(Practical and Computational) introduction to Optimal Transport

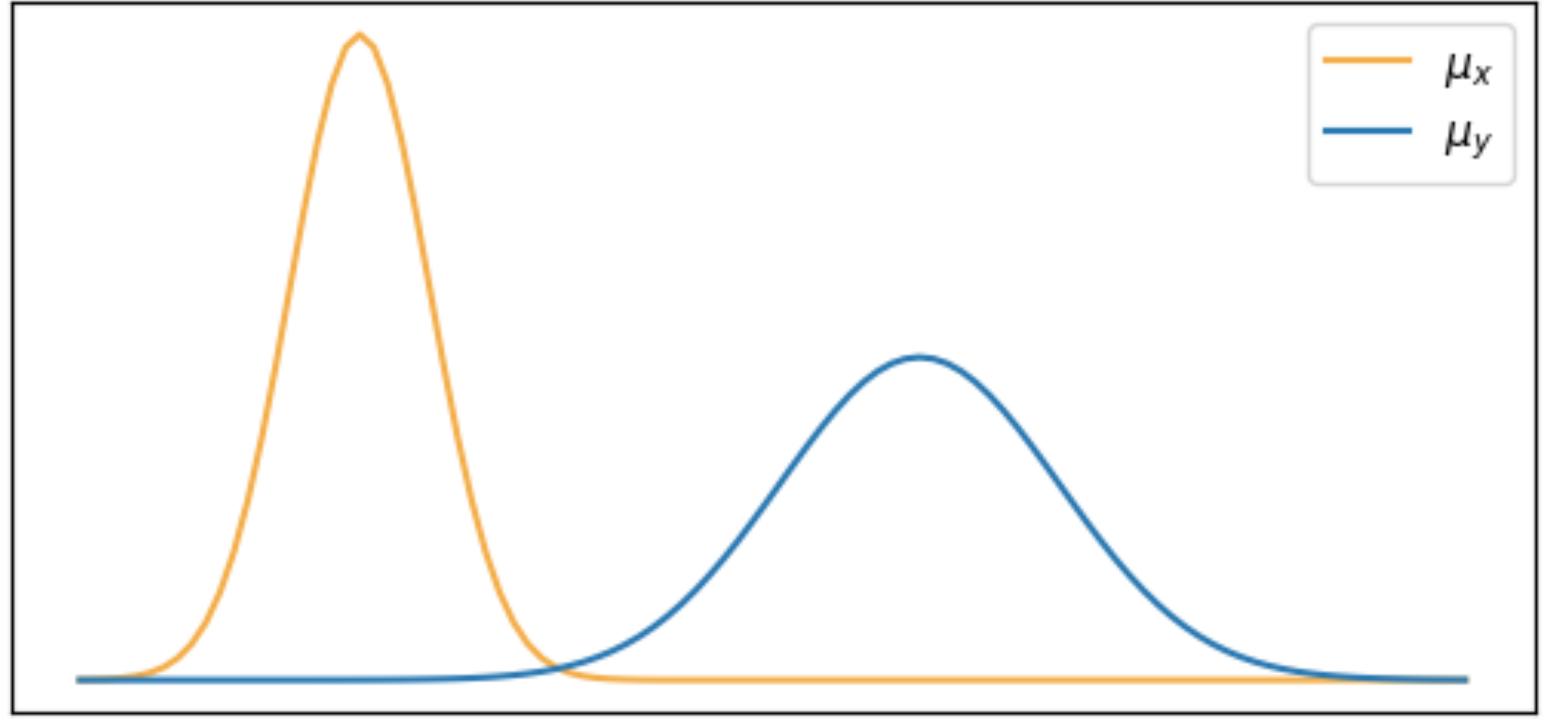
Laetitia Chapel (IRISA, Obelix team - Institut Agro Rennes-Angers)

Summer School Geometry and Data - Strasbourg - August 2023

Why optimal transport?

Need for a « meaningful » measure of distance between probability measures

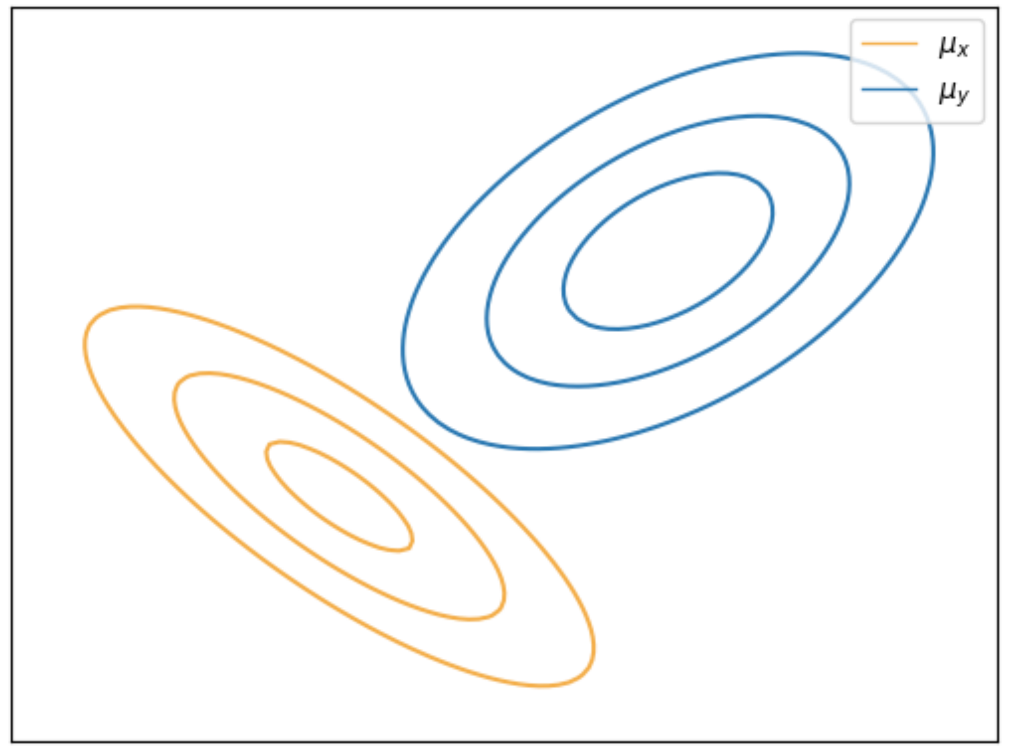
Continuous probability distributions



μ_x and μ_y $\in \mathscr{P}(\mathbb{R})$

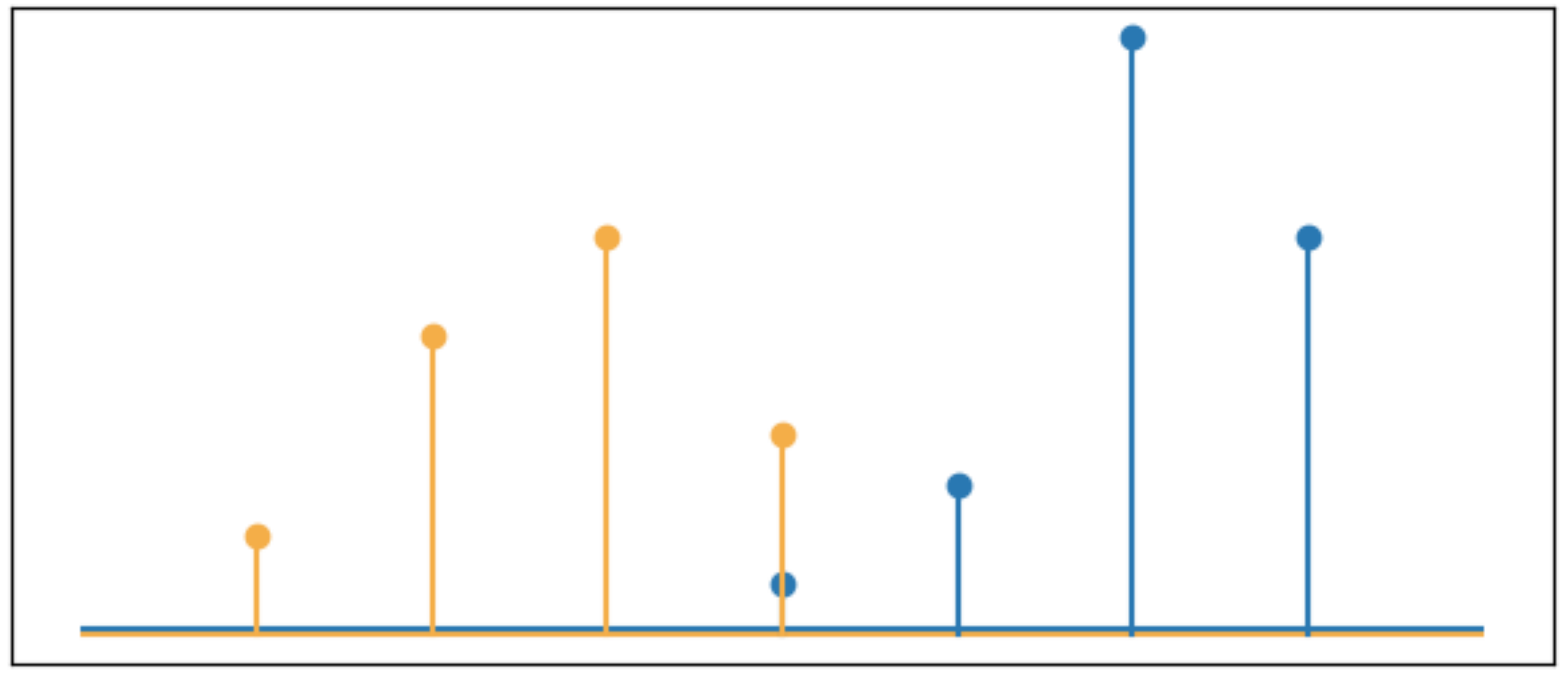
$$\mu_x(S) = \int_S \rho_x(x)dx$$

with ρ_x assigning a proba density to every point



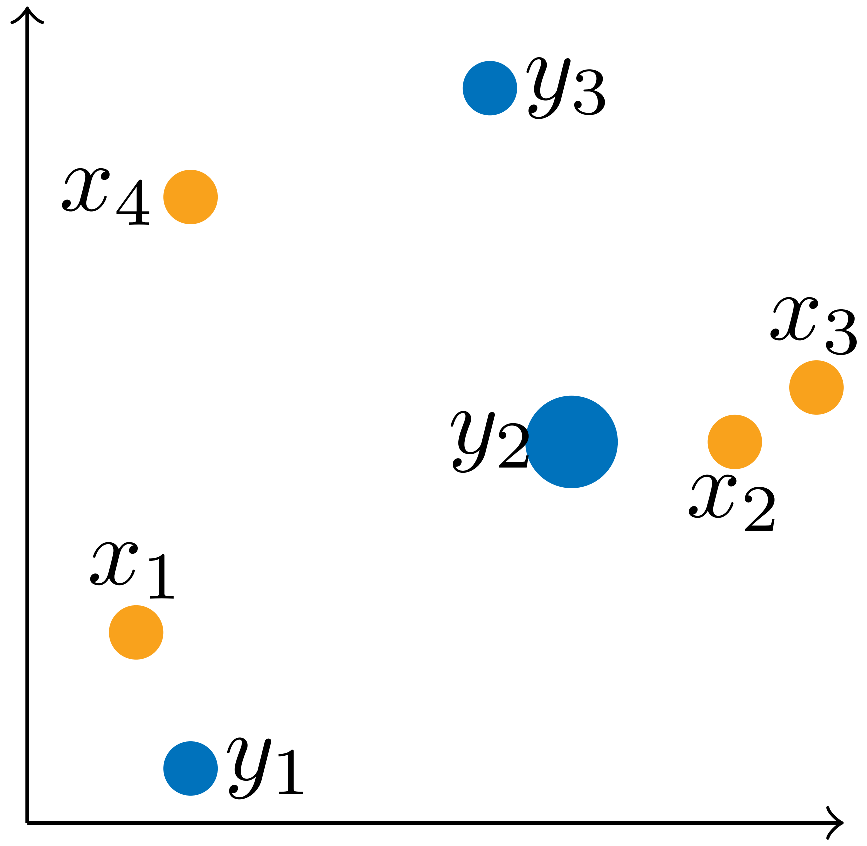
2d densities

Discrete distributions



$$\mu_X = \sum_{i=1}^{n} h_i \delta_{\boldsymbol{x}_i}$$

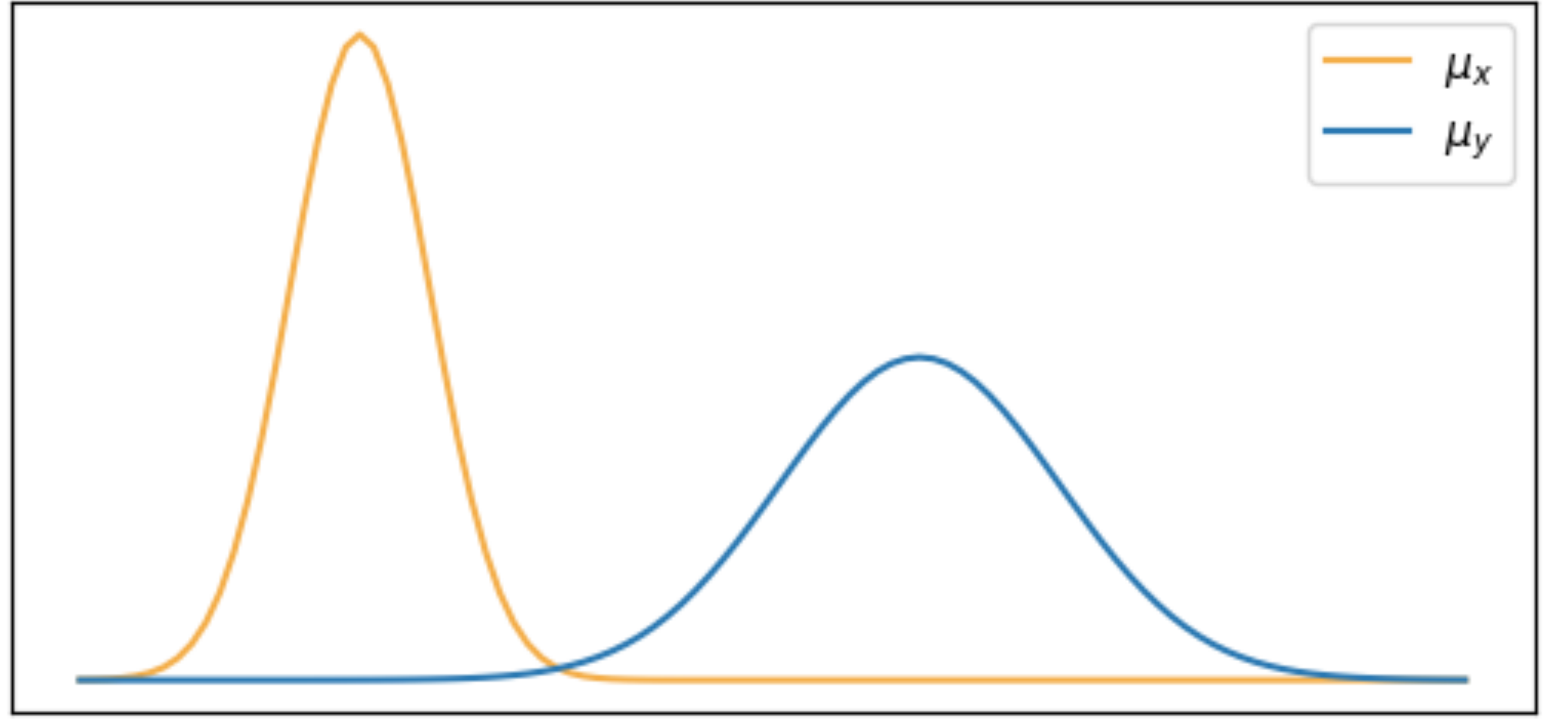
$$\mu_Y = \sum_{j=1}^{m} g_j \delta_{\boldsymbol{y}_j}$$



Why optimal transport?

Need for a « meaningful » measure of distance between probability measures

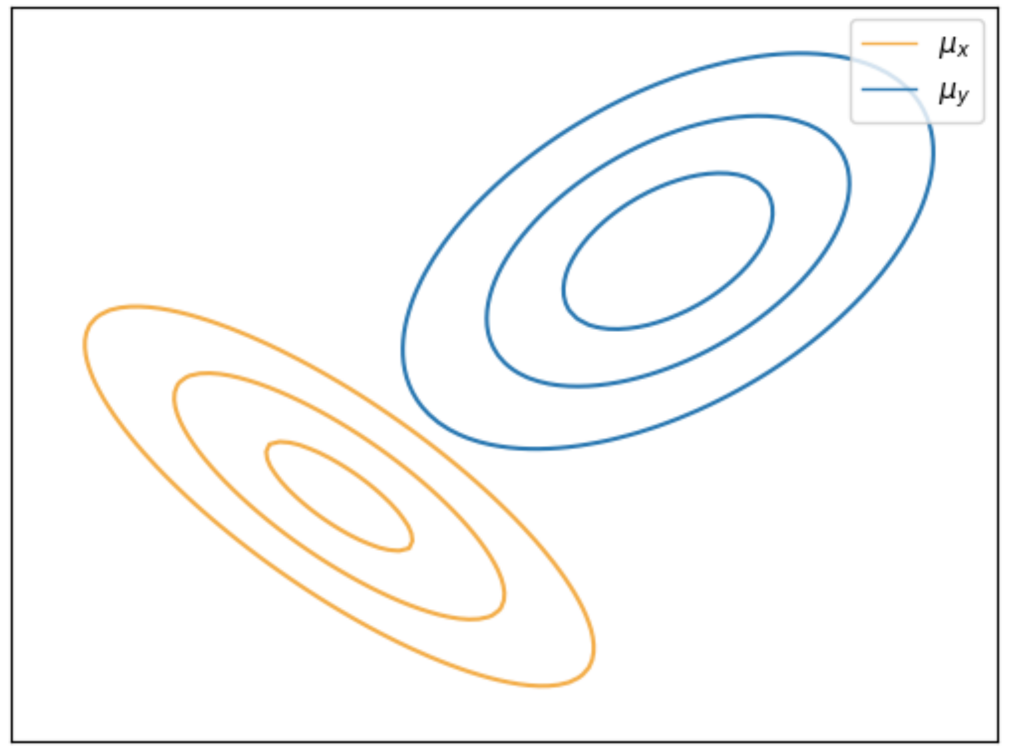
Continuous probability distributions



μ_x and $\mu_y \in \mathscr{P}(\mathbb{R})$

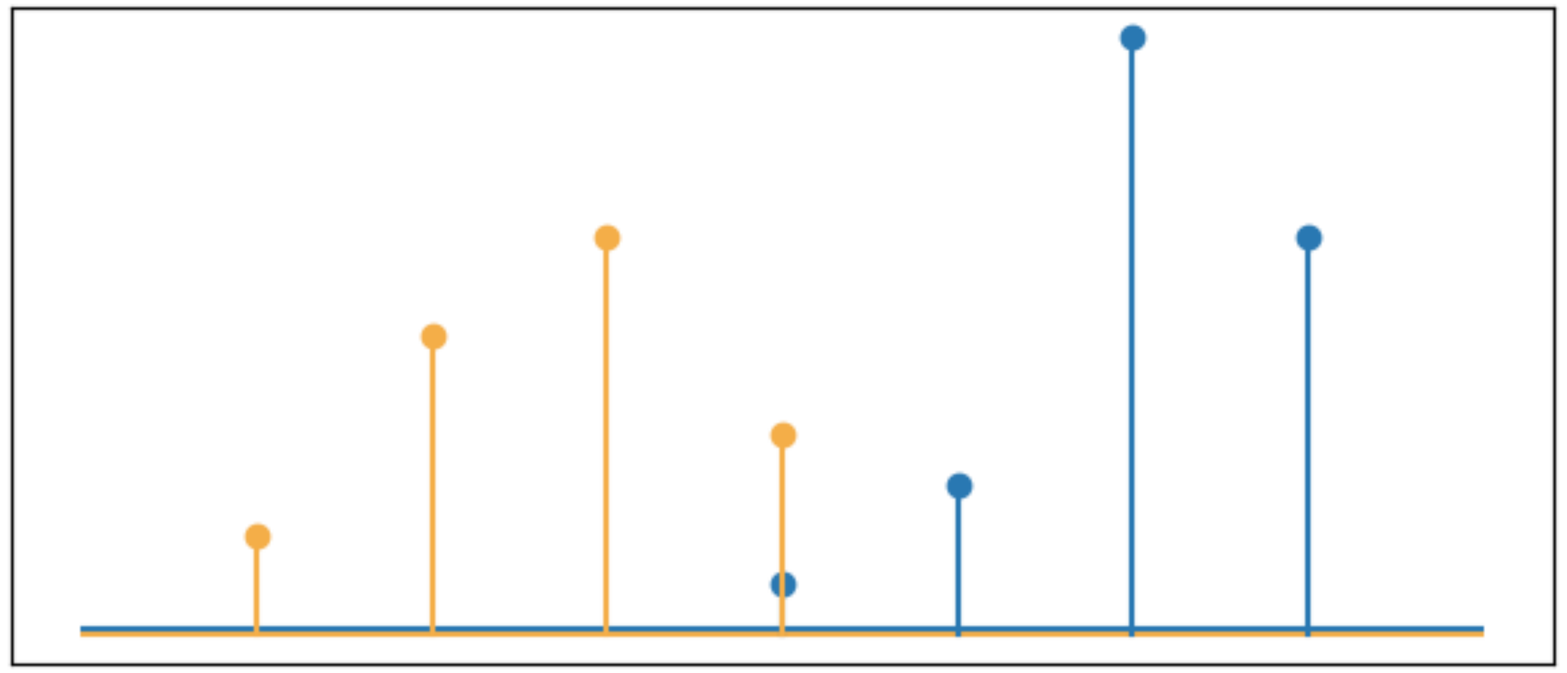
$$\mu_x(S) = \int_S \rho_x(x)dx$$

with ρ_x assigning a proba density to every point



2d densities

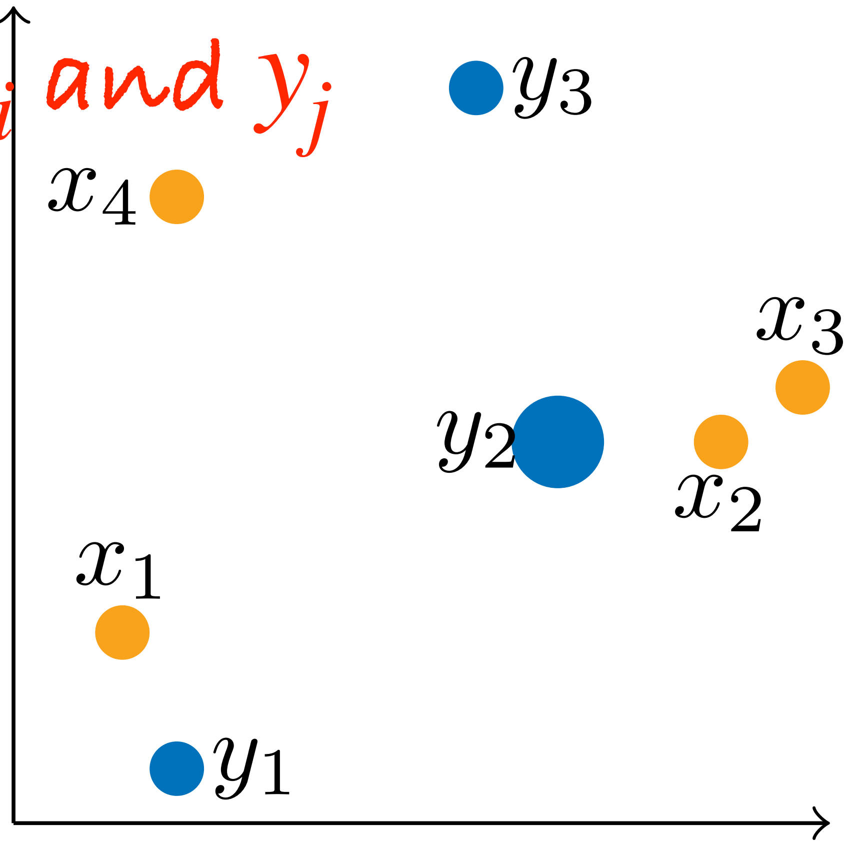
Discrete distributions



dirac at location x_i and y_j

$$\mu_X = \sum_{i=1}^{n} h_i \delta_{\boldsymbol{x}_i}$$

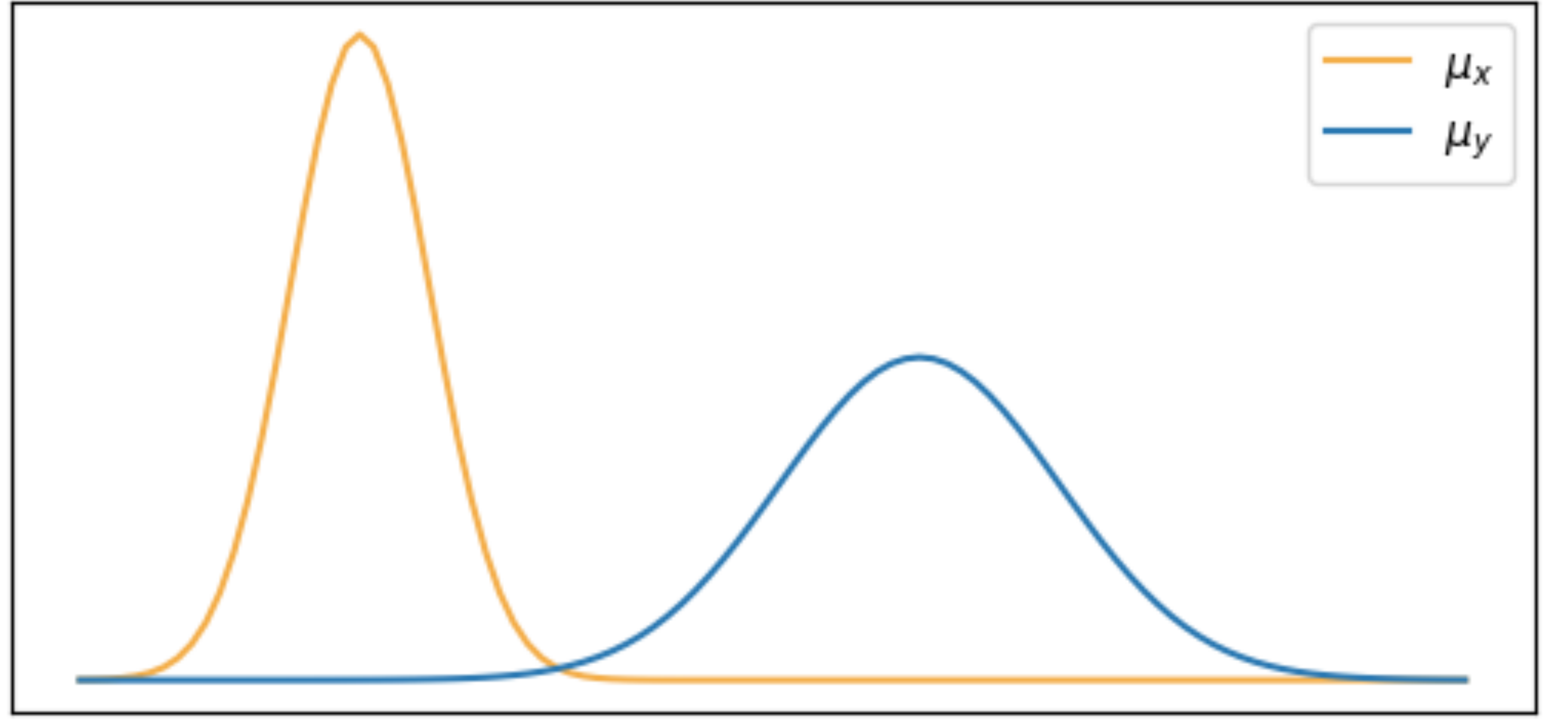
$$\mu_Y = \sum_{j=1}^{m} g_j \delta_{\boldsymbol{y}_j}$$



Why optimal transport?

Need for a « meaningful » measure of distance between probability measures

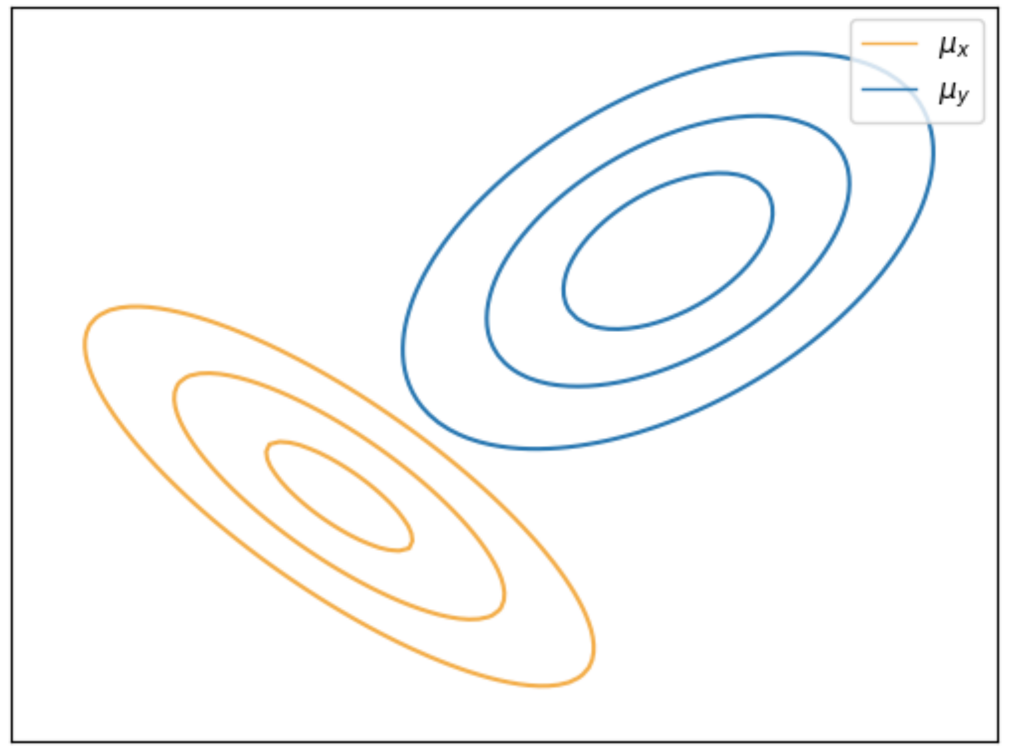
Continuous probability distributions



μ_x and μ_y $\in \mathscr{P}(\mathbb{R})$

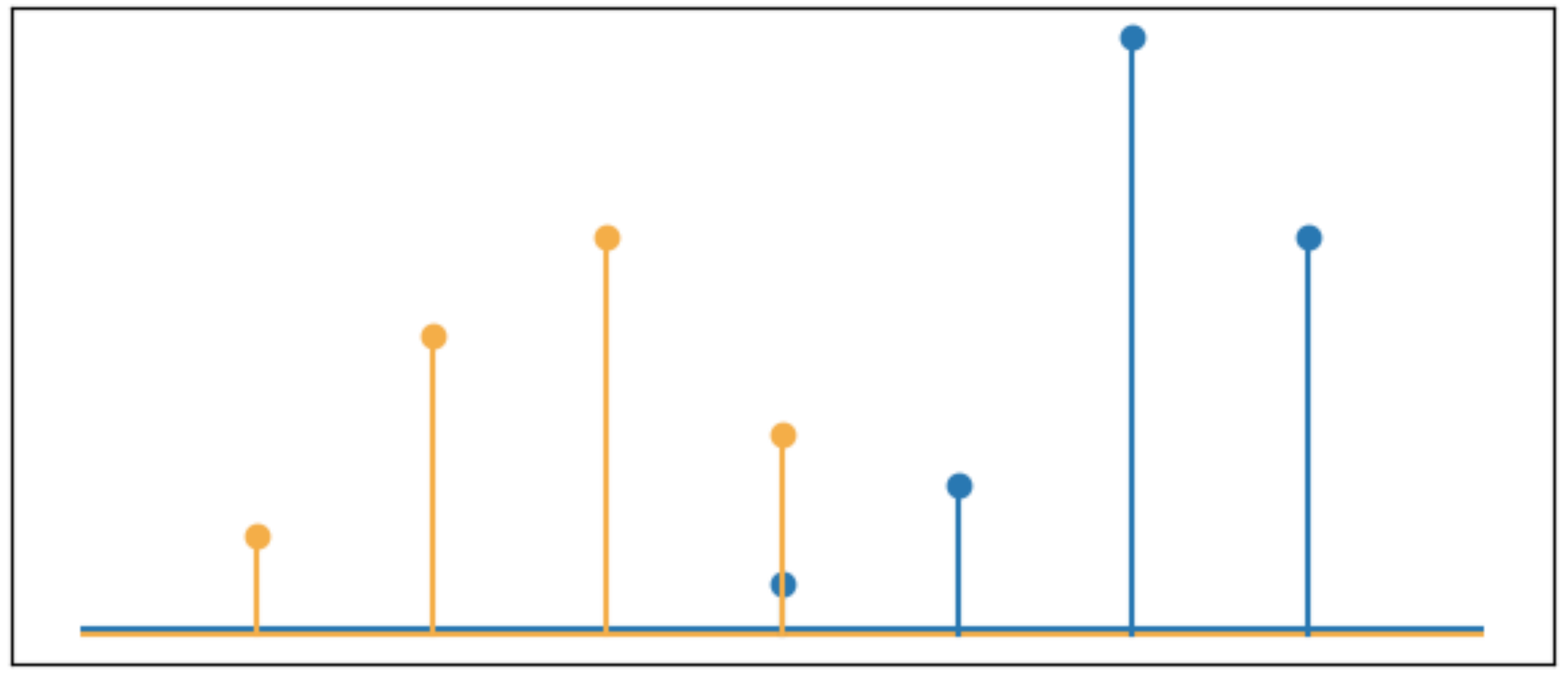
$$\mu_x(S) = \int_S \rho_x(x)dx$$

with ρ_x assigning a proba density to every point



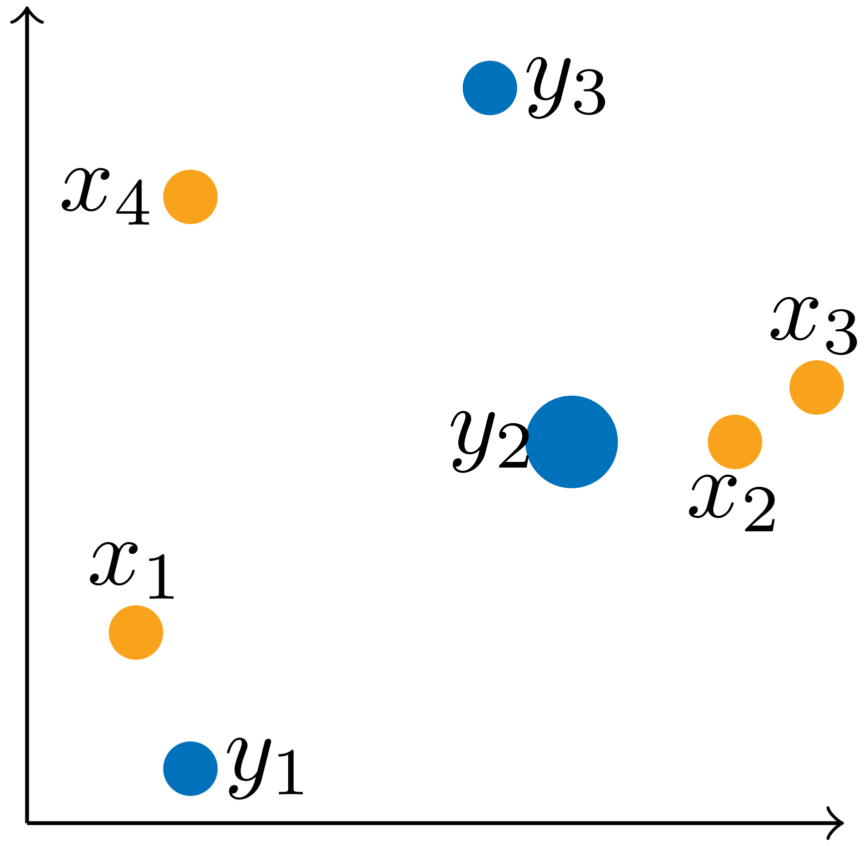
2d densities

Discrete distributions



$$\mu_X = \sum_{i=1}^{n} h_i \delta_{\boldsymbol{x}_i}$$

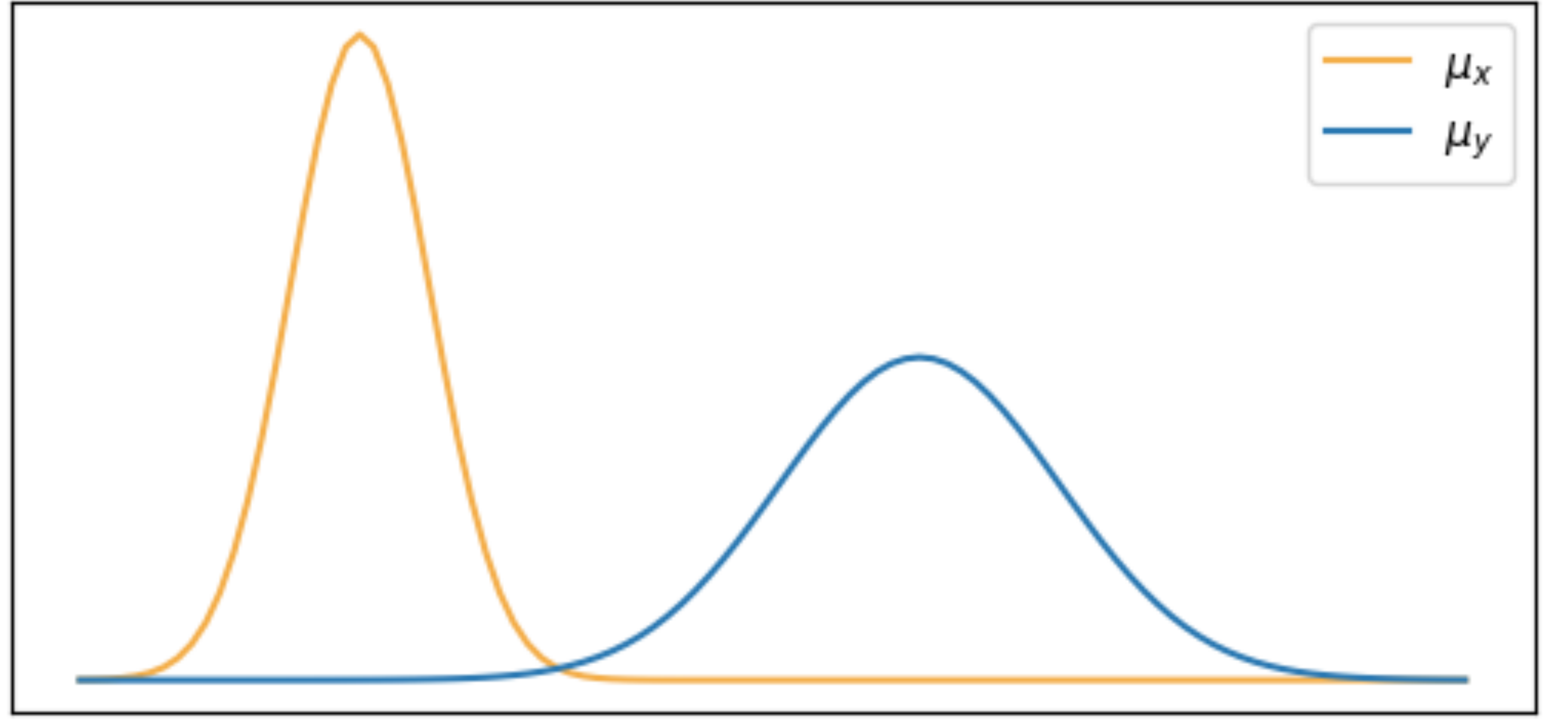
$$\mu_Y = \sum_{j=1}^{m} g_j \delta_{\boldsymbol{y}_j}$$



Why optimal transport?

Need for a « meaningful » measure of distance between probability measures

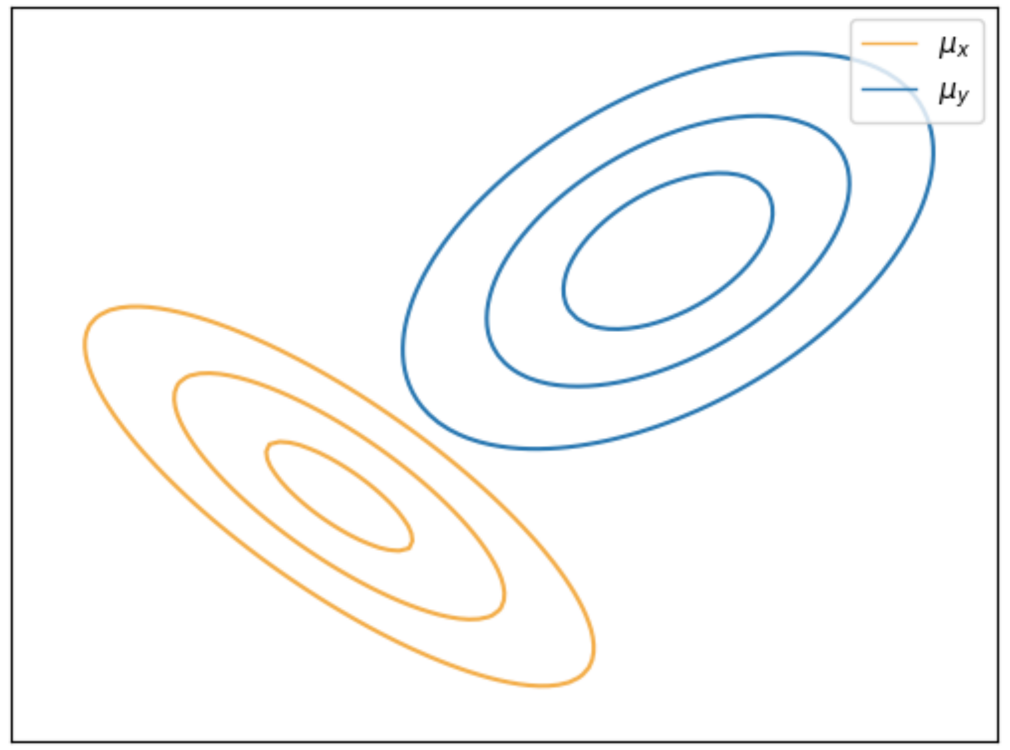
Continuous probability distributions



μ_x and μ_y $\in \mathscr{P}(\mathbb{R})$

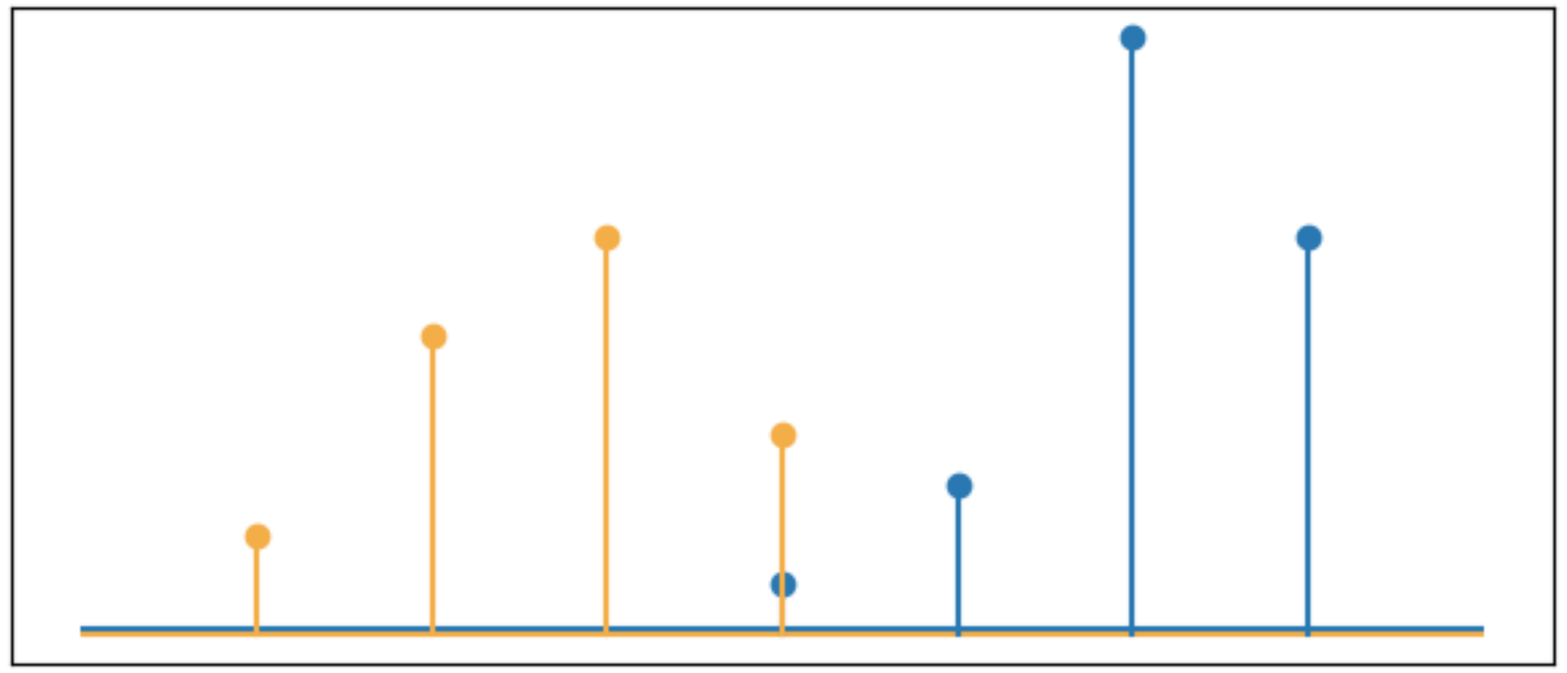
$$\mu_x(S) = \int_S \rho_x(x)dx$$

with ρ_x assigning a proba density to every point



2d densities

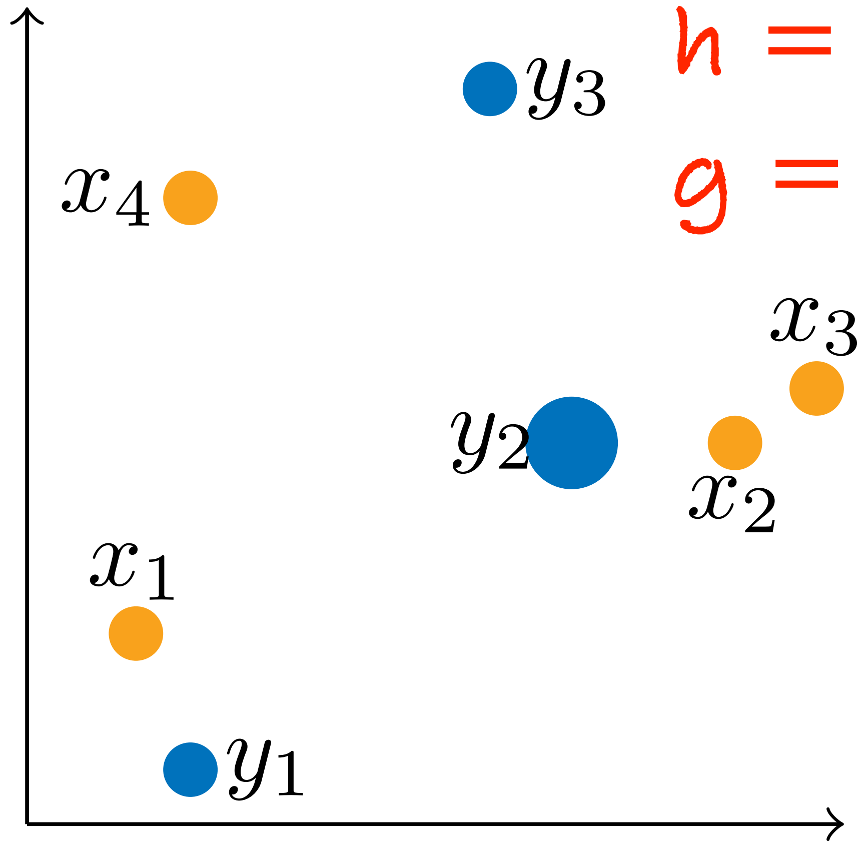
Discrete distributions



weights, masses

$$\mu_X = \sum_{i=1}^{n} h_i \delta_{\boldsymbol{x}_i}$$

$$\mu_Y = \sum_{j=1}^{m} g_j \delta_{\boldsymbol{y}_j}$$

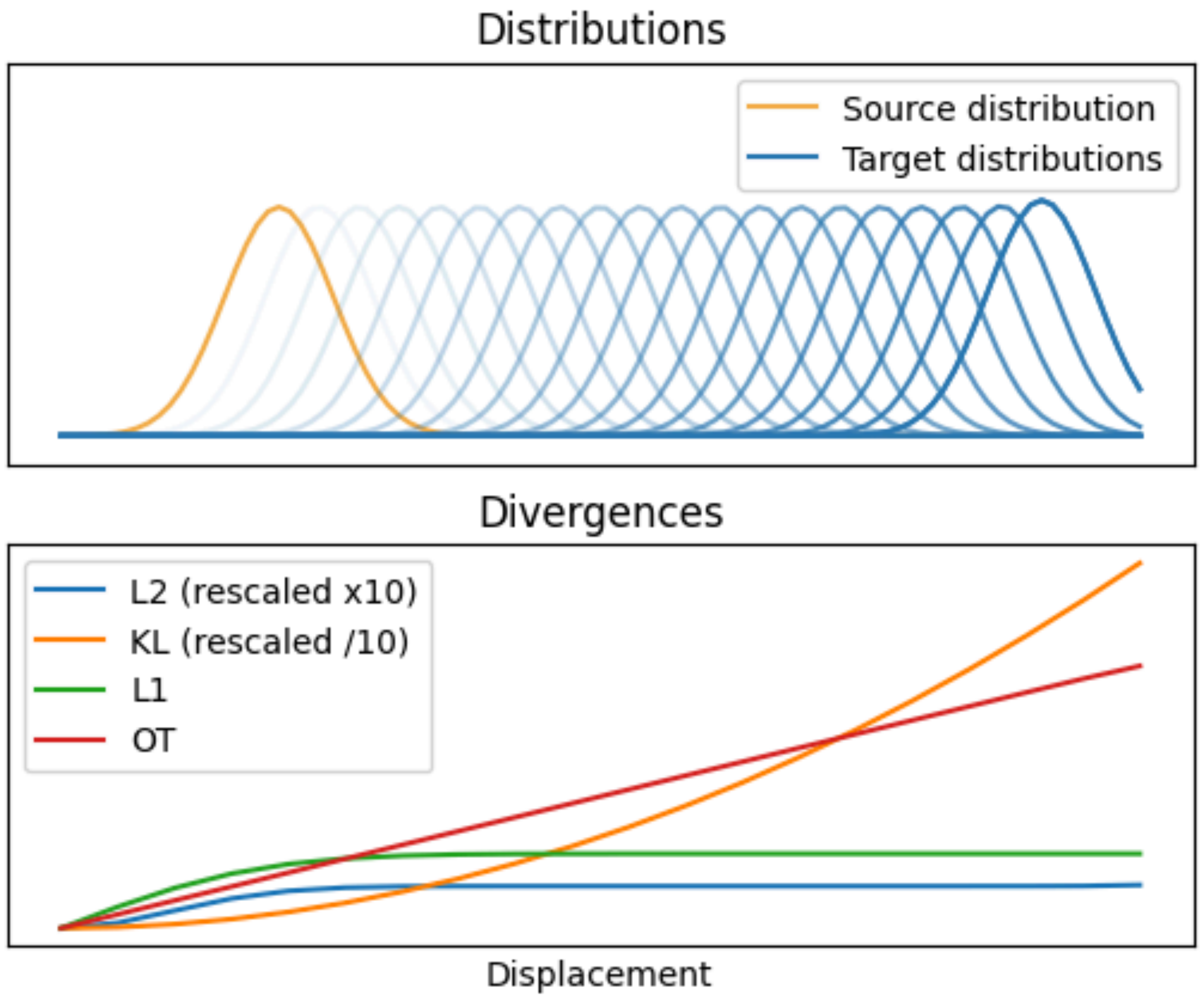


h = [1,1,1,1]/4

g = [1,2,1]/4

Why optimal transport?

Need for a « meaningful » measure of distance between probability measures



source distribution function ρ_x

target distribution function ρ_y

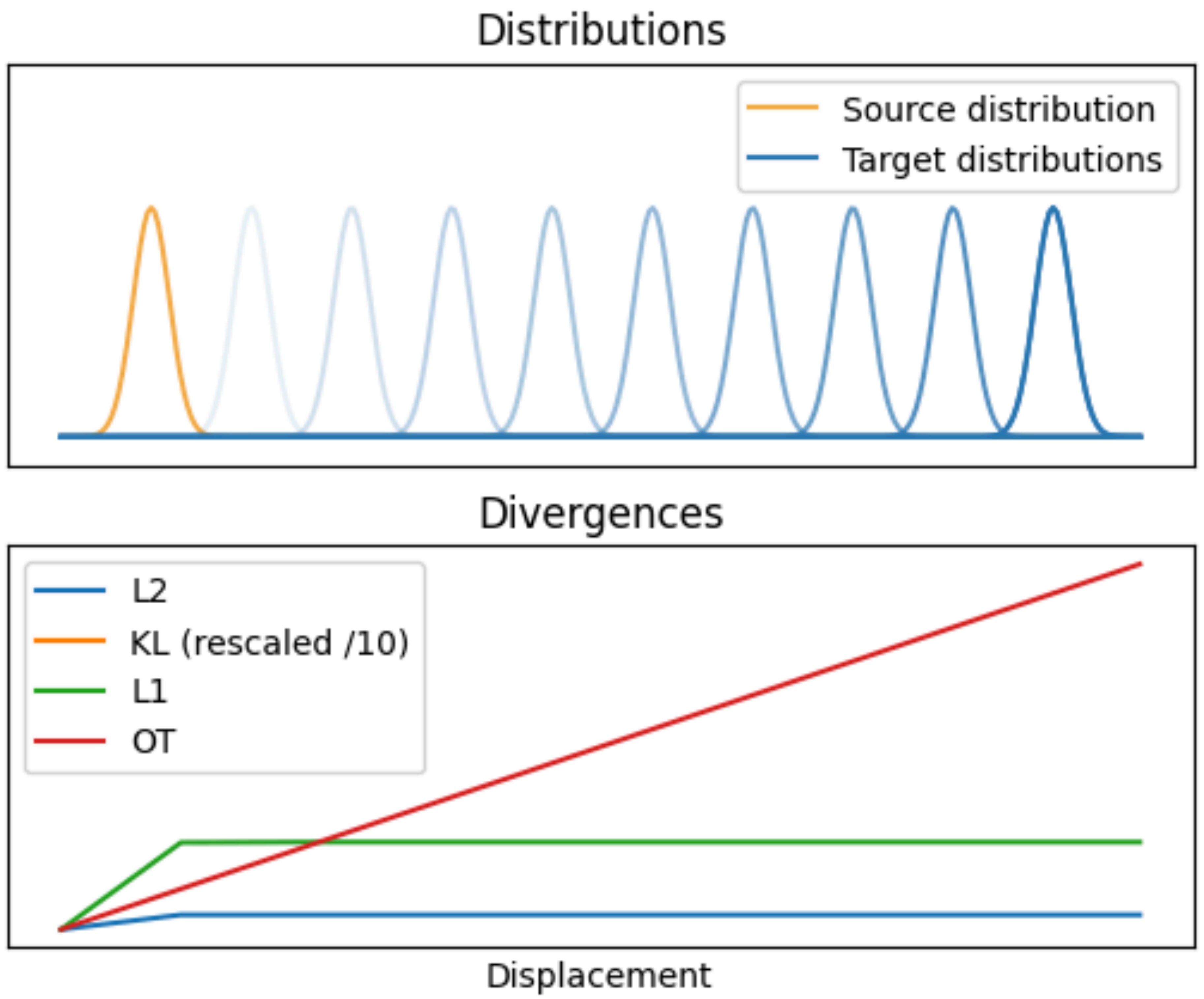
$$d_{L_1}(\rho_x, \rho_y) = \int_{\mathbb{R}} |\rho_x(x) - \rho_y(x)| dx$$

$$d_{L_2}(\rho_x, \rho_y) = \int_{\mathbb{R}} \|\rho_x(x) - \rho_y(x)\|_2 dx$$

$$d_{KL}(\rho_x, \rho_y) = \int_{\mathbb{R}} \rho_x(x) \log\left(\frac{\rho_x(x)}{\rho_y(x)}\right) dx$$

Why optimal transport?

Need for a « meaningful » measure of distance between probability measures



source distribution function ρ_x
target distribution function ρ_y

$$d_{L_1}(\rho_x, \rho_y) = \int_{\mathbb{R}} |\rho_x(x) - \rho_y(x)| dx$$

$$d_{L_2}(\rho_x, \rho_y) = \int_{\mathbb{R}} \|\rho_x(x) - \rho_y(x)\|_2 dx$$

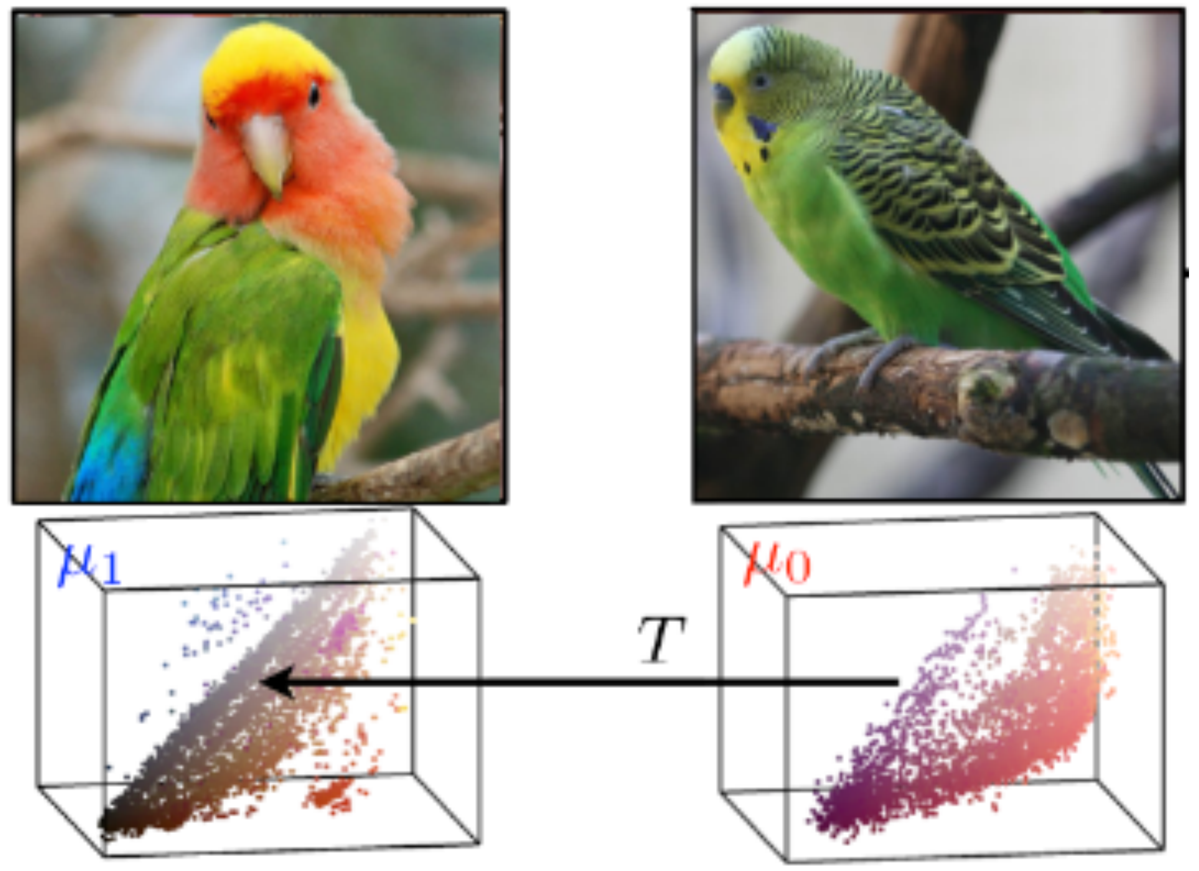
$$d_{KL}(\rho_x, \rho_y) = \int_{\mathbb{R}} \rho_x(x) \log \left(\frac{\rho_x(x)}{\rho_y(x)} \right) dx$$

Why optimal transport?

Probability measures are ubiquitous in data science

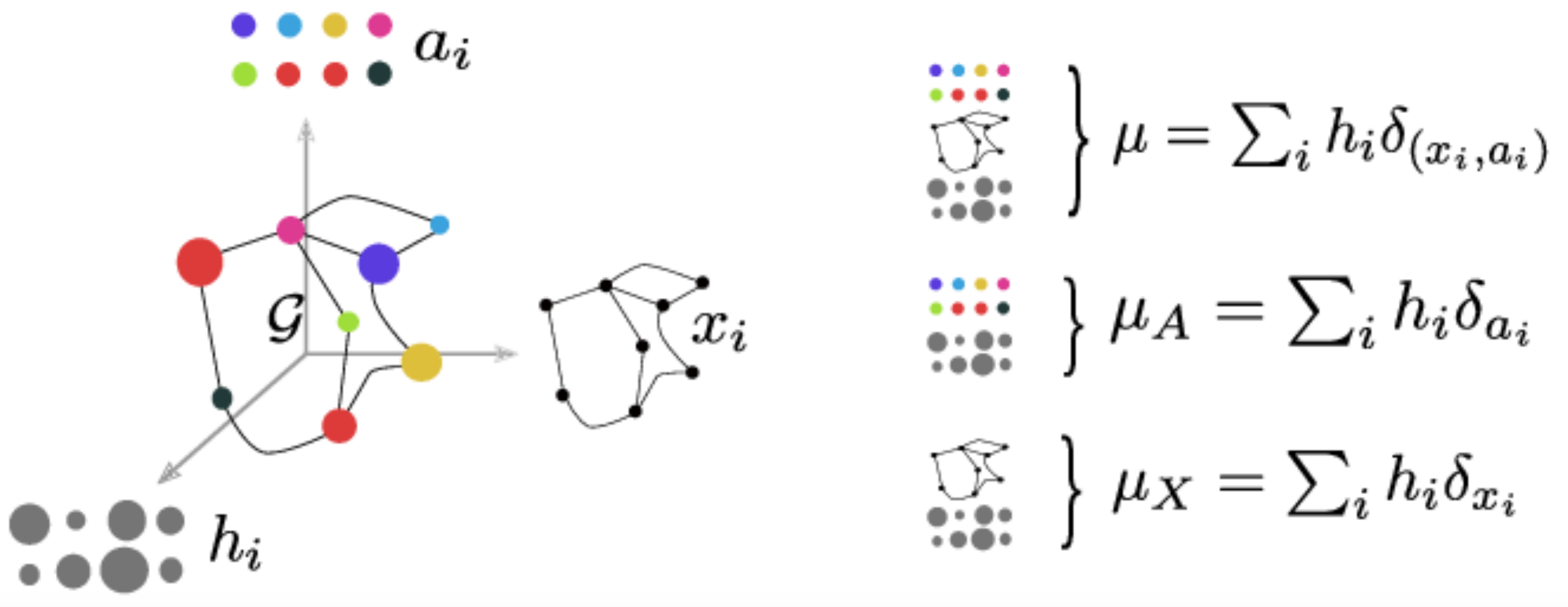
Images

@N. Papadakis (HDR)

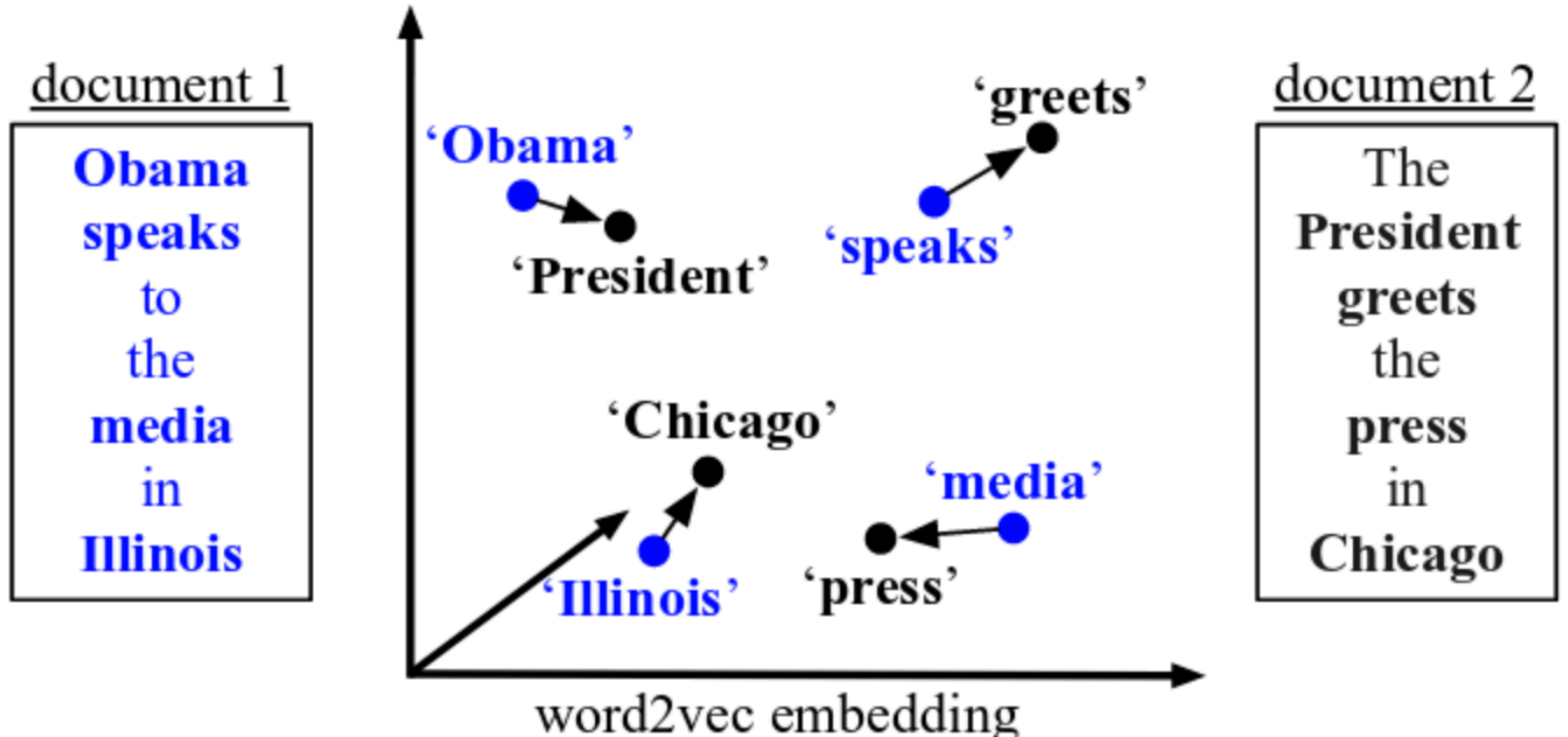


Graphs

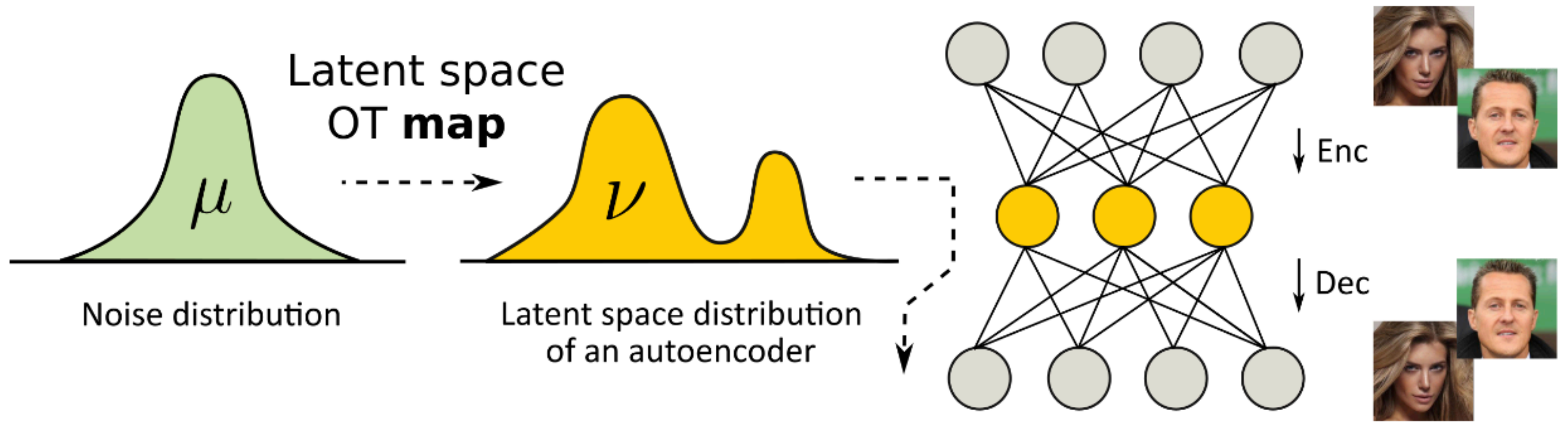
@T. Vayer (PhD thesis)


a_i
$\mathcal{G}$
x_i
h_i
$\mu = \sum_i h_i \delta_{(x_i, a_i)}$
$\mu_A = \sum_i h_i \delta_{a_i}$
$\mu_X = \sum_i h_i \delta_{x_i}$

Bag of features [Kusner 2015]



Generative models [Rout 2022]



Optimal Transport

Lots of applications!

Wasserstein style transfer

[Mroueh, 2020]

OT as a loss for classification

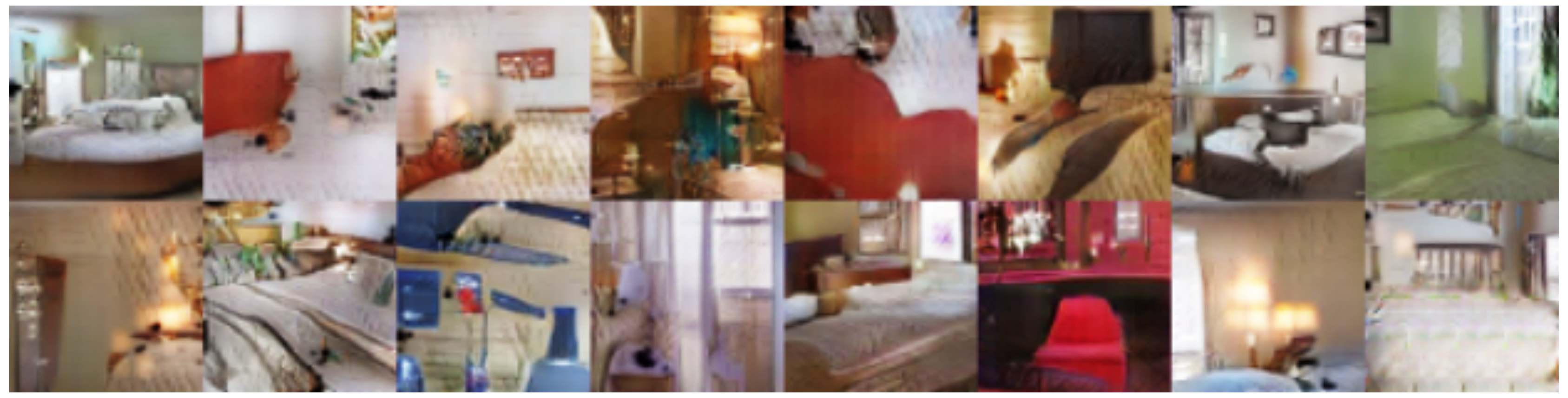
[Frogner, 2015]

Siberian husky

Eskimo dog

Flickr : street, parade, dragon
Prediction : people, protest, parade

Flickr : water, boat, ref ection, sun-shine
Prediction : water, river, lake, summer;

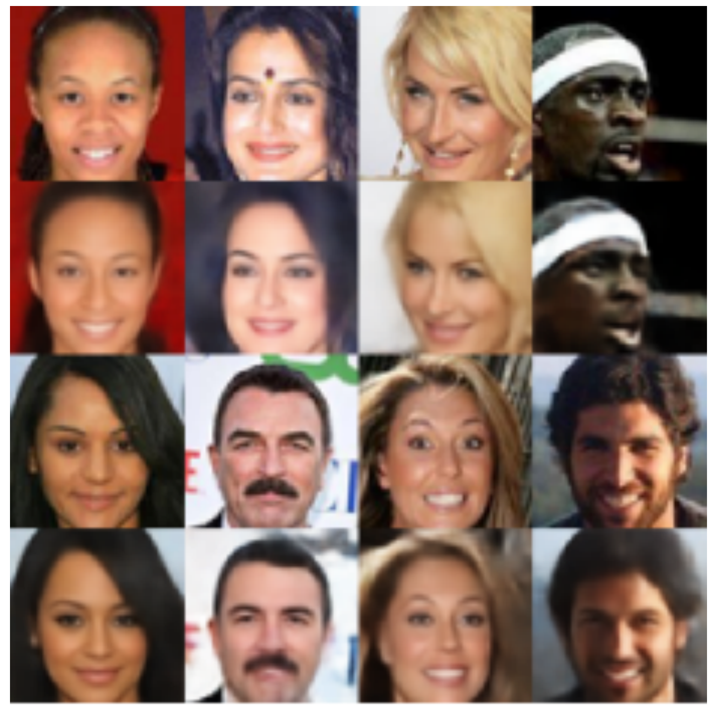


Wasserstein GAN

[Arjovsky 2017]

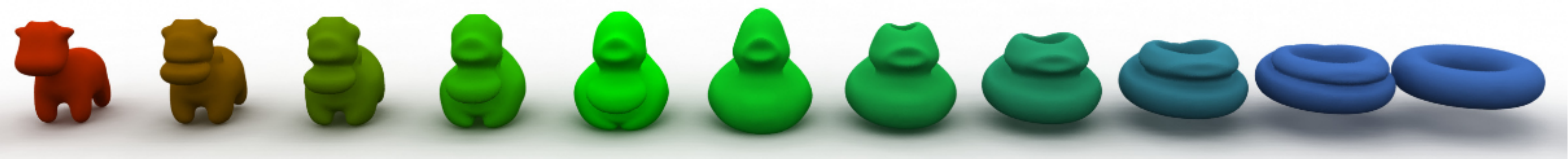
Wasserstein AE

[Tolstikhin 2018]



Shape interpolation

[Solomon, 2015]



Outline

1. **History and basics of optimal transport**

2. Wasserstein distances

3. Computational OT

Practical session (with POT toolbox)

4. Variants of OT : unbalanced OT and Gromov-Wasserstein

5. Some applications of OT in machine learning

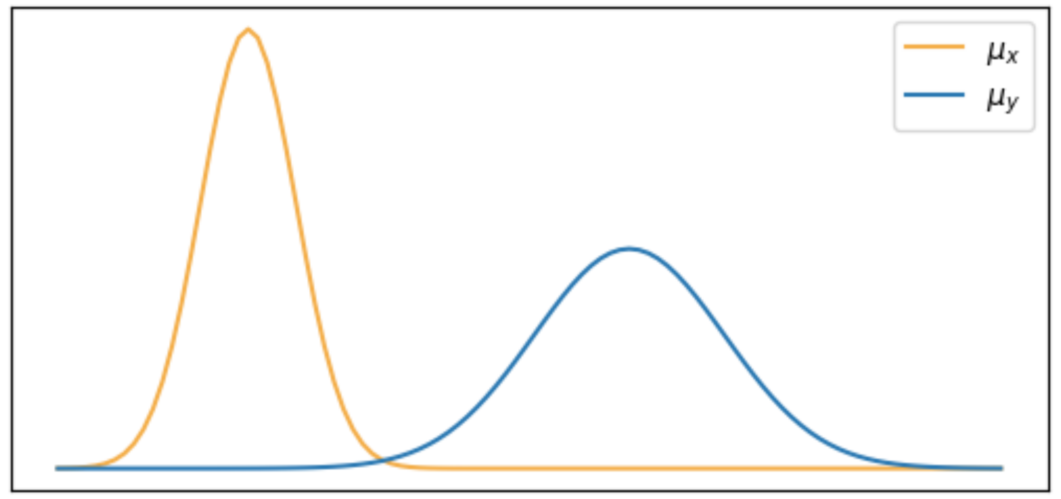
Optimal Transport in a nutshell

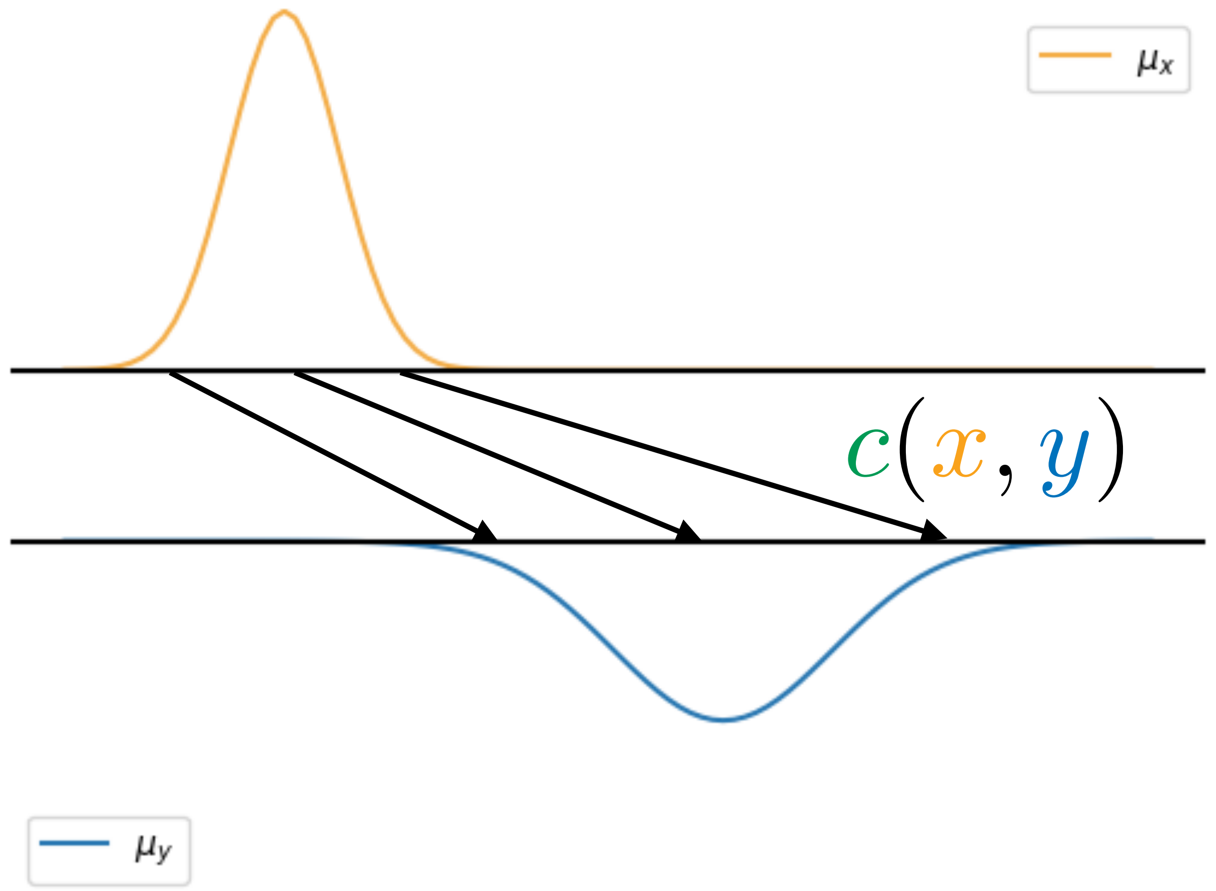
The origins of OT

Monge

1781: How to move dirt from one place (déblais) to another (remblais) while minimizing the total effort?

Assumption: there is an effort for moving dirt, function of the quantity of dirt and of the cost for transporting one shipment of dirt from x to y





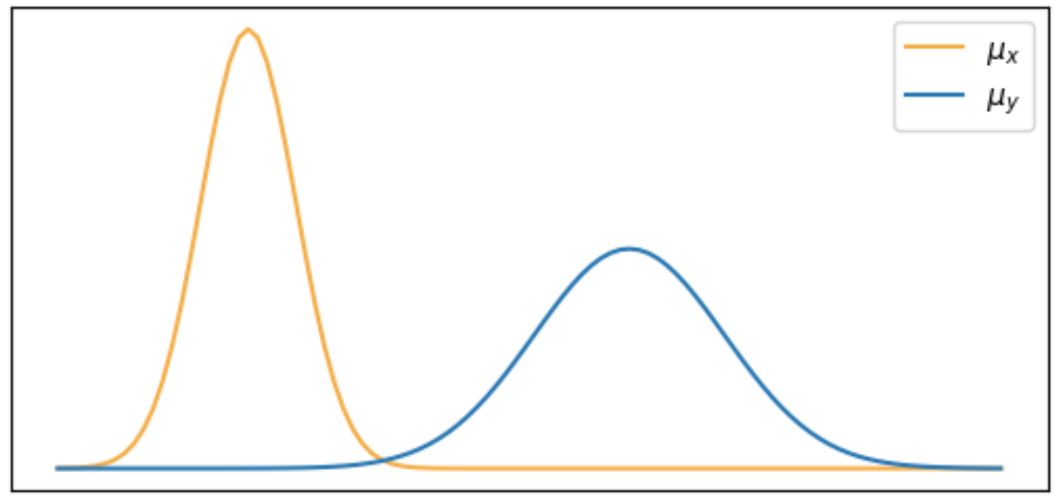
Optimal Transport in a nutshell

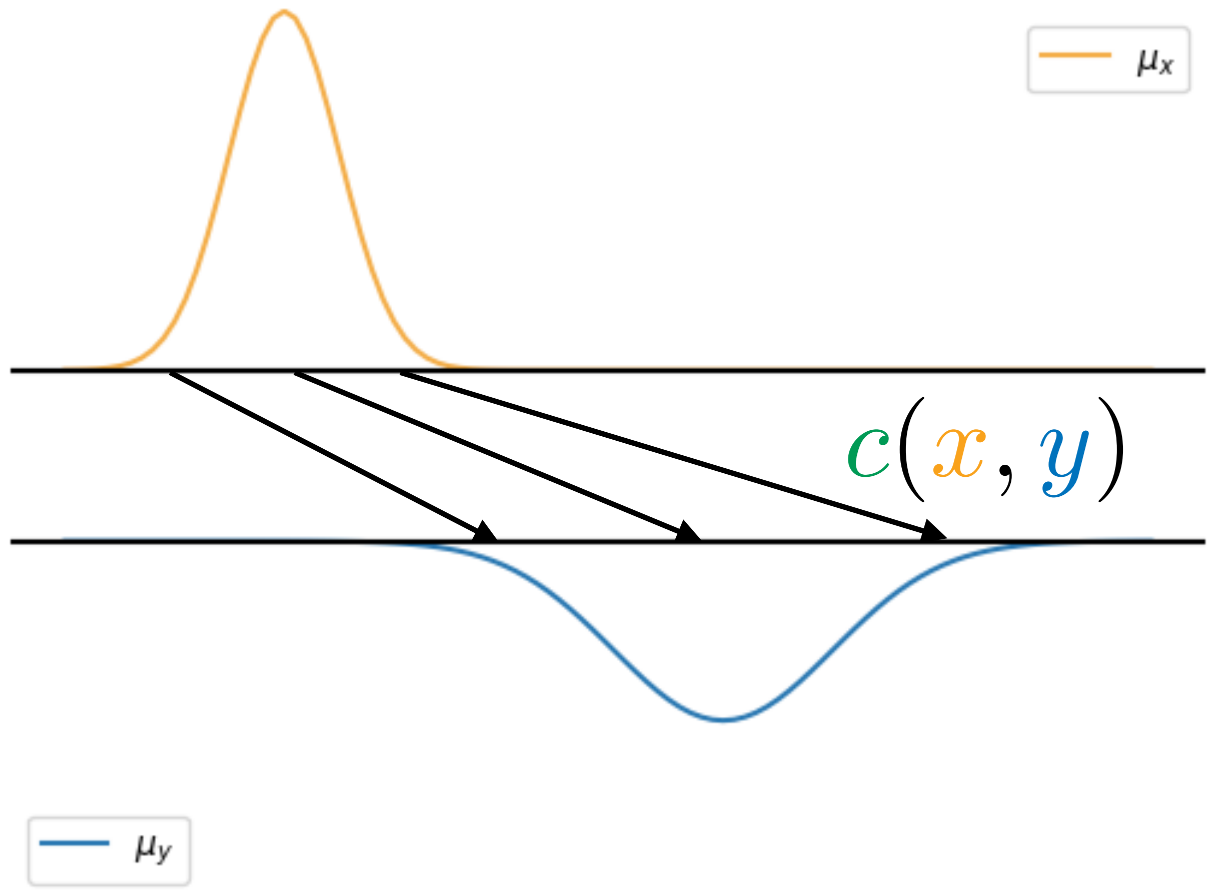
The origins of OT

Monge

1781: How to move dirt from one place (déblais) to another (remblais) while minimizing the total effort?

Assumption: there is an effort for moving dirt, function of the quantity of dirt and of the cost for transporting one shipment of dirt from x to y





Among all the possible solutions, there is one, called **optimal transport**, which is of minimal cost

Optimal Transport in a nutshell

The origins of OT

Monge

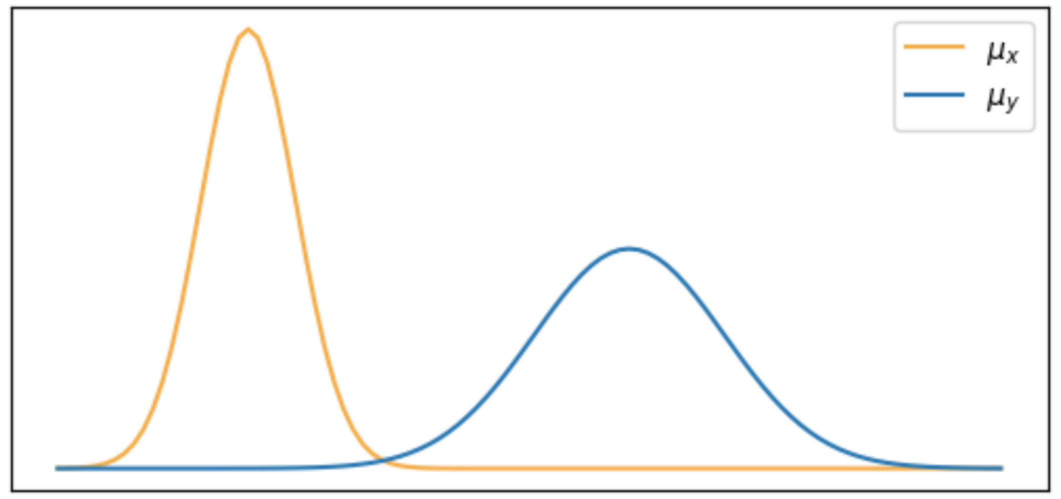
source distribution μ_x

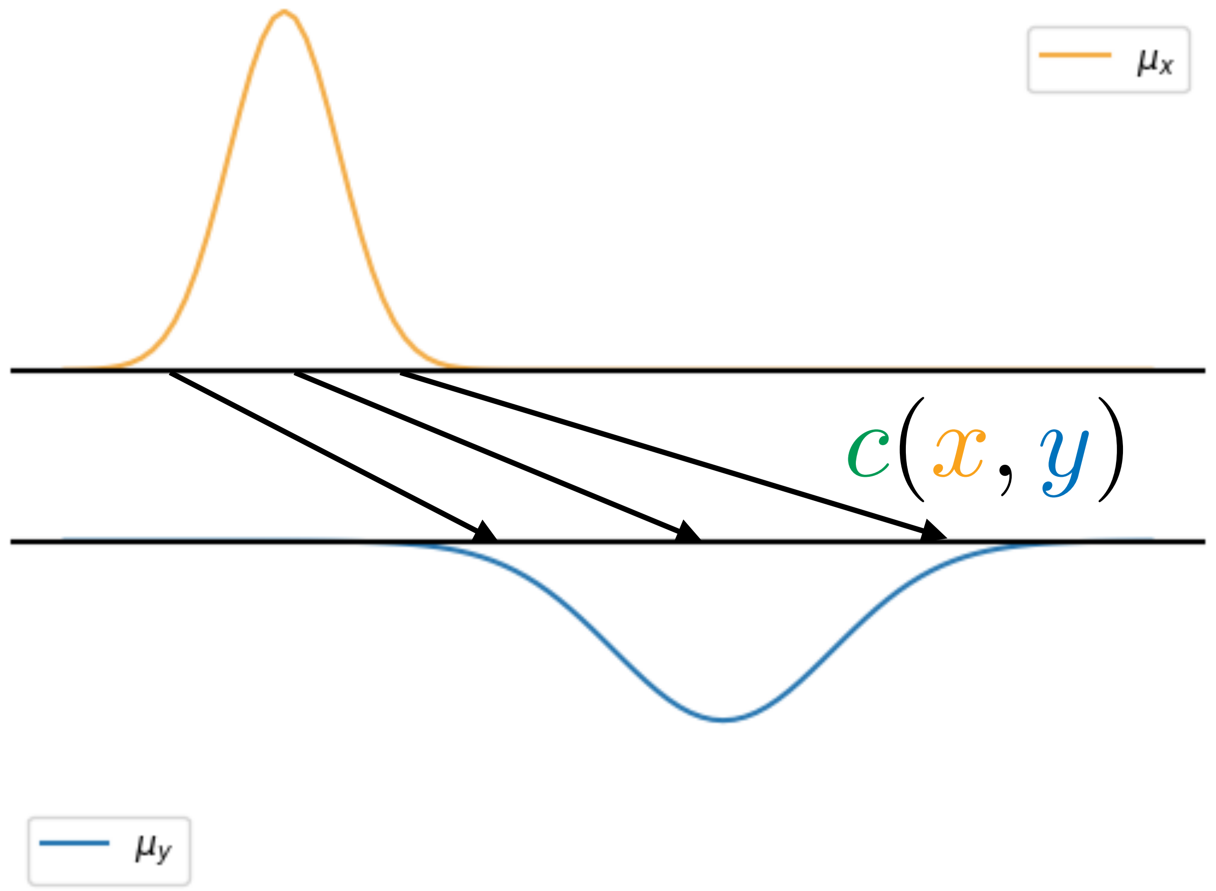
target distribution μ_y

cost of moving from x to y $c(x, y)$

Minimize the overall transportation cost

$$\inf_{T\#\mu_s=\mu_t} \int c(x, T(x))\mu_s(x)dx$$





T is the transport **map**

$T\#\mu$ is the push forward operator

Optimal Transport in a nutshell

The origins of OT

Monge

source distribution μ_x

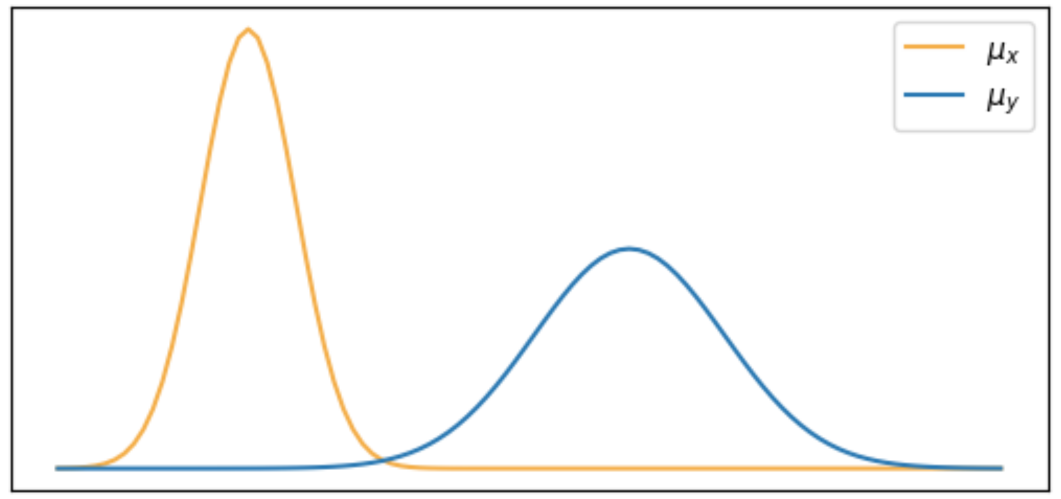
target distribution μ_y

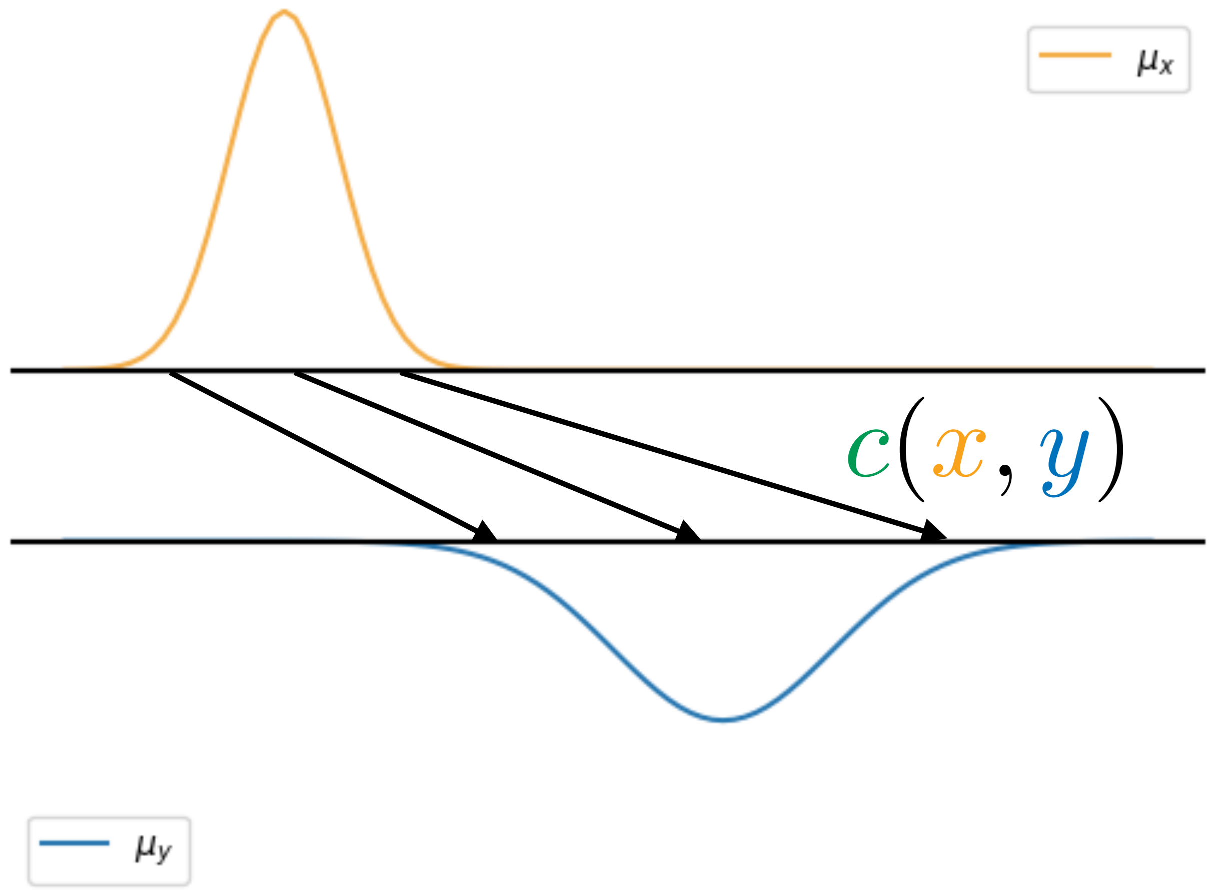
cost of moving from x to y $c(x, y)$

Minimize the overall transportation cost

$$\inf_{T\#\mu_s=\mu_t} \int c(x, T(x))\mu_s(x)dx$$

x is transported to T(x)





T is the transport **map**

$T\#\mu$ is the push forward operator

Optimal Transport in a nutshell

The origins of OT

Monge

source distribution μ_x

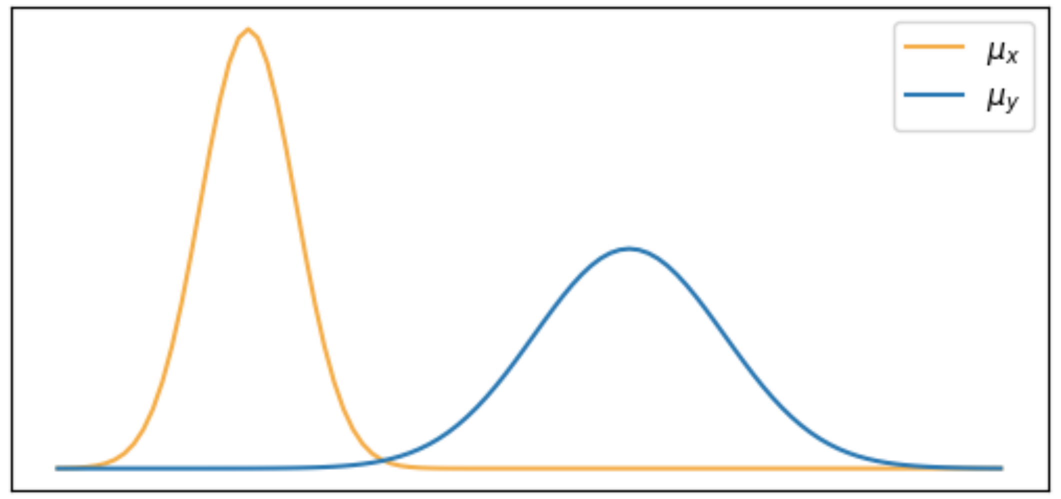
target distribution μ_y

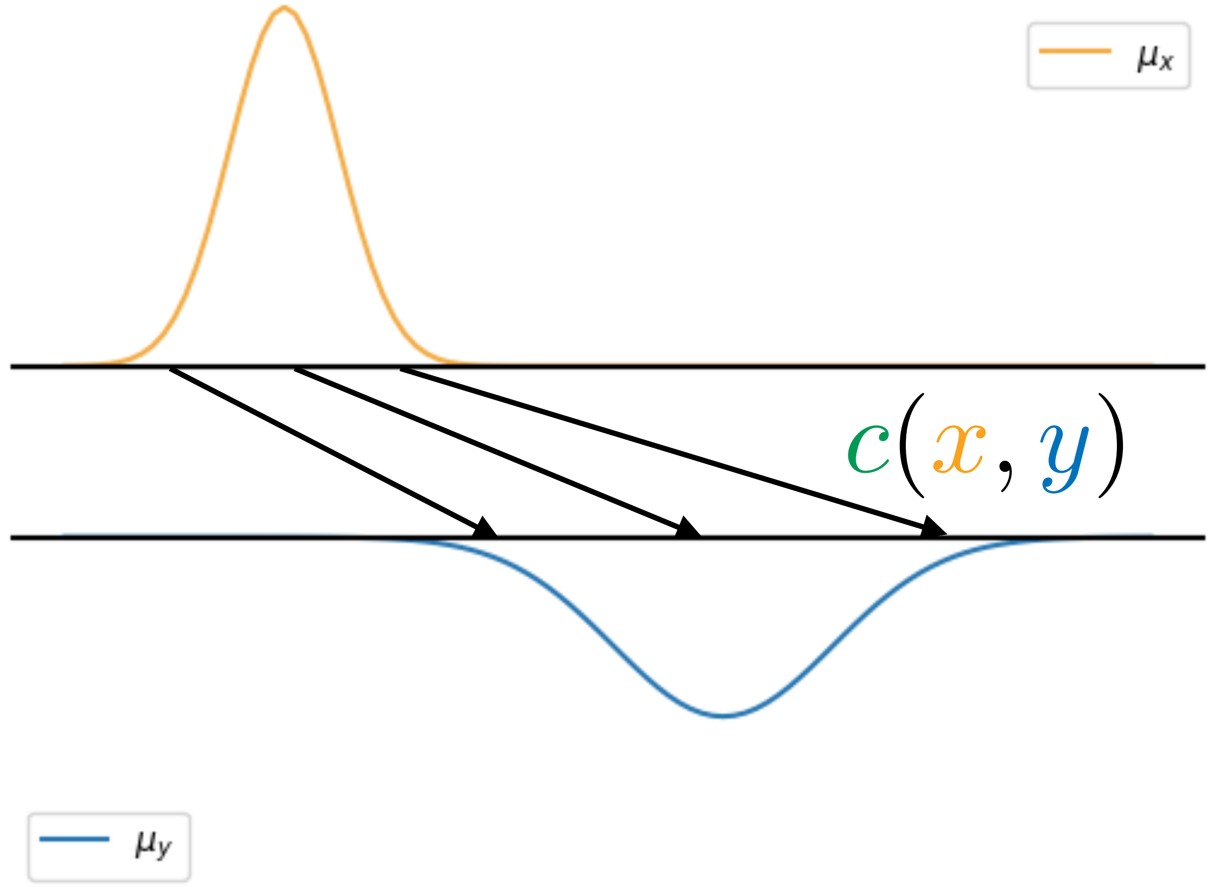
cost of moving from x to y $c(x, y)$

Minimize the overall transportation cost

$$\inf_{T\#\mu_s=\mu_t} \int c(x, T(x))\mu_s(x)dx$$

x is transported to T(x)





T is the transport **map**

$T\#\mu$ is the push forward operator

Constraint:

$T\#\mu_s = \mu_t$, i.e. no mass creation nor destruction

Optimal Transport in a nutshell

The origins of OT

Monge

source distribution μ_x

target distribution μ_y

cost of moving from x to y $c(x, y)$

Minimize the overall transportation cost

$$\inf_{T\#\mu_s=\mu_t} \int c(x, T(x))\mu_s(x)dx$$

Optimal Transport in a nutshell

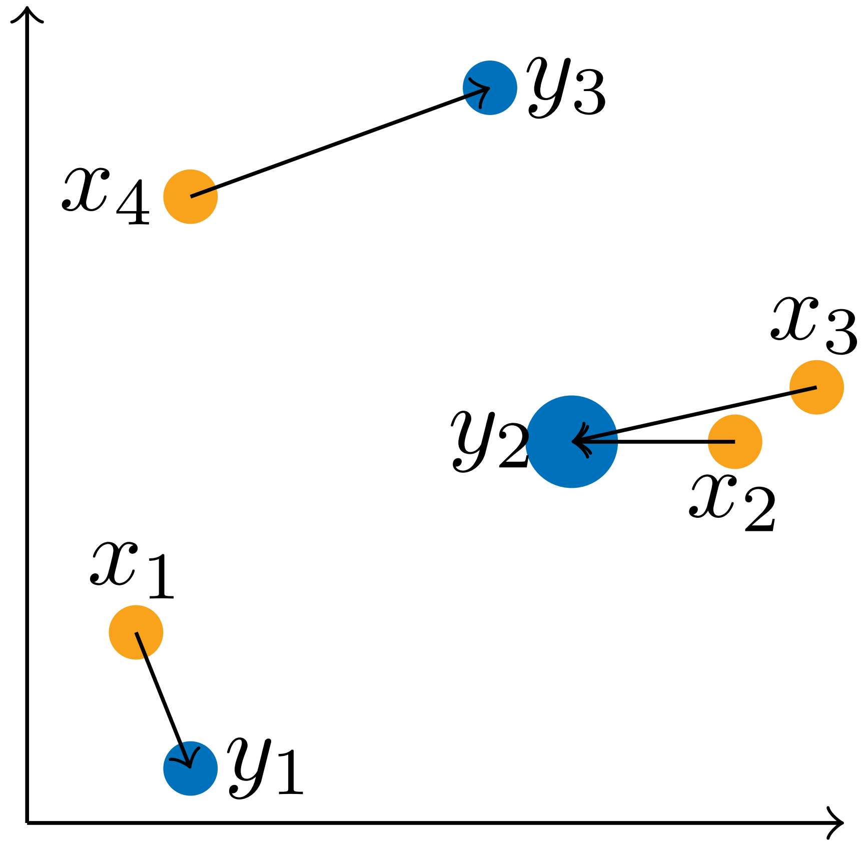
The origins of OT

Monge

source distribution μ_x
target distribution μ_y
cost of moving from x to y $c(x,y)$

Minimize the overall transportation cost

$$\inf_{T\#\mu_s=\mu_t} \int c(x,T(x))\mu_s(x)dx$$



Find a permutation such that

$$\min_{\sigma} \sum_i c(x_i, y_{\sigma(i)})$$

+ same mass

Optimal Transport in a nutshell

The origins of OT

Monge

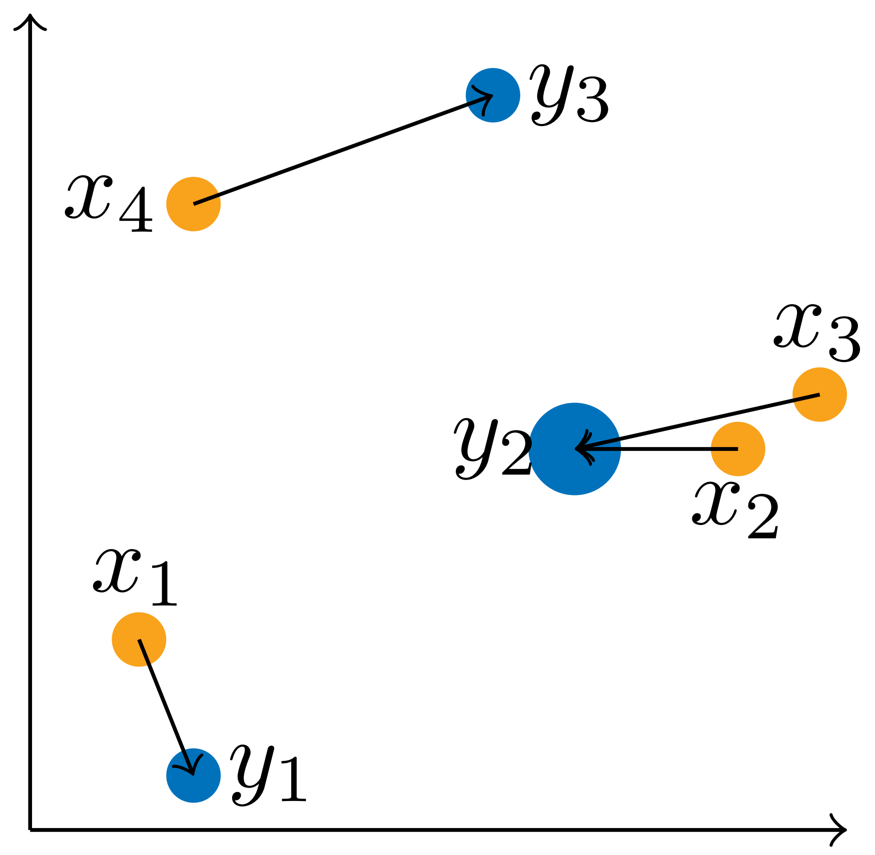
source distribution μ_x

target distribution μ_y

cost of moving from x to y $c(x, y)$

Minimize the overall transportation cost

$$\inf_{T\#\mu_s=\mu_t} \int c(x, T(x))\mu_s(x)dx$$



Find a permutation such that

$$\min_{\sigma} \sum_i c(x_i, y_{\sigma(i)})$$

+ same mass

$$T(x_1) = y_1, T(x_2) = y_2, T(x_3) = y_2, T(x_4) = y_3$$
$$\sigma(1) = 1, \sigma(2) = 2, \sigma(3) = 2, \sigma(4) = 3$$

Optimal Transport in a nutshell

The origins of OT

Monge

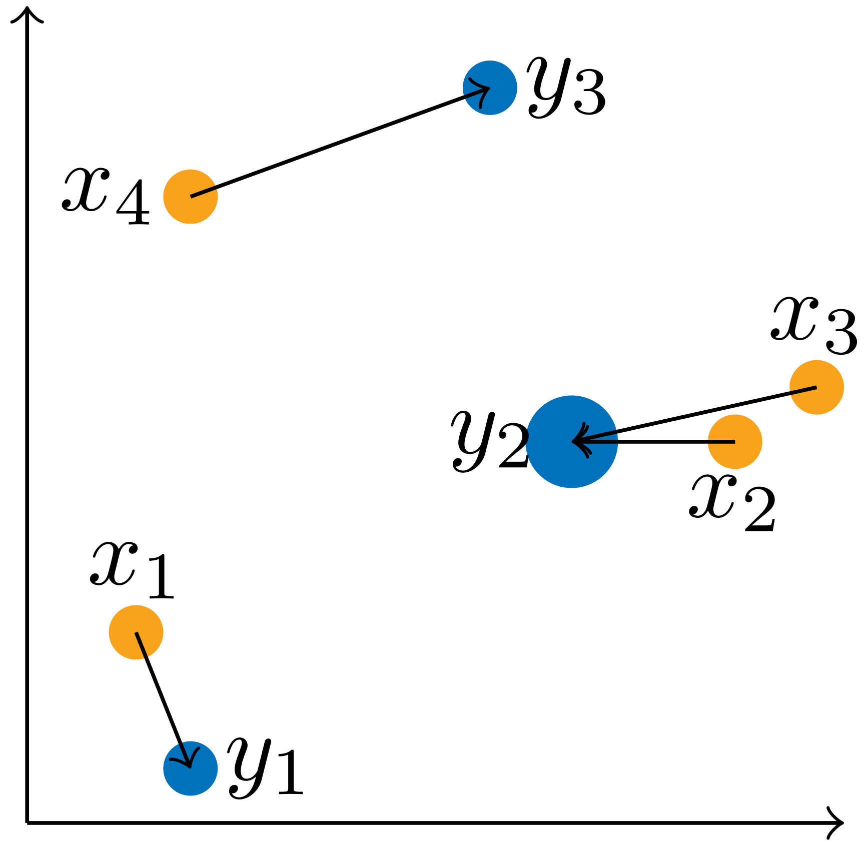
source distribution μ_x

target distribution μ_y

cost of moving from x to y $c(x, y)$

Minimize the overall transportation cost

$$\inf_{T\#\mu_s=\mu_t} \int c(x, T(x))\mu_s(x)dx$$



Find a permutation such that

$$\min_{\sigma} \sum_i c(x_i, y_{\sigma(i)})$$

+ same mass

Optimal Transport in a nutshell

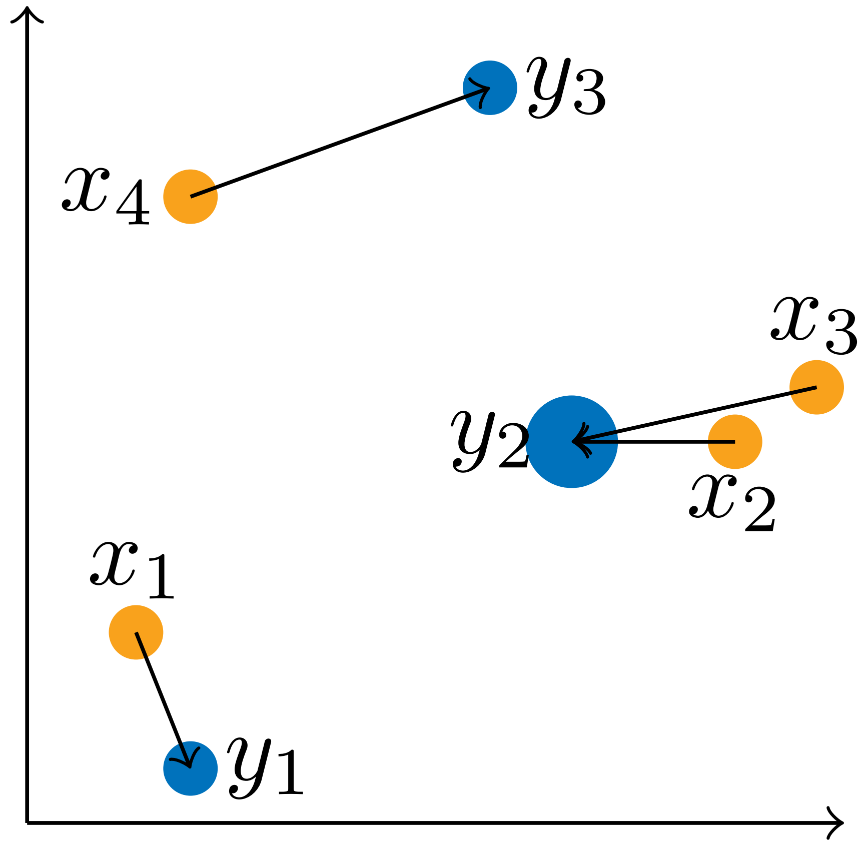
The origins of OT

Monge

source distribution μ_x
target distribution μ_y
cost of moving from x to y $c(x, y)$

Minimize the overall transportation cost

$$\inf_{T\#\mu_s=\mu_t} \int c(x, T(x))\mu_s(x)dx$$



Find a permutation such that

$$\min_{\sigma} \sum_i c(x_i, y_{\sigma(i)})$$

+ same mass

Existence of the map?
Unicity of the solution?

Optimal Transport in a nutshell

Kantorovich relaxation

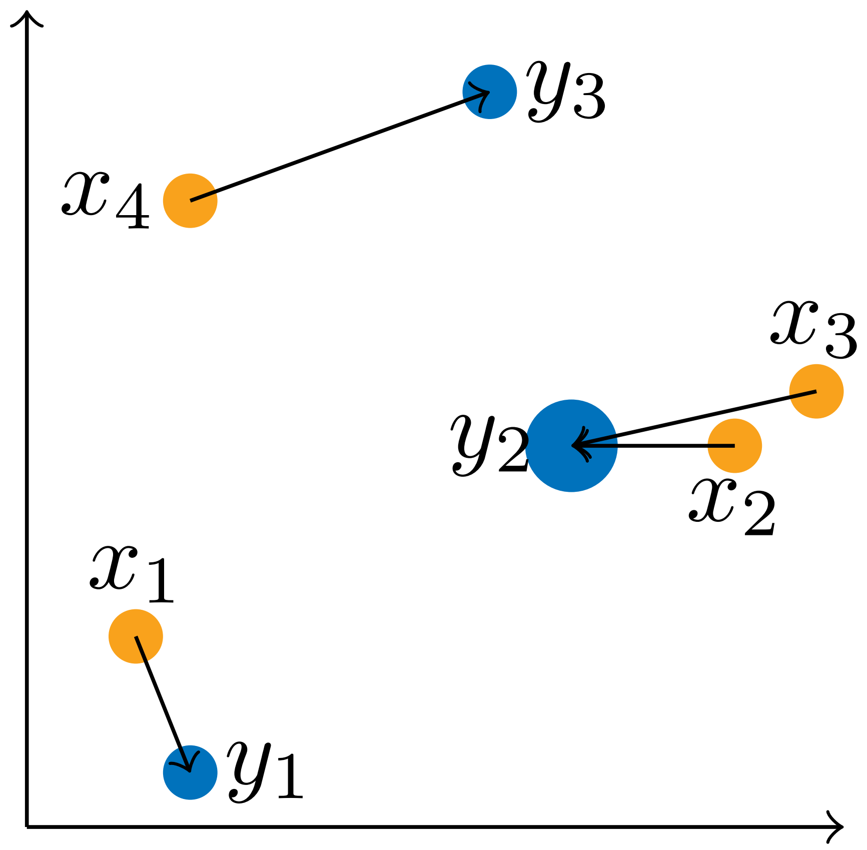
Same problem, different formulation

Kantorovich

Two discrete measures $\mu_X = \sum_{i=1}^{n} h_i \delta_{\boldsymbol{x}_i}$ and $\mu_Y = \sum_{j=1}^{m} g_j \delta_{\boldsymbol{y}_j}$

T is a probabilistic **coupling** (or OT **plan**), with **marginal** constraints

$$\boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g}) = \left\{ \boldsymbol{T} \in \mathbb{R}_+^{n \times m} | \boldsymbol{T} 1_m = \boldsymbol{h}, \boldsymbol{T}^\top 1_n = \boldsymbol{g} \right\}.$$



Optimal Transport in a nutshell

Kantorovich relaxation

Same problem, different formulation

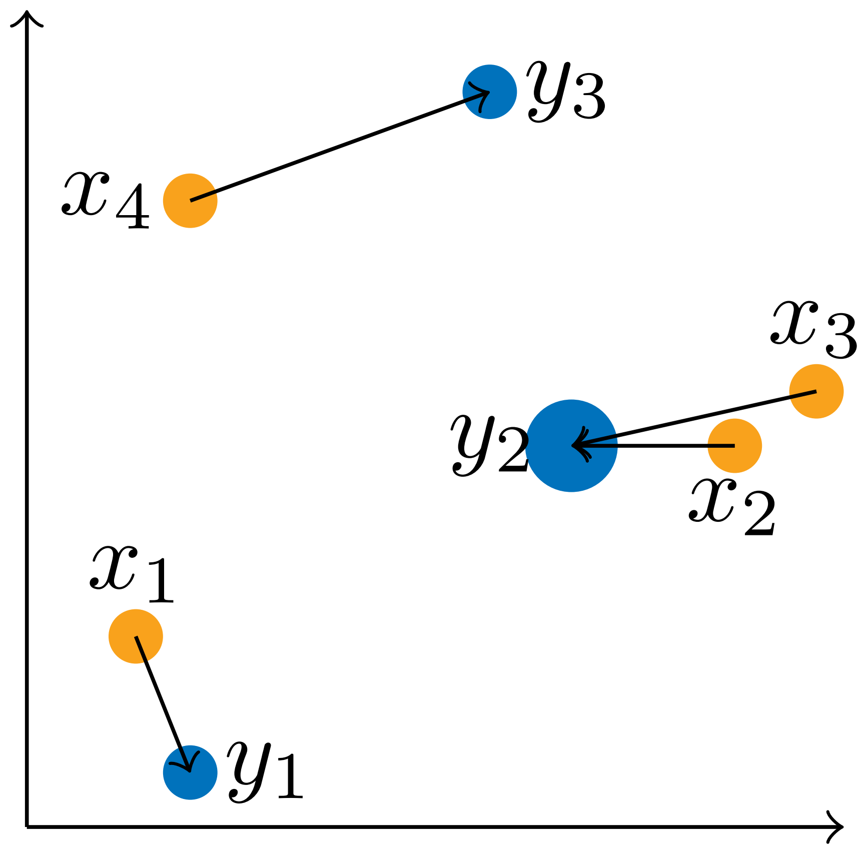
Kantorovich

Two discrete measures $\mu_X = \sum_{i=1}^{n} h_i \delta_{\boldsymbol{x}_i}$ and $\mu_Y = \sum_{j=1}^{m} g_j \delta_{\boldsymbol{y}_j}$

T is a probabilistic **coupling** (or OT **plan**), with **marginal** constraints

$$\mathbf{\Pi}(\boldsymbol{h}, \boldsymbol{g}) = \left\{ \boldsymbol{T} \in \mathbb{R}_+^{n \times m} \middle| \boldsymbol{T} 1_m = \boldsymbol{h}, \boldsymbol{T}^\top 1_n = \boldsymbol{g} \right\}.$$

it is now a matrix, or OT plan



Optimal Transport in a nutshell

Kantorovich relaxation

Same problem, different formulation

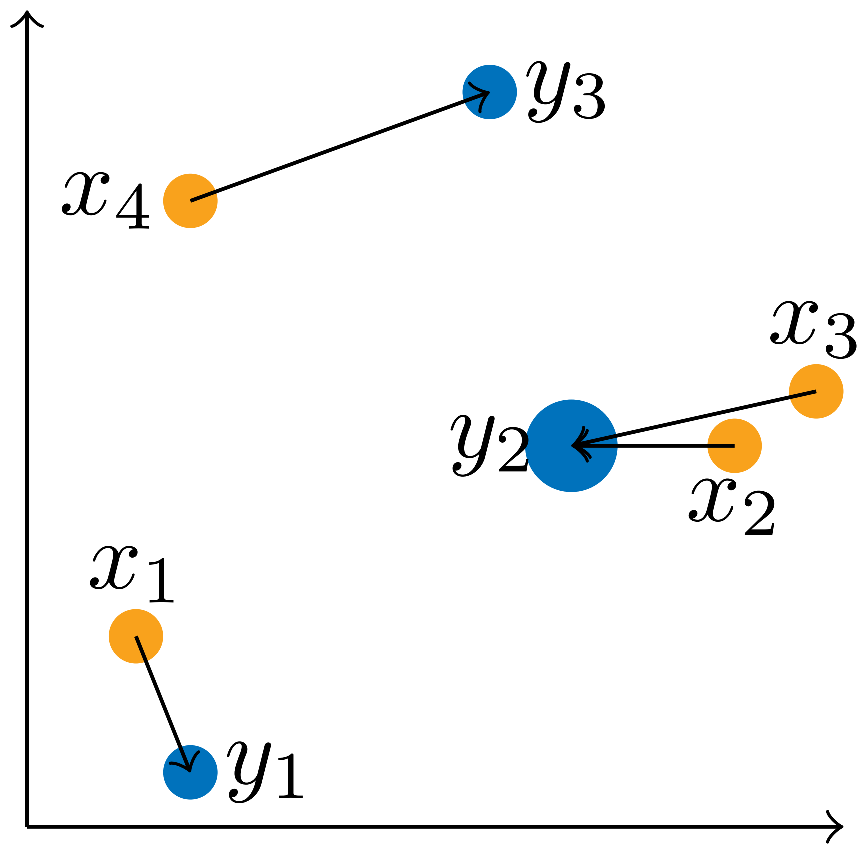
Kantorovich

Two discrete measures $\mu_X = \sum_{i=1}^{n} h_i \delta_{\boldsymbol{x}_i}$ and $\mu_Y = \sum_{j=1}^{m} g_j \delta_{\boldsymbol{y}_j}$

T is a probabilistic **coupling** (or OT **plan**), with **marginal** constraints

$$\boldsymbol{\Pi}(\boldsymbol{h},\boldsymbol{g}) = \left\{ \boldsymbol{T} \in \mathbb{R}_+^{n\times m} \,\middle|\, \boldsymbol{T} 1_m = \boldsymbol{h}, \boldsymbol{T}^\top 1_n = \boldsymbol{g} \right\}.$$

it is now a matrix, or OT plan



$$\begin{array}{cccc|c}
\multicolumn{3}{c}{T^* \text{ matrix}} & & \boldsymbol{h} \\
\frac{1}{4} & 0 & 0 & & \frac{1}{4} \\
0 & \frac{1}{4} & 0 & & \frac{1}{4} \\
0 & \frac{1}{4} & 0 & & \frac{1}{4} \\
0 & 0 & \frac{1}{4} & & \frac{1}{4} \\
\hline
\frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \boldsymbol{g} &
\end{array}$$

$$\sum_j T_{ij} = h_i$$

$$\sum_i T_{ij} = g_j$$

Optimal Transport in a nutshell

Kantorovich relaxation

Same problem, different formulation

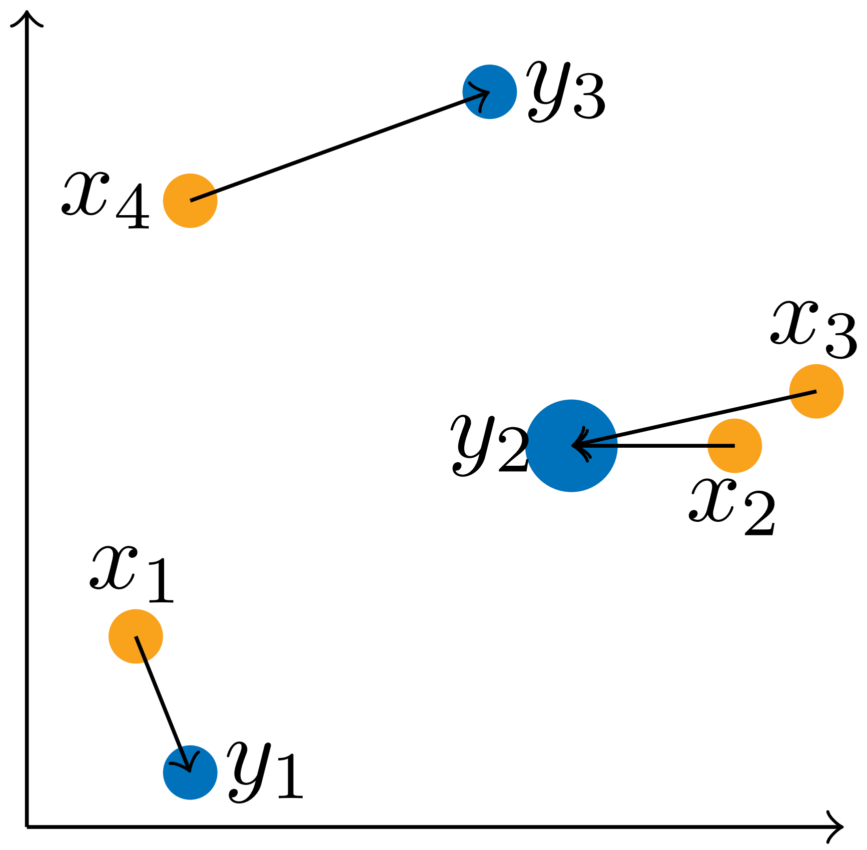
Kantorovich

Two discrete measures $\mu_X = \sum_{i=1}^{n} h_i \delta_{\boldsymbol{x}_i}$ and $\mu_Y = \sum_{j=1}^{m} g_j \delta_{\boldsymbol{y}_j}$

T is a probabilistic **coupling** (or OT **plan**), with **marginal** constraints

$$\mathbf{\Pi}(\boldsymbol{h}, \boldsymbol{g}) = \left\{ \boldsymbol{T} \in \mathbb{R}_+^{n \times m} \middle| \boldsymbol{T} 1_m = \boldsymbol{h}, \boldsymbol{T}^\top 1_n = \boldsymbol{g} \right\}.$$

it is now a matrix, or OT plan



$$\begin{array}{cccc|c} \multicolumn{3}{c}{T^* \text{ matrix}} & & \boldsymbol{h} \\ \frac{1}{4} & 0 & 0 & & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 & & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 & & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} & & \frac{1}{4} \\ \hline \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \boldsymbol{g} & \end{array}$$

$$\sum_j T_{ij} = h_i$$

no mass creation, nor destruction

$$\sum_i T_{ij} = g_j$$

Optimal Transport in a nutshell

Kantorovich relaxation

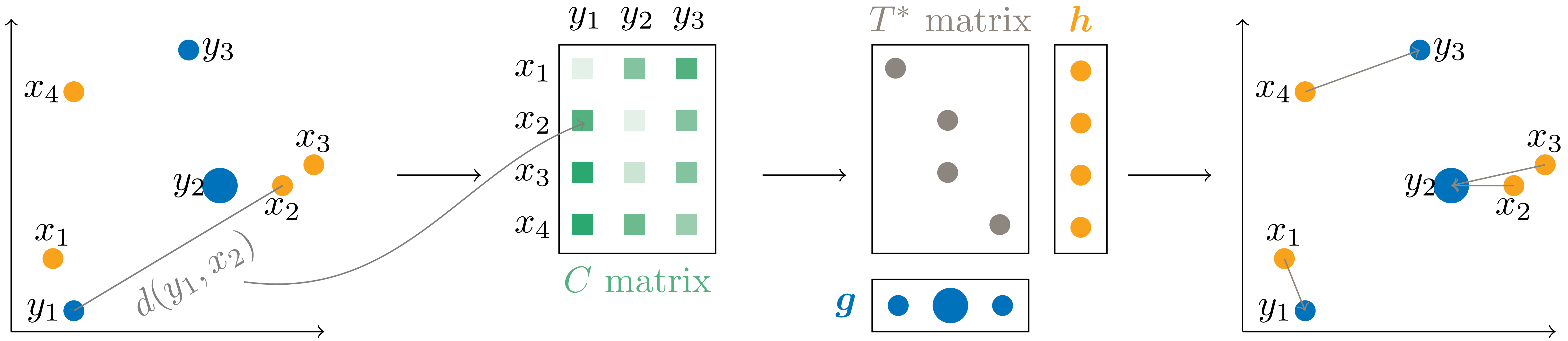
Same problem, different formulation

Kantorovich

The Kantorovich relaxation aims to solve

$$\mathrm{OT}(\boldsymbol{h},\boldsymbol{g}) = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \langle \boldsymbol{C}, \boldsymbol{T}\rangle = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \sum_{i,j} C_{i,j}T_{i,j}$$

with the constraint $\mathbf{\Pi}(\boldsymbol{h},\boldsymbol{g}) = \left\{\boldsymbol{T} \in \mathbb{R}_+^{n\times m} | \boldsymbol{T}1_m = \boldsymbol{h}, \boldsymbol{T}^\top 1_n = \boldsymbol{g}\right\}.$



Optimal Transport in a nutshell

Kantorovich relaxation

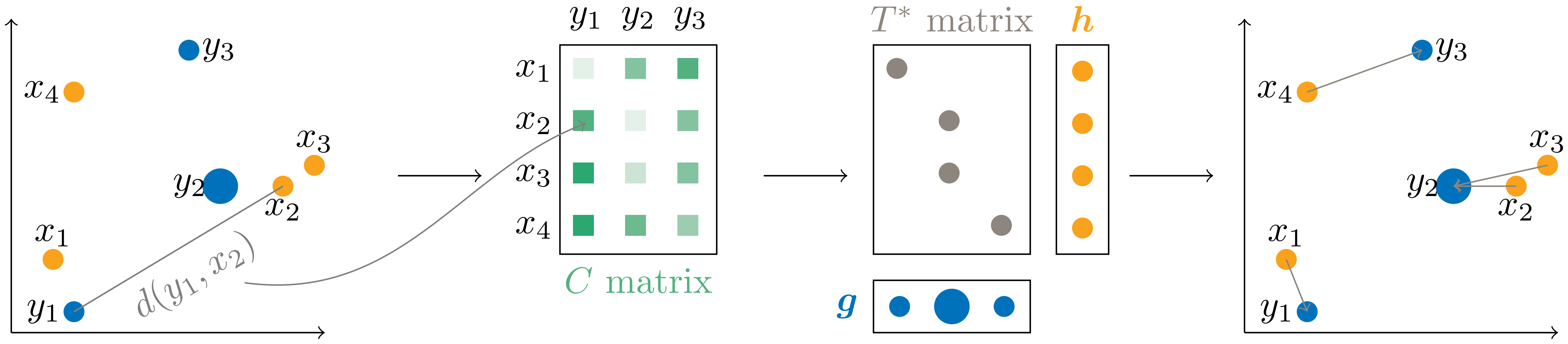
Same problem, different formulation

Kantorovich

The Kantorovich relaxation aims to solve

$$\mathrm{OT}(\boldsymbol{h},\boldsymbol{g}) = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \langle \boldsymbol{C},\boldsymbol{T}\rangle = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \sum_{i,j} C_{i,j}T_{i,j}$$

with the constraint $\boldsymbol{\Pi}(\boldsymbol{h},\boldsymbol{g}) = \left\{\boldsymbol{T}\in\mathbb{R}_+^{n\times m} | \boldsymbol{T}1_m = \boldsymbol{h}, \boldsymbol{T}^\top 1_n = \boldsymbol{g}\right\}.$

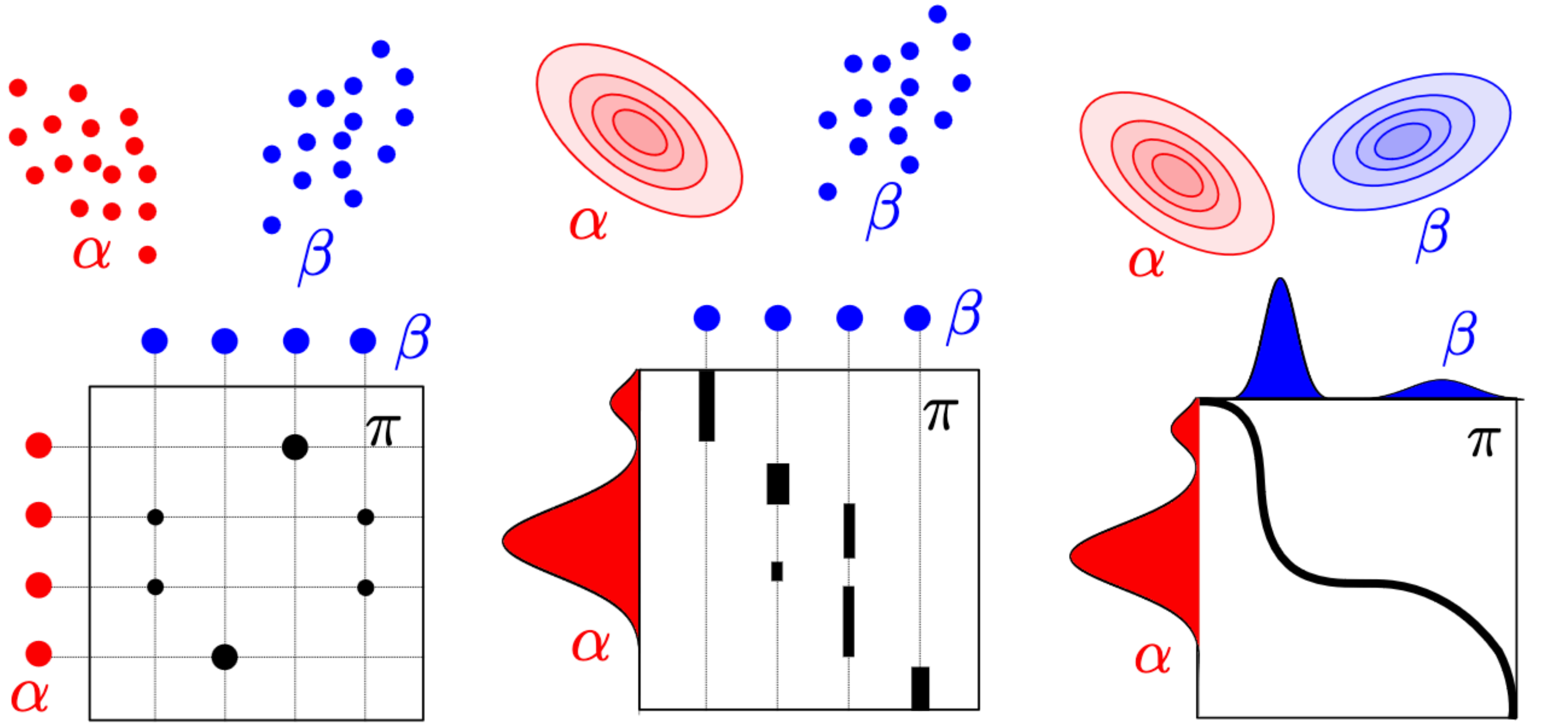


The coupling matrix $\boldsymbol{T}$ always exists as soon as $\boldsymbol{\Pi}(\boldsymbol{h},\boldsymbol{g})$ is not empty

Optimal Transport in a nutshell

Different scenarios for Kantorovich

Kantorovich



Discrete

Semidiscrete

Continuous

Figure 2.5: Schematic viewed of input measures (α, β) and couplings $\mathcal{U}(\alpha, \beta)$ encountered in the three main scenarios for Kantorovich OT. Chapter 5 is dedicated to the semidiscrete setup.

Illustration from [Peyré and Cuturi, 2019]

Outline

1. History and basics of optimal transport

2. Wasserstein distances

3. Computational OT

Practical session (with POT toolbox)

4. Variants of OT : unbalanced OT and Gromov-Wasserstein

5. Some applications of OT in machine learning

Wasserstein distances

Discrete measures

$$\mathrm{W}_p(\boldsymbol{h},\boldsymbol{g}) = \min_{\{\boldsymbol{T}1_m=\boldsymbol{h},\boldsymbol{T}^\top 1_n=\boldsymbol{g}\}} \left(\sum_{i,j} d(x_i,y_j)^p T_{i,j}\right)^{1/p}$$

Continuous measures

$$\mathrm{W}_p(\boldsymbol{\rho_0},\boldsymbol{\rho_1}) = \min_{\{\int_{\mathbb{R}} T(x,y)dy=\rho_0(x),\int_{\mathbb{R}} T(x,y)dx=\rho_1(y)\}} \left(\int\int_{\mathbb{R}^2} d(x,y)^p dT(x,y)\right)^{1/p}$$

Wasserstein distances

Discrete measures

$$\mathrm{W}_p(\boldsymbol{h},\boldsymbol{g}) = \min_{\{\boldsymbol{T}\mathbb{1}_m=\boldsymbol{h},\boldsymbol{T}^\top\mathbb{1}_n=\boldsymbol{g}\}} \left(\sum_{i,j} d(x_i,y_j)^p T_{i,j}\right)^{1/p}$$

must be a distance

Continuous measures

$$\mathrm{W}_p(\boldsymbol{\rho_0},\boldsymbol{\rho_1}) = \min_{\{\int_{\mathbb{R}} T(x,y)dy=\rho_0(x),\int_{\mathbb{R}} T(x,y)dx=\rho_1(y)\}} \left(\int\int_{\mathbb{R}^2} d(x,y)^p dT(x,y)\right)^{1/p}$$

Wasserstein distances

Discrete measures

$$\mathrm{W}_p(\boldsymbol{h},\boldsymbol{g}) = \min_{\{\boldsymbol{T}1_m=\boldsymbol{h},\boldsymbol{T}^\top 1_n=\boldsymbol{g}\}} \left(\sum_{i,j} d(x_i,y_j)^p T_{i,j} \right)^{1/p}$$

defined as the p-Wasserstein distance

(sometimes to the power of p W_p^p)

must be a distance

Continuous measures

$$\mathrm{W}_p(\boldsymbol{\rho_0},\boldsymbol{\rho_1}) = \min_{\{\int_{\mathbb{R}} T(x,y)dy=\rho_0(x),\int_{\mathbb{R}} T(x,y)dx=\rho_1(y)\}} \left(\int\int_{\mathbb{R}^2} d(x,y)^p dT(x,y) \right)^{1/p}$$

Wasserstein distances

Discrete measures

$$\mathrm{W}_p(\boldsymbol{h},\boldsymbol{g}) = \min_{\{T 1_m = \boldsymbol{h},\, T^\top 1_n = \boldsymbol{g}\}} \left(\sum_{i,j} d(x_i, y_j)^p T_{i,j}\right)^{1/p}$$

defined as the p-Wasserstein distance
(sometimes to the power of p W_p^p)

must be a distance

Continuous measures

$$\mathrm{W}_p(\boldsymbol{\rho_0},\boldsymbol{\rho_1}) = \min_{\{\int_{\mathbb{R}} T(x,y)dy=\rho_0(x),\, \int_{\mathbb{R}} T(x,y)dx=\rho_1(y)\}} \left(\int\int_{\mathbb{R}^2} d(x,y)^p dT(x,y)\right)^{1/p}$$

marginal constraints

Wasserstein distances

Discrete measures

$$\mathrm{W}_p(\boldsymbol{h}, \boldsymbol{g}) = \min_{\{\boldsymbol{T}1_m=\boldsymbol{h}, \boldsymbol{T}^\top 1_n=\boldsymbol{g}\}} \left(\sum_{i,j} d(x_i, y_j)^p T_{i,j} \right)^{1/p}$$

defined as the p-Wasserstein distance (sometimes to the power of p W_p^p)

must be a distance

Continuous measures

$$\mathrm{W}_p(\boldsymbol{\rho_0}, \boldsymbol{\rho_1}) = \min_{\{\int_{\mathbb{R}} T(x,y)dy=\rho_0(x), \int_{\mathbb{R}} T(x,y)dx=\rho_1(y)\}} \left(\int\int_{\mathbb{R}^2} d(x,y)^p dT(x,y) \right)^{1/p}$$

marginal constraints

When $d(x,y)$ is a general cost, we recover the Kantorovitch formulation

Wasserstein distances

$$\mathrm{W}_p(\boldsymbol{h}, \boldsymbol{g}) = \min_{\{\boldsymbol{T}1_m = \boldsymbol{h}, \boldsymbol{T}^\top 1_n = \boldsymbol{g}\}} \left(\sum_{i,j} d(x_i, y_j)^p T_{i,j} \right)^{1/p}$$

Some properties

Is a distance when $p \geq 1$

Also known as the Earth Mover Distance when $p = 1$ [Rubner 2000]

Admits a dual formulation

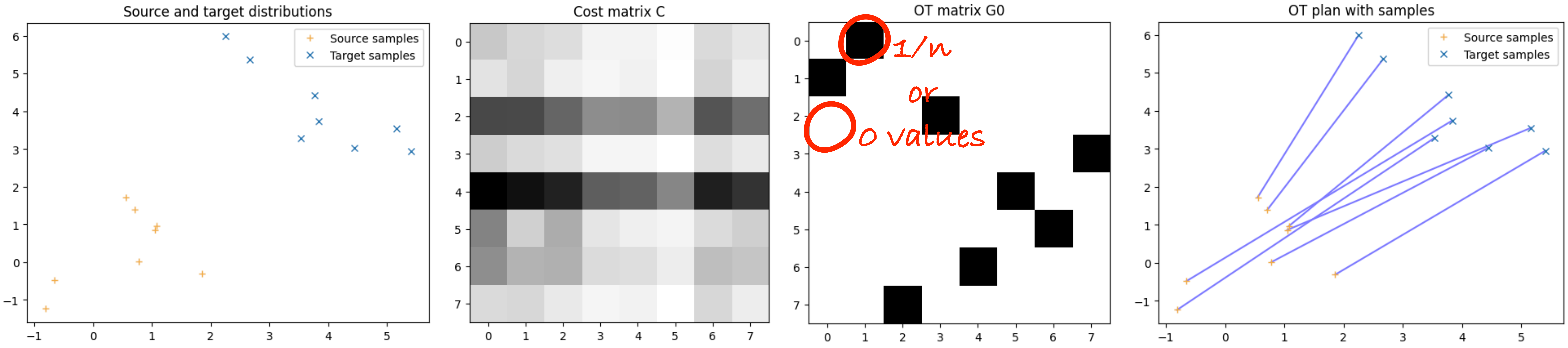
Is a linear problem with linear constraints: $O(n^3)$ complexity

Wasserstein distances

Sparsity of the transport plan

If $n = m$ and $g_i = h_j = \frac{1}{n}$, then there are exactly n non-null values for the coupling

Otherwise, there are at most $n + m + 1$ non-null values

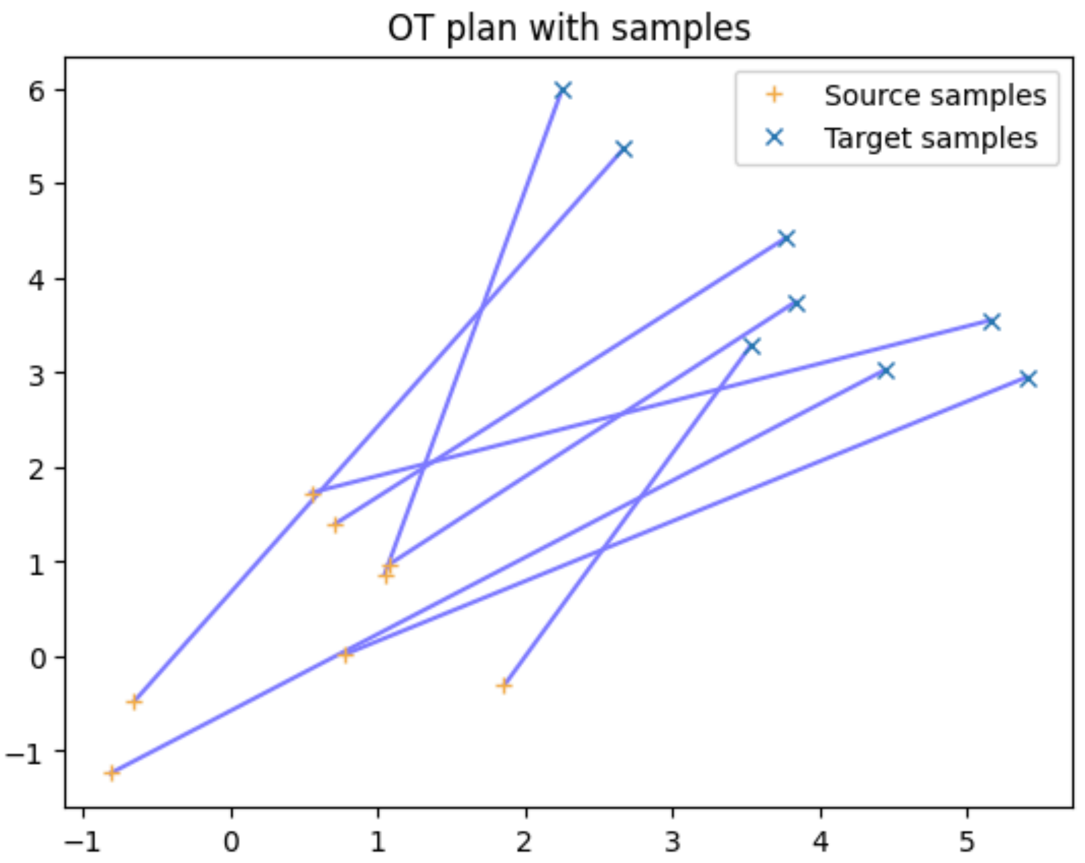


Wasserstein distances

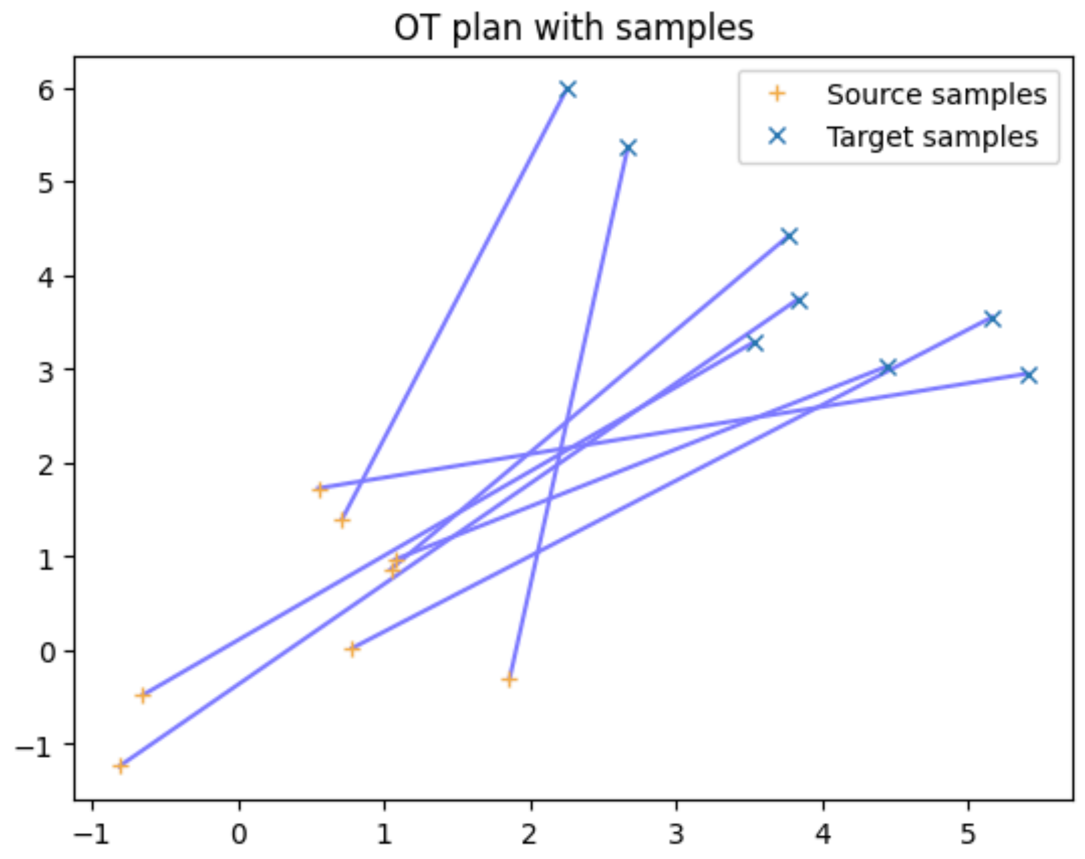
Sparsity of the transport plan

If $n = m$ and $g_i = h_j = \frac{1}{n}$, then there are exactly n non-null values for the coupling

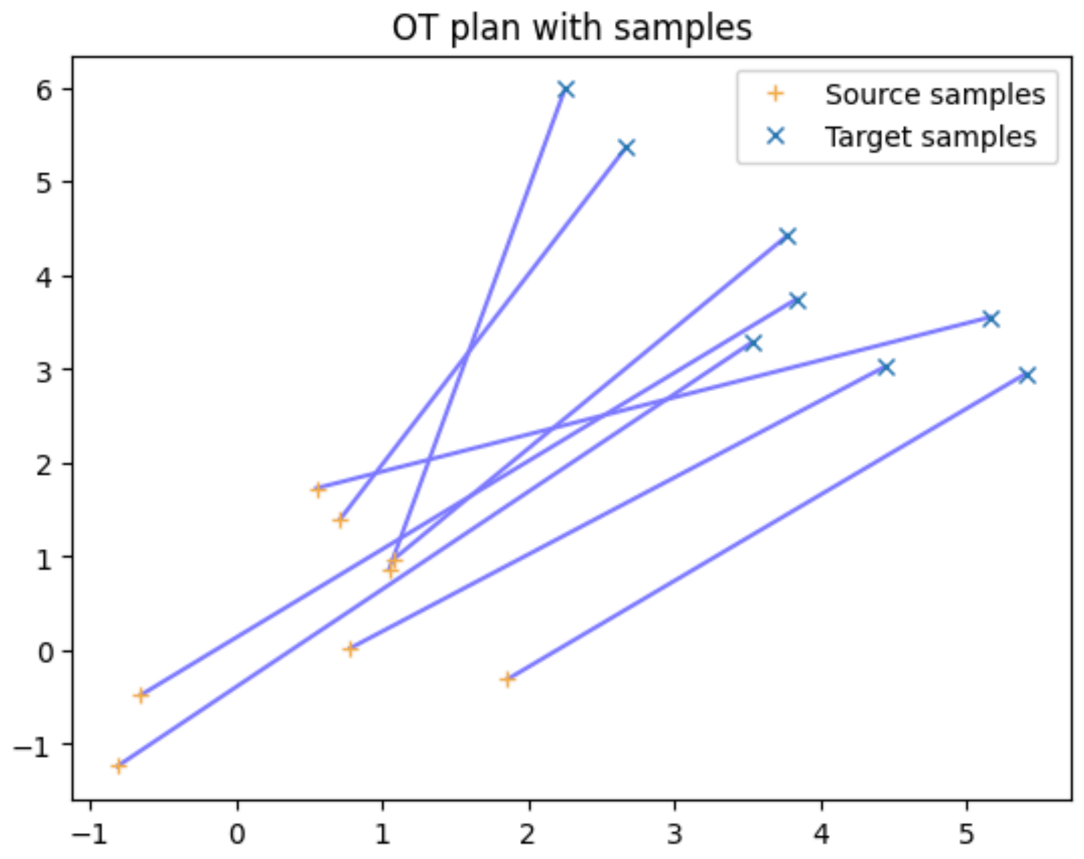
Otherwise, there are at most $n + m + 1$ non-null values



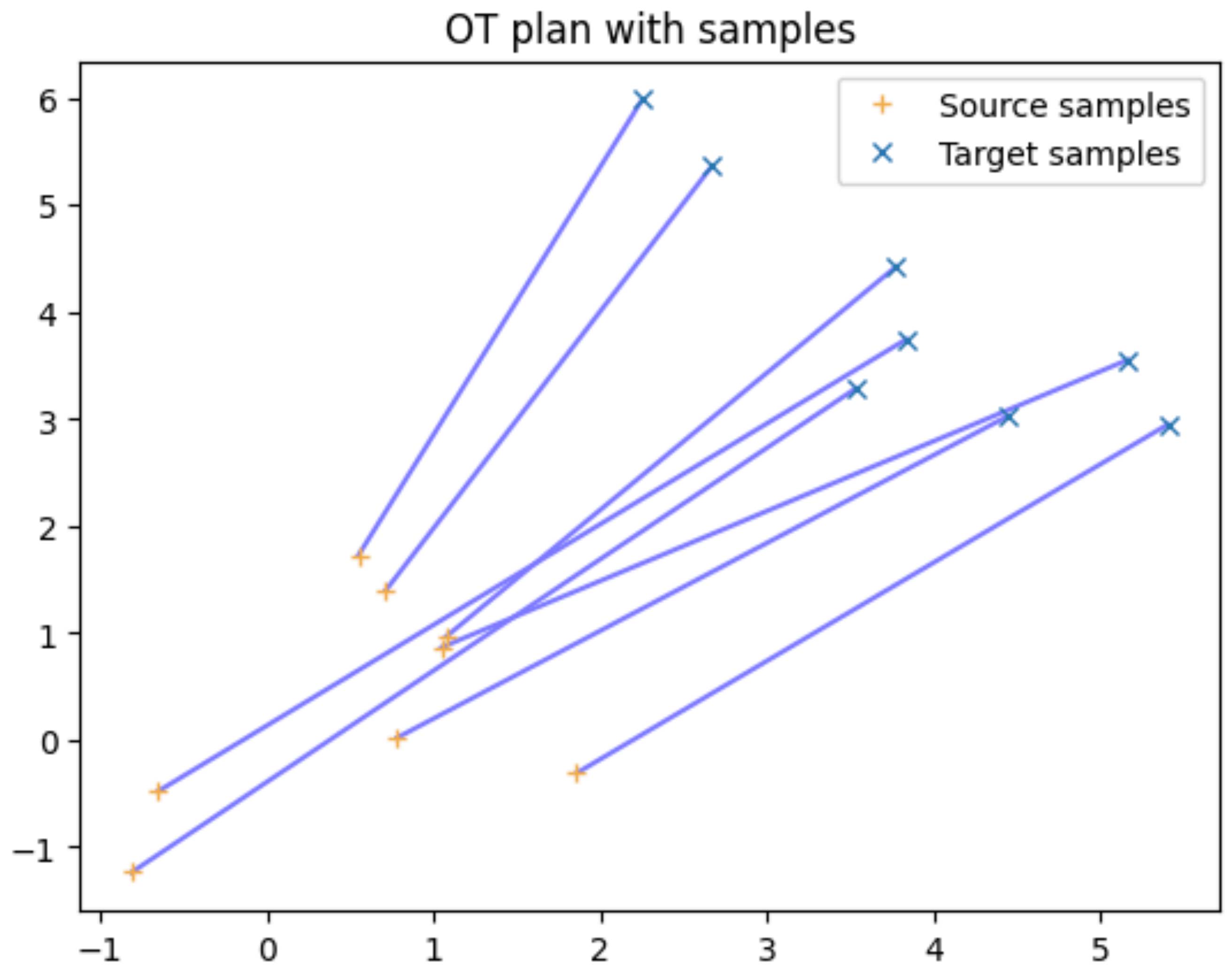
Cost = 27.8



Cost = 28.3



Cost = 26.8



Cost = 25.8

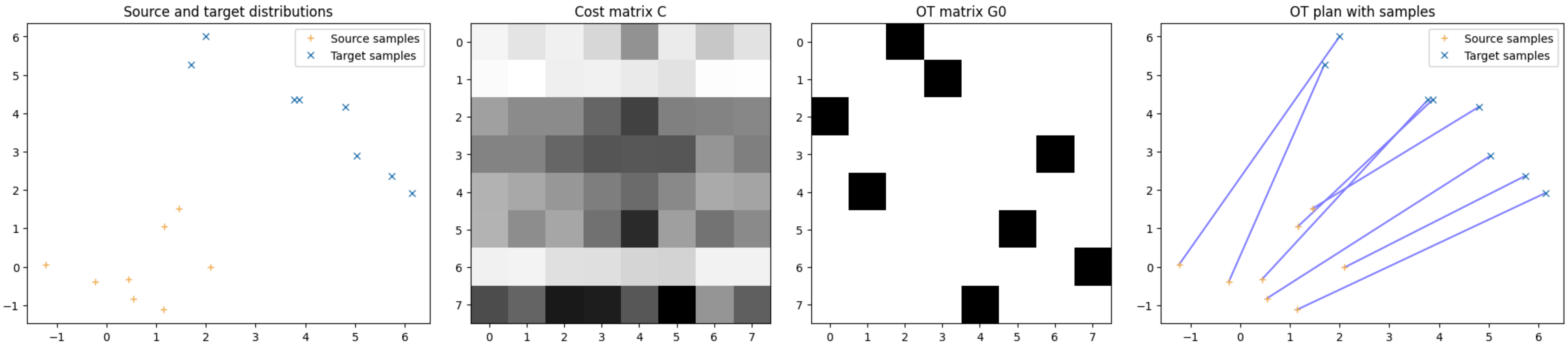
OT cost = minimal cost of coupling

Wasserstein distances

Sparsity of the transport plan

If $n = m$ and $g_i = h_j = \frac{1}{n}$, then there are exactly n non-null values for the coupling

Otherwise, there are at most $n + m + 1$ non-null values

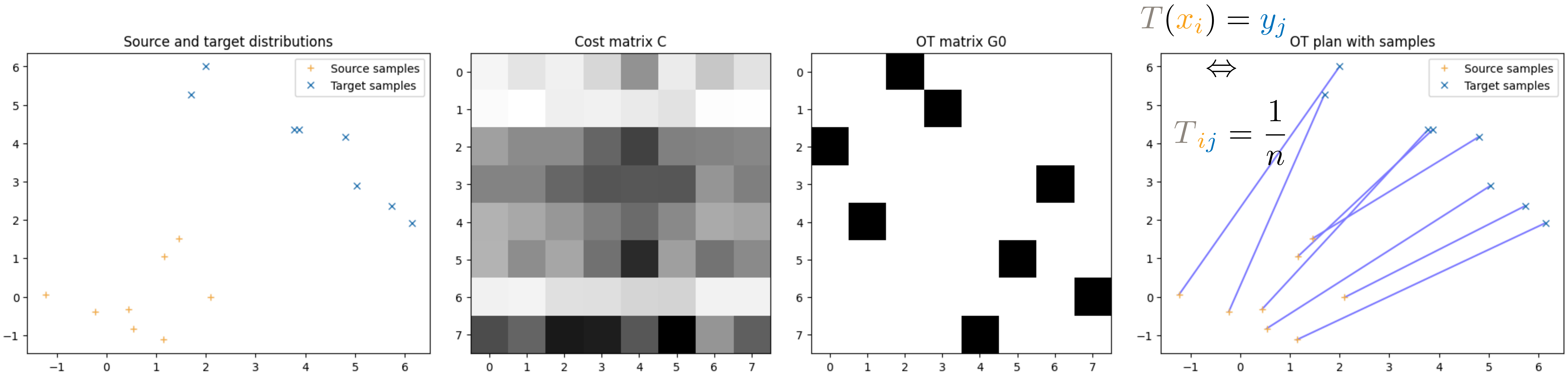


Wasserstein distances

Sparsity of the transport plan

If $n = m$ and $g_i = h_j = \frac{1}{n}$, then there are exactly n non-null values for the coupling

Otherwise, there are at most $n + m + 1$ non-null values



$T(x_i) = y_j$

⇔

$T_{ij} = \frac{1}{n}$

In this case, the Monge and Kantorovich solutions are equivalent

Wasserstein distances

Sparsity of the transport plan

If $n = m$ and $g_i = h_j = \frac{1}{n}$, then there are exactly n non-null values for the coupling

Otherwise, there are at most $n + m + 1$ non-null values

Source and target distributions
Source samples
Target samples
Cost matrix C
OT matrix G0
OT plan with samples
Source samples
Target samples
6
5
4
3
2
1
0
−1
−4
−2
0
2
4
6
0
1
2
3
4
5
6
7
0
2
4
6
8

Wasserstein distances

Sparsity of the transport plan

If $n = m$ and $g_i = h_j = \dfrac{1}{n}$, then there are exactly n non-null values for the coupling

Otherwise, there are at most $n + m + 1$ non-null values

mass splitting

Source and target distributions
Source samples
Target samples
Cost matrix C
OT matrix G0
OT plan with samples
Source samples
Target samples
−4
−2
0
2
4
6
−1
0
1
2
3
4
5
6
7
8

In this case, the Monge problem may have no solution

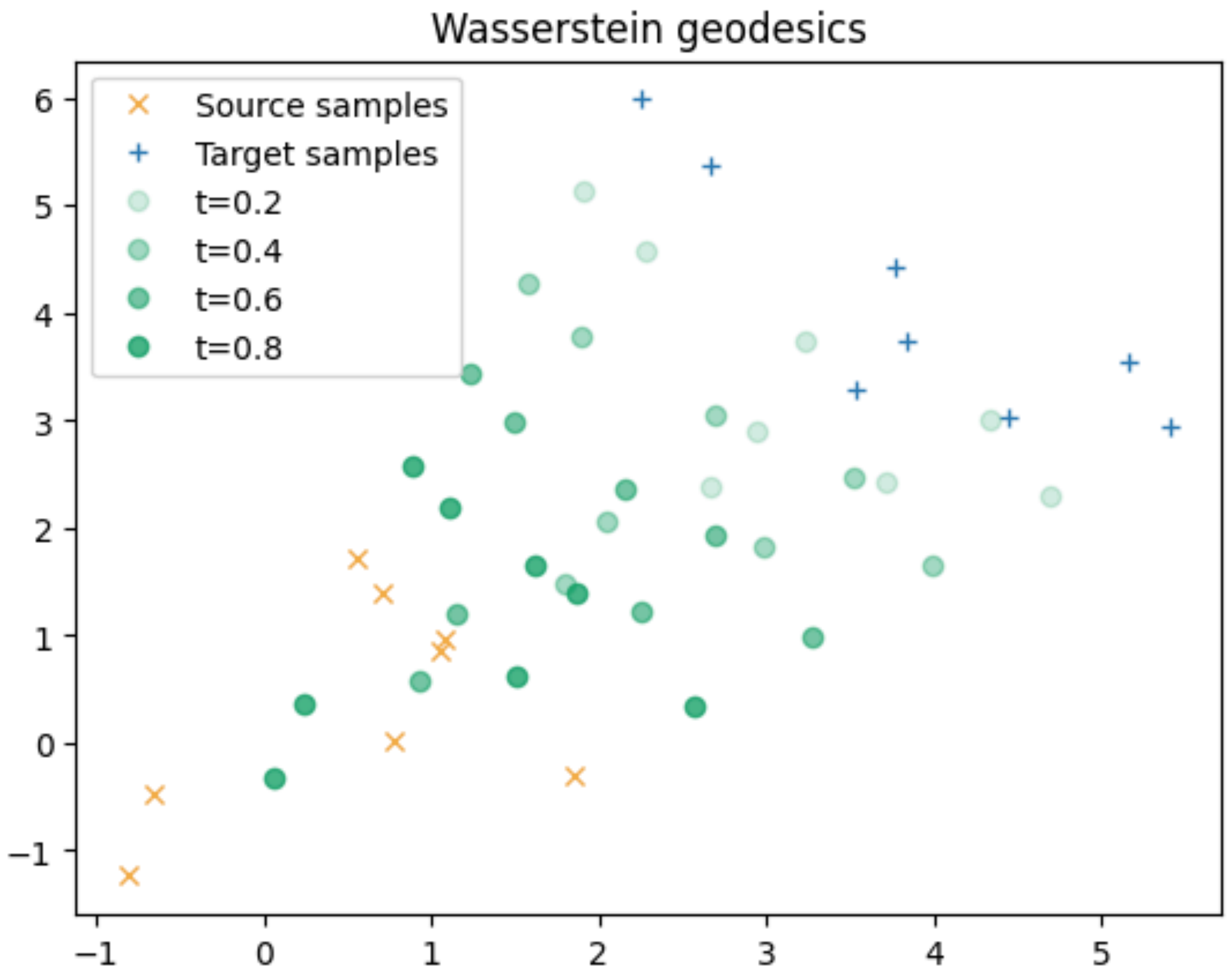
Wasserstein distances

Wasserstein Geometry

Geodesics [Ambrosio 2005]

Geodesics are shortest curves that link two distributions $((1-t)id + tT)\#\mu$

The space of probability distributions with a Wasserstein metric defines a geodesic space



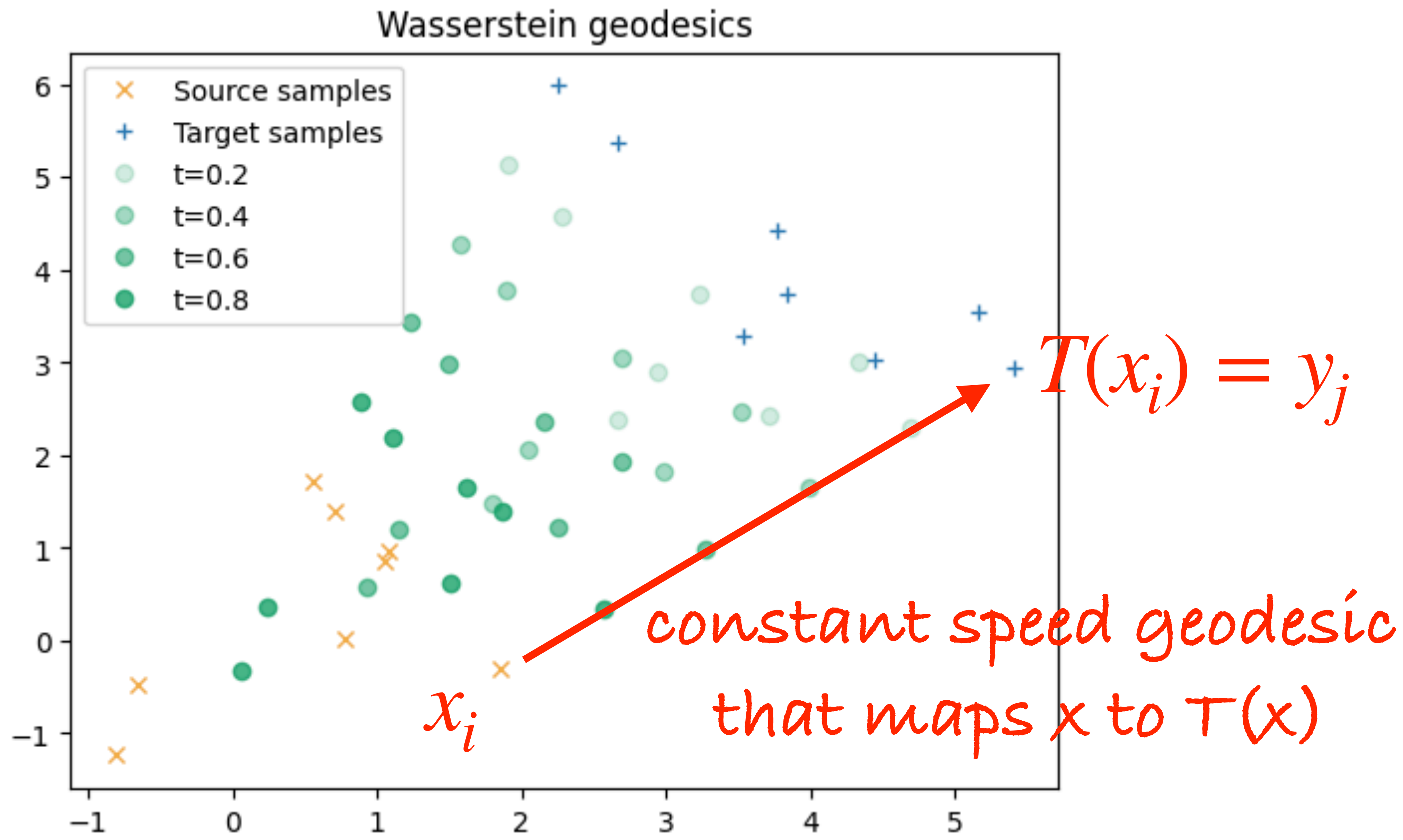
Wasserstein distances

Wasserstein Geometry

Geodesics [Ambrosio 2005]

Geodesics are shortest curves that link two distributions $((1-t)id + tT)\#\mu$

The space of probability distributions with a Wasserstein metric defines a geodesic space



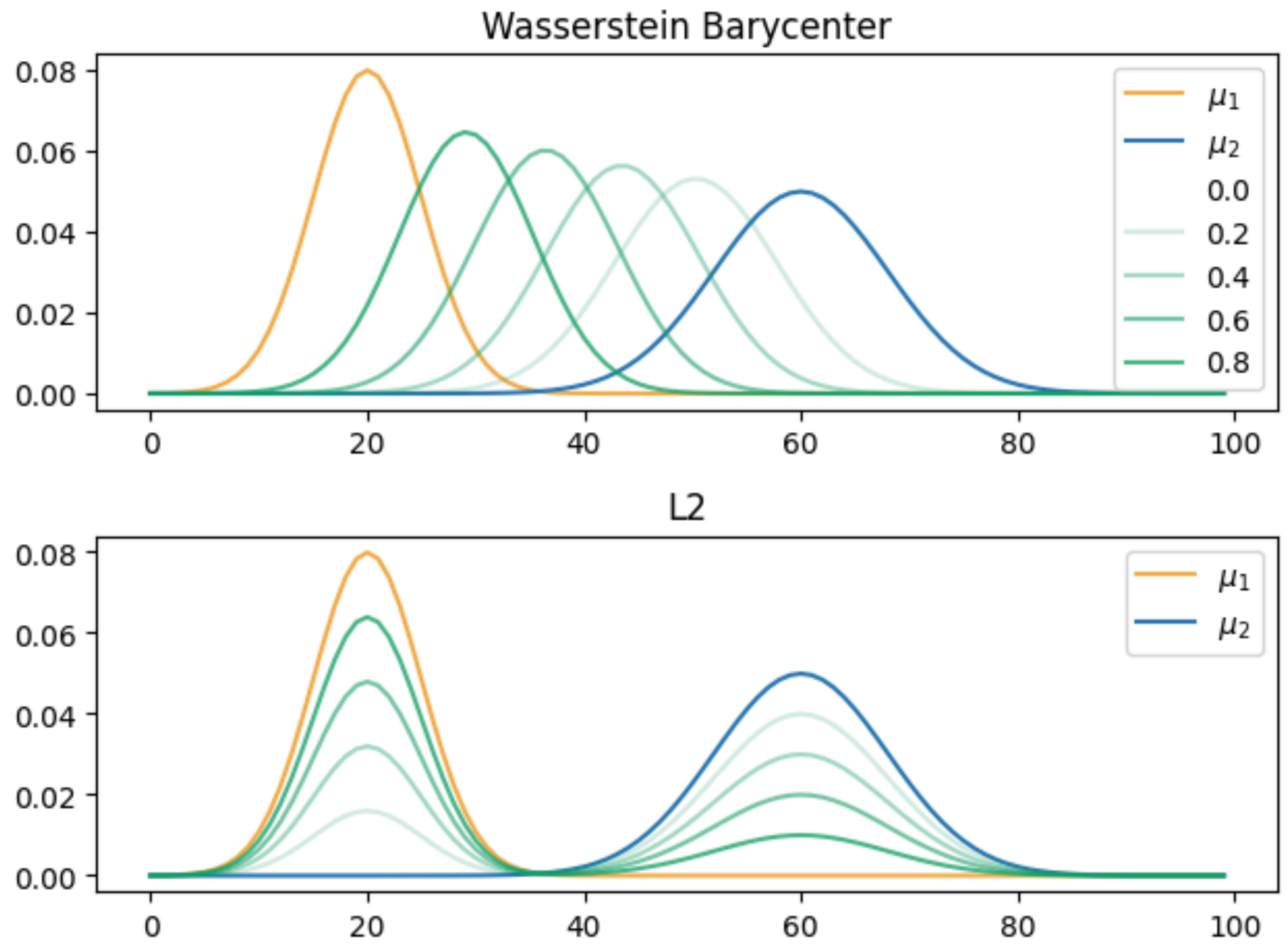
$T(x_i) = y_j$

constant speed geodesic
that maps x to T(x)

x_i

Wasserstein distances

Wasserstein Geometry


Wasserstein Barycenter

L2

Barycenters [Agueh 2011]

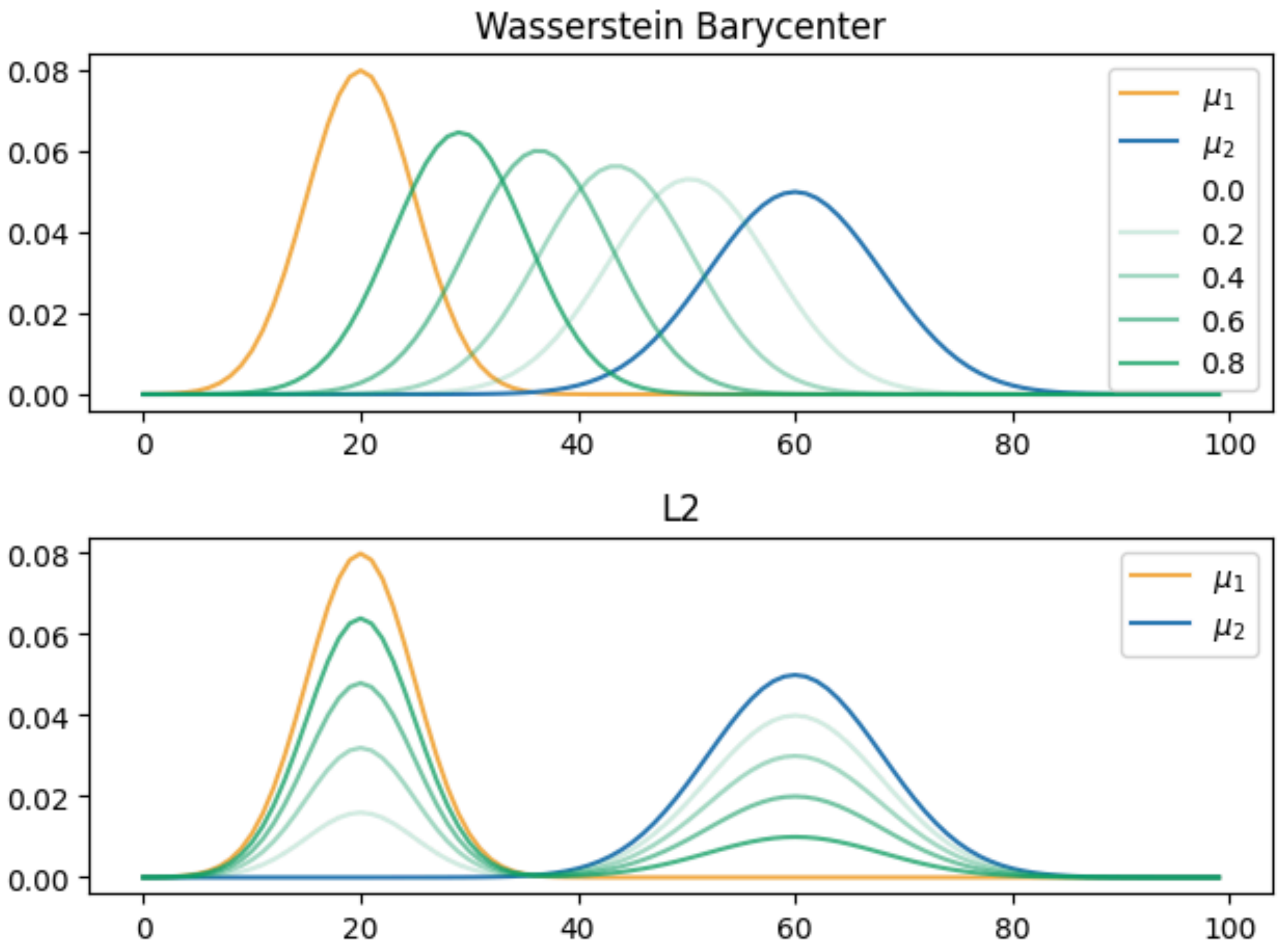
(Empirical) Wasserstein Fréchet mean

$$\arg\min_{\boldsymbol{b}} \lambda_i W_p^p(\boldsymbol{h_i}, \boldsymbol{b})$$

where λ_i are the weights associated ($\sum \lambda_i = 1$)

Wasserstein distances

Wasserstein Geometry



Barycenters [Agueh 2011]

(Empirical) Wasserstein Fréchet mean

$$\arg\min_{\boldsymbol{b}} \lambda_i W_p^p(\boldsymbol{h_i}, \boldsymbol{b})$$

where λ_i are the weights associated ($\sum \lambda_i = 1$)

Barycenters with free support (fixed weights)

$$\arg\min_{\{\boldsymbol{x_i}\}} \lambda_i W_p^p(\mu_i, \mu)$$

$$\text{such that } \mu = \sum_i^n h_i \delta_{\boldsymbol{x_i}}$$

Outline

1. History and basics of optimal transport

2. Wasserstein distances

3. **Computational OT**

Practical session (with POT toolbox)

4. Variants of OT : unbalanced OT and Gromov-Wasserstein

5. Some applications of OT in machine learning

Computational OT

Outline

$\mathrm{OT}(\boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \in \Pi(\boldsymbol{h}, \boldsymbol{g})} \langle \boldsymbol{C}, \boldsymbol{T} \rangle = \min_{\boldsymbol{T} \in \Pi(\boldsymbol{h}, \boldsymbol{g})} \sum_{i,j} C_{i,j} T_{i,j}$ → $n \times m$ variables, $n + m$ constraints, $O(n^3)$

Computational OT

Outline

$$\mathrm{OT}(\boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \in \Pi(\boldsymbol{h}, \boldsymbol{g})} \langle \boldsymbol{C}, \boldsymbol{T} \rangle = \min_{\boldsymbol{T} \in \Pi(\boldsymbol{h}, \boldsymbol{g})} \sum_{i,j} C_{i,j} T_{i,j}$$ → $n \times m$ variables, $n + m$ constraints, $O(n^3)$

Easier in some special cases (e.g. 1d or Gaussian distributions)

Need for solvers that provide approximate solutions! See [Peyré et Cuturi 2019]

1. Sliced Wasserstein
2. Regularized OT $\min_{\boldsymbol{T} \in \Pi(\boldsymbol{h}, \boldsymbol{g})} \langle \boldsymbol{C}, \boldsymbol{T} \rangle + \lambda \Omega(T)$

Computational OT

OT is a linear problem

$\mathrm{OT}(\boldsymbol{h},\boldsymbol{g}) = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \langle \boldsymbol{C},\boldsymbol{T}\rangle = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \sum_{i,j} C_{i,j}T_{i,j}$ → $n \times m$ variables, $n+m$ constraints, $O(n^3)$

We can rewrite the OT problem in a vectorial form

$$\min_{\boldsymbol{t}\geq 0} \quad F(t) = \underbrace{\boldsymbol{c}^\top t}_{\text{vectorized OT cost}}$$

$$\text{such that}$$

$$\boldsymbol{H}t = [\boldsymbol{h},\boldsymbol{g}]^\top$$

with $\boldsymbol{H}$ a $(n+m)\times nm$
matrix that encodes the constraints

Computational OT

OT is a linear problem

$\mathrm{OT}(\boldsymbol{h},\boldsymbol{g}) = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \langle \boldsymbol{C},\boldsymbol{T}\rangle = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \sum_{i,j} C_{i,j}T_{i,j}$ → $n \times m$ variables, $n+m$ constraints, $O(n^3)$

We can rewrite the OT problem in a vectorial form

$$\min_{\boldsymbol{t}\geq 0} \quad F(t) = \underbrace{\boldsymbol{c}^\top t}_{\text{vectorized OT cost}}$$

$$\text{such that}$$

$$\boldsymbol{H}t = [\boldsymbol{h},\boldsymbol{g}]^\top$$

with $\boldsymbol{H}$ a $(n+m)\times nm$ matrix that encodes the constraints

Combination of identity matrices and matrices of ones

Computational OT

OT is a linear problem

$\mathrm{OT}(\boldsymbol{h}, \boldsymbol{g}) = \min_{T \in \Pi(\boldsymbol{h},\boldsymbol{g})} \langle \boldsymbol{C}, \boldsymbol{T} \rangle = \min_{T \in \Pi(\boldsymbol{h},\boldsymbol{g})} \sum_{i,j} C_{i,j} T_{i,j}$ → $n \times m$ variables, $n + m$ constraints, $O(n^3)$

We can rewrite the OT problem in a vectorial form

$$\min_{\boldsymbol{t} \geq 0} \quad F(t) = \underbrace{\boldsymbol{c}^\top t}_{\text{vectorized OT cost}}$$

$$\text{such that}$$

$$\boldsymbol{H} t = [\boldsymbol{h}, \boldsymbol{g}]^\top$$

with $\boldsymbol{H}$ a $(n+m) \times nm$ matrix that encodes the constraints

Dual formulation

$$\max_{\boldsymbol{h} \geq 0} \quad D(t) = [\boldsymbol{h}, \boldsymbol{g}]^\top \boldsymbol{h}$$

$$\text{such that}$$

$$\boldsymbol{H}^\top \boldsymbol{h} \leq \boldsymbol{c}$$

with $\boldsymbol{h}$ a $(n+m)$ vector

Computational OT

OT is a linear problem

$$\mathrm{OT}(\boldsymbol{h},\boldsymbol{g}) = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \langle \boldsymbol{C}, \boldsymbol{T}\rangle = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \sum_{i,j} C_{i,j} T_{i,j}$$ → $n \times m$ variables, $n+m$ constraints, $O(n^3)$

We can rewrite the OT problem in a vectorial form

$$\min_{\boldsymbol{t}\geq 0} \quad F(t) = \underbrace{\boldsymbol{c}^\top t}_{\text{vectorized OT cost}}$$

$$\text{such that}$$

$$\boldsymbol{H}t = [\boldsymbol{h},\boldsymbol{g}]^\top$$

with $\boldsymbol{H}$ a $(n+m)\times nm$
matrix that encodes the constraints

Dual formulation

$$\max_{\boldsymbol{h}\geq 0} \quad D(t) = [\boldsymbol{h},\boldsymbol{g}]^\top \boldsymbol{h}$$

$$\text{such that}$$

$$\boxed{\boldsymbol{H}^\top \boldsymbol{h} \leq \boldsymbol{c}}$$

$h[1:n]+h[n+1:m] \leq c_{ij}$

with $\boldsymbol{h}$ a $(n+m)$
vector

Computational OT

Wasserstein on the line

When $c(x, y)$ is a strictly convex function (e.g. quadratic cost), and when $x, y \in \mathbb{R}$, there exists a closed form

$$\forall p \geq 1, W_p^p(\mu_x, \mu_y) = \int_0^1 |F^{-1}(\mu_x) - F^{-1}(\mu_y)|^p du$$

where F^{-1} is the quantile function.

Computational OT

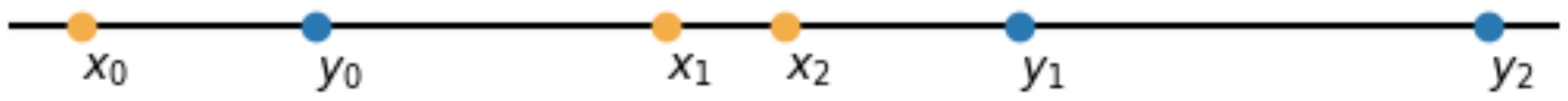
Wasserstein on the line

When $c(x, y)$ is a strictly convex function (e.g. quadratic cost), and when $x, y \in \mathbb{R}$, there exists a closed form

$$\forall p \geq 1, W_p^p(\mu_x, \mu_y) = \int_0^1 |F^{-1}(\mu_x) - F^{-1}(\mu_y)|^p du$$

where $F^{-1}$is the quantile function.

For empirical distributions, it comes down to sorting the 2 distributions



$O(n \log n)$ algorithm

Computational OT

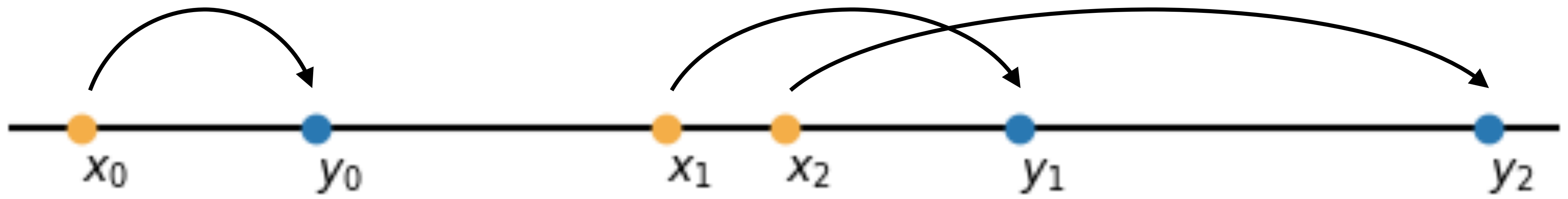
Wasserstein on the line

When $c(x, y)$ is a strictly convex function (e.g. quadratic cost), and when $x, y \in \mathbb{R}$, there exists a closed form

$$\forall p \geq 1, W_p^p(\mu_x, \mu_y) = \int_0^1 |F^{-1}(\mu_x) - F^{-1}(\mu_y)|^p du$$

where F^{-1} is the quantile function.

For empirical distributions, it comes down to sorting the 2 distributions



$O(n \log n)$ algorithm

Computational OT

Wasserstein between Gaussians

When $\mu_x = \mathcal{N}(m_x, \Sigma_x)$ and $\mu_y = \mathcal{N}(m_y, \Sigma_y)$, there also exists a closed form

$$W_2^2(\mu_x, \mu_y) = \|m_x - m_y\|^2 + \mathcal{B}(\Sigma_x, \Sigma_y)^2$$

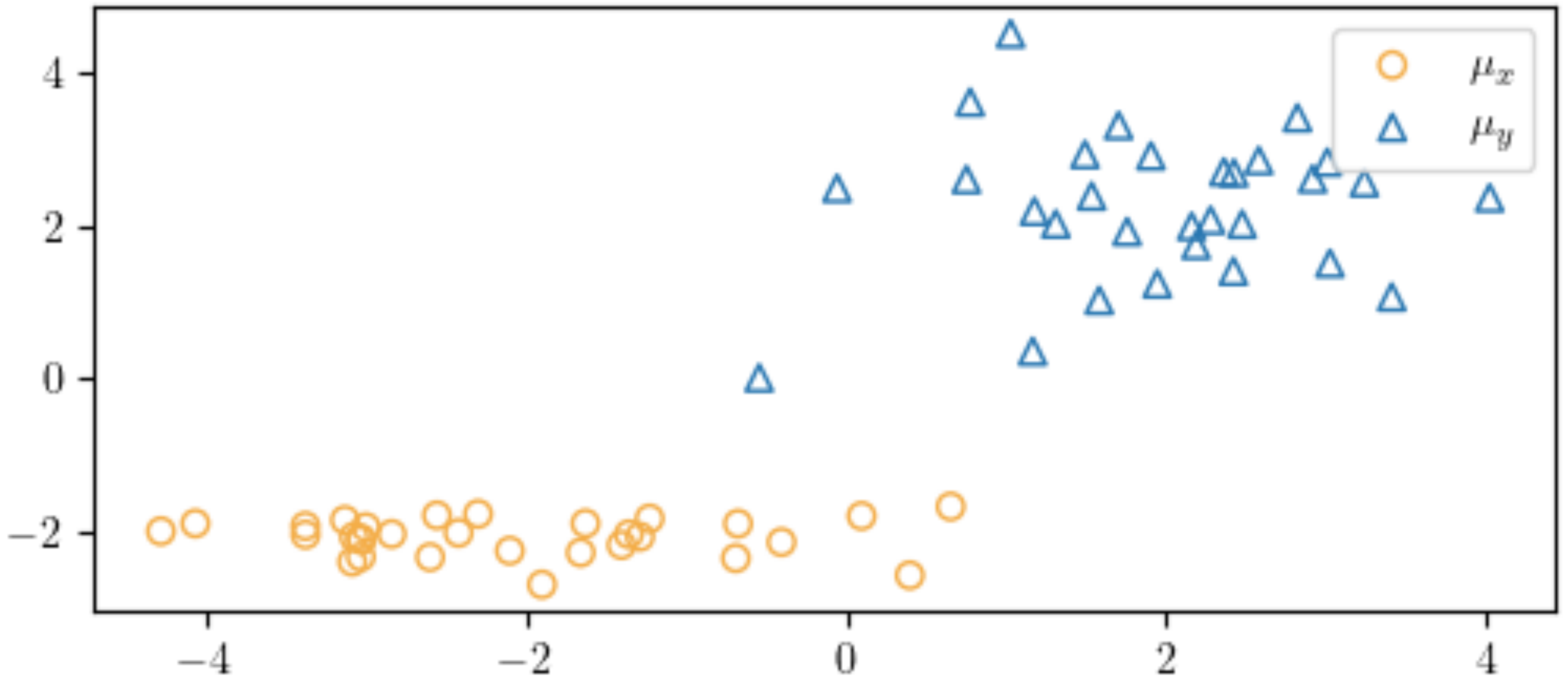
$$\text{with}$$

$$\mathcal{B}(\Sigma_x, \Sigma_y)^2 = \mathrm{trace}(\Sigma_x + \Sigma_y - 2(\Sigma_x^{1/2}\Sigma_y\Sigma_x^{1/2})^{1/2})$$

Computational OT

Sliced Wasserstein

Assume that $n = m$ and $f_i = g_j = \dfrac{1}{n}$ (not compulsory)



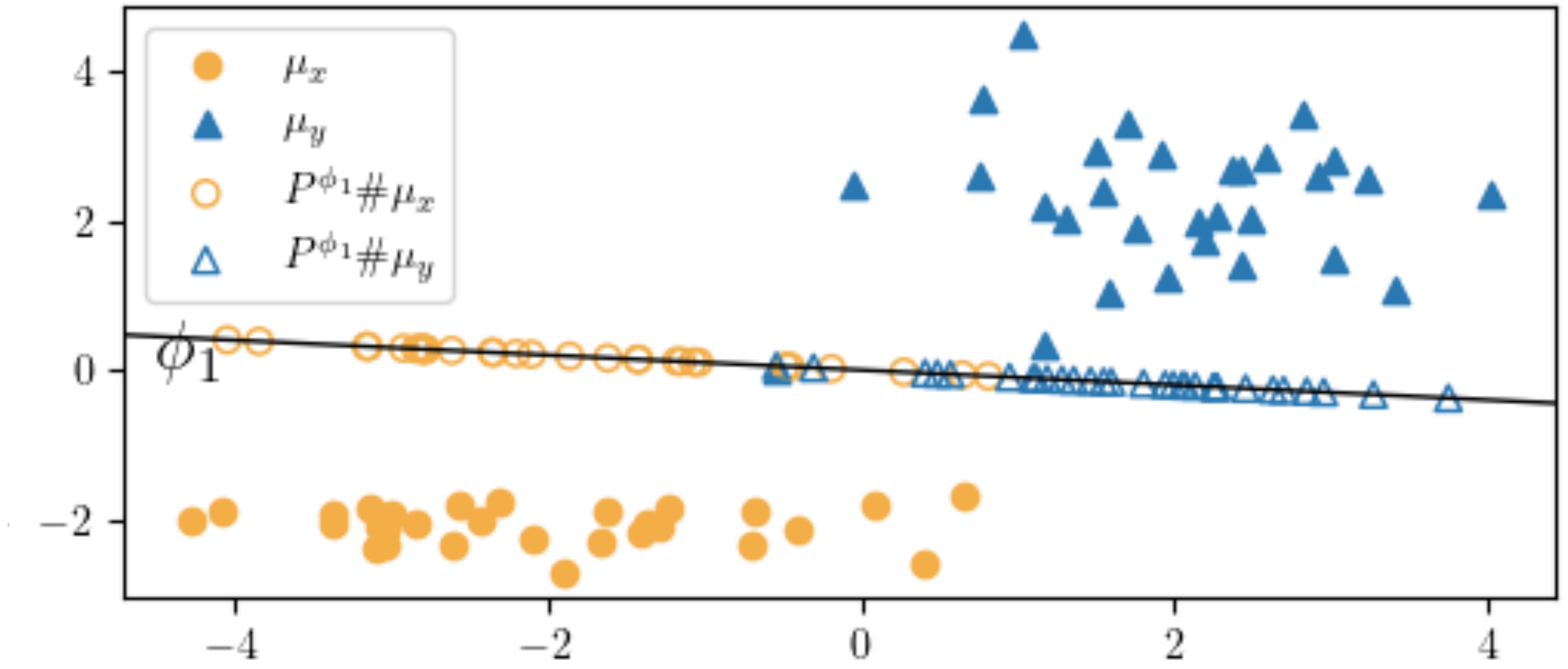
$W_2^2(\mu_x, \mu_y)$?

We look for a (fast) approximation $SW_2^2(\mu_x, \mu_y)$

Computational OT

Sliced Wasserstein

Assume that $n = m$ and $f_i = g_j = \dfrac{1}{n}$ (not compulsory)



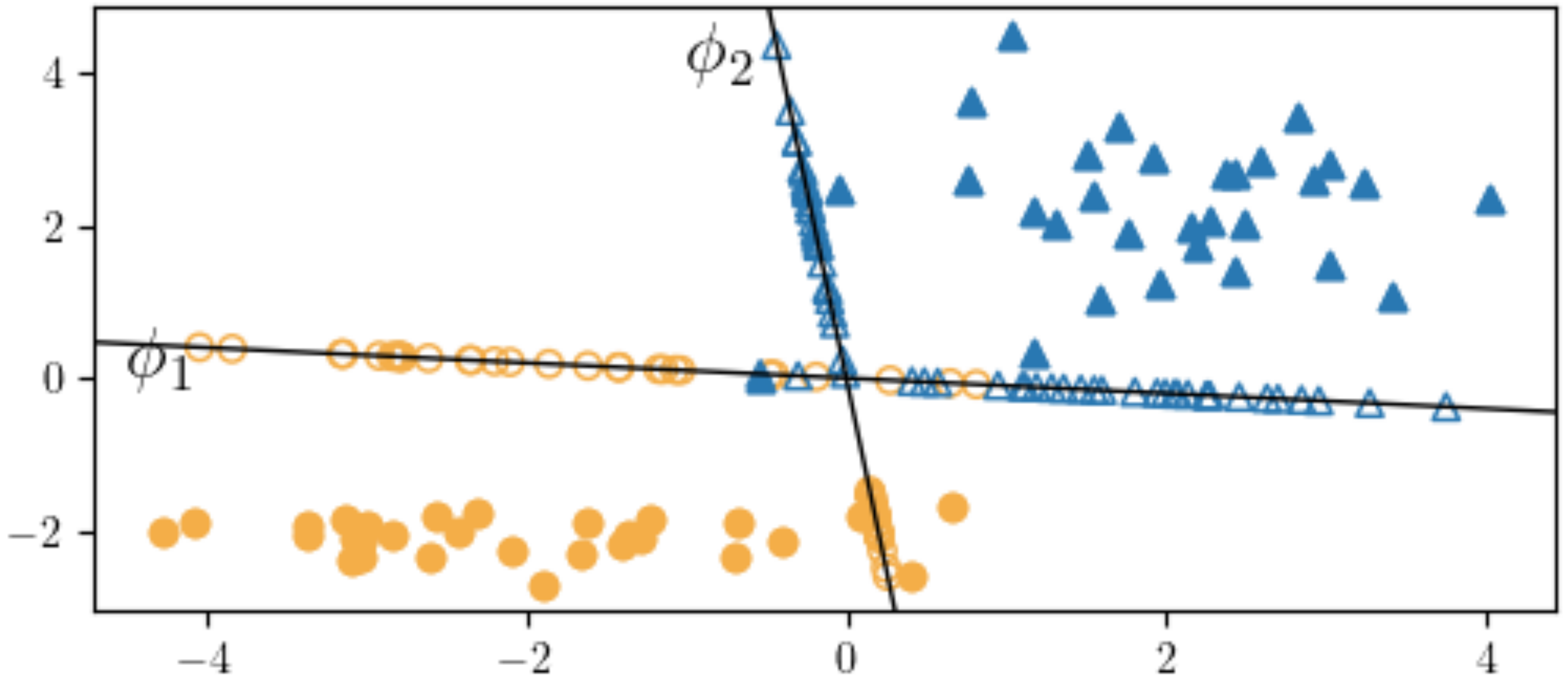
$$P^{\phi}(x) = \langle x, \phi \rangle,\ \ \phi \sim Unif(S^{d-1})$$

$W_p^p(P^{\phi_1}\#\mu_x, P^{\phi_1}\#\mu_y)$ has a closed form (O(n log n))

Computational OT

Sliced Wasserstein

Assume that $n = m$ and $f_i = g_j = \dfrac{1}{n}$ (not compulsory)



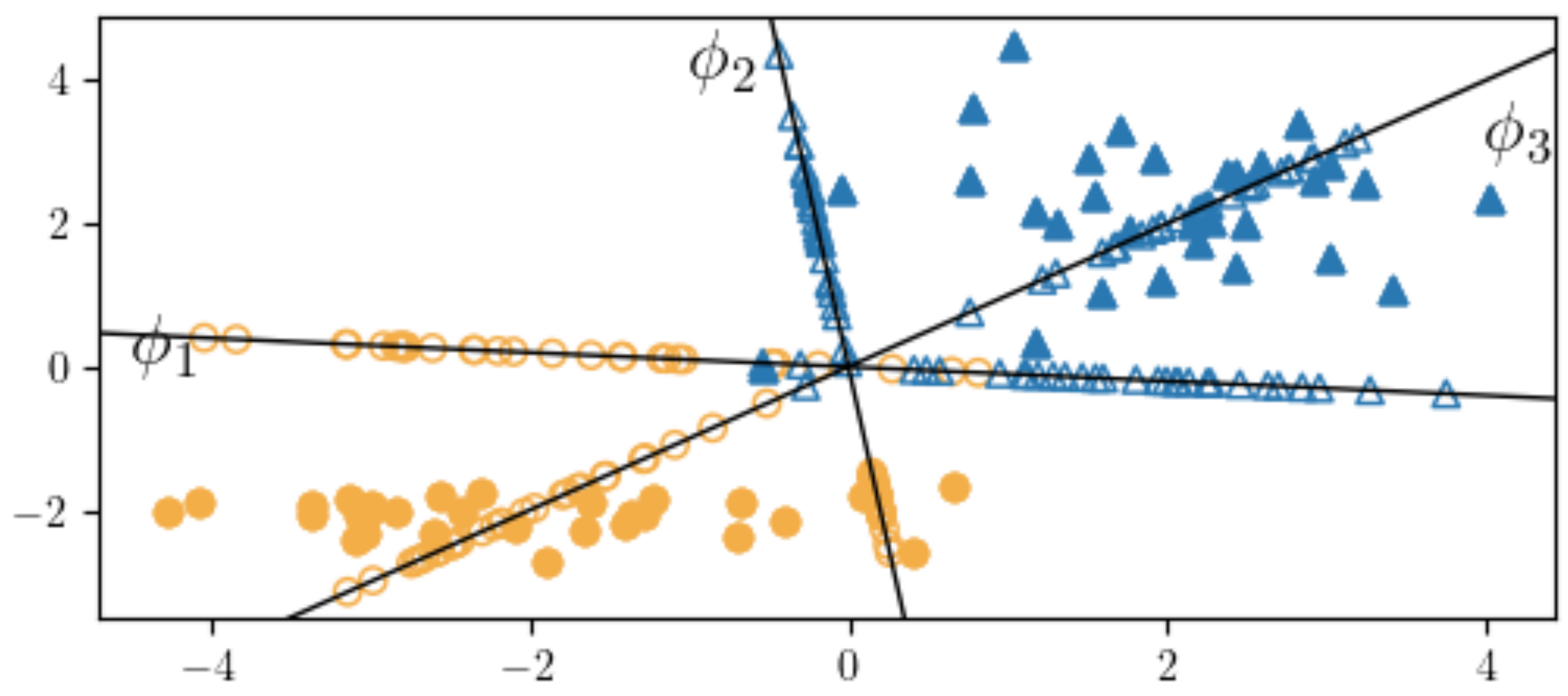
$$P^{\phi}(x) = \langle x, \phi \rangle,\ \ \phi \sim Unif(S^{d-1})$$

$W_p^p(P^{\phi_1}\#\mu_x, P^{\phi_1}\#\mu_y)$ has a closed form (O(n log n))

Computational OT

Sliced Wasserstein

Assume that $n = m$ and $f_i = g_j = \dfrac{1}{n}$ (not compulsory)



$P^{\phi}(x) = \langle x, \phi \rangle,\ \phi \sim Unif(S^{d-1})$ the unit sphere

$W_p^p(P^{\phi_1}\#\mu_x, P^{\phi_1}\#\mu_y)$ has a closed form (O(n log n))

The sliced Wasserstein distance is defined as

$$SW_p^p(\mu_x, \mu_y) = \frac{1}{L}\sum_{\ell=1}^{L} W_p^p(P^{\phi_\ell}\#\mu_x, P^{\phi_\ell}\#\mu_y)$$

Computational OT

Sliced Wasserstein

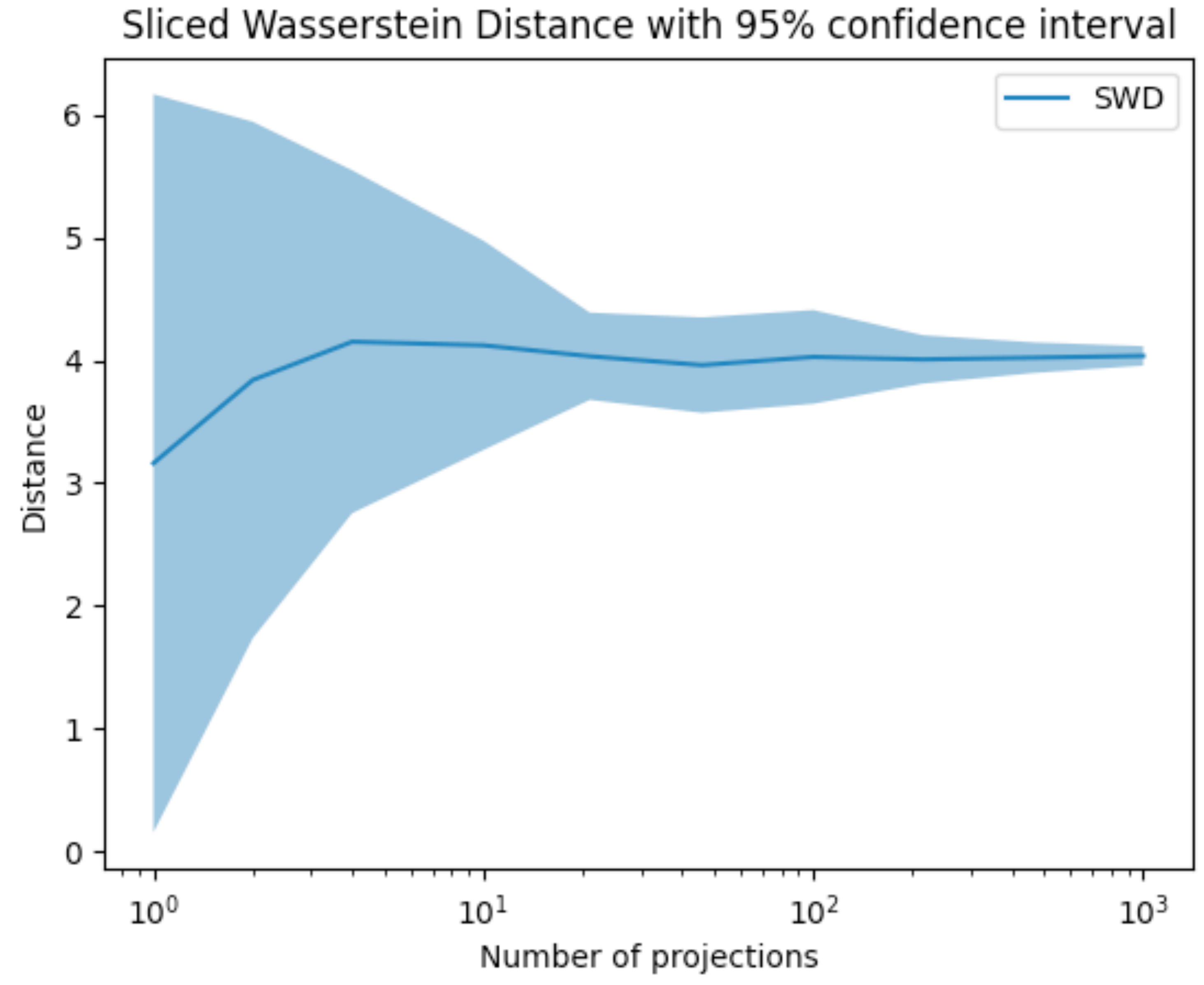
Assume that $n = m$ and $f_i = g_j = \dfrac{1}{n}$ (not compulsory)

The sliced Wasserstein distance is defined as [Rabin 2012]

$$SW_p^p(\mu_x, \mu_y) = \frac{1}{L}\sum_{\ell=1}^{L} W_p^p(P^{\phi_\ell}\#\mu_x, P^{\phi_\ell}\#\mu_y)$$

Properties

1. It is a distance
2. Similar topological properties than Wasserstein
3. Computation in $O(Ln\log n)$



Source: POT

Computational OT

Sliced Wasserstein

Assume that $n = m$ and $f_i = g_j = \frac{1}{n}$ (not compulsory)

The sliced Wasserstein distance is defined as [Rabin 2012]

$$SW_p^p(\mu_x, \mu_y) = \frac{1}{L}\sum_{\ell=1}^{L} W_p^p(P^{\phi_\ell}\#\mu_x, P^{\phi_\ell}\#\mu_y)$$

Properties

1. It is a distance
2. Similar topological properties than Wasserstein
3. Computation in $O(Ln \log n)$

But

- Does not provide the transport plan
- Can not be optimized
- Issues when d is large

Computational OT

Sliced Wasserstein

The sliced Wasserstein distance is defined as

$$SW_p^p(\mu_x, \mu_y) = \frac{1}{L}\sum_{\ell=1}^{L} W_p^p(P^{\phi_\ell}\#\mu_x, P^{\phi_\ell}\#\mu_y)$$

Several variants exists

1. In different geometric spaces (sphere, hyperbolic spaces)
2. With projections onto curves, different samplings of the line

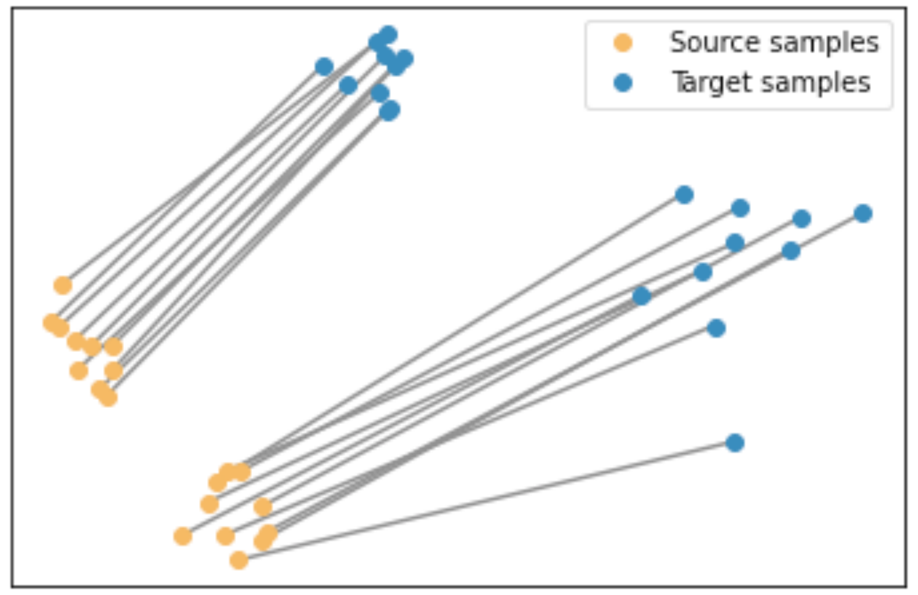
An example: sliced Wasserstein Generalized Geodesic (SWGG) [Mahey 2023]

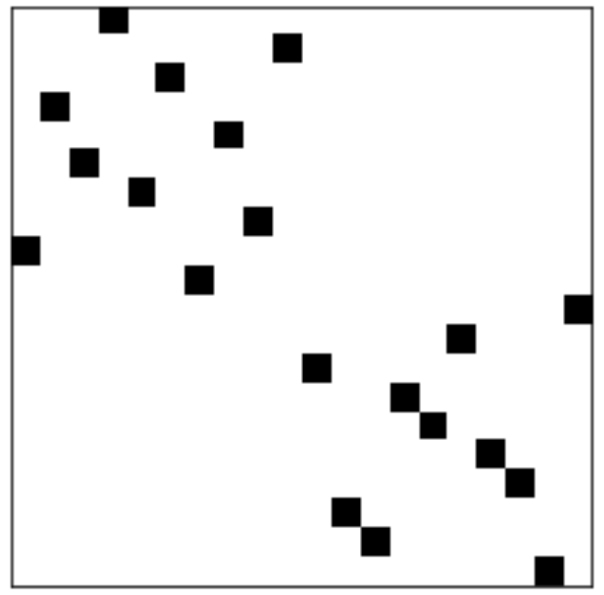
$$\min -\mathrm{SWGG}_2^2(\mu_x, \mu_y) = \min_{\phi \in S^{d-1}} \left(\frac{1}{n} \sum_{i=1}^{n} \| x_{\sigma_{\phi(i)}} - y_{\tau_{\phi(i)}} \|_2^2 \right)$$

Computational OT

Regularized OT

$$\mathrm{OT}(\boldsymbol{h},\boldsymbol{g}) = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \langle \boldsymbol{C},\boldsymbol{T}\rangle = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \sum_{i,j} C_{i,j}T_{i,j}$$ → $n \times m$ variables, $n+m$ constraints, $O(n^3)$

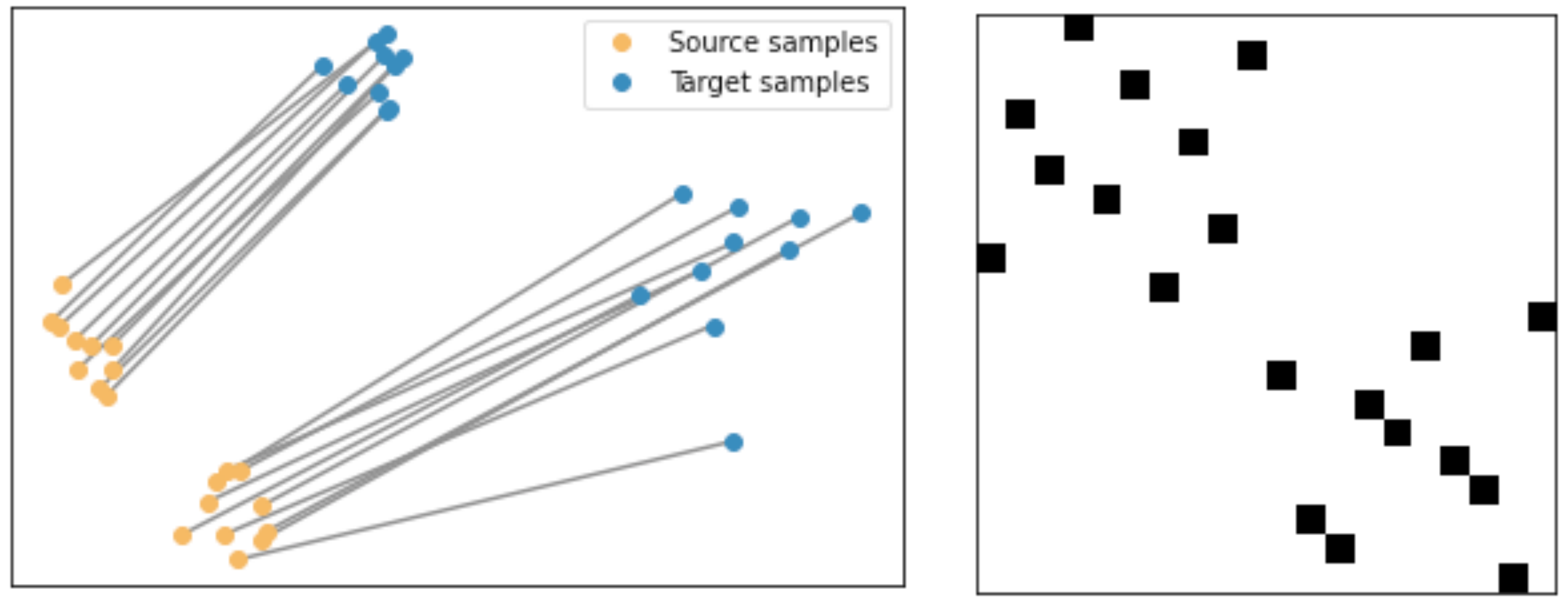




Computational OT

Regularized OT

$\mathrm{OT}(\boldsymbol{h},\boldsymbol{g}) = \min\limits_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \langle \boldsymbol{C}, \boldsymbol{T}\rangle = \min\limits_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \sum\limits_{i,j} C_{i,j}T_{i,j}$ → $n \times m$ variables, $n+m$ constraints, $O(n^3)$



Regularization of OT $\quad \min\limits_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \langle \boldsymbol{C}, \boldsymbol{T}\rangle + \lambda\Omega(T)$

Why?

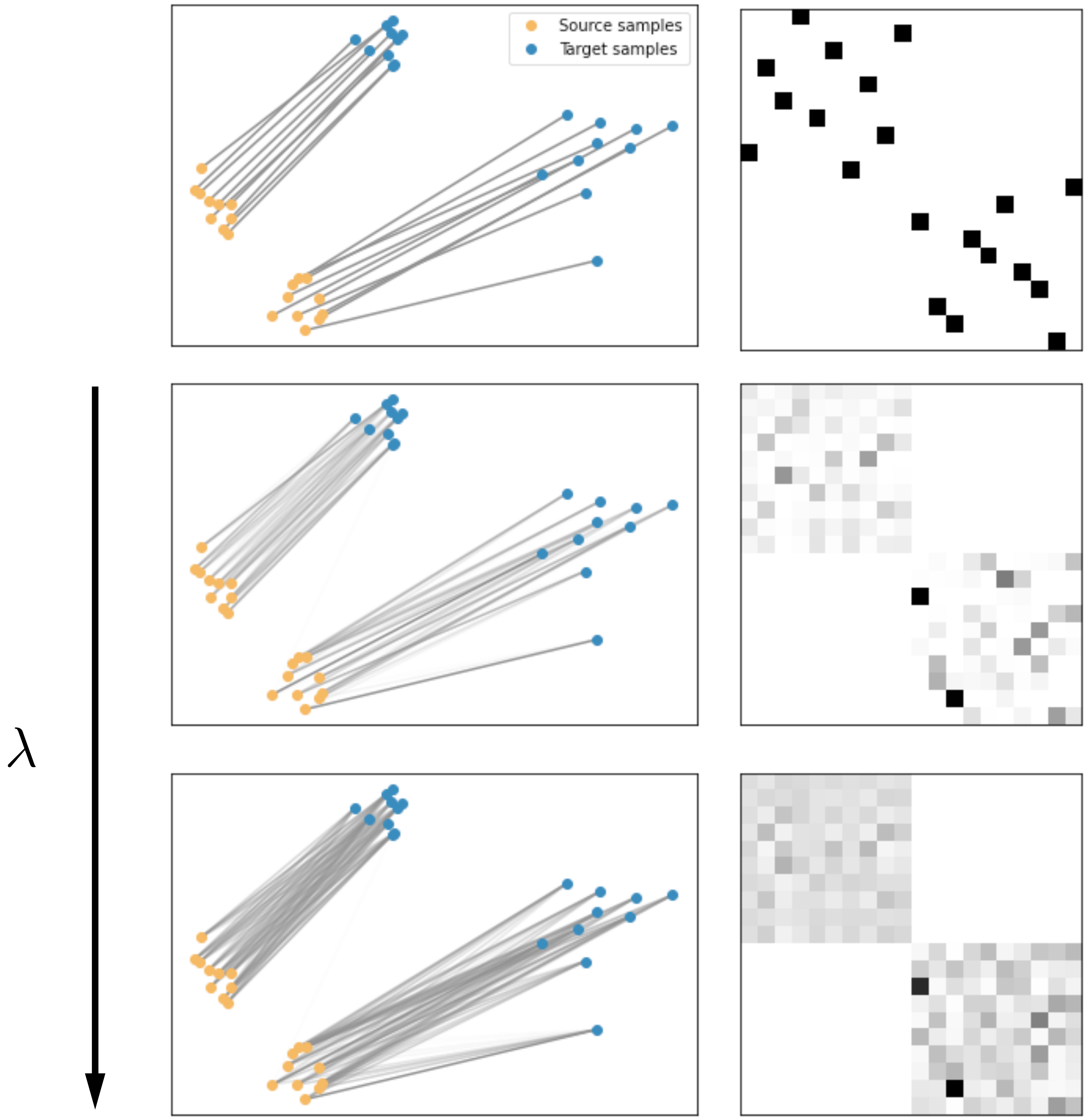
- define fast algorithms for solving the problem
- better defined problem
- encode prior knowledge on the data
  (e.g. group sparsity constraint)

Computational OT

Regularized OT

$\mathrm{OT}(\boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \in \Pi(\boldsymbol{h},\boldsymbol{g})} \langle \boldsymbol{C}, \boldsymbol{T} \rangle = \min_{\boldsymbol{T} \in \Pi(\boldsymbol{h},\boldsymbol{g})} \sum_{i,j} C_{i,j} T_{i,j}$ → $n \times m$ variables, $n + m$ constraints, $O(n^3)$

Regularization of OT $\quad \min_{\boldsymbol{T} \in \Pi(\boldsymbol{h},\boldsymbol{g})} \langle \boldsymbol{C}, \boldsymbol{T} \rangle + \lambda \Omega(T)$



Entropic regularization [Cuturi 2013]

$$\Omega(T) = \sum_{i,j} T_{ij} (\log T_{ij} - 1)$$

λ controls the « smoothing » of the solution

$\lambda \rightarrow 0$: we recover the unconstrained solution

$\lambda \rightarrow \infty$: $T = \boldsymbol{h}^T \boldsymbol{C} \boldsymbol{g}$

Computational OT

Regularized OT

$$\mathrm{OT}(\boldsymbol{h},\boldsymbol{g}) = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \langle \boldsymbol{C}, \boldsymbol{T}\rangle = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \sum_{i,j} C_{i,j} T_{i,j}$$ → $n \times m$ variables, $n + m$ constraints, $O(n^3)$

Regularization of OT $$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \langle \boldsymbol{C}, \boldsymbol{T}\rangle + \lambda\Omega(T)$$

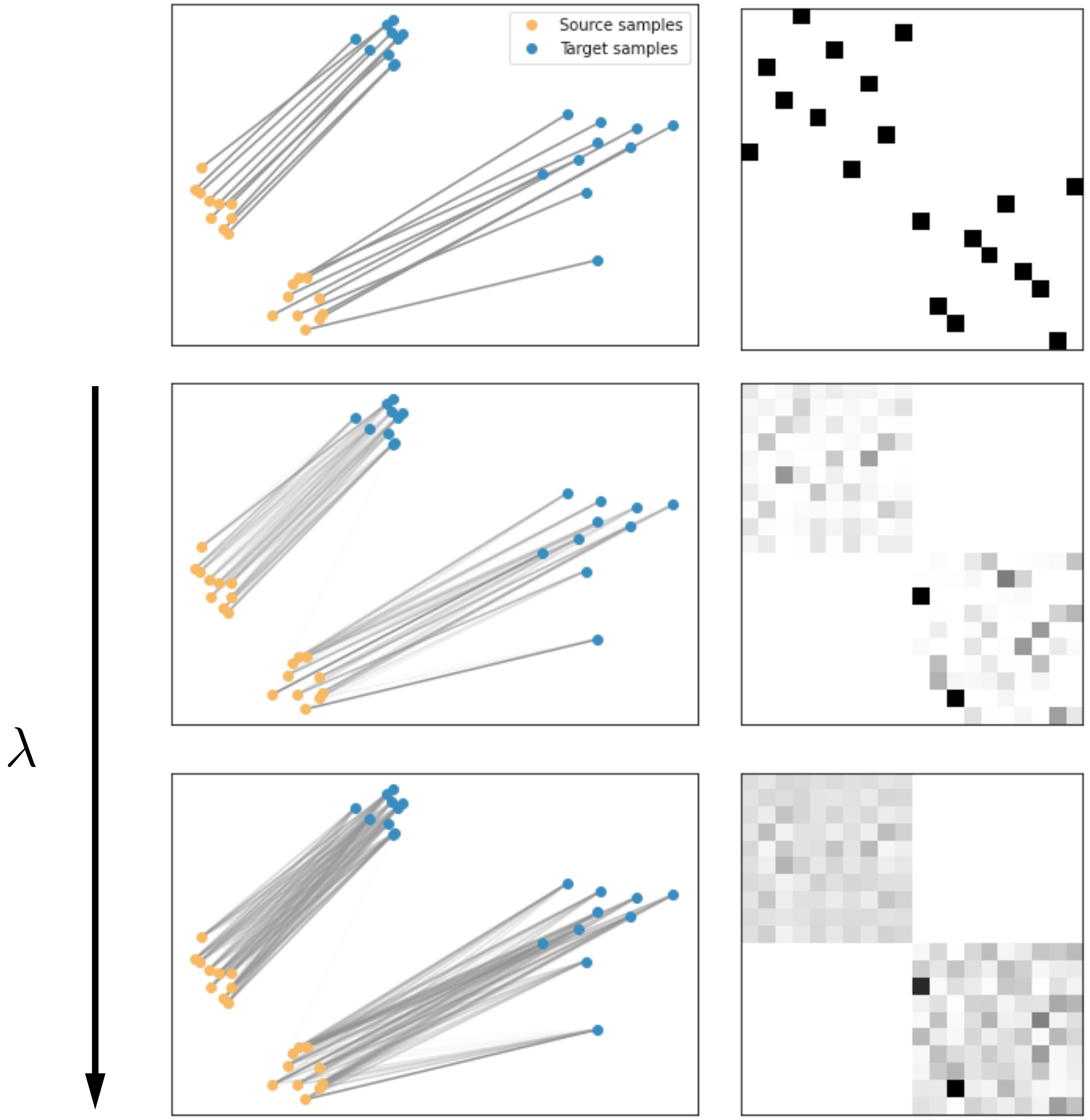
Entropic regularization [Cuturi 2013]

$$\Omega(T) = \sum_{i,j} T_{ij}(\log T_{ij} - 1)$$

Iterative algorithm with deterministic updates

$$T_{\lambda}^{(k+1)} = \mathrm{diag}\left(\boldsymbol{u}^{(k)}\right) \exp\left(-\frac{\boldsymbol{C}}{\lambda}\right) \mathrm{diag}\left(\boldsymbol{v}^{(k)}\right)$$

Sinkhorn theorem: $\boldsymbol{u}^{(k)}$ and $\boldsymbol{v}^{(k)}$ exist and are unique



Computational OT

Regularized OT

Entropic regularization of OT $T_\lambda = \min_{\boldsymbol{T} \in \Pi(\boldsymbol{h}, \boldsymbol{g})} \langle \boldsymbol{C}, \boldsymbol{T} \rangle + \lambda \sum_{i,j} T_{ij} (\log T_{ij} - 1)$

Pro

- Complexity $O(n^2)$
- Smoothes the coupling
- Amenable to optimization, GPU friendly

Cons

- Smoothes the coupling
- Parameter to tune, lots of iterations for small λ
- Is not a distance: $\mathrm{OT}_\lambda(h, h) = \langle \boldsymbol{C}, \boldsymbol{T_\lambda} \rangle \neq 0$ ⇒ Sinkhorn divergence [Feydy 2019]

$$S_\lambda(h, g) = \mathrm{OT}_\lambda(h, g) - \frac{1}{2} \mathrm{OT}_\lambda(h, h) - \frac{1}{2} \mathrm{OT}_\lambda(g, g)$$

Some challenges of OT

Scalability of the algorithms

Unstable, not robust to outliers

Needs a common metric space

The Kantorovich relaxation aims to solve

$$\mathrm{OT}(\boldsymbol{h},\boldsymbol{g}) = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \langle \boldsymbol{C},\boldsymbol{T}\rangle = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \sum_{i,j} C_{i,j}T_{i,j}$$

with the constraint $\boldsymbol{\Pi}(\boldsymbol{h},\boldsymbol{g}) = \left\{\boldsymbol{T}\in\mathbb{R}_+^{n\times m} | \boldsymbol{T}1_m = \boldsymbol{h}, \boldsymbol{T}^\top 1_n = \boldsymbol{g}\right\}.$

Some challenges of OT

Scalability of the algorithms

Unstable, not robust to outliers

Needs a common metric space

The Kantorovich relaxation aims to solve

$$\mathrm{OT}(\boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \in \Pi(\boldsymbol{h}, \boldsymbol{g})} \langle \boldsymbol{C}, \boldsymbol{T} \rangle = \min_{\boldsymbol{T} \in \Pi(\boldsymbol{h}, \boldsymbol{g})} \sum_{i,j} C_{i,j} T_{i,j}$$

with the constraint $\boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g}) = \left\{ \boldsymbol{T} \in \mathbb{R}_+^{n \times m} | \boldsymbol{T} 1_m = \boldsymbol{h}, \boldsymbol{T}^\top 1_n = \boldsymbol{g} \right\}.$

Linear problem with $n \times m$ variables, $n + m$ constraints

Some challenges of OT

Scalability of the algorithms

Unstable, not robust to outliers

Needs a common metric space

The Kantorovich relaxation aims to solve

$$\mathrm{OT}(\boldsymbol{h},\boldsymbol{g}) = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \langle \boldsymbol{C}, \boldsymbol{T}\rangle = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \sum_{i,j} C_{i,j}T_{i,j}$$

with the constraint $\boldsymbol{\Pi}(\boldsymbol{h},\boldsymbol{g}) = \left\{\boldsymbol{T}\in\mathbb{R}_+^{n\times m} | \boldsymbol{T}1_m = \boldsymbol{h}, \boldsymbol{T}^\top 1_n = \boldsymbol{g}\right\}.$

Global optimization problem with constraints $\boldsymbol{T}1_m = \boldsymbol{h}, \boldsymbol{T}^\top 1_n = \boldsymbol{g}$

Some challenges of OT

Scalability of the algorithms

Unstable, not robust to outliers

Needs a common metric space

The Kantorovitch relaxation aims to solve

$$\mathrm{OT}(\boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \in \Pi(\boldsymbol{h}, \boldsymbol{g})} \langle \boldsymbol{C}, \boldsymbol{T} \rangle = \min_{\boldsymbol{T} \in \Pi(\boldsymbol{h}, \boldsymbol{g})} \sum_{i,j} C_{i,j} T_{i,j}$$

with the constraint $\boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g}) = \left\{ \boldsymbol{T} \in \mathbb{R}_+^{n \times m} | \boldsymbol{T} 1_m = \boldsymbol{h}, \boldsymbol{T}^\top 1_n = \boldsymbol{g} \right\}.$

Cost $c(x, y)$

Outline

1. History and basics of optimal transport

2. Wasserstein distances

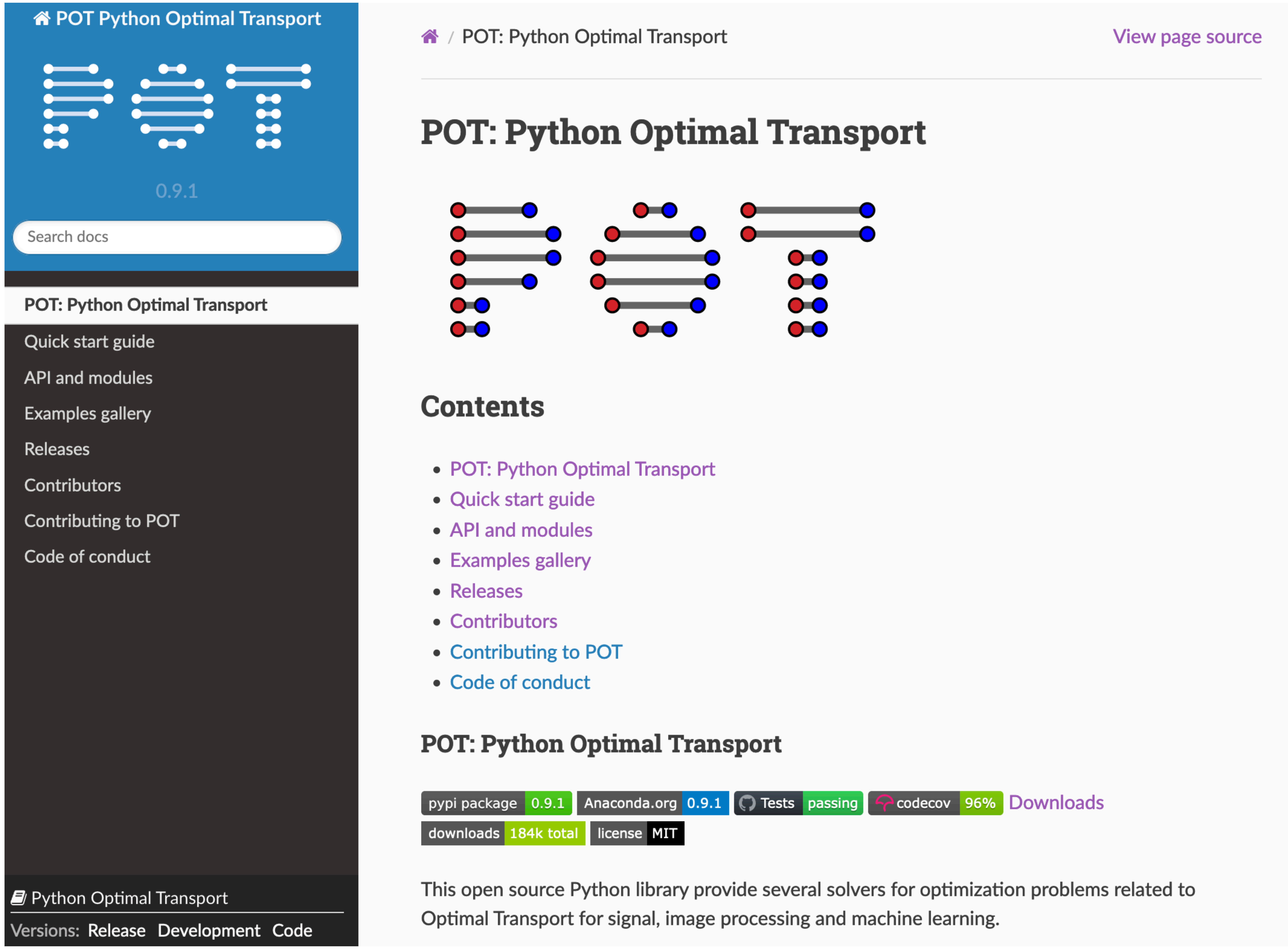
3. Computational OT

Practical session (with POT toolbox)

4. Variants of OT : unbalanced OT and Gromov-Wasserstein

5. Some applications of OT in machine learning

POT toolbox



POT Python Optimal Transport

0.9.1

Search docs

POT: Python Optimal Transport

Quick start guide

API and modules

Examples gallery

Releases

Contributors

Contributing to POT

Code of conduct

Python Optimal Transport

Versions: Release Development Code

POT: Python Optimal Transport

View page source

POT: Python Optimal Transport

Contents

- POT: Python Optimal Transport
- Quick start guide
- API and modules
- Examples gallery
- Releases
- Contributors
- Contributing to POT
- Code of conduct

POT: Python Optimal Transport

pypi package 0.9.1 | Anaconda.org 0.9.1 | Tests passing | codecov 96% | Downloads

downloads 184k total | license MIT

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

https://pythonot.github.io/

Outline

1. History and basics of optimal transport

2. Wasserstein distances

3. Computational OT

Practical session (with POT toolbox)

4. Variants of OT: unbalanced OT and Gromov-Wasserstein

5. Some applications of OT in data science / machine learning

Recall

Aim of OT: find a « meaningful » measure of distance between probability measures

Recall

Aim of OT: find a « meaningful » measure of distance between probability measures

Monge

Minimize the overall transportation cost

T is the transport map (may not exist)

$$\inf_{T\#\mu_s=\mu_t} \int c(x, \boxed{T(x)})\mu_s(x)dx$$

Recall

Aim of OT: find a « meaningful » measure of distance between probability measures

Monge

Minimize the overall transportation cost

T is the transport map (may not exist)

$$\inf_{T\#\mu_s=\mu_t} \int c(x,T(x))\mu_s(x)dx$$

$$\mathrm{OT}(\boldsymbol{h},\boldsymbol{g}) = \min_{T\in\Pi(\boldsymbol{h},\boldsymbol{g})} \langle \boldsymbol{C}, \boldsymbol{T}\rangle = \min_{T\in\Pi(\boldsymbol{h},\boldsymbol{g})} \sum_{i,j} C_{i,j}T_{i,j}$$

T is an OT plan or coupling matrix (always exists)

with the constraint $\boldsymbol{\Pi}(\boldsymbol{h},\boldsymbol{g}) = \left\{\boldsymbol{T} \in \mathbb{R}_+^{n\times m} | \boldsymbol{T}1_m = \boldsymbol{h}, \boldsymbol{T}^\top 1_n = \boldsymbol{g}\right\}.$

Kantorovich

Recall

Aim of OT: find a « meaningful » measure of distance between probability measures

Monge

Minimize the overall transportation cost

T is the transport map (may not exist)

$$\inf_{T\#\mu_s=\mu_t} \int c(x, T(x))\mu_s(x)dx$$

$$\mathrm{OT}(\boldsymbol{h},\boldsymbol{g}) = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \langle \boldsymbol{C}, \boldsymbol{T}\rangle = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \sum_{i,j} C_{i,j}T_{i,j}$$

T is an OT plan or coupling matrix (always exists)

with the constraint $\mathbf{\Pi}(\boldsymbol{h},\boldsymbol{g}) = \left\{ \boldsymbol{T}\in\mathbb{R}_+^{n\times m} | \boldsymbol{T}1_m = \boldsymbol{h}, \boldsymbol{T}^\top 1_n = \boldsymbol{g}\right\}.$

Kantorovich

Linear problem with linear constraints: $O(n^3)$ complexity

Entropic-regularized OT, $\min_{\boldsymbol{T}\in\Pi(\boldsymbol{h},\boldsymbol{g})} \langle \boldsymbol{C}, \boldsymbol{T}\rangle + \lambda\Omega(T)$ with $\Omega(T) = \sum_{i,j} T_{ij}(\log T_{ij} - 1)$, $O(n^2)$ complexity

Unbalanced Optimal Transport

Relaxing the set of constraints

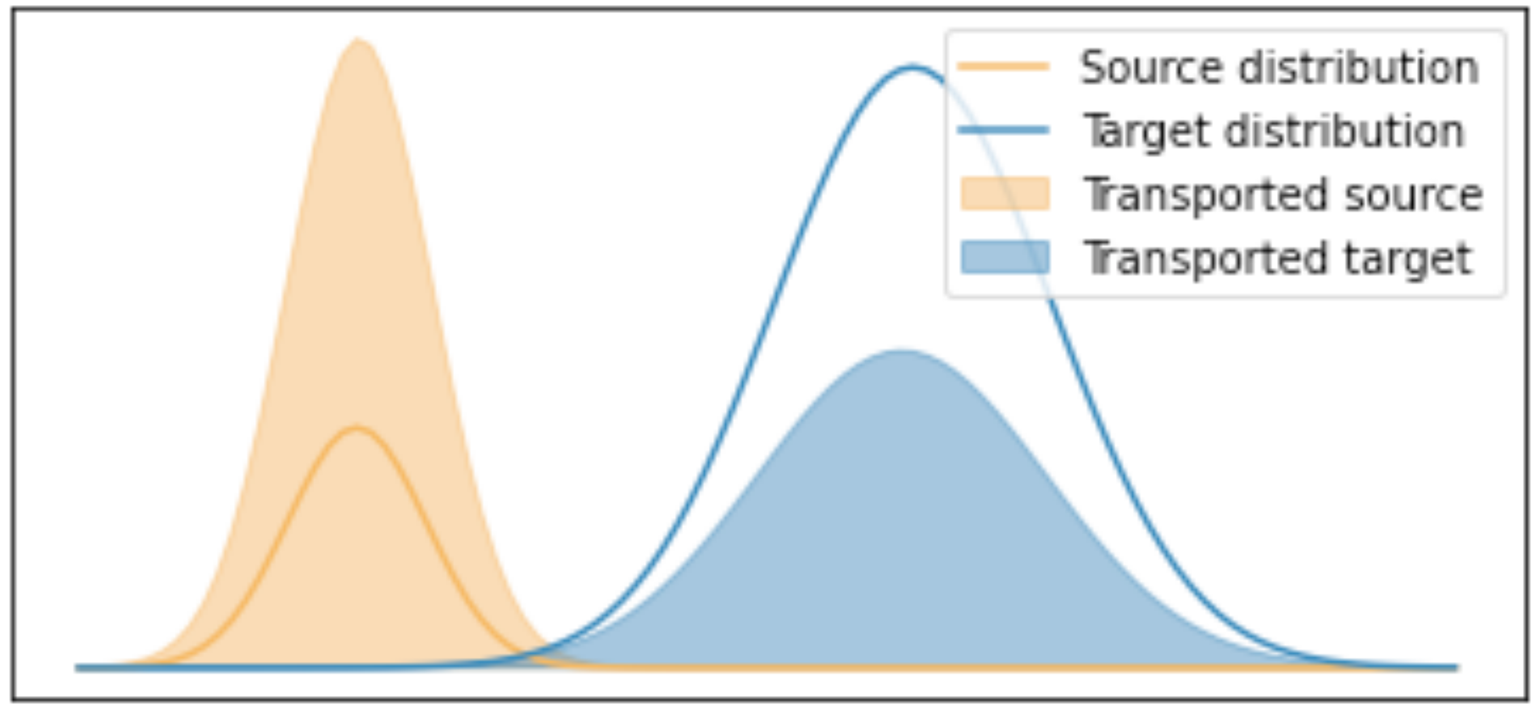
$$\mathbf{\Pi}(\boldsymbol{h}, \boldsymbol{g}) = \left\{\boldsymbol{T} \in \mathbb{R}_+^{n \times m} | \boldsymbol{T} 1_m = \boldsymbol{h}, \boldsymbol{T}^\top 1_n = \boldsymbol{g}\right\}.$$

Unbalanced Optimal Transport

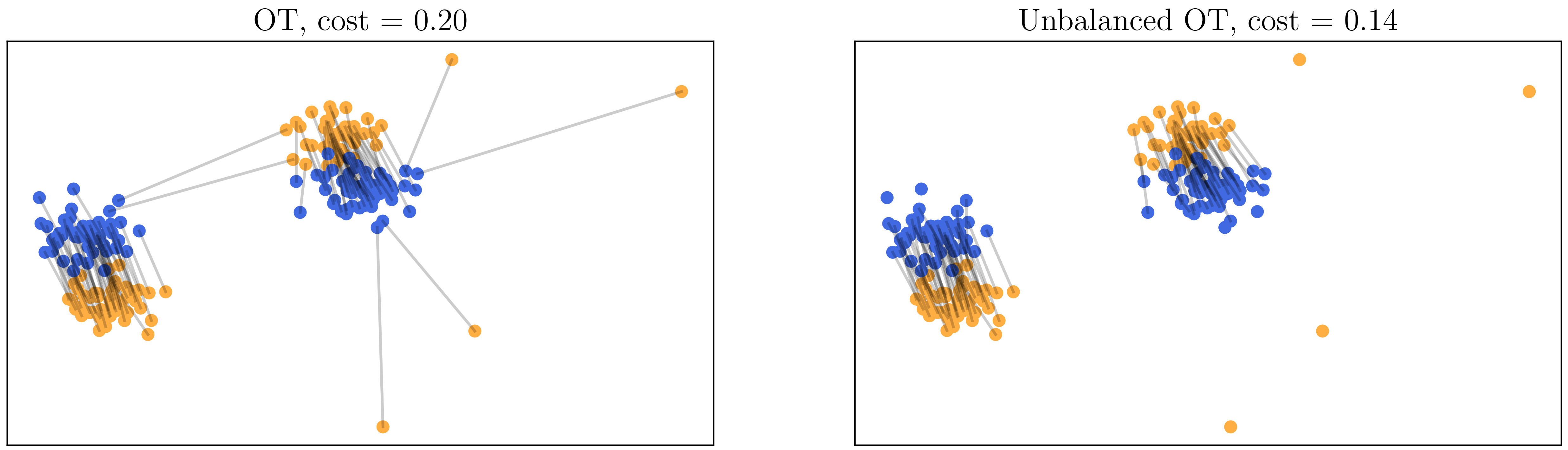
Relaxing the set of constraints

$$\mathbf{\Pi}(\boldsymbol{h}, \boldsymbol{g}) = \left\{ \boldsymbol{T} \in \mathbb{R}_+^{n \times m} | \boldsymbol{T} 1_m = \boldsymbol{h}, \boldsymbol{T}^\top 1_n = \boldsymbol{g} \right\}.$$

How to work with unnormalized histograms?



How to deal with outliers or noisy samples ?

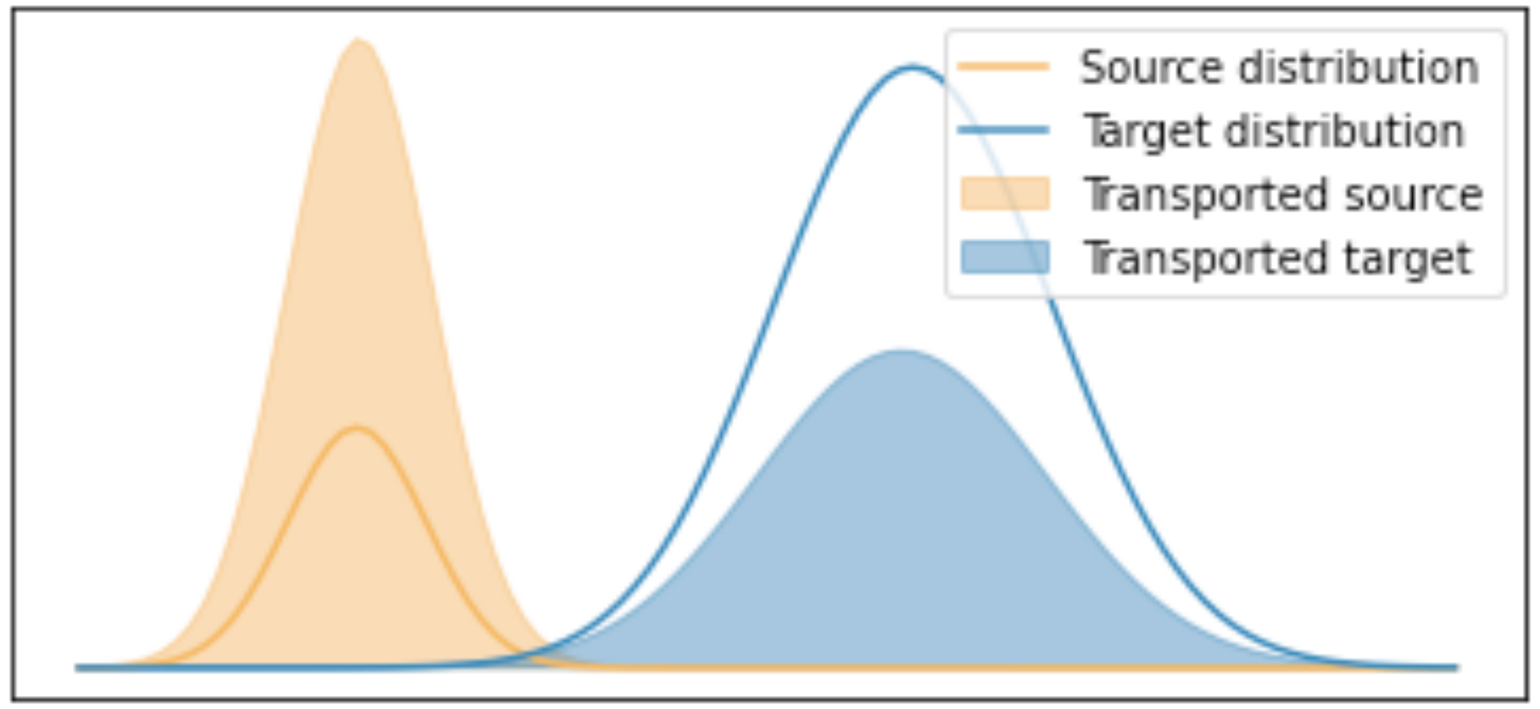


Unbalanced Optimal Transport

Relaxing the set of constraints

$$\mathbf{\Pi}(\boldsymbol{h}, \boldsymbol{g}) = \left\{ \boldsymbol{T} \in \mathbb{R}_+^{n\times m} | \boldsymbol{T}1_m = \boldsymbol{h}, \boldsymbol{T}^\top 1_n = \boldsymbol{g} \right\}.$$

How to work with unnormalized histograms?



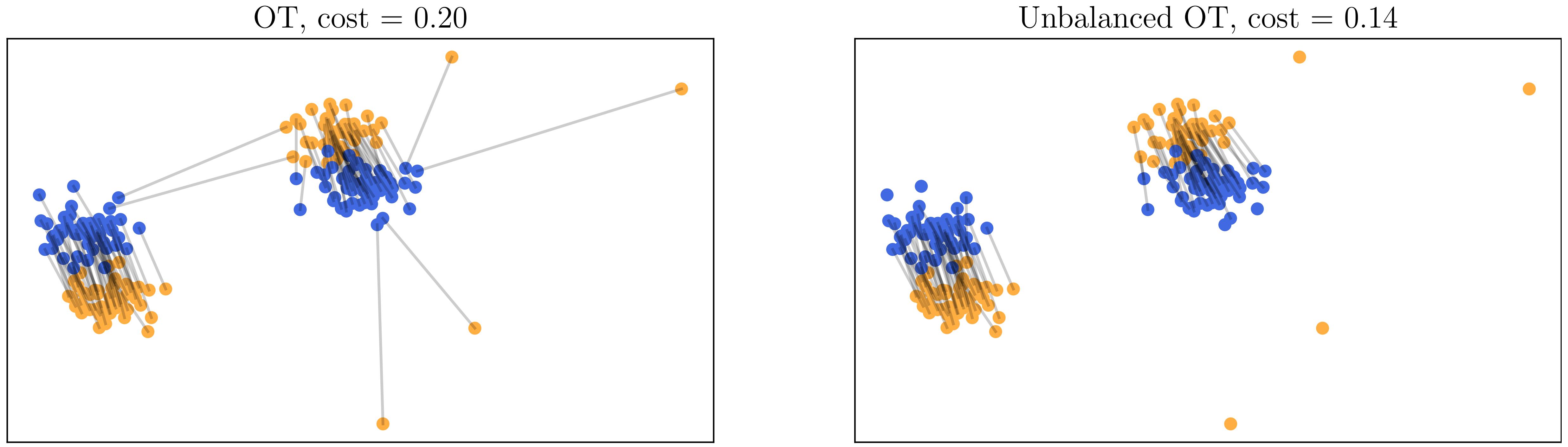
→ Regularize or relax the set of constraints

Unbalanced optimal transport [Benamou 2003]

$$\mathrm{UOT}_\lambda(\boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T}\geq 0} \langle \boldsymbol{C}, \boldsymbol{T} \rangle + \lambda \left(D_\varphi(\boldsymbol{T}1_m, \boldsymbol{h}) + D_\varphi(\boldsymbol{T}^\top 1_n, \boldsymbol{g}) \right)$$

with D_φ a divergence

How to deal with outliers or noisy samples ?

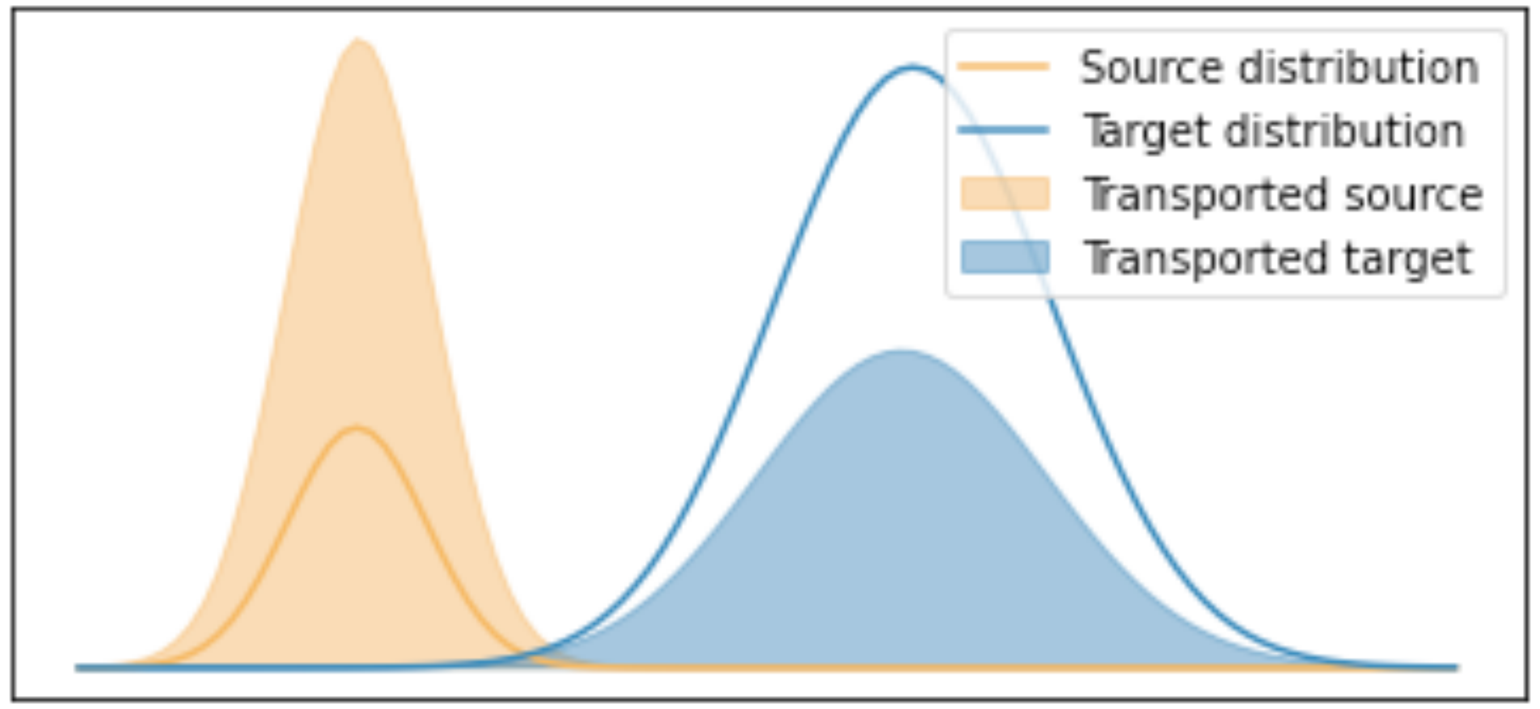


Unbalanced Optimal Transport

Relaxing the set of constraints

$$\mathbf{\Pi}(\boldsymbol{h}, \boldsymbol{g}) = \left\{ T \in \mathbb{R}_+^{n \times m} | T\mathbb{1}_m = \boldsymbol{h}, T^\top \mathbb{1}_n = \boldsymbol{g} \right\}.$$

How to work with unnormalized histograms?



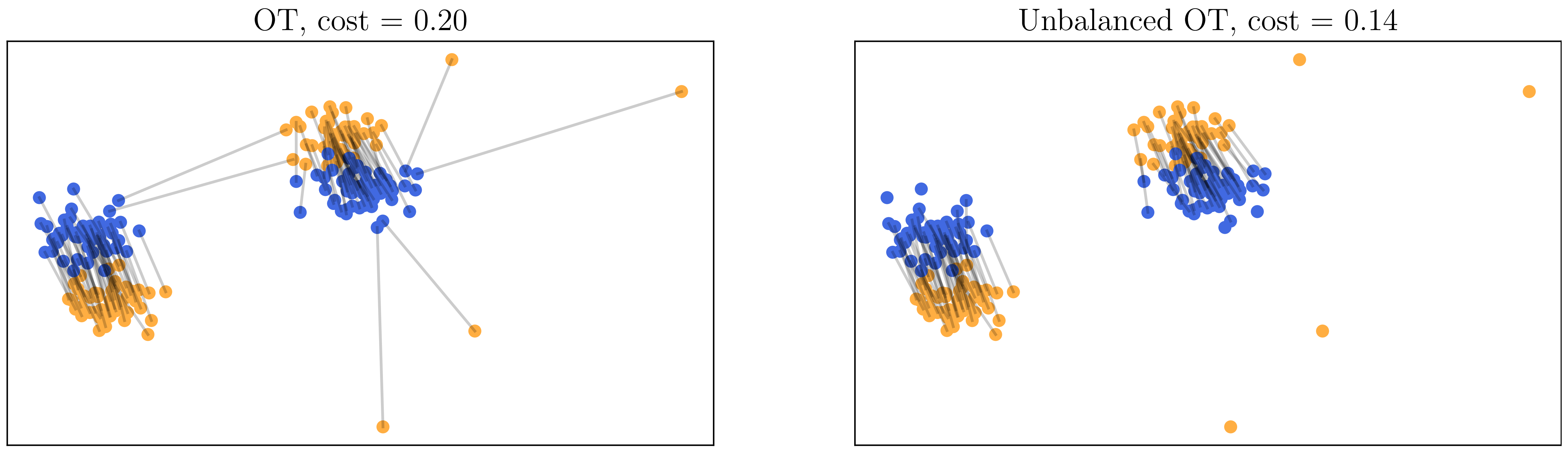
→ Regularize or relax the set of constraints

Unbalanced optimal transport [Benamou 2003]

$$\mathrm{UOT}_\lambda(\boldsymbol{h}, \boldsymbol{g}) = \min_{T \geq 0} \langle \boldsymbol{C}, T \rangle + \lambda \Big(D_\varphi(T\mathbb{1}_m, \boldsymbol{h}) + D_\varphi(T^\top \mathbb{1}_n, \boldsymbol{g}) \Big)$$

with D_φ a divergence

How to deal with outliers or noisy samples ?

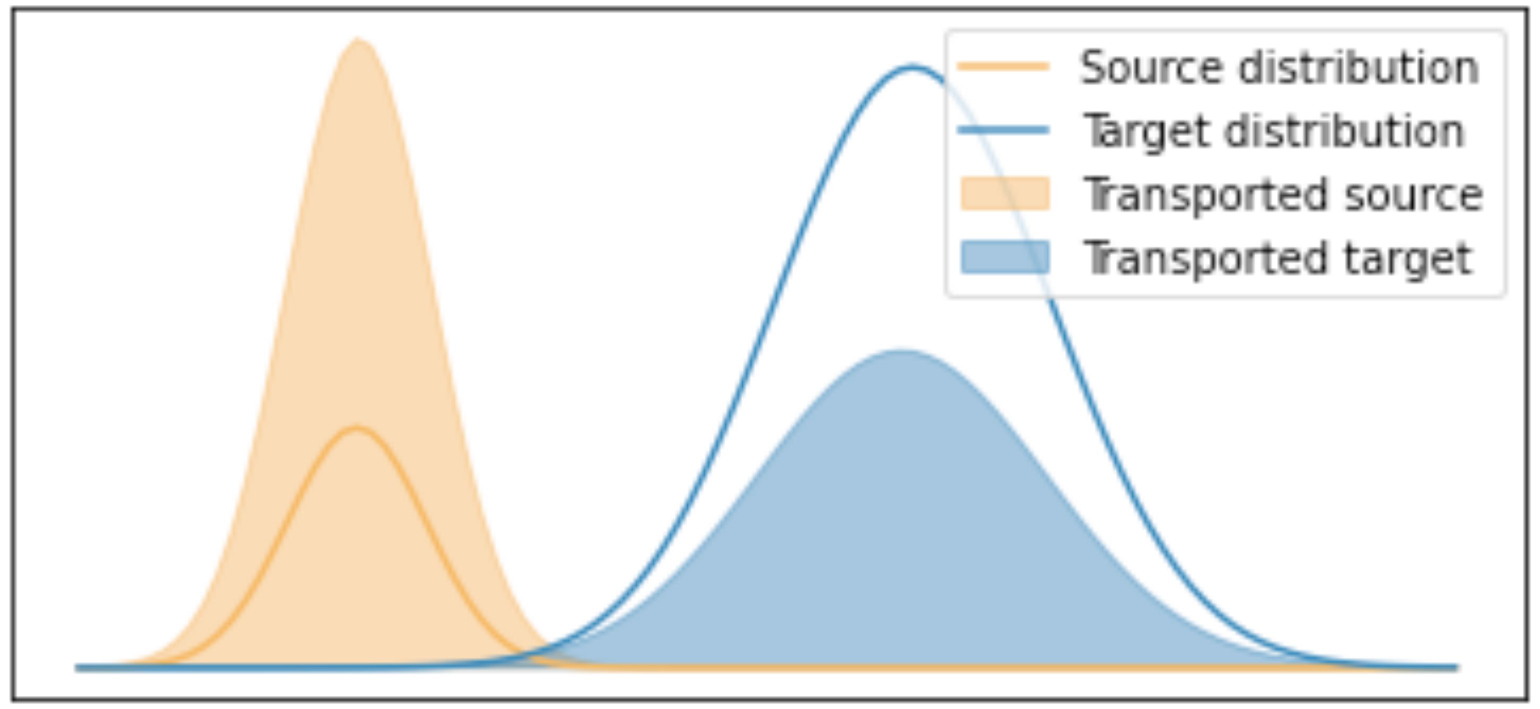


Unbalanced Optimal Transport

Relaxing the set of constraints

$$\mathbf{\Pi}(\boldsymbol{h},\boldsymbol{g}) = \left\{\boldsymbol{T} \in \mathbb{R}_+^{n\times m} | \boldsymbol{T}1_m = \boldsymbol{h}, \boldsymbol{T}^\top 1_n = \boldsymbol{g}\right\}.$$

How to work with unnormalized histograms?



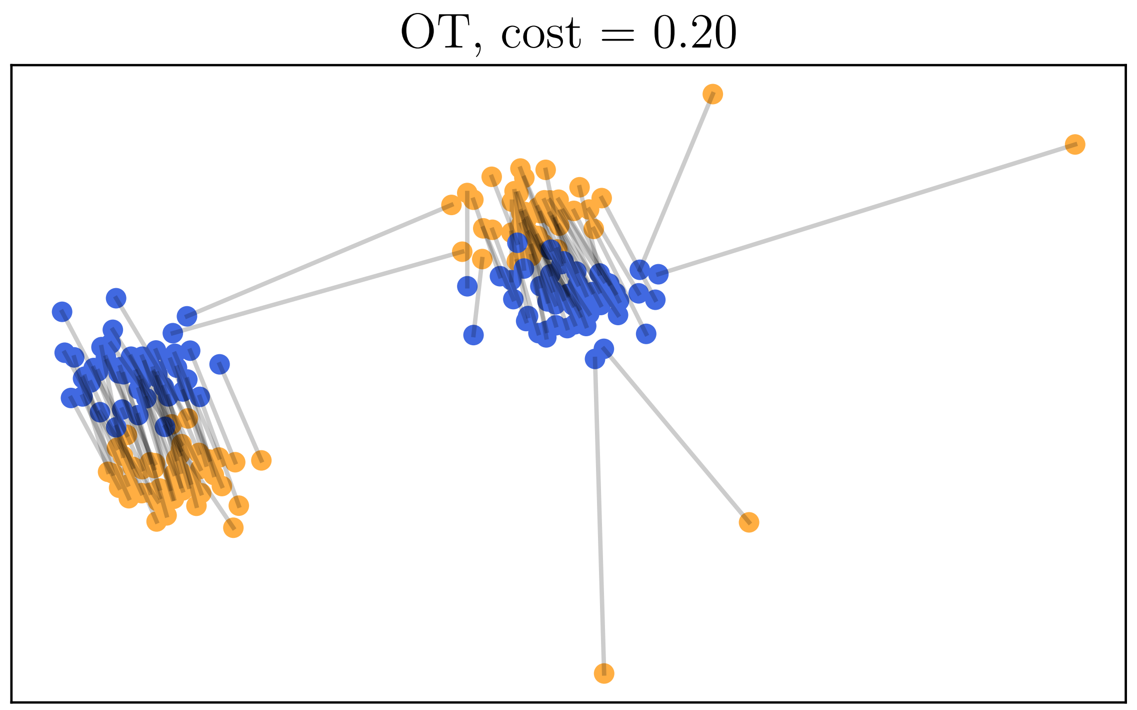
→ Regularize or relax the set of constraints

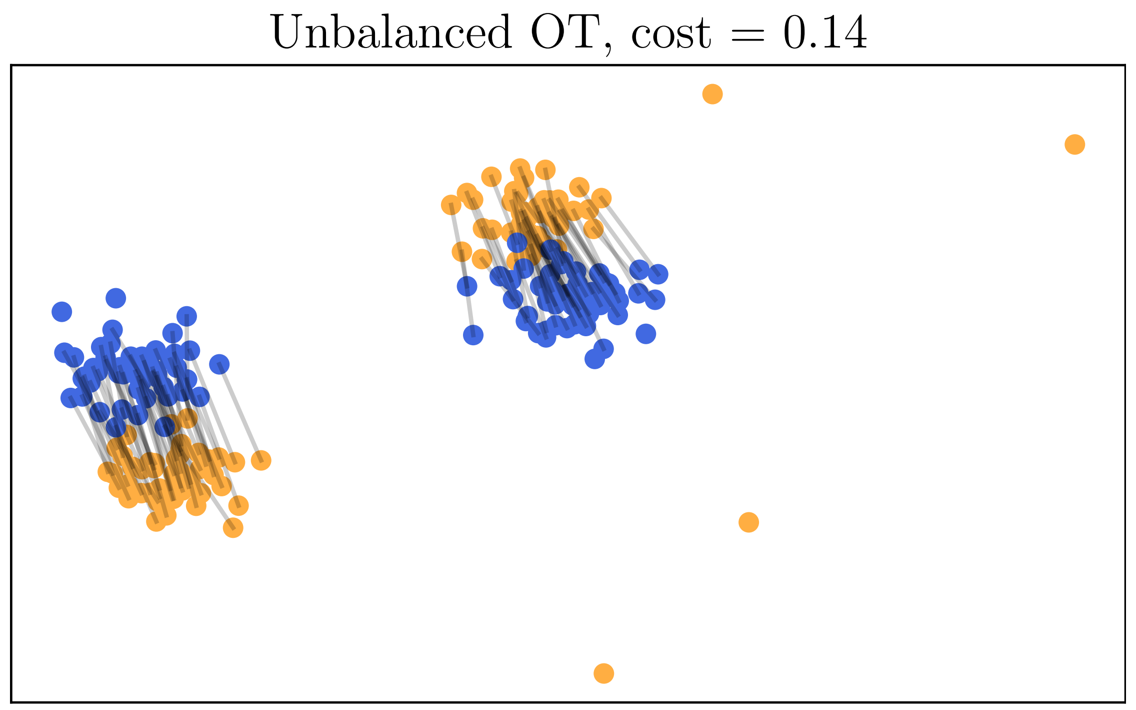
Unbalanced optimal transport [Benamou 2003]

$$\mathrm{UOT}_\lambda(\boldsymbol{h},\boldsymbol{g}) = \min_{\boldsymbol{T}\geq 0} \langle \boldsymbol{C}, \boldsymbol{T}\rangle + \lambda\left(D_\varphi(\boldsymbol{T}1_m, \boldsymbol{h}) + D_\varphi(\boldsymbol{T}^\top 1_n, \boldsymbol{g})\right)$$

with D_φ a divergence

How to deal with outliers or noisy samples ?


OT, cost = 0.20

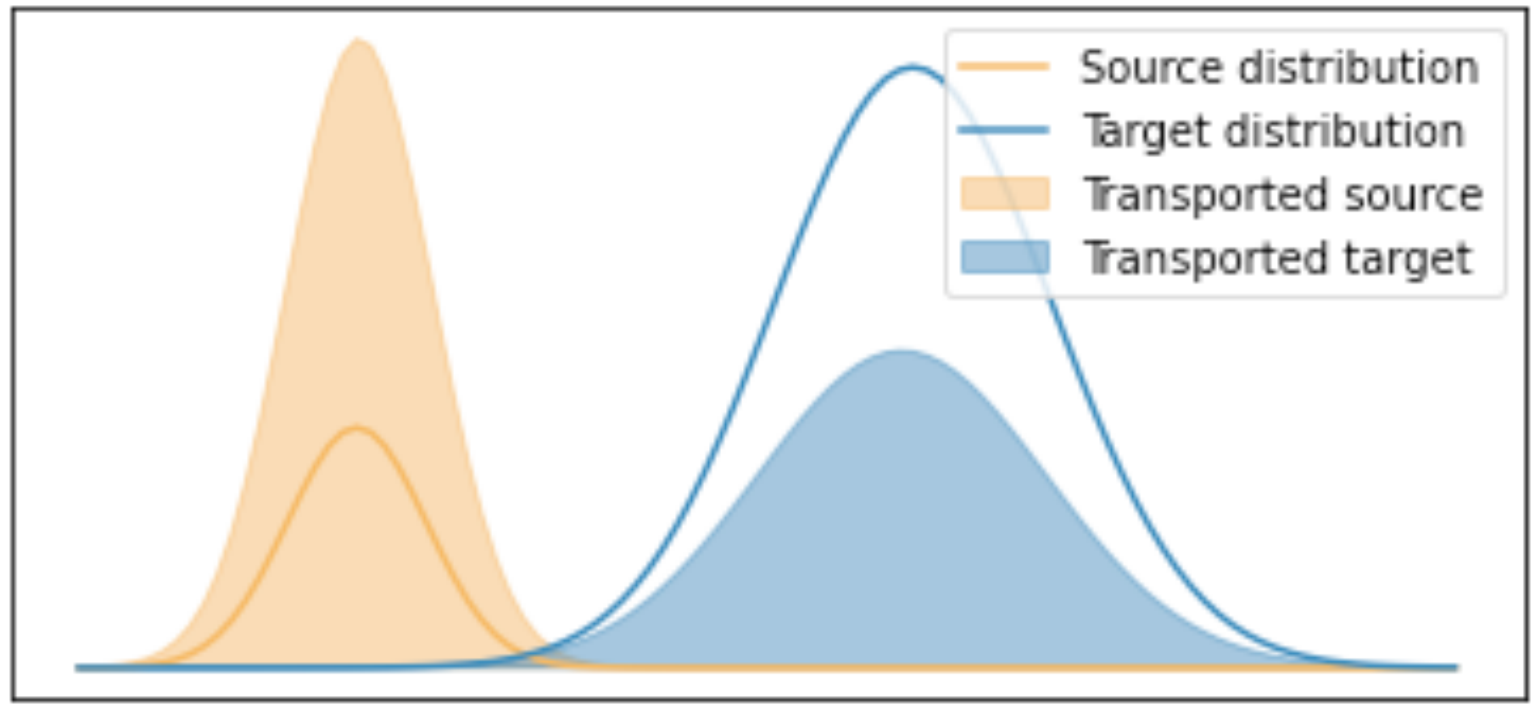

Unbalanced OT, cost = 0.14

Unbalanced Optimal Transport

Relaxing the set of constraints

$$\mathbf{\Pi}(\boldsymbol{h},\boldsymbol{g}) = \left\{ \boldsymbol{T} \in \mathbb{R}_+^{n\times m} \middle| \boldsymbol{T}1_m = \boldsymbol{h}, \boldsymbol{T}^\top 1_n = \boldsymbol{g} \right\}$$

How to work with unnormalized histograms?



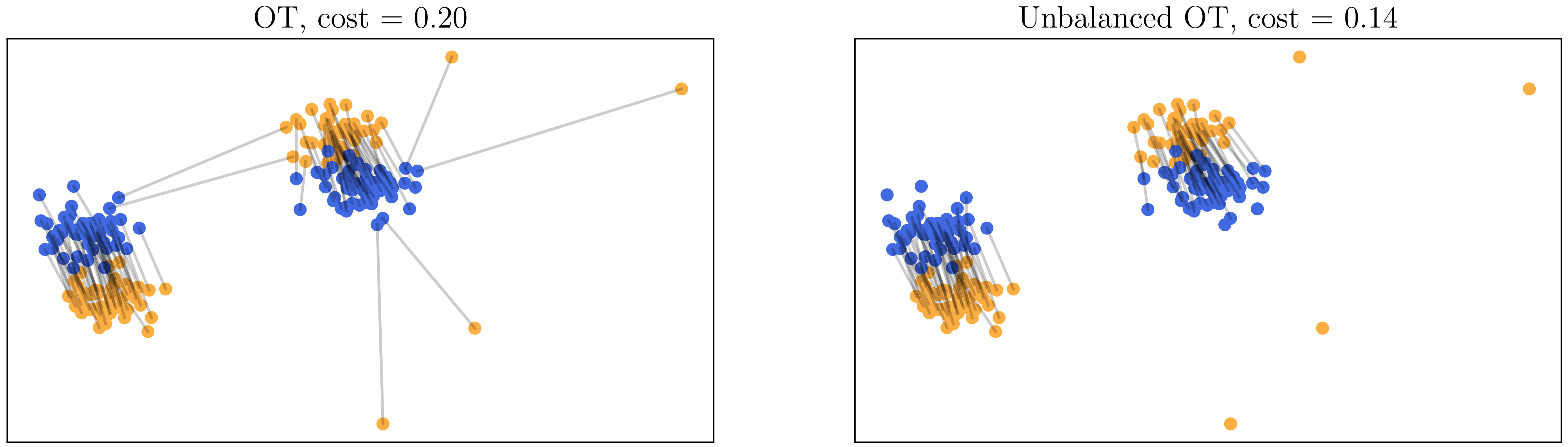
→ Regularize or relax the set of constraints

Unbalanced optimal transport [Benamou 2003]

$$\mathrm{UOT}_\lambda(\boldsymbol{h},\boldsymbol{g}) = \min_{\boldsymbol{T}\geq 0} \langle \boldsymbol{C}, \boldsymbol{T} \rangle + \lambda \left(D_\varphi(\boldsymbol{T}1_m, \boldsymbol{h}) + D_\varphi(\boldsymbol{T}^\top 1_n, \boldsymbol{g}) \right)$$

with D_φ a divergence

How to deal with outliers or noisy samples ?



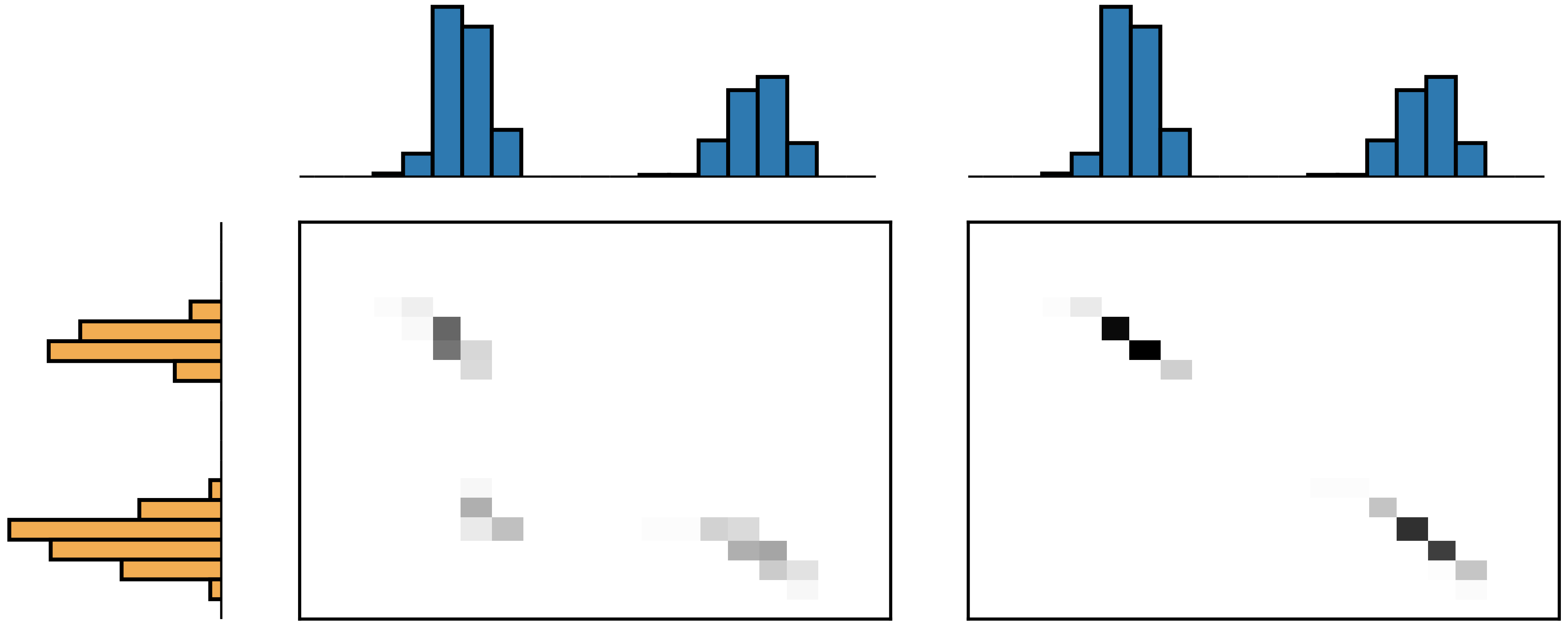
Unbalanced Optimal Transport

Relaxing the set of constraints

$$\mathrm{UOT}_\lambda(\boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \geq 0} \langle \boldsymbol{C}, \boldsymbol{T} \rangle + \lambda \left(D_\varphi(\boldsymbol{T} 1_m, \boldsymbol{h}) + D_\varphi(\boldsymbol{T}^\top 1_n, \boldsymbol{g}) \right) + \lambda_\epsilon \Omega(\boldsymbol{T})$$

When $\lambda = 0$, no mass is transported

and $\lambda \to \infty$, we recover the *balanced* OT problem (when $\|\boldsymbol{h}\|_1 = \|\boldsymbol{g}\|_1$)



Balanced OT

Unbalanced OT

Unbalanced Optimal Transport

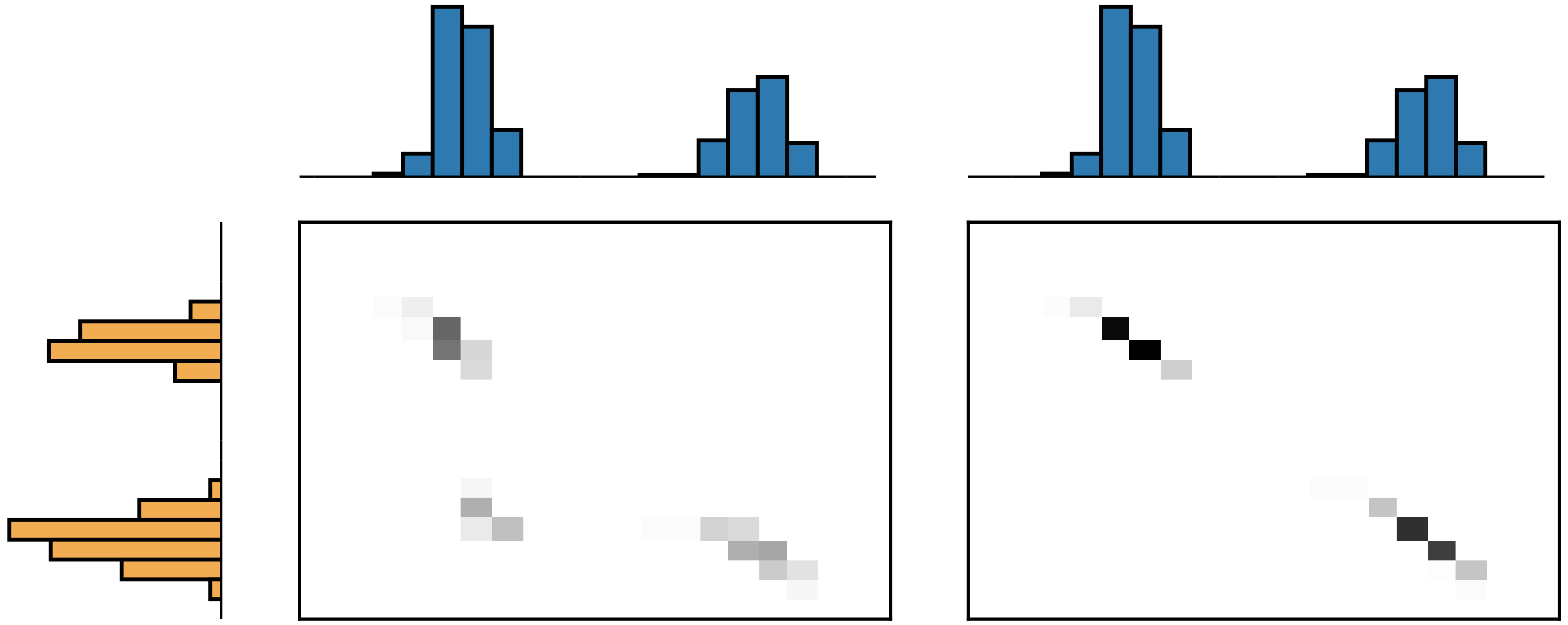
Relaxing the set of constraints

$$\mathrm{UOT}_\lambda(\boldsymbol{h},\boldsymbol{g}) = \min_{\boldsymbol{T}\geq 0} \langle \boldsymbol{C}, \boldsymbol{T}\rangle + \lambda\left(D_\varphi(\boldsymbol{T}1_m, \boldsymbol{h}) + D_\varphi(\boldsymbol{T}^\top 1_n, \boldsymbol{g})\right) + \lambda_\epsilon \Omega(\boldsymbol{T})$$

entropic penalization

When $\lambda = 0$, no mass is transported

and $\lambda \to \infty$, we recover the *balanced* OT problem (when $\|\boldsymbol{h}\|_1 = \|\boldsymbol{g}\|_1$)



Balanced OT

Unbalanced OT

Unbalanced Optimal Transport

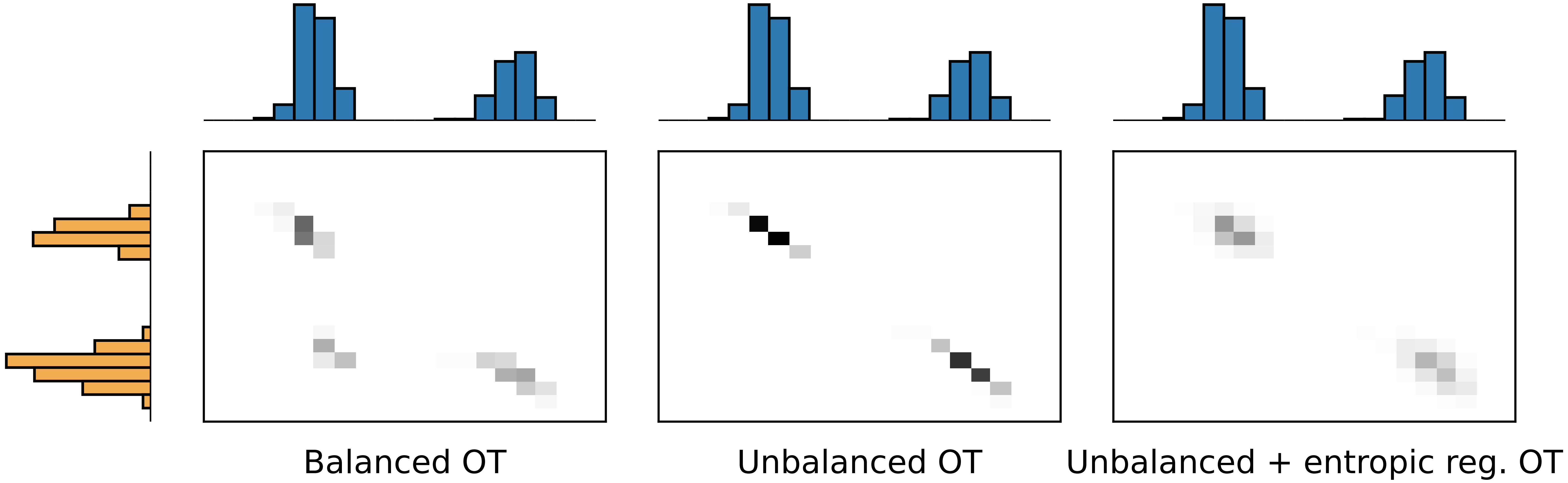
Relaxing the set of constraints

$$\mathrm{UOT}_\lambda(\boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T}\geq 0} \langle \boldsymbol{C}, \boldsymbol{T} \rangle + \lambda \Big(D_\varphi(\boldsymbol{T}1_m, \boldsymbol{h}) + D_\varphi(\boldsymbol{T}^\top 1_n, \boldsymbol{g}) \Big) + \lambda_\epsilon \Omega(\boldsymbol{T})$$

entropic penalization

When $\lambda = 0$, no mass is transported

and $\lambda \to \infty$, we recover the *balanced* OT problem (when $\|\boldsymbol{h}\|_1 = \|\boldsymbol{g}\|_1$)



Unbalanced Optimal Transport

Relaxing the set of constraints

$$\mathrm{UOT}_\lambda(\boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \geq 0} \langle \boldsymbol{C}, \boldsymbol{T} \rangle + \lambda \left(D_\varphi(\boldsymbol{T} 1_m, \boldsymbol{h}) + D_\varphi(\boldsymbol{T}^\top 1_n, \boldsymbol{g}) \right) + \lambda_\epsilon \Omega(\boldsymbol{T})$$

When $\lambda = 0$, no mass is transported

and $\lambda \to \infty$, we recover the *balanced* OT problem (when $\|\boldsymbol{h}\|_1 = \|\boldsymbol{g}\|_1$)

We will consider several cases:

- D_φ is L1: Partial OT problem
- D_φ is L2
- D_φ is KL

Unbalanced Optimal Transport

Partial Optimal Transport (D_φ is L1)

Fix the amount of mass s to be transported

$$\mathbf{\Pi}^u(\boldsymbol{h}, \boldsymbol{g}) = \left\{\boldsymbol{T} \in \mathbb{R}_+^{n\times m} | \boldsymbol{T}1_m \leq \boldsymbol{h}, \boldsymbol{T}^\top 1_n \leq \boldsymbol{g}, 1_n^\top \boldsymbol{T}1_m = s\right\}.$$

Unbalanced OT with L1 divergence $\mathrm{UOT}_\lambda(\boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T}\geq 0} \langle \boldsymbol{C}, \boldsymbol{T}\rangle + \lambda \left(|\boldsymbol{T}1_m - \boldsymbol{h}| + |\boldsymbol{T}^\top 1_n - \boldsymbol{g}|\right)$

Unbalanced Optimal Transport

Partial Optimal Transport (D_φ is L1)

Fix the amount of mass s to be transported

$$\boldsymbol{\Pi}^u(\boldsymbol{h}, \boldsymbol{g}) = \left\{ \boldsymbol{T} \in \mathbb{R}_+^{n \times m} | \boldsymbol{T} 1_m \leq \boldsymbol{h}, \boldsymbol{T}^\top 1_n \leq \boldsymbol{g}, 1_n^\top \boldsymbol{T} 1_m = s \right\}.$$

Unbalanced OT with L1 divergence $\quad \mathrm{UOT}_\lambda(\boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \geq 0} \langle \boldsymbol{C}, \boldsymbol{T} \rangle + \lambda \left(|\boldsymbol{T} 1_m - \boldsymbol{h}| + |\boldsymbol{T}^\top 1_n - \boldsymbol{g}| \right)$

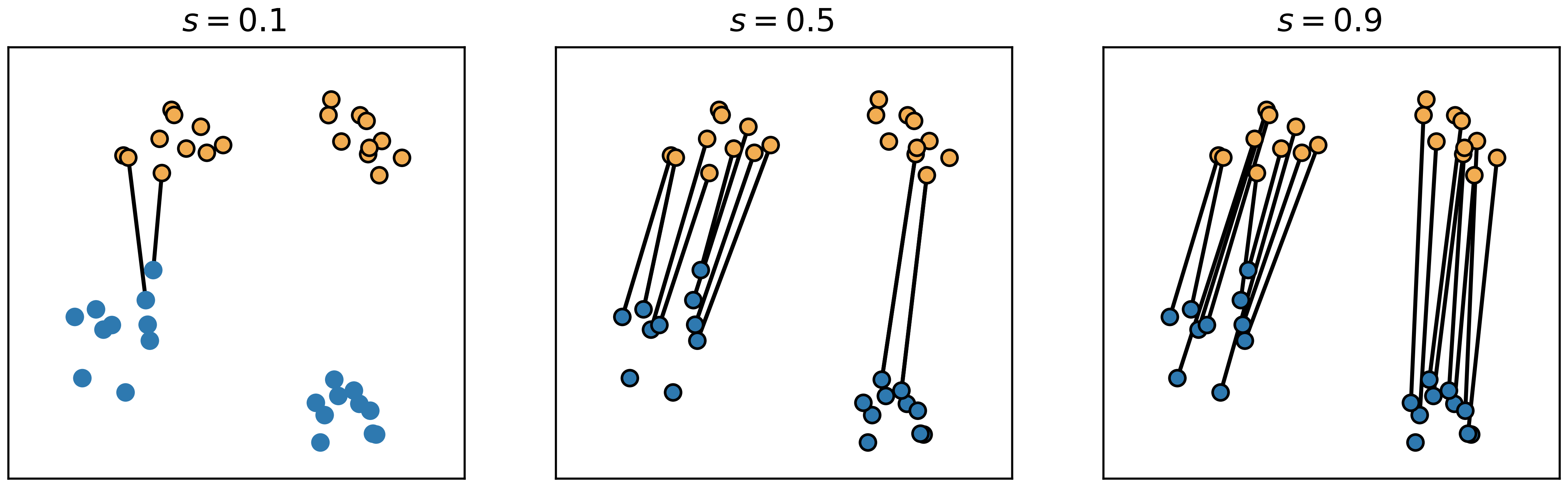
Unbalanced Optimal Transport

Partial Optimal Transport (D_φ is L1)

Fix the amount of mass s to be transported

$$\mathbf{\Pi}^u(\boldsymbol{h}, \boldsymbol{g}) = \left\{ \boldsymbol{T} \in \mathbb{R}_+^{n \times m} | \boldsymbol{T} 1_m \leq \boldsymbol{h}, \boldsymbol{T}^\top 1_n \leq \boldsymbol{g}, 1_n^\top \boldsymbol{T} 1_m = s \right\}.$$

Unbalanced OT with L1 divergence $\quad \mathrm{UOT}_\lambda(\boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \geq 0} \langle \boldsymbol{C}, \boldsymbol{T} \rangle + \lambda \left(|\boldsymbol{T} 1_m - \boldsymbol{h}| + |\boldsymbol{T}^\top 1_n - \boldsymbol{g}| \right)$



Unbalanced Optimal Transport

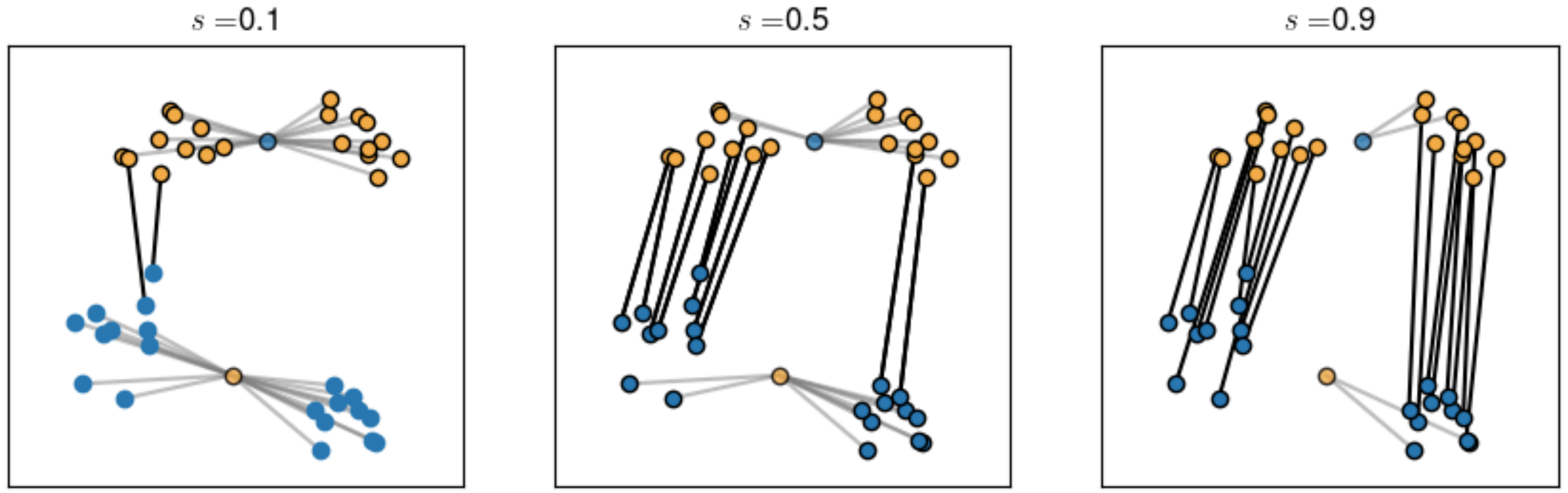
Partial Optimal Transport (D_φ is L1)

Fix the amount of mass s to be transported

$$\mathbf{\Pi}^u(\mathbf{h},\mathbf{g}) = \left\{ T \in \mathbb{R}_+^{n\times m} | T1_m \leq \mathbf{h}, T^\top 1_n \leq \mathbf{g}, 1_n^\top T 1_m = s \right\}.$$

Unbalanced OT with L1 divergence $\mathrm{UOT}_\lambda(\mathbf{h},\mathbf{g}) = \min_{T\geq 0} \langle \mathbf{C}, T\rangle + \lambda \left(|T1_m - \mathbf{h}| + |T^\top 1_n - \mathbf{g}| \right)$

→ add dummy points with mass $h_{n+1} = \|\mathbf{g}\|_1 - s$ and $g_{m+1} = \|\mathbf{h}\|_1 - s$ with null cost [Chapel 2020]



Solving an exact OT problem
⇒ sparsity of the solution

Unbalanced Optimal Transport

D_φ is KL - MM algorithm

When the divergence is Kullback-Leibler

$$\mathrm{UOT}_\lambda(\boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \geq 0} \langle \boldsymbol{C}, \boldsymbol{T} \rangle + \lambda \left(\mathrm{KL}(\boldsymbol{T} 1_m, \boldsymbol{h}) + \mathrm{KL}(\boldsymbol{T}^\top 1_n, \boldsymbol{g}) \right)$$

Unbalanced Optimal Transport

D_φ is KL - MM algorithm

$$KL(x,y) = \sum_i x_i \log \frac{x_i}{y_j} - x_i + y_i$$

When the divergence is Kullback-Leibler

$$\mathrm{UOT}_\lambda(\boldsymbol{h},\boldsymbol{g}) = \min_{\boldsymbol{T}\geq 0} \langle \boldsymbol{C},\boldsymbol{T}\rangle + \lambda\Big(\mathrm{KL}(\boldsymbol{T}1_m,\boldsymbol{h}) + \mathrm{KL}(\boldsymbol{T}^\top 1_n,\boldsymbol{g})\Big)$$

Unbalanced Optimal Transport

D_φ is KL - MM algorithm

When the divergence is Kullback-Leibler

$$KL(x, y) = \sum_i x_i \log \frac{x_i}{y_j} - x_i + y_i$$

$$\mathrm{UOT}_\lambda(\boldsymbol{h}, \boldsymbol{g}) = \min_{T \geq 0} \langle \boldsymbol{C}, \boldsymbol{T} \rangle + \lambda \Big(\mathrm{KL}(\boldsymbol{T} 1_m, \boldsymbol{h}) + \mathrm{KL}(\boldsymbol{T}^\top 1_n, \boldsymbol{g}) \Big)$$

Majorization-minimisation

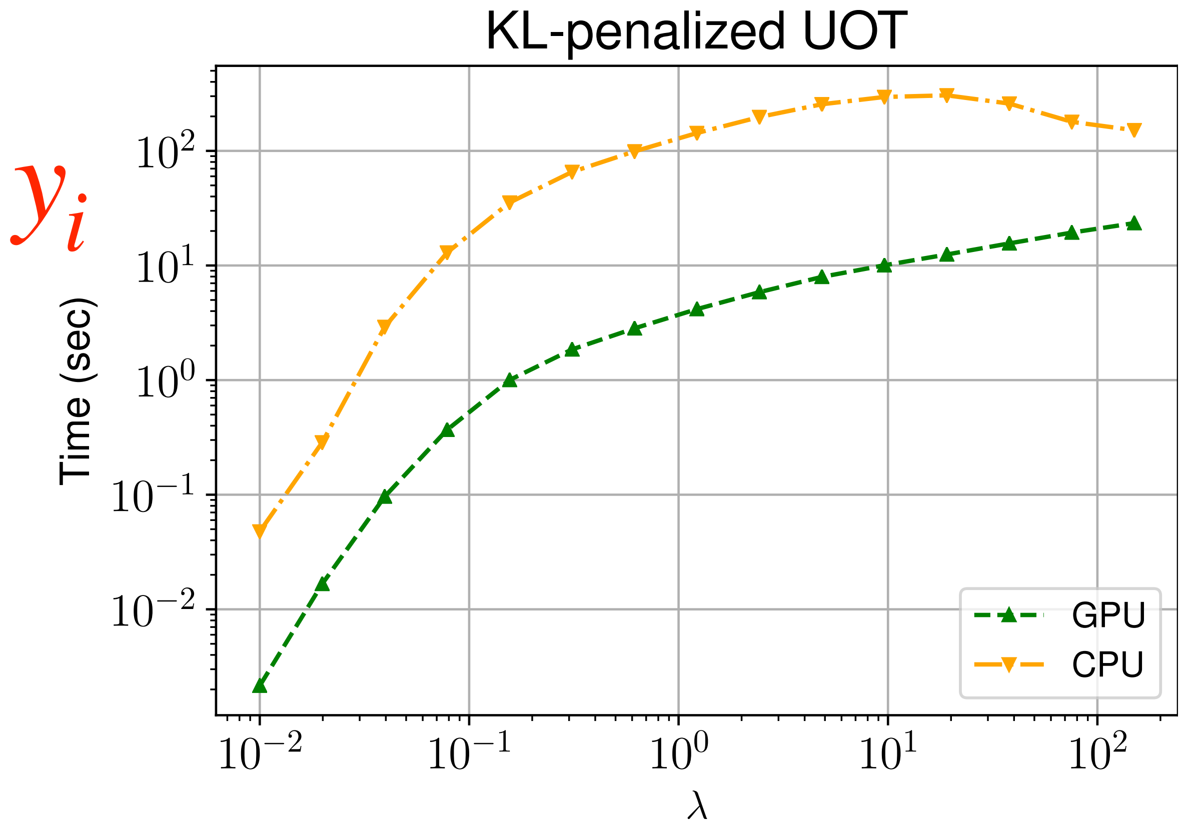
define a surrogate of the obj

optimize the surrogate

Unbalanced Optimal Transport

D_φ is KL - MM algorithm

$$KL(x,y) = \sum_i x_i \log \frac{x_i}{y_j} - x_i + y_i$$



When the divergence is Kullback-Leibler

$$\mathrm{UOT}_\lambda(h, g) = \min_{T \geq 0} \langle C, T \rangle + \lambda \Big(\mathrm{KL}(T 1_m, h) + \mathrm{KL}(T^\top 1_n, g) \Big)$$

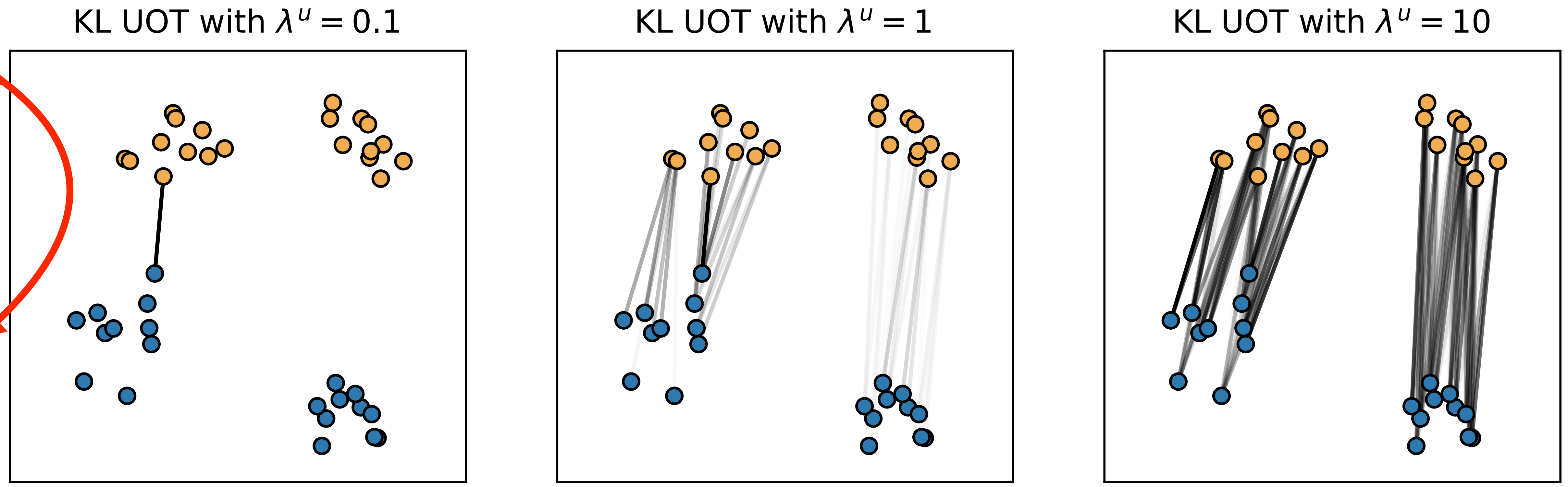
Iterative algorithm that resembles the Sinkhorn algorithm [Chapel 2021]

$$T^{(k+1)} = \mathrm{diag}\left(\frac{g}{T^{(k)} 1_m} \right)^{\frac{1}{2}} \left(T^{(k)} \odot \exp\left(-\frac{C}{2\lambda} \right) \right) \mathrm{diag}\left(\frac{h}{T^{(k)\top} 1_n} \right)^{\frac{1}{2}}$$

Majorization-minimisation

define a surrogate of the obj

optimize the surrogate


KL UOT with $\lambda^u = 0.1$

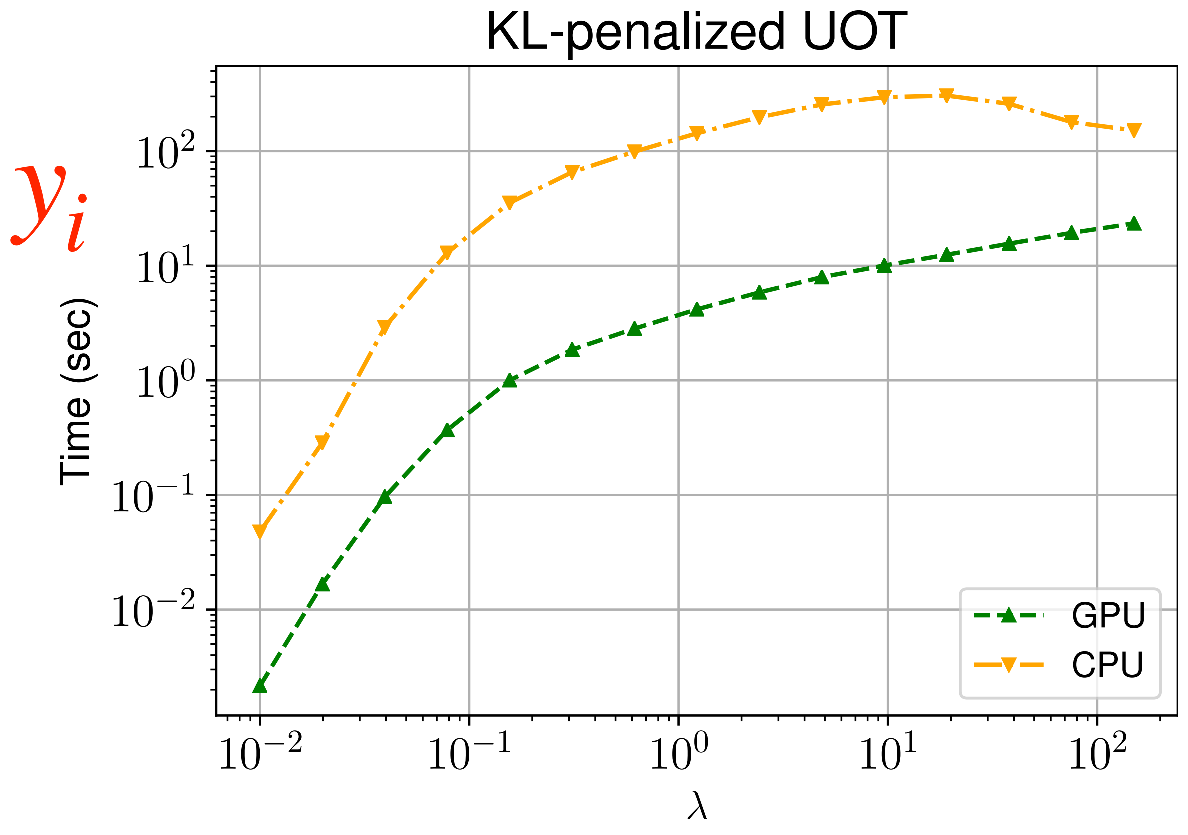
KL UOT with $\lambda^u = 1$

KL UOT with $\lambda^u = 10$

Unbalanced Optimal Transport

D_φ is KL - MM algorithm

$$KL(x,y) = \sum_i x_i \log \frac{x_i}{y_j} - x_i + y_i$$



When the divergence is Kullback-Leibler

$$\mathrm{UOT}_\lambda(\boldsymbol{h},\boldsymbol{g}) = \min_{\boldsymbol{T}\geq 0} \langle \boldsymbol{C},\boldsymbol{T}\rangle + \lambda\left(\mathrm{KL}(\boldsymbol{T}1_m,\boldsymbol{h}) + \mathrm{KL}(\boldsymbol{T}^\top 1_n,\boldsymbol{g})\right)$$

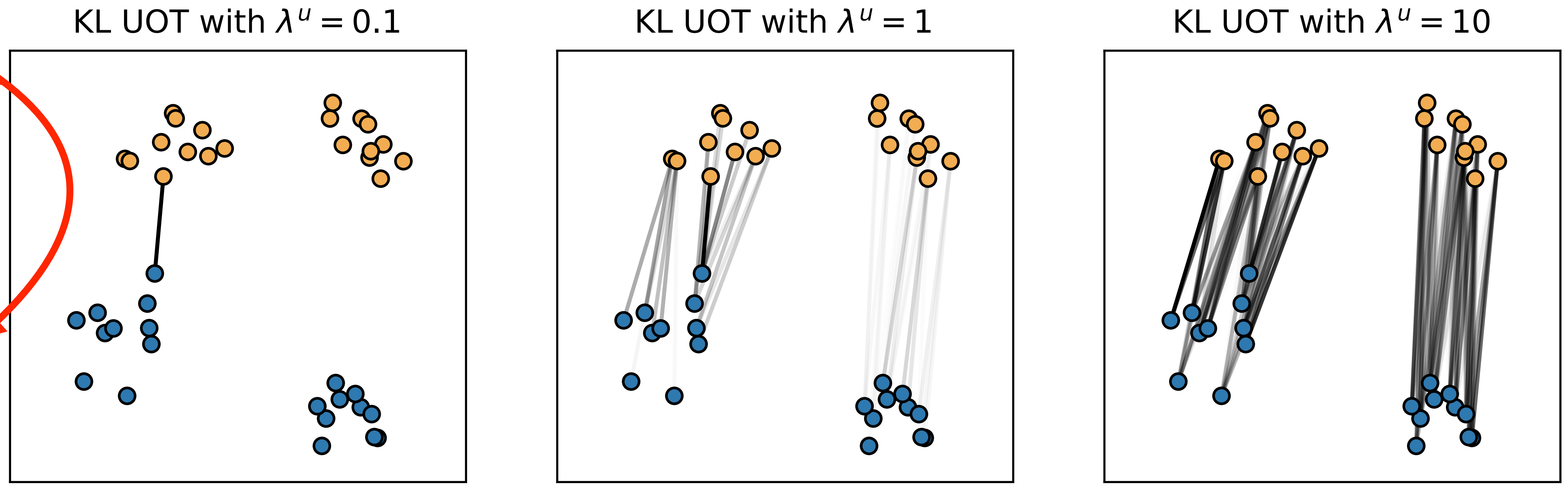
Iterative algorithm that resembles the Sinkhorn algorithm [Chapel 2021]

$$\boldsymbol{T}^{(k+1)} = \mathrm{diag}\left(\frac{\boldsymbol{g}}{\boldsymbol{T}^{(k)}1_m}\right)^{\frac{1}{2}} \left(\boldsymbol{T}^{(k)} \odot \exp\left(-\frac{\boldsymbol{C}}{2\lambda}\right)\right) \mathrm{diag}\left(\frac{\boldsymbol{h}}{\boldsymbol{T}^{(k)\top}1_n}\right)^{\frac{1}{2}}$$

Majorization-minimisation

define a surrogate of the obj

optimize the surrogate


KL UOT with $\lambda^u = 0.1$

KL UOT with $\lambda^u = 1$

KL UOT with $\lambda^u = 10$

Smoothes the OT plan

Amenable to GPU computation

Unbalanced Optimal Transport

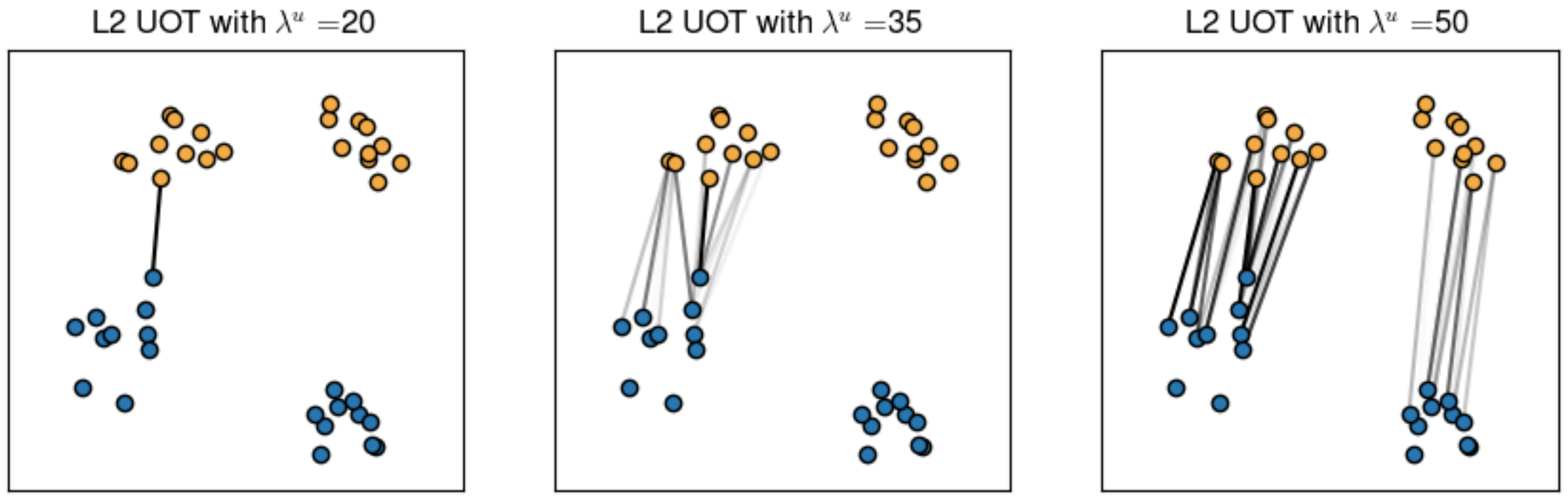
D_φ is L2 - MM algorithm

We can also define an iterative algorithm for a L2 divergence

$$\mathrm{UOT}_\lambda(\boldsymbol{h},\boldsymbol{g}) = \min_{\boldsymbol{T}\geq 0} \langle \boldsymbol{C}, \boldsymbol{T}\rangle + \lambda \left(\|\boldsymbol{T}^\top 1_m - \boldsymbol{h}\|_2^2 + \|\boldsymbol{T}^\top 1_n - \boldsymbol{g}\|_2^2 \right)$$

(another) Iterative algorithm with deterministic updates [Chapel 2021]

$$\boldsymbol{T}^{(k+1)} = \boldsymbol{T}^{(k)} \odot \frac{\max(0, \boldsymbol{g}1_m^\top + 1_n\boldsymbol{h}^\top - \frac{1}{\lambda}C)}{\boldsymbol{T}^{(k)}O_m + O_n\boldsymbol{T}^{(k)}} \text{ with } O_\ell = 1_\ell 1_\ell^\top$$



Unbalanced Optimal Transport

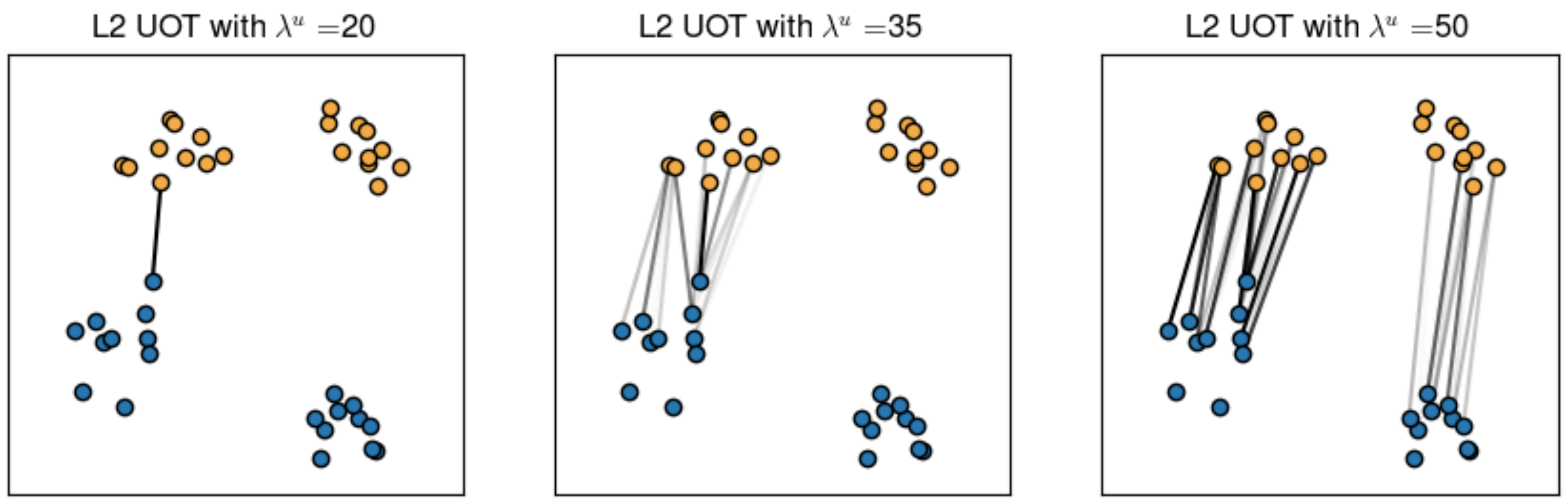
D_φ is L2 - MM algorithm

We can also define an iterative algorithm for a L2 divergence

$$\mathrm{UOT}_\lambda(\boldsymbol{h},\boldsymbol{g}) = \min_{\boldsymbol{T}\geq 0} \langle \boldsymbol{C},\boldsymbol{T}\rangle + \lambda\left(\|\boldsymbol{T}^\top 1_m - \boldsymbol{h}\|_2^2 + \|\boldsymbol{T}^\top 1_n - \boldsymbol{g}\|_2^2\right)$$

(another) Iterative algorithm with deterministic updates [Chapel 2021]

$$\boldsymbol{T}^{(k+1)} = \boldsymbol{T}^{(k)} \odot \frac{\max(0, \boldsymbol{g}1_m^\top + 1_n\boldsymbol{h}^\top - \frac{1}{\lambda}C)}{\boldsymbol{T}^{(k)}O_m + O_n\boldsymbol{T}^{(k)}} \text{ with } O_\ell = 1_\ell 1_\ell^\top$$



Smoothes the OT plan (but less than KL!)
Also amenable to GPU computation

Unbalanced Optimal Transport

D_φ is L2 - regularization path

We can rewrite the UOT problem in a vectorial form

$$\min_{t \geq 0} \underbrace{\boldsymbol{c}^\top t}_{\text{OT cost}} + \lambda \underbrace{D_\varphi(\boldsymbol{H}t - \boldsymbol{y})}_{\text{deviation of the marginals}} = \min_{t \geq 0} \lambda \|\boldsymbol{H}t - \boldsymbol{y}\|_2^2 + \boldsymbol{c}^\top t$$

with $\boldsymbol{y} = [\boldsymbol{h}, \boldsymbol{g}]^\top$

Unbalanced Optimal Transport

D_φ is L2 - regularization path

We can rewrite the UOT problem in a vectorial form

Combination of identity matrices
and matrices of ones

$$\min_{t \geq 0} \underbrace{\boldsymbol{c}^\top t}_{\text{OT cost}} + \lambda \underbrace{D_\varphi(\boldsymbol{H} t - \boldsymbol{y})}_{\text{deviation of the marginals}} = \min_{t \geq 0} \lambda \|\boldsymbol{H} t - \boldsymbol{y}\|_2^2 + \boldsymbol{c}^\top t$$

with $\boldsymbol{y} = [\boldsymbol{h}, \boldsymbol{g}]^\top$

Unbalanced Optimal Transport

D_φ is L2 - regularization path

We can rewrite the UOT problem in a vectorial form

Combination of identity matrices
and matrices of ones

$$\min_{t\geq 0} \underbrace{\boldsymbol{c}^\top t}_{\text{OT cost}} + \lambda \underbrace{D_\varphi(\boldsymbol{H}t - \boldsymbol{y})}_{\text{deviation of the marginals}} = \min_{t\geq 0} \underbrace{\lambda \|\boldsymbol{H}t - \boldsymbol{y}\|_2^2}_{\text{(least square problem)}} + \boldsymbol{c}^\top t$$

with $\boldsymbol{y} = [\boldsymbol{h}, \boldsymbol{g}]^\top$

Unbalanced Optimal Transport

D_φ is L2 - regularization path

We can rewrite the UOT problem in a vectorial form

Combination of identity matrices
and matrices of ones

$$\min_{t\geq 0} \underbrace{\boldsymbol{c}^\top t}_{\text{OT cost}} + \lambda \underbrace{D_\varphi(\boldsymbol{H}t - \boldsymbol{y})}_{\text{deviation of the marginals}} = \min_{t\geq 0} \underbrace{\lambda \|\boldsymbol{H}t - \boldsymbol{y}\|_2^2}_{\text{(least square problem)}} + \boldsymbol{c}^\top t$$

with $\boldsymbol{y} = [\boldsymbol{h}, \boldsymbol{g}]^\top$

→ Classical linear regression with positivity constraints, a sparse design matrix $\boldsymbol{H}$ and a weighted L1 (Lasso) regularization $\frac{1}{\lambda}\boldsymbol{c}^\top t = \frac{1}{\lambda}\sum_k c_k |t_k|$ [Chapel 2021]

Unbalanced Optimal Transport

D_φ is L2 - regularization path

We can rewrite the UOT problem in a vectorial form

Combination of identity matrices and matrices of ones

$$\min_{t\geq 0} \underbrace{\boldsymbol{c}^\top t}_{\text{OT cost}} + \lambda \underbrace{D_\varphi(\boldsymbol{H}t - \boldsymbol{y})}_{\text{deviation of the marginals}} = \min_{t\geq 0} \lambda \|\boldsymbol{H}t - \boldsymbol{y}\|_2^2 + \boldsymbol{c}^\top t$$

(least square problem)

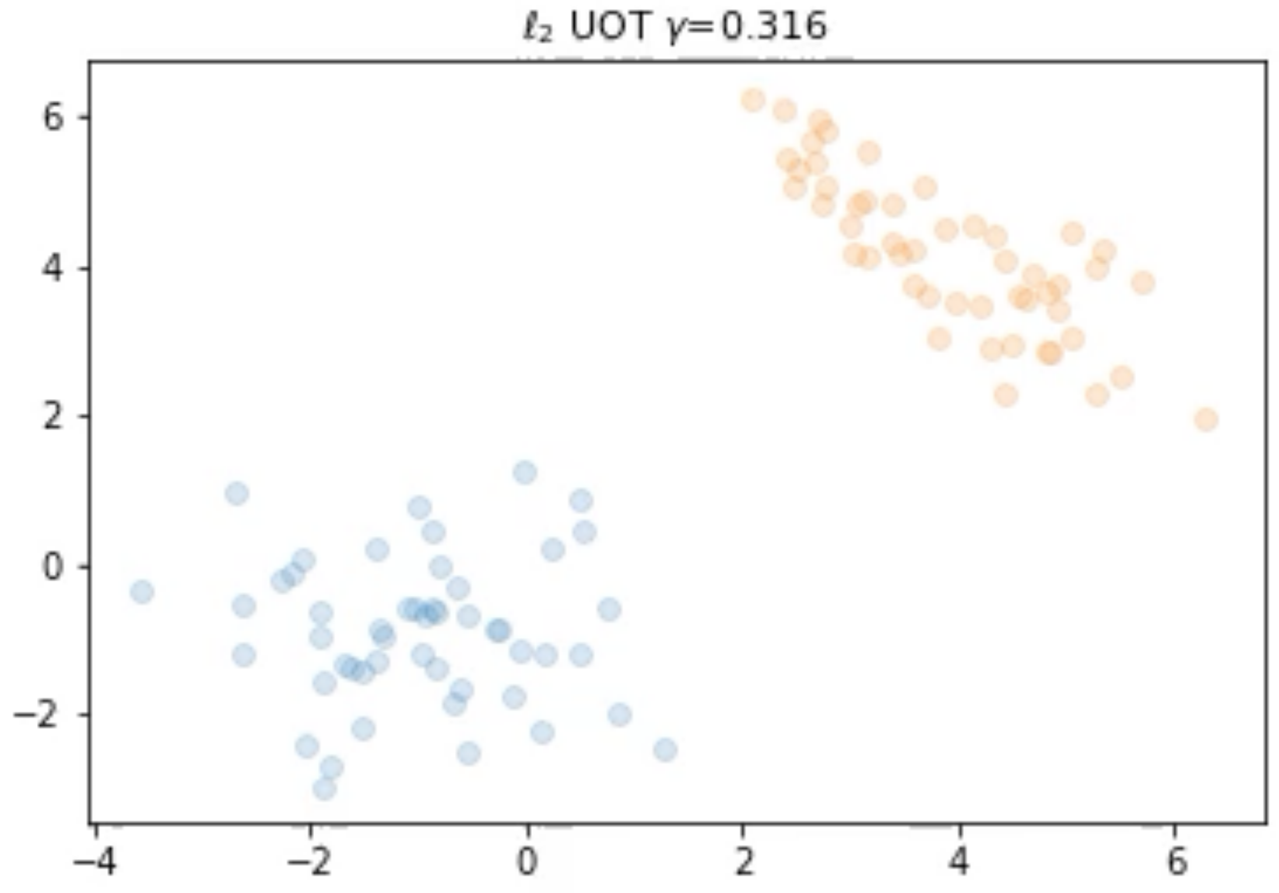
with $\boldsymbol{y} = [\boldsymbol{h}, \boldsymbol{g}]^\top$

→ Classical linear regression with positivity constraints, a sparse design matrix $\boldsymbol{H}$ and a weighted L1 (Lasso) regularization $\frac{1}{\lambda}\boldsymbol{c}^\top t = \frac{1}{\lambda}\sum_k c_k|t_k|$ [Chapel 2021]

→ Borrow the tools from a large literature on solving those problems

Unbalanced Optimal Transport

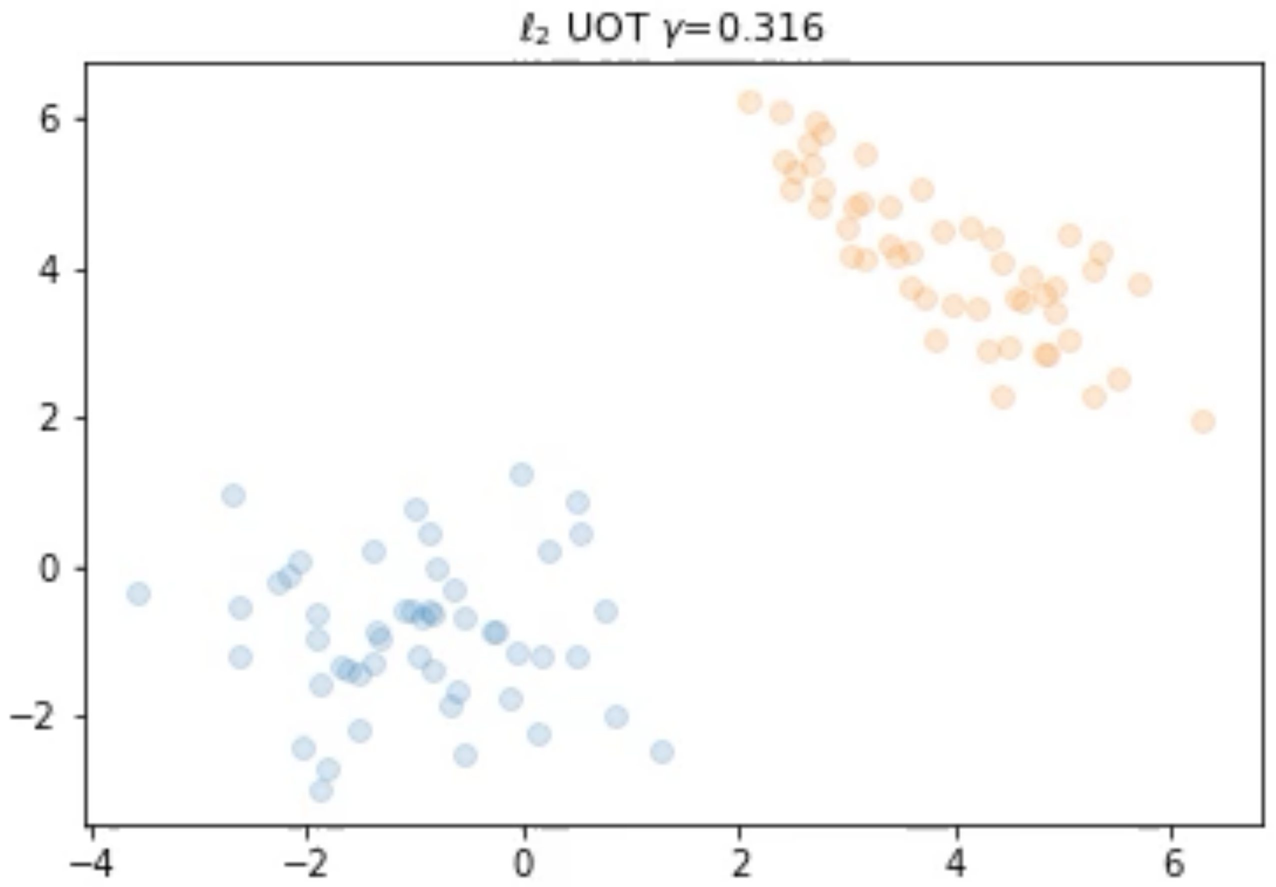
D_φ is L2 - regularization path



Unbalanced Optimal Transport

D_φ is L2 - regularization path

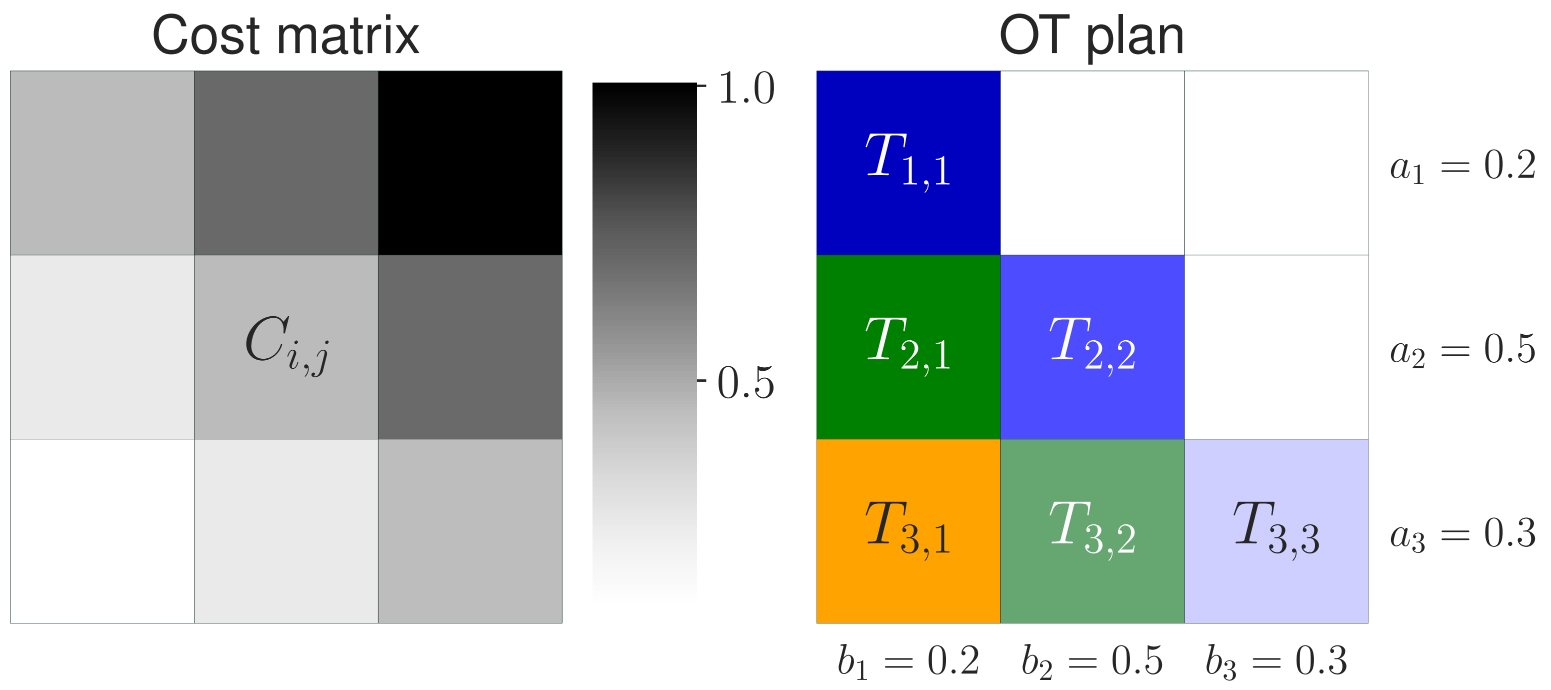
similarly to the LARS algorithm
find the set of ALL solutions

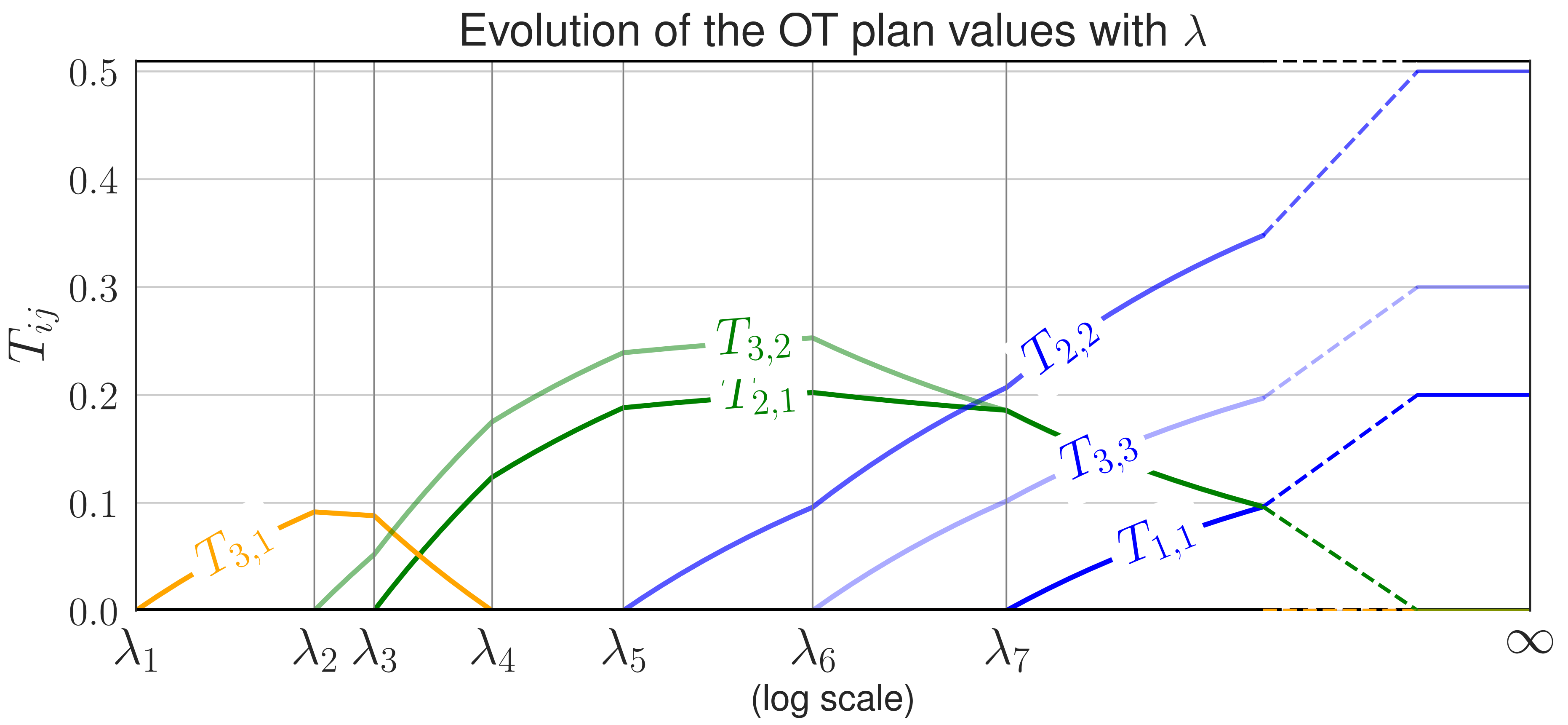


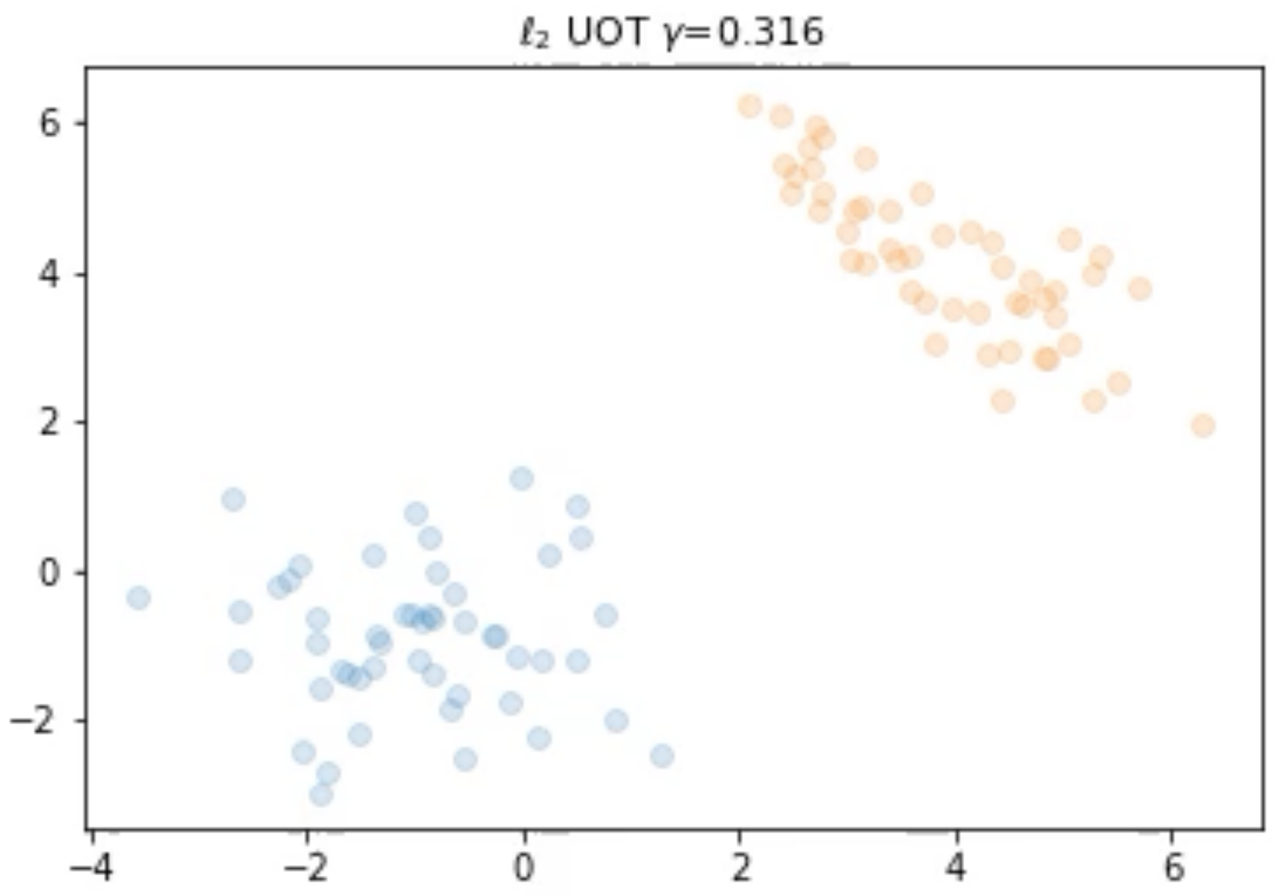
Unbalanced Optimal Transport

D_φ is L2 - regularization path

similarly to the LARS algorithm
find the set of ALL solutions







With quadratic divergence, solutions are piecewise linear with $\gamma = \dfrac{1}{\lambda}$

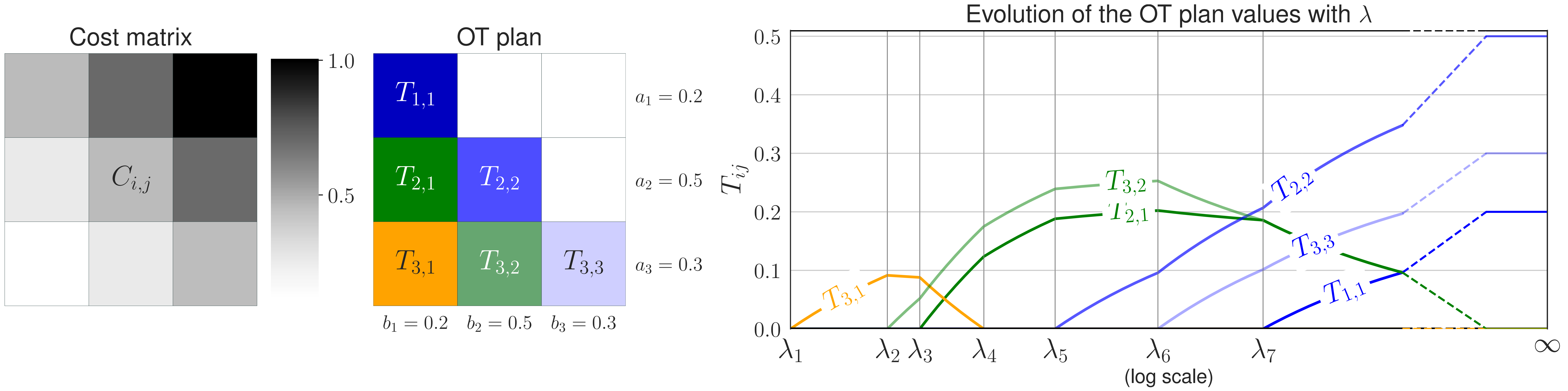
$$\left.\begin{array}{l} \text{2. increase } \lambda \text{ until there is a change on the support of } \boldsymbol{t} \\ \text{3. update } \boldsymbol{t} \text{ (incremental resolution of linear equations)} \end{array}\right\} O(nm)$$

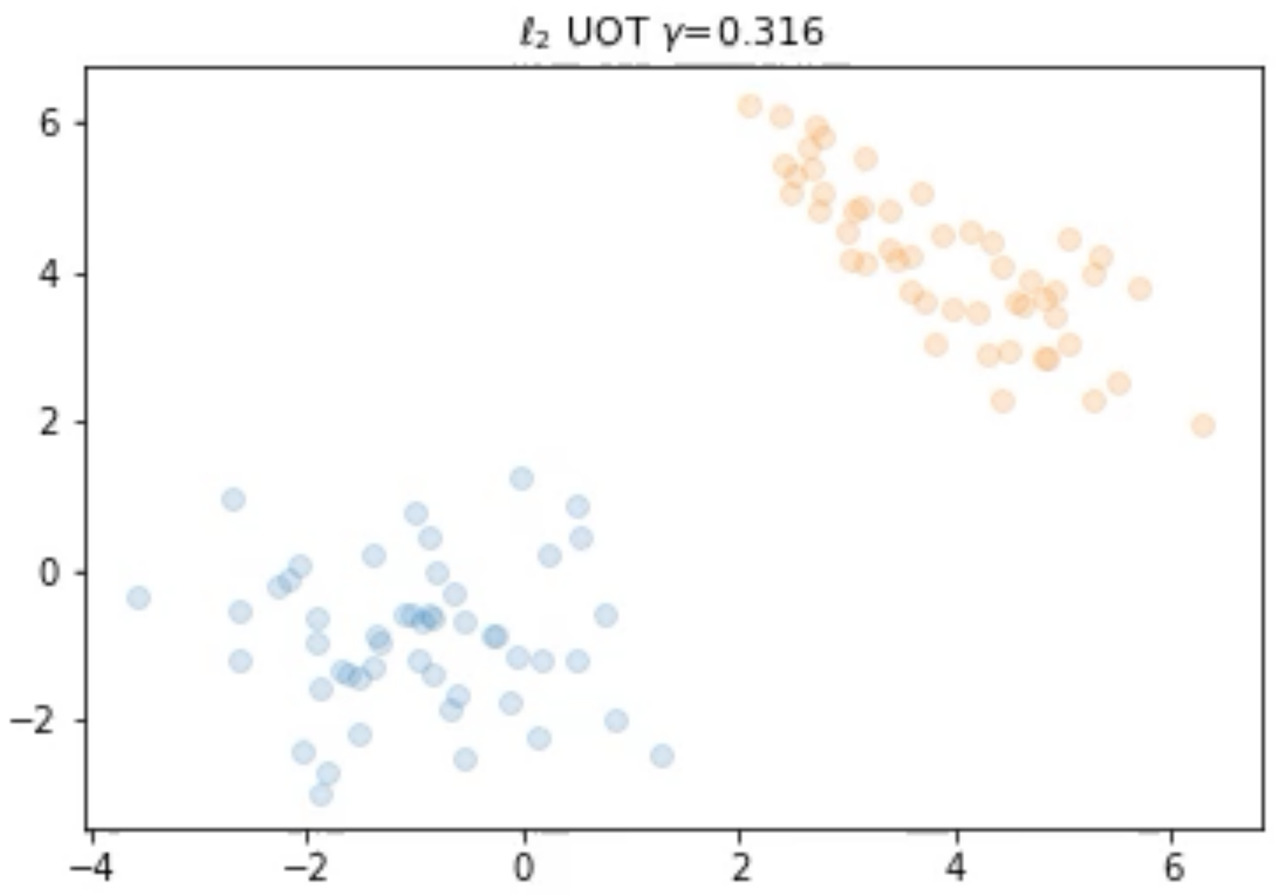
1. start with $\lambda = 0$
2. increase λ until there is a change on the support of $\boldsymbol{t}$
3. update $\boldsymbol{t}$ (incremental resolution of linear equations)
4. repeat until $\lambda = +\infty$

Unbalanced Optimal Transport

D_φ is L2 - regularization path

similarly to the LARS algorithm
find the set of ALL solutions





With quadratic divergence, solutions are piecewise linear with $\gamma = \dfrac{1}{\lambda}$

1. start with $\lambda = 0$
2. increase λ until there is a change on the support of $\boldsymbol{t}$
3. update $\boldsymbol{t}$ (incremental resolution of linear equations) $\Big\} O(nm)$
4. repeat until $\lambda = +\infty$

Unbalanced Optimal Transport

The problem has been formalized, and there exists some (efficient) algorithms

Some open questions (among others!)
- how choosing the *right* regularization parameter?
- does it really deal with outliers? which guarantees on the solution?

Some challenges of OT

Scalability of the algorithms

Unstable, not robust to outliers

Needs a common metric space: Gromov-Wasserstein distance on stage [Memoli 2011]

$$GW(\boldsymbol{C}_X, \boldsymbol{C}_Y, \boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \in \boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g})} \langle L(\boldsymbol{C}_X, \boldsymbol{C}_Y) \otimes \boldsymbol{T}, \boldsymbol{T} \rangle = \min_{\boldsymbol{T} \in \boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g})} \sum_{i,j,k,l} L(d_X(\boldsymbol{x}_i, \boldsymbol{x}_k), d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)) T_{i,j} T_{k,l}$$

Some challenges of OT

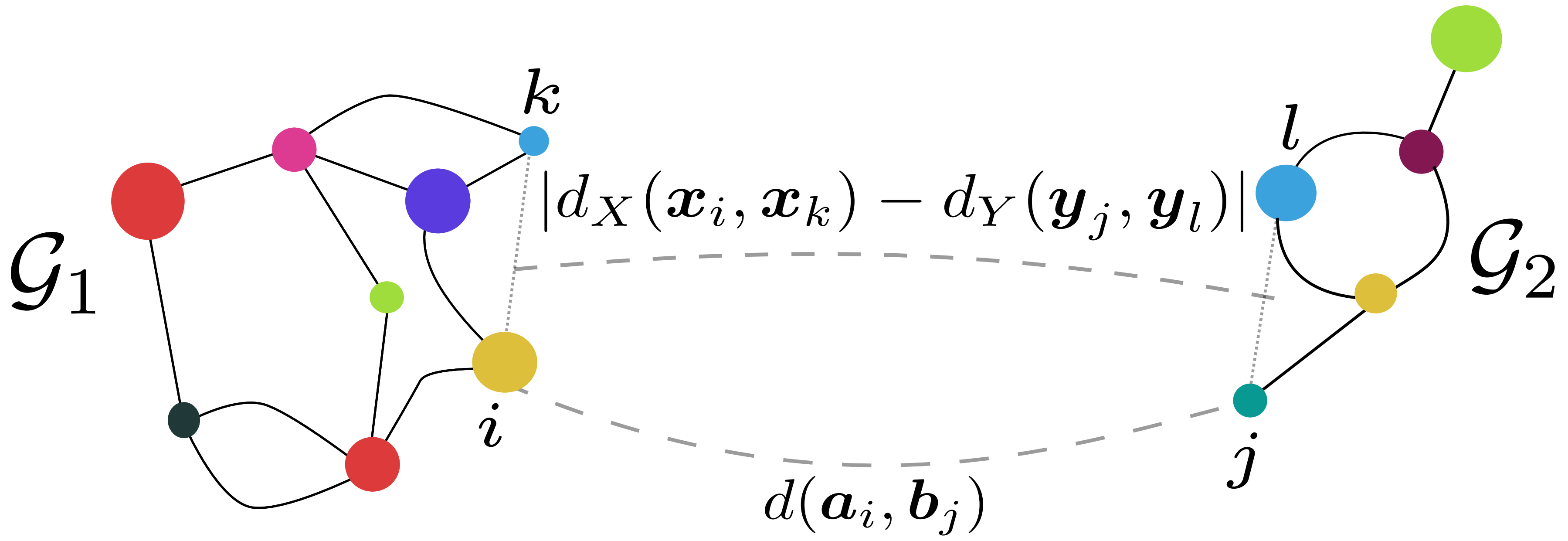
Scalability of the algorithms

Unstable, not robust to outliers

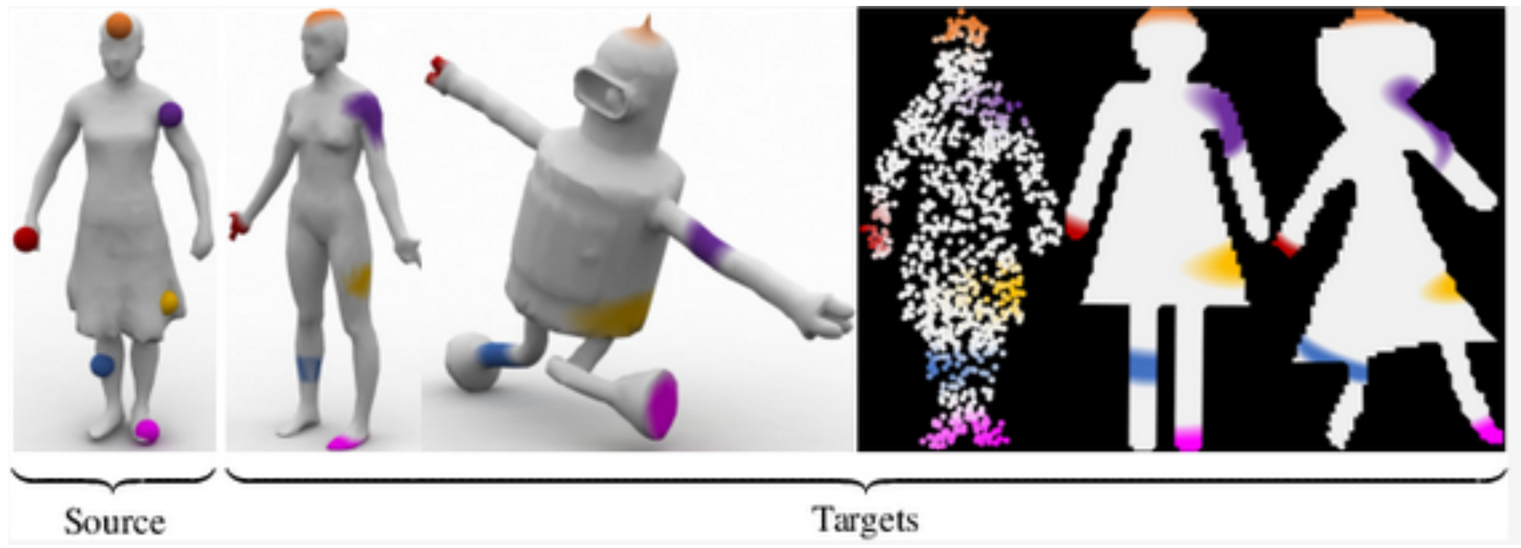
Needs a common metric space: Gromov-Wasserstein distance on stage [Memoli 2011]

$$GW(\boldsymbol{C}_X,\boldsymbol{C}_Y,\boldsymbol{h},\boldsymbol{g}) = \min_{\boldsymbol{T}\in\boldsymbol{\Pi}(\boldsymbol{h},\boldsymbol{g})} \langle L(\boldsymbol{C}_X,\boldsymbol{C}_Y)\otimes \boldsymbol{T},\boldsymbol{T}\rangle = \min_{\boldsymbol{T}\in\boldsymbol{\Pi}(\boldsymbol{h},\boldsymbol{g})} \sum_{i,j,k,l} L(d_X(\boldsymbol{x}_i,\boldsymbol{x}_k), d_Y(\boldsymbol{y}_j,\boldsymbol{y}_l)) T_{i,j}T_{k,l}$$

Useful for comparing graphs (FGW [Vayer 2019])

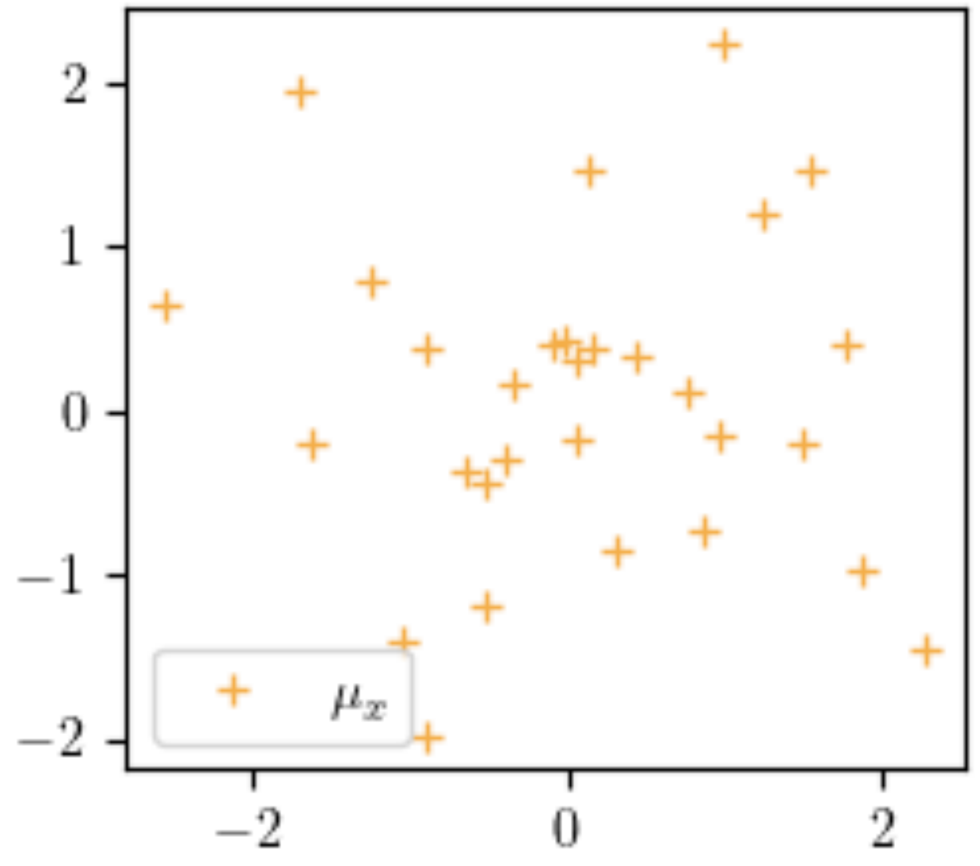

$\mathcal{G}_1$
k
$|d_X(\boldsymbol{x}_i,\boldsymbol{x}_k) - d_Y(\boldsymbol{y}_j,\boldsymbol{y}_l)|$
l
$\mathcal{G}_2$
i
j
$d(\boldsymbol{a}_i,\boldsymbol{b}_j)$

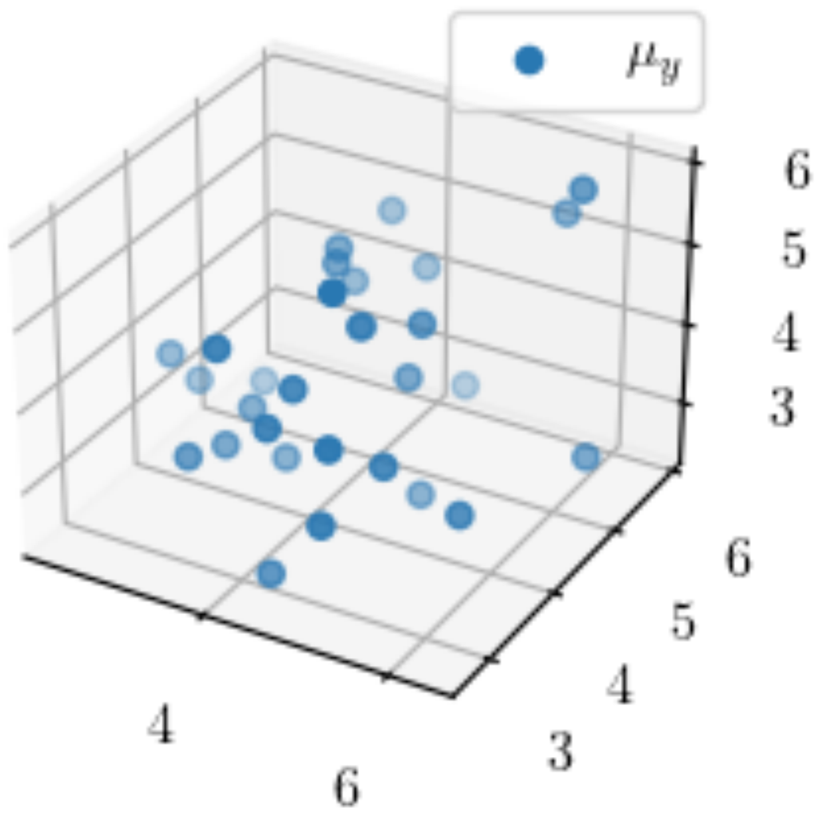
or for shape registration [Peyré 2016]



Gromov-Wasserstein

Unregistered spaces





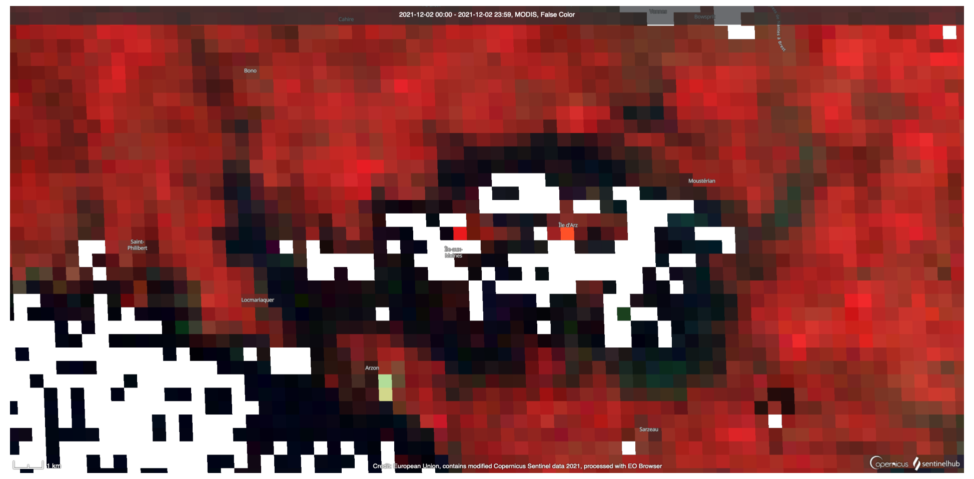
Cost $c(x, y)$?

$$\mu_x \in \Omega_x, \quad \mu_y \in \Omega_y$$

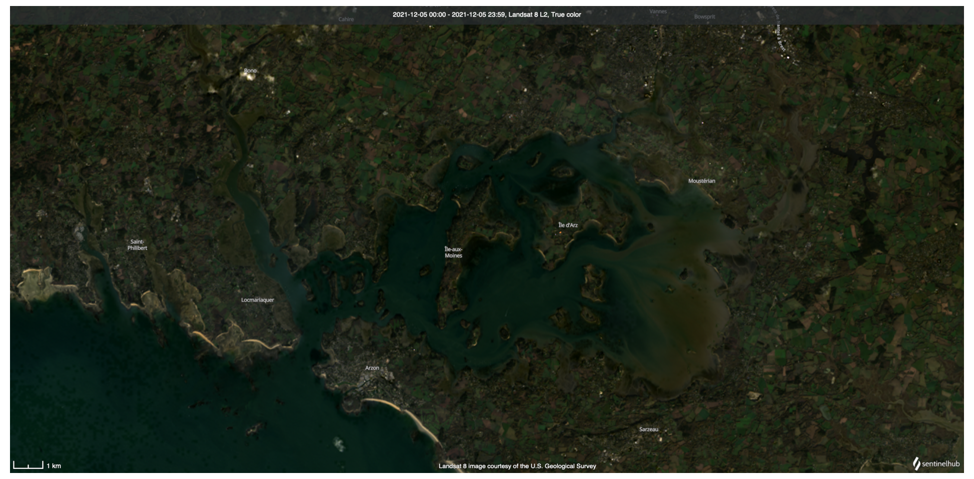
Related but not registered objects
e.g. same object observed by
different modalities

Gromov-Wasserstein

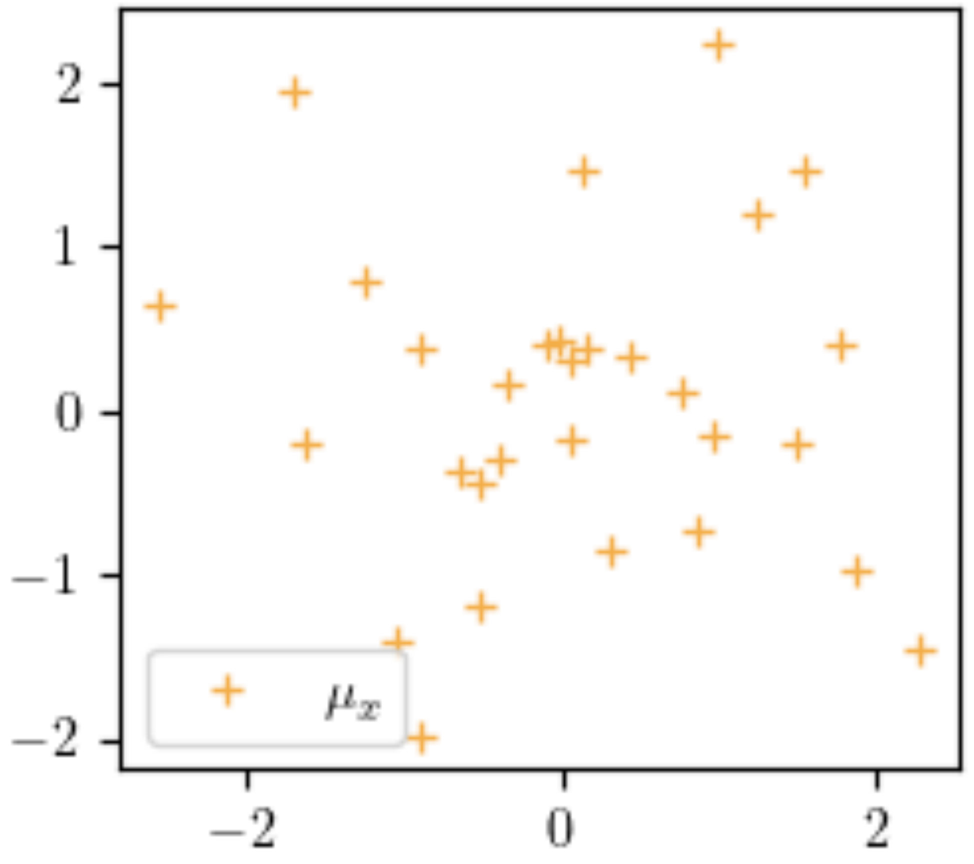
Unregistered spaces

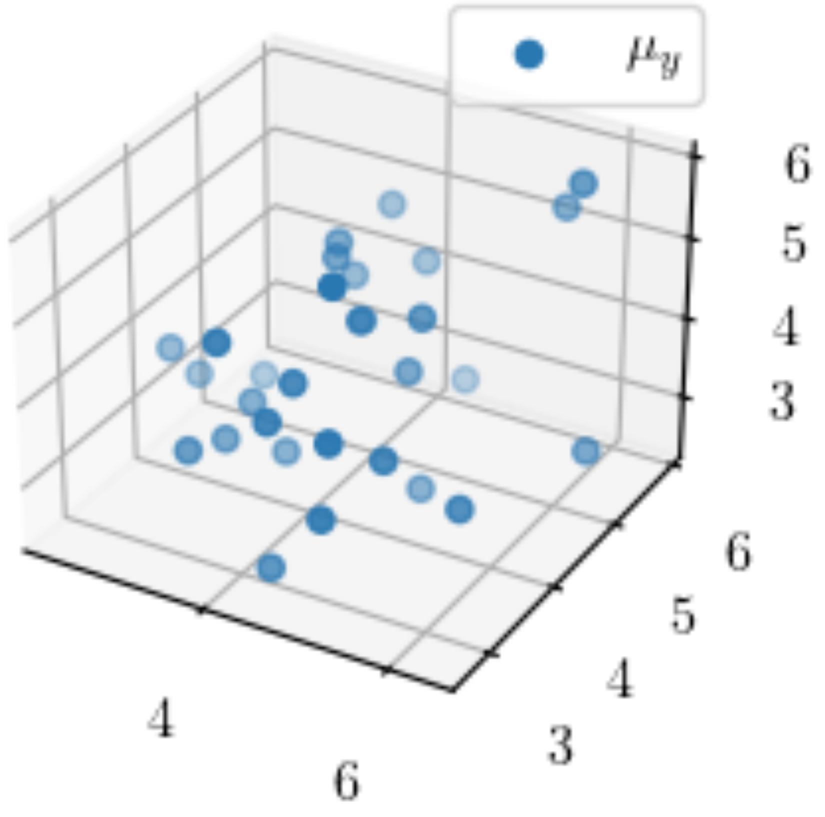


MODIS (36 bands)



Landsat8 (11 bands)





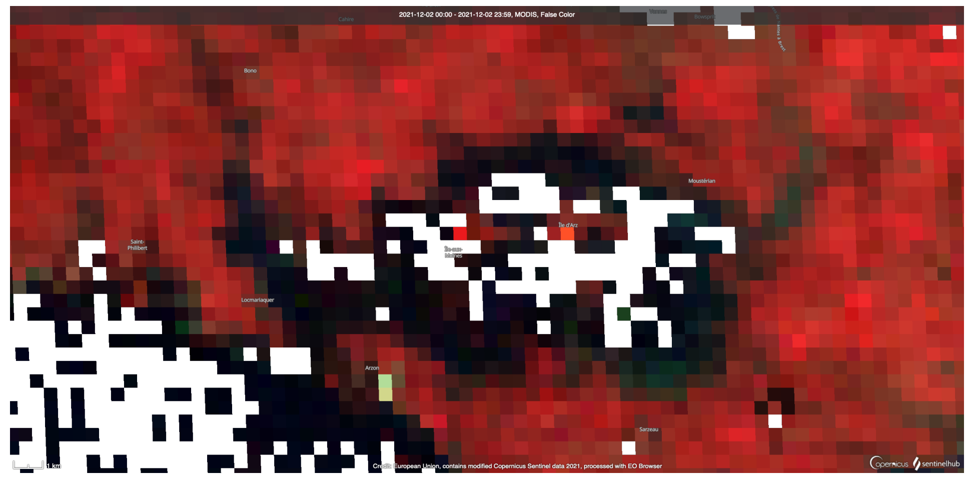
Cost $c(x, y)$?

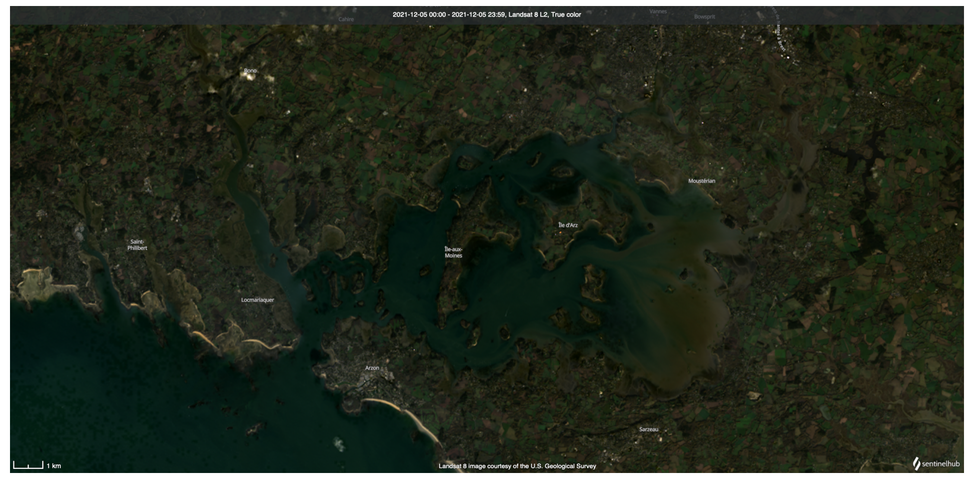
$$\mu_x \in \Omega_x, \ \mu_y \in \Omega_y$$

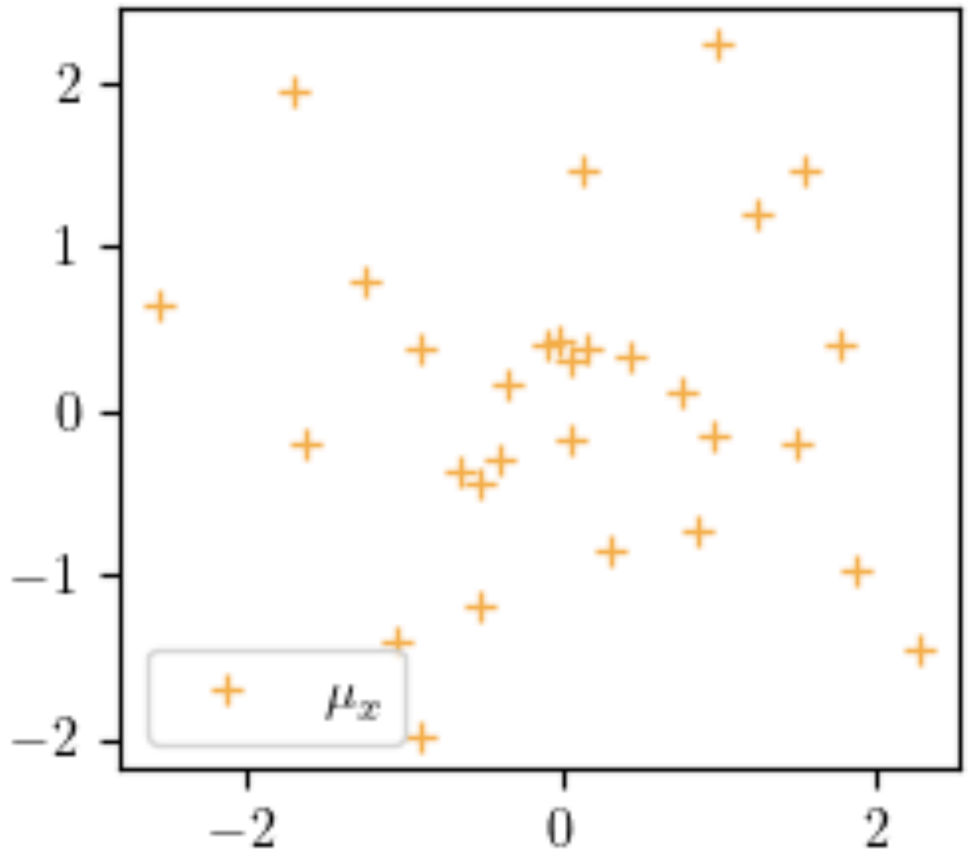
Related but not registered objects e.g. same object observed by different modalities

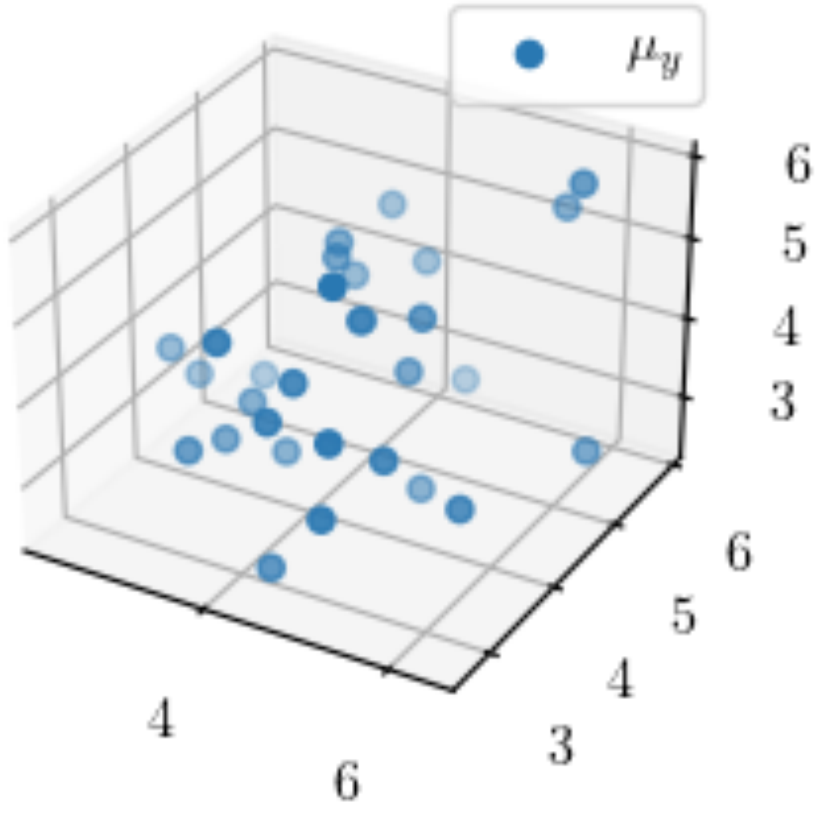
Gromov-Wasserstein

Unregistered spaces


MODIS (36 bands)


Landsat8 (11 bands)



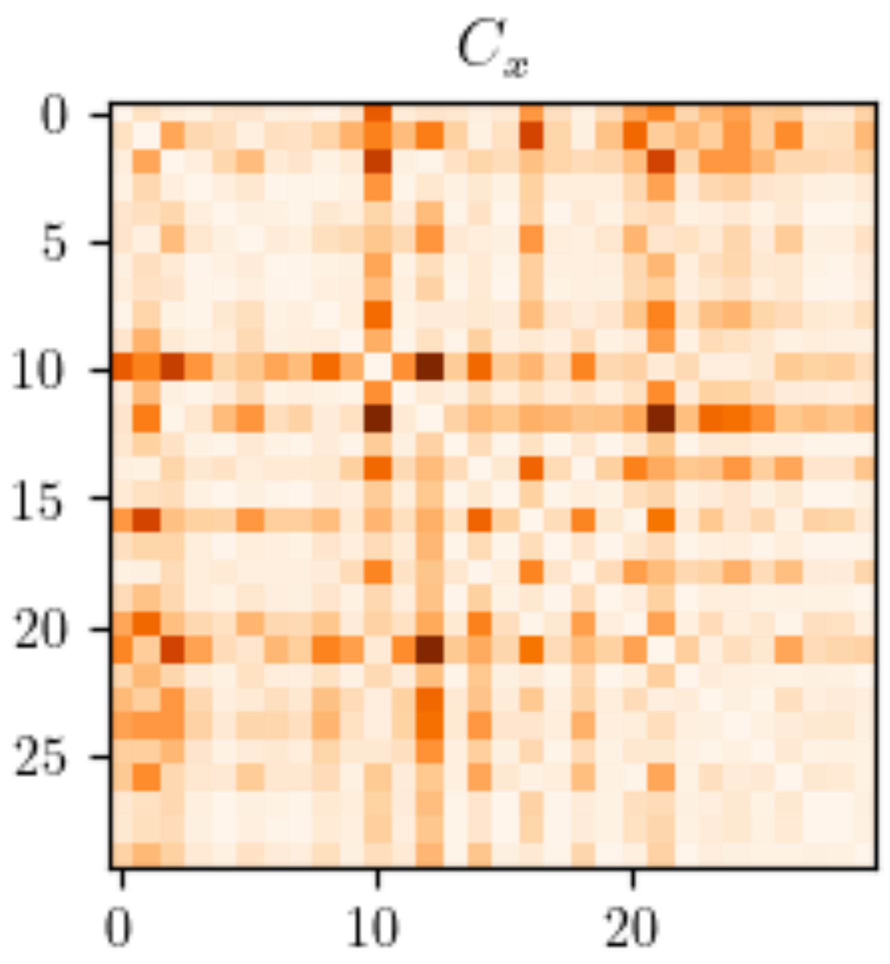


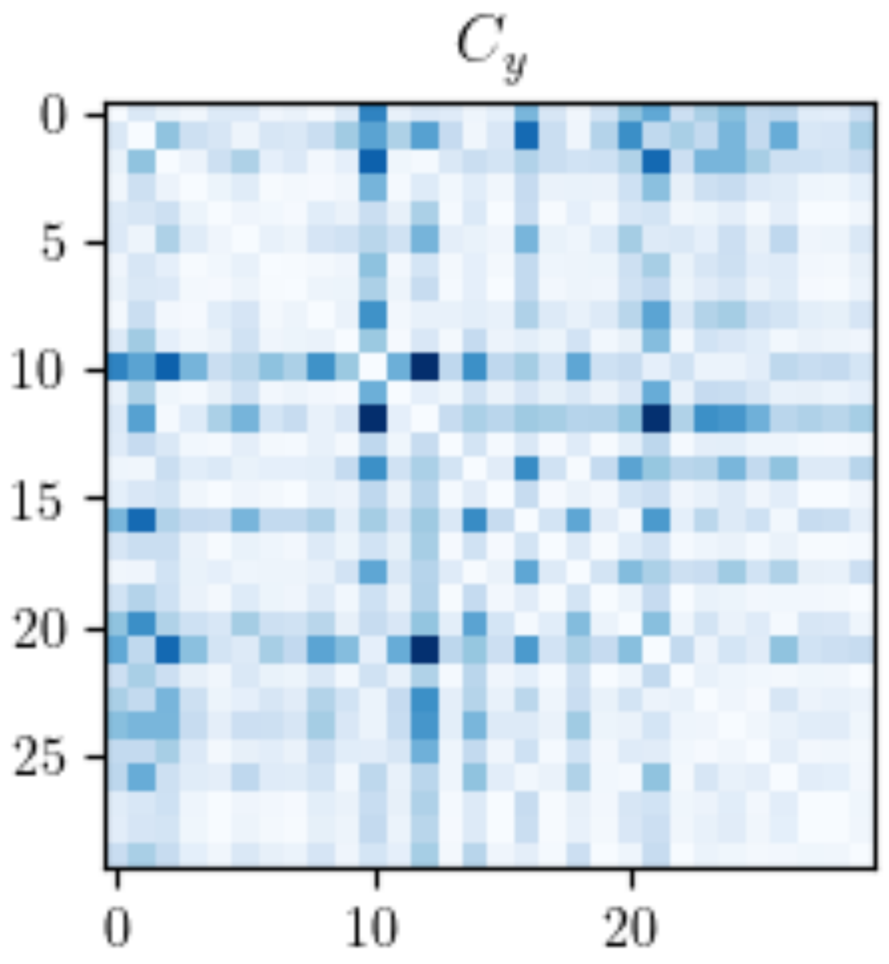
$$\mu_x \in \Omega_x, \quad \mu_y \in \Omega_y$$

Related but not registered objects e.g. same object observed by different modalities

Cost $c(x, y)$?

$$c_x(x_i, x_k) \in \mathbb{R}^{n\times n}, c_y(y_j, y_l) \in \mathbb{R}^{m\times m}$$





Gromov-Wasserstein

$$GW(\boldsymbol{C}_X, \boldsymbol{C}_Y, \boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \in \boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g})} \langle L(\boldsymbol{C}_X, \boldsymbol{C}_Y) \otimes \boldsymbol{T}, \boldsymbol{T} \rangle = \min_{\boldsymbol{T} \in \boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g})} \sum_{i,j,k,l} L(d_X(\boldsymbol{x}_i, \boldsymbol{x}_k), d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)) T_{i,j} T_{k,l}$$

Gromov-Wasserstein

coupling matrix

$$GW(\boldsymbol{C}_X, \boldsymbol{C}_Y, \boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T}\in\boldsymbol{\Pi}(\boldsymbol{h},\boldsymbol{g})} \langle L(\boldsymbol{C}_X, \boldsymbol{C}_Y) \otimes \boldsymbol{T}, \boldsymbol{T} \rangle = \min_{\boldsymbol{T}\in\boldsymbol{\Pi}(\boldsymbol{h},\boldsymbol{g})} \sum_{i,j,k,l} L(d_X(\boldsymbol{x}_i, \boldsymbol{x}_k), d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)) T_{i,j} T_{k,l}$$

marginal constraints

Gromov-Wasserstein

$$GW(\boldsymbol{C}_X, \boldsymbol{C}_Y, \boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \in \boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g})} \langle L(\boldsymbol{C}_X, \boldsymbol{C}_Y) \otimes \boldsymbol{T}, \boldsymbol{T} \rangle = \min_{\boldsymbol{T} \in \boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g})} \sum_{i,j,k,l} L(d_X(\boldsymbol{x}_i, \boldsymbol{x}_k), d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)) T_{i,j} T_{k,l}$$

Gromov-Wasserstein

4d-tensor

$$GW(\mathbf{C}_X, \mathbf{C}_Y, \mathbf{h}, \mathbf{g}) = \min_{\mathbf{T}\in\mathbf{\Pi}(\mathbf{h},\mathbf{g})} \langle L(\mathbf{C}_X, \mathbf{C}_Y) \otimes \mathbf{T}, \mathbf{T}\rangle = \min_{\mathbf{T}\in\mathbf{\Pi}(\mathbf{h},\mathbf{g})} \sum_{i,j,k,l} L(d_X(\mathbf{x}_i, \mathbf{x}_k), d_Y(\mathbf{y}_j, \mathbf{y}_l)) T_{i,j} T_{k,l}$$

quadratic problem

Gromov-Wasserstein

4d-tensor

$$GW(\boldsymbol{C}_X,\boldsymbol{C}_Y,\boldsymbol{h},\boldsymbol{g}) = \min_{\boldsymbol{T}\in\boldsymbol{\Pi}(\boldsymbol{h},\boldsymbol{g})} \langle L(\boldsymbol{C}_X,\boldsymbol{C}_Y)\otimes \boldsymbol{T},\boldsymbol{T}\rangle = \min_{\boldsymbol{T}\in\boldsymbol{\Pi}(\boldsymbol{h},\boldsymbol{g})} \sum_{i,j,k,l} L(d_X(\boldsymbol{x}_i,\boldsymbol{x}_k), d_Y(\boldsymbol{y}_j,\boldsymbol{y}_l))T_{i,j}T_{k,l}$$

quadratic problem

$$L(d_X(\boldsymbol{x}_i,\boldsymbol{x}_k), d_Y(\boldsymbol{y}_j,\boldsymbol{y}_l)) = |d_X(\boldsymbol{x}_i,\boldsymbol{x}_k) - d_Y(\boldsymbol{y}_j,\boldsymbol{y}_l)|^p$$

Gromov-Wasserstein

4d-tensor

$$GW(\boldsymbol{C}_X, \boldsymbol{C}_Y, \boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \in \boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g})} \langle L(\boldsymbol{C}_X, \boldsymbol{C}_Y) \otimes \boldsymbol{T}, \boldsymbol{T} \rangle = \min_{\boldsymbol{T} \in \boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g})} \sum_{i,j,k,l} L(d_X(\boldsymbol{x}_i, \boldsymbol{x}_k), d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)) T_{i,j} T_{k,l}$$

quadratic problem

$$\downarrow$$

$$L(d_X(\boldsymbol{x}_i, \boldsymbol{x}_k), d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)) = |d_X(\boldsymbol{x}_i, \boldsymbol{x}_k) - d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)|^p$$

Search for an OT plan that preserve the pairwise relationships between samples
→avoid couplings when $|d_X(\boldsymbol{x}_i, \boldsymbol{x}_k) - d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)|^p$ is large

Gromov-Wasserstein

4d-tensor

$$GW(\boldsymbol{C}_X, \boldsymbol{C}_Y, \boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T}\in\boldsymbol{\Pi}(\boldsymbol{h},\boldsymbol{g})} \langle L(\boldsymbol{C}_X, \boldsymbol{C}_Y) \otimes \boldsymbol{T}, \boldsymbol{T} \rangle = \min_{\boldsymbol{T}\in\boldsymbol{\Pi}(\boldsymbol{h},\boldsymbol{g})} \sum_{i,j,k,l} L(d_X(\boldsymbol{x}_i, \boldsymbol{x}_k), d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)) T_{i,j} T_{k,l}$$

quadratic problem

$$L(d_X(\boldsymbol{x}_i, \boldsymbol{x}_k), d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)) = |d_X(\boldsymbol{x}_i, \boldsymbol{x}_k) - d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)|^p$$

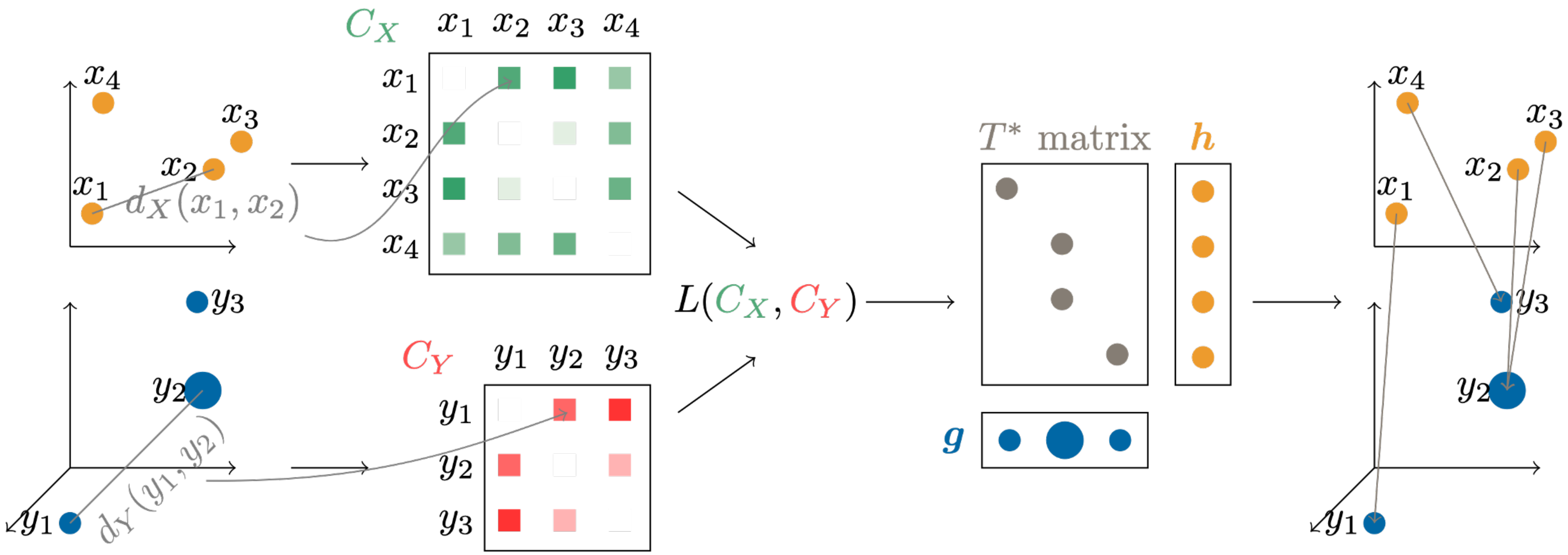
Gromov-Wasserstein

4d-tensor

$$GW(\boldsymbol{C}_X, \boldsymbol{C}_Y, \boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T}\in\boldsymbol{\Pi}(\boldsymbol{h},\boldsymbol{g})} \langle L(\boldsymbol{C}_X, \boldsymbol{C}_Y) \otimes \boldsymbol{T}, \boldsymbol{T}\rangle = \min_{\boldsymbol{T}\in\boldsymbol{\Pi}(\boldsymbol{h},\boldsymbol{g})} \sum_{i,j,k,l} L(d_X(\boldsymbol{x}_i, \boldsymbol{x}_k), d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)) T_{i,j} T_{k,l}$$

quadratic problem

$$L(d_X(\boldsymbol{x}_i, \boldsymbol{x}_k), d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)) = |d_X(\boldsymbol{x}_i, \boldsymbol{x}_k) - d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)|^p$$



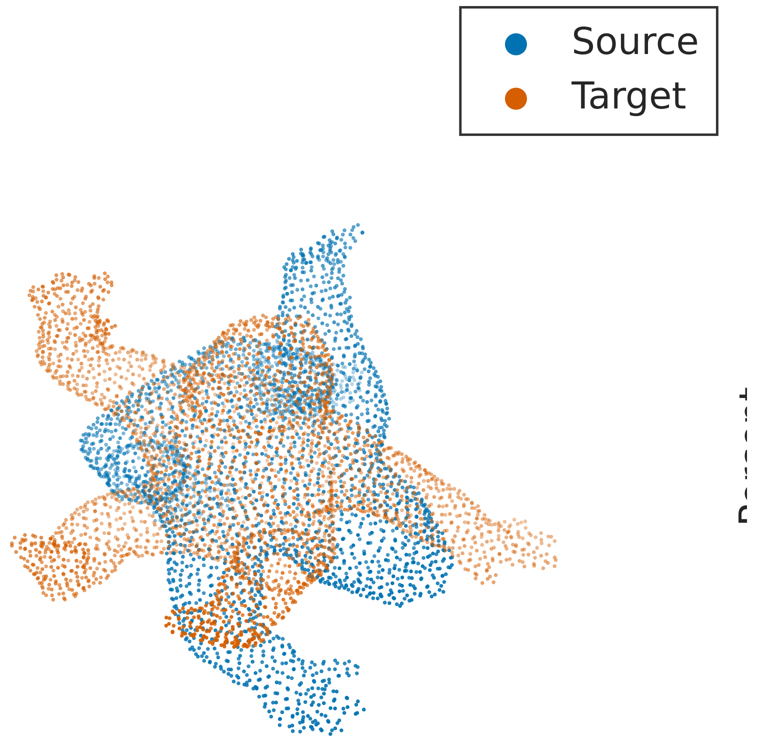
Gromov-Wasserstein

Gromov-Wasserstein

$$GW(\boldsymbol{C}_X, \boldsymbol{C}_Y, \boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \in \boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g})} \langle L(\boldsymbol{C}_X, \boldsymbol{C}_Y) \otimes \boldsymbol{T}, \boldsymbol{T} \rangle = \min_{\boldsymbol{T} \in \boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g})} \sum_{i,j,k,l} L(d_X(\boldsymbol{x}_i, \boldsymbol{x}_k), d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)) T_{i,j} T_{k,l}$$

GW is a quadratic problem: complexity $O(n^4)$
and is not a convex problem

Invariant to isometries such that rotations and translations



$$GW(\boldsymbol{C}_X, \boldsymbol{C}_{X^R}, \boldsymbol{h}, \boldsymbol{h}) = 0$$

Gromov-Wasserstein

Solving the problem

$$GW(\boldsymbol{C}_X, \boldsymbol{C}_Y, \boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \in \boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g})} \langle L(\boldsymbol{C}_X, \boldsymbol{C}_Y) \otimes \boldsymbol{T}, \boldsymbol{T} \rangle = \min_{\boldsymbol{T} \in \boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g})} \sum_{i,j,k,l} L(d_X(\boldsymbol{x}_i, \boldsymbol{x}_k), d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)) T_{i,j} T_{k,l}$$

Optimization algorithms

Local solutions can be obtained with a Frank-Wolfe algorithm [Vayer 2018]
Iterative algorithm, which solves at each step an OT problem

For the entropic version, local solutions can be obtained with a KL mirror descent [Peyré 2016]
Iterative algorithm, which solves at each step a Sinkhorn problem

Gromov-Wasserstein

Solving the problem

$$GW(\boldsymbol{C}_X, \boldsymbol{C}_Y, \boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \in \boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g})} \langle L(\boldsymbol{C}_X, \boldsymbol{C}_Y) \otimes \boldsymbol{T}, \boldsymbol{T} \rangle = \min_{\boldsymbol{T} \in \boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g})} \sum_{i,j,k,l} L(d_X(\boldsymbol{x}_i, \boldsymbol{x}_k), d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)) T_{i,j} T_{k,l}$$

Optimization algorithms

Local solutions can be obtained with a Frank-Wolfe algorithm [Vayer 2018]
Iterative algorithm, which solves at each step an OT problem

Solve several iterations of
a $O(n^3)$ problem

For the entropic version, local solutions can be obtained with a KL mirror descent [Peyré 2016]
Iterative algorithm, which solves at each step a Sinkhorn problem

Solve several iterations of
a $O(n^2)$ problem

Gromov-Wasserstein

Solving the problem

$$GW(\boldsymbol{C}_X, \boldsymbol{C}_Y, \boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \in \boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g})} \langle L(\boldsymbol{C}_X, \boldsymbol{C}_Y) \otimes \boldsymbol{T}, \boldsymbol{T} \rangle = \min_{\boldsymbol{T} \in \boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g})} \sum_{i,j,k,l} L(d_X(\boldsymbol{x}_i, \boldsymbol{x}_k), d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)) T_{i,j} T_{k,l}$$

Optimization algorithms

Local solutions can be obtained with a Frank-Wolfe algorithm [Vayer 2018]
Iterative algorithm, which solves at each step an OT problem

Solve several iterations of
a $O(n^3)$ problem

For the entropic version, local solutions can be obtained with a KL mirror descent [Peyré 2016]
Iterative algorithm, which solves at each step a Sinkhorn problem

Solve several iterations of
a $O(n^2)$ problem

Difficult (non convex) and costly problem to solve
Approximations exist, but still an open problem

Unbalanced Gromov-Wasserstein

Partial Gromov-Wasserstein (D_φ is L1)

$$\min_{\boldsymbol{T} \geq \mathbf{0}} \quad \langle L(\boldsymbol{C}_X, \boldsymbol{C}_Y) \otimes \boldsymbol{T}, \boldsymbol{T} \rangle + \lambda \left(D_\varphi(\boldsymbol{T} 1_m, \boldsymbol{h}) + D_\varphi(\boldsymbol{T}^\top 1_n, \boldsymbol{g}) \right)$$

Partial OT: fix the amount of mass s that has to be transported

Exact partial-GW can be computed by solving Frank Wolfe iterations with partial-W [Chapel 2020]

Unbalanced Gromov-Wasserstein

Partial Gromov-Wasserstein (D_φ is L1)

$$\min_{\boldsymbol{T}\geq \boldsymbol{0}} \langle L(\boldsymbol{C}_X, \boldsymbol{C}_Y)\otimes \boldsymbol{T}, \boldsymbol{T}\rangle + \lambda \left(D_\varphi(\boldsymbol{T}1_m, \boldsymbol{h}) + D_\varphi(\boldsymbol{T}^\top 1_n, \boldsymbol{g})\right)$$

same penalization as for the UOT problem

Partial OT: fix the amount of mass s that has to be transported

Exact partial-GW can be computed by solving Frank Wolfe iterations with partial-W [Chapel 2020]

Unbalanced Gromov-Wasserstein

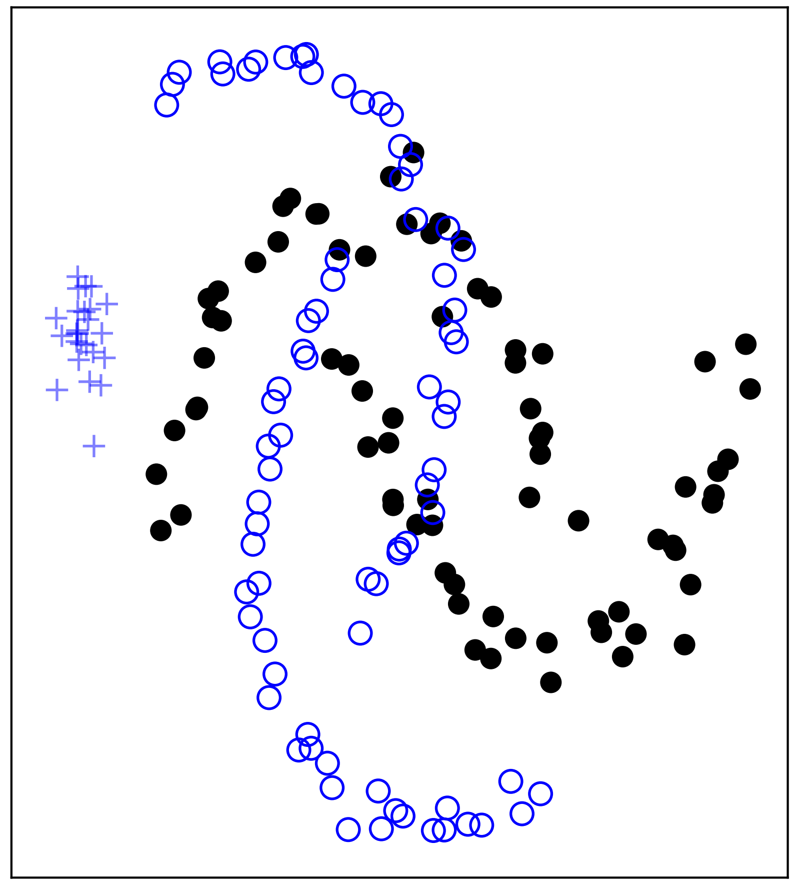
Partial Gromov-Wasserstein (D_φ is L1)

$$\min_{\boldsymbol{T}\geq\mathbf{0}} \quad \langle L(\boldsymbol{C}_X, \boldsymbol{C}_Y)\otimes \boldsymbol{T}, \boldsymbol{T}\rangle + \lambda \boxed{\left(D_\varphi(\boldsymbol{T}1_m, \boldsymbol{h}) + D_\varphi(\boldsymbol{T}^\top 1_n, \boldsymbol{g})\right)}$$

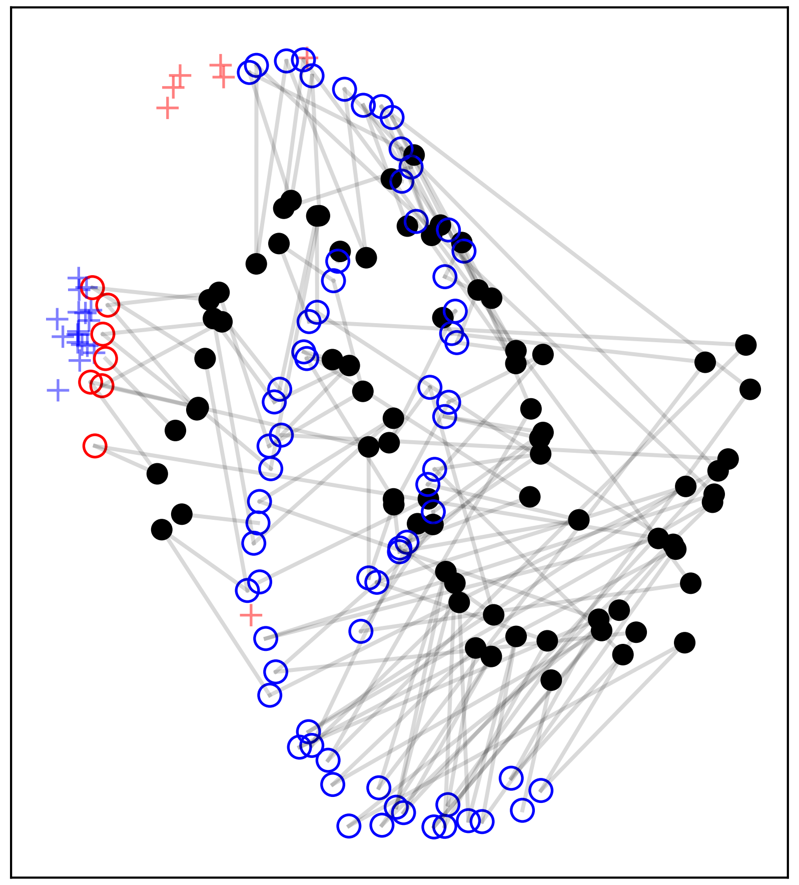
same penalization as for the UOT problem

Partial OT: fix the amount of mass s that has to be transported

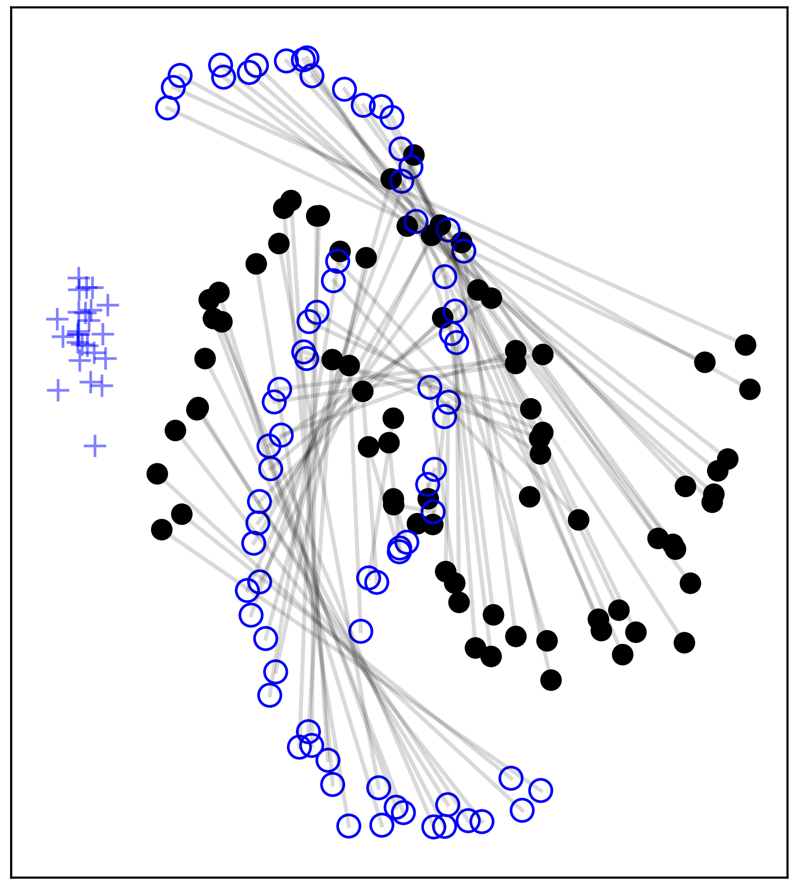
Exact partial-GW can be computed by solving Frank Wolfe iterations with partial-W [Chapel 2020]



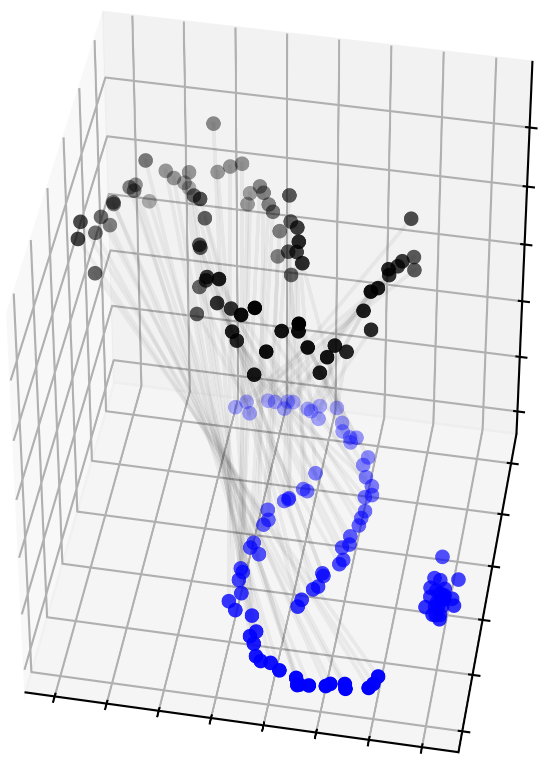
source and target distributions



partial-W



partial-GW



partial-GW

Unbalanced Gromov-Wasserstein

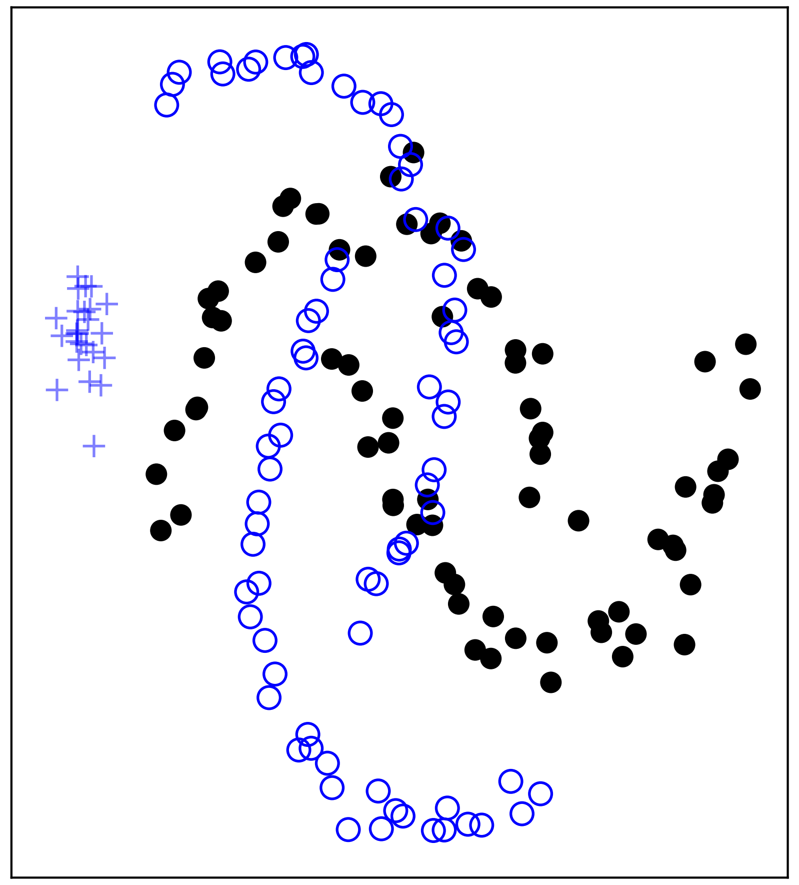
Partial Gromov-Wasserstein (D_φ is L1)

$$\min_{\boldsymbol{T}\geq \boldsymbol{0}} \quad \langle L(\boldsymbol{C}_X, \boldsymbol{C}_Y)\otimes \boldsymbol{T}, \boldsymbol{T}\rangle + \lambda \left(D_\varphi(\boldsymbol{T}1_m, \boldsymbol{h}) + D_\varphi(\boldsymbol{T}^\top 1_n, \boldsymbol{g})\right)$$

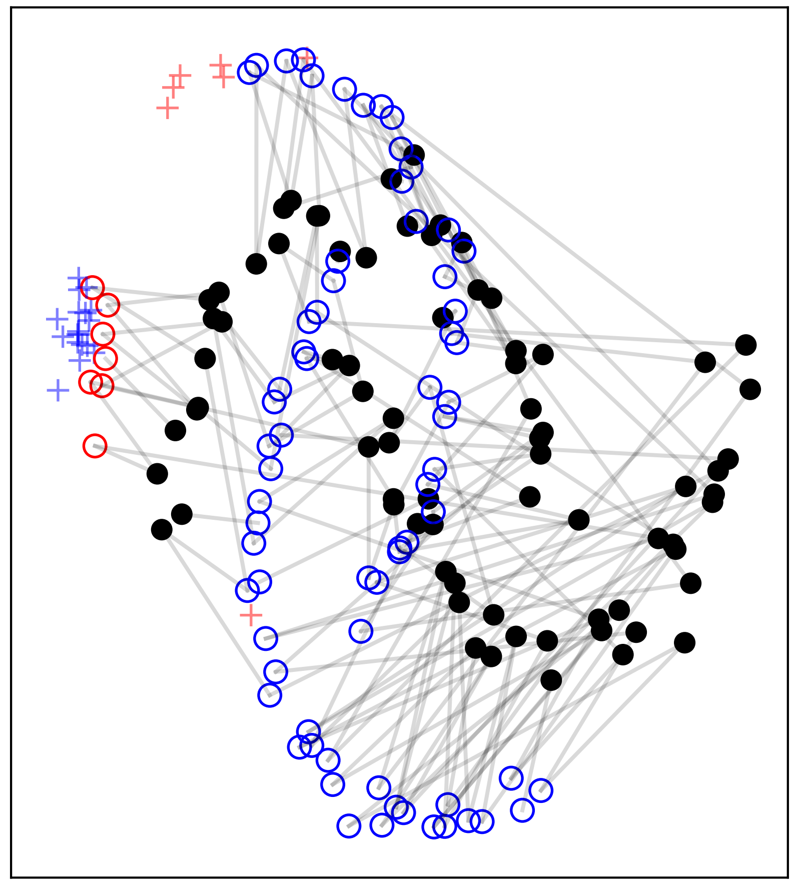
same penalization as for the UOT problem

Partial OT: fix the amount of mass s that has to be transported

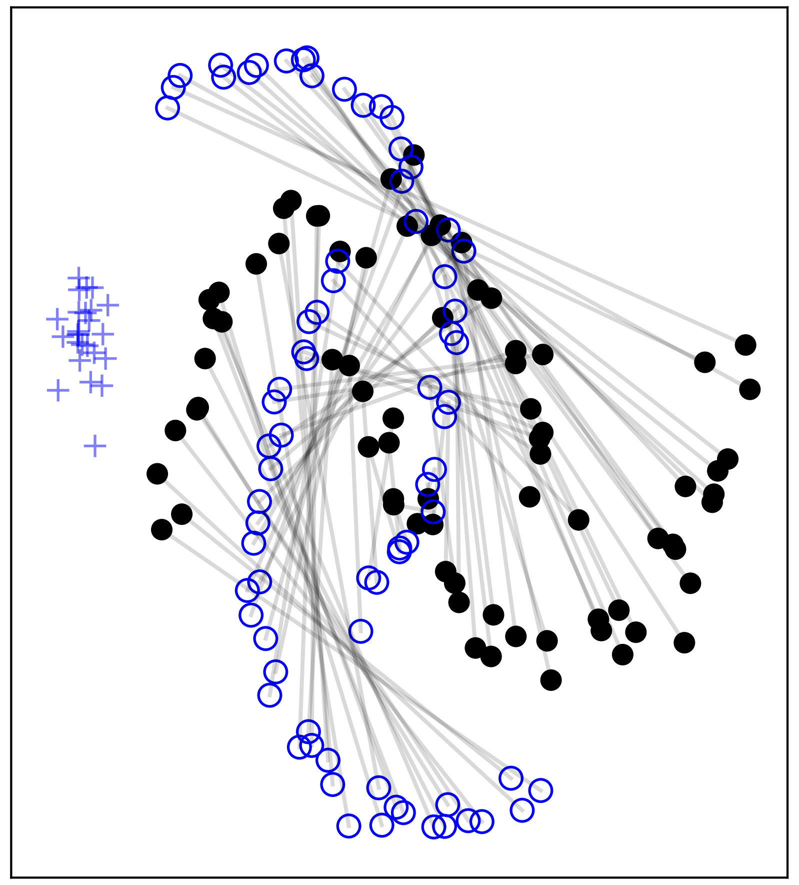
Exact partial-GW can be computed by solving Frank Wolfe iterations with partial-W [Chapel 2020]



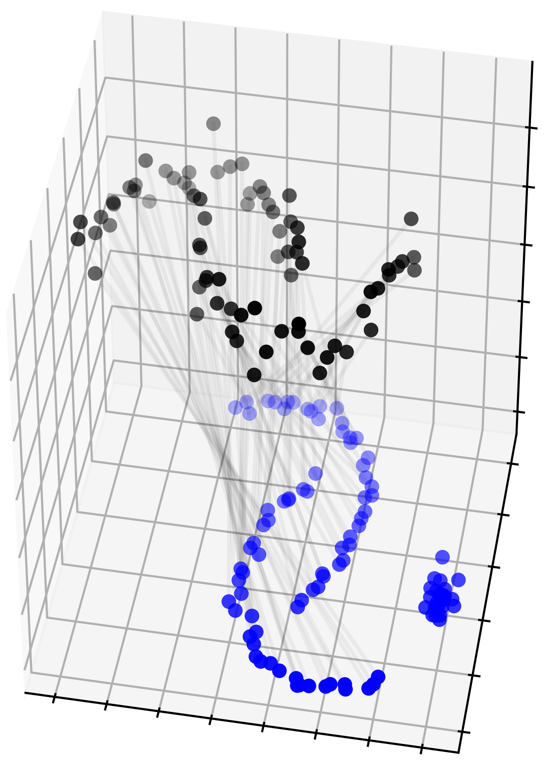
source and target distributions



partial-W



partial-GW



partial-GW

Solving a GW problem when the spaces are the same can be interesting as well (invariances)

Unbalanced Gromov-Wasserstein

Unbalanced Gromov-Wasserstein (D_φ is KL)

Can also consider quadratic penalties [Séjourné 2021], relying on Sinkhorn algorithm

$$\min_{\boldsymbol{T} \geq \mathbf{0}} \quad \langle L(\boldsymbol{C}_X, \boldsymbol{C}_Y) \otimes \boldsymbol{T}, \boldsymbol{T} \rangle + \lambda \left(D_\varphi(\boldsymbol{T} 1_m \otimes \boldsymbol{T} 1_m, \boldsymbol{h} \otimes \boldsymbol{h}) + D_\varphi(\boldsymbol{T}^\top 1_n \otimes \boldsymbol{T}^\top 1_n, \boldsymbol{g} \otimes \boldsymbol{g}) \right)$$

Unbalanced Gromov-Wasserstein

Unbalanced Gromov-Wasserstein (D_φ is KL)

Can also consider quadratic penalties [Séjourné 2021], relying on Sinkhorn algorithm

$$\min_{\boldsymbol{T}\geq\mathbf{0}} \quad \langle L(\boldsymbol{C}_X, \boldsymbol{C}_Y)\otimes \boldsymbol{T}, \boldsymbol{T}\rangle + \lambda\Big(D_\varphi(\boldsymbol{T}1_m \otimes \boldsymbol{T}1_m, \boldsymbol{h}\otimes\boldsymbol{h}) + D_\varphi(\boldsymbol{T}^\top 1_n \otimes \boldsymbol{T}^\top 1_n, \boldsymbol{g}\otimes\boldsymbol{g})\Big)$$

quadratic problem: quadratic penalties

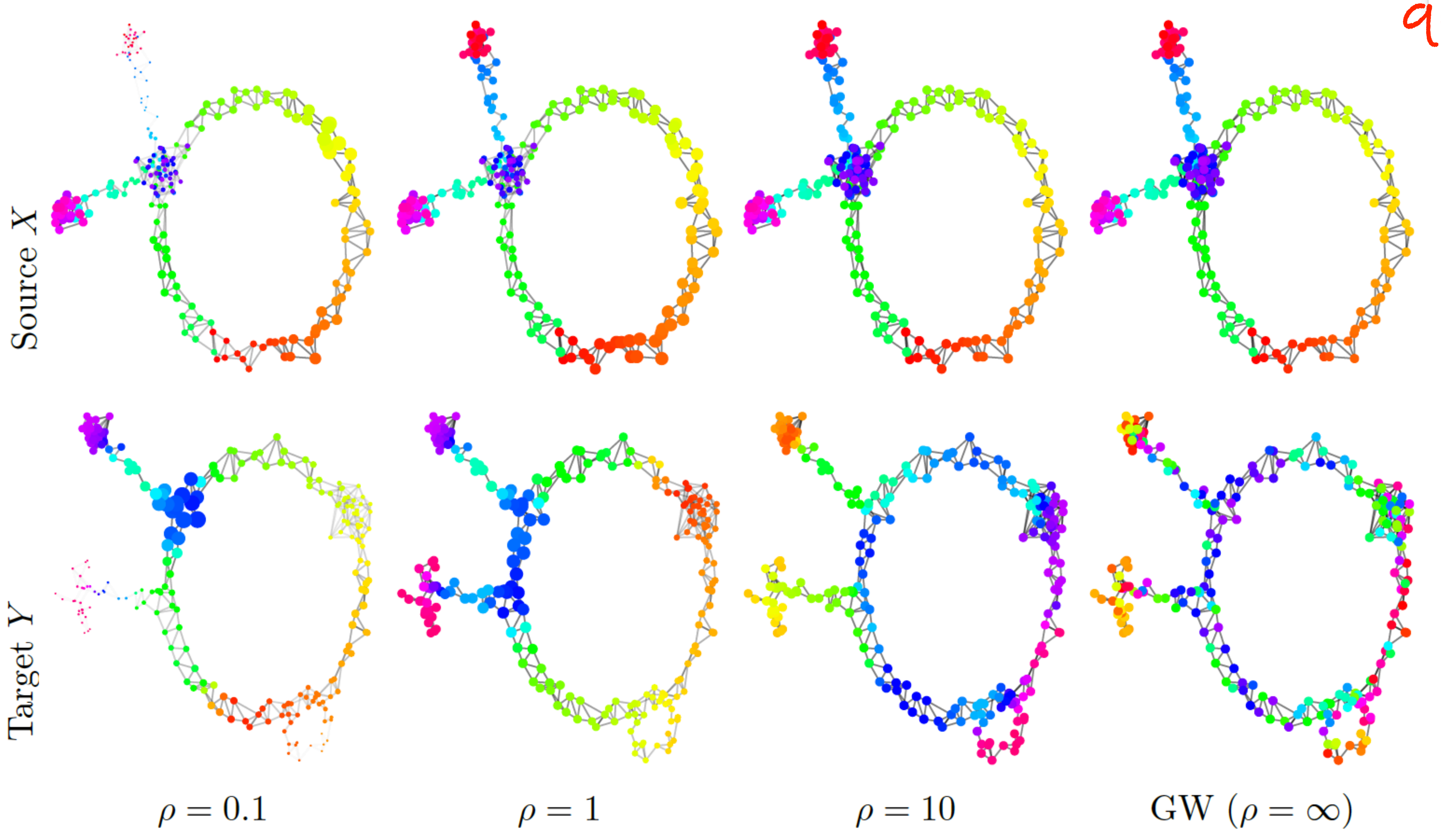
Unbalanced Gromov-Wasserstein

Unbalanced Gromov-Wasserstein (D_φ is KL)

Can also consider quadratic penalties [Séjourné 2021], relying on Sinkhorn algorithm

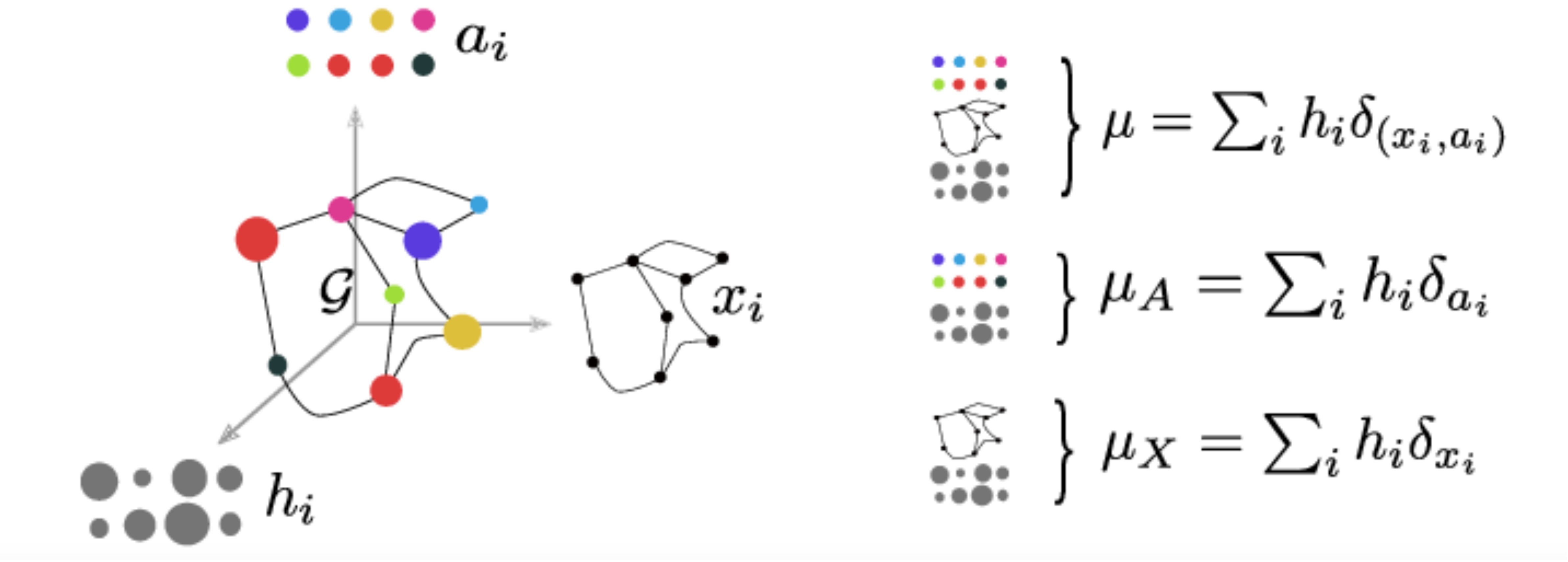
$$\min_{\boldsymbol{T}\geq\boldsymbol{0}} \quad \langle L(\boldsymbol{C}_X, \boldsymbol{C}_Y)\otimes \boldsymbol{T}, \boldsymbol{T}\rangle + \lambda\Big(D_\varphi(\boldsymbol{T}1_m\otimes \boldsymbol{T}1_m, \boldsymbol{h}\otimes \boldsymbol{h}) + D_\varphi(\boldsymbol{T}^\top 1_n\otimes \boldsymbol{T}^\top 1_n, \boldsymbol{g}\otimes \boldsymbol{g})\Big)$$

quadratic problem: quadratic penalties



Fused Gromov-Wasserstein

Labeled Graphs as probability distributions


a_i
$\mathcal{G}$
x_i
h_i
$\mu = \sum_i h_i \delta_{(x_i, a_i)}$
$\mu_A = \sum_i h_i \delta_{a_i}$
$\mu_X = \sum_i h_i \delta_{x_i}$

Nodes are weighted by their mass h_i

Features a_i can be compared through a common metric

No common metric between the structure x_i of two graphs

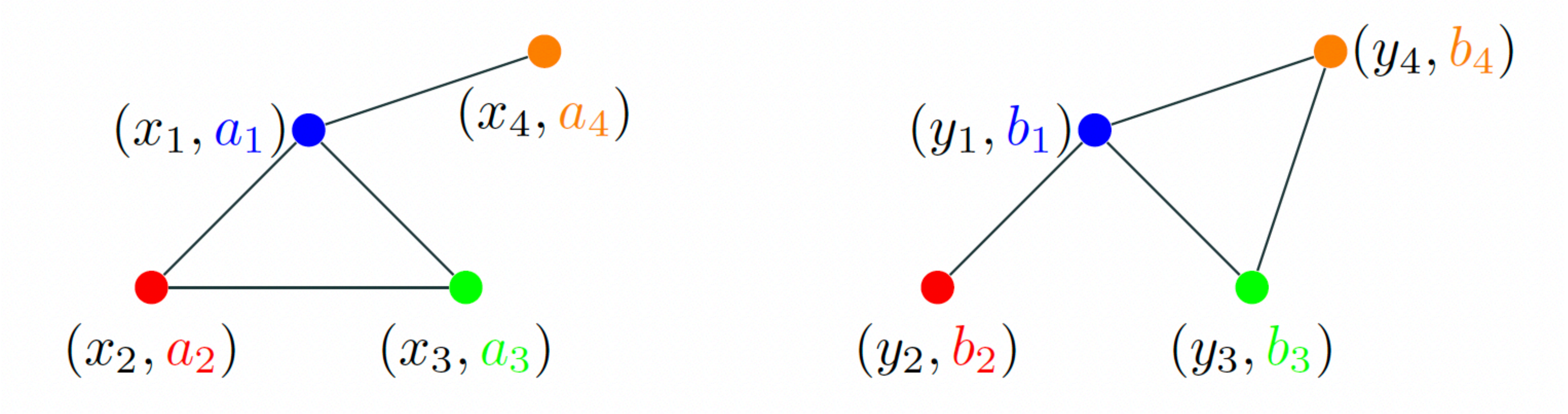
Fused Gromov-Wasserstein

Labeled Graphs as probability distributions

Two distributions $\mu_x = \sum_i h_i \delta_{(x_i, a_i)}$ and $\mu_y = \sum_i g_i \delta_{(y_i, b_i)}$

The fused Gromov-Wasserstein distance is defined as

$$FGW_{p,q,\alpha}^{q}(\boldsymbol{C}_X, \boldsymbol{C}_Y, \boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \in \boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g})} \sum_{ijkl} \left((1-\alpha)|a_i - b_j|^p + \alpha |d_X(\boldsymbol{x}_i, \boldsymbol{x}_k) - d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)|^p\right)^q T_{ik} T_{jl}$$



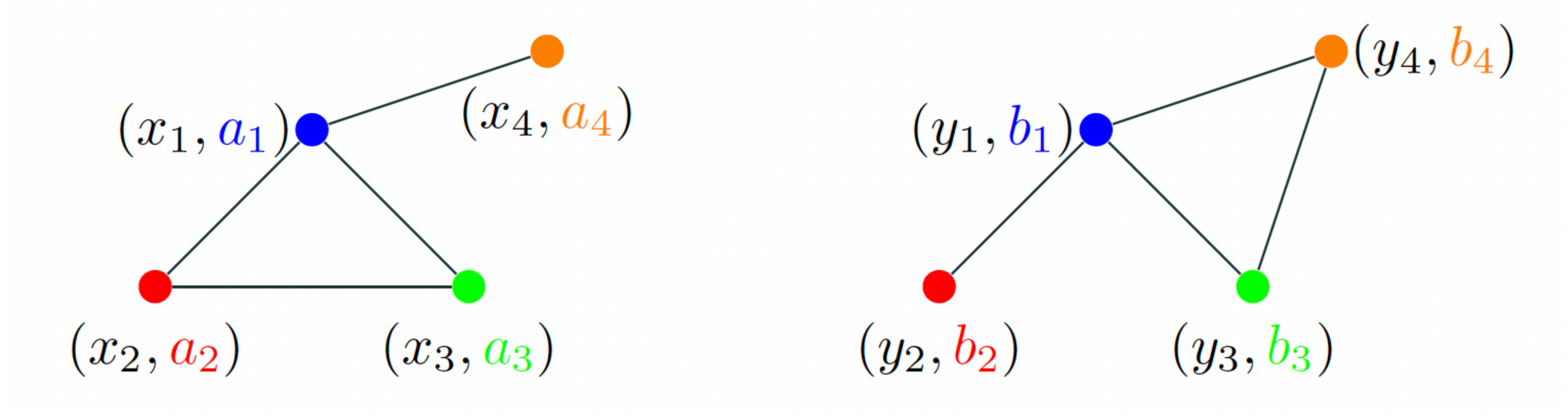
Fused Gromov-Wasserstein

Labeled Graphs as probability distributions

Two distributions $\mu_x = \sum_i h_i \delta_{(x_i,a_i)}$ and $\mu_y = \sum_i g_i \delta_{(y_i,b_i)}$

The fused Gromov-Wasserstein distance is defined as

$$FGW^q_{p,q,\alpha}(\boldsymbol{C}_X, \boldsymbol{C}_Y, \boldsymbol{h}, \boldsymbol{g}) = \min_{T \in \boldsymbol{\Pi}(\boldsymbol{h},\boldsymbol{g})} \sum_{ijkl} \left((1-\alpha)\underbrace{|a_i - b_j|^p}_{\text{Compares features}} + \underbrace{\alpha|d_X(\boldsymbol{x}_i, \boldsymbol{x}_k) - d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)|^p}_{\text{Compares structures}}\right)^q T_{ik}T_{jl}$$



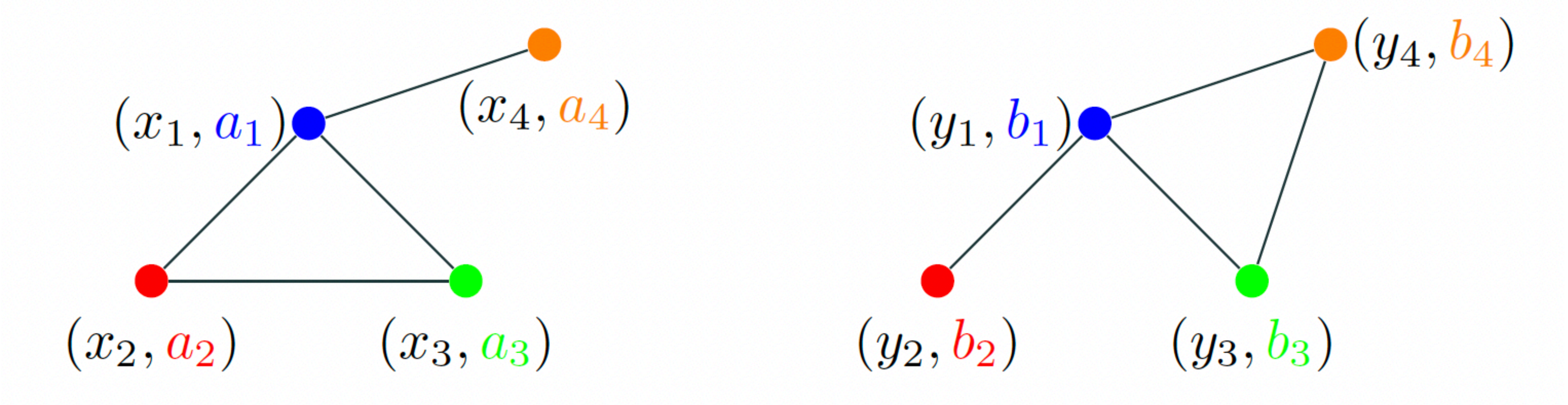
Fused Gromov-Wasserstein

Labeled Graphs as probability distributions

Two distributions $\mu_x = \sum_i h_i \delta_{(x_i, a_i)}$ and $\mu_y = \sum_i g_i \delta_{(y_i, b_i)}$

The fused Gromov-Wasserstein distance is defined as

$$FGW^q_{p,q,\alpha}(\boldsymbol{C}_X, \boldsymbol{C}_Y, \boldsymbol{h}, \boldsymbol{g}) = \min_{\boldsymbol{T} \in \boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g})} \sum_{ijkl} \left((1-\alpha)|a_i - b_j|^p + \alpha |d_X(\boldsymbol{x}_i, \boldsymbol{x}_k) - d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)|^p\right)^q T_{ik} T_{jl}$$



Fused Gromov-Wasserstein

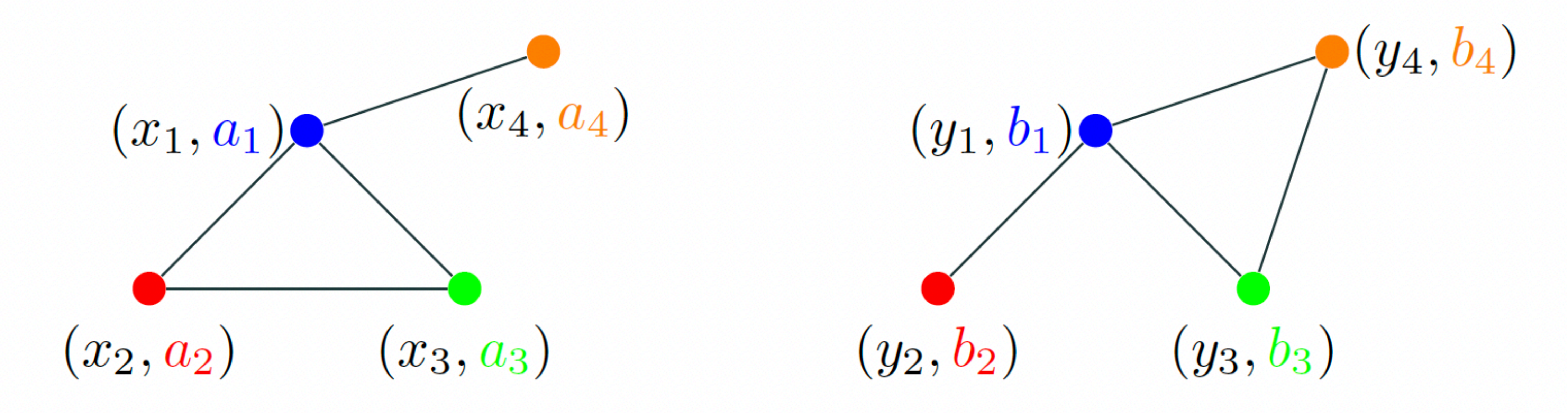
Labeled Graphs as probability distributions

Two distributions $\mu_x = \sum_i h_i \delta_{(x_i, a_i)}$ and $\mu_y = \sum_i g_i \delta_{(y_i, b_i)}$

The fused Gromov-Wasserstein distance is defined as

$$FGW^q_{p,q,\alpha}(\boldsymbol{C}_X, \boldsymbol{C}_Y, \boldsymbol{h}, \boldsymbol{g}) = \min_{T \in \boldsymbol{\Pi}(\boldsymbol{h},\boldsymbol{g})} \sum_{ijkl} \left(\boxed{(1-\alpha)} |a_i - b_j|^p + \boxed{\alpha} |d_X(\boldsymbol{x}_i, \boldsymbol{x}_k) - d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)|^p \right)^q T_{ik} T_{jl}$$

$\alpha \in [0,1]$



Fused Gromov-Wasserstein

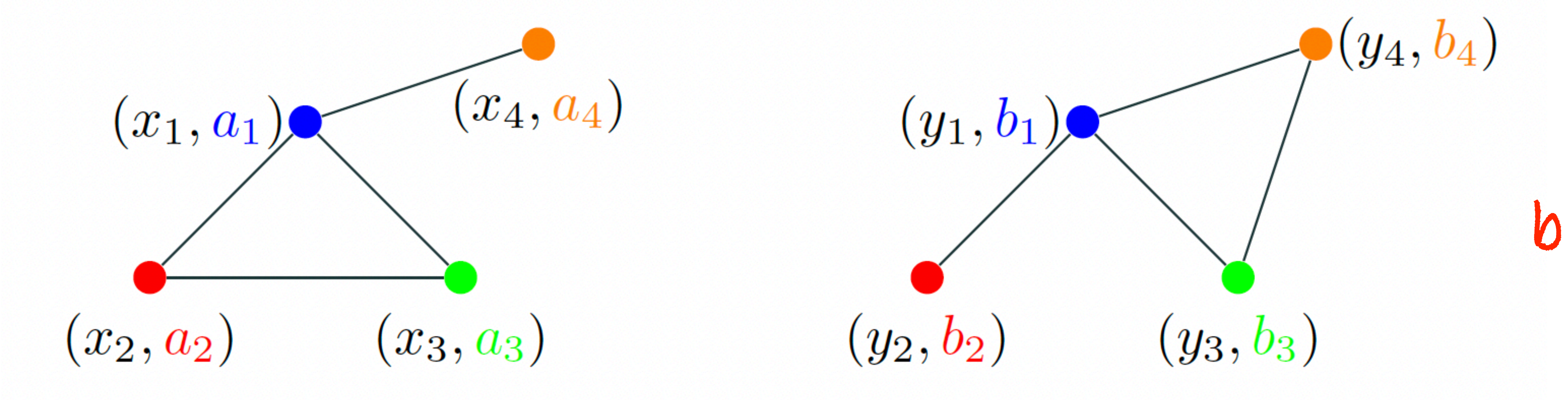
Labeled Graphs as probability distributions

Two distributions $\mu_x = \sum_i h_i \delta_{(x_i, a_i)}$ and $\mu_y = \sum_i g_i \delta_{(y_i, b_i)}$

The fused Gromov-Wasserstein distance is defined as

$$FGW^q_{p,q,\alpha}(\boldsymbol{C}_X, \boldsymbol{C}_Y, \boldsymbol{h}, \boldsymbol{g}) = \min_{T \in \boldsymbol{\Pi}(\boldsymbol{h}, \boldsymbol{g})} \sum_{ijkl} \left((1-\alpha)|a_i - b_j|^p + \alpha|d_X(\boldsymbol{x}_i, \boldsymbol{x}_k) - d_Y(\boldsymbol{y}_j, \boldsymbol{y}_l)|^p\right)^q T_{ik} T_{jl}$$

$\alpha \in [0,1]$



Same features OT(h,g)=0

same structure GW(h,g)=0

but different structure AND features

FGW(h,g)≠0

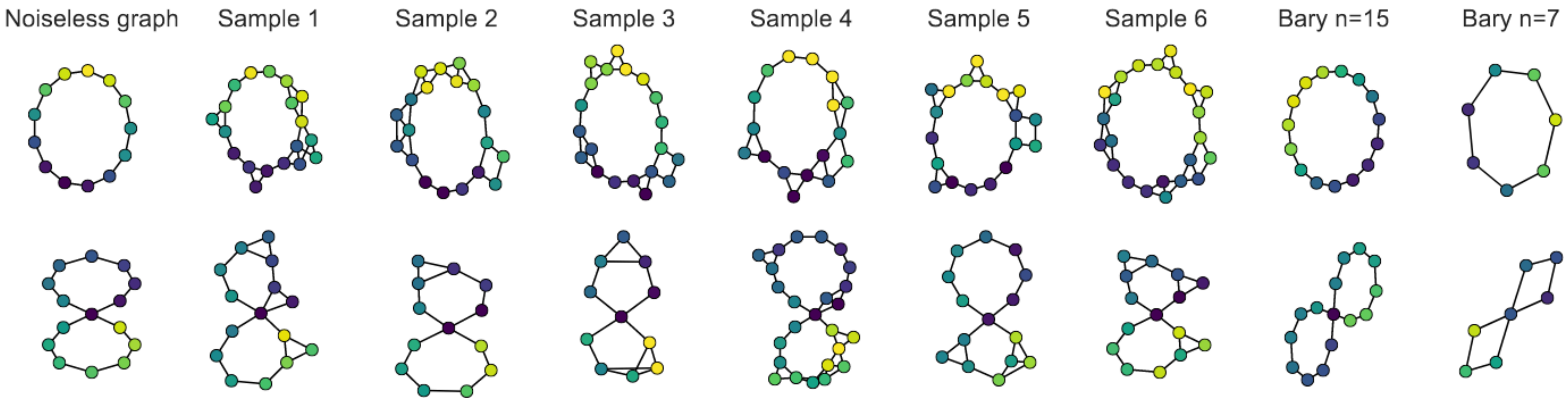
Fused Gromov-Wasserstein

FGW properties and barycenters

Interpolates between W ($\alpha = 0$) and GW ($\alpha = 1$)

It is a distance for $p = 1$

Constant speed geodesics can be defined



Outline

1. History and basics of optimal transport

2. Wasserstein distances

3. Computational OT

Practical session (with POT toolbox)

4. Variants of OT : unbalanced OT and Gromov-Wasserstein

5. Some applications of OT in data analysis / machine learning

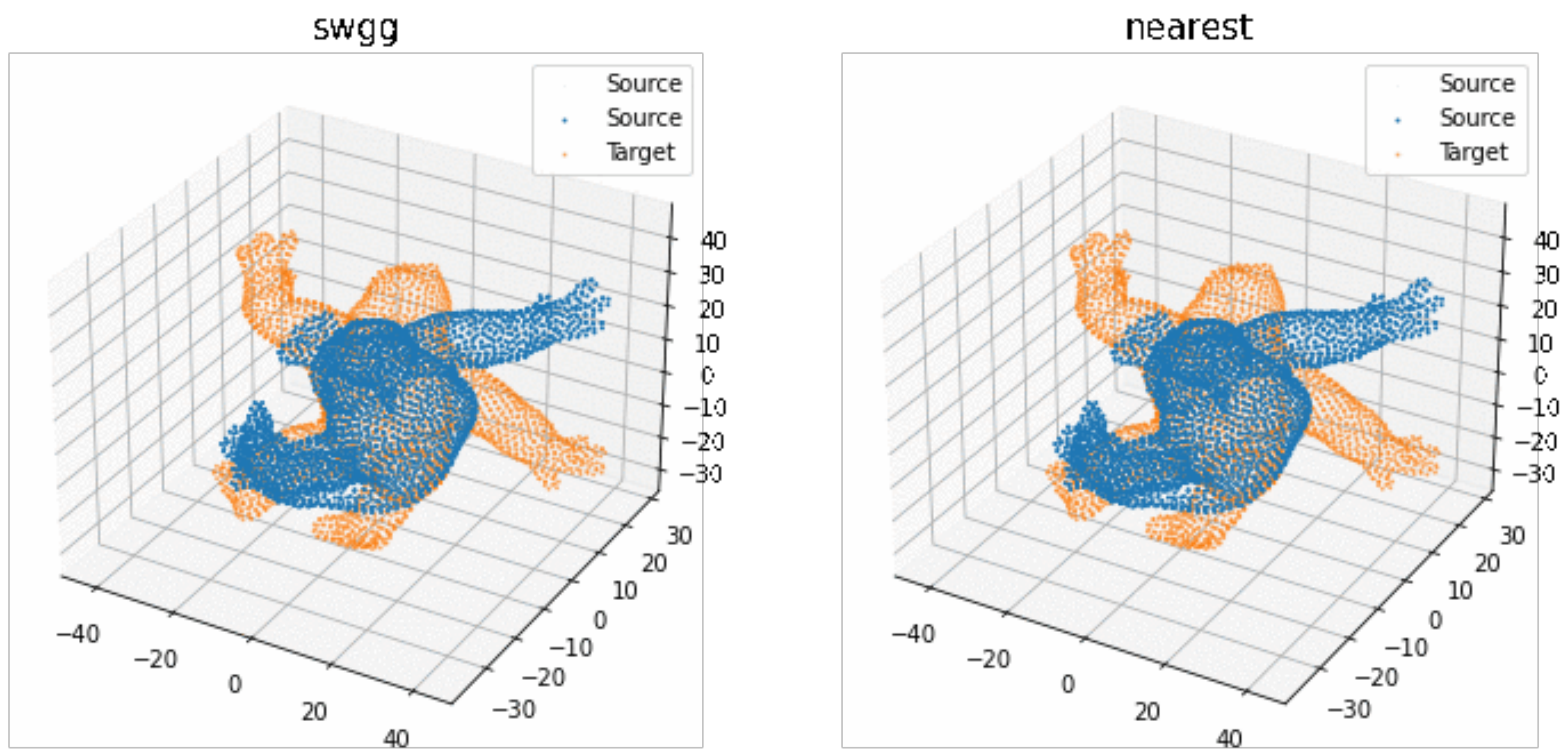
Some applications of OT

2 different aspects:

- transporting with OT (the plan is sought)
- using the divergence between (empirical) distributions

Some applications of OT

Transporting with OT



OT for shape registration [Bonneel 2019]

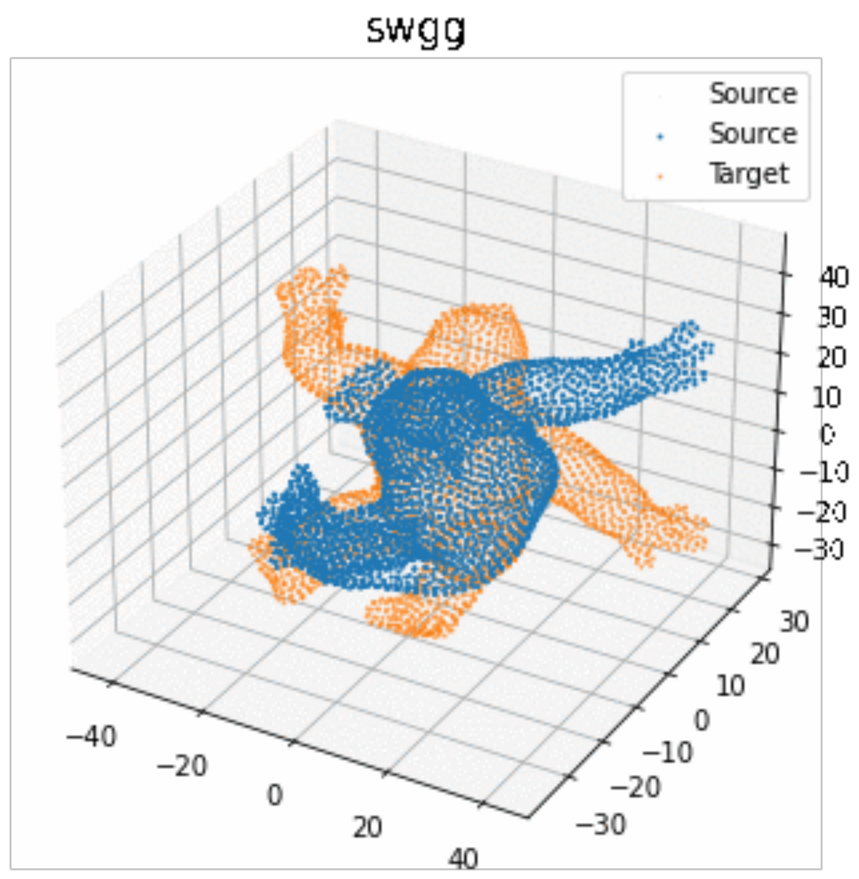
Iterative Closest Point (ICP) for aligning point clouds

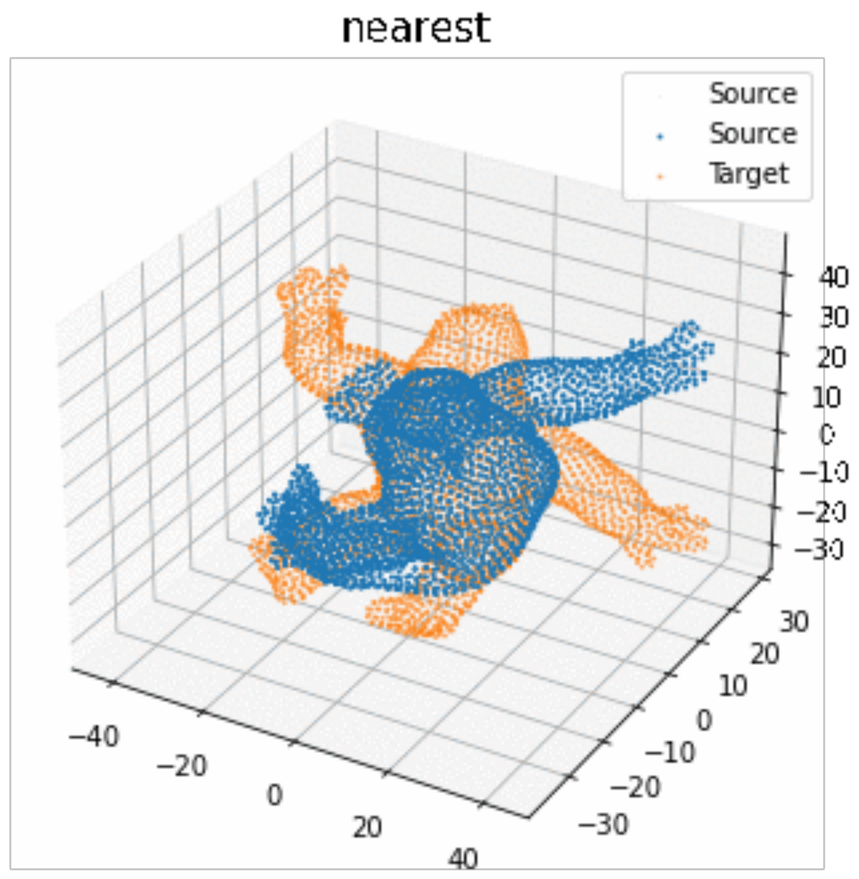
Defines a one-to-one correspondance, computes a rigid transformation (e.g rotation), moves the samples and iterates until convergence

$$\arg \min_{(\Omega,t)\in O(d)\times\mathbb{R}^d} \|\Omega(\boldsymbol{X}-t)-\boldsymbol{Y}\|_2^2$$

Some applications of OT

Transporting with OT





OT for shape registration [Bonneel 2019]

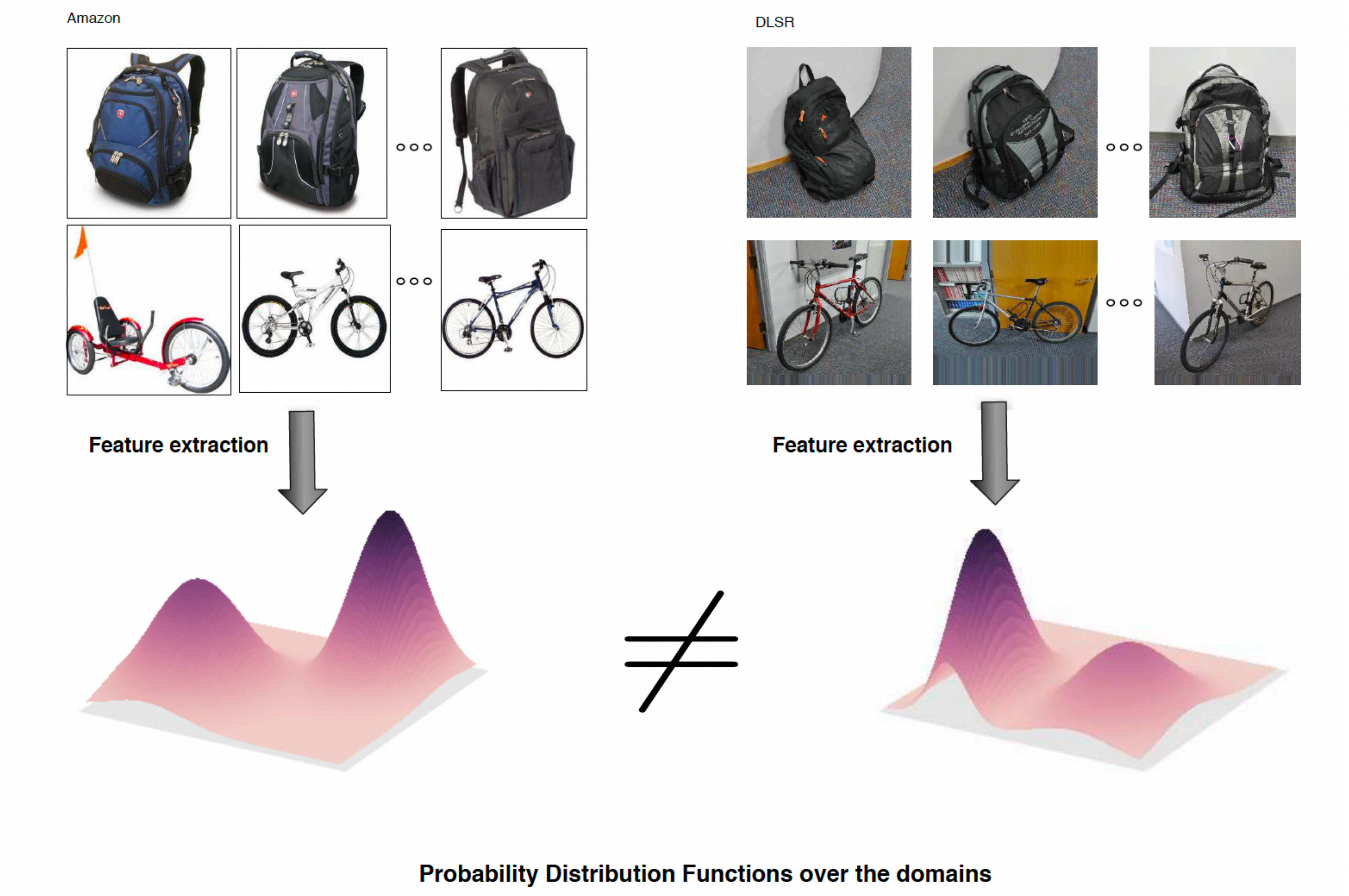
Iterative Closest Point (ICP) for aligning point clouds

Defines a one-to-one correspondance, computes a rigid transformation (e.g rotation), moves the samples and iterates until convergence

$$\arg \min_{(\Omega,t)\in O(d)\times\mathbb{R}^d} \|\Omega(\boldsymbol{X} - t) - \boldsymbol{Y}\|_2^2$$

Some applications of OT

Transporting with OT



Probability Distribution Functions over the domains

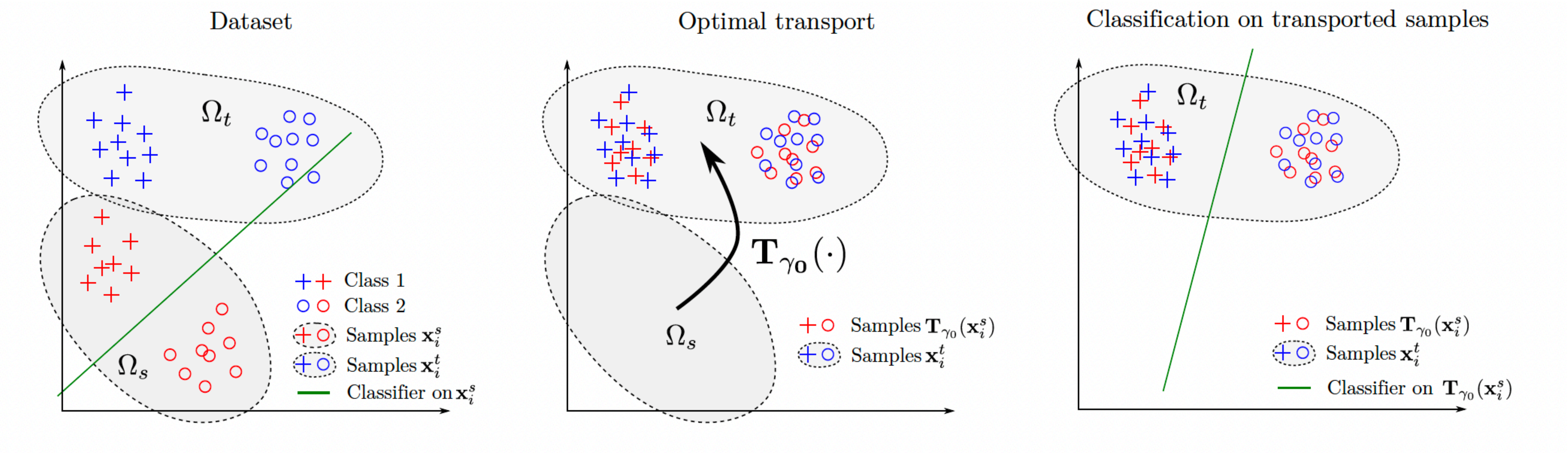
OT for domain adaptation [courty et al. 2016]

Two different (yet related domains)

Classification problem, labels available on the source domain but not on the target domain

Some applications of OT

Transporting with OT

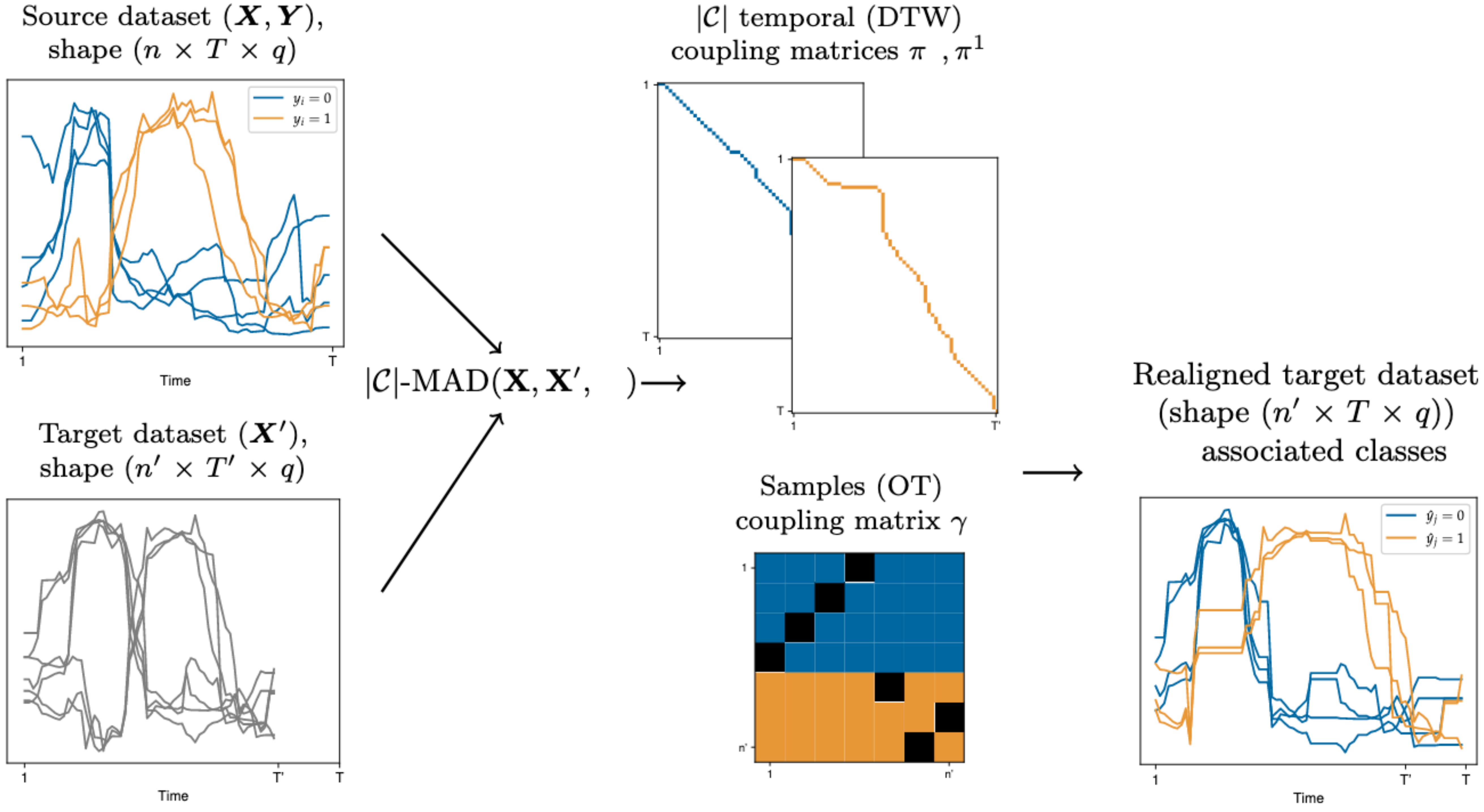

Dataset
Ω_t
Ω_s
Class 1
Class 2
Samples $\mathbf{x}_i^s$
Samples $\mathbf{x}_i^t$
Classifier on $\mathbf{x}_i^s$
Optimal transport
Ω_t
$\mathbf{T}_{\gamma_0}(\cdot)$
Ω_s
Samples $\mathbf{T}_{\gamma_0}(\mathbf{x}_i^s)$
Samples $\mathbf{x}_i^t$
Classification on transported samples
Ω_t
Samples $\mathbf{T}_{\gamma_0}(\mathbf{x}_i^s)$
Samples $\mathbf{x}_i^t$
Classifier on $\mathbf{T}_{\gamma_0}(\mathbf{x}_i^s)$

OT for domain adaptation [courty et al. 2016]

step 1: compute the OT coupling between the 2 domains
step 2: transport the source onto the target domain
step 3: **classify** the transported source samples based on the classification rule computed on the target domain

Some applications of OT

Transporting with OT



Source dataset $(\boldsymbol{X}, \boldsymbol{Y})$,
shape $(n \times T \times q)$
$y_i = 0$
$y_i = 1$
1
T
Time
Target dataset $(\boldsymbol{X}')$,
shape $(n' \times T' \times q)$
1
T'
T
Time
$|\mathcal{C}|$-MAD$(\mathbf{X}, \mathbf{X}', \quad) \longrightarrow$
$|\mathcal{C}|$ temporal (DTW)
coupling matrices $\pi \ \ , \pi^1$
1
T
1
1
T
1
T'
Samples (OT)
coupling matrix γ
1
n'
1
n'
$\longrightarrow$
Realigned target dataset
(shape $(n' \times T \times q)$)
associated classes
$\hat{y}_j = 0$
$\hat{y}_j = 1$
1
T'
T
Time

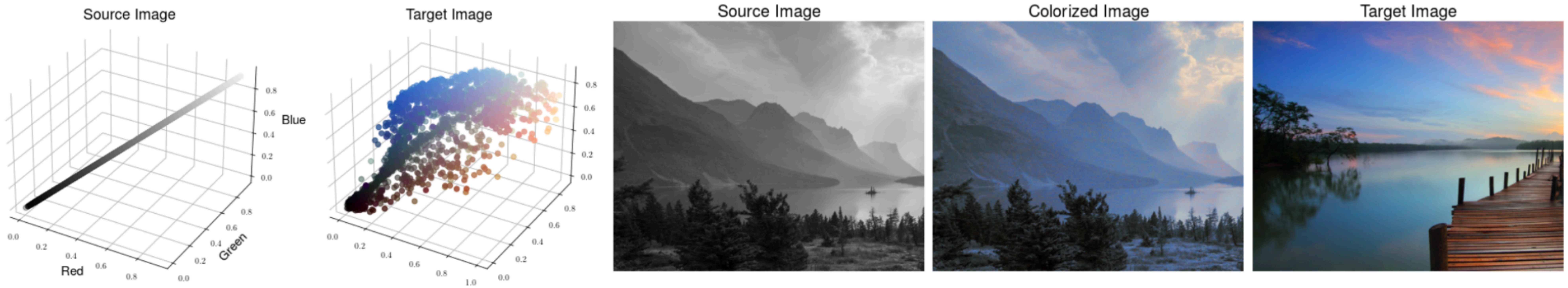
OT for domain adaptation for time series [Painblanc 2023]

versatility of OT thanks to the definition of the cost function: example with time series, where the cost is DTW

Some applications of OT

Transporting with OT

Color transfer

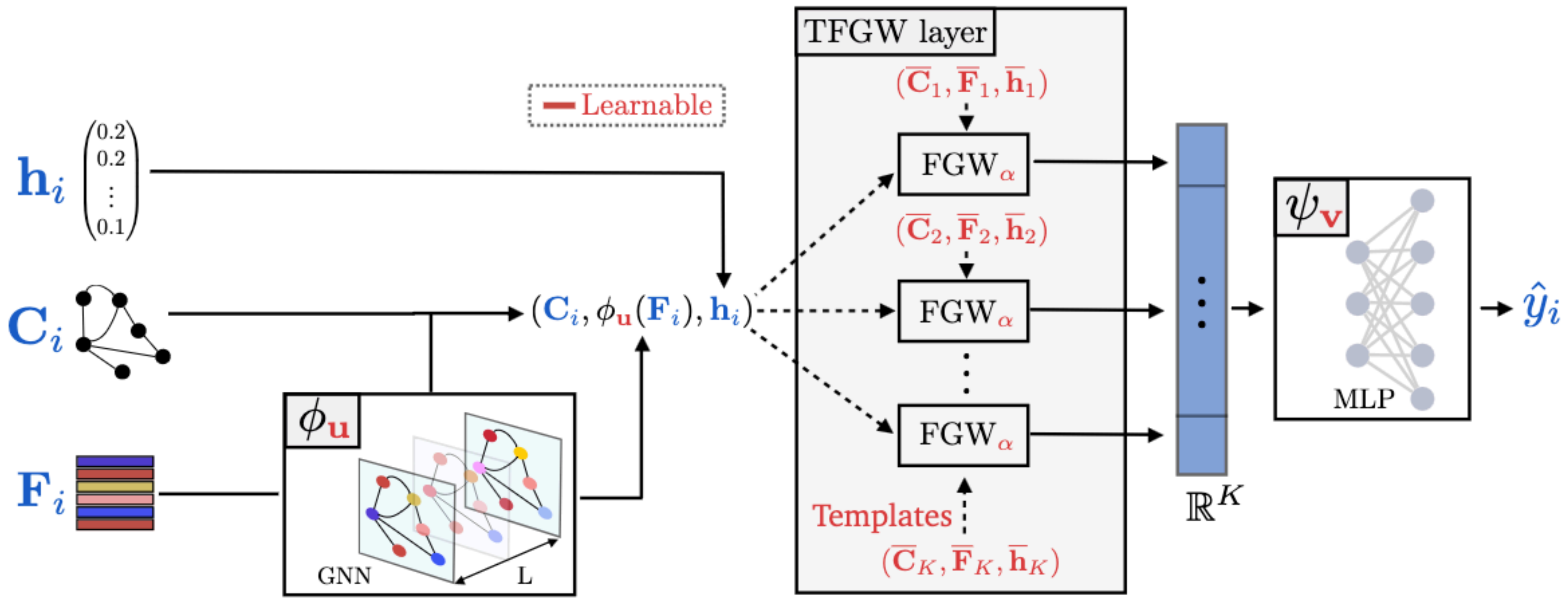


We aim to transport the color of one source image onto a target image
Input distributions: histograms of colors
When one distribution is supported on a line, there exists a closed form [Mahey2023]
The OT coupling is used to transfer the colors

Some applications of OT

Use of the divergence between empirical distributions

Template based Graph Neural Network with Optimal Transport Distances [Vincent-Cuaz 2022]

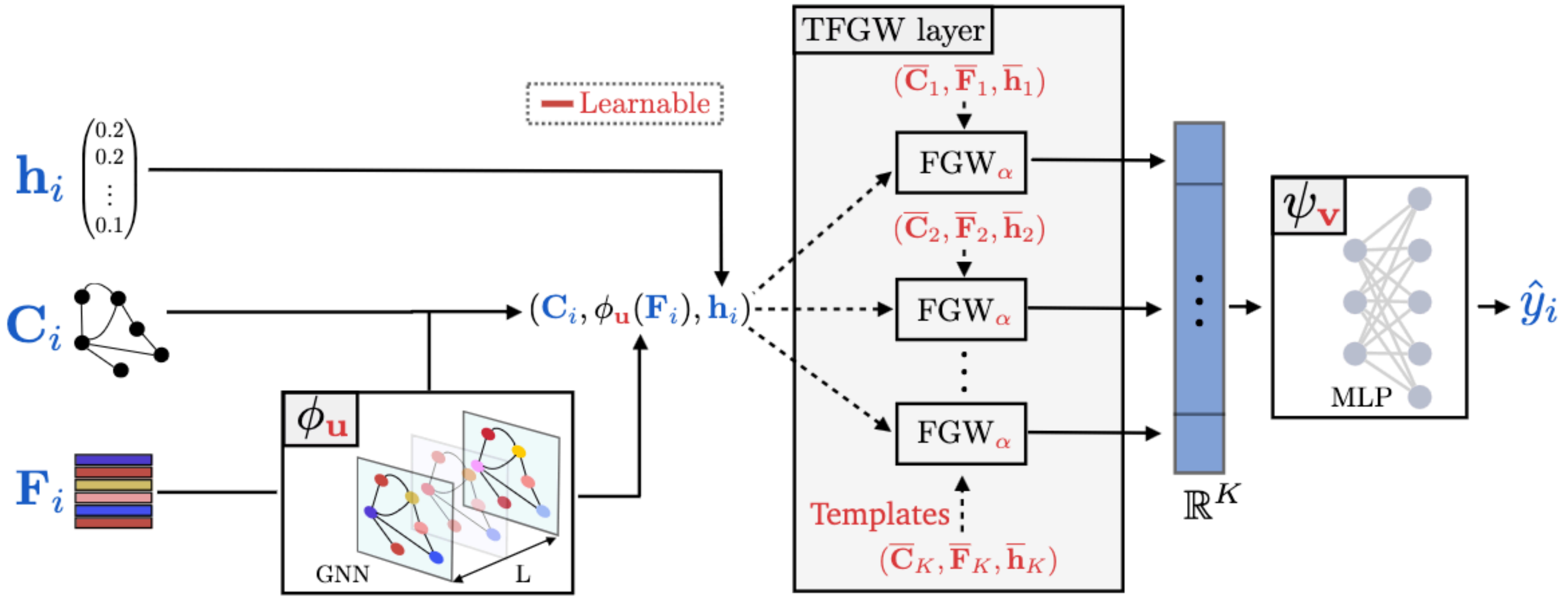

TFGW layer
Learnable
$\mathbf{h}_i$
$\begin{pmatrix}0.2\\0.2\\\vdots\\0.1\end{pmatrix}$
$\mathbf{C}_i$
$\mathbf{F}_i$
$\phi_{\mathbf{u}}$
GNN
L
$(\mathbf{C}_i, \phi_{\mathbf{u}}(\mathbf{F}_i), \mathbf{h}_i)$
$(\overline{\mathbf{C}}_1, \overline{\mathbf{F}}_1, \overline{\mathbf{h}}_1)$
FGW_α
$(\overline{\mathbf{C}}_2, \overline{\mathbf{F}}_2, \overline{\mathbf{h}}_2)$
FGW_α
FGW_α
Templates
$(\overline{\mathbf{C}}_K, \overline{\mathbf{F}}_K, \overline{\mathbf{h}}_K)$
$\mathbb{R}^K$
$\psi_{\mathbf{v}}$
MLP
$\hat{y}_i$

Compute the FGW distance of a graph to several graph *templates*
New feature representation of the graph: vector of distances
This vector is then feed into a MLP to predict the class of the graph

Some applications of OT

Use of the divergence between empirical distributions

Template based Graph Neural Network with Optimal Transport Distances [Vincent-Cuaz 2022]



$\mathbf{h}_i$
0.2
0.2
⋮
0.1
$\mathbf{C}_i$
$\mathbf{F}_i$
$\phi_{\mathbf{u}}$
GNN
L
Learnable
$(\mathbf{C}_i, \phi_{\mathbf{u}}(\mathbf{F}_i), \mathbf{h}_i)$
TFGW layer
$(\overline{\mathbf{C}}_1, \overline{\mathbf{F}}_1, \overline{\mathbf{h}}_1)$
FGW_α
$(\overline{\mathbf{C}}_2, \overline{\mathbf{F}}_2, \overline{\mathbf{h}}_2)$
FGW_α
FGW_α
Templates
$(\overline{\mathbf{C}}_K, \overline{\mathbf{F}}_K, \overline{\mathbf{h}}_K)$
$\mathbb{R}^K$
$\psi_{\mathbf{v}}$
MLP
$\hat{y}_i$

Compute the FGW distance of a graph to several graph *templates*
New feature representation of the graph: vector of distances
This vector is then feed into a MLP to predict the class of the graph

Gives better classification results than GNNs or kernel-based algorithms

Some applications of OT

Use of the divergence between empirical distributions

Wasserstein Generative Adversarial Networks [Arjovski 2017]

$$\min_G \max_D E_{\boldsymbol{x}\sim\boldsymbol{\mu_d}} \log D(x) + E_{\boldsymbol{z}\sim N(0,1)} \log(1 - D(G(z)))$$

Learn a Generator *G* that outputs realistic samples from data μ_x
Learn a Discriminator *D* able to discriminate generated and true samples

Some applications of OT

Use of the divergence between empirical distributions

Wasserstein Generative Adversarial Networks [Arjovski 2017]

$$\min_G \max_D E_{\boldsymbol{x}\sim\boldsymbol{\mu_d}} \log D(x) + E_{\boldsymbol{z}\sim N(0,1)} \log(1 - D(G(z)))$$

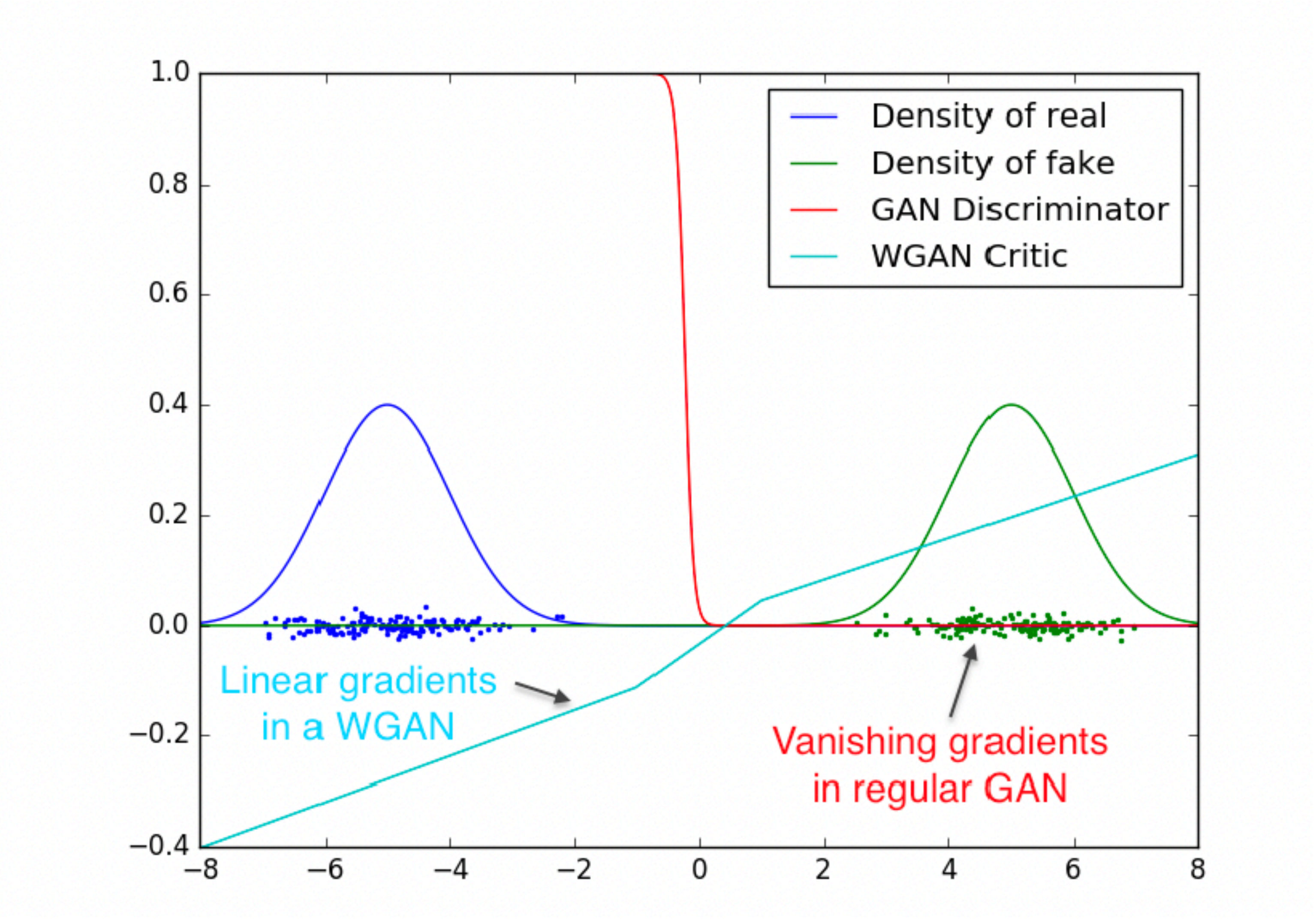
Learn a Generator *G* that outputs realistic samples from data μ_x
Learn a Discriminator *D* able to discriminate generated and true samples

Hard to train because of the vanishing gradients

Some applications of OT

Use of the divergence between empirical distributions

Wasserstein Generative Adversarial Networks



Wasserstein GAN minimizes the Wasserstein distance

$$\min_G W_1^1(G\#\mu_t, \mu_s)$$

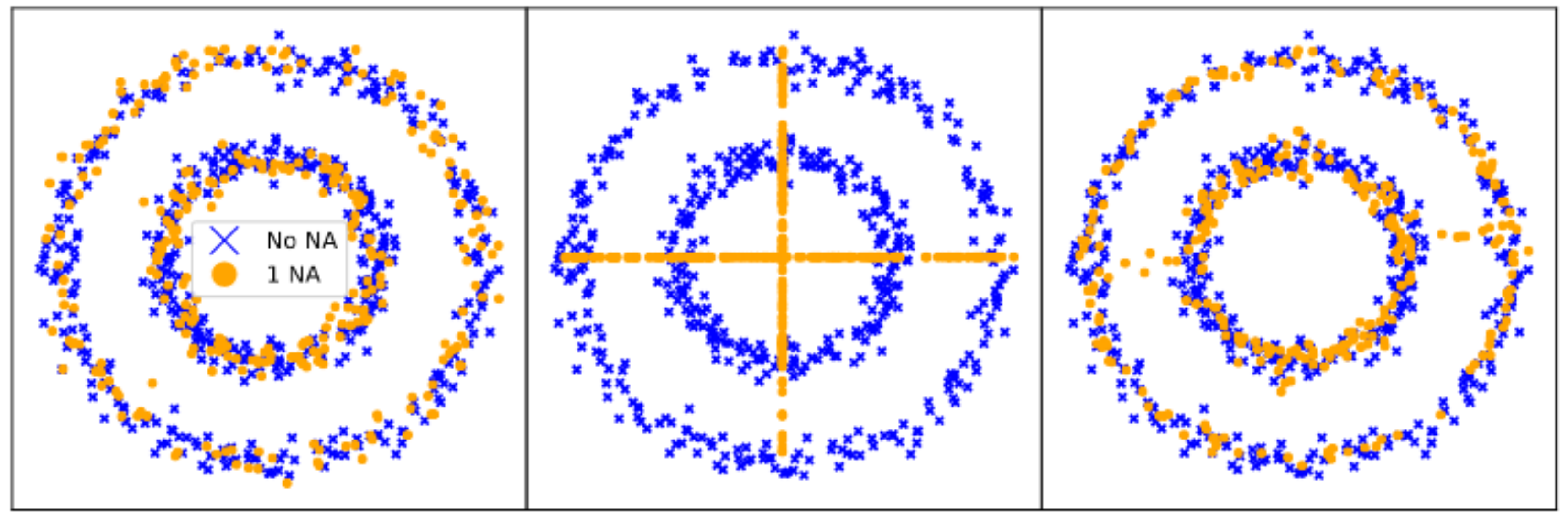
with the target distribution being a Gaussian $N(0,1)$

Gives better results in practice (and is easier to optimize)

Some applications of OT

Use of the divergence between empirical distributions

Missing data imputation [Muzellec 2020]



Data imputation: fills missing entries with plausible values

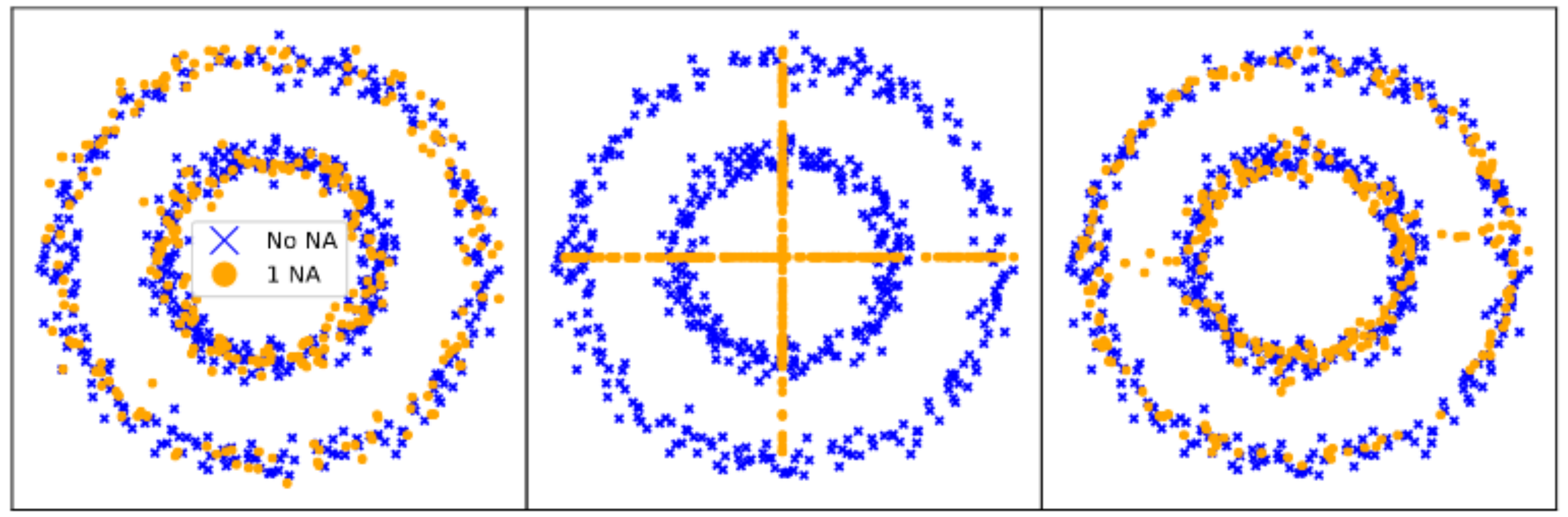
Assomption: two batches extracted randomly from the same dataset should share the same distribution
Suppose that values on some of the features are missing for one distribution

$$\min_{X^{imp}} \sum SD(\mu_m \boldsymbol{X}_K, \mu_m \boldsymbol{X}_L)$$

Some applications of OT

Use of the divergence between empirical distributions

Missing data imputation [Muzellec 2020]



Data imputation: fills missing entries with plausible values

Assomption: two batches extracted randomly from the same dataset should share the same distribution
Suppose that values on some of the features are missing for one distribution

$$\min_{X^{imp}} \sum SD(\mu_m(\boldsymbol{X}_K), \mu_m(\boldsymbol{X}_L))$$

complete

contains missing values

Summary

OT is a theoretically grounded way for comparing distributions
Different formulations: Monge (defines a map) or Kantorovitch (defines a plan)
Ground metric provides some geometry of the space (geodesics, barycenters)
Several variants: Unbalanced OT and Gromov-Wasserstein for unregistered distributions

OT is not robust to outliers: Unbalanced/partial OT relaxes the marginal constraints.
Solving OT is a linear program, GW is a quadratic problem
Reference for Computational OT [Peyre et Cuturi, 2019] or OT for applied mathematicians [Santambrogio 2015]
Regularizing the problem helps in reducing the complexity
There exist some tools for OT, for instance POT

References

[Kusner 2015] Kusner, M., Sun, Y., Kolkin, N., & Weinberger, K. From word embeddings to document distances. In IICML 2025

[Rout 2022] Rout, L., Korotin, A., & Burnaev, E. Generative Modeling with Optimal Transport Maps. In ICLR 2022.

[Mroueh 2020] Mroueh, Wasserstein Style Transfer, AISTATS, 2020.

[Frogner 2015] Frogner, C., Zhang, C., Mobahi, H., Araya, M., & Poggio, T. A. Learning with a Wasserstein loss. *NeurIPS,* 2015.

[Solomon 2015] Solomon, J., De Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A. & Guibas, L. Convolutional wasserstein distances. ACM Transactions on Graphics, 2015

[Arjovsky 2017] Arjovsky, M., Chintala, S., & Bottou, L. Wasserstein generative adversarial networks. ICML, 2017.

[Tolstikhin 2018] Tolstikhin, I., Bousquet, O., Gelly, S., & Schoelkopf, B. Wasserstein Auto-Encoders. ICLR, 2018.

[Monge 1781] Monge, G. Mémoire sur la théorie des déblais et des remblais. Mem. Math. Phys. Acad. Royale Sci., 666-704., 1781

[Peyré and Cuturi, 2019] Peyré G. & Cuturi M. (2019). Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*

[Rubner 2000] Rubner, Y., Tomasi, C., & Guibas, L. J. The earth mover's distance as a metric for image retrieval. International journal of computer vision

[Ambrosio 2005] Ambrosio, L., Gigli, N., & Savaré, G. Gradient flows: in metric spaces and in the space of probability measures. Springer, 2005

[Agueh 2011] Agueh, M., & Carlier, G. Barycenters in the Wasserstein space. SIAM Journal on Mathematical Analysis, 2011

[Rabin 2012] Rabin, J., Peyré, G., Delon, J., & Bernot, M. (2012). Wasserstein barycenter and its application to texture mixing. In SSVM 2011

[Mahey 2023] Mahey, G., Chapel, L, Gasso, G., Bonet, C., Courty N. Fast Optimal Transport through Sliced Wasserstein Generalized Geodesics. arXiv 2023.

References

[Cuturi 2013] Cuturi, M.. Sinkhorn distances: Lightspeed computation of optimal transport. Advances in neural information processing systems, 2013.
[Feydy 2019] Feydy, J., Séjourné, T., Vialard, F. X., Amari, S. I., Trouvé, A., & Peyré, G. Interpolating between optimal transport and mmd using sinkhorn divergences. In The AISTATS 2019
[Benamou 2003] Benamou J.D. Numerical resolution of an "unbalanced" mass transport problem. ESAIM: Mathematical Modelling and Numerical Analysis, 2003.
[Chapel 2020] Chapel, L., Alaya, M. Z., & Gasso, G. Partial optimal transport with applications on positive-unlabeled learning. *NeurIPS*, 2020.
[Chapel 2021] Chapel, L., Flamary, R., Wu, H., Févotte, C., & Gasso, G. Unbalanced optimal transport through non-negative penalized linear regression. *NeurIPS*, 2021.
[Memoli 2011] Mémoli. Gromov–Wasserstein distances and the metric approach to object matching. Foundations of computational mathematics, 2011.
[Vayer 2019] Vayer T., Chapel, L., Flamary, R., Tavenard, R. & Courty N.. Optimal transport for structured data with application on graphs. *ICML*, 2019.
[Peyré 2016] Peyré, G., Cuturi, M., & Solomon, J. Gromov-wasserstein averaging of kernel and distance matrices. *ICML*, 2016.
[Séjourné 2021] Séjourné, T., Vialard, F. X., & Peyré, G. The unbalanced gromov wasserstein distance: Conic formulation and relaxation. *NeurIPS*, 2021.
[Bonneel 2019] Bonneel, N., & Coeurjolly, D. Spot: sliced partial optimal transport. ACM Transactions on Graphics (TOG), 38(4), 1-13. 2019
[Courty 2016] COurty, N., Flamary, R., Tuia, D., & Rakotomamonjy, A. Optimal transport for domain adaptation. IEEE Trans. Pattern Anal. Mach. Intell, 2016.
[Painblanc 2023] Painblanc, F., Chapel, L., Courty, N., Friguet, C., Pelletier, C., & Tavenard, R. Match-And-Deform: Time Series Domain Adaptation through Optimal Transport and Temporal Alignment. In ECML, 2023
[Vincent-Cuaz 2022] Vincent-Cuaz, C., Flamary, R., Corneli, M., Vayer, T., & Courty, N. Template based graph neural network with optimal transport distances. NeurIPS 2022
[Muzellec 2020] Muzellec, B., Josse, J., Boyer, C., & Cuturi, M. Missing data imputation using optimal transport. In ICML 2020
[Santambrogio 2015] Santambrogio, F. Optimal transport for applied mathematicians. *Birkäuser,* 2015.