ON THE EXPRESSIVE POWER OF GEOMETRIC GRAPH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

The expressive power of Graph Neural Networks (GNNs) has been studied extensively through the lens of the Weisfeiler-Leman (WL) graph isomorphism test. Yet, many graphs arising in real-world applications come embedded in Euclidean space with an additional notion of geometric isomorphism, which is not covered by the WL framework. In this work, we propose a geometric version of the WL test (GWL) for discriminating geometric graphs while respecting the underlying physical symmetries: permutation, rotation, reflection, and translation. We use GWL to characterise the expressive power of GNNs that are invariant or equivariant to physical symmetries by studying the classes of geometric graphs that can or cannot be distinguished by these architectures. This allows us to formalise the advantages equivariant GNN layers have over their invariant counterparts in the Geometric Deep Learning blueprint. Finally, we connect our discrimination-based perspective with the universal approximation properties of geometric GNNs and prove they are two sides of the same coin.

1 INTRODUCTION

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 Graph Neural Networks (GNNs) (Xu et al., 2019; Morris et al., 2019). The WL framework has been a major driver of progress for more expressive GNNs (Chen et al., 2019; Maron et al., 2019; Morris et al., 2020; Bodnar et al., 2021b;a). However, WL does not directly apply to the increasingly relevant special case of *geometric graphs* – graphs embedded in Euclidean space – which come equipped with a stronger notion of isomorphism that also takes spatial symmetries into account. The lack of theoretical tools is becoming more apparent as geometric graphs are increasingly used to model systems in biochemistry (Jamasb et al., 2021), material science (Chanussot et al., 2021), physical simulations (Sanchez-Gonzalez et al., 2020), and multiagent robotics (Li et al., 2020). Graph Neural Networks (GNNs) with Euclidean symmetries 'baked in' have emerged as the architecture of choice for these domains (Geiger & Smidt, 2022).

GNNs 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. In addition to the permutation group, the geometric attributes of the graph (e.g. node coordinates) are acted upon by a Lie group such as the group of rotations ($SO(d)$) or rotations and reflections ($O(d)$). We use $\mathfrak{G}$ as a generic symbol for such a Lie group. Based on this, we consider two classes of GNNs for geometric graphs: (1) **$\mathfrak{G}$-equivariant models**, where the intermediate features are geometric quantities which are $\mathfrak{G}$-equivariant (Thomas et al., 2018; Anderson et al., 2019; Jing et al., 2020; Satorras et al., 2021; Brandstetter et al., 2022); and (2) **$\mathfrak{G}$-invariant models**, which only update scalar features in a $\mathfrak{G}$-invariant manner (Schütt et al., 2018; Xie & Grossman, 2018; Gasteiger et al., 2020). Invariant GNNs have shown strong performance for protein design (Zhang et al., 2022; Dauparas et al., 2022) and electrocatalysis (Gasteiger et al., 2021; Shi et al., 2022), while equivariant GNNs are being used within learnt interatomic potentials for molecular dynamics (Schütt et al., 2021; Batzner et al., 2022; Batatia et al., 2022b).

Despite promising empirical results, key theoretical questions remain unanswered: (1) How to characterise the *expressive power* of geometric GNNs? What effects do spatial symmetry constraints have on the expressive power of GNNs? And (2) what is the tradeoff between $\mathfrak{G}$-equivariant and

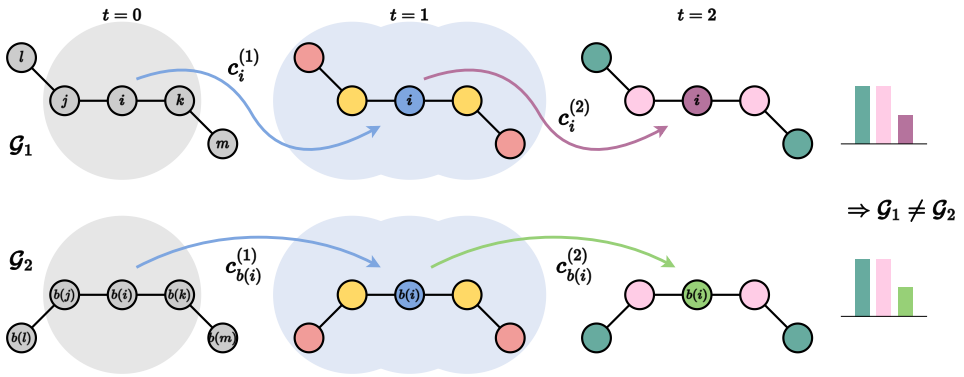

Figure 1: **Geometric Weisfeiler-Leman Test.** GWL distinguishes non-isomorphic geometric graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ by injectively assigning colours to distinct neighbourhood patterns, up to global symmetries (here $\mathfrak{G} = O(d)$). Each iteration expands the neighbourhood from which geometric information can be gathered (shaded for node i). Example inspired by Schütt et al. (2021).

$\mathfrak{G}$-invariant GNNs? Do we need computationally expensive $\mathfrak{G}$-equivariant models, or are more tractable $\mathfrak{G}$-invariant GNNs sufficiently powerful?

Contributions. In this work, we study the expressive power of geometric GNNs from the perspective of discriminating non-isomorphic geometric graphs while respecting the underlying spatial symmetries encoded by the Lie group $\mathfrak{G}$. To that end, we propose a geometric version of the Weisfeiler-Leman graph isomorphism test, termed GWL, in Section 3. The GWL test is an iterative node-colouring procedure that progressively refines the colour of each node by propagating $\mathfrak{G}$-invariant and $\mathfrak{G}$-equivariant information from its neighbours (Figure 1). In Section 4, we show how GWL can be used for studying existent architectures while removing implementation details that vary from one model to another. We use GWL to formally characterise classes of graphs that can and cannot be distinguished by $\mathfrak{G}$-invariant and $\mathfrak{G}$-equivariant GNNs. These results enable us to demonstrate the advantages of $\mathfrak{G}$-equivariant GNN layers over $\mathfrak{G}$-invariant layers. Additionally, in Section 5, we follow Chen et al. (2019) and prove an equivalence between a model's ability to discriminate geometric graphs and its ability to universally approximate $\mathfrak{G}$-invariant functions, both in a finite and a continuous feature setting.

2 Background

Graph Isomorphism and Weisfeiler-Leman. An attributed graph $\mathcal{G} = (\boldsymbol{A}, \boldsymbol{S})$ with a node set $\mathcal{V}$ of size n consists of an $n \times n$ adjacency matrix $\boldsymbol{A}$ and a matrix of scalar features $\boldsymbol{S} \in \mathbb{R}^{n \times f}$. Two attributed graphs $\mathcal{G}, \mathcal{H}$ are *isomorphic* if there exists an edge-preserving bijection $b : \mathcal{V}(\mathcal{G}) \to \mathcal{V}(\mathcal{H})$ such that $\boldsymbol{s}_i^{(\mathcal{G})} = \boldsymbol{s}_{b(i)}^{(\mathcal{H})}$, where the subscripts index rows and columns in the corresponding matrices.

The *Weisfeiler-Leman test* (WL) is an algorithm for testing whether two (attributed) graphs are isomorphic (Weisfeiler & Leman, 1968). At iteration zero the algorithm 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 node colouring by producing a new $\boldsymbol{c}_i^{(t)} \in C$:

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

where HASH is an injective map (i.e. a perfect hash map) that assigns a unique colour to each input and $\{\!\{\cdot\}\!\}$ denotes a multiset – a set that allows for repeated elements. 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, the WL test is inconclusive, and we say it 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 the group $\mathfrak{G}$ on a space X by $\mathfrak{g} \cdot x$. If $\mathfrak{G}$ acts on spaces X and Y, we say a function $f : X \to Y$ is $\mathfrak{G}$-*equivariant* if $f(\mathfrak{g} \cdot x) = \mathfrak{g} \cdot f(x)$. A function $f : X \to Y$ is $\mathfrak{G}$-*invariant* if $f(\mathfrak{g} \cdot x) = f(x)$. 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 $x_1, x_2 \in X$. Necessarily, 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 the translation group $T(d)$ is conventionally handled using relative positions. Given one of the standard 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 $\mathcal{G} = (\boldsymbol{A}, \boldsymbol{S}, \vec{\boldsymbol{V}}, \vec{\boldsymbol{X}})$ with a node set $\mathcal{V}$ 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[1] $\vec{\boldsymbol{V}} \in \mathbb{R}^{n \times d}$ (e.g. velocity, acceleration). 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* (denoted $\mathcal{G} \simeq \mathcal{H}$) 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 $\vec{\boldsymbol{t}} \in T(d)$:

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

Geometric graph isomorphism and distinguishing (sub-)graph geometries has important practical implications for representation learning. For *e.g.*, in molecular dynamics or protein design, an ideal architecture should map distinct local structural environments around atoms or amino acids to distinct embeddings in representation space (Pozdnyakov et al., 2020).

Geometric Graph Neural Networks. We consider two broad classes of geometric GNN architectures. $\mathfrak{G}$-equivariant GNN layers update scalar and vector features from iteration t to $t+1$ via learnable aggregate and update functions, ψ and ϕ, respectively:

$$\boldsymbol{s}_i^{(t+1)}, \vec{\boldsymbol{v}}_i^{(t+1)} := \phi\left((\boldsymbol{s}_i^{(t)}, \vec{\boldsymbol{v}}_i^{(t)}),\ \psi\left(\{\!\{(\boldsymbol{s}_i^{(t)}, \vec{\boldsymbol{v}}_i^{(t)}, \boldsymbol{s}_j^{(t)}, \vec{\boldsymbol{v}}_j^{(t)}, \vec{\boldsymbol{x}}_{ij}) \mid j \in \mathcal{N}_i\}\!\}\right)\right). \tag{3}$$

Alternatively, $\mathfrak{G}$-invariant layers only propagate scalar features and do not update vector features:

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

For both architectures, the scalar features $\{\boldsymbol{s}_i^{(T)}\}$ at the final iteration T are mapped to graph-level features via a permutation-invariant readout $f : \mathbb{R}^{n \times f} \to \mathbb{R}^{f'}$.

3 The Geometric Weisfeiler-Leman Test

Assumptions. Analogously 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 the WL test to geometric graphs, we note that WL uses a local, node-centric, procedure to update colours. At each iteration, the 1-hop *neighbourhood* $\mathcal{N}_i$ around each node i is extracted and the updated colour $c_i^{(t)}$ is computed based on the colour distribution in $\mathcal{N}_i$ (c.f. Eq. 1). In the geometric setting, the neighbourhood $\mathcal{N}_i$ is not just a tuple but an attributed point cloud around the central node i. As a result, each neighbourhood carries two types of information: (1) information on the neighbourhood type (invariant to $\mathfrak{G}$) and (2) information

[1] Without loss of generality, we work with a single vector feature per node. Our results generalise to multiple vector features or higher-order geometric tensors per node.

on the 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, the procedure must meet two properties:

Property 1: Orbit injectivity of colours. If a geometric graph is rotated in space, the colouring of the nodes should not change. At the same time, if two neighbourhoods are not the same up to an action of $\mathfrak{G}$ (e.g. a rotation), then the colours of the corresponding central nodes should differ. We thus need the colouring to 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 Equation 1 is that the performed 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 of the constituent nodes (Hinton et al., 2011). Therefore, we need to keep and aggregate auxiliary *geometric information* variables in a way that is $\mathfrak{G}$-equivariant and injective over the countable set of neighbourhoods in the dataset.

Geometric Weisfeiler-Leman (GWL). These intuitions motivate the following definition of the GWL test (Figure 1). At initialisation time, 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:

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

where HASH simply denotes an injective map over the scalar attributes $\boldsymbol{s}_i$ of node i, e.g. the HASH of standard WL. Note that HASH need not be continuous.

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:

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

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:

$$\mathfrak{g} \cdot \boldsymbol{g}_i^{(0)} := \left(c_i^{(0)},\ \boldsymbol{Q}_{\mathfrak{g}} \vec{\boldsymbol{v}}_i\right), \quad \mathfrak{g} \cdot \boldsymbol{g}_i^{(t)} := \left((c_i^{(t-1)}, \mathfrak{g} \cdot \boldsymbol{g}_i^{(t-1)}),\ \{\!\!\{(c_j^{(t-1)}, \mathfrak{g} \cdot \boldsymbol{g}_j^{(t-1)}, \boldsymbol{Q}_{\mathfrak{g}} \vec{\boldsymbol{x}}_{ij}) \mid j \in \mathcal{N}_i\}\!\!\}\right)$$

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:

$$c_i^{(t)} := \text{I-}\textsc{Hash}^{(t)}\left(\boldsymbol{g}_i^{(t)}\right), \tag{7}$$

by using a $\mathfrak{G}$-orbit injective and $\mathfrak{G}$-invariant function that we denote by I-HASH. That is for any geometric objects $\boldsymbol{g}, \boldsymbol{g}'$, $\text{I-}\textsc{Hash}(\boldsymbol{g}) = \text{I-}\textsc{Hash}(\boldsymbol{g}')$ if and only if there exists $\mathfrak{g} \in \mathfrak{G}$ such that $\boldsymbol{g} = \mathfrak{g} \cdot \boldsymbol{g}'$. Note that since we have by assumption a countable number of orbits, there exists an injective function Z sending each orbit (treated as an equivalence class) to a natural number and then the map sending $\boldsymbol{g} \mapsto Z(\mathcal{O}_{\mathfrak{G}}(\boldsymbol{g})) \in \mathbb{N}$ is orbit-space injective. This proves the existence of an I-HASH map.

Overview. With each iteration, $\boldsymbol{g}_i^{(t)}$ aggregates geometric information in progressively larger neighbourhoods around the node i. Thus, t iterations of GWL enable each node i to access its t-hop subgraph $\mathcal{N}_i^{(t)}$ via $\boldsymbol{g}_i^{(t)}$. The node colours summarise the structure of these t-hops via the $\mathfrak{G}$-invariant aggregation performed by I-HASH. The colouring 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 being geometrically non-isomorphic. Otherwise, we say the test cannot distinguish the two graphs.

Invariant GWL. Since we are interested in understanding the role of $\mathfrak{G}$-equivariance, we also consider a more restrictive version of GWL that only updates node colours using the $\mathfrak{G}$-orbit injective

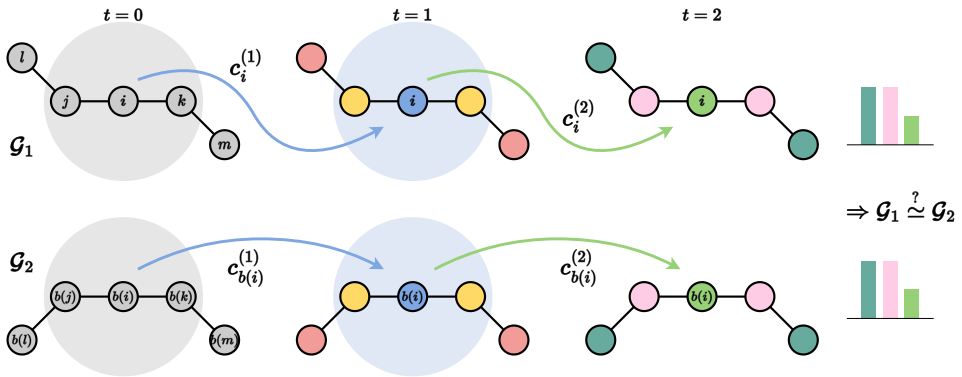

Figure 2: **Invariant GWL Test.** IGWL cannot distinguish $\mathcal{G}_1$ and $\mathcal{G}_2$ as they are 1-hop identical: The $\mathfrak{G}$-orbit of the 1-hop neighbourhood around each node is the same, and IGWL cannot propagate geometric orientation information beyond 1-hop (here $\mathfrak{G} = O(d)$).

I-HASH function and does not propagate the auxiliary geometric information (Figure 2). We term this the Invariant GWL (IGWL) test, which differs from GWL only in its update rule:

$$c_i^{(t)} := \text{I-HASH}\left((c_i^{(t-1)}, \vec{\boldsymbol{v}}_i)\ ,\ \{\!\{(c_j^{(t-1)}, \vec{\boldsymbol{v}}_j,\ \vec{\boldsymbol{x}}_{ij}) \mid j \in \mathcal{N}_i\}\!\}\right). \tag{8}$$

IGWL with k-body scalars. In order to further analyse the construction of the node colouring function I-HASH, we consider variations of IGWL 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)). We consider IGWL$_{(k)}$ given by the update rule:

$$c_i^{(t)} := \text{I-HASH}_{(k)}\left((c_i^{(t-1)}, \vec{\boldsymbol{v}}_i)\ ,\ \{\!\{(c_j^{(t-1)}, \vec{\boldsymbol{v}}_j,\ \vec{\boldsymbol{x}}_{ij}) \mid j \in \mathcal{N}_i\}\!\}\right), \tag{9}$$

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

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

where $\boldsymbol{j} = [j_1, \ldots, j_k]$ are all possible k-tuples formed of elements of $\mathcal{N}_i$. Therefore, 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, 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 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 (Bartók et al., 2013; Pozdnyakov et al., 2020). Therefore, I-HASH$_{(k)}$ with lower k makes the colouring weaker.

4 CHARACTERISING THE EXPRESSIVE POWER OF GWL

4.1 WHAT GEOMETRIC GRAPHS CAN GWL AND IGWL DISTINGUISH?

In order to formalise the expressive power of GWL and IGWL, let us consider what geometric graphs can and cannot be distinguished by the tests. Recall that each iteration of GWL aggregates geometric information $\boldsymbol{g}_i^{(t)}$ from progressively larger neighbourhoods $\mathcal{N}_i^{(t)}$ around the node i, and distinguishes (sub-)graphs via comparing $\mathfrak{G}$-orbit injective colouring of $\boldsymbol{g}_i^{(t)}$. As a simple first observation, we note that when all coordinates and vectors 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.

Next, let us consider 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 (bio)chemical modelling, where molecules occur in different conformations, but with the same graph topology given by the covalent bonding structure. 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 $\mathcal{N}_i^{(k)}$ and $\mathcal{N}_{b(i)}^{(k)}$ are identical up to group actions.

We can now formalise what geometric graphs can and cannot be distinguished by GWL.

Proposition 1. *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.*

An example illustrating this result is included in Figure 1.

Proposition 2. *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 3. *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 4. *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.*

See Figure 2 for an example.

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 $\mathcal{G}_2 = (\boldsymbol{A}_2, \boldsymbol{S}_2, \vec{\boldsymbol{V}}_2, \vec{\boldsymbol{X}}_2)$ to be non-isomorphic and constructed from point clouds using radial cutoffs.

Proposition 5. *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 $\mathcal{G}_2$.*

These results enable us to compare the expressive powers of GWL and IGWL.

Theorem 6. *GWL is strictly more powerful than IGWL.*

This statement formalises the advantage of $\mathfrak{G}$-equivariant intermediate layers for graphs and geometric data, 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 (colloquially called the "Picasso problem"). Similarly, our results explain how IGWL fails to understand how the various 1-hops of a graph are stitched together. Finally, we identify a setting where this distinction between the two approaches disappears.

Proposition 7. *IGWL has the same expressive power as GWL for fully connected geometric graphs.*

4.2 Characterising the Expressive Power of Geometric GNNs

We would like to characterise the maximum expressive power of geometric GNNs based on the GWL test. Firstly, we show that any message passing geometric GNN can be at most as powerful as GWL in distinguishing non-isomorphic geometric (sub-)graphs.

Theorem 8. *Any pair of geometric graphs distinguishable by a $\mathfrak{G}$-equivariant GNN is also distinguishable by GWL.*

Similarly, $\mathfrak{G}$-invariant GNNs of the form in Equation 4 can be at most as powerful as IGWL.

Theorem 9. *Any pair of geometric graphs distinguishable by a $\mathfrak{G}$-invariant GNN is also distinguishable by IGWL.*

With a sufficient number of 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 10. *$\mathfrak{G}$-equivariant GNNs have the same expressive power as GWL if the following conditions hold: (1) The aggregation ψ is an injective, $\mathfrak{G}$-equivariant multiset function. (2) The*

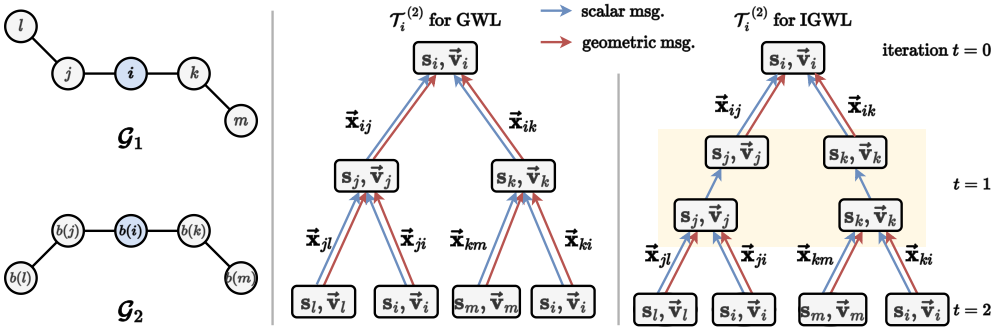

Figure 3: **Geometric Computation Trees for GWL and IGWL.** Unlike GWL, geometric orientation information cannot flow from the leaves to the root in IGWL, restricting its expressive power. IGWL cannot distinguish $\mathcal{G}_1$ and $\mathcal{G}_2$ as all 1-hop neighbourhoods are computationally identical.

scalar part of the update ϕ_s is a $\mathfrak{G}$-orbit injective, $\mathfrak{G}$-invariant multiset function. (3) The vector part of the update ϕ_v is an injective, $\mathfrak{G}$-equivariant multiset function. (4) The graph-level readout f is an injective multiset function.

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

Proposition 11. *$\mathfrak{G}$-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.*

Thus, we can directly transfer our results about GWL and IGWL to the class of GNNs bounded by the respective tests. This has several interesting practical implications.

Practical Implications. Firstly, Propositions 1 and 4 together suggest that $\mathfrak{G}$-equivariant GNNs should be preferred to $\mathfrak{G}$-invariant GNNs when message passing is restricted to local radial neighbourhoods around each node, as is conventional for large geometric graphs such as macromolecules with thousands of nodes. On the other hand, Proposition 7 suggests that $\mathfrak{G}$-equivariant and $\mathfrak{G}$-invariant GNNs can be made equally powerful when working on fully connected geometric graphs and all-to-all message passing. These results support the empirical success of recent $\mathfrak{G}$-invariant 'Graph Transformer' architectures (Joshi, 2020; Shi et al., 2022) for small molecules with tens of nodes, where working with full graphs is tractable. Additionally, Proposition 3 suggests that using wider cutoff radii when constructing geometric graphs can be effective for improving $\mathfrak{G}$-invariant GNNs, as larger neighbourhoods may be easier to distinguish via scalars.

Geometric Computation Trees. Stacking multiple $\mathfrak{G}$-equivariant layers enables the computation of compositional geometric features. Figure 3 illustrates this via computation trees (Garg et al., 2020). A computation tree $\mathcal{T}_i^{(t)}$ represents the maximum information contained in GWL/IGWL colours or GNN features 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. Critically, geometric orientation information cannot flow from one level to another in IGWL as it only updates scalar information, restricting its expressive power.

4.3 Understanding the IGWL$_{(k)}$ Hierarchy

Let us analyse the construction of the I-Hash function via the k-body variations IGWL$_{(k)}$. In doing so, we will make connections between IGWL and WL for non-geometric graphs. Firstly, we show that IGWL$_{(2)}$ is equivalent to WL when all the pairwise distances between the nodes are the same. A similar observation was recently made by Pozdnyakov & Ceriotti (2022).

Proposition 12. *Let $\mathcal{G} = (\boldsymbol{A}, \boldsymbol{S}, \vec{\boldsymbol{X}})$ and $\mathcal{H} = (\boldsymbol{A}', \boldsymbol{S}', \vec{\boldsymbol{X}}')$ be two geometric graphs with the property that all edges have equal length. Then, IGWL$_{(2)}$ distinguishes the two graphs if and only if WL can distinguish the attributed graphs $(\boldsymbol{A}, \boldsymbol{S})$ and $(\boldsymbol{A}', \boldsymbol{S}')$.*

This equivalence also points out the limitations of models like SchNet (Schütt et al., 2018), which only rely on pairwise distances and consequently suffer from all well-known failure cases of WL. For instance, such models cannot distinguish two equilateral triangles from the regular hexagon (Gasteiger et al., 2020). Additionally, we can formalise the relationship between the maximum cardinality of local neighbourhoods and the injectivity of I-HASH$_{(k)}$.

Proposition 13. 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.*

While building provably injective I-HASH$_{(k)}$ functions may require intractably high k, the hierarchy of 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 Tensor Field Networks (Thomas et al., 2018) use distances; DimeNet (Gasteiger et al., 2020) uses distances and angles; and MACE (Batatia et al., 2022b) constructs scalars up to $k \leq 4$. We can state the following about the hierarchy of IGWL$_{(k)}$ tests.

Proposition 14. *IGWL$_{(k)}$ is at least as powerful as IGWL$_{(k-1)}$. For $k \leq 5$, IGWL$_{(k)}$ is strictly more powerful than IGWL$_{(k-1)}$.*

5 Discrimination and Universality

Overview. First, we build upon the work of Chen et al. (2019), which proved an equivalence between graph discrimination and universal approximation. With minor adaptations, we generalise these results to any isomorphism induced by a compact Lie group $\mathfrak{G}$. This provides a connection between the the universal approximation capabilities of various $\mathfrak{G}$-invariant and equivariant models (Gasteiger et al., 2021; Dym & Maron, 2020; Villar et al., 2021) and the discrimination-based perspective introduced by GWL. In the second part, we study the number of invariant aggregators that are required in a continuous setting to distinguish any two neighbourhoods.

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 $n \times n$ geometric graphs and $Y = \mathcal{X}^{n\times n}$, where $\mathcal{X} \subseteq \mathbb{R}^d \times \mathbb{R}^f$.

Definition 15. *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 16. *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, which is also the setting where the GWL test operates. Therefore, we will assume that Y is a countable subset of X.

Theorem 17. *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 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 18. *If $\mathcal{C}$ is pairwise $Y_{\mathfrak{G}}$ discriminating, then $\mathcal{C}^{+2}$ is universally approximating over Y.*

Continous 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 19. *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 20. *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 *continous* orbit-space injective functions on a manifold X. 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 21. *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 22. *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 23. *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 $S_n \times \{e_{\text{SO(d)}}\} \simeq S_n$ we obtain a statement holding for all n and generalising a result from PNA 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 24. *Any S_n-invariant function $f : \mathbb{R}^{n\times d} \to \mathbb{R}^q$ requires $q \geq nd$.*

6 Discussion and Related Work

This work proposes a geometric version of the Weisfeiler-Leman graph isomorphism test (GWL) for discriminating geometric graphs while respecting the underlying spatial symmetries. We use GWL to characterise the expressive power of geometric GNNs and connect the universal approximation properties of these models to discriminating geometric graphs.

Related Work. 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). Recent work on geometric GNNs and their universality has shown that Tensor Field Networks (Dym & Maron, 2020), GemNet (Gasteiger et al., 2021), and GVP (Jing et al., 2020) can be universal approximators of continuous, $\mathfrak{G}$-equivariant or $\mathfrak{G}$-invariant multiset function over point clouds. In contrast, the GWL framework takes a discriminatory angle and additionally considers the graph structure and how its sparsity affects the propagation of geometric information via generic message passing procedures. Overall, our work formalises what classes of geometric graphs can and cannot be distinguished by message passing $\mathfrak{G}$-invariant/equivariant GNNs while abstracting away implementation details.

Future Work. GWL provides an abstraction to study the limits of geometric GNNs, but in practice it is challenging to build maximally powerful GNNs that satisfy the conditions of Proposition 10 as GWL relies on $\mathfrak{G}$-orbit injective colouring and $\mathfrak{G}$-equivariant propagation of auxiliary geometric information. Based on the intuitions gained from GWL, future work will explore building provably powerful, *practical* geometric GNNs for applications in biochemistry, material science, and multiagent robotics, and better characterise the trade-offs related to practical implementation choices.

REFERENCES

Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. *Advances in neural information processing systems*, 32, 2019.

Albert P Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Physical Review B*, 87(18):184115, 2013.

Ilyes Batatia, Simon Batzner, Dávid Péter Kovács, Albert Musaelian, Gregor NC Simm, Ralf Drautz, Christoph Ortner, Boris Kozinsky, and Gábor Csányi. The design space of e (3)-equivariant atom-centered interatomic potentials. *arXiv preprint arXiv:2205.06643*, 2022a.

Ilyes Batatia, Dávid Péter Kovács, Gregor NC Simm, Christoph Ortner, and Gábor Csányi. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. *arXiv preprint arXiv:2206.07697*, 2022b.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1): 1–11, 2022.

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Lio, Guido F Montufar, and Michael Bronstein. Weisfeiler and lehman go cellular: Cw networks. *Advances in Neural Information Processing Systems*, 34:2625–2640, 2021a.

Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido F Montufar, Pietro Lio, and Michael Bronstein. Weisfeiler and lehman go topological: Message passing simplicial networks. In *International Conference on Machine Learning*, pp. 1026–1037. PMLR, 2021b.

Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling. Geometric and physical quantities improve e(3) equivariant message passing. In *International Conference on Learning Representations*, 2022.

Glen E Bredon. *Introduction to compact transformation groups*. Academic press, 1972.

Glen E Bredon. *Topology and geometry*, volume 139. Springer Science & Business Media, 2013.

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

Lowik Chanussot, Abhishek Das, Siddharth Goyal, Thibaut Lavril, Muhammed Shuaibi, Morgane Riviere, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, Aini Palizhati, Anuroop Sriram, Brandon Wood, Junwoong Yoon, Devi Parikh, C. Lawrence Zitnick, and Zachary Ulissi. Open catalyst 2020 (oc20) dataset and community challenges. *ACS Catalysis*, 2021. doi: 10.1021/acscatal.0c04525.

Zhengdao Chen, Soledad Villar, Lei Chen, and Joan Bruna. On the equivalence between graph isomorphism testing and function approximation with gnns. *Advances in neural information processing systems*, 32, 2019.

Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. Principal neighbourhood aggregation for graph nets. *Advances in Neural Information Processing Systems*, 33:13260–13271, 2020.

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning based protein sequence design using proteinmpnn. *bioRxiv*, 2022.

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

Genevieve Dusson, Markus Bachmayr, Gabor Csanyi, Ralf Drautz, Simon Etter, Cas van der Oord, and Christoph Ortner. Atomic cluster expansion: Completeness, efficiency and stability. *arXiv preprint arXiv:1911.03550*, 2019.

Nadav Dym and Haggai Maron. On the universality of rotation equivariant point cloud networks. In *International Conference on Learning Representations*, 2020.

Vikas Garg, Stefanie Jegelka, and Tommi Jaakkola. Generalization and representational limits of graph neural networks. In *International Conference on Machine Learning*, pp. 3419–3430. PMLR, 2020.

Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*, 2020.

Johannes Gasteiger, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. In *Advances in Neural Information Processing Systems*, 2021.

Mario Geiger and Tess Smidt. e3nn: Euclidean neural networks. *arXiv preprint arXiv:2207.09453*, 2022.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.

Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International conference on artificial neural networks*, 2011.

Arian R Jamasb, Ramon Viñas, Eric J Ma, Charlie Harris, Kexin Huang, Dominic Hall, Pietro Lió, and Tom L Blundell. Graphein-a python library for geometric deep learning and network analysis on protein structures and interaction networks. 2021.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2020.

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

Qingbiao Li, Fernando Gama, Alejandro Ribeiro, and Amanda Prorok. Graph neural networks for decentralized multi-robot path planning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11785–11792. IEEE, 2020.

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. *Advances in neural information processing systems*, 32, 2019.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, 2019.

Christopher Morris, Gaurav Rattan, and Petra Mutzel. Weisfeiler and leman go sparse: Towards scalable higher-order graph embeddings. *Advances in Neural Information Processing Systems*, 33:21824–21840, 2020.

Sergey N Pozdnyakov and Michele Ceriotti. Incompleteness of graph convolutional neural networks for points clouds in three dimensions. *arXiv preprint arXiv:2201.07136*, 2022.

Sergey N Pozdnyakov, Michael J Willatt, Albert P Bartók, Christoph Ortner, Gábor Csányi, and Michele Ceriotti. Incompleteness of atomic structure representations. *Physical Review Letters*, 125(16):166001, 2020.

Ronald C Read and Derek G Corneil. The graph isomorphism disease. *Journal of graph theory*, 1 (4):339–363, 1977.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning*, pp. 8459–8468. PMLR, 2020.

Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International Conference on Machine Learning*, pp. 9323–9332. PMLR, 2021.

Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International Conference on Machine Learning*, pp. 9377–9388. PMLR, 2021.

Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018.

Alexander V Shapeev. Moment tensor potentials: A class of systematically improvable interatomic potentials. *Multiscale Modeling & Simulation*, 14(3):1153–1173, 2016.

Yu Shi, Shuxin Zheng, Guolin Ke, Yifei Shen, Jiacheng You, Jiyan He, Shengjie Luo, Chang Liu, Di He, and Tie-Yan Liu. Benchmarking graphormer on large-scale molecular modeling datasets. *arXiv preprint arXiv:2203.04810*, 2022.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Soledad Villar, David W Hogg, Kate Storey-Fisher, Weichi Yao, and Ben Blum-Smith. Scalars are universal: Equivariant machine learning, structured like classical physics. *Advances in Neural Information Processing Systems*, 34:28848–28863, 2021.

Boris Weisfeiler and Andrei Leman. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series*, 2(9):12–16, 1968.

Tian Xie and Jeffrey C. Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys. Rev. Lett.*, 120:145301, Apr 2018. doi: 10.1103/PhysRevLett.120.145301.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Anthony Zee. *Group theory in a nutshell for physicists*, volume 17. Princeton University Press, 2016.

Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. *arXiv preprint arXiv:2203.06125*, 2022.

A PROOFS FOR WHAT GWL AND IGWL CAN DISTINGUISH

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 1. *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 1. 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 2. *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 2. 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 3. *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 3. 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 4. *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 4. 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 5. *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 5. 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 $\mathcal{G}_2$. □

Theorem 6. *GWL is strictly more powerful than IGWL.*

Proof of Theorem 6. 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.* $\boldsymbol{g}_i^{(t)} = \boldsymbol{g}_i^{(t-1)} = \boldsymbol{g}_i^{(0)} = ((\boldsymbol{s}_i, \vec{\boldsymbol{v}}_i),\ \emptyset)$. 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 1 states that GWL can distinguish any such graphs, while Proposition 4 states that IGWL cannot distinguish any such graphs. An example is the pair of graphs in Figures 1 and 2. □

Proposition 7. *IGWL has the same expressive power as GWL for fully connected geometric graphs.*

Proof of Proposition 7. 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 1 and Proposition 4, $\mathcal{G}_1$ 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. □

B PROOFS FOR EQUIVALENCE BETWEEN GWL AND GEOMETRIC GNNS

Our proofs adapt the techniques used in (Xu et al., 2019) for connecting 1-WL with GNNs. Note that we omit including the relative position vectors $\vec{\boldsymbol{x}}_{ij} = \vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_j$ in GWL and geometric GNN updates for brevity, as relative positions vectors can be merged into the vector features.

Theorem 8. *Any pair of geometric graphs distinguishable by a $\mathfrak{G}$-equivariant GNN is also distinguishable by GWL.*

Proof of Theorem 8. 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:

$$\left\{(c_i^{(t)}, \boldsymbol{g}_i^{(t)}),\ \{\!\{(c_j^{(t)}, \boldsymbol{g}_j^{(t)}) \mid j \in \mathcal{N}_i\}\!\}\right\} = \left\{(c_k^{(t)}, \mathfrak{g} \cdot \boldsymbol{g}_k^{(t)}),\ \{\!\{(c_j^{(t)}, \mathfrak{g} \cdot \boldsymbol{g}_j^{(t)}) \mid j \in \mathcal{N}_k\}\!\}\right\} \tag{10}$$

By our assumption on iteration t,

$$\left\{(\boldsymbol{s}_i^{(t)}, \vec{\boldsymbol{v}}_i^{(t)}),\ \{\!\{(\boldsymbol{s}_j^{(t)}, \vec{\boldsymbol{v}}_j^{(t)}) \mid j \in \mathcal{N}_i\}\!\}\right\} = \left\{(\boldsymbol{s}_k^{(t)}, \boldsymbol{Q}_{\mathfrak{g}} \vec{\boldsymbol{v}}_k^{(t)}),\ \{\!\{(\boldsymbol{s}_j^{(t)}, \boldsymbol{Q}_{\mathfrak{g}} \vec{\boldsymbol{v}}_j^{(t)}) \mid j \in \mathcal{N}_k\}\!\}\right\} \tag{11}$$

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. □

Theorem 9. *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 8. □

Proposition 10. *$\mathfrak{G}$-equivariant GNNs have the same expressive power as GWL if the following conditions hold: (1) The aggregation ψ is an injective, $\mathfrak{G}$-equivariant multiset function. (2) The scalar part of the update ϕ_s is a $\mathfrak{G}$-orbit injective, $\mathfrak{G}$-invariant multiset function. (3) The vector part of the update ϕ_v is an injective, $\mathfrak{G}$-equivariant multiset function. (4) The graph-level readout f is an injective multiset function.*

Proof of Theorem 10. 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, followed by an injective map. The result is $\mathfrak{G}$-orbit injective.

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

$$\boldsymbol{s}_i^{(t)} = \phi_s \left(\left[(\boldsymbol{s}_i^{(t-1)}, \vec{\boldsymbol{v}}_i^{(t-1)}) \ , \ \psi \left(\{\!\{ (\boldsymbol{s}_i^{(t-1)}, \boldsymbol{s}_j^{(t-1)}, \vec{\boldsymbol{v}}_i^{(t-1)}, \vec{\boldsymbol{v}}_j^{(t-1)}) \mid j \in \mathcal{N}_i \}\!\} \right) \right] \right) \tag{12}$$

$$\vec{\boldsymbol{v}}_i^{(t)} = \phi_v \left((\boldsymbol{s}_i^{(t-1)}, \vec{\boldsymbol{v}}_i^{(t-1)}) \ , \ \psi \left(\{\!\{ (\boldsymbol{s}_i^{(t-1)}, \boldsymbol{s}_j^{(t-1)}, \vec{\boldsymbol{v}}_i^{(t-1)}, \vec{\boldsymbol{v}}_j^{(t-1)}) \mid j \in \mathcal{N}_i \}\!\} \right) \right) \tag{13}$$

with the aggregation function ψ being $\mathfrak{G}$-equivariant and injective, the scalar update function ϕ_s being $\mathfrak{G}$-invariant and injective, and the vector update function ϕ_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:

$$c_i^{(t)} = h_s \left(\left[(c_i^{(t-1)}, \boldsymbol{g}_i^{(t-1)}) \ , \ \{\!\{ (c_j^{(t-1)}, \boldsymbol{g}_j^{(t-1)}) \mid j \in \mathcal{N}_i \}\!\} \right] \right), \tag{14}$$

$$\boldsymbol{g}_i^{(t)} = h_v \left((c_i^{(t-1)}, \boldsymbol{g}_i^{(t-1)}) \ , \ \{\!\{ (c_j^{(t-1)}, \boldsymbol{g}_j^{(t-1)}) \mid j \in \mathcal{N}_i \}\!\} \right), \tag{15}$$

where h_s is $\mathfrak{G}$-invariant and injective, and h_v is $\mathfrak{G}$-equivariant and injective.

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

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

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

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

$$\boldsymbol{s}_i^{(t+1)} = g_s \left(\left[(c_i^{(t)}, \boldsymbol{g}_i^{(t)}) \ , \ \{\!\{ (c_j^{(t)}, \boldsymbol{g}_j^{(t)}) \mid j \in \mathcal{N}_i \}\!\} \right] \right), \tag{16}$$

$$\vec{\boldsymbol{v}}_i^{(t+1)} = g_v \left((c_i^{(t)}, \boldsymbol{g}_i^{(t)}) \ , \ \{\!\{ (c_j^{(t)}, \boldsymbol{g}_j^{(t)}) \mid j \in \mathcal{N}_i \}\!\} \right), \tag{17}$$

We can then consider:

$$\boldsymbol{s}_i^{(t+1)} = g_s \circ h_s^{-1} \ h_s \left(\left[(c_i^{(t)}, \boldsymbol{g}_i^{(t)}) \ , \ \{\!\{ (c_j^{(t)}, \boldsymbol{g}_j^{(t)}) \mid j \in \mathcal{N}_i \}\!\} \right] \right), \tag{18}$$

$$\vec{\boldsymbol{v}}_i^{(t+1)} = g_v \circ h_v^{-1} \ h_v \left((c_i^{(t)}, \boldsymbol{g}_i^{(t)}) \ , \ \{\!\{ (c_j^{(t)}, \boldsymbol{g}_j^{(t)}) \mid j \in \mathcal{N}_i \}\!\} \right), \tag{19}$$

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.

□

C PROOFS FOR THE IGWL$_{(k)}$ HEIRARCHY

Proposition 12. *Let $\mathcal{G} = (\boldsymbol{A}, \boldsymbol{S}, \vec{\boldsymbol{X}})$ and $\mathcal{H} = (\boldsymbol{A}', \boldsymbol{S}', \vec{\boldsymbol{X}}')$ be two geometric graphs with the property that all edges have equal length. Then, IGWL$_{(2)}$ distinguishes the two graphs if and only if WL can distinguish the attributed graphs $(\boldsymbol{A}, \boldsymbol{S})$ and $(\boldsymbol{A}', \boldsymbol{S}')$.*

Proof. Let c and k the colours produced by 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 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 $k_i^{(t+1)} = k_j^{(t+1)}$ □

Proposition 13. 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 13. 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 14. *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 14. 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$_{(k-1)}$: (1) For $k = 3$ and $\mathfrak{G} = O(3)/SO(3)$, 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$_{(2)}$. (2) For $k = 4$ and $\mathfrak{G} = O(3)/SO(3)$, the pair of local neighbourhoods from Figure 1 in Pozdnyakov et al. (2020) can be distinguished by IGWL$_{(4)}$ but not by IGWL$_{(3)}$. (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 IGWL$_{(5)}$ but not by IGWL$_{(4)}$. (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 IGWL$_{(5)}$ but not by IGWL$_{(4)}$. □

D UNIVERSALITY AND DISCRIMINATION PROOFS

D.1 EQUIVALENCE BETWEEN UNIVERSALITY AND DISCRIMINATION

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

Theorem 17. *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 25. *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 $\overline{\delta}_{y,y'} \in \mathcal{C}$:

$$\overline{\delta}_{y,y'}(x) = |\delta_{y,y'}(x) - \delta_{y,y'}(y)| \to \begin{cases} = 0 & \text{if } x \simeq y \\ > 0 & \text{if } x \simeq y' \\ \geq 0 & \text{otherwise} \end{cases} \tag{20}$$

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 $\overline{\delta}_{y,y'}$:

$$\delta_y(x) = \sum_{y' \in Y, y' \not\simeq y} \overline{\delta}_{y,y'}(x) \to \begin{cases} = 0 & \text{if } x \simeq y \\ > 0 & \text{if } x \not\simeq y \text{ and } x \in \mathcal{O}_{\mathfrak{G}}(Y) \supseteq Y \\ \geq 0 & \text{otherwise} \end{cases} \tag{21}$$

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 26. *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:

$$h(x) = \sum_{y \in Y} k_y f(y) b_{r_y}(\delta_y(x)) \tag{22}$$

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

Theorem 18. *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 27. *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 19. *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 27 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 28. *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 29. *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 30. *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 28, 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 $\text{supp}(\phi_{y_0}) = \delta_{y_0}^{-1}([0, r_{y_0}))$ 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:

$$f(y') = f(y') \sum_{y_0 \in Y_0} \psi_{y_0}(y') = \sum_{y_0 \in Y_0 \,:\, y' \in \delta_{y_0}^{-1}([0, r_{y_0}))} f(y') \psi_{y_0}(y') \tag{23}$$

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:

$$\left| f(y') - \sum_{y_0 \in Y_0} f(y_0) \psi_{y_0}(y') \right| = \left| f(y') - \sum_{y_0 \in Y_0 \,:\, y' \in \delta_{y_0}^{-1}([0, r_{y_0}))} f(y_0) \psi_{y_0}(y') \right| \tag{24}$$

$$= \sum_{y_0 \in Y_0 \,:\, y' \in \delta_{y_0}^{-1}([0, r_{y_0}))} |f(y') - f(y_0)| \, \psi_{y_0}(y') < \varepsilon \tag{25}$$

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 20. *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. □

D.2 Number of aggregators in continous setting

Theorem 21. *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, $Y/\mathfrak{G}$ 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 23. *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 21, 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. □

Theorem 22. *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 21 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. □

Proposition 24. *Any S_n-invariant 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 21 yields the result. □

E ALTERNATIVE GWL FORMULATION FOR $SO(2)$

The version of GWL presented in the main text makes use of a geometric neighbourhood object $\boldsymbol{g}$ that keeps increasing in size at each iteration. A natural question to ask is whether a more compact encoding of the orientation of a neighbourhood can be achieved. In this section, we show this is possible for $SO(2)$, which has some particularly convenient properties: it is commutative and its (discrete) subgroups are particularly simple. First, we point out why when working with the standard representation of $SO(d)$ acting on vectors in $\mathbb{R}^d$, the orientation of a neighbourhood cannot be (injectively) encoded into another vector obtained from permutation invariant aggregation.

Definition 31 (Representation)**.** *A representation of a group $\mathfrak{G}$ on a vector space V over is a group homomorphism from $\mathfrak{G}$ to $GL(V)$, the general linear group on V. That is, a representation is a map $\rho : \mathfrak{G} \to GL(V)$, such that $\rho(\mathfrak{g}_1\mathfrak{g}_2) = \rho(\mathfrak{g}_1)\rho(\mathfrak{g}_2)$.*

Denote by ρ^0 the usual representation of $O(d)$ on $\mathbb{R}^d$. Let $\mathcal{X} \subset \mathbb{R}^d$ be a countable feature space, $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\} \subset \mathcal{X}$ a multi-set of features and associated representations $\rho_1, \dots, \rho_n : \mathfrak{G} \to SO(d)$. Furthermore assume that $\mathfrak{G}$ acts on $\mathbb{R}^{n\times d}$ via $g\boldsymbol{X} := [\rho^1(\boldsymbol{x}_1), \dots, \rho^n(\boldsymbol{x}_n)]$. We want to find a multi-set function $f : \mathcal{X}^n \to \mathbb{R}^d$ that is $O(d)$-equivariant and injective over the orbit of X, denoted by $\mathcal{O}_{\mathfrak{G}}(X)$. A first observation is that if $O(d)$ acts (faithfully) via ρ^0 on $\mathbb{R}^d$, then there is no such function due to the presence of discrete rotational symmetries in certain sets of input vectors (i.e. snowflake like structures).

Proposition 32 (The Discrete Symmetry Problem)**.** *Let $\boldsymbol{X} \in \mathbb{R}^{n\times 2}$ be a point cloud with n points having (discrete) rotational symmetry, i.e. there exists $\pi \in S_n$ and $\mathfrak{g} \neq \mathrm{id} \in O(2)$ s.t. $\pi\boldsymbol{X} = \mathfrak{g}\boldsymbol{X}$. Then, any permutation-invariant and $O(d)$-equivariant function $f : \mathbb{R}^{n\times 2} \to \mathbb{R}^2$ has the property that $f(\boldsymbol{X}) = f(\mathfrak{g}\boldsymbol{X}) = 0$ for all $\mathfrak{g}$.*

Proof. $\pi\boldsymbol{X} = \mathfrak{g}\boldsymbol{X} \implies f(\pi\boldsymbol{X}) = f(\mathfrak{g}\boldsymbol{X}) \implies f(\boldsymbol{X}) = \rho^0(\mathfrak{g})f(\boldsymbol{X}) \implies f(\boldsymbol{X}) = 0$ □

This result easily extends to any dimension d by finding such a set of vectors that span a two dimensional plane and performing a rotation in that plane. The proposition above shows that we need to find an alternative representation ρ. We show that this is possible for $\mathfrak{G} = SO(2)$ and for the rest of this section assume $d = 2$.

Theorem 33. *There exists a representation $\rho : \mathfrak{G} \to SO(\mathbb{R}^2)$ and a multi-set function $\textsc{Hash}_v$ sending X to $\mathbb{R}^2 \setminus \{0\}$ that is unique for each multi-set $X \subset \mathcal{X}$ of bounded size and $\textsc{Hash}_v(\mathfrak{g}X) = \rho(\mathfrak{g})\textsc{Hash}_v(X)$. Furthermore, for any two $X_1, X_2 \subset \mathcal{X}$ we have that $\|\textsc{Hash}_v(X_1)\| = \|\textsc{Hash}_v(X_2)\|$ iff there is $\mathfrak{g} \in \mathfrak{G}$ such that $gX_1 = X_2$.*

Proof. Let $\mathcal{O}_{\mathfrak{G}}(X)$ be an orbit of X generated by the action of the group $\mathfrak{G} = SO(2)$ acting on $\mathbb{R}^{n\times d}/S_n$ (i.e. the space of multi-sets of 2D vectors) as in the statement of Theorem 33. Then, $SO(2)$ acts transitively on $\mathcal{O}_{\mathfrak{G}}(X)$ by construction.

$$\begin{tikzcd}
\mathcal{O}_{\mathfrak{G}}(X) \arrow[r, "\phi_X"] & SO(2)/\mathfrak{G}(X) \arrow[r, "\eta"] & SO(\mathbb{R}^2) \arrow[r, "\omega_{\boldsymbol{u}}"] & S^1 \subset \mathbb{R}^2 \\
& SO(2) \arrow[u, "\pi"] \arrow[ur, "\rho"] \arrow[urr, "\sigma_{\boldsymbol{u}}"] & &
\end{tikzcd}$$

Denoting by $\mathfrak{G}(X)$ its isotropy/stabilizer group of X, the function $\phi_X : \mathcal{O}(X) \to SO(2)/\mathfrak{G}(X)$ given by $\phi_X(Y) = \phi_X(\mathfrak{g}X) = \mathfrak{G}(X)$ with $g \in SO(2)$ is well-defined and, moreover, it is an equivariant homeomorphism of $\mathfrak{G}$-spaces (e.g. see Proposition 4.1 in Bredon (1972)).

Notice that because $SO(2)$ is abelian, $\mathfrak{G}(X)$ is a normal subgroup and $SO(2)/\mathfrak{G}(X)$ is not only a manifold, but also a (Lie) group. We will show that we can find a faithful orthogonal representation of this quotient group. Because we are working with finite neighbourhoods, the group $\mathfrak{G}(X)$ is a finite cyclic subgroup (standard result) if $X \neq 0$. Otherwise, $\mathfrak{G}(X) = SO(2)$.

Therefore, let us denote by $\alpha(\mathfrak{g}\mathfrak{G}(X))$ the smallest angle of a rotation in the equivalence class $\mathfrak{g}\mathfrak{G}(X)$. If $\mathfrak{G}(X)$ is a finite cyclic subgroup generated by rotations of an angle θ, then $\alpha(SO(2)/\mathfrak{G}(X)) = [0, \theta)$. If $\mathfrak{G}(X) = SO(2)$, then $\alpha(SO(2)/\mathfrak{G}(X)) = 0$. Thus, we can

construct the orthogonal representation $\eta : SO(2)/\mathfrak{G}(X) \to SO(\mathbb{R}^d)$ given by $\eta(\mathfrak{g}\mathfrak{G}(X)) = \boldsymbol{R}_{\alpha(SO(2)/\mathfrak{G}(X))2\pi/\theta}$ if $\mathfrak{G}(X)$ is finite and $\eta(\mathfrak{g}\mathfrak{G}(X)) = \boldsymbol{I}$, otherwise. It can be checked that in either case, η is an injective group homomorphism and, therefore, a faithful representation of the group.

Given $\boldsymbol{u} = [1, 0] \in S^1 \subset R^2$, we can define a map $\omega_{\boldsymbol{u}} : SO(\mathbb{R}^d) \to S^1$ given by $\omega_{\boldsymbol{u}}(\boldsymbol{R}_\alpha) = \boldsymbol{R}_\alpha \boldsymbol{u}$, which is another homeomorphism. Thus, the composition is $\psi = \omega_{\boldsymbol{u}} \circ \eta \circ \phi_X : \mathcal{O}_{\mathfrak{G}}(X) \to \mathbb{R}^2$ is an injective function. Moreover, because the quotient map $\pi : SO(2) \to SO(2)/\mathfrak{G}(X)$ is a group homomorphism, we can compose it with η to lift the representation of $SO(2)/\mathfrak{G}(X)$ to a representation $\rho = \eta \circ \pi$ of $SO(2)$. Therefore, it follows that ψ is also equivariant with respect to this representation of $SO(2)$.

Finally, we can extend ψ to a function that is injective for any multi-set by constructing a $\psi_{\mathcal{O}_{\mathfrak{G}}(X)}$ for each orbit $\mathcal{O}_{\mathfrak{G}}(X)$, where X is a representative of the orbit, by using a map $\omega_{\mathcal{O}_{\mathfrak{G}}(X)}$ where $\boldsymbol{u}$ = $[\text{I-HASH}(X), 0]$ (i.e. the norm encodes the orbit). □

Alternative $SO(2)$ Geometric Weisfeiler-Leman Test. Given a geometric graph $(\mathcal{G}, \boldsymbol{X}, \boldsymbol{H})$, the $SO(2)$-GWL algorithm consists of three steps:

1. For each node $v \in \mathcal{G}$ and $u \in \mathcal{N}(v)$, initialise:

$$\boldsymbol{m}^0_{v,u} := \text{HASH}(\boldsymbol{h}_v, \boldsymbol{h}_u, \|\text{HASH}_v(\boldsymbol{x}_v - \boldsymbol{x}_u)\|) \frac{\text{HASH}_v(\boldsymbol{x}_v - \boldsymbol{x}_u)}{\|\text{HASH}_v(\boldsymbol{x}_v - \boldsymbol{x}_u)\|}$$
$$\boldsymbol{m}^0_v := \text{HASH}_v(\boldsymbol{m}^0_{v,u_1}, \ldots, \boldsymbol{m}^0_{v,u_{d(v)}})$$

2. For each node $v \in \mathcal{G}$, update $\boldsymbol{m}^{l+1}_v := \text{HASH}_v(\boldsymbol{m}^l_{u_1}, \ldots, \boldsymbol{m}^l_{u_{d(v)}})$.
3. Go back to step 2 until the partition induced by $\{\!\{\|\boldsymbol{m}^l_v\| : v \in \mathcal{G}\}\!\}$ becomes stable.
4. Return the colours of the nodes $\{\!\{\|\boldsymbol{m}^L_v\| : v \in \mathcal{G}\}\!\}$ at the last iteration L.