



## Optimal transport for graph data

Barycenters and dictionary learning

---

**R. Flamary** - CMAP, École Polytechnique, Institut Polytechnique de Paris

November 10 2021

Journée NormaSTIC, Lisieux

# Collaborators



N. Courty



A. Rakotomamonjy



D. Tuia



A. Habrard



M. Perrot



M. Ducoffe



M. Cuturi



K. Lounici



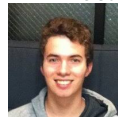
A. Férrari



C. Févotte



V. Emiya



V. Seguy



B. Damodaran



T. Vayer



L. Chapel



R. Tavenard



K. Fatras



I. Redko



H. Janati



T. Séjourné



H. Tran



G. Gasso



M. Corneli



C. Vincent-Cuaz

## **Optimal Transport and Gromov-Wasserstein**

Discrete Optimal Transport (OT)

Gromov-Wasserstein divergence

Applications of Gromov Wasserstein

## **Fused Gromov-Wasserstein**

Labeled graphs as distributions

Fused Gromov-Wasserstein distance

Applications on graphs

## **Online Graph Dictionary Learning**

Linear modeling and unmixing of graphs

Learning a dictionary of graphs

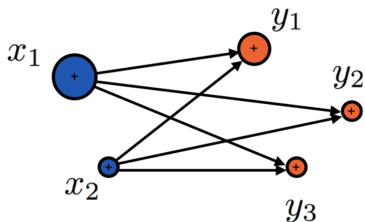
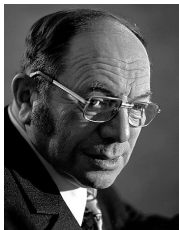
Numerical experiments

# Optimal Transport and Gromov-Wasserstein

---

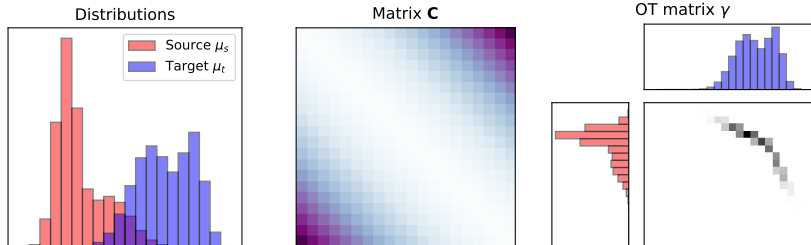


# Optimal transport



- Problem introduced by Gaspard Monge in his memoire [Monge, 1781].
- How to move mass while minimizing a cost (mass + cost)
- Monge formulation seeks for a mapping between two mass distribution.
- Reformulated by Leonid Kantorovich (1912–1986), Economy nobelist in 1975
- Focus on where the mass goes, allow splitting [Kantorovich, 1942].
- Applications mainly for resource allocation problems

# Optimal transport between discrete distributions



## Kantorovich formulation : OT Linear Program

When  $\mu_s = \sum_{i=1}^{n_s} a_i \delta_{\mathbf{x}_i^s}$  and  $\mu_t = \sum_{i=1}^{n_t} b_i \delta_{\mathbf{x}_i^t}$

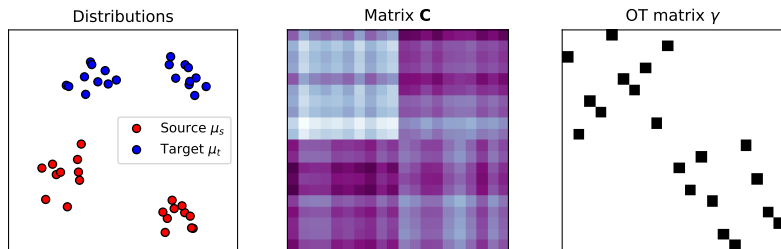
$$W_p^p(\mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \left\{ \langle \mathbf{T}, \mathbf{C} \rangle_F = \sum_{i,j} T_{i,j} c_{i,j} \right\}$$

where  $\mathbf{C}$  is a cost matrix with  $c_{i,j} = c(\mathbf{x}_i^s, \mathbf{x}_j^t) = \|\mathbf{x}_i^s - \mathbf{x}_j^t\|^p$  and the constraints are

$$\Pi(\mu_s, \mu_t) = \left\{ \mathbf{T} \in (\mathbb{R}^+)^{n_s \times n_t} \mid \mathbf{T} \mathbf{1}_{n_t} = \mathbf{a}, \mathbf{T}^T \mathbf{1}_{n_s} = \mathbf{b} \right\}$$

- Linear program with  $n_s n_t$  variables and  $n_s + n_t$  constraints.
- Solving the OT problem with network simplex is  $O(n^3 \log(n))$  for  $n = n_s = n_t$ .
- $W_p(\mu_s, \mu_t)$  is called the Wasserstein distance (EMD for  $p = 1$ ).

# Optimal transport between discrete distributions



## Kantorovich formulation : OT Linear Program

When  $\mu_s = \sum_{i=1}^{n_s} a_i \delta_{\mathbf{x}_i^s}$  and  $\mu_t = \sum_{i=1}^{n_t} b_i \delta_{\mathbf{x}_i^t}$

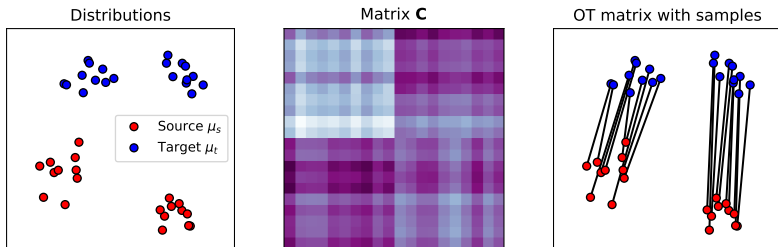
$$W_p^p(\mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \left\{ \langle \mathbf{T}, \mathbf{C} \rangle_F = \sum_{i,j} T_{i,j} c_{i,j} \right\}$$

where  $\mathbf{C}$  is a cost matrix with  $c_{i,j} = c(\mathbf{x}_i^s, \mathbf{x}_j^t) = \|\mathbf{x}_i^s - \mathbf{x}_j^t\|^p$  and the constraints are

$$\Pi(\mu_s, \mu_t) = \left\{ \mathbf{T} \in (\mathbb{R}^+)^{n_s \times n_t} \mid \mathbf{T} \mathbf{1}_{n_t} = \mathbf{a}, \mathbf{T}^T \mathbf{1}_{n_s} = \mathbf{b} \right\}$$

- Linear program with  $n_s n_t$  variables and  $n_s + n_t$  constraints.
- Solving the OT problem with network simplex is  $O(n^3 \log(n))$  for  $n = n_s = n_t$ .
- $W_p(\mu_s, \mu_t)$  is called the Wasserstein distance (EMD for  $p = 1$ ).

# Optimal transport between discrete distributions



## Kantorovich formulation : OT Linear Program

When  $\mu_s = \sum_{i=1}^{n_s} a_i \delta_{\mathbf{x}_i^s}$  and  $\mu_t = \sum_{i=1}^{n_t} b_i \delta_{\mathbf{x}_i^t}$

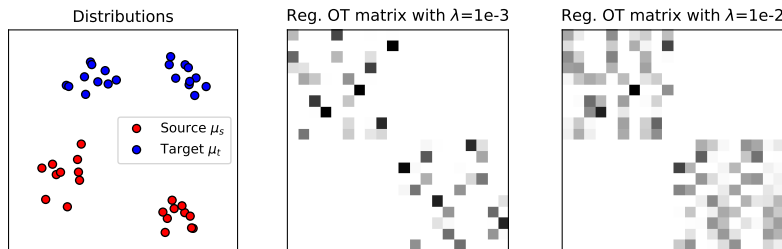
$$W_p^p(\mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \left\{ \langle \mathbf{T}, \mathbf{C} \rangle_F = \sum_{i,j} T_{i,j} c_{i,j} \right\}$$

where  $\mathbf{C}$  is a cost matrix with  $c_{i,j} = c(\mathbf{x}_i^s, \mathbf{x}_j^t) = \|\mathbf{x}_i^s - \mathbf{x}_j^t\|^p$  and the constraints are

$$\Pi(\mu_s, \mu_t) = \left\{ \mathbf{T} \in (\mathbb{R}^+)^{n_s \times n_t} \mid \mathbf{T} \mathbf{1}_{n_t} = \mathbf{a}, \mathbf{T}^T \mathbf{1}_{n_s} = \mathbf{b} \right\}$$

- Linear program with  $n_s n_t$  variables and  $n_s + n_t$  constraints.
- Solving the OT problem with network simplex is  $O(n^3 \log(n))$  for  $n = n_s = n_t$ .
- $W_p(\mu_s, \mu_t)$  is called the Wasserstein distance (EMD for  $p = 1$ ).

# Entropic regularized optimal transport

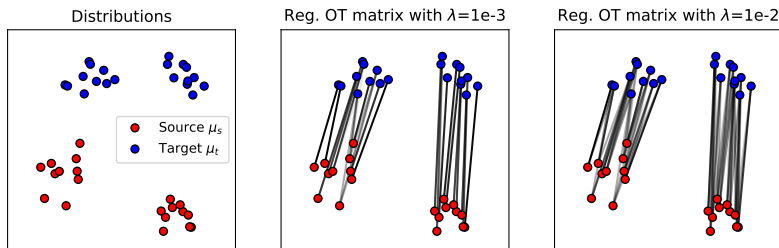


## Entropic regularization [Cuturi, 2013]

$$W_\epsilon(\mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \langle \mathbf{T}, \mathbf{C} \rangle_F + \epsilon \sum_{i,j} T_{i,j} \log T_{i,j}$$

- Regularization with the negative entropy  $-H(\mathbf{T})$ .
- Looses sparsity, but strictly convex optimization problem [Benamou et al., 2015].
- Can be solved with the very efficient Sinkhorn-Knopp matrix scaling algorithm.
- Loss and OT matrix are differentiable and have better statistical properties [Genevay et al., 2018].
- Classical OT needs distributions lying in the same space  $\rightarrow$  Gromov-Wasserstein.

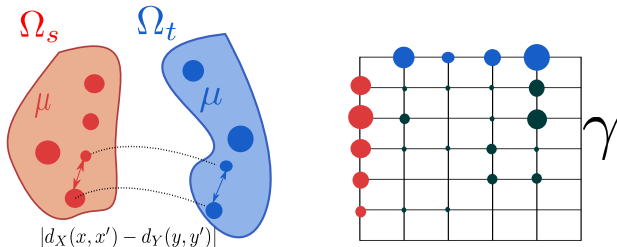
# Entropic regularized optimal transport



## Entropic regularization [Cuturi, 2013]

$$W_\epsilon(\mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \langle \mathbf{T}, \mathbf{C} \rangle_F + \epsilon \sum_{i,j} T_{i,j} \log T_{i,j}$$

- Regularization with the negative entropy  $-H(\mathbf{T})$ .
- Loses sparsity, but strictly convex optimization problem [Benamou et al., 2015].
- Can be solved with the very efficient Sinkhorn-Knopp matrix scaling algorithm.
- Loss and OT matrix are differentiable and have better statistical properties [Genevay et al., 2018].
- Classical OT needs distributions lying in the same space  $\rightarrow$  Gromov-Wasserstein.



Inspired from Gabriel Peyré

## GW for discrete distributions [Memoli, 2011]

$$\mathcal{GW}_p(\mu_S, \mu_T) = \left( \min_{T \in \Pi(\mu_S, \mu_T)} \sum_{i,j,k,l} |D_{i,k} - D'_{j,l}|^p T_{i,j} T_{k,l} \right)^{\frac{1}{p}}$$

with  $\mu_S = \sum_i a_i \delta_{\mathbf{x}_i^S}$  and  $\mu_T = \sum_j b_j \delta_{\mathbf{x}_j^T}$  and  $D_{i,k} = \|\mathbf{x}_i^S - \mathbf{x}_k^S\|$ ,  $D'_{j,l} = \|\mathbf{x}_j^T - \mathbf{x}_l^T\|$

- Distance between metric measured spaces : across different spaces.
- Search for an OT plan that preserve the pairwise relationships between samples.
- Invariant to isometry in either spaces (e.g. rotations and translation).

$$\mathcal{GW}_p^p(\mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \sum_{i,j,k,l} |D_{i,k} - D'_{j,l}|^p T_{i,j} T_{k,l}$$

with  $\mu_s = \sum_i a_i \delta_{\mathbf{x}_i^s}$  and  $\mu_t = \sum_j b_j \delta_{\mathbf{x}_j^t}$  and  $D_{i,k} = \|\mathbf{x}_i^s - \mathbf{x}_k^s\|$ ,  $D'_{j,l} = \|\mathbf{x}_j^t - \mathbf{x}_l^t\|$

## Optimization problem

- Quadratic Program (Wasserstein is a linear program).
- Nonconvex, NP-hard, related to Quadratic Assignment Problem (QAP).

## Optimization algorithm

- Large problem and non convexity forbid standard QP solvers.
- Local solution can be obtained with conditional gradient (Frank-Wolfe) [Vayer et al., 2018] (each iteration is an OT problems).
- Gromov in 1D has a close form (solved in discrete with a sort) [Vayer et al., 2019].
- Can be regularized by entropy similarly to classical OT.



## Optimization Problem [Peyré et al., 2016]

$$\mathcal{GW}_{p,\epsilon}^p(\mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \sum_{i,j,k,l} |D_{i,k} - D'_{j,l}|^p T_{i,j} T_{k,l} + \epsilon \sum_{i,j} T_{i,j} \log T_{i,j} \quad (1)$$

with  $\mu_s = \sum_i a_i \delta_{\mathbf{x}_i^s}$  and  $\mu_t = \sum_j b_j \delta_{\mathbf{x}_j^t}$  and  $D_{i,k} = \|\mathbf{x}_i^s - \mathbf{x}_k^s\|$ ,  $D'_{j,l} = \|\mathbf{x}_j^t - \mathbf{x}_l^t\|$

- Smoothing the original GW with a convex and smooth entropic term.

## Solving the entropic GW [Peyré et al., 2016]

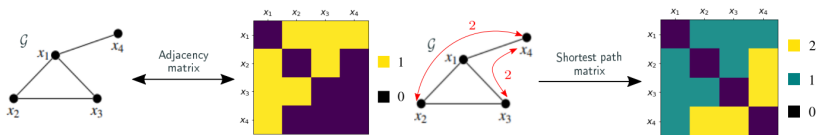
- Problem (1) can be solved using a KL mirror descent.
- This is equivalent to solving at each iteration  $t$

$$\mathbf{T}^{(t+1)} = \min_{\mathbf{T} \in \mathcal{P}} \left\langle \mathbf{T}, \mathbf{G}^{(t)} \right\rangle_F + \epsilon \sum_{i,j} T_{i,j} \log T_{i,j}$$

Where  $G_{i,j}^{(t)} = 2 \sum_{k,l} |D_{i,k} - D'_{j,l}|^p T_{k,l}^{(t)}$  is the gradient of the GW loss at previous point  $\mathbf{T}^{(k)}$ .

- Problem above can be solved using a Sinkhorn-Knopp algorithm of entropic OT.
- Very fast approximation exist for low rank distances [Scetbon et al., 2021].

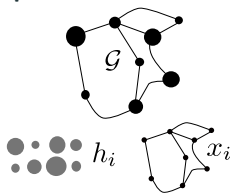
# Gromov-Wasserstein between graphs



## Modeling the graph structure with a pairwise matrix $D$

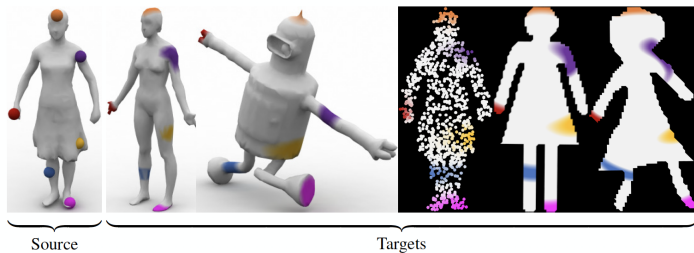
- An undirected graph  $\mathcal{G} := (\mathbf{V}, \mathbf{E})$  is defined by  $\mathbf{V} = \{\mathbf{x}_i\}_{i \in [\mathbf{N}]}$  set of the  $\mathbf{N}$  nodes and  $\mathbf{E} = \{(\mathbf{x}_i, \mathbf{x}_j) | \mathbf{x}_i \leftrightarrow \mathbf{x}_j\}$  set of edges.
- Structure represented as a symmetric matrix  $D$  of relations between the nodes.
- Possible choices : **Adjacency matrix** (used in this study), Laplacian matrix, Shortest path matrix.

## Graph as a distribution $(D, h)$

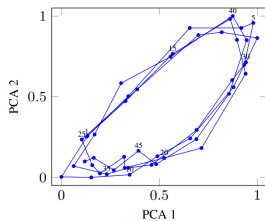
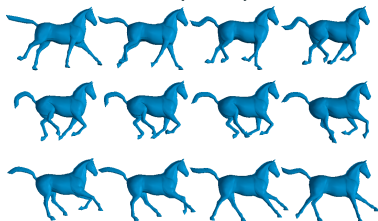


- Graph represented as a discrete distribution  $\mu_X = \sum_i h_i \delta_{x_i}$ .
- The positions  $x_i$  are implicit and represented as the pairwise matrix  $D$ .
- $h_i$  are the masses on the nodes of the graphs (uniform by default).

## Shape matching between 3D and 2D surfaces

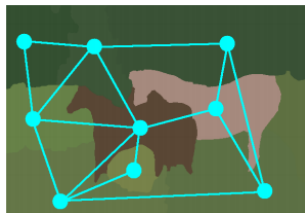


## Multidimensional scaling (MDS) of shape collection



## Fused Gromov-Wasserstein

---

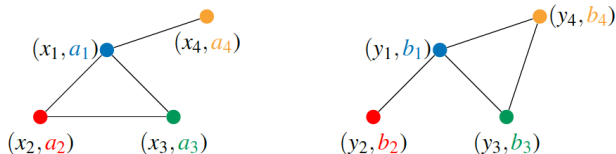


## Structured data

- Some structured data can be viewed as a combination of features informations linked within each other by some structural information.
- Can be seen as a distribution on a joint feature/structure space.
- Example : labeled graph.

## Meaningful distances on labeled structured data

- Us both features (labels) and structure (graph).
- Allows for comparison, classification.
- Data science (statistics, means, concentration).



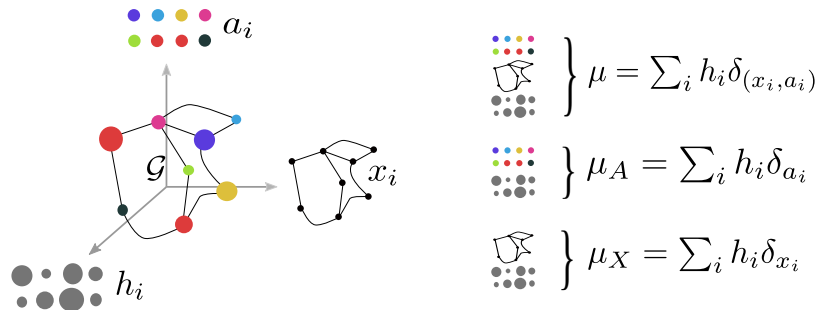
### Structured data

- Some structured data can be viewed as a combination of features informations linked within each other by some structural information.
- Can be seen as a distribution on a joint feature/structure space.
- Example : labeled graph.

### Meaningful distances on labeled structured data

- Us both features (labels) and structure (graph).
- Allows for comparison, classification.
- Data science (statistics, means, concentration).

# Structured data as distributions

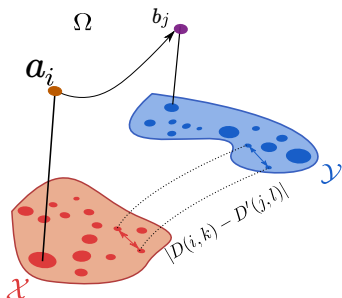


## Graph data representation

$$\mu = \sum_{i=1}^n h_i \delta_{(x_i a_i)}$$

- Nodes are weighted by their mass  $h_i$ .
- But no common metric between the structure points  $x_i$  of two different graphs.
- Features values  $a_i$  can be compared through the common metric

# Fused Gromov-Wasserstein distance



## Fused Gromov Wasserstein distance

$$\mu_s = \sum_{i=1}^n h_i \delta_{x_i, a_i} \text{ and } \mu_t = \sum_{j=1}^m g_j \delta_{y_j, b_j}$$

$$\mathcal{FGW}_{p,q,\alpha}(D, D', \mu_s, \mu_t) = \left( \min_{T \in \Pi(\mu_s, \mu_t)} \sum_{i,j,k,l} ((1-\alpha)C_{i,j}^q + \alpha |D_{i,k} - D'_{j,l}|^q)^p T_{i,j} T_{k,l} \right)^{\frac{1}{p}}$$

with  $D_{i,k} = \|x_i - x_k\|$  and  $D'_{j,l} = \|y_j - y_l\|$  and  $C_{i,j} = \|a_i - b_j\|$

- Parameters  $q > 1, \forall p \geq 1$ .
- $\alpha \in [0, 1]$  is a trade off parameter between structure and features.



$$\mathcal{FGW}_{p,q,\alpha}^p(D, D', \mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \sum_{i,j,k,l} ((1 - \alpha)C_{i,j}^q + \alpha|D_{i,k} - D'_{j,l}|^q)^p T_{i,j} T_{k,l}$$

## Metric properties [Vayer et al., 2020]

- $\mathcal{FGW}$  defines a metric over structured data with **measure and features preserving isometries** as invariants.
- $\mathcal{FGW}$  is a metric for  $q = 1$  a semi metric for  $q > 1$ ,  $\forall p \geq 1$ .
- The distance is nul *iff* :
  - There exists a Monge map  $T \# \mu_s = \mu_t$ .
  - Structures are equivalent through this Monge map (isometry).
  - Features are equal through this Monge map.

## Other properties for continuous distributions

- Interpolation between  $\mathcal{W}$  ( $\alpha = 0$ ) and  $\mathcal{GW}$  ( $\alpha = 1$ ) distances.
- Geodesic properties (constant speed, unicity).

$$\mathcal{FGW}_{p,q,\alpha}(D, D', \mu_s, \mu_t) = \left( \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \sum_{i,j,k,l} ((1-\alpha)C_{i,j}^q + \alpha|D_{i,k} - D'_{j,l}|^q)^p T_{i,j} T_{k,l} \right)^{\frac{1}{p}}$$

### Bounds and convergence to finite samples [Vayer et al., 2020]

- The following inequalities hold:

$$\mathcal{FGW}(\mu_s, \mu_t) \geq (1 - \alpha)\mathcal{W}(\mu_A, \mu_B)^q$$

$$\mathcal{FGW}(\mu_s, \mu_t) \geq \alpha\mathcal{GW}(\mu_X, \mu_Y)^q$$

- Bound when  $\mathcal{X} = \mathcal{Y}$ :

$$\mathcal{FGW}(\mu_s, \mu_t)^p \leq 2\mathcal{W}(\mu_s, \mu_t)^p$$

- Convergence of finite samples when  $\mathcal{X} = \mathcal{Y}$  with  $d = \text{Dim}(\mathcal{X}) + \text{Dim}(\Omega)$  :

$$\mathbb{E}[\mathcal{FGW}(\mu, \mu_n)] = O\left(n^{-\frac{1}{d}}\right)$$

# Application of FGW distance on structured data classification

VECTOR ATTRIBUTES	AIDS	BZR	COX2	CUNEIFORM	ENZYMES	PROTEIN	SYNTHETIC
FGW SP	99.44+/-0.47	85.12+/-4.15	77.23+/-4.86	<b>76.67+/-7.04</b>	71.00+/-6.76	74.55+/-2.74	<b>100.00+/-0.00</b>
FGW SP REGUL	-	<b>85.61+/-5.05</b>	77.66+/-4.17	-	70.17+/-6.81	74.64+/-2.99	-
FGW WSP	99.55+/-0.35	84.88+/-4.34	78.09+/-3.81	-	69.50+/-7.30	75.09+/-2.34	-
FGWDMM SP	-	84.39+/-5.48	76.81+/-4.30	-	61.67+/-7.19	75.00+/-2.59	-
FGWDMM WSP	-	83.17+/-5.05	78.30+/-3.53	-	59.17+/-6.55	75.09+/-3.03	-
HOPPER ALL CV	99.50+/-0.59	84.15+/-5.26	<b>79.57+/-3.46</b>	32.59+/-8.73	45.33+/-4.00	71.96+/-3.22	90.67+/-4.67
PROPA ALL CV	98.45+/-1.06	79.51+/-5.02	77.66+/-3.95	12.59+/-6.67	<b>71.67+/-5.63</b>	61.34+/-4.38	64.67+/-6.70
PSCN K=10	99.80+/-0.24	80.00+/-4.47	71.70+/-3.57	25.19+/-7.73	26.67+/-4.77	67.95+/-11.28	<b>100.00+/-0.00</b>
PSCN K=5	<b>99.85+/-0.23</b>	82.20+/-4.23	71.91+/-3.40	24.81+/-7.23	27.33+/-4.16	71.79+/-3.39	<b>100.00+/-0.00</b>

## Graph classification

- Classification accuracy on classical graph datasets.
- Comparison with state-of-the-art graph kernel approaches and Graph CNN.
- We use  $\exp(-\gamma \mathcal{FGW})$  as a non-positive kernel for an SVM [Loosli et al., 2015] (FGW).
- Train Wassertsein Distance Measure Machine [Rakotomamonjy et al., 2018] (FGWDMM).

# Application of FGW distance on structured data classification

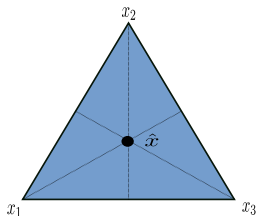
DISCRETE ATTRIBUTES	MUTAG	NC11	PTC
FGW RAW SP	83.26+/-10.30	72.82+/-1.46	55.71+/-6.74
FGW WL H=2 SP	86.42+/-7.81	85.82+/-1.16	63.20+/-7.68
FGW WL H=2 SP REGUL	84.74+/-8.03	-	63.37+/-6.75
FGW WL H=4 SP	<b>88.42+/-5.67</b>	<b>86.42 +/- 1.63</b>	<b>65.31+/-7.90</b>
FGW WL H=4 SP REGUL	86.42+/-8.81	-	63.83+/-7.83
GK K=3	82.42+/-8.40	60.78+/-2.48	56.46+/-8.03
PSCN K=10	83.47+/-10.26	70.65+/-2.58	58.34+/-7.71
PSCN K=5	83.05+/-10.80	69.85+/-1.79	55.37+/-8.28
RW ALL CV	79.47+/-8.17	58.63+/-2.44	55.09+/-7.34
SP ALL CV	82.95+/-8.19	74.26+/-1.53	-
WL ALL CV	86.21+/-8.48	85.77+/-1.07	62.86+/-7.23
WL H=2	86.21+/-8.15	81.85+/-2.28	61.60+/-8.14
WL H=4	83.68+/-9.13	85.13+/-1.61	62.17+/-7.80

WITHOUT ATTRIBUTE	IMDB-B	IMDB-M
FGW RAW SP	<b>63.80+/-3.49</b>	<b>48.00+/-3.22</b>
GK K=3	56.00+/-3.61	41.13+/-4.68
SP ALL CV	55.80+/-2.93	38.93+/-5.12

## Graph classification

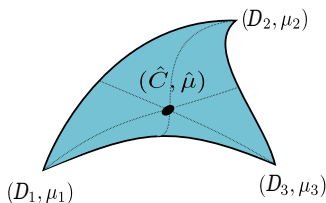
- Classification accuracy on classical graph datasets.
- Comparison with state-of-the-art graph kernel approaches and Graph CNN.
- We use  $\exp(-\gamma \mathcal{FGW})$  as a non-positive kernel for an SVM [Loosli et al., 2015] (FGW).
- Train Wassertsein Distance Measure Machine [Rakotomamonjy et al., 2018] (FGWDMM).

Euclidean barycenter



$$\min_x \sum_k \lambda_k \|x - x_k\|^2$$

FGW barycenter



$$\min_{D \in \mathbb{R}^{n \times n}, \mu} \sum_i \lambda_i \mathcal{FGW}(D_i, D, \mu_i, \mu)$$

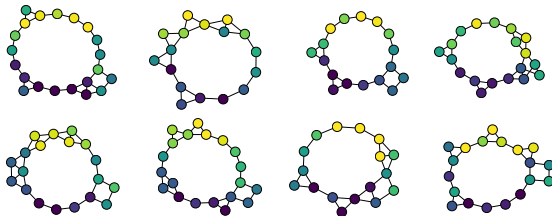
## FGW barycenter $p = 1, q = 2$

- Estimate FGW barycenter using Frechet means (similar to [Peyré et al., 2016]).
- Barycenter optimization solved via block coordinate descent (on  $\mathbf{T}, D, \{a_i\}_i$ ).
- Can chose to fix the structure ( $D$ ) or the features  $\{a_i\}_i$  in the barycenter.
- $a_{ii}$ , and  $D$  updates are weighted averages using  $\mathbf{T}$ .

Noiseless graph



Noisy graphs samples



## Barycenter of noisy graphs

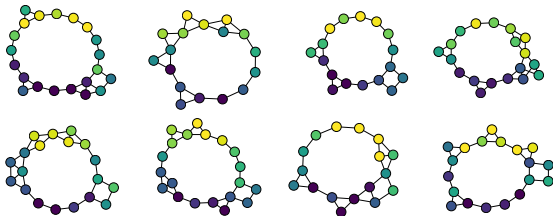
- We select a clean graph, change the number of nodes and add label noise and random connections.
- We compute the barycenter on  $n = 15$  and  $n = 7$  nodes.
- Barycenter graph is obtained through thresholding of the  $D$  matrix.

# FGW barycenter on labeled graphs

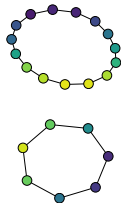
Noiseless graph



Noisy graphs samples



Barycenter



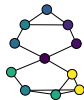
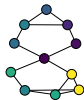
## Barycenter of noisy graphs

- We select a clean graph, change the number of nodes and add label noise and random connections.
- We compute the barycenter on  $n = 15$  and  $n = 7$  nodes.
- Barycenter graph is obtained through thresholding of the  $D$  matrix.

Noiseless graph



Noisy graphs samples



## Barycenter of noisy graphs

- We select a clean graph, change the number of nodes and add label noise and random connections.
- We compute the barycenter on  $n = 15$  and  $n = 7$  nodes.
- Barycenter graph is obtained through thresholding of the  $D$  matrix.

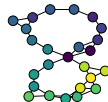


# FGW barycenter on labeled graphs

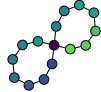
Noiseless graph



Noisy graphs samples



Barycenter

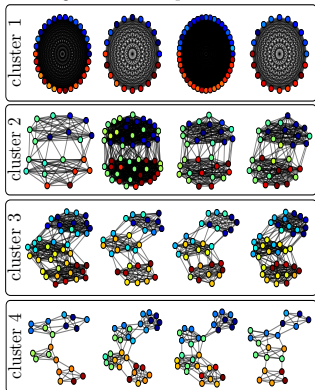


## Barycenter of noisy graphs

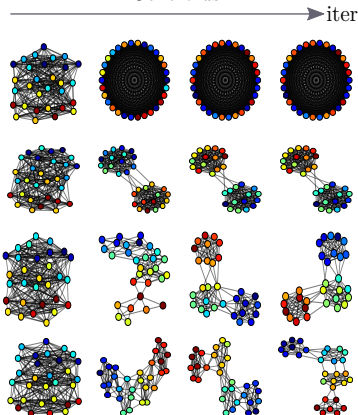
- We select a clean graph, change the number of nodes and add label noise and random connections.
- We compute the barycenter on  $n = 15$  and  $n = 7$  nodes.
- Barycenter graph is obtained through thresholding of the  $D$  matrix.

# FGW for graphs based clustering

Training dataset examples



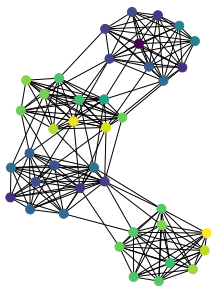
Centroids



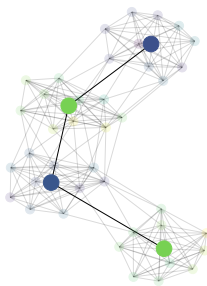
- Clustering of multiple real-valued graphs. Dataset composed of 40 graphs (10 graphs  $\times$  4 types of communities)
- $k$ -means clustering using the  $FGW$  barycenter

# FGW barycenter for community clustering

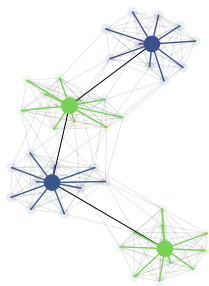
Graph with communities



Approximate Graph



Clustering with transport matrix



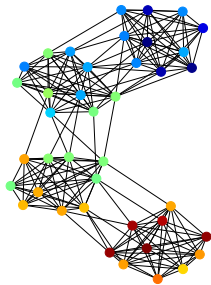
## Graph approximation and community clustering

$$\min_{\mathbf{D}, \mu} \mathcal{FGW}(\mathbf{D}, \mathbf{D}_0, \mu, \mu_0)$$

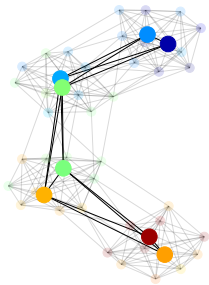
- Approximate the graph  $(\mathbf{D}_0, \mu_0)$  with a small number of nodes.
- OT matrix give the clustering affectation.
- Works for single and multiple modes in the clusters.

# FGW barycenter for community clustering

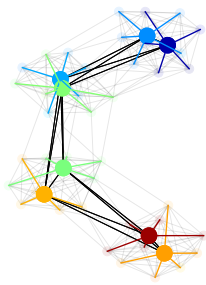
Graph with bimodal communities



Approximate Graph



Clustering with transport matrix

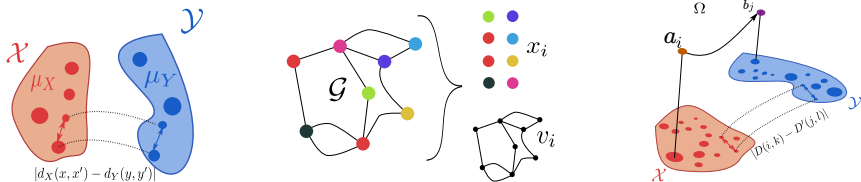


## Graph approximation and community clustering

$$\min_{\mathbf{D}, \mu} \mathcal{FGW}(\mathbf{D}, \mathbf{D}_0, \mu, \mu_0)$$

- Approximate the graph  $(\mathbf{D}_0, \mu_0)$  with a small number of nodes.
- OT matrix give the clustering affectation.
- Works for single and multiple modes in the clusters.

# GW and FGW for graph modeling



## Gromov-Wasserstein distance [Memoli, 2011]

- Divergence between distributions across metric spaces.
- Can be used to measure similarity between graphs seen as distribution their pairwise node relationship.

## Fused Gromov-Wasserstein distance [Vayer et al., 2018]

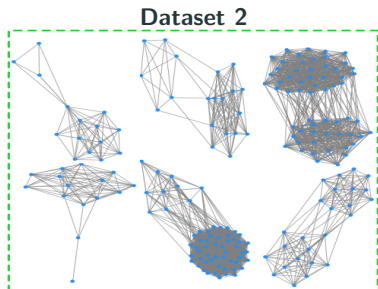
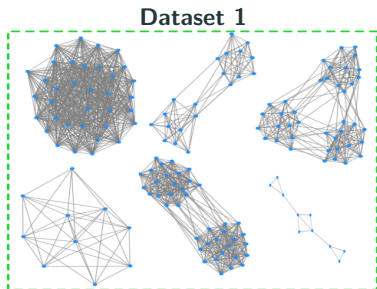
- Model labeled structured data as joint structure/labels distributions.
- New versatile method for comparing structured data based on Optimal Transport
- New notion of barycenter of structured data such as graphs or time series

How to use GW/FGW to model data variability in a dataset of graphs?

## Online Graph Dictionary Learning

---

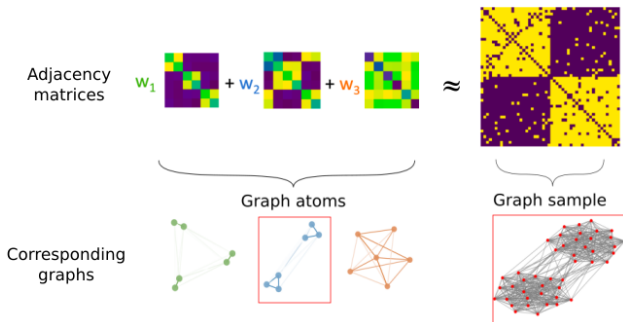
# Datasets of graphs



SBM with balanced communities  $\{1, 2, 3\}$ .

Two communities of variable proportions.

- We have access to **large datasets of graphs** with variable number of nodes.
- How to model the variability of those graphs?
- A natural formulation is to use **factorization**.
- We propose to use a **linear** model for representing the graph associated to and estimation of the linear basis : **Dictionary learning**.



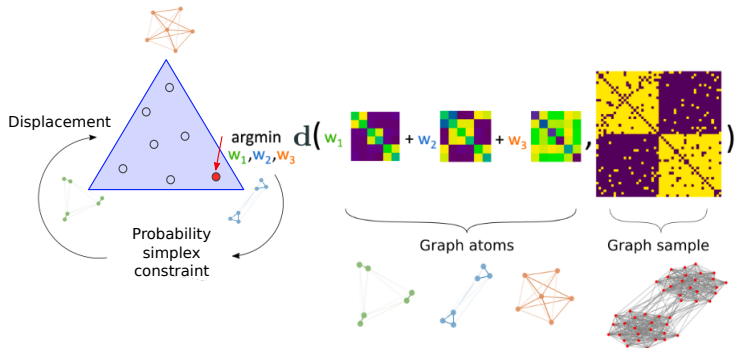
## Linear modeling of graphs

$$D \approx \sum_{s \in [S]} w_s \overline{D}_s \quad (2)$$

- Approximate a given graph structure  $D$  as a non-negative weighted sum of template graphs  $\overline{D}_s$ .
- $\{\overline{D}_s\}_s$  is the dictionary of templates that all have the same order (nb. of nodes).



# Gromov-Wasserstein Linear unmixing

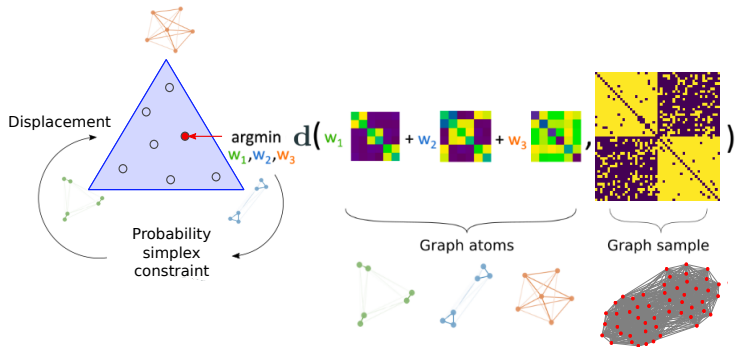


## Sparse linear unmixing with Gromov-Wasserstein

$$\min_{\mathbf{w} \in \Sigma_S} \mathcal{GW}_2^2 \left( \sum_{s \in [S]} w_s \overline{D}_s, D \right) - \lambda \|\mathbf{w}\|_2^2 \quad (3)$$

- Estimate the linear representation on the simplex  $\mathbf{w}$  minimizing the GW distance *w.r.t.* the target graph  $D$  (non-negative unmixing).
- $\lambda \in \mathbb{R}_+$ , **negative quadratic regularization** promotes sparsity on the simplex [Li et al., 2016] while keeping a nonconvex QP.

# Gromov-Wasserstein Linear unmixing

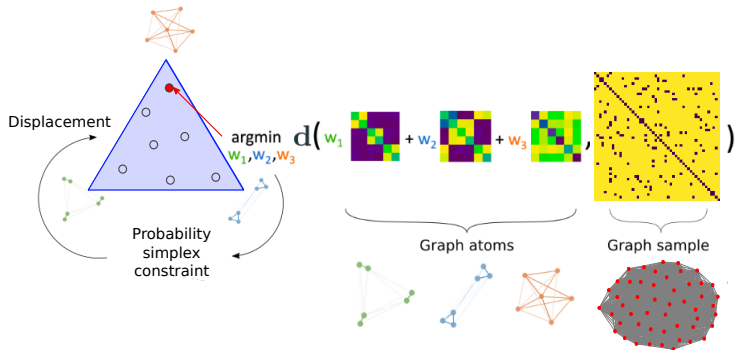


## Sparse linear unmixing with Gromov-Wasserstein

$$\min_{\mathbf{w} \in \Sigma_S} \text{GW}_2^2 \left( \sum_{s \in [S]} w_s \overline{D}_s, D \right) - \lambda \|\mathbf{w}\|_2^2 \quad (3)$$

- Estimate the linear representation on the simplex  $\mathbf{w}$  minimizing the GW distance *w.r.t.* the target graph  $D$  (non-negative unmixing).
- $\lambda \in \mathbb{R}_+$ , **negative quadratic regularization** promotes sparsity on the simplex [Li et al., 2016] while keeping a nonconvex QP.

# Gromov-Wasserstein Linear unmixing



## Sparse linear unmixing with Gromov-Wasserstein

$$\min_{\mathbf{w} \in \Sigma_S} \text{GW}_2^2 \left( \sum_{s \in [S]} w_s \overline{D}_s, D \right) - \lambda \|\mathbf{w}\|_2^2 \quad (3)$$

- Estimate the linear representation on the simplex  $\mathbf{w}$  minimizing the GW distance *w.r.t.* the target graph  $D$  (non-negative unmixing).
- $\lambda \in \mathbb{R}_+$ , **negative quadratic regularization** promotes sparsity on the simplex [Li et al., 2016] while keeping a nonconvex QP.

# Approximating GW in the linear embedding

## GW Upper bound [Vincent-Cuaz et al., 2021]

Let two graphs of order  $N$  in the linear embedding  $\left(\sum_s w_s^{(1)} \overline{\mathbf{D}}_s\right)$  and  $\left(\sum_s w_s^{(2)} \overline{\mathbf{D}}_s\right)$ , the  $\mathcal{GW}$  divergence can be upper bounded by

$$\mathcal{GW}_2 \left( \sum_{s \in [S]} w_s^{(1)} \overline{\mathbf{D}}_s, \sum_{s \in [S]} w_s^{(2)} \overline{\mathbf{D}}_s \right) \leq \|\mathbf{w}^{(1)} - \mathbf{w}^{(2)}\|_M \quad (4)$$

with  $M$  a PSD matrix of components  $M_{p,q} = \langle \mathbf{D}_h \overline{\mathbf{D}}_p, \overline{\mathbf{D}}_q \mathbf{D}_h \rangle_F$ ,  $\mathbf{D}_h = \text{diag}(\mathbf{h})$ .

## Discussion

- The upper bound is the value of GW for a transport  $T = \text{diag}(\mathbf{h})$  assuming that the nodes are already aligned.
- The bound is exact when the weights  $\mathbf{w}^{(1)}$  and  $\mathbf{w}^{(2)}$  are close.
- Solving  $\mathcal{GW}$  with FW is  $O(N^3 \log(N))$  at each iterations.
- Computing the Mahalanobis upper bound is  $O(S^2)$ : very fast alternative to GW for nearest neighbors retrieval.

## GDL optimization problem

$$\min_{\{\mathbf{w}^{(k)}\}_{k \in [K]}, \{\overline{\mathbf{D}}_s\}_{s \in [S]}} \sum_{k=1}^K \mathcal{GW}_2^2 \left( \mathbf{D}^{(k)}, \sum_{s \in [S]} w_s^{(k)} \overline{\mathbf{D}}_s \right) - \lambda \|\mathbf{w}^{(k)}\|_2^2 \quad (5)$$

- On a dataset of  $K$  undirected graphs  $\{\mathbf{D}^{(k)} \in S_{N^{(k)}}(\mathbb{R})\}_{k \in [K]}$ .
- We want to estimate simultaneously the unmixing  $\mathbf{w}^{(k)}$  of each graphs and the optimal dictionary  $\{\overline{\mathbf{D}}_s\}_{s \in [S]}$ .
- Very similar to classical DL (Non-negative Matrix Factorization) approach but with GW as a data fitting term.
- We propose to solve it an adaptation of the online algorithm [Mairal et al., 2009]

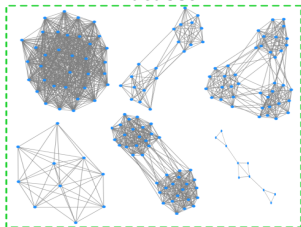
## Stochastic/Online update [Vincent-Cuaz et al., 2021]

- 1: Sample a minibatch of graphs  $\mathcal{B} := \{\mathbf{D}^{(k)}\}_{k \in \mathcal{B}}$ .
- 2: Compute  $\{(\mathbf{w}^{(k)}, \mathbf{T}^{(k)})\}_{k \in [B]}$  from solving  $B$  independent unmixings.
- 3: Compute the gradient  $\tilde{\nabla}_{\overline{\mathbf{D}}_s}$  on the minibatch with fixed  $\{(\mathbf{w}^{(k)}, \mathbf{T}^{(k)})\}_{k \in [B]}$ .
- 4: Projected gradient step,  $\forall s \in [S], \overline{\mathbf{D}}_s \leftarrow Proj_{S_N(\mathbb{R})}(\overline{\mathbf{D}}_s - \eta_C \tilde{\nabla}_{\overline{\mathbf{D}}_s})$

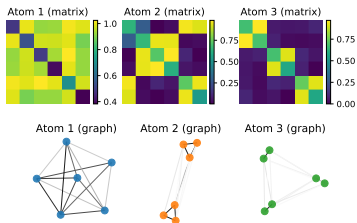
# Experiments - Unsupervised representation learning

- Stochastic block model with  $\{1, 2, 3\}$  blocks

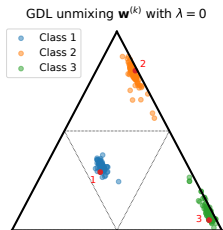
## Dataset



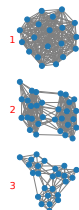
## Learned atoms



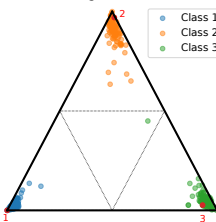
## Embedding space



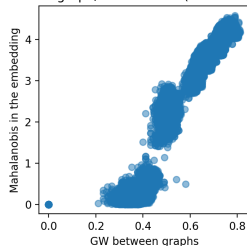
Examples



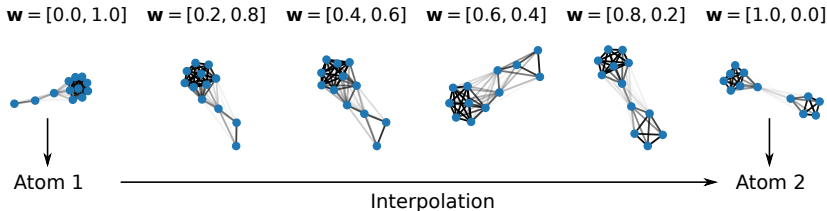
GDL unmixing  $\mathbf{w}^{(k)}$  with  $\lambda = 0.001$



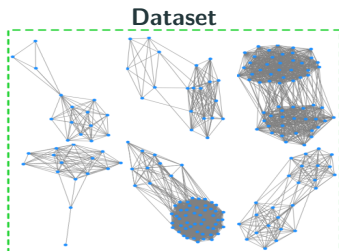
GW graph/Mahalanobis (corr=0.96)



# Experiments - Unsupervised representation learning



## Learned Dictionary: Interpolation $\sim$ 1D Manifold



- Stochastic block model with 2 blocks and varying proportions of block size.
- GDL with 2 atoms can recover the extreme points.
- Linear interpolation recover a continuous variation of proportion.

Table 1. Clustering: Rand Index computed for benchmarked approaches on real datasets.

models	no attribute		discrete attributes		real attributes			
	IMDB-B	IMDB-M	MUTAG	PTC-MR	BZR	COX2	ENZYMES	PROTEIN
GDL(ours)	<b>51.64(0.59)</b>	55.41(0.20)	<b>70.89(0.11)</b>	<b>51.90(0.54)</b>	<b>66.42(1.96)</b>	<b>59.48(0.68)</b>	66.97(0.93)	<b>60.49(0.71)</b>
GWF-r	51.24 (0.02)	<b>55.54(0.03)</b>	-	-	52.42(2.48)	56.84(0.41)	<b>72.13(0.19)</b>	59.96(0.09)
GWF-f	50.47(0.34)	54.01(0.37)	-	-	51.65(2.96)	52.86(0.53)	71.64(0.31)	58.89(0.39)
GW-k	50.32(0.02)	53.65(0.07)	57.56(1.50)	50.44(0.35)	56.72(0.50)	52.48(0.12)	66.33(1.42)	50.08(0.01)
SC	50.11(0.10)	54.40(9.45)	50.82(2.71)	50.45(0.31)	42.73(7.06)	41.32(6.07)	70.74(10.60)	49.92(1.23)

## Clustering Experiments on real datasets

- Different data fitting losses:
  - Graphs without node attributes : Gromov-Wasserstein.
  - Graphs with node attributes (discrete and real): Fused Gromov-Wasserstein.
- We learn a dictionary on the dataset and perform K-means in the embedding using the Mahalanobis distance approximation.
- Compared to GW factorization [Xu, 2020] and spectral clustering.
- Similar performance for supervised classification (using GW in a kernel).

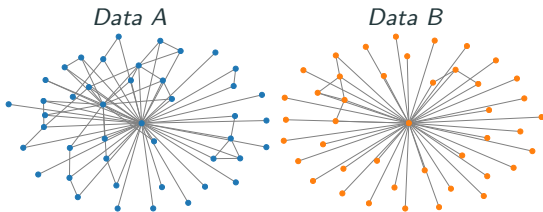


# Experiments - Online Learning

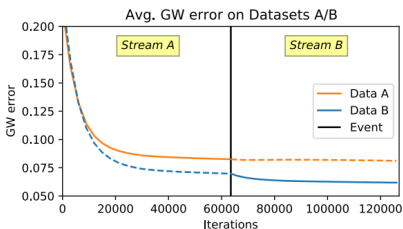
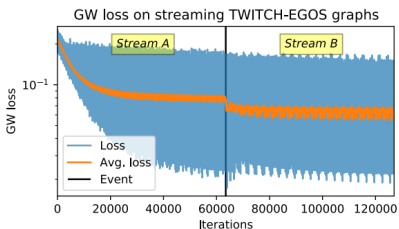
- **Streaming graphs:** Stochastic update for each new incoming graph

- Dataset: **TWITCH-EGOS**

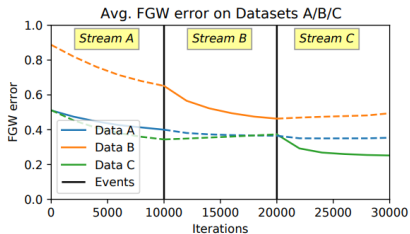
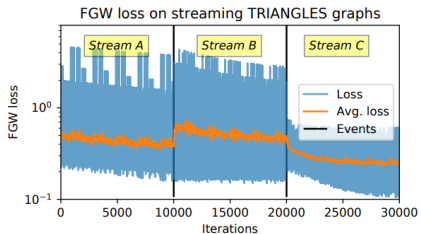
- 120.000+ graphs
- 2 classes
- shared hub structure

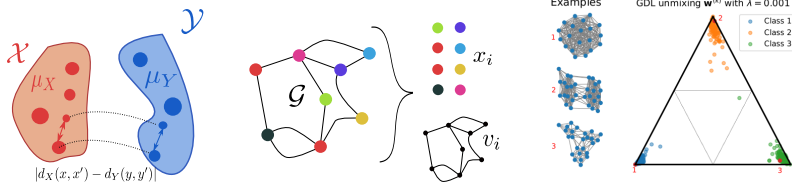


- **Simulated stream:** data A (class 1)  $\rightarrow$  data B (class 2)



- **Streaming graphs:** Stochastic update for each new incoming graph
- Dataset : **TRIANGLES**
  - 30.000+ labeled graphs
  - 10 classes
- **Simulated stream:** data A (4 classes) → data B (3 classes) → data C (3 classes)





## Gromov-Wasserstein family for graph modeling

- Graphs modelled as distributions,  $\mathcal{GW}$  can measure their similarity.
- Extensions of GW for labeled graphs and Frechet means can be computed.
- Nonlinear and linear dictionaries of graphs using  $\mathcal{GW}$  provide a good modeling.

## Open questions

- Stability of the  $\mathcal{GW}$  plan to perturbations of  $\mathbf{D}$  (related to the GDL upper bound).
- Use  $\mathcal{GW}$  as a "kernel" for structured prediction ( $\mathcal{GW}$  barycenters).
- Weights on the nodes are important but rarely available : relax the constraints [Séjourné et al., 2020] or even remove one of them (WIP).

Python code available on GitHub:

<https://github.com/PythonOT/POT>

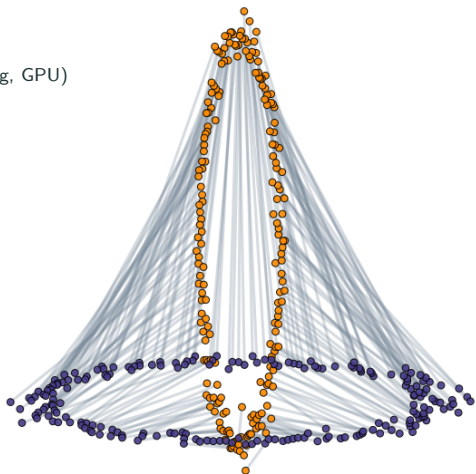
- OT LP solver, Sinkhorn (stabilized,  $\epsilon$ -scaling, GPU)
- Domain adaptation with OT.
- Barycenters, Wasserstein unmixing.
- Wasserstein Discriminant Analysis.

Tutorial on OT for ML:

<http://tinyurl.com/otml-isbi>

Papers available on my website:

<https://remi.flamary.com/>



## Optimization problem

$$\min_{\mathbf{w} \in \Sigma_S} \mathcal{GW}_2^2 \left( \sum_{s \in [S]} w_s \overline{\mathbf{D}}_s, \mathbf{D} \right) - \lambda \|\mathbf{w}\|_2^2$$

- Non-convex Quadratic Program *w.r.t.*  $\mathbf{T}$  and  $\mathbf{w}$ .
- GW for fixed  $\mathbf{w}$  already have an existing Frank-Wolfe solver.
- We proposed a Block Coordinate Descent algorithm

## BCD Algorithm for sparse GW unmixing [Tseng, 2001]

- 1: **repeat**
  - 2:   Compute OT matrix  $\mathbf{T}$  of  $\mathcal{GW}_2^2(\mathbf{D}, \sum_s w_s \overline{\mathbf{D}}_s)$ , with FW [Vayer et al., 2018].
  - 3:   Compute the optimal  $\mathbf{w}$  given  $\mathbf{T}$  with Frank-Wolfe algorithm.
  - 4: **until** convergence
- Since the problem is quadratic optimal steps can be obtained for both FW.
  - BCD convergence in practice in a few tens of iterations.

## GDL on labeled graphs

- For datasets with labeled graphs, one can learn simultaneously a dictionary of the structure  $\{\overline{\mathbf{D}}_s\}_{s \in [S]}$  and a dictionary on the labels/features  $\{\overline{\mathbf{F}}_s\}_{s \in [S]}$ .
- Data fitting is Fused Gromov-Wasserstein distance  $\mathcal{FGW}$ , same stochastic algorithm.

## Dictionary on weights

$$\min_{\substack{\{(\mathbf{w}^{(k)}, \mathbf{v}^{(k)})\}_k \\ \{(\overline{\mathbf{D}}_s, \overline{\mathbf{h}}_s)\}_s}} \sum_{k=1}^K \mathcal{GW}_2^2 \left( \mathbf{D}^{(k)}, \sum_s w_s^{(k)} \overline{\mathbf{D}}_s, \mathbf{h}^{(k)}, \sum_s v_s^{(k)} \overline{\mathbf{h}}_s \right) - \lambda \|\mathbf{w}^{(k)}\|_2^2 - \mu \|\mathbf{v}^{(k)}\|_2^2$$

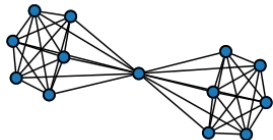
- We model the graphs as a linear model on the structure and the node weights

$$(\mathbf{D}^{(k)}, \mathbf{h}^{(k)}) \longrightarrow \left( \sum_s w_s^{(k)} \overline{\mathbf{D}}_s, \sum_s v_s^{(k)} \overline{\mathbf{h}}_s \right)$$

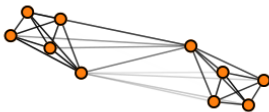
- This allows for sparse weights  $\mathbf{h}$  so embedded graphs with different order.
- We provide in [Vincent-Cuaz et al., 2021] subgradients of GW *w.r.t.* the mass  $\mathbf{h}$ .

# Experiments - Unsupervised representation learning

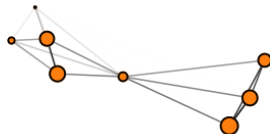
Graph from dataset



Model unif.  $\mathbf{h}$  (GW=0.09)







Model est.  $\tilde{\mathbf{h}}$  (GW=0.08)



## Comparison of fixed and learned weights dictionaries

- Graph taken from the IMBD dataset.
- Show original graph and representation after projection on the embedding.
- Uniform weight  $\mathbf{h}$  has a hard time representing a central node.
- Estimated weights  $\tilde{\mathbf{h}}$  recover a central node.
- In addition some nodes are discarded with 0 weight (graphs can change order).

-  Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., and Peyré, G. (2015).  
**Iterative Bregman projections for regularized transportation problems.**  
*SISC*.
-  Cuturi, M. (2013).  
**Sinkhorn distances: Lightspeed computation of optimal transportation.**  
In *Neural Information Processing Systems (NIPS)*, pages 2292–2300.
-  Genevay, A., Chizat, L., Bach, F., Cuturi, M., and Peyré, G. (2018).  
**Sample complexity of sinkhorn divergences.**  
*arXiv preprint arXiv:1810.02733*.
-  Kantorovich, L. (1942).  
**On the translocation of masses.**  
*C.R. (Doklady) Acad. Sci. URSS (N.S.)*, 37:199–201.





Li, P., Rangapuram, S. S., and Slawski, M. (2016).

**Methods for sparse and low-rank recovery under simplex constraints.**

*arXiv preprint arXiv:1605.00507.*



Loosli, G., Canu, S., and Ong, C. S. (2015).

**Learning svm in krein spaces.**

*IEEE transactions on pattern analysis and machine intelligence*, 38(6):1204–1216.



Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2009).

**Online dictionary learning for sparse coding.**

In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696.



Memoli, F. (2011).

**Gromov wasserstein distances and the metric approach to object matching.**

*Foundations of Computational Mathematics*, pages 1–71.



Monge, G. (1781).

**Mémoire sur la théorie des déblais et des remblais.**

De l'Imprimerie Royale.



Peyré, G., Cuturi, M., and Solomon, J. (2016).

**Gromov-wasserstein averaging of kernel and distance matrices.**

In *ICML*, pages 2664–2672.



Rakotomamonjy, A., Traore, A., Berar, M., Flamary, R., and Courty, N. (2018).

**Wasserstein Distance Measure Machines.**

preprint.



Scetbon, M., Peyré, G., and Cuturi, M. (2021).

**Linear-time gromov wasserstein distances using low rank couplings and costs.**

*arXiv preprint arXiv:2106.01128.*



Séjourné, T., Vialard, F.-X., and Peyré, G. (2020).

**The unbalanced gromov wasserstein distance: Conic formulation and relaxation.**

*arXiv preprint arXiv:2009.04266.*



Solomon, J., Peyré, G., Kim, V. G., and Sra, S. (2016).

**Entropic metric alignment for correspondence problems.**

*ACM Transactions on Graphics (TOG)*, 35(4):72.



Tseng, P. (2001).





**Convergence of a block coordinate descent method for nondifferentiable minimization.**

*Journal of optimization theory and applications*, 109(3):475–494.



Vayer, T., Chapel, L., Flamary, R., Tavenard, R., and Courty, N. (2018).

**Fused gromov-wasserstein distance for structured objects: theoretical foundations and mathematical properties.**

-  Vayer, T., Chapel, L., Flamary, R., Tavenard, R., and Courty, N. (2020).  
**Fused gromov-wasserstein distance for structured objects.**  
*Algorithms*, 13 (9):212.
-  Vayer, T., Flamary, R., Tavenard, R., Chapel, L., and Courty, N. (2019).  
**Sliced gromov-wasserstein.**  
In *Neural Information Processing Systems (NeurIPS)*.
-  Vincent-Cuaz, C., Vayer, T., Flamary, R., Corneli, M., and Courty, N. (2021).  
**Online graph dictionary learning.**  
In *International Conference on Machine Learning (ICML)*.
-  Xu, H. (2020).  
**Gromov-wasserstein factorization models for graph clustering.**  
In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34,  
pages 6478–6485.