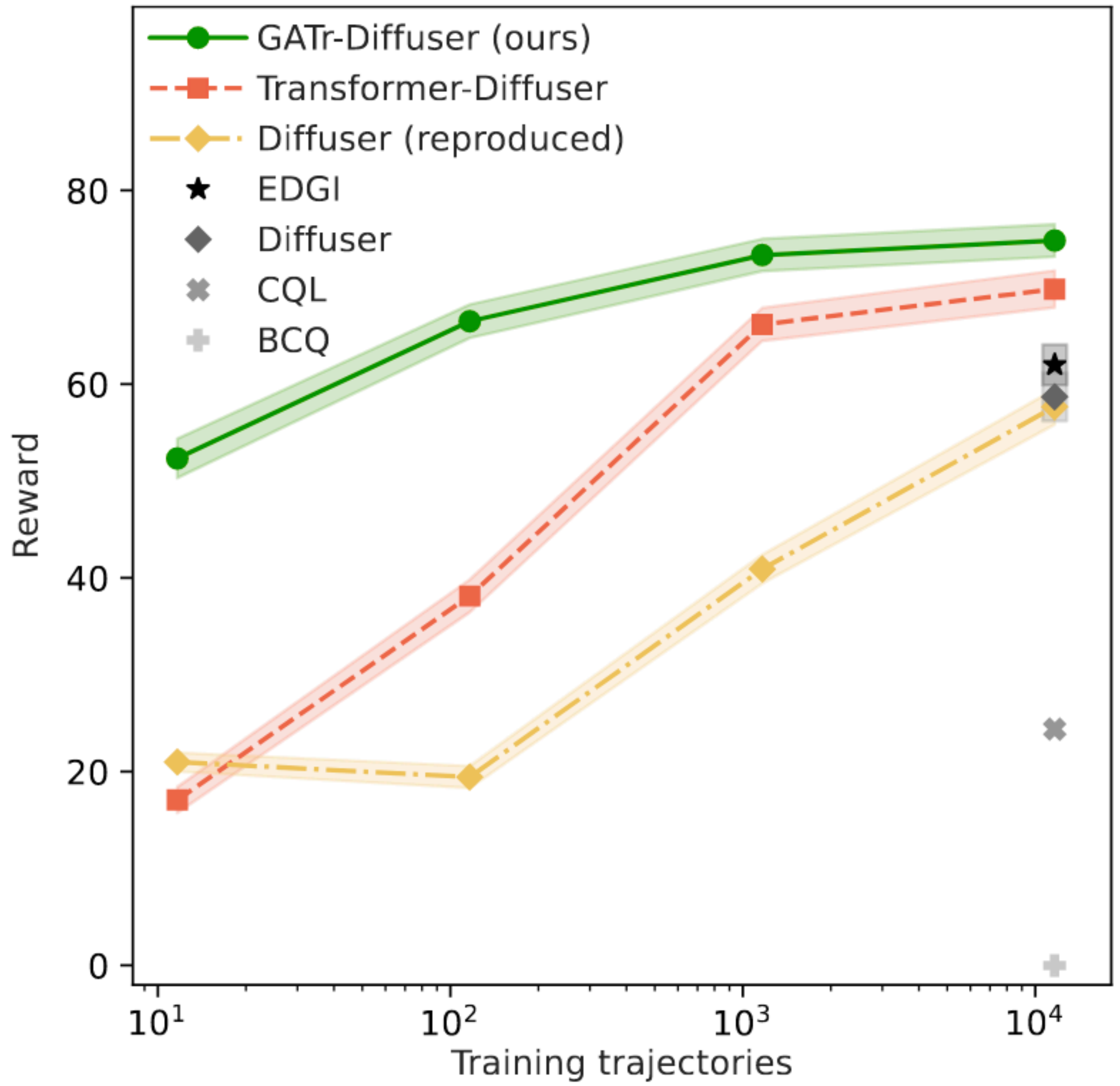


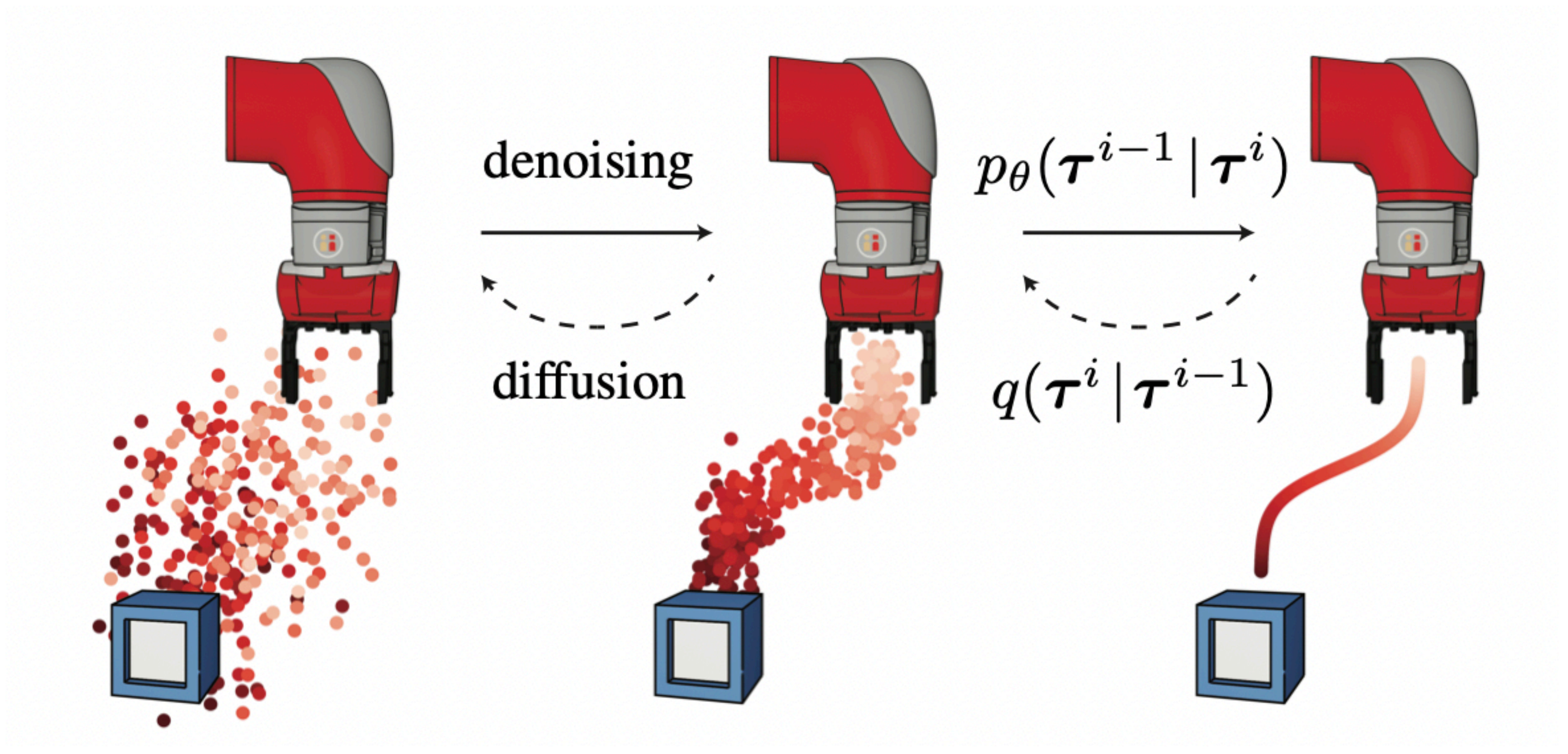


Brehmer, Johann, Pim de Haan, Sönke Behrends, and Taco Cohen. “Geometric Algebra Transformer,” NeurIPS 2023

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



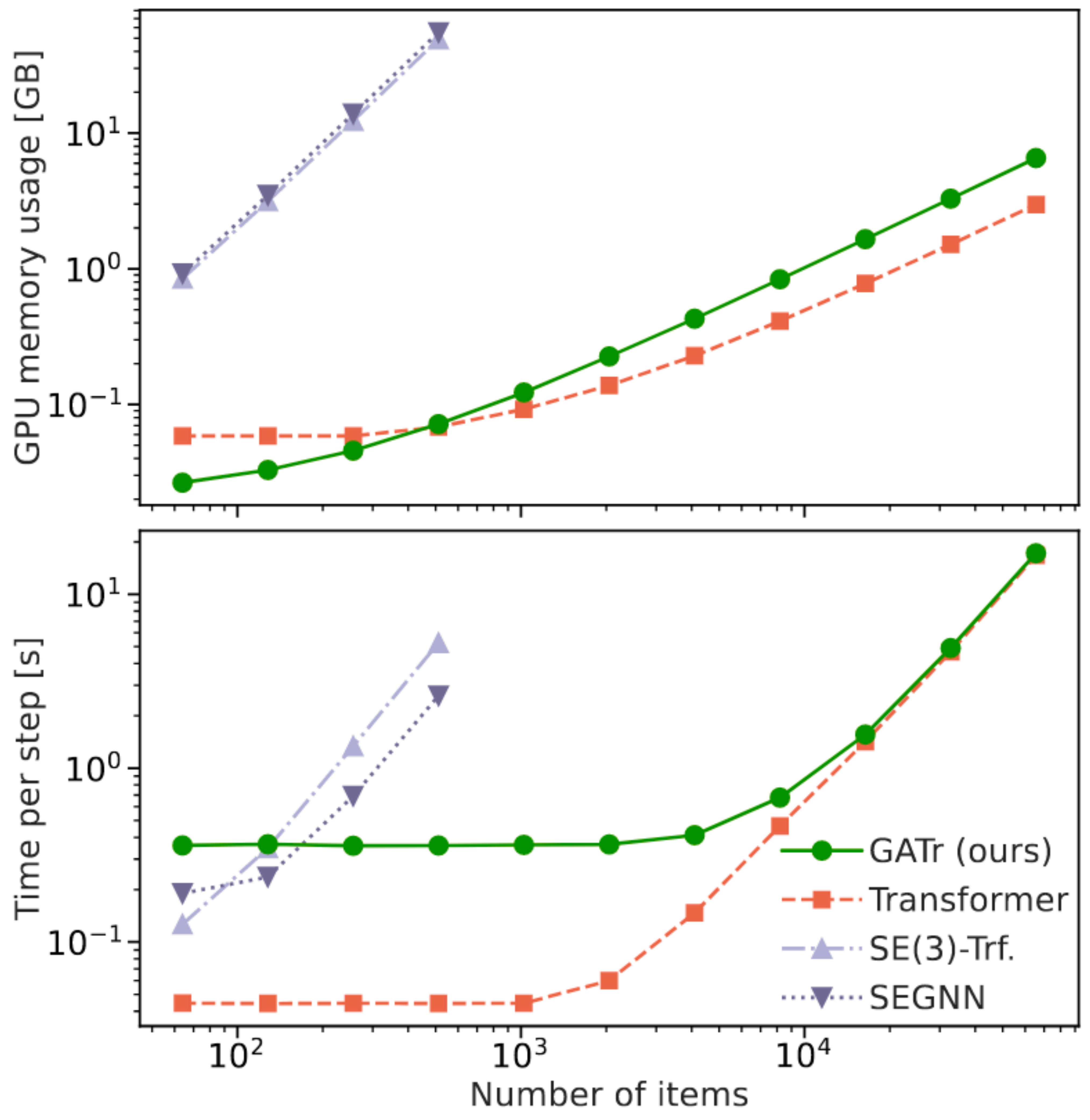

denoising
diffusion
$p_\theta(\boldsymbol{\tau}^{i-1} \mid \boldsymbol{\tau}^{i})$
$q(\boldsymbol{\tau}^{i} \mid \boldsymbol{\tau}^{i-1})$

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

Graph neural networks

E(n) Equivariant Graph Neural Networks

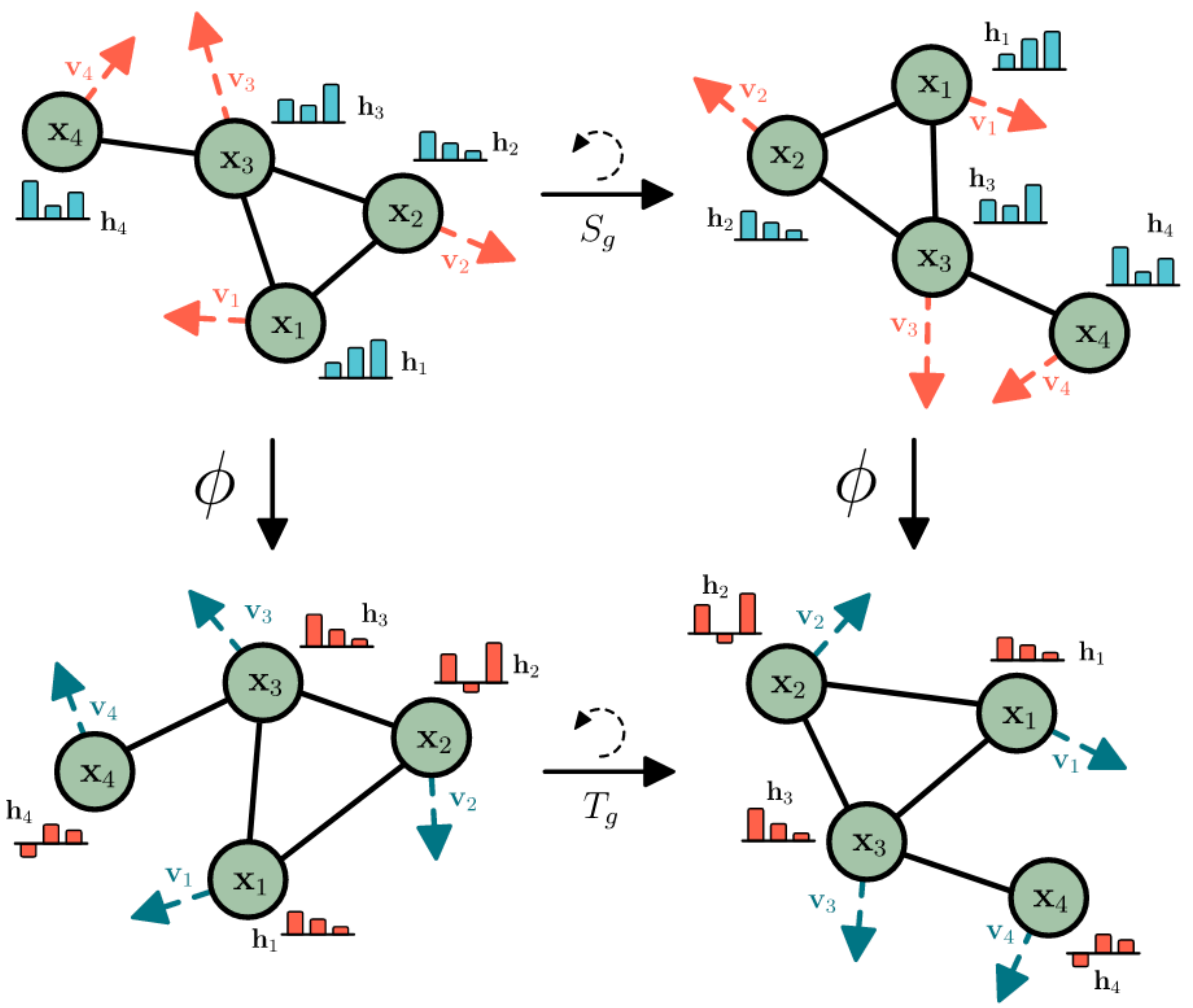
Victor Garcia Satorras[1] Emiel Hoogeboom[1] Max Welling[1]

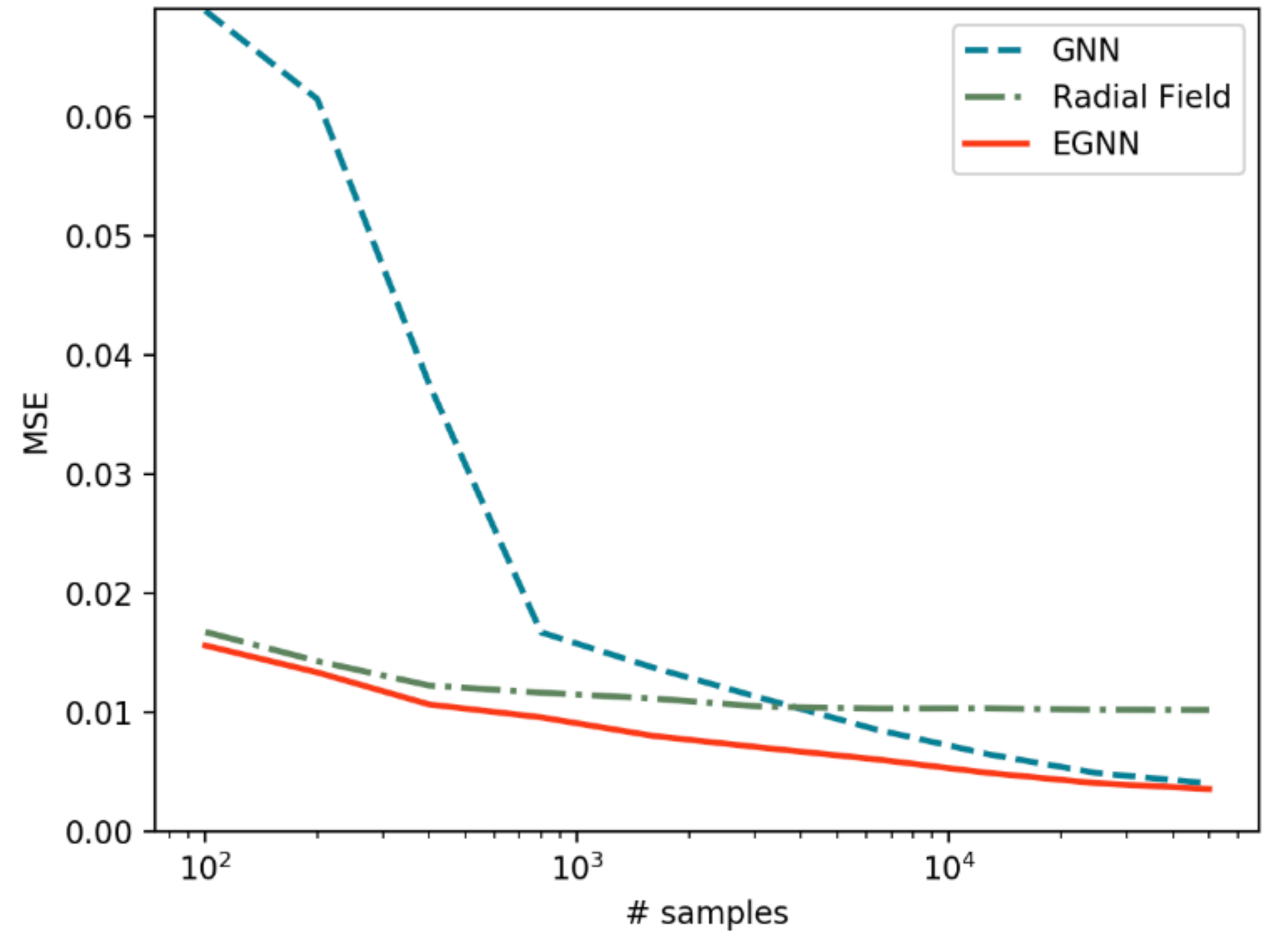
$$\mathbf{m}_{ij} = \phi_e\left(\mathbf{h}_i^l, \mathbf{h}_j^l, \left\|\mathbf{x}_i^l - \mathbf{x}_j^l\right\|^2, a_{ij}\right)$$

$$\mathbf{x}_i^{l+1} = \mathbf{x}_i^l + C\sum_{j\neq i}\left(\mathbf{x}_i^l - \mathbf{x}_j^l\right)\phi_x\left(\mathbf{m}_{ij}\right)$$

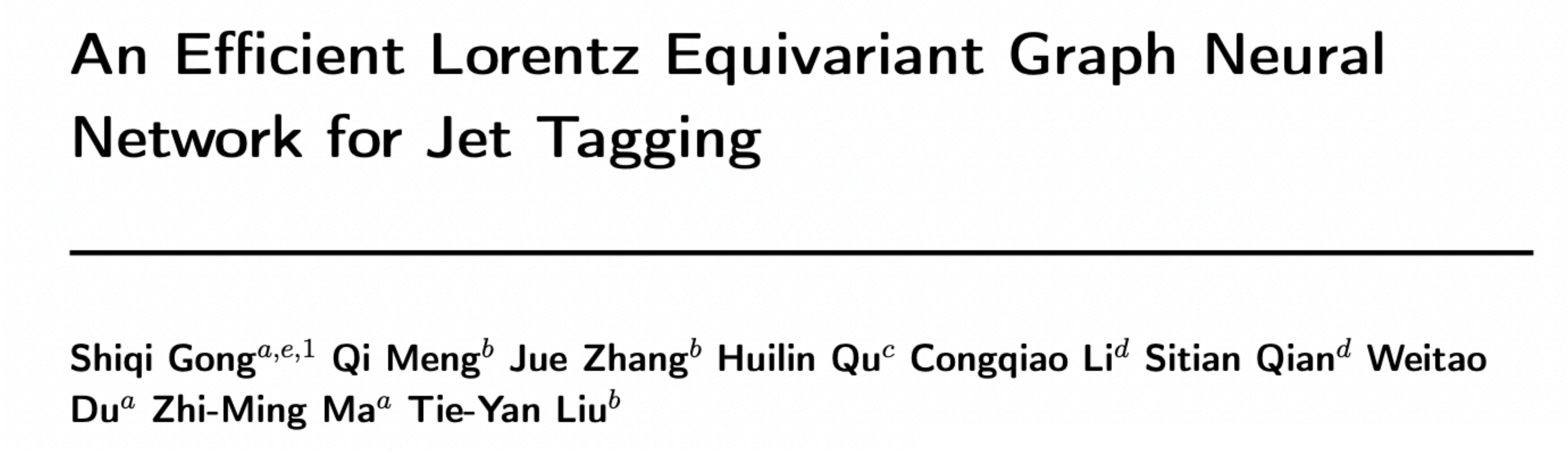
$$\mathbf{m}_i = \sum_{j\neq i}\mathbf{m}_{ij}$$

$$\mathbf{h}_i^{l+1} = \phi_h\left(\mathbf{h}_i^l, \mathbf{m}_i\right)$$





Graph neural networks



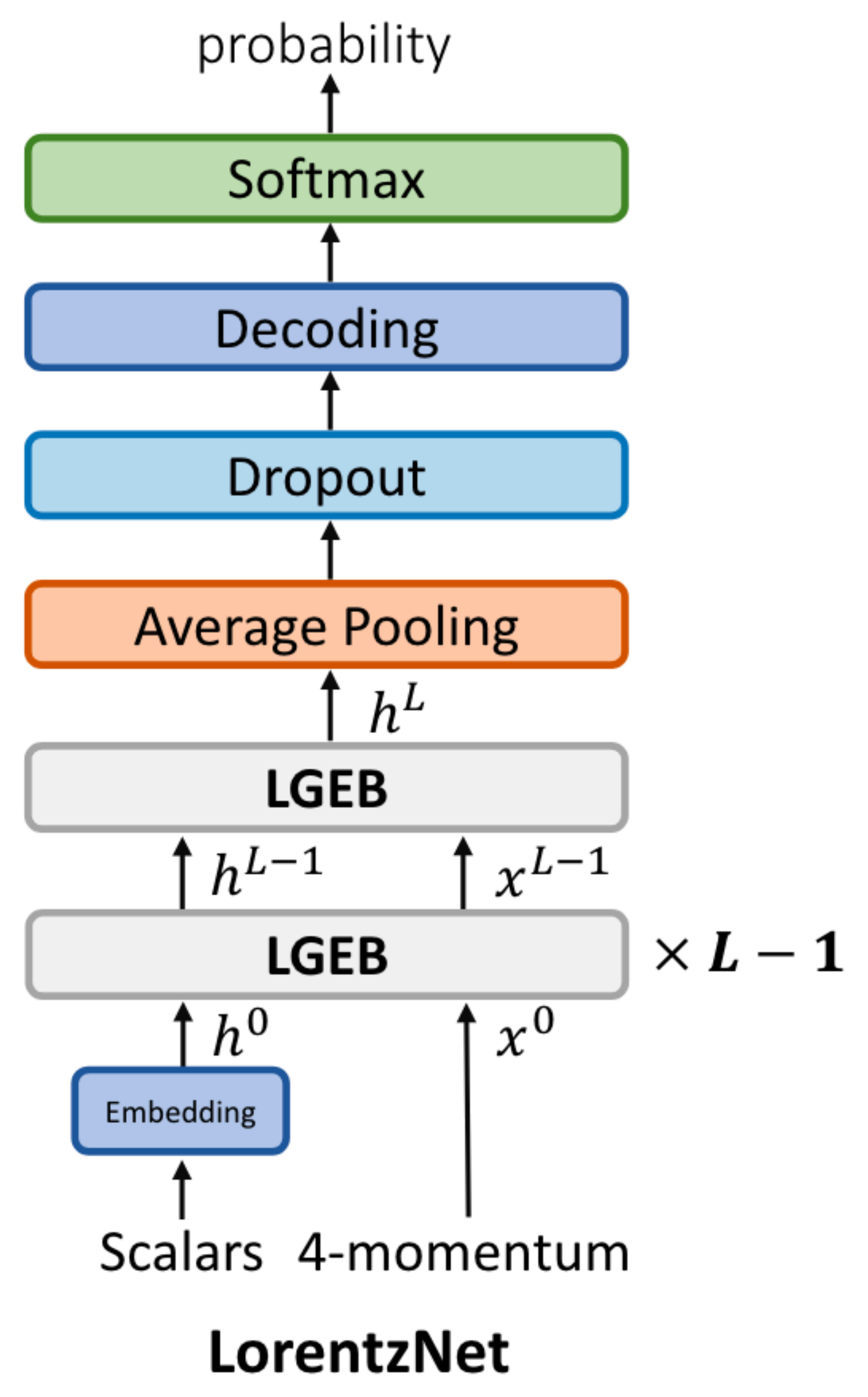
An Efficient Lorentz Equivariant Graph Neural Network for Jet Tagging

Shiqi Gong[a,e,1] Qi Meng[b] Jue Zhang[b] Huilin Qu[c] Congqiao Li[d] Sitian Qian[d] Weitao Du[a] Zhi-Ming Ma[a] Tie-Yan Liu[b]

Proposition 3.1. *[64] A continuous function $\phi : (\mathbb{R}^{N\times 4}) \to \mathbb{R}^4$ is Lorentz-equivariant if and only if*

$$\phi(v_1, v_2, \cdots, v_N) = \sum_{i=1}^{N} g_i(\langle v_i, v_j\rangle_{i,j=1}^{N}) v_i, \qquad (3.1)$$

where g_i are continuous Lorentz-invariant scalar functions, and $\langle\cdot,\cdot\rangle$ is the Minkowski inner product.



$$m_{ij}^l = \phi_e\left(h_i^l, h_j^l, \psi(\|x_i^l - x_j^l\|^2), \psi(\langle x_i^l, x_j^l\rangle)\right)$$

$$x_i^{l+1} = x_i^l + c\sum_{j\in[N]} \phi_x(m_{ij}^l)\cdot x_j^l \qquad h_i^{l+1} = h_i^l + \phi_h(h_i^l, \sum_{j\in[N]} w_{ij} m_{ij}^l)$$

Graph neural networks

Top tagging

Model	Accuracy	AUC	$1/\varepsilon_B$ $(\varepsilon_S = 0.5)$	$1/\varepsilon_B$ $(\varepsilon_S = 0.3)$
ResNeXt	0.936	0.9837	302 ± 5	1147 ± 58
P-CNN	0.930	0.9803	201 ± 4	759 ± 24
PFN	0.932	0.9819	247 ± 3	888 ± 17
ParticleNet	0.940	0.9858	397 ± 7	1615 ± 93
EGNN	0.922	0.9760	148 ± 8	540 ± 49
LGN	0.929	0.9640	124 ± 20	435 ± 95
LorentzNet	**0.942**	**0.9868**	$\mathbf{498 \pm 18}$	$\mathbf{2195 \pm 173}$

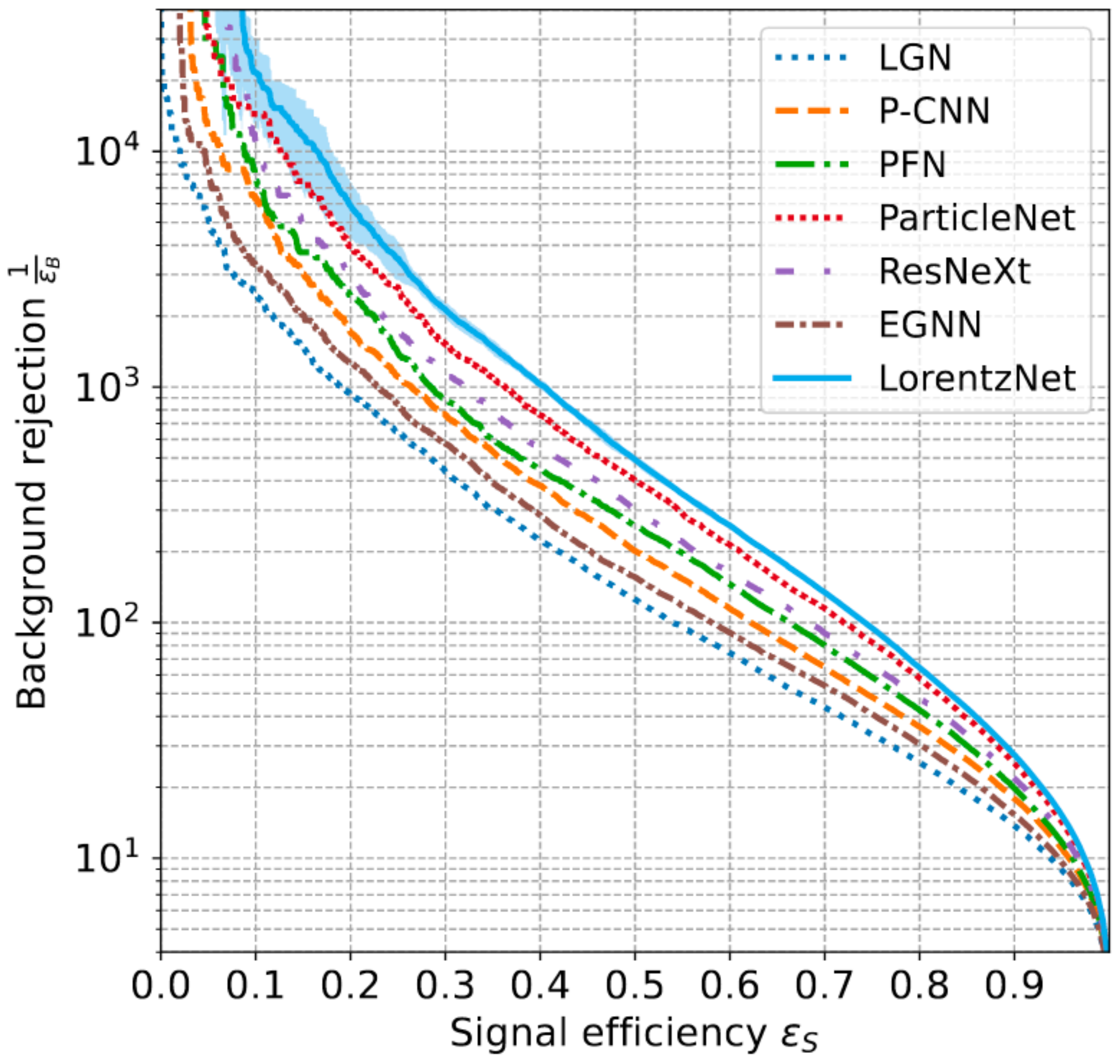
Compute

Model	Equivariance	Time on CPU (ms/batch)	Time on GPU (ms/batch)	#Params
ResNeXt	✗	5.5	0.34	1.46M
P-CNN	✗	**0.6**	**0.11**	348k
PFN	✗	**0.6**	0.12	82k
ParticleNet	✗	11.0	0.19	366k
EGNN	E(4)	30.0	0.30	222k
LGN	$SO^+(1,3)$	51.4	1.66	4.5k
LorentzNet	$SO^+(1,3)$	32.9	0.34	224k

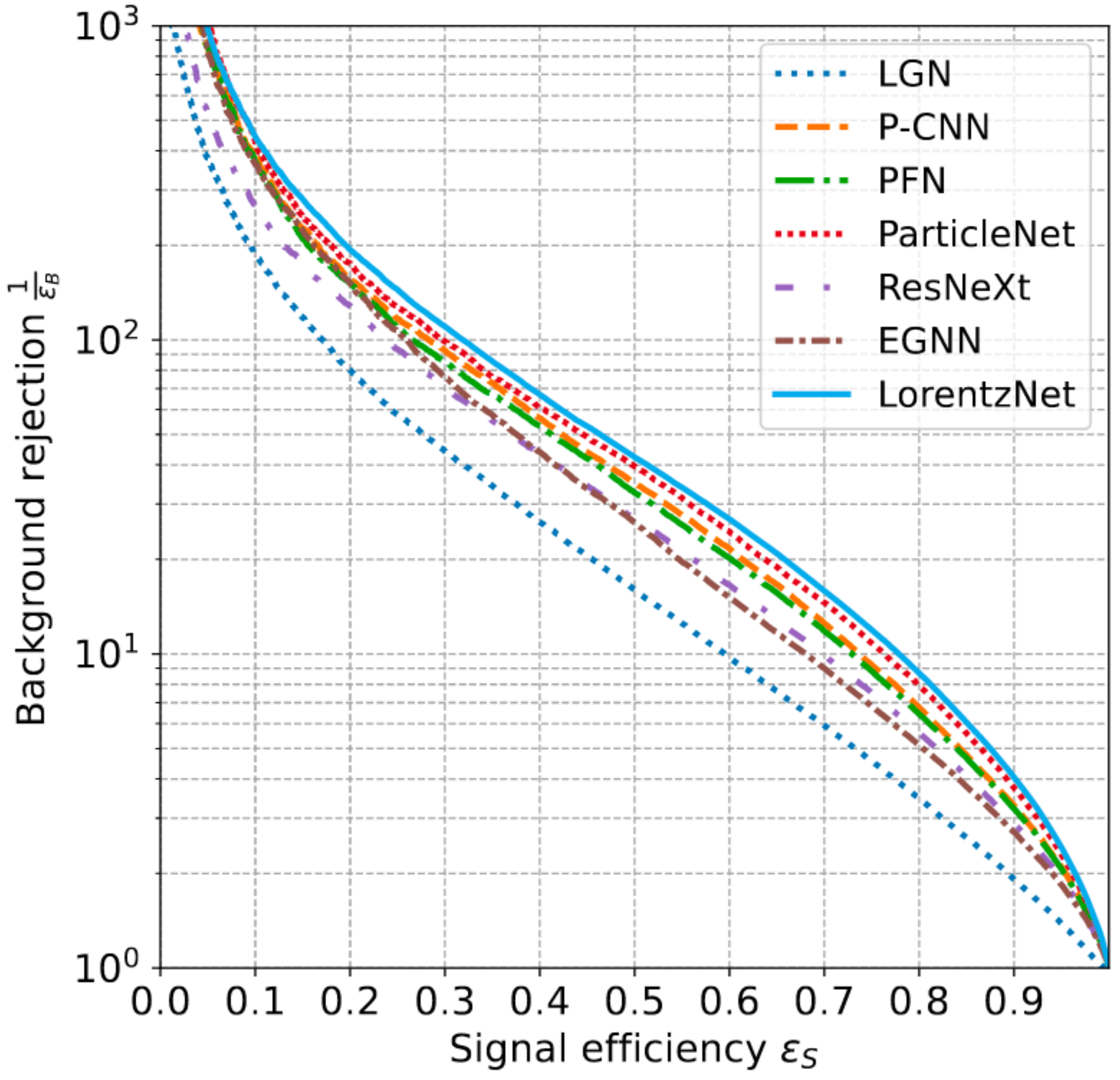
Gong, Shiqi, Qi Meng, Jue Zhang, Huilin Qu, Congqiao Li, Sitian Qian, Weitao Du, Zhi-Ming Ma, and Tie-Yan Liu. "An Efficient Lorentz Equivariant Graph Neural Network for Jet Tagging." *Journal of High Energy Physics* 2022

Graph neural networks

Top tagging



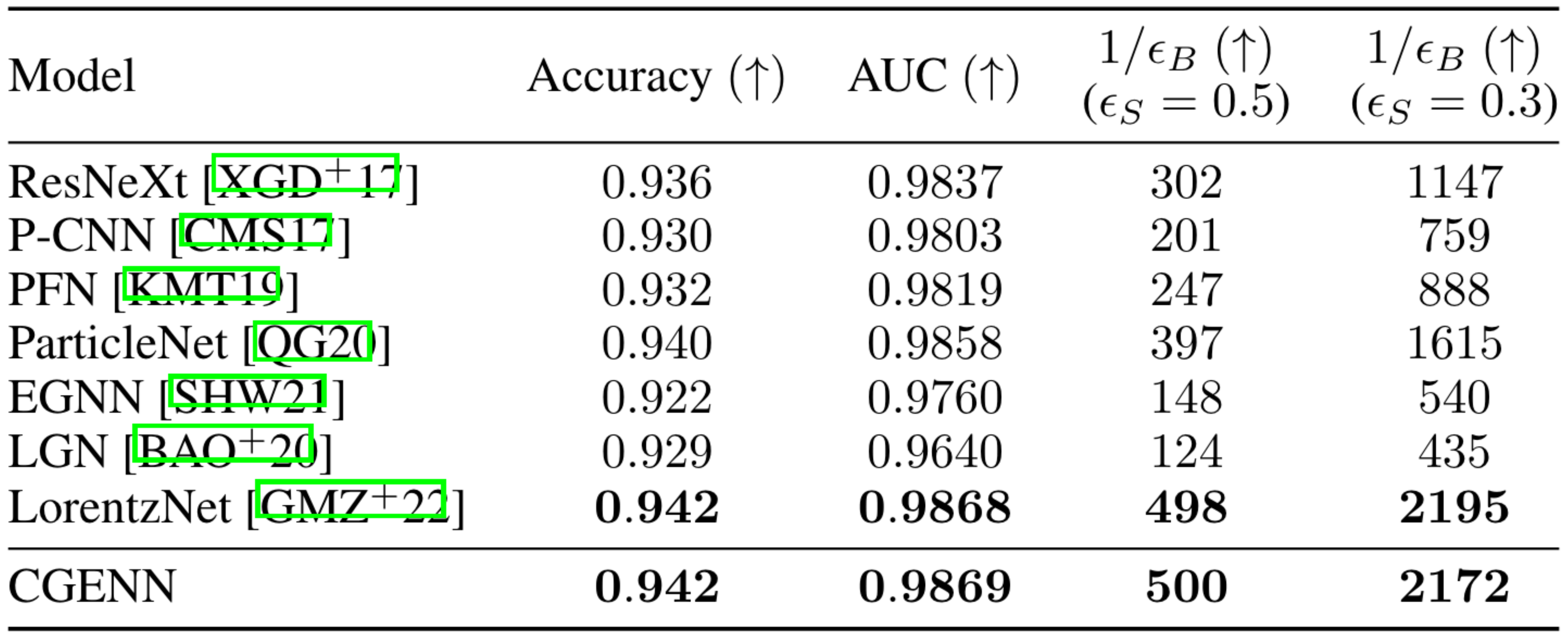
Quark gluon tagging



Gong, Shiqi, Qi Meng, Jue Zhang, Huilin Qu, Congqiao Li, Sitian Qian, Weitao Du, Zhi-Ming Ma, and Tie-Yan Liu. "An Efficient Lorentz Equivariant Graph Neural Network for Jet Tagging." *Journal of High Energy Physics* 2022

Graph neural network

More geometric algebra



Model	Accuracy (↑)	AUC (↑)	$1/\epsilon_B$ (↑) ($\epsilon_S = 0.5$)	$1/\epsilon_B$ (↑) ($\epsilon_S = 0.3$)
ResNeXt [XGD$^+$17]	0.936	0.9837	302	1147
P-CNN [CMS17]	0.930	0.9803	201	759
PFN [KMT19]	0.932	0.9819	247	888
ParticleNet [QG20]	0.940	0.9858	397	1615
EGNN [SHW21]	0.922	0.9760	148	540
LGN [BAO$^+$20]	0.929	0.9640	124	435
LorentzNet [GMZ$^+$22]	**0.942**	**0.9868**	**498**	**2195**
CGENN	**0.942**	**0.9869**	**500**	**2172**

Table 2: Performance comparison between our proposed method and alternative algorithms on the top tagging experiment. We present the accuracy, Area Under the Receiver Operating Characteristic Curve (AUC), and background rejection $1/\epsilon_B$ and at signal efficiencies of $\epsilon_S = 0.3$ and $\epsilon_S = 0.5$.

Ruhe, David, Johannes Brandstetter, and Patrick Forré. "Clifford group equivariant neural networks", NeurIPS 2024

Molecule generation

EGNN + Diffusion

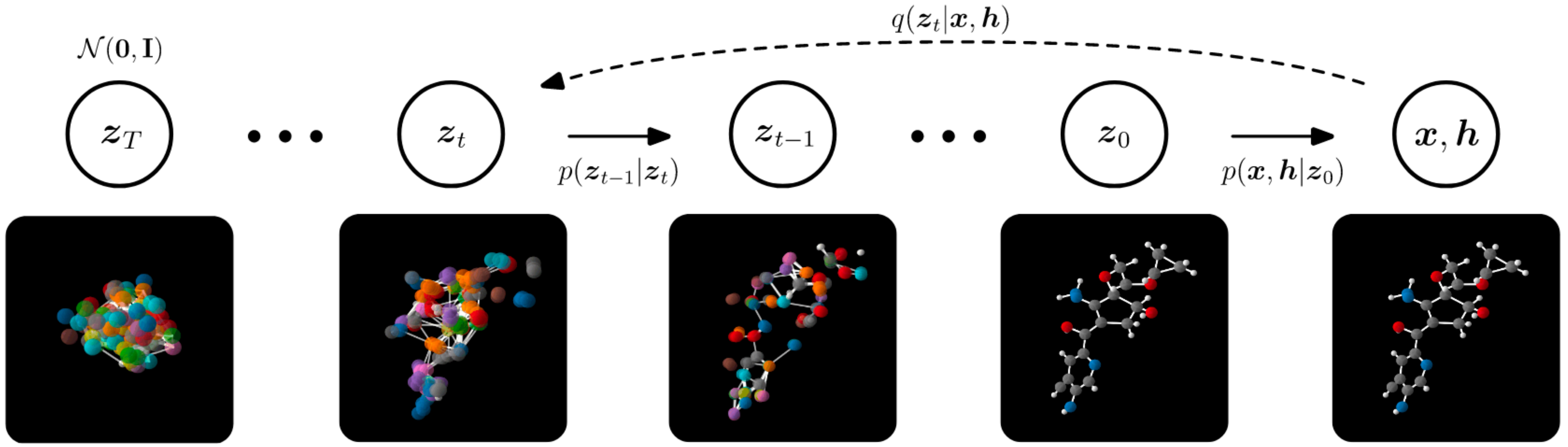

$\mathcal{N}(\mathbf{0}, \mathbf{I})$
$\boldsymbol{z}_T$
$\boldsymbol{z}_t$
$p(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t)$
$\boldsymbol{z}_{t-1}$
$q(\boldsymbol{z}_t|\boldsymbol{x}, \boldsymbol{h})$
$\boldsymbol{z}_0$
$p(\boldsymbol{x}, \boldsymbol{h}|\boldsymbol{z}_0)$
$\boldsymbol{x}, \boldsymbol{h}$

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

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

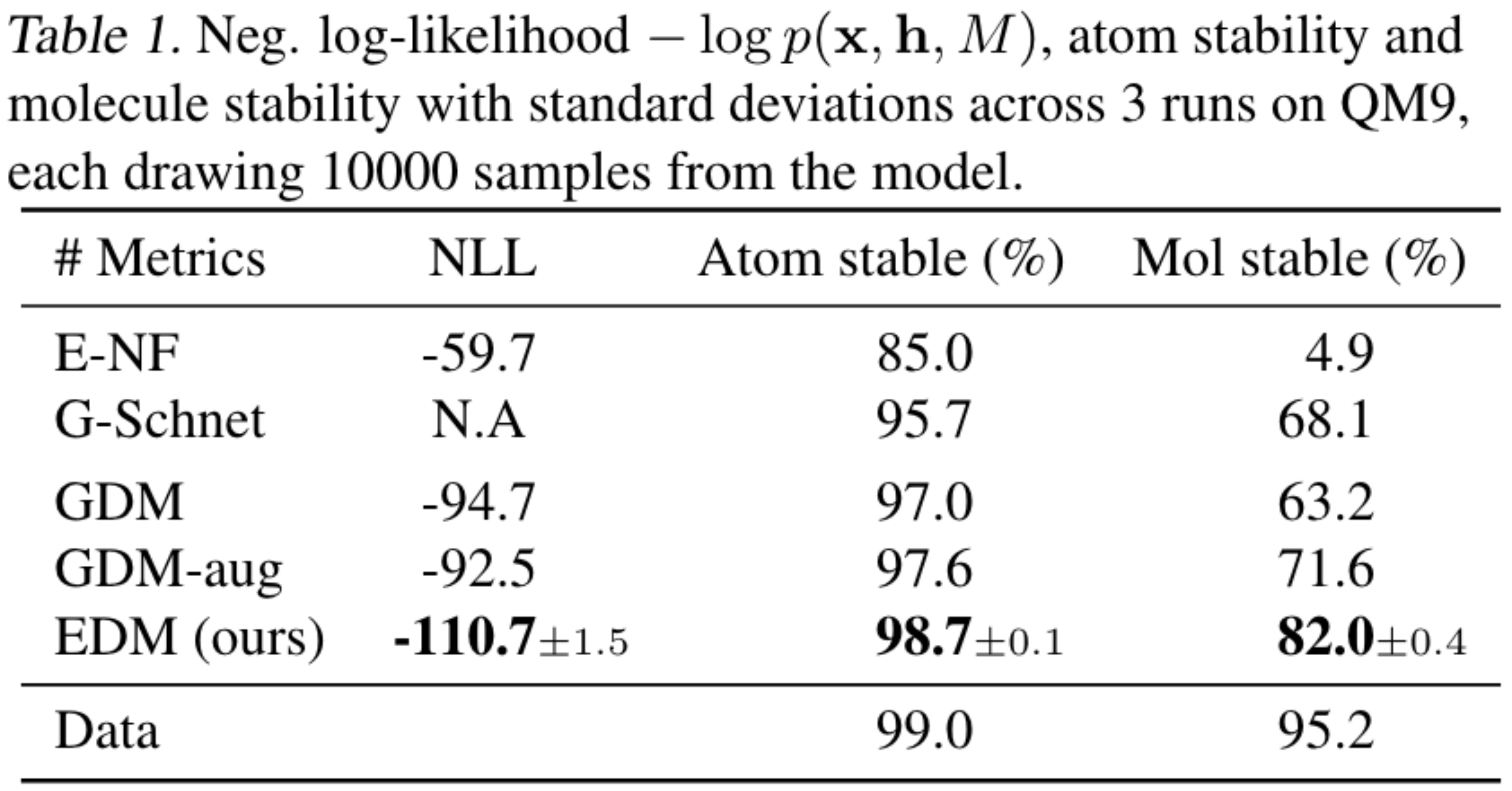
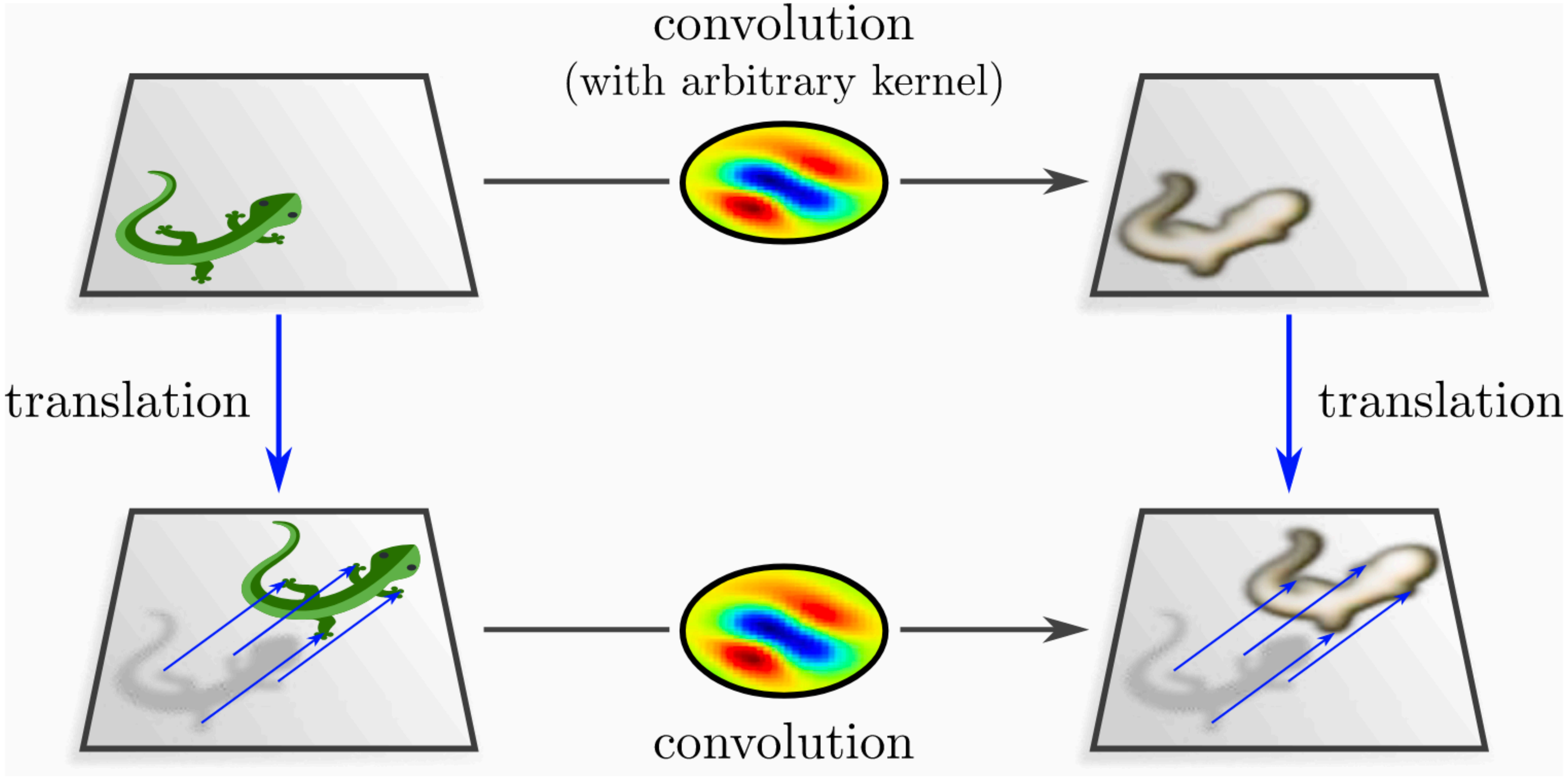


Table 1. Neg. log-likelihood $-\log p(\mathbf{x}, \mathbf{h}, M)$, atom stability and molecule stability with standard deviations across 3 runs on QM9, each drawing 10000 samples from the model.

# 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

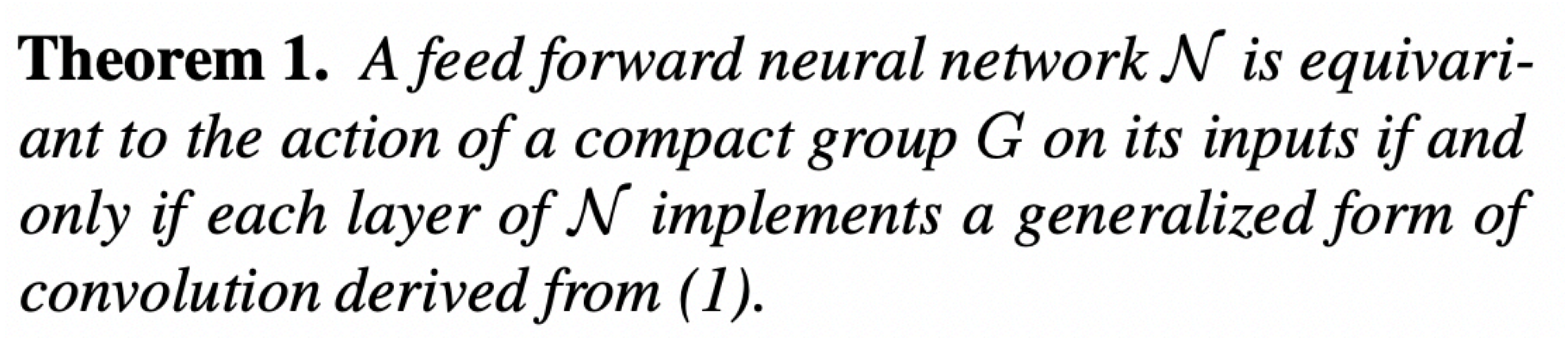
Hoogeboom, Emiel, Victor Garcia Satorras, Clement Vignac, and Max Welling. “Equivariant Diffusion for Molecule Generation in 3D.” ICML, 2022

Convolutional neural networks

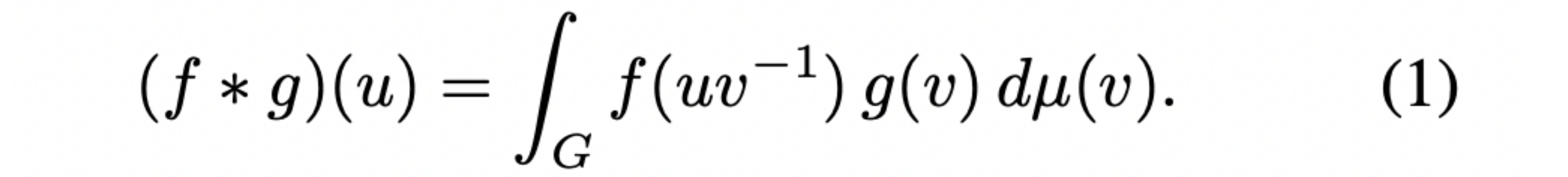


Convolutional neural networks

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).*

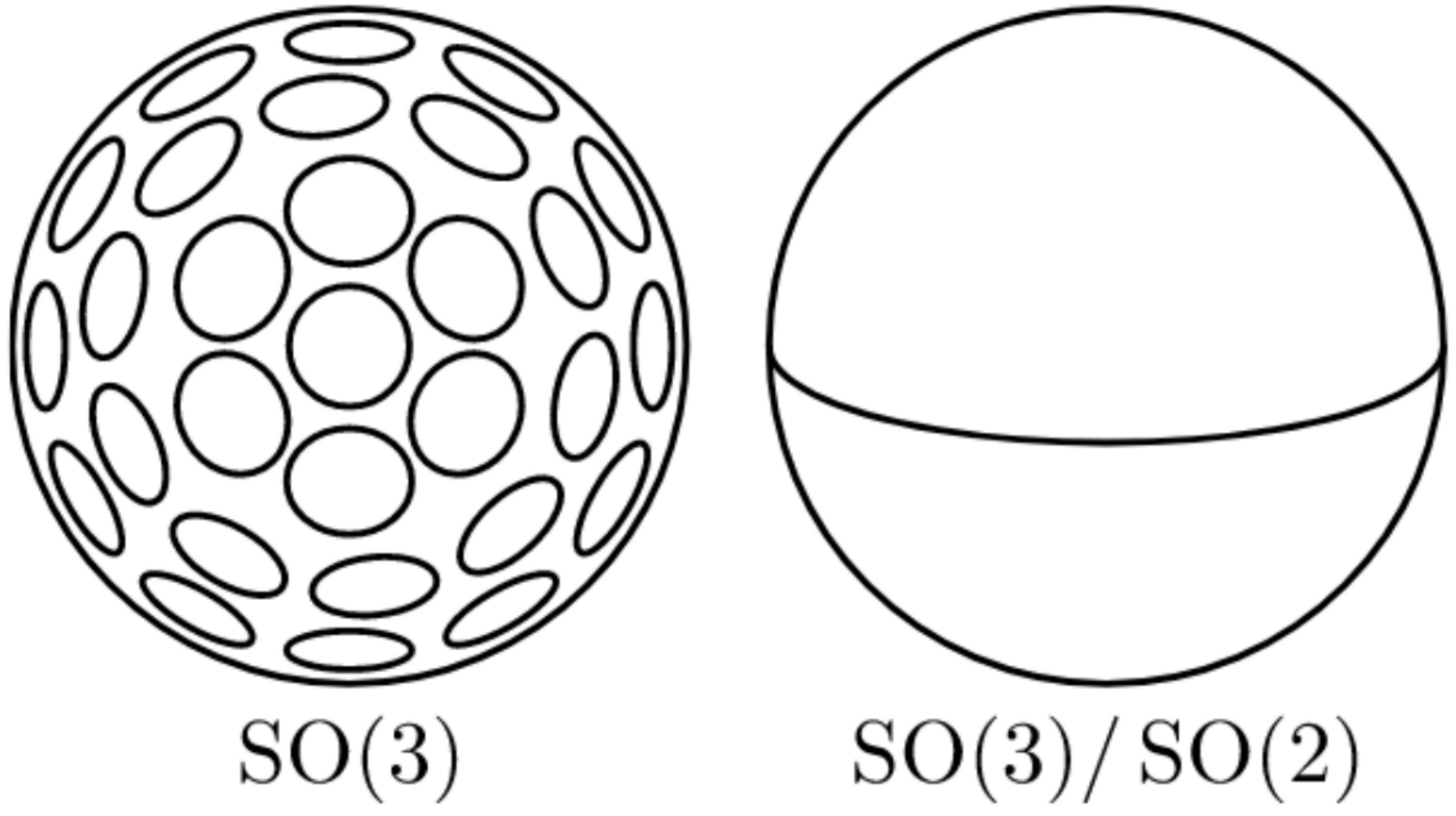


$$(f * g)(u) = \int_G f(uv^{-1})\, g(v)\, d\mu(v). \qquad (1)$$

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



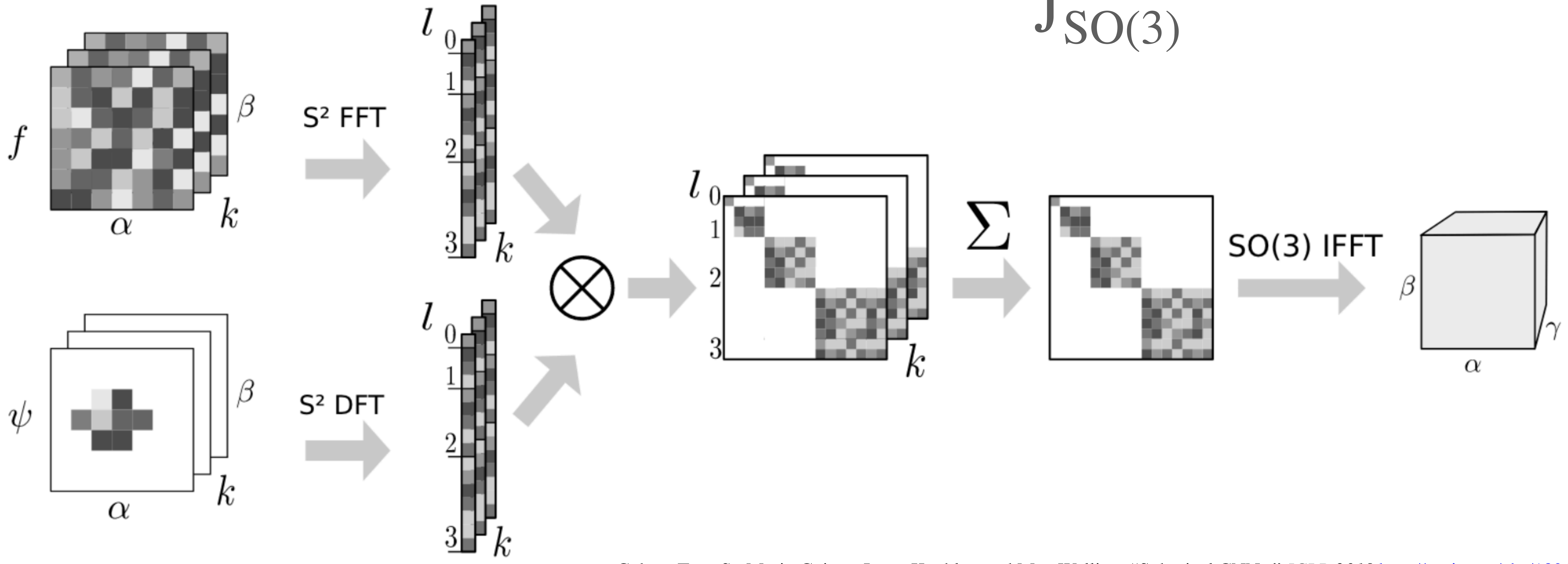
$$\psi : \mathrm{SO}(3) \to \mathbb{R}$$
$$f : \mathrm{SO}(3) \to \mathbb{R}$$

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

Group convolutions on homogeneous spaces

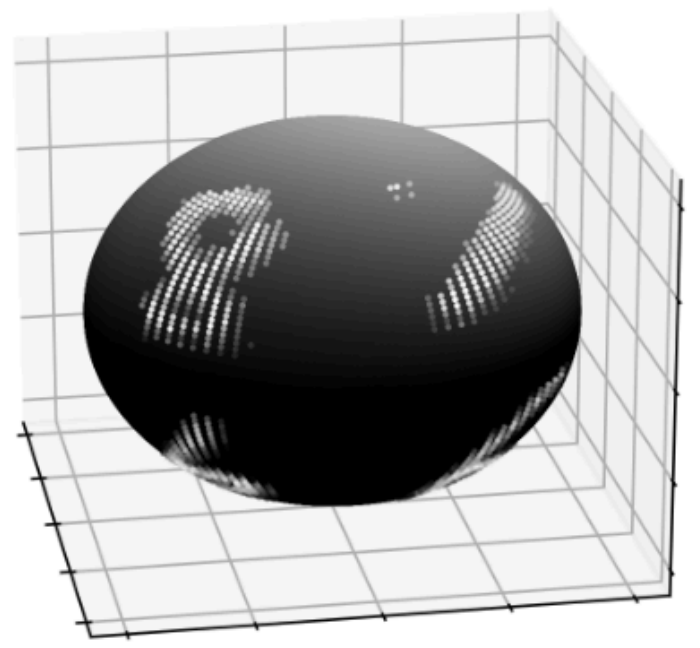
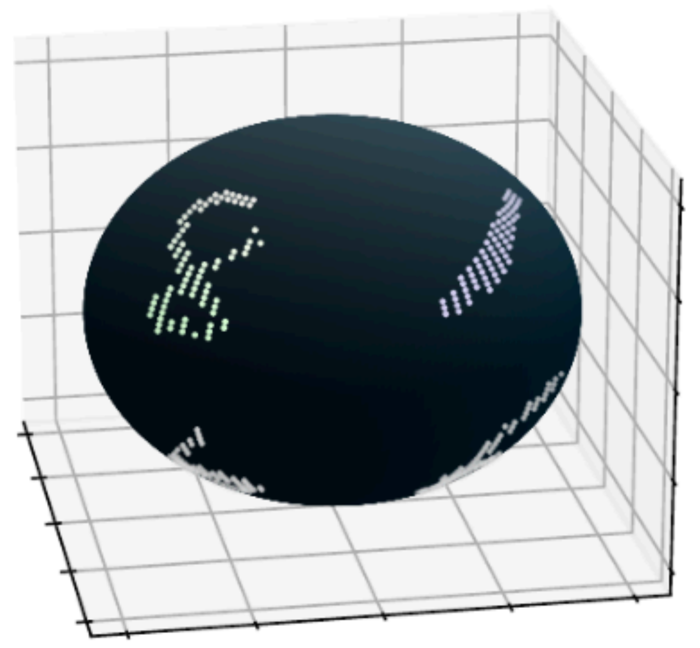
Equivariant neural networks on the sphere

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

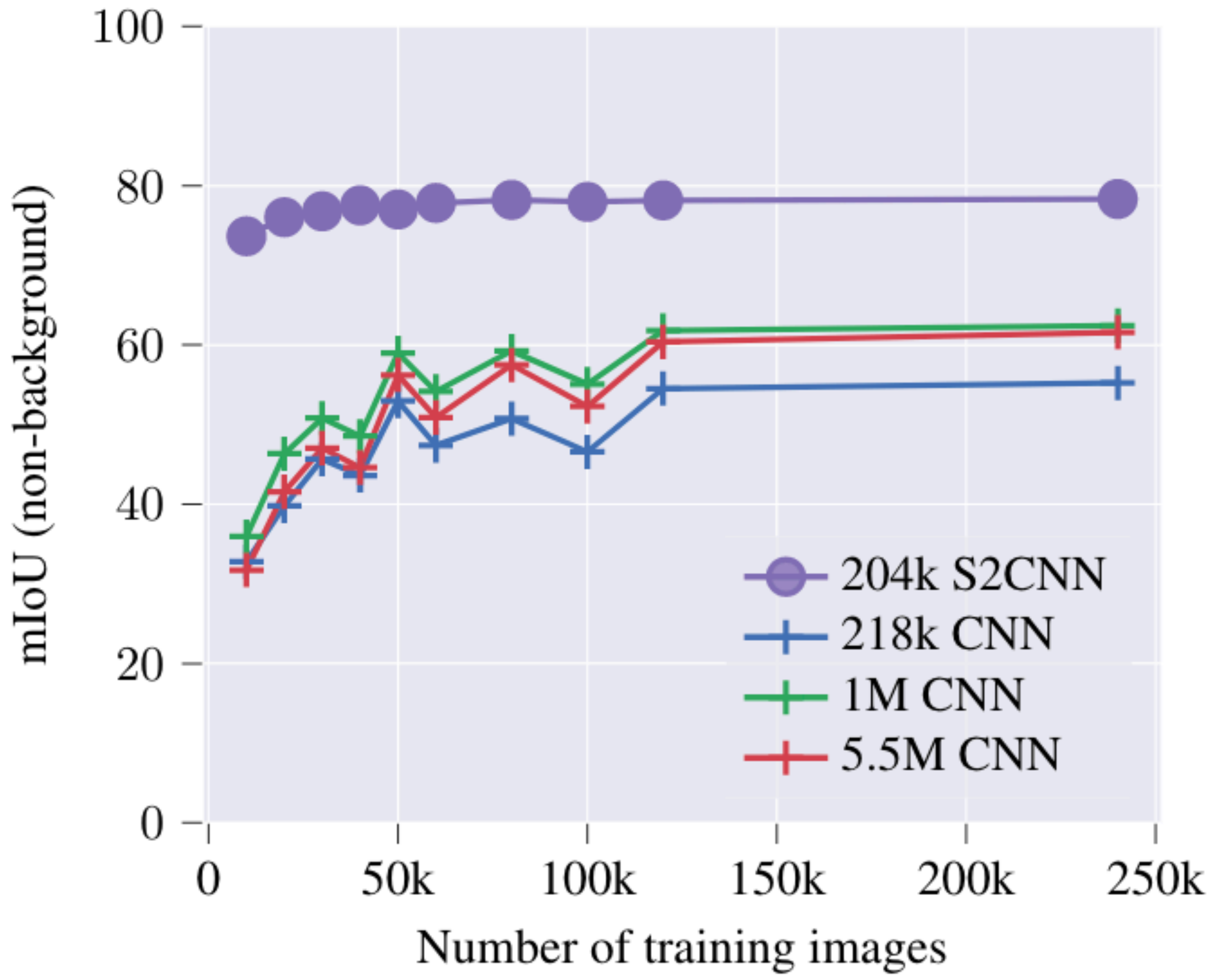


Geometric deep learning

Equivariance vs augmentation

60 x 60 Driscoll-Healy grid

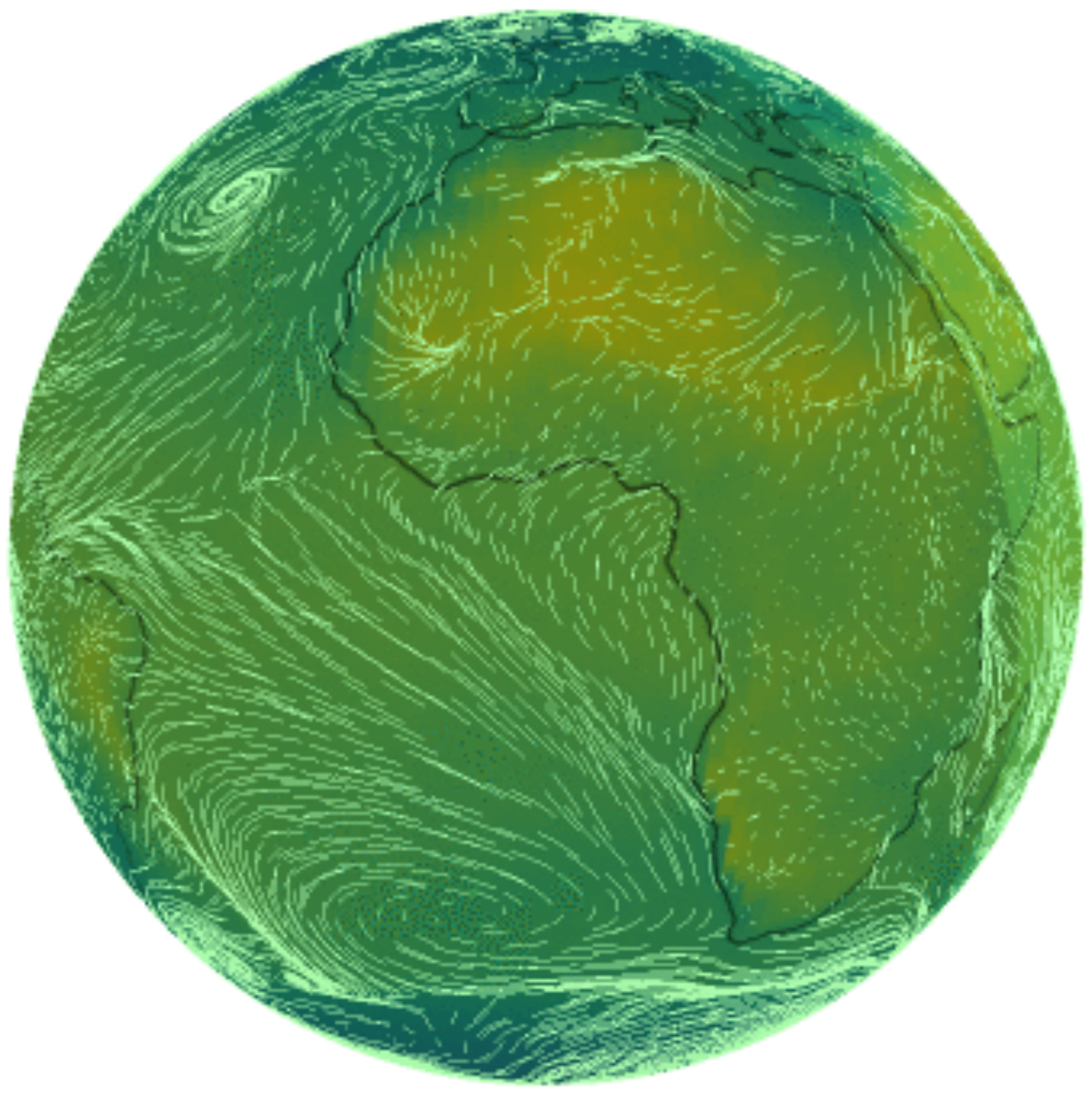


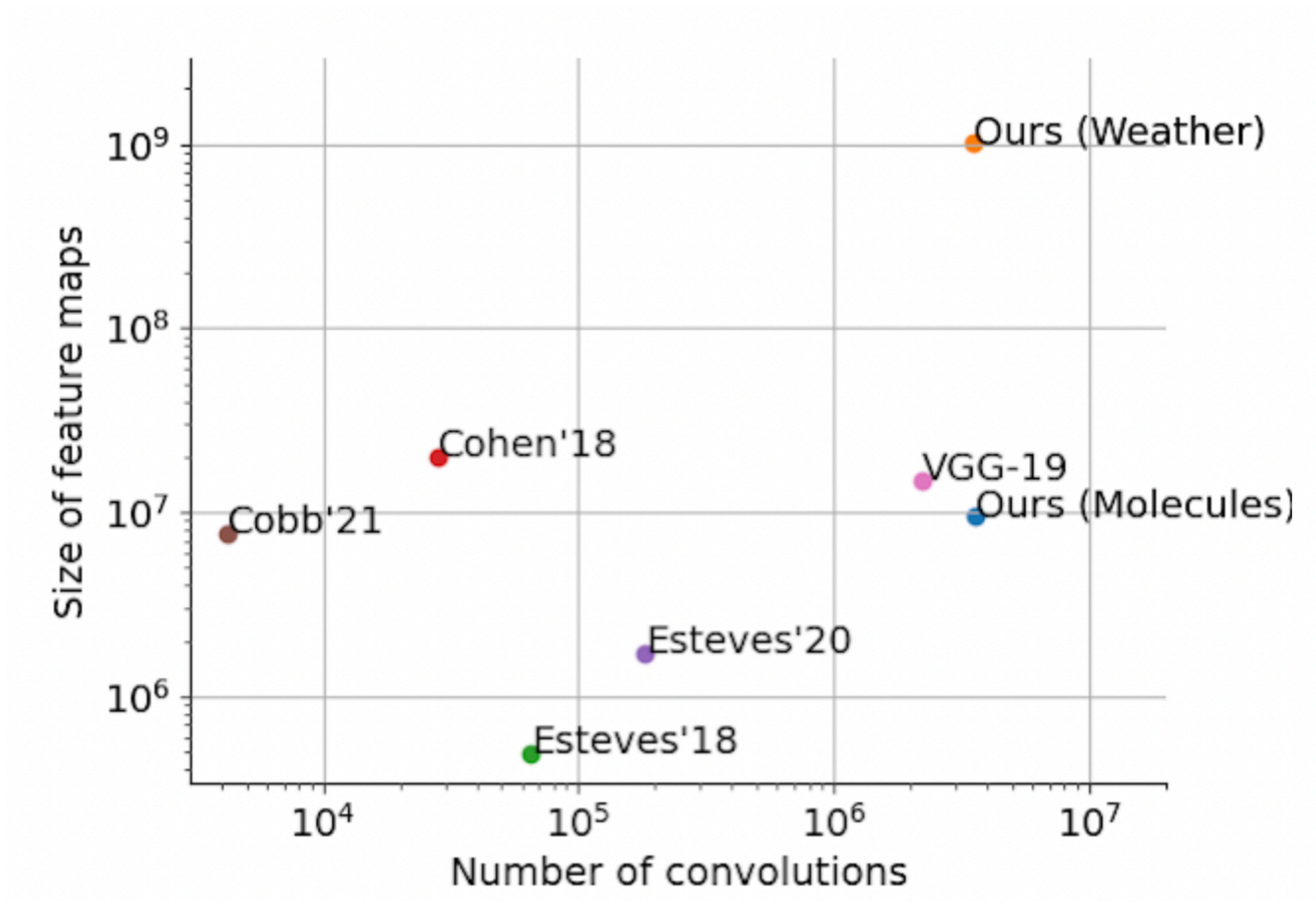
	Batch size	Latency (ms)	Throughput (N/s)
S2CNN	1	111 ± 0.6	9.0 ± 0.04
CNN	1	5.93 ± 0.24	169 ± 5.8

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

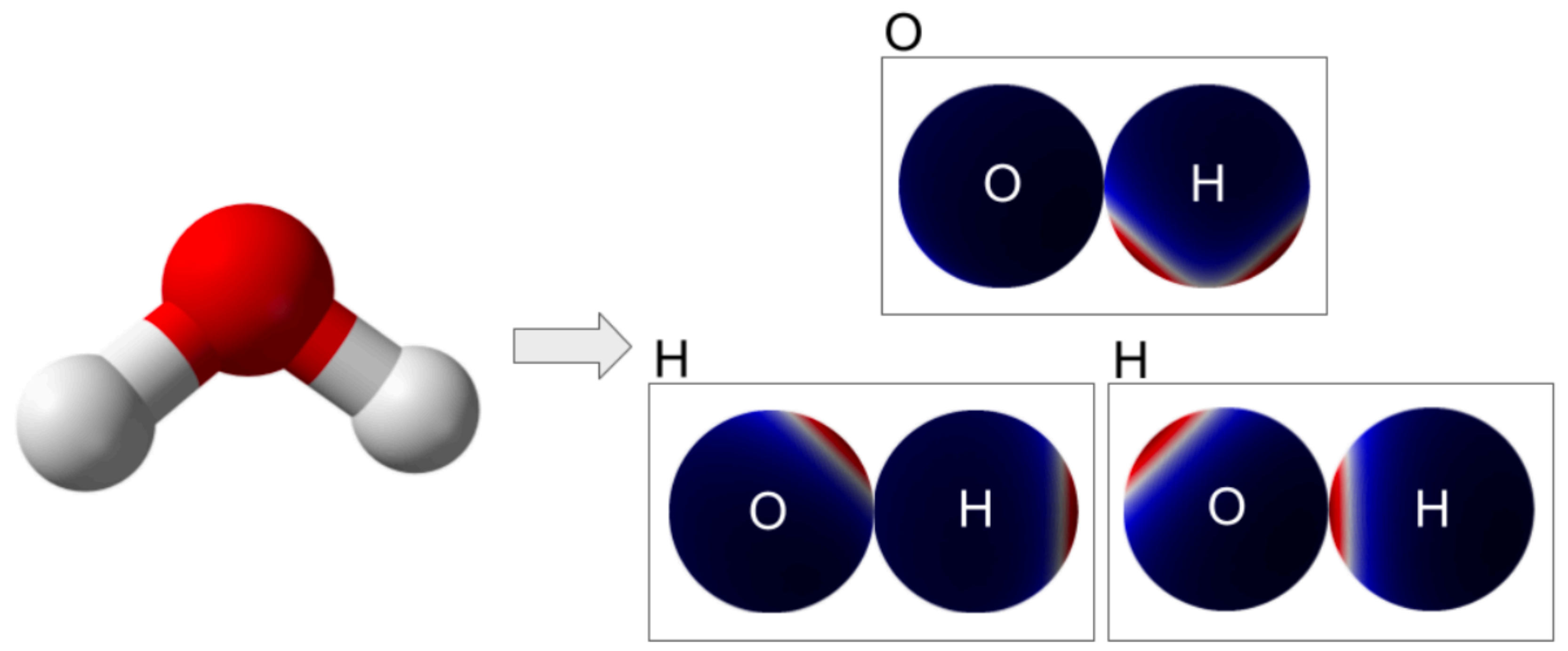


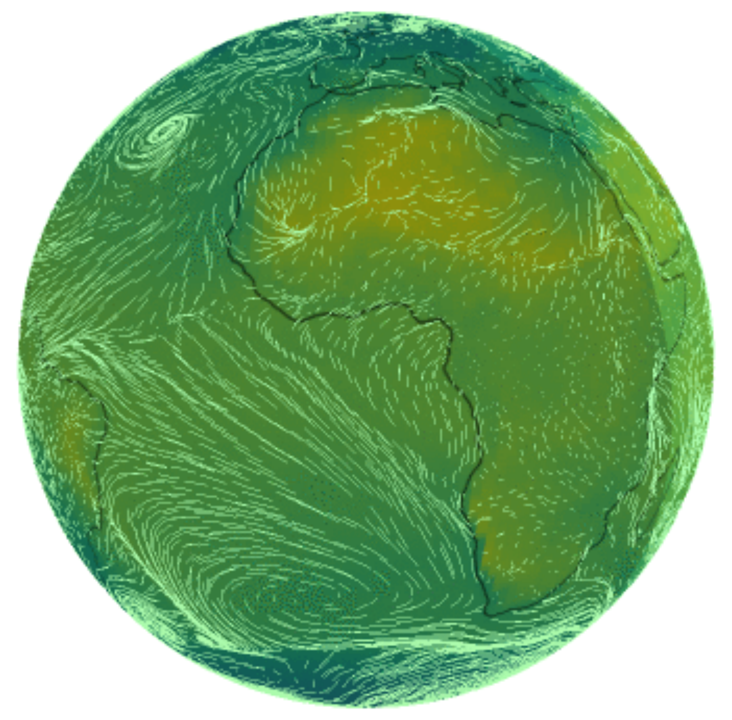


Esteves, Carlos, Ameesh Makadia, and Kostas Daniilidis. "Spin-weighted spherical cnns." *NeurIPS* (2020)
Esteves, Carlos, Jean-Jacques Slotine, and Ameesh Makadia. "Scaling spherical cnns." *arXiv:2306.05420* (2023)

Group convolutions on homogeneous spaces

Equivariant neural networks on the sphere





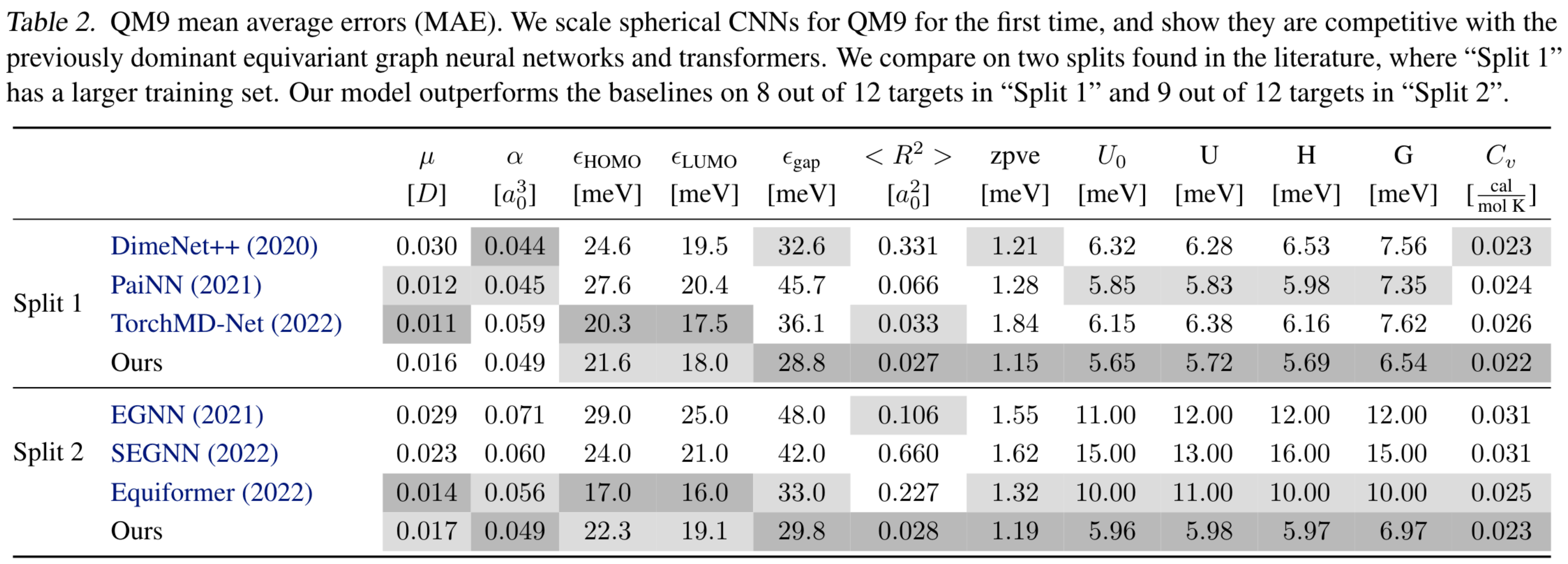
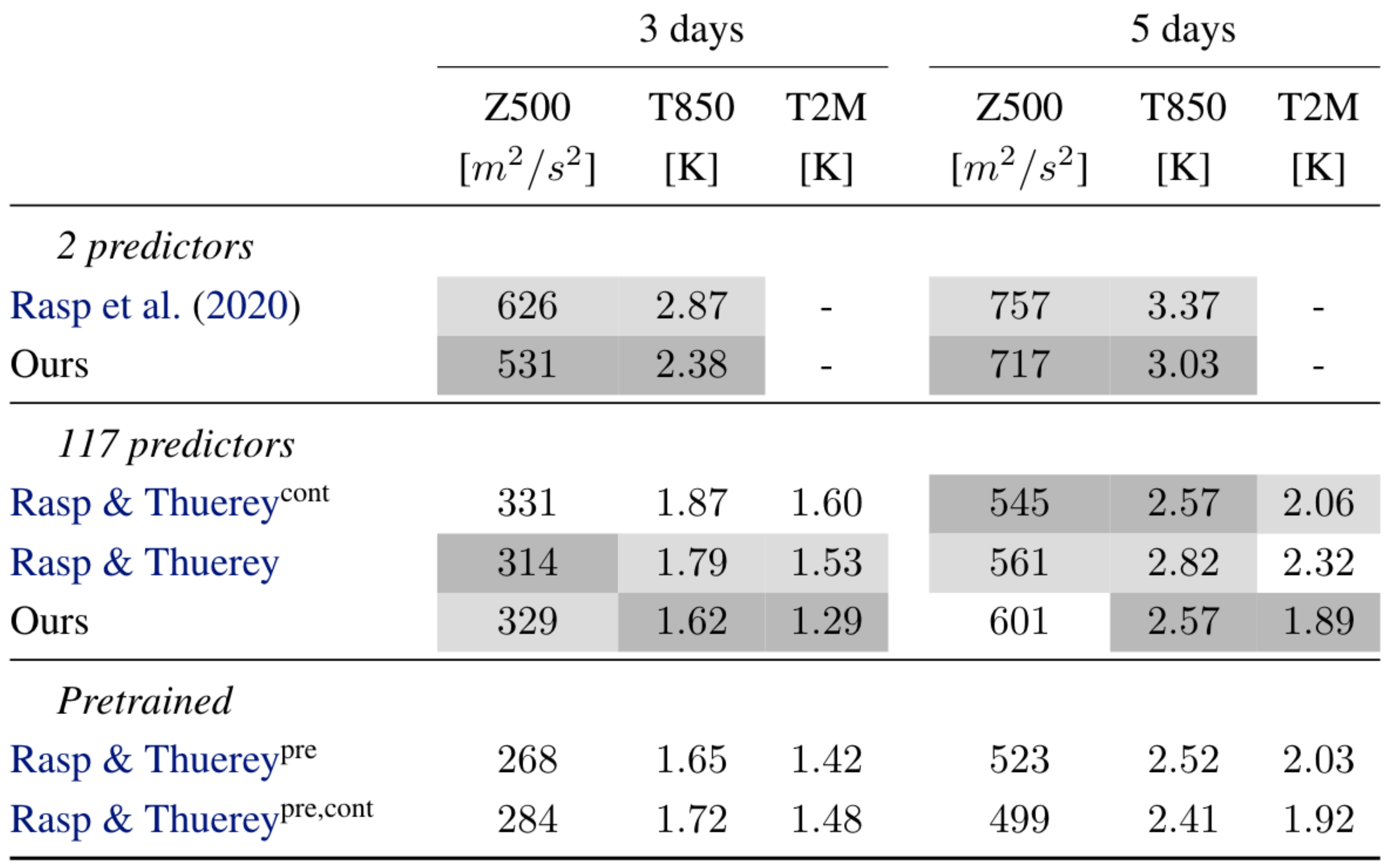


Table 2. QM9 mean average errors (MAE). We scale spherical CNNs for QM9 for the first time, and show they are competitive with the previously dominant equivariant graph neural networks and transformers. We compare on two splits found in the literature, where "Split 1" has a larger training set. Our model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

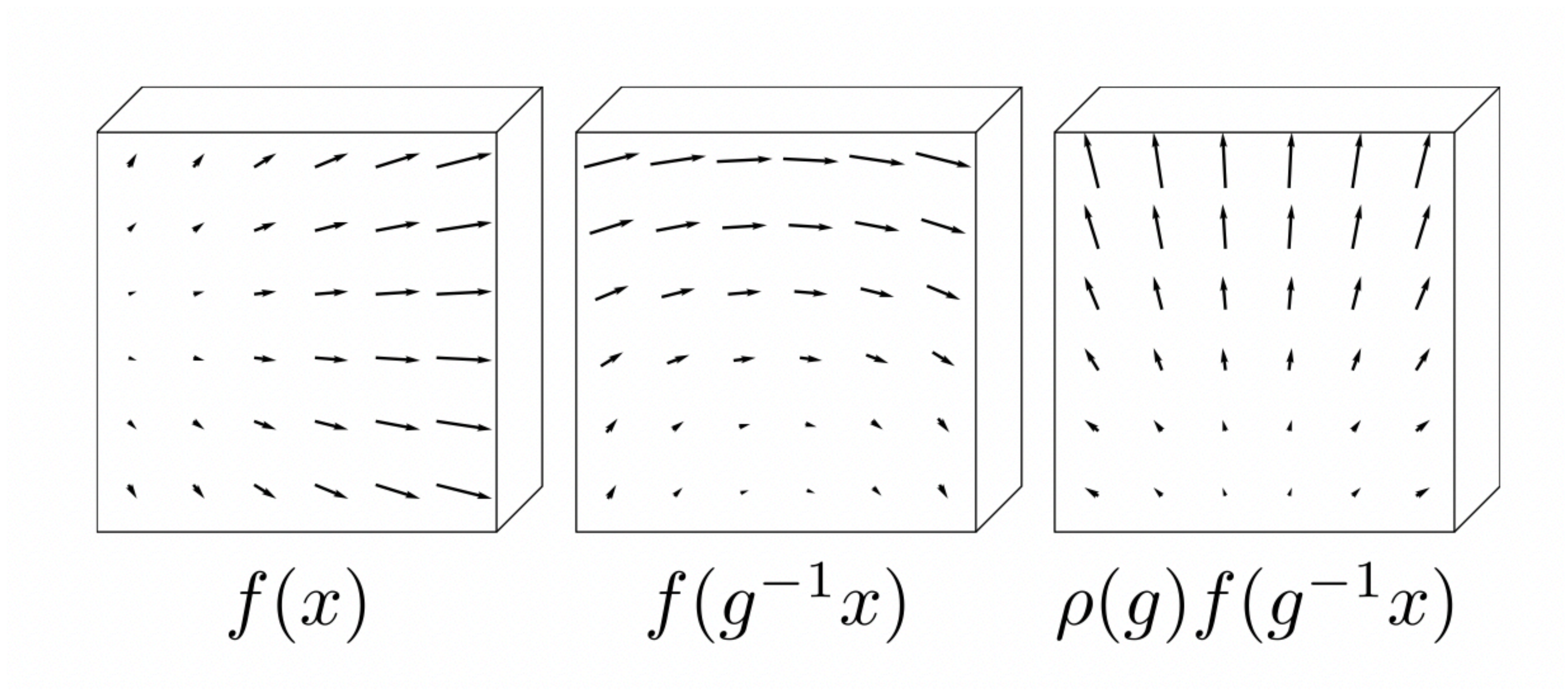
JAX implementation
https://github.com/google-research/spherical-cnn

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

Steerable convolution

Equivariant convolution on non-scalar features

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



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

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

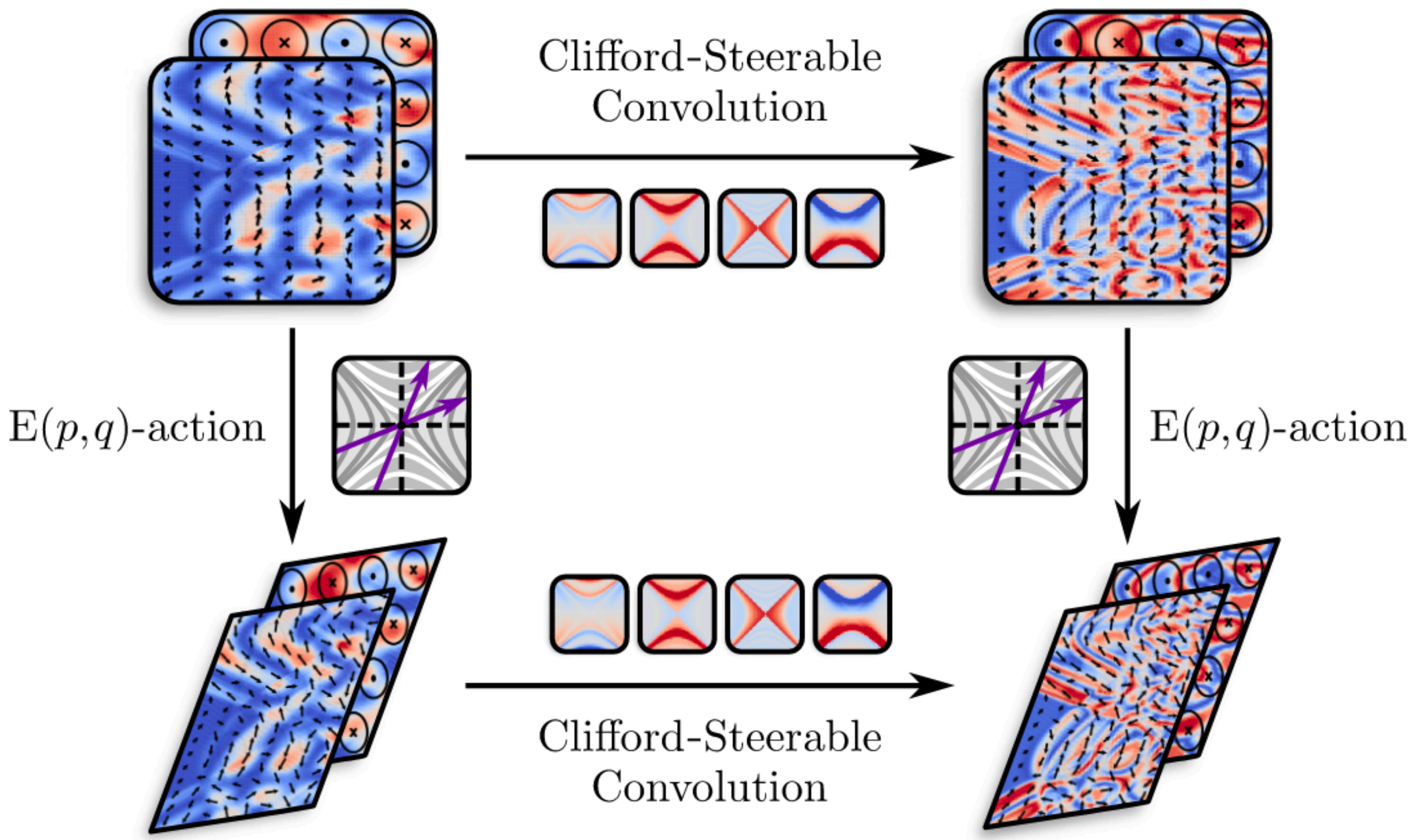
Convolution equivariant iff k satisfies constraint

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

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

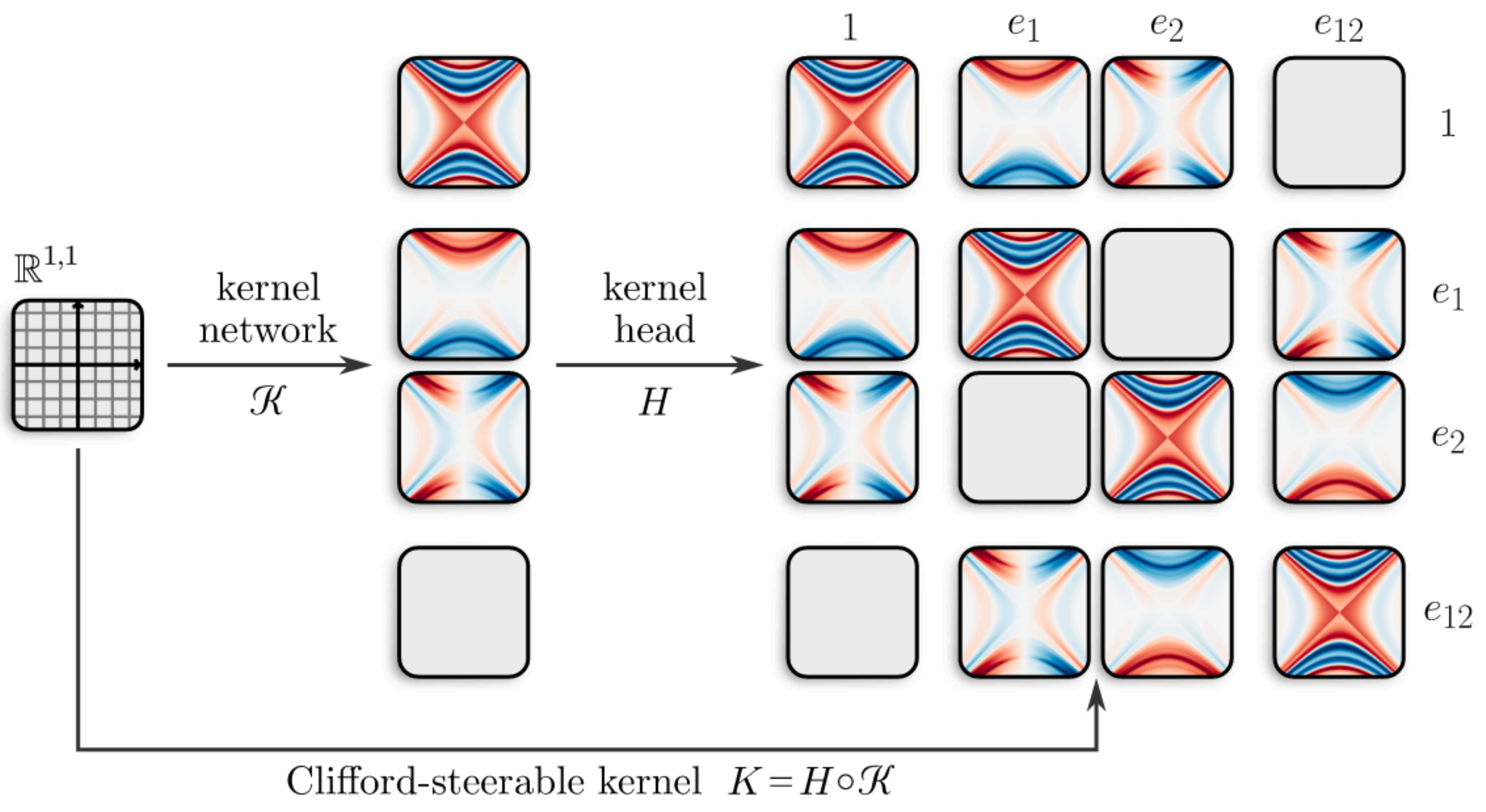
Clifford-Steerable convolution


Clifford-Steerable Convolution
E(p,q)-action
E(p,q)-action
Clifford-Steerable Convolution

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

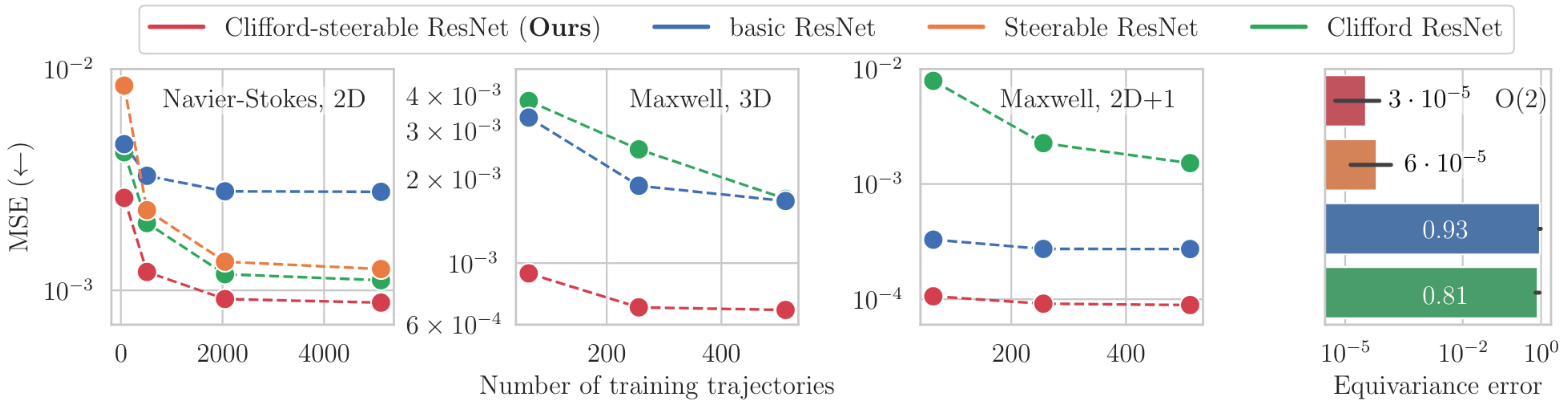
Clifford-Steerable convolution

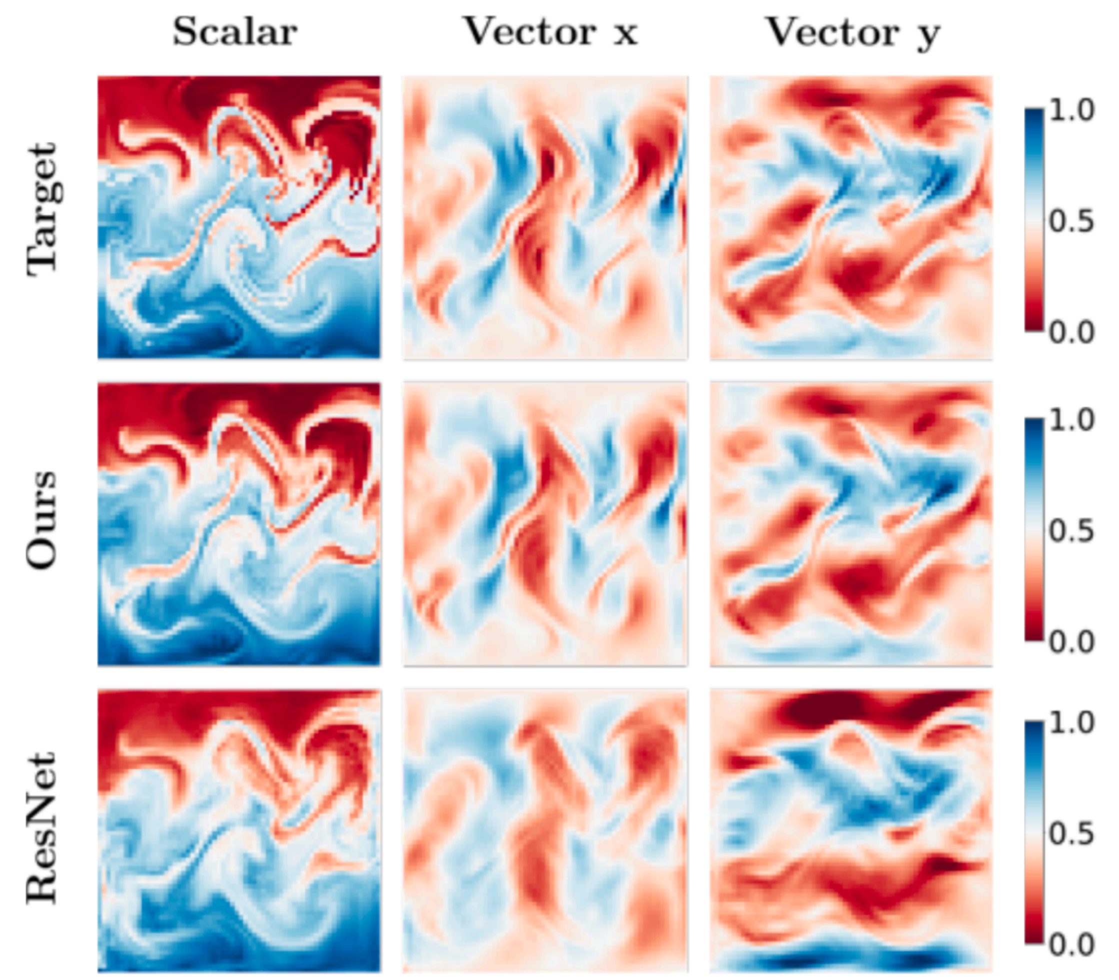
Solving the steerable kernel constraint



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*

hampus.linander@pm.me

gapindnns.github.io