Weisfeiler and Lehman Go Paths: Learning Topological Features via Path Complexes

Quang Truong*, **Peter Chin**

Thayer School of Engineering, Dartmouth College
cong.minh.quang.truong.th@dartmouth.edu, peter.chin@dartmouth.edu

Abstract

Graph Neural Networks (GNNs), despite achieving remarkable performance across different tasks, are theoretically bounded by the 1-Weisfeiler-Lehman test, resulting in limitations in terms of graph expressivity. Even though prior works on topological higher-order GNNs overcome that boundary, these models often depend on assumptions about substructures of graphs. Specifically, topological GNNs leverage the prevalence of cliques, cycles, and rings to enhance the message-passing procedure. Our study presents a novel perspective by focusing on simple paths within graphs during the topological message-passing process, thus liberating the model from restrictive inductive biases. We prove that by lifting graphs to path complexes, our model can generalize the existing works on topology while inheriting several theoretical results on simplicial complexes and regular cell complexes. Without making prior assumptions about graph sub-structures, our method outperforms earlier works in other topological domains and achieves state-of-the-art results on various benchmarks.

1 Introduction

Graph-based learning presents an intricate problem due to the inherent ambiguity in defining geometric properties such as canonical vertex ordering (Bouritsas et al. 2023). Exploiting these properties is however challenging, given the pervasive utility of graphs across an array of domains. Some early prominent efforts on graph neural networks (GNNs) shed light upon this field (Bruna et al. 2014; Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2017), which effectively paved the path for the recent advancements (Xu et al. 2018, 2019; Veličković et al. 2018). However, the graph expressivity of GNNs is upper-bounded by the 1-WL test (Weisfeiler and Lehman 1968), which is proven by (Xu et al. 2019; Morris et al. 2019). This limitation has led to interests in higher-order GNNs, a sub-family of GNNs that extend beyond simple pairwise vertex interactions to encompass broader relationships (Huang and Yang 2021; Feng et al. 2018; Yadati et al. 2019; Bodnar et al. 2021b,a; Ebli, Defferrard, and Spreemann 2020; Roddenberry, Glaze, and Segarra 2021; Giusti et al. 2022, 2023; Hajij et al. 2023; Papillon et al. 2023). For example, SIN(Bodnar et al. 2021b) and CIN (Bodnar et al. 2021a) are proven to be not less powerful than the 3-WL test, k-GNNs (Morris et al. 2019) and PPGNs (Maron et al. 2019a) are proven to be as powerful as the k-WL test (Grohe and Otto 2015; Grohe 2017), and δ-k-LGNN (Morris, Rattan, and Mutzel 2020) is strictly more powerful than the k-WL test. Topological GNNs, which extend graphs to topological domains such as simplicial complexes (Bodnar et al. 2021b; Ebli, Defferrard, and Spreemann 2020; Roddenberry, Glaze, and Segarra 2021; Schaub et al. 2020) and regular cell complexes (Bodnar et al. 2021a; Giusti et al. 2022, 2023) to learn higher-order features, demonstrate outstanding performance on graph classification tasks. Yet, these models are predicated on the premise that graphs should contain cliques, cycles, or induced cycles (rings).

Drawing inspiration from path complexes (Grigor'yan et al. 2020; Grigor'yan et al. 2013), we relax the above assumption by focusing only on simple paths, foundational yet universal elements in graphs, during message propagation. Specifically, we lift our graphs to a topological domain referred to as **path complexes**, with **elementary paths** serving as the basis elements. Under certain conditions, path complexes generalize simplicial complexes (Grigor'yan et al. 2013; Grigor'yan et al. 2020), thus offering a more flexible structure to work with. Even though path complexes cannot generalize regular cell complexes, we prove that our proposed Path Weisfeiler-Lehman (PWL) test is at least as powerful as CWL(k-IC) (Bodnar et al. 2021a), in which k-IC is a ring-based lifting map attaching 2-cells to rings of a maximum size k. The realization of the PWL test via neural message-passing procedure (Gilmer et al. 2017) is called Path Complex Networks (PCN), which surpasses its counterparts MPSN (Bodnar et al. 2021b) and CWN (Bodnar et al. 2021a) in performance across various benchmarks.

Main Contributions Our work introduces a novel graph isomorphism test PWL and topological message-passing scheme PCN operating on path complexes, which encapsulate several theoretical properties of SWL (Bodnar et al. 2021b) and CWL (Bodnar et al. 2021a). We provide theoretical connections between PWL and the latter higher-order WL tests, and we prove that PWL can indeed generalize SWL (Bodnar et al. 2021b) and CWL(k-IC) (Bodnar et al. 2021a). Empirical validation of our assertions is offered

*Corresponding author

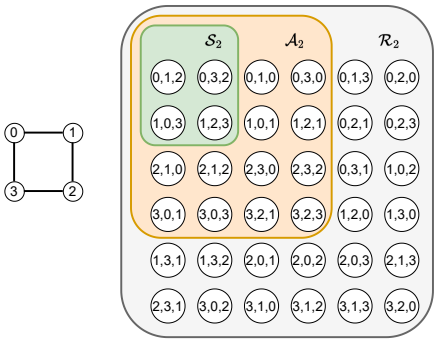

Figure 1: An illustration of different 2-path spaces of a path complex arising from the graph on the left. Each element in each space is an elementary 2-path that spans the corresponding space.

through evaluations of PCN on various real-world benchmarks and a collection of 9 strongly regular graph (SRGs) families. Notably, our proposed approach achieves superior performance without the need for assumptions about graph sub-structures other than paths, which are inherent in every connected graph.

2 Preliminaries

2.1 Path Complex

Definition 1. *(Grigor'yan et al. 2013; Grigor'yan et al. 2020) Given a finite non-empty set V whose element is called vertex, an **elementary p-path** on set V is any sequence of vertices with length $p+1$. Elementary p-path is denoted by $e_{i_0 \dots i_p}$.*

An elementary path can be understood as the most fundamental element of a path complex. We construct a linear space defined over a field $\mathbb{K}$ called Λ_p that contains all possible linear combinations of elementary p-paths. Each member of Λ_p is identified as a **p-path**.

Grigor'yan et al. also defines boundary operator $\partial : \Lambda_p \to \Lambda_{p-1}$ for elementary p-paths, an operator analogous to the corresponding operator for simplicial complexes.

Definition 2. *(Grigor'yan et al. 2013; Grigor'yan et al. 2020) Boundary operator on elementary $p-$paths is defined as:*

$$\partial\, e_{i_0 \dots i_p} = \sum_{q=0}^{p} (-1)^q e_{i_0 \dots \hat{i}_q \dots i_p},$$

where $\hat{i}_q$ indicates the removal of the index i_q from the sequence $i_0 \dots i_p$.

Elementary paths may contain vertices consecutively repeated in a sequence, and these are termed as **non-regular** elementary paths. If this is not the case, the elementary paths are classified as **regular**. If we define $\mathcal{R}_p \subset \Lambda_p$ the space encompassing all possible linear combinations of regular elementary p-paths and confine the boundary operator to $\mathcal{R}_p$ only, the **regular boundary operator** $\partial : \mathcal{R}_p \to \mathcal{R}_{p-1}$ will exclude any non-regular elementary $(p-1)$-paths from the boundary set of an elementary p-path.

Definition 3. *(Grigor'yan et al. 2013; Grigor'yan et al. 2020) Given a finite non-empty set V, a **path complex** P is a non-empty collection of elementary paths such that for any sequence of vertices that belong to P, the truncated sequences, in which either the first vertex or the last vertex is removed, are also included in P.*

We denote $P_p \subset P$ where P_p contains all elementary paths with length p. Elements of P_p are called **allowed elementary p-paths**, while any sequences that do not exist in P_p are called **non-allowed elementary p-paths**. Similarly, we can construct $\mathcal{A}_p \subset \Lambda_p$ such that $\mathcal{A}_p$ contains all possible linear combinations of allowed elementary p-paths. However, it may happen that $\partial\, \mathcal{A}_p \not\subset \mathcal{A}_{p-1}$, as an index-omitted sequence may be non-allowed. Therefore, Grigor'yan et al. constructs another subspace $\Omega_p \subseteq \mathcal{A}_p$, which is termed as **regular space of boundary-invariant p-paths**:

$$\Omega_p = \{v \in \mathcal{A}_p : \partial\, v \in \mathcal{A}_{p-1}\}$$

in which boundary operation is well-defined.

For instance, consider Figure 1 which visualizes 2-spaces associated with a path complex extended from the graph shown on the left. In this example, $\mathcal{A}_2$ is a subset of $\mathcal{R}_2$ due to the absence of self-loops in the graph. As illustrated in Figure 1, none of the simple paths (no repeating vertices in a sequence) can constitute a basis for Ω_2 as there always exists a non-allowed 1-path (a diagonal edge of the square) after applying boundary operation on that simple path. However, consider 2-path $v = e_{012} - e_{032}$. Applying boundary operation on v:

$$\begin{aligned} \partial\, v &= e_{12} - e_{02} + e_{01} - e_{32} + e_{02} - e_{03} \\ &= e_{12} + e_{01} - e_{32} - e_{03} \end{aligned}$$

The outcome is a linear combination of allowed elementary 1-paths, hence $\partial\, v \in \mathcal{A}_1$. As such, v can serve as a base for Ω_2.

2.2 Complex Lifting Transformations

Given a simple graph $G = (\mathcal{V}, \mathcal{E})$ with a finite vertex set $\mathcal{V}$ and edge set $\mathcal{E}$, we can apply lifting transformation such as clique complex lifting (Bodnar et al. 2021b) or cell complex lifting (Bodnar et al. 2021a). Complex lifting transformations are graph pre-processing techniques such that we extend G to a complex K, wherein members of K are related by part-whole relations (Papillon et al. 2023) (members under a topological hierarchy) or adjacent relations (members with shared boundaries or co-boundaries). Members can be either k-simplices, k-cells, or elementary k-paths.

Definition 4 (Boundary incidence relation)**.** *(Bodnar et al. 2021b,a) For any σ and τ members of K, σ is considered a boundary of τ (denoted as $\sigma \prec \tau$) if and only if $\sigma \subset \tau$ and there does not exist $\delta \in K$ such that $\sigma \prec \delta \prec \tau$.*

Definition 5 (Relations between members)**.** *(Bodnar et al. 2021b,a) For any member σ of K, there are four types of relations:*

- *Boundary* $\mathcal{B}(\sigma) = \{\tau \mid \tau \prec \sigma\}$
- *Co-boundary* $\mathcal{C}(\sigma) = \{\tau \mid \sigma \prec \tau\}$

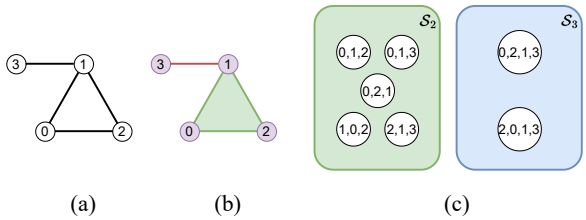

Figure 2: (a) Original graph; (b) Simplicial complex, which contains a 2-simplex, 4 1-simplices, and 4 0-simplices, arising from the original graph. Regular cell complex coincides with the simplicial complex in this case; (c) Simple path spaces $\mathcal{S}_2$ and $\mathcal{S}_3$ corresponding to the path complex arising from the original graph. Elementary paths of $\mathcal{S}_0$ and $\mathcal{S}_1$ are indeed 0-simplices (0-cells) and 1-simplices (1-cells) of the simplicial complex (regular cell complex).

- *Upper-adjacent neighborhood* $\mathcal{N}_{\uparrow}(\sigma) = \{\tau \mid \sigma \prec \delta \ \wedge \ \tau \prec \delta\}$
- *Lower-adjacent neighborhood* $\mathcal{N}_{\downarrow}(\sigma) = \{\tau \mid \delta \prec \sigma \ \wedge \ \delta \prec \tau\}$

2.3 Weisfeiler-Lehman Tests

Weisfeiler-Lehman test (1-WL test) is a basic algorithm to investigate the isomorphism of two graphs (Weisfeiler and Lehman 1968). The algorithm serves as a standard method for GNNs' expressivity evaluation (Maron et al. 2019a; Xu et al. 2019; Morris et al. 2019; Chen et al. 2020; Bodnar et al. 2021b,a). 1-WL test initiates every node with the same color from a color palette, and then iteratively updates the color of each node based on its current color and the colors of its neighboring nodes. Specifically, the new color $c_v^{(t+1)}$ of node v at time step $(t+1)$ is updated by an injective HASH function that maps the current color $c_v^{(t)}$ and a collection (multi-set) of neighbors' color:

$$c_v^{(t+1)} = \text{HASH}\left(c_v^{(t)}, \left\{\!\!\left\{ c_w^{(t)} \mid w \in \mathcal{N}(v) \right\}\!\!\right\}\right)$$

The algorithm concludes when colors reach a stable state. The final graph representation is the histogram of the stable colors. If two graphs do not have the same histogram, they are not isomorphic. However, the converse does not necessarily hold true.

The algorithm proposed in (Grohe and Otto 2015; Grohe 2017), which is called k-dimension Weisfeiler-Lehman test (k-WL test), is an extension of the 1-WL test that assigns colors to k-tuples of nodes (Morris et al. 2019; Huang and Villar 2021). Another variant of WL tests is k-folklore-WL test (Cai, Furer, and Immerman 1989), which is proven to be as powerful as $(k+1)$-WL test (Grohe and Otto 2015; Grohe 2017) for $k \geq 2$.

3 Path Weisfeiler-Lehman Test

3.1 Procedure

Inspired by (Bodnar et al. 2021b,a), we would like to introduce a new Weisfeiler-Lehman test based on path complex lifting transformations. However, unlike SWL(Bodnar et al. 2021b) and CWL (Bodnar et al. 2021a) which depend on clique or ring substructures, our method eases that prior assumption by performing color refinement solely on elementary paths. We distinguish our work with the k-set WL test proposed by (Morris et al. 2019), since k-sets are non-local and may not preserve the graph's topology as elementary paths do.

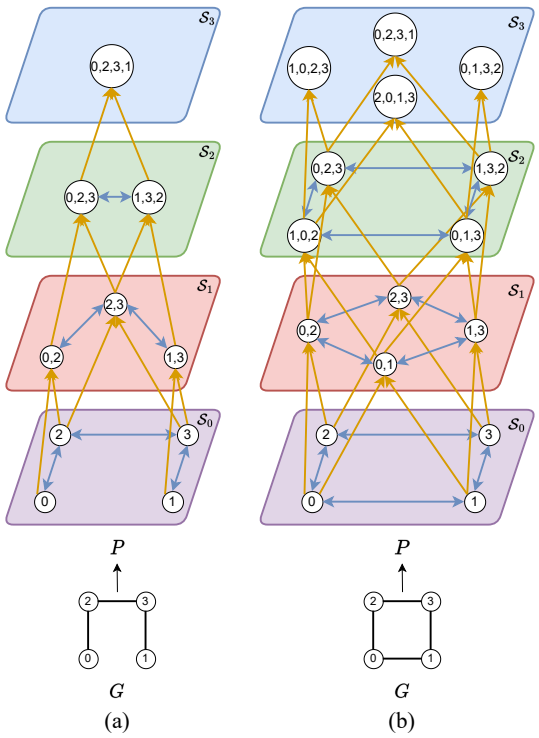

Figure 3: Examples of path complexes arising from (a) a simple path with length of 3 and (b) a ring with size of 4. Blue arrows demonstrate upper-adjacent relations, while orange arrows demonstrate boundary relations.

Directly working with the regular space of boundary-invariant paths Ω_p is non-trivial, because, as outlined in Section 2.1, some bases that span Ω_p are indeed formal linear combinations of elementary p-paths. Thus, we construct a new space $\mathcal{S}_p$ where $\mathcal{S}_p$ is spanned by all simple paths with length p, which are allowed elementary p-paths of a path complex K extended by graph G. Clearly, $\mathcal{S}_p \subset \mathcal{A}_p \cap \mathcal{R}_p$. As we only work with simple graphs, we regard any sequence and its reverse as identical. Assume there exists an injective function f that assigns each vertex in a vertex set V of a graph G a real value. Thus, we restrict S_p to only sequences where the first vertex is smaller than the last vertex. Figure 2 illustrates the main difference between our lifting transformation and the ones proposed by (Bodnar et al. 2021b,a). We can observe the advantage of path complex lifting transformation, as we can theoretically lift to a higher dimension than that of simplicial complex and regular cell complex. Define P a path complex with the highest dimension p such that for any dimension $k \leq p$, P_k contains all elementary k-paths that span $\mathcal{S}_k$, and boundary set of any

elementary k-paths is restricted to elementary $(k-1)$-paths in $\mathcal{S}_{k-1}$. Figure 3 illustrates our idea of extending graphs to path complexes.

Drawing motivation from SWL test (Bodnar et al. 2021b) and CWL test (Bodnar et al. 2021a), we define the following multi-set of colors that represents the part-whole and adjacent relations of elementary paths as defined in Definition 5.

Definition 6 (Color representations of relations)**.** *Given an elementary path σ with the associate color c^t_σ at time-step t, we have the following multi-set of colors:*

- $c^t_{\mathcal{B}}(\sigma) = \{\!\{c^t_\tau \,|\, \tau \in \mathcal{B}(\sigma)\}\!\}$
- $c^t_{\mathcal{C}}(\sigma) = \{\!\{c^t_\tau \,|\, \tau \in \mathcal{C}(\sigma)\}\!\}$
- $c^t_{\uparrow}(\sigma) = \{\!\{(c^t_\tau, c^t_\delta) \,|\, \tau \in \mathcal{N}_{\uparrow}(\sigma) \,\wedge\, \delta \in \mathcal{C}(\sigma,\tau)\}\!\}$
- $c^t_{\downarrow}(\sigma) = \{\!\{(c^t_\tau, c^t_\delta) \,|\, \tau \in \mathcal{N}_{\downarrow}(\sigma) \,\wedge\, \delta \in \mathcal{B}(\sigma,\tau)\}\!\}$

where $\mathcal{C}(\sigma,\tau) = \mathcal{C}(\sigma) \cap \mathcal{C}(\tau)$ and $\mathcal{B}(\sigma,\tau) = \mathcal{B}(\sigma) \cap \mathcal{B}(\tau)$.

Since two elementary paths may share more than one boundary or co-boundary, we adopt the coloring scheme proposed by CWL (Bodnar et al. 2021a) as opposed to the one in SWL (Bodnar et al. 2021b). Path Weisfeiler-Lehman Test (PWL) performs color refinements on elementary paths that belong to P by following the below procedure:

1. Initialize all elementary paths in path complex P with the same color.
2. For every elementary path σ, injectively map the current color of σ and the associated multi-sets of colors to a new color $c^{t+1}_\sigma = \textsc{Hash}\left(c^t_\sigma, c^t_{\mathcal{B}}(\sigma), c^t_{\mathcal{C}}(\sigma), c^t_{\downarrow}(\sigma), c^t_{\uparrow}(\sigma)\right)$.
3. Represent colorings of P by a collection of all elementary paths. Repeat Step 2 until the colorings of P are stable. Two path complexes are non-isomorphic if they have different stable color histograms of elementary paths.

3.2 Theoretical Results

We aim to establish theoretical connections between PWL and SWL, as well as CWL. Similar to SWL and CWL, after performing **path coloring**, we compare two path complexes via **c-similarity** as proposed by Bodnar et al..

Definition 7. *A **path coloring** is a mapping c that maps a path complex P and one of its elementary paths σ to a color in a fixed color palette. The mapped color is denoted by c^P_σ.*

Definition 8. *Given a path coloring c, two path complexes X and Y are said to be **c-similar**, denoted by $c^X = c^Y$, if the number of elementary paths with a certain color in X is equal to the number of elementary paths with that color in Y. Otherwise, $c^X \neq c^Y$.*

Our paper focuses on path coloring c satisfying the condition that if X and Y are isomorphic, they are c-similar. Similar to cells in regular cell complexes, members with the same dimension in our re-defined path complexes may have different numbers of lower-dimensional elementary paths on their boundaries. Thus, we can translate the below existing theoretical results on regular cell complexes proven in (Bodnar et al. 2021a) to path complexes. We advise the audience to read (Bodnar et al. 2021b,a) for detailed proofs of the following results.

Definition 9 (Color refinement on elementary paths)**.** *A path coloring c refines a path coloring d, denoted by $c \sqsubseteq d$, if for all elementary paths σ and τ in path complexes X and Y respectively, $c^X_\sigma = c^Y_\tau$ implies $d^X_\sigma = d^Y_\tau$.*

Corollary 10. *Consider two path colorings c and d on path complexes X and Y respectively such that $c \sqsubseteq d$. If $d^X \neq d^Y$, then $c^X \neq c^Y$.*

Theorem 11. *PWL with $\textsc{Hash}\left(c^t_\sigma, c^t_{\mathcal{B}}(\sigma), c^t_{\uparrow}(\sigma)\right)$ is as powerful as PWL with generalized update rule $\textsc{Hash}\left(c^t_\sigma, c^t_{\mathcal{B}}(\sigma), c^t_{\mathcal{C}}(\sigma), c^t_{\uparrow}(\sigma), c^t_{\downarrow}(\sigma)\right)$.*

Unless stated otherwise, we use PWL with $\textsc{Hash}\left(c^t_\sigma, c^t_{\mathcal{B}}(\sigma), c^t_{\uparrow}(\sigma)\right)$ for the rest of the paper.

According to (Grigor'yan et al. 2013; Grigor'yan et al. 2020), any simplicial complex can be extended naturally to a path complex if they meet two criteria: 1) The path complex should be perfect, which means any subsequence of any path must also be present in the path complex and 2) There must exist an injective real-valued vertex labeling function such that every vertex along any path must be in a monotonically increasing order. We have the following theoretical result.

Theorem 12. *PWL is at least as powerful as SWL at distinguishing non-isomorphic graphs.*

Similar to cells, elementary paths can have different boundary sizes. Thus, we can also obtain the following result for PWL, which is similar to that for CWL (Bodnar et al. 2021a).

Proposition 13. *For any two path complexes X and Y, if $\sigma \in X$ and $\tau \in Y$ have different boundary sizes $|\mathcal{B}(\sigma)| \neq |\mathcal{B}(\tau)|$, their colorings are different $c^{X,t}_\sigma \neq c^{Y,t}_\tau$ for $t > 0$.*

While an elementary paths may coincide with simplex, a single elementary path could not encode a k-cell in a regular cell complex where $k \geq 2$. For example, as shown in Figure 3, the elementary 3-path e_{0231} shows up in both path complexes, but for (a), it does not coincide with 2-cell if we extend the graph to a regular cell complex. However, we can prove that PWL is at least as powerful as CWL(k-IC) with the maximum dimension of 2 when distinguishing non-isomorphic simple graphs via comparing color representations of **cyclic-shifting families**.

Definition 14 (Cyclic-shifting Family)**.** *Suppose σ is an elementary n-path of a path complex extended from a graph G. If σ is also a 2-cell of a regular cell complex extended from the same graph in which $|\mathcal{B}(\sigma)| = n+1$, for every $p \leq n$, there is a set of elementary p-paths which contains all subsequences with length p of cyclic-shifted variants of the sequence of σ. We denote the set of such sequences $\mathcal{F}_p(\sigma)$, which is called the cyclic-shifting p-family of σ.*

For example, consider Figure 3, suppose we have a 2-cell $\sigma = e_{1023}$. The cyclic-shifting families of σ are:

$$\mathcal{F}_3(\sigma) = \{e_{1023}, e_{0231}, e_{0132}, e_{2013}\}$$
$$\mathcal{F}_2(\sigma) = \{e_{013}, e_{023}, e_{102}, e_{132}\}$$
$$\mathcal{F}_1(\sigma) = \{e_{01}, e_{02}, e_{13}, e_{23}\}$$
$$\mathcal{F}_0(\sigma) = \{e_0, e_1, e_2, e_3\}$$

Even though e_{0132} and e_{2013} are not cyclic-shifted from e_{1023} and e_{0231}, we include them instead of e_{2310} and e_{3102} in $\mathcal{F}_3(\sigma)$ because we regard sequences and their reversed sequences identical as proposed in Section 3.1. Similarly, we obtain $\mathcal{F}_2(\sigma), \mathcal{F}_1(\sigma)$ and $\mathcal{F}_0(\sigma)$ as shown above. Even though we do not explicitly take cyclic-shifting families into account when performing PWL, we can prove that cyclic-shifting families play an equivalent role to 2-cells in CWL(k-IC) via the color representations of cyclic-shifting families.

Definition 15 (Color Representation of Cyclic-shifting Family)**.** *For any path complex X, given an elementary p-path $\sigma \in X$ and σ also coincides with 2-cells with the boundary size of $p+1$, the color representation of $\mathcal{F}_p(\sigma)$ at time-step t is denoted as a multi-set:*

$$C_p^{X,t}(\sigma) = \left\{\!\!\left\{ c_{\delta_\sigma}^{X,t} \mid \delta_\sigma \in \mathcal{F}_p(\sigma) \right\}\!\!\right\}$$

Then, we have the following proposition.

Proposition 16. *For any two path complexes X and Y, if two cyclic-shifting p-families have similar color representation $C_p^{X,t+p}(\sigma) = C_p^{Y,t+p}(\tau)$ in which $\sigma \in X$ and $\tau \in Y$, cyclic-shifting k-families also have similar color representation $C_k^{X,t+k}(\sigma) = C_k^{Y,t+k}(\tau)$ where $k \leq p$.*

Proposition 16 implies that similar higher-order cyclic-shifting families can induce similar lower-order cyclic-shifting families. This is particularly relevant for 1-families, which include 1-paths and 1-cells. We have the following lemmas.

Lemma 17. *For any pair of path complexes X and Y arising from two graphs G_1 and G_2, if two elementary 0-paths (1-paths whose co-boundaries are not belong to any cyclic-shifting 2-families) $\sigma \in X$ and $\tau \in Y$ have similar colorings $c_\sigma^{X,t} = c_\tau^{Y,t}$ performed by PWL at time-step t, we have two corresponding 0-cells (1-cells not adjacent to any 2-cells) $\sigma \in A$ and $\tau \in B$ with similar color representations $d_\sigma^{A,t} = d_\tau^{B,t}$ performed by CWL(k-IC) at time-step t, where A and B are regular cell complexes arising from G_1 and G_2.*

Lemma 18. *For any two path complexes X and Y arising from two graphs G_1 and G_2, if cyclic-shifting $(n-1)$-families of $\sigma \in X$ and $\tau \in Y$ have similar color representations $C_{n-1}^{X,t+n-3}(\sigma) = C_{n-1}^{Y,t+n-3}(\tau)$ performed by PWL at time-step $t+n-3$, we have two corresponding 2-cells $\alpha \in A$ and $\beta \in B$ with similar colorings $d_\alpha^{A,t} = d_\beta^{B,t}$ performed by CWL(k-IC) at time-step t, in which A and B are regular cell complexes arising from G_1 and G_2 and n is the boundary size of the 2-cells.*

Lemma 19. *For any two path complexes X and Y arising from two graphs G_1 and G_2, if two elementary 1-paths $\alpha \in X$ and $\beta \in Y$ have the same coloring $c_\alpha^{X,t} = c_\beta^{Y,t}$ and their cyclic-shifting $(n-1)$-families have similar color representations $C_{n-1}^{X,t+n-3}(\sigma) = C_{n-1}^{Y,t+n-3}(\tau)$ performed by PWL at time-step $t+n-3$, we have two corresponding 1-cells $\alpha \in A$ and $\beta \in B$ with the same coloring $d_\alpha^{A,t} = d_\beta^{B,t}$ performed by CWL(k-IC) at time-step t, in which A and B are regular cell complexes arising from G_1 and G_2 and n is the boundary size of the 2-cells that are adjacent to α and β.*

Given the above lemmas, as all cells with different orders of 2-dimensional regular cell complexes can be induced from path complexes and their corresponding cyclic-shifting families, we come up with the following theorem and corollaries.

Theorem 20. *PWL is at least as powerful as CWL(k-IC) with the maximum dimension of 2 at distinguishing non-isomorphic graphs.*

Corollary 21. *PWL is strictly more powerful than WL at distinguishing non-isomorphic graphs.*

Corollary 22. *PWL is not less powerful than 3-WL at distinguishing non-isomorphic graphs.*

4 Realization of PWL Test via Message-passing Neural Networks

Similar to (Bodnar et al. 2021b,a), a natural way to realize PWL is through Message Passing Neural Networks (MPNNs) (Gilmer et al. 2017), in which MPNNs pass messages along elementary paths based on part-whole and adjacent relations as defined in Section 2.2. As demonstrated by Theorem 9 proven by Bodnar et al., we can exclude the message from co-boundaries and lower-adjacent neighbors. Thus, we update feature of an elementary path by the following:

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

$$m_{\uparrow}^{(t+1)}(\sigma) = \text{AGG}_{\tau \in \mathcal{N}_\uparrow(\sigma), \delta \in \mathcal{C}(\sigma,\tau)} \left(\text{M}_{\uparrow}(h_\sigma^{(t)}, h_\tau^{(t)}, h_\delta^{(t)}) \right)$$

$$h_\sigma^{(t+1)} = \text{UP} \left(h_\sigma^{(t)}, m_{\mathcal{B}}^{(t)}(\sigma), m_{\uparrow}^{(t+1)}(\sigma) \right)$$

in which $\text{M}_{\mathcal{B}}, \text{M}_\uparrow$ and UP can be modeled by learnable functions. PCN inherit all properties of MPSN (Bodnar et al. 2021b) and CWN (Bodnar et al. 2021a), given that they all operate under the same mechanism. Thus, we naturally have the following results, whose proofs are identical to those in (Bodnar et al. 2021a).

Theorem 23. *PCNs are at most powerful as PWL. PCNs can be as powerful as PWL if PCNs are equipped with a sufficient number of layers and injective aggregators.*

Theorem 24. *PCN layers are elementary path permutation equivariant.*

For the rest of the paper, we leverage the following model called Path Isomorphism Network (PIN). The architecture of the model is identical to SIN (Bodnar et al. 2021b) and CIN (Bodnar et al. 2021a), albeit with slight alterations. The main distinguishing factor lies in the graph lifting transformation. This family of models is commonly regarded as a higher-order version of GIN (Xu et al. 2019) because of the similar update formula. The update formulae are detailed in the Appendix.

Dataset	PROTEINS	NCI1	NCI109	IMDB-B
RWK	59.6 ± 0.1	> 3 days	N/A	N/A
GK (k=3)	71.4 ± 0.3	62.5 ± 0.3	62.4 ± 0.3	N/A
PK	73.7 ± 0.7	82.5 ± 0.5	N/A	N/A
WL Kernel	75.0 ± 3.1	86.0 ± 1.8 ♦	N/A	73.8 ± 3.9
DCNN	61.3 ± 1.6	56.6 ± 1.0	N/A	49.1 ± 1.4
DGCNN	75.5 ± 0.9	74.4 ± 0.5	N/A	70.0 ± 0.9
IGN	76.6 ± 5.5	74.3 ± 2.7	72.8 ± 1.5	72.0 ± 5.5
GIN	76.2 ± 2.8	82.7 ± 1.7	N/A	75.1 ± 5.1
PPGNs	77.2 ± 4.7	83.2 ± 1.1	82.2 ± 1.4	73.0 ± 5.8
Natural GN	71.7 ± 1.0	82.4 ± 1.3	N/A	73.5 ± 2.0
GSN	76.6 ± 5.0	83.5 ± 2.0	N/A	77.8 ± 3.3 ♦
pathGCN	80.4 ± 4.2 ▲	83.3 ± 1.3	N/A	N/A
PathNN	75.2 ± 3.9	82.3 ± 1.9	N/A	72.6 ± 3.3
SIN †	76.4 ± 3.3	82.7 ± 2.1	N/A	75.6 ± 3.2 •
CIN †	77.0 ± 4.3	83.6 ± 1.4	84.0 ± 1.6 •	75.6 ± 3.7
CAN	78.2 ± 2.0	84.5 ± 1.6	83.6 ± 1.2	N/A
CIN++	80.5 ± 3.9 ♦	85.3 ± 1.2 ▲	84.5 ± 2.4 ♦	N/A
PIN (Ours)	78.8 ± 4.4 •	85.1 ± 1.5 •	84.0 ± 1.5 ▲	76.6 ± 2.9 ▲

Table 1: TUDataset Benchmarks. The first part consists of graph kernel methods, the second part consists of GNNs, and the third part consists of higher-order GNNs. The top-3 methods in each benchmark are denoted by ♦ (1st place), ▲ (2nd place), and • (3rd place). Baselines are denoted by †.

5 Experiments

We evaluate PIN on several real-world datasets across different domains such as molecular graphs or social graphs. We also provide an in-depth empirical study, which supports our claim that PWL is at least as powerful as CWL(k-IC), on SRGs. Detailed hyperparameter settings along with relevant ablation studies are documented in the Appendix.

5.1 TUDataset Benchmarks

TUDataset benchmarks, encompassing a broad spectrum of graph datasets from biology, chemistry, and social networks, are proposed in (Morris et al. 2020). In this paper, we evaluate our model on 4 different benchmarks on classification tasks: PROTEINS, NCI1, NCI109, and IMDB-B. However, an evaluation of our model on large graphs such as those in RDT-B is unattainable due to the prohibitive time complexity. The evaluation methodology adheres to the procedure outlined by (Xu et al. 2019). Specifically, we report the highest mean test accuracy across a 10-fold cross-validation as indicated in (Xu et al. 2019).

As illustrated in Table 1, an updated version of the tables presented in (Bodnar et al. 2021a; Giusti et al. 2023), PIN exhibits superior accuracy in comparison to SIN (Bodnar et al. 2021b) and CIN (Bodnar et al. 2021a), even on molecular graph datasets, where rings hold considerable significance.

For the TUDataset Benchmarks, we compare performance of three types of methods: graph kernels (RWK (Gärtner, Flach, and Wrobel 2003), GK (Shervashidze et al. 2009), PK (Neumann et al. 2016), WL Kernel (Shervashidze et al. 2011)), GNNs (DCNN (Atwood and Towsley 2016), DGCNN (Zhang et al. 2018), IGN (Maron et al. 2019b), GIN (Xu et al. 2019), PPGNs (Maron et al. 2019a), Natural GN (de Haan, Cohen, and Welling 2020), GSN (Bouritsas et al. 2023), pathGCN (Eliasof, Haber, and Treister 2022), PathNN (Michel et al. 2023)) and higher-order GNNs (SIN (Bodnar et al. 2021b), CIN (Bodnar et al. 2021a), CAN (Giusti et al. 2022), CIN++ (Giusti et al. 2023)).

5.2 ZINC

We also evaluate our model on the ZINC dataset, a common graph benchmark for regression tasks (Sterling and Irwin 2015; Dwivedi et al. 2023). As shown in Table 2, which is an updated table of (Bouritsas et al. 2023) and (Giusti et al. 2023), our model's performance is lower than that of the baseline CIN (Bodnar et al. 2021a) on the test set of ZINC. Our conjecture regarding this performance gap is primarily attributed to the low dimensionality of the path complex, which cannot fully represent large ring structures, and the shallowness of our architectures, which cannot propagate messages efficiently from elementary 0-paths to the highest-dimensional elementary paths. However, our model outperforms the remaining state-of-the-art methods by large margins.

For the ZINC dataset, we compare performance of GCN (Kipf and Welling 2017), GAT (Veličković et al. 2018), GatedGCN (Bresson and Laurent 2018), GIN (Xu et al. 2019), PNA (Corso et al. 2020), DGN (Beaini et al. 2020), HIMP (Fey, Yuen, and Weichert 2020), GSN (Bouritsas et al. 2023), PathNN (Michel et al. 2023), CIN (Bodnar et al. 2021a), and CIN++ (Giusti et al. 2023).

5.3 OGBG-MOLHIV

We also evaluate our model on the OGBG-MOLHIV dataset (Hu et al. 2020), which contains about 41k graphs for the graph binary classification task. Hu et al. applies a scaffold splitting procedure (Wu et al. 2018) for the OGBG-MOLHIV dataset; thus we report the performance of our model on both the validation and test sets in Table 2. Analogous to the results observed from the ZINC dataset, our model performs less effectively than the baselines. For OGBG-MOLHIV, we compare our model to the same methods as in ZINC.

5.4 Strongly Regular Graphs

The original SRG data is publicly available[1]. Similar to (Bodnar et al. 2021b,a; Bouritsas et al. 2023), we adopt the SRG benchmark, which poses as hard examples for the task of distinguishing non-isomorphic graphs, to support our theory behind PWL and PCN. Figure 4 illustrates the performance of PCN when lifting graphs to a dimension of 3, which is denoted by PCN(3). As we increase the number of layers for PCN, PCN outperforms CWN(4-IC), and achieves an on-par performance with CWN(5-IC), even though we do not explicitly include the inductive bias of ring-shaped structures. This result aligns with our theory that PWL can generalize CWL(k-IC) if the highest-order elementary p-paths align with the 2-cells with boundary size of $p + 1$, which can be empirically validated by comparing CWN(4-IC) and PCN(3). The result also illustrates that PWL takes a longer

[1] http://users.cecs.anu.edu.au/bdm/data/graphs.html

Dataset	ZINC		OGBG-MOLHIV	
	No Edge Feat.	W/ Edge Feat.	Test ROC-AUC	Val. ROC-AUC
GCN	0.469 ± 0.002	N/A	N/A	N/A
GAT	0.463 ± 0.002	N/A	N/A	N/A
GatedGCN	0.422 ± 0.006	0.363 ± 0.009	N/A	N/A
GIN	0.408 ± 0.008	0.252 ± 0.014	77.07 ± 1.49	84.79 ± 0.68
PNA	0.320 ± 0.032	0.188 ± 0.004	79.05 ± 1.32	85.19 ± 0.99
DGN	0.219 ± 0.010	0.168 ± 0.003	79.70 ± 0.97	84.70 ± 0.47
HIMP	N/A	0.151 ± 0.006	78.80 ± 0.82	N/A
GSN	0.140 ± 0.006	0.115 ± 0.012	77.99 ± 1.00	**86.58 ± 0.84**
PathNN	N/A	0.090 ± 0.004	79.17 ± 1.09	N/A
CIN †	**0.115 ± 0.003**	0.079 ± 0.006	**80.94 ± 0.57**	N/A
CIN++	N/A	**0.077 ± 0.004**	80.63 ± 0.94	N/A
PIN (Ours)	0.139 ± 0.004	0.096 ± 0.006	79.44 ± 1.40	82.41 ± 0.96

Table 2: ZINC and OGBG-MOLHIV datasets. Bold texts indicate the best performance. Performance on ZINC is evaluated by Mean Squared Error, while performance on OGBG-MOLHIV is evaluated by ROC-AUC. Baseline is denoted by †.

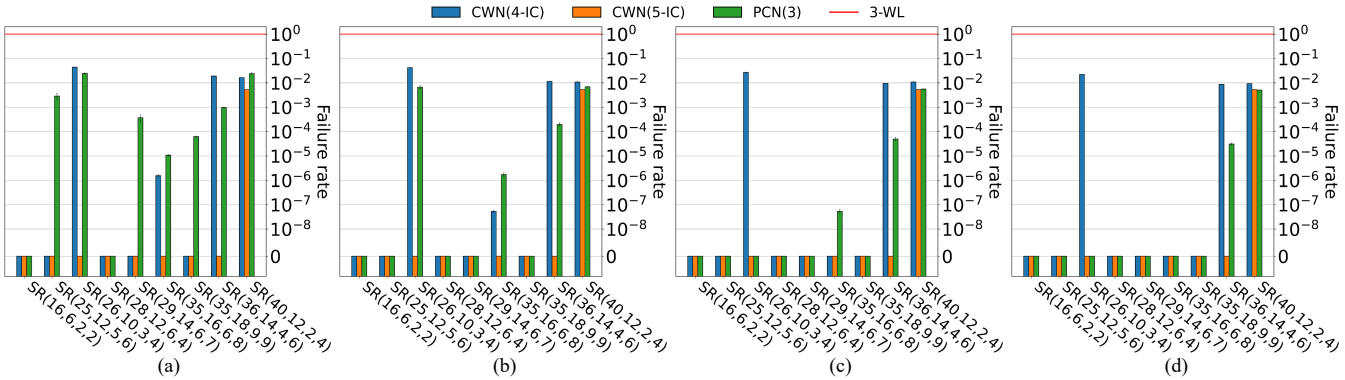

Figure 4: Failure rate comparison between CWN and PCN on SRG Families over 10 different seeds. (a) 3 message-passing layers. (b) 4 message-passing layers. (c) 5 message-passing layers. (d) 6 message-passing layers.

time to reach stable colorings than CWN does, as messages between two further edges of a ring must be passed through all dimensions of a path complex.

6 Discussion and Related Works

Higher-order GNNs Topological deep learning, which investigates beyond pair-wise interactions between vertices, has gained significant attention lately. Papillon et al. and Hajij et al. recently conducted a survey on this field, and classify learning domains as either part-whole relations (simplicial complex (Bodnar et al. 2021b; Ebli, Defferrard, and Spreemann 2020; Roddenberry, Glaze, and Segarra 2021) and cellular complex (Bodnar et al. 2021a; Giusti et al. 2023)), set-type relations (hypergraphs (Huang and Yang 2021; Feng et al. 2018; Yadati et al. 2019)) or both (combinatorial complex (Hajij et al. 2023)). Some methods such as (Maron et al. 2019a; Morris et al. 2019) are provably as powerful as the k-WL test (Grohe and Otto 2015; Grohe 2017), or even strictly more expressive (Morris, Rattan, and Mutzel 2020). However, these approaches carry a greater computational cost than ours, due to the necessity of k-tuple inputs, leading to potential loss of locality and an increased risk of overfitting. Bodnar et al. introduce message-passing models over simplicial complexes and cellular complexes, which are provably not less powerful than the 3-WL test. Our learning domain falls into part-whole relations, and our proposed model is proven to generalize SWL (Bodnar et al. 2021b) and CWL(k-IC) (Bodnar et al. 2021a) while still retaining locality.

Path-based Graph Learning Paths constitute a vital component of all graphs, providing valuable insights into the graphs. pathGCN (Eliasof, Haber, and Treister 2022) proposes to learn spatial operators from randomly sampled paths as a way to encode multi-level local neighborhoods. PathNN (Michel et al. 2023) updates a node representation based on paths with different lengths passing through that node. GSN (Bouritsas et al. 2023) explicitly encodes nodes or edges by counting the numbers of certain sub-structures that nodes or edges belong to. Bouritsas et al. empirically finds that counting paths with the maximum length of 6 is enough to distinguish all graphs in the SRG families; while counting paths with the maximum length of 3 is insufficient to distinguish any graphs. Our experiment shows that our model can tell apart almost every SRG family with a dimension of 3, which empirically proves that message passing between elementary paths have greater capabilities beyond

path counting.

Limitations Similar to (Bodnar et al. 2021b,a), our approach incurs high time and space complexities. A naive way to enumerate all paths with length k for a graph with n nodes has worst time complexity of $\mathcal{O}(n^{k+1})$. If we take the branching factor b of the graph into account, the worst time complexity is $\mathcal{O}(nb^k)$(Michel et al. 2023). In practice, the majority of graph datasets that we evaluated are sparse, and lifting them to the third dimension, or even higher, is feasible. The only exceptions are PROTEINS, IMDB-B, and SR(35,16,6,8). However, with path complexes with a dimension of only 3, we can achieve superior performance than our baselines on the SRG experiments even though theoretically we cannot represent the rings with a size above 4. For PROTEINS and IMDB-B, these datasets are relatively small (about 1,000 graphs), and thus lifting to high dimensions may cause overfitting. Unlike CWL (Bodnar et al. 2021a) where the furthest edges on the boundary of a ring can exchange messages after 2 propagations, PWL requires passing messages on every order of a path complex. Yet, increasing the number of message-passing layers can lead to over-smoothing (Oono and Suzuki 2020; Li, Han, and Wu 2018). Path complex can be visualized as TREE-NEIGHBORSMATCH problem proposed in (Alon and Yahav 2021), where each node can be considered as an elementary p-path having at least two children nodes, which are elementary $(p-1)$-paths on the boundary. Thus, over-squashing may occur for high-dimensional path complexes.

Future Works High-dimensional elementary paths are more prevalent and trivial than their counterparts, which are simplices and cells. However, high-dimensional path complexes are prone with over-smoothing (Oono and Suzuki 2020; Li, Han, and Wu 2018) and over-squashing (Alon and Yahav 2021), which have not yet been encountered in other topological domains. An in-depth study into these problems could unlock advancements in higher-order graph augmentation techniques such as dropout (Rong et al. 2020; Papp et al. 2021) or graph rewiring (Topping et al. 2022). PWL and PWN rely on spaces of simple paths $\mathcal{S}_p$, while the original path complex is well-defined on the regular space of boundary-invariant Ω_p, which suggests there may exist a more powerful representation of path complexes.

7 Conclusion

We have demonstrated how path complexes can be an alternative topological domain for simplicial and regular cell complexes. We verify, theoretically and experimentally, the generalization of path complexes over the other two topological domains. The versatility of our approach, without the necessity for inductive bias on graph substructures, underscores the centrality and universality of paths as the foundational elements of any graph.

8 Appendix

Please refer to the Appendix of the extended version[2] for the detailed proofs, formulae, experiments, computational analysis, and other resources.

[2]https://arxiv.org/abs/2308.06838

References

Alon, U.; and Yahav, E. 2021. On the Bottleneck of Graph Neural Networks and its Practical Implications. In *International Conference on Learning Representations*.

Atwood, J.; and Towsley, D. 2016. Diffusion-Convolutional Neural Networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, 2001–2009. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510838819.

Beaini, D.; Passaro, S.; L'etourneau, V.; Hamilton, W. L.; Corso, G.; and Lio', P. 2020. Directional Graph Networks. In *International Conference on Machine Learning*.

Biewald, L. 2020. Experiment Tracking with Weights and Biases. https://www.wandb.com/. Accessed: 2023-12-31.

Bodnar, C.; Frasca, F.; Otter, N.; Wang, Y.; Liò, P.; Montufar, G. F.; and Bronstein, M. 2021a. Weisfeiler and Lehman Go Cellular: CW Networks. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 2625–2640. Curran Associates, Inc.

Bodnar, C.; Frasca, F.; Wang, Y.; Otter, N.; Montufar, G. F.; Lió, P.; and Bronstein, M. 2021b. Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 1026–1037. PMLR.

Bouritsas, G.; Frasca, F.; Zafeiriou, S.; and Bronstein, M. 2023. Improving Graph Neural Network Expressivity via Subgraph Isomorphism Counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1): 657–668.

Bresson, X.; and Laurent, T. 2018. Residual Gated Graph ConvNets. arXiv:1711.07553.

Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2014. Spectral Networks and Locally Connected Networks on Graphs. In Bengio, Y.; and LeCun, Y., eds., *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Cai, J.-Y.; Furer, M.; and Immerman, N. 1989. An optimal lower bound on the number of variables for graph identification. In *30th Annual Symposium on Foundations of Computer Science*, 612–617.

Chen, Z.; Chen, L.; Villar, S.; and Bruna, J. 2020. Can Graph Neural Networks Count Substructures? In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 10383–10395. Curran Associates, Inc.

Clevert, D.; Unterthiner, T.; and Hochreiter, S. 2016. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). In Bengio, Y.; and LeCun, Y., eds., *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Corso, G.; Cavalleri, L.; Beaini, D.; Liò, P.; and Veličković, P. 2020. Principal Neighbourhood Aggregation for Graph Nets. In *Advances in Neural Information Processing Systems*.

de Haan, P.; Cohen, T.; and Welling, M. 2020. Natural Graph Networks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713829546.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In Lee, D.; Sugiyama, M.; Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Dwivedi, V. P.; Joshi, C. K.; Luu, A. T.; Laurent, T.; Bengio, Y.; and Bresson, X. 2023. Benchmarking Graph Neural Networks. *Journal of Machine Learning Research*, 24(43): 1–48.

Ebli, S.; Defferrard, M.; and Spreemann, G. 2020. Simplicial Neural Networks. In *TDA & Beyond*.

Eliasof, M.; Haber, E.; and Treister, E. 2022. pathGCN: Learning General Graph Spatial Operators from Paths. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvari, C.; Niu, G.; and Sabato, S., eds., *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 5878–5891. PMLR.

Feng, Y.; You, H.; Zhang, Z.; Ji, R.; and Gao, Y. 2018. Hypergraph Neural Networks. *AAAI 2019*.

Fey, M.; and Lenssen, J. E. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Fey, M.; Yuen, J. G.; and Weichert, F. 2020. Hierarchical Inter-Message Passing for Learning on Molecular Graphs. In *ICML Graph Representation Learning and Beyond (GRL+) Workhop*.

Gärtner, T.; Flach, P.; and Wrobel, S. 2003. On Graph Kernels: Hardness Results and Efficient Alternatives. In Schölkopf, B.; and Warmuth, M. K., eds., *Learning Theory and Kernel Machines*, 129–143. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-45167-9.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, 1263–1272. JMLR.org.

Giusti, L.; Battiloro, C.; Testa, L.; Lorenzo, P. D.; Sardellitti, S.; and Barbarossa, S. 2022. Cell Attention Networks. arXiv:2209.08179.

Giusti, L.; Reu, T.; Ceccarelli, F.; Bodnar, C.; and Liò, P. 2023. CIN++: Enhancing Topological Message Passing. arXiv:2306.03561.

Grigor'yan, A.; Lin, Y.; Muranov, Y.; and Yau, S.-T. 2013. Homologies of path complexes and digraphs. ArXiv:1207.2834 [math].

Grigor'yan, A. A.; Lin, Y.; Muranov, Y. V.; and Yau, S.-T. 2020. Path Complexes and their Homologies. *Journal of Mathematical Sciences*, 248(5): 564–599.

Grohe, M. 2017. *Descriptive Complexity, Canonisation, and Definable Graph Structure Theory*. Lecture Notes in Logic. Cambridge University Press.

Grohe, M.; and Otto, M. 2015. Pebble Games and Linear Equations. *The Journal of Symbolic Logic*, 80(3): 797–844.

Hagberg, A. A.; Schult, D. A.; and Swart, P. J. 2008. Exploring Network Structure, Dynamics, and Function using NetworkX. In Varoquaux, G.; Vaught, T.; and Millman, J., eds., *Proceedings of the 7th Python in Science Conference*, 11 – 15. Pasadena, CA USA.

Hajij, M.; Zamzmi, G.; Papamarkou, T.; Miolane, N.; Guzmán-Sáenz, A.; Ramamurthy, K. N.; Birdal, T.; Dey, T. K.; Mukherjee, S.; Samaga, S. N.; Livesay, N.; Walters, R.; Rosen, P.; and Schaub, M. T. 2023. Topological Deep Learning: Going Beyond Graph Data. arXiv:2206.00606.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020. Open Graph Benchmark: Datasets for Machine Learning on Graphs. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713829546.

Huang, J.; and Yang, J. 2021. UniGNN: a Unified Framework for Graph and Hypergraph Neural Networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*.

Huang, N. T.; and Villar, S. 2021. A Short Tutorial on The Weisfeiler-Lehman Test And Its Variants. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Li, Q.; Han, Z.; and Wu, X.-M. 2018. Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press. ISBN 978-1-57735-800-8.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Maron, H.; Ben-Hamu, H.; Serviansky, H.; and Lipman, Y. 2019a. Provably Powerful Graph Networks. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; Alché-Buc, F. d.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Maron, H.; Ben-Hamu, H.; Shamir, N.; and Lipman, Y. 2019b. Invariant and Equivariant Graph Networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Michel, G.; Nikolentzos, G.; Lutzeyer, J.; and Vazirgiannis, M. 2023. Path Neural Networks: Expressive and Accurate Graph Neural Networks. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*.

Morris, C.; Kriege, N. M.; Bause, F.; Kersting, K.; Mutzel, P.; and Neumann, M. 2020. TUDataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*.

Morris, C.; Rattan, G.; and Mutzel, P. 2020. Weisfeiler and Leman Go Sparse: Towards Scalable Higher-Order Graph Embeddings. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713829546.

Morris, C.; Ritzert, M.; Fey, M.; Hamilton, W. L.; Lenssen, J. E.; Rattan, G.; and Grohe, M. 2019. Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press. ISBN 978-1-57735-809-1. Event-place: Honolulu, Hawaii, USA.

Neumann, M.; Garnett, R.; Bauckhage, C.; and Kersting, K. 2016. Propagation kernels: efficient graph kernels from propagated information. *Mach. Learn.*, 102(2): 209–245.

Oono, K.; and Suzuki, T. 2020. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. In *International Conference on Learning Representations*.

OpenAI. 2022. GPT-4. https://www.openai.com/gpt-4/. Accessed: 2023-12-31.

Papillon, M.; Sanborn, S.; Hajij, M.; and Miolane, N. 2023. Architectures of Topological Deep Learning: A Survey on Topological Neural Networks. ArXiv:2304.10031 [cs].

Papp, P. A.; Martinkus, K.; Faber, L.; and Wattenhofer, R. 2021. DropGNN: Random Dropouts Increase the Expressiveness of Graph Neural Networks. In *35th Conference on Neural Information Processing Systems (NeurIPS)*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, 8024–8035. Curran Associates, Inc.

Peixoto, T. P. 2014. The graph-tool python library. *figshare*.

Roddenberry, T. M.; Glaze, N.; and Segarra, S. 2021. Principled Simplicial Neural Networks for Trajectory Prediction. In Meila, M.; and 0001, T. Z., eds., *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, 9020–9029. PMLR.

Rong, Y.; Huang, W.; Xu, T.; and Huang, J. 2020. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. In *International Conference on Learning Representations*.

Schaub, M. T.; Benson, A. R.; Horn, P.; Lippner, G.; and Jadbabaie, A. 2020. Random Walks on Simplicial Complexes and the normalized Hodge 1-Laplacian. *SIAM Review*, 62(2): 353–391. ArXiv:1807.05044 [physics].

Shervashidze, N.; Schweitzer, P.; van Leeuwen, E. J.; Mehlhorn, K.; and Borgwardt, K. M. 2011. Weisfeiler-Lehman Graph Kernels. *Journal of Machine Learning Research*, 12(77): 2539–2561.

Shervashidze, N.; Vishwanathan, S.; Petri, T.; Mehlhorn, K.; and Borgwardt, K. 2009. Efficient Graphlet Kernels for Large Graph Comparison. *12th International Conference on Artificial Intelligence and Statistics (AISTATS), Society for Artificial Intelligence and Statistics, 488-495 (2009)*, 5.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(56): 1929–1958.

Sterling, T.; and Irwin, J. J. 2015. ZINC 15 – Ligand Discovery for Everyone. *Journal of Chemical Information and Modeling*, 55(11): 2324–2337.

Topping, J.; Giovanni, F. D.; Chamberlain, B. P.; Dong, X.; and Bronstein, M. 2022. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations (ICLR)*. ArXiv: 1710.10903 version: 3.

Weisfeiler, B.; and Lehman, A. 1968. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsiya*.

Wu, Z.; Ramsundar, B.; Feinberg, E.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; and Pande, V. 2018. MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.*, 9: 513–530.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*.

Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; Kawarabayashi, K.-i.; and Jegelka, S. 2018. Representation Learning on Graphs with Jumping Knowledge Networks. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 5453–5462. PMLR.

Yadati, N.; Nimishakavi, M.; Yadav, P.; Nitin, V.; Louis, A.; and Talukdar, P. 2019. HyperGCN: A New Method For Training Graph Convolutional Networks on Hypergraphs. In *Advances in Neural Information Processing Systems (NeurIPS) 32*, 1509–1520. Curran Associates, Inc.

Zhang, M.; Cui, Z.; Neumann, M.; and Chen, Y. 2018. An End-to-End Deep Learning Architecture for Graph Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).