
Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks

Cristian Bodnar^{*1} Fabrizio Frasca^{*2,3} Yu Guang Wang^{*4,5,6}
Nina Otter⁷ Guido Montúfar^{*4,7} Pietro Liò¹ Michael M. Bronstein^{2,3}

Abstract

The pairwise interaction paradigm of graph machine learning has predominantly governed the modelling of relational systems. However, graphs alone cannot capture the multi-level interactions present in many complex systems and the expressive power of such schemes was proven to be limited. To overcome these limitations, we propose Message Passing Simplicial Networks (MPSNs), a class of models that perform message passing on simplicial complexes (SCs). To theoretically analyse the expressivity of our model we introduce a Simplicial Weisfeiler-Lehman (SWL) colouring procedure for distinguishing non-isomorphic SCs. We relate the power of SWL to the problem of distinguishing non-isomorphic graphs and show that SWL and MPSNs are strictly more powerful than the WL test and not less powerful than the 3-WL test. We deepen the analysis by comparing our model with traditional graph neural networks (GNNs) with ReLU activations in terms of the number of linear regions of the functions they can represent. We empirically support our theoretical claims by showing that MPSNs can distinguish challenging strongly regular graphs for which GNNs fail and, when equipped with orientation equivariant layers, they can improve classification accuracy in oriented SCs compared to a GNN baseline.

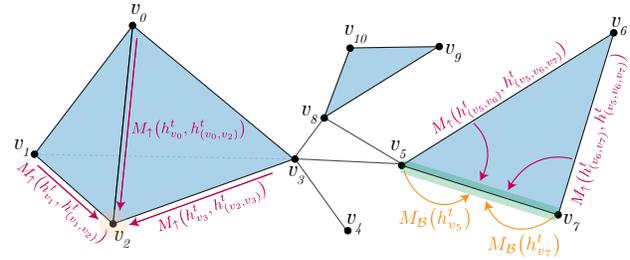


Figure 1. Message Passing with upper and boundary adjacencies illustrated for vertex v_2 and edge (v_5, v_7) in the complex.

1. Introduction

Graphs are among the most common abstractions for complex systems of relations and interactions, arising in a broad range of fields from social science to high energy physics. Graph neural networks (GNNs), which trace their origins to the 1990s (Sperduti, 1994; Goller & Kuchler, 1996; Gori et al., 2005; Scarselli et al., 2009; Bruna et al., 2014; Li et al., 2015), have recently achieved great success in learning tasks with graph-structured data.

GNNs typically apply a local permutation-invariant functions aggregating the neighbour features for each node, resulting in a permutation equivariant function on the graph (Maron et al., 2018). The design of the local aggregator is important: if chosen to be an injective function, the resulting GNN is equivalent in its expressive power to the Weisfeiler-Lehman (WL) graph isomorphism test (Weisfeiler & Leman, 1968; Xu et al., 2019b; Morris et al., 2019). Due to their equivalence to the WL algorithm, message-passing type GNNs are unable to learn certain tasks on graphs. In particular, they are limited in their capability of detecting graph structure such as triangles or cliques (Chen et al., 2020). While other architectures equivalent to higher-dimensional k -WL tests have been proposed (Maron et al., 2019), they suffer from high computational and memory complexity, and lack the key property of GNNs: Locality.

We tackle this problem by considering local higher-order interactions. Among many modelling frameworks that have been proposed to describe complex systems with higher-order relations (Battiston et al., 2020), we specifically focus on simplicial complexes, a convenient middle ground between graphs (which are a particular case of simplicial

^{*}Equal contribution ¹Department of Computer Science and Technology, University of Cambridge, UK ²Twitter, UK ³Department of Computing, Imperial College London, UK ⁴Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany ⁵Institute of Natural Sciences and School of Mathematical Sciences, Shanghai Jiao Tong University, China ⁶School of Mathematics and Statistics, University of New South Wales, Sydney, Australia ⁷Department of Mathematics and Department of Statistics, University of California, Los Angeles, USA. Correspondence to: Cristian Bodnar <cb2015@cam.ac.uk>, Fabrizio Frasca <ffrasca@twitter.com>, Yu Guang Wang <yuguang.wang@mis.mpg.de>.

complexes) and more general hypergraphs. Importantly, they offer strong mathematical connections to algebraic and differential topology and geometry. The simplicial Hodge Laplacian (Barbarossa & Sardellitti, 2020; Schaub et al., 2020), a discrete counterpart of the Laplacian operator in Hodge–de Rham theory (Rosenberg, 1997), provides a connection with the theory of spectral analysis and signal processing on these higher-dimensional domains.

We start by constructing a Simplicial Weisfeiler-Lehman (SWL) test for distinguishing non-isomorphic simplicial complexes. Motivated by this theoretical construction, we propose Message Passing Simplicial Networks (MPSNs), a message passing neural architecture for simplicial complexes that extends previous approaches such as GNNs and spectral simplicial convolutions (Bunch et al., 2020; Ebli et al., 2020). We then show that the proposed MPSN is as powerful as SWL. Strictly better than the conventional WL test (Weisfeiler & Leman, 1968), MPSN can be used to distinguish non-isomorphic graphs. We also show that the SWL test and MPSN are not less powerful than the 3-WL test (Cai et al., 1992; Morris et al., 2019).

Moreover, we explore the expressive power of GNNs and MPSNs in terms of the number of linear regions of the functions they can represent (Pascanu et al., 2013; Montúfar et al., 2014). We obtain bounds for the maximal number of linear regions of MPSNs and show a higher functional complexity than GNNs and simplicial convolutional neural networks (SCNNs) (Ebli et al., 2020), for which we provide optimal bounds that might be of independent interest. Proofs are presented in the Appendix.

2. Background

In this section, we focus on introducing the required background on (oriented) simplicial complexes. We assume basic familiarity with GNNs and the WL test.

Definition 1 (Nanda (2021)). *Let V be a non-empty vertex set. A simplicial complex \mathcal{K} is a collection of nonempty subsets of V with the property that it is closed under taking subsets and it contains all the singleton subsets of V .*

A member $\sigma = \{v_0, \dots, v_k\} \in \mathcal{K}$ with cardinality $k + 1$ is called a k -dimensional simplex or simply a k -simplex. Geometrically, one can see 0-simplices as *vertices*, 1-simplices as *edges*, 2-simplices as *triangles*, and so on (see Figure 1).

Definition 2 (Boundary incidence relation). *We say $\sigma \prec \tau$ iff $\sigma \subset \tau$ and there is no δ such that $\sigma \subset \delta \subset \tau$.*

This relation describes what simplices are on the boundary of another simplex. For instance vertices $\{v_1\}, \{v_2\}$ are on the boundary of edge $\{v_1, v_2\}$ and edge $\{v_5, v_7\}$ is on the boundary of triangle $\{v_5, v_6, v_7\}$.

In algebraic topology (Nanda, 2021) and discrete differen-

tial geometry (Crane et al., 2013), it is common to equip the simplices in a complex with an additional structure called an *orientation*. An *oriented k -simplex* is a k -simplex with a chosen order for its vertices. They can be uniquely represented as a tuple of vertices (v_0, \dots, v_k) . A simplicial complex with a chosen orientation for all of its simplices is called oriented. We note that mathematically, the choice of orientation is completely arbitrary.

Intuitively, orientations describe a walk over the vertices of a simplex. For instance an edge $\{v_1, v_2\}$ has two orientations (v_1, v_2) and (v_2, v_1) , conveying a movement from v_1 to v_2 and from v_2 to v_1 , respectively. Similarly, triangle $\{v_0, v_1, v_2\}$ has six possible orientations given by all the permutations of its vertices. However, some of these, like (v_0, v_1, v_2) and (v_2, v_0, v_1) , or (v_0, v_2, v_1) and (v_2, v_1, v_0) , are equivalent, because (ignoring the starting point) they describe the same clockwise or counter-clockwise movement over the triangle. These two equivalence classes can be generalised to simplices of any dimension based on how the vertices are permuted (see Hatcher (2000) for details).

Definition 3. *An oriented k -simplex is positively oriented if its vertices form an even permutation and negatively oriented otherwise.*

If \mathcal{K} is oriented, we can equip \prec with this additional information. Consider two oriented simplices with $\tau \prec \sigma$ and $\dim(\sigma) > 1$. We say τ and σ have the same orientation $\tau \prec_+ \sigma$ if τ shows up in some even permutation of the vertices of σ . Otherwise, we have $\tau \prec_- \sigma$. Edges (v_i, v_j) are a special case and we have $v_j \prec_+ (v_i, v_j)$ and $v_i \prec_- (v_i, v_j)$. The oriented boundary relations \prec_+, \prec_- can be encoded by the *signed boundary matrices*. The k -th boundary matrix $B_k \in \mathbb{R}^{S_{k-1} \times S_k}$ has entries $B_k(i, j) = 1$ if $\tau_i \prec_+ \sigma_j$, -1 if $\tau_i \prec_- \sigma_j$ and 0, otherwise, where $\dim(\sigma_j) = k$, $\dim(\tau_i) = k - 1$, and S_k denotes the number of simplices of dimension k . Similarly, the boundary relation \prec is encoded by $|B_k|$, the unsigned version of the matrix B_k . The k -th Hodge Laplacian of the simplicial complex (Lim, 2015; Barbarossa & Sardellitti, 2020; Schaub et al., 2020), a diffusion operator for signals defined over the oriented k -simplices is defined as

$$L_k = B_k^\top B_k + B_{k+1} B_{k+1}^\top. \quad (1)$$

We note that L_0 gives the well-known graph Laplacian.

3. Simplicial WL Test

The deep theoretical link between the WL graph isomorphism test and message passing GNNs are well known (Xu et al., 2019b). We exploit this connection to develop a simplicial version of the WL test with the ultimate goal of deriving a message passing procedure that can retain the expressive power of the test. We call this simplicial colouring

algorithm *Simplicial WL* (SWL). We outline below the steps of SWL in the most general sense.

1. Given a complex \mathcal{K} , all the simplices $\sigma \in \mathcal{K}$ are initialised with the same colour.
2. Given the colour c_σ^t of simplex σ at iteration t , we compute the colour of simplex σ at the next iteration c_σ^{t+1} , by perfectly hashing the multi-sets of colours belonging to the adjacent simplices of σ .
3. The algorithm stops once a stable colouring is reached. Two simplicial complexes are considered non-isomorphic if the colour histograms at any level of the complex are different.

A crucial choice has to be made about what simplices are considered to be adjacent in step two. The incidence relation from Definition 2 can be used to construct four types of (local) adjacencies. While all these adjacencies show up in graphs in some form, only one of them is typically used due to the lack of simplices of dimension above one.

Definition 4. Consider a simplex $\sigma \in \mathcal{K}$. Four types of adjacent simplices can be defined:

1. *Boundary adjacencies* $\mathcal{B}(\sigma) = \{\tau \mid \tau \prec \sigma\}$.
2. *Co-boundary adjacencies* $\mathcal{C}(\sigma) = \{\tau \mid \sigma \prec \tau\}$.
3. *Lower-adjacencies* $\mathcal{N}_\downarrow(\sigma) = \{\tau \mid \exists \delta, \delta \prec \tau \wedge \delta \prec \sigma\}$
4. *Upper-adjacencies* $\mathcal{N}_\uparrow(\sigma) = \{\tau \mid \exists \delta, \tau \prec \delta \wedge \sigma \prec \delta\}$

To see how these adjacencies are already present in graphs, we provide a few examples. The boundary simplices of an edge are given by its vertices. The co-boundary simplices of a vertex are given by the edges they are part of. The lower-adjacent edges are given by the common line-graph adjacencies. Finally, upper adjacencies between vertices give the regular graph adjacencies.

Let us use these adjacencies to precisely define the multi-sets of colours used in the update rule of SWL in step two.

Definition 5. Let c^t be a colouring of SWL for the simplices in \mathcal{K} at iteration t . We define the following multi-sets of colours, corresponding to each type of adjacency:

1. $c_{\mathcal{B}}^t(\sigma) = \{\{c_\tau^t \mid \tau \in \mathcal{B}(\sigma)\}\}$.
2. $c_{\mathcal{C}}^t(\sigma) = \{\{c_\tau^t \mid \tau \in \mathcal{C}(\sigma)\}\}$.
3. $c_{\mathcal{N}_\downarrow}^t(\sigma) = \{\{(c_\tau^t, c_{\sigma \cap \tau}^t) \mid \tau \in \mathcal{N}_\downarrow(\sigma)\}\}$
4. $c_{\mathcal{N}_\uparrow}^t(\sigma) = \{\{(c_\tau^t, c_{\sigma \cup \tau}^t) \mid \tau \in \mathcal{N}_\uparrow(\sigma)\}\}$

Note that in the last two types of adjacencies, for adjacent simplices σ and τ , we also include the colour of the common simplex on their boundary (i.e. $\sigma \cap \tau$) and the common simplex they are both on the boundary of (i.e. $\sigma \cup \tau$), respectively.

Finally, we obtain the following update rule, which contains

the complete set of adjacencies:

$$c_\sigma^{t+1} = \text{HASH}\{c_\sigma^t, c_{\mathcal{B}}^t(\sigma), c_{\mathcal{C}}^t(\sigma), c_{\mathcal{N}_\downarrow}^t(\sigma), c_{\mathcal{N}_\uparrow}^t(\sigma)\}.$$

Starting from this formulation, we will now show that certain adjacencies can be removed without sacrificing the expressive power of the SWL test in terms of simplicial complexes that can be distinguished.

Theorem 6. SWL with $\text{HASH}(c_\sigma^t, c_{\mathcal{B}}^t(\sigma), c_{\mathcal{N}_\uparrow}^t(\sigma))$ is as powerful as SWL with the generalised update rule $\text{HASH}(c_\sigma^t, c_{\mathcal{B}}^t(\sigma), c_{\mathcal{C}}^t(\sigma), c_{\mathcal{N}_\downarrow}^t(\sigma), c_{\mathcal{N}_\uparrow}^t(\sigma))$.

We note that other possible combinations of adjacencies might also fully preserve the expressive power of the general SWL test. Additionally, this result comes from a (theoretical) colour-refinement perspective and it does not imply that the pruned adjacencies cannot be useful in practice.

However, this particular choice presents two important properties that make it preferable over other potential ones. First, by not considering lower adjacencies, the test has a computational complexity that is linear in the total number of simplices in the complex. These computational aspects are discussed in more detail in Section 4. Second, when the test is applied to 1-simplicial complexes, i.e. graphs, and only vertex colours are considered, SWL corresponds to the WL test. This is due to the fact that vertices have no boundary simplices and upper adjacencies are the usual graph adjacencies between nodes.

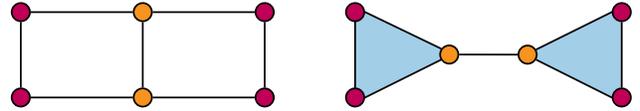


Figure 2. Two graphs that cannot be distinguished by 1-WL, but have distinct clique complexes (the second contains triangles).

Clique Complexes and WL The *clique complex* of a graph G is the simplicial complex \mathcal{K} with the property that if nodes $\{v_0, \dots, v_k\}$ form a clique in G , then simplex $\{v_0, \dots, v_k\} \in \mathcal{K}$. In other words, every $(k+1)$ -clique in G becomes a k -simplex in \mathcal{K} . Evidently, this is an injective transformation from the space of non-isomorphic graphs to the space of non-isomorphic simplicial complexes. Therefore, the SWL procedure can be used to distinguish a pair of non-isomorphic graphs, by comparing their clique complexes. We call this a *lifting transformation*. By taking this pre-processing step, we can link the expressive powers of WL and SWL:

Theorem 7. SWL with a clique complex lifting is strictly more powerful than WL.

We present in Figure 2 a pair of graphs that cannot be distinguished by the WL test, but whose clique complexes can be distinguished by SWL.

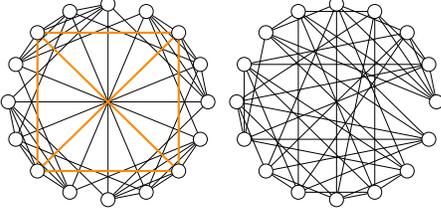


Figure 3. Rook’s 4x4 graph and the Shrikhande graph: Strongly Regular non-isomorphic graphs with parameters $SR(16,6,2,2)$. Our approach can distinguish them: only Rook’s graph (left) possesses 4-cliques (orange) and thus the two graphs are associated with distinct clique complexes. The 3-WL test fails to distinguish them.

By applying SWL to clique-complexes, even harder examples of non-isomorphic graph pairs can be told apart. One such example is the pair of graphs reported in Figure 3. This pair corresponds to the smallest pair of Strongly Regular non-isomorphic graphs with same parameters $SR(16,6,2,2)$. Our approach can distinguish the two graphs while the 3-WL test fails, which is due to the fact that the two are associated with distinct clique complexes. This illustrates that our approach is no less powerful than the 3-WL test. The following theorem confirms this observation.

Theorem 8. *SWL is not less powerful than 3-WL.*

4. Message Passing Simplicial Networks

MPSN We propose a message passing model using the following message passing operations based on the four types of messages discussed in the previous section. For a simplex σ in a complex \mathcal{K} we have:

$$m_{\mathcal{B}}^{t+1}(\sigma) = \text{AGG}_{\tau \in \mathcal{B}(\sigma)} \left(M_{\mathcal{B}}(h_{\sigma}^t, h_{\tau}^t) \right) \quad (2)$$

$$m_{\mathcal{C}}^{t+1}(\sigma) = \text{AGG}_{\tau \in \mathcal{C}(\sigma)} \left(M_{\mathcal{C}}(h_{\sigma}^t, h_{\tau}^t) \right) \quad (3)$$

$$m_{\downarrow}^{t+1}(\sigma) = \text{AGG}_{\tau \in \mathcal{N}_{\downarrow}(\sigma)} \left(M_{\downarrow}(h_{\sigma}^t, h_{\tau}^t, h_{\sigma \cap \tau}^t) \right) \quad (4)$$

$$m_{\uparrow}^{t+1}(\sigma) = \text{AGG}_{\tau \in \mathcal{N}_{\uparrow}(\sigma)} \left(M_{\uparrow}(h_{\sigma}^t, h_{\tau}^t, h_{\sigma \cup \tau}^t) \right). \quad (5)$$

Then, the update operation takes into account these four types of incoming messages and the previous colour of the simplex:

$$h_{\sigma}^{t+1} = U \left(h_{\sigma}^t, m_{\mathcal{B}}^t(\sigma), m_{\mathcal{C}}^t(\sigma), m_{\downarrow}^{t+1}(\sigma), m_{\uparrow}^{t+1}(\sigma) \right). \quad (6)$$

To obtain a global embedding for a p -simplicial complex \mathcal{K} from an MPSN with L layers, the readout function takes as input the multi-sets of colours corresponding to all the dimensions of the complex:

$$h_{\mathcal{K}} = \text{READOUT}(\{\{h_{\sigma}^L\}_{\dim(\sigma)=0}, \dots, \{h_{\sigma}^L\}_{\dim(\sigma)=p}\}).$$

Orientations in Message Passing If the underlying complex also has a particular orientation, the message, aggregate,

update and readout functions can be parametrised to use this information. More specifically, they can make use of the *relative orientations* between adjacent simplices, which can be ± 1 . The relative orientations of the neighbours of the i -th k -simplex are given by the non-zero entries in $B_k(\cdot, i)$ (boundary adjacencies), $B_{k+1}(i, \cdot)$ (co-boundary adjacencies), $B_k^{\top} B_k(i, \cdot)$ (lower adjacencies), and $B_{k+1} B_{k+1}^{\top}(i, \cdot)$ (upper adjacencies). For the last two adjacencies, the matrix multiplications encode how two adjacent simplices are oriented with respect to the lower- or higher-dimensional simplex they share, respectively.

Expressive Power by WL We now describe expressive power of MPSNs in relation to SWL and WL. First, we need to analyse the ability of MPSNs to distinguish non-isomorphic simplicial complexes.

Lemma 9. *MPSNs are at most as powerful as SWL in distinguishing non-isomorphic simplicial complexes.*

One may wonder whether MPSN can achieve the power of SWL. The answer is affirmative:

Theorem 10. *MPSNs with sufficient layers and injective neighbourhood aggregators are as powerful as SWL.*

This theorem, combined with Theorem 7, provides an important corollary showing that MPSNs are not only suitable for statistical tasks on higher-dimensional simplicial complexes, but could potentially improve over standard GNNs on graph machine learning tasks.

Corollary 11. *There exists an MPSN that is more powerful than WL at distinguishing non-isomorphic graphs when using a clique-complex lifting.*

In particular, based on Theorem 6, it is sufficient for such an MPSN to use boundary and upper adjacencies. This result relies on MPSN’s ability of mapping multi-sets of colours injectively and the fact that neural networks that can learn such functions for multi-sets of bounded size exist (Xu et al., 2019b; Corso et al., 2020).

Relation to Spectral Convolutions GNNs are also known for their relationship with spectral convolution operators on graphs obtained via graph Laplacian (Hammond et al., 2011). Analogously to this, we show MPSNs generalise certain spectral convolutions on graphs derived from the higher-order simplicial Hodge Laplacian. The derivation and a more detailed discussion about simplicial spectral convolution are deferred to Appendix C.

Theorem 12. *MPSNs generalise certain spectral convolution operators (Ebli et al., 2020; Bunch et al., 2020) defined over simplicial complexes.*

Equivariance and Invariance We now discuss how MPSNs handle the symmetries present in simplicial complexes.

We offer here a high-level view of how these manifest and postpone a more rigorous treatment of both symmetries for Appendix D.

Generalising GNNs, MPSN layers are equivariant with respect to relabelings of simplices in the complex.

Theorem 13 (Informal). *An MPSN layer is (simplex) permutation equivariant.*

Additionally, if the complex is oriented, from a mathematical point of view, the choice of orientation is irrelevant. Therefore, MPSNs should be equivariant with respect to changes in the orientation of the complex. Intuitively, this amounts to changes in the signature of the boundary matrices B_k and the signature of the features.

Theorem 14 (Informal). *Consider an MPSN layer with a message function that multiplies the features of the neighbours by their relative orientation (i.e. ± 1). If the message, aggregate and update functions are odd (i.e. $f(x) = -f(-x)$), the layer is orientation equivariant.*

To construct an invariant model with respect to these symmetries, it suffices to stack multiple equivariant layers like the ones above and to use an appropriate invariant readout function (more in Appendix D).

Message Passing Complexity A d -simplex σ of a p -complex has $d + 1$ boundary simplices and there are $\binom{d+1}{2}$ upper adjacencies between them. Then, a message passing procedure relying on Theorem 6, which considers only these adjacencies, has a computational complexity $\Theta(\sum_{d=0}^p (d+1)S_d + \binom{d+1}{2}S_d) = \Theta(\sum_{d=0}^p \binom{d+1}{2}S_d)$. If we consider p to be a small constant, which is common for many real-world datasets, then the binomial coefficients can be absorbed in the bound, which results in a linear computational complexity in the size of the complex $\Theta(\sum_{d=0}^p S_d)$. Including co-boundaries does not increase this complexity, but lower adjacencies do. Because any d -complex can be down adjacent to any other d -complex, the worst case complexity is quadratic in the size of the complex $\mathcal{O}(\sum_{d=0}^p S_d^2)$.

Clique Complex Complexity The number of k -cliques in a graph with n nodes is upper-bounded by $\mathcal{O}(n^k)$ and they can be listed in $\mathcal{O}(a(G)^{k-2}m)$ time (Chiba & Nishizeki, 1985), where $a(G)$ is the arboricity of the graph (i.e. a measure of graph sparsity) and m is the number of edges. Since the arboricity can be shown to be at most $\mathcal{O}(m^{1/2})$ and $m \leq n^2$, all k -cliques can be listed in $\mathcal{O}(n^{k-2}m)$. In particular, all triangles can be found in $\mathcal{O}(m^{3/2})$. For certain classes, such as planar graphs, where $a(G) \leq 3$, the complexity becomes linear in the size of the graph. Overall, the fact that the algorithm takes advantage of the sparsity of the graph makes its complexity strictly better than the $\Omega(n^k)$ of all k -GNNs (Morris et al., 2019; 2020b; Maron et al., 2019).

5. Simplicial Networks by Linear Regions

While the WL test has been used for studying the expressive power of GNNs, other tools have been used to study the expressive power of conventional neural networks, like fully connected and convolutional. One such tool is based on the number of linear regions of networks using piece-wise linear activations. This number has been used to draw distinctions between the expressive power of shallow and deep network architectures (Pascanu et al., 2013; Montúfar et al., 2014). It can also be related to the approximation properties of the networks (Telgarsky, 2016) and it has also been considered to shed light into the representational power of (standard) convolutional networks (Xiong et al., 2020). We show how this tool can also be used to approximate the number of linear regions of the functions represented by graph, simplicial, and message passing simplicial networks. We focus on the case where the message function is a linear layer and AGG is sum followed by ReLU. We obtain new results in all cases, showing superior capacity of MPSNs. The details of the proofs of Theorems 15, 16, and 19 are relegated to Appendix B.

GNN We start with the simple case of Graph Neural Networks (GNNs). A graph $G = (V, E, \omega)$ is a set of triplets with vertices $V = \{v_i\}_{i=1}^{S_0}$, edges $E \subseteq V \times V$, and edge weight function $\omega: E \rightarrow \mathbb{R}$. The graph has an adjacency matrix A with the (i, j) th entry $a_{ij} = \omega(v_i, v_j)$. Each node has a d -dimensional feature, and we collect the feature vectors into a matrix $H^{\text{in}} \in \mathbb{R}^{S_0 \times d}$. We consider a GNN convolutional layer of the form

$$H^{\text{out}} = \psi(\mathcal{H}(A, H^{\text{in}})W_0), \quad (7)$$

where ψ is the entry-wise ReLU, $\mathcal{H}(A, H)$ is an aggregation mapping, and $W_0 \in \mathbb{R}^{d \times m}$ are the trainable weights.

Theorem 15 (Number of linear regions of a GNN layer). *Consider a graph G with S_0 nodes, node input features of dimension $d \geq 1$, and node output features of dimension m . Suppose the aggregation function \mathcal{H} as function of H is linear and invertible. Then, the number of linear regions of the functions represented by a ReLU GNN layer (7) has the optimal upper bound*

$$R_{\text{GNN}} = \left(2 \sum_{i=0}^{d-1} \binom{m-1}{i} \right)^{S_0}. \quad (8)$$

This applies to aggregation functions with no trainable parameters including GCN convolution (Kipf & Welling, 2017), spectral GNN (Defferrard et al., 2016; Bruna et al., 2014), and traditional message passing (Gilmer et al., 2017).

The above result should be compared with the optimal upper bound for a standard dense ReLU layer without biases, which for d inputs and m outputs is $2 \sum_{i=0}^{d-1} \binom{m-1}{i}$.

The invertibility condition for the aggregation function \mathcal{H} can be relaxed, but is satisfied by many commonly used graph convolutions: i) For an undirected graph, the normalised adjacency matrix has non-negative eigenvalues. If the eigenvalues are all positive, the aggregation function is invertible. ii) The Fourier transform is the square matrix of eigenvectors, as used in the spectral GNN (Bruna et al., 2014). When the graph Laplacian is non-singular, the Fourier transform matrix is invertible. iii) For the transform Φ by graph wavelet basis, Haar wavelet basis or graph framelets, Φ is invertible in all cases (Xu et al., 2019a; Li et al., 2020; Zheng et al., 2020a;b; 2021; Wang et al., 2020). So the bound in Theorem 15 applies to them.

SCNN Simplicial Complex Neural Networks (SCNNs) were proposed by Ebli et al. (2020). We consider a version of their model using only the first power of a Laplacian matrix, generically denoted here by M_n :

$$H_n^{\text{out}} = \psi(M_n H_n^{\text{in}} W_n), \quad n = 0, \dots, p. \quad (9)$$

In this type of layer, the features on simplices of different dimensions $n = 0, 1, \dots, p$ are computed in parallel.

Theorem 16 (Number of linear regions for an SCNN layer). *Consider a p -dimensional simplicial complex with S_n n -simplices for $n = 0, 1, \dots, p$. Suppose that each M_n is invertible. Then the number of linear regions of the functions represented by a ReLU SCNN layer (9) has the optimal upper bound*

$$R_{\text{SCNN}} = \prod_{n=0}^p \left(2 \sum_{i=0}^{d_n-1} \binom{m_n-1}{i} \right)^{S_n}, \quad (10)$$

where, for each of the n -simplices, d_n is the input feature dimension and m_n is the number of output features.

The product over n in (10) reflects the fact that the features over simplices of different dimensions do not interact. The GNN bound in Theorem 15 is recovered as the special case of the SCNN bound with $p = 0$.

It is instructive to compare the SCNN bound in Theorem 16 with the optimal bound for a dense layer. By Roth’s lemma (Roth, 1934), $\text{vec}(M_n H_n^{\text{in}} W_n) = (W_n^{\text{T}} \otimes M_n) \cdot \text{vec}(H_n^{\text{in}})$, where vec denotes column-by-column vectorization and \otimes the Kronecker product. Hence, for each n , we can regard the SCNN layer as a standard layer $\psi(Ux)$ with weight matrix $U = (W_n^{\text{T}} \otimes M_n) \in \mathbb{R}^{(m_n S_n) \times (S_n d_n)}$ and input vector $x = \text{vec}(H_n^{\text{in}}) \in \mathbb{R}^{S_n d_n}$. Notice that for the SCNN layer, the weight matrix has a specific structure. A standard dense layer with $S_n d_n$ inputs and $m_n S_n$ ReLUs with generic weights and no biases computes functions with $2 \sum_{i=0}^{S_n d_n-1} \binom{m_n S_n-1}{i}$ regions.

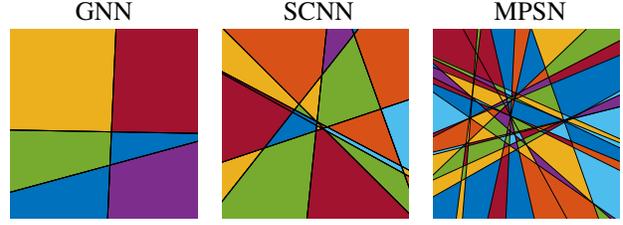


Figure 4. A 2D slice of the input feature spaces of GNN, SCNN, MPSN layers with $S_0 = S_1 = 3, S_2 = 1$ (the complex is a triangle), $d_0 = d_1 = d_2 = 1, m = 3$, colored by linear regions of the represented functions, for a random choice of the weights.

MPSN In our Message Passing Simplicial Network (MPSN), the features on simplices of different dimensions are allowed to interact. For a p -dimensional complex, denote the set of n -simplices by \mathcal{S}_n with $S_n = |\mathcal{S}_n|$. Denote the n -simplex input feature dimension by d_n , and the output feature dimension by $m_n = m, n = 0, \dots, p$. We consider an MPSN layer with linear message functions, sum aggregation for all messages and an update function taking the sum of the messages followed by a ReLU activation. For each dimension n , the output feature matrix H_n^{out} equals:

$$\psi \left(M_n H_n^{\text{in}} W_n + U_n H_{n-1}^{\text{in}} W_{n-1} + O_n H_{n+1}^{\text{in}} W_{n+1} \right), \quad (11)$$

where ψ is an entry-wise activation ($s \mapsto \max\{0, s\}$ for ReLU), $W_n \in \mathbb{R}^{d_n \times m_n}$ are trainable weight matrices and $M_n \in \mathbb{R}^{S_n \times S_n}, U_n \in \mathbb{R}^{S_n \times S_{n-1}},$ and $O_n \in \mathbb{R}^{S_n \times S_{n+1}}$ are some choice of adjacency matrices for the simplicial complex. These could be the Hodge Laplacian matrix L_n and the corresponding boundary matrices B_n^{T}, B_{n+1} , or one of their variants (e.g. normalised).

It is convenient to write the entire layer output in standard form. Using Roth’s lemma and concatenating over n we can write (11) as (details in Appendix B)

$$H^{\text{out}} = \psi(W H^{\text{in}}), \quad (12)$$

where $H^{\text{in}} = \text{vec}([H_0^{\text{in}} | H_1^{\text{in}} | \dots | H_p^{\text{in}}]) \in \mathbb{R}^N, N = \sum_{n=0}^p S_n d_n, H^{\text{out}} = \text{vec}([H_0^{\text{out}} | H_1^{\text{out}} | \dots | H_p^{\text{out}}]) \in \mathbb{R}^M, M = \sum_{n=0}^p S_n m,$ and

$$W = \begin{bmatrix} W_0^{\text{T}} \otimes M_0 & W_1^{\text{T}} \otimes O_0 & & & \\ W_0^{\text{T}} \otimes U_1 & W_1^{\text{T}} \otimes M_1 & W_2^{\text{T}} \otimes O_1 & & \\ & W_1^{\text{T}} \otimes U_2 & W_2^{\text{T}} \otimes M_2 & W_3^{\text{T}} \otimes O_2 & \\ & & & & \ddots \end{bmatrix}. \quad (13)$$

We study the number of linear regions of the function (12) with ReLU based on the matrix $W \in \mathbb{R}^{M \times N}$. For each of the output coordinates $i \in \{1, \dots, M\}$, the ReLU splits the input space \mathbb{R}^N into two regions separated by a hyperplane $\{H^{\text{in}} \in \mathbb{R}^N : W_{i,:} H^{\text{in}} = 0\}$ with normal $W_{i,:}^{\text{T}} \in \mathbb{R}^N$.

In order to count the total number of regions, we will use results from the theory of hyperplane arrangements. Zaslavsky

(1975, Theorem A) shows that the number of regions $r(\mathcal{A})$ defined by an arrangement \mathcal{A} of hyperplanes in \mathbb{R}^N is

$$r(\mathcal{A}) = (-1)^N \chi_{\mathcal{A}}(-1),$$

where $\chi_{\mathcal{A}}$ is the characteristic polynomial of the arrangement. By virtue of a theorem of Whitney (see Stanley 2004, Theorem 2.4 and Orlik & Terao 1992, Lemma 2.55), it can be written as $\chi_{\mathcal{A}}(t) = \sum (-1)^{|\mathcal{B}|} t^{N - \text{rank}(\mathcal{B})}$, where the sum runs over subarrangements $\mathcal{B} \subseteq \mathcal{A}$ that are central (hyperplanes in \mathcal{B} have a nonempty intersection), and $\text{rank}(\mathcal{B})$ denotes the dimension spanned by the normals to the hyperplanes in \mathcal{B} . In our case, \mathcal{A} is a central arrangement with normals given by the rows of the matrix W in (13). Hence:

Lemma 17. *The number of linear regions of the function (12) with $W \in \mathbb{R}^{M \times N}$ and ψ being ReLU is equal to*

$$r(\mathcal{A}) = \sum_{B \subseteq \{1, \dots, M\}} (-1)^{|B| - \text{rank}(W_{B \cdot})},$$

where $W_{B \cdot}$ denotes the submatrix of rows $i \in B$.

This formula counts the linear regions of any particular function represented by our layer. Some interesting cases can be computed explicitly. For instance:

Proposition 18. *Consider some $K \leq N$. If $\text{rank}(W_{B \cdot}) = \min\{|B|, K\}$ for any B , then $r(\mathcal{A}) = 2 \sum_{j=0}^{K-1} \binom{M-1}{j}$.*

We obtain the following bounds.

Theorem 19 (Number of linear regions of an MPSN layer). *With the above settings, the maximum number of linear regions of the functions represented by a ReLU MPSN layer (12) is upper bounded by*

$$R_{\text{MPSN}} \leq \prod_{n=0}^p \left(2 \sum_{i=0}^{d_{n-1} + d_n + d_{n+1} - 1} \binom{m-1}{i} \right)^{S_n},$$

where we set $d_{-1} = d_{p+1} = 0$. We also note the ‘trivial’ upper bound, with $N := \sum_{n=0}^p S_n d_n$ and $M := \sum_{n=0}^p S_n m$,

$$R_{\text{MPSN}} \leq 2 \sum_{j=0}^{N-1} \binom{M-1}{j}.$$

Moreover, if $\text{rank}((O_n)_{C \cdot}) \geq \text{rank}((M_n)_{C \cdot})$ for any selection C of rows and $d_{n+1} \geq d_n$, for $n = 0, \dots, p-1$, then for networks with outputs $H_0^{\text{out}}, \dots, H_{p-1}^{\text{out}}$ we have

$$R_{\text{MPSN}} \geq R_{\text{SCNN}}. \quad (14)$$

By a more careful analysis of the rank conditions it is possible to obtain improvements of these bounds in specific cases, an endeavor that we leave for future work.

We note that the MPSN lower bound (14) surpasses the SCNN upper bound (10). The GNN bound (8) is a special

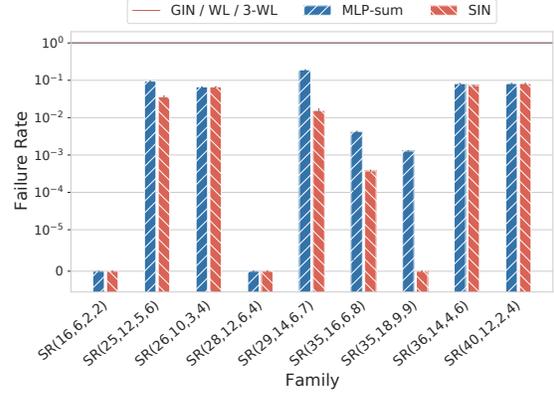


Figure 5. Failure rate on the task of distinguishing SR graphs; log-scale, the smaller the better. GIN fails to distinguish all graph pairs in all families.

case of the SCNN bound (10) with $p = 0$. The regions for the three network architectures are illustrated in Figure 4 for a complex with $S_0 = S_1 = 3$ and $S_2 = 1$, each input dimension 1 and output dimension $m = 3$. It shows that from GNN, SCNN to MPSN the number of linear regions increases in turn, which is consistent with the theory.

MPSN with Populated Higher-Features We also consider a situation of interest where we are given a simplicial complex but only vertex features. To still exploit the structure of the simplicial complex, we can populate the higher features as linear functions of the vertex features. We show this strategy can increase the functional complexity, i.e. $R_{\text{MPSN}} \geq R_{\text{SCNN}}$. See Proposition 32 in Appendix B.

6. Experiments

Strongly Regular Graphs We experimentally validate our theoretical result on the expressive power of our proposed architecture on the task of distinguishing hard pairs of non-isomorphic graphs. In particular, similarly to Bouritsas et al. (2020), we benchmark it on 9 synthetic datasets comprising families of Strongly Regular (SR) graphs. Strongly Regular graphs represent ‘hard’ instances of graph isomorphism, as pairs thereof cannot provably be distinguished by the 3-WL test (we refer readers to Section A.1 for a formal proof). In the experiments, we consider two graphs to be isomorphic if the Euclidean distance between their representations is below a fixed threshold ε . In particular, each graph is lifted to a d -dimensional simplicial complex, with $(d+1)$ the size of the largest clique in the family it belongs to. Lifted graphs are embedded by an untrained MPSN architecture parameterised similarly to GIN (Xu et al., 2019b), which we refer to as Simplicial Isomorphism Network (SIN). Details are included in Appendix F.

Results are illustrated in Figure 5, where we show performance on our isomorphism problem in terms of failure

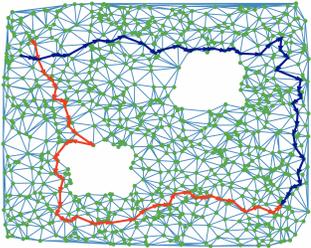


Figure 6. Two samples from the two different classes of trajectories. The two trajectories correspond to approximately orthogonal directions in the space of harmonic functions of L_1 associated with the two holes.

rate, that is the fraction of non-distinguished pairs. The experiment is performed for 10 different random weight-initialisations and we report mean failure rate along with standard error (vertical error bars). We additionally report the performance of an MLP model with sum readout on the same inputs to assess the contribution of message passing to the disambiguation ability. As it can be observed, our (untrained) architecture is able to distinguish the majority of graph pairs in all families, since, in contrast to standard GNNs, it is able to access information related to the presence and number of cliques therein. Additionally, SIN outperforms the strong ‘MLP-sum’ baseline on certain families, showing the favourable inductive bias intrinsic in simplicial message passing.

Edge-Flow Classification We now turn our attention to two applications involving edge flows, which are represented as signals on oriented simplicial complexes. First, inspired by Schaub et al. (2020), we generate a synthetic dataset of trajectories on the simplicial complex in Figure 6, where we treat each triangle as a 2-simplex. All trajectories pass either through the bottom-left or the top-right corner, thus giving rise to two different classes that we aim to distinguish. Due to the two holes present in the complex, the trajectories of the two classes approximately correspond to orthogonal directions in the space of harmonic eigenfunctions of the L_1 Hodge-Laplacian (Schaub et al., 2020). Therefore, we hypothesise, that an orientation invariant MPSN network with orientation equivariant layers should be able to distinguish the two classes. The dataset contains 1000 train trajectories and 200 test trajectories. To make the task more challenging for non orientation invariant models, all the training complexes use the same orientation for the edges, while the test trajectories use random orientations. More details are in Appendix F.

Additionally, we consider a real-world equivalent of the synthetic benchmark above. Again, we adapt a benchmark from Schaub et al. (2020) containing ocean drifter trajectories around the island of Madagascar between years 2011-2018. To obtain a simplicial complex, we discretise the surface of

Table 1. Trajectory classification accuracy. Models with triangle awareness and orientation equivariance generalise better.

| Method | Synthetic Flow | | Ocean Drifters | |
|-----------------|------------------|-----------------|------------------|-----------------|
| | Train | Test | Train | Test |
| GNN L_0 -inv | 63.9±2.4 | 61.0±4.2 | 70.1±2.3 | 63.5±6.0 |
| MPSN L_0 -inv | 88.2±5.1 | 85.3±5.8 | 84.6±4.0 | 71.5±4.1 |
| MPSN - ReLU | 100.0±0.0 | 50.0±0.0 | 100.0±0.0 | 46.5±5.7 |
| MPSN - Id | 88.0±3.1 | 82.6±3.0 | 94.6±0.9 | 73.0±2.7 |
| MPSN - Tanh | 97.9±0.7 | 95.2±1.8 | 99.7±0.5 | 72.5±0.0 |

the map into a simplicial complex containing a hole in the center, representing the island. The task is to distinguish between the clockwise and counter-clockwise flows around the island. As in the previous benchmark, the presence of the hole makes the harmonic signal associated with it extremely important for solving the task. The dataset has 160 train trajectories and 40 test trajectories. As before, the test flows use random orientations for each trajectory to make the task more difficult for non-invariant models.

We evaluate multiple MPSN models with lower and upper adjacencies, each being orientation invariant. The first model, MPSN L_0 -inv, is made invariant from the first layer by simply using the absolute value of the features and ignoring the relative orientations between simplices. Two other MPSN models have equivariant layers like in Theorem 14 and use odd activation functions (Id and Tanh). They end with a final permutation invariant readout layer. The last MPSN model is similar to the previous two, but uses a ReLU activation function, which breaks its invariance. We also consider a GNN baseline operating in the line graph, unaware of the triangles. Like the first MPSN baseline above, it is made invariant by using absolute values.

Results for both benchmarks over 5 seeds are shown in Table 1. We notice that the GNN’s unawareness of the triangles makes it perform worse, since it cannot extract the harmonic part of the signal. The L_0 -inv model performs well on both benchmarks, but it generally lags behind the models that have equivariant layers (i.e. Id and Tanh) and use a final invariant layer. Among these last two, we see that the non-linear model generally performs better. Finally, we remark the ReLU model perfectly fits the training dataset of both benchmarks, which use a fixed orientation, but does not generalise to the random orientations of the test set.

Real-World Graph Classification Finally, we study the practical impact of considering higher-order interactions via (non-oriented) clique-complexes and report results for a few popular graph classification tasks commonly used for benchmarking GNNs (Morris et al., 2020a). We follow the same experimental setting and evaluation procedure described in Xu et al. (2019b). Accordingly, we report the best mean test accuracy computed in a 10-fold cross-validation fashion. We employ a SIN model similar to

Table 2. Graph classification results on the TUDatasets benchmark. The table contains: dataset details (*top*), graph kernel methods (*middle*), and graph neural networks (*bottom*).

| Dataset | Proteins | NCII | IMDB-B | IMDB-M | RDT-B | RDT-M5K |
|-------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|
| Avg Δ | 27.4 | 0.05 | 392.0 | 305.9 | 24.8 | 21.8 |
| Med Δ | 21.0 | 0.0 | 119.5 | 56.0 | 11.0 | 11.0 |
| RWK | 59.6 \pm 0.1 | >3 days | N/A | N/A | N/A | N/A |
| GK (k=3) | 71.4 \pm 0.31 | 62.5 \pm 0.3 | N/A | N/A | N/A | N/A |
| PK | 73.7 \pm 0.7 | 82.5 \pm 0.5 | N/A | N/A | N/A | N/A |
| WL kernel | 75.0 \pm 3.1 | 86.0 \pm 1.8 | 73.8 \pm 3.9 | 50.9 \pm 3.8 | 81.0 \pm 3.1 | 52.5 \pm 2.1 |
| DCNN | 61.3 \pm 1.6 | 56.6 \pm 1.0 | 49.1 \pm 1.4 | 33.5 \pm 1.4 | N/A | N/A |
| DGCNN | 75.5 \pm 0.9 | 74.4 \pm 0.5 | 70.0 \pm 0.9 | 47.8 \pm 0.9 | N/A | N/A |
| IGN | 76.6 \pm 5.5 | 74.3 \pm 2.7 | 72.0 \pm 5.5 | 48.7 \pm 3.4 | N/A | N/A |
| GIN | 76.2 \pm 2.8 | 82.7 \pm 1.7 | 75.1 \pm 5.1 | 52.3 \pm 2.8 | 92.4 \pm 2.5 | 57.5 \pm 1.5 |
| PPGNs | 77.2 \pm 4.7 | 83.2 \pm 1.1 | 73.0 \pm 5.8 | 50.5 \pm 3.6 | N/A | N/A |
| Natural GN | 71.7 \pm 1.0 | 82.4 \pm 1.3 | 73.5 \pm 2.0 | 51.3 \pm 1.5 | N/A | N/A |
| GSN | 76.6 \pm 5.0 | 83.5 \pm 2.0 | 77.8 \pm 3.3 | 54.3 \pm 3.3 | N/A | N/A |
| SIN (Ours) | 76.5 \pm 3.4 | 82.8 \pm 2.2 | 75.6 \pm 3.2 | 52.5 \pm 3.0 | 92.2 \pm 1.0 | 57.3 \pm 1.6 |

that employed in the SR graph experiments. We lift the original graphs to 2-complexes by considering 3-cliques (triangles) as 2-simplices (see Appendix F for more details). The performance of SIN are reported in Table 2, along with those of graph kernel methods (RWK (Gärtner et al., 2003), GK (Shervashidze et al., 2009), PK (Neumann et al., 2016), WL kernel (Shervashidze et al., 2011)) and other GNNs (DCNN (Atwood & Towsley, 2016), DGCNN (Zhang et al., 2018), IGN (Maron et al., 2018), GIN (Xu et al., 2019b), PPGNs (Maron et al., 2019), Natural GN (de Haan et al., 2020), GSN (Bouritsas et al., 2020)). We observe that our model achieves its best results on the IMDB datasets, which have the largest mean and median number of triangles. In contrast, on datasets like NCII, where the number of higher-order structures is close to zero, the model shows the same mean accuracy as GIN. Overall, we observe SIN to perform on-par with other GNN approaches.

7. Discussion and Conclusion

Provably Powerful GNNs In order to overcome the limited expressive power of standard GNN architectures, several works have proposed variants inspired by the higher-order k -WL procedures (see Appendix). Maron et al. (2019) introduced a model equivalent in power to 3-WL, Morris et al. (2019) proposed k -GNNs, graph neural networks equivalents of set-based k -WL tests. By performing message passing on all possible k -tuples of nodes and across non-local neighborhoods, these models trade locality of computation for expressive power, and thus suffer from high spatial and temporal complexities. Local k -WL variants were introduced by Morris et al. (2020b), and distinguish local and global neighbors. The authors also propose provably powerful neural counterparts. Although more efficient, in contrast to our method, this approach still accounts for all possible node k -tuples in a graph. An alternative approach to improving GNN expressivity has been adopted in (Bouritsas et al., 2020), where isomorphism counting of graph substructures is employed as a symmetry breaking mechanism to disambiguate neighbours. Similarly to

ours, this approach retains locality of operations; however, message passing is only performed at the node level.

Beyond Pairwise Interactions We note that other graph lifting transformations could also be used for applying SWL and MPSNs to graph domains. While clique complexes are the commonest such transformation, many others exist (Ferri et al., 2018) and they could be used to emphasise motifs that are relevant for the task (Milo et al., 2002). More broadly, the transformation could target a much wider output space such as cubical complexes (Kaczynski et al., 2004) (see Appendix E), or cell complexes (Hatcher, 2000), for which a message passing procedure has already been proposed (Hajjij et al., 2020). One can study even more flexible structures described by incidence tensors (Albooyeh et al., 2020), which can also encode simplicial complexes. Alternatively, if the goal is to learn higher-order representations, one could also do so in an unsupervised manner directly from pairwise interactions (Cotta et al., 2020) or by explicitly modelling subgraphs (Alsentzer et al., 2020). We conclude this part by mentioning a large body of work on neural networks for hypergraphs, which subsume simplicial complexes (Yadati et al., 2019; Feng et al., 2019; Zhang et al., 2020; Yadati, 2020). However, none of these study their expressive power in relationship to the WL hierarchy.

Conclusion We introduce a provably powerful message passing procedure for simplicial complexes relying on local higher-order interactions. We motivate our message passing framework by the introduction of SWL, a colouring algorithm for simplicial complex isomorphism testing, generalising the WL test for graphs. We prove that when graphs are lifted in the simplicial complex space via their clique complex, SWL and MPSNs are more expressive than the WL test. We also analyse MPSNs through the lens of symmetry by showing they are simplex permutation equivariant and can be made orientation equivariant. Furthermore, we produce an estimate for the number of linear regions of GNNs, SCNNs and MPSNs, which also reveals the superior expressive power of our model. We empirically confirm these results by distinguishing between challenging non-isomorphic SR graphs, on real-world graph classification benchmarks, on edge flow classification tasks, and by computing 2D slices through the linear regions of the models.

Acknowledgements

YW and GM acknowledge support from the ERC under the EU’s Horizon 2020 programme (grant agreement n° 757983). MB is supported in part by ERC Consolidator grant n° 724228 (LEMAN). We would also like to thank Teodora Stoleru for creating Figure 1 and Ben Day for providing valuable comments on an early draft. We are also grateful to the anonymous reviewers for their feedback.

References

- Albooyeh, M., Bertolini, D., and Ravanbakhsh, S. Incidence networks for geometric deep learning. *arXiv preprint arXiv:1905.11460*, 2020.
- Alsentzer, E., Finlayson, S., Li, M., and Zitnik, M. Subgraph neural networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 8017–8029. Curran Associates, Inc., 2020.
- Atwood, J. and Towsley, D. Diffusion-convolutional neural networks. In *NIPS*, pp. 1993–2001, 2016.
- Baik, K. H. and Miller, L. L. Topological approach for testing equivalence in heterogeneous relational databases. *The Computer Journal*, 33, 1990.
- Barbarossa, S. and Sardellitti, S. Topological signal processing over simplicial complexes. *IEEE Transactions on Signal Processing*, 68:2992–3007, 2020.
- Battiston, F., Cencetti, G., Iacopini, I., Latora, V., Lucas, M., Patania, A., Young, J.-G., and Petri, G. Networks beyond pairwise interactions: structure and dynamics. *Physics Reports*, 2020.
- Bodnar, C., Cangea, C., and Liò, P. Deep graph mapper: Seeing graphs through the neural lens. *arXiv preprint arXiv:2002.03864*, 2020.
- Boissonnat, J.-D. and Maria, C. The simplex tree: An efficient data structure for general simplicial complexes. *Algorithmica*, 70:406–427, 2014.
- Bouritsas, G., Frasca, F., Zafeiriou, S., and Bronstein, M. M. Improving graph neural network expressivity via subgraph isomorphism counting. *arXiv:2006.09252*, 2020.
- Bruna, J., Zaremba, W., Szlam, A., and Lecun, Y. Spectral networks and locally connected networks on graphs. In *ICLR*, 2014.
- Bunch, E., You, Q., Fung, G., and Singh, V. Simplicial 2-complex convolutional neural networks. In *NeurIPS Workshop on Topological Data Analysis and Beyond*, 2020.
- Cai, J.-Y., Fürer, M., and Immerman, N. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992.
- Cangea, C., Veličković, P., Jovanović, N., Kipf, T., and Liò, P. Towards sparse hierarchical graph classifiers. In *Proc. Workshop on Relational Representation Learning*, 2018.
- Chen, Z., Chen, L., Villar, S., and Bruna, J. Can graph neural networks count substructures? In *NeurIPS*, 2020.
- Chiba, N. and Nishizeki, T. Arboricity and subgraph listing algorithms. *SIAM J. Comput.*, 14:210–223, 1985.
- Clevert, D.-A., Unterthiner, T., and Hochreiter, S. Fast and accurate deep network learning by exponential linear units (ELUs). In *ICLR*, 2016.
- Corso, G., Cavalleri, L., Beaini, D., Liò, P., and Veličković, P. Principal neighbourhood aggregation for graph nets. In *NeurIPS*, volume 33, pp. 13260–13271, 2020.
- Cotta, L., H. C. Teixeira, C., Swami, A., and Ribeiro, B. Unsupervised joint k-node graph representations with compositional energy-based models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 17536–17547. Curran Associates, Inc., 2020.
- Crane, K., de Goes, F., Desbrun, M., and Schröder, P. Digital geometry processing with discrete exterior calculus. In *ACM SIGGRAPH 2013 courses*, SIGGRAPH ’13, New York, NY, USA, 2013. ACM.
- de Haan, P., Cohen, T., and Welling, M. Natural graph networks. In *NeurIPS*, 2020.
- Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, 2016.
- Ebli, S., Defferrard, M., and Spreemann, G. Simplicial neural networks. In *NeurIPS Workshop on Topological Data Analysis and Beyond*, 2020.
- Feng, Y., You, H., Zhang, Z., Ji, R., and Gao, Y. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3558–3565, 2019.
- Ferri, M., Bergomi, D. M. G., and Zu, L. Simplicial complexes from graphs towards graph persistence. *arXiv:1805.10716*, 2018.
- Fey, M. and Lenssen, J. E. Fast graph representation learning with pytorch geometric. In *Proc. Workshop on Representation Learning on Graphs and Manifolds*, 2019.
- Gärtner, T., Flach, P., and Wrobel, S. On graph kernels: Hardness results and efficient alternatives. In *Learning theory and kernel machines*, pp. 129–143. Springer, 2003.
- Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *ICML*, pp. 1263–1272, 2017.
- Glaze, N., Roddenberry, T. M., and Segarra, S. Principled simplicial neural networks for trajectory prediction. In *ICML*, 2021.

- Goller, C. and Kuchler, A. Learning task-dependent distributed representations by backpropagation through structure. In *ICNN*, 1996.
- Gori, M., Monfardini, G., and Scarselli, F. A new model for learning in graph domains. In *ICNN*, 2005.
- Grünbaum, B. *Arrangements of Hyperplanes*, pp. 432–454. Springer New York, New York, NY, 2003. ISBN 978-1-4613-0019-9. doi: 10.1007/978-1-4613-0019-9_18. URL https://doi.org/10.1007/978-1-4613-0019-9_18.
- Hajij, M., Istvan, K., and Zamzmi, G. Cell complex neural networks. In *NeurIPS Workshop on Topological Data Analysis and Beyond*, 2020.
- Hammond, D. K., Vandergheynst, P., and Gribonval, R. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.
- Hatcher, A. *Algebraic topology*. Cambridge Univ. Press, Cambridge, 2000.
- Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.
- Kaczynski, T., Mischaikow, K., and Mrozek, M. *Computational Homology*. Springer-Verlag New York, 2004.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.
- Li, M., Ma, Z., Wang, Y. G., and Zhuang, X. Fast Haar transforms for graph neural networks. *Neural Networks*, pp. 188–198, 2020.
- Li, Y., Tarlow, D., Brockschmidt, M., and Zemel, R. Gated graph sequence neural networks. *arXiv:1511.05493*, 2015.
- Lim, L.-H. Hodge laplacians on graphs. *arXiv preprint arXiv:1507.05379*, 2015.
- Maron, H., Ben-Hamu, H., Shamir, N., and Lipman, Y. Invariant and equivariant graph networks. In *ICLR*, 2018.
- Maron, H., Ben-Hamu, H., Serviansky, H., and Lipman, Y. Provably powerful graph networks. In *NeurIPS*, 2019.
- Matsaglia, G. and Styan, G. P. H. Equalities and inequalities for ranks of matrices. *Linear and Multilinear Algebra*, 2(3):269–292, 1974. doi: 10.1080/03081087408817070.
- Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.
- Montúfar, G. F., Pascanu, R., Cho, K., and Bengio, Y. On the number of linear regions of deep neural networks. In *NIPS*, 2014.
- Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. Weisfeiler and Leman go neural: Higher-order graph neural networks. In *AAAI*, 2019.
- Morris, C., Kriege, N. M., Bause, F., Kersting, K., Mutzel, P., and Neumann, M. TUDataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020a.
- Morris, C., Rattan, G., and Mutzel, P. Weisfeiler and Leman go sparse: Towards scalable higher-order graph embeddings. In *NeurIPS*, 2020b.
- Nanda, V. Computational algebraic topology lecture notes, March 2021.
- Neumann, M., Garnett, R., Bauckhage, C., and Kersting, K. Propagation kernels: efficient graph kernels from propagated information. *Machine Learning*, 102(2):209–245, 2016.
- Orlik, P. and Terao, H. *Arrangements of Hyperplanes*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 1992.
- Pascanu, R., Montúfar, G., and Bengio, Y. On the number of response regions of deep feed forward networks with piece-wise linear activations. In *ICLR*, 2013.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pp. 8024–8035. Curran Associates, Inc., 2019.
- Rosenberg, S. *The Laplacian on a Riemannian Manifold: An Introduction to Analysis on Manifolds*. London Mathematical Society Student Texts. Cambridge University Press, 1997.
- Roth, W. E. On direct product matrices. *Bulletin of the American Mathematical Society*, 40(6):461–468, 06 1934.
- Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

- Schaub, M. T., Benson, A. R., Horn, P., Lippner, G., and Jadbabaie, A. Random walks on simplicial complexes and the normalized Hodge 1-Laplacian. *SIAM Review*, 62(2):353–391, 2020.
- Shervashidze, N., Vishwanathan, S., Petri, T., Mehlhorn, K., and Borgwardt, K. Efficient graphlet kernels for large graph comparison. In *Artificial intelligence and statistics*, pp. 488–495. PMLR, 2009.
- Shervashidze, N., Schweitzer, P., Leeuwen, E. J. v., Mehlhorn, K., and Borgwardt, K. M. Weisfeiler-lehman graph kernels. *JMLR*, 12(Sep):2539–2561, 2011.
- Sperduti, A. Encoding labeled graphs by labeling raam. In *NIPS*, 1994.
- Stanley, R. P. An introduction to hyperplane arrangements. In *Lecture notes, IAS/Park City Mathematics Institute*, 2004.
- Telgarsky, M. Benefits of depth in neural networks. In *COLT*, 2016.
- The GUDHI Project. *GUDHI User and Reference Manual*. GUDHI Editorial Board, 3.4.1 edition, 2021.
- Wagner, H., Chen, C., and Vucini, E. Efficient computation of persistent homology for cubical data. In *Topological Methods in Data Analysis and Visualization II. Mathematics and Visualization*. Springer, Berlin, Heidelberg, 2011.
- Wang, Y. G., Li, M., Ma, Z., Montúfar, G., Zhuang, X., and Fan, Y. Haar graph pooling. In *ICML*, 2020.
- Weisfeiler, B. and Leman, A. The reduction of a graph to canonical form and the algebra which appears therein. *NTI Series*, 2(9):12–16, 1968.
- Xiong, H., Huang, L., Yu, M., Liu, L., Zhu, F., and Shao, L. On the number of linear regions of convolutional neural networks. In *ICML*, 2020.
- Xu, B., Shen, H., Cao, Q., Qiu, Y., and Cheng, X. Graph wavelet neural network. In *ICLR*, 2019a.
- Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.-i., and Jegelka, S. Representation learning on graphs with jumping knowledge networks. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5453–5462. PMLR, 10–15 Jul 2018.
- Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *ICLR*, 2019b.
- Yadati, N. Neural message passing for multi-relational ordered and recursive hypergraphs. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3275–3289. Curran Associates, Inc., 2020.
- Yadati, N., Nimishakavi, M., Yadav, P., Nitin, V., Louis, A., and Talukdar, P. Hypergcn: A new method for training graph convolutional networks on hypergraphs. In Walsch, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- Zaslavsky, T. *Facing up to Arrangements: Face-Count Formulas for Partitions of Space by Hyperplanes: Face-count Formulas for Partitions of Space by Hyperplanes*. Memoirs of the American Mathematical Society. American Mathematical Society, 1975.
- Zhang, M., Cui, Z., Neumann, M., and Chen, Y. An end-to-end deep learning architecture for graph classification. In *AAAI*, 2018.
- Zhang, R., Zou, Y., and Ma, J. Hyper-saggn: a self-attention based graph neural network for hypergraphs. In *International Conference on Learning Representations*, 2020.
- Zheng, X., Zhou, B., Li, M., Wang, Y. G., and Gao, J. MathNet: Haar-like wavelet multiresolution-analysis for graph representation and learning. *arXiv:2007.11202*, 2020a.
- Zheng, X., Zhou, B., Wang, Y. G., and Zhuang, X. Decimated framelet system on graphs and fast G-framelet transforms. *arXiv:2012.06922*, 2020b.
- Zheng, X., Zhou, B., Gao, J., Wang, Y. G., Liò, P., Li, M., and Montúfar, G. How framelets enhance graph neural networks. *arXiv preprint arXiv:2102.06986*, 2021.