

# Learning to Predict Graphs with Fused Gromov-Wasserstein Barycenters

Luc Brogat-Motte<sup>1</sup>, Rémi Flamary<sup>2</sup>, Céline Brouard<sup>3</sup>, Juho Rousu<sup>4</sup>, Florence d'Alché-Buc<sup>1</sup>

<sup>1</sup> LTCI, Télécom Paris, Institut Polytechnique de Paris, France

<sup>2</sup> Ecole Polytechnique, Institut Polytechnique de Paris, CMAP, France

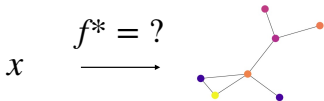
<sup>3</sup> Université de Toulouse, INRAE, UR MIAT, France

<sup>4</sup> Department of Computer Science, Aalto University, Finland

# Problem setting

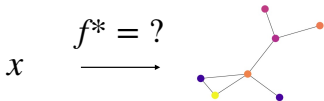
# Problem setting

**Supervised graph prediction.** Estimate  $f^* : \mathcal{X} \rightarrow \mathcal{Y}$  that predicts a graph  $y$  from input  $x$ , thanks to a data set  $(x_i, y_i)_{i=1}^N$ .

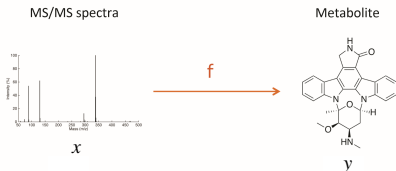


# Problem setting

**Supervised graph prediction.** Estimate  $f^* : \mathcal{X} \rightarrow \mathcal{Y}$  that predicts a graph  $y$  from input  $x$ , thanks to a data set  $(x_i, y_i)_{i=1}^N$ .



**Example (the metabolite identification problem).** The goal is to identify a molecule  $y$  from a mass spectrum  $x$ .



# Challenges and contributions

# Challenges and contributions

Challenges of graph prediction.

# Challenges and contributions

## Challenges of graph prediction.

- (Modelling). Defining a relevant way of interpolating in graph spaces.

# Challenges and contributions

## Challenges of graph prediction.

- (Modelling). Defining a relevant way of interpolating in graph spaces.
- (Computational). Defining a model whose predictions can be efficiently computed, and train it.

# Challenges and contributions

## Challenges of graph prediction.

- (Modelling). Defining a relevant way of interpolating in graph spaces.
- (Computational). Defining a model whose predictions can be efficiently computed, and train it.

## Contributions.

# Challenges and contributions

## Challenges of graph prediction.

- (Modelling). Defining a relevant way of interpolating in graph spaces.
- (Computational). Defining a model whose predictions can be efficiently computed, and train it.

## Contributions.

- We propose a **novel model for graph prediction** with two different training strategies: **kernel method or neural network**.

# Challenges and contributions

## Challenges of graph prediction.

- (Modelling). Defining a relevant way of interpolating in graph spaces.
- (Computational). Defining a model whose predictions can be efficiently computed, and train it.

## Contributions.

- We propose a **novel model for graph prediction** with two different training strategies: **kernel method or neural network**.
- We provide **theoretical guarantees** in the nonparametric case.

# Challenges and contributions

## Challenges of graph prediction.

- (Modelling). Defining a relevant way of interpolating in graph spaces.
- (Computational). Defining a model whose predictions can be efficiently computed, and train it.

## Contributions.

- We propose a **novel model for graph prediction** with two different training strategies: **kernel method or neural network**.
- We provide **theoretical guarantees** in the nonparametric case.
- We assess the method on a synthetic and a real-world problem.

# Graphs as metric measure spaces

# Graphs as metric measure spaces

**Feature space.** Each node of a graph have a label represented as a vector in  $\mathbb{R}^d$ .

$$\mathcal{F} \subset \mathbb{R}^d \quad \text{with } |\mathcal{F}| < \infty \quad (1)$$

# Graphs as metric measure spaces

**Feature space.** Each node of a graph have a label represented as a vector in  $\mathbb{R}^d$ .

$$\mathcal{F} \subset \mathbb{R}^d \quad \text{with } |\mathcal{F}| < \infty \quad (1)$$

**Output space: discrete graph space.**

$$\mathcal{Y} = \left\{ (C, F, h) \mid n \leq n_{max}, C \in \{0, 1\}^{n \times n}, C^T = C, F = (F_i)_{i=1}^n \in \mathcal{F}^n, h = \frac{1}{n} \mathbf{1}_n \right\} \quad (2)$$

# Graphs as metric measure spaces

**Feature space.** Each node of a graph have a label represented as a vector in  $\mathbb{R}^d$ .

$$\mathcal{F} \subset \mathbb{R}^d \quad \text{with } |\mathcal{F}| < \infty \quad (1)$$

**Output space: discrete graph space.**

$$\mathcal{Y} = \left\{ (C, F, h) \mid n \leq n_{max}, C \in \{0, 1\}^{n \times n}, C^T = C, F = (F_i)_{i=1}^n \in \mathcal{F}^n, h = \frac{1}{n} \mathbf{1}_n \right\} \quad (2)$$

**Prediction space: continuous relaxed graph space.**

$$\mathcal{Z}_n = \left\{ (C, F, h) \mid C \in [0, 1]^{n \times n}, C^T = C, F \in \text{Conv}(\mathcal{F})^n, h = \frac{1}{n} \mathbf{1}_n \right\} \quad (3)$$

# Defining a distance between graphs

# Defining a distance between graphs

The FGW distance (Vayer et al., 2020).  $\beta \in [0, 1]$ ,  $z_1 = (C_1, F_1)$  and  $z_2 = (C_2, F_2)$ :

$$\text{FGW}_2^2(z_1, z_2) = \min_{\pi \in \mathcal{P}_{n_1, n_2}} \sum_{i, k, j, l} \left[ (1 - \beta) \|F_1(i) - F_2(j)\|_{\mathbb{R}^d}^2 + \beta (C_1(i, k) - C_2(j, l))^2 \right] \pi_{i, j} \pi_{k, l}.$$

# Defining a distance between graphs

The FGW distance (Vayer et al., 2020).  $\beta \in [0, 1]$ ,  $z_1 = (C_1, F_1)$  and  $z_2 = (C_2, F_2)$ :

$$\text{FGW}_2^2(z_1, z_2) = \min_{\pi \in \mathcal{P}_{n_1, n_2}} \sum_{i, k, j, l} \left[ (1 - \beta) \|F_1(i) - F_2(j)\|_{\mathbb{R}^d}^2 + \beta (C_1(i, k) - C_2(j, l))^2 \right] \pi_{i, j} \pi_{k, l}.$$

Remarks.

# Defining a distance between graphs

The FGW distance (Vayer et al., 2020).  $\beta \in [0, 1]$ ,  $z_1 = (C_1, F_1)$  and  $z_2 = (C_2, F_2)$ :

$$\text{FGW}_2^2(z_1, z_2) = \min_{\pi \in \mathcal{P}_{n_1, n_2}} \sum_{i, k, j, l} \left[ (1 - \beta) \|F_1(i) - F_2(j)\|_{\mathbb{R}^d}^2 + \beta (C_1(i, k) - C_2(j, l))^2 \right] \pi_{i, j} \pi_{k, l}.$$

## Remarks.

- Two graphs are closed if there exists a transport plan matching their nodes while preserving the labels and the pairwise similarities between the nodes.

# Defining a distance between graphs

The FGW distance (Vayer et al., 2020).  $\beta \in [0, 1]$ ,  $z_1 = (C_1, F_1)$  and  $z_2 = (C_2, F_2)$ :

$$\text{FGW}_2^2(z_1, z_2) = \min_{\pi \in \mathcal{P}_{n_1, n_2}} \sum_{i, k, j, l} \left[ (1 - \beta) \|F_1(i) - F_2(j)\|_{\mathbb{R}^d}^2 + \beta (C_1(i, k) - C_2(j, l))^2 \right] \pi_{i, j} \pi_{k, l}.$$

## Remarks.

- Two graphs are closed if there exists a transport plan matching their nodes while preserving the labels and the pairwise similarities between the nodes.
- It is an extension of the GW distance on graphs to attributed graphs.

# Defining a distance between graphs

The FGW distance (Vayer et al., 2020).  $\beta \in [0, 1]$ ,  $z_1 = (C_1, F_1)$  and  $z_2 = (C_2, F_2)$ :

$$\text{FGW}_2^2(z_1, z_2) = \min_{\pi \in \mathcal{P}_{n_1, n_2}} \sum_{i, k, j, l} \left[ (1 - \beta) \|F_1(i) - F_2(j)\|_{\mathbb{R}^d}^2 + \beta (C_1(i, k) - C_2(j, l))^2 \right] \pi_{i, j} \pi_{k, l}.$$

## Remarks.

- Two graphs are closed if there exists a transport plan matching their nodes while preserving the labels and the pairwise similarities between the nodes.
- It is an extension of the GW distance on graphs to attributed graphs.
- When  $n$  is big enough  $\mathcal{Y} \underset{\text{FGW}}{\subset} \mathcal{Z}_n$ .

# Proposed method

# Proposed method

I) FGW as a loss function. We propose to estimate a minimizer  $f: \mathcal{X} \rightarrow \mathcal{Z}_n$  of

$$\mathcal{R}(f) = \mathbb{E}[\text{FGW}_2^2(f(x), y)] \quad (4)$$

# Proposed method

I) FGW as a loss function. We propose to estimate a minimizer  $f: \mathcal{X} \rightarrow \mathcal{Z}_n$  of

$$\mathcal{R}(f) = \mathbb{E}[\text{FGW}_2^2(f(x), y)] \quad (4)$$

II) Proposed FGW barycentric model. Given  $M$  template graphs  $\bar{z}_j \in \mathcal{Z}$

$$f_\theta(x) = \arg \min_{z \in \mathcal{Z}_n} \sum_{j=1}^M \alpha_j(x; W) \text{FGW}_2^2(z, \bar{z}_j) \quad (5)$$

# Proposed method

I) FGW as a loss function. We propose to estimate a minimizer  $f: \mathcal{X} \rightarrow \mathcal{Z}_n$  of

$$\mathcal{R}(f) = \mathbb{E}[\text{FGW}_2^2(f(x), y)] \quad (4)$$

II) Proposed FGW barycentric model. Given  $M$  template graphs  $\bar{z}_j \in \mathcal{Z}$

$$f_\theta(x) = \arg \min_{z \in \mathcal{Z}_n} \sum_{j=1}^M \alpha_j(x; W) \text{FGW}_2^2(z, \bar{z}_j) \quad (5)$$

**Remark.** The model's parameters are  $W$  and the templates  $(\bar{z}_j)_{j=1}^M$ .

# Two fitting strategies

# Two fitting strategies

A) **Kernel method.** Given a p. d. kernel  $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ ,  $M = N$  and  $\bar{z}_j = z_j$

$$\alpha(x) = (K + \lambda I_N)^{-1} k_x \quad (6)$$

with  $K = (k(x_i, x_j))_{ij} \in \mathbb{R}^{N \times N}$  and  $k_x^T = (k(x, x_1), \dots, k(x, x_N))$ .

# Two fitting strategies

A) **Kernel method.** Given a p. d. kernel  $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ ,  $M = N$  and  $\bar{z}_j = z_j$

$$\alpha(x) = (K + \lambda I_N)^{-1} k_x \quad (6)$$

with  $K = (k(x_i, x_j))_{ij} \in \mathbb{R}^{N \times N}$  and  $k_x^T = (k(x, x_1), \dots, k(x, x_N))$ .

B) **Neural network.**

- $\alpha : \mathcal{X} \rightarrow \mathbb{R}^M$  is a neural network.
- $\alpha$  and the template graphs  $(\bar{z}_j)_{j=1}^M$  are learned using **stochastic gradient descent**.
- We propose a method to compute a sub-gradient of the loss  $\text{FGW}(f_\theta(x_i), y_i)$ .

# Theoretical guarantees

# Theoretical guarantees

Under technical assumptions, the two following guarantees hold for the kernel-based estimator.

# Theoretical guarantees

Under technical assumptions, the two following guarantees hold for the kernel-based estimator.

**Consistency.** With probability 1,

$$\lim_{N \rightarrow +\infty} \mathcal{R}(\hat{f}) = \mathcal{R}(f^*). \quad (7)$$

# Theoretical guarantees

Under technical assumptions, the two following guarantees hold for the kernel-based estimator.

**Consistency.** With probability 1,

$$\lim_{N \rightarrow +\infty} \mathcal{R}(\hat{f}) = \mathcal{R}(f^*). \quad (7)$$

**Excess-risk bound.** With probability  $1 - \delta$ ,

$$\mathcal{R}(\hat{f}) - \mathcal{R}(f^*) \leq c \log(4/\delta) N^{-1/4}, \quad (8)$$

with  $c$  a constant independent of  $N$  and  $\delta$ .

# Theoretical guarantees

Under technical assumptions, the two following guarantees hold for the kernel-based estimator.

**Consistency.** With probability 1,

$$\lim_{N \rightarrow +\infty} \mathcal{R}(\hat{f}) = \mathcal{R}(f^*). \quad (7)$$

**Excess-risk bound.** With probability  $1 - \delta$ ,

$$\mathcal{R}(\hat{f}) - \mathcal{R}(f^*) \leq c \log(4/\delta) N^{-1/4}, \quad (8)$$

with  $c$  a constant independent of  $N$  and  $\delta$ .

**Remark.** These results are based on the Implicit Loss Embedding framework (Ciliberto et al., 2020).

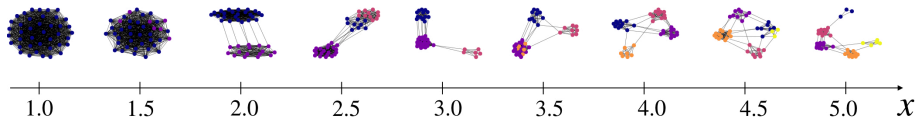
# Synthetic experiment with neural network

# Synthetic experiment with neural network

**True map.** We defined a smooth map  $f^* : [1, 5] \rightarrow \mathcal{Y}$  which maps  $x$  to a stochastic block model with  $x$  blocks.

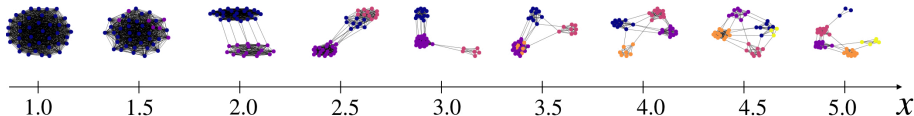
# Synthetic experiment with neural network

**True map.** We defined a smooth map  $f^* : [1, 5] \rightarrow \mathcal{Y}$  which maps  $x$  to a stochastic block model with  $x$  blocks.



# Synthetic experiment with neural network

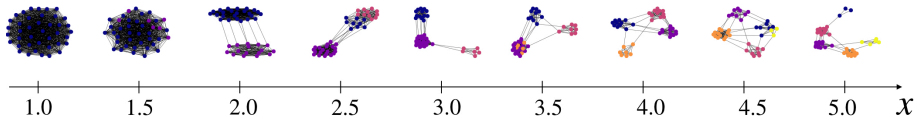
**True map.** We defined a smooth map  $f^* : [1, 5] \rightarrow \mathcal{Y}$  which maps  $x$  to a stochastic block model with  $x$  blocks.



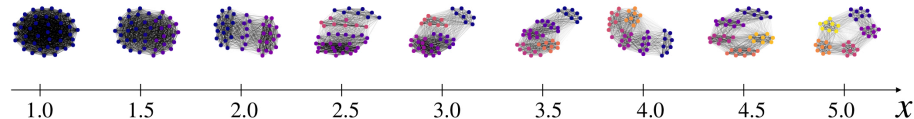
**Learned map.** We trained the neural network model with 8 templates using 100 i.i.d couples  $(x_i, y_i)_{i=1}^{100}$ . We obtained the following estimated map  $\hat{f} : [1, 5] \rightarrow \mathcal{Y}$

# Synthetic experiment with neural network

**True map.** We defined a smooth map  $f^* : [1, 5] \rightarrow \mathcal{Y}$  which maps  $x$  to a stochastic block model with  $x$  blocks.



**Learned map.** We trained the neural network model with 8 templates using 100 i.i.d couples  $(x_i, y_i)_{i=1}^{100}$ . We obtained the following estimated map  $\hat{f} : [1, 5] \rightarrow \mathcal{Y}$

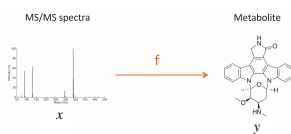


# The metabolite identification problem

Kernel-based graph prediction estimator.

$$\hat{f}(x) = \arg \min_{y \in \mathcal{Y}(x)} \sum_{i=1}^N \alpha_i(x) D(y, y_i) \quad (9)$$

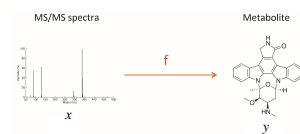
# The metabolite identification problem



Kernel-based graph prediction estimator.

$$\hat{f}(x) = \arg \min_{y \in \mathcal{Y}(x)} \sum_{i=1}^N \alpha_i(x) D(y, y_i) \quad (9)$$

# The metabolite identification problem

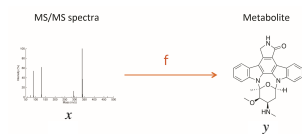


Kernel-based graph prediction estimator.

$$\hat{f}(x) = \arg \min_{y \in \mathcal{Y}(x)} \sum_{i=1}^N \alpha_i(x) D(y, y_i) \quad (9)$$

**Experimental setting.** Comparison of various graph metrics  $D$  in terms of test Top-k accuracies.

# The metabolite identification problem



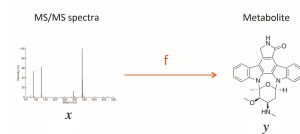
Kernel-based graph prediction estimator.

$$\hat{f}(x) = \arg \min_{y \in \mathcal{Y}(x)} \sum_{i=1}^N \alpha_i(x) D(y, y_i) \quad (9)$$

**Experimental setting.** Comparison of various graph metrics  $D$  in terms of test Top-k accuracies.

- PPK as input kernel.

# The metabolite identification problem



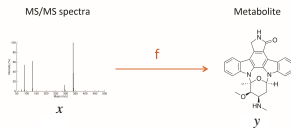
Kernel-based graph prediction estimator.

$$\hat{f}(x) = \arg \min_{y \in \mathcal{Y}(x)} \sum_{i=1}^N \alpha_i(x) D(y, y_i) \quad (9)$$

**Experimental setting.** Comparison of various graph metrics  $D$  in terms of test Top-k accuracies.

- PPK as input kernel.
- Used given candidate set  $\mathcal{Y}(x)$  for the computation of the barycenter.

# The metabolite identification problem



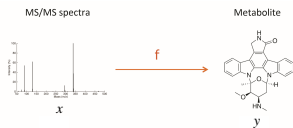
Kernel-based graph prediction estimator.

$$\hat{f}(x) = \arg \min_{y \in \mathcal{Y}(x)} \sum_{i=1}^N \alpha_i(x) D(y, y_i) \quad (9)$$

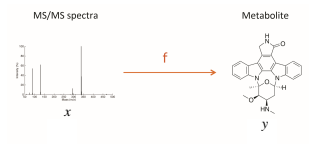
**Experimental setting.** Comparison of various graph metrics  $D$  in terms of test Top-k accuracies.

- PPK as input kernel.
- Used given candidate set  $\mathcal{Y}(x)$  for the computation of the barycenter.
- Compared graph metrics  $D$ : FGW with various label distances, Fingerprints, Gaussian Fingerprints (Brouard et al., 2016), deep graph representations MoFlow (Zang et al, 2020), Weisfeiler-Lehman distance.

# Overview of the obtained results

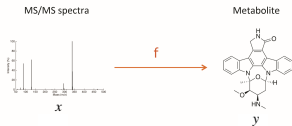


# Overview of the obtained results



Experimental results for the metabolite identification problem.

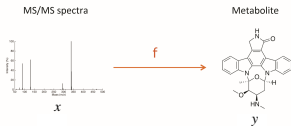
# Overview of the obtained results



## Experimental results for the metabolite identification problem.

- **Adaptability.** Few engineering on the distance between labels (atoms) allows to improve the accuracies.

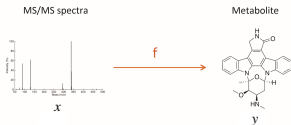
# Overview of the obtained results



## Experimental results for the metabolite identification problem.

- **Adaptability.** Few engineering on the distance between labels (atoms) allows to improve the accuracies.
- **Better than all generic graph distance tested.** MoFlow representations and Weisfeiler-Lehman distance.

# Overview of the obtained results



## Experimental results for the metabolite identification problem.

- **Adaptability.** Few engineering on the distance between labels (atoms) allows to improve the accuracies.
- **Better than all generic graph distance tested.** MoFlow representations and Weisfeiler-Lehman distance.
- **Fingerprint representations remains SOTA.**

# Conclusion

# Conclusion

Summary.

# Conclusion

## Summary.

- Novel method for graph prediction leveraging computational optimal transport tools.

# Conclusion

## Summary.

- Novel method for graph prediction leveraging computational optimal transport tools.
- Theoretical guarantees in the kernel-based case.

# Conclusion

## Summary.

- Novel method for graph prediction leveraging computational optimal transport tools.
- Theoretical guarantees in the kernel-based case.
- Promising experimental results on a synthetic and a real-world graph prediction problem.

# Conclusion

## Summary.

- Novel method for graph prediction leveraging computational optimal transport tools.
- Theoretical guarantees in the kernel-based case.
- Promising experimental results on a synthetic and a real-world graph prediction problem.

Thank you for your attention!