

Weisfeiler and Lehman Go Cellular: CW Networks

Cristian Bodnar*
 University of Cambridge
cb2015@cam.ac.uk

Fabrizio Frasca*
 Imperial College London & Twitter
ffrasca@twitter.com

Nina Otter
 UCLA
otter@math.ucla.edu

Yu Guang Wang
 MPI-MIS, SJTU & UNSW
yuguang.wang@unsw.edu.au

Pietro Liò
 University of Cambridge
pl219@cam.ac.uk

Guido Montúfar
 MPI-MIS & UCLA
montufar@math.ucla.edu

Michael Bronstein
 Imperial College London & Twitter
mbronstein@twitter.com

Abstract

Graph Neural Networks (GNNs) are limited in their expressive power, struggle with long-range interactions and lack a principled way to model higher-order structures. These problems can be attributed to the strong coupling between the computational graph and the input graph structure. The recently proposed Message Passing Simplicial Networks naturally decouple these elements by performing message passing on the clique complex of the graph. Nevertheless, these models are severely constrained by the rigid combinatorial structure of Simplicial Complexes (SCs). In this work, we extend recent theoretical results on SCs to regular Cell Complexes, topological objects that flexibly subsume SCs and graphs. We show that this generalisation provides a powerful set of graph “lifting” transformations, each leading to a unique hierarchical message passing procedure. The resulting methods, which we collectively call CW Networks (CWNs), are strictly more powerful than the WL test and, in certain cases, not less powerful than the 3-WL test. In particular, we demonstrate the effectiveness of one such scheme, based on rings, when applied to molecular graph problems. The proposed architecture benefits from provably larger expressivity than commonly used GNNs, principled modelling of higher-order signals and from compressing the distances between nodes. We demonstrate that our model achieves state-of-the-art results on a variety of molecular datasets.

1 Introduction

The operations performed by message passing Graph Neural Networks (GNNs) emulate the structure of the input graph. While this property has clear computational advantages, it brings with it a series of fundamental limitations. As observed by Xu et al. [72] and Morris et al. [53] the local neighbourhood aggregations used by GNNs are at most as powerful as the Weisfeiler-Lehman (WL) test [69] in distinguishing non-isomorphic graphs. Therefore, GNNs fail to detect certain higher-order meso-scale structures such as cliques or (induced) cycles [2, 14], which are particularly important in applications dealing with social and biological networks or molecular graphs. At the same time, many such layers have to be stacked to make long-range interactions in the graph possible. Besides the computational burden this incurs, deep GNNs typically come with additional problems such as over-smoothing [49] and over-squashing [1] of the node representations.

* Authors contributed equally.

To address these problems, we propose a novel message passing procedure based on (regular) cell complexes, also known as CW complexes², topological objects that form the building block of algebraic topology [36]. When paired with a theoretically-justified “lifting” transformation augmenting the graph with higher-dimensional constructs called “cells”, our method results in a multi-dimensional and hierarchical message passing procedure over the input graph. Our approach generalises and subsumes the recently proposed Message Passing Simplicial Networks (MPSNs) [7], which operate on simplicial complexes (SCs), topological generalisations of graphs. However, SCs have a rigid combinatorial structure that significantly limits the range of lifting transformations one could use to meaningfully modulate the message passing procedure. In contrast, we show that cell complexes, which in turn generalise simplicial complexes and come with additional flexibility, allow one to construct new and better ways of decoupling the input and computational graphs.

Main Contributions To summarise, we propose a message passing scheme operating on regular cell complexes. We call this family of models CW Networks (CWNs) and study their expressive power using a cellular version of the WL test. We show that for an entire class of “lifting” transformations CWNs are at least as powerful as the WL test. Furthermore, we prove that for some of the maps in this class, CWNs can be strictly more powerful than WL, Simplicial WL and also not less powerful than 3-WL. We also express the fundamental symmetries of these models and show how they can be seen as generalised convolutional operators on cell complexes. Experimentally, we focus our attention on a particular “lifting” map based on induced cycles. When applied to molecular graphs, it leads to an intuitive hierarchical message passing procedure involving atoms, the bonds between them and the chemical rings of the molecules. We demonstrate that this provably powerful approach obtains state-of-the-art results on popular large-scale molecular graph datasets and other related tasks.

2 Background

cells \times_0

Definition 1 (Hansen and Ghrist [34]). A **regular cell complex** is a topological space X together with a partition $\{X_\sigma\}_{\sigma \in P_X}$ of subspaces X_σ of X called **cells**, and such that

1. For each $x \in X$ there exists an open neighborhood of x that intersects finitely many cells.
2. Let $\overline{X_\sigma}$ be the closure of a cell. For all σ, τ we have that $X_\tau \cap \overline{X_\sigma} \neq \emptyset$ iff $X_\tau \subseteq \overline{X_\sigma}$.
3. Every cell is homeomorphic to \mathbb{R}^n for some n .
4. (Regularity) For every $\sigma \in P_X$ there is a homeomorphism ϕ of a closed ball in \mathbb{R}^{n_σ} to $\overline{X_\sigma}$ such that the restriction of ϕ to the interior of the ball is a homeomorphism onto X_σ .

partially ordered set
We note that by condition (2) the indexing set P_X has a poset structure $\tau \leq \sigma \Leftrightarrow X_\tau \subseteq \overline{X_\sigma}$, while condition (4) guarantees that this poset structure encodes all the topological information about X . Thus, we can identify a regular cell complex X with this poset, called **face poset** of X . We also use $\tau < \sigma$ for the strict version of this partial order.

Intuitively, one constructs a cell complex through a hierarchical gluing procedure. One starts with a set of vertices (0-cells). Then edges (1-cells) are attached to these by gluing the endpoints of closed line segments to them. We have now only described a (multi) graph. However, one can generalise this even further by taking a two-dimensional closed disk and glue its boundary (i.e. a circle) to any simple cycle in the (multi) graph previously built as in Figure 2. While we are generally not concerned with dimensions above two, this can be further generalised by gluing the boundary of n -dimensional balls to certain $(n-1)$ -cells in the complex.

Consider the examples in Figure 1. The shown sphere is a cell complex obtained from two 0-cells (i.e. vertices), to which two 1-cells (i.e. edges), which form the equator, were attached. The boundary of two 2-dimensional disks (i.e. the two hemispheres) were glued to the equator to form a sphere. The second example is a tetrahedron with empty interior. It is a particular type of cell complex called

*example for what we can do with CW complexes
but cannot with simplicial complexes?*

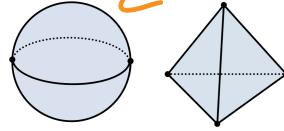


Figure 1: Examples of regular cell complexes: a sphere and an empty tetrahedron. The latter is also a simplicial complex.

²We use these terms interchangeably. For the latter, the C stands for “closure-finite”, and the W for “weak” topology. The term was coined by Whitehead [70].

a simplicial complex (SC). The only 2-cells it allows are triangles. More generally, the n -dimensional cells of SCs are n -simplices. This makes them more rigid structures.

Definition 2. The **k -skeleton** of a cell complex X , denoted $X^{(k)}$, is the subcomplex of X consisting of cells of dimension at most k .

This definition is useful for referring to certain parts of the complex. For instance, $X^{(0)}$ contains the vertices in the complex, while $X^{(1)}$ contains the vertices *and* the edges (i.e. the underlying graph).

The combinatorial structure of the complex can be more compactly described by an incidence relation called the *boundary relation*, whose reflexive and transitive closure gives the partial order defined above. The boundary relation describes what cells are on the boundary of other cells. For instance, the edges of the sphere in Figure 1 are on the boundary of the 2 cells forming the two hemispheres.

Definition 3. We have the **boundary relation** $\sigma \prec \tau$ iff $\sigma < \tau$ and there is no cell δ such that $\sigma < \delta < \tau$.

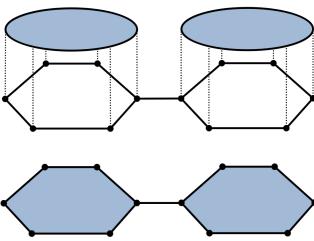
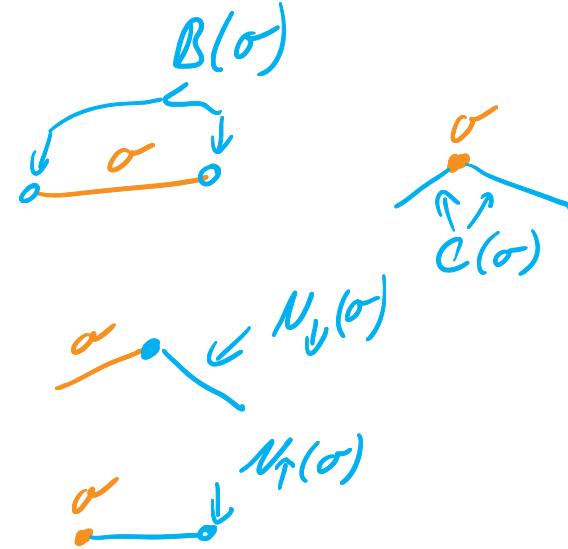


Figure 2: Closed 2D disks are glued to the boundary of the rings present in the graph (top). The result is a 2D regular cell complex (bottom).

We can use this to define the four types of (local) adjacencies present in cell complexes. These adjacencies will be the fundamental building block of our message passing procedure. To explain these in more familiar terms, for each adjacency, we exemplify how it shows up in graphs.

Definition 4 (Cell complex adjacencies). For a cell complex X and a cell $\sigma \in P_X$, we define:

1. The boundary adjacent cells $\mathcal{B}(\sigma) = \{\tau \mid \tau \prec \sigma\}$. These are the lower-dimensional cells on the boundary of σ . For instance, the boundary cells of an edge are its vertices.
2. The co-boundary adjacent cell $\mathcal{C}(\sigma) = \{\tau \mid \sigma \prec \tau\}$. These are the higher-dimensional cells with σ on their boundary. For instance, the co-boundary cells of a vertex are the edges it is part of.
3. The lower adjacent cells $\mathcal{N}_\downarrow(\sigma) = \{\tau \mid \exists \delta \text{ such that } \delta \prec \sigma \text{ and } \delta \prec \tau\}$. These are the cells of the same dimension as σ that share a lower dimensional cell on their boundary. The line graph adjacencies between the edges are a classic example of this.
4. The upper adjacent cells $\mathcal{N}_\uparrow(\sigma) = \{\tau \mid \exists \delta \text{ such that } \sigma \prec \delta \text{ and } \tau \prec \delta\}$. These are the cells of the same dimension as σ that are on the boundary of the same higher-dimensional cell as σ . The typical graph adjacencies between vertices are the canonical example here.



3 Cellular Weisfeiler Lehman

Overview The results in this section show how one can transform graphs into higher-dimensional cell complexes in such a way that performing colour refinement on the resulting cell complexes makes it easier to test their isomorphism. The message passing model from Section 4 will take advantage of these theoretical results. All proofs can be found in Appendix A.

Definition 5. Let c be a colouring of the cells in a complex X with c_σ denoting the colour assigned to cell $\sigma \in P_X$. Define $\mathcal{B}(\sigma, \tau) := \mathcal{B}(\sigma) \cap \mathcal{B}(\tau)$ and $\mathcal{C}(\sigma, \tau) := \mathcal{C}(\sigma) \cap \mathcal{C}(\tau)$. We define the following multi-sets of colours:

1. The colours of the boundary cells of σ : $c_{\mathcal{B}}(\sigma) = \{c_\tau \mid \tau \in \mathcal{B}(\sigma)\}$. *take all colours of boundary into the set (vertices of edge)*
2. The colours of the co-boundary cells of σ : $c_{\mathcal{C}}(\sigma) = \{c_\tau \mid \tau \in \mathcal{C}(\sigma)\}$. *all colors of co-boundary (edges of vertex)*
3. The lower adjacent colours of σ : $c_\downarrow(\sigma) = \{(c_\tau, c_\delta) \mid \tau \in \mathcal{N}_\downarrow(\sigma) \text{ and } \delta \in \mathcal{B}(\sigma, \tau)\}$. *connected by vertices*
4. The upper adjacent colours of σ : $c_\uparrow(\sigma) = \{(c_\tau, c_\delta) \mid \tau \in \mathcal{N}_\uparrow(\sigma) \text{ and } \delta \in \mathcal{C}(\sigma, \tau)\}$. *connected by edges*

Note that unlike in graphs and simplicial complexes, the sets $\mathcal{B}(\sigma, \tau)$ and $\mathcal{C}(\sigma, \tau)$ can have more than one element. For instance, two (closed) 2-cells might intersect in more than one edge (e.g. the two hemispheres in Figure 1), and conversely, two edges might be on the boundary of the same two 2-cells. This illustrates the more flexible combinatorial structure of cell complexes.

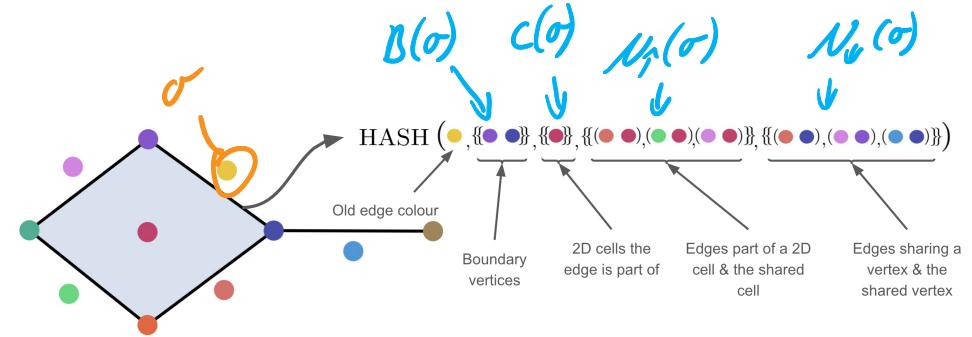


Figure 3: The CWL colouring procedure for the yellow edge of the cell complex. All cells have been assigned unique colours to aid the visualisation of the adjacencies. Note that the yellow edge aggregates long-range information from the light green edge.

Cellular WL (CWL) We consider CWL, a colour refinement scheme for cell complexes that generalises the Simplicial WL [7] and WL [69] tests. We use c_σ^t to refer to the colour assigned by CWL to cell σ at iteration t of the algorithm. When the input is a simplicial complex, this recovers the SWL algorithm. A step of the algorithm is graphically depicted in Figure 3 for a single cell.

1. Given a regular cell complex X , all the cells σ are initialised with the same colour.
 2. Given the colour c_σ^t of cell σ at iteration t , we compute the colour of cell σ at the next iteration c_σ^{t+1} by injectively mapping the multi-sets of colours belonging to the adjacent cells of σ using a fixed HASH function:
- $$c_\sigma^{t+1} = \text{HASH}(c_\sigma^t, c_B^t(\sigma), c_C^t(\sigma), c_D^t(\sigma), c_U^t(\sigma)).$$
3. The algorithm stops when a stable colouring is reached. Two cell complexes are considered non-isomorphic if their colour histograms are different. Otherwise, the test is inconclusive.

First, we state the following theorem from Bodnar et al. [7] involving SWL and simplicial complexes. This theorem shows that on simplicial complexes, certain adjacencies can be pruned without affecting the non-isomorphic SCs that can be distinguished. This has important computational implications.

Theorem 6. *SWL without coboundary and lower-adjacencies has the same expressive power in distinguishing non-isomorphic simplicial complexes as SWL with the complete set of adjacencies.*

It is not immediately clear whether an equivalent theorem would also hold for cell complexes. This is because cells, unlike simplices, can have widely different shapes and, as described above, the adjacencies between them take more complicated forms. Nevertheless, we show that a positive result can be obtained.

Theorem 7. *CWL without coboundary and lower-adjacencies has the same expressive power in distinguishing non-isomorphic cell complexes as CWL with the complete set of adjacencies.*

We note this does not mean that the removed adjacencies are completely redundant in practice. Even if they are not needed from a (theoretical) colour refinement perspective, they might still include important inductive biases that make them suitable for certain tasks.

We are now interested in examining various procedures for mapping, or “lifting”, graphs into the space of regular cell complexes. Such a procedure can be used to test the isomorphism of two graphs by performing colour refinement on the cell complexes they are mapped to. The hope is that CWL applied to these cell complexes is more powerful than WL applied to the initial graphs. We will later show that for a wide range of transformations, this is indeed the case. We start by rigorously defining what we mean by a “lifting”.

Definition 8. A *cellular lifting map* is a function $f : \mathcal{G} \rightarrow \mathcal{X}$ from the space of graphs \mathcal{G} to the space of regular cell complexes \mathcal{X} with the property that two graphs G_1, G_2 are isomorphic iff the cell complexes $f(G_1), f(G_2)$ are isomorphic.

This property ensures that testing the isomorphism of the two cell complexes is equivalent to testing the isomorphism in the input graphs. This would not be the case if two non-isomorphic graphs were mapped to the same cell complex.

Example 9. It can be verified that the function mapping each graph to its clique complex (i.e. every $(k+1)$ -clique in the graph becomes a k -simplex) is a cellular lifting map.

With only 1-dim cells and upper adjacency we recover the usual WL-test

With this information would only travel “upwards” to the faces?



it is enough to only use boundary and upper adjacency for distinguishing isomorphic graphs

But!

lower adjacent features and co-boundary features might still contain important information

turn graph into complex is called “lifting”

That is what we already do in the simplicial complexes paper.

But now we want MORE!



The clique complex lifting map from Example 9 has been used by Bodnar et al. [7] to show that SWL is strictly more powerful than WL. We restate this result:

Theorem 10. SWL with clique complex lifting is strictly more powerful than WL.

A natural question is what other lifting transformations make CWL strictly more powerful than WL? We now describe a space of lifting transformations that make CWL at least as powerful as WL.

Definition 11. A lifting map is **skeleton-preserving** if for any graph G , the 1-skeleton of $f(G)$ and G are isomorphic as (multi) graphs.

Intuitively, skeleton-preserving liftings ensure that the additional structure added by the lifting map comes from attaching cells of dimension at least two to the graph. These mappings keep the 0-cells and 1-cells intact and are, therefore, restricted from making modifications to the input graph structure. An important remark is that for simplicial complexes, attaching simplices based on cliques present in the graph is the only possible skeleton preserving transformation. Once again, this illustrates the limitations of simplicial complexes for adding useful higher-dimensional structures to the graph.

Example 12. The function from Example 9 is also skeleton-preserving because the 1-skeleton of the clique complex of a graph is trivially isomorphic to the graph. A lifting function mapping each graph to a multi-graph where each edge is doubled by a parallel edge is not skeleton-preserving (Figure 4).

We now show that all the maps in the skeleton-preserving class have the following desirable property:

Theorem 13. Let f be a skeleton-preserving lifting map. Then CWL using lifting f is at least as powerful as WL in distinguishing non-isomorphic graphs.

To prove that some of these make CWL strictly more powerful than WL, it is sufficient to find a pair of graphs that cannot be distinguished by WL, but can be distinguished by CWL. The following corollary gives examples of such maps.

Corollary 14. The following lifting transformations make CWL strictly more powerful than the WL test. (1) The clique complex lifting. (2) The map that attaches 2D cells to all the simple cycles in the graph. (3) The map that attaches 2D cells to all the induced cycles in the graph. (4) The union of all the transformations above.

We note that this is not a complete list. For instance, lifting maps may be constructed based on (induced) subgraphs that the WL test cannot count [2, 14]. We leave this endeavour for future work. Additionally, it is easy to extend the result above to lifting maps attaching cells only to rings / cycles of some fixed size $k \geq 3$.

We can also relate CWL with these lifting transformations to the higher-order 3-WL test.

Theorem 15. CWL with the lifting maps from Corollary 14 is not less powerful than 3-WL.

Finally, we conclude this section by showing how CWL can achieve a superior expressive power compared to SWL. This result is proven by Corollary 30 in the Appendix.

Theorem 16. CWL with a lifting map attaching cells to cliques and rings/cycles is strictly more powerful than SWL with a clique complex lifting at distinguishing non-isomorphic graphs.

4 Molecular Message Passing with CW Networks

We now describe CW Networks with an applied focus on molecular graphs to ground the discussion. Therefore, from now on we assume the use of the skeleton-preserving lifting transformation that attaches 2-cells to all the induced cycles (i.e. chordless cycles) in the graph as in Figure 2. This leads to a message passing procedure involving atoms (vertices / 0-cells), the bonds between atoms (edges / 1-cells) and chemical rings (induced cycles / 2-cells). Additionally, in virtue of Theorem 7, we consider only the boundary and upper adjacencies between these cells without sacrificing the expressive power. The equations for the other adjacencies, which we do not use, can be found in Appendix A. We note however, that the theoretical results in this section are general and not particular to these specific choices of adjacencies and lifting transformation.

before we only lifted cliques to complexes and
that is already more powerful than WL

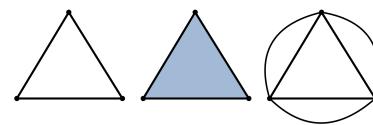


Figure 4: A graph, its clique complex and the graph with duplicated edges. The second is skeleton-preserving, while the last is not.

Our lifting operations!

- 1. Cliques as before
- 2. simple cycles
- 3. induced cycles

we only add cycles or faces

atoms to bonds + bonds to faces

*Why don't you split up these two into separate messages as well?
Could be done, but they want to keep it similar to
usual message passing*

Molecular Message Passing The cells in our CW Network receive two types of messages:

$$m_B^{t+1}(\sigma) = \text{AGG}_{\tau \in \mathcal{B}(\sigma)}(M_B(h_\sigma^t, h_\tau^t)) \quad m_{\uparrow}^{t+1}(\sigma) = \text{AGG}_{\tau \in \mathcal{N}_\uparrow(\sigma), \delta \in \mathcal{C}(\sigma, \tau)}(M_\uparrow(h_\sigma^t, h_\tau^t, h_\delta^t)).$$

The first specifies messages from atoms to bonds and from bonds to rings. The second type of message, specifies messages between atoms connected by a bond and messages between bonds that are part of the same ring. Note that for the second type of adjacency, when two atoms communicate, we include the features of the ring they communicate through. Similarly, when two bonds communicate, we include the features of the ring they are part of. The update operation takes into account these two types of incoming messages and updates the features of the cells:

$$h_\sigma^{t+1} = U(h_\sigma^t, m_B(\sigma), m_{\uparrow}^{t+1}(\sigma)). \quad (1)$$

To obtain a global embedding for a cell complex X from a model with L layers, the readout function takes as input the separate multi-sets of features corresponding to the atoms, bonds and the rings:

$$h_X = \text{READOUT}(\{h_\sigma^L\}_{\dim(\sigma)=0}, \{h_\sigma^L\}_{\dim(\sigma)=1}, \{h_\sigma^L\}_{\dim(\sigma)=2}). \quad (2)$$

*vertex to vertex +
bond to bond*

*co-boundary features: didn't you say you only
use boundary and upper adjacency features?
Yes we also use co-boundary but not as separate
message*

just pool everything as readout

Expressivity Naturally, the ability of CWNs to distinguish non-isomorphic regular cell complexes is bounded by CWL. Similarly to GNNs and WL, CWNs can also be shown to be as powerful as CWL as long as they are equipped with a sufficient number of layers and the parametric local aggregators they use can learn to be injective. Multiple such multi-set aggregators [19, 72] are known to exist and can be directly employed in our model.

Theorem 17. *CW Networks are at most as powerful as CWL. Additionally, when using injective neighbourhood aggregators and a sufficient number of layers, CWNs are as powerful as CWL.*

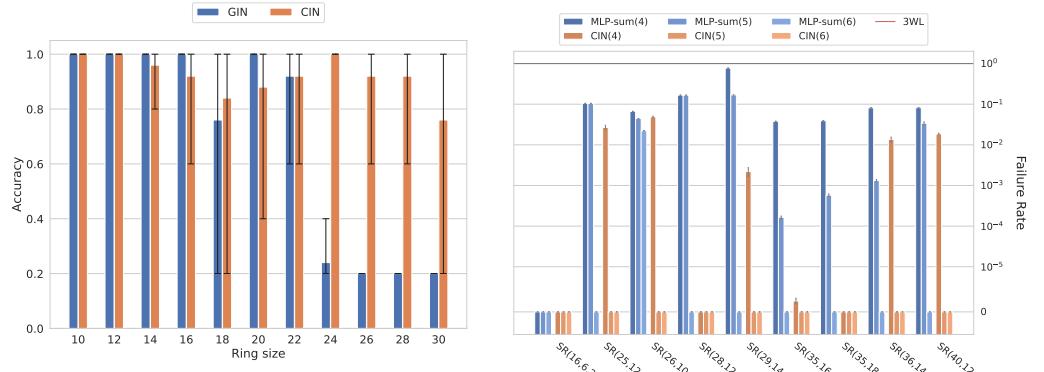
Symmetries Given a graph G with adjacency matrix A and feature matrix X , a function f is (node) permutation equivariant if $Pf(A, X) = f(PAP^T, PX)$, for any permutation matrix P . GNN layers respect this equation, which ensures they compute the same functions up to a permutation (i.e. relabeling) of the nodes. Similarly, it can be shown that CW Networks are equivariant with respect to permutations of the cells in the complex that preserve the boundary relation $\sigma \prec \tau$ between cells. We define this notion of equivariance more formally in Appendix B.

Theorem 18. *CW Network layers are (cell) permutation equivariant.*

Long-Range Interactions Several graph-related tasks require the ability to capture long-range interactions between nodes. For instance, certain molecular properties depend on atoms placed on the opposite sides of a ring [30, 58]. As a consequence of the coupling between the input and computational graphs, L message passing operations are necessary in GNNs to let a node receive information from an L -hops distant node. This intrinsic limitation has encouraged the design of *deep* GNNs, which, however, incur several drawbacks, including oversmoothing [49] and oversquashing [1] of node signals. Our hierarchical message passing schemes circumvent the aforementioned problems by reducing the overall number of layers which are needed in the presence of long-range node interactions. In fact, a cellular message passing model requires *at most* L layers to propagate information between nodes at distance L -hops in the graph, since 2-cells create shortcuts. For example, a constant number of CWN layers (3) is enough to capture dependencies between atoms on the opposite sides of a ring, independently of the ring size. In Section 5.1 we verify this in a controlled scenario. Additional experiments on real world graphs in Section 5.2 confirm that it can achieve state-of-the-art performance with a limited number of layers and/or parameters.

} 3 steps which will come later

CWNs as Generalised Convolutions Our message passing scheme can be seen a (non-linear) generalisation of linear diffusion operators on cell complexes. Recent works [12, 24] have introduced convolutional operators on SCs by employing the Hodge Laplacian [61], a generalisation of the graph Laplacian. By leveraging ~~the~~ the cellular Sheaf Laplacian [34], a similar construction can be extended to cell complexes to define cellular convolutional operators. In Appendix C we discuss this approach and show that our cellular message passing scheme subsumes it. This represents a promising avenue for studying CWNs from a spectral perspective, an endeavour we leave for future work.



(a) RingTransfer Results. Accuracy is over 5 balanced classes. A score of 0.2 is equivalent to a random guess. Error bars show the min and max. Our model obtains high-scores in average even for large rings despite using only three layers.

(b) Failure rates on the SR isomorphism task, *the smaller the better* (mean and std-error over 5 runs). In parentheses, for each model, the maximum size k of rings lifted to 2-cells.

Figure 5: Results on the RingTransfer and SR synthetic benchmarks.

Computational Complexity For all practical purposes, suppose we consider only cells of an arbitrary constant maximum dimension and constant maximum boundary size. Then, the computational complexity of our message passing scheme using the adjacencies from Theorem 7 can be shown to be linear in the size of the input complex. Separately of this, the one-time preprocessing step of computing the lifting of the graphs should also be considered. For our molecular graph application, this amounts to finding all the rings in the graph. These can be listed using, for instance, the junction tree representation [26, 41], obtained by specialised libraries such as RDKit [48]. In our experiments on molecular benchmarks the lifting procedure has been computationally negligible w.r.t. training runtimes. We include a detailed computational analysis and wall-clock time data both for the message passing and the lifting transformation in Appendix A.

not really any computational concerns for molecules at least

5 Experiments

In this section we validate the theoretical and empirical properties of our proposed message passing scheme in controlled scenarios as well as in real-world graph classification problems, with a focus on large scale molecular benchmarks. For simplicity, all experiments we employ a model which stacks CWN layers with local aggregators as in GIN [72]. We name our architecture “Cell Isomorphism Network” (CIN). 0-cells are always endowed with the original node features; higher-dimensional cells are populated in a benchmark specific manner. See Appendix D for details on feature initialisation, message passing and readout operations, hyperparameters, implementation and benchmark statistics.

5.1 Synthetic Benchmarks

Table 1: Classification accuracy on CSL.

Method	Mean	Min	Max
MP-GNNs	10.000±0.000	10.000	10.000
RingGNN	10.000±0.000	10.000	10.000
3WLGNN	97.800±10.916	30.000	100.000
CIN (Ours)	100.000±0.000	100.000	100.000

CSL Circular Skip Link dataset was first introduced in [55] and has been recently adopted as a reference benchmark to test the expressivity of GNNs [23]. It consists of 150 4-regular graphs from 10 different isomorphism classes, which we need to predict. Unsolvable by the WL test and message passing approaches [13, 55], we use it to validate the expressive power of CWNs.

We follow the same evaluation setting as Dwivedi et al. [23]: 5-fold cross validation procedure and 20 different random weight initialisations. For our model, we set the maximum ring size $k = 8$. In Table 1 we follow the common practice on this dataset and report the mean, minimum and maximum test accuracy obtained by CIN over the 100 runs, along with the results by the baselines presented in Dwivedi et al. [23]. MP-GNNs, that is classic message passing GNNs (GAT [67], MoNet [52], GIN [72], etc.), and RingGNN [13] perform as random guessers. In contrast, our model is able to identify the isomorphism class of each test graph in every run while featuring only a fraction of the computational complexity of 3WLGNN, the best performing reference baseline [23, 51].

Table 2: TUDatasets. The first section of the table includes the accuracy of graph kernel methods, while the second includes GNNs. The top three are highlighted by **First**, **Second**, **Third**.

Dataset	MUTAG	PTC	PROTEINS	NCI1	NCI109	IMDB-B	IMDB-M	RDT-B
RWK [28]	79.2 \pm 2.1	55.9 \pm 0.3	59.6 \pm 0.1	>3 days	N/A	N/A	N/A	N/A
GK ($k = 3$) [62]	81.4 \pm 1.7	55.7 \pm 0.5	71.4 \pm 0.31	62.5 \pm 0.3	62.4 \pm 0.3	N/A	N/A	N/A
PK [56]	76.0 \pm 2.7	59.5 \pm 2.4	73.7 \pm 0.7	82.5 \pm 0.5	N/A	N/A	N/A	N/A
WL kernel [63]	90.4 \pm 5.7	59.9 \pm 4.3	75.0 \pm 3.1	86.0 \pm 1.8	N/A	73.8 \pm 3.9	50.9 \pm 3.8	81.0 \pm 3.1
DCNN [3]	N/A	N/A	61.3 \pm 1.6	56.6 \pm 1.0	N/A	49.1 \pm 1.4	33.5 \pm 1.4	N/A
DGCNN [74]	85.8 \pm 1.8	58.6 \pm 2.5	75.5 \pm 0.9	74.4 \pm 0.5	N/A	70.0 \pm 0.9	47.8 \pm 0.9	N/A
IGN [50]	83.9 \pm 13.0	58.5 \pm 6.9	76.6 \pm 5.5	74.3 \pm 2.7	72.8 \pm 1.5	72.0 \pm 5.5	48.7 \pm 3.4	N/A
GIN [72]	89.4 \pm 5.6	64.6 \pm 7.0	76.2 \pm 2.8	82.7 \pm 1.7	N/A	75.1 \pm 5.1	52.3 \pm 2.8	92.4 \pm 2.5
PPGNs [51]	90.6 \pm 8.7	66.2 \pm 6.6	77.2 \pm 4.7	83.2 \pm 1.1	82.2 \pm 1.4	73.0 \pm 5.8	50.5 \pm 3.6	N/A
Natural GN [20]	89.4 \pm 1.6	66.8 \pm 1.7	71.7 \pm 1.0	82.4 \pm 1.3	N/A	73.5 \pm 2.0	51.3 \pm 1.5	N/A
GSN [9]	92.2 \pm 7.5	68.2 \pm 7.2	76.6 \pm 5.0	83.5 \pm 2.0	N/A	77.8 \pm 3.3	54.3 \pm 3.3	N/A
SIN [7]	N/A	N/A	76.4 \pm 3.3	82.7 \pm 2.1	N/A	75.6 \pm 3.2	52.4 \pm 2.9	92.2 \pm 1.0
CIN (Ours)	92.7 \pm 6.1	68.2 \pm 5.6	77.0 \pm 4.3	83.6 \pm 1.4	84.0 \pm 1.6	75.6 \pm 3.7	52.7 \pm 3.1	92.4 \pm 2.1

subgraph
isomorphism →
counting

SR Similarly to Bodnar et al. [7] and [9], we consider Strongly Regular graphs within the same family as hard examples of non-isomorphic graphs we seek to distinguish. Any pair of graphs within the same family cannot provably be distinguished by 3-WL test [7, 9]. We reproduce the same experimental setting of Bodnar et al. [7]. In particular, we consider 9 distinct SR families³ and run our model untrained on the cell complex lifting of each graph, with $k = 4, 5, 6$. 0-cells (nodes) are initialised with a constant unitary signal, while 1- and 2-cells are initialised with the sum of the contained 0-cells. We additionally run an MLP baseline with sum readout to appreciate the contribution of message passing. We report the percentage of non-distinguished pairs in Figure 5b. CIN is able to distinguish all graphs in all families for $k \geq 5$ (0.0% failure rate). Although the baseline model achieves relatively low failure rates for the same values of k , we observe that our message passing scheme distinguishes a larger number of non-isomorphic pairs in all settings.

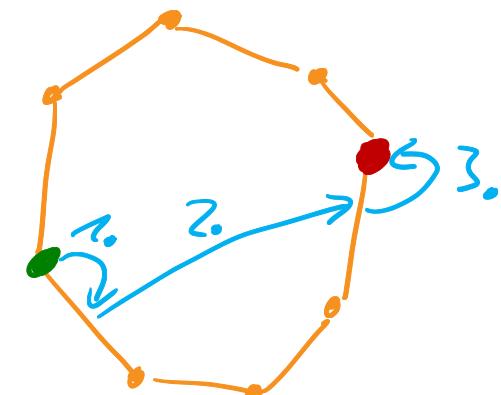
RingTransfer In order to empirically validate the ability of CIN to capture long-range node dependencies, we additionally design a third synthetic benchmark dubbed ‘RingTransfer’. Graphs in this dataset are chordless cycles (rings) of size k . In each graph we mark two special nodes as **target** and **source**, always placed at distance $\lfloor \frac{k}{2} \rfloor$. The task is for **target** to output the one-hot encoded label assigned to **source**. All other nodes in the ring are assigned a unitary constant feature vector. A model has to learn to transfer the information contained in **source** to the opposite side of the ring, where **target** resides. We initialise 1- and 2-dimensional cells with a null signal. In Figure 5a we show the performance of a 3-layer CIN as a function of the ring size k , along with that of GIN [72] baselines equipped with $\lfloor \frac{k}{2} \rfloor$ stacked layers. We observe that our model learns to solve the task with only 3 computational steps, independent of k . As for GIN, we observed degradation in the performance for $k \geq 24$, up to complete failure. We hypothesise this to be due to the difficulties of training such a deep GNN (≥ 12 layers). We further verify the (theoretically expected) failure of this model (not included) when endowed with less than $\lfloor \frac{k}{2} \rfloor$ layers.

5.2 Real-World Graph Benchmarks

TUD We test our model on 8 TUDataset benchmarks [54] with small and medium sizes from biology (**PROTEINS** [8, 22]), chemistry (i.e. molecules – **MUTAG** [43, 59], **PTC**, **NCI1** and **NCI109** [68]) to social networks (**IMDB-B**, **IMDB-M**, **RDT-B**). We consider induced cycle of size up to $k = 6$ for our graph lifting procedure. We initialise node (and 0-cell) features as described in Xu et al. [72], and higher dimensional cells by averaging or summing the features of the included 0-cells. The training setting and evaluation procedure follow those in Xu et al. [72]. We compare our CIN to the baseline methods: RWK [28], GK (with $k = 3$) [62], PK [56], WL kernel [63], DCNN [3], DGCNN [74], IGN [50], GIN [72], PPGNs [51], Natural GN [20], GSN [9], and SIN [7]. We report the results in Table 2. CIN compares more favourably with the baselines, displaying strong empirical performance on all benchmarks. The mean accuracy of CIN ranks top on four out of eight datasets. On the remaining datasets, CIN achieves the second place. We observe that the best results on datasets are from the biological and chemical domains, where rings play a relevant role.

³Data available at: <http://users.cecs.anu.edu.au/~bdm/data/graphs.html>.

Features of faces are just the sum of node features



3 steps because the information needs to go from
1. node to edge
2. edge to edge
3. edge to node

Table 3: ZINC (MAE), ZINC-FULL (MAE) and Mol-HIV (ROC-AUC).

Method	ZINC ↓		ZINC-FULL ↓	MOLHIV ↑
	No Edge Feat.	With Edge Feat.	All methods	All methods
GCN [45]	0.469±0.002	N/A	N/A	76.06±0.97
GAT [67]	0.463±0.002	N/A	N/A	N/A
GatedGCN [10]	0.422±0.006	0.363±0.009	N/A	N/A
GIN [72]	0.408±0.008	0.252±0.014	0.088±0.002	77.07±1.49
PNA [19]	0.320±0.032	0.188±0.004	N/A	79.05±1.32
DGN [5]	0.219±0.010	0.168±0.003	N/A	79.70±0.97
HIMP [26]	N/A	0.151±0.006	0.036±0.002	78.80±0.82
GSN [9]	0.139±0.007	0.108±0.018	N/A	77.99±1.00
CIN-small (Ours)	0.139±0.008	0.094±0.004	0.044±0.003	80.55±1.04
CIN (Ours)	0.115±0.003	0.079±0.006	0.022±0.002	80.94±0.57

max 100k
parameters →

ZINC We study the effectiveness of cellular message passing on larger scale molecular benchmarks from the ZINC database [66]. **ZINC** (12k graphs) and **ZINC-FULL** (250k graphs) [23, 32, 41, 73] are two graph regression task datasets for drug constrained solubility prediction. In these experiments, we consider rings up to size $k = 18$. We follow the training and evaluation procedures in [23]. Our experiments encompass different scenarios, examine the impact of ablating edge features and of constraining the parameter budget of the architecture to 100k. All results are illustrated in Table 3 where we also include the results for **ZINC-FULL** obtained by the same exact architectures. Our model exhibits particularly strong performance on these benchmarks: it attains state-of-the-art results on both the two dataset variants, outperforming other models by a significant margin. CIN attains strong results even when constrained by the parameter budget. It still achieves state-of-the-art performance on **ZINC** and is on-par with the best unconstrained baseline under edge-feature ablation.

Mol-HIV We additionally test our model on the molecular **ogbg-molhiv** dataset from the Open Graph Benchmark [38] (41k graphs). The task is to predict the capacity of compounds to inhibit HIV replication. Rings of size up to $k = 6$ are considered as 2-cells. We take the architecture in [26] as reference and replicate the same hyperparameter setting in our model, including the use of only 2 message passing layers. We report the mean of test ROC-AUC metrics at the epoch of best validation performance for 10 random weight initialisations. Similarly to ZINC, we experiment with a “small” model whose number of parameters is constrained in the order of 100k. Table 3 displays the results. CIN significantly outperforms other strong GNN baselines, even when constrained by the parameter budget. Consistently with [26], we observe that only two layers are sufficient when performing hierarchical message passing across meso-scale structures such as rings.

6 Discussion and Conclusion

Related Work Recent works have proposed the generalisation of GNNs to simplicial complexes [12, 24, 31]. All these convolutional methods are subsumed by the model in Bodnar et al. [7], which CWNs in turn subsume. To the best of our knowledge, Hajij et al. [33] is the only other example of message passing on cell complexes. In contrast to our model, their work is a proof of concept and does not include any experimental or theoretical analysis. A few works have extended GNNs to account for molecular substructures. Junction Trees (JT) conveniently represent singletons, bonds and rings as supernodes in a tree encoding their relations. They are used in molecular graph generation [41, 42], and the recent work of Fey et al. [26] employs them to design a hierarchical message passing scheme whereby messages are also exchanged on the JT structure. In contrast to CWNs, this model does not capture the full hierarchical cell complex structure. More general substructures are accounted for in GSNs [9], where isomorphism counting is employed to break symmetries between node neighbours and provably increase the expressive power of GNNs. Our model compares favourably in all benchmarks with both of these related works.

Limitations The main limitations of the model are of computational nature. While the computational complexity of the message passing procedure and its preprocessing step is suitable for molecular and geometric graphs, the number of rings (and more generally simple cycles) in general graphs can be exponential in the number of nodes. In that case, one has to resort to smaller 2-cells like triangles, which can be found efficiently in general graphs. Moreover, one has to typically use weights

Did you also try different message passing themes with lower adjacency for instance?

Yes, bad lower adjacencies will be very computational problematic

Idea: Better initialization of faces:

Is it a benzene?

A simple ring?

Include the size of the ring

specific for each dimension of the cell complex, increasing the number of parameters compared to GNNs. However, we have shown that our model can compensate this increase with a reduced number of layers and still achieve state-of-the-art results on some of the molecular benchmarks.

Societal Impacts Most of our paper is theoretical in nature and we do not see immediate direct negative societal impacts. Within the scope of social network applications, we do not yet have sufficient evidence of performance improvement on related benchmarks to justify obvious adoption in such a domain. In contrast, the empirical performance on molecular benchmarks suggests it may have a positive impact on applications of immediate interest in pharmaceutics, such as drug discovery [29].

Conclusion We have proposed a provably powerful message passing procedure on cell complexes motivated by a novel colour refinement algorithm to test their isomorphism. This allows us to consider flexible lifting operations on graphs to implement more expressive architectures which benefit from decoupling the computational and input graphs. Our methods show excellent performance on diverse synthetic and real-world molecular benchmarks.

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 are grateful to Ben Day, Gabriele Corso and Nikola Simidjievski for their helpful feedback on an early draft of this manuscript. We would also like to thank Vijay Prakash Dwivedi and Chaitanya K. Joshi for clarifying certain aspects of their Benchmarking GNNs [23] work.

References

- [1] Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. *ICLR*, 2021.
- [2] V. Arvind, Frank Fuhlbrück, Johannes Köbler, and Oleg Verbitsky. On weisfeiler-leman invariance: Subgraph counts and related graph properties. *Journal of Computer and System Sciences*, 113:42–59, 2020. ISSN 0022-0000.
- [3] James Atwood and Don Towsley. Diffusion-convolutional neural networks. In *NIPS*, pages 1993–2001, 2016.
- [4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- [5] Dominique Beaini, Saro Passaro, Vincent Létourneau, William L Hamilton, Gabriele Corso, and Pietro Liò. Directional graph networks. *ICML*, 2021.
- [6] Etienne E. Birmelé, Rui Ferreira, Roberto Grossi, Andrea Marino, Nadia Pisanti, Romeo Rizzi, and Gustavo Sacomoto. Optimal Listing of Cycles and st-Paths in Undirected Graphs. In *24th Annual ACM-SIAM Symposium on Discrete Algorithms*, New Orleans, United States, January 2013.
- [7] Cristian Bodnar, Fabrizio Frasca, Yu Guang Wang, Nina Otter, Guido Montúfar, Pietro Lio, and Michael Bronstein. Weisfeiler and lehman go topological: Message passing simplicial networks. *ICML*, 2021.
- [8] Karsten M. Borgwardt, Cheng Soon Ong, Stefan Schönauer, S. V. N. Vishwanathan, Alex J. Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1):i47–i56, 2005.
- [9] Giorgos Bouritsas, Fabrizio Frasca, Stefanos Zafeiriou, and Michael M Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *arXiv:2006.09252*, 2020.
- [10] Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.

- [11] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.
- [12] Eric Bunch, Qian You, Glenn Fung, and Vikas Singh. Simplicial 2-complex convolutional neural networks. In *NeurIPS Workshop on Topological Data Analysis and Beyond*, 2020.
- [13] Zhengdao Chen, Soledad Villar, Lei Chen, and Joan Bruna. On the equivalence between graph isomorphism testing and function approximation with gnns. In *NeurIPS*, volume 32, 2019.
- [14] Zhengdao Chen, Lei Chen, Soledad Villar, and Joan Bruna. Can graph neural networks count substructures? In *NeurIPS*, 2020.
- [15] N. Chiba and Takao Nishizeki. Arboricity and subgraph listing algorithms. *SIAM J. Comput.*, 14:210–223, 1985.
- [16] Fan Chung. *Spectral graph theory*. American Mathematical Society, 1997.
- [17] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). In *ICLR*, 2016.
- [18] Luigi P. Cordella, Pasquale Foglia, Carlo Sansone, and Mario Vento. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1367–1372, 2004.
- [19] Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. Principal neighbourhood aggregation for graph nets. In *NeurIPS*, volume 33, pages 13260–13271, 2020.
- [20] Pim de Haan, Taco Cohen, and Max Welling. Natural graph networks. In *NeurIPS*, 2020.
- [21] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, 2016.
- [22] Paul D Dobson and Andrew J Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of Molecular Biology*, 330(4):771–783, 2003.
- [23] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*, 2020.
- [24] Stefania Ebli, Michaël Defferrard, and Gard Spreemann. Simplicial neural networks. In *NeurIPS Workshop on Topological Data Analysis and Beyond*, 2020.
- [25] Rui Ferreira, Roberto Grossi, Romeo Rizzi, Gustavo Sacomoto, and Marie-France Sagot. Amortized $\tilde{O}(|V|)$ -Delay Algorithm for Listing Chordless Cycles in Undirected Graphs. In *22th Annual European Symposium on Algorithms*, volume 8737 of *Algorithms ESA 2014, Lecture Notes in Computer Science*, pages 418–429, Wroclaw, Poland, September 2014.
- [26] M. Fey, J. G. Yuen, and F. Weichert. Hierarchical inter-message passing for learning on molecular graphs. In *ICML Graph Representation Learning and Beyond (GRL+) Workshop*, 2020.
- [27] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.
- [28] Thomas Gärtner, Peter Flach, and Stefan Wrobel. On graph kernels: Hardness results and efficient alternatives. In *Learning theory and kernel machines*, pages 129–143. Springer, 2003.
- [29] Thomas Gaudelet, Ben Day, Arian R Jamasb, Jyothish Soman, Cristian Regep, Gertrude Liu, Jeremy B R Hayter, Richard Vickers, Charles Roberts, Jian Tang, David Roblin, Tom L Blundell, Michael M Bronstein, and Jake P Taylor-King. Utilizing graph machine learning within drug discovery and development. *Briefings in Bioinformatics*, 05 2021. ISSN 1477-4054.
- [30] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, 2017.

- [31] Nicholas Glaze, T. Mitchell Roddenberry, and Santiago Segarra. Principled simplicial neural networks for trajectory prediction. In *ICML*, 2021.
- [32] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018.
- [33] Mustafa Hajij, Kyle Istvan, and Ghada Zamzmi. Cell complex neural networks. In *NeurIPS Workshop on Topological Data Analysis and Beyond*, 2020.
- [34] J. Hansen and R. Ghrist. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology*, 3:315–358, 2019.
- [35] Jakob Hansen and Thomas Gebhart. Sheaf neural networks. In *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*, 2020.
- [36] Allen Hatcher. *Algebraic topology*. Cambridge Univ. Press, Cambridge, 2000.
- [37] Christoph Helma, Ross D. King, Stefan Kramer, and Ashwin Srinivasan. The predictive toxicology challenge 2000–2001. *Bioinformatics*, 17(1):107–108, 2001.
- [38] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*, 2020.
- [39] Weihua Hu*, Bowen Liu*, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020.
- [40] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.
- [41] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *ICML*, pages 2323–2332, 2018.
- [42] Wengong Jin, Kevin Yang, Regina Barzilay, and Tommi Jaakkola. Learning multimodal graph-to-graph translation for molecular optimization. In *ICLR*, 2019.
- [43] Jeroen Kazius, Ross McGuire, and Roberta Bursi. Derivation and validation of toxicophores for mutagenicity prediction. *Journal of Medicinal Chemistry*, 48(1):312–320, 2005.
- [44] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [45] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.
- [46] Nils Kriege and Petra Mutzel. Subgraph matching kernels for attributed graphs. *arXiv preprint arXiv:1206.6483*, 2012.
- [47] Casimir Kuratowski. Sur le probleme des courbes gauches en topologie. *Fundamenta mathematicae*, 15(1):271–283, 1930.
- [48] Greg Landrum. RDKit: Open-source cheminformatics software. 2016.
- [49] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, volume 32, 2018.
- [50] Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. In *ICLR*, 2018.
- [51] Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. In *NeurIPS*, 2019.

- [52] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model CNNs. In *CVPR*, pages 5425–5434, 2017.
- [53] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and Leman go neural: Higher-order graph neural networks. In *AAAI*, 2019.
- [54] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. TUDataset: A collection of benchmark datasets for learning with graphs. In *ICML Graph Representation Learning and Beyond (GRL+) Workshop*, 2020.
- [55] Ryan Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Relational pooling for graph representations. In *ICML*, volume 97, pages 4663–4673, 2019.
- [56] Marion Neumann, Roman Garnett, Christian Bauckhage, and Kristian Kersting. Propagation kernels: efficient graph kernels from propagated information. *Machine Learning*, 102(2):209–245, 2016.
- [57] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035. 2019.
- [58] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1):1–7, 2014.
- [59] Kaspar Riesen and Horst Bunke. Iam graph database repository for graph based pattern recognition and machine learning. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 287–297. Springer, 2008.
- [60] Ryoma Sato. A survey on the expressive power of graph neural networks. *arXiv preprint arXiv:2003.04078*, 2020.
- [61] Michael T Schaub, Austin R Benson, Paul Horn, Gabor Lippner, and Ali Jadbabaie. Random walks on simplicial complexes and the normalized Hodge 1-Laplacian. *SIAM Review*, 62(2):353–391, 2020.
- [62] Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. Efficient graphlet kernels for large graph comparison. In *AISTAT*, pages 488–495, 2009.
- [63] Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *JMLR*, 12(Sep):2539–2561, 2011.
- [64] Howard E. Simmons and John E. Maggio. Synthesis of the first topologically non-planar molecule. *Tetrahedron Letters*, 22(4):287–290, 1981. ISSN 0040-4039.
- [65] Amit Singer and Hau-Tieng Wu. Vector diffusion maps and the connection laplacian. *Communications in Pure and Applied Mathematics*, 68(8), 2012.
- [66] Teague Sterling and John J. Irwin. ZINC 15 – ligand discovery for everyone. *Journal of Chemical Information and Modeling*, 55(11):2324–2337, 11 2015. doi: 10.1021/acs.jcim.5b00559. URL <https://doi.org/10.1021/acs.jcim.5b00559>.
- [67] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.
- [68] Nikil Wale, Ian A Watson, and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 14(3):347–375, 2008.

- [69] Boris Weisfeiler and Andrei Leman. The reduction of a graph to canonical form and the algebra which appears therein. *NTI Series*, 2(9):12–16, 1968.
- [70] J. H. C. Whitehead. Combinatorial homotopy. I. *Bulletin of the American Mathematical Society*, 55(3.P1):213 – 245, 1949. doi: bams/1183513543.
- [71] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. In *ICML*, 2018.
- [72] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.
- [73] Jiaxuan You, Bowen Liu, Rex Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. *NeurIPS*, 2018.
- [74] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *AAAI*, 2018.

A Proofs

A.1 Cellular WL Results

In this section, we assume basic familiarity with the WL test and its higher-order variants. For an introduction to these topics, we refer the reader to the survey of Sato [60]. We begin by introducing a few useful concepts.

Definition 19. A *cellular colouring* is a map c that maps a cell complex X and one of its cells σ to a colour from a fixed colour palette. We denote this colour by c_σ^X .

Definition 20. Let X, Y be two regular cell complexes and c a cellular colouring. We say that X, Y are *c-similar*, denoted by $c^X = c^Y$, if the number of cells in X coloured with a given colour equals the number of cells in Y with the same colour. Otherwise, we have $c^X \neq c^Y$.

We emphasise that in this paper we are interested only in colourings c with the property that any two isomorphic cell complexes are c -similar.

Definition 21. A cellular colouring c *refines* a cellular colouring d , denoted by $c \sqsubseteq d$, if for all cell complexes X and Y and all $\sigma \in P_X$ and $\tau \in P_Y$, $c_\sigma^X = c_\tau^Y$ implies $d_\sigma^X = d_\tau^Y$. Additionally, if $d \sqsubseteq c$, we say the two colourings are *equivalent* and we represent it by $c \equiv d$.

We state the following result from Bodnar et al. [7] about simplicial colourings, which we translate here directly to cell complexes. The proof is however, identical, and we refer the reader to their work for that.

Proposition 22. Let X, Y be any regular cellular complexes with $A \subseteq P_X$ and $B \subseteq P_Y$. Consider two cellular colourings c, d such that $c \sqsubseteq d$. If $\{d_\sigma^X \mid \sigma \in A\} \neq \{d_\tau^Y \mid \tau \in B\}$, then $\{c_\sigma^X \mid \sigma \in A\} \neq \{c_\tau^Y \mid \tau \in B\}$.

Corollary 23. Consider two cellular colourings c, d such that $c \sqsubseteq d$. For all cell complexes X and Y , if $d^X \neq d^Y$, then $c^X \neq c^Y$.

This last result implies that if c refines d , then c can distinguish all the non-isomorphic cell complexes that d can distinguish. We say that the colouring c is at least as powerful as the colouring d .

In contrast to simplicial complexes, cell complexes have a more flexible structure. The main complication compared to the proofs in Bodnar et al. [7] is that cells can have a variable number of lower-dimensional cells on their boundary. It is therefore useful in many proofs, to separate the cells into buckets containing cells with the same boundary size. The following result helps us do that.

Proposition 24. Let c^t be the CWL colouring at iteration t . For all cells σ, τ in any cell complexes X and Y , if $|\mathcal{B}(\sigma)| \neq |\mathcal{B}(\tau)|$, then for any $t > 0$ we have $c_\sigma^t \neq c_\tau^t$.

Proof. If σ and τ have boundaries of different sizes, then $c_{\mathcal{B}}^1(\sigma) \neq c_{\mathcal{B}}^1(\tau)$, which immediately implies $c_\sigma^t \neq c_\tau^t$ for all $t > 0$. \square

Next, we show that one can drop the co-boundary adjacencies without sacrificing expressive power.

Lemma 25. CWL with $\text{HASH}(c_\sigma^t, c_{\mathcal{B}}^t(\sigma), c_\downarrow^t(\sigma), c_\uparrow^t(\sigma))$ is as powerful as CWL with the generalised update rule $\text{HASH}(c_\sigma^t, c_{\mathcal{B}}^t(\sigma), c_{\mathcal{C}}^t(\sigma), c_\downarrow^t(\sigma), c_\uparrow^t(\sigma))$.

Proof. Let a^t denote the colouring produced by CWL using the general version and b^t the colouring produced using the restricted version at iteration t . It can be verified that $a^t \sqsubseteq b^t$ because it considers the additional $c_{\mathcal{B}}^t(\sigma)$ colours in the refinement rule. We now prove $b^{t+1} \sqsubseteq a^t$ by induction. Note that to take advantage of Proposition 24, we shift the time-step by one (i.e. we use b^{t+1} as opposed to b^t).

The base case holds since a^0 assigns the same colour to all the cells. Suppose $b_\sigma^{t+2} = b_\tau^{t+2}$ for any two cells σ and τ from any cell complexes X and Y , respectively. Then we know that $b_\sigma^{t+1} = b_\tau^{t+1}$, $b_{\mathcal{B}}^{t+1}(\sigma) = b_{\mathcal{B}}^{t+1}(\tau)$, $b_\uparrow^{t+1}(\sigma) = b_\uparrow^{t+1}(\tau)$ and $b_\downarrow^{t+1}(\sigma) = b_\downarrow^{t+1}(\tau)$. The goal is to show that this also implies that $b_{\mathcal{C}}^{t+1}(\sigma) = b_{\mathcal{C}}^{t+1}(\tau)$.

Given $b_\uparrow^{t+1}(\sigma) = b_\uparrow^{t+1}(\tau)$, by definition

$$\{b_{\delta_\sigma}^{t+1} \mid (\cdot, b_{\delta_\sigma}^{t+1}) \in b_\uparrow^{t+1}(\sigma)\} = \{b_{\delta_\tau}^{t+1} \mid (\cdot, b_{\delta_\tau}^{t+1}) \in b_\uparrow^{t+1}(\tau)\}.$$

By Proposition 24, cells with different boundary sizes have different colours. Therefore, we can partition these two multi-sets by the size of the cell boundaries, while preserving the equality between these sub-multisets. Therefore, for each $n \in \mathbb{N}$:

$$\{\{b_{\delta_\sigma}^{t+1} \mid (\cdot, b_{\delta_\sigma}^{t+1}) \in b_\uparrow^{t+1}(\sigma) \text{ and } |\mathcal{B}(\delta_\sigma)| = n\}\} = \{\{b_{\delta_\tau}^{t+1} \mid (\cdot, b_{\delta_\tau}^{t+1}) \in b_\uparrow^{t+1}(\tau) \text{ and } |\mathcal{B}(\delta_\tau)| = n\}\}.$$

Let γ be an arbitrary cell. Then for each cell $\delta_\gamma \in \mathcal{C}(\gamma)$, γ exchanges messages with all the other boundary cells of δ_γ . Therefore, the colour of each δ_γ with $|\mathcal{B}(\delta_\gamma)| = n$ shows up with a multiplicity of $n - 1$ in the tuples of $b_\uparrow^{t+1}(\gamma)$. Eliminating $n - 2$ of these repeated colours for all δ_σ and δ_τ :

$$\{\{b_{\delta_\sigma}^{t+1} \mid \delta_\sigma \in \mathcal{C}(\sigma) \text{ and } |\mathcal{B}(\delta_\sigma)| = n\}\} = \{\{b_{\delta_\tau}^{t+1} \mid \delta_\tau \in \mathcal{C}(\tau) \text{ and } |\mathcal{B}(\delta_\tau)| = n\}\}.$$

Merging these in a single multi-set gives the colours of the co-boundary cells:

$$b_\mathcal{C}^{t+1}(\sigma) = \{\{b_{\delta_\sigma}^{t+1} \mid \delta_\sigma \in \mathcal{C}(\sigma)\}\} = \{\{b_{\delta_\tau}^{t+1} \mid \delta_\tau \in \mathcal{C}(\tau)\}\} = b_\mathcal{C}^{t+1}(\tau).$$

By the induction hypothesis, $a_\sigma^t = a_\tau^t$, $a_\mathcal{B}^t(\sigma) = a_\mathcal{B}^t(\tau)$, $a_\mathcal{C}^t(\sigma) = a_\mathcal{C}^t(\tau)$, $a_\uparrow^t(\sigma) = a_\uparrow^t(\tau)$ and $a_\downarrow^t(\sigma) = a_\downarrow^t(\tau)$. This implies $a_\sigma^{t+1} = a_\tau^{t+1}$. \square

The following theorem shows that we can further prune the CWL update rule by removing the colours associated with the lower adjacencies. The structure of the proof is similar to the one in Bodnar et al. [7], with the main difference being in the proof of Proposition 26.

Proof of Theorem 7. Let b^t denote the colouring of CWL using $\text{HASH}(b_\sigma^t, b_\mathcal{B}^t(\sigma), b_\uparrow^t(\sigma))$ and a^t the colouring of CWL using the rule $\text{HASH}(a_\sigma^t, a_\mathcal{B}^t(\sigma), a_\downarrow^t(\sigma), a_\uparrow^t(\sigma))$ from Lemma 25. Trivially $a^t \sqsubseteq b^t$ because of the additional argument $c_\downarrow^t(\sigma)$ in the update rule. We prove $b^{2t+1} \sqsubseteq a^t$ by induction. As before, the addition by one in $2t + 1$ is to allow us to apply Proposition 24 in the induction step. The multiplication by 2 is due to the fact that the information transmitted through the lower adjacencies in one step is propagated in two steps through the boundary adjacencies.

As before, the base case trivially holds since a^0 assigns the same colour to all cells. Suppose $b_\sigma^{2t+3} = b_\tau^{2t+3}$. By unrolling the hash function two steps in time, we obtain $b_\sigma^{2t+1} = b_\tau^{2t+1}$, $b_\mathcal{B}^{2t+1}(\sigma) = b_\mathcal{B}^{2t+1}(\tau)$, and $b_\uparrow^{2t+1}(\sigma) = b_\uparrow^{2t+1}(\tau)$. We need to prove that $b_\downarrow^{2t+1}(\sigma) = b_\downarrow^{2t+1}(\tau)$ also holds. For the sake of contradiction, assume $b_\downarrow^{2t+1}(\sigma) \neq b_\downarrow^{2t+1}(\tau)$. Then there exists a pair of colours $(\mathbb{C}_0, \mathbb{C}_1)$ that shows up (without loss of generality) more times in $b_\downarrow^{2t+1}(\sigma)$ than in $b_\downarrow^{2t+1}(\tau)$. For simplicity, we also assume $b_\sigma^{2t+1} \neq \mathbb{C}_0 \neq b_\tau^{2t+1}$ as this special case can be easily treated separately.

For all cell complexes X and all cells δ in P_X , consider the collection of multi-sets A_X indexed by δ :

$$A_X(\delta) = \{\{(b_\psi^{2t+1} = \mathbb{C}_0, b_\delta^{2t+1} = \mathbb{C}_1) \mid \psi \in \mathcal{C}(\delta)\}\}.$$

We are interested in the size of these multi-sets for some specific cells δ . To that end, for each cell $\gamma \in P_X$, we define the multi-set:

$$C_X(\gamma) = \{\{|A_X(\delta)| \mid \delta \in \mathcal{B}(\gamma)\}\}.$$

We know that $C_X(\sigma) \neq C_Y(\tau)$ since the sum of the elements of $C_X(\sigma)$, which gives the number of tuples $(\mathbb{C}_0, \mathbb{C}_1)$ in $b_\downarrow^{2t+1}(\sigma)$, is greater than the sum of the elements of $C_Y(\tau)$, which gives the number of tuples $(\mathbb{C}_0, \mathbb{C}_1)$ in $b_\downarrow^{2t+1}(\tau)$. We prove this contradicts our hypothesis that $b_\sigma^{2t+3} = b_\tau^{2t+3}$.

Proposition 26. For all regular cell complexes X, Y and all $\sigma \in P_X, \tau \in P_Y$, if $C_X(\sigma) \neq C_Y(\tau)$, then $b_\sigma^{2t+3} \neq b_\tau^{2t+3}$.

Proof. Given a cell complex X and a cell $\delta \in P_X$, consider the cellular colouring $c_\delta^X = |A_X(\delta)|$. The idea of the proof is to show that $b^{2t+2} \sqsubseteq c$, which allows us to use Proposition 22 for the multi-sets $C_X(\sigma)$ and $C_Y(\tau)$.

Let δ_1, δ_2 be two arbitrary cells from any regular cell complexes X, Y such that $c_{\delta_1}^X \neq c_{\delta_2}^Y$. Assume without loss of generality that $|A_X(\delta_1)| > |A_Y(\delta_2)|$. Two cases can be distinguished for this inequality. In the first case, $b_{\delta_2}^{2t+1} \neq \mathbb{C}_1$, which implies $|A_X(\delta_1)| > |A_Y(\delta_2)| = 0$ and, therefore, $b_{\delta_1}^{2t+1} = \mathbb{C}_1$. Then $b_{\delta_1}^{2t+2} \neq b_{\delta_2}^{2t+2}$.

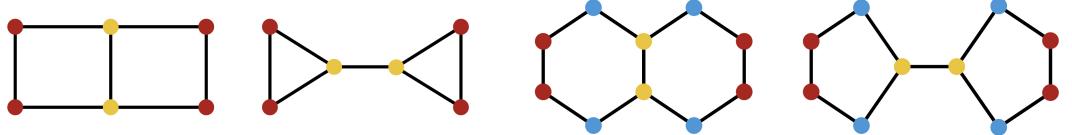


Figure 6: (Left) A pair of non-isomorphic graphs indistinguishable by WL, but distinguishable by CWL with a clique complex, ring or cycle-based lifting. (Right) A pair of non-isomorphic molecular graphs (Decalin and Bicyclopentyl) indistinguishable by WL but distinguishable by CWL with a ring-based or cycle-based lifting. The node colours show the stable colouring reached by WL.

In the second case, $b_{\delta_2}^{2t+1} = \mathbb{C}_1$, which implies $|A_X(\delta_1)| > |A_Y(\delta_2)| \geq 0$ and $b_{\delta_1}^{2t+1} = \mathbb{C}_1$. Then, the difference in the size of the multi-sets is made by the number of times \mathbb{C}_0 shows up in $A_X(\delta_1)$ and $A_Y(\delta_2)$, respectively. By Proposition 24, all k -cells γ with $k > 0$ and $b_{\gamma}^{2t+1} = \mathbb{C}_0$ must have a fixed boundary size $|\mathcal{B}(\gamma)| = n$. Because each cell $\delta \in \mathcal{B}(\gamma)$ is upper adjacent with every other cell in $\mathcal{B}(\gamma)$, b_{γ}^{2t+1} appears $n - 1$ times in the tuples inside $b_{\gamma}^{2t+1}(\delta)$. Additionally, note that since the cell complex is regular, self-loops are not allowed and, therefore, $n > 1$.

Applying this to δ_1 and δ_2 , \mathbb{C}_0 shows up $|A_X(\delta_1)| \times (n - 1)$ times in $b_{\delta_1}^{2t+1}(\delta_1)$ and $|A_Y(\delta_2)| \times (n - 1)$ times in $b_{\delta_2}^{2t+1}(\delta_2)$. Therefore, $b_{\delta_1}^{2t+1}(\delta_1) \neq b_{\delta_2}^{2t+1}(\delta_2)$ and, similarly to the first case, $b_{\delta_1}^{2t+2} \neq b_{\delta_2}^{2t+2}$. The results obtained for the two cases prove $b^{2t+2} \sqsubseteq c$.

Applying Proposition 22 for the multi-sets $C_X(\sigma)$ and $C_Y(\tau)$, we obtain two non-equal multi-sets:

$$b_{\mathcal{B}}^{2t+2}(\sigma) = \{\{b_{\delta_1}^{2t+2} \mid \delta_1 \in \mathcal{B}(\sigma)\}\} \neq \{\{b_{\delta_2}^{2t+2} \mid \delta_2 \in \mathcal{B}(\tau)\}\} = b_{\mathcal{B}}^{2t+2}(\tau)$$

Since these two multi-sets are used in the colour updating rule, $b_{\sigma}^{2t+3} \neq b_{\tau}^{2t+3}$. \square

Therefore, $b_{\downarrow}^{2t+1}(\sigma) = b_{\downarrow}^{2t+1}(\tau)$. Finally, applying the induction hypothesis, we have that $a_{\sigma}^t = a_{\tau}^t$, $a_{\mathcal{B}}^t(\sigma) = a_{\mathcal{B}}^t(\tau)$, $a_{\uparrow}^t(\sigma) = a_{\uparrow}^t(\tau)$ and $a_{\downarrow}^t(\sigma) = a_{\downarrow}^t(\tau)$. Then $a_{\sigma}^{t+1} = a_{\tau}^{t+1}$. \square

Proof of Theorem 13. Consider the map $f : \mathcal{G} \rightarrow \mathcal{X}$, a skeleton-preserving lifting transformation from the space of graphs \mathcal{G} , to the space of regular cell complexes \mathcal{X} . Let $g_G : V_G \rightarrow P_{f(G)^{(0)}}$ be the graph isomorphism associated to f between the vertices of G and the 0-cells of $f(G)$ for all $G \in \mathcal{G}$. Let $c^{G,t}$ be the WL colouring of graph G at iteration t and $a^{f(G),t}$ the colouring of $f(G)^{(1)}$ induced by the isomorphism g_G (i.e $a_{g(v)}^{f(G),t} := c_v^{G,t}$) at the same time step t .

Because $f(G)^{(1)}$ and G are isomorphic as graphs and WL is invariant under isomorphism, $a^{f(G),t} = c^{G,t}$. It follows that for all graphs $G_1, G_2 \in \mathcal{G}$, if $c^{G_1,t} \neq c^{G_2,t}$ then $a^{f(G_1),t} \neq a^{f(G_2),t}$. Let b^t be the CWL colouring of the 0-cells at iteration t . The goal is to show that for all regular cell complexes $X, Y \in f(\mathcal{G})$, $b^t \sqsubseteq a^t$. By transitivity and combined with Corollary 23, it follows that if $c^{G_1,t} \neq c^{G_2,t}$, then $b^{f(G_1),t} \neq b^{f(G_2),t}$.

The base case trivially holds. Let σ, τ be two 0-cells in $X \in f(\mathcal{G})$ and $Y \in f(\mathcal{G})$, respectively such that $b^{t+1}(\sigma) = b^{t+1}(\tau)$. Since 0-cells have only upper adjacencies, the equality implies that $b_{\sigma}^t = b_{\tau}^t$ and $b_{\uparrow}^t(\sigma) = b_{\uparrow}^t(\tau)$. The latter multi-set equality further implies

$$\{\{b_{\delta_{\sigma}}^t \mid (b_{\delta_{\sigma}}^t, \cdot) \in b_{\uparrow}^t(\sigma)\}\} = \{\{b_{\delta_{\tau}}^t \mid (b_{\delta_{\tau}}^t, \cdot) \in b_{\uparrow}^t(\tau)\}\}.$$

Equivalently, for 0-cells of a cell complex whose 1-skeleton is a graph (i.e. not a multi-graph), this can be rewritten as

$$\{\{b_{\delta_{\sigma}}^t \mid \delta_{\sigma} \in \mathcal{N}_{\uparrow}(\sigma)\}\} = \{\{b_{\delta_{\tau}}^t \mid \delta_{\tau} \in \mathcal{N}_{\uparrow}(\tau)\}\}.$$

By the induction hypothesis we have $a_{\sigma}^t = a_{\tau}^t$ and

$$\{\{a_{\delta_{\sigma}}^t \mid \delta_{\sigma} \in \mathcal{N}_{\uparrow}(\sigma)\}\} = \{\{a_{\delta_{\tau}}^t \mid \delta_{\tau} \in \mathcal{N}_{\uparrow}(\tau)\}\}.$$

These equalities imply $a^{t+1}(\sigma) = a^{t+1}(\tau)$. \square

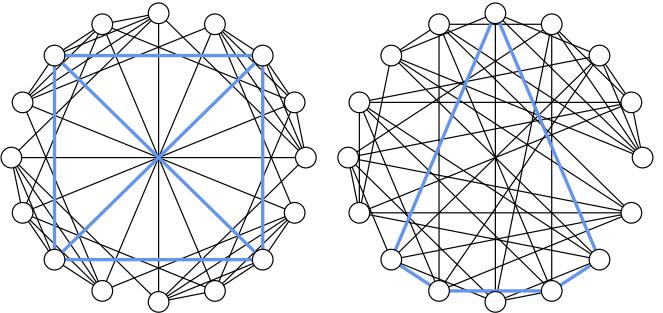


Figure 7: The two SR graphs in family $\text{SR}(16, 6, 2, 2)$: Rook's 4×4 (left) and Shrikhande (right). The 3-WL test is not able to deem them as non-isomorphic. Contrary to the Shrikhande graph, Rook's graph possesses 4-cliques. The Shrikhande graph, however, features 5-rings, not present in Rook's. Instances of these substructures are marked in blue. With appropriate lifting procedures, CWL can distinguish between them.

Proof of Corollary 14. Due to Theorem 13, it is sufficient to find some examples of non-isomorphic graph pairs that WL cannot distinguish, but CWL can with the given lifting transformations. Figure 6 includes such examples. Based on Proposition 24, CWL can distinguish these graphs since it can count the number of substructures (e.g. triangles, rings, cycles) that the lifting is based on. \square

The next proposition shows that CWL can identify cells that are n -simplices.

Proposition 27 (Simplex Identification). *Let X, Y be regular cell complexes and $\sigma \in P_X, \tau \in P_Y$ two cells. Denote by c^t the CWL colouring at iteration t . Suppose σ is an n -simplex and τ is not. Then $c^t(\sigma) \neq c^t(\tau)$ for all $t \geq n + 1$.*

Proof. The base case holds since $c_\sigma^1 \neq c_\tau^1$ if σ is a vertex and τ is a cell of another dimension. This is because σ has no boundary adjacencies, while τ does.

Suppose the statement holds for n -simplices. Then, an $(n + 1)$ -simplex can be identified by having $n + 2$ n -simplices on its boundary. By Proposition 24, the colour of σ encodes the boundary size. Furthermore, by the induction hypothesis $c_B^{n+1}(\sigma)$ encodes the fact that the boundary cells are n -simplices. \square

Proof of Theorem 15. Most of the sub-results of the theorem can be proven by finding pairs of graphs from the same family of Strongly Regular Graphs that can be distinguished by CWL with the corresponding lifting transformations. Graphs in this family are provably indistinguishable by the higher-order 3-WL test [7].

Ring-based lifting We can show that there is a pair of SR graphs in the same family with a different number of induced cycles of a certain size. We include such an example in Figure 7. In the example, the graph on the left does not possess 5-rings, contrary to the graph on the right (one such example is marked in blue). This proves that 3-WL cannot count induced cycles of size 5. In fact, a computational search reveals that the two graphs *also* differ in the number of 4-, 6- and 8-rings (see Table 4), which indirectly proves this algorithm cannot count induced cycles of these sizes. It is therefore natural to conjecture that 3-WL cannot count induced cycles of size strictly larger than 3.

Clique complex lifting We can leverage on the same example: the graph on the right does not possess 4-cliques, contrary to the graph on the left (one such example is marked in blue). This proves that 3-WL cannot count cliques of size 4. As shown by Bodnar et al. [7], this result immediately implies that SWL (and consequently CWL) with a clique complex lifting is not less powerful than 3-WL.

Cycle-based lifting To prove the result for this lifting transformation we leverage on a result by Arvind et al. [2], who show that 2-Folklore WL (which is equivalent to 3-WL [60]) cannot

Table 4: Number of induced cycles (rings) on the SR graphs in family SR(16, 6, 2, 2).

Graph ↓ / Size →	3 (Tri.)	4	5	6	7	8
Rook's 4×4	32	36	0	96	0	72
Shrikhande	32	12	96	64	0	36

count subgraph cycles of size larger than 7. Since CWL can count the number of cycles when the corresponding lifting is used no matter the size of the cycle (see Proposition 24), this proves the result.

The union of the transformations By Proposition 27, CWL can distinguish n -simplices ($(n+1)$ -cliques) from general 2-cells. Therefore, CWL can still distinguish the graphs from Figure 7 based on the number of cliques when using such a lifting transformation. \square

We note that while the proof above is purely based on substructure counts, the superior expressive power of CWL is very likely not limited to counting the substructures involved in the lifting transformation. We have seen evidence in favour of this claim in the SR experiment in Section 5, where message passing layers reduced the failure rate.

Next, we prove a statement comparing Simplicial WL and CWL. This will later be used to show that CWNs are strictly more powerful than MPSNs when a lifting transformation based on the clique complex and rings is used.

Definition 28. A subset L of a cell complex X is called a **subcomplex** if it is a union of cells of X containing the closures of these cells.

Theorem 29. Let $f : \mathcal{G} \rightarrow \mathcal{X}$ be a skeleton-preserving transformation such that for any graph G , the clique complex of G is a subcomplex of $f(G)$. Then CWL using the lifting transformation f is at least as powerful as SWL using the clique complex lifting at distinguishing non-isomorphic graphs.

Proof. Let c^t be the simplicial colouring performed by SWL. We can extend it into a cellular colouring a^t defined as follows:

$$a_\sigma^{X,t} := \begin{cases} c_\sigma^{L,t} & \text{if } X_\sigma \subseteq L \\ \circ & \text{otherwise} \end{cases}$$

where L is the maximal simplicial complex that is a subcomplex of X and \circ is a special colour assigned to the cells that are not simplices. Let $h : \mathcal{G} \rightarrow \mathcal{X}$ be the clique-complex lifting map. Then, it is easy to see that for all graphs $G_1, G_2 \in \mathcal{G}$, if $c^{h(G_1),t} \neq c^{h(G_2),t}$, then $a^{f(G_1),t} \neq a^{f(G_2),t}$. Let b^t be the CWL colouring map at iteration t . We aim to show that $b^{t+n+1} \sqsubseteq a^t$ by using Proposition 27. Then, by transitivity and using Corollary 22, if $c^{h(G_1),t} \neq c^{h(G_2),t}$, then $b^{f(G_1),t+n+1} \neq b^{f(G_2),t+n+1}$.

Let n be the maximum dimension of the cells used by the lifting transformation f . As usual, the base case holds at initialisation since a^0 assigns the same colour to all the cells. Let σ, τ be two cells from the regular cell complexes $X, Y \in f(\mathcal{G})$. When σ and τ are not simplices, then $a_\sigma^t = a_\tau^t = \circ$. Suppose σ and τ are both simplices and $b_\sigma^{t+n+2} = b_\tau^{t+n+2}$. Then we know that $b_\sigma^{t+n+1} = b_\tau^{t+n+1}, b_B^{t+n+1}(\sigma) = b_B^{t+n+1}(\tau)$ and $b_\sigma^{t+n+1} = b_\tau^{t+n+1}$. Since σ and τ are simplices, their boundary cells are also lower-dimensional simplices, so by induction hypothesis, $a_B^t(\sigma) = a_B^t(\tau)$.

Let us consider the equality between the colours involving the upper adjacent cells. By expanding the definition we have:

$$\begin{aligned} & \{(b_{\delta_1}^{t+n+1}, b_{\delta_2}^{t+n+1}) \mid \delta_1 \in \mathcal{N}_\uparrow(\sigma), \delta_2 \in \mathcal{C}(\sigma, \delta_1)\} \\ &= \{(b_{\delta_1}^{t+n+1}, b_{\delta_2}^{t+n+1}) \mid \delta_1 \in \mathcal{N}_\uparrow(\tau), \delta_2 \in \mathcal{C}(\tau, \delta_1)\}. \end{aligned}$$

Generally, not all of these adjacencies involve simplices. For instance, a 2-simplex could incident to a general 3-cell. However, by Proposition 27 this equality must still hold if we restrict the multi-sets

to the colour of those cells that are simplices:

$$\begin{aligned} & \{(b_{\delta_1}^{t+n+1}, b_{\delta_2}^{t+n+1}) \mid \delta_1 \in \mathcal{N}_\uparrow(\sigma), \delta_2 \in \mathcal{C}(\sigma, \delta_1), \text{ and } \delta_1, \delta_2 \text{ are simplices}\} \\ &= \{(b_{\delta_1}^{t+n+1}, b_{\delta_2}^{t+n+1}) \mid \delta_1 \in \mathcal{N}_\uparrow(\tau), \delta_2 \in \mathcal{C}(\tau, \delta_1), \text{ and } \delta_1, \delta_2 \text{ are simplices}\}. \end{aligned}$$

These multi-sets, give exactly the upper adjacencies used by SWL for computing its colouring map c^t . Therefore, by the induction hypothesis, $a_\uparrow^t(\sigma) = a_\uparrow^t(\tau)$. Finally, this proves $a_\sigma^{t+1} = a_\tau^{t+1}$. \square

Corollary 30. CWL using a lifting attaching cells to cliques and rings/cycles is strictly more powerful than SWL using the clique complex lifting.

Proof. The second pair of graphs from Figure 6 cannot be distinguished by SWL because it has no cliques greater than 2, but it can be distinguished by CWL with the liftings above. \square

A.2 CW Network Proof

Proof of Theorem 17. Let c^t denote the colouring of CWL at iteration t and h^t the colouring (i.e. features) produced by a CW-Network as described in Section 4. Without loss of generality (Theorem 7), we use only boundary and upper adjacencies for both methods.

To show CWNs are at most as powerful as CWL, we must show $c^t \sqsubseteq h^t$. Again, we show this by induction. For a CWN with L layers we assume $h^t = h^L$ for all $t > L$. Let σ, τ be two cells with $c_\sigma^{t+1} = c_\tau^{t+1}$. Then, $c_\sigma^t = c_\tau^t$, $c_B^t(\sigma) = c_B^t(\tau)$ and $c_\uparrow^t(\sigma) = c_\uparrow^t(\tau)$. By the induction hypothesis, $h_\sigma^t = h_\tau^t$, $h_B^t(\sigma) = h_B^t(\tau)$ and $h_\uparrow^t(\sigma) = h_\uparrow^t(\tau)$.

If $t + 1 > L$, then $h_\sigma^{t+1} = h_\sigma^t = h_\tau^t = h_\tau^{t+1}$. Otherwise, h^{t+1} is given by Equation 1 involving the update function U , the aggregate function AGG and the message functions M_B, M_\uparrow . Given that the inputs passed to these functions are equal for σ and τ , $h_\sigma^{t+1} = h_\tau^{t+1}$.

We now prove that CWNs can be as powerful as CWL. Suppose the aggregation from Equation 1 is injective and the model is equipped with a number of layers L sufficient to guarantee the convergence of the colouring. Then, we show that $h^t \sqsubseteq c^t$. Let σ, τ be two cells with $h_\sigma^{t+1} = h_\tau^{t+1}$. Then, since the local aggregation is injective $h_\sigma^t = h_\tau^t$, $h_B^t(\sigma) = h_B^t(\tau)$ and $h_\uparrow^t(\sigma) = h_\uparrow^t(\tau)$. By the induction hypothesis, $c_\sigma^t = c_\tau^t$, $c_B^t(\sigma) = c_B^t(\tau)$ and $c_\uparrow^t(\sigma) = c_\uparrow^t(\tau)$. Finally, $c_\sigma^{t+1} = c_\tau^{t+1}$. \square

The consequence of this result is that CWNs inherit all the properties of CWL. We summarise these in the following Corollary.

Corollary 31. CWNs have the following properties:

1. They are at least as powerful as the WL test when using skeleton-preserving lifting transformations.
2. They are strictly more powerful than the WL test when using the lifting maps from Corollary 14.
3. They are not less powerful than 3-WL when using the lifting transformations from Theorem 15.
4. They are at least as powerful as MPSNs using the clique complex lifting [7] when using a lifting transformation whose output complexes have the clique complex as a subcomplex.
5. They are strictly more powerful than MPSNs when using a transformation attaching cells to cliques and rings/cycles. In particular, CWNs using rings are strictly more powerful than MPSNs using a lifting based on triangles (i.e. 2-simplices), since triangles are rings of size 3.

The latter point regarding triangles is important because Bodnar et al. [7] do not use simplices of dimension higher than two in practice.

A.3 Equations for Other Adjacencies

For completeness, we include in this section the equations for the co-boundary and lower adjacent messages.

$$m_C^{t+1}(\sigma) = \text{AGG}_{\tau \in \mathcal{C}(\sigma)}(M_C(h_\sigma^t, h_\tau^t)), \quad m_\downarrow^{t+1}(\sigma) = \text{AGG}_{\tau \in \mathcal{N}_\downarrow(\sigma), \delta \in \mathcal{B}(\sigma, \tau)}(M_\downarrow(h_\sigma^t, h_\tau^t, h_\delta^t)).$$

Together with the adjacencies described in the main text, the update rule takes the form

$$h_\sigma^{t+1} = U\left(h_\sigma^t, m_B^t(\sigma), m_C^t(\sigma), m_\downarrow^{t+1}(\sigma), m_\uparrow^{t+1}(\sigma)\right).$$

As mentioned before, even though these adjacencies are redundant from a colour refinement perspective when the others are used, they might still be employed in other combinations that preserve the expressive power of the test. Additionally, for certain applications, they might still encode important inductive biases.

A.4 Computational Analysis

Let X be a d -dimensional regular cell complex. For an arbitrary p -cell σ with boundary size k , the number of \uparrow -messages between the $(p - 1)$ -cells on its boundary is $2 * \binom{k}{2}$ and the number of \mathcal{B} -messages it receives is k . Let B_p be the maximum boundary size of a p -cell in X and S_p the number of p -cells. The computational complexity of our message passing scheme is thus $\mathcal{O}\left(\sum_{p=1}^d B_p S_p + 2 * \binom{B_p}{2} S_p\right)$. For instance, consider the skeleton-preserving lifting based on induced cycles. There, the dimension of the complex is $d = 2$ and we have $B_0 = 0$, $B_1 = 2$, and B_2 equals the size of the maximum induced cycle considered. For all practical purposes, we can consider d and B_p as fixed constants. Then the complexity can be rewritten as $\Theta\left(\sum_{p=1}^d S_p\right)$. This is optimal because the complexity is linear in the size of the cell complex and a linear time is required to read the cell complex.

Table 5: Wall-clock training and evaluation times on ZINC; mean, std over 10 runs (seconds).

Model	Training (Epoch)	Eval (Train)	Eval (Val)	Eval (Test)
GIN	4.582 ± 0.012	3.138 ± 0.071	0.310 ± 0.002	0.309 ± 0.001
GIN-small	3.737 ± 0.012	3.070 ± 0.058	0.304 ± 0.002	0.303 ± 0.003
CIN	10.828 ± 0.059	4.679 ± 0.051	0.470 ± 0.002	0.471 ± 0.003
CIN-small	7.082 ± 0.041	3.682 ± 0.056	0.365 ± 0.002	0.373 ± 0.030

Table 6: Wall-clock training and evaluation times on ZINC-FULL; mean, std over 10 runs (seconds).

Model	Training (Epoch)	Eval (Train)	Eval (Val)	Eval (Test)
GIN	106.268 ± 1.991	73.051 ± 1.742	7.874 ± 0.174	1.618 ± 0.039
GIN-small	87.581 ± 2.343	71.160 ± 1.865	7.714 ± 0.206	1.583 ± 0.037
CIN	249.334 ± 17.927	107.510 ± 1.637	11.759 ± 0.642	2.398 ± 0.028
CIN-small	163.282 ± 8.016	85.342 ± 2.637	9.251 ± 0.431	1.876 ± 0.044

In practice, we observed the empirical training runtimes to be contained, even on the largest benchmarks. We performed timing analyses on ZINC and ZINC-FULL, measuring the time required to complete one training epoch and a full performance evaluation on train, validation and test sets. We report the runtimes in Tables 5 and 6 the runtimes measured for our best performing CIN and CIN-small models and by GIN baselines with, approximately, the same number of parameters. We observe that the evaluation runtimes are relatively comparable to those of GIN models and that the difference decreases significantly at inference time (i.e. no backprop). The training runtimes are significantly reduced on CIN-small architectures, which always perform on-par or even better than state-of-the-art baselines, regardless of the imposed parameter budget (see Table 3). These experiments were run over an NVIDIA® Tesla V100 GPU device on an Amazon Web Services (AWS) Elastic Cloud (EC) 2 p3.16xlarge instance.

Other than the computational complexity of message passing we need to consider the (one-off) complexity pertaining the graph lifting procedures. Lifting procedures that are more likely to find immediate practical applications involve clique, cycle and induced cycle listing. For cliques, we refer readers to Bodnar et al. [7], where the authors report theoretical results regarding clique-listing complexity and the practical impact of employing specialised topological data analysis libraries.

As for cycle-based liftings, specialised cycle-listing algorithms exist. The algorithm in Birmelé et al. [6] is able to list all simple cycles in a graph in $\mathcal{O}(m + \sum_{c \in C(G)} |c|)$, where m is the number of edges, $C(G)$ is the set of simple cycles in graph G and $|c|$ is the size of the cycle. As for *induced* cycles, the algorithm presented in Ferreira et al. [25] has a listing time of $\tilde{\mathcal{O}}(m + nC)$, with n and C being the number of nodes and induced cycles, respectively. In certain types of graphs, a better complexity can be obtained. Chiba and Nishizeki [15] show linear time complexity to list triangles and 4-rings in planar graphs. This is very important because almost all molecules are planar in a graph-theoretic sense [64] as a direct consequence of the chemical implications of Kuratowski’s theorem [47]. However, we are not aware of any improved bounds for finding general induced cycles in planar graphs. Finally, we remind the reader that molecular rings can also be listed from the junction tree representation [26, 41], obtained by specialised molecular libraries such as RDKit [48].

Table 7: Wall-clock lifting times, mean and std over 5 runs (seconds).

Dataset ↓ / Processes →	Seq.	2	4	8	16	32
ZINC (12k)	320.27 ± 0.54	169.95 ± 0.32	84.90 ± 0.21	43.38 ± 0.07	23.17 ± 0.68	18.59 ± 0.68
Mol-HIV (41k)	1178.98 ± 3.90	635.58 ± 0.83	319.01 ± 0.40	164.26 ± 0.52	86.92 ± 0.77	60.62 ± 2.05
ZINC-FULL (250k)	6805.35 ± 16.50	3549.16 ± 7.73	1782.41 ± 3.84	918.38 ± 3.46	492.77 ± 6.13	383.92 ± 3.30

In our experiments, we implemented a lifting procedure based on the *generic* substructure matching algorithm exposed by the graph-tool Python library, which internally employs VF2 [18] to perform subgraph isomorphism. Noticing that the lifting procedure is embarrassingly parallel w.r.t. the independent graphs in a dataset, we easily parallelised the procedure via Python’s Joblib library. On molecular benchmarks we observed the effective time required by preprocessing routines to always be modest compared to the training times. In Table 7 we report the wall clock runtimes, averaged over 5 runs, to lift all the graphs in the largest datasets amongst our benchmarks: ZINC, Mol-HIV and ZINC-FULL. The analysis has been conducted considering rings up to size 18 and by varying the number of parallel processing jobs on a server with an Intel® Xeon E5-2686 v4 processor with 64 vCPUs. It is possible to observe that the empirical lifting runtime scales linearly with the number of jobs in the range [1, 16], and that such a simple parallelisation scheme dramatically reduces the preprocessing time on all datasets. When employing 32 parallel jobs, less than 19 seconds are required to preproceess the whole ZINC dataset, only 1 minute is required for Mol-HIV, and we needed slightly more than 6 minutes to lift all the 250k graphs in ZINC-FULL. We remark once more that these experiments have been conducted with a *generic* subgraph matching algorithm, and that even more parsimonious computation would be possible by using optimised ring-listing routines.

B Symmetries

In line with a recent effort in Geometric Deep Learning to understand different models through the lens of symmetry [11], we aim here to give a description of the underlying equivariance properties of CW Networks.

First, let us define the following matrix representation of the boundary relation from Definition 3.

Definition 32. Let X be a regular cell complex with S_k denoting the number of cells in dimension k . The k -th unsigned boundary matrix $\mathbf{B}_k \in \mathbb{R}^{S_{k-1} \times S_k}$ of X is given by $B_k(i, j) = 1$ if $\sigma_i \prec \sigma_j$ and 0, otherwise.

Let X be a regular cell complex of dimension n with boundary matrices $\mathbf{B} = (\mathbf{B}_1, \dots, \mathbf{B}_n)$ and feature matrices $\mathbf{X} = (\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_n)$ for the cells of different dimensions. Additionally, consider a sequence of permutation matrices $\mathbf{P} = (\mathbf{P}_0, \dots, \mathbf{P}_n)$. Denote by $\mathbf{PX} = (\mathbf{P}_0\mathbf{X}_0, \dots, \mathbf{P}_n\mathbf{X}_n)$ and $\mathbf{PBP}^T = (\mathbf{P}_0\mathbf{B}_1\mathbf{P}_1^T, \dots, \mathbf{P}_{n-1}\mathbf{B}_n\mathbf{P}_n^T)$.

Definition 33. A function f mapping $(\mathbf{X}, \mathbf{B}) \mapsto \mathbf{X}' = (\mathbf{X}'_0, \dots, \mathbf{X}'_n)$ with the property that $\mathbf{P}f(\mathbf{X}, \mathbf{B}) = f(\mathbf{PX}, \mathbf{PBP}^T)$ for any \mathbf{P} is called cell permutation equivariant.

Proof of Theorem 18. Definition 33 is similar to the (simplex) permutation equivariance definition from Bodnar et al. [7], with the subtle difference that the boundary matrices now have a more flexible structure in the case of cell complexes. The high-level idea is to see that all the adjacency matrices used by CWNs (i.e. $\mathbf{B}_k, \mathbf{B}_{k+1}, \mathbf{B}_k^\top \mathbf{B}_k, \mathbf{B}_{k+1} \mathbf{B}_{k+1}^\top$) are permuted accordingly by the permutation

matrices in \mathbf{P} . Therefore, a CWNs layers computes the same function up to a permutation of the cells. The proof follows a similar logic to the one in Bodnar et al. [7] for simplicial networks, and we refer the reader to their work for a detailed proof. \square

It is common in algebraic topology and differential geometry to equip the incidence relation $\sigma \prec \tau$ with additional structure that makes it a signed incidence relation.

Definition 34 (Hansen and Ghrist [34]). *A signed incidence relation on P_X is a map $[\cdot : \cdot]: P_X \times P_X \rightarrow \{0, \pm 1\}$ with the properties:*

1. If $[\sigma : \tau] \neq 0$, then $\sigma \prec \tau$.
2. For any $\sigma \leq \tau$, $\sum_{\gamma \in P_x} [\sigma : \gamma][\gamma : \tau] = 0$.

This signed incidence relation can be encoded by the signed incidence (boundary) matrices of X . We define these below:

Definition 35. *Let X be a regular cell complex with a signed incidence relation $[\cdot : \cdot]$. Let S_k denote the number of cells in dimension k . The k -th signed boundary matrix $\mathbf{B}_k \in \mathbb{R}^{S_{k-1} \times S_k}$ of X is given by $B_k(i, j) = [\sigma_i : \sigma_j]$.*

The difference with respect to the unsigned boundary matrices is that the non-zero values of the matrix can be ± 1 , not just 1. This can be used to define a notion of orientation equivariance for CW Networks. This ensures that when changing the orientation of the cell complex X (i.e. changing $[\cdot : \cdot]$) one computes the same function up to that change in orientation.

Let X be a regular cell complex of dimension n described by the *signed* boundary matrices $\mathbf{B} = (\mathbf{B}_1, \dots, \mathbf{B}_n)$ and feature matrices $\mathbf{X} = (\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_n)$ for the cells of different dimensions. Additionally, consider a sequence of diagonal matrices $\mathbf{T} = (\mathbf{T}_0, \dots, \mathbf{T}_n)$ with values in ± 1 . Additionally, let $\mathbf{T}_0 = \mathbf{I}$. Denote by $\mathbf{TX} = (\mathbf{T}_0 \mathbf{X}_0, \dots, \mathbf{T}_n \mathbf{X}_n)$ and $\mathbf{TBT} = (\mathbf{T}_0 \mathbf{B}_1 \mathbf{T}_0, \dots, \mathbf{T}_n \mathbf{B}_n \mathbf{T}_n)$.

Definition 36. *A function f mapping $(\mathbf{X}, \mathbf{B}) \mapsto \mathbf{X}' = (\mathbf{X}'_0, \dots, \mathbf{X}'_n)$ with the property that $\mathbf{Tf}(\mathbf{X}, \mathbf{B}) = f(\mathbf{TX}, \mathbf{TBT})$ for any \mathbf{T} is called orientation equivariant.*

Making CWNs orientation equivariant requires imposing additional constraints on the layers of the model. This proceeds similarly to MPSNs [7]. Since applications involving oriented simplicial complexes are out of the scope of this work, we refer the reader to Bodnar et al. [7] for an intuition of how this can be extended to cell complexes.

C Sheaves, Laplacians and Convolutions

It is useful on cell complexes to derive a Laplacian operator based on cellular sheaves [34], since many interesting Laplacians, such as the (normalised) graph Laplacian [16], the Hodge Laplacian [61] and the connection Laplacian [65] can be obtained as particular cases. Intuitively, a cellular sheaf is a construction that assigns a vector space to each cell in the complex and a (linear) map for each face relation in the complex $\sigma \leq \tau$. Additionally, these linear maps must satisfy some compositionality constraints imposed by the structure of P_X .

C.1 Sheaf Laplacian

Definition 37. *Let (X, P_X) be a regular cell complex, and denote by Hilb_K the class of Hilbert spaces over a field K . A **weighted cellular sheaf** \mathcal{F} is given by the assignment*

$$\begin{aligned} \mathcal{F}: P_X &\rightarrow \text{Hilb}_K \\ \sigma &\mapsto \mathcal{F}(\sigma) \end{aligned}$$

together with a bounded linear map $\mathcal{F}_{\sigma \leq \tau}: \mathcal{F}(\sigma) \rightarrow \mathcal{F}(\tau)$ for any $\sigma \leq \tau$.

This data satisfies that $\mathcal{F}_{\sigma \leq \sigma} = \text{id}$ for all $\sigma \in P_X$ and $\mathcal{F}_{\sigma \leq \omega} = \mathcal{F}_{\tau \leq \omega} \circ \mathcal{F}_{\sigma \leq \tau}$ whenever $\sigma \leq \tau \leq \omega$.

Given a weighted cellular sheaf \mathcal{F} , we define a chain complex as follows. For each $k = 0, 1, \dots$ we set

$$C^k(X; \mathcal{F}) = \bigoplus_{\dim(\sigma)=k} \mathcal{F}(\sigma).$$

Further, we define coboundary maps $\delta^k: C^k(X; \mathcal{F}) \rightarrow C^{k+1}(X; \mathcal{F})$ by

$$\delta^k(x)_\tau = \sum_{\dim(\sigma)=k} [\sigma : \tau] \mathcal{F}_{\sigma \leq \tau}(x_\sigma),$$

where $[\cdot : \cdot]: P_X \times P_X \rightarrow \{0, \pm 1\}$ is a signed incidence relation (see Definition 34).

Given Hilbert spaces V and W and a bounded linear map $T: V \rightarrow W$, the adjoint of T is the unique bounded linear map $T^*: W \rightarrow V$ satisfying that for all $v \in V$ and all $w \in W$:

$$\langle w, T v \rangle = \langle T^* w, v \rangle.$$

Definition 38. Let $C^\bullet = C^0 \rightarrow C^1 \rightarrow \dots$ be a chain complex of Hilbert spaces. The **Hodge Laplacian** is the graded linear map defined in degree k as $\Delta^k: C^k \rightarrow C^k$ with $\Delta^k = (\delta^k)^* \delta^k + \delta^{k-1} (\delta^{k-1})^*$. When $C^\bullet = C^0 \rightarrow C^1 \rightarrow \dots$ is the complex of cochains of a weighted cellular sheaf \mathcal{F} , the Hodge Laplacian is called the **sheaf Laplacian** of X .

In particular, the Hodge Laplacian of a cell complex can be obtained by considering the *constant weighted cellular sheaf* with a standard inner product. That is the cellular sheaf where $\mathcal{F}(\sigma) = \mathbb{R}$ and the restriction maps $\mathcal{F}_{\sigma \leq \tau} = \text{id}$. A normalised version of it can also be obtained by carefully adjusting the inner products associated with each $\mathcal{F}(\sigma)$. This normalisation is always possible for finite cell complexes (see Hansen and Ghrist [34] for details). This is very useful because finding normalised versions of Hodge Laplacians is not trivial and even on simplicial complexes [61], the process of constructing one can be quite involved.

C.2 Convolutional Operators

One can use the general sheaf Laplacian to define linear, local diffusion operators which, in the GNN literature, are broadly addressed as ‘convolutional’. Diffusion operators built from the standard graph Laplacian have been employed in several graph neural network architectures [21, 45]. Recent works [12, 24] have introduced convolutional operators on SCs by employing the Hodge Laplacian [61], interpreted as a generalisation of the graph Laplacian. As for cell complexes, here we focus, for simplicity, on the case of a constant sheaf with a standard inner product in \mathbb{R}^n . Then, the matrix representations of δ^k and $(\delta^k)^*$ are the signed incidence matrices \mathbf{B}_k^T and \mathbf{B}_k , respectively. Therefore, the Hodge Laplacian can be written in matrix form as

$$\mathbf{L}_k = \mathbf{B}_k^T \mathbf{B}_k + \mathbf{B}_{k+1} \mathbf{B}_{k+1}^T.$$

A convenient way to define a convolutional operator on cochains is by designing a learnable filter parameterised as a polynomial of the Hodge Laplacian. This approach has been already adopted on graphs using the standard graph Laplacian [21] or more general sheaf Laplacians [35], and also on simplicial complexes [24]. The advantage of this approach is that of retaining a connection with spectral constructions [21, 24] while not requiring any explicit diagonalisation of the operator itself. A polynomial convolutional filter of this kind, when applied to the p -cells of a d -cell complex, would take the form

$$H^{t+1} = \psi \left(\sum_{r=0}^R \mathbf{L}_p^r H^t W_r^{t+1} \right) = \psi \left(H^t W_0^{t+1} + \sum_{r=1}^R \mathbf{L}_p^r H^t W_r^{t+1} \right). \quad (3)$$

where H^t is a matrix gathering p -cell representations at layer t , W_r^{t+1} are learnable parameters, and ψ summarises the application of a bias term and a non-linearity.

Proof of Theorem 18. While the structure of the boundary matrices is more flexible in a cell complex than in a simplicial complex, algebraically, the proof is very similar to the proof showing MPSNs generalise simplicial convolutions in Bodnar et al. [7]. We offer here a high-level view of the proof and refer the reader to Appendix C of their paper for a detailed version.

For a generic p -cell σ , and $r > 0$, the application of the r -power of the Hodge Laplacian effectively induces an information flow from a generalised notion of r -upper and r -lower adjacent p -cells, i.e. p -cells τ such that there exists a sequence of upper- (respectively, lower-) adjacent p -cells $[\gamma_0, \gamma_1, \dots, \gamma_r]$ such that $\gamma_0 = \sigma, \gamma_r = \tau$.

Therefore, the convolution described above is easily interpreted in terms of a cellular message passing scheme which only exchanges \uparrow - and \downarrow -messages. Intuitively, the upper- and lower- message functions would share their parameters W_r^{t+1} and compute messages by linearly projecting the representations of upper- and lower-adjacent cells (ignoring any information in shared (co)boudaries). Such messages would then be aggregated by summation into an overall message, taken as input by an update function parameterised by ψ and W_0^{t+1} . A formal derivation of how the equation (3) is rewritten in terms of cellular message passing would closely follow the one provided in Bodnar et al. [7] for SCs, and we therefore refer readers to Section C of such work. \square

Normalised versions of the aforementioned Hodge Laplacian can be used to design a model in the spirit of the popular Graph Convolutional Network of Kipf and Welling [45]. To this aim, one could resort to normalised sheaves as suggested in [34]. Additionally, one could explicitly make use of the (co)boundary operators defined in Section B to let information flow from lower- and higher-dimensional cells contained in cell (co)boundaries, effectively extending the Simplicial Convolutional Networks recently introduced in Bunch et al. [12]. We defer these research directions to future developments of this work.

D Experimental details and additional results

D.1 Used Code Assets

The model has been implemented in PyTorch [57] and by building on top of the PyTorch Geometric library [27]. Lifting operations use the graph-tool⁴ Python library and are parallelised via Joblib⁵. PyTorch, NumPy, SciPy and Joblib are made available under the BSD license, Matplotlib under the PSF license, graph-tool under the GNU LGPL v3 license. PyTorch Geometric is made available under the MIT license.

D.2 Used Computer Resources

All experiments were run on NVIDIA® GPUs. Experiments on **SR**, **Mol-HIV** and molecular **TUDatasets** were run on Tesla V100 GPUs with 5,120 CUDA cores and 16GB GPU memory on a p3.16xlarge Amazon Web Services (AWS) Elastic Cloud (EC) 2 instance. Experiments on the social **TUDatasets** were run on the same GPU devices but with 32GB HBM2 memory mounted on an HPC cluster. All remaining experiments, that is **CSL**, **RingTransfer** and **ZINC**, were run on a machine with TITAN Xp GPUs with 12GB GPU memory and an Intel® Xeon® CPU E5-2630 v4 @ 2.20GHz CPU.

D.3 Model

In all cases, we apply our model to the 2-dimensional cell complexes obtained by ring-lifting the original graphs, i.e. we consider nodes and edges as 0- and 1-cells, and each induced cycle of size up to k as a 2-cell. 0-cell are always endowed with the original node features or learnt node embeddings, if the benchmark prescribes so. The way higher dimensional cells are assigned features depend on the specific benchmark.

Throughout all experiments, we employ cellular message passing layers which update the representation of p -cell σ as follows:

$$\begin{aligned} h_\sigma^{t+1} = & \text{MLP}_{U,p}^t \left(\text{MLP}_{B,p}^t \left((1 + \epsilon_B) h_\sigma^t + \sum_{\tau \in \mathcal{B}(\sigma)} h_\tau^t \right) \parallel \right. \\ & \left. \text{MLP}_{\uparrow,p}^t \left((1 + \epsilon_\uparrow) h_\sigma^t + \sum_{\tau \in \mathcal{N}_\uparrow(\sigma), \delta \in \mathcal{C}(\sigma, \tau)} \text{MLP}_{M,p}^t(h_\tau^t \parallel h_\delta^t) \right) \right) \end{aligned} \quad (4)$$

Here, \parallel indicates concatenation, $\text{MLP}_{B,p}^t$, $\text{MLP}_{\uparrow,p}^t$ are 2-Layer Perceptrons and $\text{MLP}_{U,p}^t$, $\text{MLP}_{M,p}^t$ consist of a dense layer followed by a non-linearity. We neglect messages from cofaces and down-adjacent cells consistently with Theorem 7. We name an architecture which stacks L layers of this

⁴<https://graph-tool.skewed.de/>

⁵<https://joblib.readthedocs.io/en/latest/>

form as ‘Cell Isomorphism Network’ (CIN). Readout operations are performed as follows. First, for $p \in 0, 1, 2$, we compute the joint representation h_p of the cells at dimension p by applying a mean or sum readout operation. Then, for complex \mathcal{K} , we compute an overall representation $h_{\mathcal{K}} = \sum_{p=0,1,2} \text{MLP}_{R,p}(h_p)$, where each $\text{MLP}_{R,p}$ is parameterised as a single dense layer followed by a non-linearity. Complex-wise predictions are obtained by a final dense layer preceded by dropout. All MLP layers internally apply Batch Normalization [40] and ReLU activations, unless otherwise specified. All training procedures are performed with the Adam optimiser [44].

D.4 Additional experimental details

CSL Each of the 150 4-regular graphs in the CSL dataset comprises $N = 48$ nodes and is characterized by *skip number* parameter $C \in \mathcal{C} = \{2, 3, 4, 5, 6, 9, 11, 12, 13, 16\}$. Parameters N and C determine the isomorphism class $\mathcal{G}_{N,C}$ of each graph, which we seek to predict. The number of possible classes is $|\mathcal{C}| = 10$. We employ the same stratified dataset folds in Dwivedi et al. [23]. Consistently with the adopted reference procedure, 0-cells share the same learnt embedding, while 1- and 2-cells are endowed with the sum of the embeddings of the included 0-cells. As for the optimisation procedure, we set the batch size to 12 and the initial learning rate of 5e-4, which is halved whenever the validation performance does not improve after a patience value of 20. The training is early stopped as soon as it falls below 1e-6, at which step we measure the model test accuracy. The size of hidden layers in our model is set to 160 and we stack 3 cellular message passing layers. In this benchmark, we replace Batch Normalisation with Layer Normalization [4], as the former was observed to produce instabilities in the optimisation procedure. At each dimension, cell embeddings are readout via averaging.

SR We initialise the cell complexes associated with SR graph by populating 0-cells with constant, scalar, unitary signal, and 1- and 2-dimensional cells with the sum of the contained 0-cells. Complexes are embedded in a 16-dimensional space and, coherently with Bodnar et al. [7] and Bouritsas et al. [9], if the L_2 -distance between the embeddings of two complexes is larger than $\epsilon = 0.01$, we deem the corresponding graphs to be non-isomorphic. In accordance with Bodnar et al. [7], we run an untrained CIN model with 5 cellular message passing layers, whose hidden layers comprise 16 units. At each dimension, cell embeddings are readout via summation. As the number of induced cycles of a certain size may be enough to tell apart non-isomorphic SR graphs (see Table 4), an MLP with sum readouts represents a strong baseline, which we additionally run. Such a model applies non-linear dense layers at each cell dimension, and then performs readout operations as in CIN. We set the size of hidden layers to 256, while the final complex embeddings are embedded in a 16-dimensional space as in our model. Both approaches are equipped with ELU nonlinearities [17].

RingTransfer This benchmark dataset comprises 5000 training graphs. Each graph is randomly associated with one of the 5 independent labels, which are also assigned as node features to **source** nodes. Labels are uniformly represented. On this benchmark we run a CIN model with 3 stacked message passing layers, independently on the ring size. The hidden size of the layers is set to 64 and we do not apply Batch Normalisation. Differently than in the other benchmarks, we do not need to perform readout operations to compute complex-wise embeddings; instead, we simply take the representation of the 0-cell corresponding to node **target** at the last layer of the architecture and use it to predict the label of **source**. GIN models have always $\lfloor \frac{k}{2} \rfloor$ standard message passing layers with hidden size 64. The models are trained with an initial learning rate of 10^{-3} , decayed by a factor of 0.5 and a patience of 5 epochs. The training is stopped when the learning rate drops below 10^{-5} .

TUD Amongst the datasets from this benchmarking suite: the task in **MUTAG** is to recognise mutagenic molecular compounds for potentially marketable drug [43, 59]; the one in **PTC** is to recognise the chemical compounds according to carcinogenicity on rodents [37, 46]; **PROTEINS** is about to categorising proteins into enzyme and non-enzyme structures [8, 22]; **NCI1** and **NCI109** deal with identifying chemical compounds against the activity of non-small lung cancer and ovarian cancer cells, respectively [68]; **REDDIT-BINARY** or **RDT-B** is a social network dataset where the task is to predict whether a graph belongs to a question-answer-based community or a discussion-based community. On these datasets, we followed the approach in Xu et al. [72], which prescribes to run a 10-fold cross-validation procedure and report the maximum of the average validation accuracy across folds. Consistently with such work, we train our model starting from an initial learning

Table 8: Hyperparameter configurations on TUDatasets.

Hyperparameter	MUTAG	PTC	PROTEINS	NCI1	NCI109	IMDB-B	IMDB-M	RDT-B
Batch Size	32	32	128	32	32	128	128	32
Initial LR	0.01	0.01	0.01	0.001	0.001	0.001	0.0005	0.001
LR Dec. Steps	20	50	20	20	20	50	20	50
LR Dec. Strength	0.5	0.9	0.5	0.5	0.5	0.5	0.5	0.5
Hidden Dim.	64	16	32	16	64	16	64	64
Drop. Rate	0.5	0.0	0.0	0.5	0.0	0.0	0.5	0.0
Drop. Pos.	cell read.	comp read.						
Initialisation	sum	mean						
Cobound. in \uparrow -msg	N	N	Y	Y	Y	N	N	N
Num. Layers	4	4	3	4	4	4	4	4

rate which is decayed after a fixed amount of epochs and we apply cell-readout operations on the multiscale representations obtained by a Jumping Knowledge scheme [71] by performing averaging or summation depending on the dataset, still in accordance with Xu et al. [72]. We ran a grid-search to tune batch size, hidden dimension, dropout rate, initial learning rate along with its decay steps and strengths, feature initialisation strategy of higher-dimensional cells (mean vs. sum), inclusion of coboundary features in \uparrow -messages, number of layers and the dropout position (immediately after readout on cells (“cell read.”) or the final readout on the complex (“comp read.”)). We report the hyperparameter configurations in Table 8. We finally report that we did not employ Batch Normalization layers in **RDT-B** since they were observed to produce severe instabilities in the training procedure.

ZINC The ZINC benchmarks dataset have been constructed by the ZINC database provided by the Irwin and Shoichet Laboratories in the Department of Pharmaceutical Chemistry at the University of California, San Francisco (UCSF) [66]. Each graph represents a molecule, with node features indicating the atom type and edge features the type of chemical bond between two atoms. Graph targets correspond to the penalised water-octanol partition coefficient – logP [32]. In these experiments, rings up to size $k = 18$ are mapped to 2-cells, and are assigned feature values as the sum of the learnable atom embeddings for the included 0-cells (nodes). 1-cells are assigned learnable bond embeddings if edge-features are considered, otherwise we apply the same policy employed for 2-cells. We employ the same predefined training, validation and test splits as in Dwivedi et al. [23], and train our model by minimising the Mean Absolute Error (MAE) loss on the train targets. As prescribed by the benchmark, the optimisation procedure employs a batch size of 128 and a dynamic learning rate which starts from 10^{-3} and is halved whenever the validation loss does not improve after a patience value we set to 20. The training is early stopped as soon as it falls below 10^{-5} . We repeat the training with 10 different weight initialisations and report the mean of the test MAEs at the time of early stopping. In accordance with the best performing baselines, our CIN model does not use any dropout, and stacks 4 message passing layers with hidden size 128. In order to enforce the parameter budget we reduce the size of hidden layers to 48 and only perform 2 message passing layers. At each dimension, cell embeddings are readout via summation.

Mol-HIV This dataset comprises 41127 molecular graphs associated with a binary label representing their capacity to inhibit HIV replication. The benchmark provides predefined train, validation and test sets based on the “scaffold splitting” procedure, which separates molecules based on their two-dimensional structural frameworks [38]. As in **ZINC**, graphs are attributed at the level of nodes and edges, and we directly employ the atom and bond embedding layers provided by the benchmarking platform⁶ to populate 0- and 1-dimensional cells. Rings of size up to $k = 6$ are considered as 2-cells, and are endowed with feature vectors with the same procedure as in **ZINC**. The value $k = 6$ has been chosen from the pool of values $\{6, 8, 18\}$ as it yielded the highest validation performance. The architecture hyperparameters are directly replicated from the HIMP model in Fey et al. [26]: 2 message passing layers, dropout rate of 0.5 applied after each layer, 64 as size of hidden layers, constant learning rate of 10^{-4} , batch size of 128. We train our model for 150 epochs. The small CIN model is obtained by simply reducing the size of hidden layers to 48. At each dimension, cell embeddings are readout via averaging.

⁶https://github.com/snap-stanford/ogb/blob/master/ogb/graphproppred/mol_encoder.py

D.5 Ablation study on ZINC

Table 9: ZINC Ablation with edge features. The ablation shows the benefits of integrating rings into the message passing procedure.

Method	MAE
GatedGCN [10]	0.363±0.009
GIN [72]	0.252±0.014
PNA [19]	0.188±0.004
DGN [5]	0.168±0.003
HIMP [26]	0.151±0.006
GSN [9]	0.108±0.018
GIN-E Custom	0.196±0.007
CIN No-Rings small	0.174±0.006
CIN No-Rings	0.159±0.007
CIN-small	0.094±0.004
CIN	0.080±0.004

this model updates edge representations at each layer as a function of the present representations and those of the incident nodes (0-cells). As an additional confirmation of this hypothesis, we implemented an architecture which replicates the same structure as “CIN No-Rings”, but replaces cellular message passing with GIN-E layers [39]. These layers extend the message passing scheme in GIN by accounting for edge features. We refer to this model as “GIN-E Custom”. Contrary to CIN, it does not update edge representations and performs readout only at the node level. As expected, we observed that “GIN-E Custom” is outperformed by all our models, including, in particular, “CIN No-Rings”.

We end this section by reporting the results of an ablation study we conducted on the ZINC dataset to appreciate the contribution of including rings. In Table 9 we show the average test MAE for two additional CIN models: “CIN No-Rings small” and “CIN No-Rings”, which differ from their original counterparts in that they neglect 2-cells when performing message passing. In these experiments we always make use of edge features and use the same hyperparameters as our original CIN model. In line with our expectations, we observe a decrease in the overall performance of both versions. They are outperformed by the GSN [9] and HIMP [26] models, which either include structural information from cycle isomorphism counting (GSN) or additionally perform message passing on the Junction Tree representation of molecules (where rings are considered as nodes). At the same time, we observe “CIN No-Rings” still outperforms all other ring-agnostic baselines. We attribute such strong performance to the more natural and richer modelling of edge signals (1-cells):