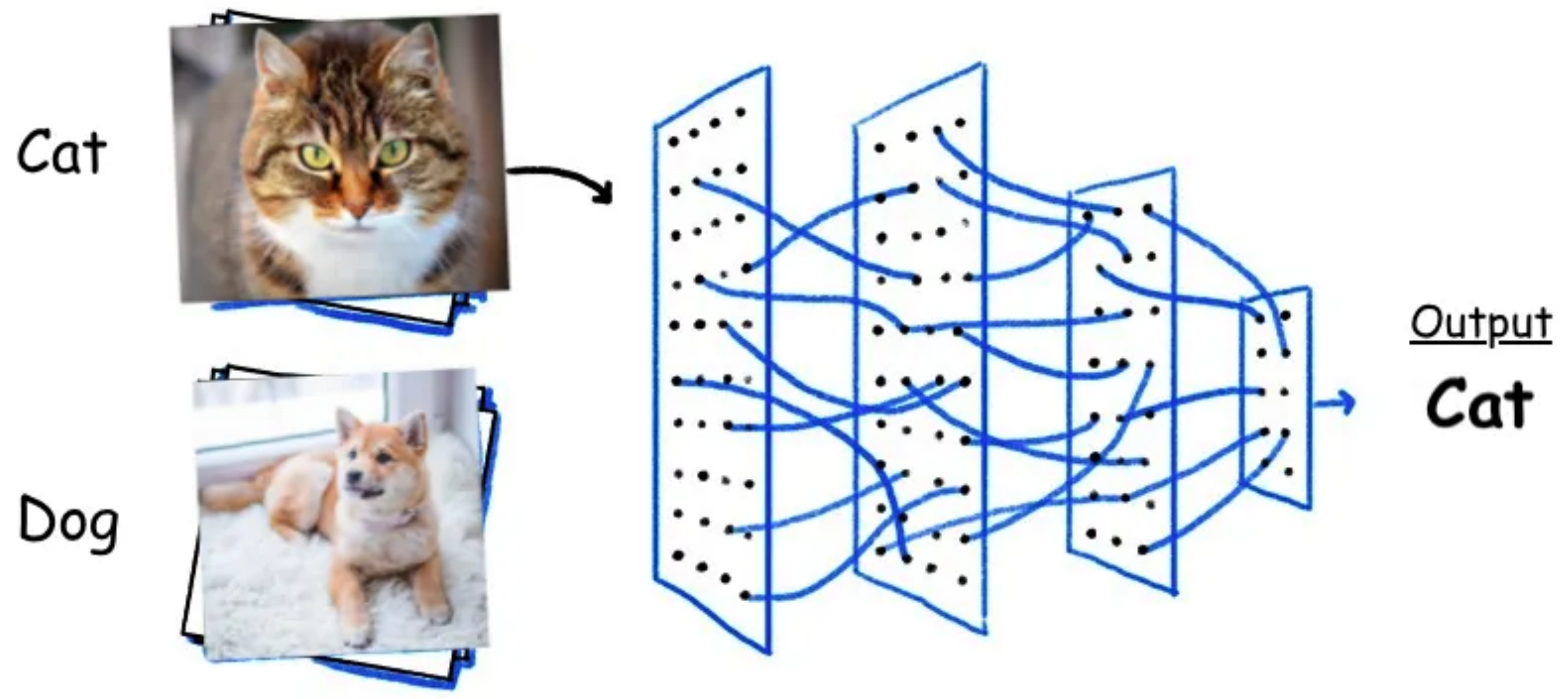
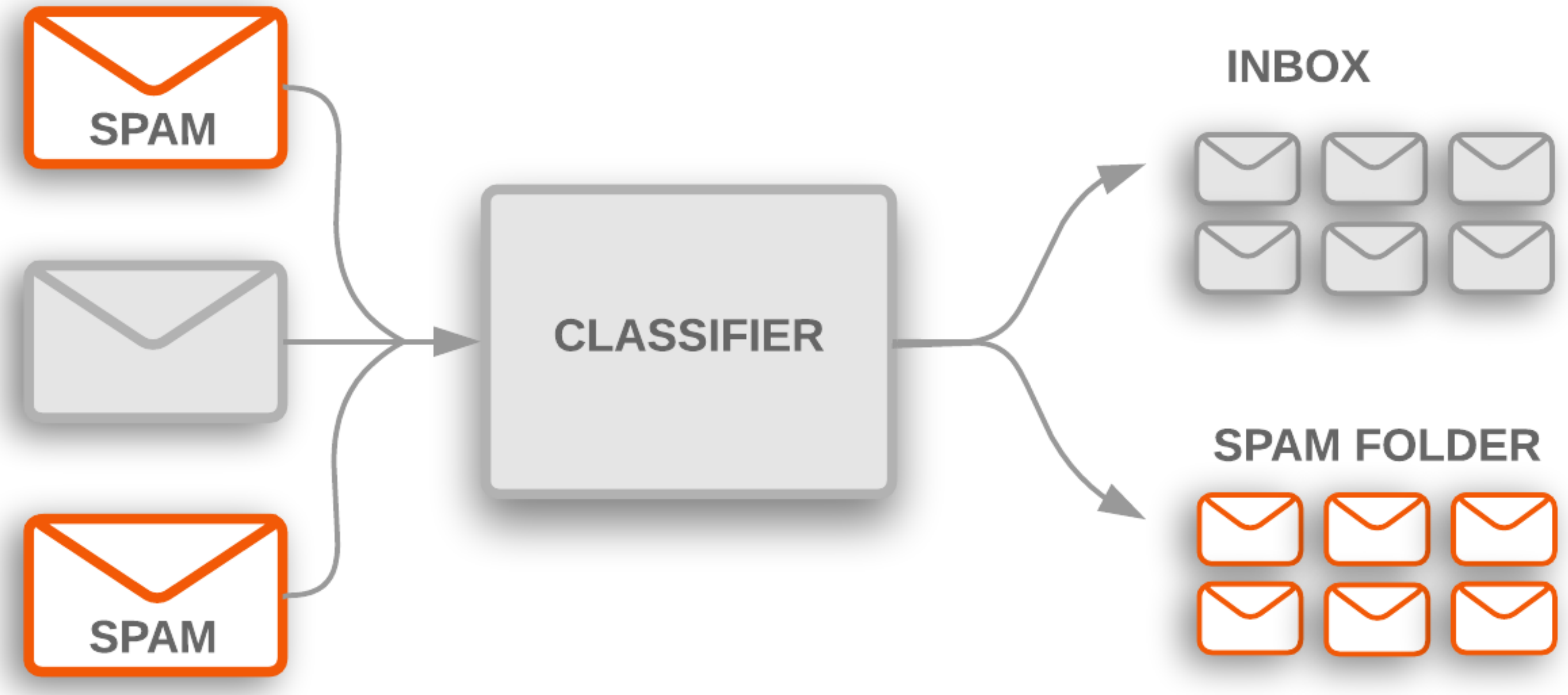
Geometric Graph Neural Network Bootcamp

Yuning You, BBE Postdoc @ Caltech

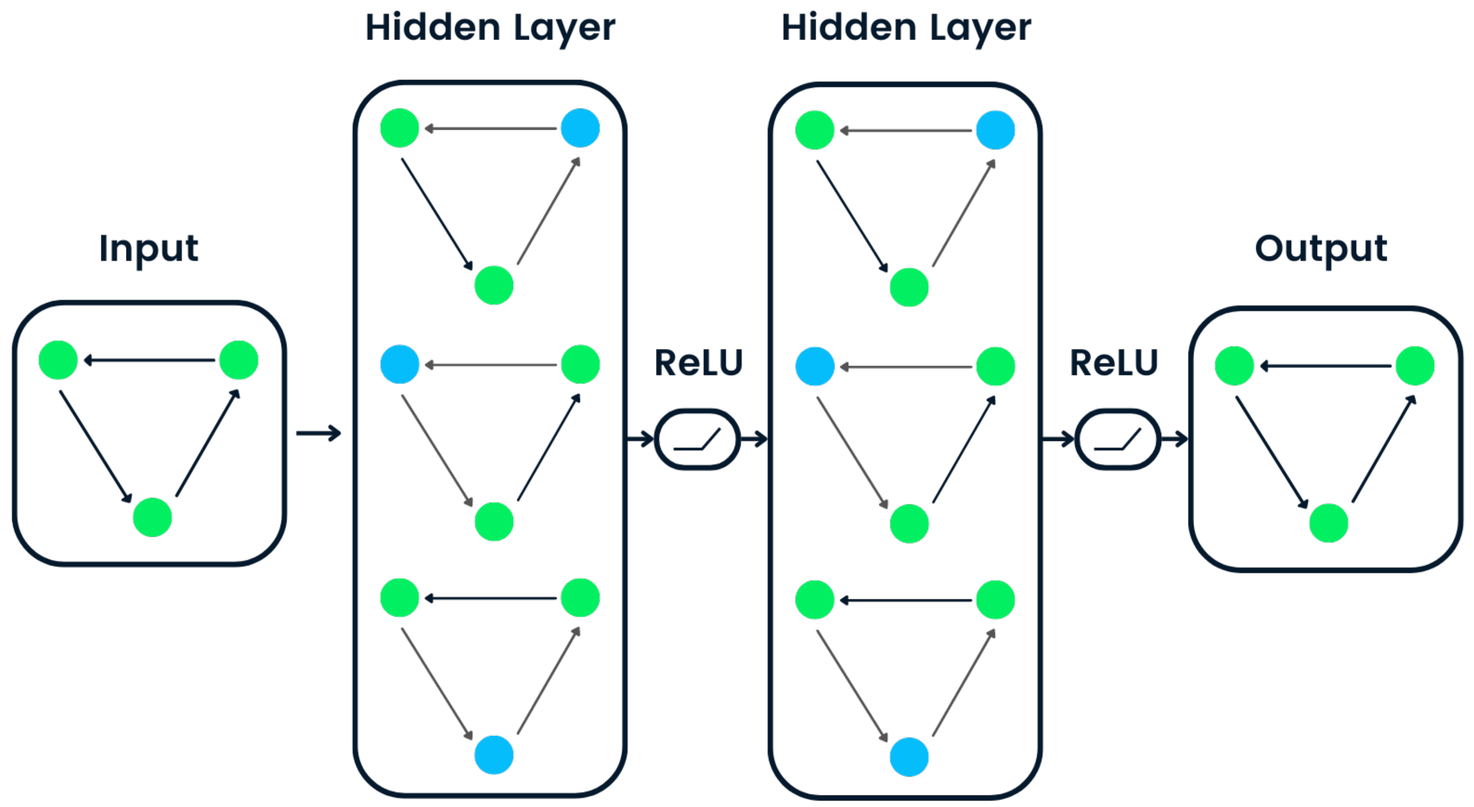
Recap: Neural Networks

- Data $X \in \mathbb{X}$, labels $Y \in \mathbb{Y}$
- Neural network $f : \mathbb{X} \to \mathbb{Y}$
  - Mappings from data to labels
  - Mappings are learned from labeled data
- Images, texts, …

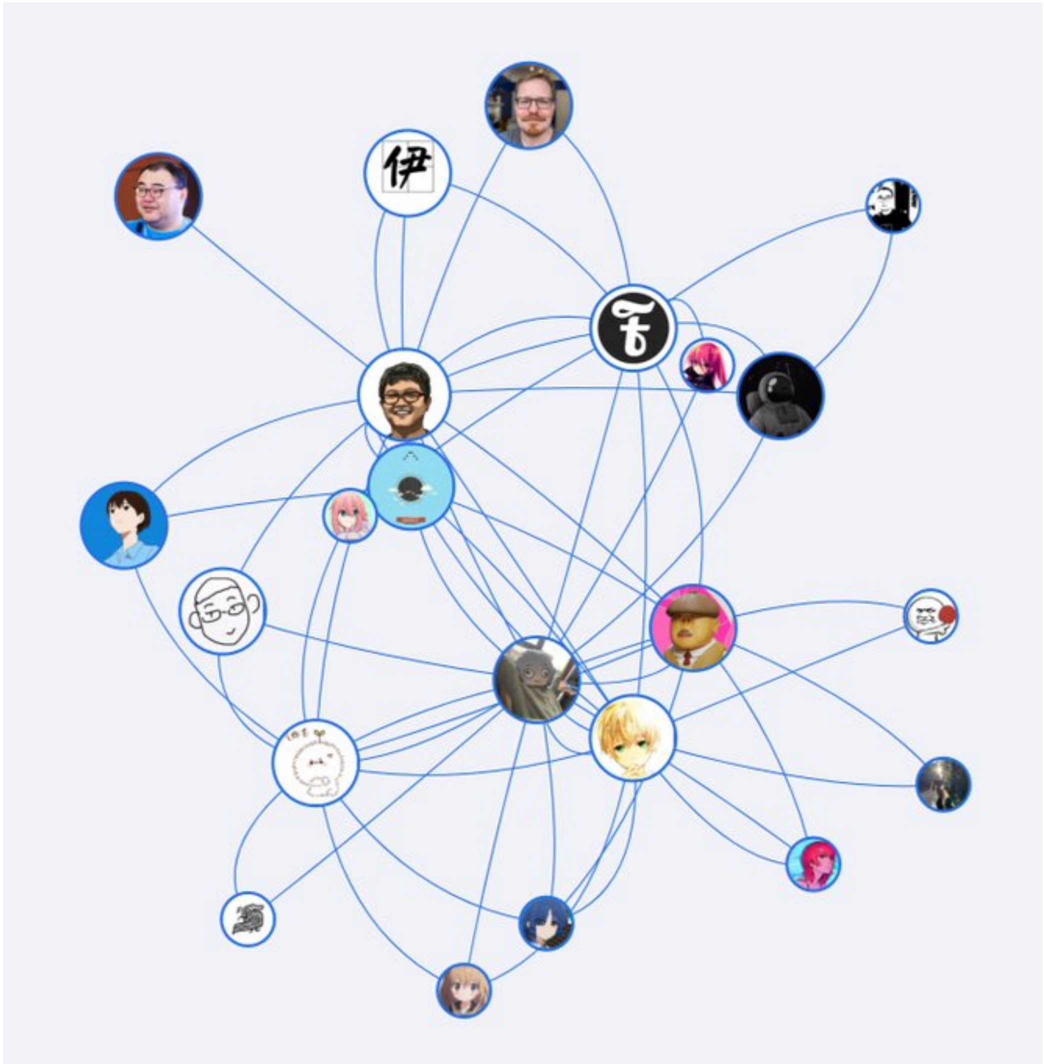


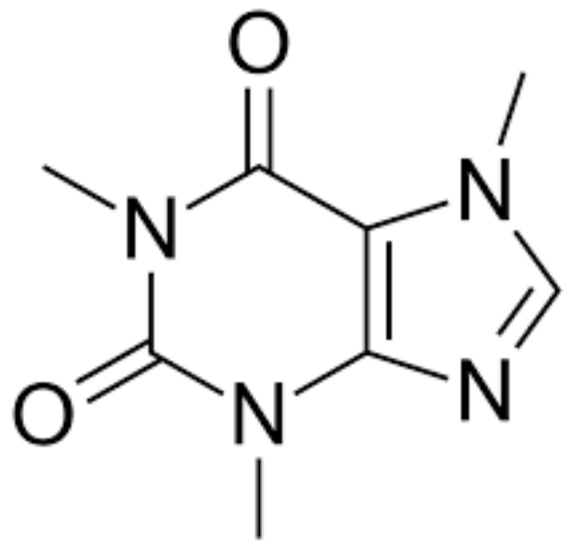


Recap: Graph Neural Networks (GNNs)



- Graph data $G \in \mathbb{G}$, labels $Y \in \mathbb{Y}$
- A graph $G = \{X, A\}$ with N nodes contains
  - Node features $X \in \mathbb{R}^{N \times D}$
  - Edge (adjacency matrix) $A \in \{0,1\}^{N \times N}$
- Labels at graph $Y \in \mathbb{Y}$ or node level $Y \in \mathbb{Y}^N$
- Graph neural network $f : \mathbb{G} \to \mathbb{Y}$
- Social networks, molecules, …

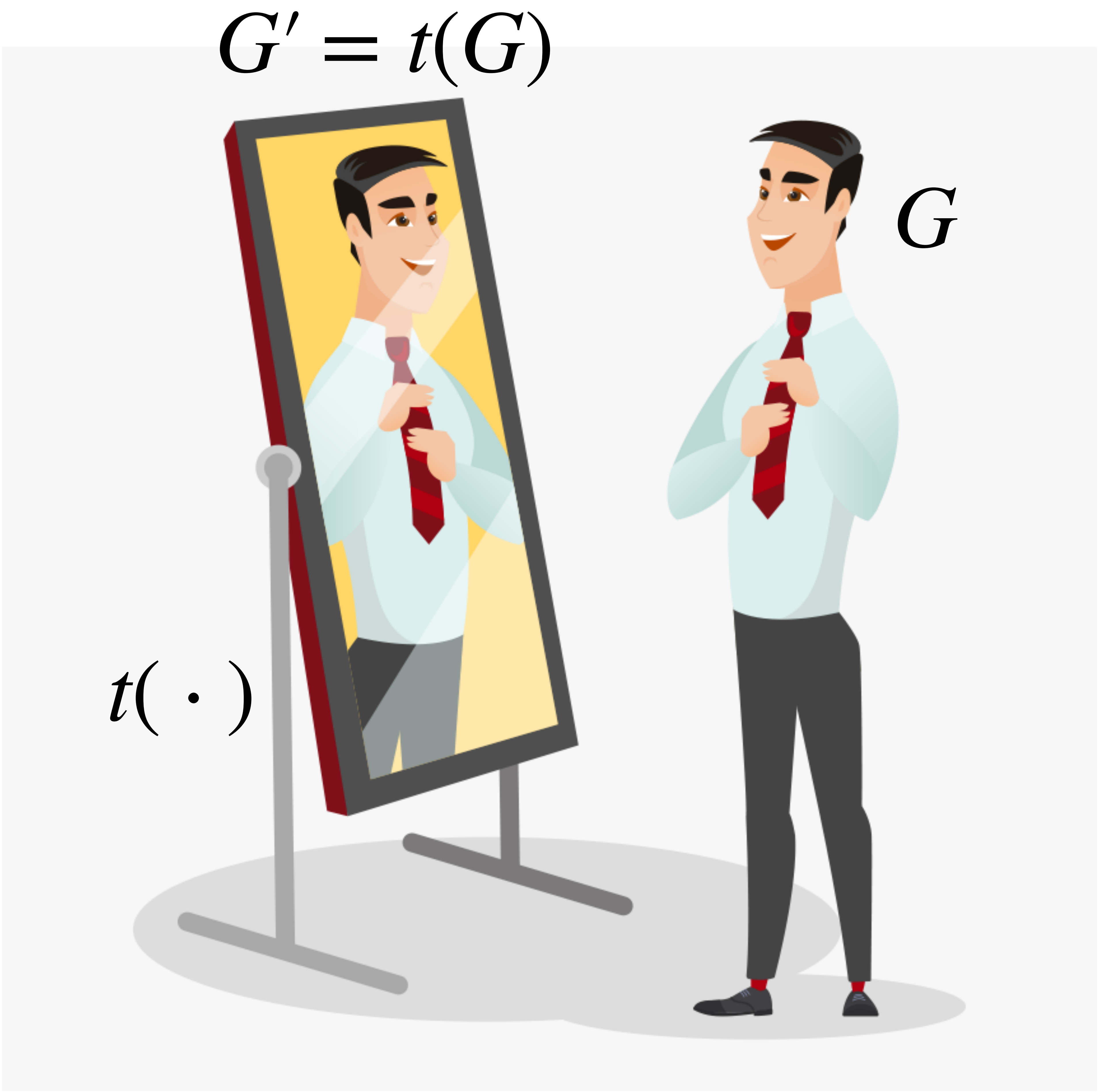




Key Feature of GNNs: Respecting Symmetry

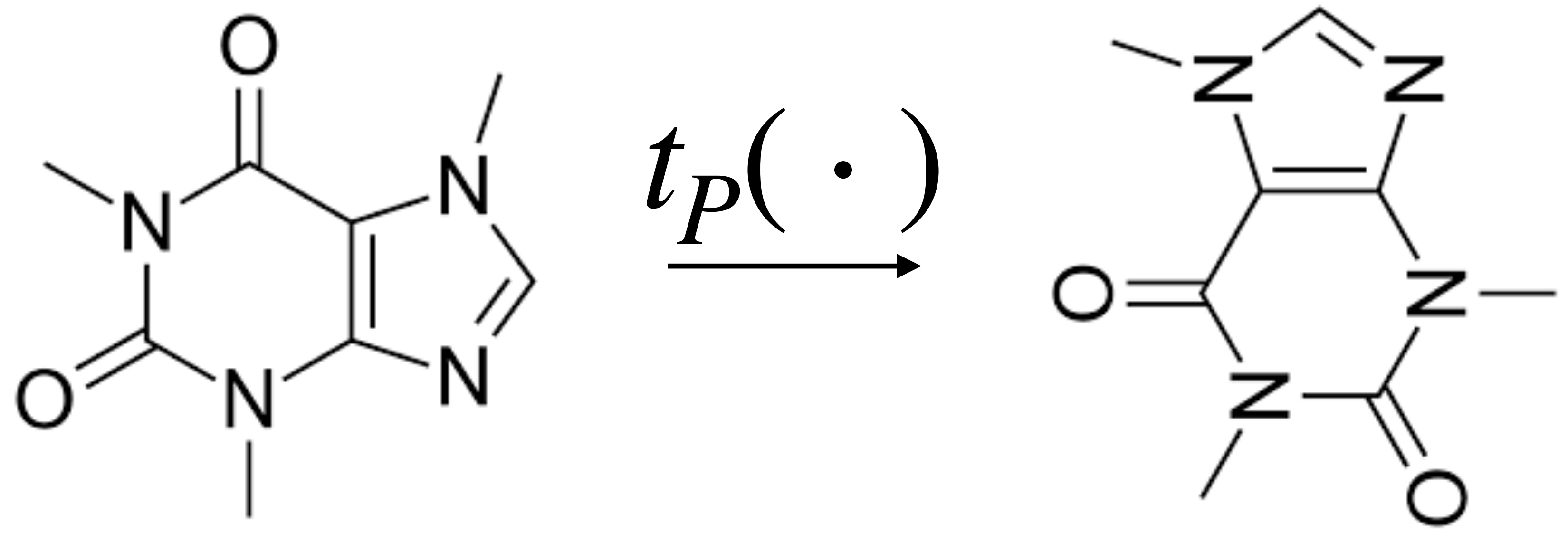
GNN model $f(\cdot)$ will say:

G and G' are "the same"



Symmetry in Graph Features: Permutation

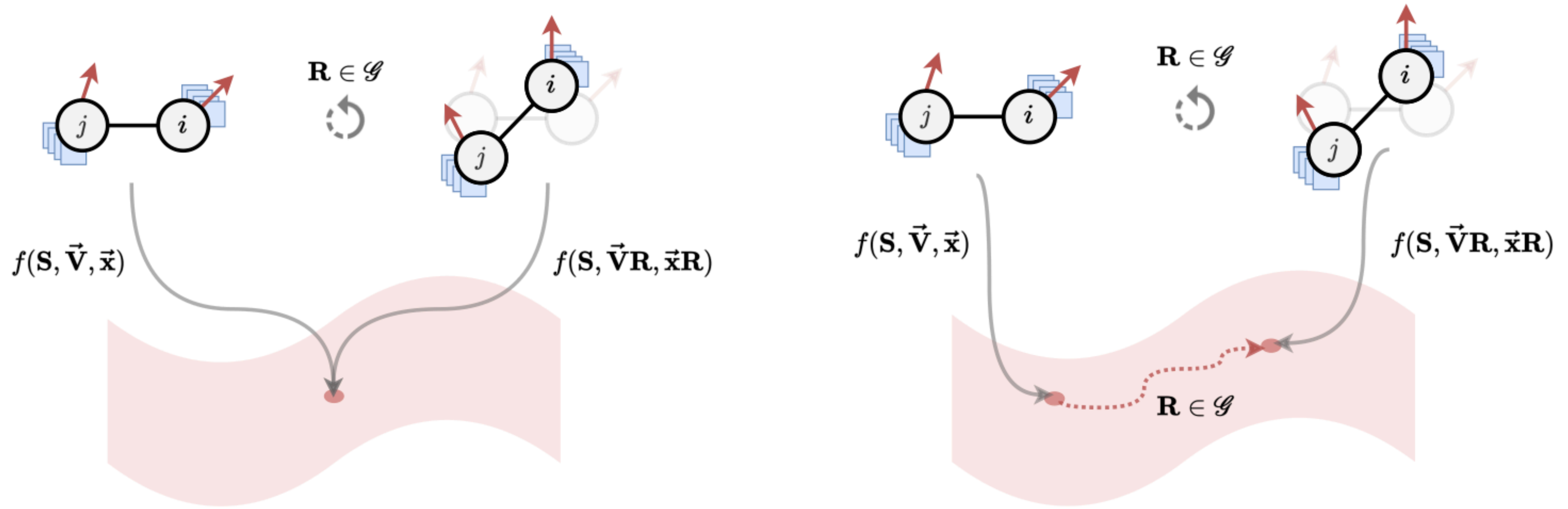
- Graph $G = \{X, A\}$
- Permutation on graph features
  - $G' = t_P(G) = \{PX, PAP^\top\}$
- Identities of G, G' are the same



GNNs Respect Symmetry in Graph Features

— Permutation-Invariance and Equivariance

- Graph neural networks $f : \mathbb{G} \to \mathbb{Y}$ (graph-level) or $f : \mathbb{G} \to \mathbb{Y}^N$ (node-level)
- Permutation-**invariance**: $f(t(G)) = f(G)$
- For $Y \in \mathbb{Y}^N$, suppose $t_P(Y) = PY$
- Permutation-**equivariance**: $f(t(G)) = t(f(G))$

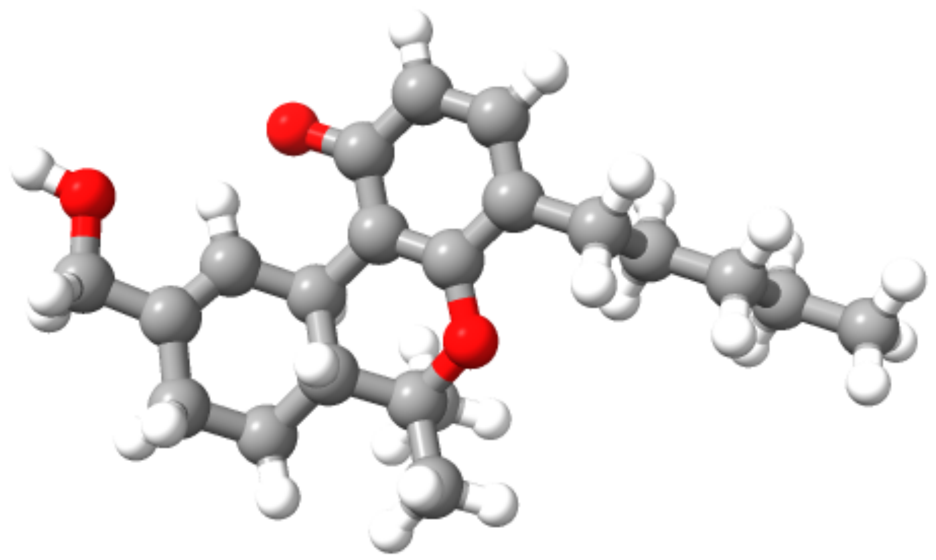
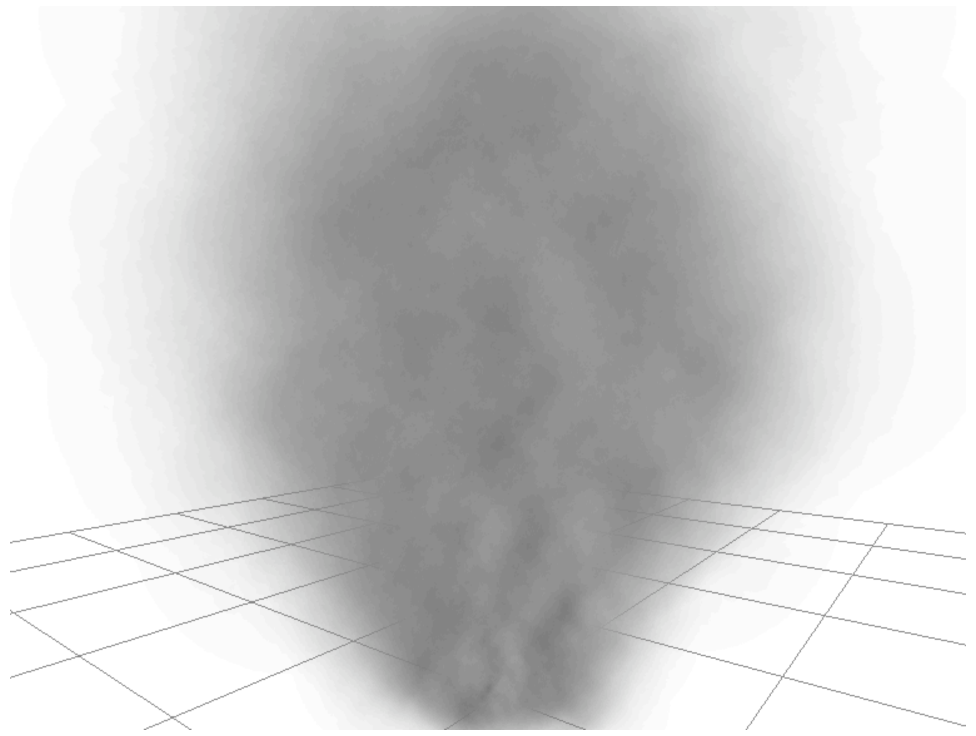
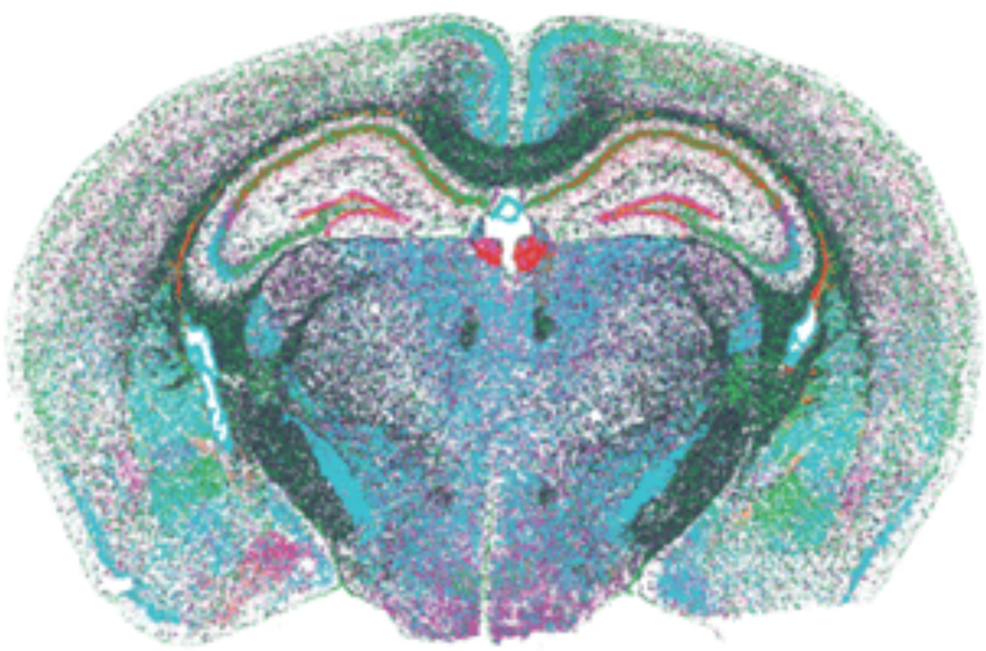


Duval et al., 2024

Geometric Graphs are Everywhere in Science

or Termed Spatial Graphs, etc.

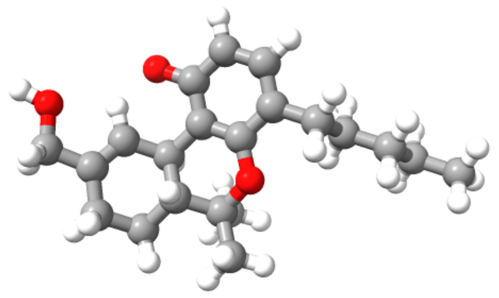
- Graph $G = \{X, A\}$
- Geometric graph $G = \{X \oplus C, A\}$ or $G = \{X, A, C\}$
    - Additional feature: coordinate feature $C \in \mathbb{R}^{N\times 3}$
- 3D molecules, physical systems, spatial genomics…

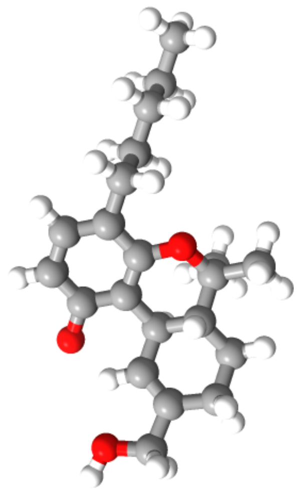
New Symmetry in Geometric Graph Features

— Global Translation and Rotation (beyond Permutation)

- Permutation on $G = \{X, A, C\}$
  - $G' = t_P(G) = \{PX, PAP^\top, PC\}$
- Plus global translation and rotation
  - $G' = t_{I,T,R}(G) = \{X, A, CR + T\}$
- Identities of G, G' are the same

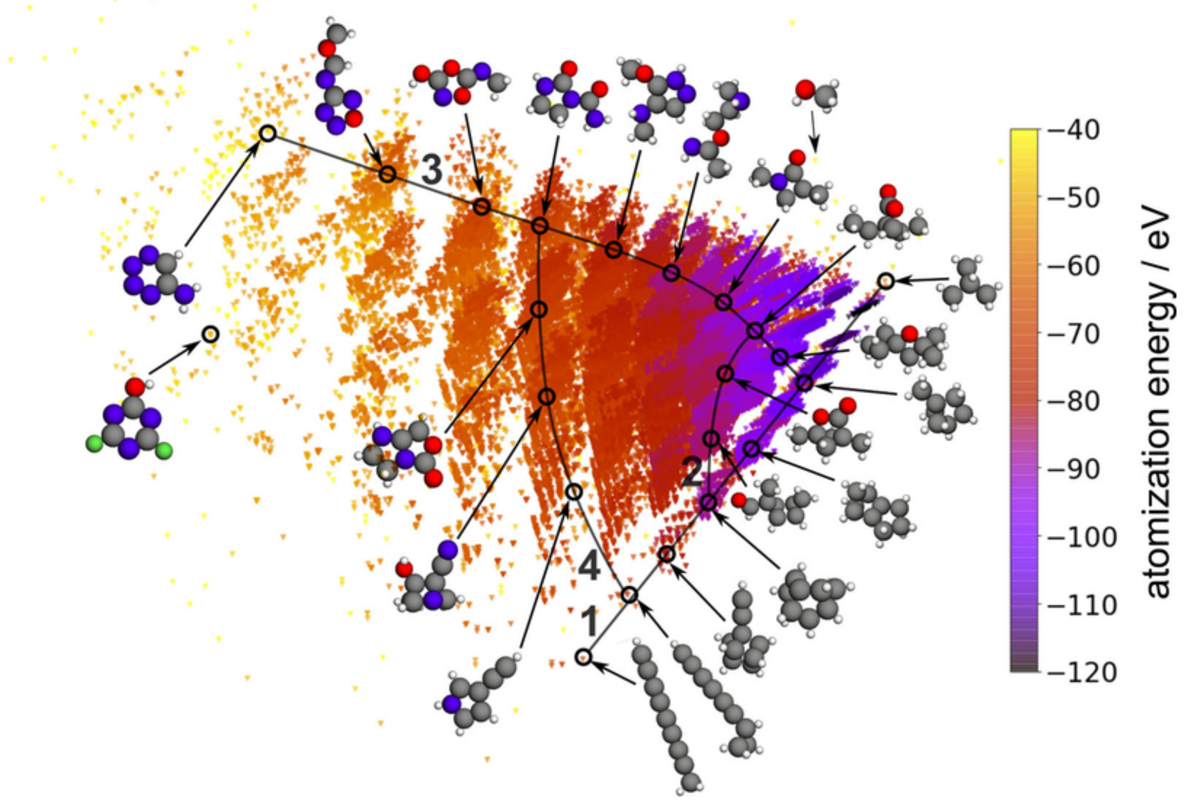


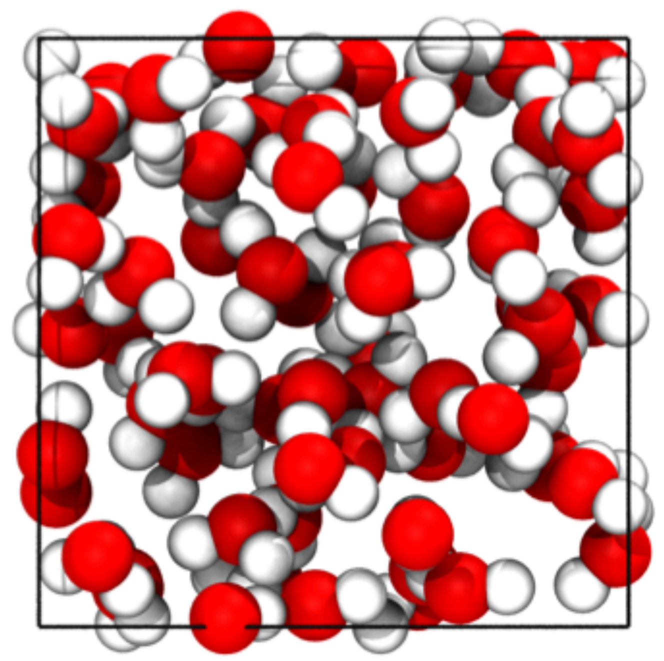
$\xrightarrow{t_{P,T,R}(\,\cdot\,)}$

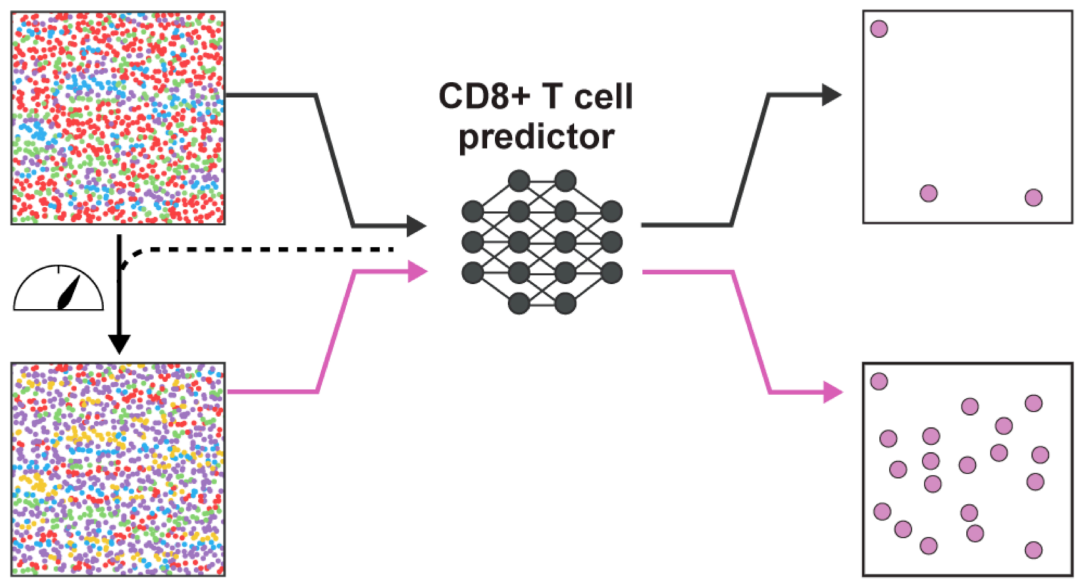


Geometric Graph Neural Networks

- Geometric graph $G = \{X, A, C\}$
- Geometric GNN $f : \mathbb{G} \rightarrow \mathbb{Y}$ (graph-level) or $f : \mathbb{G} \rightarrow \mathbb{Y}^N$ (node-level)
- Example tasks
    - Molecular property prediction
    - Force field simulation
    - Cellular microenvironment analysis
    - …



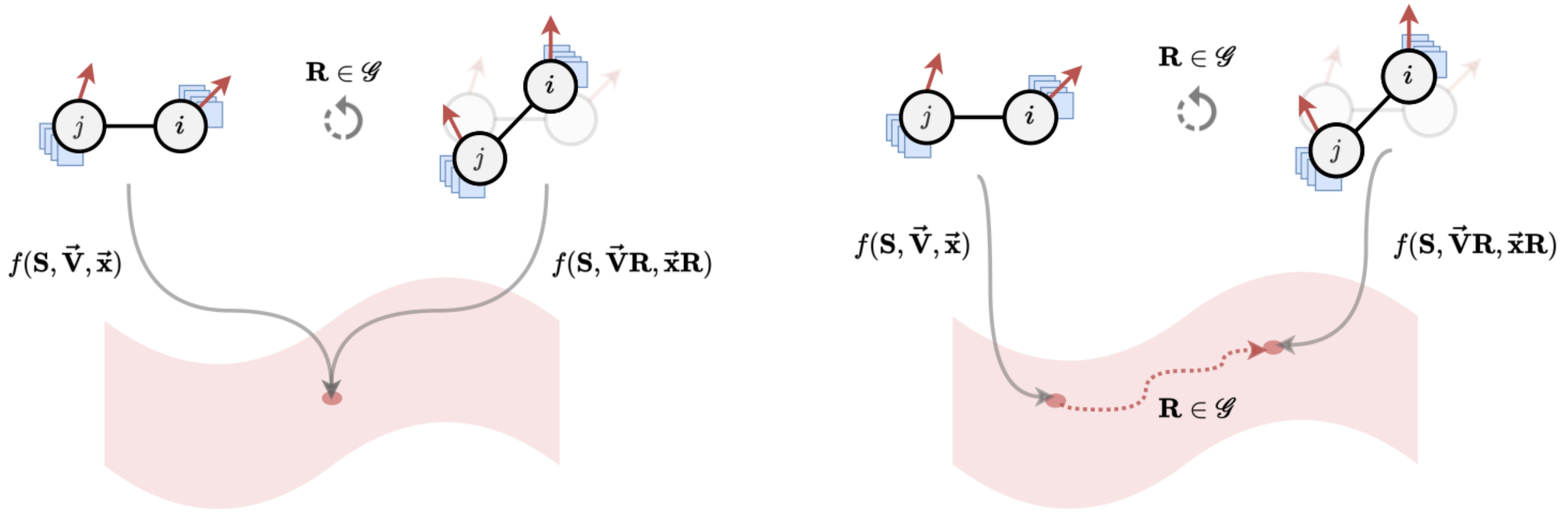




Geometric GNNs Respect Geometric Symmetry

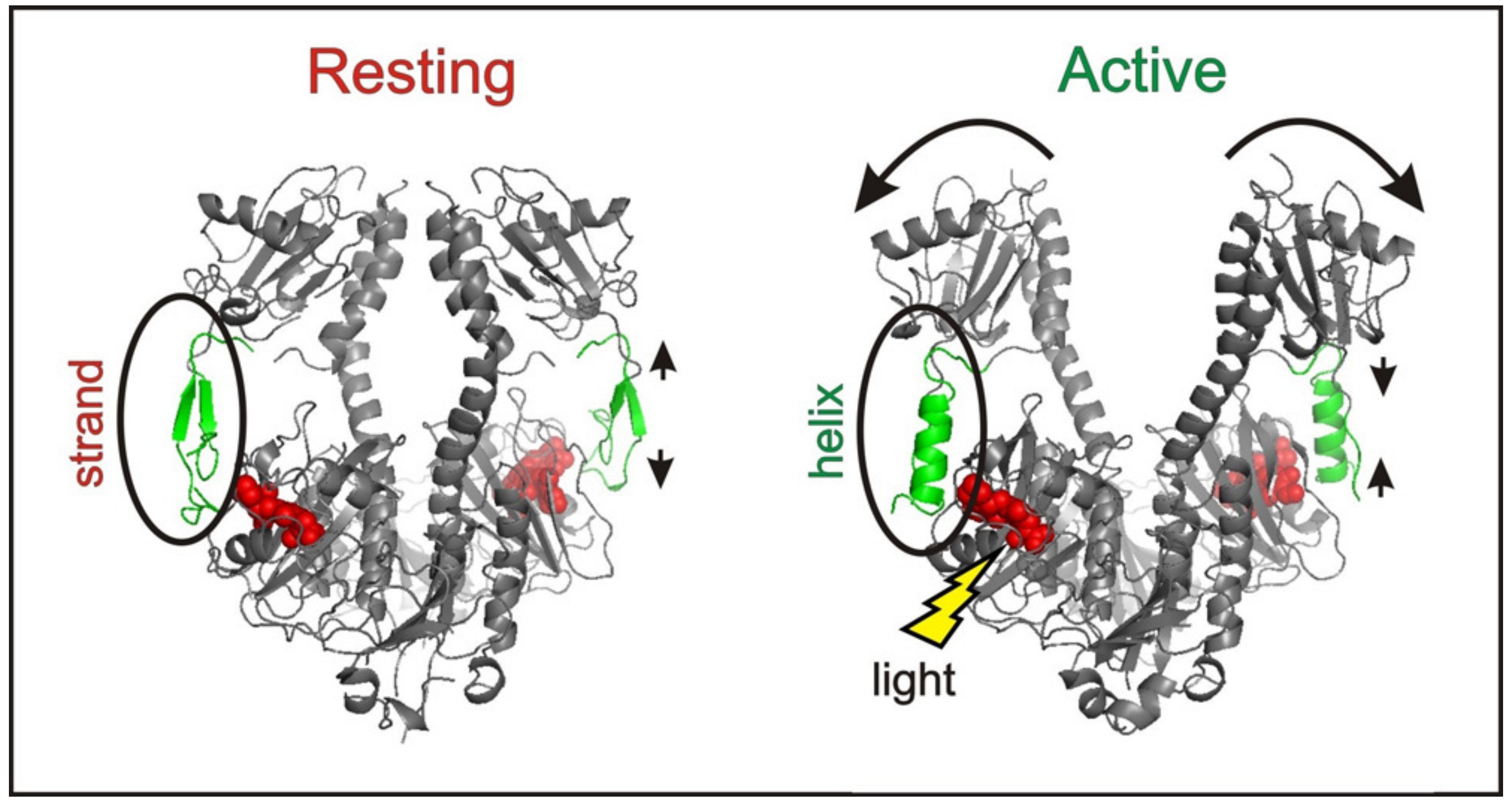
— Translation/Rotation-Invariance and Equivariance

- Geometric GNN $f : \mathbb{G} \to \mathbb{Y}$ (graph-level) or $f : \mathbb{G} \to \mathbb{Y}^N$ (node-level)
- Translation/Rotation-**invariance**
  - $f(\{X, A, CR + T\}) = f(\{X, A, C\})$
- Suppose $t_{I,T,R}(Y) = YR + T$
- Translation/Rotation-**equivariance**
  - $f(\{X, A, CR + T\}) = f(\{X, A, C\})R + T$



An Example: Predicting Conformational Change of Biomolecules

- Input: Biomolecule featurized as geometric graph $G = \{X, A, C\}$
  - X is customized feature for atom/amino-acid, C is coordinate
  - A can be constructed based on spatial proximity or chemical rules
- Output/label: Evolved (node-level) coordinates $Y \in \mathbb{R}^{N \times 3}$
- Geometric GNNs need to be equivariant that
  - $f(\{X, A, CR + T\}) = f(\{X, A, C\})R + T$

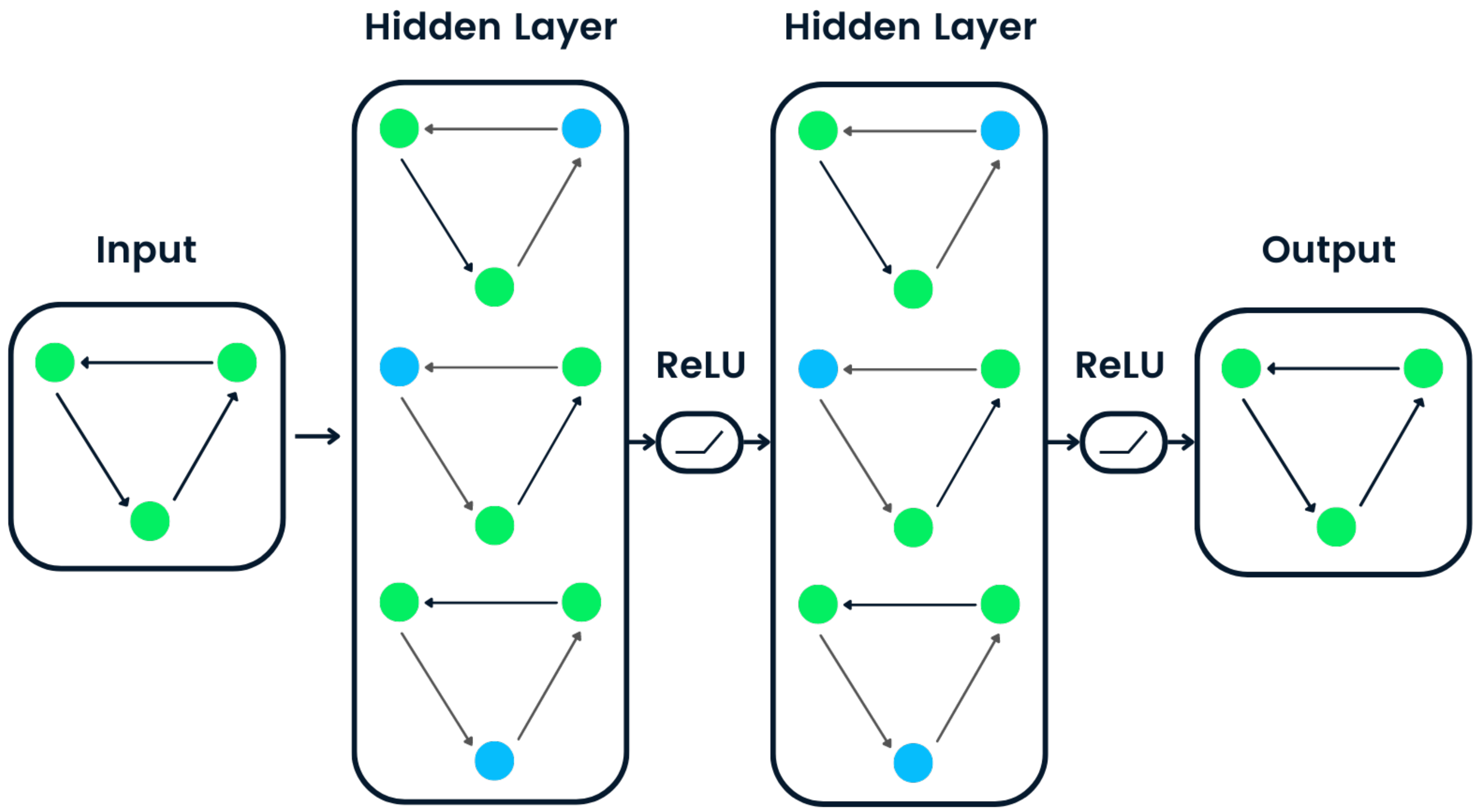


A Specific Architecture: Equivariant Graph Neural Networks

E(n) Equivariant Graph Neural Networks

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

- Graph neural networks
  - Message passing across connected nodes



$$
\begin{aligned}
\mathbf{m}_{ij} &= \phi_e\left(\mathbf{h}_i^l, \mathbf{h}_j^l, a_{ij}\right) \\
\mathbf{m}_i &= \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij} \\
\mathbf{h}_i^{l+1} &= \phi_h\left(\mathbf{h}_i^l, \mathbf{m}_i\right)
\end{aligned}
\qquad (2)
$$

A Specific Architecture: Equivariant Graph Neural Networks

E(n) Equivariant Graph Neural Networks

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

- Equivariant graph neural networks
  - Message passing across connected nodes, **plus considering geometry**

$$\mathbf{m}_{ij} = \phi_e\left(\mathbf{h}_i^l, \mathbf{h}_j^l, \left\|\mathbf{x}_i^l - \mathbf{x}_j^l\right\|^2, a_{ij}\right) \tag{3}$$

$$\mathbf{x}_i^{l+1} = \mathbf{x}_i^l + C\sum_{j\neq i}\left(\mathbf{x}_i^l - \mathbf{x}_j^l\right)\phi_x\left(\mathbf{m}_{ij}\right) \tag{4}$$

$$\mathbf{m}_i = \sum_{j\in\mathcal{N}(i)}\mathbf{m}_{ij} \tag{5}$$

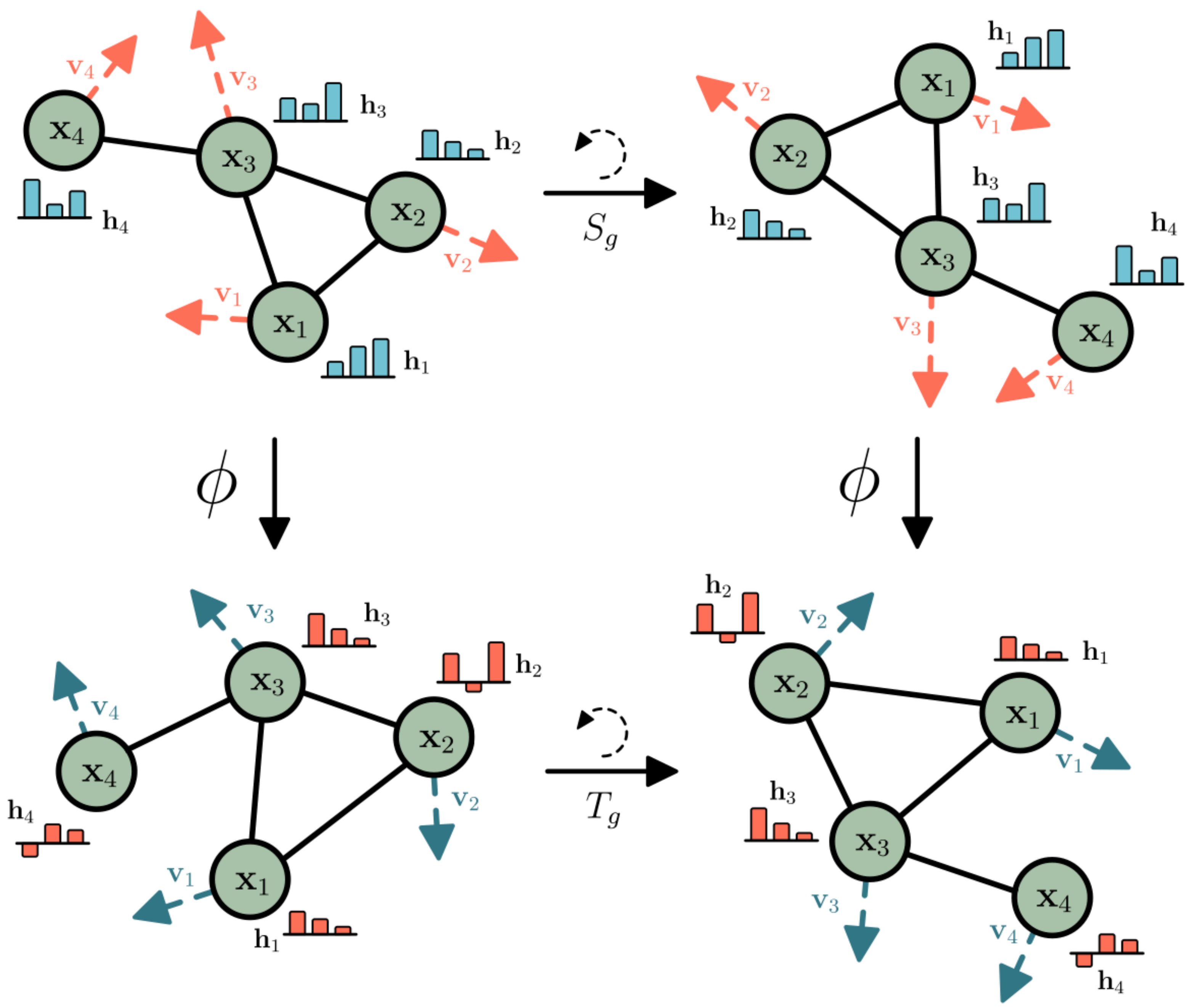
$$\mathbf{h}_i^{l+1} = \phi_h\left(\mathbf{h}_i^l, \mathbf{m}_i\right) \tag{6}$$

A Specific Architecture: Equivariant Graph Neural Networks

E(n) Equivariant Graph Neural Networks

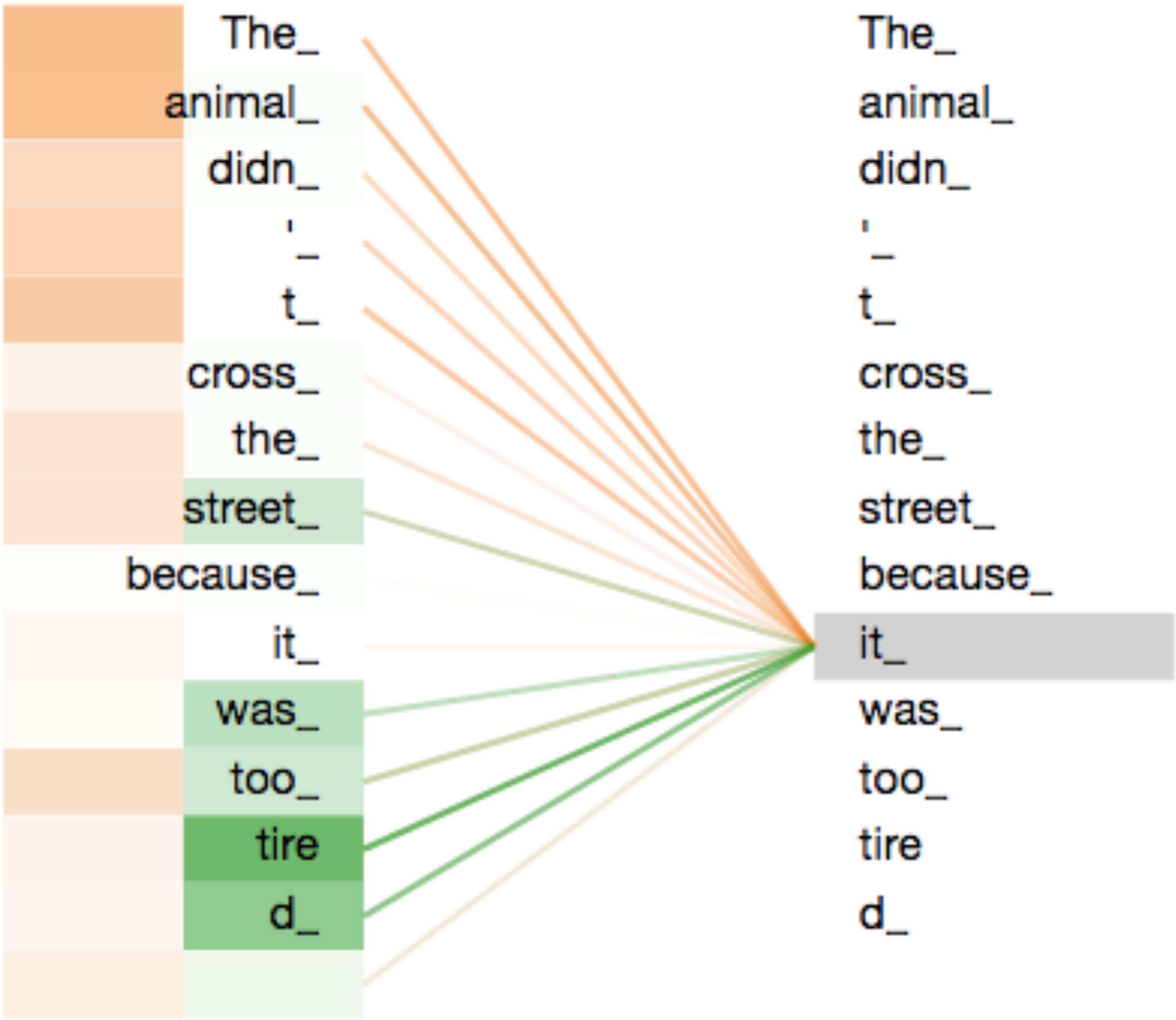
Victor Garcia Satorras [1] Emiel Hoogeboom [1] Max Welling [1]

- Equivariant graph neural networks are equivariant to translation, rotation, and reflection



Another Specific Architecture: SE(3)-Transformer

- Transformers
    - Message passing on fully-connected graphs



SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks

Fabian B. Fuchs[*†]
Bosch Center for Artificial Intelligence
A2I Lab, Oxford University
fabian@robots.ox.ac.uk

Daniel E. Worrall[*]
Amsterdam Machine Learning Lab, Philips Lab
University of Amsterdam
d.e.worrall@uva.nl

Volker Fischer
Bosch Center for Artificial Intelligence
volker.fischer@de.bosch.com

Max Welling
Amsterdam Machine Learning Lab
University of Amsterdam
m.welling@uva.nl

Another Specific Architecture: SE(3)-Transformer

- SE(3)-Transformers
  - Message passing on fully-connected graphs
  - **Plus considering geometry**

SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks

Fabian B. Fuchs[*†]
Bosch Center for Artificial Intelligence
A2I Lab, Oxford University
fabian@robots.ox.ac.uk

Daniel E. Worrall[*]
Amsterdam Machine Learning Lab, Philips Lab
University of Amsterdam
d.e.worrall@uva.nl

Volker Fischer
Bosch Center for Artificial Intelligence
volker.fischer@de.bosch.com

Max Welling
Amsterdam Machine Learning Lab
University of Amsterdam
m.welling@uva.nl

$$\mathbf{f}^{\ell}_{\text{out},i} = \underbrace{\mathbf{W}^{\ell\ell}_{V}\mathbf{f}^{\ell}_{\text{in},i}}_{③\text{ self-interaction}} + \sum_{k\geq 0}\sum_{j\in\mathcal{N}_i\setminus i} \underbrace{\alpha_{ij}}_{①\text{ attention}} \underbrace{\mathbf{W}^{\ell k}_{V}(\mathbf{x}_j - \mathbf{x}_i)\mathbf{f}^{k}_{\text{in},j}}_{②\text{ value message}}. \tag{10}$$

$$\alpha_{ij} = \frac{\exp(\mathbf{q}_i^\top \mathbf{k}_{ij})}{\sum_{j'\in\mathcal{N}_i\setminus i}\exp(\mathbf{q}_i^\top \mathbf{k}_{ij'})}, \quad \mathbf{q}_i = \bigoplus_{\ell\geq 0}\sum_{k\geq 0}\mathbf{W}^{\ell k}_{Q}\mathbf{f}^{k}_{\text{in},i}, \quad \mathbf{k}_{ij} = \bigoplus_{\ell\geq 0}\sum_{k\geq 0}\mathbf{W}^{\ell k}_{K}(\mathbf{x}_j - \mathbf{x}_i)\mathbf{f}^{k}_{\text{in},j}. \tag{11}$$

Another Specific Architecture: SE(3)-Transformer

- SE(3)-Transformers are equivariant to
- translation and rotation

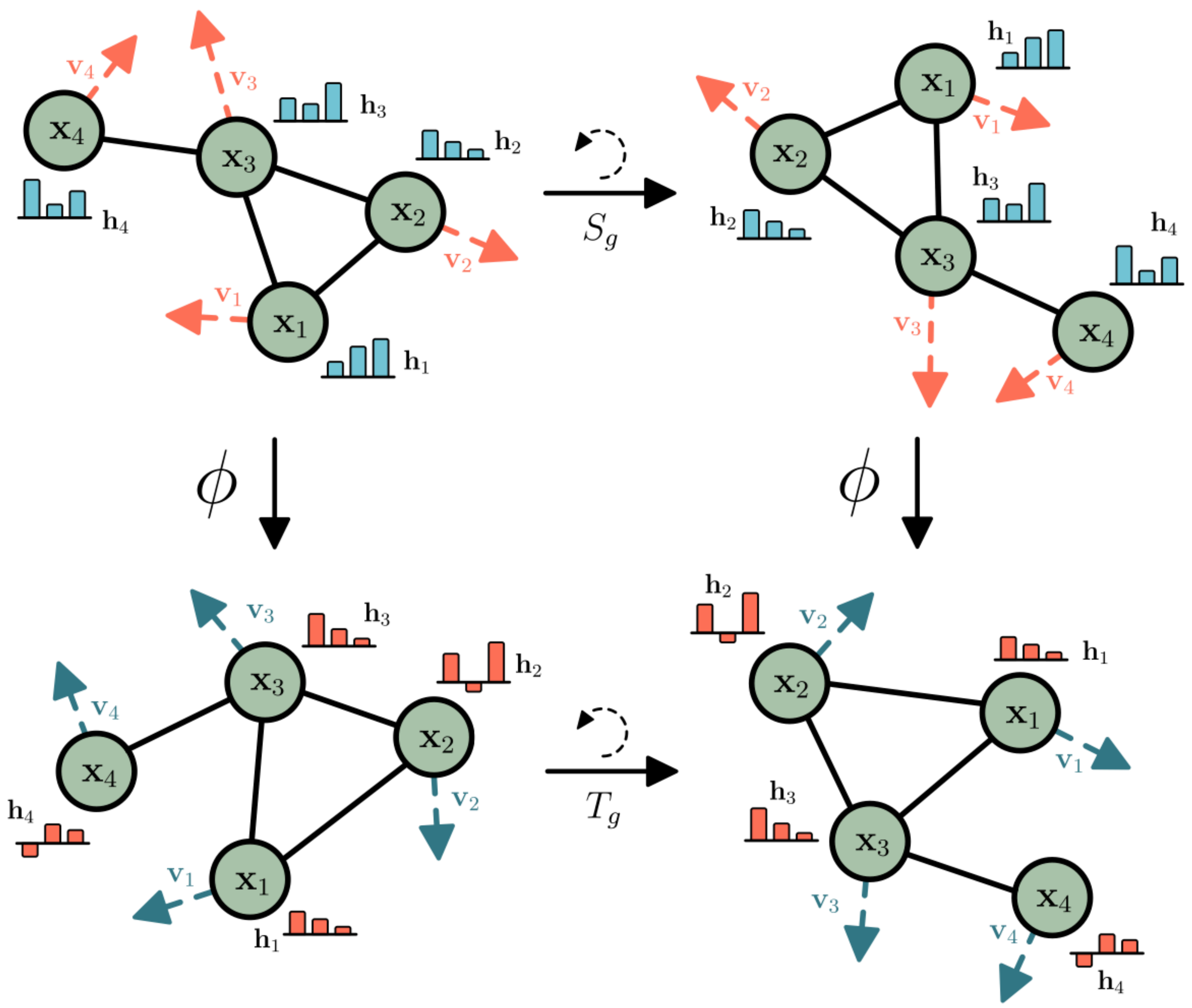
SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks

Fabian B. Fuchs[*†]
Bosch Center for Artificial Intelligence
A2I Lab, Oxford University
`fabian@robots.ox.ac.uk`

Daniel E. Worrall[*]
Amsterdam Machine Learning Lab, Philips Lab
University of Amsterdam
`d.e.worrall@uva.nl`

Volker Fischer
Bosch Center for Artificial Intelligence
`volker.fischer@de.bosch.com`

Max Welling
Amsterdam Machine Learning Lab
University of Amsterdam
`m.welling@uva.nl`



Language of Group Theory to Describe NN Symmetry

- The transformations w.r.t. symmetry can be more conveniently described in the language of group theory
  - Permutations are elements of the **permutation group**
  - Translation and rotations (e.g. in 3D) are elements of the special Euclidean group SE(3)
  - SE(3)-equivariance: The model is equivariant to all elements in SE(3)
    - $f(\{X, A, CR + T\}) = f(\{X, A, C\})R + T$

Language of Group Theory to Describe NN Symmetry

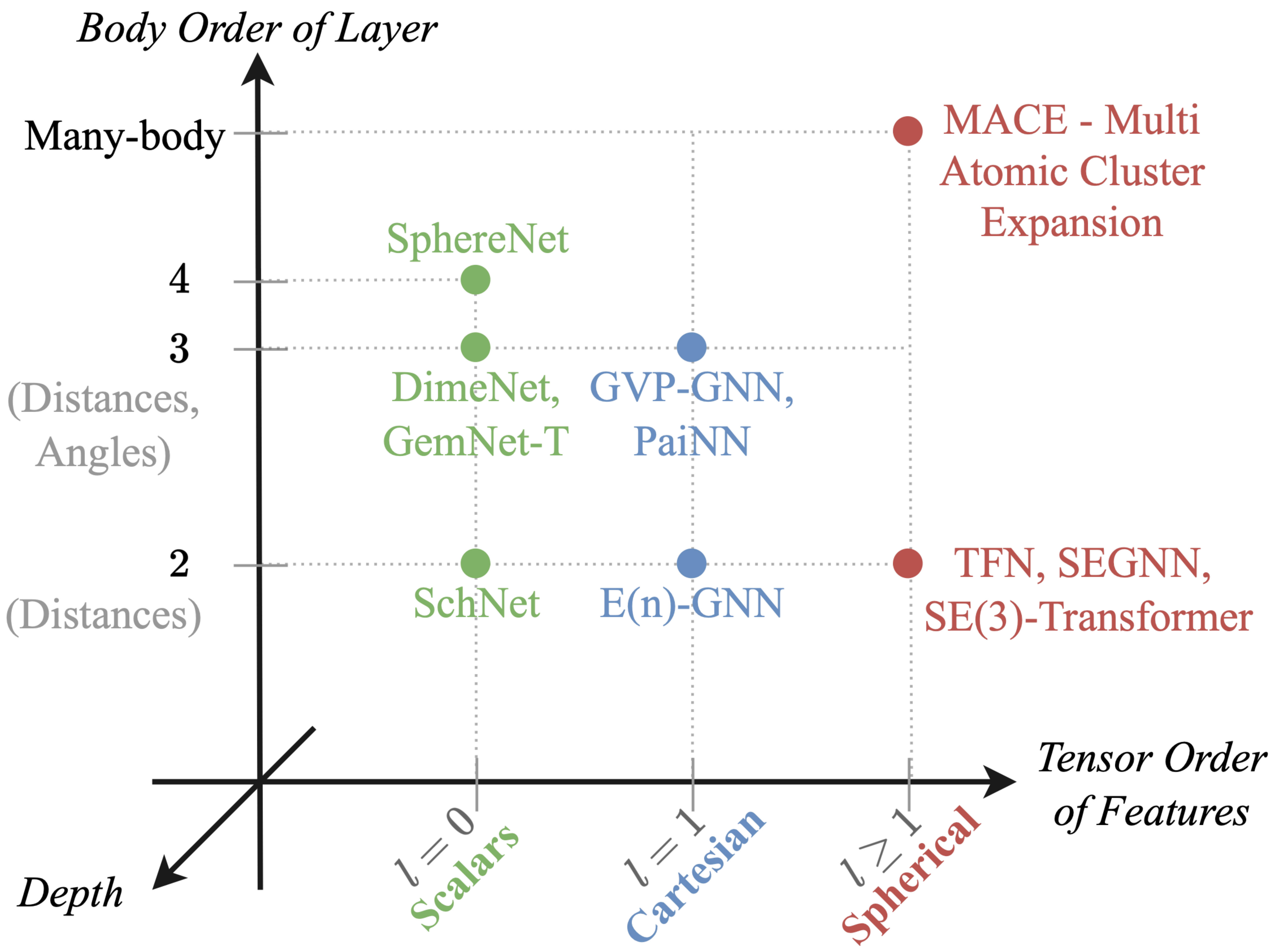
- A group is composed of a set $\mathcal{S}$ and a binary operator $* : \mathcal{S} \times \mathcal{S} \rightarrow \mathcal{S}$ that
  - Closure: $\forall x, y \in \mathcal{S}, x * y \in \mathcal{S}$ (perm after perm is still perm)
  - Associativity: $\forall x, y, z \in \mathcal{S}, x * (y * z) = (x * y) * z$
  - Identity: $\exists e, \forall x \in \mathcal{S}, e * x = x * e = x$
  - Inverse: $\forall x, \exists y, x * y = y * x = e$
- SE(3)-equivariance: The model is equivariant to all elements in SE(3)
  - $f(\{X, A, CR + T\}) = f(\{X, A, C\})R + T$

Invariant/Equivariant Model can be Constructed Layer-by-Layer

- Concatenation of equivariant layers is still equivariant
- $f\Big(\{X, A, f(\{X, A, CR + T\})\Big) = f\Big(\{X, A, f(\{X, A, C\})\Big) R + T$
- Concatenation of equivariant layers and a final invariant layer is invariant

$$
\begin{tikzcd}
\mathcal{F}_0 \arrow[r, "L_1"] \arrow[d, "g \triangleright_0"'] & \mathcal{F}_1 \arrow[r, "L_2"] & \mathcal{F}_2 \arrow[r, "L_3"] & \cdots \arrow[r, "L_{N-1}"] & \mathcal{F}_{N-1} \arrow[r, "L_N"] & \mathcal{F}_N \arrow[d, "g \triangleright_N"'] \\
\mathcal{F}_0 \arrow[r, "L_1"'] & \mathcal{F}_1 \arrow[r, "L_2"'] & \mathcal{F}_2 \arrow[r, "L_3"'] & \cdots \arrow[r, "L_{N-1}"'] & \mathcal{F}_{N-1} \arrow[r, "L_N"'] & \mathcal{F}_N
\end{tikzcd}
$$

There are a lot of Geometric GNNs so far…



Joshi et al., 2024

Thank You