



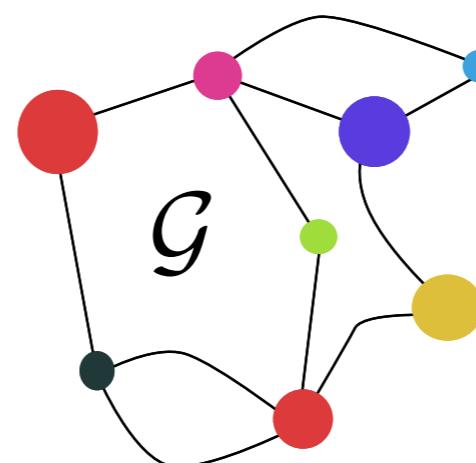
The optimal transportation problem for structured data (and heterogeneous?)

Titouan Vayer

Post-Doc

Seminar

11/01/2021



Nicolas Courty



Laetitia Chapel



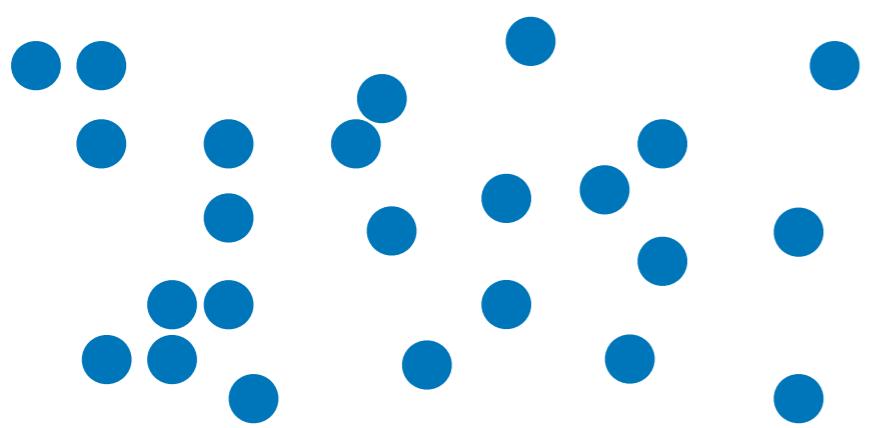
Romain Tavenard



Rémi Flamary

In short:

Machine Learning: Learn to make decision from data



In short:

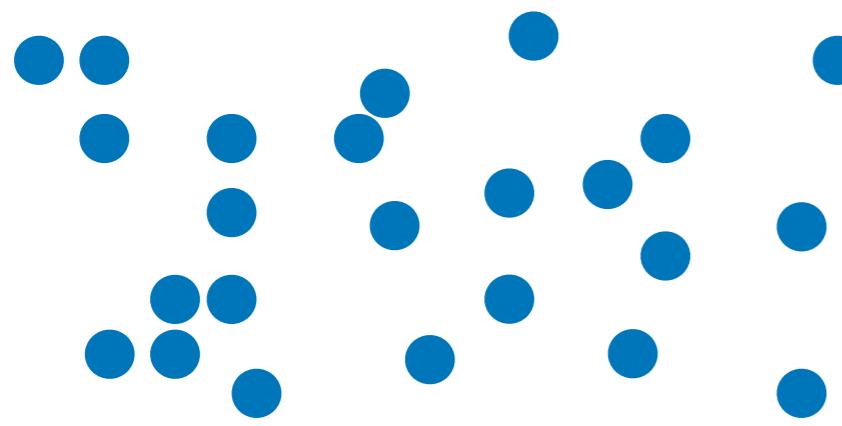
Machine Learning: Learn to make decision from **data**

| **How to represent data?**

| **How to operate on them?**

Mathematical representation

Tools which build upon this representation

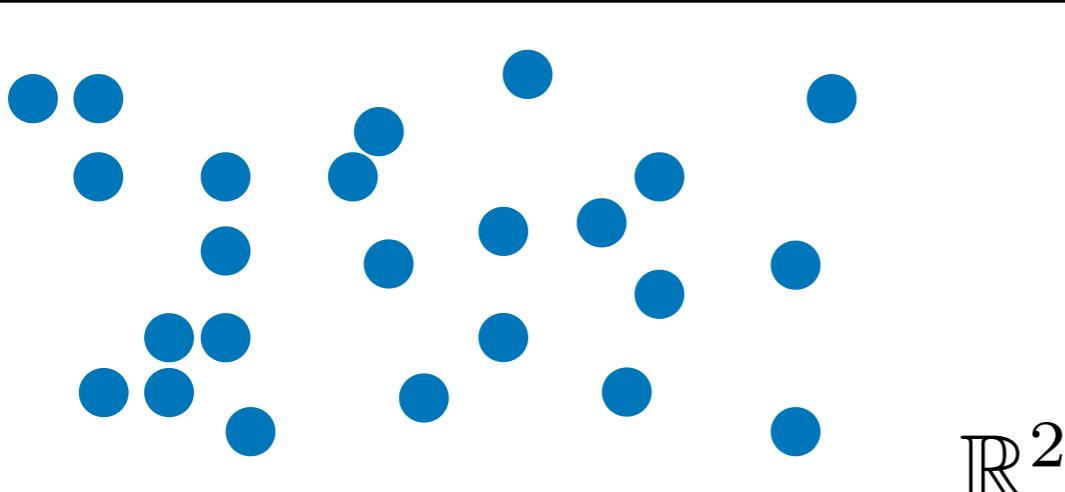


In short:

Machine Learning: Learn to make decision from data

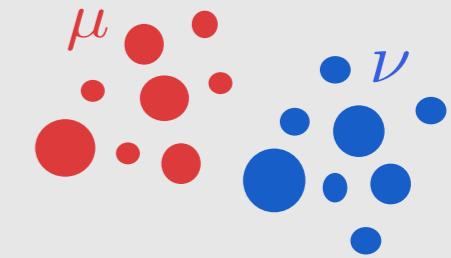
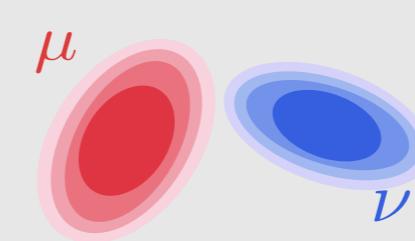
How to represent data?

How to operate on them?



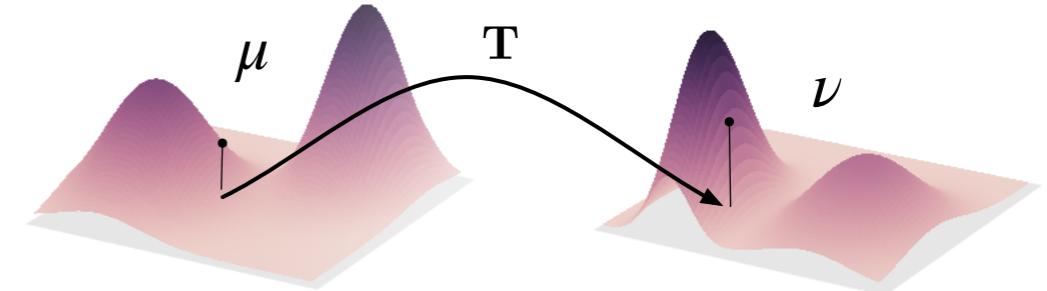
Mathematical representation

As probability distributions

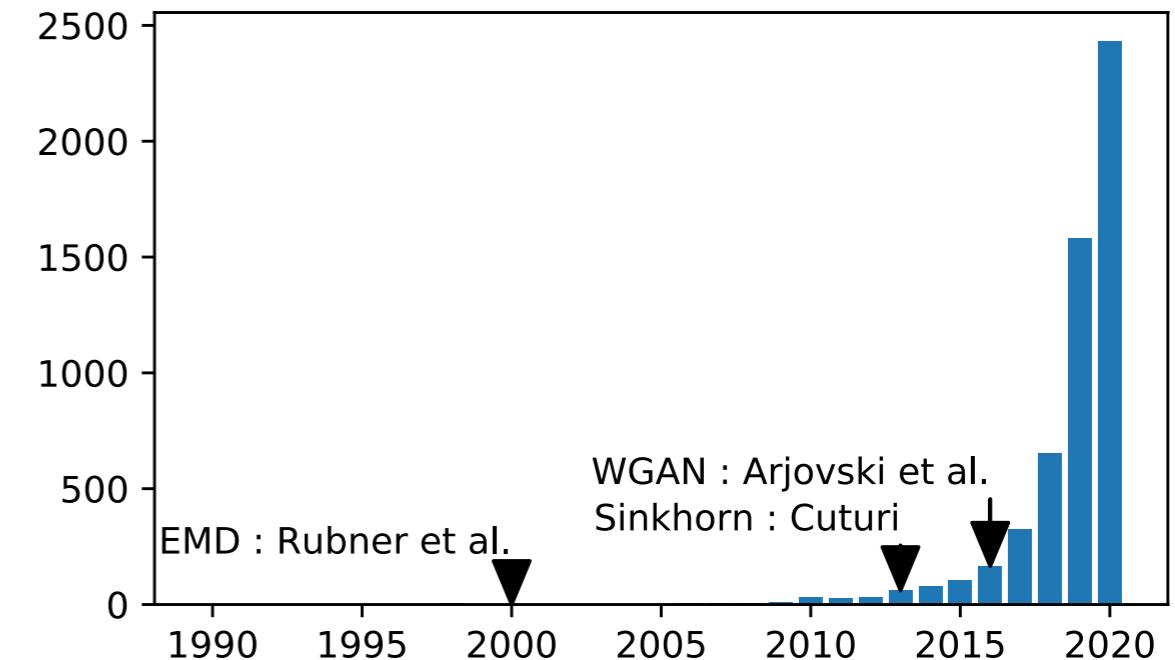


Tools which build upon this representation

Optimal Transport theory



Occurrences of OT+ML in Google Scholar



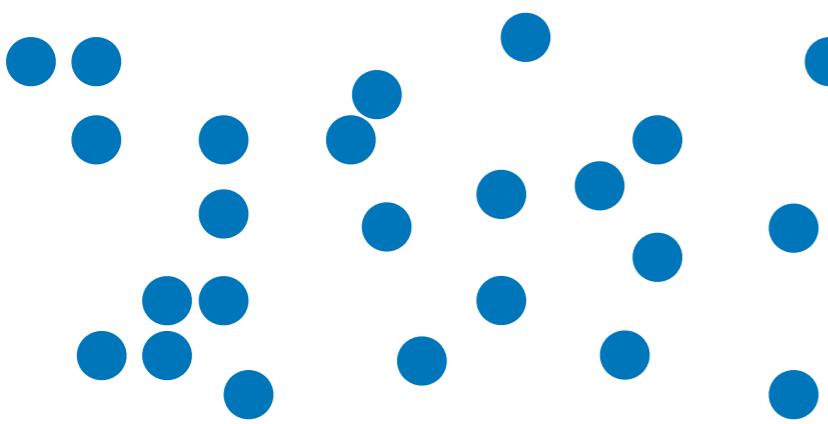
In short:

Machine Learning: Learn to make decision from **data**

| **How to represent data?**

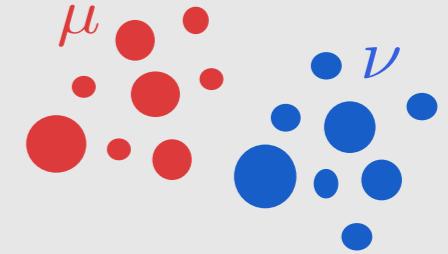
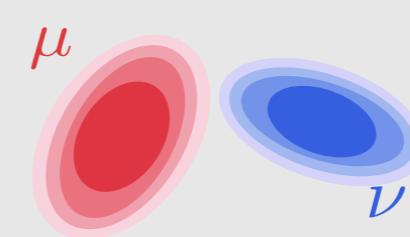
| **How to operate on them?**

Particularly challenging: highly structured data, heterogeneous spaces



Mathematical representation

As probability distributions



Tools which build upon this representation

Optimal Transport theory

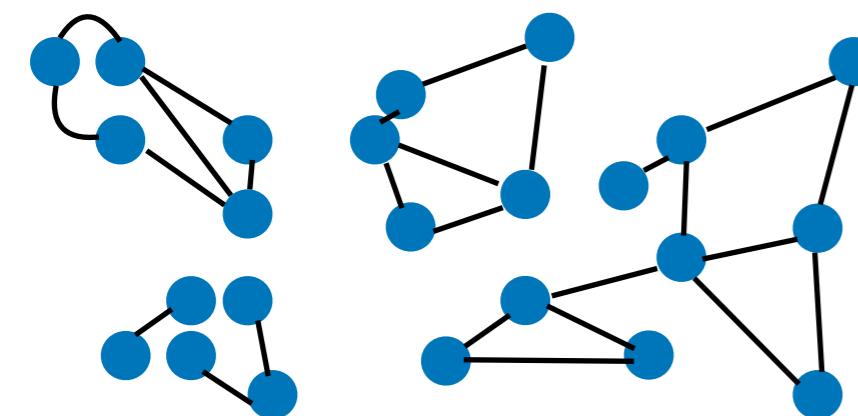
In short:

Machine Learning: Learn to make decision from **data**

| How to represent data?

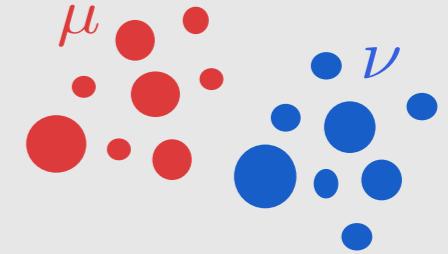
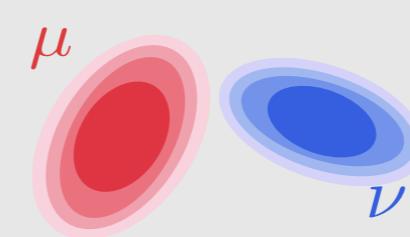
| How to operate on them?

Particularly challenging: highly structured data, heterogeneous spaces



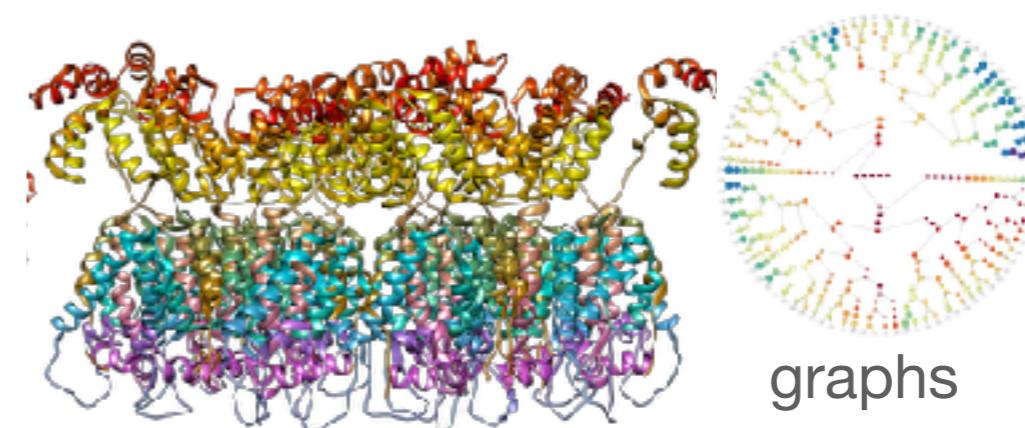
Mathematical representation

As probability distributions



Tools which build upon this representation

Optimal Transport theory



molecules, sequences..

In short:

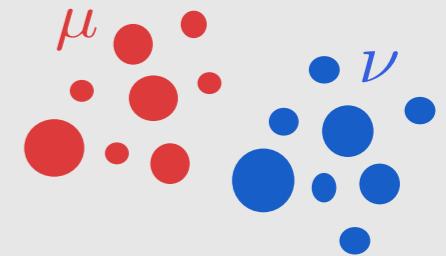
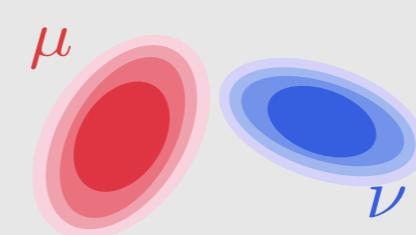
Machine Learning: Learn to make decision from **data**

How to represent data?

How to operate on them?

Mathematical representation

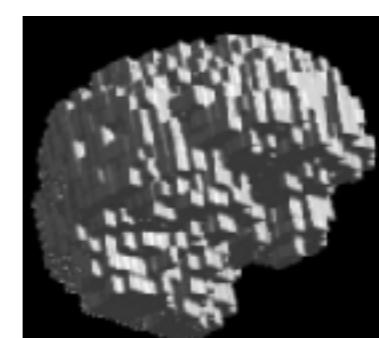
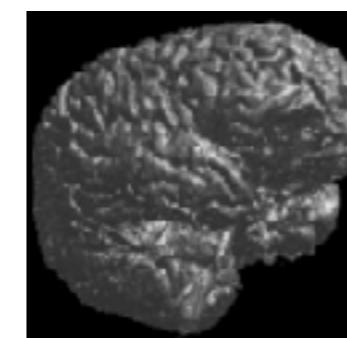
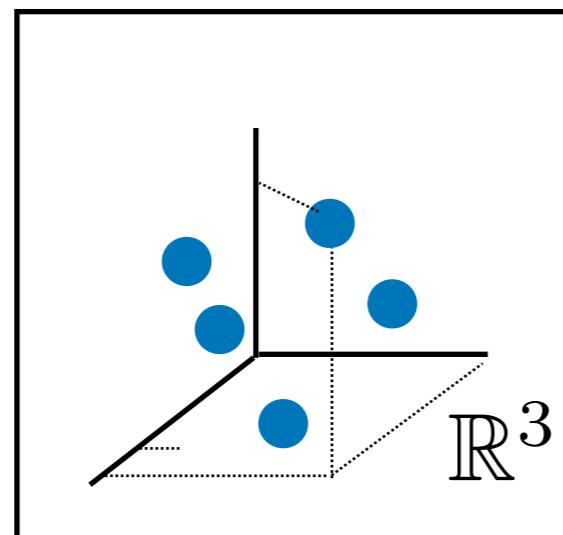
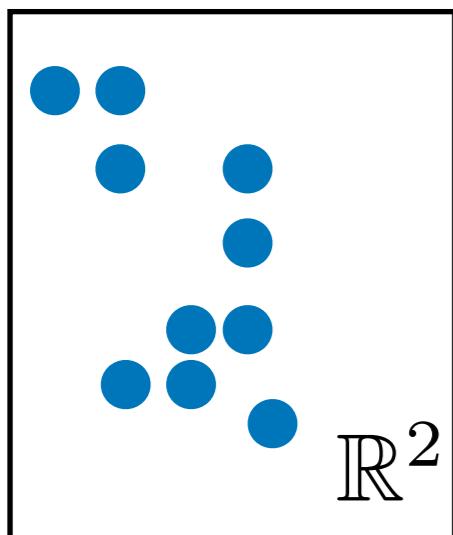
As probability distributions



Tools which build upon this representation

Optimal Transport theory

Particularly challenging: highly structured data, **heterogeneous spaces**



high & low resolution images

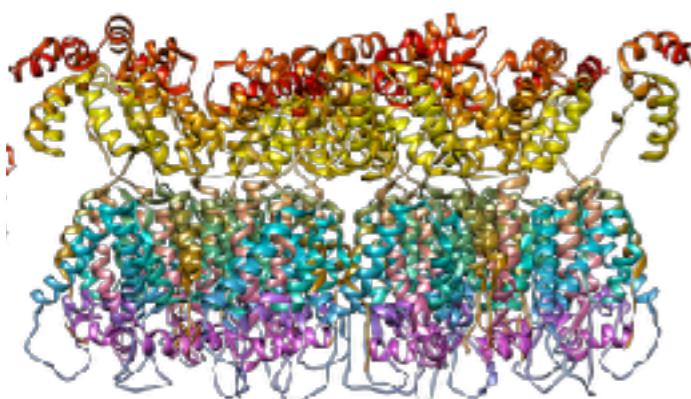
In short:

Machine Learning: Learn to make decision from data

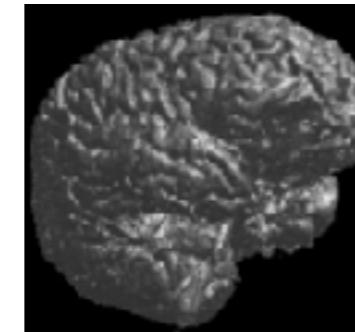
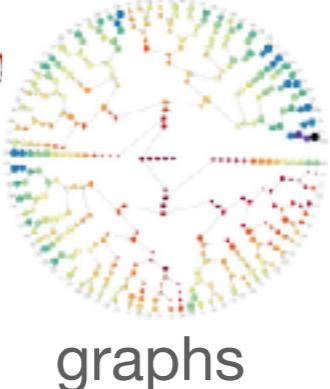
| How to represent data?

| How to operate on them?

Particularly challenging: highly structured data, heterogeneous spaces



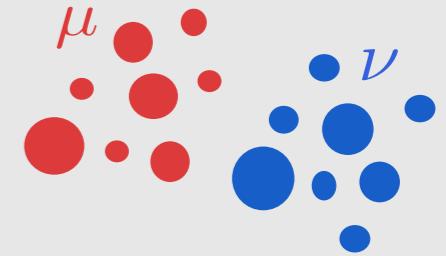
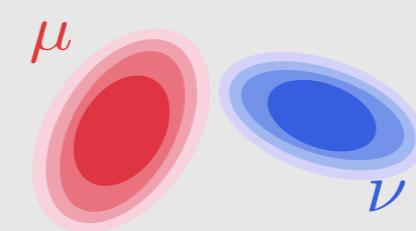
molecules, sequences..



high & low resolution images

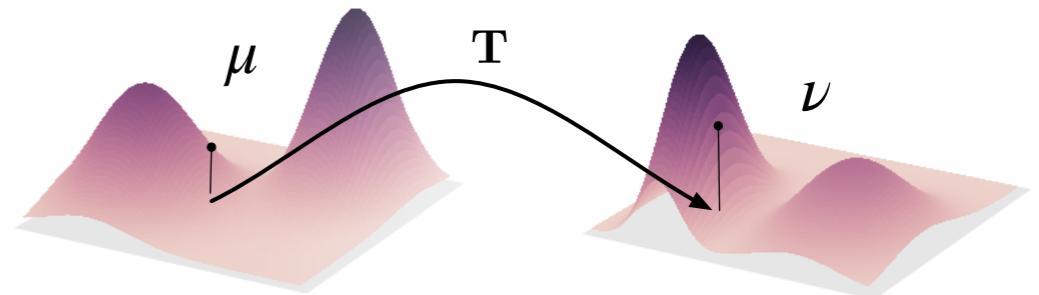
Mathematical representation

As probability distributions



Tools which build upon this representation

Optimal Transport theory



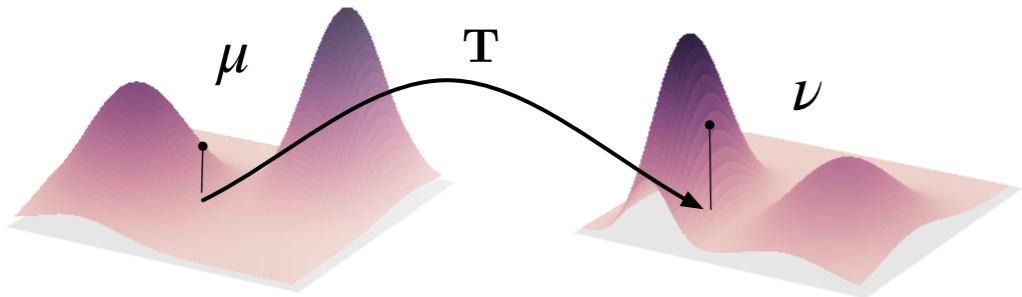
Use + Develop the Optimal transport theory in this challenging scenario

| Applicability

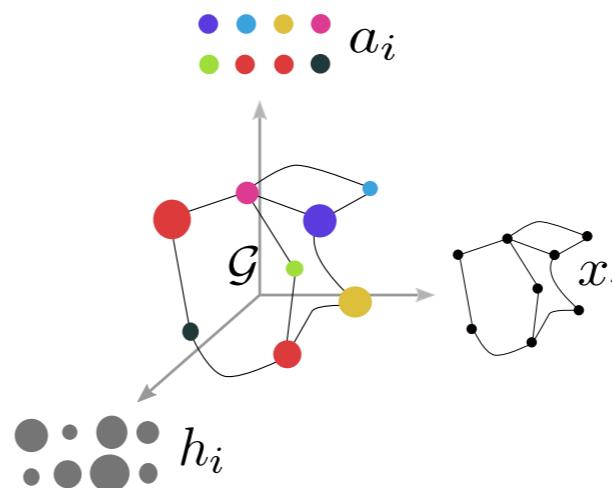
| Mathematical foundations

Overview of the talk

Part I: Optimal Transport « in short »



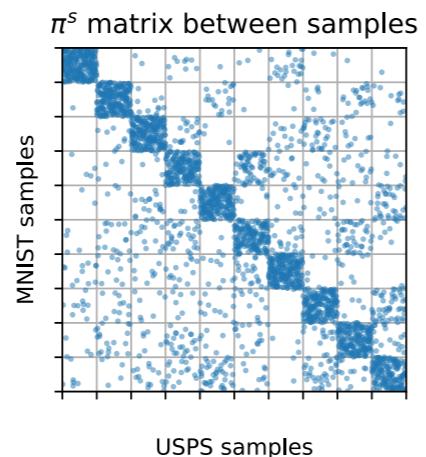
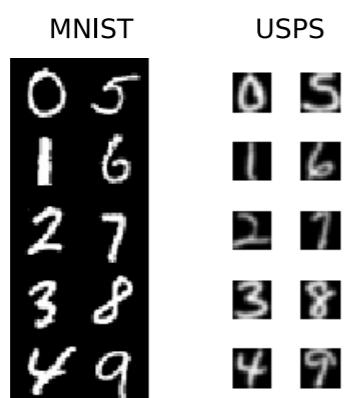
Part II: Optimal Transport for structured data



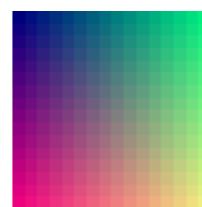
$$\left. \begin{array}{l} \text{---} \\ \text{---} \end{array} \right\} \mu = \sum_i h_i \delta_{(x_i, a_i)}$$
$$\left. \begin{array}{l} \text{---} \\ \text{---} \end{array} \right\} \mu_A = \sum_i h_i \delta_{a_i}$$
$$\left. \begin{array}{l} \text{---} \\ \text{---} \end{array} \right\} \mu_X = \sum_i h_i \delta_{x_i}$$

ICML' 2019

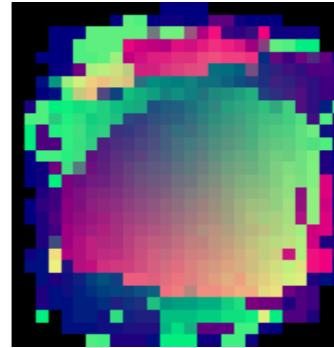
Part III: CO-Optimal Transport



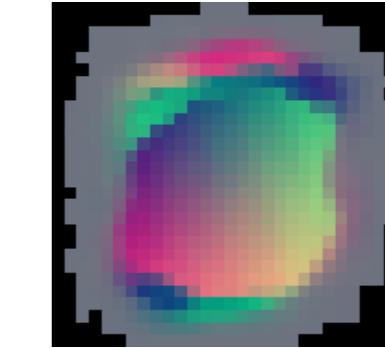
USPS colored coded pixels



MNIST pixels through π^v



MNIST pixels through entropic π^v



NeurIPS' 2020

From linear Optimal Transport to Gromov-Wasserstein

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

From linear Optimal Transport...

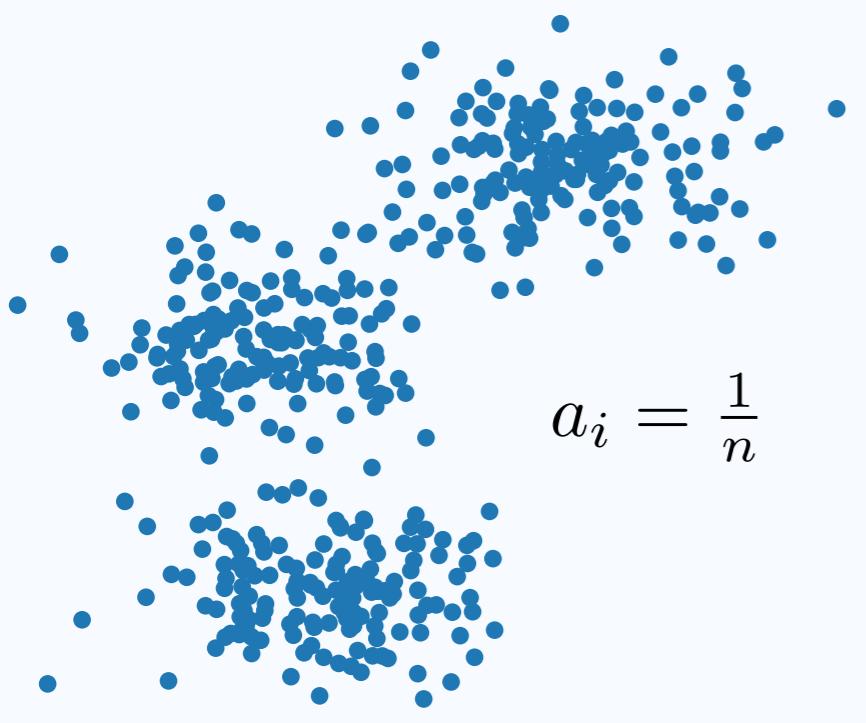
Why do we care about probability distributions?

Measure and probability distributions are at the core of Machine learning

A point of view on the data

Data: $(\mathbf{x}_i)_{i \in \llbracket n \rrbracket}$; $\mathbf{x}_i \in \mathbb{R}^d \longrightarrow$ A probability distribution describing the data

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



$$a_i = \frac{1}{n}$$

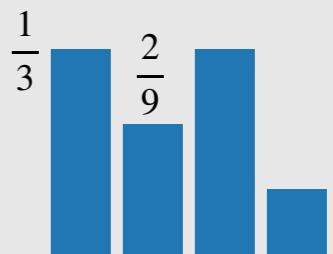
(point clouds)

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

Probability simplex

$$\mathbf{a} = (a_i)_{i \in \llbracket n \rrbracket} \in \Sigma_n$$

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



From linear Optimal Transport...

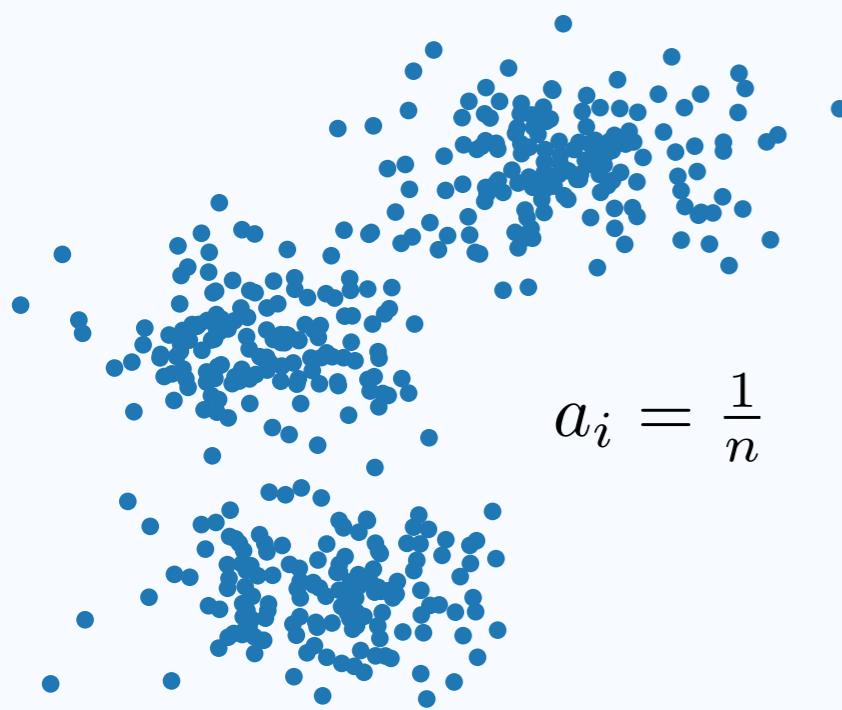
Why do we care about probability distributions?

Measure and probability distributions are at the core of Machine learning

A point of view on the data

Data: $(\mathbf{x}_i)_{i \in \llbracket n \rrbracket}$; $\mathbf{x}_i \in \mathbb{R}^d \longrightarrow$ A probability distribution describing the data

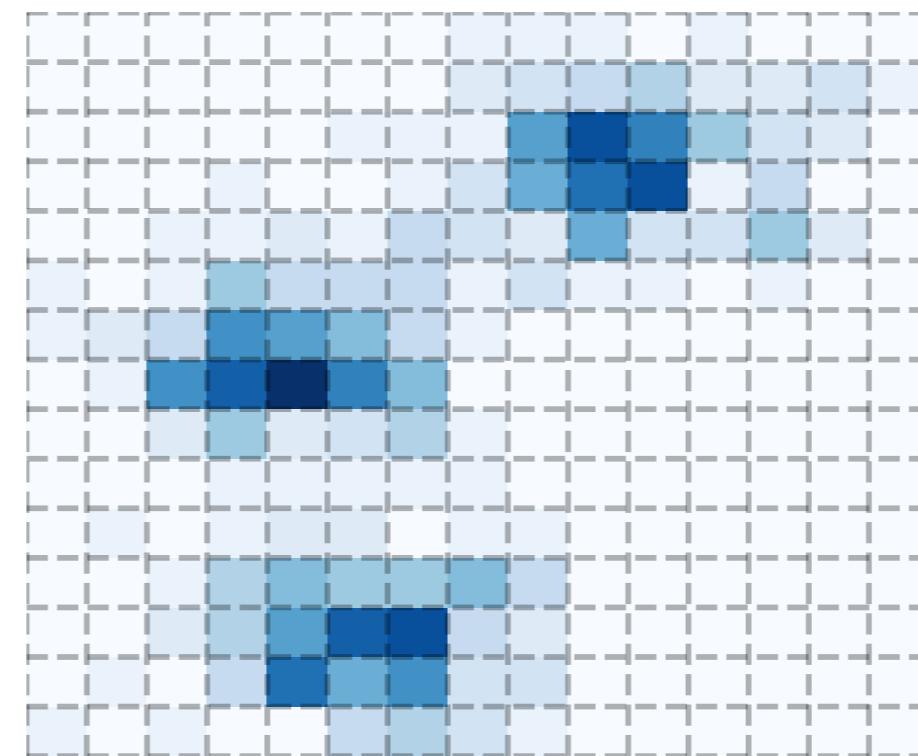
Lagrangian: $\sum_{i=1}^n a_i \delta_{x_i}$



(point clouds)

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

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



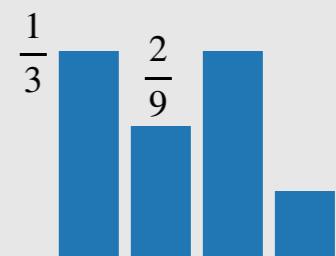
(histograms)

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

Probability simplex

$$\mathbf{a} = (a_i)_{i \in \llbracket n \rrbracket} \in \Sigma_n$$

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



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: $(\mathbf{x}_i)_{i \in \llbracket n \rrbracket}$; $\mathbf{x}_i \in \mathbb{R}^d \longrightarrow$ A probability distribution describing the data

A formalism for many machine learning paradigms

$$\text{(ERM)} \quad \min_f \underset{(x,y) \sim \mu}{\mathbb{E}} [L(f(x), y)] \xrightarrow{\text{follow the law given by the prob.}} \mu = \frac{1}{n} \sum_{i=1}^n \delta_{(\mathbf{x}_i, y_i)}$$

From linear Optimal Transport...

Why do we care about probability distributions?

Measure and probability distributions are at the core of Machine learning

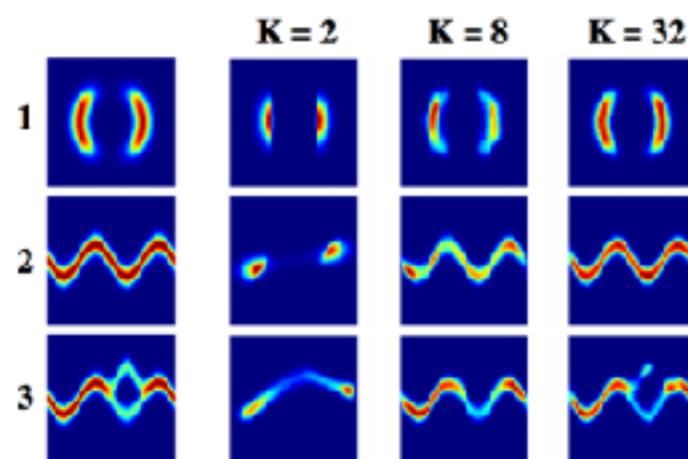
A point of view on the data

Data: $(\mathbf{x}_i)_{i \in \llbracket n \rrbracket}$; $\mathbf{x}_i \in \mathbb{R}^d \longrightarrow$ A probability distribution describing the data

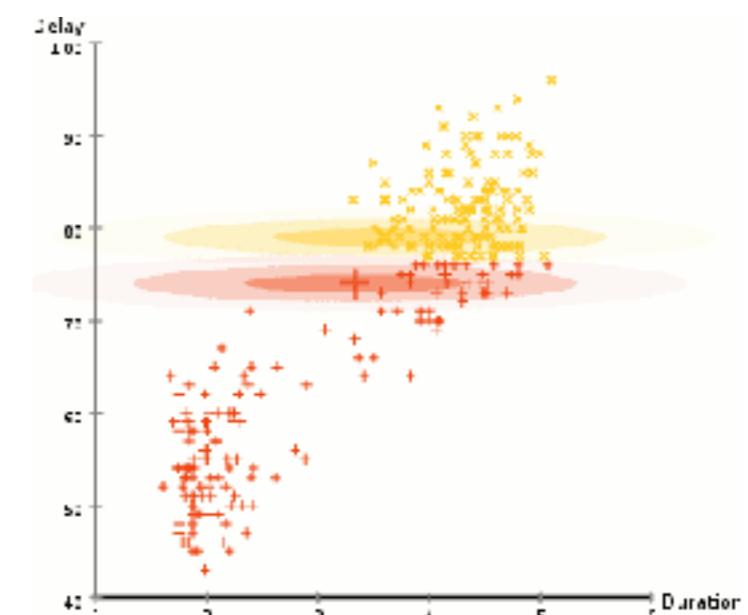
A formalism for many machine learning paradigms

$$\text{(ERM)} \quad \min_f \mathbb{E}_{(x,y) \sim \mu} [L(f(x), y)] \quad \mu = \frac{1}{n} \sum_{i=1}^n \delta_{(\mathbf{x}_i, y_i)}$$

$$\text{(Likelihood)} \quad \max_{\theta \in \Theta} \mathbb{E}_{\mathbf{x} \sim \mu} [\log(\mathbb{P}_{\theta}(\mathbf{x}))] \quad \mu = \frac{1}{n} \sum_{i=1}^n \delta_{\mathbf{x}_i}$$



[Rezende 2016]



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: $(\mathbf{x}_i)_{i \in \llbracket n \rrbracket}$; $\mathbf{x}_i \in \mathbb{R}^d$ \longrightarrow A probability distribution describing the data

A formalism for many machine learning paradigms

$$\text{(ERM)} \quad \min_f \mathbb{E}_{(x,y) \sim \mu} [L(f(x), y)] \quad \mu = \frac{1}{n} \sum_{i=1}^n \delta_{(\mathbf{x}_i, y_i)}$$

$$\text{(Likelihood)} \quad \max_{\theta \in \Theta} \mathbb{E}_{\mathbf{x} \sim \mu} [\log(\mathbb{P}_{\theta}(\mathbf{x}))] \quad \mu = \frac{1}{n} \sum_{i=1}^n \delta_{\mathbf{x}_i}$$

$$\text{(GAN)} \quad \min_{\theta \in \Theta} D(\mu_{\theta}, \nu)$$



[Radford 2015]

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: $(\mathbf{x}_i)_{i \in \llbracket n \rrbracket}$; $\mathbf{x}_i \in \mathbb{R}^d \longrightarrow$ A probability distribution describing the data

A formalism for many machine learning paradigms

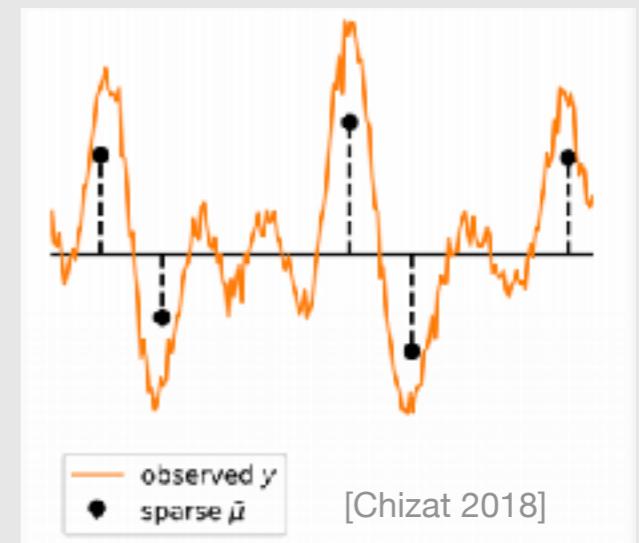
| (ERM) $\min_f \mathbb{E}_{(x,y) \sim \mu} [L(f(x), y)]$ $\mu = \frac{1}{n} \sum_{i=1}^n \delta_{(\mathbf{x}_i, y_i)}$

| (Likelihood) $\max_{\theta \in \Theta} \mathbb{E}_{\mathbf{x} \sim \mu} [\log(\mathbb{P}_{\theta}(\mathbf{x}))]$ $\mu = \frac{1}{n} \sum_{i=1}^n \delta_{\mathbf{x}_i}$

| (GAN) $\min_{\theta \in \Theta} D(\mu_{\theta}, \nu)$

| (Signal processing) Recover a sparse signal

$$\min_{\mu \in \mathcal{M}(\Theta)} \frac{1}{2} \|\mathbf{y} - \phi * \mu\|_{L^2}^2 + R(\mu) \quad \bar{\mu} = \sum_i w_i \delta_{\theta_i}$$



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: $(\mathbf{x}_i)_{i \in \llbracket n \rrbracket}$; $\mathbf{x}_i \in \mathbb{R}^d \longrightarrow$ A probability distribution describing the data

A formalism for many machine learning paradigms

$$\text{(ERM)} \quad \min_f \mathbb{E}_{(x,y) \sim \mu} [L(f(x), y)] \quad \mu = \frac{1}{n} \sum_{i=1}^n \delta_{(\mathbf{x}_i, y_i)}$$

$$\text{(Likelihood)} \quad \max_{\theta \in \Theta} \mathbb{E}_{\mathbf{x} \sim \mu} [\log(\mathbb{P}_{\theta}(\mathbf{x}))] \quad \mu = \frac{1}{n} \sum_{i=1}^n \delta_{\mathbf{x}_i}$$

$$\text{(GAN)} \quad \min_{\theta \in \Theta} D(\mu_{\theta}, \nu)$$

$$\text{(Signal processing)} \quad \min_{\mu \in \mathcal{M}(\Theta)} \frac{1}{2} \|\mathbf{y} - \phi * \mu\|_{L^2}^2 + R(\mu)$$

Advocates for finding an appropriate way of comparing probability distributions

From linear Optimal Transport...

Formulation



Two probability distributions

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

A cost function

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

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

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

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

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

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

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

A cost function

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

Bakeries = quantity of breads

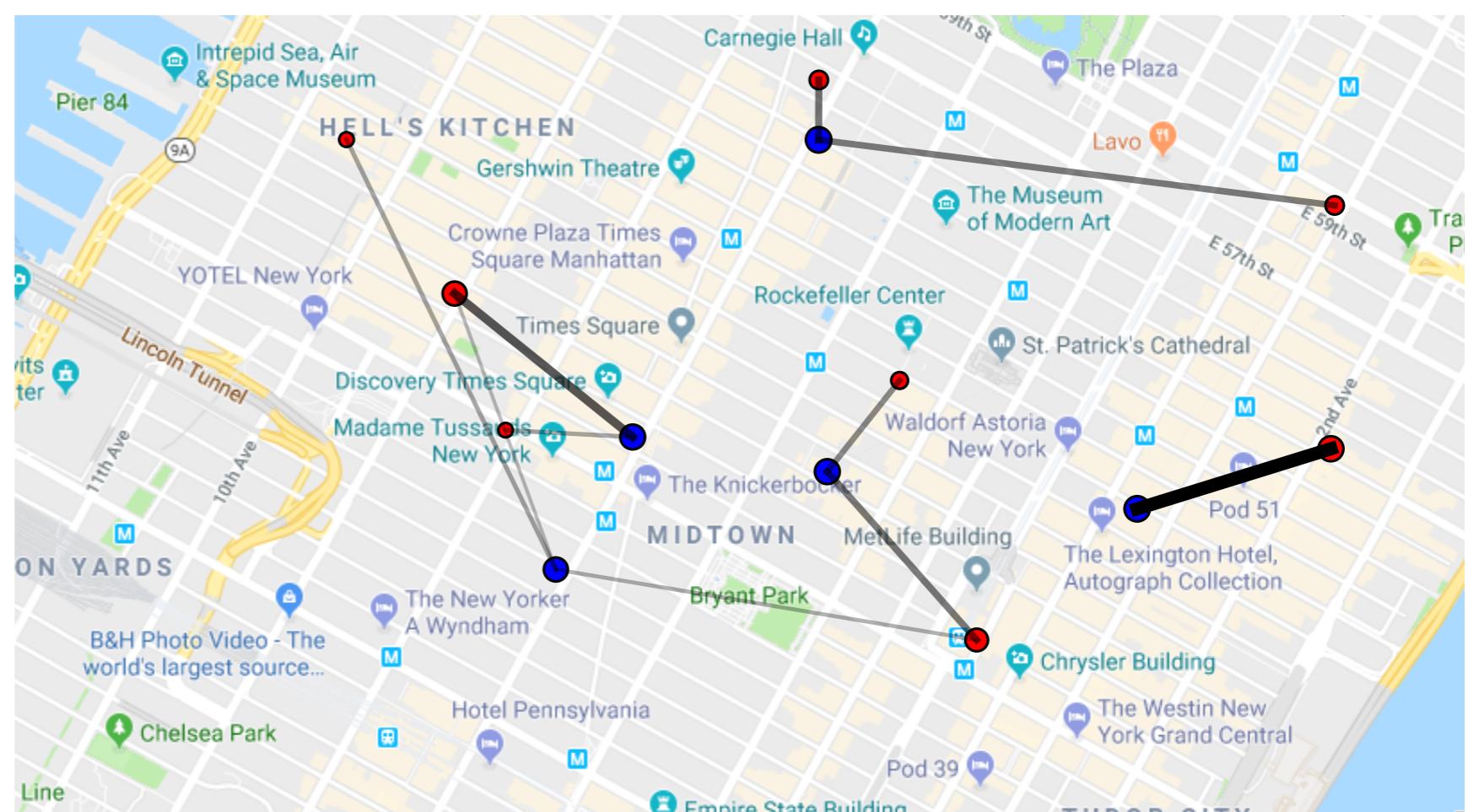
loc: x_i quantity: a_i

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

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

A cost function

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

Kantorovitch formulation

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

From linear Optimal Transport...

Kantorovitch Formulation: an example



Two probability distributions

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

A cost function

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

Kantorovitch formulation

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



Set of couplings/
transport plans

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

From linear Optimal Transport...

Kantorovitch Formulation: an example



Two probability distributions

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

A cost function

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

Kantorovitch formulation

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



How much is shifted
from x_i to y_j

From linear Optimal Transport...

Kantorovitch Formulation: an example



Two probability distributions

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

A cost function

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

Kantorovitch formulation

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



Cost of moving masses
from x_i to y_j

From linear Optimal Transport...

Kantorovitch Formulation: an example



Two probability distributions

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

A cost function

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

Kantorovitch formulation

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

Total cost

From linear Optimal Transport...

Kantorovitch Formulation: an example



Two probability distributions

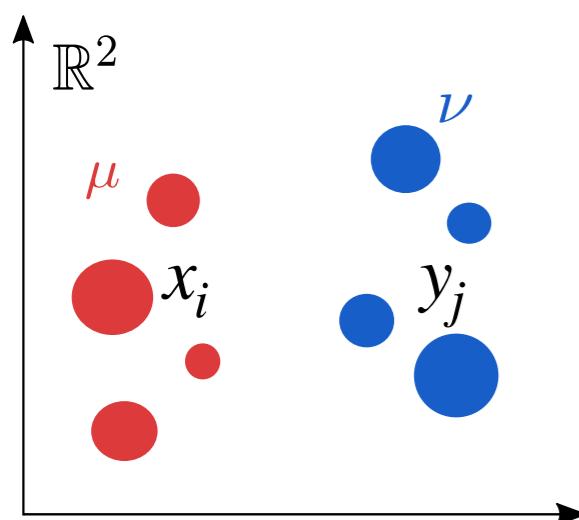
$$\mu = \sum_{i=1}^n a_i \delta_{x_i} \quad \nu = \sum_{j=1}^m b_j \delta_{y_j}$$

A cost function

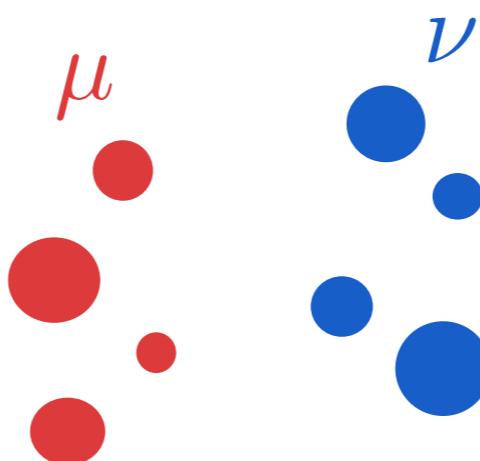
$$c(x, y) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$$

Kantorovitch formulation

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



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



From linear Optimal Transport...

Kantorovitch Formulation: an example



Two probability distributions

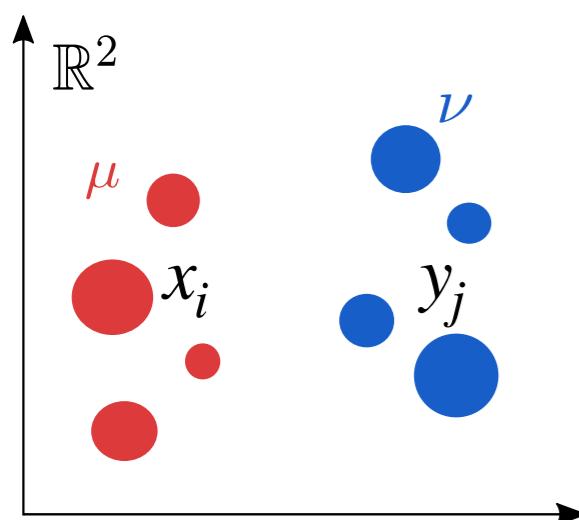
$$\mu = \sum_{i=1}^n a_i \delta_{x_i} \quad \nu = \sum_{j=1}^m b_j \delta_{y_j}$$

A cost function

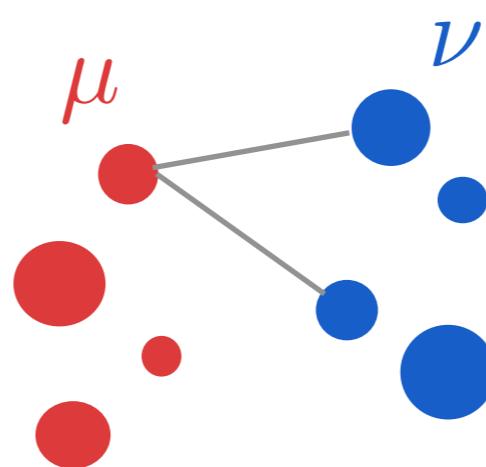
$$c(x, y) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$$

Kantorovitch formulation

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



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



From linear Optimal Transport...

Kantorovitch Formulation: an example



Two probability distributions

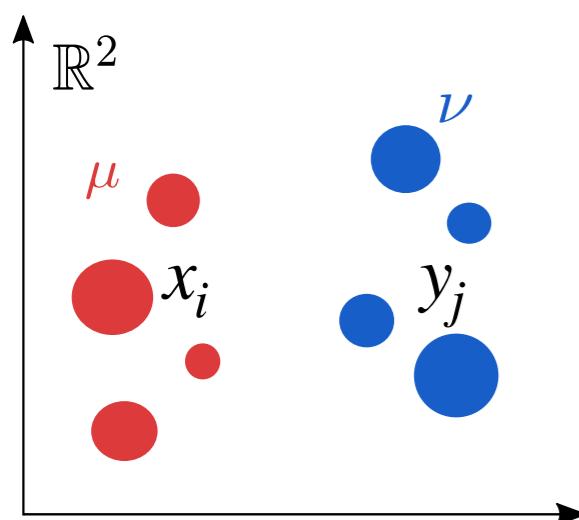
$$\mu = \sum_{i=1}^n a_i \delta_{x_i} \quad \nu = \sum_{j=1}^m b_j \delta_{y_j}$$

A cost function

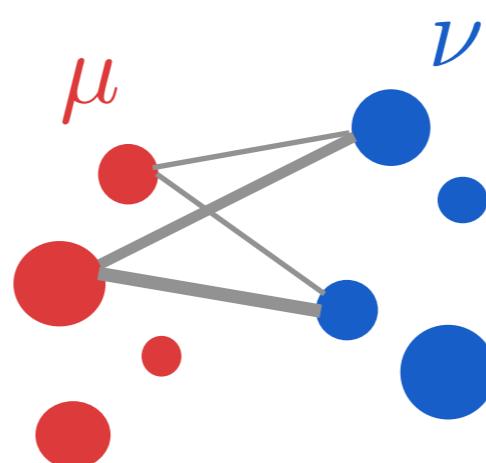
$$c(x, y) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$$

Kantorovitch formulation

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



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



From linear Optimal Transport...

Kantorovitch Formulation: an example



Two probability distributions

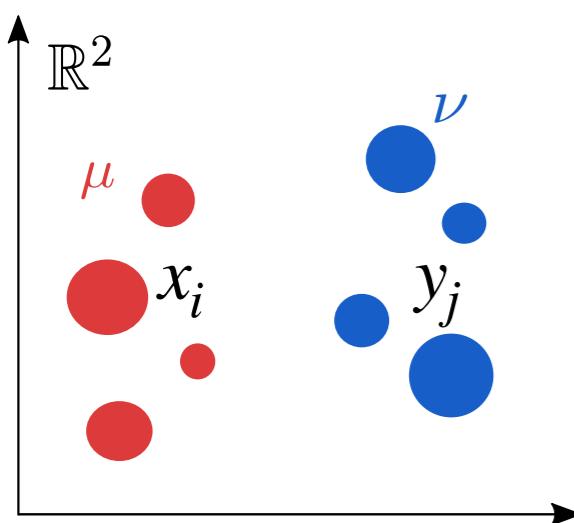
$$\mu = \sum_{i=1}^n a_i \delta_{x_i} \quad \nu = \sum_{j=1}^m b_j \delta_{y_j}$$

A cost function

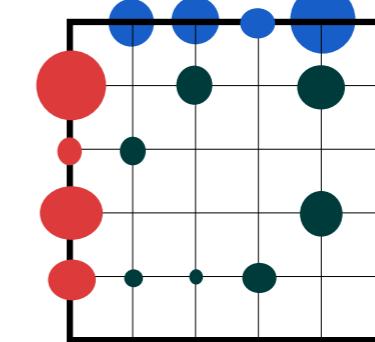
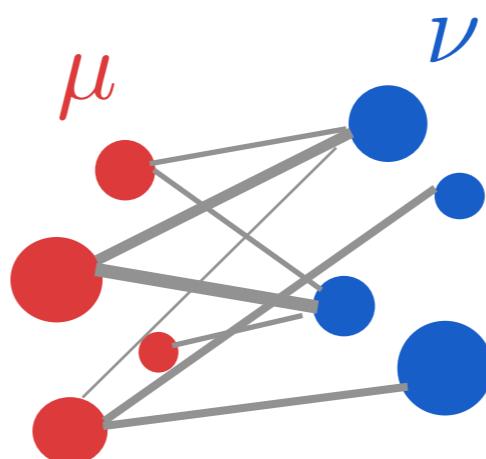
$$c(x, y) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$$

Kantorovitch formulation

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



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



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

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

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

Kantorovitch formulation

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

From linear Optimal Transport...

Wasserstein distance



Two probability distributions

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

A distance

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

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

Wasserstein distance

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

$\mathcal{P}(\Omega)$ 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

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

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

Wasserstein distance

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

$\mathcal{P}(\Omega)$ 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

$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y})$ with $\mathcal{X}, \mathcal{Y} \not\subseteq \Omega$

A cost function ?????

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

⇒ 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

$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y})$ with $\mathcal{X}, \mathcal{Y} \not\subseteq \Omega$

A cost function ?????

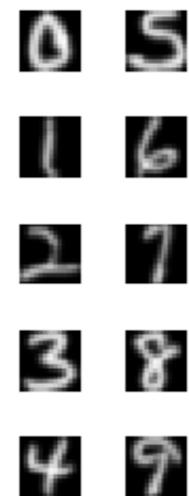
$c(x, y) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$

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

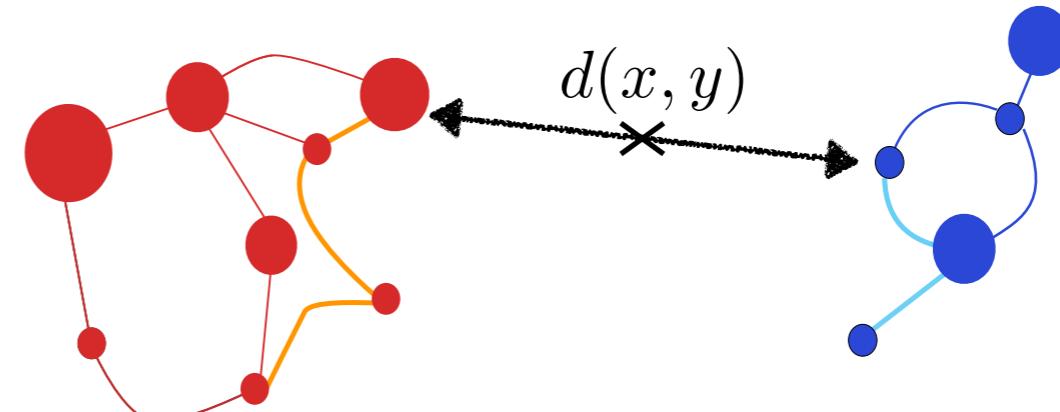
Different Euclidean spaces



USPS



Samples = nodes of different graphs



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

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

...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

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

Gromov-Wasserstein distance

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

...to Gromov-Wasserstein

Gromov-Wasserstein distance



Two probability distributions

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

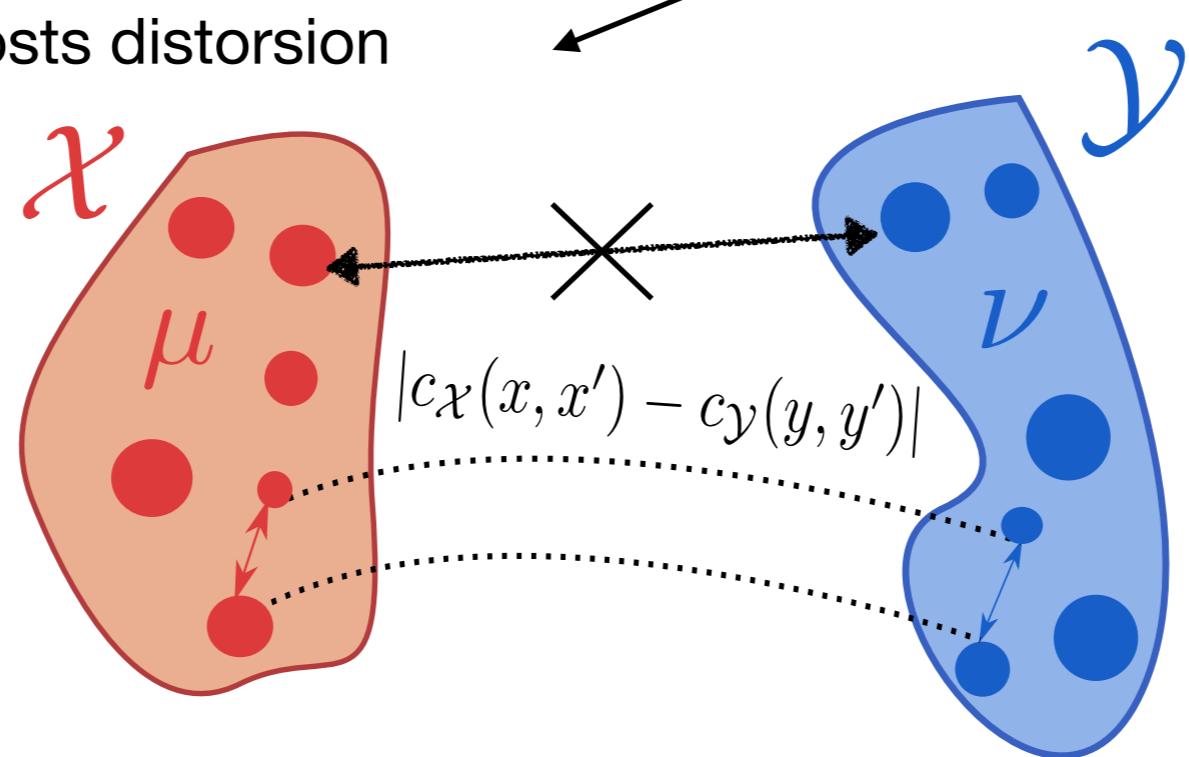
Two « intra-domain » costs

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

Gromov-Wasserstein distance

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

Measure the costs distortion



...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

$$c_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$$

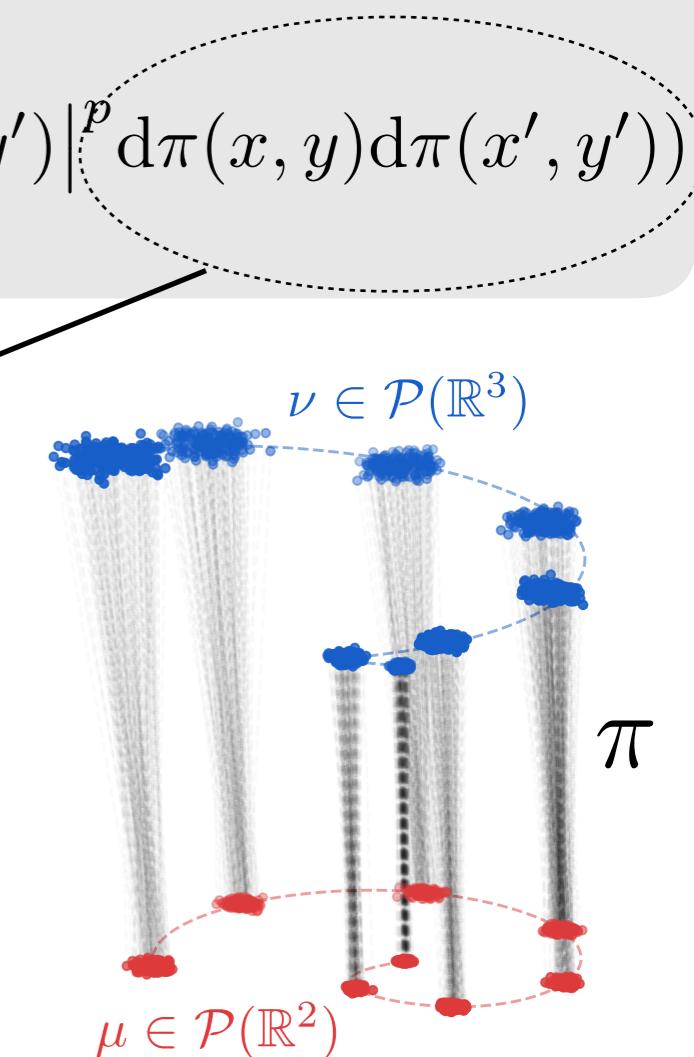
$$c_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$$

Gromov-Wasserstein distance

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

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

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

A distance w.r.t isomorphism

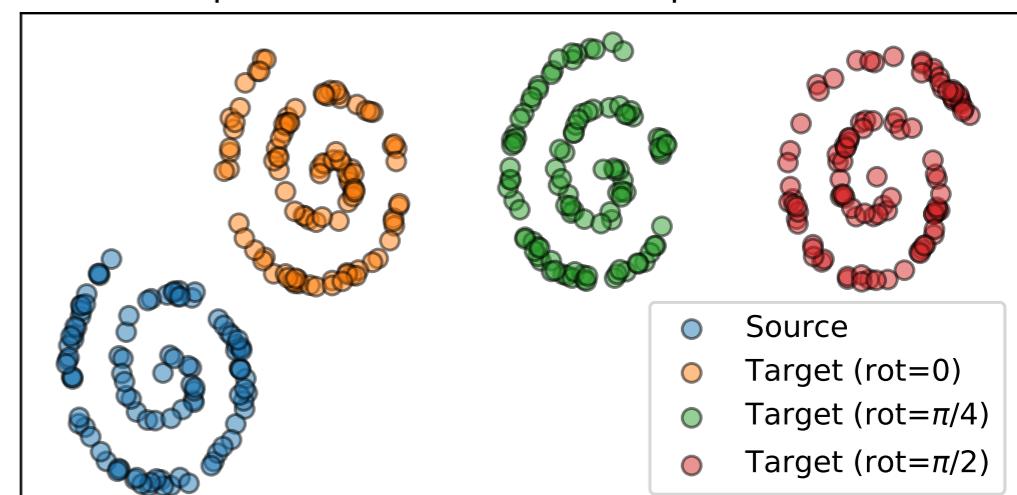
GW is a distance on the "space of all spaces":

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

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

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

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

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 $\exists \phi : \mathcal{X} \rightarrow \mathcal{Y}$

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 $\phi\#\mu$

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

...to Gromov-Wasserstein

Gromov-Wasserstein distance



Gromov-Wasserstein distance

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

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 $\exists \phi : \mathcal{X} \rightarrow \mathcal{Y}$

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

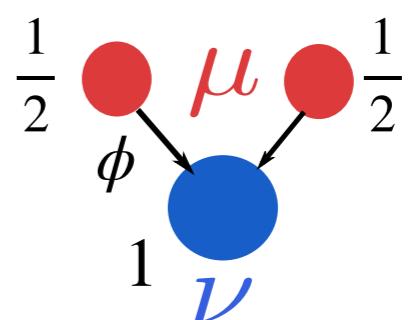
ϕ is measure-preserving: $\phi\#\mu = \nu$

(Weights are compatible)

Push-forward $\phi\#\mu$

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

Compatible



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

...to Gromov-Wasserstein

Gromov-Wasserstein distance



Gromov-Wasserstein distance

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

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 $\exists \phi : \mathcal{X} \rightarrow \mathcal{Y}$

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

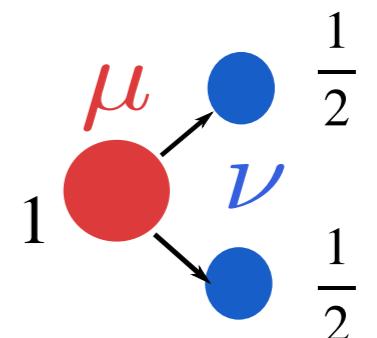
ϕ is measure-preserving: $\phi\#\mu = \nu$

(Weights are compatible)

Push-forward $\phi\#\mu$

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

Not compatible



$$1 \not\rightarrow \left(\frac{1}{2}, \frac{1}{2}\right)$$

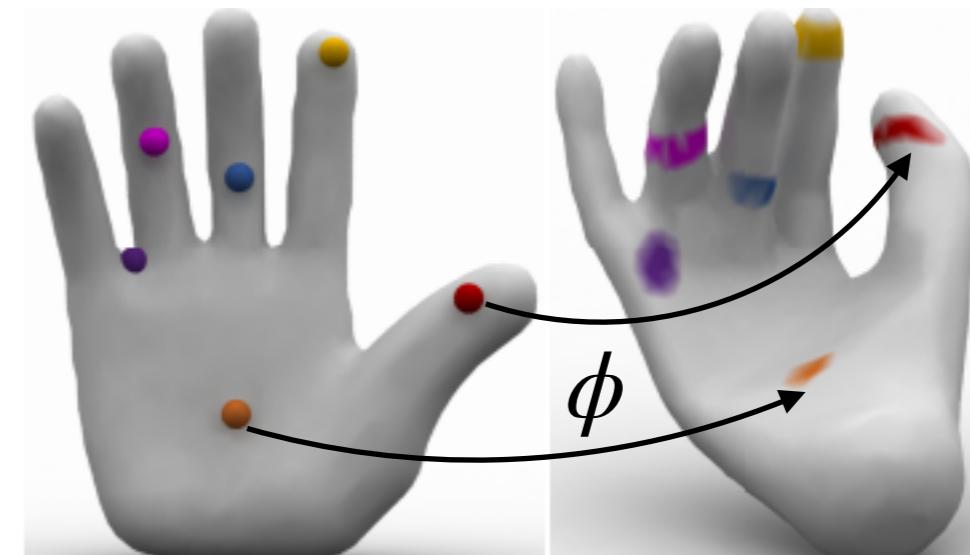
...to Gromov-Wasserstein

Gromov-Wasserstein distance



Gromov-Wasserstein = a bending invariant distance

- $GW_p(d_{\mathcal{X}}, d_{\mathcal{Y}}, \mu, \nu) = 0$ iff $\exists \phi : \mathcal{X} \rightarrow \mathcal{Y}$
 - ϕ is a isometry $d_{\mathcal{X}}(x, x') = d_{\mathcal{Y}}(\phi(x), \phi(x'))$
 - ϕ is measure-preserving $\phi \#\mu = \nu$

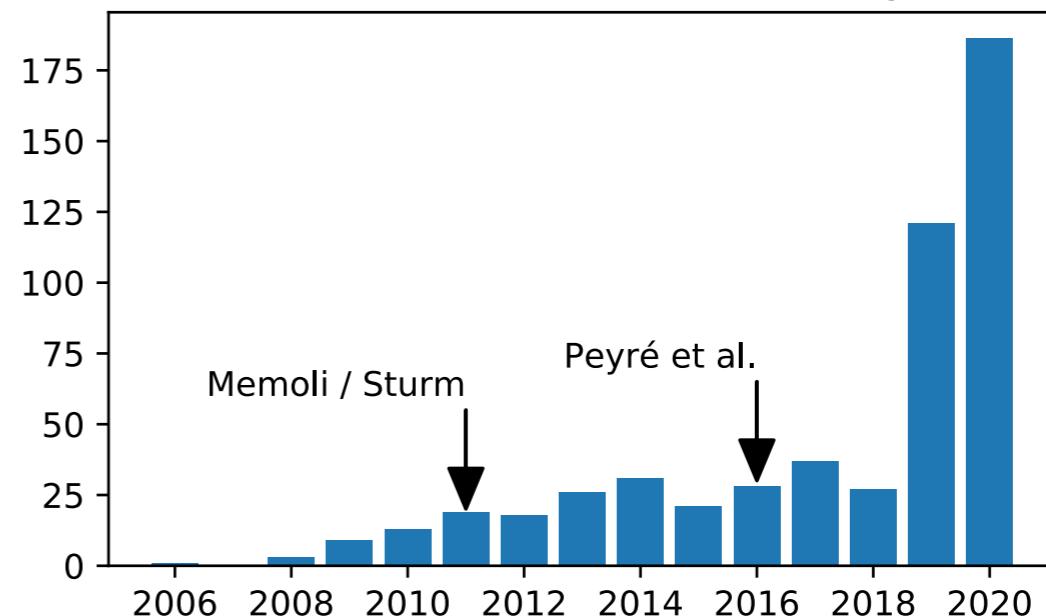


[Solomon 2016]

Applications for geometric data

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

Occurrences Gromov-Wasserstein in Google Scholar





Solving OT

Solving OT

A linear problem

Discrete probability measures

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

Linear Program:

$$\min_{\pi \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{ij} c_{i,j} \pi_{i,j} = \min_{\pi \in \Pi(\mathbf{a}, \mathbf{b})} \langle \mathbf{C}, \boldsymbol{\pi} \rangle$$

| Simplex, Network flow, Hungarian algorithms $\sim O(n^3 \log(n))$

Solving OT

A linear problem

Discrete probability measures

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

Linear Program:

$$\min_{\pi \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{ij} c_{i,j} \pi_{i,j} = \min_{\pi \in \Pi(\mathbf{a}, \mathbf{b})} \langle \mathbf{C}, \boldsymbol{\pi} \rangle$$

Simplex, Network flow, Hungarian algorithms $\sim O(n^3 \log(n))$

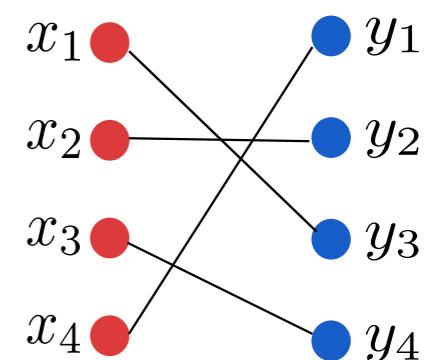
Uniform weights

$$\mathbf{a} = \mathbf{b} = \frac{\mathbf{1}_n}{n}$$

Monge Problem

$$\min_{\sigma \in S_n} \sum_{i=1}^n c_{i,\sigma(i)}$$

One-to-one



Solving OT

A linear problem

Discrete probability measures

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

Linear Program:

$$\min_{\pi \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{ij} c_{i,j} \pi_{i,j} = \min_{\pi \in \Pi(\mathbf{a}, \mathbf{b})} \langle \mathbf{C}, \boldsymbol{\pi} \rangle$$

Simplex, Network flow, Hungarian algorithms $\sim O(n^3 \log(n))$

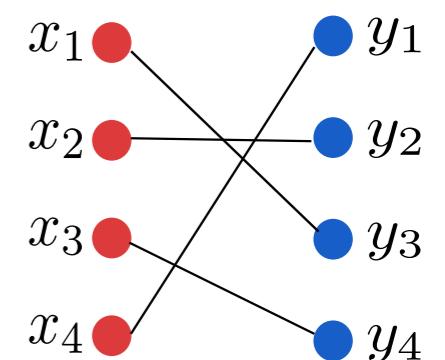
Uniform weights

$$\mathbf{a} = \mathbf{b} = \frac{\mathbf{1}_n}{n}$$

Monge Problem

$$\min_{\sigma \in S_n} \sum_{i=1}^n c_{i,\sigma(i)}$$

One-to-one

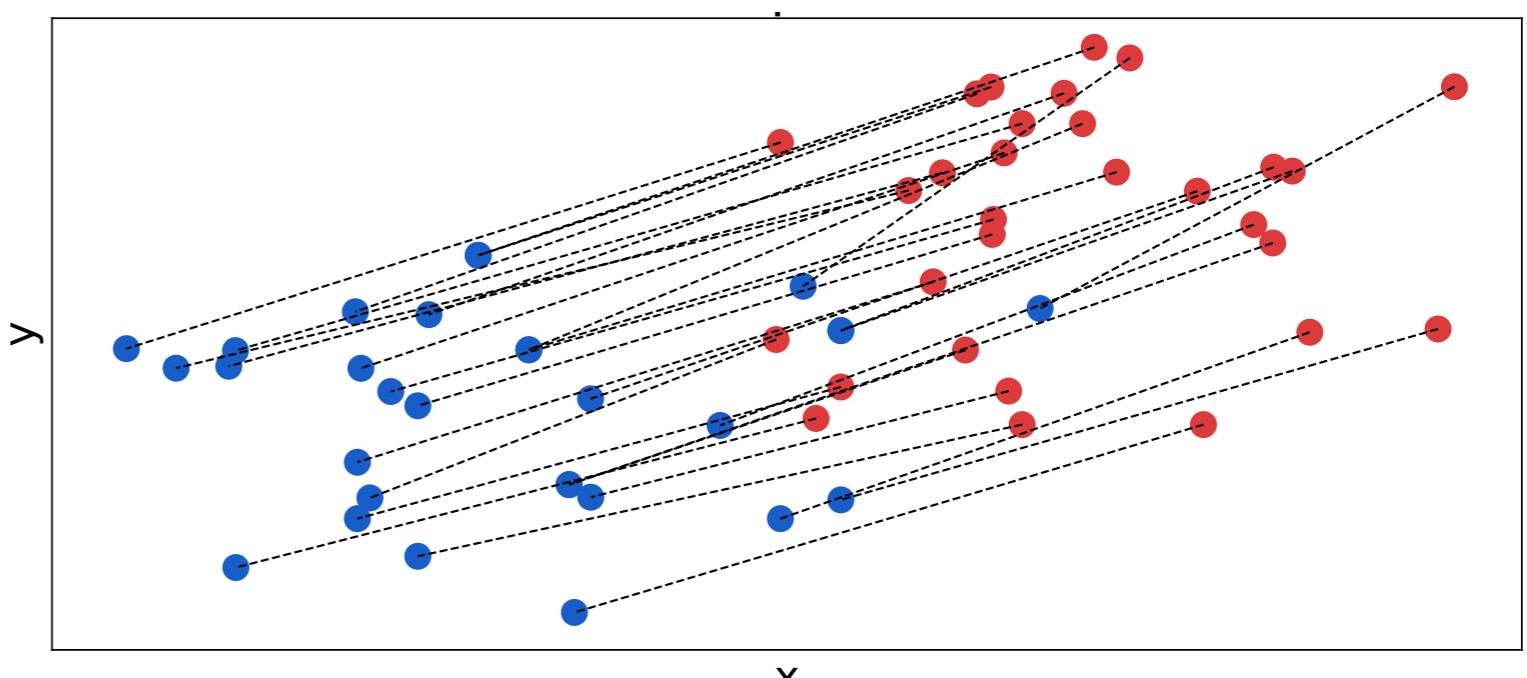


Fundamental theorem LP:

$$\pi^* \leftrightarrow \sigma^* \in S_n$$

Optimal coupling is a permutation

Solves the Monge Problem

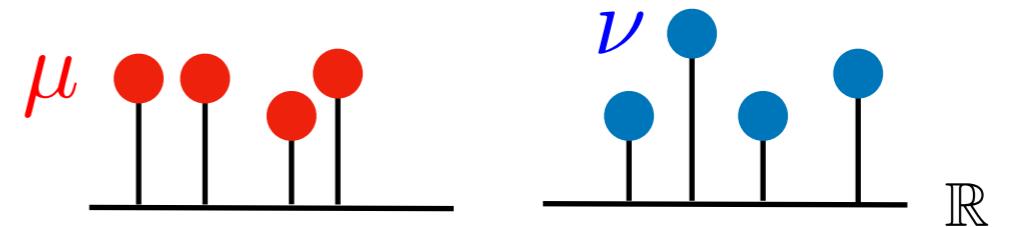


Solving OT

A real line problem

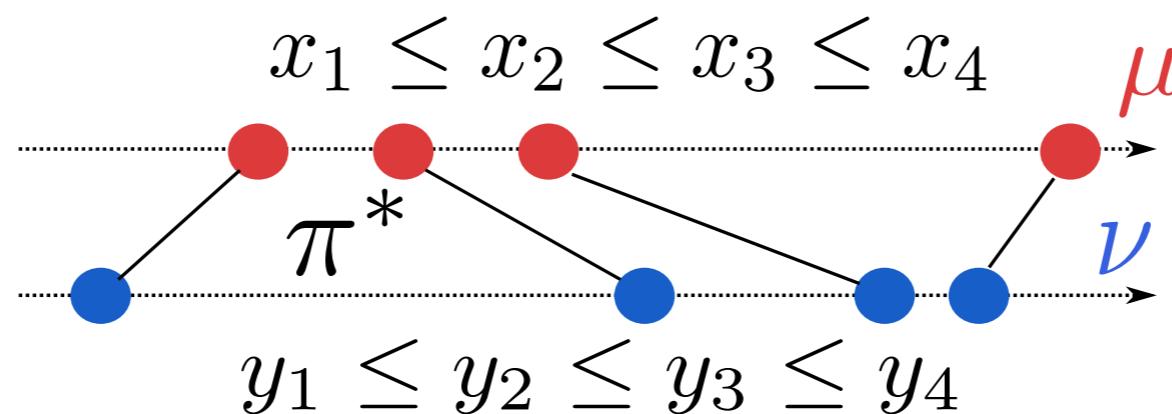
Two discrete probability distributions

$$\mu = \frac{1}{n} \sum_{i=1}^n \delta_{x_i}, \nu = \frac{1}{n} \sum_{j=1}^n \delta_{y_j}$$
$$x_i, y_j \in \mathbb{R}$$



In the case of Wasserstein can be solved by simple sorts

$\sim O(n \log(n))$



$$\min_{\pi \in \Pi(\frac{\mathbf{1}_n}{n}, \frac{\mathbf{1}_n}{n})} \sum_{ij} (x_i - y_j)^2 \pi_{i,j} = \min_{\sigma \in S_n} \sum_{ij} (x_i - y_{\sigma(i)})^2 \rightarrow Id$$

Solving OT

Entropic regularization

Discrete probability measures

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

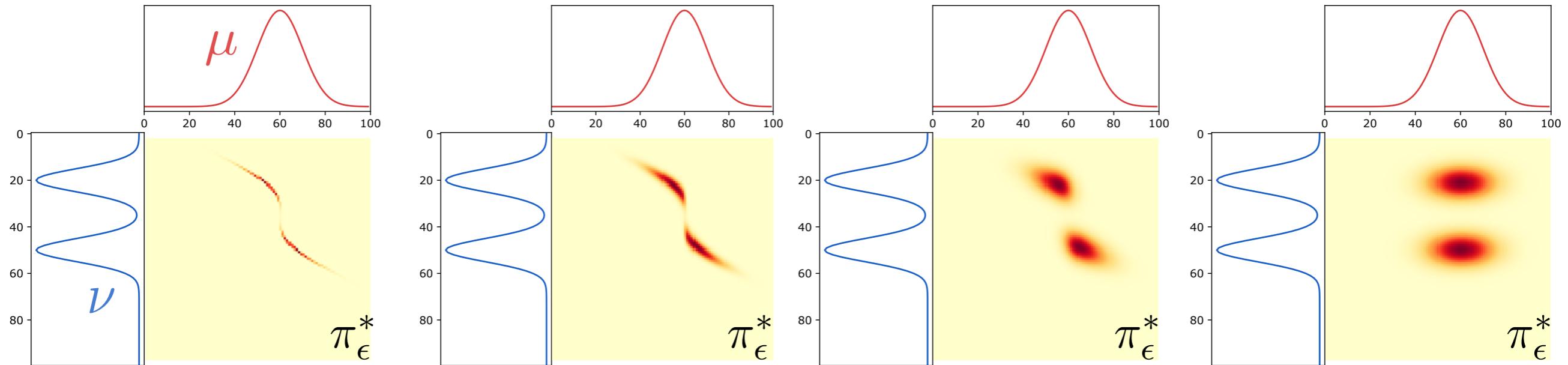
Strongly convex problem:

$$\min_{\pi \in \Pi(\mathbf{a}, \mathbf{b})} \langle \mathbf{C}, \boldsymbol{\pi} \rangle - \varepsilon H(\boldsymbol{\pi})$$

| Entropy term $H(\boldsymbol{\pi}) = - \sum_{ij} (\log(\pi_{ij}) - 1) \pi_{ij}$

| Sinkhorn-Knopp algorithm: 1) fast 2) based on matrix multiplication

| τ approximate solution $\sim O(n^2 \log(n) \tau^{-3})$



$0 \leftarrow \epsilon$

$\epsilon \rightarrow +\infty$

Solving OT

Entropic regularization

Discrete probability measures

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

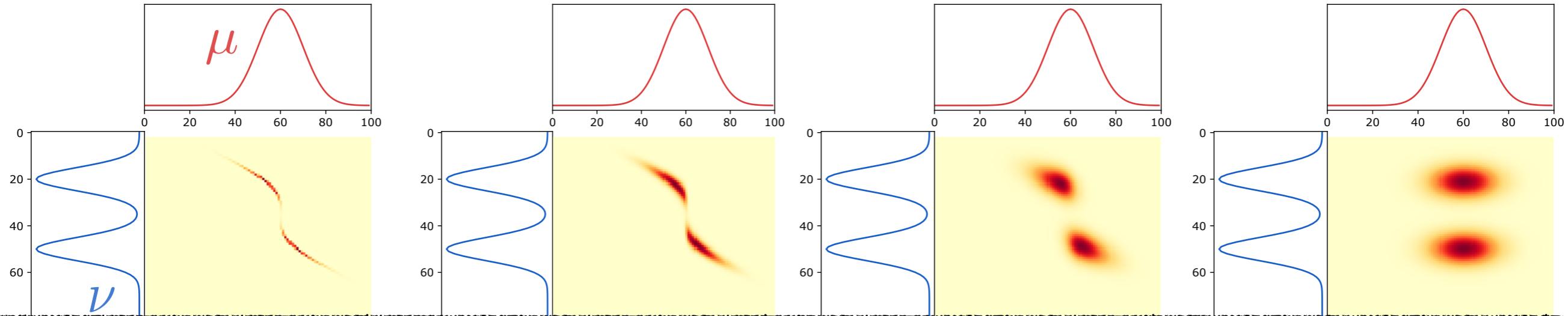
Strongly convex problem:

$$\min_{\pi \in \Pi(\mathbf{a}, \mathbf{b})} \langle \mathbf{C}, \boldsymbol{\pi} \rangle - \varepsilon H(\boldsymbol{\pi})$$

| Entropy term $H(\boldsymbol{\pi}) = - \sum_{ij} (\log(\pi_{ij}) - 1) \pi_{ij}$

| Sinkhorn-Knopp algorithm: 1) fast 2) based on matrix multiplication

| τ approximate solution $\sim O(n^2 \log(n) \tau^{-3})$



Linear OT: costly but solvable in practice

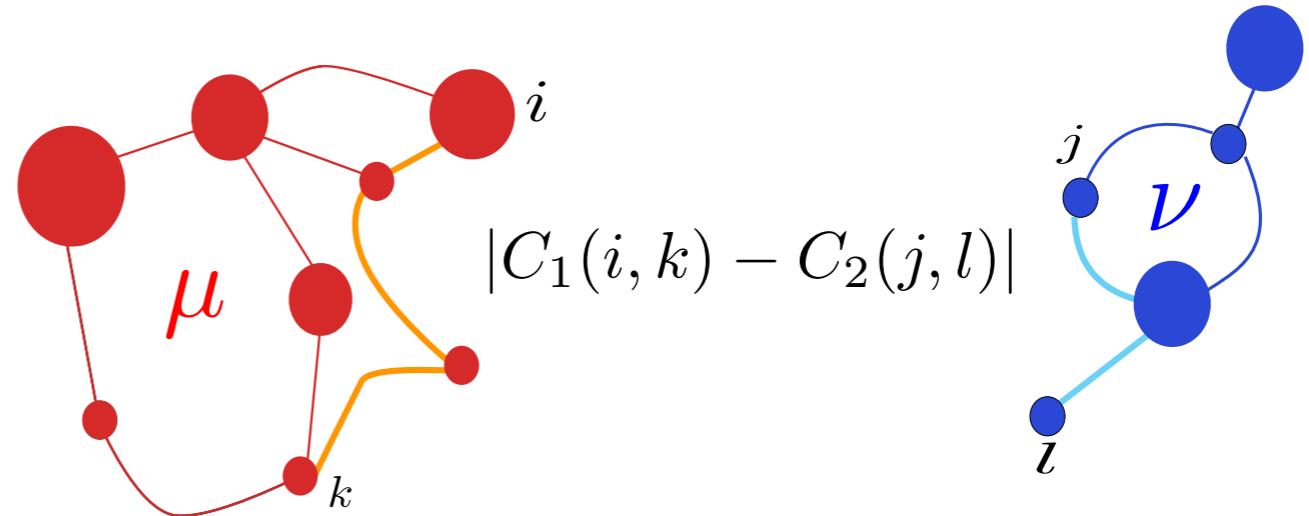
Solving OT

A quadratic problem (QP)

Discrete probability measures

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

$$\mathcal{X}, \mathcal{Y} \not\subset \Omega$$



$$\min_{\pi \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{ijkl} |C_1(i, k) - C_2(j, l)|^p \pi_{ij} \pi_{kl}$$

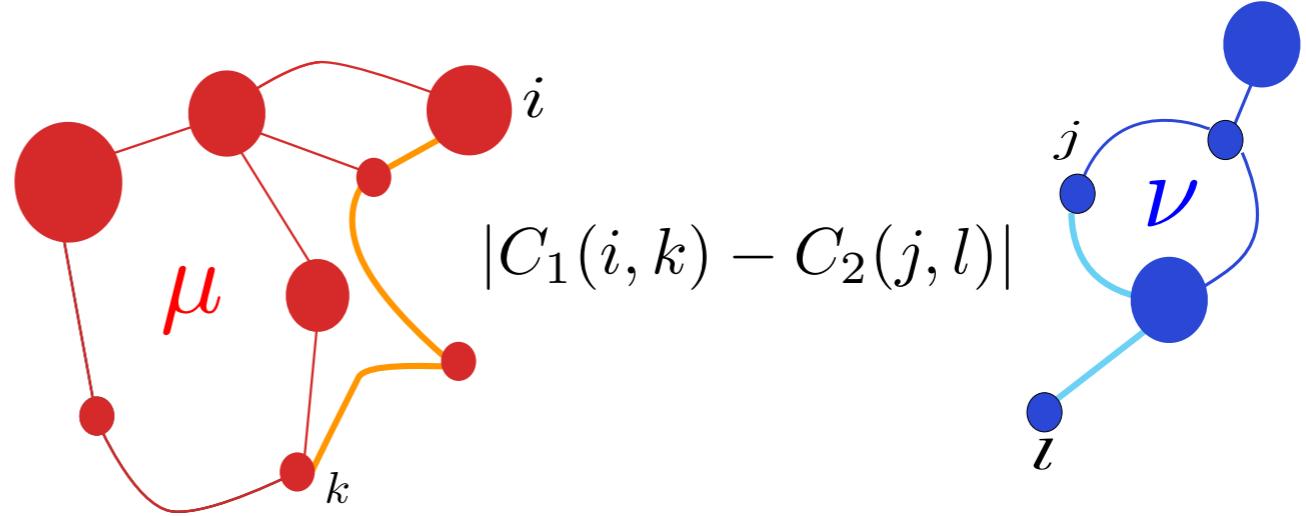
Solving OT

A quadratic problem (QP)

Discrete probability measures

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

$$\mathcal{X}, \mathcal{Y} \not\subset \Omega$$



$$\min_{\pi \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{ijkl} |C_1(i, k) - C_2(j, l)|^p \pi_{ij} \pi_{kl}$$

Non convex QP: NP-hard in general

(graph matching problem)

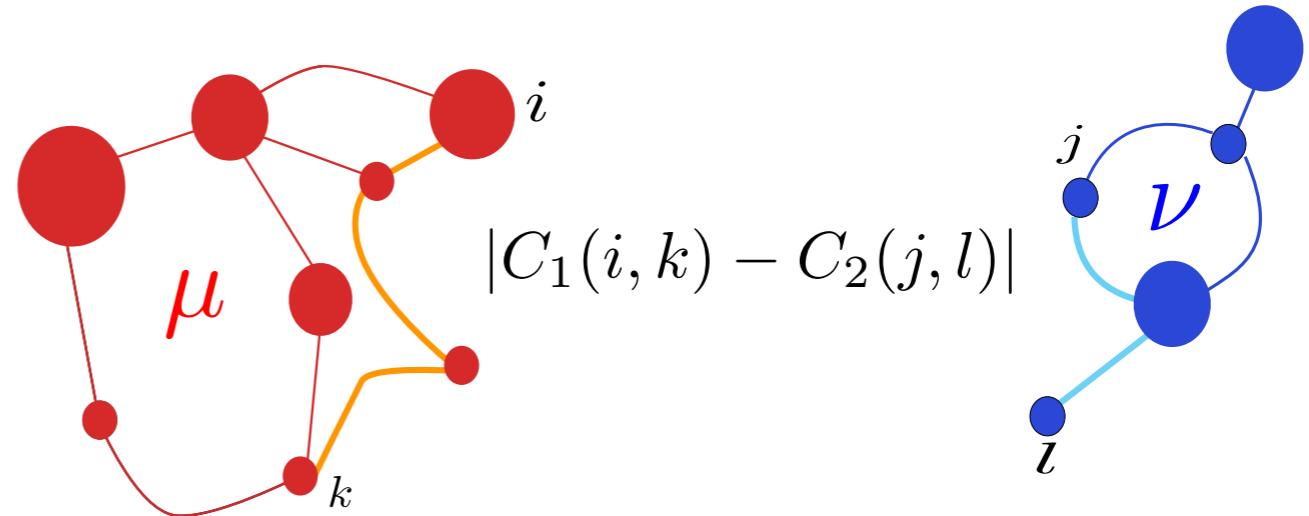
Solving OT

A quadratic problem (QP)

Discrete probability measures

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

$$x, y \notin \Omega$$



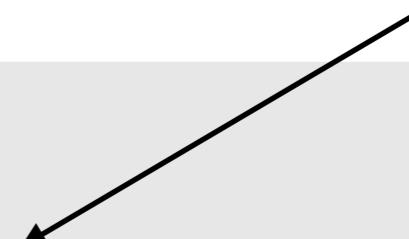
$$\min_{\pi \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{ijkl} |C_1(i, k) - C_2(j, l)|^p \pi_{ij} \pi_{kl} - \varepsilon H(\pi)$$

Non convex QP: NP-hard in general

With entropic regularization [Peyré 2016, Solomon 2016]

Can be solved using projected gradient descent under KL geometry

Each gradient step: Sinkhorn algorithm



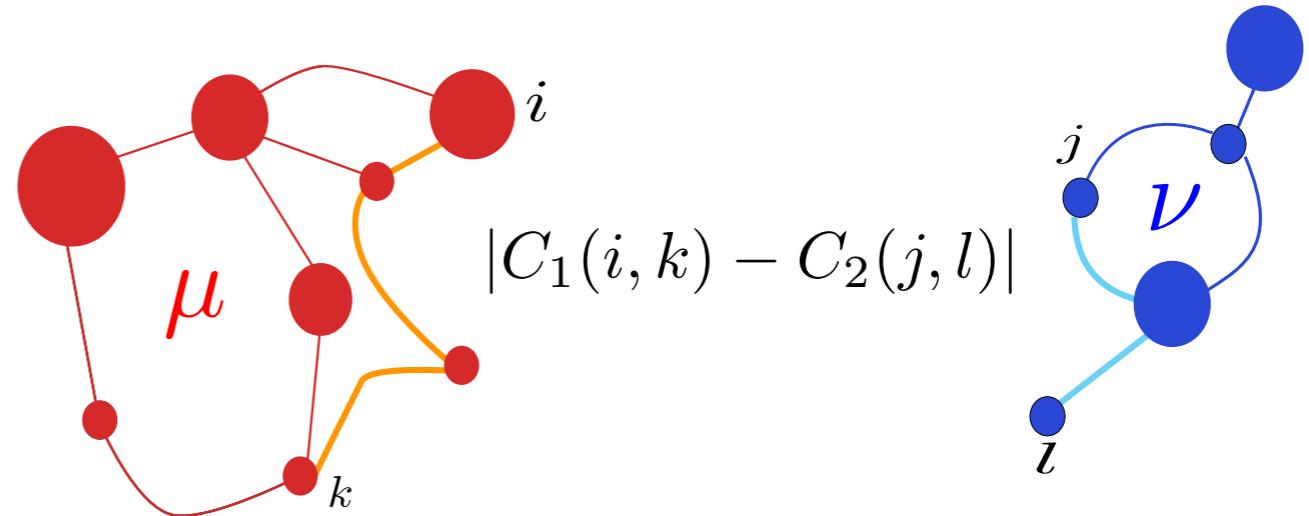
Solving OT

A quadratic problem (QP)

Discrete probability measures

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

$$x, y \notin \Omega$$



$$\min_{\pi \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{ijkl} |C_1(i, k) - C_2(j, l)|^p \pi_{ij} \pi_{kl} - \varepsilon H(\boldsymbol{\pi})$$

Non convex QP: NP-hard in general

With entropic regularization [Peyré 2016, Solomon 2016] $\sim O(n_{iter} * n^2 \log(n))$

Can be solved using projected gradient descent under KL geometry

Each gradient step: Sinkhorn algorithm

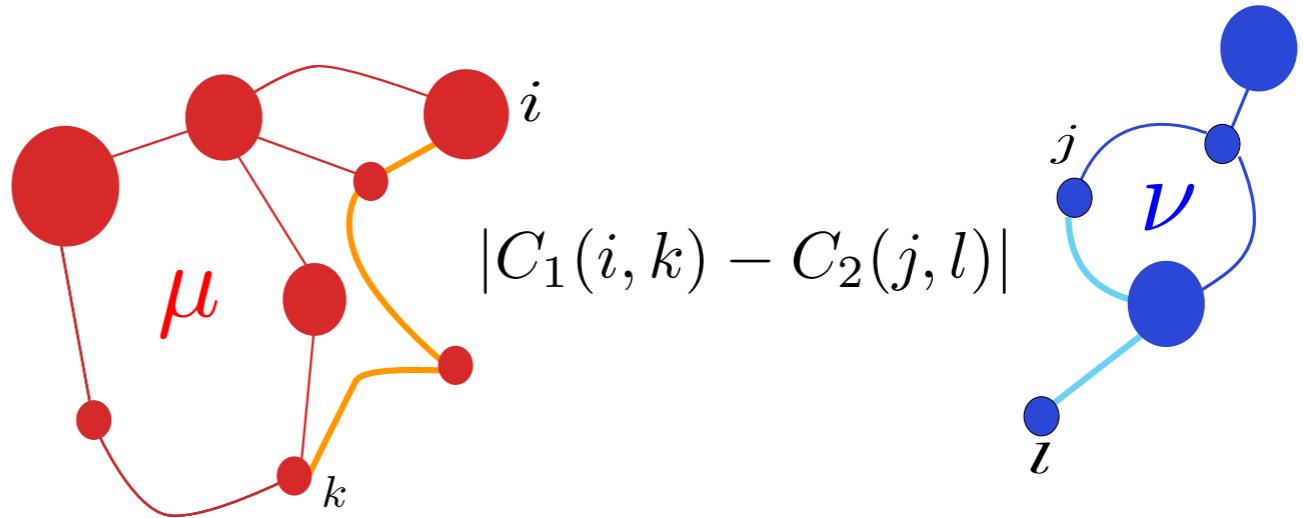
Solving OT

A quadratic problem (QP)

Discrete probability measures

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

$$\mathcal{X}, \mathcal{Y} \not\subset \Omega$$



$$\min_{\pi \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{ijkl} |C_1(i, k) - C_2(j, l)|^p \pi_{ij} \pi_{kl}$$

Non convex QP: NP-hard in general

With entropic regularization [Peyré 2016, Solomon 2016] $\sim O(n_{iter} * n^2 \log(n))$

Can be solved using projected gradient descent under KL geometry

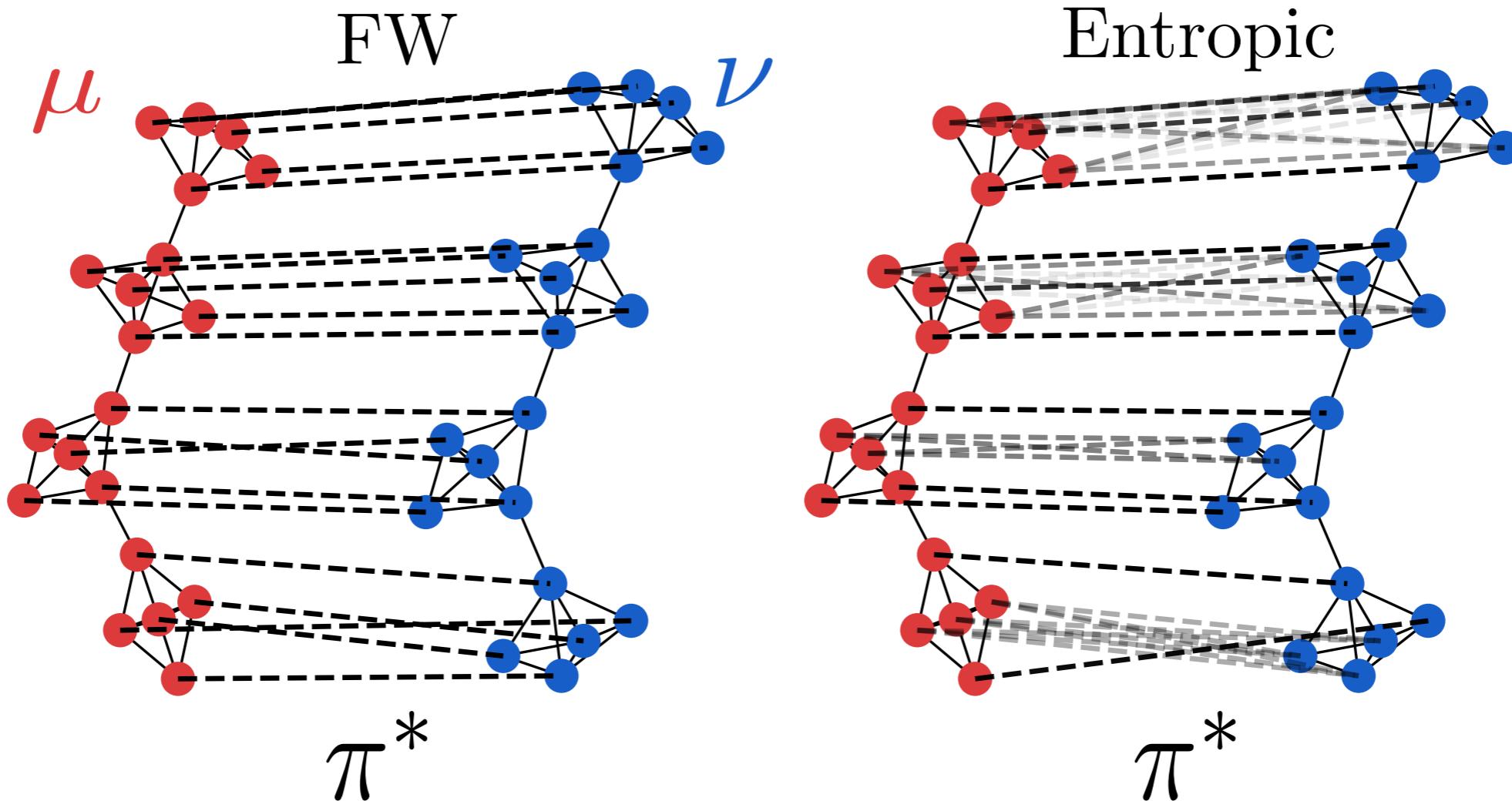
Each gradient step: Sinkhorn algorithm

Hard to solve and even to approximate...

...to Gromov-Wasserstein

An example on graphs

C_1, C_2 are the shortest path distance in each graph



From linear Optimal Transport...

What is it?

Input:

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

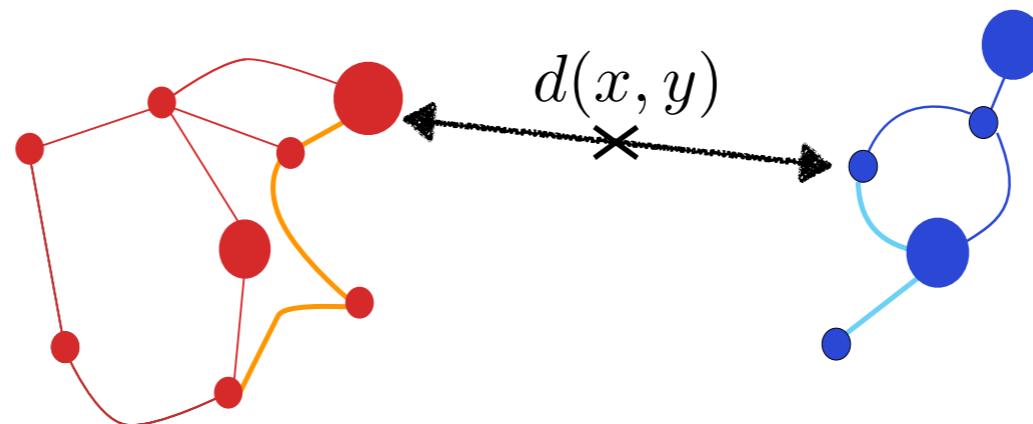
Two probability distributions

Output:

Geometric notion of distance between these distributions

Find correspondences/relations between the samples

Optimal transport for structured data



Optimal Transport for structured data

Motivations

Motivation: Is the Optimal transport framework suited for structured data ?

Problem 1: How do we model structured data ?

As probability distributions!

Problem 2: How do we compare structured data ?

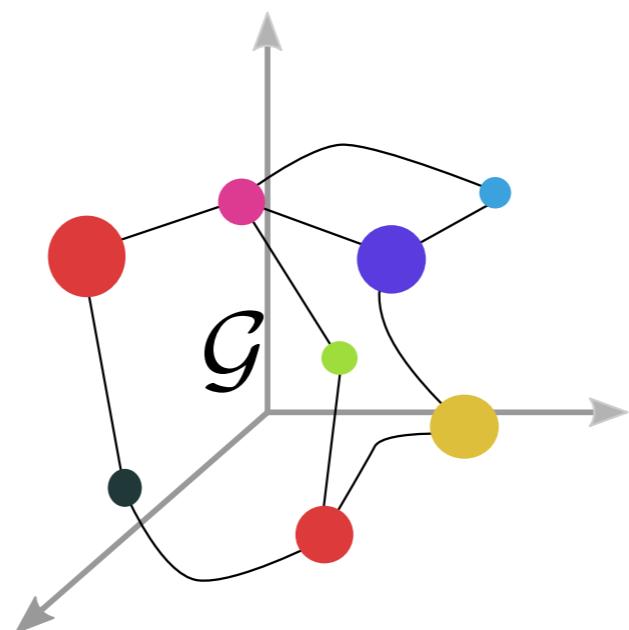
Based on the theories of Wasserstein and Gromov-Wasserstein

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

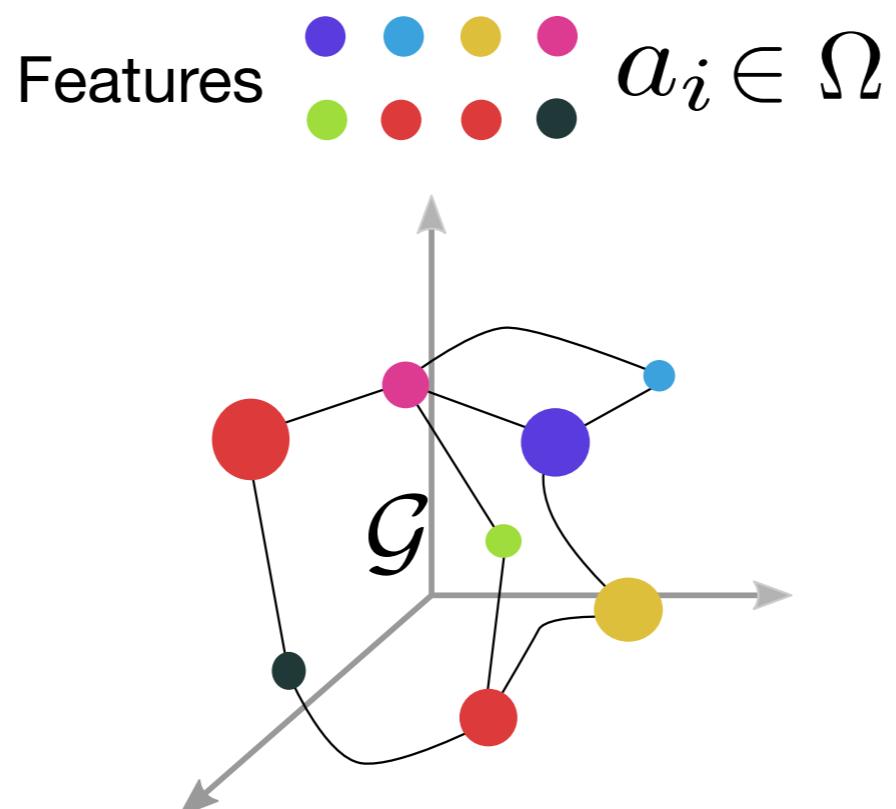


Optimal Transport for structured data

Structured data as probability distribution

Discrete case

- | Structured data can be seen as a labeled graph
- | Combines a feature **and** a structure information

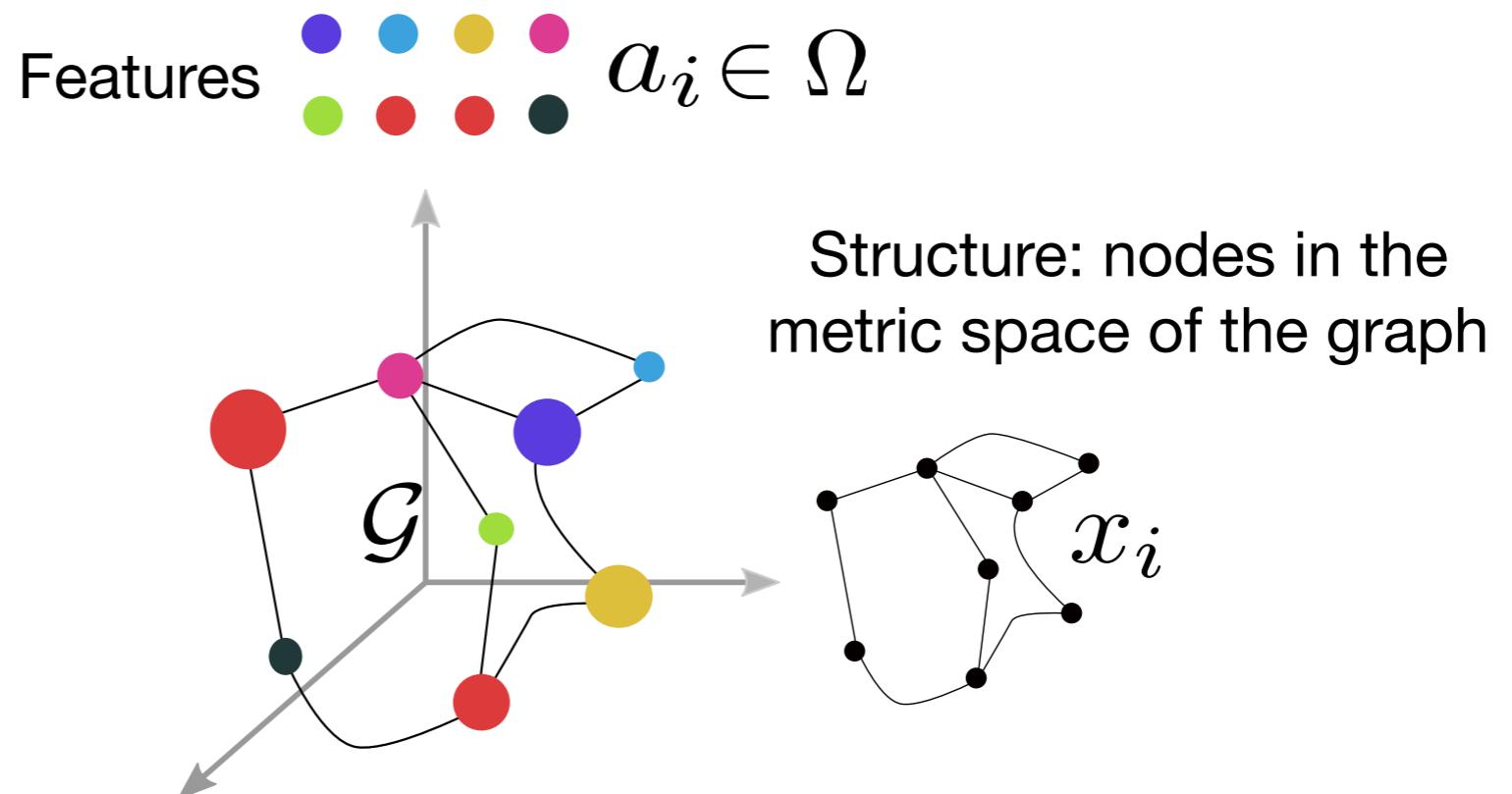


Optimal Transport for structured data

Structured data as probability distribution

Discrete case

- | Structured data can be seen as a labeled graph
- | Combines a feature **and** a structure information

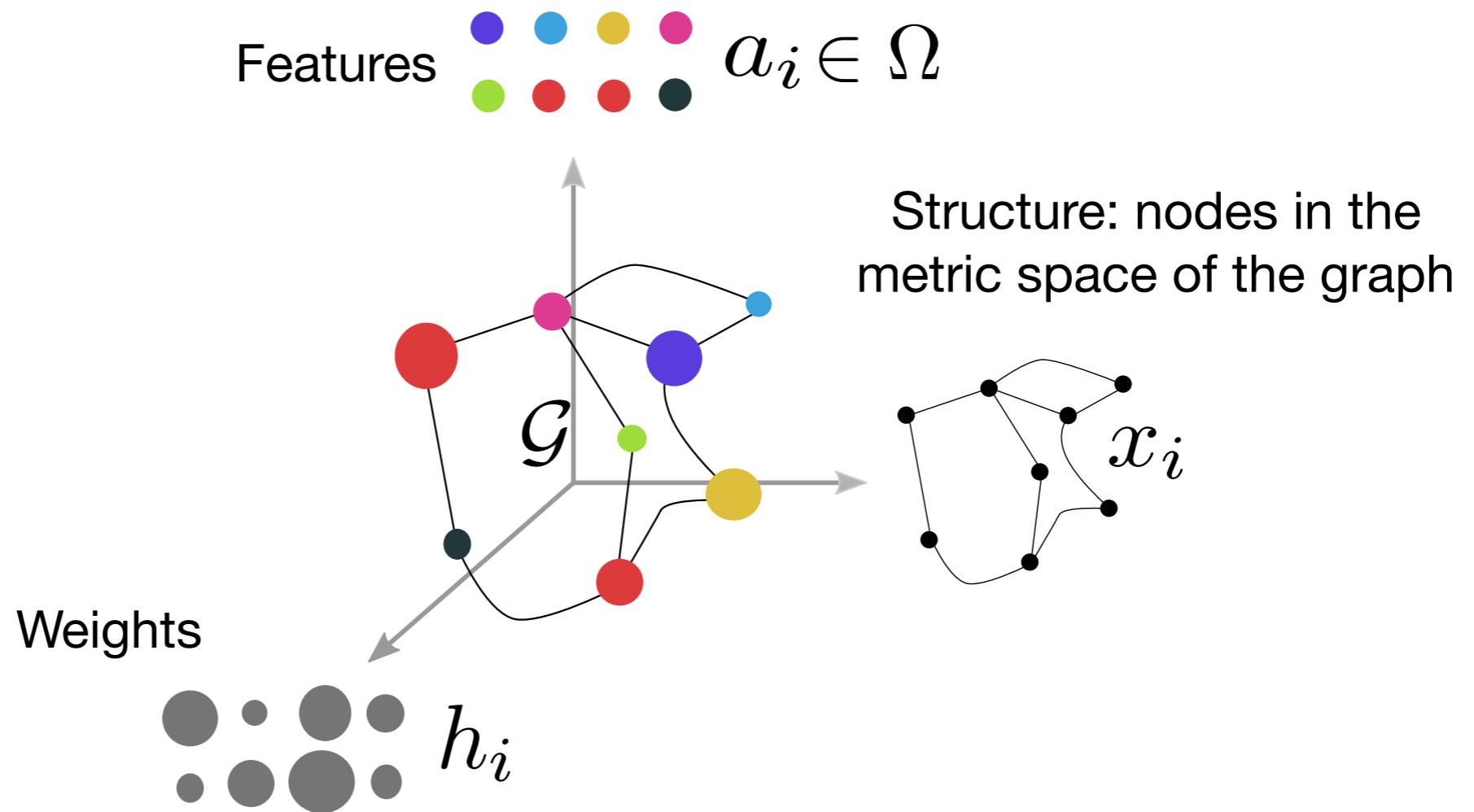


Optimal Transport for structured data

Structured data as probability distribution

Discrete case

- | Structured data can be seen as a labeled graph
- | Combines a feature **and** a structure information
- | Add weights that encodes the relative importance of the nodes



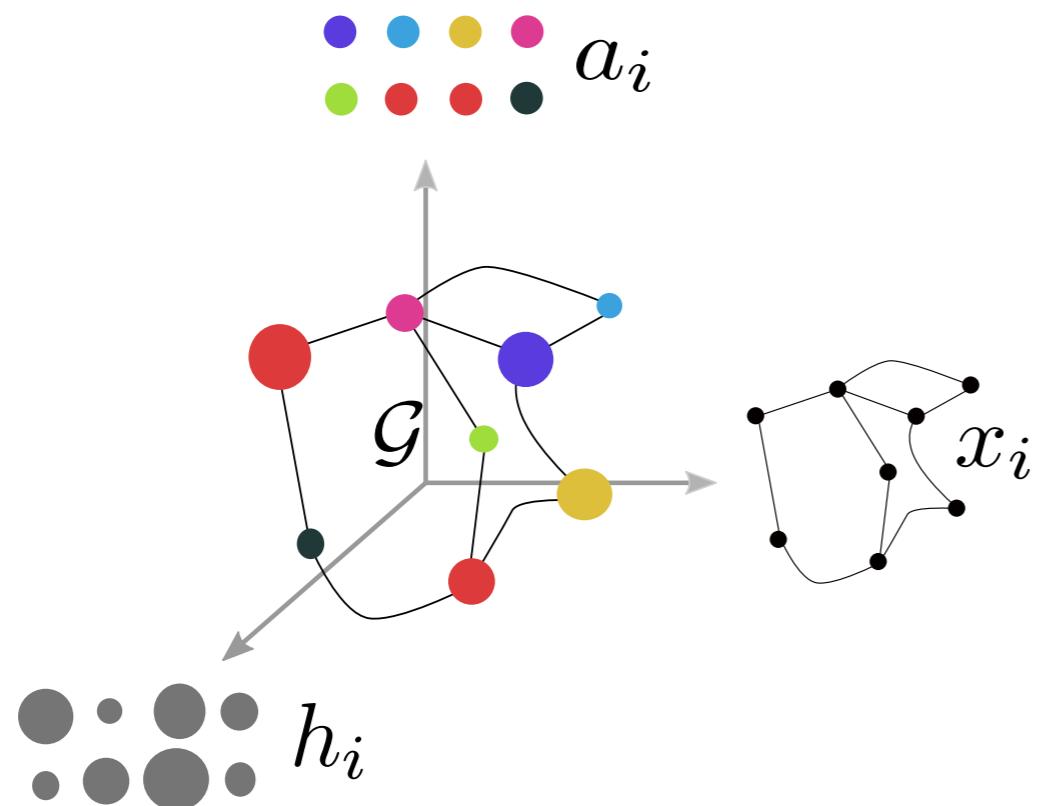
Optimal Transport for structured data

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



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

$$\left. \begin{array}{c} \text{colorful dots} \\ \text{graph} \\ \text{gray circles} \end{array} \right\} \mu_A = \sum_i h_i \delta_{a_i}$$

$$\left. \begin{array}{c} \text{colorful dots} \\ \text{graph} \\ \text{gray circles} \end{array} \right\} \mu_X = \sum_i h_i \delta_{x_i}$$

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

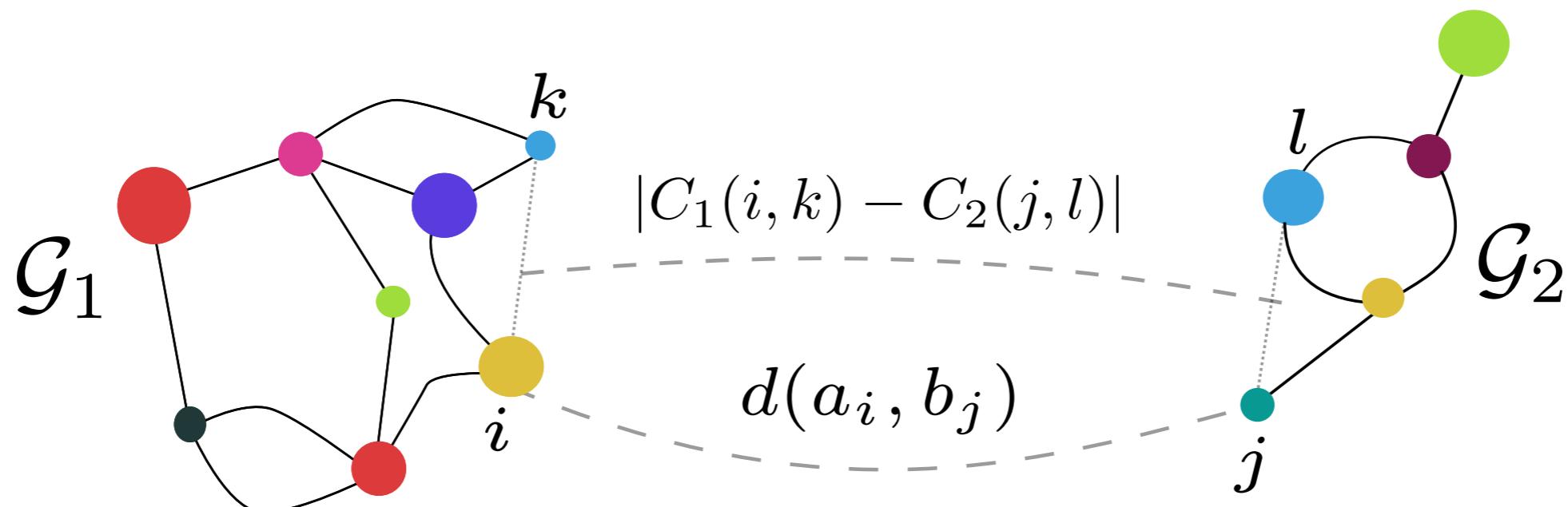
$$\mathbf{C}_1, \mathbf{C}_2$$

A distance between labels

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

Fused Gromov-Wasserstein distance

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



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

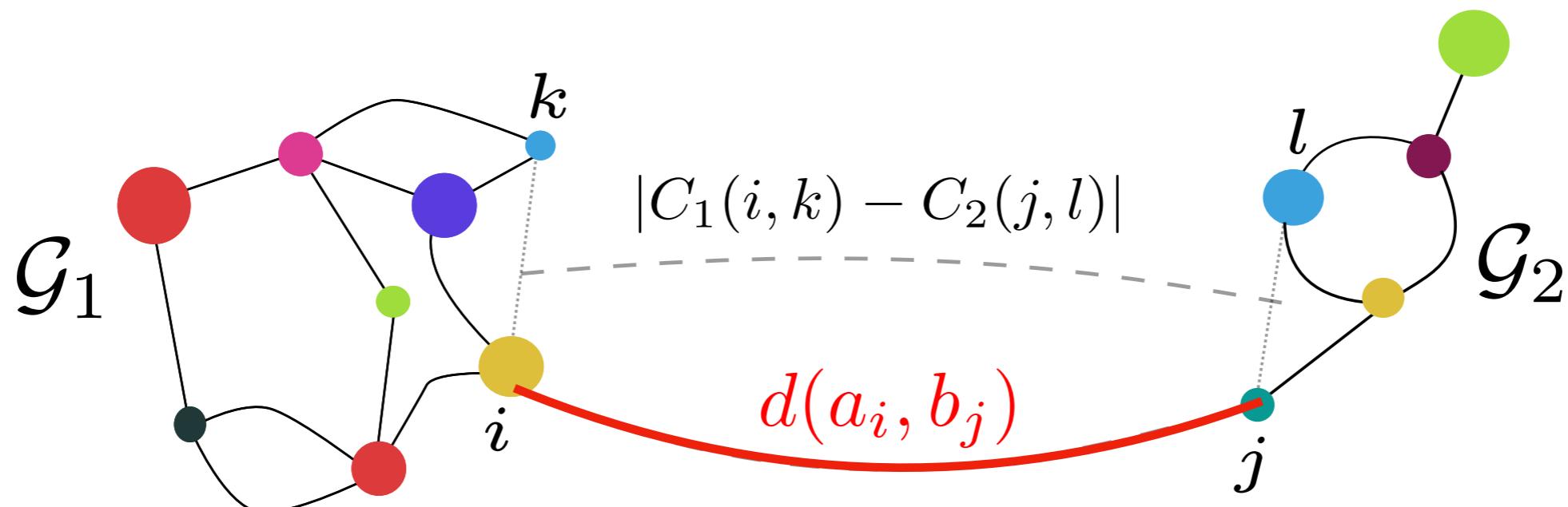
$$\mathbf{C}_1, \mathbf{C}_2$$

A distance between labels

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

Fused Gromov-Wasserstein distance

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



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

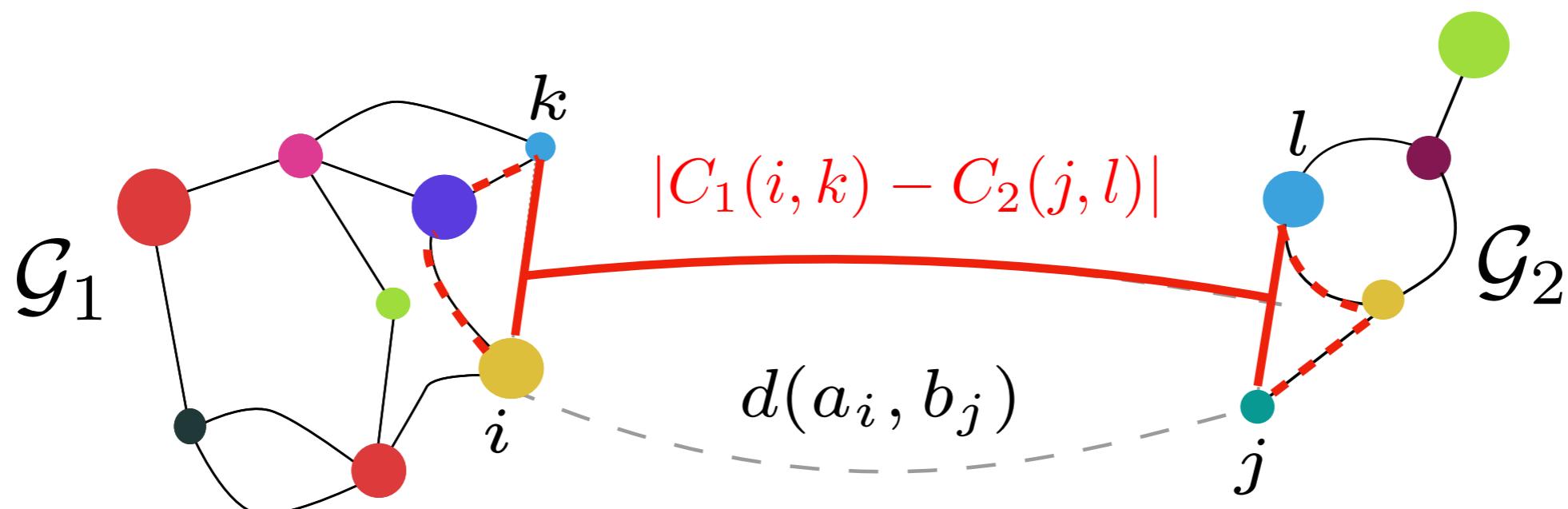
$$\mathbf{C}_1, \mathbf{C}_2$$

A distance between labels

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

Fused Gromov-Wasserstein distance

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



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

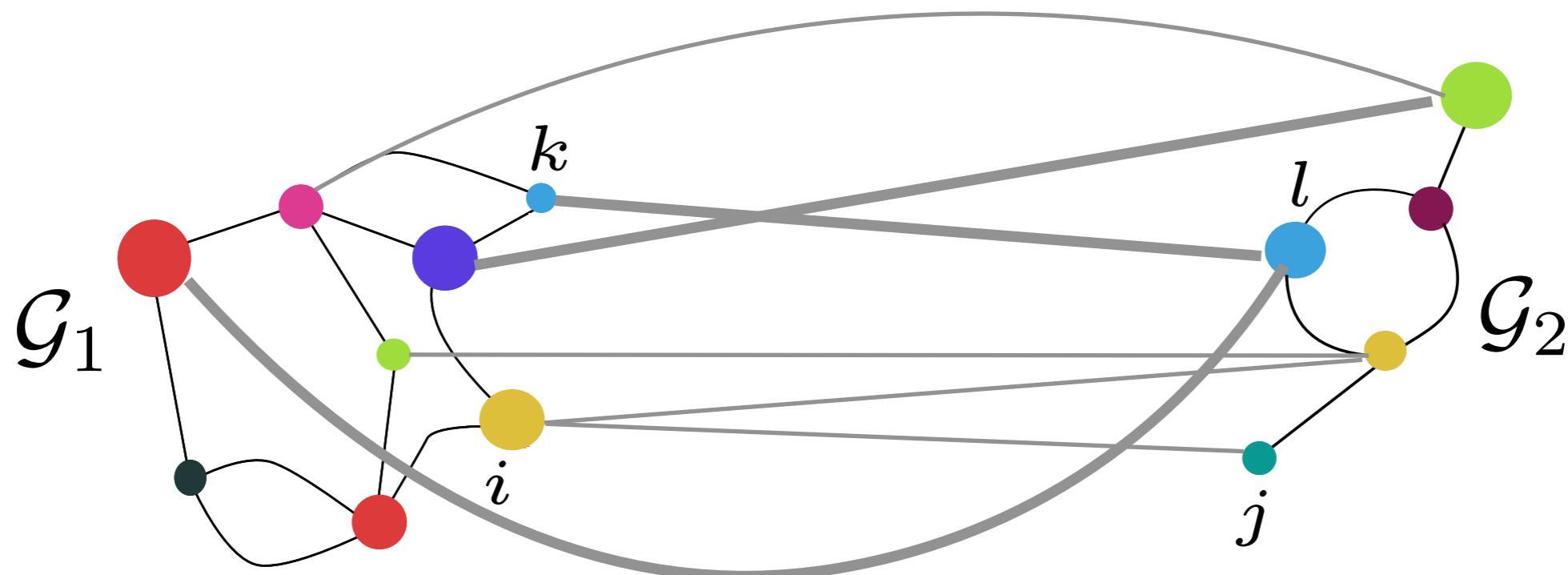
$$\mathbf{C}_1, \mathbf{C}_2$$

A distance between labels

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

Fused Gromov-Wasserstein distance

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

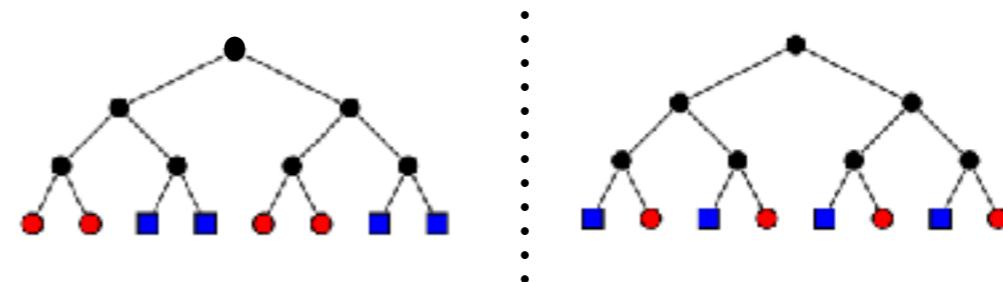


π provides a soft assignment of the nodes

Optimal Transport for structured data

Fused Gromov-Wasserstein distance: example

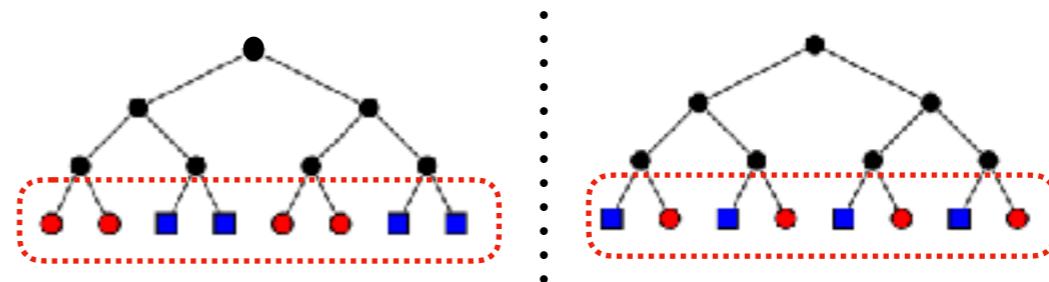
Consider two trees



Optimal Transport for structured data

Fused Gromov-Wasserstein distance: example

Consider two trees

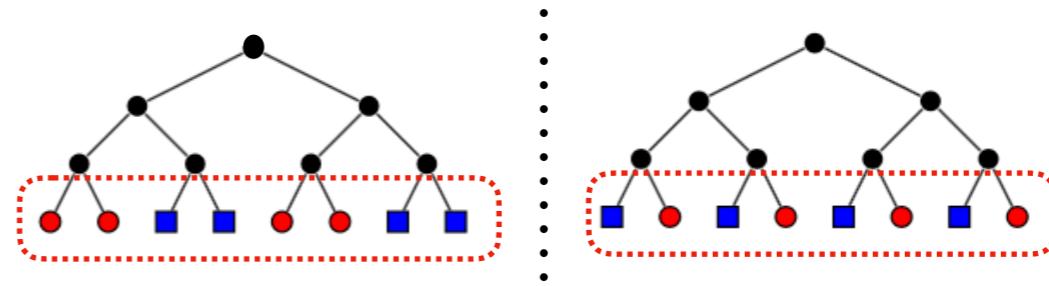


We want to compare the leaves of the trees

Optimal Transport for structured data

Fused Gromov-Wasserstein distance: example

Consider two trees

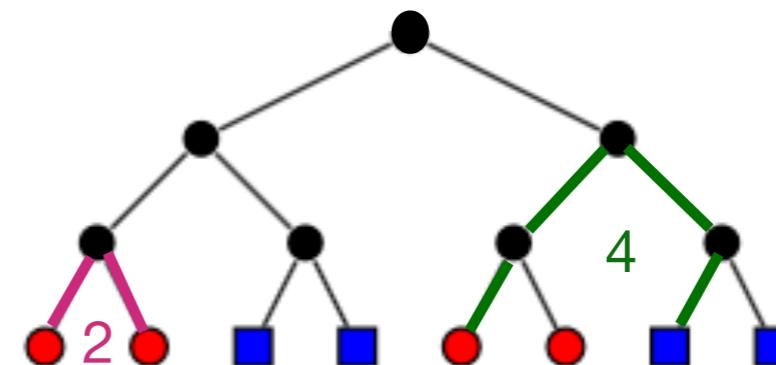


We want to compare the leaves of the trees

Features: blue or red

• • ■ ■ • • ■ ■ : ■ • ■ ■ • ■ • ■ •

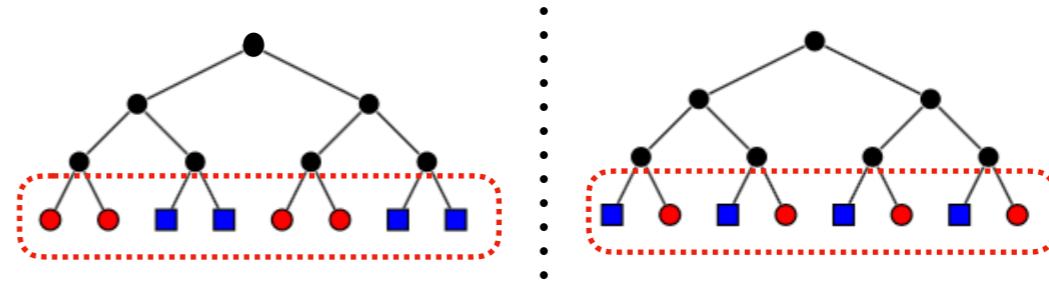
Structures : shortest path between the leaves



Optimal Transport for structured data

Fused Gromov-Wasserstein distance: example

Consider two trees

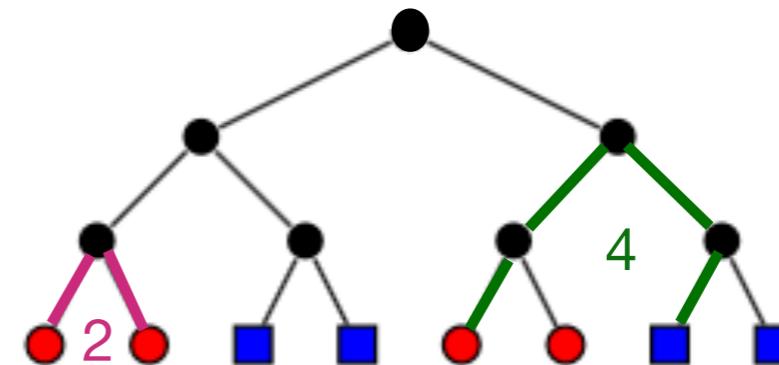


We want to compare the leaves of the trees

Features: blue or red

• • ■ ■ • • ■ ■ : ■ • ■ ■ • ■ • ■ •

Structures : shortest path between the leaves

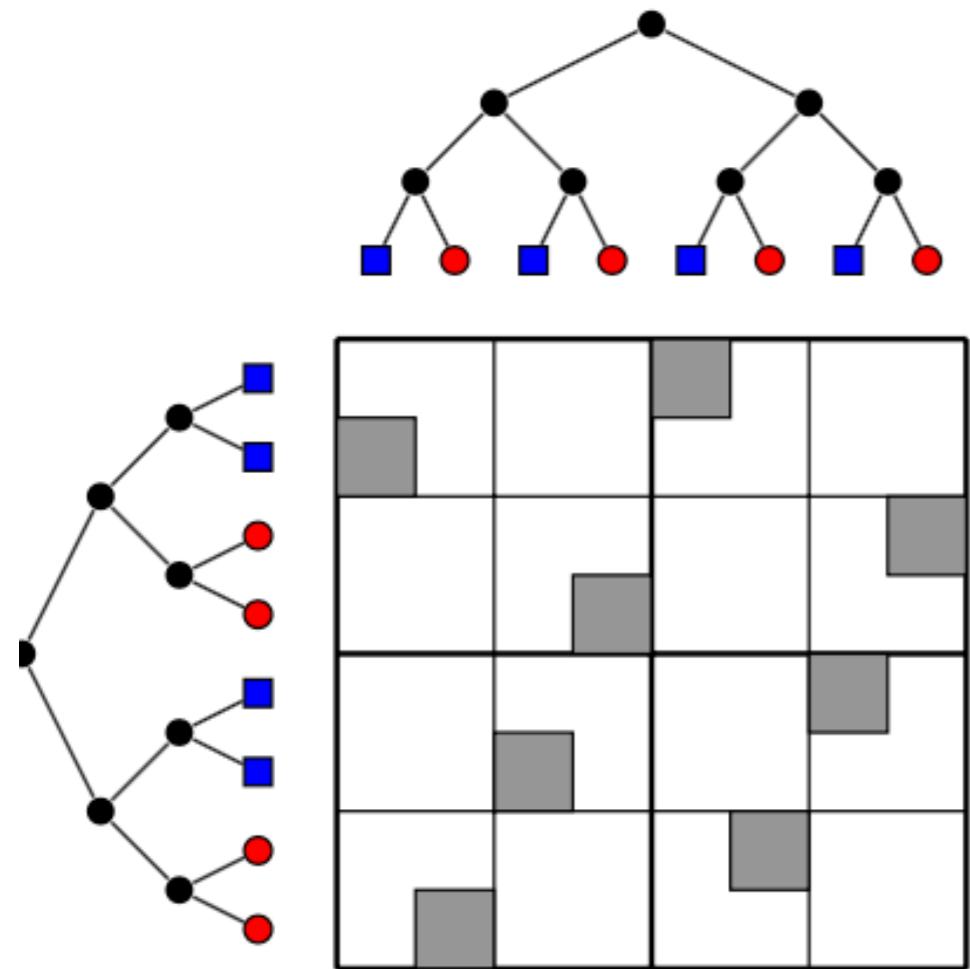


Taking both the structures and the features into account
with FGW

Optimal Transport for structured data

Fused Gromov-Wasserstein distance: example

Wasserstein distance
(features only)

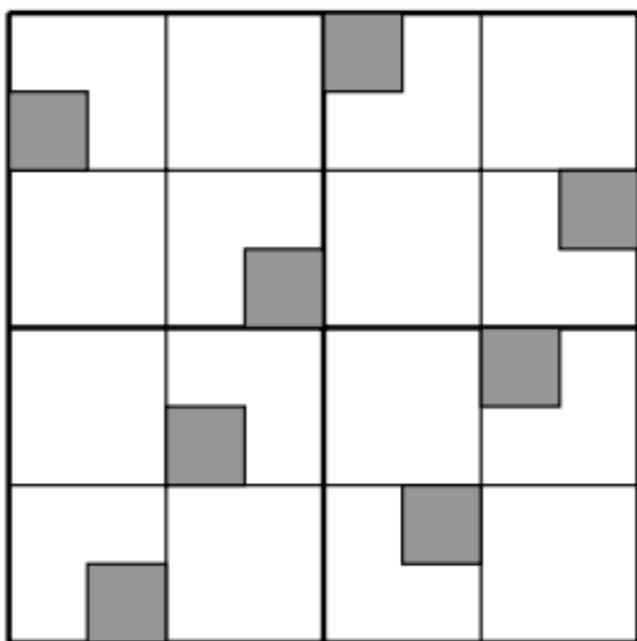
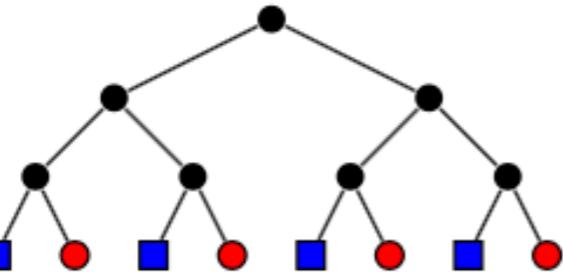


$$W = 0$$

Optimal Transport for structured data

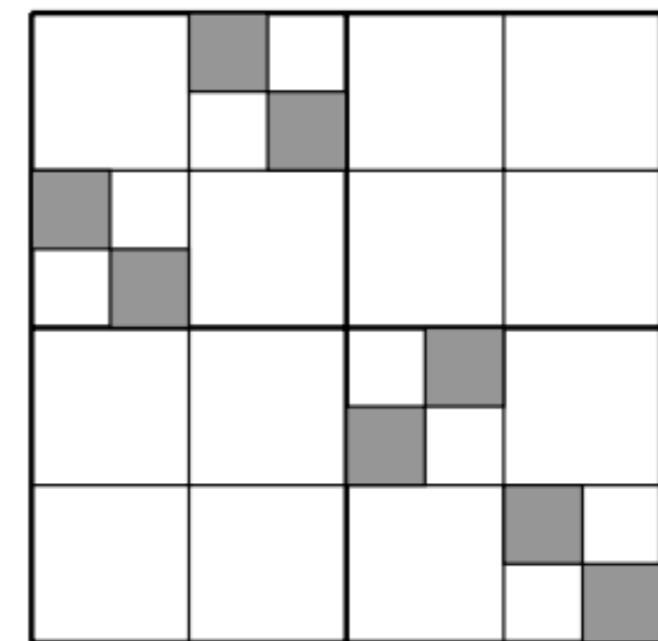
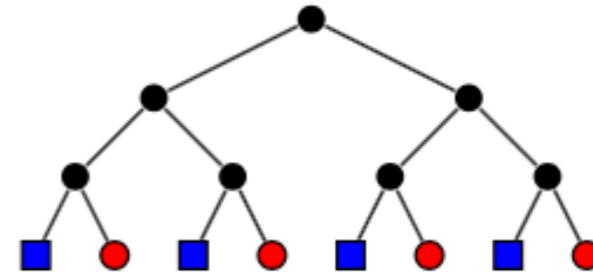
Fused Gromov-Wasserstein distance: example

Wasserstein distance
(features only)



$$W = 0$$

Gromov-Wasserstein distance
(structures only)

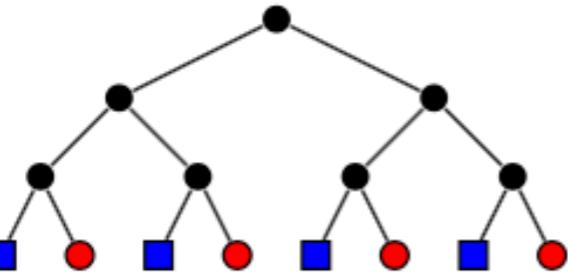


$$GW = 0$$

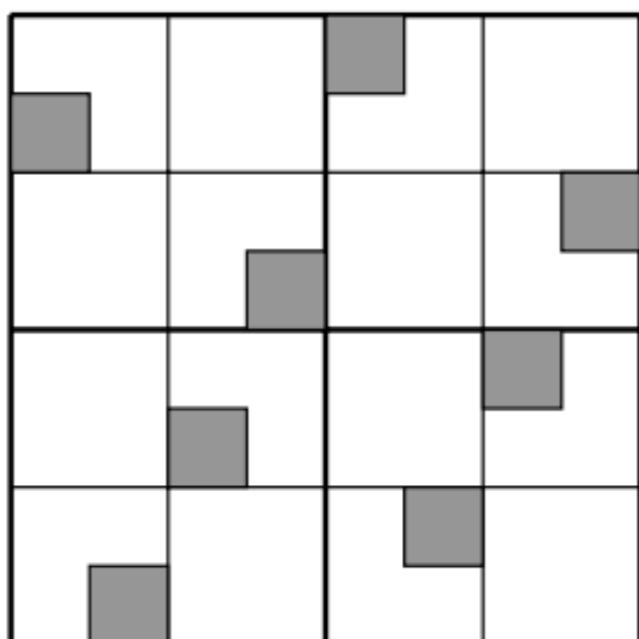
Optimal Transport for structured data

Fused Gromov-Wasserstein distance: example

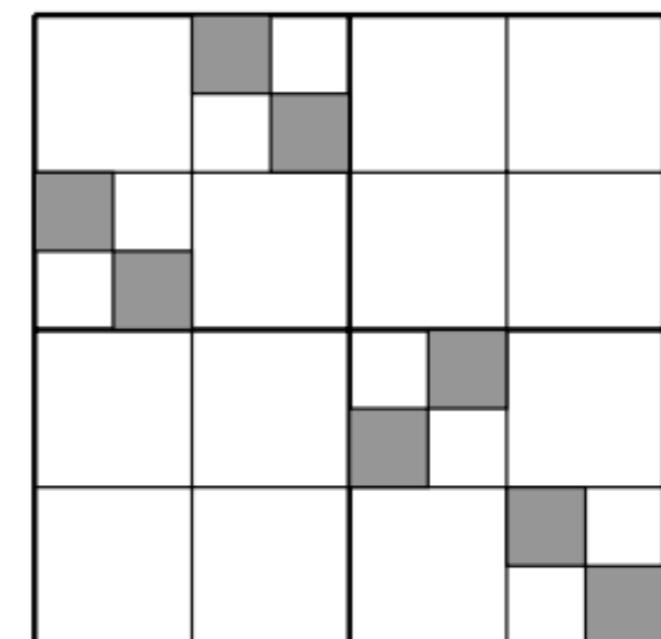
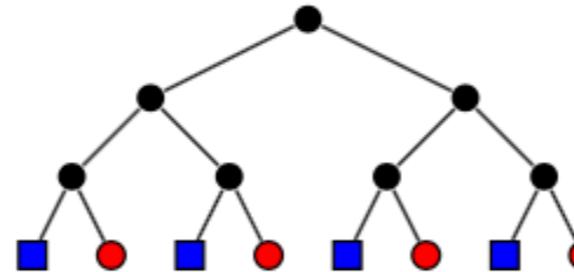
Wasserstein distance
(features only)



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

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

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

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

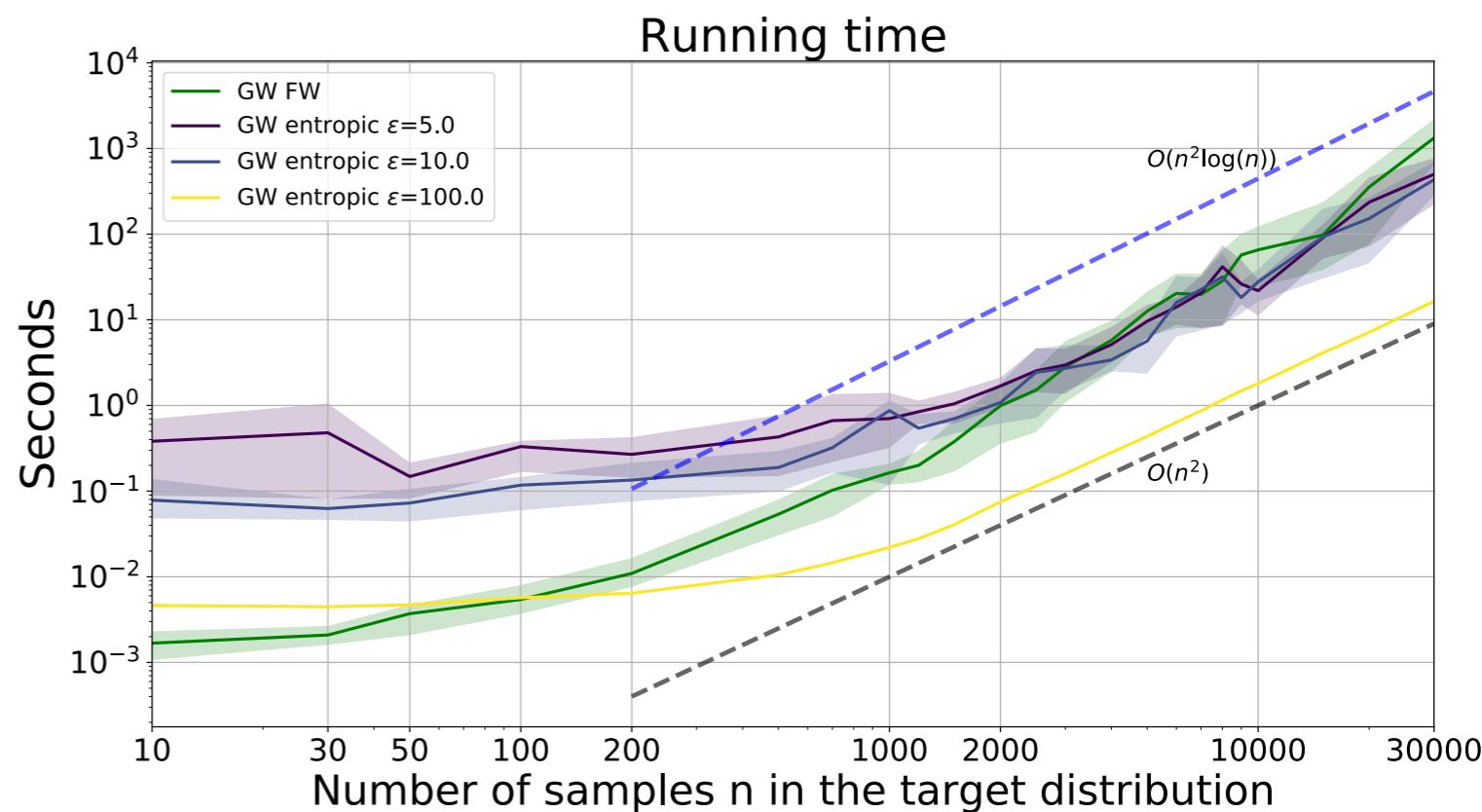
Find a **sparse** solution. FW gap = $O\left(\frac{1}{\sqrt{n_{iter}}}\right)$

Algorithm 1 Conditional Gradient (CG) for FGW

```
1:  $\pi^{(0)} \leftarrow \mathbf{h}\mathbf{g}^\top$ 
2: for  $i = 1, \dots$ , do
3:    $\mathbf{G} \leftarrow$  Gradient from GW loss w.r.t.  $\pi^{(i-1)}$ 
4:    $\tilde{\pi}^{(i)} \leftarrow$  Solve OT with ground loss  $\mathbf{G}$ 
5:    $\tau^{(i)} \leftarrow$  Line-search for GW loss with  $\tau \in (0, 1)$  (closed-form)
6:    $\pi^{(i)} \leftarrow (1 - \tau^{(i)})\pi^{(i-1)} + \tau^{(i)}\tilde{\pi}^{(i)}$ 
7: 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
 - (conservation of the weights) $h_i = g_{\sigma(i)}$
 - (conservation of the features) $a_i = b_{\sigma(i)}$
 - (conservation of the structures) $C_1(i, k) = C_2(\sigma(i), \sigma(k))$

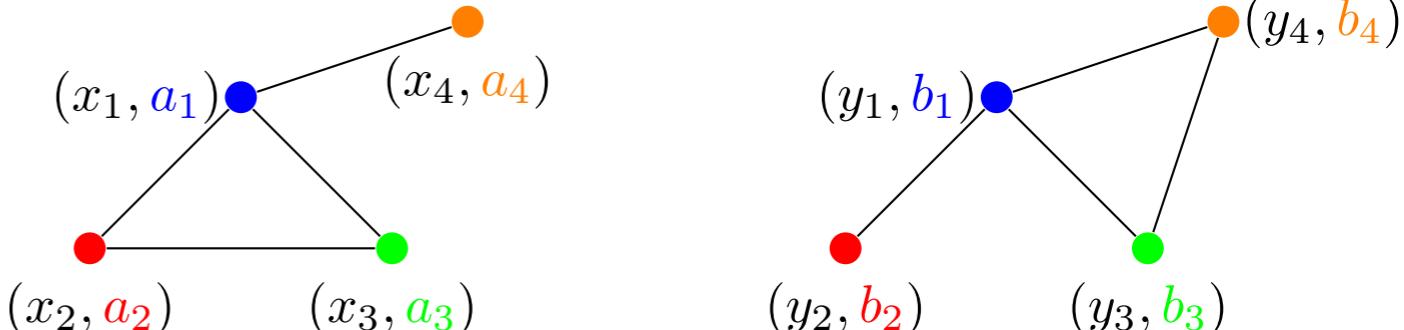
Optimal Transport for structured data

Fused Gromov-Wasserstein distance

A distance w.r.t strong isomorphism

- $FGW \geq 0$ and satisfies the triangle inequality
- C_1, C_2 distances. $FGW = 0$ iff $\exists \sigma$ permutations of the nodes
 - (conservation of the weights) $h_i = g_{\sigma(i)}$
 - (conservation of the features) $a_i = b_{\sigma(i)}$
 - (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
 - (conservation of the weights) $h_i = g_{\sigma(i)}$
 - (conservation of the features) $a_i = b_{\sigma(i)}$
 - (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 $e^{-\frac{1}{\beta} FGW(\mathcal{G}_i, \mathcal{G}_j)}$

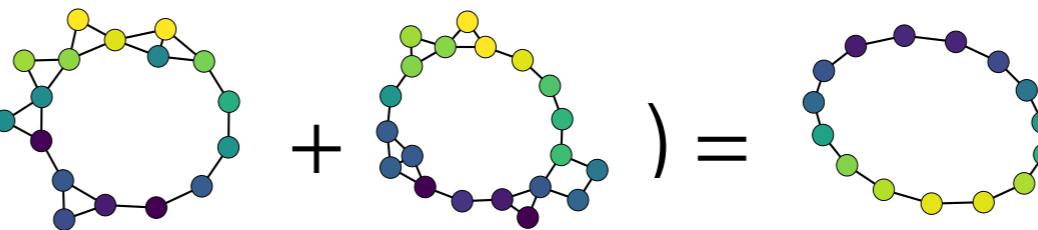
Compare with graph kernel approaches + GCN on benchmark datasets

DATASET	LABELED GRAPHS			SOCIAL GRAPHS IMDB-B	VECTOR ATTRIBUTES GRAPH		
	MUTAG	PTC	NCI1		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

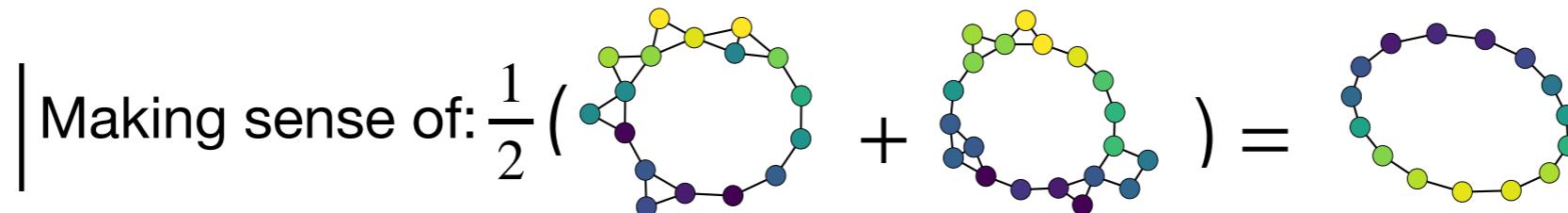
FGW barycenter

Making sense of: $\frac{1}{2} \left(\text{Diagram 1} + \text{Diagram 2} \right) = \text{Diagram 3}$



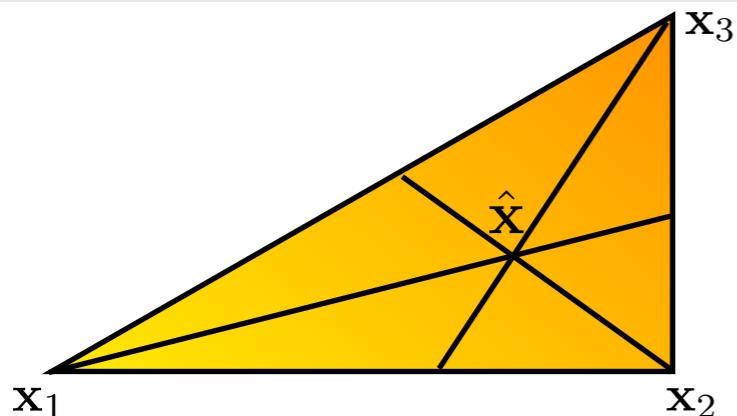
Optimal Transport for structured data

FGW barycenter



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

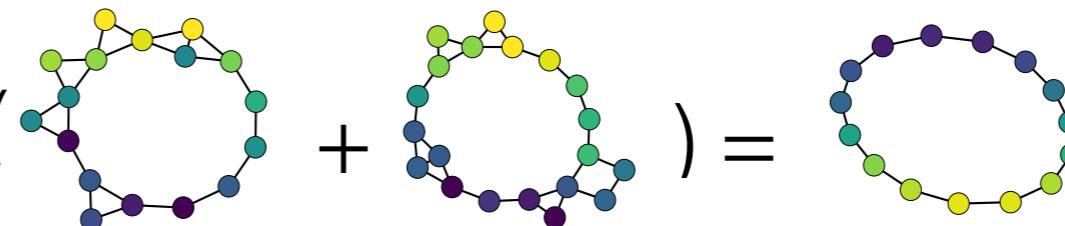
$$\inf_{\mathbf{x} \in \mathbb{R}^d} \sum_{i=1}^n \lambda_i \|\mathbf{x} - \mathbf{x}_i\|_2^2$$



Optimal Transport for structured data

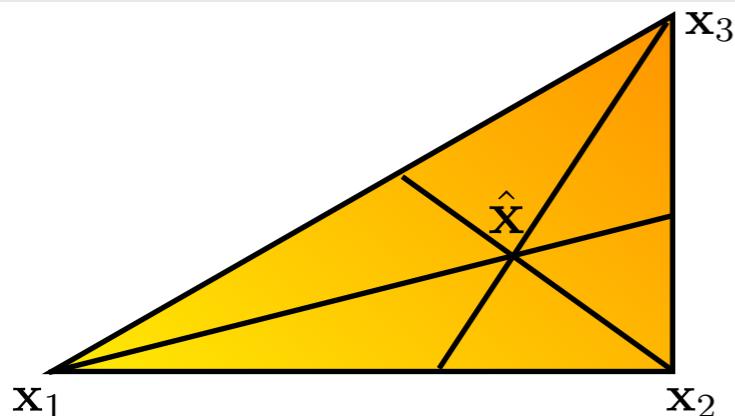
FGW barycenter

Making sense of: $\frac{1}{2} \left(\text{Diagram 1} + \text{Diagram 2} \right) = \text{Diagram 3}$



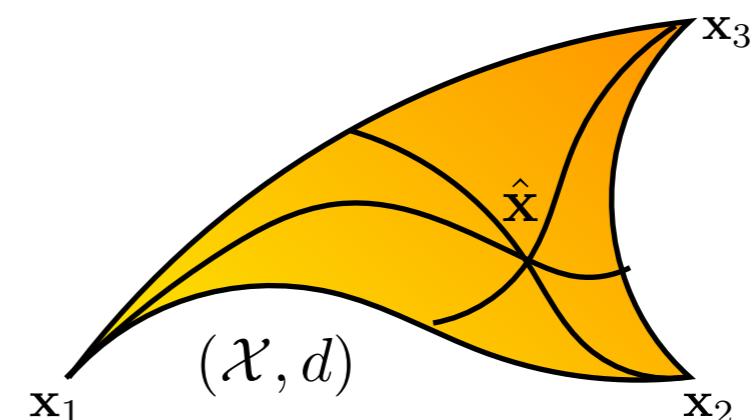
Euclidean Barycenter: $(\mathbb{R}^d, \|\cdot\|_2)$

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



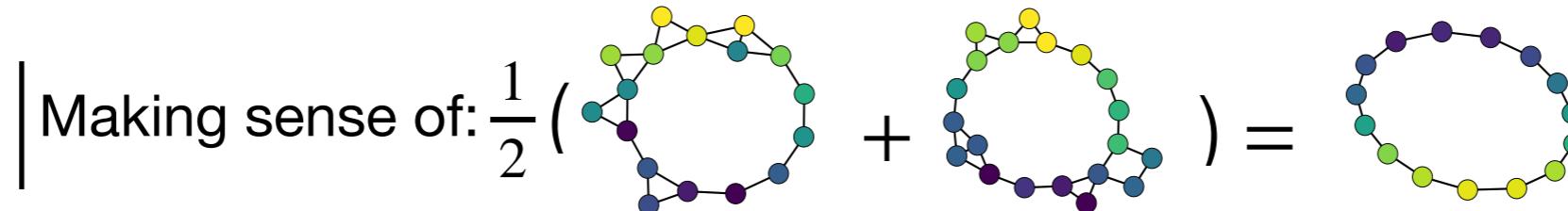
Fréchet Barycenter: (\mathcal{X}, d) metric space

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



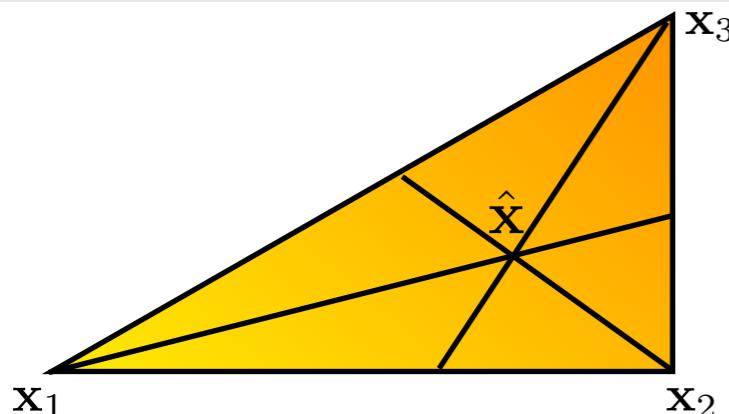
Optimal Transport for structured data

FGW barycenter



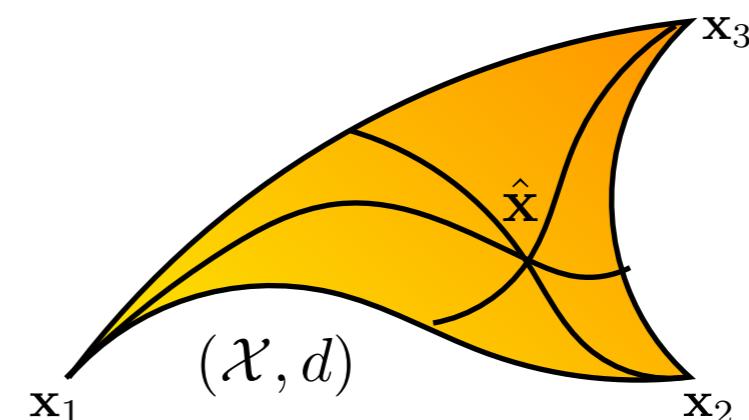
Euclidean Barycenter: $(\mathbb{R}^d, \|\cdot\|_2)$

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



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

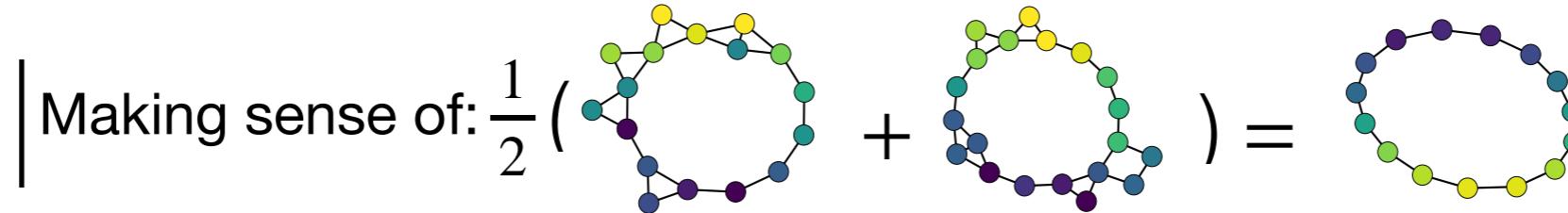
$$\min_{\mu} \sum_{k=1}^K \lambda_k FGW_{q,\alpha}(\mu, \mu_k)$$

Barycenter of labeled graphs, relational data with attributes

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

Optimal Transport for structured data

FGW barycenter



FGW barycenter

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

Barycenter of labeled graphs, relational data with attributes

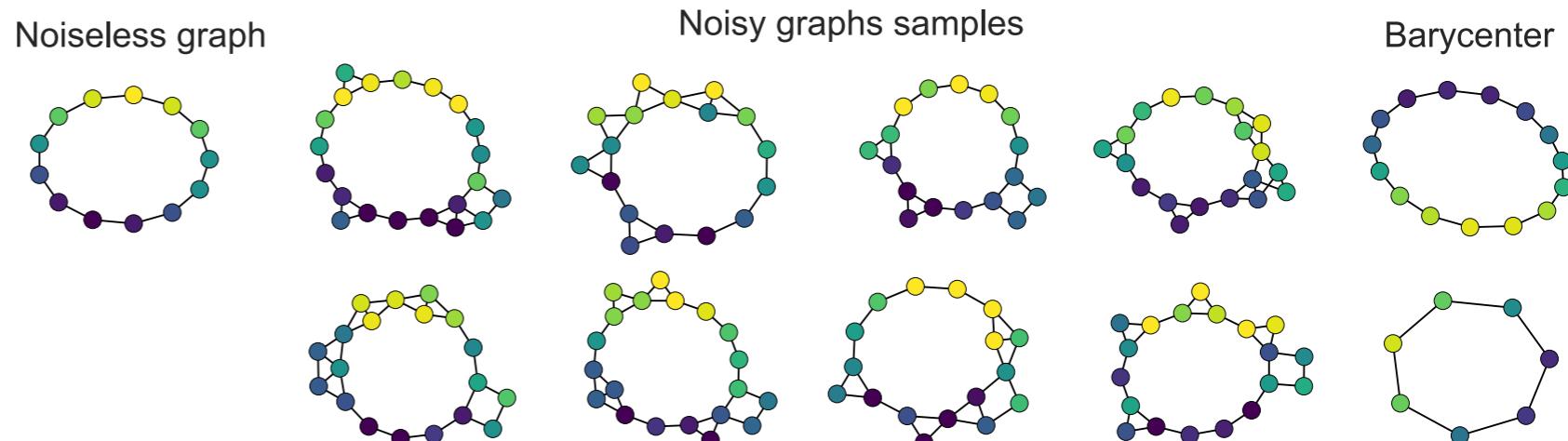
Consider feature space $\Omega = (\mathbb{R}^d, \|\cdot\|_2^2)$ structured data $(\mathbf{C}_k, \mathbf{B}_k, \mathbf{h}_k)_{k=1}^K$

Algorithm 1 FGW barycenter

```

1: Initialize  $\mathbf{C} \leftarrow \mathbf{C}_0, \mathbf{A} \leftarrow \mathbf{A}_0$ .
2: while not converged do
3:   for  $k = 1 \dots K$  do
4:      $\pi_k \leftarrow FGW(\mathbf{M}_{\mathbf{AB}_k}, \mathbf{C}, \mathbf{C}_k, \mathbf{h}, \mathbf{h}_k)$ 
5:   end for
6:    $\mathbf{C} \leftarrow \frac{1}{\mathbf{h}\mathbf{h}^T} \sum_{k=1}^K \lambda_k \pi_k^T \mathbf{C}_k \pi_k$ 
7:    $\mathbf{A} \leftarrow \sum_{k=1}^K \lambda_k \mathbf{B}_k \pi_k^T \text{diag}(\frac{1}{\mathbf{h}})$ 
8: end while

```



Optimal Transport for structured data

Summarization of graph

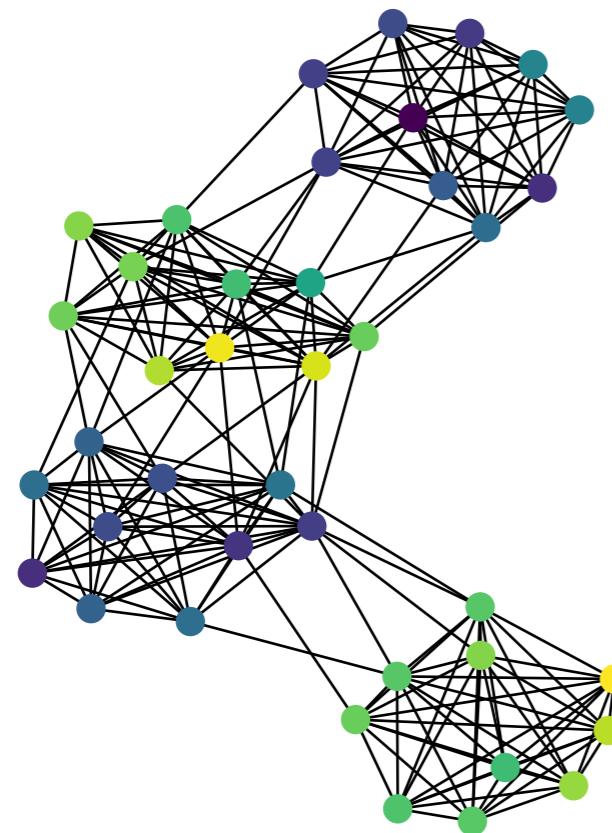
FGW coarsening

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

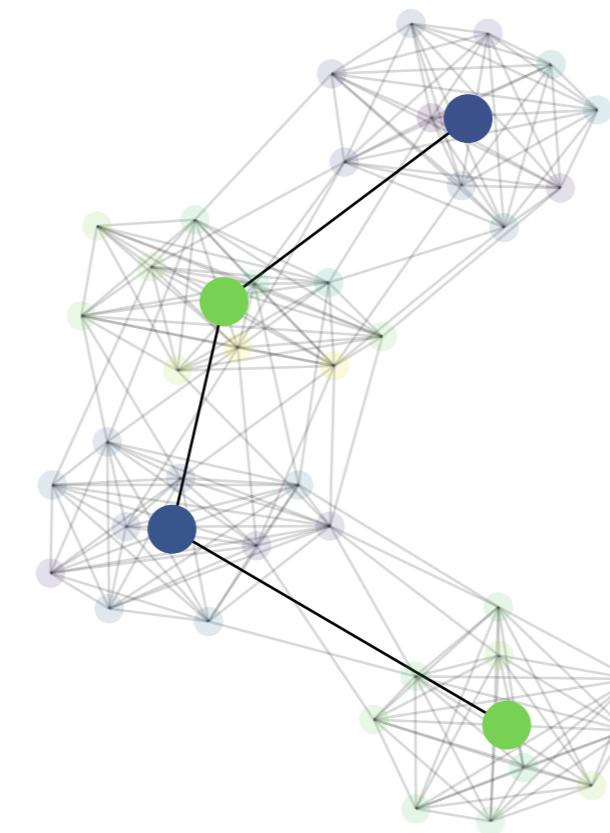
Given a labeled graph we look for the closest graph w.r.t FGW with fewer nodes

Projection w.r.t FGW \rightarrow barycenter problem with $K = 1$

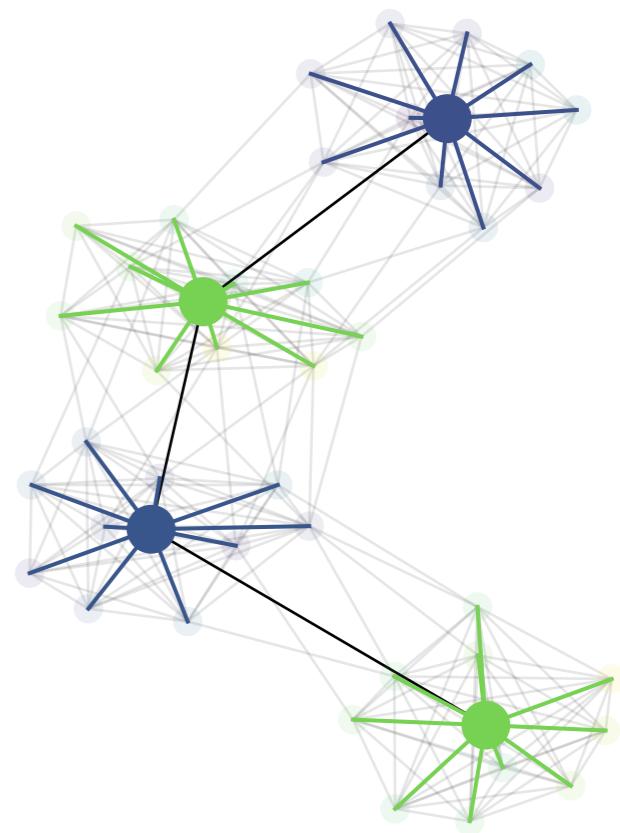
Graph with communities



Approximate Graph



Clustering with transport matrix



Optimal Transport for structured data

Summarization of graph

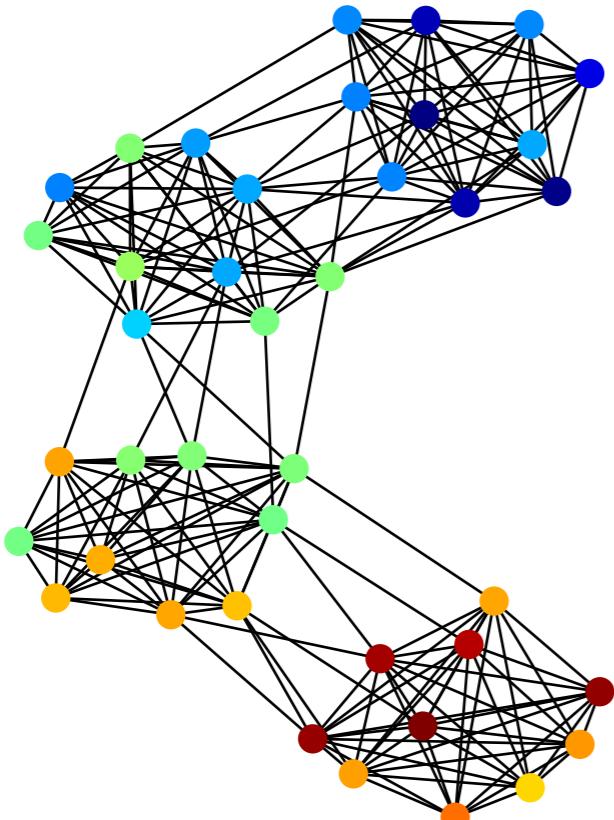
FGW coarsening

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

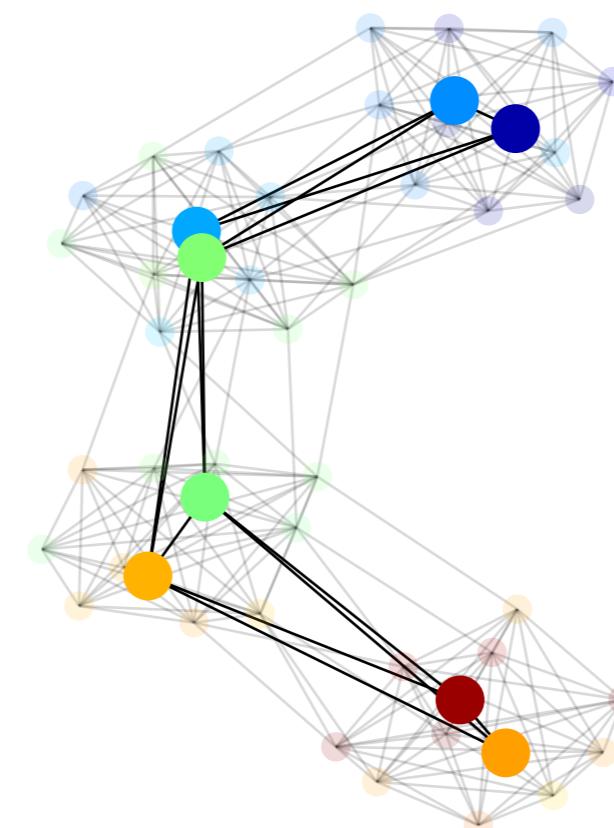
Given a labeled graph we look for the closest graph w.r.t FGW with fewer nodes

Projection w.r.t FGW \rightarrow barycenter problem with $K = 1$

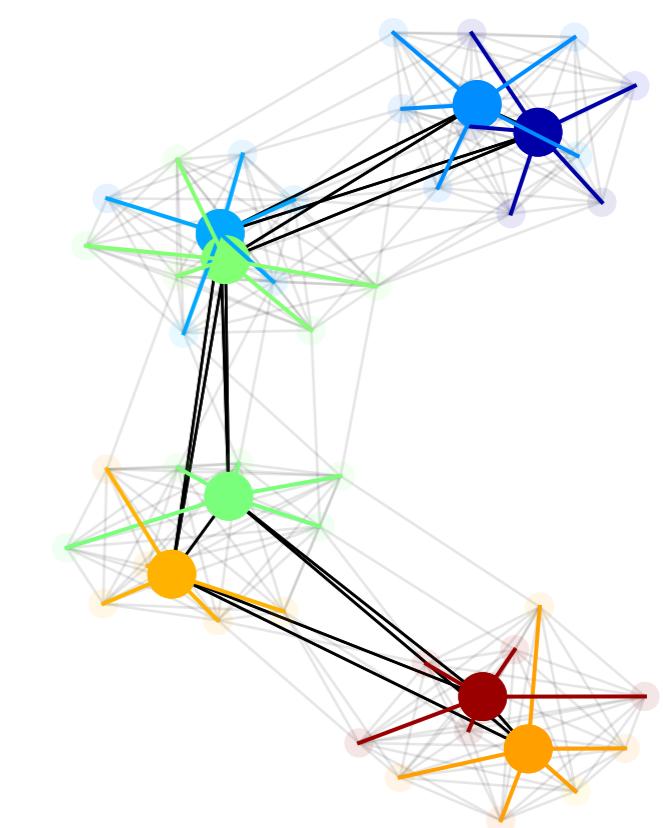
Graph with bimodal communities



Approximate Graph



Clustering with transport matrix



Optimal Transport for structured data

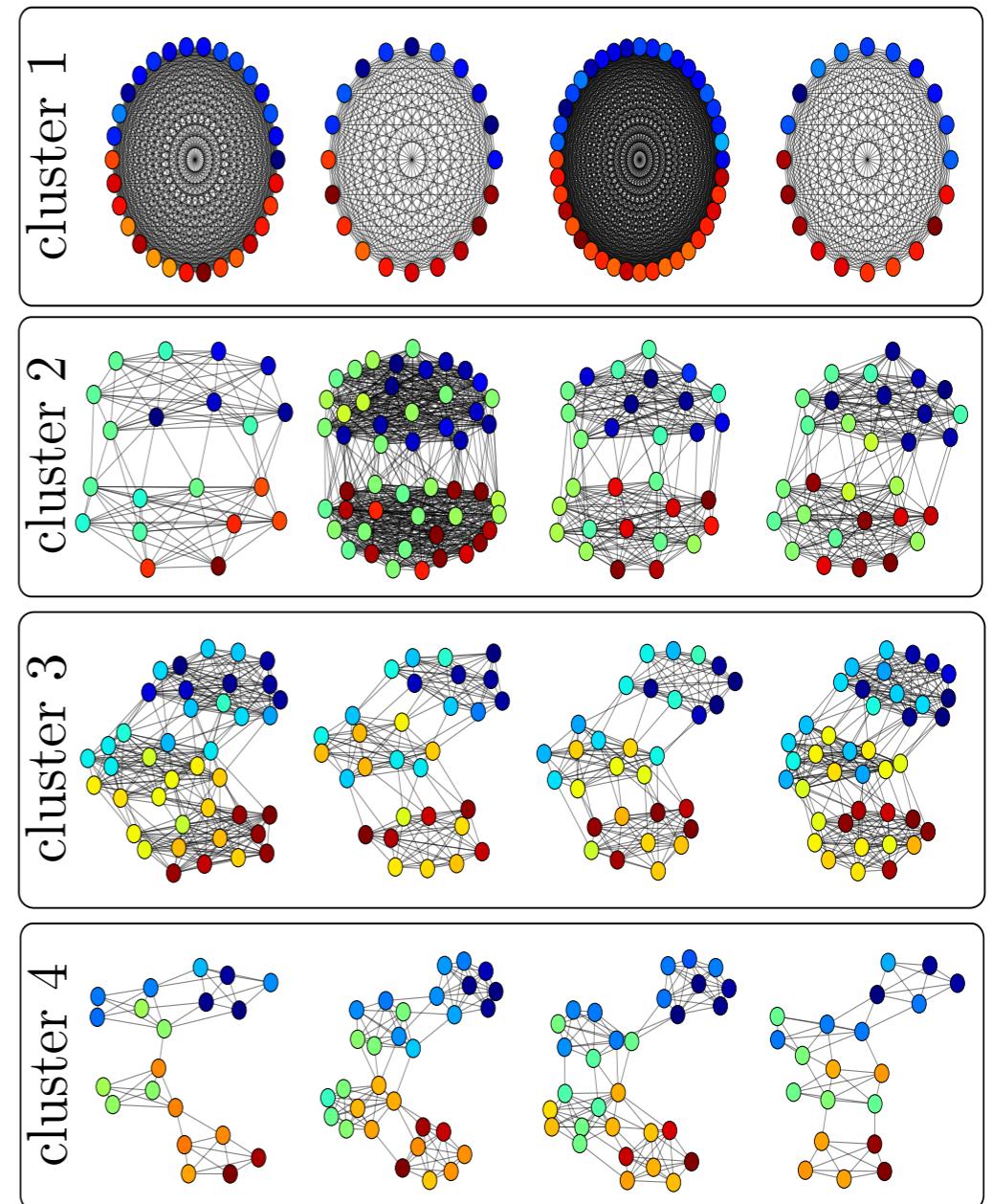
FGW clustering

Given a set of labeled graphs -> k-means using FGW barycenter

Algorithm 1 FGW clustering

```
1: Number of clusters  $K$ . Labeled graphs  $(\mathbf{C}_i, \mathbf{B}_i, \mathbf{h}_i)_{i \in [N]}$ 
2: Initialize centroids  $\forall k \in [K], \mathbf{C}_k \leftarrow \mathbf{C}_0, \mathbf{A}_k \leftarrow \mathbf{A}_0$ .
3: while not converged do
4:   Calculate  $N \times K$  FGW distances.
5:   for  $i = 1 \dots N$  do
6:     Assign  $(\mathbf{C}_i, \mathbf{B}_i, \mathbf{h}_i)$  to a cluster  $k \in [K]$ 
7:   end for
8:   for  $k = 1 \dots K$  do
9:      $\mathbf{C}_k, \mathbf{A}_k \leftarrow \text{FGW barycenter}((\mathbf{C}_i, \mathbf{B}_i, \mathbf{h}_i)_{i \in \text{cluster } k})$ 
10:  end for
11: end while
```

Training dataset examples

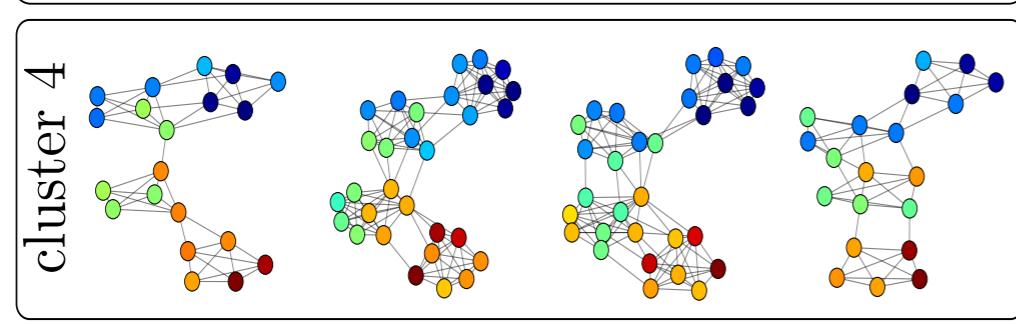
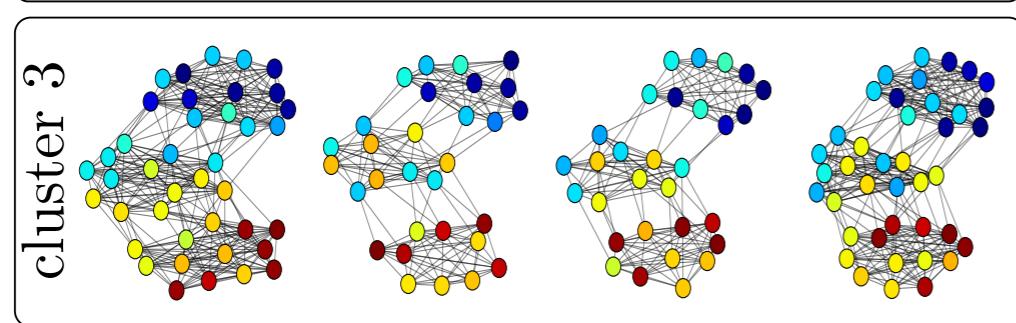
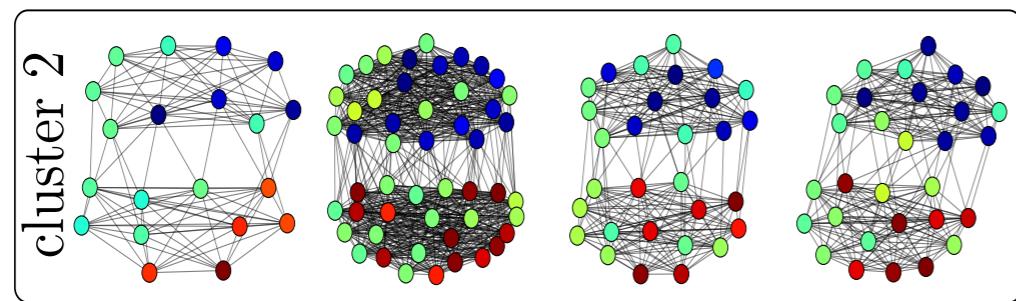
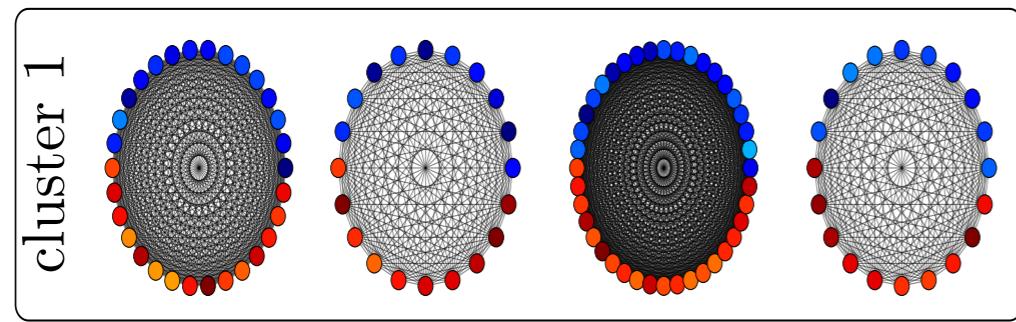


Optimal Transport for structured data

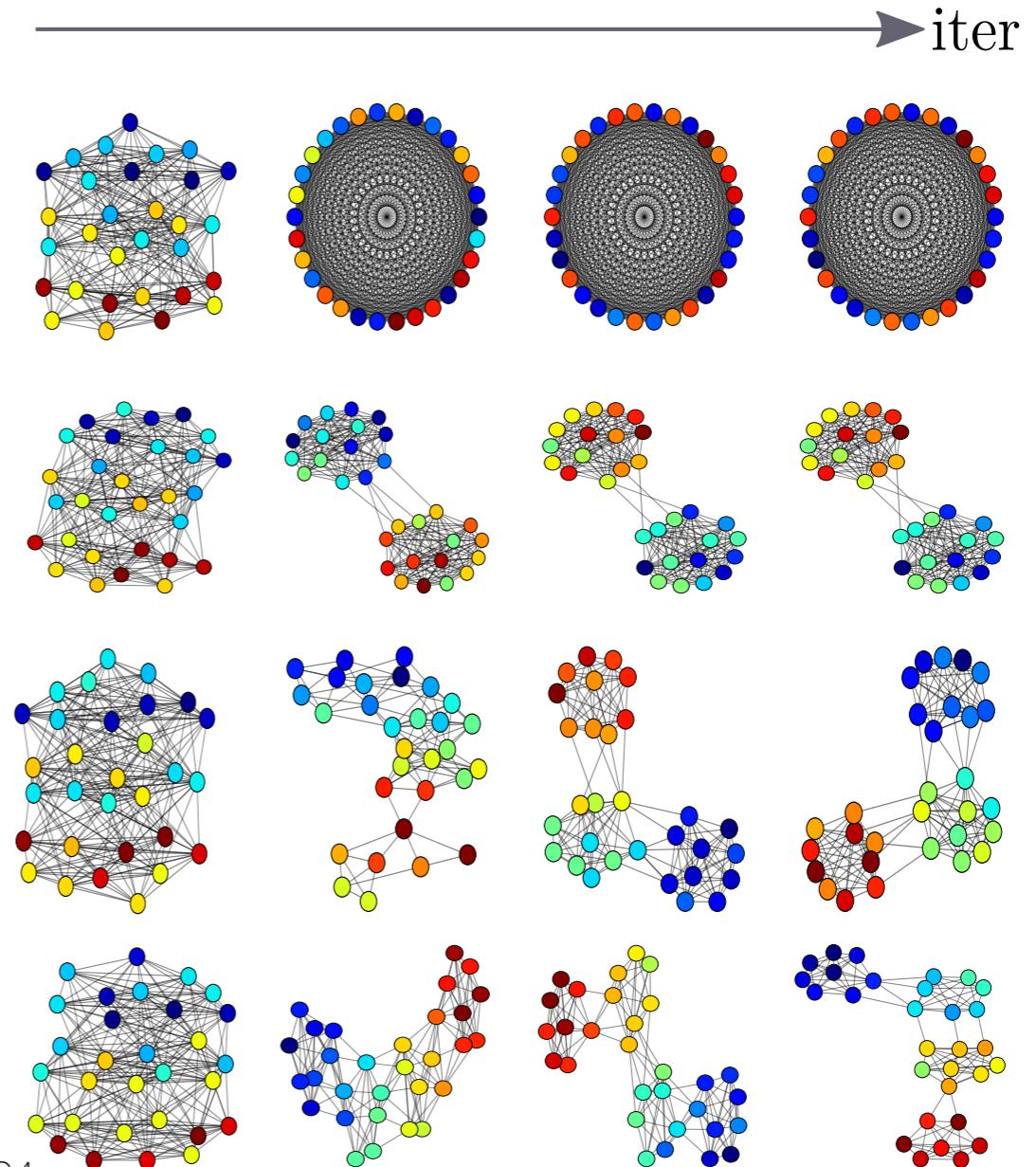
FGW clustering

Given a set of labeled graphs -> k-means using FGW barycenter

Training dataset examples



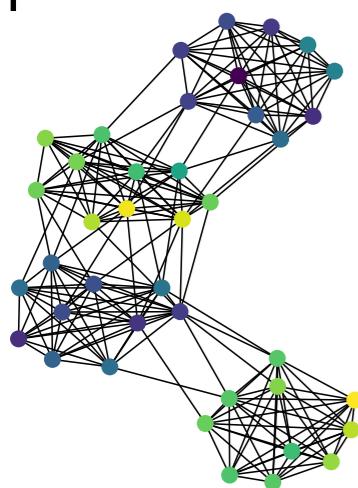
Centroids



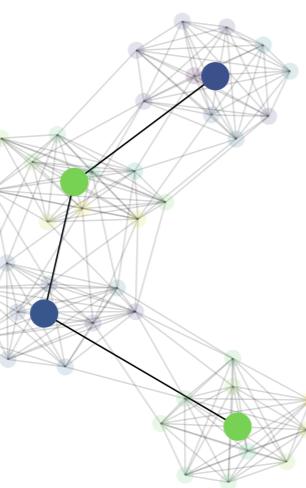
Optimal Transport for structured data

Conclusion

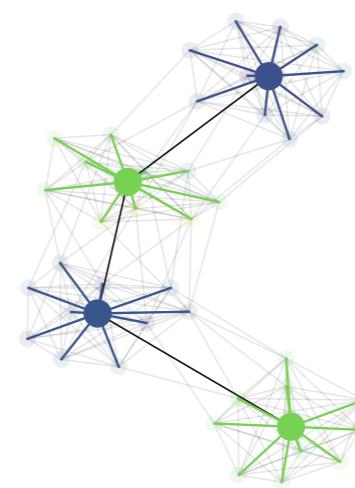
Graph with communities



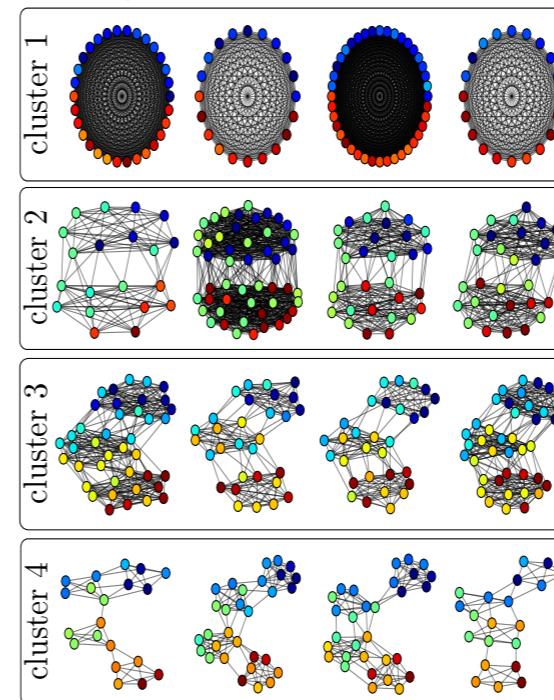
Approximate Graph



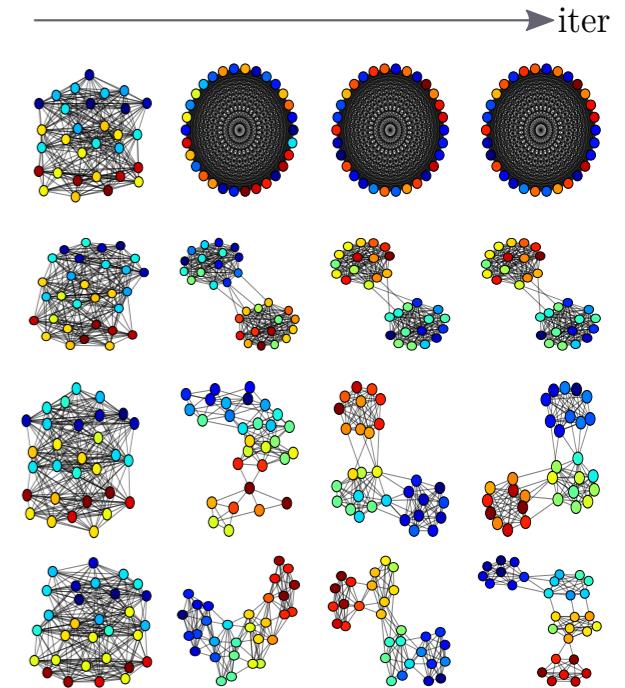
Clustering with transport matrix



Training dataset examples



Centroids



FGW

| OT method for structured data (**whatever sizes of graphs**)

| Provides a soft assignments of nodes + **distance between labeled graphs**

| Can be used for classification + summarization + clustering

Perspectives

| Learn structure matrices

| Use it for dynamic graph: add a temporal part

| Other formulation: match only a small portion of the nodes

CO-Optimal Transport

CO-Optimal Transport

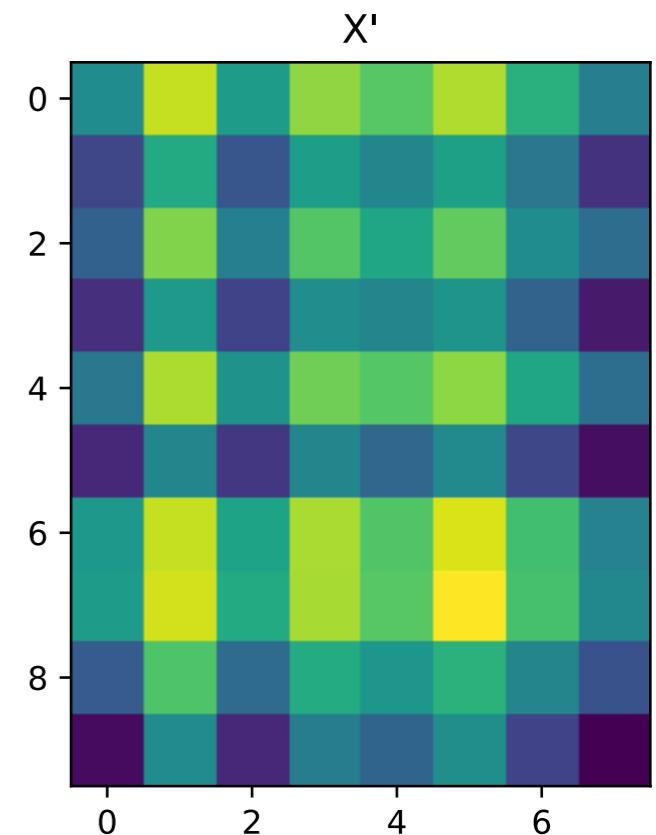
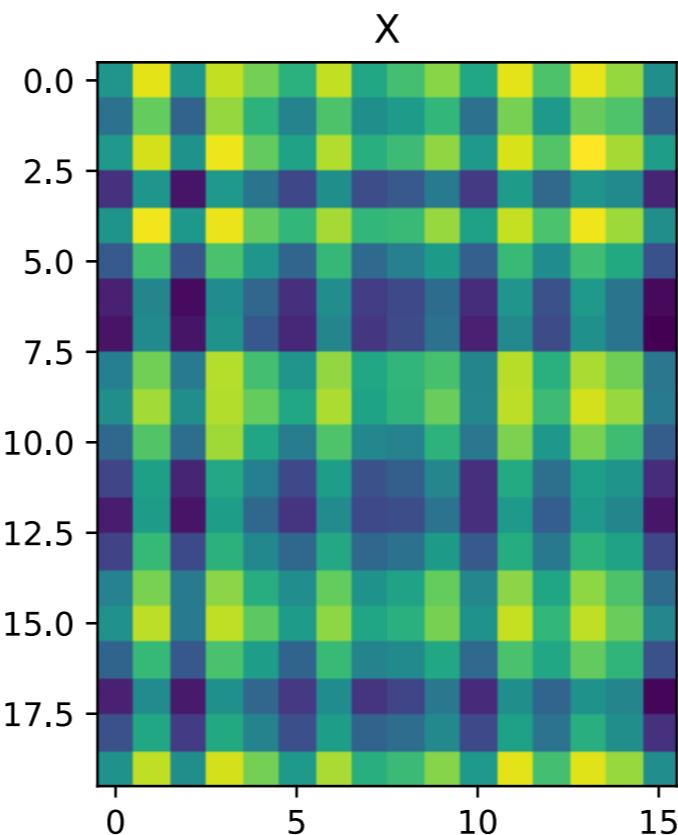
Motivations

Two heterogeneous datasets

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$$

$$\mathbf{X}' = [\mathbf{x}'_1, \dots, \mathbf{x}'_{n'}]^T \in \mathbb{R}^{n' \times d'}$$

Row= samples, columns= features



We want to measure the similarity of these two datasets (interpretable way)

Image registration [Haker 2001], HDA [Yang 2018], Word embeddings [Alvarez 2018]

CO-Optimal Transport

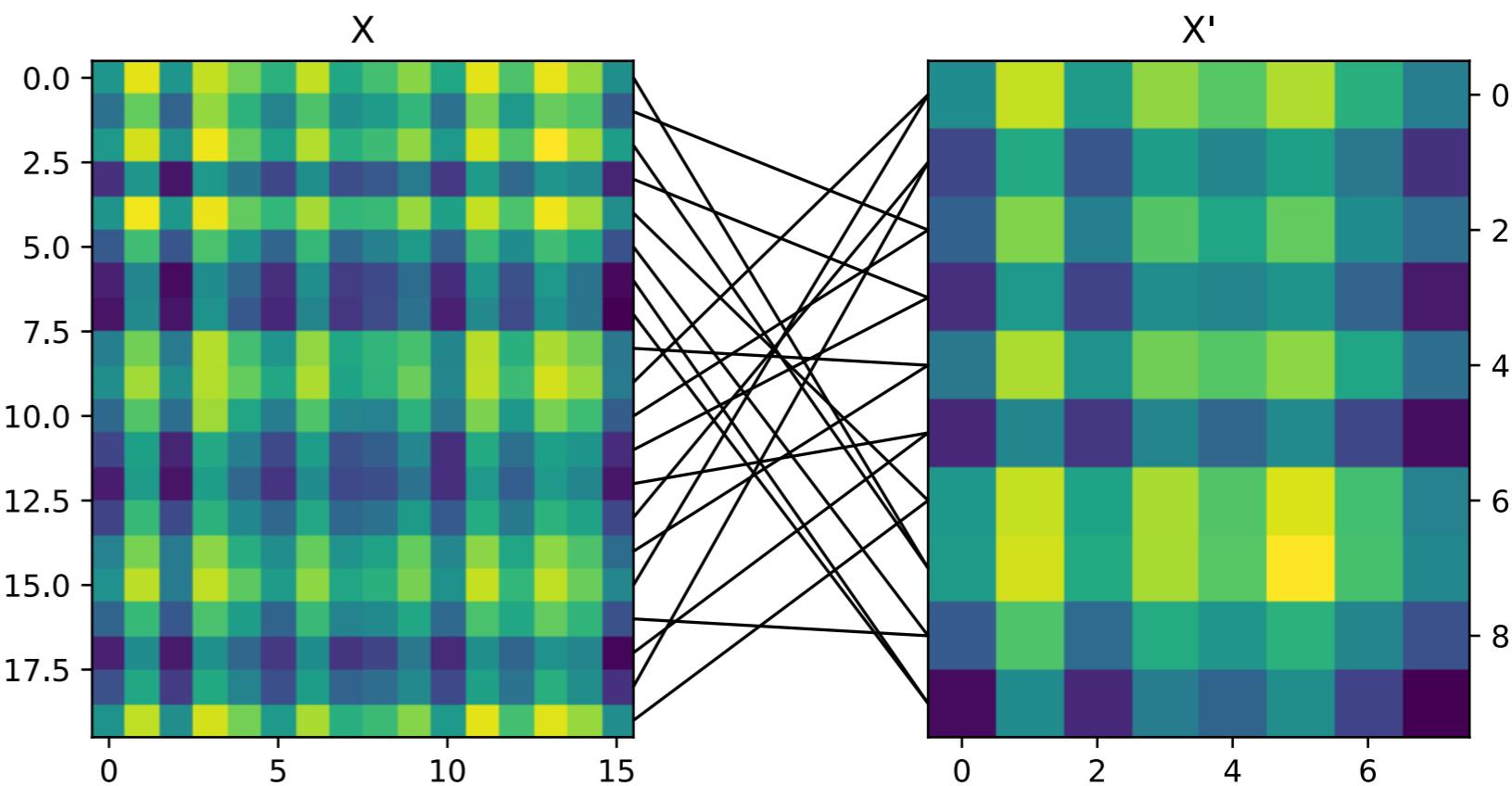
Motivations

Two heterogeneous datasets

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$$

$$\mathbf{X}' = [\mathbf{x}'_1, \dots, \mathbf{x}'_{n'}]^T \in \mathbb{R}^{n' \times d'}$$

Row= samples, columns= features/variable



We can apply Gromov-Wasserstein based on the pairwise distances

$$c_X(\mathbf{x}_i, \mathbf{x}_j)$$
$$c_{X'}(\mathbf{x}'_i, \mathbf{x}'_j)$$

The OT matrix gives a reordering of the samples

CO-Optimal Transport

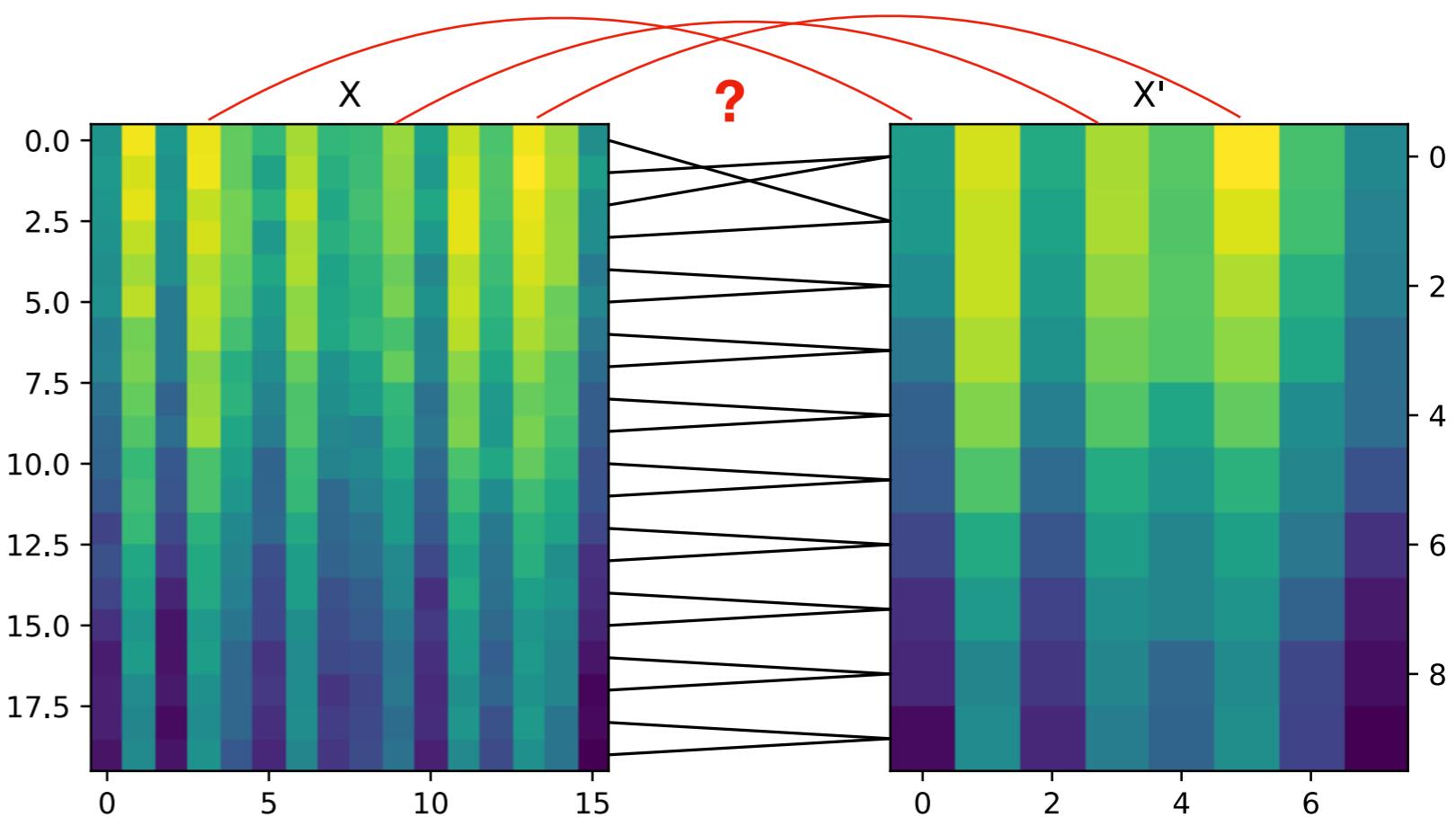
Motivations

Two heterogeneous datasets

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$$

$$\mathbf{X}' = [\mathbf{x}'_1, \dots, \mathbf{x}'_{n'}]^T \in \mathbb{R}^{n' \times d'}$$

Row= samples, columns= features/variable



We can apply Gromov-Wasserstein based on the pairwise distances

$$c_X(\mathbf{x}_i, \mathbf{x}_j)$$
$$c_{X'}(\mathbf{x}'_i, \mathbf{x}'_j)$$

The OT matrix gives a reordering of the samples

But discards the relationship between **the features**...

CO-Optimal Transport

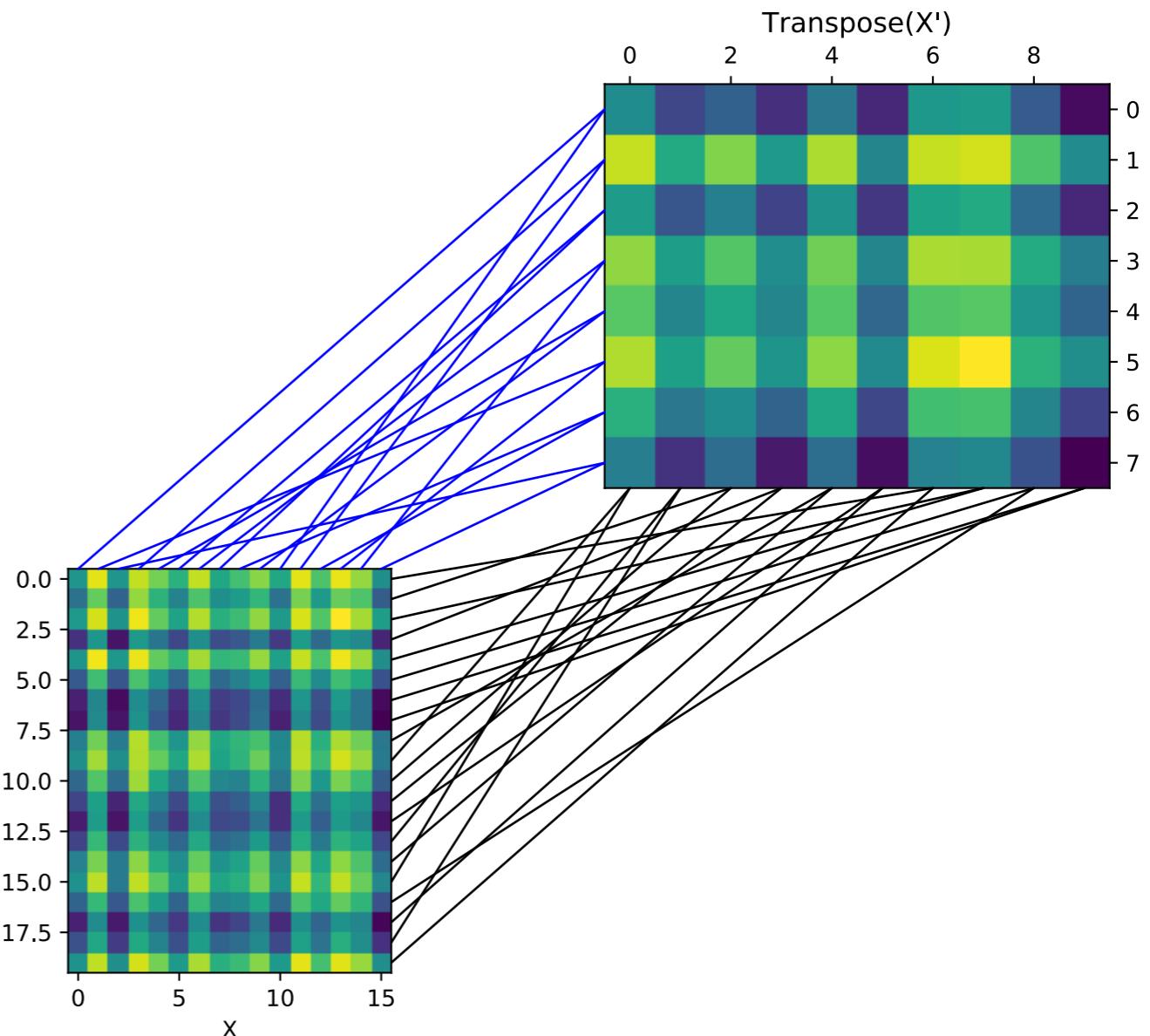
Motivations

Two heterogeneous datasets

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$$

$$\mathbf{X}' = [\mathbf{x}'_1, \dots, \mathbf{x}'_{n'}]^T \in \mathbb{R}^{n' \times d'}$$

Row= samples, columns= features/variable



- | The objective of COOT is to estimate a transport matrix between the samples **and** one between the features
- | These matrices are estimated jointly and can be used for interpreting relationships across spaces

CO-Optimal Transport

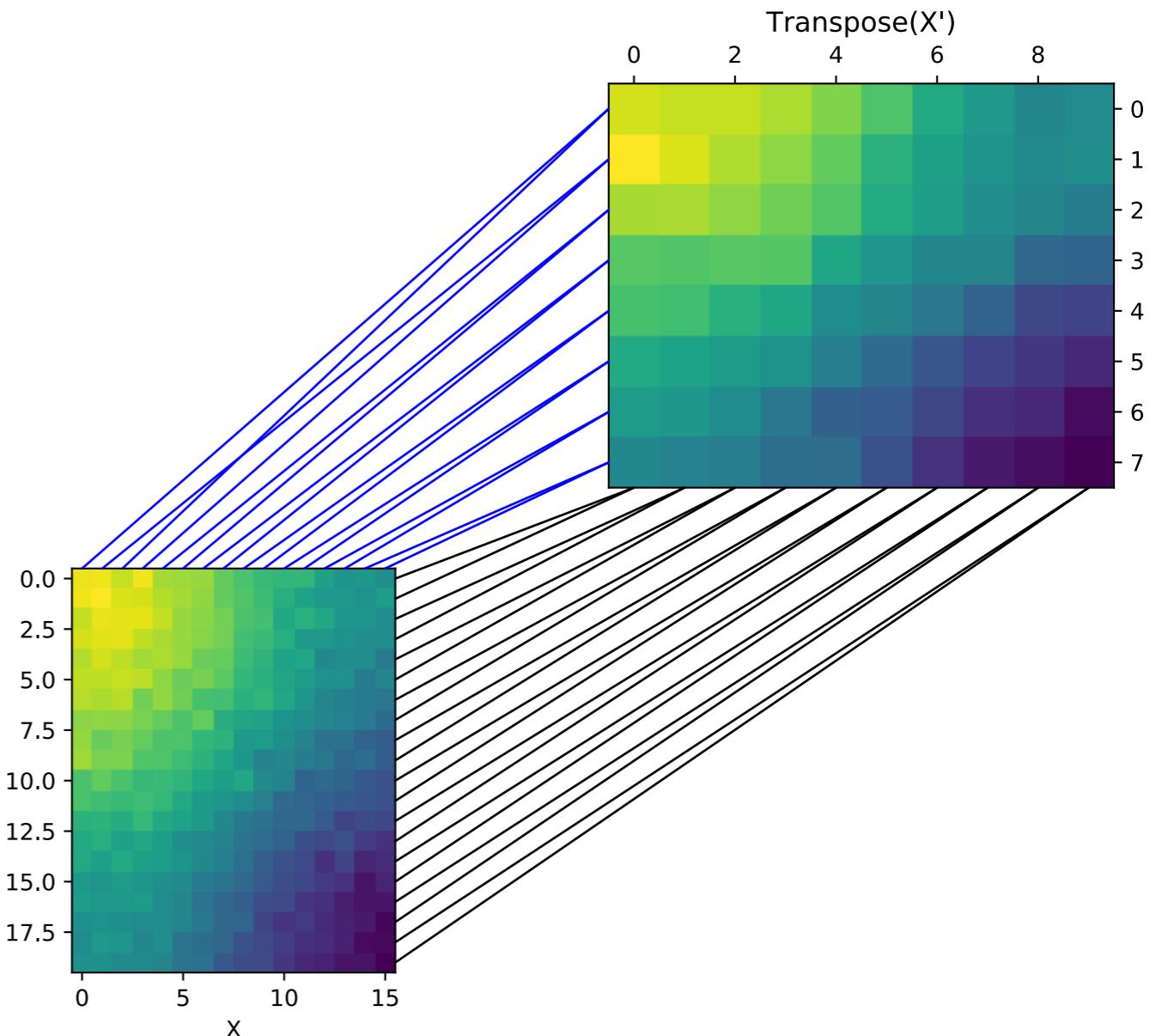
Motivations

Two heterogeneous datasets

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$$

$$\mathbf{X}' = [\mathbf{x}'_1, \dots, \mathbf{x}'_{n'}]^T \in \mathbb{R}^{n' \times d'}$$

Row= samples, columns= features/variable



- | The objective of COOT is to estimate a transport matrix between the samples **and** one between the features
- | These matrices are estimated jointly and can be used for interpreting relationships across spaces

CO-Optimal Transport

Motivations

Two heterogeneous datasets

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$$

$$\mathbf{X}' = [\mathbf{x}'_1, \dots, \mathbf{x}'_{n'}]^T \in \mathbb{R}^{n' \times d'}$$

Weights (histograms)

Samples: $\mathbf{w} \in \Sigma_n, \mathbf{w}' \in \Sigma_{n'}$

Features: $\mathbf{v} \in \Sigma_d, \mathbf{v}' \in \Sigma_{d'}$

CO-Optimal Transport

$$\min_{\begin{array}{l} \boldsymbol{\pi}^s \in \Pi(\mathbf{w}, \mathbf{w}') \\ \boldsymbol{\pi}^v \in \Pi(\mathbf{v}, \mathbf{v}') \end{array}} \sum_{i,j,k,l} |\mathbf{X}_{i,k} - \mathbf{X}'_{j,l}|^p \boldsymbol{\pi}_{i,j}^s \boldsymbol{\pi}_{k,l}^v$$

$\boldsymbol{\pi}^s$: transport matrix between the samples

$\boldsymbol{\pi}^v$: transport matrix between the features/variables

CO-Optimal Transport

Motivations

Two heterogeneous datasets

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$$

$$\mathbf{X}' = [\mathbf{x}'_1, \dots, \mathbf{x}'_{n'}]^T \in \mathbb{R}^{n' \times d'}$$

Weights (histograms)

Samples: $\mathbf{w} \in \Sigma_n, \mathbf{w}' \in \Sigma_{n'}$

Features: $\mathbf{v} \in \Sigma_d, \mathbf{v}' \in \Sigma_{d'}$

CO-Optimal Transport

$$\min_{\begin{array}{l} \boldsymbol{\pi}^s \in \Pi(\mathbf{w}, \mathbf{w}') \\ \boldsymbol{\pi}^v \in \Pi(\mathbf{v}, \mathbf{v}') \end{array}} \sum_{i,j,k,l} |\mathbf{X}_{i,k} - \mathbf{X}'_{j,l}|^p \boldsymbol{\pi}_{i,j}^s \boldsymbol{\pi}_{k,l}^v$$

$\boldsymbol{\pi}^s$: transport matrix between the samples

$\boldsymbol{\pi}^v$: transport matrix between the features/variables

Regularized version: add an entropy term for each transport matrix

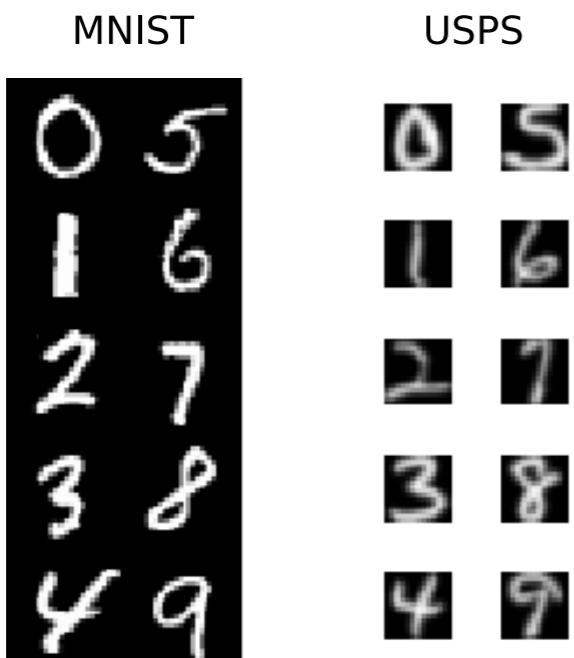
CO-Optimal Transport

Formulation & example

CO-Optimal Transport

$$\min_{\begin{array}{l} \boldsymbol{\pi}^s \in \Pi(\mathbf{w}, \mathbf{w}') \\ \boldsymbol{\pi}^v \in \Pi(\mathbf{v}, \mathbf{v}') \end{array}} \sum_{i,j,k,l} |\mathbf{X}_{i,k} - \mathbf{X}'_{j,l}|^p \boldsymbol{\pi}_{i,j}^s \boldsymbol{\pi}_{k,l}^v$$

MNIST/USPS example:



Samples: images, Features: pixels

$$n = n' = 300$$

$$d = 256, d' = 784$$

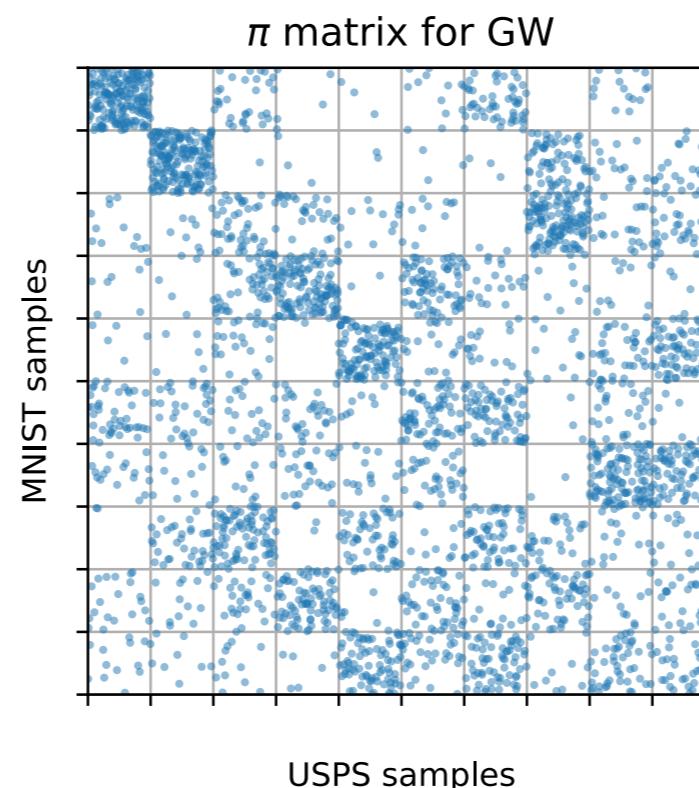
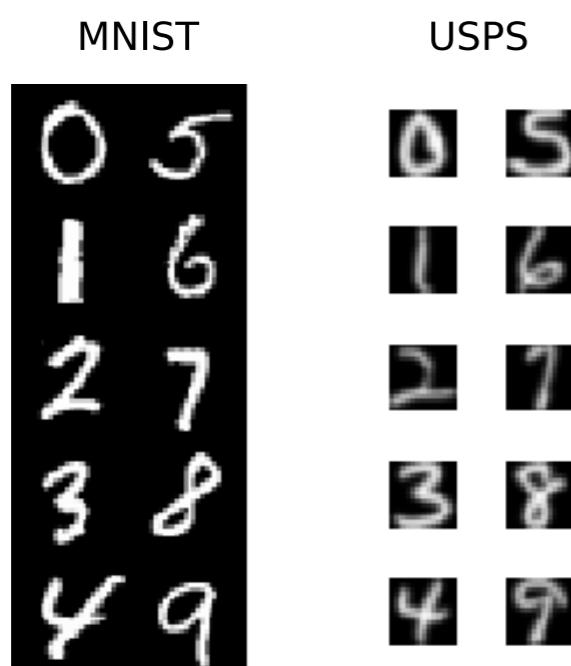
CO-Optimal Transport

Formulation & example

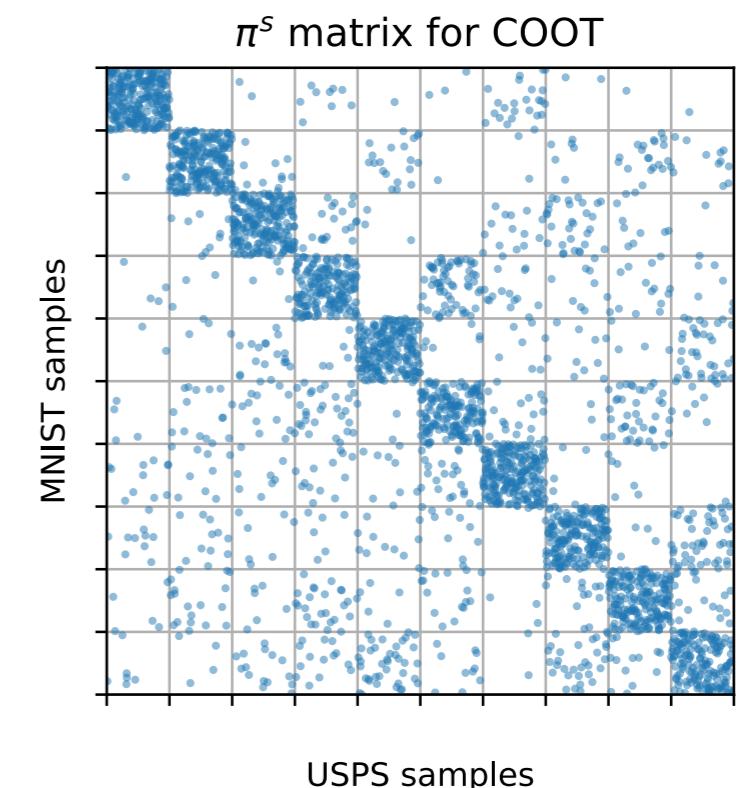
CO-Optimal Transport

$$\min_{\begin{array}{l} \pi^s \in \Pi(\mathbf{w}, \mathbf{w}') \\ \pi^v \in \Pi(\mathbf{v}, \mathbf{v}') \end{array}} \sum_{i,j,k,l} |\mathbf{X}_{i,k} - \mathbf{X}'_{j,l}|^p \pi^s_{i,j} \pi^v_{k,l}$$

MNIST/USPS example:



Visualization of π^s



Better class correspondence

CO-Optimal Transport

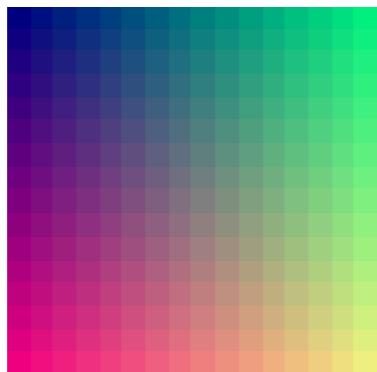
Formulation & example

CO-Optimal Transport

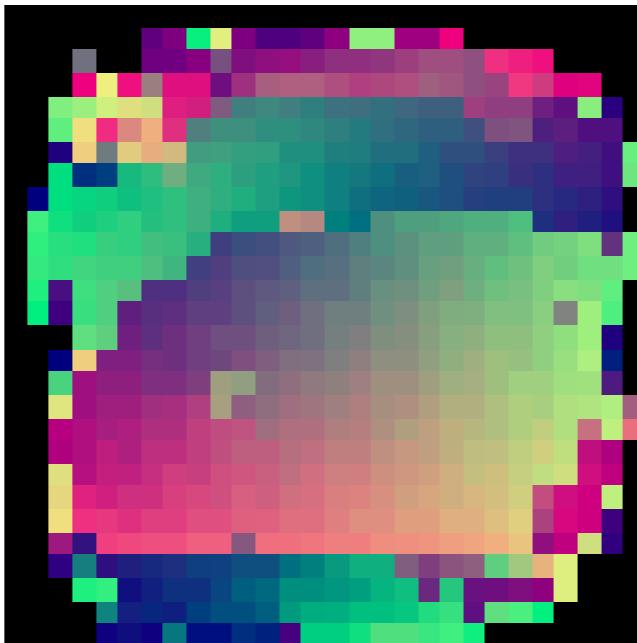
$$\min_{\begin{array}{l} \pi^s \in \Pi(\mathbf{w}, \mathbf{w}') \\ \pi^v \in \Pi(\mathbf{v}, \mathbf{v}') \end{array}} \sum_{i,j,k,l} |\mathbf{X}_{i,k} - \mathbf{X}'_{j,l}|^p \pi^s_{i,j} \pi^v_{k,l}$$

MNIST/USPS example:

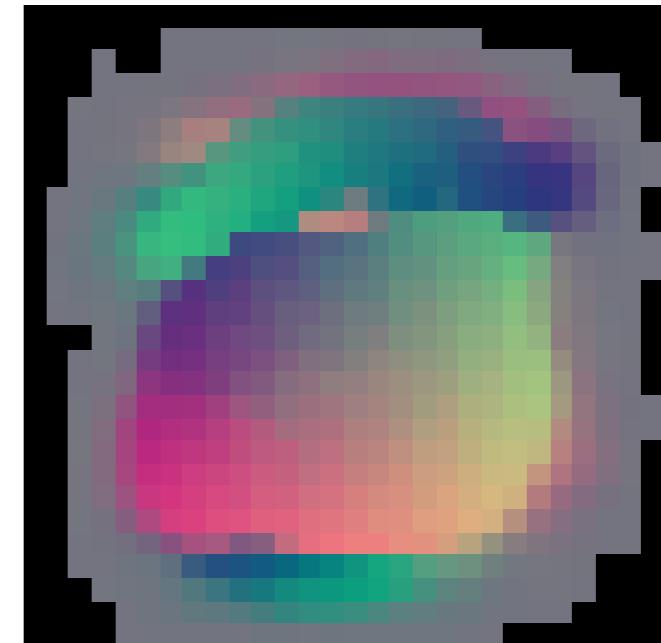
USPS colored pixels



MNIST pixels through π^v



Visualization of π^v



MNIST pixels through entropic π^v

Spatial structure preserved (without supervision!)

CO-Optimal Transport

Properties

A distance w.r.t permutations of the datasets

Theorem. *COOT is a distance*

- COOT symmetric and satisfies the triangular inequality,

$$\text{COOT}(\mathbf{X}, \mathbf{X}'') \leq \text{COOT}(\mathbf{X}, \mathbf{X}') + \text{COOT}(\mathbf{X}', \mathbf{X}'')$$

- Uniform weights. $\text{COOT}(\mathbf{X}, \mathbf{X}') = 0$ iff $n = n', d = d', \exists \sigma_1 \in S_n$ (samples) and $\exists \sigma_2 \in S_d$ (features):

$$\forall i, k \quad \mathbf{X}_{i,k} = \mathbf{X}'_{\sigma_1(i), \sigma_2(k)}$$

CO-Optimal Transport

Properties

A distance w.r.t permutations of the datasets

Theorem. *COOT is a distance*

- COOT symmetric and satisfies the triangular inequality,

$$\text{COOT}(\mathbf{X}, \mathbf{X}'') \leq \text{COOT}(\mathbf{X}, \mathbf{X}') + \text{COOT}(\mathbf{X}', \mathbf{X}'')$$

- Uniform weights. $\text{COOT}(\mathbf{X}, \mathbf{X}') = 0$ iff $n = n', d = d', \exists \sigma_1 \in S_n$ (samples) and $\exists \sigma_2 \in S_d$ (features):

$$\forall i, k \quad \mathbf{X}_{i,k} = \mathbf{X}'_{\sigma_1(i), \sigma_2(k)}$$

Relation with Gromov-Wasserstein

Theorem. • Let $\mathbf{C} \in \mathbb{R}^{n \times n}, \mathbf{C}' \in \mathbb{R}^{n' \times n'}$ be any symmetric matrices, then:

$$\text{COOT}(\mathbf{C}, \mathbf{C}', \mathbf{w}, \mathbf{w}', \mathbf{w}, \mathbf{w}') \leq \text{GW}(\mathbf{C}, \mathbf{C}', \mathbf{w}, \mathbf{w}').$$

- When \mathbf{C} and \mathbf{C}' are squared Euclidean distance matrices:

$$\text{COOT}(\mathbf{C}, \mathbf{C}', \mathbf{w}, \mathbf{w}', \mathbf{w}, \mathbf{w}') = \text{GW}(\mathbf{C}, \mathbf{C}', \mathbf{w}, \mathbf{w}')$$

and the optimal transport matrices $\pi_{108}^{\text{GW}} = \pi^s = \pi^v$.

CO-Optimal Transport

Solving COOT

CO-Optimal Transport

$$\min_{\begin{array}{l} \boldsymbol{\pi}^s \in \Pi(\mathbf{w}, \mathbf{w}') \\ \boldsymbol{\pi}^v \in \Pi(\mathbf{v}, \mathbf{v}') \end{array}} \sum_{i,j,k,l} |\mathbf{X}_{i,k} - \mathbf{X}'_{j,l}|^p \boldsymbol{\pi}_{i,j}^s \boldsymbol{\pi}_{k,l}^v$$

| Non-convex bilinear program: NP-Hard

| BCD procedure: alternates OT problems -> converges to a local minima [Konno 1976]

Algorithm 1 BCD for COOT

```
1:  $\boldsymbol{\pi}_{(0)}^s \leftarrow \mathbf{w}\mathbf{w}'^T, \boldsymbol{\pi}_{(0)}^v \leftarrow \mathbf{v}\mathbf{v}'^T, k \leftarrow 0$ 
2: while  $k < \text{maxIt}$  and  $err > 0$  do
3:    $\boldsymbol{\pi}_{(k)}^v \leftarrow OT(\mathbf{v}, \mathbf{v}', \mathbf{L}(\mathbf{X}, \mathbf{X}') \otimes \boldsymbol{\pi}_{(k-1)}^s)$ 
4:    $\boldsymbol{\pi}_{(k)}^s \leftarrow OT(\mathbf{w}, \mathbf{w}', \mathbf{L}(\mathbf{X}, \mathbf{X}') \otimes \boldsymbol{\pi}_{(k-1)}^v)$ 
5:    $err \leftarrow \|\boldsymbol{\pi}_{(k-1)}^v - \boldsymbol{\pi}_{(k)}^v\|_F$ 
6:    $k \leftarrow k + 1$ 
7: end while
```

CO-Optimal Transport

Solving COOT

CO-Optimal Transport

$$\min_{\begin{array}{l} \boldsymbol{\pi}^s \in \Pi(\mathbf{w}, \mathbf{w}') \\ \boldsymbol{\pi}^v \in \Pi(\mathbf{v}, \mathbf{v}') \end{array}} \sum_{i,j,k,l} |\mathbf{X}_{i,k} - \mathbf{X}'_{j,l}|^p \boldsymbol{\pi}_{i,j}^s \boldsymbol{\pi}_{k,l}^v$$

| Non-convex bilinear program: NP-Hard

| BCD procedure: alternates OT problems -> converges to a local minima [Konno 1976]

Algorithm 1 BCD for COOT

```
1:  $\boldsymbol{\pi}_{(0)}^s \leftarrow \mathbf{w}\mathbf{w}'^T, \boldsymbol{\pi}_{(0)}^v \leftarrow \mathbf{v}\mathbf{v}'^T, k \leftarrow 0$ 
2: while  $k < \text{maxIt}$  and  $err > 0$  do
3:    $\boldsymbol{\pi}_{(k)}^v \leftarrow OT(\mathbf{v}, \mathbf{v}', \mathbf{L}(\mathbf{X}, \mathbf{X}') \otimes \boldsymbol{\pi}_{(k-1)}^s) \sim O(n^3 \log(n))$  ( $p = 2$ )
4:    $\boldsymbol{\pi}_{(k)}^s \leftarrow OT(\mathbf{w}, \mathbf{w}', \mathbf{L}(\mathbf{X}, \mathbf{X}') \otimes \boldsymbol{\pi}_{(k-1)}^v) \sim O(d^3 \log(d))$  ( $p = 2$ )
5:    $err \leftarrow \|\boldsymbol{\pi}_{(k-1)}^v - \boldsymbol{\pi}_{(k)}^v\|_F$ 
6:    $k \leftarrow k + 1$ 
7: end while
```

CO-Optimal Transport

Solving COOT

CO-Optimal Transport

$$\min_{\substack{\pi^s \in \Pi(\mathbf{w}, \mathbf{w}') \\ \pi^v \in \Pi(\mathbf{v}, \mathbf{v}')}} \sum_{i,j,k,l} |\mathbf{X}_{i,k} - \mathbf{X}'_{j,l}|^p \pi_{i,j}^s \pi_{k,l}^v$$

Non-convex bilinear program: NP-Hard

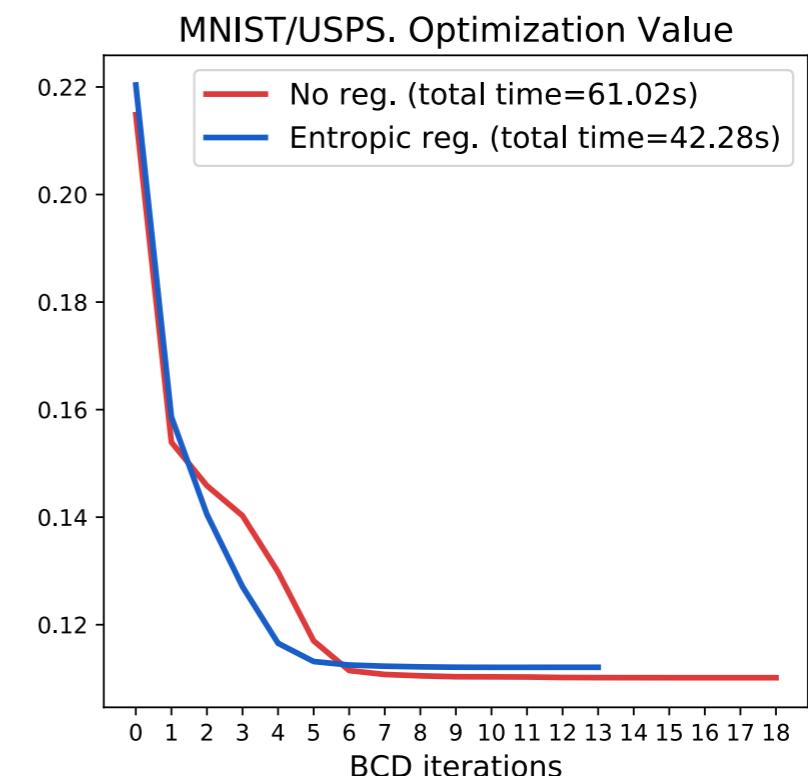
BCD procedure: alternates OT problems -> converges to a local minima [Konno 1976]

In practice BCD converges in few iterations

Algorithm 1 BCD for COOT

```

1:  $\pi_{(0)}^s \leftarrow \mathbf{w}\mathbf{w}'^T, \pi_{(0)}^v \leftarrow \mathbf{v}\mathbf{v}'^T, k \leftarrow 0$ 
2: while  $k < \text{maxIt}$  and  $err > 0$  do
3:    $\pi_{(k)}^v \leftarrow OT(\mathbf{v}, \mathbf{v}', \mathbf{L}(\mathbf{X}, \mathbf{X}') \otimes \pi_{(k-1)}^s)$ 
4:    $\pi_{(k)}^s \leftarrow OT(\mathbf{w}, \mathbf{w}', \mathbf{L}(\mathbf{X}, \mathbf{X}') \otimes \pi_{(k-1)}^v)$ 
5:    $err \leftarrow \|\pi_{(k-1)}^v - \pi_{(k)}^v\|_F$ 
6:    $k \leftarrow k + 1$ 
7: end while
```



CO-Optimal Transport

Domain adaptation in a nutshell

Given a source domain with labels

$$\mathbf{X}_s = \{\mathbf{x}_i^s\}_{i=1}^{N_s}$$

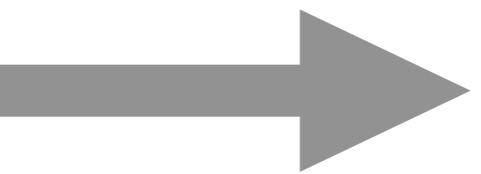
$$\mathbf{Y}_s = \{\mathbf{y}_i^s\}_{i=1}^{N_s}$$

CO-Optimal Transport

Domain adaptation in a nutshell

Given a source domain with labels

$$\mathbf{X}_s = \{\mathbf{x}_i^s\}_{i=1}^{N_s}$$
$$\mathbf{Y}_s = \{\mathbf{y}_i^s\}_{i=1}^{N_s}$$



A target domain

$$\mathbf{X}_t = \{\mathbf{x}_i^t\}_{i=1}^{N_t}$$

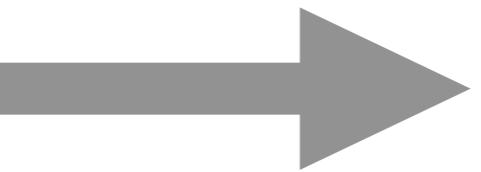
Apply/learn a classifier on

CO-Optimal Transport

Domain adaptation in a nutshell

Given a source domain with labels

$$\mathbf{X}_s = \{\mathbf{x}_i^s\}_{i=1}^{N_s}$$
$$\mathbf{Y}_s = \{\mathbf{y}_i^s\}_{i=1}^{N_s}$$



A target domain

$$\mathbf{X}_t = \{\mathbf{x}_i^t\}_{i=1}^{N_t}$$

Apply/learn a classifier on

related but
different domains..



CO-Optimal Transport

Domain adaptation in a nutshell

Given a source domain with labels

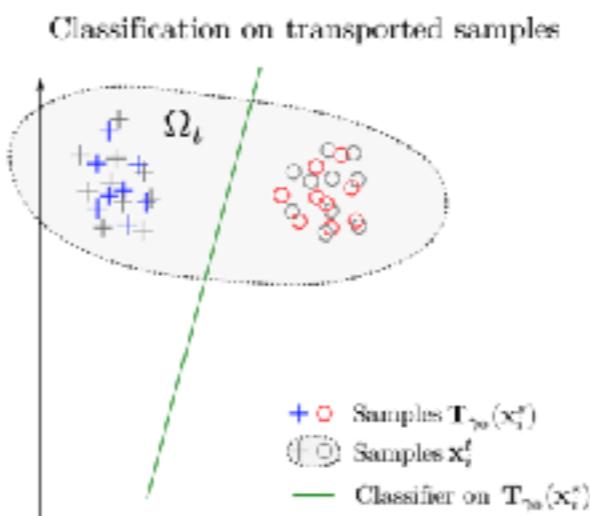
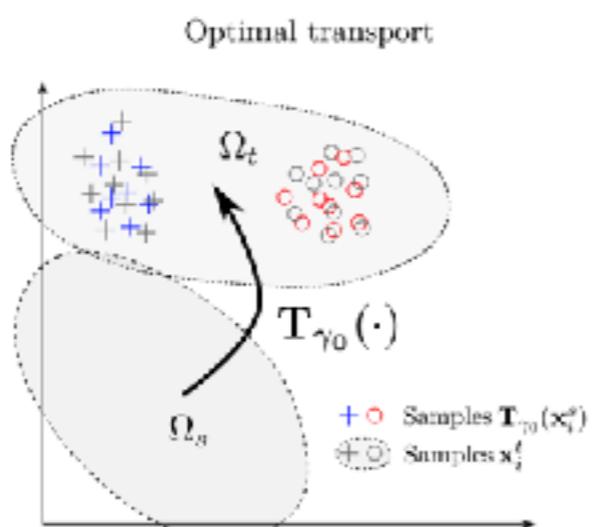
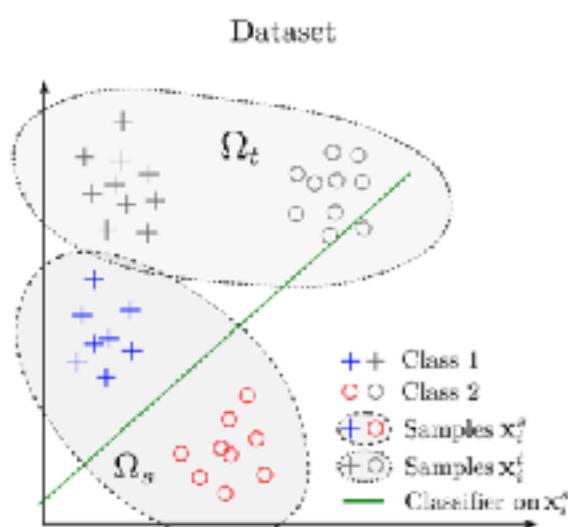
$$\mathbf{X}_s = \{\mathbf{x}_i^s\}_{i=1}^{N_s}$$
$$\mathbf{Y}_s = \{\mathbf{y}_i^s\}_{i=1}^{N_s}$$



A target domain

$$\mathbf{X}_t = \{\mathbf{x}_i^t\}_{i=1}^{N_t}$$

Apply/learn a classifier on



[Courty 2015]

$$\pi^s \leftarrow OT(\mathbf{X}_s, \mathbf{X}_t)$$

Barycentric mapping:

$$\hat{\mathbf{X}}_s = T_{\pi^s}(\mathbf{X}_s) = N_s \pi^s \mathbf{X}_t$$

[Redko 2019]

Label propagation:

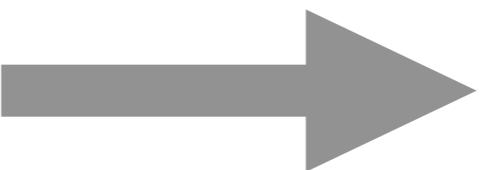
$$\hat{\mathbf{Y}}_t = \pi^s \mathbf{Y}_s$$

CO-Optimal Transport

Domain adaptation in a nutshell

Given a source domain with labels

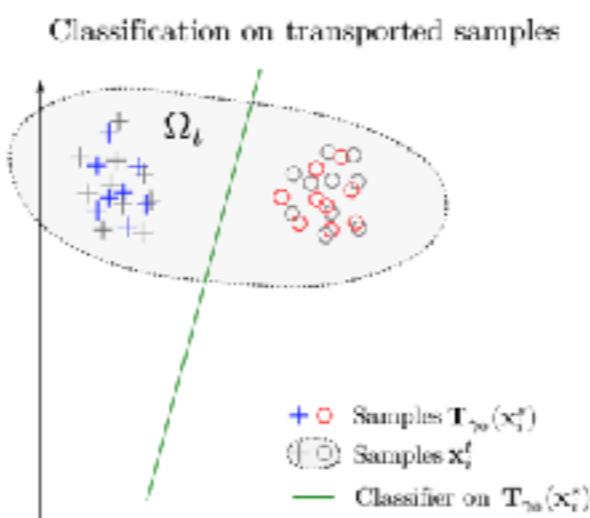
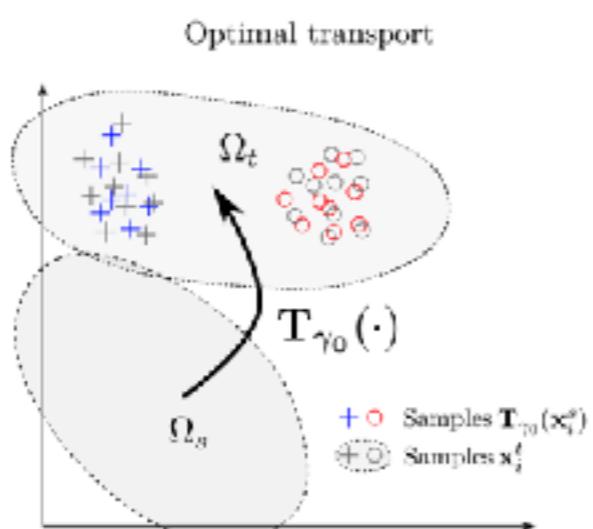
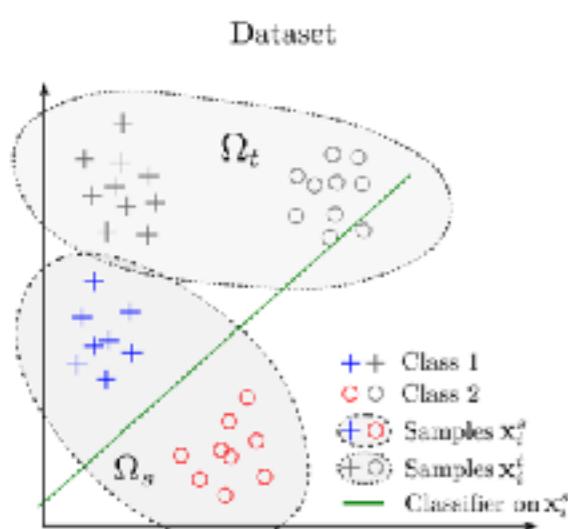
$$\mathbf{X}_s = \{\mathbf{x}_i^s\}_{i=1}^{N_s}$$
$$\mathbf{Y}_s = \{\mathbf{y}_i^s\}_{i=1}^{N_s}$$



A target domain

$$\mathbf{X}_t = \{\mathbf{x}_i^t\}_{i=1}^{N_t}$$

Apply/learn a classifier on



[Courty 2015]

$$\pi^s \leftarrow OT(\mathbf{X}_s, \mathbf{X}_t)$$

Barycentric mapping:

$$\hat{\mathbf{X}}_s = T_{\pi^s}(\mathbf{X}_s) = N_s \pi^s \mathbf{X}_t$$

[Redko 2019]

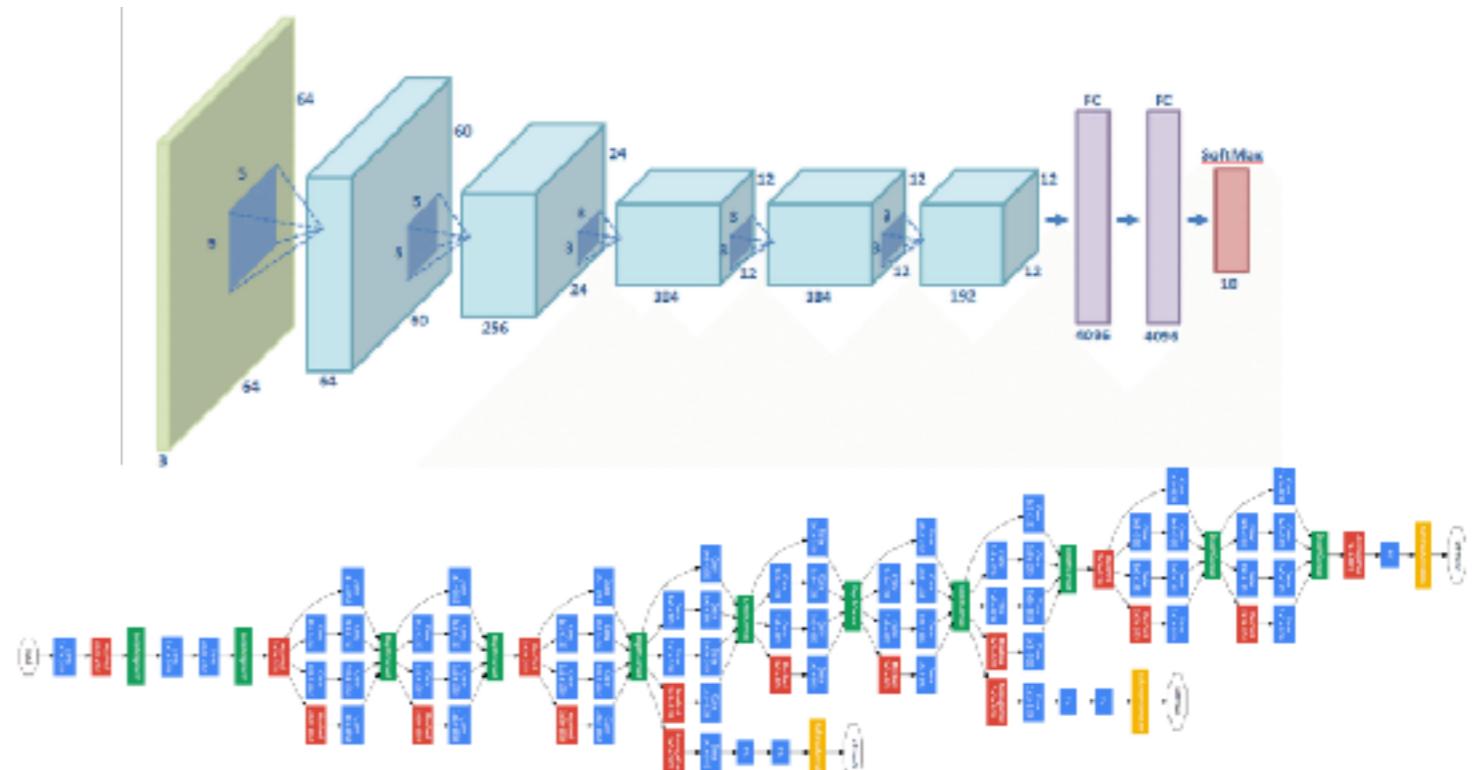
Label propagation:

$$\hat{\mathbf{Y}}_t = \pi^s \mathbf{Y}_s$$

(HDA) $\mathbf{X}_s \in \mathbb{R}^{N_s \times d}$ and $\mathbf{X}_t \in \mathbb{R}^{N_t \times d'}$ with $d \neq d'$

CO-Optimal Transport

COOT in action: Heterogeneous Domain Adaptation



Caltech/Office dataset [Saenko 2010]

$$\pi^s, \pi^v \leftarrow \text{COOT}(\mathbf{X}_s, \mathbf{X}_t)$$

Adaptation from two different embeddings from Decaf to GoogleNet $\mathbb{R}^{4096} \rightarrow \mathbb{R}^{1024}$

Unsupervised HDA + Semi supervised HDA (3 samples per class)

Label propagation [Redko 2019] $\hat{\mathbf{Y}}_t = \pi^s \mathbf{Y}_s$

CO-Optimal Transport

COOT in action: Heterogeneous Domain Adaptation

Domains	No-adaptation baseline	CCA	KCCA	EGW	SGW	COOT
Semi supervised HDA	C→W 69.12 ± 4.82	11.47 ± 3.78	66.76 ± 4.40	11.35 ± 1.93	<u>78.88 ± 3.90</u>	83.47 ± 2.60
	W→C 83.00 ± 3.95	19.59 ± 7.71	76.76 ± 4.70	11.00 ± 1.05	<u>92.41 ± 2.18</u>	93.65 ± 1.80
	W→W 82.18 ± 3.63	14.76 ± 3.15	78.94 ± 3.94	10.18 ± 1.64	<u>93.12 ± 3.14</u>	93.94 ± 1.84
	W→A 84.29 ± 3.35	17.00 ± 12.41	78.94 ± 6.13	7.24 ± 2.78	<u>93.41 ± 2.18</u>	94.71 ± 1.49
	A→C <u>83.71 ± 1.82</u>	15.29 ± 3.88	76.35 ± 4.07	9.82 ± 1.37	80.53 ± 6.80	89.53 ± 2.34
	A→W 81.88 ± 3.69	12.59 ± 2.92	81.41 ± 3.93	12.65 ± 1.21	<u>87.18 ± 5.23</u>	92.06 ± 1.73
	A→A <u>84.18 ± 3.45</u>	13.88 ± 2.88	80.65 ± 3.03	14.29 ± 4.23	82.76 ± 6.63	92.12 ± 1.79
	C→C 67.47 ± 3.72	13.59 ± 4.33	60.76 ± 4.38	11.71 ± 1.91	<u>77.59 ± 4.90</u>	83.35 ± 2.31
	C→A 66.18 ± 4.47	13.71 ± 6.15	63.35 ± 4.32	11.82 ± 2.58	<u>75.94 ± 5.58</u>	82.41 ± 2.79
	Mean	78.00 ± 7.43	14.65 ± 2.29	73.77 ± 7.47	11.12 ± 1.86	<u>84.65 ± 6.62</u>
p-value		<.001	<.001	<.001	<.001	-

Caltech/Officie dataset [Saenko 2010]

$$\pi^s, \pi^v \leftarrow \text{COOT}(\mathbf{X}_s, \mathbf{X}_t)$$

Adaptation from two different embeddings from Decaf to GoogleNet $\mathbb{R}^{4096} \rightarrow \mathbb{R}^{1024}$

Unsupervised HDA + Semi supervised HDA (3 samples per class)

Label propagation [Redko 2019] $\hat{\mathbf{Y}}_t = \pi^s \mathbf{Y}_s$

CO-Optimal Transport

COOT in action: Heterogeneous Domain Adaptation

Unsupervised
HDA

Domains	CCA	KCCA	EGW	COOT
C → W	14.20±8.60	<u>21.30±15.64</u>	10.55±1.97	25.50±11.76
	W → C	13.35±3.70	<u>18.60±9.44</u>	10.60±0.94
	W → W	10.95±2.36	<u>13.25±6.34</u>	10.25±2.26
	W → A	14.25±8.14	<u>23.00±22.95</u>	9.50±2.47
	A → C	11.40±3.23	<u>11.50±9.23</u>	11.35±1.38
	A → W	19.65±17.85	<u>28.35±26.13</u>	11.60±1.30
	A → A	11.75±1.82	<u>14.20±4.78</u>	13.10±2.35
	C → C	12.00±4.69	<u>14.95±6.79</u>	12.90±1.46
	C → A	15.35±6.30	<u>23.35±17.61</u>	12.95±2.63
Mean	13.66±2.55	<u>18.72±5.33</u>	11.42±1.24	33.28±7.61
p-value	<.001	<.001	<.001	-

Caltech/Officie dataset [Saenko 2010]

$$\pi^s, \pi^v \leftarrow \text{COOT}(\mathbf{X}_s, \mathbf{X}_t)$$

Adaptation from two different embeddings from Decaf to GoogleNet $\mathbb{R}^{4096} \rightarrow \mathbb{R}^{1024}$

Unsupervised HDA + Semi supervised HDA (3 samples per class)

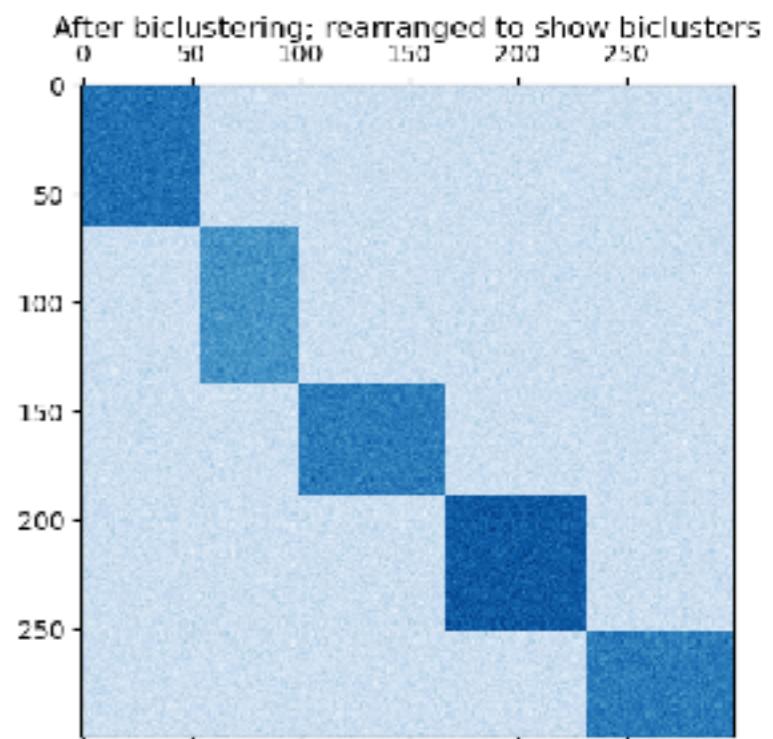
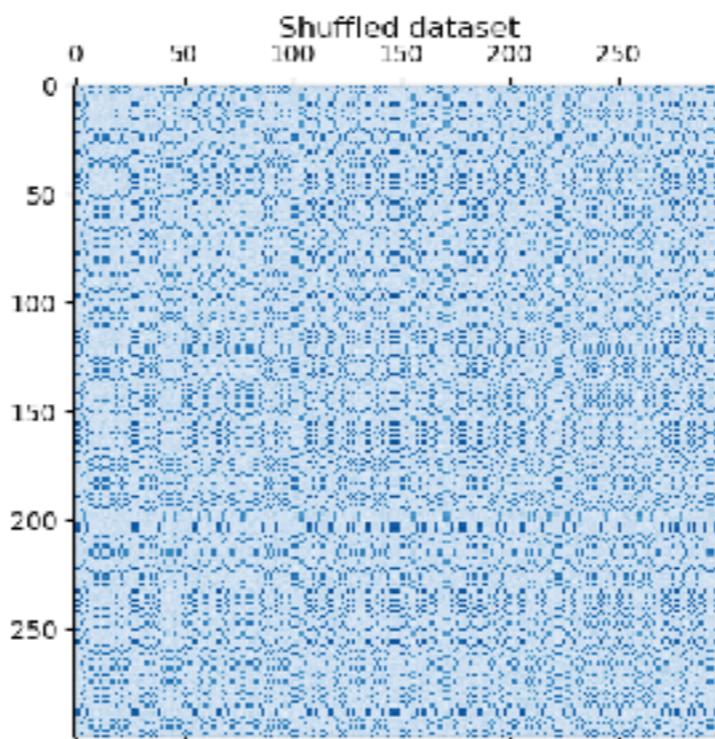
Label propagation [Redko 2019] $\hat{\mathbf{Y}}_t = \pi^s \mathbf{Y}_s$

CO-Optimal Transport

COOT in action: CO-clustering

Search for a simultaneous clustering of both samples and features of a dataset

$$\mathbf{X} \in \mathbb{R}^{n \times d}$$



CO-Optimal Transport

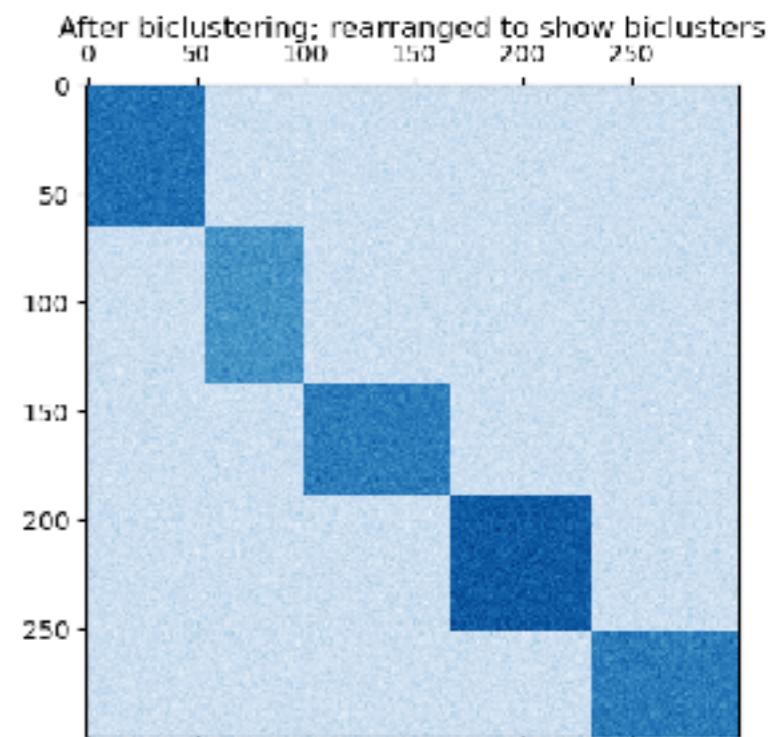
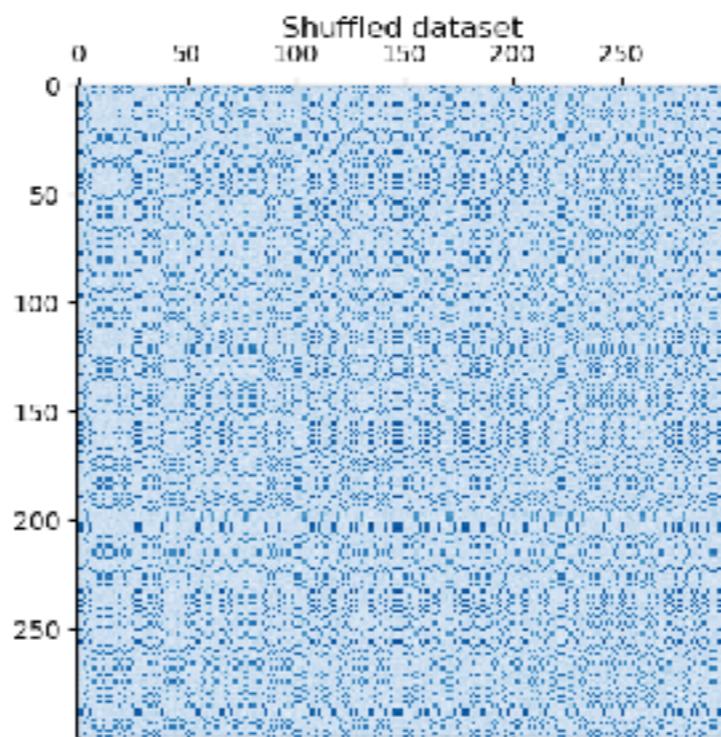
COOT in action: CO-clustering

Search for a simultaneous clustering of both samples and features of a dataset

$$\mathbf{X} \in \mathbb{R}^{n \times d}$$

COOT CO-clustering

$$\min_{\mathbf{X}_c \in \mathbb{R}^{n' \times d'}} \text{COOT}(\mathbf{X}, \mathbf{X}_c)$$



\mathbf{X}_c with $n' < n$, $d' < d$ that summarizes \mathbf{X} in the best way possible.

Solved by BCD

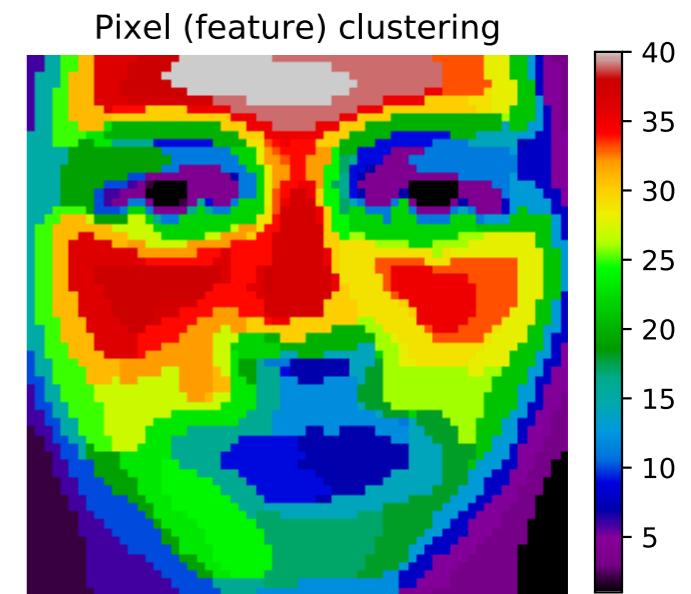
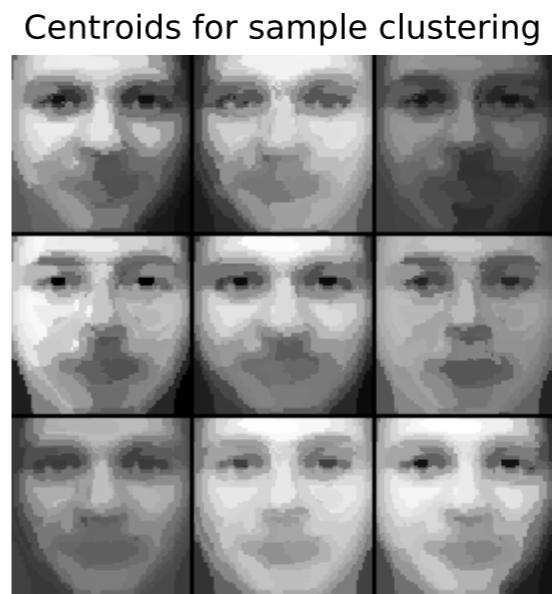
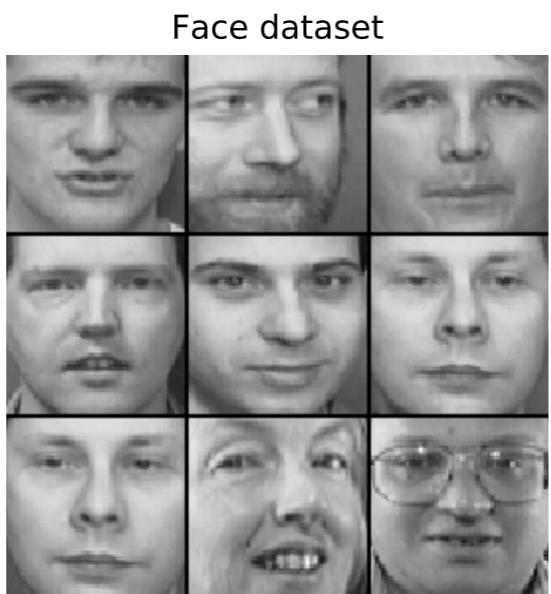
1. Obtain π^s and π^v by solving $\text{COOT}(\mathbf{X}, \mathbf{X}_c)$
2. Set \mathbf{X}_c to $n'd'\pi^{s\top}\mathbf{X}\pi^v$.

CO-Optimal Transport

COOT in action: CO-clustering

$$\min_{\mathbf{X}_c \in \mathbb{R}^{n' \times d'}} \text{COOT}(\mathbf{X}, \mathbf{X}_c)$$

Olivetti faces dataset
[Samaria 1994]

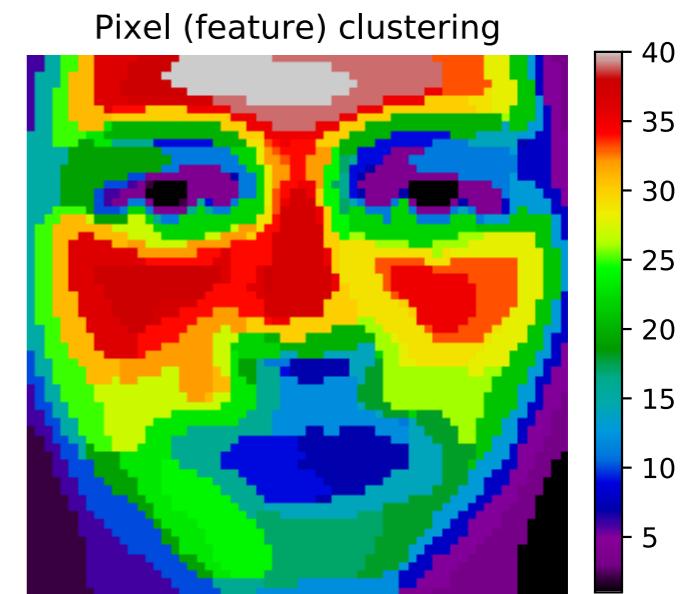
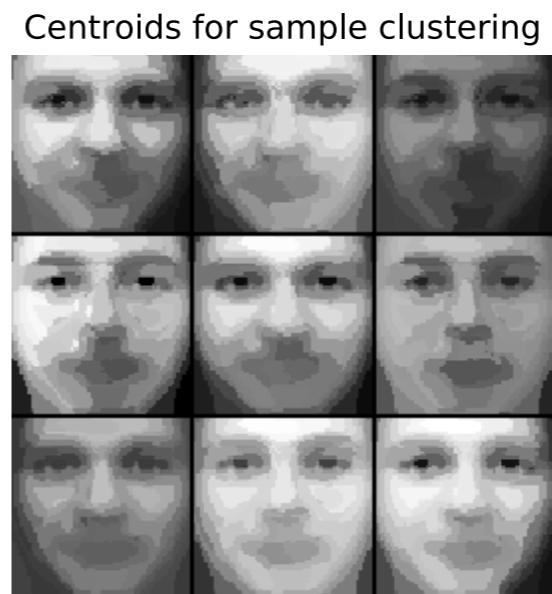
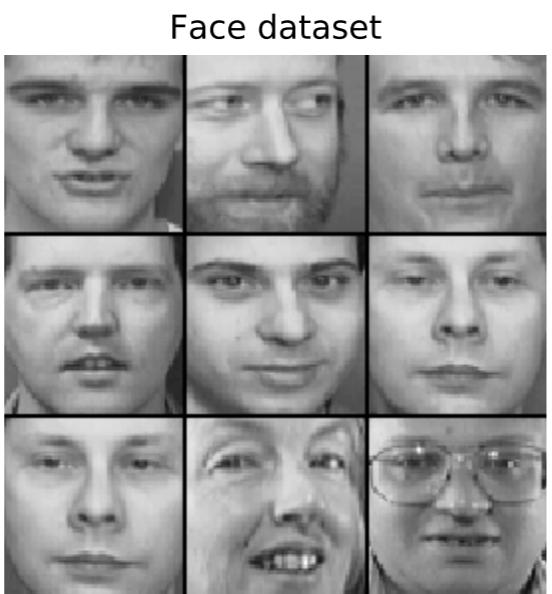


CO-Optimal Transport

COOT in action: CO-clustering

$$\min_{\mathbf{X}_c \in \mathbb{R}^{n' \times d'}} \text{COOT}(\mathbf{X}, \mathbf{X}_c)$$

Olivetti faces dataset
[Samaria 1994]



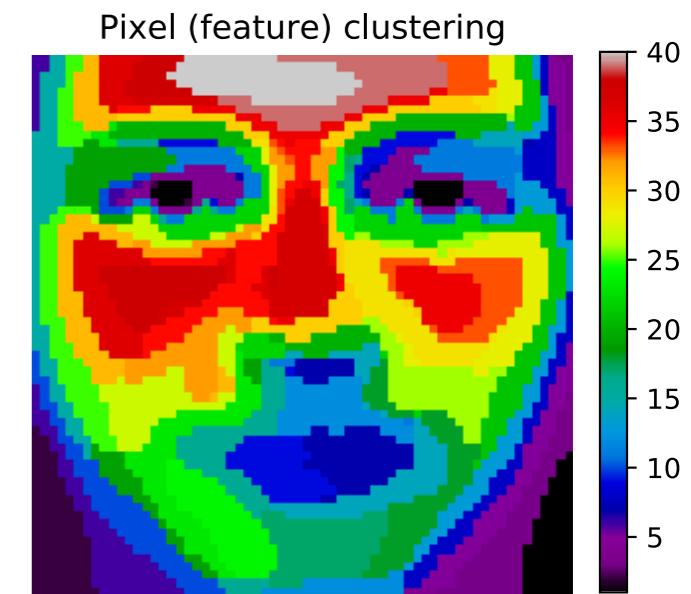
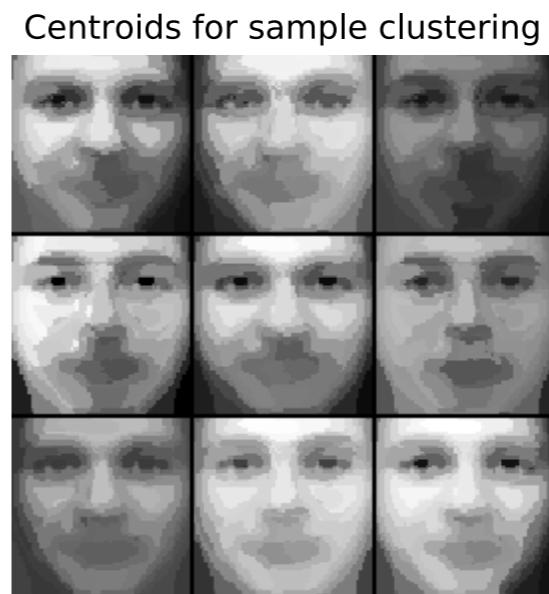
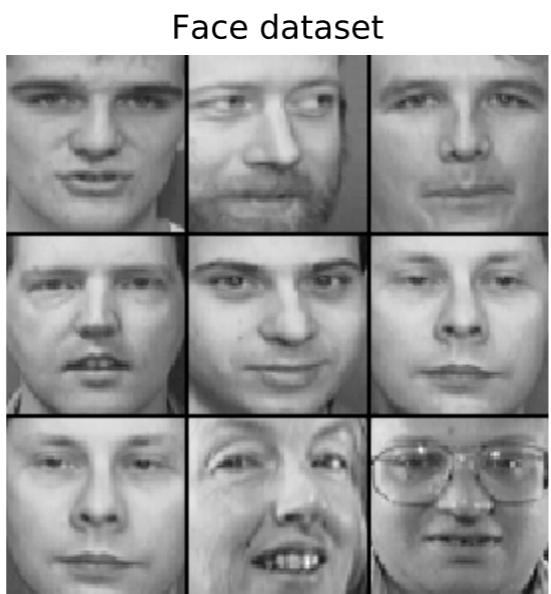
SOTA on simulated benchmark dataset from [Laclau 2017]

CO-Optimal Transport

COOT in action: CO-clustering

$$\min_{\mathbf{X}_c \in \mathbb{R}^{n' \times d'}} \text{COOT}(\mathbf{X}, \mathbf{X}_c)$$

Olivetti faces dataset
[Samaria 1994]



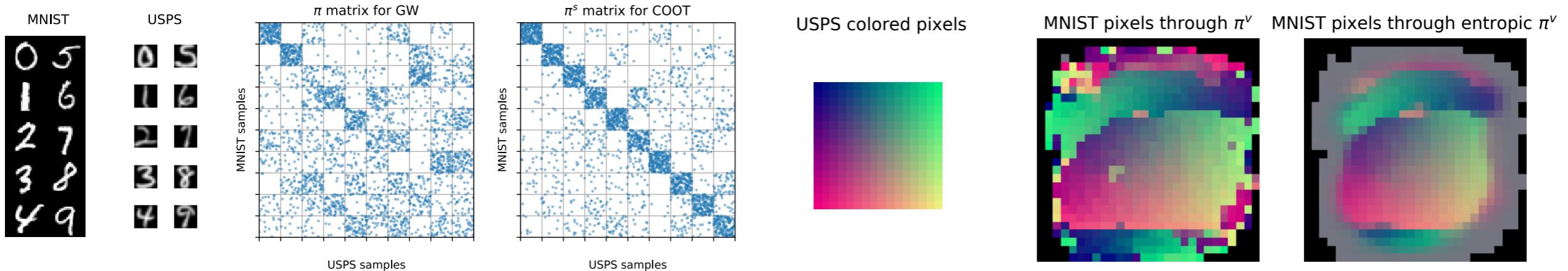
SOTA on simulated benchmark dataset from [Laclau 2017]

Movielens dataset (users and films)

M1	M20
Shawshank Redemption (1994)	Police Story 4: Project S (Chao ji ji hua) (1993)
Schindler's List (1993)	Eye of Vichy, The (Oeil de Vichy, L') (1993)
Casablanca (1942)	Promise, The (Versprechen, Das) (1994)
Rear Window (1954)	To Cross the Rubicon (1991)
Usual Suspects, The (1995)	Daens (1992)

CO-Optimal Transport

Conclusion: take away messages



COOT

- | OT method for heterogeneous dataset
- | Provides interpretable correspondences between samples and features
- | Works well for HDA + Can be applied for co-clustering

Perspectives

- | Study the statistics of COOT ($n, d \rightarrow \infty ?$)
- | Other formulations (unbalanced, extension to labeled dataset)
- | Effect of the entropic regularization (convergence), effect of the feature weights)

Thank you!

