On the Expressive Power of Geometric Graph Neural Networks

Anonymous Author(s)
Anonymous Affiliation
Anonymous Email

Abstract

We propose a geometric version of the Weisfeiler-Leman graph isomorphism test (GWL) for discriminating geometric graphs while respecting the underlying physical symmetries: permutations, rotation, reflection, and translation. We use GWL to characterise the expressive power of Graph Neural Networks (GNNs) that are invariant or equivariant to physical symmetries in terms of the classes of geometric graphs they can distinguish. This allows us to formalise the advantages of equivariant GNNs over invariant GNNs: equivariant layers have greater expressive power as they enable propagating geometric information beyond local neighbourhoods, while invariant layers only reason locally via scalars and cannot discriminate geometric graphs with different non-local properties.

1 Introduction

The graph isomorphism problem and the Weisfeiler-Leman (WL) [1] test for distinguishing non-isomorphic graphs have become powerful tools for analysing the expressive power of Graph Neural Networks (GNNs) [2, 3]. The WL framework has been a major driver of progress for more expressive GNNs [4–8]. 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 [9], material science [10], physical simulations [11], and multiagent robotics [12]. Graph Neural Networks (GNNs) with Euclidean symmetries ‘baked in’ have emerged as the architecture of choice for these domains [13].

Geometric GNNs follow the message passing paradigm [14] 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 nodes (e.g., coordinates, velocity) transform along with Euclidean transformations of the system, i.e. they are equivariant to a Lie group such as the group of rotations (SO(d)) or rotations and reflections (O(d)). We use $\mathcal{G}$ as a generic symbol for such a Lie group. Based on this, we consider two classes of GNNs for geometric graphs: (1) $\mathcal{G}$-equivariant models, where the intermediate features and propagated messages are $\mathcal{G}$-equivariant geometric quantities such as vectors or tensors [15–19]; and (2) $\mathcal{G}$-invariant models, which only propagate $\mathcal{G}$-invariant scalar features such as distances and angles [20–22]. Despite promising empirical results for both classes of architectures, key theoretical questions remain unanswered: (1) How to characterise the expressive power of geometric GNNs? And (2) what is the tradeoff between $\mathcal{G}$-equivariant and $\mathcal{G}$-invariant GNNs?

Contributions. In this work, we study the expressive power of geometric GNNs from the perspective of discriminating non-isomorphic geometric graphs. We propose a geometric version of the Weisfeiler-Leman graph isomorphism test, termed GWL. (Figure 1). We use GWL to formally characterise classes of graphs that can and cannot be distinguished by $\mathcal{G}$-invariant and $\mathcal{G}$-equivariant GNNs. We show how invariant models have limited expressive power as they only reason locally via scalar quantities, while equivariant models distinguish a larger class of graphs by propagating geometric vector quantities beyond local neighbourhoods.

For Background and Preliminaries, please see Appendix A.
On the Expressive Power of Geometric Graph Neural Networks

Figure 1: Geometric Weisfeiler-Leman Test. GWL distinguishes non-isomorphic geometric graphs \(G_1\) and \(G_2\) by injectively assigning colours to distinct neighbourhood patterns, up to global symmetries (here \(\mathcal{G} = O(d)\)). Each iteration expands the neighbourhood from which geometric information can be gathered (shaded for node \(i\)). Example inspired by [23].

2 The Geometric Weisfeiler-Leman Test

Assumptions. Analogous to the WL test, the geometric and scalar features the nodes are equipped with come from countable subsets \(C \subset \mathbb{R}^d\) and \(C^\prime \subset \mathbb{R}\), respectively. As a consequence, when we require functions to be injective, we require them to be injective over the countable set of \(\mathcal{G}\)-orbits that are obtained by acting with the symmetry group \(\mathcal{G}\) on the dataset. This mimics the practically relevant situation of finite datasets, in which we have a finite pool \(P\) of geometric graphs (and their symmetry transformations) which we would like to distinguish.

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 the colour of each node \(i\) using the colours of its the 1-hop neighbourhood \(N_i\). In the geometric setting, \(N_i\) is an attributed point cloud around the central node \(i\). As a result, each neighbourhood carries two types of information: (1) neighbourhood type (invariant to \(\mathcal{G}\)) and (2) neighbourhood geometric orientation (equivariant to \(\mathcal{G}\)). From an axiomatic point of view, our generalisation of the WL neighbourhood aggregation procedure must meet two properties:

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

Property 2: Preservation of local geometry. A key property of WL is that the aggregation is injective. A \(\mathcal{G}\)-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 [24]. Thus, we must additionally update auxiliary geometric information variables in a way that is \(\mathcal{G}\)-equivariant and injective.

Geometric Weisfeiler-Leman (GWL). These intuitions motivate the following definition of the GWL test. At initialisation, we assign to each node \(i \in V\) a scalar node colour \(c_i \in C^\prime\) and an auxiliary object \(g_i\) containing the geometric information associated to it:

\[ c_i^{(0)} := \text{HASH}(s_i), \quad g_i^{(0)} := \left( c_i^{(0)}, \bar{v}_i \right), \]  

where HASH denotes an injective map over the scalar attributes \(s_i\) of node \(i\). 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:

\[ g_i^{(t)} := \left( c_i^{(t-1)}, \left\{ g_j^{(t-1)} \right\}, \left\{ (c_j^{(t-1)}, g_j^{(t-1)}, \bar{x}_{ij}) \mid j \in N_i \right\} \right), \]  

Here \(\left\{ \cdot \right\}\) denotes a multiset – a set in which elements may occur more than once. Importantly, the group \(\mathcal{G}\) 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:
\[
\mathbf{g} \cdot \mathbf{g}_i^{(t)} := (c_i^{(0)}, Q_h \mathbf{v}_i), \quad \mathbf{g} : \mathbf{g}_i^{(t)} := \left\{ (c_i^{(t-1)}, \mathbf{g} \cdot \mathbf{g}_i^{(t-1)}), \big\{ (c_j^{(t-1)}, \mathbf{g} \cdot \mathbf{g}_j^{(t-1)}, Q_h \mathbf{x}_{ij}) \mid j \in \mathcal{N}_i \big\} \right\}
\]

Clearly, the aggregation building $\mathbf{g}_i$ for any time-step $t$ is injective and $\mathcal{G}$-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 $i$:
\[
c_i^{(t)} := \text{I-HASH}^{(t)} \left( \mathbf{g}_i^{(t)} \right),
\]

by using a $\mathcal{G}$-orbit injective and $\mathcal{G}$-invariant function that we denote by $\text{I-HASH}$. That is for any geometric objects $\mathbf{g}, \mathbf{g}'$, $\text{I-HASH}(\mathbf{g}) = \text{I-HASH}(\mathbf{g}')$ if and only if there exists $\mathbf{g} \in \mathcal{G}$ such that $\mathbf{g} = \mathbf{g} \cdot \mathbf{g}'$.

**Overview.** With each iteration, $\mathbf{g}_i^{(t)}$ aggregates geometric information in progressively larger $t$-hop subgraph neighbourhoods $\mathcal{N}_i^{(t)}$ around the node $i$. The node colours summarise the structure of these $t$-hops via the $\mathcal{G}$-invariant aggregation performed by $\text{I-HASH}$. The procedure terminates when the partitions of the nodes induced by the colours do not change from the previous iteration. Finally, given two geometric graphs $\mathcal{G}$ and $\mathcal{H}$, if there exists some iteration $t$ for which $\left\{ c_i^{(t)} \mid i \in \mathcal{V}(\mathcal{G}) \right\} \neq \left\{ c_i^{(t)} \mid i \in \mathcal{V}(\mathcal{H}) \right\}$, 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 $\mathcal{G}$-equivariance, we also consider a more restrictive Invariant GWL (IGWL) that only updates node colours using the $\mathcal{G}$-orbit injective $\text{I-HASH}$ function and does not propagate geometric information:
\[
c_i^{(t)} := \text{I-HASH} \left( (c_i^{(t-1)}, \mathbf{v}_i), \big\{ (c_j^{(t-1)}, \mathbf{v}_j, \mathbf{x}_{ij}) \mid j \in \mathcal{N}_i \big\} \right),
\]

**IGWL with $k$-body scalars.** In order to further analyse the construction of the node colouring function $\text{I-HASH}$, we consider IGWL$_{(k)}$ based on the maximum number of nodes involved in the computation of $\mathcal{G}$-invariant scalars (also known as the ‘body order’ [25]):
\[
c_i^{(t)} := \text{I-HASH} (k) \left( (c_i^{(t-1)}, \mathbf{v}_i), \big\{ (c_j^{(t-1)}, \mathbf{v}_j, \mathbf{x}_{ij}) \mid j \in \mathcal{N}_i \big\} \right),
\]

and $\text{I-HASH} (k+1)$ is defined as:
\[
\text{HASH} \left( \big\{ \text{I-HASH} \left( (c_i^{(t-1)}, \mathbf{v}_i), \big\{ (c_j^{(t-1)}, \mathbf{v}_j, \mathbf{x}_{ij}), \ldots, (c_k^{(t-1)}, \mathbf{v}_k, \mathbf{x}_{jk}) \big\} \mid j \in (\mathcal{N}_i)^k \big\} \right),
\]

where $j = [j_1, \ldots, j_k]$ 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, $\text{I-HASH}(2)$ can identify neighbourhoods only up to pairwise distances among the central node and any of its neighbours (i.e. a 2-body scalar), while $\text{I-HASH}(3)$ up to distances and angles formed by any two edges (i.e. a 3-body scalar). Notably, distances and angles alone are incomplete descriptors of local geometry [26, 27]. Therefore, IGWL$_{(k)}$ with lower $k$ makes the colouring weaker.

3 Characterising the Expressive Power of GWL

3.1 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. 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 $\mathcal{G}_1 = (A_1, S_1, \mathbf{V}_1, \mathbf{X}_1)$ and $\mathcal{G}_2 = (A_2, S_2, \mathbf{V}_2, \mathbf{X}_2)$ such that the underlying attributed graphs $(A_1, S_1)$ and $(A_2, S_2)$ 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.

Addressed

R4.6

Addressed

R4.13
Recall that each iteration of GWL aggregates geometric information $g_i^{(k)}$ from progressively larger neighbourhoods $N_i^{(k)}$ around the node $i$, and distinguishes (sub-)graphs via comparing $G$-orbit injective colouring of $g_i^{(k)}$. We say $G_1$ and $G_2$ are $k$-hop distinct if for all graph isomorphisms $h$, there is some node $i \in V_1$, $b(i) \in V_2$ such that the corresponding $k$-hop subgraphs $N_i^{(k)}$ and $N_{b(i)}^{(k)}$ are distinct. Otherwise, we say $G_1$ and $G_2$ are $k$-hop identical if all $N_i^{(k)}$ and $N_{b(i)}^{(k)}$ 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 $G_1$ and $G_2$ where the underlying attributed graphs are isomorphic, and $k$ iterations are sufficient.

**Proposition 2.** Up to $k$ iterations, GWL cannot distinguish any $k$-hop identical geometric graphs $G_1$ and $G_2$ 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 $G_1$ and $G_2$ 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 $G_1$ and $G_2$ where the underlying attributed graphs are isomorphic.

We can now consider the more general case where the underlying attributed graphs for $G_1 = (A_1, S_1, \hat{V}_1, \hat{X}_1)$ and $G_2 = (A_2, S_2, \hat{V}_2, \hat{X}_2)$ are non-isomorphic and constructed from point clouds using radial cutoffs, as conventional for biochemistry and material science applications.

**Proposition 5.** Assuming geometric graphs are constructed from point clouds using radial cutoffs, GWL can distinguish any geometric graphs $G_1$ and $G_2$ where the underlying attributed graphs are non-isomorphic. At most $k_{\text{Max}}$ iterations are sufficient, where $k_{\text{Max}}$ is the maximum graph diameter among $G_1$ and $G_2$.

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 $G$-equivariant intermediate layers for graphs and geometric data, as prescribed in the Geometric Deep Learning blueprint [28], in addition to echoing similar intuitions in the computer vision community. As remarked by [24], 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.

### 3.2 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 $G$-equivariant 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 $G$-equivariant GNN is also distinguishable by GWL.

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

**Proposition 9.** $G$-equivariant GNNs have the same expressive power as GWL if the following conditions hold: (1) The aggregation $\text{AGG}$ is an injective, $G$-equivariant multiset function. (2) The scalar part of the update $\text{UPD}_s$ is a $G$-orbit injective, $G$-invariant multiset function. (3) The vector part of the update $\text{UPD}_v$ is an injective, $G$-equivariant multiset function. (4) The graph-level readout $f$ is an injective multiset function.

Similar statements can be made for $G$-invariant GNNs and IGWL. 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, discussed in Appendix E.
On the Expressive Power of Geometric Graph Neural Networks

References


A Background

A.1 Graph Isomorphism and the Weisfeiler-Leman Test

An attributed graph $\mathcal{G} = (A, S)$ with a node set $\mathcal{V}$ of size $n$ consists of an $n \times n$ adjacency matrix $A$ and a matrix of scalar features $S \in \mathbb{R}^{n \times f}$. 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 $s_i^{\mathcal{G}} = s_{b(i)}^{\mathcal{H}}$, 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 [1, 29]. 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 $c_i^{(t)} \in C$:

$$c_i^{(t)} := \text{HASH} \left( c_i^{(t-1)}, \{ c_j^{(t-1)} | j \in \mathcal{N}_i \} \right),$$

where $\text{HASH}$ is a hash function (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 $\{ c_i^{(t)} | i \in \mathcal{V}(\mathcal{G}) \} \neq \{ c_i^{(t)} | i \in \mathcal{V}(\mathcal{H}) \}$, then the graphs are not isomorphic. Otherwise, the WL test is inconclusive, and we say it cannot distinguish the two graphs.

Ever since Xu et al. [2], Morris et al. [3] noticed that Graph Neural Networks are at most as powerful as the Weisfeiler-Leman (WL) [1] test at distinguishing non-isomorphic graphs, the WL hierarchy became a powerful tool for analysing the expressive power of GNNs and guided the search for more expressive models [4–8].

A.2 Group Theory

We assume basic familiarity with group theory, see [30] for an overview. We denote the action of the group $\mathfrak{G}$ on a space $X$ by $g \cdot x$. If $\mathfrak{G}$ acts on spaces $X$ and $Y$, we say a function $f : X \to Y$ is $\mathfrak{G}$-equivariant if $f(g \cdot x) = g \cdot f(x)$. A function $f : X \to Y$ is $\mathfrak{G}$-invariant if $f(g \cdot x) = f(x)$.

The $\mathfrak{G}$-orbit of $x \in X$ is $\mathcal{O}_\mathfrak{G}(x) = \{ g \cdot x | g \in \mathfrak{G} \} \subseteq X$. 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 $f(g \cdot x) = f(x)$.

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 or by subtracting the centre of mass from all node positions. Given one of the standard groups above, for an element $g$ we denote by $M_g$ (or another capital letter) its standard matrix representation.

A.3 Geometric Graphs

A geometric graph $\mathcal{G} = (A, S, \tilde{V}, \tilde{X})$ with a node set $\mathcal{V}$ is an attributed graph that is also decorated with geometric attributes: node coordinates $\tilde{X} \in \mathbb{R}^{n \times d}$ and (optionally) vector features $\tilde{V} \in \mathbb{R}^{n \times d}$ (e.g. velocity, acceleration). Without loss of generality, we work with a single vector feature per node.

Our results generalise to multiple vector features or higher-order geometric tensors per node.

The geometric attributes transform as follows under the action of the relevant groups: (1) $S_n$ acts on the graph via $P_S \mathcal{G} := (P_S A P_\sigma, P_S S, P_S \tilde{V}, P_S \tilde{X})$; (2) Orthogonal transformations $Q_\mathfrak{G} \in \mathfrak{G}$ act on $\tilde{V}, \tilde{X}$ via $\tilde{V} Q_\mathfrak{G}, \tilde{X} Q_\mathfrak{G}$; and (3) Translations $T \in T(d)$ act on the coordinates $\tilde{X}$ via $\tilde{x}_i + T$ 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 $Q_\mathfrak{G} \in \mathfrak{G}$ and $T \in T(d)$:

$$\left( s_i^{\mathcal{G}}, \tilde{v}_i^{\mathcal{G}}, \tilde{x}_i^{\mathcal{G}} \right) = \left( s_{b(i)}^{\mathcal{H}}, Q_\mathfrak{G} \tilde{v}_{b(i)}^{\mathcal{H}}, Q_\mathfrak{G} \left( \tilde{x}_{b(i)}^{\mathcal{H}} + T \right) \right) \quad \text{for all } i \in \mathcal{V}(\mathcal{G}).$$

Addressed R3.1, R2.3, R2.7, R2.8, R4.8
On the Expressive Power of Geometric Graph Neural Networks

Note that if two geometric graphs are geometrically isomorphic, they are also isomorphic as attributed graphs. However, the converse is not true.

Systems in biochemistry [9], material science [10], physical simulations [11], and multiagent robotics [12] are conventionally modelled as geometric graphs. For example, molecules are represented as a set of nodes corresponding to atoms, which contain information about the atom type as well as its 3D spatial coordinates and other geometric quantities such as velocity or acceleration. The geometric attributes transform along with Euclidean transformations of the system. In biochemistry and material science, the conventional procedure for constructing the geometric graph \( \mathcal{G} = (\mathcal{A}, \mathcal{S}, \vec{V}, \vec{X}) \) is via the underlying point cloud \((\mathcal{S}, \vec{V}, \vec{X})\) using a predetermined radial cutoff \(r\). Thus, the adjacency matrix is defined as 

\[
[\mathcal{A}]_{ij} = \begin{cases} 
1, & \text{if } \| \vec{x}_i - \vec{x}_j \|_2 \leq r, \\
0, & \text{otherwise.}
\end{cases}
\]

Geometric graph isomorphism and distinguishing (sub-)graph geometries has important practical implications for representation learning. For e.g., in molecular systems, an ideal architecture should map distinct local structural environments around atoms to distinct embeddings in representation space [26, 27].

A.4 Geometric Graph Neural Networks

We consider two broad classes of geometric GNN architectures. \(\mathfrak{G}\)-equivariant GNN layers [15–19] update scalar and vector features from iteration \(t\) to \(t+1\) via learnable aggregate and update functions, \(\text{AGG}\) and \(\text{UPD}\), respectively:

\[
\begin{align*}
\mathbf{m}_i^{(t)}(s_j^{(t)}, \vec{v}_j^{(t)}, \vec{x}_{ij}) := & \text{AGG}\left( \left\{ (s_j^{(t)}, s_j^{(t)}, \vec{v}_j^{(t)}, \vec{x}_{ij}) \right\} \right) \quad \text{(Aggregate)} \quad (8) \\
\mathbf{s}_i^{(t+1)}(s_j^{(t)}, \vec{v}_j^{(t+1)}) := & \text{UPD}\left( (s_j^{(t)}, \vec{v}_j^{(t+1)}), (\mathbf{m}^{(t)}, \hat{\mathbf{m}}^{(t)}) \right) \quad \text{(Update)} \quad (9)
\end{align*}
\]

For e.g., PaiNN [23] interaction layers aggregate scalar and vector features via learnt radial filters:

\[
\begin{align*}
\mathbf{s}_i^{(t+1)} := & \mathbf{s}_i^{(t)} + \sum_{j \in \mathcal{N}_i} f_1\left( s_j^{(t)}, \| \vec{x}_{ij} \| \right) \quad (10) \\
\vec{v}_i^{(t+1)} := & \vec{v}_i^{(t)} + \sum_{j \in \mathcal{N}_i} f_2\left( s_j^{(t)}, \| \vec{x}_{ij} \| \right) \odot \vec{v}_j^{(t)} + \sum_{j \in \mathcal{N}_i} f_3\left( s_j^{(t)}, \| \vec{x}_{ij} \| \right) \odot \vec{x}_{ij} \quad (11)
\end{align*}
\]

Alternatively, \(\mathfrak{G}\)-invariant layers [20–22] do not update vector features and only aggregate scalar quantities from local neighbourhoods:

\[
\begin{align*}
\mathbf{s}_i^{(t+1)} := & \text{UPD}\left( \left\{ (s_j^{(t)}, \vec{v}_j, \vec{x}_{ij}) \right\} \right) \quad (12)
\end{align*}
\]

For e.g., SchNet [20] uses relative distances to scalarise local geometric information, while DimeNet [22] uses both distances and angles, as follows:

\[
\begin{align*}
\mathbf{s}_i^{(t+1)} := & \mathbf{s}_i^{(t)} + \sum_{j \in \mathcal{N}_i} f_1\left( s_j^{(t)}, \| \vec{x}_{ij} \| \right) \quad \text{(SchNet)} \quad (13) \\
\mathbf{s}_i^{(t+1)} := & \sum_{j \in \mathcal{N}_i} f_1\left( s_j^{(t)}, \mathbf{s}_j^{(t)}, \sum_{k \in \mathcal{N}_i \setminus \{j\}} f_2\left( s_k^{(t)}, \| \vec{x}_{ij} \|, \vec{x}_{ij} : \vec{x}_{ik} \right) \right) \quad \text{(DimeNet)} \quad (14)
\end{align*}
\]

For both \(\mathfrak{G}\)-invariant and \(\mathfrak{G}\)-equivariant architectures, the scalar features \(\{ s_j^{(T)} \} \) at the final iteration \(T\) are mapped to graph-level features via a permutation-invariant readout \(f : \mathbb{R}^{n \times f} \rightarrow \mathbb{R}^{f'}\).

Invariant GNNs have shown strong performance for protein design [31, 32] and electrocatalysis [33, 34], while equivariant GNNs are being used within learnt interatomic potentials for molecular dynamics [23, 35, 36].

B Discussion

Practical Implications. Proposition 10, together with Propositions 1 and 4, highlight critical theoretical limitations of \(\mathfrak{G}\)-invariant GNNs. Our results suggest that \(\mathfrak{G}\)-equivariant GNNs should be preferred when working with large geometric graphs such as macromolecules with thousands of nodes, where message passing is restricted to local radial neighbourhoods around each node.
Motivated by these limitations, two straightforward approaches to improving $\mathcal{G}$-invariant GNNs may be: (1) pre-computing non-local geometric properties as input features, e.g. models such as GemNet [33] and GearNet [31] successfully use two-hop dihedral angles. And (2) working with fully connected geometric graphs, as Proposition 7 suggests that $\mathcal{G}$-equivariant and $\mathcal{G}$-invariant GNNs can be made equally powerful when performing all-to-all message passing. This is supported by the empirical success of recent $\mathcal{G}$-invariant ‘Graph Transformers’ [34, 37] for small molecules with tens of nodes, where working with full graphs is tractable.

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, $\mathcal{G}$-equivariant multiset functions [27, 38–40]. Recent theoretical work on geometric GNNs and their universality has shown that Tensor Field Networks [41], GemNet [33], and GVP [17] can be universal approximators of continuous, $\mathcal{G}$-equivariant or $\mathcal{G}$-invariant multiset function over point clouds (not sparse graphs). In contrast, the GWL framework studies the expressive power of geometric GNNs from the perspective of geometric graph isomorphism. Geometric graph matching has also been studied from the perspective of finding global isometries in the computer vision community[42]. Our notion of geometric graph isomorphism is more general as it considers local message passing procedures as well as both scalar and geometric node attributes. Overall, our work formalises what classes of geometric graphs can and cannot be distinguished by message passing $\mathcal{G}$-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 9 as GWL relies on $\mathcal{G}$-orbit injective colouring and $\mathcal{G}$-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.

C 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 $\mathcal{G}$-orbits of the corresponding geometric multisets $g_1$ and $g_2$ are mutually exclusive, i.e. $\mathcal{O}_G(g_1) \cap \mathcal{O}_G(g_2) = \emptyset$. 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 $\mathcal{G}$-orbits would overlap, i.e. $g_1 = g_2$, for some $g \in \mathcal{G}$ and $\mathcal{O}_G(g_1) = \mathcal{O}_G(g_2) \Rightarrow c_1 = c_2$.

Proposition 1. GWL can distinguish any $k$-hop distinct geometric graphs $G_1$ and $G_2$ where the underlying attributed graphs are isomorphic, and $k$ iterations are sufficient.

Proof of Proposition 1. The $k$-th iteration of GWL identifies the $\mathcal{G}$-orbit of the $k$-hop subgraph $\mathcal{N}^{(k)}_i$ at each node $i$ via the geometric multiset $g^{(k)}_i$. $G_1$ and $G_2$ 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 $\mathcal{N}^{(k)}_i$ and $\mathcal{N}^{(k)}_{b(i)}$ are distinct. Thus, the $\mathcal{G}$-orbits of the corresponding geometric multisets $g^{(k)}_i$ and $g^{(k)}_{b(i)}$ are mutually exclusive, i.e. $\mathcal{O}_G(g^{(k)}_i) \cap \mathcal{O}_G(g^{(k)}_{b(i)}) = \emptyset \Rightarrow c^{(k)}_i \neq c^{(k)}_{b(i)}$. Thus, $k$ iterations of GWL are sufficient to distinguish $G_1$ and $G_2$. \hfill $\square$

Proposition 2. Up to $k$ iterations, GWL cannot distinguish any $k$-hop identical geometric graphs $G_1$ and $G_2$ where the underlying attributed graphs are isomorphic.

Proof of Proposition 2. The $k$-th iteration of GWL identifies the $\mathcal{G}$-orbit of the $k$-hop subgraph $\mathcal{N}^{(k)}_i$ at each node $i$ via the geometric multiset $g^{(k)}_i$. $G_1$ and $G_2$ 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 $\mathcal{N}^{(k)}_i$ and $\mathcal{N}^{(k)}_{b(i)}$ are identical up to group actions. Thus, the $\mathcal{G}$-orbits of the corresponding geometric multisets $g^{(k)}_i$ and
On the Expressive Power of Geometric Graph Neural Networks

Figure 2: Invariant GWL Test. IGWL cannot distinguish $G_1$ and $G_2$ as they are 1-hop identical: The $\Theta$-orbit of the 1-hop neighbourhood around each node is the same, and IGWL cannot propagate geometric orientation information beyond 1-hop (here $\Theta = O(d)$).

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 $G_1$ and $G_2$ as all 1-hop neighbourhoods are computationally identical.

\[ g_i^{(k)} \text{ overlap, i.e. } O_\Theta(g_i^{(k)}) = O_\Theta(g_{b(i)}^{(k)}) \Rightarrow c_i^{(k)} = c_{b(i)}^{(k)}. \] Thus, up to $k$ iterations of GWL cannot distinguish $G_1$ and $G_2$.\[ \Box \]

Proposition 3. IGWL can distinguish any 1-hop distinct geometric graphs $G_1$ and $G_2$ where the underlying attributed graphs are isomorphic, and 1 iteration is sufficient.

\[ \begin{array}{c}
\text{Proof of Proposition 3. Each iteration of IGWL identifies the } \Theta\text{-orbit of the 1-hop local neighbourhood } N_i^{(k=1)} \text{ at each node } i. \ G_1 \text{ and } G_2 \text{ being 1-hop distinct implies that there exists some bijection } b \text{ and some node } i \in V_1, b(i) \in V_2 \text{ such that the corresponding 1-hop local neighbourhoods } N_i^{(1)} \text{ and } N_{b(i)}^{(1)} \text{ are distinct. Thus, the } \Theta\text{-orbits of the corresponding geometric multisets } g_i^{(1)} \text{ and } g_{b(i)}^{(1)} \text{ are mutually exclusive, i.e. } O_\Theta(g_i^{(1)}) \cap O_\Theta(g_{b(i)}^{(1)}) \equiv \emptyset \Rightarrow c_i^{(1)} \neq c_{b(i)}^{(1)}. \text{ Thus, 1 iteration of IGWL is sufficient to distinguish } G_1 \text{ and } G_2. \ \Box \\
\end{array} \]

Proposition 4. Any number of iterations of IGWL cannot distinguish any 1-hop identical geometric graphs $G_1$ and $G_2$ where the underlying attributed graphs are isomorphic.

\[ \begin{array}{c}
\text{Proof of Proposition 4. Each iteration of IGWL identifies the } \Theta\text{-orbit of the 1-hop local neighbourhood } N_i^{(k=1)} \text{ at each node } i, \ G_1 \text{ and } G_2 \text{ being 1-hop identical implies that for all bijections } b \text{ and all nodes } i \in V_1, b(i) \in V_2, \text{ the corresponding 1-hop local neighbourhoods } N_i^{(k)} \text{ and } N_{b(i)}^{(k)} \text{ are identical up to group actions. Thus, the } \Theta\text{-orbits of the corresponding geometric multisets} \\
\end{array} \]
\[ g_i^{(1)} \text{ and } g_{b(i)}^{(1)} \text{ overlap, i.e. } \mathcal{O}_e(g_i^{(1)}) = \mathcal{O}_e(g_{b(i)}^{(1)}) \Rightarrow c_i^{(k)} = c_{b(i)}^{(k)}. \] Thus, any number of IGWL iterations cannot distinguish \( G_1 \) and \( G_2 \).

Proposition 5. Assuming geometric graphs are constructed from point clouds using radial cutoffs, GWL can distinguish any geometric graphs \( G_1 \) and \( G_2 \) where the underlying attributed graphs are non-isomorphic. At most \( k_{\text{Max}} \) iterations are sufficient, where \( k_{\text{Max}} \) is the maximum graph diameter among \( G_1 \) and \( G_2 \).

Proof of Proposition 5. We assume that a geometric graph \( G = (A, S, \bar{V}, \bar{X}) \) is constructed from a point cloud \((S, \bar{V}, \bar{X})\) using a predetermined radial cutoff \( r \). Thus, the adjacency matrix is defined as \( a_{ij} = 1 \) if \( \| \bar{x}_i - \bar{x}_j \| \leq r \), or 0 otherwise, for all \( a_{ij} \in A \). Such construction procedures are conventional for geometric graphs in biochemistry and material science.

Given geometric graphs \( G_1 \) and \( G_2 \) where the underlying attributed graphs are non-isomorphic, identify \( k_{\text{Max}} \) the maximum of the graph diameters of \( G_1 \) and \( G_2 \), and chose any arbitrary nodes \( i \in V_1, j \in V_2 \). We can define the \( k_{\text{Max}} \)-hop subgraphs \( N_i^{(k_{\text{Max}})} \) and \( N_j^{(k_{\text{Max}})} \) at \( i \) and \( j \), respectively. Thus, \( N_i^{(k_{\text{Max}})} = V_1 \) for all \( i \in V_1 \), and \( N_j^{(k_{\text{Max}})} = V_2 \) for all \( j \in V_2 \). Due to the assumed construction procedure of geometric graphs, \( N_i^{(k_{\text{Max}})} \) and \( N_j^{(k_{\text{Max}})} \) must be distinct. Otherwise, if \( N_i^{(k_{\text{Max}})} \) and \( N_j^{(k_{\text{Max}})} \) were identical up to group actions, the sets \((S_1, \bar{V}_1, \bar{X}_1)\) and \((S_2, \bar{V}_2, \bar{X}_2)\) would have yielded isomorphic graphs.

The \( k_{\text{Max}} \)-th iteration of GWL identifies the \( \mathfrak{S} \)-orbit of the \( k_{\text{Max}} \)-hop subgraph \( N_i^{(k_{\text{Max}})} \) at each node \( i \) via the geometric multiset \( g_i^{(k_{\text{Max}})} \). As \( N_i^{(k_{\text{Max}})} \) and \( N_j^{(k_{\text{Max}})} \) are distinct for any arbitrary nodes \( i \in V_1, j \in V_2 \), the \( \mathfrak{S} \)-orbits of the corresponding geometric multisets \( g_i^{(k_{\text{Max}})} \) and \( g_j^{(k_{\text{Max}})} \) are mutually exclusive, i.e. \( \mathcal{O}_e(g_i^{(k_{\text{Max}})}) \cap \mathcal{O}_e(g_j^{(k_{\text{Max}})}) = \emptyset \Rightarrow c_i^{(k_{\text{Max}})} \neq c_j^{(k_{\text{Max}})}. \) Thus, \( k_{\text{Max}} \) iterations of GWL are sufficient to distinguish \( G_1 \) and \( G_2 \).

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 \( g_i \) for all \( i \in V \), i.e. \( g_i^{(0)} = g_i^{(t-1)} = g_i^{(0)} \equiv (s_i, \bar{v}_i) \). Thus, GWL is at least as powerful as IGWL, which does not update \( g_i \).

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 them. 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 \( G_1 \) and \( G_2 \) which GWL can distinguish, but IGWL cannot.

If the underlying attributed graphs of \( G_1 \) and \( G_2 \) are isomorphic, by Proposition 1 and Proposition 4, \( G_1 \) and \( G_2 \) are 1-hop identical but \( k \)-hop distinct for some \( k > 1 \). For all bijections \( b \) and all nodes \( i \in V_1, b(i) \in V_2 \), the local neighbourhoods \( N_i^{(1)} \) and \( N_{b(i)}^{(1)} \) are identical up to group actions, and \( \mathcal{O}_e(g_i^{(1)}) = \mathcal{O}_e(g_{b(i)}^{(1)}) \Rightarrow c_{i}^{(1)} = c_{b(i)}^{(1)}. \) Additionally, there exists some bijection \( b \) and some nodes \( i \in V_1, b(i) \in V_2 \) such that the \( k \)-hop subgraphs \( N_i^{(k)} \) and \( N_{b(i)}^{(k)} \) are distinct, and \( \mathcal{O}_e(g_i^{(k)}) \cap \mathcal{O}_e(g_{b(i)}^{(k)}) = \emptyset \Rightarrow c_{i}^{(k)} \neq c_{b(i)}^{(k)}. \) However, as \( G_1 \) and \( G_2 \) are fully connected, for any \( k \), \( N_i^{(1)} = N_i^{(k)} \) and \( N_{b(i)}^{(1)} = N_{b(i)}^{(k)} \) are identical up to group actions. Thus, \( \mathcal{O}_e(g_i^{(1)}) = \mathcal{O}_e(g_i^{(k)}) = \mathcal{O}_e(g_{b(i)}^{(1)}) = \mathcal{O}_e(g_{b(i)}^{(k)}) \Rightarrow c_{i}^{(1)} = c_{i}^{(k)} = c_{b(i)}^{(1)} = c_{b(i)}^{(k)}. \) This is a contradiction.

If \( G_1 \) and \( G_2 \) are non-isomorphic and fully connected, for any arbitrary \( i \in V_1, j \in V_2 \) and any \( k \)-hop neighbourhood, we know that \( N_i^{(1)} = N_i^{(k)} \) and \( N_j^{(1)} = N_j^{(k)} \). Thus, a single iteration of GWL and
IGWL identify the same $\mathcal{O}$-orbits and assign the same node colours, i.e. $\mathcal{O}_\mathcal{G}(g_i^{(1)}) = \mathcal{O}_\mathcal{G}(g_k^{(k)}) \Rightarrow c_i^{(1)} = c_k^{(k)}$ and $\mathcal{O}_\mathcal{G}(g_j^{(1)}) = \mathcal{O}_\mathcal{G}(g_j^{(k)}) \Rightarrow c_j^{(1)} = c_j^{(k)}$. This is a contradiction.\hfill $\square$

### D Proofs for equivalence between GWL and Geometric GNNs

Our proofs adapt the techniques used in [2, 3] for connecting 1-WL with GNNs. Note that we omit including the relative position vectors $\vec{x}_{ij} = \vec{x}_i - \vec{x}_j$ 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 $\mathcal{G}$-equivariant GNN is also distinguishable by GWL.

**Proof of Theorem 8.** Consider two geometric graphs $G$ and $H$. The theorem implies that if the GNN graph-level readout outputs $f(G) \neq f(H)$, then the GWL test will always determine $G$ and $H$ to be non-isomorphic, i.e. $G \not\cong H$.

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

We will show that on the same graph for nodes $i$ and $k$, if $(c_i^{(t)}, g_i^{(t)}) = (c_k^{(t)}, g \cdot g_i^{(t)})$, we always have GNN features $(s_i^{(t)}, \vec{v}_i^{(t)}) = (s_k^{(t)}, Q_g \vec{v}_k^{(t)})$ 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 $k$, $(c_i^{(t+1)}, g_i^{(t+1)}) = (c_k^{(t+1)}, g \cdot g_i^{(t+1)})$, then:

$$\left\{(c_i^{(t)}, g_i^{(t)}) \mid j \in \mathcal{N}_i\right\} = \left\{(c_k^{(t)}, g \cdot g_i^{(t)}) \mid j \in \mathcal{N}_k\right\} \quad (15)$$

By our assumption on iteration $t$,

$$\left\{(s_i^{(t)}, \vec{v}_i^{(t)}) \mid j \in \mathcal{N}_i\right\} = \left\{(s_k^{(t)}, Q_g \vec{v}_k^{(t)}) \mid j \in \mathcal{N}_k\right\} \quad (16)$$

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, $(s_k^{(t+1)}, \vec{v}_k^{(t+1)}) = (s_k^{(t+1)}, Q_g \vec{v}_k^{(t+1)})$. By induction, if $(c_i^{(t)}, g_i^{(t)}) = (c_k^{(t)}, g \cdot g_i^{(t)})$, we always have GNN node features $(s_i^{(t)}, \vec{v}_i^{(t)}) = (s_k^{(t)}, Q_g \vec{v}_k^{(t)})$ for any iteration $t$. This creates valid mappings $\phi_s, \phi_v$ such that $s_i^{(t)} = \phi_s(c_i^{(t)})$ and $\vec{v}_i^{(t)} = \phi_v(c_i^{(t)}, g_i^{(t)})$ for any $i \in \mathcal{V}$.

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

$$\left\{(s_j^{(t)}, \vec{v}_i^{(t)}) \mid j \in \mathcal{N}_i\right\} = \left\{\phi_s(c_i^{(t)}), \phi_v(c_i^{(t)}, g_i^{(t)}) \mid j \in \mathcal{N}_i\right\} \quad (17)$$

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

**Proposition 9.** $\mathcal{G}$-equivariant GNNs have the same expressive power as GWL if the following conditions hold: (1) The aggregation $A_G^G$ is an injective, $\mathcal{G}$-equivariant multiset function. (2) The scalar part of the update $U_{PD}$ is a $\mathcal{G}$-orbit injective, $\mathcal{G}$-invariant multiset function. (3) The vector part of the update $U_{PD}$ is an injective, $\mathcal{G}$-equivariant multiset function. (4) The graph-level readout $f$ is an injective multiset function.

**Proof of Theorem 9.** 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. $s^{(t)} \equiv c^{(t)}$.\hfill $\square$
Let $G$ and $H$ 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 $G$ and $H$ into different multisets of node features.

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

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

\[
s^{(t)}_i = \text{UPD}_s \left( \left( s^{(t-1)}_i, v^{(t-1)}_i \right), \text{AGG} \left( \left\{ (s^{(t-1)}_j, s^{(t-1)}_j, v^{(t-1)}_j, v^{(t-1)}_j) | j \in \mathcal{N}_i \right\} \right) \right)
\]

\[
v^{(t)}_i = \text{UPD}_v \left( \left( v^{(t-1)}_i \right), \text{AGG} \left( \left\{ (s^{(t-1)}_j, v^{(t-1)}_j, v^{(t-1)}_j) | j \in \mathcal{N}_i \right\} \right) \right)
\]

with the aggregation function $\text{AGG}$ being $\Phi$-equivariant and injective, the scalar update function $\text{UPD}_s$ being $\Phi$-invariant and injective, and the vector update function $\text{UPD}_v$ being $\Phi$-equivariant and injective.

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

\[
c^{(t)}_i = h_s \left( \left( c^{(t-1)}_i, g^{(t-1)}_i \right), \left\{ (c^{(t-1)}_j, g^{(t-1)}_j) | j \in \mathcal{N}_i \right\} \right),
\]

\[
g^{(t)}_i = h_v \left( \left( c^{(t-1)}_i, g^{(t-1)}_i \right), \left\{ (c^{(t-1)}_j, g^{(t-1)}_j) | j \in \mathcal{N}_i \right\} \right),
\]

where $h_s$ is a $\Phi$-invariant and injective map, and $h_v$ is a $\Phi$-equivariant and injective operation (e.g. in equation 2, expanding the geometric multiset by copying).

We will show by induction that at any iteration $t$, there always exist injective functions $\varphi_s$ and $\varphi_v$ such that $s^{(t)}_i = \varphi_s(c^{(t)}_i)$ and $v^{(t)}_i = \varphi_v(c^{(t)}_i, g^{(t)}_i)$. This holds for $t = 0$ because the initial node features are the same for GWL and GNN, $c^{(0)}_i \equiv s^{(0)}_i$ and $g^{(0)}_i \equiv (s^{(0)}_i, v^{(0)}_i)$ for all $i \in \mathcal{V}(G), \mathcal{V}(H)$.

Suppose this holds for iteration $t$. At iteration $t + 1$, substituting $s^{(t)}_i$ with $\varphi_s(c^{(t)}_i)$, and $v^{(t)}_i$ with $\varphi_v(c^{(t)}_i, g^{(t)}_i)$ gives us

\[
s^{(t+1)}_i = \text{UPD}_s \left( \left( \varphi_s(c^{(t)}_i), \varphi_v(c^{(t)}_i, g^{(t)}_i) \right), \text{AGG} \left( \left\{ (\varphi_s(c^{(t)}_j), \varphi_s(c^{(t)}_j), \varphi_v(c^{(t)}_j, g^{(t)}_j), \varphi_v(c^{(t)}_j, g^{(t)}_j)) | j \in \mathcal{N}_i \right\} \right) \right)
\]

\[
v^{(t+1)}_i = \text{UPD}_v \left( \left( \varphi_v(c^{(t)}_i), \varphi_v(c^{(t)}_i, g^{(t)}_i) \right), \text{AGG} \left( \left\{ (\varphi_v(c^{(t)}_j), \varphi_v(c^{(t)}_j, \varphi_v(c^{(t)}_j, g^{(t)}_j), \varphi_v(c^{(t)}_j, g^{(t)}_j)) | j \in \mathcal{N}_i \right\} \right) \right)
\]

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

\[
s^{(t+1)}_i = g_s \left( \left( c^{(t)}_i, g^{(t)}_i \right), \left\{ (c^{(t)}_j, g^{(t)}_j) | j \in \mathcal{N}_i \right\} \right),
\]

\[
v^{(t+1)}_i = g_v \left( \left( c^{(t)}_i, g^{(t)}_i \right), \left\{ (c^{(t)}_j, g^{(t)}_j) | j \in \mathcal{N}_i \right\} \right),
\]

We can then consider:

\[
s^{(t+1)}_i = g_s \circ h^{-1}_s \circ h_s \left( \left( c^{(t)}_i, g^{(t)}_i \right), \left\{ (c^{(t)}_j, g^{(t)}_j) | j \in \mathcal{N}_i \right\} \right),
\]

\[
v^{(t+1)}_i = g_v \circ h^{-1}_v \circ h_v \left( \left( c^{(t)}_i, g^{(t)}_i \right), \left\{ (c^{(t)}_j, g^{(t)}_j) | j \in \mathcal{N}_i \right\} \right),
\]

Then, we can denote $\varphi_s = g_s \circ h^{-1}_s$ and $\varphi_v = g_v \circ h^{-1}_v$ as injective functions because the composition of injective functions is injective. Hence, for any iteration $t + 1$, there exist injective functions $\varphi_s$ and $\varphi_v$ such that:

\[
s^{(t+1)}_i = \varphi_s \left( c^{(t+1)}_i \right)
\]

\[
v^{(t+1)}_i = \varphi_v \left( c^{(t+1)}_i, g^{(t+1)}_i \right).
\]

At the $T$-th iteration, the GWL test decides that $G$ and $H$ are non-isomorphic, which means the multisets of node colours $\{c^{(T)}_i\}$ are different for $G$ and $H$. The GNN’s node scalar features $\{s^{(T)}_i\} = \{\varphi_s(c^{(T)}_i)\}$ must also be different for $G$ and $H$ because of the injectivity of $\varphi_s$. □
E Understanding the Design Space of Geometric GNNs via GWL

We can use the GWL framework to better understand key design choices for building geometric GNNs [25]: (1) Depth or number of layers; and (2) Body order of invariant scalars. In doing so, we formalise theoretical limitations of current geometric GNNs and provide practical implications.

E.1 Role of Depth: Propagating Geometric Information

Each iteration of GWL expands the neighbourhood from which geometric information can be gathered. Thus, stacking multiple \( \varnothing \)-equivariant GNN layers enables the computation of compositional geometric features.

This can be understood via a geometric version of computation trees [43]. A computation tree \( T_i^{(t)} \) represents the maximum information contained in GWL/IGWL colours or GNN features at iteration \( t \) by an ‘unrolling’ of the message passing procedure. Geometric computation trees are constructed recursively: \( T_i^{(0)} = (s_i, \vec{v}_i) \) for all \( i \in V \). For \( t > 0 \), we start with a root node \((s_j, \vec{v}_j)\) and add a child subtree \( T_j^{(t-1)} \) for all \( j \in N_i \), along with the relative position \( \vec{x}_{ij} \) along the edge, as shown in Figure 3. To obtain the root node’s embedding or colour, both scalar and geometric information is propagated from the leaves up to the root. Thus, if two nodes have identical geometric computation trees, they will be mapped to the same node embedding or colour.

Critically, geometric orientation information cannot flow from one level to another in the computation trees for IGWL and \( \varnothing \)-invariant GNNs, as they only update scalar information. In the recursive construction procedure, we must insert a connector node \((s_j, \vec{v}_j)\) before adding the child subtree \( T_j^{(t-1)} \) for all \( j \in N_i \) and prevent geometric information propagation between them. Consequently, even the most powerful \( \varnothing \)-invariant GNNs are restricted in their ability to compute global and non-local geometric properties.

**Proposition 10.** IGWL and \( \varnothing \)-invariant GNNs cannot decide several geometric graph properties: (1) perimeter, surface area, and volume of the bounding box/sphere enclosing the geometric graph; (2) distance from the centroid or centre of mass; and (3) dihedral angles.

**Practical Implications.** Proposition 10, together with Propositions 1 and 4, highlight critical theoretical limitations of \( \varnothing \)-invariant GNNs. Our results suggest that \( \varnothing \)-equivariant GNNs should be preferred when working with large geometric graphs such as macromolecules with thousands of nodes, where message passing is restricted to local radial neighbourhoods around each node.

Motivated by these limitations, two straightforward approaches to improving \( \varnothing \)-invariant GNNs may be: (1) pre-computing non-local geometric properties as input features, e.g. models such as GemNet [33] and GearNet [31] successfully use two-hop dihedral angles. And (2) working with fully connected geometric graphs, as Proposition 7 suggests that \( \varnothing \)-equivariant and \( \varnothing \)-invariant GNNs can be made equally powerful when performing all-to-all message passing. This is supported by the empirical success of recent \( \varnothing \)-invariant ‘Graph Transformers’ [34, 37] for small molecules with tens of nodes, where working with full graphs is tractable.

**Synthetic Experiment: \( k \)-chains.** We present a simple synthetic experiment to demonstrate the role of depth in propagating geometric information beyond local neighbourhoods. We consider \( k \)-chain geometric graphs, which generalise the examples from [23] and the pair of graphs presented in Figure 3. Each pair of \( k \)-chains consists of \( k + 2 \) nodes with \( k \) nodes arranged in a line and differentiated by the orientation of the 2 end points. Thus, \( k \)-chain graphs are \((k-1)\)-hop distinguishable, and \((k-1)\) iterations of GWL are sufficient to distinguish them. In Table 1, we find that \((k-1)\) layers of \( \varnothing \)-equivariant message passing can learn to perfectly distinguish any \( k \)-chains, while \( \varnothing \)-invariant GNNs are unable to distinguish \( k \)-chains for \( k \geq 2 \).

E.2 Role of Body Order: Distinguishing \( \varnothing \)-Orbits

At each iteration of GWL and IGWL, the I-HASH function assigns a \( \varnothing \)-invariant colouring to distinct geometric neighbourhood patterns. For geometric GNNs, this corresponds to scalarising local geometric information when updating the scalar features, as shown in equation 13 and equation 14. 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 1-WL for non-geometric graphs.
Table 1: \(k\)-chain geometric graphs. \(k\)-chain graphs are \((k-1)\)-hop distinguishable, and \((k-1)\) iterations of \(\Phi\)-equivariant message passing are sufficient to distinguish them, while \(\Phi\)-invariant message passing is unable to distinguish \(k\)-chains for \(k \geq 2\) (for \(\Phi = O(3)\)).

<table>
<thead>
<tr>
<th>GNN Layer</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rand. guess</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>Non-geom.</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>(O(3))-Inv.</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>(O(3))-Equiv.</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Figure 4: Two geometric graphs for which IGWL and \(\Phi\)-invariant GNNs cannot distinguish their geometric properties such as perimeter, surface area, and volume of the bounding box/sphere, distance from the centroid, and dihedral angles. The centroid is denoted by a red point and distances from it are denoted by dotted red lines. The bounding box enclosing the geometric graph is denoted by the dotted green lines.

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

**Proposition 11.** \(I\)-HASH\(_{(m)}\) is \(\Phi\)-orbit injective for \(m = \max\{|N_i| \mid i \in V\}\), the maximum cardinality of all local neighbourhoods \(N_i\) 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 \(\Phi\)-invariant aggregators used in current geometric GNN layers, e.g. SchNet [20], E-GNN [18], and Tensor Field Networks [15] use distances; DimeNet [22] uses distances and angles; and MACE [36] constructs scalars up to arbitrary \(k\) via Atomic Cluster Expansion [39]. We can state the following about the IGWL\(_{(k)}\) hierarchy and the corresponding GNNs.

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

Finally, we show that IGWL\(_{(2)}\) is equivalent to 1-WL when all the pairwise distances between the nodes are the same. A similar observation was recently made by [44].

**Proposition 13.** Let \(G_1 = (A_1, S_1, \bar{X}_1)\) and \(G_2 = (A_2, S_2, \bar{X}_2)\) be two geometric graphs with the property that all edges have equal length. Then, IGWL\(_{(2)}\) distinguishes the two graphs if and only if 1-WL can distinguish the attributed graphs \((A_1, S_1)\) and \((A_1, S_1)\).

This equivalence also points out the limitations of \(\Phi\)-invariant models like SchNet [20] and CGCNN [21] which only rely on pairwise distances and consequently suffer from all well-known failure cases of 1-WL. For instance, such models cannot distinguish two equilateral triangles from the regular hexagon [22].
E.3 Geometric GNN Design Space Proofs

Proposition 10. IGWL and \(\varnothing\)-invariant GNNs cannot decide several geometric graph properties:
1) perimeter, surface area, and volume of the bounding box/sphere enclosing the geometric graph;
2) distance from the centroid or centre of mass; and (3) dihedral angles.

Proof of Proposition 10. Following [43], we say that a class of models decides a geometric graph property if there exists a model belonging to this class such that for any two geometric graphs that differ in the property, the model is able to distinguish the two geometric graphs.

In Figure 4 we provide an example of two geometric graphs that demonstrate the proposition. \(G_1\) and \(G_2\) differ in the following geometric graph properties:

- Perimeter, surface area, and volume of the bounding box enclosing the geometric graph\(^1\): (32 units, 40 units\(^2\), 16 units\(^3\)) vs. (28 units, 24 units\(^2\), 8 units\(^3\)).
- Multiset of distances from the centroid or centre of mass: \(\{0.00, 1.00, 1.00, 2.45, 2.45\}\) vs. \(\{0.40, 1.08, 1.08, 2.32, 2.32\}\).
- Dihedral angles: \(\angle(1jk) = \frac{(|\vec{x}_j \times \vec{x}_k| \cdot |\vec{x}_j \times \vec{x}_m|)}{|\vec{x}_j \times \vec{x}_k| \cdot |\vec{x}_j \times \vec{x}_m|}\) are clearly different for the two graphs.

However, according to Proposition 4 and Theorem ??, both IGWL and \(\varnothing\)-invariant GNNs cannot distinguish these two geometric graphs, and therefore, cannot decide all these properties.

We can also show this by constructing geometric computation trees for any number of IGWL or \(\varnothing\)-invariant GNN iterations, as illustrated in Figure 3. We observe that the geometric computation trees of any pair of isomorphic nodes are identical, as all 1-hop neighbourhoods are computationally identical. Therefore, the set of node colours or node scalar features will also be identical, which implies that \(G_1\) and \(G_2\) cannot be distinguished.

Proposition 11. I-HASH\(_{(m)}\) is \(\varnothing\)-orbit injective for \(m = \max(\{|N_i| \mid i \in V\})\), the maximum cardinality of all local neighbourhoods \(N_i\) in a given dataset.

Proof of Proposition 11. As \(m\) is the maximum cardinality of all local neighbourhoods \(N_i\) under consideration, any distinct neighbourhoods \(N_1\) and \(N_2\) 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 \(N_1\) and \(N_2\). Thus, I-HASH\(_{(m)}\) is \(\varnothing\)-orbit injective.

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

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

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

- For \(k = 3\) and \(\varnothing = O(3)\) or \(SO(3)\), for the local neighbourhood from Figure 1 in [23], two configurations with different angles between the neighbouring nodes can be distinguished by IGWL\(_{(3)}\) but not by IGWL\(_{(2)}\).
- For \(k = 4\) and \(\varnothing = O(3)\) or \(SO(3)\), the pair of local neighbourhoods from Figure 1 in [27] can be distinguished by IGWL\(_{(4)}\) but not by IGWL\(_{(3)}\).
- For \(k = 5\) and \(\varnothing = O(3)\), the pair of local neighbourhoods from Figure 2(e) in [27] can be distinguished by IGWL\(_{(5)}\) but not by IGWL\(_{(4)}\).

\(^1\)The same result applies for the bounding sphere, not shown in the figure.
• For $k = 5$ and $\mathfrak{g} = SO(3)$, the pair of local neighbourhoods from Figure 2(f) in [27] can be
distinguished by IGWL(5) but not by IGWL(4).

Proposition 13. Let $G_1 = (A_1, S_1, \vec{X}_1)$ and $G_2 = (A_2, S_2, \vec{X}_2)$ be two geometric graphs with the
property that all edges have equal length. Then, IGWL(2) distinguishes the two graphs if and only if
1-WL can distinguish the attributed graphs $(A_1, S_1)$ and $(A_1, S_1)$.

Proof of Proposition 13. Let $c$ and $k$ the colours produced by IGWL(2) and 1-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 $\{ (c_p^{(t)}, \|\vec{x}_{ip}\|) | p \in N_i \} = \{ (c_p^{(t)}, \|\vec{x}_{jp}\|) | p \in N_j \}$,
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 $\{ k_p^{(t)} | p \in N_i \} = \{ k_p^{(t)} | p \in N_j \}$. Therefore, this is equivalent to $k_i^{(t+1)} = k_j^{(t+1)}$.