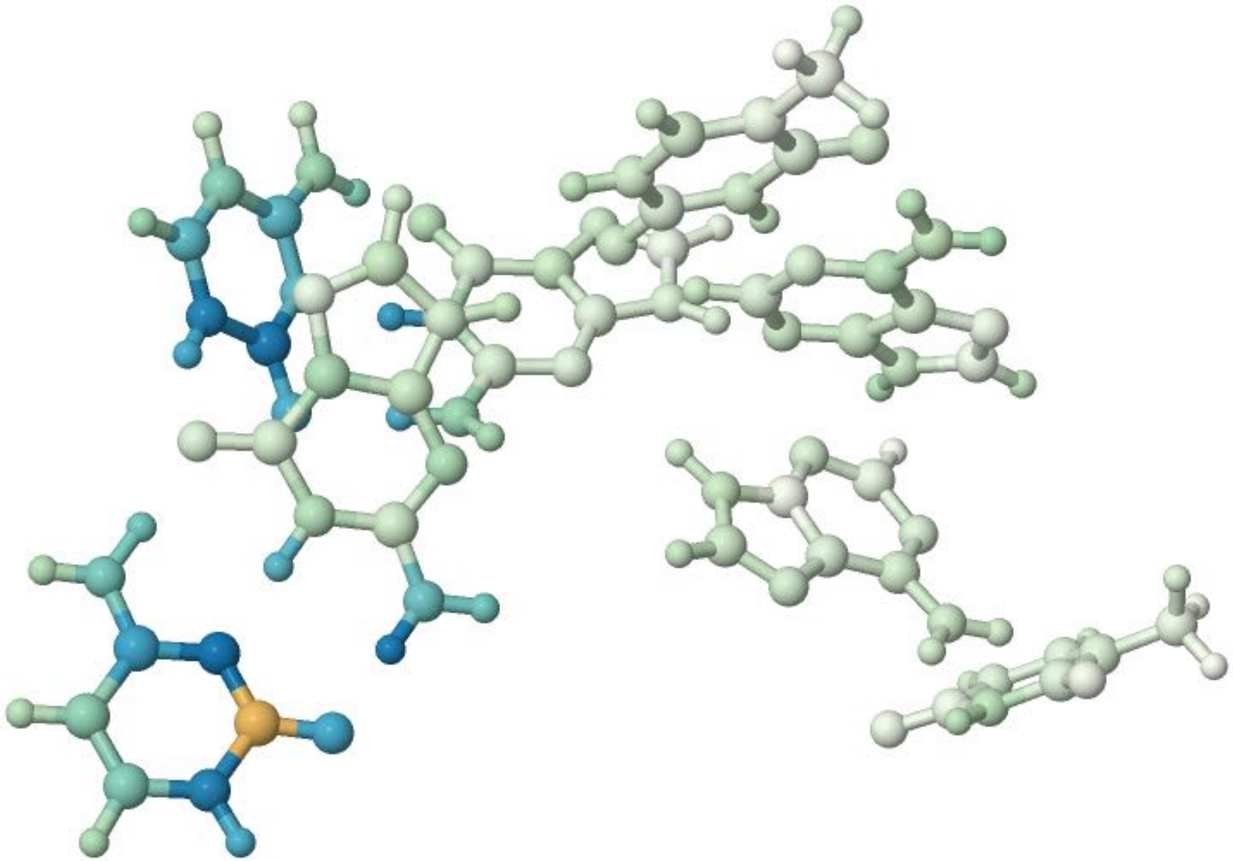
Equivariant Neural Networks

Deep Learning 2



SoSe 2023

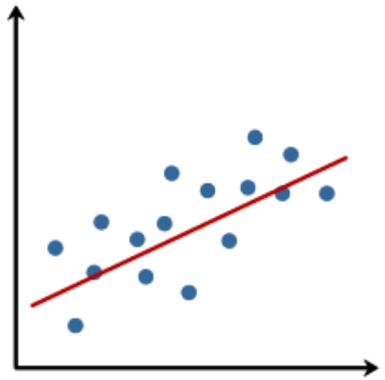
Inductive biases

Assumptions enable generalization

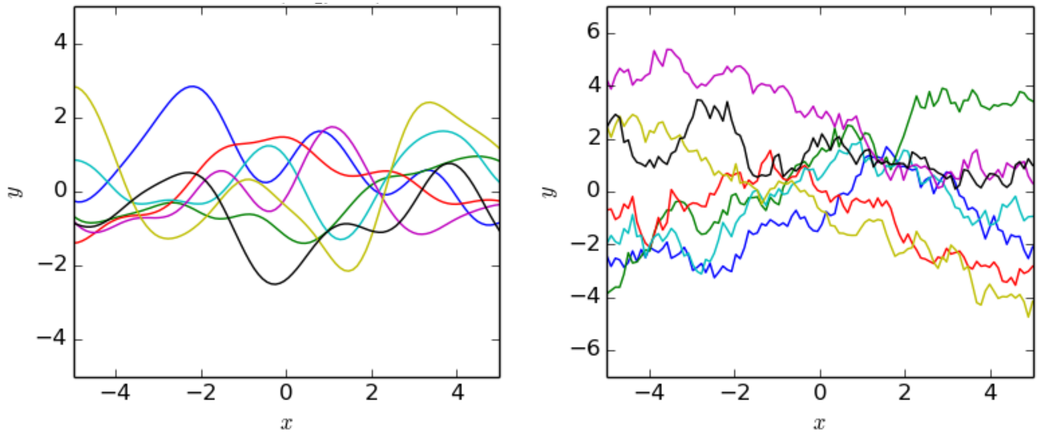
Fundamental for regression tasks: smoothness of the target function
e.g. differentiability, band-limitedness, boundedness, Lipschitz, …

$$\|f(\mathbf{x}_1) - f(\mathbf{x}_2)\| \leq C\|\mathbf{x}_1 - \mathbf{x}_2\| \quad \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$$

e.g linear functions:



e.g wide vs. narrow signal frequency bands:

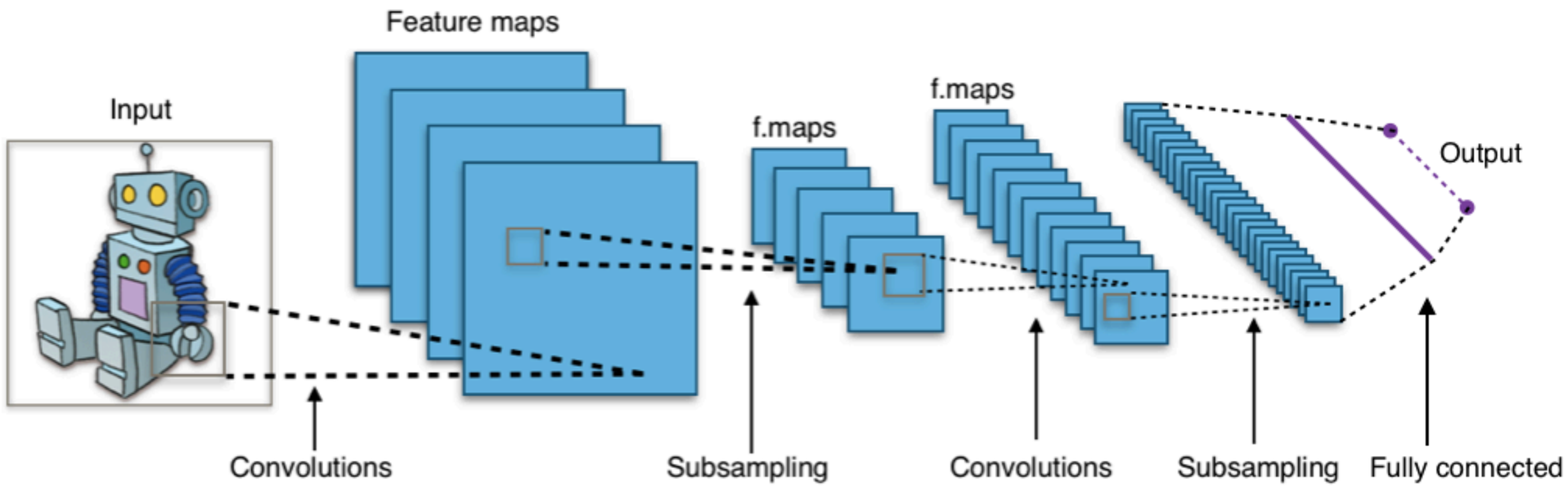


Complex target functions require **complex models** / **more training data**.

Inductive biases

Assumptions enable generalization

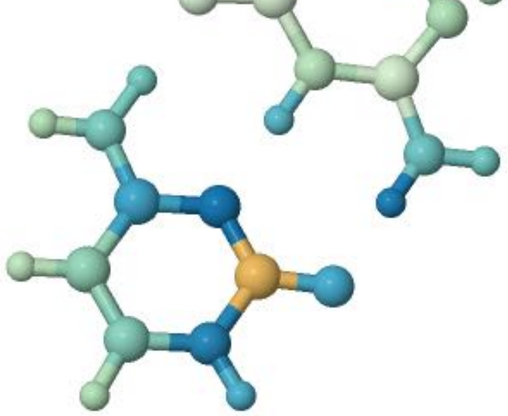
Locality assumptions
e.g. convolutional neural networks (CNNs)



Trainable weights are shared
(inputs are broken down into hierarchical local feature patterns)

Inductive biases

Assumptions enable generalization



Locality assumptions
e.g. convolutional neural networks (CNNs)

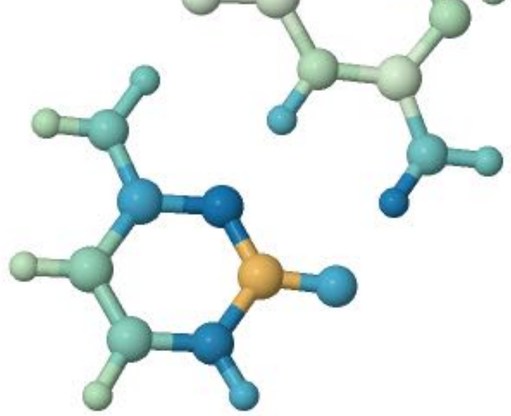
Limited interactions scales in atomic systems.

“Donor-Bridge-Acceptor"
(E)-N,N-dimethyl-4-(4-nitrostyryl)aniline

“Nearsightedness” of electronic matter

Inductive biases

Assumptions enable generalization



Conservation laws: physical symmetries

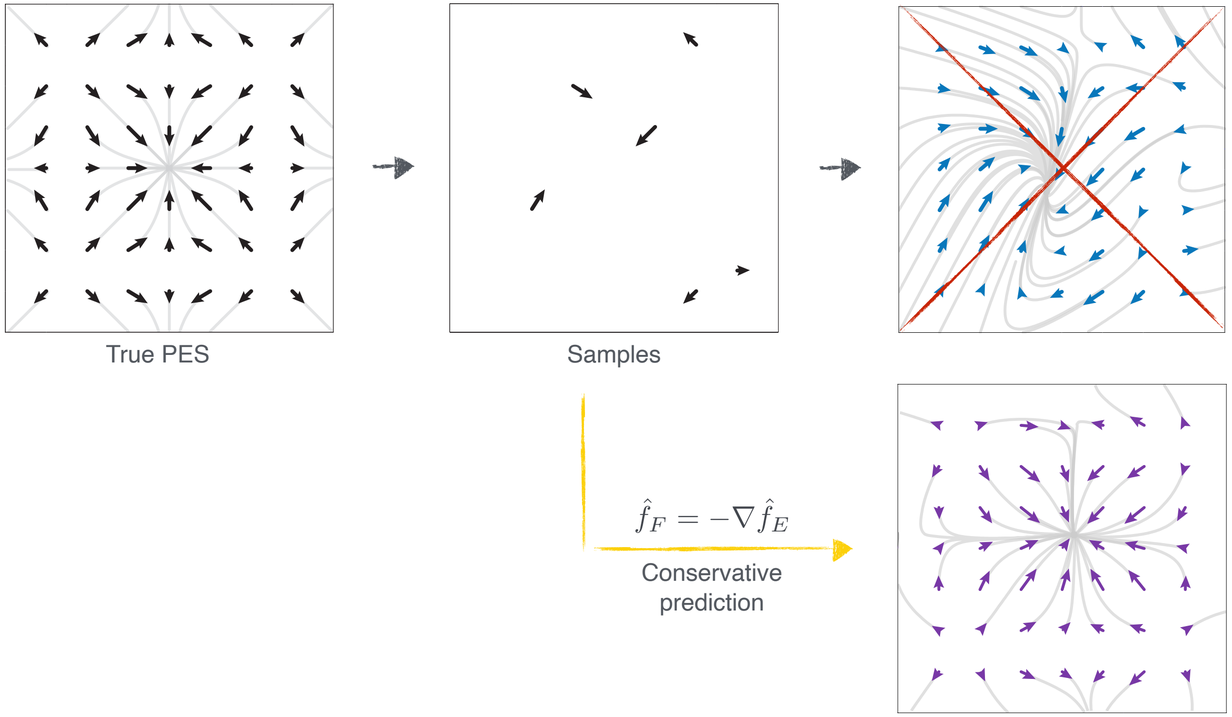
Noether's theorem:
"Conservation laws describe invariant properties of closed physical systems."

e.g. **energy-conservation** in molecular force fields $\hat{f}_F = -\nabla \hat{f}_E$ (gradient of the potential energy)

Any one of the following conditions hold:

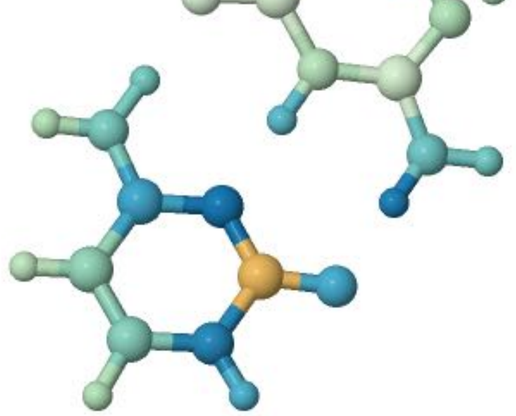
- vanishing curl: $\nabla \times \hat{f}_F = 0$
- $\int \hat{f}_F \cdot d\mathbf{r}$ is path independent
- $\oint \hat{f}_F \cdot d\mathbf{r} = 0$ (closed paths)

(temporal invariance)



Inductive biases

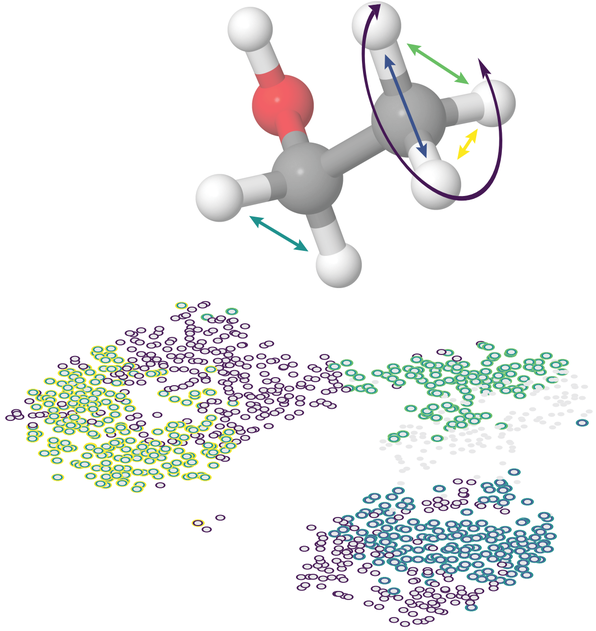
Assumptions enable generalization



Conservation laws: physical symmetries

Noether's theorem:
"Conservation laws describe invariant properties of closed physical systems."

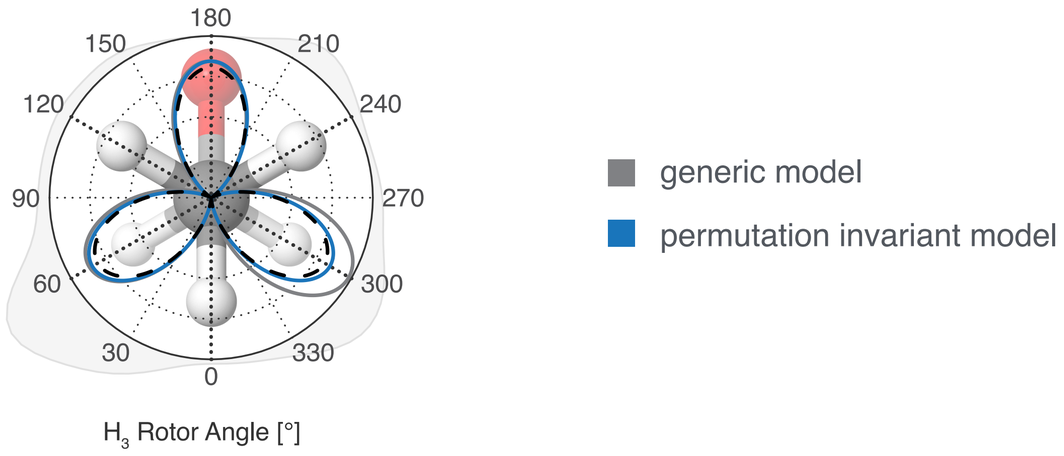
e.g. atom permutations:



(arbitrary order / indexing)

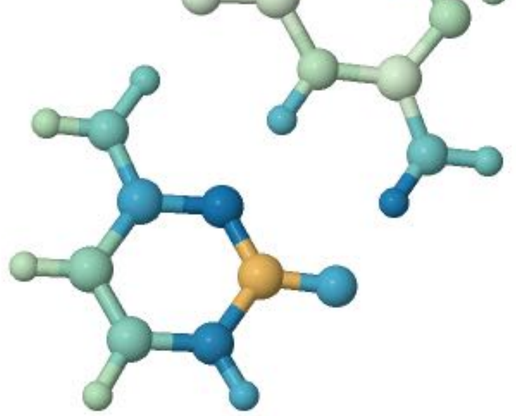
Symmetry of the methyl rotor:

$$\hat{f}_E(r_1, r_2, r_3) = \hat{f}_E(r_2, r_3, r_1) = \hat{f}_E(r_3, r_1, r_2)$$



Inductive biases

Assumptions enable generalization



Conservation laws: physical symmetries

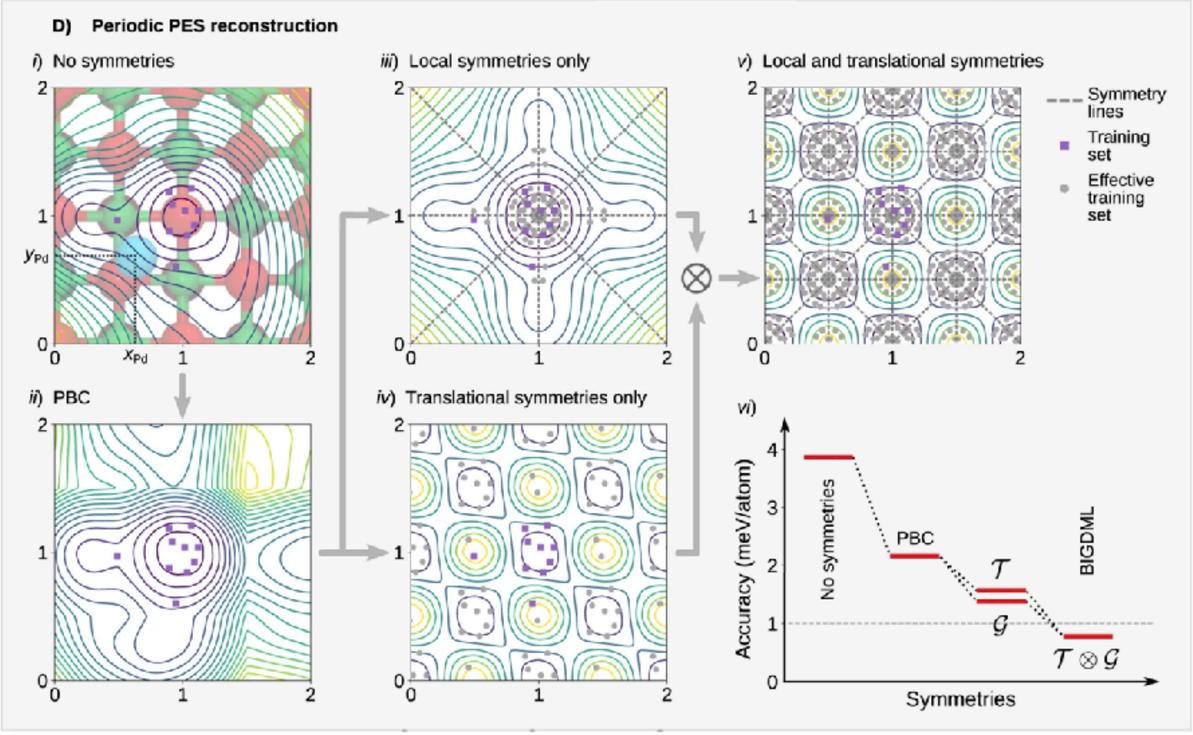
Noether's theorem:
"Conservation laws describe invariant properties of closed physical systems."

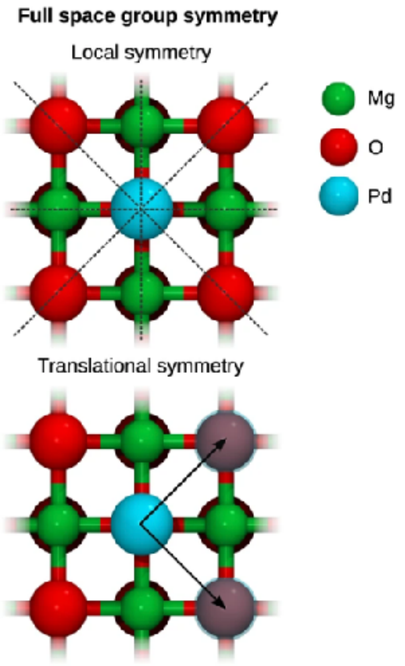
e.g. (local) translational invariance:

Molecular dynamics:

- pristine and defect-containing materials
- molecular adsorbates on surfaces

(another permutation invariance)





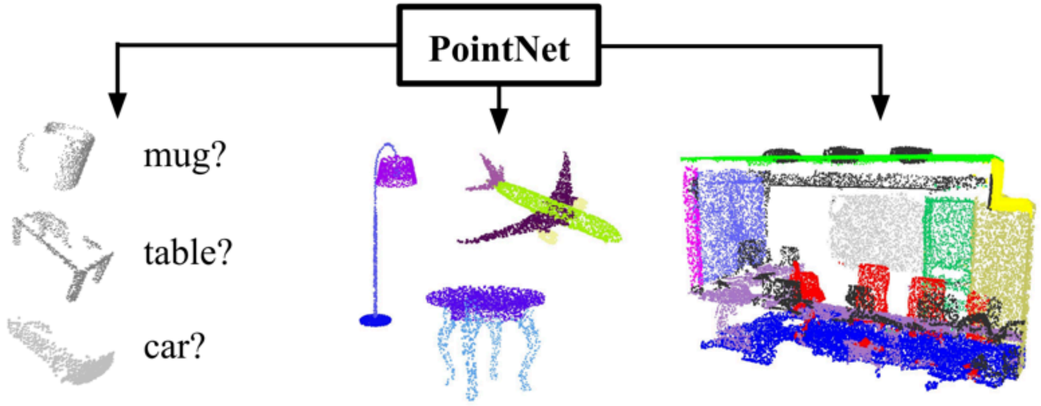
Inductive biases

Assumptions enable generalization

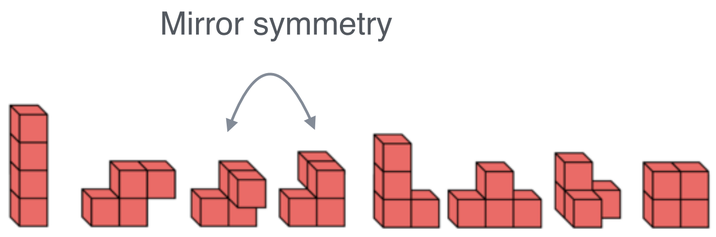
Conservation laws: physical symmetries

Noether's theorem:
"Conservation laws describe invariant properties of closed physical systems."

Symmetries are useful beyond physics
(learning on sets):



Point clouds [Qi et al., 2017]

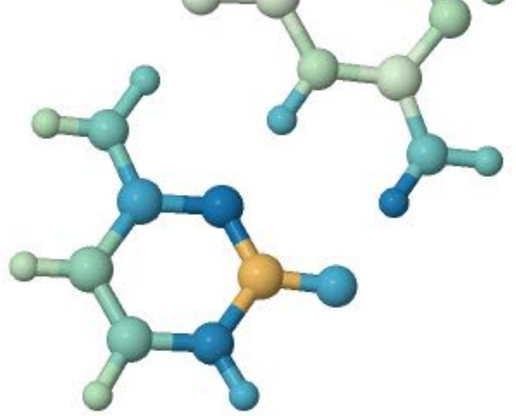


3D Tetris shapes

Shape classification (using Tensor Field Networks)
[Thomas et al., 2018]

Inductive biases

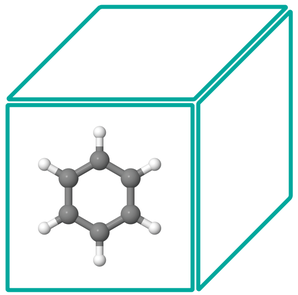
Assumptions enable generalization



Summary:

- Models endowed with strong biases perform better
- Few ML tasks are true black boxes (without any insights to exploit)
- Generic models + data (augmentation) are not robust

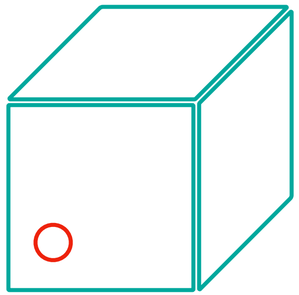
In the natural sciences:



Combinatorial state spaces

e.g. Hilbert space associated with N-particle system is d^N dimensional!

Physically feasible subspace is smaller



Conservation laws / symmetries → e.g. Noether's theorem, Pauli exclusion principle, geometric priors, etc.

Equivariance and invariance

$$\hat{f}(\mathbf{R}x) = \hat{\mathbf{R}}\hat{f}(x)$$

$$\hat{f}(\mathbf{R}x) = \hat{f}(x)$$

"Two representations of the same group action"

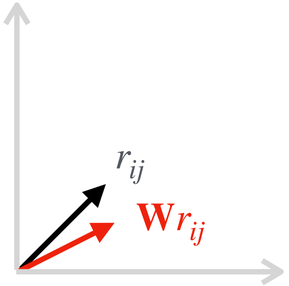
How to use symmetry groups in ML models?

Creating equivariant / invariant models

Only allow transformations / intermediates that do not affect equivariance / invariance property.

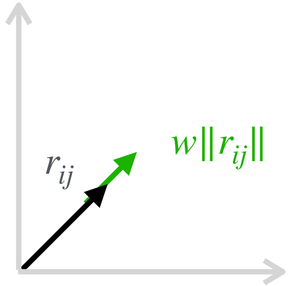
Example 1: equivariance wrt. rotation: $f(\mathbf{R}r_{ij}) = \hat{\mathbf{R}}f(r_{ij})$

$\mathbf{R}, \hat{\mathbf{R}}$ - rotation matrices



General transformation Wr_{ij} changes orientation!

> Not equivariant!

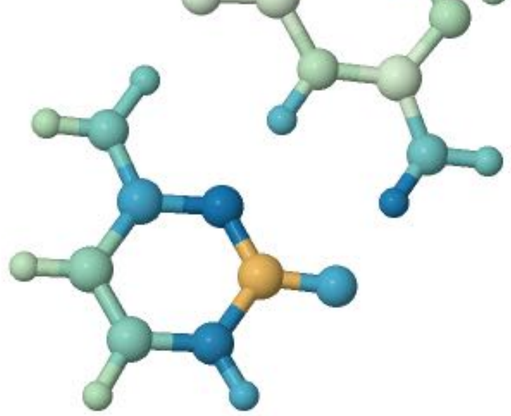


Scaling vector norm preserves rotation.

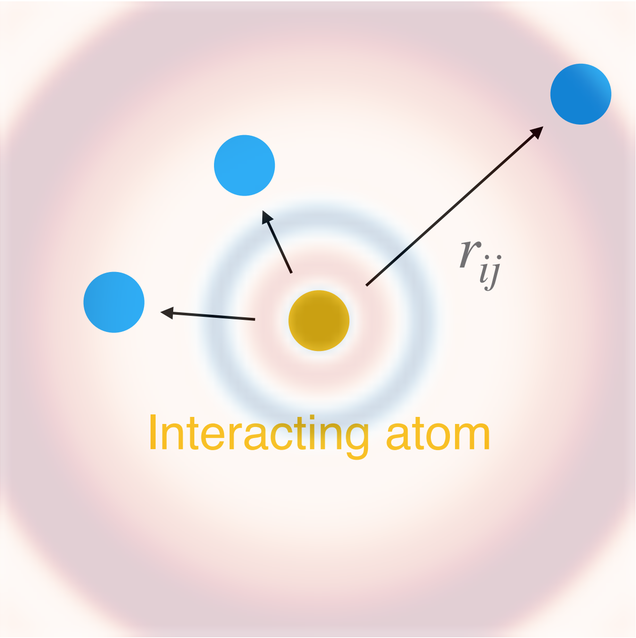
> Invariant / equivariant (if directions are preserved)

How to use symmetry groups in ML models?

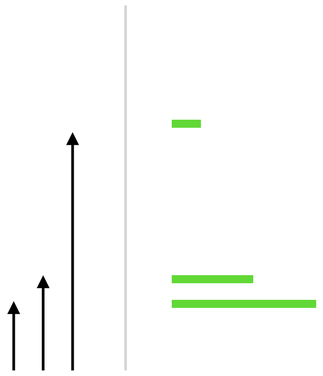
Creating equivariant / invariant models



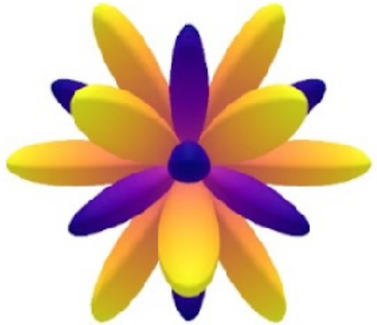
Example: Isometric convolutions in some GNNs



Learnable parameters
for each distance, e.g. $W(||r_{ij}||)$



What about directional
messages $W(r_{ij})$?



Convolution filters operate on
histograms of distances (invariant).

First: some group theory basics…

What is:

- a group
- a group action
- a representation (of a group action)

Group theory

Formal definitions

Group: A pair $(G, \cdot\,)$ of a set G and a binary operation on G that fulfills:

- Associativity: $\forall a, b, c \in G : (a \cdot b) \cdot c = a \cdot (b \cdot c)$
- Unique identity element: $\exists e : e \cdot a = a, a \cdot e = a$
- Inverse element: $\forall a \in G \; \exists b \in G : a \cdot b = e, b \cdot a = e$

Examples:

- Integer addition: $(\mathbb{Z}, +\,),\; e = 0,\; b = -\,a$
- Permutation groups (e.g. for 2 elements): $(\{\underbrace{(1)}_{e}, \underbrace{(1\,2)}_{a=b}\}, \cdot\,)$
- *Special Orthogonal Group* $SO(3)$: group of 3×3 orthogonal matrices with $det(R) = 1$, which is isomorphic to rotations in 3D
- *Special Euclidean Group* $SE(3) : SO(3) + T(3)$ (translation)

Group theory

Formal definitions: group action

Group action: The binary operation of the group acting on the members of the group.

A map $f(g, x) = gx$ from a group G and a vector space X: $f : G \times X \to X$, satisfying

- $ex = x$ (identity)
- $g(hx) = (g \cdot h)x$ (compatibility)

Example: Given two permutations $p_1 = p_2 = (1\ 2)$, we apply them to vector:

$$p_2 p_1 \begin{pmatrix} 3 \\ 4 \end{pmatrix} = p_2 \begin{pmatrix} 4 \\ 3 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$$

Group theory

Formal definitions: group representation

Group representation: The general linear group $GL(V)$ is the group of all automorphisms of V. (it includes all invertible linear transformations $V \to V$). A group homomorphism from a given group to the general linear group of a given vector space is called *group representation*.

Example 1: For the previous example $p_1 = p_2 = (1\ 2)$, the corresponding representation in a vector space $\mathbb{R}^2$ is a permutation matrix

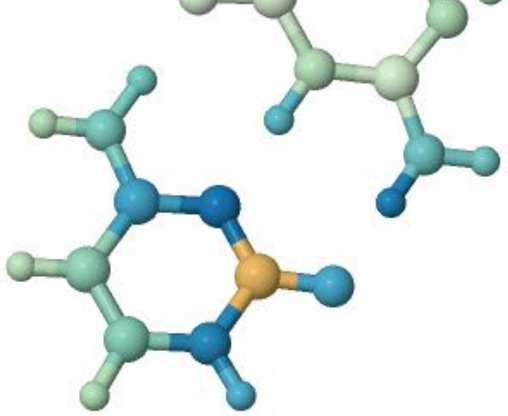
$$P = \rho(p_1) = \rho(p_2) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

The corresponding group operation is a matrix multiplication and the group action on V is the matrix-vector product:

$$\rho(p_2\ p_1)\begin{pmatrix} 3 \\ 4 \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}}_{p_2 p_1 = e}\begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$$

Conservation laws and symmetries

Example: roto-translation group SE(3)



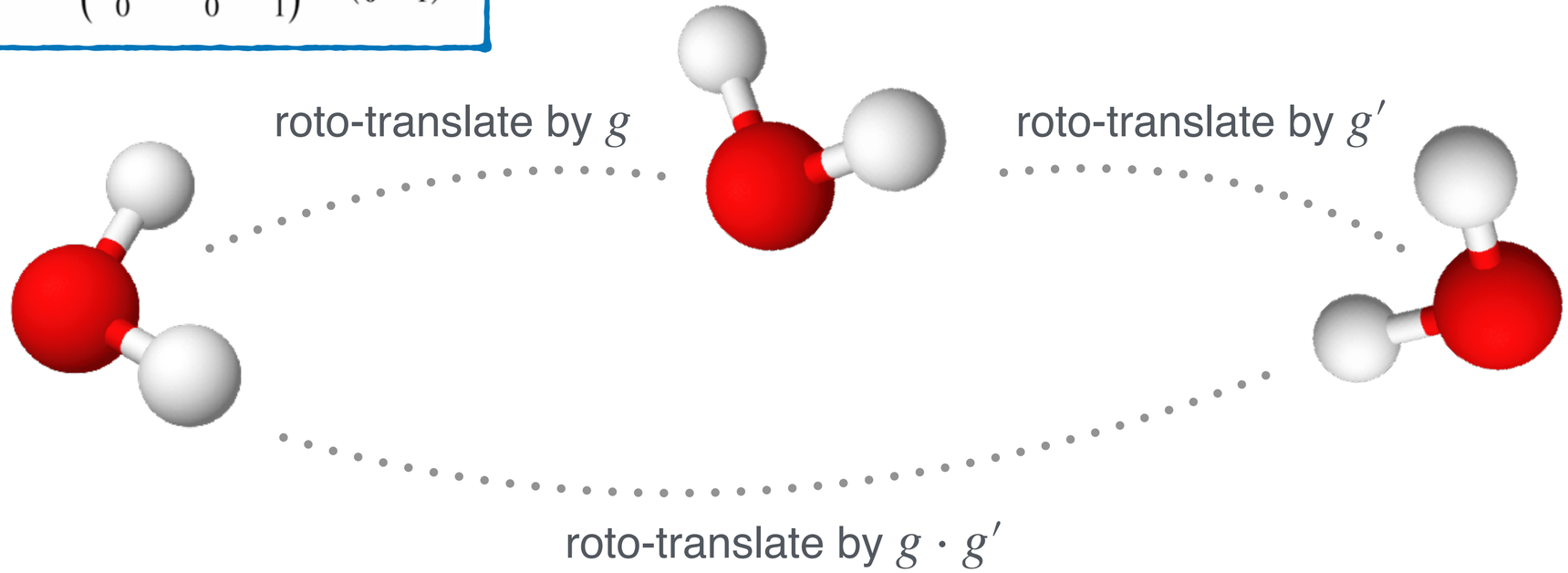
with group actions

$$g = (\mathbf{x}, \mathbf{R}_{\theta,\phi}),\ g' = (\mathbf{x}', \mathbf{R}_{\theta',\phi'})$$

$$g \cdot g' = (\mathbf{x}, \mathbf{R}_{\theta,\phi}) \cdot (\mathbf{x}', \mathbf{R}_{\theta',\phi'})$$ group product

$$g^{-1} = (-\mathbf{R}_{\theta,\phi}^{-1}\mathbf{x}, \mathbf{R}_{\theta,\phi}^{-1})$$ group inverse

in 2D: $\leftrightarrow$ $\mathbf{G} = \begin{pmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R}_\theta & \mathbf{x} \\ \mathbf{0}^T & 1 \end{pmatrix}$



Equivariance and invariance

$$\hat{f}(\mathbf{R}x) = \hat{\mathbf{R}}\hat{f}(x)$$

$$\hat{f}(\mathbf{R}x) = \hat{f}(x)$$

"Two representations of the same group action"

Generalize the notion of equivariance to arbitrary group actions: $\hat{f}(\rho_1(g)\,x) = \rho_2(g)\,\hat{f}((x) \quad \forall g \in G$, with ρ_1, ρ_2 being representation of the group action in the respective vector spaces.

How to use symmetry groups in ML models?

Ingredients for constructing equivariant / invariant models

A Lifting features to the group action

B Convolutions

C Tensor products

A

Lifting to the group action

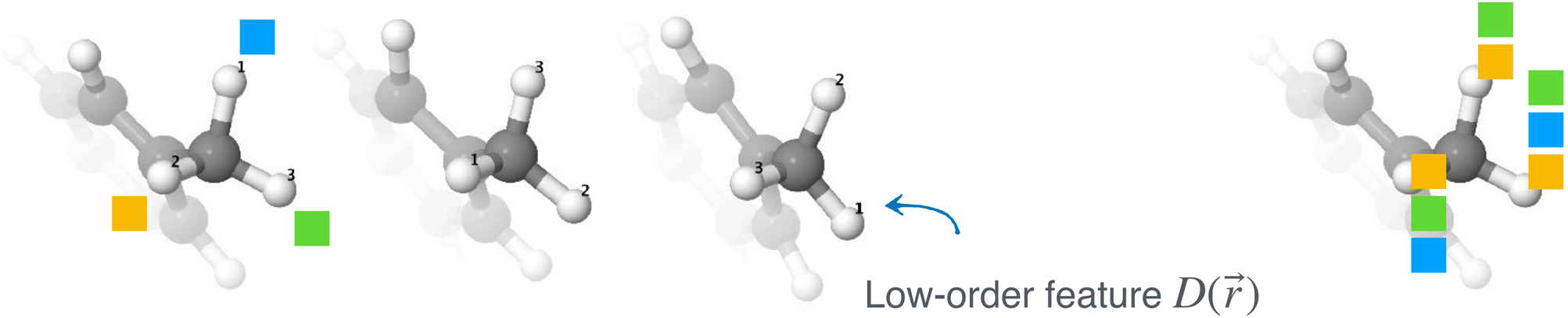
Lifting to the group

Learn on $D(\vec{r}, \tau)$ instead of $D(\vec{r})$

Example: Permutation group (discrete)

Group-lifted features

Representation of group action: $\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$



Low-order feature $D(\vec{r})$

$$\vec{r} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \qquad \mathbf{P}\begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \qquad (\mathbf{P} \circ \mathbf{P})\begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \longrightarrow \begin{pmatrix} [r_1 r_3 r_2] \\ [r_2 r_1 r_3] \\ [r_3 r_2 r_1] \end{pmatrix}$$

Features

Apply group action

(Equivariant) feature field $D(\vec{r}, \tau)$

Invariant upon summation.

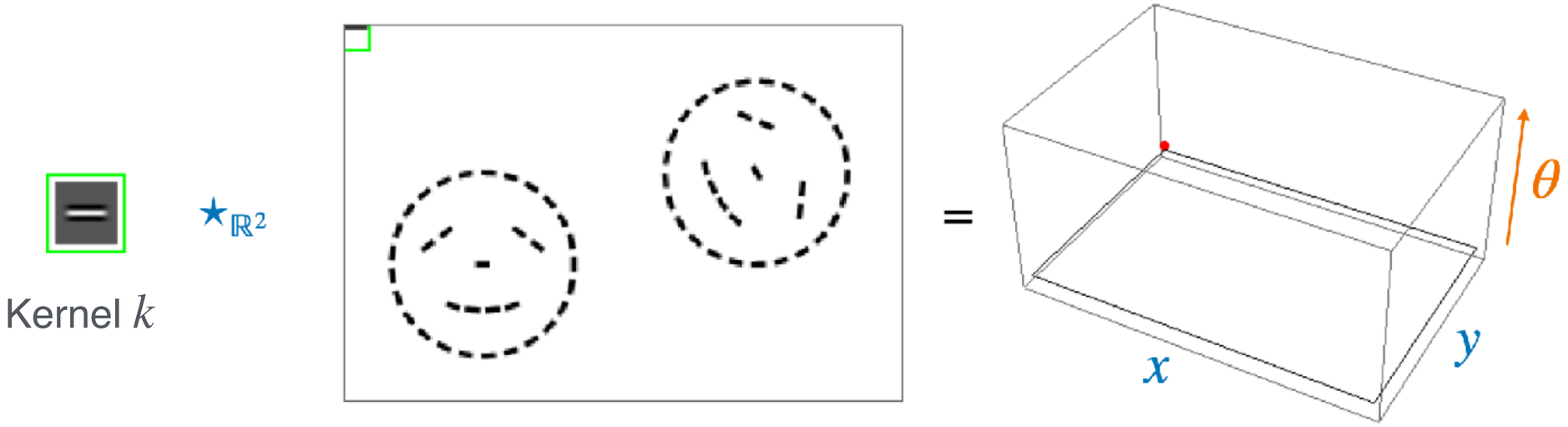
Lifting to the group

Example: 2D rotation group SO(2) (discrete)

(translation)

Lifting to the group $k(R_\theta^{-1}(x'-x))$

(rotation)



Kernel k

feature map f^{in}

feature map f^{out}

Adapted from lecture series: “An Introduction to Group Equivariant Deep Learning” by Erik Bekkers

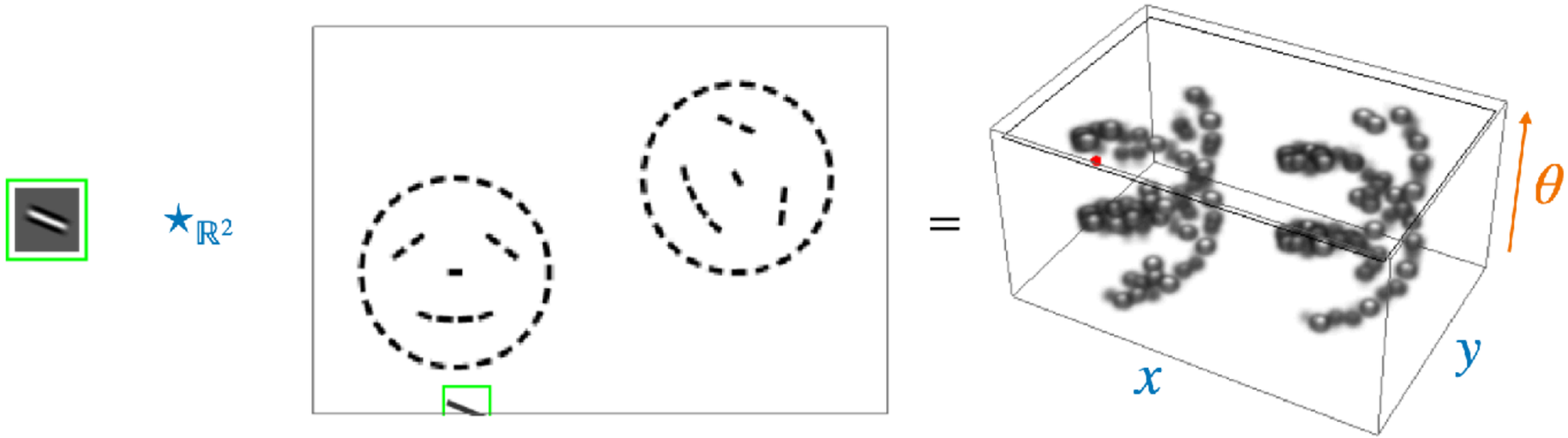
Lifting to the group

Example: 2D rotation group SO(2) (discrete)

(translation)

Lifting to the group $k(R_\theta^{-1}(x'-x))$

(rotation)



feature map f^{in}

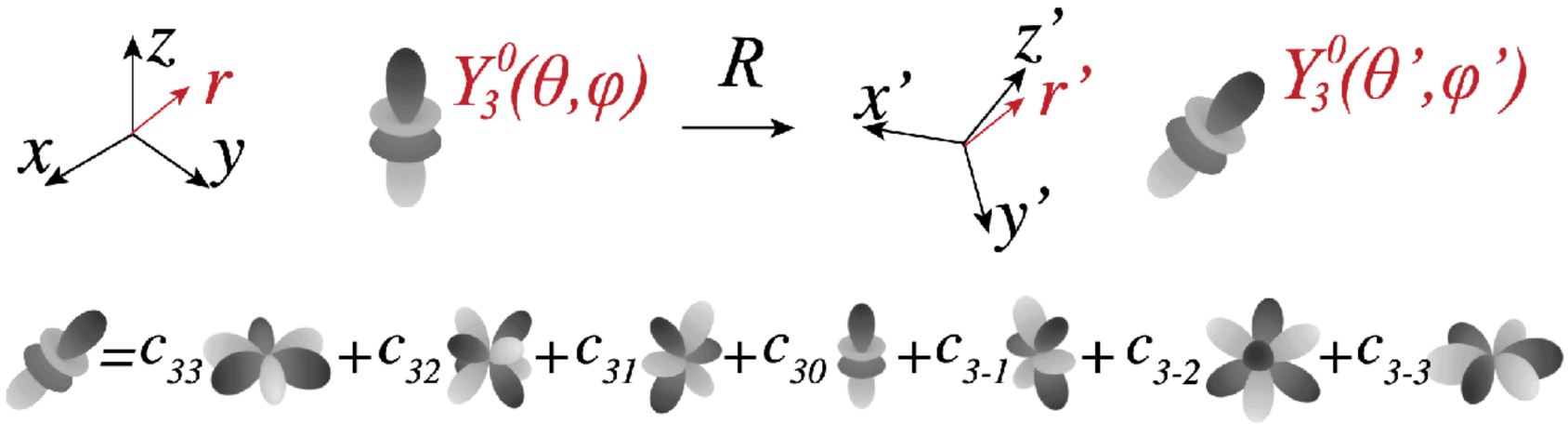
feature map f^{out}

Adapted from lecture series: “An Introduction to Group Equivariant Deep Learning” by Erik Bekkers

Lifting to the group

Example: 3D rotation group SO(3) (continuous)

To lift continuous groups: represent filter in a steerable basis



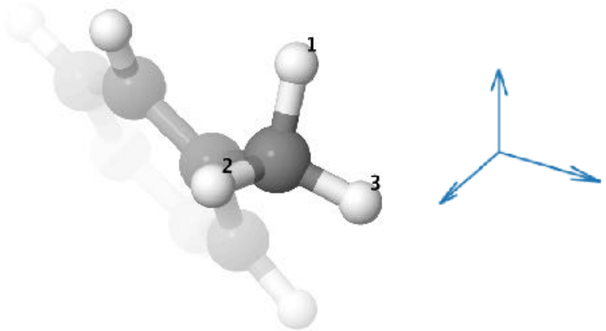
Steerable basis: Each degree l of spherical harmonics transforms into each other.

Wigner-D matrices are (irreducible) representations of the group action (rotation) in this space.

Lifting to the group

Example: 3D rotation group SO(3) (continuous)

Group lifting to continuous groups



From point cloud…

… to density representation
(in a “steerable” basis)

Spherical harmonics basis

A steerable basis for functions on the sphere

Order $l = 0$

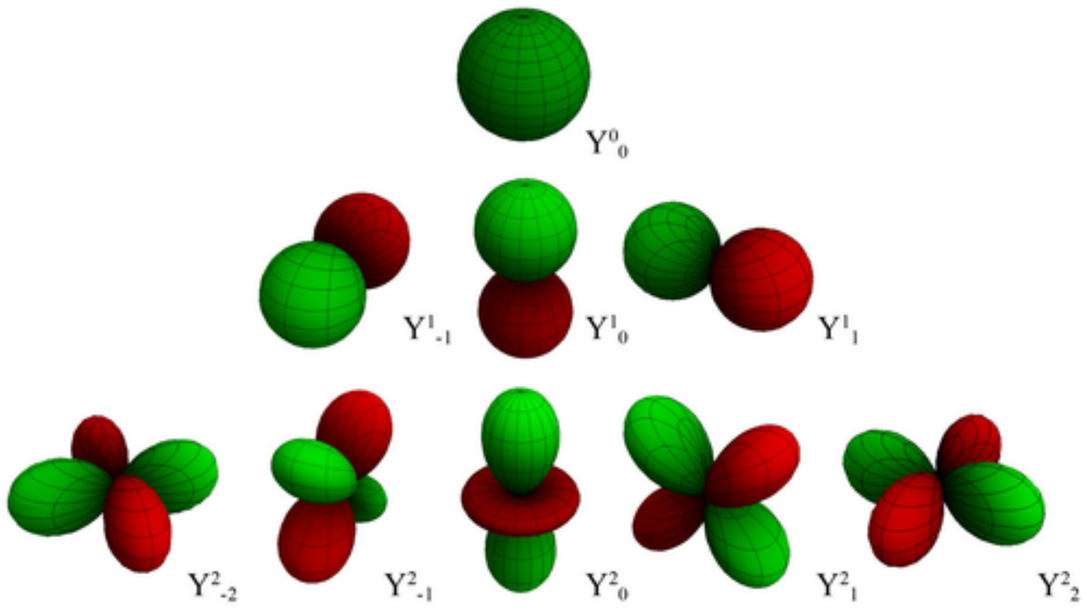
$$Y_0^0 = \frac{1}{2}\sqrt{\frac{1}{\pi}}$$

Order $l = 1$

$$Y_{-1}^1 = \frac{1}{2}\sqrt{\frac{3}{\pi}}\,\frac{y}{r} \qquad m = -1$$

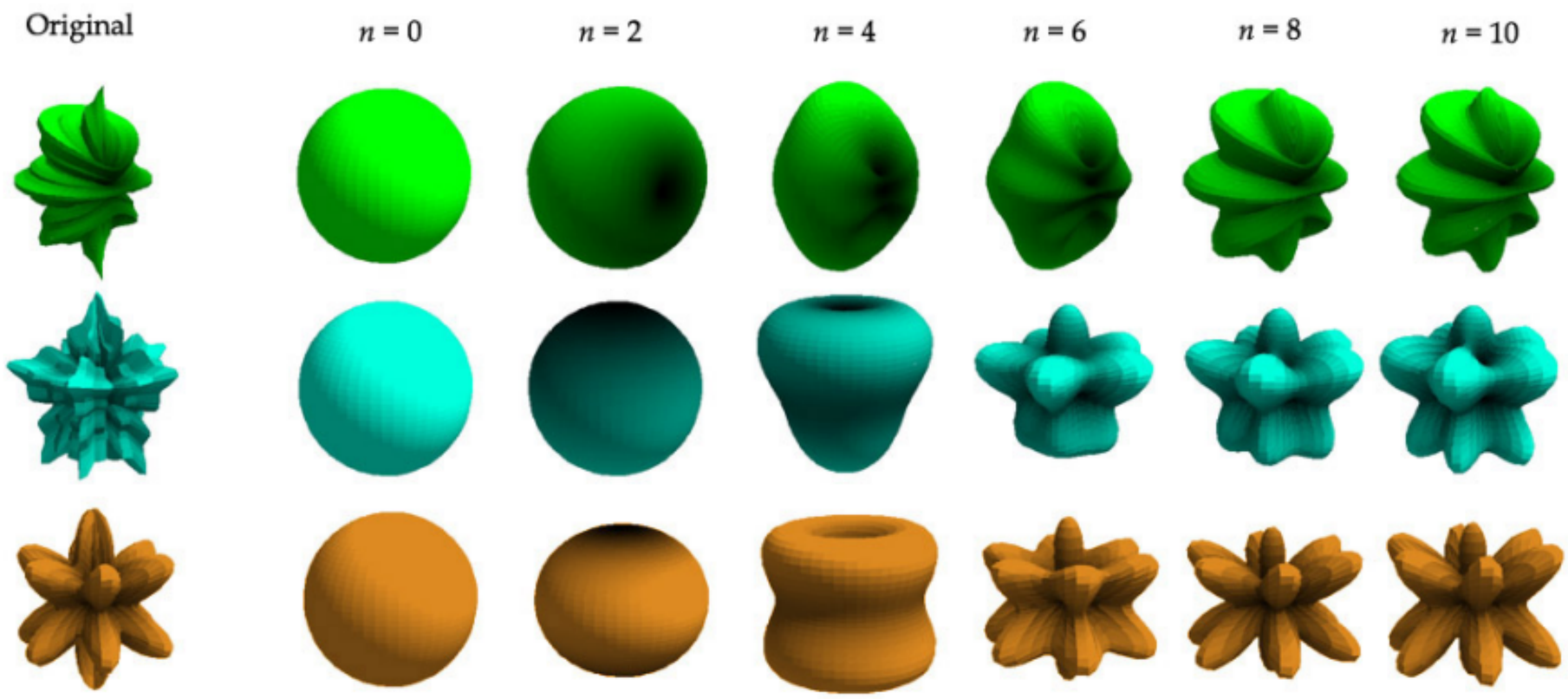
$$Y_0^1 = \frac{1}{2}\sqrt{\frac{3}{\pi}}\,\frac{z}{r} \qquad m = 0$$

$$Y_1^1 = \frac{1}{2}\sqrt{\frac{3}{\pi}}\,\frac{x}{r} \qquad m = 1$$



Spherical harmonics basis

Representing function on the sphere



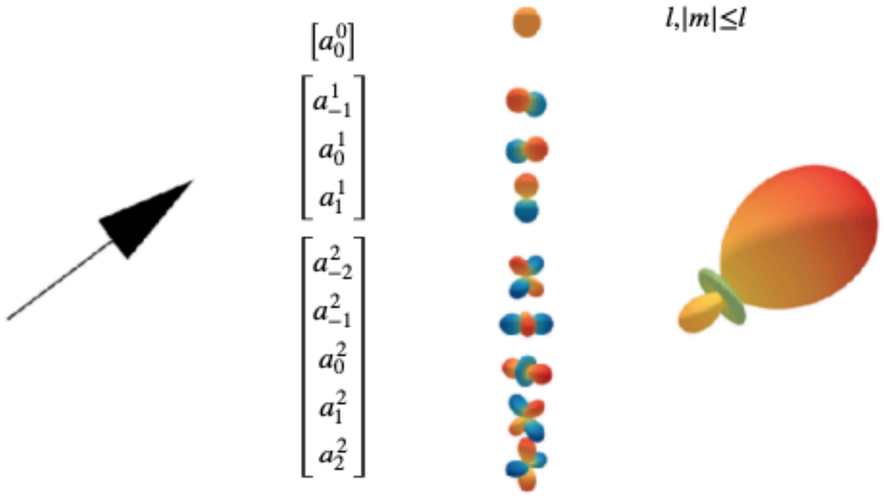
Lifting to the group

Example: 3D rotation group SO(3)

Direction

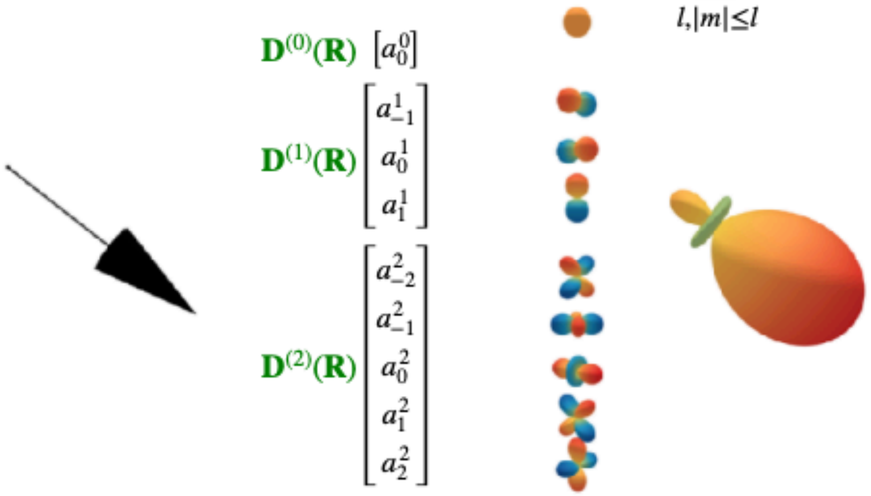
Representation in Spherical Harmonics

$\mathbf{n}$ $\quad \hat{\mathbf{a}}^{(l)} = Y(\mathbf{n}) \quad Y(\cdot) \quad \sum_{l,|m|\leq l} a^l_m Y^{(l)}_m(\cdot)$



Irreps of SO(3) (Wigner D-matrices)

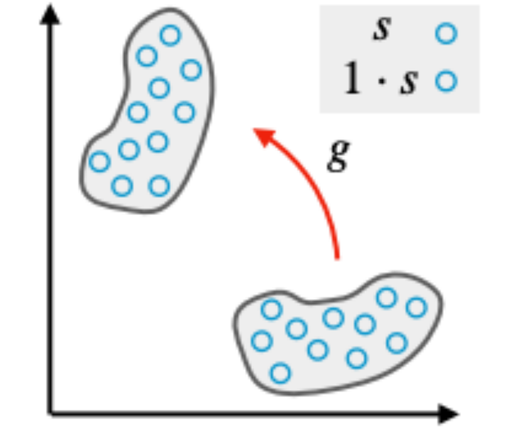
$\mathbf{R}\,\mathbf{n} \quad \hat{\mathbf{a}}^{(l)} = \mathbf{D}^{(l)}(\mathbf{R}) Y^{(l)}(\mathbf{n}) \quad Y(\cdot) \quad \sum_{l,|m|\leq l} a^l_m Y^{(l)}_m(\cdot)$



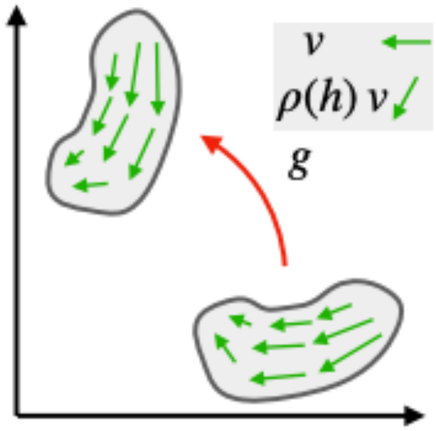
Adapted from lecture series: “An Introduction to Group Equivariant Deep Learning” by Erik Bekkers

Feature vector fields

Equivariance via “steerable” feature fields



Scalar field ($\rho(h) = 1$)



vector field

A feature field that transforms via a representation h of a group action.

Figure adapted from: Weiler, M., & Cesa, G. (2019). General e(2)-equivariant steerable CNNs. NeurIPS

B

Convolutions

… in the space lifted to the group action: **equivariant**

… integration over lifting dimension: **invariant** space

Convolutions / cross-correlations

Fundamental building block in ML: pattern matching

Linear integral operator

$$\hat{f}(\vec{x}) = T_k f(\vec{x}) = \int_{\mathcal{X}} k(\vec{x}, \vec{x}') f(\vec{x}') \, \mathrm{d}\vec{x}'$$

Operator response

Acts on (scalar) features

Convolutional neural networks

Generalizations:

- message-passing
- self-attention

Deep architectures

Discrete analogon: $\hat{f}(\vec{x}) = K\vec{\alpha} = \sum_{i}^{M} k(\vec{x}, \vec{x}_i)\alpha_i$

Selects model from Hilbert space of eigenfunctions:

$$T_k \varphi_i = \lambda_i \varphi_i$$

Kernel regression

Group convolution

Generalization to construct equivariant models

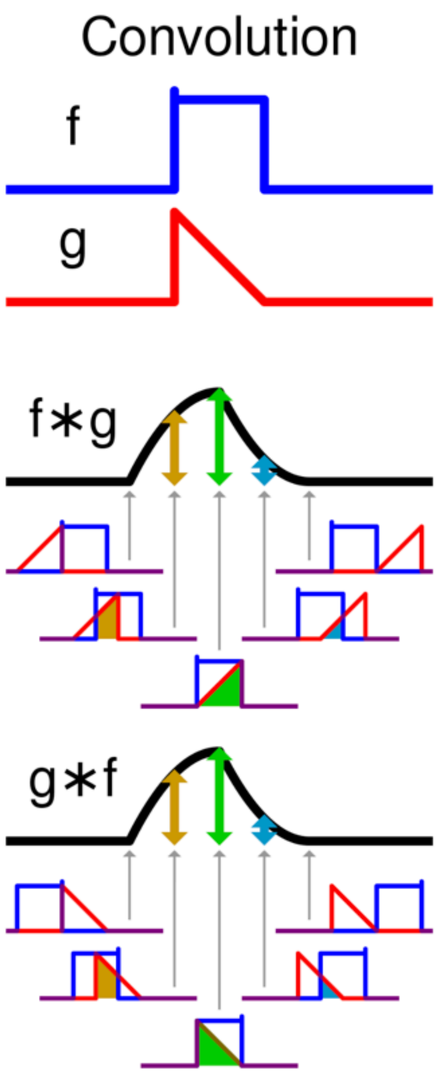
Convolutions represent an equivariant map to the translation group

$$(f \star W)(x) = \sum_{y \in V} f(y) W(y - x),$$

where f is the signal (e.g. an image) and W is the convolution filter.

Group convolutions: Generalization to arbitrary groups (note: translation is switched with desired group operation):

$$(f \star W)(x) = \sum_{g \in G} f(g) W(g^{-1} x)$$



Convolutions / cross-correlations

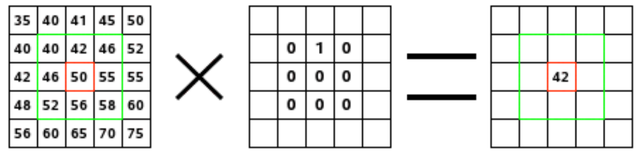
Fundamental building block in ML: pattern matching

Linear integral operator

$$\hat{f}(\vec{x}) = T_k f(\vec{x}) = \int_{\mathcal{X}} k(\vec{x}, \vec{x}') f(\vec{x}')\, \mathrm{d}\vec{x}'$$

Operator response

Acts on (scalar) features



(discrete convolution example)

“Convolutions” on feature fields

Generalization to feature fields

$$T_k f(\vec{x}) = \int_{\mathcal{X}} k(\vec{x}, \vec{x}') \otimes f(\vec{x}')\, \mathrm{d}\vec{x}'$$

Tensor product lets (high-order) features interact.

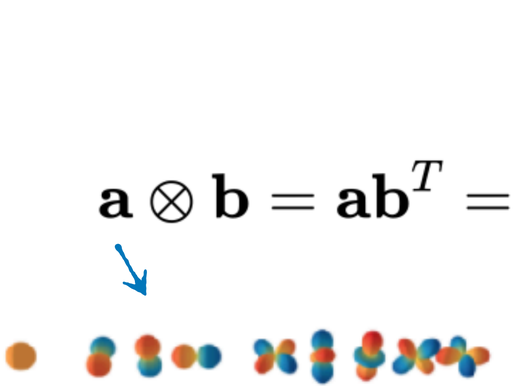
C

Tensor products

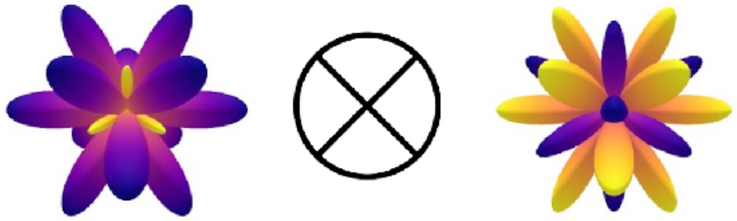
Coupling & decoupling of different-order features

Tensor products

Tensor product between two vectors $\mathbf{a} \in V_{l_a}$ and $\mathbf{b} \in V_{l_b}$.



$$\mathbf{a} \otimes \mathbf{b} = \mathbf{a}\mathbf{b}^T = \begin{pmatrix} a_1 b_1 & a_1 b_2 & \cdots \\ a_2 b_1 & a_2 b_2 & \\ \vdots & & \ddots \end{pmatrix}$$



Tensor rotates via $\mathbf{a} \mapsto \mathbf{D}^{(l_a)}(\mathbf{R})$ and $\mathbf{b} \mapsto \mathbf{D}^{(l_b)}(\mathbf{R})$

$$\mathbf{a} \otimes \mathbf{b} \mapsto \mathbf{D}^{(l_a)}(\mathbf{R})\mathbf{a} \otimes \mathbf{b}\mathbf{D}^{(l_b)}(\mathbf{R})^T$$

Tensor product is "steerable"

$$\mathrm{vec}(\mathbf{a} \otimes \mathbf{b}) \mapsto \left(\mathbf{D}^{(l_b)}(\mathbf{R}^{-1}) \otimes \mathbf{D}^{(l_a)}(\mathbf{R})\right) \mathrm{vec}(\mathbf{a} \otimes \mathbf{b})$$

Figure adapted from: Lecture: "Neural Networks with Euclidean Symmetry for the Physical Sciences", Tess Smidt

Coupling & decoupling of different-order features

Tensor products

Tensor product is "steerable"

$$\mathrm{vec}(\mathbf{a} \otimes \mathbf{b}) \mapsto \left(\mathbf{D}^{(l_b)}\left(\mathbf{R}^{-1}\right) \otimes \mathbf{D}^{(l_a)}(\mathbf{R})\right) \mathrm{vec}(\mathbf{a} \otimes \mathbf{b})$$

Clebsch-Gordan coefficients diagonalize the tensor product!

$$\mathbf{Q}^{-1} \begin{pmatrix} \mathbf{D}^{l_1}(\mathbf{R}) & 0 & 0 & 0 \\ 0 & \mathbf{D}^{l_2}(\mathbf{R}) & 0 & 0 \\ 0 & 0 & \mathbf{D}^{l_3}(\mathbf{R}) & 0 \\ 0 & 0 & 0 & \ddots \end{pmatrix} \mathbf{Q} \quad \underline{\mathrm{vec}(\mathbf{a} \otimes \mathbf{b})}$$

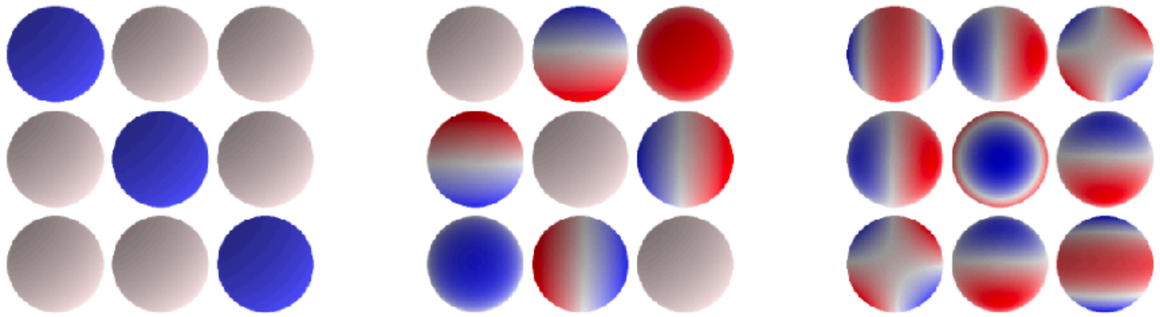
Decouples features of different orders again!

Figure adapted from: Lecture: "Neural Networks with Euclidean Symmetry for the Physical Sciences", Tess Smidt

Lifting to the group

Example: irreducible SO(3) features (continuous)

Representation of group action: rotation matrix $R^{3\times3}$



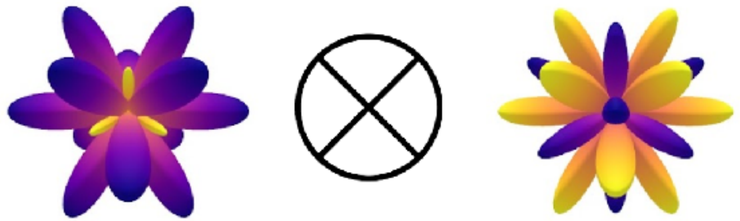
trace (1 dimension)	anti-symmetric (3 dimensions)	traceless symmetric (5 dimensions)
(identity)	(curl $\nabla \wedge$)	(gradient of divergence $\nabla \nabla \cdot$)

Invariant component

Equivariant components

Group convolution

Generalization to construct equivariant models



SE(3) equivariant convolutions:

$$(f \star W)(x, L, M) = \sum_{y, l_1, m_1, l_2, m_2} f(y, l_1, m_1) W(y - x, l_1, m_1, l_2, m_2)$$

(... for each YLM feature)

$$W(d, l_1, m_1, l_2, m_2) = C^{(L,M)}_{(l1,m2),(l_2,m_2)} w_{l1}(\|d\|) \; Y^{l2}_{m2}(d/\|d\|)$$

Clebsch-Gordon coefficients $C^{(L,M)}_{(l1,m2),(l_2,m_2)}$ determine how the irreps of the input features and filters couple with each other to produce a given output feature.

Summary

Using symmetry constraints as inductive bias in ML models

A Lifting features to the group action

B Convolutions

C Tensor products