On the Expressive Power of Geometric Graph Neural Networks

Chaitanya K. Joshi [*1] **Cristian Bodnar** [*1] **Simon V. Mathis** [1] **Taco Cohen** [2] **Pietro Liò** [1]

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`

1. Introduction

Systems in biochemistry (Jamasb et al., 2022), material science (Chanussot et al., 2021), physical simulations (Sanchez-Gonzalez et al., 2020), and multiagent robotics (Li et al., 2020) contain both geometry and relational structure.

[*]Equal contribution [1]University of Cambridge, UK [2]Qualcomm AI Research, The Netherlands. Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc. Correspondence to: Chaitanya K. Joshi <chaitanya.joshi@cl.cam.ac.uk>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

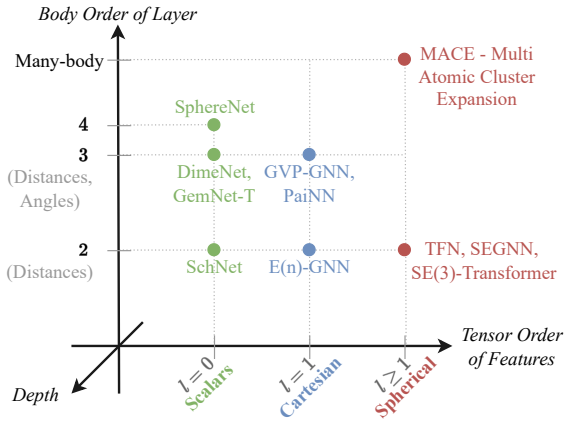


Figure 1: **Axes of geometric GNN expressivity:** (1) *Body order*: increasing scalarisation body order builds expressive local descriptors; (2) *Tensor order*: higher order tensors determine the relative orientation of neighbourhoods; and (3) *Depth*: deep equivariant layers propagate geometric information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

Such systems can be modelled via *geometric graphs* embedded in Euclidean space. For example, molecules are represented as a set of nodes which contain information about each atom and its 3D spatial coordinates as well as other geometric quantities such as velocity or acceleration. Notably, the geometric attributes transform along with Euclidean transformations of the system, *i.e.* they are equivariant to symmetry groups of rotations, reflections, and translation. Standard Graph Neural Networks (GNNs) which do not take spatial symmetries into account are ill-suited for geometric graphs, as the geometric attributes would no longer retain their physical meaning and transformation behaviour (Bogatskiy et al., 2022; Bronstein et al., 2021).

GNNs specialised for geometric graphs follow the message passing paradigm (Gilmer et al., 2017) where node features are updated in a permutation equivariant manner by aggregating features from local neighbourhoods. Crucially, in addition to permutations, the updated geometric features of the nodes retain the transformation behaviour of the initial attributes, *i.e.* they are also equivariant to the Lie group of rotations ($SO(d)$) or rotations and reflections ($O(d)$). We use $\mathfrak{G}$ as a generic symbol for these Lie groups. We consider two classes of geometric GNNs: (1) **$\mathfrak{G}$-equivariant models**,

where the intermediate features and propagated messages are equivariant geometric quantities such as vectors (Satorras et al., 2021) or tensors (Thomas et al., 2018); and (2) **$\mathfrak{G}$-invariant models**, which only propagate local invariant scalar features such as distances and angles (Schütt et al., 2018; Gasteiger et al., 2020).

Both classes of architectures have shown promising results for applications ranging from protein structure prediction (Baek et al., 2021) and design (Dauparas et al., 2022) to molecular dynamics (Batzner et al., 2022) and catalysis (Gasteiger et al., 2021). At the same time, key theoretical questions remain unanswered: (1) How to characterise the *expressive power* of geometric GNNs? and (2) What is the tradeoff between $\mathfrak{G}$-equivariant and $\mathfrak{G}$-invariant GNNs?

The graph isomorphism problem (Read & Corneil, 1977) and the Weisfeiler-Leman (WL) (Weisfeiler & Leman, 1968) test for distinguishing non-isomorphic graphs have become a powerful tool for analysing the expressive power of non-geometric GNNs (Xu et al., 2019; Morris et al., 2019). The WL framework has been a major driver of progress in graph representation learning (Chen et al., 2019; Maron et al., 2019; Dwivedi et al., 2023; Bodnar et al., 2021b;a). However, WL does not directly apply to geometric graphs as they exhibit a stronger notion of geometric isomorphism that accounts for spatial symmetries.

Contributions. In this work, we study the expressive power of geometric GNNs from the perspective of discriminating non-isomorphic geometric graphs.

- In Section 3, we propose a geometric version of the Weisfeiler-Leman graph isomorphism test, termed GWL, which is a theoretical upper bound on the expressive power of geometric GNNs.
- In Section 4, we use the GWL framework to formalise how key design choices influence geometric GNN expressivity, summarised in Figure 1. Notably, invariant GNNs cannot distinguish graphs where one-hop neighbourhoods are the same and fail to compute non-local geometric properties such as volume, centroid, etc. Equivariant GNNs distinguish a larger class of graphs as stacking equivariant layers propagates geometric information beyond local neighbourhoods.
- Synthetic experiments in Section 5 demonstrate practical implications for building maximally powerful geometric GNNs, *s.a.* geometric oversquashing with increased depth, and counterexamples that highlight the utility of higher order order spherical tensors.
- Finally, in Section 6, we prove an equivalence between a model's ability to discriminate geometric graphs and its universal approximation capabilities. While universality is binary, GWL's discrimination-based perspective provides a more granular and practically insightful lens into geometric GNN expressivity.

2. Background

Graph Isomorphism. An attributed graph $\mathcal{G} = (\boldsymbol{A}, \boldsymbol{S})$ is a set $\mathcal{V}$ of n nodes connected by edges. $\boldsymbol{A}$ denotes an $n \times n$ adjacency matrix where each entry $a_{ij} \in \{0, 1\}$ indicates the presence or absence of an edge connecting nodes i and j. The matrix of *scalar* features $\boldsymbol{S} \in \mathbb{R}^{n \times f}$ stores attributes $\boldsymbol{s}_i \in \mathbb{R}^f$ associated with each node i, where the subscripts index rows and columns in the corresponding matrices.

The graph isomorphism problem asks whether two graphs are the same, but drawn differently. Two attributed graphs $\mathcal{G}, \mathcal{H}$ are *isomorphic* (denoted $\mathcal{G} \simeq \mathcal{H}$) if there exists an edge-preserving bijection $b : \mathcal{V}(\mathcal{G}) \to \mathcal{V}(\mathcal{H})$ such that $\boldsymbol{s}_i^{(\mathcal{G})} = \boldsymbol{s}_{b(i)}^{(\mathcal{H})}$.

Weisfeiler-Leman Test (WL). WL tests whether two (attributed) graphs are isomorphic. At iteration zero, WL assigns a *colour* $c_i^{(0)} \in C$ from a countable space of colours C to each node i. Nodes are coloured the same if their features are the same, otherwise, they are coloured differently. In subsequent iterations t, WL iteratively updates the colouring $\boldsymbol{c}_i^{(t)} \in C$:

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

where HASH is an injective map (*i.e.* a perfect hash map) that assigns a unique colour to each input. The test terminates when the partition of the nodes induced by the colours becomes stable. Given two graphs $\mathcal{G}$ and $\mathcal{H}$, if there exists some iteration t for which $\{\!\{c_i^{(t)} \mid i \in \mathcal{V}(\mathcal{G})\}\!\} \neq \{\!\{c_i^{(t)} \mid i \in \mathcal{V}(\mathcal{H})\}\!\}$, then the graphs are not isomorphic. Otherwise, WL is inconclusive and we cannot distinguish the two graphs.

Group Theory. We assume basic familiarity with group theory, see Zee (2016) for an overview. We denote the action of a group $\mathfrak{G}$ on a space X by $\mathfrak{g} \cdot x$. If $\mathfrak{G}$ acts on spaces X and Y, we say: (1) A function $f : X \to Y$ is $\mathfrak{G}$-*invariant* if $f(\mathfrak{g} \cdot x) = f(x)$, *i.e.* the output remains unchanged under transformations of the input; and (2) A function $f : X \to Y$ is $\mathfrak{G}$-*equivariant* if $f(\mathfrak{g} \cdot x) = \mathfrak{g} \cdot f(x)$, *i.e.* a transformation of the input must result in the output transforming equivalently.

We also consider $\mathfrak{G}$-*orbit injective* functions: The $\mathfrak{G}$-*orbit* of $x \in X$ is $\mathcal{O}_{\mathfrak{G}}(x) = \{\mathfrak{g} \cdot x \mid \mathfrak{g} \in \mathfrak{G}\} \subseteq X$. When x and x' are part of the same orbit, we write $x \simeq x'$. We say a function $f : X \to Y$ is $\mathfrak{G}$-*orbit injective* if we have $f(x_1) = f(x_2)$ if and only if $x_1 \simeq x_2$ for any $x_1, x_2 \in X$. Such a function is $\mathfrak{G}$-invariant, since $f(\mathfrak{g} \cdot x) = f(x)$.

We work with the permutation group over n elements S_n and the Lie groups $\mathfrak{G} = SO(d)$ or $\mathfrak{G} = O(d)$. Invariance to translations $T(d)$ is conventionally handled using relative positions. Given one of the groups above, for an element $\mathfrak{g}$ we denote by $\boldsymbol{M}_{\mathfrak{g}}$ (or another capital letter) its standard matrix representation.

Geometric Graphs. A geometric graph $\mathcal{G} = (\boldsymbol{A}, \boldsymbol{S}, \vec{\boldsymbol{V}}, \vec{\boldsymbol{X}})$ is an attributed graph that is also decorated with geometric attributes: node coordinates $\vec{\boldsymbol{X}} \in \mathbb{R}^{n \times d}$ and (optionally) vector features $\vec{\boldsymbol{V}} \in \mathbb{R}^{n \times d}$ (*e.g.* velocity, acceleration). Without loss of generality, we work with a single vector feature per node. Our setup generalises to multiple vector features or higher-order tensors. The geometric attributes transform as follows under the action of the relevant groups: (1) S_n acts on the graph via $\boldsymbol{P}_\sigma \mathcal{G} := (\boldsymbol{P}_\sigma \boldsymbol{A} \boldsymbol{P}_\sigma^\top, \boldsymbol{P}_\sigma \boldsymbol{S}, \boldsymbol{P}_\sigma \vec{\boldsymbol{V}}, \boldsymbol{P}_\sigma \vec{\boldsymbol{X}})$; (2) Orthogonal transformations $\boldsymbol{Q}_\mathfrak{g} \in \mathfrak{G}$ act on $\vec{\boldsymbol{V}}, \vec{\boldsymbol{X}}$ via $\vec{\boldsymbol{V}} \boldsymbol{Q}_\mathfrak{g}, \vec{\boldsymbol{X}} \boldsymbol{Q}_\mathfrak{g}$; and (3) Translations $\vec{\boldsymbol{t}} \in T(d)$ act on the coordinates $\vec{\boldsymbol{X}}$ via $\vec{\boldsymbol{x}}_i + \vec{\boldsymbol{t}}$ for all nodes i.

Two geometric graphs $\mathcal{G}$ and $\mathcal{H}$ are *geometrically isomorphic* if there exists an attributed graph isomorphism b such that the geometric attributes are equivalent, up to global group actions $\boldsymbol{Q}_\mathfrak{g} \in \mathfrak{G}$ and $\vec{\boldsymbol{t}} \in T(d)$:

$$\left(\boldsymbol{s}_{b(i)}^{(\mathcal{H})},\ \boldsymbol{Q}_\mathfrak{g} \vec{\boldsymbol{v}}_{b(i)}^{(\mathcal{H})},\ \boldsymbol{Q}_\mathfrak{g}(\vec{\boldsymbol{x}}_{b(i)}^{(\mathcal{H})} + \vec{\boldsymbol{t}})\right) = \left(\boldsymbol{s}_i^{(\mathcal{G})}, \vec{\boldsymbol{v}}_i^{(\mathcal{G})}, \vec{\boldsymbol{x}}_i^{(\mathcal{G})}\right) \quad \text{for all } i \in \mathcal{V}(\mathcal{G}). \quad (2)$$

Geometric graph isomorphism and distinguishing (sub-) graph geometries has important practical implications for representation learning. For *e.g.*, for molecular systems, an ideal architecture should map distinct local structural environments around atoms to distinct representations (Bartók et al., 2013; Pozdnyakov et al., 2020), which can then be used for downstream predictive task.

Geometric Graph Neural Networks. We consider two broad classes of geometric GNN architectures. (1) $\mathfrak{G}$-*equivariant GNN layers* (Thomas et al., 2018; Satorras et al., 2021) propagate scalar as well as geometric vector features from iteration t to $t+1$ via learnable aggregate and update functions, AGG and UPD, respectively:

$$\boldsymbol{s}_i^{(t+1)}, \vec{\boldsymbol{v}}_i^{(t+1)} := \text{UPD}\Big((\boldsymbol{s}_i^{(t)}, \vec{\boldsymbol{v}}_i^{(t)}) \,, \text{AGG}\left(\{\!\{ (\boldsymbol{s}_i^{(t)}, \vec{\boldsymbol{v}}_i^{(t)}), (\boldsymbol{s}_j^{(t)}, \vec{\boldsymbol{v}}_j^{(t)}), \vec{\boldsymbol{x}}_{ij} \mid j \in \mathcal{N}_i \}\!\} \right) \Big), \quad (3)$$

where $\vec{\boldsymbol{x}}_{ij} = \vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_j$ denote relative position vectors. Alternatively, (2) $\mathfrak{G}$-*invariant GNN layers* (Schütt et al., 2018; Gasteiger et al., 2020) do not update vector features and only propagate invariant scalar quantities computed using geometric information within local neighbourhoods:

$$\boldsymbol{s}_i^{(t+1)} := \text{UPD}\Big(\boldsymbol{s}_i^{(t)} \,, \text{AGG}\left(\{\!\{ (\boldsymbol{s}_i^{(t)}, \vec{\boldsymbol{v}}_i^{(t)}), (\boldsymbol{s}_j^{(t)}, \vec{\boldsymbol{v}}_j^{(t)}), \vec{\boldsymbol{x}}_{ij} \mid j \in \mathcal{N}_i \}\!\} \right) \Big). \quad (4)$$

For both models, the scalar features at the final iteration are mapped to graph-level predictions via a permutation-invariant readout $f : \mathbb{R}^{n \times f} \to \mathbb{R}^{f'}$. **In Appendix B, we provide unified equations and examples of geometric GNNs covered by our framework.**

3. The Geometric Weisfeiler-Leman Test

Assumptions. Analogous to the WL test, we assume for the rest of this section that all geometric graphs we want to distinguish are finite in size and come from a countable dataset. In other words, the geometric and scalar features the nodes are equipped with come from countable subsets $C \subset \mathbb{R}^d$ and $C' \subset \mathbb{R}$, respectively. As a result, when we require functions to be injective, we require them to be injective over the countable set of $\mathfrak{G}$-orbits that are obtained by acting with $\mathfrak{G}$ on the dataset.

Intuition. For an intuition of how to generalise WL to geometric graphs, we note that WL uses a local, node-centric, procedure to update the colour of each node i using the colours of its the 1-hop *neighbourhood* $\mathcal{N}_i$. In the geometric setting, $\mathcal{N}_i$ is an attributed point cloud around the central node i. As a result, each neighbourhood carries two types of information: (1) neighbourhood type (invariant to $\mathfrak{G}$) and (2) neighbourhood geometric orientation (equivariant to $\mathfrak{G}$). This leads to constraints on the generalisation of the neighbourhood aggregation procedure of Equation 1. From an axiomatic point of view, our generalisation of the WL aggregation procedure must meet two properties:

Property 1: Orbit injectivity of colours. If two neighbourhoods are the same up to an action of $\mathfrak{G}$ (*e.g.* rotation), then the colours of the corresponding central nodes should be the same. Thus, the colouring must be $\mathfrak{G}$-orbit injective – which also makes it $\mathfrak{G}$-invariant – over the countable set of all orbits of neighbourhoods in our dataset.

Property 2: Preservation of local geometry. A key property of WL is that the aggregation is injective. A $\mathfrak{G}$-invariant colouring procedure that purely satisfies Property 1 is not sufficient because, by definition, it loses spatial properties of each neighbourhood such as the relative pose or orientation (Hinton et al., 2011). Thus, we must additionally update auxiliary *geometric information* variables in a way that is $\mathfrak{G}$-equivariant and injective.

Geometric Weisfeiler-Leman (GWL). These intuitions motivate the following definition of the GWL test. At initialisation, we assign to each node $i \in \mathcal{V}$ a scalar node colour $c_i \in C'$ and an auxiliary object $\boldsymbol{g}_i$ containing the geometric information associated to it:

$$c_i^{(0)} := \text{HASH}(\boldsymbol{s}_i), \quad \boldsymbol{g}_i^{(0)} := \left(c_i^{(0)}, \vec{\boldsymbol{v}}_i\right), \quad (5)$$

where HASH denotes an injective map over the scalar attributes $\boldsymbol{s}_i$ of node i, *e.g.* the HASH of standard WL. To define the inductive step, assume we have the colours of the nodes and the associated geometric objects at iteration $t-1$. Then, we can aggregate the geometric information around node i into a new object as follows:

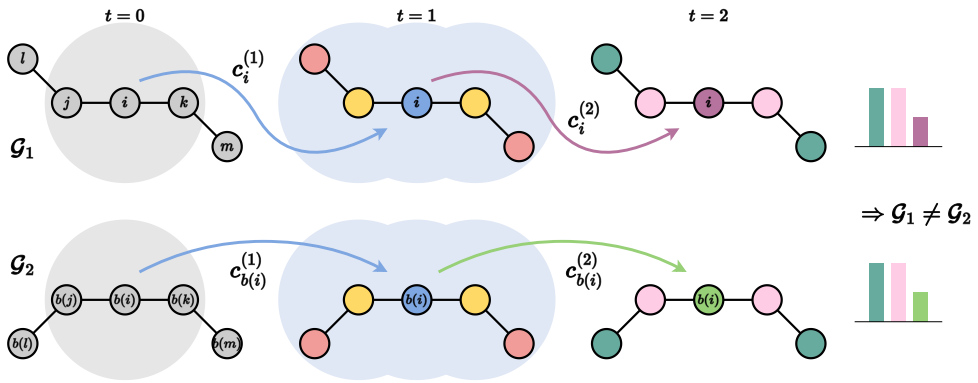


Figure 2: **Geometric Weisfeiler-Leman Test.** GWL distinguishes non-isomorphic geometric graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ by injectively assigning colours to distinct neighbourhood patterns, up to global symmetries. Each iteration expands the neighbourhood from which geometric information can be gathered (shaded for node i). Example inspired by Schütt et al. (2021), $\mathfrak{G} = O(d)$.

$$\boldsymbol{g}_i^{(t)} := \big((c_i^{(t-1)}, \boldsymbol{g}_i^{(t-1)}),\ \{\!\{(c_j^{(t-1)}, \boldsymbol{g}_j^{(t-1)}, \vec{\boldsymbol{x}}_{ij}) \mid j \in \mathcal{N}_i\}\!\}\big), \quad (6)$$

Here $\{\!\{\cdot\}\!\}$ denotes a multiset – a set in which elements may occur more than once. Importantly, the group $\mathfrak{G}$ can act on the geometric objects above inductively by acting on the geometric information inside it. This amounts to rotating (or reflecting) the entire t-hop neighbourhood contained inside:

$$\mathfrak{g} \cdot \boldsymbol{g}_i^{(0)} := \left(c_i^{(0)},\ \boldsymbol{Q}_{\mathfrak{g}} \vec{\boldsymbol{v}}_i\right), \quad (7)$$

$$\mathfrak{g} \cdot \boldsymbol{g}_i^{(t)} := \Big((c_i^{(t-1)}, \mathfrak{g} \cdot \boldsymbol{g}_i^{(t-1)}),\ \{\!\{(c_j^{(t-1)}, \mathfrak{g} \cdot \boldsymbol{g}_j^{(t-1)}, \boldsymbol{Q}_{\mathfrak{g}} \vec{\boldsymbol{x}}_{ij}) \mid j \in \mathcal{N}_i\}\!\}\Big) \quad (8)$$

Clearly, the aggregation building $\boldsymbol{g}_i$ for any time-step t is injective and $\mathfrak{G}$-equivariant. Finally, we can compute the node colours at iteration t for all $i \in \mathcal{V}$ by aggregating the geometric information in the neighbourhood around node i:

$$c_i^{(t)} := \text{I-HASH}^{(t)}\left(\boldsymbol{g}_i^{(t)}\right), \quad (9)$$

by using a $\mathfrak{G}$-orbit injective and $\mathfrak{G}$-invariant function that we denote by I-HASH. That is for any geometric objects $\boldsymbol{g}, \boldsymbol{g}'$, $\text{I-HASH}(\boldsymbol{g}) = \text{I-HASH}(\boldsymbol{g}')$ if and only if there exists $\mathfrak{g} \in \mathfrak{G}$ such that $\boldsymbol{g} = \mathfrak{g} \cdot \boldsymbol{g}'$. Note that I-HASH is an idealised $\mathfrak{G}$-orbit injective function, similar to the HASH function used in WL, which is not necessarily continuous.

Overview. With each iteration, $\boldsymbol{g}_i^{(t)}$ aggregates geometric information in progressively larger t-hop subgraph neighbourhoods $\mathcal{N}_i^{(t)}$ around the node i. The node colours summarise the structure of these t-hops via the $\mathfrak{G}$-invariant aggregation performed by I-HASH. The procedure terminates when the partitions of the nodes induced by the colours do not change from the previous iteration. Finally, given two geometric graphs $\mathcal{G}$ and $\mathcal{H}$, if there exists some iteration t for which $\{\!\{c_i^{(t)} \mid i \in \mathcal{V}(\mathcal{G})\}\!\} \neq \{\!\{c_i^{(t)} \mid i \in \mathcal{V}(\mathcal{H})\}\!\}$, then GWL deems the two graphs as geometrically non-isomorphic. Otherwise, GWL cannot distinguish them.

Invariant GWL. Since we are interested in understanding the role of $\mathfrak{G}$-equivariance, we also consider a more restrictive Invariant GWL (IGWL) that only updates node colours using the $\mathfrak{G}$-orbit injective I-HASH function and does not propagate geometric information:

$$c_i^{(t)} := \text{I-HASH}\Big((c_i^{(t-1)}, \vec{\boldsymbol{v}}_i),\ \{\!\{(c_j^{(t-1)}, \vec{\boldsymbol{v}}_j,\ \vec{\boldsymbol{x}}_{ij}) \mid j \in \mathcal{N}_i\}\!\}\Big). \quad (10)$$

IGWL with k-body scalars. In order to further analyse the construction of the node colouring function I-HASH, we consider $\text{IGWL}_{(k)}$ based on the maximum number of nodes involved in the computation of $\mathfrak{G}$-invariant scalars (also known as the 'body order' (Batatia et al., 2022a)):

$$c_i^{(t)} := \text{I-HASH}_{(k)}\Big((c_i^{(t-1)}, \vec{\boldsymbol{v}}_i),\ \{\!\{(c_j^{(t-1)}, \vec{\boldsymbol{v}}_j,\ \vec{\boldsymbol{x}}_{ij}) \mid j \in \mathcal{N}_i\}\!\}\Big), \quad (11)$$

and $\text{I-HASH}_{(k+1)}$ is defined as:

$$\text{HASH}\Big(\{\!\{\text{I-HASH}\Big((c_i^{(t-1)}, \vec{\boldsymbol{v}}_i), \{\!\{(c_{j_1}^{(t-1)}, \vec{\boldsymbol{v}}_{j_1},\ \vec{\boldsymbol{x}}_{ij_1}), \ldots, (c_{j_k}^{(t-1)}, \vec{\boldsymbol{v}}_{j_k},\ \vec{\boldsymbol{x}}_{ij_k})\}\!\}\Big) \mid \boldsymbol{j} \in (\mathcal{N}_i)^k\}\!\}\Big), \quad (12)$$

where $\boldsymbol{j} = [j_1, \ldots, j_k]$ are all possible k-tuples formed of elements of $\mathcal{N}_i$. Therefore, $\text{IGWL}_{(k)}$ is now constrained to extract information only from all the possible k-sized tuples of nodes (including the central node) in a neighbourhood. For instance, $\text{I-HASH}_{(2)}$ can identify neighbourhoods only up to pairwise distances among the central node and any of its neighbours (*i.e.* a 2-body scalar), while $\text{I-HASH}_{(3)}$ up to

distances and angles formed by any two edges (*i.e.* a 3-body scalar). Notably, distances and angles alone are incomplete descriptors of local geometry as there exist several counterexamples that require higher body-order invariants to be distinguished (Bartók et al., 2013; Pozdnyakov et al., 2020). Therefore, I-HASH$_{(k)}$ with lower scalarisation body order k makes the colouring weaker.

3.1. Characterising the expressivity of geometric GNNs

Following Xu et al. (2019); Morris et al. (2019), we upper bound the expressive power of geometric GNNs by the GWL test. We show that any $\mathfrak{G}$-equivariant GNN can be at most as powerful as GWL in distinguishing non-isomorphic geometric graphs. Proofs are available in Appendix E.

Theorem 1. *Any pair of geometric graphs distinguishable by a $\mathfrak{G}$-equivariant GNN is also distinguishable by GWL.*

With sufficient iterations, the output of $\mathfrak{G}$-equivariant GNNs can be equivalent to GWL if certain conditions are met regarding the aggregate, update and readout functions.

Proposition 2. *$\mathfrak{G}$-equivariant GNNs have the same expressive power as GWL if the following conditions hold: (1) The aggregation* AGG *is an injective, $\mathfrak{G}$-equivariant multiset function. (2) The scalar part of the update* UPD$_s$ *is a $\mathfrak{G}$-orbit injective, $\mathfrak{G}$-invariant multiset function. (3) The vector part of the update* UPD$_v$ *is an injective, $\mathfrak{G}$-equivariant multiset function. (4) The graph-level readout f is an injective multiset function.*

Similar statements can be made for $\mathfrak{G}$-invariant GNNs and IGWL. Thus, we can directly transfer theoretical results between GWL/IGWL, which are abstract mathematical tools, and the class of geometric GNNs upper bounded by the respective tests. This connection has several interesting practical implications, discussed subsequently.

4. Understanding the Design Space of Geometric GNNs via GWL

Overview. We demonstrate the utility of the GWL framework for understanding how geometric GNN design choices (Batatia et al., 2022a) influence expressivity: (1) Depth or number of layers; (2) Invariant vs. equivariant message passing; and (3) Body order of scalarisation. In doing so, we formalise theoretical limitations of current architectures and provide practical implications for designing maximally powerful models. Proofs are available in Appendix D.

4.1. Role of depth: propagating geometric information

Let us consider the simplified setting of two geometric graphs $\mathcal{G}_1 = (\boldsymbol{A}_1, \boldsymbol{S}_1, \vec{\boldsymbol{V}}_1, \vec{\boldsymbol{X}}_1)$ and $\mathcal{G}_2 = (\boldsymbol{A}_2, \boldsymbol{S}_2, \vec{\boldsymbol{V}}_2, \vec{\boldsymbol{X}}_2)$ such that the underlying attributed graphs $(\boldsymbol{A}_1, \boldsymbol{S}_1)$ and $(\boldsymbol{A}_2, \boldsymbol{S}_2)$ are isomorphic. This case frequently occurs in chemistry, where molecules occur in different conformations, but with the same graph topology given by the covalent bonding structure.

Each iteration of GWL aggregates geometric information $\boldsymbol{g}_i^{(k)}$ from progressively larger neighbourhoods $\mathcal{N}_i^{(k)}$ around the node i, and distinguishes (sub-)graphs via comparing $\mathfrak{G}$-orbit injective colouring of $\boldsymbol{g}_i^{(k)}$. We say $\mathcal{G}_1$ and $\mathcal{G}_2$ are *k-hop distinct* if for all graph isomorphisms b, there is some node $i \in \mathcal{V}_1, b(i) \in \mathcal{V}_2$ such that the corresponding k-hop subgraphs $\mathcal{N}_i^{(k)}$ and $\mathcal{N}_{b(i)}^{(k)}$ are distinct. Otherwise, we say $\mathcal{G}_1$ and $\mathcal{G}_2$ are *k-hop identical* if all corresponding k-hop subgraphs $\mathcal{N}_i^{(k)}$ and $\mathcal{N}_{b(i)}^{(k)}$ are the same up to group actions. We can now formalise what geometric graphs can and cannot be distinguished by GWL and maximally powerful geometric GNNs, in terms of the number of iterations.

Proposition 3. *GWL can distinguish any k-hop distinct geometric graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ where the underlying attributed graphs are isomorphic, and k iterations are sufficient.*

Proposition 4. *Up to k iterations, GWL cannot distinguish any k-hop identical geometric graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ where the underlying attributed graphs are isomorphic.*

Additionally, we can state the following results about the more constrained IGWL.

Proposition 5. *IGWL can distinguish any 1-hop distinct geometric graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ where the underlying attributed graphs are isomorphic, and 1 iteration is sufficient.*

Proposition 6. *Any number of iterations of IGWL cannot distinguish any 1-hop identical geometric graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ where the underlying attributed graphs are isomorphic.*

An example illustrating Propositions 3 and 6 is shown in Figures 2 and 3, respectively.

We can now consider the more general case where the underlying attributed graphs for $\mathcal{G}_1 = (\boldsymbol{A}_1, \boldsymbol{S}_1, \vec{\boldsymbol{V}}_1, \vec{\boldsymbol{X}}_1)$ and $\mathcal{G}_2 = (\boldsymbol{A}_2, \boldsymbol{S}_2, \vec{\boldsymbol{V}}_2, \vec{\boldsymbol{X}}_2)$ are non-isomorphic and constructed from point clouds using radial cutoffs, as conventional for biochemistry and material science applications.

Proposition 7. *Assuming geometric graphs are constructed from point clouds using radial cutoffs, GWL can distinguish any geometric graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ where the underlying attributed graphs are non-isomorphic. At most k_{Max} iterations are sufficient, where k_{Max} is the maximum graph diameter among $\mathcal{G}_1$ and $\mathcal{G}_2$.*

4.2. Limitations of invariant message passing: failure to capture global geometry

Propositions 3 and 6 enable us to compare the expressive powers of GWL and IGWL.

Theorem 8. *GWL is strictly more powerful than IGWL.*

This statement formalises the advantage of $\mathfrak{G}$-equivariant intermediate layers for geometric graphs, as prescribed in the Geometric Deep Learning blueprint (Bronstein et al., 2021), in addition to echoing similar intuitions in the computer vision community. As remarked by Hinton et al. (2011), translation invariant models do not understand the relationship between the various parts of an image (termed the "Picasso problem"). Similarly, our results point to IGWL and $\mathfrak{G}$-invariant GNNs failing to understand how the 1-hop neighbourhoods in a graph are oriented w.r.t. each other.

As a result, even the most powerful $\mathfrak{G}$-invariant GNNs are restricted in their ability to compute global and non-local geometric properties.

Proposition 9. *IGWL and $\mathfrak{G}$-invariant GNNs cannot decide several geometric graph properties: (1) perimeter, surface area, and volume of the bounding box/sphere enclosing the geometric graph; (2) distance from the centroid or centre of mass; and (3) dihedral angles.*

Finally, we identify a setting where this distinction between the two approaches disappears.

Proposition 10. *IGWL has the same expressive power as GWL for fully connected geometric graphs.*

Practical Implications. Proposition 9 highlights critical theoretical limitations of $\mathfrak{G}$-invariant GNNs. This suggests that $\mathfrak{G}$-equivariant GNNs should be preferred when working with large geometric graphs such as macromolecules with thousands of nodes, where message passing is restricted to local radial neighbourhoods around each node. Stacking multiple $\mathfrak{G}$-equivariant layers enables the computation of compositional geometric features.

Two straightforward approaches to overcoming the limitations of $\mathfrak{G}$-invariant GNNs may be: (1) pre-computing non-local geometric properties as input features, *e.g.* models such as GemNet (Gasteiger et al., 2021) and ComENet (Wang et al., 2022) use two-hop dihedral angles. And (2) working with fully connected geometric graphs, as Proposition 10 suggests that $\mathfrak{G}$-equivariant and $\mathfrak{G}$-invariant GNNs can be made equally powerful when performing all-to-all message passing. This is supported by the empirical success of recent $\mathfrak{G}$-invariant *Graph Transformers* (Joshi, 2020; Shi et al., 2022) for small molecules with tens of nodes, where working with full graphs is tractable.

4.3. Role of scalarisation body order: identifying neighbourhood $\mathfrak{G}$-orbits

At each iteration of GWL and IGWL, the I-HASH function assigns a $\mathfrak{G}$-invariant colouring to distinct geometric neighbourhood patterns. I-HASH is an idealised $\mathfrak{G}$-orbit injective function which is not necessarily continuous. In geometric GNNs, this corresponds to scalarising local geometric information when updating the scalar features; examples are shown in equation 14 and equation 15. We can analyse the construction of the I-HASH function and the scalarisation step in geometric GNNs via the k-body variations $\text{IGWL}_{(k)}$, described in Section 3. In doing so, we will make connections between IGWL and WL for non-geometric graphs.

Firstly, we formalise the relationship between the injectivity of $\text{I-HASH}_{(k)}$ and the maximum cardinality of local neighbourhoods in a given dataset.

Proposition 11. $\text{I-HASH}_{(m)}$ *is $\mathfrak{G}$-orbit injective for $m = max(\{|\mathcal{N}_i| \mid i \in \mathcal{V}\})$, the maximum cardinality of all local neighbourhoods $\mathcal{N}_i$ in a given dataset.*

Practical Implications. While building provably injective $\text{I-HASH}_{(k)}$ functions may require intractably high k, the hierarchy of $\text{IGWL}_{(k)}$ tests enable us to study the expressive power of practical $\mathfrak{G}$-invariant aggregators used in current geometric GNN layers, *e.g.* SchNet (Schütt et al., 2018), E-GNN (Satorras et al., 2021), and TFN (Thomas et al., 2018) use distances, while DimeNet (Gasteiger et al., 2020) uses distances and angles. Notably, MACE (Batatia et al., 2022b) constructs a *complete* basis of scalars up to arbitrary body order k via Atomic Cluster Expansion (Dusson et al., 2019), which can be $\mathfrak{G}$-orbit injective if the conditions in Proposition 11 are met. We can state the following about the $\text{IGWL}_{(k)}$ hierarchy and the corresponding GNNs.

Proposition 12. *$\text{IGWL}_{(k)}$ is at least as powerful as $\text{IGWL}_{(k-1)}$. For $k \leq 5$, $\text{IGWL}_{(k)}$ is strictly more powerful than $\text{IGWL}_{(k-1)}$.*

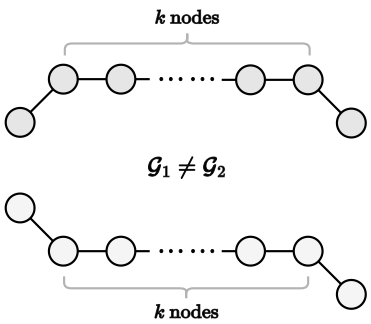
Finally, we show that $\text{IGWL}_{(2)}$ is equivalent to WL when all the pairwise distances are the same. A similar observation was recently made by Pozdnyakov & Ceriotti (2022).

Proposition 13. *Let $\mathcal{G}_1 = (\boldsymbol{A}_1, \boldsymbol{S}_1, \vec{\boldsymbol{X}}_1)$ and $\mathcal{G}_2 = (\boldsymbol{A}_2, \boldsymbol{S}_2, \vec{\boldsymbol{X}}_2)$ be two geometric graphs with the property that all edges have equal length. Then, $\text{IGWL}_{(2)}$ distinguishes the two graphs if and only if WL can distinguish the attributed graphs $(\boldsymbol{A}_1, \boldsymbol{S}_1)$ and $(\boldsymbol{A}_1, \boldsymbol{S}_1)$.*

This equivalence points to limitations of distance-based $\mathfrak{G}$-invariant models like SchNet (Schütt et al., 2018). These models suffer from all well-known failure cases of WL, *e.g.* they cannot distinguish two equilateral triangles from the regular hexagon (Gasteiger et al., 2020).

5. Synthetic Experiments on Expressivity

Overview. We provide synthetic experiments to supplement our theoretical results and highlight practical challenges in building maximally powerful geometric GNNs, *s.a.* oversquashing of geometric information with increased depth (Table 1) and the need for higher order tensors in $\mathfrak{G}$-equivariant GNNs (Table 2). We hope that the accom-



	($k = 4$-chains)	Number of layers				
	GNN Layer	$\lfloor \frac{k}{2} \rfloor$	$\lfloor \frac{k}{2} \rfloor + 1 = \mathbf{3}$	$\lfloor \frac{k}{2} \rfloor + 2$	$\lfloor \frac{k}{2} \rfloor + 3$	$\lfloor \frac{k}{2} \rfloor + 4$
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**
Inv.	IGWL	50%	50%	50%	50%	50%
	SchNet	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
	DimeNet	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
	SphereNet	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
	$\text{SchNet}_{\text{full graph}}$	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**
	$\text{SchNet}_{\text{global feat}}$	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**

Table 1: *k-chain geometric graphs.* k-chains are $(\lfloor \frac{k}{2} \rfloor + 1)$-hop distinguishable and $(\lfloor \frac{k}{2} \rfloor + 1)$ GWL iterations are theoretically sufficient to distinguish them. We train geometric GNNs with an increasing number of layers to distinguish $k = 4$-chains. **$\mathfrak{G}$-equivariant GNNs may require more iterations that prescribed by GWL, pointing to preliminary evidence of oversquashing when geometric information is propagated across multiple layers using fixed dimensional feature spaces.** IGWL and $\mathfrak{G}$-invariant GNNs are unable to distinguish k-chains for any $k \geq 2$ and $\mathfrak{G} = O(3)$. $\mathfrak{G}$-invariant GNNs with precomputed non-local features (volume of bounding box) or message passing on fully connected graphs can trivially solve the task. Anomalous results are marked in red and expected results in green.

panying codebase, `geometric-gnn-dojo`[1], can be a pedagogical resource for advancing geometric GNN architectures in a principled manner. See Appendix C for details on experimental setup and hyperparameters.

5.1. Experiment in Table 1: Depth, non-local geometric properties, and oversquashing

GWL assumes perfect propagation of $\mathfrak{G}$-equivariant geometric information at each iteration, which implies that the test can be run for any number of iterations without loss of information. In geometric GNNs, $\mathfrak{G}$-equivariant information is propagated via summing features from multiple layers in fixed dimensional spaces, which may lead to distortion or loss of information from distant nodes.

Experiment. To study the practical implications of depth in propagating geometric information beyond local neighbourhoods, we consider k-chain geometric graphs which generalise the examples from Schütt et al. (2021). Each pair of k-chains consists of $k + 2$ nodes with k nodes arranged in a line and differentiated by the orientation of the 2 end points. Thus, k-chain graphs are $(\lfloor \frac{k}{2} \rfloor + 1)$-hop distinguishable, and $(\lfloor \frac{k}{2} \rfloor + 1)$ GWL iterations are theoretically sufficient to distinguish them. In Table 1, we train $\mathfrak{G}$-equivariant and $\mathfrak{G}$-invariant GNNs with an increasing number of layers to distinguish k-chains.

Results. Despite the supposed simplicity of the task, we find that popular $\mathfrak{G}$-equivariant GNNs such as E-GNN (Satorras et al., 2021) and TFN (Thomas et al., 2018) may require more iterations that prescribed by GWL. Notably, for chains larger than $k = 4$, all $\mathfrak{G}$-equivariant GNNs tended to require more than $(\lfloor \frac{k}{2} \rfloor + 1)$ iterations to solve the task. Additionally, IGWL and $\mathfrak{G}$-invariant GNNs are unable to distinguish k-chains.

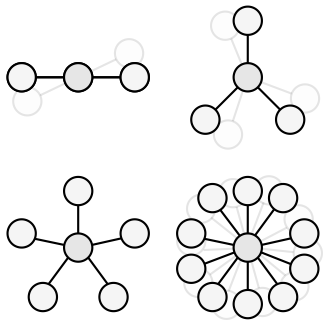
Table 1 points to preliminary evidence of the *oversquashing* phenomenon (Alon & Yahav, 2021; Topping et al., 2022) for equivariant features in geometric GNNs. The issue is most evident for E-GNN, which uses a single vector feature to aggregate and propagate geometric information. This may have implications in modelling macromolecules where long-range interactions often play important roles.

5.2. Experiment in Table 2: Higher order tensors and rotationally symmetric structures

In addition to perfect propagation, GWL is also able to injectively aggregate $\mathfrak{G}$-equivariant information by making use of an auxiliary nested geometric object $\boldsymbol{g}_i$. On the other hand, $\mathfrak{G}$-equivariant GNNs aggregate geometric information via summing neighbourhood features represented by Cartesian vectors (tensor order 1) or higher order spherical tensors. This choice of basis often comes with trade-offs between computational tractability and empirical performance.

Experiment. To demonstrate the utility of higher order tensors in $\mathfrak{G}$-equivariant GNNs, we study how rotational symmetries interact with tensor order. In Table 2, we evaluate current $\mathfrak{G}$-equivariant layers on their ability to distinguish the orientation of structures with rotational symmetry. An

[1]Code available on GitHub: `https://github.com/chaitjo/geometric-gnn-dojo`



	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**

Table 2: *Rotationally symmetric structures.* We train single layer $\mathfrak{G}$-equivariant GNNs to distinguish two *distinct* rotated versions of each L-fold symmetric structure. **We find that layers using order L tensors are unable to identify the orientation of structures with rotation symmetry higher than L-fold.** This issue is particularly prevalent for layers using cartesian vectors (tensor order 1).

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 symmetric structure and train single layer $\mathfrak{G}$-equivariant GNNs to classify the two orientations using the updated equivariant features.

Results. We find that layers using order L spherical tensors are unable to identify the orientation of structures with rotation symmetry higher than L-fold, *i.e.* two distinct rotated versions of the input having the same equivariant features. We attribute this observation to spherical harmonics, which serve as an orthonormal basis for spherical tensor features and exhibit rotational symmetry themselves.

Similar to the Fourier expansion for signals, the spherical harmonic expansion is employed for converting Cartesian vectors to spherical signals in $\mathfrak{G}$-equivariant GNNs. The tensor order of the spherical harmonic bases determines the rate of oscillation of the approximated function on the sphere. We believe that this oscillation rate is closely linked to the rotational fold of a set of symmetric vectors.

In the Fourier expansion, it is not feasible to accurately approximate a high-frequency function solely using low-frequency sinusoidal waves. Similarly, when truncating the spherical harmonic expansion to an order lower than the fold of the rotational symmetry, the rotationally symmetric vectors area act as a higher frequency function. Consequently, the lower frequency bases cannot preserve the orientation of these vectors.

Layers such as E-GNN (Satorras et al., 2021) and GVP-GNN (Jing et al., 2020) using Cartesian vectors are popular as higher order tensors can be computationally intractable for many applications. However, E-GNN and GVP-GNN are particularly poor at discriminating the orientation of rotationally symmetric structures. This may have implications for the modelling of periodic materials which naturally exhibit such symmetries (Levine & Steinhardt, 1984).

6. Discrimination and Universality

Overview. Following Chen et al. (2019), we study the equivalence between the universal approximation capabilities of geometric GNNs (Dym & Maron, 2020; Villar et al., 2021) and the perspective of discriminating geometric graphs introduced by GWL, generalising their results to any isomorphism induced by a compact Lie group $\mathfrak{G}$. We further study the number of invariant aggregators that are required in a continuous setting to distinguish any two neighbourhoods. Proofs are available in Appendix F.

In the interest of generality, we use a general space X acted upon by a compact group $\mathfrak{G}$ and we are interested in the capacity of $\mathfrak{G}$-invariant functions over X to separate points in Y. The restriction to a smaller subset Y is useful because we would like to separately consider the case when Y is countable due to the use of countable features. Therefore, in general, the action of $\mathfrak{G}$ on Y might not be strictly defined since it might yield elements outside Y. For our setting, the reader could take $X = (\mathbb{R}^d \times \mathbb{R}^f)^{n\times n}$ to be the space of geometric graphs and $Y = \mathcal{X}^{n\times n}$, where $\mathcal{X} \subseteq \mathbb{R}^d \times \mathbb{R}^f$.

Definition 14. *Let $\mathfrak{G}$ be a compact group and $\mathcal{C}$ a collection of $\mathfrak{G}$-invariant functions from a set X to $\mathbb{R}$. For a subset $Y \subseteq X$, we say the $\mathcal{C}$ is* ***pairwise $\mathbf{Y}_{\mathfrak{G}}$ discriminating*** *if for any $y_1, y_2 \in Y$ such that $y_1 \not\simeq y_2$, there exists a function $h \in \mathcal{C}$ such that $h(y_1) \neq h(y_2)$.*

We note here that h is not necessarily injective, *i.e.* there might be y_1', y_2' for which $h(y_1') = h(y_1')$. Therefore, pairwise discrimination is a weaker notion of discrimination than the one GWL uses.

Definition 15. *Let $\mathfrak{G}$ be a compact group and $\mathcal{C}$ a collection of $\mathfrak{G}$-invariant functions from X to $\mathbb{R}$. For $Y \subseteq X$, we say the $\mathcal{C}$ is* ***universally approximating*** *over Y if for all $\mathfrak{G}$-invariant functions f from X to $\mathbb{R}$ and for all $\epsilon > 0$, there exists $h_{\varepsilon,f} \in \mathcal{C}$ such that $\|f - h_{\varepsilon,f}\|_Y := \sup_{y\in Y} |f(y) - h(y)| < \varepsilon$.*

Countable Features. We first focus on the countable feature setting that the GWL test operates in. Therefore, we will assume that Y is a countable subset of X.

Theorem 16. *If $\mathcal{C}$ is universally approximating over Y, then $\mathcal{C}$ is also pairwise $Y_{\mathfrak{G}}$ discriminating.*

This result further motivates the interest in discriminating geometric graphs, since a model that cannot distinguish two non-isomorphic geometric graphs is not universal. By further assuming that Y is finite, we obtain a result in the opposite direction. Given a collection of functions $\mathcal{C}$, we define like in Chen et al. (2019) the class $\mathcal{C}^{+L}$ given by all the functions of the form $\mathrm{MLP}([f_1(x), \ldots, f_k(x)])$ with $f_i \in \mathcal{C}$ and finite k, where the MLP has L layers with ReLU hidden activations.

Theorem 17. *If $\mathcal{C}$ is pairwise $Y_{\mathfrak{G}}$ discriminating, then $\mathcal{C}^{+2}$ is universally approximating over Y.*

Continuous Features. The symmetries characterising geometric graphs are naturally continuous (*e.g.* rotations). Therefore, it is natural to ask how the results above translate to *continuous* $\mathfrak{G}$-invariant functions over a continuous subspace Y. Therefore, for the rest of this section, we assume that (X, d) is a metric space, Y is a compact subset of X and $\mathfrak{G}$ acts continuously on X.

Theorem 18. *If $\mathcal{C}$ can universally approximate any continuous $\mathfrak{G}$-invariant functions on Y, then $\mathcal{C}$ is also pairwise $Y_{\mathfrak{G}}$ discriminating.*

With the additional assumption that $X = Y$, we can also prove the converse.

Theorem 19. *If $\mathcal{C}$, a class of functions over Y, is pairwise $Y_{\mathfrak{G}}$ discriminating, then $\mathcal{C}^{+2}$ can also universally approximate any continuous function over Y.*

We now produce an estimate for the number of aggregators needed to learn *continuous* orbit-space injective functions on a manifold X based on results from differential geometry (Lee, 2013). A group $\mathfrak{G}$ acts freely on X if $\mathfrak{g}x = x$ implies $\mathfrak{g} = e_{\mathfrak{G}}$, where $e_{\mathfrak{G}}$ is the identity element in $\mathfrak{G}$.

Theorem 20. *Let X be a smooth n-dim manifold and $\mathfrak{G}$ an m-dim compact Lie group acting continuously on X. Suppose there exists a smooth submanifold Y of X of the same dimension such that $\mathfrak{G}$ acts freely on it. Then, any $\mathfrak{G}$-orbit injective function $f : X \to \mathbb{R}^d$ requires that $d \geq n - m$.*

We now apply this theorem to the local aggregation operation performed by geometric GNNs. Let $X = \mathbb{R}^{n\times d}$ and $\mathfrak{G} = S_n \times O(d)$ or $S_n \times SO(d)$. Let $\boldsymbol{P}_{\mathfrak{g}}$ and $\boldsymbol{Q}_{\mathfrak{g}}$ be the permutation matrix and the orthogonal matrix associated with the group element $\mathfrak{g} \in \mathfrak{G}$. Then $\mathfrak{g}$ acts on matrices $\boldsymbol{X} \in X$ continuously via $\boldsymbol{P}_{\mathfrak{g}}\boldsymbol{X}\boldsymbol{Q}_{\mathfrak{g}}^{\top}$. Then, $\mathfrak{G}$ orbit-space injective functions on X are functions on point clouds of size n that can distinguish any two different point clouds.

Theorem 21. *For $n \geq d - 1 > 0$ or $n = d = 1$, any continuous $S_n \times SO(d)$ orbit-space injective function $f : \mathbb{R}^{n\times d} \to \mathbb{R}^q$ requires that $q \geq nd - d(d-1)/2$.*

We can also generalise this to $O(d)$, with the slightly stronger assumption that $n \geq d$.

Theorem 22. *For $n \geq d > 0$, any continuous $S_n \times O(d)$ orbit-space injective function $f : \mathbb{R}^{n\times d} \to \mathbb{R}^q$ requires that $q \geq nd - d(d-1)/2$.*

Overall, these results show that when working with point clouds in $\mathbb{R}^3$ as is common in molecular or physical applications, at least $q = 3(n-1)$ aggregators are required. This result argues why a bigger representational width can help distinguish neighbourhoods. Finally, in the particular case of the zero-dimensional subgroup $S_n \times \{e_{\mathrm{SO(d)}}\} \simeq S_n$ we obtain a statement holding for all n and generalising a result from Corso et al. (2020) regarding the aggregators for non-geometric GNNs. The original PNA result considers the case $d = 1$ and here we extend it to arbitrary d.

Proposition 23. *Any S_n-invariant injective function $f : \mathbb{R}^{n\times d} \to \mathbb{R}^q$ requires $q \geq nd$.*

7. Conclusion

We propose the Geometric Weisfeiler-Leman (GWL) framework to characterise the expressive power of geometric GNNs. Our work fills a key research gap as standard GNNs and the associated theoretical tools are inapplicable for geometric graphs. (1) We demonstrate the utility of GWL for understanding how key design choices influence geometric GNN expressivity, (2) contribute synthetic experiments highlighting practical challenges for building expressive models, and (3) connect GWL's discrimination-based perspective to universal approximation.

Future Work. GWL provides an abstraction to study the theoretical limits of geometric GNNs. In practice, it is challenging to build provably powerful models that satisfy the conditions of Proposition 2, as GWL relies on perfect neighbourhood aggregation and colouring functions. Based on the understanding gained from GWL, future work will develop maximally powerful, *practical* geometric GNNs.

Acknowledgements. We would like to thank Andrew Blake, Challenger Mishra, Charles Harris, Dávid Kovács, Erik Thiede, Gabor Csanyi, Gregor Simm, Hannes Stärk, Ilyes Batatia, Iulia Duta, Justin Tan, Limei Wang, Mario Geiger, Petar Veličković, Ramon Vinãs, Rob Hesselink, Soledad Villar, Weihua Hu, and the anonymous reviewers for helpful discussions. CKJ was supported by the A*STAR Singapore National Science Scholarship (PhD). SVM was supported by the UKRI Centre for Doctoral Training in Application of Artificial Intelligence to the study of Environmental Risks (EP/S022961/1).

References

Alon, U. and Yahav, E. On the bottleneck of graph neural networks and its practical implications. In *ICLR*, 2021.

Anderson, B., Hy, T. S., and Kondor, R. Cormorant: Covariant molecular neural networks. *NeurIPS*, 2019.

Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 2021.

Bartók, A. P., Kondor, R., and Csányi, G. On representing chemical environments. *Physical Review B*, 2013.

Batatia, I., Batzner, S., Kovács, D. P., Musaelian, A., Simm, G. N., Drautz, R., Ortner, C., Kozinsky, B., and Csányi, G. The design space of e (3)-equivariant atom-centered interatomic potentials. *arXiv preprint*, 2022a.

Batatia, I., Kovács, D. P., Simm, G. N., Ortner, C., and Csányi, G. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. In *NeurIPS*, 2022b.

Batzner, S., Musaelian, A., Sun, L., Geiger, M., Mailoa, J. P., Kornbluth, M., Molinari, N., Smidt, T. E., and Kozinsky, B. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 2022.

Bodnar, C., Frasca, F., Otter, N., Wang, Y., Lio, P., Montufar, G. F., and Bronstein, M. Weisfeiler and lehman go cellular: Cw networks. *NeurIPS*, 2021a.

Bodnar, C., Frasca, F., Wang, Y., Otter, N., Montufar, G. F., Lio, P., and Bronstein, M. Weisfeiler and lehman go topological: Message passing simplicial networks. In *ICML*, 2021b.

Bogatskiy, A., Ganguly, S., Kipf, T., Kondor, R., Miller, D. W., Murnane, D., Offermann, J. T., Pettee, M., Shanahan, P., Shimmin, C., et al. Symmetry group equivariant architectures for physics. *arXiv preprint*, 2022.

Brandstetter, J., Hesselink, R., van der Pol, E., Bekkers, E. J., and Welling, M. Geometric and physical quantities improve e(3) equivariant message passing. In *ICLR*, 2022.

Bredon, G. E. *Topology and geometry*. Springer Science & Business Media, 2013.

Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint*, 2021.

Chanussot, L., Das, A., Goyal, S., Lavril, T., Shuaibi, M., et al. Open catalyst 2020 (oc20) dataset and community challenges. *ACS Catalysis*, 2021.

Chen, Z., Villar, S., Chen, L., and Bruna, J. On the equivalence between graph isomorphism testing and function approximation with gnns. *NeurIPS*, 2019.

Corso, G., Cavalleri, L., Beaini, D., Liò, P., and Veličković, P. Principal neighbourhood aggregation for graph nets. *NeurIPS*, 2020.

Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I., Courbet, A., de Haas, R. J., Bethel, N., et al. Robust deep learning based protein sequence design using proteinmpnn. *Science*, 2022.

Drautz, R. Atomic cluster expansion for accurate and transferable interatomic potentials. *Physical Review B*, 2019.

Du, W., Zhang, H., Du, Y., Meng, Q., Chen, W., Zheng, N., Shao, B., and Liu, T.-Y. Se (3) equivariant graph neural networks with complete local frames. In *ICML*, 2022.

Dusson, G., Bachmayr, M., Csanyi, G., Drautz, R., Etter, S., van der Oord, C., and Ortner, C. Atomic cluster expansion: Completeness, efficiency and stability. *arXiv preprint*, 2019.

Dwivedi, V. P., Joshi, C. K., Luu, A. T., Laurent, T., Bengio, Y., and Bresson, X. Benchmarking graph neural networks. *JMLR*, 2023.

Dym, N. and Maron, H. On the universality of rotation equivariant point cloud networks. In *ICLR*, 2020.

Fey, M. and Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop*, 2019.

Fuchs, F., Worrall, D., Fischer, V., and Welling, M. Se (3)-transformers: 3d roto-translation equivariant attention networks. *NeurIPS*, 2020.

Garg, V., Jegelka, S., and Jaakkola, T. Generalization and representational limits of graph neural networks. In *ICML*, 2020.

Gasteiger, J., Groß, J., and Günnemann, S. Directional message passing for molecular graphs. In *ICLR*, 2020.

Gasteiger, J., Becker, F., and Günnemann, S. Gemnet: Universal directional graph neural networks for molecules. In *NeurIPS*, 2021.

Geiger, M. and Smidt, T. e3nn: Euclidean neural networks. *arXiv preprint*, 2022.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *ICML*, 2017.

Hinton, G., Krizhevsky, A., and Wang, S. D. Transforming auto-encoders. In *ICANN*, 2011.

Jamasb, A. R., Torné, R. V., Ma, E. J., Du, Y., Harris, C., Huang, K., Hall, D., Lio, P., and Blundell, T. L. Graphein - a python library for geometric deep learning and network analysis on biomolecular structures and interaction networks. In *NeurIPS*, 2022.

Jing, B., Eismann, S., Suriana, P., Townshend, R. J. L., and Dror, R. Learning from protein structure with geometric vector perceptrons. In *ICLR*, 2020.

Joshi, C. Transformers are graph neural networks. *The Gradient*, 2020.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 2021.

Lee, J. M. Smooth manifolds. In *Introduction to smooth manifolds*, pp. 540–555. Springer, 2013.

Levine, D. and Steinhardt, P. J. Quasicrystals: a new class of ordered structures. *Physical review letters*, 1984.

Li, Q., Gama, F., Ribeiro, A., and Prorok, A. Graph neural networks for decentralized multi-robot path planning. In *IROS*, 2020.

Liu, Y., Wang, L., Liu, M., Lin, Y., Zhang, X., Oztekin, B., and Ji, S. Spherical message passing for 3d molecular graphs. In *ICLR*, 2022.

Maron, H., Ben-Hamu, H., Serviansky, H., and Lipman, Y. Provably powerful graph networks. *NeurIPS*, 2019.

Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. Weisfeiler and leman go neural: Higher-order graph neural networks. In *AAAI*, 2019.

Pozdnyakov, S. N. and Ceriotti, M. Incompleteness of graph convolutional neural networks for points clouds in three dimensions. *arXiv preprint*, 2022.

Pozdnyakov, S. N., Willatt, M. J., Bartók, A. P., Ortner, C., Csányi, G., and Ceriotti, M. Incompleteness of atomic structure representations. *Physical Review Letters*, 2020.

Read, R. C. and Corneil, D. G. The graph isomorphism disease. *Journal of graph theory*, 1977.

Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., and Battaglia, P. Learning to simulate complex physics with graph networks. In *ICML*, 2020.

Satorras, V. G., Hoogeboom, E., and Welling, M. E (n) equivariant graph neural networks. In *ICML*, 2021.

Schütt, K., Unke, O., and Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *ICML*, 2021.

Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A., and Müller, K.-R. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 2018.

Shapeev, A. V. Moment tensor potentials: A class of systematically improvable interatomic potentials. *Multiscale Modeling & Simulation*, 2016.

Shi, Y., Zheng, S., Ke, G., Shen, Y., You, J., He, J., Luo, S., Liu, C., He, D., and Liu, T.-Y. Benchmarking graphormer on large-scale molecular modeling datasets. *arXiv preprint*, 2022.

Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., and Riley, P. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint*, 2018.

Topping, J., Giovanni, F. D., Chamberlain, B. P., Dong, X., and Bronstein, M. M. Understanding over-squashing and bottlenecks on graphs via curvature. In *ICLR*, 2022.

Villar, S., Hogg, D. W., Storey-Fisher, K., Yao, W., and Blum-Smith, B. Scalars are universal: Equivariant machine learning, structured like classical physics. *NeurIPS*, 2021.

Wang, L., Liu, Y., Lin, Y., Liu, H., and Ji, S. Comenet: Towards complete and efficient message passing for 3d molecular graphs. In *NeurIPS*, 2022.

Wang, L., Liu, H., Liu, Y., Kurtin, J., and Ji, S. Learning hierarchical protein representations via complete 3d graph networks. In *ICLR*, 2023.

Weiler, M., Geiger, M., Welling, M., Boomsma, W., and Cohen, T. S. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. *NeurIPS*, 2018.

Weisfeiler, B. and Leman, A. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series*, 1968.

Xie, T. and Grossman, J. C. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys. Rev. Lett.*, 2018.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *ICLR*, 2019.

Zee, A. *Group theory in a nutshell for physicists*. Princeton University Press, 2016.

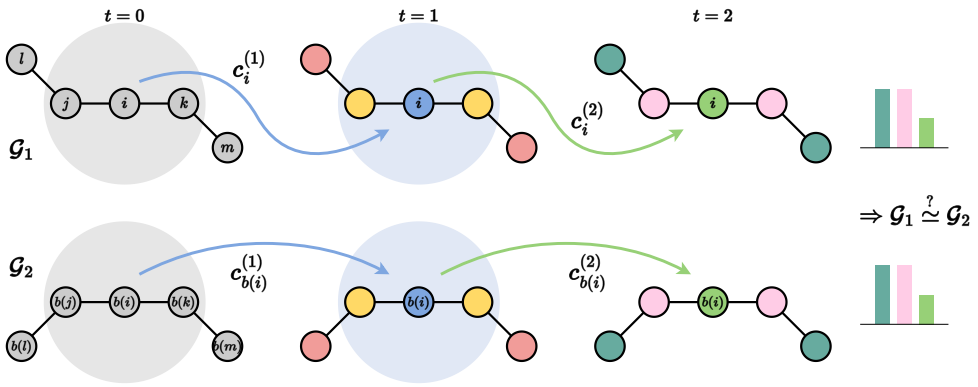


Figure 3: **Invariant GWL Test.** IGWL cannot distinguish $\mathcal{G}_1$ and $\mathcal{G}_2$ as they are 1-hop identical: The $\mathfrak{G}$-orbit of the 1-hop neighbourhood around each node is the same, and IGWL cannot propagate geometric orientation information beyond 1-hop.

A. Discussion

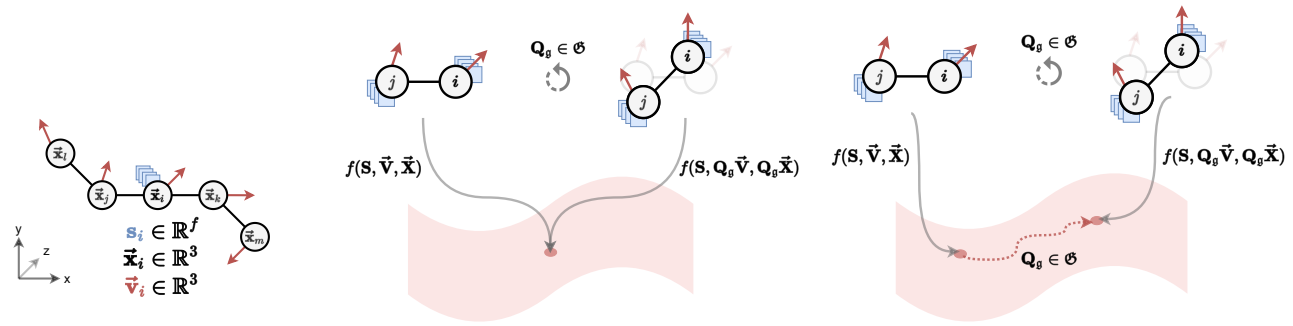
Related Work. In the molecular simulations community, literature on the *completeness* of atom-centred interatomic potentials has focused on distinguishing 1-hop local neighbourhoods (point clouds) around atoms by building spanning sets for continuous, $\mathfrak{G}$-equivariant multiset functions (Shapeev, 2016; Drautz, 2019; Dusson et al., 2019; Pozdnyakov et al., 2020). GWL generalises and extends this analysis to generic geometric graph isomorphism problems beyond local atom-centred neighbourhoods.

Recent theoretical work on geometric GNNs and their universality (Dym & Maron, 2020; Villar et al., 2021; Gasteiger et al., 2021; Jing et al., 2020) has shown that architectures such as TFN, GemNet and GVP-GNN can be universal approximators of continuous, $\mathfrak{G}$-equivariant or $\mathfrak{G}$-invariant multiset functions over point clouds, *i.e.* fully connected graphs. In contrast, the GWL framework studies the expressive power of geometric GNNs operating on *sparse* graphs from the perspective of discriminating geometric graphs. The discrimination lens is potentially more granular and practically insightful than universality. A model may either be universal or not. On the other hand, there could be multiple degrees of discrimination depending on the classes of geometric graphs that can and cannot be distinguished, which our work aims to formalise.

Limitations. *Can GWL distinguish all geometric graphs in theory?* Proposition 7 shows that GWL can distinguish any non-isomorphic geometric graphs for the practical case where the graph structure is computed using radial cutoffs. However, for the general case where the topology of the geometric graph is independent of the coordinates, GWL may not be theoretically complete as there may exist pathological *edge cases* where the test fails. For example, when all coordinates and vector features are set equal to zero, GWL coincides with the standard 1-WL. In this edge case, GWL has the same expressive power as 1-WL and inherits all well-known failure cases of 1-WL.

Does GWL characterise all classes of geometric GNNs? GWL is a node-centric framework for analysing the expressive power of geometric GNNs which perform local message passing *without* canonical reference frames. Non-local architectures such as GemNet-Q (Gasteiger et al., 2021), which aggregates dihedral angles from two-hop neighbourhoods, are not directly covered by GWL. Similarly, the analysis of geometric GNNs based on canonical reference frames (Du et al., 2022; Wang et al., 2022) is left to future work. This class of architectures use a local or global frame of reference to scalarise equivariant features into invariant ones, offering an alternative modelling technique when canonical reference frames can easily be defined (*e.g.* protein backbone structures (Jumper et al., 2021; Baek et al., 2021; Wang et al., 2023)).

Notably, Wang et al. (2022) introduced another notion of *completeness*, different from completeness of atomic structured representations (Pozdnyakov et al., 2020) which GWL builds upon. To the best of our knowledge, building *provably* complete geometric GNNs under either definition for all possible geometric graphs remains an open question that our framework can help the community explore. For instance, Wang et al. (2022) assume that 4-body aggregators such as SphereNet are locally complete. While this is likely the case for many practically occurring geometric graphs, the results in Table 3 for the counterexamples from Pozdnyakov et al. (2020) show that SphereNet is indeed not $\mathfrak{G}$-orbit injective.



(a) A geometric graph (b) $\mathfrak{G}$-invariant function (c) $\mathfrak{G}$-equivariant function

Figure 4: **Invariant and equivariant functions on geometric graphs.** (a) Geometric graphs are attributed graphs embedded in Euclidean space. The geometric attributes transform along with Euclidean transformations of the system, such as global rotations $\mathfrak{G}$. (b) The output of $\mathfrak{G}$-invariant functions remains unchanged under transformations of the input. (c) For $\mathfrak{G}$-equivariant functions, transformations of the input must result in the output transforming equivalently.

B. Additional Background on Geometric Graph Neural Networks

GNNs specialised for geometric graphs can be categorised according to the type of intermediate representations: (1) *Equivariant models*, where the intermediate features and propagated messages are geometric quantities; and (2) *Invariant models*, which only propagate local invariant scalar features such as distances and angles. As summarised in Figure 1, the GWL framework can be used to characterise the expressive power of the following classes of geometric GNNs.

B.1. $\mathfrak{G}$-invariant GNNs

$\mathfrak{G}$-invariant GNN layers aggregate scalar quantities from local neighbourhoods via scalarising the geometric information. Scalar features are update from iteration t to $t+1$ via learnable aggregate and update functions, AGG and UPD, respectively:

$$\boldsymbol{s}_i^{(t+1)} := \text{UPD}\left(\boldsymbol{s}_i^{(t)}\,,\ \text{AGG}\left(\{\!\{(\boldsymbol{s}_i^{(t)}, \vec{\boldsymbol{v}}_i^{(t)}), (\boldsymbol{s}_j^{(t)}, \vec{\boldsymbol{v}}_j^{(t)}), \vec{\boldsymbol{x}}_{ij} \mid j \in \mathcal{N}_i\}\!\}\right)\right). \tag{13}$$

SchNet (Schütt et al., 2018) and **CGCNN** (Xie & Grossman, 2018) use relative distances $\|\vec{\boldsymbol{x}}_{ij}\|$, a 2-body invariant, to scalarise local geometric information:

$$\boldsymbol{s}_i^{(t+1)} := \boldsymbol{s}_i^{(t)} + \sum_{j \in \mathcal{N}_i} f_1\left(\boldsymbol{s}_j^{(t)},\ \|\vec{\boldsymbol{x}}_{ij}\|\right) \tag{14}$$

DimeNet (Gasteiger et al., 2020) and **GemNet-T** (Gasteiger et al., 2021) use both distances and angles $\vec{\boldsymbol{x}}_{ij} \cdot \vec{\boldsymbol{x}}_{ik}$ among triplets (3-body invariants), as follows:

$$\boldsymbol{s}_i^{(t+1)} := \sum_{j \in \mathcal{N}_i} f_1\Big(\boldsymbol{s}_i^{(t)},\ \boldsymbol{s}_j^{(t)},\ \sum_{k \in \mathcal{N}_i \setminus \{j\}} f_2\left(\boldsymbol{s}_j^{(t)},\ \boldsymbol{s}_k^{(t)},\ \|\vec{\boldsymbol{x}}_{ij}\|,\ \vec{\boldsymbol{x}}_{ij} \cdot \vec{\boldsymbol{x}}_{ik}\right)\Big) \tag{15}$$

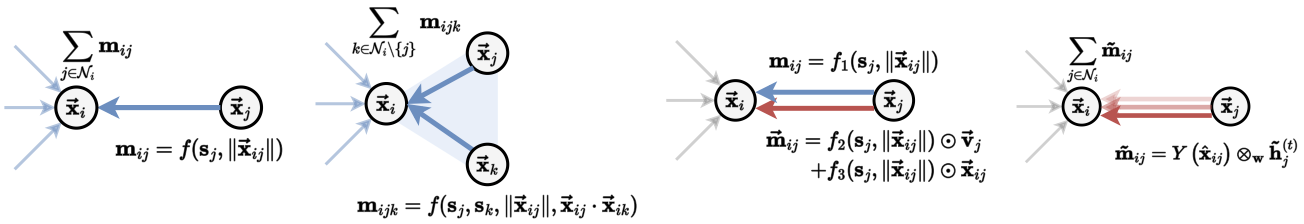
The updated scalar features are both $\mathfrak{G}$-invariant and $T(d)$-invariant as the only geometric information used are the relative distances and angles, both of which remain unchanged under the action of $\mathfrak{G}$ or translations.

B.2. $\mathfrak{G}$-equivariant GNNs using Cartesian vectors

$\mathfrak{G}$-equivariant GNN layers update both scalar and vector features by propagating scalar as well as vector messages, $\boldsymbol{m}_i^{(t)}$ and $\vec{\boldsymbol{m}}_i^{(t)}$, respectively:

$$\boldsymbol{m}_i^{(t)}, \vec{\boldsymbol{m}}_i^{(t)} := \text{AGG}\left(\{\!\{(\boldsymbol{s}_i^{(t)}, \vec{\boldsymbol{v}}_i^{(t)}), (\boldsymbol{s}_j^{(t)}, \vec{\boldsymbol{v}}_j^{(t)}), \vec{\boldsymbol{x}}_{ij} \mid j \in \mathcal{N}_i\}\!\}\right) \qquad \text{(Aggregate)} \tag{16}$$

$$\boldsymbol{s}_i^{(t+1)}, \vec{\boldsymbol{v}}_i^{(t+1)} := \text{UPD}\left((\boldsymbol{s}_i^{(t)}, \vec{\boldsymbol{v}}_i^{(t)})\,,\ (\boldsymbol{m}_i^{(t)}, \vec{\boldsymbol{m}}_i^{(t)})\right) \qquad \text{(Update)} \tag{17}$$



(a) SchNet (b) DimeNet (c) PaiNN (d) Tensor Field Network

Figure 5: **Geometric GNN message passing.** $\mathfrak{G}$-invariant layers only propagate local scalar quantities such as distances (SchNet, equation 14) or distances and angles (DimeNet, equation 15). In contrast, $\mathfrak{G}$-equivariant layers propagated geometric quantities such as vectors and relative positions (PaiNN, equation 19) or higher order tensors (Tensor Field Network, equation 21).

PaiNN (Schütt et al., 2021) interaction layers aggregate scalar and vector messages via learnt filters conditioned on the relative distance (2-body invariants):

$$\boldsymbol{m}_i^{(t)} := \boldsymbol{s}_i^{(t)} + \sum_{j\in\mathcal{N}_i} f_1\left(\boldsymbol{s}_j^{(t)},\ \|\vec{\boldsymbol{x}}_{ij}\|\right) \tag{18}$$

$$\vec{\boldsymbol{m}}_i^{(t)} := \vec{\boldsymbol{v}}_i^{(t)} + \sum_{j\in\mathcal{N}_i} f_2\left(\boldsymbol{s}_j^{(t)},\ \|\vec{\boldsymbol{x}}_{ij}\|\right)\odot\vec{\boldsymbol{v}}_j^{(t)} + \sum_{j\in\mathcal{N}_i} f_3\left(\boldsymbol{s}_j^{(t)},\ \|\vec{\boldsymbol{x}}_{ij}\|\right)\odot\vec{\boldsymbol{x}}_{ij} \tag{19}$$

E-GNN (Satorras et al., 2021) and **GVP-GNN** (Jing et al., 2020) use similar operations. The update step applies a gated non-linearity (Weiler et al., 2018) on the vector features, which learns to scale their magnitude using their norm concatenated with the scalar features:

$$\boldsymbol{s}_i^{(t+1)} := \boldsymbol{m}_i^{(t)} + f_4\left(\boldsymbol{m}_i^{(t)}, \|\vec{\boldsymbol{m}}_i^{(t)}\|\right), \qquad \vec{\boldsymbol{v}}_i^{(t+1)} := \vec{\boldsymbol{m}}_i^{(t)} + f_5\left(\boldsymbol{m}_i^{(t)}, \|\vec{\boldsymbol{m}}_i^{(t)}\|\right)\odot\vec{\boldsymbol{m}}_i^{(t)}. \tag{20}$$

The update step implicitly builds 3-body invariants as taking the norm $\|\vec{\boldsymbol{m}}_i^{(t)}\| = \langle\vec{\boldsymbol{m}}_i^{(t)}, \vec{\boldsymbol{m}}_i^{(t)}\rangle$ computes a weighted sum of inner products $\langle\vec{\boldsymbol{x}}_{ij}, \vec{\boldsymbol{x}}_{ik}\rangle$ for all pairs $j, k \in \mathcal{N}_i$.

The updated scalar features are both $\mathfrak{G}$-invariant and $T(d)$-invariant as the only geometric information used is the relative distances, while the updated vector features are $\mathfrak{G}$-equivariant and $T(d)$-invariant as they aggregate $\mathfrak{G}$-equivariant, $T(d)$-invariant vector quantities from the neighbours.

B.3. $\mathfrak{G}$-equivariant GNNs using spherical tensors

Another example of $\mathfrak{G}$-equivariant GNNs is the `e3nn` framework (Geiger & Smidt, 2022), which can be used to instantiate **Tensor Field Network** (Thomas et al., 2018), **Cormorant** (Anderson et al., 2019), **SE(3)-Transformers** (Fuchs et al., 2020), **SEGNN** (Brandstetter et al., 2022)[2], and **MACE** (Batatia et al., 2022b).

These models use higher order spherical tensors $\tilde{\boldsymbol{h}}_{i,l} \in \mathbb{R}^{2l+1\times f}$ as node feature, starting from order $l = 0$ up to arbitrary $l = L$. The first two orders correspond to scalar features $\boldsymbol{s}_i$ and vector features $\vec{\boldsymbol{v}}_i$, respectively. The higher order tensors $\tilde{\boldsymbol{h}}_i$ are updated via tensor products $\otimes$ of neighbourhood features $\tilde{\boldsymbol{h}}_j$ for all $j \in \mathcal{N}_i$ with the higher order spherical harmonic representations Y of the relative displacement $\frac{\vec{\boldsymbol{x}}_{ij}}{\|\vec{\boldsymbol{x}}_{ij}\|} = \hat{\boldsymbol{x}}_{ij}$:

$$\tilde{\boldsymbol{h}}_i^{(t+1)} := \tilde{\boldsymbol{h}}_i^{(t)} + \sum_{j\in\mathcal{N}_i} Y\left(\hat{\boldsymbol{x}}_{ij}\right)\otimes_{\boldsymbol{w}} \tilde{\boldsymbol{h}}_j^{(t)}, \tag{21}$$

where the weights $\boldsymbol{w}$ of the tensor product are computed via a learnt radial basis function of the relative distance, *i.e.* $\boldsymbol{w} = f\left(\|\vec{\boldsymbol{x}}_{ij}\|\right)$. To obtain the entry $m_3 \in \{-l_3, \ldots, +l_3\}$ for the order-l_3 part of the updated higher order tensors $\tilde{\boldsymbol{h}}_i^{(t+1)}$,

[2]We recommend the appendix of Brandstetter et al. (2022) for a clear and concise introduction to this class of models.

we can expand the tensor product in equation 21 as:

$$\tilde{\boldsymbol{h}}_{i,l_3m_3}^{(t+1)} := \tilde{\boldsymbol{h}}_{i,l_3m_3}^{(t)} + \sum_{l_1m_1,l_2m_2}^{l_3m_3} C_{l_1m_1,l_2m_2}^{l_3m_3} \sum_{j\in\mathcal{N}_i} f_{l_1l_2l_3}\left(\|\vec{\boldsymbol{x}}_{ij}\|\right) Y_{l_1}^{m_1}\left(\hat{\boldsymbol{x}}_{ij}\right) \tilde{\boldsymbol{h}}_{j,l_2m_2}^{(t)}, \tag{22}$$

where $C_{l_1m_1,l_2m_2}^{l_3m_3}$ are the Clebsch-Gordan coefficients ensuring that the updated features are $\mathfrak{G}$-equivariant. Notably, when restricting the tensor product to only scalars (up to $l = 0$), we obtain updates of the form similar to equation 14. Similarly, when using only scalars and vectors (up to $l = 1$), we obtain updates of the form similar to equation 18 and equation 19.

MACE (Batatia et al., 2022b) provides an efficient approach to computing high k-body order features in the `e3nn` framework via Atomic Cluster Expansion (Dusson et al., 2019): They first aggregate neighbourhood features analogous to equation 21 (the A functions in Batatia et al. (2022b) (eq.9)) and then take $k-1$ repeated self-tensor products of these neighbourhood features. In our formalism, this corresponds to:

$$\tilde{\boldsymbol{h}}_i^{(t+1)} := \underbrace{\tilde{\boldsymbol{h}}_i^{(t+1)} \otimes_{\boldsymbol{w}} \cdots \otimes_{\boldsymbol{w}} \tilde{\boldsymbol{h}}_i^{(t+1)}}_{k-1 \text{ times}}, \tag{23}$$

This approach saves the effort of having to symmetrise or generate all k-tuples in more standard many-body expansions *s.a.* DimeNet (Gasteiger et al., 2020). As an analogy, equation 23 amounts to calculating the product $(a + b + \dots)^k$, which implicitly includes terms such as $a^l b^{k-l}$, instead of calculating each of the $a^l b^{k-l}$ terms individually.

C. Details on Synthetic Experiments

Architectures. The `geometric-gnn-dojo`[3] provides unified implementations of several popular geometric GNN architectures characterised by GWL:

- $\mathfrak{G}$-invariant GNNs: SchNet (Schütt et al., 2018), DimeNet (Gasteiger et al., 2020), and SphereNet (Liu et al., 2022);
- $\mathfrak{G}$-equivariant GNNs using cartesian vectors: E-GNN (Satorras et al., 2021) and GVP-GNN (Jing et al., 2020);
- $\mathfrak{G}$-equivariant GNNs using spherical tensors: TFN (Thomas et al., 2018) and MACE (Batatia et al., 2022b).

Tasks. We design synthetic experiments to highlight practical challenges in building expressive geometric GNNs:

- *Distinguishing k-chains*, which test a model's ability to propagate geometric information non-locally and demonstrate geometric oversquashing with increased depth; see Table 1.
- *Rotationally symmetric structures*, which test a layer's ability to identify neighbourhood orientation and highlight the utility of higher order tensors in $\mathfrak{G}$-equivariant GNNs; see Table 2.
- *Counterexamples from Pozdnyakov et al. (2020)*, which test a layer's ability to create distinguishing fingerprints for local neighbourhoods and highlight the need for higher body order of scalarisation; see Table 3.

Hyperparameters. For SchNet and DimeNet, we use the implementation from PyTorch Geometric (Fey & Lenssen, 2019). For SphereNet, E-GNN, GVP-GNN, and MACE, we adapt implementations from the respective authors. Our TFN implementation is based on `e3nn` (Geiger & Smidt, 2022), and we also re-implement MACE by incorporating the `EquivariantProductBasisBlock` from its authors into our TFN layer. We set scalar feature channels to 128 for SchNet, DimeNet, SphereNet, and E-GNN. We set scalar/vector/tensor feature channels to 64 for GVP-GNN, TFN, MACE. TFN and MACE use order $L = 2$ tensors by default. MACE uses local body order 4 by default. We train all models for 100 epochs using the Adam optimiser, with an initial learning rate $1e-4$, which we reduce by a factor of 0.9 and patience of 25 epochs when the performance plateaus. All results are averaged across 10 random seeds.

[3]Code available on GitHub: `https://github.com/chaitjo/geometric-gnn-dojo`

	GNN Layer	Counterexample from Pozdnyakov et al. (2020) 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
	$\text{SphereNet}_{\text{4-body}}$	**100.0 ± 0.0**	**100.0 ± 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**

Table 3: *Counterexamples from Pozdnyakov et al. (2020).* This task evaluates single layer geometric GNNs at distinguishing counterexample structures that are indistinguishable using k-body scalarisation. **Geometric GNN layers with body order k cannot distinguish the corresponding counterexample.** The 3-body counterexample is from Fig.1(b) (Pozdnyakov et al., 2020), 4-body is from Fig.2(f) (Pozdnyakov et al., 2020), and 2-body is based on the two local neighbourhoods in our running example.

D. Geometric GNN Design Space Proofs

D.1. Role of depth (Section 4.1)

The following results are a consequence of the construction of GWL as well as the definitions of k-hop distinct and k-hop identical geometric graphs. Note that k-hop distinct geometric graphs are also $(k+1)$-hop distinct. Similarly, k-hop identical geometric graphs are also $(k-1)$-hop identical, but not necessarily $(k+1)$-hop distinct.

Given two distinct neighbourhoods $\mathcal{N}_1$ and $\mathcal{N}_2$, the $\mathfrak{G}$-orbits of the corresponding geometric multisets $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are mutually exclusive, *i.e.* $\mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_1) \cap \mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_2) \equiv \emptyset$. By the properties of I-HASH this implies $c_1 \neq c_2$. Conversely, if $\mathcal{N}_1$ and $\mathcal{N}_2$ were identical up to group actions, their $\mathfrak{G}$-orbits would overlap, *i.e.* $\boldsymbol{g}_1 = \mathfrak{g}\, \boldsymbol{g}_2$ for some $\mathfrak{g} \in \mathfrak{G}$ and $\mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_1) = \mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_2) \Rightarrow c_1 = c_2$.

Proposition 3. *GWL can distinguish any k-hop distinct geometric graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ where the underlying attributed graphs are isomorphic, and k iterations are sufficient.*

Proof of Proposition 3. The k-th iteration of GWL identifies the $\mathfrak{G}$-orbit of the k-hop subgraph $\mathcal{N}_i^{(k)}$ at each node i via the geometric multiset $\boldsymbol{g}_i^{(k)}$. $\mathcal{G}_1$ and $\mathcal{G}_2$ being k-hop distinct implies that there exists some bijection b and some node $i \in \mathcal{V}_1, b(i) \in \mathcal{V}_2$ such that the corresponding k-hop subgraphs $\mathcal{N}_i^{(k)}$ and $\mathcal{N}_{b(i)}^{(k)}$ are distinct. Thus, the $\mathfrak{G}$-orbits of the corresponding geometric multisets $\boldsymbol{g}_i^{(k)}$ and $\boldsymbol{g}_{b(i)}^{(k)}$ are mutually exclusive, *i.e.* $\mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_i^{(k)}) \cap \mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_{b(i)}^{(k)}) \equiv \emptyset \Rightarrow c_i^{(k)} \neq c_{b(i)}^{(k)}$. Thus, k iterations of GWL are sufficient to distinguish $\mathcal{G}_1$ and $\mathcal{G}_2$. □

Proposition 4. *Up to k iterations, GWL cannot distinguish any k-hop identical geometric graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ where the underlying attributed graphs are isomorphic.*

Proof of Proposition 4. The k-th iteration of GWL identifies the $\mathfrak{G}$-orbit of the k-hop subgraph $\mathcal{N}_i^{(k)}$ at each node i via the geometric multiset $\boldsymbol{g}_i^{(k)}$. $\mathcal{G}_1$ and $\mathcal{G}_2$ being k-hop identical implies that for all bijections b and all nodes $i \in \mathcal{V}_1, b(i) \in \mathcal{V}_2$, the corresponding k-hop subgraphs $\mathcal{N}_i^{(k)}$ and $\mathcal{N}_{b(i)}^{(k)}$ are identical up to group actions. Thus, the $\mathfrak{G}$-orbits of the corresponding geometric multisets $\boldsymbol{g}_i^{(k)}$ and $\boldsymbol{g}_{b(i)}^{(k)}$ overlap, *i.e.* $\mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_i^{(k)}) = \mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_{b(i)}^{(k)}) \Rightarrow c_i^{(k)} = c_{b(i)}^{(k)}$. Thus, up to k iterations of GWL cannot distinguish $\mathcal{G}_1$ and $\mathcal{G}_2$. □

Proposition 5. *IGWL can distinguish any 1-hop distinct geometric graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ where the underlying attributed graphs are isomorphic, and 1 iteration is sufficient.*

Proof of Proposition 5. Each iteration of IGWL identifies the $\mathfrak{G}$-orbit of the 1-hop local neighbourhood $\mathcal{N}_i^{(k=1)}$ at each node i. $\mathcal{G}_1$ and $\mathcal{G}_2$ being 1-hop distinct implies that there exists some bijection b and some node $i \in \mathcal{V}_1, b(i) \in \mathcal{V}_2$ such that the corresponding 1-hop local neighbourhoods $\mathcal{N}_i^{(1)}$ and $\mathcal{N}_{b(i)}^{(1)}$ are distinct. Thus, the $\mathfrak{G}$-orbits of the corresponding geometric multisets $\boldsymbol{g}_i^{(1)}$ and $\boldsymbol{g}_{b(i)}^{(1)}$ are mutually exclusive, *i.e.* $\mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_i^{(1)}) \cap \mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_{b(i)}^{(1)}) \equiv \emptyset \;\Rightarrow\; c_i^{(1)} \neq c_{b(i)}^{(1)}$. Thus, 1 iteration of IGWL is sufficient to distinguish $\mathcal{G}_1$ and $\mathcal{G}_2$. □

Proposition 6. *Any number of iterations of IGWL cannot distinguish any* 1*-hop identical geometric graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ where the underlying attributed graphs are isomorphic.*

Proof of Proposition 6. Each iteration of IGWL identifies the $\mathfrak{G}$-orbit of the 1-hop local neighbourhood $\mathcal{N}_i^{(k=1)}$ at each node i, but cannot identify $\mathfrak{G}$-orbits beyond 1-hop by the construction of IGWL as no geometric information is propagated. $\mathcal{G}_1$ and $\mathcal{G}_2$ being 1-hop identical implies that for all bijections b and all nodes $i \in \mathcal{V}_1, b(i) \in \mathcal{V}_2$, the corresponding 1-hop local neighbourhoods $\mathcal{N}_i^{(k)}$ and $\mathcal{N}_{b(i)}^{(k)}$ are identical up to group actions. Thus, the $\mathfrak{G}$-orbits of the corresponding geometric multisets $\boldsymbol{g}_i^{(1)}$ and $\boldsymbol{g}_{b(i)}^{(1)}$ overlap, *i.e.* $\mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_i^{(1)}) = \mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_{b(i)}^{(1)}) \;\Rightarrow\; c_i^{(k)} = c_{b(i)}^{(k)}$. Thus, any number of IGWL iterations cannot distinguish $\mathcal{G}_1$ and $\mathcal{G}_2$. □

Proposition 7. *Assuming geometric graphs are constructed from point clouds using radial cutoffs, GWL can distinguish any geometric graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ where the underlying attributed graphs are non-isomorphic. At most k_{Max} iterations are sufficient, where k_{Max} is the maximum graph diameter among $\mathcal{G}_1$ and $\mathcal{G}_2$.*

Proof of Proposition 7. We assume that a geometric graph $\mathcal{G} = (\boldsymbol{A}, \boldsymbol{S}, \vec{\boldsymbol{V}}, \vec{\boldsymbol{X}})$ is constructed from a point cloud $(\boldsymbol{S}, \vec{\boldsymbol{V}}, \vec{\boldsymbol{X}})$ using a predetermined radial cutoff r. Thus, the adjacency matrix is defined as $a_{ij} = 1$ if $\|\vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_j\|_2 \leq r$, or 0 otherwise, for all $a_{ij} \in \boldsymbol{A}$. Such construction procedures are conventional for geometric graphs in biochemistry and material science.

Given geometric graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ where the underlying attributed graphs are non-isomorphic, identify k_{Max} the maximum of the graph diameters of $\mathcal{G}_1$ and $\mathcal{G}_2$, and chose any arbitrary nodes $i \in \mathcal{V}_1, j \in \mathcal{V}_2$. We can define the k_{Max}-hop subgraphs $\mathcal{N}_i^{(k_{\text{Max}})}$ and $\mathcal{N}_j^{(k_{\text{Max}})}$ at i and j, respectively. Thus, $\mathcal{N}_i^{(k_{\text{Max}})} = \mathcal{V}_1$ for all $i \in \mathcal{V}_1$, and $\mathcal{N}_j^{(k_{\text{Max}})} = \mathcal{V}_2$ for all $j \in \mathcal{V}_2$. Due to the assumed construction procedure of geometric graphs, $\mathcal{N}_i^{(k_{\text{Max}})}$ and $\mathcal{N}_j^{(k_{\text{Max}})}$ must be distinct. Otherwise, if $\mathcal{N}_i^{(k_{\text{Max}})}$ and $\mathcal{N}_j^{(k_{\text{Max}})}$ were identical up to group actions, the sets $(\boldsymbol{S}_1, \vec{\boldsymbol{V}}_1, \vec{\boldsymbol{X}}_1)$ and $(\boldsymbol{S}_2, \vec{\boldsymbol{V}}_2, \vec{\boldsymbol{X}}_2)$ would have yielded isomorphic graphs.

The k_{Max}-th iteration of GWL identifies the $\mathfrak{G}$-orbit of the k_{Max}-hop subgraph $\mathcal{N}_i^{(k_{\text{Max}})}$ at each node i via the geometric multiset $\boldsymbol{g}_i^{(k_{\text{Max}})}$. As $\mathcal{N}_i^{(k_{\text{Max}})}$ and $\mathcal{N}_j^{(k_{\text{Max}})}$ are distinct for any arbitrary nodes $i \in \mathcal{V}_1, j \in \mathcal{V}_2$, the $\mathfrak{G}$-orbits of the corresponding geometric multisets $\boldsymbol{g}_i^{(k_{\text{Max}})}$ and $\boldsymbol{g}_j^{(k_{\text{Max}})}$ are mutually exclusive, *i.e.* $\mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_i^{(k_{\text{Max}})}) \cap \mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_j^{(k_{\text{Max}})}) \equiv \emptyset \;\Rightarrow\; c_i^{(k_{\text{Max}})} \neq c_j^{(k_{\text{Max}})}$. Thus, k_{Max} iterations of GWL are sufficient to distinguish $\mathcal{G}_1$ and $\mathcal{G}_2$. □

D.2. Limitations of invariant message passing (Section 4.2)

Theorem 8. *GWL is strictly more powerful than IGWL.*

Proof of Theorem 8. Firstly, we can show that the GWL class contains IGWL if GWL can learn the identity when updating $\boldsymbol{g}_i$ for all $i \in \mathcal{V}$, *i.e.* $\boldsymbol{g}_i^{(t)} = \boldsymbol{g}_i^{(t-1)} = \boldsymbol{g}_i^{(0)} \equiv (\boldsymbol{s}_i, \vec{\boldsymbol{v}}_i)$. Thus, GWL is at least as powerful as IGWL, which does not update $\boldsymbol{g}_i$.

Secondly, to show that GWL is strictly more powerful than IGWL, it suffices to show that there exist a pair of geometric graphs that can be distinguished by GWL but not by IGWL. We may consider any k-hop distinct geometric graphs for $k > 1$, where the underlying attributed graphs are isomorphic. Proposition 3 states that GWL can distinguish any such graphs, while Proposition 6 states that IGWL cannot distinguish them. An example is the pair of graphs in Figures 2 and 3. □

Proposition 9. *IGWL and $\mathfrak{G}$-invariant GNNs cannot decide several geometric graph properties: (1) perimeter, surface area, and volume of the bounding box/sphere enclosing the geometric graph; (2) distance from the centroid or centre of mass; and (3) dihedral angles.*

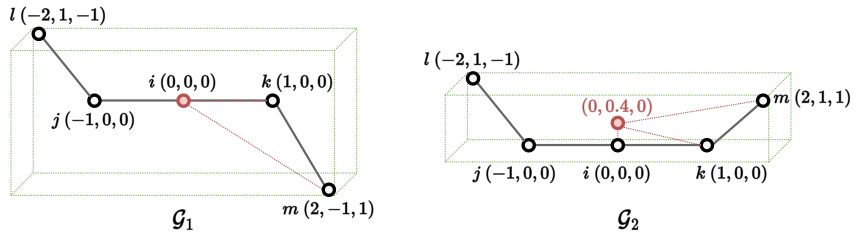


Figure 6: Two geometric graphs for which IGWL and $\mathfrak{G}$-invariant GNNs cannot distinguish their perimeter, surface area, volume of the bounding box/sphere, distance from the centroid, and dihedral angles. The centroid is denoted by a red point and distances from it are denoted by dotted red lines. The bounding box enclosing the geometric graph is denoted by the dotted green lines.

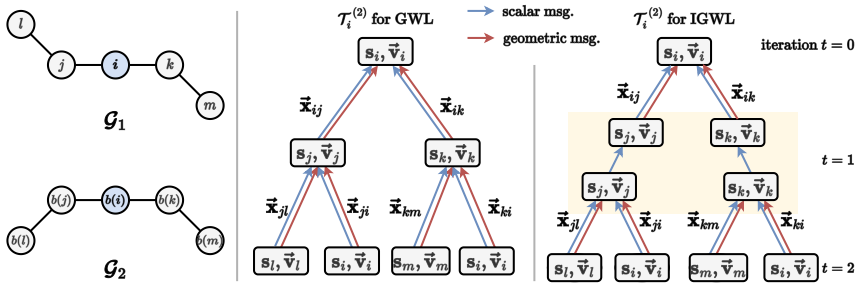


Figure 7: **Geometric Computation Trees for GWL and IGWL.** Unlike GWL, geometric orientation information cannot flow from the leaves to the root in IGWL, restricting its expressive power. IGWL cannot distinguish $\mathcal{G}_1$ and $\mathcal{G}_2$ as all 1-hop neighbourhoods are computationally identical.

Proof of Proposition 9. Following Garg et al. (2020), we say that a class of models *decides* a geometric graph property if there exists a model belonging to this class such that for any two geometric graphs that differ in the property, the model is able to distinguish the two geometric graphs.

In Figure 6, we provide an example of two geometric graphs that demonstrate the proposition. $\mathcal{G}_1$ and $\mathcal{G}_2$ differ in the following geometric graph properties:

- Perimeter, surface area, and volume of the bounding box enclosing the geometric graph[4]: (32 units, 40 units^2, 16 units^3) vs. (28 units, 24 units^2, 8 units^3).
- Multiset of distances from the centroid or centre of mass: $\{0.00, 1.00, 1.00, 2.45, 2.45\}$ vs. $\{0.40, 1.08, 1.08, 2.32, 2.32\}$.
- Dihedral angles: $\angle(ljkm) = \frac{(\vec{\boldsymbol{x}}_{jk} \times \vec{\boldsymbol{x}}_{lj}) \cdot (\vec{\boldsymbol{x}}_{jk} \times \vec{\boldsymbol{x}}_{mk})}{|\vec{\boldsymbol{x}}_{jk} \times \vec{\boldsymbol{x}}_{lj}||\vec{\boldsymbol{x}}_{jk} \times \vec{\boldsymbol{x}}_{mk}|}$ are clearly different for the two graphs.

However, according to Proposition 6 and Theorem 24, both IGWL and $\mathfrak{G}$-invariant GNNs cannot distinguish these two geometric graphs, and therefore, cannot decide all these properties.

We can also show this via a geometric version of computation trees (Garg et al., 2020), for any number of IGWL or $\mathfrak{G}$-invariant GNN iterations, as illustrated in Figure 7. A computation tree $\mathcal{T}_i^{(t)}$ represents the maximum information contained in GWL/IGWL colours or GNN features for node i at iteration t by an 'unrolling' of the message passing procedure. GWL, IGWL, and the corresponding classes of GNNs can be intuitively understood as colouring geometric computation trees.

Geometric computation trees are constructed recursively: $\mathcal{T}_i^{(0)} = (\boldsymbol{s}_i, \vec{\boldsymbol{v}}_i)$ for all $i \in \mathcal{V}$. For $t > 0$, we start with a root node $(\boldsymbol{s}_i, \vec{\boldsymbol{v}}_i)$ and add a child subtree $\mathcal{T}_j^{(t-1)}$ for all $j \in \mathcal{N}_i$ along with the relative position $\vec{\boldsymbol{x}}_{ij}$ along the edge. To obtain the root

[4]The same result applies for the bounding sphere, not shown in the figure.

node's embedding or colour, both scalar and geometric information is propagated from the leaves up to the root. Thus, if two nodes have identical geometric computation trees, they will be mapped to the same node embedding or colour.

Critically, geometric orientation information cannot flow from one level to another in the computation trees for IGWL and $\mathfrak{G}$-invariant GNNs, as they only update scalar information. In the recursive construction procedure, we must insert a connector node $(\boldsymbol{s}_j, \vec{\boldsymbol{v}}_j)$ before adding the child subtree $\mathcal{T}_j^{(t-1)}$ for all $j \in \mathcal{N}_i$ and prevent geometric information propagation between them.

Following the construction procedure for the geometric graphs in Figure 6, we observe that the IGWL computation trees of any pair of isomorphic nodes are identical, as all 1-hop neighbourhoods are computationally identical. Therefore, the set of node colours or node scalar features will also be identical, which implies that $\mathcal{G}_1$ and $\mathcal{G}_2$ cannot be distinguished. □

Proposition 10. *IGWL has the same expressive power as GWL for fully connected geometric graphs.*

Proof of Proposition 10. We will prove by contradiction. Assume that there exist a pair of fully connected geometric graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ which GWL can distinguish, but IGWL cannot.

If the underlying attributed graphs of $\mathcal{G}_1$ and $\mathcal{G}_2$ are isomorphic, by Proposition 3 and Proposition 6, $\mathcal{G}_1$ and $\mathcal{G}_2$ are 1-hop identical but k-hop distinct for some $k > 1$. For all bijections b and all nodes $i \in \mathcal{V}_1, b(i) \in \mathcal{V}_2$, the local neighbourhoods $\mathcal{N}_i^{(1)}$ and $\mathcal{N}_{b(i)}^{(1)}$ are identical up to group actions, and $\mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_i^{(1)}) = \mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_{b(i)}^{(1)}) \Rightarrow c_i^{(1)} = c_{b(i)}^{(1)}$. Additionally, there exists some bijection b and some nodes $i \in \mathcal{V}_1, b(i) \in \mathcal{V}_2$ such that the k-hop subgraphs $\mathcal{N}_i^{(k)}$ and $\mathcal{N}_{b(i)}^{(k)}$ are distinct, and $\mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_i^{(k)}) \cap \mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_{b(i)}^{(k)}) \equiv \emptyset \Rightarrow c_i^{(k)} \neq c_{b(i)}^{(k)}$. However, as $\mathcal{G}_1$ and $\mathcal{G}_2$ are fully connected, for any k, $\mathcal{N}_i^{(1)} = \mathcal{N}_i^{(k)}$ and $\mathcal{N}_{b(i)}^{(1)} = \mathcal{N}_{b(i)}^{(k)}$ are identical up to group actions. Thus, $\mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_i^{(1)}) = \mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_i^{(k)}) = \mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_{b(i)}^{(1)}) = \mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_{b(i)}^{(k)}) \Rightarrow c_i^{(1)} = c_i^{(k)} = c_{b(i)}^{(k)} = c_{b(i)}^{(k)}$. This is a contradiction.

If $\mathcal{G}_1$ and $\mathcal{G}_2$ are non-isomorphic and fully connected, for any arbitrary $i \in \mathcal{V}_1, j \in \mathcal{V}_2$ and any k-hop neighbourhood, we know that $\mathcal{N}_i^{(1)} = \mathcal{N}_i^{(k)}$ and $\mathcal{N}_j^{(1)} = \mathcal{N}_j^{(k)}$. Thus, a single iteration of GWL and IGWL identify the same $\mathfrak{G}$-orbits and assign the same node colours, *i.e.* $\mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_i^{(1)}) = \mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_i^{(k)}) \Rightarrow c_i^{(1)} = c_i^{(k)}$ and $\mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_j^{(1)}) = \mathcal{O}_{\mathfrak{G}}(\boldsymbol{g}_j^{(k)}) \Rightarrow c_j^{(1)} = c_j^{(k)}$. This is a contradiction. □

D.3. Role of scalarisation body order (Section 4.3)

Proposition 11. I-HASH$_{(m)}$ *is $\mathfrak{G}$-orbit injective for $m = max(\{|\mathcal{N}_i| \mid i \in \mathcal{V}\})$, the maximum cardinality of all local neighbourhoods $\mathcal{N}_i$ in a given dataset.*

Proof of Proposition 11. As m is the maximum cardinality of all local neighbourhoods $\mathcal{N}_i$ under consideration, any distinct neighbourhoods $\mathcal{N}_1$ and $\mathcal{N}_2$ must have distinct multisets of m-body scalars. As I-HASH$_{(m)}$ computes scalars involving up to m nodes, it will be able to distinguish any such $\mathcal{N}_1$ and $\mathcal{N}_2$. Thus, I-HASH$_{(m)}$ is $\mathfrak{G}$-orbit injective. □

Proposition 12. *IGWL$_{(k)}$ is at least as powerful as IGWL$_{(k-1)}$. For $k \leq 5$, IGWL$_{(k)}$ is strictly more powerful than IGWL$_{(k-1)}$.*

Proof of Proposition 12. By construction, I-HASH$_{(k)}$ computes $\mathfrak{G}$-invariant scalars from all possible tuples of up to k nodes formed by the elements of a neighbourhood and the central node. Thus, the I-HASH$_{(k)}$ class contains I-HASH$_{(k-1)}$, and I-HASH$_{(k)}$ is at least as powerful as I-HASH$_{(k-1)}$. Thus, the corresponding test IGWL$_{(k)}$ is at least as powerful as IGWL$_{(k-1)}$.

Secondly, to show that IGWL$_{(k)}$ is strictly more powerful than IGWL$_{(k-1)}$ for $k \leq 5$, it suffices to show that there exist a pair of geometric neighbourhoods that can be distinguished by IGWL$_{(k)}$ but not by IGWL$_{(k-1)}$:

- For $k = 3$ and $\mathfrak{G} = O(3)$ or $SO(3)$, for the local neighbourhood from Figure 1 in (Schütt et al., 2021), two configurations with different angles between the neighbouring nodes can be distinguished by IGWL$_{(3)}$ but not by IGWL$_{(2)}$.
- For $k = 4$ and $\mathfrak{G} = O(3)$ or $SO(3)$, the pair of local neighbourhoods from Figure 1 in (Pozdnyakov et al., 2020) can be distinguished by IGWL$_{(4)}$ but not by IGWL$_{(3)}$.

- For $k = 5$ and $\mathfrak{G} = O(3)$, the pair of local neighbourhoods from Figure 2(e) in (Pozdnyakov et al., 2020) can be distinguished by $\text{IGWL}_{(5)}$ but not by $\text{IGWL}_{(4)}$.
- For $k = 5$ and $\mathfrak{G} = SO(3)$, the pair of local neighbourhoods from Figure 2(f) in (Pozdnyakov et al., 2020) can be distinguished by $\text{IGWL}_{(5)}$ but not by $\text{IGWL}_{(4)}$.

□

Proposition 13. *Let $\mathcal{G}_1 = (\boldsymbol{A}_1, \boldsymbol{S}_1, \vec{\boldsymbol{X}}_1)$ and $\mathcal{G}_2 = (\boldsymbol{A}_2, \boldsymbol{S}_2, \vec{\boldsymbol{X}}_2)$ be two geometric graphs with the property that all edges have equal length. Then, $\text{IGWL}_{(2)}$ distinguishes the two graphs if and only if WL can distinguish the attributed graphs $(\boldsymbol{A}_1, \boldsymbol{S}_1)$ and $(\boldsymbol{A}_1, \boldsymbol{S}_1)$.*

Proof of Proposition 13. Let c and k the colours produced by $\text{IGWL}_{(2)}$ and WL, respectively, and let i and j be two nodes belonging to any two graphs like in the statement of the result. We prove the statement inductively.

Clearly, $c_i^{(0)} = k_i^{(0)}$ for all nodes i and $c_i^{(0)} = c_j^{(0)}$ if and only if $k_i^{(0)} = k_j^{(0)}$. Now, assume that the statement holds for iteration t. That is $c_i^{(t)} = c_j^{(t)}$ if and only if $k_i^{(t)} = k_j^{(t)}$ holds for all i. Note that $c_i^{(t+1)} = c_j^{(t+1)}$ if and only if $c_i^{(t)} = c_j^{(t)}$ and $\{\!\{(c_p^{(t)}, \|\vec{\boldsymbol{x}}_{ip}\|) \mid p \in \mathcal{N}_i\}\!\} = \{\!\{(c_p^{(t)}, \|\vec{\boldsymbol{x}}_{jp}\|) \mid p \in \mathcal{N}_j\}\!\}$, since the norm of the relative vectors is the only injective invariant that $\text{IGWL}_{(2)}$ can compute (up to a scaling). Since all the norms are equal, by the induction hypothesis, this is equivalent to $k_i^{(t)} = k_j^{(t)}$ and $\{\!\{k_p^{(t)} \mid p \in \mathcal{N}_i\}\!\} = \{\!\{k^{(t)} \mid p \in \mathcal{N}_j\}\!\}$. Therefore, this is equivalent to $k_i^{(t+1)} = k_j^{(t+1)}$ □

E. Proofs for equivalence between GWL and Geometric GNNs (Section 3.1)

Our proofs adapt the techniques used in Xu et al. (2019); Morris et al. (2019) for connecting WL with GNNs. Note that we omit including the relative position vectors $\vec{\boldsymbol{x}}_{ij}$ in GWL and geometric GNN updates for brevity, as relative positions vectors can be merged into the vector features.

Theorem 1. *Any pair of geometric graphs distinguishable by a $\mathfrak{G}$-equivariant GNN is also distinguishable by GWL.*

Proof of Theorem 1. Consider two geometric graphs $\mathcal{G}$ and $\mathcal{H}$. The theorem implies that if the GNN graph-level readout outputs $f(\mathcal{G}) \neq f(\mathcal{H})$, then the GWL test will always determine $\mathcal{G}$ and $\mathcal{H}$ to be non-isomorphic, *i.e.* $\mathcal{G} \neq \mathcal{H}$.

We will prove by contradiction. Suppose after T iterations, a GNN graph-level readout outputs $f(\mathcal{G}) \neq f(\mathcal{H})$, but the GWL test cannot decide $\mathcal{G}$ and $\mathcal{H}$ are non-isomorphic, *i.e.* $\mathcal{G}$ and $\mathcal{H}$ always have the same collection of node colours for iterations 0 to T. Thus, for iteration t and $t+1$ for any $t = 0 \dots T-1$, $\mathcal{G}$ and $\mathcal{H}$ have the same collection of node colours $\{c_i^{(t)}\}$ as well as the same collection of neighbourhood geometric multisets $\left\{(c_i^{(t)}, \boldsymbol{g}_i^{(t)}),\ \{\!\{(c_j^{(t)}, \boldsymbol{g}_j^{(t)}) \mid j \in \mathcal{N}_i\}\!\}\right\}$ up to group actions. Otherwise, the GWL test would have produced different node colours at iteration $t+1$ for $\mathcal{G}$ and $\mathcal{H}$ as different geometric multisets get unique new colours.

We will show that on the same graph for nodes i and k, if $(c_i^{(t)}, \boldsymbol{g}_i^{(t)}) = (c_k^{(t)}, \mathfrak{g} \cdot \boldsymbol{g}_k^{(t)})$, we always have GNN features $(\boldsymbol{s}_i^{(t)}, \vec{\boldsymbol{v}}_i^{(t)}) = (\boldsymbol{s}_k^{(t)}, \boldsymbol{Q}_{\mathfrak{g}} \vec{\boldsymbol{v}}_k^{(t)})$ for any iteration t. This holds for $t = 0$ because GWL and the GNN start with the same initialisation. Suppose this holds for iteration t. At iteration $t+1$, if for any i and k, $(c_i^{(t+1)}, \boldsymbol{g}_i^{(t+1)}) = (c_k^{(t+1)}, \mathfrak{g} \cdot \boldsymbol{g}_k^{(t+1)})$, then:

$$\left\{(c_i^{(t)}, \boldsymbol{g}_i^{(t)}),\ \{\!\{(c_j^{(t)}, \boldsymbol{g}_j^{(t)}) \mid j \in \mathcal{N}_i\}\!\}\right\} = \left\{(c_k^{(t)}, \mathfrak{g} \cdot \boldsymbol{g}_k^{(t)}),\ \{\!\{(c_j^{(t)}, \mathfrak{g} \cdot \boldsymbol{g}_j^{(t)}) \mid j \in \mathcal{N}_k\}\!\}\right\} \tag{24}$$

By our assumption on iteration t,

$$\left\{(\boldsymbol{s}_i^{(t)}, \vec{\boldsymbol{v}}_i^{(t)}),\ \{\!\{(\boldsymbol{s}_j^{(t)}, \vec{\boldsymbol{v}}_j^{(t)}) \mid j \in \mathcal{N}_i\}\!\}\right\} = \left\{(\boldsymbol{s}_k^{(t)}, \boldsymbol{Q}_{\mathfrak{g}} \vec{\boldsymbol{v}}_k^{(t)}),\ \{\!\{(\boldsymbol{s}_j^{(t)}, \boldsymbol{Q}_{\mathfrak{g}} \vec{\boldsymbol{v}}_j^{(t)}) \mid j \in \mathcal{N}_k\}\!\}\right\} \tag{25}$$

As the same aggregate and update operations are applied at each node within the GNN, the same inputs, *i.e.* neighbourhood features, are mapped to the same output. Thus, $(\boldsymbol{s}_i^{(t+1)}, \vec{\boldsymbol{v}}_i^{(t+1)}) = (\boldsymbol{s}_k^{(t+1)}, \boldsymbol{Q}_{\mathfrak{g}} \vec{\boldsymbol{v}}_k^{(t+1)})$. By induction, if $(c_i^{(t)}, \boldsymbol{g}_i^{(t)}) = (c_k^{(t)}, \mathfrak{g} \cdot \boldsymbol{g}_k^{(t)})$, we always have GNN node features $(\boldsymbol{s}_i^{(t)}, \vec{\boldsymbol{v}}_i^{(t)}) = (\boldsymbol{s}_k^{(t)}, \boldsymbol{Q}_{\mathfrak{g}} \vec{\boldsymbol{v}}_k^{(t)})$ for any iteration t. This creates valid mappings ϕ_s, ϕ_v such that $\boldsymbol{s}_i^{(t)} = \phi_s(c_i^{(t)})$ and $\vec{\boldsymbol{v}}_i^{(t)} = \phi_v(c_i^{(t)}, \boldsymbol{g}_i^{(t)})$ for any $i \in \mathcal{V}$.

Thus, if $\mathcal{G}$ and $\mathcal{H}$ have the same collection of node colours and geometric multisets, then $\mathcal{G}$ and $\mathcal{H}$ also have the same collection of GNN neighbourhood features

$$\left\{(\boldsymbol{s}_i^{(t)}, \vec{\boldsymbol{v}}_i^{(t)}) \ , \ \{\!\{(\boldsymbol{s}_j^{(t)}, \vec{\boldsymbol{v}}_j^{(t)}) \mid j \in \mathcal{N}_i\}\!\}\right\} = \left\{(\phi_s(c_i^{(t)}), \phi_v(c_i^{(t)}, \boldsymbol{g}_i^{(t)})) \ , \ \{\!\{(\phi_s(c_j^{(t)}), \phi_v(c_i^{(t)}, \boldsymbol{g}_i^{(t)})) \mid j \in \mathcal{N}_i\}\!\}\right\}$$

Thus, the GNN will output the same collection of node scalar features $\{\boldsymbol{s}_i^{(T)}\}$ for $\mathcal{G}$ and $\mathcal{H}$ and the permutation-invariant graph-level readout will output $f(\mathcal{G}) = f(\mathcal{H})$. This is a contradiction. □

Similarly, $\mathfrak{G}$-invariant GNNs can be at most as powerful as IGWL.

Theorem 24. *Any pair of geometric graphs distinguishable by a $\mathfrak{G}$-invariant GNN is also distinguishable by IGWL.*

Proof. The proof follows similarly to the proof for Theorem 1. □

Proposition 2. *$\mathfrak{G}$-equivariant GNNs have the same expressive power as GWL if the following conditions hold: (1) The aggregation* AGG *is an injective, $\mathfrak{G}$-equivariant multiset function. (2) The scalar part of the update* UPD_s *is a $\mathfrak{G}$-orbit injective, $\mathfrak{G}$-invariant multiset function. (3) The vector part of the update* UPD_v *is an injective, $\mathfrak{G}$-equivariant multiset function. (4) The graph-level readout f is an injective multiset function.*

Proof of Theorem 2. Consider a GNN where the conditions hold. We will show that, with a sufficient number of iterations t, the output of this GNN is equivalent to GWL, *i.e.* $\boldsymbol{s}^{(t)} \equiv c^{(t)}$.

Let $\mathcal{G}$ and $\mathcal{H}$ be any geometric graphs which the GWL test decides as non-isomorphic at iteration T. Because the graph-level readout function is injective, *i.e.* it maps distinct multiset of node scalar features into unique embeddings, it suffices to show that the GNN's neighbourhood aggregation process, with sufficient iterations, embeds $\mathcal{G}$ and $\mathcal{H}$ into different multisets of node features.

For this proof, we replace $\mathfrak{G}$-orbit injective functions with injective functions over the equivalence class generated by the actions of $\mathfrak{G}$. Thus, all elements belonging to the same $\mathfrak{G}$-orbit will first be mapped to the same representative of the equivalence class, denoted by the square brackets $[\dots]$, followed by an injective map. The result is $\mathfrak{G}$-orbit injective.

Let us assume the GNN updates node scalar and vector features as:

$$\boldsymbol{s}_i^{(t)} = \text{UPD}_s\left(\left[(\boldsymbol{s}_i^{(t-1)}, \vec{\boldsymbol{v}}_i^{(t-1)}) \ , \ \text{AGG}\left(\{\!\{(\boldsymbol{s}_i^{(t-1)}, \boldsymbol{s}_j^{(t-1)}, \vec{\boldsymbol{v}}_i^{(t-1)}, \vec{\boldsymbol{v}}_j^{(t-1)}) \mid j \in \mathcal{N}_i\}\!\}\right)\right]\right) \tag{26}$$

$$\vec{\boldsymbol{v}}_i^{(t)} = \text{UPD}_v\left((\boldsymbol{s}_i^{(t-1)}, \vec{\boldsymbol{v}}_i^{(t-1)}) \ , \ \text{AGG}\left(\{\!\{(\boldsymbol{s}_i^{(t-1)}, \boldsymbol{s}_j^{(t-1)}, \vec{\boldsymbol{v}}_i^{(t-1)}, \vec{\boldsymbol{v}}_j^{(t-1)}) \mid j \in \mathcal{N}_i\}\!\}\right)\right) \tag{27}$$

with the aggregation function AGG being $\mathfrak{G}$-equivariant and injective, the scalar update function UPD_s being $\mathfrak{G}$-invariant and injective, and the vector update function UPD_v being $\mathfrak{G}$-equivariant and injective.

The GWL test updates the node colour $c_i^{(t)}$ and geometric multiset $\boldsymbol{g}_i^{(t)}$ as:

$$c_i^{(t)} = h_s\left(\left[(c_i^{(t-1)}, \boldsymbol{g}_i^{(t-1)}) \ , \ \{\!\{(c_j^{(t-1)}, \boldsymbol{g}_j^{(t-1)}) \mid j \in \mathcal{N}_i\}\!\}\right]\right), \tag{28}$$

$$\boldsymbol{g}_i^{(t)} = h_v\left((c_i^{(t-1)}, \boldsymbol{g}_i^{(t-1)}) \ , \ \{\!\{(c_j^{(t-1)}, \boldsymbol{g}_j^{(t-1)}) \mid j \in \mathcal{N}_i\}\!\}\right), \tag{29}$$

where h_s is a $\mathfrak{G}$-invariant and injective map, and h_v is a $\mathfrak{G}$-equivariant and injective operation (e.g. in equation 6, expanding the geometric multiset by copying).

We will show by induction that at any iteration t, there always exist injective functions φ_s and φ_v such that $\boldsymbol{s}_i^{(t)} = \varphi_s(c_i^{(t)})$ and $\vec{\boldsymbol{v}}_i^{(t)} = \varphi_v(c_i^{(t)}, \boldsymbol{g}_i^{(t)})$. This holds for $t = 0$ because the initial node features are the same for GWL and GNN, $c_i^{(0)} \equiv \boldsymbol{s}_i^{(0)}$ and $\boldsymbol{g}_i^{(0)} \equiv (\boldsymbol{s}_i^{(0)}, \vec{\boldsymbol{v}}_i^{(0)})$ for all $i \in \mathcal{V}(\mathcal{G}), \mathcal{V}(\mathcal{H})$. Suppose this holds for iteration t. At iteration $t+1$, substituting $\boldsymbol{s}_i^{(t)}$ with $\varphi_s(c_i^{(t)})$, and $\vec{\boldsymbol{v}}_i^{(t)}$ with $\varphi_v(c_i^{(t)}, \boldsymbol{g}_i^{(t)})$ gives us

$$\boldsymbol{s}_i^{(t+1)} = \text{UPD}_s\left(\left[(\varphi_s(c_i^{(t)}), \varphi_v(c_i^{(t)}, \boldsymbol{g}_i^{(t)})) \ , \ \text{AGG}\left(\{\!\{(\varphi_s(c_i^{(t)}), \varphi_s(c_j^{(t)}), \varphi_v(c_i^{(t)}, \boldsymbol{g}_i^{(t)}), \varphi_v(c_j^{(t)}, \boldsymbol{g}_j^{(t)})) \mid j \in \mathcal{N}_i\}\!\}\right)\right]\right)$$

$$\vec{\boldsymbol{v}}_i^{(t+1)} = \text{UPD}_v\left((\varphi_s(c_i^{(t)}), \varphi_v(c_i^{(t)}, \boldsymbol{g}_i^{(t)})) \ , \ \text{AGG}\left(\{\!\{(\varphi_s(c_i^{(t)}), \varphi_s(c_j^{(t)}), \varphi_v(c_i^{(t)}, \boldsymbol{g}_i^{(t)}), \varphi_v(c_j^{(t)}, \boldsymbol{g}_j^{(t)})) \mid j \in \mathcal{N}_i\}\!\}\right)\right)$$

The composition of multiple injective functions is injective. Therefore, there exist some injective functions g_s and g_v such that:

$$\boldsymbol{s}_i^{(t+1)} = g_s\left(\left[(c_i^{(t)}, \boldsymbol{g}_i^{(t)}),\ \{\!\{(c_j^{(t)}, \boldsymbol{g}_j^{(t)}) \mid j \in \mathcal{N}_i\}\!\}\right]\right), \tag{30}$$

$$\vec{\boldsymbol{v}}_i^{(t+1)} = g_v\left((c_i^{(t)}, \boldsymbol{g}_i^{(t)}),\ \{\!\{(c_j^{(t)}, \boldsymbol{g}_j^{(t)}) \mid j \in \mathcal{N}_i\}\!\}\right), \tag{31}$$

We can then consider:

$$\boldsymbol{s}_i^{(t+1)} = g_s \circ h_s^{-1}\ h_s\left(\left[(c_i^{(t)}, \boldsymbol{g}_i^{(t)}),\ \{\!\{(c_j^{(t)}, \boldsymbol{g}_j^{(t)}) \mid j \in \mathcal{N}_i\}\!\}\right]\right), \tag{32}$$

$$\vec{\boldsymbol{v}}_i^{(t+1)} = g_v \circ h_v^{-1}\ h_v\left((c_i^{(t)}, \boldsymbol{g}_i^{(t)}),\ \{\!\{(c_j^{(t)}, \boldsymbol{g}_j^{(t)}) \mid j \in \mathcal{N}_i\}\!\}\right), \tag{33}$$

Then, we can denote $\varphi_s = g_s \circ h_s^{-1}$ and $\varphi_v = g_v \circ h_v^{-1}$ as injective functions because the composition of injective functions is injective. Hence, for any iteration $t+1$, there exist injective functions φ_s and φ_v such that $\boldsymbol{s}_i^{(t+1)} = \varphi_s\left(c_i^{(t+1)}\right)$ and $\vec{\boldsymbol{v}}_i^{(t+1)} = \varphi_v\left(c_i^{(t+1)}, \boldsymbol{g}_i^{(t+1)}\right)$.

At the T-th iteration, the GWL test decides that $\mathcal{G}$ and $\mathcal{H}$ are non-isomorphic, which means the multisets of node colours $\{c_i^{(T)}\}$ are different for $\mathcal{G}$ and $\mathcal{H}$. The GNN's node scalar features $\{\boldsymbol{s}_i^{(T)}\} = \{\varphi_s(c_i^{(T)})\}$ must also be different for $\mathcal{G}$ and $\mathcal{H}$ because of the injectivity of φ_s.

□

A weaker set of conditions is sufficient for a $\mathfrak{G}$-invariant GNN to be at least as expressive as IGWL.

Proposition 25. *$\mathfrak{G}$-invariant GNNs have the same expressive power as IGWL if the following conditions hold: (1) The aggregation ψ and update ϕ are $\mathfrak{G}$-orbit injective, $\mathfrak{G}$-invariant multiset functions. (2) The graph-level readout f is an injective multiset function.*

Proof. The proof follows similarly to the proof for Theorem 2. □

F. Universality and Discrimination Proofs (Section 6)

F.1. Equivalence between universality and discrimination

The results in this subsection use the proofs from Chen et al. (2019) with minor adaptations.

Theorem 16. *If $\mathcal{C}$ is universally approximating over Y, then $\mathcal{C}$ is also pairwise $Y_{\mathfrak{G}}$ discriminating.*

Proof. Given any $y \in Y$, we can construct the $\mathfrak{G}$-invariant function over X, $\delta_y(x) = 0$ if $y \simeq x$ and 1 otherwise. Therefore, δ_y can be approximated with some $\epsilon < 0.5$ over Y by some function $h \in \mathcal{C}$. Hence, $h(y) \neq h(y')$ for any $y, y' \in Y$ and $\mathcal{C}$ is pairwise $Y_{\mathfrak{G}}$ discriminating. □

The following two Lemmas follow from Chen et al. (2019) with minor adaptations.

Lemma 26. *If $\mathcal{C}$ is pairwise $Y_{\mathfrak{G}}$ discriminating, then for all $y \in Y$, there exists a function $\delta_y \in \mathcal{C}^{+1}$ such that for all y', $\delta_y(y') = 0$ if and only if $y \simeq y'$.*

Proof. For any $y_1, y_2 \in Y$ such that $y_1 \not\simeq y_2$, let δ_{y_1,y_2} be the function that distinguishes y_1, y_2. That is $\delta_{y_1,y_2}(y_1) \neq \delta_{y_1,y_2}(y_2)$. Then, we can define a function $\overline{\delta}_{y,y'} \in \mathcal{C}$:

$$\overline{\delta}_{y,y'}(x) = |\delta_{y,y'}(x) - \delta_{y,y'}(y)| \to \begin{cases} = 0 & \text{if } x \simeq y \\ > 0 & \text{if } x \simeq y' \\ \geq 0 & \text{otherwise} \end{cases} \tag{34}$$

This function is already similar to the δ_y function whose existence we want to prove. To obtain a function that is strictly positive over all the $x \in Y$ with $x \not\simeq y$, we can construct δ_y as a sum over all the $\overline{\delta}_{y,y'}$:

$$\delta_y(x) = \sum_{y' \in Y, y' \not\simeq y} \overline{\delta}_{y,y'}(x) \to \begin{cases} = 0 & \text{if } x \simeq y \\ > 0 & \text{if } x \not\simeq y \text{ and } x \in \mathcal{O}_{\mathfrak{G}}(Y) \supseteq Y \\ \geq 0 & \text{otherwise} \end{cases} \tag{35}$$

Given the finite set of functions $\{\delta_{y,y'}\}$, notice that

$$\overline{\delta}_{y,y'}(x) = \text{ReLU}\big(\delta_{y,y'}(x) - \delta_{y,y'}(y)\big) \ + \ \text{ReLU}\big(\delta_{y,y'}(y) - \delta_{y,y'}(x)\big).$$

Then δ_y is obtained by summing all these functions over $y' \in Y$ with $y' \not\simeq y$, so $\delta_y \in \mathcal{C}^{+1}$. □

Lemma 27. *Let $\mathcal{C}$ is a class of $\mathfrak{G}$-invariant functions from $X \to \mathbb{R}$ such that for any $y, y' \in Y \subseteq X$, where Y is finite, there is a $\delta_y \in \mathcal{C}$ with the property $\delta_y(y') = 0$ if and only if $y \simeq y'$. Then $\mathcal{C}^{+1}$ is universally approximating over Y.*

Proof. For $y \in Y$, define $r_y := \frac{1}{2}\min_{y' \in Y, y' \not\simeq y} \delta_y(y')$. Define the bump function with radius $r > 0$, $b_r : \mathbb{R} \to \mathbb{R}$ as $b_r(s) = \psi(\frac{s}{r})$, where

$$\psi(z) = \text{ReLU}(z+1) + \text{ReLU}(1-z) - 2\text{ReLU}(z).$$

Define $k_y := |Y \cap \mathcal{O}_{\mathfrak{G}}(y)|^{-1}$. Since Y is finite and the intersection with the orbit of y contains y, k_y is finite and well-defined. We can define the $\mathfrak{G}$-invariant function h from X to $\mathbb{R}$ as:

$$h(x) = \sum_{y \in Y} k_y f(y) b_{r_y}(\delta_y(x)) \tag{36}$$

Notice that $h|_Y = f|_Y$ and $h \in \mathcal{C}^{+1}$. Therefore, $\mathcal{C}^{+1}$ is universally approximating. □

Theorem 17. *If $\mathcal{C}$ is pairwise $Y_{\mathfrak{G}}$ discriminating, then $\mathcal{C}^{+2}$ is universally approximating over Y.*

Proof. Result follows directly from the two Lemmas above. □

Lemma 28. *Let X, Y be topological spaces and $h : X \times Y \to \mathbb{R}$ a continuous function. Then, if Y is compact, $f(x) = \inf_{y \in Y} h(x, y)$ is continuous.*

Proof. The open sets $(-\infty, a)$ and (b, ∞) form a basis for the topology of $\mathbb{R}$. Thus, we show that their preimage under f is open. First, notice $x \in f^{-1}((-\infty, a))$ if and only if $(x, y) \in h^{-1}((-\infty, a))$ for some $y \in Y$. Therefore, $f^{-1}((-\infty, a)) = p_X(h^{-1}((-\infty, a)))$, where $p_X : X \times Y \to X$ is the function projecting in the first argument. Since p_X is continuous and open, it follows $p_X(h^{-1}((-\infty, a)))$ is open.

When $x \in f^{-1}((b, \infty))$, it implies that for all $y \in Y, h(x, y) > b$. This means that for all $x \in f^{-1}((b, \infty))$ and $y \in Y$, we have $(x, y) \in h^{-1}((b, \infty))$. Since $h^{-1}((b, \infty)$ is open, then there exists an open box $U_{x,y} \times V_{x,y} \subseteq h^{-1}((b, \infty)$ containing (x, y). Then, the union $\cup_{y \in Y}\big(U_{x,y} \times V_{x,y}\big)$ covers $\{x\} \times Y$. Since Y is compact, there exists a finite subcover $\cup_{y_k}^{K_x}\big(U_{x,y_k} \times V_{x,y_k}\big)$ of size K_x. Then notice that the open set $A_x := \cap_{y_k}^{K_x} U_{x,y_k}$ is a neighbourhood around x and $A_x \times Y \subseteq h^{-1}((b, \infty))$. Therefore, $A_x \subseteq f^{-1}((b, \infty))$ and since $x \in f^{-1}((b, \infty))$ was chosen arbitrarily, $f^{-1}((b, \infty))$ is open. □

Theorem 18. *If $\mathcal{C}$ can universally approximate any continuous $\mathfrak{G}$-invariant functions on Y, then $\mathcal{C}$ is also pairwise $Y_{\mathfrak{G}}$ discriminating.*

Proof. Consider $y, y' \in Y$ such that $y \not\simeq y'$. Then, the function $\delta_y(x) = \inf_{\mathfrak{g} \in \mathfrak{G}} d(y, \mathfrak{g}x) = \min_{\mathfrak{g} \in \mathfrak{G}} d(y, \mathfrak{g}x) > 0$, where the second equality follows from the compactness of $\mathfrak{G}$. This function is $\mathfrak{G}$-invariant. To show that it is continuous, notice that the function $h(x, \mathfrak{g}) = d(y, \mathfrak{g}x)$ is given by the composition $d_y \circ a$, where $a : X \times \mathfrak{G} \to X$ is the continuous group action and $d_y : X \to \mathbb{R}$ is given by $d_y(x) = d(y, x)$, which is also continuous. Since composition of continuous functions is continuous and $\delta_y(x) = \inf_{\mathfrak{g} \in \mathfrak{G}} h(x, \mathfrak{G})$, it follows from Lemma 28 that δ_y is a continuous function.

Given a universally approximating class of functions $\mathcal{C}$, we can find a function f approximating δ_y with precision $\epsilon < \frac{\delta_y(y')}{2}$ and, therefore, $f(y) \neq f(y')$. □

Definition 29. *Let $\mathcal{C}$ be a class of functions $X \to \mathbb{R}$ and $Y \subseteq X$. We say that $\mathcal{C}$ can locate every orbit over Y if for any $y \in Y$ and any $\varepsilon > 0$ there exists $\delta_y \in \mathcal{C}$ such that:*

1. *For all $y' \in Y, \delta_y(y') \geq 0$.*
2. *For all $y' \in Y$, if $y \simeq y'$, then $\delta_y(y') = 0$.*
3. *There exists $r_y > 0$ such that if $\delta_y(y') < r_y$ for any $y' \in Y$, then there is a $\mathfrak{g} \in \mathfrak{G}$ such that $d(y', \mathfrak{g} \cdot y) < \varepsilon$.*

Notice that since $\delta_y \in \mathcal{C}$, it is $\mathfrak{G}$-invariant and then for any $y^* \in \mathcal{O}_{\mathfrak{G}}(y')$, $\delta_y(y') = \delta_y(y^*)$ and there exists $\mathfrak{g} \in \mathfrak{G}$ such that $d(y^*, \mathfrak{g} \cdot y) < \varepsilon$. Therefore, intuitively one should see δ_y as some sort of "distance function" measuring how far all $y^* \in \mathcal{O}_{\mathfrak{G}}(y')$ are from the orbit of y. In other words, when $\delta_y(y^*)$ is low, it means that the entire orbit of y^* is close to the orbit of y.

Lemma 30. *If $\mathcal{C}$ is a collection of continuous $\mathfrak{G}$-invariant functions from $X \to \mathbb{R}$ that is pairwise $Y_{\mathfrak{G}}$-discriminating, then $\mathcal{C}^{+1}$ is able to locate every orbit over Y.*

Proof. Select an arbitrary $y \in Y$. For $y' \not\simeq y$, let $\delta_{y,y'}$ be the function in $\mathcal{C}$ separating y and y'. Consider the radius $r_{y,y'} := \frac{1}{2}|\delta_{y,y'}(y) - \delta_{y,y'}(y')| > 0$ and define the set

$$A_{y'} := \delta_{y,y'}^{-1}\left(\delta_{y,y'}(y') - r_y, \delta_{y,y'}(y') + r_y\right)$$

Since $\delta_{y,y'}$ is continuous, $A_{y'}$ is open. If $y' \simeq y$, then we define $A_y = B(y, \varepsilon)$, the open ball in X centred at y with radius ε. Clearly, $\bigcup_{y' \in Y} A_{y'}$ forms a cover for Y. Since Y is compact, then there exists a finite subcover given by a finite subset $Y_0 \subseteq Y$ such that $\bigcup_{y' \in Y_0} A_{y'}$.

We construct the function $\delta_y(y')$ over X as $\delta_y(y') := \sum_{y' \in Y_0 \setminus \mathcal{O}_{\mathfrak{G}}(y^*)} \bar{\delta}_{y,y'}(y^*)$, where $\bar{\delta}_{y,y'}(y^*) := \max(\frac{4}{3}r_{y,y'} - |\delta_{y,y'}(y^*) - \delta_{y,y'}(y')|, 0)$. Since $\delta_{y,y'}$ is continuous and $\mathfrak{G}$-invariant, so is δ_y. Finally, it can be shown that δ_y can indeed locate the orbit of y over Y.

1. Clearly, $\delta_y(x) \geq 0$ for any $x \in X$.
2. For any $y^* \in Y$, if $y \simeq y^*$, then because $\delta_{y,y'}$ is $\mathfrak{G}$-invariant, we have $\delta_{y,y'}(y^*) = \delta_{y,y'}(y)$, so $\delta_y(y^*) = 0$.
3. First, consider $y^* \in Y$ such that $\forall \mathfrak{g} \in \mathfrak{G}$, $d(\mathfrak{g} \cdot y^*, y) \geq \varepsilon$. Then, $y^* \in Y \setminus \bigcup_{y' \in \mathcal{O}_{\mathfrak{G}}(y)} A_{y'}$. Then, there must be a $y' \in Y_0 \setminus \mathcal{O}_{\mathfrak{G}}(y)$, such that $y^* \in A_{y'}$. Therefore, $|\delta_{y,y'}(y^*) - \delta_{y,y'}(y')| < r_{y,y'} < \frac{4}{3}r_{y,y'}$. Then, we have $\frac{4}{3}r_{y,y'} - |\delta_{y,y'}(y^*) - \delta_{y,y'}(y')| > \frac{4}{3}r_{y,y'} - r_{y,y'} = \frac{1}{3}r_{y,y'} > 0 \Rightarrow \bar{\delta}_{y,y'}(y^*) > \frac{1}{3}r_{y,y'}$. Therefore, we can set $r_y := \frac{1}{3}\min_{y' \in Y_0 \setminus \mathcal{O}_{\mathfrak{G}}(y)} r_{y,y'} > 0$. If $\delta_y(y^*) < r_y$ it follows that for all $y' \in Y_0 \setminus \mathcal{O}_{\mathfrak{G}}(y^*)$, $\bar{\delta}_{y,y'}(y^*) < \frac{1}{3}r_{y,y'}$, which implies $y^* \in \bigcup_{\mathfrak{g} \in \mathfrak{G}} B(\mathfrak{g} \cdot y, \varepsilon)$. Finally, this proves there is a $\mathfrak{g} \in \mathfrak{G}$ such that $d(y^*, \mathfrak{g} \cdot y) < \varepsilon$.

Since the absolute value function can be realised using ReLU activations, it is easy to see that $\delta_y \in \mathcal{C}^{+1}$. □

Lemma 31. *If $\mathcal{C}$, a class of functions over a compact set Y, can locate every isomorphism class, then $\mathcal{C}^{+2}$ is universal approximating over Y.*

Proof. Consider any continuous and $\mathfrak{G}$-invariant function on X. Since $Y \subseteq X$ is compact, then f is uniformly continuous when restricted to Y. In other words, for all $\varepsilon > 0$, there exists $r > 0$, such that for all $y_1, y_2 \in Y$, if $d(y_1, y_2) < r$, then $|f(y_1) - f(y_2)| < \varepsilon$.

Let $y \in Y$ and define $B_{\mathfrak{G}}(y, r) := \bigcup_{y' \in \mathcal{O}_{\mathfrak{G}}(y)} B(y', r)$ to be the union of all the open balls of radius r on the orbit of y. Using the function δ_y from Definition 29, there exists r_y such that $\delta_y^{-1}([0, r_y)) \subseteq B_{\mathfrak{G}}(y, r)$ for any $y \in Y$. Since δ_y is continuous, $\delta_y^{-1}([0, r_y))$ is open. Therefore, $\{\delta_y^{-1}([0, r_y))\}_{y \in Y}$ is an open cover for Y. Since Y is compact, we can find a finite subcover $\{\delta_y^{-1}([0, r_y))\}_{y \in Y_0}$, where Y_0 is a finite subset of Y.

We can now use the functions δ_y to construct a set of continuous $\mathfrak{G}$-invariant functions that forms a partition of unity for this finite cover. For $y_0 \in Y_0$ we construct the function $\phi_{y_0}(y') = \max(r_{y_0} - \delta_{y_0}(y'), 0)$ and the function $\phi(y') =$

$\sum_{y^* \in Y_0} \phi_{y^*}(y')$, both of which are continuous. Noticing that $\mathrm{supp}(\phi_{y_0}) = \delta_{y_0}^{-1}([0, r_{y_0}))$ and that $\phi_{y^*}(y') > 0$ for any $y' \in Y$, the set of functions $\psi_{y_0}(y') = \frac{\phi_{y_0}(y')}{\phi(y')}$ form a partition of unity with $\sum_{y_0 \in Y_0} \psi_{y_0}(y') = 1$ for all $y' \in Y$.

Notice that we can write any $\mathfrak{G}$-invariant function f as:

$$f(y') = f(y') \sum_{y_0 \in Y_0} \psi_{y_0}(y') = \sum_{y_0 \in Y_0 \,:\, y' \in \delta_{y_0}^{-1}([0, r_{y_0}))} f(y')\psi_{y_0}(y') \tag{37}$$

The intuition is that because f is continuous, we can approximate $f(y')$ in the expression above by the value of $f(y_0)$ since y' is in the neighbourhood of some y_0. Thus, the function that approximates f is $h(y') = \sum_{y_0 \in Y_0} f(y_0)\psi_{y_0}(y')$. We now show that h can approximate f with arbitrary accuracy.

If $y' \in \delta_{y_0}^{-1}([0, r_{y_0}))$, then there exists $\mathfrak{g} \in \mathfrak{G}$ such that $d(y', \mathfrak{g} \cdot y_0) < r$. Using the fact that f is continous, this implies $|f(y') - f(\mathfrak{g} \cdot y_0)| < \varepsilon$. Because f is invariant, $f(y_0) = f(\mathfrak{g} \cdot y_0)$, which implies $|f(y') - f(y_0)| < \varepsilon$. Then we have:

$$\left|f(y') - \sum_{y_0 \in Y_0} f(y_0)\psi_{y_0}(y')\right| = \left|f(y') - \sum_{y_0 \in Y_0 \,:\, y' \in \delta_{y_0}^{-1}([0, r_{y_0}))} f(y_0)\psi_{y_0}(y')\right| \tag{38}$$

$$= \sum_{y_0 \in Y_0 \,:\, y' \in \delta_{y_0}^{-1}([0, r_{y_0}))} |f(y') - f(y_0)|\, \psi_{y_0}(y') < \varepsilon \tag{39}$$

Finally, to see that h is in $\mathcal{C} + 2$, we can use an MLP with one hidden layer to approximate ψ_{y_0} followed by one final layer to compute the linear combination of the ψ_{y_0}. □

Theorem 19. *If $\mathcal{C}$, a class of functions over Y, is pairwise $Y_{\mathfrak{G}}$ discriminating, then $\mathcal{C}^{+2}$ can also universally approximate any continuous function over Y.*

Proof. The proof follows from the two Lemmas above. □

F.2. Number of aggregators in continuous setting

Theorem 20. *Let X be a smooth n-dim manifold and $\mathfrak{G}$ an m-dim compact Lie group acting continuously on X. Suppose there exists a smooth submanifold Y of X of the same dimension such that $\mathfrak{G}$ acts freely on it. Then, any $\mathfrak{G}$-orbit injective function $f : X \to \mathbb{R}^d$ requires that $d \geq n - m$.*

Proof. Suppose for the sake of contradiction that there exists an orbit-space injective and continuous function $f : X \to \mathbb{R}^m$, with $m < d$. Since Y is a submanifold of the same dimension as X, then f must also be injective over Y. By the Quotient Manifold Theorem (Lee, 2013), $Y/\mathfrak{G}$ is a topological manifold of dimension $d = \dim X - \dim \mathfrak{G}$. The map f induces an injective function $g : Y/\mathfrak{G} \to \mathbb{R}^m$. This map is also continuous because for an open set $V \in \mathbb{R}^m$, $g^{-1}(V) = \pi_Y(f^{-1}(V))$. Because f is continuous and π_Y is an open map, this set is open.

$$\begin{tikzcd}
 & Y \arrow[d, "\pi_Y"] \arrow[dr, "f"] & \\
\mathbb{R}^d \arrow[r, "\psi^{-1}"] & Y/\mathfrak{G} \arrow[r, "g"] & \mathbb{R}^m
\end{tikzcd}$$

Because $Y/\mathfrak{G}$ is a manifold, there exist an open set $U \subseteq Y/\mathfrak{G}$ and a homeomorphism $\psi : U \to \mathbb{R}^d$. Then the composition $h = g \circ \psi^{-1}$ is a continuous and injective map from $\psi(U) \subseteq \mathbb{R}^m$ to $\mathbb{R}^m$. By the Invariance of Domain Theorem (Bredon, 2013) (Corollary 19.9), h is open and it is a homeomorphism onto its image $h(\psi(U)) \subseteq \mathbb{R}^m$. By the Invariance of Dimension Theorem (Bredon, 2013) (Corollary 19.10), $d = m$. □

Theorem 21. *For $n \geq d - 1 > 0$ or $n = d = 1$, any continuous $S_n \times SO(d)$ orbit-space injective function $f : \mathbb{R}^{n \times d} \to \mathbb{R}^q$ requires that $q \geq nd - d(d-1)/2$.*

Proof. We now consider the case when $\mathfrak{G} = S_n \times SO(d)$. First, notice that the proof above also holds for this group since it is a subgroup of $S_n \times O(d)$. However, we can obtain a stronger result and show the result holds for $n \geq d-1$. In what follows, we reuse the notation from the proof above.

We define the set

$$Z' = \{\boldsymbol{X} \in X \mid \exists\, 1 \leq i_1 < \ldots < i_{d-1} \leq n \text{ s.t } \boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_{d-1}} \text{ are linearly independent } \},$$

containing $d-1$ row-vectors that are linearly independent. Define $M_{\boldsymbol{X}}$ to be the set of all $(d-1) \times (d-1)$ minors of the matrix $\boldsymbol{X}$. Then, we can construct a continuous function $h(\boldsymbol{X}) = \max_{m \in M_{\boldsymbol{X}}} |m|$ and notice that Z' coincides with the open set $h^{-1}((0,\infty))$. Then, the set $V = Y \cap Z'$ is also open and non-empty. Therefore, V is a submanifold of X of the same dimension and the action of $\mathfrak{G}$ is well-defined and continuous on W. We can show again this action is free.

As in the proof above, we have that $\boldsymbol{P}_{\mathfrak{g}} = \boldsymbol{I}_n$, so it remains to inspect the case $\boldsymbol{X} = \boldsymbol{X}\boldsymbol{Q}_{\mathfrak{g}}$. Any non-trivial rotation $\boldsymbol{Q}_{\mathfrak{g}}$ must rotate at least a two-dimensional subspace of $\mathbb{R}^d$. Since the rows of the matrix $\boldsymbol{X}$ span a $(d-1)$-dimensional subspace of $\mathbb{R}^d$, then $\boldsymbol{Q}_{\mathfrak{g}}$ cannot leave $\boldsymbol{X}$ invariant unless $\boldsymbol{Q}_{\mathfrak{g}} = \boldsymbol{I}_d$. Applying Theorem 20 again yields the result.

For $n = d = 1$, $\|\cdot\| : \mathbb{R}^{1\times d} \to \mathbb{R}$ is, as before, $\mathfrak{G}$ orbit-space injective. □

Theorem 22. *For $n \geq d > 0$, any continuous $S_n \times O(d)$ orbit-space injective function $f : \mathbb{R}^{n\times d} \to \mathbb{R}^q$ requires that $q \geq nd - d(d-1)/2$.*

Proof. First, suppose that $n \geq d > 1$. Consider the subspace $Y = \{\boldsymbol{X} \in X \mid \|\boldsymbol{x}_i\| \neq \|\boldsymbol{x}_j\|, \forall i < j\}$, where the norm is just the standard Euclidean norm. Consider the function $g : X \to \mathbb{R}$ given by $g(\boldsymbol{X}) = \min_{i<j} |\|\boldsymbol{x}_i\| - \|\boldsymbol{x}_j\||$. By standard analysis, this function is continuous and notice that $Y = g^{-1}((0,\infty))$, which means that Y is open in X. We also define the set

$$Z = \{\boldsymbol{X} \in X \mid \exists\, i_1 < \ldots < i_d < n, |\det(\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_d})| > 0\},$$

containing row-vectors that span $\mathbb{R}^d$. As above, this set is the preimage of the absolute determinant over $(0,\infty)$, which makes Z open in X. Then, the set $W = Y \cap Z$ is also open and non-empty. Therefore, W is a submanifold of X of the same dimension and the action of $\mathfrak{G}$ is well-defined and continuous on W. We can show this action is free.

We investigate the solutions of the equation $\boldsymbol{P}_{\mathfrak{g}}\boldsymbol{X}\boldsymbol{Q}_{\mathfrak{g}}^\top = \boldsymbol{X} \iff \boldsymbol{P}_{\mathfrak{g}}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{Q}_{\mathfrak{g}}$ for $\boldsymbol{X} \in W$. Since orthogonal transformations preserve norms and the rows of $\boldsymbol{X}$ have different norms, it follows that $\boldsymbol{P}_{\mathfrak{g}} = \boldsymbol{I}_n \Rightarrow \boldsymbol{X} = \boldsymbol{X}\boldsymbol{Q}_{\mathfrak{g}}$. We know that a subset of the rows of $\boldsymbol{X}$ span the whole of $\mathbb{R}^d$. Define the sub-matrix of $\boldsymbol{X}$ containing these rows by $\boldsymbol{X}^* \in \mathbb{R}^{d\times d}$. Then, we have $\boldsymbol{X}^*\boldsymbol{Q}_{\mathfrak{g}} = \boldsymbol{X}^* \Rightarrow \boldsymbol{Q}_{\mathfrak{g}} = (\boldsymbol{X}^*)^{-1}\boldsymbol{X}^* = \boldsymbol{I}_d$. This proves that the action is free and applying Theorem 20, yields the result.

For the trivial case when $n = 1$, notice that $\|\cdot\| : \mathbb{R}^{1\times d} \to \mathbb{R}$ is $\mathfrak{G}$ orbit-space injective. □

Proposition 23. *Any S_n-invariant injective function $f : \mathbb{R}^{n\times d} \to \mathbb{R}^q$ requires $q \geq nd$.*

Proof. Reusing the notation from above, notice that for all $n \geq 1$, S_n acts freely on the sub-manifold Y as shown above. Seeing S_n as a zero-dimensional Lie group and applying Theorem 20 yields the result. □