On Inductive Biases for Machine Learning in Data Constrained Settings

Grégoire Mialon

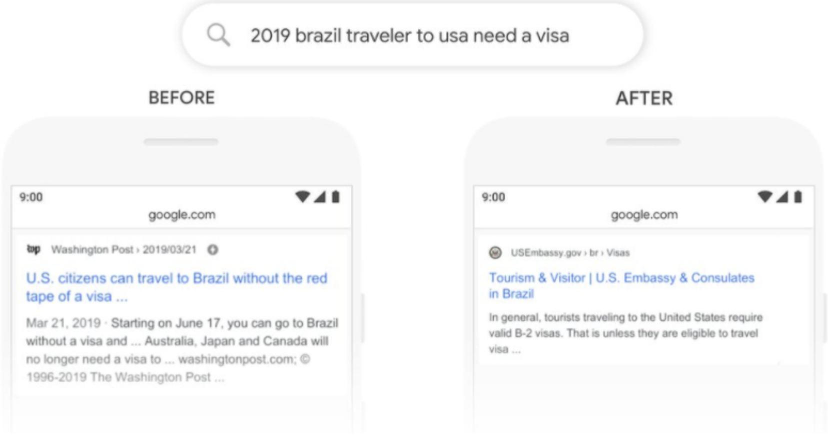
Inria Sierra, Inria Thoth

PhD defense. January 19, 2022

Rapporteurs: Alexandre Gramfort (Inria), Gabriel Peyré (ENS/CNRS)
Examinateurs: Michael Bronstein (Oxford/Twitter), Pascal Frossard (EPFL), Anna Korba (ENSAE/CREST)
Encadrants: Alexandre d'Aspremont (ENS/CNRS), Julien Mairal (Inria)

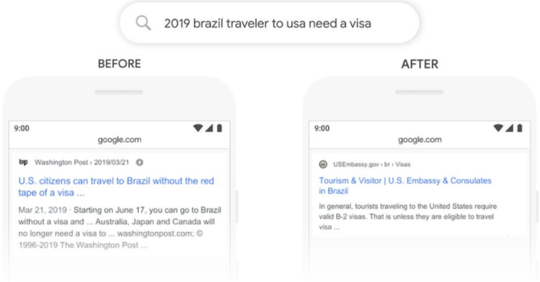
Outline

Introduction: Recent success of machine learning



Improved web search engines.

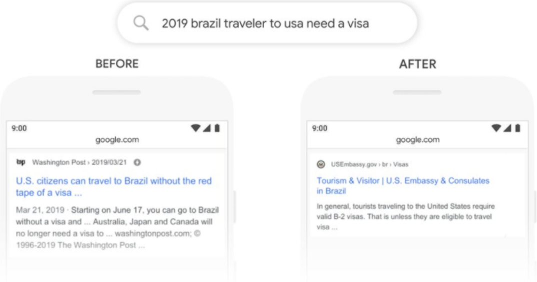
Introduction: Recent success of machine learning



Improved web search engines.

`https://thispersondoesnotexist.com/`

Introduction: Recent success of machine learning



Improved web search engines.

`https://thispersondoesnotexist.com/`

- And also bioinformatics, speech recognition, and many other domains...

Introduction: How does this work?

Recipe: Huge models + huge data + learning problem + optimization algorithm + computing power

Introduction: How does this work?

Today: $\boxed{\text{Huge models}}$ + huge data + learning problem + $\boxed{\text{optimization algorithm}}$ + computing power

Introduction: How does machine learning work? A canonical example

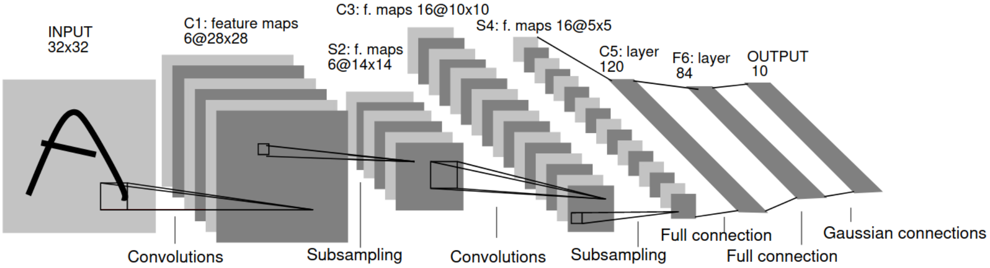
Recipe: Huge models + huge data + learning problem + optimization algorithm + computing power

- Supervised model f takes an input x (*e.g* an image) and outputs a “label” $f(x)$ (e.g a letter).

Introduction: How does machine learning work? A canonical example

Recipe: Huge models + huge data + learning problem + optimization algorithm + computing power

- Supervised model f takes an input x (*e.g* an image) and outputs a "label" $f(x)$ (e.g a letter).
- A neural network model f: $f(x) = W_n(\sigma_n(\dots W_1\sigma_1(x)\dots))$.

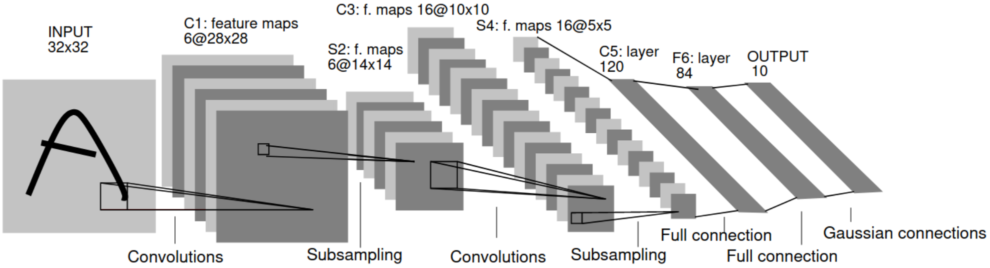


A convolutional neural network (from LeCun et al., 1998).

Introduction: How does machine learning work? A canonical example

Recipe: $\boxed{\text{Huge models}}$ + huge data + learning problem + optimization algorithm + computing power

- Supervised model f takes an input x (*e.g* an image) and outputs a "label" $f(x)$ (e.g a letter).
- A neural network model f: $f(x) = W_n(\sigma_n(\dots W_1\sigma_1(x)\dots))$.



A convolutional neural network (from LeCun et al., 1998).

- Today: Millions of adjustable parameters.

Introduction: How does machine learning work? A canonical example

Recipe: Huge models + $\boxed{\text{huge data}}$ + learning problem + optimization algorithm + computing power

Samples from ImageNet (1.2M images).

Introduction: How does machine learning work? A catnonical example

Recipe: Huge models + huge data + $\boxed{\text{learning problem}}$ + optimization algorithm + computing power

I am organized but lazy: how to automatically classify these images as "cat" or "dog"?

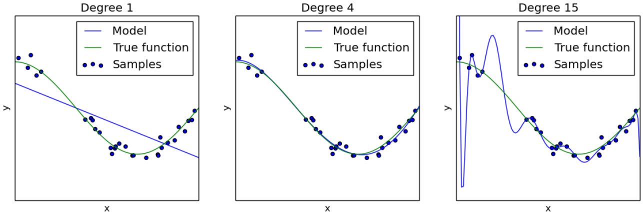
Introduction: How does machine learning work? A canonical example

Recipe: Huge models + huge data + $\boxed{\text{learning problem}}$ + optimization algorithm + computing power

Empirical risk minimization:

$$\min_{\theta \in \mathcal{H}} \mathcal{L}(\theta) = \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell(f_\theta(x_i), y_i)}_{\text{Empirical risk, data fit}} + \underbrace{\lambda R(f_\theta)}_{\text{Regularization}},$$

with f a neural network with parameters θ, x_i an image and y_i a label, here "cat" or "dog".

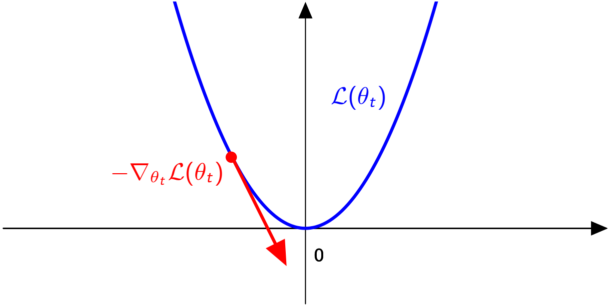


Regularization penalizes the complexity of the model (from `scikit-learn.org`).

Introduction: How does machine learning work? A canonical example

Recipe: Huge models + huge data + learning problem + $\boxed{\text{optimization algorithm}}$ + computing power

Gradient descent:



$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta_t}\mathcal{L}(\theta_t).$$

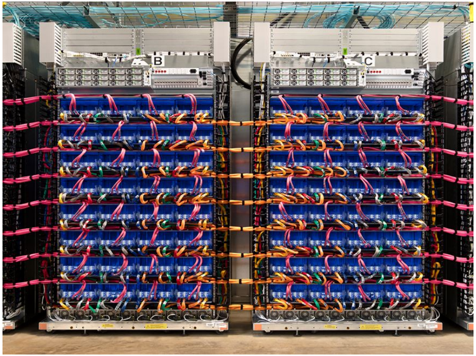
Introduction: How does machine learning work? A canonical example

Recipe: Huge models + huge data + learning problem + optimization algorithm + computing power

Jean Zay supercalculator in Saclay is notably equipped with Tesla V100 computing chips.

Getting back to our introductory example

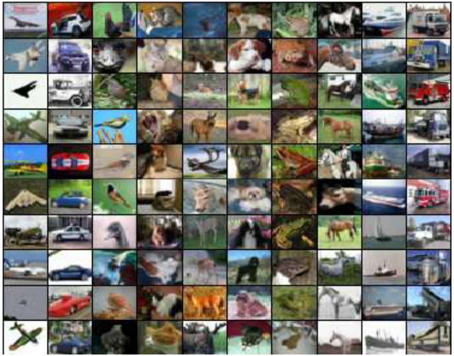
Google "new" search engine [Devlin et al., 2019]: Transformer (340M params) + $\sim$ 33k books + Sentence completion + Stochastic gradient descent + 64 TPUs for 4 days.



TPU chips in a Google data center.

Problem: Deep learning does not work that great on smaller datasets

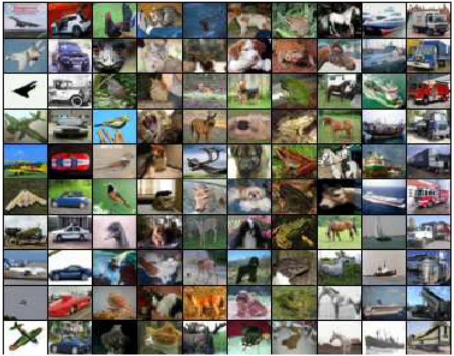
Method	VGG-11	ResNet-18
All (60k) samples	91.0	93.0
5k samples [Bietti et al., 2019]	72.8	73.1
1k samples [Bietti et al., 2019]	51.3	44.9
Random	10.0	10.0



Classification accuracies of convolutional neural networks trained on the image dataset CIFAR-10 (with data augmentation).

Problem: Deep learning does not work that great on smaller datasets

Method	VGG-11	ResNet-18
All (60k) samples	91.0	93.0
5k samples [Bietti et al., 2019]	72.8	73.1
1k samples [Bietti et al., 2019]	51.3	44.9
Random	10.0	10.0



Classification accuracies of convolutional neural networks trained on the image dataset CIFAR-10 (with data augmentation).

- No clear regularization scheme [Bietti, Mialon, Chen and Mairal, ICML 2019].

Learning with smaller datasets is one of the biggest problems in machine learning

Important problems often mean medium or small data.

Learning with smaller datasets is one of the biggest problems in machine learning

Important problems often mean medium or small data.

- Molecules or proteins with rare properties.

Learning with smaller datasets is one of the biggest problems in machine learning

Important problems often mean medium or small data.

- Molecules or proteins with rare properties.
- Less than 30k people per rare disease in France (2021).

Learning with smaller datasets is one of the biggest problems in machine learning

Important problems often mean medium or small data.

- Molecules or proteins with rare properties.
- Less than 30k people per rare disease in France (2021).
- Expensive or complex data collection for fundamental science/econometrics.

Learning with smaller datasets is one of the biggest problems in machine learning

Important problems often mean medium or small data.

- Molecules or proteins with rare properties.
- Less than 30k people per rare disease in France (2021).
- Expensive or complex data collection for fundamental science/econometrics.
- Rare events in self-driving cars datasets.

Learning with smaller datasets is one of the biggest problems in machine learning

Important problems often mean medium or small data.

- Molecules or proteins with rare properties.
- Less than 30k people per rare disease in France (2021).
- Expensive or complex data collection for fundamental science/econometrics.
- Rare events in self-driving cars datasets.

A path towards better models?

Our approach: A slightly different recipe

This thesis: $\boxed{\text{Models + inductive bias}}$ + $\boxed{\text{possibly smaller datasets}}$ + learning problem + optimization algorithm + $\boxed{\text{computing power}}$

Our approach: A slightly different recipe

This thesis: $\boxed{\text{Models + inductive bias}}$ + $\boxed{\text{possibly smaller datasets}}$ + learning problem + optimization algorithm + $\boxed{\text{computing power}}$

Inductive bias: Constraining some parts of the model so that it efficiently learns from the data.

Our approach: A slightly different recipe

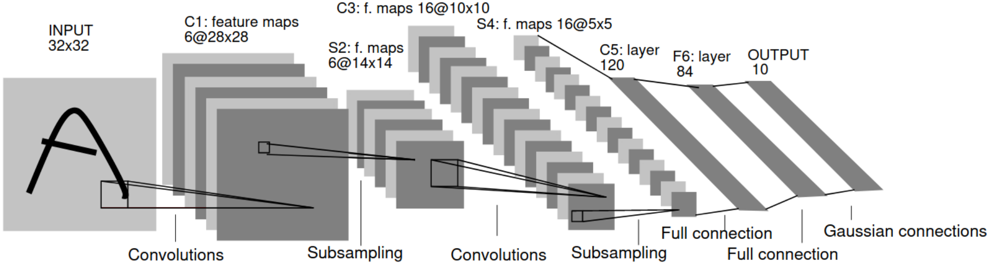
This thesis: $\boxed{\text{Models + inductive bias}}$ + $\boxed{\text{possibly smaller datasets}}$ + learning problem + optimization algorithm + $\boxed{\text{computing power}}$

Inductive bias: Constraining some parts of the model so that it efficiently learns from the data.

Regularization, a simple example of inductive bias:

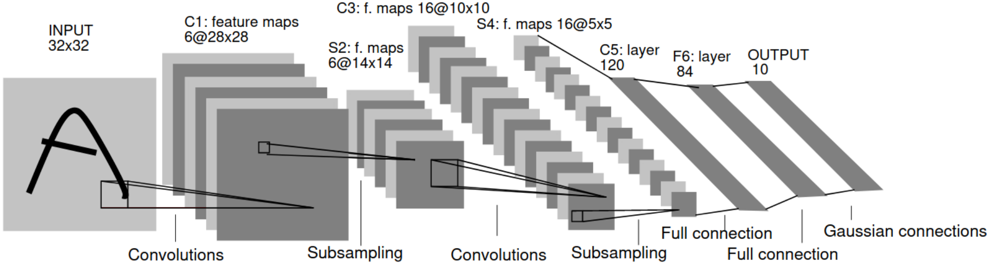
$$\min_{(\theta_1,\theta_0)\in\mathcal{H}} \mathcal{L}(\theta) = \underbrace{\frac{1}{n}\sum_{i=1}^{n}\ell(\theta_1^\top x_i + \theta_0, y_i)}_{\text{Empirical risk, data fit}} + \underbrace{\lambda||\theta_1||_1}_{\text{Regularization}} .$$

Our approach. Another example of inductive bias.



Inductive bias in CNNs:

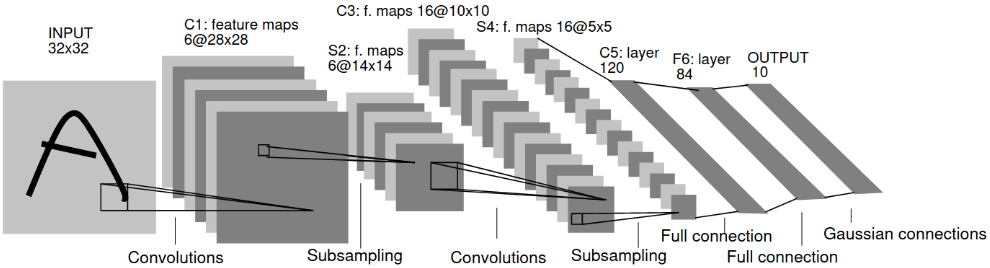
Our approach. Another example of inductive bias.



Inductive bias in CNNs:

- Local pooling.

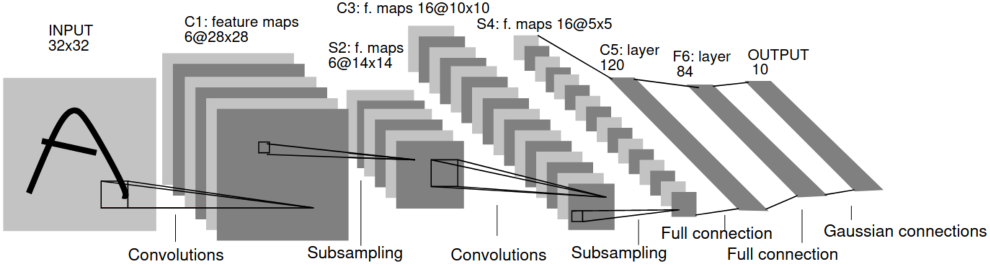
Our approach. Another example of inductive bias.



Inductive bias in CNNs:

- Local pooling.
- Multi-scale modeling.

Our approach. Another example of inductive bias.



Inductive bias in CNNs:

- Local pooling.
- Multi-scale modeling.

Useful for efficient learning from natural images.

Contributions

This thesis: Models + inductive bias + possibly smaller datasets + learning problem + optimization algorithm + computing power

Contributions

This thesis: $\boxed{\text{Models + inductive bias}}$ + possibly smaller datasets + learning problem + optimization algorithm + computing power

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

- A. Bietti*, G. Mialon*, D. Chen, J. Mairal. A Kernel Perspective for Regularizing Deep Neural Networks (ICML, 2019).
- G. Mialon*, D. Chen*, A. d'Aspremont, J. Mairal. A Trainable Optimal Transport Embedding for Feature Aggregation and its Relationship to Attention (ICLR, 2021).
- G. Mialon*, D. Chen*, M. Selosse*, J. Mairal. GraphiT: Encoding Graph Structure in Transformers (arXiv:2106.05667, 2021).

Contributions

This thesis: Models + inductive bias + possibly smaller datasets + learning problem + optimization algorithm + computing power

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

- A. Bietti*, G. Mialon*, D. Chen, J. Mairal. A Kernel Perspective for Regularizing Deep Neural Networks (ICML, 2019).
- G. Mialon*, D. Chen*, A. d'Aspremont, J. Mairal. A Trainable Optimal Transport Embedding for Feature Aggregation and its Relationship to Attention (ICLR, 2021).
- G. Mialon*, D. Chen*, M. Selosse*, J. Mairal. GraphiT: Encoding Graph Structure in Transformers (arXiv:2106.05667, 2021).

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).

Outline

Sets are an important data modality

```
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
L: 2  D: 2  K: 1  V: 2  E: 2  A: 1  Q: 1  I: 2  R: 1  T: 1  G: 1
```

Top: Short part of mRNA sequence for the SARS-Cov-2 spike protein.
Middle: Each triplet codes for an amino acid.
Bottom: Set representation of the sequence (1-grams).

- Biological sequences, *e.g*, proteins.

Sets are an important data modality

```
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
L: 2  D: 2  K: 1  V: 2  E: 2  A: 1  Q: 1  I: 2  R: 1  T: 1  G: 1
```

Top: Short part of mRNA sequence for the SARS-Cov-2 spike protein.
Middle: Each triplet codes for an amino acid.
Bottom: Set representation of the sequence (1-grams).

- Biological sequences, *e.g*, proteins.
- Sentences in natural language processing (NLP), 3D point cloud in computer vision.

Sets are an important data modality

```
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
L: 2   D: 2   K: 1   V: 2   E: 2   A: 1   Q: 1   I: 2   R: 1   T: 1   G: 1
```

Top: Short part of mRNA sequence for the SARS-Cov-2 spike protein.
Middle: Each triplet codes for an amino acid.
Bottom: Set representation of the sequence (1-grams).

- Biological sequences, *e.g*, proteins.
- Sentences in natural language processing (NLP), 3D point cloud in computer vision.
- Different cardinalities, potentially **long**, with **few labelled sample** per class.

Focusing 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
L: 2   D: 2   K: 1   V: 2   E: 2   A: 1   Q: 1   I: 2   R: 1   T: 1   G: 1
```

Top: Short part of mRNA sequence for the SARS-Cov-2 spike protein.
Middle: Each triplet codes for an amino acid.
Bottom: Set representation of the sequence (1-grams).

Existing methods do not yield satisfactory results for our data.

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

Focusing 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
L: 2   D: 2   K: 1   V: 2   E: 2   A: 1   Q: 1   I: 2   R: 1   T: 1   G: 1
```

Top: Short part of mRNA sequence for the SARS-Cov-2 spike protein.
Middle: Each triplet codes for an amino acid.
Bottom: Set representation of the sequence (1-grams).

Existing methods do not yield satisfactory results for our data.

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

Focusing 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
L: 2  D: 2  K: 1  V: 2  E: 2  A: 1  Q: 1  I: 2  R: 1  T: 1  G: 1
```

Top: Short part of mRNA sequence for the SARS-Cov-2 spike protein.
Middle: Each triplet codes for an amino acid.
Bottom: Set representation of the sequence (1-grams).

Existing methods do not yield satisfactory results for our data.

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

How to represent sets with low data and memory requirements?

An attractive kernel for sets

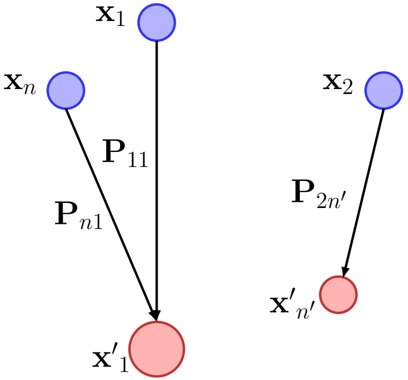
Kernel methods [Schölkopf and Smola, 2001] allow rich representation of the data.

- Let $\mathbf{x} \in \mathbb{R}^{n \times d}, \mathbf{x}' \in \mathbb{R}^{n' \times d}$ be two sets of feature vectors. The Optimal Transport Match Kernel is defined as

$$K_{\mathrm{OT}}(\mathbf{x}, \mathbf{x}') = \sum_{i,j} \mathbf{P}_{ij} \langle \mathbf{x}_i, \mathbf{x}'_j \rangle,$$

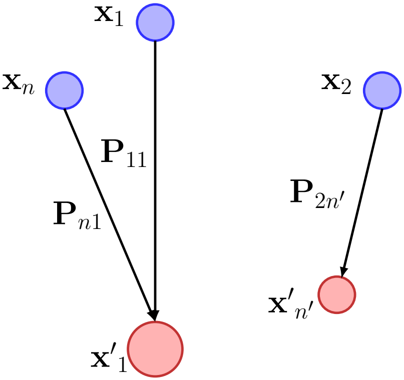
  where $\mathbf{P} \in \mathbb{R}^{n \times n'}$ is the solution to the regularized optimal transport problem between $\mathbf{x}$ and $\mathbf{x}'$.
- Intuitively, $K_{\mathrm{OT}}(\mathbf{x}, \mathbf{x}')$ high if $\mathbf{x}$ and $\mathbf{x}'$ are easy to align.

(Regularized) Optimal transport



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

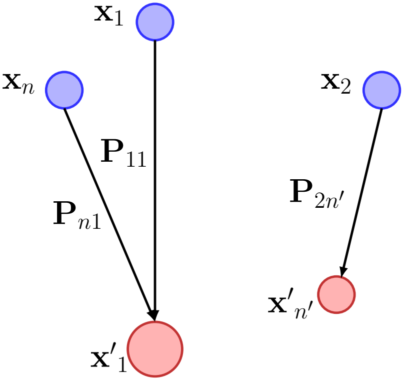
(Regularized) Optimal transport



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

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

(Regularized) Optimal transport



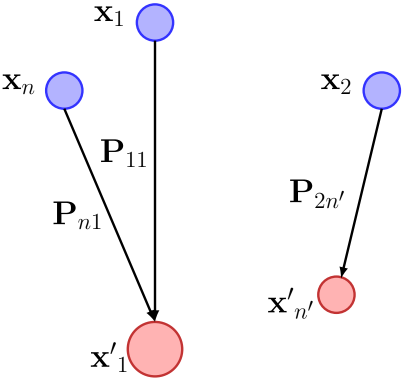
$$\mathbf{P} = \begin{pmatrix} \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} \end{pmatrix}$$

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

$$\min_{\mathbf{P} \in U} \sum_{ij} \mathbf{C}_{ij} \mathbf{P}_{ij} - \varepsilon H(\mathbf{P}),$$

with $H(\mathbf{P}) = -\sum_{ij} \mathbf{P}_{ij}(\log(\mathbf{P}_{ij}) - 1)$, and U, the space of admissible couplings.

(Regularized) Optimal transport



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

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

$$\min_{\mathbf{P} \in U} \sum_{ij} \mathbf{C}_{ij} \mathbf{P}_{ij} - \varepsilon H(\mathbf{P}),$$

  with $H(\mathbf{P}) = -\sum_{ij} \mathbf{P}_{ij}(\log(\mathbf{P}_{ij}) - 1)$, and U, the space of admissible couplings.
- GPU-friendly [Sinkhorn and Knopp, 1967, Cuturi and Doucet, 2013].

Back to our problem

$$K_{\mathrm{OT}}(\mathbf{x}, \mathbf{x}') = \sum_{i,j} \mathbf{P}_{ij} \langle \mathbf{x}_i, \mathbf{x}'_j \rangle.$$

We cannot directly use K_{OT}.

- K_{OT} is not positive definite [Gardner et al., 2018].

[Mialon et al., 2021a]

Back to our problem

$$K_{\mathrm{OT}}(\mathbf{x}, \mathbf{x}') = \sum_{i,j} \mathbf{P}_{ij} \langle \mathbf{x}_i, \mathbf{x}'_j \rangle.$$

We cannot directly use K_{OT}.

- K_{OT} is not positive definite [Gardner et al., 2018].
- Observation:

$$\mathbf{P}_{\mathbf{z}}(\mathbf{x}, \mathbf{x}') := p \times \mathbf{P}(\mathbf{x}, \mathbf{z}) \mathbf{P}(\mathbf{x}', \mathbf{z})^\top$$

  is a valid transport plan between $\mathbf{x}'$ and $\mathbf{x}$ [Peyré and Cuturi, 2019].

[Mialon et al., 2021a]

Back to our problem

$$K_{\mathrm{OT}}(\mathbf{x}, \mathbf{x}') = \sum_{i,j} \mathbf{P}_{ij} \langle \mathbf{x}_i, \mathbf{x}'_j \rangle.$$

We cannot directly use K_{OT}.

- K_{OT} is not positive definite [Gardner et al., 2018].
- Observation:

  $$\mathbf{P}_{\mathbf{z}}(\mathbf{x}, \mathbf{x}') := p \times \mathbf{P}(\mathbf{x}, \mathbf{z}) \mathbf{P}(\mathbf{x}', \mathbf{z})^\top$$

  is a valid transport plan between $\mathbf{x}'$ and $\mathbf{x}$ [Peyré and Cuturi, 2019].
- Positive definite surrogate for K_{OT}:

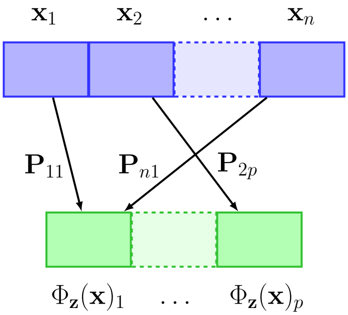
  $$K_{\mathbf{z}}(\mathbf{x}, \mathbf{x}') := \langle \mathbf{P}_{\mathbf{z}}(\mathbf{x}, \mathbf{x}'), \kappa(\mathbf{x}, \mathbf{x}') \rangle = \langle \Phi_{\mathbf{z}}(\mathbf{x}), \Phi_{\mathbf{z}}(\mathbf{x}') \rangle,$$

  with

  $$\Phi_{\mathbf{z}}(\mathbf{x}) = \sqrt{p} \times \mathbf{P}(\mathbf{x}, \mathbf{z})^\top \mathbf{x}.$$

[Mialon et al., 2021a]

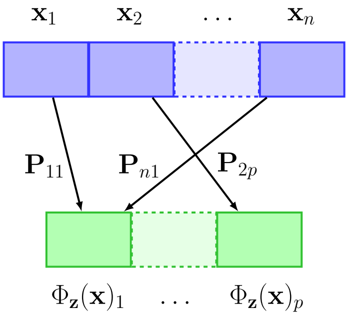
Core contribution: An optimal transport pooling



Global, similarity-based pooling in p bins.

[Mialon et al., 2021a]

Core contribution: An optimal transport pooling

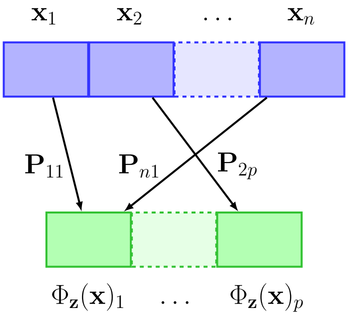


[Mialon et al., 2021a]

Global, similarity-based pooling in p bins.

- To each bin corresponds a prototype (parameter) $z_j \in \mathbb{R}^d$, $j = 1 \dots p$.

Core contribution: An optimal transport pooling

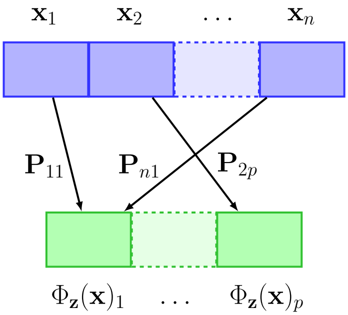


Global, similarity-based pooling in p bins.

- To each bin corresponds a prototype (parameter) $z_j \in \mathbb{R}^d$, $j = 1 \dots p$.
- Input: set or sequence $\mathbf{x} \in \mathbb{R}^{n \times d}$.

[Mialon et al., 2021a]

Core contribution: An optimal transport pooling



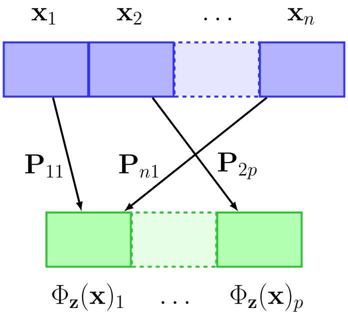
Global, similarity-based pooling in p bins.

- To each bin corresponds a prototype (parameter) $z_j \in \mathbb{R}^d$, $j = 1 \dots p$.
- Input: set or sequence $\mathbf{x} \in \mathbb{R}^{n \times d}$.
- Output: $\Phi_z(\mathbf{x})_j \in \mathbb{R}^{p \times d}$

$$\Phi_z(\mathbf{x})_j = \sum_{i=1}^{n} \mathbf{P}_{ij} \mathbf{x}_i.$$

[Mialon et al., 2021a]

Core contribution: An optimal transport pooling



Global, similarity-based pooling in p bins.

- To each bin corresponds a prototype (parameter) $z_j \in \mathbb{R}^d$, $j = 1 \dots p$.
- Input: set or sequence $\mathbf{x} \in \mathbb{R}^{n \times d}$.
- Output: $\Phi_z(\mathbf{x})_j \in \mathbb{R}^{p \times d}$

$$\Phi_z(\mathbf{x})_j = \sum_{i=1}^{n} \mathbf{P}_{ij} \mathbf{x}_i.$$

- **z** learned **with or without** supervision.

[Mialon et al., 2021a]

Results

Results in various domains: Images, text, biological sequences.

[Mialon et al., 2021a]

Results

Results in various domains: Images, text, biological sequences.

SST-2 (70k paragraphs, classification): Classifying movie reviews in English into positive or negative.

[Mialon et al., 2021a]

Results

Results in various domains: Images, text, biological sequences.

SST-2 (70k paragraphs, classification): Classifying movie reviews in English into positive or negative.

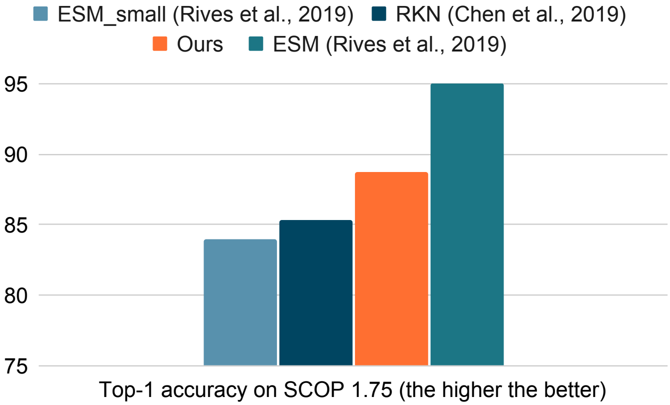
Classification accuracies on validation set, averaged from 10 different runs (q references $\times$ p supports).

Method	Unsupervised	Supervised
[CLS] embedding [Devlin et al., 2019]	84.6±0.3	90.3±0.1
Mean Pooling of BERT features [Devlin et al., 2019]	85.3±0.4	**90.8±0.1**
Approximate Rep the Set [Skianis et al., 2020]	Not available.	86.8±0.9
Rep the Set [Skianis et al., 2020]	Not available.	87.1±0.5
Set Transformer [Lee et al., 2019]	Not available.	87.9±0.8
Ours (Unsupervised: 1×300. Supervised: 4×30)	**86.8±0.3**	88.1±0.8

[Mialon et al., 2021a]

Results

SCOP 1.75 (20k sequences, classification): Predicting protein folding.

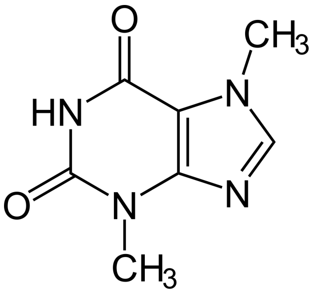


- ESM trained on 250M protein sequences!

[Mialon et al., 2021a]

Outline

Graph data are an important research topic

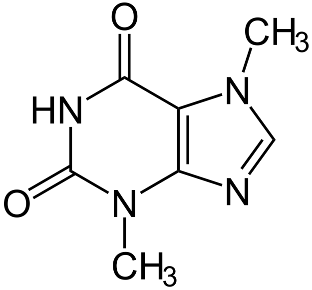


A molecule of theobromin, or why chocolate makes us feel good.

Graph data are very valuable...

- Molecules in chemoinformatics.

Graph data are an important research topic

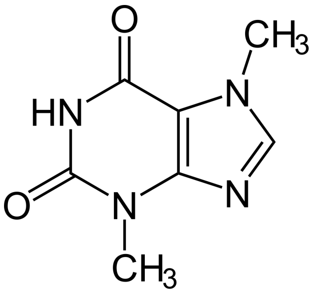


A molecule of theobromin, or why chocolate makes us feel good.

Graph data are very valuable...

- Molecules in chemoinformatics.
- Proteins in computational biology.

Graph data are an important research topic

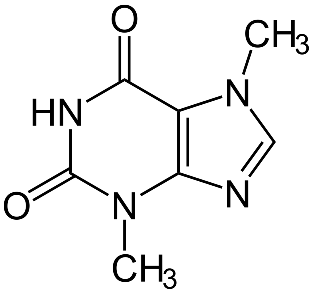


A molecule of theobromin, or why chocolate makes us feel good.

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.

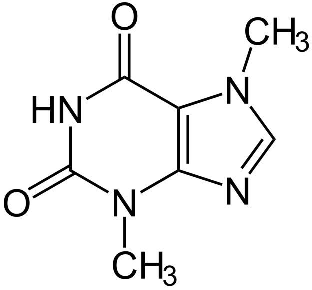
Graph data are very valuable...

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

...but delicate to exploit.

- Non-Euclidean structure.

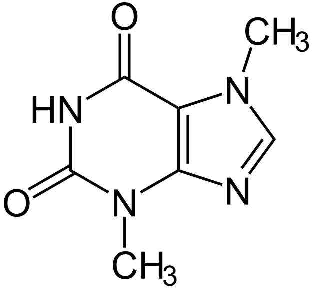
Success and current limits of neural networks for graphs



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

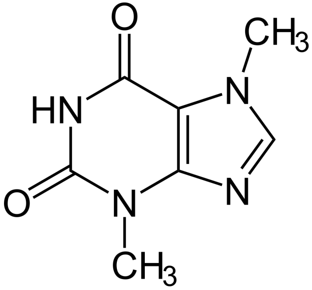


A molecule of theobromin, or why chocolate makes us feel good.

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

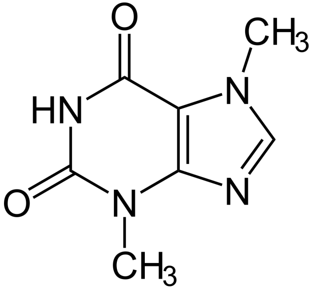


A molecule of theobromin, or why chocolate makes us feel good.

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

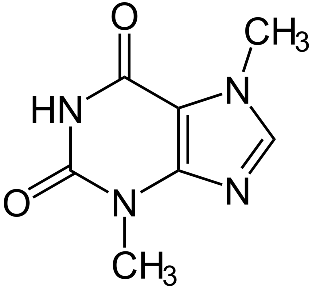


A molecule of theobromin, or why chocolate makes us feel good.

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 for graph are tempting but not straightforward

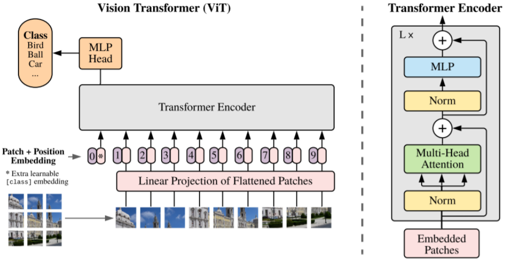


Image transformer (from [Dosovitskiy et al., 2021]).
Input: image seen as a set of patches.
Output: class label.

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

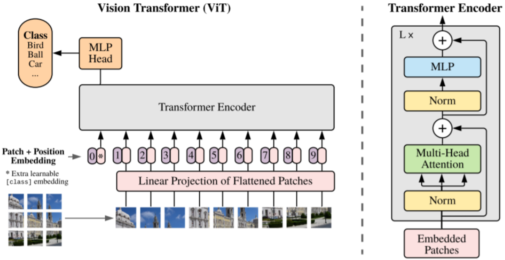


Image transformer (from [Dosovitskiy et al., 2021]).
Input: image seen as a set of patches.
Output: class label.

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

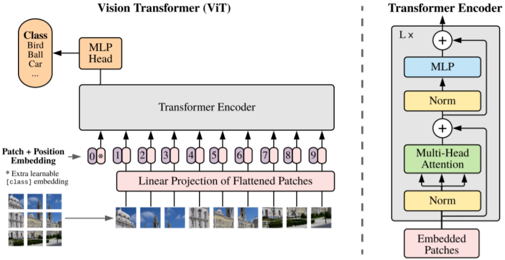


Image transformer (from [Dosovitskiy et al., 2021]).
Input: image seen as a set of patches.
Output: class label.

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

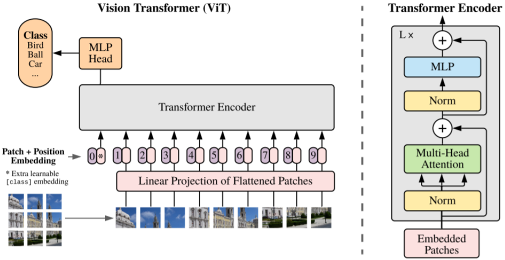


Image transformer (from [Dosovitskiy et al., 2021]).
Input: image seen as a set of patches.
Output: class label.

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

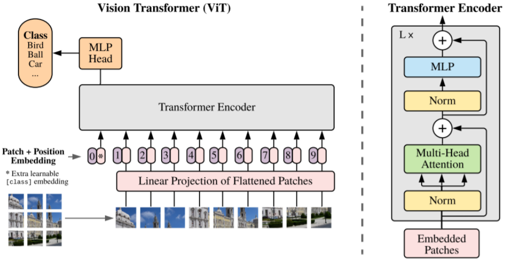


Image transformer (from [Dosovitskiy et al., 2021]).
Input: image seen as a set of patches.
Output: class label.

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

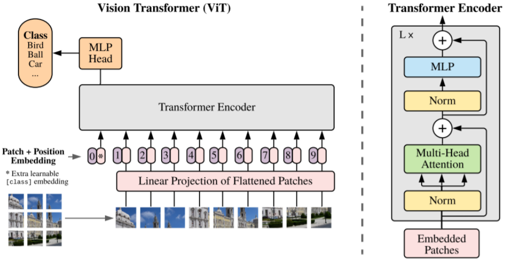


Image transformer (from [Dosovitskiy et al., 2021]).
Input: image seen as a set of patches.
Output: class label.

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

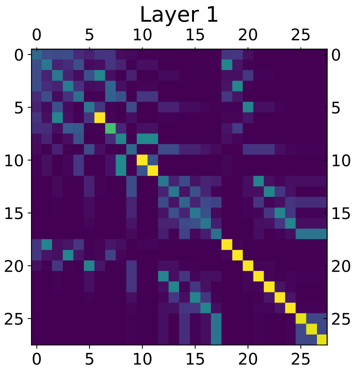
- Text [Devlin et al., 2019],
  Proteins [Rives et al., 2019],
  Images [Dosovitskiy et al., 2021].
- Rarely used for graphs.

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?

Our contribution: GraphiT, encoding graph structure in transformers

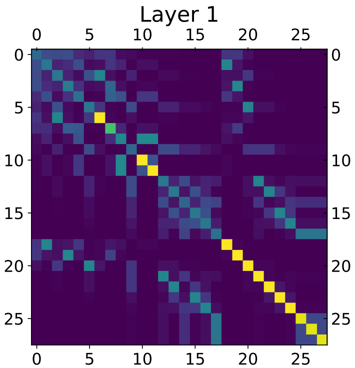


Diffusion kernel between the nodes of a Mutagenicity sample graph ($\beta = 1$).

[Mialon et al., 2021b]

We propose two mechanisms:

Our contribution: GraphiT, encoding graph structure in transformers



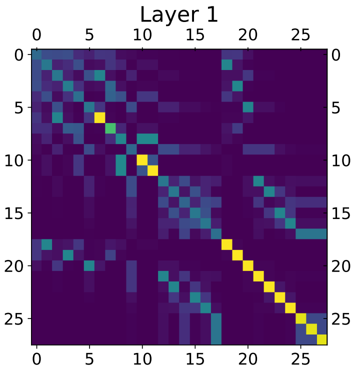
Diffusion kernel between the nodes of a Mutagenicity sample graph ($\beta = 1$).

[Mialon et al., 2021b]

We propose two mechanisms:

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

Our contribution: GraphiT, encoding graph structure in transformers



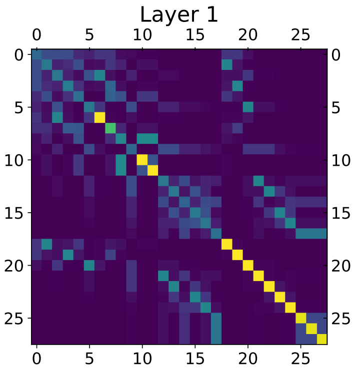
Diffusion kernel between the nodes of a Mutagenicity sample graph ($\beta = 1$).

[Mialon et al., 2021b]

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].

Our contribution: GraphiT, encoding graph structure in transformers



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

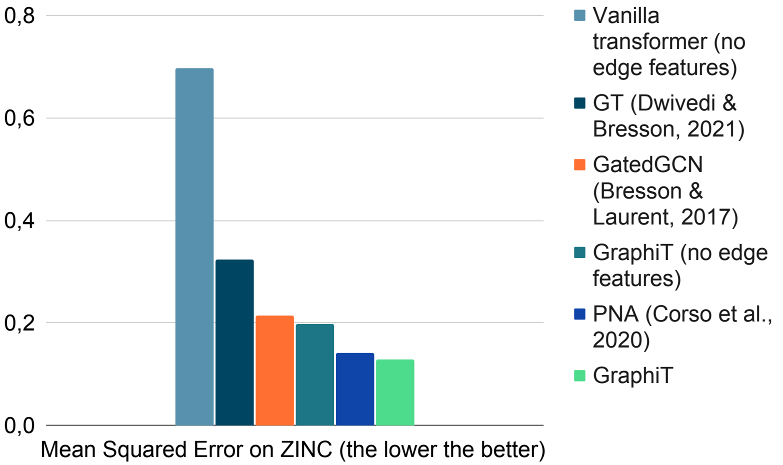
[Mialon et al., 2021b]

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.

GraphiT is able to outperform popular GNNs

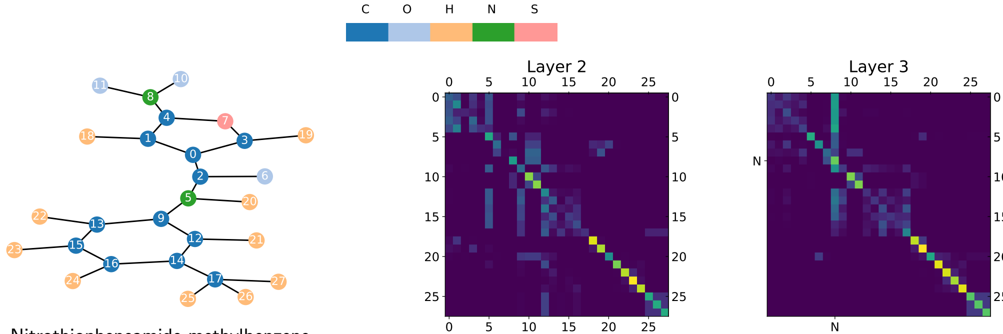
ZINC (12k graphs, regression): Predicting the constrained differential solubility of molecules.



[Mialon et al., 2021b]

GraphiT captures meaningful interactions

Mutagenicity: 4k samples (binary classification).



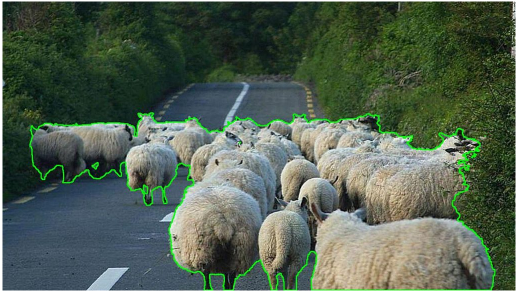
Nitrothiopheneamide-methylbenzene

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., 2021b]

Outline

Safe sample screening

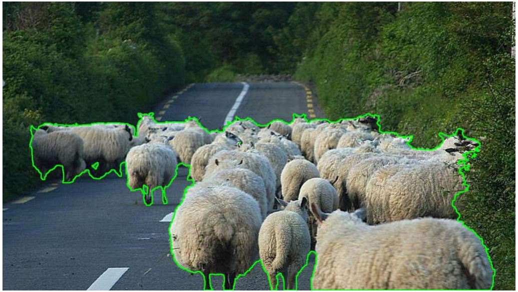


Self-driving cars critically need to detect anomalies.

Why getting rid of data?

- To detect anomalies.

Safe sample screening

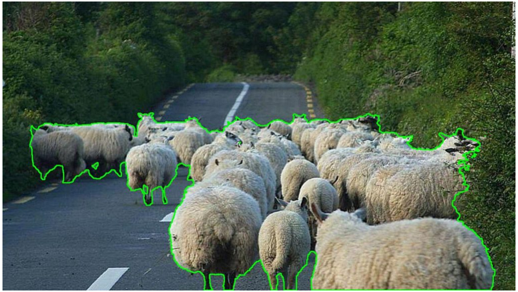


Self-driving cars critically need to detect anomalies.

Why getting rid of data?

- To detect anomalies.
- To accelerate solvers.

Safe sample screening

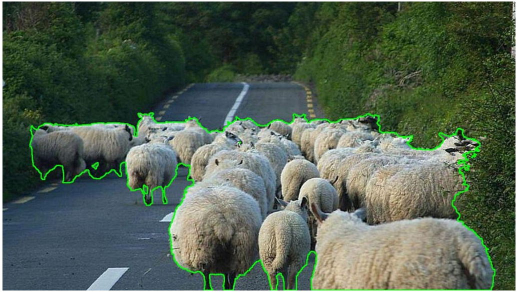


Self-driving cars critically need to detect anomalies.

Why getting rid of data?

- To detect anomalies.
- To accelerate solvers.
- Because it is interesting.

Safe sample screening



Self-driving cars critically need to detect anomalies.

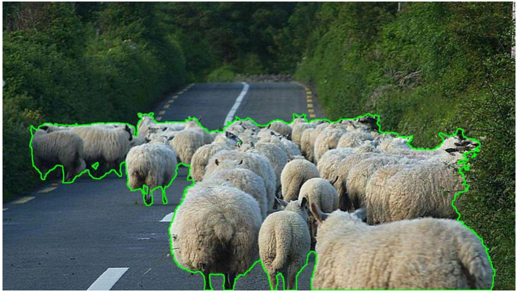
Why getting rid of data?

- To detect anomalies.
- To accelerate solvers.
- Because it is interesting.

Context:

- Convex problems.

Safe sample screening



Self-driving cars critically need to detect anomalies.

Why getting rid of data?

- To detect anomalies.
- To accelerate solvers.
- Because it is interesting.

Context:

- Convex problems.
- Rich literature for feature screening [Ghaoui et al., 2010, Fercoq et al., 2015, Massias et al., 2018].

A simple observation

Empirical risk minimization problem:

$$\min_{x \in \mathbb{R}^p, t \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} f(t_i) + \lambda R(x)$$
$$\text{s.t } t = \mathbf{diag}(b) A x,$$

with f a convex loss and $t = b_i x^\top a_i$ (classification).

A simple observation

Empirical risk minimization problem:

$$\min_{x \in \mathbb{R}^p, t \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} f(t_i) + \lambda R(x)$$
$$\text{s.t } t = \textbf{diag}(b) A x,$$

with f a convex loss and $t = b_i x^\top a_i$ (classification).

Dual problem:

$$\max_{\nu \in \mathbb{R}^n} D(\nu) = \frac{1}{n} \sum_{i=1}^{n} -f_i^*(\nu_i) - \lambda R^* \left(-\frac{A^T \nu}{\lambda n} \right).$$

At the optimum, $x^\star = -\frac{A^\top \nu^\star}{\lambda n}$, with $x^\star$ and $\nu^\star$ resp. the optimal primal and dual variables.

A simple observation

Empirical risk minimization problem:

$$\min_{x\in\mathbb{R}^p, t\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n} f(t_i) + \lambda R(x)$$
$$\text{s.t } t = \mathbf{diag}(b)Ax,$$

with f a convex loss and $t = b_i x^\top a_i$ (classification).

Dual problem:

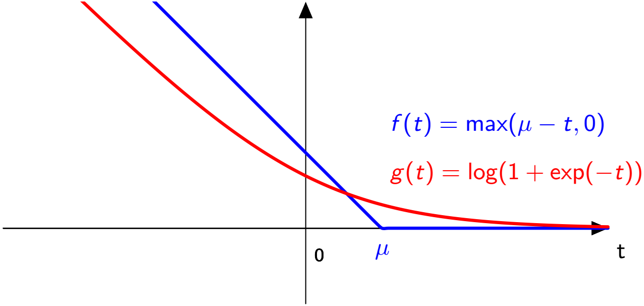
$$\max_{\nu\in\mathbb{R}^n} D(\nu) = \frac{1}{n}\sum_{i=1}^{n} -f_i^*(\nu_i) - \lambda R^*\left(-\frac{A^T\nu}{\lambda n}\right).$$

At the optimum, $x^\star = -\frac{A^\top \nu^\star}{\lambda n}$, with $x^\star$ and $\nu^\star$ resp. the optimal primal and dual variables.

Lemma (Safe loss and dual sparsity)

Consider the primal dual problems above. We have for all $i = 1, \ldots, n$, $\nu_i^\star \in \partial f_i(a_i^\top x^\star)$.

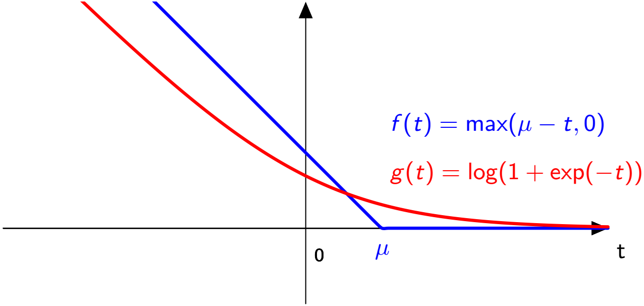
A simple observation



- The sparsity of the dual solution is related to loss functions that have **flat regions**:

$$\nu_i^\star \in \partial f_i(a_i^\top x^\star).$$

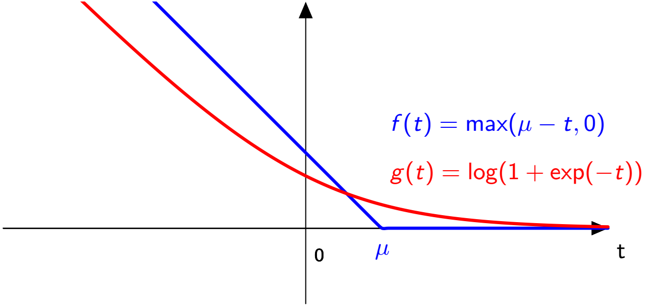
A simple observation



- The sparsity of the dual solution is related to loss functions that have **flat regions**:
$$\nu_i^\star \in \partial f_i(a_i^\top x^\star).$$

- Consider $\mathcal{X}$ such that it contains $x^\star$.

A simple observation

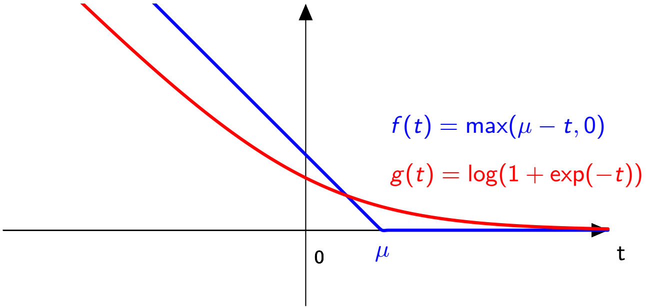


- The sparsity of the dual solution is related to loss functions that have **flat regions**:

$$\nu_i^\star \in \partial f_i(a_i^\top x^\star).$$

- Consider $\mathcal{X}$ such that it contains $x^\star$.
- If for a given sample a_i and every $x \in \mathcal{X}$ we are beyond μ, we can **delete the sample**.

A simple observation



- The sparsity of the dual solution is related to loss functions that have **flat regions**:

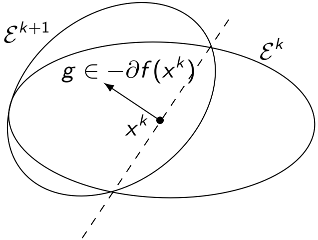
$$\nu_i^\star \in \partial f_i(a_i^\top x^\star).$$

- Consider $\mathcal{X}$ such that it contains $x^\star$.
- If for a given sample a_i and every $x \in \mathcal{X}$ we are beyond μ, we can **delete the sample**.
- Sample screening rule:

$$\min_{x \in \mathcal{X}} b_i a_i^\top x > \mu?$$

Core contribution: A generic algorithm for finding a region containing $x^\star$

Ellipsoid method [Nemirovskii and Yudin, 1979].

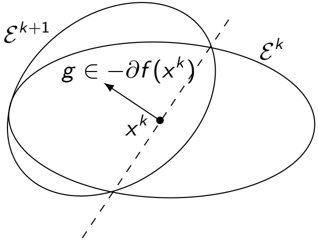


One step of the ellipsoid method.

[Mialon et al., 2020]

Core contribution: A generic algorithm for finding a region containing $x^\star$

Ellipsoid method [Nemirovskii and Yudin, 1979].



One step of the ellipsoid method.

Why ellipsoid method?

- Ellipsoidal region $\mathcal{X}$ enables a **closed-form** test.
- Does not require strong convexity.

[Mialon et al., 2020]

Core contribution: A generic algorithm for finding a region containing $x^\star$

Method	Strongly convex	Non strongly convex	Generic
Pathwise SVM [Ogawa et al., 2013]	✓	✗	✗
Duality Gap [Shibagaki et al., 2016]	✓	✗	✓
Ellipsoid (Ours)	✓	✓	✓

[Mialon et al., 2020]

Core contribution: A generic algorithm for finding a region containing $x^\star$

Method	Strongly convex	Non strongly convex	Generic
Pathwise SVM [Ogawa et al., 2013]	✓	✗	✗
Duality Gap [Shibagaki et al., 2016]	✓	✗	✓
Ellipsoid (Ours)	✓	✓	✓

Perspectives:

- With ellipsoid method, finding a good test region is often **as costly as solving the problem**.
- Preferred use case: warm start, or within a solver [Fercoq et al., 2015].

[Mialon et al., 2020]

Outline

Conclusion

1 - Optimal transport embedding [Mialon et al., 2021a]

- Handling long sequences with few data.
- New pooling mechanism connected to a recent line of work on transformers.

Conclusion

1 - Optimal transport embedding [Mialon et al., 2021a]

- Handling long sequences with few data.
- New pooling mechanism connected to a recent line of work on transformers.

2 - GraphiT [Mialon et al., 2021b]

- Inductive bias of transformers is valid for graphs.
- Promising interpretation capabilities.

Conclusion

1 - Optimal transport embedding [Mialon et al., 2021a]

- Handling long sequences with few data.
- New pooling mechanism connected to a recent line of work on transformers.

2 - GraphiT [Mialon et al., 2021b]

- Inductive bias of transformers is valid for graphs.
- Promising interpretation capabilities.

Are inductive biases still useful?

- Difficult to rival with huge pre-trained models, but pre-training is not always possible.

Conclusion

1 - Optimal transport embedding [Mialon et al., 2021a]

- Handling long sequences with few data.
- New pooling mechanism connected to a recent line of work on transformers.

2 - GraphiT [Mialon et al., 2021b]

- Inductive bias of transformers is valid for graphs.
- Promising interpretation capabilities.

Are inductive biases still useful?

- Difficult to rival with huge pre-trained models, but pre-training is not always possible.
- AlphaFold2 [Jumper et al., 2021]: physically motivated inductive biases.

Conclusion

1 - Optimal transport embedding [Mialon et al., 2021a]

- Handling long sequences with few data.
- New pooling mechanism connected to a recent line of work on transformers.

2 - GraphiT [Mialon et al., 2021b]

- Inductive bias of transformers is valid for graphs.
- Promising interpretation capabilities.

Are inductive biases still useful?

- Difficult to rival with huge pre-trained models, but pre-training is not always possible.
- AlphaFold2 [Jumper et al., 2021]: physically motivated inductive biases.

3 - Safe sample screening [Mialon et al., 2020]

- Better understanding of screening rules.

Perspectives

Further work

- Optimal transport embedding: Further theoretical study needed.

Perspectives

Further work

- Optimal transport embedding: Further theoretical study needed.
- GraphiT: Transformers vs/with GNNs for graphs.

Perspectives

Further work

- Optimal transport embedding: Further theoretical study needed.
- GraphiT: Transformers vs/with GNNs for graphs.
- Both: application in fundamental science.

Drug design, a potential application of ML on sequences and graphs?

Perspectives

Further work

- Sample screening: Application in differential privacy?

Perspectives

Further work

- Sample screening: Application in differential privacy?

Recipe: Huge models + huge data + $\boxed{\text{learning problem}}$ + optimization algorithm + computing power

Seek progress elsewhere? Inductive biases can be found in learning paradigms...

Perspectives

Further work

- Sample screening: Application in differential privacy?

Recipe: Huge models + huge data + $\boxed{\text{learning problem}}$ + optimization algorithm + computing power

Seek progress elsewhere? Inductive biases can be found in learning paradigms...

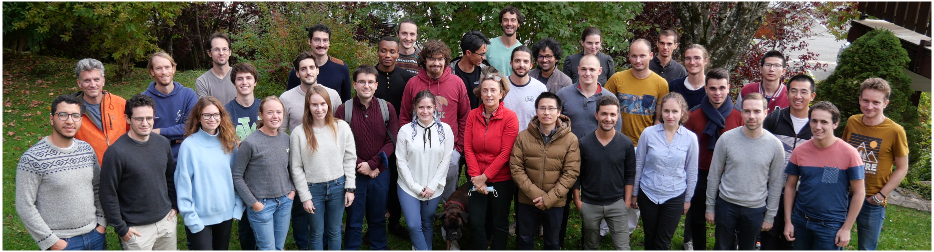
- Data augmentation and loss in self-supervised learning [He et al., 2020, Caron et al., 2020, Grill et al., 2020, Zbontar et al., 2021].

Collaborators





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*.

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)*.

Bietti, A., Mialon, G., Chen, D., and Mairal, J. (2019). A kernel perspective for regularizing deep neural networks. In *International Conference on Machine Learning*, pages 664–674. PMLR.

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)*.

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 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)*.

Fercoq, O., Gramfort, A., and Salmon, J. (2015). Mind the duality gap: safer rules for the Lasso. In *International Conference on Machine Learning (ICML)*.

Gardner, A., Duncan, C. A., Kanno, J., and Selmic, R. R. (2018). On the definiteness of earth mover's distance and its relation to set intersection. *IEEE Transactions on Cybernetics*, 48(11):3184–3196.

Ghaoui, L. E., Viallon, V., and Rabbani, T. (2010). Safe feature elimination for the lasso and sparse supervised learning problems. *arXiv preprint arXiv:1009.4219*.

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 III

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.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589.

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.

References IV

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*.

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

Massias, M., Gramfort, A., and Salmon, J. (2018). Celer: a Fast Solver for the Lasso with Dual Extrapolation. In *International Conference on Machine Learning (ICML)*.

Mialon, G., Chen, D., d'Aspremont, A., and Mairal, J. (2021a). A trainable optimal transport embedding for feature aggregation and its relationship to attention. In *International Conference on Learning Representations (ICLR)*.

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

References V

Mialon, G., Mairal, J., and d'Aspremont, A. (2020). Screening data points in empirical risk minimization via ellipsoidal regions and safe loss functions. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Nemirovskii, A. and Yudin, D. (1979). Problem complexity and method efficiency in optimization. *Nauka*.

Nesterov, Y. (2005). Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152.

Ogawa, K., Suzuki, Y., and Takeuchi, I. (2013). Safe screening of non-support vectors in pathwise svm computation. In *International Conference on Machine Learning (ICML)*.

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*.

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 VI

Schölkopf, B. and Smola, A. J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Shibagaki, A., Karasuyama, M., Hatano, K., and Takeuchi, I. (2016). Simultaneous Safe Screening of Features and Samples in Doubly Sparse Modeling. In *International Conference on Machine Learning (ICML)*.

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.

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 VII

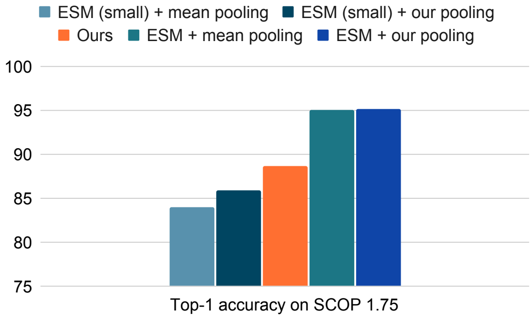
Xie, T., Bapst, V., Gaunt, A. L., Obika, A., Back, T., Hassabis, D., Kohli, P., and Kirkpatrick, J. (2021). Atomistic graph networks for experimental materials property prediction.

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

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.



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

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

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

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

- 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.

Safe algorithm

Algorithm 1 Building ellipsoidal test regions

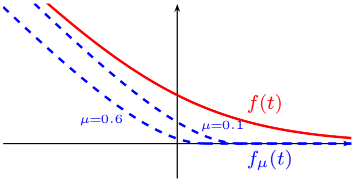
initialization: Given $\mathcal{E}^0(x_0, E_0)$ containing $x^\star$;
while $k < nb_{\text{steps}}$ **do**
  • Compute a gradient g of the objective in x_k;
  • $\tilde{g} \leftarrow g / \sqrt{g^T E_k g}$;
  • $x_{k+1} \leftarrow x_k - \frac{1}{p+1} E_k \tilde{g}$;
  • $E_{k+1} \leftarrow \frac{p^2}{p^2-1}(E_k - \frac{2}{p+1} E_k \tilde{g} \tilde{g}^T E_k)$;
For classification problems:
for each sample a_i in A **do**
  if $\min b_i x^\top a_i \geq \mu$ for $x \in \mathcal{E}^{nb_{\text{steps}}}$ **then**
    Discard a_i from A.

Example of safe loss

Logistic loss: $f(t) = \log(1 + e^{-t})$ and $\Omega(x) = -x\log(-x) + \mu|x|$ for $x \in [-1, 0]$. We have $\Omega^*(y) = -e^{y+\mu-1}$. Convolving Ω^* with f yields

$$f_\mu(x) = \begin{cases} e^{x+\mu-1} - (x+\mu) & \text{if } x + \mu - 1 \leq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Smooth and asymptotically robust. The entropic part of Ω makes this penalty strongly convex hence f_μ is smooth [Nesterov, 2005]. Finally, the ℓ_1 penalty ensures that the dual is sparse thus making the screening usable.



Safe logistic loss.