Abstract

This paper introduces higher-order attention networks (HOANs), a novel class of attention-based neural networks defined on a generalized higher-order domain called a combinatorial complex (CC). Similar to hypergraphs, CCs admit arbitrary set-like relations between a collection of abstract entities. Simultaneously, CCs permit the construction of hierarchical higher-order relations analogous to those supported by cell complexes. Thus, CCs effectively generalize both hypergraphs and cell complexes and combine their desirable characteristics. By exploiting the rich combinatorial nature of CCs, HOANs define a new class of message-passing attention-based networks that unifies higher-order neural networks. Our evaluation of HOANs on tasks related to mesh shape analysis and graph learning demonstrates a competitive, and in some examples superior, predictive performance in comparison to state-of-the-art neural networks.

1 Introduction

Graphs have been widely used to model relational data in various fields and represent associations ranging from social interactions to chemical and physical interactions. By leveraging ideas from classical signal processing [71], graph theory [89], and recent advances in deep learning [41, 56], graph-based models have emerged as a powerful set of tools for processing relational data. Although these models achieved promising performance in various applications [17, 51], they only model binary relations among data, thus not being applicable in the presence of multi-way or higher-order relations [19, 11]. Nevertheless, higher-order relations are prevalent in numerous real-world applications such
as higher-order interactions in physical complex systems [17], brain functions [33] and higher-order protein interactions [103].

Recently, machine learning models have been developed to model data with support on topological spaces modeled as simplicial complexes [108, 86, 31, 77, 22, 45, 44, 11], cell complexes [43, 82, 20, 78] and hypergraphs [101, 9, 50, 4, 35, 10, 37]. These models have been used in link prediction [26, 44], optimal homology generator detection [52], and trajectory prediction [77]. This new research trend, collectively described as higher-order models, intertwines topological data analysis [42, 24], topological signal processing [80, 107, 88, 78, 11, 75, 81] and geometric deep learning [109, 23, 36, 65, 16, 70, 15, 54]. Current higher-order models have been defined on domains that allow modeling hierarchical and geometric-based relations (e.g., simplicial models [86, 31, 77, 22, 45] and cellular models [43]) or set-based relations (e.g., hypergraph models [98, 106, 50, 4, 35, 37, 73]). However, current models fail to represent relations that are hierarchical and they do not admit the rigid geometric constraints of simplicial or cellular complexes, as demonstrated in Figure 1.

The majority of higher-order deep learning models focus on layers that use isotropic aggregation, where neighbors in the vicinity of an element contribute equally to an update of that element’s representation. In contrast, attention-based learning [27] allows deep learning models to assign a probability distribution to neighbors in local vicinity of elements in the domain and highlight components with the most task-relevant information [96]. Attention-based models are successful in practice as they ignore noise in the underlying domain, thus improving the signal-to-noise ratio [58, 69]. Such models achieved remarkable success on various tasks, including node classification and link prediction [59], node ranking [91], and attention-based embeddings [27, 57].

**Contribution.** In this paper, we propose a generalized higher-order domain called combinatorial complex (CC) and utilize it to build a new class of attention-based neural networks called higher-order attention networks (HOANs). CCs generalize many discrete domains that are of practical importance such as point clouds, 3D shapes, (hyper)graphs, simplicial complexes, and cell complexes [48], as illustrated in Figure 1. The topological structure of a CC encodes arbitrary higher-order interactions among elements of the CC. By exploiting the rich combinatorial and topological structure of CCs, HOANs define a new class of higher-order message passing attention-based networks that unify existing higher-order models based on hypergraphs and cell complexes. We demonstrate the reducibility of any CC to a special graph called the Hasse graph, which allows the characterization of certain aspects of HOANs in terms of graph-based models. Further, we define and study the permutation equivariance of HOANs. Finally, we demonstrate the predictive capacity of HOANs in shape analysis and in graph learning, competing against state-of-the-art task-specific neural networks.

**Roadmap.** Section 2 reviews previous work. Sections 3 and 4 introduce CCs and HOANs, respectively. Section 5 inspects theoretical aspects of HOANs including their relation to augmented Hasse graphs and HOAN equivariance. Section 6 provides experiments that demonstrate the competitive or superior performance of HOANs across datasets from different domains; e.g., shapes and graphs. The proofs of propositions and a glossary of mathematical notation are provided in the appendices.
2 Related Work

Higher-order models. An increasing interest in higher-order models has manifested itself in recent years. In the signal processing literature, a Hodge-theoretic approach over simplicial complexes has been introduced [87,11]. This effort has been extended to hypergraphs in [13,87] and cell complexes in [82,78]. In the deep learning literature, convolutional operators and message passing algorithms have been developed for higher-order models. For example, a convolutional operator on hypergraphs has been proposed in [50,35]. Exploiting the Hodge Laplacian operator for linear filtering (e.g., [85,11,12,87]), the work proposed in [76] utilizes the 1-Hodge operator to define an edge-based convolutional deep learning model. The work of [43] introduced a higher-order message passing framework that encompasses those given in [31,22,38,77] and utilizes various local neighborhood aggregation schemes. Further, a generalization of skip connection [49,79] to simplicial complexes has been introduced in [44], allowing training higher-order deep models. In [70], a higher-order graph neural network that takes higher-order graph structures at multiple scales into account was proposed.

Attention-based models. Real-world relational data are large, unstructured, sparse, and noisy, which causes Graph Neural Network (GNNs) [102] to learn suboptimal data representations and significantly impact GNN performance [29,6]. To address these issues, various attention mechanisms [25] have been integrated into GNNs to focus on the most relevant parts of a given graph while ignoring irrelevant parts. Based on the used mechanism, existing graph attention approaches can be divided into weight-based attention (e.g., GAKE [34], GAT [96], CCM [110]), similarity-based attention (e.g., AGNN [92], graph2seq [105]), and attention-guided walk (e.g., AttentionWalks [2], GAM [57]).

The majority of attention-based models, with the exception of [40,59,10,53], are designed for graphs. The attention model proposed in [40] is a straightforward generalisation of the graph attention model given in [96]. In [39], the authors utilize a model based on Hodge decomposition, similar to the one suggested in [77], to introduce an attention model on simplicial complexes. The hypergraph attention models introduced in [10,53] provide a generalization of the attention model similarly to [96]. All above attention models neither allow for higher-order attention between entities of different dimensions nor combine these attention blocks, thus greatly limiting the space of architectures and the scope of applications of these models. We propose and design HOANs to overcome these limitations.

3 Combinatorial Complexes (CCs)

This section introduces CCs, a novel class of objects that encompass both hypergraphs and cell complexes. Similar objects exist in geometric topology [5,14,84]; in fact, we have adapted the definitions of these objects to utilize them as the domains upon which our higher-order deep models are developed. CCs have both hierarchical and set-like characteristics, as illustrated in Figure 1. The hierarchical properties improve the expressive power [70] and allow simultaneous processing of signals on various ranked elements of a domain [11]. On the other hand, many applications based on higher-order relations (e.g., protein interactions [104]) require flexibility in the nature of these relations, beyond the confines of geometric-based relations present in cell or simplicial complexes, which is why set-like characteristics are also desirable.

Definition 1. Let $S$ be a non-empty finite set and $P(S)$ its power set. A combinatorial complex (CC) is a tuple $(X, \iota)$ formed by a set $X \subset P(S) \setminus \emptyset$ together with a rank function $\iota: X \to \mathbb{Z}_+^+$ such that (i) $\iota(\{x\}) = 0$ for all $x \in S$, and (ii) for all $x, y \in X$, if $x \subset y$ then $\iota(x) < \iota(y)$.

Definition 1 directly yields $S \subseteq X$. The elements of $X$ are called cells. The value of the rank function $\iota(x)$ on the cell $x \in X$ is called the rank of $x$. Given $k \in \mathbb{Z}_+$, cells with rank $k$ are called $k$-cells. The 0-cells, i.e., the element of $S$, are called the nodes or the vertices of the CC. Similarly, the 1-cells are called the edges of the CC. Thereby, edges can have more than two nodes. In this paper, we will mainly work with CCs whose edges have exactly two vertices (graph-based CCs). A cell $x$ of rank $k$ is denoted by $x^k$ when we need to make its rank explicit. The dimension of $X$, denoted by dim($X$), is defined as the maximal rank among its cells.

To exemplify Definition 1 consider the set $S = \{s_0, s_1, s_2\}$, depicted by three orange dots in Figure 2. We set $X$ to be the set consisting of the following five subsets of $S$: $x_0 = \{s_0\}$, $x_1 = \{s_1\}$ and $x_2 = \{s_2\}$, $x_3 = \{s_1, s_2\}$ (purple) and $x_4 = \{s_0, s_1, s_2\} = S$ (blue). We turn $X$ into a CC by
We define the $k$-th adjacency and coadjacency between $x$ and $y$ as the sets of cells incident to $x$ and $y$, respectively.

CCs generalize hypergraphs. Indeed, hyperedges in a hypergraph correspond to cells in a CC, where the rank of the higher order cells is ignored. The relations of CCs to other complexes are explained in more detail in Appendix C. Maps relating CCs are defined next.

Definition 2. Let $X$ and $Y$ be two CCs. A function $f : X \rightarrow Y$ is a CC-homomorphism if for every $x, y \in X$ with $x \subseteq y$, we have $f(x) \subseteq f(y)$. The CCs $X$ and $Y$ are considered to be equivalent if $f$ is a bijection and $f^{-1}$ is a CC-homomorphism.

Definition 3. Let $A \subseteq S$ be a subset of all the possible vertices of a CC $(X, i)$, where $X \in \mathcal{P}(S) \setminus \emptyset$. The set $X_A = \{x \in X | x \subseteq A\}$ along with the restriction $i|_{X_A}$ define a CC that we call the subcomplex of $X$ induced by $A$.

In the previous example (Figure 2), the set $X_A = \{\{s_1\}, \{s_2\}, \{s_1, s_2\}\}$ is the subcomplex induced by $A = \{s_1, s_2\}$ in $X$. According to Definition 3, any subcomplex is a CC and any cell in the set $X$ of a CC $(X, i)$ is a subcomplex. The term complex object will be used to refer to a cell or to a subcomplex in $X$.

Incidence in a CC. We introduce the combinatorial structure of ‘incidence’, which will be used to extend message passing schemes on graphs and on cell complexes to a general message passing scheme on CCs. In what follows, we assume that cells in the set $X$ of a CC $(X, i)$ are given a fixed order. For $k \in \mathbb{Z}^+$, we consider the set of all $k$-cells, called the $k$-skeleton of $X$ and denoted by $X^k = i^{-1}(|k|)$. Moreover, we denote the cardinality of $X^k$, that is the number of $k$-cells, by $|X^k|$.

Definition 4. Two cells $x$ and $y$ of a CC $(X, i)$ are said to be incident if either $x \subseteq y$ or $y \subseteq x$. For any $r, k \in \mathbb{Z}^+$ with $0 \leq r < k \leq \dim(X)$, the $(r, k)$-incidence matrix $B_{r,k}$ between skeletons $X^r$ and $X^k$ of the CC is defined to be the $|X^r| \times |X^k|$ matrix with $[B_{r,k}]_{ij} = 1$ if $x_i^r$ is incident to $x_j^k$ and $[B_{r,k}]_{ij} = 0$ otherwise.

Figure 3 exemplifies Definition 4 by visualizing the notion of incidence matrix $B_{0,2}$ on a triangular mesh. Incidence matrices on CCs generalize incidence matrices on (hyper)graphs. The set of faces of a cell $x \in X$ is defined by $N_\Delta(x) = \{y \in X | y \subseteq x, i(y) = i(x) + 1\}$. Similarly, the set of cofaces of a cell $x$ is defined by $N_\nabla(x) = \{y \in X | y \subseteq x, i(y) = i(x) - 1\}$. In Figure 3 the blue triangles that are incident to the central node (central orange dot) are marked by black arrows, and correspond to non-zero values on the central node’s row in $B_{0,2}$.

Adjacency in a CC. The incidence matrices $B_{r,k}$ of Definition 4 induce a neighboring structure, which is utilized to construct a higher-order message passing scheme framework on CCs in Section 4.2. Besides, CCs admit other neighboring structures. For instance, for a CC that is reduced to a graph, a more natural neighboring structure is the adjacency relation. Along these lines, for an arbitrary CC $(X, i)$, we define the adjacency set of a cell $x \in X$ as

$$N_\Delta(x) = \{y \in X | i(y) = i(x), \exists z \text{ with } i(z) = i(x) + 1 \text{ such that } x, y \subseteq z\},$$

and the coadjacency set of $x$ as

$$N_\nabla(x) = \{y \in X | i(y) = i(x), \exists z \text{ with } i(z) = i(x) - 1 \text{ such that } z \subseteq x\}.$$  

We define the $k$-adjacency matrix $|A_k|$ and $k$-coadjacency matrix $|\text{co}A_k|$ as the matrices induced by the adjacency and coadjacency between $k$-cells, respectively. Note that incidence defines relations among cells of different ranks while adjacency defines relations among cells of similar ranks.

1The entries of the matrices $|A_k|$ and $|\text{co}A_k|$ are in $\{0, 1\}$ as they encode (co)adjacency set relations. We also work with a signed version $A_k$ and $\text{co}A_k$ of these matrices, where the entries are written with respect to a given reference orientation on the cells.
Cochain spaces and maps. We are interested in processing data defined over a CC \((X, i)\). To this end, let \(C^k(X, \mathbb{R}^d)\) be the \(\mathbb{R}\)-vector space of functions \(x_k : X^k \to \mathbb{R}^d\) for a rank \(k \in \mathbb{Z}^+\) and data dimension \(d\). For \(d = 1\), we use the notation \(C^k(X)\) or \(C^k\) when the underlying CC is clear. Elements in \(C^k(X, \mathbb{R}^d)\) are called \(k\)-cochains, and \(C^k(X, \mathbb{R})\) is the \(k\)-cochain space. Intuitively, a \(k\)-cochain can be interpreted as a signal defined on the \(k\)-cells of \(X\) [42]. In particular, when \(X\) is a graph, 0-cochains correspond to the graph signals [22]. We also work with maps between cochain spaces, which are called cochain maps. For instance, an incidence matrix \(B_{r,k} : C^k(X) \to C^r(X)\) that sends a \(k\)-cochain \(x_k \in C^k(X)\) to an \(r\)-cochain \(B_{r,k}(x_k) \in C^r(X)\). Similarly, an adjacency matrix \(|A_k|\) induces a map \(|A_k| : C^k(X) \to C^k(X)\).

4 Higher-Order Attention Networks (HOANs)

Section 4.1 introduces the building blocks for constructing HOANs in Section 4.2.

4.1 Higher-order Attention Building Blocks

Consider a target complex object \(Y_0 \subset X\) and a set \(N_{Y_0} = \{Y_1, \ldots, Y_{|N_{Y_0}|}\}\) of complex objects in the vicinity of \(Y_0\), where the notion of vicinity is task-specific. A higher-order attention is a function \(f : Y_0 \times N_{Y_0} \to [0, 1]\) that provides a weight to each object in \(N_{Y_0}\) such that \(\sum_{i=1}^{|N_{Y_0}|} f(Y_0, Y_i) = 1\). We describe the notion of vicinity of \(X\) in terms of a neighborhood matrix; each column \(j\) of the neighborhood matrix describes a local relation of a complex object \(j\) to a collection of other complex objects in \(X\) (see Fig 5(a),(c)). The incidence and adjacency matrices introduced in Section 3 can be used as neighborhood matrices. The shape of a neighborhood matrix impacts the associated attention block.

Squared neighborhood matrices. Examples of squared neighborhood matrices include the adjacency \(|A_k|\), coadjacency \(|\text{co}A_k|\) and their powers \(|A_k|^n\), \(|\text{co}A_k|^n\), which describe the \(n\)-hop neighborhood of a \(k\)-cell. Squared neighborhood matrices can be used to induce attention among cells of the same dimension. For these matrices, higher-order attention is a relatively straightforward generalization of graph attention [96] (see Definition 5). In what follows, \([\cdot]\) denotes concatenation.

Definition 5. Let \(\mathcal{A} : C^* (X) \to C^* (X)\) be a squared neighborhood matrix. The higher-order attention block (HB) induced by \(\mathcal{A}\) is a cochain map \(\text{HB}_\mathcal{A}(W_{s,p}) : C^*(X, \mathbb{R}^{d_{s,in}}) \to C^*(X, \mathbb{R}^{d_{s,out}})\) defined via

\[
x^l_s \to x^{l+1}_s := \phi \left( \sum_{p=1}^m (A^p \odot \text{att}_p^l) * x^l_s * W_{s,p}^l \right),
\]

where \(\phi\) is a non-linearity, \(\odot\) is the Hadamard product, \(W_{s,p}^l \in \mathbb{R}^{d_{s,in} \times d_{s,out}}\) are trainable parameters, and \(\text{att}_p^l : C^*(X) \to C^*(X)\) is a higher-order attention matrix that has the same dimension as matrix \(\mathcal{A}\). The \((i, j)\)-th entries of \(\text{att}_p^l\) are given by

\[
\text{att}_p^l(i, j) = \frac{e_{ij}^l}{\sum_{k \in N_{\mathcal{A}_p}(i) \in e_{ik}^l}},
\]

where \(e_{ij}^l = \text{LeakyReLU}(a_i^l)^T W_{s,p}^l x^l_s || W_{s,p}^l x^l_s\), \(a_i^l \in \mathbb{R}^{2 \times s_{out}}\), and \(N_{\mathcal{A}_p}(i)\) represents the neighborhood of the cell \(i\) with respect to the matrix \(\mathcal{A}_p\).

Non-squared neighborhood matrices. The incidence matrices of Definition 4 are examples of (generally) non-squared neighborhood matrices. Particularly, an incidence matrix \(B_{k-1,k}\) represents a neighborhood relation between cells of dimensions \(k\) and \(k-1\). The \(n\)-hop incidence matrices, which are defined as \((B_{k-1,k})^n := B_{k-1,k} \cdot A_k^n + B_{k-1,k} \cdot \text{co}A_k^n\), are also examples of non-squared neighborhood matrices. Figure 3 visualizes examples of 0-hop and 1-hop incidence matrices. Non-squared neighborhood matrices can be used to induce attention among cells of different dimensions. For these matrices, Figure 4: The 0-hop boundary of the blue edge has two orange nodes, while the 1-hop boundary of the same edge consists of four red nodes.
we introduce a new higher-order attention mechanism (see Definition 6). In what follows, \([\cdot, \cdot, \cdot]\) denotes concatenation.

**Definition 6.** Let \(A : C^s(\mathcal{X}) \rightarrow C^t(\mathcal{X})\) be a neighborhood matrix with \(s \neq t\). The higher-order attention block induced by \(A\) is a cochain map \(H_{B}(W_s^l, W_t^l) : C^s(\mathcal{X}, \mathbb{R}^{d_{in}}) \times C^t(\mathcal{X}, \mathbb{R}^{d_{out}}) \rightarrow C^t(\mathcal{X}, \mathbb{R}^{d_{out}^t}) \times C^s(\mathcal{X}, \mathbb{R}^{d_{out}^s})\) defined via

\[
(x_s^l, x_t^l) \rightarrow (x_s^{l+1}, x_t^{l+1}),
\]

with

\[
x_s^{l+1} = \phi((A \circ att_{s \rightarrow t}^l) \ast x_s^l * W_s^l), \quad x_t^{l+1} = \phi((A^T \circ att_{t \rightarrow s}^l) \ast x_t^l * W_t^l),
\]

where \(W_s^l \in \mathbb{R}^{d_{in} \times d_{out}^s}, W_t^l \in \mathbb{R}^{d_{in} \times d_{out}^t}\) are trainable parameters, and \(att_{s \rightarrow t}^k : C^s(\mathcal{X}) \rightarrow C^t(\mathcal{X})\) are higher-order attention matrices that have the same dimensions as the matrices \(A\) and \(A^T\), respectively. The \((i, j)\)-th entries of matrices \(att_{s \rightarrow t}^l\) and \(att_{t \rightarrow s}^l\) are defined as

\[
att_{s \rightarrow t}^l(i, j) = \frac{e_{ij}^l}{\sum_{k \in N_A(i)} e_{ik}^l}, \quad att_{t \rightarrow s}^l(i, j) = \frac{f_{ij}^l}{\sum_{k \in N_{A^T}(i)} f_{ik}^l},
\]

where

\[
e_{ij}^l = \text{LeakyRelu}((a^l)^T[W_s^l x_s^l][W_t^l x_t^l]), \quad f_{ij}^l = \text{LeakyRelu}((\text{rev}(a^l))^T[W_s^l x_s^l][W_t^l x_t^l]),
\]

with \(a^l \in \mathbb{R}^{t_{out} \times s_{out}}\) and \(\text{rev}(a^l) = [a^l[: \ t_{out}]] || a^l[t_{out} :]]\).

In Figure 5 we illustrate the notion of attention for non-squared matrices. Figure 5(c) shows the non-squared matrix \(B_{1,2}\) associated with the CC displayed in Figure 5(a). The attention block \(H_{B_{1,2}}\) learns two incidence matrices \(att_{s \rightarrow t}\) and \(att_{t \rightarrow s}\). The matrix \(att_{s \rightarrow t}\) has the same shape as \(B_{1,2}\), and non-zero elements exactly where \(B_{1,2}\) has 1’s. Each column \(i\) in \(att_{s \rightarrow t}\) represents a probability distribution that defines the attention of the \(i\)-th 2-cell to its incident 1-cells. The matrix \(att_{t \rightarrow s}\) has the same shape as \(B_{1,2}^T\), and similarly represents the attention of 1-cells to 2-cells.

![Diagram](image)

**Figure 5:** (a) Each 2-cell (blue face) attends to its incident 1-cells (pink edges). (b) The attention coefficients reside on a graph constructed from the cells and their incidence relations, see Section 4.4 for details. (c) Incidence matrix \(B_{1,2}\) of the CC given in (a). The non-zero elements in column \([1, 2, 3]\) correspond to the neighborhood \(N_{B_{1,2}}([1, 2, 3])\) of \([1, 2, 3]\) with respect to \(B_{1,2}\).

### 4.2 Construction of Higher-Order Attention Models (HOANs)

**Definition 7.** A higher-order attention network supported on a complex \(\mathcal{X}\) is a function of the form

\[
\text{HOAN}_{G, W} : C^{i_1} \times C^{i_2} \times \ldots \times C^{i_m} \rightarrow C^{j_1} \times C^{j_2} \times \ldots \times C^{j_n},
\]

where \(W\) is a trainable parameter and \(G = \{G_1, \ldots, G_k\}\) is a finite sequence of cochain maps defined on \(\mathcal{X}\).

Intuitively, a higher-order attention network takes a vector of cochains \((x_{i_1}, \ldots, x_{i_m})\) as input and returns a vector of cochains \((x_{j_1}, \ldots, x_{j_n})\). The maps \(\{G_i\}_{i=1}^k\) reflect the structure of the underlying complex \(\mathcal{X}\) and are used to determine the flow of the signal on the complex. The choice of these maps, which can be any of the matrices mentioned in Section 4.1, depends on the learning task.
Building HOANs: tensor diagrams. Unlike graphs that involve node or edge signals, higher-order networks entail a higher number of signals. Thus, constructing a HOAN network requires building a non-trivial amount of interacting sub-networks. Due to the relatively large number of cochains in a HOAN, we introduce tensor diagrams, a diagrammatic notation for higher-order networks. A tensor diagram represents a higher-order network on a complex $X$ via a directed graph. The signals on a tensor diagram flow from the source nodes to the target nodes. The source and target nodes correspond to the domain and codomain of the function in Equation (9).

Figure 6 visualizes examples of tensor diagrams. For instance, the tensor diagram of Figure 6(d) represents a HOAN that maps a signal in $C^{i_1} \times C^{i_2} \times \ldots \times C^{i_m} \to C^j$ given by:

$$\sum_{l} M \left( x_{i_1}^l, \ldots, x_{i_m}^l \right) \xrightarrow{M} x_{i+1}^l = M \left( x_{i_1}^l, \ldots, x_{i_m}^l \right),$$

where $G_k : C^{i_k}(X) \to C^j(X)$ are cochain maps and $W$ are trainable parameters. We think of $M$ as a message passing function that takes into account the messages induced by maps $G_1, \ldots, G_m$, which collectively act on a cochain vector $(x_{i_1}, \ldots, x_{i_m})$, to obtain an updated cochain $x_{i+1}$.

Figure 6(c) shows a merge node example. In its simplest form, a merge node is a sum of the cochains obtained from the building blocks of Section 4.2, but in general it can be any trainable function. Vertical concatenation of two tensor diagrams represents the composition of their corresponding networks.

5 Reducibility of HOANs to Graph-Based Models

Section 5.1 shows that any CC can be reduced to a special graph called the Hasse graph, allowing the characterization of certain computational and conceptual aspects HOANs in terms of graph-based models. In Section 5.2, the Hasse graph is used to provide a definition of equivalence for HOANs.

5.1 Reducing CCs to Graphs: Hasse Graphs

Definition 2 implies that a CC is a poset; i.e., a partially ordered set with partial order relation given by set inclusion. It also implies that equivalence among CCs is equivalence among posets. Definition 8 defines the Hasse graph $\mathcal{H}_X$ of a CC, which is a directed graph associated with a finite poset.

**Definition 8.** The Hasse graph $\mathcal{H}_X = (V(\mathcal{H}_X), E(\mathcal{H}_X))$ of a CC $(X, \iota)$ is the graph with vertices $V(\mathcal{H}_X) = X$ and edges $E(\mathcal{H}_X) = \{ (x, y) : x \subseteq y, \iota(x) = \iota(k) - 1 \}$, i.e., the graph whose vertices are the cells of $X$ and whose edges are determined by the immediate incidence among these cells.

Figure 7 shows an example of a Hasse graph of a CC. The information provided via the edges of the Hasse graph is the same as the information given by the incidence matrices $\{ B_{k,k+1} \}_{k=0}^{\dim(X)-1}$.

In the following, the structure of a CC is the CC equivalence class according to Definition 2.

**Proposition 5.1.** Let $(X, \iota)$ be a CC. (i) The CC structure is determined by the matrices $\{ B_{k,k+1} \}_{k=0}^{\dim(X)-1}$. (ii) The CC structure is determined by the matrices $\{ |\operatorname{coA}_k| \}_{k=0}^{\dim(X)-1}$. (iii) The CC structure is determined by the matrices $\{ |\operatorname{coA}_k| \}_{k=0}^{\dim(X)-1}$.

2Graphical notation is common in geometric topology literature [48, 94] and typically used to construct functions built of simpler building blocks.

3For related structures (e.g., simplicial/cell complexes), this poset is typically called the poset face [97].
We use \( \text{Proj}_P \) to denote a similar permutation of the output of the GNN model.\(^{9}\)

Matrices with \( P \)

Definition 9. Let \( X \) be a CC. Let \( G = \{ G_k \} \) be a sequence of cochain maps defined on \( X \) with \( G_k : C^{i_k} \to C^{j_k}, 0 < i_k, j_k \leq \dim(X) \). Let \( \mathbb{P} = \{ P_i \}_{i=0}^{\dim(X)} \) be a collection of permutation matrices with \( P_i \in \{ 0, 1 \}^{|X| \times |X|} \). We define \( \mathbb{P}(G) \) to be the sequence \( \{ P_{j_k} G_k P_i^T \} \).

The notion of permutation equivariance on HOANs is given in Definition 10. A practical version of this equivariance is provided in Proposition 5.3. Specifically, a vanilla graph-based model equivariance is specified by an adjacency matrix of the underlying graph, which encodes the graph structure up to a permutation of the vertex set of the underlying graph. The edges specified by the matrices \( B_{k,k+1} \) determine the message passing structure when executing a given higher-order model defined on \( CC \). From this perspective, it is sometimes desirable to augment the Hasse graph with additional edges other than the ones specified by the poset partial order relation of the CC. Specifically, we may be given neighborhood structures specified by the matrices \( G = \{ G_1, \ldots, G_n \} \), which define a set of additional edges among the nodes of a Hasse graph.

We call the graph with this augmented structure the augmented Hasse graph and we denote it by \( H_X(G) \). For instance, the graph given in Figure 7(c) is \( H_X(|A_0|, |coA_2|) \). Every merge node can be characterized in terms of a subgraph of the augmented Hasse graph as follows.

Proposition 5.2. Any computation executed via a merge node \( M_{G,W} \) as given in Equation (10) can be reduced to a corresponding computation on the augmented graph \( H_X(G) \).

5.2 Equivariance for HOANs

While the Hasse graph shows that certain computations for HOANs can be reduced to a graph-based model computations. Care must be taken when considering such a reduction because higher-order models in general, and HOANs in particular, should always be considered in conjunction with their underlying equivariance version. Specifically, a vanilla graph-based model equivariance is specified by an adjacency matrix of the underlying graph, which encodes the graph structure up to a permutation of the vertex set of the graph. On the other hand, a Hasse graph in our context, while essentially a graph, is structurally characterized by the equivariance induced by Definition 2. Hence, HOANs equivariance must take into account Definition 2. We refer the reader to [95] for a related discussion. Motivated by Proposition 5.1 which characterizes the structure of CC, this section defines the notions of permutation equivariance of HOANs. Our definition generalizes the equivariance definition given in [77].

Permutation equivariance. The notion of permutation equivariance on HOANs is given in Definition 10. A practical version of this equivariance is provided in Proposition 5.3.

Definition 9. Let \( X \) be a CC. Let \( G = \{ G_k \} \) be a sequence of cochain maps defined on \( X \) with \( G_k : C^{i_k} \to C^{j_k}, 0 < i_k, j_k \leq \dim(X) \). Let \( \mathbb{P} = \{ P_i \}_{i=0}^{\dim(X)} \) be a collection of permutation matrices with \( P_i \in \{ 0, 1 \}^{|X| \times |X|} \). We define \( \mathbb{P}(G) \) to be the sequence \( \{ P_{j_k} G_k P_i^T \} \).

We use \( \text{Proj}_k : C^1 \times \cdots \times C^m \to C^k \) to denote the standard \( k \)-th projection for \( 1 \leq k \leq m \).

Definition 10. Let \( X \) be a CC and let \( G = \{ G_k \} \) be a finite sequence of cochain maps defined on \( X \). Let \( \mathbb{P} = \{ P_i \}_{i=0}^{\dim(X)} \) be a collection of permutation matrices with \( P_i \in \{ 0, 1 \}^{|X| \times |X|} \). A HOAN of the form (9) is permutation-equivariant if

\[
\text{Proj}_k \circ \text{HOAN}_{G,W}(x_{i_1}, \cdots, x_{i_m}) = P_k \text{Proj}_k \circ \text{HOAN}_{\mathbb{P}(G),W}(\{P_i x_{i_1}, \cdots, P_i x_{i_m}\})
\]

\(^{9}\)Recall that a GNN model is said to be equivariant if a permutation of the vertex set of the underlying graph yields a similar permutation of the output of the GNN model.
for all $1 \leq k \leq m$ and for any $(x_{i_1}, \ldots, x_{i_m}) \in C^{i_1} \times C^{i_2} \times \cdots \times C^{i_m}$.

The following proposition characterizes the equivariance of a general HOAN in terms of merge nodes.

**Proposition 5.3.** A network $\text{HOAN}_{G,W}$ as expressed in Definition 9 is permutation equivariant if and only if every merge node in $\text{HOAN}_{G,W}$ is permutation equivariant.

## 6 Experimental Results

HOANs can be used to construct different network architectures for diverse learning tasks. In this section, we demonstrate the domain generality and the efficacy of HOANs by evaluating their predictive performance on shape analysis and graph learning.

### 6.1 Shape Analysis: Mesh Classification and Segmentation

Table 1: Predictive accuracy on test sets related to shape analysis, namely on human body, COSEG (vase, chair, alien) and SHREC; red and blue colors indicate best and second best results, respectively.

| Method       | Segmentation tasks | Classification task |
|--------------|-------------------|---------------------|
|              | Human Body | Vase | Chair | Alien | SHREC |
| HodgeNet     | 85.03     | 90.30 | 95.68 | 96.03  | 99.10  |
| PD-MeshNet   | 85.61     | 95.36 | 97.23 | 98.18  | 99.70  |
| MeshCNN      | 85.39     | 92.36 | 92.99 | 96.26  | 98.60  |
| HOAN (ours)  | 87.30     | 93.40 | 98.30 | 93.70  | 99.17  |

We test HOANs on mesh segmentation using the Human Body [65] and the Shape COSEG (vase, chair, and alien) [99] datasets. For each mesh of these four datasets, the utilized CC structure is the one induced by the triangulation of the meshes, although other variations in the CC structure yield comparable results. Furthermore, three $k$-cochains are constructed for $0 \leq k \leq 2$ and are utilized in HOAN training. The architecture design, training parameters, train/test splits, and exact cochain constructions are provided in Appendix D. We also evaluate our method on mesh classification using the SHREC dataset [60] based on the same cochains and CC structure used in the segmentation experiments. As shown in Table 1, HOANs outperform three neural networks tailored to mesh analysis (HodgeNet, PD-MeshNet and MeshCNN) on two out of five datasets, and are among the best two networks for four out of five datasets.

### 6.2 Graph Classification

Table 2: Predictive accuracy on the test set of Bianchi graph classification; red and blue colors indicate best and second best results, respectively.

| Dataset | Method   | Graculus | NDP | DiffPool | Top-K | SAGPool | MinCutPool | HOAN (ours) |
|---------|----------|----------|-----|----------|-------|---------|------------|-------------|
| Easy    |          | 97.81    | 97.93 | 98.64    | 82.47 | 84.23   | 99.02      | 98.90       |
| Hard    |          | 69.08    | 72.67 | 69.98    | 42.80 | 37.71   | 73.80      | 75.79       |

For this task, we use the graph classification benchmark provided in [18]. The dataset consists of graphs with three different labels. For each graph, the feature vector on each node (the 0-cochain) is a one-hot vector of size 5 and it stores the relative position of the nodes on the graph. We also construct the 1-cochain and 2-cochain on the 2-clique complex of the graph by concatenating the nodes of one-hot vectors. Further details about this experiment are given in Appendix D. Table 2 reports the results on the easy and the hard versions of the dataset and compares them to six state-of-the-art graph neural networks. As shown in Table 2, HOANs outperform all other six networks on the hard dataset, and five of the other networks on the easy dataset. Our HOAN outperforms MinCutPool on the hard dataset, while it attains comparable performance to MinCutPool on the easy dataset.

5The difficulty in these data is controlled by the compactness degree of the graph clusters; clusters in the ‘easy’ data have more in-between cluster connections, while clusters in the ‘hard’ data are more isolated [18]
References

[1] P. Abramenko and K. S. Brown. Buildings: theory and applications, volume 248. Springer Science & Business Media, 2008.

[2] S. Abu-El-Haija, B. Perozzi, R. Al-Rfou, and A. A. Alemi. Watch your step: Learning node embeddings via graph attention. Advances in neural information processing systems, 31, 2018.

[3] V. Arvind, F. Fuhlbrück, J. Köbler, and O. Verbitsky. On weisfeiler-leman invariance: Subgraph counts and related graph properties. Journal of Computer and System Sciences, 113:42–59, 2020.

[4] D. Arya and M. Worrning. Exploiting relational information in social networks using geometric deep learning on hypergraphs. In Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval, pages 117–125, 2018.

[5] M. Aschbacher. Combinatorial cell complexes. In Progress in Algebraic Combinatorics, pages 1–80. Mathematical Society of Japan, 1996.

[6] N. A. Asif, Y. Sarker, R. K. Chakrabortty, M. J. Ryan, M. H. Ahamed, D. K. Saha, F. R. Badal, S. K. Das, M. F. Ali, S. I. Moyeen, et al. Graph neural network: A comprehensive review on non-euclidean space. IEEE Access, 2021.

[7] M. Atzmon, H. Maron, and Y. Lipman. Point convolutional neural networks by extension operators. arXiv preprint arXiv:1803.10091, 2018.

[8] L. Babai. Graph isomorphism in quasipolynomial time. In Proceedings of the forty-eighth annual ACM symposium on Theory of Computing, pages 684–697, 2016.

[9] J. Bai, B. Gong, Y. Zhao, F. Lei, C. Yan, and Y. Gao. Multi-scale representation learning on hypergraph for 3d shape retrieval and recognition. IEEE Transactions on Image Processing, 30:5327–5338, 2021.

[10] S. Bai, F. Zhang, and P. H. Torr. Hypergraph convolution and hypergraph attention. Pattern Recognition, 110:107637, 2021.

[11] S. Barbarossa and S. Sardellitti. Topological signal processing over simplicial complexes. IEEE Transactions on Signal Processing, 68:2992–3007, 2020.

[12] S. Barbarossa and S. Sardellitti. Topological signal processing: Making sense of data building on multiway relations. IEEE Signal Processing Magazine, 37(6):174–183, 2020.

[13] S. Barbarossa and M. Tsitsvero. An introduction to hypergraph signal processing. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6425–6429. IEEE, 2016.

[14] T. Basak. Combinatorial cell complexes and poincaré duality. Geometriae Dedicata, 147(1):357–387, 2010.

[15] P. W. Battaglia, R. Pascanu, M. Lai, D. Rezende, and K. Kavukcuoglu. Interaction networks for learning about objects, relations and physics. arXiv preprint arXiv:1612.00222, 2016.

[16] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al. Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261, 2018.

[17] F. Battiston, E. Amico, A. Barrat, G. Bianconi, G. Ferraz de Arruda, B. Franceschiello, I. Iacopini, S. Kéfi, V. Latora, Y. Moreno, et al. The physics of higher-order interactions in complex systems. Nature Physics, 17(10):1093–1098, 2021.

[18] F. M. Bianchi, C. Gallicchio, and A. Micheli. Pyramidal reservoir graph neural network. Neurocomputing, 470:389–404, 2022.

[19] C. Bick, E. Gross, H. A. Harrington, and M. T. Schaub. What are higher-order networks? arXiv preprint arXiv:2104.11329, 2021.
[20] C. Bodnar, F. Frasca, N. Otter, Y. Wang, P. Lio, G. F. Montufar, and M. Bronstein. Weisfeiler and lehman go cellular: CW networks. *Advances in Neural Information Processing Systems*, 34:2625–2640, 2021.

[21] G. Bouritsas, F. Frasca, S. Zafeiriou, and M. M. Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *arXiv preprint arXiv:2006.09252*, 2020.

[22] E. Bunch, Q. You, G. Fung, and V. Singh. Simplicial 2-complex convolutional neural nets. *NeurIPS workshop on Topological Data Analysis and Beyond*, 2020.

[23] W. Cao, Z. Yan, Z. He, and Z. He. A comprehensive survey on geometric deep learning. *IEEE Access*, 8:35929–35949, 2020.

[24] G. Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[25] S. Chaudhari, V. Mithal, G. Polatkan, and R. Ramanath. An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 12(5):1–32, 2021.

[26] Y. Chen, Y. R. Gel, and H. V. Poor. BScNets: Block simplicial complex neural networks. *arXiv preprint arXiv:2112.06826*, 2021.

[27] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. Sun. Gram: graph-based attention model for healthcare representation learning. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 787–795, 2017.

[28] K. Crane, F. De Goes, M. Desbrun, and P. Schröder. Digital geometry processing with discrete exterior calculus. In *ACM SIGGRAPH 2013 Courses*, pages 1–126. Association for Computing Machinery, 2013.

[29] E. Dai, C. Aggarwal, and S. Wang. Nrgnn: Learning a label noise resistant graph neural network on sparsely and noisily labeled graphs. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 227–236, 2021.

[30] M. Desbrun, E. Kanso, and Y. Tong. Discrete differential forms for computational modeling. In *Discrete differential geometry*, pages 287–324. Springer, 2008.

[31] S. Ebli, M. Defferrard, and G. Spreemann. Simplicial neural networks. *NeurIPS workshop on Topological Data Analysis and Beyond*, 2020.

[32] H. Edelsbrunner and J. Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

[33] P. Expert and G. Petri. *Higher-order description of brain function*, pages 401–415. Springer International Publishing, 2022.

[34] J. Feng, M. Huang, Y. Yang, and X. Zhu. GAKE: Graph aware knowledge embedding. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 641–651, 2016.

[35] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao. Hypergraph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3558–3565, 2019.

[36] M. Fey and J. E. Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[37] Y. Gao, Z. Zhang, H. Lin, X. Zhao, S. Du, and C. Zou. Hypergraph learning: Methods and practices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[38] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[39] L. Giusti, C. Battiloro, P. Di Lorenzo, S. Sardellitti, and S. Barbarossa. Simplicial attention networks. *arXiv preprint arXiv:2203.07485*, 2022.
[40] C. W. J. Goh, C. Bodnar, and P. Lio. Simplicial attention networks. In ICLR 2022 Workshop on Geometrical and Topological Representation Learning, 2022.

[41] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. Deep learning, volume 1. MIT press Cambridge, 2016.

[42] L. J. Grady and J. R. Polimeni. Discrete calculus: Applied analysis on graphs for computational science, volume 3. Springer, 2010.

[43] M. Hajij, K. Istvan, and G. Zamzmi. Cell complex neural networks. NeurIPS 2020 Workshop TDA and Beyond, 2020.

[44] M. Hajij, K. N. Ramamurthy, A. Saenz, and G. Zamzmi. High skip networks: A higher order generalization of skip connections. In ICLR 2022 Workshop on Geometrical and Topological Representation Learning, 2022.

[45] M. Hajij, G. Zamzmi, T. Papamarkou, V. Maroulas, and X. Cai. Simplicial complex representation learning. Machine Learning on Graphs (MLoG) Workshop at 15th ACM International WSD Conference, 2022.

[46] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or. MeshCNN: a network with an edge. ACM Transactions on Graphics (TOG), 38(4):1–12, 2019.

[47] J. Hansen and R. Ghrist. Toward a spectral theory of cellular sheaves. Journal of Applied and Computational Topology, 3(4):315–358, 2019.

[48] A. Hatcher. Algebraic topology. Cambridge University Press, 2005.

[49] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[50] J. Jiang, Y. Wei, Y. Feng, J. Cao, and Y. Gao. Dynamic hypergraph neural networks. In IJCAI, pages 2635–2641, 2019.

[51] X. Jing and J. Xu. Fast and effective protein model refinement using deep graph neural networks. Nature Computational Science, 1(7):462–469, 2021.

[52] A. D. Keros, V. Nanda, and K. Subr. Dist2cycle: A simplicial neural network for homology localization. arXiv preprint arXiv:2110.15182, 2021.

[53] E.-S. Kim, W. Y. Kang, K.-W. On, Y.-J. Heo, and B.-T. Zhang. Hypergraph attention networks for multimodal learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14581–14590, 2020.

[54] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016.

[55] R. Klette. Cell complexes through time. In Vision Geometry IX, volume 4117, pages 134–145. SPIE, 2000.

[56] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. nature, 521(7553):436–444, 2015.

[57] J. B. Lee, R. Rossi, and X. Kong. Graph classification using structural attention. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 1666–1674, 2018.

[58] J. B. Lee, R. A. Rossi, S. Kim, N. K. Ahmed, and E. Koh. Attention models in graphs: A survey. ACM Transactions on Knowledge Discovery from Data (TKDD), 13(6):1–25, 2019.

[59] Z. Li, H. Liu, Z. Zhang, T. Liu, and N. N. Xiong. Learning knowledge graph embedding with heterogeneous relation attention networks. IEEE Transactions on Neural Networks and Learning Systems, 2021.
[60] Z. Lian, A. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, G. Lavoua, P. D. Suetens, et al. Shape retrieval on non-rigid 3D watertight meshes. In *Eurographics workshop on 3d object retrieval (3DOR)*. Citeseer, 2011.

[61] L.-H. Lim. Hodge Laplacians on graphs. *Siam Review*, 62(3):685–715, 2020.

[62] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[63] A. Loukas. What graph neural networks cannot learn: depth vs width. *arXiv preprint arXiv:1907.03199*, 2019.

[64] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang. The expressive power of neural networks: A view from the width. In *Advances in neural information processing systems*, pages 6231–6239, 2017.

[65] H. Maron, M. Galun, N. Aigerman, M. Trope, N. Dym, E. Yumer, V. G. Kim, and Y. Lipman. Convolutional neural networks on surfaces via seamless toric covers. *ACM Trans. Graph.*, 36(4):71–1, 2017.

[66] H. Maron, H. Ben-Hamu, H. Serviansky, and Y. Lipman. Provably powerful graph networks. *arXiv preprint arXiv:1905.11136*, 2019.

[67] H. Maron, E. Fetaya, N. Segol, and Y. Lipman. On the universality of invariant networks. In *International conference on machine learning*, pages 4363–4371. PMLR, 2019.

[68] D. Mejia, O. Ruiz-Salgueiro, and C. A. Cadavid. Spectral-based mesh segmentation. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 11(3):503–514, 2017.

[69] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. *Advances in neural information processing systems*, 27, 2014.

[70] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe. Weisfeiler and Leman go neural: Higher-order graph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4602–4609, 2019.

[71] S. J. Orfanidis. *Introduction to signal processing*. Prentice-Hall, Inc., 1995.

[72] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, 2018.

[73] J. Payne. Deep hyperedges: a framework for transductive and inductive learning on hypergraphs. *arXiv preprint arXiv:1910.02633*, 2019.

[74] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein. On the expressive power of deep neural networks. In *international conference on machine learning*, pages 2847–2854. PMLR, 2017.

[75] M. Robinson. *Topological signal processing*, volume 81. Springer, 2014.

[76] T. M. Roddenberry and S. Segarra. HodgeNet: Graph neural networks for edge data. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pages 220–224. IEEE, 2019.

[77] T. M. Roddenberry, N. Glaze, and S. Segarra. Principled simplicial neural networks for trajectory prediction. In *International Conference on Machine Learning*, pages 9020–9029. PMLR, 2021.

[78] T. M. Roddenberry, M. T. Schaub, and M. Hajij. Signal processing on cell complexes. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022.

[79] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
[80] S. Sardellitti and S. Barbarossa. Robust signal processing over simplicial complexes. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 8857–8861. IEEE, 2022.

[81] S. Sardellitti and S. Barbarossa. Topological signal representation and processing over cell complexes. arXiv preprint arXiv:2201.08993, 2022.

[82] S. Sardellitti, S. Barbarossa, and L. Testa. Topological signal processing over cell complexes. Proceeding IEEE Asilomar Conference. Signals, Systems and Computers, 2021.

[83] R. Sato. A survey on the expressive power of graph neural networks. arXiv preprint arXiv:2003.04078, 2020.

[84] M. Savoy. Combinatorial cell complexes: Duality, reconstruction and causal cobordisms. arXiv preprint arXiv:2201.12846, 2022.

[85] M. T. Schaub and S. Segarra. Flow smoothing and denoising: Graph signal processing in the edge-space. In 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pages 735–739, 2018.

[86] M. T. Schaub, A. R. Benson, P. Horn, G. Lippner, and A. Jadbabaie. Random walks on simplicial complexes and the normalized hodge 1-laplacian. SIAM Review, 62(2):353–391, 2020.

[87] M. T. Schaub, J.-B. Seby, T. M. Roddenberry, and S. Segarra. Signal processing on higher-order networks: Livin’on the edge... and beyond. Signal Processing, 187:108149, 2021.

[88] M. T. Schaub, J.-B. Seby, F. Frantzen, T. M. Roddenberry, Y. Zhu, and S. Segarra. Signal processing on simplicial complexes. In Higher-Order Systems, pages 301–328. Springer, 2022.

[89] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Processing Magazine, 30(3):83–98, 2013.

[90] D. Smirnov and J. Solomon. HodgeNet: learning spectral geometry on triangle meshes. ACM Transactions on Graphics (TOG), 40(4):1–11, 2021.

[91] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu. RankClus: integrating clustering with ranking for heterogeneous information network analysis. In Proceedings of the 12th international conference on extending database technology: advances in database technology, pages 565–576, 2009.

[92] K. K. Thekumparampil, C. Wang, S. Oh, and L.-J. Li. Attention-based graph neural network for semi-supervised learning. arXiv preprint arXiv:1803.03735, 2018.

[93] N. Trask, A. Huang, and X. Hu. Enforcing exact physics in scientific machine learning: a data-driven exterior calculus on graphs. Journal of Computational Physics, page 110969, 2022.

[94] V. G. Turaev. Quantum invariants of knots and 3-manifolds, volume 18. Walter de Gruyter GmbH & Co KG, 2016.

[95] P. Veličković. Message passing all the way up. ICLR 2022 Workshop on Geometrical and Topological Representation Learning, 2022.

[96] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. In International Conference on Learning Representations, 2018.

[97] M. L. Wachs. Poset topology: tools and applications. arXiv preprint math/0602226, 2006.

[98] J. Wang, K. Ding, L. Hong, H. Liu, and J. Caverlee. Next-item recommendation with sequential hypergraphs. In Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, pages 1101–1110, 2020.
[99] Y. Wang, S. Asafi, O. Van Kaick, H. Zhang, D. Cohen-Or, and B. Chen. Active co-analysis of a set of shapes. *ACM Transactions on Graphics (TOG)*, 31(6):1–10, 2012.

[100] B. Weisfeiler and A. Leman. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series*, 2(9):12–16, 1968.

[101] H. Wu and M. K. Ng. Hypergraph convolution on nodes-hyperedges network for semi-supervised node classification. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(4):1–19, 2022.

[102] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 2020.

[103] Z. Xiao and Y. Deng. Graph embedding-based novel protein interaction prediction via higher-order graph convolutional network. *PloS one*, 15(9):e0238915, 2020.

[104] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[105] K. Xu, L. Wu, Z. Wang, Y. Feng, M. Witbrock, and V. Sheinin. Graph2seq: Graph to sequence learning with attention-based neural networks. *arXiv preprint arXiv:1804.00823*, 2018.

[106] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, and P. Talukdar. Hypergcn: A new method for training graph convolutional networks on hypergraphs. *Advances in neural information processing systems*, 32, 2019.

[107] M. Yang, E. Isufi, M. T. Schaub, and G. Leus. Finite impulse response filters for simplicial complexes. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pages 2005–2009. IEEE, 2021.

[108] M. Yang, E. Isufi, and G. Leus. Simplicial convolutional neural networks. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8847–8851. IEEE, 2022.

[109] S.-X. Zhang, X. Zhu, J.-B. Hou, C. Liu, C. Yang, H. Wang, and X.-C. Yin. Deep relational reasoning graph network for arbitrary shape text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9699–9708, 2020.

[110] H. Zhou, T. Young, M. Huang, H. Zhao, J. Xu, and X. Zhu. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, pages 4623–4629, 2018.
A Appendix: Higher-Order Domains

Roughly speaking, a higher-order domain is a generalization of a graph. It admits relations between its entities that are not necessarily binary, whereas a graph only admits binary relations among a non-empty set of abstract entities. There are many types of higher-order domains, including hypergraphs, simplicial complexes, and cell complexes. The main difference between various types of higher-order domains lies in the way higher-order relations are defined and related to each other (see Figure 1 for an illustration of different higher-order relations among various types of higher-order domains).

In this section, we give a brief review of the higher-order domains studied here - namely combinatorial cell complexes and regular cell complexes in Subsection A.2, cubical complexes in Subsection A.3, simplicial complexes in Subsection A.4, hypergraphs in Subsection A.5 - and show their relations to the combinatorial complex (CC) introduced in the main paper. Table 3 summarizes the main features for all complexes presented in this section.

Table 3: Tabular summary of the main features of different higher-order domains. Given a set \( S \) of abstract entities, a higher-order domain is a non-empty set \( X \) of the powerset of \( S \) that may satisfy other conditions. A relation is an element of \( X \). A particular higher-order domain is specified via its relations and via the way these relations are related to each other. Desirable features of these relations are indicated in the table’s rows. (i) A hierarchy of relations implies that relations of the higher-order domain have different ranking. (ii) Set-based relations are free from constraints among relations or their lengths. (iii) Relations among relations impose constraints among the higher-order relations of \( X \). (iv) Same rank \( \Rightarrow \) same cardinality indicates that relations with the same ranking in a given hierarchy on the higher-order domain do not have the same cardinality.

| Relation features                        | CC                     | CCC                   | Hypergraph  | Cubical complex | SC     | Graph |
|------------------------------------------|------------------------|-----------------------|-------------|----------------|--------|-------|
| Hierarchy of relations                   | ✓                      | ✓                     |             | ✓              | ✓      | ✓     |
| Set-based relations                      | ✓                      | ✓                     | ✓           |                |        |       |
| Relations among relations                | ✓                      | ✓                     | ✓           | ✓              | ✓      |       |
| Same rank \( \Rightarrow \) same cardinality | ✓                      | ✓                     | ✓           | ✓              | ✓      |       |

A.1 Combinatorial Complexes

The higher-order domains considered in this paper are combinatorial complexes (CCs). CCs are formally introduced in Definition 1.

A.2 Regular and Combinatorial Cell Complexes

Regular cell complexes share many of the properties that combinatorial complexes (CC) have. Indeed, many of the terms we define on a CC are typically defined in the context of cell complexes. Similar to CCs, regular cell complexes are generalized spaces that consist of simpler primitives called cells. Regular cell complexes admit combinatorial structures specified by a collection of (co)adjacency and incidence matrices. Their combinatorial structure allows us to naturally extend the general message passing schemes of deep learning protocols [43] into a unifying scheme that encompasses several state-of-the-art graph neural networks [36, 63, 16, 70, 15, 54]. While regular cell complexes admit a combinatorial structure specified by a set of incidence and/or (co)adjacency matrices, cell complexes are topological objects.

**Definition 11.** [48, 47] A regular cell complex \( \mathcal{X} \) is a topological space with a partition into subspaces (cells) \( \{ x_\alpha \}_{\alpha \in I} \) indexed by an index set \( I \), such that (a) \( \mathcal{X} = \bigcup_{\alpha \in I} \text{int}(x_\alpha) \), where \( \text{int}(x) \) refers to the interior of cell \( x \); (b) for all \( \alpha \in I \) there exists a homeomorphism \( \psi \) from \( c_\alpha \) to \( \mathbb{R}^{n_\alpha} \) for some \( n_\alpha \in \mathbb{N} \), called the dimension of the cell; (c) (regularity condition) for each \( x_\alpha \in \mathcal{X} \), the boundary \( \partial x_\alpha \) is a union of finitely many cells of dimension less than that of \( x_\alpha \).

Examples of regular cell complexes are given in Figure 8. The regularity condition in Definition 11 assures that each cell in the complex \( \mathcal{X} \) meets each 0-cell of \( \mathcal{X} \) at most once. Practically, it is desirable to write Definition 11 in purely combinatorial terms; this characterization can be realized by observing that cells in a regular cell complex have a poset structure [14]. Specifically, a combinatorial cell complex is a CC with a few additional properties as given in the following definition.
Definition 11. [48] The pair \((X, i)\) is called a combinatorial cell complex (CCC) if it satisfies the following conditions: (i) \((X, i)\) is a CC, (ii) for all \(x, y \in X\) with \(y \subseteq x\), there exists a cell \(y' \in X\) such that \(i(y') = i(y) + 1\) and \(y' \subseteq x\), (iii) (regularity condition) for all \(x \in X\), with \(rk(x) > 0\), \(x = \cup_{y \in \Delta(x)} y\), (iv) (diamond property) for all \(x, y \in X\), if \(x \not\subseteq y\) and \(i(x) = i(y) - 2\) then there are exactly two cells \(z_1, z_2\) with \(\nabla(x) \cap \Delta(y) = \{z_1, z_2\}\).

See also [55] for related constructions. Similar to Definition 11, the regularity condition in Definition 12 also assures that each cell in the complex \(X\) meets each 0-cell of \(X\) at most once. Further, the regularity condition shows that every cell \(x\) is determined via its face set \(\Delta(x)\). The diamond property is important to define Hodge-theory on CCC; the geometric intuition behind it is explained in Figure 9.

Orientation of cells in a CCC. While CCCs form a subclass of CCs, the neighborhood relations on CCCs are typically described via the signed incidence matrices \(B_k\), which we shall define momentarily, rather than with the incidence matrices \(B_{r,k}\) of CCs. The matrices \(B_k\) and \(B_{r,k}\) are related but they have different meanings. On one hand, \(B_k\) is the matrix induced by the so called boundary map \([48]\), a linear map that is typically defined on a CCC and completely characterizes the combinatorial structure of the CCC. To define boundary maps, one typically specifies orientations on the cells of a CCC. The orientation information is stored in the sign of non-zero elements of \(B_k\). On the other hand, the matrix \(B_{r,k}\) stores pure set-based incidence and is typically more common in the context of hypergraphs. Thus, it is common when considering cochain spaces and maps on a CCC to work with oriented cells.

An oriented \(k\)-cell \(x^k\) is an ordered tuple \(x^k = [x_1^{k-1}, \ldots, x_l^{k-1}]\), where two consecutive \((k-1)\)-cells share a common \((k-2)\)-cell. A cyclic permutation of the tuple \([x_1^{k-1}, \ldots, x_l^{k-1}]\) defines the same oriented cell. Two incident cells \(x^r\) and \(x^k\) with \(r < k\) are said to be consistent, and denoted by \(x^r \sim x^k\), if the orientations defined on \(x^k\) when restricted to \(x^r\) and on \(x^r\) are equivalent.

Signed incidence matrices. When \(X\) is a CCC, we need to augment the \(k\)-cochain space \(C_k(X)\) by one more aspect. Namely, if \(x^k = \sum a_i x_i^k \in C_k(X)\) for some \(a_i \in \mathbb{R}\), then an orientation change of one of the \(k\)-cells \(x_i^k\) in \(X^k\) is set to correspond to a change in the sign of the coefficient \(a_i\). The structure of a CCC \(X\) is determined by a sequence of matrices \(\{B_k\}_{k=0}^{\dim(X)}\), called the signed...
As Proposition A.1 proof is standard, we skip it and refer the interested reader to [14] for details.

(e.g., geometry processing [28]), it is desirable that adjacent cells intersect only in a single cell.

**Proposition A.1.** For any combinatorial cell complex, the incidence matrices \( \{B_k\}_{k=0}^N \) satisfy the fundamental property \( B_k B_{k+1} = 0 \) for all \( k \geq 1 \).

![Figure 10: Illustration of cochain maps on a complex \( \mathcal{X} \). The figure shows the incidence matrices \( B_k \), the \( k \)-combinatorial Hodge Laplacians \( L_k \) and their transpose counterparts \( B_k^T, L_k^T \).](image)

As Proposition’s proof is standard, we skip it and refer the interested reader to [14] for details.

**The \( k \)-Hodge-Laplacian operator of a CCC.** The signed incidence matrices are used to define the combinatorial \( k \)-Hodge Laplacian \( L_k : C^k(\mathcal{X}) \to C^{k-1}(\mathcal{X}) \) as \( L_k = L_k^\uparrow + L_k^{\downarrow} \), where \( L_k^{\downarrow} = B_k^T B_k \) and \( L_k^\uparrow = B_{k+1} B_k^T \). See Figure 10 for an illustration. In particular, \( L_0 \) is the graph Laplacian that describes node adjacency based on the edge adjacency of nodes. The operator \( L_0 \) measures the degree to which a given node signal deviates from its mean value in each 1-hop neighborhood. Thus, it characterizes a variety of phenomena such as heat diffusion and wave propagation. Similar to the graph Laplacian, the combinatorial Hodge-Laplacian is a positive semidefinite operator; it generalizes desirable properties of graph Laplacian signal processing [78, 11] and it measures the diffusion dynamics on the substructures of the complex \( \mathcal{X} \) beyond the node level [61]. The non-diagonal entries of \( L_i^{\uparrow} \) and \( L_i^{\downarrow} \) represent the signed higher-order adjacency/co-adjacency between \( k \)-cells in the complex \( \mathcal{X} \), which are denoted by \( A_k \) and \( coA_k \), respectively.

**Computational example of a CCC and its matrices.** In this example, we demonstrate how the combinatorial Hodge-Laplacian operator is built for the complex shown in Figure 11. We first order the cells in this complex using lexicographic ordering. The final signed incidence matrices \( B_1 \) and \( B_2 \) are given in Figure 11. Observe that, due to the regularity condition imposed on a general CCC, the entries of the incidence matrices are in \( \{0, \pm 1\} \). The figure also shows the matrices \( L_i^{\uparrow} \), \( 0 \leq i \leq 1 \), \( L_i^{\downarrow} \), \( 1 \leq i \leq 2 \) along with the corresponding (co)adjacency matrices.

### A.3 Cubical Complexes

In a general regular cell complex, adjacent cells may have in their intersection set more than one lower dimensional incident cells. For instance, we can see in the second figure from the left in Figure 9 that the two pink 1-cells around the meridian of the torus intersect in two vertices. In many applications (e.g., geometry processing [28]), it is desirable that adjacent cells intersect only in a single cell.

**Definition 13.** A cubical complex is a CCC \( \mathcal{X} \) such that for any two cells \( x, y \in \mathcal{X} \), their intersection satisfies \( x \cap y \in \mathcal{X} \cup \{\emptyset\} \).

Figure 8(second complex from left) and Figure 12(complex on the right-hand side) provide illustrative examples of cubical complexes.

### A.4 Simplicial Complexes

Simplicial complexes (SCs) form an important subclass of cell complexes. The simple and uniform combinatorial structure of SCs makes many of the constructions and theory typically defined on
Figure 11: Example of a CCC of dimension 2, with its incidence matrices $B_1$ and $B_2$, as well as the Hodge-Laplacian matrices $L_{\text{up}}$, $L_{\text{down}}$, $L_{\text{up}}^\text{co}$, and $L_{\text{down}}^\text{co}$.

Figure 12: Left: an example of a CC that is not a cubical complex. Right: an example of a cubical complex. Note that the intersection of the two blue faces (2-cells) in the right figure is a single pink edge (1-cell).

graphs extendable to SCs. Triangular meshes, which constitute an important data type with many applications in computer graphics, are special cases of SCs. We refer the reader to [ 87, 28] for a relevant introduction on SCs.

Definition 14. A simplicial complex (SC) on a non-empty set $S$ is a subset $X$ of $\mathcal{P}(S) \setminus \{\emptyset\}$ such that $x \in X$ and $y \subset x$ imply $y \in X$.

It is easy to verify that a SC $X$ is a CC with rank function that satisfies $i(\{x\}) = |x| - 1$, $x \in X$, where $|x|$ is the cardinality of set $x$.

A.5 Hypergraphs

Similar to graphs, simplicial, and cell complexes, hypergraphs model abstract relations among elements of a set of elements. Unlike graphs, simplicial and cell complexes, hypergraphs model relations with arbitrary length and with no restrictions among the relations.

Definition 15. Let $V$ be a non-empty set. A hypergraph on $V$ is a pair $H = (V, E)$, where $E$ is a set of non-empty subsets of the powerset of $V$, which are called hyperedges or edges.
Hypergraphs can also be seen to be a special case of CCs. Specifically, hyperedges in a hypergraph correspond to cells in a CC, where the rank of the higher order cells is ignored.

B Appendix: Learning Discrete Exterior Calculus Operators

The product $A_{tr} := A \circ att$ presented in Equations (5) and (6) has an interesting and important interpretation. First, recall that $A_{tr}$ has the same shape as the original operator $A$. More importantly, the operator $A_{tr}$ can be thought of as a learned version of the operator $A$. For instance, if $A$ is the $k$-Hodge Laplacian $L_k$, then the learned attention version $A_{tr}$ of it represents a $k$-Hodge Laplacian that is adapted to the domain $X'$ for the learning task at hand. This perspective allows to view our attention framework as a framework to learn discrete exterior calculus (DEC) operators [30]. The works of [90, 93] are along these lines, where neural networks are used to learn the Laplacian operator for a given shape analysis task. Concretely, one of the main building blocks of discrete exterior calculus is a collection of linear operators of the form

$$A : C^i(X) \to C^j(X')$$

that act on a cochain $x$ to produce another cochain $A(x)$. An example of an operator $A$ is the graph Laplacian. There are seven primitive discrete exterior calculus operators, including the discrete exterior derivative, the hodge star and the wedge product. The seven primitive operators can be combined together to form other operators. In our setting, the discrete exterior derivatives are precisely the signed incidence matrices defined in Appendix A.2. In other words, it is common in the discrete exterior calculus setting [30] to refer to $B^i_k$ as the $k$-th discrete exterior derivative $d^k$. The matrices $B^i_0$, $B^i_1$ and $B^i_2$ have important discrete calculus interpretation; in particular, $d^0(x)$, $d^1(x)$, and $d^2(x)$ of some 0, 1 and 2 cochains defined on $X'$ are the discrete analogs of the gradient $\nabla x$, curl $\nabla \times x$ and divergence $\nabla \cdot x$ of a smooth function defined on a smooth surface. We refer the reader to [30] for a coherent list of discrete exterior calculus operators and their interpretation. Together, cochains and the operators that act on them, provide a concrete framework that facilitates computing a cochain of interest, such as a cochain obtained by solving a partial differential equation on a discrete surface. Our attention framework can be viewed as a non-linear version of the discrete exterior calculus based on linear operators $A$ and can be used to learn the discrete exterior calculus operators on the domain $X'$ for a particular learning task. More specifically, Equation (12), in the classical discrete exterior calculus setting, can be considered as a special case of Equations (3) and (6). Note that unlike existing work [90, 93], our operator learning setting facilitated via HOANs is general and applicable on all domains where DEC is typically applicable such as triangular and polygonal meshes [28]. Moreover, existing operator learning methods cannot be used to learn all DEC operators and are only defined for particular types. Unlike existing related approaches [90, 93], our setting determined with the two HOANs blocks can be used to learn any DEC operator.

C Appendix: Lifting Maps

This section shows how we can lift higher-order domains to CCs or CCCs. Such lifting is useful as it allows HOANs to be applied to common data types, including graphs and cell/simplicial complexes. We note that this section only scratches the surface, and there remains a large space of lifting constructions to be explored and analysed.

C.1 The n-hop CC of a Graph

Let $G = (V(G), E(G))$ be a graph and $n \geq 2$ an integer. The $n$-hop CC of $G$, denoted by $CC_{n-hop}(G)$, is a CC structure associated with $G$ and it is defined as follows. First, declare the 0-cells and the 1-cells in $CC_{n-hop}(G)$ to be the nodes and the edges of $G$, respectively. Then, declare the $n$-cells in $CC_{n-hop}(G)$ to be the set nodes in $n$-hop neighborhoods of the nodes in $G$. It is easy to verify that $CC_{n-hop}(G)$ is a CC of dimension $n$.

C.2 The Path-Based and Subgraph-Based CC of a Graph

Let $G = (V(G), E(G))$ be a graph. A natural CC structure defined on $G$ considers the paths of $G$. We call this CC a path-based CC and denote it by $CC_p(G)$. The CC structure of $CC_p(G)$ is defined
as follows. First, declare the \( X^0 \) and \( X^1 \) in \( CCC_p(G) \) to be the nodes and the edges of \( G \), respectively. We now explain how to construct a 2-cell in \( CCC_p(G) \). Let \( P \) be a path in \( G \) with length larger than or equal to 2 (i.e., with two or more edges). An element \( x_P \) in \( X^2 \) induced by \( P \) is defined to be \( x_P = \cup_{v \in P} \{ v \} \). The final set \( X^2 \) in \( CCC_p(G) \) is simply a non-empty set of elements of the form \( x_P \). It is easy to verify that \( CCC_p(G) \) has a CC structure with \( \text{dim}(CCC_p(G)) = 2 \). Note that we may replace the path \( P \) by a tree/subgraph \( G' \) of graph \( G \) and obtain a similar CC structure induced by the tree/subgraph of \( G \).

### C.3 The Loop-Based CCC of a Graph

Let \( G = (V(G), E(G)) \) be a graph. We associate a CCC structure to \( G \) that considers the loops in \( G \). We denote this structure \( CCC_{loop}(G) \) and we define it as follows. Start by declaring \( X^0 \) and \( X^1 \) in \( CCC_{loop}(G) \) to be the nodes and the edges of \( G \), respectively. A 2-cell is a set \( C = \{ x^0_1, \ldots, x^0_k \} \subset X^0 \), such that \( \{ x^0_i, x^0_{i+1} \}, 1 \leq i \leq k - 1 \) and \( \{ x^0_k, x^0_1 \} \) are the only edges of \( X^1 \cap C \). The final set \( X^2 \) in \( CCC_{loop}(G) \) is simply a non-empty set of elements of the form \( C \). It is easy to verify that \( CCC_{loop}(G) \) has a CCC structure with \( \text{dim}(CCC_{loop}(G)) = 2 \). Note that the sequence \( [x^0_1, \ldots, x^0_k] \) defines a loop in graph \( G \). This loop is called the loop that characterizes 2-cell \( C \). Similar constructions are suggested in \([5][14][84][78]\).

### C.4 The Coface CC of a SC/CCC

Here, we describe a method to lift a SC of dimension 2 to a CC of dimension 3. The method can be easily generalized to other dimensions. For a SC \( Y \) of dimension 2, the coface CC of \( Y \), denoted by \( CCC_{SC}(Y) \), is defined as follows. Start by declaring \( Y^0, Y^1 \) and \( Y^2 \) in \( CCC_{SC}(Y) \) to be the nodes, the edges, and the triangles in \( Y \), respectively. We now explain how to construct a 3-cell in \( CCC_{SC}(Y) \). Let \( x^3 \) be a 2-cell in \( Y \). The 3-cell in \( CCC_{SC}(Y) \) associated with \( x^3 \) is the union of all 0-cells in \( N_{sc}(x^3) \setminus x^3 \). \( Y^3 \) in \( CCC_{SC}(Y) \) is defined to be the set of all 3-cells in associated with all 2-cells in \( Y \) as defined above. It is easy to verify that \( CCC_{SC}(Y) \) has a CC structure with \( \text{dim}(CCC_{SC}(Y)) = 3 \). A similar construction can be defined to lift a given CCC to its coface CC.

### D Appendix: Experiment Details

Here, we provide additional details for the experiments in Section 6 of the paper.

#### D.1 Experimental Details of Mesh Analysis

##### D.1.1 CC Structure

The CC structure used for this experiment is the obvious one induced by the triangulation of the meshes. Specifically, the 0, 1 and 2 cells are the nodes, edges, and faces of the mesh, respectively. The matrices used for the HOANs are \( B_{0,1}, B_{0,2} \), their transpose matrices, and the (co)adjacency matrices \( |A_{1}|, |coA_{1}| \) and \( |coA_{2}| \).

##### D.1.2 Building Input Mesh Cohains for Shape Analysis

A HOAN takes a vector of cochains as input features. For shape analysis tasks, we consider simple choices of cochains, whose features are built directly from the vertex coordinates of the underlying mesh. We note that other choices (e.g., spectral-based cochains \([68]\)) can also be included. Our shape analysis learning tasks have three input cochains: the node, edge and face cochains. Each node cochain has two input features: the position and the normal associated with the node. Similar to \([56]\), each edge cochain consists of 5 features: the length of the edge, the dihedral angle (Figure 13), two inner angles, and two edge-length ratios for each face. Finally, each input face cochain consists of three input features: the face area, face normal, and the three face angles.

##### D.1.3 Segmentation Network Architecture

For the Human Body dataset \([65]\), we built a HOAN that produces an edge class. The tensor diagram of the architecture is shown in Figure 4(a). For the COSEG dataset \([99]\), we built a HOAN that
combines our proposed feature vectors defined on nodes, edges, and faces to learn the final face class. The architecture uses the incidence matrices as well as the (co)adjacency matrices to construct a signal flow as demonstrated in Figure 14(b). Specifically, the tensor diagram displays three non-squared attention-blocks and three squared attention blocks. The depth of the model is chosen to be $2$, as indicated in Figure 14(b).

Figure 14: The tensor diagrams of the two HOANs used in the mesh segmentation tasks. The orange, pink and blue colors correspond to the 0, 1, and 2-cochain spaces in each input triangulated mesh. The greyed out nodes represent parts of the non-squared attention block that is not used during the computation. (a) The HOAN tensor diagram used in the Human Body segmentation dataset. (b) The HOAN tensor diagram for the COSEG segmentation dataset.

Note that the architectures chosen for the COSEG and for the Human Body datasets have the same number and types of building blocks; compare Figures 14(a) and 14(b). For both of these architectures, a softmax activation is applied on the output tensor. All our segmentation models are trained for 600 epochs and optimized using the standard cross entropy loss.

**D.1.4 Mesh Classification**

The HOAN architecture for mesh classification is demonstrated in Figure 15(a). The final layer of this HOAN, depicted as a grey node in Figure 15(a), is a simple pooling operation that sums all embeddings of the CC after mapping them to the same Euclidean space. The HOAN classification network is trained with both tanh and identity activation functions; we use anisotropic scaling and random rotations for data augmentation. Each mesh is centered around the vertex center of the mass and rescaled to fit inside the unit cube. The HOAN with identity activations and the HOAN with tanh activations achieved predictive accuracies of $96.67\%$ and $99.17\%$, respectively. It is worth mentioning that the mesh classification HOAN requires a significantly lower number of epochs to train (40 epochs) as compared to the mesh segmentation HOANs. It is trained using the standard cross entropy loss.
D.2 Experimental Details of Graph Analysis

D.2.1 CC Structure

We use the 2-clique complex of the input graph. The matrices used for the HOAN are $B_{0,1}$, $B_{0,2}$, $B_{0,2}$, their transpose matrices, and the (co)adjacency matrices $|A_1|$, $|coA_2|$.

D.2.2 Building Input Graph Cochains for Graph Analysis

The cochains used to build the graph classification HOAN are constructed as follows. For each graph in the dataset, we set the 0-cochain to be the one-hot vector of size 5 provided with the dataset [18]. This one-hot vector stores the relative position of the nodes on the graph. We also construct the 1-cochain and 2-cochain on the 2-clique complex of the graph by concatenating the nodes of one-hot vectors.

D.2.3 Graph Classification Network Architecture

The exact HOAN architecture for graph classification is depicted in detail in Figure 15(b). The grey node in Figure 15(b) indicates a pooling operation.

D.3 Additional Details

D.3.1 Training Details

All networks in this work are trained using the Adam optimizer [62] with a learning rate of 0.0001, a batch size of 1, and standard backpropagation. We have implemented our pipeline in PyTorch, and run our experiments using a Microsoft Windows back-end on a single GPU.

D.3.2 Human Segmentation Dataset

The original human segmentation dataset presented in [7] contains relatively large meshes with size up to 12,000 vertices. The segmentation labels provided in this dataset are set per-face; the segmentation accuracy is defined to be the ratio of the correctly classified faces over the total number of faces in the entire dataset. We use a simplified version of the original dataset, as provided by [46], in which meshes have less than 1,000 nodes and segmentation labels are remapped to edges.
E Appendix: Equivariance of HOANs

The height of a tensor diagram is the longest path from any source node to any target node. For instance, the height of the tensor diagram given in Figure 1 is 3. The following proposition allows us to express tensor diagrams of height 1 in terms of merge nodes.

**Proposition E.1.** Let \( \text{HOAN}_{G,W} : C^{i_1} \times C^{i_2} \times \ldots \times C^{i_m} \to C^{j_1} \times C^{j_2} \times \ldots \times C^{j_n} \) be a HOAN with a tensor diagram of height 1. Then

\[
\text{HOAN}_{G,W} = (M_{G_{i_1},W_1}, \ldots, M_{G_{j_n},W_n}),
\]

where \( G_k \subseteq G \).

Proposition E.1 simply states that every target node \( j_k \) in a tensor diagram of height 1 is a merge node specified by the operators \( G_k \) formed by the labels of the edges with target \( j_k \). In general, working with Definition 10 may be cumbersome. It is easier to characterize the equivariance in terms of merge nodes.

**Permutation equivariance.** Definitions 2 and 10 introduce permutation equivariance on HOANs. The following definition is a special case of Definition 10.

**Definition 16.** Let \( \mathcal{X} \) be a CC and let \( G = \{G_k\} \) be a finite sequence of cochain operators defined on \( \mathcal{X} \) with \( G_k : C^{i_k}(\mathcal{X}) \to C^{j_k}(\mathcal{X}) \). Let \( P = \{P_i\}_{i=0}^{\dim(\mathcal{X})} \) be a collection of permutation matrices with \( P_i \in \{0, 1\}^{|\mathcal{X}||\times|\mathcal{X}||} \), We say that the merge node given in Equation (10) is permutation equivariant if

\[
M_{G,W}(x_{i_1}, \ldots, x_{i_m}) = P_j M_{P_j(G),W}(P_i x_{i_1}, \ldots, P_i x_{i_m})
\]

for any \((x_{i_1}, \ldots, x_{i_m}) \in C^{i_1} \times C^{i_2} \times \ldots \times C^{i_m}\).

**Proposition E.2.** Let \( \text{HOAN}_{G,W} : C^{i_1} \times C^{i_2} \times \ldots \times C^{i_m} \to C^{j_1} \times C^{j_2} \times \ldots \times C^{j_n} \) be a HOAN with a tensor diagram of height 1. Then \( \text{HOAN}_{G,W} \) is orientation/permutation equivariant if and only the merge nodes \( M_{G_{j_k},W_k} \) given in Equation (13) are orientation/permutation equivariant for \( 1 \leq k \leq n \).

**Orientation equivariance.** Analogous to Definition 9, we have the following defined on a CCC.

**Definition 17.** Let \( \mathcal{X} \) be a CCC. Let \( G = \{G_k\} \) be a sequence of cochain operators defined on \( \mathcal{X} \) with \( G_k : C^{i_k} \to C^{j_k} \). Let \( D = \{D_i\}_{i=0}^{\dim(\mathcal{X})} \) be a collection of diagonal matrices with diagonals \( \pm 1 \) of size \( D_k \in \{0, 1\}^{|\mathcal{X}||\times|\mathcal{X}||} \) such that \( D_0 = I \). We define \( D(G) \) to be the sequence \( \{D_{j_k}G_kD_{i_k}\} \).

The orientation equivariance is defined similar to the permutation equivariance.

**Definition 18.** Let \( \mathcal{X} \) be a CCC and let \( G = \{G_k\} \) be a finite sequence of cochain operators defined on \( \mathcal{X} \). Let \( D = \{D_i\}_{i=0}^{\dim(\mathcal{X})} \) be a collection of permutation matrices with \( D_i \in \{0, 1\}^{|\mathcal{X}||\times|\mathcal{X}||} \). A HOAN of the form

\[
\text{HOAN}_{G,W} : C^{i_1} \times C^{i_2} \times \ldots \times C^{i_m} \to C^{j_1} \times C^{j_2} \times \ldots \times C^{j_n}
\]

is orientation equivariant if

\[
\text{Proj}_k \circ \text{HOAN}_{G,W}(x_{i_1}, \ldots, x_{i_m}) = D_k \text{Proj}_k \circ \text{HOAN}_{D(G),W}((D_1 x_{i_1}, \ldots, D_k x_{i_m}))
\]

for all \( 1 \leq k \leq m \) and for any \((x_{i_1}, \ldots, x_{i_m}) \in C^{i_1} \times C^{i_2} \times \ldots \times C^{i_m}\).

**Characterization of equivariance in terms of merge nodes.** Propositions E.1 and E.2 can be stated for the orientation equivariance case. We skip stating these facts here and only state the main proposition that characterizes the orientation equivariance in terms of merge nodes.

**Proposition E.3.** A network \( \text{HOAN}_{G,W} \) of the form

\[
\text{HOAN}_{G,W} : C^{i_1} \times C^{i_2} \times \ldots \times C^{i_m} \to C^{j_1} \times C^{j_2} \times \ldots \times C^{j_n}
\]

is orientation equivariant if and only if every merge node in \( \text{HOAN}_{G,W} \) is orientation equivariant.
Appendix: The Expressive Power of HOANs

Metrics of expressivity are used for characterizing the ability of neural networks to approximate functions \[74, 64\]. Such expressivity metrics may also incorporate aspects of discrete domains upon which we conduct our deep learning calculations \[83\]. For instance, in the context of GNNs, the Weisfeiler and Lehman test (WL test) \[100, 104, 67\] has recently gained popularity as a metric to study the expressive power of GNNs. The WL test is a graph-recolouring procedure that provides a coloring of a graph’s nodes. This coloring gives a necessary condition for two graphs to be isomorphic. Within this context, and due to their similarity to the way the WL updates colors on graphs, GNNs with their vanilla message passing scheme \[38\] have been shown to be as powerful as the WL test; hence, not being able to solve the graph isomorphism problem \[104, 66, 3, 21\]. This has motivated the definition of higher order WL tests (k-WL tests) \[8\], which take into account non-local information. Higher order WL tests are able to identify more complex structures in the underlying graph. This line of research emphasizes the importance of incorporating higher order abstractions in the design of neural network models on discrete domains. Using the higher-order message passing defined on cell complexes \[43\], the work of \[20\] extended the WL test to cell complexes.

Inspired by WL tests on graphs, we define a coloring procedure for CCs. To this end, let \(\mathcal{X}\) be a CC and let \(G = \{G_1, ..., G_n\}\) be a collection of neighborhood matrices on \(\mathcal{X}\) as defined in Section 5. The \(G\)-CCWL test is outlined as follows:

1. Start coloring each cell \(x \in \mathcal{X}\) in a given CC by a single color \(C_x, 0 = 1\).
2. At iteration \(t\), each cell \(x\) aggregates the colors \(C_{y, t-1}\) of all its neighbours \(y \in \mathcal{N}_G(x)\) and represents them as a multiset. Note that the neighbourhood of a cell is a function of our choice.
3. For each cell \(x\), hash the current multiset into a unique new color and set this multiset as the new color of \(x\).
4. If a stable coloring is reached, then stop, otherwise go to step 2.

The above procedure provides a CC version of the WL test, denoted by CCWL. Similar to the regular WL test, the CCWL tests give a necessary condition for two CCs to be isomorphic. Further, the relation between the CCWL test and HOANs is now clear. Specifically, one may think of any message passing scheme defined on a CC induced by the neighborhood matrices \(G = \{G_1, ..., G_n\}\) as a coloring procedure. Perhaps it is easiest to think of the CCWL test in terms of the augmented Hasse graph of the underlying CC. Namely, the \(G\)-CCWL test is equivalent to a coloring procedure defined on the augmented Hasse graph \(\mathcal{H}_{\mathcal{X}}(G)\).

Appendix: Combinatorial Properties of Combinatorial Complexes

Combinatorial complexes and the morphisms defined among them (Definitions 1 and 2) induce a new category of combinatorial objects. The relation between CCs and less general objects, such as (hyper)graphs and other complexes, makes CCs worthy of further investigation. For instance, the WL-test can be extended easily to this category as shown in Section F; hence, the WL-test can be used as an invariant for CCs. Beyond the application of such invariants in the expressiveness of neural networks, the classification problem of combinatorial objects in this new category and the study of other invariants defined on them is an interesting research direction.

Appendix: Proofs

Proof of Proposition 5.1. The proof of the three parts of the proposition follows by noting that the structure of a CC is determined completely by its Hasse graph representation. For part (i), edges in the Hasse graph are precisely the non-zero entries of the matrices \(\{B_{k, k+1}\}_{k=0}^{\dim(\mathcal{X})-1}\). Part (ii) follows by observing that two \((k - 1)\) cells \(x^{k-1}\) and \(y^{k-1}\) are adjacent if and only if there exists a \(k\)-cell \(z^k\) that is incident to \(x^{k-1}\) and \(y^{k-1}\). Finally, for part (iii), two \((k + 1)\) cells \(x^{k+1}\) and \(y^{k+1}\) are adjacent if and only if there exists a \(k\)-cell \(z^k\) that is incident to \(x^{k+1}\) and \(y^{k+1}\). The proofs of parts (ii) and (iii) are also visually supported by Figure 16.
Figure 16: Relation between immediate incidence and (co)adjacency on the Hasse graph of a CC. (a) Two \((k - 1)\) cells \(x^{k-1}\) and \(y^{k-1}\) (orange nodes) are adjacent if and only if there exists a \(k\)-cell \(z^k\) (pink node) that is incident to \(x^{k-1}\) and \(y^{k-1}\). (b) Two \((k + 1)\) cells \(x^{k+1}\) and \(y^{k+1}\) (blue nodes) are adjacent if and only if there exists a \(k\)-cell \(z^k\) (pink node) that is incident to \(x^{k+1}\) and \(y^{k+1}\).

**Proof of Proposition 5.2.** Let \(X\) be a CC. Let \(G = \{G_1, \ldots, G_n\}\) be a sequence of cochain operators defined on \(X\). Let \(\mathcal{H}_X(G)\) be the augmented Hasse graph determined by \(G\). By definition of the augmented Hasse graph, there is a one-to-one correspondence between the nodes of \(\mathcal{H}_X(G)\) and the cells in \(X\). For every cell \(x \in X\), we denote its corresponding node in \(\mathcal{H}_X(G)\) by \(x'\). Let \(m_{x,y}\) be a message (computation) that is executed between two cells \(x\) and \(y\) on \(X\) as a part of the computation of function \(M_{G,W}\). Hence, the two cells \(x\) and \(y\) must have a corresponding non-zero entry in one of the matrices \(G\). By the definition of the augmented Hasse graph, this non-zero entry corresponds to an edge in \(\mathcal{H}_X(G)\) between \(x'\) and \(y'\). Thus, the message \(m_{x,y}\) between cells \(x\) and \(y\) can be computed on the Hasse graph as the message \(m_{x',y'}\) between the corresponding nodes \(x'\) and \(y'\). □

**Proof of Proposition E.1.** Let \(\text{HOAN}_{G,W} : C^{i_1} \times C^{i_2} \times \ldots \times C^{i_m} \rightarrow C^{j_1} \times C^{j_2} \times \ldots \times C^{j_n}\) be a HOAN with a tensor diagram of height 1. Since the codomain of the function \(\text{HOAN}_{G,W}\) is \(C^{j_1} \times C^{j_2} \times \ldots \times C^{j_n}\), then \(\text{HOAN}_{G,W}\) is determined by \(n\) functions \(F_k : C^{i_1} \times C^{i_2} \times \ldots \times C^{i_m} \rightarrow C^{j_k}\) for \(1 \leq k \leq n\). Since the height of the tensor diagram of \(\text{HOAN}_{G,W}\) is 1, then each function \(F_k\) is also of height 1 and it is thus a merge node by definition. The result follows. □

**Proof of Proposition E.2.** If a HOAN is of height 1, then by Proposition E.1

\[\text{Proj}_k \circ \text{HOAN}_{G,W}(x_{i_1}, \ldots, x_{i_m}) = M_{G,j_k,W_k}\]

Hence, the result follows from the definition of merge node permutation equivalence (Definition 16) and the definition of permutation equivalence on HOAN (Definition 9). □

**Proof of Proposition E.3.** Proposition E.2 proves this fact for HOANs of height 1. For HOANs of height \(n\), it is enough to observe that a HOAN of height \(n\) is a composition of \(n\) HOANs of height 1 and the fact that composition of two permutation equivariant networks is a computationally equivariant network. □

**Proof of Proposition E.3.** The proof is similar to the proof of Proposition 5.3. □

I Appendix: Glossary

Table 4 summarizes the paper’s main notations and their definitions.
Table 4: The notations in this work are summarized below.

| Symbol | Description |
|--------|-------------|
| $S$    | Non-empty finite set of abstract entities |
| $\mathcal{P}(S)$ | Power set of a set $S$ |
| $CC$   | Combinatorial complex |
| $CCC$  | Combinatorial cell complex |
| $N_a(x)$ | Adjacency set of a cell $x$ |
| $N_{co}(x)$ | Coadjacency set of a cell $x$ |
| $N_{G_1,...,G_n}(x)$ | The neighbors of $x$ in a complex $\mathcal{X}$ specified by the neighborhood matrices $\{G_1,...,G_n\}$ |
| $\Delta(x)$ | Set of faces of a cell $x$ |
| $\nabla(x)$ | Set of cofaces of a cell $x$ |
| $Z^+$ | Set of non-negative integers |

**Complex and cell notations**

- $\mathcal{X}$: A complex
- $x^k$: A cell $x$ of rank $k$
- $\dim(\mathcal{X})$: Dimension of the combinatorial complex $\mathcal{X}$
- $\{c_{\alpha}\}_{\alpha \in I}$: A partition into subspaces (cells) indexed by an index set $I$
- $\text{int}(x)$: The interior of cell $x$ in regular cell complex
- $n_\alpha \in \mathbb{N}$: The dimension of a cell in regular cell complex
- 0-cells: The nodes or vertices of the CC
- 1-cells: The edges of the CC
- $k$-cells: Cells with rank $k$
- $|\mathcal{X}|$: The cardinality or number of cells
- $CC_{r,hop}(G)$: The $r$-hop CC of a graph $G$, which is a CC structure induced by the edges of $G$
- $CC_p(G)$: A natural CC structure defined on a graph $G$
- $CC_{\text{loop}}(G)$: The CCC structure associated to a graph $G$
- $CC_{\text{hop}}(G)$: CCC structure associated to a graph $G$
- $Y$: The coface CC associated with a simplicial/cell complex $Y$

**Higher-order notations**

- $W$: Trainable parameter
- $C^k(\mathcal{X}, \mathbb{R}^d)$: The $k$-cochain space with features in $\mathbb{R}^d$
- $C^k(\mathcal{X}, \mathbb{R})$: The $k$-cochain space with features in $\mathbb{R}$
- $C^k$: The $k$-cochain space with features in some Euclidean space
- $\mathcal{G} = \{G_1, \ldots, G_m\}$: A set of cochain maps $G_i$, defined on defined on $\mathcal{X}$
- $\mathcal{M}_{G_i} : C^i \times \cdots \times C^{i_m} \rightarrow C^j$: A merge node
- $\mathcal{A} : C^i(\mathcal{X}) \rightarrow C^i(\mathcal{X})$: A cochain map
- $\mathcal{V}(x_1, \ldots, x_m)$: Vector of cochains
- $\mathcal{H}^k : C^k(\mathcal{X}) \rightarrow C^k(\mathcal{X})$: Higher-order attention matrix
- $\mathcal{X}^k_{\text{core}} = \{X_1, \ldots, X_m\}$: Set of a complex object in the vicinity of $X_0$
- $f : X_0 \times X_{\text{core}} \rightarrow [0, 1]$: Higher-order attention function
- $\text{HOAN}_{G,W} : \mathcal{X} \rightarrow \mathcal{Y}$: The G-based WL test on CC.

**Matrix notations**

- $B_{r,k} : \mathcal{X}^r \times \mathcal{X}^k \rightarrow \mathbb{R}$: Incidence matrices between two skeletons
- $B_k : \mathcal{X}^k \rightarrow \mathcal{X}^{k-1}$: Signed incidence matrices between two skeletons
- $L_k : \mathcal{X}^k \rightarrow \mathcal{X}^{k-1}$: The $k$-Hodge Laplacian
- $L_k : \mathcal{X}^k \rightarrow \mathcal{X}^{k}$: The $k$-Hodge up Laplacian
- $L_{\text{down}} : \mathcal{X}^k \rightarrow \mathcal{X}^{k-1}$: The $k$-Hodge down Laplacian
- $A_k : \mathcal{X}^k \rightarrow \mathcal{X}^{k}$: The $k$-signed adjacency matrix
- $\text{coA}_k : \mathcal{X}^k \rightarrow \mathcal{X}^{k}$: The $k$-coadjacency matrix
- $|A_k| : \mathcal{X}^k \rightarrow \mathcal{X}^{k}$: The $k$-adjacency matrix
- $\text{coA}_k : \mathcal{X}^k \rightarrow \mathcal{X}^{k}$: The $k$-coadjacency matrix
- $(B_{k-1,k})_{k=0}^{\dim(\mathcal{X})}$: Signed incidence matrices of a CCC
- $(B_{k-1,k})_{k=0}^{n}$: $n$-hop boundary matrix