

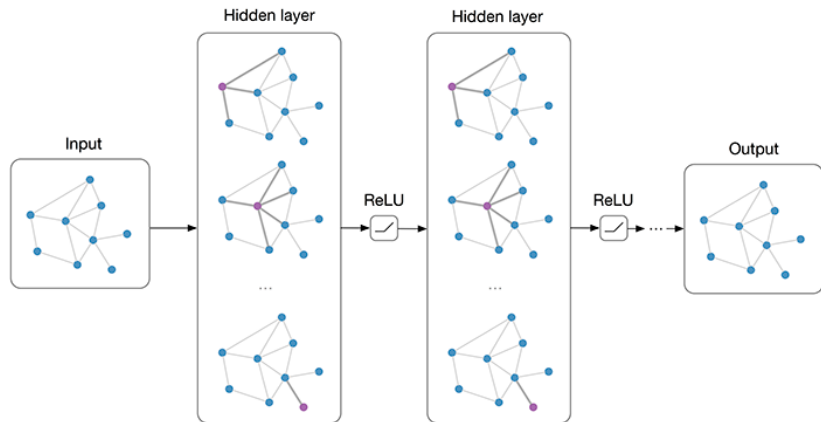
Large Scale GNN and Transformer Models and for Genomics

Presenter: Jack Lanchantin

University of Virginia
<https://qdata.github.io/deep2Read/>

201905

GCN



$$x_i^{l+1} = \sigma \left(\sum_{j \in \mathcal{N}(i)} x_j^l W^l \right)$$

$$X^{l+1} = \sigma(AX^lW^l) \quad (1)$$

Time : $O(LEd^2)$

Space : $O(LEd)$

where L is the number of layers, d is the embedding dimension, and E is the number of edges.

Outline

Graphs

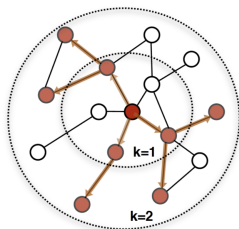
Sequences

Genomics

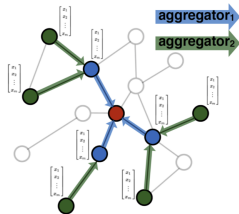
GraphSAGE

[4]

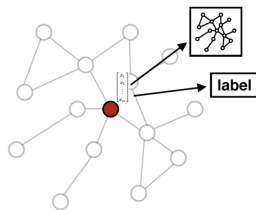
Sample only k neighboring nodes at each layer and update those



1. Sample neighborhood



2. Aggregate feature information from neighbors



3. Predict graph context and label using aggregated information

GraphSAGE

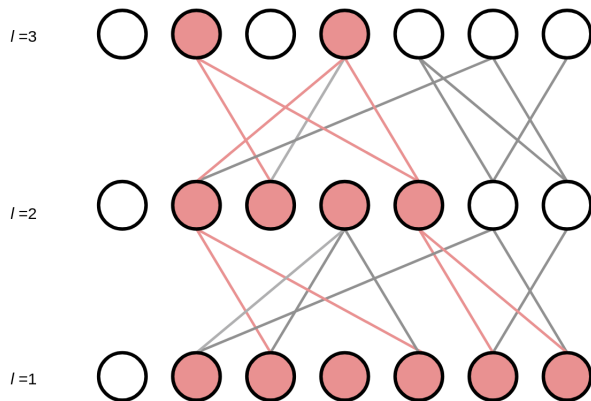
[4]

$$X_{[k]}^{l+1} = \sigma(A_{[k]} X_{[k]}^l W^l) \quad (2)$$

$$O((k^2)^L d)$$

GraphSAGE

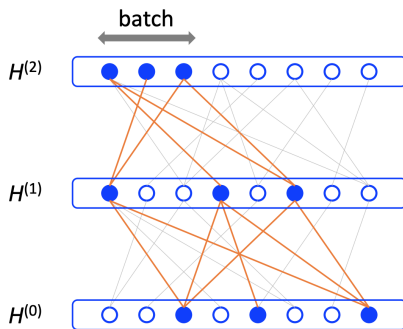
[4]



FastGCN

[1]

Sample only k nodes at each layer and update those



FastGCN

[1]

$$q(u) = \|A(:, u)\|^2 / \sum_{u' \in V} \|A(:, u')\|^2, \quad u \in V \quad (3)$$

$$H^{(l+1)}(v, ;) = \sigma \left(\frac{1}{t_l} \sum_{j=1}^{t_l} \frac{A(v, u_j^{(l)}) H^{(l)}(u_j^{(l)}, :)}{q(u_j^{(l)})} W^{(l)} \right), \quad u_j^{(l)} \sim q \quad (4)$$

FastGCN

[1]

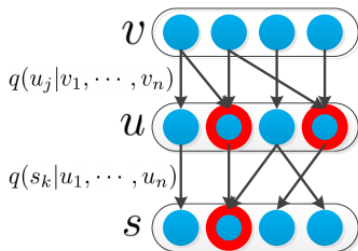
$$X_{[k]}^{l+1} = \sigma(A_{[k]} X_{[k]}^l W^l) \quad (5)$$

$O(k^2)$

Adaptive Sampling

[6]

Sample only k nodes at each layer conditioned on sampled nodes at the previous layer



Adaptive Sampling

[6]

$$X_{[k]}^{l+1} = \sigma(A_{[k]} X_{[k]}^l W^l) \quad (6)$$

$O(k^2)$

Multi-Level Framework Scalable Graph Embedding (MILE)

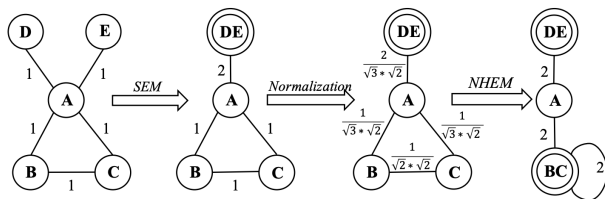
[8]

3-step process:

1. Repeatedly coarsen graph into smaller ones
2. Compute embeddings on coarsest graph using existing embedding method
 - ▶ Inexpensive and less memory than full graph
 - ▶ Captures global structure
3. Novel refinement model - learn graph convolution network to refine the embeddings from the coarsest graph to the original graph

Multi-Level Framework Scalable Graph Embedding (MILE)

[8]



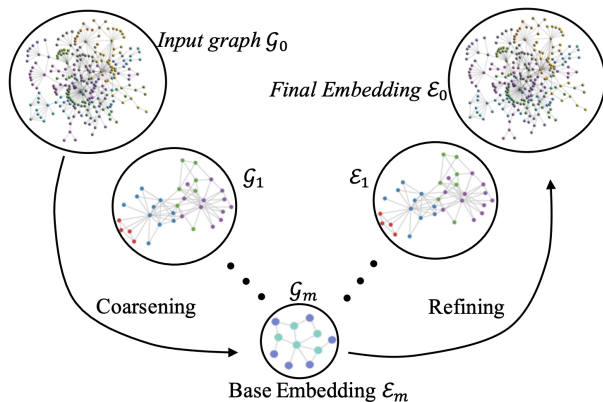
$$A_0 = \begin{matrix} & \begin{matrix} \text{A} & \text{B} & \text{C} & \text{D} & \text{E} \end{matrix} \\ \begin{matrix} \text{A} \\ \text{B} \\ \text{C} \\ \text{D} \\ \text{E} \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

$$M_{0,1} = \begin{matrix} & \begin{matrix} \text{A} & \text{BC} & \text{DE} \end{matrix} \\ \begin{matrix} \text{A} \\ \text{B} \\ \text{C} \\ \text{D} \\ \text{E} \end{matrix} & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

$$A_1 = M_{0,1}^T A_0 M_{0,1} = \begin{pmatrix} 0 & 2 & 2 \\ 2 & 2 & 0 \\ 2 & 0 & 0 \end{pmatrix}$$

Multi-Level Framework Scalable Graph Embedding (MILE)

[8]



Multi-Level Framework Scalable Graph Embedding (MILE)

[8]

Projected Embeddings:

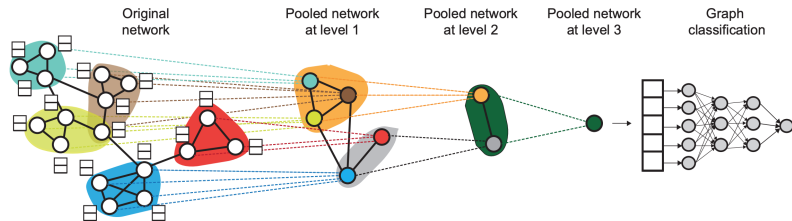
$$\mathcal{E}_i^P = M_{i,i+1}\mathcal{E}_{i+1} \quad (7)$$

Refined Embeddings:

$$\mathcal{E}_i = \sigma(\mathbf{A}_i\mathcal{E}_i^PW) \quad (8)$$

Hierarchical Graph Representation Learning with Differentiable Pooling

[13]



Matrix Factorization Methods

- ▶ LanczosNet: Multi-Scale Deep Graph Convolutional Networks
 - ▶ Exploits the low rank approximation of the graph Laplacian
- ▶

Outline

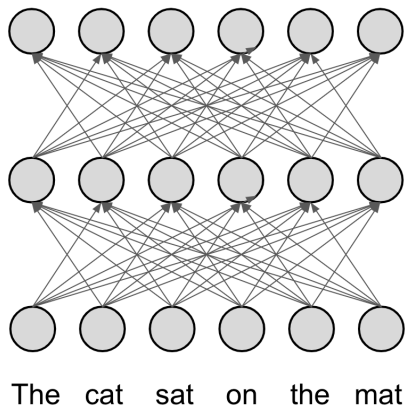
Graphs

Sequences

Genomics

Transformer/BERT

[12]



Transformer/BERT

GCN:

$$x_i^{l+1} = \sigma \left(\sum_{j \in \mathcal{N}(i)} x_j^l W^l \right)$$

Transformer:

$$x_i^{l+1} = \sigma \left(\sum_{j \neq i} \alpha_j x_j^l W^l \right)$$

Transformer/BERT

[12]

$$X^{l+1} = \sigma(\alpha^l X^l W^l) \quad (9)$$

$$\alpha = \text{Attn}(X^l, X^l, X^l) \quad (10)$$

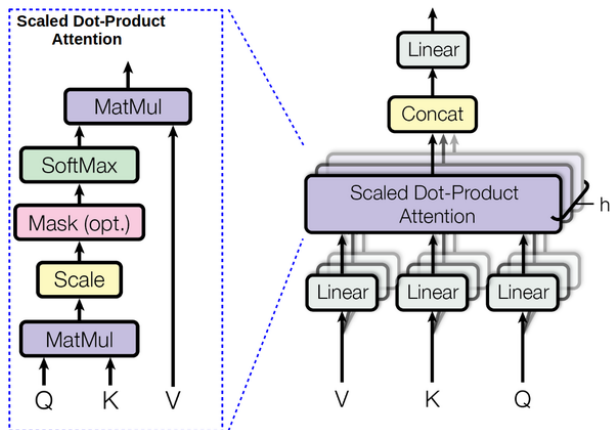
$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \quad (11)$$

$$X^0 = \text{lookupTable}(x) + \text{positionEncoding}(x)$$

$$O(LN^2d)$$

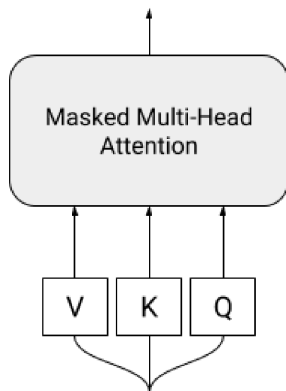
Transformer/BERT

[12]



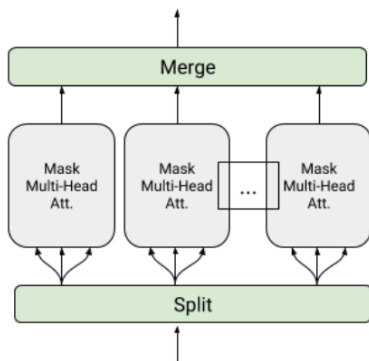
Transformer/BERT

$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \quad (12)$$



Local Attention

[9]

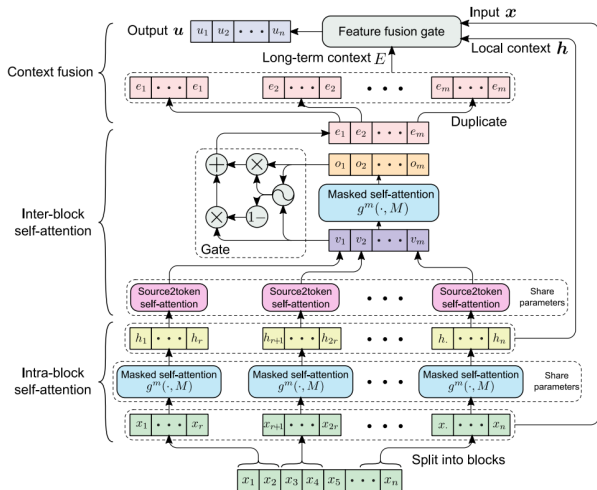


$$O(k^2)$$

where k is the block size and $B = \frac{N}{k}$ is the number of blocks

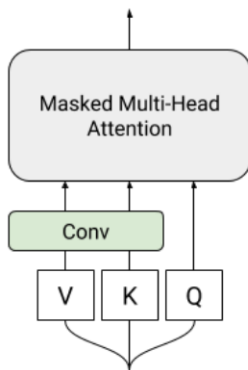
Block Self Attention

[11]



Memory Compressed Attention

[9]



Reduce the number of keys and values by using a strided convolution. The number of queries remains unchanged.

$$O(N \frac{N}{k} d)$$

Music Transformer

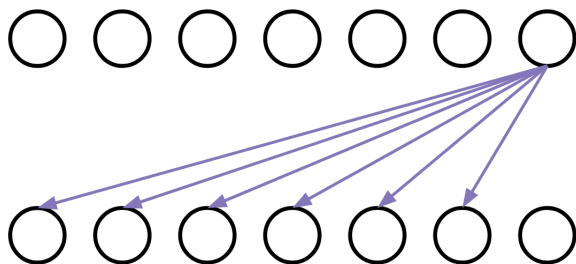
[5]

$$\text{Relative Attention} = \text{Softmax} \left(\frac{QK^{\top} + S^{rel}}{\sqrt{D_h}} \right) V \quad (13)$$

- ▶ S^{rel} , an $L \times L$ dimensional logits matrix which modulates the attention probabilities for each head.
- ▶ $S^{rel} = QR^{\top}$, where R is a tensor of shape (L, L, D_h) containing the embeddings that correspond to the relative distances between all keys and queries.

Generating Long Sequences with Sparse Transformers

[2]

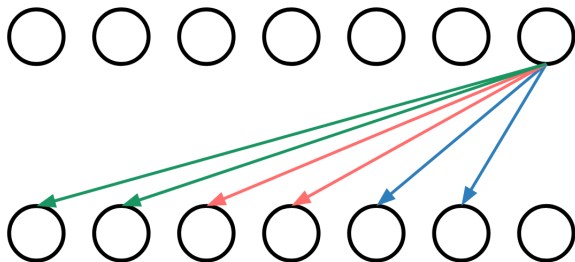


$$\text{Attn}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

$$O(N^2 d)$$

Generating Long Sequences with Sparse Transformers

[2]



$$O(k^2 \frac{d}{p})$$

given p attention heads, each with a window size of k

Generating Long Sequences with Sparse Transformers

[2]

Given $S = \{S_1, \dots, S_n\}$ where S_i denotes the set of indices of the input vectors to which the i th output vector attends,

$$\begin{aligned} \text{Attend}(X, S) &= (a(x_i, S_i))_{i \in \{1, \dots, n\}} \\ a(x_i, S_i) &= \text{softmax} \left(\frac{(W_q x_i) K_{S_i}^T}{\sqrt{d}} \right) V_{S_i} \\ K_{S_i} &= (W_k x_j)_{j \in S_i} \\ V_{S_i} &= (W_v x_j)_{j \in S_i} \end{aligned} \tag{14}$$

Factorized self-attention instead has p separate attention heads, where the m th head defines a subset of the indices $A_i^{(m)} \subset \{j : j \leq i\}$ and lets $S_i = A_i^{(m)}$ where $|A_i^{(m)}| \propto \sqrt[p]{n}$

Outline

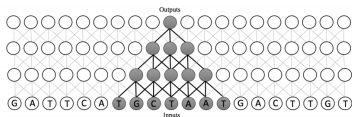
Graphs

Sequences

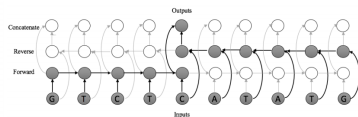
Genomics

Dilated CNNs for Long-Distance Genomic Dependencies

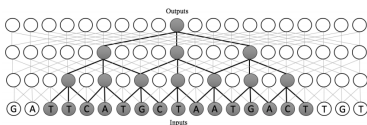
[3]



(a) Convolution



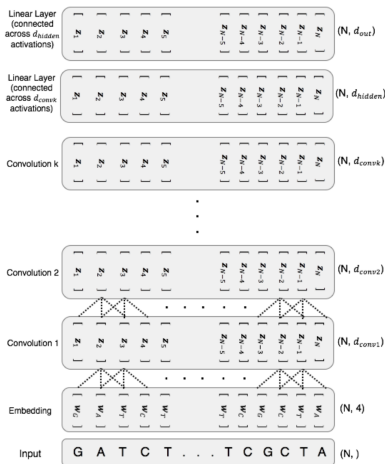
(b) Bidirectional LSTM



(c) Dilated Convolution

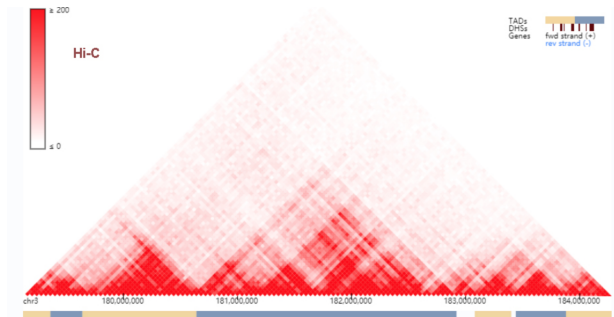
Dilated CNNs for Long-Distance Genomic Dependencies

[3]

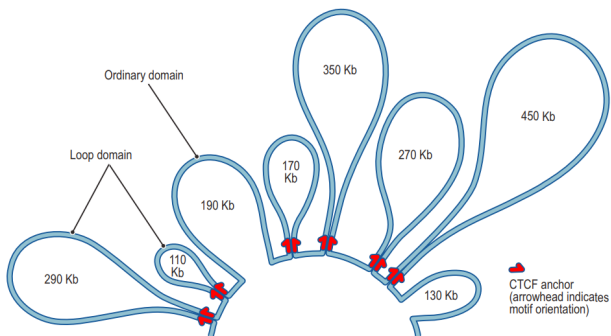


Input length: 25,000 bp, Output labels: 919

Hi-C



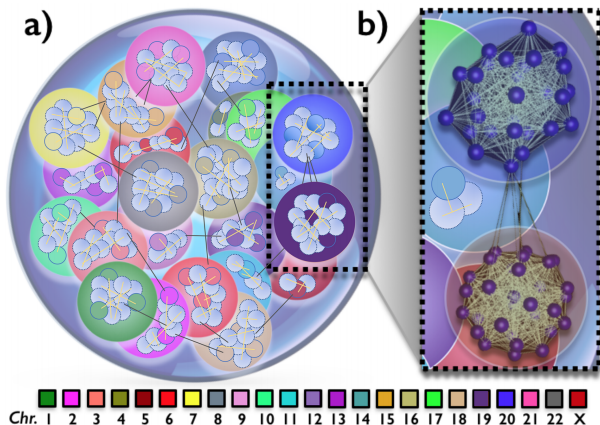
Hi-C



However, Hi-C maps are cell-line specific

Structure of the human chromosome interaction network

[10]



Structure of the human chromosome interaction network

[10]

- ▶ Intra-chromosomal contacts broadly occur between epigenomically homologous regions
- ▶ Inter-chromosomal contacts are especially associated with regions rich in highly expressed genes.
- ▶ GNN is a good strategy for Using HiC for genomics [7]

Structure of the human chromosome interaction network

[10]

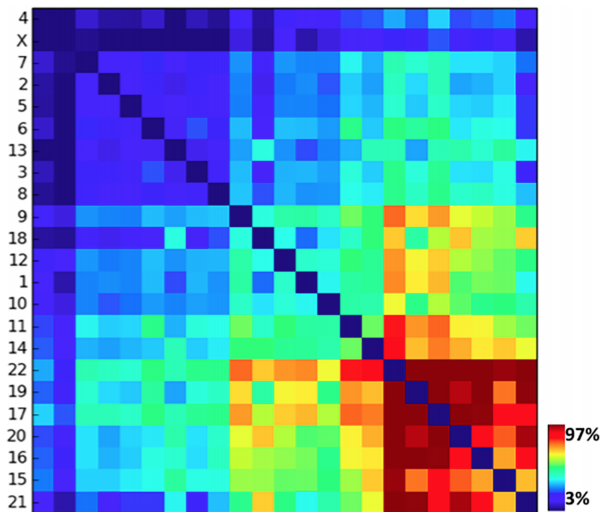


Fig 4. The heatmap of the overall contacts between chromosome pairs. While different chromosomal pairs have different degrees of interactions, the RSS analysis points out that there are no significant isolated subgroups and the system forms a single nuclear network.

Conclusion

- ▶ GNN sampling methods aren't directly transferable to sequence methods such as Transformer
- ▶ Block transformers are still the current method for long range dependencies

References I

- [1] Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247*, 2018.
- [2] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.
- [3] Ankit Gupta and Alexander M Rush. Dilated convolutions for modeling long-distance genomic dependencies. *arXiv preprint arXiv:1710.01278*, 2017.
- [4] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

References II

- [5] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, and Douglas Eck. An improved relative self-attention mechanism for transformer with application to music generation. *arXiv preprint arXiv:1809.04281*, 2018.
- [6] Wenbing Huang, Tong Zhang, Yu Rong, and Junzhou Huang. Adaptive sampling towards fast graph representation learning. In *Advances in Neural Information Processing Systems*, pages 4558–4567, 2018.
- [7] Jack Lanchantin and Yanjun Qi. Graph convolutional networks for epigenetic state prediction using both sequence and 3D genome data. *Bioinformatics*, 36(Supplement₂) : i659 – –i667, 122020.
- [8] Jiongqian Liang, Saket Gurukar, and Srinivasan Parthasarathy. Mile: A multi-level framework for scalable graph embedding. *arXiv preprint arXiv:1802.09612*, 2018.

References III

- [9] Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*, 2018.
- [10] Sergio Sarnataro, Andrea M Chiariello, Andrea Esposito, Antonella Prisco, and Mario Nicodemi. Structure of the human chromosome interaction network. *PLoS one*, 12(11):e0188201, 2017.
- [11] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. Bi-directional block self-attention for fast and memory-efficient sequence modeling. *arXiv preprint arXiv:1804.00857*, 2018.
- [12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

References IV

- [13] Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. *arXiv preprint arXiv:1806.08804*, 2018.