APPENDIX

A. Symmetries

Theorem 1. *Cell Attention Networks are permutation invariant.* In literature, a GNN is permutation invariant if a permutation of nodes produces the same output without the permutation. More formally, a GNN $f(\cdot)$ taking an input graph $\mathcal{G}$ with adjacency matrix $\mathbf{A}$ and input node features matrix $\mathbf{X} = \{\mathbf{x}_i\}_{i\in\mathcal{V}}$ is (node) permutation invariant if $f(\mathbf{PAP}^T, \mathbf{PX}) = f(\mathbf{A}, \mathbf{X})$ for any permutation matrix $\mathbf{P}$. In the same way, Cell Attention Networks are permutation invariant w.r.t. permutations of nodes, edges and polygons.

Proof. We can assert, w.l.o.g., that Attentional Lift is permutation equivariant by construction. The operation $g(\cdot)$ and $a(\cdot)$ are both learnable functions acting on edges, and since $a(\cdot)$ is symmetric by definition, both $a(\cdot)$ and $g(\cdot)$ are symmetric w.r.t. to the vertices that are endpoint of the edges we are considering. This leads to have the whole function permutation equivariant.

The scheme followed in Edge Pooling is the selecting $top-k$ element of edge set referring to self-attentional coefficients γ_e. In order to select the $top-k$ elements of $\mathcal{E}$, the vector γ_e must be sorted. So no matter what is the permutation on the set, after sorting we obtain always the same result. For this reason Edge Pooling is permutation invariant.

Finally, since the proposed CAN architecture is the composition of a permutation equivariant function (i.e., the attentional lift) and a permutation invariant function (i.e., the edge pooling), it readily follows that CAN are permutation equivariant. Please notice that the proposed architecture without the pooling stage is clearly permutation equivariant. ∎

B. Experimental Details

In our experiments, we employ cell attention networks to regular cell complexes of order two obtained by applying the structural lifting map to the original graphs, i.e. we consider nodes as 0-cells and edges as 1-cells, and the chordless cycles of size up to $R = 6$ as 2-cells. In our case, each node of the original graphs is always equipped with an input feature vector. Throughout all experiments, we employ cell attention networks with the following structure. The attentional lift mechanism in Eq. (3) is given by:

$$\mathbf{h}_e^0 = \mathbf{x}_e = \Big\|_{k=1}^{F} \phi_n \underbrace{\left(\left(\left(\mathbf{a}_n^k\right)^T \left[\mathbf{x}_i || \mathbf{x}_j\right]\right) || \tilde{\mathbf{x}}_e\right)}_{a_n^k}, \quad i,j \in \mathcal{B}(e), \tag{9}$$

where $\tilde{\mathbf{x}}_e$ is the input feature vector of the edge e. If not provided by the specific benchmark, $\tilde{\mathbf{x}}_e$ can be considered as an empty vector. Also, $\mathbf{a}_n^k \in \mathbb{R}^{2F_n}$ is the vector of attention coefficients associated to the k-th feature of the input edge feature vector, and ϕ_n is the non-linear activation function for the lift layer. Please notice that the employed functions a_n^k are not symmetric, but they give the best learning performance on the proposed tasks.

The lower and upper attentional functions $a_d(\mathbf{h}_e, \mathbf{h}_k)$ and $a_u(\mathbf{h}_e, \mathbf{h}_k)$ in Eq. (4) are chosen as two independent masked self-attention schemes. They can be chosen following any of the known approaches from graphs [9], [53]. In this paper we follow the approach from [9]: formally, let

$$\omega_{e,k}^{l,d} = \phi_a\left((\mathbf{a}_d)^T \left[\mathbf{W}_d^l \mathbf{h}_e^l || \mathbf{W}_d^l \mathbf{h}_k^l\right]\right) \tag{10}$$

$$\omega_{e,k}^{l,u} = \phi_a\left((\mathbf{a}_u)^T \left[\mathbf{W}_u^l \mathbf{h}_e^l || \mathbf{W}_u^l \mathbf{h}_k^l\right]\right), \tag{11}$$

where $\mathbf{a}_d, \mathbf{a}_u \in \mathbb{R}^{2F_e}$ are two independent vectors of attention coefficients, $\mathbf{W}_d^l, \mathbf{W}_u^l \in \mathbb{R}^{F_e \times F_e}$ are two learnable linear transformations shared by the lower and upper neighbourhoods of the complex, respectively, and ϕ_a is a pointwise non-linear activation. The coefficients ω^u and ω^d in Eq. (10)(11) represent the importance of the features of edge k when exchanging messages with edge e over lower and upper neighborhoods, respectively. It worth to emphasize that since the attention schemes are decoupled, these importance coefficients will be different over the upper and lower neighborhoods.

In line with the approach of [9], we make coefficients easily comparable across different edge by normalizing them across all choices of k using the softmax function:

$$\alpha_{e,k}^{l,d} = \frac{\exp\left(\omega_{e,k}^{l,d}\right)}{\sum_{\iota\in\mathcal{N}_d^l(e)} \exp\left(\omega_{e,\iota}^{l,d}\right)} \tag{12}$$

$$\alpha_{e,k}^{l,u} = \frac{\exp\left(\omega_{e,k}^{l,u}\right)}{\sum_{\iota\in\mathcal{N}_u^l(e)} \exp\left(\omega_{e,\iota}^{l,u}\right)} \tag{13}$$

Thus, for layer l we have that $a_d(\mathbf{h}_e^l, \mathbf{h}_k^l) = \alpha_{e,k}^{l,d}$ and $a_u(\mathbf{h}_e^l, \mathbf{h}_k^l) = \alpha_{e,k}^{l,u}$. Once the attention coefficients have been normalized, to update the representation of an edge e, a linear combination of the edge features and the normalized attention coefficients corresponding to them is computed for both the lower and upper neighbourhoods and the results are aggregated alongside with the current edge representation. Formally:

$$\widetilde{\mathbf{h}}_e^l = \phi\Bigg((1+\varepsilon)\,\mathbf{W}_s^l \mathbf{h}_e^l + \sum_{k\in\mathcal{N}_d^l(e)} \alpha_{e,k}^{l,d}\,\mathbf{W}_d^l \mathbf{h}_k^l + \sum_{k\in\mathcal{N}_u^l(e)} \alpha_{e,k}^{l,u}\,\mathbf{W}_u^l \mathbf{h}_k^l\Bigg),$$

here $\mathbf{W}_s^l \in \mathbb{R}^{F_e \times F_e}$ is a shared linear transformation applied to the current hidden representation of the edges of the complex. Notice that the functions $\psi_d(\mathbf{h}_k^l)$ and $\psi_u(\mathbf{h}_k^l)$ in Eq. (4) are implemented respectively as: $\mathbf{W}_d^l \mathbf{h}_k^l$ and $\mathbf{W}_u^l \mathbf{h}_k^l$.

In the pooling layer, the hidden feature vectors are updated using Eq. (6) by scaling the features $\widetilde{\mathbf{h}}_e^l$ with the corresponding score γ_e^l:

$$\mathbf{h}_e^{l+1} = \underbrace{\phi_p\left((\mathbf{a}_p)^T \tilde{\mathbf{h}}_e^l\right)}_{\gamma_e^l} \tilde{\mathbf{h}}_e^l, \quad \forall e \in \mathcal{E}^{l+1}, \tag{14}$$

where the vector $\mathbf{a}_p$ plays the role of a collection of attention coefficient that weight the features of $\tilde{\mathbf{h}}_e^l$ to compute the corresponding score γ_e^l, which that represents the importance of edge e in the learning task. Following the approach of [35], the weight $(\mathbf{a}_p)^T\tilde{\mathbf{h}}_e^l$ of the edge e is also forwarded into a non-linear activation function ϕ_p to produce the score $\gamma_e^l \in \mathbb{R}$, which is then multiplied to $\tilde{\mathbf{h}}_e^l$ to obtain $\mathbf{h}_e^{l+1}$.

Readout operations are performed as follows: If the pooling approach is hierarchical, the readout is performed layer-wise. In particular, we choose the sum as the *permutation equivariant aggregation function* of Eq. (7) which results in a *hierarchical representation* of the complex i.e. a collection $\{\mathbf{h}_{\mathcal{C}^l}\}_{l=0}^{L-1}$ of hidden representations. Then, Eq. (8) is computed as the sum over the collection defined previously.

In the case of a global pooling, the readout is computed only in the last layer, which constitute the overall representation of the complex:

$$\mathbf{h}_{\mathcal{C}} = \mathbf{h}_{\mathcal{C}^{L-1}} = \sum_{e \in \mathcal{E}^{L-1}} \mathbf{h}_e^{L-1}. \tag{15}$$

Once $\mathbf{h}_{\mathcal{C}}$ is obtained, it is forwarded into a 2-Layer MLP with ϕ as activation function to perform the prediction.

In all layers, we adopt a Batch Normalization technique and all training operations are performed with the AdamW optimization algorithm. In Table III we report the hyper-parameters used in our experiments for each dataset.

TABLE III: Hyperparameter used for the experiments on TUDatasets.

Parameter	MUTAG	PTC	PROTEINS	NCI1	NCI109
Lift Heads	1	32	256	128	128
Lift Activation	*ELU*	*ELU*	*ELU*	*ELU*	*ELU*
Lift Dropout	0.0	0.0	0.05	0.2	0.2
Hidden Features	$[32, 32]$	$[32, 32]$	$[128, 128]$	$[32, 32, 32, 32]$	$[32, 32, 32, 32]$
Attention Heads	$[1, 1]$	$[2, 2]$	$[1, 1]$	$[4, 4, 4, 4]$	$[4, 4, 4, 4]$
Attention Aggregation	-	*cat*	-	*cat*	*cat*
Attention Activation	*LReLU*	*LReLU*	*Tanh*	*Tanh*	*Tanh*
Activation	ELU	ELU	Tanh	ELU	ELU
MLP Neurons	8	4	128	256	32
Batch Size	64	128	128	128	128
Neg. Slope	0.1	0.1	0.3	0.08	0.07
Pool Ratio	1.0	0.75	0.6	0.5	0.75
Pool Type	*Hier.*	*Glob.*	*Hier.*	*Glob.*	*Glob.*
Dropout	0.1	0.6	0.3	0.15	0.05
Learning Rate	$3e-3$	$1e-3$	$3e-3$	$3e-4$	$3e-3$