

Recent developments in geometric machine learning: foundations, models, and more

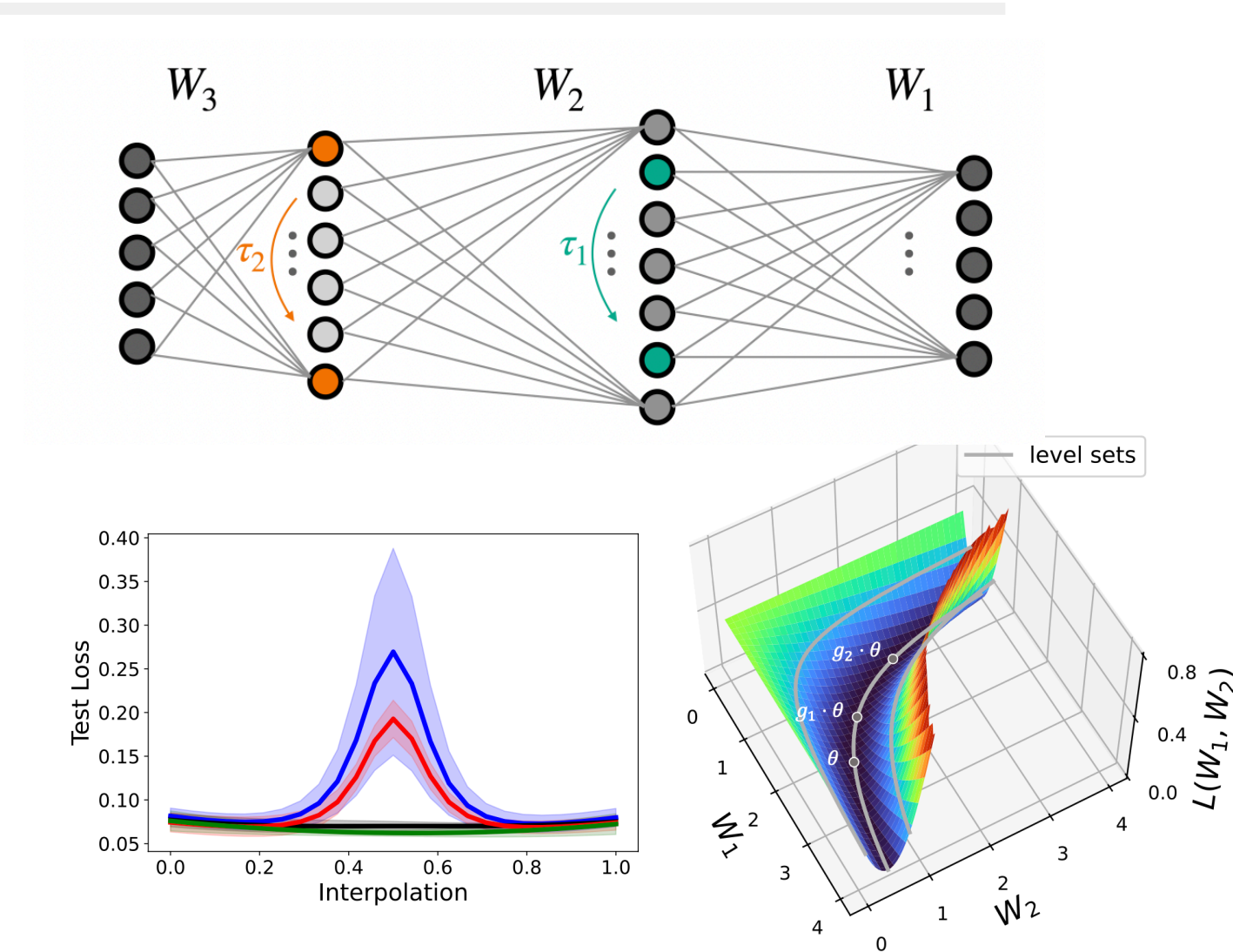
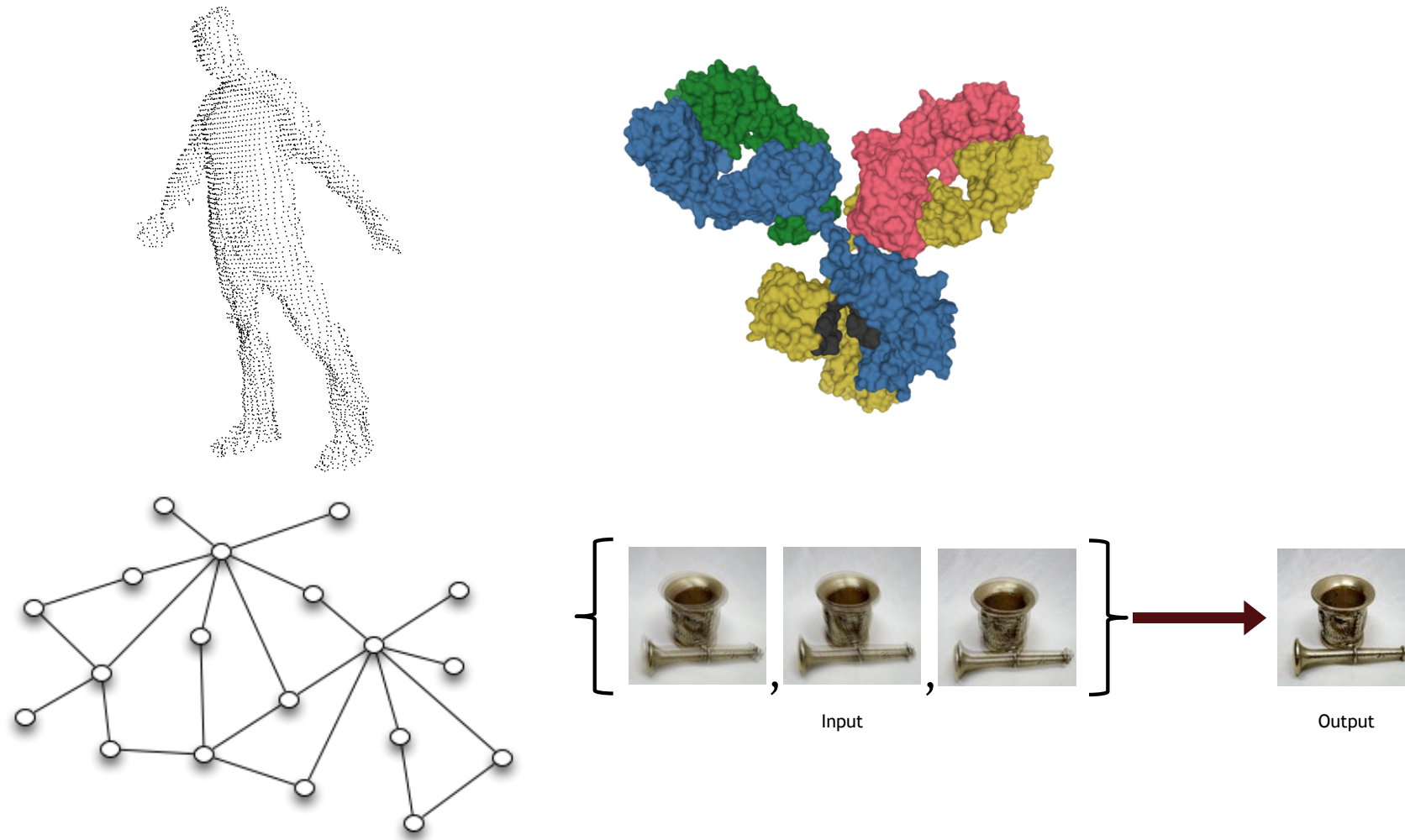
Stefanie Jegelka and Behrooz Tahmasebi

Technical University of Munich (TUM) and MIT
Harvard University

NeurIPS 2025, San Diego, CA
December 2025

What is this tutorial about?

Symmetries in data / tasks...



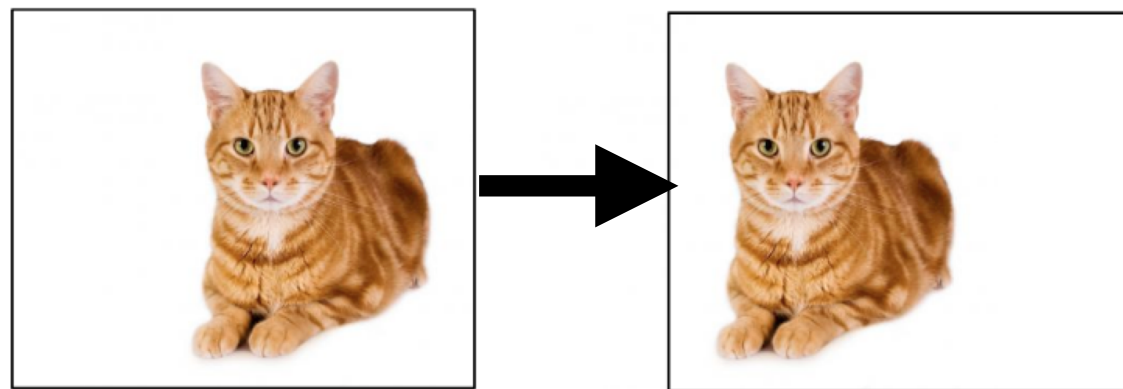
... Symmetries in models

Affects learning, sample complexity, generalization, robustness, generation, optimization, sampling, interpretability, model analysis, model merging, IP, ...

What is a “symmetry”?

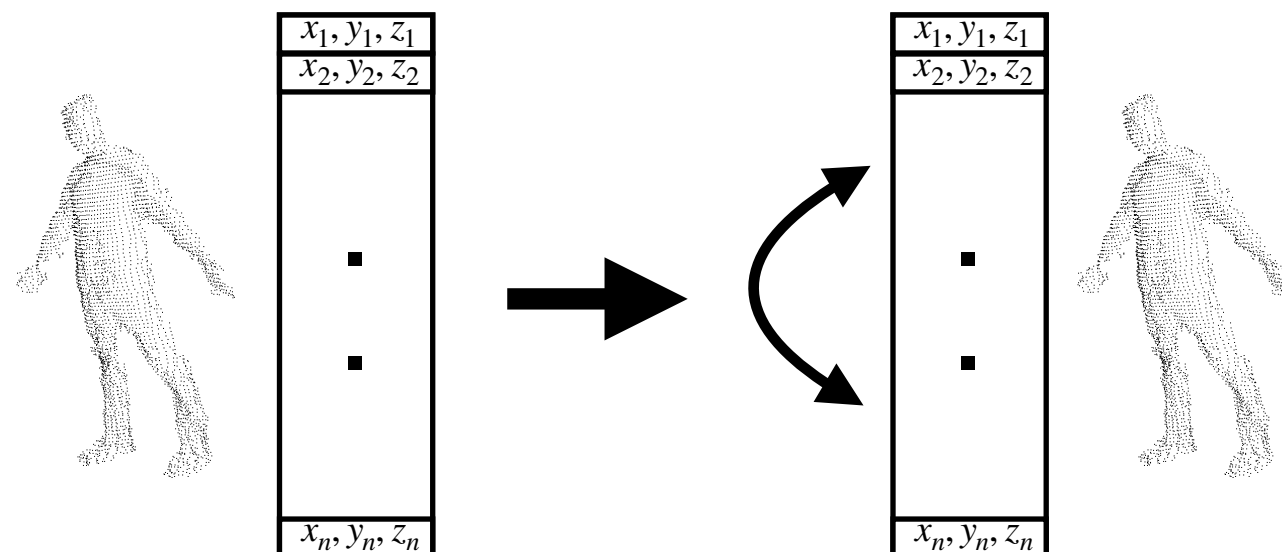
Symmetry of an object: **transformations** that leave properties of the object **unchanged**

Invariance: $f(T(x)) = f(x)$



$$f(\text{shift}(x)) = f(x)$$

image: H. Maron



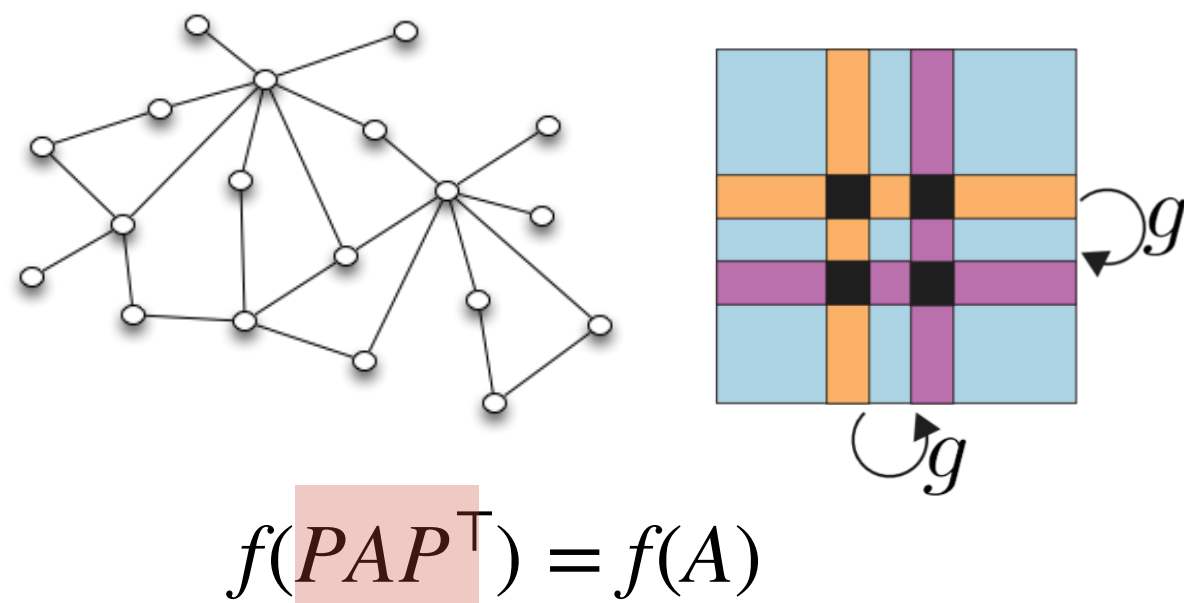
$$f(\text{permute}(x)) = f(x)$$

image: H. Maron

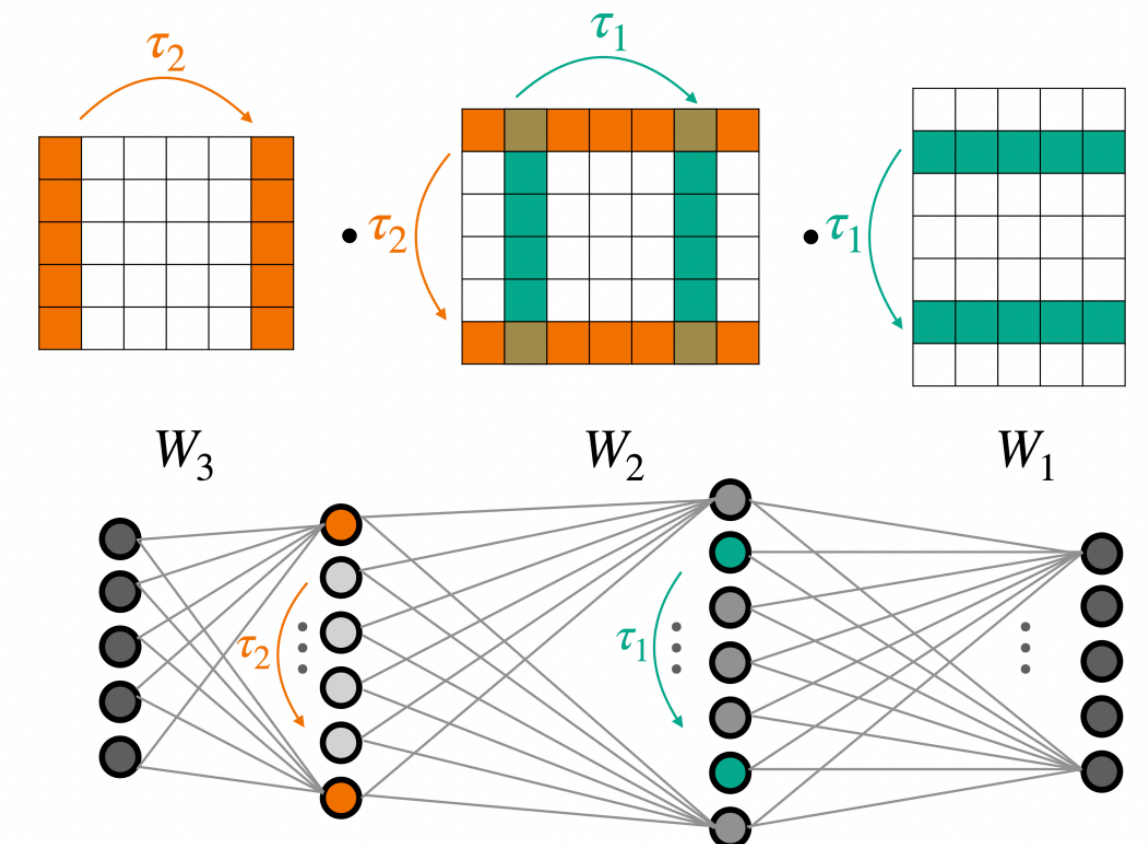
Invariance

Symmetry of an object: **transformations** that leave properties of the object **unchanged**

Invariance: $f(T(x)) = f(x)$

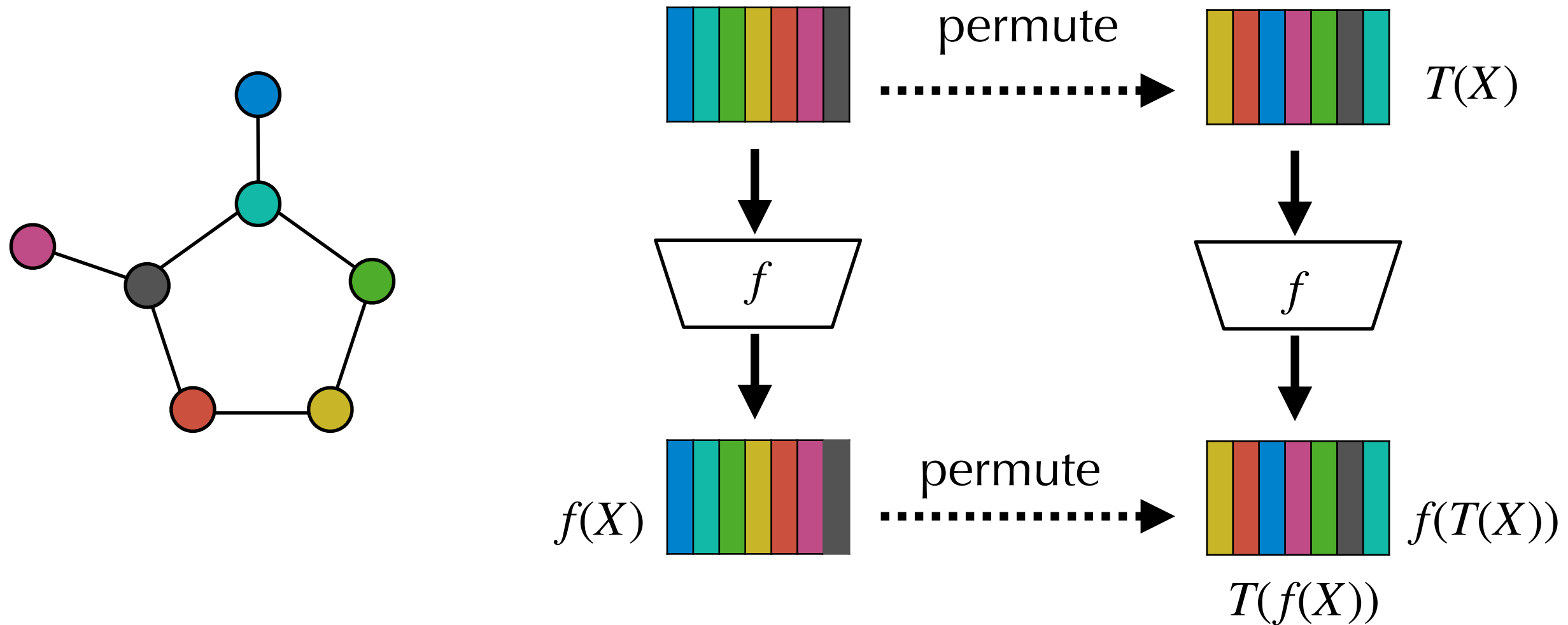


$$f(\cdot, PW_1, W_2P^T) = f(\cdot, W_1, W_2)$$



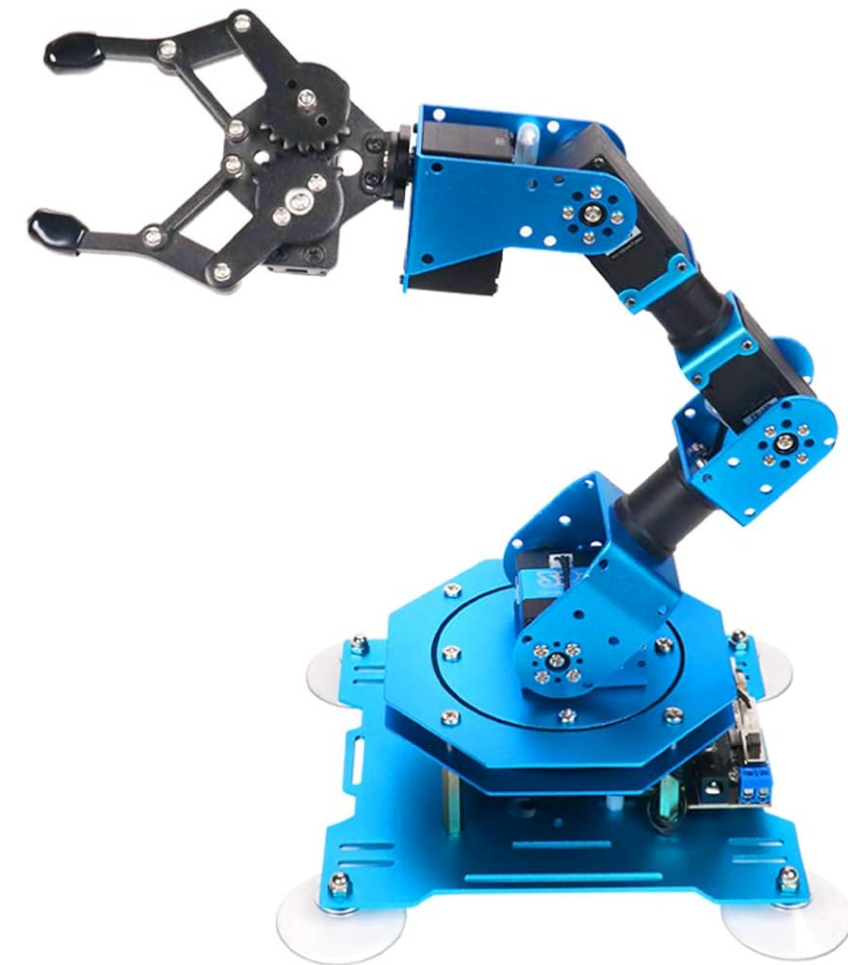
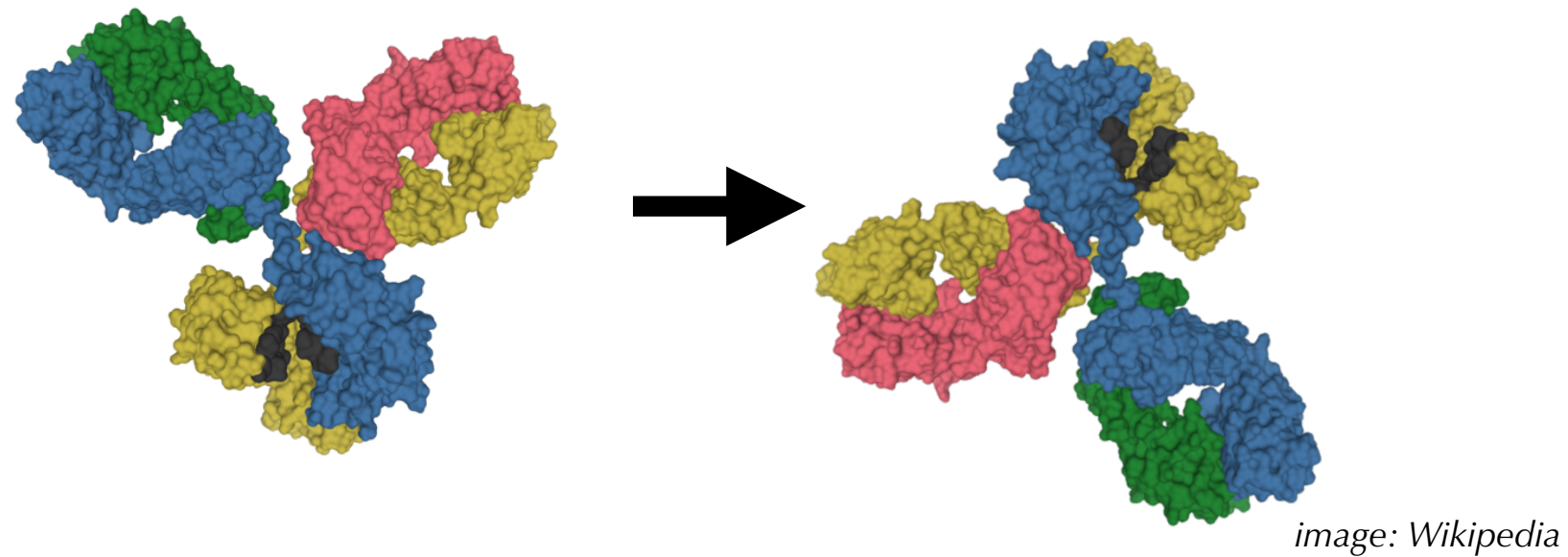
Equivariance

Equivariance: $f(T(x)) = T(f(x))$



Equivariance

Equivariance: $f(T(x)) = T(f(x))$



<https://www.amazon.com/Programming-Hiwonder-xArm1S-Educational-Building-Wireless/dp/B0793PFGCY?th=1>

Invariance is a special case of equivariance with scalar outputs!

Equivariant/invariant ML models: why?

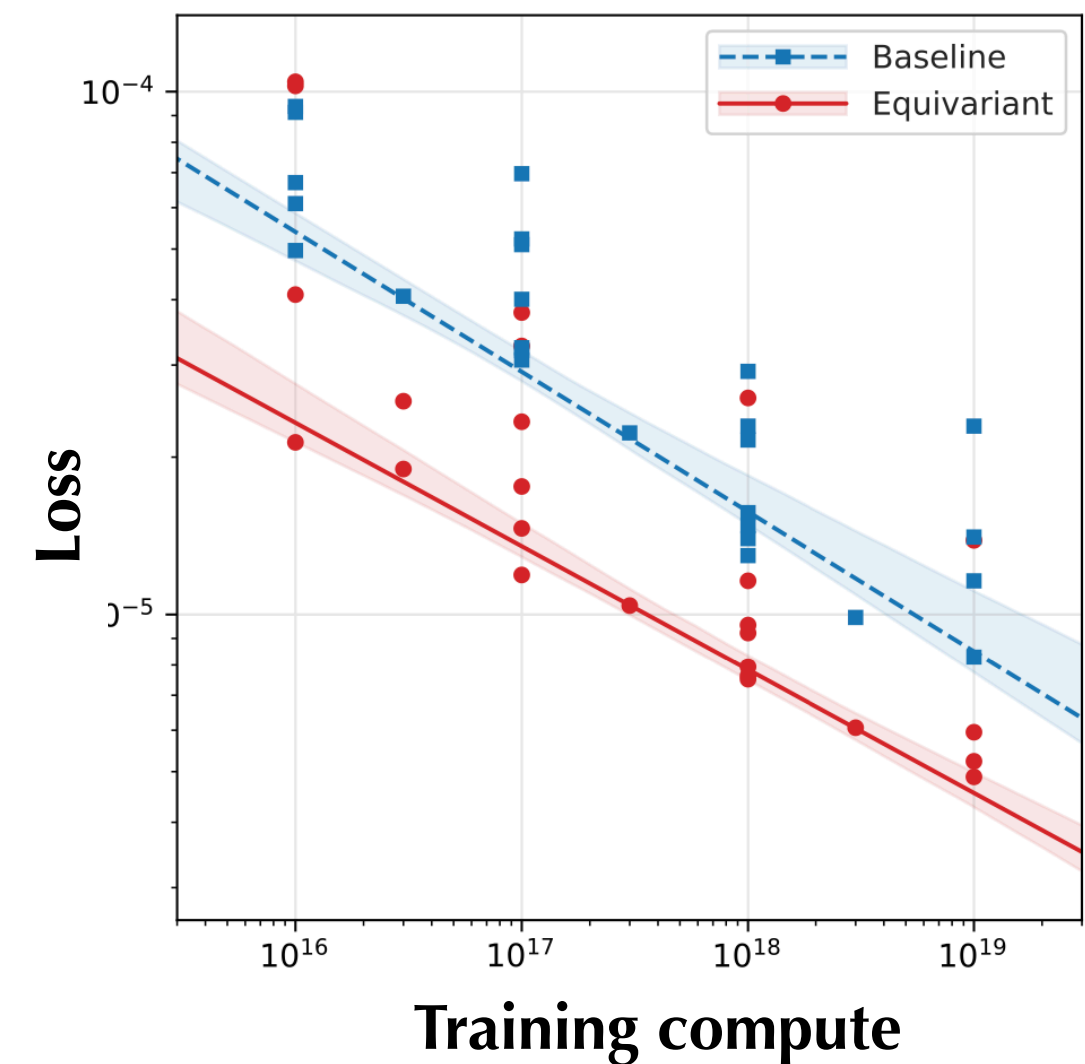
- Fewer degrees of freedom: better **data efficiency**

*“NequIP **outperforms existing models with up to three orders of magnitude fewer training data** [...]”*

The high data efficiency of the method allows for the construction of accurate potentials using high-order quantum chemical level of theory as reference and enables high-fidelity molecular dynamics simulations over long time scales.

(Batzner et al, Nature Communications 2022)

- **Robustness**, OOD generalization



(Brehmer et al 2025)

This has been an active area...

Geometric Deep Learning Grids, Groups, Graphs, Geodesics, and Gauges

Michael M. Bronstein¹, Joan Bruna², Taco Cohen³, Petar Veličković⁴

Tutorial: (Track2) Equivariant Networks

Risi Kondor, Taco Cohen

Tutorial and Q&A: Mon, Dec 7th, 2020 @ 11:30 – 14:00 CET

Extra Q&A session: Tue, Dec 8th, 2020 @ 11:00 – 11:50 CET

Many developments since!

- New modeling strategies
- New areas (e.g. neural parameter symmetries and weight space learning)
- New theory
- New observations, insights, discussions, ...

Roadmap

Part 1

- Introduction and basics
- Techniques for equivariance, with examples

Invariance: $f(g \cdot x) = f(x)$
Equivariance: $f(g \cdot x) = g \cdot f(x)$

Part 2

- Neural parameter symmetries and other recent directions
- Theory results and directions
- A bit of discussion

Disclaimer:
This is a tutorial
and hence
cannot cover all works

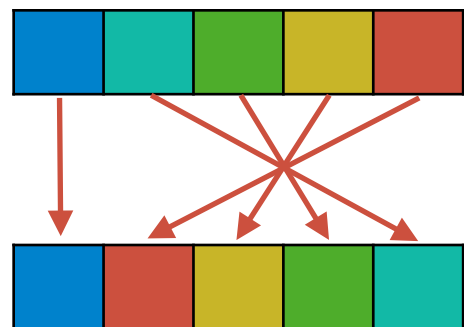
Groups

Usually, invariance/equivariance for a set of transformations that forms a **group**.

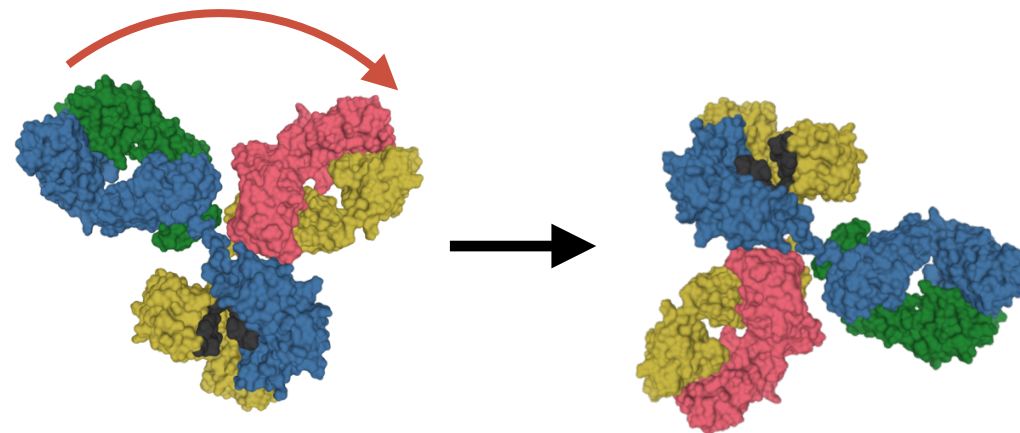
A group is a set G along with a binary operation (composition) satisfying:

- Associativity: $(gh)\ell = g(h\ell)$ for all $g, h, \ell \in G$.
- Identity: there is a unique $e \in G$ with $eg = ge = g$ for all $g \in G$.
- Inverse: for each $g \in G$ there is a unique inverse $g^{-1} \in G : gg^{-1} = g^{-1}g = e$
- Closure: for any $g, h \in G$ we have $gh \in G$

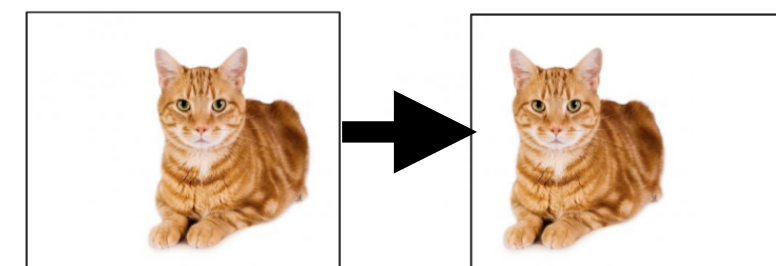
e.g. permutations:



e.g. rotations



e.g. translation



Group actions and group representations (informal)

- **Group action:** G acts on X as $g \cdot x$ for $g \in G, x \in X$

e.g.: permutations $G = S_n$ acts on $1, \dots, n$ as $\sigma \cdot i = \sigma(i)$

invertible matrices $GL(n)$ act on \mathbb{R}^n as $A \cdot x = Ax$

- **Group representation:** represent group action as a linear matrix operation (e.g. permutation, rotation)

e.g. permutations of vectors: $\sigma \cdot x = P_\sigma x$

rotations of vectors: $g \cdot x = Rx = \rho(g)x$

need: $\rho(gh) = \rho(g)\rho(h)$

Group actions and group representations (informal)

- **Group action:** G acts on X as $g \cdot x$ for $g \in G, x \in X$

e.g.: permutations $G = S_n$ acts on $1, \dots, n$ as $\sigma \cdot i = \sigma(i)$

invertible matrices $GL(n)$ act on \mathbb{R}^n as $A \cdot x = Ax$

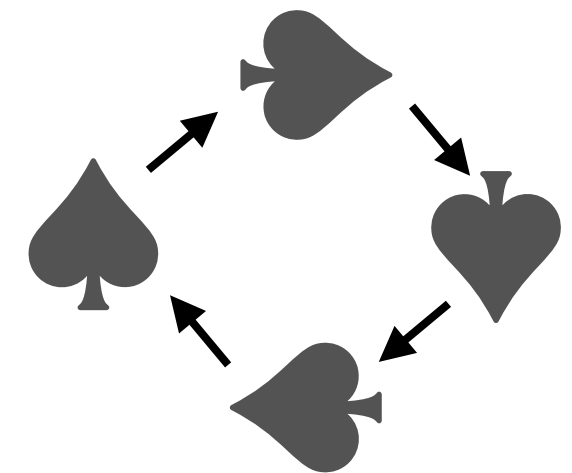
- **Group representation:** represent group action as a linear matrix operation (e.g. permutation, rotation)

e.g. permutations of vectors: $\sigma \cdot x = P_\sigma x$

rotations of vectors: $g \cdot x = Rx = \rho(g)x$

- **Orbit** of x : all elements reachable from x via group action

$$\text{Orb}_G(x) = \{g \cdot x \mid g \in G\}$$



How can we get equivariant models?

Using an off-the-shelf (non-equivariant) model:

- Data augmentation
- Canoni(cali)zation
- Group / frame averaging

Invariance: $f(g \cdot x) = f(x)$

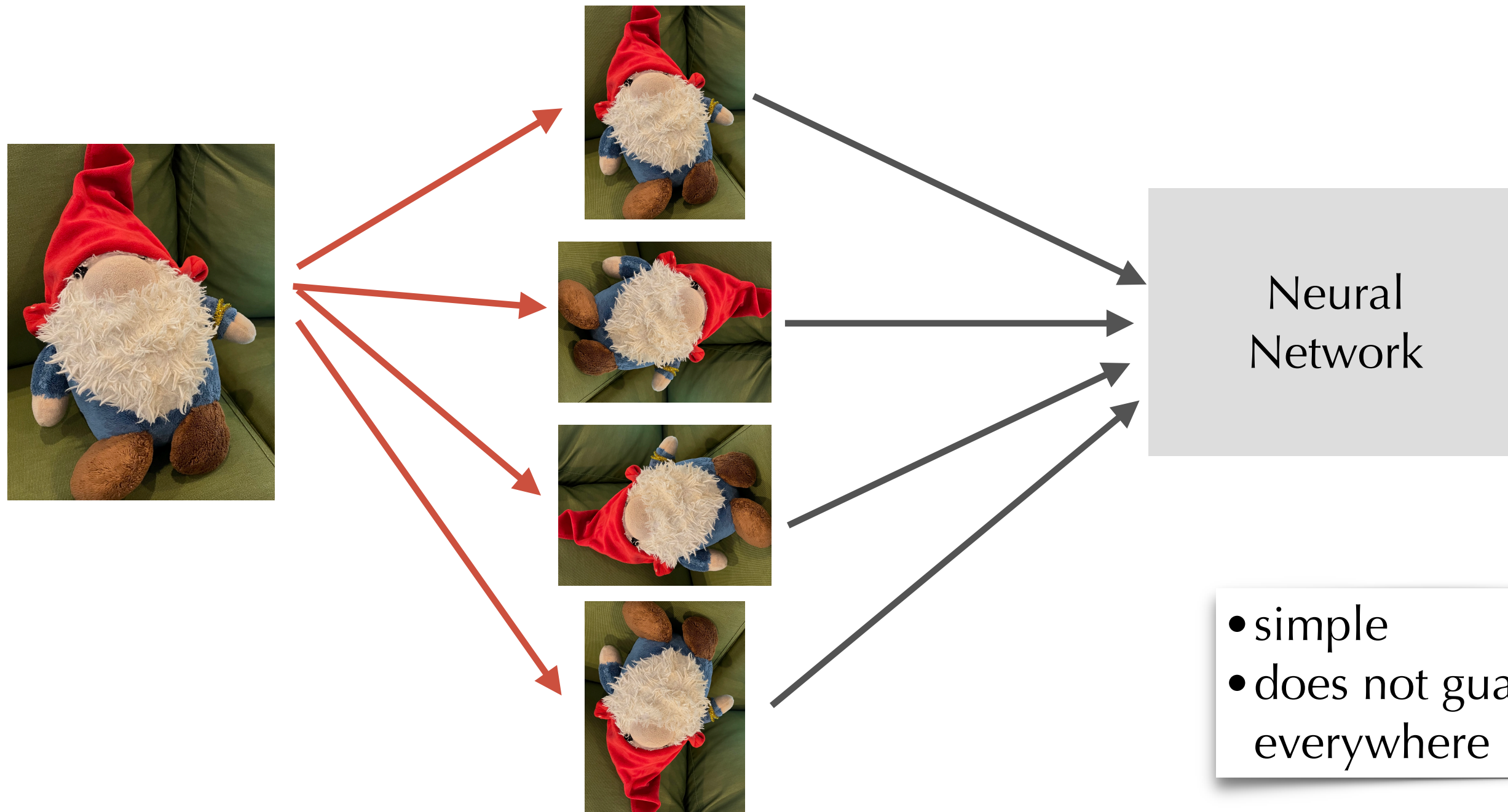
Equivariance: $f(g \cdot x) = g \cdot f(x)$

Constructing equivariant models:

- Linear equivariant layers + nonlinearity: parameter sharing, group convolution
- Parameterizing representations of invariant functions (invariant theory)
- Representation theory

Data augmentation

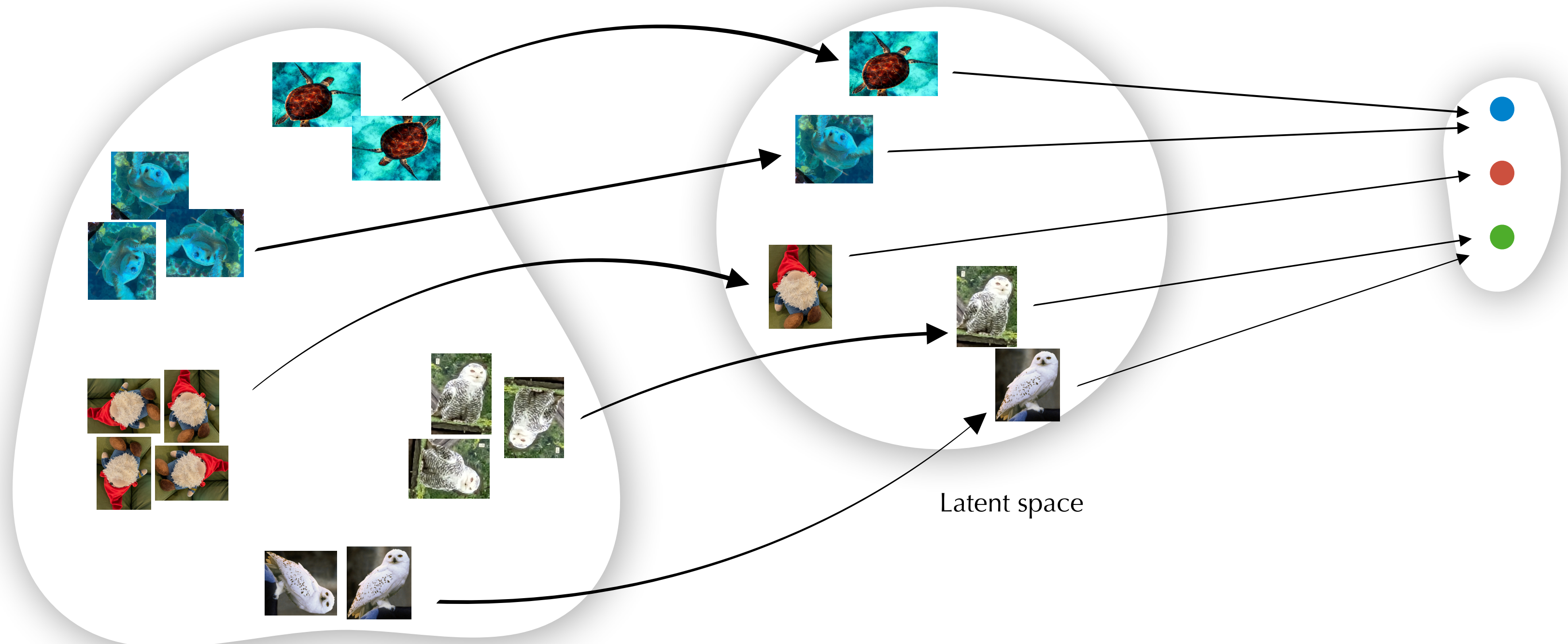
- For data point x , add transformed samples $g \cdot x$ to the training data, for $g \in G$



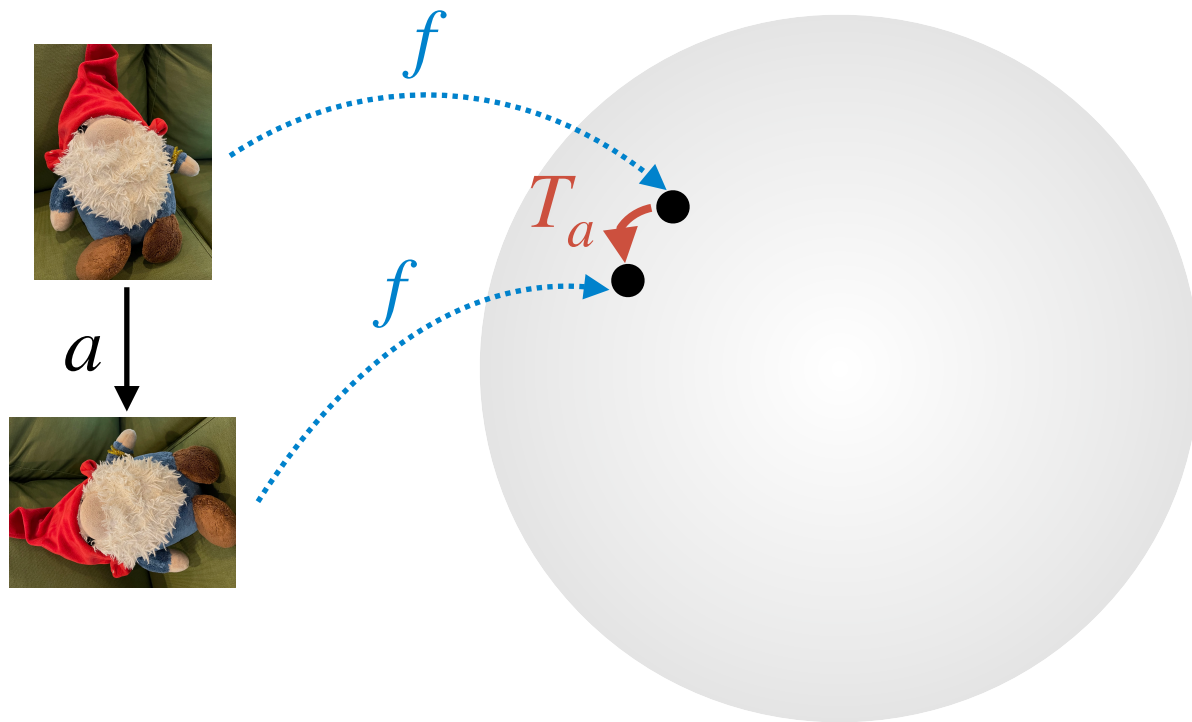
- simple
- does not guarantee invariance everywhere

Learning invariant representations

- e.g., via contrastive learning



Equivariant representation learning

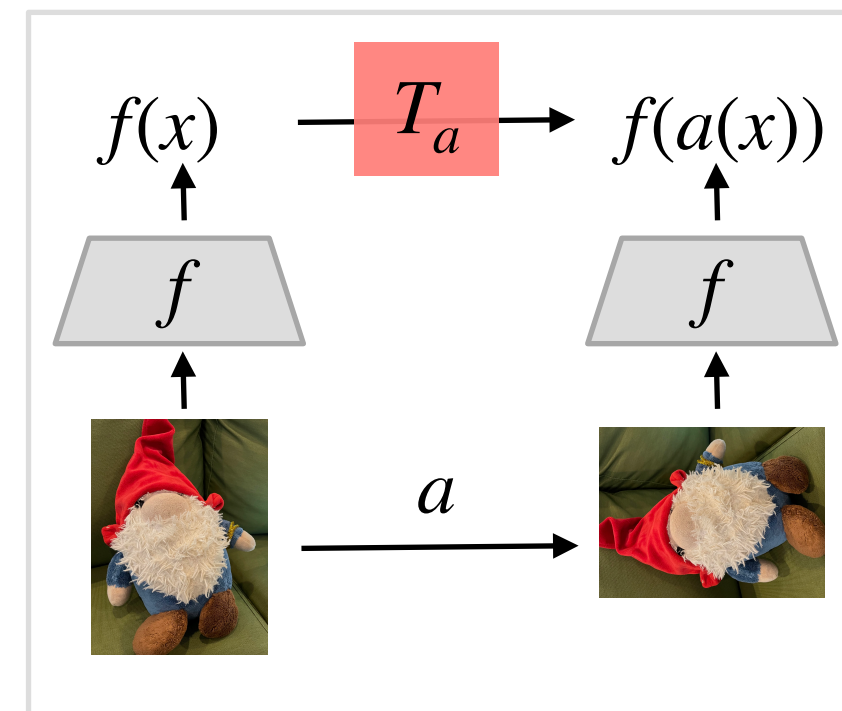


- May not be consistent with compositions:

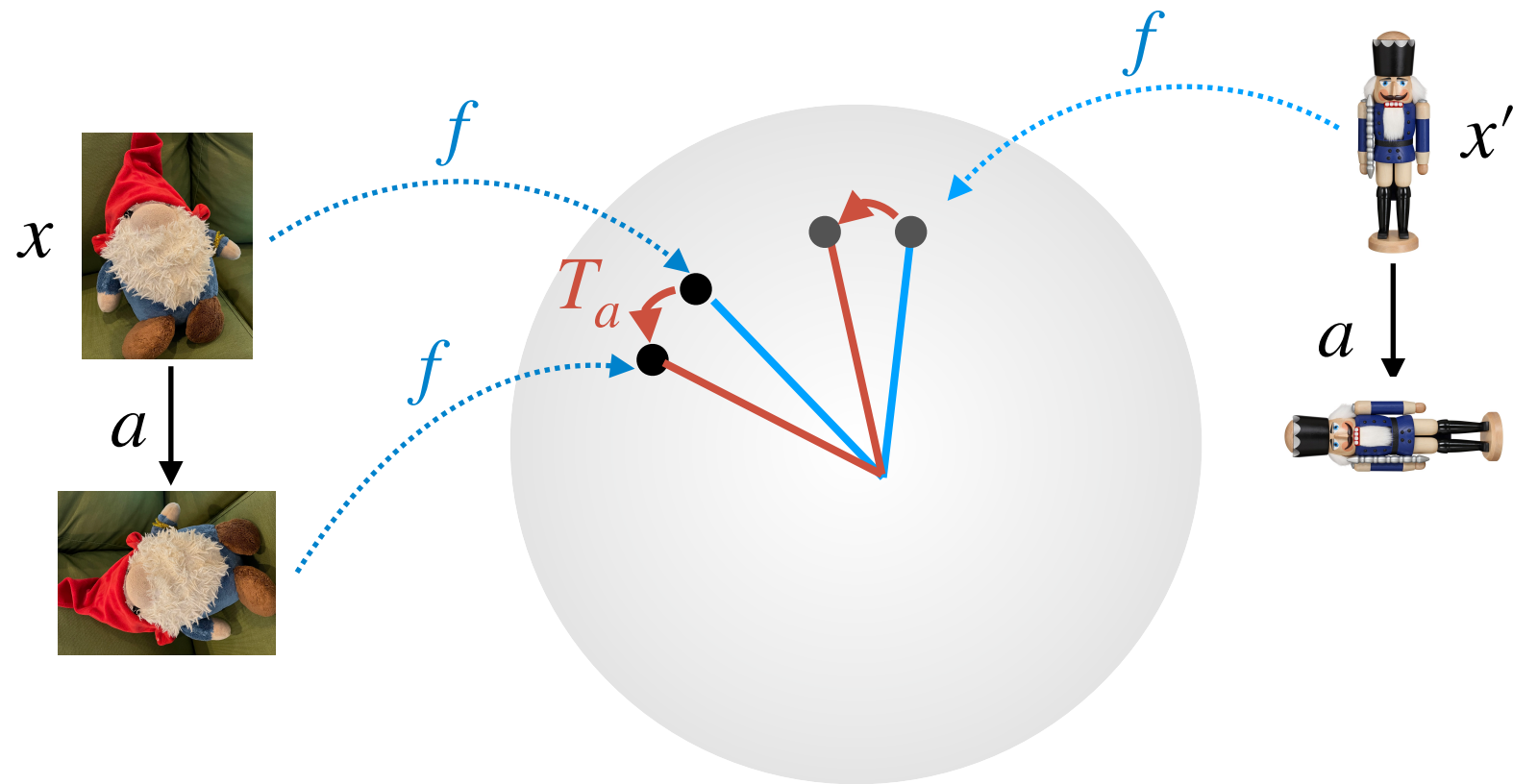
$$T_{a_2 \circ a_1} \neq T_{a_2} T_{a_1}$$

- Want: $f(a(x)) = T_a f(x)$
- $\text{MLP}(a, f(x))$

- Loss: $\|f(a(x)) - \text{MLP}(a, f(x))\|$



Equivariant representation learning



When loss is zero, then each T_a is encoded as a rotation R_a and preserves composition.

$T_a = R_a = \rho(a)$ is a group representation

Input transformation need not be a rotation!
(Peter & Weyl 1927)

- Want: $f(a(x)) = T_a f(x)$
- Can make T_a linear: rotation

- Loss:
rotation preserves inner products

$$\mathbb{E}_{a \sim \mathcal{A}} \mathbb{E}_{x, x'} \left[f(a(x'))^\top f(a(x)) - f(x)^\top f(x') \right]^2$$

How can we get equivariant models?

Using an off-the-shelf (non-equivariant) model:

- Data augmentation
- Canoni(cali)zation
- Group / frame averaging

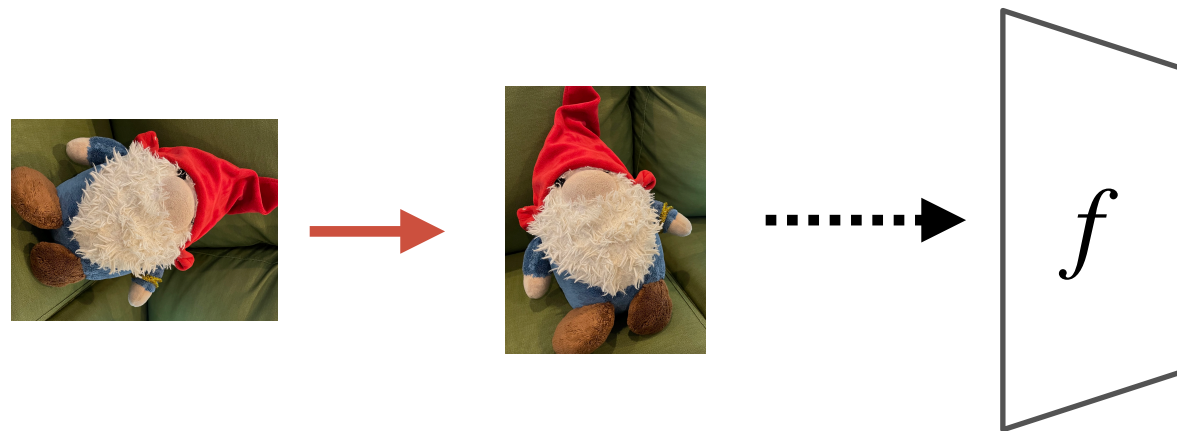
Constructing equivariant models:

- Linear equivariant layers + nonlinearity: parameter sharing, group convolution
- Parameterizing representations of invariant functions (invariant theory)
- Representation theory

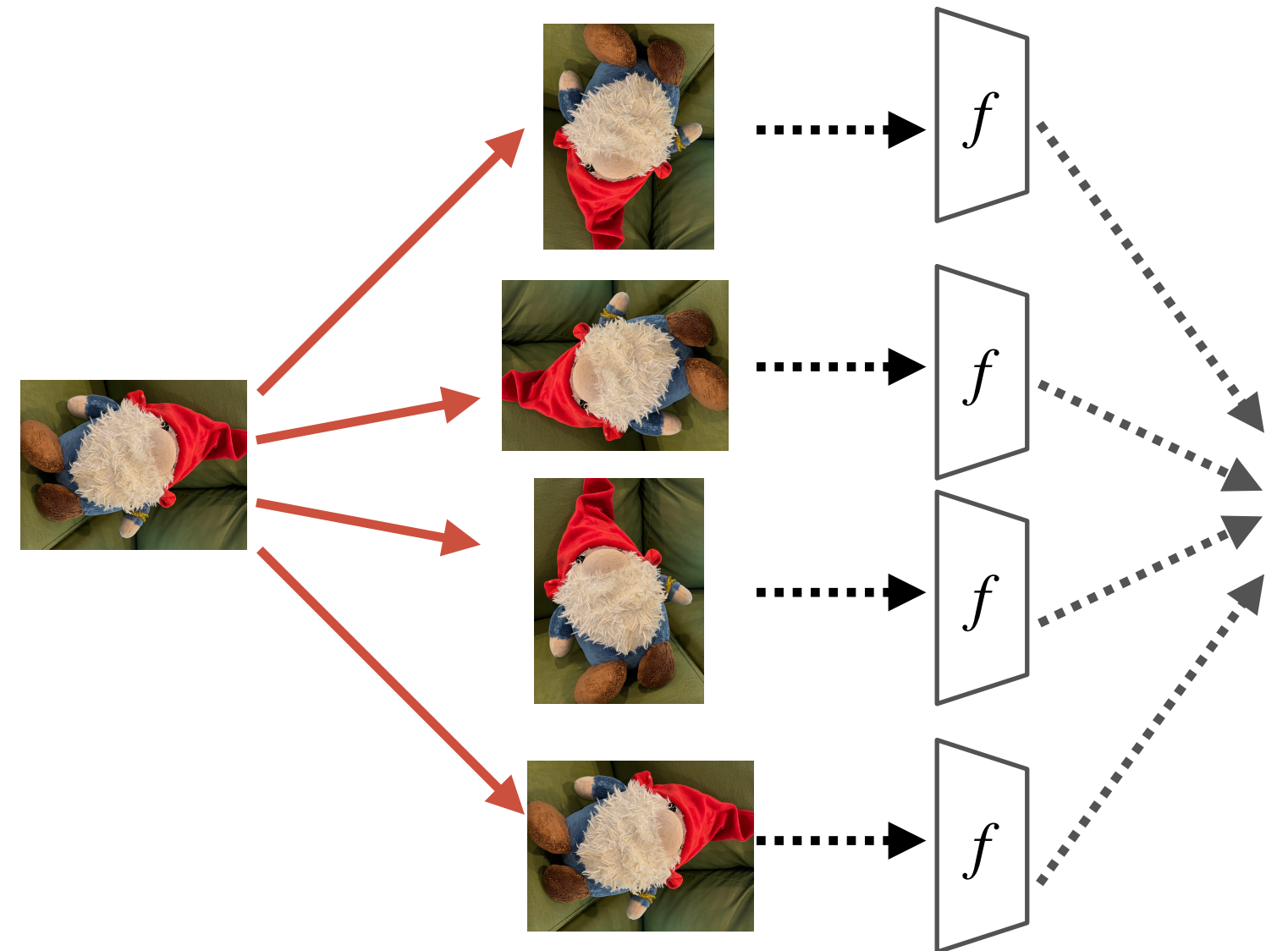
Two ideas for making f invariant

have: off-the-shelf neural network f

canonicalization

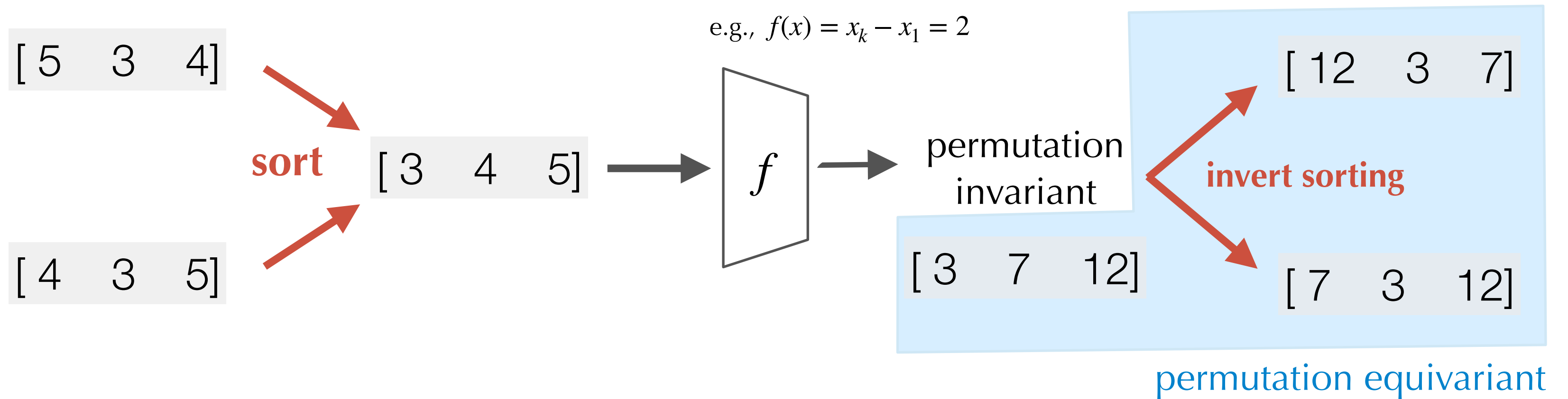


symmetrization / group averaging



Canonicalization

- Apply (arbitrary) f to a “**standardized**” representation (element in the orbit) of x

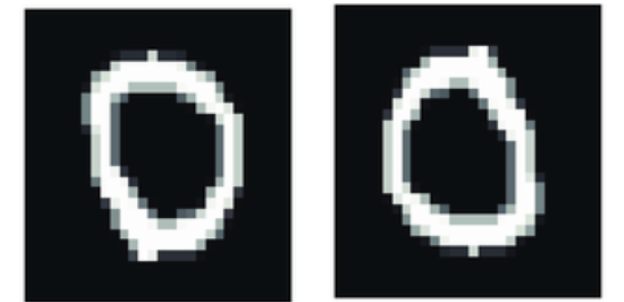


- Canonicalization can also be learned! (*Kaba et al 2022, Zhang et al 2019, Leo et al 2022*)

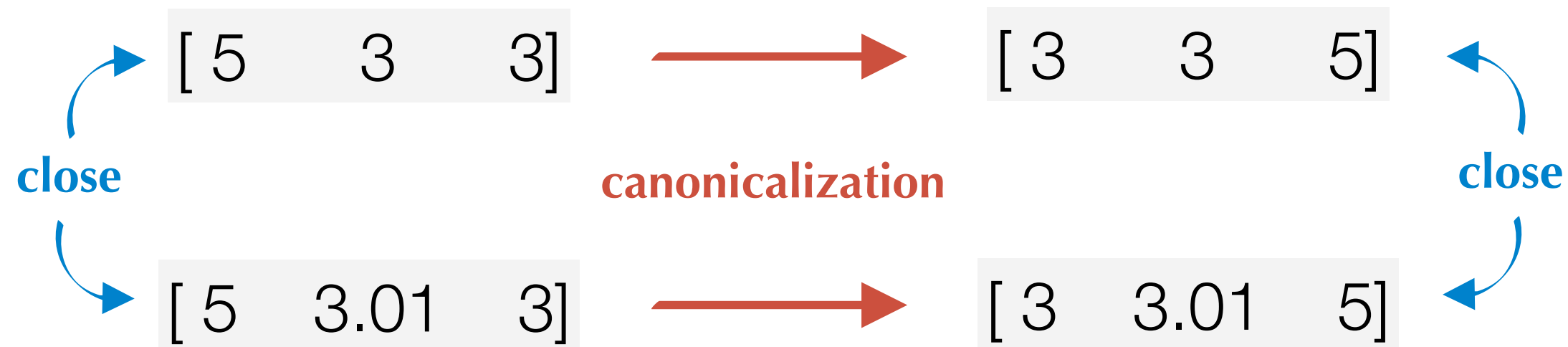
(*Yüceer & Oflazer 1993, Lowe 2004, Niepert et al 2016, Winter et al 2022, Vadgama et al 2022,...*)

Canonicalization: challenges

- Canonicalization not always easy
(e.g. graphs, rotation+permutation for point clouds)
- Symmetries in the input
- Continuity (*Zhang et al 2020, Dym et al 2024*)

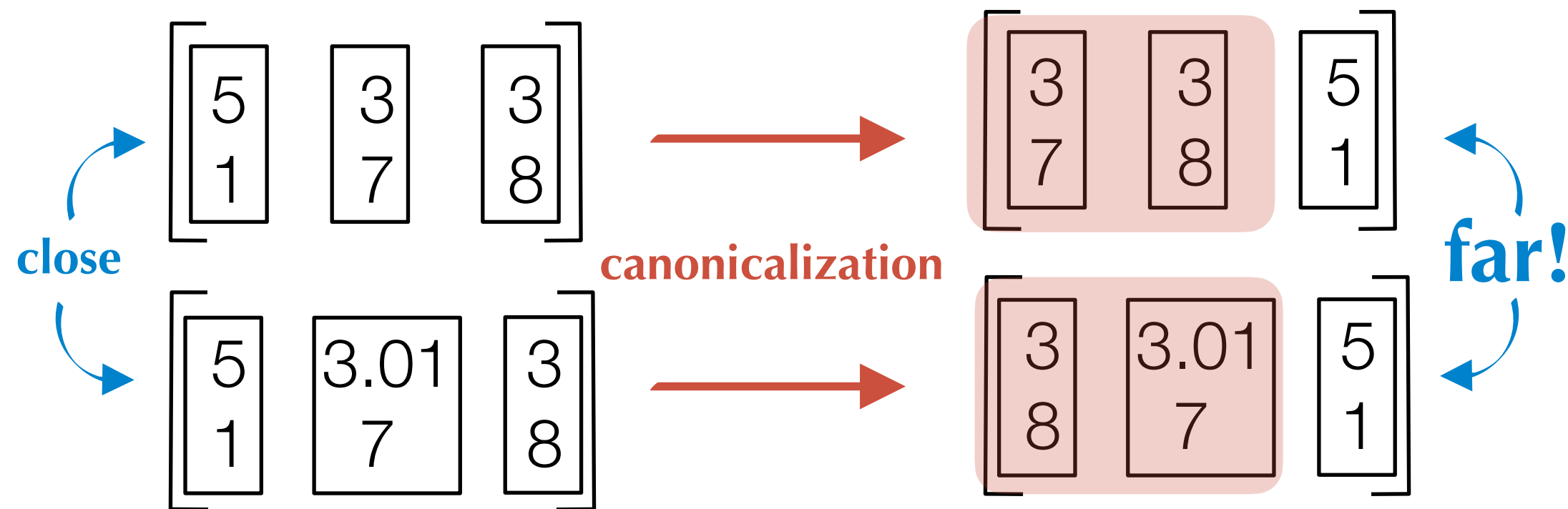


[5 3 3]



Canonicalization: challenges

- Continuity (*Zhang et al 2020, Dym et al 2024*)

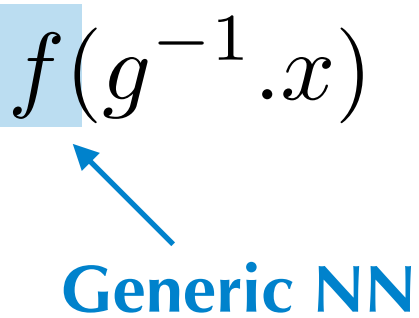


For permutations of $n > 1$ vectors in $d > 1$ dimensions, and for $SO(2)$, there is **no continuous canonicalization**.
(*Dym et al 2024, Zhang et al 2020*)

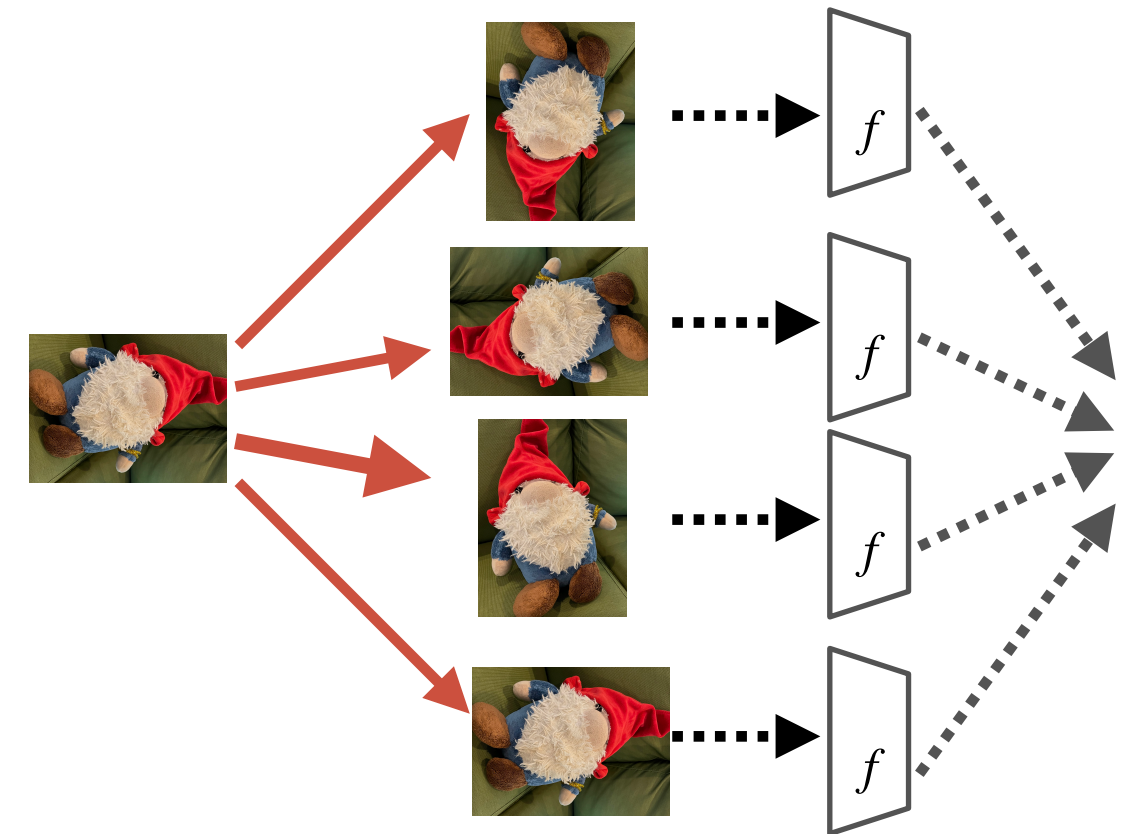
Group averaging (Reynolds operator)

- Universal approximation of invariant / equivariant functions (*Yarotsky, 2021*)

invariant
$$\bar{f}(x) = \frac{1}{|G|} \sum_{g \in G} f(g^{-1} \cdot x)$$


Generic NN

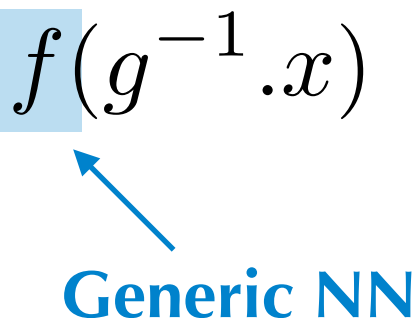
equivariant
$$\bar{f}(x) = \frac{1}{|G|} \sum_{g \in G} g \cdot f(g^{-1} \cdot x)$$



Simple and expressive,
but can be **challenging** if group is large / infinite
-> approximations

Group averaging

$$\bar{f}(x) = \frac{1}{|G|} \sum_{g \in G} f(g^{-1} \cdot x)$$

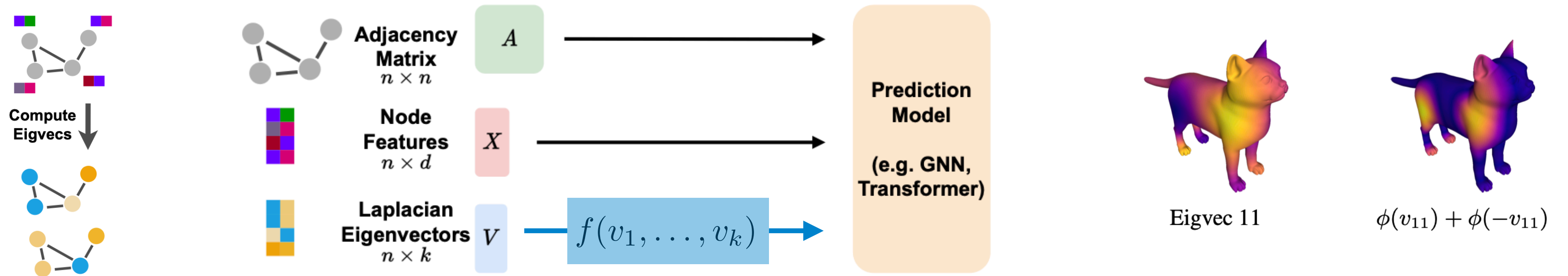

Generic NN

- **Example 1:** Janossy pooling for permutations (*Murphy et al 2019*), relational pooling for more expressive graph NNs (*Murphy et al 2019*)

$$\bar{f}(x_1, \dots, x_k) = \frac{1}{|G|} \sum_{\sigma \in S_k} f(x_{\sigma(1)}, \dots, x_{\sigma(k)})$$

Group averaging

- **Example 2:** Neural networks on eigenvectors (sign flips)
graph positional encodings, neural fields on manifolds



If v is an eigenvector, then so is $-v$!

Want: $f(v_i) = f(-v_i)$

SignNet (Lim, Robinson et al 2023):

$$f(v) = h\left(\left[\phi(v_i) + \phi(-v_i)\right]_{i=1}^k\right)$$

$\swarrow \quad \searrow$
learned

Anything in between?

canonization

- + one transformation:
cheap
- not continuous

frame averaging

average over a subset
 $\mathcal{F}(x) \subseteq G$
of group elements

$$\frac{1}{|\mathcal{F}(x)|} \sum_{g \in \mathcal{F}(x)} f(g^{-1} \cdot x) \quad \text{invariant}$$

$$\frac{1}{|\mathcal{F}(x)|} \sum_{g \in \mathcal{F}(x)} g \cdot f(g^{-1} \cdot x) \quad \text{equivariant}$$

(Puny et al 2022, Atzmon et al 2022)

group averaging

- + continuous
- full group:
can be expensive,
approximations

frame should be equivariant

Example: representations of point clouds / proteins

- Equivariance to Euclidean group $E(3)$ (rotation, reflection, translation)
(variation without reflection possible, $SE(3)$)
- PCA frame: $\mathcal{F}(X) = \{([\alpha_1 v_1, \dots, \alpha_d v_d], t) \mid \alpha_i \in \{-1, 1\}\}$
(Puny et al 2022)

eigenvectors centroid

$$\frac{1}{|\mathcal{F}(x)|} \sum_{(R,t) \in \mathcal{F}(X)} \phi(XR - \mathbf{1}t) R^\top + \mathbf{1}t$$

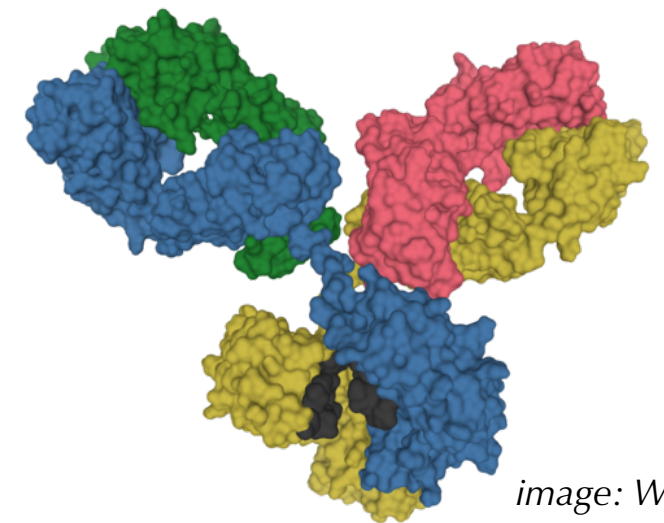
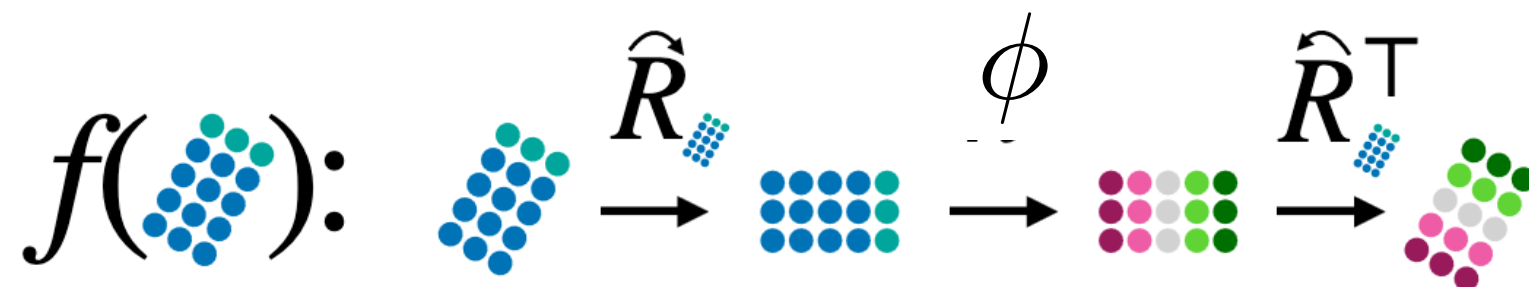


image: Wikipedia

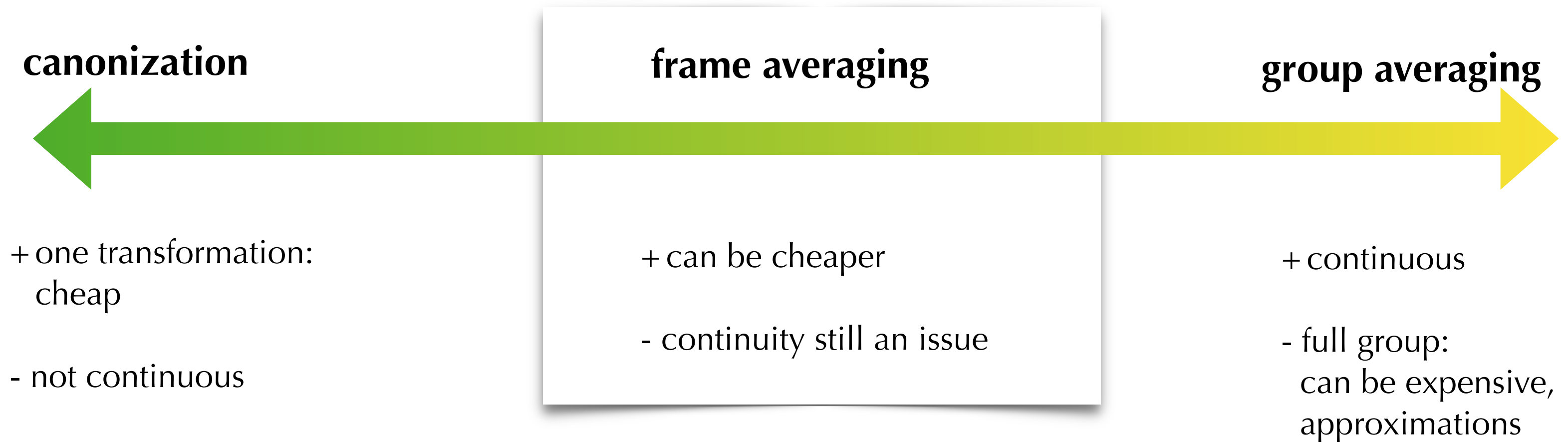
For protein binding/antibodies: also $SE(3)$
e.g. (Martinkus et al 2023, Jin et al 2023)



Do frames solve the continuity issue?

- Not in general:
 - The only continuous frame for permutations S_n on $\mathbb{R}^{d \times n}$ ($n, d > 1$) is the entire group
 - Continuous frame for $SO(2)$ in general is not finite
- But: **weighted frames** can help: smaller frames possible *(Dym et al 2024)*

Symmetrization



General advantages:

- Preserves **expressiveness** of the base model:
reuse successful existing models & training methods
- Can easily **combine symmetries** if base model is equivariant to another group
- Easy to apply

How can we get equivariant models?

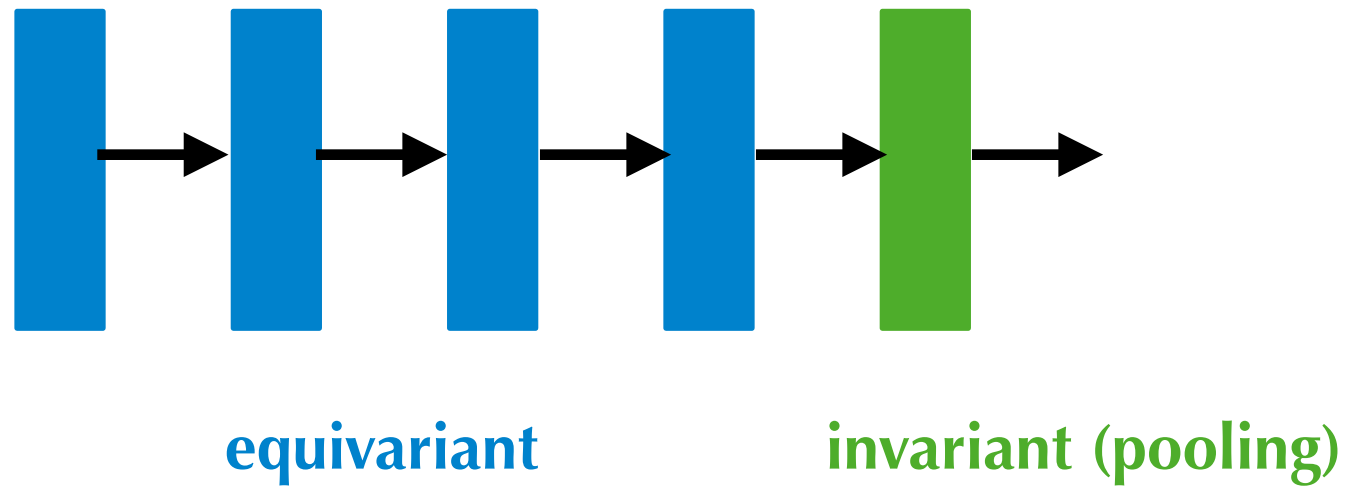
Using an off-the-shelf (non-equivariant) model:

- Data augmentation
- Canoni(cali)zation
- Group / frame averaging

Constructing equivariant models:

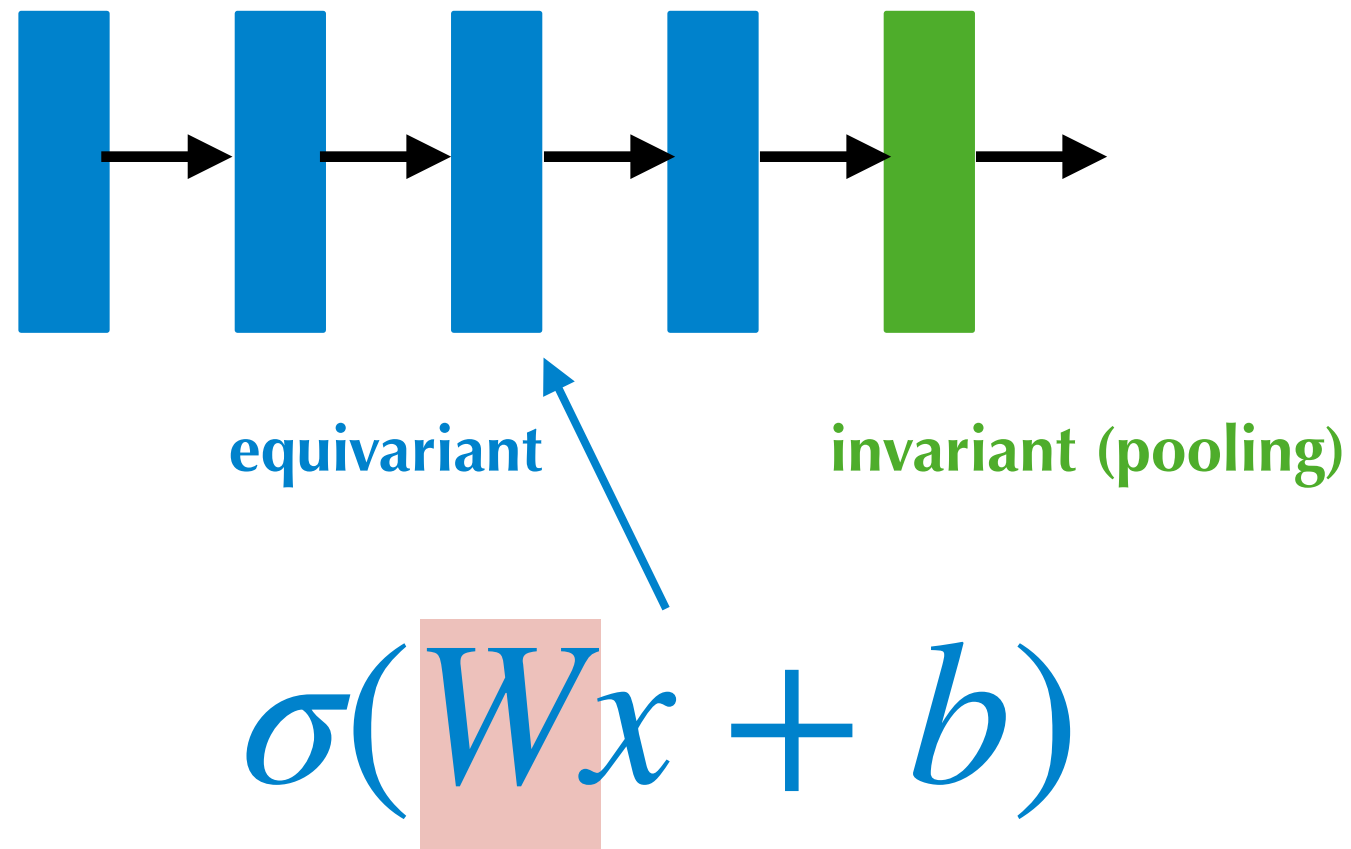
- **Linear equivariant layers + nonlinearity**: parameter sharing, group convolution
- Parameterizing representations of invariant functions (invariant theory)
- Representation theory

Generic construction



- Equivariant layers: e.g.,
 - linear equivariant + nonlinearity
 - group convolutions
 - equivariant polynomials
- e.g. *CNNs, GNNs, DeepSet, ...*

Linear equivariant layers



Symmetry corresponds to structure in W :
parameter **sharing**

Equivariance constraint for W :

$$WP_g x = P_g Wx$$

for all $g \in G$ and

$$P_g = \rho(g)$$

Linear equivariant layers

- Let G be a subgroup of the permutation group.
- Linear layer W is equivariant to G iff:

$$\begin{array}{c} \text{If} \\ (\tau(i), \tau(j)) = (k, l) \text{ for some } \tau \in G \\ \text{then} \\ W_{ij} = W_{kl} \end{array}$$

$$\text{Constraint: } P_{\tau} W P_{\tau}^{\top} = W$$

$$P_{\tau} W = W P_{\tau}$$

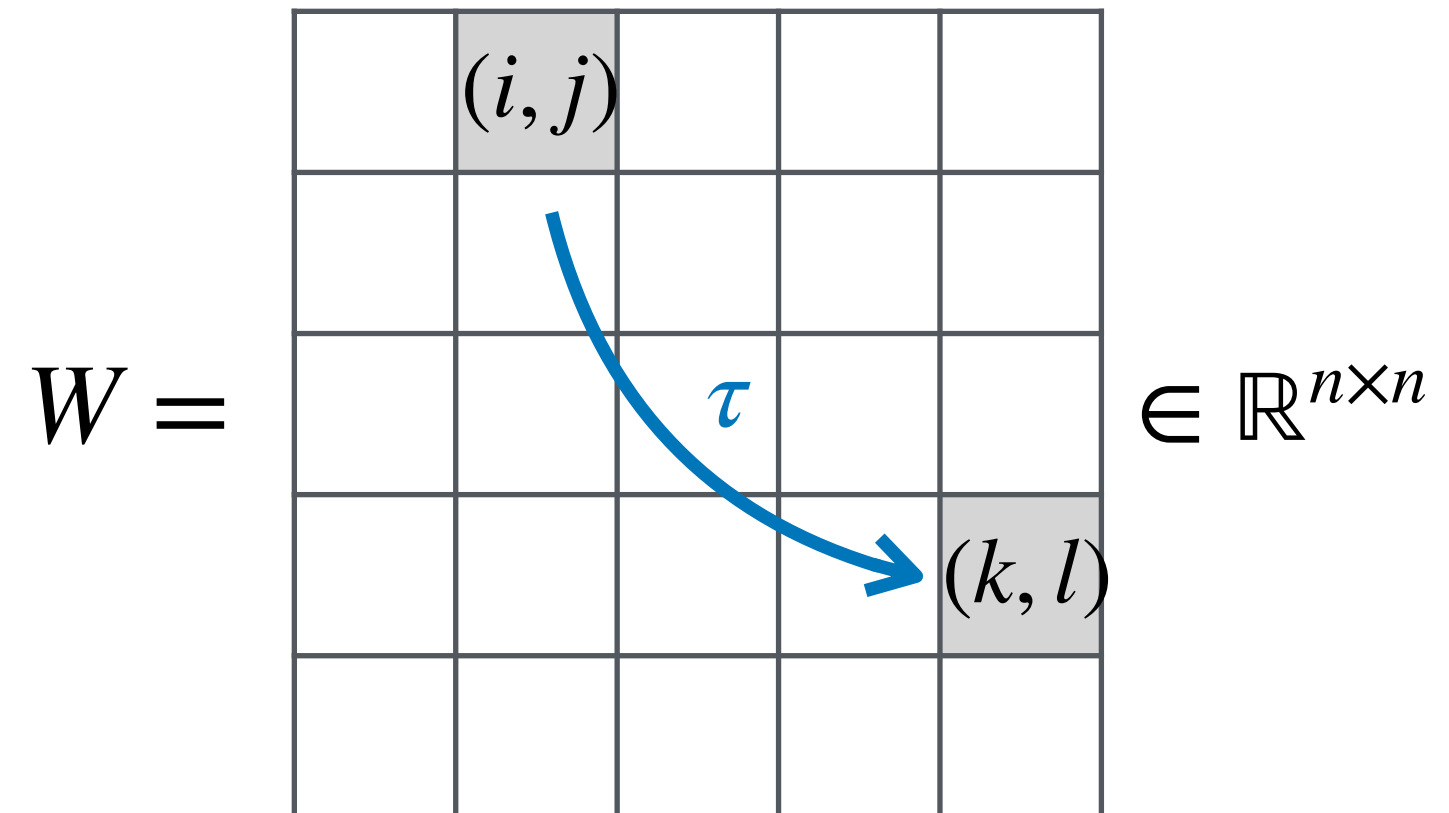


illustration: H. Maron

(Ravanbakhsh et al. 2017, Wood and Shawe-Taylor 1996, Maron et al. 2019, 2020)

Example: shift in 1D

- Group: shift (and wrap around):

$$G = C_n = \{t_0, \dots, t_{n-1}\}$$

$$t_i(j) = i + j \pmod{n}$$

- n parameters

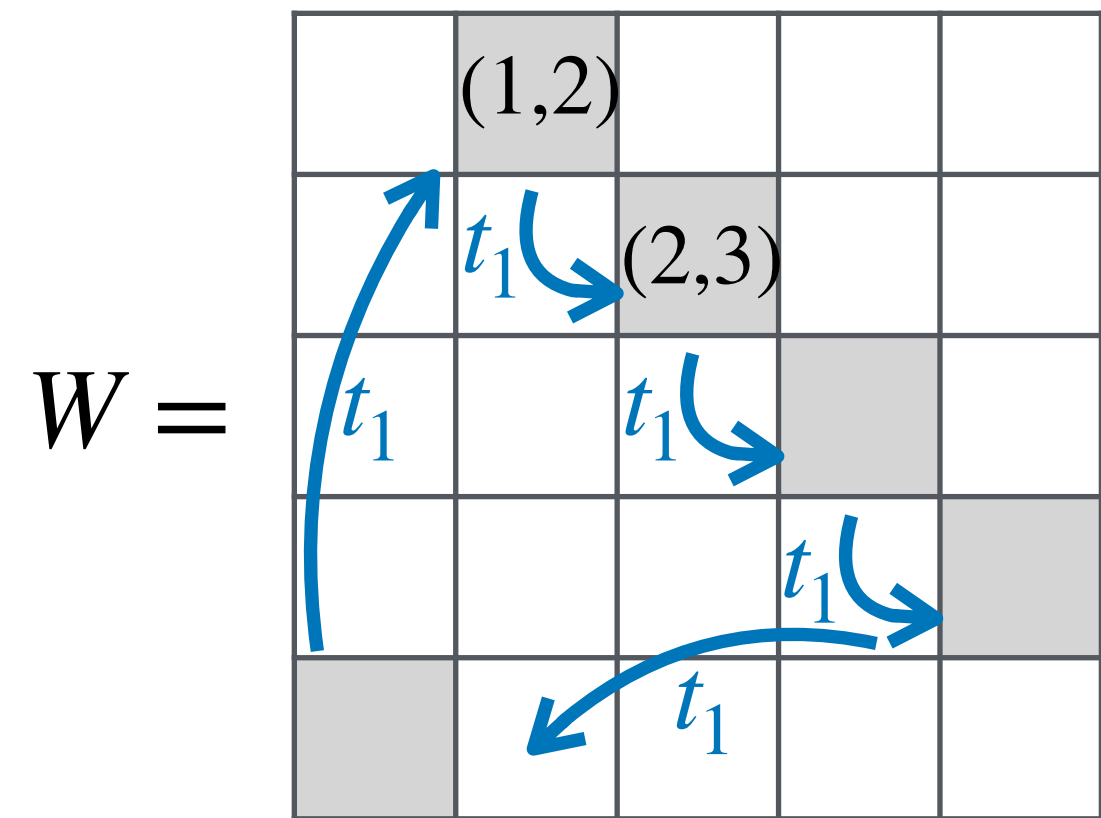


illustration: H. Maron

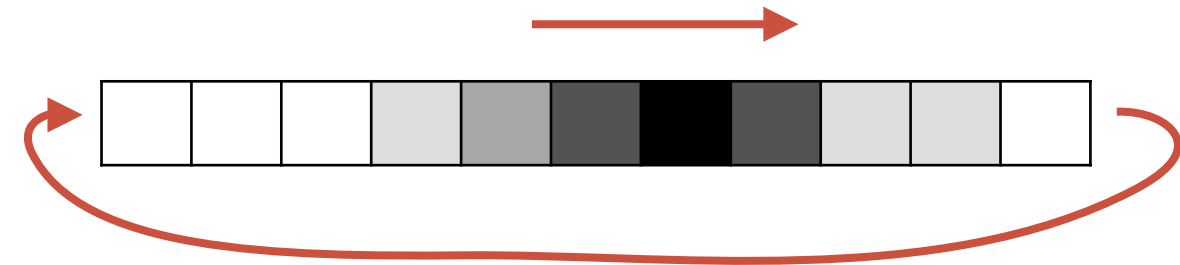
Example: shift in 1D

- Group: shift (and wrap around):

$$G = C_n = \{t_0, \dots, t_{n-1}\}$$

$$t_i(j) = i + j \pmod{n}$$

- n parameters



$W =$

illustration: H. Maron

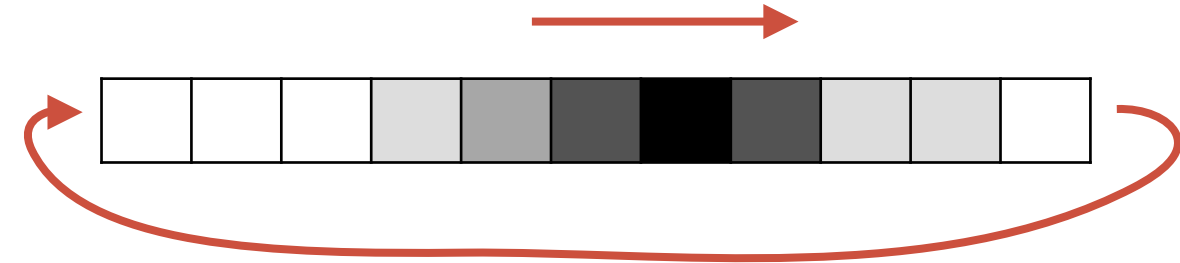
Example: shift in 1D

- Group: shift (and wrap around):

$$G = C_n = \{t_0, \dots, t_{n-1}\}$$

$$t_i(j) = i + j \pmod{n}$$

- n parameters



$W =$

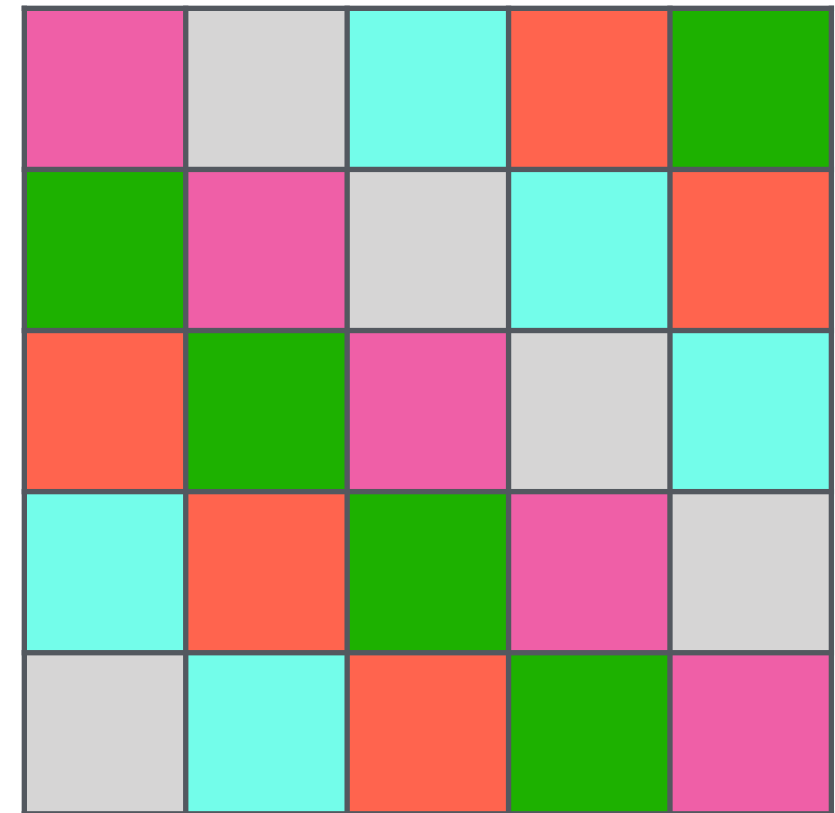
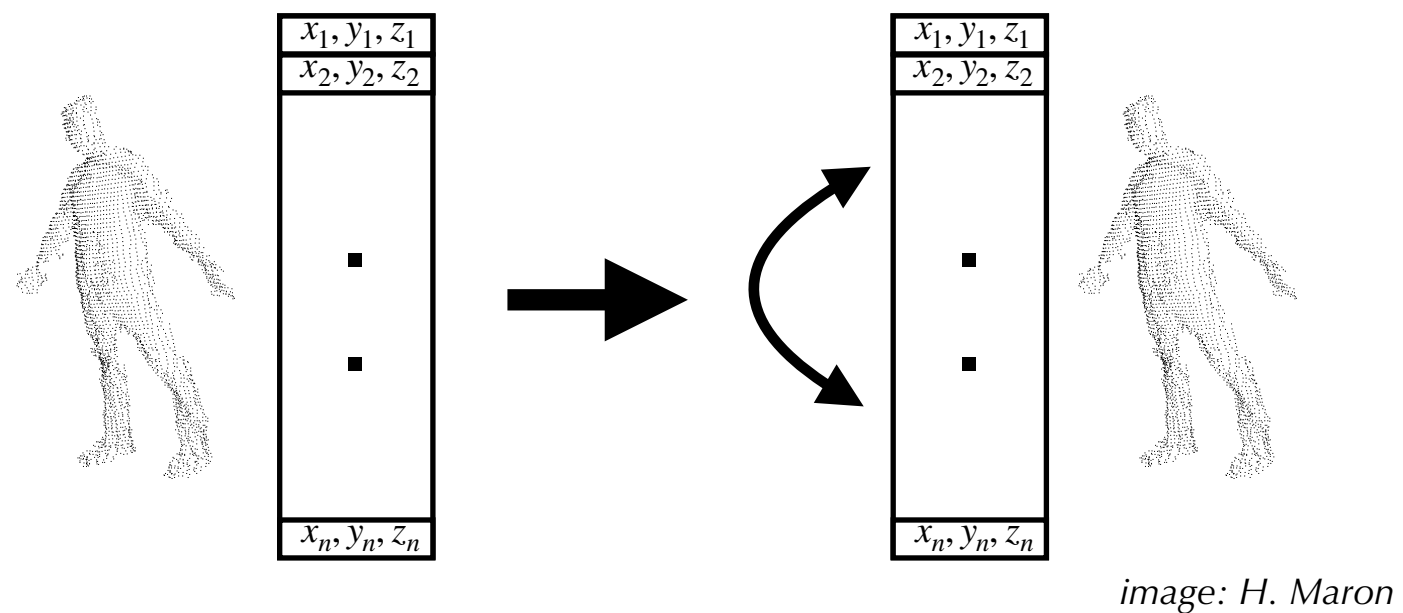
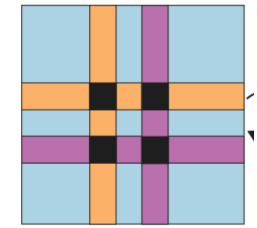


illustration: H. Maron

Example: all permutations of a set

- Input: set of vectors
- Group: all permutations
- i.e.: permute rows and columns of W simultaneously

$$\text{Constraint: } P_{\tau} W P_{\tau}^{\top} = W$$



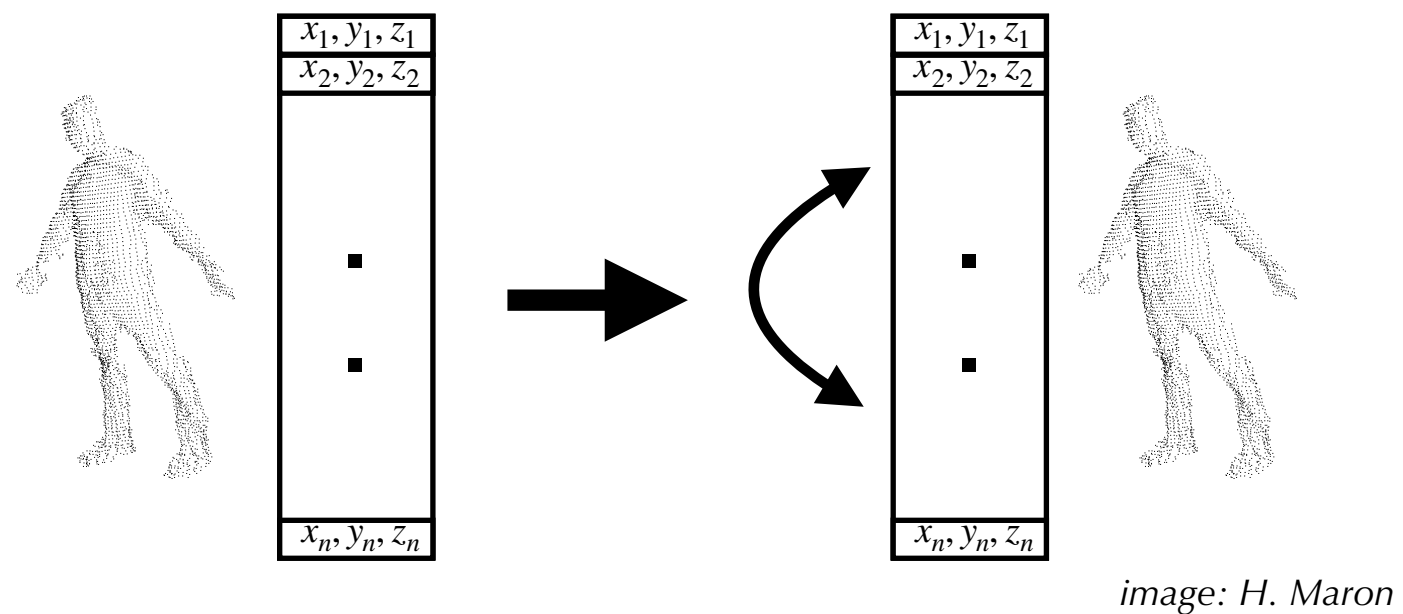
$W =$

(1,1)				
	(2,2)			

illustration: H. Maron

Example: all permutations of a set

- Input: set of vectors
- Group: all permutations
- i.e.: permute rows and columns of W simultaneously



τ with $\tau(1) = 1, \tau(2) = 3$

$W =$

illustration: H. Maron

Example: all permutations of a set

- Input: set of vectors
- Group: all permutations
- i.e.: permute rows and columns of W simultaneously

- Basis elements: identity and sum over all elements
- DeepSet, PointNet architectures
(*Zaheer et al 2017, Qi et al 2017*)

$$[F(X)]_i = \sigma\left(\alpha_1 x_i + \alpha_2 \sum_j x_j\right)$$

may use $\tilde{W}x_j$

$W =$

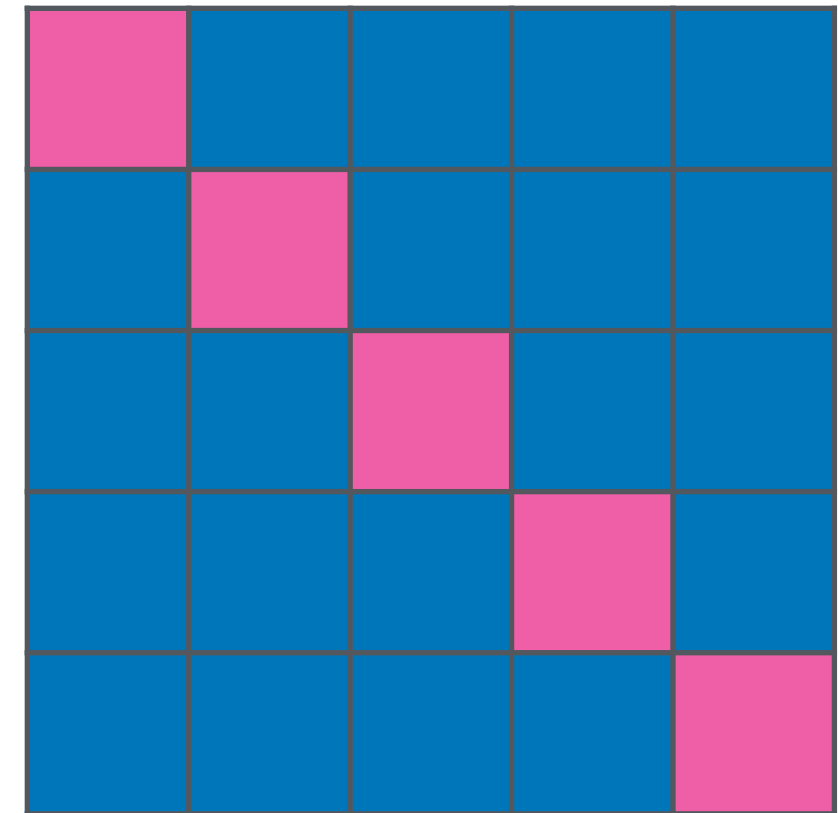
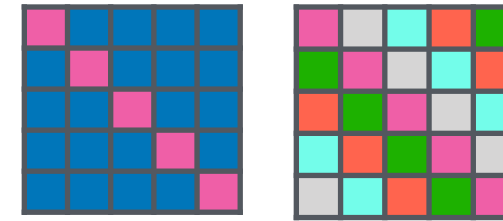


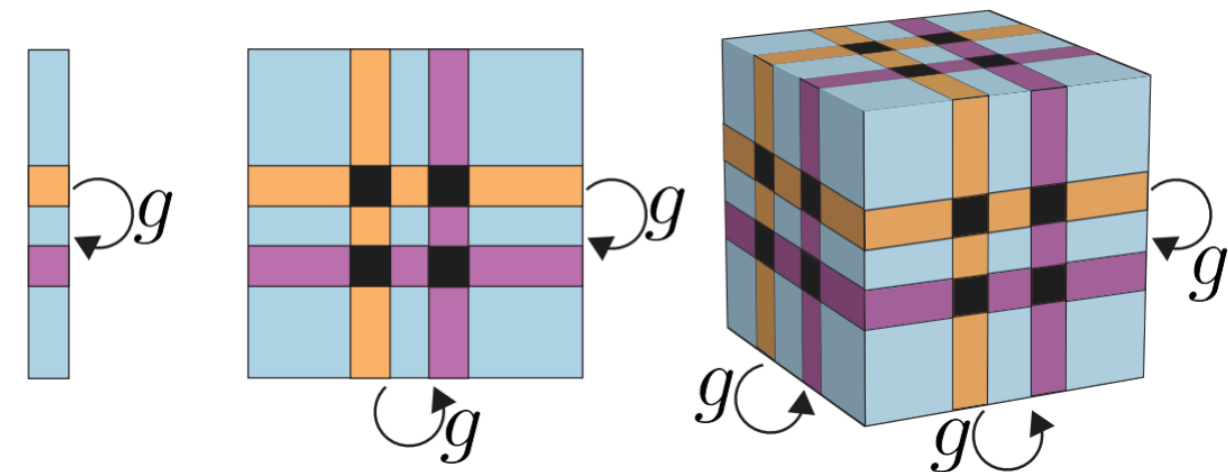
illustration: H. Maron

What do we get from this?

- Colors = basis elements: one parameter per color
- Other examples:
 - Invariant Graph networks: permutations of adjacency matrix (basis size = 15)
 - Generalizations to tensors possible: higher-order GNNs
 - Sets of symmetric elements



- Generic strategy
- But **not always applicable** or universally **expressive**



(Maron et al 2019)

Group convolution

- Regular convolution (“backward pass”) over 2D grid:

$$f \star \psi(x) := \sum_{y \in \mathbb{Z}^2} f(y) \psi(y - x)$$

sum over all translations

shifted filter

- Convolution over groups: same idea!

$$f \star \psi(g) := \sum_{h \in G} f(h) \psi(g^{-1}h)$$

sum over group

- Group equivariant convolutional neural networks

How can we get equivariant models?

Using an off-the-shelf (non-equivariant) model:

- Data augmentation
- Canoni(cali)zation
- Group / frame averaging

Constructing equivariant models:

- Linear equivariant layers + nonlinearity: parameter sharing, group convolution
- Parameterizing representations of invariant functions (invariant theory)
- Representation theory

Invariant polynomials

- Idea: approximate invariant polynomials
- Invariant polynomials from same constraints as for linear equivariant layers:

$$f(g \cdot x) = g \cdot f(x)$$

- In several relevant cases, invariant polynomials (hence functions) can be written as **functions of simple invariant scalar quantities**

“obvious” that these functions are invariant, but great that this is universal

Examples

Functions on point clouds (n points in \mathbb{R}^d) *rotations & reflections*

- Any function that is **invariant to $O(d)$** can be written as

$$f(v_1, \dots, v_n) = h\left((v_i^\top v_j)_{i,j=1}^n\right)$$

First fundamental theorem for $O(d)$

Not all inner products are needed

- Any **$O(d)$ -equivariant** function can be written as

$$f(v_1, \dots, v_n) = \sum_{t=1}^n h\left((v_i^\top v_j)_{i,j=1}^n\right) v_t$$

Can parametrize h by
standard neural network

- Similar formulations for rotations (add subdeterminants), Lorentz group (different inner product), $O(d)$ and permutations

Examples

Functions on point clouds (n points in \mathbb{R}^d)  *rotations & reflections*

- Any function that is **invariant to $O(d)$** can be written as

$$f(v_1, \dots, v_n) = h\left(\left(v_i^\top v_j\right)_{i,j=1}^n\right)$$

First fundamental theorem for $O(d)$

Not all inner products are needed

- Any **$O(d)$ -equivariant** function can be written as

$$f(v_1, \dots, v_n) = \sum_{t=1}^n h\left(\left(v_i^\top v_j\right)_{i,j=1}^n\right) v_t$$

Can parametrize h by
standard neural network

- Any **translation-invariant** function can be written as

$$f(v_1, \dots, v_n) = h(v_2 - v_1, v_3 - v_1, \dots, v_n - v_1)$$

Applications: equivariant GNNs

- Each node has node features + coordinates

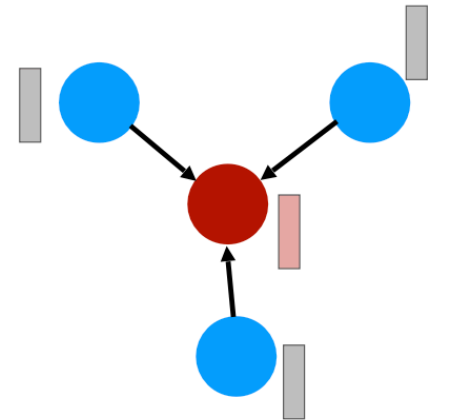
Permutation equivariance
GNN / message passing

Translation & rotation equivariance
equivariant message passing

- Equivariant updates of coordinate features:

$$x_i^{\ell+1} = x_i^{\ell} + C \sum_{j \neq i} (x_i^{\ell} - x_j^{\ell}) \phi \left(h_i^{\ell}, h_j^{\ell}, \|x_i^{\ell} - x_j^{\ell}\|^2, a_{ij} \right)$$

other node and edge features



Application: protein generation models

- **Central works in protein generation** (Chroma (*Ingraham et al 2023*), RFDiffusion (*Watson et al 2023*)) use equivariant GNNs and local coordinate systems

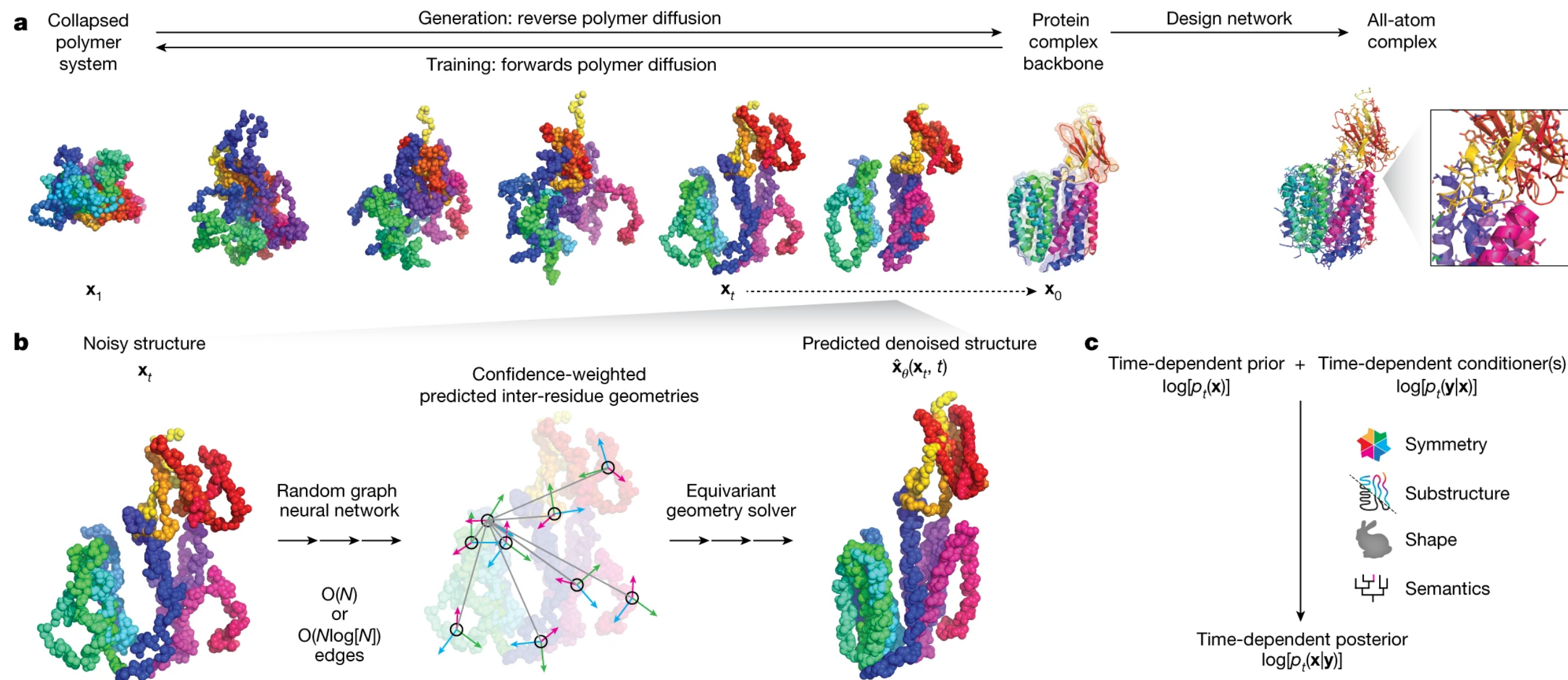


Figure: Ingraham et al 2023

How can we get equivariant models?

Using an off-the-shelf (non-equivariant) model:

- Data augmentation
- Canoni(cali)zation
- Group / frame averaging

Constructing equivariant models:

- Linear equivariant layers + nonlinearity: parameter sharing, group convolution
- Parameterizing representations of invariant functions (invariant theory)
- Representation theory

Tensor methods I

- Equivariant multi-layer perceptron (MLP):

$$f(x) = L_T \circ \sigma \circ L_{T-1} \circ \cdots \circ L_2 \circ \sigma \circ L_1(x)$$

equivariant activation

linear equivariant layer

- MLPs can **universal** approximate **any** continuous function
- Equivariant MLPs may not be universal

*Example: $x \in \mathbb{R}^2$ under 2D rotation
 $L : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ equiv. linear map identity*

Proof: Schur's lemma (representation theory) \implies linear equiv. maps are restricted

Fact: tensor products of vectors (under symmetries) have very rich representation theoretic structure

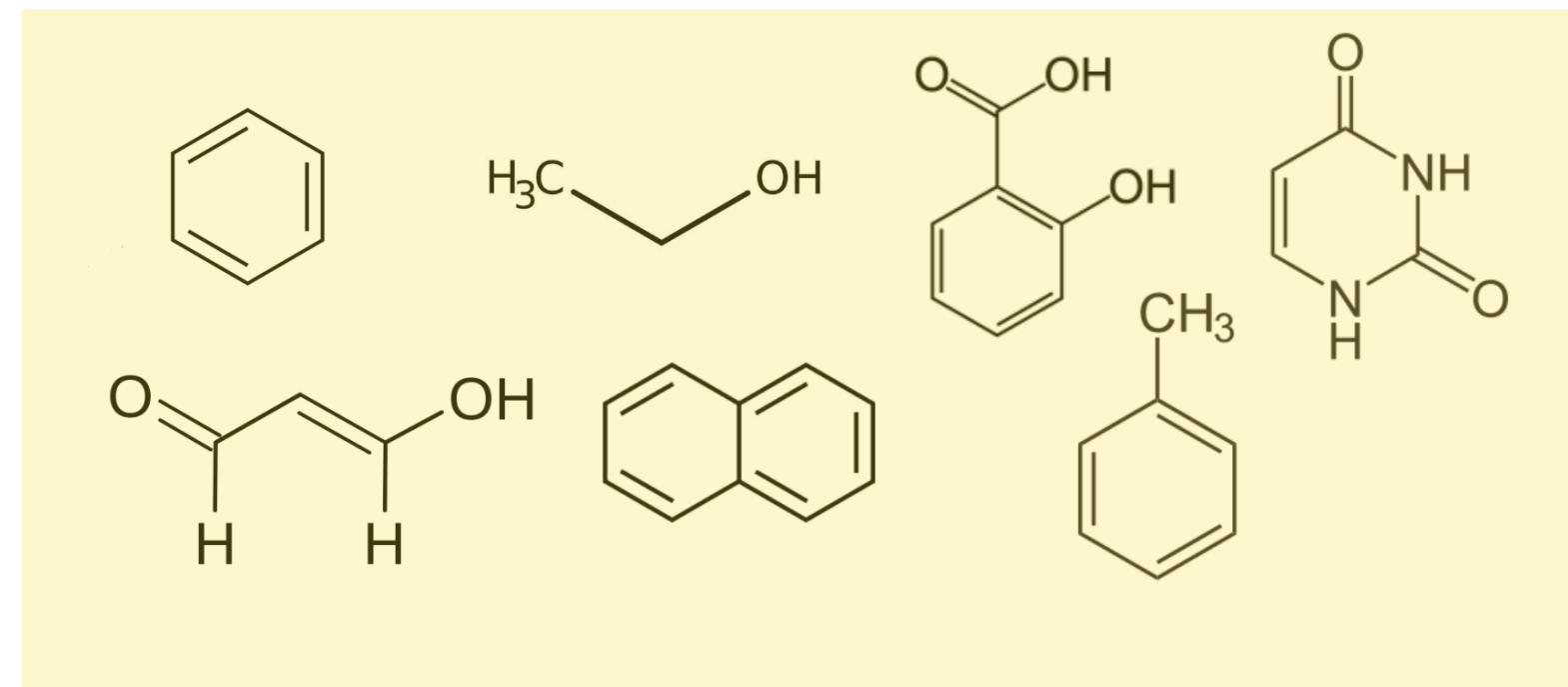
Tensor methods II

- Given $x \in \mathbb{R}^d$ consider its **tensor power** $x^{\otimes k} \in (\mathbb{R}^d)^k$
- Use an equivariant MLP on $x^{\otimes k} \implies$ universality (for large enough $k \in \mathbb{N}$)
- Challenge: how to find/decompose the group representation on $(\mathbb{R}^d)^k$
 - Easy for rotations/reflections $SO(d)$
 - Hard for arbitrary finite groups
 - Clebsch–Gordan problem in representation theory

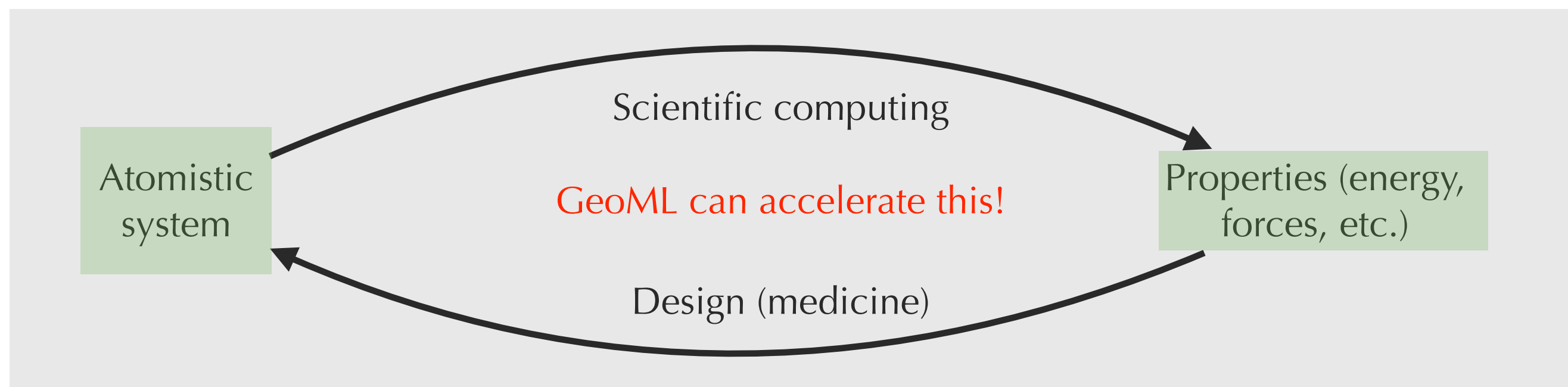
Atomistic systems and geometric ML I

- Atomistic systems: a **graph** of atoms
- Often a lot of symmetries

- Translation
- Rotation (change of basis)
- Permutation



- Goal: advancing atomistic science with geometric ML



Atomistic systems and geometric ML II

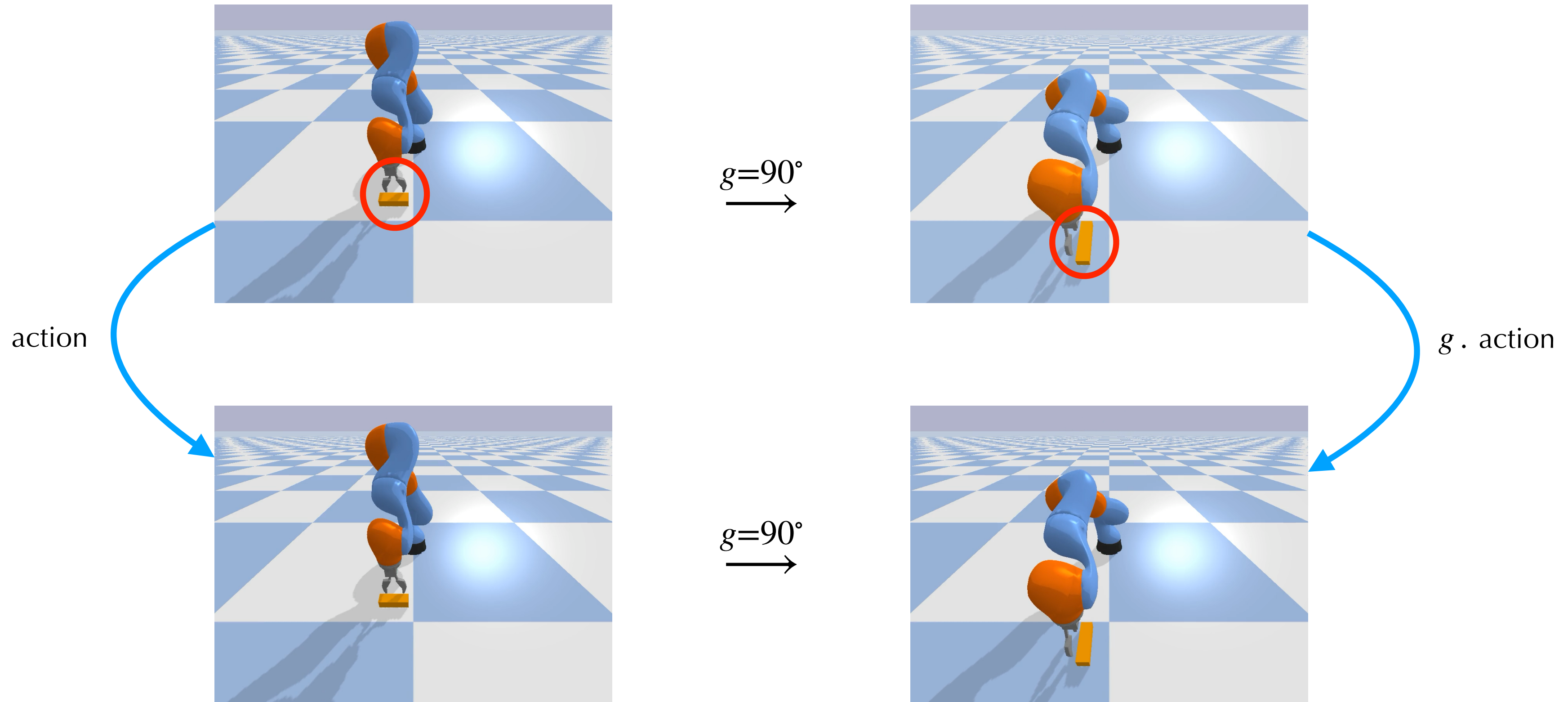
Core methods:

- NequIP (neural equivariant interatomic potentials):
 - $E(3)$ (Euclidean) **message-passing** graph neural network
 - Potential/energy prediction
 - Force prediction is for free!

$$\text{Force} = \frac{\partial \text{energy}}{\partial \text{position}} \implies \text{auto-grad!}$$

- Equivariant **transformer** (equiformer): equivariant graph transformers

Symmetries in robotics I



Symmetries in robotics II

- Under equivariance, Markov Decision Processes (MDPs) have equivariant optimal policies π^\star :

$$\pi^\star(g \cdot \text{state}) = g \cdot \pi^\star(\text{state}), \quad \forall g \in G$$

- Leverage equivariant architectures to robotics
- $SO(2)$ and $SE(3)$ rotational/translational symmetries

Dian Wang, Robin Walters, Xupeng Zhu, and Robert Platt. "Equivariant Q-Learning in Spatial Action Spaces." *Conference on Robotics Learning (CoRL)*, 2021.

Dian Wang, Robin Walters, Robert Platt, "SO(2)-Equivariant Reinforcement Learning." *ICLR*, 2022.

Xupeng Zhu, Dian Wang, Guanang Su, Ondrej Biza, Robin Walters, Robert Platt. "On Robot Grasp Learning Using Equivariant Models." *Autonomous Robots*. 2023.

Xupeng Zhu, Dian Wang, Ondrej Biza, Guanang Su, Robin Walters, Robert Platt, "Sample Efficient Grasp Learning Using Equivariant Models," *RSS* 2022.

Haojie Huang, Dian Wang, Xupeng Zhu, Robin Walters, Robert Platt, "Edge Grasp Network: Graph-Based SE(3)-invariant Approach to Grasp Detection, *ICRA*, 2023.

Dian Wang, Mingxi Jia, Xupeng Zhu, Robin Walters, Robert Platt, "On-Robot Learning With Equivariant Models," *CoRL*, 2022.

Symmetry breaking

- Equivariant architectures cannot **break** symmetry:

Proof: input x and output y such that $y = f(x)$

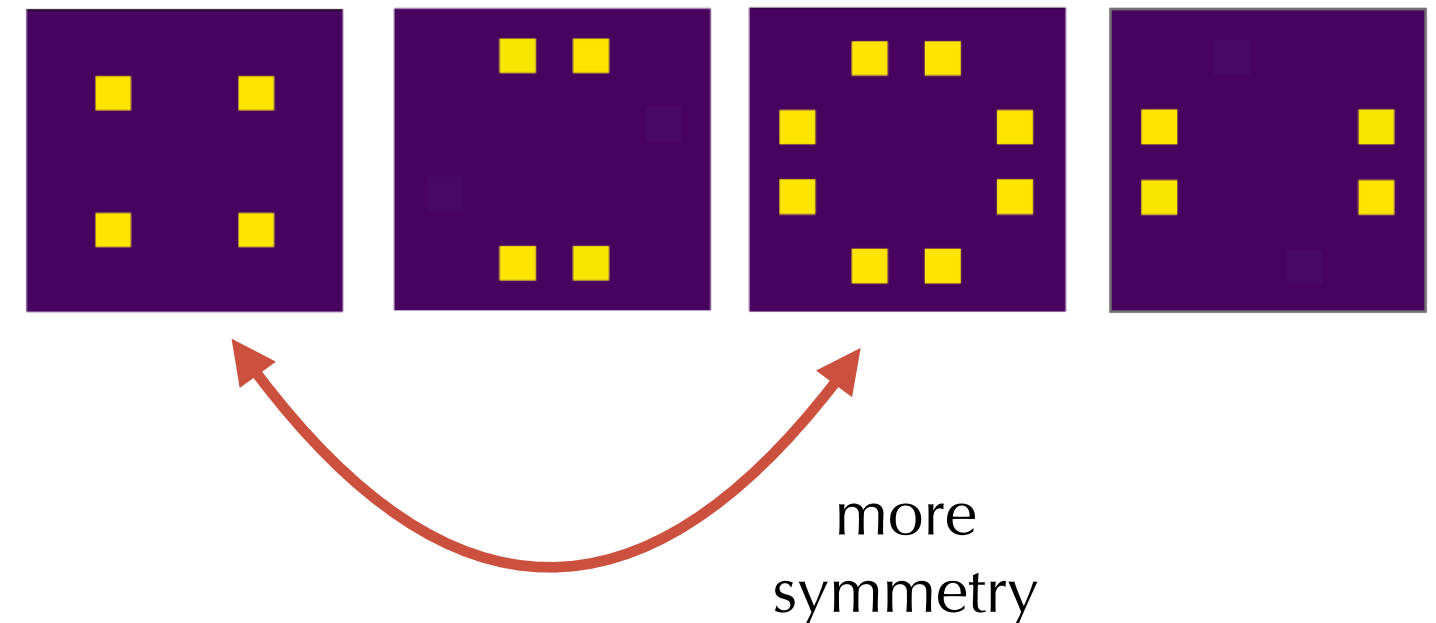
$f(\cdot)$ is equivariant: $g^{-1} \cdot f(g \cdot x) = f(x) \quad \forall g \in G$

$G_x = \{g : g \cdot x = x\} \subseteq G$ the isotropy group

$$g \in G_x \implies g \cdot y = g \cdot f(x) = g \cdot g^{-1} f(g \cdot x) = f(g \cdot x) = f(x) = y$$

$$G_x \subseteq G_y$$

- Output is always **more symmetric!**
- How to break?
 - Weight perturbation, gradients, sets



Roadmap

Part 1

- Introduction and basics
- Techniques for equivariance, with examples

Invariance: $f(g \cdot x) = f(x)$
Equivariance: $f(g \cdot x) = g \cdot f(x)$

Part 2

- Neural parameter symmetries and other recent directions
- Theory results and directions
- A bit of discussion

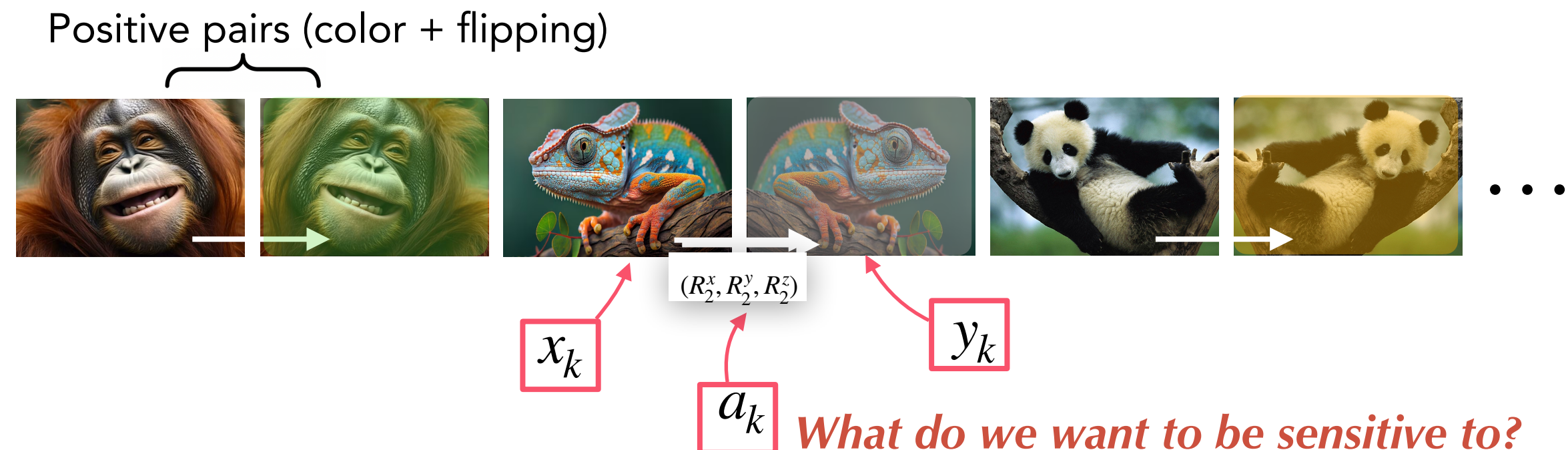
Disclaimer:
This is a tutorial
and hence
cannot cover all works

Flexible symmetries: adaptivity

- Hard-coding symmetries a priori can be restrictive: foundation models??
- Instead: **encode multiple symmetries jointly** and **adapt** to input data

Examples:

- Contextual World models: in-context selection of a subset of symmetries (*Gupta et al 2024*)



Flexible symmetries: adaptivity

- Hard-coding symmetries a priori can be restrictive: foundation models??

► Instead: **encode multiple symmetries jointly** and **adapt** to input data

Examples:

- Contextual World models: in-context selection of a subset of symmetries (*Gupta et al 2024*)
- Any-subgroup equivariant networks: symmetry-breaking special input selects subgroup (*Goel et al 2025*)

$$f(x) = h(x, \mathbf{v})$$

if $g \cdot \mathbf{v} = \mathbf{v}$, then $f(g \cdot x) = g \cdot f(x)$

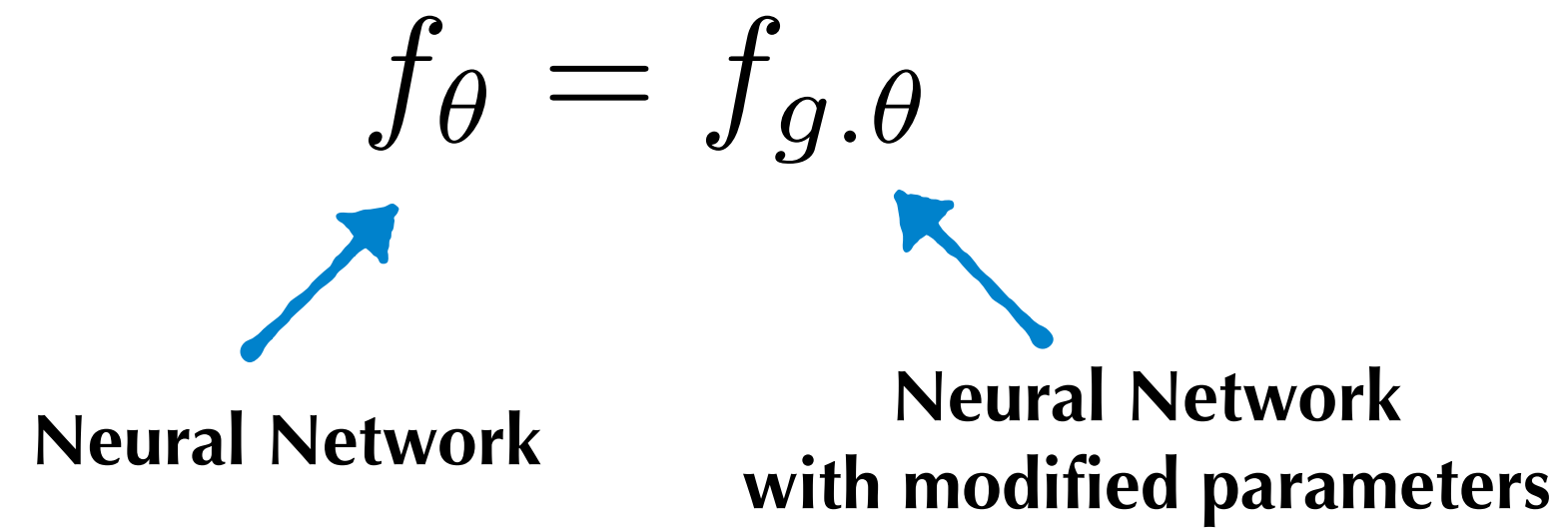
Flexible symmetries: symmetry discovery

- What if I don't know the symmetries a priori? **Learn them!**

Examples:

- Learn input data transformations / augmentations (supervised or unsupervised)
(Benton et al 2020, Desai et al 2022, Yang et al 2023, Yang et al 2024, Santos Escriche & J 2025)
- Learn architectural constraints (e.g., weight sharing, weighted convolution)
(Dehmamy et al, Zhou et al 2021, Romero & Lohit 2023, van der Linden et al 2023,...)
- Probe trained networks
(Krippendorf & Syvaeri 2020, Moskalev et al 2023)

Neural Parameter Symmetries

$$f_{\theta} = f_{g.\theta}$$


Neural Network

Neural Network
with modified parameters

Neural Parameter Symmetries

$$f_{\theta} = f_{g.\theta}$$

Example: MLPs $f_{W_1, W_2}(x) = W_2 \sigma(W_1 x)$

- **Scale invariance** (homogeneous activation fcts):

$$W_2 \sigma(W_1 x) = \alpha W_2 \sigma\left(\frac{1}{\alpha} W_1 x\right)$$

- **Permutation invariance** (pointwise activation fcts):

$$W_2 \sigma(W_1 x) = W_2 P^{\top} \sigma(P W_1 x)$$

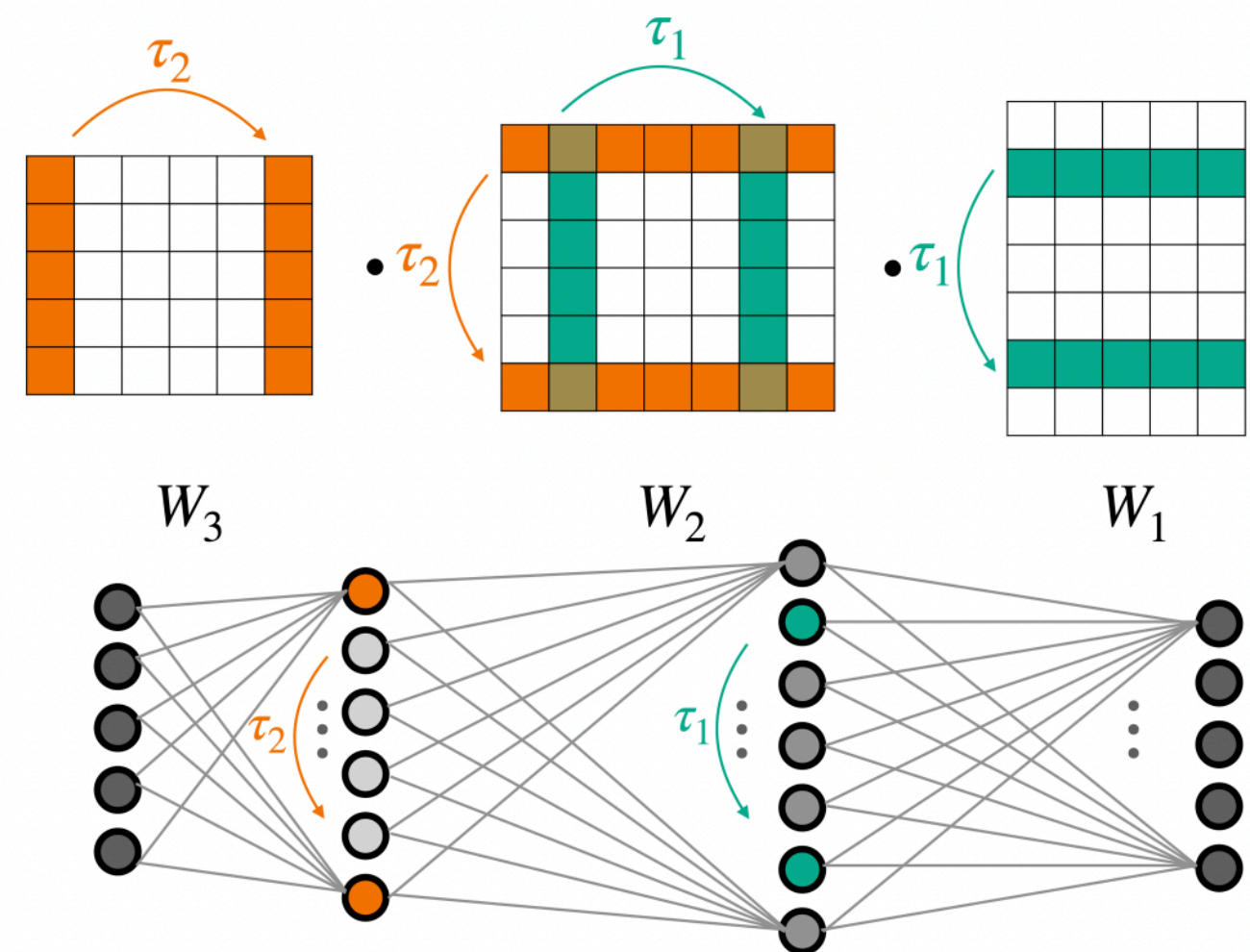


Figure: Navon et al. 2023

Neural Parameter Symmetries

$$f_{\theta} = f_{g.\theta}$$

Example: matrix products (LoRAs, Attention)

$$W_i \mapsto W_i + U_i V_i^{\top} \quad \text{softmax} \left(\frac{1}{\sqrt{d}} Q W^Q (W^K)^{\top} K \right) V W^V W^O$$

- **GL(r)-invariance**

$$U R R^{-1} V^{\top} = U V^{\top}$$



any invertible matrix R

Neural Parameter Symmetries

$$f_{\theta} = f_{g.\theta}$$

... and many more in other general components: normalization, softmax, ..

Example: some symmetries in **Transformers**

- Symmetries within MLPs
- Permutation of Attention Heads
- RMSNorm (residual): orthogonal group
- Attention and LoRAs (matrix product): GL(r)-invariance

Types of symmetries

- Same neural function: $f_{\theta} = f_{g.\theta}$
- Same loss: $L(f_{\theta}) = L(f_{g.\theta})$
*e.g. contrastive loss uses $f(x)^{\top} f(x')$:
rotated output gives same loss*
- Invariance only for a subset of data or in expectation over a data distribution

Why do we care?

1. **Affects loss landscape:** optimization properties and model merging

- Local optima
- (Linear) mode connectivity

- ▶ Exploit this for
 - ▶ optimization
 - ▶ model merging
 - ▶ Bayesian inference
 - ▶ generalization analysis

**Symmetries can
help / hinder /
obfuscate!**

2. **Neural networks as data:** weight space learning

Symmetry in Neural Network Parameter Spaces

Bo Zhao
University of California, San Diego
bozhao@ucsd.edu

Robin Walters
Northeastern University
r.walters@northeastern.edu

Rose Yu
University of California, San Diego
roseyu@ucsd.edu

ICLR 2025 Workshop on Weight Space Learning

Neural Network Weights as a New Data Modality

Loss landscapes and parameter symmetries

$$L(f_{\theta}) = L(f_{g.\theta})$$

If θ is a (local) minimum, then so is $g.\theta$

- Affects level sets, saddle points, minima,...

Continuous* symmetries:

- **Mode connectivity**: minima joined by continuous low-loss curves in parameter space
- (Partial) explanation for empirical observations!
(*Garipov et al 2018, Draxler et al 2018*)
- Useful for model averaging / ensembling

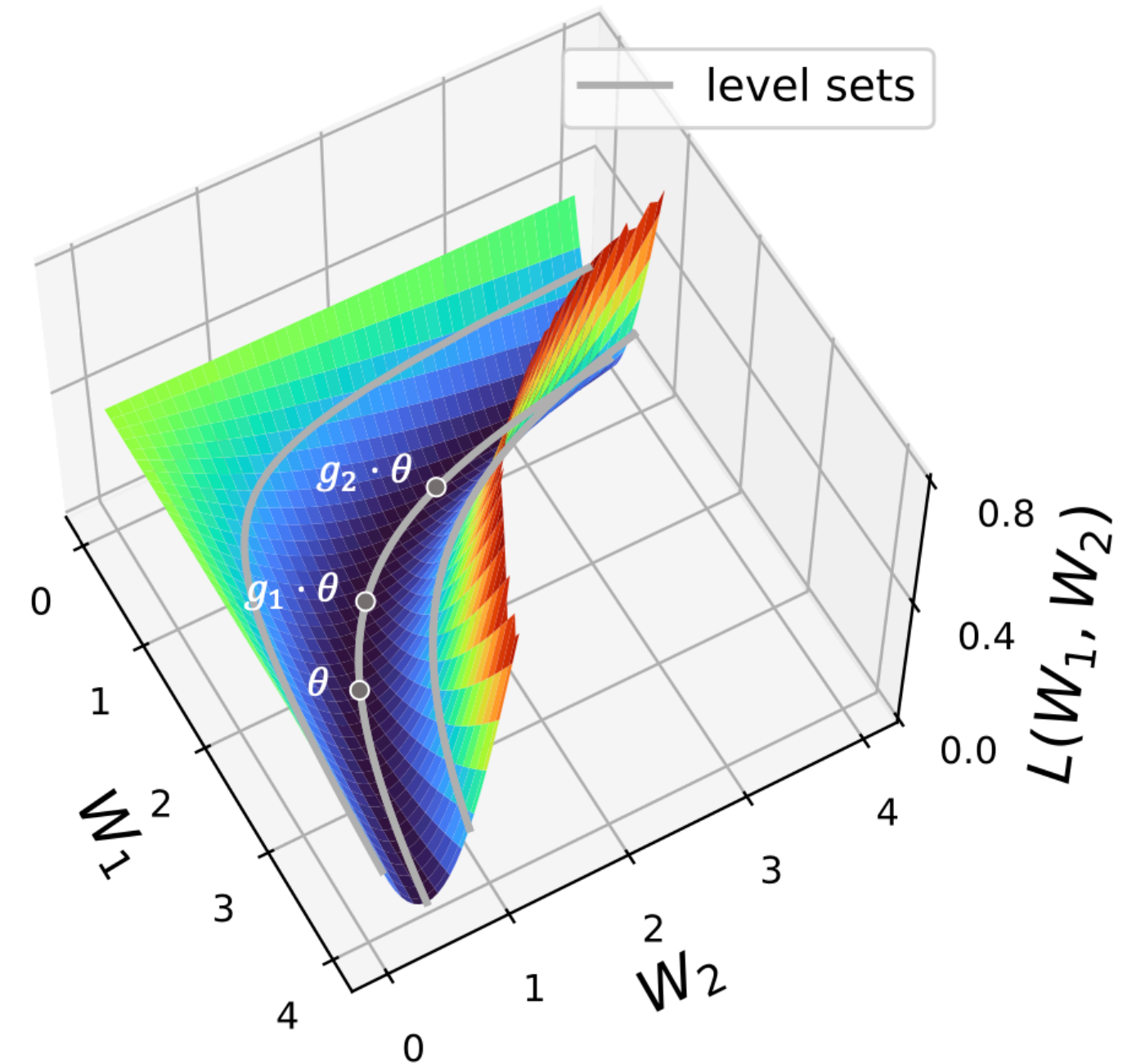


figure: Zhao et al 2025

*Lie group with continuous group action

Loss landscapes and parameter symmetries

$$L(f_{\theta}) = L(f_{g.\theta})$$

If θ is a (local) minimum, then so is $g.\theta$

- Affects level sets, saddle points, minima,...

Discrete symmetries:

- Replica of minima, not necessarily connected
- Alignment via permutation gives
linear mode connectivity

(Frankle et al 2020, Singh & Jaggi 2020, Wang et al 2020, Tatro et al 2020, Mirzadeh et al 2021, Entezari et al 2021, Yunis et al 2022, Ainsworth et al 2022, Guerrero Peña et al 2023, Navon et al 2023, Verma & Elbayad 2024, Ferbach et al 2024, Lim et al 2024, Theus et al 2025)

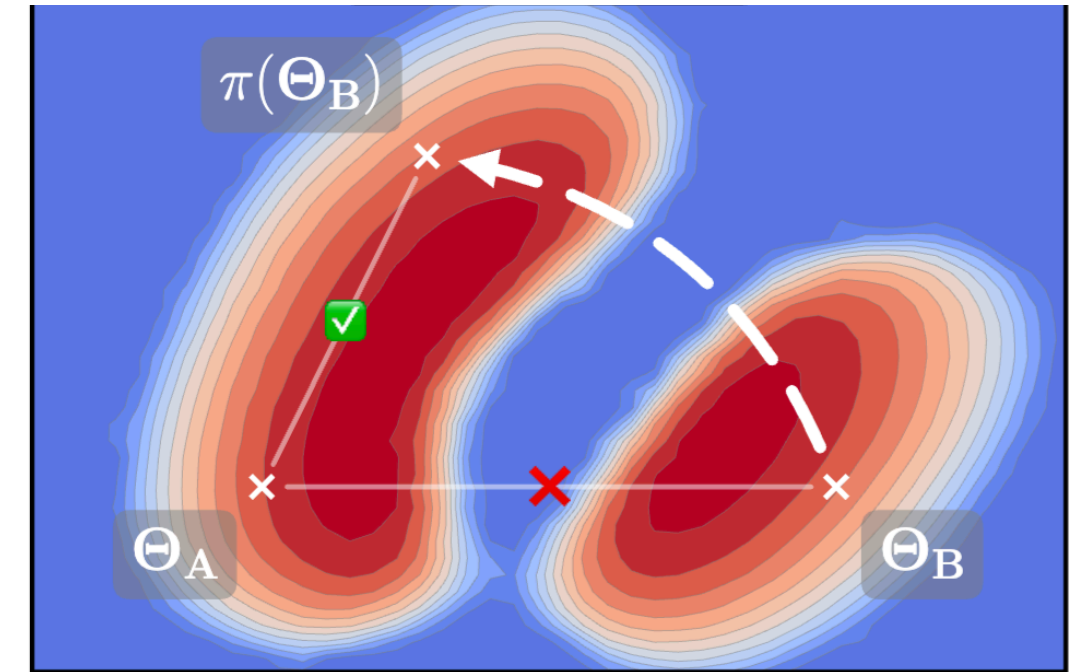


Figure: Ainsworth et al 2023

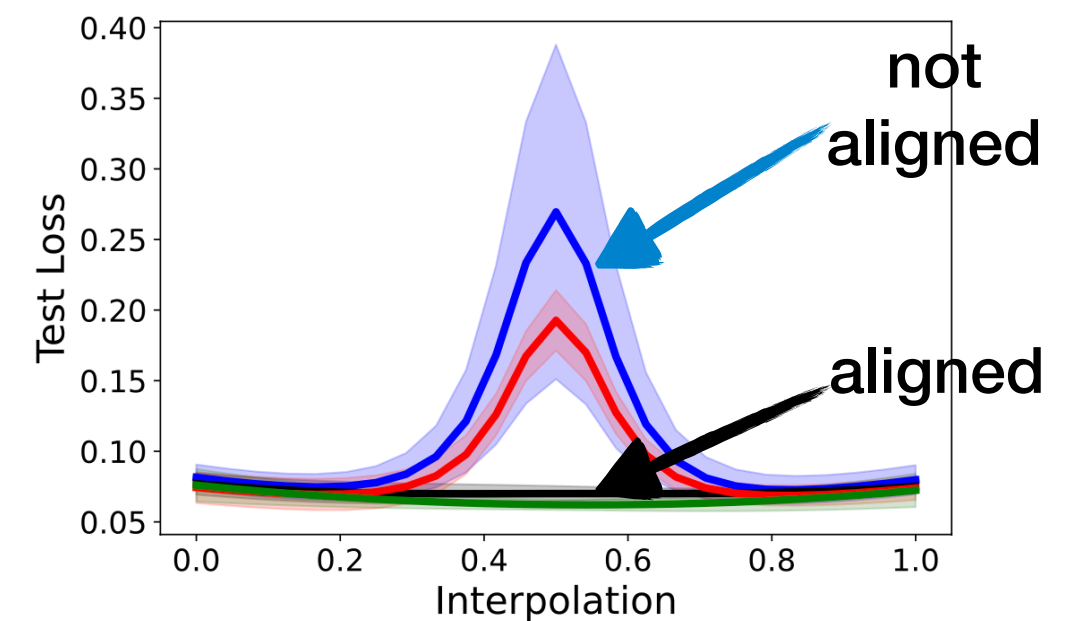
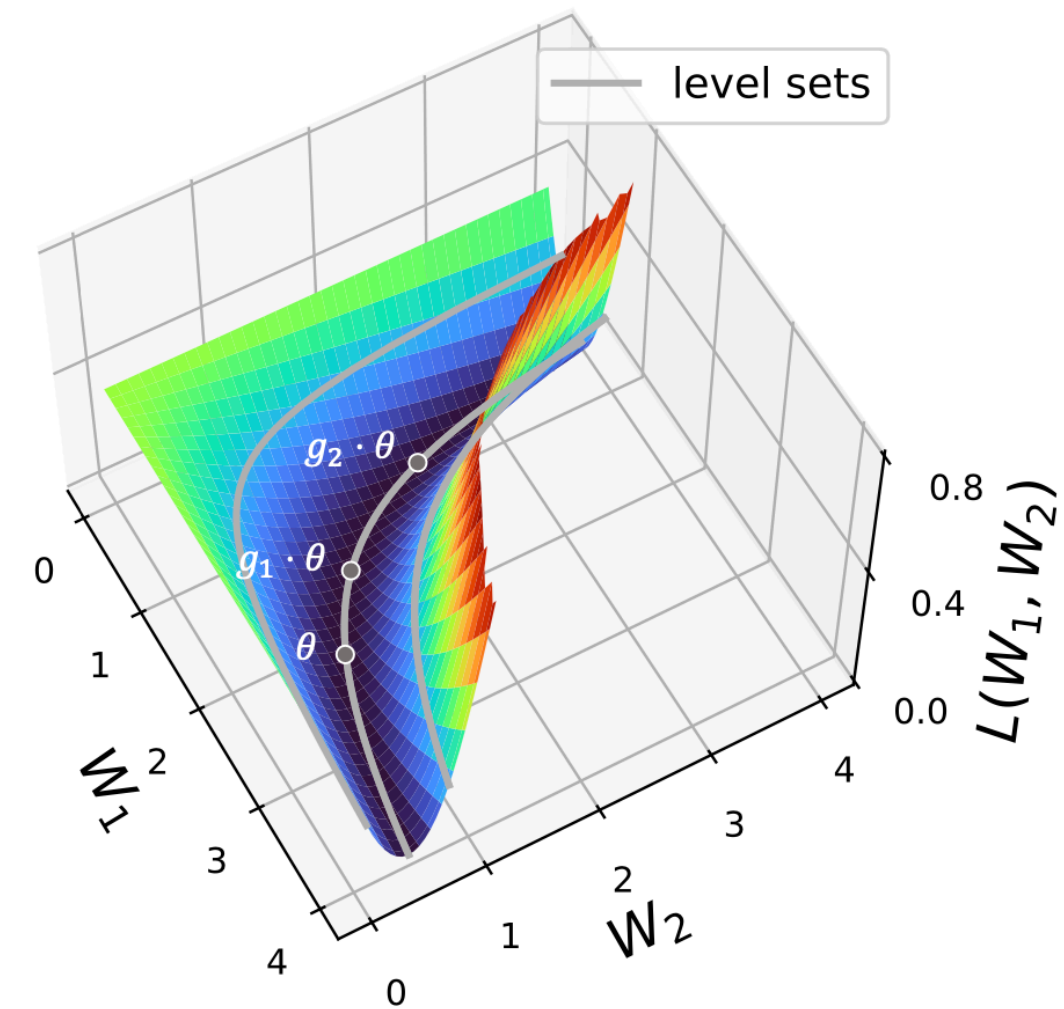


figure: Lim et al 2024

Implications for optimization

- **Learning dynamics** at θ and $g \cdot \theta$ may be **different** (e.g., norms, gradients, curvature,...)
(*Van Laarhoven 2017, Tanaka & Kunin 2021*)
- Given θ , search over $g \in G$ for a good point $g \cdot \theta$
(*Bamler & Mandate 2018, Stock et al 2019, Saul 2023, Armenta et al 2023, Zhao et al 2022, 2024*)
- Invariant optimization algorithms (e.g. Path-SGD for scale invariance)
(*Neyshabur et al 2015, Badrinarayanan et al 2015, Meng et al 2019, Huang et al 2020, Yi 2022, Kristiadi et al 2023*)



Other implications

- **Removing symmetries** simplifies landscape for **optimization, Bayesian inference** (*Hecht-Nielsen 1990, Leake & Vishnu 2021, Wiese et al 2023, Xiao et al 2023, Laurent et al 2024, Lim et al 2024*)
- Symmetries lead to **conserved quantities** during optimization: help analysis of **optimization dynamics and generalization** (e.g. imbalance)

$$W_\ell W_\ell^\top - W_{\ell+1}^\top W_{\ell+1}$$

Why do we care?

1. **Affects loss landscape:** optimization properties and model merging

- Local optima
- (Linear) mode connectivity

- ▶ Exploit this for
 - ▶ optimization
 - ▶ model merging
 - ▶ Bayesian inference
 - ▶ generalization analysis

2. **Neural networks as data:** weight space learning

Symmetry in Neural Network Parameter Spaces

Bo Zhao
University of California, San Diego
bozhao@ucsd.edu

Robin Walters
Northeastern University
r.walters@northeastern.edu

Symmetries can
help / hinder /
obfuscate!

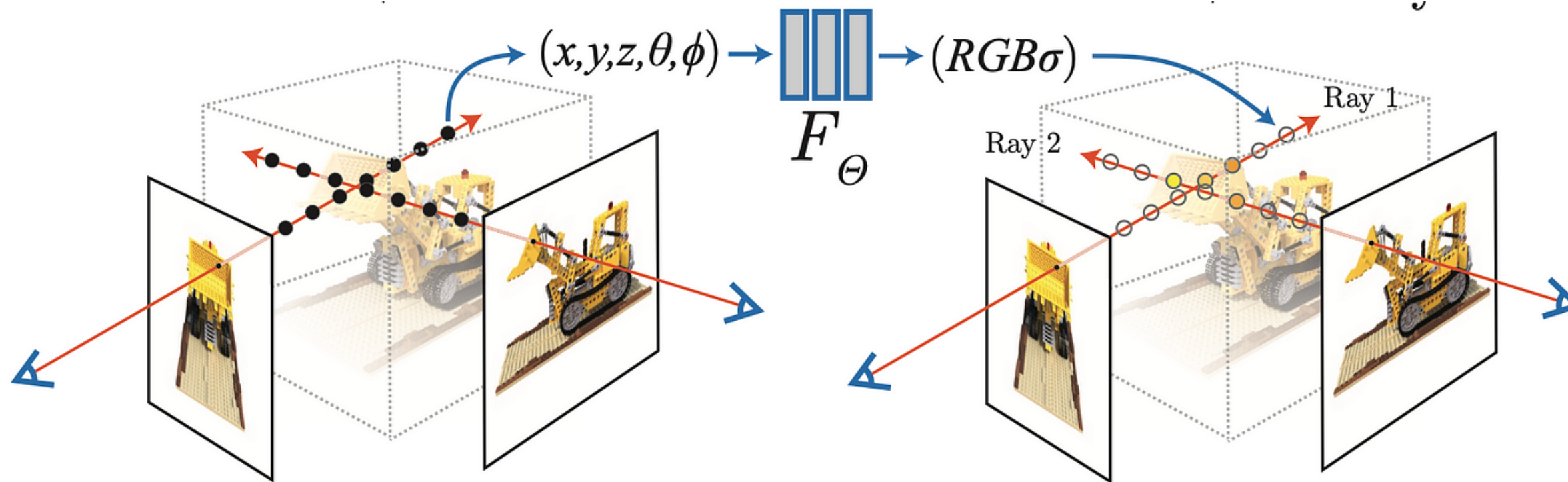
Metanetworks / neural functionals:
neural networks that take NN (weights)

Goals:
analyze, understand, edit
model collections

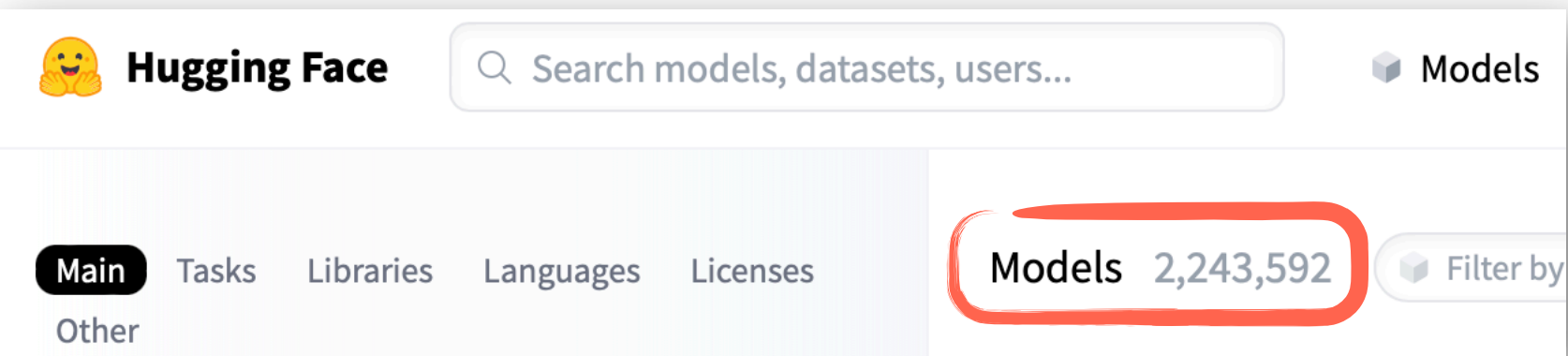
Challenges:
scale
symmetries

Neural networks as data: model populations

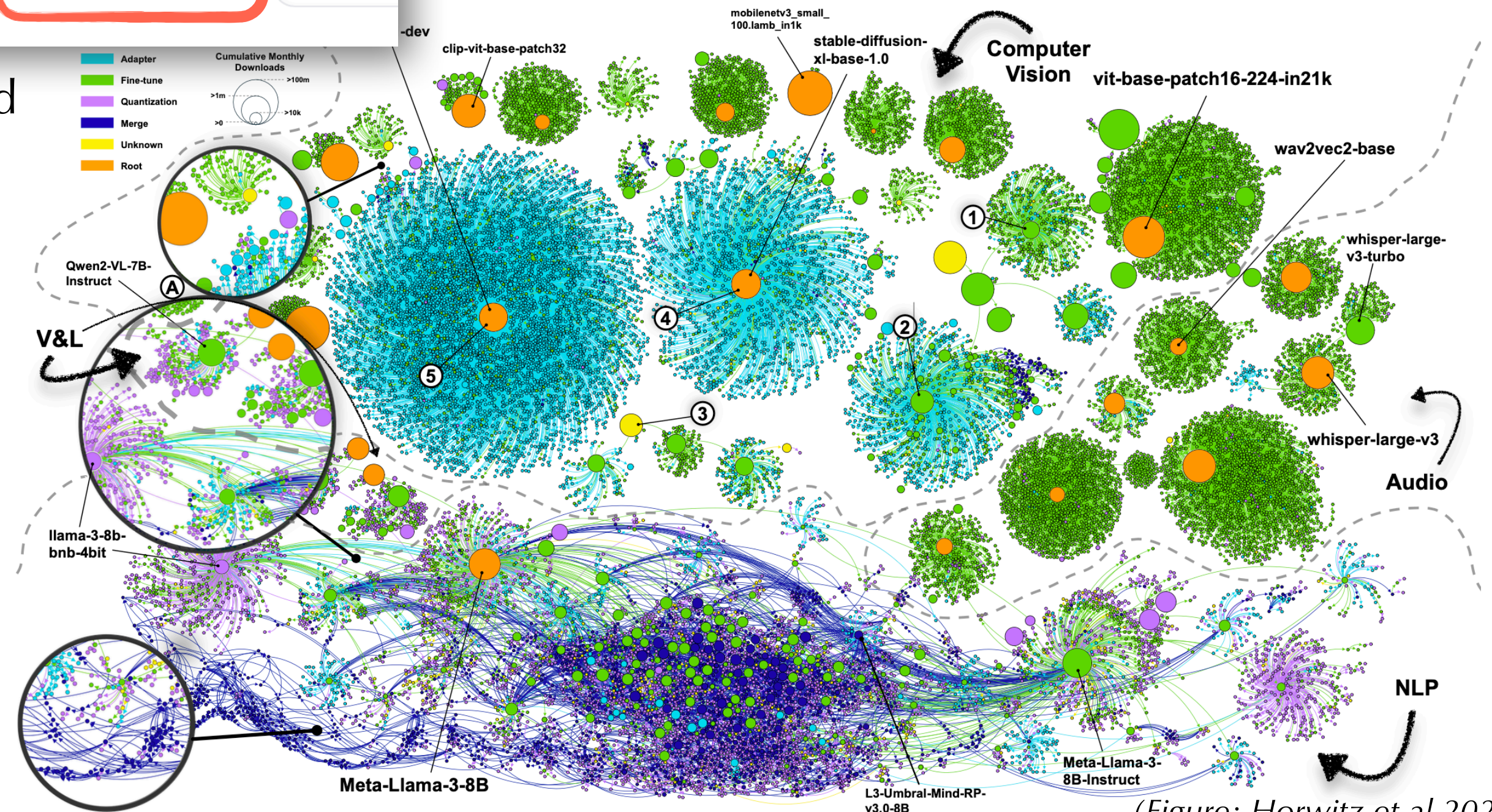
- Networks that represent data samples:
implicit neural representations (INRs), neural radiance fields (NeRFs)
- Dataset of 3D scenes / objects is a dataset of neural networks
- **Goal:** edit / classify / generate INRs



Neural Networks as Data: model populations



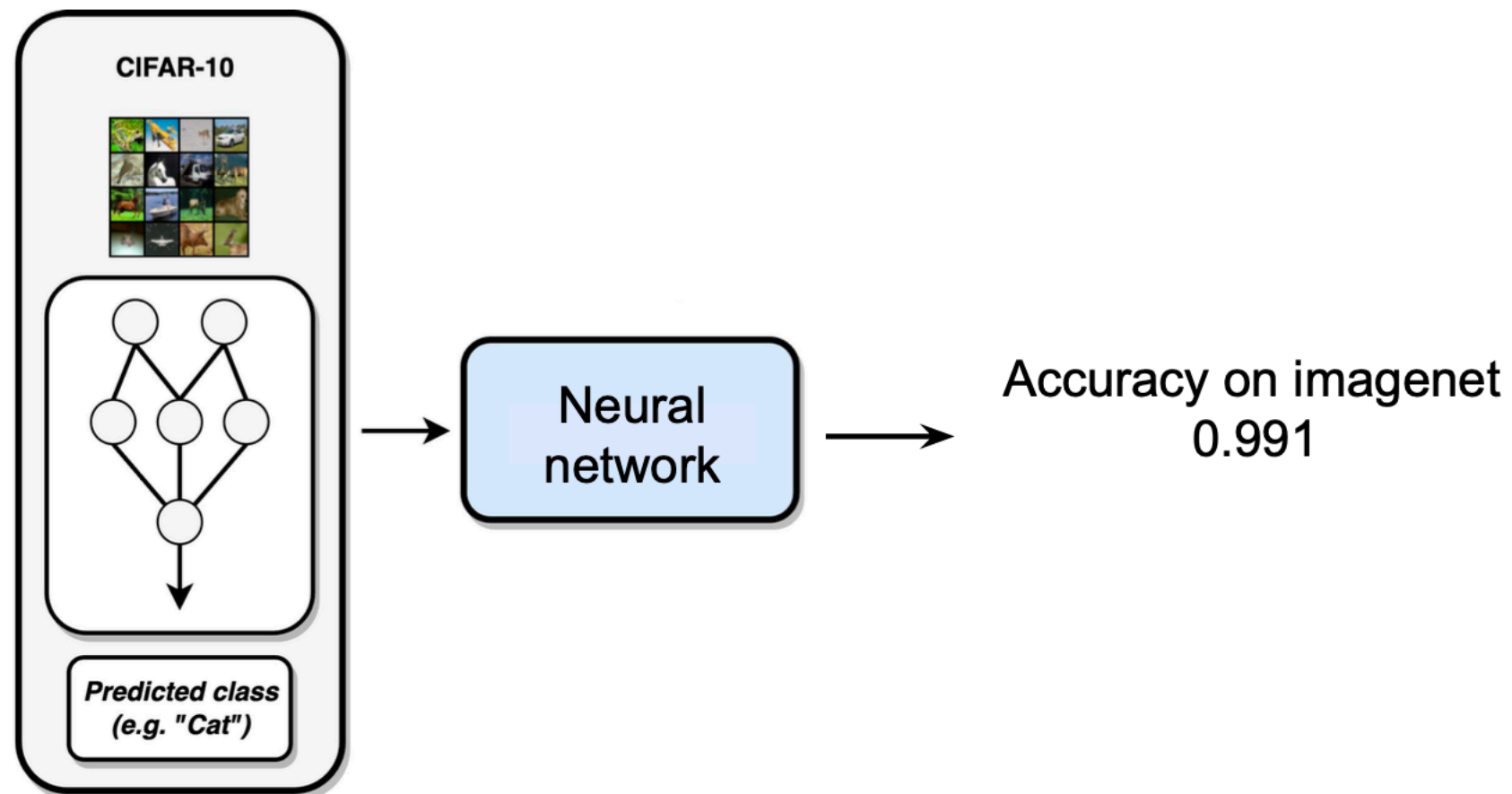
... not all are well documented



(Figure: Horwitz et al 2025)

Neural networks as data: why?

- predict accuracy without evaluating model



Can be **much faster!**

e.g. LoRA meta networks can predict CLIP score / commonsense reasoning accuracy $\geq 50,000$ times faster (Putterman et al 2025)

Neural networks as data: why?

- derive information about models, compare models, model finding

Was this model derived from... ?

Was it trained on... ?

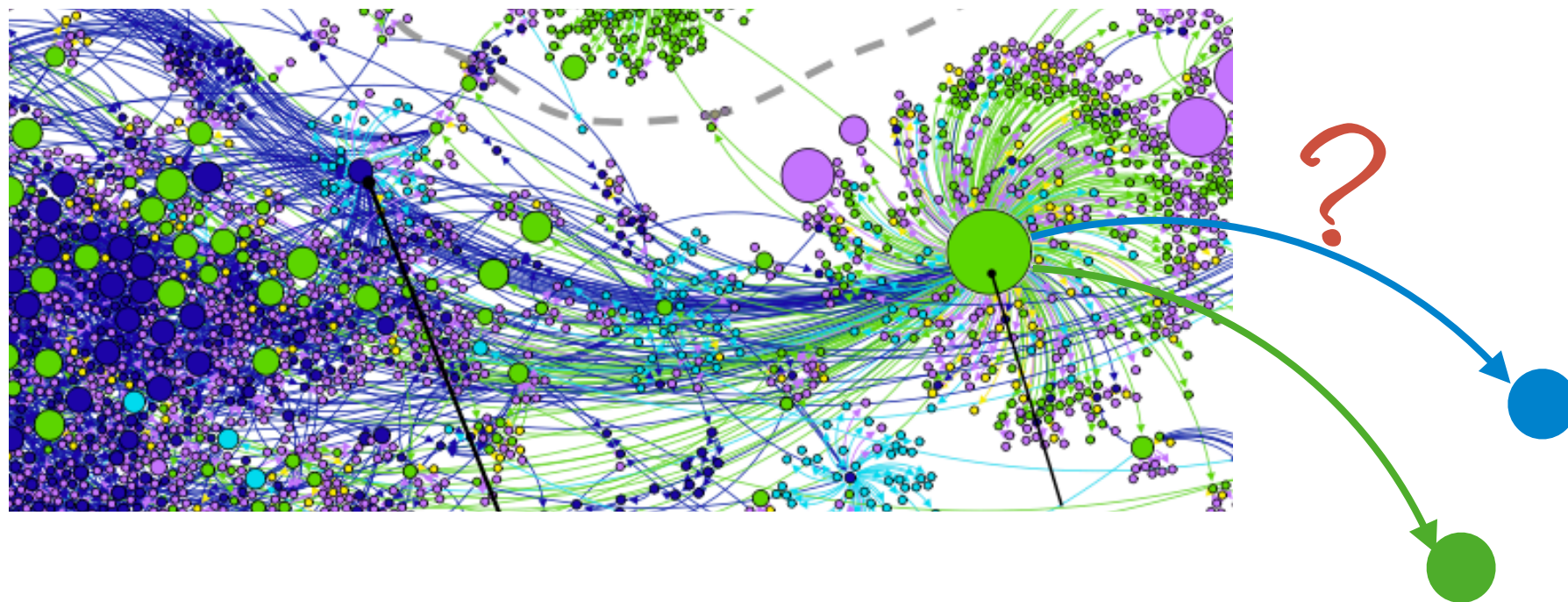
IP, licensing
safety

bias estimation

misuse (prevention)

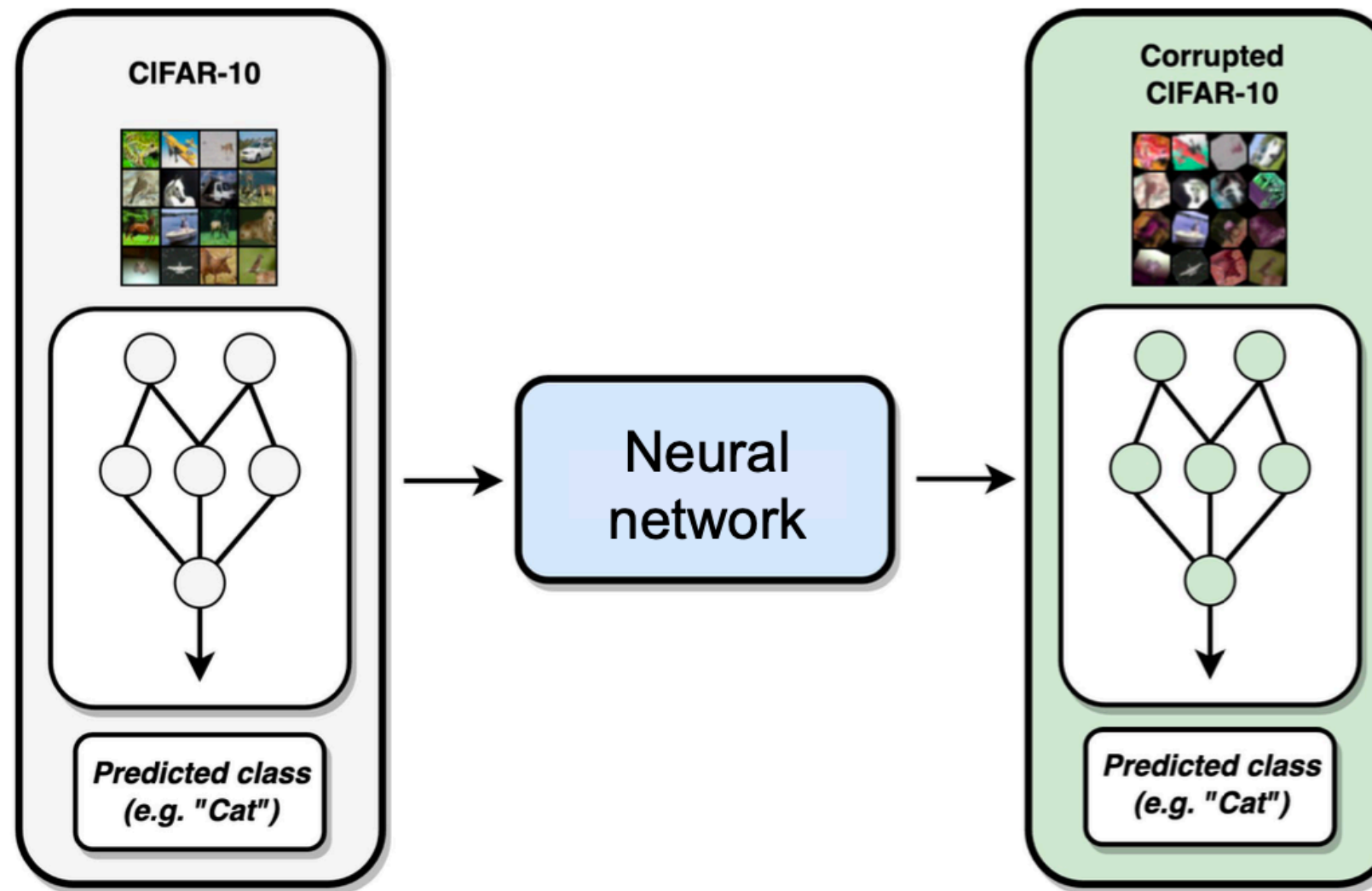
model atlas

...



Neural networks as data: why?

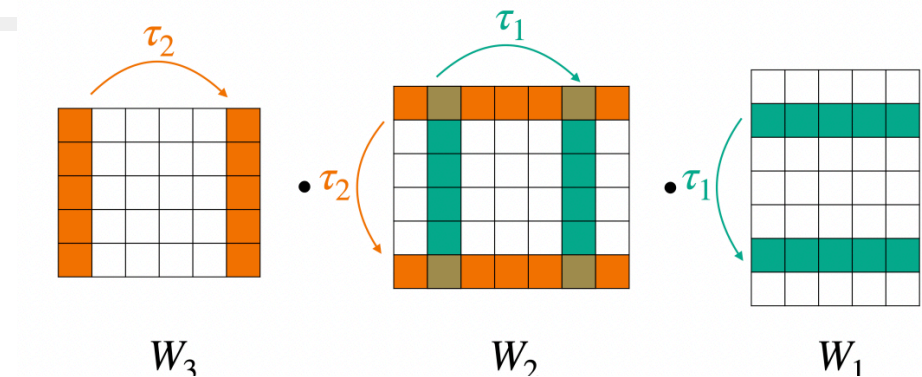
- editing networks: domain adaptation, pruning, compression, ...



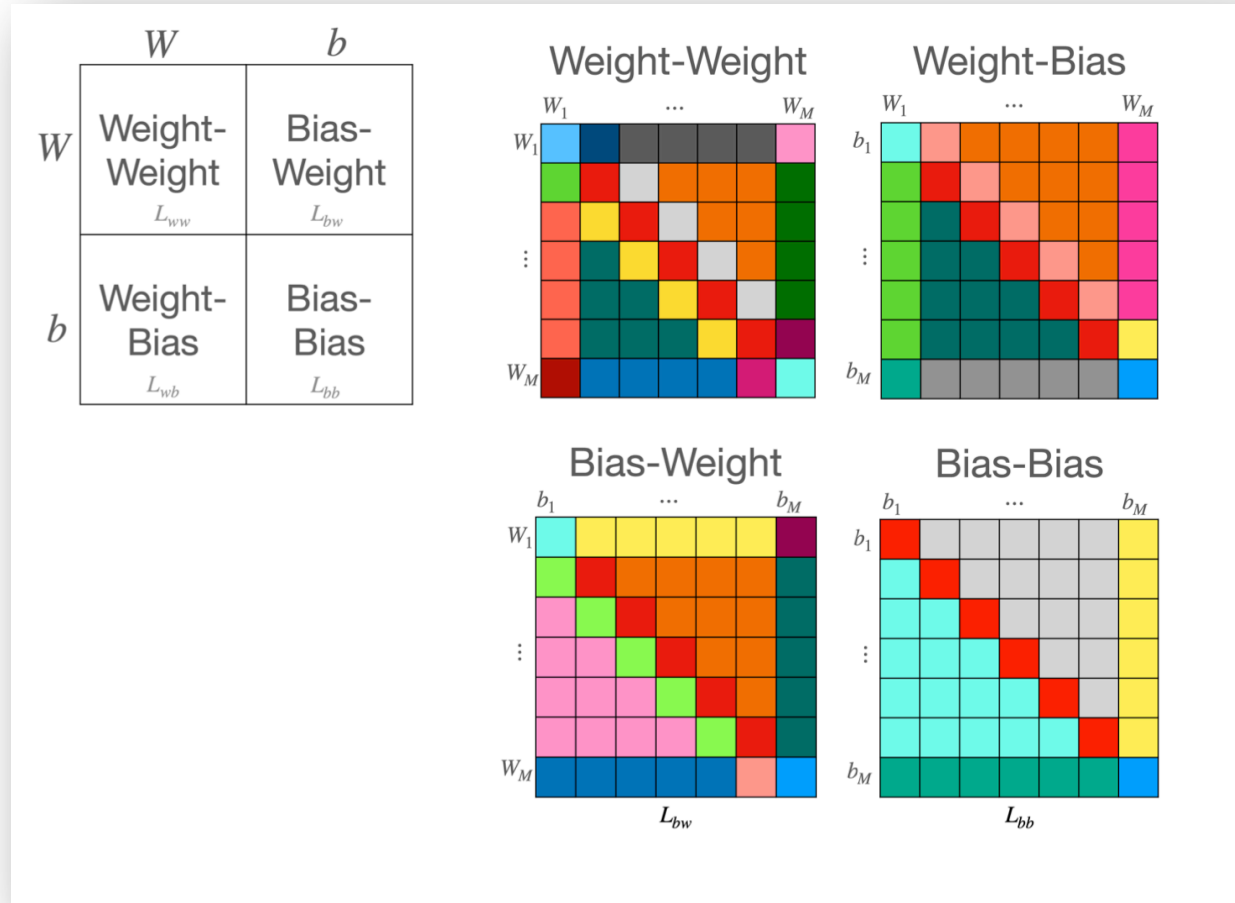
Weight space learning: example techniques

Equivariant meta-networks, e.g.:

- Permutation equivariant layers (*Zhou et al 2023, Zhou et al 2023, Navon et al 2023*)

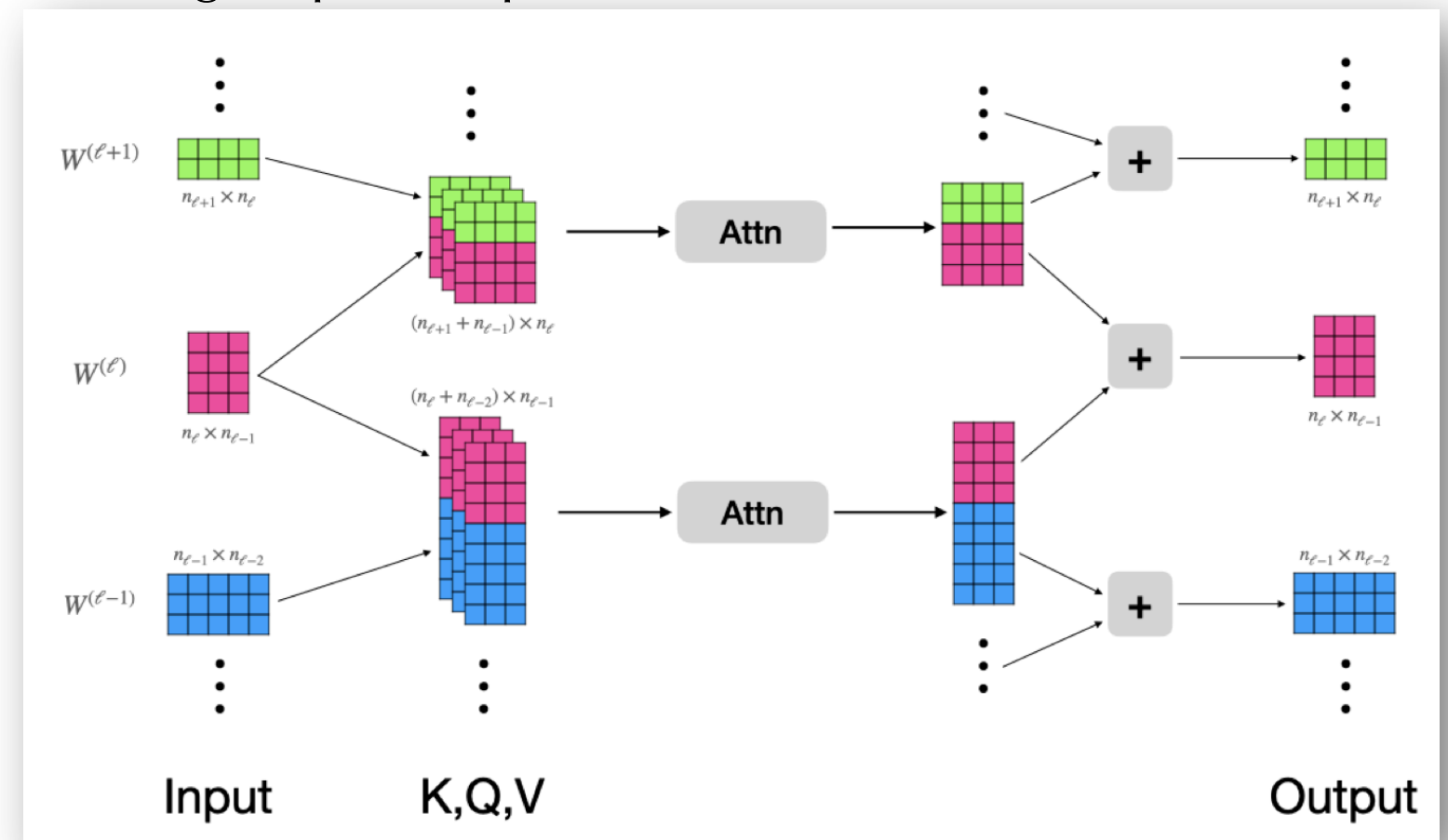


Deep weight space networks:
linear equivariant layers



(Navon et al 23)

Neural functional transformers:
weight space equivariance + attention



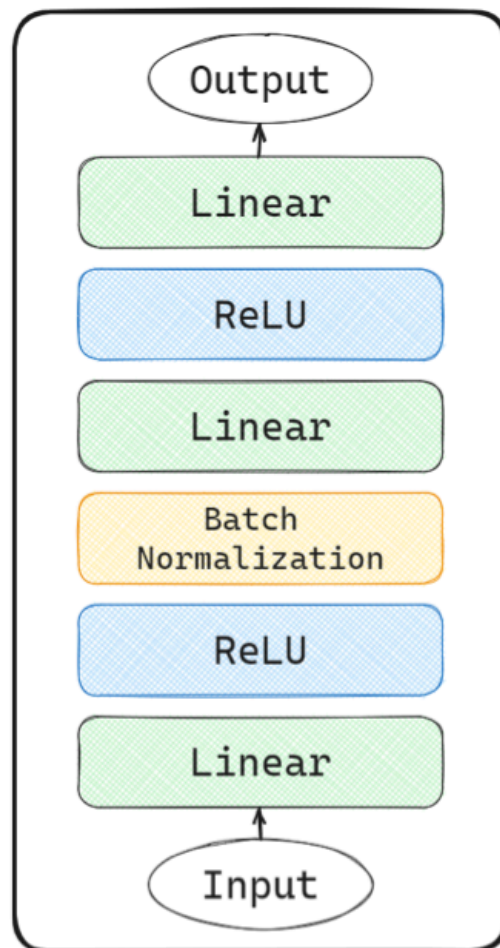
(Zhou et al 23)

Weight space learning: example techniques

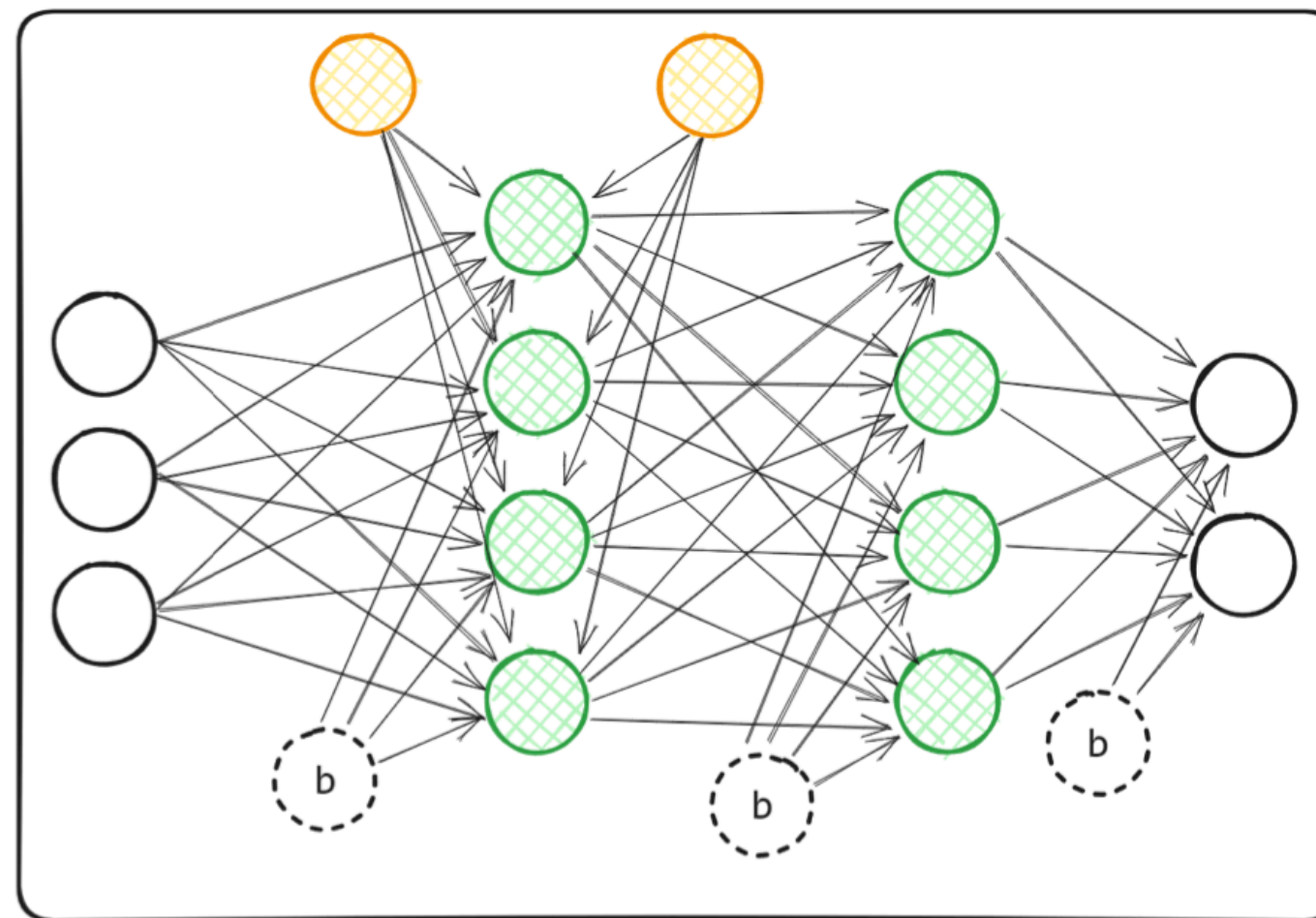
Equivariant meta-networks, e.g.:

- GNN on “parameter” graph: applicable across input models
permutation symmetries = graph automorphisms

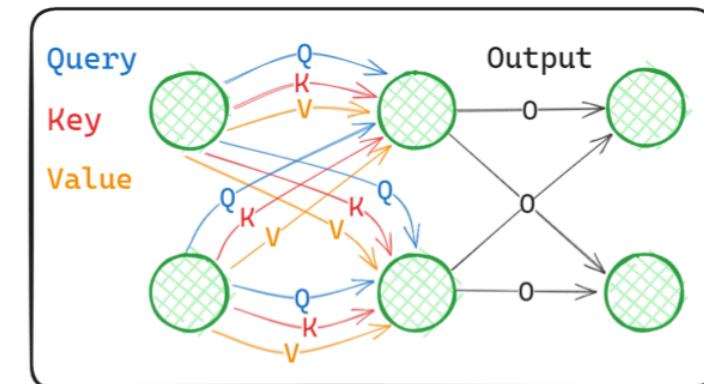
Network Definition



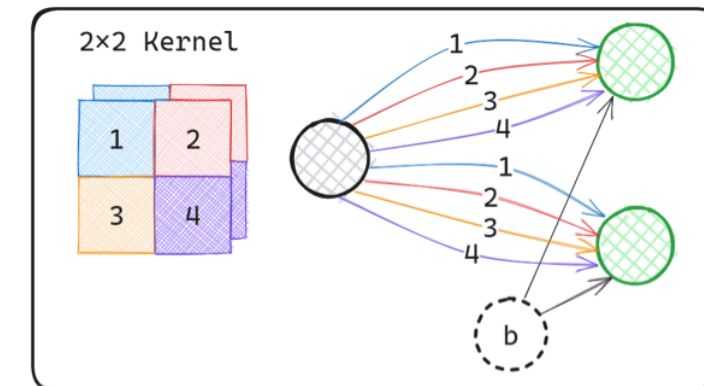
Graph Definition



Attention Layer



2D Convolution Layer



Figures: Lim et al 2023

Further questions & comments

- Does equivariance help?

Example: LoRA metanetworks (*Putterman et al 2025*)

		Qwen2-ARC-LoRA		Llama3.2-ARC-LoRA
LoL Model		Val Loss (R^2)	ARC-C Acc (R^2)	Val Loss (R^2)
Naive Models	MLP($[U, V]$)	.113 \pm .059	.107 \pm .035	.091 \pm .030
	Transformer($[U, V]$)	.856 \pm .061	.630 \pm .045	.828 \pm .120
	MLP(UV^T)	.987 \pm .003	.981 \pm .002	OOM
Efficient Invariant	MLP(O-Align($[U, V]$))	.821 \pm .078	.965 \pm .004	.562 \pm .103
	MLP($\sigma(UV^T)$)	.999 \pm .000	.983 \pm .002	.998 \pm .000
	GL-net	.998 \pm .000	.987 \pm .001	.995 \pm .000

symmetry-aware but expensive

canonicalization (rotation, reflection)

invariant features

equivariant layers

Symmetry-aware methods
generally perform well
in predicting model performance

naive models do not perform
as robustly

Further questions & comments

- Does equivariance help?
- Scalability
- When to use invariance (e.g., for which quantities)
- Equivariant models or invariant descriptors
- Exploit model atlas

Roadmap for theory

- **Recent results:**

- I: **Challenges** — groups are huge, **expanders** and small informative group subsets

Applications: computational complexity, approximate symmetry, data augmentation

- II: **Validation** of symmetry — **testing** for symmetry in data
- III: Theoretical benefits — sample complexity, **generalization** bounds

- **New directions:**

- **Any-dimensional** models
- **Geometry** beyond symmetries (topological deep learning, curvature)

Recent results I: Expanders


Groups are challenging!

- Permutation group S_d is a very **large discrete** set!

Example: $d = 50 \implies |S_d| \approx 3 \times 10^{64}$

It is way bigger than the number of atoms in the Earth!

- Big finite groups make **group averaging** become less practical:

impractical 

$$\frac{1}{|G|} \sum_{g \in G} f(g \cdot x)$$

Question: Can **small** subsets of a **large finite group** retain enough of the group's structure for our GeoML applications (data augmentation, averaging, optimization)?

Generating sets

- A subset $S \subseteq G$ of a finite group G is called a **generating** set iff

$$\forall g \in G \exists s_1, \dots, s_k \in S \text{ such that } g = s_1 s_2 \cdots s_k$$

They contain enough “**information**” to determine the entire group.

$$\text{Example: } f^\star(s \cdot x) = f^\star(x), \forall s \in S \implies f^\star(g \cdot x) = f^\star(x), \forall g \in G$$

\implies choose the generating set with **minimum** size $|S|$

Finding a minimum size S ? Unfortunately NP-hard

In some cases it is easy to find them: For all permutations S_d , use $\sigma = (i, i + 1)$; $i = 1, 2, \dots, (d - 1)$

Minimum generating set

- Beyond computational hardness, what is the minimum size of such S ?

Theorem: For any finite group G a generating set S with $|S| \leq \log_2 |G|$ exists.

- Exponential saving!
- Unfortunately, proof is not algorithmic
- Can we find generating sets with $|S| \leq C \log |G|$ in polynomial time?
- Randomness!

worse constant

Random Cayley graphs

Theorem: a randomly chosen subset S with size $\left\lceil 2.67 \times \left(\log |G| + \log \frac{1}{\delta} + 0.7 \right) \right\rceil$ is a generating set with probability $1 - \delta$

- **Random** Cayley graphs are expanders!
- Introduced in 90's for **sampling** from groups, etc.

We can efficiently find a generating set within a **constant-factor** of the minimum size


Proof: Fourier analysis over groups, representation theory (irreducible representations)

What is the application of this in geometric machine learning?

Recent results I: Applications of expanders

Computational complexity of learning with invariances

- Can we learn an **exactly** invariant function in Sobolev regression?
- One can show that it is equivalent to solving the following optimization:

convex 

$$\arg \min_{\theta} \{L(\theta) + \eta \|\theta\|_2^2\}$$
$$\text{s.t. } \forall g \in G : \rho(g) \cdot \theta = \theta$$

- $\forall g, h \in G : \rho(gh) = \rho(g)\rho(h)$ (unitary) group **representation**
- One can replace it with the following “for free”

$$\arg \min_{\theta} \{L(\theta) + \eta \|\theta\|_2^2\}$$
$$\text{s.t. } \forall g \in S : \rho(g) \cdot \theta = \theta$$

Theorem: One can find an exactly invariant estimator in time $O_{n,d} ((\log |G|)^3)$

Data augmentation

- Original data $\{x_i : i \in [n]\}$ replaced with $\{gx_i : i \in [n], g \in G\}$
- **Full group** data augmentation is **prohibitive** for large finite groups

Question: Can we choose a **small** subset of group $S \subseteq G$ for data augmentation and still achieve **full** group (data augmentation) **statistical benefits**?

- Consider a classical situation (density estimation in Sobolev spaces, non-parametric regression via kernels)

Theorem: only $O(\log |G|)$ random elements suffice to achieve full statistical gain!

Example: permutation group $|S_d| \approx \exp(d \log(d))$

But only $|S| = O(d \log d)$ random elements suffice!

Exponential improvement!

Approximate group averaging I

Replace $\frac{1}{|G|} \sum_{g \in G} f(g \cdot x)$ (hard) with $\frac{1}{|S|} \sum_{g \in S} f(g \cdot x)$ (easy)

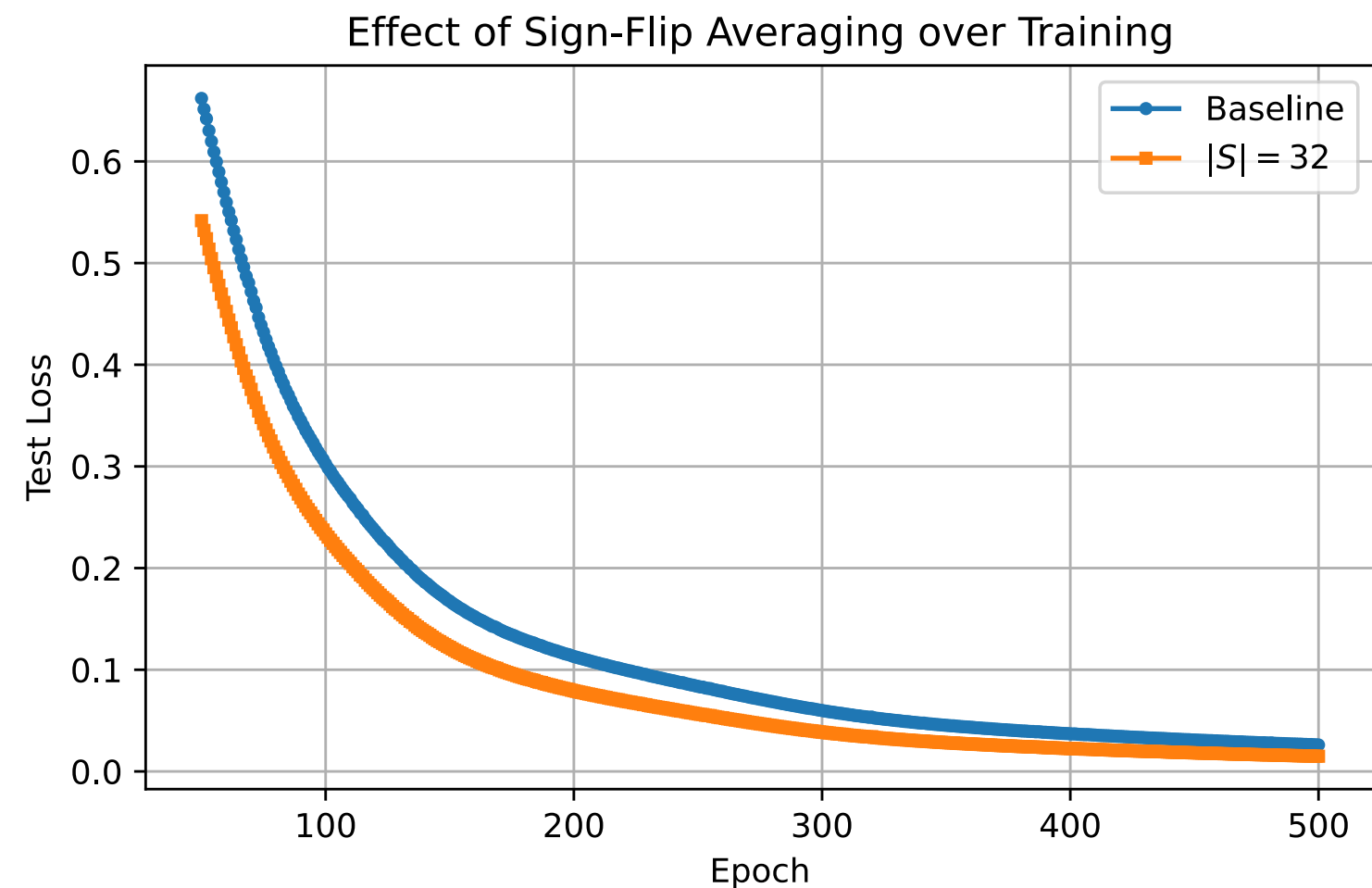
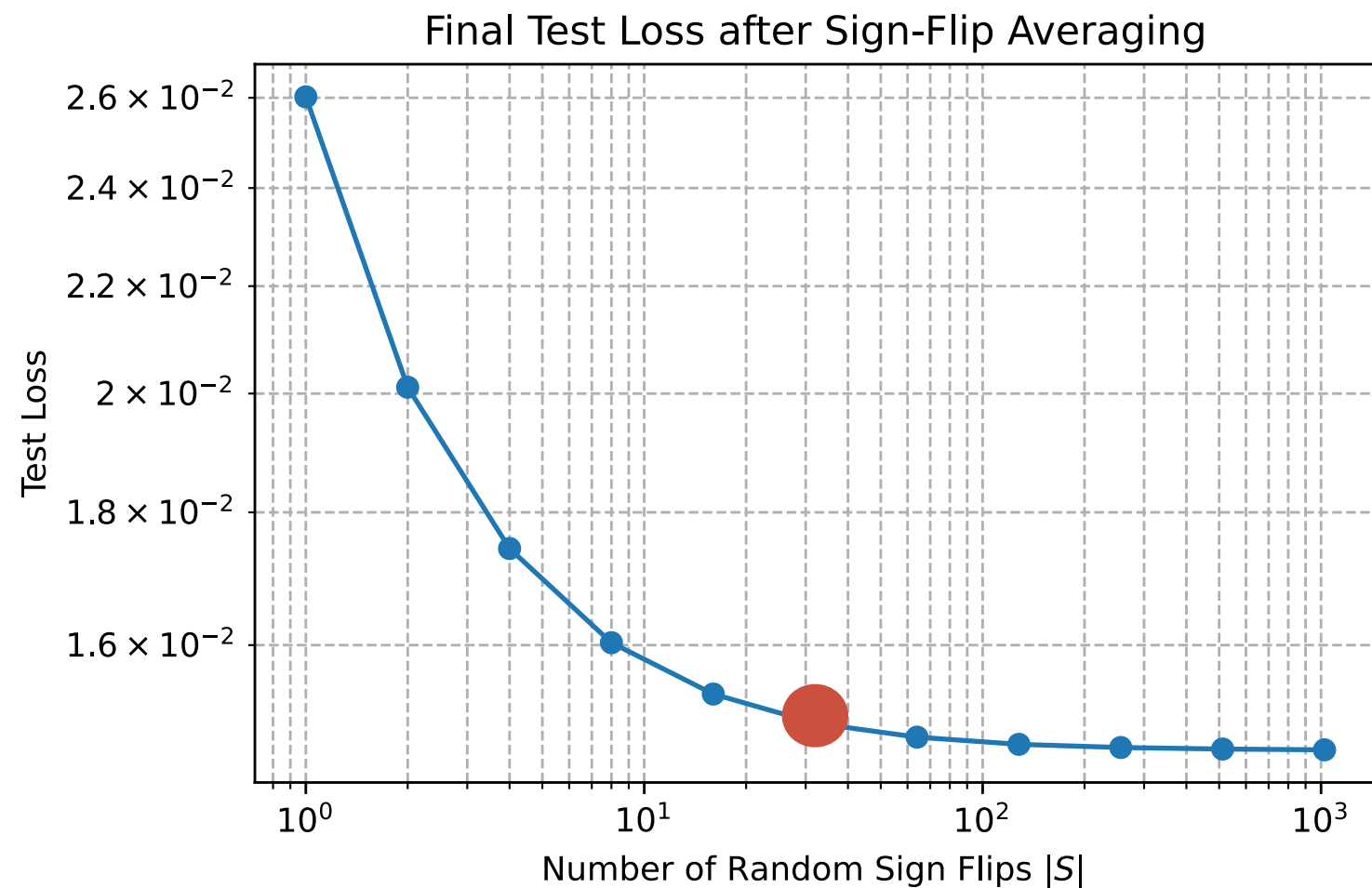
- Question: Can we **uniformly** upper bound the approximation error? How large $|S|$? Random set?

 f can change during training

Theorem: with high probability, $\sup_{f: \|f\|_2 \leq 1} \left\| \frac{1}{|G|} \sum_{g \in G} f(g \cdot x) - \frac{1}{|S|} \sum_{g \in S} f(g \cdot x) \right\|_2^2 \leq O\left(\frac{\log |G|}{|S|}\right)$

- Classical large deviation theory does not work in this regime!

Approximate group averaging II



Toy experiment: predicting absolute value (sign-invariance) with three layer ReLU networks.

- $|G| = 2^{20} \approx 10^6$, $|S| = 32$ is enough, and **uniformly** helpful over all epochs (with just one draw)

Approximate group averaging III

Replace $\frac{1}{|G|} \sum_{g \in G} f(g \cdot x)$ (hard) with $\frac{1}{|S|} \sum_{g \in S} f(g \cdot x)$ (easy)

What if we want exactness $\frac{1}{|G|} \sum_{g \in G} f(g \cdot x) = \frac{1}{|S|} \sum_{g \in S} \alpha(g) f(g \cdot x)$ for all $f \in F$?

- Constant functions $f \equiv C \implies$ always satisfy this! weight desired functions

Theorem: even moderate degree polynomial function space require full group averaging!

Corollary: Enforcing **exact** symmetry \implies hard (via averaging)

Enforcing **approximate** symmetry \implies easy (via averaging)

Achieving full **statistical gain** \implies easy (via averaging)

Exponential separation

Expanders in geometric machine learning

- Interesting mathematical tools, relevant to ML applications
- Fourier analysis on groups, expanders, representation theory of neural networks
- Optimization under symmetry

References:

- A. Soleymani*, [B. Tahmasebi](#)*, S. Jegelka, P. Jaillet “Learning with Exact Invariances in Polynomial Time,” ICML 2025 (spotlight).
- [B. Tahmasebi](#), M. Weber, S. Jegelka “Data Augmentation: A Fourier Analysis Perspective,” preprint, 2025
- [B. Tahmasebi](#), M. Weber “Achieving Approximate Symmetry is Exponentially Easier than Exact Symmetry,” preprint, 2025
- N. Alon, Y. Roichman “Random Cayley graphs and expanders,” Random Structures & Algorithms, 1994

Recent results II: Symmetry validation (testing)

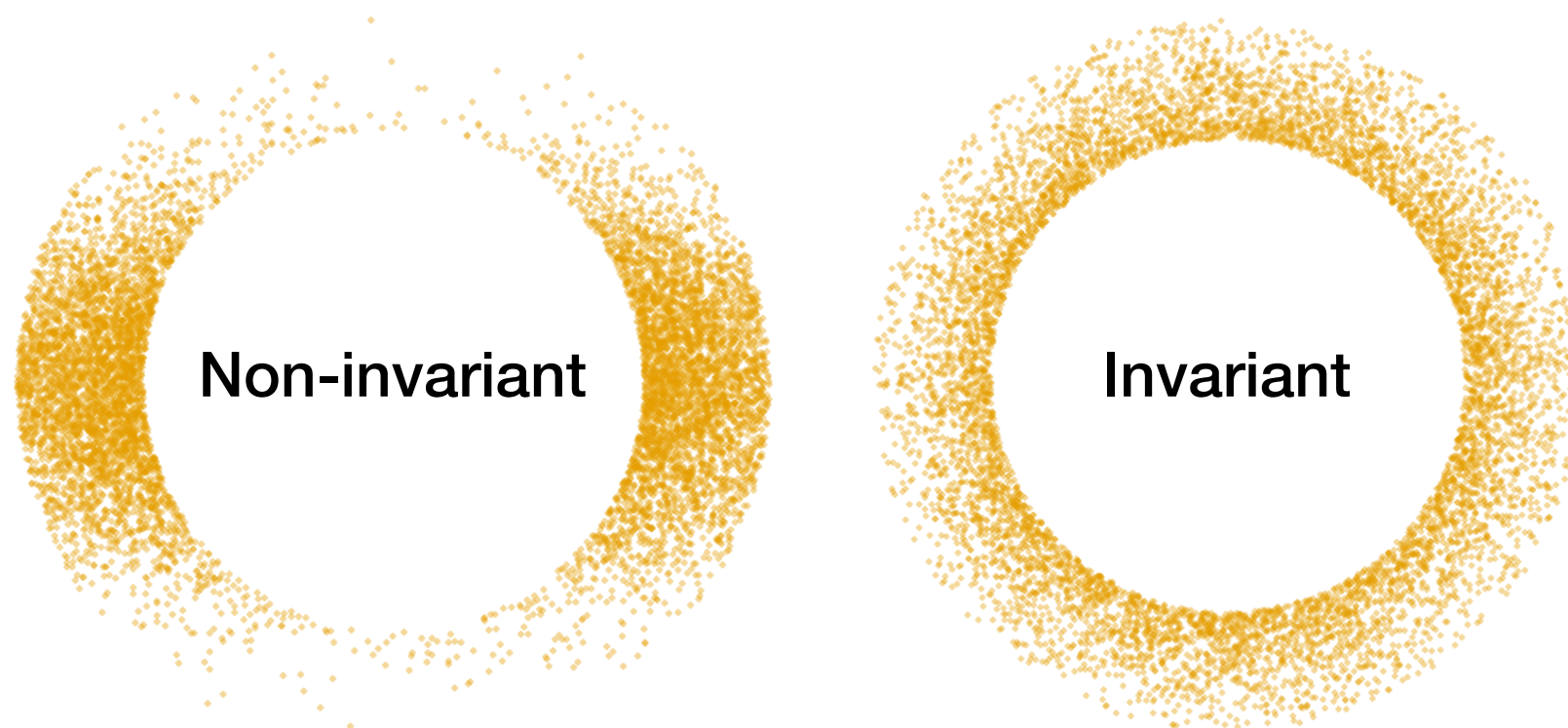
Testing distributional symmetry in data

Given i.i.d. data $x_1, x_2, \dots, x_n \sim \mu$ decide whether:

Data distribution μ is **invariant** or

Data distribution μ is **NOT invariant**

Example: 2D rotational invariance



Applications:

- Classic: e.g., testing exchangeability

$$(Y, Z) \sim (Z, Y)$$

- GeoML **assumes** invariance to transformations (rotations, scaling, translations, permutations)
- True invariance must be **verified** in real-world data (Are MRI images really rotationally symmetric?)
- Subtle asymmetries can lead to **biased** models and incorrect conclusions, significantly impacting learning and generalization
- Detecting **violations** to improve model robustness and interpretability

Statistical formulation I

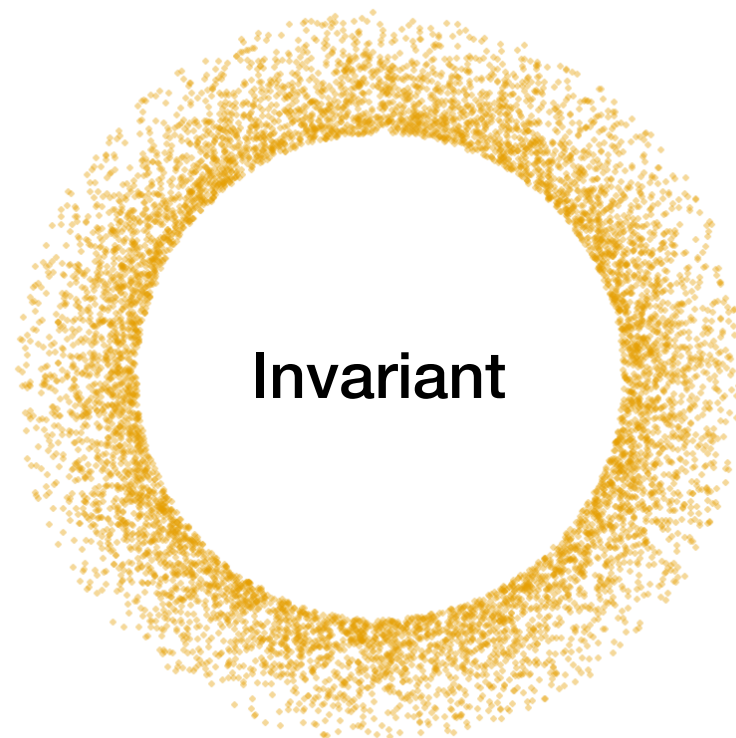
Data distribution μ is invariant:

$$H_0 : \mu \equiv g . \mu \text{ for all } g \in G$$

Null hypothesis

Example: $G =$ 2D rotation matrices: $f(g . x) \equiv f(x) \implies f(x) = \phi(\|x\|_2)$

$f :=$ prob. dens. func. of μ



$$\begin{pmatrix} x \\ y \end{pmatrix} \sim \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, \quad \forall \theta \in [0, 2\pi]$$

same distribution

Statistical formulation II

Data distribution μ is NOT invariant:

$$H_1 : \sup_{g \in G} D(g \cdot \mu, \mu) \geq \epsilon$$

Alternative hypothesis

There exists a $g \in G$ **violating** invariance

Probability divergence

Parameter



$\theta = \frac{\pi}{2}$ violates invariance

$D(\mu, \nu)$ is a probability **pseudometric** (triangle inequality + non-negative)

Example: optimal transport, kernel max-mean discrepancy (MMD), total variation, ℓ_p -distance, Hellinger, neural network features, cross-entropy)

Assumption: $D(g \cdot \mu, g \cdot \nu) = D(\mu, \nu)$ for all μ, ν (satisfied all examples)

Decide either μ is invariant, or

μ is NOT invariant and find $g \in G$ violating symmetry

Statistical formulation II

Data distribution μ is NOT invariant:

$$H_1 : \sup_{g \in G} D(g \cdot \mu, \mu) \geq \epsilon$$

Alternative hypothesis

There exists a $g \in G$ **violating** invariance

Probability divergence

Parameter



$\theta = \frac{\pi}{2}$ violates invariance

$D(\mu, \nu)$ is a probability **pseudometric** (triangle inequality + non-negative)

Example: optimal transport, kernel max-mean discrepancy (MMD), total variation, ℓ_p -distance, Hellinger, neural network features, cross-entropy)

Assumption: $D(g \cdot \mu, g \cdot \nu) = D(\mu, \nu)$ for all μ, ν (satisfied all examples)

Decide either μ is invariant, or

μ is NOT invariant and find $g \in G$ violating symmetry

Results (hardness, randomized methods)

- Even with **infinite** data, finding $g \in G$ can be computationally challenging!

Theorem: There exists a setting such that $\arg \sup_{g \in G} D(g \cdot \mu, \mu)$ is NP-hard

Proof: A variant of Traveling Salesman Problem (TSP) problem reduces to this

- But **randomness** can help

Theorem: It is always the case that

$$\mathbb{E}_{g \sim G} [D(g \cdot \mu, \mu)] \leq \sup_{g \in G} D(g \cdot \mu, \mu) \leq 2 \mathbb{E}_{g \sim G} [D(g \cdot \mu, \mu)]$$

approximate from data

Proof: Second-order probability, group theory

sample from group

Corollary: random $g \in G$ **certificates** symmetry/asymmetry

Sample from group, then test!

Deterministic algorithms

- How to avoid **sampling** $g \in G$ from the group?

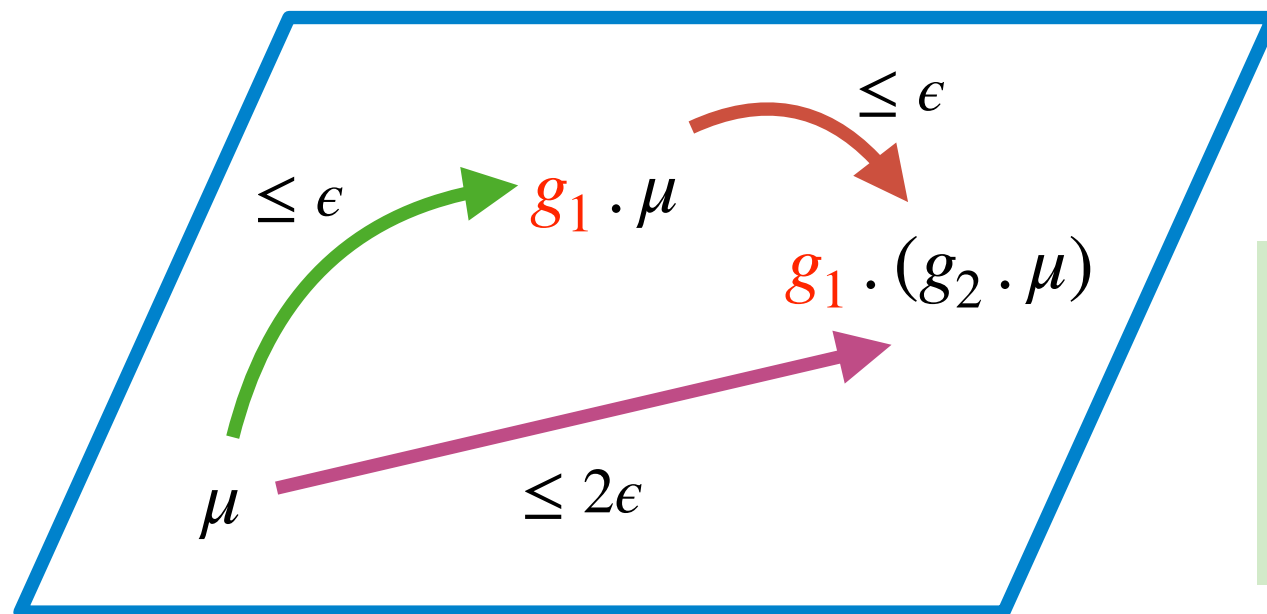
Classical idea: covering the group

$$\sup_{g \in G} D(g \cdot \mu, \mu) \approx \max_{g \in S} D(g \cdot \mu, \mu), \quad \text{if } S \approx G$$

But group is large, non-standard geometry, covering is not-constructive

New idea: use the fact that G is a group and a lot of redundancies!

$$D(g_1 \cdot (g_2 \cdot \mu), \mu) \leq D(g_1 \cdot \mu) + D(g_2 \cdot \mu, \mu)$$



Not captured in covering arguments!

- For rotation group $SO(d)$ one can achieve **reduced** coverings of size $\mathcal{O}\left(\frac{d^2}{\epsilon}\right)$ instead of $\mathcal{O}\left(\epsilon^{-d^2}\right)$

Deterministic algorithms

- How to avoid **sampling** $g \in G$ from the group?

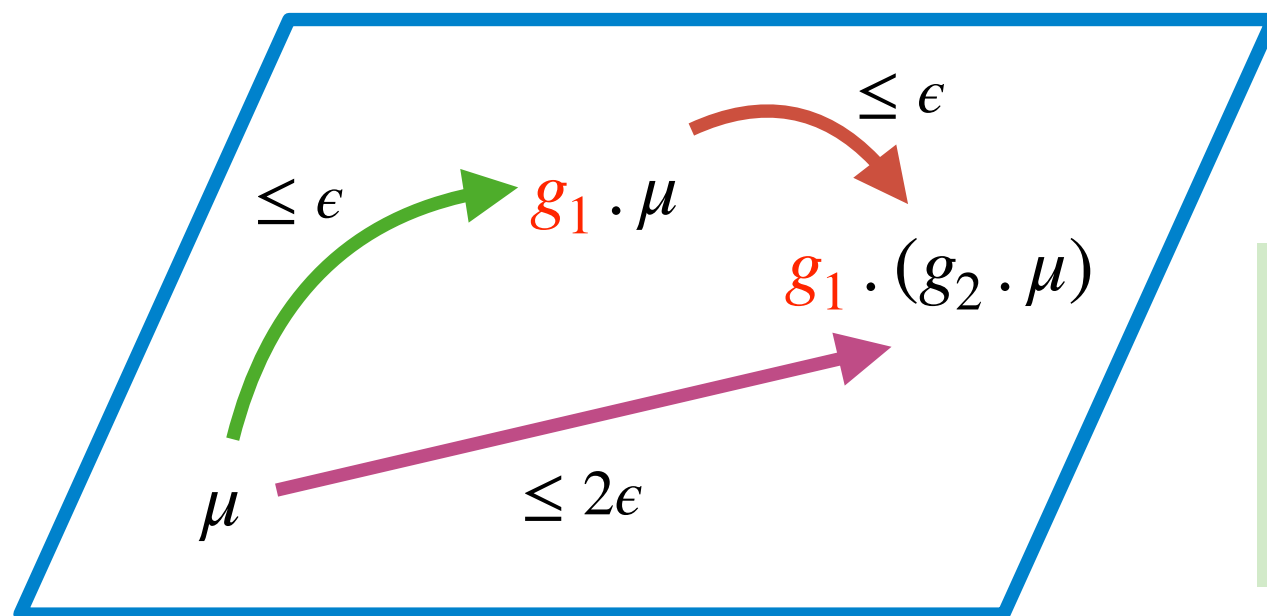
Classical idea: covering the group

$$\sup_{g \in G} D(g \cdot \mu, \mu) \approx \max_{g \in S} D(g \cdot \mu, \mu), \quad \text{if } S \approx G$$

But group is large, non-standard geometry, covering is not-constructive

New idea: use the fact that G is a group and a lot of redundancies!

$$D(g_1 \cdot (g_2 \cdot \mu), \mu) \leq D(g_1 \cdot \mu, \mu) + D(g_2 \cdot \mu, \mu)$$



Not captured in covering arguments!

exponential improvement!

- For rotation group $SO(d)$ one can achieve **reduced** coverings of size $\mathcal{O}\left(\frac{d^2}{\epsilon}\right)$ instead of $\mathcal{O}\left(\epsilon^{-d^2}\right)$

Open directions

- Statistics under symmetries need theory **beyond** classical statistics
 - Can we obtain **optimal (reduced) coverings** for groups?
 - Coverings for **optimization** over groups?
- Can we **integrate** testing and symmetry-aware architecture (applied question)?

Reference:

A. Soleymani*, **B. Tahmasebi***, S. Jegelka, P. Jaillet “A Robust Kernel Statistical Test of Invariance: Detecting Subtle Asymmetries,” oral presentation at AISTATS 2025 (longer version to *The Annals of Statistics*).

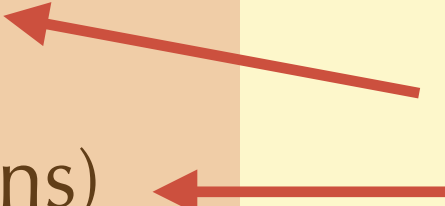
Recent results III:

Sample complexity benefit

Provable gains in sample complexity?

- In **practice**, learning with invariances works better
- In **theory**, do we get **gains** in sample complexity?

Given various **domains and group actions** in practice, a **relevant** theory should apply to holds under minimal assumptions:

- Data domain unit sphere $\{x \in \mathbb{R}^d : \|x\|_2 = 1\}$
 - Specific group (particular permutations, 2D rotations)
- 
- too restrictive!

A relevant theory should work with:
generic **manifold** data domains,
any **continuous group actions**

Example: **sets of 3D points**, under **translation**
and **change of coordinates**

\implies Non-spherical data, non-standard group

General result

Assume:

- x_1, x_2, \dots, x_n i.i.d. data, uniformly from a compact manifold M
- $y_i = f^\star(x_i) + \epsilon_i$ with i.i.d. $\epsilon_i \sim N(0, \sigma^2)$
- $f^\star(g \cdot x) = f^\star(x), \quad \forall g \in G$
- G acts continuously on manifold M
- Sobolev regression $f^\star \in H^s(M)$

Square-integrable derivatives up to order $s \in \mathbb{N}$

Estimator (kernel ridge regression (KRR)):

$$\hat{f} := \arg \min_{f \in F} \frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2 + \eta \|\hat{f}\|_{H^s(M)}^2$$

invariant + Sobolev

General result

Assume:

- x_1, x_2, \dots, x_n i.i.d. data, uniformly from a compact manifold M
- $y_i = f^\star(x_i) + \epsilon_i$ with i.i.d. $\epsilon_i \sim N(0, \sigma^2)$
- $f^\star(g \cdot x) = f^\star(x), \quad \forall g \in G$
- G acts continuously on manifold M
- Sobolev regression $f^\star \in H^s(M)$

Estimator (kernel ridge regression (KRR)):

$$\hat{f} := \arg \min_{f \in F} \frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2 + \eta \|\hat{f}\|_{H^s(M)}^2$$

Theorem:

$$\mathbb{E} \left[\|\hat{f} - f^\star\|_2^2 \right] \leq \left(\frac{C_d \text{vol}(M/G)}{n} \right)^{s/(s+d)}$$

- Multiplicative gain:

- $\text{vol}(M/G)$ is the volume of **quotient**

- Exponential gain: $d := \dim(M) - \dim(G)$

General result

Assume:

- x_1, x_2, \dots, x_n i.i.d. data, uniformly from a compact manifold M
- $y_i = f^\star(x_i) + \epsilon_i$ with i.i.d. $\epsilon_i \sim N(0, \sigma^2)$
- $f^\star(g \cdot x) = f^\star(x), \quad \forall g \in G$
- G acts continuously on manifold M
- Sobolev regression $f^\star \in H^s(M)$

Estimator (kernel ridge regression (KRR)):

$$\hat{f} := \arg \min_{f \in F} \frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2 + \eta \|\hat{f}\|_{H^s(M)}^2$$

Theorem:

$$\mathbb{E} \left[\|\hat{f} - f^\star\|_2^2 \right] \leq \left(\frac{C_d \text{vol}(M/G)}{n} \right)^{s/(s+d)}$$

• Multiplicative gain:

• $\text{vol}(M/G)$ is the volume of **quotient**

• Exponential gain: $d := \dim(M) - \dim(G)$

Theorem: rate is mini-max optimal!

\Rightarrow The **exact** sample complexity gain

Proof idea (spectral methods on manifolds)

Example: $M \equiv [0, 2\pi]$ unit circle

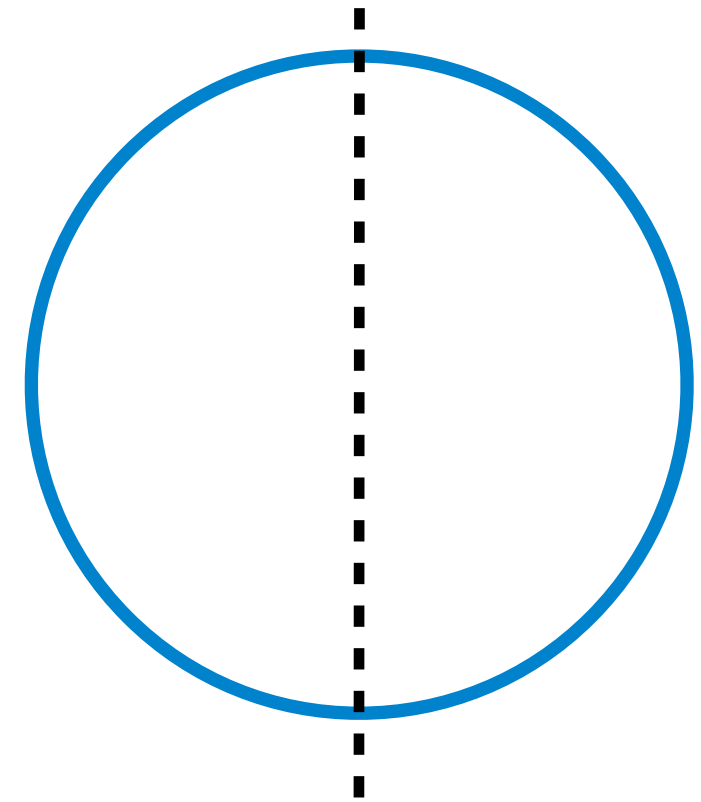
Fourier series: $\sin(k\theta), \cos(k\theta), k = 0, 1, \dots$

Invariant functions: $f^\star\left(\frac{\pi}{2} - \theta\right) = f^\star(\theta)$

$\implies \sin((2k+1)\theta), \cos(2k\theta), k = 0, 1, \dots$

$\frac{1}{2}$ of Fourier modes survive, $\text{vol}(M/G) = \frac{1}{2}\text{vol}(M)$

• Fourier sparsity \implies sample complexity gain



$G =$ reflection about y-axis

Proof idea (spectral methods on manifolds)

Theorem: (extended Weyl's law) for any compact connected manifold, any continuous Lie group of isometries, for the number of eigenvalues of Laplace-Beltrami operator:

$$N(\lambda; G) \approx \frac{\omega_d}{(2\pi)^d} \text{vol}(M/G) \lambda^{d/2}$$



volume of unit ball in \mathbb{R}^d

Proof idea (spectral methods on manifolds)

Theorem: (extended Weyl's law) for any compact connected manifold, any continuous Lie group of isometries, for the number of eigenvalues of Laplace-Beltrami operator:

$$N(\lambda; G) \approx \frac{\omega_d}{(2\pi)^d} \text{vol}(M/G) \lambda^{d/2}$$

Proof challenge: Weyl's law proved via **PDEs** and differential geometric

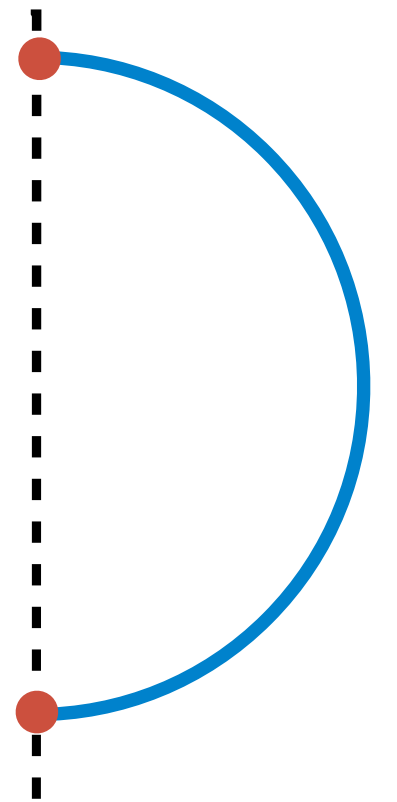
Boundary emerges, boundary condition?

Neumann condition emerges!

$$f^\star(\theta) = \sin((2k+1)\theta), \cos(2k\theta), k = 0, 1, \dots$$

$$\frac{df}{d\theta} = 0 \quad \text{on boundary}$$

$\text{vol}(M/G)$ needs to be defined! Quotient is not a **manifold**!



Beyond learning

- Extensions to probability divergences (e.g., **optimal transport**)
 - Sometimes (kernel maximum-mean discrepancy (MMD)) the gain depends on **other geometric properties** (zeta function of manifold)
- Extensions of equivariances and geometric stability, MDP, RL, diffusion, etc.

References:

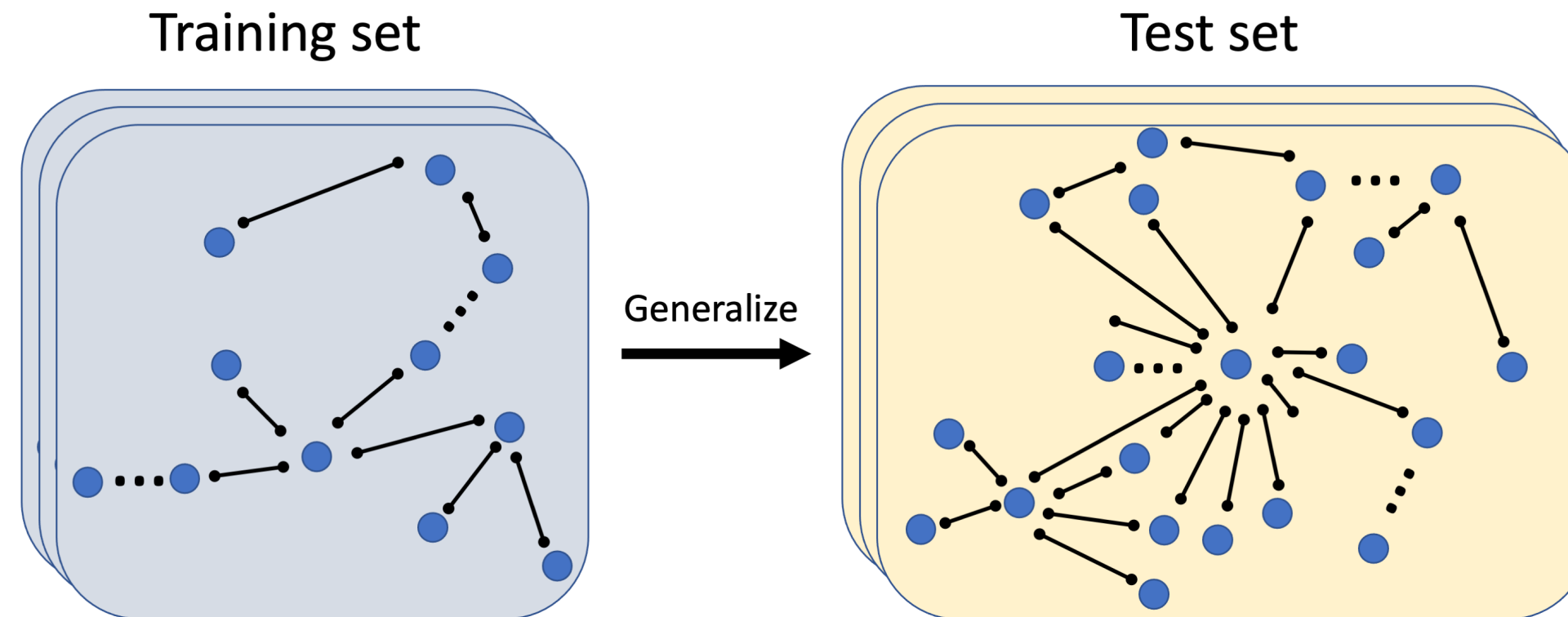
- **B. Tahmasebi**, S. Jegelka “The Exact Sample Complexity Gain from Invariances for Kernel Regression,” NeurIPS 2023 (spotlight)
- **B. Tahmasebi**, S. Jegelka “Sample Complexity Bounds for Estimating Probability Divergences under Invariances,” ICML 2024
- M. Petrache, S. Trivedi “Approximation-Generalization Trade-offs under (Approximate) Group Equivariance,” NeurIPS 2023
- B. Elesedy “Provably Strict Generalisation Benefit for Invariance in Kernel Methods,” NeurIPS 2021
- A. Bietti, L. Venturi J. Bruna “On the Sample Complexity of Learning under Invariance and Geometric Stability,” NeurIPS 2021
- S. Mei , T. Misiakiewicz, A. Montanari “Learning with invariances in random features and kernel models ,” COLT 2021
- B. Elesedy, S. Zaidi, “Provably Strict Generalisation Benefit for Equivariant Models,” ICML 2021
- Z. Chen, M. Katsoulakis, L. Rey-Bellet, W. Zhu, “Sample Complexity of Probability Divergences under Group Symmetry,” ICML 2023

New directions: Any-dimensional models

Any-dimensional machine learning

- Machine learning models can be defined on a **fixed set of parameters** and evaluated on inputs of **varying sizes**.

Example: graph neural networks, deep sets, machine learning models on point clouds



Train on **small** data (graph, set, point cloud, sequence) and generalize on **large** data (size generalization)

Representation stability

- How fixed set of parameters are enough?

Example: Linear functions of $x = (x_1, x_2, \dots, x_d)^\top \in \mathbb{R}^d$ are

$$f(x) = w_1x_1 + w_2x_2 + \dots + w_dx_d \implies d \text{ parameters}$$

What if we consider **permutation-invariant** linear functions?

$$\begin{aligned} f(x_{\sigma_1}, x_{\sigma_2}, \dots, x_{\sigma_d}) &= f(x_1, x_2, \dots, x_d), \quad \forall \sigma \in S_d \text{ group of permutations} \\ \implies f(x_1, x_2, \dots, x_d) &= w x_1 + w x_2 + \dots + w x_d \end{aligned}$$

- Only **one parameter** is enough!
- Learn one parameter from **low-dim. train data**, then generalize to high-dim. test data
- Representation stability

Example: Beyond linear? Quadratic perm. Inv. polynomials of the form $w_1 \sum_i x_i^2 + w_2 \sum_{i \neq j} x_i x_j \implies$ two dim!

General dimension k polynomials? dimension = $p(k) \approx \exp(C\sqrt{k})$

Recent advances

- Early works on “**graph invariant networks**” use this idea!
- More systematic treatment recently proposed (Levin & Diaz, 2024)
- Learning theory (invariant finite dimensional kernels) uses these ideas
 - Avoid **group averaging**, learn features that generalize over size
- Any-dimensional models for learning **PDEs** (e.g., for physics)
- Any-dimensional **optimization** (convex cone of PSD matrices, optimization over graphs)
 - Graph properties (max-cut, clique, etc) well-defined over size
- Any-dimensional **information theory**
 - Symmetry simplifies optimization for finding capacity of feedback channels

Questions:

Q1: How to identify any-dim behavior?

Q2: How to parametrize model?

Q3: How to evaluate model?

Open questions

- Identify **application** and build any-dim **models** (physics and PDEs, biology, material science, language and sequence models, etc.).
- Do we lose expressivity? **Universality** is problem specific!
- How to **evaluate** models on objects of different sizes and dimensions? What assumptions?

References:

- B. Farb, "Representation stability," ICM 2014
- T. Church, B. Farb, "Representation theory and homological stability," Adv. Math. 2013
- H. Maron, H. Ben-Hamu, N. Shamir, Y. Lipman, "Invariant and Equivariant Graph Networks," ICLR 2019
- E. Levin, M. Diaz, "Any-dimensional equivariant neural networks," AISTATS 2024
- E. Levin, V. Chandrasekaran, "Dimension-Free Descriptions of Convex Sets," arXiv, 2025
- E. Levin, Y. Ma, M. Diaz, S. Villar, "On Transferring Transferability: Towards a Theory for Size Generalization," arXiv 2025
- M. Diaz, D. Drusvyatskiy, J. Kendrick, R. Thomas, "Invariant Kernels: Rank Stabilization and Generalization Across Dimensions," arXiv 2025
- E. Atay, E. Levin, V. Chandrasekaran, V. Kostina, "Poset-Markov Channels: Capacity via Group Symmetry," arXiv 2025
- T. Phan, G. Kevrekidis, S. Villar, Y. Kehrekidis, J. Bello-Rivas, "Towards Coordinate- and Dimension-Agnostic Machine Learning for Partial Differential Equations," arXiv 2025

New direction: Geometry beyond symmetries

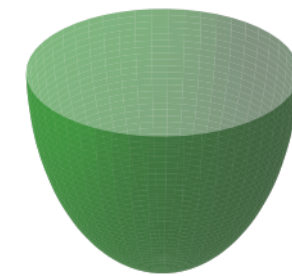
Learning theory and curvature

- Learning in high-dimensional \implies sample complexity blow-up (curse of dim)
- **Manifold hypothesis**: data lies on low-dim manifold \implies breaking curse of dim

- Under **statistical query (SQ)** lower bounds, (ReLU) neural networks are difficult to (PAC) learn

- Under manifold hypothesis:
 - Easy: if we can construct manifold from **samples**
 - **Positive** lower bounds on curvature suffice
 - Hard: even if curvature and intrinsic dim bounded
 - Real-world data?

Going beyond “intrinsic” dim for understanding neural networks performance under manifold hypothesis is essential



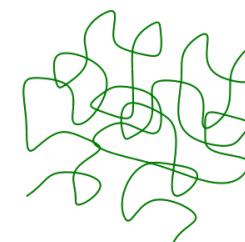
(Learnable)
Efficiently Sampleable Regime

Example: Positive Bounded Ricci Curvature Manifold



(Potentially Learnable)
Heterogeneous Regime

Example: Manifolds with varying intrinsic dimension (e.g. as in real world data)



(Provably Hard)
Bounded Curvature, Unbounded Volume Regime

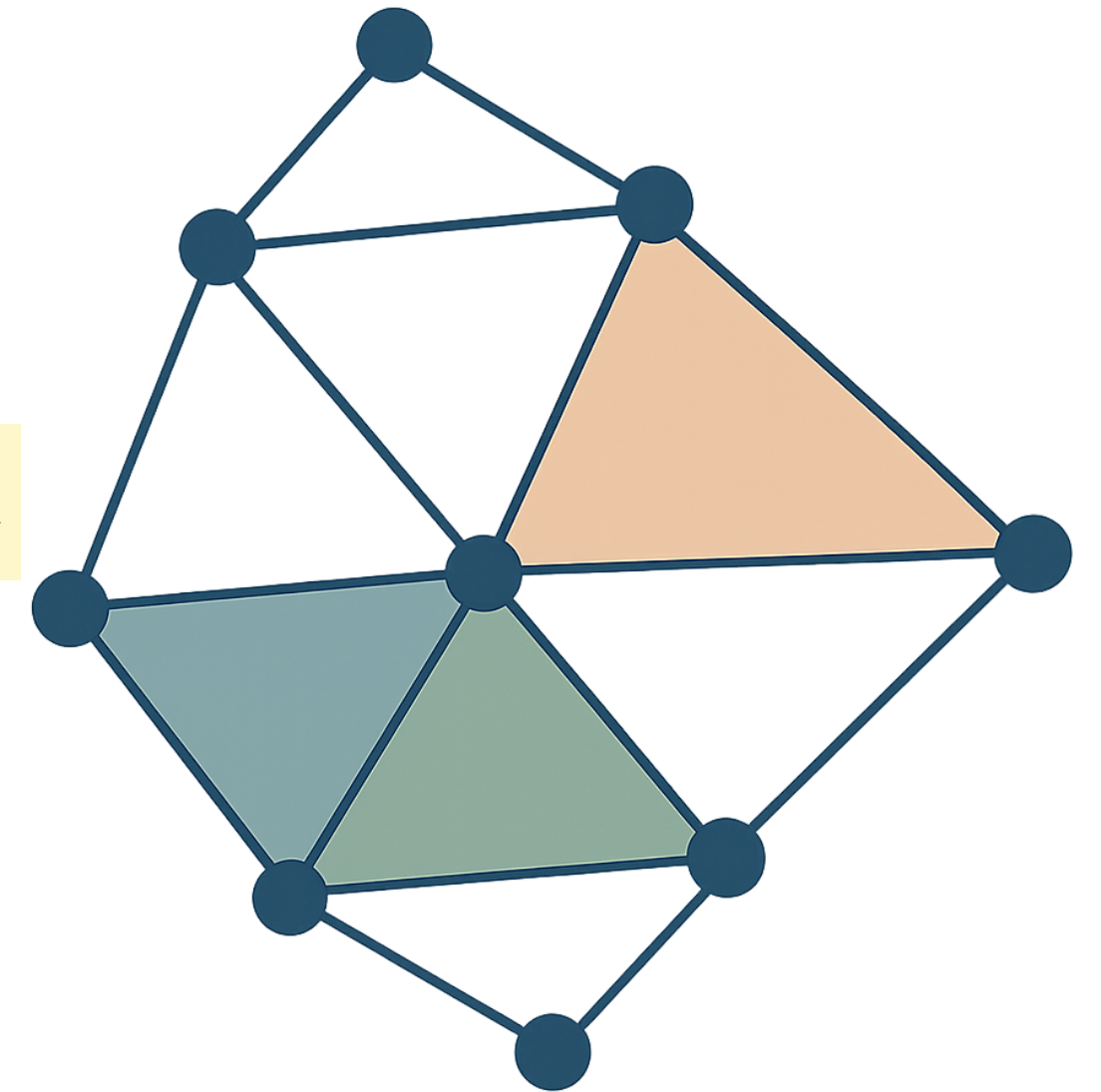
Example: Space filling manifold

B. Kiani, J. Wang, M. Weber, “Hardness of Learning Neural Networks under the Manifold Hypothesis,” NeurIPS 2024

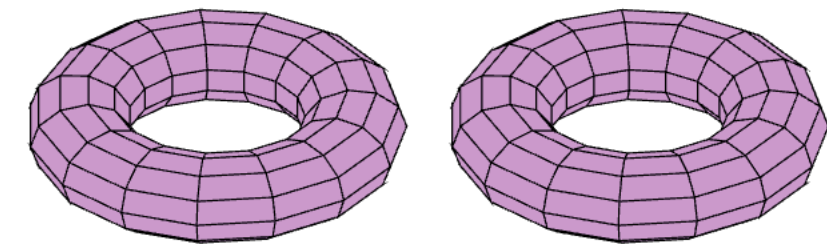
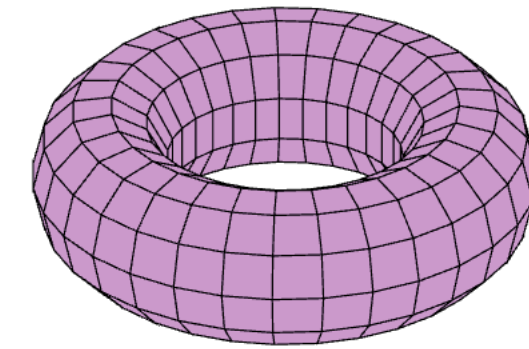
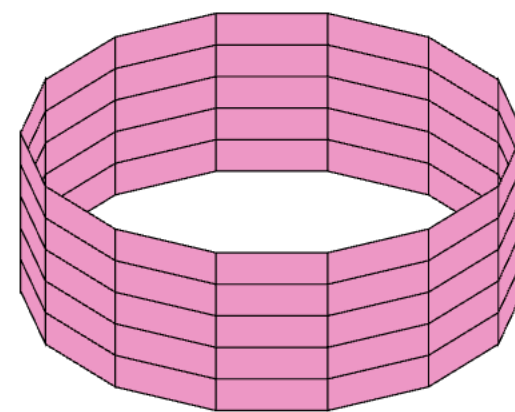
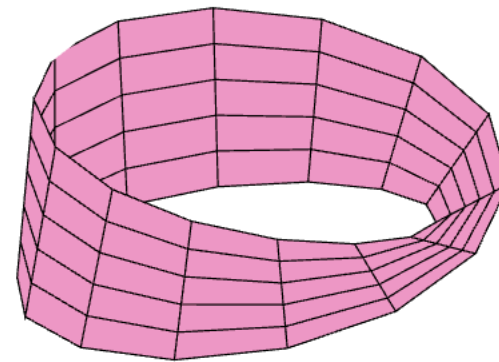
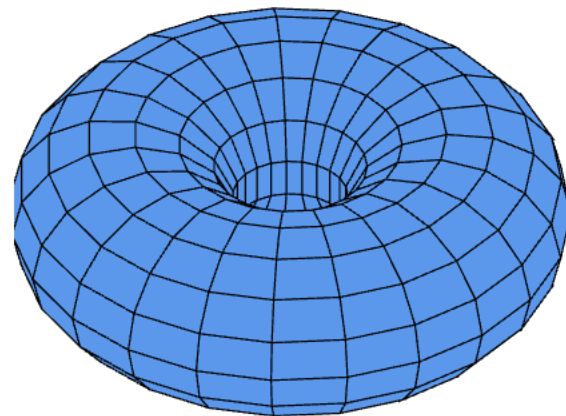
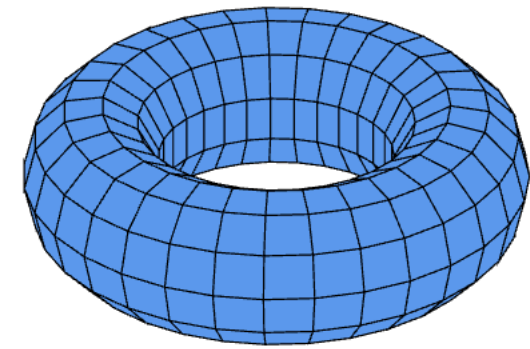
N. Trillos, M. Weber, “Continuum Limits of Ollivier's Ricci Curvature on data clouds: pointwise consistency and global lower bounds,” arXiv 2023.

Topological deep learning I

- Leverage **topological structure** in data:
 - 3D shapes, drug discovery, molecules
- Simplicial complex, cell complex, combinatorial complex
- Higher-order message-passing (**HOMP**) similar to MPNNs
- Topological features: connected components, loops, orientability, planarity



Topological deep learning II



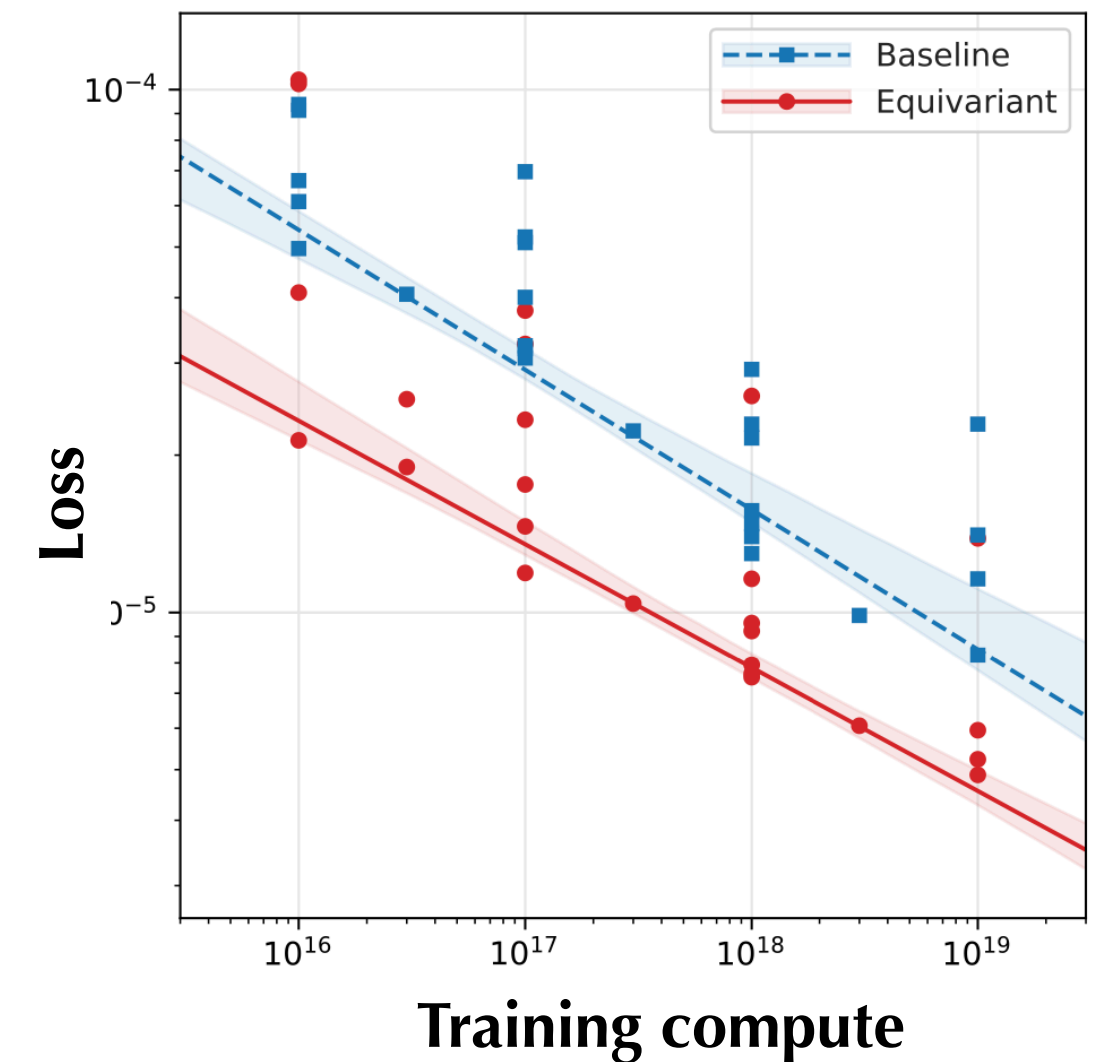
- Limitation of HOMP:
 - Diameter, orientability and planarity, connectivity (homology group)
- Multi-cellular networks (MCN) and scaled MCN (SMCN)

Encode symmetries or not?

Great successes with geometric models in the past, some new large models don't use them as much...

Efficiency:

- Inductive bias often helps with smaller / expensive data
- Specialized models may not be necessary at scale but equivariance can improve scaling laws / efficiency
(*Brehmer et al 2025*)



(*Brehmer et al 2025*)

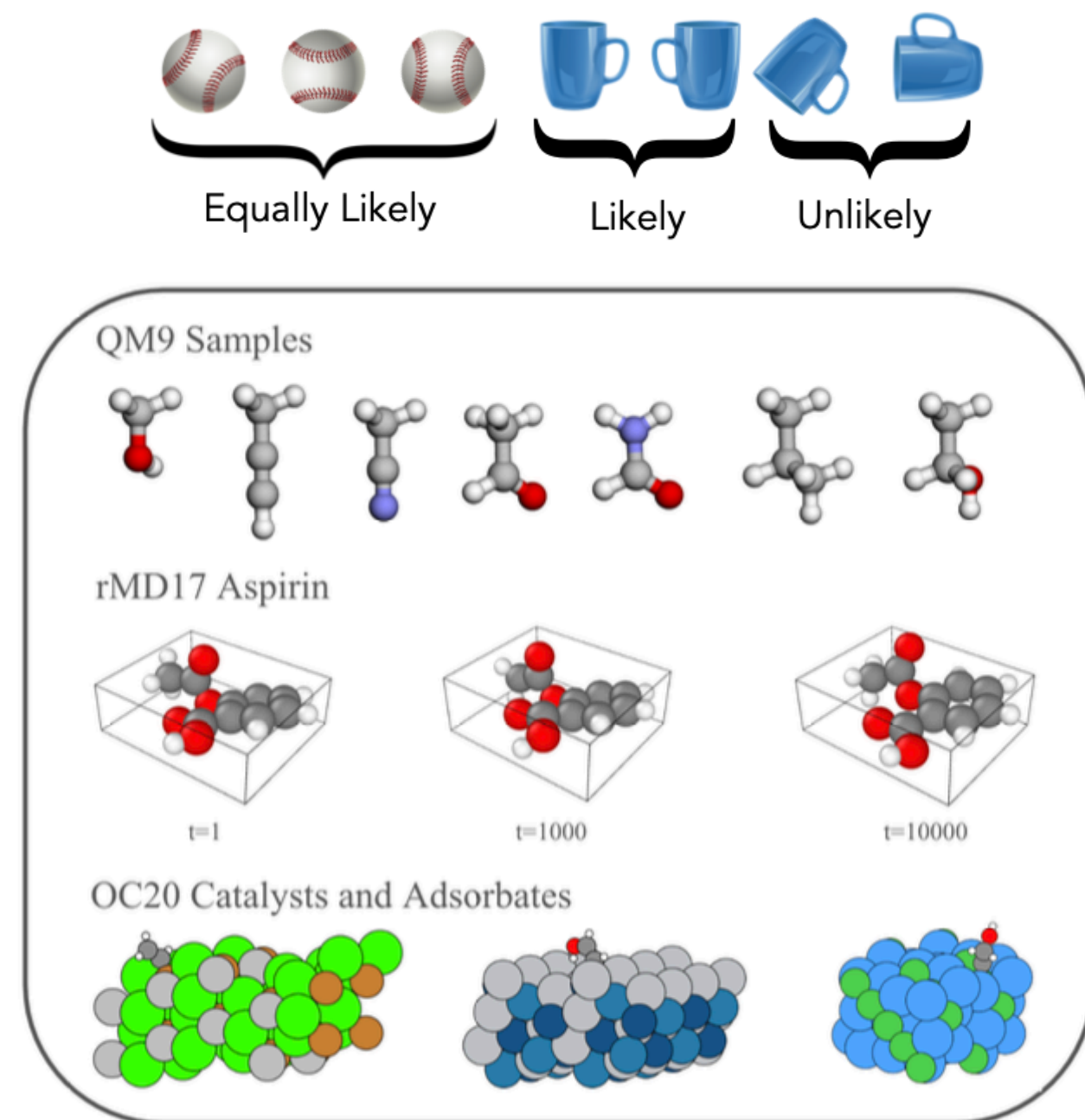
Encode symmetries or not?

Great successes with geometric models in the past, some new large models don't use them as much...

What kind of symmetries?

- Approximate or data-dependent symmetries vs. exact symmetries (e.g. eigenvectors, neural parameter symmetries)
- How relevant are the symmetries for the task?
On what data am I testing?
Some benchmark datasets are canonicalized!

(Lawrence et al 2025)



Images: Lawrence et al 2025

Encode symmetries or not?

Great successes with geometric models in the pas, some new large models don't use them as much...

Flexibility / generality

- Sometimes, symmetry can be restrictive
e.g. for generative models
-> e.g., symmetry breaking methods
- Specialized symmetries vs broad foundation models:
-> develop methods for flexible, adaptive symmetries
-> specialized models for tool use

Acknowledgements

Acknowledgements (1/3)

- We acknowledge support from:
 - The NSF AI Institute for Learning-enabled Optimization at Scale (TILOS)
 - Alexander von Humboldt Foundation
 - NSF AI Institutes Virtual Organization (AIVO)



Harvard John A. Paulson
School of Engineering
and Applied Sciences



Acknowledgements (2/3)



Andreas Loukas



Haggai Maron



Tomaso Poggio



Soledad Villar



Tess Smidt



Vikas Garg



Melanie Weber



Nadav Dym



Ron Levie



Robin Walters

Acknowledgements (3/3)

- Jegelka Research Group (MIT and TUM)
- Special thanks to Derek Lim and Eduardo Santos-Escriche
- Weber Research Group (GeoML at Harvard)



Harvard John A. Paulson
School of Engineering
and Applied Sciences



Massachusetts
Institute of
Technology

Questions, comments, further discussion?

Email: bzt@mit.edu



Thank you!