SIMPLICIAL ATTENTION NETWORKS

Christopher Wei Jin Goh, Cristian Bodnar & Pietro Liò
Department of Computer Science
University of Cambridge
{cwjg4, cb2015, pl219}@cam.ac.uk

ABSTRACT

Graph representation learning methods have mostly been limited to the modelling of node-wise interactions. Recently, there has been an increased interest in understanding how higher-order structures can be utilised to further enhance the learning abilities of graph neural networks (GNNs) in combinatorial spaces. Simplicial Neural Networks (SNNs) naturally model these interactions by performing message passing on simplicial complexes, higher-dimensional generalisations of graphs. Nonetheless, the computations performed by most existent SNNs are strictly tied to the combinatorial structure of the complex. Leveraging the success of attention mechanisms in structured domains, we propose Simplicial Attention Networks (SAT), a new type of simplicial network that dynamically weighs the interactions between neighbouring simplicies and can readily adapt to novel structures. Additionally, we propose a signed attention mechanism that makes SAT orientation equivariant, a desirable property for models operating on (co)chain complexes. We demonstrate that SAT outperforms existent convolutional SNNs and GNNs in two image and trajectory classification tasks.

1 INTRODUCTION & RELATED WORK

Graph Neural Networks (GNNs) (Sperduti, 1994; Goller & Kuchler, 1996; Gori et al., 2005; Scarselli et al., 2008; Bruna et al., 2014; Defferrard et al., 2016; Kipf & Welling, 2017; Gilmer et al., 2017) have become a successful model for statistical problems characterised by an underlying structure. Nonetheless, the simple combinatorial structure of graphs, relying exclusively on dyadic interactions, has recently driven towards extending these approaches to higher-dimensional generalisations such as simplicial (Ebli et al., 2020; Bunch et al., 2020; Bodnar et al., 2021b) and cellular complexes (Bodnar et al., 2021a; Hajij et al., 2020)

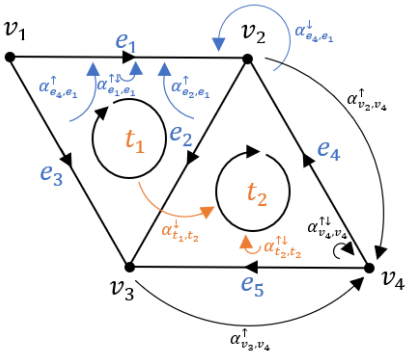

Figure 1: Pictorial diagram of attention coefficients being calculated for node v_4, edge e_1 and triangle t_2

Simplicial Nerual Networks (SNNs) have recently been successfully applied to various problems such as missing data imputation (Ebli et al., 2020), graph classification (Bunch et al., 2020; Bodnar et al., 2021b), molecular property prediction (Bodnar et al., 2021a), edge prediction Chen et al. (2021), trajectory classification (Bodnar et al., 2021b) and prediction (Glaze et al., 2021) as well as homology localisation (Keros et al., 2021). All these approaches can be seen as particular instances of the message passing framework (Bodnar et al., 2021b;a).

Nonetheless, the computations performed by convolutional approaches are tightly coupled to the combinatorial structure of the complex, which could hinder generalisation to unseen simplicial structures. Therefore, motivated by the recent success of attention-mechanisms in graph representation learning (Veličković et al., 2018) and structured domains more generally (Vaswani et al., 2017), we propose Simplicial Attention Networks (SAT). Generalising GAT (Veličković et al., 2018), SAT dynamically learns to attend over different neighbouring simplices based on their features. Additionally, we show how SAT can be made orientation equivariant (Glaze et al., 2021; Bodnar et al., 2021b) via signed attention coefficients, thus making it suitable for oriented simplicial complexes.

In practice, we consider an image classification task based on superpixel graphs and a trajectory classification benchmark involving oriented simplicial complexes. We evaluate SAT in both settings against several GNN and SNN baselines and demonstrate it outperforms them. Code for model and experiments can be found at `https://github.com/ggoh29/Simplicial-neural-network-benchmark`.

2 BACKGROUND

Simplicial Complexes Simplicial complexes are a class of topological spaces that are made of nicely glued simplices of various dimensions. Given a set of vertices V, a k-simplex is an unordered subset $\{v_0, v_1, \ldots, v_k\}$ where $v_i \in V$ and $v_i \neq v_j$ for all $i \neq j$. For a k-simplex $\sigma = \{v_0, v_1, \ldots, v_k\}$, we say its faces are all the (k-1)-simplices that are also subsets of σ, while its cofaces are all (k+1)-simplices that have σ as a face. A simplex can also have an orientation, denoted by $[v_0, v_1, \ldots, v_k]$, where there is a chosen orientation for its vertices. Two orientations are considered equivalent if they differ by an even permutation, or can be expressed by an even number of transpositions. The choice of orientation is arbitrary and it is used only for bookkeeping purposes.

Adjacencies Similarly to how we consider two nodes to be adjacent if there exists an edge that connects the two of them together, there is a notion of adjacency for simplicial complexes. However, adjacency can exist in two forms. Two k-simplices σ_i and σ_j are upper adjacent if both are faces of some $(k+1)$-simplex τ. If the complex is oriented, we further say σ_i and σ_j are similarly oriented with respect to τ if the orientations of σ_i and σ_j agree with the ones induced by τ. If not, they are dissimilarly oriented. Similarly, two k-simplices σ_i and σ_j are lower adjacent if both have a common face. We denote the upper adjacent simplices of σ by $\mathcal{N}_\sigma^\uparrow$ and the down adjacent simplices by $\mathcal{N}_\sigma^\downarrow$. For the purposes of this paper, we also assume that $\sigma \in \mathcal{N}_\sigma^\uparrow$ and $\sigma \in \mathcal{N}_\sigma^\downarrow$. Given two adjacent d-dim simplices σ, τ, denote by $o_{\sigma,\tau} \in \{\pm 1\}$ the relative orientation between them with $o_{\sigma,\sigma} = 1$ for all σ. If the complex is not oriented, we assume $o_{\sigma,\tau} = 1$ for all adjacent simplices σ, τ.

Hodge Laplacian Consider a simplicial complex K. Denote by $\mathbf{C}_k$ the vector space with coefficients in $\mathbb{R}$ having the oriented k-simplices of K as its basis. Elements of this vector space are called k-chains. Then we can define a boundary operator $\partial_k : \mathbf{C}_k(X) \to \mathbf{C}_{k-1}(X)$ acting on the basis elements via $\partial_k[v_0, \ldots, v_k] := \sum_i (-1)^i [v_0, \ldots, \hat{v}_i, \ldots, v_k]$, where $\hat{v}_i$ denotes the face obtained by excluding v_i. This can be represented as a matrix $\mathbf{B}_k$ where the rows are indexed by (k-1)-simplices and the columns are indexed by k-simplices.

Based on the boundary matrix and their transpose, we can define the Hodge Laplacian, a linear operator $\boldsymbol{L}_k : \mathbf{C}_k(X) \to \mathbf{C}_k(X)$, which is a higher order generalisation of the graph Lapacian (Lim, 2019). In matrix form, the k-th Hodge Laplacian is defined as:

$$\boldsymbol{L}_k = \boldsymbol{B}_k^\top \boldsymbol{B}_k + \boldsymbol{B}_{k+1} \boldsymbol{B}_{k+1}^\top \tag{1}$$

One point to note is that $\boldsymbol{B}_k^\top \boldsymbol{B}_k$ correspond to the lower adjacencies, whereas $\boldsymbol{B}_{k+1} \boldsymbol{B}_{k+1}^\top$ correspond to the upper adjacencies. We can refer to them as $\boldsymbol{L}_k^{down}$ and $\boldsymbol{L}_k^{up}$ respectively. The existent simplicial convolutional networks rely on this Laplacian (and its normalised versions) to weigh the different adjacencies between simplices. Instead, we propose using a learned attention matrix that can easily be used both for signals that are k-chains and arbitrary signals on unoriented simplicial complexes.

3 SIMPLICIAL ATTENTION NETWORKS

Let K be a simplicial complex. We will describe our model for an arbitrary dimension of the complex $d \leq \dim(K)$. We compute attention coefficients for the up $\alpha_{\sigma,\tau}^\uparrow$ and down $\alpha_{\sigma,\tau}^\downarrow$ adjacencies via the following equations:

$$\alpha_{\sigma,\tau}^\uparrow = o_{\sigma,\tau} \cdot \mathrm{softmax}_{\tau \in \mathcal{N}_\sigma^\uparrow}(a(\boldsymbol{W}_1 \boldsymbol{h}_\sigma^k, \boldsymbol{W}_1 \boldsymbol{h}_\tau^k)), \tag{2}$$

$$\alpha_{\sigma,\tau}^\downarrow = o_{\sigma,\tau} \cdot \mathrm{softmax}_{\tau \in \mathcal{N}_\sigma^\downarrow}(a(\boldsymbol{W}_2 \boldsymbol{h}_\sigma^k, \boldsymbol{W}_2 \boldsymbol{h}_\tau^k)), \tag{3}$$

where a is a function for computing attention coefficients. In an oriented simplicial complex, this effectively becomes a form of signed attention. Note that when working at the node-level, where

the relative orientation between nodes is trivial and only upper adjacencies are present, one recovers GAT Veličković et al. (2018).

Denote by $\boldsymbol{H}^d$ the features associated to the d-dimensional simplices of K. SAT layers are described by the following message passing equation weighting the neighbours by the attention coefficients:

$$\boldsymbol{h}_\sigma^{k+1} = \phi\Big(\sum_{\tau \in \mathcal{N}_\sigma^\uparrow} \alpha_{\sigma,\tau}^\uparrow \boldsymbol{W}_1^k \boldsymbol{h}_\tau^k, \sum_{\tau \in \mathcal{N}_\sigma^\downarrow} \alpha_{\sigma,\tau}^\downarrow \boldsymbol{W}_2^k \boldsymbol{h}_\tau^k \Big) \tag{4}$$

Here, ϕ is an update function that aggregates the two incoming messages from the lower and upper adjacencies and updates the representation of σ. More generally, each of these arguments can effectively be augmented with Z attention heads:

$$\boldsymbol{h}_\sigma^{k+1} = \|_{z \leq Z} \phi\Big(\sum_{\tau \in \mathcal{N}_\sigma^\uparrow} \alpha_{\sigma,\tau}^{\uparrow,z} \boldsymbol{W}_1^k \boldsymbol{h}_\tau^k, \sum_{\tau \in \mathcal{N}_\sigma^\downarrow} \alpha_{\sigma,\tau}^{\downarrow,z} \boldsymbol{W}_2^k \boldsymbol{h}_\tau^k \Big) \tag{5}$$

In oriented simplicial complexes, the choice of orientation is arbitrary and we would like the model to be aware of this symmetry. Mathematically, we would like SAT to be *orientation equivariant* (Bodnar et al., 2021b; Glaze et al., 2021).

Definition 1 (Bodnar et al. (2021b))**.** *Let K be a simplicial complex described by boundary matrices $\{\boldsymbol{B}_i\}$. A function $f : \boldsymbol{C}_k \to \boldsymbol{C}_k$ is orientation equivariant if for any diagonal matrix $\boldsymbol{T}$ with ± 1 on the diagonal, $f(\boldsymbol{T}\boldsymbol{H}^k, \boldsymbol{B}_k\boldsymbol{T}, \boldsymbol{T}\boldsymbol{B}_{k+1}) = \boldsymbol{T} f(\boldsymbol{H}^k, \boldsymbol{B}_k, \boldsymbol{B}_{k+1})$.*

We will now show that under certain constraints on the functions a and ϕ the model is orientation equivariant.

Proposition 2. *If the function a is even in both of its arguments and ϕ is odd in both of its arguments, SAT is orientation equivariant.*

Proof Sketch. At the message passing level, the equivariance equation is respected if and only if for any simplex σ the local aggregation is invariant to the changes in orientation of the neighbour simplices and if σ changes its orientation, the output features $\boldsymbol{h}_\sigma^{k+1}$ pick up a minus sign (see Bodnar et al. (2021b) for details).

Because a is an even function, the product $\alpha_{\sigma,\tau}\boldsymbol{h}_\tau$ is invariant with respect to changes in the orientation of τ because the change in sign for $\alpha_{\sigma,\tau}$ and $\boldsymbol{h}_\tau$ cancel each other. Therefore, the aggregation is invariant with respect to changes in the orientations of the neighbours. Furthermore, since the function ϕ is odd and $\alpha_{\sigma,\sigma} \geq 0$ for any orientation of σ, it follows that the local aggregation picks up a minus sign when the orientation of σ is flipped. □

4 RESULTS

4.1 SUPERPIXEL GRAPHS

Dataset We use the classification of superpixel graphs as our first benchmark to demonstrate the capabilities of SAT. A superpixel graph is the graph representation of an image in which pixels are grouped into nodes representing perceptually meaningful regions, such as a region of similar intensity (Achanta et al., 2012). By representing the image as a graph, it is possible to change the task of image classification to graph classification. This application of using graph neural networks on superpixel graphs for the task of image classification was first done by Monti et al. (2016) and has since been a popular framework for testing graph neural networks. Here, we use images from the Modified National Institute of Standards and Technology database (MNIST) (LeCun & Cortes, 2010) containing handwritten digits from 0 to 9.

To construct a superpixel dataset from MNIST, we use the Simple Linear Iterative Clustering (SLIC) (Achanta et al., 2012) algorithm. The superpixel nodes are then connected to directly adjacent nodes by means of a region adjacency graph to get the final graph. The simplicial complex we consider is the two-dimensional clique complex of this graph. Triangles are of particular importance as they tend to encode tightly connected regions constituting a form of higher-order interaction between the superpixels.

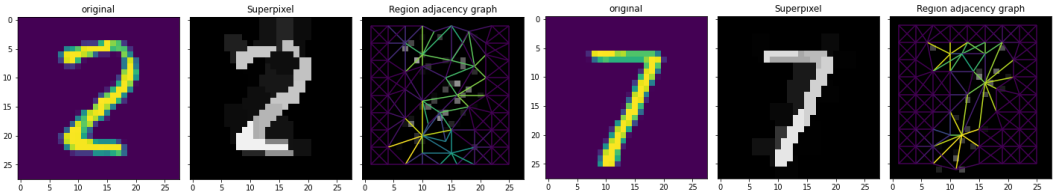

Figure 2: The stages from image to graph for two different MNIST images

We follow the work of Long et al. (2021) in setting up the node features and how the GNN layers are used. The node features of node cluster j are $\boldsymbol{h}_j = [{}^1\!/_n \sum_i^{n_j} (a_i, b_i, g_i)]$ where the a_i and b_i are the x and y coordinate of pixel i respectively, and g_i is the greyscale value of pixel i. The features for the 1-simplices and 2-simplices will be the concatenation of the node features that make up the simplex in the order given by pixel values.

Architecture Our layers are arranged in the same architecture proposed by Long et al. (2021) known as a hierarchical GNN architecture. This was used to avoid the over-smoothing problem (Cai & Wang, 2020; Oono & Suzuki, 2019). This architecture comprises of three GNN/SNN layers. The resulting node features from each layer, also known as the residual, are concatenated, resulting in the penultimate feature vector comprising of features vectors from previous layers. This is then passed through a mean pooling function, before being fed to a multi-layer perceptron with 10 outputs followed by a softmax activation.

Experimental setup For the experiment, we set the superpixel algorithm SLIC to generate approximately 75 nodes. For all experiments we set a budget of 100 epochs for training, with a batch size of 32 images, using a split of 55k images for training and 5k images for validation. The model corresponding to the epoch with the best validation performance on the validation set is saved and tested against the MNIST test set of 10k images. For the optimisation, we use the Adam optimiser, a weight decay of 0.0005, and a learning rate of 0.001 for all models tested.

We compare SAT against (GCN) (Kipf & Welling, 2017), GAT (Veličković et al., 2018), SCCONV (Bunch et al., 2020), and SCN (Ebli et al., 2020). All models were used in a hierarchical GNN setup explained above with 3 layers each. We set the number of feature channels for each model such that the total number of parameters across all models was roughly 10k. All models use a ReLU activation function except SCN which uses Leaky ReLU as in the original paper. Both GAT and SAT use two attention heads.

Results We repeat the training procedure described above across five seeds and report the mean accuracy and standard deviation. Additionally, we report the exact number of parameters of each model. As the table shows, SAT outperforms its GNN counterpart (GAT), as well as other simplicial convolutional networks that have been recently proposed.

Model	Parameters	Test Accuracy
GCN	10634	63.65 ± 1.82
GAT	9862	88.95 ± 0.99
SCN	10612	84.16 ± 1.23
SCCONV	10315	89.06 ± 0.47
SAT (Ours)	10186	$\mathbf{92.99} \pm \mathbf{0.71}$

Table 1: Image classification accuracy. SAT outperforms its GNN counterpart (GAT) as well as other recently proposed SNNs.

4.2 TRAJECTORY CLASSIFICATION

To demonstrate the superior performance of SAT when operating on k-chains as well as to display the benefits of orientation equivariance in this setting, we now consider a trajectory classification task (Schaub et al., 2020; Bodnar et al., 2021b) involving oriented simplicial complexes.

Dataset The trajectory classification dataset is a synthetic dataset constructed by sampling 1000 points within a unit square followed by a Delauney triangulation. Two holes within the graph are created by removing certain nodes and the edges connected to these nodes. Trajectories are generated by randomly sampling a starting point from the top-left corner and an endpoint from the bottom right corner. Two different types of such trajectories are generated. One class of trajectories traverses via the top-right corner, while the second class traverses via the bottom-left corner. We randomly generate 1000 train and 200 test trajectories in this manner. In order for the test to be challenging for non-orientation invariant models, all trajectories from the training dataset use the same orientation for the edges, while trajectories from the test dataset use random orientations. Two trajectories from this dataset, belonging to two different classes, can be seen in Figure 3.

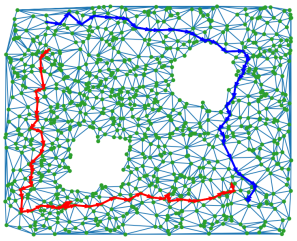

Figure 3: Example of the two different types of trajectories from the trajectory classification dataset.

Experimental setup All models use the same hyperparameters. We use 4 layers, with the residual size set to 32 for all layers. Following the 4 layers, we take the absolute of the output and do a mean-pooling to make it orientation invariant. We then pass this through two multi-layer perceptron layers before finally taking the softmax of the output. For all experiments, we set a budget of 100 epochs for training, with a batch size of 4. Since there is no validation set, the model corresponding to the epoch with the best performance on the training set is saved and used for testing. For the optimisation, we once again use the Adam optimiser, a weight decay of 0.0005, and a learning rate of 0.001 for all models tested.

Results We repeat the training procedure described above across five seeds and three different activation functions and report the mean accuracy and standard deviation. Models that use odd activation functions (i.e. the identity of tanh), which makes them orientation equivariant, perform better on the test set. Furthermore, the orientation equivariant SAT outperforms other orientation equivariant SNNs.

	activation function		
Model	Id	ReLU	Tanh
SCN	53.10 ± 2.27	49.70 ± 2.77	52.80 ± 3.11
SCCONV	62.80 ± 3.11	50.80 ± 1.63	62.30 ± 3.97
SAT (Ours)	$\mathbf{92.90 \pm 2.22}$	49.70 ± 0.60	$\mathbf{93.80 \pm 1.33}$

Table 2: Trajectory classification accuracy. SAT outperforms other recently proposed SNNs.

5 CONCLUSION

In this paper, we introduced Simplicial Attention Networks (SAT)[1], a novel SNN that is able to incorporate attention-mechanisms in order to assign different importance weights to neighbouring simplices as well generalise to unseen simplicial structures. For applications involving orientated simplices, we also illustrated how SAT can be made orientation equivariant via signed attention coefficients. Empirically, we showed SAT outperforms other GNNs and SNNs at the tasks of classifying superpixel images generated from the MNIST dataset and classifying trajectories represented as 1-chains on a simplicial complex.

[1]We note that a similar model was proposed concurrently by Giusti et al. (2022)

REFERENCES

Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yu Guang Wang, Pietro Liò, Guido Montúfar, and Michael Bronstein. Weisfeiler and Lehman Go Cellular: CW Networks. In *NeurIPS*, 2021a.

Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido F Montufar, Pietro Lió, and Michael Bronstein. Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks. In *ICML*, 2021b.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *ICLR*, 2014.

Eric Bunch, Qian You, Glenn Fung, and Vikas Singh. Simplicial 2-complex convolutional neural networks. In *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*, 2020. URL `https://openreview.net/forum?id=TLbnsKrt6J-`.

Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. *arXiv:2006.13318*, 2020.

Yuzhou Chen, Yulia R. Gel, and H. Vincent Poor. Bscnets: Block simplicial complex neural networks, 2021.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, 2016.

Stefania Ebli, Michaël Defferrard, and Gard Spreemann. Simplicial neural networks. In *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*, 2020. URL `https://openreview.net/forum?id=nPCt39DVIfk`.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, 2017.

L. Giusti, C. Battiloro, P. Di Lorenzo, S. Sardellitti, and S. Barbarossa. Simplicial attention neural networks, 2022. URL `https://arxiv.org/abs/2203.07485`.

Nicholas Glaze, T. Mitchell Roddenberry, and Santiago Segarra. Principled simplicial neural networks for trajectory prediction. In *ICML*, 2021.

Christoph Goller and Andreas Kuchler. Learning task-dependent distributed representations by backpropagation through structure. In *ICNN*, 1996.

Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *IJCNN*, 2005.

Mustafa Hajij, Kyle Istvan, and Ghada Zamzmi. Cell complex neural networks. In *NeurIPS Workshop on Topological Data Analysis and Beyond*, 2020.

Alexandros Dimitrios Keros, Vidit Nanda, and Kartic Subr. Dist2cycle: A simplicial neural network for homology localization, 2021.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

Lek-Heng Lim. Hodge laplacians on graphs, 2019.

Jianwu Long, Zeran yan, and Hongfa chen. A graph neural network for superpixel image classification. *Journal of Physics: Conference Series*, 1871(1):012071, apr 2021. doi: 10.1088/1742-6596/1871/1/012071. URL `https://doi.org/10.1088/1742-6596/1871/1/012071`.

Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. *CoRR*, abs/1611.08402, 2016. URL http://arxiv.org/abs/1611.08402.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. *arXiv:1905.10947*, 2019.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Trans. Neural Networks*, 20(1):61–80, 2008.

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.

Alessandro Sperduti. Encoding labeled graphs by labeling RAAM. In *NIPS*, 1994.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv:1706.03762*, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.