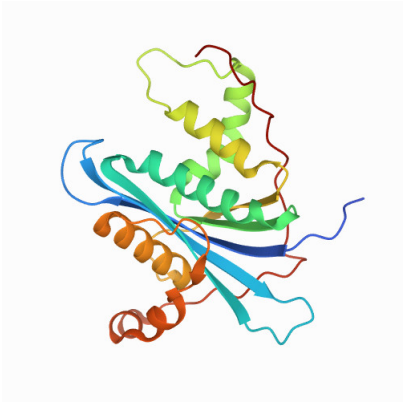
GraphiT: Encoding Graph Structure in Transformers

Grégoire Mialon, Dexiong Chen, Margot Selosse, Julien Mairal

NYU/Willow seminar

Graph data are an important research topic



LC9-RNase H1 from *Escherichia Coli*

Graph data are very valuable...

- Proteins in computational biology [Senior et al., 2020].
- Molecules in chemoinformatics [Duvenaud et al., 2015].
- Shapes in computer vision and computer graphics [Verma et al., 2018], etc.

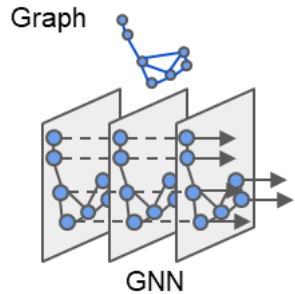
... but delicate to handle.

- Irregular structure.
- How to make use of neural networks?

Learning on graph data today

Graph Neural Networks (GNNs).

- Introduced as an extension of neural networks for graph-structured data [Scarselli et al., 2008].
- Message passing paradigm in which feature vectors are exchanged between neighboring nodes.
- Node representations are updated using neural networks.
- Many strategies to aggregate features of neighboring nodes [Duvenaud et al., 2015, Bronstein et al., 2017, Veličković et al., 2018].



(From Shi and Rajkumar, 2020)

Inductive biases for graphs

In GNNs, messages flow between neighbors only.

- Exploits the structure of the graph.
- But n layers for n-hop neighbors to communicate.
- Could global aggregation better capture long-range interactions?

Transformers perform global aggregation!

Transformers

Transformers (we only use encoder).

- A sequence of layers processing an input set of n elements X in $\mathbb{R}^{n \times d_{\text{in}}}$, and compute another set in $\mathbb{R}^{n \times d_{\text{out}}}$.
- Self-attention mechanism:

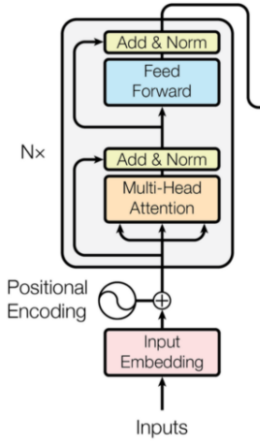
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_{out}}}\right) V \in \mathbb{R}^{n \times d_{\text{out}}}, \tag{1}$$

  with $Q^\top = W_Q X^\top$ and $K^\top = W_K X^\top$ resp. query and key matrices, $V^\top = W_V X^\top$ the value matrix, and W_Q, W_K, W_V in $\mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ learned projection matrices.
- During forward pass, feature map X updated via:

$$X = X + \text{Attention}(Q, K, V).$$

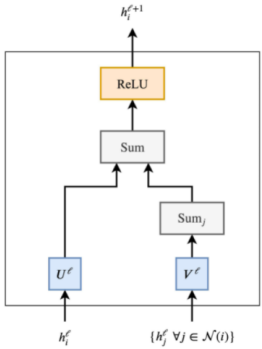
- LayerNorm then "element-wise" feed-forward.
- Repeat.

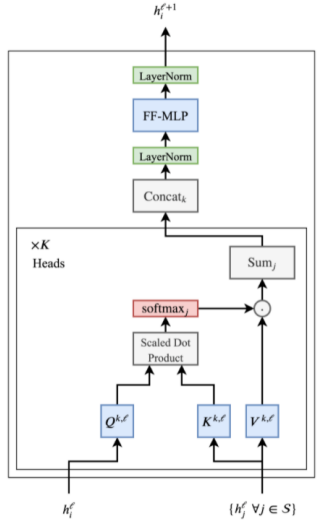
Transformers



Transformer encoder (from Vaswani et al., 2017)

GNNs and transformers are tightly connected



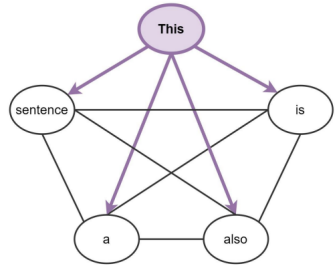


Left: GNN. *Right*: Transformer (from Joshi, 2020).

Challenging GNNs with transformers encoding graph structure

GNNs and transformers are tightly connected, but...

- GNNs are the standard architecture for learning on graphs. Inductive bias: message passing between neighbors.
- Transformers: all input elements are allowed to communicate.
- Self-attention layer is permutation invariant.
- Without structure encoding, a bag of graph nodes model!



(From Joshi, 2020)

How to provide the transformer with graph structural information?

- Our work, GraphiT, tackles this issue.

Our contribution: Two mechanisms for encoding graph structure in transformers

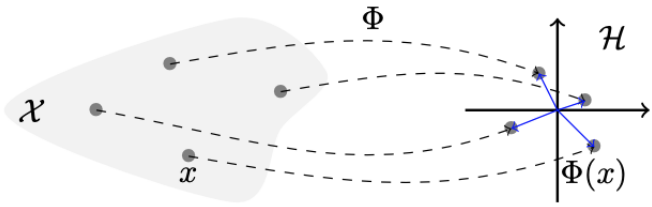
I- Node position encoding.

- Position encoding: adding positional only information to the feature vector of an input node or to the attentions scores.
- As opposed to sequences or images, encoding positions of the elements in a graph is not trivial.
- [Dwivedi and Bresson, 2021] proposed absolute position encoding strategy based on the eigenvectors of the Laplacian. Transferability between graphs?
- **We propose a relative position encoding based on kernels on graphs.**

II- Encoding substructures.

- Substructures: carry local positional information and content, *e.g* walks, subtrees, graphlets.
- Heavily used within graph kernels [Borgwardt et al., 2020].
- **We propose to encode local substructure using a method from [Chen et al., 2020].**

Reminder: Kernel methods



(From Bietti, 2019)

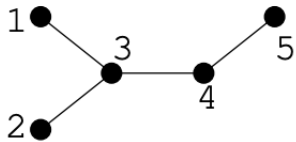
Learning with Kernel methods

- Map data x to high-dimensional space, $\Phi(x) \in \mathcal{H}$ (RKHS).
- Φ associated to a positive definite kernel K: $K(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}}$.
- Convex optimization for learning linear decision function in the RKHS (non-linear in the original space, kernel trick).

I - Kernel on graphs

Spectral graph analysis.

- The Laplacian of a graph with n nodes defined as $L = D - A$. D is a $n \times n$ diagonal matrix of node degrees and A the adjacency matrix.



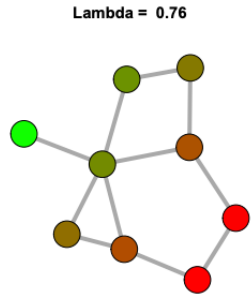
$$L = D - A = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

(From JP Vert's course on kernel methods)

I - Kernel on graphs

Spectral graph analysis.

- Eigenvalue decomposition $L = \sum_i \lambda_i u_i u_i^\top$.
- Eigenvalues $\lambda_i = u_i^\top L u_i = \sum_{j \sim k} (u_i(x_j) - u_i(x_k))^2$.
- Characterizes the amount of oscillation of the corresponding eigenvector u_i (a function on the nodes).
- "Discrete equivalent" to sine/cosine Fourier basis in $\mathbb{R}^n$ and associated frequencies.

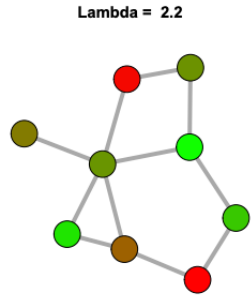


(From JP Vert's course on kernel methods)

I - Kernel on graphs

Spectral graph analysis.

- Eigenvalue decomposition $L = \sum_i \lambda_i u_i u_i^\top$.
- Eigenvalues $\lambda_i = u_i^\top L u_i = \sum_{j \sim k} (u_i(x_j) - u_i(x_k))^2$.
- Characterizes the amount of oscillation of the corresponding eigenvector u_i (a function on the nodes).
- "Discrete equivalent" to sine/cosine Fourier basis in $\mathbb{R}^n$ and associated frequencies.



(From JP Vert's course on kernel methods)

I - Kernels on graphs

Laplacian based kernels [Smola and Kondor, 2003].

- It is possible to define a family of p.d. kernels on the graph by applying a regularization function r to the spectrum of L.
- We get a rich class of kernels

$$K_r = \sum_{i=1}^{m} r(\lambda_i) u_i u_i^\top, \tag{2}$$

associated with the norm $\|f\|_r^2 = \sum_{i=1}^{m} (f_i^\top u_i)^2 / r(\lambda_i)$ from a reproducing kernel Hilbert space (RKHS), where $r : \mathbb{R} \mapsto \mathbb{R}_*^+$ is a non-increasing function such that smoother functions on the graph would have smaller norms in the RKHS.

I - Kernels on graphs

Diffusion Kernel [Kondor and Vert, 2004].

- When $r(\lambda_i) = e^{-\beta\lambda_i}$,

$$K_D = \sum_{i=1}^{m} e^{-\beta\lambda_i} u_i u_i^\top = e^{-\beta L} = \lim_{p\to+\infty} \left(I - \frac{\beta}{p} L\right)^p.$$

- Discrete equivalent of the Gaussian kernel, a solution of the heat equation in the continuous setting, hence its name.
- Interpretation in terms of diffusion of a substance in the graph, controlled by β.

I - Kernels on graphs

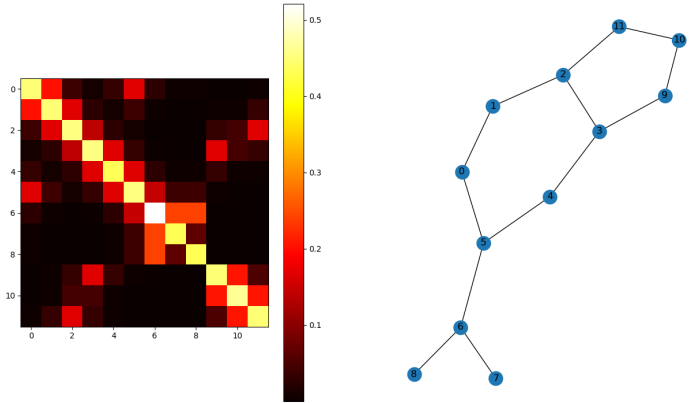


Figure 1: Diffusion kernel between the nodes of a MUTAG sample graph ($\beta = 1$).

I - Relative position encoding with kernels on graphs

Regular attention.

- Self-attention layer

$$\text{Attention}(Q, V) = \text{normalize}\left(\exp\left(\frac{QQ^\top}{\sqrt{d_{\text{out}}}}\right)\right) V \in \mathbb{R}^{n \times d_{\text{out}}}. \tag{3}$$

- During forward pass, feature map X is updated as follows:

$$X = X + \text{Attention}(Q, V). \tag{4}$$

Remark. As in [Tsai et al., 2019], we use the same matrices for Q and K.

I - Relative position encoding with kernels on graphs

Modulated attention.

- Self-attention layer becomes

$$\text{PosAttention}(Q, V, K_r) = \text{normalize}\left(\exp\left(\frac{QQ^\top}{\sqrt{d_{\text{out}}}}\right) \odot K_r\right) V \in \mathbb{R}^{n \times d_{\text{out}}}, \tag{5}$$

  with the same Q and V matrices, and K_r a kernel on the graph.

- During forward pass, feature map X is updated as follows:

$$X = X + D^{-\frac{1}{2}}\,\text{PosAttention}(Q, V, K_r), \tag{6}$$

  with D the matrix of node degrees and K_r a kernel on the graph.

Remark. As opposed to absolute position encoding, the model does not rely on the transferability of eigenvectors between different Laplacians.

I - Kernel smoothing interpretation

Self-attention as a kernel smoothing.

- We can rewrite self-attention:

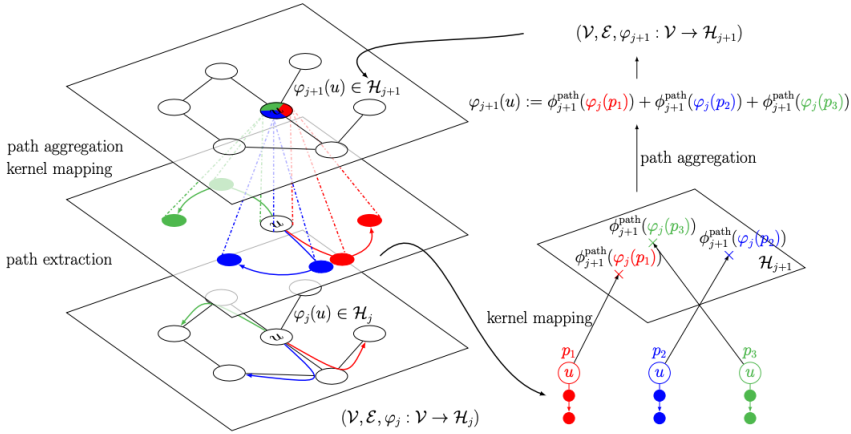
$$\text{Attention}(Q,K,V)_i = \sum_{j=1}^{n} \frac{exp\left(\frac{Q_i K_j^\top}{\sqrt{d_{out}}}\right)}{\sum_{j'=1}^{n} exp\left(\frac{Q_i K_{j'}^\top}{\sqrt{d_{out}}}\right)} V_j \in \mathbb{R}^{d_{\text{out}}}$$

$$= \sum_{j=1}^{n} \frac{k(X_i,X_j)}{\sum_{j'=1}^{n} k(X_i,X_j)} v(X_j) \in \mathbb{R}^{d_{\text{out}}},$$

with $Q_i = W_Q X_i$, $K_j = W_K X_j$, $v(X_j) = W_V X_j$, k a non-negative kernel function: we get a kernel smoothing.

- Different choices for k suggest different transformers architectures [Tsai et al., 2019].
- We replaced $k(X_i,X_j)$ by $k(X_i,X_j) \times K_r(i,j)$. k based on node content, K_r based on node structural similarity.
- Related to relative positional encoding [Shaw et al., 2018].

II - Leveraging substructures via kernel embedding of paths



(from Chen et al., 2020)

II - Leveraging substructures via kernel embedding of paths

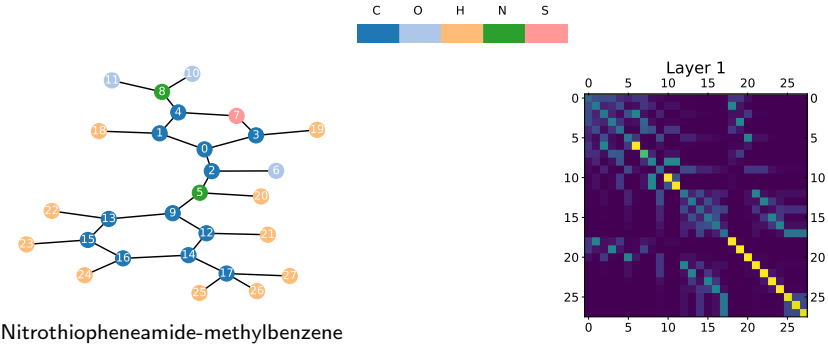
In practice.

- Encoding method from [Chen et al., 2020].
- We add the vector encoding the local substructure around node u to the feature vector of u at the transformer input.
- Similar approach in [Dosovitskiy et al., 2021].

GraphiT is able to outperform popular GNNs

Method / Dataset	MUTAG	PROTEINS	PTC	NCI1	ZINC (no edge feat.)
Size	188	1113	344	4110	12k
Max. number of nodes	28	620	109	111	37
GCN [Kipf and Welling, 2017]	78.9±10.1	75.8±5.5	54.0±6.3	75.9±1.6	0.367±0.011
GAT [Veličković et al., 2018]	80.3±8.5	74.8±4.1	55.0±6.0	76.8±2.1	0.384±0.007
GIN [Xu et al., 2019]	82.6±6.2	73.1±4.6	55.0±8.7	**81.7±1.7**	0.387±0.015
[Dwivedi and Bresson, 2021]	83.9±6.5	70.1±3.2	57.7±3.1	80.0±1.9	0.323±0.013
Transformers (T)	82.2±6.3	75.6±4.9	58.1±10.5	70.0±4.5	0.696±0.007
T + LapPE	85.8±5.9	74.6±2.7	55.6±5.0	74.6±1.9	0.507±0.003
T + Adj PE	87.2±9.8	72.4±4.9	59.9±5.9	79.7±2.0	0.243±0.005
T + 2-step RW kernel	85.3±6.9	72.8±4.5	**62.0±9.4**	78.0±1.5	0.243±0.010
T + 3-step RW kernel	83.3±6.3	**76.2±4.4**	61.0±6.2	77.6±3.6	0.244±0.011
T + Diffusion kernel	82.7±7.6	74.6±4.2	59.1±7.4	78.9±1.6	0.255±0.010
T + GCKN	84.4±7.8	69.5±3.8	61.5±5.8	78.1±5.1	0.274±0.011
T + GCKN + Adj PE	**90.5±7.0**	71.1±6.9	57.9±4.2	81.4±2.2	0.211±0.010
T + GCKN + Diffusion kernel	90.0±6.8	72.4±4.9	55.9±8.1	81.0±2.0	**0.197±0.002**

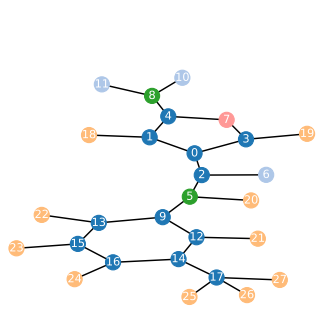
GraphiT seems to capture meaningful patterns



Nitrothiopheneamide-methylbenzene

Figure 2: *Left*: A molecule from the Mutagenicity data set [Kersting et al., 2016]. *Right*: approximate diffusion kernel for the molecular graph.

GraphiT seems to capture meaningful patterns



Nitrothiopheneamide-methylbenzene

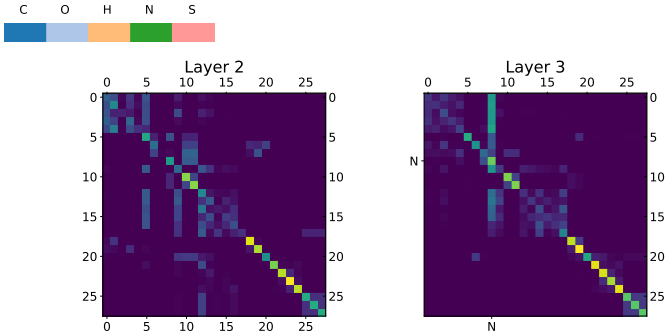
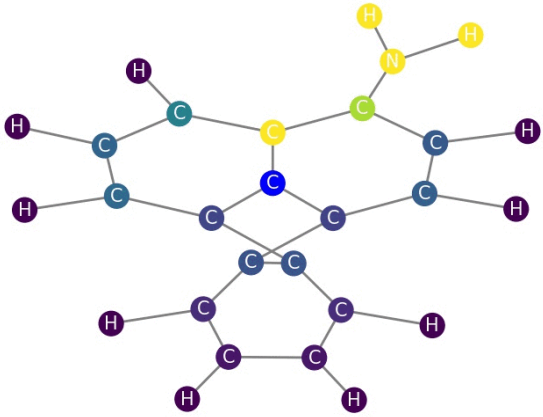
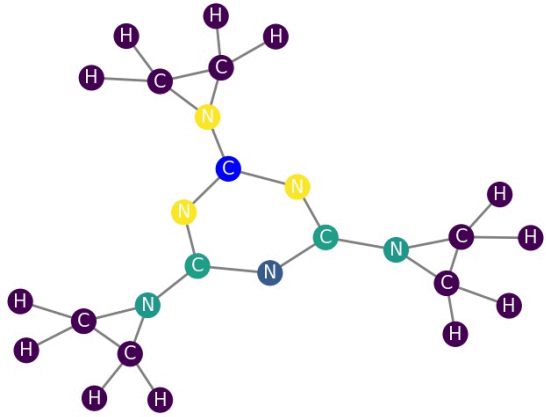


Figure 3: *Left*: A molecule from the Mutagenicity data set [Kersting et al., 2016]. *Right*: nodes 8 (N of NO_2) is salient. NO_2 group is known for its mutagenetic properties. The attention scores are averaged by heads.

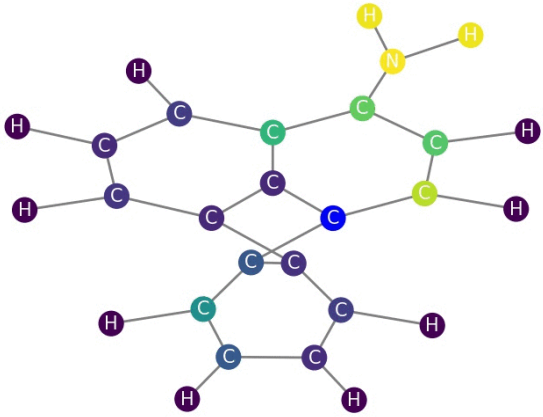
Attention from C atom



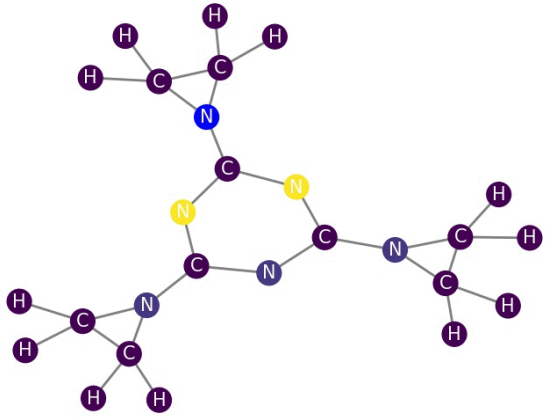
Attention from C atom



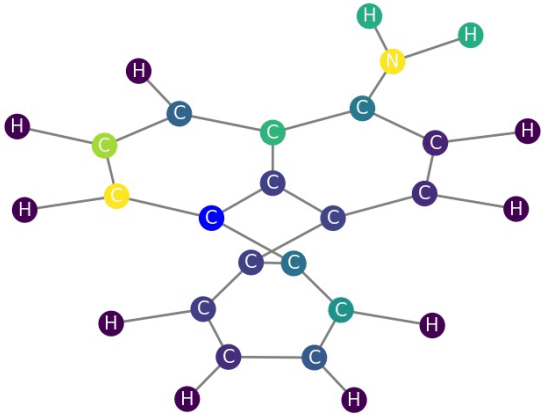
Attention from C atom



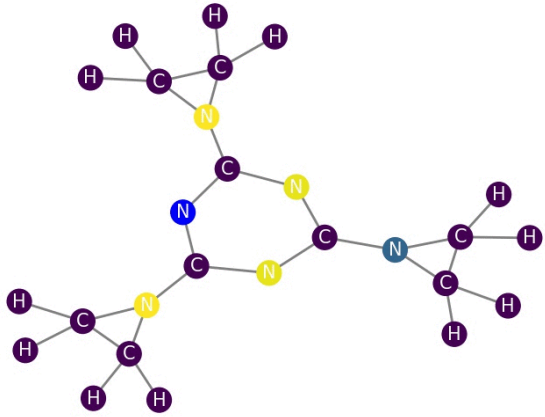
Attention from N atom



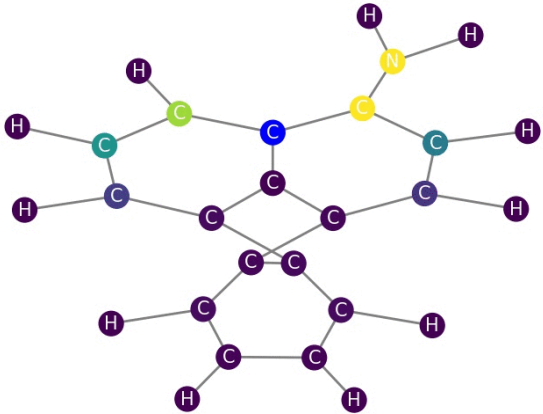
Attention from C atom



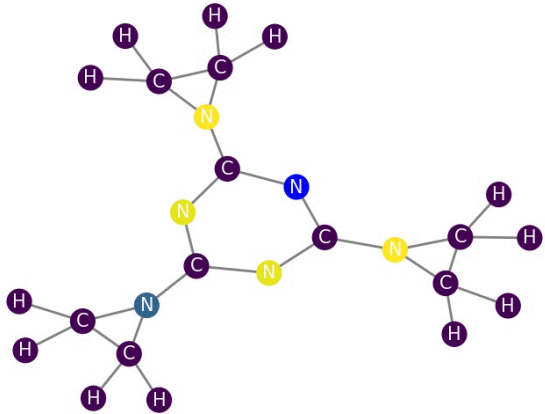
Attention from N atom



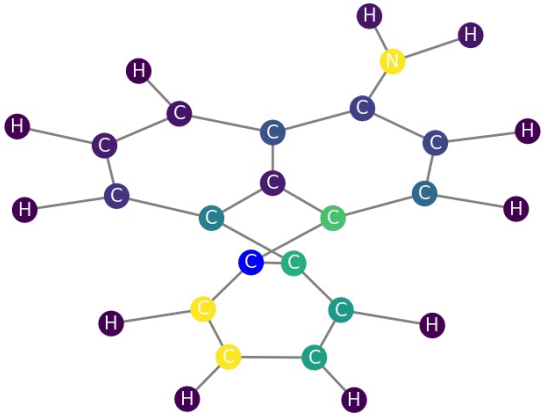
Attention from C atom



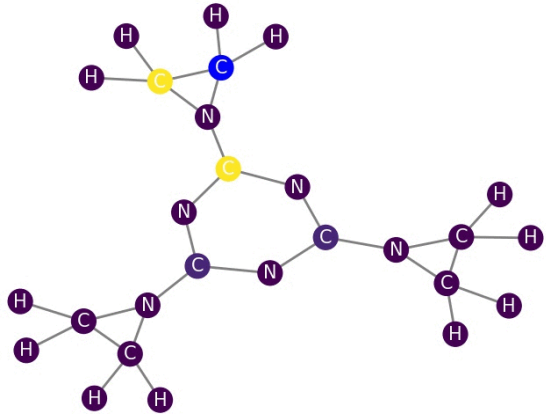
Attention from N atom



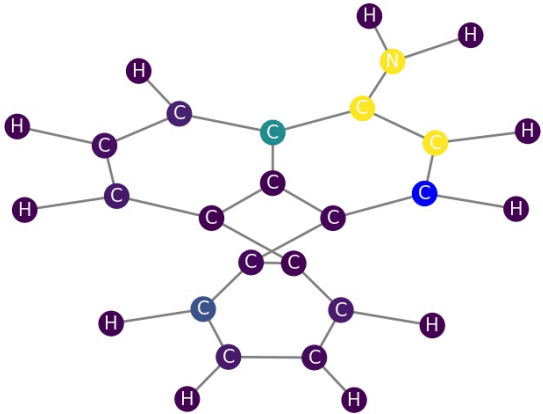
Attention from C atom



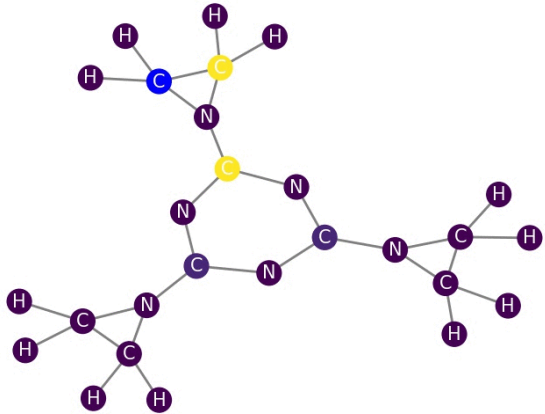
Attention from C atom



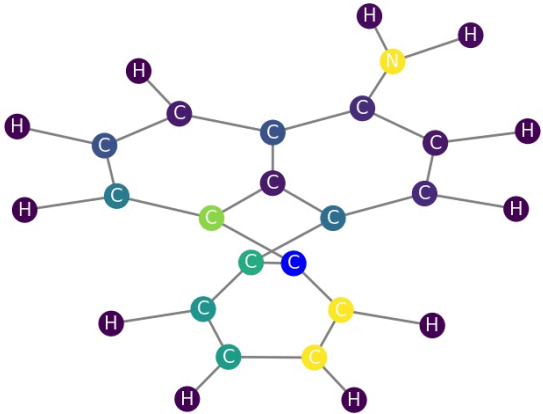
Attention from C atom



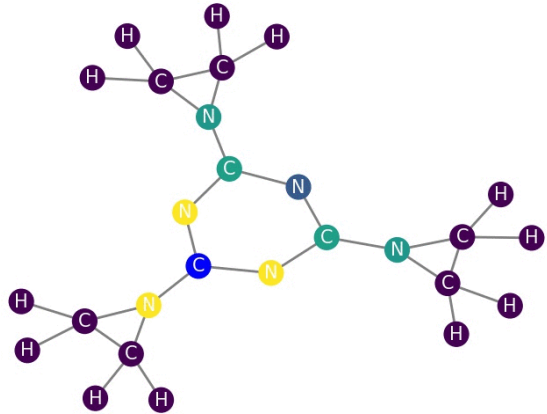
Attention from C atom



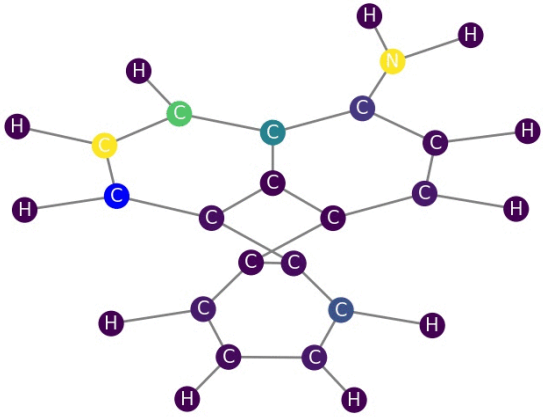
Attention from C atom



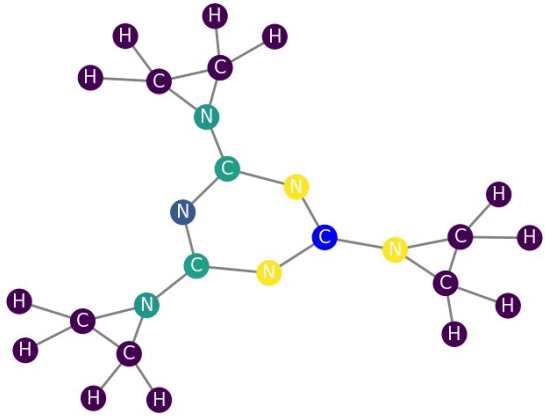
Attention from C atom



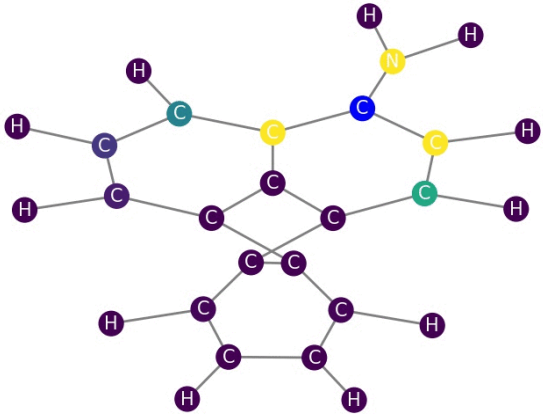
Attention from C atom



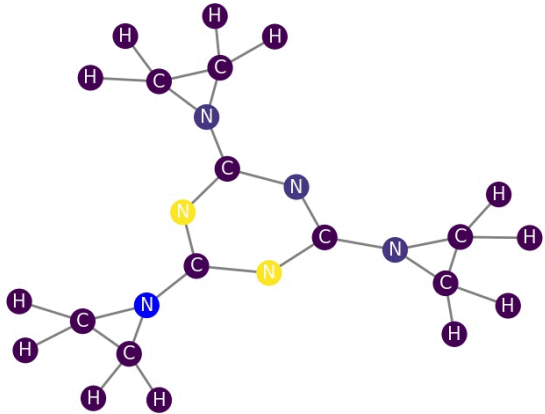
Attention from C atom



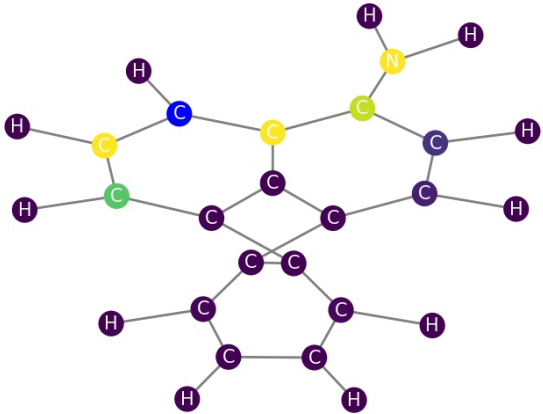
Attention from C atom



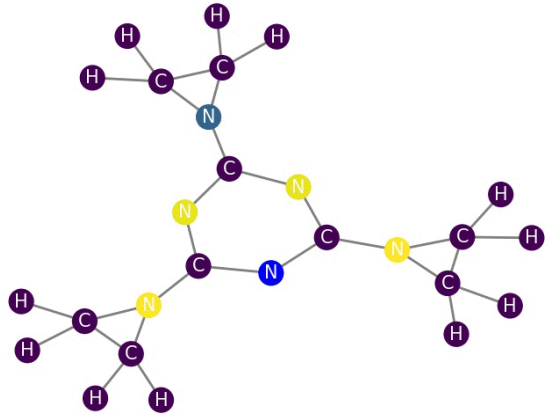
Attention from N atom



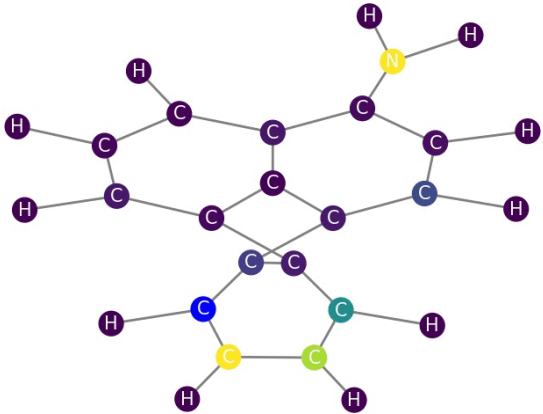
Attention from C atom



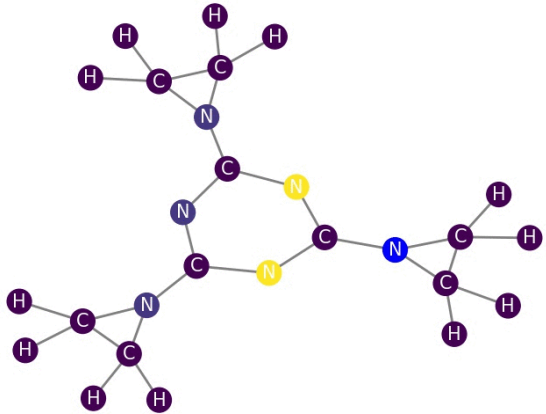
Attention from N atom



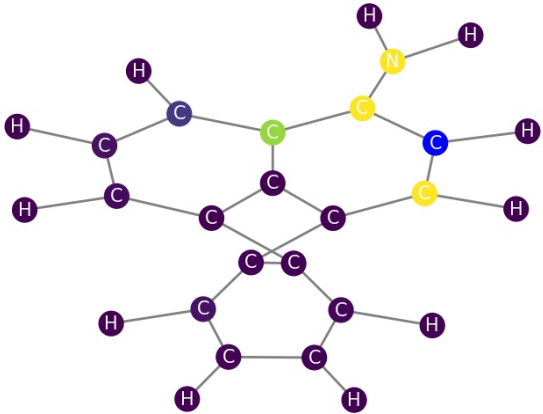
Attention from C atom



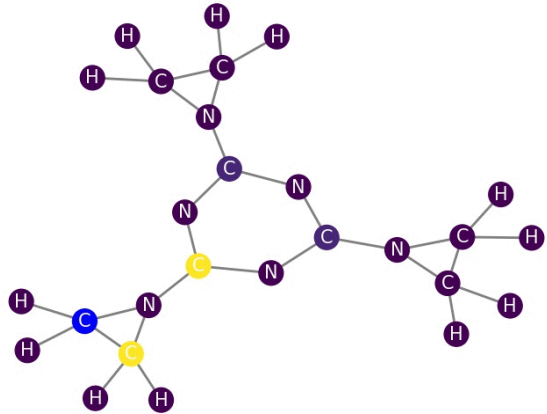
Attention from N atom



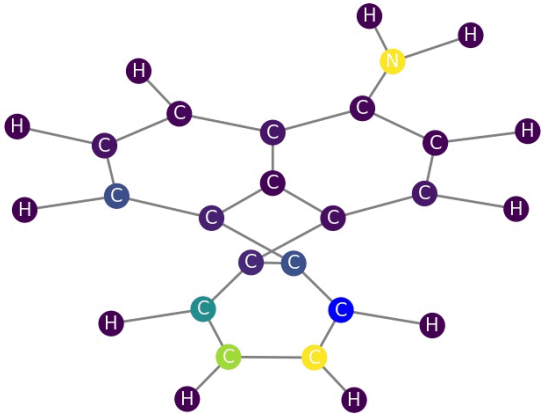
Attention from C atom



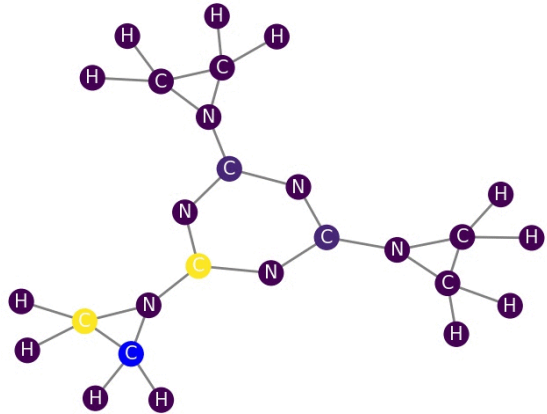
Attention from C atom



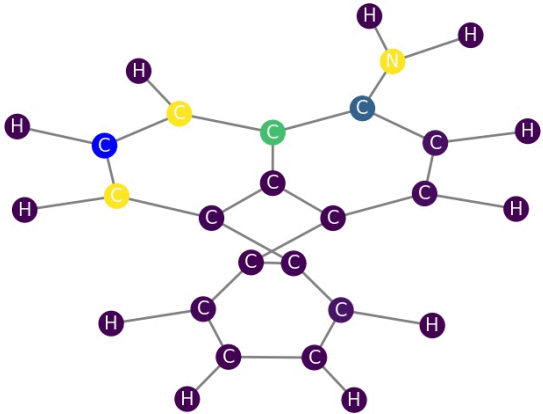
Attention from C atom



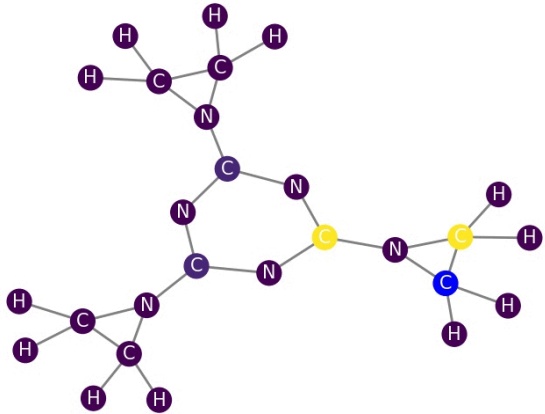
Attention from C atom



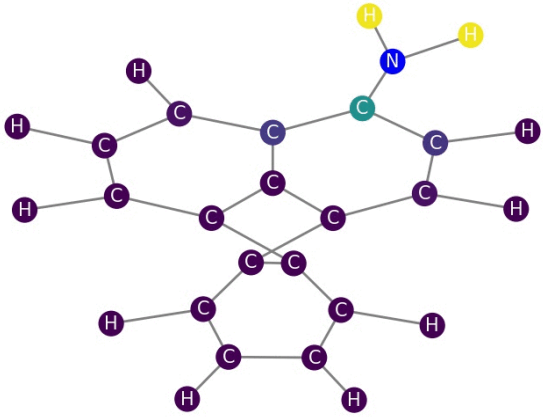
Attention from C atom



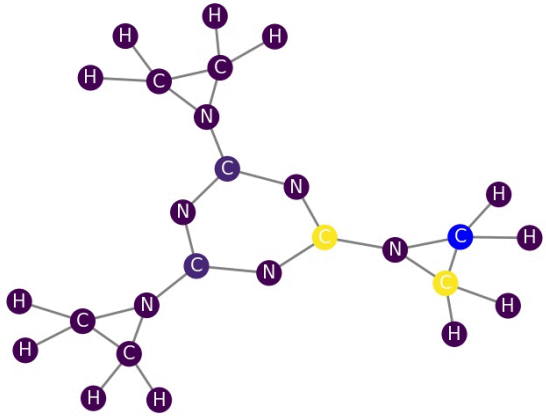
Attention from C atom



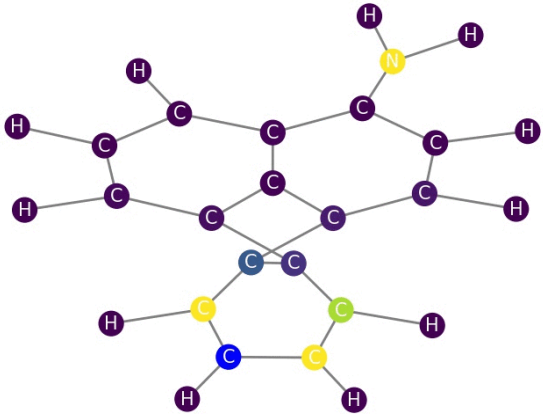
Attention from N atom



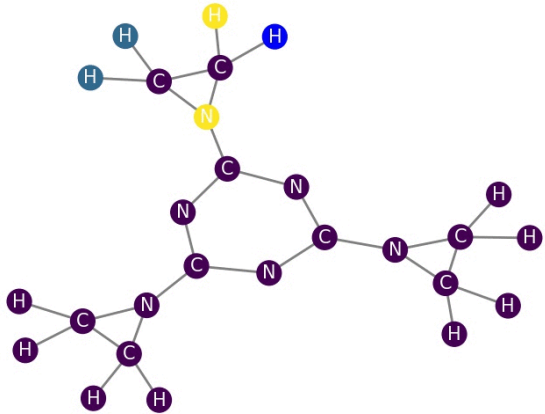
Attention from C atom



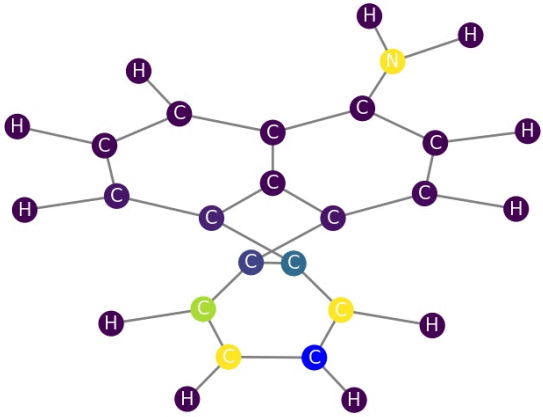
Attention from C atom



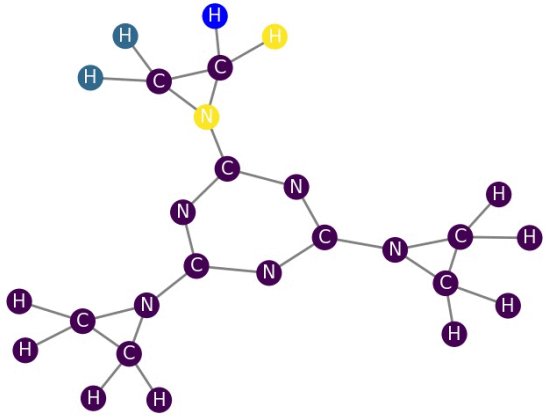
Attention from H atom



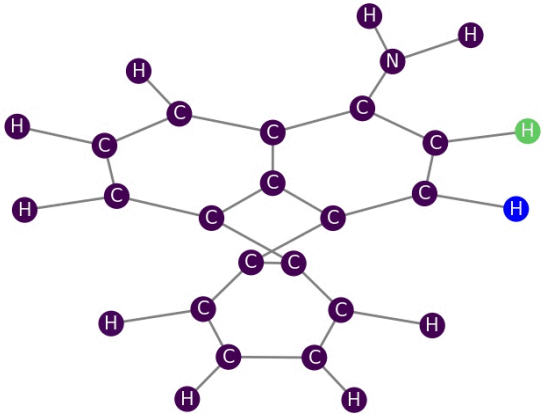
Attention from C atom



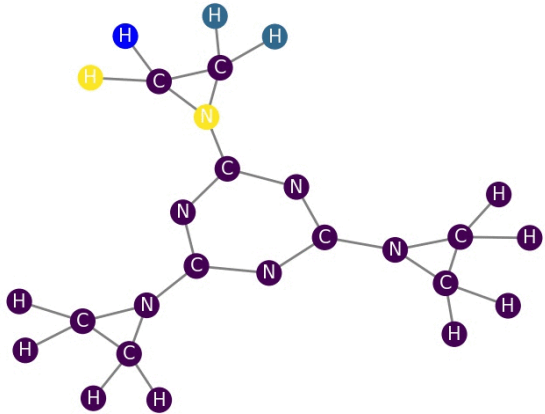
Attention from H atom



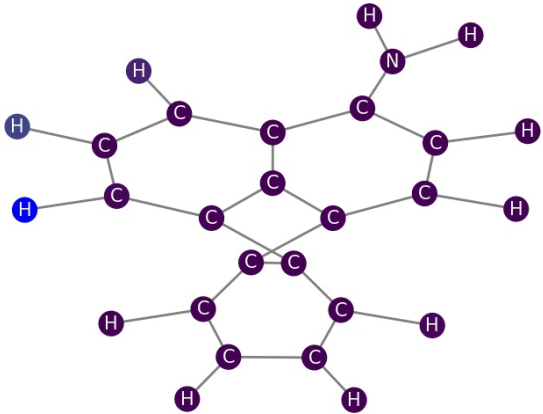
Attention from H atom



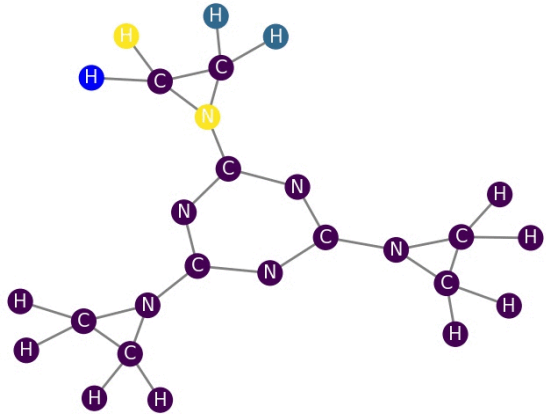
Attention from H atom



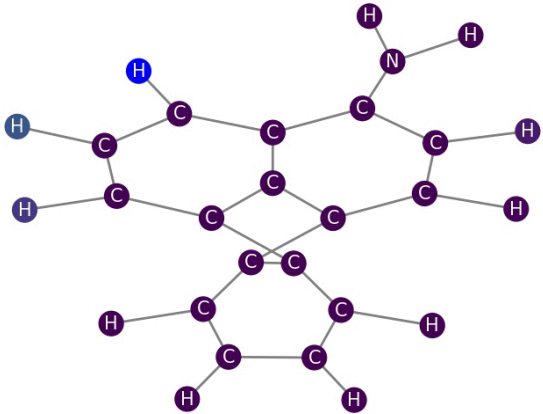
Attention from H atom



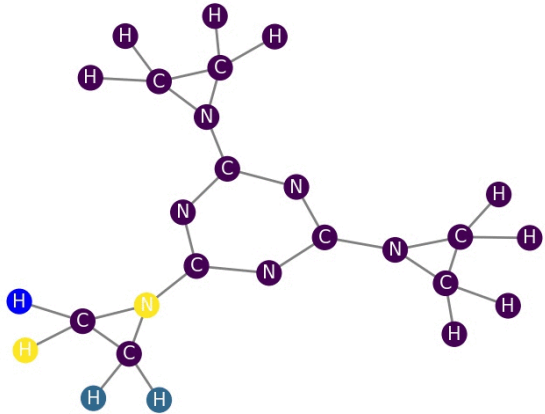
Attention from H atom



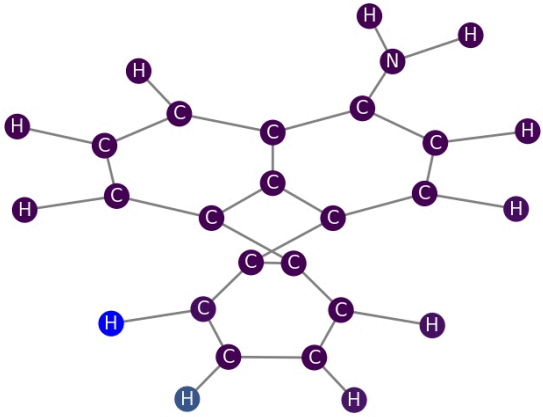
Attention from H atom



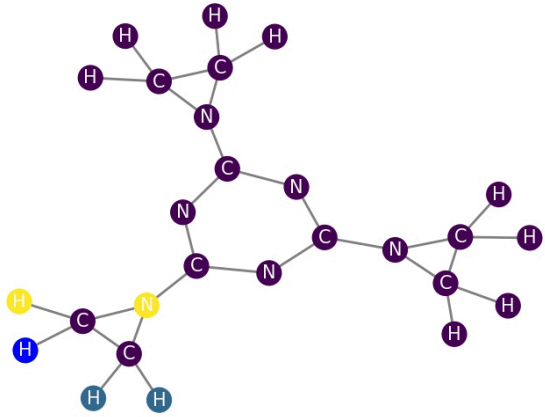
Attention from H atom



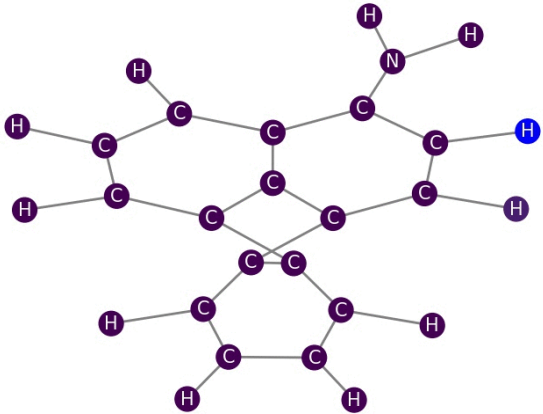
Attention from H atom



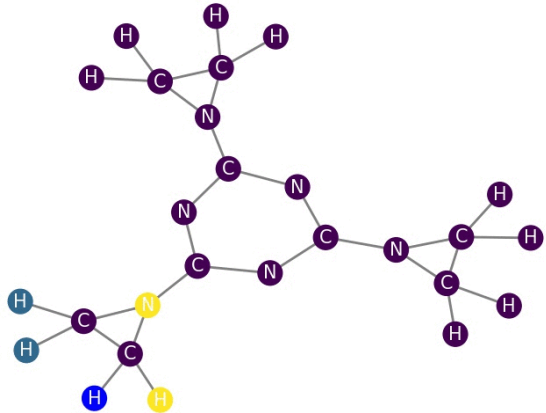
Attention from H atom



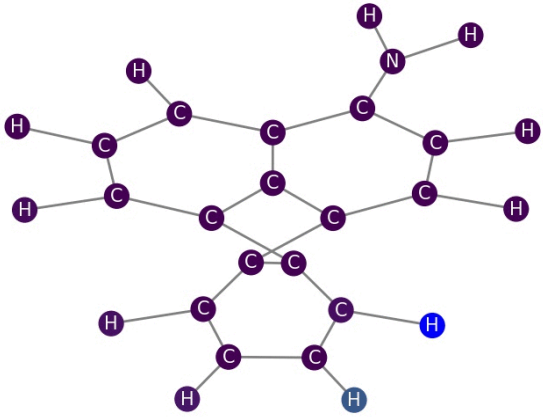
Attention from H atom



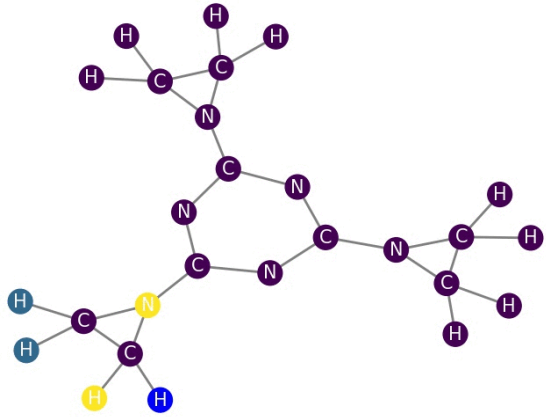
Attention from H atom



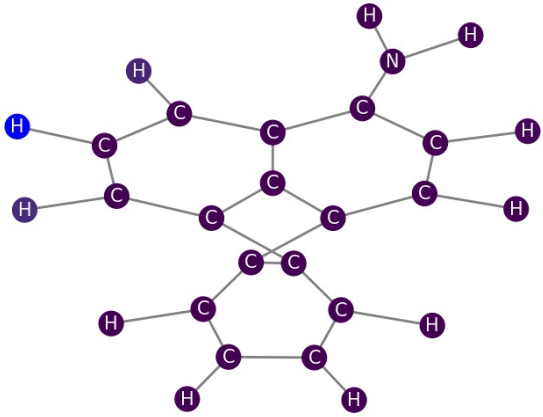
Attention from H atom



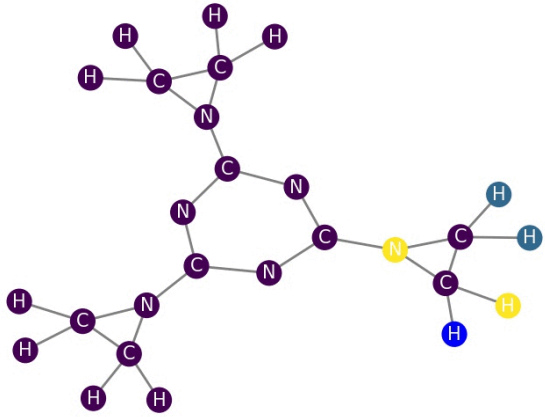
Attention from H atom



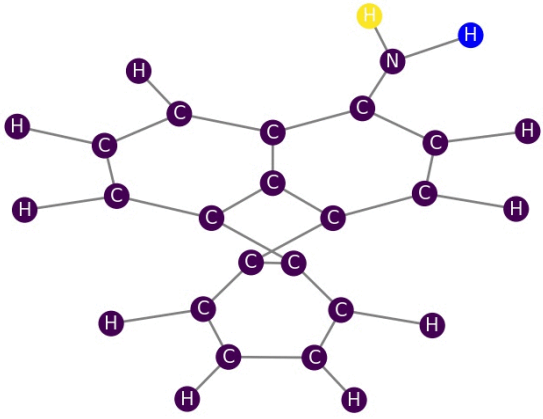
Attention from H atom



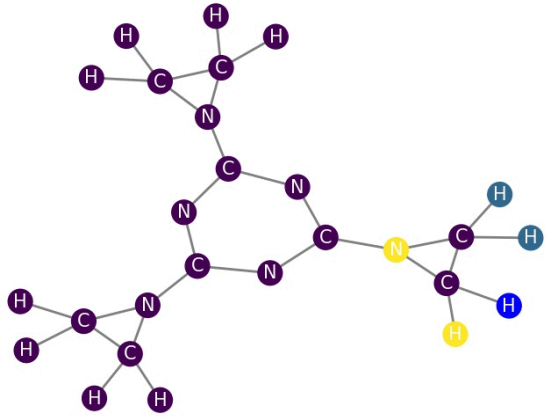
Attention from H atom



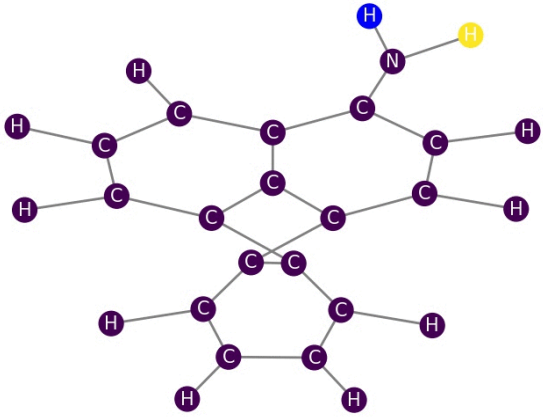
Attention from H atom



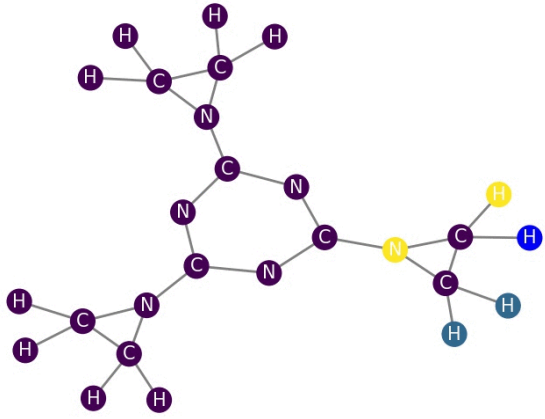
Attention from H atom



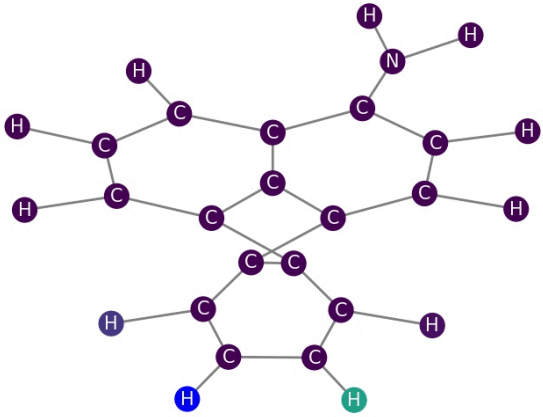
Attention from H atom



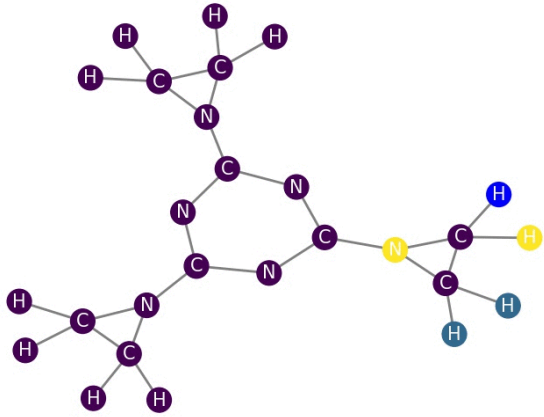
Attention from H atom



Attention from H atom



Attention from H atom



Conclusion

GraphiT.

- Inductive bias of transformers is valid with graphs.
- Attention provides promising intepretation tools.
- Paper available at `https://arxiv.org/abs/2106.05667`.
- Code available at `https://github.com/inria-thoth/GraphiT`.

References I

Borgwardt, K., Ghisu, E., Llinares-López, F., O'Bray, L., and Rieck, B. (2020). Graph kernels: State-of-the-art and future challenges. *arXiv preprint arXiv:2011.03854*.

Bronstein, M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. (2017). Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42.

Chen, D., Jacob, L., and Mairal, J. (2020). Convolutional kernel networks for graph-structured data. In *International Conference on Machine Learning (ICML)*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*.

Duvenaud, D., Maclaurin, D., Aguilera-Iparraguirre, J., Gómez-Bombarelli, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. (2015). Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems (NeurIPS)*.

References II

Dwivedi, V. P. and Bresson, X. (2021). A generalization of transformer networks to graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*.

Kersting, K., Kriege, N. M., Morris, C., Mutzel, P., and Neumann, M. (2016). Benchmark data sets for graph kernels.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Kondor, R. and Vert, J.-P. (2004). Diffusion kernels. In *Kernel Methods in Computational Biology*, pages 171–192. MIT Press.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2008). The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80.

Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., et al. (2020). Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710.

References III

Shaw, P., Uszkoreit, J., and Vaswani, A. (2018). Self-attention with relative position representations. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Smola, A. J. and Kondor, R. (2003). Kernels and regularization on graphs. In Schölkopf, B. and Warmuth, M. K., editors, *Learning Theory and Kernel Machines*, pages 144–158. Springer Berlin Heidelberg.

Tsai, Y.-H. H., Bai, S., Yamada, M., Morency, L.-P., and Salakhutdinov, R. (2019). Transformer dissection: A unified understanding of transformer's attention via the lens of kernel. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. In *International Conference on Learning Representations (ICLR)*.

Verma, N., Boyer, E., and Verbeek, J. (2018). Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.

References IV

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2019). How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*.