



INSTITUT
POLYTECHNIQUE
DE PARIS

Optimal transport for graph modeling and Graph Neural Networks

Rémi Flamary - CMAP, École Polytechnique, Institut Polytechnique de Paris

May 02 2023

Séminaire Palaisien, ENSAE

Collaborators



N. Courty



A. Rakotomamonjy



D. Tuia



A. Habrard



M. Perrot



M. Ducoffe



M. Cuturi



K. Lounici



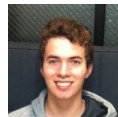
A. Férrari



C. Févotte



V. Emiya



V. Seguy



B. Damodaran



T. Vayer



L. Chapel



R. Tavenard



K. Fatras



I. Redko



H. Janati



T. Séjourné



H. Tran



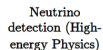
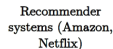
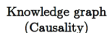
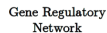
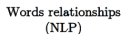
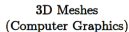
G. Gasso



M. Corneli



C. Vincent-Cuaz



Optimal Transport and divergences between graphs

- Discrete Optimal Transport (OT)

- Gromov-Wasserstein divergence and applications on graphs

- Fused Gromov-Wasserstein and applications on attributed graphs

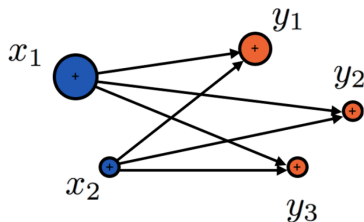
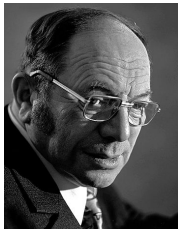
Template based Graph Neural Network with Optimal Transport Distances

- Graph Neural Network

- Template based GNN with FGW

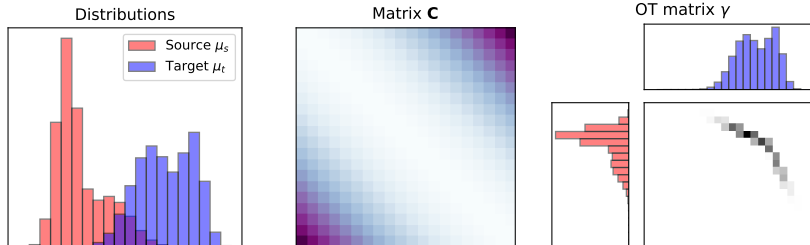
- Numerical experiments

Optimal Transport and divergences between graphs



- Problem introduced by Gaspard Monge in his memoir [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 originally for resource allocation problems

Optimal transport between discrete distributions



Kantorovitch formulation : OT Linear Program

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

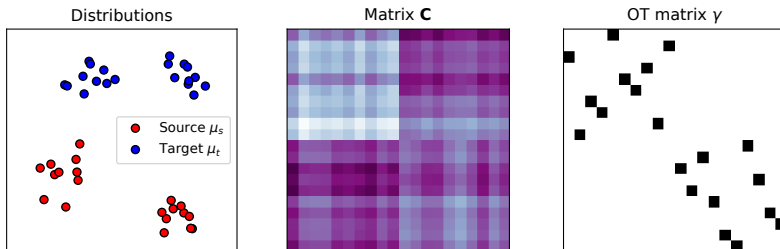
$$W_p^p(\mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \left\{ \langle \mathbf{T}, \mathbf{C} \rangle_F = \sum_{i,j} T_{i,j} c_{i,j} \right\}$$

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

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

- $W_p(\mu_s, \mu_t)$ is called the Wasserstein distance (EMD for $p = 1$).
- Entropic regularization solved efficiently with Sinkhorn [Cuturi, 2013].
- Classical OT needs distributions lying in the same space $\rightarrow$ Gromov-Wasserstein.

Optimal transport between discrete distributions



Kantorovitch formulation : OT Linear Program

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

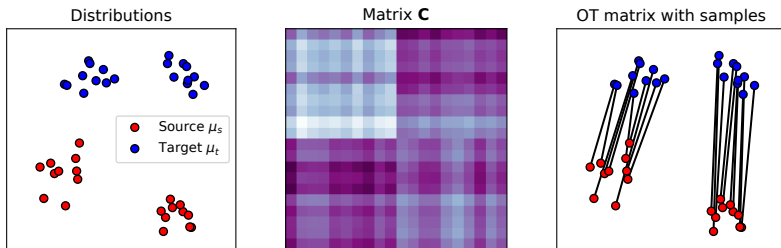
$$W_p^p(\mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \left\{ \langle \mathbf{T}, \mathbf{C} \rangle_F = \sum_{i,j} T_{i,j} c_{i,j} \right\}$$

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

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

- $W_p(\mu_s, \mu_t)$ is called the Wasserstein distance (EMD for $p = 1$).
- Entropic regularization solved efficiently with Sinkhorn [Cuturi, 2013].
- Classical OT needs distributions lying in the same space $\rightarrow$ Gromov-Wasserstein.

Optimal transport between discrete distributions



Kantorovitch formulation : OT Linear Program

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

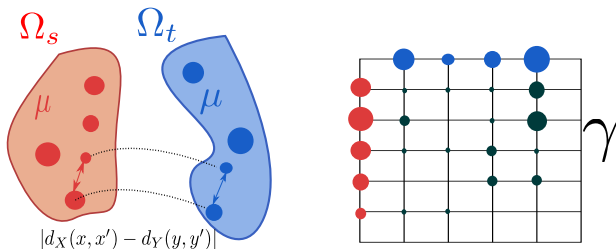
$$W_p^p(\mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \left\{ \langle \mathbf{T}, \mathbf{C} \rangle_F = \sum_{i,j} T_{i,j} c_{i,j} \right\}$$

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

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

- $W_p(\mu_s, \mu_t)$ is called the Wasserstein distance (EMD for $p = 1$).
- Entropic regularization solved efficiently with Sinkhorn [Cuturi, 2013].
- Classical OT needs distributions lying in the same space $\rightarrow$ Gromov-Wasserstein.

Gromov-Wasserstein divergence



Inspired from Gabriel Peyré

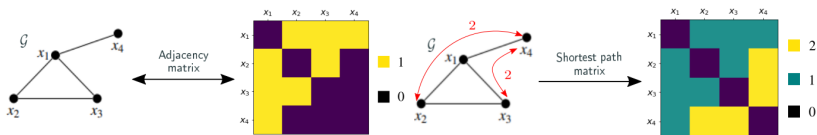
GW for discrete distributions [Memoli, 2011]

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

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).
- Entropy regularize GW proposed in [Peyré et al., 2016].

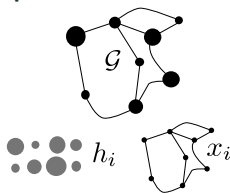
Gromov-Wasserstein between graphs



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 or geodesic distance matrix.

Graph as a distribution (D, h)

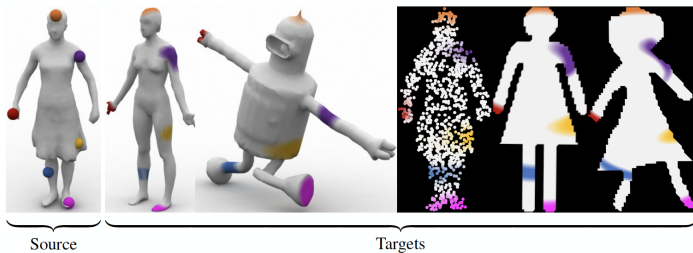


- Graph represented as a discrete distribution:

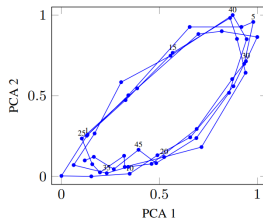
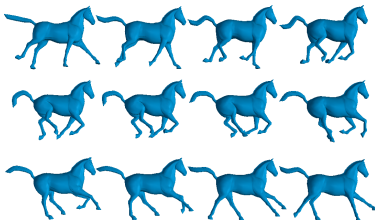
$$\mu_X = \sum_i h_i \delta_{x_i}$$

- 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).

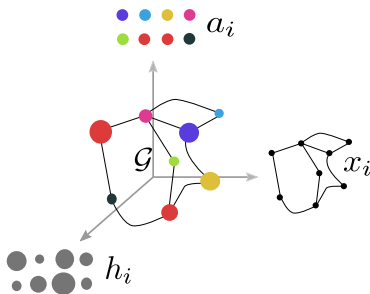
Shape matching between 3D and 2D surfaces



Multidimensional scaling (MDS) of shape collection



Attributed graphs as distributions

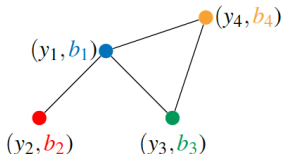
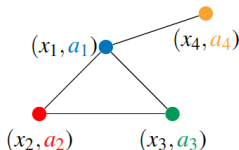


$$\left\{ \begin{array}{c} \text{graph icon} \\ \text{dots} \end{array} \right\} \mu = \sum_i h_i \delta_{(x_i, a_i)}$$

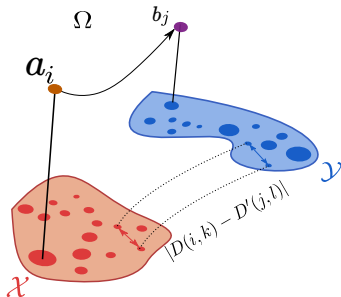
$$\left\{ \begin{array}{c} \text{dots} \\ \text{dots} \end{array} \right\} \mu_A = \sum_i h_i \delta_{a_i}$$

$$\left\{ \begin{array}{c} \text{graph icon} \\ \text{dots} \end{array} \right\} \mu_X = \sum_i h_i \delta_{x_i}$$

- Joint distribution μ in the feature/structure space.
 - Nodes are weighted by their mass h_i .
 - Structure encoded by x_i (no common metric between two different graphs).
 - Features values a_i can be compared through the common metric.
- Importance of the joint modeling:



Fused Gromov-Wasserstein distance



Fused Gromov Wasserstein distance [Vayer et al., 2020]

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

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

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.

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

Metric properties [Vayer et al., 2020]

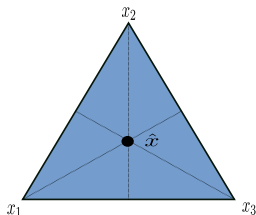
- $\mathcal{FGW}$ defines a metric over structured data with **measure and features preserving isometries** as invariants.
- $\mathcal{FGW}$ is a metric for $q = 1$ a semi metric for $q > 1$, $\forall p \geq 1$.
- The distance is nul *iff* :
 - There exists a Monge map $T \# \mu_s = \mu_t$.
 - Structures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

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

- $\mathcal{FGW}(\mu_s, \mu_t)$ is lower bounded by $(1 - \alpha)\mathcal{W}(\mu_A, \mu_B)^q$ and $\alpha\mathcal{GW}(\mu_X, \mu_Y)^q$
- Convergence of finite samples when $\mathcal{X} = \mathcal{Y}$ with $d = \text{Dim}(\mathcal{X}) + \text{Dim}(\Omega)$:

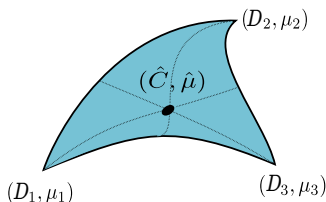
$$\mathbb{E}[\mathcal{FGW}(\mu, \mu_n)] = O\left(n^{-\frac{1}{d}}\right)$$

Euclidean barycenter



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

FGW barycenter



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

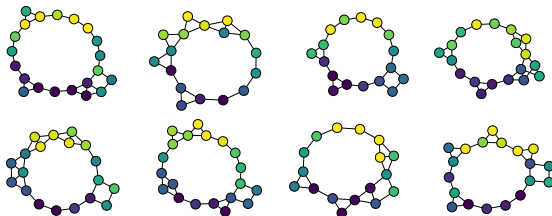
FGW barycenter $p = 1, q = 2$

- 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



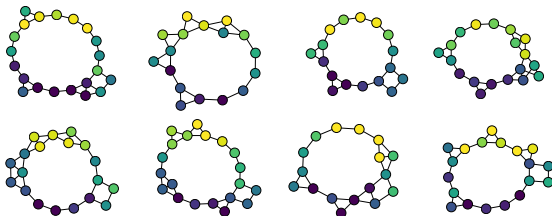
Barycenter of noisy graphs

- 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.

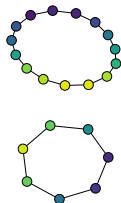
Noiseless graph



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



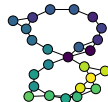
Barycenter of noisy graphs

- 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.

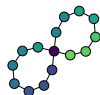
Noiseless graph



Noisy graphs samples



Barycenter

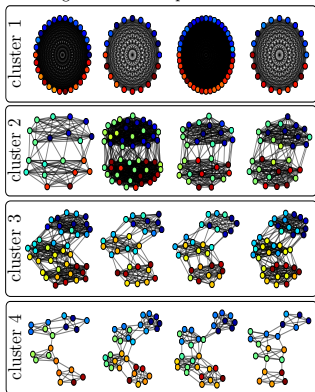


Barycenter of noisy graphs

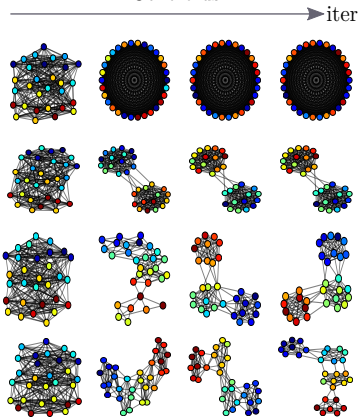
- We select a clean graph, change the number of nodes and add label noise and random connections.
- We compute the barycenter on $n = 15$ and $n = 7$ nodes.
- Barycenter graph is obtained through thresholding of the D matrix.

FGW for graphs based clustering

Training dataset examples



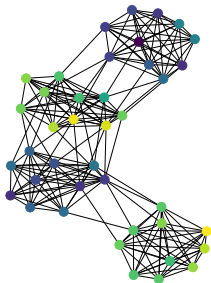
Centroids



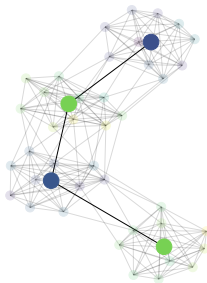
- Clustering of multiple real-valued graphs. Dataset composed of 40 graphs (10 graphs $\times$ 4 types of communities)
- k -means clustering using the FGW barycenter

FGW barycenter for community clustering

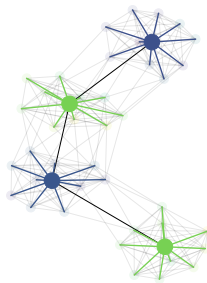
Graph with communities



Approximate Graph



Clustering with transport matrix

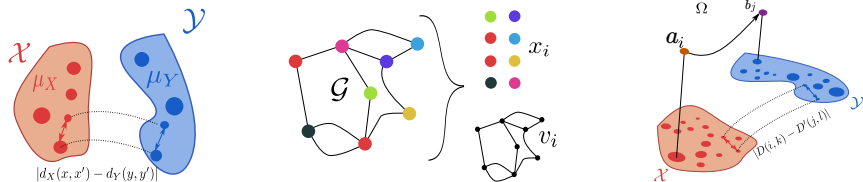


Graph approximation and community clustering

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

- Approximate the graph $(\mathbf{D}_0, \mu_0)$ with a small number of nodes.
- Can be seen as a FGW (compressed) barycenter for one graph.
- 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]

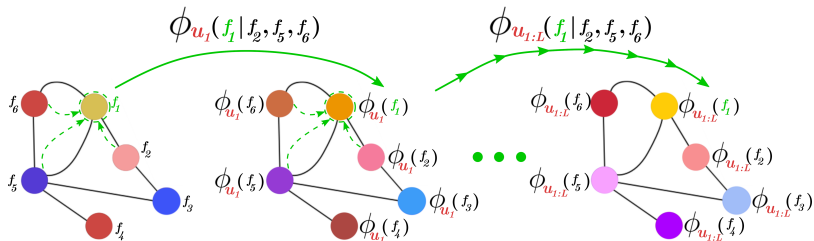
- Divergence between distributions across metric spaces.
- Can be used to measure similarity between graphs seen as distribution their pairwise node relationship.

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?

Template based Graph Neural Network with Optimal Transport Distances

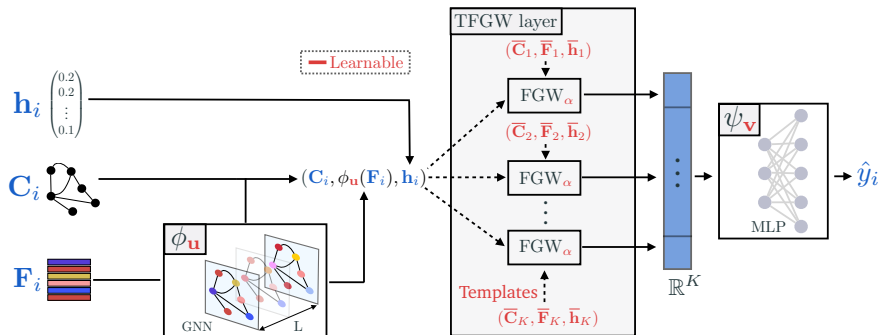


Principle [Bronstein et al., 2017]

- Each layer of the GNN compute features on graph node using the values from the connected neighbors : message passing principle.
- A step of global aggregation or pooling allows to go from a complex graph object to a vector representation.
- The pooling step must remain invariant to permutations (min, max, mean).

Can we encode graphs as distributions in GNN?

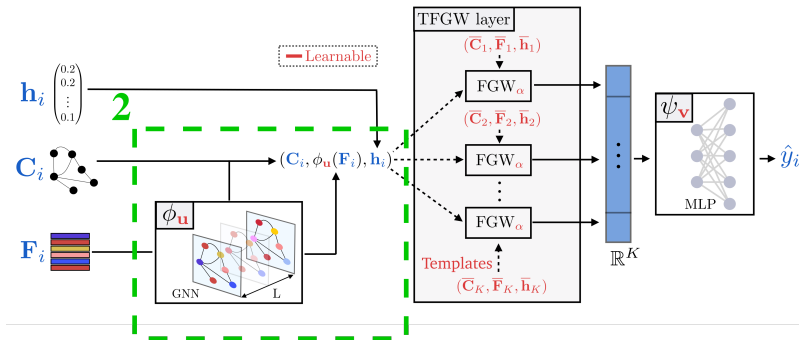
Template based Graph Neural Network with OT Distances



Template based FGW layer (TFGW) [Vincent-Cuaz et al., 2022b]

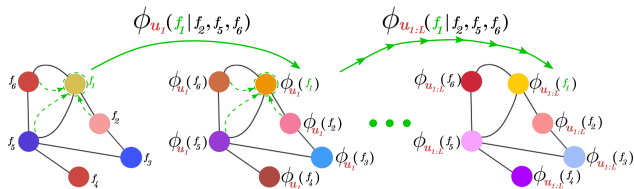
- Principle: represent a graph through its distances to learned templates.
- Novel pooling layer derived from OT distances.
- New end-to-end GNN models for graph-level tasks.
- Learnable parameters are illustrated in red above.

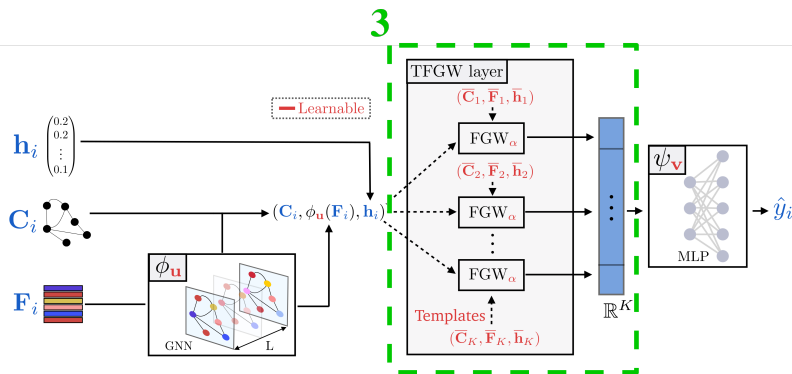
Template based Graph Neural Network with Optimal Transport Distances



2. Node embeddings

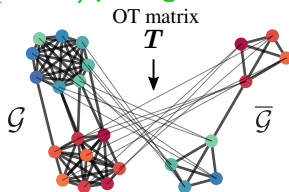
- $\phi_{\mathbf{u}}$: GNN of L layers parameterized by $\mathbf{u}$ e.g GIN, GAT, etc.
- Promotes discriminant features on the nodes $\phi_{\mathbf{u}}(\mathcal{F}_i)$



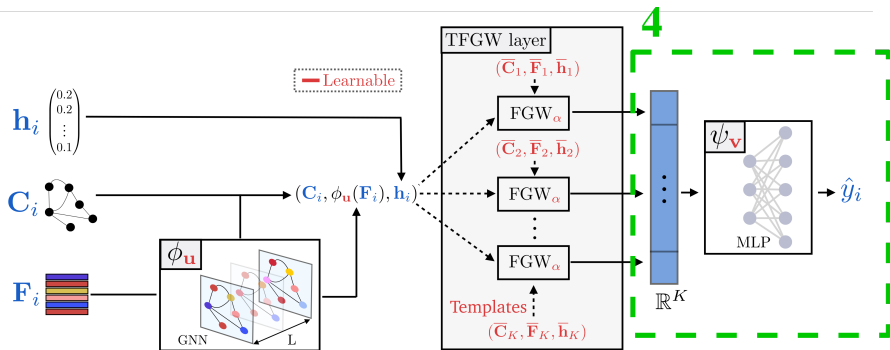


3. Template-based Fused Gromov-Wasserstein (TFGW) pooling

- FGW $_{\alpha}$: OT soft graph matching distance.
- $\alpha \in [0; 1]$: relative importance between structure $\mathbf{D}_i$ and node features $\phi_{\mathbf{u}}(\mathcal{F}_i)$.
- $\{\bar{\mathbf{D}}_k, \bar{\mathcal{F}}_k, \bar{\mathbf{h}}_k\}$: FGW distances to K templates used as graph representation.



Template based Graph Neural Network with Optimal Transport Distances



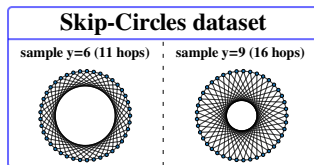
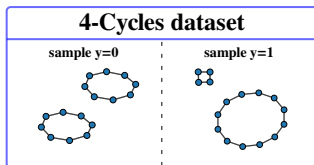
4. Final MLP for predictions

- $\psi_{\mathbf{v}}$: MLP with non-linearities fed with the distance embeddings.
- $\hat{y}_i$: final prediction for graph-level tasks (classification or regression).
- End-to-end optimization of all parameters:
 - $\mathbf{u}$ and $\mathbf{v}$ parameters of GNN $\phi_{\mathbf{u}}$ and final MLP $\psi_{\mathbf{v}}$.
 - $\{\bar{\mathbf{D}}_k, \bar{\mathcal{F}}_k, \bar{\mathbf{h}}_k\}$ TFGW graph templates.

category	model	MUTAG	PTC	ENZYMES	PROTEIN	NCI1	IMDB-B	IMDB-M	COLLAB
Ours ($\phi_u = \text{GIN}$)	TFGW ADJ (L=2)	96.4(3.3)	72.4(5.7)	<u>73.8(4.6)</u>	82.9(2.7)	88.1(2.5)	78.3(3.7)	56.8(3.1)	84.3(2.6)
	TFGW SP (L=2)	<u>94.8(3.5)</u>	<u>70.8(6.3)</u>	75.1(5.0)	<u>82.0(3.0)</u>	<u>86.1(2.7)</u>	<u>74.1(5.4)</u>	<u>54.9(3.9)</u>	80.9(3.1)
OT emb.	OT-GNN (L=2)	91.6(4.6)	68.0(7.5)	66.9(3.8)	76.6(4.0)	82.9(2.1)	67.5(3.5)	52.1(3.0)	80.7(2.9)
	OT-GNN (L=4)	92.1(3.7)	65.4(9.6)	67.3(4.3)	78.0(5.1)	83.6(2.5)	69.1(4.4)	51.9(2.8)	81.1(2.5)
	WEGL	91.0(3.4)	66.0(2.4)	60.0(2.8)	73.7(1.9)	75.5(1.4)	66.4(2.1)	50.3(1.0)	79.6(0.5)
GNN	PATCHYSAN	91.6(4.6)	58.9(3.7)	55.9(4.5)	75.1(3.3)	76.9(2.3)	62.9(3.9)	45.9(2.5)	73.1(2.7)
	GIN	90.1(4.4)	63.1(3.9)	62.2(3.6)	76.2(2.8)	82.2(0.8)	64.3(3.1)	50.9(1.7)	79.3(1.7)
	DropGIN	89.8(6.2)	62.3(6.8)	65.8(2.7)	76.9(4.3)	81.9(2.5)	66.3(4.5)	51.6(3.2)	80.1(2.8)
	PPGN	90.4(5.6)	65.6(6.0)	66.9(4.3)	77.1(4.0)	82.7(1.8)	67.2(4.1)	51.3(2.8)	81.0(2.1)
	DIFFPOOL	86.1(2.0)	45.0(5.2)	61.0(3.1)	71.7(1.4)	80.9(0.7)	61.1(2.0)	45.8(1.4)	80.8(1.6)
Kernels	FGW - ADJ	82.6(7.2)	55.3(8.0)	72.2(4.0)	72.4(4.7)	74.4(2.1)	70.8(3.6)	48.9(3.9)	80.6(1.5)
	FGW - SP	84.4(7.3)	55.5(7.0)	70.5(6.2)	74.3(3.3)	72.8(1.5)	65.0(4.7)	47.8(3.8)	77.8(2.4)
	WL	87.4(5.4)	56.0(3.9)	69.5(3.2)	74.4(2.6)	85.6(1.2)	67.5(4.0)	48.5(4.2)	78.5(1.7)
	WWL	86.3(7.9)	52.6(6.8)	71.4(5.1)	73.1(1.4)	85.7(0.8)	71.6(3.8)	52.6(3.0)	<u>81.4(2.1)</u>
Gain with TFGW		+4.3	+4.4	+2.9	+4.9	+2.4	+6.7	+4.2	+2.9

- Comparison with state of the art approach from GNN and graph kernel methods.
- Systematic and significant gain of performance with GIN+TFGW.
- Gain independent of GNN architecture (GIN or GAT).

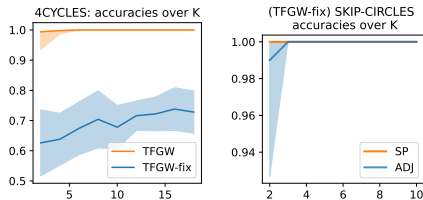
Going beyond Weisfeiler-Lehman Isomorphism tests



TFGW with $K = \#labels$

model	4-Cycles	Skip-Circles
TFGW	0.99(0.03)	1.00(0.00)
TFGW-fix	0.63(0.11)	1.00(0.00)
OT-GNN	0.50(0.00)	0.10(0.00)
GIN	0.50(0.00)	0.10(0.00)
DropGIN	1.00(0.01)	0.82(0.28)
PPGN	1.00(0.01)	0.90(0.11)

TFGW with various K

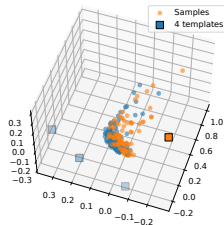


- TFGW has good expressivity on problems beyond WL test.
- Learning the templates is important : $TFGW > TFGW\ fix.$

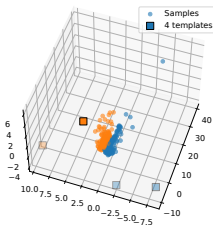
Graph OT distance embedding

TFGW embedding (PCA)

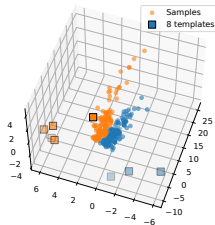
PCA in the FGW embedding for PTC with no GNN



PCA in the FGW embedding for PTC with 2 GIN layers

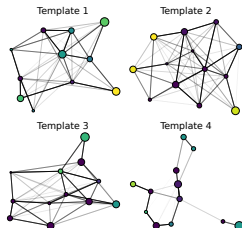


PCA in the FGW embedding for PTC with 2 GIN layers

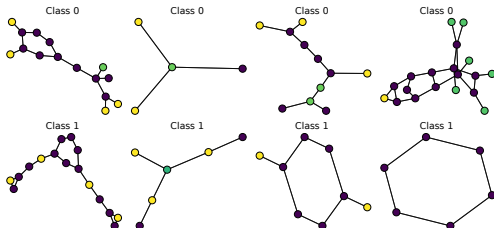


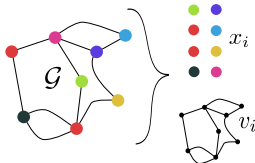
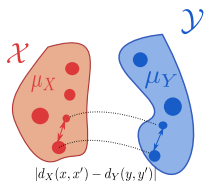
Learned templates (left) and data samples (right)

FGW templates for PTC

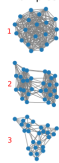


Samples from PTC

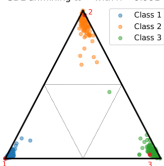




Examples



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

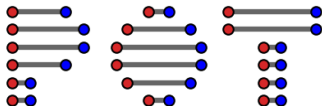


Gromov-Wasserstein family for graph modeling

- Graphs modelled as distributions, $\mathcal{GW}$ can measure their similarity.
- Extensions of GW for labeled graphs and Frechet means can be computed.
- Weights on the nodes are important but rarely available : relax the constraints [Séjourné et al., 2020] or even remove one of them [Vincent-Cuaz et al., 2022a].
- Many applications of FGW from brain imagery [Thual et al., 2022] to Graph Neural Networks [Vincent-Cuaz et al., 2022b].

Thank you

Python code available on GitHub:



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

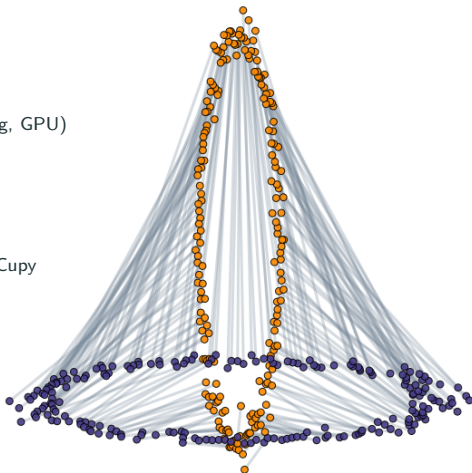
- OT LP solver, Sinkhorn (stabilized, ϵ -scaling, GPU)
- Domain adaptation with OT.
- Barycenters, Wasserstein unmixing.
- Gromov Wasserstein.
- Solvers for Numpy/Pytorch/Jax/tensorflow/Cupy

Tutorial on OT for ML:

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

Papers available on my website:

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





Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., and Peyré, G. (2015).
Iterative Bregman projections for regularized transportation problems.
SISC.



Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P.
(2017).
Geometric deep learning: going beyond euclidean data.
IEEE Signal Processing Magazine, 34(4):18–42.



Cuturi, M. (2013).
Sinkhorn distances: Lightspeed computation of optimal transportation.
In *Neural Information Processing Systems (NIPS)*, pages 2292–2300.



Frank, M. and Wolfe, P. (1956).
An algorithm for quadratic programming.
Naval research logistics quarterly, 3(1-2):95–110.



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.



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.



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.

 Thual, A., Tran, H., Zemskova, T., Courty, N., Flamary, R., Dehaene, S., and Thirion, B. (2022).

Aligning individual brains with fused unbalanced gromov-wasserstein.

In *Neural Information Processing Systems (NeurIPS)*.

 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., Flamary, R., Corneli, M., Vayer, T., and Courty, N. (2022a).
Semi-relaxed gromov wasserstein divergence with applications on graphs.
In *International Conference on Learning Representations (ICLR)*.
-  Vincent-Cuaz, C., Flamary, R., Corneli, M., Vayer, T., and Courty, N. (2022b).
Template based graph neural network with optimal transport distances.
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)*.

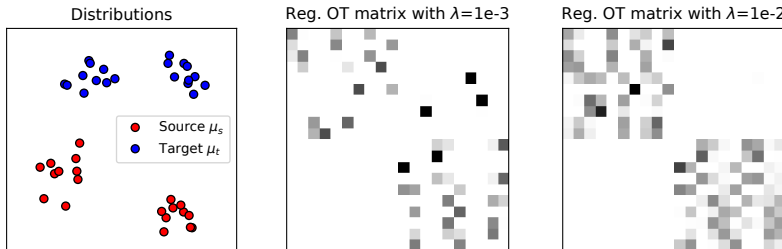


Xu, H. (2020).

Gromov-wasserstein factorization models for graph clustering.

In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6478–6485.

Entropic regularized optimal transport

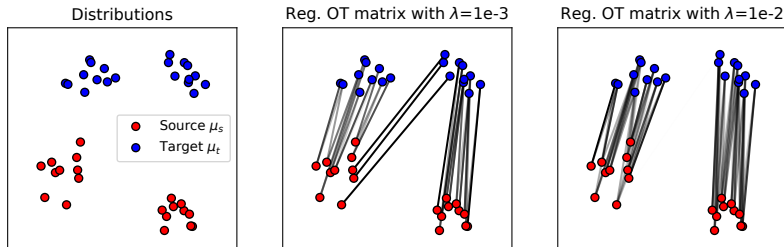


Entropic regularization [Cuturi, 2013]

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

- 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].

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].

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

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

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.

Solving the Gromov Wasserstein optimization problem

Optimization problem

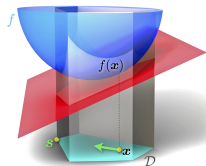
$$\mathcal{GW}_p^p(\mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \sum_{i,j,k,l} |D_{i,k} - D'_{j,l}|^p T_{i,j} T_{k,l}$$

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 $D_{i,k} = \|\mathbf{x}_i^s - \mathbf{x}_k^s\|$, $D'_{j,l} = \|\mathbf{x}_j^t - \mathbf{x}_l^t\|$

- Quadratic Program (Wasserstein is a linear program).
- Nonconvex, NP-hard, related to Quadratic Assignment Problem (QAP).
- Large problem and non convexity forbid standard QP solvers.

Optimization algorithms

- Local solution with conditional gradient algorithm (Frank-Wolfe) [Frank and Wolfe, 1956].
- Each FW iteration requires solving an OT problems.
- Gromov in 1D has a close form (solved in discrete with a sort) [Vayer et al., 2019].
- With entropic regularization, one can use mirror descent [Peyré et al., 2016] or fast low rank approximations [Scetbon et al., 2021].



Optimization Problem

$$\mathcal{GW}_{p,\epsilon}^p(\mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \sum_{i,j,k,l} |D_{i,k} - D'_{j,l}|^p T_{i,j} T_{k,l} + \epsilon \sum_{i,j} T_{i,j} \log T_{i,j} \quad (2)$$

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 $\mathcal{GW}$ [Peyré et al., 2016]

- Problem (2) 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 solved using a Sinkhorn-Knopp algorithm of entropic OT.
- Very fast approximation exist for low rank distances [Scetbon et al., 2021].

Optimization problem

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

- 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 $\mathcal{GW}_2^2(D, \sum_s w_s \overline{D_s})$, 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

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

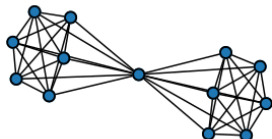
- We model the graphs as a linear model on the structure and the node weights

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

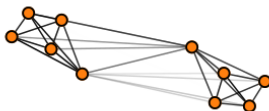
- This allows for sparse weights $\mathbf{h}$ so embedded graphs with different order.
- We provide in [Vincent-Cuaz et al., 2021] subgradients of GW *w.r.t.* the mass $\mathbf{h}$.

Experiments - Unsupervised representation learning

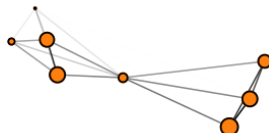
Graph from dataset



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).

Experiments - Clustering benchmark

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

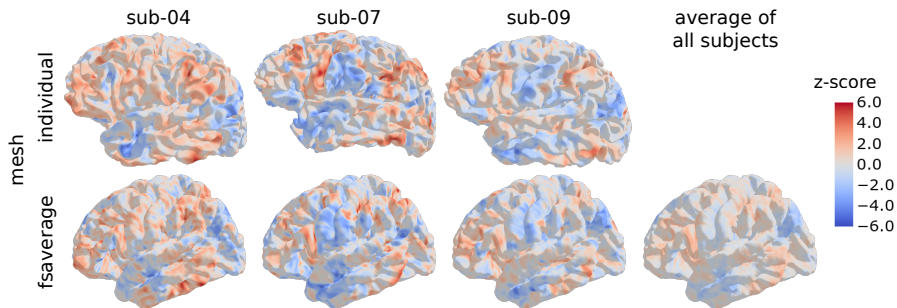
	no attribute		discrete attributes		real attributes			
models	IMDB-B	IMDB-M	MUTAG	PTC-MR	BZR	COX2	ENZYMES	PROTEIN
GDL(ours)	51.64(0.59)	55.41(0.20)	70.89(0.11)	51.90(0.54)	66.42(1.96)	59.48(0.68)	66.97(0.93)	60.49(0.71)
GWF-r	51.24 (0.02)	55.54(0.03)	-	-	52.42(2.48)	56.84(0.41)	72.13(0.19)	59.96(0.09)
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 (GWF) [Xu, 2020] and spectral clustering.
- Similar performance for supervised classification (using GW in a kernel).

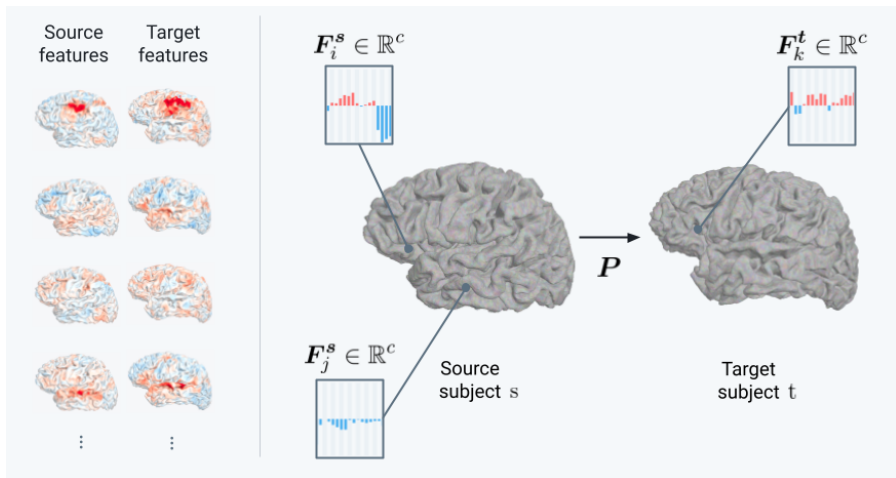
Aligning individual brains with Fused Unbalanced Gromov-Wasserstein

Inter-subject anatomical and functional variability



- Math-nonmath contrast map from the Mathlang protocol for 3 IBC subjects
- Each subject has different surfaces (mesh) and signal
- Traditional approach maps the signal on an average mesh (fsaverage bottom line) to compute an average.

Aligning individual brains with optimal transport



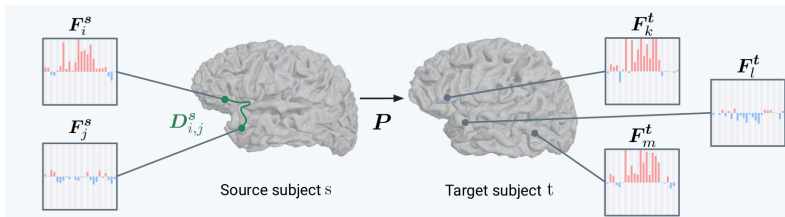
- Two subjects recorded with fMRI doing the same mental task.
- We seek an alignment preserving both the local features and the cortex geometry.
- Area can have different surface : relax marginal constraints.

Fused Unbalanced Gromov-Wasserstein

$$\begin{aligned}
 L_{\theta}(P) \triangleq & (1 - \alpha) \underbrace{\sum_{\substack{0 \leq i < n \\ 0 \leq j < p}} \|F_i^s - F_j^t\|_2^2 P_{i,j}}_{\text{Wasserstein loss } L_W(P)} + \alpha \underbrace{\sum_{\substack{0 \leq i, k < n \\ 0 \leq j, l < p}} |D_{i,k}^s - D_{j,l}^t|^2 P_{i,j} P_{k,l}}_{\text{Gromov-Wasserstein loss } L_{GW}(P)} \\
 & + \rho \left(\underbrace{\text{KL}(P_{\#1} \otimes P_{\#1} | w^s \otimes w^s) + \text{KL}(P_{\#2} \otimes P_{\#2} | w^t \otimes w^t)}_{\text{Marginal constraints } L_U(P)} \right) + \varepsilon \underbrace{E(P)}_{\text{Entropy}}
 \end{aligned}$$

Principle [Thual et al., 2022]

- Preserve the features (Wasserstein loss) and the cortex geometry (GW loss).
- Relax the marginal constraints to allows creation/destruction of mass to encore change in surface of the areas in the brain across subjects.



Features VS structure/anatomy preservation

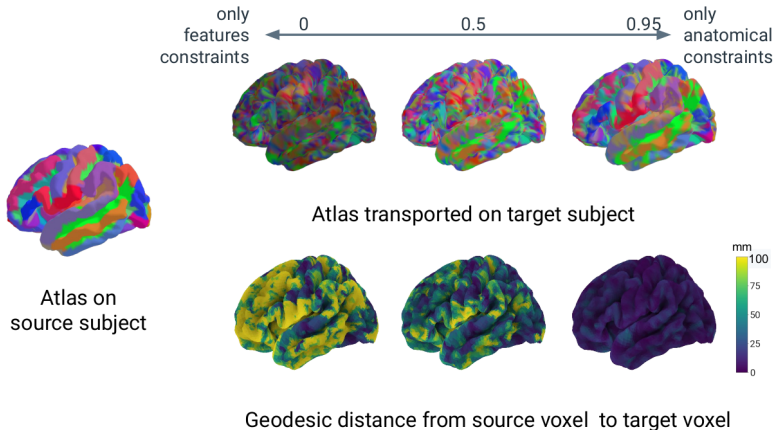
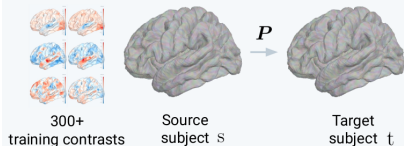


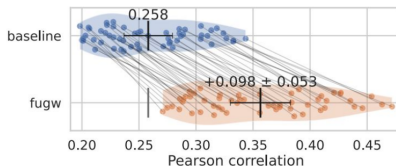
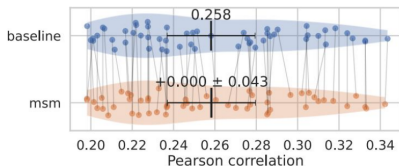
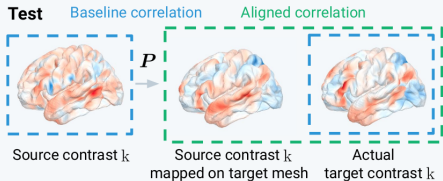
Illustration of the maps (transported atlas) and displacement on the geodesic as a function of α .

Aligning pairs of individuals with FUGW

Training (cross-validated grid-search)



Test

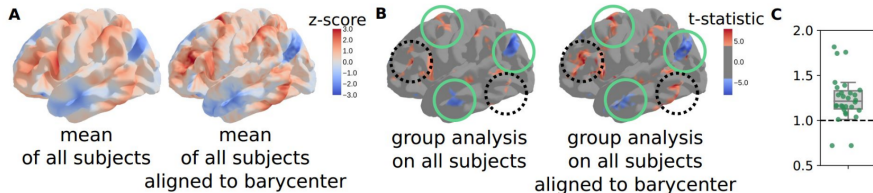


11

- Aligning with FUGW leads to significantly increased correlations across subjects.
- Similar gains on other types of stimuli and acquisition time.

Aligning individuals to a FUGW barycenter

$$\mathcal{X}^B = (F^B, D^B, w^B) \in \arg \min_{\mathcal{X}} \sum_{s \in \mathcal{S}} \text{FUGW}(\mathcal{X}^s, \mathcal{X})$$



Principle

- We compute a barycenter of different subjects (with fixed anatomy).
- FUGW barycenter significantly increases statistical power of group averages