Weisfeiler and Lehman Go Cellular: CW Networks

Anonymous Author(s)
Affiliation
Address
email

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. [53] and Morris et al. [37] the local neighbourhood aggregations used by GNNs are at most as powerful as the Weisfeiler-Lehman (WL) test [50] in distinguishing non-isomorphic graphs. Therefore, these local neighbourhood aggregators fail to detect certain higer-order meso-scale structures such as cliques or induced cycles, 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 incurred by this, deep GNNs typically come with additional problems such as over-smoothing [33] and over-squashing [1] of the node representations.

To address these problems, we propose a novel message passing procedure based on (regular) cell complexes, also known as CW complexes[1], topological objects that form the building block of algebraic topology [26]. When paired with a theoretically-justified "lifting" transformation augmenting the graph with higher-dimensional constructs called "cells", our method results in a multi-dimensional or hierarchical message passing procedure over the input graph. Our approach generalises and subsumes the recently proposed Message Passing Simplicial Networks (MPSNs) [4], which operate on simplicial complexes (SCs), topological generalisations of graphs. However, SCs

[1] We 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 [51].

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

have a rigid combinatorial structure that significantly limits the range of lifting transformations one could use to meaningfully modulate the graph message 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 are strictly more powerful than MPSNs 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

Definition 1. *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 neighbordhood of x that intersects finitely many cells.*
2. *For all σ, τ we have that $X_\tau \cap \overline{X_\sigma} \neq \emptyset$ iff $X_\tau \subseteq \overline{X_\sigma}$. Here $\overline{X_\sigma}$ is the closure of a cell.*
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 $\mathbb{R}^{n_\sigma}$ to $\overline{X_\sigma}$ such that the restriction of ϕ to the interior of the ball is a homeomorphism onto X_σ.*

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.

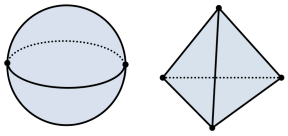


Figure 1: Examples of regular cell complexes: a sphere and an empty tetrahedron. The latter is also a simplicial complex.

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 their endpoints to the vertices. 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 to any simple cycle in the (multi) graph previously built (see Figure 2). While we are generally not concerned with dimensions above two in this work, this can be further generalised.

To illustrate this, 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 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. *(k-skeleton) 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 will be useful for referring for 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 the following (binary) relation called the boundary relation, whose reflexive 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 (Boundary relation)**.** *We have the boundary relation $\sigma \prec \tau$ iff $\sigma < \tau$ and there is no cell δ such that $\sigma < \delta < \tau$.*

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 σ, 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 $\mathcal{N}_{\downarrow}(\sigma) = \{\tau \mid \exists \delta$ such that $\delta \prec \sigma$ 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 $\mathcal{N}_{\uparrow}(\sigma) = \{\tau \mid \exists \delta$ such that $\sigma \prec \delta$ 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 distinguish non-isomorphic graphs. These insights will be used in Section 4 to specify a message passing model that can 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 the complex. Define $\mathcal{B}(\sigma, \tau) := \mathcal{B}(\sigma) \cap \mathcal{B}(\tau)$ and $\mathcal{C}(\sigma, \tau) := \mathcal{C}(\sigma) \cap \mathcal{C}(\tau)$. For a given cell σ, we define the following multi-sets of colours*

1. *The colours of the boundary cells of σ: $c_{\mathcal{B}}(\sigma) = \{\!\{c_\tau \mid \tau \in \mathcal{B}(\sigma)\}\!\}$.*
2. *The colours of the co-boundary cells of σ: $c_{\mathcal{C}}(\sigma) = \{\!\{c_\tau \mid \tau \in \mathcal{C}(\sigma)\}\!\}$.*
3. *The lower adjacent colours of σ: $c_{\downarrow}(\sigma) = \{\!\{(c_\tau, c_\delta) \mid \tau \in \mathcal{N}_{\downarrow}(\sigma)$ and $\delta \in \mathcal{B}(\sigma, \tau)\}\!\}$.*
4. *The upper adjacent colours of σ: $c_{\uparrow}(\sigma) = \{\!\{(c_\tau, c_\delta) \mid \tau \in \mathcal{N}_{\uparrow}(\sigma)$ and $\delta \in \mathcal{C}(\sigma, \tau)\}\!\}$.*

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 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.

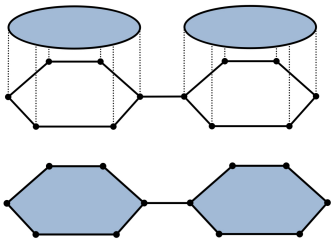


Figure 2: Closed 2D disks are glued to the boundary of the rings present in the graph (top). The result is a 2D regular cell complex (bottom).

Cellular WL (CWL) We consider CWL, a colour refinement scheme for cell complexes that generalises the Simplicial WL [4] and WL tests. We use c^t_σ 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^t_σ of cell σ at iteration t, we compute the colour of cell σ at the next iteration c^{t+1}_σ by injectively mapping the multi-sets of colours belonging to the adjacent cells of σ using a fixed HASH function. The algorithm stops when the colouring converges.

$$c^{t+1}_\sigma = \mathrm{HASH}\big(c^t_\sigma, c^t_{\mathcal{B}}(\sigma), c^t_{\mathcal{C}}(\sigma), c^t_{\downarrow}(\sigma), c^t_{\uparrow}(\sigma)\big).$$

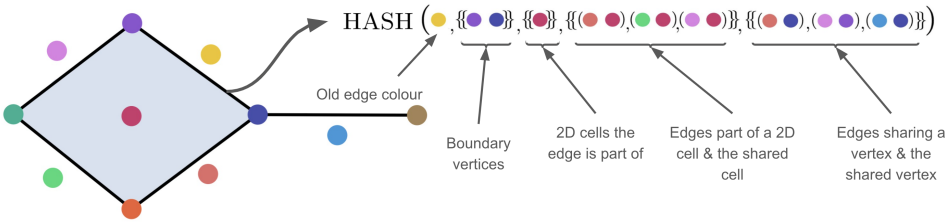


Figure 3: The CWL colouring procedure for the yellow edge of the cell complex. All cells have been assigned unique colours to aid the visualisation of the adjacencies. Note that the yellow edge aggregates long-range information from the light green edge.

Two cell complexes are considered non-isomorphic if their colour histograms are different. If the colour histograms are the same, the cell complexes might be isomorphic, but the test is inconclusive.

First, we state the following theorem from Bodnar et al. [4] 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 ([4])**.** *SWL without coface 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 the 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 coface 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 refinment 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 (Cellular lifting map)**.** *We call a (cellular) lifting a map $f : G \mapsto X$ from the space of graphs to the space of regular cell complexes 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 graphs in the cell complex space is equivalent to testing their isomorphism in the graph space. This would not be the case if two non-isomoprhic graphs are 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.*

The clique complex lifting map from Example 9 has been used by Bodnar et al. [4] to show that SWL is strictly more powerful than WL. We restate this result:

Theorem 10 ([4])**.** *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 (Skeleton-preserving lifting map)**.** *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 gluing 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(f) is at least as powerful as WL in distinguishing non-isomorphic graphs.*

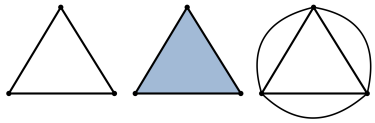


Figure 4: A graph, its clique complex and the graph with duplicated edges. The second is skeleton-preserving, while the last is not.

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 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 by no-means a complete list. In fact, due to the inability of WL to count connected induced subgraphs involving more than 3 nodes [10], it is likely that most non-trivial skeleton-preserving lifting maps are strictly more powerful than WL, since all the higher-dimensional cells involve such patterns. We provide more evidence in support of this conjecture in Appendix A.

Finally, 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.*

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.

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

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

The first specifies messages from atoms to bonds and from bonds to rings, respectively. 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 bond between them. 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:

$$h_\sigma^{t+1} = U\Big(h_\sigma^t, m_{\mathcal{B}}^t(\sigma), m_{\uparrow}^{t+1}(\sigma)\Big). \tag{1}$$

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:

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

Expressivity Naturally, the ability of CWNs to distinguish non-isomorphic graphs 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 [12, 53] are known to exist and can be directly employed in our model.

Theorem 16. *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 $\boldsymbol{A}$ and feature matrix $\boldsymbol{X}$, a function f is (node) permutation equivariant if $\boldsymbol{P}f(\boldsymbol{A}, \boldsymbol{X}) = f(\boldsymbol{P}\boldsymbol{A}\boldsymbol{P}^T, \boldsymbol{P}\boldsymbol{X})$, for any permutation matrix $\boldsymbol{P}$. 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 state and prove the following result formally in Appendix B.

Theorem 17. *CW Network layers are (cell) permutation equivariant.*

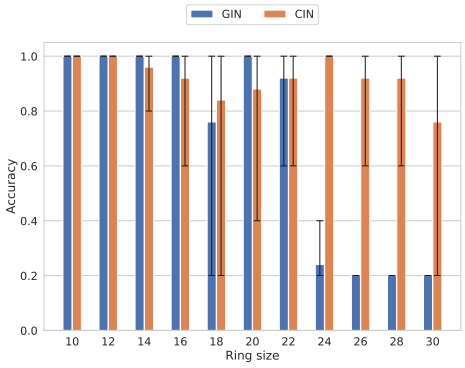
Long-Range Interactions Several graph-related tasks require the ability to capture long-range interactions between nodes. For instance, certain molecular properties depend on atoms placed on the opposite sides of a ring [21, 42]. As a consequence of the coupling between the input and computational graphs, 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 [33] 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.

CWNs as generalised convolutions Our message passing scheme can be seen a (non-linear) generalisation of linear diffusion operators on cell complexes. Recent works [8, 17] have introduced convolutional operators on SCs by employing the Hodge Laplacian [44], a generalisation of the graph Laplacian. By leveraging on the cellular Sheaf Laplacian [25], 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.

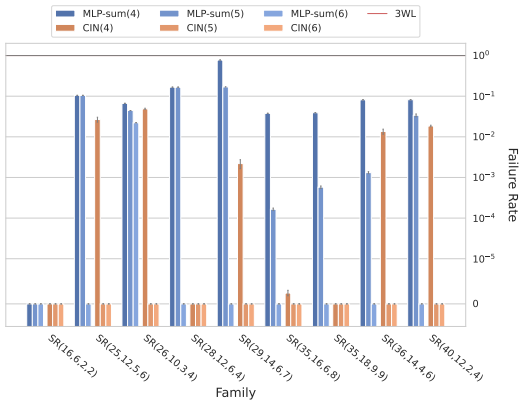
Computational Complexity For all practical purposes, suppose we consider only cells of an arbitrary constant maximum dimension and constant maximum boundary size. Then, the computational complexity of our message passing scheme using the adjacencies from Theorem 7 can be shown to be linear in the size 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 [18, 28], obtained by specialised libraries such as RDKit [32]. 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.

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 [53]. We name our architecture "Cell Isomorphism Network" (CIN). 0-cells are always endowed with the original node features; higher-dimensional cells



(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 parantheses, for each model, the maximum size k of rings lifted to 2-cells.

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

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 [39] and has been recently adopted as a reference benchmark to test the expressivity of GNNs [16]. 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 [9, 39], we use it to validate the expressive power of CWNs.

We follow the same evaluation setting as Dwivedi et al. [16]: 5-fold cross validation procedure and 20 different random weight initialisations. For our model, we set the maximum ring size $k = 8$. In Table 4 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. [16]. MP-GNNs, that is classic message passing GNNs (GAT [48], MoNet [36], GIN [53], etc.), and RingGNN [9] 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 [16, 35].

SR Similarly to Bodnar et al. [4] and [6], 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 [4, 6]. We reproduce the same experimental setting of Bodnar et al. [4]. In particular, we consider 9 distinct SR families[2] 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 6b. 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 as 'RingTransfer'. Graphs in this dataset are chordless cycles (rings) of size k. In each graph we mark two special nodes

[2]Data available at: `http://users.cecs.anu.edu.au/~bdm/data/graphs.html`.

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 [20]	79.2±2.1	55.9±0.3	59.6±0.1	>3 days	N/A	N/A	N/A	N/A
GK ($k = 3$) [45]	81.4±1.7	55.7±0.5	71.4±0.31	62.5±0.3	62.4±0.3	N/A	N/A	N/A
PK [40]	76.0±2.7	59.5±2.4	73.7±0.7	82.5±0.5	N/A	N/A	N/A	N/A
WL kernel [46]	90.4±5.7	59.9±4.3	75.0±3.1	**86.0**±1.8	N/A	73.8±3.9	50.9±3.8	81.0±3.1
DCNN [2]	N/A	N/A	61.3±1.6	56.6±1.0	N/A	49.1±1.4	33.5±1.4	N/A
DGCNN [55]	85.8±1.8	58.6±2.5	75.5±0.9	74.4±0.5	N/A	70.0±0.9	47.8±0.9	N/A
IGN [34]	83.9±13.0	58.5±6.9	**76.6**±5.5	74.3±2.7	**72.8**±1.5	72.0±5.5	48.7±3.4	N/A
GIN [53]	89.4±5.6	64.6±7.0	76.2±2.8	82.7±1.7	N/A	75.1±5.1	52.3±2.8	**92.4**±2.5
PPGNs [35]	**90.6**±8.7	66.2±6.6	**77.2**±4.7	83.2±1.1	**82.2**±1.4	73.0±5.8	50.5±3.6	N/A
Natural GN [14]	89.4±1.6	**66.8**±1.7	71.7±1.0	82.4±1.3	N/A	73.5±2.0	51.3±1.5	N/A
GSN [6]	**92.2** ± 7.5	**68.2** ± 7.2	**76.6** ± 5.0	**83.5** ± 2.0	N/A	**77.8** ± 3.3	**54.3** ± 3.3	N/A
SIN [4]	N/A	N/A	76.4 ± 3.3	82.7 ± 2.1	N/A	**75.6** ± 3.2	**52.4** ± 2.9	**92.2** ± 1.0
CIN (Ours)	**92.7** ± 6.1	**68.2** ± 5.6	**77.0** ± 4.3	**83.6** ± 1.4	**84.0** ± 1.6	**75.6** ± 3.7	**52.7** ± 3.1	**92.4** ± 2.1

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 6a we show the performance of a 3-layer CIN as a function of the ring size k, along with that of GIN [53] 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 [38] with small and medium sizes from biology (**PROTEINS** [5, 15]), chemistry (i.e. molecules – **MUTAG** [30, 43], **PTC**, **NCI1** and **NCI109** [49]) 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. [53], 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. [53]. We compare our CIN to the baseline methods: RWK [20], GK (with $k = 3$) [45], PK [40], WL kernel [46], DCNN [2], DGCNN [55], IGN [34], GIN [53], PPGNs [35], Natural GN [14], GSN [6], and SIN [4]. We report the results in Table 5. CIN compares more than 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.

ZINC We study the effectiveness of cellular message passing on larger scale molecular benchmarks from the ZINC database [47]. **ZINC** (12k graphs) and **ZINC-FULL** (250k graphs) [16, 23, 28, 54] 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 [16]. 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 [27] (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 [18] as reference and replicate the same hyperparameter setting in our model, including the use of only 2

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 [31]	0.469±0.002	N/A	N/A	76.06±0.97
GAT [48]	0.463±0.002	N/A	N/A	N/A
GatedGCN [7]	0.422±0.006	0.363±0.009	N/A	N/A
GIN [53]	0.408±0.008	0.252±0.014	0.088±0.002	77.07±1.49
PNA [12]	0.320±0.032	0.188±0.004	N/A	79.05±1.32
DGN [3]	0.219±0.010	0.168±0.003	N/A	79.70±0.97
HIMP [18]	N/A	0.151±0.006	0.036±0.002	78.80±0.82
GSN [6]	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**	**0.080±0.004**	**0.022±0.002**	**80.94±0.57**

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 [18], 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 [8, 17, 22]. All these convolutional methods are subsumed by the model in Bodnar et al. [4], which CWNs in turn subsume. To the best of our knowledge, Hajij et al. [24] 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 [28, 29], and the recent work of Fey et al. [18] 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. Also, the JT representation is limited to molecular modelling. More general substructures are accounted for in GSNs [6], 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 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.

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. Our code is supplied with the supplementary material.

References

[1] Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. *ICLR*, 2021.

[2] James Atwood and Don Towsley. Diffusion-convolutional neural networks. In *NIPS*, pages 1993–2001, 2016.

[3] Dominique Beaini, Saro Passaro, Vincent Létourneau, William L Hamilton, Gabriele Corso, and Pietro Liò. Directional graph networks. *ICML*, 2021.

[4] 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.

[5] 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.

[6] Giorgos Bouritsas, Fabrizio Frasca, Stefanos Zafeiriou, and Michael M Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *arXiv:2006.09252*, 2020.

[7] Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.

[8] 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.

[9] 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.

[10] Zhengdao Chen, Lei Chen, Soledad Villar, and Joan Bruna. Can graph neural networks count substructures? In *NeurIPS*, 2020.

[11] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). In *ICLR*, 2016.

[12] 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.

[13] Justin Michael Curry. *Sheaves, cosheaves and applications*. Thesis, University of Pennsylvania, 2014.

[14] Pim de Haan, Taco Cohen, and Max Welling. Natural graph networks. In *NeurIPS*, 2020.

[15] Paul D Dobson and Andrew J Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of Molecular Biology*, 330(4):771–783, 2003.

[16] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*, 2020.

[17] Stefania Ebli, Michaël Defferrard, and Gard Spreemann. Simplicial neural networks. In *NeurIPS Workshop on Topological Data Analysis and Beyond*, 2020.

[18] 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+) Workhop*, 2020.

[19] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[20] 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.

[21] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, 2017.

[22] Nicholas Glaze, T. Mitchell Roddenberry, and Santiago Segarra. Principled simplicial neural networks for trajectory prediction. In *ICML*, 2021.

[23] 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.

[24] Mustafa Hajij, Kyle Istvan, and Ghada Zamzmi. Cell complex neural networks. In *NeurIPS Workshop on Topological Data Analysis and Beyond*, 2020.

[25] J. Hansen and R. Ghrist. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology*, 3:315–358, 2019.

[26] Allen Hatcher. *Algebraic topology*. Cambridge Univ. Press, Cambridge, 2000.

[27] 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.

[28] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *ICML*, pages 2323–2332, 2018.

[29] Wengong Jin, Kevin Yang, Regina Barzilay, and Tommi Jaakkola. Learning multimodal graph-to-graph translation for molecular optimization. In *ICLR*, 2019.

[30] Jeroen Kazius, Ross McGuire, and Roberta Bursi. Derivation and validation of toxicophores for mutagenicity prediction. *Journal of Medicinal Chemistry*, 48(1):312–320, 2005.

[31] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[32] Greg Landrum. RDKit: Open-source cheminformatics software. 2016.

[33] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, volume 32, 2018.

[34] Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. In *ICLR*, 2018.

[35] Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. In *NeurIPS*, 2019.

[36] 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.

[37] 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.

[38] 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+) Workhop*, 2020.

[39] Ryan Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Relational pooling for graph representations. In *ICML*, volume 97, pages 4663–4673, 2019.

[40] Marion Neumann, Roman Garnett, Christian Bauckhage, and Kristian Kersting. Propagation kernels: efficient graph kernels from propagated information. *Machine Learning*, 102(2): 209–245, 2016.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. Efficient graphlet kernels for large graph comparison. In *AISTAT*, pages 488–495, 2009.

[46] Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *JMLR*, 12(Sep):2539–2561, 2011.

[47] 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`.

[48] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[49] 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.

[50] 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.

[51] J. H. C. Whitehead. Combinatorial homotopy. I. *Bulletin of the American Mathematical Society*, 55(3.P1):213 – 245, 1949. doi: bams/1183513543. URL `https://doi.org/`.

[52] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. In *ICML*, 2018.

[53] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

[54] Jiaxuan You, Bowen Liu, Rex Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. *NeurIPS*, 2018.

[55] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *AAAI*, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Provided in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Details are provided in Section 5 and Appendix D.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 5 and Appendix D.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We used multiple existing datasets described and referenced in Section 5.
   (b) Did you mention the license of the assets? [Yes] See Appendix D.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We have added our code to the supplementary material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We are only using publicly available datasets.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] None of the datasets we are using contain personally identifiable information or offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]