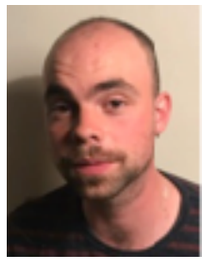
Optimal Transport for Graphs

Definitions, Applications to graph-signal processing

Nicolas Courty
Team Leader Obelix Team
ANR Chair AI OTTOPIA
PR Université Bretagne Sud / IRISA

Joint work with invaluable collaborators:



Titouan Vayer
INRIA / Dante

Rémi Flamary
CMLA / X

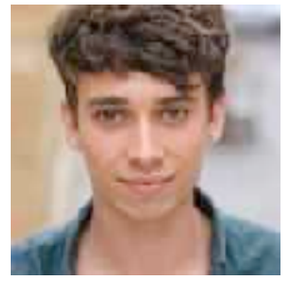
Laetitia Chapel
IRISA / Obelix

Romain Tavenard
U Rennes 2

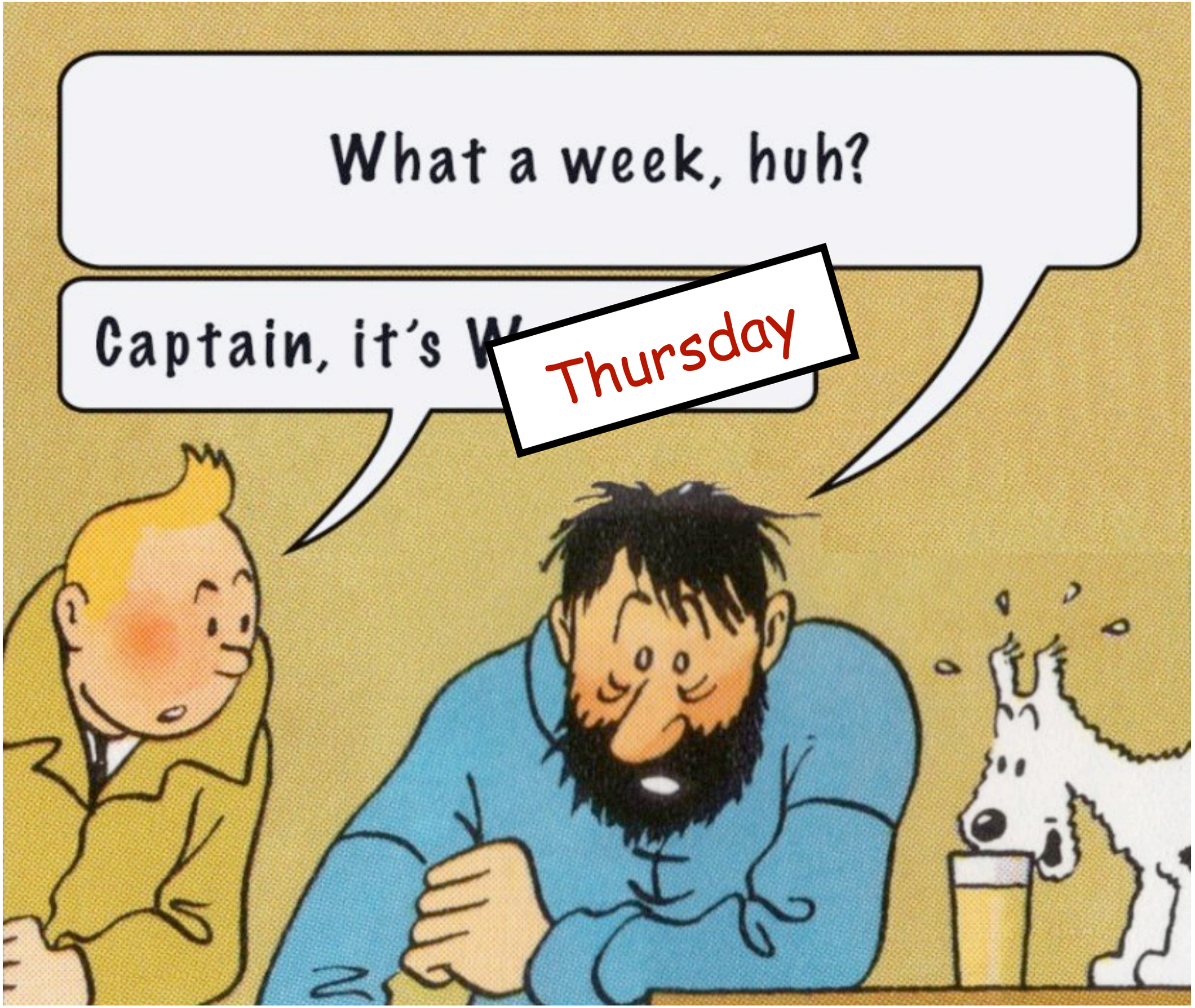
Cédric Vincent-Cuaz
INRIA / Massai

Marco Corneli
INRIA / Massai

Huy Tran
CMLA / IRISA

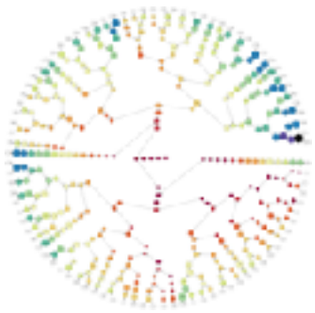


Alexis Thual
INRIA / Parietal


What a week, huh?
Captain, it's W
Thursday

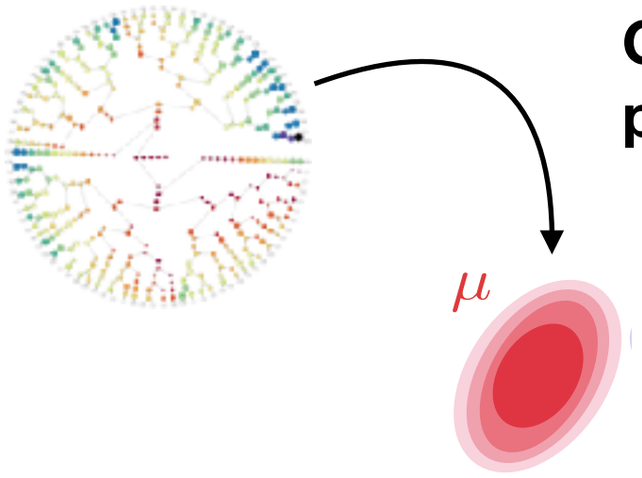
What I will discuss today

Given a graph and a signal over it



What I will discuss today

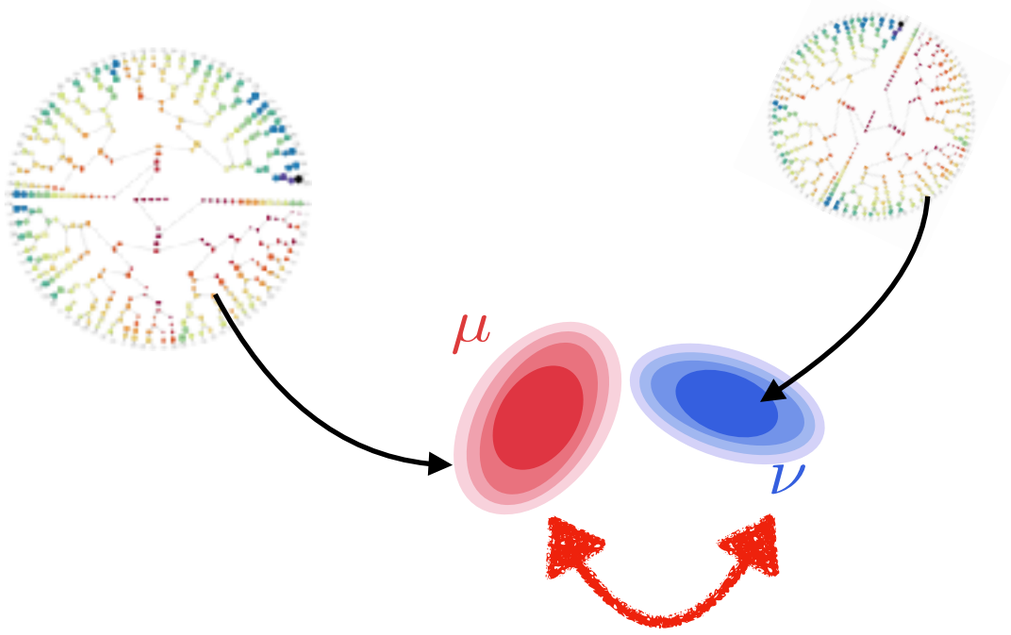
Given a graph and a signal over it



Can we represent it as some probability distribution in some metric space ?

What I will discuss today

Given two graphs and signals over them



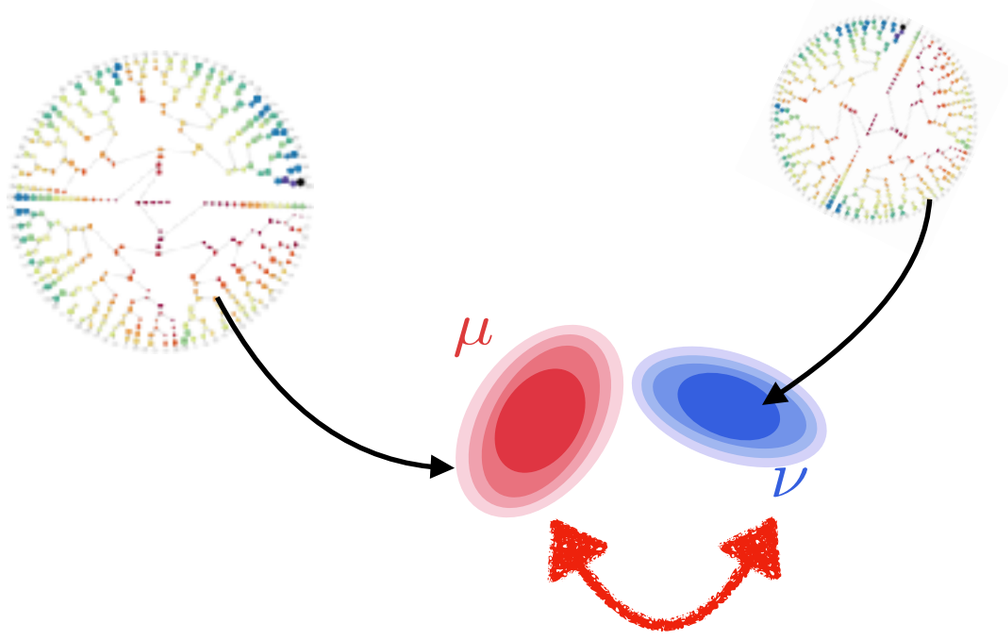
Fused Gromov Wasserstein

Can we find a suitable distance function with nice properties ?

—Optimal Transport !

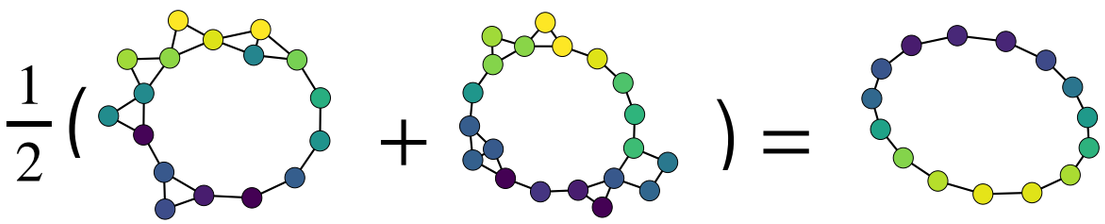
What I will discuss today

Given two graphs and signals over them

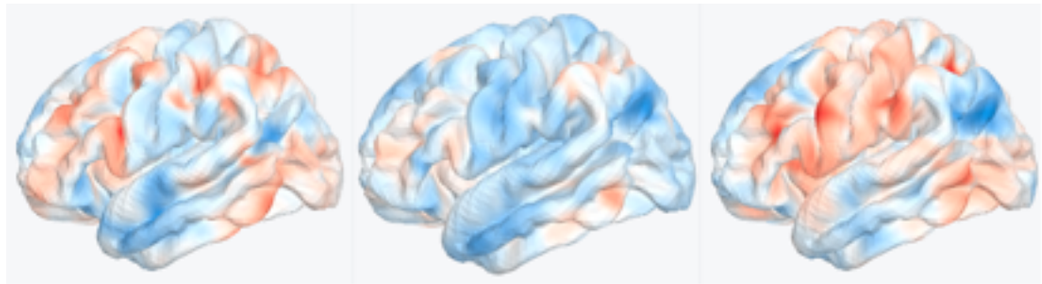


Fused Gromov Wasserstein

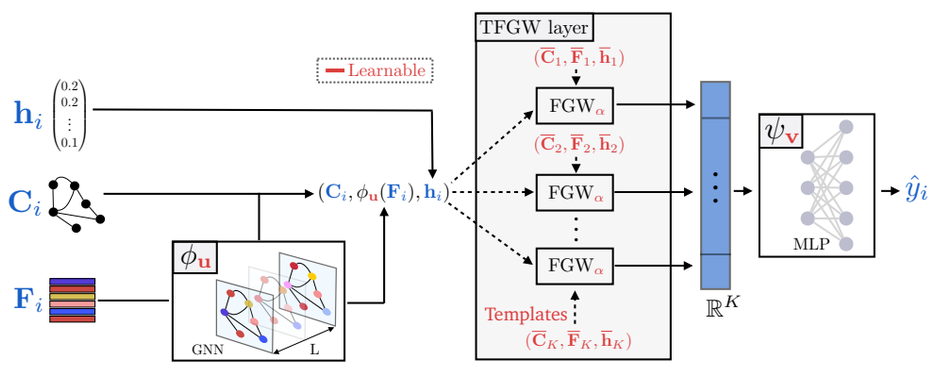
Applications



Graphs averages

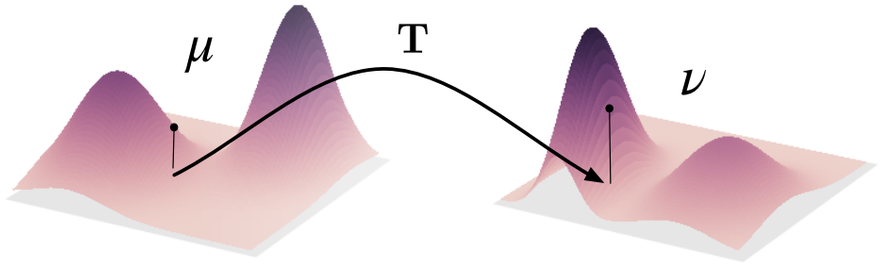


fMRI registrations

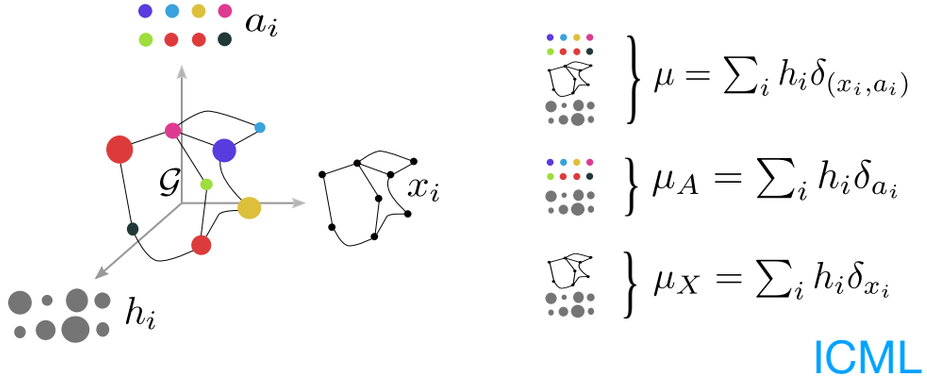


New global pooling layers for GNNs

Part I: Optimal Transport

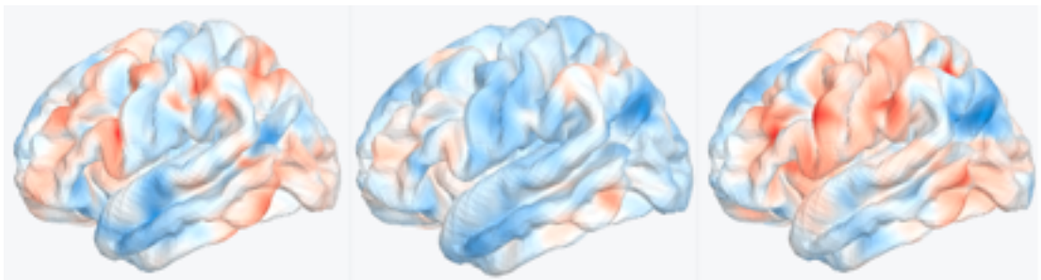


Part II: Optimal Transport for structured data



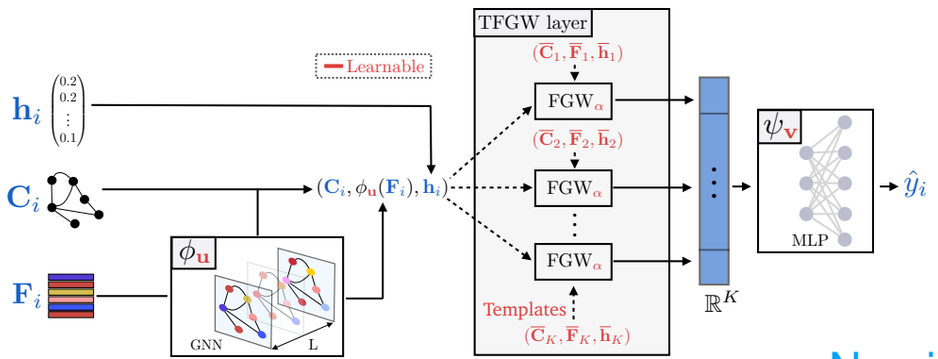
ICML 2019

Part III: Functional Brain Registration

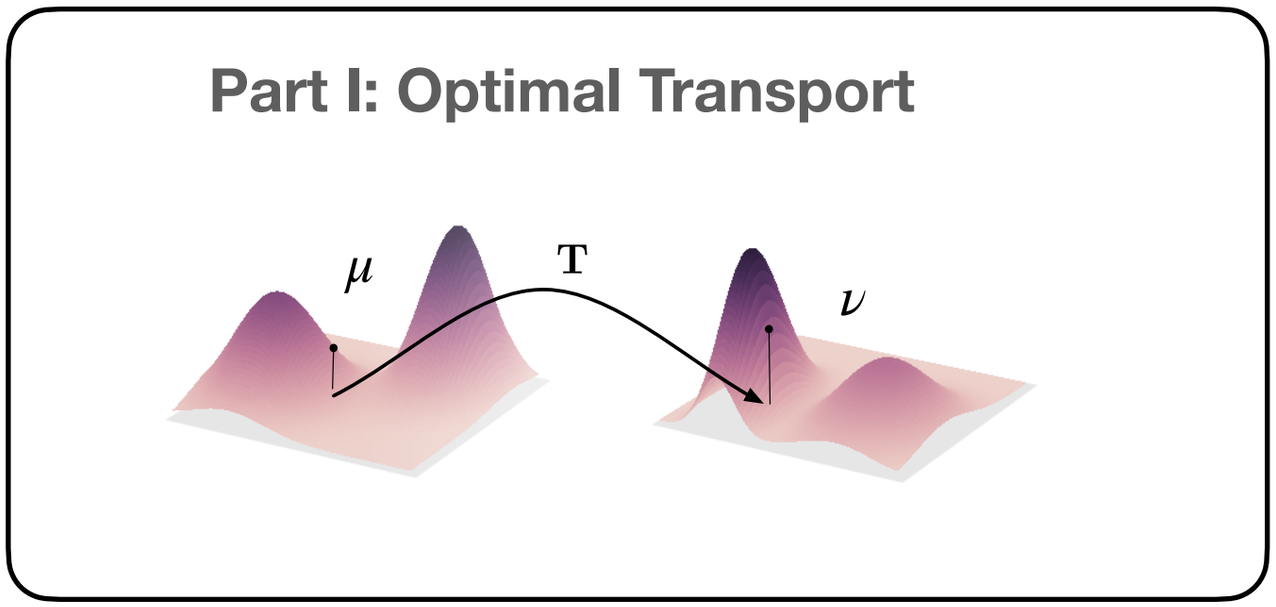


Neurips 2022

Part IV: A new Pooling Layer in GNN

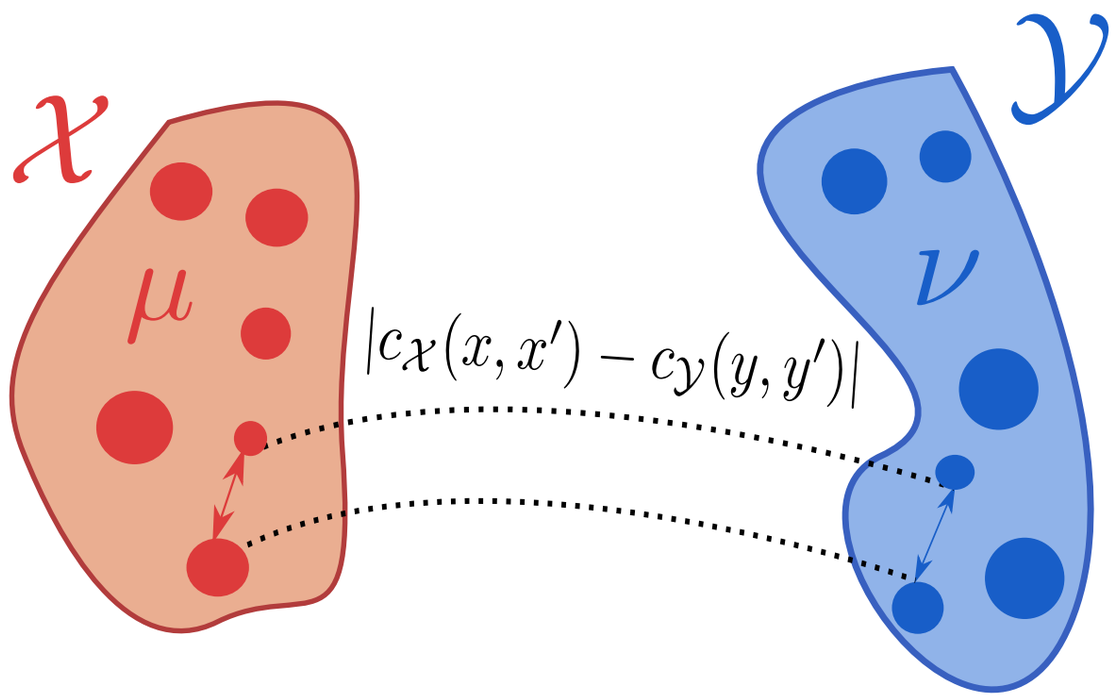


Neurips 2022



Linear to Quadratic Optimal transport

From Wasserstein to Gromov Wasserstein distance



From linear Optimal Transport...

What is it?

Input:

$$\mu \in \mathcal{P}(\mathcal{X}),\ \nu \in \mathcal{P}(\mathcal{Y})$$

Two probability distributions

From linear Optimal Transport…

What is it?

Input:

$$\mu \in \mathcal{P}(\mathcal{X}),\ \nu \in \mathcal{P}(\mathcal{Y})$$

Two probability distributions

Output:

Geometric notion of distance between these distributions

Find correspondences/relations between the samples

From linear Optimal Transport…

Why do we care about probability distributions?

> Measure and probability distributions are at the core of Machine learning

From linear Optimal Transport…

Why do we care about probability distributions?

Measure and probability distributions are at the core of Machine learning

A point of view on the data

Data: $(\mathbf{x}_i)_{i \in [\![n]\!]}$; $\mathbf{x}_i \in \mathbb{R}^d$ ⟶ A probability distribution describing the data

From linear Optimal Transport…

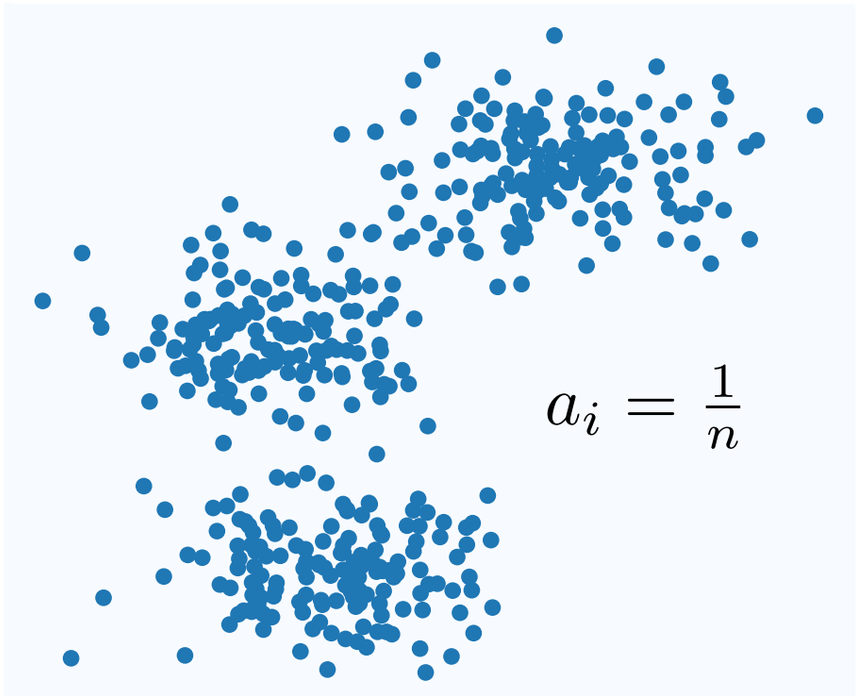
Why do we care about probability distributions?

Measure and probability distributions are at the core of Machine learning

A point of view on the data

Data: $(\mathbf{x}_i)_{i\in[\![n]\!]}$; $\mathbf{x}_i \in \mathbb{R}^d$ ⟶ A probability distribution describing the data

Lagrangian: $\sum_{i=1}^n a_i \delta_{x_i}$



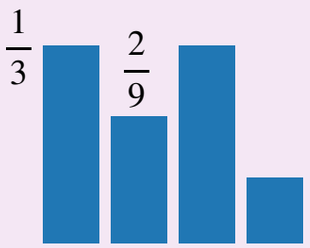
(point clouds)

$\delta_{\mathbf{x}_i}(\mathbf{x}) = 1$ if $\mathbf{x} = \mathbf{x}_i$ else 0

Probability simplex

$\mathbf{a} = (a_i)_{i\in[\![n]\!]} \in \Sigma_n$

$a_i \geq 0, \sum_{i=1}^n a_i = 1$



From linear Optimal Transport…

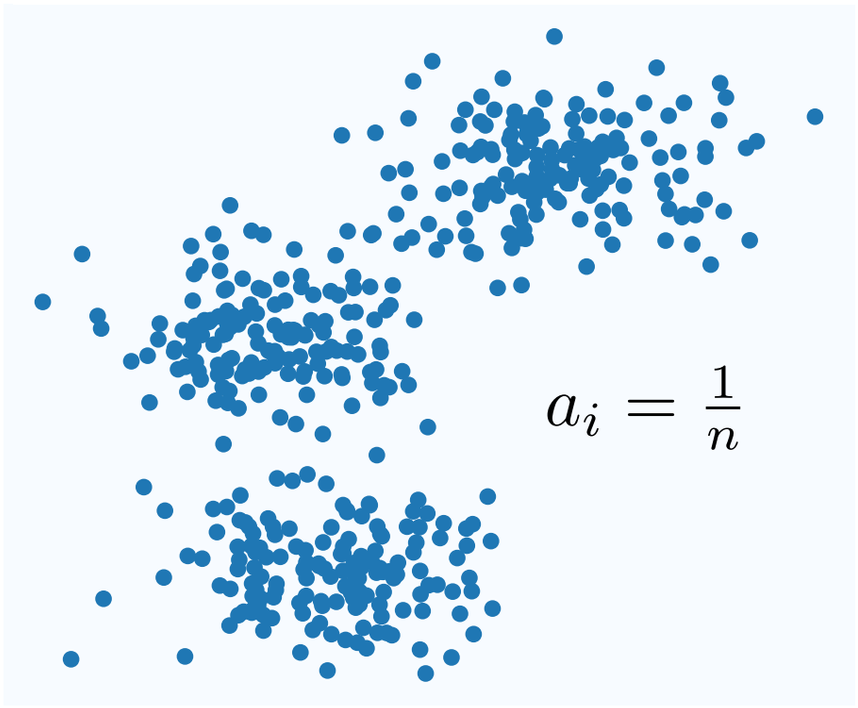
Why do we care about probability distributions?

Measure and probability distributions are at the core of Machine learning

A point of view on the data

Data: $(\mathbf{x}_i)_{i\in[\![n]\!]}$; $\mathbf{x}_i \in \mathbb{R}^d$ ⟶ A probability distribution describing the data

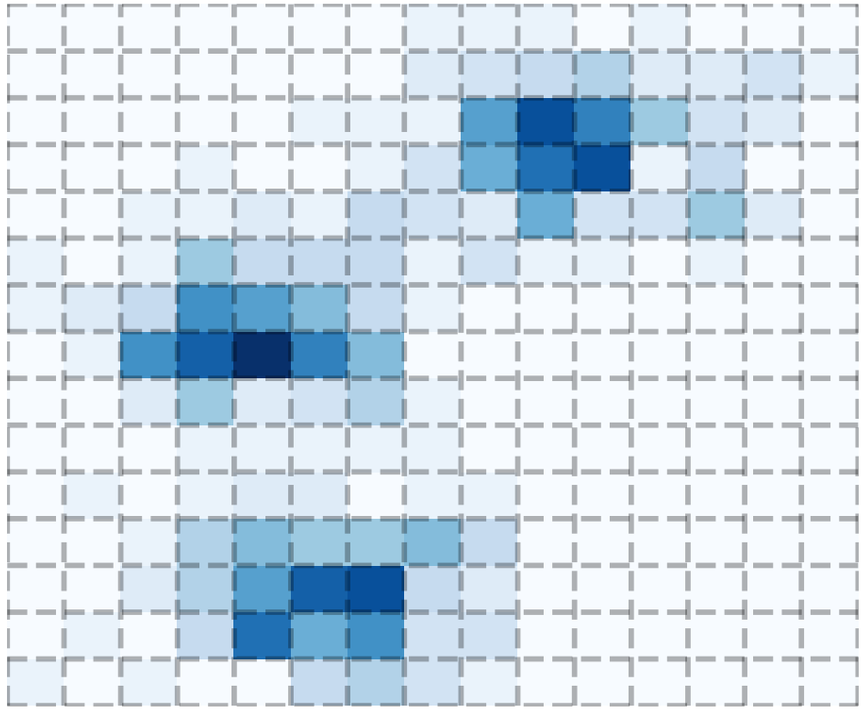
Lagrangian: $\sum_{i=1}^n a_i \delta_{x_i}$



(point clouds)

$\delta_{\mathbf{x}_i}(\mathbf{x}) = 1$ if $\mathbf{x} = \mathbf{x}_i$ else 0

Eulerian: $\sum_{i=1}^N a_i \delta_{\hat{x}_i}$



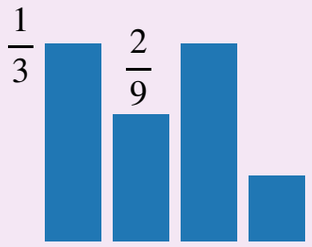
(histograms)

$\hat{x}_i$ fixed position (grid)

Probability simplex

$\mathbf{a} = (a_i)_{i\in[\![n]\!]} \in \Sigma_n$

$a_i \geq 0, \sum_{i=1}^n a_i = 1$



From linear Optimal Transport…

Formulation

Two probability distributions

$$\mu \in \mathcal{P}(\mathcal{X}),\ \nu \in \mathcal{P}(\mathcal{Y})$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Optimal Transport

From linear Optimal Transport…

Kantorovitch Formulation

Two probability distributions

$$\mu \in \mathcal{P}(\mathcal{X}),\ \nu \in \mathcal{P}(\mathcal{Y})$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Optimal Transport

All the mass of μ is **transported** to ν by a **transport plan** $\pi \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$

From linear Optimal Transport…

Kantorovitch Formulation

Two probability distributions

$$\mu \in \mathcal{P}(\mathcal{X}),\ \nu \in \mathcal{P}(\mathcal{Y})$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Optimal Transport

All the mass of μ is **transported** to ν by a **transport plan** $\pi \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$

We want to find the plan that **minimizes the overall cost** of moving all the points

From linear Optimal Transport…

Kantorovitch Formulation: an example

Two probability distributions

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Bakeries = quantity of breads

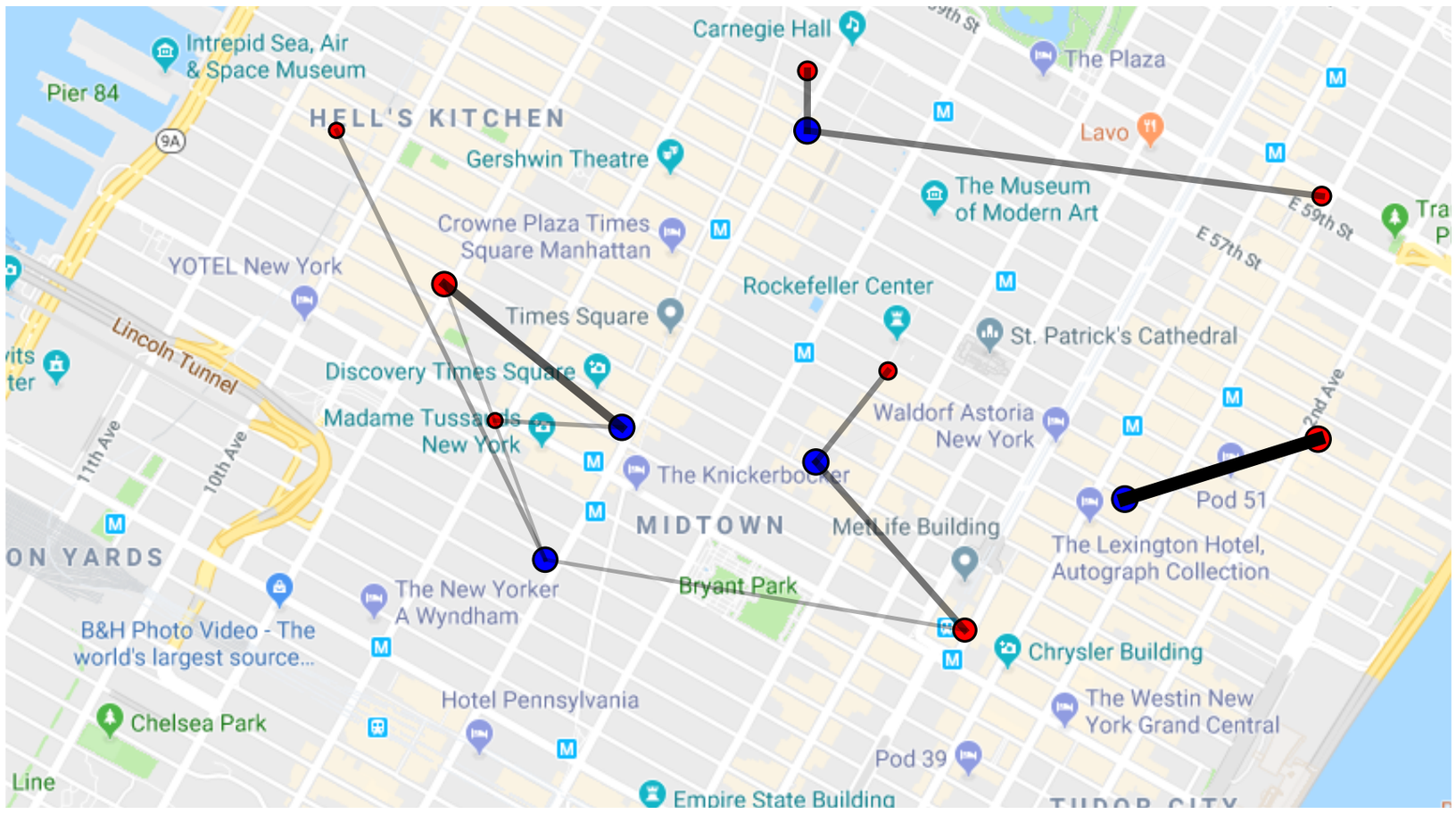
loc: x_i quantity: a_i

Cafés = demand of breads

loc: y_j demand: b_j

Distance between bakeries and cafés

$$c(x_i, y_j)$$



We want to route all the breads from bakeries to cafés the cheapest way

From linear Optimal Transport…

Kantorovitch Formulation: an example

Two probability distributions

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Kantorovitch formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j)\pi_{ij}$$

From linear Optimal Transport…

Kantorovitch Formulation: an example

Two probability distributions

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Kantorovitch formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j)\pi_{ij}$$

Set of couplings/ transport plans

$$\Pi(\mathbf{a}, \mathbf{b})$$

From linear Optimal Transport…

Kantorovitch Formulation: an example

Two probability distributions

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Kantorovitch formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j) \pi_{ij}$$

How much is shifted from x_i to y_j

From linear Optimal Transport…

Kantorovitch Formulation: an example

Two probability distributions

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Kantorovitch formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j)\pi_{ij}$$

Cost of moving masses from x_i to y_j

From linear Optimal Transport…

Kantorovitch Formulation: an example

Two probability distributions

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Kantorovitch formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j)\pi_{ij}$$

Total cost

From linear Optimal Transport…

Kantorovitch Formulation: an example

Two probability distributions

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

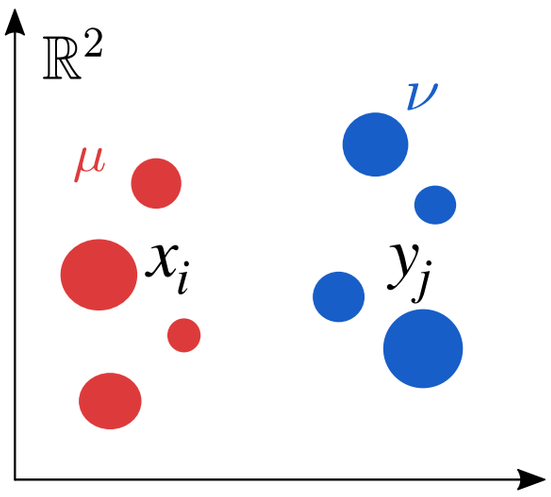
A cost function

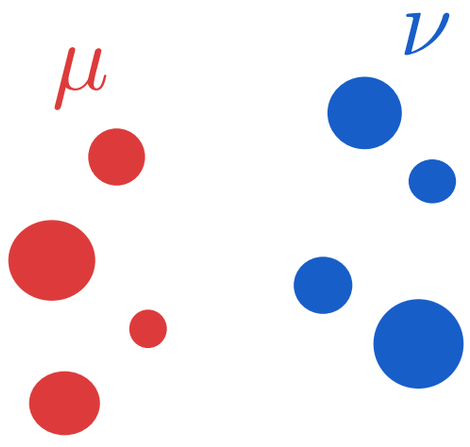
$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Kantorovitch formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j)\pi_{ij}$$

$$\Pi(\mathbf{a}, \mathbf{b}) = \{\boldsymbol{\pi} \in \mathbb{R}_+^{n \times m} \mid \forall (i,j), \sum_{j=1}^{m} \pi_{ij} = a_i, \ \sum_{i=1}^{n} \pi_{ij} = b_j\}$$





From linear Optimal Transport…

Kantorovitch Formulation: an example

Two probability distributions

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

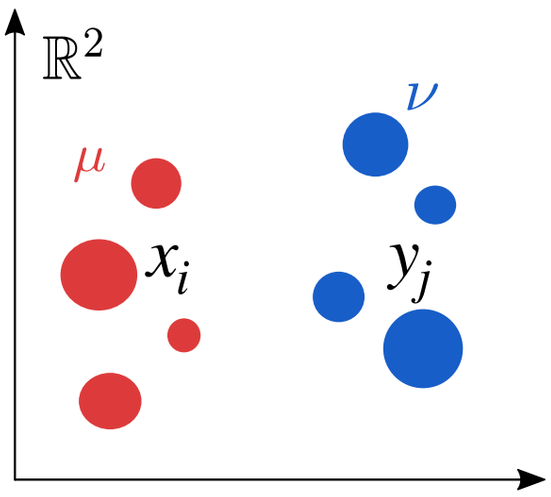
A cost function

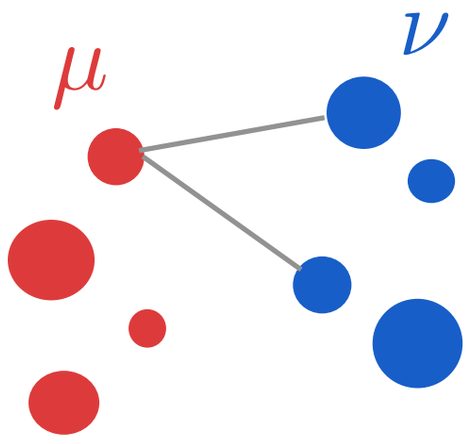
$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Kantorovitch formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j)\pi_{ij}$$

$$\Pi(\mathbf{a}, \mathbf{b}) = \{\boldsymbol{\pi} \in \mathbb{R}_+^{n \times m} \mid \forall (i,j), \sum_{j=1}^{m} \pi_{ij} = a_i, \ \sum_{i=1}^{n} \pi_{ij} = b_j\}$$





From linear Optimal Transport…

Kantorovitch Formulation: an example

Two probability distributions

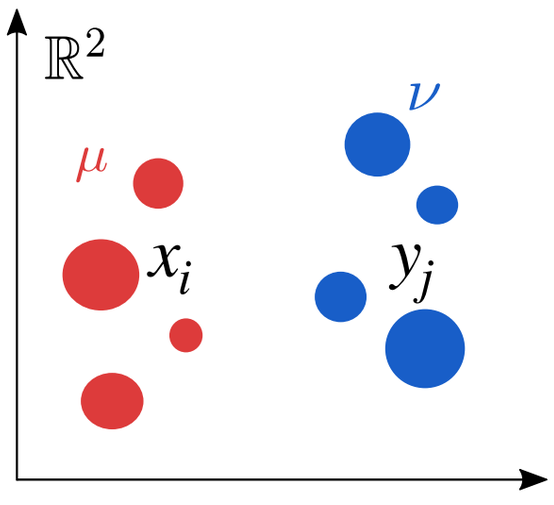
$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

A cost function

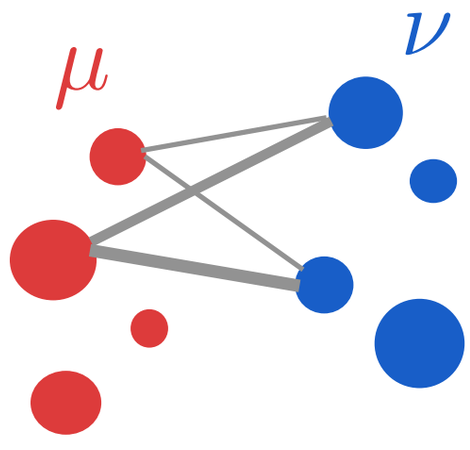
$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Kantorovitch formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j)\pi_{ij}$$



$$\Pi(\mathbf{a}, \mathbf{b}) = \{\boldsymbol{\pi} \in \mathbb{R}_+^{n \times m} \mid \forall (i,j), \sum_{j=1}^{m} \pi_{ij} = a_i, \ \sum_{i=1}^{n} \pi_{ij} = b_j\}$$



From linear Optimal Transport…

Kantorovitch Formulation: an example

Two probability distributions

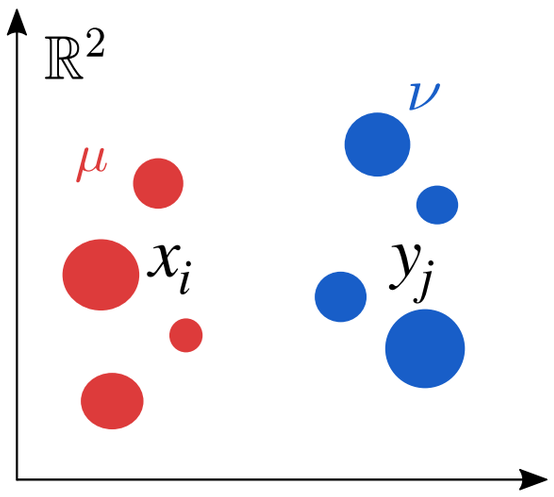
$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

A cost function

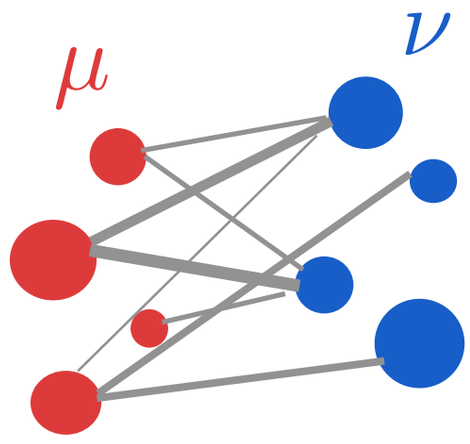
$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

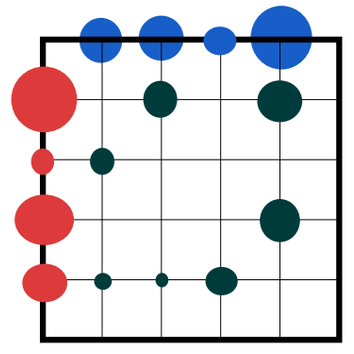
Kantorovitch formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j)\pi_{ij}$$



$$\Pi(\mathbf{a}, \mathbf{b}) = \{\boldsymbol{\pi} \in \mathbb{R}_+^{n \times m} \mid \forall (i,j), \sum_{j=1}^{m} \pi_{ij} = a_i, \ \sum_{i=1}^{n} \pi_{ij} = b_j\}$$





$\pi \in \mathbb{R}_+^{n \times m}$

From linear Optimal Transport...

Kantorovitch Formulation: general case

Two probability distributions

$$\mu \in \mathcal{P}(\mathcal{X}),\ \nu \in \mathcal{P}(\mathcal{Y})$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Kantorovitch formulation

$$\mathcal{T}_c(\mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y) \mathrm{d}\pi(x, y)$$

From linear Optimal Transport…

Wasserstein distance

Two probability distributions

$$\mu \in \mathcal{P}(\Omega), \nu \in \mathcal{P}(\Omega)$$

A distance

$$d : \Omega \times \Omega \to \mathbb{R}_+$$

Example: $\Omega = \mathbb{R}^d$

Wasserstein distance

$$W_p^p(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\Omega \times \Omega} d^p(x, y) \mathrm{d}\pi(x, y)$$

$\mathcal{P}(\Omega)$ is a metric space

$W_p(\mu, \nu) = 0 \iff \mu = \nu$

From linear Optimal Transport…

Wasserstein distance

Two probability distributions

$$\mu \in \mathcal{P}(\Omega), \nu \in \mathcal{P}(\Omega)$$

A distance

$$d : \Omega \times \Omega \to \mathbb{R}_+$$

Example: $\Omega = \mathbb{R}^d$

Wasserstein distance

$$W_p^p(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\Omega \times \Omega} d^p(x, y) \mathrm{d}\pi(x, y)$$

$\mathcal{P}(\Omega)$ is a metric space

$W_p(\mu, \nu) = 0 \iff \mu = \nu$

Powerful tool for comparing probability distributions on the same space

...to Gromov-Wasserstein

What if ?

Data are in Incomparable spaces

Two probability distributions

$$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y}) \text{ with } \mathcal{X}, \mathcal{Y} \nsubseteq \Omega$$

A cost function ?????

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

⇨ Not straightforward to find a suitable cost (e.g. no distance available)

...to Gromov-Wasserstein

What if ?

Data are in Incomparable spaces

Two probability distributions

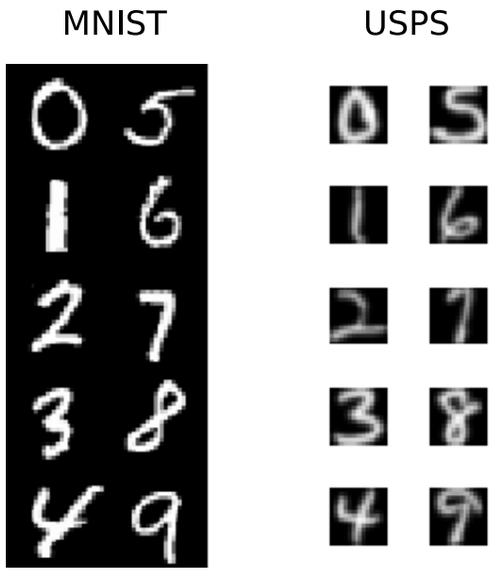
$$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y}) \text{ with } \mathcal{X}, \mathcal{Y} \nsubseteq \Omega$$

A cost function ?????

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

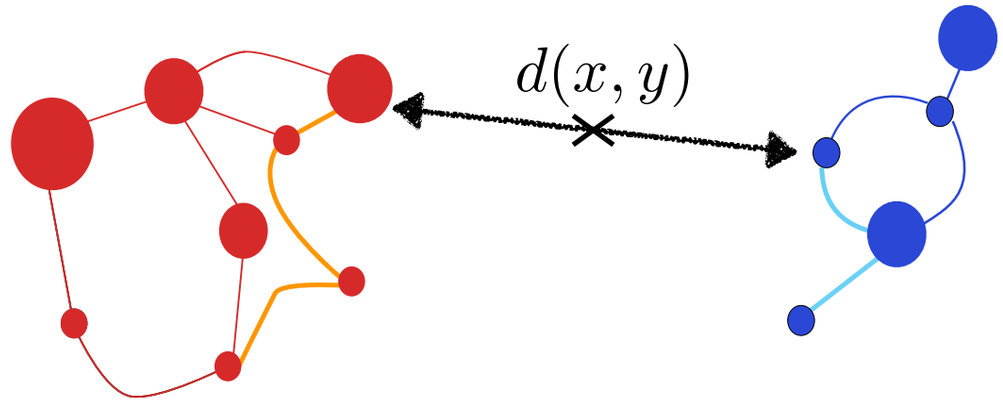
⇨ Not straightforward to find a suitable cost (e.g. no distance available)

Different Euclidean spaces



Example: $\mathcal{X} = \mathbb{R}^{28*28}, \mathcal{Y} = \mathbb{R}^{16*16}$

Samples = nodes of different graphs



Example: $\mathcal{X} = \text{Graph } 1, \mathcal{Y} = \text{Graph } 2$

...to Gromov-Wasserstein

Gromov-Wasserstein distance

Two probability distributions

$$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y})$$

Two « intra-domain » costs

$$c_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$
$$c_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$

Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left|c_{\mathcal{X}}(x,x') - c_{\mathcal{Y}}(y,y')\right|^p \mathrm{d}\pi(x,y)\mathrm{d}\pi(x',y'))$$

…to Gromov-Wasserstein

Gromov-Wasserstein distance

Two probability distributions

$$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y})$$

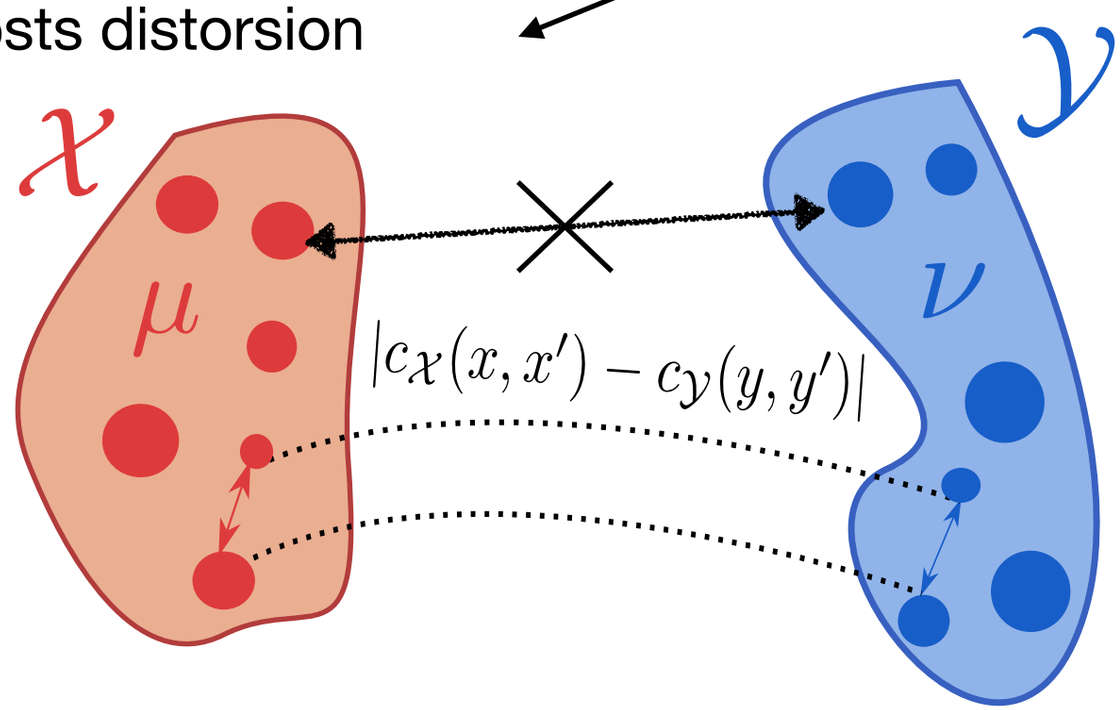
Two « intra-domain » costs

$$c_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$
$$c_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$

Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left|c_{\mathcal{X}}(x,x') - c_{\mathcal{Y}}(y,y')\right|^p \mathrm{d}\pi(x,y)\mathrm{d}\pi(x',y'))$$

Measure the costs distorsion



...to Gromov-Wasserstein

Gromov-Wasserstein distance

Two probability distributions

$$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y})$$

Two « intra-domain » costs

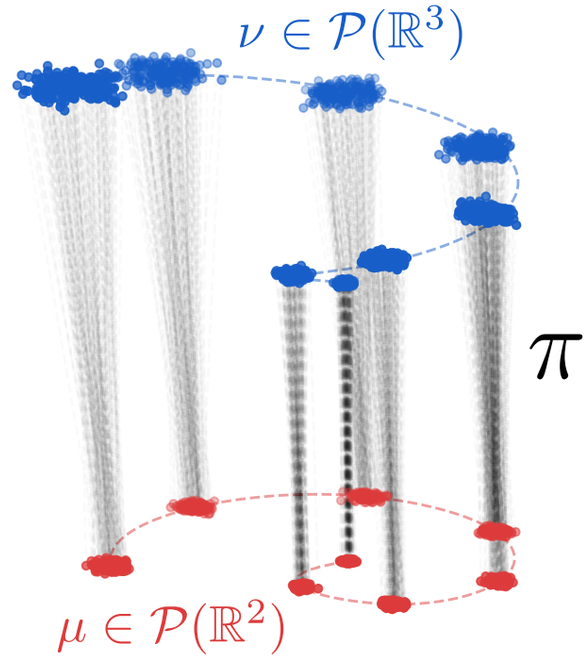
$$c_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$
$$c_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$

Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left|c_{\mathcal{X}}(x,x') - c_{\mathcal{Y}}(y,y')\right|^p \mathrm{d}\pi(x,y)\mathrm{d}\pi(x',y'))$$

The transportation problem is not linear anymore but **quadratic**

Associate pair of points with similar costs in each space



...to Gromov-Wasserstein

Gromov-Wasserstein distance

Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left|c_{\mathcal{X}}(x,x') - c_{\mathcal{Y}}(y,y')\right|^p \mathrm{d}\pi(x,y)\mathrm{d}\pi(x',y'))$$

A distance w.r.t isomorphism

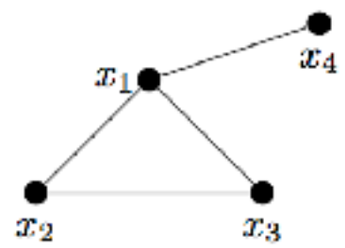
GW is a distance on the "space of all spaces":

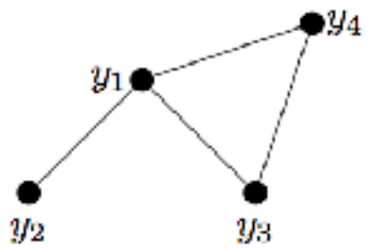
$$\mathbb{X} = \{(\mathcal{X}, d_{\mathcal{X}}, \mu \in \mathcal{P}(\mathcal{X})); d_{\mathcal{X}} \text{ metric } \} \text{ (mm-spaces)}$$

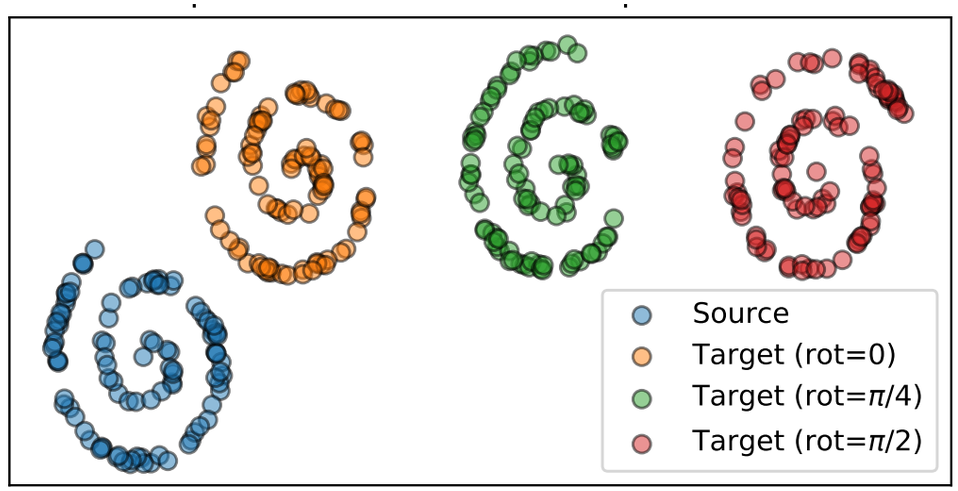
- $GW_p(d_{\mathcal{X}}, d_{\mathcal{Y}}, \mu, \nu) = 0$ iff $\exists \phi : \mathcal{X} \to \mathcal{Y}$

  ϕ is a isometry $d_{\mathcal{X}}(x,x') = d_{\mathcal{Y}}(\phi(x), \phi(x'))$

Isometry: permutations, rotations, translations,...







...to Gromov-Wasserstein

Gromov-Wasserstein distance


Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left|c_{\mathcal{X}}(x,x') - c_{\mathcal{Y}}(y,y')\right|^p \mathrm{d}\pi(x,y)\mathrm{d}\pi(x',y'))$$

A distance w.r.t isomorphism

GW is a distance on the "space of all spaces":

$\mathbb{X} = \{(\mathcal{X}, d_{\mathcal{X}}, \mu \in \mathcal{P}(\mathcal{X})); d_{\mathcal{X}} \text{ metric } \}$ (mm-spaces)

- $GW_p(d_{\mathcal{X}}, d_{\mathcal{Y}}, \mu, \nu) = 0$ iff $\exists \phi : \mathcal{X} \to \mathcal{Y}$

  ϕ is a isometry $d_{\mathcal{X}}(x,x') = d_{\mathcal{Y}}(\phi(x), \phi(x'))$

  ϕ is measure-preserving: $\phi \# \mu = \nu$

Push-forward $\phi \# \mu$

$$\mu = \sum_{i=1}^n a_i \delta_{x_i} \underset{\phi\#\mu}{\to} \sum_{i=1}^n a_i \delta_{\phi(x_i)}$$

...to Gromov-Wasserstein

Gromov-Wasserstein distance

Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left|c_{\mathcal{X}}(x,x') - c_{\mathcal{Y}}(y,y')\right|^p \mathrm{d}\pi(x,y)\mathrm{d}\pi(x',y'))$$

A distance w.r.t isomorphism

GW is a distance on the "space of all spaces":

$$\mathbb{X} = \{(\mathcal{X}, d_{\mathcal{X}}, \mu \in \mathcal{P}(\mathcal{X})); d_{\mathcal{X}} \text{ metric } \} \text{ (mm-spaces)}$$

- $GW_p(d_{\mathcal{X}}, d_{\mathcal{Y}}, \mu, \nu) = 0$ iff $\exists \phi : \mathcal{X} \to \mathcal{Y}$

  ϕ is a isometry $d_{\mathcal{X}}(x,x') = d_{\mathcal{Y}}(\phi(x), \phi(x'))$

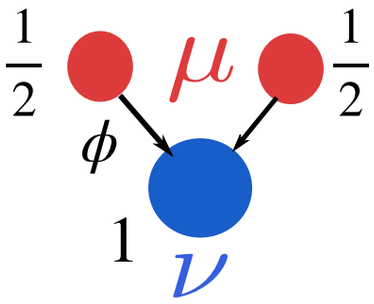
  ϕ is measure-preserving: $\phi \# \mu = \nu$

(Weights are compatible)

Push-forward $\phi \# \mu$

$$\mu = \sum_{i=1}^n a_i \delta_{x_i} \underset{\phi\#\mu}{\to} \sum_{i=1}^n a_i \delta_{\phi(x_i)}$$

Compatible



$$\frac{1}{2} + \frac{1}{2} \to 1$$

...to Gromov-Wasserstein

Gromov-Wasserstein distance


Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left|c_{\mathcal{X}}(x,x') - c_{\mathcal{Y}}(y,y')\right|^p \mathrm{d}\pi(x,y)\mathrm{d}\pi(x',y'))$$

A distance w.r.t isomorphism

GW is a distance on the "space of all spaces":

$$\mathbb{X} = \{(\mathcal{X}, d_{\mathcal{X}}, \mu \in \mathcal{P}(\mathcal{X})); d_{\mathcal{X}} \text{ metric } \} \text{ (mm-spaces)}$$

- $GW_p(d_{\mathcal{X}}, d_{\mathcal{Y}}, \mu, \nu) = 0$ iff $\exists \phi : \mathcal{X} \to \mathcal{Y}$

  ϕ is a isometry $d_{\mathcal{X}}(x,x') = d_{\mathcal{Y}}(\phi(x), \phi(x'))$

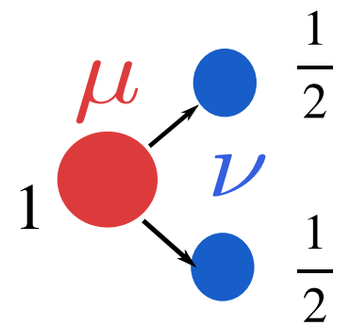
  ϕ is measure-preserving: $\phi\#\mu = \nu$

(Weights are compatible)

Push-forward $\phi\#\mu$

$$\mu = \sum_{i=1}^n a_i \delta_{x_i} \underset{\phi\#\mu}{\to} \sum_{i=1}^n a_i \delta_{\phi(x_i)}$$

Not compatible



$$1 \nrightarrow (\frac{1}{2}, \frac{1}{2})$$

...to Gromov-Wasserstein

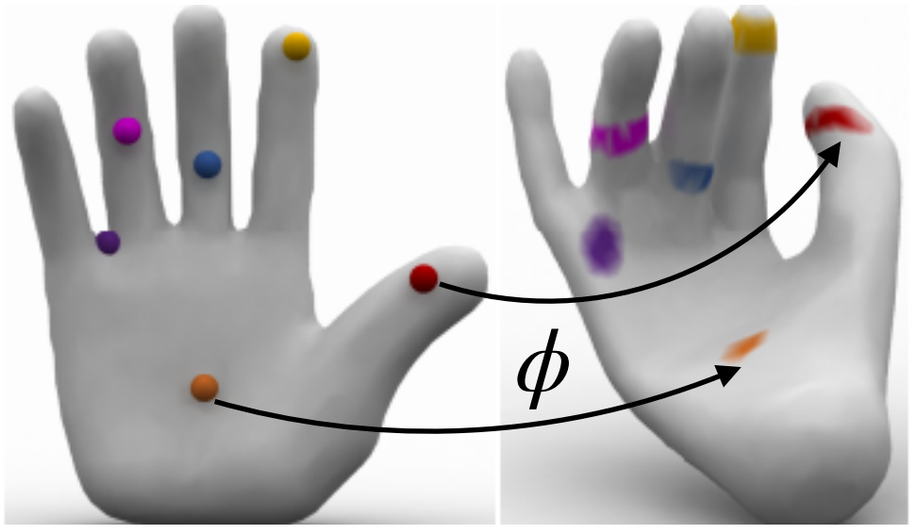
Gromov-Wasserstein distance

Gromov-Wasserstein = a bending invariant distance

- $GW_p(d_{\mathcal{X}}, d_{\mathcal{Y}}, \mu, \nu) = 0$ iff $\exists \phi : \mathcal{X} \to \mathcal{Y}$

  ϕ is a isometry $d_{\mathcal{X}}(x, x') = d_{\mathcal{Y}}(\phi(x), \phi(x'))$

  ϕ is measure-preserving $\phi \# \mu = \nu$

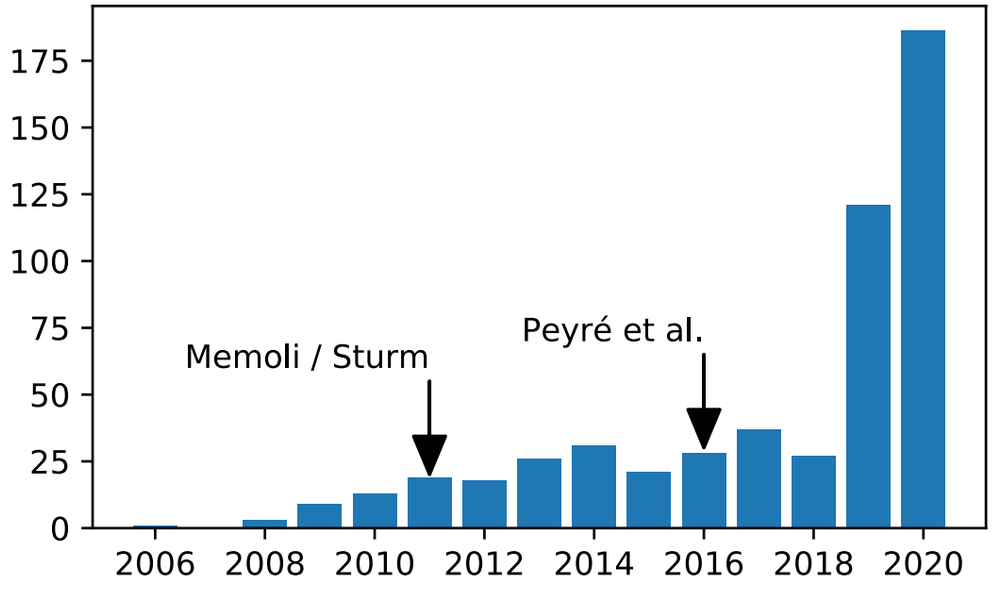


[Solomon 2016]

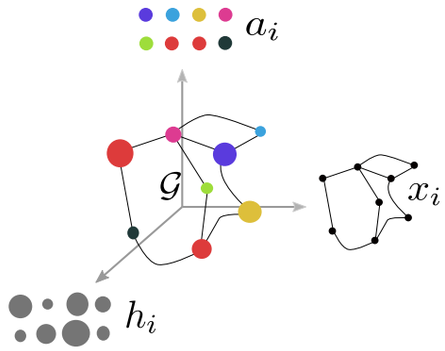
Applications for geometric data

- Barycenter of relational data [Peyré 2016], Point clouds/meshes [Ezuz 2017]
- Shape comparison [Mémoli 2011, Solomon 2016]
- Graphs [Titouan 2019 lol, Xu 2019, Fey 2020], biology [Demetci 2020], generative modeling [Bunne 2019]

Occurences Gromov-Wasserstein in Google Scholar



Part II: Optimal Transport for structured data



$\mu = \sum_i h_i \delta_{(x_i, a_i)}$

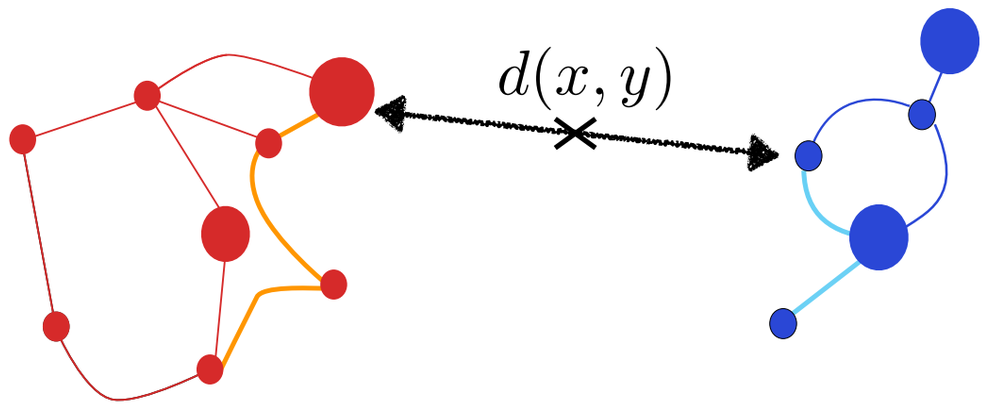
$\mu_A = \sum_i h_i \delta_{a_i}$

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

ICML 2019

Optimal transport for structured data

Fused Gromov Wasserstein



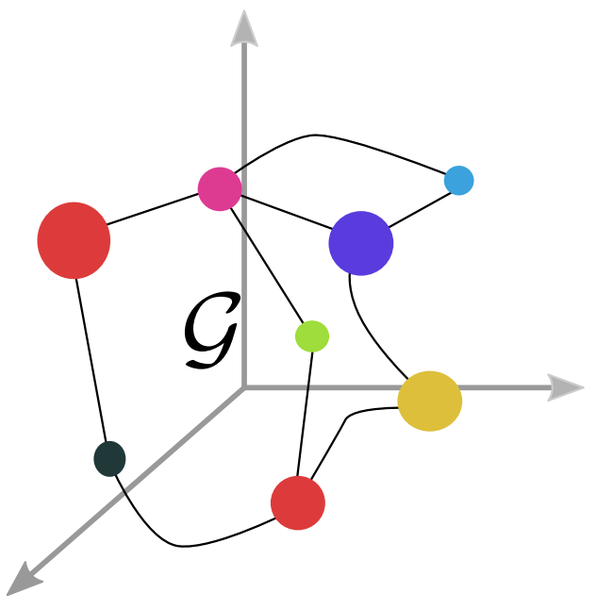
Optimal Transport for structured data

Structured data as probability distribution

Discrete case

Structured data can be seen as a labeled graph

Combines a feature **and** a structure information



Optimal Transport for structured data

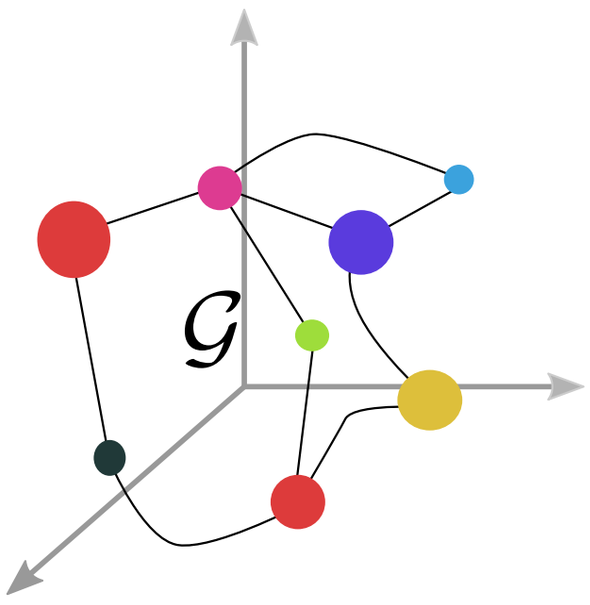
Structured data as probability distribution

Discrete case

Structured data can be seen as a labeled graph

Combines a feature **and** a structure information

Features  $a_i \in \Omega$



Optimal Transport for structured data

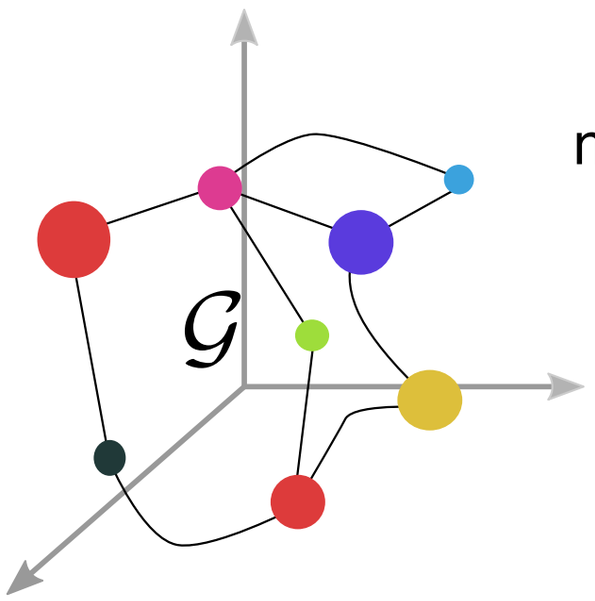
Structured data as probability distribution

Discrete case

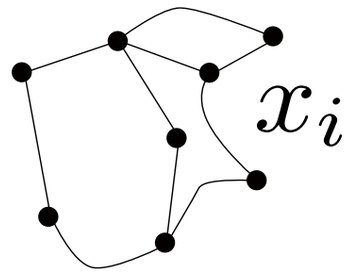
Structured data can be seen as a labeled graph

Combines a feature **and** a structure information

Features  $a_i \in \Omega$



Structure: nodes in the metric space of the graph



Optimal Transport for structured data

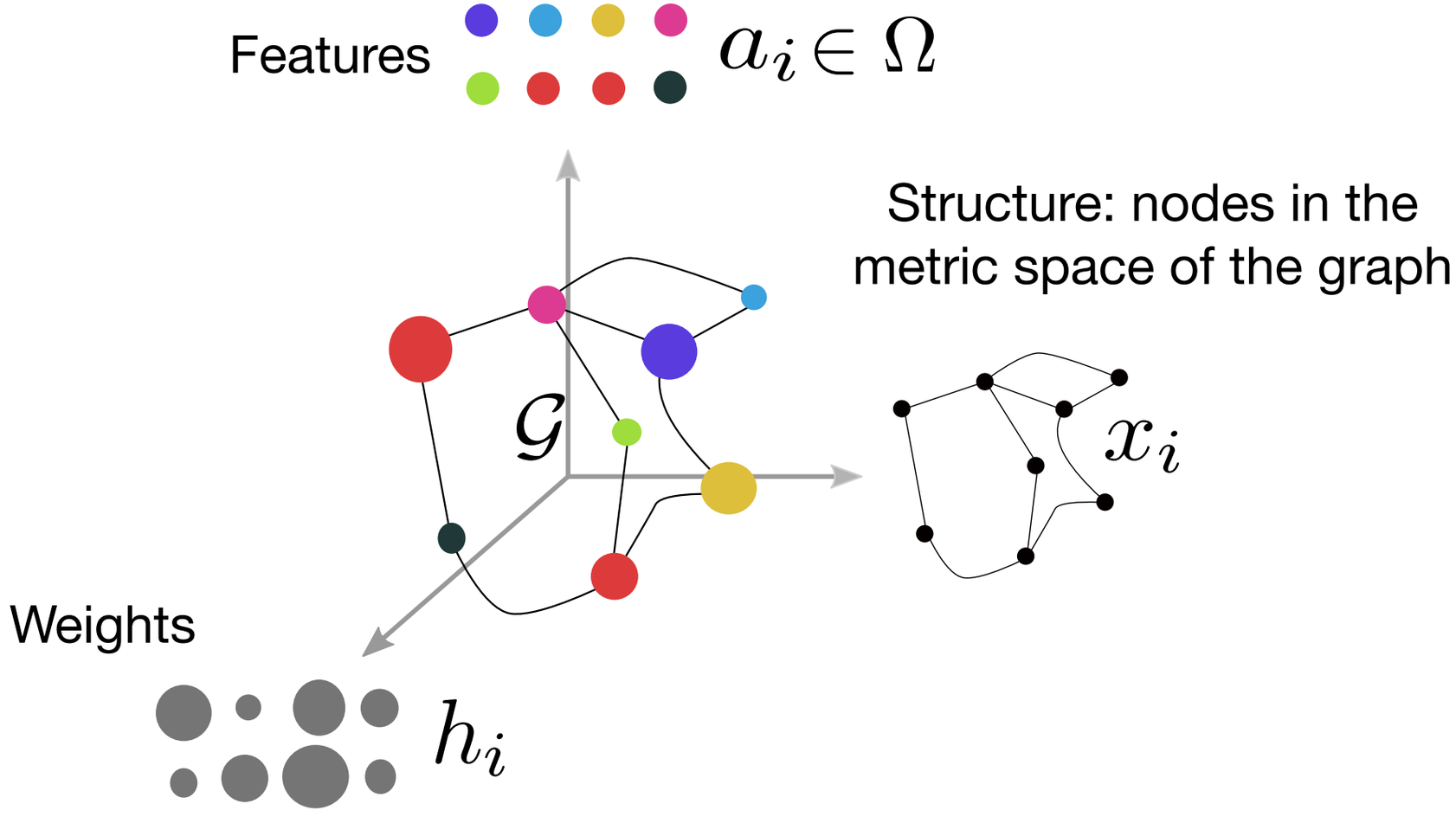
Structured data as probability distribution

Discrete case

Structured data can be seen as a labeled graph

Combines a feature **and** a structure information

Add weights that encodes the relative importance of the nodes



Optimal Transport for structured data

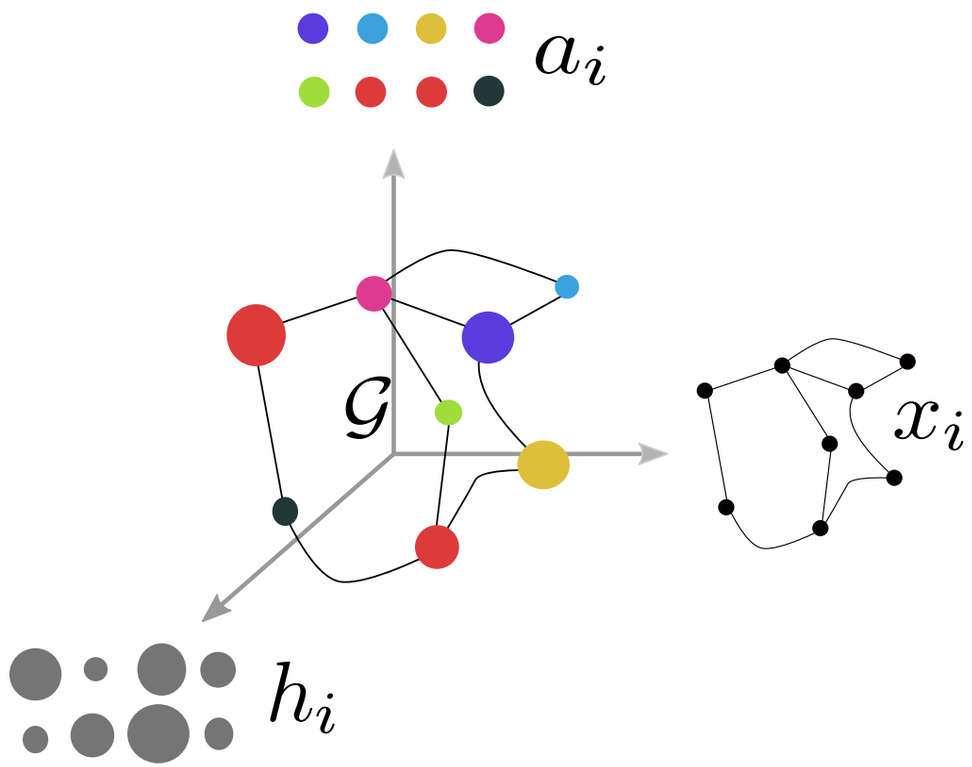
Structured data as probability distribution

Discrete case

Structured data can be seen as a labeled graph

Combines a feature **and** a structure information

Add weights that encodes the relative importance of the nodes



Form a probability measure

$$\mu = \sum_i h_i \delta_{(x_i, a_i)}$$

$$\mu_A = \sum_i h_i \delta_{a_i}$$

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

Optimal Transport for structured data

Fused Gromov-Wasserstein distance

Two structured data

$\mu = \sum_i h_i \delta_{(x_i, a_i)}, \nu = \sum_j g_j \delta_{(y_j, b_j)}$

Two matrices describing structures

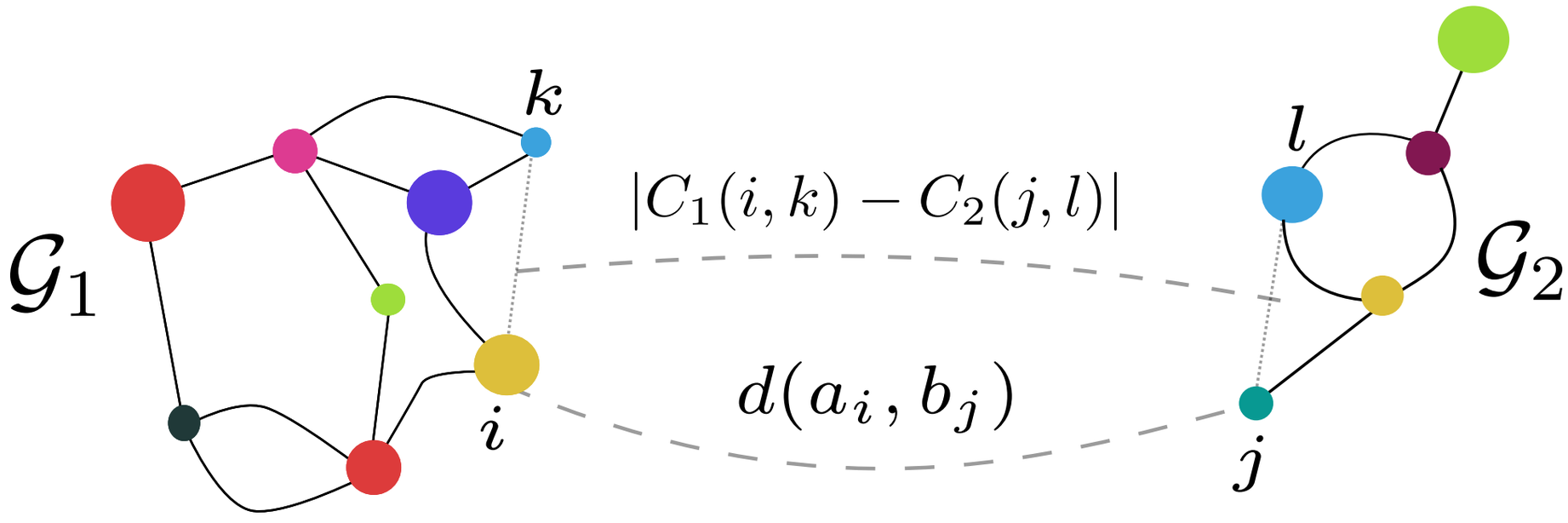
$\mathbf{C}_1, \mathbf{C}_2$

A distance between labels

$d : \Omega \times \Omega \to \mathbb{R}_+$

Fused Gromov-Wasserstein distance

$$FGW(\mathbf{M_{AB}}, \mathbf{C}_1, \mathbf{C}_2, \mathbf{h}, \mathbf{g}) = \min_{\boldsymbol{\pi} \in \Pi(\mathbf{h}, \mathbf{g})} \sum_{i,j,k,l} (1-\alpha) d(a_i, b_j)^q + \alpha |C_1(i,k) - C_2(j,l)|^q \pi_{i,j} \pi_{k,l}$$



Optimal Transport for structured data

Fused Gromov-Wasserstein distance

Two structured data

$\mu = \sum_i h_i \delta_{(x_i, a_i)}, \nu = \sum_j g_j \delta_{(y_j, b_j)}$

Two matrices describing structures

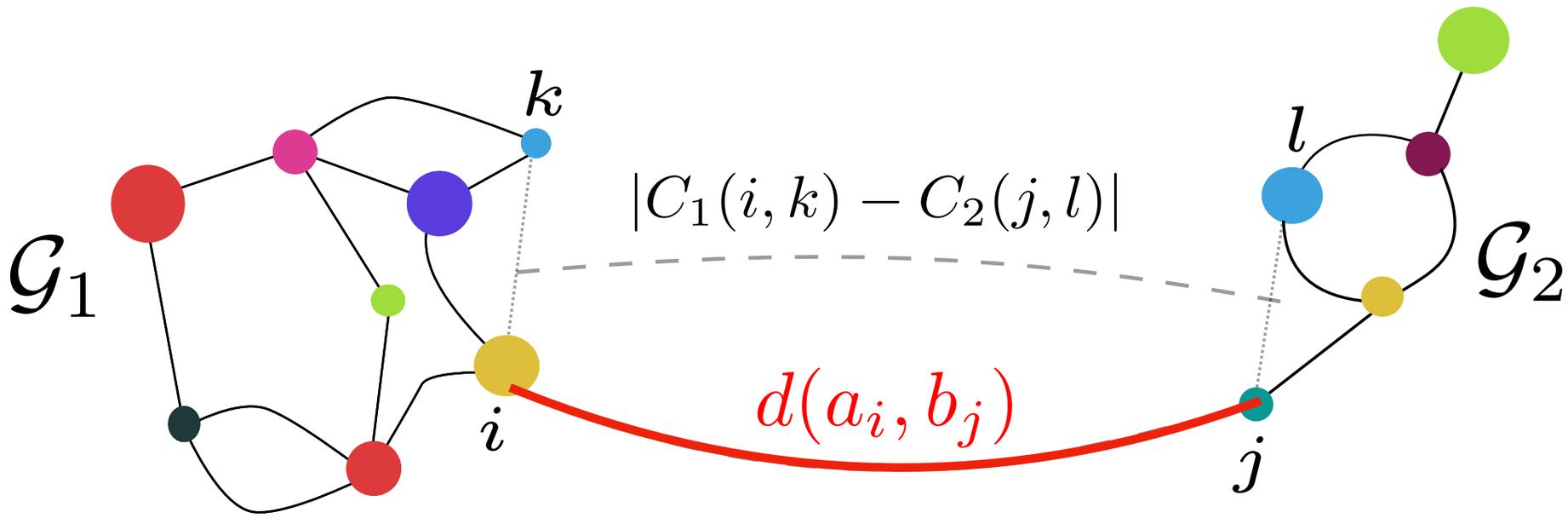
$\mathbf{C}_1, \mathbf{C}_2$

A distance between labels

$d : \Omega \times \Omega \to \mathbb{R}_+$

Fused Gromov-Wasserstein distance

$$FGW(\mathbf{M_{AB}}, \mathbf{C}_1, \mathbf{C}_2, \mathbf{h}, \mathbf{g}) = \min_{\boldsymbol{\pi} \in \Pi(\mathbf{h}, \mathbf{g})} \sum_{i,j,k,l} (1-\alpha) d(a_i, b_j)^q + \alpha |C_1(i,k) - C_2(j,l)|^q \pi_{i,j} \pi_{k,l}$$



Optimal Transport for structured data

Fused Gromov-Wasserstein distance

Two structured data

$\mu = \sum_i h_i \delta_{(x_i, a_i)}, \nu = \sum_j g_j \delta_{(y_j, b_j)}$

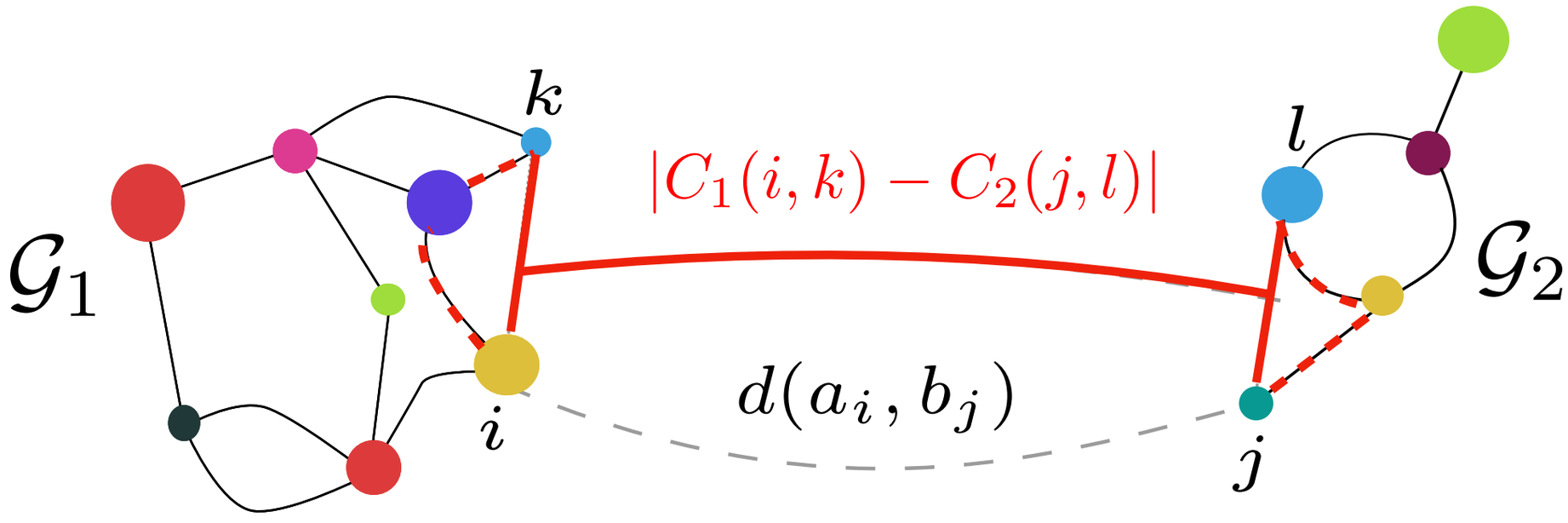
Two matrices describing structures

$\mathbf{C}_1, \mathbf{C}_2$

A distance between labels

$d : \Omega \times \Omega \to \mathbb{R}_+$

$$FGW(\mathbf{M_{AB}}, \mathbf{C}_1, \mathbf{C}_2, \mathbf{h}, \mathbf{g}) = \min_{\boldsymbol{\pi} \in \Pi(\mathbf{h}, \mathbf{g})} \sum_{i,j,k,l} (1-\alpha) d(a_i, b_j)^q + \alpha |C_1(i,k) - C_2(j,l)|^q \pi_{i,j} \pi_{k,l}$$



Optimal Transport for structured data

Fused Gromov-Wasserstein distance

Two structured data

$\mu = \sum_i h_i \delta_{(x_i, a_i)}, \nu = \sum_j g_j \delta_{(y_j, b_j)}$

Two matrices describing structures

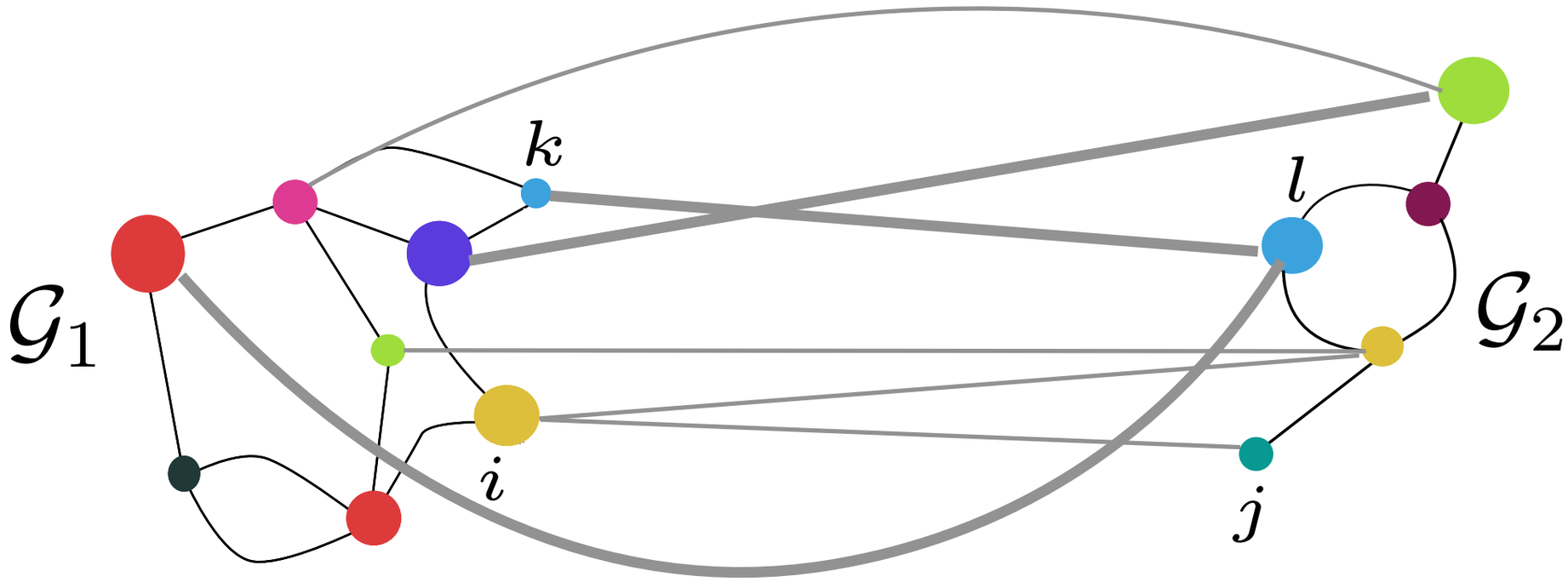
$\mathbf{C}_1, \mathbf{C}_2$

A distance between labels

$d : \Omega \times \Omega \to \mathbb{R}_+$

Fused Gromov-Wasserstein distance

$$FGW(\mathbf{M_{AB}}, \mathbf{C}_1, \mathbf{C}_2, \mathbf{h}, \mathbf{g}) = \min_{\boldsymbol{\pi} \in \Pi(\mathbf{h}, \mathbf{g})} \sum_{i,j,k,l} (1-\alpha) d(a_i, b_j)^q + \alpha |C_1(i,k) - C_2(j,l)|^q \pi_{i,j} \pi_{k,l}$$

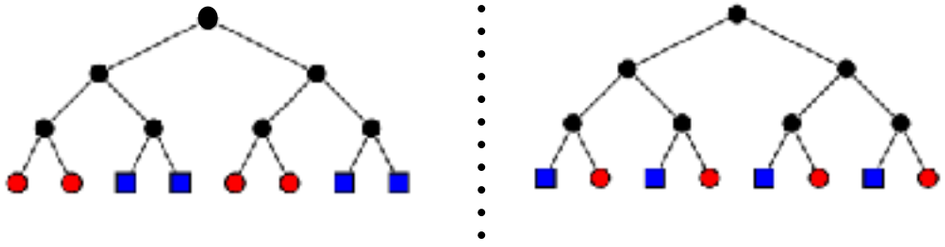


$\boldsymbol{\pi}$ **provides a soft assignment of the nodes**

Optimal Transport for structured data

Fused Gromov-Wasserstein distance: example

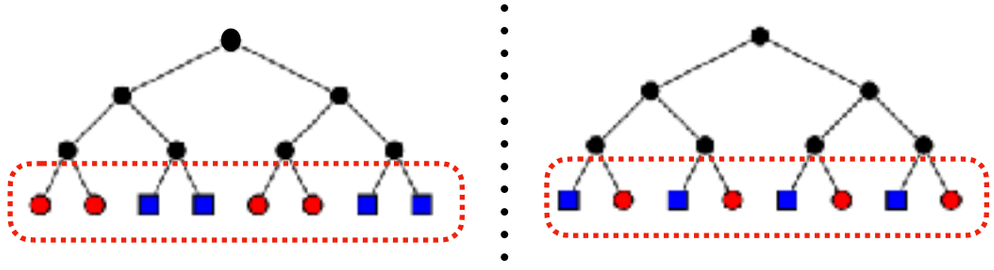
Consider two trees



Optimal Transport for structured data

Fused Gromov-Wasserstein distance: example

Consider two trees

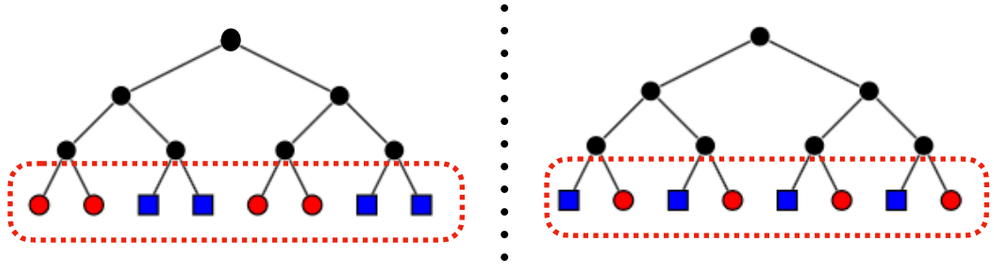


We want to compare the leaves of the trees

Optimal Transport for structured data

Fused Gromov-Wasserstein distance: example

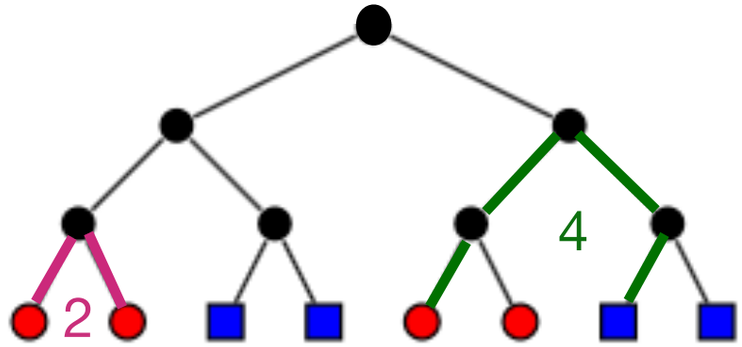
Consider two trees



We want to compare the leaves of the trees

Features: blue or red

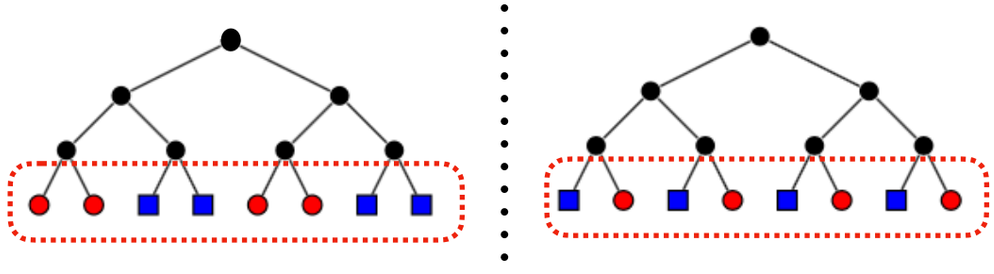
Structures : shortest path between the leaves



Optimal Transport for structured data

Fused Gromov-Wasserstein distance: example

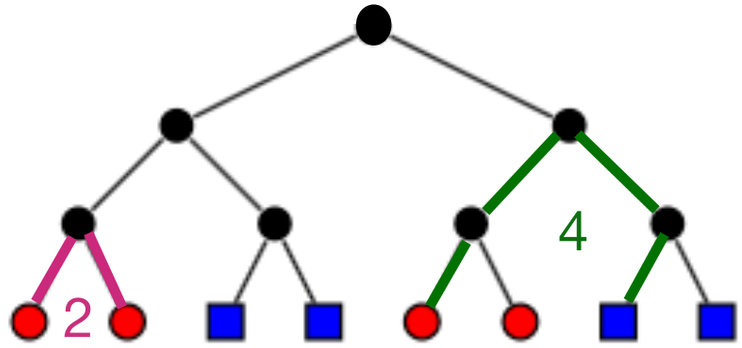
Consider two trees



We want to compare the leaves of the trees

Features: blue or red

Structures : shortest path between the leaves

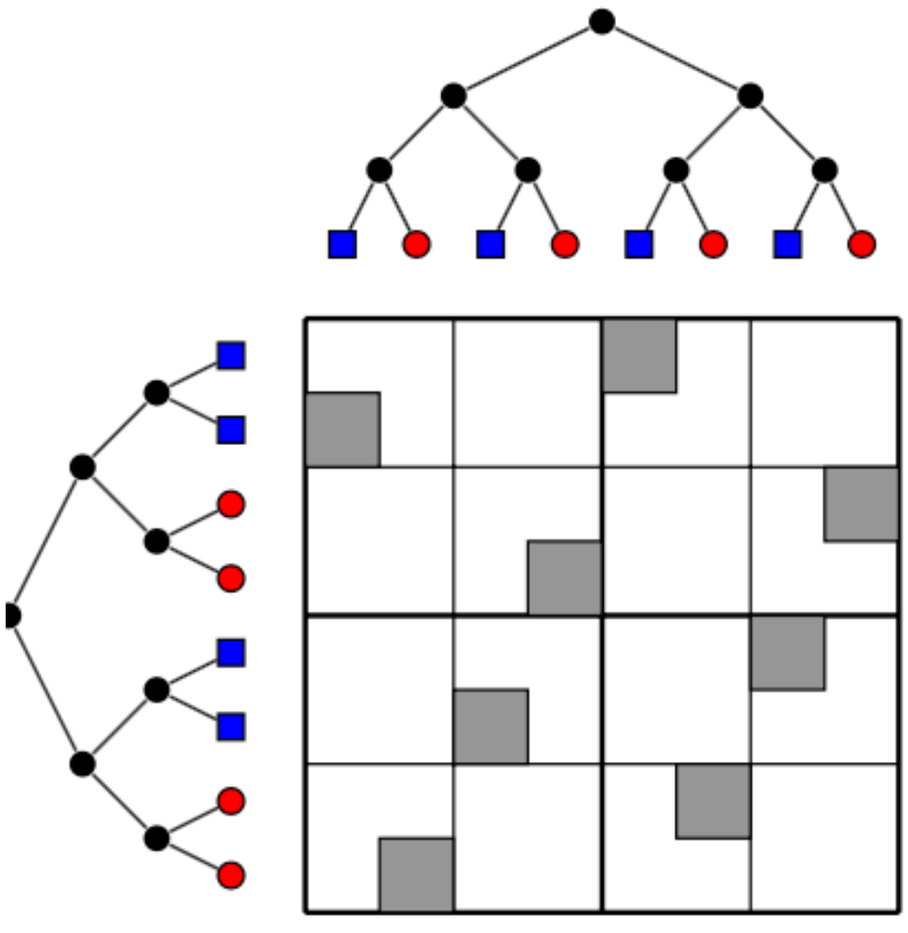


Taking both the structures and the features into account with FGW

Optimal Transport for structured data

Fused Gromov-Wasserstein distance: example

Wasserstein distance (features only)

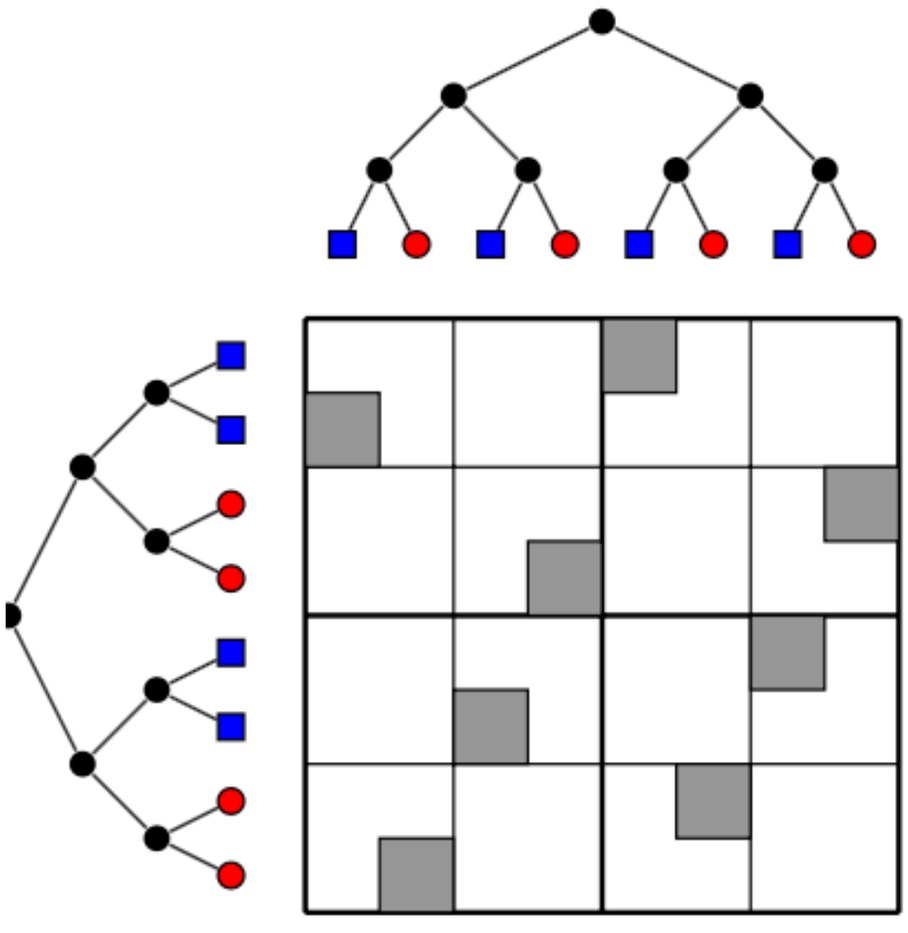


$W = 0$

Optimal Transport for structured data

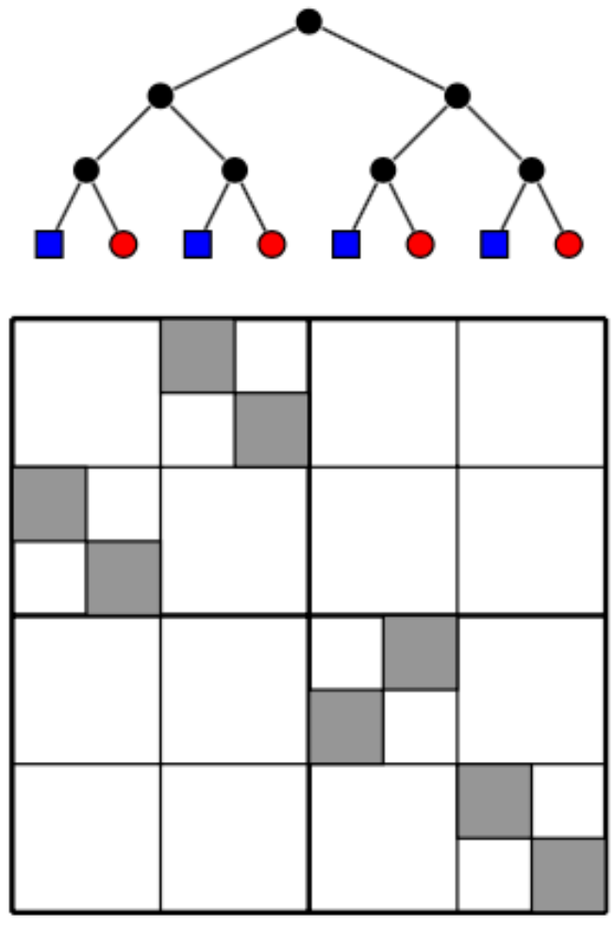
Fused Gromov-Wasserstein distance: example

Wasserstein distance (features only)



$W = 0$

Gromov-Wasserstein distance (structures only)



$GW = 0$

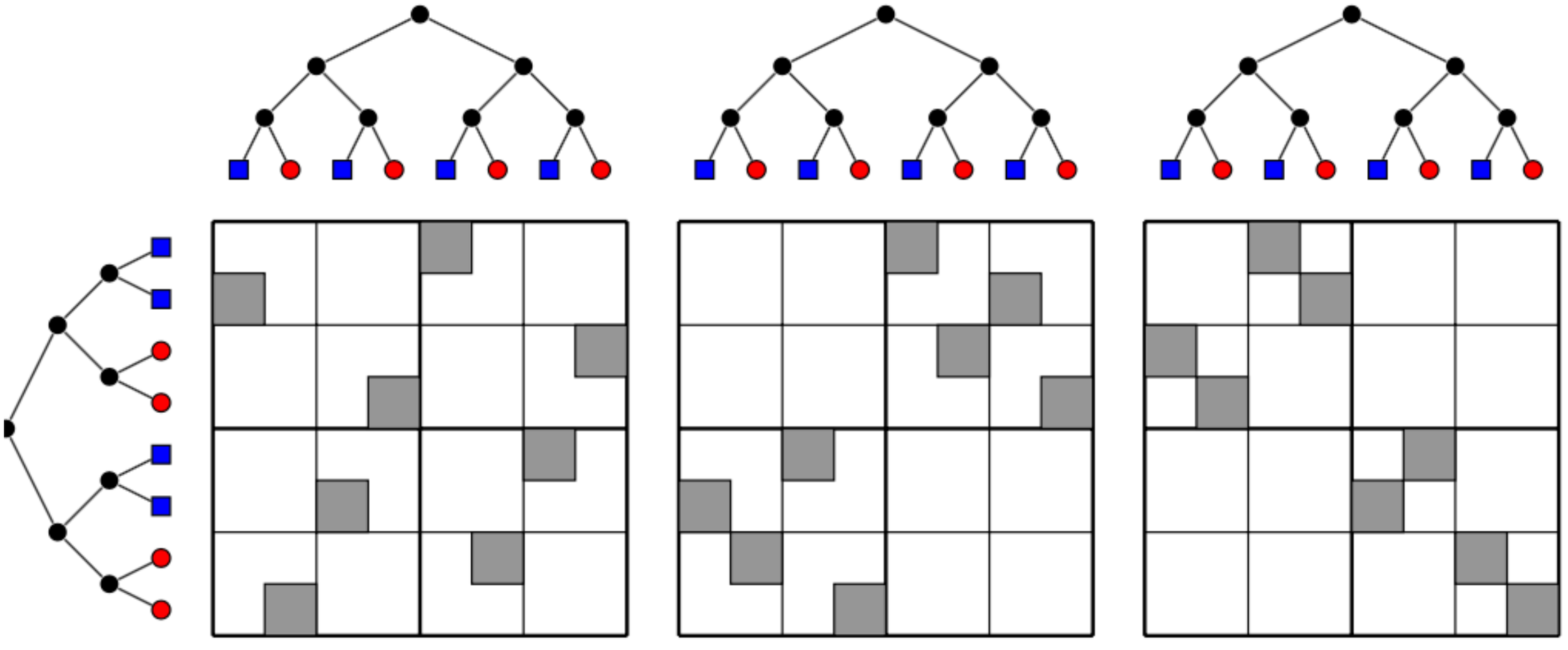
Optimal Transport for structured data

Fused Gromov-Wasserstein distance: example

Wasserstein distance (features only)

FGW

Gromov-Wasserstein distance (structures only)



$W = 0$

$FGW > 0$

$GW = 0$

Optimal Transport for structured data

Computing FGW (and GW!)

Solving FGW: a non convex QP

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{h},\mathbf{g})} \sum_{i,j,k,l} (1-\alpha) d(a_i, b_j)^q + \alpha |C_1(i,k) - C_2(j,l)|^q \pi_{i,j} \pi_{k,l}$$

Quadratic function over polytope -> Conditional Gradient algorithm (a.k.a Frank-Wolfe)

Non convex but converges to a **local optimal solution** [Lacoste-Julien 2016]

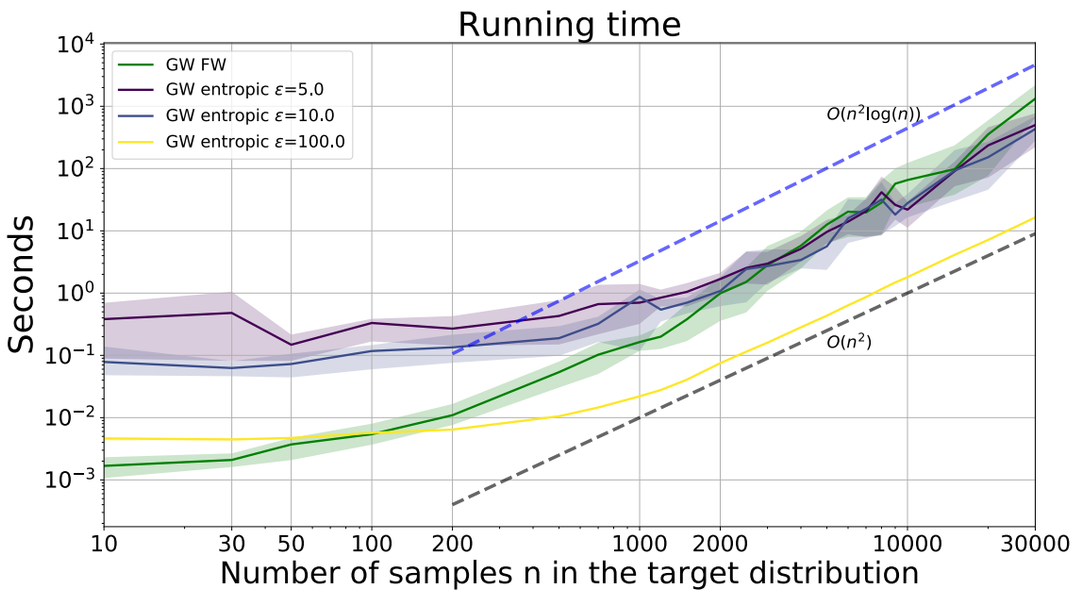
Find a **sparse** solution. FW gap = $O(\frac{1}{\sqrt{n_{iter}}})$

Algorithm 1 Conditional Gradient (CG) for FGW

$\pi^{(0)} \leftarrow \mathbf{h}\mathbf{g}^\top$
for $i = 1, \ldots,$ **do**
  $\mathbf{G} \leftarrow$ Gradient from GW loss $w.r.t.$ $\boldsymbol{\pi}^{(i-1)}$
  $\tilde{\boldsymbol{\pi}}^{(i)} \leftarrow$ Solve OT with ground loss $\mathbf{G}$
  $\tau^{(i)} \leftarrow$ Line-search for GW loss with $\tau \in (0, 1)$ (closed-form)
  $\boldsymbol{\pi}^{(i)} \leftarrow (1 - \tau^{(i)})\boldsymbol{\pi}^{(i-1)} + \tau^{(i)}\tilde{\boldsymbol{\pi}}^{(i)}$
end for

Complexity

$$O(n_{iter}\ n^3)$$



Optimal Transport for structured data

Fused Gromov-Wasserstein distance

A distance w.r.t strong isomorphism

- $FGW \geq 0$ and satisfies the triangle inequality
- $\mathbf{C}_1, \mathbf{C}_2$ distances. $FGW = 0$ iff $\exists \sigma$ permutations of the nodes

$$\text{(conservation of the weights) } h_i = g_{\sigma(i)}$$

$$\text{(conservation of the features) } a_i = b_{\sigma(i)}$$

$$\text{(conservation of the structures) } C_1(i,k) = C_2(\sigma(i), \sigma(k))$$

Optimal Transport for structured data

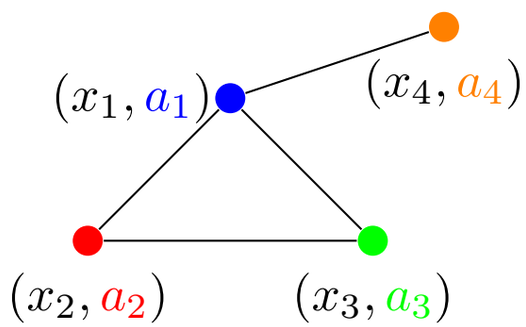
Fused Gromov-Wasserstein distance

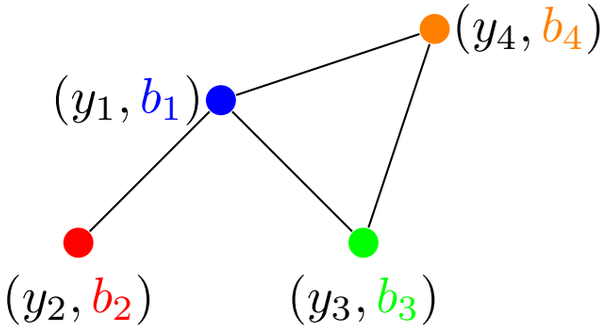
A distance w.r.t strong isomorphism

- $FGW \geq 0$ and satisfies the triangle inequality
- $\mathbf{C}_1, \mathbf{C}_2$ distances. $FGW = 0$ iff $\exists \sigma$ permutations of the nodes

$$\text{(conservation of the weights) } h_i = g_{\sigma(i)}$$
$$\text{(conservation of the features) } a_i = b_{\sigma(i)}$$
$$\text{(conservation of the structures) } C_1(i,k) = C_2(\sigma(i), \sigma(k))$$

Same weights, same labels at the same place up to a permutation





Isometric + same features but not strongly isomorphic

Optimal Transport for structured data

Fused Gromov-Wasserstein distance

A distance w.r.t strong isomorphism

- $FGW \geq 0$ and satisfies the triangle inequality
- $\mathbf{C}_1, \mathbf{C}_2$ distances. $FGW = 0$ iff $\exists \sigma$ permutations of the nodes

$$\text{(conservation of the weights) } h_i = g_{\sigma(i)}$$
$$\text{(conservation of the features) } a_i = b_{\sigma(i)}$$
$$\text{(conservation of the structures) } C_1(i,k) = C_2(\sigma(i), \sigma(k))$$

Other properties

Interpolates GW between the structures and W between the features

Extends to the continuous setting: geodesic properties + sample complexity

FGW in action

Optimal Transport for structured data

FGW in action

Graph classification

A set of labeled graphs $(\mathcal{G}_i, y_i)$. Structure matrices shortest path

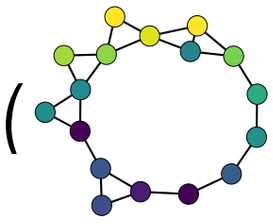
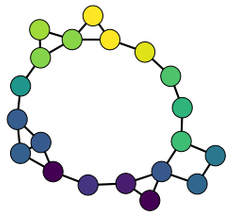
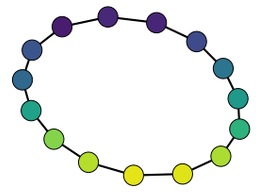
Linear classifier: SVM on the indefinite kernel $e^{-\frac{1}{\beta}FGW(\mathcal{G}_i,\mathcal{G}_j)}$

Compare with graph kernel approaches + GCN on benchmark datasets

Dataset	Labeled Graphs			Social Graphs	Vector attributes Graph		
	MUTAG	PTC	NCI1	IMDB-B	SYNTHETIC	PROTEIN	CUNEIFORM
WL	86.21±8.15	62.17±7.80	85.13±1.61	Unapplicable(U)	U	U	U
GK	82.42±8.40	56.46±8.03	60.78±2.48	56.00±3.61	41.13±4.68	U	U
RW	79.47±8.17	55.09±7.34	58.63±2.44	U	U	U	U
SP	85.79±2.51	58.53±2.55	73.00±0.51	55.80±2.93	38.93±5.12	U	U
HOPPER	U	U	U	U	90.67±4.67	71.96±3.22	32.59±8.73
PROPA	U	U	U	U	64.67±6.70	61.34±4.38	12.59± 6.67
PSCN $k = 10$	83.47±10.26	58.34±7.71	70.65±2.58	U	**100.00±0.00**	67.95±11.28	25.19±7.73
FGW	**88.42±5.67**	**65.31±7.90**	**86.42±1.63**	**63.80±3.49**	**100.00±0.00**	**74.55±2.74**	**76.67±7.04**

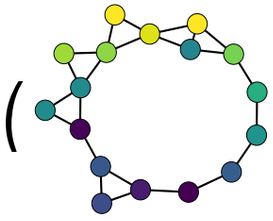
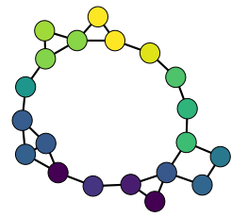
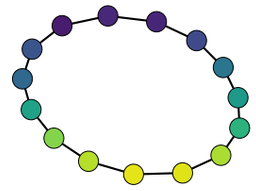
Optimal Transport for structured data

FGW barycenter

Making sense of: $\frac{1}{2}\Big($  $+$  $\Big) =$ 

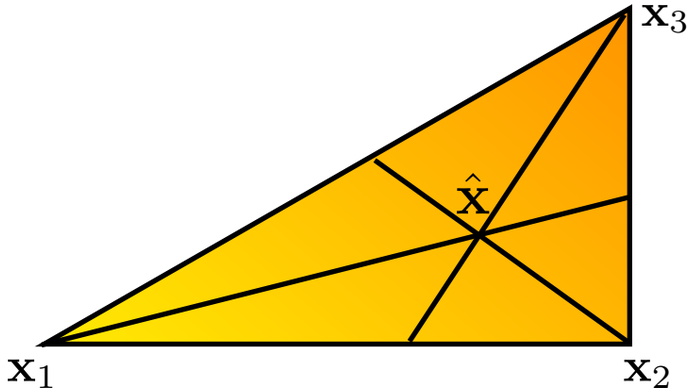
Optimal Transport for structured data

FGW barycenter

Making sense of: $\frac{1}{2}($  $+$  $) =$ 

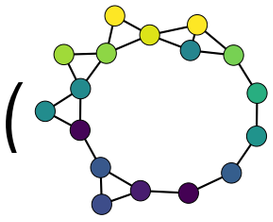
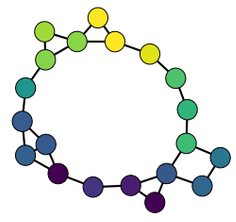
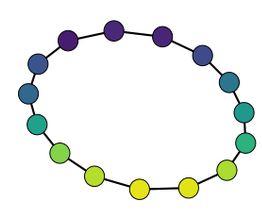
Euclidean Barycenter: $(\mathbb{R}^d, \|.\|_2)$

$$\inf_{\mathbf{x}\in\mathbb{R}^d} \sum_{i=1}^{n} \lambda_i \|\mathbf{x} - \mathbf{x}_i\|_2^2$$



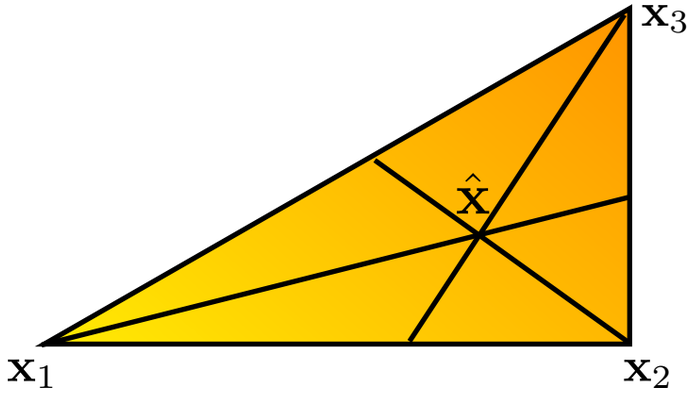
Optimal Transport for structured data

FGW barycenter

Making sense of: $\frac{1}{2}($  $+$  $) =$ 

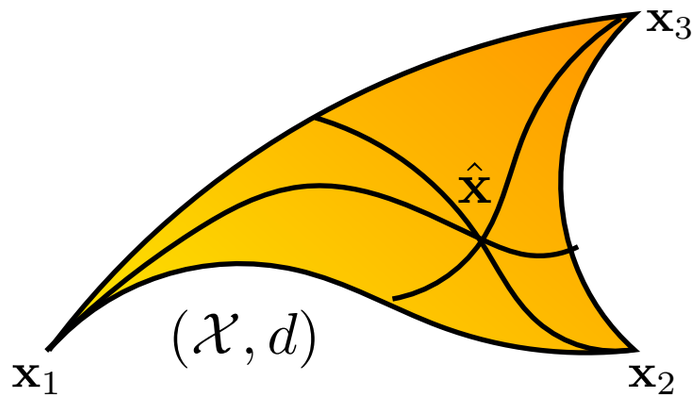
Euclidean Barycenter: $(\mathbb{R}^d, \|.\|_2)$

$$\inf_{\mathbf{x}\in\mathbb{R}^d} \sum_{i=1}^n \lambda_i \|\mathbf{x} - \mathbf{x}_i\|_2^2$$



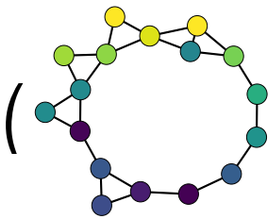
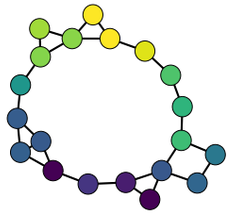
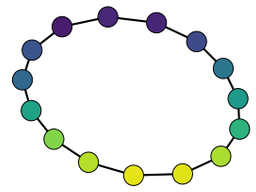
Fréchet Barycenter: $(\mathcal{X}, d)$ metric space

$$\inf_{x\in\mathcal{X}} \sum_{i=1}^n \lambda_i d(x, x_i)^p$$



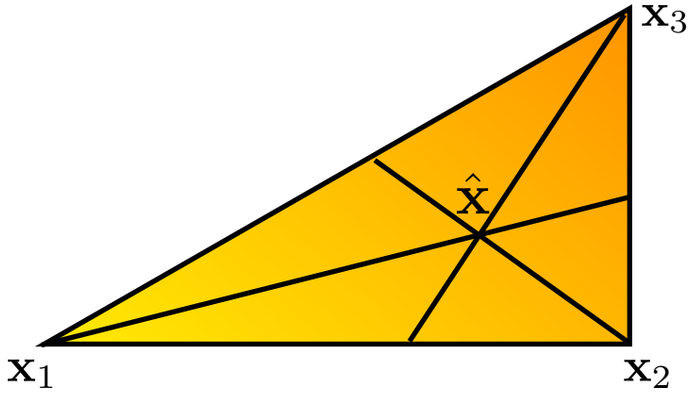
Optimal Transport for structured data

FGW barycenter

Making sense of: $\frac{1}{2}($  $+$  $) =$ 

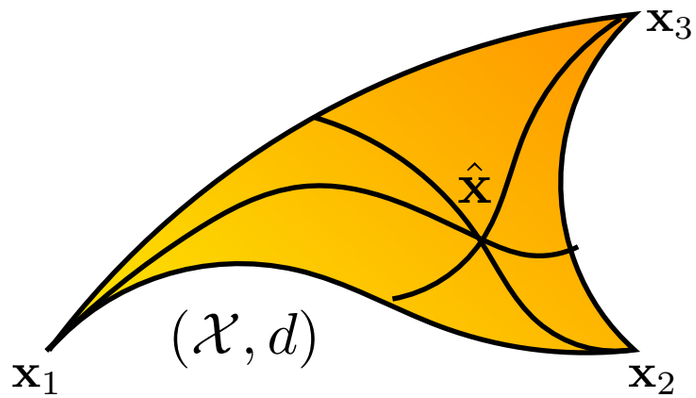
Euclidean Barycenter: $(\mathbb{R}^d, \|.\|_2)$

$$\inf_{\mathbf{x}\in\mathbb{R}^d} \sum_{i=1}^n \lambda_i \|\mathbf{x} - \mathbf{x}_i\|_2^2$$



Fréchet Barycenter: $(\mathcal{X}, d)$ metric space

$$\inf_{x\in\mathcal{X}} \sum_{i=1}^n \lambda_i d(x, x_i)^p$$



FGW barycenter

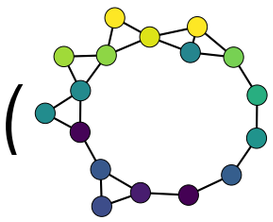
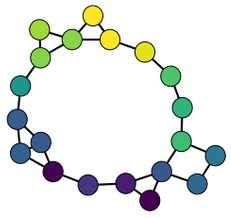
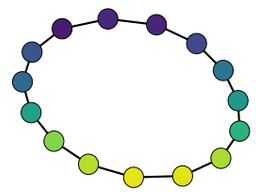
$$\min_{\mu} \sum_{k=1}^K \lambda_k FGW_{q,\alpha}(\mu, \mu_k)$$

Barycenter of labeled graphs, relational data with attributes

Consider feature space $\Omega = (\mathbb{R}^d, \|.\|_2^2)$ structured data $(\mathbf{C}_k, \mathbf{B}_k, \mathbf{h}_k)_{k=1}^K$

Optimal Transport for structured data

FGW barycenter

Making sense of: $\frac{1}{2}($  $+$  $) =$ 

FGW barycenter

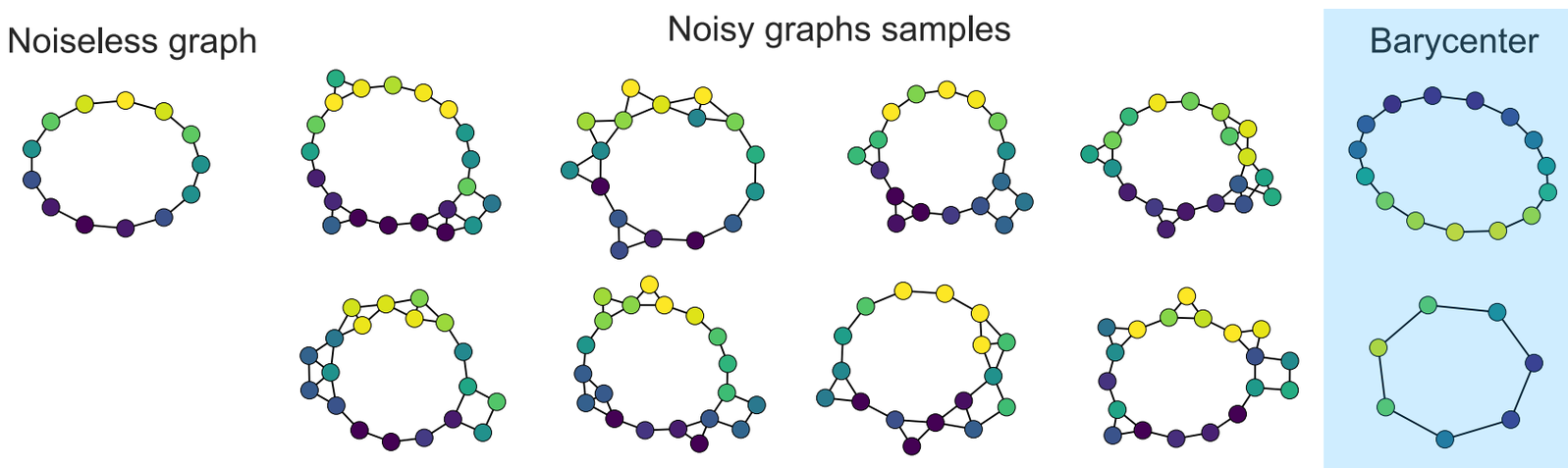
$$\min_{\mu} \sum_{k=1}^{K} \lambda_k FGW_{q,\alpha}(\mu, \mu_k)$$

Barycenter of labeled graphs, relational data with attributes

Consider feature space $\Omega = (\mathbb{R}^d, \|.\|_2^2)$ structured data $(\mathbf{C}_k, \mathbf{B}_k, \mathbf{h}_k)_{k=1}^K$

Algorithm 1 FGW barycenter

Initialize $\mathbf{C} \leftarrow \mathbf{C}_0, \mathbf{A} \leftarrow \mathbf{A}_0$.
while not converged **do**
  for $k = 1 \dots K$ **do**
    $\boldsymbol{\pi}_k \leftarrow FGW(\mathbf{M}_{\mathbf{AB}_k}, \mathbf{C}, \mathbf{C}_k, \mathbf{h}, \mathbf{h}_k)$
  end for
  $\mathbf{C} \leftarrow \frac{1}{\mathbf{h}\mathbf{h}^T} \sum_{k=1}^{K} \lambda_k \boldsymbol{\pi}_k^T \mathbf{C}_k \boldsymbol{\pi}_k$
  $\mathbf{A} \leftarrow \sum_{k=1}^{K} \lambda_k \mathbf{B}_k \boldsymbol{\pi}_k^T \mathrm{diag}(\frac{1}{\mathbf{h}})$
end while



Optimal Transport for structured data

Summarization of graph

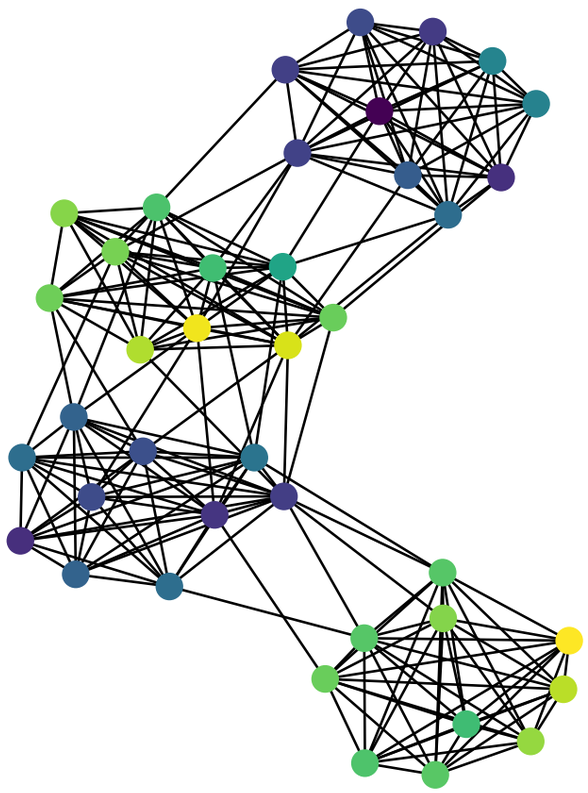
FGW coarsening

$$\min_{\mu} FGW(\mu, \nu) = \min_{\mathbf{A}, \mathbf{C}_1} FGW(\mathbf{M}_{\mathbf{AB}}, \mathbf{C}_1, \mathbf{C}_2, \mathbf{h}, \mathbf{g})$$

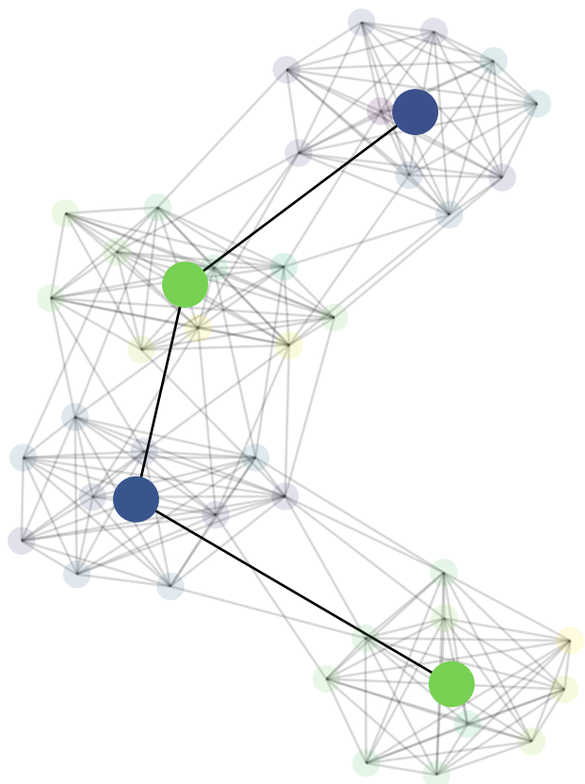
Given a labeled graph we look for the closest graph w.r.t FGW with fewer nodes

Projection w.r.t FGW -> barycenter problem with $K = 1$

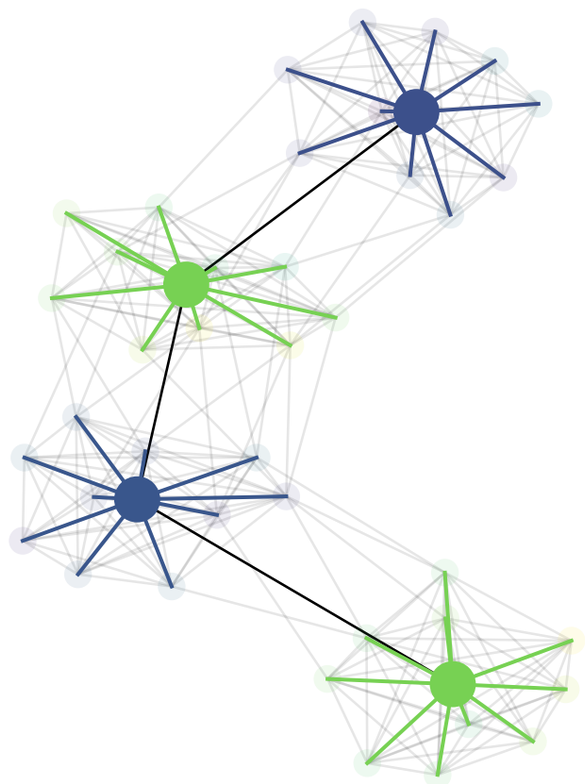
Graph with communities



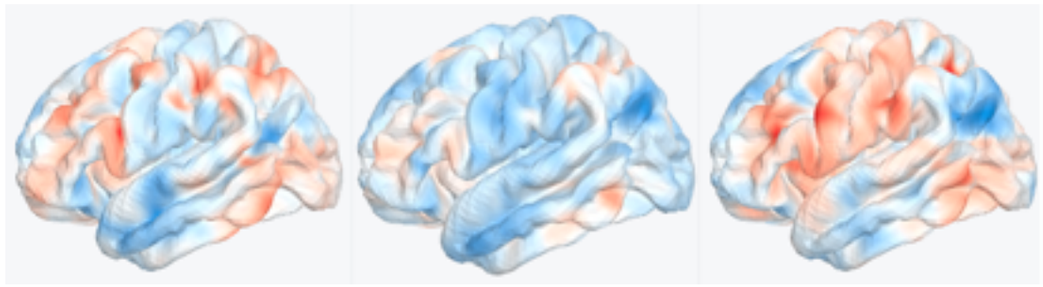
Approximate Graph



Clustering with transport matrix



Part III: Functional Brain Registration



Neurips 2022

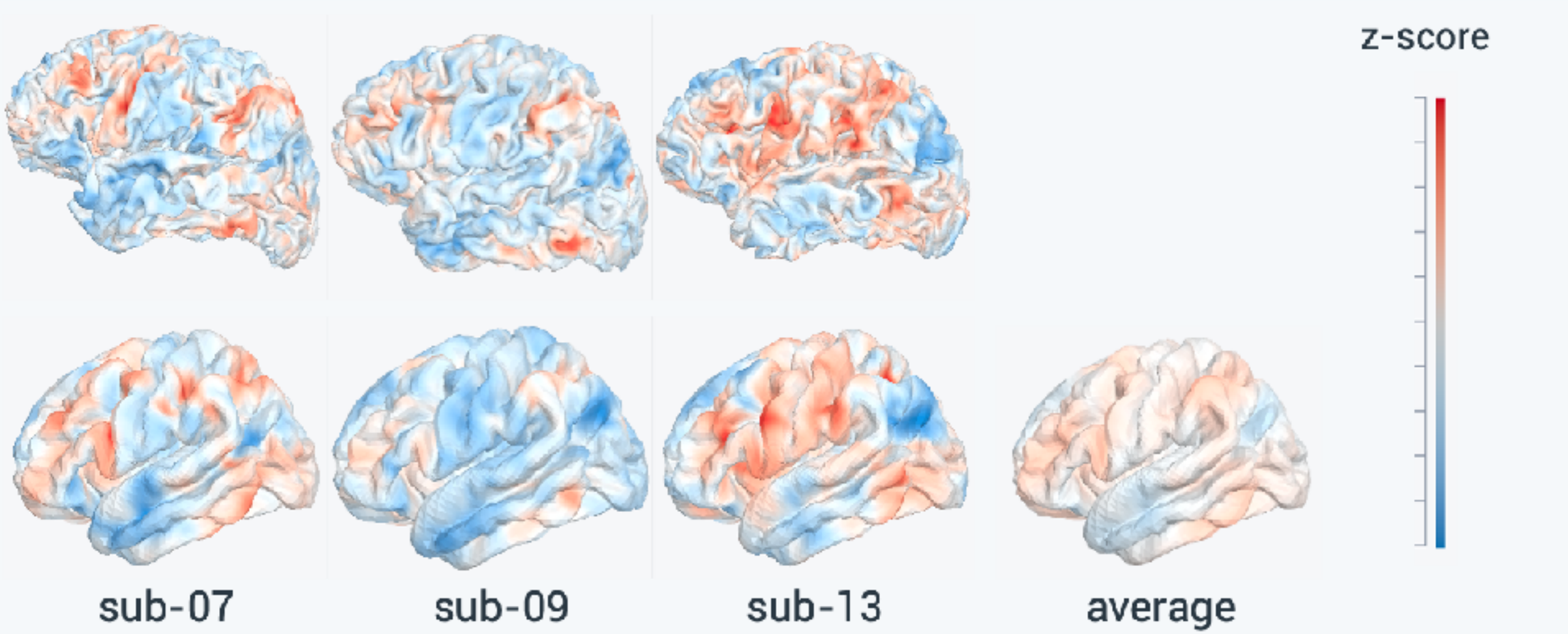
Functional Brain Registration

Fused Unbalanced Gromov Wasserstein

Matching cortical surfaces using fMRI data

Motivation

High inter-subject anatomical and functional variability

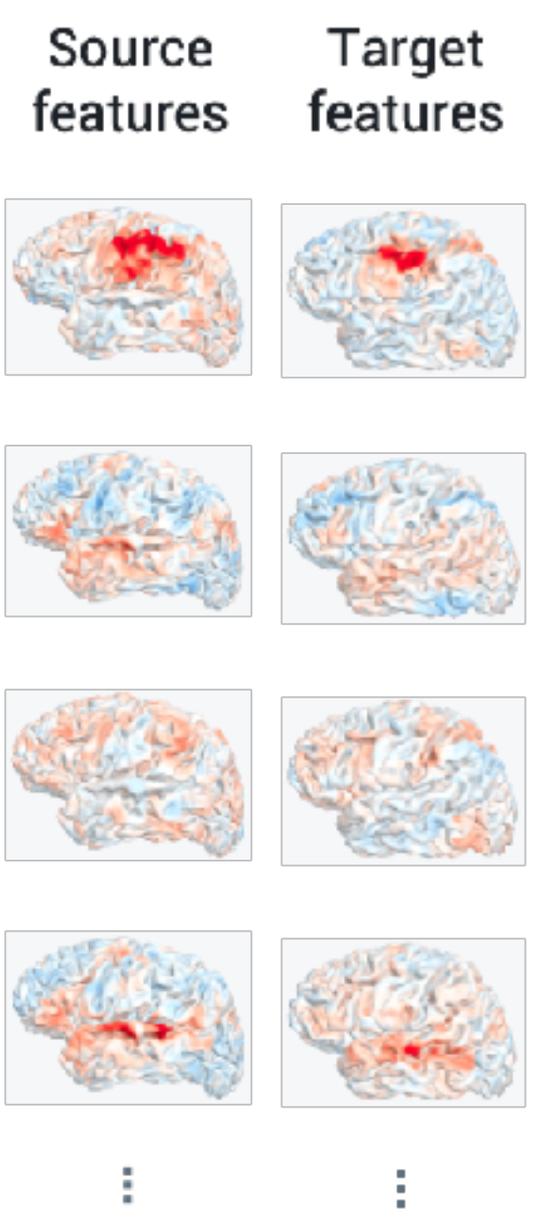


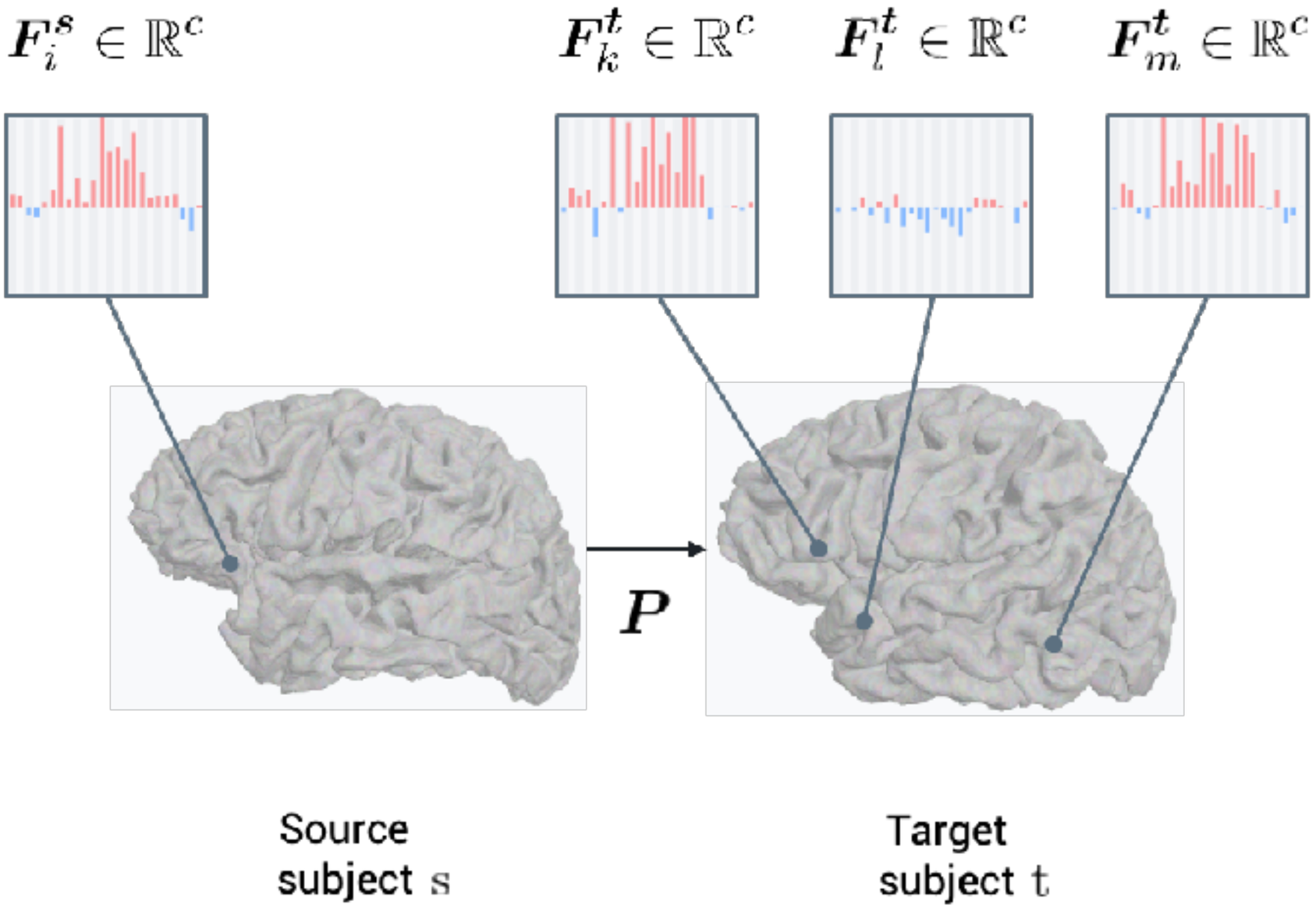
Math-nonmath contrast map from the Mathlang protocol for 3 IBC subjects,
on their individual anatomies (top row) or projected on fsaverage5 (bottom row)

Matching cortical surfaces using fMRI data

Motivation

Need to align locations on cortical surfaces





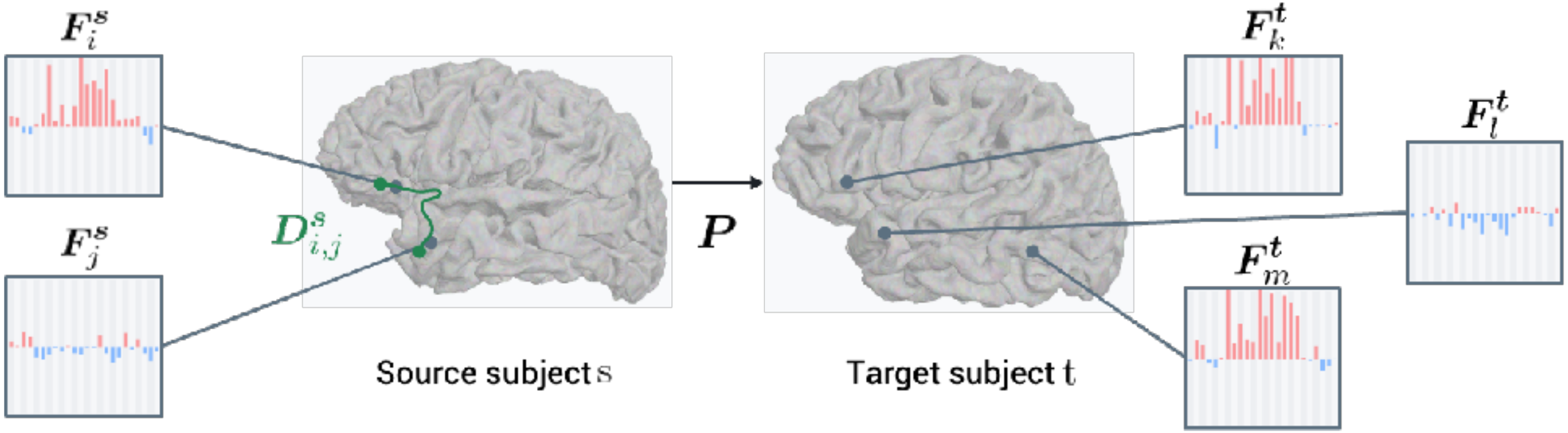
Fused Unbalanced Gromov-Wasserstein

Problem formulation and neuroscientific interpretation

Need to align locations on cortical surfaces

$$\underset{\boldsymbol{P}}{\operatorname{argmin}}$$

Computed alignment should



Fused Unbalanced Gromov-Wasserstein

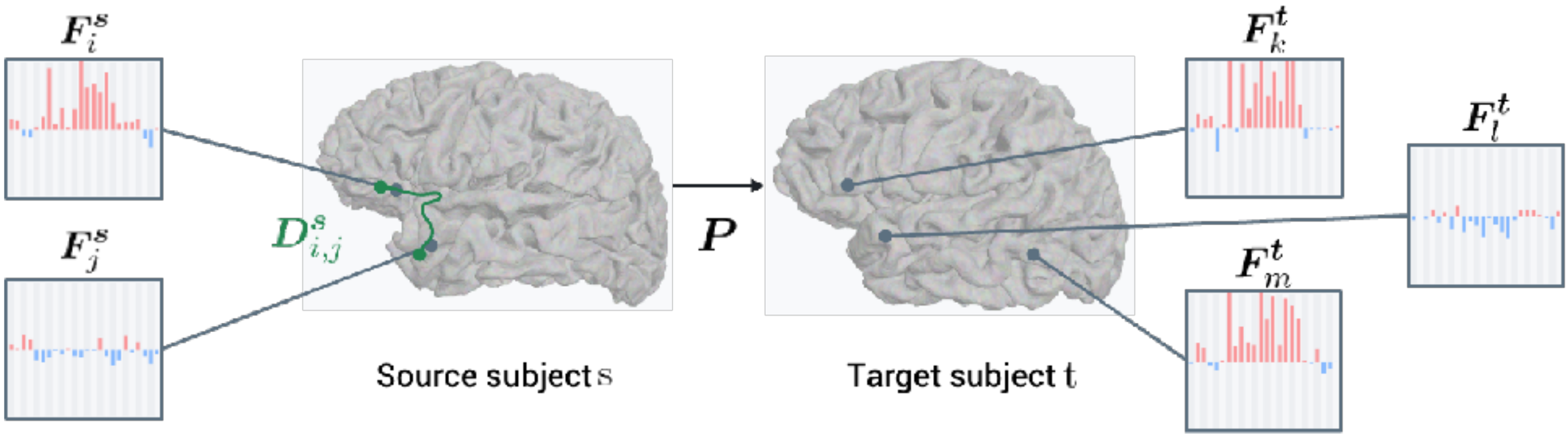
Problem formulation and neuroscientific interpretation

Need to align locations on cortical surfaces

$$\underset{\boldsymbol{P}}{\operatorname{argmin}}\ (1-\alpha) \sum_{\substack{0\le i<n \\ 0\le k<p}} ||\boldsymbol{F}_i^s - \boldsymbol{F}_k^t||_2^2 \boldsymbol{P}_{i,k}$$

Computed alignment should

match voxels with similar functional activity



Fused Unbalanced Gromov-Wasserstein

Problem formulation and neuroscientific interpretation

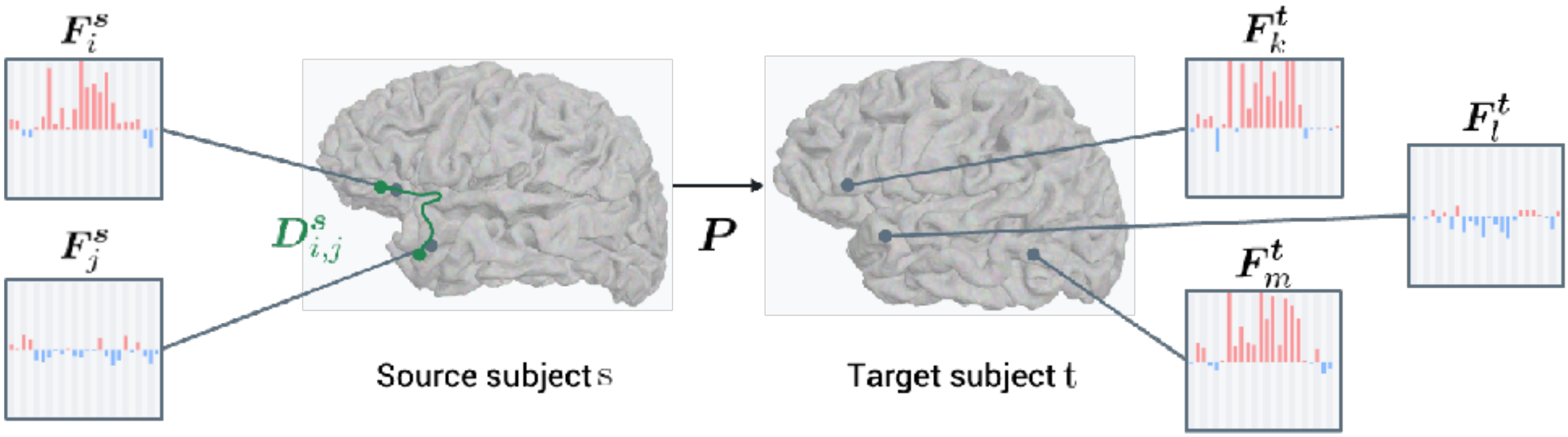
Need to align locations on cortical surfaces

$$\underset{\boldsymbol{P}}{\operatorname{argmin}}\ (1-\alpha)\sum_{\substack{0\le i<n\\0\le k<p}}||\boldsymbol{F}^{\boldsymbol{s}}_i-\boldsymbol{F}^{\boldsymbol{t}}_k||^2_2\boldsymbol{P}_{i,k}+\alpha\sum_{\substack{0\le i,j<n\\0\le k,l<p}}|\boldsymbol{D}^{\boldsymbol{s}}_{i,j}-\boldsymbol{D}^{\boldsymbol{t}}_{k,l}|^2\boldsymbol{P}_{i,k}\boldsymbol{P}_{j,i}$$

Computed alignment should

match voxels with similar functional activity

penalize anatomical neighbourhoods of matched voxels that don't look alike



Fused Unbalanced Gromov-Wasserstein

Problem formulation and neuroscientific interpretation

Need to align locations on cortical surfaces

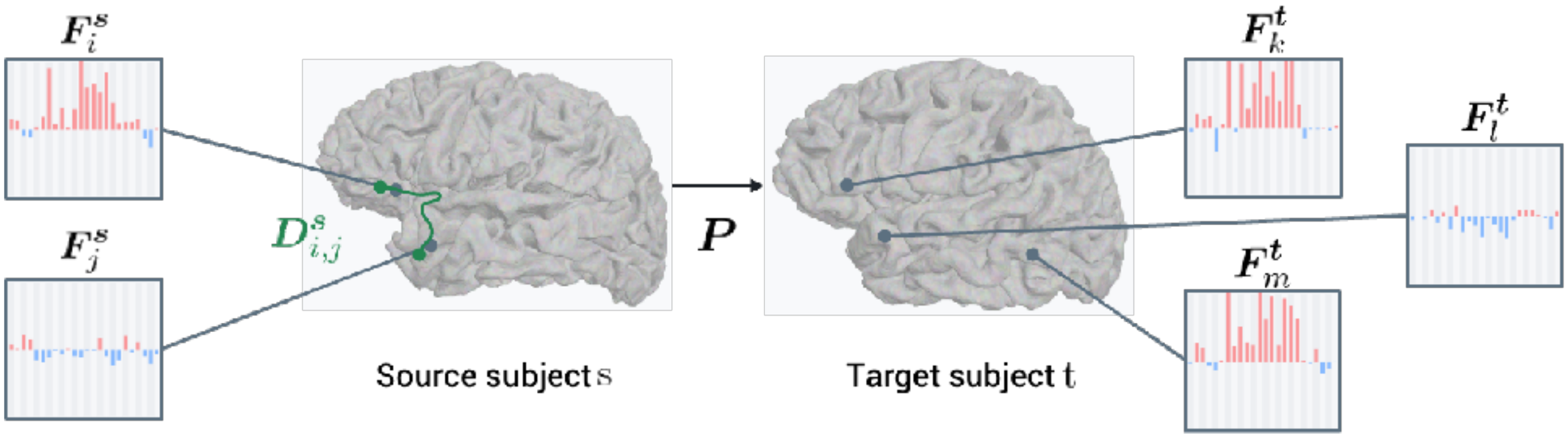
$$\underset{\boldsymbol{P}}{\operatorname{argmin}}\ (1-\alpha)\underbrace{\sum_{\substack{0\le i<n\\0\le k<p}}\|\boldsymbol{F}^s_i-\boldsymbol{F}^t_k\|_2^2\boldsymbol{P}_{i,k}}+\alpha\underbrace{\sum_{\substack{0\le i,j<n\\0\le k,l<p}}|\boldsymbol{D}^s_{i,j}-\boldsymbol{D}^t_{k,l}|^2\boldsymbol{P}_{i,k}\boldsymbol{P}_{j,i}}-\rho\underbrace{\left(\mathrm{KL}(\boldsymbol{P}_{\#1}\otimes\boldsymbol{P}_{\#1}|\boldsymbol{w}^s\otimes\boldsymbol{w}^s)+\mathrm{KL}(\boldsymbol{P}_{\#2}\otimes\boldsymbol{P}_{\#2}|\boldsymbol{w}^t\otimes\boldsymbol{w}^t)\right)}+\varepsilon\,\boldsymbol{E}(\boldsymbol{P})$$

Computed alignment should

match voxels with similar functional activity

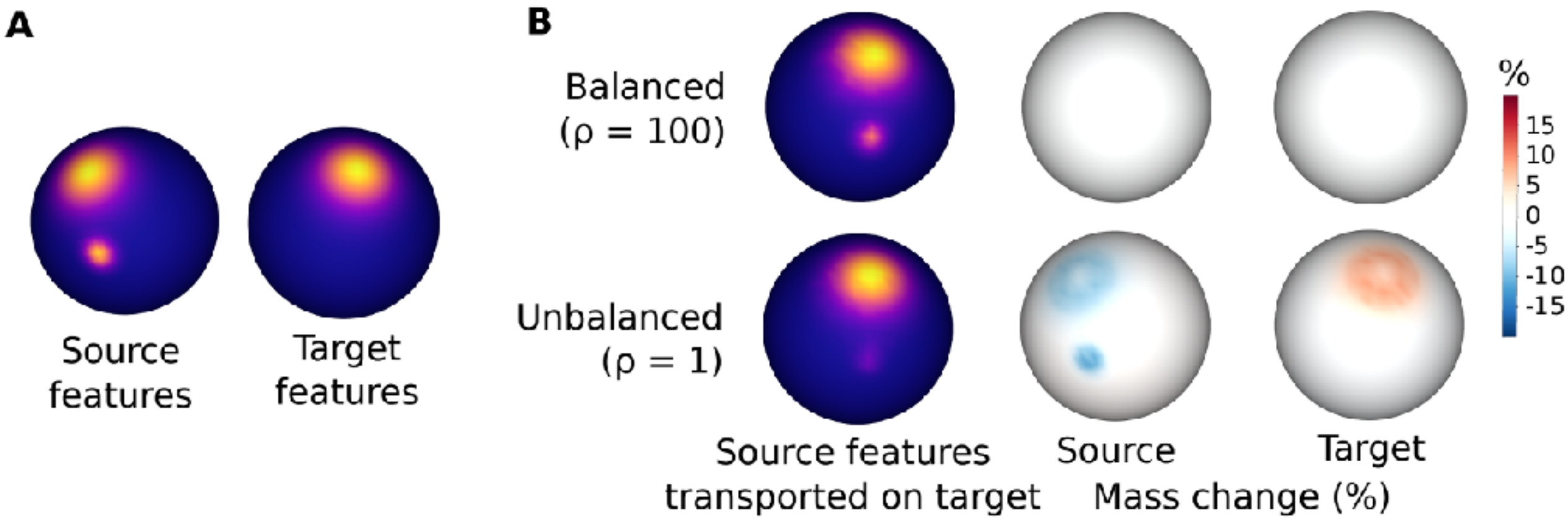
penalize anatomical neighbourhoods of matched voxels that don't look alike

allow voxel not to be transported if can't find a good match



Fused Unbalanced Gromov-Wasserstein

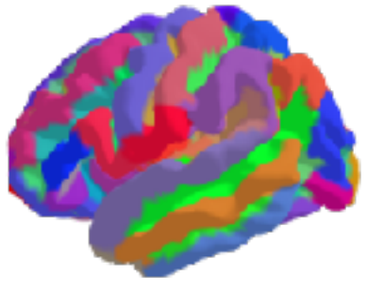
Why Unbalanced ?



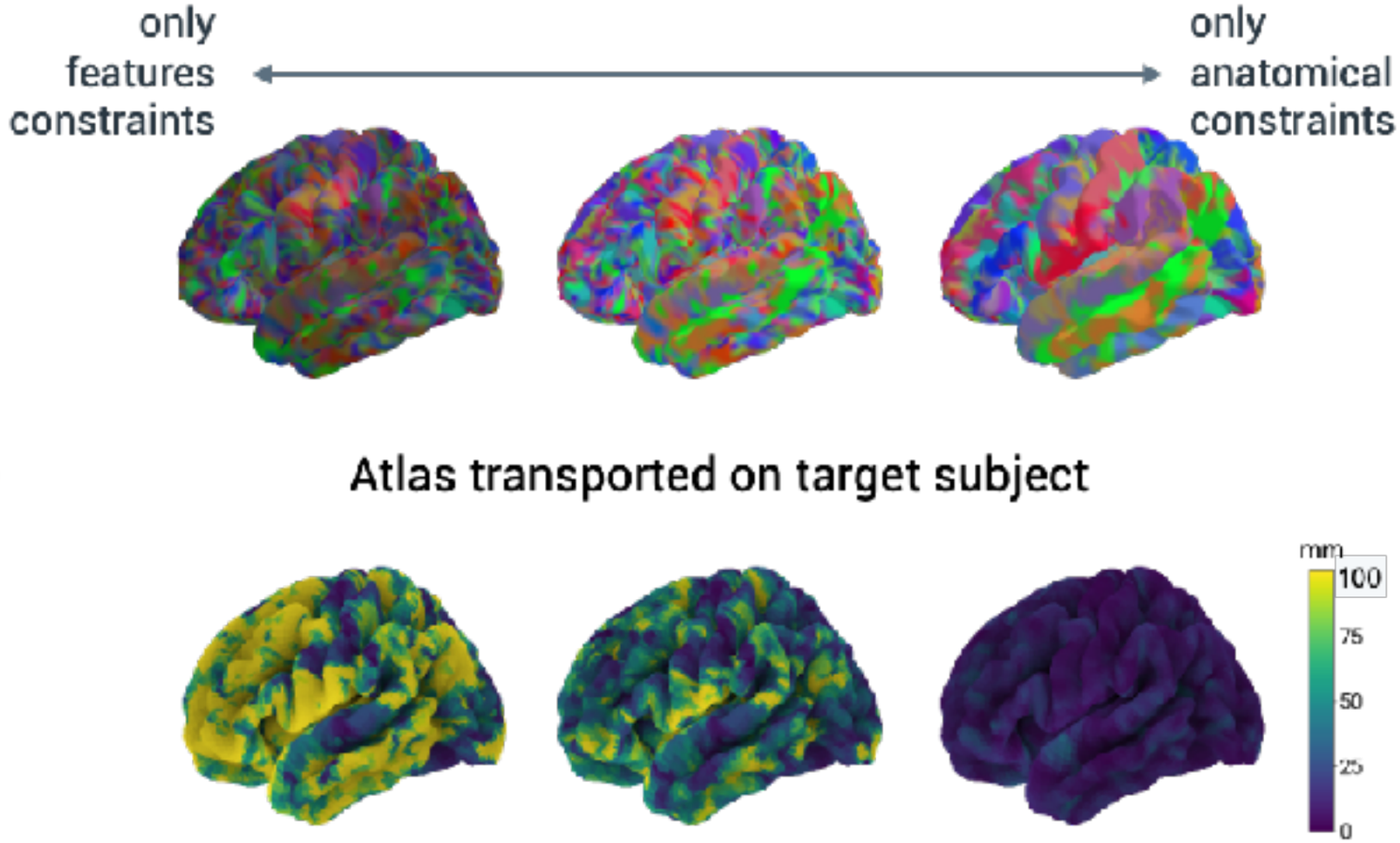
Fused Unbalanced Gromov-Wasserstein

Impact of the α parameter

Eliminating anatomically implausible couplings



Atlas on source subject



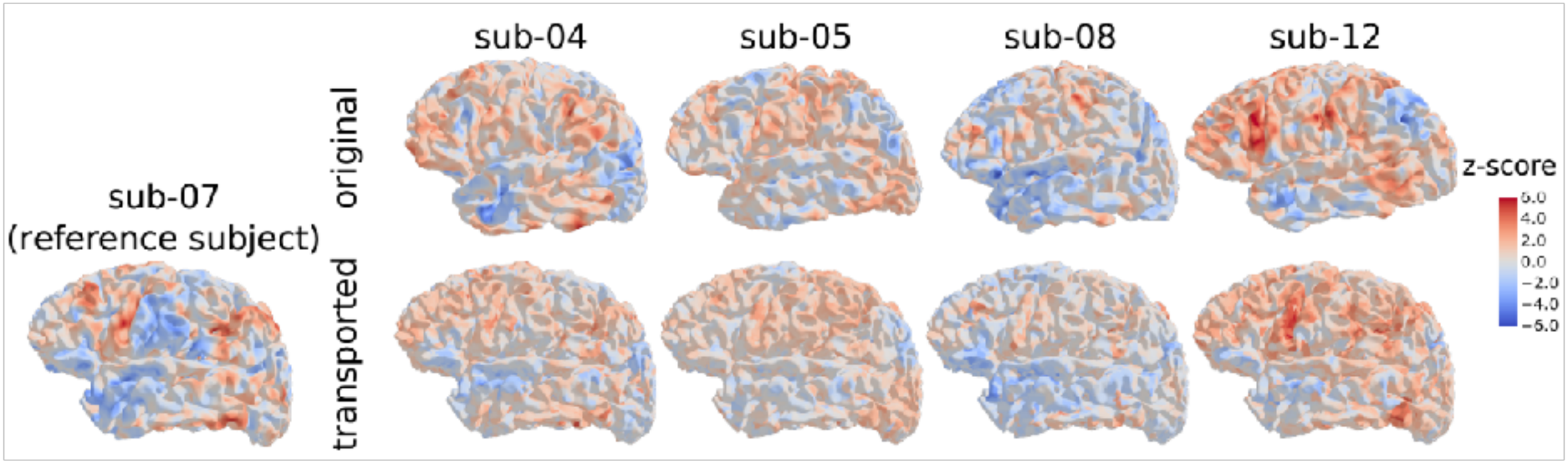
Atlas transported on target subject

Geodesic distance from source voxel to target voxel

Fused Unbalanced Gromov-Wasserstein

Projecting contrast maps using computed alignments

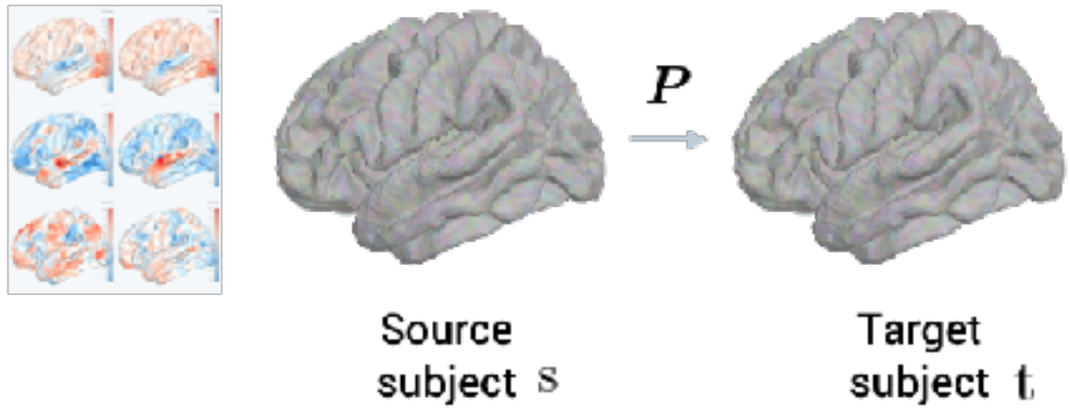
$$\phi_{s \to t} : \boldsymbol{X} \in \mathbb{R}^{n \times q} \mapsto \left(\left(\boldsymbol{P}^{\boldsymbol{s},\boldsymbol{t}} \right)^T \boldsymbol{X} \right) \oslash \boldsymbol{P}^{\boldsymbol{s},\boldsymbol{t}}_{\#2} \in \mathbb{R}^{p \times q}$$

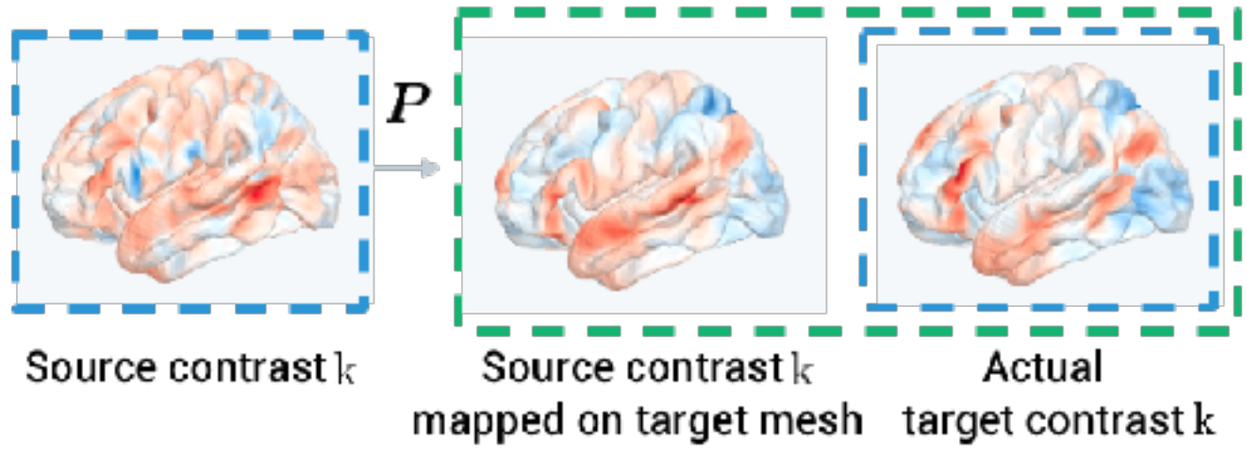


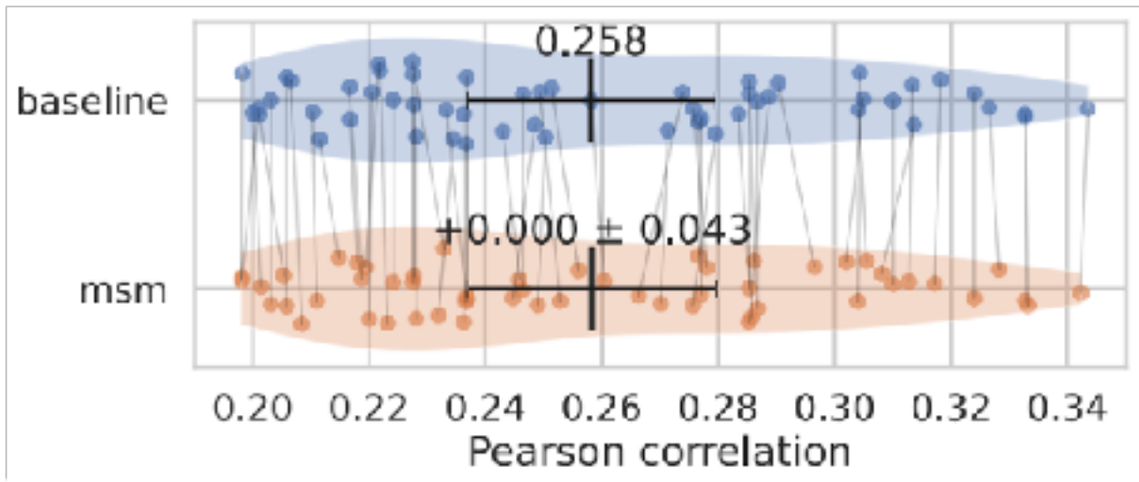
Contrast map from MathLang (math-nonmath) for each individual (top row) and after they have been projected onto a reference individual (bottom row)

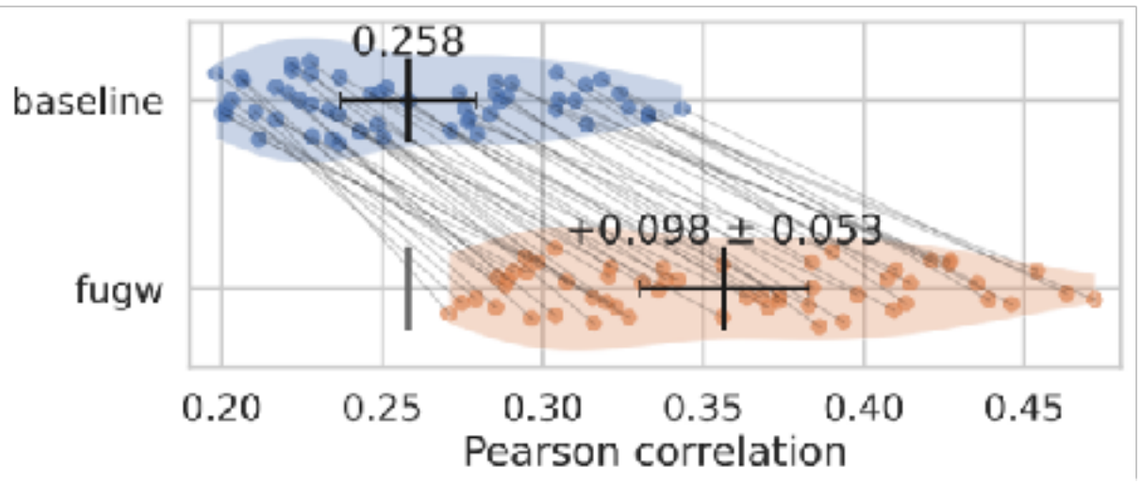
Fused Unbalanced Gromov-Wasserstein

Aligning pairs of individuals with FUGW significantly increases correlation





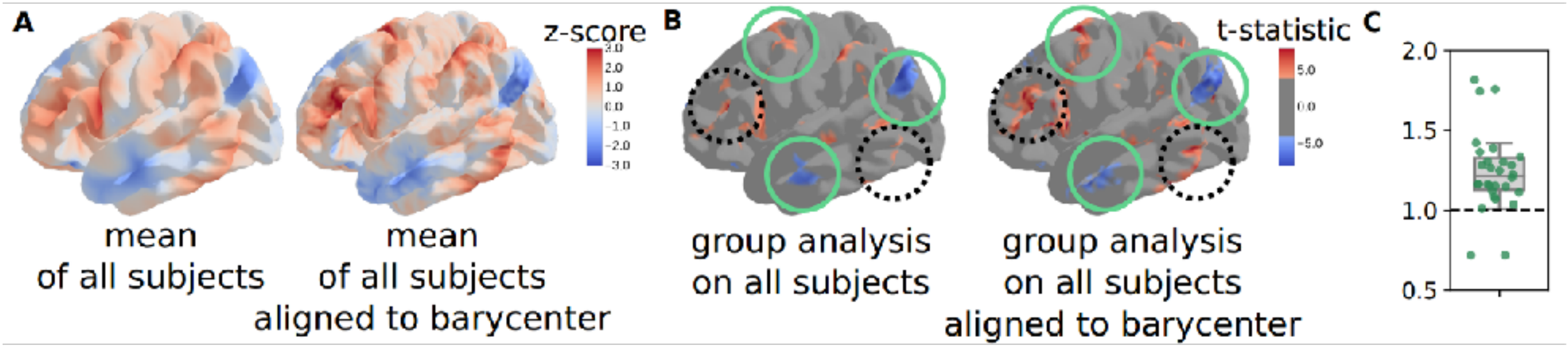


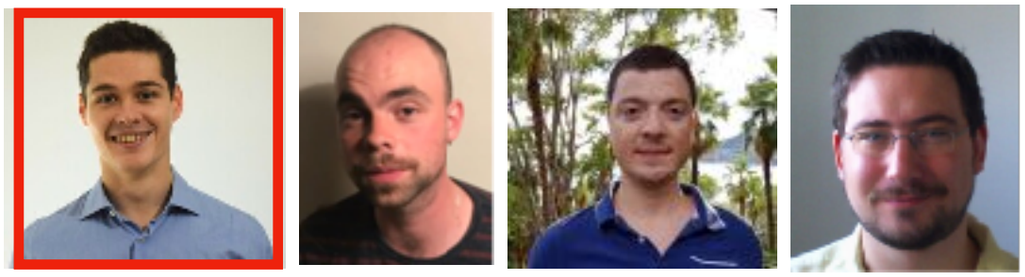


Fused Unbalanced Gromov-Wasserstein

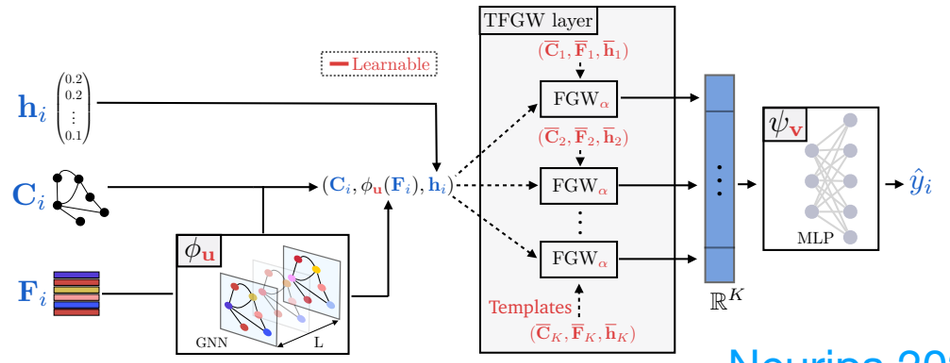
Aligning individuals to a functional barycenter significantly increases statistical power of group averages

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





Part IV: A new Pooling Layer in GNN



Neurips 2022 (oral)

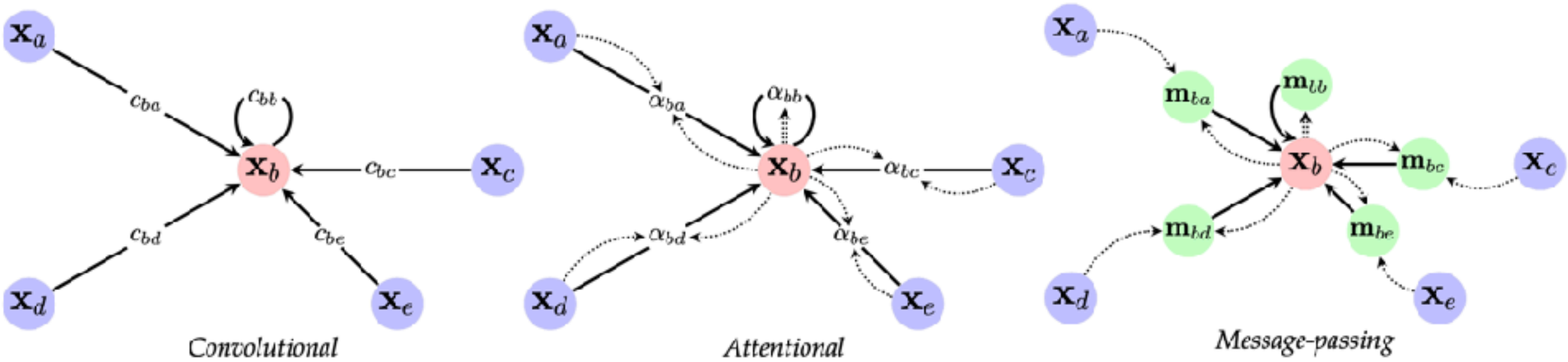
Application to GNNs

Template Fused Gromov Wasserstein

Graph Neural Networks

SOTA on many graph learning problems

Most of graph neural networks are built on variants of the **message-passing** principle



Followed by a global aggregation or pooling:

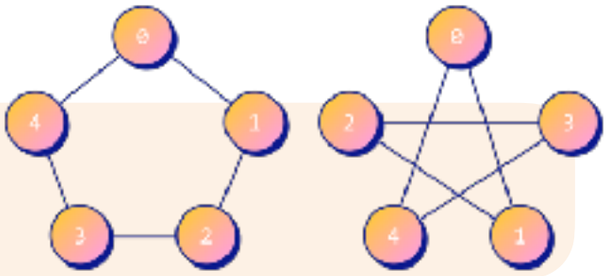
- to produce a finite and dimensionally constant representation of the graph
- Which is usually produced by a **min**, **sum** or **max** operator

[Image extracted from] M. M. Bronstein, J. Bruna, T. Cohen, P. Velickovic, Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges (2021)

Graph Neural Networks

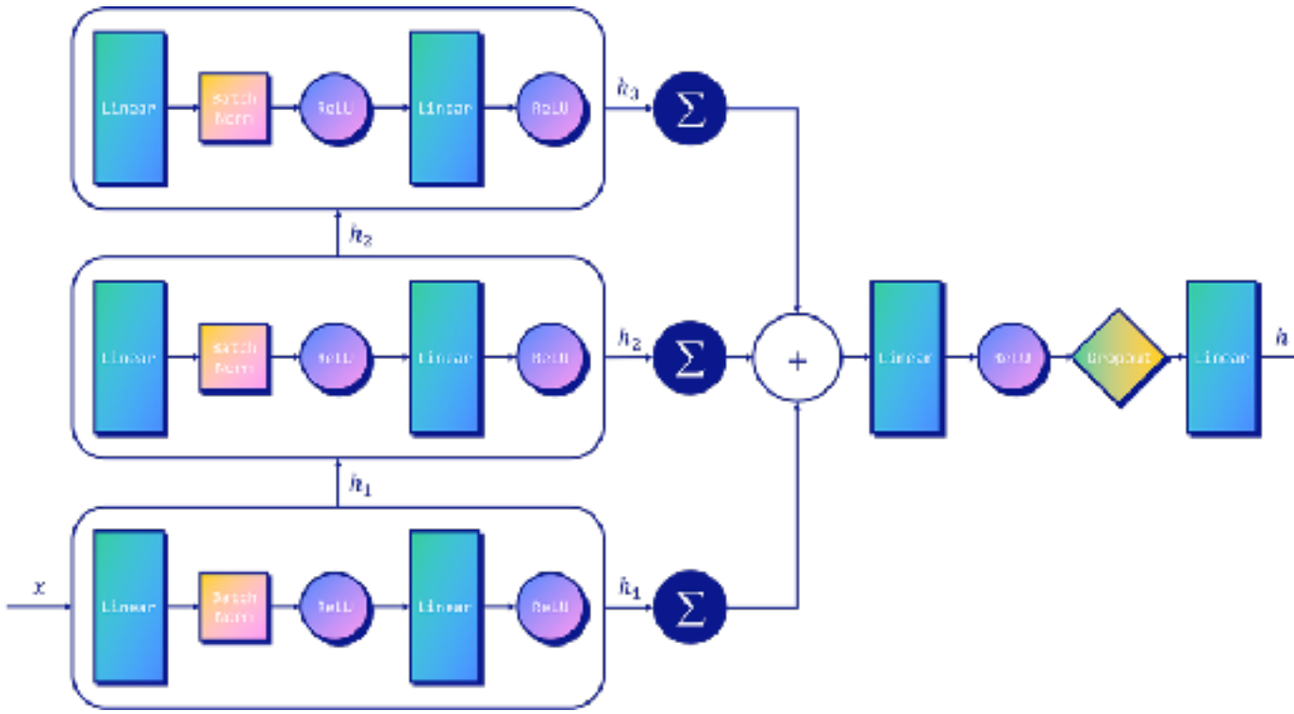
A relevant baseline: GIN

Based on the idea of the Weisfeiler-Lehman isomorphism test



$$h_i = MLP\left((1+\varepsilon)\cdot x_i + \sum_{j\in\mathcal{N}_i} x_j\right)$$

Combine node information and neighbors

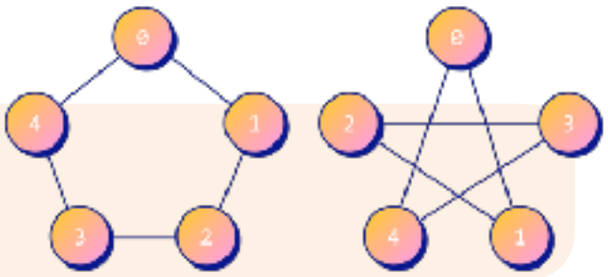


[Image extracted from] https://towardsdatascience.com/how-to-design-the-most-powerful-graph-neural-network-3d18b07a6e66 (2021)

Graph Neural Networks

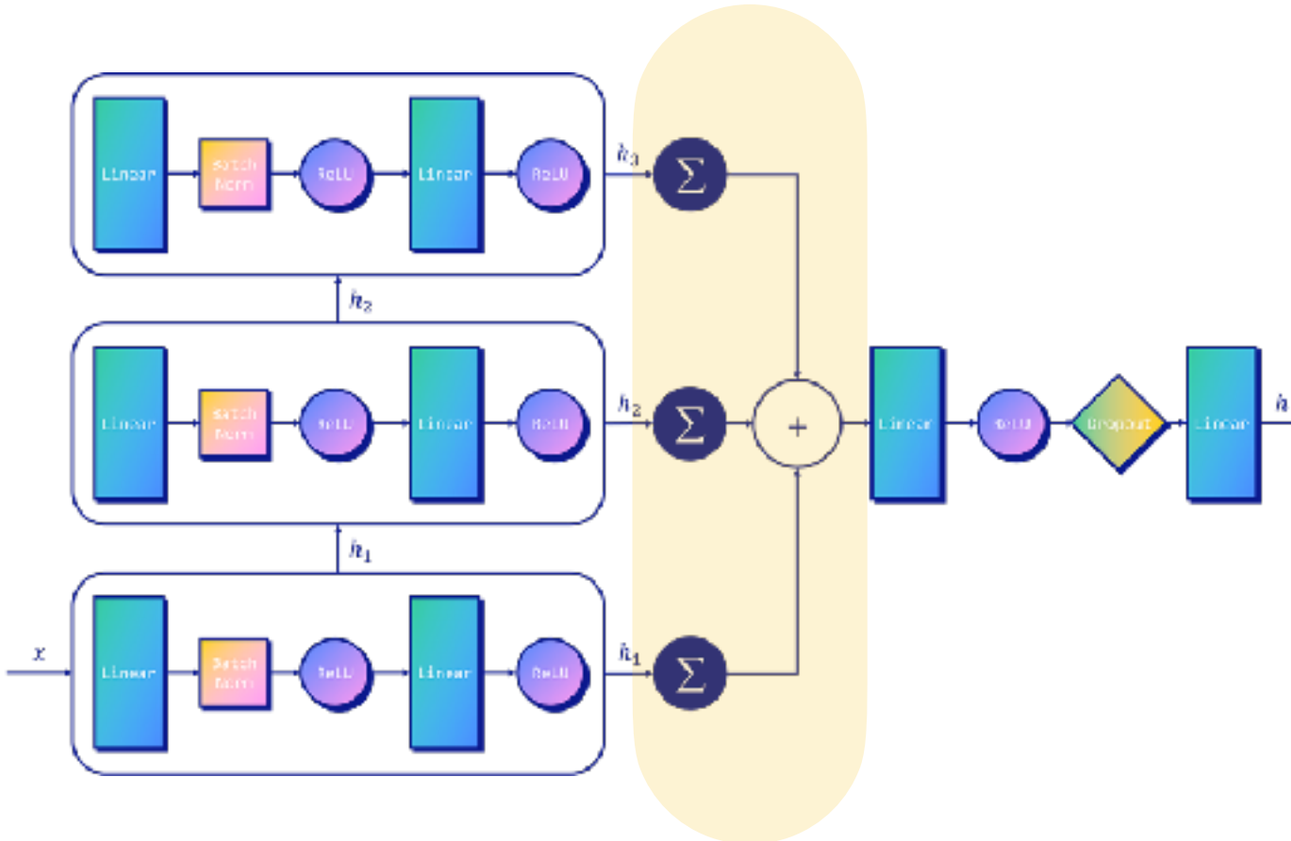
A relevant baseline: GIN

Based on the idea of the Weisfeiler-Lehman isomorphism test



$$h_i = MLP\left((1+\varepsilon)\cdot x_i + \sum_{j \in \mathcal{N}_i} x_j\right)$$

Combine node information and neighbors



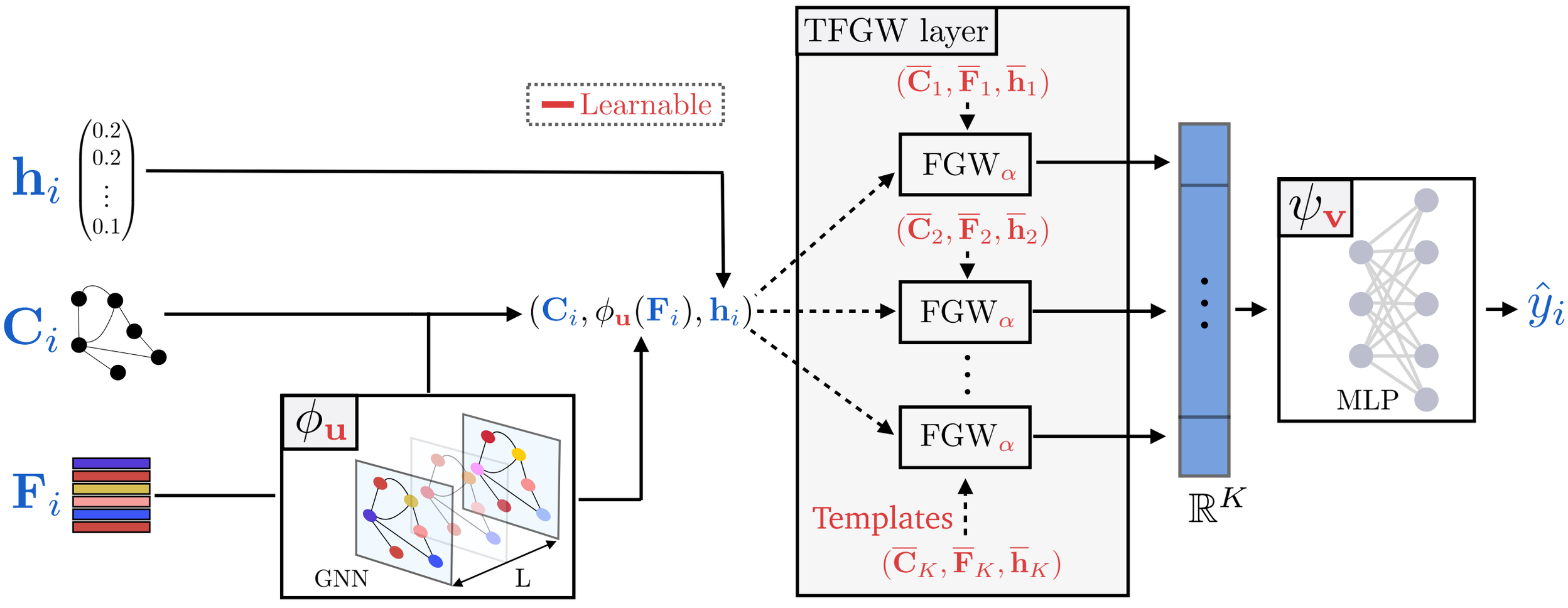
Global Pooling

[Image extracted from] https://towardsdatascience.com/how-to-design-the-most-powerful-graph-neural-network-3d18b07a6e66 (2021)

Graph Neural Networks

A new pooling layer based on FGW

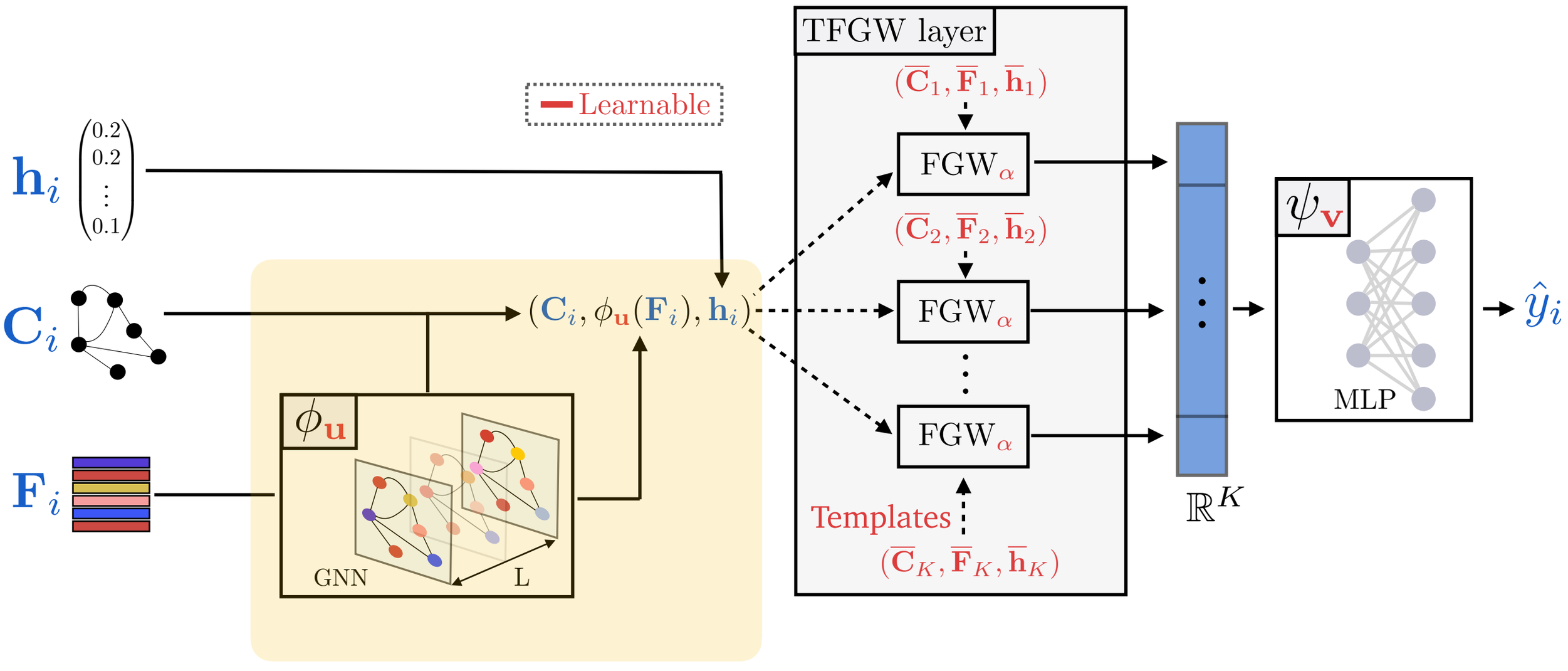
We propose a new pooling layer based on computing distances to prototypes graphs



Graph Neural Networks

A new pooling layer based on FGW

We propose a new pooling layer based on computing distances to prototypes graphs

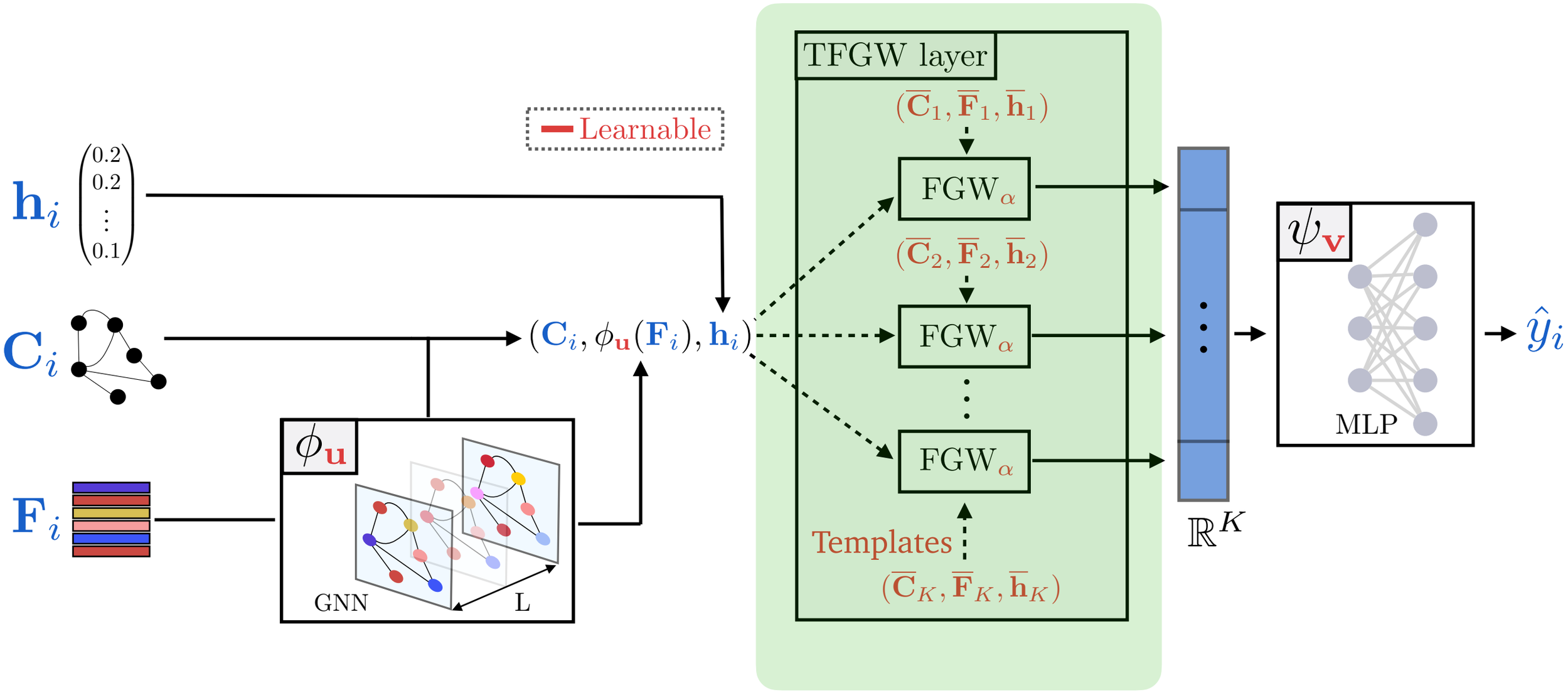


Compute more discriminant features

Graph Neural Networks

A new pooling layer based on FGW

We propose a new pooling layer based on computing distances to prototypes graphs



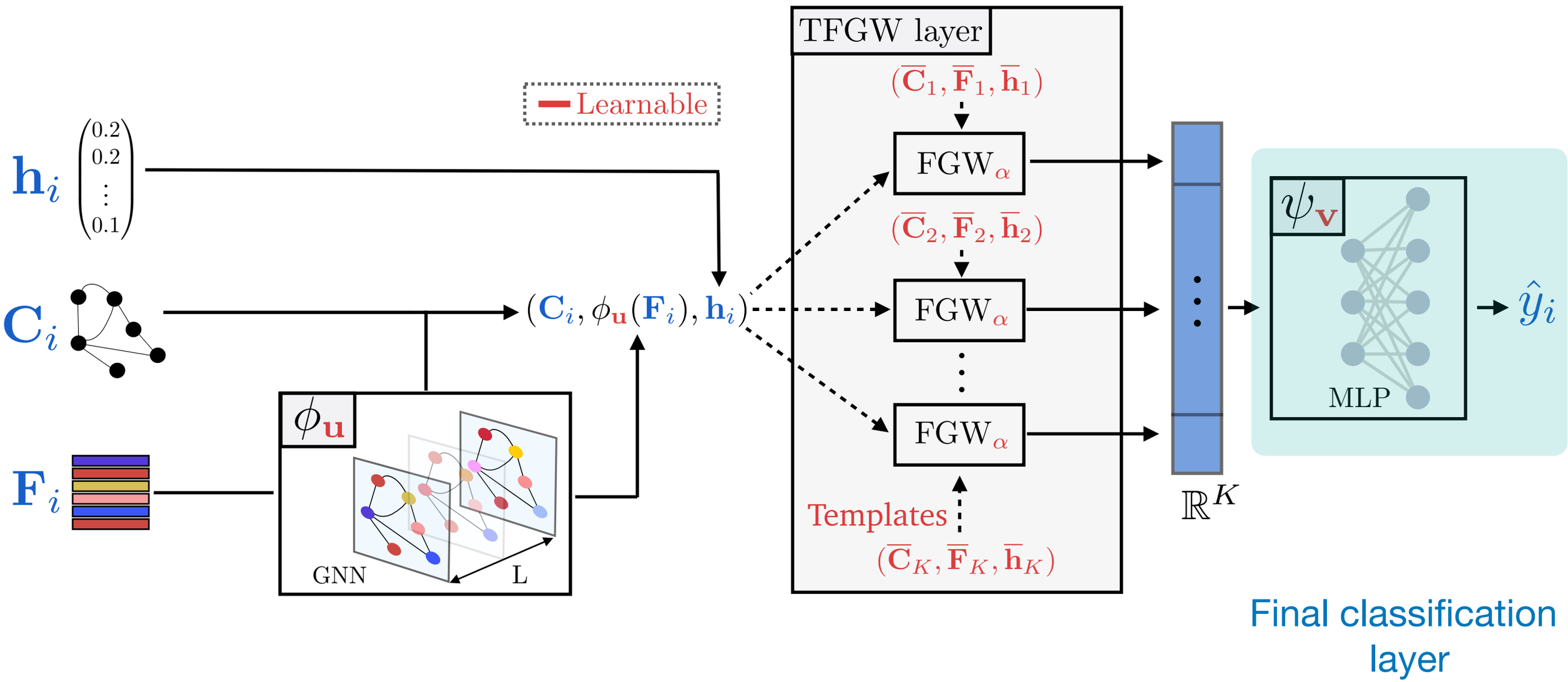
FGW distances to K template graphs

(Note that the α parameter can be learnt)

Graph Neural Networks

A new pooling layer based on FGW

We propose a new pooling layer based on computing distances to prototypes graphs



Final classification layer

Graph Neural Networks

A new pooling layer based on FGW

TFGW is SOTA on graph classification

	MUTAG	PTC	ENZYMES	PROTEIN	NCI1	IMDB-B	IMDB-M	COLLAB
Best competitor	OTGNN	OTGNN	FGW	OTGNN	WWL	WWL	WWL	WWL
accuracy (%)	92.1	68.0	72.2	78.0	85.7	71.6	52.6	81.4
TFGW+GIN (ours)	96.4	72.4	75.1	82.9	88.1	78.3	56.8	84.3
accuracy gain	**+4.3**	**+4.4**	**+2.9**	**+4.9**	**+2.4**	**+6.7**	**+4.2**	**+2.9**

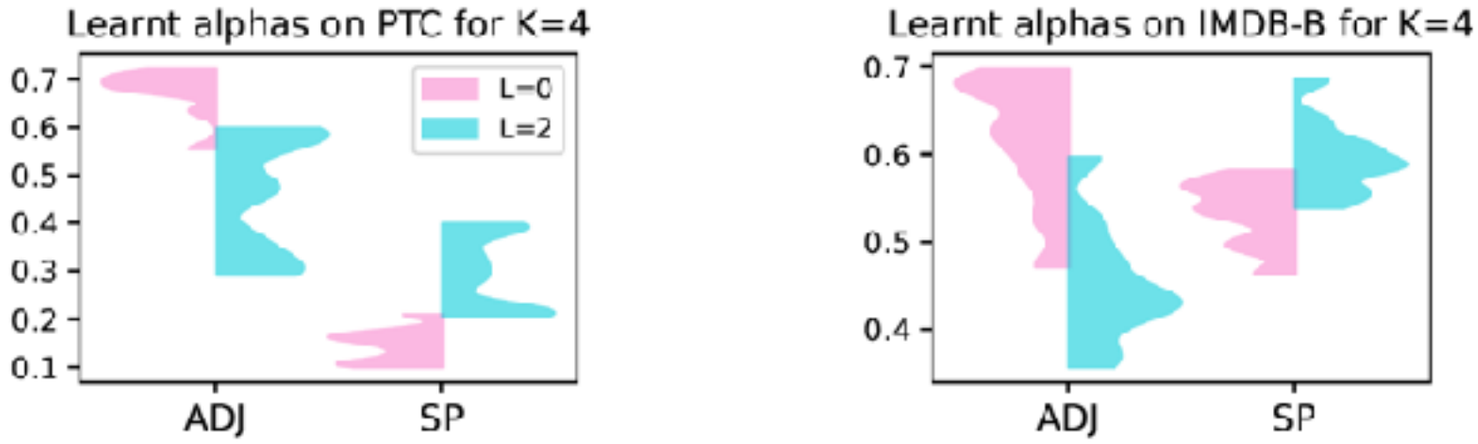
TFGW can be plugged with any GNNs that work on features

category	model	MUTAG	PTC	PROTEIN
Ours	TFGW ADJ (L=2)	95.4(3.5)	68.7(5.8)	**83.4(2.8)**
	TFGW SP (L=2)	96.2(3.0)	67.9(5.8)	82.6(2.9)
$\phi_{\boldsymbol{u}} = \text{GAT}$	TFGW ADJ (L=1)	94.8(3.1)	66.9(5.4)	82.1(3.3)
	TFGW SP (L=1)	**96.4(3.3)**	68.3(6.0)	82.3(3.1)
Ours	TFGW ADJ (L=2)	**96.4(3.3)**	**72.4(5.7)**	82.9(2.7)
	TFGW SP (L=2)	94.8(3.5)	70.8(6.3)	82.0((3.0))
$\phi_{\boldsymbol{u}} = \text{GIN}$	TFGW ADJ (L=1)	94.8(3.1)	68.7(5.8)	81.5(2.8)
	TFGW SP (L=1)	95.4(3.5)	70.9(5.5)	82.1(3.4)
sum	GAT (L=4)	91.2(2.8)	50.9(5.8)	77.6(2.7)
pooling	GIN (L=4)	90.1(4.4)	63.1(3.9)	76.2(2.8)

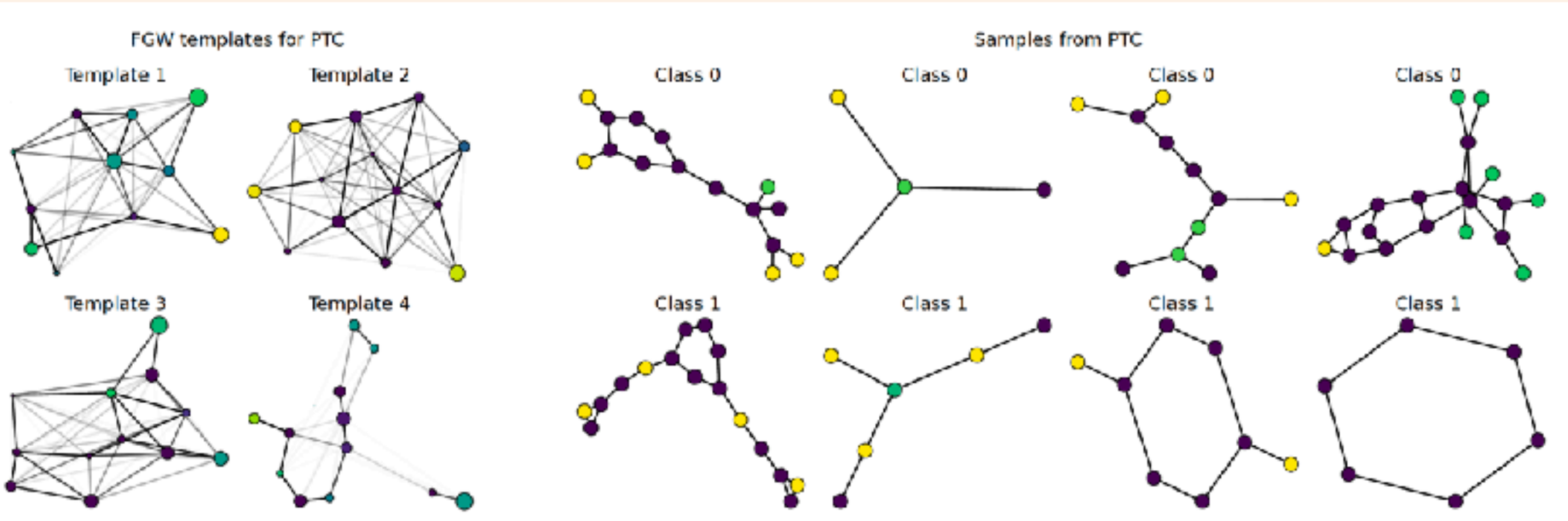
Graph Neural Networks

A new pooling layer based on FGW

Balance between structure and features: we need a mix of the two !



Are the templates interpretable ? Unfortunately not …..



Optimal Transport for Graph-signal processing

Conclusion

Conclusion

- Optimal transport is a powerful, geometric tool for comparing probability distributions
  - But computationally expensive
- Fused Gromov-Wasserstein tools and its variants allow to consider at the same time the geometry and the signal
  - Can we find something else than an additive/separable distances ?
- What I did not discuss today (and hesitated)
  - Optimal Transport on Hyperbolic spaces !

POT (PYTHON OPTIMAL TRANSPORT TOOLBOX)

https://pythonot.github.io/

README.md

POT: Python Optimal Transport

pypi package 0.4.0 | build passing | docs passing

This open source Python library provide several solvers for optimization problems related to Optimal Transport for signal, image processing and machine learning.

It provides the following solvers:

- OT solver for the linear program/ Earth Movers Distance [1].
- Entropic regularization OT solver with Sinkhorn Knopp Algorithm [2] and stabilized version [9][10] with optional GPU implementation (required cudamat).
- Bregman projections for Wasserstein barycenter [3] and unmixing [4].
- Optimal transport for domain adaptation with group lasso regularization [5]
- Conditional gradient [6] and Generalized conditional gradient for regularized OT [7].
- Joint OT matrix and mapping estimation [8].
- Wasserstein Discriminant Analysis [11] (requires autograd + pymanopt).
- Gromov-Wasserstein distances and barycenters [12]

Some demonstrations (both in Python and Jupyter Notebook format) are available in the examples folder.

Installation

The library has been tested on Linux, MacOSX and Windows. It requires a C++ compiler for using the EMD solver and relies on the following Python modules:

- Numpy (>=1.11)