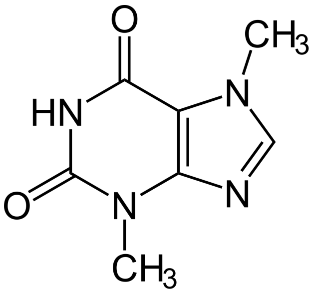
Transformers on graphs: challenge and perspectives

Grégoire Mialon

SGDMAL, Rennes. 21 juin 2022

Joint work with: Dexiong Chen (ETH Zürich), Margot Selosse, Julien Mairal (Inria).

Graph data are an important research topic

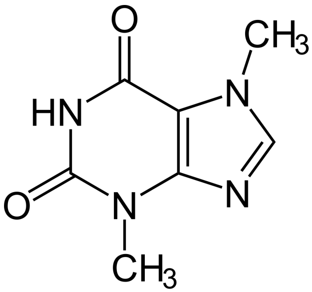


A molecule of theobromin, or why chocolate makes us feel good.

Graph data are very valuable...

- Molecules in chemoinformatics.

Graph data are an important research topic

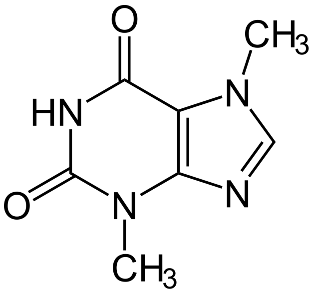


A molecule of theobromin, or why chocolate makes us feel good.

Graph data are very valuable...

- Molecules in chemoinformatics.
- Proteins in computational biology.

Graph data are an important research topic

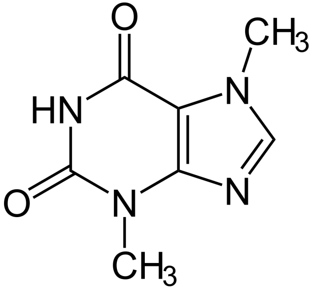


A molecule of theobromin, or why chocolate makes us feel good.

Graph data are very valuable...

- Molecules in chemoinformatics.
- Proteins in computational biology.
- Physical systems, *e.g*, particle interaction.

Graph data are an important research topic



A molecule of theobromin, or why chocolate makes us feel good.

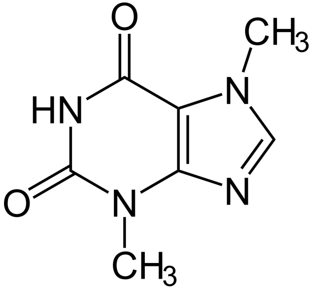
Graph data are very valuable...

- Molecules in chemoinformatics.
- Proteins in computational biology.
- Physical systems, *e.g*, particle interaction.

...but delicate to exploit.

- Non-Euclidean structure.

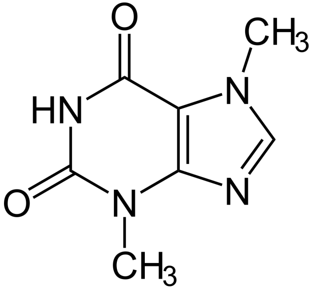
Success and current limits of neural networks for graphs



A molecule of theobromin, or why chocolate makes us feel good.

Graph neural networks [Gori et al., 2005, Scarselli et al., 2008] (GNNs), very active research topic.

Success and current limits of neural networks for graphs

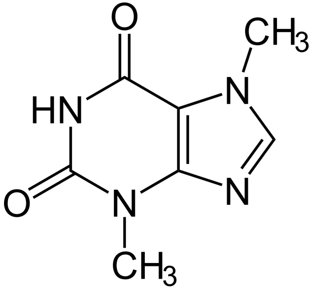


A molecule of theobromin, or why chocolate makes us feel good.

Graph neural networks [Gori et al., 2005, Scarselli et al., 2008] (GNNs), very active research topic.

- Direct connections between **neighboring nodes only**.

Success and current limits of neural networks for graphs

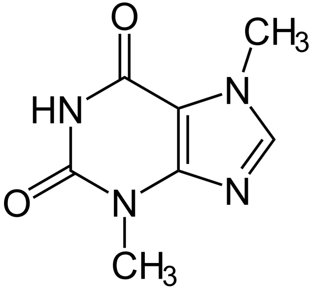


A molecule of theobromin, or why chocolate makes us feel good.

Graph neural networks [Gori et al., 2005, Scarselli et al., 2008] (GNNs), very active research topic.

- Direct connections between **neighboring nodes only**.
- Success of GNNs (Molecules [Duvenaud et al., 2015], physical systems [Battaglia et al., 2016], materials [Xie et al., 2021]).

Success and current limits of neural networks for graphs

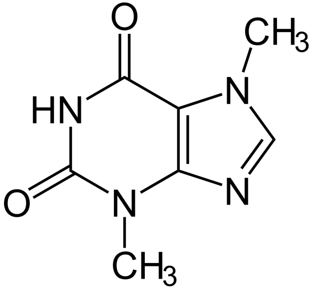


A molecule of theobromin, or why chocolate makes us feel good.

Graph neural networks [Gori et al., 2005, Scarselli et al., 2008] (GNNs), very active research topic.

- Direct connections between **neighboring nodes only**.
- Success of GNNs (Molecules [Duvenaud et al., 2015], physical systems [Battaglia et al., 2016], materials [Xie et al., 2021]).
- Current limitations of GNNs ([Li et al., 2018, Alon and Yahav, 2021]).

Success and current limits of neural networks for graphs



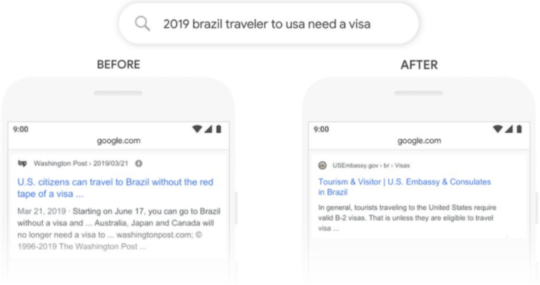
A molecule of theobromin, or why chocolate makes us feel good.

Graph neural networks [Gori et al., 2005, Scarselli et al., 2008] (GNNs), very active research topic.

- Direct connections between **neighboring nodes only**.
- Success of GNNs (Molecules [Duvenaud et al., 2015], physical systems [Battaglia et al., 2016], materials [Xie et al., 2021]).
- Current limitations of GNNs ([Li et al., 2018, Alon and Yahav, 2021]).

Let us connect all the nodes!

Transformers: a scalable, multi-purpose architecture



Improved web search engines.

"Vibrant portrait painting of Salvador Dalí with a robotic half face".

Transformers for graph are tempting but not straightforward

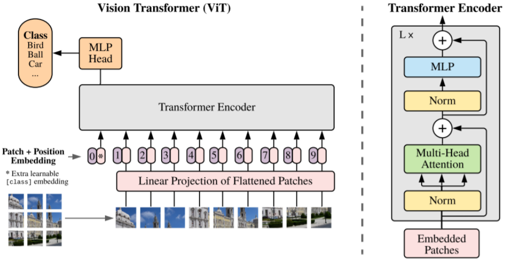


Image transformer (from [Dosovitskiy et al., 2021]).
Input: image seen as a set of patches.
Output: class label.

Success of transformers [Vaswani et al., 2017].

- Text [Devlin et al., 2019], Proteins [Rives et al., 2019], Images [Dosovitskiy et al., 2021].

Transformers for graph are tempting but not straightforward

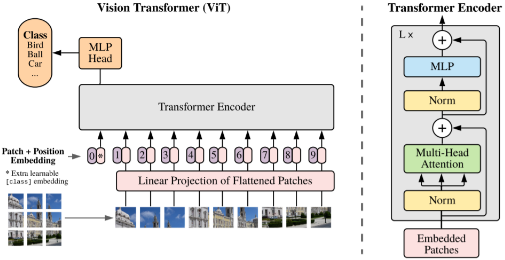


Image transformer (from [Dosovitskiy et al., 2021]).
Input: image seen as a set of patches.
Output: class label.

Success of transformers [Vaswani et al., 2017].

- Text [Devlin et al., 2019], Proteins [Rives et al., 2019], Images [Dosovitskiy et al., 2021].
- Rarely used for graphs.

Transformers for graph are tempting but not straightforward

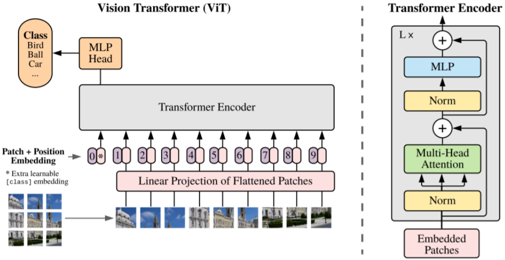


Image transformer (from [Dosovitskiy et al., 2021]).
Input: image seen as a set of patches.
Output: class label.

Success of transformers [Vaswani et al., 2017].

- Text [Devlin et al., 2019],
  Proteins [Rives et al., 2019],
  Images [Dosovitskiy et al., 2021].
- Rarely used for graphs.

A nice inductive bias for graphs?

- All input elements communicate...

Transformers for graph are tempting but not straightforward

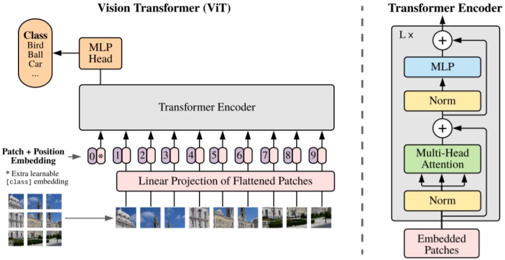


Image transformer (from [Dosovitskiy et al., 2021]).
Input: image seen as a set of patches.
Output: class label.

Success of transformers [Vaswani et al., 2017].

- Text [Devlin et al., 2019], Proteins [Rives et al., 2019], Images [Dosovitskiy et al., 2021].
- Rarely used for graphs.

A nice inductive bias for graphs?

- All input elements communicate...
- ...but model **blind to the input structure**.

Transformers for graph are tempting but not straightforward

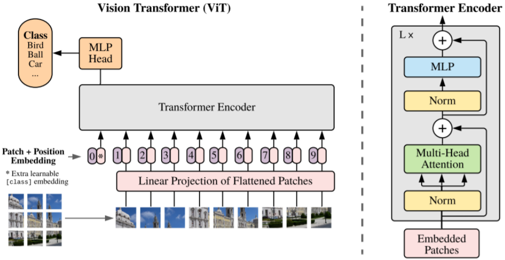


Image transformer (from [Dosovitskiy et al., 2021]).
Input: image seen as a set of patches.
Output: class label.

Success of transformers [Vaswani et al., 2017].

- Text [Devlin et al., 2019],
  Proteins [Rives et al., 2019],
  Images [Dosovitskiy et al., 2021].
- Rarely used for graphs.

A nice inductive bias for graphs?

- All input elements communicate...
- ...but model **blind to the input structure**.
- Hence, position encoding often required.

Transformers for graph are tempting but not straightforward

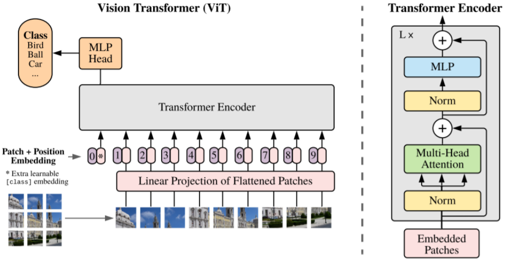


Image transformer (from [Dosovitskiy et al., 2021]).
Input: image seen as a set of patches.
Output: class label.

Success of transformers [Vaswani et al., 2017].

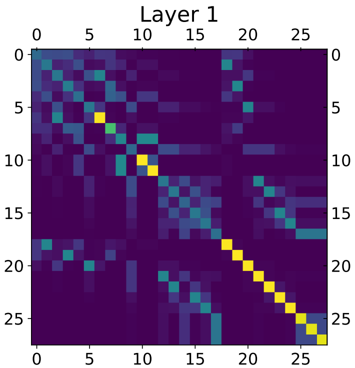
- Text [Devlin et al., 2019], Proteins [Rives et al., 2019], Images [Dosovitskiy et al., 2021].
- Rarely used for graphs.

A nice inductive bias for graphs?

- All input elements communicate...
- ...but model **blind to the input structure**.
- Hence, position encoding often required.

How to provide information on the structure of the graphs?

One example: GraphiT, encoding graph structure in transformers

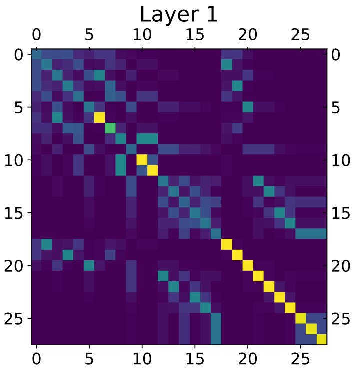


Diffusion kernel between the nodes of a Mutagenicity sample graph ($\beta = 1$).

[Mialon et al., 2021]

We propose two mechanisms:

One example: GraphiT, encoding graph structure in transformers



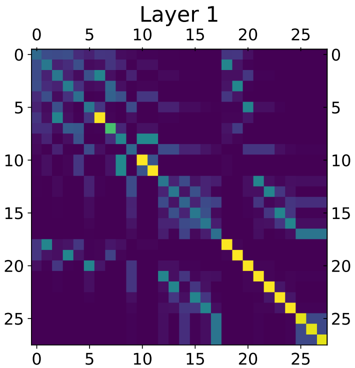
Diffusion kernel between the nodes of a Mutagenicity sample graph ($\beta = 1$).

[Mialon et al., 2021]

We propose two mechanisms:

- Modulating attention with **kernels on the graph** [Tsai et al., 2019, Kondor and Vert, 2004].

One example: GraphiT, encoding graph structure in transformers



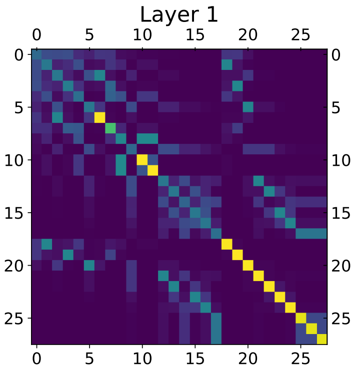
Diffusion kernel between the nodes of a Mutagenicity sample graph ($\beta = 1$).

[Mialon et al., 2021]

We propose two mechanisms:

- Modulating attention with **kernels on the graph** [Tsai et al., 2019, Kondor and Vert, 2004].
- Encoding **local neighborhood** of each node [Chen et al., 2020].

One example: GraphiT, encoding graph structure in transformers



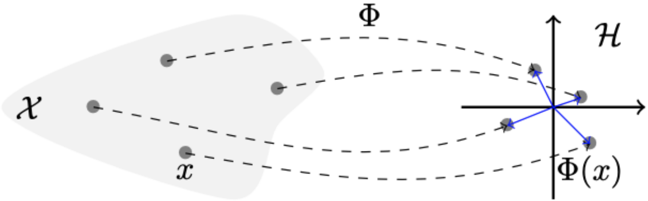
Diffusion kernel between the nodes of a Mutagenicity sample graph ($\beta = 1$).

[Mialon et al., 2021]

We propose two mechanisms:

- Modulating attention with **kernels on the graph** [Tsai et al., 2019, Kondor and Vert, 2004].
- Encoding **local neighborhood** of each node [Chen et al., 2020].
- Possible to encode edge features in both mechanisms.

Reminder: Kernel methods

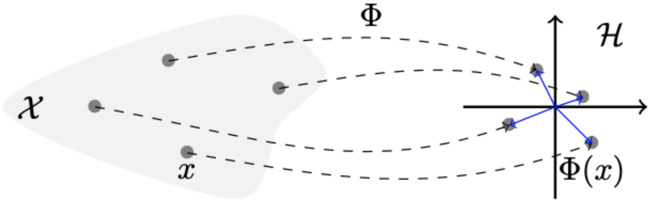


(From Bietti, 2019)

Learning with Kernel methods.

- Positive definite kernel K: defines a measure of similarity (prior?) between x and x'.

Reminder: Kernel methods

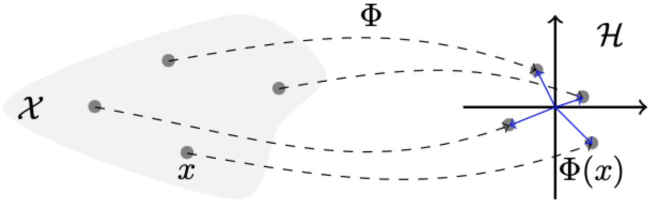


(From Bietti, 2019)

Learning with Kernel methods.

- Positive definite kernel K: defines a measure of similarity (prior?) between x and x'.
- Associated to rich embedding Φ via $K(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}}$.

Reminder: Kernel methods

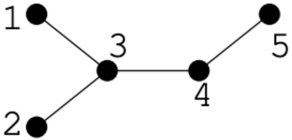


(From Bietti, 2019)

Learning with Kernel methods.

- Positive definite kernel K: defines a measure of similarity (prior?) between x and x'.
- Associated to rich embedding Φ via $K(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}}$.
- A surrogate for Φ can be learned with or without supervision [Williams and Seeger, 2001].

Reminder: Graph Laplacians



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

(From Vert, 2021)

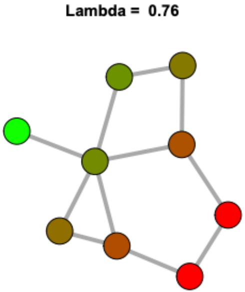
The Laplacian is a representation of the graph.

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

Reminder: Graph Laplacian

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.

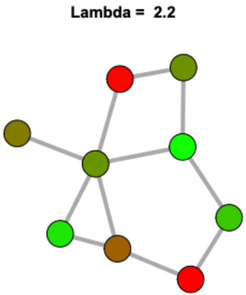


(From Vert, 2021)

Reminder: Graph Laplacian

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.

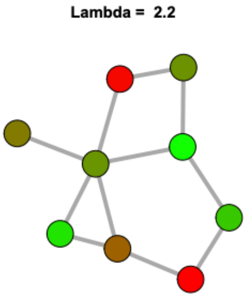


(From Vert, 2021)

Reminder: Graph Laplacian

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.



(From Vert, 2021)

"Discrete equivalent" to sine/cosine Fourier basis in $\mathbb{R}^n$.

Kernels on graphs

Laplacian based kernels [Smola and Kondor, 2003].

- Rich family of p.d. kernels on the graph by applying regularization function r to the spectrum of L

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

Kernels on graphs

Laplacian based kernels [Smola and Kondor, 2003].

- Rich family of p.d. kernels on the graph by applying regularization function r to the spectrum of L

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

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

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

A famous kernel on graphs: the diffusion kernel

Diffusion Kernel [Kondor and Vert, 2004].

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

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

- Physical interpretation: diffusion of a substance in the graph, controlled by β.

A famous kernel on graphs: the diffusion kernel

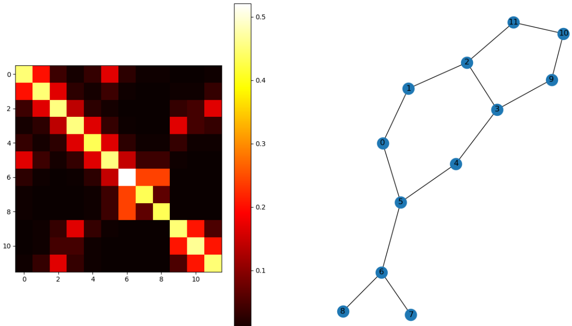
Diffusion Kernel [Kondor and Vert, 2004].

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

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

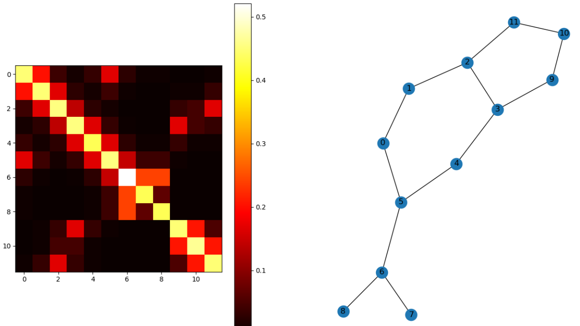
- Physical interpretation: diffusion of a substance in the graph, controlled by β.
- Discrete equivalent of the Gaussian kernel, a solution to the heat equation in the continuous setting.

Kernels on graphs provide smooth structural similarity between nodes



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

Kernels on graphs provide smooth structural similarity between nodes



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:

$$\text{Attention}(Q, V) = \text{normalize}\left(\exp\left(\frac{QQ^\top}{\sqrt{d_{\text{out}}}}\right)\right) V \in \mathbb{R}^{n \times d_{\text{out}}}. \tag{2}$$

Mechanism 1: node position encoding with kernels on graphs

Regular attention.

- Self-attention:

$$\text{Attention}(Q, V) = \text{normalize}\left(\exp\left(\frac{QQ^\top}{\sqrt{d_{\text{out}}}}\right)\right) V \in \mathbb{R}^{n \times d_{\text{out}}}. \tag{2}$$

- Feature map X gets:

$$X = X + \text{Attention}(Q, V). \tag{3}$$

Mechanism 1: node position encoding with kernels on graphs

Regular attention.

- Self-attention:

$$\text{Attention}(Q, V) = \text{normalize}\left(\exp\left(\frac{QQ^\top}{\sqrt{d_{\text{out}}}}\right)\right) V \in \mathbb{R}^{n \times d_{\text{out}}}. \tag{2}$$

- Feature map X gets:

$$X = X + \text{Attention}(Q, V). \tag{3}$$

Remark. Same matrices for Q and K [Tsai et al., 2019].

Mechanism 1: node position encoding with kernels on graphs

Modulated attention.

- Self-attention:

$$\text{PosAttention}(Q, V, K_r) = \text{normalize}\left(\exp\left(\frac{QQ^\top}{\sqrt{d_{\text{out}}}}\right) \odot K_r\right) V \in \mathbb{R}^{n \times d_{\text{out}}}, \tag{4}$$

with K_r a kernel on the graph.

Mechanism 1: node position encoding with kernels on graphs

Modulated attention.

- Self-attention:

$$\text{PosAttention}(Q, V, K_r) = \text{normalize}\left(\exp\left(\frac{QQ^\top}{\sqrt{d_{\text{out}}}}\right) \odot K_r\right) V \in \mathbb{R}^{n \times d_{\text{out}}}, \quad (4)$$

  with K_r a kernel on the graph.
- Feature map X gets:

$$X = X + D^{-\frac{1}{2}} \text{PosAttention}(Q, V, K_r), \quad (5)$$

  with D the matrix of node degrees.

Mechanism 2: leveraging substructures via path embedding

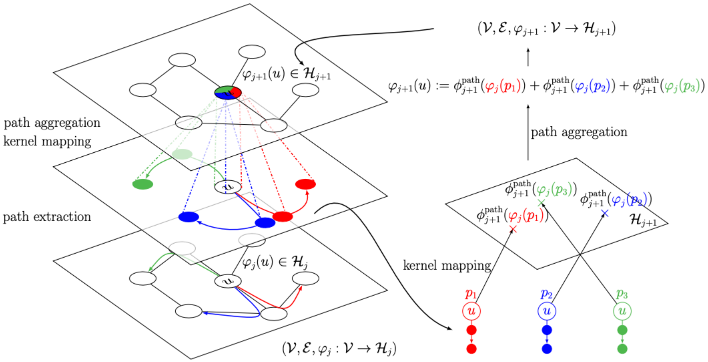
- Substructures: local positional information *and* content, *e.g* paths [Borgwardt et al., 2020].

Mechanism 2: leveraging substructures via path embedding

- Substructures: local positional information *and* content, *e.g* paths [Borgwardt et al., 2020].
- Augmenting node features u using kernel neighborhood encoding [Chen et al., 2020].

Mechanism 2: leveraging substructures via path embedding

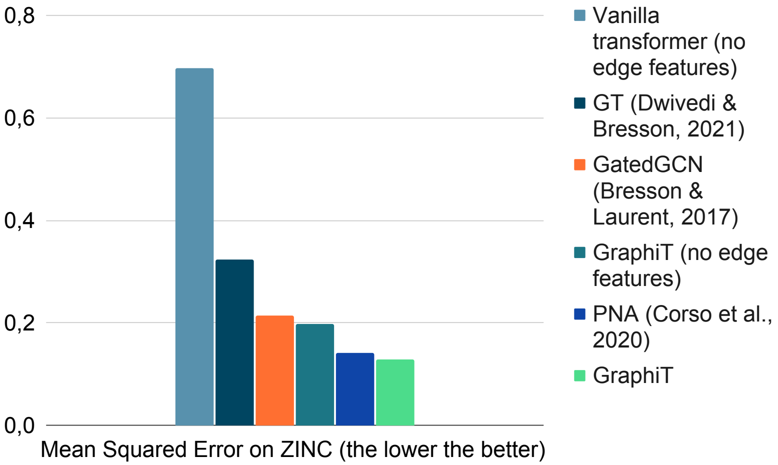
- Substructures: local positional information *and* content, *e.g* paths [Borgwardt et al., 2020].
- Augmenting node features u using kernel neighborhood encoding [Chen et al., 2020].
- Kernel encoding learned with or without supervision.



(from Chen et al.)

GraphiT is able to outperform popular GNNs

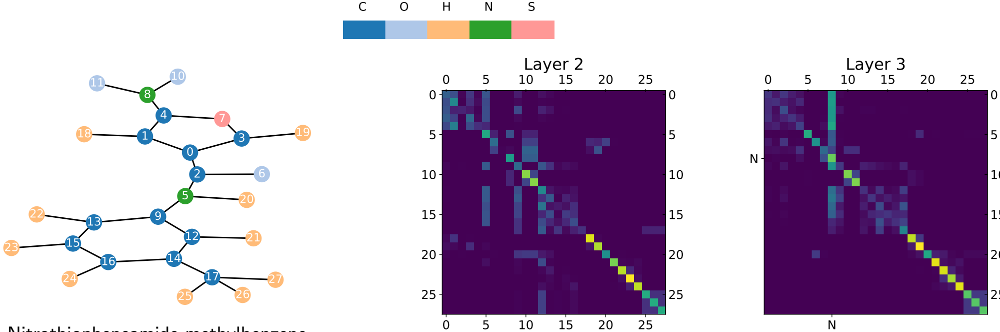
ZINC (12k graphs, regression): Predicting the constrained differential solubility of molecules.



[Mialon et al., 2021]

GraphiT captures meaningful interactions

Mutagenicity: 4k samples (binary classification).



Nitrothiopheneamide-methylbenzene

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.

[Mialon et al., 2021]

Related work

There are many ways to incorporate graph structure into the transformer.

- Position encoding with eigenvectors of L [Dwivedi and Bresson, 2021].
- Fully learned position encoding [Ying et al., 2021].
- Message passing with position encoding [Dwivedi et al., 2021].
- And many others! [Kreuzer et al., 2021, Choromanski et al., 2021] . . .

Scaling to larger datasets

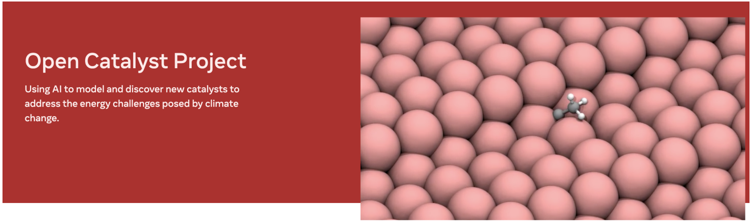
PCQM4M-LSC 2021 [Hu et al., 2020]:

- Goal: chemistry knowledge gain by pre-training.
- Task: predict an energy gap of molecules from DFT simulations.
- 3.8M graphs.
- Winner: (Ensemble of) Graphormer [Ying et al., 2021].
- 47M parameters per model.
- ? on NVIDIA V100 GPUs on Microsoft Azure Cloud.

Scaling to larger datasets

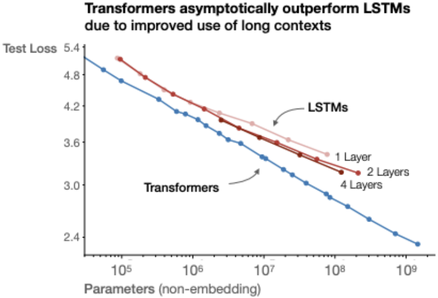
Open Catalyst 2020 [Zitnick et al., 2020]:

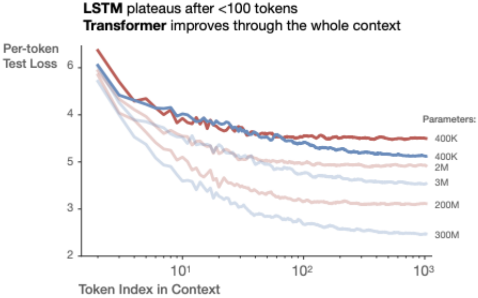
- Goal: accelerating catalyst discovery for systems such as renewable fertilizer, energy storage.
- Task: predicting an adsorbate-catalyst energy from simulations.
- 140M structure-energy estimation.
- Winner: Graphormer [Ying et al., 2021].
- 150M parameters?
- 1.5 days on 8 Nvidia A100.



Conclusion

- Inductive bias of transformers is valid for graphs.
- Promising interpretation capabilities.
- Scaling laws with respect to graphs?





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*.

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*.

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

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. (2021). Rethinking attention with performers. In *International Conference on Learning Representations*.

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 II

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)*.

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

Dwivedi, V. P., Luu, A. T., Laurent, T., Bengio, Y., and Bresson, X. (2021). Graph neural networks with learnable structural and positional representations.

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.

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.

References III

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

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.

Li, Q., Han, Z., and Wu, X.-M. (2018). Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI conference on artificial intelligence*.

Mialon, G., Chen, D., Selosse, M., and Mairal, J. (2021). Graphit: Encoding graph structure in transformers.

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*.

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.

References IV

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)*.

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)*.

Williams, C. K. and Seeger, M. (2001). Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Xie, T., Bapst, V., Gaunt, A. L., Obika, A., Back, T., Hassabis, D., Kohli, P., and Kirkpatrick, J. (2021). Atomistic graph networks for experimental materials property prediction.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T.-Y. (2021). Do transformers really perform badly for graph representation? In *Thirty-Fifth Conference on Neural Information Processing Systems*.

References V

Zitnick, C. L., Chanussot, L., Das, A., Goyal, S., Heras-Domingo, J., Ho, C., Hu, W., Lavril, T., Palizhati, A., Riviere, M., Shuaibi, M., Sriram, A., Tran, K., Wood, B., Yoon, J., Parikh, D., and Ulissi, Z. (2020). An introduction to electrocatalyst design using machine learning for renewable energy storage.