PATH COMPLEX MESSAGE PASSING FOR MOLECULAR PROPERTY PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Geometric deep learning (GDL) has demonstrated enormous power in molecular data analysis. However, GDL faces challenges in achieving high efficiency and expressivity in molecular representations when high-order terms of the atomic force fields are not sufficiently learned. In this work, we introduce message passing on path complexes, called the Path Complex Message Passing, for molecular prediction. Path complexes represent the geometry of paths and can model the chemical and non-chemical interactions of atoms in a molecule across various dimensions. Our model defines messages on path complexes and employs neural message passing to learn simplex features, enabling feature communication within and between different dimensions. Since messages on high-order and low-order path complexes reflect different aspects of molecular energy, they are updated sequentially according to their order. The higher the order of the path complex, the richer the information it contains, and the higher its priority during inference. It can thus characterize various types of molecular interactions specified in molecular dynamics (MD) force fields. Our model has been extensively validated on benchmark datasets and achieves state-of-the-art results. The code is available at `https://anonymous.4open.science/r/Path-Complex-Neural-Network-32D6`

1 INTRODUCTION

Accurate prediction of molecular properties is crucial in fields such as drug design Zhang et al. (2017); Chen et al. (2018); Mak & Pichika (2019); Chan et al. (2019), biology Townshend et al. (2021); Jamasb et al. (2022), chemistry Qiao et al. (2022), and materials science Vlassis et al. (2020). Geometric Deep Learning (GDL) has demonstrated significant potential in molecular sciences, leading to a surge in studies employing GDL models for effective molecular representation learning Bronstein et al. (2017); Atz et al. (2021); Ingraham et al. (2023). Among the three types of representations used in GDL models—topological, geometric, and functional—the molecular graph has become the most popular due to its simplicity, flexibility, and efficiency Wieder et al. (2020); Yu & Gao (2022); Atz et al. (2021); Li et al. (2022); Wang et al. (2022b). However, relying solely on graph representations fails to capture the many-body interactions inherent in complex systems, thereby limiting the expressiveness and predictive power of this approach Bodnar et al. (2021b). This paper develops a path complex-based neural message passing for molecule prediction, where the molecular energy of force field can be well represented.

In Graph Neural Networks (GNNs), the molecular graph is typically constructed based on covalent bonds. Node features are usually derived from atomic properties and are updated by aggregating information from neighboring nodes Huang et al. (2020); Shindo & Matsumoto (2019); Shui & Karypis (2020a); Schütt et al. (2017); Unke & Meuwly (2019). To enhance GNN performance, researchers have proposed several approaches. One major strategy is to design more complex molecular graphs that incorporate non-covalent interactions. The most common method involves introducing edges between any two atoms within a specified cutoff distance, effectively capturing non-covalent interactions. Additionally, molecule-based line graph models have been developed, where nodes represent atomic bonds and edges represent bond angles Choudhary & DeCost (2021).

The second approach focuses on incorporating global physical features and local geometric information into GNN models. Global physical attributes such as temperature, pressure, and entropy

have been added to GNN architectures to better characterize molecular states and environments, as demonstrated in MEGNet Chen et al. (2019) and SphereNet Liu et al. (2022). Local geometric features—particularly bond lengths, bond angles Schütt et al. (2018); Flam-Shepherd et al. (2021), dihedral angles Wang et al. (2022a), and torsion angles, which are crucial to molecular properties—have been extensively considered in models such as DimeNet Gasteiger et al. (2020), GemNet Gasteiger et al. (2021), ALIGNN Choudhary & DeCost (2021), and GEM Fang et al. (2022).

Another approach involves designing efficient message-passing modules for invariant features, equivariant properties, and higher-order tensors. The expressivity of GNNs is closely related to the message-passing mechanisms used in layers that process invariant, equivariant, or higher-order tensor features. These three approaches are often synergistically integrated to enhance model performance.

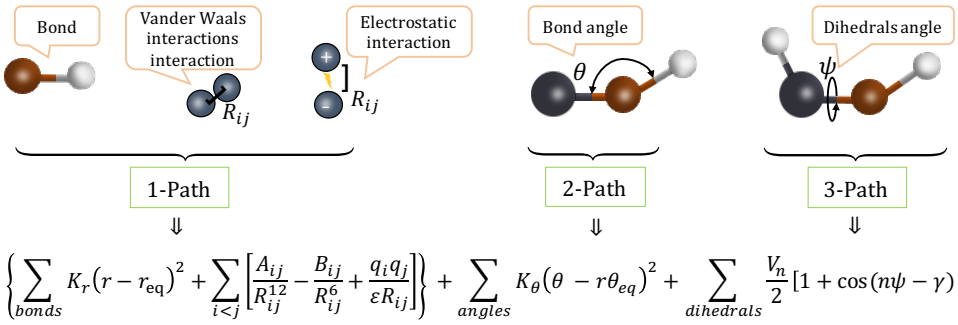


Figure 1: Terms of the approximate equation to molecular dynamics force field correspond to path complices of order one to three, which have been used in path complex message passing.

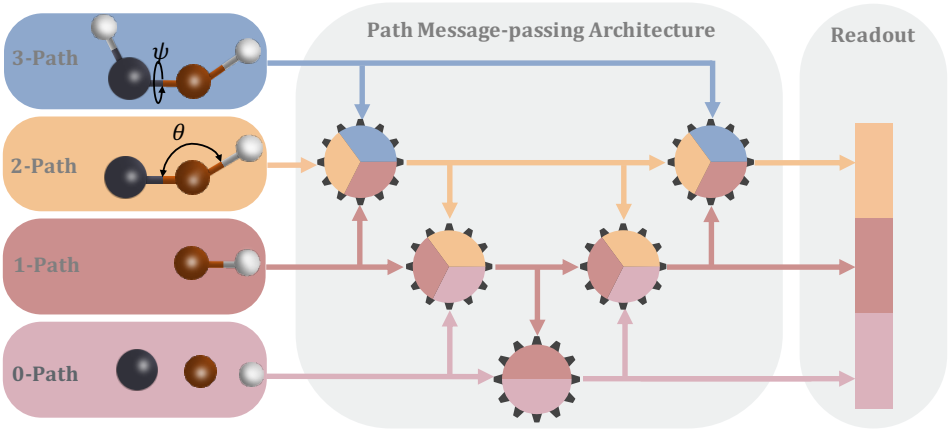


Figure 2: The architecture of PCMP utilizing path complexes up to order 3 is depicted. At each layer l, each path complex message of a given order updates its features using messages from path complexes of the same order and adjacent orders from the previous layer $l-1$. Higher-order path complex messages are updated before lower-order ones because the former encompass the paths of the latter. Additionally, the interplay between high-order and low-order path complexes is learned through message passing.

In this work, we develop path complex-based molecular representation and path complex message passing (PCNN) model for molecular property analysis. PCNN is a neural message passing Gilmer et al. (2017) on path complices. A path is a sequence of points, and a path complex is a subset of all possible paths. In the context of a molecule or molecular graph, a path corresponds to the geometry defined by chemical or non-chemical bonds. Our path complexes are specifically designed — based on molecular graphs that include both covalent and non-covalent bonds — to characterize different types of energy specified in molecular dynamics (MD) force fields, as shown in Figure 1. The MD potential energy Mayo et al. (1990); González (2011); Leach (2001) comprises bond terms (E_B,

two-body interactions), bond-angle terms (E_A, three-body interactions), and dihedral-angle terms (E_T, four-body interactions), which are effectively characterized by our 1-path, 2-path, and 3-path features, respectively.

Each path complex is assigned a message. Similar to classical neural message passing on graphs Gilmer et al. (2017) and simplicial complexes Bodnar et al. (2021a;b), the propagation of messages in a path complex is influenced by the messages of its "adjacent" path complexes at different orders. A higher-order path complex contains longer paths and includes the shorter paths of lower-order path complexes. Therefore, we need to update messages according to the order of the path complexes: messages in higher-order path complexes have priority in being updated. However, since path complexes of adjacent orders are interconnected, we incorporate interactions between higher-order and lower-order path complex information during message passing. Specifically, the higher-order message first updates the lower-order one, and then the updated lower-order message exerts a reverse effect on the higher-order information. Figure 2 illustrates a neural message passing process among path complexes of different orders designed based on this principle.

PCNN thus enables information passing between path complex features, using the aggregated information to predict molecular properties. Testing on benchmark datasets demonstrates promising performance. Our contributions are as follows:

1. We have developed a path complex-based molecular representation that explicitly characterizes different terms in the molecular dynamics (MD) force field.
2. We propose constructing path complexes using the unique topologies of graphs, simplicial complexes, and hypergraphs. This method enables systematic exploration of connectivity and interaction, offering a powerful tool for analyzing complex systems and networks.
3. Our PCMP model has been rigorously tested and validated on benchmark molecular tasks, consistently achieving state-of-the-art results.

2 Related Work

Graph Neural Networks for Molecular Property Prediction Graph neural network models have played an pivotal role in molecular data analysis. Traditional GNN models represent molecules as the de factor covalent-bond-based molecular graphs, and use major GNN architectures, such as GIN Xu et al. (2018), GAT Velickovic et al. (2017), GCN Kipf & Welling (2016a), SGCN Danel et al. (2020) and GTtransformer Rong et al. (2020), to learn molecular properties Yang et al. (2019); Xiong et al. (2019); Choudhary & DeCost (2021); Fang et al. (2022). With the importance of non-covalent bonds, cutoff-distance-based molecular graph representations have been widely employed in GNN models, such as DimeNet Gasteiger et al. (2020), HMGNN Shui & Karypis (2020b), GeoGNN Fang et al. (2022), Mol-GDL Shen et al. (2023), etc. Further, higher-order interactions (beyond pair-wise forces) has been explicitly incorporated into GNN models, including ALIGNN Choudhary & DeCost (2021), GEM Fang et al. (2022), DimeNet Gasteiger et al. (2020), GemNet Gasteiger et al. (2021), etc, by the consideration of bond angles, dihedral angles, torsion angles, and other local geometric information. In particular, these higher-order terms can be directly related to MD force field information Halgren (1996); Choudhary et al. (2018). Finally, pre-training process has been adopted to further improve the accuracy of GNN models, such as N-Gram Liu et al. (2019), PretrainGNN Hu et al. (2019), GEM Fang et al. (2022), MolCLR Wang et al. (2022b), DMP Zhu et al. (2023), etc.

Topological Deep Learning (TDL) Topological Deep Learning (TDL) Hajij et al. (2022); Bodnar (2022) leverages novel topological tools to characterize data with complicated higher-order structures. Different from graph-based data representation, TDL uses topological representations from algebraic topology, including simplicial complexes Bodnar (2022); Schaub et al. (2022), cell complexes Hajij et al. (2020); Roddenberry et al. (2022); Giusti et al. (2023), sheaves Hansen & Ghrist (2019); Bodnar et al. (2021b), hypergraphs Feng et al. (2019); Kim et al. (2020); Bai et al. (2021), and combinatorial complexes Hajij et al. (2022) to model not only pair-wise interactions (as in graphs), but also higher-order interactions among three or more elements. In fact, these algebraic topology-based molecular representations have already achieved great success in molecular data analysis, including protein flexibility and dynamic analysis Xia & Wei (2014); Sverrisson et al.

(2021), drug design Cang & Wei (2017), virus analysis Chen et al. (2022), materials property analysis Reiser et al. (2022); Townsend et al. (2020). Further, TDL uses a generalized message-passing mechanism thus enables the communication of information from simplices of different dimensions. In contrast to GNNs, where information is passing among nodes or edges, TDL allows information to propagate through any neighborhood relation Roddenberry et al. (2021).

Recently, path complex and its related models, including path homology Grigor'yan et al. (2018), persistent path homology Chowdhury & Mémoli (2018); Liu et al. (2023); Chen et al. (2023), path Laplacian Wang & Wei (2023), a special path-complex-based topological message passing model Truong & Chin (2024) has been developed and demonstrated great potential for the analysis of molecular structures.

Geometric Deep Learning and Molecular Representation Generally speaking, molecules in GDL models are characterized by three types of molecular representations, including topological representations (such as molecular graphs), geometric representation (such as molecular surfaces), and function representation (such as molecular density). Deep learning models including (3D) convolutional neural networks, graph neural networks (GNNs), recurrent neural networks, and others, have been constructed based on these representations Wieder et al. (2020); Yu & Gao (2022); Atz et al. (2021); Li et al. (2022); Wang et al. (2022b). With its simplicity, flexibility and efficiency, molecular graphs are the most popular of various types of GNN models have been proposed, including graph recurrent neural networks (GraphRNN) You et al. (2018), graph convolutional networks (GCN) Welling & Kipf (2016), graph autoencoders Kipf & Welling (2016b), graph transformers Yun et al. (2019), etc. These GNN models have been widely used in molecular data analysis.

3 Path Complex Message Passing

Path complex was originally developed on directed graph (or digraph) and set, by Grigoryan, Lin, Muranov and Yau in 2012 Grigor'yan et al. (2012). They also proposed a new homology theory for path complex, called path homology, and use it to explore topological invariant information of digraphs Grigor'yan et al. (2014). Mathematically, path homology provides a novel framework to systematically explore intrinsic topological information of more general structures Grigor'yan et al. (2019); Grigor'yan et al. (2020). Details of path complex and path homology can be found in the Appendix B.

Here we propose a generalized way to construct path complex based on undirected graph, simplicial complex, and hypergraph. On undirected graph, we propose graph collapse and expansion operations, and use them to systematically study graph isomorphism by their path complex homology groups. We found that the path complex homology is a graph weak isomorphism invariant. For simplicial complex and hypergraph, we propose simplex- and hyperedge- based path complex.

3.1 Generalized Path Complex

Path complex for undirected-graph Firstly, we give the construction of path complex for undirected graphs. Secondly, we introduce the graph weak isomorphism and related mathematical properties. Finally, we states the weak isomorphism invariance of path complex homology for graphs.

Definition 3.1 (Path)**.** Given a simple undirected graph $G = (V, E)$ over the verset set V, an n-path σ_n of G is defined as any sequence of $n+1$ vertices $v_0 v_1 \cdots v_n (v_i \in V)$ such that every two vertices are distinct and every two adjacent vertices form an edge.

Note that for each n-path $\sigma_n = v_0 v_1 \cdots v_n$, $\sigma'_n = v_n \cdots v_1 v_0$ is also an n-path, we identify these two paths as the same one. For an n-path $\sigma_n = v_0 \cdots v_n$, the $(n-1)$-paths by removing the first or last vertex, denoted by $\partial^L_{\sigma_n}$ and $\partial^R_{\sigma_n}$ respectively, are called the faces of σ_n. Two n-paths are neighbors if they are faces of a common $(n+1)$-path. Let $\mathcal{N}(\sigma_n)$ be the set of neighbors of σ_n.

Definition 3.2 (Path complex from undirected graphs)**.** Given a simple undirected graph $G = (V, E)$, all paths of G form a path complex P_G. We call P_G the path complex derived from G.

Definition 3.3 (Graph collapse and expansion)**.** Given a graph $G = (V, E)$, take an edge $(v_1, v_2) \in E$ such that $deg(v_1) = 1$. Let $V' = V \backslash \{v_1\}$, $E' = E \backslash \{(v_1, v_2)\}$, then $G' = (V', E')$ is a new graph. We say that G' is derived from G by a graph collapse and G is derived from G' by a graph expansion.

Definition 3.4 (Weak isomorphism)**.** Given two graphs G_1, G_2, G_1 and G_2 are called weak isomorphic if G_1 can be derive from G_2 by a sequence of graph collapse and expansion operations.

It can be seen that two graphs are weak isomorphic if they are isomorphic.

Theorem 3.5. *If two graphs G_1 and G_2 are weak isomorphic, then, 1) The number of connected components of G_1 and G_2 are same; 2) The number of cycles of G_1 and G_2 are same.*

Theorem 3.6. *Given two graphs G_1, G_2, let P_{G_1}, P_{G_2} be the path complexs derived from G_1 and G_2 respectively. If G_1 and G_2 are weak isomorphic, then*

$$\mathrm{H}_k(P_{G_1}) \cong \mathrm{H}_k(P_{G_2}) \ (k \geqslant 0)$$

Theorem 3.6 means the path complex homology is a graph weak isomorphism invariant. Consequently, for two graphs G_1 and G_2, if there exists k such that $\mathrm{H}_k(P_{G_1}) \ncong \mathrm{H}_k(P_{G_2})$, then G_1 and G_2 are not weak isomorphic and not isomorphic.

The profound theoretical relationship between the Weisfeiler-Lehman (WL) graph isomorphism test and message-passing graph neural networks (GNNs) has been extensively documented Xu et al. (2018).

Definition 3.7 (PWL)**.** The steps of general PWL are as follows:

1. Given a path complex P, all the paths of P are initialized with the same color.
2. For the color c_σ^t of path σ at iteration t, the color c_σ^{t+1} of σ at the next iteration is computed by perfectly hashing the color multi-set of the neighbors of σ.
3. The algorithm stops once a stable coloring is reached. Two path complexes are considered non-isomorphic if their color histograms are different at some dimensions.

Based on the four neighbor definitions, including face neighbor $\mathcal{B}(\sigma)$, coface neighbor $\mathcal{C}(\sigma)$, upper adjacent neighbor $\mathcal{N}_\uparrow(\sigma)$ and lower adjacent neighbor $\mathcal{N}_\downarrow(\sigma)$, we have four types of neighbor color multi-sets. Let c^t be the coloring of PWL for path complex P at iteration t, four types of color multi-sets are as follows

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

Having the neighbor color multi-sets, we obtain the following update rule that contains all four types of neighbors:

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

Actually, certain neighbors can be removed without affecting the expressive power of PWL test in terms of path complex that can be differentiated.

Theorem 3.8. *PWL with* $\mathrm{HASH}\{c_\sigma^t, c_{\mathcal{B}}^t(\sigma), c_\uparrow^t(\sigma)\}$ *is as powerful as PWL with the updating strategy* $\mathrm{HASH}\{c_\sigma^t, c_{\mathcal{B}}^t(\sigma), c_{\mathcal{C}}^t(\sigma), c_\uparrow^t(\sigma), c_\downarrow^t(\sigma)\}$.

Theorem 3.9. *PWL is strictly more powerful than WL.*

Figure 7 shows two graphs that cannot be distinguished by the WL test, but their derived path complexes can be distinguished by PWL.

Path Complex for simplicial complex and hypergraph Generally speaking, a path complex is a set of paths that is closed under the removing of the first or last vertices of each path. So we can also construct path complex from simplicial complex and hypergraph by defining simplex-paths and hyperedge-paths. Various kinds of paths can be defined by considering the lower adjacent, upper adjacent, face and coface relations among simplices and hyperedges. Figure 3 shows examples of path complexes constructed from graph, simplicial complex and hypergraph. Details can be found in Appendix B.1.

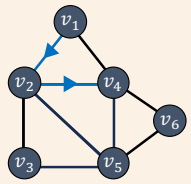
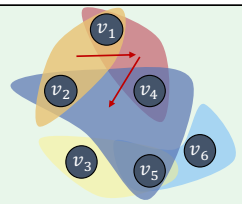
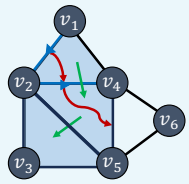
Path complex			 Path types in Simplicial complex: vertex-path, edge-path, triangle-path		
	Graph path complex	Hypergraph path complex	Simplicial path complex		
			Vertex-path	Edge-path	Triangle-path
{0-path}	$\{v_1\}$	$\{v_1 v_2\}$	$\{v_1\}$	$\{v_1 v_2\}$	$\{v_1 v_2 v_4\}$
{1-path}	$\{v_1 - v_2\}$	$\{v_1 v_2 - v_1 v_4\}$	$\{v_1 - v_2\}$	$\{v_1 v_2 - v_2 v_4\}$	$\{v_1 v_2 v_4 - v_2 v_4 v_5\}$
{2-path}	$\{v_1 - v_2 - v_4\}$	$\{v_1 v_2 - v_1 v_4 - v_2 v_4 v_5\}$	$\{v_1 - v_2 - v_4\}$	$\{v_1 v_2 - v_2 v_4 - v_4 v_5\}$	$\{v_1 v_2 v_4 - v_2 v_4 v_5 - v_2 v_3 v_5\}$

Figure 3: The path complexes of graphs, simplicial complexes, and hypergraphs. The table lists the 0-paths, 1-paths, and 2-paths, where the red arrows indicate the selected 2-paths. Specifically, for simplicial complexes, we enumerate the path complexes from vertixes (0-simplices), edges (1-simplices) and triangles (2-simplices), respectively.

3.2 Molecular Path Complex Representation and Path Features

Molecular Path Complex Representation Currently, covalent-bond molecular graphs serve as the standard for molecular topological representations. These graphs underpin the molecular force fields used in molecular dynamics simulations, incorporating terms for both covalent bonds—such as bond lengths, angles, and dihedral angles—and non-covalent interactions like electrostatic and van der Waals forces. To enhance molecular representations with comprehensive force field data, we introduce the molecular path complex. This model utilizes path simplices across different dimensions to distinctly represent both covalent and non-covalent bond terms. As depicted in Figure 4, the C_2H_6O molecule is illustrated alongside its corresponding path simplices. Specifically, our 1-path simplex captures bond lengths, the 2-path simplex details bond angles, and the 3-path simplex reflects dihedral angles.

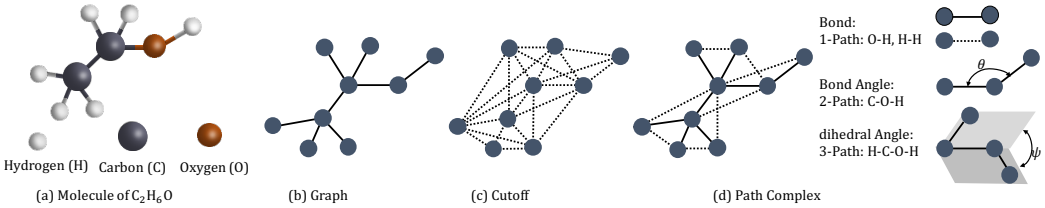


Figure 4: Different Representations of the C_2H_6O Molecule. (a) displays the molecular structure of C_2H_6O, including the oxygen (O), carbon (C), and hydrogen (H) atoms. (b) shows the graph representation based on chemical bonds. (c) illustrates the nearly fully connected graph generated based on a distance threshold (cutoff). (d) presents the representation using the path complex method and its physical implications. In the diagrams, solid lines represent chemical bonds, while dashed lines represent cutoff connections.

Path Features Our path (simplex) features are meticulously designed to encapsulate the various atomic properties and interactions detailed in molecular dynamics (MD) force fields. Specifically, our 0-path features—comprising atomic number, radius, and electronegativity—are derived using Rdkit, akin to the approach in CGCNN Xie & Grossman (2018). Table 5 (in Appendix A.1) presents a comprehensive listing of our 1-path, 2-path, and 3-path features. Importantly, our model employs detailed local geometric properties of the path complex as path features. This method allows us to explicitly learn covalent bond terms defined in the MD force fields, while also implicitly capturing non-bond interactions.

3.3 MOLECULAR PCMP MODEL

Path Complex Message Passing (PCMP) introduces a novel method for message passing in graphs by leveraging path complexes, which are composed of paths of varying lengths. In contrast to traditional graph neural networks (GNNs) that primarily aggregate information from local neighbors, PCMP prioritizes message propagation along higher-order path complexes. By incorporating longer paths, PCMP effectively captures long-range dependencies within the graph, enhancing its ability to model complex relationships.

A key feature of PCMP is the hierarchical message passing mechanism between different orders of path complexes. First, messages in higher-order paths are updated, reflecting the broader structure of the graph. These updated messages are then propagated to lower-order paths, ensuring that global information from longer paths informs to lower-order paths. After this, a feedback mechanism is employed, where updated messages from lower-order paths influence the higher-order paths, thus refining the representation at all levels. This bidirectional interaction between higher- and lower-order path allows PCMP to seamlessly integrate both global and local information effectively.

Path message-passing module A central component of our PCMP model is path (simplex) Grigor'yan et al. (2024) message-passing module, where path features are updates based on path neighbors (same order paths), cofaces (higher-order paths), and faces (lower-order paths). Mathematically, each n-path will always have two unique $(n-1)$-faces, but many n-path neighbors and $(n+1)$-cofaces. In our PCMP framework, the simplex message-passing module contains two parts, i.e., message embedding and message updating. Two message embedding modules, i.e., upper embedding and lower embedding, are considered. In upper embedding module, path message will be generated using its neighbors and cofaces, while for lower embedding module, path messages will be generated using its neighbors and faces. The path feature will be updated using path messages from both upper and lower embedding through a message updating module. An illustration of our PCMP module is shown in Figure 5.

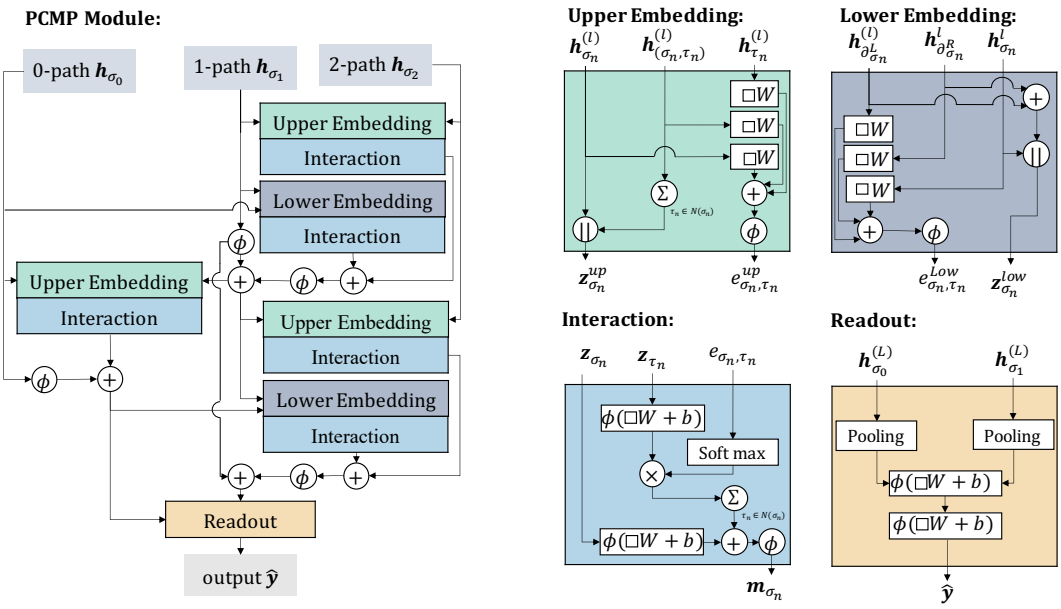


Figure 5: The PCMP Module. □ denotes the layer's input, ‖ concatenation, and ϕ a non-linearity. Upper embedding and Upper interaction refer to utilizing high-order path features to update low-order path features, while Lower embedding and Lower interaction refer to using low-order path features to update high-order path features.

The upper embedding module generates path message based on path neighbors and cofaces. For an n-path σ_n and its neighbors τ_n, we denote their path feature vectors as $\boldsymbol{h}_{\sigma_n}$ and $\boldsymbol{h}_{\tau_n}$ respectively. The common coface of σ_n and τ_n is denoted as (τ_n, σ_n) and $\boldsymbol{h}_{(\tau_n,\sigma_n)}$ the associate path feature. The

upper attention score $e^{up}_{\sigma_n,\tau_n}$ and upper concatenated feature $\boldsymbol{z}^{up}_{\sigma_n}$ can be expressed as,

$$\boldsymbol{f}_{\sigma_n} = \boldsymbol{h}_{\sigma_n}\mathbf{W}^{up}_n, \quad \boldsymbol{f}_{\tau_n} = \boldsymbol{h}_{\tau_n}\mathbf{W}^{up}_n, \quad \boldsymbol{f}_{(\sigma_n,\tau_n)} = \boldsymbol{h}_{(\sigma_n,\tau_n)}\mathbf{W}_{n+1},$$
$$e^{up}_{\sigma_n,\tau_n} = \mathrm{ReLU}(\boldsymbol{f}_{\sigma_n} + \boldsymbol{f}_{\tau_n} + \boldsymbol{f}_{(\sigma_n,\tau_n)}), \quad \boldsymbol{z}^{up}_{\sigma_n} = [\, \boldsymbol{h}_{\sigma_n} \parallel \sum_{\tau_n \in \mathcal{N}(\sigma_n)} \frac{1}{|\mathcal{N}(\sigma_n)|}\boldsymbol{h}_{(\sigma_n,\tau_n)} \,],$$

where ReLU is a non-linear activation function and $\mathbf{W}^{up}_n$ and $\mathbf{W}_{n+1}$ are weight matrices. Note that $\parallel$ is the concatenation operator, $\mathcal{N}(\sigma_n)$ denotes the neighbors of path σ_n, and $|\mathcal{N}(\sigma_n)|$ is the total number of neighbors of path σ_n.

The lower embedding module generates path message based on path neighbors and faces. For an n-path σ_n, we use $\partial^R_{\sigma_n}$ and $\partial^L_{\sigma_n}$ to represent its right and left faces. The lower attention score $e^{low}_{\sigma_n,\tau_n}$ and lower concatenated feature $\boldsymbol{z}^{low}_{\sigma_n}$ can be expressed as,

$$\boldsymbol{f}_{\sigma_n} = \boldsymbol{h}_{\sigma_n}\mathbf{W}^{low}_n, \quad \boldsymbol{f}_{\partial^L_{\sigma_n}} = \boldsymbol{h}_{\partial^L_{\sigma_n}}\mathbf{W}_{n-1}, \quad \boldsymbol{f}_{\partial^R_{\sigma_n}} = \boldsymbol{h}_{\partial^R_{\sigma_n}}\mathbf{W}_{n-1},$$
$$e^{low}_{\sigma_n,\tau_n} = \mathrm{ReLU}(\boldsymbol{f}_{\partial^L_{\sigma_n}} + \boldsymbol{f}_{\partial^R_{\sigma_n}} + \boldsymbol{f}_{\sigma_n}), \quad \boldsymbol{z}^{low}_{\sigma_n} = [\, (\boldsymbol{h}_{\partial^L_{\sigma_n}} + \boldsymbol{h}_{\partial^R_{\sigma_n}}) \parallel \boldsymbol{h}_{\sigma_n} \,],$$

where $\mathbf{W}^{low}_n$ and $\mathbf{W}_{n-1}$ are weight matrices.

The path feature is updated by using message from both upper embedding and low embedding. First, upper and lower path message is generated from the upper embedding and low embedding respectively as follows,

$$\boldsymbol{c}^{up/low}_{\sigma_n} = \mathrm{ReLU}(\boldsymbol{z}^{up/low}_{\sigma_n}\mathbf{W}^{up/low}_n + b^{up/low}), \quad \alpha^{up/low}_{\sigma_n,\tau_n} = \frac{e^{up/low}_{\sigma_n,\tau_n}}{\sum_{\kappa_n \in \mathcal{N}(\sigma_n)} e^{up/low}_{\sigma_n,\kappa_n}},$$
$$\boldsymbol{m}^{up/low}_{\sigma_n} = \mathrm{LeakyReLU}(\boldsymbol{c}^{up/low}_{\sigma_n} + \sum_{\tau_n \in \mathcal{N}(\sigma_n)} \alpha^{up/low}_{\sigma_n,\tau_n}\boldsymbol{c}^{up/low}_{\tau_n}),$$

then path feature is updated by using both messages as follows,

$$\boldsymbol{h}^{(l+1)}_{\sigma_n} = \mathrm{LeakyReLU}((\boldsymbol{m}^{low}_{\sigma_n})^{(l)} + (\boldsymbol{m}^{up}_{\sigma_n})^{(l)}).$$

Note that $h^{(l+1)}_{\sigma_n}$ means the updated feature vector for n-path σ_n at the $(l+1)$-th layer. It depends on the upper and lower message information at the l-th layer.

Theorem 3.10. *A Path Complex Message Passing (PCMP) with sufficient layers and injective neighborhood aggregators achieves the same expressive power as the PWL.*

4 EXPERIMENTS

4.1 BENCHMARK DATASETS AND MODELS

To thoroughly validate our PCMP model, we use three widely recognized benchmark datasets from MoleculeNet Wu et al. (2018) and MolBench Jiang et al. (2023). During data preprocessing, we employ the Merck molecular force field (MMFF94) function from RDKit to generate 3D molecular structures. The datasets are split into training, validation, and test sets using the scaffold splitting method, with a ratio of 8:1:1. Detailed descriptions of the datasets, preprocessing steps, and splitting method are provided in Appendix A.2.

We compare the performance of our PCMP model against state-of-the-art GNN models, both with and without pre-training. The non-pre-trained GNN models include (1) widely-used architectures such as GIN Xu et al. (2018), GAT Velickovic et al. (2017), and GCN Kipf & Welling (2016a); (2) recent models incorporating 3D molecular geometry, including SGCN Danel et al. (2020), DimeNet Gasteiger et al. (2020), and HMGNN Shui & Karypis (2020b); and (3) architectures specifically designed for molecular representation, such as D-MPNN Yang et al. (2019), AttentiveFP Xiong et al. (2019), and Mol-GDL Shen et al. (2023). For pre-trained models, we compare against N-Gram Liu et al. (2019), PretrainGNN Hu et al. (2019), GROVER Rong et al. (2020), GEM Fang et al. (2022), DMP Zhu et al. (2023), and SMPT Li et al. (2024).

Table 1: Comparison with GNN architectures. The best performance is indicated as **bold**, and the subindex indicates standard deviation values. * indicates that the result is not available for the model.

Method		QM7	QM9	Tox21	HIV	MUV
GNN	GIN	$110.3_{(7.2)}$	$0.00886_{(0.00005)}$	$0.740_{(0.008)}$	$0.753_{(0.019)}$	$0.718_{(0.003)}$
	GAT	$103.0_{(4.4)}$	$0.01117_{(0.00018)}$	$0.745_{(0.006)}$	$0.724_{(0.008)}$	$0.671_{(0.011)}$
	GCN	$100.0_{(3.8)}$	$0.00923_{(0.00019)}$	$0.709_{(0.003)}$	$0.740_{(0.003)}$	$0.716_{(0.004)}$
	D-MPNN	$103.5_{(8.6)}$	$0.00812_{(0.00009)}$	$0.759_{(0.007)}$	$0.771_{(0.005)}$	$0.786_{(0.014)}$
	Attentive FP	$72.0_{(2.7)}$	$0.00812_{(0.00001)}$	$0.761_{(0.005)}$	$0.757_{(0.014)}$	$0.766_{(0.015)}$
	GTransformer	$161.3_{(7.1)}$	$0.00923_{(0.00019)}$	*	*	*
	SGCN	$131.3_{(11.6)}$	$0.01459_{(0.00055)}$	*	*	*
	DimNet	$95.6_{(4.1)}$	$0.01031_{(0.00076)}$	*	*	*
	HMGNN	$101.6_{(3.2)}$	$0.01239_{(0.00001)}$	*	*	*
	Mol-GDL	$62.2_{(0.4)}$	$0.00952_{(0.00013)}$	$0.791_{(0.005)}$	$0.808_{(0.007)}$	$0.675_{(0.014)}$
Pretrain_GNN	N-Gram$_{RF}$	$92.8_{(4.0)}$	$0.01037_{(0.00016)}$	$0.743_{(0.004)}$	$0.772_{(0.001)}$	$0.769_{(0.007)}$
	N-Gram$_{XGB}$	$81.9_{(1.9)}$	$0.00964_{(0.00031)}$	$0.758_{(0.009)}$	$0.787_{(0.004)}$	$0.748_{(0.002)}$
	PretrainGNN	$113.2_{(0.6)}$	$0.00922_{(0.00004)}$	$0.781_{(0.006)}$	$0.799_{(0.007)}$	$0.813_{(0.021)}$
	GROVER$_{base}$	$94.5_{(3.8)}$	$0.00986_{(0.00055)}$	$0.743_{(0.001)}$	$0.625_{(0.009)}$	$0.673_{(0.018)}$
	GROVER$_{large}$	$92.0_{(0.9)}$	$0.00986_{(0.00025)}$	$0.735_{(0.001)}$	$0.682_{(0.011)}$	$0.673_{(0.018)}$
	MolCLR	$66.8_{(2.3)}$	*	$0.750_{(0.002)}$	$0.781_{(0.005)}$	$0.796_{(0.019)}$
	GEM	$58.9_{(0.8)}$	$0.00746_{(0.00001)}$	$0.781_{(0.001)}$	$0.806_{(0.009)}$	$0.817_{(0.005)}$
	DMP	$74.4_{(1.2)}$	*	$0.791_{(0.004)}$	$0.814_{(0.004)}$	*
	SMPT	*	*	$0.797_{(0.001)}$	$0.812_{(0.001)}$	$0.822_{(0.008)}$
PCMP		$\mathbf{53.6}_{(2.1)}$	$\mathbf{0.00683}_{(0.00005)}$	$\mathbf{0.801}_{(0.002)}$	$\mathbf{0.823}_{(0.004)}$	$\mathbf{0.827}_{(0.015)}$

4.2 RESULTS

The comparison of our PCMP model with existing models on benchmark datasets is presented in Table 1. Detailed parameter settings for PCMP can be found in Section A.3 (Appendix A). Our PCMP model shows a significant performance advantage across datasets, mainly due to its advanced feature extraction capabilities and superior recognition of complex molecular structures. The message-passing mechanism in PCMP is organized into two distinct layers: the upper embedding, which considers upper adjacent neighbors, and the lower embedding, which incorporates both upper adjacent and face neighbors. This dual-layered approach integrates path information from multiple perspectives, enhancing the model's ability to capture both local and global graph structures. Each path is updated not only based on the features of its constituent nodes but also by incorporating information from both higher-order and lower-order connected paths. This sophisticated mechanism enables the model to detect subtle structural variations within molecules that are often difficult to distinguish. Compared to models that rely heavily on traditional pre-training, PCMP reduces computational demands by bypassing extensive pre-training phases while achieving excellent results, making it a promising solution for molecular property prediction.

4.3 ABLATION STUDY OF PCMP

Impact of Message Passing Mechanisms In the PCMP model, high-order features are first updated and then used as inputs to update low-order features, with the updated low-order features subsequently used to update high-order features. To evaluate the significance of this interaction, we introduced three variant models: PCMP-PARAL, PCMP-HL, and PCMP-LH, each limiting the information exchange between different feature levels. Specifically, PCMP-PARAL restricts both high-to-low and low-to-high feature updates; PCMP-HL limits message passing from high-order to low-order features; and PCMP-LH restricts message passing from low-order to high-order features. Table 2 compares the performance of these models—PCMP-PARAL, PCMP-HL, PCMP-LH—against the standard PCMP on benchmark datasets.

Impact of Input Path The PCMP model integrates various path-based features to improve its ability to accurately capture molecular structures. These paths, ranging from 0-path to 3-path, represent different levels of molecular interaction complexity, from the simplest to the most intricate. Table 3 shows the performance results for different path inputs on the benchmark datasets. As indicated, incorporating higher-order paths (2-path and 3-path) generally enhances the model's performance,

Table 2: Results on benchmark datasets with different message passing mechanisms.

Method	QM7	QM9	Tox21	HIV	MUV
PCMP-PARAL	$56.9_{(1.5)}$	$0.00751_{(0.00015)}$	$0.779_{(0.008)}$	$0.794_{(0.016)}$	$0.808_{(0.006)}$
PCMP-HL	$54.8_{(1.0)}$	$0.00727_{(0.00006)}$	$0.796_{(0.002)}$	$0.803_{(0.020)}$	$0.814_{(0.012)}$
PCMP-LH	$55.3_{(1.9)}$	$0.00764_{(0.00005)}$	$0.793_{(0.004)}$	$0.793_{(0.009)}$	$0.806_{(0.004)}$
PCMP	$\mathbf{53.6}_{(2.1)}$	$\mathbf{0.00683}_{(0.00005)}$	$\mathbf{0.801}_{(0.004)}$	$\mathbf{0.823}_{(0.004)}$	$\mathbf{0.827}_{(0.015)}$

with the inclusion of the 3-path yielding the best results across all datasets. This underscores its effectiveness in capturing complex molecular interactions. However, the slight performance degradation when excluding the 2-path and 3-path elements suggests that lower-order information remains crucial, especially for the QM7 dataset, where simpler molecular representations are sufficient.

Table 3: The results for input different path of the benchmark datasets.

Input-Path	QM7	QM9	Tox21	HIV	MUV
{0,1}-path	$57.0_{(1.4)}$	$0.00898_{(0.00012)}$	$0.786_{(0.006)}$	$0.782_{(0.007)}$	$0.767_{(0.003)}$
{0,1,2}-path	$56.9_{(1.1)}$	$0.00700_{(0.00006)}$	$0.792_{(0.002)}$	$0.803_{(0.005)}$	$0.815_{(0.012)}$
{0,1,2,3}-path	$\mathbf{53.6}_{(2.1)}$	$\mathbf{0.00683}_{(0.00005)}$	$\mathbf{0.801}_{(0.002)}$	$\mathbf{0.823}_{(0.004)}$	$\mathbf{0.827}_{(0.015)}$

Impact of Readout Path To investigate whether feature outputs at different levels can improve model performance, we designed several output strategies. As shown in Table 4, utilizing outputs from multiple levels allows the PCMP model to capture both low-order and high-order molecular features, significantly enhancing the model's ability to represent complex structures and improving its overall expressiveness.

Table 4: The results for Readout different path of the benchmark datasets.

Output-Path	QM7	QM9	Tox21	HIV	MUV
{0}-path	$56.8_{(1.2)}$	$0.00697_{(0.00004)}$	$0.789_{(0.008)}$	$0.784_{(0.010)}$	$0.806_{(0.018)}$
{0,1}-path	$56.4_{(1.5)}$	$0.00698_{(0.00010)}$	$0.793_{(0.009)}$	$0.800_{(0.007)}$	$0.808_{(0.007)}$
{0,1,2}-path	$\mathbf{53.6}_{(2.1)}$	$\mathbf{0.00683}_{(0.00005)}$	$\mathbf{0.801}_{(0.002)}$	$\mathbf{0.823}_{(0.004)}$	$\mathbf{0.827}_{(0.015)}$

We examined the model's sensitivity to various hyperparameters, and the experimental results are presented in Table 8.

5 Conclusion

In this study, we introduced the path complex message passing, a novel model for molecular structure representation based on path complexes, designed to predict molecular properties. By integrating force fields with path complexes, the model enhances our understanding of the relationship between molecular structure and function, offering valuable insights for both theoretical research and practical applications in molecular design and materials science. The PCMP model employs 0-paths for atomic properties, 1-paths for pairwise interactions, 2-paths for bond angle terms, and 3-paths for dihedral angle information. These paths are used to compute attention scores, enabling efficient message propagation and feature integration across various levels of molecular information. Validation on five benchmark datasets has demonstrated the PCMP's superior predictive capabilities. Ablation studies further confirm that incorporating higher-order features significantly improves performance, pointing to promising directions for future research in molecular simulation and design.

References

Kenneth Atz, Francesca Grisoni, and Gisbert Schneider. Geometric deep learning on molecular representations. *Nature Machine Intelligence*, 3(12):1023–1032, 2021.

Song Bai, Feihu Zhang, and Philip HS Torr. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 110:107637, 2021.

Lorenz C Blum and Jean-Louis Reymond. 970 million druglike small molecules for virtual screening in the chemical universe database gdb-13. *Journal of the American Chemical Society*, 131(25): 8732–8733, 2009.

Cristian Bodnar. *Topological Deep Learning: Graphs, Complexes, Sheaves*. PhD thesis, Apollo - University of Cambridge Repository, 2022. URL `https://www.repository.cam.ac.uk/handle/1810/350982`.

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Lio, Guido F Montufar, and Michael Bronstein. Weisfeiler and lehman go cellular: Cw networks. *Advances in neural information processing systems*, 34:2625–2640, 2021a.

Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido F Montufar, Pietro Lio, and Michael Bronstein. Weisfeiler and lehman go topological: Message passing simplicial networks. In *International Conference on Machine Learning*, pp. 1026–1037. PMLR, 2021b.

Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

Zixuan Cang and Guo-Wei Wei. Topologynet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLoS computational biology*, 13 (7):e1005690, 2017.

HC Stephen Chan, Hanbin Shan, Thamani Dahoun, Horst Vogel, and Shuguang Yuan. Advancing drug discovery via artificial intelligence. *Trends in pharmacological sciences*, 40(8):592–604, 2019.

Chi Chen, Weike Ye, Yunxing Zuo, Chen Zheng, and Shyue Ping Ong. Graph networks as a universal machine learning framework for molecules and crystals. *Chemistry of Materials*, 31(9):3564–3572, 2019.

Dong Chen, Jian Liu, Jie Wu, Guo-Wei Wei, Feng Pan, and Shing-Tung Yau. Path topology in molecular and materials sciences. *The journal of physical chemistry letters*, 14(4):954–964, 2023.

Hongming Chen, Ola Engkvist, Yinhai Wang, Marcus Olivecrona, and Thomas Blaschke. The rise of deep learning in drug discovery. *Drug discovery today*, 23(6):1241–1250, 2018.

Jiahui Chen, Yuchi Qiu, Rui Wang, and Guo-Wei Wei. Persistent laplacian projected omicron ba. 4 and ba. 5 to become new dominating variants. *Computers in biology and medicine*, 151:106262, 2022.

Kamal Choudhary and Brian DeCost. Atomistic line graph neural network for improved materials property predictions. *npj Computational Materials*, 7(1):185, 2021.

Kamal Choudhary, Brian DeCost, and Francesca Tavazza. Machine learning with force-field-inspired descriptors for materials: Fast screening and mapping energy landscape. *Physical review materials*, 2(8):083801, 2018.

Samir Chowdhury and Facundo Mémoli. Persistent path homology of directed networks. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1152–1169. SIAM, 2018.

Samir Chowdhury, Steve Huntsman, and Matvey Yutin. Path homologies of motifs and temporal network representations. *Applied Network Science*, 7(1):4, 2022.

Tomasz Danel, Przemysław Spurek, Jacek Tabor, Marek Śmieja, Łukasz Struski, Agnieszka Słowik, and Łukasz Maziarka. Spatial graph convolutional networks. In *International Conference on Neural Information Processing*, pp. 668–675. Springer, 2020.

Xiaomin Fang, Lihang Liu, Jieqiong Lei, Donglong He, Shanzhuo Zhang, Jingbo Zhou, Fan Wang, Hua Wu, and Haifeng Wang. Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence*, 4(2):127–134, 2022.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 3558–3565, 2019.

Daniel Flam-Shepherd, Tony C Wu, Pascal Friederich, and Alan Aspuru-Guzik. Neural message passing on high order paths. *Machine Learning: Science and Technology*, 2(4):045009, 2021.

Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*, 2020.

Johannes Gasteiger, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. *Advances in Neural Information Processing Systems*, 34:6790–6802, 2021.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.

Lorenzo Giusti, Claudio Battiloro, Lucia Testa, Paolo Di Lorenzo, Stefania Sardellitti, and Sergio Barbarossa. Cell attention networks. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2023.

Miguel A González. Force fields and molecular dynamics simulations. *École thématique de la Société Française de la Neutronique*, 12:169–200, 2011.

Alexander Grigor'yan, Yong Lin, Yuri Muranov, and Shing-Tung Yau. Homologies of path complexes and digraphs. *arXiv preprint arXiv:1207.2834*, 2012.

Alexander Grigor'yan, Rolando Jimenez, Yuri Muranov, and Shing-Tung Yau. Homology of path complexes and hypergraphs. *Topology and its Applications*, 267:106877, 2019.

A.A. Grigor'yan, Y. Lin, Y.V. Muranov, et al. Path complexes and their homologies. *Journal of Mathematical Sciences*, 248:564–599, 2020. doi: 10.1007/s10958-020-04897-9. URL `https://doi.org/10.1007/s10958-020-04897-9`.

Alexander Grigor'yan, Yong Lin, Yuri Muranov, and Shing-Tung Yau. Homotopy theory for digraphs. *Pure and Applied Mathematics Quarterly*, 10(4):619–674, 2014.

Alexander Grigor'yan, Rolando Jimenez, Yuri Muranov, and Shing-Tung Yau. On the path homology theory of digraphs and eilenberg–steenrod axioms. *Homology, Homotopy and Applications*, 20(2):179–205, 2018.

Alexander Grigor'yan, Yong Lin, and Shing-Tung Yau. Analytic and reidemeister torsions of digraphs and path complexes. *Pure and Applied Mathematics Quarterly*, 20(2):703–755, 2024.

Mustafa Hajij, Kyle Istvan, and Ghada Zamzmi. Cell complex neural networks. In *TDA {\&} Beyond*, 2020.

Mustafa Hajij, Ghada Zamzmi, Theodore Papamarkou, Nina Miolane, Aldo Guzmán-Sáenz, Karthikeyan Natesan Ramamurthy, Tolga Birdal, Tamal K Dey, Soham Mukherjee, Shreyas N Samaga, et al. Topological deep learning: Going beyond graph data. *arXiv e-prints*, pp. arXiv–2206, 2022.

Thomas A Halgren. Merck molecular force field. i. basis, form, scope, parameterization, and performance of mmff94. *Journal of computational chemistry*, 17(5-6):490–519, 1996.

Jakob Hansen and Robert Ghrist. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology*, 3(4):315–358, 2019.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2019.

Kexin Huang, Tianfan Fu, Lucas Glass, Marinka Zitnik, Cao Xiao, and Jimeng Sun. Deeppurpose: a deep learning library for drug–target interaction prediction. *Bioinformatics*, 36:5545 – 5547, 2020. URL https://api.semanticscholar.org/CorpusID:220496219.

John B Ingraham, Max Baranov, Zak Costello, Karl W Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M Lord, Christopher Ng-Thow-Hing, Erik R Van Vlack, et al. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, 2023.

Arian Jamasb, Ramon Viñas Torné, Eric Ma, Yuanqi Du, Charles Harris, Kexin Huang, Dominic Hall, Pietro Lió, and Tom Blundell. Graphein-a python library for geometric deep learning and network analysis on biomolecular structures and interaction networks. *Advances in Neural Information Processing Systems*, 35:27153–27167, 2022.

Xiuyu Jiang, Liqin Tan, Jianhuan Cen, and Qingsong Zou. Molbench: A benchmark of ai models for molecular property prediction. In *International Symposium on Benchmarking, Measuring and Optimization*, pp. 53–70. Springer, 2023.

Eun-Sol Kim, Woo Young Kang, Kyoung-Woon On, Yu-Jung Heo, and Byoung-Tak Zhang. Hypergraph attention networks for multimodal learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14581–14590, 2020.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016a.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *Advances in neural information processing systems*, 2016b.

Andrew R Leach. *Molecular modelling: principles and applications*. Pearson education, 2001.

Shuangli Li, Jingbo Zhou, Tong Xu, Dejing Dou, and Hui Xiong. GeomGCL: geometric graph contrastive learning for molecular property prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 4541–4549, 2022.

Yishui Li, Wei Wang, Jie Liu, and Chengkun Wu. Pre-training molecular representation model with spatial geometry for property prediction. *Computational Biology and Chemistry*, 109:108023, 2024.

Jian Liu, Dong Chen, Feng Pan, and Jie Wu. Neighborhood path complex for the quantitative analysis of the structure and stability of carboranes. *Journal of Computational Biophysics and Chemistry*, 22(04):503–511, 2023.

Shengchao Liu, Mehmet F Demirel, and Yingyu Liang. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. *Advances in neural information processing systems*, 32, 2019.

Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d molecular graphs. In *International Conference on Learning Representations (ICLR)*, 2022.

Kit-Kay Mak and Mallikarjuna Rao Pichika. Artificial intelligence in drug development: present status and future prospects. *Drug discovery today*, 24(3):773–780, 2019.

Stephen L Mayo, Barry D Olafson, and William A Goddard. Dreiding: a generic force field for molecular simulations. *Journal of Physical chemistry*, 94(26):8897–8909, 1990.

Zhuoran Qiao, Anders S Christensen, Matthew Welborn, Frederick R Manby, Anima Anandkumar, and Thomas F Miller III. Informing geometric deep learning with electronic interactions to accelerate quantum chemistry. *Proceedings of the National Academy of Sciences*, 119(31): e2205221119, 2022.

Bharath Ramsundar, Peter Eastman, Pat Walters, and Vijay Pande. *Deep learning for the life sciences: applying deep learning to genomics, microscopy, drug discovery, and more*. " O'Reilly Media, Inc.", 2019.

Patrick Reiser, Marlen Neubert, André Eberhard, Luca Torresi, Chen Zhou, Chen Shao, Houssam Metni, Clint van Hoesel, Henrik Schopmans, Timo Sommer, et al. Graph neural networks for materials science and chemistry. *Communications Materials*, 3(1):93, 2022.

T Mitchell Roddenberry, Nicholas Glaze, and Santiago Segarra. Principled simplicial neural networks for trajectory prediction. In *International Conference on Machine Learning*, pp. 9020–9029. PMLR, 2021.

T Mitchell Roddenberry, Michael T Schaub, and Mustafa Hajij. Signal processing on cell complexes. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8852–8856. IEEE, 2022.

Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33:12559–12571, 2020.

Lars Ruddigkeit, Ruud Van Deursen, Lorenz C Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of chemical information and modeling*, 52(11):2864–2875, 2012.

Michael T Schaub, Jean-Baptiste Seby, Florian Frantzen, T Mitchell Roddenberry, Yu Zhu, and Santiago Segarra. Signal processing on simplicial complexes. In *Higher-Order Systems*, pp. 301–328. Springer, 2022.

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24), 2018.

Cong Shen, Jiawei Luo, and Kelin Xia. Molecular geometric deep learning. *Cell Reports Methods*, 3(11), 2023.

Hiroyuki Shindo and Yuji Matsumoto. Gated graph recursive neural networks for molecular property prediction. *ArXiv*, abs/1909.00259, 2019. URL `https://api.semanticscholar.org/CorpusID:202541698`.

Zeren Shui and George Karypis. Heterogeneous molecular graph neural networks for predicting molecule properties. *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 492–500, 2020a. URL `https://api.semanticscholar.org/CorpusID:221971188`.

Zeren Shui and George Karypis. Heterogeneous molecular graph neural networks for predicting molecule properties. In *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 492–500. IEEE, 2020b.

Freyr Sverrisson, Jean Feydy, Bruno E Correia, and Michael M Bronstein. Fast end-to-end learning on protein surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15272–15281, 2021.

Jacob Townsend, Cassie Putman Micucci, John H Hymel, Vasileios Maroulas, and Konstantinos D Vogiatzis. Representation of molecular structures with persistent homology for machine learning applications in chemistry. *Nature communications*, 11(1):3230, 2020.

Raphael JL Townshend, Stephan Eismann, Andrew M Watkins, Ramya Rangan, Masha Karelina, Rhiju Das, and Ron O Dror. Geometric deep learning of rna structure. *Science*, 373(6558): 1047–1051, 2021.

Quang Truong and Peter Chin. Weisfeiler and lehman go paths: Learning topological features via path complexes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 15382–15391, 2024.

Oliver T Unke and Markus Meuwly. Physnet: A neural network for predicting energies, forces, dipole moments, and partial charges. *Journal of chemical theory and computation*, 15(6):3678–3693, 2019.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. Graph attention networks. *stat*, 1050(20):10–48550, 2017.

Nikolaos N Vlassis, Ran Ma, and WaiChing Sun. Geometric deep learning for computational mechanics part i: Anisotropic hyperelasticity. *Computer Methods in Applied Mechanics and Engineering*, 371:113299, 2020.

Limei Wang, Yi Liu, Yuchao Lin, Haoran Liu, and Shuiwang Ji. Comenet: Towards complete and efficient message passing for 3d molecular graphs. *Advances in Neural Information Processing Systems*, 35:650–664, 2022a.

Rui Wang and Guo-Wei Wei. Persistent path laplacian. *Foundations of data science (Springfield, Mo.)*, 5(1):26–55, 2023.

Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3):279–287, 2022b.

Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*, 2016.

Oliver Wieder, Stefan Kohlbacher, Mélaine Kuenemann, Arthur Garon, Pierre Ducrot, Thomas Seidel, and Thierry Langer. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies*, 37:1–12, 2020.

Shuang Wu, Xiang Liu, Ang Dong, Claudia Gragnoli, Christopher Griffin, Jie Wu, Shing-Tung Yau, and Rongling Wu. The metabolomic physics of complex diseases. *Proceedings of the National Academy of Sciences*, 120(42):e2308496120, 2023.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

Kelin Xia and Guo-Wei Wei. Persistent homology analysis of protein structure, flexibility, and folding. *International journal for numerical methods in biomedical engineering*, 30(8):814–844, 2014.

Tian Xie and Jeffrey C Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical review letters*, 120(14):145301, 2018.

Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of medicinal chemistry*, 63(16):8749–8760, 2019.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.

Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. GraphRNN: Generating realistic graphs with deep auto-regressive models. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5708–5717. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/you18a.html`.

Zhaoning Yu and Hongyang Gao. Molecular graph representation learning via heterogeneous motif graph construction. *arXiv preprint arXiv:2202.00529*, 2022.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019.

Lu Zhang, Jianjun Tan, Dan Han, and Hao Zhu. From machine learning to deep learning: progress in machine intelligence for rational drug discovery. *Drug discovery today*, 22(11):1680–1685, 2017.

Jinhua Zhu, Yingce Xia, Lijun Wu, Shufang Xie, Wengang Zhou, Tao Qin, Houqiang Li, and Tie-Yan Liu. Dual-view molecular pre-training. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3615–3627, 2023.

A APPENDIX / SUPPLEMENTAL MATERIAL

A.1 INITIALIZATION FEATURES

To fully incorporate MD force field information into molecular representation, we propose molecular path complex, which uses path simplices at different dimensions to explicitly characterize force field (covalent) bond terms. More specifically, our 1-path simplex represents bond length information, 2-path simplex describe bond angles, and 3-path simplex characterizes dihedral angle.

Table 5: MD Encoder for Path Features

Features Type		Description	Type	Size
1-Path (bond)	Bond Directionality	None, Beginwedge, Begindash, etc.	One-Hot	7
	Bond Type	Single, Double, Triple, or Aromatic.	One-Hot	4
	Bond Length	Numerical length of the bond.	Float	1
	In Ring	Indicates if the bond is part of a chemical ring.	One-Hot	2
1-Path (non-bond) cutoff=3	Atom charges	Atoms charges in Molecular $(q_i, q_j, q_i \cdot q_j)$	Float	3
	Distance between atoms	Distance between atoms $(1/d_{ij}, 1/d_{ij}^6, 1/d_{ij}^{12})$	Float	3
2-Path	Centroid	Centroid position of the triangle formed by 2-path	Float	3
	Distance	Three bond lengths (two for covalent bond and one for non-covalent bond)	Float	3
	Area	Triangle area spanned by 2-path	Float	1
	Bond Angle	Bond angle for 2-path	Float	1
3-Path	Volume	Volume spanned by 3-path	Float	1
	Dihedral	Dihedral angle for 3-path	Float	1
	Total Area	Total Area of the corresponding four triangles	Float	1
	Bond Length	Non-covalent bond length ($\{v_1v_3\}$,$\{v_2v_4\}$, $\{v_1v_4\}$)	Float	3

A.2 DATASET DETAILS, MIN-MAX SCALING, SPLITTING METHO AND MEAN ABSOLUTE ERR

In this study, we analyzed five datasets from MoleculeNet Wu et al. (2018) and MolBench Jiang et al. (2023): QM7 Blum & Reymond (2009), QM9 Ruddigkeit et al. (2012), Tox21, Hiv and Muv, all of which are publicly available on the MoleculeNet website: https://moleculenet.org/datasets-1. Details about these datasets are in Table 6. Note that the subindex indicates standard deviation

Table 6: The details of the datasets. Note that the subindex indicates standard deviation values.

Dataset	QM7	QM9	Tox21	HIV	MUV
No. molecules	6,830	133,885	7831	41127	93808
No. average atoms	$16_{(3)}$	$18_{(3)}$	$36_{(23)}$	$46_{(24)}$	$43_{(10)}$
No. tasks	1	3	12	1	17
Task type	Regression	Regression	Classification	Classification	Classification
Evaluation	MAE	MAE	ROC-AUC	ROC-AUC	ROC-AUC

values. For instance, the element $16_{(13)}$ means the number of average atoms in QM7 is 16, with 13 as its standard deviation. The QM7 dataset is a subset of the GDB-13 database Blum & Reymond (2009), which contains approximately 1 billion organic molecules with up to seven "heavy"

atoms (C, N, O, S). The QM9 dataset, a subset of the GDB-17 database, provides twelve properties, encompassing geometric, energetic, electronic, and thermodynamic properties. Tox21 is qualitative toxicity measurements on 12 biological targets, including nuclear receptors and stress response pathways. HIV is experimentally measured abilities to inhibit HIV replication. MUV is subset of PubChem BioAssay by applying a refined nearest neighbor analysis, designed for validation of virtual screening techniques.

Min-Max Scaling Given that QM7 and QM9 involve regression, we applied min-max normalization to scale target values between 0 and 1. In multiple-target regression tasks, Min-Max Scaling is commonly used to normalize the targets. This technique linearly transforms the target values to a specified range between a minimum and maximum value. The transformation follows the formula:

$$\overline{y} = \frac{y - y_{\text{min}}}{y_{\text{max}} - y_{\text{min}}}, \quad y_{\text{scal}} = y_{\text{max}} - y_{\text{min}} \tag{1}$$

Here, $\overline{y}$ represents the normalized target value, y is the original target value, y_{min} is the minimum value of the target, and y_{max} is the maximum value of the target.

During prediction, the normalized predictions obtained from the model need to be transformed back to the original scale of the target values. The transformation is performed using the formula:

$$\tilde{y} = \hat{y} \cdot y_{\text{scal}} + y_{\text{min}}, \quad y = \overline{y} \cdot y_{\text{scal}} + y_{\text{min}} \tag{2}$$

where $\hat{y}$ is the model output, and $\tilde{y}$ and y are used for loss function computation and evaluation.

This normalization process ensures that all target values are scaled within a fixed range, typically between 0 and 1. It facilitates better convergence during model training and helps in handling targets with varying scales effectively. Furthermore, Min-Max Scaling maintains the relative relationships between target values while bringing them into the desired range, making it a suitable choice for multiple-target regression tasks.

Splitting Method Following the work of Bharath Ramsundar Ramsundar et al. (2019), we employed scaffold splitting to partition all datasets. This method segments molecules based on their scaffolds (molecular substructures). Scaffold splitting is a more challenging partitioning approach that can better evaluate a model's generalization ability on out-of-distribution data samples. To ensure a fair comparison with other models, we adopted the same scaffold splitting method to divide the task datasets into training, validation, and test sets with a ratio of 8:1:1.

MAE (Mean Absolute Error) The Mean Absolute Error (MAE) is defined as:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{3}$$

where y_i and $\tilde{y}_i$ represent the true value and predicted value of the ith sample respectively. MAE is a commonly used metric for evaluating regression performance. A lower MAE value indicates higher prediction accuracy, with a decrease in MAE typically suggesting improved model performance.

Table 7: Hyperparameters set up.

Dataset	QM7	QM9	Tox21	HIV	MUV
Learning rate	1e-4	1e-3	1.5e-4	1e-3	1e-4
Batch size	512	64	512	512	512
No.heads	1	6	6	2	1
No.layers	2	2	2	2	2
Train/Valid/Test	8:1:1	8:1:1	8:1:1	8:1:1	8:1:1
Loss function	L1	L1	BCE	BCE	BCE
Optimizer	ADAM	ADAM	ADAM	ADAM	ADAM
Epochs	500	500	1000	1000	1000
Seed	42	42	42	42	42

A.3 HYPERPARAMETERS SETUP

Hyperparameters We have set up a set of hyperparameters for training the model are summarized in Table 7. Inaddition, the optimizer selected as ADAM, and the loss function chosen as L1. All models are trained using NVIDIA RTX A5000 32GB GPUs.

Sensitivity of Hyperparameters We explored the model's sensitivity to hyperparameters and the experimental results are displayed in Table 8. According to the results, the model's performance metrics are generally stable across different hyperparameters settings.

Table 8: Sensitivity of hyperparameters for benchmark datasets.

Hyperparameters		QM7	QM9	Tox21	HIV	MUV
Head	1	**53.6**$_{(2.1)}$	0.00931$_{(0.00007)}$	0.748$_{(0.006)}$	0.793$_{(0.004)}$	**0.827**$_{(0.015)}$
	2	54.7$_{(1.8)}$	0.00747$_{(0.00006)}$	0.754$_{(0.007)}$	**0.823**$_{(0.004)}$	0.787$_{(0.019)}$
	4	55.9$_{(1.0)}$	0.00721$_{(0.00006)}$	0.759$_{(0.005)}$	0.799$_{(0.013)}$	0.801$_{(0.017)}$
	6	58.3$_{(2.5)}$	**0.00683**$_{(0.00005)}$	**0.801**$_{(0.004)}$	0.807$_{(0.016)}$	0.814$_{(0.013)}$
	8	58.5$_{(1.2)}$	0.00826$_{(0.00008)}$	0.765$_{(0.004)}$	0.808$_{(0.004)}$	0.783$_{(0.008)}$
Batch Size	64	59.6$_{(1.7)}$	0.00868$_{(0.00012)}$	0.784$_{(0.004)}$	0.796$_{(0.003)}$	0.785$_{(0.007)}$
	128	57.5$_{(1.0)}$	0.01040$_{(0.00007)}$	0.778$_{(0.003)}$	0.803$_{(0.007)}$	0.819$_{(0.016)}$
	256	54.3$_{(0.9)}$	0.00756$_{(0.00014)}$	0.778$_{(0.003)}$	0.807$_{(0.015)}$	0.789$_{(0.022)}$
	512	**53.6**$_{(2.1)}$	**0.00683**$_{(0.00004)}$	**0.801**$_{(0.004)}$	**0.823**$_{(0.004)}$	**0.827**$_{(0.015)}$
LR	5e-3	58.5$_{(1.2)}$	0.00784$_{(0.00004)}$	0.782$_{(0.004)}$	0.805$_{(0.009)}$	0.808$_{(0.014)}$
	1e-3	56.2$_{(1.4)}$	**0.00683**$_{(0.00005)}$	**0.791**$_{(0.009)}$	0.811$_{(0.013)}$	**0.827**$_{(0.015)}$
	5e-4	54.7$_{(1.3)}$	0.00880$_{(0.00006)}$	0.781$_{(0.012)}$	**0.823**$_{(0.005)}$	0.814$_{(0.012)}$
	1e-4	**53.6**$_{(2.1)}$	0.00962$_{(0.00012)}$	0.789$_{(0.006)}$	0.799$_{(0.018)}$	0.822$_{(0.013)}$
	5e-5	64.9$_{(3.1)}$	0.00784$_{(0.00011)}$	0.724$_{(0.007)}$	0.788$_{(0.024)}$	0.818$_{(0.019)}$

B MATHEMATICAL ANALYSIS OF PATH COMPLEX

B.1 PATH COMPLEX

Definition B.1 (Elementary path Grigor'yan et al. (2012))**.** Given a set V, an elementary n-path of V is any sequence of $n+1$ elements $v_0v_1\cdots v_n$ of V, denoted by $\sigma_n = v_0v_1\cdots v_n$

Definition B.2 (Path complex Grigor'yan et al. (2012))**.** A path complex P over the vertex set V is a collection of elementary paths of V such that $\forall\sigma_n = v_0v_1\cdots v_n \in P$, $v_1\cdots v_n \in P, v_0v_1\cdots v_{n-1} \in P$.

The element σ_n of P that has $n+1$ vertices is called an n-path of P. The path σ is called a face of the path τ if σ is derived from τ by removing the first or last vertex. The n-path τ_n is called a coface of $(n-1)$-path σ_{n-1} if σ_{n-1} is a face of τ_n. Two n-paths are upper adjacent if they are faces of a common $(n+1)$-path, lower adjacent if they have a common $(n-1)$-path as face. For an n-path σ_n, let $\mathcal{B}(\sigma_n)$ be the set of faces of σ_n, $\mathcal{C}(\sigma_n)$ be the set of cofaces of σ_n, $\mathcal{N}_\uparrow(\sigma_n)$ be the set of n-paths that are upper adjacent with σ_n, $\mathcal{N}_\downarrow(\sigma_n)$ be the set of n-paths that are lower adjacent with σ_n. Note that we can use the above four relations, including face-relation, coface-relation, upper adjacency and lower adjacency, to define the neighbors of an n-path σ_n. We give construction of path complex from graphs, simplicial complexes and hypergraphs. We give construction of path complex from graphs, simplicial complexes and hypergraphs.

B.1.1 PATH COMPLEX FROM GRAPHS

Definition B.3 (Path)**.** Given an undirected graph $G = (V, E)$ over the verset set V, we define the n-path σ_n of G as any sequence of $n+1$ vertices $v_0v_1\cdots v_n (v_i \in V)$ satisfying the following conditions:

1. $\forall i(0 \leqslant i < n)$, $(v_i, v_{i+1}) \in E$ or $(v_{i+1}, v_i) \in E$.

2. $\forall i \neq j, v_i \neq v_j$.

Note that for each n-path $\sigma_n = v_0 v_1 \cdots v_n$, $\sigma'_n = v_n \cdots v_1 v_0$ is also an n-path, we identify these two paths as the same one.

Definition B.4 (Path complex from graphs)**.** Given an undirected graph $G = (V, E)$, let P_n be the set of all n-paths of G, then $P_G = \bigcup_n P_n$ form a path complex. We call P_G the path complex derived from G.

It can be seen that the path complex P_G derived from G is determined by P_0 and P_1.

B.1.2 Path complex from simplicial complex

Definition B.5 (Simplicial complex)**.** A simplicial complex $\mathcal{K}$ over the vertex set V is a collection of vertex subsets of V satisfying that if $\sigma \in \mathcal{K}, \tau \subset \sigma, \tau \in \mathcal{K}$.

The element σ_k of $\mathcal{K}$ that has $k+1$ vertices is called an k-simplex. A simplex σ is called a face of the simplex τ and τ is called a coface of σ if $\sigma \subset \tau$. For any two k-simplices $\sigma_k, \tau_k \in \mathcal{K}$, σ_k and τ_k are called upper adjacent if they are both faces of an $(k+1)$-simplex $\alpha_{k+1} \in \mathcal{K}$. Two k-simplices σ_k, τ_k are called lower adjacent if they share a common $(k-1)$-simplex as faces.

Definition B.6 (Path)**.** Given a simplicial complex $\mathcal{K}$, we define an (k, n)-path e_n^k of $\mathcal{K}$ as a sequence of $n+1$ k-simplices $\sigma_k^0 \sigma_k^1 \cdots \sigma_k^n$ satisfying the following conditions:

1. $\forall i \ (0 \leqslant i < n)$, σ_k^i and σ_k^{i+1} are upper adjacent.
2. $\forall i \neq j, \sigma_k^i \neq \sigma_k^j$

We can also use lower adjacent, face and the coface relation to define paths. Note that for each (k, n)-path $\sigma_k^0 \sigma_k^1 \cdots \sigma_k^n$, there is an (k, n)-path $\sigma_k^n \cdots \sigma_k^0$. We identify these two paths as the same one.

Definition B.7 (Path complex from simplicial complex)**.** Given a simplicial complex $\mathcal{K}$, let P_n^k be the set of all (k, n)-paths of $\mathcal{K}$, then $P_{\mathcal{K}}^k = \bigcup_n P_n^k$ form a path complex.

For the simplicial complex $\mathcal{K}$, its one-skeleton forms a graph $\mathcal{K}_1$, we have $P_{\mathcal{K}_1} = P_{\mathcal{K}}^0$.

B.1.3 Path complex from hypergraphs

Definition B.8 (Hypergraph)**.** A hypergraph $\mathcal{H}$ over the vertex set V is a collection of vertex subsets of V.

The element σ_k of $\mathcal{H}$ that has $k+1$ vertices is called an k-hyperedge. Two hyperedges are called lower adjacent if their intersection is not empty.

Definition B.9 (Path)**.** Given a hypergraph $\mathcal{H}$ over the vertex set V, we define an n-path of $\mathcal{H}$ as a sequence of $n+1$ hyperedges $\sigma^0 \sigma^1 \cdots \sigma^n$ such that any two adjacent hyperedges are lower adjacent and any two hyperedges are not same.

Definition B.10 (Path complex from hypergraphs)**.** Given a hypergraph $\mathcal{H}$, let P_n be the set of all n-paths of $\mathcal{H}$, then $P_{\mathcal{H}} = \bigcup_n P_n$ form a path complex.

B.2 Homology of Path Complex

The homology of path complex is a new homology theory that breaks the landscape of classical homology theory in algebraic topology, introducing a new framework for exploring the topology of more general mathematical structures Grigor'yan et al. (2012); Grigor'yan et al. (2014; 2020). This homology theory was initially called path homology and renamed GLMY homology in 2022, which advances the study of topological foundations for complex networks Chowdhury & Mémoli (2018); Chowdhury et al. (2022) and has been successfully applied in complex disease Wu et al. (2023), biology and material sciences Chen et al. (2023). Next, we give the construction of homology of path complexes.

Given a path complex P over V, We fix a field coefficient $\mathbb{F}$, let $\Lambda_n(P)$ be the vector space spanned by all the elementary n-paths of P. Considering the standard boundary operator $\partial_n : \Lambda_n(P) \to \Lambda_{n-1}(P)$

$$\forall \sigma_n = v_0 v_1 \cdots v_n \in P, \qquad \partial_n(\sigma_n) = \sum_{i=0}^{n} (-1)^i v_0 \cdots v_{i-1} v_{i+1} \cdots v_n$$

We have $\partial_n \partial_{n+1} = 0$. Let $\mathcal{A}_n(P)$ be the vector space spanned by all the n-paths of P, usually $\partial(\mathcal{A}_n) \not\subset \mathcal{A}_{n-1}(P)$. We consider the following subspace $\Omega_n(P)$ of $\mathcal{A}_n(P)$

$$\Omega_n(P) = \{u \in \mathcal{A}_n(P) | \partial(u) \in \mathcal{A}_{n-1}(P)\}$$

Then we have $\partial_n(\Omega_n(P)) \subset \Omega_{n-1}(P)$. Consequently, we get a chain complex $(\Omega_*(P), \partial_*)$

$$\cdots \to \Omega_{n+1}(P) \xrightarrow{\partial_{n+1}} \Omega_n(P) \xrightarrow{\partial_n} \Omega_{n-1}(P) \to \cdots$$

Definition B.11 (Homology of path complex)**.** Given a path complex P, its k-homology is defined as the k-th homology of the chain complex $(\Omega_*(P), \partial_*)$

$$\mathrm{H}_k(P) = \mathrm{H}_k((\Omega_*, \partial_*))$$

This definition can be directly applied to the path complexes derived from graphs, simplicial complexes and hypergraphs.

B.3 Weak Isomorphism Invariance of the Path Complex Homology

For an undirected graph $G = (V, E)$, the degree of a vertex $v \in V$ is the number of edges that contain v and we denoted it by $deg(v)$.

Definition B.12 (Graph collapse and expansion)**.** Given a graph $G = (V, E)$, take an edge $(v_1, v_2) \in E$ such that $deg(v_1) = 1$. Let $V' = V \backslash \{v_1\}$, $E' = E \backslash \{(v_1, v_2)\}$, then $G' = (V', E')$ is a new graph. We say that G' is derived from G by a graph collapse and G is derived from G' by a graph expansion.

Definition B.13 (Weak isomorphic)**.** Given two graphs G_1, G_2, G_1 and G_2 are called weak isomorphic if G_1 can be derive from G_2 by a sequence of graph collapse and expansion operations.

It is obvious that two graphs are weak isomorphic if they are isomorphic.

Theorem B.14. *If two graphs G_1 and G_2 are weak isomorphic, then*

1. *The number of connected components of G_1 and G_2 are same.*
2. *The number of cycles of G_1 and G_2 are same.*

Proof. Let $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$, without loss of generality, we can assume that G_2 is derived by collapsing an edge $(v_1, v_2) \in E_1$ from G_1 and $deg(v_1) = 1$.

1. This is obvious.
2. Let $C(G_i)$ be the set of cycles of G_i, then we have $C(G_2) \subset C(G_1)$ because G_2 is a subgraph of G_1. $\forall c \in C(G_1)$, c is a sequence of vertices such that the degree of each vertex is 2. So v_1 is not contained in c, $c \in C(G_2)$. Consequently,

$$C(G_1) = C(G_2)$$

□

Theorem B.15. *Given two graphs G_1, G_2, let P_{G_1}, P_{G_2} be the path complexs derived from G_1 and G_2 respectively. If G_1 and G_2 are weak isomorphic, then*

$$\mathrm{H}_k(P_{G_1}) \cong \mathrm{H}_k(P_{G_2}) \ (k \geqslant 0)$$

Proof. Let $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$, without loss of generality, we can assume that G_2 is derived by collapsing an edge $(v_1, v_2) \in E_1$ from G_1 and $deg(v_1) = 1$.

1. $k = 0$

$$\Omega_0(P_{G_1}) =< v_1 > \oplus \Omega_0(P_{G_2}), \quad \Omega_0(P_{G_2}) =< v| \ v \in V_2 >$$
$$\Omega_1(P_{G_1}) =< v_1 v_2 > \oplus \Omega_1(P_{G_2}), \quad \Omega_1(P_{G_2}) =< e| \ e \in E_2 >$$

We have $Ker\partial|_{\Omega_0(P_{G_1})} =< v_1 > \oplus Ker\partial|_{\Omega_0(P_{G_2})}$, $Im\partial|_{\Omega_1(P_{G_1})} =< v_2 - v_1 > \oplus Im\partial|_{\Omega_1(P_{G_2})} = < v_1 > \oplus Im\partial|_{\Omega_1(P_{G_2})}$. Consequently,

$$\begin{aligned} \mathrm{H}_0(P_{G_1}) &= \frac{Ker\partial|_{\Omega_0(P_{G_1})}}{Im\partial|_{\Omega_1(P_{G_1})}} \\ &= \frac{< v_1 > \oplus Ker\partial|_{\Omega_0(P_{G_2})}}{< v_1 > \oplus Im\partial|_{\Omega_1(P_{G_2})}} \\ &= \frac{Ker\partial|_{\Omega_0(P_{G_2})}}{Im\partial|_{\Omega_1(P_{G_2})}} \\ &= \mathrm{H}_0(P_{G_2}) \end{aligned}$$

2. $k = 1$

$$\Omega_1(P_{G_1}) =< v_1 v_2 > \oplus \Omega_1(P_{G_2}), \quad \Omega_1(P_{G_2}) =< e| \ e \in E_2 >$$

The degree of vertex v_1 is one means v_1 only appears in the 1-path (v_1, v_2), so (v_1, v_2) cannot be contained in the kernel of ∂ on $\Omega_1(P_{G_1})$, which means that

$$Ker\partial|_{\Omega_1(P_{G_1})} = Ker\partial|_{\Omega_1(P_{G_2})}$$
$$\mathcal{A}_2(P_{G_1}) =< v_1 v_2 v | v_2 \neq v \in V_2 > \oplus \mathcal{A}_2(P_{G_2})$$

Note that (v_1, v) is not a 1-path for any $v \in V_2 (v \neq v_2)$, so

$$\Omega_2(P_{G_1}) = \Omega_2(P_{G_2})$$

Consequently,

$$\begin{aligned} \mathrm{H}_1(P_{G_1}) &= \frac{Ker\partial|_{\Omega_1(P_{G_1})}}{Im\partial|_{\Omega_2(P_{G_1})}} \\ &= \frac{Ker\partial|_{\Omega_1(P_{G_2})}}{Im\partial|_{\Omega_2(P_{G_2})}} \\ &= \mathrm{H}_1(P_{G_2}) \end{aligned}$$

3. $k \geqslant 2$. It suffices to prove that

$$\Omega_k(P_{G_1}) = \Omega_k(P_{G_2}) \ (k \geqslant 2)$$

It is obvious that $\Omega_k(P_{G_2}) \subset \Omega_k(P_{G_1})$. So we only need to prove that $\Omega_k(P_{G_1}) \subset \Omega_k(P_{G_2})$.

   (a) We prove that

$$\Omega_k(P_{G_1}) \subset \mathcal{A}_k(P_{G_2})$$

   $\forall \omega_k \in \Omega_k(P_{G_1})$, $\omega_k \in \mathcal{A}_k(P_{G_1})$, $\partial(\omega_k) \in \mathcal{A}_{k-1}(P_{G_1})$. Note that every k-path in P_{G_1} is either an k-path in P_{G_2} or starts with $v_1 v_2$, so ω_k can be represented as

$$\omega_k = v_1 v_2 e_{k-2} + e_k$$

   where $e_{k-2} \in \mathcal{A}_{k-2}(P_{G_2})$ is a linear combination of $(k-2)$-paths $\sum v_{i_0} v_{i_1} \cdots v_{i_{k-2}}$ $(v_{i_0} \neq v_1)$ of P_{G_2} and e_k is an k-path of P_{G_2}. We have

$$\partial(\omega_k) = (v_2 - v_1) e_{k-2} + v_1 v_2 \partial(e_{k-2}) + \partial(e_k)$$

   Note that $v_1 e_{k-2}$ is a linear combination of $(k-1)$-paths $v_1 v_{i_0} \cdots v_{i_{k-2}}$ $(v_1 \neq v_{i_0})$. Since $v_1 v_{i_0}$ is not an edge, these paths are not contained in P_{G_1}, but $\partial(w_k) \in \mathcal{A}_{k-1}(P_{G_1})$, so $v_1 e_{k-2}$ must add some item in the right part to become zero. There is not any item in the right part of the equation has $v_1 v_{i_0} \cdots v_{i_{k-2}}$, so $v_1 e_{k-2}$ must be zero, which means that e_{k-2} is zero. Consequently,

$$\omega_k = e_k \in \mathcal{A}_k(P_{G_2})$$

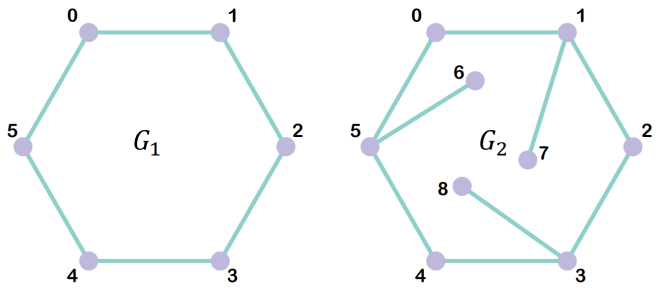


Figure 6: Illustration of the graph weak isomorphism. G_1 can be derived by doing the graph collapse operations on G_2 through $\{7,(1,7)\}$, $\{6,(5,6)\}$ and $\{8,(3,8)\}$ one by one, so G_1 and G_2 are weak isomorphic.

(b) We prove that

$$\Omega_k(P_{G_2}) = \mathcal{A}_k(P_{G_2}) \cap \Omega_k(P_{G_1})$$

It is obvious that $\Omega_k(P_{G_2}) \subset \mathcal{A}_k(P_{G_2}) \cap \Omega_k(P_{G_1})$, so we only need to prove that $\mathcal{A}_k(P_{G_2}) \cap \Omega_k(P_{G_1}) \subset \Omega_k(P_{G_2})$.
$\forall e \in \mathcal{A}_k(P_{G_2}) \cap \Omega_k(P_{G_1})$, Since $e \in \mathcal{A}_k(P_{G_2})$, v_1 will not appear in e, which means e is a path of P_{G_2}. Note that $e \in \Omega_k(P_{G_1})$, so $\partial(e) \in \mathcal{A}_{k-1}(P_{G_1})$, with the property that $e \in P_{G_2}$, we have $\partial(e) \in \mathcal{A}_{k-1}(P_{G_2})$, which means

$$e \in \Omega_k(P_{G_2})$$

Combining the results of (a) and (b), we have

$$\Omega_k(P_{G_1}) \subset \mathcal{A}_k(P_{G_2}) \cap \Omega_k(P_{G_1}) = \Omega_k(P_{G_2})$$

□

Theorem B.15 means the path complex homology is a graph weak isomorphism invariant. Consequently, for two graphs G_1 and G_2, if there exists k such that $\mathrm{H}_k(P_{G_1}) \ncong \mathrm{H}_k(P_{G_2})$, then G_1 and G_2 are not weak isomorphic and not isomorphic.

Figure 6 illustrates an example of the graph weak isomorphism. As shown in Figure 1, $G_1 = (V_1, E_1)$ where $V_1 = \{0,1,2,3,4,5\}$, $E_1 = \{(0,1),(1,2),(2,3),(3,4),(4,5),(0,5)\}$. $G_2 = (V_2, E_2)$ where $V_2 = \{0,1,2,3,4,5,6,7,8\}$ and $E_2 = \{(0,1),(1,2),(2,3),(3,4),(4,5),(0,5),(5,6),(1,7),(3,8)\}$. G_1 can be derived by doing the graph collapse operation on G_2 through $\{7,(1,7)\}$, $\{6,(5,6)\}$ and $\{8,(3,8)\}$ one by one, so G_1 and G_2 are weak isomorphic. It can be seen that G_1 and G_2 both have one cycle and one connected component.

C Path Weisfeiler Lehman (PWL) Test

C.1 Path Complex

Definition C.1 (Path Complex Isomorphism)**.** Given two path complexes P_1, P_2 over the vertices V_1, V_2. P_1 and P_2 are called isomorphic if there is a map $f: V_1 \to V_2$ such that $\sigma_n = v_0 v_1 \cdots v_n \in P_1 \iff f(\sigma) = f(v_0)f(v_1)\cdots f(v_n) \in P_2$.

Theorem C.2. *Given two graphs G_1, G_2, let P_{G_1}, P_{G_2} be the path complexes derived from G_1, G_2 respectively. We have*

$$G_1 \cong G_2 \iff P_{G_1} \cong P_{G_2}$$

C.2 Path Complex Coloring

Definition C.3 (Path Coloring)**.** A path coloring is a map c such that for each path complex P and any path σ of P, $c(\sigma)$ is a color from a fixed color table. We denote this color by c_σ^P.

We will often omit P in the subscript when the underlying path complex is arbitrary.

Definition C.4. Given two path complexes P_1, P_2 and a path coloring c. P_1 and P_2 are called c-similar, denoted by $c^{P_1} = c^{P_2}$, if for any dimension n, we have the color multi-sets equality

$$\{\{c_\sigma^{P_1} | dim(\sigma) = n, \sigma \in P_1\}\} = \{\{c_\tau^{P_2} | dim(\tau) = n, \tau \in P_2\}\}$$

Definition C.5 (PWL)**.** We give a path complex version of the WL test to derive a message passing procedure that can retain the expressive power of the test. We call this the Path WL (PWL), the steps of general PWL are as follows:

1. Given a path complex P, all the paths of P are initialized with the same color.
2. For the color c_σ^t of path σ at iteration t, the color c_σ^{t+1} of σ at the next iteration is computed by perfectly hashing the color multi-set of the neighbors of σ.
3. The algorithm stops once a stable coloring is reached. Two path complexes are considered non-isomorphic if their color histograms are different at some dimensions.

Neighbor Color Multi-set Based on the four neighbor definitions, we have four types of neighbor color multi-sets. Let c^t be the coloring of PWL for path complex P at iteration t, four types of color multi-sets are as follows

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

Having the neighbor color multi-sets, we obtain the following update rule that contains all four types of neighbors:

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

Actually, certain neighbors can be removed without affecting the expressive power of PWL test in terms of path complex that can be differentiated.

Theorem C.6. *PWL with* $\mathrm{HASH}\{c_\sigma^t, c_{\mathcal{B}}^t(\sigma), c_{\uparrow}^t(\sigma)\}$ *is as powerful as PWL with the four-neighbor-updating strategy* $\mathrm{HASH}\{c_\sigma^t, c_{\mathcal{B}}^t(\sigma), c_{\mathcal{C}}^t(\sigma), c_{\uparrow}^t(\sigma), c_{\downarrow}^t(\sigma)\}$.

Theorem C.7. *PWL is strictly more powerful than WL.*

Theorem C.8. *PWL is no less powerful than SWL Bodnar et al. (2021b) with the clique complex lifting.*

C.3 Path Complex Message Passing

We propose a general Path Complex Message Passing (PCMP) using the following messages passing operations. For a path σ in P, we have

$$m_{\mathcal{B}}^{t+1}(\sigma) = AGG_{\tau\in\mathcal{B}(\sigma)}(M_{\mathcal{B}}(h_\sigma^t, h_\tau^t)) \tag{4}$$

$$m_{\uparrow}^{t+1}(\sigma) = AGG_{\tau\in\mathcal{N}_{\uparrow}(\sigma)}(M_{\uparrow}(h_\sigma^t, h_\tau^t, h_{\sigma\cup\tau}^t)) \tag{5}$$

Then, the updating function considers these two types of messages and the previous color of σ:

$$h^{t+1}(\sigma) = U(h_\sigma^t, m_{\mathcal{B}}^t(\sigma), m_{\uparrow}^t(\sigma)) \tag{6}$$

After L layers of the message passing process, the readout function takes the color multi-sets at all dimensions as input:

$$h_P = \mathrm{READOUT}(\{\{h_\sigma^L\}\}_{dim(\sigma)=0}, \cdots, \{\{h_\tau^L\}\}_{dim(\tau)=p}) \tag{7}$$

Theorem C.9. *PCMP with sufficient layers and injective neighborhood aggregators are as powerful as PWL.*

C.4 PROOF OF MAIN RESULTS

In order to prove the main results, we give some notations.

Definition C.10 (Path Coloring Refinement)**.** A path coloring c refines a path coloring d, denoted by $c \sqsubseteq d$, if for any path complex P_1, P_2 and $\sigma \in P_1$, $\tau \in P_2$, $c_\sigma^{P_1} = c_\tau^{P_2}$ implies $d_\sigma^{P_1} = d_\tau^{P_2}$. Additionally, if $d \sqsubseteq c$, we say that c and d are equivalent.

Lemma C.11. *Given two path complexes P_1, P_2 with $A \subset P_1$, $B \subset P_2$. Assume c and d are two path coloring such that $c \sqsubseteq d$. If $\{\{d_\sigma^{P_1} | \sigma \in A\}\} \neq \{\{d_\tau^{P_2} | \tau \in B\}\}$, then $\{\{c_\sigma^{P_1} | \sigma \in A\}\} \neq \{\{c_\tau^{P_2} | \tau \in B\}\}$.*

Proof. Let $C_1 = \{\{c_\sigma^{P_1} | \sigma \in A\}\}$, $C_2 = \{\{c_\tau^{P_2} | \tau \in B\}\}$. Assume $C_1 = C_2$, then there is a bijection $f : A \to B$ such that $\forall \sigma \in A$, $\tau = f(\sigma)$, we have $c_\sigma^{P_1} = c_\tau^{P_2}$. From $c \sqsubseteq d$ we know $d_\sigma^{P_1} = d_\tau^{P_2}$. Consequently, $\{\{d_\sigma^{P_1} | \sigma \in A\}\} = \{\{d_{f(\sigma)}^{P_2} | \sigma \in A\}\} = \{\{d_\tau^{P_2} | \tau \in B\}\}$, which contradicts with the condition that $\{\{d_\sigma^{P_1} | \sigma \in A\}\} \neq \{\{d_\tau^{P_2} | \tau \in B\}\}$. Hence the assumption is wrong. □

Corollary C.12. *Given two path colorings c and d such that $c \sqsubseteq d$. If $d^{P_1} \neq d^{P_2}$, then $c^{P_1} \neq c^{P_2}$.*

Proof. This follows by replacing the subsets A, B by the sets of n-paths of P_1 and P_2 respectively in the proof of Lemma C.11. □

The above corollary C.12 means that if c refines d, then c is able to distinguish all the path complex pairs that d can distinguish. In this sense, we can say that c is at least as powerful as d. If c and d are equivalent, we say they have the same expressive power.

Proof of Theorem C.2. It is easy to see that if $G_1 \cong G_2$, then $P_{G_1} \cong P_{G_2}$. The inverse statement follows from the fact that any graph is a subcomplex of its derived path complex by considering the 0-paths and 1-paths. □

Proof of Theorem C.6. Let a^t be the coloring at iteration t of the updating startegy

$$\mathrm{HASH}\{a_\sigma^t, a_{\mathcal{B}}^t(\sigma), a_{\mathcal{C}}^t(\sigma), a_\uparrow^t(\sigma), a_\downarrow^t(\sigma)\}$$

b^t be the coloring at iteration t of the updating strategy

$$\mathrm{HASH}\{b_\sigma^t, b_{\mathcal{B}}^t(\sigma), b_\uparrow^t(\sigma), b_\downarrow^t(\sigma)\}$$

c^t be the coloring at iteration t of the updating strategy

$$\mathrm{HASH}\{c_\sigma^t, c_{\mathcal{B}}^t(\sigma), c_\uparrow^t(\sigma)\}$$

We firstly prove that a^t and b^t are equivalent, then prove that b^t and c^t are equivalent.

1. a^t and b^t are equivalent. We have $a^t \sqsubseteq b^t$ because a^t contains additional colors of its coface neighbors in the color updating rule. It suffices to prove that $b^t \sqsubseteq a^t$. We do this by induction. The base case holds since all the paths are initialized with the same color. Assume the result holds for $t = k$, we prove that $b^{k+1} \sqsubseteq a^{k+1}$. Let $\sigma \in P_1$ and $\tau \in P_2$ be two n-paths from two arbitrary path complexes, suppose $b_\sigma^{k+1} = b_\tau^{k+1}$, we prove that $a_\sigma^{k+1} = a_\tau^{k+1}$.

   The equation $b_\sigma^{k+1} = b_\tau^{k+1}$ means that the hash function at iteration t+1 have the same arguments. Consequently, $b_\sigma^k = b_\tau^k$, $b_{\mathcal{B}}^k(\sigma) = b_{\mathcal{B}}^k(\tau)$, $b_\uparrow^k(\sigma) = b_\uparrow^k(\tau)$, $b_\downarrow^k(\sigma) = b_\downarrow^k(\tau)$. We prove that $b_{\mathcal{C}}^k(\sigma) = b_{\mathcal{C}}^k(\tau)$.

   We have $b_\uparrow^k(\sigma) = b_\uparrow^k(\tau)$ and

   $$b_\uparrow^k(\sigma) = \{\{(b_e^k, b_{\sigma \cup e}^k) | e \in \mathcal{N}_\uparrow(\sigma)\}\}, b_\uparrow^k(\tau) = \{\{(b_e^k, b_{\tau \cup e}^k) | e \in \mathcal{N}_\uparrow(\tau)\}\} \tag{8}$$

   Replacing the first component of the tuple by the same color, we have

   $$\{\{(-, b_{\sigma \cup e}^k) | e \in \mathcal{N}_\uparrow(\sigma)\}\} = \{\{(-, b_{\tau \cup e}^k) | e \in \mathcal{N}_\uparrow(\tau)\}\} \tag{9}$$

By the definition of upper adjacency and coface we have

$$b^k_{\mathcal{C}}(\sigma) = \{\{b^k_w | w \in \mathcal{C}(\sigma)\}\} = \{\{b^k_{\sigma \cup e} | e \in \mathcal{N}_\uparrow(\sigma)\}\} \tag{10}$$

$$b^k_{\mathcal{C}}(\tau) = \{\{b^k_w | w \in \mathcal{C}(\tau)\}\} = \{\{b^k_{\tau \cup e} | e \in \mathcal{N}_\uparrow(\tau)\}\} \tag{11}$$

Combining Equation (8), (9), (10), (11), we have $b^k_{\mathcal{C}}(\sigma) = b^k_{\mathcal{C}}(\tau)$.

From the induction hypothesis, we have $a^k_\sigma = a^k_\tau$, $a^k_{\mathcal{B}}(\sigma) = a^k_{\mathcal{B}}(\tau)$, $a^k_{\mathcal{C}}(\sigma) = a^k_{\mathcal{C}}(\tau)$, $a^k_\uparrow(\sigma) = a^k_\uparrow(\tau)$, $a^k_\downarrow(\sigma) = a^k_\downarrow(\tau)$, so $a^{k+1}_\sigma = a^{k+1}_\tau$.

2. b^t and c^t are equivalent. Similarly we have $b^t \sqsubseteq c^t$, we further prove that $c^{2t} \sqsubseteq b^t$. We do this by induction. The base case is obvious because all the paths are initialized with the same color. Assume the results holds for $t = k$, we prove that $c^{2k+2} \sqsubseteq b^{k+1}$. Let $\sigma \in P_1$ and $\tau \in P_2$ be two n-paths from two arbitrary path complexes, suppose $c^{2k+2}_\sigma = c^{2k+2}_\tau$, we prove that $b^{k+1}_\sigma = b^{k+1}_\tau$.

   For $c^{2k+2}_\sigma = c^{2k+2}_\tau$, by going back two steps of the hash function, we have $c^{2k}_\sigma = c^{2k}_\tau, c^{2k}_{\mathcal{B}}(\sigma) = c^{2k}_{\mathcal{B}}(\tau), c^{2k}_\uparrow(\sigma) = c^{2k}_\uparrow(\tau)$. We want to prove that $c^{2k}_\downarrow(\sigma) = c^{2k}_\downarrow(\tau)$.

   Assume $c^{2k}_\downarrow(\sigma) \neq c^{2k}_\downarrow(\tau)$, then there is a color pair (c_0, c_1) such that (c_0, c_1) appears more times in $c^{2k}_\downarrow(\sigma)$ (without loss of generality) than in $c^{2k}_\downarrow(\tau)$. For any path δ and λ, define

$$A(\delta) = \{\{(c^{2k}_\phi = c_0, c^{2k}_\delta = c_1) | \phi \in \mathcal{C}(\delta)\}\} \tag{12}$$

$$C_\lambda = \{\{|A(\delta)| | \delta \in \mathcal{B}(\lambda)\}\} \tag{13}$$

   Then we have

$$C_\sigma = \{\{|A(\delta)| | \delta \in \mathcal{B}(\sigma)\}\} = \{\{|(c^{2k}_\phi = c_0, c^{2k}_\delta = c_1)| | \delta \in \phi \cap \sigma\}\} \tag{14}$$

$$C_\tau = \{\{|A(\delta)| | \delta \in \mathcal{B}(\tau)\}\} = \{\{|(c^{2k}_\phi = c_0, c^{2k}_\delta = c_1)| | \delta \in \phi \cap \tau\}\} \tag{15}$$

   So $C_\sigma \neq C_\tau$.

   Considering the path coloring $d(\delta) = |A(\delta)|$. For two n-paths δ_1, δ_2, if $d(\delta_1) \neq d(\delta_2)$, we can assume that $|A(\delta_1)| > |A(\delta_2)|$ without loss of generality, then the number of upper adjacent neighbors of δ_1 and δ_2 up to color pair (c_0, c_1) are different, which means $c^{2k}_\uparrow(\delta_1) \neq c^{2k}_\uparrow(\delta_2)$. So $c^{2k+1}_{\delta_1} \neq c^{2k+1}_{\delta_2}$, which means $c^{2k+1} \sqsubseteq d$.

   Applying Lemma C.11 to $\mathcal{B}(\sigma)$ and $\mathcal{B}(\tau)$, we have

$$\{\{c^{2k+1}_{\delta_1} | \delta_1 \in \mathcal{B}(\sigma)\}\} \neq \{\{c^{2k+1}_{\delta_2} | \delta_2 \in \mathcal{B}(\tau)\}\} \tag{16}$$

   The above multi-sets are exactly the color multi-sets of the faces of σ and τ, which means $c^{2k+1}_{\mathcal{B}}(\sigma) \neq c^{2k+1}_{\mathcal{B}}(\tau)$. Consequently, $c^{2k+2}_\sigma \neq c^{2k+2}_\tau$, which contradicts with the induction hypothesis, so $c^{2k}_\downarrow(\sigma) = c^{2k}_\downarrow(\tau)$.

   From the induction hypothesis, we have $b^k_\sigma = b^k_\tau$, $b^k_{\mathcal{B}}(\sigma) = b^k_{\mathcal{B}}(\tau)$, $b^k_\uparrow(\sigma) = b^k_\uparrow(\tau)$, $b^k_\downarrow(\sigma) = b^k_\downarrow(\tau)$, so $b^{k+1}_\sigma = b^{k+1}_\tau$.

□

Proof of Theorem C.7. Given a path complex P, let a^t be the coloring of the vertices of P at iteration t of WL and b^t be the coloring of the same vertices at iteration t of PWL. We firstly prove that $b^t \sqsubseteq a^t$, then give a pair of graphs to show that they cannot be differentiated by WL but can be differentiated by PWL.

1. $b^t \sqsubseteq a^t$. We do this by induction. The base case holds because all vertices are initialized with the same color. Suppose the result holds for $t = k$, we prove that $b^{k+1} \sqsubseteq a^{k+1}$. Let v and w be two vertices of two arbitrary path complexes P_1, P_2, suppose $b^{k+1}_v = b^{k+1}_w$, we prove that $a^{k+1}_v = a^{k+1}_w$.

Note that vertices only has upper adjacent neighbors, so we have $b_v^k = b_w^k, b_\uparrow^k(v) = b_\uparrow^k(w)$. The second equation means

$$\{\{b_x^k | (b_x^k, -) \in b_\uparrow^k(v)\}\} = \{\{b_y^k | (b_y^k, -) \in b_\uparrow^k(w)\}\}$$

This can be equivalently written as

$$\{\{b_x^k | x \in \mathcal{N}_\uparrow(v)\}\} = \{\{b_y^k | y \in \mathcal{N}_\uparrow(w)\}\}$$

From the induction hypothesis, we have $a_v^k = a_w^k$ and

$$\{\{a_x^k | x \in \mathcal{N}_\uparrow(v)\}\} = \{\{a_y^k | y \in \mathcal{N}_\uparrow(w)\}\}$$

These are the arguments of the hash function for WL to compute the colors of v and w in the next iteration, so $a_v^{k+1} = a_w^{k+1}$.

2. Considering the graphs in Figure 7, they cannot be differentiated by WL test. In PWL test, the path complex derived from the right graph has not any 3-path while the derived path complex from the left graph has 3-paths.

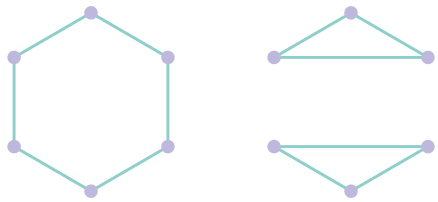


Figure 7: Two graphs that cannot be distinguished by WL but can be differentiated by PWL.

□

Proof of Theorem C.8. Considering the graphs in Figure 8, they cannot be differentiated by SWL test. In PWL test, the path complex derived from the right graph has not any 4-path while the derived path complex from the left graph has 4-paths. □

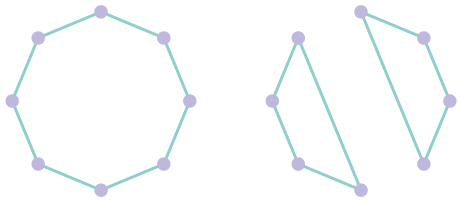


Figure 8: Two graphs that cannot be distinguished by SWL but can be differentiated by PWL.

Proof of Theorem C.9. Let b^t and d^t be the coloring at iteration t of PWL and the t-th layer of an PCMP respectively. Assume the PCMP has L layers and assume $d^t = d^L (t > L)$. We use induction to prove that $d^t \sqsubseteq b^t$. The base case holds by definition. Suppose the result holds for $t = k$, when $t = k + 1$, we prove that $d^{k+1} \sqsubseteq b^{k+1}$. For any two n-paths σ, τ of any two path complexes P_1, P_2 such that $d_\sigma^{k+1} = d_\tau^{k+1}$, we prove that $b_\sigma^{k+1} = b_\tau^{k+1}$.

The condition means all the update, aggregate and message functions are injective, so their composition is also injective. Hence $d_\sigma^k = d_\tau^k$, $d_\mathcal{B}^k(\sigma) = d_\mathcal{B}^k(\tau)$, $d_\uparrow^k(\sigma) = d_\uparrow^k(\tau)$.

$d_\mathcal{B}^k(\sigma) = d_\mathcal{B}^k(\tau)$ means

$$\{\{d_s^k | s \in \mathcal{B}(\sigma)\}\} = \{\{d_t^k | t \in \mathcal{B}(\tau)\}\}$$

$d^k_\uparrow(\sigma) = d^k_\uparrow(\tau)$ means

$$\{\{(d^k_s, d^k_{s\cup\sigma}) | s \in \mathcal{N}_\uparrow(\sigma)\}\} = \{\{(d^k_t, d^k_{t\cup\tau}) | t \in \mathcal{N}_\uparrow(\tau)\}\}$$

By the induction hypothesis, we have $b^k_\sigma = b^k_\tau$.

$$\{\{b^k_s | s \in \mathcal{B}(\sigma)\}\} = \{\{b^k_t | t \in \mathcal{B}(\tau)\}\}$$

$$\{\{(b^k_s, b^k_{s\cup\sigma}) | s \in \mathcal{N}_\uparrow(\sigma)\}\} = \{\{(b^k_t, b^k_{t\cup\tau}) | t \in \mathcal{N}_\uparrow(\tau)\}\}$$

So $b^k_\sigma = b^k_\tau$, $b^k_{\mathcal{B}}(\sigma) = b^k_{\mathcal{B}}(\tau)$, $b^k_\uparrow(\sigma) = b^k_\uparrow(\tau)$, these are the arguments of the hash function in PWL, so $b^{k+1}_\sigma = b^{k+1}_\tau$.

□