Cell Attention Networks

Lorenzo Giusti¹, Claudio Battiloro², Lucia Testa¹, Paolo Di Lorenzo², Stefania Sardellitti², Sergio Barbarossa²
¹ DIAG Department, Sapienza University of Rome, Rome, Italy
² DIET Department, Sapienza University of Rome, Rome, Italy
{lorenzo.giusti, claudio.battiloro, lucia.testa, paolo.dilorenzo, stefania.sardelliti, sergio.barbarossa}@uniroma1.it

Abstract—Since their introduction, graph attention networks achieved outstanding results in graph representation learning tasks. However, these networks consider only pairwise relations between features associated to the nodes and then are unable to fully exploit higher-order and long-range interactions present in many real world data-sets. In this paper, we introduce a neural architecture operating on data defined over the nodes and the edges of a graph, represented as the 1-skeleton of a regular cell complex, able to capture insightful higher-order and long-range interactions. In particular, we exploit the lower and upper neighborhoods, encoded in the cell complex, to design two independent masked self-attention mechanisms, thus generalizing the conventional graph attention strategy. The approach used is hierarchical and it incorporates the following steps: i) a lifting algorithm that learns (additional) edge features from node features; ii) a cell attention mechanism to find the optimal combination of edge features over both lower and upper neighbors; iii) a hierarchical edge pooling mechanism to extract a compact meaningful set of features. The experimental results show that this method compares favorably with state of the art results on graph-based learning tasks while maintaining a low complexity.

Index Terms—Topological deep learning, geometric deep learning, attention networks, cell complexes.

I. INTRODUCTION

Graph Neural Networks (GNNs) find applications in a plethora of fields, like computational chemistry [1], social networks and physics simulations. Since their introduction [2], GNNs have shown remarkable results in learning tasks when data are defined over a graph domain, where the flexibility of neural networks is coupled with prior knowledge about data relationships, expressed in terms of the underlying topology. The literature on GNNs is large and numerous techniques have been studied, usually categorized in spectral [3] and non-spectral methods [4]. Generally speaking, the idea is learning representations of node attributes using local aggregation, where the neighborhood is formally represented by the graph topology. By leveraging this basic but powerful idea, outstanding performance has been achieved in many traditional tasks such as classification for nodes or entire graphs [4] or link prediction [5] as well as more specialized ones such as protein folding [6] and neural algorithmic reasoning [7]. At the same time, a major performance boost to deep learning algorithms has been offered by the inclusion of attention mechanisms, introduced to handle sequence-based tasks [8], enabling for variable sized inputs, to concentrate on their most important features. Then, pioneering works introduced graph attention networks [9] achieving state-of-the-art results in most of the aforementioned tasks. Graphs can also be seen as a simple instance of a topological space, able to capture pairwise interactions through the presence of an edge between any pair of directly interacting nodes. However, despite their overwhelming popularity, graph-based representations are unable to consistently model higher order relations, which play a crucial role in many practical applications. Examples where multiway relations cannot be reduced to an ensemble of pairwise relations are network neuroscience [10] or complex systems [11], where some reactions occur only when a set of multiple (not only two) entities interact. To overcome the limitations of graph representations, more general architectures defined on hypergraphs have been proposed [12]–[14], including structures incorporating attention mechanisms for hypergraphs neural networks [15], [16]. In the meanwhile, pioneering works on Topological Signal Processing (TSP) [17], [18] demonstrated the benefits of processing signals defined on simplicial and cell complexes, which are specific examples of hyper-graphs having a rich algebraic description, that can easily encode multi-way relationships hidden in the data in a low-complexity fashion. Within the last few years, there has been an increasing interest to exploit the benefits of TSP to perform learning tasks over graph-structured data [19]–[24]. In particular, learning over simplicial complexes has been investigated to perform missing data imputation over citation networks [19], trajectory prediction [25], and graph representation learning tasks in [20], using a principled framework. By their nature, simplicial complexes suffer from the rigidity of their defining inclusion property: if, for example, a multi-way relation among four nodes is encoded in the complex, then all possible triple-wise relations among them must also be present. To relax this constraint, while maintaining a topological representation, in [22] the simplicial complexes have been replaced with cell complexes. Moreover, the authors of [21] extended the work of [20] using cell complexes as the underlying topological domain. Finally, very recently, the work in [23] introduced attentional architectures operating on combinatorial complexes (see Sec. II for a detailed comparison with our work). Contribution. In this work, we exploit the advantages of cell complexes and masked self-attention to introduce cell attention networks, a fully attentional low-complexity architecture able to learn from data defined over the nodes and the edges of a graph incorporated within a cell complex in order to extract higher-order and long-range interactions within the data. The intuition is to provide an attention mechanism that handles relations among nearby edges, where the neighborhood is formally represented by a cell complex. To this end, we exploit a hierarchical approach that...
lifts up the node features to derive (additional) edge features and then it computes the attention coefficients between nearby edges to find the optimal way to combine them. In particular, we devise a multi-head attentional lift mechanism that learns the data over the edges of the complex leveraging a self-attention mechanism over the features of the nodes that are on their boundaries. Consequently, our architecture is also equipped with a cellular lifting map algorithm that embeds the graph domain into a regular cell complex (CW). Other graph lifting techniques have been used in previous works, such as clique complexes, or more sophisticated structures based on incidence tensors [26]. Moreover, the proposed architecture presents nice features of explainability due to its fully attention-driven design: by simply inspecting the attention coefficients, it is possible to understand the contribution of each cell to the learning task. For instance, in computational chemistry, the upper attention coefficients may represent the importance of the interaction between two atoms within a molecular ring, in network traffic problems, the attention coefficients might indicate obsolete links, in social networks they might help to detect communities.

II. RELATED WORKS

Despite Topological Deep Learning is an emerging research area that has been introduced quite recently, numerous pioneering works appeared in this field. In [20], message passing neural networks (MPNNs) [1] were adapted to simplicial complexes (SCs), and Simplicial Weisfeiler-Lehman (SWL) colouring procedure for distinguishing non-isomorphic SCs was introduced. The architectures in [20], namely Message Passing Simplicial Networks (MPSNs), are a generalization of a Graph Isomorphism Network (GIN) [27]. In [22], the authors took the underlying space topology into account and generalized the message passing mechanism over cell complexes. The previous strategy was then refined in [21], where the authors introduced CW Networks (CWNs), proven to be not less powerful than the 3-WL test, and able to perform representation learning on graph-structured data. The works in [28], [29] introduced Neural Sheaf Diffusion Models, neural architectures grounded in the theory of cellular sheaves able to improve learning performance on graph-based tasks, especially in heterophilic settings. In [30], the authors introduced Tangent Bundle Neural Networks, i.e. deep convolutional architectures able to process data defined on the tangent bundle of Riemann manifolds, showing that Neural Sheaf Models can be obtained as a discretization of the underlying manifold network. In [31], a novel attention neural architecture operating on data defined on simplicial complexes leveraging masked self-attention layers was introduced, taking into account lower and upper neighbourhoods and introducing a proper technique to process the harmonic component of the data based on the design of a sparse projection operator. A similar architecture was proposed in [32], which re-weights the interactions between neighbouring simplices through an orientation equivariant attention mechanism. However, none of these works considered masked self-attention mechanisms for architectures designed to handle data defined over cell complexes. Finally, the work in [23] introduced a broad class of attentional architectures operating on generalized higher-order domains called combinatorial complexes. Although the work in [23] proposed general high-order attention mechanisms, crucial differences exist with our work. First of all, the high-order attention block proposed in [23, eq. (3)] is induced by a single adjacency (or incidence) operator, which leads to a single attention mechanism per each block, whereas our architecture exploits two attention mechanisms (cf. 4), working on upper and lower neighborhoods of the cell, per each attentional block. Secondly, we perform node features lifting in a different way with respect to [23]. In particular, we exploit an attentional lifting mechanism that learns edge features from node features exploiting a multi-head attention procedure, whereas the inter-order attention block in [23, eq. (6)] performs lifting implementing learnable weighted sums of boundary element features. Thirdly, our attentional message passing scheme allows for more general aggregation operators, while the attentional block in [23] exploits only sum aggregation functions. Finally, our architecture is endowed with an attentional pooling layer, which performs a pooling operation over the cells of the complex, whereas the method in [23, Appendix D] considers a very simple pooling operation given by summation of embeddings. In general, our architecture is designed to exploit higher-order interactions (at the edge level) induced by the cell complex in order to solve learning tasks from data defined over graphs with low-complexity. On the other side, the architecture in [23] is very general since it operates at all cell-orders in a modular fashion, thus applying to a broader class of learning tasks.

III. BACKGROUND

In this section we recall the basics of regular cell complexes, which are topological spaces enabling an efficient representation of high-order interaction systems, generalizing graphs and simplicial complexes.In particular, we first introduce the definition of a regular cell complex and then we recall a few additional properties enabling the representation of cell complexes via boundary operators.

Definition 1 (Regular cell complex) [21], [33]. A regular cell complex is a topological space $C$ together with a partition \( \{X_\sigma\}_{\sigma \in P_C} \) of subspaces $X_\sigma$ of $C$ called cells, where $P_C$ is the indexing set of $C$, such that

1. For each $c \in C$, every sufficient small neighborhood of $c$ intersects finitely many $X_\sigma$;
2. For all $\tau, \sigma$ we have that $X_\tau \cap X_\sigma \neq \emptyset$ iff $X_\tau \subseteq X_\sigma$, where $\overline{X_\sigma}$ is the closure of the cell;
3. Every $X_\sigma$ is homeomorphic to $\mathbb{R}^k$ for some $k$;
4. For every $\sigma \in P_C$ there is a homeomorphism $\phi$ of a closed ball in $\mathbb{R}^k$ to $\overline{X_\sigma}$ such that the restriction of $\phi$ to the interior of the ball is a homeomorphism onto $X_\sigma$.

Condition (2) implies that the indexing set $P_C$ has a poset structure, given by $\tau \leq \sigma$ iff $X_\tau \subseteq X_\sigma$. This is known as the face poset of $C$. The regularity condition (4) implies that all topological information about $C$ is encoded in the poset structure of $P_C$. Then, a regular cell complex can be identified
with its face poset. For this reason, from now on we will indicate the cell \( X_\sigma \) with its corresponding face poset element \( \sigma \). The dimension \( \dim(\sigma) \) of a cell \( \sigma \) is \( k \).

**Definition 2 (k-skeleton).** A \( k \)-skeleton of a cell complex \( C \), denoted \( C^{(k)} \), is the subcomplex of \( C \) consisting of cells of dimension at most \( k \).

From Definitions 1 and 2, it is easy to check that the 0-skeleton of a cell complex is the set of its nodes \( C^{(0)} = V \) and the 1-skeleton is the underlying graph \( C^{(1)} = G(V, E) \), where \( E \) is the edge set; we refer to 2-cells as polygons. Regular cell complexes can be described via an incidence relation with a reflexive and transitive closure that is consistent with the partial order introduced in Definition 1. The boundary relation describes which cells are on the boundary of other cells.

**Definition 3 (Boundary relation).** We have the boundary relation \( \sigma \prec \tau \) iff \( \dim(\sigma) \leq \dim(\tau) \) and there is no cell \( \delta \) such that \( \sigma \leq \delta \leq \tau \).

We can use the previous definitions to define the four types of (local) adjacencies present in cell complexes, following the approach from [21]:

**Definition 4 (Cell complex adjacencies) [21].** For a cell complex \( C \) and a cell \( \sigma \in \mathcal{P}_C \), we define:

- The boundary adjacent cells \( \mathcal{B}(\sigma) = \{ \tau \mid \tau \prec \sigma \} \), are the lower-dimensional cells on the boundary of \( \sigma \). For instance, the boundary cells of an edge are its nodes and the boundary cells of a polygon are its edges.

- The co-boundary adjacent cells \( \mathcal{C}(\sigma) = \{ \tau \mid \sigma \prec \tau \} \), are the higher-dimensional cells with \( \sigma \) on their boundary. E.g., the co-boundary cells of a node are the edges having that node as an endpoint and the co-boundary of an edge are the polygons having that edge as one of its sides.

- The lower adjacent cells \( \mathcal{N}_l(\sigma) = \{ \tau \mid \exists \delta \text{ such that } \delta \prec \sigma \text{ and } \delta \prec \tau \} \), are the cells of the same dimension as \( \sigma \) that share a lower dimensional cell on their boundary. The line graph adjacencies between the edges are a classic example of this.

- The upper adjacent cells \( \mathcal{N}_u(\sigma) = \{ \tau \mid \exists \delta \text{ such that } \sigma \prec \delta \text{ and } \tau \prec \delta \} \). These are the cells of the same dimension as \( \sigma \) that are on the boundary of the same higher-dimensional cell as \( \sigma \).

Finally, given a graph \( G \), it is possible to build a higher order cell complex \( C^G \) on the graph by associating filled polygons to closed paths of edges having no internal chords.

### A. Data over Cell Complexes

Let \( D_k \) be the set of \( k \)-th order cells in a cell complex \( C \), with \( |D_k| = N_k \). In most of the cases the focus is on complexes of order up to two, thus a set of nodes \( V \) with \( |V| = V \), a set of edges \( E \) with \( |E| = E \) and a set of polygons \( \mathcal{P} \) with \( |\mathcal{P}| = P \) are considered, resulting in \( D_0 = V \) (cells of order 0), \( D_1 = E \) (cells of order 1) and \( D_2 = \mathcal{P} \) (cells of order 2). In Fig. 1 we sketch an example of a cell complex of order 2.

A \( k \)-cell signal is defined as a collection of mappings from the set of all \( k \)-cells contained in the complex to real numbers:

\[
x_k = [x_k(\sigma_k^1), \ldots, x_k(\sigma_k^1), \ldots, x_k(\sigma_k^1)] \in \mathbb{R}^{N_k},
\]

where \( x_k : \mathcal{D}_k \rightarrow \mathbb{R} \). The order of the signal is one less the cardinality of the elements of \( C \). Therefore, for a complex \( C^{(2)} \), the \( k \)-cell signals are defined as the following mappings:

\[
x_0 : V \rightarrow \mathbb{R}, \quad x_1 : E \rightarrow \mathbb{R}, \quad x_2 : \mathcal{P} \rightarrow \mathbb{R},
\]

representing node, edge and polygon signals, respectively.

In this work, we will consider only node and edge signals. In particular, we will refer to an instance of the former as \( x_i \) while to an instance of the latter will be referred as \( x_a \).

### IV. Cell Attention Networks

The aim of this work is to extend Graph Attention Networks introduced in [9] to account for multi-way relationships, i.e., performing a masked self-attention mechanisms at the edge level. The proposed hierarchical architecture, which we refer to as Cell Attention Network (CAN) (Fig. 4), starts with the embedding of the input graphs in regular cell complexes via a skeleton-preserving cellular lifting map and an attentional lift procedure enabling the derivation of edge features from node features. Then, we introduce a novel edge-level attentional message passing scheme. After each round of message passing, we perform a novel edge pooling operation and a local readout to reduce complexity; finally, after the last message-passing round, a global readout is applied. As in the previous sections, we denote the input graph(s) with \( G = (V, E) \) and the input node features of node \( i \in V \) with \( x_i \in \mathbb{R}^{F_n} \).

#### A. Cellular Lifting Map

We first need to incorporate input graphs in regular cell complexes. To address this challenge, we exploit the notion of skeleton-preserving cellular lifting map presented in [21].

**Definition 5 (Skeleton-Preserving Cellular Lifting Map).** A cellular lifting map \( s : G \rightarrow C^G \) is a skeleton preserving function that incorporates a graph \( G \) into a regular cell complex \( C^G \), such that, for any graph \( G \), the 1-skeleton (i.e., the underlying graph) of \( s(G) \) and \( G \) are isomorphic.

Informally, Definition 5 just requires that the lifting map keeps the underlying graph structure unchanged. Several cellular
lifting map can be exploited, in this work we opted for a lifting map that attach cells to all the induced (or chordless) cycles, where \( k \) can be considered a hyperparameter to be chosen arbitrarily, and which controls the maximum size of the polygons (2-cells) of the complex.

**Computational Complexity and Learnable Parameters:** Although this operation can be pre-computed for the entire dataset and the connectivity results stored for a later usage, it worth to elicit its complexity noticing that for some applications the storage of the upper and lower connectivity for the *entire* dataset might be not possible. We considered Cellular Lifting Maps that assign 2-cells to all the chordless cycles of a graph with a maximum number of nodes in the cycles up to \( R \) as maximum cycle size. The chord-less cycles in a graph can thus be enumerated in \( O((|E| + |V||R|) \text{polylog}|V|) \) time [34]. Similar to [21], in our experimental setup we have that \( R \) can upper bounded by a small constant. Thus, the complexity of this operation can be approximated to be linear in the size of the complex. Intuitively this operation does not involve any parameter to be learned during the network’s training phase due to the fact that lifting the input graph \( \mathcal{G} \) to a cell complex \( \mathcal{C}_G \) is an operation that assigns a cell \( \sigma \) to all the chord-less cycles of \( \mathcal{G} \) up to a maximum cycle size \( R \). Thus, the number of learnable parameters for the lift is \( O(1) \).

**B. Attentional Lift**

After the Cellular Lifting Map, we need to learn edge features, thus performing a lift operation on the node features that we refer to as attentional lift. To this aim, we exploit a masked multi-head self-attention mechanism [9]. The procedure is based on the computation of \( F_e \) attention heads such that, for each pair of nodes \( i, j \in \mathcal{V} \) connected by an edge \( e \in \mathcal{E} \), the corresponding edge features \( x_e \in \mathbb{R}^{F_{e}} \) are given by the concatenation of the resulting attention scores.

**Definition 6 (Attentional Lift).** An Attentional Lift is a learnable function \( g : \mathbb{R}^{F_{e}} \times \mathbb{R}^{F_{e}} \rightarrow \mathbb{R}^{F_{e}} \), of the form:

\[
x_e = g(x_i, x_j) = \left\| a_k^e(x_i, x_j), \; \forall e \in \mathcal{E}. \right.
\]

where \( a_k^e : \mathbb{R}^{F_{e}} \times \mathbb{R}^{F_{e}} \rightarrow \mathbb{R} \) is the \( k \)-th learnable attention function shared across nodes, and \( || \) is the concatenation operator. Since the order of the nodes connected by an edge should not change the corresponding lifted edge features, we assume that the functions \( a_k^e \) are symmetric. It might happen that data comes naturally with edge features. In that case, we concatenate them to \( x_e \) and consider \( F_e \) as the sum of the number of learned features and the provided ones.

**Computational Complexity and Learnable Parameters:** The complexity of this operation consists of a multi-head attention message passing scheme over the entire graph [9]. For a single node pair \( i, j \in \mathcal{V} \) connected by an edge \( e \in \mathcal{E} \), the attentional lift defined in Eq. (3) can be decomposed into \( F_e \) independent self-attention schemes. Each attention scheme requires \( O(F_{e}) \) computations, where \( F_{e} \) is the number of input node features. Thus, for the pair \( i, j \), the attentional lift is performed in \( O(F_{e}F_{n}) \), where \( F_{e} \) is a parameter to be chosen as the number of input edge features. Accounting all the edges of the complex yields an amount of \( O(|\mathcal{E}|F_{e}F_{n}) \) operations to lift node features into edge ones. In the context of lifting a pair of graph node features \( x_i, x_j \in \mathbb{R}^{F_{e}} \) to obtain edge features \( x_e \in \mathbb{R}^{F_{e}} \), we have to learn the attention functions parameters. In the case of the usual GAT-like attention functions [9], we need to learn a vector of attention coefficients \( a_n \in \mathbb{R}^{2F_{e}} \) for each edge feature. The vector \( a_n \) has a number of learnable parameters in \( \Theta(F_{e}) \).

**C. Cell Attention**

In this section we introduce the cell attention mechanism, i.e., an attentional message passing scheme operating at edges level on the learned edges features of Eq. (3), exploiting the connectivity given by the regular cell complex \( \mathcal{C}_G \) computed via the cellular lifting map of Definition 5. Before describing the proposed scheme, please notice that, as previously introduced, we will perform an edge pooling operation after the message-passing round at each layer \( l \in \{1, \ldots, L\} \), meaning that the architecture will produce a sequence of cell complexes \( \{\mathcal{C}_l\}_l \) such that \( \mathcal{C}_l+1 \subseteq \mathcal{C}_l \) (due to the fact that the corresponding edge sets are such that \( \mathcal{E}_{l+1} \subseteq \mathcal{E}_l \)); we will describe the edge pooling in details in the next section.

As already introduced in Section III, there are various types of adjacencies that can be exploited when dealing with cell complexes. In this work, we employ only the upper and lower neighbourhoods. This choice is due to the fact that the upper neighbourhood allows to capture (possibly) long-range relationships within the cells of the complex while the lower neighbourhood maintains local information. Therefore, this approach can be seen as a broad-wise learning of the interactions, while, for example, in [21] the approach is bottom-up. Moreover, as explained later in the section, our approach keeps the number of operations to be linear in the initial number of edges of the complex, which can be intended as a favorable trade-off between complexity and performance. In particular, since the message exchange happen at the edges level, in each layer \( l \), our message-passing scheme exploits upper and lower edge adjacencies \( \mathcal{N}_u^l(e) \) and \( \mathcal{N}_l^l(e) \), associated with the cell complex \( \mathcal{C}_l \), \( l = 1, \ldots, L \). At each layer \( l \), an upper attention function \( a_n : \mathbb{R}^{F_{e}} \times \mathbb{R}^{F_{e}} \rightarrow \mathbb{R} \), responsible to evaluate the reciprocal importance of two edges that are part of the same polygon, and a lower attention
wise nonlinear activation (Eq. (4)). For a layer $l$ where $\psi$ is cell attention, the complexity receives an overhead induced by the number of messages that and $e_e$ receives is bounded by $O(|E_e|)$. The same computation extends to all edges of the complex via a pooling learnable attention function $a_d : \mathbb{R}^{F_e} \times \mathbb{R}^{F_e} \rightarrow \mathbb{R}$, responsible to evaluate the reciprocal importance of two edges that share a common node, are introduced. Therefore, edges embedding are updated in the $l$-th message passing round as:

$$\tilde{h}_e^l = \phi \left( h_e^l, \bigoplus_{k \in N^l_d(e)} a_d(h_e^l, h_k^l) \psi_d(h_k^l), \bigoplus_{u \in N^l_u(e)} a_u(h_e^l, h_u^l) \psi_u(h_u^l) \right),$$

where $\bigoplus$ is any permutation invariant aggregation function (e.g., sum, mean, max, ...), $\phi$ is a (possibly) learnable function, $\psi_u$ and $\psi_d$ are learnable functions sharing the weights with $a_u$ and $a_d$, respectively. $h_e^0 = x_e$, $C^0 = C_d$ (thus $E^0 = E$). Obviously, multi-head attention can be trivially injected following the usual concatenation or averaging approach [9]. A pictorial example of upper and lower attention is depicted in Fig. 2.

Computational Complexity and Learnable Parameters: This operation consists in two independent masked self-attention message passing schemes over the upper and lower neighbourhoods of the complex, namely cell attention, an inner linear transformation of the edges’ features and an outer pointwise nonlinear activation (Eq. (4)). For a layer $l$, we have that the number of messages that an edge $e$ receives from its lower neighbourhood is equal to $|N^l_d(e)|$, the number of edges that share a common node with $e$. The same computation yields for the upper neighbourhood: we have that the edge $e$ receives $|N^l_u(e)|$ messages, from edges that are in the same cell’s boundaries as $e$. Recalling that $E^{l+1} \subseteq E^l$ and $R$ is upper bounded by a small constant [21], we have that in a single message passing the number of messages that and edge $e$ receives is bounded by $O(|E^0|)$. The inner linear transformation that propagates the information contained in $h_e^l$ is upper bounded by $O(F_e^2)$. Extending this to all edges of the complex, we have that the complexity of a cell attention layer can be rewritten as $O(|E^0| F_e^2)$. In the case of a multi-head cell attention, the complexity receives an overhead induced by the number of attention heads involved within the layer, i.e., a multiplication by a factor $H_e$, the number of cell attention heads. In terms of learnable parameters and in the case of the usual GAT-like attention functions [9], a single cell attention layer is composed of: two independent vectors of attention coefficients $a_d, a_u \in \mathbb{R}^{2F_e}$ for properly weighting the lower and upper neighbourhoods, respectively. Moreover, the layer is equipped with three linear transformations, $W^l_d, W^l_u, W^l_e \in \mathbb{R}^{F_e \times F_e}$, acting respectively on: $h_e^l$, the hidden feature vector of edge $e$ at layer $l$ and the hidden feature vectors $h_k^l$ in the lower and upper neighbourhoods of the edge $e$. Thus, the number of learnable parameters of a cell attention layer is $O(F_e^2)$.

D. Edge Pooling

In this section, we present a self-attention edge pooling technique, adopting a variation of the method used in [35]. Let $\tilde{h}_e^l \in \mathbb{R}^{F_e}$ be the hidden feature vector associated to edge $e$ obtained via the attentional message-passing from Eq. (4) after the $l$-th message-passing round. The edge attention pooling operation consists in computing a self-attention score $\gamma_e^l \in \mathbb{R}$ for each edge of the complex via a pooling learnable attention function $a_p : \mathbb{R}^{F_e} \rightarrow \mathbb{R}$:

$$\gamma_e^l = a_p \left( \tilde{h}_e^l \right), \quad \forall e \in E^l. \tag{5}$$

Let $k \in (0, 1]$ be the pooling ratio, i.e., the fraction of the edges that will be retained over the number of edges in input to the self-attention edge pooling layer. At this point, we keep the $\lceil k|E^l| \rceil$ edges belonging to the set $E^{l+1} = \{ e \in E^l \} \cap \gamma_e^l \in \text{top-}k \{ \gamma_e^l \}_{e \in E^l}, \lceil k|E^l| \rceil \subseteq E^l$ where top-$k \{ \gamma_e^l \}_{e \in E^l}, \lceil k|E^l| \rceil$ is the set of the highest $\lceil k|E^l| \rceil$ self-attention scores. Finally, the feature vectors that will be kept after the pooling stage are scaled as:

$$h_e^{l+1} = \gamma_e^l \tilde{h}_e^l, \quad \forall e \in E^{l+1}. \tag{6}$$

After the edge pooling, we consequently need to adjust the structure of the cell complex $C^l$ to obtain a consistent updated complex $C^{l+1}$. To this aim, we apply the procedure depicted in Fig. 3: If an edge $e$ belongs to $E^l$ but is not contained in $E^{l+1}$, the lower connectivity is updated by disconnecting the nodes that are on the boundary of $e$, while the upper connectivity is updated by removing the polygons that have $e$ on their
with edge features, we concatenate them to the result of the lift layer (Eq. (3)). We included small molecules with class labels such as MUTAG [38] and PTC [39]. In the former dataset, the task is to identify mutagenic molecular compounds for potentially commercial drugs, while in the latter the goal is to identify chemical compounds based on their carcinogenicity in rodents. The PROTEINS dataset [40] is composed mainly by macromolecules. Here, nodes represent secondary structure elements and are annotated by their type. Nodes are connected by an edge if the two nodes are neighbours on the amino acid sequence or one of three nearest neighbors in space; the task is to understand if a protein is an enzyme or not. Using these type of data in a Cell Complex based architecture has an underlying importance since molecules have polyadic structures. Finally, NCI1 and NCI109 are two datasets aimed at identifying chemical compounds against the activity of non-small lung cancer and ovarian cancer cells [41]. Considering the aforementioned datasets, we compare CAN with other state of the art techniques in graph representation learning. Since there are no official splits for the training and test sets, to validate the proposed architecture, we followed the method used in [21]: we run a 10-fold cross-validation reporting the maximum of the average validation accuracy across folds. The performance of CAN is reported in Table II, along with those of graph kernel methods: Random Walk Kernel (RWK, [42]), Graph Kernel (GK, [43]), Propagation Kernels (PK, [44]), Weisfeiler-Lehman graph kernels (WLK, [45]); other GNNs: Diffusion-Convolutional Neural Networks (DCNN, [46]), Deep Graph Convolutional Neural Networks (DGCNN, [47]), and a perceptron (MLP) if needed for the learning task. To compute the computational complexity of this layer it is linear in the dimension of the complex, the overall complexity of this layer in can be upper bounded by the sorting algorithm, i.e., \( \mathcal{O}(|E| \log(|E|)) \). For this layer, learnable parameters are employed only in computing the self-attention scores (Eqs. (5)). In the case of the usual GAT-like attention functions, they consist of a shared vector of attentional scores’ coefficients \( \mathbf{a}_p \in \mathbb{R}^{F_x} \), similarly to the lift layer, leading to \( \Theta(F_x) \).

V. EXPERIMENTAL RESULTS

In this section we assess the performance of the proposed architecture on several real-world graph classification problems, focusing on the well known molecular benchmarks TUDataset [37]. In every experiment, if the dataset is equipped with edge features, we concatenate them to the result of the

\[ h_{\mathcal{C}^{l+1}} = \bigoplus_{e \in E^{l+1}} h_{e^{l+1}}^{e^{l+1}}. \]  

\[ h_{\mathcal{C}} = \bigoplus_{l} h_{\mathcal{C}^l}. \]

Finally, the result of the final aggregation is fed to a multi-layer perceptron (MLP) if needed for the learning task. **Computational Complexity and Learnable Parameters:** The operations involved in the pooling layer can be decomposed in: (i) computing the self-attention scores for each edge of the complex (\( \gamma_e^{l} \) in Eq. (5)); (ii) select the highest \( \lceil k |E^l| \rceil \) values from a collection of self-attention scores (top-k\{\( \gamma_e^{l} \)\}_\( e \in E^l \), \( \lceil k |E^l| \rceil \)); and (iii) adjust the connectivity of the complex (see Fig. 3). To compute the computational complexity of this layer it is convenient to see the selection operation as a combination of a sorting algorithm over a collection of self-attention scores and a selection of the first \( \lceil k |E^l| \rceil \) elements from the sorted collection. Since the computations involved in (i) and (iii) are linear in the dimension of the complex, the overall complexity of this layer in can be upper bounded by the sorting algorithm, i.e., \( \mathcal{O}(|E^l| \log(|E^l|)) \). For this layer, learnable parameters are employed only in computing the self-attention scores (\( \gamma_e^{l} \), Eq. (5)). In the case of the usual GAT-like attention functions, they consist of a shared vector of attentional scores’ coefficients \( \mathbf{a}_p \in \mathbb{R}^{F_x} \), similarly to the lift layer, leading to \( \Theta(F_x) \).

V. EXPERIMENTAL RESULTS

In this section we assess the performance of the proposed architecture on several real-world graph classification problems, focusing on the well known molecular benchmarks TUDataset [37]. In every experiment, if the dataset is equipped with edge features, we concatenate them to the result of the

\[ h_{\mathcal{C}^{l+1}} = \bigoplus_{e \in E^{l+1}} h_{e^{l+1}}^{e^{l+1}}. \]  

\[ h_{\mathcal{C}} = \bigoplus_{l} h_{\mathcal{C}^l}. \]
Table II: Experimental results on TUDatasets. The first part shows the accuracy of graph kernel methods, while the second assess graph neural networks.

| Dataset | MUTAG | PTC | PROTEINS | NCI1 | NCI109 |
|---------|-------|-----|----------|------|--------|
| RWK     | 79.2±2.1 | 55.9±1.5 | 59.6±1.0 | N/A  | N/A    |
| GIN     | 81.4±1.7 | 57.5±0.5 | 71.4±0.3 | 62.5±0.3 | 62.4±0.3 |
| PK      | 76.0±2.7 | 59.5±2.4 | 73.7±0.7 | 82.5±0.5 | N/A    |
| WLC     | 90.4±5.7 | 89.9±4.3 | 75.0±2.1 | 86.3±1.8 | N/A    |
| DCCN    | N/A    | N/A | 61.3±1.6 | 56.6±1.0 | N/A    |
| DGCNN   | 85.8±1.8 | 58.6±2.5 | 75.5±0.9 | 74.4±0.5 | N/A    |
| IGN     | 83.9±13.0 | 58.5±6.9 | 76.6±5.5 | 74.3±2.7 | 72.8±1.5 |
| GIN     | 89.4±5.6 | 64.6±7.0 | 76.2±2.8 | 82.7±1.7 | N/A    |
| PPGNs   | 90.6±8.7 | 66.2±6.6 | 77.2±4.7 | 83.2±1.1 | 82.2±1.4 |
| NNG     | 89.4±16 | 66.8±17 | 71.7±10 | 82.4±1.3 | N/A    |
| GSNN    | 92.2±7.5 | 68.2±7.2 | 76.6±5.0 | 83.5±2.0 | N/A    |
| CCNN    | 92.2±4.7 | 65.6±2.9 | 75.3±2.8 | 85.2±0.8 | 76.0±6.9 |
| SIN     | N/A | N/A | 76.4±3.3 | 82.7±2.1 | N/A    |
| CIN     | 92.7±6.1 | 68.2±7.2 | 77.0±4.3 | 83.6±1.4 | 84.0±1.6 |
| CAN     | 94.2±3.9 | 73.8±7.8 | 78.8±1.8 | 84.9±1.8 | 83.8±1.4 |

Network (DGCNN, [47]), Invariant and Equivariant Graph Networks (IGN, [48]), Graph Isomorphism Networks (GIN, [27]), Provably Powerful Graph Networks (PPGNs, [49]), Natural Graph Networks (NGN, [50]), Graph Substructure Network (GSN [51]) and topological networks: Convolutional Cell Complex Neural Networks (CCCN [22]), Simplicial Isomorphism Network (SIN, [20]), Cell Isomorphism Network (CIN, [21]). As we can see from Table II, CAN achieves the best performance on four out of five benchmarks, while performing very similarly to CIN in the last experiment (i.e., NCI109). Since CAN has a much lower computational complexity than CIN, these results support the validity and the performance obtained of the proposed architecture.

A. Ablation Study

In this section we take a detailed look at the performance of each operation involved in cell attention networks by performing different ablation studies and show their individual importance and contributions. We followed the same experimental setup used in Section V by fixing the hyper-parameters and removing one-by-one the cell attention network operations: removing the lift refers to assign a feature \( x_i \) to an edge \( e \) using a linear function that takes the feature vectors \( x_i, x_j \) from the nodes \( i, j \in B(e) \), i.e. a simple scalar product between \( x_i \) and \( x_j \) (\( x_i = \langle x_i, x_j \rangle \)). Removing the attention can be intended as setting \( a_{u} \) and/or \( a_{d} \) to yield the coefficients of the upper and lower Laplacians indexed by the label of their input. The case in which both attentions are removed can be seen as a particular implementation of the cell complex neural architecture proposed in [22]. Removing the pooling means to detach the pooling layer from the network and remove eventual intermediate readout computations involved in the hierarchical pooling setup. As shown in Figure 5, we observe a decrease in the overall performance when removing parts of the cell attention network architecture as expected. Of particular interest is the ablation study on NCI1, which shows a slightly higher accuracy in every case we kept the attention coefficients fixed and without the pooling but a drastic drop in the performance when the edge features are no longer learned. Moreover we see that there are no evident “patters” inside the ablation study with the except that for NCI109 we observe the same behaviour of NCI1 when removing the lift layer. This fact can be explained by noticing that the aforementioned datasets experience, on average, a very similar topology (Table I).

VI. CONCLUSION AND DISCUSSION

In this work we presented Cell Attention Networks (CANs), novel neural architectures operating on data defined over the nodes of a graph incorporated into a regular cell complex, exploiting generalized masked self-attention mechanisms. It builds on skeleton-preserving cellular lifting maps, a novel attentional features lift and a novel edge-level attentional message-passing scheme with two attention functions that operate on the upper and lower connectivities induced by the cell complex. The proposed architecture is also equipped with a novel hierarchical edge pooling technique that leverage a self-attention mechanism to downsample the data in the network’s hidden layers while extracting significant features for the learning task. The Cell Attention Network architecture proposed and tested in the previous sections shows promising results and it is grounded in the theory of regular cell complexes; however, some directions can be explored to enrich the proposed formulation. In particular, a signal processing perspective [18] can be exploited to reinterpret and modify the proposed architecture following a similar approach to [31]; an expressivity analysis can be carried out based on the renewed Weisfeiler-Lehman approach [27], on its generalization to cell complexes [21], [23], or based on spectral approaches [52]. We leave these problems to be addressed in future works.
