Tutorial:
Optimal Transport for Machine Learning

LOGML Summer School 2024

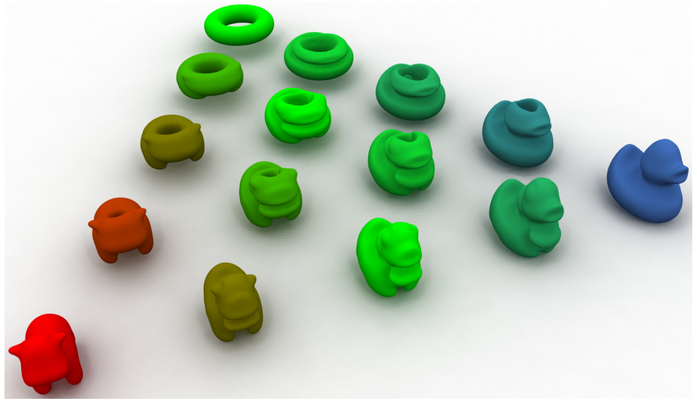
London, UK

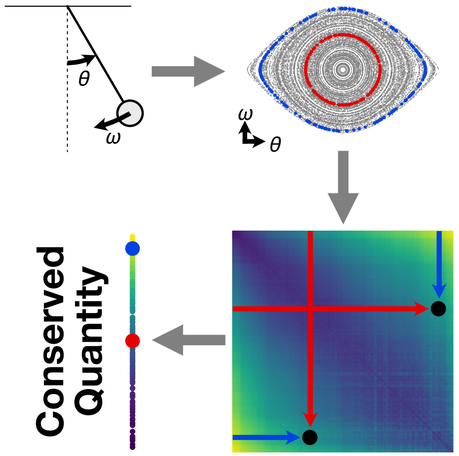
Peter Y. Lu

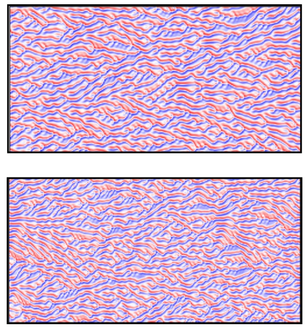
THE UNIVERSITY OF CHICAGO

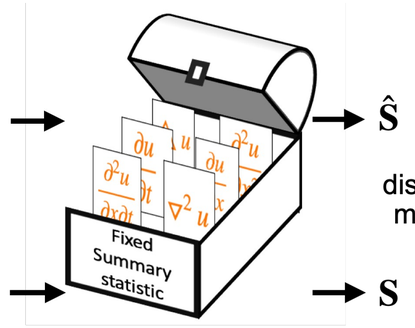
Schmidt Sciences

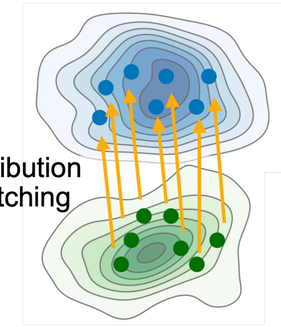
Overview: Optimal Transport for Machine Learning

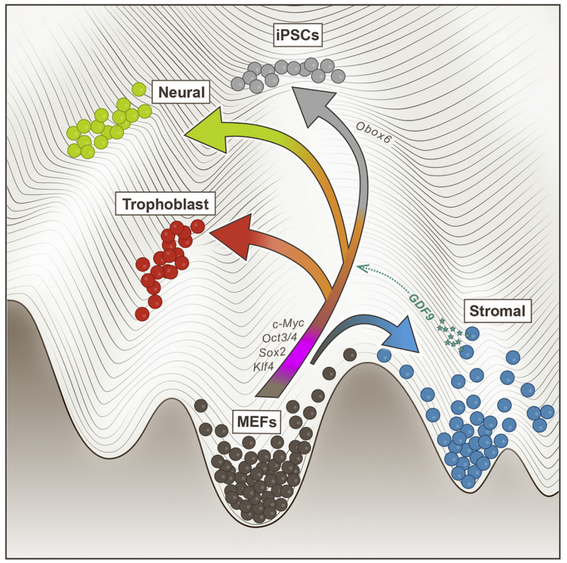


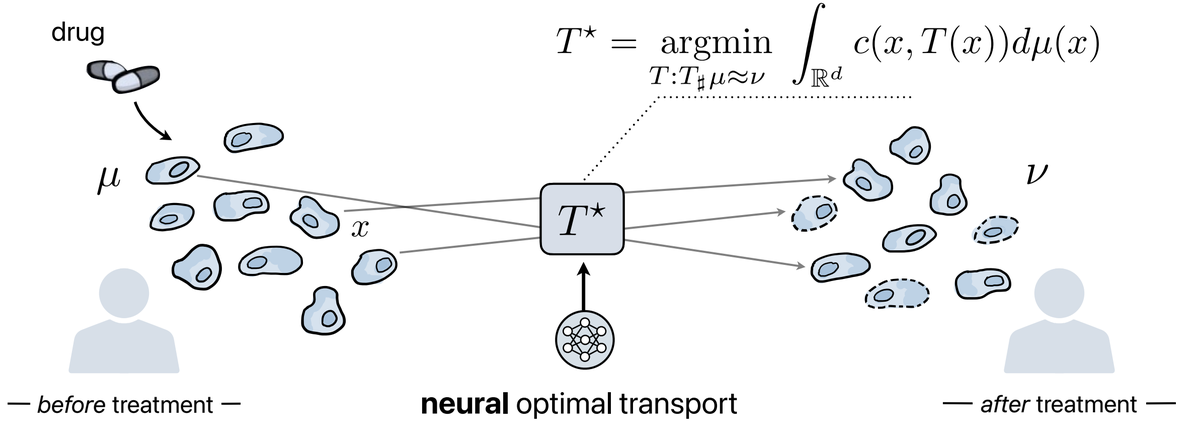








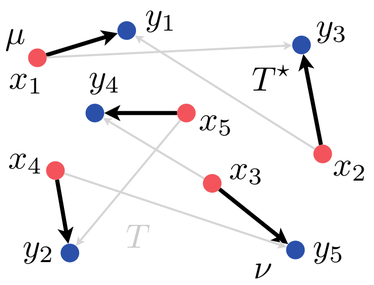


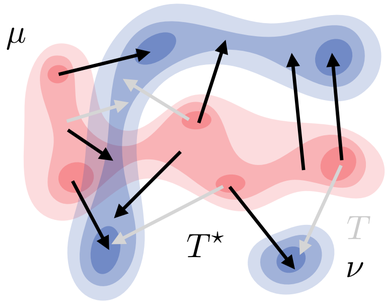


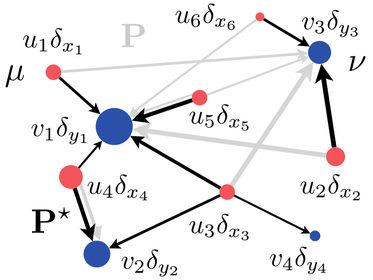
Overview: Optimal Transport for Machine Learning

Optimal Transport Theory

- Monge & Kantorovich Formulations
- Wasserstein Distances



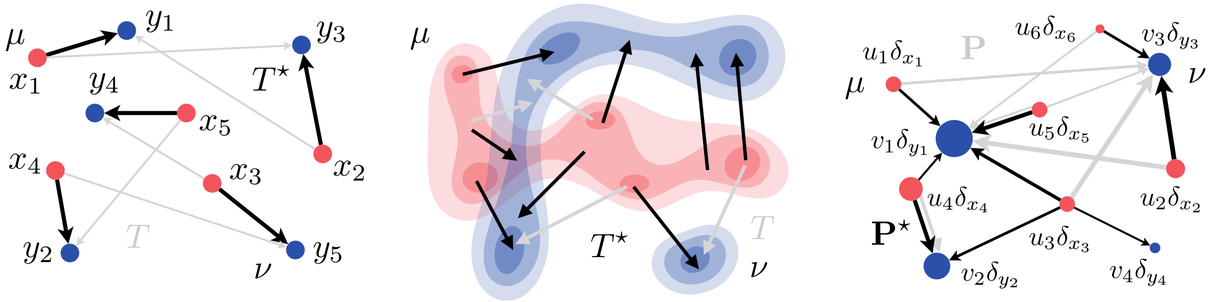




Overview: Optimal Transport for Machine Learning

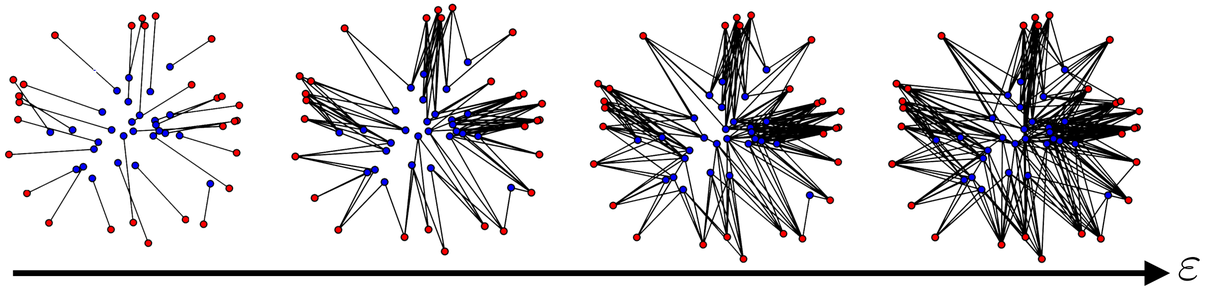
Optimal Transport Theory

- Monge & Kantorovich Formulations
- Wasserstein Distances



Computational Optimal Transport

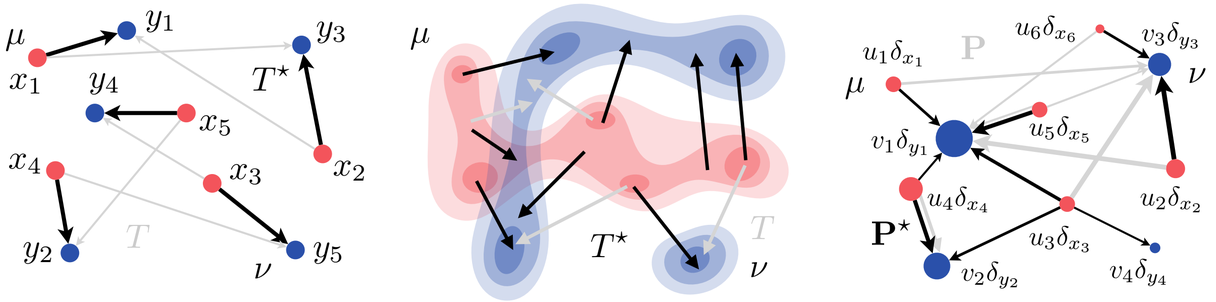
- 1D Optimal Transport
- Entropy-Regularized Optimal Transport



Overview: Optimal Transport for Machine Learning

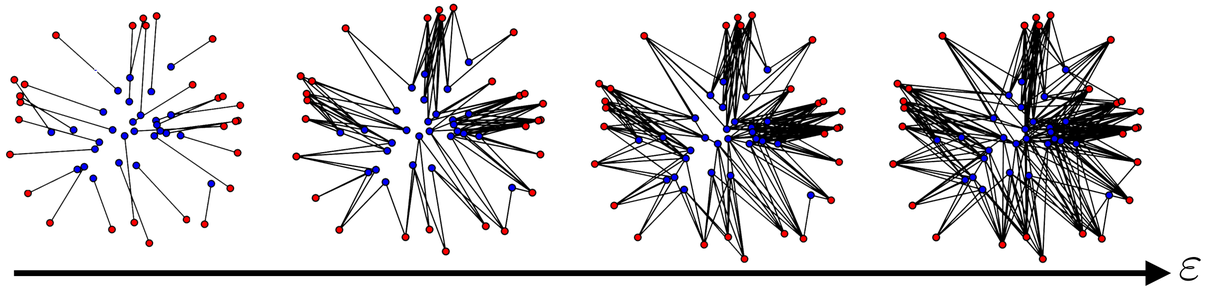
Optimal Transport Theory

- Monge & Kantorovich Formulations
- Wasserstein Distances



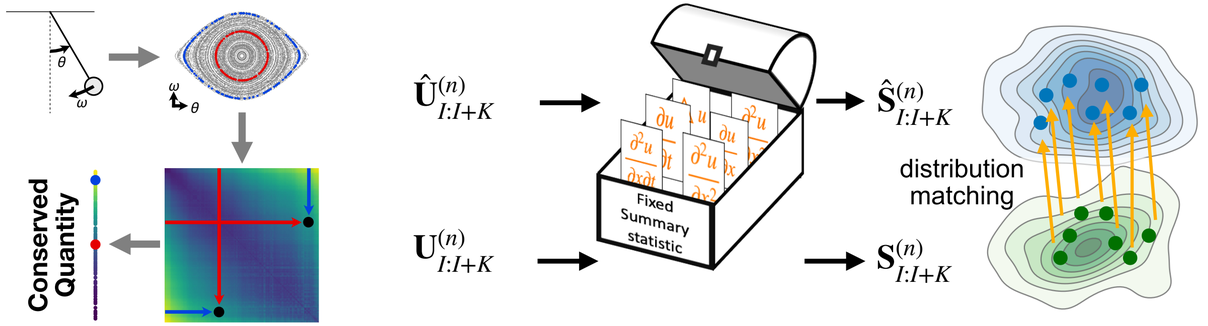
Computational Optimal Transport

- 1D Optimal Transport
- Entropy-Regularized Optimal Transport



Machine Learning Applications

- Shape Analysis
- Deep Learning



Overview: Optimal Transport for Machine Learning

Optimal Transport Theory

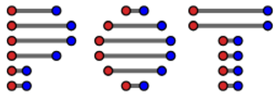
- Monge & Kantorovich Formulations
- Wasserstein Distances

Computational Optimal Transport

- 1D Optimal Transport
- Entropy-Regularized Optimal Transport

Machine Learning Applications

- Shape Analysis
- Deep Learning



POT: Python Optimal Transport (*NumPy*)

pythonot.github.io

OTT: Optimal Transport Tools (*JAX*)

ott-jax.readthedocs.io

GeomLoss: Geometric Loss (*PyTorch*)

kernel-operations.io/geomloss

Gabriel Peyré & Marco Cuturi

Computational Optimal Transport

optimaltransport.github.io

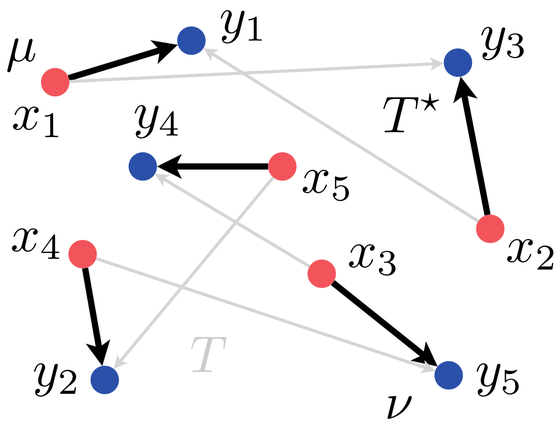
Charlotte Bunne & Marco Cuturi

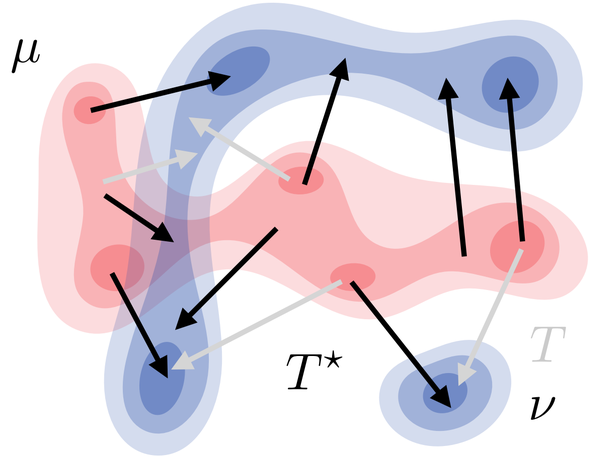
Optimal Transport in

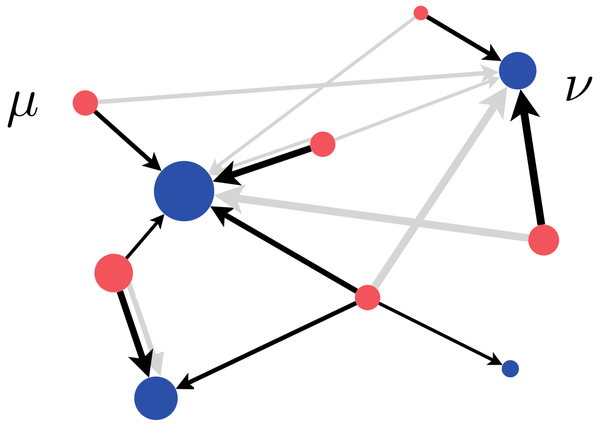
Learning, Control, and Dynamical Systems

bunne.ch/ot_tutorial

Optimal Transport Theory



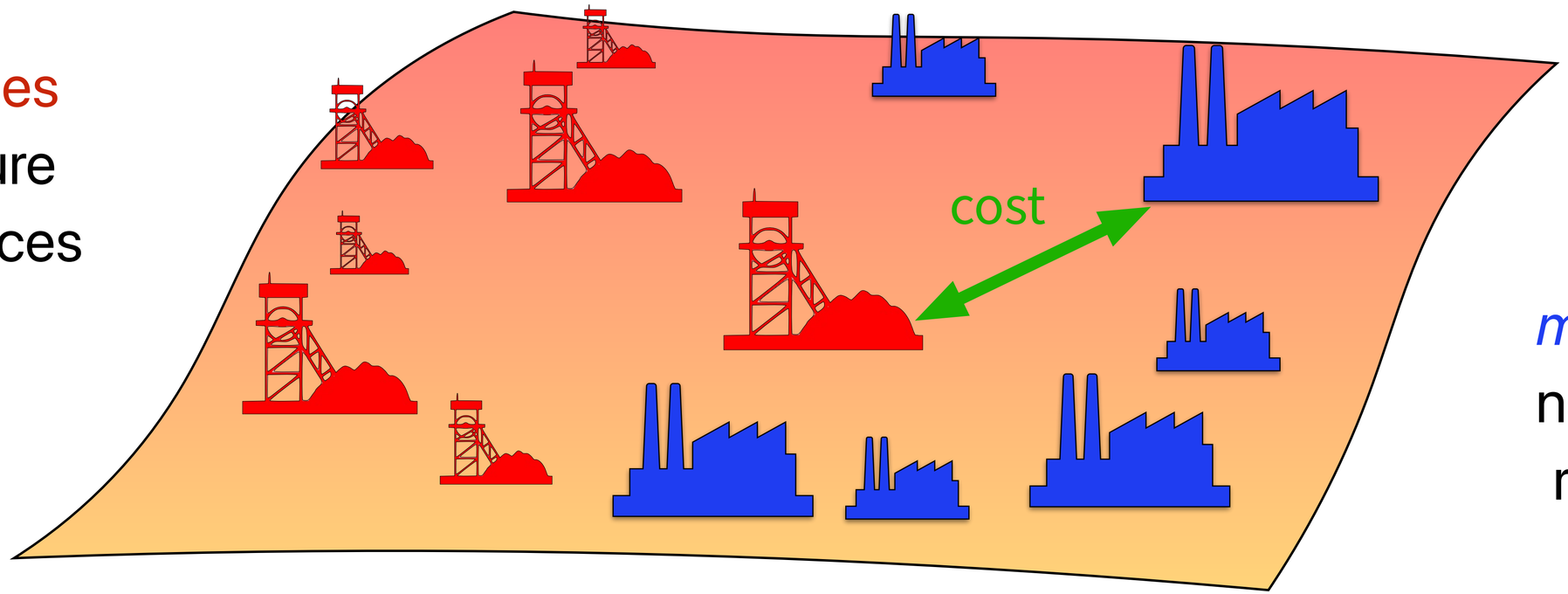




Motivation: Transporting Resources

Goal: Optimize ***cost*** of moving resources from ***sources*** to ***targets***.

n mines capture resources



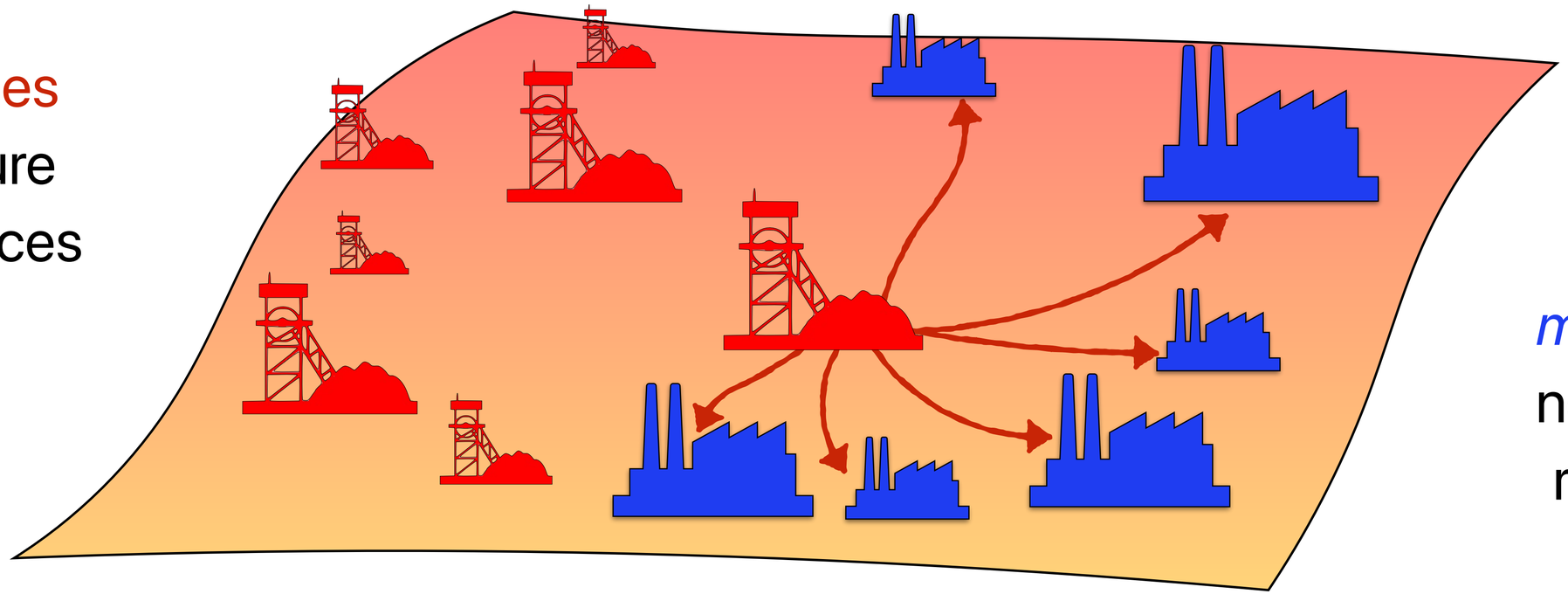
m factories need those resources

Bunne & Cuturi, ICML 2023

Motivation: Transporting Resources

Each of the *n* mines must dispatch its production

n mines capture resources

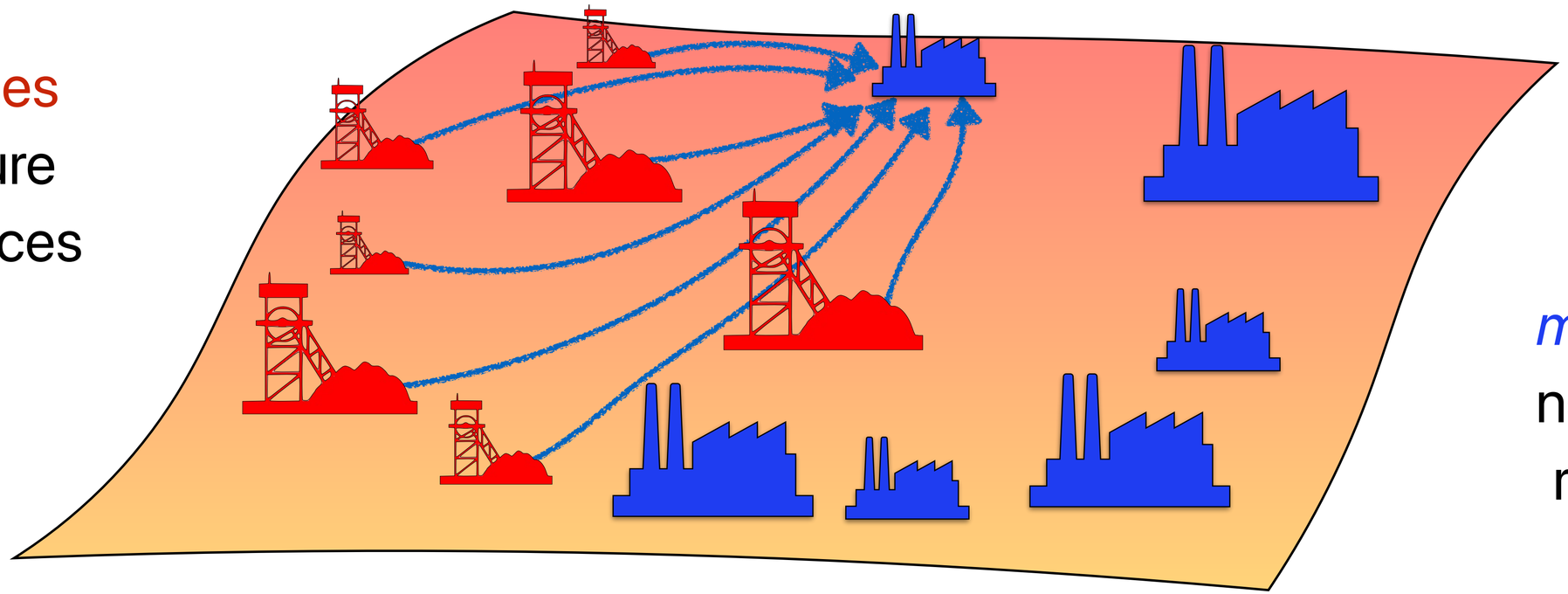


m factories need those resources

Bunne & Cuturi, ICML 2023

Motivation: Transporting Resources

n mines capture resources



m factories need those resources

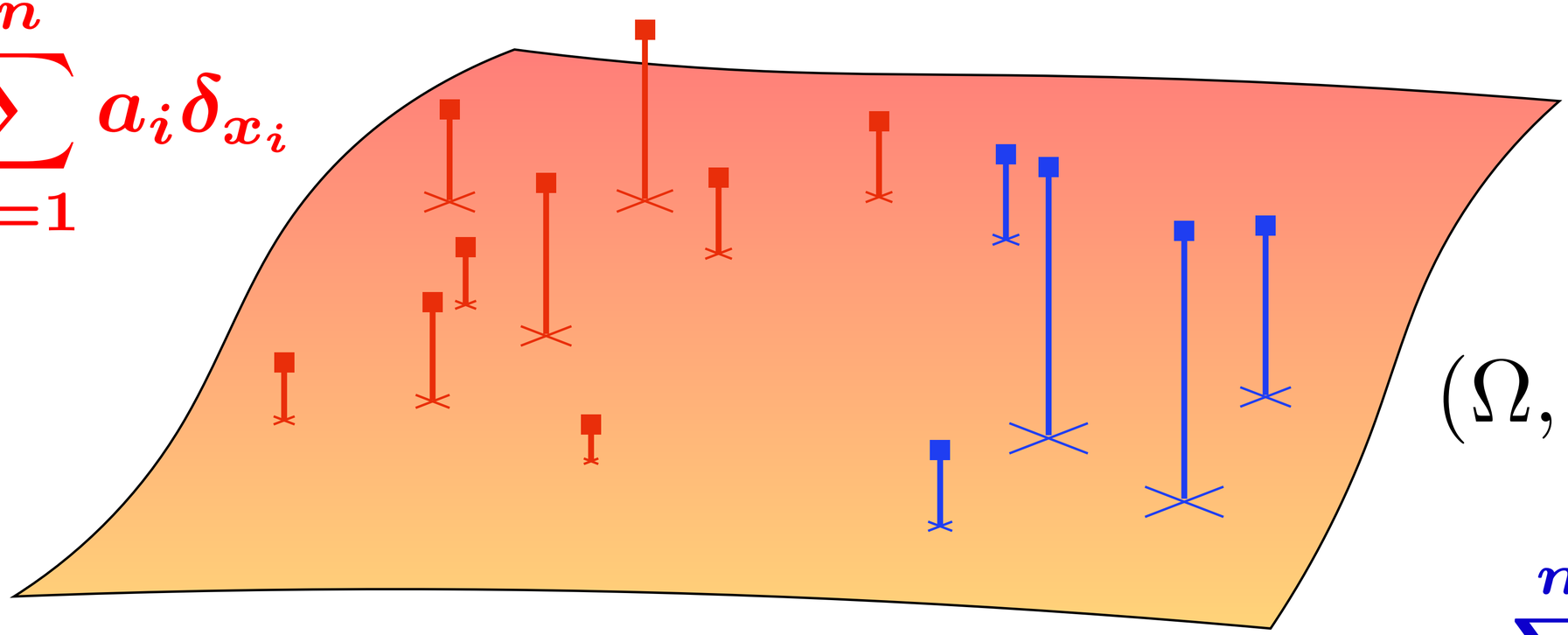
Each of the *m* factories must get resource it needs

Bunne & Cuturi, ICML 2023

Motivation: Transporting Resources

Goal: Optimize ***cost*** of moving resources from ***sources*** to ***targets***.

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



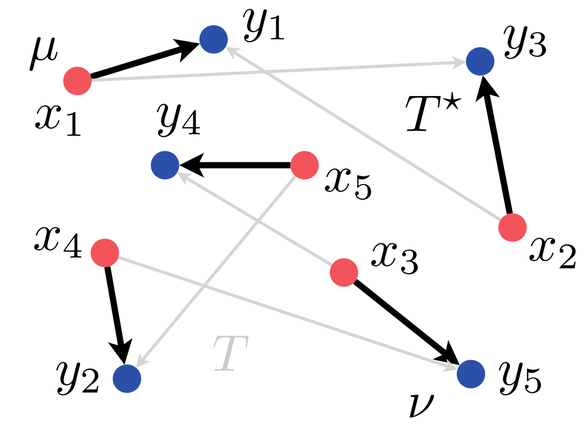
$$(\Omega, c)$$

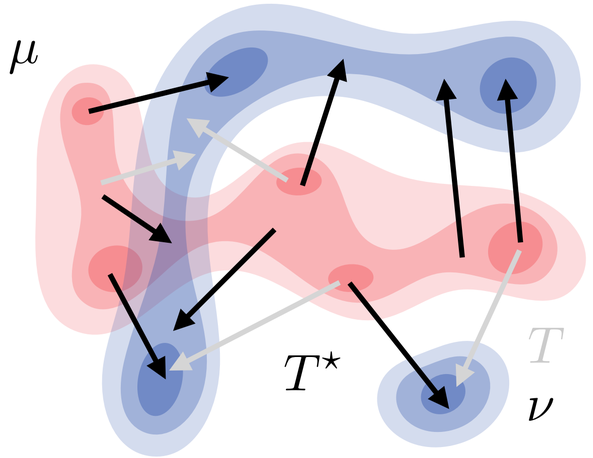
$$\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$$

Bunne & Cuturi, ICML 2023

Monge Formulation: Definitions

Probability measures μ, ν with support on spaces X, Y define *source* and *target* distributions.



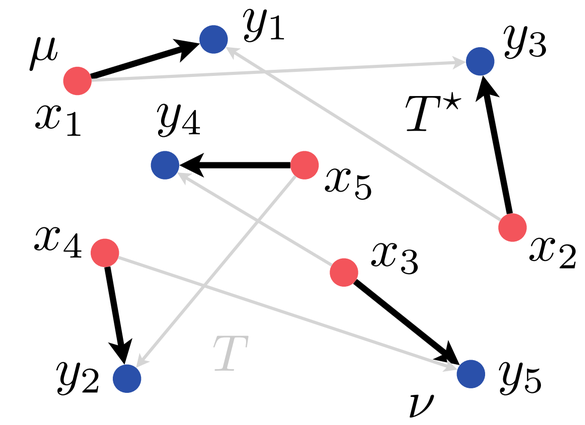


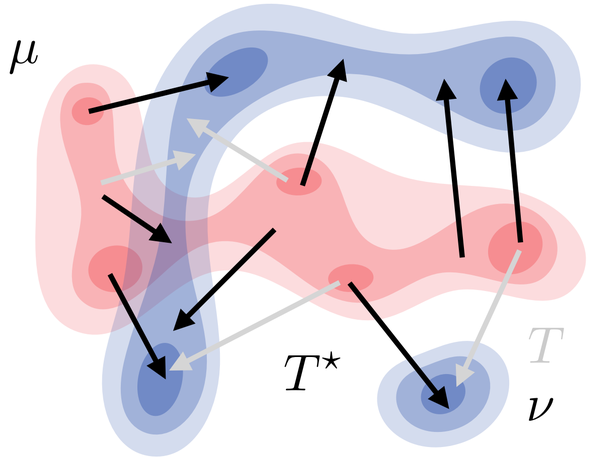
Bunne & Cuturi, ICML 2023

Monge Formulation: Definitions

Probability measures μ, ν with support on spaces X, Y define *source* and *target* distributions.

Cost function $c : X \times Y \to \mathbb{R}$ defines cost $c(x, y)$ of transporting one unit of resource from $x \in X$ to $y \in Y$.





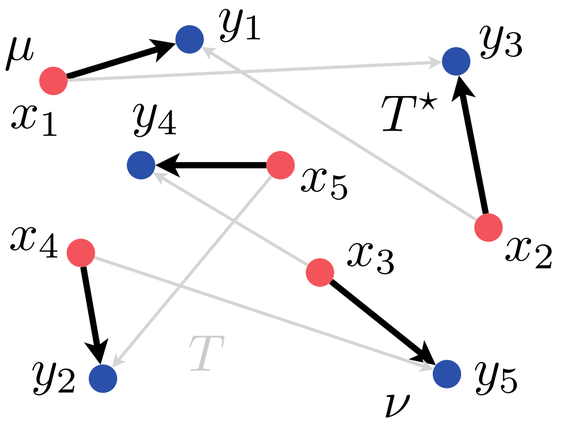
Bunne & Cuturi, ICML 2023

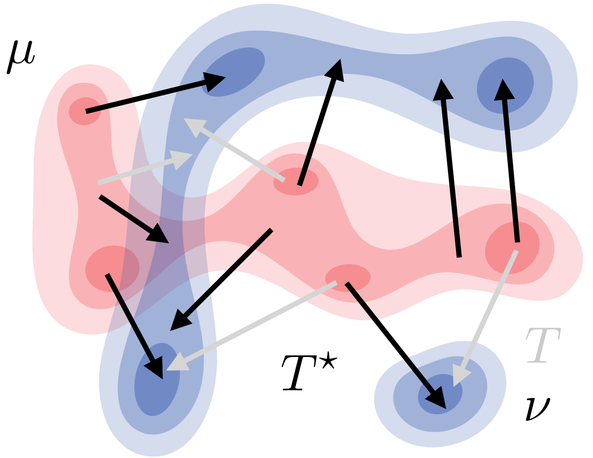
Monge Formulation: Definitions

Probability measures μ, ν with support on spaces X, Y define *source* and *target* distributions.

Cost function $c : X \times Y \to \mathbb{R}$ defines cost $c(x, y)$ of transporting one unit of resource from $x \in X$ to $y \in Y$.

Most common cost function $c(x, y) = d(x, y)^p$ is written in terms of distance metric d, known as the *ground metric*.

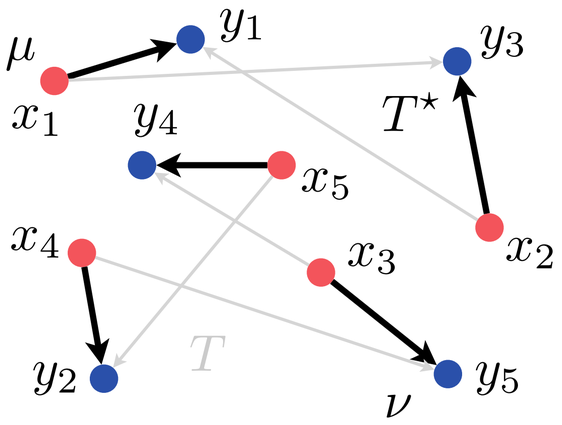




Bunne & Cuturi, ICML 2023

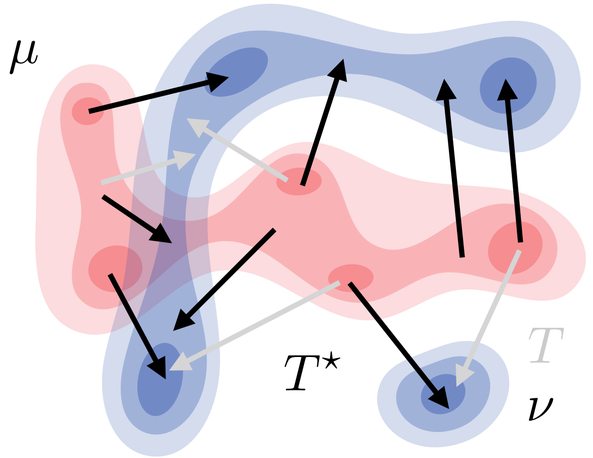
Monge Formulation: Definitions

Probability measures μ, ν with support on spaces X, Y define *source* and *target* distributions.



Cost function $c : X \times Y \to \mathbb{R}$ defines cost $c(x, y)$ of transporting one unit of resource from $x \in X$ to $y \in Y$.

Most common cost function $c(x, y) = d(x, y)^p$ is written in terms of distance metric d, known as the *ground metric*.



Transport map $T : X \to Y$ defines plan for transporting resources such that all resources are moved from the *source* to the *target* distribution, that is $T_\sharp \mu = \nu$.

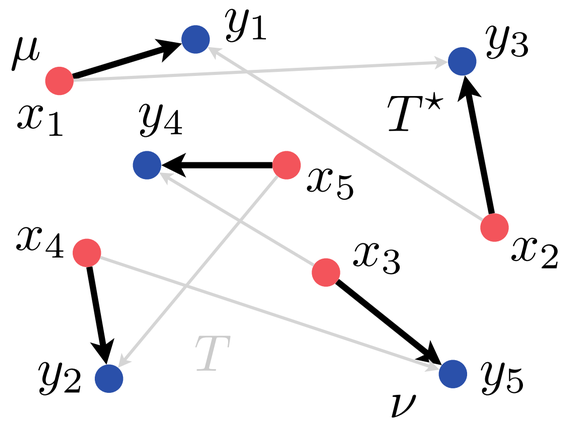
Bunne & Cuturi, ICML 2023

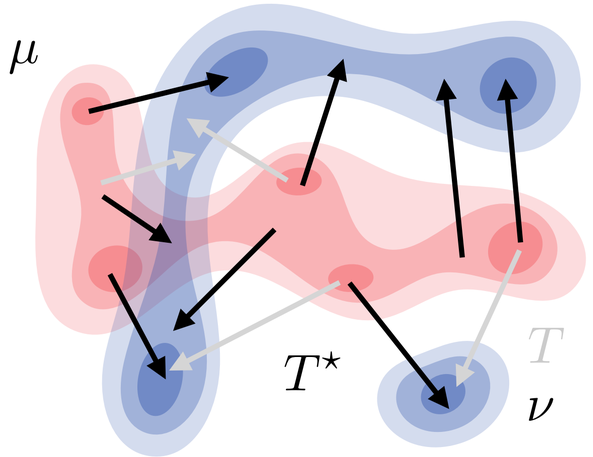
Monge Formulation: Optimal Transport Problem

Optimal transport map $T^\star$ has *minimal* total cost

$$\int_X c(x, T(x))\, \mathrm{d}\mu(x),$$

subject to constraint $T_\sharp \mu = \nu$.





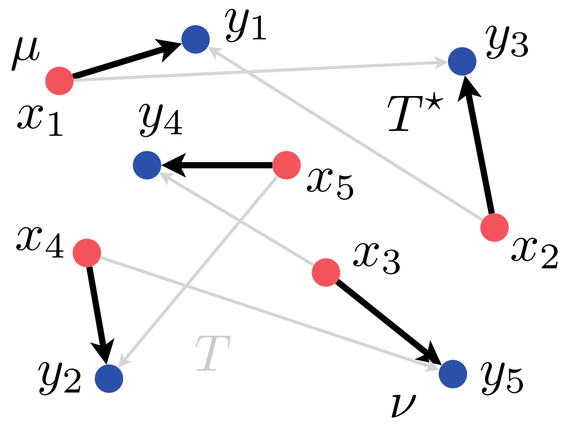
Bunne & Cuturi, ICML 2023

Monge Formulation: Optimal Transport Problem

Optimal transport map $T^\star$ has *minimal* total cost

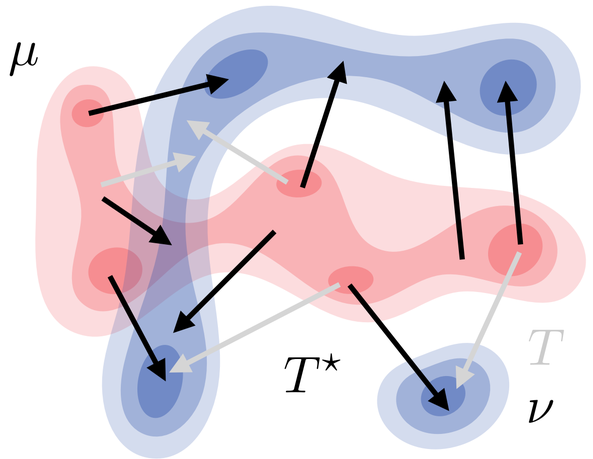
$$\int_X c(x, T(x))\, \mathrm{d}\mu(x),$$

subject to constraint $T_\sharp \mu = \nu$.



Optimal transport cost $L_c(\mu, \nu) = \int_X c(x, T^\star(x))\, \mathrm{d}\mu(x)$

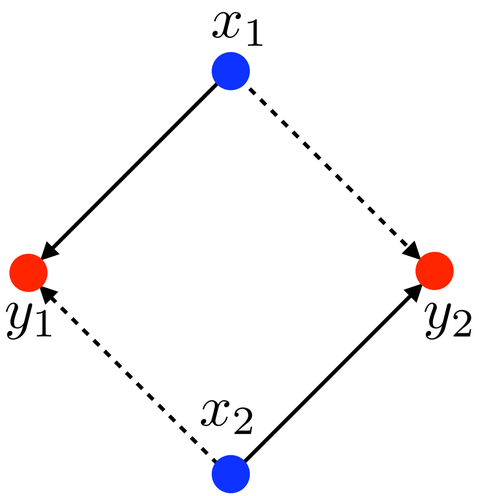
$$= \min_T \left\{ \int_X c(x, T(x))\, \mathrm{d}\mu(x) \,:\, T_\sharp \mu = \nu \right\}.$$



Bunne & Cuturi, ICML 2023

Monge Formulation: Complications

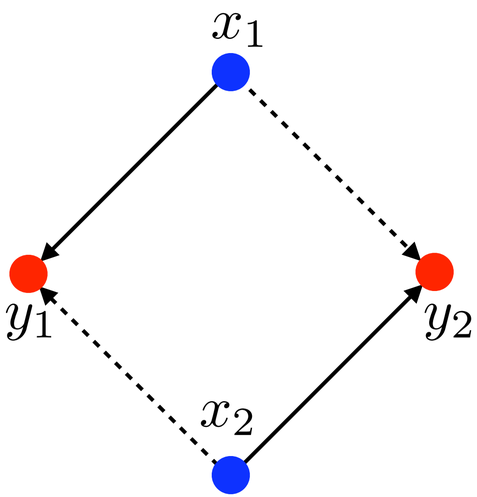
Non-uniqueness



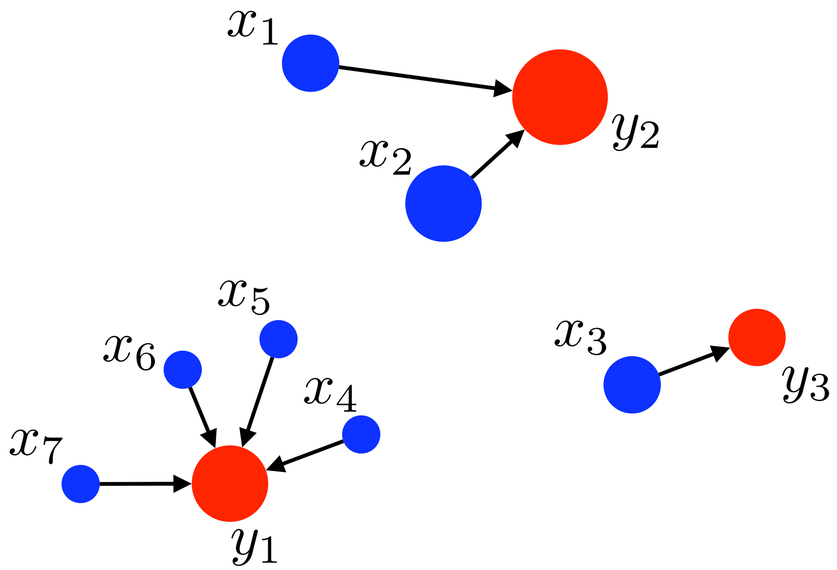
Peyré & Cuturi, 2019

Monge Formulation: Complications

Non-uniqueness



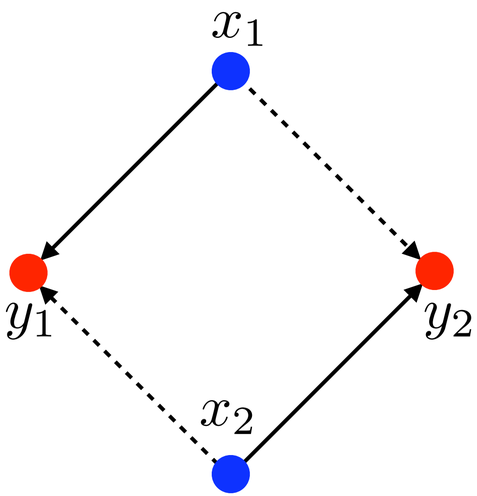
Ill-posed and asymmetric



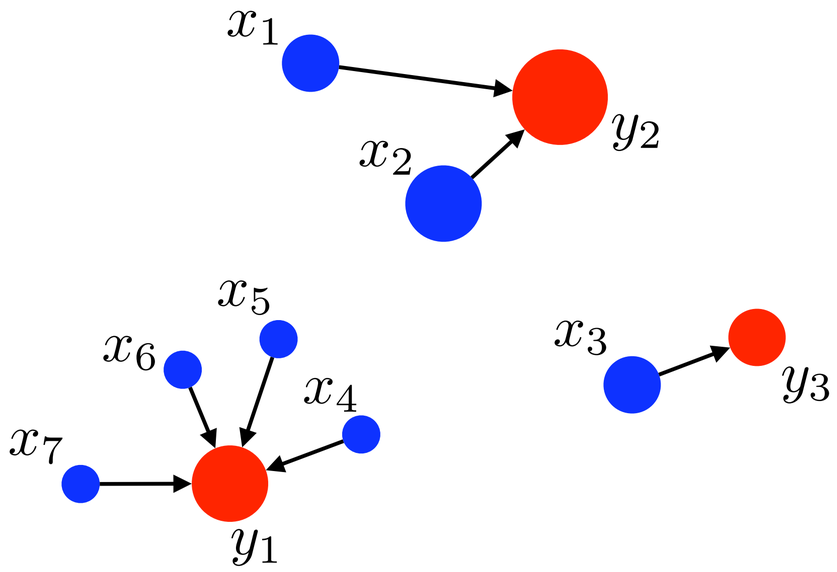
Peyré & Cuturi, 2019

Monge Formulation: Complications

Non-uniqueness



Ill-posed and asymmetric



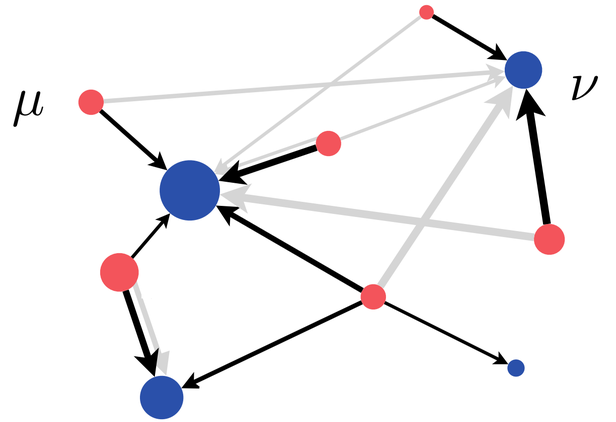
Highly nonlinear constraint $T_\sharp \mu = \nu$

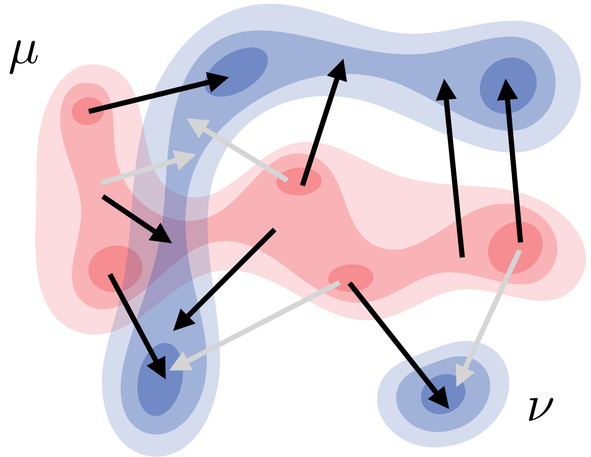
Peyré & Cuturi, 2019

Kantorovich Formulation: Convex Relaxation

Transport plan $\pi \in \Pi(\mu, \nu)$ defines probability measure on $X \times Y$ such that marginals of π match μ and ν, that is

$$\Pi(\mu, \nu) = \left\{ \pi : \int_X \mathrm{d}\pi(x, y) = \mathrm{d}\nu(y), \int_Y \mathrm{d}\pi(x, y) = \mathrm{d}\mu(x) \right\}.$$



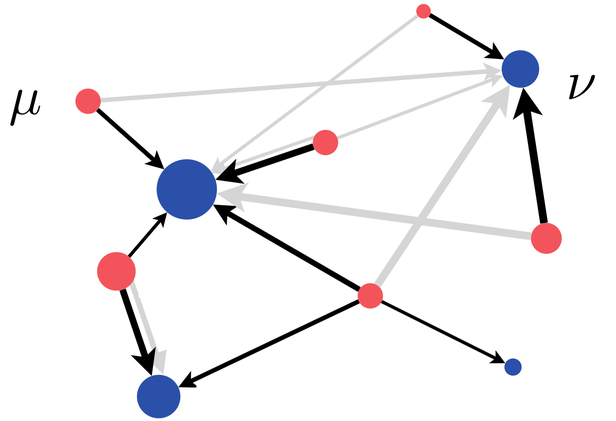


Bunne & Cuturi, ICML 2023

Kantorovich Formulation: Convex Relaxation

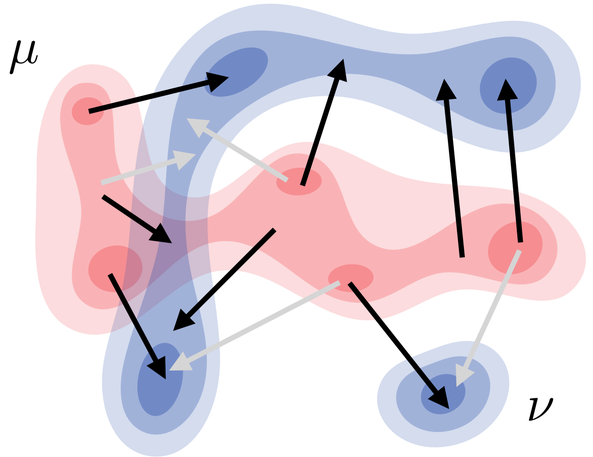
Transport plan $\pi \in \Pi(\mu, \nu)$ defines probability measure on $X \times Y$ such that marginals of π match μ and ν, that is

$$\Pi(\mu, \nu) = \left\{ \pi : \int_X \mathrm{d}\pi(x, y) = \mathrm{d}\nu(y), \int_Y \mathrm{d}\pi(x, y) = \mathrm{d}\mu(x) \right\}.$$



Optimal transport cost is given by

$$L_c(\mu, \nu) = \min_{\pi} \left\{ \int_{X \times Y} c(x, y)\, \mathrm{d}\pi(x, y) : \pi \in \Pi(\mu, \nu) \right\}.$$

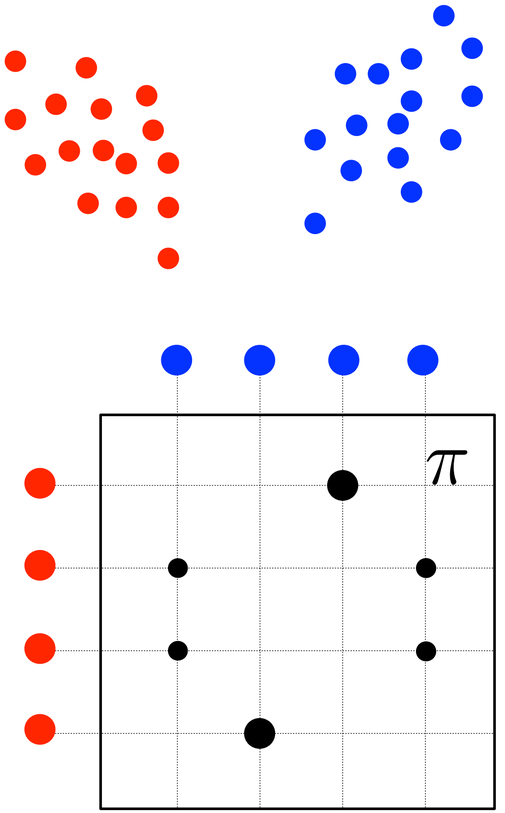


Bunne & Cuturi, ICML 2023

Kantorovich Formulation: Discrete Case

Transport plan matrix $P \in U(a, b)$, where

$$U(a, b) = \left\{ P : \sum_i P_{ij} = b_j, \sum_j P_{ij} = a_i, P_{ij} \geq 0 \right\}.$$

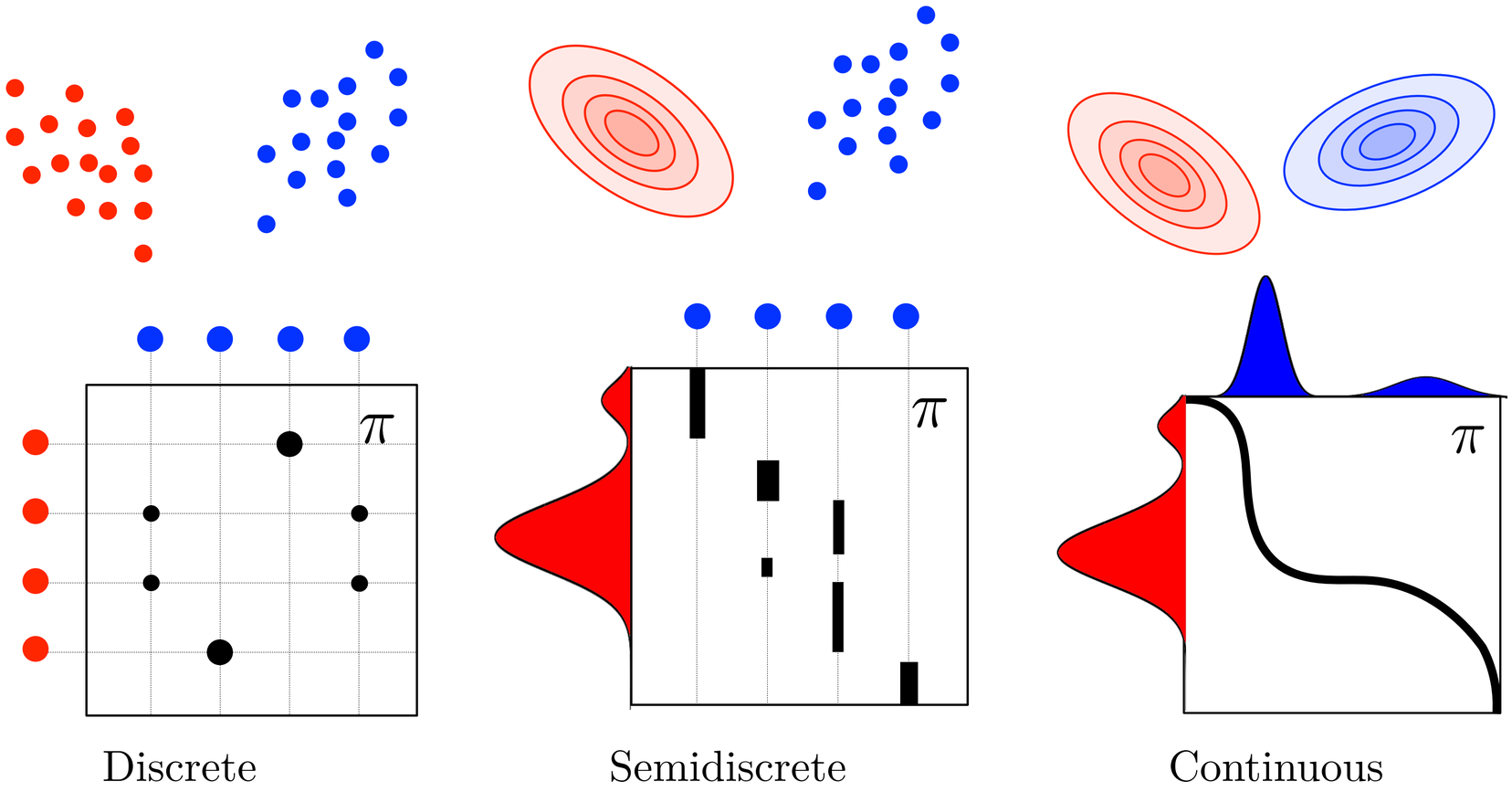


Optimal transport cost is given by

$$L_C(a, b) = \min_P \left\{ \sum_{i,j} C_{ij} P_{ij} : P \in U(a, b) \right\}.$$

Bunne & Cuturi, ICML 2023

Kantorovich Formulation: Convex Relaxation

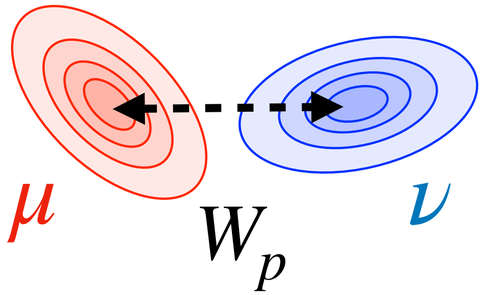


Peyré & Cuturi, 2019

Wasserstein Distances: Metric on Space of Distributions

With $c(x, y) = d(x, y)^p$ for ground metric d,
the p-**Wasserstein distance** is given by

$$W_p(\mu, \nu) = L_c(\mu, \nu)^{1/p}.$$

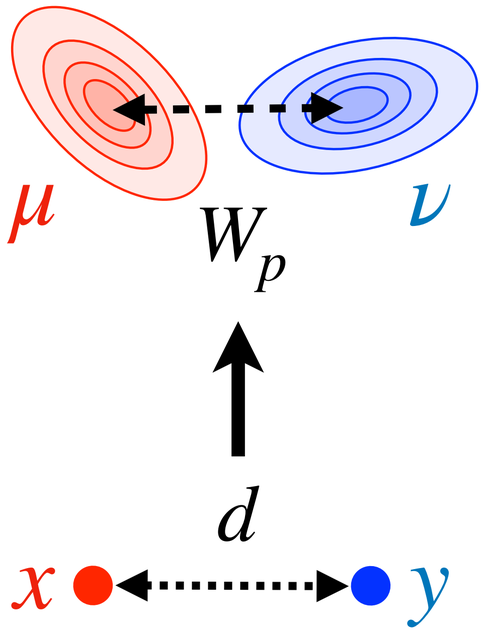


Wasserstein Distances: Metric on Space of Distributions

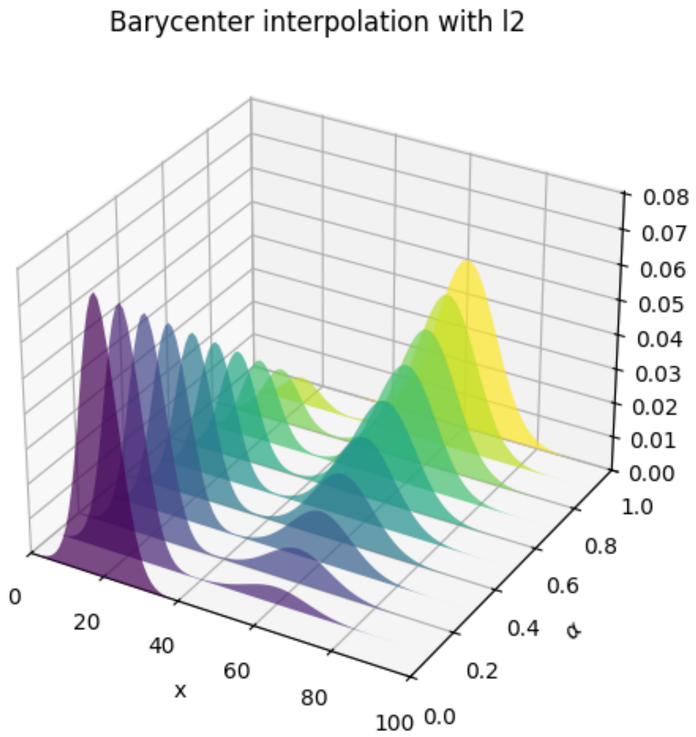
With $c(x, y) = d(x, y)^p$ for ground metric d, the p-**Wasserstein distance** is given by

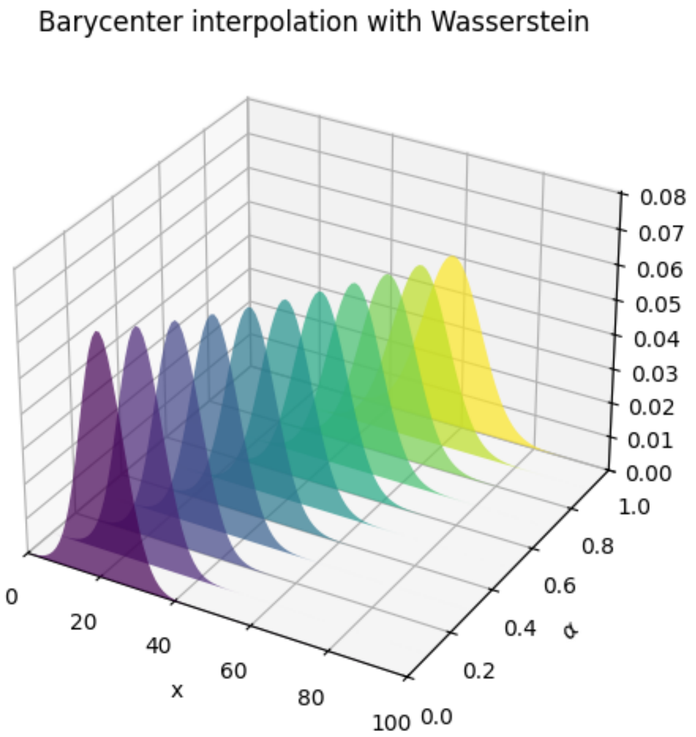
$$W_p(\mu, \nu) = L_c(\mu, \nu)^{1/p}.$$

Optimal transport lifts metric in ambient space to ***metric in distribution space***!

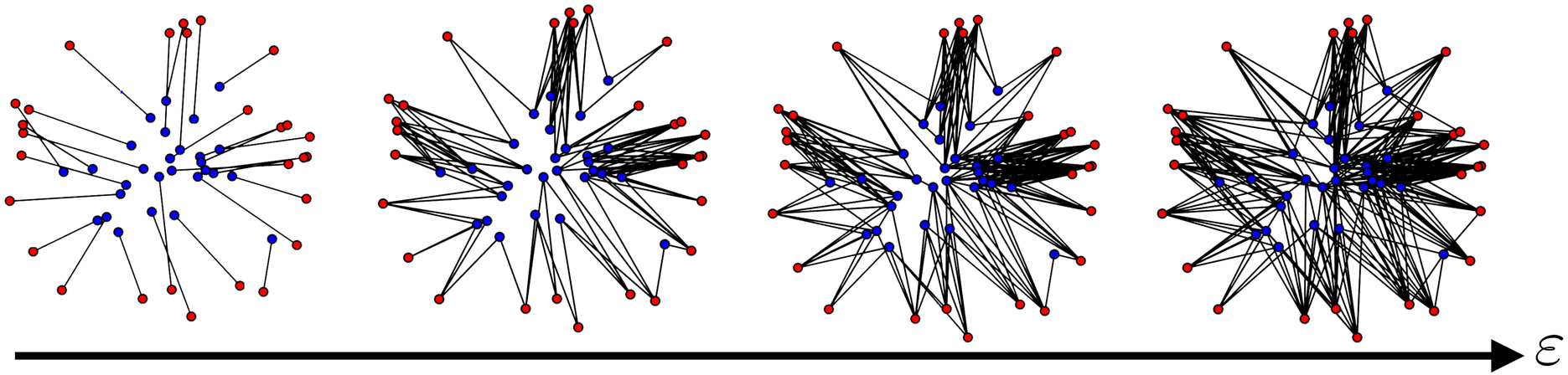


Wasserstein Distances: Barycenter Interpolation





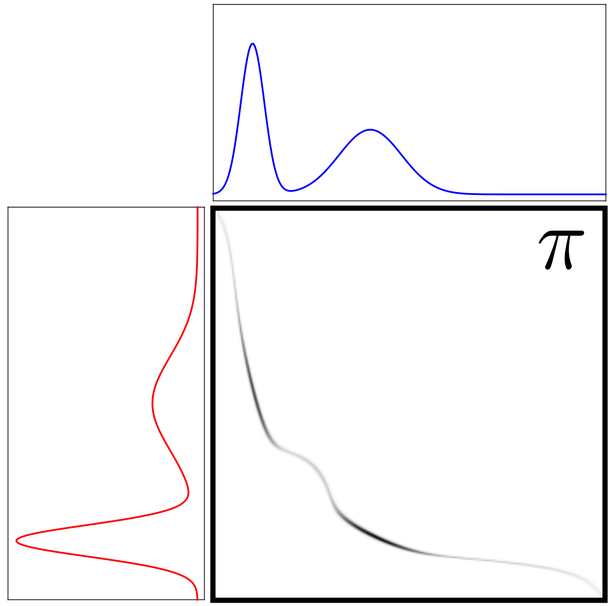
Computational Optimal Transport

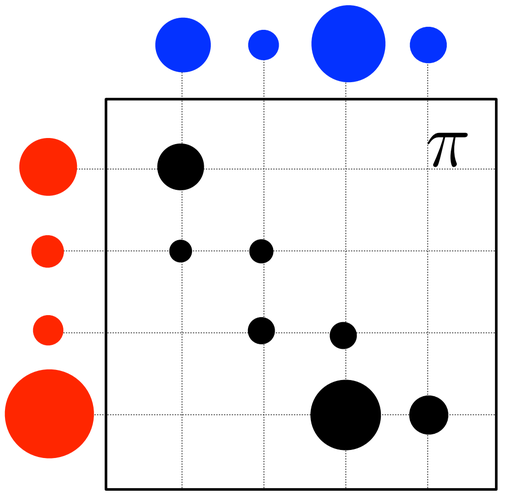


1D Optimal Transport: Exact Solution

***p*-Wasserstein distance** for *1D Euclidean ground metric*

$$c(x, y) = |x - y|^p$$



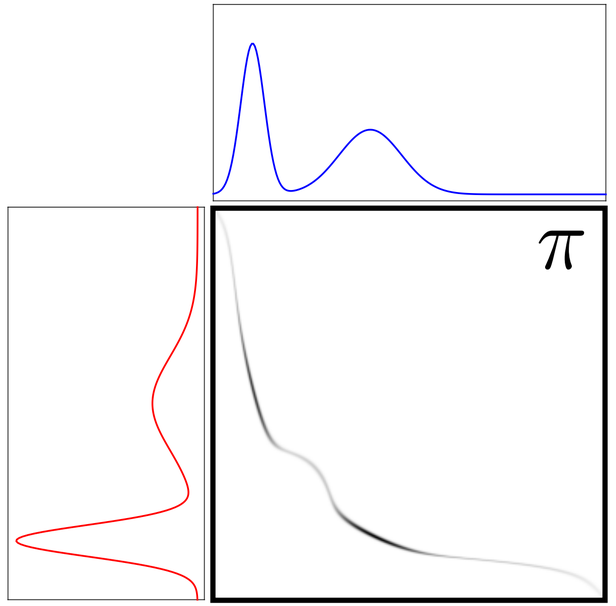


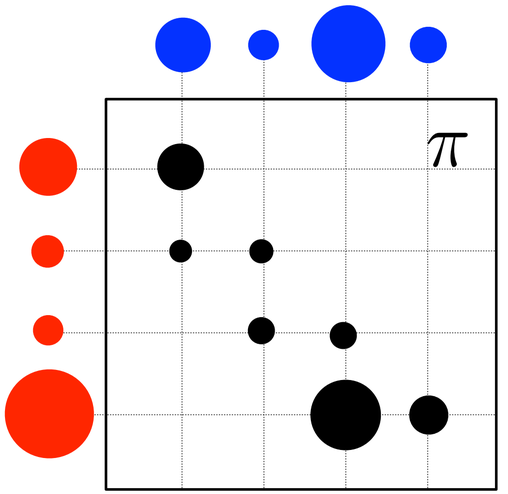
Peyré & Cuturi, 2019

1D Optimal Transport: Exact Solution

***p*-Wasserstein distance** for *1D Euclidean ground metric* $c(x, y) = |x - y|^p$ can be directly computed as

$$W_p(\mu, \nu) = \left(\int_0^1 \left| \mathrm{CDF}_\mu^{-1}(r) - \mathrm{CDF}_\nu^{-1}(r) \right|^p \, \mathrm{d}r \right)^{1/p}.$$



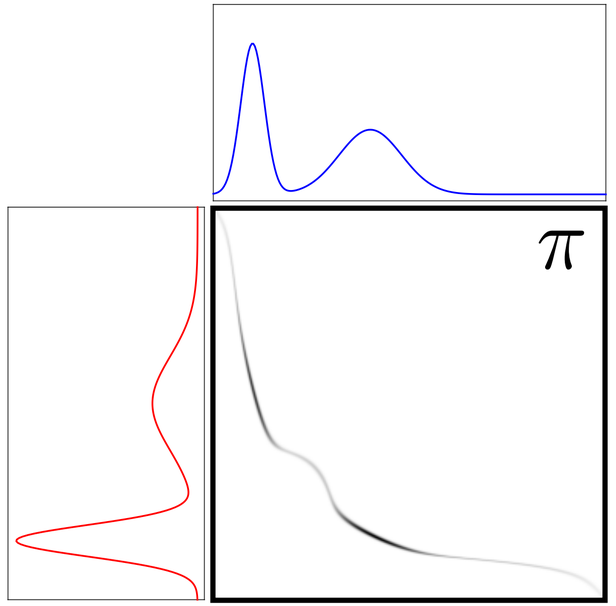


Peyré & Cuturi, 2019

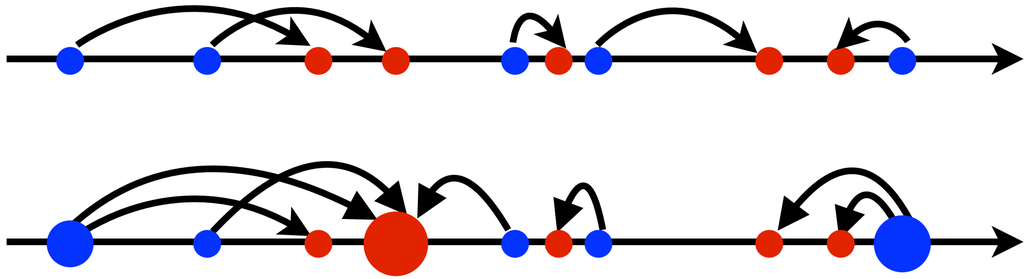
1D Optimal Transport: Exact Solution

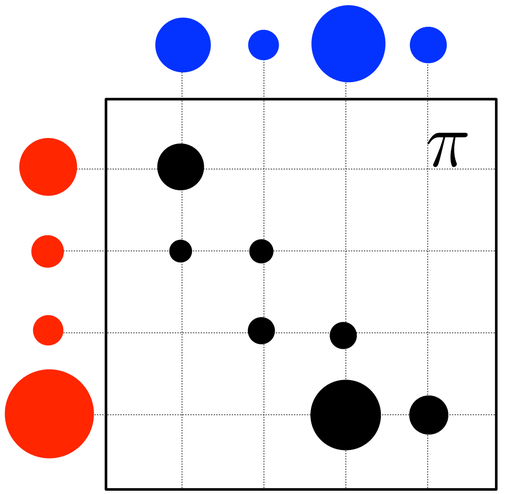
p-**Wasserstein distance** for *1D Euclidean ground metric* $c(x, y) = |x - y|^p$ can be directly computed as

$$W_p(\mu, \nu) = \left(\int_0^1 \left| \mathrm{CDF}_\mu^{-1}(r) - \mathrm{CDF}_\nu^{-1}(r) \right|^p \, \mathrm{d}r \right)^{1/p}.$$



Discrete case solved by **sorting**!





Peyré & Cuturi, 2019

1D Optimal Transport: Sliced Wasserstein Distances

Sliced Wasserstein distance is defined in terms of *1D Wasserstein distances*:

$$\mathrm{SW}_p(\mu, \nu) = \left(\int_{\mathbb{S}^{d-1}} W_p(P_{\theta,\sharp}\mu, P_{\theta,\sharp}\nu)^p \, \mathrm{d}\theta \right)^{1/p}$$

where $P_\theta : \mathbb{R}^d \to \mathbb{R}$ is a projection onto a 1D subspace $x \mapsto x \cdot \theta$.

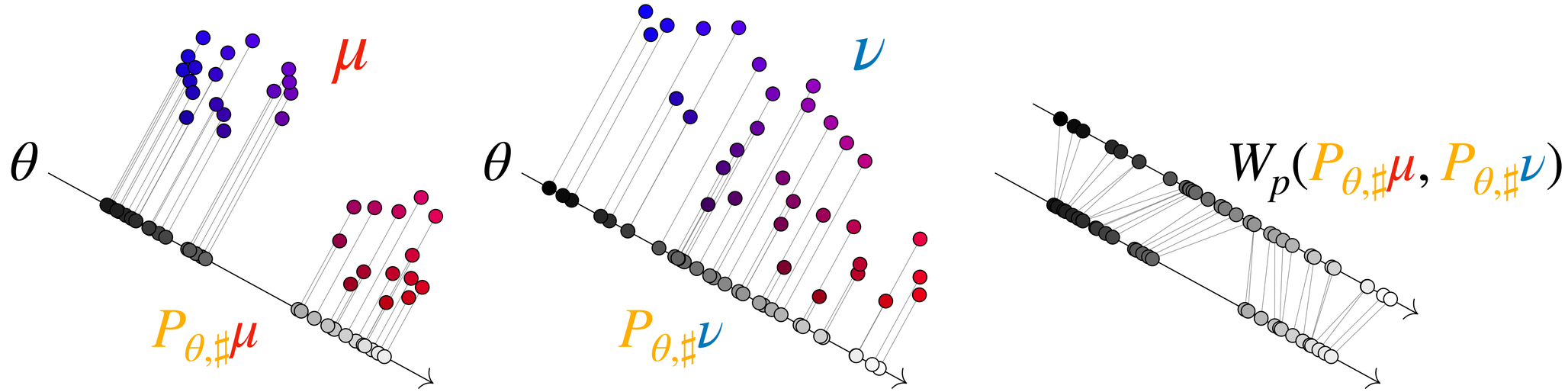
Bonneel et al., 2014

1D Optimal Transport: Sliced Wasserstein Distances

Sliced Wasserstein distance is defined in terms of *1D Wasserstein distances*:

$$\mathrm{SW}_p(\mu, \nu) = \left(\mathbf{E}_{\theta \sim \mathbb{S}^{d-1}} \left[W_p(P_{\theta,\sharp}\mu, P_{\theta,\sharp}\nu)^p \right] \right)^{1/p}$$

where $P_\theta : \mathbb{R}^d \to \mathbb{R}$ is a projection onto a 1D subspace $x \mapsto x \cdot \theta$.



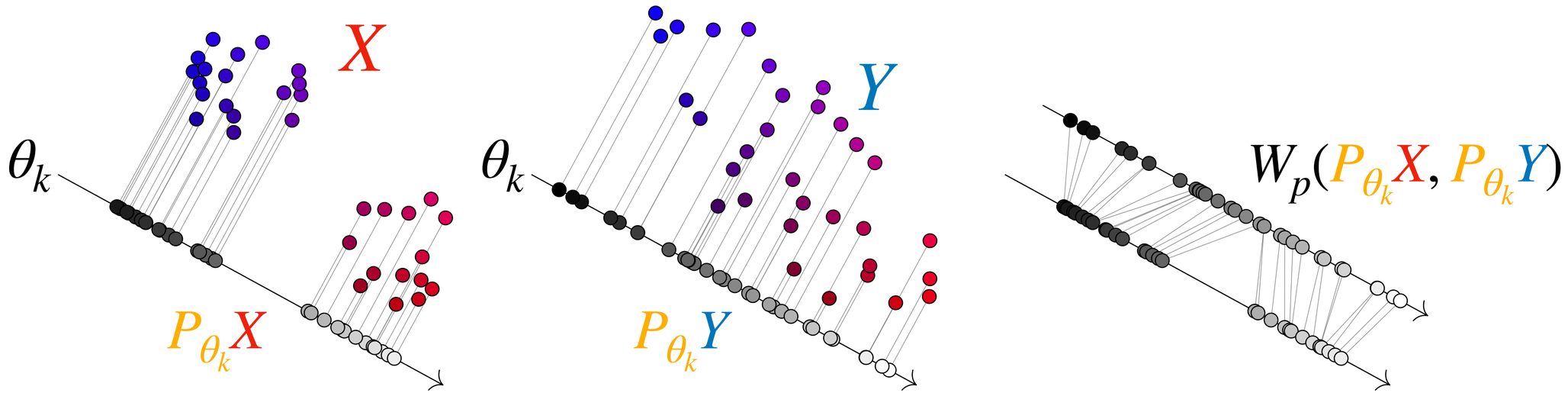
Bonneel et al., 2014

Heitz et al., 2021

1D Optimal Transport: Exercise

Implement **sliced Wasserstein distance** for samples $X = \{x_i\}_{i=1}^n$, $Y = \{y_j\}_{j=1}^n$:

$$\mathrm{SW}_p^{(L)}(X, Y) = \left(\frac{1}{L} \sum_{k=1}^{L} \left[W_p(\{\theta_k \cdot x_i\}_{i=1}^n, \{\theta_k \cdot y_j\}_{j=1}^n)^p \right] \right)^{1/p}$$

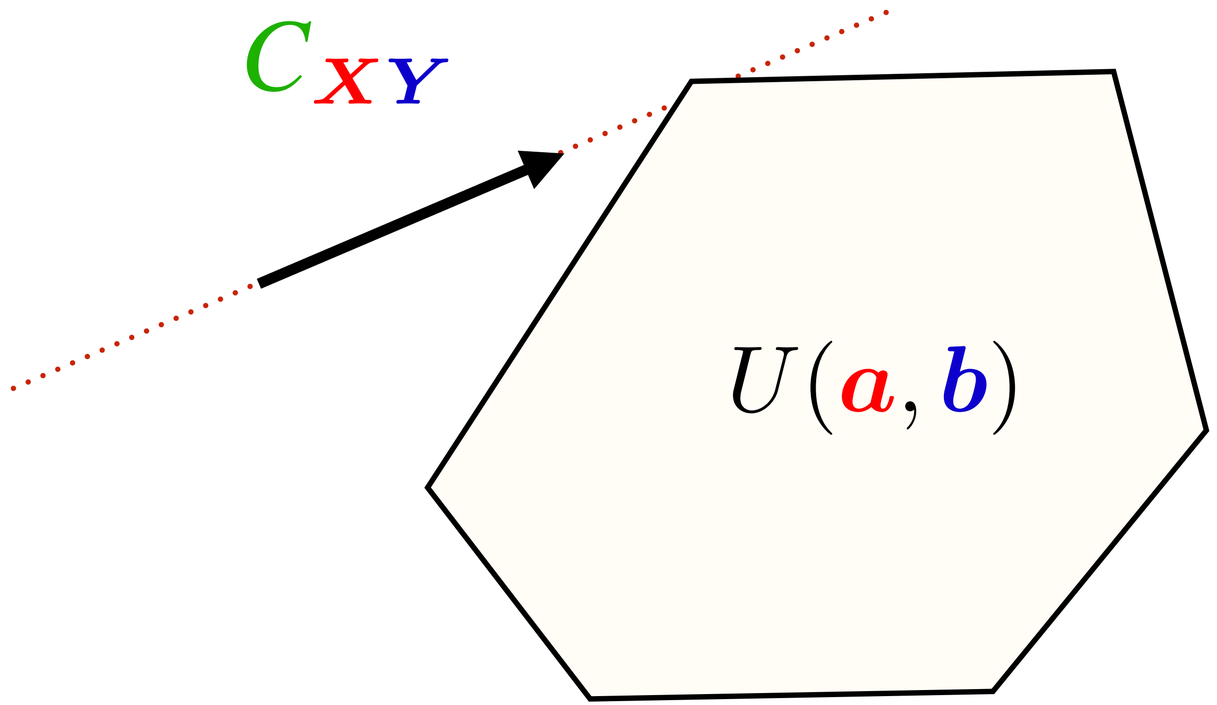


Bonneel et al., 2014

Heitz et al., 2021

Entropy Regularized OT: Motivation

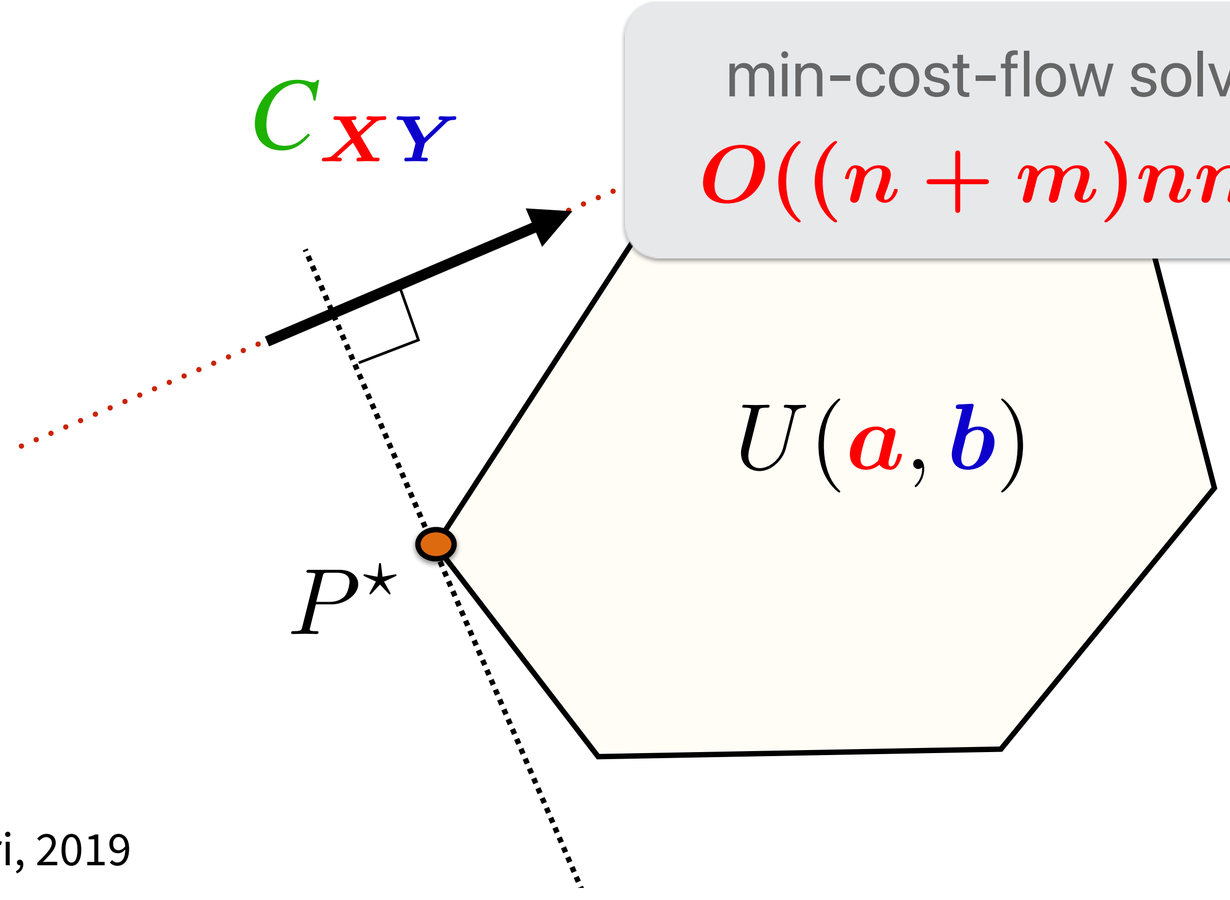
Efficiency: General linear programming solvers are **computationally expensive**.



Peyré & Cuturi, 2019

Entropy Regularized OT: Motivation

Efficiency: General linear programming solvers are **computationally expensive**.



min-cost-flow solver used in practice.

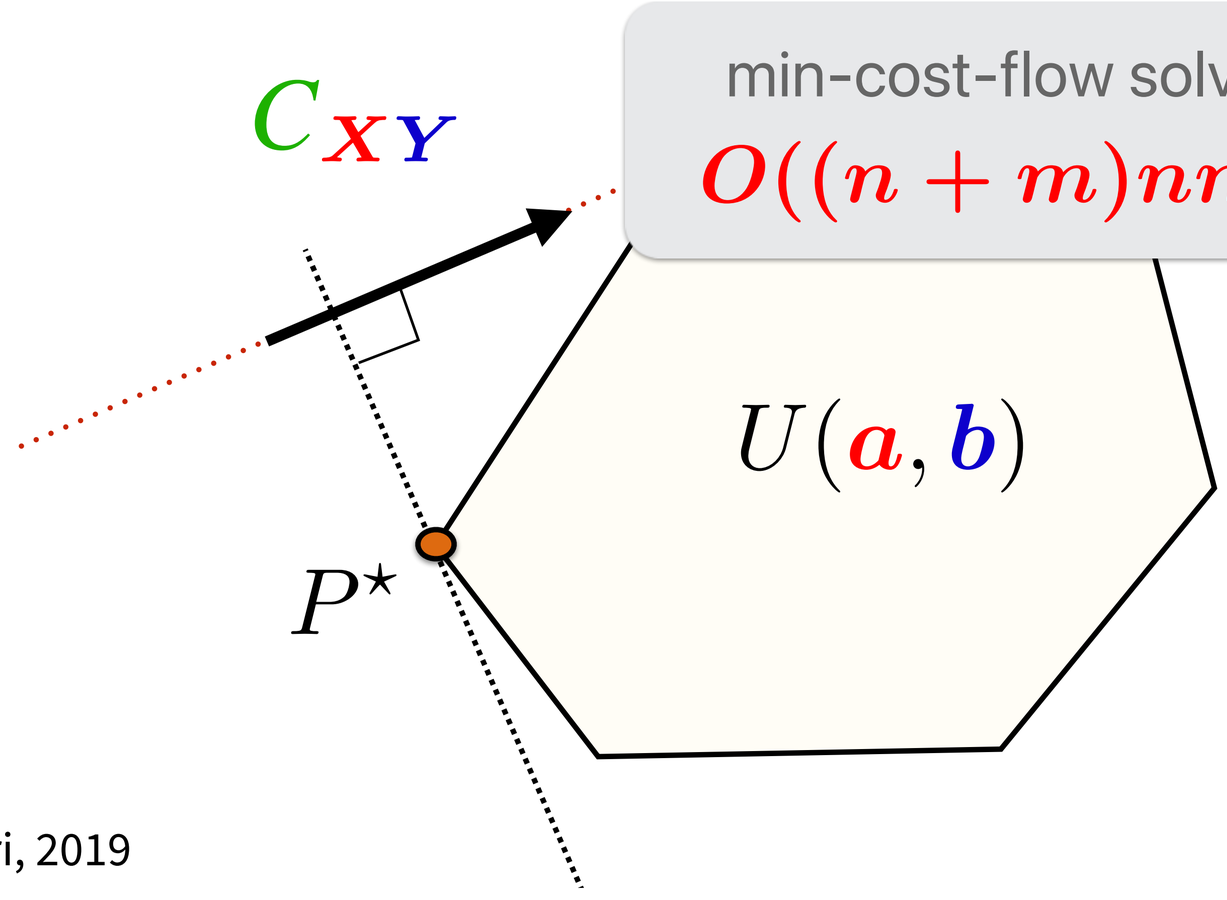
$$O((n+m)nm\log(n+m))$$

Peyré & Cuturi, 2019

Entropy Regularized OT: Motivation

Efficiency: General linear programming solvers are **computationally expensive**.

Smoothness: Exact solution may not be **unique** or **differentiable**.



min-cost-flow solver used in practice.

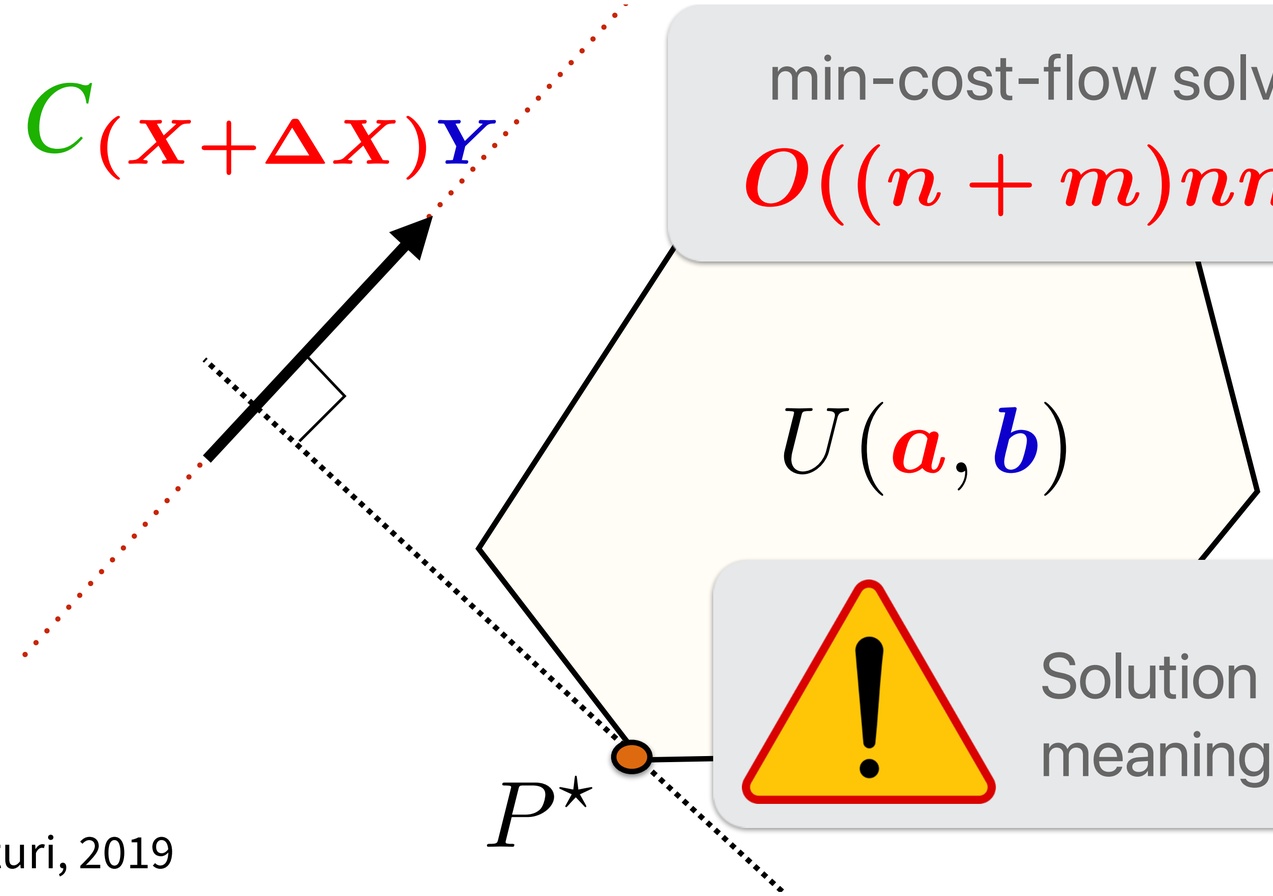
$$O((n+m)nm\log(n+m))$$

Peyré & Cuturi, 2019

Entropy Regularized OT: Motivation

Efficiency: General linear programming solvers are **computationally expensive**.

Smoothness: Exact solution may not be **unique** or **differentiable**.



min-cost-flow solver used in practice.

$$O((n+m)nm\log(n+m))$$

Solution $P^\star$ is not unique, and has no meaningful Jacobian w.r.t input *X* or *Y*.

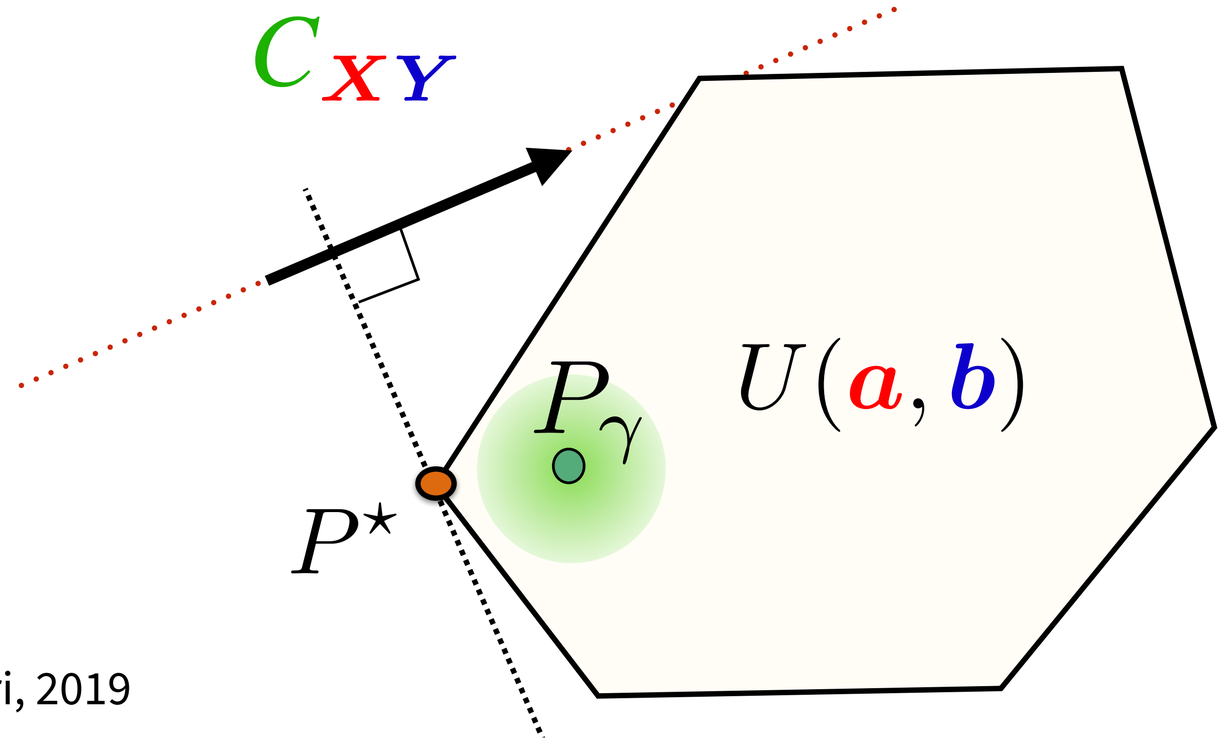
Peyré & Cuturi, 2019

Entropy Regularized OT: Motivation

Efficiency: General linear programming solvers are **computationally expensive**.

Smoothness: Exact solution may not be **unique** or **differentiable**.

Robustness: Do not want to **overfit** on finite samples.



Peyré & Cuturi, 2019

Entropy Regularized OT: Formulation

$$L_c^{(\varepsilon)}(\mu, \nu) = \min_{\pi} \left\{ \int_{X \times Y} c(x, y)\, \mathrm{d}\pi(x, y) + \varepsilon\, \mathrm{KL}(\pi \mid \mu \otimes \nu) : \pi \in \Pi(\mu, \nu) \right\}$$

total cost

relative entropy

constraint

Entropy Regularized OT: Formulation

total cost relative entropy constraint

$$L_c^{(\varepsilon)}(\mu,\nu) = \min_{\pi} \left\{ \int_{X\times Y} c(x,y)\,\mathrm{d}\pi(x,y) + \varepsilon\,\mathrm{KL}(\pi \mid \mu\otimes\nu) : \pi \in \Pi(\mu,\nu) \right\}$$

regularization parameter

Entropy Regularized OT: Formulation

$$L_c^{(\varepsilon)}(\mu,\nu) = \min_{\pi}\left\{\int_{X\times Y} c(x,y)\,\mathrm{d}\pi(x,y) + \varepsilon\,\mathrm{KL}(\pi \mid \mu\otimes\nu) : \pi\in\Pi(\mu,\nu)\right\}$$

total cost

relative entropy

constraint

regularization parameter

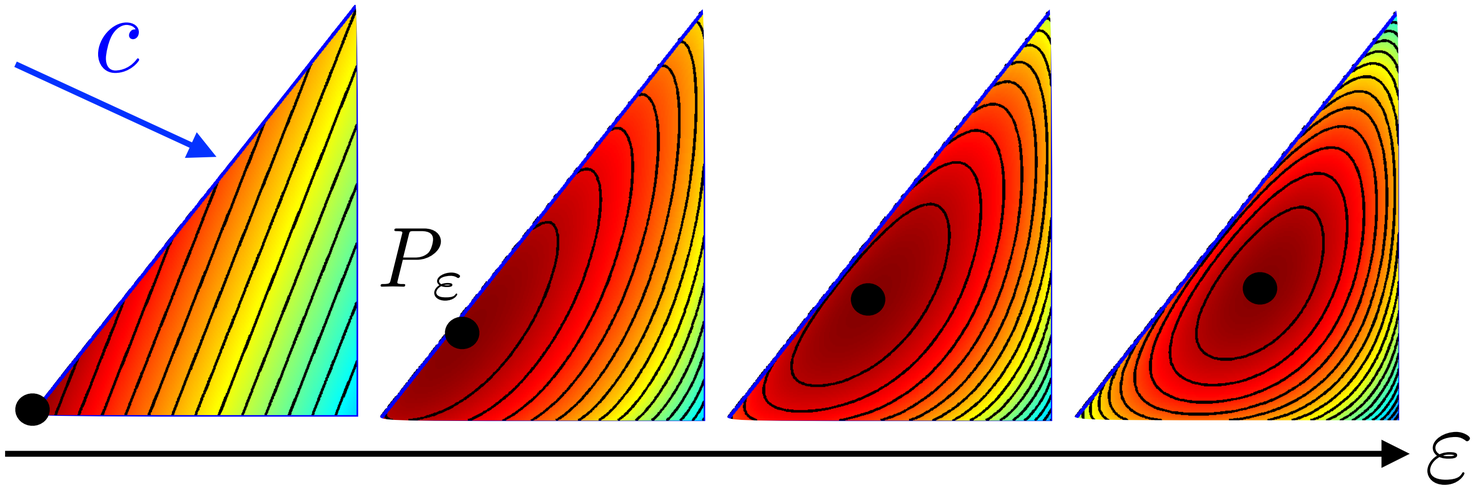
reference distribution

Entropy Regularized OT: Formulation

total cost relative entropy constraint

$$L_c^{(\varepsilon)}(\mu,\nu) = \min_{\pi} \left\{ \int_{X\times Y} c(x,y)\,\mathrm{d}\pi(x,y) + \varepsilon\,\mathrm{KL}(\pi \mid \mu\otimes\nu) : \pi \in \Pi(\mu,\nu) \right\}$$

regularization parameter reference distribution



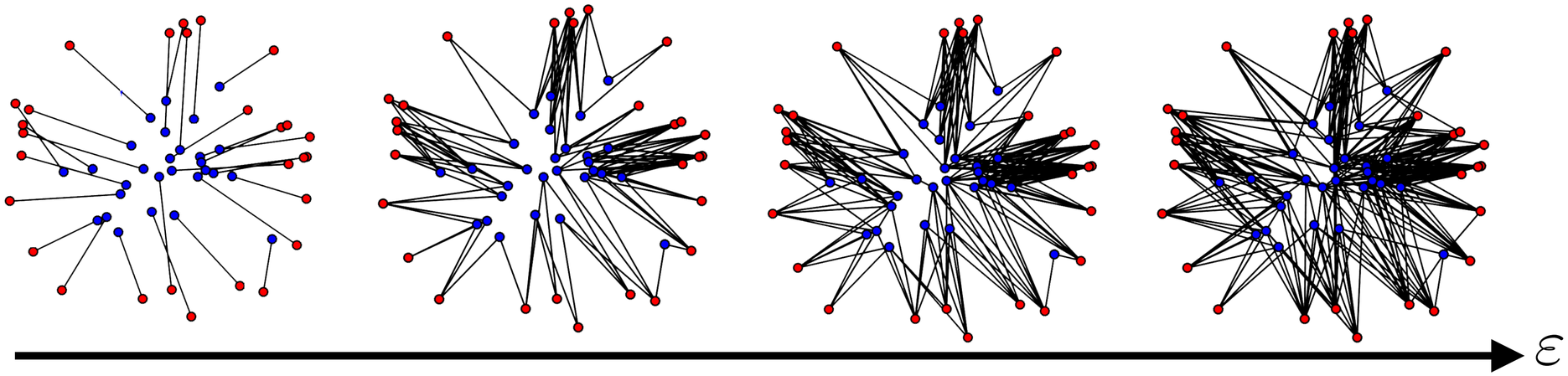
Peyré & Cuturi, 2019

Entropy Regularized OT: Formulation

total cost relative entropy constraint

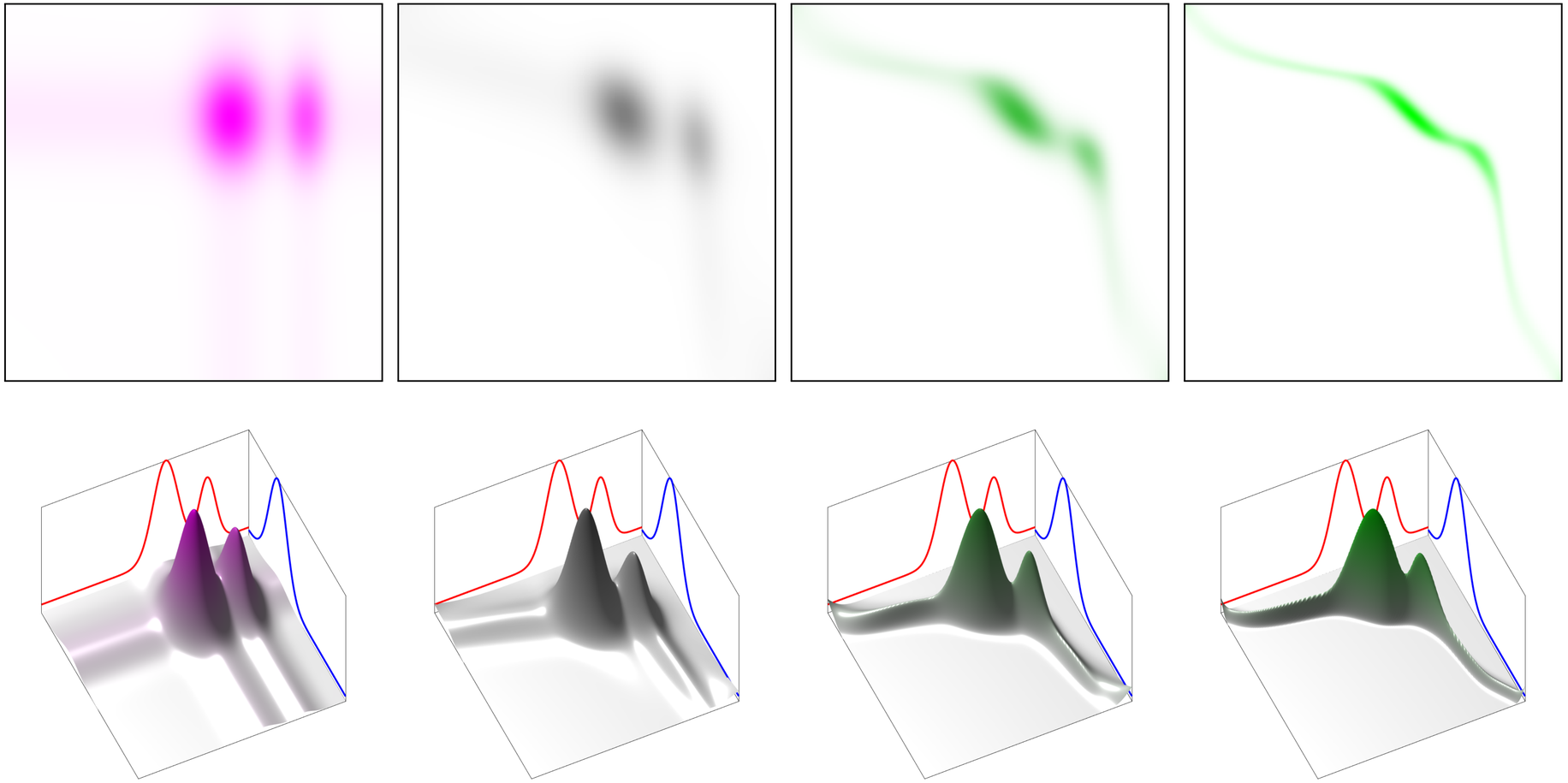
$$L_c^{(\varepsilon)}(\mu,\nu) = \min_{\pi} \left\{ \int_{X\times Y} c(x,y)\,\mathrm{d}\pi(x,y) + \varepsilon\,\mathrm{KL}(\pi \mid \mu\otimes\nu) : \pi \in \Pi(\mu,\nu) \right\}$$

regularization parameter reference distribution



Peyré & Cuturi, 2019

Entropy Regularized OT: Formulation



$\varepsilon = 10$ $\varepsilon = 1$ $\varepsilon = 10^{-1}$ $\varepsilon = 10^{-2}$

Peyré & Cuturi, 2019

Entropy Regularized OT: Discrete Formulation

$$L_C^{(\varepsilon)}(a, b) = \min_P \left\{ \sum_{i,j} C_{ij} P_{ij} - \varepsilon \, \mathrm{H}(P) : P \in U(a, b) \right\}$$

Entropy Regularized OT: Discrete Formulation

$$L_C^{(\varepsilon)}(a, b) = \min_P \left\{ \sum_{i,j} C_{ij} P_{ij} - \varepsilon \, \mathrm{H}(P) : P \in U(a, b) \right\}$$

Dual formulation:

$$\widetilde{L}_C^{(\varepsilon)}(f, g) = \min_P \left\{ \sum_{i,j} C_{ij} P_{ij} - \varepsilon \, \mathrm{H}(P) - \sum_i f_i \left(\sum_j P_{ij} - a_i \right) - \sum_j g_j \left(\sum_i P_{ij} - b_j \right) \right\}$$

Entropy Regularized OT: Discrete Formulation

$$L_C^{(\varepsilon)}(a, b) = \min_P \left\{ \sum_{i,j} C_{ij} P_{ij} - \varepsilon \, \mathrm{H}(P) : P \in U(a, b) \right\}$$

Dual formulation:

$$\widetilde{L}_C^{(\varepsilon)}(f, g) = \min_P \left\{ \sum_{i,j} C_{ij} P_{ij} - \varepsilon \, \mathrm{H}(P) - \sum_i f_i \left(\sum_j P_{ij} - a_i \right) - \sum_j g_j \left(\sum_i P_{ij} - b_j \right) \right\}$$

$$P_{ij} = u_i K_{ij} v_j \quad \text{where} \quad u_i = \exp(f_i/\varepsilon), v_j = \exp(g_j/\varepsilon), K_{ij} = \exp(-C_{ij}/\varepsilon)$$

Entropy Regularized OT: Sinkhorn Algorithm

$$P_{ij} = u_i K_{ij} v_j \quad \text{where} \quad u_i = \exp(f_i/\varepsilon), v_j = \exp(g_j/\varepsilon), K_{ij} = \exp(-C_{ij}/\varepsilon)$$

Impose constraints: $\sum_i P_{ij} = b_j, \sum_j P_{ij} = a_i, P_{ij} \geq 0$

Cuturi, NeurIPS 2013

Entropy Regularized OT: Sinkhorn Algorithm

$$P_{ij} = u_i K_{ij} v_j \quad \text{where} \quad u_i = \exp(f_i/\varepsilon), v_j = \exp(g_j/\varepsilon), K_{ij} = \exp(-C_{ij}/\varepsilon)$$

Impose constraints: $\sum_i P_{ij} = b_j, \sum_j P_{ij} = a_i, P_{ij} \geq 0$

Sinkhorn iterations—also known as **iterative proportional fitting (IPF):**

$$u_i^{(l+1)} = a_i / (K v^{(l)})_i$$

$$v_j^{(l+1)} = b_j / (K^\top u^{(l+1)})_j$$

Cuturi, NeurIPS 2013

Entropy Regularized OT: Sinkhorn Algorithm

$$P_{ij} = u_i K_{ij} v_j \quad \text{where} \quad u_i = \exp(f_i/\varepsilon), v_j = \exp(g_j/\varepsilon), K_{ij} = \exp(-C_{ij}/\varepsilon)$$

Impose constraints: $\sum_i P_{ij} = b_j, \sum_j P_{ij} = a_i, P_{ij} \geq 0$

Sinkhorn iterations—also known as **iterative proportional fitting (IPF):**

$$\mathcal{O}(n^2 \log n)$$

$$u_i^{(l+1)} = a_i / (K v^{(l)})_i$$

$$v_j^{(l+1)} = b_j / (K^\top u^{(l+1)})_j$$

Cuturi, NeurIPS 2013

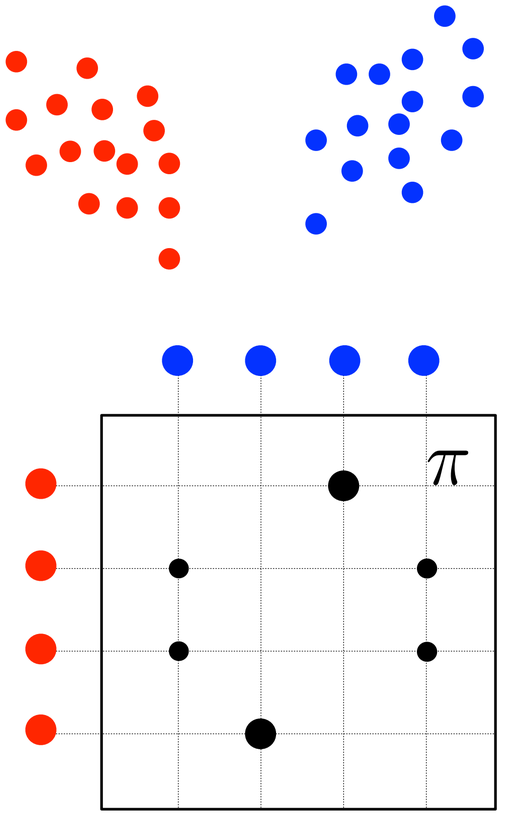
Entropy Regularized OT: Exercise

Implement the **Sinkhorn algorithm** for samples $X = \{x_i\}_{i=1}^n$, $Y = \{y_j\}_{j=1}^n$.

$$P_{ij} = u_i K_{ij} v_j, \quad K_{ij} = \exp(-C_{ij}/\varepsilon)$$

$$u_i^{(l+1)} = a_i / (K v^{(l)})_i$$

$$v_j^{(l+1)} = b_j / (K^\top u^{(l+1)})_j$$



Peyré & Cuturi, 2019

Entropy Regularized OT: Sinkhorn Divergence

Define a distance $W_p^{(\varepsilon)}(\mu, \nu) = L_c^{(\varepsilon)}(\mu, \nu)^{1/p}$?

Entropy Regularized OT: Sinkhorn Divergence

Define a distance $W_p^{(\varepsilon)}(\mu, \nu) = L_c^{(\varepsilon)}(\mu, \nu)^{1/p}$?

$$L_c^{(\varepsilon)}(\mu, \nu) = \min_{\pi} \left\{ \int_{X \times Y} c(x, y)\, \mathrm{d}\pi(x, y) + \varepsilon\, \mathrm{KL}(\pi \mid \mu \otimes \nu) : \pi \in \Pi(\mu, \nu) \right\}$$

Entropy Regularized OT: Sinkhorn Divergence

Define a distance $W_p^{(\varepsilon)}(\mu, \nu) = L_c^{(\varepsilon)}(\mu, \nu)^{1/p}$?

$$L_c^{(\varepsilon)}(\mu, \nu) = \min_{\pi} \left\{ \int_{X \times Y} c(x, y)\, \mathrm{d}\pi(x, y) + \varepsilon\, \mathrm{KL}(\pi \mid \mu \otimes \nu) : \pi \in \Pi(\mu, \nu) \right\}$$

Entropic bias $L_c^{(\varepsilon)}(\mu, \mu) \neq 0$

Feydy et al., AISTATS 2019

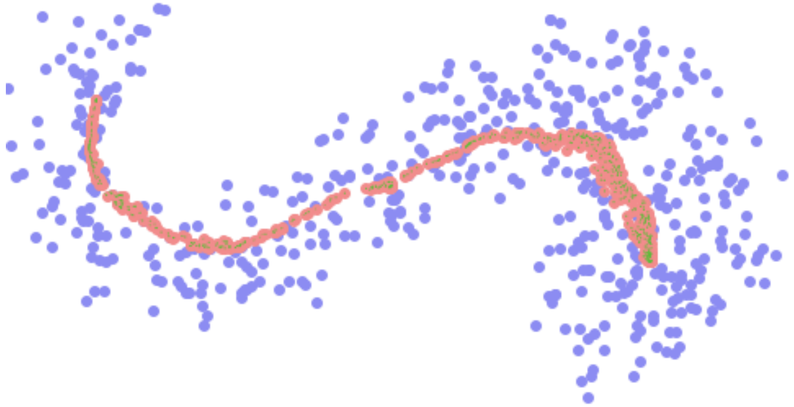
Entropy Regularized OT: Sinkhorn Divergence

Define a distance $W_p^{(\varepsilon)}(\mu, \nu) = L_c^{(\varepsilon)}(\mu, \nu)^{1/p}$?

$$L_c^{(\varepsilon)}(\mu, \nu) = \min_{\pi} \left\{ \int_{X \times Y} c(x, y)\, \mathrm{d}\pi(x, y) + \varepsilon\, \mathrm{KL}(\pi \mid \mu \otimes \nu) : \pi \in \Pi(\mu, \nu) \right\}$$

Entropic bias $L_c^{(\varepsilon)}(\mu, \mu) \neq 0$

“Shrinkage” $\min_{\mu} L_c^{(\varepsilon)}(\mu, \nu)$ ⟶



Feydy et al., AISTATS 2019

Entropy Regularized OT: Sinkhorn Divergence

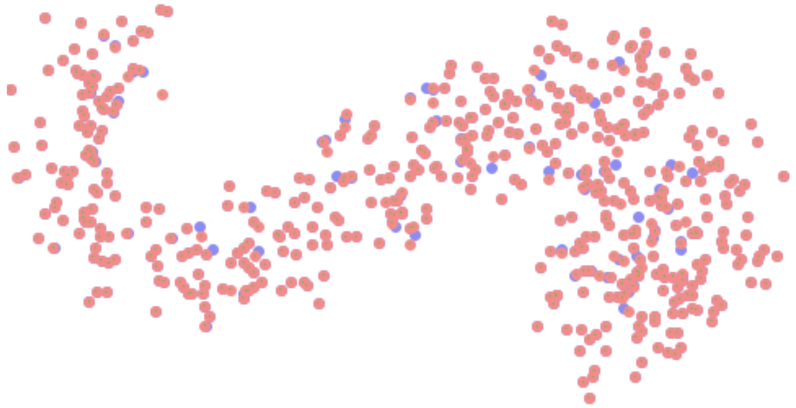
Sinkhorn divergence

$$S_c^{(\varepsilon)}(\mu,\nu) = L_c^{(\varepsilon)}(\mu,\nu) - \frac{1}{2}\left(L_c^{(\varepsilon)}(\mu,\mu) + L_c^{(\varepsilon)}(\nu,\nu)\right)$$

is an ***unbiased, smooth, symmetric, positive definite*** divergence with

$$0 = S_c^{(\varepsilon)}(\mu,\mu) \leq S_c^{(\varepsilon)}(\mu,\nu) \text{ and } \mu = \nu \iff S_c^{(\varepsilon)}(\mu,\nu) = 0.$$

$$\min_{\mu} L_c^{(\varepsilon)}(\mu,\nu) \longrightarrow$$



Feydy et al., AISTATS 2019

Entropy Regularized OT: Sinkhorn Divergence

Sinkhorn divergence

$$S_c^{(\varepsilon)}(\mu, \nu) = L_c^{(\varepsilon)}(\mu, \nu) - \frac{1}{2}\left(L_c^{(\varepsilon)}(\mu, \mu) + L_c^{(\varepsilon)}(\nu, \nu)\right)$$

is an ***unbiased, smooth, symmetric, positive definite*** divergence with

$$0 = S_c^{(\varepsilon)}(\mu, \mu) \leq S_c^{(\varepsilon)}(\mu, \nu) \text{ and } \mu = \nu \iff S_c^{(\varepsilon)}(\mu, \nu) = 0.$$

ε interpolates between **Wasserstein** and **MMD**:

$$W_p(\mu, \nu) \overset{0 \leftarrow \varepsilon}{\longleftarrow} S_c^{(\varepsilon)}(\mu, \nu) \overset{\varepsilon \to \infty}{\longrightarrow} \mathrm{MMD}_{-c}(\mu, \nu)$$

Feydy et al., AISTATS 2019

Entropy Regularized OT: Sinkhorn Divergence

Sinkhorn divergence (alternate definition)

$$\overline{S}_c^{(\varepsilon)}(\mu, \nu) = \overline{L}_c^{(\varepsilon)}(\mu, \nu) - \frac{1}{2}\left(\overline{L}_c^{(\varepsilon)}(\mu, \mu) + \overline{L}_c^{(\varepsilon)}(\nu, \nu)\right),$$

where $\overline{L}_c^{(\varepsilon)}(\mu, \nu) = \int_{X \times Y} c(x, y)\, \mathrm{d}\pi^{(\varepsilon)\star}(x, y)$ is evaluated *without the entropy term*.

Genevay et al., AISTATS 2018

Entropy Regularized OT: Sinkhorn Divergence

Sinkhorn divergence (alternate definition)

$$\overline{S}_c^{(\varepsilon)}(\mu, \nu) = \overline{L}_c^{(\varepsilon)}(\mu, \nu) - \frac{1}{2}\left(\overline{L}_c^{(\varepsilon)}(\mu, \mu) + \overline{L}_c^{(\varepsilon)}(\nu, \nu)\right),$$

where $\overline{L}_c^{(\varepsilon)}(\mu, \nu) = \int_{X \times Y} c(x, y)\, \mathrm{d}\pi^{(\varepsilon)\star}(x, y)$ is evaluated *without the entropy term*.

$\overline{S}_c^{(\varepsilon)}(\mu, \nu)$ has similar properties to $S_c^{(\varepsilon)}(\mu, \nu)$.

POT library implements $\overline{S}_c^{(\varepsilon)}(\mu, \nu)$, while **OTT** and **GeomLoss** implement $S_c^{(\varepsilon)}(\mu, \nu)$.

Genevay et al., AISTATS 2018

Entropy Regularized OT: Sinkhorn Divergence

Define a distance $W_p^{(\varepsilon)}(\mu, \nu) = S_c^{(\varepsilon)}(\mu, \nu)^{1/p}$?

Entropy Regularized OT: Sinkhorn Divergence

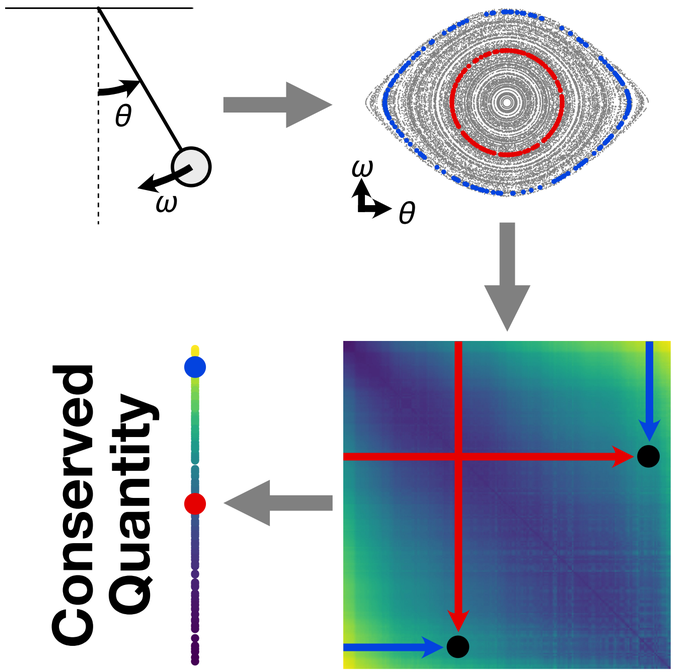
Define a distance $W_p^{(\varepsilon)}(\mu, \nu) = S_c^{(\varepsilon)}(\mu, \nu)^{1/p}$?

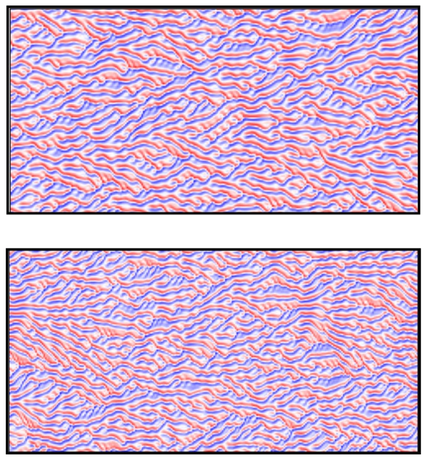
Unfortunately, this does not always satisfy **triangle inequality**

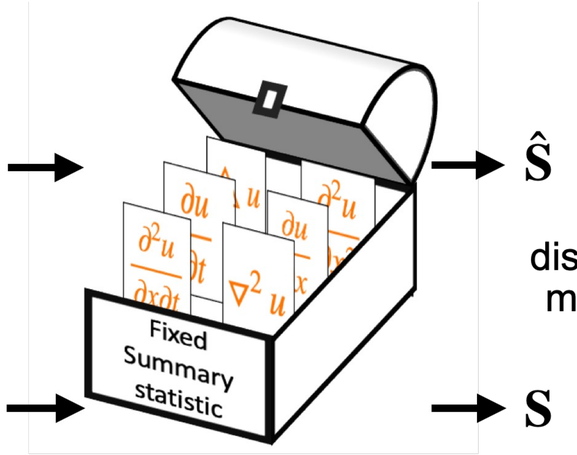
$$W_p^{(\varepsilon)}(\mu, \rho) + W_p^{(\varepsilon)}(\rho, \nu) \ngeq W_p^{(\varepsilon)}(\mu, \nu)$$

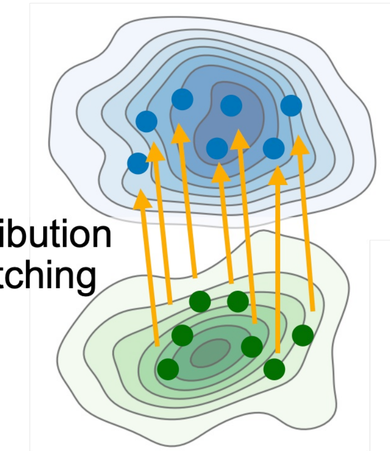
but, nevertheless, is often used in practice as a loss function.

Machine Learning Applications





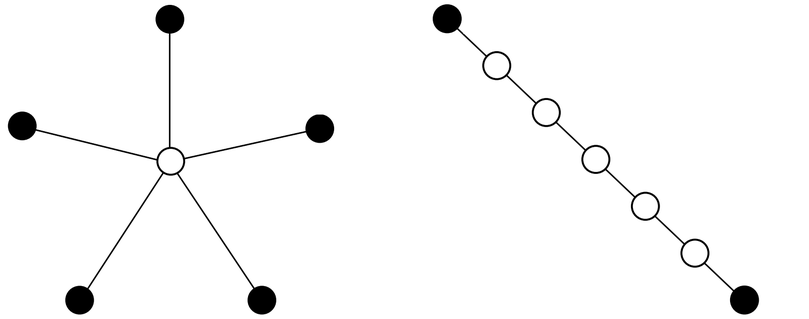




Shape Analysis: Barycenters & Interpolation

Weighted Fréchet mean:

$$\min_{\mu} \sum_{i} \alpha_i W_2(\mu, \nu_i)^2$$



Barycenter

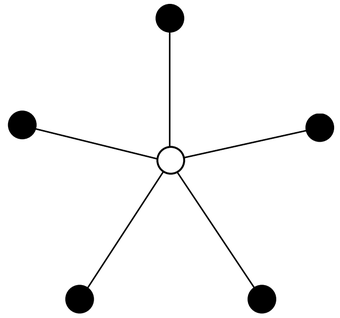
Displacement interpolation

Solomon et al., ACM Graphics 2015

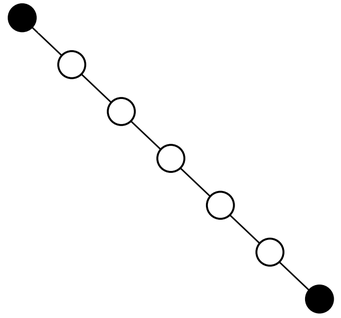
Shape Analysis: Barycenters & Interpolation

Weighted Fréchet mean:

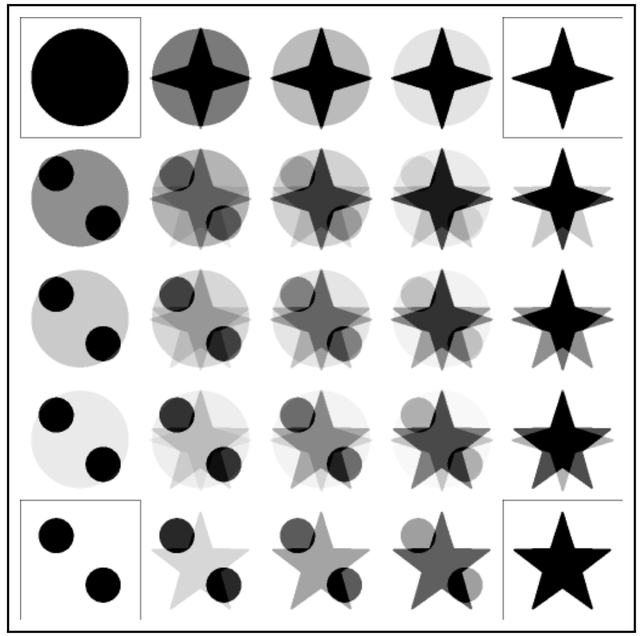
$$\min_{\mu} \sum_i \alpha_i W_2(\mu, \nu_i)^2$$



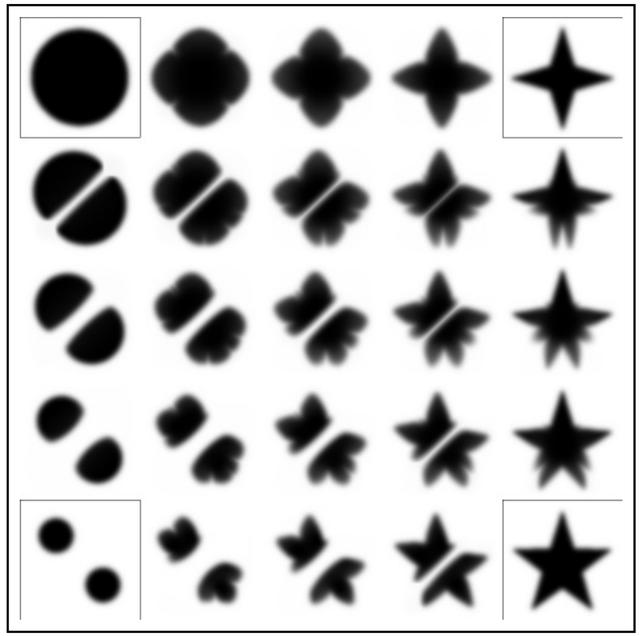
Barycenter



Displacement interpolation



Euclidean barycenter



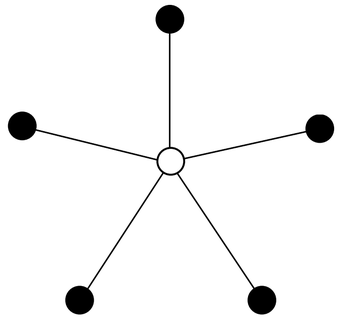
Wasserstein barycenter

Solomon et al., ACM Graphics 2015

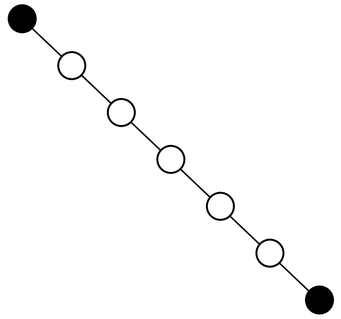
Shape Analysis: Barycenters & Interpolation

Weighted Fréchet mean:

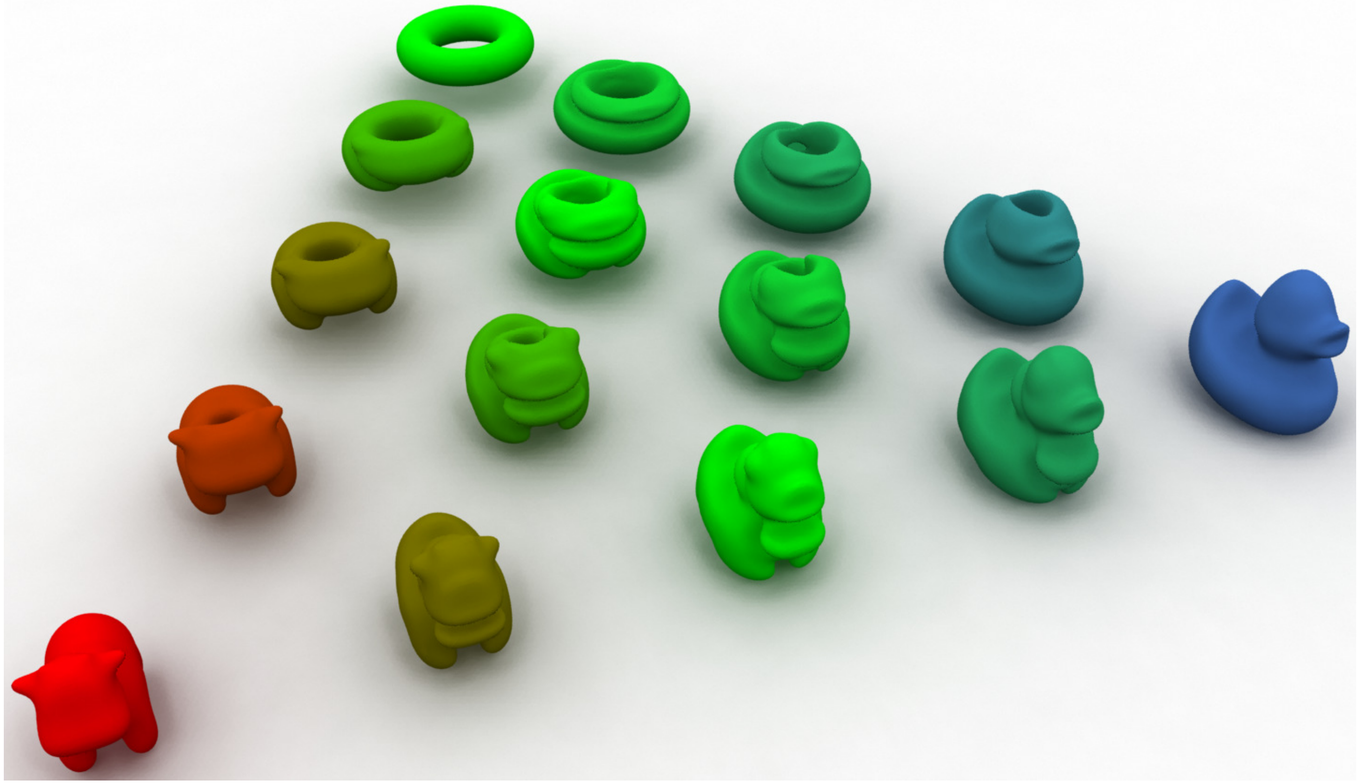
$$\min_{\mu} \sum_{i} \alpha_i W_2(\mu, \nu_i)^2$$



Barycenter



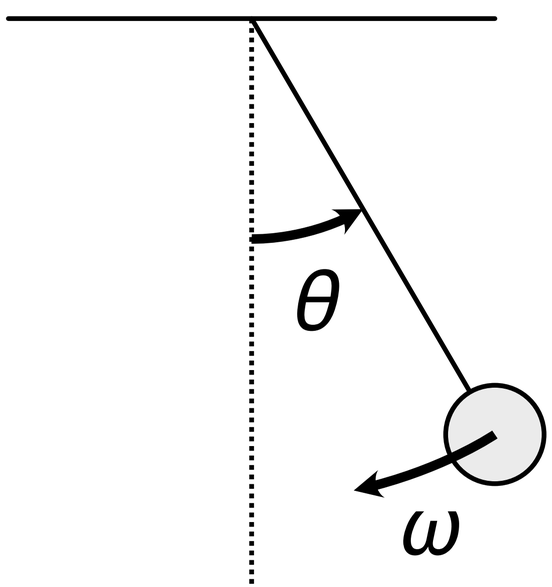
Displacement interpolation

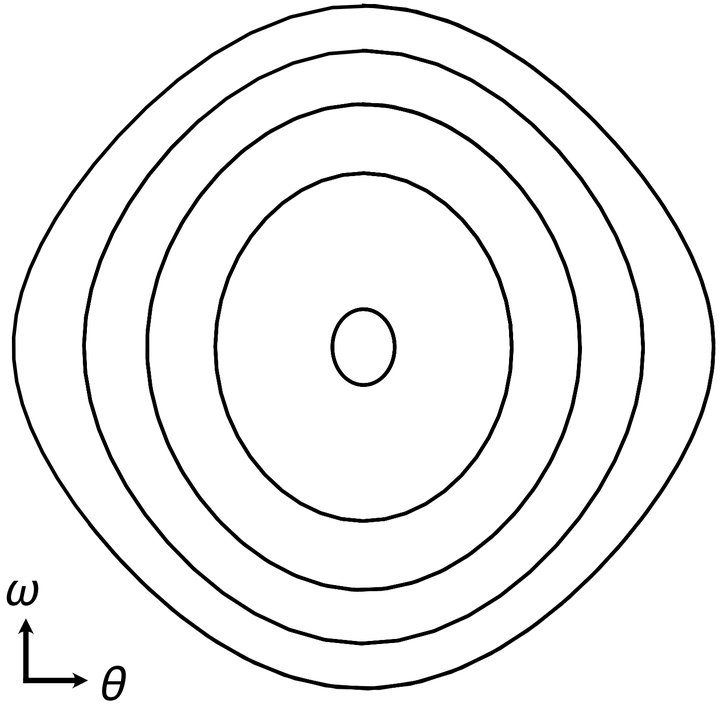


Solomon et al., ACM Graphics 2015

Shape Analysis: Manifold Learning


Discovering conservation laws: OT + diffusion maps

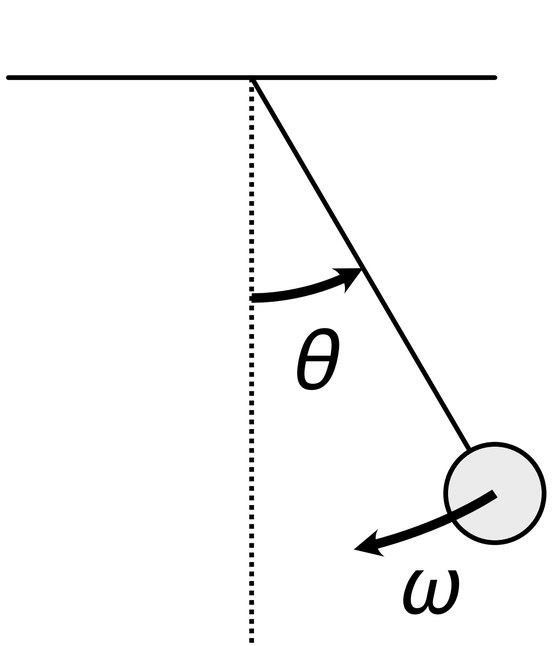


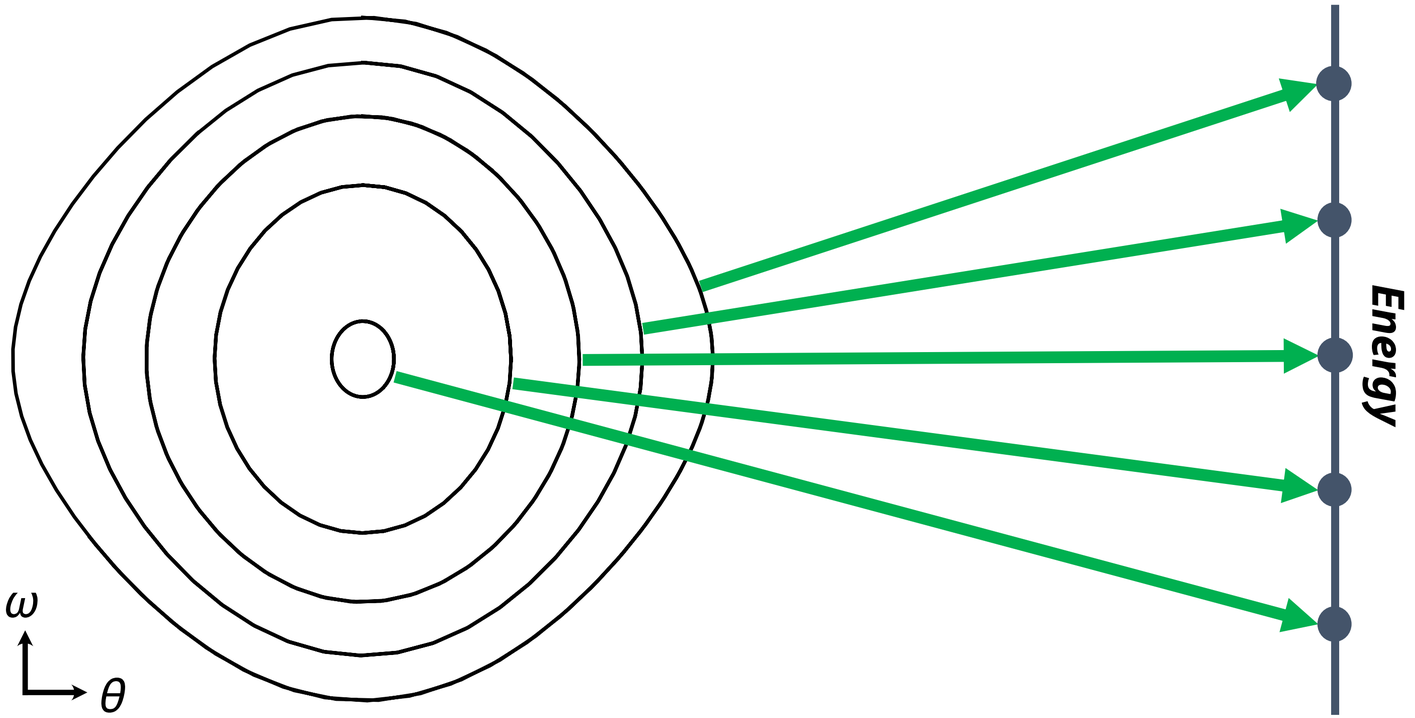


Lu et al., Nat Comm 2023

Shape Analysis: Manifold Learning


Discovering conservation laws: OT + diffusion maps

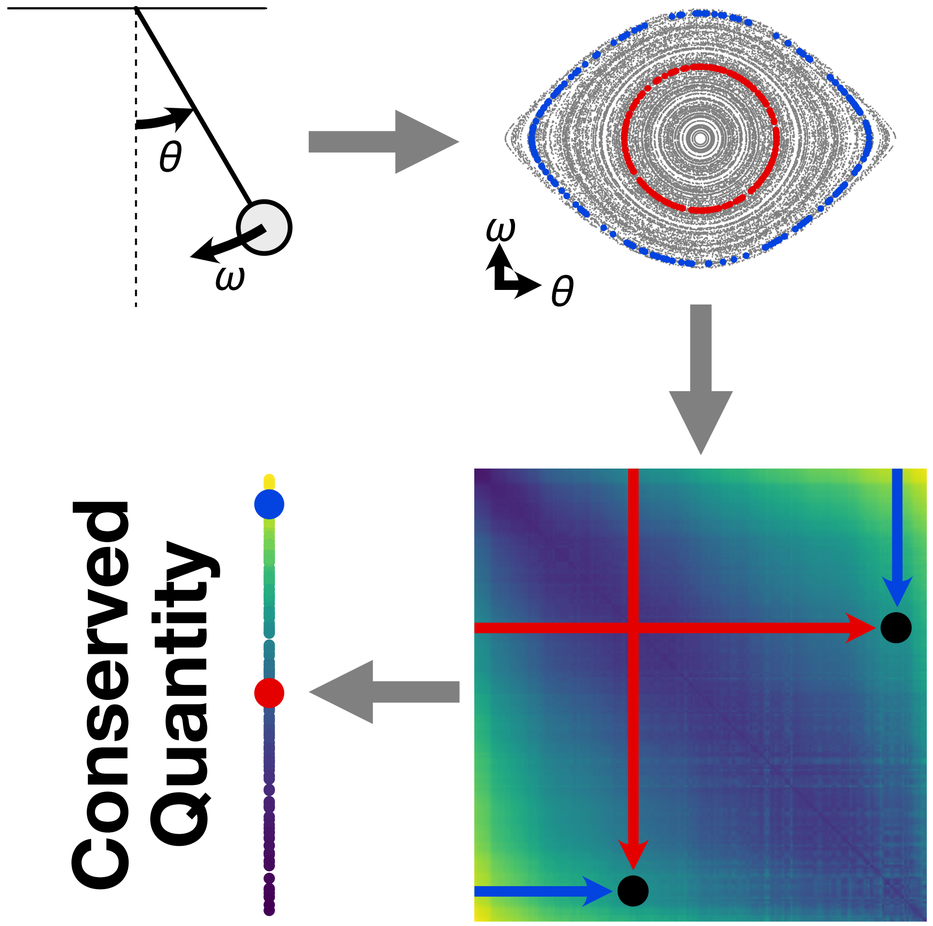


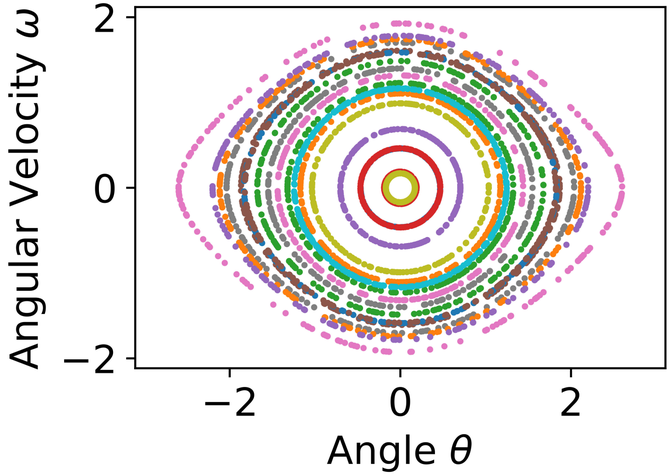


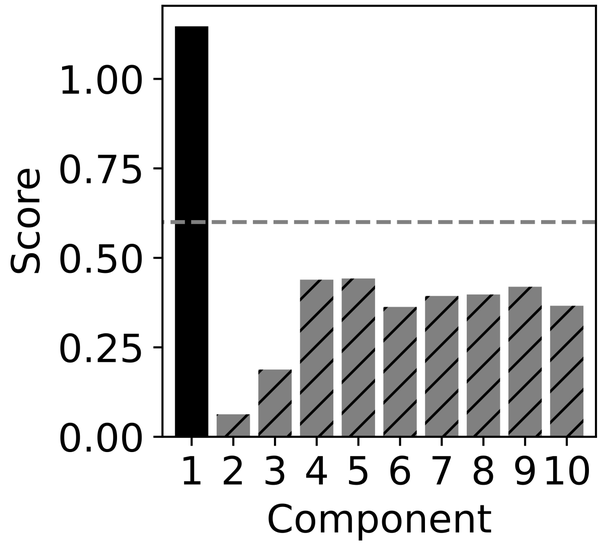
Lu et al., Nat Comm 2023

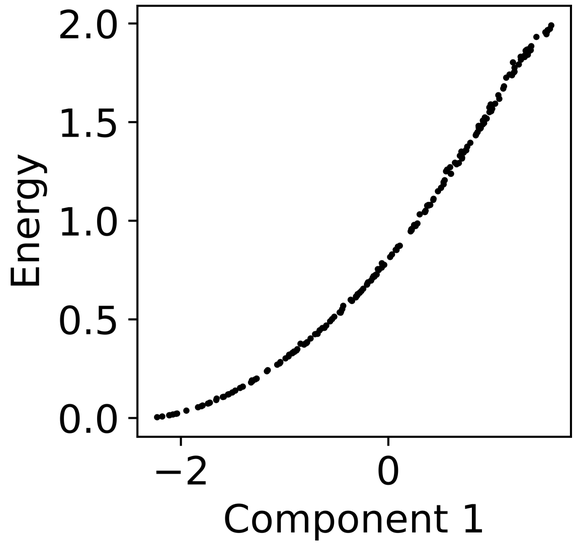
Shape Analysis: Manifold Learning


Discovering conservation laws: OT + diffusion maps





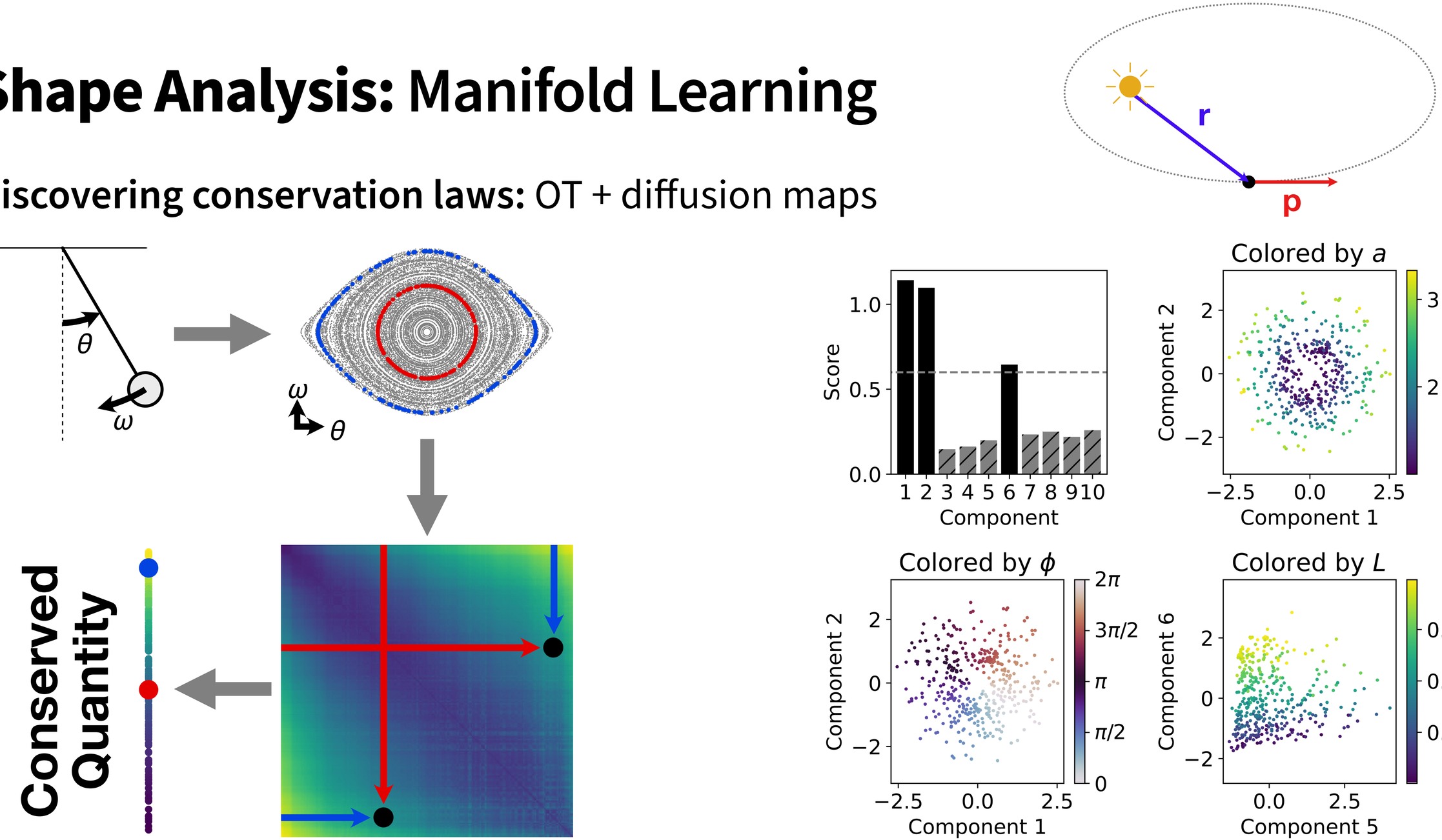




Lu et al., Nat Comm 2023

Shape Analysis: Manifold Learning

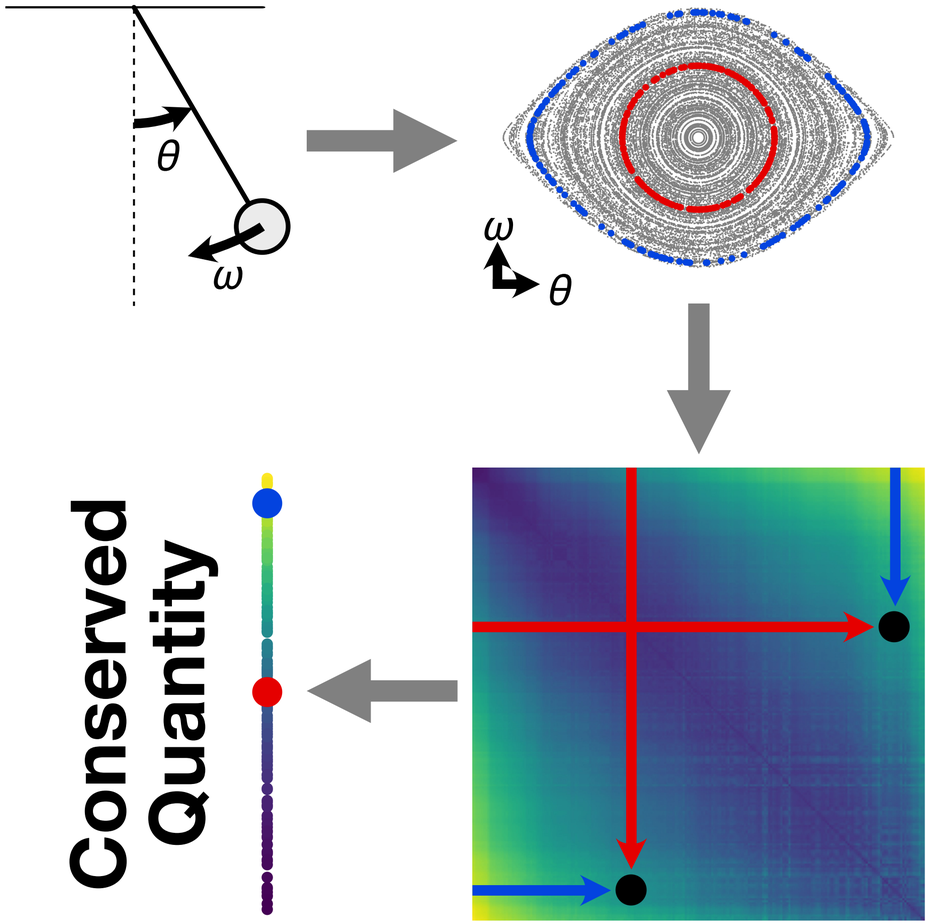
Discovering conservation laws: OT + diffusion maps

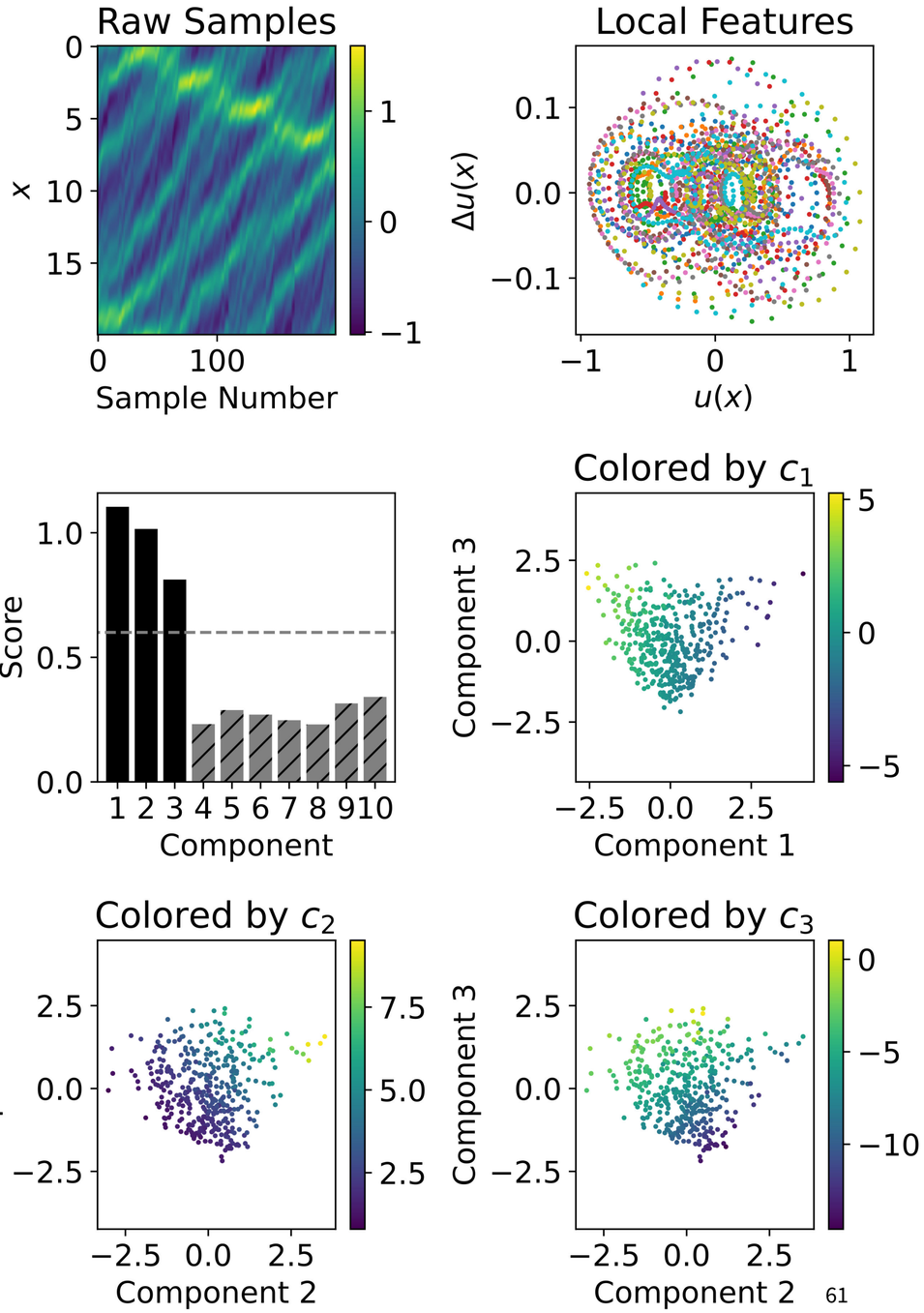


Lu et al., Nat Comm 2023

Shape Analysis: Manifold Learning

Discovering conservation laws: OT + diffusion maps

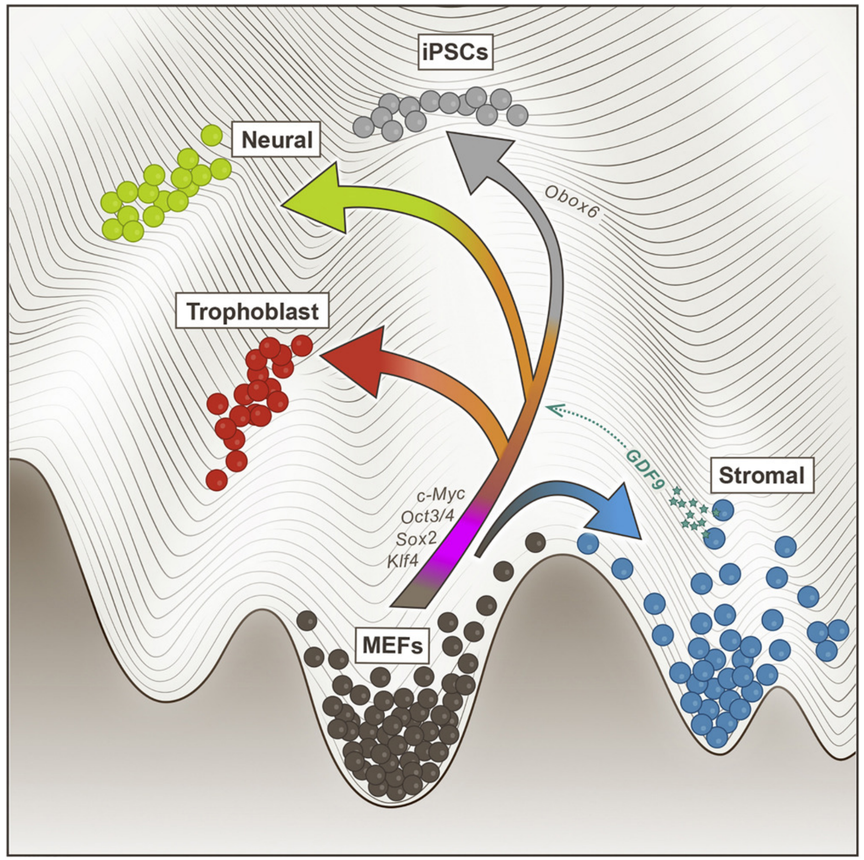




Lu et al., Nat Comm 2023

Shape Analysis: Manifold Learning

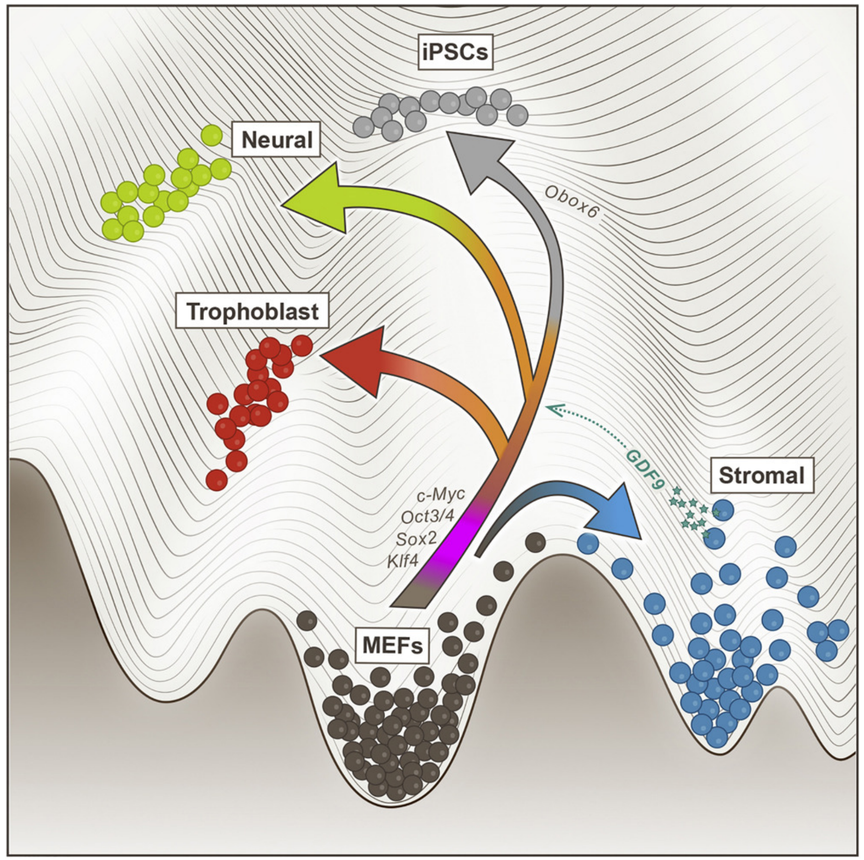
Developmental biology: inferring population dynamics



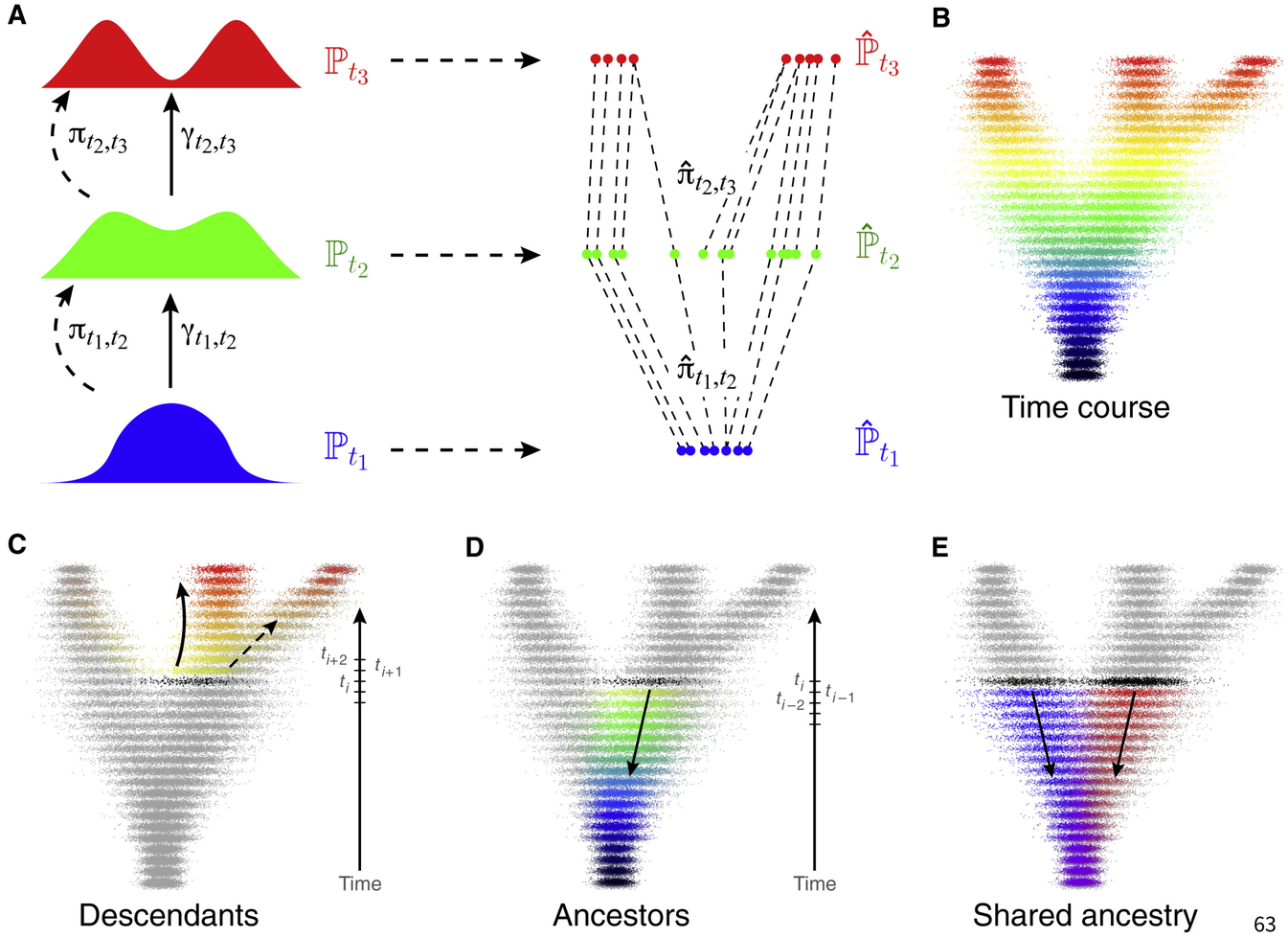
Schiebinger et al., Cell 2019

Shape Analysis: Manifold Learning

Developmental biology: inferring population dynamics

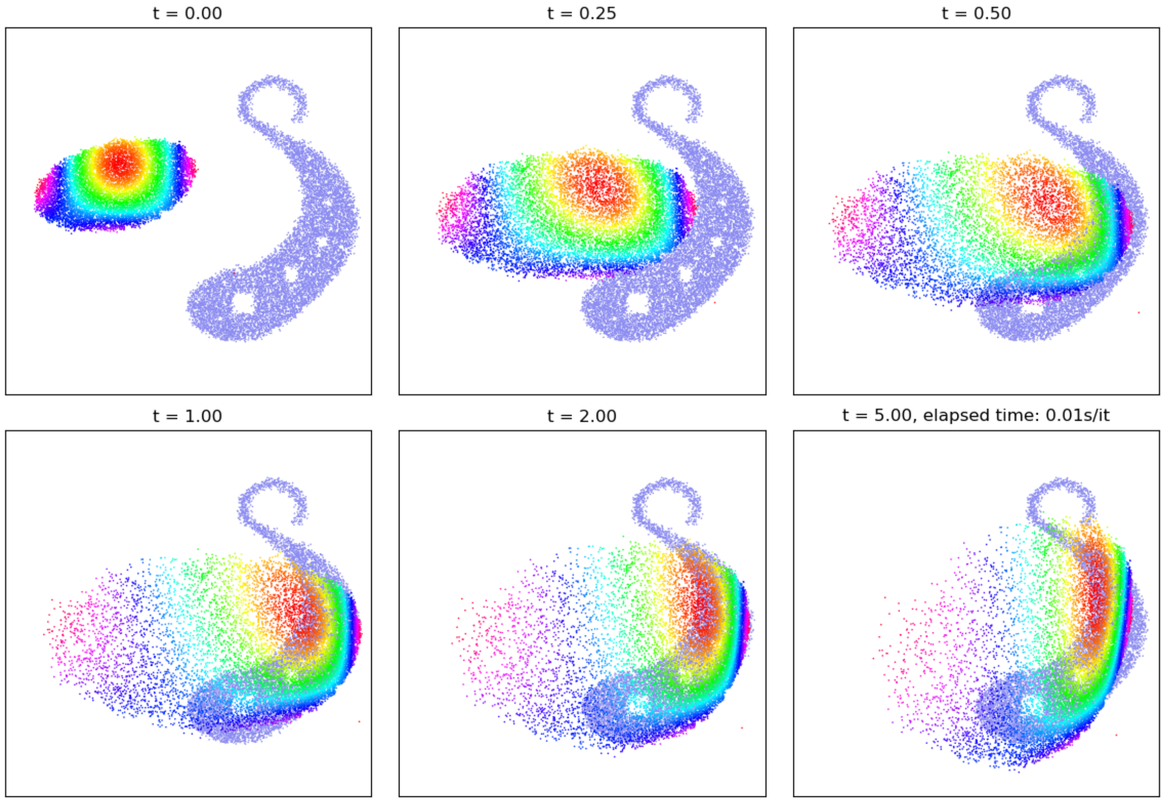


Schiebinger et al., Cell 2019

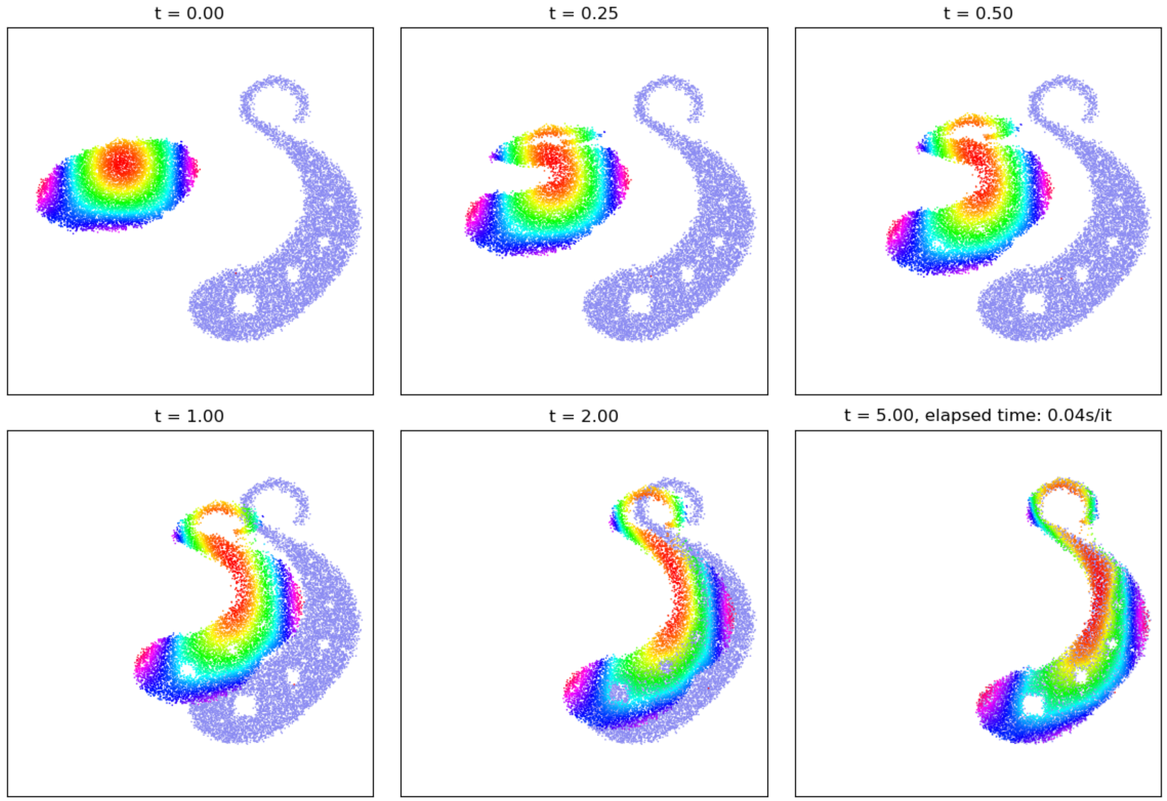


Deep Learning: Sinkhorn Loss

Gaussian MMD



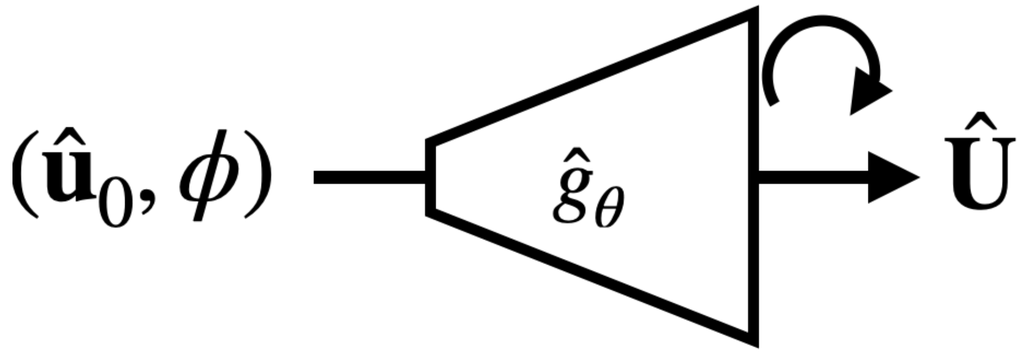
Sinkhorn



Deep Learning: Sinkhorn Loss



Training emulators for chaos: matching attractor statistics using OT

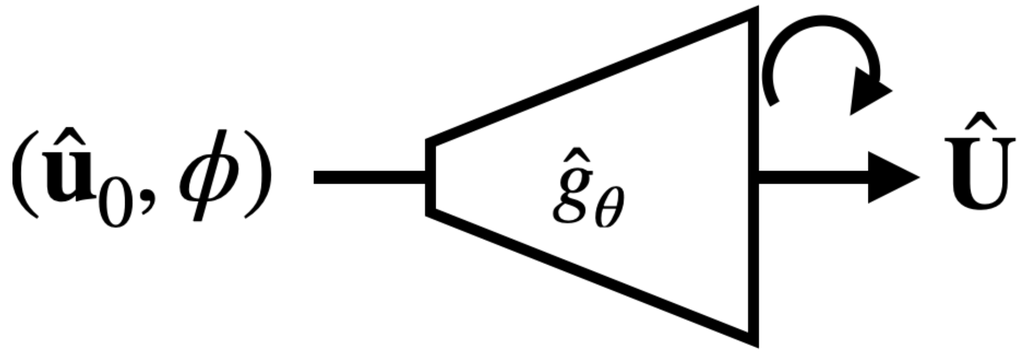


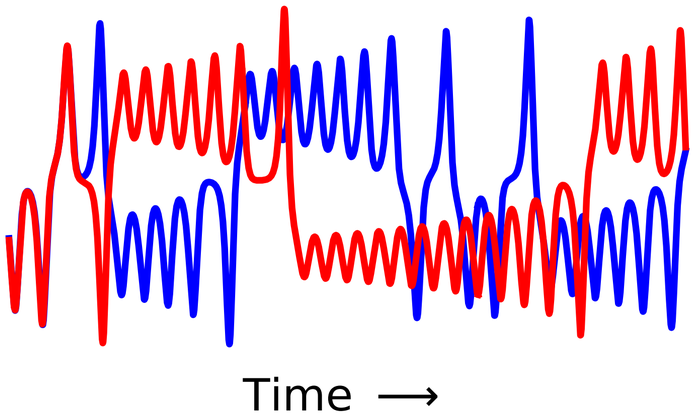
Jiang et al., NeurIPS 2023

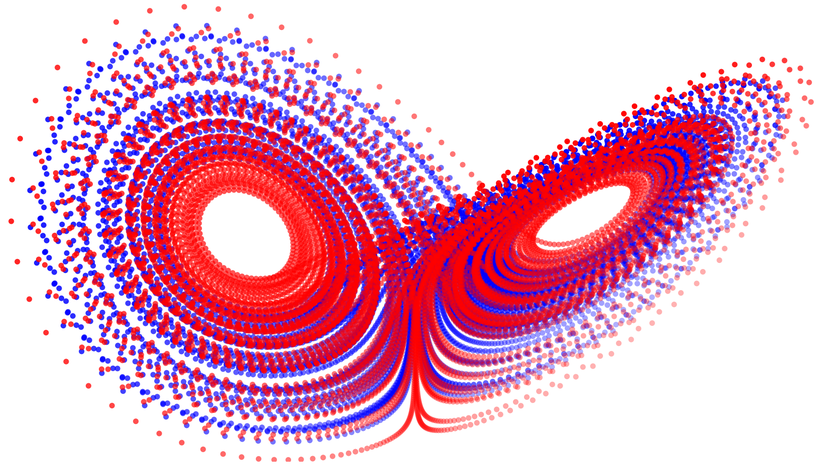
Deep Learning: Sinkhorn Loss



Training emulators for chaos: matching attractor statistics using OT





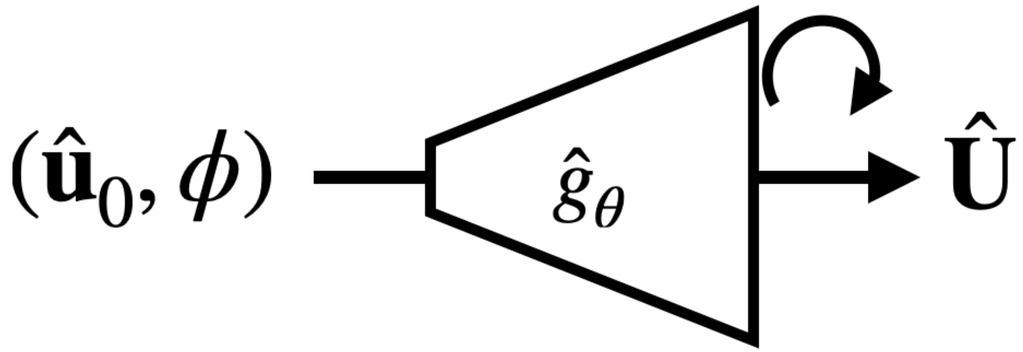


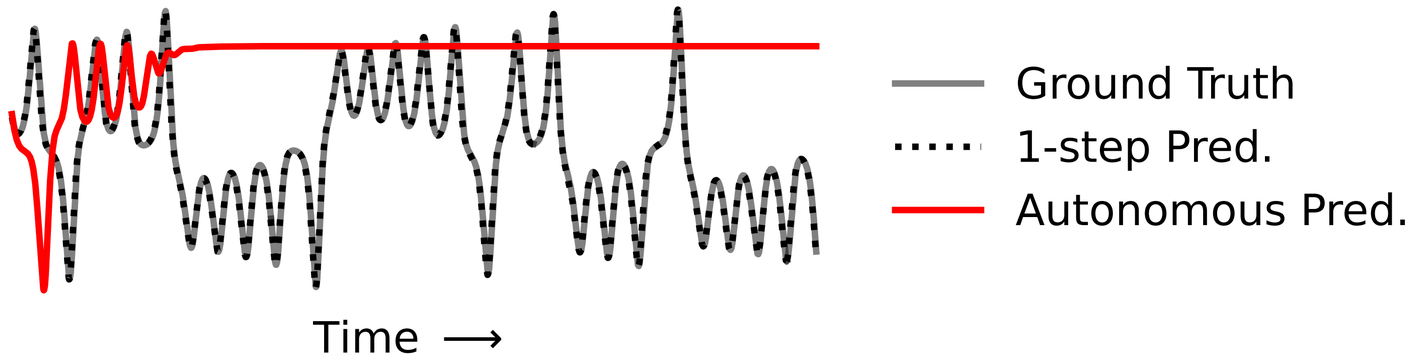
Jiang et al., NeurIPS 2023

Deep Learning: Sinkhorn Loss



Training emulators for chaos: matching attractor statistics using OT



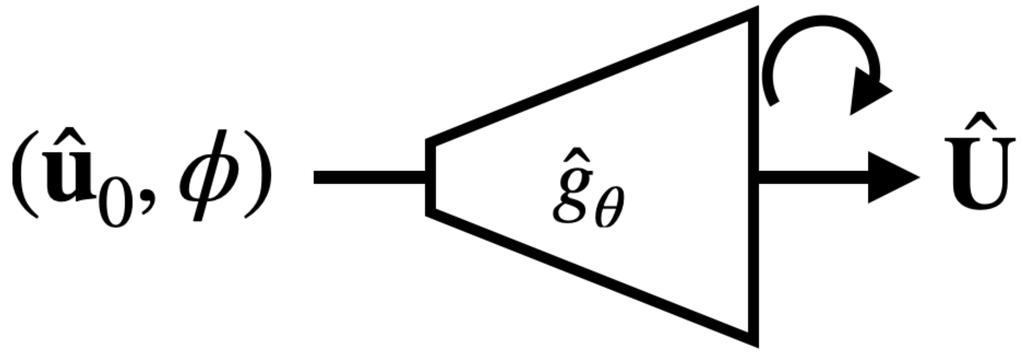


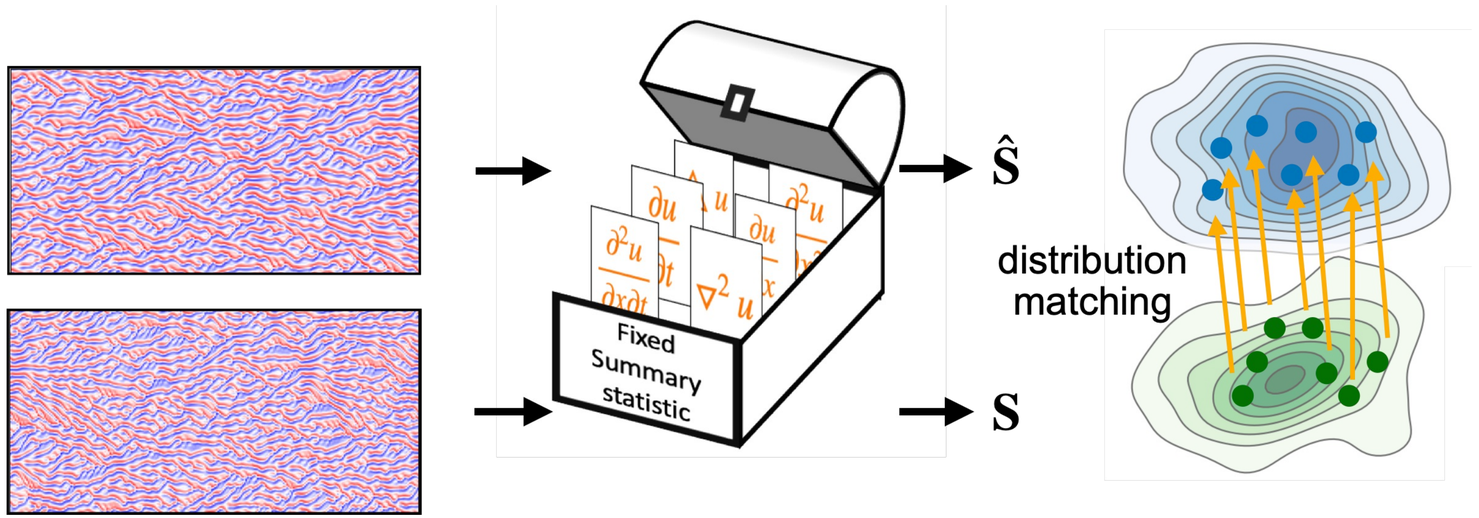
Jiang et al., NeurIPS 2023

Deep Learning: Sinkhorn Loss



Training emulators for chaos: matching attractor statistics using OT

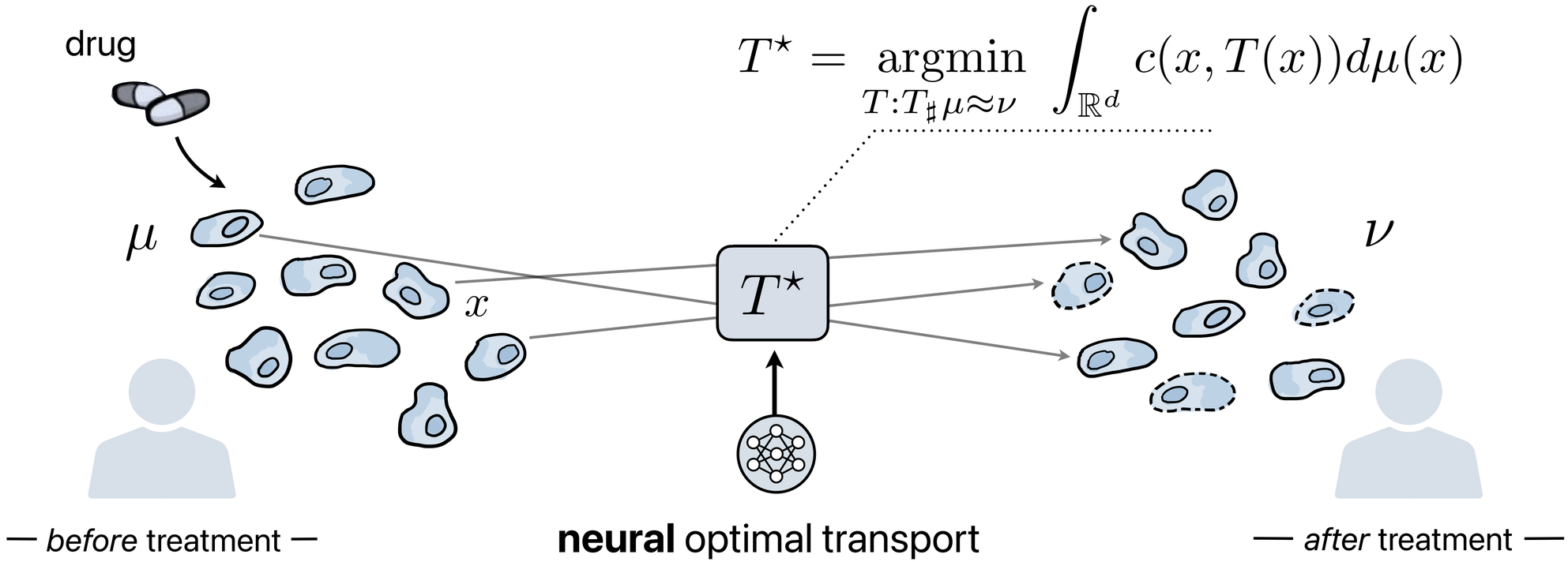




Jiang et al., NeurIPS 2023

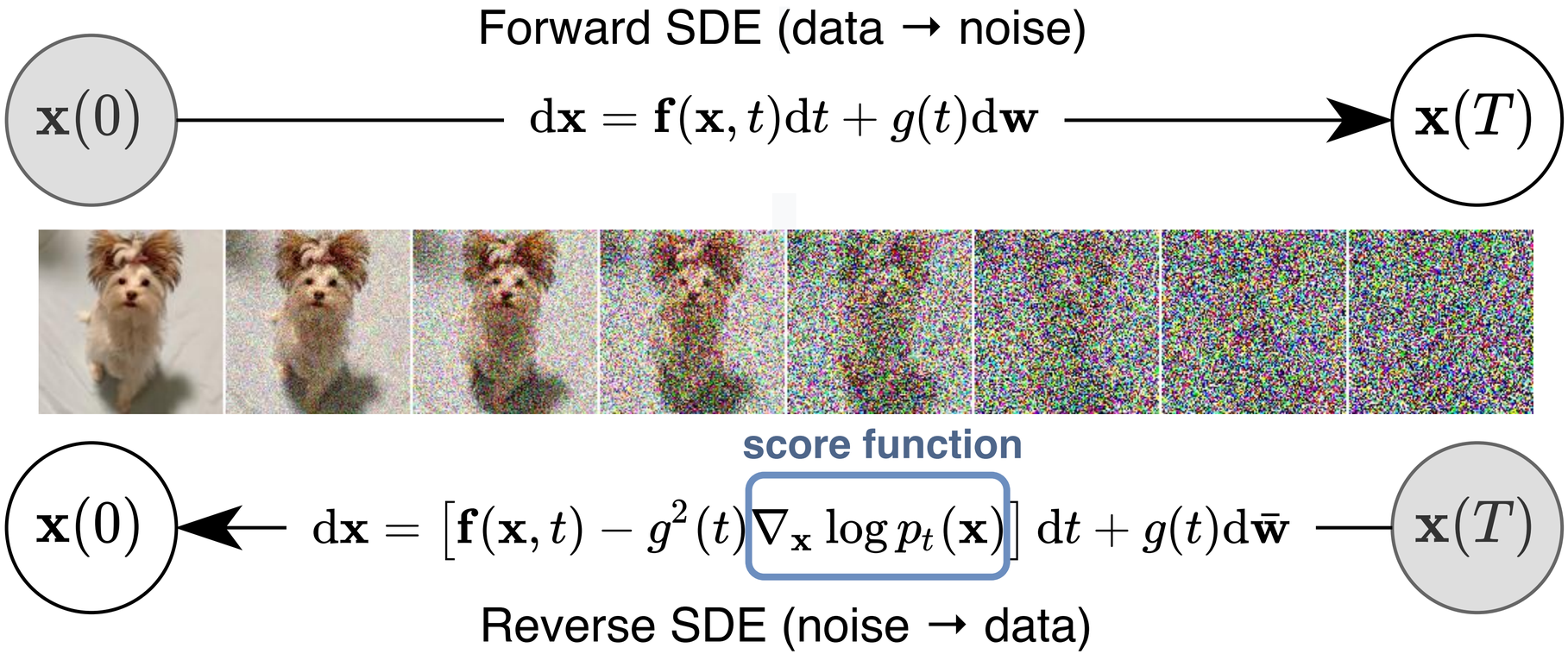
Deep Learning: Neural Optimal Transport

Personalized medicine: predict outcomes of cell populations



Bunne et al., Nat Methods 2023

Deep Learning: Generative Modeling



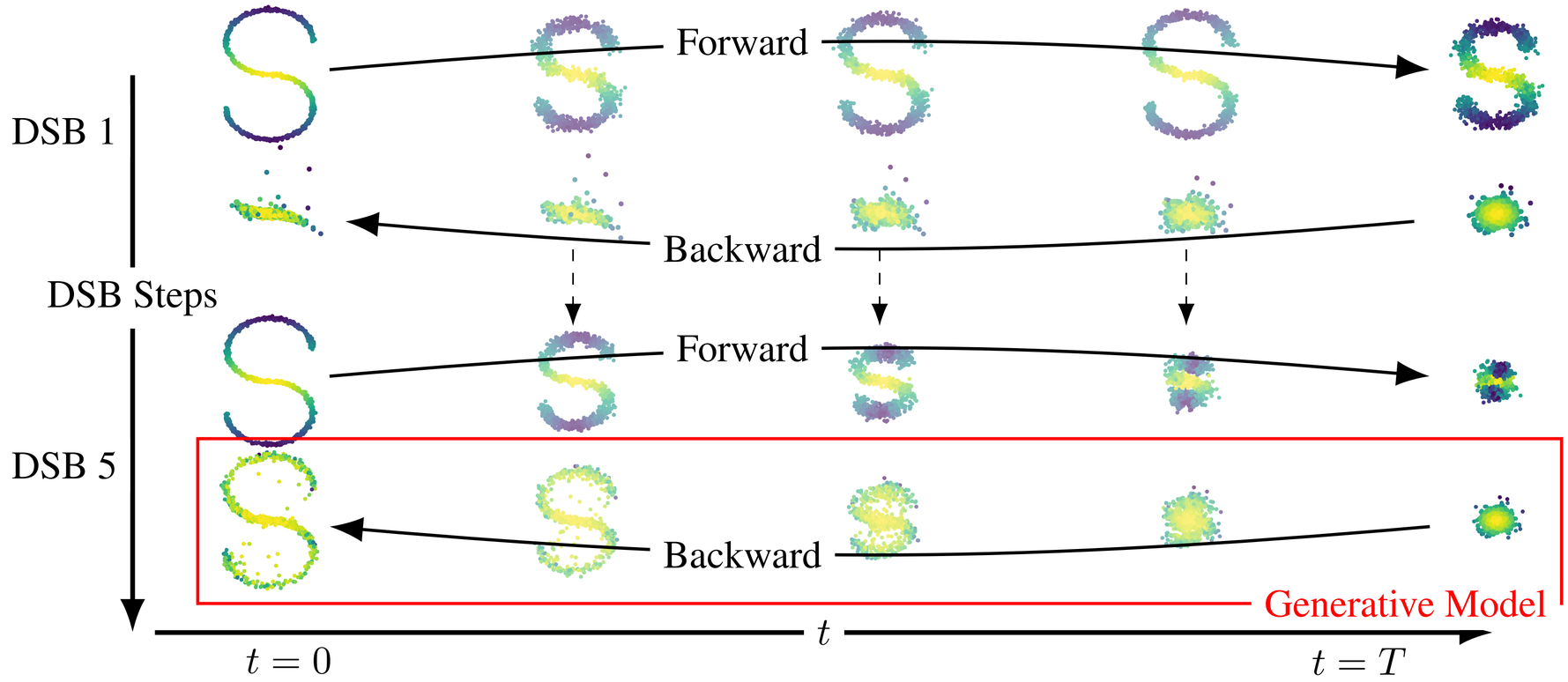
Song et al., ICLR 2021

Deep Learning: Generative Modeling

Song et al., ICLR 2021

Deep Learning: Generative Modeling

Diffusion Schrödinger bridge: speeding up diffusion models using IPF (Sinkhorn)



De Bortoli et al. NeurIPS 2021

Summary: Optimal Transport for Machine Learning

Optimal Transport Theory

- Deep mathematical foundations
- Wasserstein metric on distributions

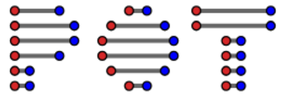
Computational Optimal Transport

- 1D OT → slice Wasserstein distance
- Entropy-Regularized OT → Sinkhorn algorithm

Machine Learning Applications

- Shape Analysis: barycenters, manifold learning, …
- Deep Learning: Sinkhorn loss, generative modeling, …

lup@uchicago.edu **peterparity.github.io**



POT: Python Optimal Transport (*NumPy*)
pythonot.github.io

OTT: Optimal Transport Tools (*JAX*)
ott-jax.readthedocs.io

GeomLoss: Geometric Loss (*PyTorch*)
kernel-operations.io/geomloss

Gabriel Peyré & Marco Cuturi
Computational Optimal Transport
optimaltransport.github.io

Charlotte Bunne & Marco Cuturi
Optimal Transport in
Learning, Control, and Dynamical Systems
bunne.ch/ot_tutorial