UNIVERSITY OF CAMBRIDGE

On the Expressive Power of Geometric Graph Neural Networks

Chaitanya K. Joshi*, Cristian Bodnar*, Simon V. Mathis,
Taco Cohen, and Pietro Liò

NeurIPS 2022 Workshop on Symmetry and Geometry — Oral presentation

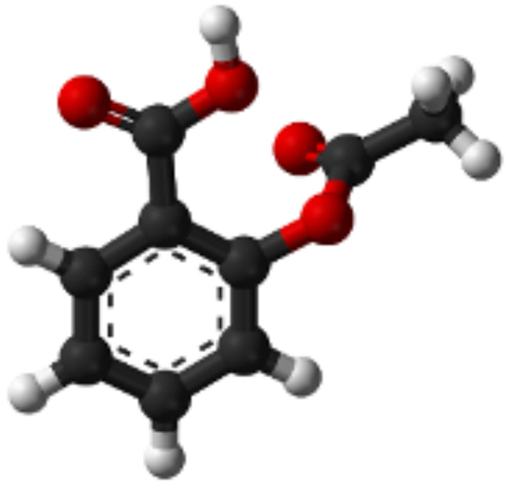
PDF: arxiv.org/abs/2301.09308

Code: github.com/chaitjo/geometric-gnn-dojo

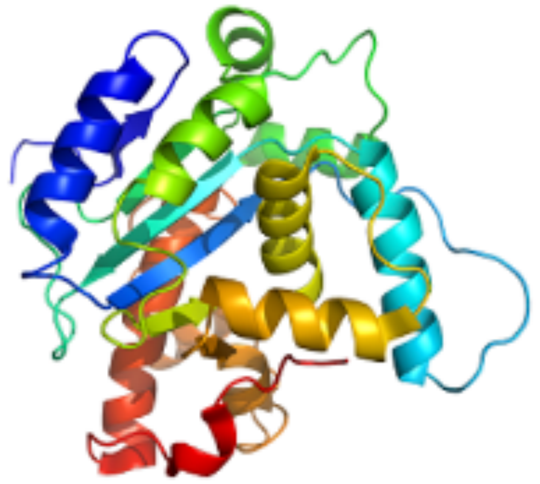
Video: youtu.be/5ulJMtpiKGc

* Equal first authors

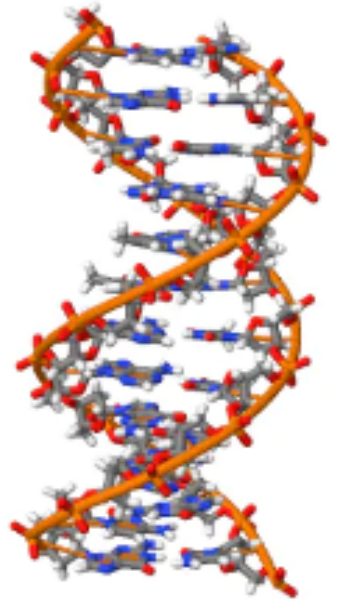
Systems with geometric & relational structure



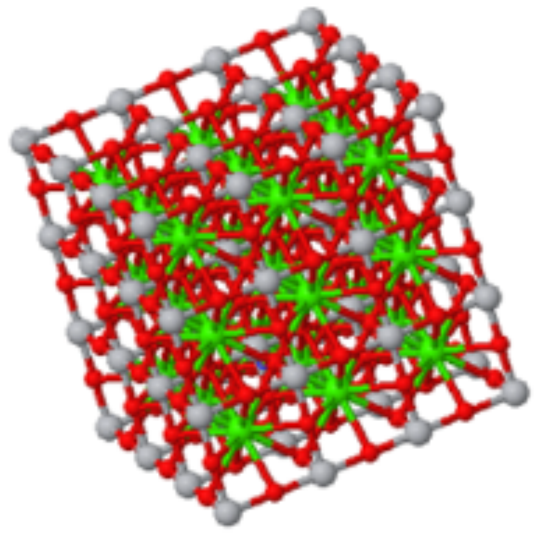
Small Molecules



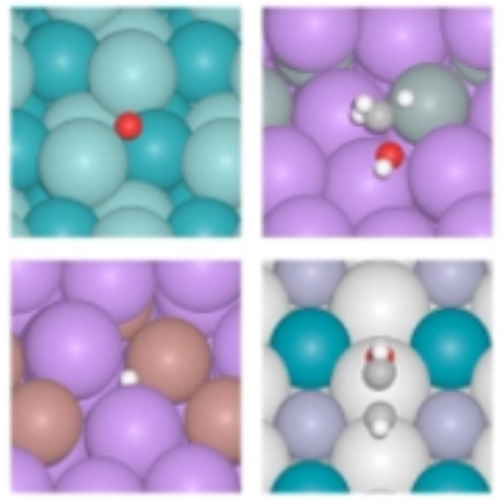
Proteins



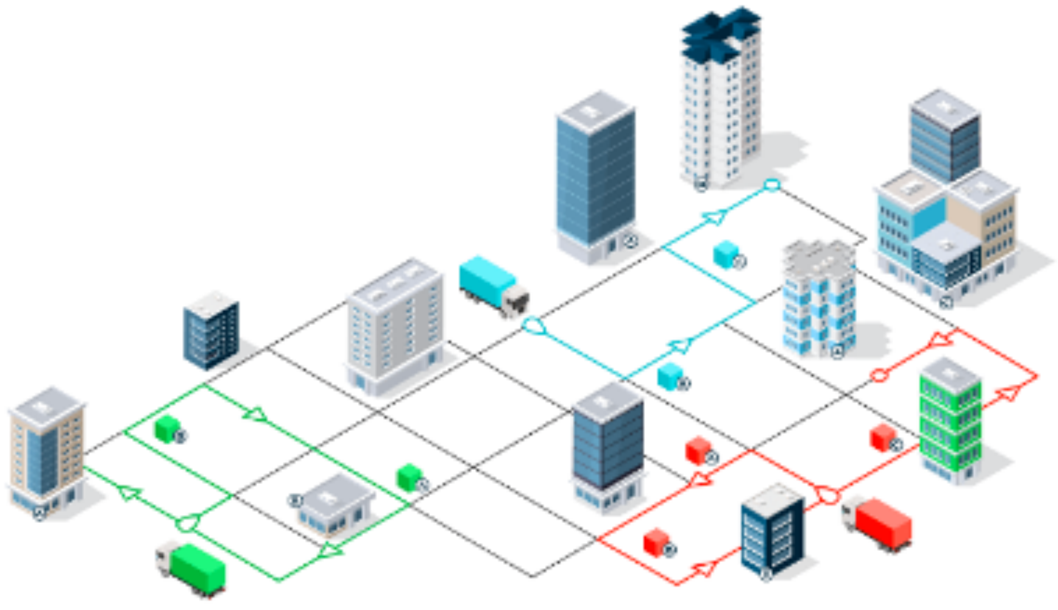
DNA/RNA



Inorganic Crystals

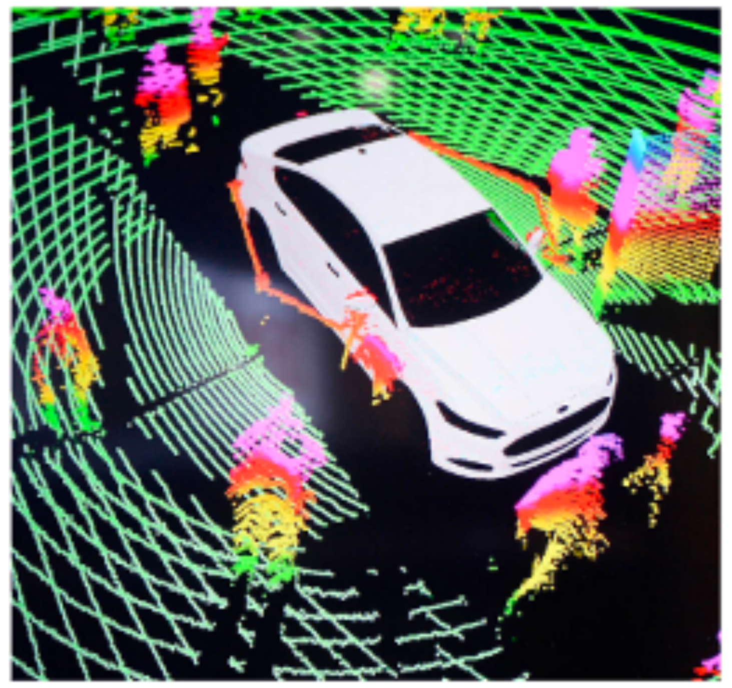


Catalysis Systems



Transportation & Logistics

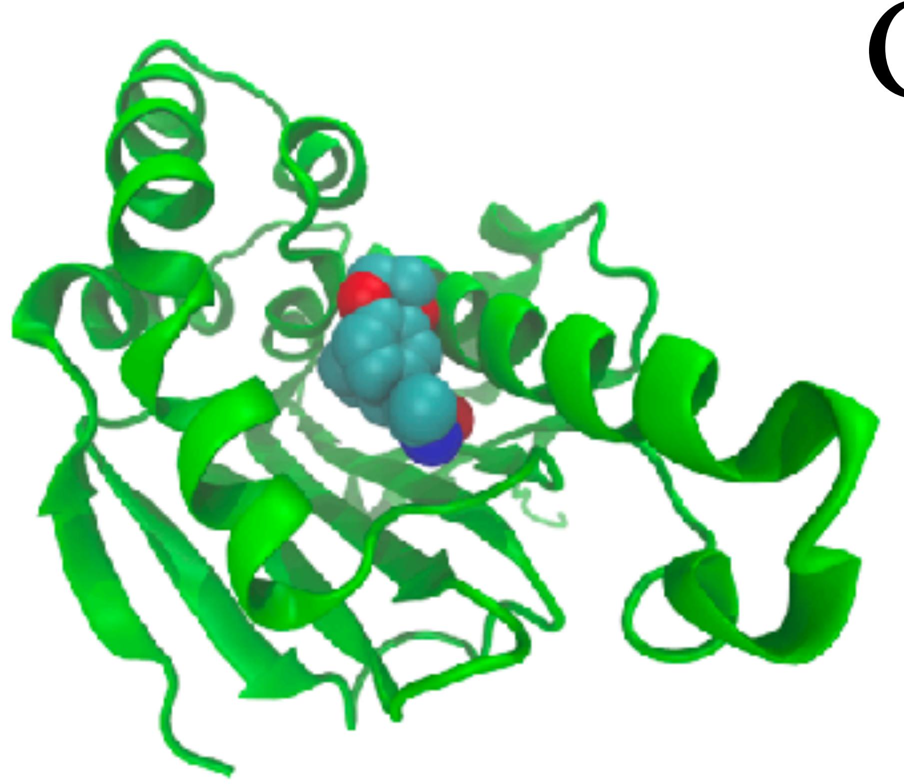
Robotic Navigation



3D Computer Vision

Geometric Graph Neural Networks

Fundamental tool for machine learning on geometric graphs



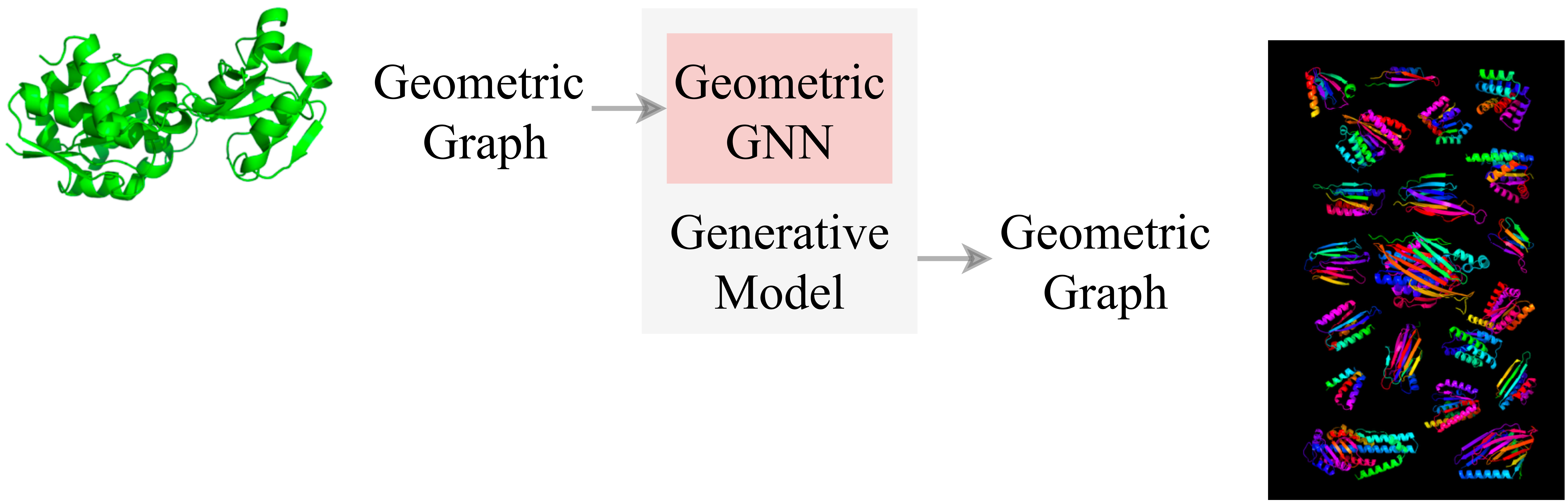
Geometric Graph → Geometric GNN → Prediction

- Functional properties?
- Ligand binding affinity?
- Ligand efficacy?

[1] Gligorijević et al., DeepFRI, Nature Communications, 2021. [2] Zhang et al., GearNet, ICLR, 2023. [3] Townshend et al., ATOM3D, NeurIPS, 2021.

Geometric Graph Neural Networks

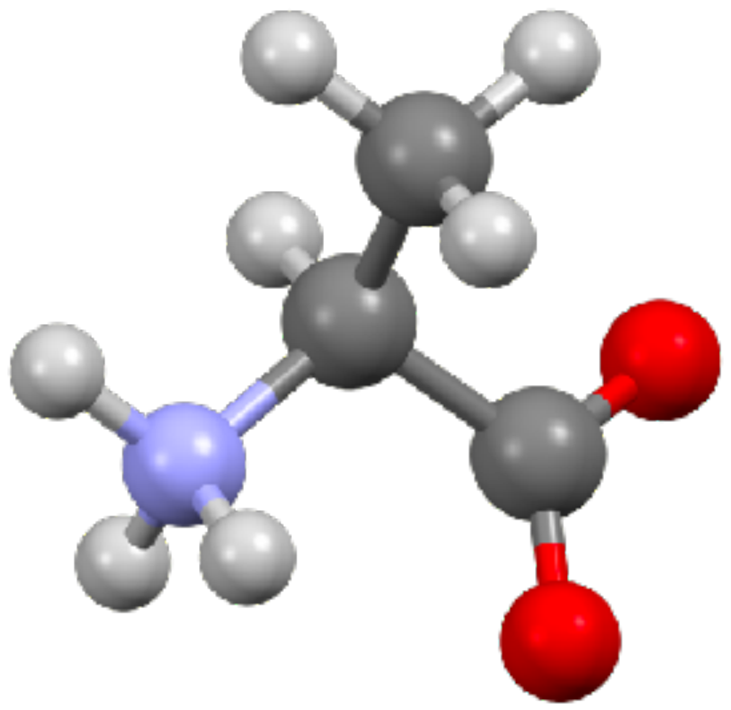
Fundamental tool for machine learning on geometric graphs

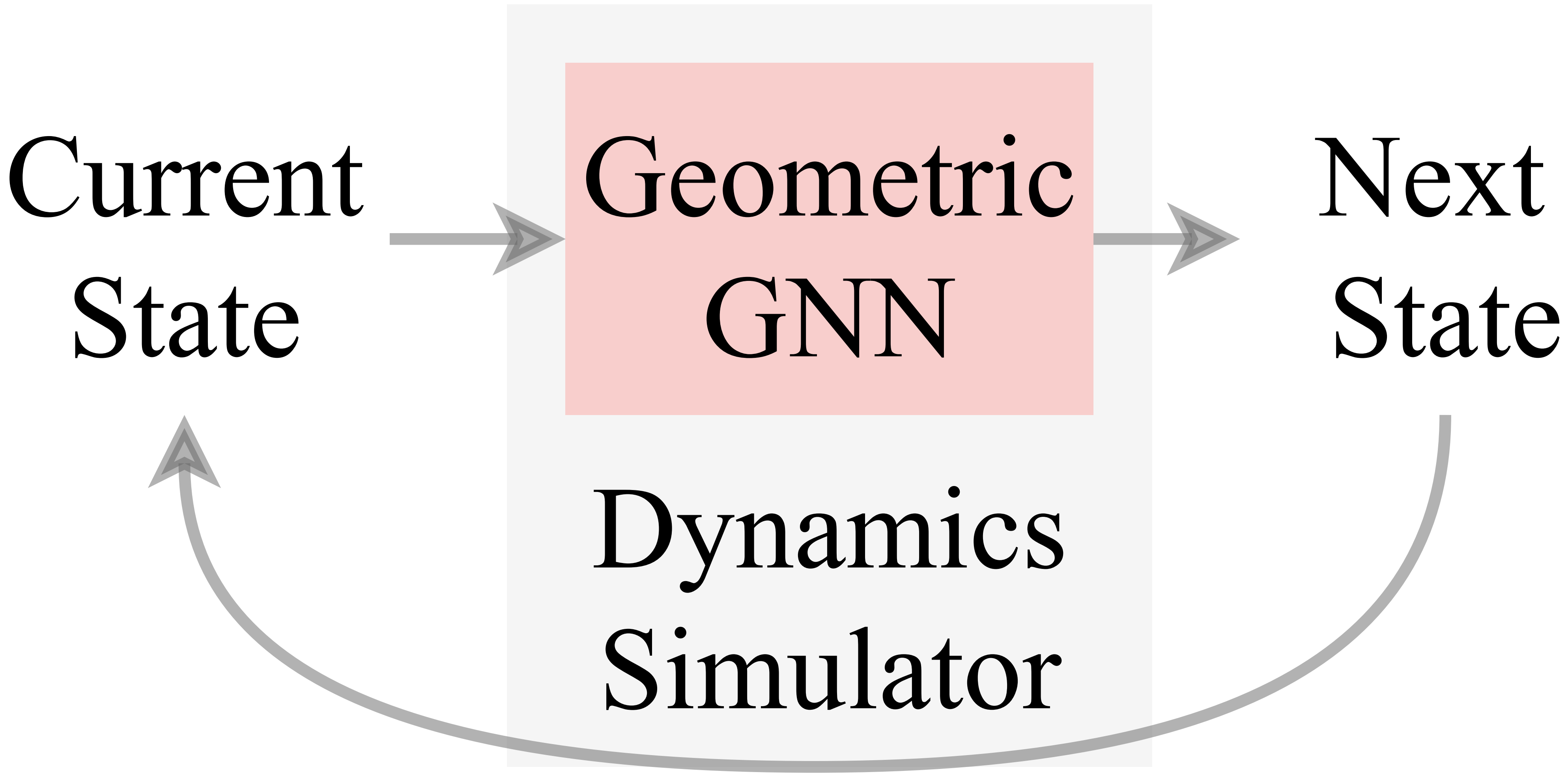


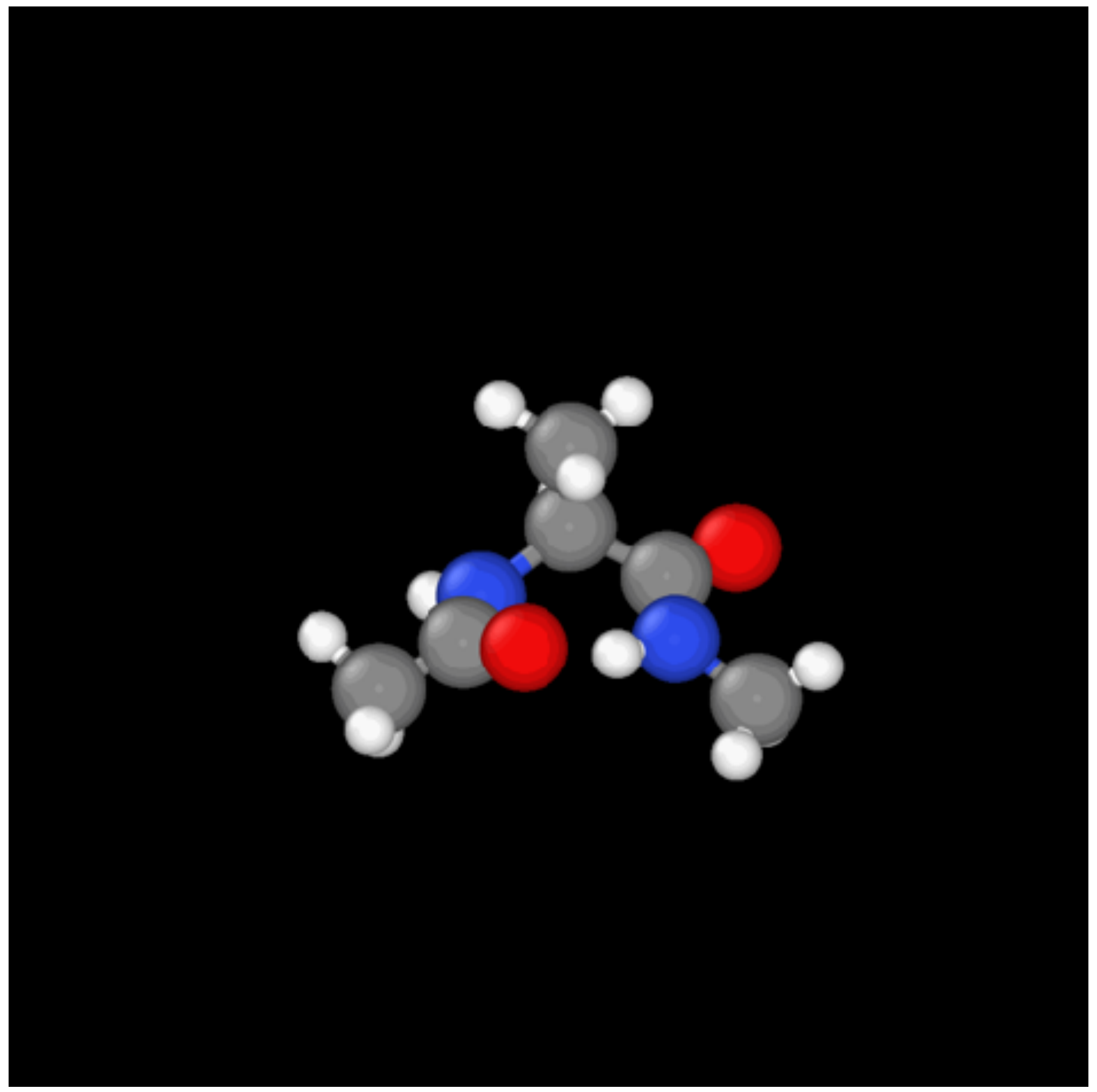
[1] Corso, Stärk, Jing, et al., DiffDock, ICLR, 2023.
[2] Ingram et al., Chroma, 2022.
[3] Dauparas et al., ProteinMPNN, Science, 2022.

Geometric Graph Neural Networks

Fundamental tool for machine learning on geometric graphs



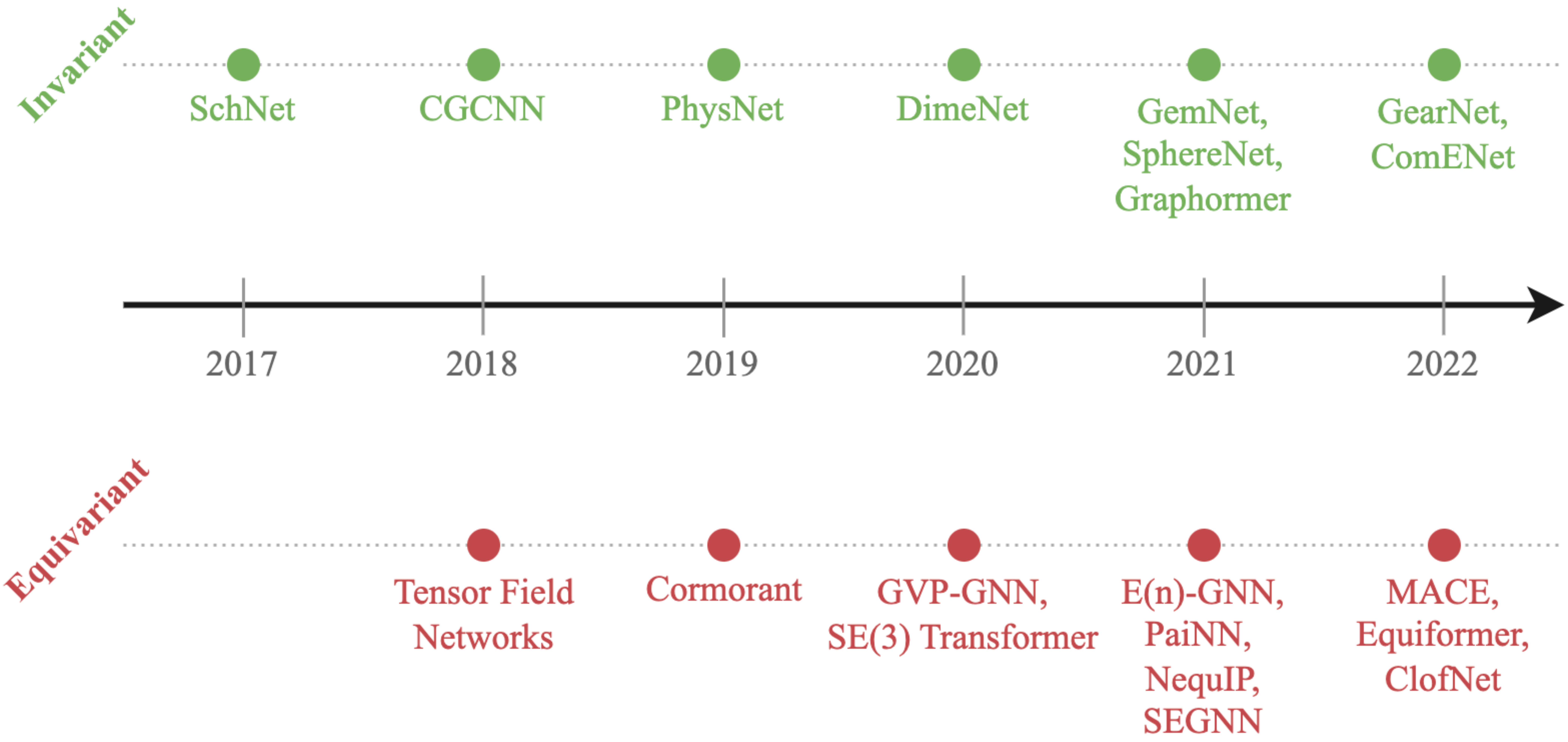




[1] Batzner et al., NequIP, Nature Communications, 2022. [2] Fu et al., Forces are not enough, arXiv 2022.

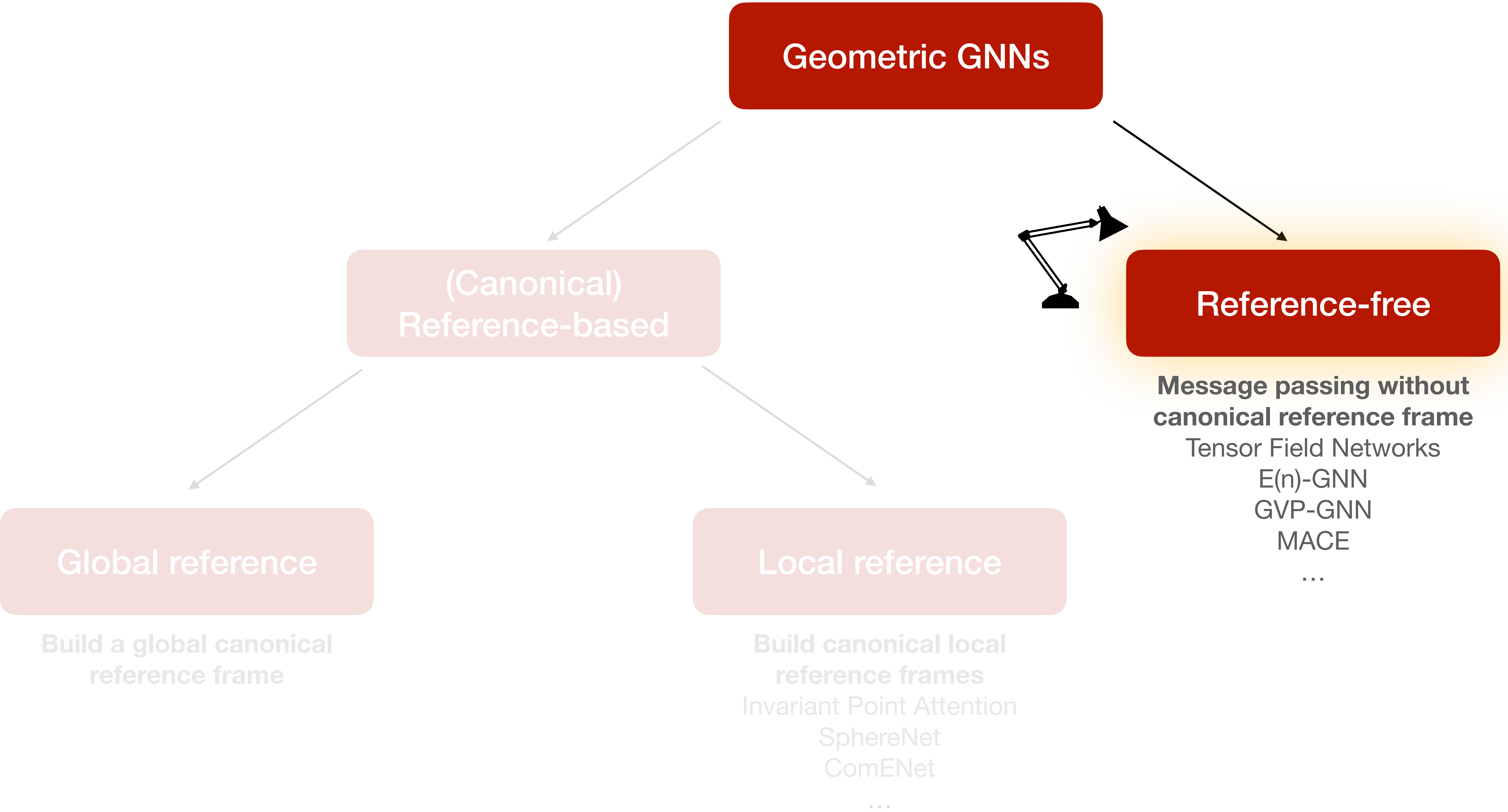
Timeline of Geometric GNN architectures

Categorised by intermediate features within layers



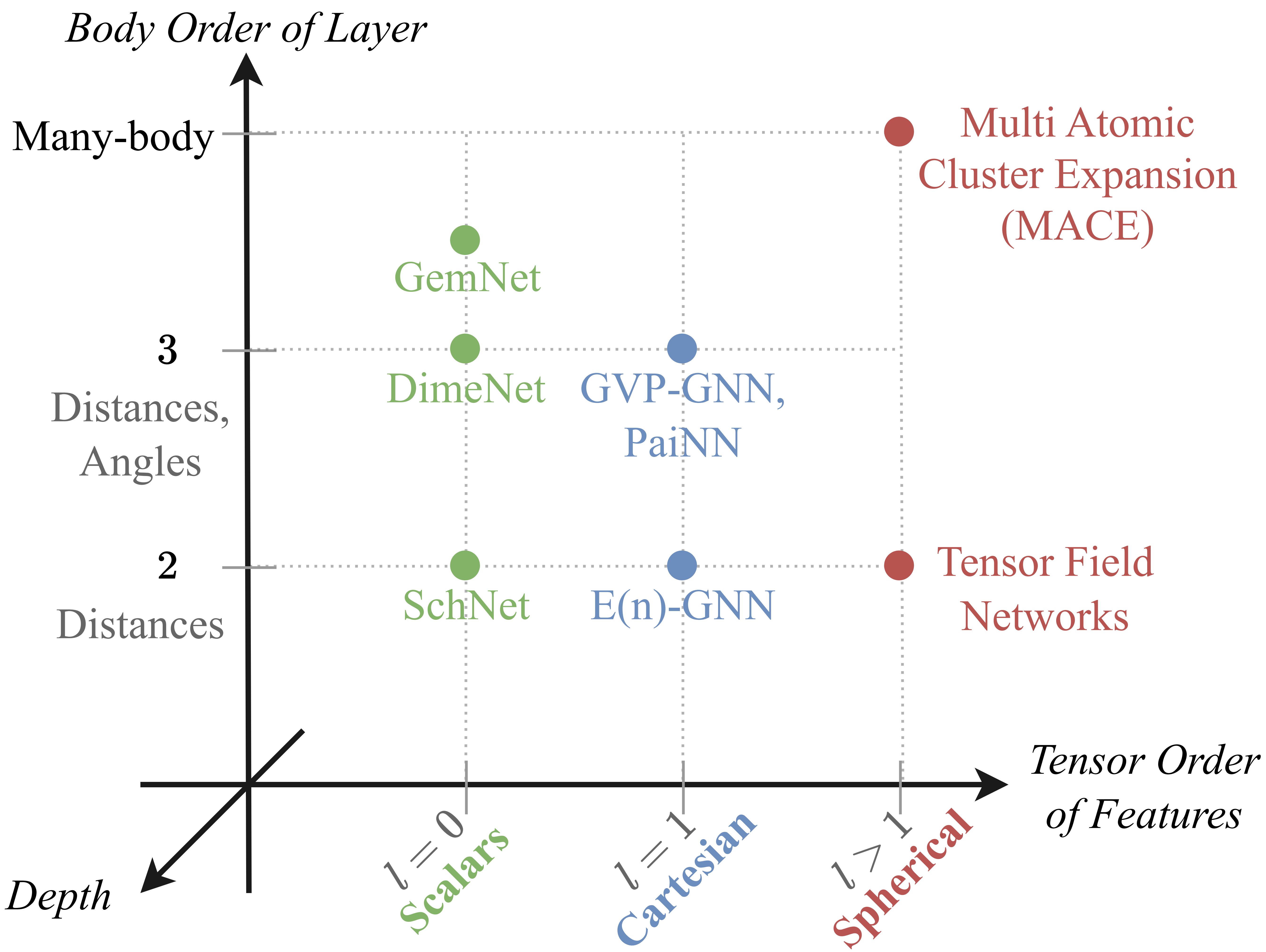
Geometric GNN land

An informal taxonomy of approaches



Axes of Geometric GNN expressivity

Key takeaway: deeper understanding of (reference-free) Geometric GNN design space

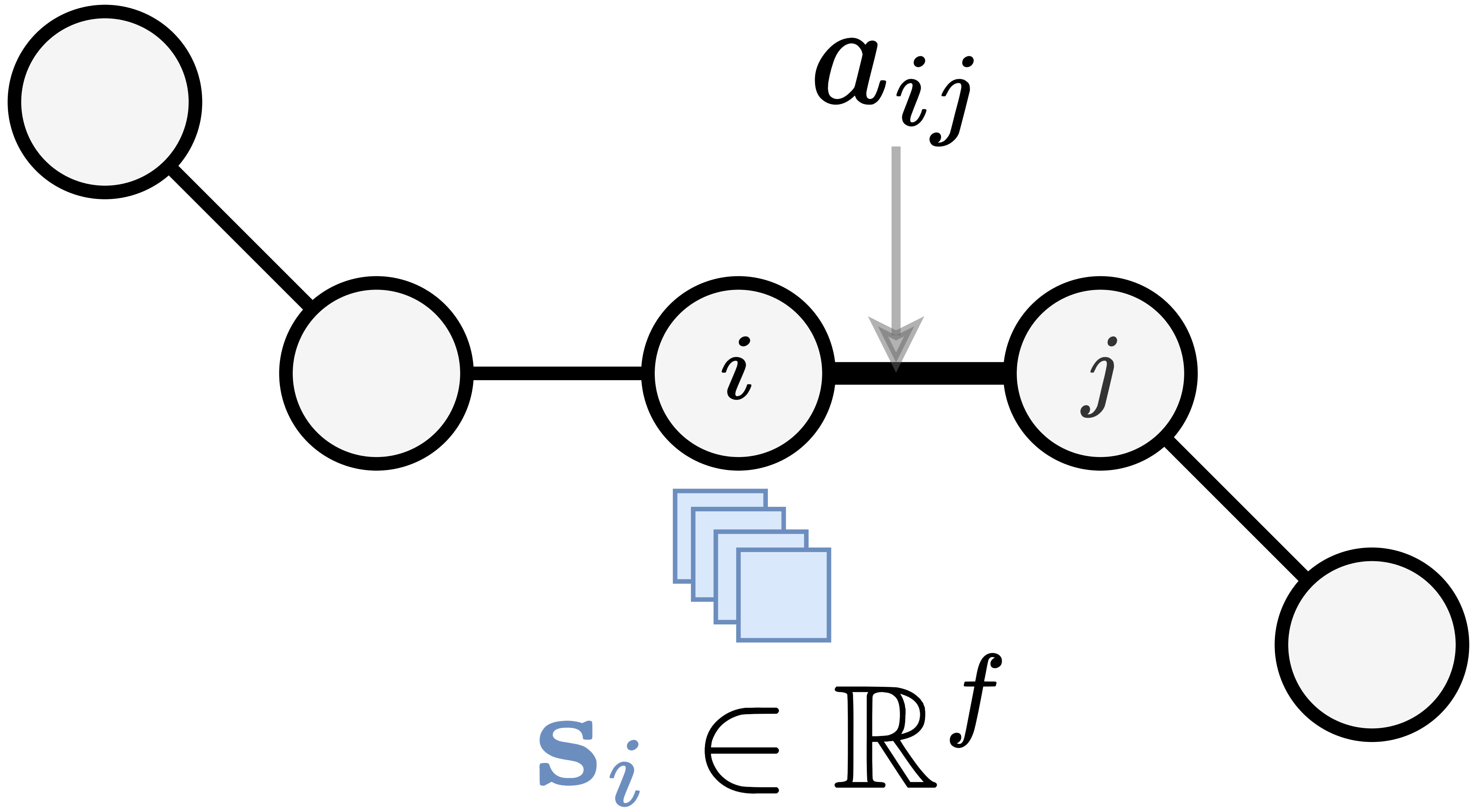


1. **Invariant layers:** limited expressivity, cannot distinguish one-hop identical geometric graphs.
2. **Equivariant layers:** distinguish larger classes of graphs, propagate geometric information beyond local neighbourhoods.
3. Demonstrates utility of **higher order tensors & scalarisation** for maximally powerful geometric GNNs.

Background:
Graph Neural Networks for Geometric Graphs

Normal graphs

A graph is a set of nodes connected by edges



E.g. atom type

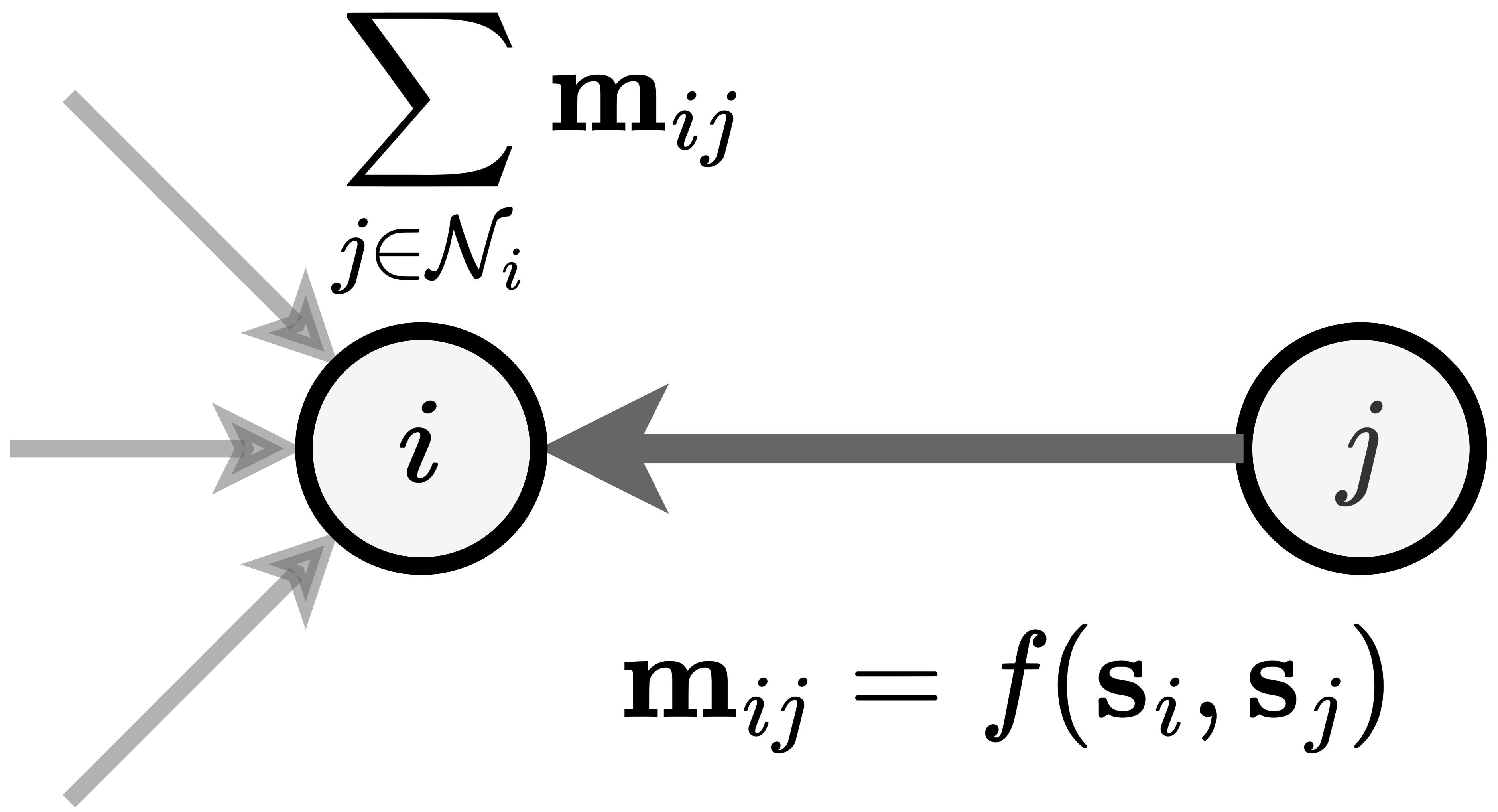
$$\mathcal{G} = (\boldsymbol{A}, \boldsymbol{S})$$

Scalar features $\in \mathbb{R}^{n \times f}$

$n \times n$ adjacency matrix

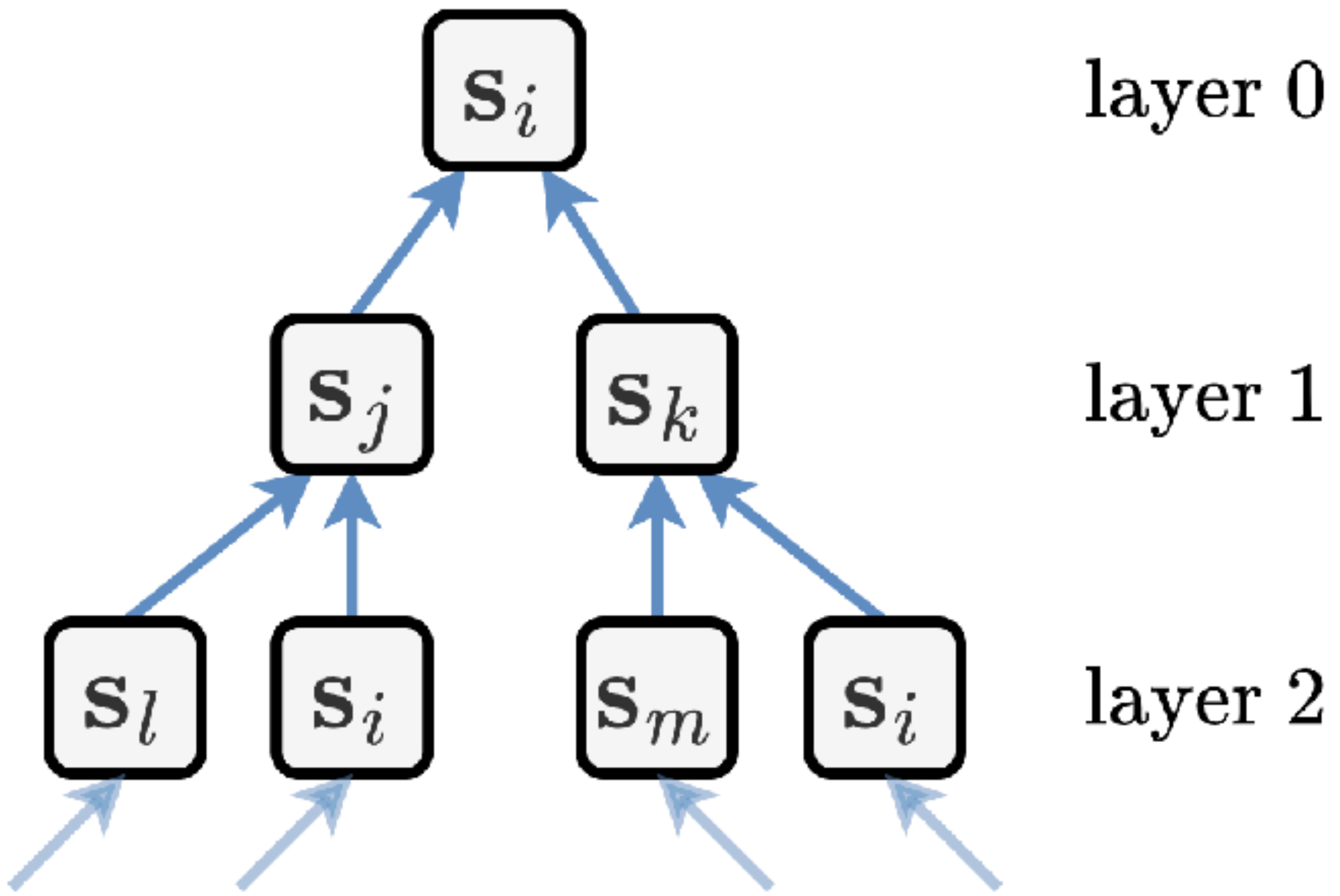
Normal Graph Neural Networks

Message passing updates node features using local aggregation



$$\boldsymbol{m}_i^{(t)} := \mathrm{AGG}\left(\{\!\{(\boldsymbol{s}_i^{(t)}, \boldsymbol{s}_j^{(t)}) \mid j \in \mathcal{N}_i\}\!\}\right),$$

$$\boldsymbol{s}_i^{(t+1)} := \mathrm{UPD}\left(\boldsymbol{s}_i^{(t)}, \boldsymbol{m}_i^{(t)}\right),$$

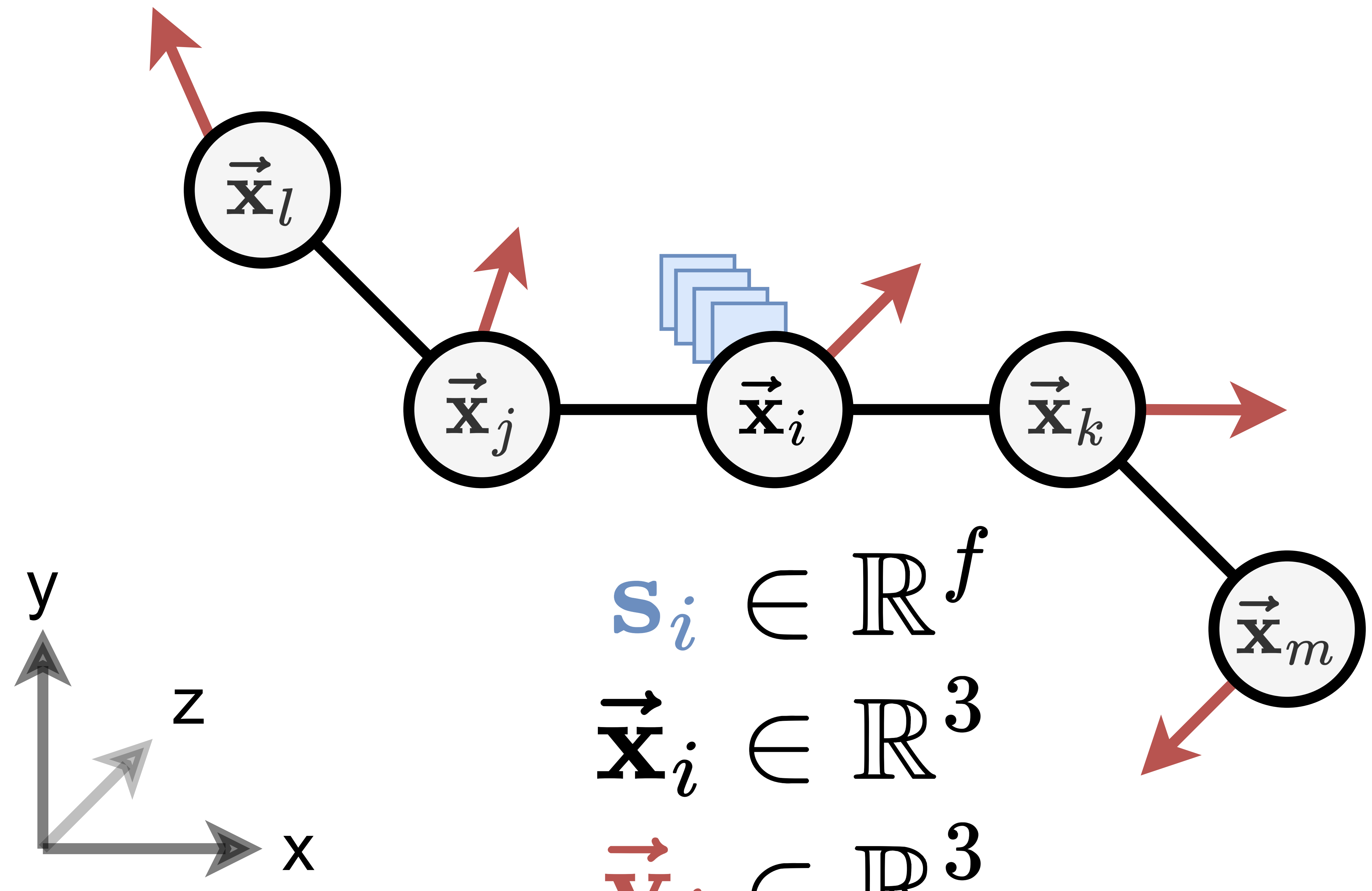


Computation tree:
Message passing gathers & propagates features beyond local neighbourhoods.

Geometric graphs

Each node is:

- **embedded in Euclidean space** *e.g.* atoms in 3D
- **decorated with geometric attributes** *s.a.* velocity



$$\mathbf{s}_i \in \mathbb{R}^f$$
$$\vec{\mathbf{x}}_i \in \mathbb{R}^3$$
$$\vec{\mathbf{v}}_i \in \mathbb{R}^3$$

$$\mathcal{G} = (\boldsymbol{A}, \boldsymbol{S}, \vec{\boldsymbol{X}}, \vec{\boldsymbol{V}})$$

Geometric features* $\in \mathbb{R}^{n \times 3}$

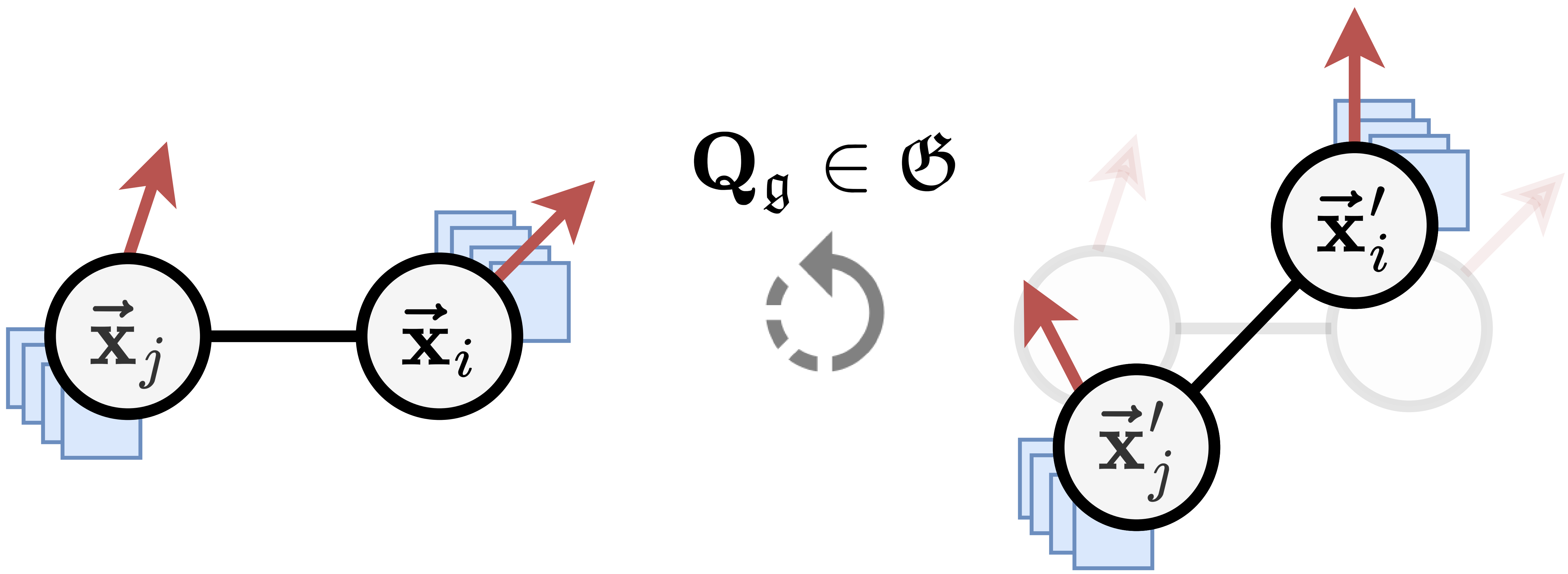
Coordinates $\in \mathbb{R}^{n \times 3}$

* We work with a single vector feature per node, but our setup generalises to multiple vector features and higher-order tensors.

Physical symmetries

Geometric attributes transform with Euclidean transformations of the system

Rotations & **Reflections** $Q_{\mathfrak{g}} \in \mathfrak{G}$ act on only vectors $\vec{V}$ and coordinates $\vec{X}$:



$$\mathbf{s}_i' = \mathbf{s}_i$$

$$\vec{\mathbf{x}}_i' = \mathbf{Q}_{\mathfrak{g}} \vec{\mathbf{x}}_i$$

$$\vec{\mathbf{v}}_i' = \mathbf{Q}_{\mathfrak{g}} \vec{\mathbf{v}}_i$$

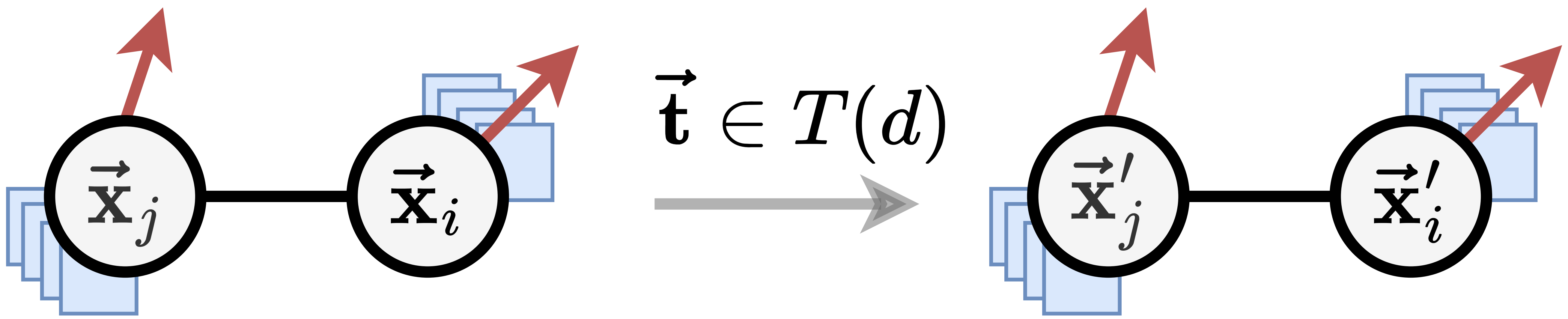
Scalar features remain unchanged $\rightarrow$ **invariant.**

* We use $\mathfrak{G}$ to denote rotations SO(d) or rotations and reflections O(d)

Physical symmetries

Geometric attributes transform with Euclidean transformations of the system

Translations $\vec{t} \in T(d)$ act on only the coordinates $\vec{X}$:



$$\mathbf{s}'_i = \mathbf{s}_i$$

$$\vec{\mathbf{x}}'_i = \vec{\mathbf{x}}_i + \vec{\mathbf{t}}$$

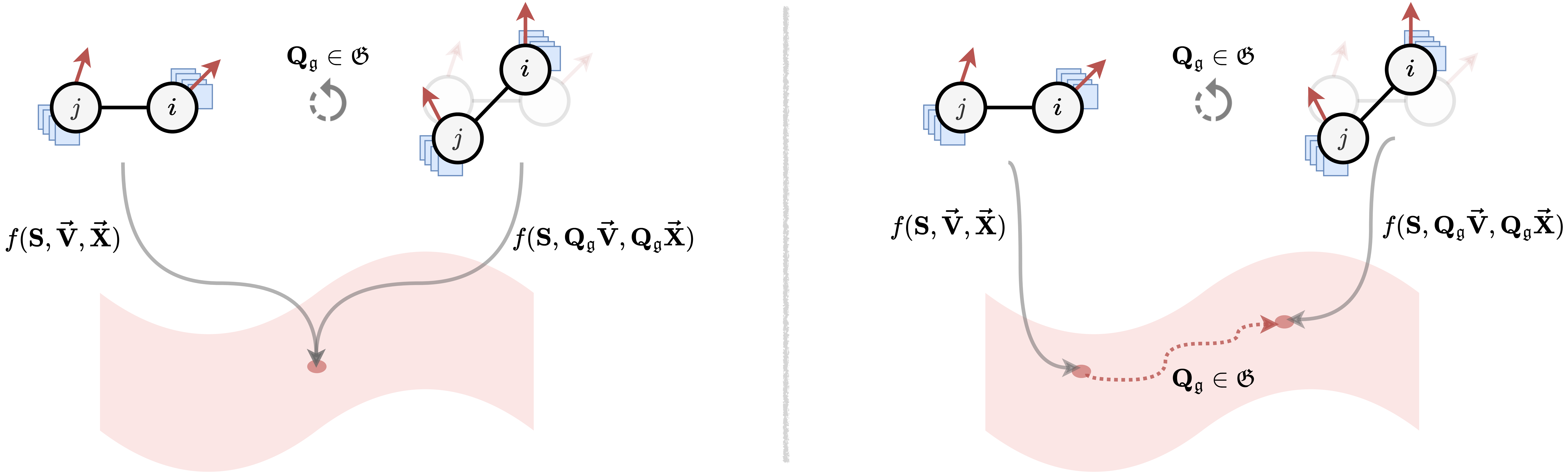
$$\vec{\mathbf{v}}'_i = \vec{\mathbf{v}}_i$$

Scalar and vector features remain unchanged $\rightarrow$ **invariant.**

Building blocks of Geometric GNNs

Normal GNNs do not retain the transformation semantics:

- Scalar features **must be updated in an invariant manner.**
- Vector features **must be updated in an equivariant manner.**

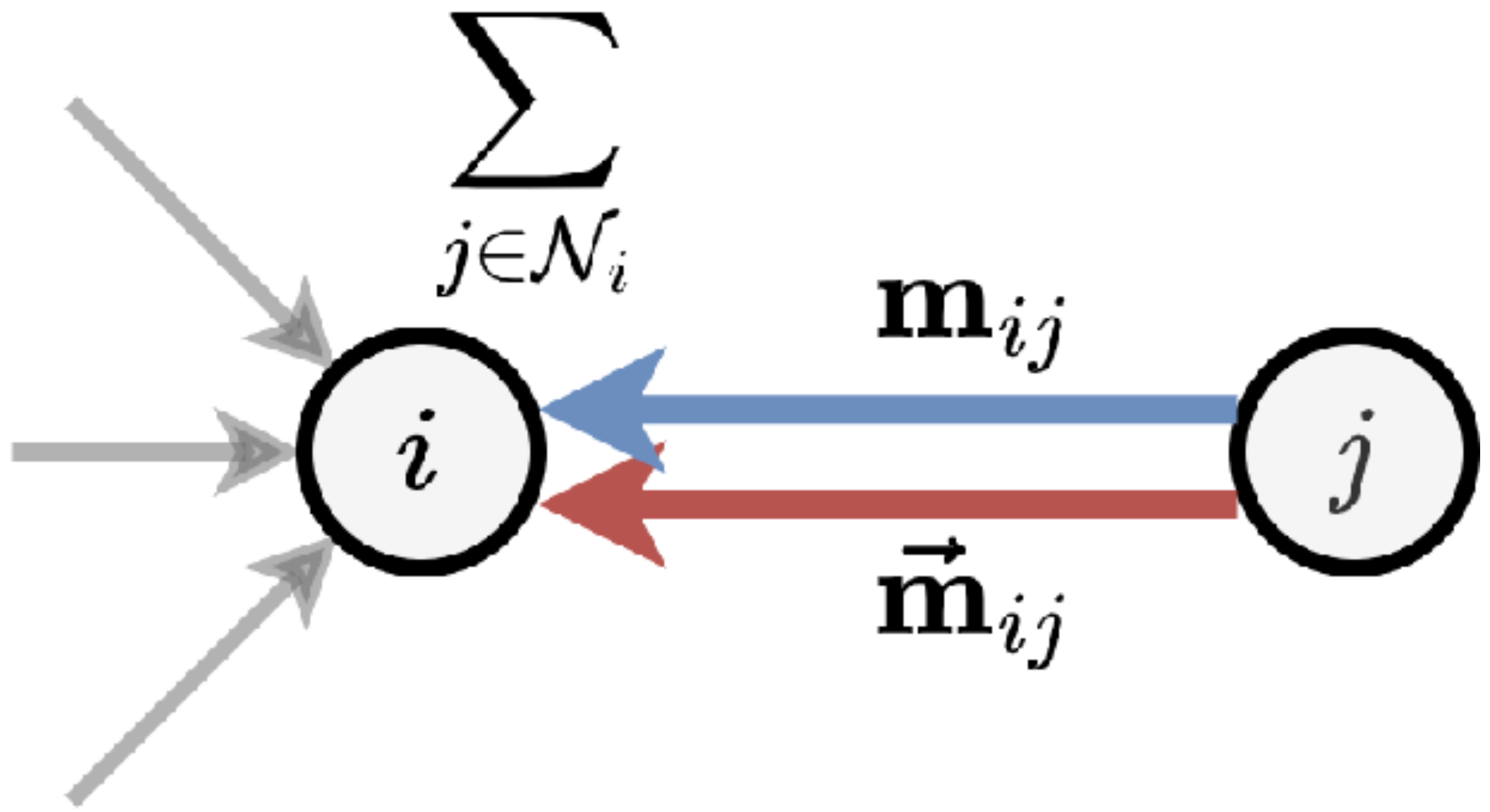


Invariant functions vs. Equivariant functions

Geometric GNN message passing

Geometric GNNs:

- **update** **scalar** **and** (optionally) **vector features**
- **aggregate and update functions which retain transformation semantics**



$$\boldsymbol{m}_i^{(t)}, \vec{\boldsymbol{m}}_i^{(t)} := \textsc{Agg}\left(\{\!\{(\boldsymbol{s}_i^{(t)}, \boldsymbol{s}_j^{(t)}, \vec{\boldsymbol{v}}_i^{(t)}, \vec{\boldsymbol{v}}_j^{(t)}, \vec{\boldsymbol{x}}_{ij}) \mid j \in \mathcal{N}_i\}\!\}\right) \quad \text{(Aggregate)}$$

$$\boldsymbol{s}_i^{(t+1)}, \vec{\boldsymbol{v}}_i^{(t+1)} := \textsc{Upd}\left((\boldsymbol{s}_i^{(t)}, \vec{\boldsymbol{v}}_i^{(t)}) \;,\; (\boldsymbol{m}_i^{(t)}, \vec{\boldsymbol{m}}_i^{(t)})\right) \quad \text{(Update)}$$

Design Space of Geometric GNNs

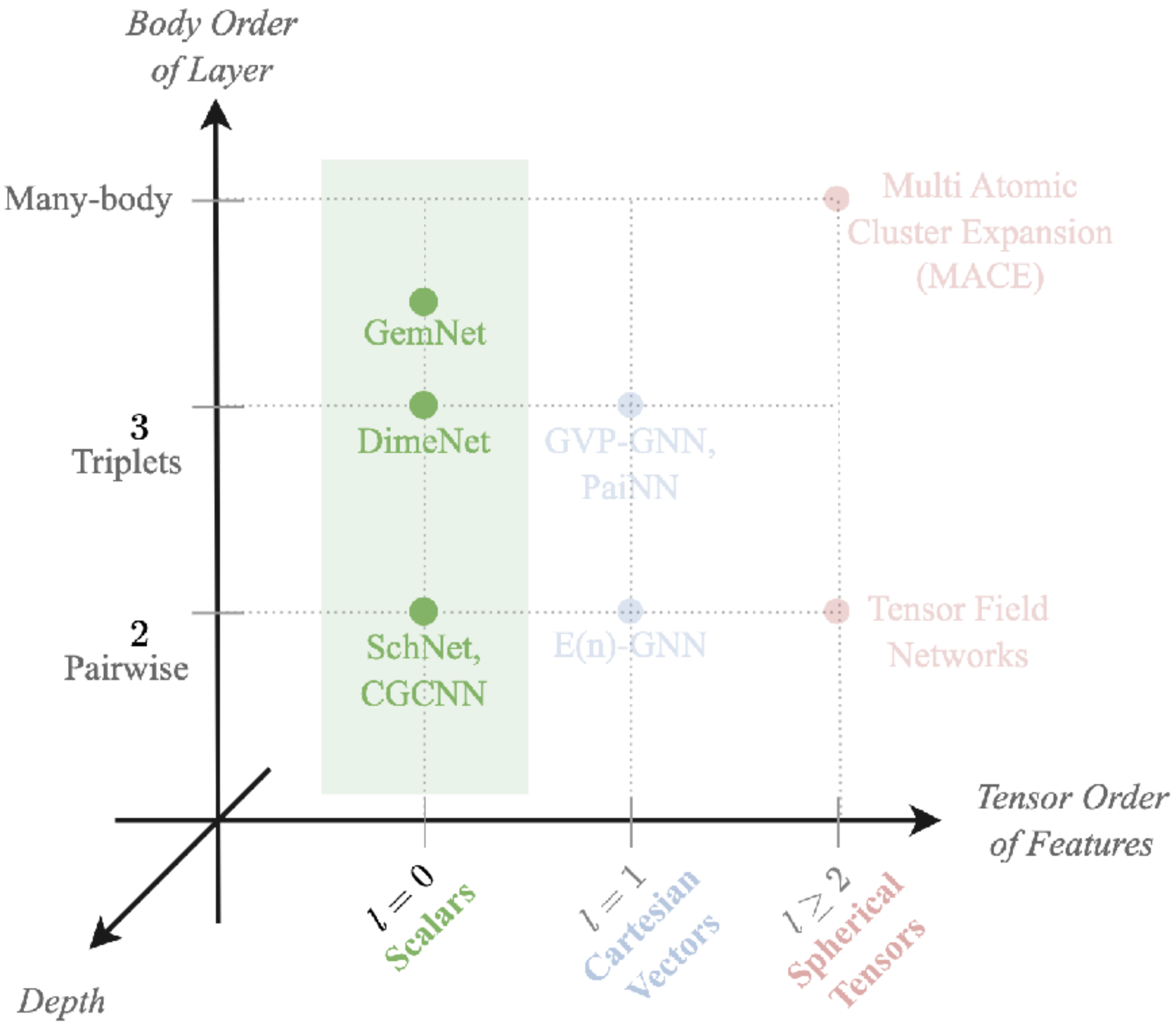
- **Body order of scalarisation**
- **Invariance vs. Equivariance**
- **Tensor order of features**

[1] Batatia, Batzner, et al., The Design Space of E(3)-Equivariant Atom-Centered Interatomic Potentials, 2022.

𝔊-invariant Geometric GNNs

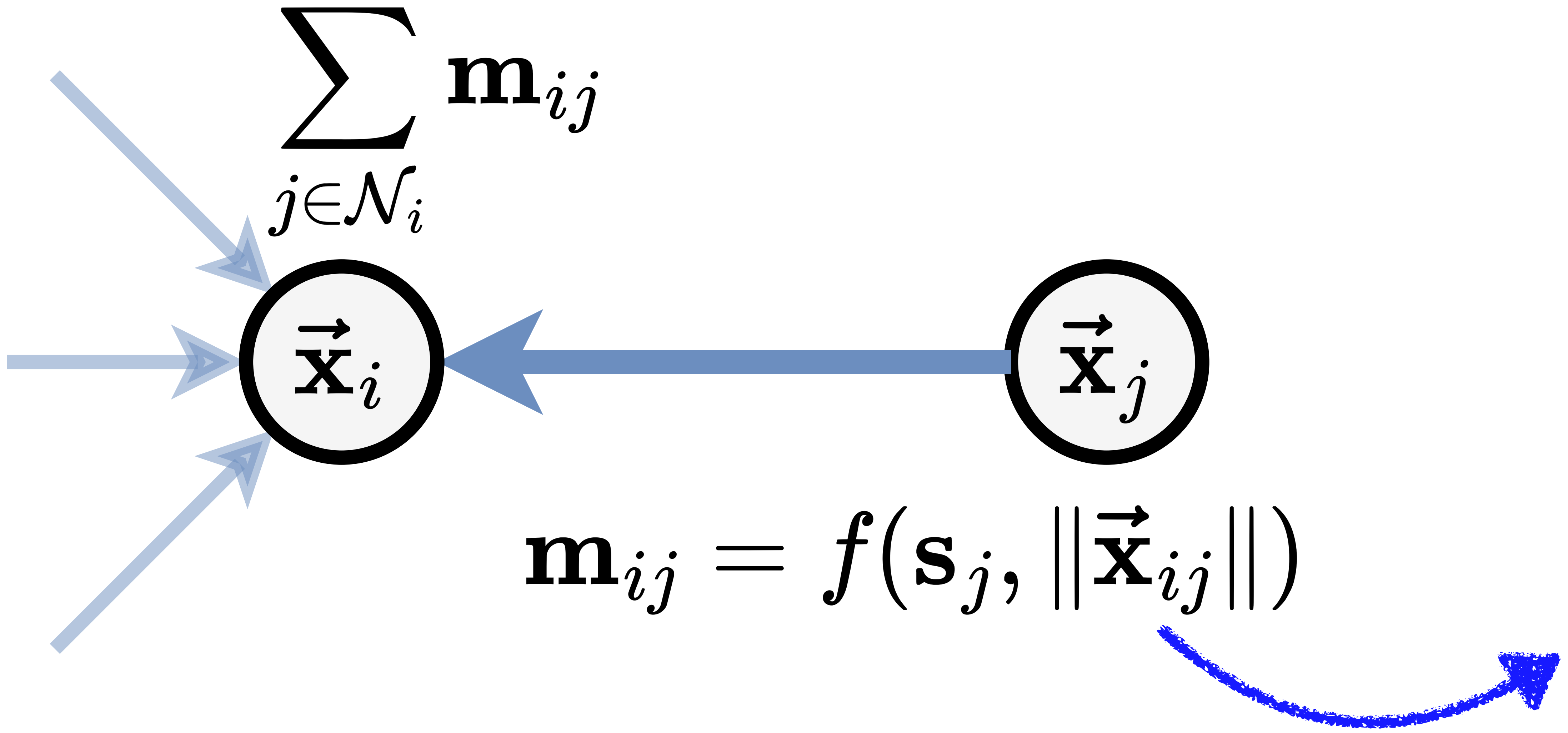
Only update scalar features via scalarising local geometric information

Key design choice:
Body order of scalarisation



SchNet [1]

SchNet uses relative distances $\|\vec{x}_i - \vec{x}_j\|$ to scalarise local geometry



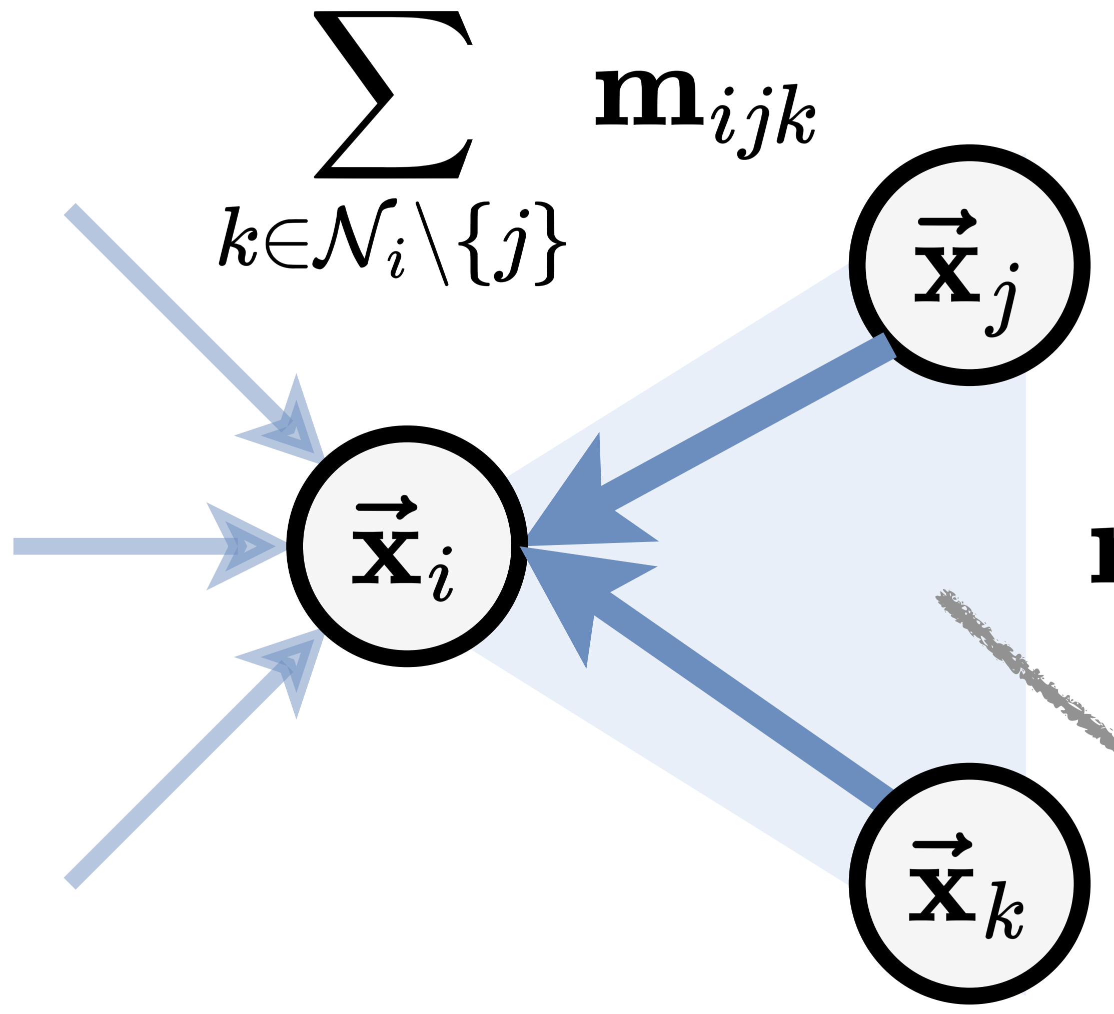
Relative distances are invariant to global rotations & translations.

$$\boldsymbol{s}_i^{(t+1)} := \boldsymbol{s}_i^{(t)} + \sum_{j \in \mathcal{N}_i} f_1\left(\boldsymbol{s}_j^{(t)},\ \|\vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_j\|\right)$$

[1] Schütt et al., SchNet, Journal of Chemical Physics, 2018.
[2] Xie and Grossman, CGCNN, Phys. Rev. Letters, 2018.
[3] Li et al., IROS, 2020. Similar architecture for multi-agent robotics.
[4] Sanchez-Gonzalez et al., ICML, 2020. Similar architecture for physical simulation

DimeNet [1]

DimeNet uses distances $\|\vec{x}_{ij}\|$ and angles $\vec{x}_{ij} \cdot \vec{x}_{ik}$ among triplets



$$\mathbf{m}_{ijk} = f(\mathbf{s}_j, \mathbf{s}_k, \|\vec{\mathbf{x}}_{ij}\|, \vec{\mathbf{x}}_{ij} \cdot \vec{\mathbf{x}}_{ik})$$

Body order of scalarisation:
number of nodes involved in computing local invariant scalars.

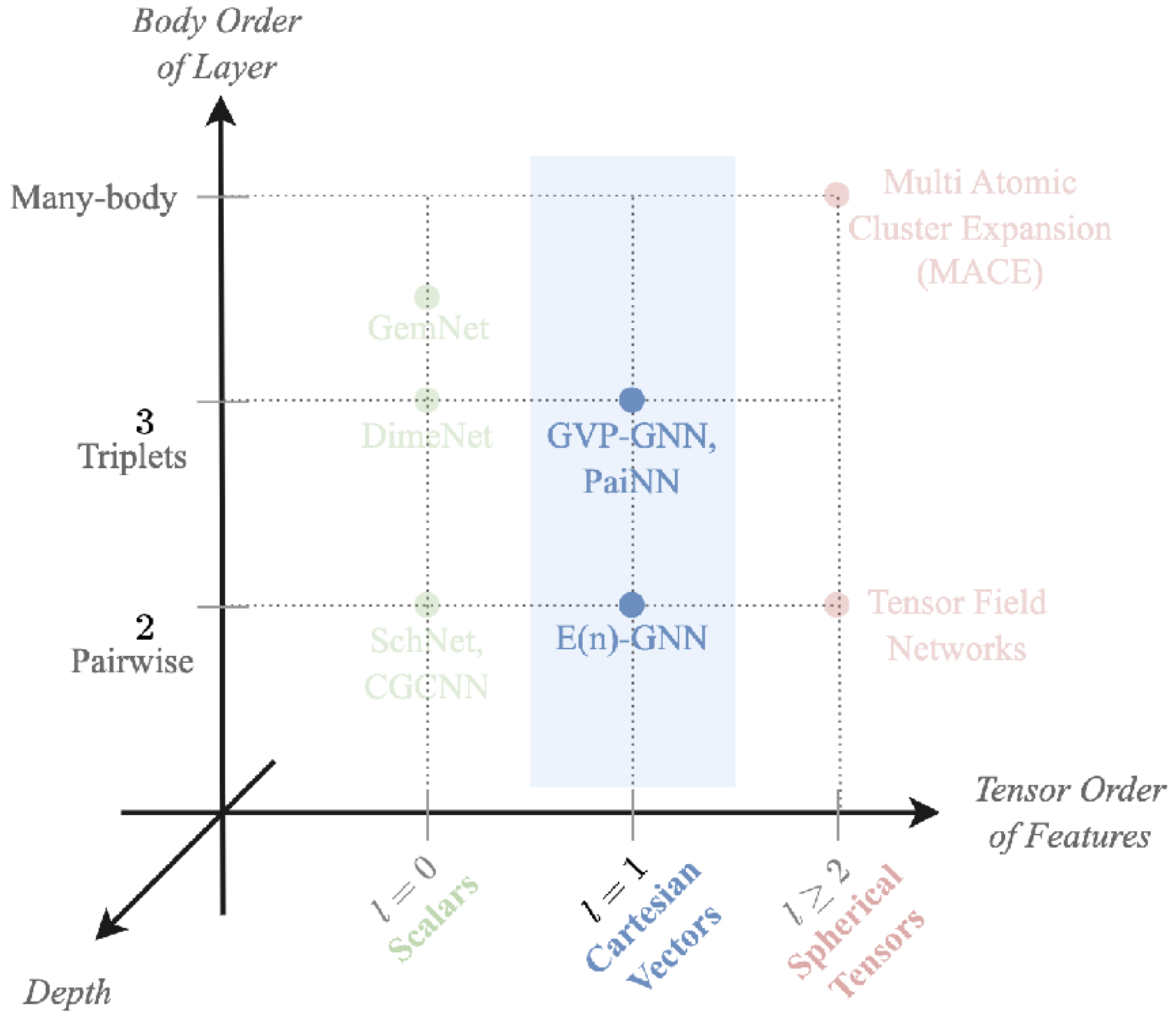
$$\boldsymbol{s}_i^{(t+1)} := \sum_{j \in \mathcal{N}_i} f_1\Big(\boldsymbol{s}_i^{(t)},\ \boldsymbol{s}_j^{(t)},\ \sum_{k \in \mathcal{N}_i \setminus \{j\}} f_2\left(\boldsymbol{s}_j^{(t)},\ \boldsymbol{s}_k^{(t)},\ \|\vec{\boldsymbol{x}}_{ij}\|,\ \vec{\boldsymbol{x}}_{ij} \cdot \vec{\boldsymbol{x}}_{ik}\right)\Big)$$

[1] Gasteiger et al., DimeNet, ICLR, 2020.

Cartesian 𝔊-equivariant Geometric GNNs

Update scalar and vector features in cartesian basis

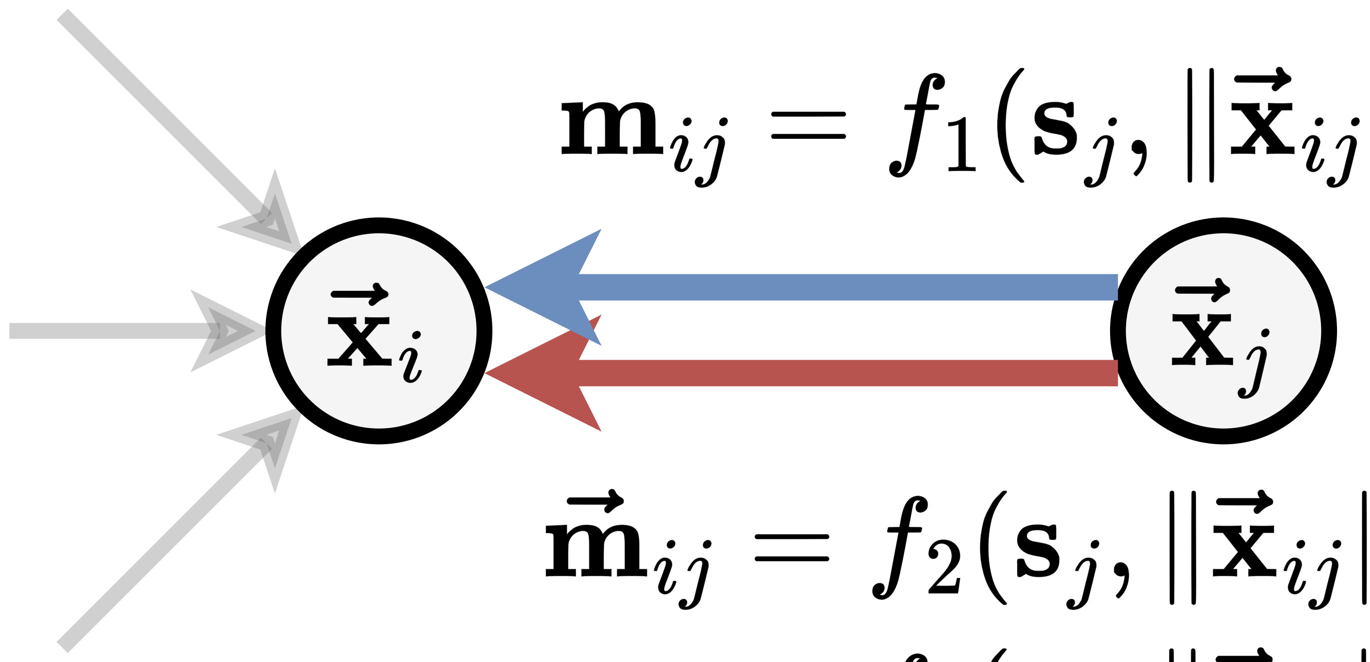
Key design choice:
From invariant to equivariant message passing



PaiNN [1]

Update both scalar & vector features by propagating geometric messages

$$\mathbf{m}_{ij} = f_1(\mathbf{s}_j, \|\vec{\mathbf{x}}_{ij}\|)$$



$$\vec{\mathbf{m}}_{ij} = f_2(\mathbf{s}_j, \|\vec{\mathbf{x}}_{ij}\|) \odot \vec{\mathbf{v}}_j + f_3(\mathbf{s}_j, \|\vec{\mathbf{x}}_{ij}\|) \odot \vec{\mathbf{x}}_{ij}$$

Ensuring equivariance:
gated non-linearity, no ReLU on vectors, limited to sum/dot/cross products.

$$\boldsymbol{s}_i^{(t+1)} := \sum_{j \in \mathcal{N}_i} f_1\left(\boldsymbol{s}_j^{(t)},\ \|\vec{\boldsymbol{x}}_{ij}\|\right)$$

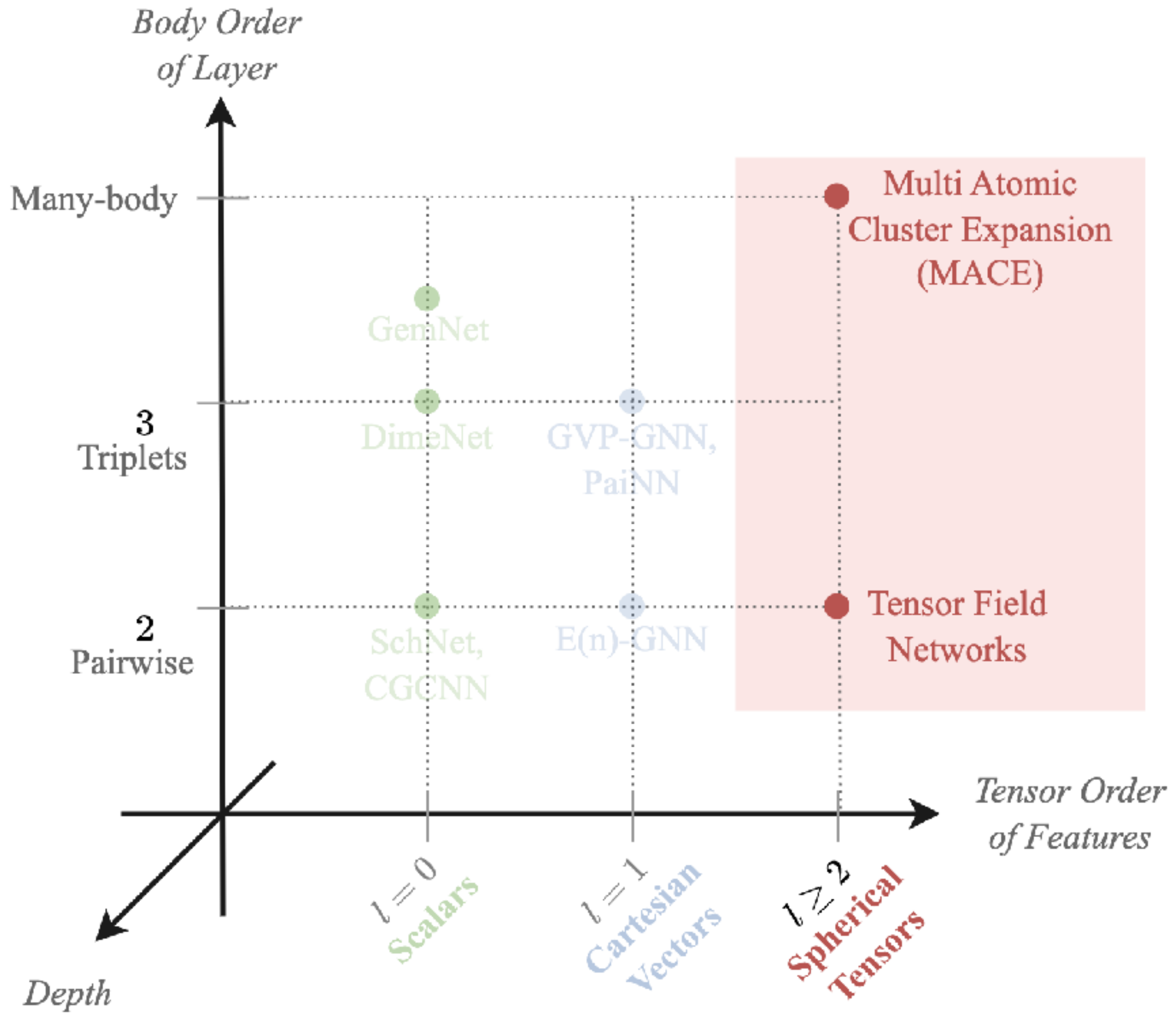
$$\vec{\boldsymbol{v}}_i^{(t+1)} := \sum_{j \in \mathcal{N}_i} f_2\left(\boldsymbol{s}_j^{(t)},\ \|\vec{\boldsymbol{x}}_{ij}\|\right) \odot \vec{\boldsymbol{v}}_j^{(t)} + \sum_{j \in \mathcal{N}_i} f_3\left(\boldsymbol{s}_j^{(t)},\ \|\vec{\boldsymbol{x}}_{ij}\|\right) \odot \vec{\boldsymbol{x}}_{ij}$$

[1] Schütt et al., PaiNN, ICML, 2021. [2] Satorras et al., E(n)-GNN, ICML, 2021. [3] Jing et al., GVP-GNN, ICLR, 2020.

Spherical 𝔊-equivariant Geometric GNNs

Update higher order spherical tensor features

Key design choice:
Tensor order of equivariant features



Tensor Field Networks [1]

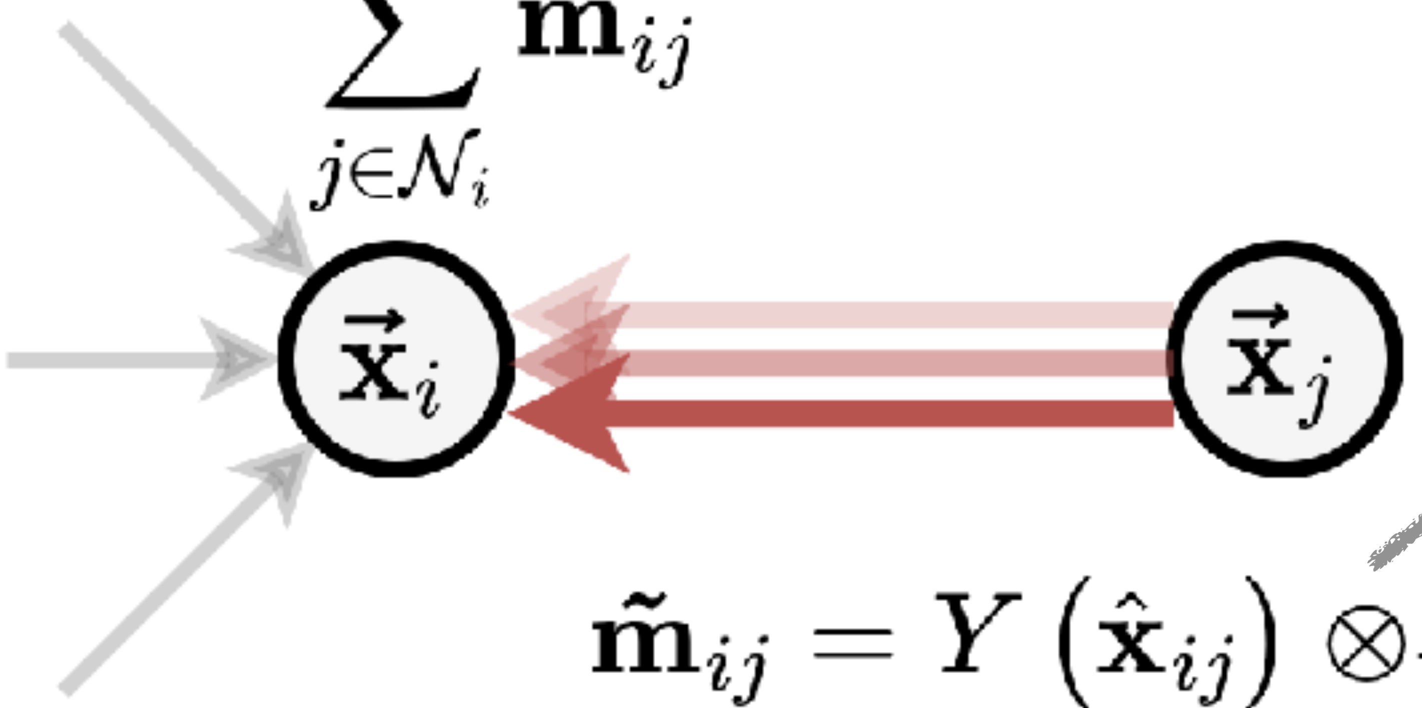
- **Higher order spherical tensors as node features** $\tilde{\boldsymbol{h}}_{i,l} \in \mathbb{R}^{2l+1 \times f}, \; l = 0, \ldots, L$
- **...updated via tensor products** $\otimes_{\boldsymbol{w}}$ **of neighbourhood features**
- **...with spherical harmonic expansion of displacement** $Y_l\left(\hat{\boldsymbol{x}}_{ij}\right) \in \mathbb{R}^{2l+1}$

Connection with cartesian basis:

$$\tilde{\boldsymbol{h}}_{i,l=0} \in \mathbb{R}^{1 \times f} \equiv \boldsymbol{s}_i$$

$$\tilde{\boldsymbol{h}}_{i,l=1} \in \mathbb{R}^{3 \times f} \equiv \vec{\boldsymbol{v}}_i$$

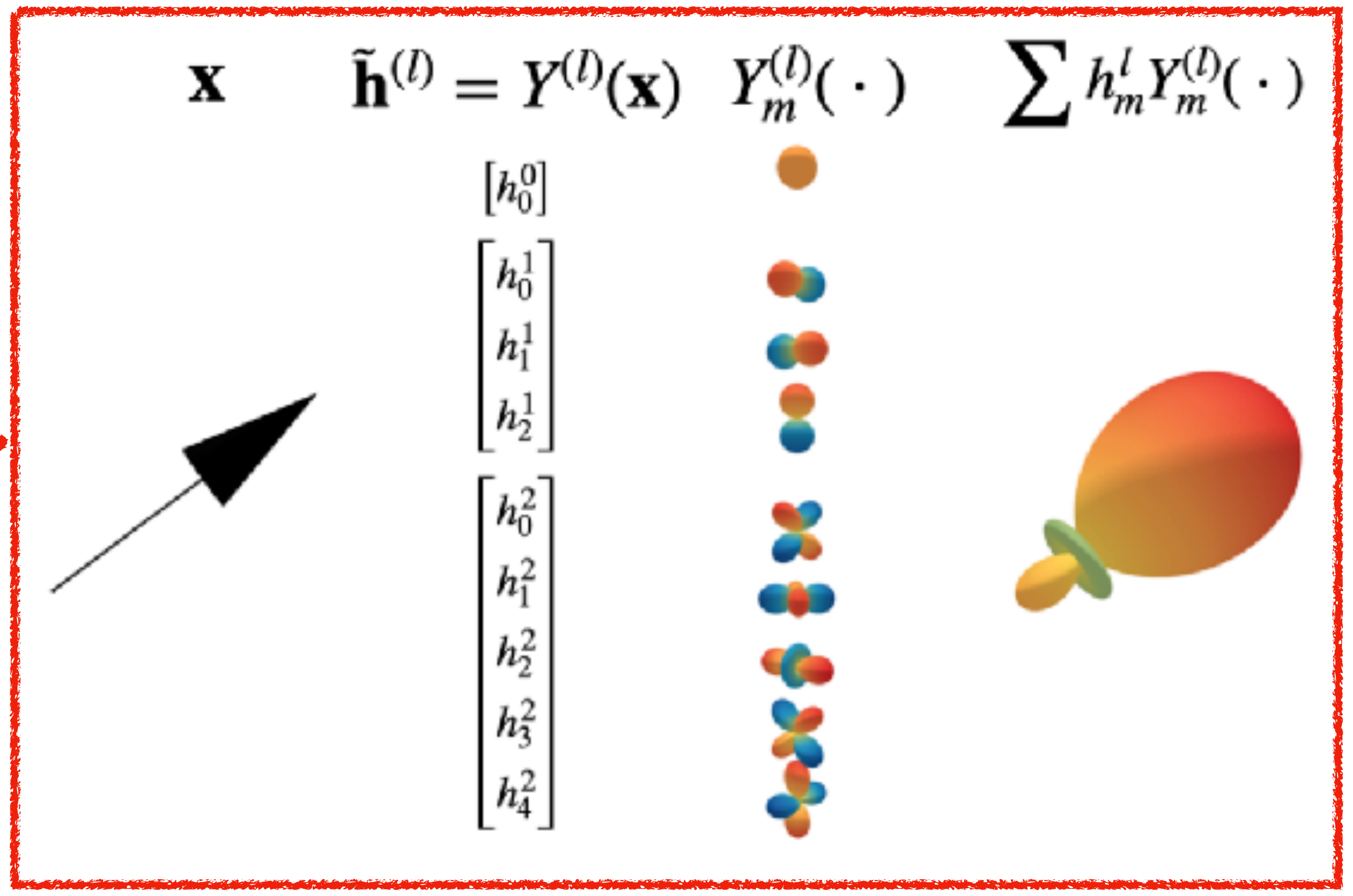
$$\sum_{j \in \mathcal{N}_i} \tilde{\mathbf{m}}_{ij}$$



$$\tilde{\mathbf{m}}_{ij} = Y\left(\hat{\mathbf{x}}_{ij}\right) \otimes_{\mathbf{w}} \tilde{\mathbf{h}}_j^{(t)}$$

Tensor product weights:

$$\boldsymbol{w} = f_{\mathrm{RBF}}\left(\boldsymbol{s}_j, \|\vec{\boldsymbol{x}}_{ij}\|\right)$$



$$\tilde{\boldsymbol{h}}_i^{(t+1)} := \tilde{\boldsymbol{h}}_i^{(t)} + \sum_{j \in \mathcal{N}_i} Y\left(\hat{\boldsymbol{x}}_{ij}\right) \otimes_{\boldsymbol{w}} \tilde{\boldsymbol{h}}_j^{(t)},$$

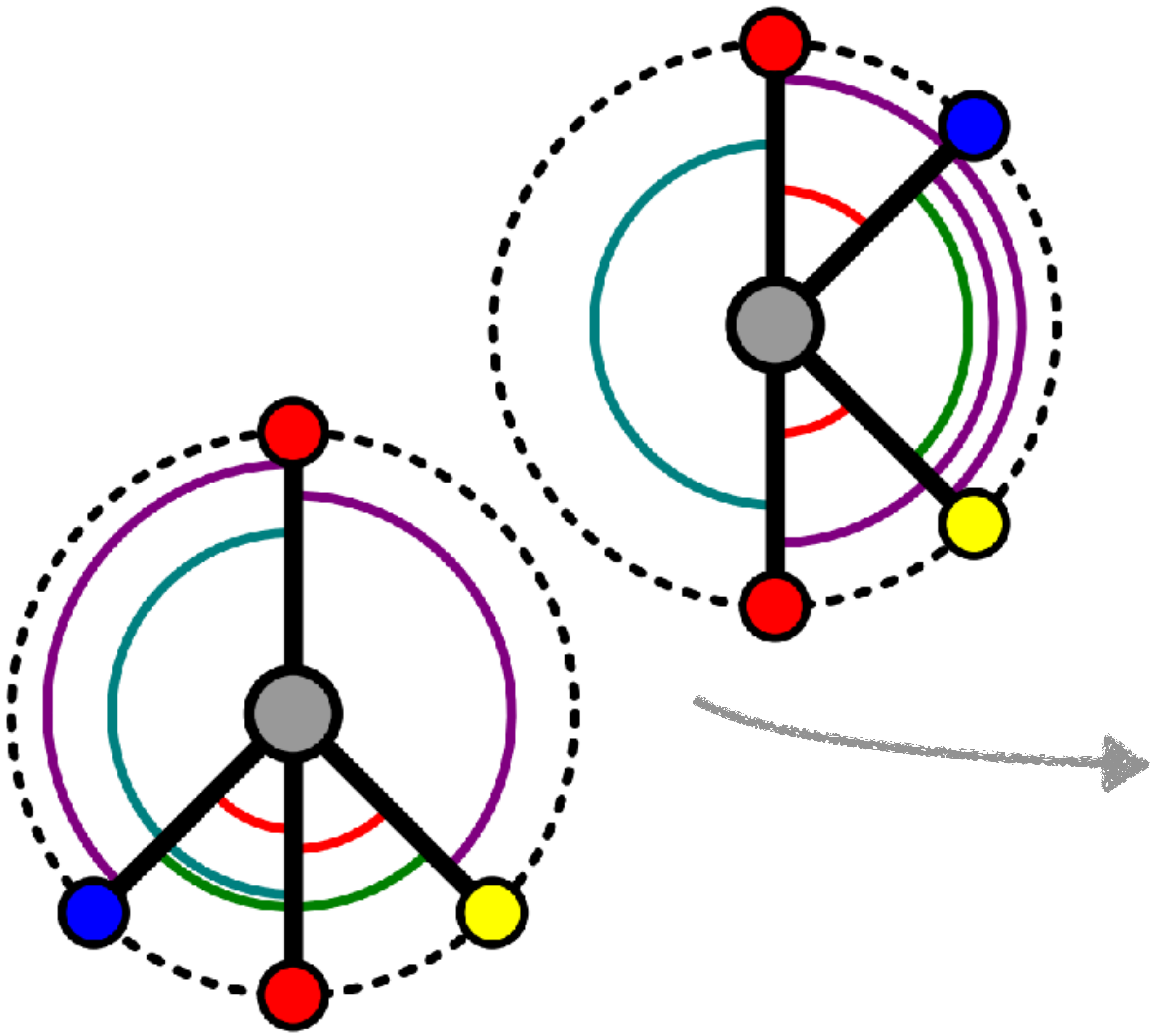
[1] Thomas, Smidt, et al., TFN, 2018. [2] Brandstetter, Hesselink, et al., SEGNN, ICLR,2022. [3] Batatia et al., MACE, NeurIPS, 2022. [4] Geiger and Smidt, e3nn, 2022.

Motivation:
How powerful are Geometric GNNs?

- **How do key design choices impact expressive power?**
- **Connect theoretical limitations — practical implications**

Distinguishing geometric neighbourhoods

Can you tell these two local neighbourhoods apart using the **<u>unordered set</u> of distances** and **angles**, only?[2]



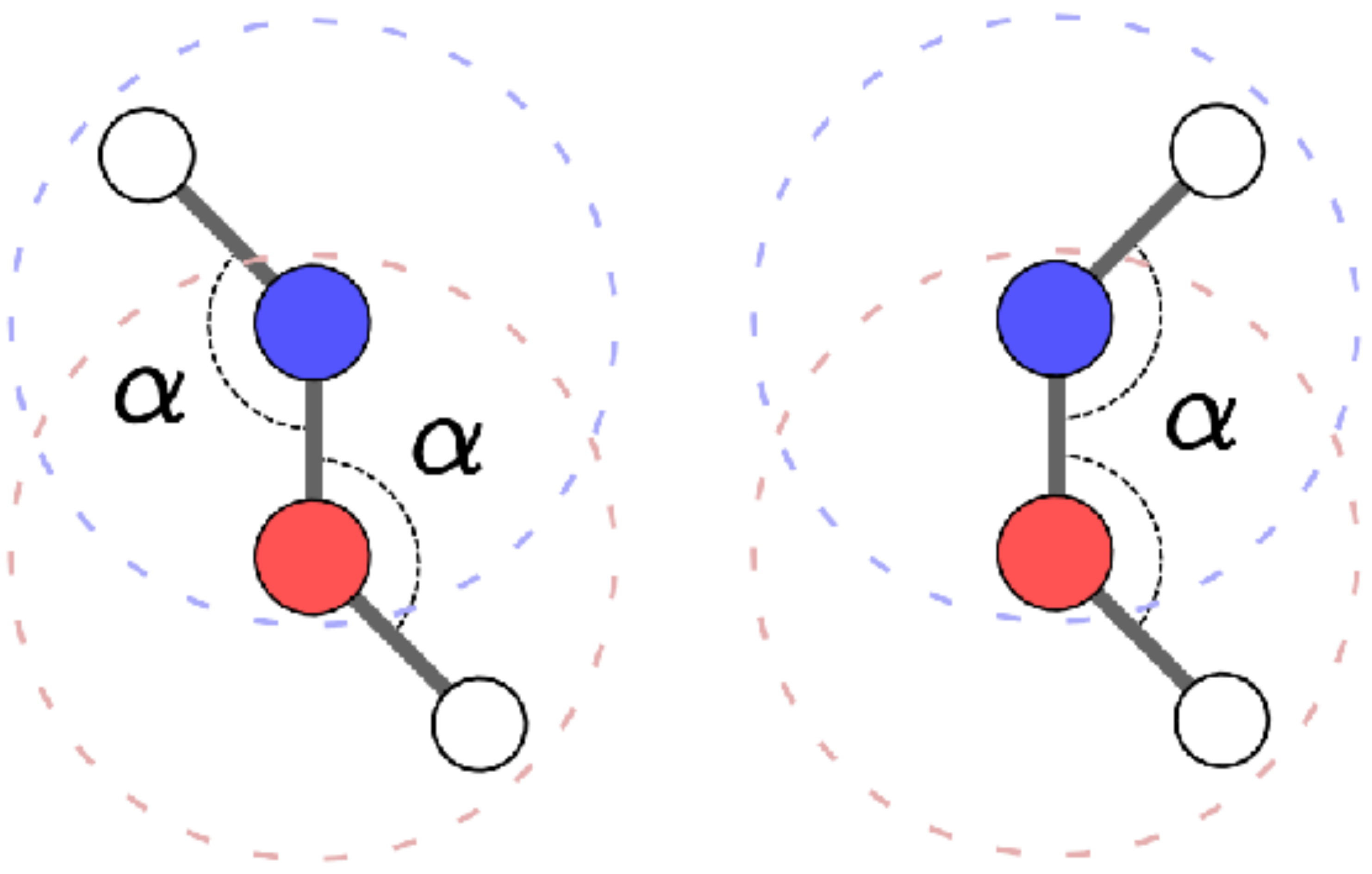
Relevant for local scalarisation in geometric GNNs — the ideal aggregator would distinguish all neighbourhoods.

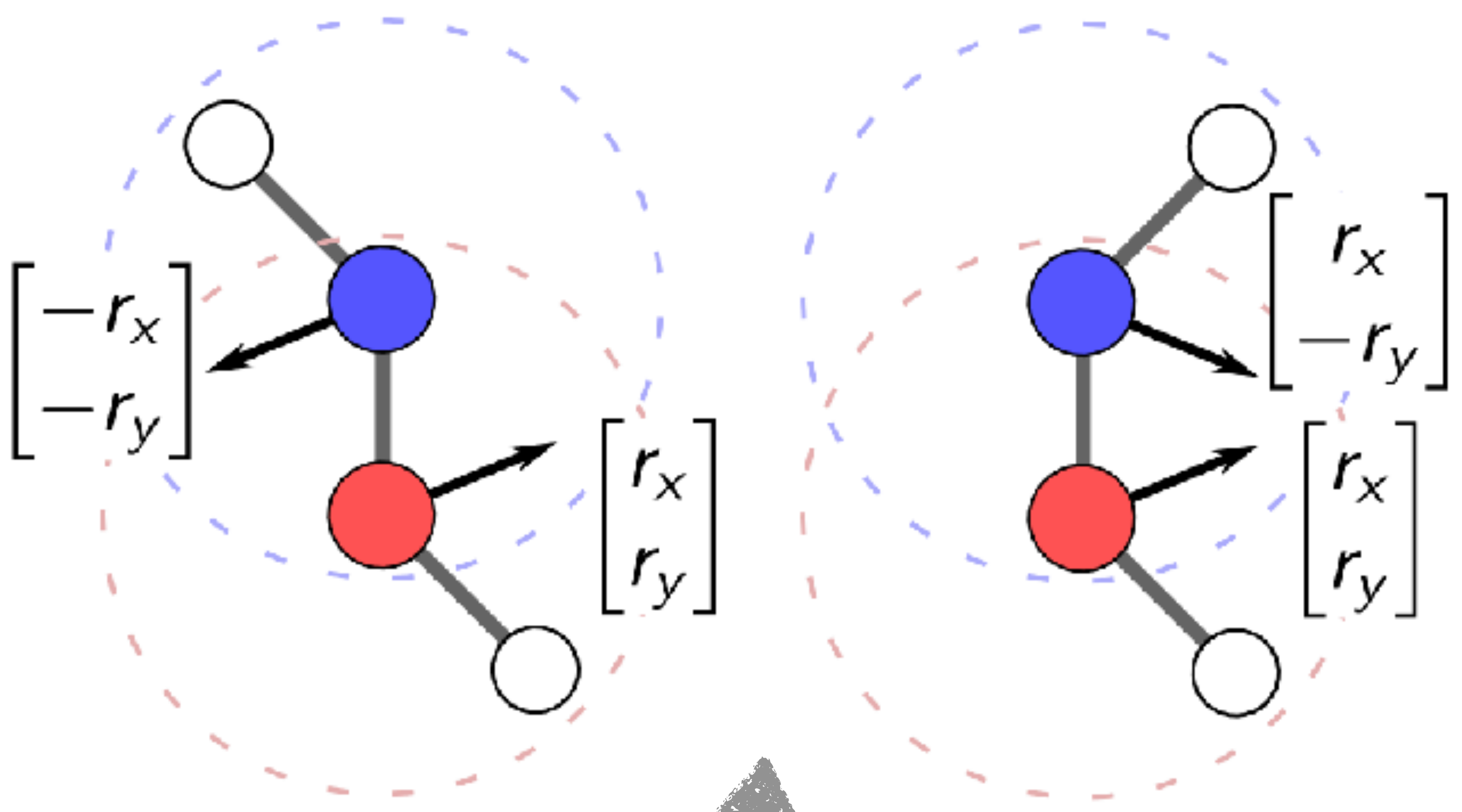
[1] Bartok et al., Phys. Rev. B, 2013. [2] Pozdynakov et al., Incompleteness of atomic structural representations, Phys. Rev. Letters, 2020.

Discriminating geometric graphs

What if all local neighbourhoods have identical invariant scalars?[1]

Pair of graphs cannot be discriminated using only scalars.





How to distinguish them?
Relative orientation of local neighbourhoods, *i.e.* geometric information.

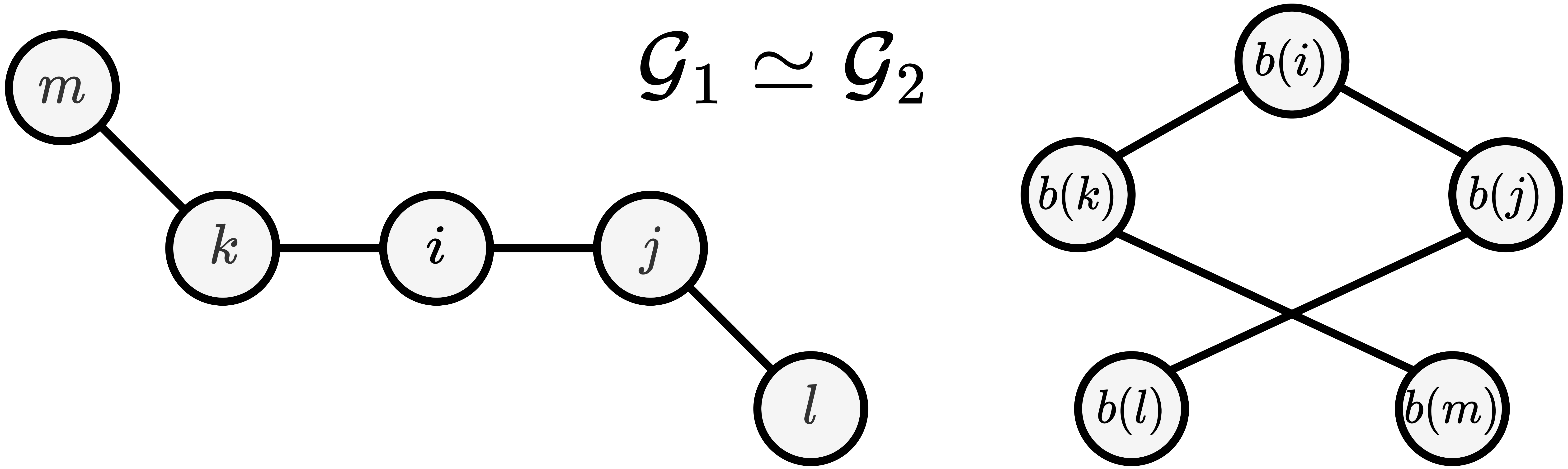
[1] Schütt et al., PaiNN, ICML, 2021.

💡Central idea:

Formalise the problem of **geometric graph isomorphism** in the context of geometric GNNs.

Recap: **Normal Graph Isomorphism**

Are two graphs the same, but 'drawn' differently?



- Two attributed graphs $\mathcal{G}, \mathcal{H}$ are isomorphic if there exists an edge-preserving bijection $b : \mathcal{V}(\mathcal{G}) \to \mathcal{V}(\mathcal{H})$ such that $s_i^{(\mathcal{G})} = s_{b(i)}^{(\mathcal{H})}$

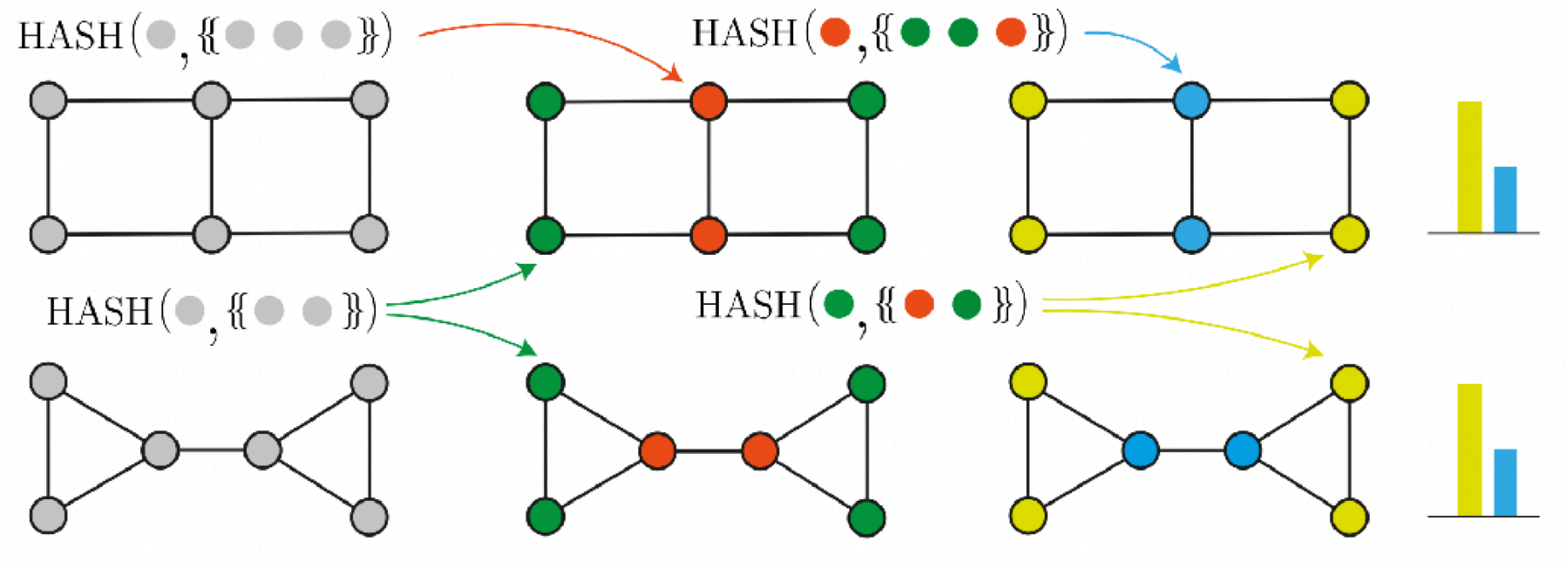
- Weisfeiler-Leman (WL) algorithm tests whether two graphs are isomorphic.

[1] Read and Corneil, The graph isomorphism disease 🤮, Journal of graph theory, 1977.

Recap: **Weisfeiler-Leman Test (WL)**

WL iteratively updates node colours via an **injective colouring function** – **unique colour** to each **unique neighbourhood pattern**.

- WL assigns a colour $c_i^{(0)} \in C$ from a countable space of colours to each node.



WL updates the node colouring by producing a new $\boldsymbol{c}_i^{(t)}$:

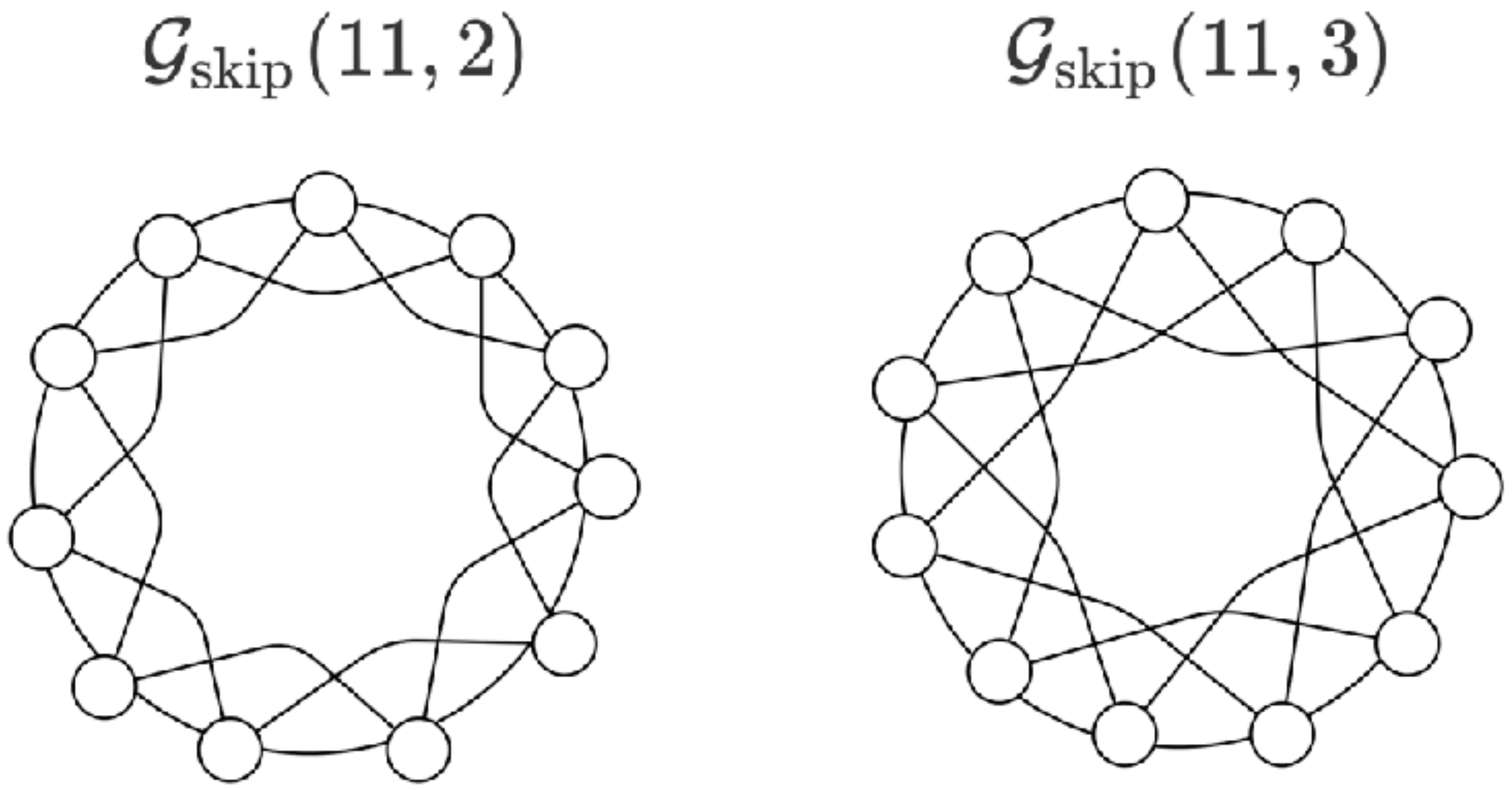
$$\boldsymbol{c}_i^{(t)} := \textsc{Hash}\left(\boldsymbol{c}_i^{(t-1)}, \{\!\{\boldsymbol{c}_j^{(t-1)} \mid j \in \mathcal{N}_i\}\!\}\right),$$

where HASH is an **injective map** that assigns a unique colour to each input.

- Given two graphs $\mathcal{G}, \mathcal{H}$, if $\{\!\{c_i^{(\mathcal{G})}\}\!\} \neq \{\!\{c_i^{(\mathcal{H})}\}\!\}$, then the graphs are **not isomorphic**.
- Otherwise, **WL cannot distinguish** the two graphs (as in this example).

Recap: **WL upper-bounds GNN expressivity**

Message passing GNNs can be **<u>at most as powerful</u>** as **WL** at **distinguishing non-isomorphic graphs**, if their aggregate-update steps are injective.



- WL has become an **abstract tool** for understanding the capabilities and **theoretical limitations** of GNNs.
- Major driver of progress towards more expressive GNNs.

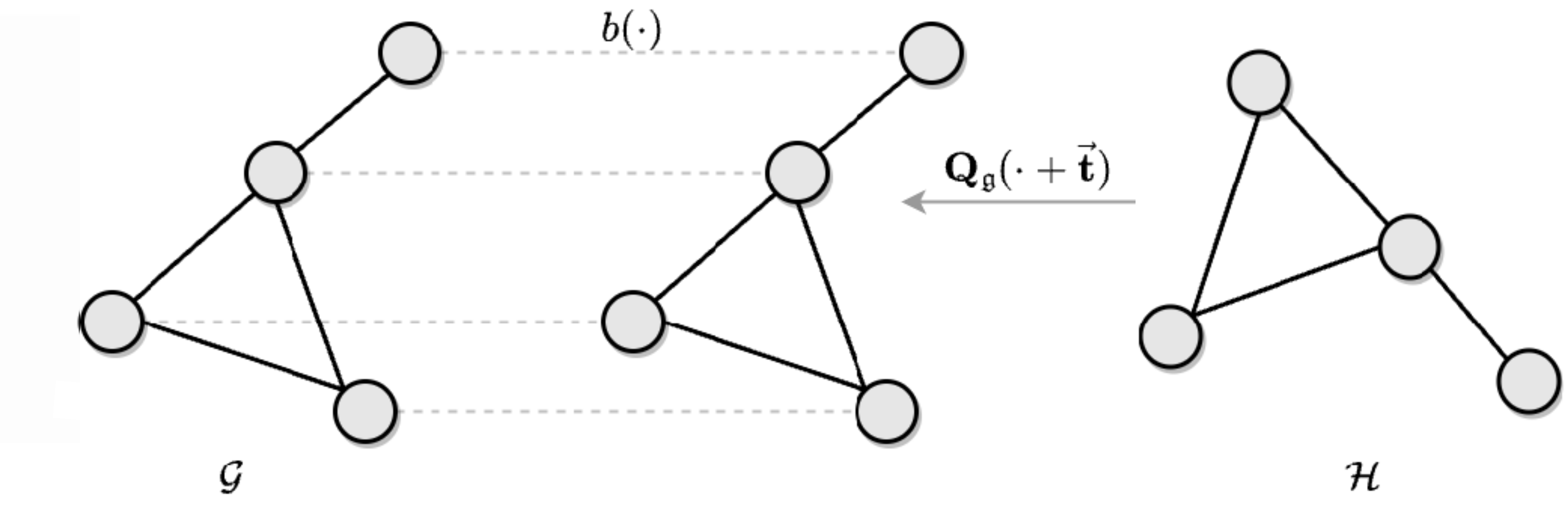
[1] Xu et al., ICLR, 2019. [2] Morris et al., AAAI, 2019. [3] Maron et al., NeurIPS, 2019. [4] Loukas, ICLR, 2020. [5] Bodnar, Frasca, et al., NeurIPS, 2021.

❗ Research gap:

Theoretical tools for normal GNNs, such as the WL framework, are **inapplicable for geometric GNNs** due to physical symmetries.

Geometric graph isomorphism

$\mathcal{G}, \mathcal{H}$ **are geometrically isomorphic if:**



- there exists an attributed **graph isomorphism** $b : \mathcal{V}(\mathcal{G}) \to \mathcal{V}(\mathcal{H})$
- …s.t. **geometric attributes are equivalent** for all nodes, up to some **rotation** $Q_{\mathfrak{g}} \in \mathfrak{G}$ and some **translation** $\vec{t} \in T(d)$:

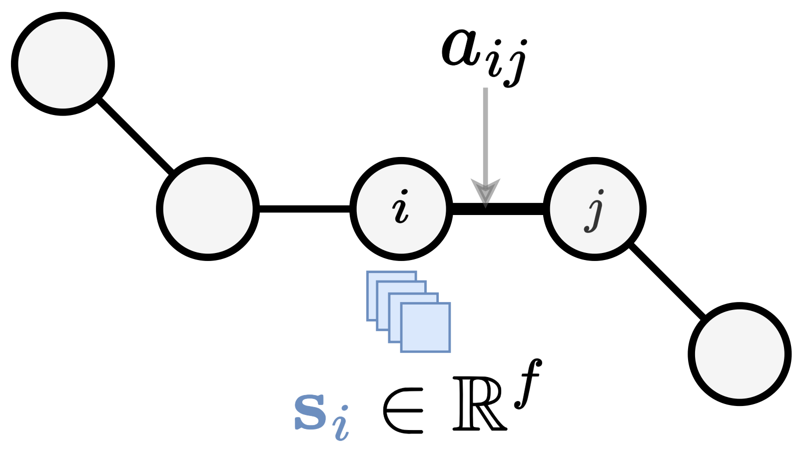
$$\left(\boldsymbol{s}_i^{(\mathcal{G})}, \vec{\boldsymbol{v}}_i^{(\mathcal{G})}, \vec{\boldsymbol{x}}_i^{(\mathcal{G})} \right) = \left(\boldsymbol{s}_{b(i)}^{(\mathcal{H})},\ \boldsymbol{Q}_{\mathfrak{g}} \vec{\boldsymbol{v}}_{b(i)}^{(\mathcal{H})},\ \boldsymbol{Q}_{\mathfrak{g}} (\vec{\boldsymbol{x}}_{b(i)}^{(\mathcal{H})} + \vec{\boldsymbol{t}}) \right)$$

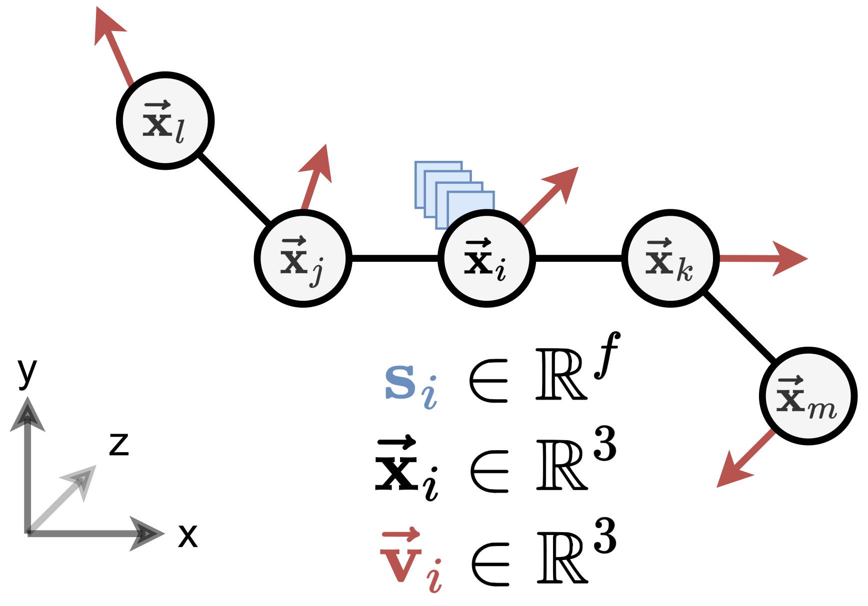
Geometric Weisfeiler-Leman Test

Theoretical upper bound on Geometric GNN expressivity

Intuition: **generalising WL to geometric graphs**

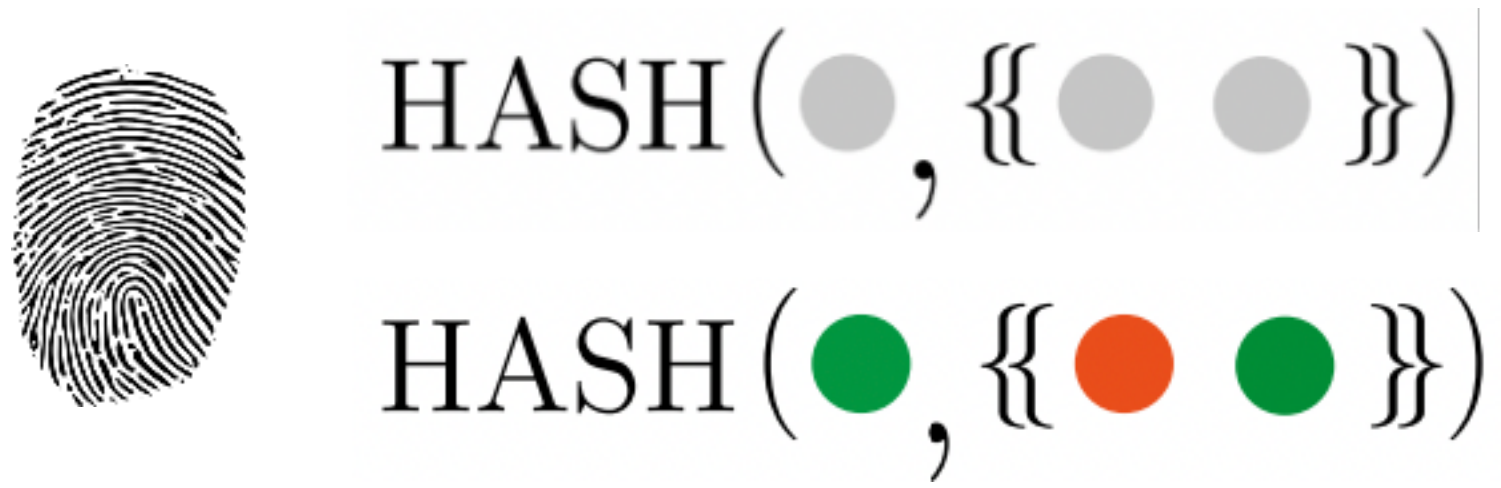
Key property: **node-centric** procedure, **injective aggregation** from local neighbours





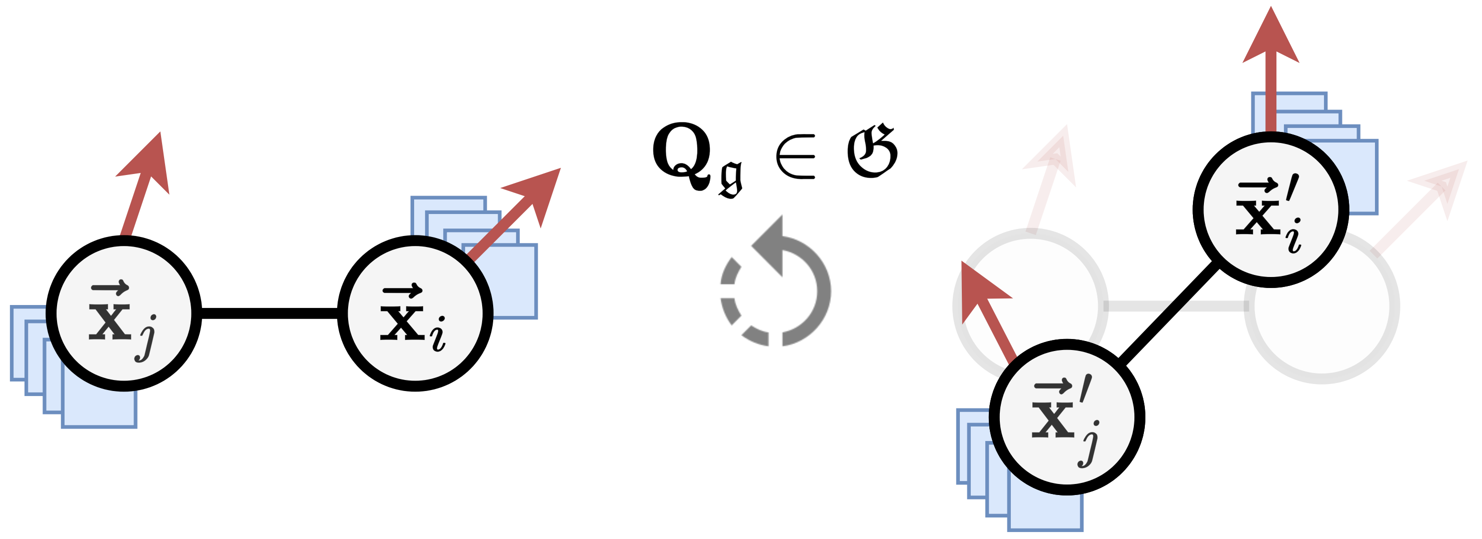
Standard WL

- **Neighbourhood:** set of **invariant scalar** features.
- **Node colouring:** unique for every neighbourhood type, i.e. (central node, neighbourhood) pattern.



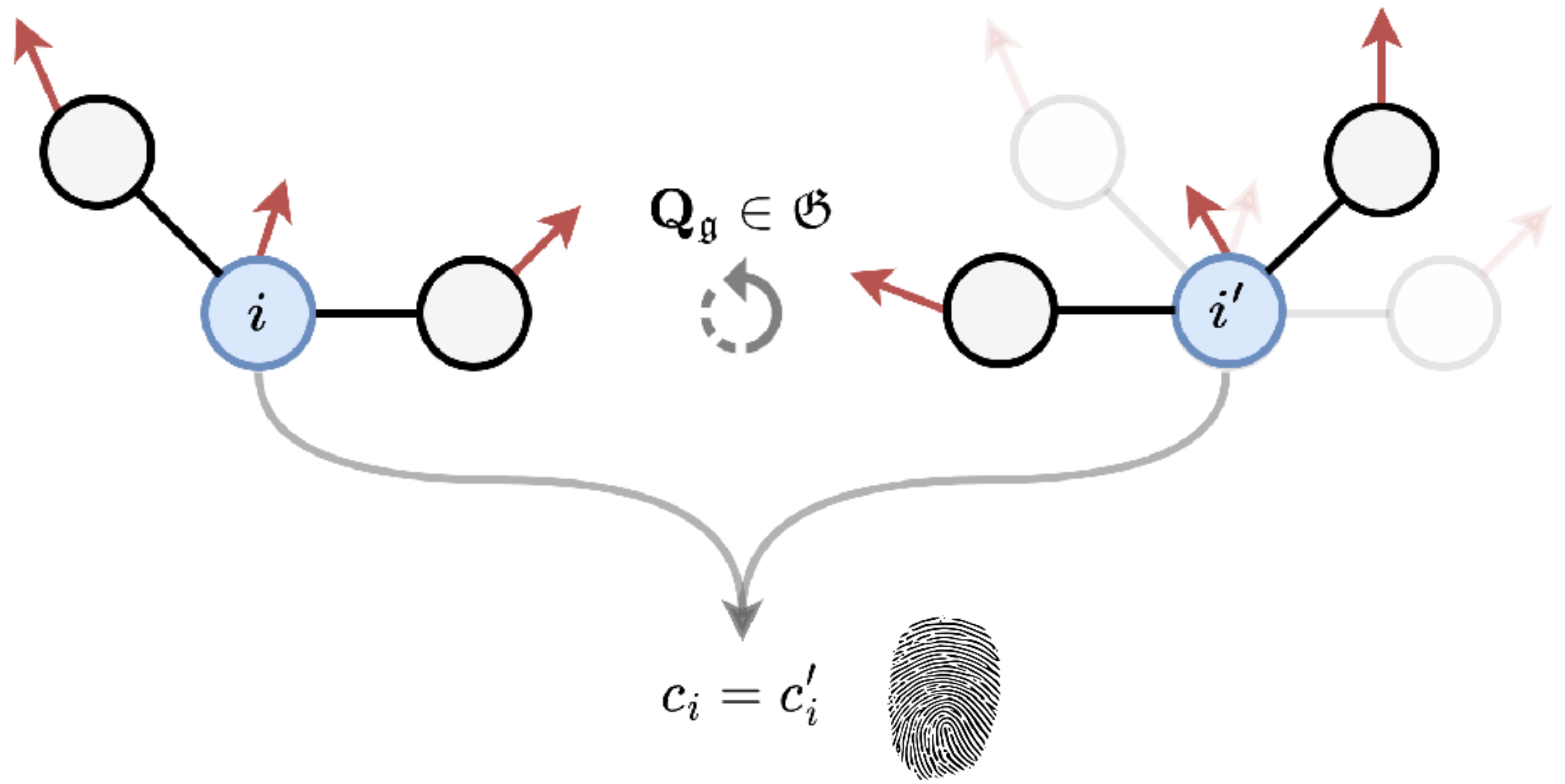
Geometric WL

- **Neighbourhood:** set of **invariant** + **equivariant geometric features**.
- **Node colouring:** unique for every neighbourhood type i.e. (central node, neighbourhood) pattern.
- **Geometric information:** how that neighbourhood type is oriented/rotated in space.



GWL Property #1: **Orbit injectivity of colours**

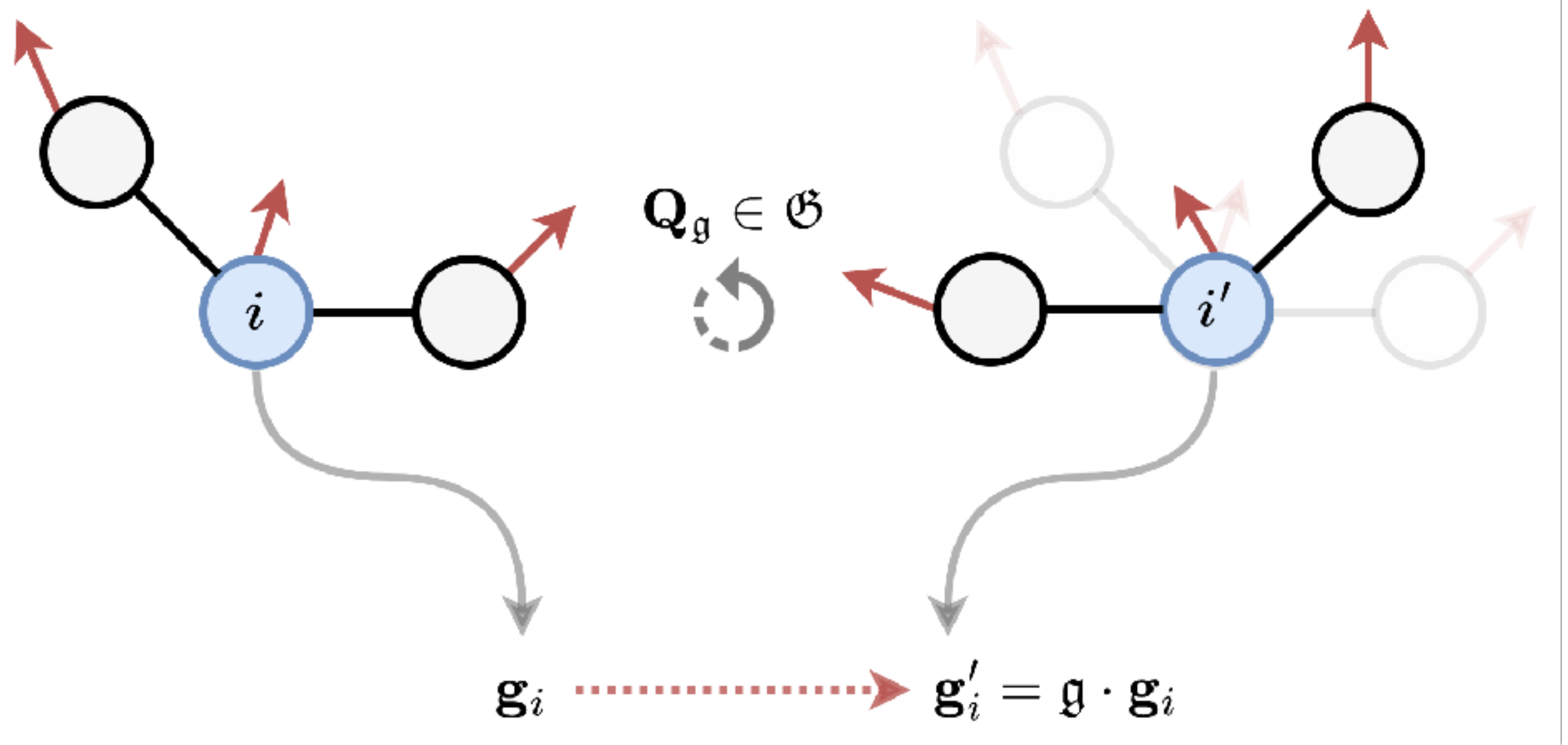
If two neighbourhoods are the **same up to rotations**, their **colours should be the same**, i.e. the colouring must be **$\mathfrak{G}$-orbit injective**.



The $\mathfrak{G}$-orbit injective colouring function is also $\mathfrak{G}$-invariant by definition.

GWL Property #2: **Preservation of local geometry**

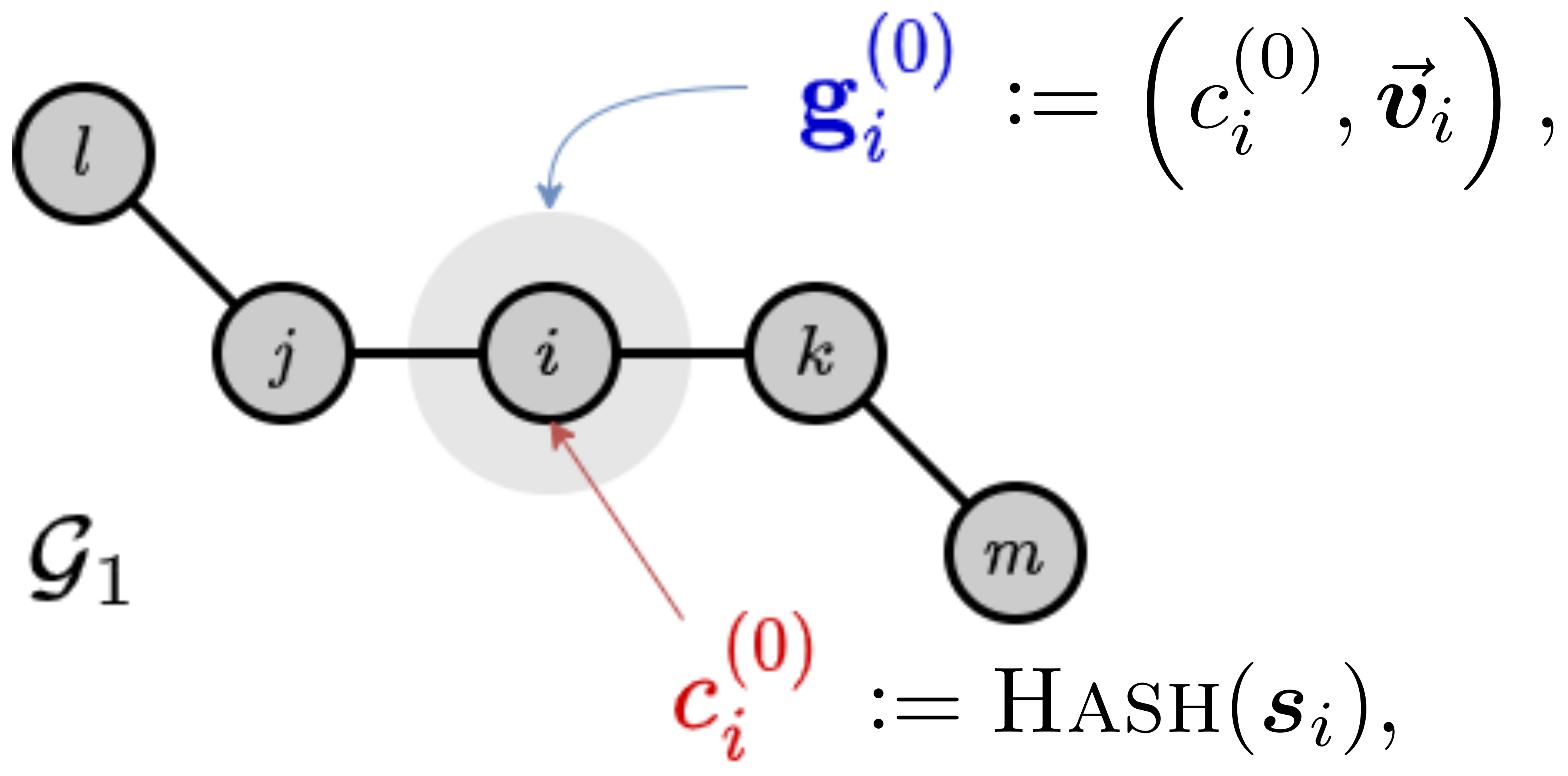
- Satisfying **Property #1** (orbit injectivity) will by definition **lose orientation information** — this is **no longer injective**, unlike WL.
- Thus, we must update auxiliary **geometric information variables** g_i in a way that is **<u>injective</u>** and **<u>$\mathfrak{G}$-equivariant</u>**.



GWL Step 0: **Initialisation Step**

We assign to each node:

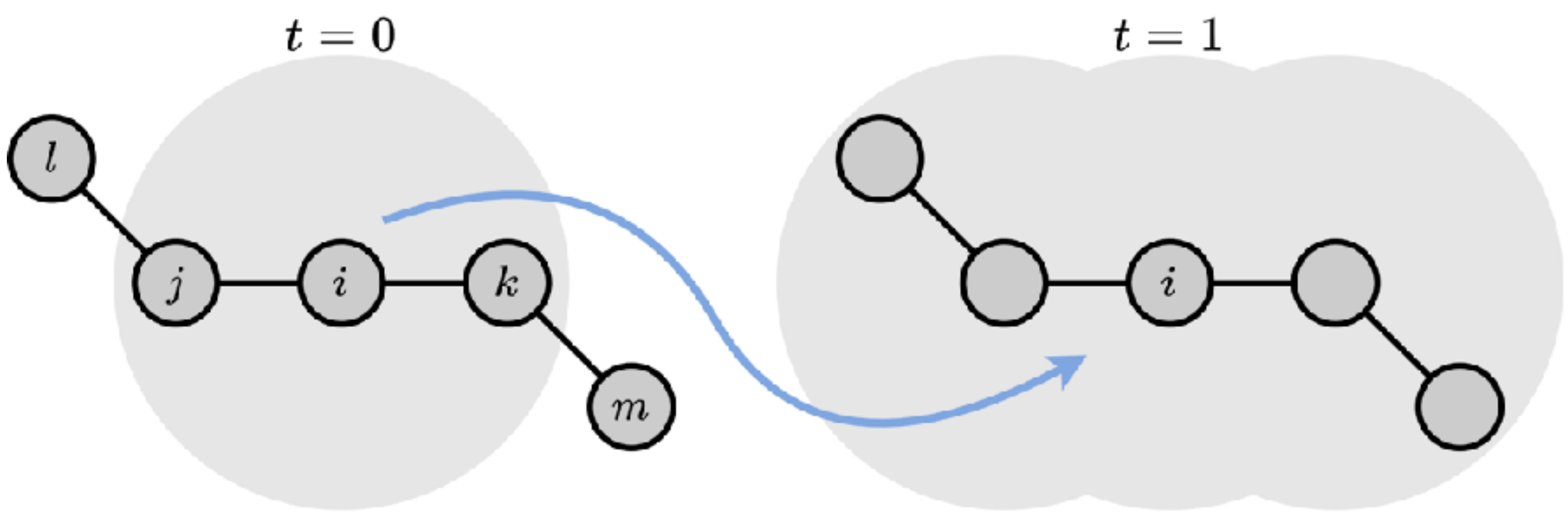
- **a scalar node colour** $c_i \in C'$
- **an auxiliary object** g_i **for geometric information associated with the sub-graph around each node** i

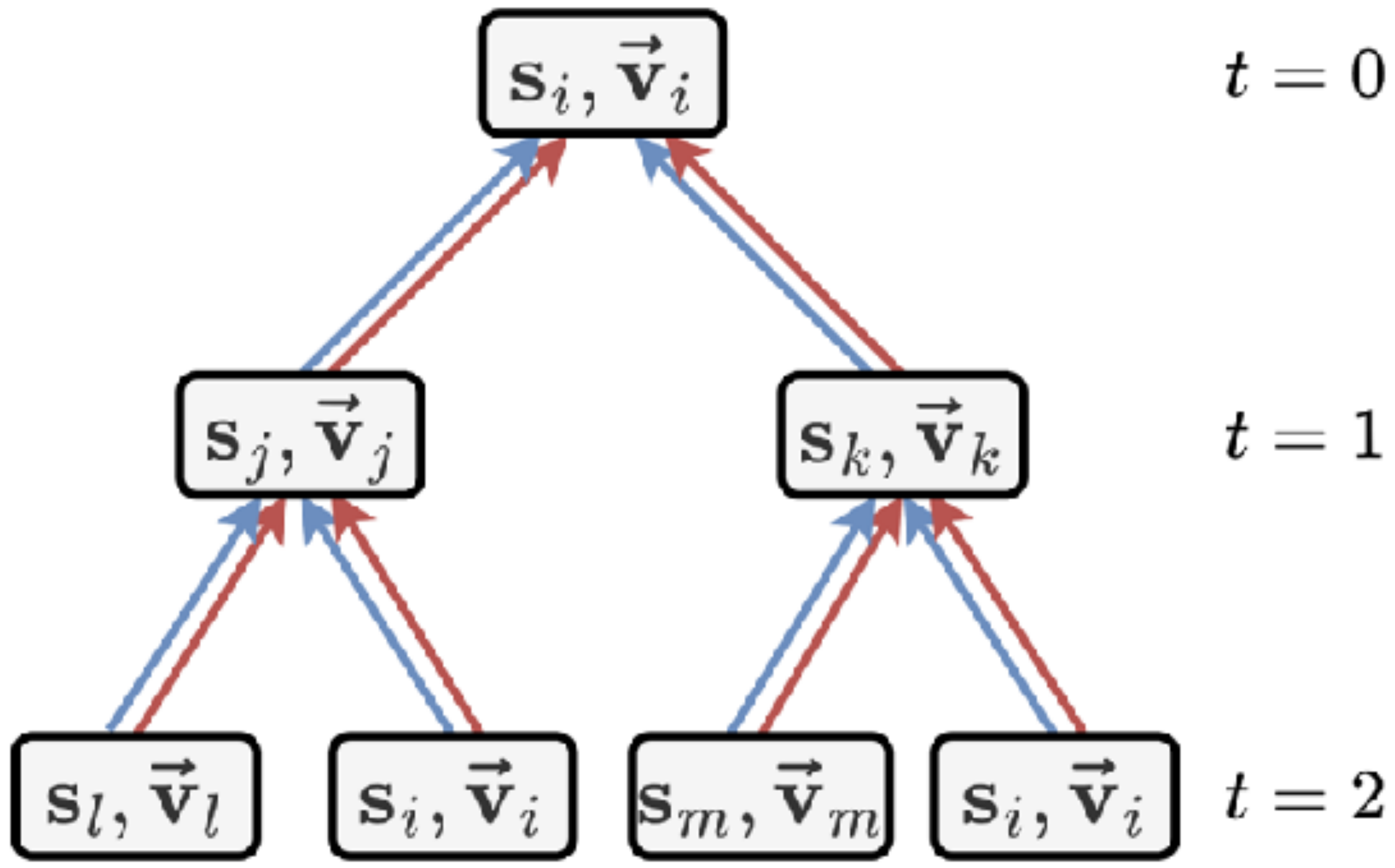


GWL Step 1: **Aggregate local information**

Copy-paste aggregation: At each iteration, **aggregate the geometric information around node *i*** into a **new (nested) object $g_i^{(t)}$:**

$$\boldsymbol{g}_i^{(t)} := \left(\left(c_i^{(t-1)}, \boldsymbol{g}_i^{(t-1)}\right) \, , \, \{\!\!\{ \left(c_j^{(t-1)}, \boldsymbol{g}_j^{(t-1)}\right) \mid j \in \mathcal{N}_i \}\!\!\} \right) ,$$



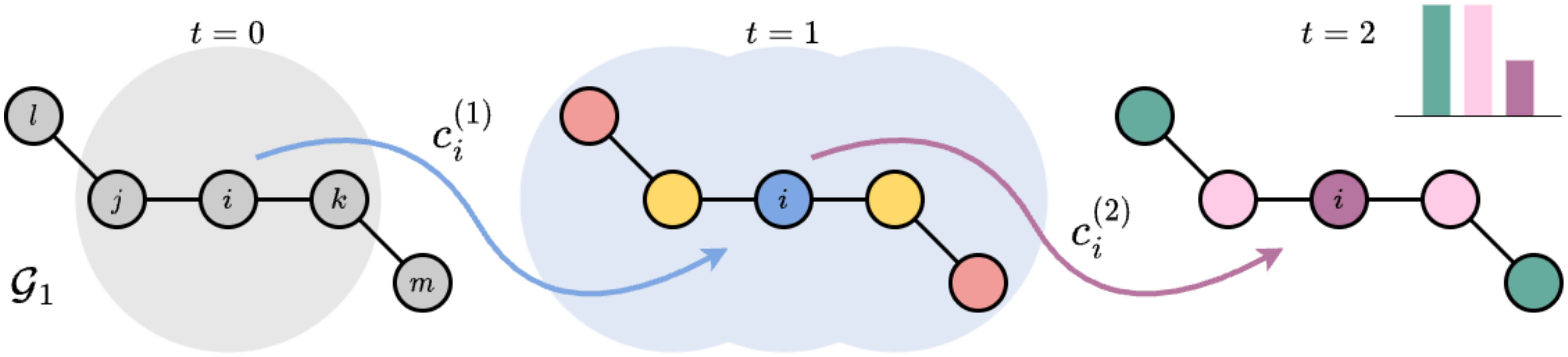


- This nested aggregation is **injective** and **$\mathfrak{G}$-equivariant**
- Each iteration can progressively expand $g_i^{(t)}$ to **larger *t*-hop subgraphs** $\mathcal{N}_i^{(t)}$

GWL Step 2: **Update node colouring**

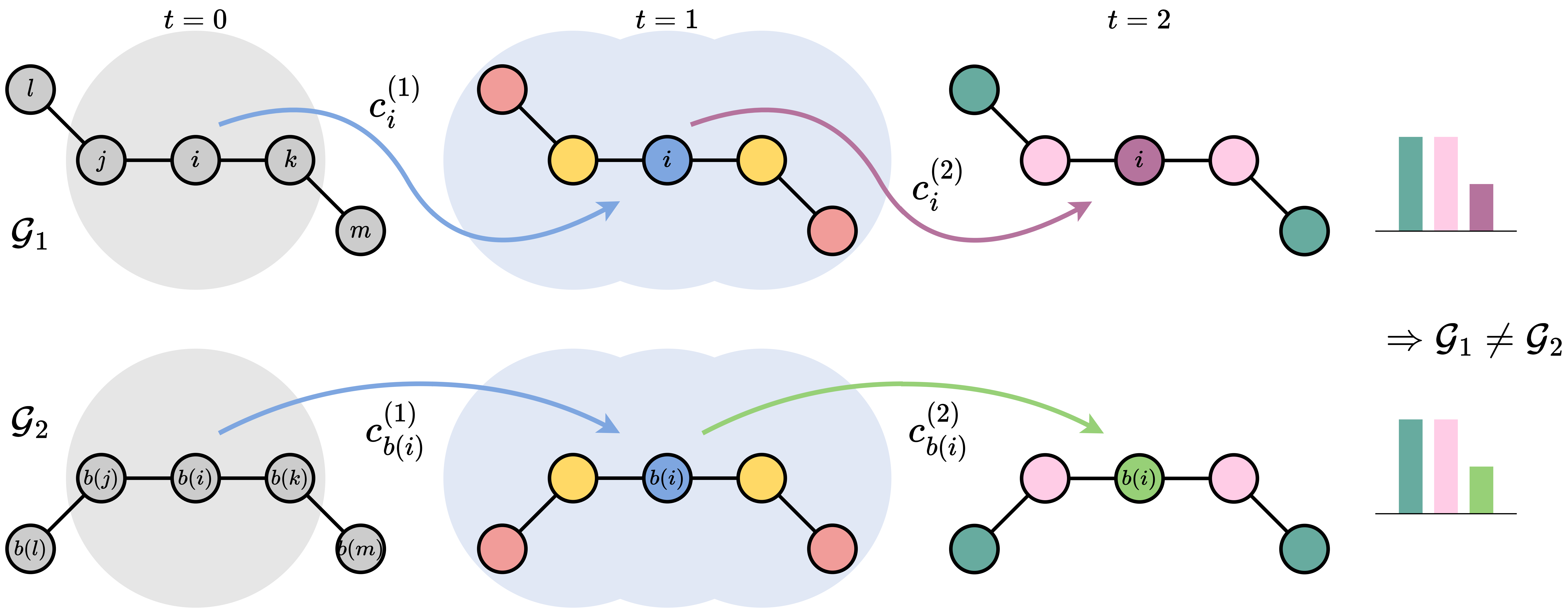
Node colouring $c_i^{(t)}$ summarises the information in $g_i^{(t)}$ by using $\mathfrak{G}$-orbit injective and $\mathfrak{G}$-invariant colouring function (I-HASH):

$$c_i^{(t)} := \text{I-}\textsc{Hash}\left(\boldsymbol{g}_i^{(t)}\right),$$



In geometric GNNs, I-HASH corresponds to **scalarisation** from subsets of neighbours (body order).

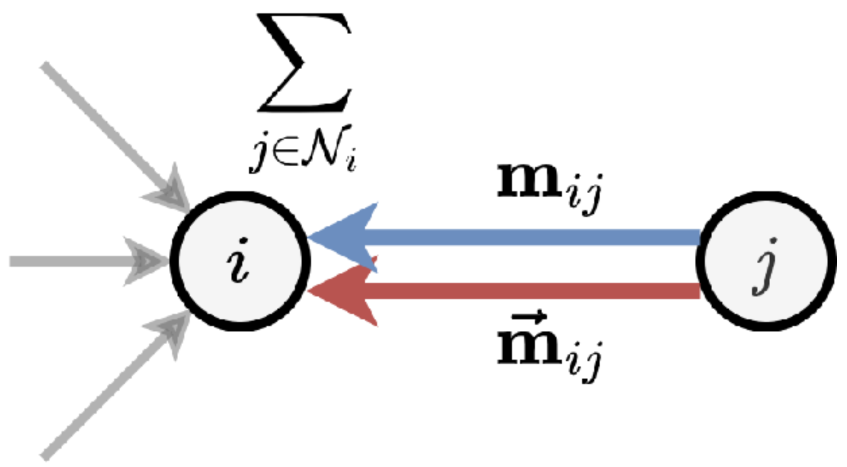
GWL Step 3: **Termination upon stable colouring**



Given geometric graphs $\mathcal{G}, \mathcal{H}$, if $\{\!\{c_i^{(\mathcal{G})}\}\!\} \neq \{\!\{c_i^{(\mathcal{H})}\}\!\}$, then **they are not isomorphic**. Otherwise, GWL cannot distinguish them.

Upper bounding geometric GNN expressivity

Equivariant GNNs can be at most as powerful as GWL in distinguishing non-isomorphic geometric graphs.



$$\boldsymbol{m}_i^{(t)}, \vec{\boldsymbol{m}}_i^{(t)} := \textsc{Agg}\left(\{\!\{(\boldsymbol{s}_i^{(t)}, \boldsymbol{s}_j^{(t)}, \vec{\boldsymbol{v}}_i^{(t)}, \vec{\boldsymbol{v}}_j^{(t)}, \vec{\boldsymbol{x}}_{ij}) \mid j \in \mathcal{N}_i\}\!\}\right)$$

$$\boldsymbol{s}_i^{(t+1)}, \vec{\boldsymbol{v}}_i^{(t+1)} := \textsc{Upd}\left((\boldsymbol{s}_i^{(t)}, \vec{\boldsymbol{v}}_i^{(t)}) \ , \ (\boldsymbol{m}_i^{(t)}, \vec{\boldsymbol{m}}_i^{(t)})\right)$$

$$c_i^{(0)} := \textsc{Hash}(\boldsymbol{s}_i), \qquad \boldsymbol{g}_i^{(0)} := \left(c_i^{(0)}, \vec{\boldsymbol{v}}_i\right),$$

$$\boldsymbol{g}_i^{(t)} := \left((c_i^{(t-1)}, \boldsymbol{g}_i^{(t-1)}) \ , \ \{\!\{(c_j^{(t-1)}, \boldsymbol{g}_j^{(t-1)}) \mid j \in \mathcal{N}_i\}\!\}\right),$$

$$c_i^{(t)} := \text{I-}\textsc{Hash}\left(\boldsymbol{g}_i^{(t)}\right),$$

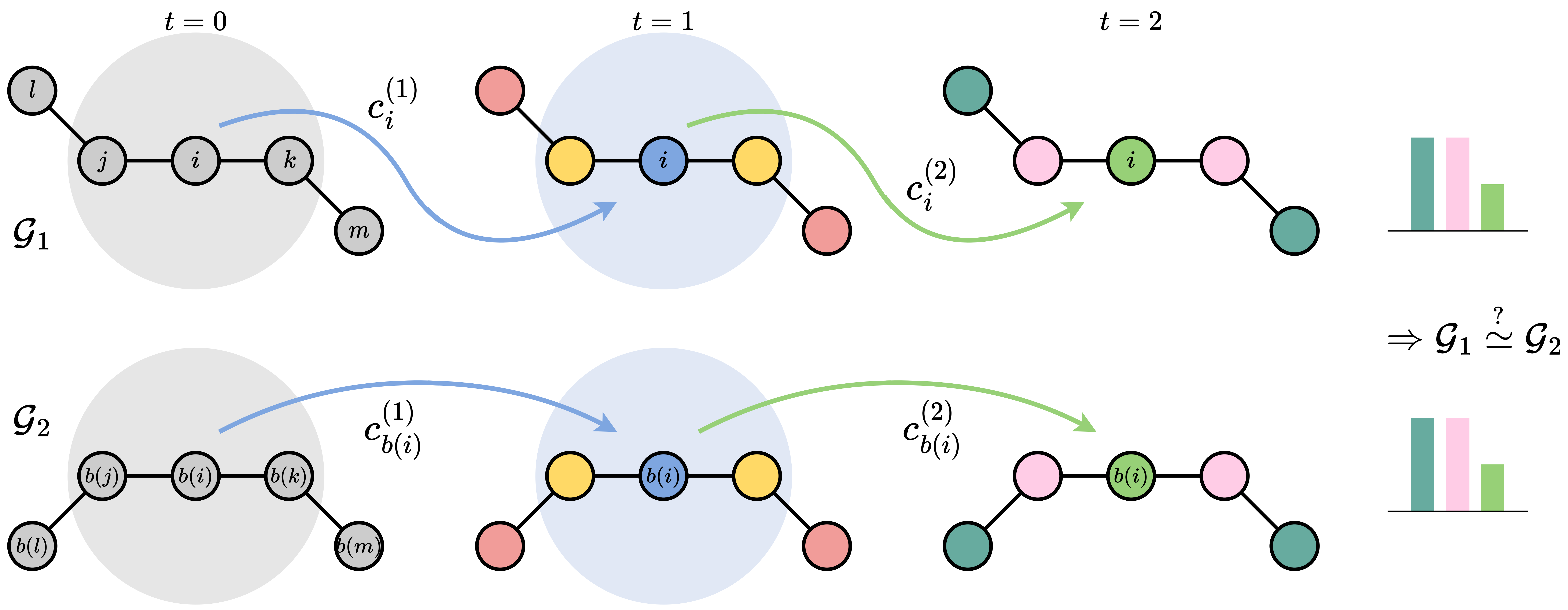
…and equivariant GNNs have the **same expressive power** as GWL if equipped with **injective** aggregation and injective/orbit injective update functions.

Invariant version of GWL

IGWL is a restricted version of GWL which

- **only updates node colours using orbit injective I-HASH function**
- **does not propagate geometric information**

$$c_i^{(t)} := \text{I-HASH}\left((c_i^{(t-1)}, \vec{\boldsymbol{v}}_i) \ , \ \{\!\{ (c_j^{(t-1)}, \vec{\boldsymbol{v}}_j) \mid j \in \mathcal{N}_i \}\!\} \right)$$

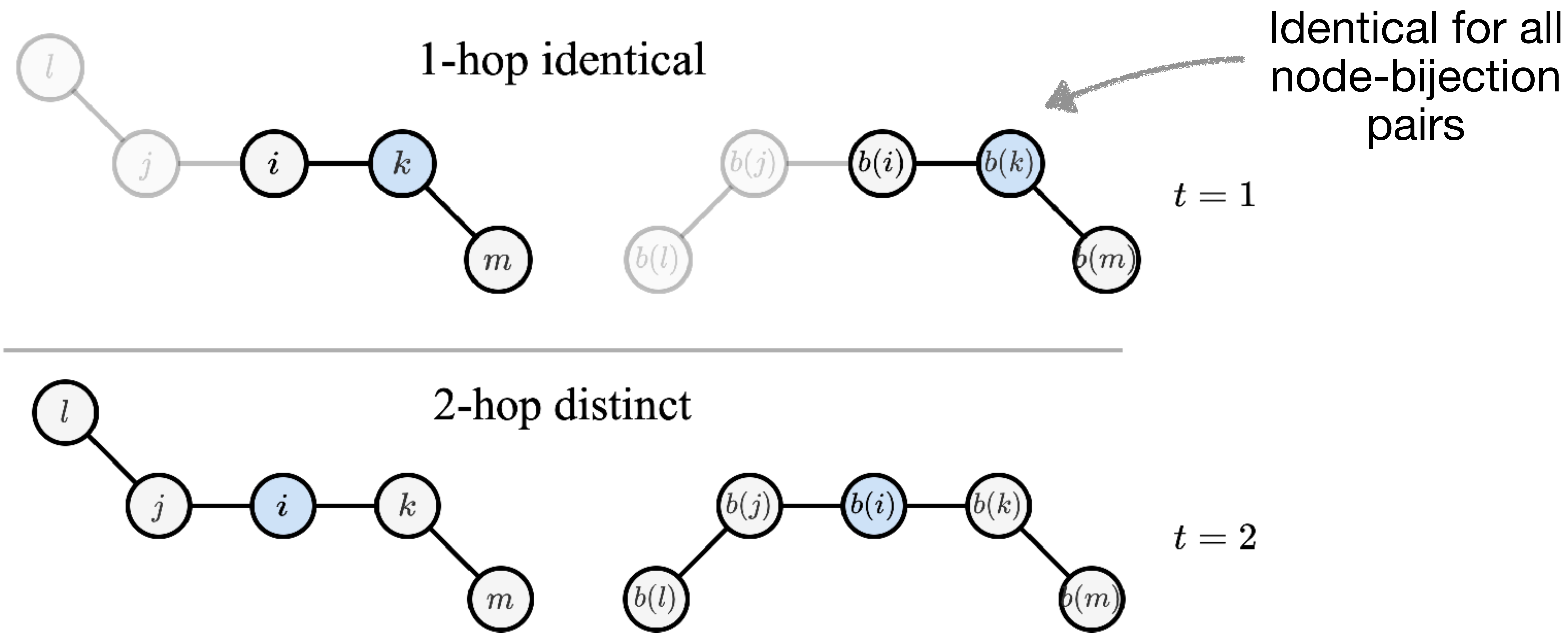


Role of depth in geometric GNNs

Propagating geometric information

k-hop distinct and identical geometric graphs

Consider two geometric graphs such that the underlying attributed graphs are isomorphic:

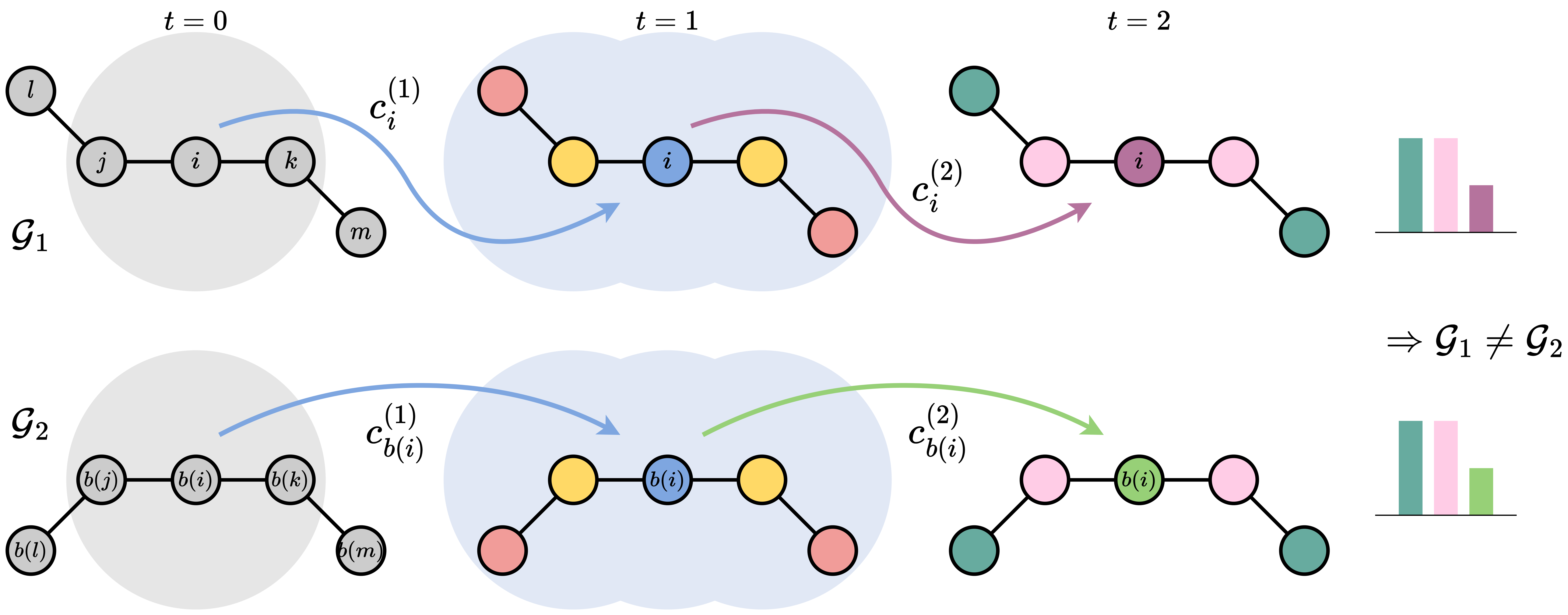


* We also considers the general case without an attributed graph isomorphism in the full paper.

Characterising what GWL <u>can</u> distinguish

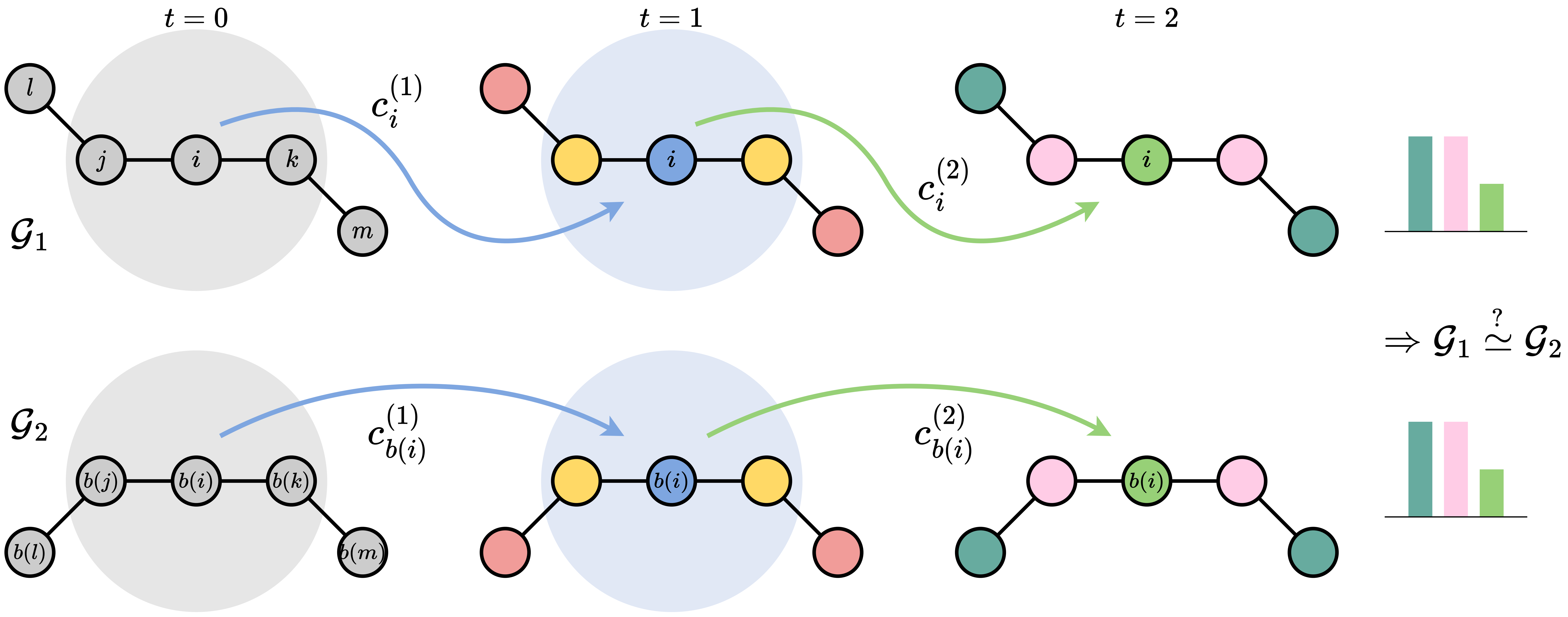
GWL can distinguish

- **any k-hop distinct geometric graphs**
- **k iterations are sufficient**



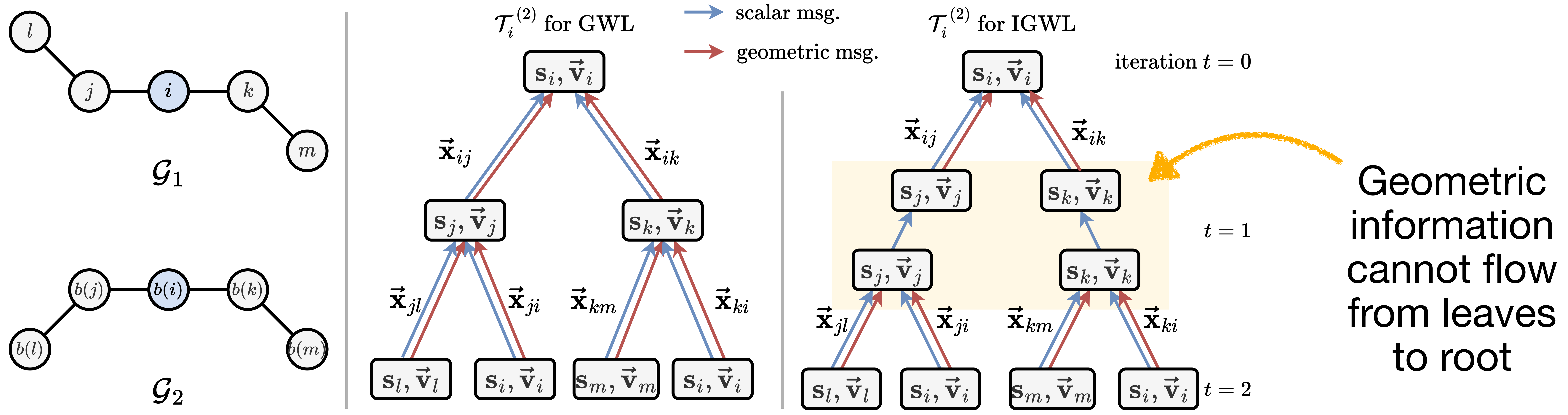
Characterising what IGWL cannot distinguish

Any number of iterations of IGWL cannot distinguish any 1-hop identical geometric graphs



Comparing the expressivity of GWL & IGWL

GWL is strictly more powerful than IGWL, as GWL can distinguish a broader class of geometric graphs.



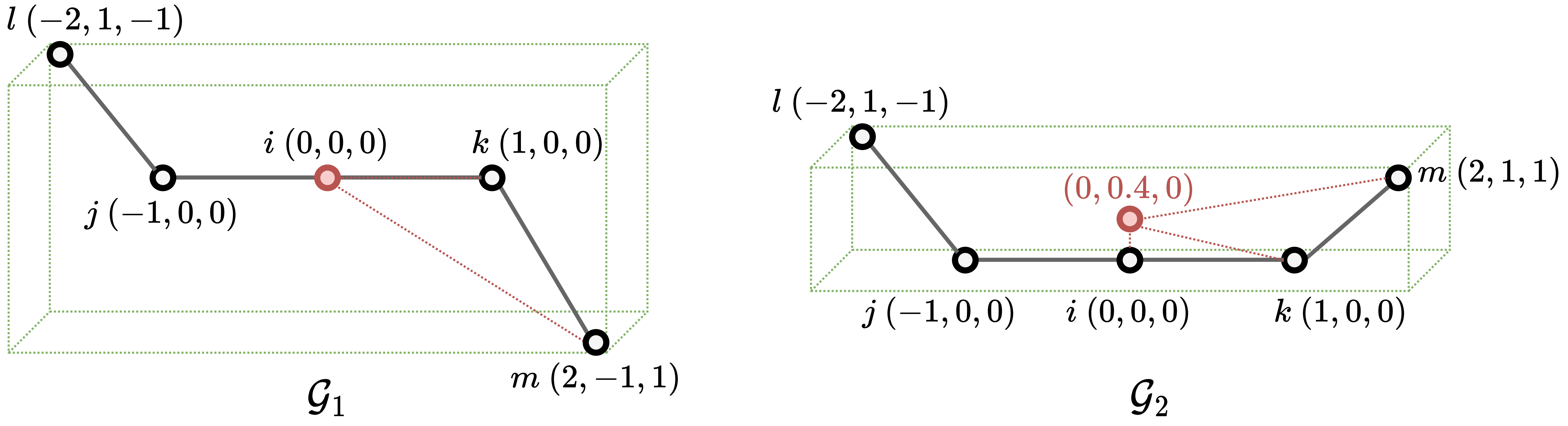
IGWL and invariant GNNs fail to understand how various
1-hop neighbourhoods in a graph are **oriented w.r.t. each other**.

Limitations of invariant message passing

Failure to capture global geometry

Invariant GNNs fail for non-local geometric properties

IGWL and invariant GNNs cannot decide:[4]
(1) area, volume of bounding box/sphere; (2) distance from centroid; and (3) dihedral angles.

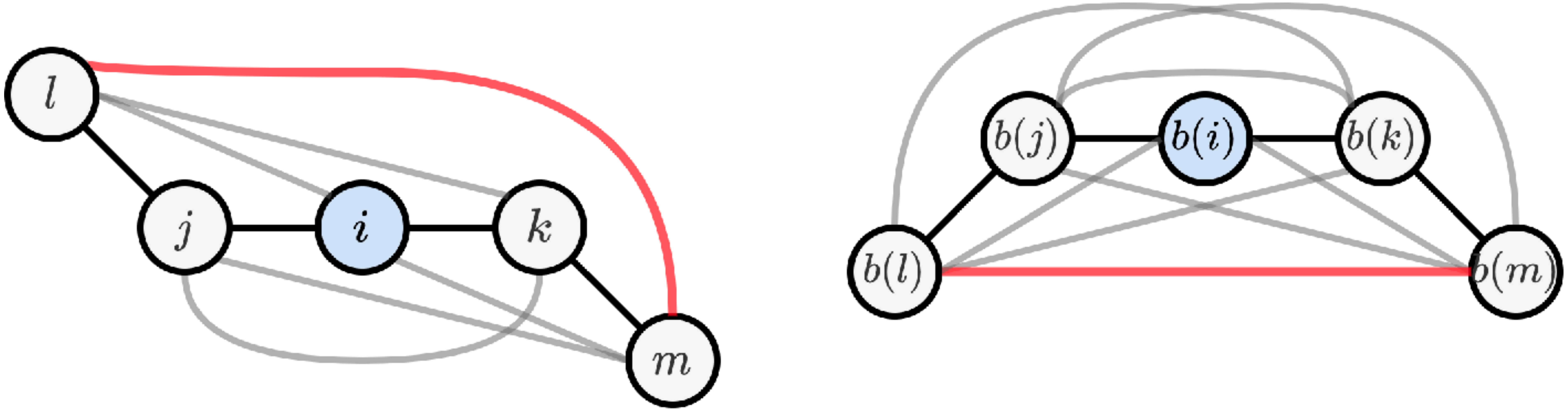


How to overcome these limitations?
Pre-computing non-local geometric properties as input features[1][2][3]

[1] Gasteiger et al., GemNet, NeurIPS, 2021. [2] Liu et al., SphereNet, ICLR, 2022. [3] Wang et al., ComENet, NeurIPS, 2022. [4] Garg et al., ICML, 2020.

When is invariance ‘all you need’?

IGWL has the same expressive power as GWL for fully connected geometric graphs, i.e. point clouds.



Supported by the empirical success of geometric ‘**graph Transformers**’[1][2]

[1] Joshi, Transformers are GNNs, The Gradient, 2020. [2] Shi et al., Benchmarking Graphormer, 2022.

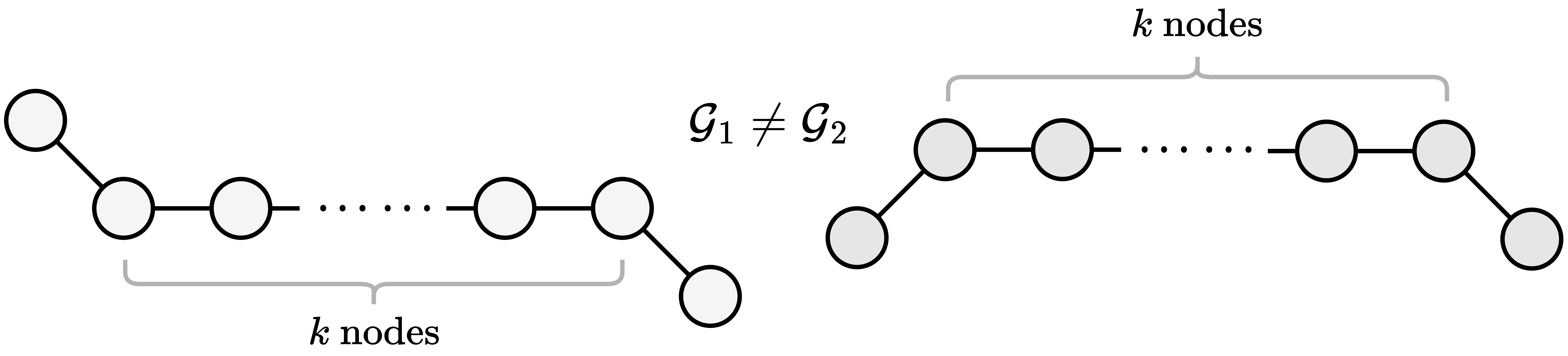
Synthetic experiments on Geometric GNN expressivity

Code + Geometric GNN 101 tutorial:

github.com/chaitjo/geometric-gnn-dojo

Experiment 1: **Depth, non-local properties, & oversquashing**

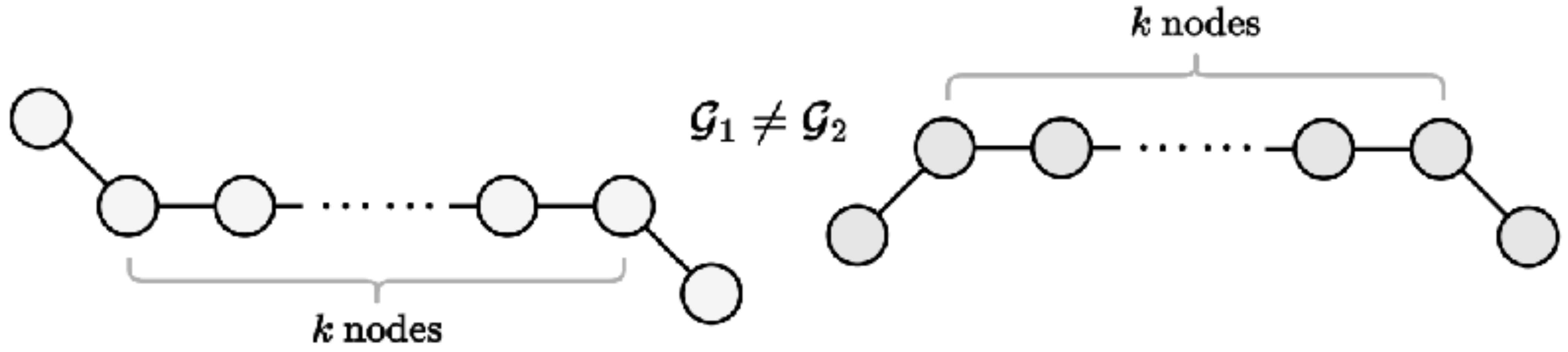
(Theory) **GWL: perfectly propagate geometric information with each iteration.**
(Practice) **Geometric GNNs: stacking layers may distort distant information?**



- ***k*-chain graphs[1]:** $\left(\left\lfloor \frac{k}{2} \right\rfloor + 1\right)$-hop distinguishable geometric graphs — Thus, $\left(\left\lfloor \frac{k}{2} \right\rfloor + 1\right)$ GWL iterations are theoretically sufficient to distinguish them.
- We train geometric GNNs with increasing #layers to distinguish k-chains.

[1] Generalisation of the example from Schütt et al., ICML, 2021.

Experiment 1: **Depth, non-local properties, & oversquashing**

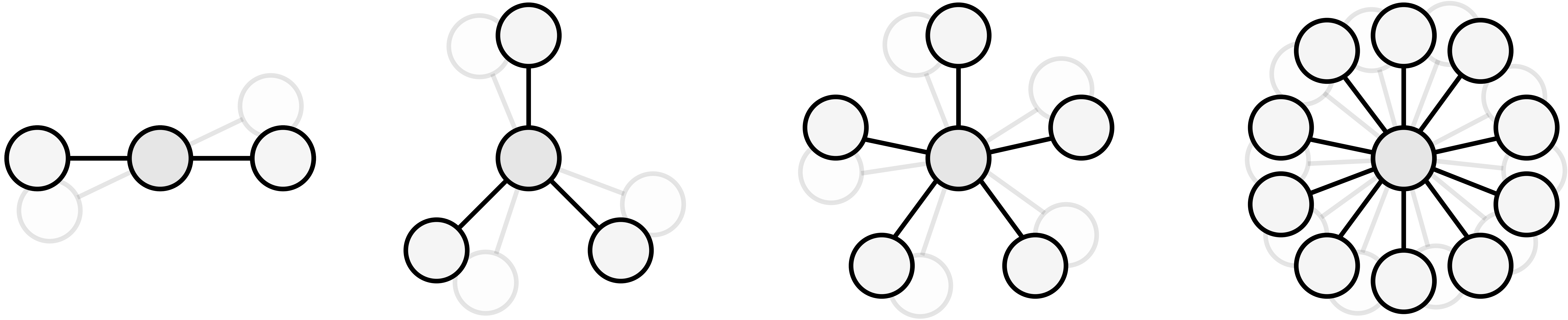


	(k = 4-chains) GNN Layer	Number of layers $\lfloor \frac{k}{2} \rfloor$	$\lfloor \frac{k}{2} \rfloor + 1 = \mathbf{3}$	$\lfloor \frac{k}{2} \rfloor + 2$	$\lfloor \frac{k}{2} \rfloor + 3$	$\lfloor \frac{k}{2} \rfloor + 4$
Inv.	IGWL	50%	50%	50%	50%	50%
	SchNet	50.0 ± 0.00	50.0 ± 0.00	50.0 ± 0.00	50.0 ± 0.00	50.0 ± 0.00
	DimeNet	50.0 ± 0.00	50.0 ± 0.00	50.0 ± 0.00	50.0 ± 0.00	50.0 ± 0.00
Equiv.	GWL	50%	**100%**	**100%**	**100%**	**100%**
	E-GNN	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0	**100.0 ± 0.0**
	GVP-GNN	50.0 ± 0.0	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**
	TFN	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0	**80.0 ± 24.5**	**85.0 ± 22.9**
	MACE	50.0 ± 0.0	**90.0 ± 20.0**	**90.0 ± 20.0**	**95.0 ± 15.0**	**95.0 ± 15.0**

- Invariant GNNs are **unable** to distinguish k-chains (as expected).
- Equivariant GNNs may require **more iterations that prescribed** by GWL — preliminary evidence of **oversquashing of geometric information** across multiple layers.

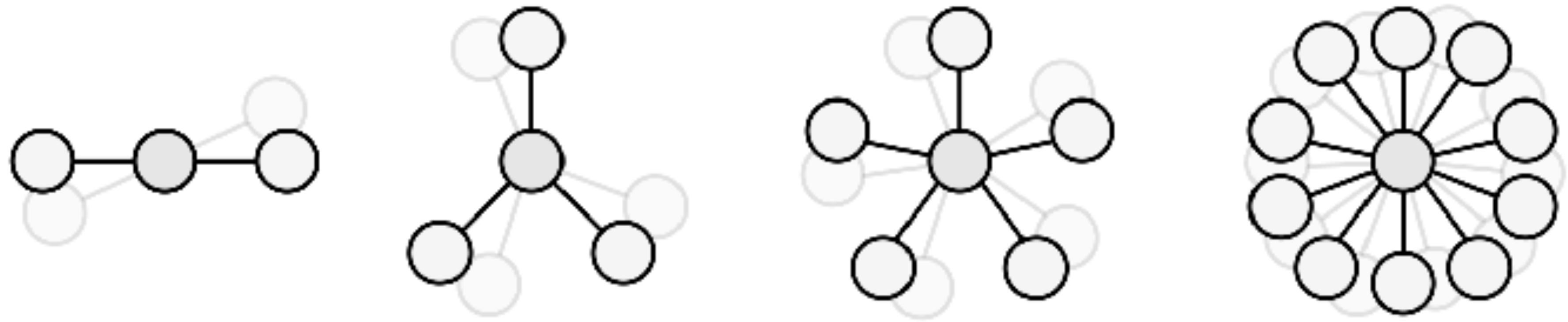
Experiment 2: **Higher order tensors & rotational symmetry**

(Theory) **GWL: perfectly aggregates equivariant geometric information via copying.**
(Practice) **Geometric GNNs: tradeoffs for cartesian vs. spherical, and tensor rank?**



- **L-fold symmetric structure:** does not change when rotated by an angle $\frac{2\pi}{L}$ around a point (in 2D) or axis (3D).
- We consider **two distinct rotated versions** of each L-fold symmetric structure and train single layer equivariant GNNs to identify the two orientations.

Experiment 2: **Higher order tensors & rotational symmetry**

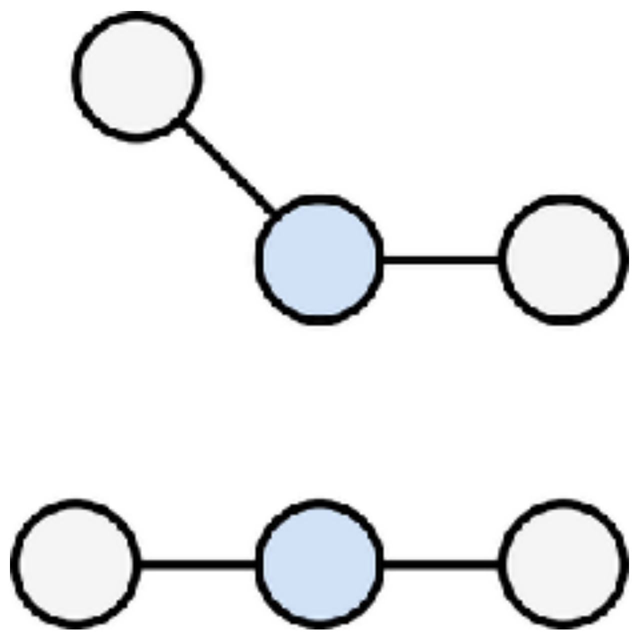


	GNN Layer	Rotational symmetry 2 fold	3 fold	5 fold	10 fold
Cart.	E-GNN$_{L=1}$	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
	GVP-GNN$_{L=1}$	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
Spherical	TFN/MACE$_{L=1}$	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
	TFN/MACE$_{L=2}$	**100.0 ± 0.0**	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
	TFN/MACE$_{L=3}$	**100.0 ± 0.0**	**100.0 ± 0.0**	50.0 ± 0.0	50.0 ± 0.0
	TFN/MACE$_{L=5}$	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	50.0 ± 0.0
	TFN/MACE$_{L=10}$	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**

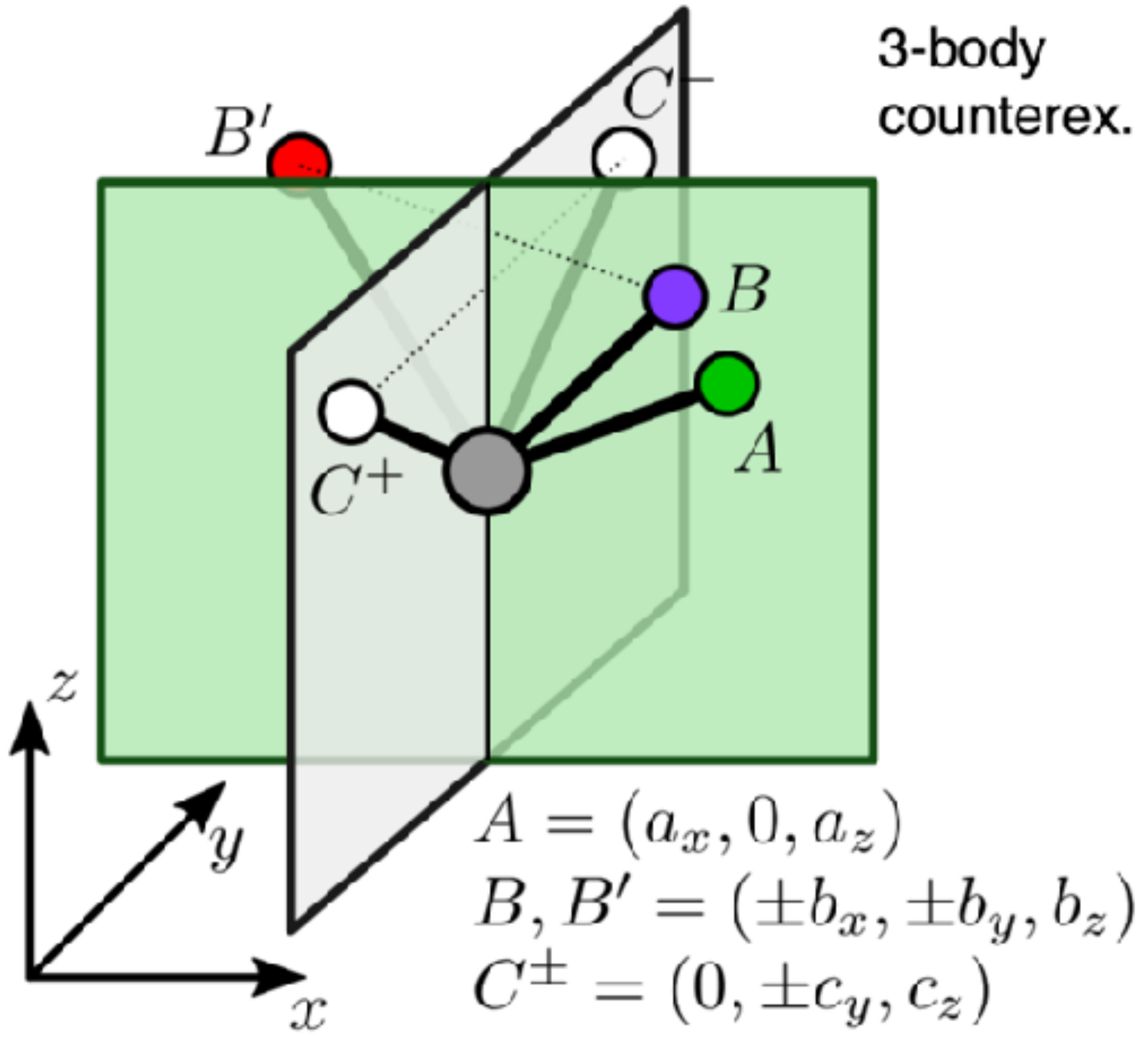
- Layers using order L tensors are **unable to identify the orientation** of structures with rotation symmetry $> L$-fold.
  - Why? **Spherical harmonics:** underlying orthonormal basis, rotationally symmetric.
- Issue is particularly prevalent for **E-GNN** and **GVP-GNN** (Tensor order 1).

Experiment 3: **Body order & neighbourhood fingerprints**

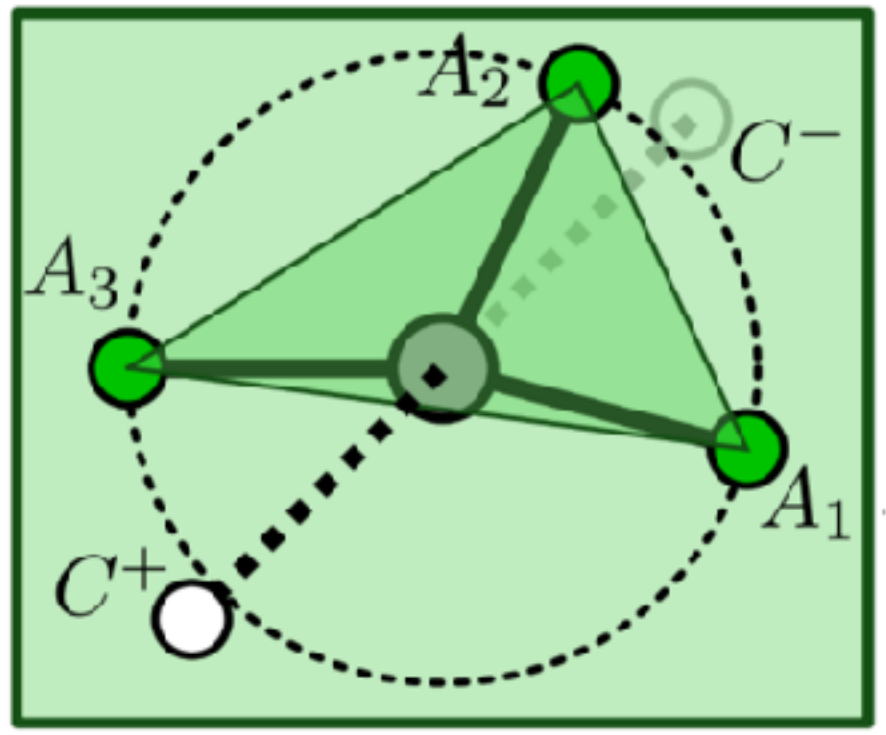
Counterexamples[1]: pairs of local neighbourhoods that **cannot be distinguished** when comparing their **set of k-body scalars.**



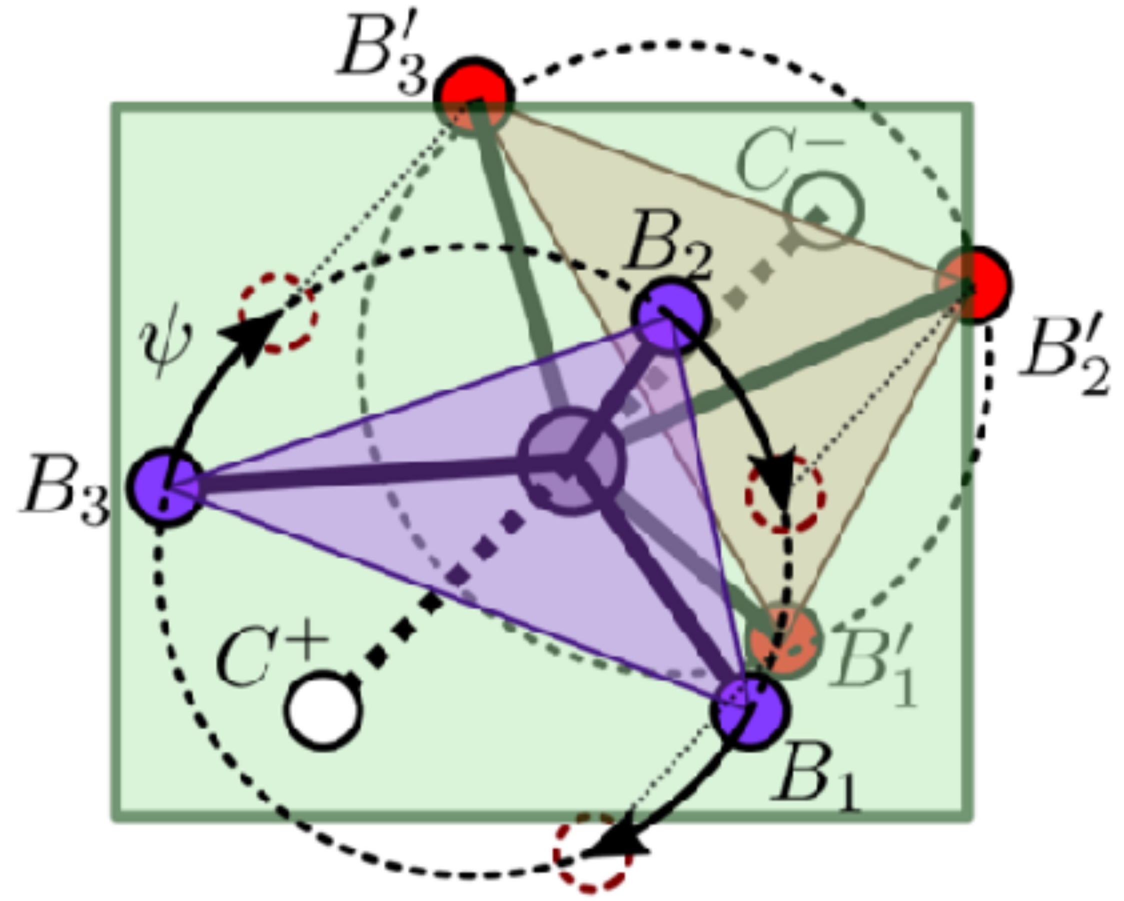
2-body
counterex.



3-body
counterex.



4-body **chiral**
counterex.



4-body **non-chiral**
counterex.

We train single layer geometric GNNs to distinguish the counterexamples.

[1] Pozdynakov et al., Phys. Rev. Letters, 2020. [2] Pozdynakov and Ceriotti, 2022.

Experiment 3: **Body order & neighbourhood fingerprints**

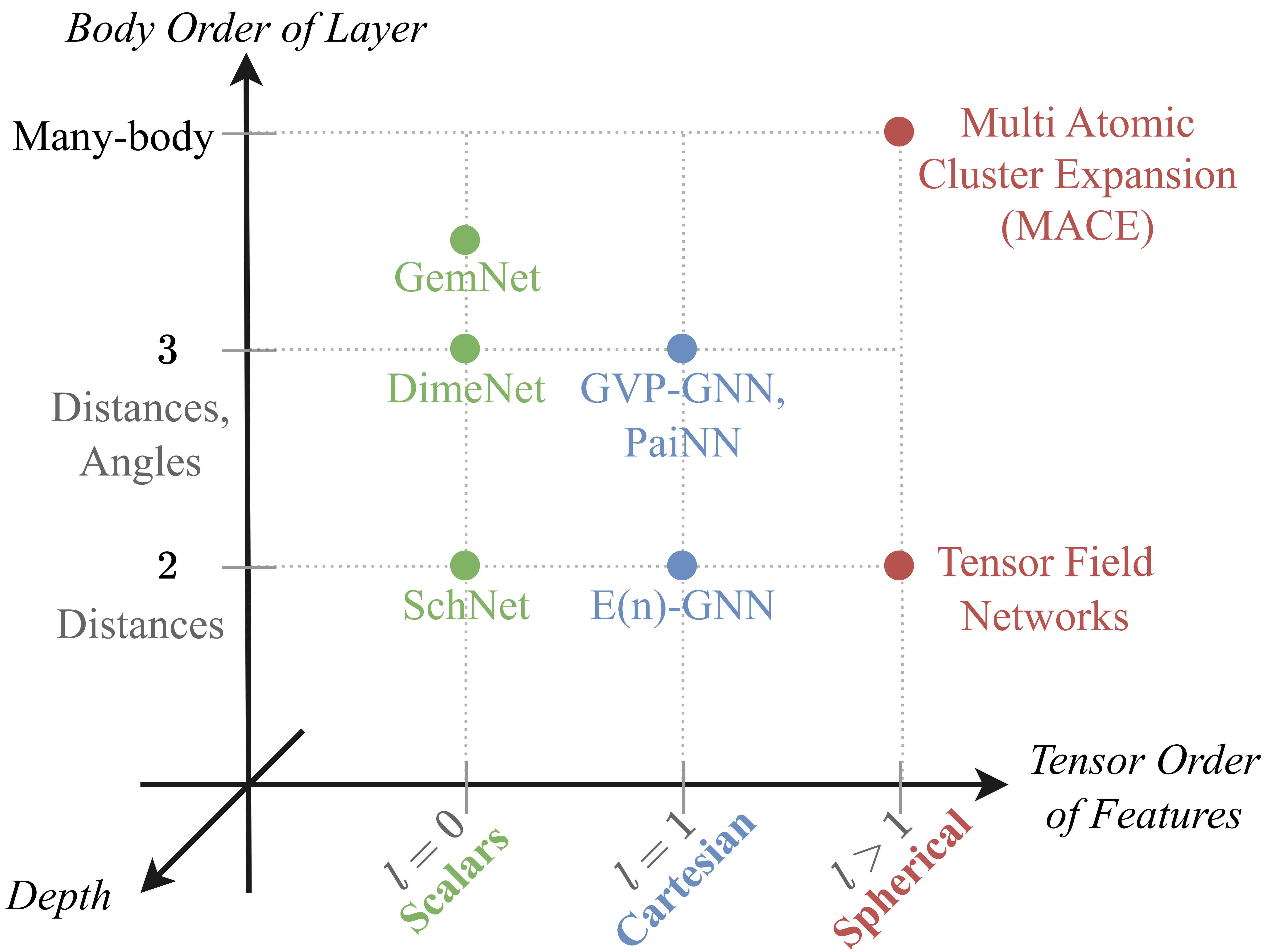
	GNN Layer	Counterexample from Pozdnyakov et al. [34] 2-body	3-body (Fig.1(b))	4-body (Fig.2(f))
Inv.	$\text{SchNet}_{\text{2-body}}$	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
	$\text{DimeNet}_{\text{3-body}}$	**100.0 ± 0.0**	50.0 ± 0.0	50.0 ± 0.0
$O(3)$-Equiv.	$\text{E-GNN}_{\text{2-body}}$	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
	$\text{GVP-GNN}_{\text{3-body}}$	**100.0 ± 0.0**	50.0 ± 0.0	50.0 ± 0.0
	$\text{TFN}_{\text{2-body}}$	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
	$\text{MACE}_{\text{3-body}}$	**100.0 ± 0.0**	50.0 ± 0.0	50.0 ± 0.0
	$\text{MACE}_{\text{4-body}}$	**100.0 ± 0.0**	**100.0 ± 0.0**	50.0 ± 0.0
	$\text{MACE}_{\text{5-body}}$	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**

Layers with body order k cannot distinguish the corresponding counterexample.

Conclusion & Key Takeaways

Axes of Geometric GNN expressivity

Key takeaway: deeper understanding of Geometric GNN design space



1. **Invariant layers:** limited expressivity, cannot distinguish one-hop identical geometric graphs.

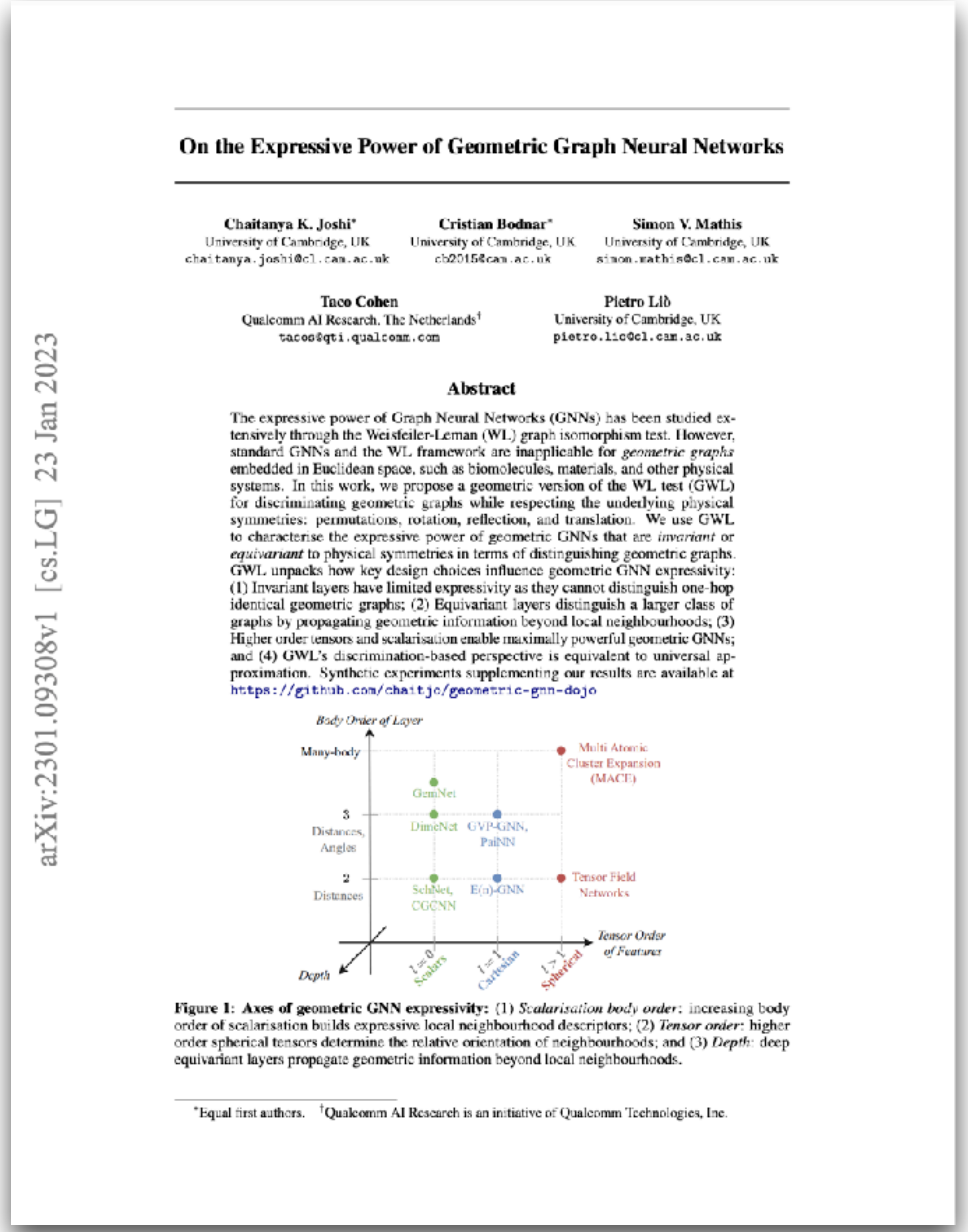
2. **Equivariant layers:** distinguish larger classes of graphs, propagate geometric information beyond local neighbourhoods.

3. Utility of **higher order tensors & scalarisation** for maximally powerful geometric GNNs.

What's in the full paper?

PDF: arxiv.org/abs/2301.09308

- **Geometric WL framework**: more + general results, details on scalarisation body order.
- **Connections with universality[1]:** equivalence between model's ability to discriminate geometric graphs and universal approximation.[2][3]
- **Future work:** towards maximally powerful geometric GNNs using insights from GWL & `geometric-gnn-dojo`.



arXiv:2301.09308v1 [cs.LG] 23 Jan 2023

On the Expressive Power of Geometric Graph Neural Networks

Chaitanya K. Joshi* — University of Cambridge, UK — chaitanya.joshi@cl.cam.ac.uk
Cristian Bodnar* — University of Cambridge, UK — cb2015@cam.ac.uk
Simon V. Mathis — University of Cambridge, UK — simon.mathis@cl.cam.ac.uk
Taco Cohen — Qualcomm AI Research, The Netherlands† — tacos@qti.qualcomm.com
Pietro Liò — University of Cambridge, UK — pietro.lio@cl.cam.ac.uk

Abstract

The expressive power of Graph Neural Networks (GNNs) has been studied extensively through the Weisfeiler-Leman (WL) graph isomorphism test. However, standard GNNs and the WL framework are inapplicable for *geometric graphs* embedded in Euclidean space, such as biomolecules, materials, and other physical systems. In this work, we propose a geometric version of the WL test (GWL) for discriminating geometric graphs while respecting the underlying physical symmetries: permutations, rotation, reflection, and translation. We use GWL to characterise the expressive power of geometric GNNs that are *invariant* or *equivariant* to physical symmetries in terms of distinguishing geometric graphs. GWL unpacks how key design choices influence geometric GNN expressivity: (1) Invariant layers have limited expressivity as they cannot distinguish one-hop identical geometric graphs; (2) Equivariant layers distinguish a larger class of graphs by propagating geometric information beyond local neighbourhoods; (3) Higher order tensors and scalarisation enable maximally powerful geometric GNNs; and (4) GWL's discrimination-based perspective is equivalent to universal approximation. Synthetic experiments supplementing our results are available at https://github.com/chaitjo/geometric-gnn-dojo

Figure 1: Axes of geometric GNN expressivity: (1) *Scalarisation body order*: increasing body order of scalarisation builds expressive local neighbourhood descriptors; (2) *Tensor order*: higher order spherical tensors determine the relative orientation of neighbourhoods; and (3) *Depth*: deep equivariant layers propagate geometric information beyond local neighbourhoods.

*Equal first authors. †Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

[1] Chen et al., NeurIPS, 2019. [2] Dym & Maron, Universality of TFN, ICLR, 2021. [3] Villar et al., Scalars are universal, NeurIPS, 2021.

Thank you for attending!

Please send us your questions, comments, and feedback!

Email: chaitanya.joshi@cl.cam.ac.uk, Twitter: @chaitjo

Chaitanya K. Joshi

Cristian Bodnar

Simon V. Mathis

Taco Cohen

Pietro Liò

On the Expressive Power of Geometric Graph Neural Networks

C. K. Joshi*, C. Bodnar*, S. V. Mathis, T. Cohen, P. Liò

📄 **PDF:** arxiv.org/abs/2301.09308

💻 **GitHub:** github.com/chaitjo/geometric-gnn-dojo