# Designing Transformers with Kernel Methods

Grégoire Mialon

FAIR

# About me

- 2016: Research intern at MIT, Nuclear Engineering
- 2017: Graduated from École polytechnique (Theoretical physics, Statistics, Computer Science)
- 2017-2018: NLP data scientist at eXplain, Paris
- 2018: Graduated from Msc MVA, ENS Paris-Saclay
- Since 2018: PhD student at Inria and ENS, Paris, advised by Julien Mairal and Alexandre d'Aspremont.

# What I have been doing in the past 3.5 years

**Kernel methods and deep learning in the small/medium data regime.**

- G. Mialon*, D. Chen*, M. Selosse*, J. Mairal. GraphiT: Encoding Graph Structure in Transformers (under review).
- G. Mialon*, D. Chen*, A. d'Aspremont, J. Mairal. A Trainable Optimal Transport Embedding for Feature Aggregation and its Relationship to Attention (ICLR, 2021).
- A. Bietti*, G. Mialon*, D. Chen, J. Mairal. A Kernel Perspective for Regularizing Deep Neural Networks (ICML, 2019).

**Convex optimization**

- G. Mialon, A. d'Aspremont, J. Mairal. Screening Data Points in Empirical Risk Minimization via Ellipsoidal Regions and Safe Loss Functions (AISTATS, 2020).

# What I want to talk about today

**Kernel methods and transformers in the small/medium data regime.**

- G. Mialon*, D. Chen*, M. Selosse*, J. Mairal. GraphiT: Encoding Graph Structure in Transformers (under review).
- G. Mialon*, D. Chen*, A. d'Aspremont, J. Mairal. A Trainable Optimal Transport Embedding for Feature Aggregation and its Relationship to Attention (ICLR, 2021).

# Motivation: designing strong models even when labelled data is scarce

**Learning with few data is one of the biggest problems in machine learning.**

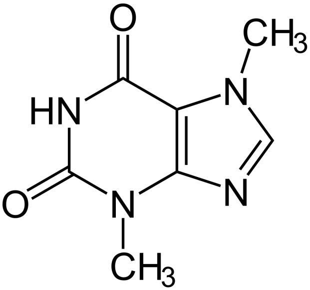
- A path towards better models.
- Or, simply because there is too few available data.

# How to design strong models in the small data regime?

# Outline: Encoding inductive bias within trainable architecture with kernel methods

1. **A new inductive bias for graphs**
2. Embedding sets with low data requirements
3. Conclusion and perspectives

# Graph data are an important research topic




A molecule of theobromin, or why chocolate is so good.

**Graph data are very valuable...**

- Proteins in computational biology.
- Molecules in chemoinformatics.
- Shapes in computer vision and computer graphics, etc.

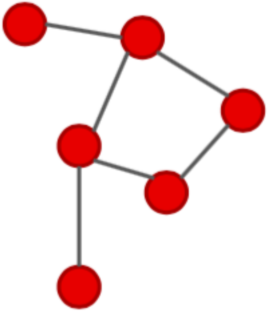
**... but delicate to handle.**

- Non-euclidean structure.

# Learning with Graph Neural Networks

**Graph Neural Networks (GNNs).**

- Introduced as an extension of neural networks for graph-structured data [Gori et al., 2005, Scarselli et al., 2008].
- Based on message passing.

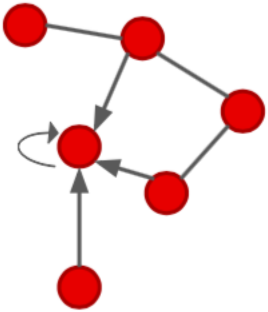


GNN, layer k

# Learning with Graph Neural Networks

**Graph Neural Networks (GNNs).**

- Introduced as an extension of neural networks for graph-structured data [Gori et al., 2005, Scarselli et al., 2008].
- Based on message passing.

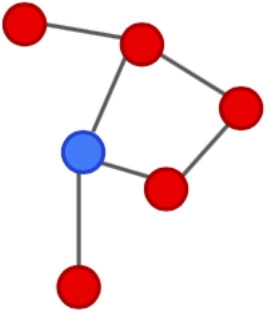


GNN, layer k

# Learning with Graph Neural Networks

**Graph Neural Networks (GNNs).**

- Introduced as an extension of neural networks for graph-structured data [Gori et al., 2005, Scarselli et al., 2008].
- Based on message passing.

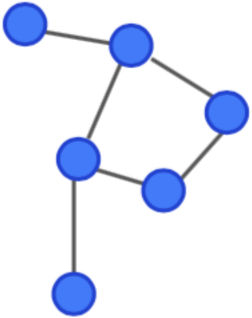


GNN, layer k+1 (for one node)

# Learning with Graph Neural Networks

**Graph Neural Networks (GNNs).**

- Introduced as an extension of neural networks for graph-structured data [Gori et al., 2005, Scarselli et al., 2008].
- Based on message passing.
- Many strategies to aggregate features of neighboring nodes [Duvenaud et al., 2015, Bronstein et al., 2017, Veličković et al., 2018].
- Applications for molecules [Duvenaud et al., 2015], physical systems [Battaglia et al., 2016], etc.

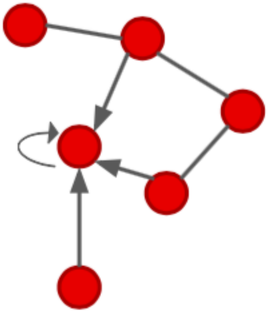


GNN, layer k+1

# Current limitations of GNNs

**In GNNs, messages flow between neighbors only.**

- Exploits the structure of the graph.
- But $n$ layers for $n$-hop neighbors to interact.
- Oversmoothing [Hamilton, 2020].



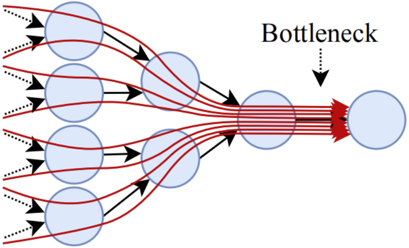
Only neighboring nodes communicate.

# Current limitations of GNNs

**In GNNs, messages flow between neighbors only.**

- Exploits the structure of the graph.
- But $n$ layers for $n$-hop neighbors to interact.
- Oversmoothing [Hamilton, 2020].
- Long-range interactions?
  Oversquashing [Alon and Yahav, 2021].

**Transformers perform global aggregation!**

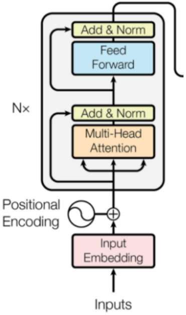



An illustration of oversquashing
(From Alon and Yahav, 2020).

# Transformers

**Why transformers?**

- Initially introduced in natural language processing [Vaswani et al., 2017, Devlin et al., 2019].
- Bioinformatics [Rives et al., 2019].
- Computer vision [Dosovitskiy et al., 2021].
- Transformers put into question the paradigm "one data modality, one preferred architecture".




Transformer encoder
(from Vaswani et al., 2017)

# Transformers

**Transformer 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.

# Kernel smoothing interpretation

**Self-attention as a kernel smoothing.**

- We can rewrite self-attention:

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

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

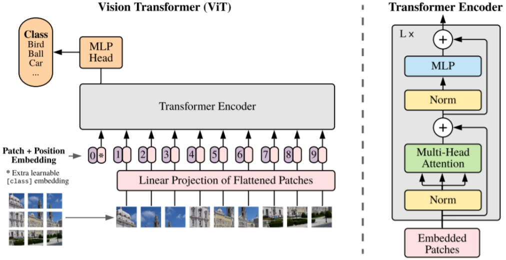
with $Q_i = W_Q X_i$, $K_j = W_K X_j$, $v(X_j) = W_V X_j$, $k$ a non-negative kernel function: we get a kernel smoothing.

**Different choices for $k$ suggest different transformers architectures [Tsai et al., 2019].**

# Transformers for graphs require position encoding

**Inductive bias of transformers for graphs?**

- All input elements are allowed to communicate.
- Self-attention layer is permutation equivariant.
- Hence, position encoding often required.
- Not trivial for graphs!

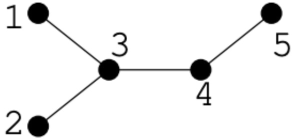



Vision transformer (from [Dosovitskiy et al., 2021])

# Previous attempts at using transformers with graphs

**Dwivedi & Bresson, 2021: absolute PE using Laplacian eigenvectors.**

- $A_{ij} = 1$ if two nodes are connected.
- Diagonal coefficients of $D$ are node degrees.



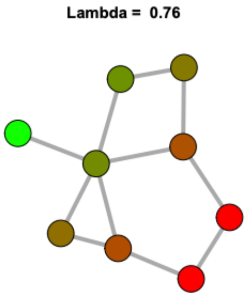

$$L = D - A = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

(From Vert, 2021)

# Previous attempts at using transformers with graphs

**Spectral graph analysis.**

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

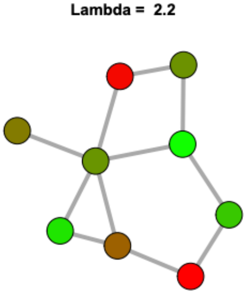



(From Vert, 2021)

# Previous attempts at using transformers with graphs

**Spectral graph analysis.**

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



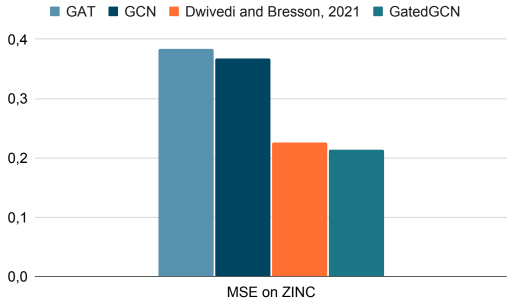

(From Vert, 2021)

# Promising results but...

**Problems with Laplacian absolute PE.**

- Flipping sign at training.
- Bounded by size of smallest graph.
- Do these vectors transfer between different graphs?

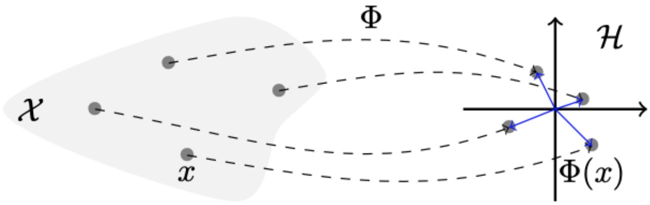
**Can we improve graph structure encoding in transformers?**



# Our contribution: GraphiT, or two mechanisms for encoding graph structure in transformers

**GraphiT: Encoding Graph Structure in Transformers**
G. Mialon, D. Chen, M. Selosse, J. Mairal, 2021
Under review.
`https://github.com/inria-thoth/GraphiT`

# Reminder: Kernel methods




(From Bietti, 2019)

**Learning with Kernel methods**

- Map data $x$ to high-dimensional space, $\Phi(x) \in \mathcal{H}$ (RKHS).
- $\Phi$ 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).

# Kernels on graphs

**Laplacian based kernels [Smola and Kondor, 2003].**

- Define a rich family of p.d. kernels on the graph by applying a regularization function $r$ to the spectrum of $L$

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

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.

# A famous kernel on graphs: the diffusion kernel

**Diffusion Kernel [Kondor and Vert, 2004].**

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

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

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

# Kernels on graphs reflect structural similarity between nodes

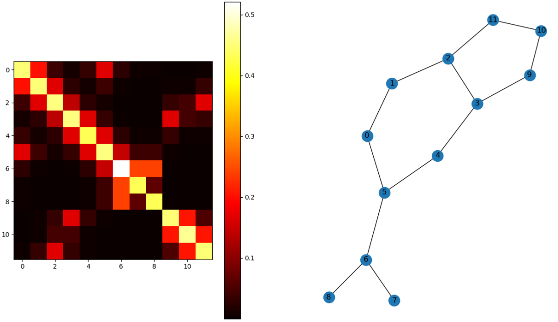



Figure 1: Diffusion kernel between the nodes of a MUTAG sample graph ($\beta = 1$).

**Use kernel matrix to modulate self-attention!**

# Mechanism 1: node position encoding with kernels on graphs

**Regular attention.**

- Self-attention layer

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

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

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

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

# Mechanism 1: node position encoding with kernels on graphs

**Modulated attention.**

- Self-attention layer becomes

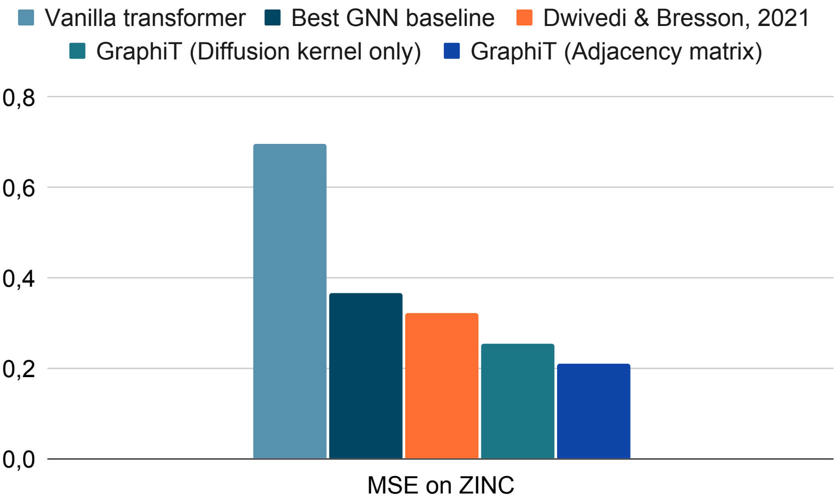
$$\text{PosAttention}(Q, V, K_r) = \text{normalize}\left(\exp\left(\frac{QQ^\top}{\sqrt{d_{\text{out}}}}\right) \odot K_r\right) V \in \mathbb{R}^{n \times d_{\text{out}}}, \tag{5}$$

  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:

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

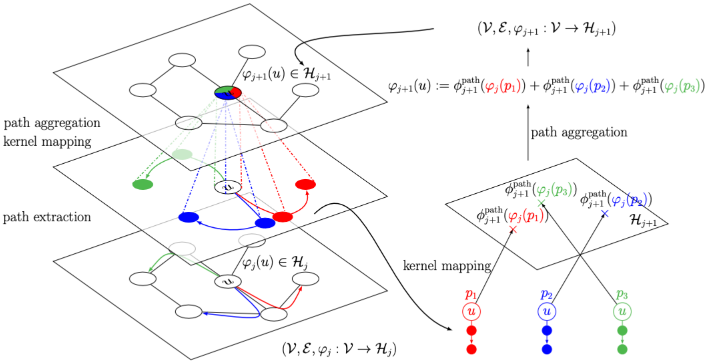
  with $D$ the matrix of node degrees and $K_r$ a kernel on the graph.

# Graph kernels allow effective position encoding



# Mechanism 2: leveraging substructures via path embedding

- Substructures: carry local positional information and content, *e.g* walks, subtrees, graphlets.
- Heavily used within graph kernels [Borgwardt et al., 2020].



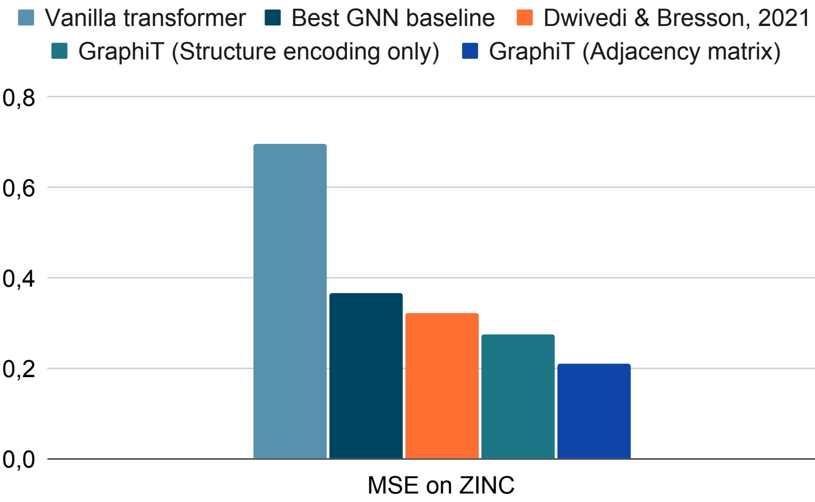

(from Chen et al., 2020)

# Leveraging substructures via path embedding

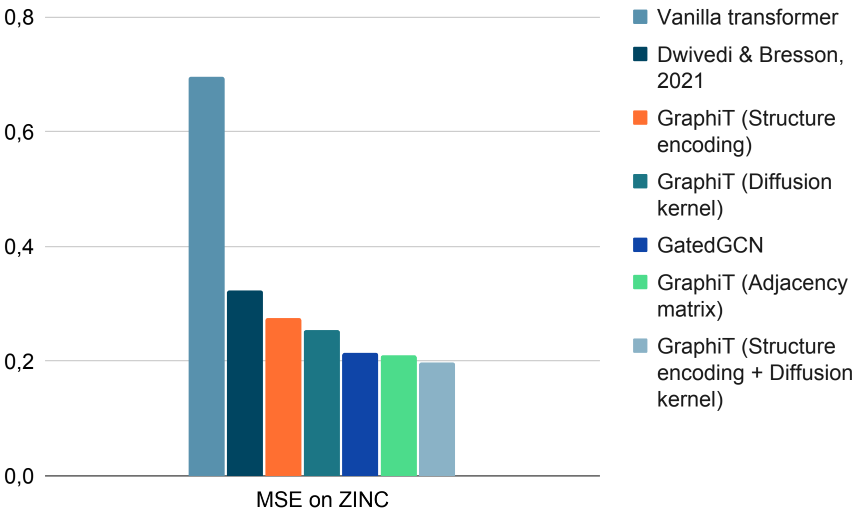
**In practice.**

- Encoding local substructure around node $u$ using method from [Chen et al., 2020].
- Add the resulting vector to the feature vector of $u$ at the transformer input.
- Somehow similar approach in [Dosovitskiy et al., 2021].

# Substructures are an effective inductive bias



# GraphiT is able to outperform popular GNNs



# GraphiT captures meaningful interactions

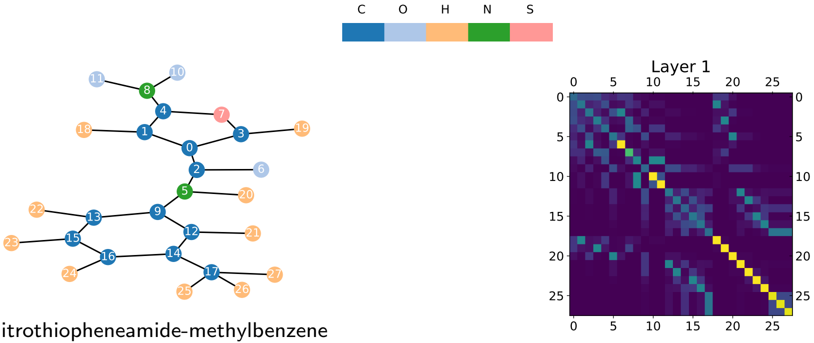



Figure 2: *Left*: A molecule from the Mutagenicity data set [Kersting et al., 2016]. *Right*: approximate diffusion kernel for the molecular graph.

# GraphiT captures meaningful interactions

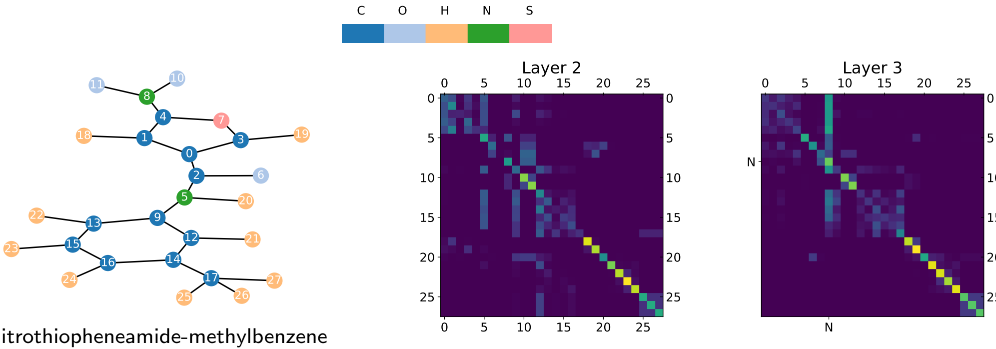
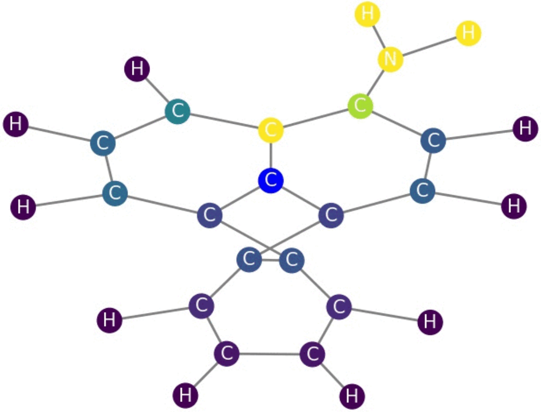
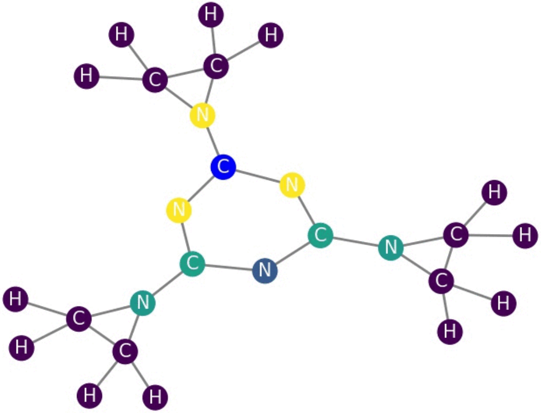



Figure 3: *Left*: A molecule from the Mutagenicity data set [Kersting et al., 2016]. *Right*: nodes 8 (N of $NO_2$) is salient. $NO_2$ group is known for its mutagenetic properties. The attention scores are averaged by heads.

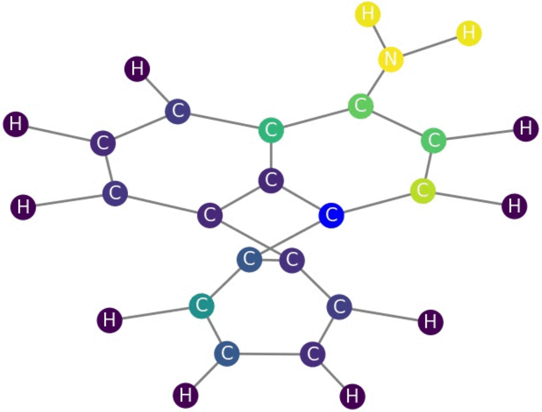
Attention from C atom



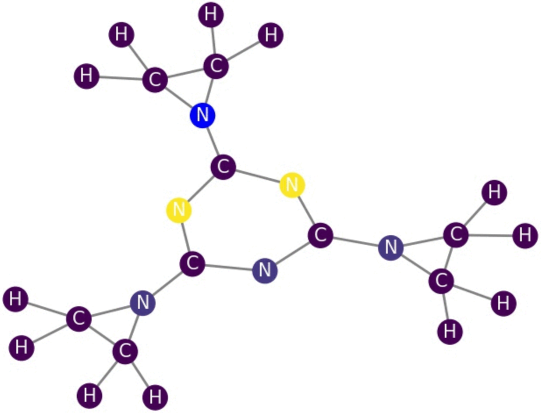
Attention from C atom



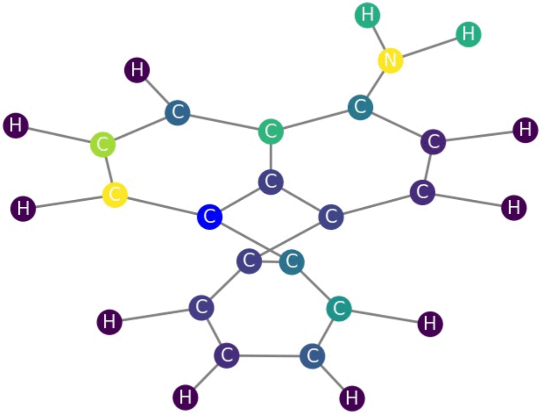
Attention from C atom



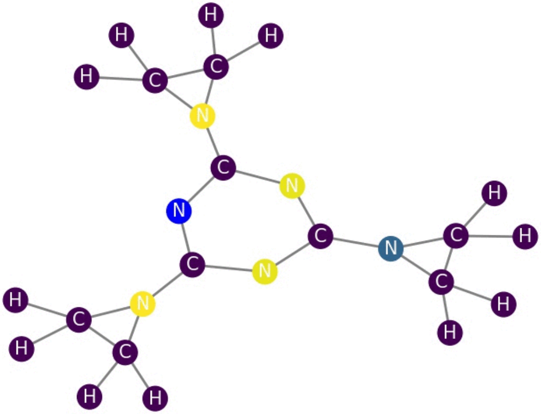
Attention from N atom



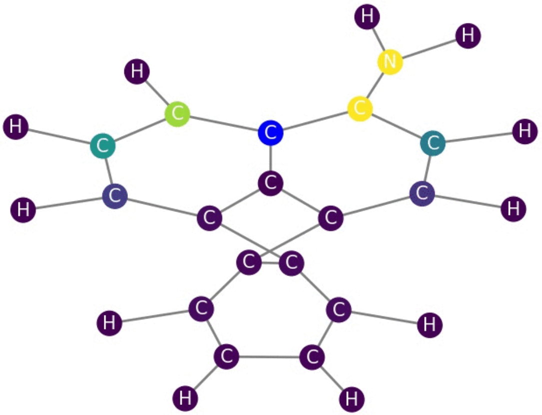
Attention from C atom



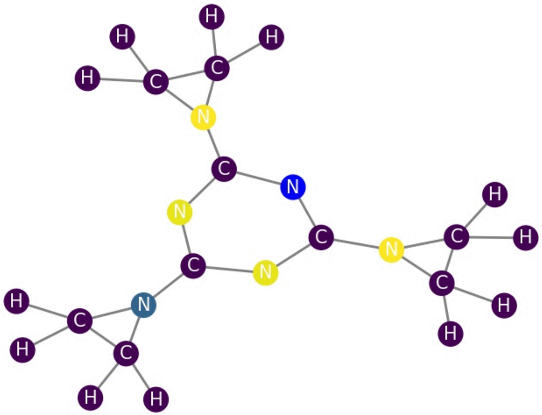
Attention from N atom



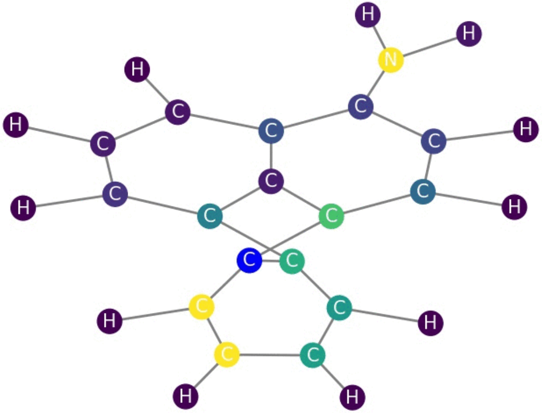
Attention from C atom



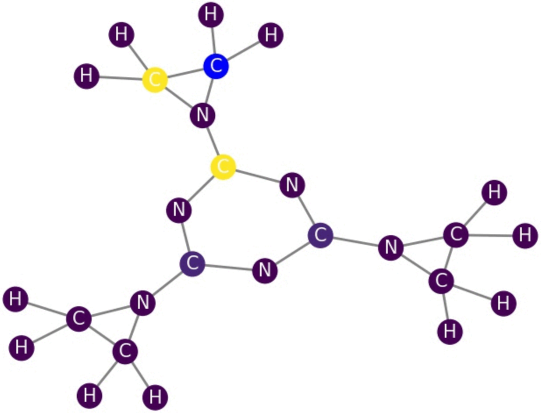
Attention from N atom



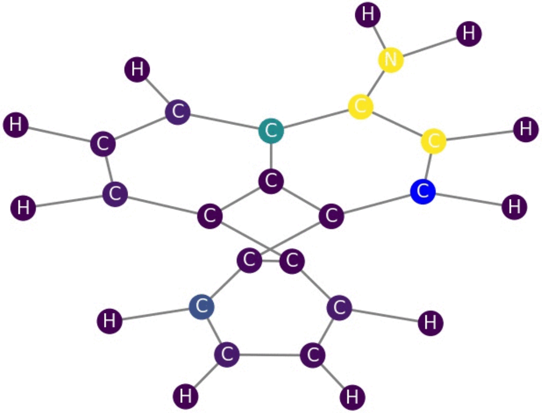
Attention from C atom



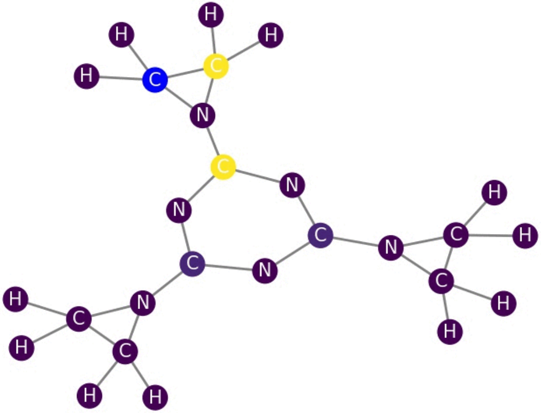
Attention from C atom



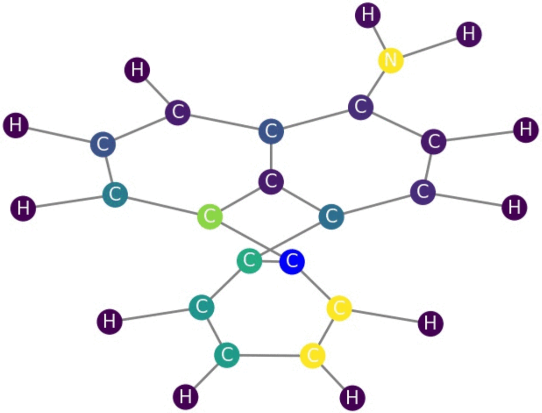
## Attention from C atom



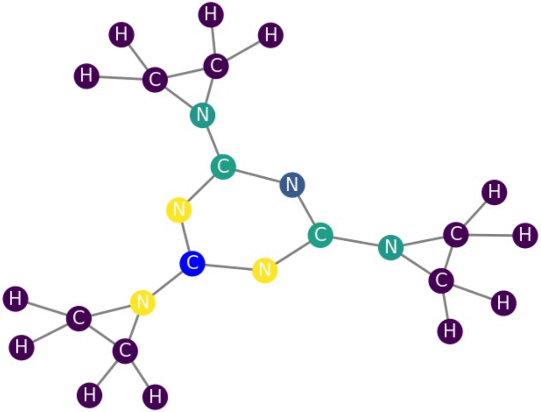
## Attention from C atom

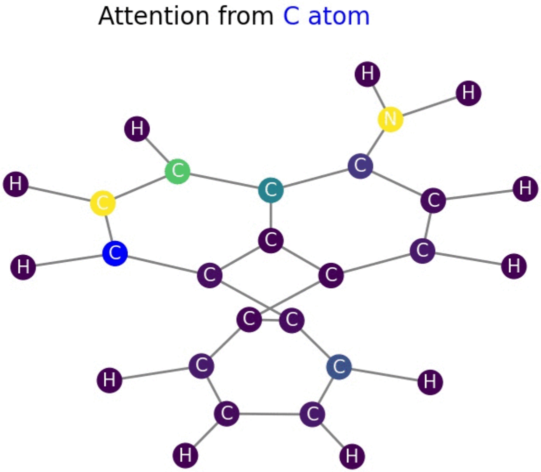


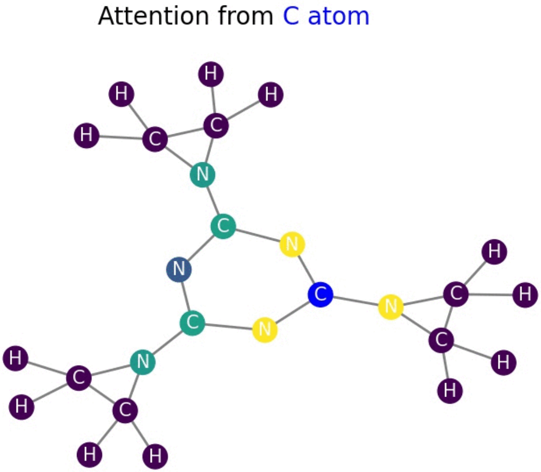
Attention from C atom



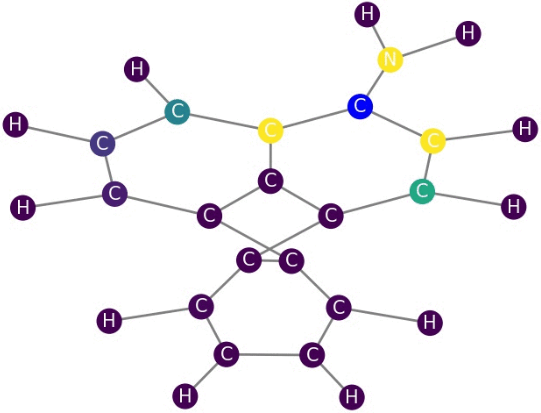
Attention from C atom



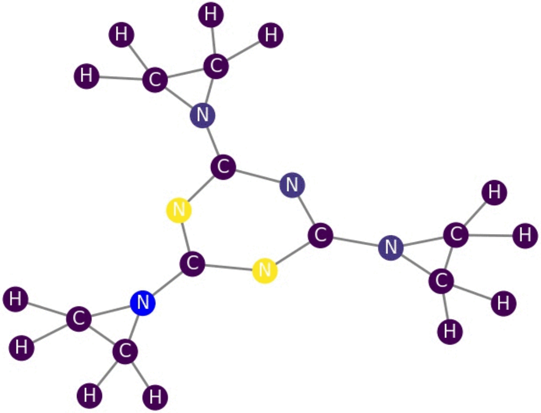

Attention from C atom


Attention from C atom

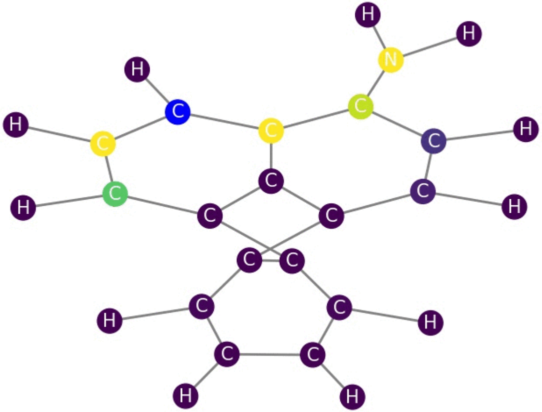
Attention from C atom



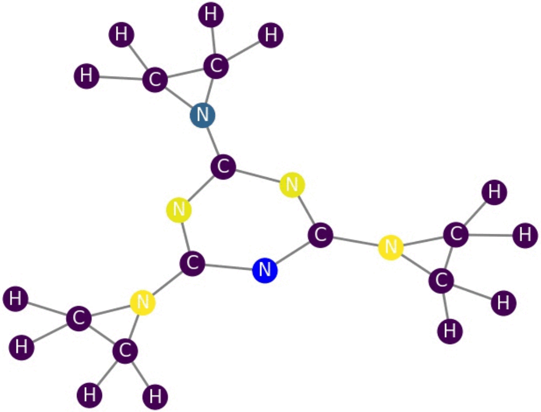
Attention from N atom



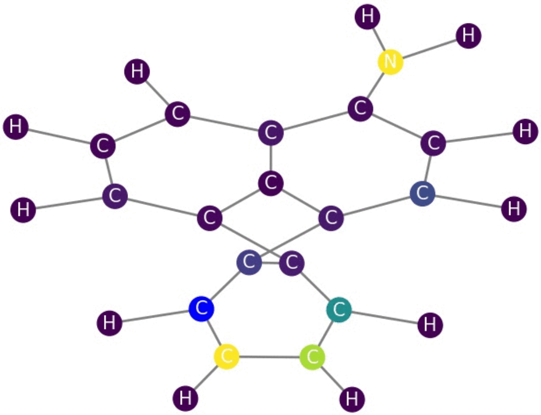
## Attention from C atom



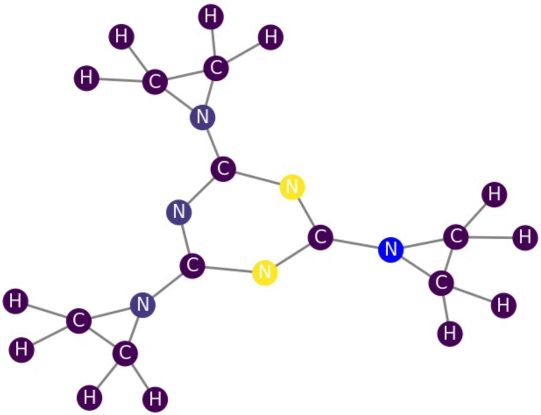
## Attention from N atom



Attention from C atom




Attention from N atom



# Inductive bias of GraphiT is competitive, but has its limitations

**Small scale datasets.**

- Varying results with graph kernel and parameter.

**Larger scale datasets: OGB [Hu et al., 2020].**

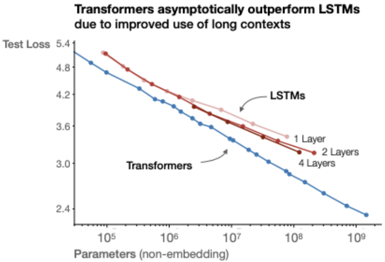
**Quadratic cost of self-attention.**

- What about large graphs?
- Recent line of work on efficient tranformers [Tay et al., 2020].

# Perspectives of GraphiT

**GraphiT opens exciting questions.**

- Scaling laws of graph transformers?
- Interactions with self-supervised learning for graphs? [Zhu et al., 2020, Thakoor et al., 2021]
- Leverage visualization for real-life applications?
- Graph transformers are currently an active domain of research [Ying et al., 2021, Kreuzer et al., 2021, Choromanski et al., 2021a].




(From [Kaplan et al., 2020]).

# Back to the smoothing formula

**Self-attention as a kernel smoothing.**

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

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.

- 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].

**What if we pick another similarity measure?**

# Outline

1. A new inductive bias for graphs
2. **Embedding sets with low data requirements**
3. Conclusion and perspectives

# Sets are also an important data modality

**Sets can be found in various domains.**

- 3D shape recognition (point clouds).
- Protein sequences (set of features where order matters) in computational biology.
- Sentences in NLP.

**Common characteristics with graphs.**

- Size may vary.
- Potential interactions between elements.

# Let's focus on biological sequences

**Biological sequences may pose more problems: SCOP 1.75 [Murzin et al., 1995].**

- Sequences may be long.
- Potentially few labelled sample per class.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CUU | GAC | AAA | GUU | GAG | GCU | GAA | GUG | CAA | AUU | GAU | AGG | UUG | AUC | ACA | GGC |
| L | D | K | V | E | A | E | V | Q | I | D | R | L | I | T | G |

Short part of mRNA sequence for the SARS-Cov-2 spike protein. Each triplet codes for an amino acid, represented below.

# Our sequences require specific embedding

**Existing methods do not yield satisfactory results for our data.**

- Kernel methods for sets [Lyu, 2004]: not expressive enough.
- Transformers, deep architectures for sets [Lee et al., 2019, Skianis et al., 2020]: empirically mixed results.

**How to represent sets with low data and memory requirements?**

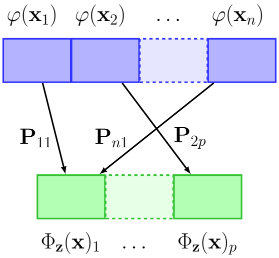
# Our contribution: OTK, a data-efficient embedding for sets

**A Trainable Optimal Transport Embedding for Feature Aggregation and its Relationship to Attention**
G. Mialon, D. Chen, A. d'Aspremont, J. Mairal
ICLR 2021.
`https://github.com/claying/OTK`

# OTK: a data-efficient embedding for sets

**Global, similarity-based pooling.**

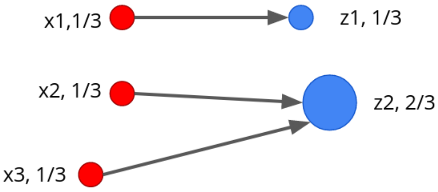
- Input set or sequence X.
- Element-wise, non-linear embedding $\varphi$.
- Pool elements $\varphi(x_i)$ in bins via weighted sums.
- To each p bin corresponds a prototype (parameter) $z_j, j = 1 \ldots p$.
- Pooling weights P reflect similarity between x and z.



# Our notion of similarity: optimal transport

**What is optimal transport?**

- Pooling weights P: optimal transport plan between X and Z.
- Most efficient way of transporting a mass distribution to another.
- Finding the transport plan minimizing a transportation cost.
- GPU-friendly solvers [Sinkhorn and Knopp, 1967, Cuturi and Doucet, 2013].




$$P = \begin{pmatrix} 1/3 & 0 & 0 \\ 0 & 1/3 & 1/3 \end{pmatrix}$$

# Our notion of similarity: optimal transport

**Two interpretations**

- Empirically: better than dot-product.
- Embeds the set in a space where $\ell_2$ distance approximates the 2-Wasserstein distance [Wang et al., 2013].
- Surrogate for a well-studied kernel [Rubner et al., 2000].

# Reasonable memory/data requirements

**Data efficient.**

- Z can be learned with or without supervision.

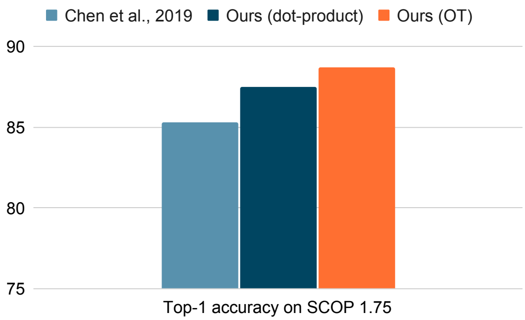
**A linearized variant of attention.**

- Kernel smoothing: we replaced $\frac{k(Q_i, K_j)}{\sum_{j'=1}^{n} k(Q_i, K_{j'})}$ by $\mathrm{P}_{\kappa}(X, Z)_{ij}$.
- Linear in the number of input elements.
- Similar ideas in efficient transformers [Wang et al., 2020, Choromanski et al., 2021b].

# OTKE: (temporarily) sota for two bioinformatics tasks

**SCOP 1.75**

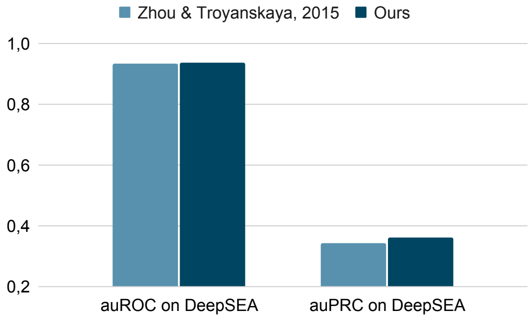
- Classify protein folding from amino-acid sequence.



# OTKE: (temporarily) sota for two bioinformatics tasks

**DeepSEA**

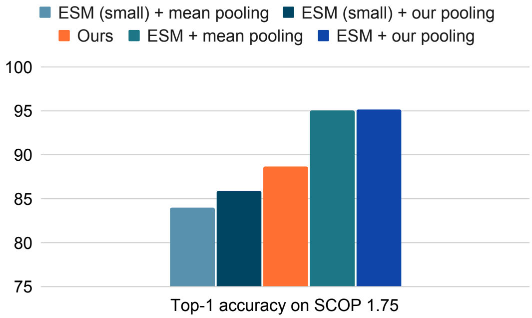
- Predicting chromatin profile from raw genomic sequence.



# What about pre-training?

**During ICLR rebuttal...**

- ESM [Rives et al., 2019], a transformer protein language model trained on 250m protein sequences.
- Simply training a linear layer on top of ESM features.



# Limitations and further work

**As an embedding.**

- Multi-layer version not trivial.
- Outperformed by available pre-trained model.

**As an adaptive pooling mechanism for deep architectures?**

- Improved pooling for graph representation [Kolouri et al., 2021] or protein representation (ICLR rebuttal).
- Dot-product instead of computationally more intensive OT?

# Outline

1. A new inductive bias for graphs
2. Embedding sets with low data requirements
3. **Conclusion and perspectives**

# How to design strong models in smaller data regimes?

**GraphiT**

- Inductive bias of transformers is valid with graph with small/medium scale datasets.
- Attention provides promising intepretation tools for graphs.

**OTK Embedding**

- Dealing with long sequences with few data.
- Interesting pooling mechanism connected to the recent line of work efficient transformers.
- Challenged by transfer learning from pre-trained models.

**Kernel methods**

- Reconciles deep learning with smaller data regimes!
- Understanding architectures via a new lens.

# Are inductive biases still useful?

**In the small data regime, inductive biases such as kernel methods are still useful!**

- Pre-training is not always possible.
- Not all domains have large data.
- Even when large unlabelled data is available, self-supervised learning frameworks may be not mature enough.

# Are inductive biases still useful?

**But in the large data regime?**

- OTK beaten by ESM [Rives et al., 2019].
- ResMLP [Touvron et al., 2021], DeiT [Touvron et al., 2020], BiT [Kolesnikov et al., 2020].
- In the large data regime, the bitter lesson of machine learning seems more true than ever.

# What about learning paradigms?

**Inductive biases are elsewhere.**

- Invariant Risk Minimization [Arjovsky et al., 2020].
- Data augmentation and loss in Self-supervised learning [He et al., 2020, Caron et al., 2020, Grill et al., 2020, Zbontar et al., 2021].

Thank you!

# References I


Alon, U. and Yahav, E. (2021). On the bottleneck of graph neural networks and its practical implications. In *International Conference on Learning Representations*.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. (2020). Invariant risk minimization.

Battaglia, P. W., Pascanu, R., Lai, M., Rezende, D. J., and Kavukcuoglu, K. (2016). Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems (NeurIPS)*.

# References II

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

Choromanski, K., Lin, H., Chen, H., and Parker-Holder, J. (2021a). Graph kernel attention transformers.

Choromanski, K. M., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J. Q., Mohiuddin, A., Kaiser, L., Belanger, D. B., Colwell, L. J., and Weller, A. (2021b). Rethinking attention with performers. In *International Conference on Learning Representations*.

Cuturi, M. and Doucet, A. (2013). Fast computation of wasserstein barycenters. In *International Conference on Machine Learning (ICML)*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

# References III


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)*.

Gori, M., Monfardini, G., and Scarselli, F. (2005). A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734 vol. 2.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. (2020). Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

# References IV


Hamilton, W. L. (2020). *Graph Representation Learning*.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. (2020). Open graph benchmark: Datasets for machine learning on graphs.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models.

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

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., and Houlsby, N. (2020). Big transfer (bit): General visual representation learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.

# References V


Kolouri, S., Naderializadeh, N., Rohde, G. K., and Hoffmann, H. (2021). Wasserstein embedding for graph learning. 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.

Kreuzer, D., Beaini, D., Hamilton, W. L., Létourneau, V., and Tossou, P. (2021). Rethinking graph transformers with spectral attention.

Lee, J., Lee, Y., Kim, J., Kosiorek, A. R., Choi, S., and Teh, Y. W. (2019). Set transformer: A framework for attention-based permutation invariant neural networks. In *International Conference on Machine Learning (ICML)*.

Lyu, S. (2004). Mercer kernels for object recognition with local features. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Murzin, A. G., Brenner, S. E., Hubbard, T., Chothia, C., et al. (1995). Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247(4):536–540.

# References VI


Rives, A., Goyal, S., Meier, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., and Fergus, R. (2019). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. In *bioRxiv 622803*.

Rubner, Y., Tomasi, C., and Guibad, L. J. (2000). The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:99–121.

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.

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)*.

Sinkhorn, R. and Knopp, P. (1967). Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2).

# References VII

Skianis, K., Nikolentzos, G., Limnios, S., and Vazirgiannis, M. (2020). Rep the set: Neural networks for learning set representations. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

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.

Tay, Y., Dehghani, M., Bahri, D., and Metzler, D. (2020). Efficient transformers: A survey.

Thakoor, S., Tallec, C., Azar, M. G., Munos, R., Velickovic, P., and Valko, M. (2021). Bootstrapped representation learning on graphs.

Touvron, H., Bojanowski, P., Caron, M., Cord, M., El-Nouby, A., Grave, E., Izacard, G., Joulin, A., Synnaeve, G., Verbeek, J., and Jégou, H. (2021). Resmlp: Feedforward networks for image classification with data-efficient training.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. (2020). Training data-efficient image transformers distillation through attention. *arXiv preprint arXiv:2012.12877*.

# References VIII


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)*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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

Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. (2020). Linformer: Self-attention with linear complexity.

Wang, W., Slepcev, D., Basu, S., Ozolek, J. A., and Rohde, G. K. (2013). A linear optimal transportation framework for quantifying and visualizing variations in sets of images. *International Journal of Computer Vision*, 101(2):254–269.

# References IX


Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T. (2021). Do transformers really perform bad for graph representation?

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. (2021). Barlow twins: Self-supervised learning via redundancy reduction.

Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., and Wang, L. (2020). Deep graph contrastive representation learning.