HIGHER-ORDER MOLECULAR LEARNING: THE CELLULAR TRANSFORMER

Melih Barsbey[*]
Department of Computing
Imperial College London
m.barsbey@imperial.ac.uk

Rubén Ballester[*]
Departament de Matemàtiques i Informàtica
Universitat de Barcelona
ruben.ballester@ub.edu

Andac Demir
Novartis
Biomedical Research
andac.demir@novartis.com

Carles Casacuberta
Departament de Matemàtiques i Informàtica
Universitat de Barcelona
carles.casacuberta@ub.edu

Pablo Hernández-García
Departamento de Matemáticas
Universidad de Salamanca
pablohg.eka@usal.es

David Pujol-Perich
Departament de Matemàtiques i Informàtica
Universitat de Barcelona
david.pujolperich@ub.edu

Sarper Yurtseven
Dipartimento di Matematica
Politecnico di Milano
sarperyn@gmail.com

Sergio Escalera
Departament de Matemàtiques i Informàtica
Universitat de Barcelona
Computer Vision Center
sescalera@ub.edu

Claudio Battiloro
Department of Biostatistics
Harvard University
cbattiloro@hsph.harvard.edu

Mustafa Hajij[†]
Department of Data Science
University of San Francisco
mhajij@usfca.edu

Tolga Birdal[†]
Department of Computing
Imperial College London
t.birdal@imperial.ac.uk

ABSTRACT

We present the **Cellular Transformer** (CT), a novel topological deep learning (TDL) framework that extends graph transformers to regular cell complexes (CCs), enabling improved modeling of higher-order molecular structures. Representing complex biomolecules effectively is a notorious challenge due to the delicate interplay between geometry (the physical conformation of molecules) and topology (their connectivity and higher-order relationships). Traditional graph-based models often struggle with these complexities, either ignoring higher-order topological features or addressing them in ad-hoc ways. In this work, we introduce a principled *cellular transformer* mechanism that natively incorporates topological cues (e.g., higher-order bonds, loops, and fused rings). To complement this, we propose the notion of *augmented molecular cell complex*, a novel and richer representation of molecules able to leverage ring-level motifs and features. Our evaluations on the MoleculeNet benchmark and graph datasets lifted into CCs reveal consistent performance gains over GNN- and transformer-based architectures. Notably, our approach achieves these without relying on graph rewiring, virtual nodes, or in-domain structural encodings, indicating the power of topologically informed attention to capture subtle, global interactions vital to drug discovery and molecular property prediction.

[1] ∗, † Equal contribution.

1 INTRODUCTION

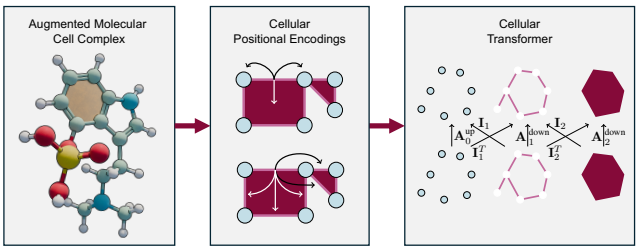


Figure 1: We propose the **cellular transformer** backed by **topological positional encodings** extracted from a novel **augmented molecular cell complex** representation.

Traditional methods for molecular modeling often rely on graph-based models such as graph neural networks (GNNs), graph transformer (Dwivedi et al., 2022) and GPS (Rampasek et al., 2022), that leverage message passing and attention mechanisms to encode molecular graphs effectively. Despite their success, existing approaches remain limited in their ability to model *higher-order molecular interactions* like ring systems, non-covalent bonds, long-range dependencies and non-trivial connectivity patterns, known to be essential to model a molecule (Jiang et al., 2021; Battiloro et al., 2025). This leads to limitations in capturing critical chemical and biological interactions. While graph transformers enjoy improved higher-order feature utilization through the attention mechanisms, they still rely on a graph-based representation that does not natively encode multi-scale molecular structures such as fused rings or higher-dimensional topological motifs. This lack of topological awareness leads to suboptimal feature extraction and generalization.

In this work, we introduce the Cellular Transformer (CT), a novel transformer-based *topological deep learning* (Hajij et al., 2022; Papamarkou et al., 2024) framework that extends molecular modeling beyond graphs (and simplicial complexes (Giusti et al., 2022)) by leveraging *cell complexes* (CCs). CT consists of a higher-order attention mechanisms incorporating topological information by interacting multiple structural levels. We further define *augmented molecular cell complexes* (AMCCs), an enriched molecular representation that captures topological motifs at the ring and bond level, leading to enhanced structural reasoning.

We evaluate CT on molecular benchmarks, including the MoleculeNet, demonstrating its effectiveness across various molecular prediction tasks. Our model consistently outperforms graph message-passing and transformer-based architectures while requiring fewer heuristics. Notably, we show that our approach captures molecular properties more effectively, particularly in datasets where higher-order structural relationships play a critical role. Our results validate the power of topologically informed transformers for molecular property prediction and pave the way for further advancements in topological deep learning for chemistry and materials science.

2 AUGMENTED MOLECULAR CELL COMPLEXES

Cell complexes. Cell complexes encompass various kinds of topological spaces used in network science, including graphs, simplicial complexes, and cubical complexes. While an exhaustive definition can be found in Hatcher (2005), in this work, we restrict ourselves to the 2-dimensional regular CCs for simplicity, although our constructions and discussion carry over to higher-dimensional regular CCs similarly. In our case, we view a *cell complex* (CC) as a triplet $\mathcal{X} = (\mathcal{X}_0, \mathcal{X}_1, \mathcal{X}_2)$ of finite sets and an incidence relationship between their elements. Objects in $\mathcal{X}_i$ are called rank i cells, or i-cells, and denoted by vertices (alternatively nodes), edges, and faces, for $i = 1, 2, 3$, respectively.

A 2-dimensional CC can be realized from a graph by attaching 0-cells to its vertices, 1-cells to its edges, and 2-cells to (possibly a subset of) its induced cycles together with the usual edge-vertices and ring-edges incidence relationships (see App. A.1 for more details). This perspective is particularly useful in molecular modeling, where chemical com-

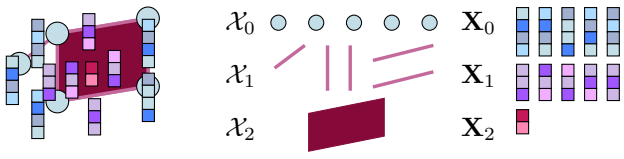


Figure 2: **An annotated CC**. Left: An ACC $\mathcal{X}$ consisting of five vertices, five edges, and one 2-cell. Center: $\mathcal{X}_k$ is the collection of k-cells of $\mathcal{X}$ for $k = 0, 1, 2$. Right: Rows depict values of a cochain $\mathbf{X}_k$ for each k, of dim. $d_0 = 4$, $d_1 = 3$ and $d_2 = 2$.

pounds are frequently represented as graphs, with chemical rings corresponding to induced cycles (Bodnar et al., 2021b; Battiloro et al., 2025).

Cochain space. As shown in Fig. 2, cochain spaces are **used to process data supported over a cell complex** $\mathcal{X} = (\mathcal{X}_0, \mathcal{X}_1, \mathcal{X}_2)$. For $k = 0, 1, 2$, we denote by $\mathcal{C}^k(\mathcal{X}, \mathbb{R}^d)$ the $\mathbb{R}$-vector space of functions $\mathcal{X}_k \to \mathbb{R}^d$, where $d \geq 1$. Here d is called *data dimension* and elements of $\mathcal{C}^k(\mathcal{X}, \mathbb{R}^d)$ are called *k-cochains* or *k-signals* on $\mathcal{X}$. Shortly, we write $\mathcal{C}^k(\mathcal{X})$ instead of $\mathcal{C}^k(\mathcal{X}, \mathbb{R})$ when $d = 1$.

Annotated cell complex. We define a CC $\mathcal{X}$ together with k-cochains $\mathbf{X}_k$ of dimension d_k for each rank $k = 0, 1, 2$ as an *annotated cell complex (ACC)*. We view $\mathbf{X}_k$ as a matrix in $\mathcal{M}(|\mathcal{X}_k|, d_k)$, that is, with $|\mathcal{X}_k|$ rows and d_k columns, whose i^{th} row is the image of the i^{th} element of $\mathcal{X}_k$. In this work, all datasets consist of ACCs sharing the same dimensions d_0, d_1, and d_2.

Augmented molecular cell complex. Prior work demonstrates the importance of higher-order structural motifs (e.g. functional groups, pharmacophores) in molecular modeling as demonstrated in prior work (Battiloro et al., 2025; Luong & Singh, 2024; Zhang et al., 2021). Therefore, we introduce *augmented molecular cell complexes* (AMCC), a novel variant of molecular CCs represented by an ACC with atoms as nodes (0-cells), bonds as edges (1-cells), and rings as faces (2-cells). We summarize AMCCs in Figure 3, and motivate and describe them in further detail in App. A.2.

3 The Cellular Transformer

In this section, we present a general transformer architecture for cell complexes, discuss performing attention on cells, and introduce cellular positional encodings.

Definition 3.1 (Cellular Transformer (CT))**.** A **Cellular Transformer** is a neural network which, given an ACC $\mathcal{X}$, induces a composition of functions $\mathrm{CT} = \mathrm{R} \circ \mathrm{CT}_L \circ \cdots \circ \mathrm{CT}_1 \circ \mathrm{P}$, called **layers**, where P is a preprocessing layer on the input data which combines positional encodings of the different cells in the ACC and their features, R is a readout layer that converts cochains on top of cells into an output prediction value, and CT_l, for $l = 1, \ldots, L$, are CT layers defined as functions of the form

$$\mathrm{CT}_l \colon \mathcal{C}^0(\mathcal{X}, \mathbb{R}^{d_0^h}) \times \cdots \times \mathcal{C}^n(\mathcal{X}, \mathbb{R}^{d_n^h}) \longrightarrow \mathcal{C}^0(\mathcal{X}, \mathbb{R}^{d_0^h}) \times \cdots \times \mathcal{C}^n(\mathcal{X}, \mathbb{R}^{d_n^h}), \tag{1}$$

where $n = \dim \mathcal{X}$ and h indicates the dimension of hidden layers. In our experiments, we set $d_0^h = \cdots = d_n^h$, whose value depends on the dataset, specified in our Appendix.

Eq. (1) only describes the function (co)domains and does not provide an explicit parametrization of the CT layer. A parametrization of the CT layers is given using tensor diagrams together with the cellular attention formulae, as described below. Transformer and preprocessing layers take as input an ACC and output the same CC with different cochains. We denote input k-cochains on the CT_l layer as $\mathbf{X}_{k,l}$. We now introduce the *pairwise cellular attention* mechanism for CTs, which generalizes self- and cross-attention and depends on the dimensions of the cells.

Definition 3.2 (Pairwise Cellular Attention (PCA))**.** Given source and target ranks $0 \leq k_s, k_t \leq \dim \mathcal{X}$ and cochains $\mathbf{X}_{k_t}$, $\mathbf{X}_{k_s}$, the single-head attention from k_s to k_t is a map $\mathcal{C}^{k_s}(\mathcal{X}, \mathbb{R}^{d_s^h}) \times \mathcal{C}^{k_t}(\mathcal{X}, \mathbb{R}^{d_t^h}) \to \mathcal{C}^{k_t}(\mathcal{X}, \mathbb{R}^{d_t^h})$ defined as

$$\mathcal{A}^{\bullet}_{k_s \to k_t}(\mathbf{X}_{k_t}, \mathbf{X}_{k_s}) = \mathrm{softmax}(\mathbf{X}_{k_t}\mathbf{Q}_{k_s \to k_t}(\mathbf{X}_{k_s}\mathbf{K}_{k_s \to k_t})^\top \star \phi(\mathbf{N}_{k_s \to k_t}))\mathbf{X}_{k_s}\mathbf{V}_{k_s \to k_t}, \tag{2}$$

where $\mathbf{Q}_{k_s \to k_t} \in \mathcal{M}(d_t^h, p)$, $\mathbf{K}_{k_s \to k_t} \in \mathcal{M}(d_s^h, p)$, and $\mathbf{V}_{k_s \to k_t} \in \mathcal{M}(d_s^h, d_t^h)$ are learnable query, key, and value real matrices with p a fixed hyperparameter shared by all transformer layers. The symbol $\bullet \in \{d, s\}$ indicates whether we perform dense or sparse attention, respectively. The symbol $\star$ is a sum or a Hadamard product for dense or sparse attention, respectively. $\mathbf{N}_{k_s \to k_t}$ is a neighborhood matrix, and ϕ is a function, possibly with learnable parameters.

The PCA mechanism performs pairwise attention between cells of arbitrary ranks according to a tensor diagram (see App. A.3), and then aggregates the outputs received for the same rank. For our experiments, we set $\phi(\mathbf{N}_{k_s \to k_t})$ to be a learnable embedding layer for dense attention and the identity otherwise. Attention formulae performs query, key, and value projections without bias for simplicity. A bias term can be added to the projections, as in most transformers. Multi-head attention can also be performed by (1) splitting the cochains $\mathbf{X}_{k_s}$ and $\mathbf{X}_{k_t}$ into multiple cochains $\mathbf{X}^1_{k_s}, \ldots, \mathbf{X}^m_{k_s}$ and $\mathbf{X}^1_{k_t}, \ldots, \mathbf{X}^m_{k_t}$ of smaller dimension; (2) performing single-head attention for each pair of cochains

$\mathbf{X}^i_{k_s}, \mathbf{X}^i_{k_t}$; (3) concatenating the outputs of the single-head attention for the different pairs into a full cochain of dimension d^h_t.

For a specific rank k_t, CT layers can produce multiple attention outputs from different rank sources k_s. In the CT layer, we adopt the standard prenorm design (Xiong et al., 2020), where for each rank k_t, the outputs from the various rank sources k_s are added to form the final output for the rank k_t. In App. A we detail the construction of $\mathbf{N}_{k_s \to k_t}$ between and within ranks, and present **tensor diagrams**, which provide a graphical abstraction illustrating the flow of information on one CT layer, guiding the construction of the CT across cochain ranks. Further specifics of the algorithm for the CT layer is detailed in App. B.

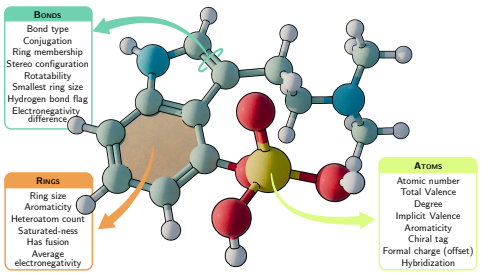


Figure 3: Our **augmented molecular cell complex** enables the topological transformer to use richer information than traditional graphs.

Positional encodings (PE). Transformers do not leverage the input structure explicitly by default (Vaswani et al., 2017). PEs help to overcome this problem by injecting positional and structural information about the input *tokens* in the preprocessing layer of the CT.

Definition 3.3 (Cellular Positional Encoding (CPE))**.** Let $0 \leq k \leq \dim \mathcal{X}$, where $\mathcal{X}$ is a CC. A cellular k-positional encoding of $\mathcal{X}_k$ is a k-cochain $\mathbf{E}_k$ that captures some structural information about $\mathcal{X}_k$ within $\mathcal{X}$[1].

In this work, we extend the Laplacian and Random Walk positional encodings (Müller et al., 2024; Dwivedi et al., 2022) from the graph transformers to the cellular domain, and propose *barycentric subdivision positional encoding* (BSPe) as a novel CPE. For brevity these definitions and further details on CPEs are deferred to App. C.1.

4 Experimental Evaluation

We extensively evaluate the proposed transformer on the MoleculeNet Wu et al. (2018) benchmark as well as in the Graph Classification Benchmark dataset Bianchi et al. (2022a). All our experiments use the AdamW optimizer. For classification tasks we use the standard cross entropy (CE) loss, and for regression we use root mean squared error (RMSE). We present extensive information regarding the architecture, runtime, and feature representations in App. B

Graph Classification Benchmark dataset. We begin by evaluating our method on the Graph Classification Benchmark (GCB) (Bianchi et al., 2022a) in its hard version. As this is a graph dataset, we obtain our CCs by adding all the rings belonging to a cycle basis using the `TopoX` library (Hajij et al., 2024). Tab. 1 presents a comparison of our best topological model using RWBSPe positional encodings against the state-of-the-art architectures evaluated for the dataset. Overall, even with a simple lifting, we outperform several graph-based methods in this dataset, and do so without the need for advanced techniques such as graph rewiring, virtual nodes, or learnable bias matrices in the attention mechanism. The results corroborate that leveraging high-order information can be useful even when the underlying domain is a graph and in such an uninformative case, our method falls back gracefully.

Table 1: Models and accuracy (↑) for the `GCB` dataset. The first seven model rows represent message-passing architectures, and the next six are classic machine learning algorithms.

Model	**Accuracy** (↑)
Graclus Dhillon et al. (2007)	0.690 ± 0.015
NDP Bianchi et al. (2022b)	0.726 ± 0.009
DiffPool Ying et al. (2018)	0.699 ± 0.019
Top-K Gao & Ji (2019)	0.427 ± 0.152
MinCutPool Bianchi et al. (2020)	0.738 ± 0.019
ESC + RBF-SVM Martino et al. (2019)	0.625 ± 0.046
ESC + L1-SVM Martino et al. (2019)	0.722 ± 0.010
ESC + L2-SVM Martino et al. (2019)	0.693 ± 0.016
Hist Kernel Martino & Rizzi (2020)	0.720 ± 0.000
Jaccard Kernel Martino & Rizzi (2020)	0.630 ± 0.000
Edit Kernel Martino & Rizzi (2020)	0.600 ± 0.000
Stratedit Kernel Martino & Rizzi (2020)	0.600 ± 0.000
CT (ours)	$\mathbf{0.754 \pm 0.017}$

[1]Positional encoding may also be defined on the entire CC $\mathcal{X}$.

MoleculeNet. Following Wang et al. (2022), we now compare our work on various subsets of MoleculeNet (Wu et al., 2018) against the strong graph baselines of GCN (Kipf & Welling, 2017), GIN (Xu et al., 2018), SchNet (Schütt et al., 2017), MGCN (Wang et al., 2020b) and DMPNN (Yang et al., 2019), as well as the PyTorch Geometric implementation of GPS (Rampasek et al., 2022) modified to use our AMCC except the 2^{nd}-order features. We also include classical RF (Breiman, 2001) and SVM (Cortes, 1995) performances for reference. We utilize the performances reported by Wang et al. (2022) for models other than GPS and CT. Tab. 2 presents our main results across various tasks where we report Avg. AUC-ROC for classification and root mean square error (RMSE) for regression. Our CT, leveraging the useful information provided by AMCC and barycentric subdivision PEs, can often surpass all methods, or remain on par. Notably, GPS, in the lack of geometric features and limited hyperparameter search, significantly underperforms. This highlights the capability of our CT in extracting strong topological signals and the importance of the utilization of our higher-order AMCC. For further details and statistics on MoleculeNet experiments we refer the reader to App. D.

Table 2: Evaluations across MoleculeNet. All values reported in % with standard deviations computed over 3 runs with different random seeds.

Dataset	**BBBP**	**Tox21**	**ClinTox**	**HIV**	**BACE**	**SIDER**	**MUV**	**FreeSolv**	**ESOL**	**Lipo**
Molecules	2,039	7,831	1,478	41,127	1,513	1,427	93,087	642	1,128	4,200
Tasks	1	12	2	1	1	27	17	1	1	1
Metric	Average AUC-ROC (↑)							RMSE (↓)		
RF	71.4 ± 0.0	76.9 ± 1.5	71.3 ± 5.6	78.1 ± 0.6	$\underline{86.7 \pm 0.8}$	$\mathbf{68.4 \pm 0.9}$	63.2 ± 2.3	-	-	-
SVM	72.9 ± 0.0	$\mathbf{81.8 \pm 1.0}$	66.9 ± 9.2	$\underline{79.2 \pm 0.0}$	86.2 ± 0.0	$\underline{68.2 \pm 1.3}$	67.3 ± 1.3	3.14 ± 0.00	1.50 ± 0.00	0.82 ± 0.00
GCN	71.8 ± 0.9	70.9 ± 2.6	62.5 ± 2.8	74.0 ± 3.0	71.6 ± 2.0	53.6 ± 3.2	71.6 ± 4.0	2.87 ± 0.14	1.43 ± 0.05	0.85 ± 0.08
GIN	65.8 ± 4.5	74.0 ± 0.8	58.0 ± 4.4	75.3 ± 1.9	70.1 ± 5.4	57.3 ± 1.6	71.8 ± 2.5	2.76 ± 0.18	1.45 ± 0.02	0.85 ± 0.07
SchNet	$\underline{84.8 \pm 2.2}$	$\underline{77.2 \pm 2.3}$	71.5 ± 3.7	70.2 ± 3.4	76.6 ± 1.1	53.9 ± 3.7	71.3 ± 3.0	3.22 ± 0.76	1.05 ± 0.06	0.91 ± 0.10
MGCN	$\mathbf{85.0 \pm 6.4}$	70.7 ± 1.6	63.4 ± 4.2	73.8 ± 1.6	73.4 ± 3.0	55.2 ± 1.8	70.2 ± 3.4	3.35 ± 0.01	1.27 ± 0.15	1.11 ± 0.04
D-MPNN	71.2 ± 3.8	68.9 ± 1.3	$\mathbf{90.5 \pm 5.3}$	75.0 ± 2.1	85.3 ± 5.3	63.2 ± 2.3	$\underline{76.2 \pm 2.8}$	2.18 ± 0.91	$\underline{0.98 \pm 0.26}$	$\mathbf{0.65 \pm 0.05}$
GPS	60.4 ± 2.6	63.6 ± 0.6	58.8 ± 7.7	66.8 ± 1.2	72.4 ± 1.0	54.3 ± 0.1	68.3 ± 1.3	$\underline{0.99 \pm 0.04}$	1.14 ± 0.23	0.84 ± 0.4
CT (ours)	71.2 ± 0.5	74.4 ± 1.0	$\underline{86.6 \pm 6.0}$	$\mathbf{79.7 \pm 0.7}$	$\mathbf{86.8 \pm 2.7}$	60.5 ± 1.8	$\mathbf{78.9 \pm 2.0}$	$\mathbf{0.66 \pm 0.04}$	$\mathbf{0.87 \pm 0.03}$	$\underline{0.69 \pm 0.02}$

Conclusion. In this work, we introduced the Cellular Transformer (CT), a TDL framework that extends molecular modeling beyond graph-based methods by leveraging cell complexes. With novel topological positional encodings and augmented molecular cell complexes (AMCCs), our approach captures higher-order molecular interactions without requiring heuristics or graph rewiring. Results presented on MoleculeNet benchmarks demonstrate state-of-the-art or competitive performance across various molecular property prediction tasks while maintaining architectural simplicity.

REFERENCES

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6836–6846, 2021.

Thomas Bailie, Yun Sing Koh, and Karthik Mukkavilli. Higher order graph attention probabilistic walk networks. *arXiv preprint arXiv:2411.12052*, 2024.

Sergio Barbarossa and Stefania Sardellitti. Topological signal processing over simplicial complexes. *IEEE Transactions on Signal Processing*, 68:2992–3007, 2020.

Claudio Battiloro, Ege Karaismailoğlu, Mauricio Tec, George Dasoulas, Michelle Audirac, and Francesca Dominici. E (n) equivariant topological neural networks. *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. Spectral clustering with graph neural networks for graph pooling. In *International Conference on Machine Learning*, pp. 874–883. PMLR, 2020.

Filippo Maria Bianchi, Claudio Gallicchio, and Alessio Micheli. Pyramidal reservoir graph neural network. *Neurocomputing*, 470:389–404, 2022a.

Filippo Maria Bianchi, Daniele Grattarola, Lorenzo Livi, and Cesare Alippi. Hierarchical representation learning in graph neural networks with node decimation pooling. *IEEE Transactions on Neural Networks and Learning Systems*, 33(5):2195–2207, 2022b. doi: 10.1109/TNNLS.2020.3044146.

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yu Guang Wang, Pietro Lio', Guido Montúfar, and Michael M. Bronstein. Weisfeiler and lehman go cellular: Cw networks. In *Neural Information Processing Systems*, 2021a.

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Lio, Guido F Montufar, and Michael Bronstein. Weisfeiler and Lehman go cellular: CW networks. *Advances in Neural Information Processing Systems*, 34:2625–2640, 2021b.

Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

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

James Clift, Dmitry Doryn, Daniel Murfet, and James Wallbridge. Logic and the 2-simplicial transformer. In *International Conference on Learning Representations*, 2020.

George E. Cooke and Ross L. Pinney. *Homology of Cell Complexes*. Princeton University Press, 1967. ISBN 9780691623139.

Corinna Cortes. Support-vector networks. *Machine Learning*, 1995.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11): 1944–1957, 2007. doi: 10.1109/TPAMI.2007.1115.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.

Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations*, 2022.

Vijay Prakash Dwivedi, Chaitanya K. Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24 (43):1–48, 2023.

William Falcon and The PyTorch Lightning team. PyTorch Lightning, 3 2019. URL `https://github.com/Lightning-AI/lightning`.

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

Hongyang Gao and Shuiwang Ji. Graph u-nets. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97, 2019.

Lorenzo Giusti, Claudio Battiloro, Paolo Di Lorenzo, Stefania Sardellitti, and Sergio Barbarossa. Simplicial attention networks. *arXiv preprint arXiv:2203.07485*, 2022.

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

Leo J Grady and Jonathan R Polimeni. *Discrete calculus: Applied analysis on graphs for computational science*, volume 3. Springer, 2010.

Aric Hagberg, Pieter J. Swart, and Daniel A. Schult. Exploring network structure, dynamics, and function using networkx, 1 2008.

Mustafa Hajij, Kyle Istvan, and Ghada Zamzmi. Cell complex neural networks. *NeurIPS Workshop TDA and Beyond*, 2020.

Mustafa Hajij, Ghada Zamzmi, Theodore Papamarkou, Nina Miolane, Aldo Guzmán-Sáenz, Karthikeyan Natesan Ramamurthy, Tolga Birdal, Tamal K Dey, Soham Mukherjee, Shreyas N Samaga, et al. Topological deep learning: Going beyond graph data. *arXiv preprint arXiv:2206.00606*, 2022.

Mustafa Hajij, Ghada Zamzmi, Theodore Papamarkou, AIdo Guzman-Saenz, ToIga Birdal, and Michael T Schaub. Combinatorial complexes: bridging the gap between cell complexes and hypergraphs. In *2023 57th Asilomar Conference on Signals, Systems, and Computers*, pp. 799–803. IEEE, 2023.

Mustafa Hajij, Mathilde Papillon, Florian Frantzen, Jens Agerberg, Ibrahem AlJabea, Rubén Ballester, Claudio Battiloro, Guillermo Bernárdez, Tolga Birdal, Aiden Brent, Peter Chin, Sergio Escalera, Simone Fiorellino, Odin Hoff Gardaa, Gurusankar Gopalakrishnan, Devendra Govil, Josef Hoppe, Maneel Reddy Karri, Jude Khouja, Manuel Lecha, Neal Livesay, Jan Meißner, Soham Mukherjee, Alexander Nikitin, Theodore Papamarkou, Jaro Prílepok, Karthikeyan Natesan Ramamurthy, Paul Rosen, Aldo Guzmán-Sáenz, Alessandro Salatiello, Shreyas N. Samaga, Simone Scardapane, Michael T. Schaub, Luca Scofano, Indro Spinelli, Lev Telyatnikov, Quang Truong, Robin Walters, Maosheng Yang, Olga Zaghen, Ghada Zamzmi, Ali Zia, and Nina Miolane. Topox: a suite of python packages for machine learning on topological domains. *Journal of Machine Learning Research*, 25(374):1–8, 2024.

Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022.

J. Hansen and R. Ghrist. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology*, 2019.

Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2005.

Daniel Hernández Serrano, Juan Hernández-Serrano, and Darío Sánchez Gómez. Simplicial degree in complex networks. Applications of topological data analysis to network science. *Chaos Solitons Fractals*, 137:109839, 21, 2020. ISSN 0960-0779.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33, 2020.

Zehua Hu, Jiahai Wang, Siyuan Chen, and Xin Du. A semi-supervised framework with efficient feature extraction and network alignment for user identity linkage. In *Database Systems for Advanced Applications: 26th International Conference, DASFAA 2021, Taipei, Taiwan, April 11–14, 2021, Proceedings, Part II 26*, pp. 675–691. Springer, 2021.

Md Shamim Hussain, Mohammed J Zaki, and Dharmashankar Subramanian. Edge-augmented graph transformers: Global self-attention is enough for graphs. *arXiv preprint arXiv:2108.03348*, 2021.

Yi Jiang, Dong Chen, Xin Chen, Tangyi Li, Guo-Wei Wei, and Feng Pan. Topological representations of crystalline compounds for the machine-learning prediction of materials properties. *npj computational materials*, 7(1):28, 2021.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, 2019.

Jinwoo Kim, Saeyoon Oh, and Seunghoon Hong. Transformers generalize deepsets and can be extended to graphs & hypergraphs. *Advances in Neural Information Processing Systems*, 34: 28016–28028, 2021.

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

Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12939–12948, 2021.

Kha-Dinh Luong and Ambuj K Singh. Fragment-based pretraining and finetuning on molecular graphs. *Advances in Neural Information Processing Systems*, 36, 2024.

Alessio Martino and Antonello Rizzi. (hyper)graph kernels over simplicial complexes. *Entropy*, 22 (10), 2020. ISSN 1099-4300. doi: 10.3390/e22101155.

Alessio Martino, Alessandro Giuliani, and Antonello Rizzi. (hyper)graph embedding and classification via simplicial complexes. *Algorithms*, 12(11), 2019. ISSN 1999-4893.

Grégoire Mialon, Dexiong Chen, Margot Selosse, and Julien Mairal. Graphit: Encoding graph structure in transformers, 2021.

Erxue Min, Runfa Chen, Yatao Bian, Tingyang Xu, Kangfei Zhao, Wenbing Huang, Peilin Zhao, Junzhou Huang, Sophia Ananiadou, and Yu Rong. Transformer for graphs: An overview from architecture perspective. *arXiv preprint arXiv:2202.08455*, 2022a.

Erxue Min, Yu Rong, Tingyang Xu, Yatao Bian, Peilin Zhao, Junzhou Huang, Da Luo, Kangyi Lin, and Sophia Ananiadou. Masked transformer for neighhourhood-aware click-through rate prediction. *CoRR*, abs/2201.13311, 2022b.

Luis Müller, Mikhail Galkin, Christopher Morris, and Ladislav Rampášek. Attending to graph transformers. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL `https://openreview.net/forum?id=HhbqHBBrfZ`.

Theodore Papamarkou, Tolga Birdal, Michael Bronstein, Gunnar Carlsson, Justin Curry, Yue Gao, Mustafa Hajij, Roland Kwitt, Pietro Liò, Paolo Di Lorenzo, Vasileios Maroulas, Nina Miolane, Farzana Nasrin, Karthikeyan Natesan Ramamurthy, Bastian Rieck, Simone Scardapane, Michael T. Schaub, Petar Veličković, Bei Wang, Yusu Wang, Guo-Wei Wei, and Ghada Zamzmi. Position paper: Challenges and opportunities in topological deep learning, 2024.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Ladislav Rampasek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and D. Beaini. Recipe for a general, powerful, scalable graph transformer. *ArXiv*, abs/2205.12454, 2022.

T Mitchell Roddenberry, Michael T Schaub, and Mustafa Hajij. Signal processing on cell complexes. *Proc. IEEE ICASSP*, 2022.

Sebastian G Rohrer and Knut Baumann. Maximum unbiased validation (muv) data sets for virtual screening based on pubchem bioactivity data. *Journal of chemical information and modeling*, 49 (2):169–184, 2009.

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

Stefania Sardellitti and Sergio Barbarossa. Topological signal representation and processing over cell complexes. *arXiv preprint arXiv:2201.08993*, 2022.

Michael T. Schaub, Austin R. Benson, Paul Horn, Gabor Lippner, and Ali Jadbabaie. Random walks on simplicial complexes and the normalized hodge 1-laplacian. *SIAM Review*, 62(2):353–391, 2020.

Kristof T Schütt, Farhad Arbabzadah, Stefan Chmiela, Klaus R Müller, and Alexandre Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature communications*, 8(1), 2017.

Cong Shen, Yipeng Zhang, Fei Han, and Kelin Xia. Molecular topological deep learning for polymer property prediction. *arXiv preprint arXiv:2410.04765*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Michelle L Wachs. Poset topology: tools and applications. *arXiv preprint math/0602226*, 2006.

Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. Next-item recommendation with sequential hypergraphs. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pp. 1101–1110, 2020a.

Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*, 2019.

Yiqun Wang, Jing Ren, Dong-Ming Yan, Jianwei Guo, Xiaopeng Zhang, and Peter Wonka. Mgcn: descriptor learning using multiscale gcns. *ACM Transactions on Graphics (TOG)*, 39(4):122–1, 2020b.

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

Kedi Wu and Guo-Wei Wei. Quantitative toxicity prediction using topology based multitask deep neural networks. *Journal of chemical information and modeling*, 58, 2018.

Zhanghao Wu, Paras Jain, Matthew Wright, Azalia Mirhoseini, Joseph E Gonzalez, and Ion Stoica. Representing long-range context for graph neural networks with global attention. *Advances in Neural Information Processing Systems*, 34, 2021.

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

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119. PMLR, 2020.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2018.

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

Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Heng-Kai Zhang, Yi-Ge Zhang, Zhi Zhou, and Yu-Feng Li. Hongat: Graph attention networks in the presence of high-order neighbors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16750–16758, 2024.

Jiawei Zhang, Haopeng Zhang, Congying Xia, and Li Sun. Graph-bert: Only attention is needed for learning graph representations. *arXiv preprint arXiv:2001.05140*, 2020a.

Ruochi Zhang, Yuesong Zou, and Jian Ma. Hyper-SAGNN: a self-attention based graph neural network for hypergraphs. In *International Conference on Learning Representations (ICLR)*, 2020b.

Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Chee-Kong Lee. Motif-based graph self-supervised learning for molecular property prediction. *Advances in Neural Information Processing Systems*, 34:15870–15882, 2021.

Cai Zhou, Xiyuan Wang, and Muhan Zhang. Facilitating graph neural networks with random walk on simplicial complexes. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Cai Zhou, Rose Yu, and Yusu Wang. On the theoretical expressive power and the design space of higher-order graph transformers, 2024.

Appendix

A Additional Details on Cellular Transformer

Here we present additional detail on cellular transformer (CT) that were deferred due to space constraints.

A.1 Neighborhood and Incidence Matrices

CCs admit a poset representation (Hansen & Ghrist, 2019), i.e., a partial order $\leq$ on $\mathcal{X}$. The poset representation enables a combinatorial description of the complex and allows us to define an incidence relation among cells.

Definition A.1 (Incidence Relation)**.** A cell τ is incident to a cell σ, denoted with $\sigma \prec \tau$, iff $\dim(\sigma) \leq \dim(\tau)$ and there is no cell δ such that $\sigma \leq \delta \leq \tau$.

An edge is incident to its endpoint nodes, and a face is incident to the edges on its sides. We can use these definitions to introduce the four types of neighborhoods present in CCs.

Definition A.2 (Neighborhoods)**.** (Hajij et al., 2020) For a cell complex $\mathcal{C}$ and a cell $\sigma \in \mathcal{P}_{\mathcal{C}}$, we define:

- The boundary cells of σ are the lower-dimensional cells σ is incident to. For instance, the boundary cells of an edge are its endpoint nodes.
- The co-boundary cells of σ are the higher-dimensional cells incident to σ. For instance, the co-boundary cells of a node are the edges having that node as an endpoint.
- The lower adjacent cells of σ are the cells of the same dimension as σ incident to common lower dimensional cells. For instance, two edges are lower adjacent if they have a common endpoint node.
- The upper adjacent cells of σ are the cells of the same dimension as σ common higher dimensional cells are incident to. For instance, two edges are upper adjacent if they are both sides of a common face.

Neighborhoods in cell complexes can be described through incidence and adjacency matrices, similar to graphs.

Definition A.3 (Incidence and adjacency matrices)**.** Given an arbitrary labeling of the cells, the entry (i, j) of the first incidence matrix $\mathbf{B}_1 \in \mathbb{R}^{|\mathcal{X}_0| \times |\mathcal{X}_1|}$ is non-zero if the j^{th} edge is incident to the i^{th} node. Similarly, the entry (i, j) of the second incidence matrix $\mathbf{B}_2 \in \mathbb{R}^{|\mathcal{X}_1| \times |\mathcal{X}_2|}$ is non-zero if the j^{th} face is incident to the i^{th} edge. The entry (i, j) of the node upper adjacency matrix $\mathbf{A}_0^{\text{up}} \in \mathbb{R}^{|\mathcal{X}_0| \times |\mathcal{X}_0|}$ (the usual graph adjacency) is non-zero if the i^{th} and the j^{th} nodes are endpoints of a common edge. The entry (i, j) of the edge lower adjacency matrix $\mathbf{A}_1^{\text{low}} \in \mathbb{R}^{|\mathcal{X}_1| \times |\mathcal{X}_1|}$ is non-zero if the i^{th} and the j^{th} edges share a common endpoint node. The entry (i, j) of the edge upper adjacency matrix $\mathbf{A}_1^{\text{up}} \in \mathbb{R}^{|\mathcal{X}_1| \times |\mathcal{X}_1|}$ is non-zero if the i^{th} and the j^{th} edges are both sides of a common face. Finally, the entry (i, j) of the face lower adjacency matrix $\mathbf{A}_2^{\text{low}} \in \mathbb{R}^{|\mathcal{X}_2| \times |\mathcal{X}_2|}$ is non-zero if the i^{th} and the j^{th} faces share a common edge.

It is clear from A.3, that the incidence matrices and their transpose encode the co-boundary and boundary, respectively. In App. C, we also introduce the notion of orientation and signed incidence matrices, necessary to define a proper discrete (algebraic) Hodge theory for cell complexes Barbarossa & Sardellitti (2020), which we use to define positional encodings on CCs. We collectively refer to incidence and adjacencies matrices as neighborhood matrices.

A.2 Augmented Molecular Cell Complexes

Recent GDL / TDL approaches to molecular modeling, especially those leveraging graph transformers, have benefited greatly from the integration of ring-level or other higher-order structural motifs (e.g., functional groups, pharmacophores) Battiloro et al. (2025); Luong & Singh (2024); Zhang et al. (2021). A prime example is cyclohexyl rings versus straight aliphatic chains. Chemically, both structures are made of carbon atoms that are sp^3-hybridized and form single σ-bonds. Hence, while atoms and bonds are formally of the same type, ring closure in cyclohexane imposes unique structural constraints (e.g., the well-known chair conformation) that differ from those in a linear alkane.

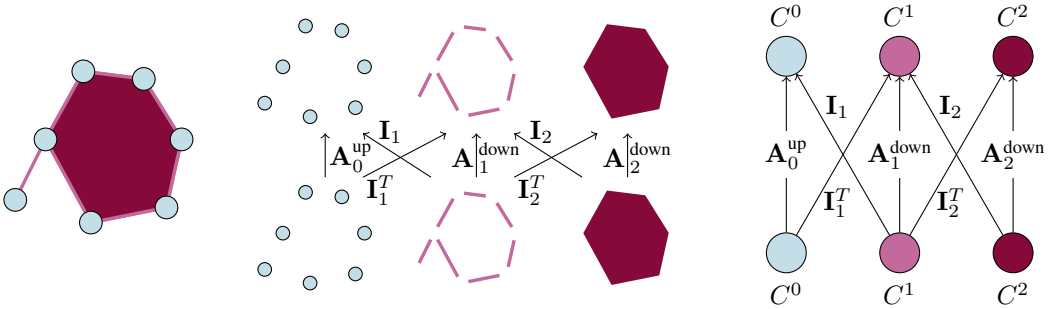


Figure 4: Attention Tensor diagram illustrating the flow of signals between cochains defined on 0-, 1-, and 2-cells. For pairwise attention, the neighborhood matrices indicate the bias $\mathbf{N}$ in the attention formula (2). For general attention, neighborhood matrices indicates how to build the bias matrix $\mathbf{N}$ by composition of smaller bias matrices $\mathbf{N}_{k_s \to k_t}$ between dimensions.

As such, methods treating each carbon atom and each bond equally often overlook these subtle but crucial differences. CCs are a natural solution to this issue, as rings can be explicitly regarded as 2-cells (faces) with their peculiar (ring-level) features, and the topology of the complex inherently captures vital structural constraints. Thus, we introduce **augmented molecular cell complexes** (AMCCs), novel variants of molecular CCs in which atoms are nodes (0-cells), bonds are edges (1-cells), and rings are faces (2-cells). Here, *augmentation* refers to a curated set of features for all the cells being more exhaustive w.r.t. to the ones in Battiloro et al. (2025), and resulting in significantly better results, shown in our ablation study in Sec. 4. An example of an AMCC listing all the features we leverage is depicted in Fig. 3. Our CC-Transformer is then naturally able to process and learn for the rich representation offered by AMCCs, as confirmed by the numerical results in Sec. 4. We provide an overview of AMCC features in App. D.3.1.

Remark A.4. *An informed reader may wonder why we adopted CCs rather than combinatorial complexes Hajij et al. (2023) as the underlying topological domain, as the latter are more flexible in modeling relational structures. The reason is that, combinatorial complexes still lack an exhaustive theoretical characterization, having neither a spectral nor a homology theory. Thus we would have not been able to define or leverage powerful PEs as the ones in App. C.1. We leave this promising avenue for future work.*

A.3 Attention Tensor Diagrams for CTs

CTs involve interactions between cochains of different ranks. Tensor diagrams Hajij et al. (2022) provide a graphical abstraction illustrating the flow of information on one CT layer. A tensor diagram portrays a CT Layer through the use of a directed graph, where the nodes represent cochain spaces for different ranks $0 \leq k \leq n$, n being the maximum allowed rank of CCs processed by that CT layer. If the input CC $\mathcal{X}$ is of lower dimension than n, the attention on ranks $k > \dim \mathcal{X}$ are ignored. In turn, edges represent either the pairwise attentions performed in the CT layer together with the bias matrices $\mathbf{N}_{k_s \to k_t}$, A missing arrow from cochains of rank k_s to cochains of rank k_t implies no pairwise attention from source cells of rank k_s for target cells k_t. Fig. 4 illustrates the tensor diagram used in our experiments.

In cell complex molecular modeling, incidence matrices in a molecular cell complex enable cross-attention, while adjacency and Hodge-Laplacian matrices enable self-attention within our CT framework. In particular, cross-attention captures multiscale interactions between structural levels, such as nodes, edges, and cycles, while self-attention uses adjacency relations to refine feature aggregation within the same rank. This dual mechanism enhances molecular representations, capturing both local and global dependencies.

Tensor diagrams provide a graphical abstraction of these interactions, guiding the construction of the CT across cochain ranks. This aids in designing custom attention mechanisms within the transformer, ensuring effective encoding of both cross-rank and self-attention patterns.

A.4 Positional Encodings on Cellular Complexes

Transformers do not leverage the input structure explicitly by default Vaswani et al. (2017). Positional encodings (PEs) help to overcome this problem by injecting positional and structural information

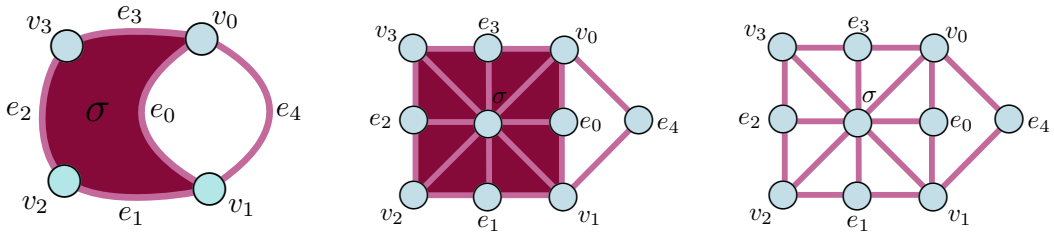


Figure 5: Left: A cell complex $\mathcal{X}$. Center: Barycentric subdivision of $\mathcal{X}$. Right: 1-skeleton of the barycentric subdivision. Each original cell of $\mathcal{X}$ is represented by a node in the 1-skeleton.

about the input *tokens*. For sequences, the first positional encoding used sine and cosine functions depending on the position of the token in the sequence. For graphs, several positional encodings have been studied such as the eigenvectors of the graph Laplacian (LapPE) Dwivedi et al. (2023) and Random Walk Positional Encodings (RWPe) Dwivedi et al. (2022), where the latter were also adapted for simplicial complex transformers Zhou et al. (2023); Schaub et al. (2020).

Definition A.5 (Cellular Positional Encoding (CPE))**.** Let $0 \leq k \leq \dim \mathcal{X}$, where $\mathcal{X}$ is a CC. A cellular k-positional encoding of $\mathcal{X}_k$ is a k-cochain $\mathbf{E}_k$ that captures some structural information about $\mathcal{X}_k$ within $\mathcal{X}$[2].

Given cochains $\mathbf{X}_k$ and positional encodings $\mathbf{E}_k$, the input for the first transformer layer is defined as a function $\mathrm{P}_k \colon \mathcal{C}^k(\mathcal{X}, \mathbb{R}^{d_k}) \times \mathcal{C}^k(\mathcal{X}, \mathbb{R}^{d_{\mathrm{pe}}}) \to \mathcal{C}^k(\mathcal{X}, \mathbb{R}^{d_k^h})$ computed at the end of the preprocessing layer with $\mathbf{X}_k^1 = \mathrm{P}_k(\mathbf{X}_k, \mathbf{E}_k)$, where P_k combines the signals and the positional encodings. Usual functions are

$$\mathrm{SumPE}(\mathbf{X}_k, \mathbf{E}_k) = \mathbf{X}_k\theta_{\mathrm{in},k} + b_{\mathrm{in},k} + \mathbf{E}_k\theta_{\mathrm{in,pe}} + b_{\mathrm{in,pe}}$$
$$\mathrm{ConcatPE}(\mathbf{X}_k, \mathbf{E}_k) = \mathrm{Concat}(\mathbf{X}_k, \mathbf{E}_k)\theta_{\mathrm{in,pe}} + b_{\mathrm{in,pe}},$$

where $\theta_\bullet$, $b_\bullet \in$ are learnable parameters. For this paper, we use $\mathrm{P}_k = \mathrm{ConcatPE}$. Next, we recall the popular positional encodings for graph transformers before introducing three novel positional encodings on cell complexes: Barycentric Subdivision, Random Walk, and Topological Slepians.

Definition A.6 (Laplacian PE (LapPE) Dwivedi et al. (2023))**.** A canonical PE for graphs is given by graph Laplacian eigenvector, by assigning to each vertex v_i a vector $\mathrm{LapPE}(v_i) = (e_i^1, \ldots, e_i^k)$, where $\{e_i^j \mid j = 1, \ldots, k\}$ are eigenvectors of the k smallest eigenvalues of the normalized graph Laplacian for a graph $G = (V, E)$, counting multiplicities, where k is a hyperparameter.

A naive extension of LapPE to cell complexes involves using the unnormalized Hodge Laplacian matrix, instead of the graph Laplacian one. We deem this version **HodgeLapPE**. We use the unnormalized version because normalizing the Hodge Laplacian for dimensions greater than zero is not a trivial task Schaub et al. (2020). HodgeLapPE are, however, not a good choice for high-order positional encodings *a priori* due to both a lack of normalization and the ambiguous information contained in Hodge Laplacians for nonzero rank. Details on LapPE for graphs and their HodgeLapPE extension can be found in App. C.

Definition A.7 (Random Walk PE (RWPe) Dwivedi et al. (2022))**.** Given a vertex $v_i \in V$, the RWPe of v_i is given by the vector $\mathrm{RWPe}(v_i) = \left(\mathrm{RW}_{ii}, \ldots, \mathrm{RW}_{ii}^k\right)$ where $\mathrm{RW} = AD^{-1}$ is the random walk operator of a graph based on edge connectivity. In the case that each vertex is assigned the probabilities of landing again on itself on the random walks from one to k steps.

While $\mathrm{RWPe}(v_i)$ is unique and does not need sign or eigenvector selection invariance, defining meaningful random walks on general cell complexes is a notoriously difficult task, first explored in Schaub et al. (2020) and then in Zhou et al. (2023) for simplicial complexes, making a special focus on edge random walks.

To address both drawbacks and overcome these drawbacks, we consider smarter ways towards higher-order extensions.

Definition A.8 (Barycentric Subdivision Positional Encoding (**BSPe**))**.** Our first PE leverages the graph Laplacian eigenvectors of the 1-skeleton of the barycentric subdivisions of the CCs, where the

[2]Positional encoding may also be defined on the entire CC $\mathcal{X}$.

barycentric subdivision of a CC $\mathcal{X}$, denoted by $\Delta(\mathcal{X})$, is the order complex of its face poset Wachs (2006), i.e., the abstract simplicial complex whose set of vertices is the set of cells of $\mathcal{X}$ and whose simplices are the totally ordered flags of cells of $\mathcal{X}$. The 1**-skeleton** of the barycentric subdivision of $\mathcal{X}$ is a graph $G = (V, E)$ where V is the set of cells of $\mathcal{X}$ and where two vertices σ_1 and σ_2 are connected if one is a face of the other.

Barycentric subdivisions yield triangulations of cell complexes that preserve their topological properties Cooke & Pinney (1967). The positional encoding of a cell σ is the Laplacian positional encoding of σ seen as a vertex in G.

This positional encoding respects the same theoretical advantages of the LapPE while assigning relative positions to all the cells *at the same time*, and thus relative positions take into account all the cells and not only the cells of a specific dimension. The positional encodings satisfying this property are called *global*, in contrast to *local* PEs, where encodings are assigned independently for each dimension.

Definition A.9 (CC Random Walk PE (CC-RWPe))**.** Similar to dfn. A.8, we propose to extend RWPe to cell complexes by taking the positional encodings given by the original RWPe for the 1-skeleton of the barycentric subdivision. We denote these *global* positional encodings as **CC-RWBSPe**. We also propose a more sophisticated, local approach, denoted **CC-RWPe**, extending the random walks from Schaub et al. (2020) to cell complexes. The full development of the random walk matrix can be found in App. C.1. From the random walk matrix, PEs are taken as in RWPe for each rank of the cell complex.

A.5 Limitations and future work

Our novel CTs excel at leveraging molecular topological structures but are less effective in directly modeling geometry. As a remedy, we aim to develop equivariant-CTs that incorporate physical conformations. We also plan to explore the spectral and homology theories of complexes to strengthen the foundation of TDL and design Combinatorial Complex Transformers for more chemically informed molecular modeling. Finally, we plan to learn the topological features, scaling to larger biomolecules, and extending TDL to scientific domains like materials science and drug discovery.

B Architectural Details and Evaluations on Graph Benchmarks

B.1 Architecture details

In this section, we describe the two different topological transformer layers proposed in the main text in detail. Following the prenorm Xiong et al. (2020) deisng, at the end of the transformer block, composed of several transformer layers, a final layer norm is applied, either for each rank in the case of the pairwise attention transformer layer, and for the whole set of cells in the case of the general attention transformer layer.

Pairwise attention transformer layer. Following the usual prenorm design Xiong et al. (2020), the output of the cellular transformer layer for a specific rank k_t is denoted $\mathbf{X}_{k_t,l+1}$ and computed in six steps, as follows:

$\mathbf{X}^1_{k_t,l} = \text{LayerNorm}_{k_t}(\mathbf{X}_{k_t,l})$,

$\mathbf{X}^1_{k_s,l} = \text{LayerNorm}_{k_s}(\mathbf{X}_{k_s,l})$ for each k_s in the tensor diagram,

$\mathbf{X}^2_{k_s\to k_t,l} = \mathcal{A}^{\bullet}_{k_s\to k_t}(\mathbf{X}^1_{k_t,l}, \mathbf{X}^1_{k_s,l})$ for each k_s in the tensor diagram,

$\mathbf{X}^3_{k_s\to k_t,l} = \text{Dropout}(\mathbf{X}^2_{k_s\to k_t,l})$ for each k_s in the tensor diagram,

$\mathbf{X}^4_{k_s\to k_t,l} = \mathbf{X}_{k_t,l} + \mathbf{X}^3_{k_s\to k_t,l}$ for each k_s in the tensor diagram,

$\mathbf{X}^5_{k_s\to k_t,l} = \text{LayerNorm}(\mathbf{X}^4_{k_s\to k_t,l})$ for each k_s in the tensor diagram,

$\mathbf{X}^6_{k_s\to k_t,l} = \text{Dropout}(\text{FFN}_2(\text{Dropout}(\text{ReLU}(\text{FFN}_1(\mathbf{X}^5_{k_s\to k_t,l})))))$ for each k_s in the tensor diagram,

$\mathbf{X}_{k_s\to k_t,l+1} = \mathbf{X}^4_{k_s\to k_t,l} + \mathbf{X}^6_{k_s\to k_t,l}$ for each k_s in the tensor diagram,

$\mathbf{X}_{k_t,l+1} = \sum_{k_s} \mathbf{X}_{k_s\to k_t,l+1}$.

The LayerNorm is unique for each rank k_t and each layer l.

General attention transformer layer. Similarly to the pairwise attention transformer layer, the general attention transformer layer performs the following steps:

$$\begin{aligned}
\mathbf{X}_l^1 &= \text{LayerNorm}(\mathbf{X}_l),\\
\mathbf{X}_l^2 &= \mathcal{A}_g^\bullet(\mathbf{X}_l^1),\\
\mathbf{X}_l^3 &= \text{Dropout}(\mathbf{X}_l^2),\\
\mathbf{X}_l^4 &= \mathbf{X}_l + \mathbf{X}_l^3,\\
\mathbf{X}_l^5 &= \text{LayerNorm}(\mathbf{X}_l^4),\\
\mathbf{X}_l^6 &= \text{Dropout}(\text{FFN}_2(\text{Dropout}(\text{ReLU}(\text{FFN}_1(\mathbf{X}_l^5))))),\\
\mathbf{X}_{l+1} &= \mathbf{X}_l^4 + \mathbf{X}_l^6.
\end{aligned}$$

Training details. All experiments use a `Cosine Annealing` scheduler with linear warmup, an `AdamW` optimizer with $\epsilon = 1^{-8}$, $(\mu_1, \mu_2) = (0.9, 0.999)$ and variable peak learning rate, and a gradient clipping norm of 5. All our transformer architectures, after the transformer layers, use a fully-connected readout whose dropout and number of hidden layers is fixed for each set of experiments, followed by a global add pool layer over all the vertex signals to perform prediction or regression. The fully-connected block begins with a number of neurons equivalent to the hidden dimension of the transformers and concludes with a number of neurons corresponding to the network's output number. Throughout the block, each hidden layer has half as many neurons as its predecessor.

Signals on cells. All graphs in the three datasets contain at least discrete signals for the vertices. For the `GCB` dataset, we associate to each edge a signal corresponding to concatenating the signals of its endpoints. For the `ogbg-molhiv` and `ZINC` datasets, the edges contain signals, so we do not change them. As a first step in the transformer architecture, we learn an embedding for the discrete features. For the edge features in `GCB`, each vertex feature is embedded individually. For the three datasets, signals on the 2-cells are given by sum of the embedded signals of their vertices.

`GCB` architectures. The first six models in Tab. 1 assesses all graph neural networks with different graph pooling layers and common architecture composition given by MP(32)-Pool-MP(32)-Pool-MP(32)-GlobalPool-Dense(Softmax), where MP(32) is a Chebyshev convolutional layer Defferrard et al. (2016) with 32 hidden units, Pool is a pooling message passing layer, GlobalPool is a global pool layer used as readout, and Dense(Softmax) is a dense layer with softmax activation. Skip connections were used. The other state-of-the-art models consist of models proposed in Martino et al. (2019); Martino & Rizzi (2020).

B.2 Dataset Statistics

We use the following molecular graph datasets, primarily obtained from MoleculeNet (Wu et al., 2018). Unless otherwise specified, we follow the scaffold splitting protocol (80/10/10) provided by OGB (Hu et al., 2020). Below, we list the fine-tuning datasets:

1. **BBBP**: Contains 2039 molecules with binary labels for blood-brain barrier penetration.
2. **Tox21**: Comprises 7831 molecules with binary labels indicating toxicity for 12 different targets.
3. **ClinTox**: Includes 1478 drugs with two binary annotations: (1) toxicity in clinical trials, and (2) FDA approval status.
4. **HIV**: A dataset of 41k molecules annotated with binary labels for their ability to inhibit HIV virus replication.
5. **BACE**: Consists of 1513 molecules with binary labels indicating binding results for inhibitors of human β-secretase 1.
6. **SIDER**: Consists of 1427 approved drugs, each annotated with 27 side-effect groups. The prediction task is to determine whether a drug belongs to each side-effect group.
7. **MUV**: Consists of 93k molecules curated from PubChem bioassays to remove screening artifacts Rohrer & Baumann (2009). It provides 17 challenging tasks typically used to assess virtual screening performance.
8. **FreeSolv**: Contains 642 molecules with hydration free energy data in water.

9. **ESOL**: Contains 1128 common organic small molecules with water solubility data (log solubility in mols per liter).
10. **Lipo**: Consists of 4200 molecules with experimental data for the octanol/water distribution coefficient.

Dataset	#Graphs	Avg. #Atoms	Avg. #Bonds	Split	#Classes/Task
BBBP	2,039	24.1	26.0	Scaffold	1 (Classification)
Tox21	7,831	18.6	19.3	Scaffold	12 (Classification)
ClinTox	1,478	26.2	27.9	Scaffold	2 (Classification)
HIV	41,127	25.5	25.5	Scaffold	1 (Classification)
BACE	1,513	34.1	36.9	Scaffold	1 (Classification)
SIDER	1,427	33.6	35.4	Scaffold	27 (Classification)
MUV	93,087	24.2	26.3	Scaffold	17 (Classification)
Freesolv	642	8.7	8.4	Scaffold	Regression
ESOL	1,427	13.3	13.7	Scaffold	Regression
LIPO	93,087	27.0	29.5	Scaffold	Regression

Table 3: Statistics of the used datasets.

B.3 Implementation and hardware resources

Implementation was performed mainly using the `TopoNetX` Hajij et al. (2024) library for cell complex representation and manipulation, `PyTorch` Paszke et al. (2019) for the deep learning pipelines, `PyTorch Geometric` Fey & Lenssen (2019) for feature pooling and dataset loading, `Deep Graph Library` Wang et al. (2019) and `Scipy` Virtanen et al. (2020) for sparse tensor algebraic operations and sparse tensor representation and manipulation, `NetworkX` Hagberg et al. (2008) for graph manipulation, and `PyTorch Lightning` Falcon & The PyTorch Lightning team (2019) as a top layer for experimentation in `PyTorch`. The most critical pieces of software implemented in this project have been the `DataLoader` and the `collate` function to batch cell complexes. The `DataLoader` is implemented in the class `TopologicalTransformerDataLoader`. The collate function is implemented in the function `collate`, both inside the file `src/datasets/cell_dataloader.py`. The `collate` function creates a cell complex batch by performing the disjoint union of cell complexes. As an input, the `collate` function receives an object with the signals for each cell, the neighborhood matrices used in the transformer architecture as bias $\mathbf{N}$ in a sparse format, and other data needed by the experiments such as the label of the dataset and the positional encodings. The neighborhood matrices are batched into a new sparse block matrix, taking into account that different cell complexes may have different dimensions and thus not all the cell complexes have the same neighborhood matrices. Currently, the `collate` function supports adjacency and boundary matrices, although the function can be extended easily. Signals, positional encodings, and labels are simply concatenated. To keep track of which signals and positional encodings correspond to each of the individual cell complex, we also return, for each dimension, a vector of size equal to total number of cells of that dimension in the disjoint union which indicates to which cell complex belong each signal or positional encoding.

The experiments were executed on a server with an AMD EPYC 7452 (128) @ 2.350GHz CPU, 503GiB of RAM memory, x4 PNY Nvidia RTX 6000 Ada Generation 48GB GPUs, and Ubuntu 22.04.4 LTS with the 6.5.0-28-generic Linux kernel. Each experiment was executed on a separated GPU device, using 12 workers per experiment.

C Mathematical details and examples

Cell complexes. An exhaustive definition of cell complexes in the context of algebraic topology can be found in Hatcher (2005). In brief, a cell complex is a topological space $\mathcal{X}$ that can be decomposed as a union of disjoint subspaces called *cells*, where each cell σ is homeomorphic to $\mathbb{R}^k$ for some integer $k \geq 0$, called the *rank* of σ. Additionally, for every cell σ, the difference $\overline{\sigma} \setminus \sigma$ is a union of finitely many cells of lower rank, where $\overline{\sigma}$ denotes the closure of σ. The *dimension* of a finite cell

complex is the maximum of the ranks of its cells. The set of cells of rank k in a cell complex $\mathcal{X}$ is denoted by $\mathcal{X}_k$. The *n-skeleton* of $\mathcal{X}$ is the cell complex spanned by $\mathcal{X}_0, \ldots, \mathcal{X}_n$, for $0 \leq n \leq \dim \mathcal{X}$.

A *characteristic map* for a cell σ of rank k is a map from the Euclidean unit closed ball of dimension k into $\overline{\sigma}$ whose restriction to the open ball is a homeomorphism. A cell complex is called *regular* if each cell σ admits a characteristic map which is itself a homeomorphism from the closed ball to $\overline{\sigma}$. For example, a decomposition of a circle as the union of a 0-cell and a 1-cell is not regular. Geometric realizations of abstract simplicial complexes are regular cell complexes. Cell complexes generalize simplicial complexes as their cells are not constrained to be simplices.

Algebraic Description of Cell Complexes. It is possible a rich algebraic representation of cell complexes This description also provides a simple tool to develop a discrete Hodge theory for cell complexes (Grady & Polimeni, 2010). To do so, it is essential to first introduce an orientation of the cells. Orienting cells is not mathematically trivial but, in the end, it is only a "bookkeeping matter" (Roddenberry et al., 2022). One of the possible ways of orienting cells (Sardellitti & Barbarossa, 2022) is via a simplicial decomposition of the complex, i.e. subdividing the cell into a set of internal k-simplices (Barbarossa & Sardellitti, 2020), so that i) two simplices share exactly one $(k-1)$-simplicial boundary element, which is not the boundary of any other k-cell in the complex; and ii) two k-simplices induce an opposite orientation on the shared $(k-1)$-boundary. Therefore, by orienting a single internal simplex, the orientation propagates on the entire cell. Given an orientation, we can introduce the signed incidence matrices.

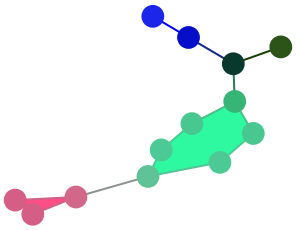


Figure 6: BSPe positional encoding of length three for a cell complex with two 2-cells. To generate a colour from the positional encoding, we normalize each coordinate of the positional encodings to the $[0, 1]$ range, generating normalized RGB colours. Note that close cells are assigned similar colours.

Definition C.1 (Signed Incidence Matrices and Hodge Laplacians)**.** Given arbitrary labeling and orientation of the cells, the entry (i, j) of the first signed incidence matrix $\widetilde{\mathbf{B}}_1 \in \{\pm 1, 0\}^{|\mathcal{X}_0| \times |\mathcal{X}_1|}$ is non-zero if the j-th edge is incident to the i-th node. Moreover, it is equal to $+1$ if the orientation of the j-th edge is coherent with the orientation of the i-th node. Similarly, the entry (i, j) of the second incidence matrix $\widetilde{\mathbf{B}}_2 \in \{\pm 1, 0\}^{|\mathcal{X}_1| \times |\mathcal{X}_2|}$ is non-zero if the j-th face is incident to the i-th edge, or -1 otherwise. Moreover, it is equal to $+1$ if the orientation of the j-th face is coherent with the orientation of the i-th edge, or -1 otherwise. From the incidence information, we build the Hodge Laplacian matrices as:

$$\mathbf{L}_0 = \widetilde{\mathbf{B}}_1 \widetilde{\mathbf{B}}_1^T, \quad \mathbf{L}_1 = \underbrace{\widetilde{\mathbf{B}}_k^T \widetilde{\mathbf{B}}_k}_{\mathbf{L}_k^{\text{down}}} + \underbrace{\widetilde{\mathbf{B}}_{k+1} \widetilde{\mathbf{B}}_{k+1}^T}_{\mathbf{L}_k^{\text{down}}}, \quad \mathbf{L}_2 = \widetilde{\mathbf{B}}_2^T \widetilde{\mathbf{B}}_2. \tag{3}$$

It is clear that, similar to the incidence and adjacencies matrices from A.3, signed incidence matrices and Hodge Laplacians encode the neighborhoods of the complex. For this reason, in this work, we jointly call *neighborhood matrices* the (signed or non-signed) incidence matrices, Laplacians, and adjacency matrices.

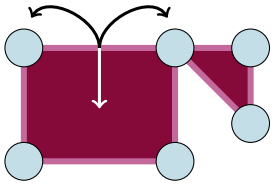
C.1 Positional encoding details

In this section, we detail and extend information presented on cellular position encodings.

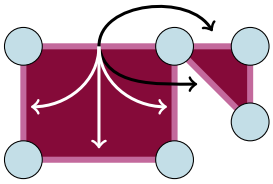
Random walks on cell complexes. Let $\mathcal{X}$ be a regular cell complex. We describe a random walk on the set of k-cells of $\mathcal{X}$. To this end, we first recall that the number of upper and lower adjacent k-cells of a given cell $\sigma \in \mathcal{X}_k$ are named respectively the $(0, k+1)$-upper and $(0, k-1)$-lower degree of σ Hernández Serrano et al. (2020),

$$\deg_U^{0,k+1}(\sigma) = \#\{\sigma' \in \mathcal{X}_k \colon \sigma \sim_U \sigma'\}; \qquad \deg_L^{0,k-1}(\sigma) = \#\{\sigma' \in \mathcal{X}_k \colon \sigma \sim_L \sigma'\}.$$

On the one hand, for each $k \geq 0$, we define a random upper k-walk based on upper adjacencies of the k-cells of $\mathcal{X}$. At each step, we move from a k-cell σ_i to any upper adjacent k-cell σ_j with probability proportional to the number of $(k+1)$-cells in common. To describe this process, we consider a weighted undirected graph G_k^{up}, whose vertices are the k-cells of $\mathcal{X}$ and the weight of each



(a) RWBSPe random walk possible transitions from the upper-left edge.



(b) RWPe random walk possible transitions from the upper-left edge.

Figure 7: Differences between RWBSPe and RWPe random walks. RWBSPe random walks can jump from a cell to all its incident and coincident cells, while RWPe random walks can jump from a cell to all its upper and lower adjacent cells.

edge (σ_i, σ_j) is the number of $(k+1)$-cells whose closure contains both cells (if a k-cell is not upper adjacent to any k-cell, we draw a loop on the corresponding vertex with weight equal to 1). Thus, the upper random k-walk is described by the left stochastic matrix $\mathbf{RW}_k^{\text{up}} = \mathbf{wA}_k^{\text{up}}(\mathbf{D}_k^{\text{up}})^{-1}$, where $\mathbf{wA}_k^{\text{up}}$ and $\mathbf{D}_k^{\text{up}}$ denote the weighted adjacency and diagonal weighted degree matrices of the graph G_k^{up}.

On the other hand, for each $k > 0$, we define a random lower k-walk through lower adjacencies of the k-cells of $\mathcal{X}$. In this case, we move from a k-cell σ_i to any lower adjacent k-cell σ_j with probability proportional to the number of $(k-1)$-faces in common. As in the previous case, the random lower walk can be described as a random walk on a weighted graph G_k^{down}, whose vertices are the k-cells of $\mathcal{X}$ and the weight of an edge (σ_i, σ_j) is set as the number of $(k-1)$-cells that both cells have in common (as before, if a k-cell is not lower adjacent to any other k-cell, then we draw a loop on it with weight equal to 1). The lower random k-walk is described by the left stochastic matrix $\mathbf{RW}_k^{\text{down}} = \mathbf{wA}_k^{\text{down}}(\mathbf{D}_k^{\text{down}})^{-1}$, where $\mathbf{wA}_k^{\text{down}}$ and $\mathbf{D}_k^{\text{down}}$ denote the corresponding weighted adjacency and diagonal weighted degree matrices of the graph G_k^{down}. The matrices $\mathbf{wA}_k^{\text{up}}$ and $\mathbf{wA}_k^{\text{down}}$ correspond respectively to the upper and lower adjacency matrices $\mathbf{A}_k^{\text{up}}$ and $\mathbf{A}_k^{\text{down}}$ with the diagonal entries in null rows replaced with 1.

We can combine both processes to obtain a random walk in which information flows through upper and lower adjacencies, in line with Schaub et al. (2020). The idea is as follows: if we are in a k-cell σ with upper and lower adjacent k-cells, we take a step with equal probability via either upper or lower connections. If σ has upper adjacent k-cells but not lower ones, we move following the random upper k-walk process, and vice versa. Lastly, if σ has neither upper nor lower connections, then we do not move.

The left stochastic matrix that describes the random k-walk is defined for $\sigma_i,\ \sigma_j \in \mathcal{X}_k$ by

$$(\mathbf{RW}_k)_{\sigma_i\sigma_j} = \begin{cases} \frac{1}{2}(\mathbf{RW}_k^{\text{up}})_{\sigma_i\sigma_j} + \frac{1}{2}(\mathbf{RW}_k^{\text{down}})_{\sigma_i\sigma_j} & \text{if } \deg_U^{0,k+1}(\sigma_j) \neq 0 \text{ and } \deg_L^{0,k-1}(\sigma_j) \neq 0 \\ (\mathbf{RW}_k^{\text{up}})_{\sigma_i\sigma_j} & \text{if } \deg_U^{0,k+1}(\sigma_j) \neq 0 \text{ and } \deg_L^{0,k-1}(\sigma_j) = 0 \\ (\mathbf{RW}_k^{\text{down}})_{\sigma_i\sigma_j} & \text{if } \deg_U^{0,k+1}(\sigma_j) = 0 \text{ and } \deg_L^{0,k-1}(\sigma_j) \neq 0 \\ \mathbb{1}(i=j) & \text{if } \deg_U^{0,k+1}(\sigma_j) = \deg_L^{(0,k-1)}(\sigma_j) = 0. \end{cases}$$

An example of the differences between transitions from an edge in the random walks described in this section and the barycentric subdivision random walks of RWBSPe are described in Fig. 7.

D Additional Details and Evaluations on MoleculeNet

D.1 Experimental Details

On exclusion of QM7, QM8, and QM9 datasets. We exclude the QM7, QM8, and QM9 datasets from the MoleculeNet suite due to their inherent dependency on 3D atomic coordinates. Our work aims to extend the capabilities of transformers to capture topological notions and our work can always be combined with a geometric one. The inclusion of this geometric information creates an incompatibility with our evaluation framework and the message we strive to convey. Concretely, these datasets derive labels (e.g., atomization energy, electronic spectra, and dipole moments) from quantum mechanical simulations that explicitly require precise interatomic distances and angles—information

not captured by our AMCC representation, which relies only on combinatorial information from the molecular structure.

D.2 HYPERPARAMETERS

We report the used hyperparameters both for the graph dataset GCB and for the MoleculeNet in Tab. 4.

Table 4: Cellular transformer hyperparameters for the GCB (left) and MoleculeNet (right) experiments. The attention type and positional encodings vary with configuration. PE stands for positional encodings.

Parameter	GCB	BBBP	Tox21	ClinTox	HIV	BACE	SIDER	MUV	FreeSolv	ESOL	Lipo
#Layers	12	8	4	4	8	3	4	8	12	12	12
Hidden dimension (d^h)	80	80	64	40	80	30	32	80	8	8	80
# Attention heads (m)	8	8	4	4	8	3	4	8	8	8	8
Hidden dimension of each head	10	10	16	10	10	10	8	10	1	1	10
Attention dropout	0.0	0.25	0.1	0.0	0.0	0.25	0.1	0.0	0.1	0.0	0.1
Embedding dropout	0.0	0.25	0.0	0.0	0.0	0.25	0.0	0.0	0.1	0.0	0.1
Readout MLP dropout	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.25
Max epochs	100	200	500	100	500	200	500	100	200	500	1000
Peak learning rate	1e−4	1e−3	1e−3	1e−4	1e−3	1e−3	1e−3	1e−4	5e−4	1e−3	1e−3
Batch size	32	64	256	64	128	64	256	512	64	256	128
Warmup epochs	10	0	0	0	0	0	0	0	0	0	0
Weight decay	1e−5	1e−2	1e−5	1e−1	1e−5	1e−2	1e−5	1e−5	1e−4	1e−4	1e−2
# hidden layers readout MLP	0	5	5	3	5	5	5	3	3	3	5
PE preprocessing	Conc.	Conc.	Conc.	Conc.	Conc.	Conc.	Conc.	Conc.	Conc.	Conc.	Conc.
Mol. descriptors used	None	MACCS	None	None	None	MACCS	MACCS	None	None	None	None

D.3 HIGHER-ORDER MOLECULAR REPRESENTATION

Molecular representation learning is typically conducted within the graph domain, where atoms are represented as nodes and bonds as edges. However, molecules exhibit higher-order structural features, such as rings, functional groups, and three-dimensional cavities, that cannot be effectively captured using simple graph-based representations. These higher-order relationships are crucial in determining molecular properties. For example, the aromaticity of a benzene ring is a property of its cyclic structure, rather than any individual atom or bond. Similarly, functional groups, such as carboxyl or hydroxyl, involve multi-atom interactions that define their reactivity and properties. Furthermore, three-dimensional cavities and pockets in proteins are critical for their binding properties and interactions with ligands.

Traditional methods, which treat each atom and bond equivalently, often overlook these subtle yet significant structural features. To address this limitation, we introduce **augmented molecular cell complexes (AMCCs)**, a novel framework that explicitly incorporates these higher-order motifs. In this approach, atoms are treated as 0-cells (nodes), bonds as 1-cells (edges), and rings or functional groups as 2-cells (faces). The augmented nature of AMCCs comes from the enhanced features assigned to these cells, allowing for a more comprehensive representation of the molecular structure compared to prior works. By explicitly incorporating these higher-order structural motifs into the molecular representation, AMCCs capture structural information that are often crucial for predicting molecular properties, as demonstrated in our experiments.

Mathematically, an augmented molecular cell complex can be defined as $\mathcal{X} = (\mathcal{X}_0, \mathcal{X}_1, \mathcal{X}_2, F_{\mathcal{X}_0}, F_{\mathcal{X}_1}, F_{\mathcal{X}_2})$. Here, $\mathcal{X}_0$ denotes the set of 0-cells (atoms), $\mathcal{X}_1$ represents the set of 1-cells (bonds), and $\mathcal{X}_2$ denotes the set of 2-cells (rings, functional groups, or other higher-order features). The corresponding feature maps $F_{\mathcal{X}_0}$, $F_{\mathcal{X}_1}$, and $F_{\mathcal{X}_2}$ represent the attributes of atoms, bonds, and higher-order features, respectively.

D.3.1 FEATURE REPRESENTATION

1. Atom-level (0-cell) Features.

Feature vector of an **atom** includes:

- **Atomic Number**: The atomic number of the atom.

- **Total Valence**: The total number of bonds to an atom.
- **Degree**: The degree of the atom, i.e., the number of neighbors it has.
- **Implicit Valence**: The number of implicit hydrogens the atom has.
- **Aromaticity**: A binary flag indicating whether the atom is part of an aromatic ring (1 if True, else 0).
- **Chiral Tag**: An integer representation of the atom's chirality.
- **Formal Charge (offset)**: The formal charge of the atom offset by a constant value (e.g., +3).
- **Hybridization**: An integer representation of the atom's hybridization state (e.g., sp2, sp3).

2. Bond-level (1-cell) Features.

Feature vector of a **bond** includes:

- **Bond Type**: An integer representation of the bond type (single, double, triple, or aromatic).
- **Conjugation**: A binary flag indicating whether the bond is conjugated (1 if True, 0 otherwise).
- **Ring Membership**: A binary flag indicating whether the bond is part of a ring (1 if True, 0 otherwise).
- **Stereo Configuration**: An integer representation of the bond's stereo configuration:
  - *No stereochemistry*
  - *Unspecified stereochemistry (cis)*
  - *Z configuration*
  - *E configuration (trans)*

  While single and triple bonds don't exhibit cis/trans or E/Z stereochemistry like double bonds, they can be involved in other types of stereochemical phenomena.
- **Rotatability**: A binary flag indicating whether the bond is rotatable (1 if rotatable, 0 otherwise).
- **Smallest Ring Size**: The smallest ring size that the bond is a part of, or 0 if not in any ring.
- **Electronegativity Difference**: The difference in electronegativity between the atoms of the bond, indicating bond polarity.
- **Hydrogen Bond Flag**: A binary flag indicating whether the bond is a hydrogen bond (1 if True, 0 otherwise).

3. Ring-level (2-cell) Features.

Feature vector of a **ring** includes:

- **Ring Size**: Number of atoms in the ring.
- **Aromaticity Flag**: A binary flag indicating whether the ring is aromatic (1 if True, 0 otherwise).
- **Heteroatom Count**: The number of non-carbon atoms in the ring.
- **Saturated-ness**: A binary flag indicating whether the ring is saturated (only single bonds, 1 if True, 0 otherwise).
- **Has Fusion**: A binary flag indicating if the ring shares any atom with another ring (1 if True, 0 otherwise).
- **Average Electronegativity**: Average electronegativity of atoms in the ring.

E Related Work

Transformers brought significant advances in various domains, including natural language processing Vaswani et al. (2017); Kenton & Toutanova (2019) or computer vision Dosovitskiy et al. (2020); Arnab et al. (2021); Han et al. (2022). Hereö we focus on graph transformers, their higher-order analogues and applications in biology / chemistry.

Graph transformers. Transformers designed to learn from data supported on graphs typically include three distinct strategies to harness the power of attention in graph contexts. The first kind integrates GNNs directly into transformers, either by stacking Wu et al. (2021); Rong et al. (2020), interweaving Lin et al. (2021), or running in parallel Zhang et al. (2020a). A second method encodes the graph structure into positional embeddings, which are then added to the input for spatial awareness Dwivedi & Bresson (2021); Hussain et al. (2021). Finally, the third approach hard codes adjacency information into the self-attention Dwivedi & Bresson (2021); Min et al. (2022b); Mialon et al. (2021). We refer the reader to Min et al. (2022a) for an extended overview. Our work adopts a combination of the second and third methods.

Higher-order transformers. Models beyond pairwise relations represent a natural progression from graph-based transformers. HOGA Bailie et al. (2024) and HONGAT Zhang et al. (2024) consider k-hop neighborhoods to develop graph attentions, but with a heuristic choice of the neighbors. The most prominent category of such higher order versions operates on hypergraphs. In many instances, they have also adopted the self-attention mechanism Kim et al. (2021); Hu et al. (2021); Zhang et al. (2020b); Wang et al. (2020a).

Transformers operating on general topological domains are scarce. To our knowledge, only two higher-order transformers targeting simplicial complexes have been proposed Clift et al. (2020); Zhou et al. (2024). While former does not consider higher order features directly, but rather leverages higher order relations to improve features on nodes, the latter Zhou et al. (2024), although proposing a fairly general object to define higher-order structures, focuses primarily on graph learning tested only on nodes and edges. A limitation of simplicial approaches is their representative power, as triangles and tetrahedra are scarce in natural data domains.

Topological deep learning in structural biology. Earlier works such as TopologyNet Cang & Wei (2017) and (Wu & Wei, 2018) benefited from persistent homology to bake the topological information in representation learning for toxicity and biomolecular property predictions, respectively. With the progress in TDL Hajij et al. (2022), convolution and message passing schemes on simplicial and cellular Bodnar et al. (2021a) structures were used to address protein classification tasks. Cell Attention Network Giusti et al. (2023) utilized liftings for molecular graph classification. Equivariant TNNs on combintorial complexes are introduced in Battiloro et al. (2025) for molecular property prediction. Recently, Mol-TDL Shen et al. (2024) has modeled polymers by a series of simplicial complexes and designed novel message passing modules.