Designing Transformers with Kernel Methods

Grégoire Mialon

DeepMind

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 machine learning (M.S. MVA), ENS Paris-Saclay.
- 2018-today: PhD candidate, advised by Julien Mairal and Alexandre d'Aspremont.

An Offshore Floating Nuclear Plant.

What I have been doing in the past 3.5 years

Kernel methods and deep learning in constrained data regimes (100 to 10k samples).

- 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).

What I have been doing in the past 3.5 years

Kernel methods and deep learning in constrained data regimes (100 to 10k samples).

- 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 constrained data regime (100 to 10k samples). Application to scientific data.

- 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 data is scarce

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

Motivation: designing strong models even when data is scarce

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

- A path towards better models?

Motivation: designing strong models even when 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:

Motivation: designing strong models even when 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:
  - Rare events: less than 30k people per rare disease in France (2021).

Motivation: designing strong models even when 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:
  - Rare events: less than 30k people per rare disease in France (2021).
  - Expensive or complex data collection for fundamental science/econometrics.

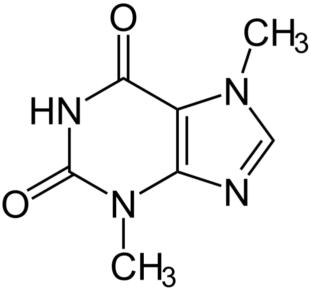
How to design strong models in constrained data regimes?

Encode inductive bias within trainable architectures with kernel methods.

Outline: Encoding inductive bias within trainable architectures with kernel methods

1. **Encoding Graph Structure in Transformers with Kernels on Graphs**
2. Embedding Sets of Features with Optimal Transport Kernels
3. Conclusion and perspectives

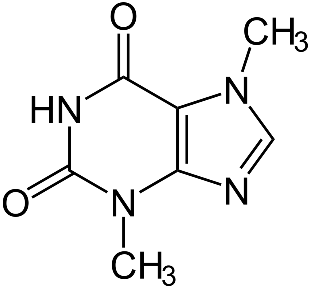
Graph data are an important research topic



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

Graph data are very valuable...

Graph data are an important research topic

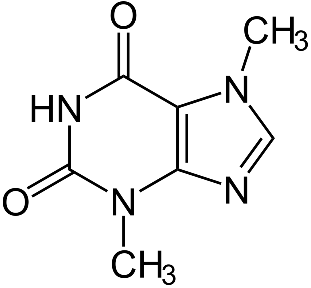


A molecule of theobromin, or why chocolate makes us feel good.

Graph data are very valuable...

- Molecules in chemoinformatics.

Graph data are an important research topic

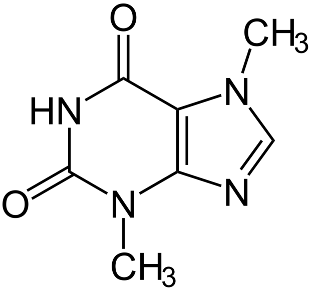


A molecule of theobromin, or why chocolate makes us feel good.

Graph data are very valuable...

- Molecules in chemoinformatics.
- Proteins in computational biology.

Graph data are an important research topic

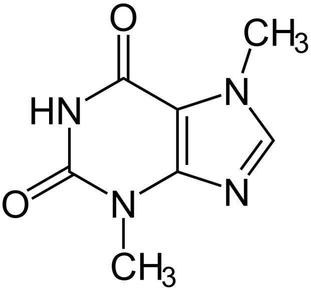


A molecule of theobromin, or why chocolate makes us feel good.

Graph data are very valuable...

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

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.
- Meshes in computer vision and computer graphics, etc.

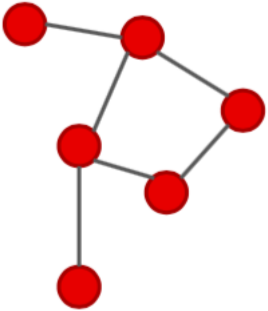
...but delicate to exploit.

- 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.

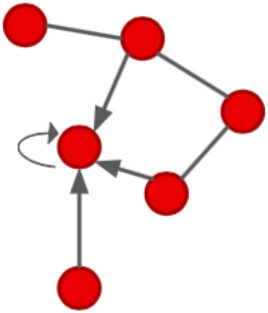


GNN, layer k

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.

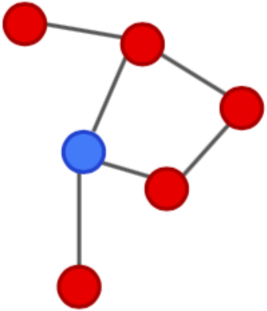


GNN, layer k

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.

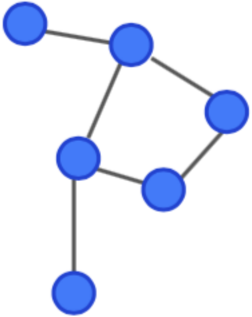


GNN, layer k+1 (for one node)

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.



GNN, layer k+1

Learning with Graph Neural Networks

Graph Neural Networks (GNNs).

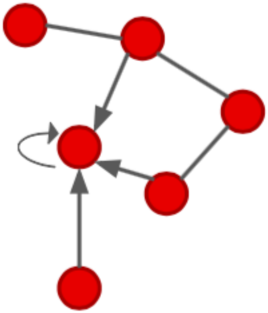
- Many strategies to aggregate features of neighboring nodes [Duvenaud et al., 2015, Bronstein et al., 2017, Veličković et al., 2018].

Learning with Graph Neural Networks

Graph Neural Networks (GNNs).

- Many strategies to aggregate features of neighboring nodes [Duvenaud et al., 2015, Bronstein et al., 2017, Veličković et al., 2018].
- Applications to molecules [Duvenaud et al., 2015], physical systems [Battaglia et al., 2016], materials [Xie et al., 2021], etc.

GNNs may struggle with long-range interactions

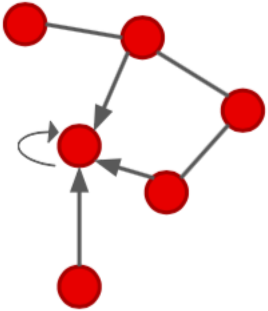


Only neighboring nodes communicate.

In GNNs, messages flow between neighbors only.

- Exploits the structure of the graph.

GNNs may struggle with long-range interactions

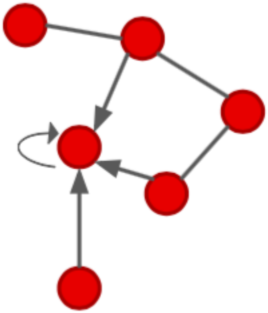


Only neighboring nodes communicate.

In GNNs, messages flow between neighbors only.

- Exploits the structure of the graph.
- But n layers for n-hop neighbors to interact.

GNNs may struggle with long-range interactions

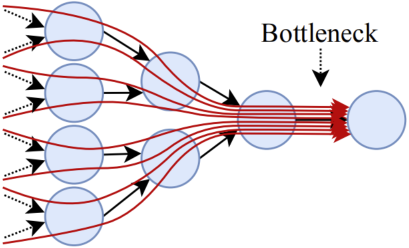


Only neighboring nodes communicate.

In GNNs, messages flow between neighbors only.

- Exploits the structure of the graph.
- But n layers for n-hop neighbors to interact.
- Oversmoothing [Li et al., 2018].

GNNs may struggle with long-range interactions

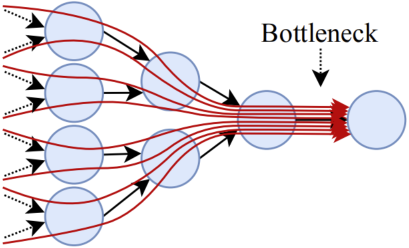


An illustration of oversquashing
(From Alon and Yahav).

In GNNs, messages flow between neighbors only.

- Exploits the structure of the graph.
- But, n layers for n-hop neighbors to interact.
- Oversmoothing [Li et al., 2018].
- Bottleneck effect [Alon and Yahav, 2021].

GNNs may struggle with long-range interactions

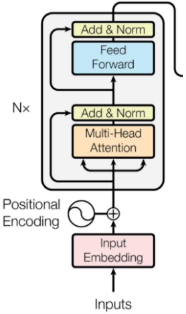


An illustration of oversquashing (From Alon and Yahav).

In GNNs, messages flow between neighbors only.

- Exploits the structure of the graph.
- But, n layers for n-hop neighbors to interact.
- Oversmoothing [Li et al., 2018].
- Bottleneck effect [Alon and Yahav, 2021].
- Attempts at solving this issue [Godwin et al., 2021].

Transformers

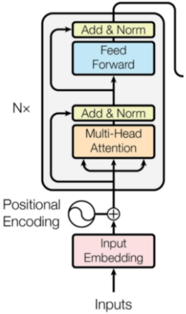


Transformer encoder (from Vaswani et al.)

Transformers perform global aggregation!

- Initially introduced in natural language processing [Vaswani et al., 2017, Devlin et al., 2019].

Transformers

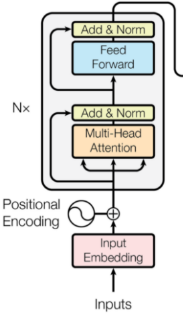


Transformer encoder
(from Vaswani et al.)

Transformers perform global aggregation!

- 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



Transformer encoder (from Vaswani et al.)

Transformers perform global aggregation!

- 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].
- Question the paradigm "one data modality, one preferred architecture".

Transformers

Transformer encoder.

- Input: set of n elements X in $\mathbb{R}^{n \times d_{\text{in}}}$. Output: another set in $\mathbb{R}^{n \times d_{\text{out}}}$.

Transformers

Transformer encoder.

- Input: set of n elements X in $\mathbb{R}^{n \times d_{\text{in}}}$. Output: another set in $\mathbb{R}^{n \times d_{\text{out}}}$.
- Feature map X updated via:

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

Transformers

Transformer encoder.

- Input: set of n elements X in $\mathbb{R}^{n \times d_{\text{in}}}$. Output: another set in $\mathbb{R}^{n \times d_{\text{out}}}$.
- Feature map X updated via:

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

- 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.

Transformers

Transformer encoder.

- Input: set of n elements X in $\mathbb{R}^{n \times d_{\text{in}}}$. Output: another set in $\mathbb{R}^{n \times d_{\text{out}}}$.
- Feature map X updated via:

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

- Self-attention mechanism:

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

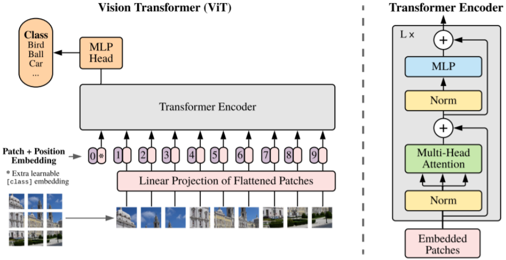
  with $Q^\top = W_Q X^\top$ and $K^\top = W_K X^\top$ resp. query and key matrices, $V^\top = W_V X^\top$ the value matrix, and W_Q, W_K, W_V in $\mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ learned projection matrices.
- LayerNorm then “element-wise” feed-forward.

Transformers

Transformer encoder.

- Input: set of n elements X in $\mathbb{R}^{n \times d_{\text{in}}}$. Output: another set in $\mathbb{R}^{n \times d_{\text{out}}}$.
- Feature map X updated via:
$$X = X + \text{Attention}(Q, K, V).$$
- 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.
- LayerNorm then “element-wise” feed-forward.
- Repeat.

Transformers for graphs require position encoding

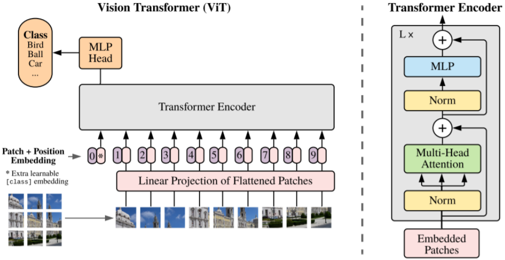


Vision transformer (from [Dosovitskiy et al., 2021])

A nice inductive bias for graphs?

- All input elements communicate...

Transformers for graphs require position encoding

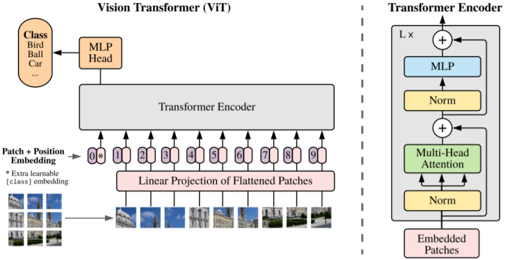


Vision transformer (from [Dosovitskiy et al., 2021])

A nice inductive bias for graphs?

- All input elements communicate...
- ... but encoder output is permutation invariant.

Transformers for graphs require position encoding

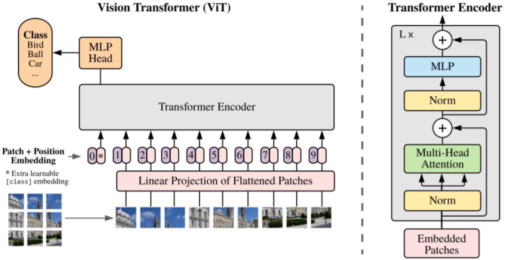


Vision transformer (from [Dosovitskiy et al., 2021])

A nice inductive bias for graphs?

- All input elements communicate...
- ... but encoder output is permutation invariant.
- Hence, position encoding often required.

Transformers for graphs require position encoding

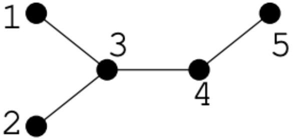


Vision transformer (from [Dosovitskiy et al., 2021])

A nice inductive bias for graphs?

- All input elements communicate...
- ... but encoder output is permutation invariant.
- Hence, position encoding often required.
- Not trivial for graphs!

Previous attempts at using transformers with graphs



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

(From Vert, 2021)

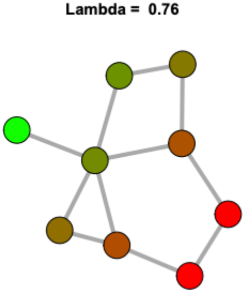
Dwivedi & Bresson, 2021: absolute PE using Laplacian eigenvectors.

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

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.

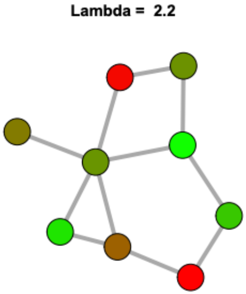


(From Vert, 2021)

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.

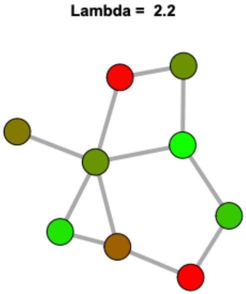


(From Vert, 2021)

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.

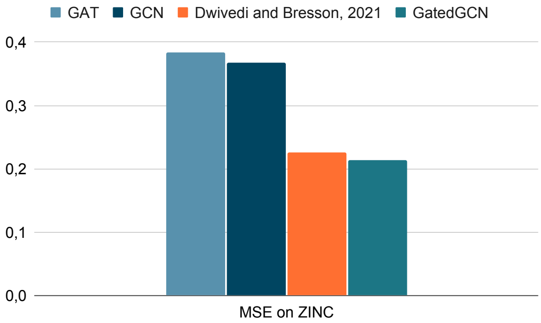


(From Vert, 2021)

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

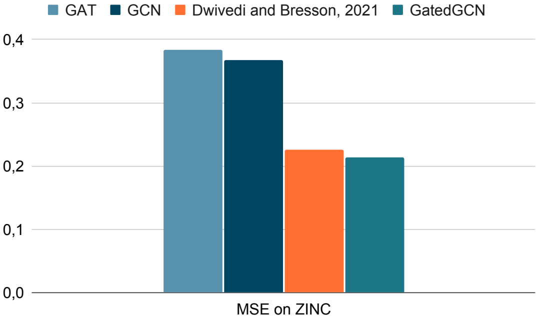
Promising results but...

ZINC: 12k graphs (regression).



Promising results but...

ZINC: 12k graphs (regression).

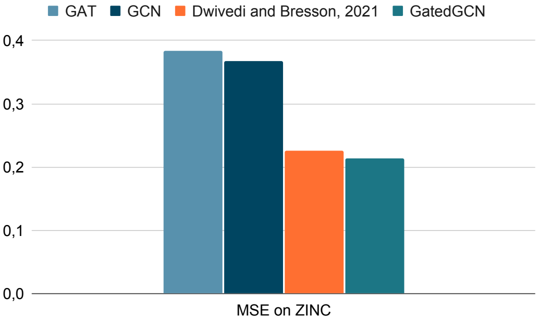


Problems with Laplacian absolute PE.

- Flipping sign at training.

Promising results but...

ZINC: 12k graphs (regression).

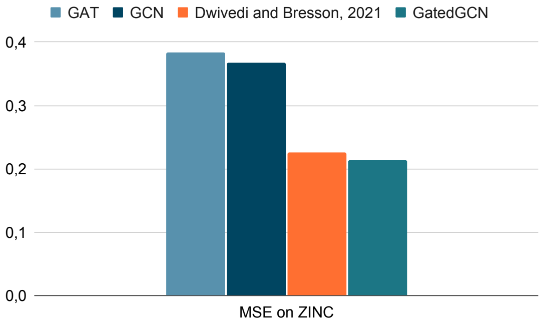


Problems with Laplacian absolute PE.

- Flipping sign at training.
- Do these vectors transfer between different graphs?

Promising results but...

ZINC: 12k graphs (regression).



Problems with Laplacian absolute PE.

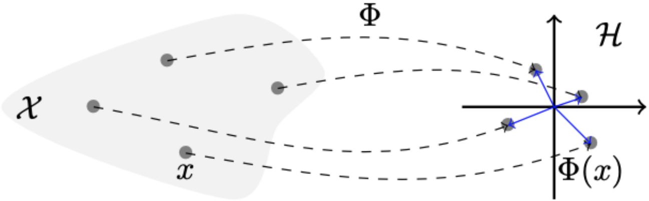
- Flipping sign at training.
- Do these vectors transfer between different graphs?

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

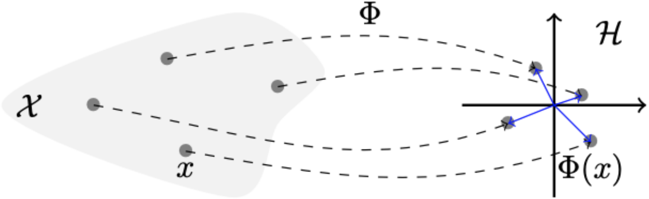


(From Bietti, 2019)

Learning with Kernel methods.

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

Reminder: Kernel methods

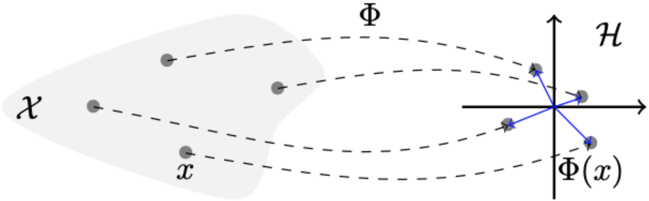


(From Bietti, 2019)

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].

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

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

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

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

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

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

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

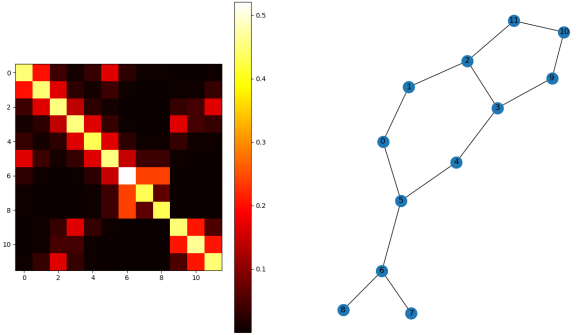
Diffusion Kernel [Kondor and Vert, 2004].

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

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

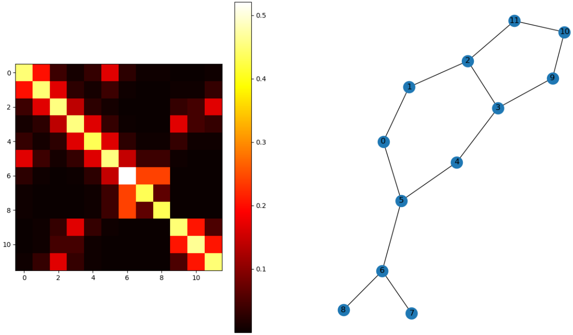
- Physical interpretation: diffusion of a substance in the graph, controlled by β.
- Discrete equivalent of the Gaussian kernel, a solution to the heat equation in the continuous setting.

Kernels on graphs reflect structural similarity between nodes



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

Kernels on graphs reflect 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:

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

Mechanism 1: node position encoding with kernels on graphs

Regular attention.

- Self-attention:

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

- Feature map X gets:

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

Mechanism 1: node position encoding with kernels on graphs

Regular attention.

- Self-attention:

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

- Feature map X gets:

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

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

Mechanism 1: node position encoding with kernels on graphs

Modulated attention.

- Self-attention:

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

with K_r a kernel on the graph.

Mechanism 1: node position encoding with kernels on graphs

Modulated attention.

- Self-attention:

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

  with K_r a kernel on the graph.

- Feature map X gets:

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

  with D the matrix of node degrees.

Mechanism 2: leveraging substructures via path embedding

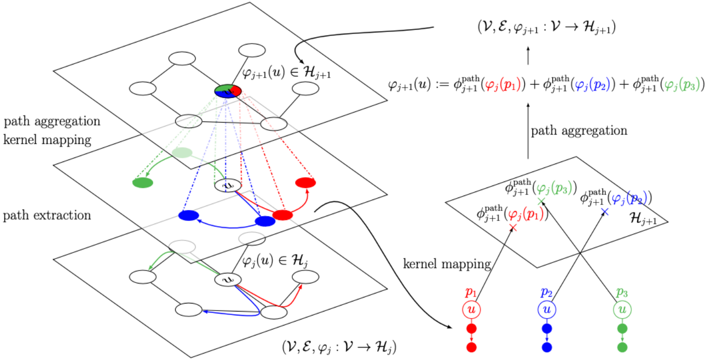
- Substructures: local positional information *and* content, *e.g* paths [Borgwardt et al., 2020].

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

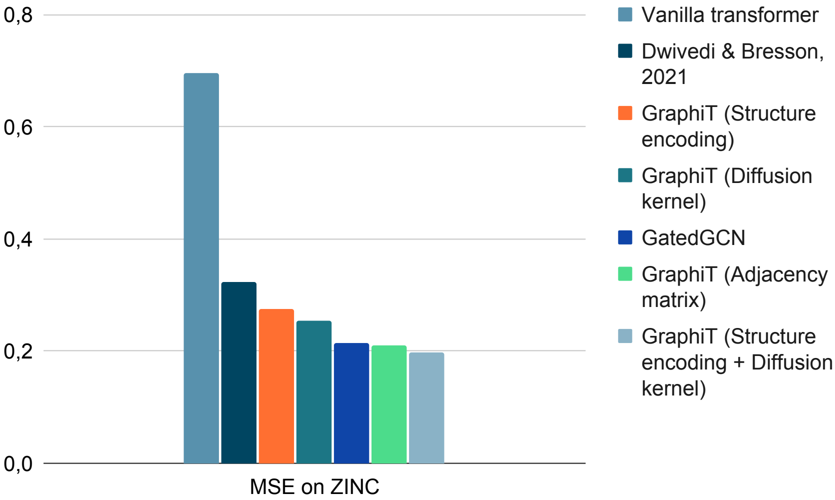
- Substructures: local positional information *and* content, *e.g* paths [Borgwardt et al., 2020].
- Augmenting node features u using kernel neighborhood encoding [Chen et al., 2020].
- Kernel encoding learned with or without supervision.



$(\mathcal{V}, \mathcal{E}, \varphi_{j+1} : \mathcal{V} \to \mathcal{H}_{j+1})$
$\varphi_{j+1}(u) \in \mathcal{H}_{j+1}$
$\varphi_{j+1}(u) := \phi^{\text{path}}_{j+1}(\varphi_j(p_1)) + \phi^{\text{path}}_{j+1}(\varphi_j(p_2)) + \phi^{\text{path}}_{j+1}(\varphi_j(p_3))$
path aggregation
kernel mapping
path aggregation
$\phi^{\text{path}}_{j+1}(\varphi_j(p_3))$
$\phi^{\text{path}}_{j+1}(\varphi_j(p_2))$
$\phi^{\text{path}}_{j+1}(\varphi_j(p_1))$
$\mathcal{H}_{j+1}$
path extraction
u
kernel mapping
$\varphi_j(u) \in \mathcal{H}_j$
u
p_1
p_2
p_3
u
u
u
$(\mathcal{V}, \mathcal{E}, \varphi_j : \mathcal{V} \to \mathcal{H}_j)$

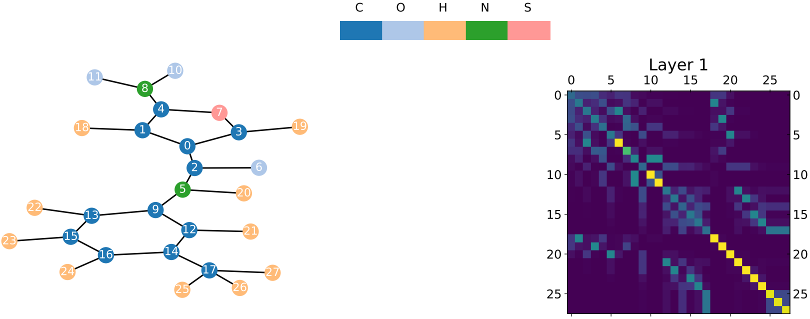
GraphiT is able to outperform popular GNNs

ZINC: 12k graphs (regression).



GraphiT captures meaningful interactions

Mutagenicity: 4k graphs (binary classification).

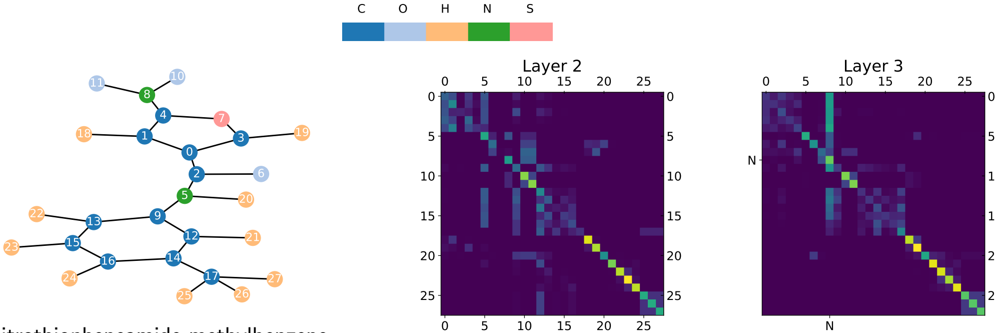


Nitrothiopheneamide-methylbenzene

Figure 1: *Left*: A molecule from the Mutagenicity data set [Kersting et al., 2016]. *Right*: approximate diffusion kernel for the molecular graph.

GraphiT captures meaningful interactions

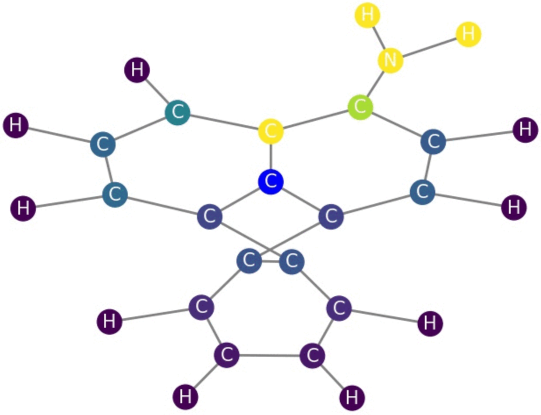
Mutagenicity: 4k samples (binary classification).



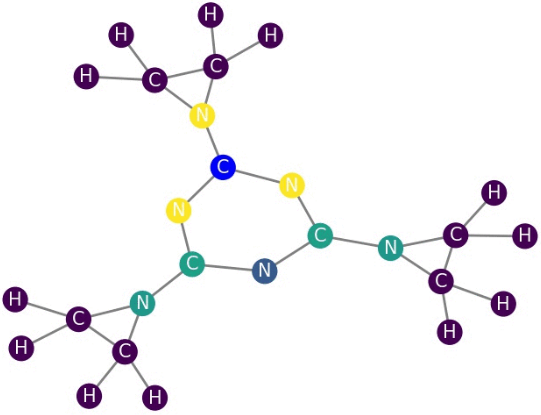
Nitrothiopheneamide-methylbenzene

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.

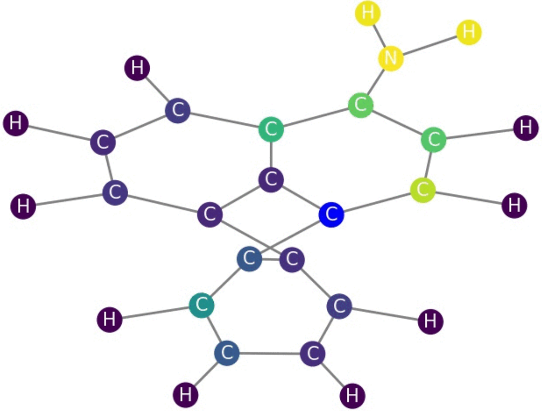
Attention from C atom



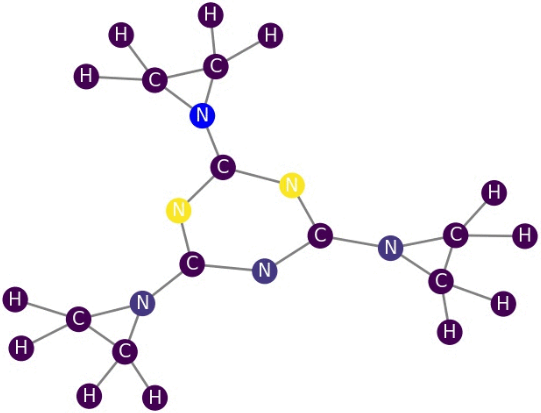
Attention from C atom



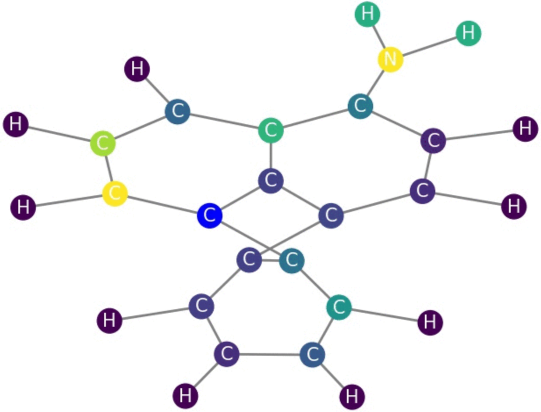
Attention from C atom



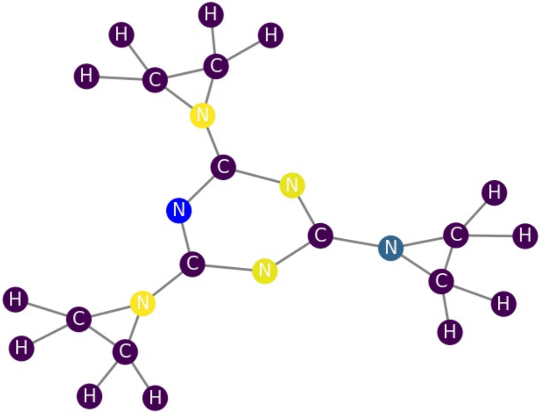
Attention from N atom

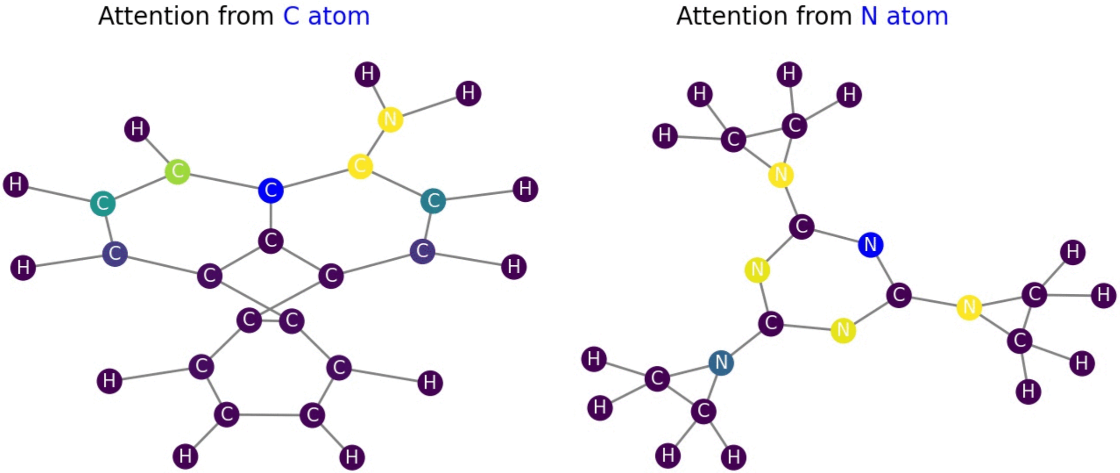


Attention from C atom



Attention from N atom

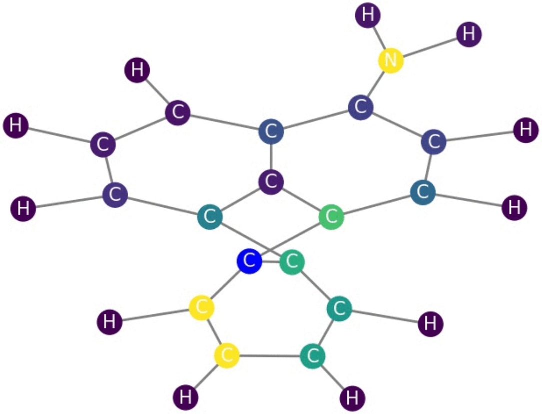




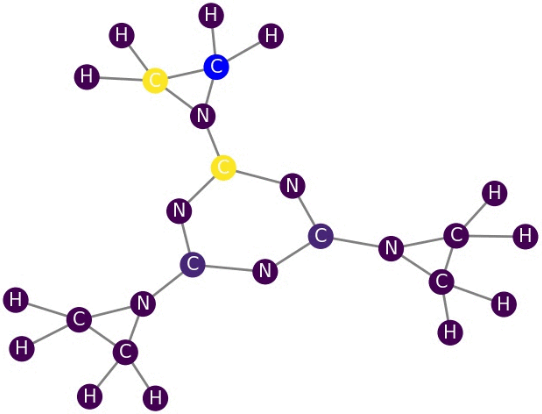
Attention from C atom

Attention from N atom

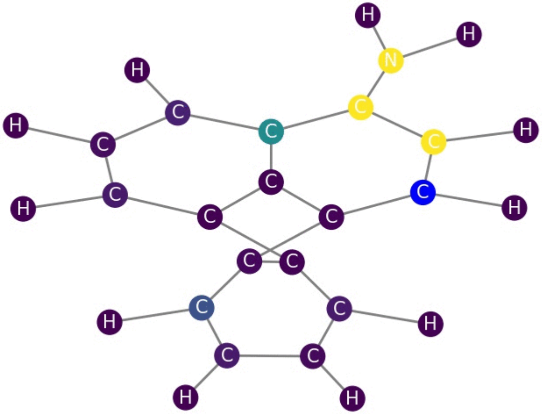
Attention from C atom



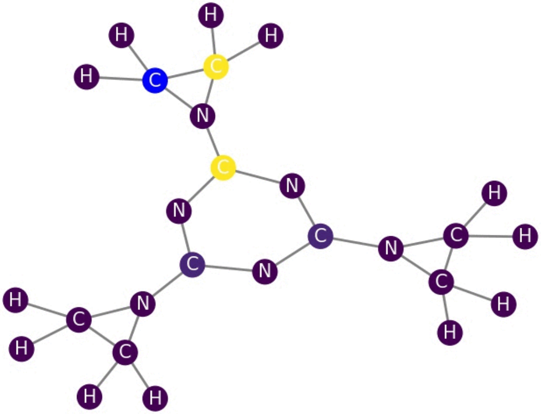
Attention from C atom



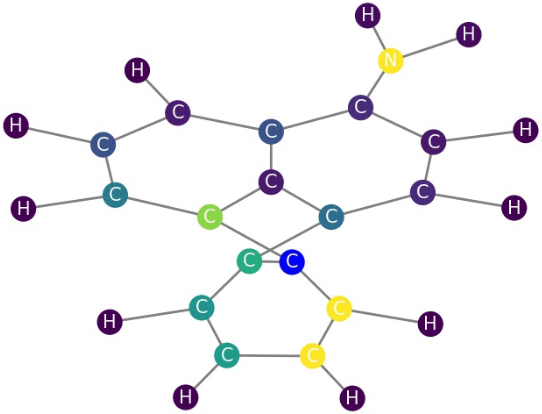
Attention from C atom



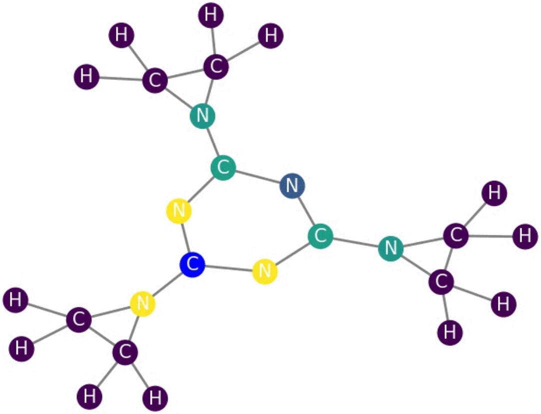
Attention from C atom



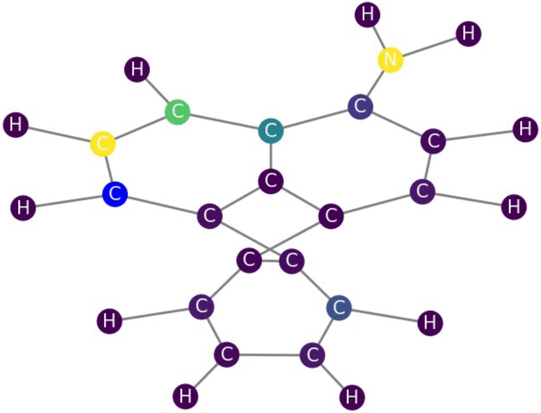
Attention from C atom



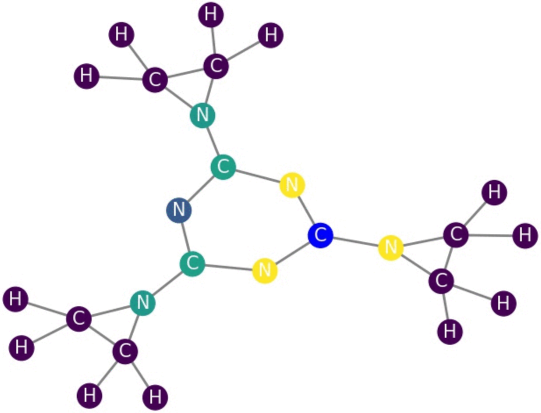
Attention from C atom



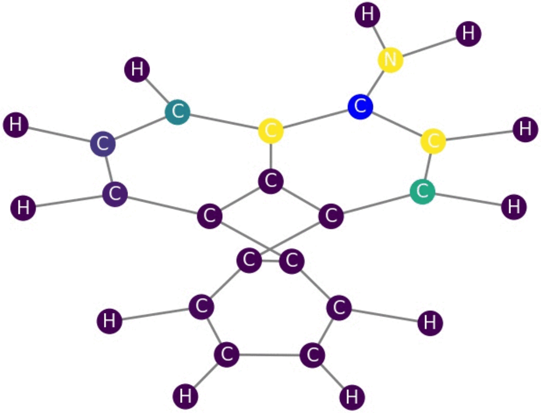
Attention from C atom



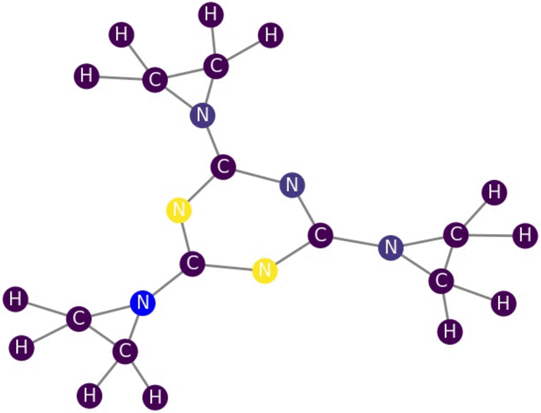
Attention from C atom



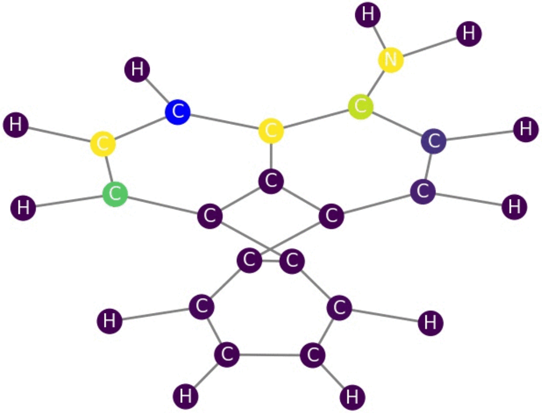
Attention from C atom



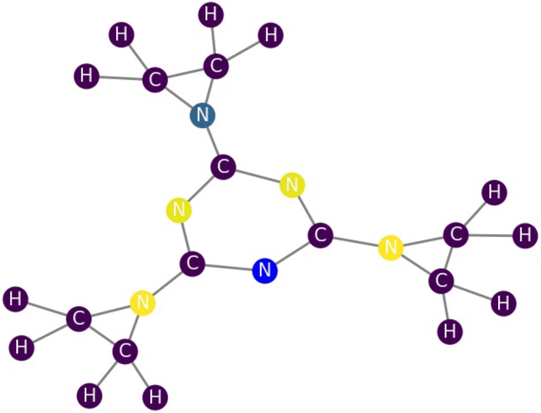
Attention from N atom



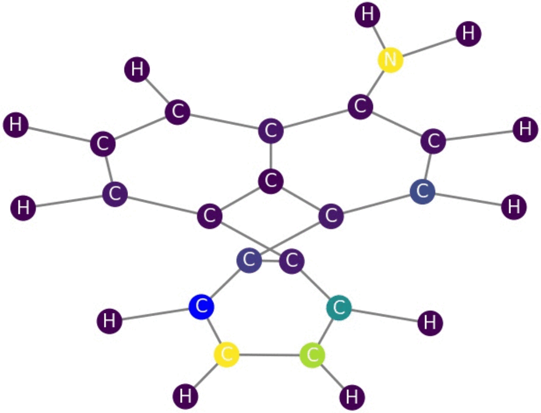
Attention from C atom



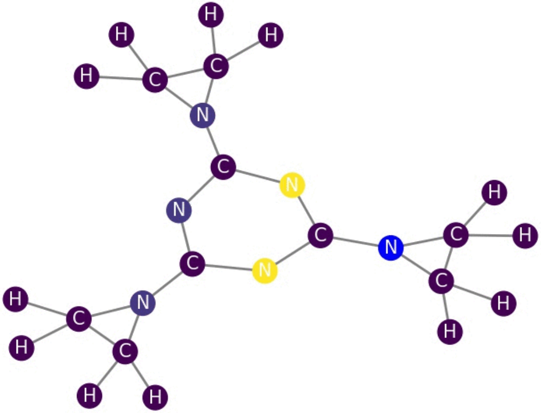
Attention from N atom



Attention from C atom



Attention from N atom



Limitations and perspectives of GraphiT

Current limitations...

- Evaluation on large scale datasets:
  OGB [Hu et al., 2020]?

Limitations and perspectives of GraphiT

Current limitations...

- Evaluation on large scale datasets: OGB [Hu et al., 2020]?
- Large graphs? Recent line of work on efficient transformers [Tay et al., 2020].

Limitations and perspectives of GraphiT

Current limitations...

- Evaluation on large scale datasets: OGB [Hu et al., 2020]?
- Large graphs? Recent line of work on efficient transformers [Tay et al., 2020].

...and exciting questions!

- Active domain of research [Ying et al., 2021, Kreuzer et al., 2021].

Limitations and perspectives of GraphiT

Current limitations...

- Evaluation on large scale datasets: OGB [Hu et al., 2020]?
- Large graphs? Recent line of work on efficient transformers [Tay et al., 2020].

...and exciting questions!

- Active domain of research [Ying et al., 2021, Kreuzer et al., 2021].
- [Dwivedi et al., 2021] improved PE for GraphiT.

Limitations and perspectives of GraphiT

Current limitations...

- Evaluation on large scale datasets: OGB [Hu et al., 2020]?
- Large graphs? Recent line of work on efficient transformers [Tay et al., 2020].

...and exciting questions!

- Active domain of research [Ying et al., 2021, Kreuzer et al., 2021].
- [Dwivedi et al., 2021] improved PE for GraphiT.
- Visualization for real-life applications?

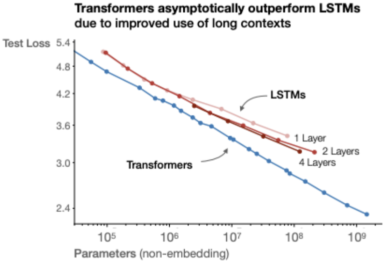
Limitations and perspectives of GraphiT

Current limitations...

- Evaluation on large scale datasets: OGB [Hu et al., 2020]?
- Large graphs? Recent line of work on efficient transformers [Tay et al., 2020].

...and exciting questions!

- Active domain of research [Ying et al., 2021, Kreuzer et al., 2021].
- [Dwivedi et al., 2021] improved PE for GraphiT.
- Visualization for real-life applications?
- Pre-trained models? Self-supervised learning for graphs? [Thakoor et al., 2021]



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

Kernel smoothing interpretation

Self-attention as a kernel smoothing [Tsai et al., 2019].

- 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.

Kernel smoothing interpretation

Self-attention as a kernel smoothing [Tsai et al., 2019].

- We can rewrite self-attention:

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

  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.

Kernel smoothing interpretation

Self-attention as a kernel smoothing [Tsai et al., 2019].

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

with $k(X_i, X_j) = exp\left(\frac{Q_i K_j^\top}{\sqrt{d_{out}}}\right)$.

- $k(X_i, X_j)$ replaced by $k(X_i, X_j) \times K_r(i, j)$. k: nodes contents similarity, K_r: nodes structural similarity.

Kernel smoothing interpretation

Self-attention as a kernel smoothing [Tsai et al., 2019].

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

with $k(X_i, X_j) = exp\left(\frac{Q_i K_j^\top}{\sqrt{d_{out}}}\right)$.

- $k(X_i, X_j)$ replaced by $k(X_i, X_j) \times K_r(i, j)$. k: nodes contents similarity, K_r: nodes structural similarity.
- Related to relative positional encoding [Shaw et al., 2018].

Kernel smoothing interpretation

Self-attention as a kernel smoothing [Tsai et al., 2019].

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

with $k(X_i, X_j) = exp\left(\frac{Q_i K_j^\top}{\sqrt{d_{out}}}\right)$.

- $k(X_i, X_j)$ replaced by $k(X_i, X_j) \times K_r(i, j)$. k: nodes contents similarity, K_r: nodes structural similarity.
- Related to relative positional encoding [Shaw et al., 2018].

What if we pick a different similarity measure k?

Outline

1. Encoding Graph Structure in Transformers with Kernels on Graphs
2. **Embedding Sets of Features with Optimal Transport Kernels**
3. Conclusion and perspectives

Sets are another important data structure

Notre-Dame de Paris, LIDAR view (Andrew Tallon)

Sets can be found in various domains.

- 3D shape recognition (point clouds).

Sets are another important data structure

Notre-Dame de Paris, LIDAR view (Andrew Tallon)

Sets can be found in various domains.

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

Sets are another important data structure

Notre-Dame de Paris, LIDAR view (Andrew Tallon)

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.

Sets are another important data structure

Notre-Dame de Paris, LIDAR view (Andrew Tallon)

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.

Sets are another important data structure

Notre-Dame de Paris, LIDAR view (Andrew Tallon)

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

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.

Let's focus on biological sequences

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.

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

- Sequences may be long.

Let’s focus on biological sequences

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.

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

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

Our sequences require specific embedding

Existing methods do not yield satisfactory results for our data.

- Kernel methods for sets [Lyu, 2004]: not expressive enough.

Our sequences require specific embedding

Existing methods do not yield satisfactory results for our data.

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

Our sequences require specific embedding

Existing methods do not yield satisfactory results for our data.

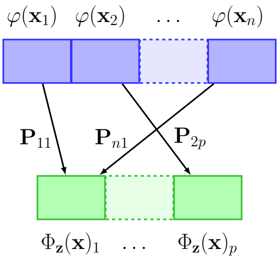
- Kernel methods for sets [Lyu, 2004]: not expressive enough.
- NN architectures for sets [Lee et al., 2019, Skianis et al., 2020]: empirically mixed results.

How to represent sets with low data and memory requirements?

Our contribution: OTKE, 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`

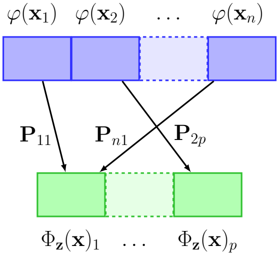
OTKE: a data-efficient embedding for sets



Global, similarity-based pooling.

- Input: set or sequence $\mathsf{X} \in \mathbb{R}^{n \times d_{\text{in}}}$.

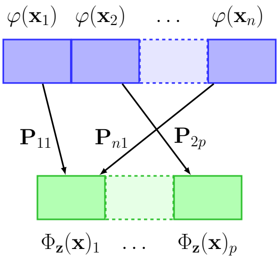
OTKE: a data-efficient embedding for sets



Global, similarity-based pooling.

- Input: set or sequence $X \in \mathbb{R}^{n \times d_{\text{in}}}$.
- Element-wise, non-linear embedding φ.

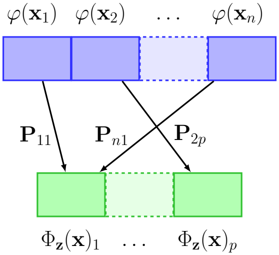
OTKE: a data-efficient embedding for sets



Global, similarity-based pooling.

- Input: set or sequence $\mathsf{X} \in \mathbb{R}^{n \times d_{\text{in}}}$.
- Element-wise, non-linear embedding φ.
- Pool elements $\varphi(\mathsf{x}_i)$ in p bins via weighted sums.

OTKE: a data-efficient embedding for sets

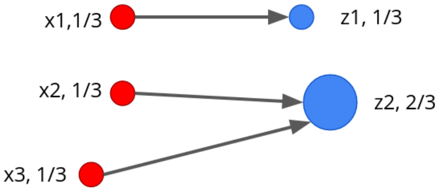


Global, similarity-based pooling.

- Input: set or sequence $\mathrm{X} \in \mathbb{R}^{n \times d_{\text{in}}}$.
- Element-wise, non-linear embedding φ.
- Pool elements $\varphi(\mathrm{x}_i)$ in p bins via weighted sums.
- To each bin corresponds a prototype (parameter) $z_j \in \mathbb{R}^{d_{\text{out}}}$, $j = 1 \ldots p$.

Pooling weight P_{ij} reflects similarity between x_i and z_j.

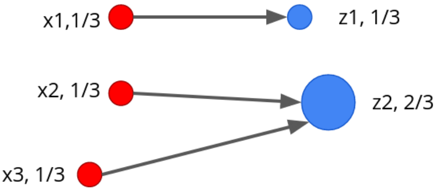
Our notion of similarity: optimal transport



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

What is optimal transport?

Our notion of similarity: optimal transport

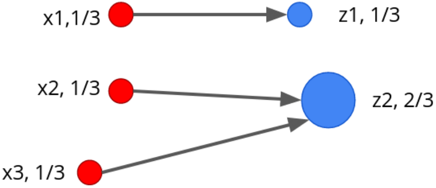


$$P=\begin{pmatrix}1/3 & 0 & 0\\ 0 & 1/3 & 1/3\end{pmatrix}$$

What is optimal transport?

- “Most efficient way of transporting a mass distribution to another” [Peyré and Cuturi, 2019].

Our notion of similarity: optimal transport

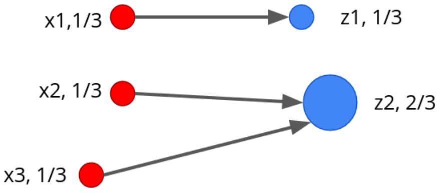


$$P=\begin{pmatrix} 1/3 & 0 & 0 \\ 0 & 1/3 & 1/3 \end{pmatrix}$$

What is optimal transport?

- "Most efficient way of transporting a mass distribution to another" [Peyré and Cuturi, 2019].
- Finding the transport plan minimizing a transportation cost.

Our notion of similarity: optimal transport



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

What is optimal transport?

- "Most efficient way of transporting a mass distribution to another" [Peyré and Cuturi, 2019].
- Finding the transport plan minimizing a transportation cost.
- GPU-friendly solvers [Sinkhorn and Knopp, 1967, Cuturi and Doucet, 2013].

Our notion of similarity: optimal transport

Dot-product vs OT.

- OT empirically better.

Our notion of similarity: optimal transport

Dot-product vs OT.

- OT empirically better.
- Softmax not needed anymore.

Our notion of similarity: optimal transport

Dot-product vs OT.

- OT empirically better.
- Softmax not needed anymore.

Two interpretations.

- Embeds the sets in a space where ℓ_2 distance approximates the 2-Wasserstein distance [Wang et al., 2013].

Our notion of similarity: optimal transport

Dot-product vs OT.

- OT empirically better.
- Softmax not needed anymore.

Two interpretations.

- Embeds the sets in a space where ℓ_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 learned with or without supervision.

Reasonable memory/data requirements

Data efficient.

- Z 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 $\mathrm{P}(X, Z)_{ij}$.

Reasonable memory/data requirements

Data efficient.

- Z 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 $\mathrm{P}(X, Z)_{ij}$.
- Linear in the number of input elements.

Reasonable memory/data requirements

Data efficient.

- Z learned with or without supervision.

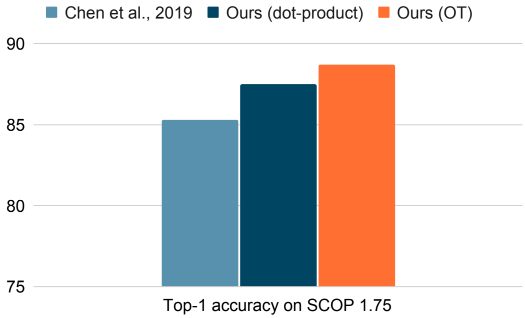
A linearized variant of attention.

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

OTKE: (temporarily) sota for our bioinformatics tasks

SCOP 1.75: 20k samples (classification).

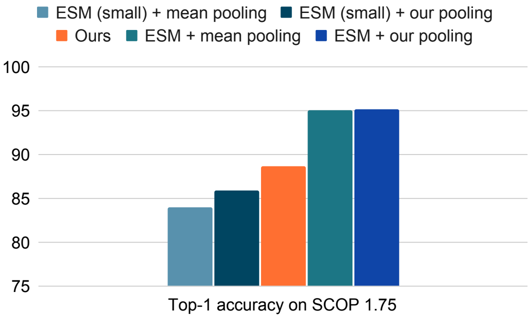
- Classify protein folding from amino-acid sequence: 1k labels!
- Sequence length from 10s to 1000s.



What about pre-trained models?

During ICLR rebuttal...

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



Limitations and perspectives of OTKE

As an embedding.

- Multi-layer version not trivial? [Jaegle et al., 2021]

Limitations and perspectives of OTKE

As an embedding.

- Multi-layer version not trivial? [Jaegle et al., 2021]
- May be outperformed by available pre-trained model.

Limitations and perspectives of OTKE

As an embedding.

- Multi-layer version not trivial? [Jaegle et al., 2021]
- May be outperformed by available pre-trained model.

Perspective: adaptive pooling mechanism for deep architectures?

- Improved pooling for graph representation [Kolouri et al., 2021] or protein representation (ICLR rebuttal).

Limitations and perspectives of OTKE

As an embedding.

- Multi-layer version not trivial? [Jaegle et al., 2021]
- May be outperformed by available pre-trained model.

Perspective: adaptive pooling mechanism for deep architectures?

- Improved pooling for graph representation [Kolouri et al., 2021] or protein representation (ICLR rebuttal).
- Interesting improvement of OTKE by [Anonymous, 2022].

Outline

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

Take-home messages

GraphiT

- Inductive bias of transformers is valid with graphs with small/medium scale datasets.
- Promising interpretation for graphs.

Take-home messages

GraphiT

- Inductive bias of transformers is valid with graphs with small/medium scale datasets.
- Promising interpretation for graphs.

OTK Embedding

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

Take-home messages

GraphiT

- Inductive bias of transformers is valid with graphs with small/medium scale datasets.
- Promising interpretation for graphs.

OTK Embedding

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

Kernel methods

- Reconcile deep learning with smaller data regimes!
- Understanding architectures via a different lens.

Are inductive biases still useful?

In constrained data regime, inductive biases such as kernel methods are still useful!

- Not all domains have large data.
- When large unlabelled data is available, self-supervised learning may not work (yet).

Are inductive biases still useful?

In constrained data regime, inductive biases such as kernel methods are still useful!

- Not all domains have large data.
- When large unlabelled data is available, self-supervised learning may not work (yet).

But in large data regime?

- OTKE beaten by ESM [Rives et al., 2019].
- MLP-Mixer [Tolstikhin et al., 2021], DeiT [Touvron et al., 2020], BiT [Kolesnikov et al., 2020]: the bitter lesson of machine learning more true than ever?

Are inductive biases still useful?

In constrained data regime, inductive biases such as kernel methods are still useful!

- Not all domains have large data.
- When large unlabelled data is available, self-supervised learning may not work (yet).

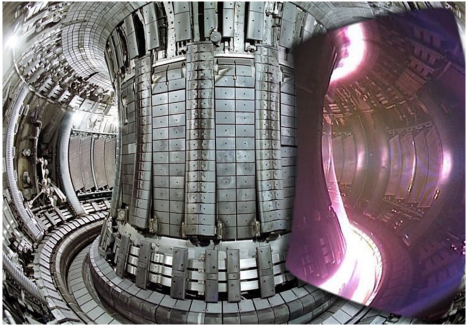
But in large data regime?

- OTKE beaten by ESM [Rives et al., 2019].
- MLP-Mixer [Tolstikhin et al., 2021], DeiT [Touvron et al., 2020], BiT [Kolesnikov et al., 2020]: the bitter lesson of machine learning more true than ever?

But in real life?

- AlphaFold2: physically motivated inductive biases.

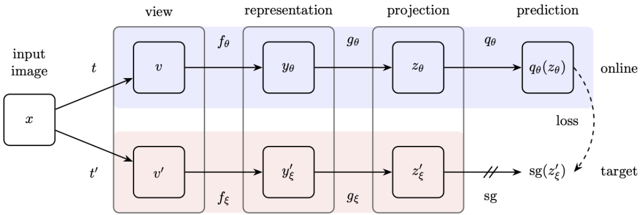
But in real life?



Seek progress elsewhere?

Inductive biases can be found in learning paradigms...

- 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].



BYOL (from Grill et al.).

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*.

Anonymous (2022). Differentiable expectation-maximization for set representation learning. In *Submitted to The Tenth International Conference on Learning Representations*. under review.

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.

References II

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)*.

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*.

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)*.

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

Godwin, J., Schaarschmidt, M., Gaunt, A., Sanchez-Gonzalez, A., Rubanova, Y., Veličković, P., Kirkpatrick, J., and Battaglia, P. (2021). Very deep graph neural networks via noise regularisation.

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.

References IV

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)*.

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.

Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., and Carreira, J. (2021). Perceiver: General perception with iterative attention. In *International Conference on Machine Learning (ICML)*.

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.

References V

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)*.

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)*.

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*.

References VI

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.

Peyré, G. and Cuturi, M. (2019). Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–206.

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.

References VII

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).

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.

References VIII

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

Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., Lucic, M., and Dosovitskiy, A. (2021). Mlp-mixer: An all-mlp architecture for vision.

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*.

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)*.

References IX

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.

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. (2021). Do transformers really perform bad for graph representation?

References X

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