An Introduction to Optimal Transport for Machine Learning

Clément Bonet

Ecole Polytechnique

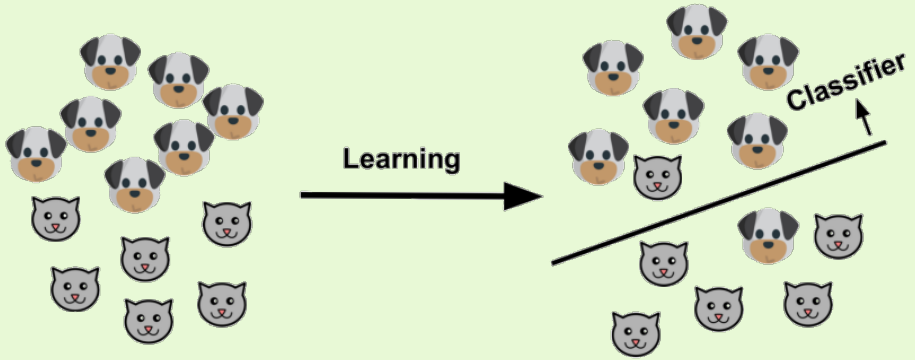
Hi! PARIS Reading groups: "OT for ML"
14/10/2025

Machine Learning

Goal: learn a model from data

Example

- Classification



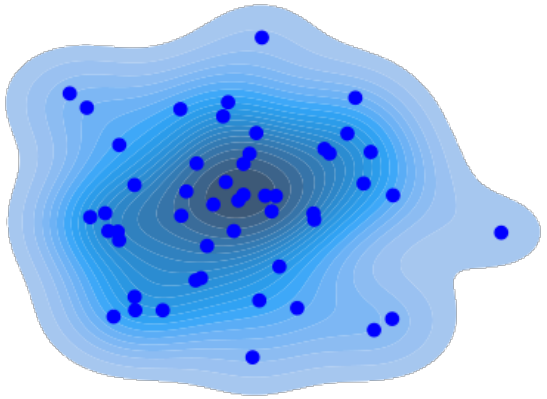
From (Goyal, 2018)

- Generative model (images, text...)

Samples from Stable Diffusion (Rombach et al., 2022)

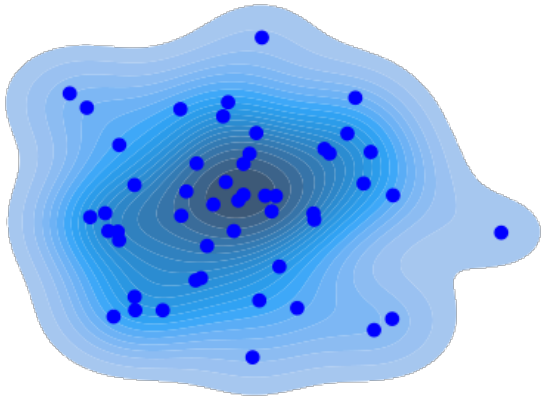
Probability Distributions and Generative Modeling

- Data: $x_1, \ldots, x_n \in \mathbb{R}^d \longleftrightarrow$ probability distribution $\mu_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}$

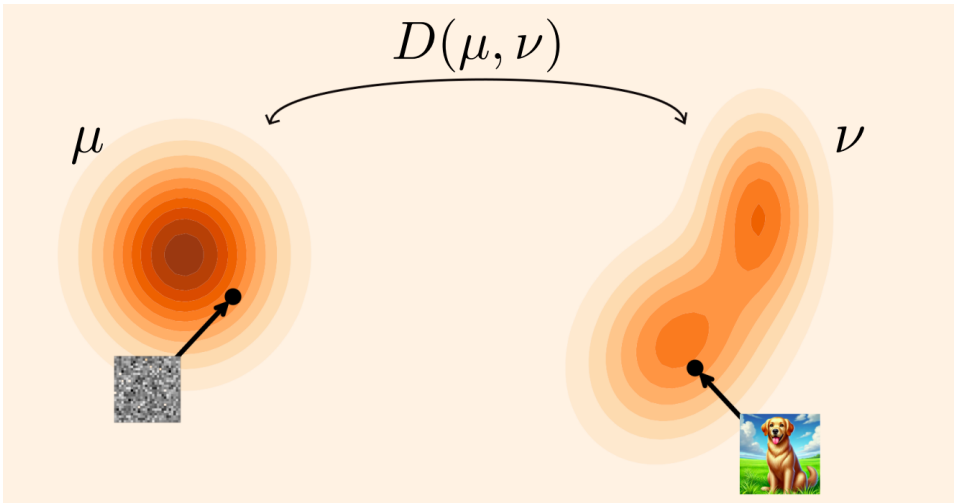


Probability Distributions and Generative Modeling

- Data: $x_1, \ldots, x_n \in \mathbb{R}^d \longleftrightarrow$ probability distribution $\mu_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}$

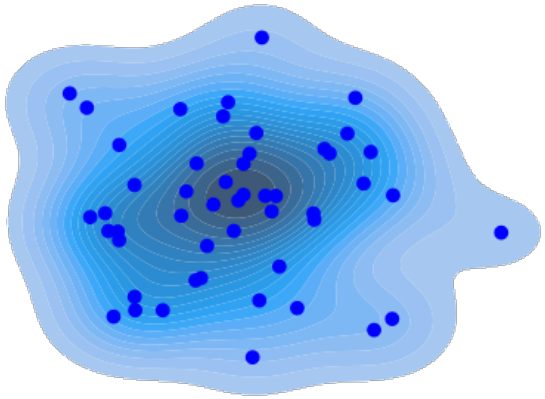


- Generative modeling:
  - Access to samples $x_1, \ldots, x_n \sim \nu$, ν unknown
  - Goal: sample from ν



Probability Distributions and Generative Modeling

- Data: $x_1, \ldots, x_n \in \mathbb{R}^d \longleftrightarrow$ probability distribution $\mu_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}$



- **Goals**:
  - Compare distributions using some discrepancy D
  - Learn/move distributions by minimizing some criterion D (*e.g.* for generative models)

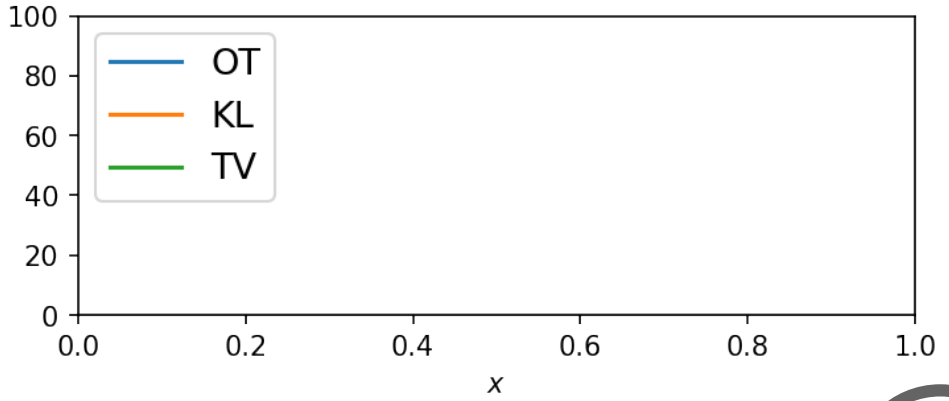
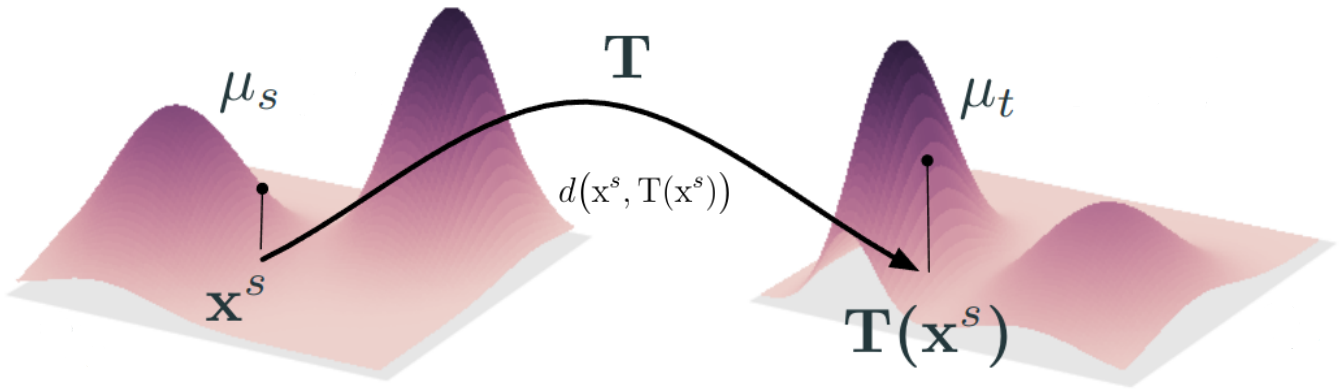


Table of Contents

Optimal Transport

Optimal Transport methods

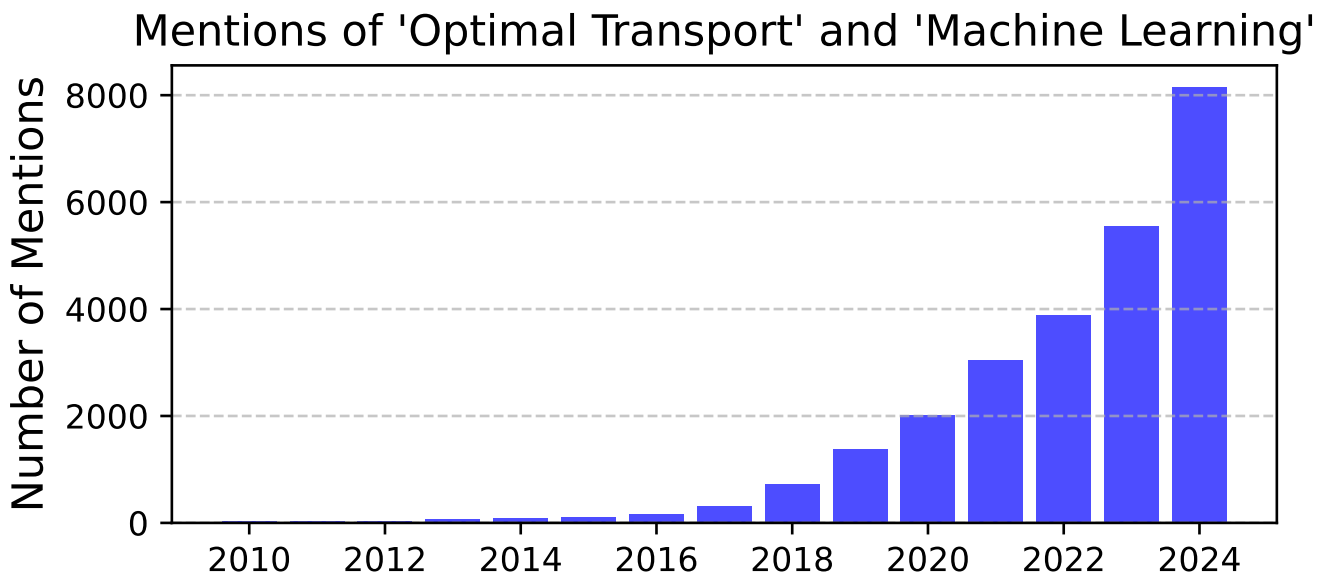
- **Compare probability distributions**
- Leverage the **geometry** of the underlying space X



Optimal Transport

Optimal Transport methods

- **Compare probability distributions**
- Leverage the **geometry** of the underlying space X
- Very **popular** in **Machine Learning** (Peyré et al., 2019)



Number of mentions of the words "Optimal Transport" and "Machine Learning" on Google Scholar

The Monge Problem (1781)

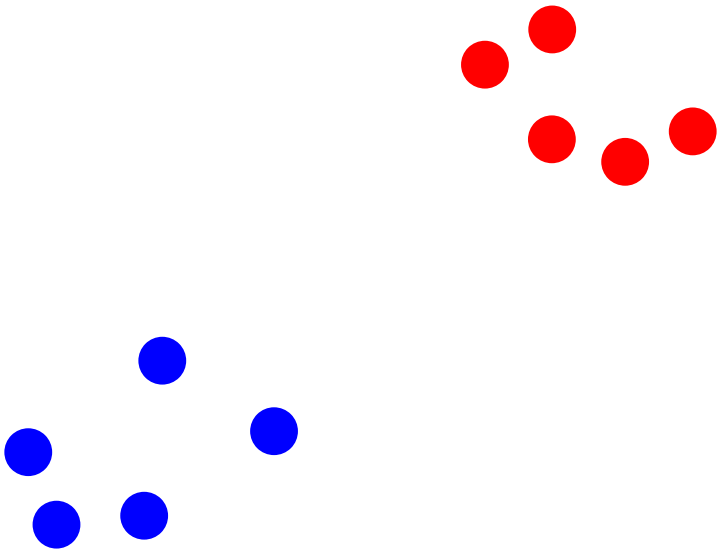
666 · MÉMOIRES DE L'ACADÉMIE ROYALE

MÉMOIRE

SUR LA

THÉORIE DES DÉBLAIS ET DES REMBLAIS.

Par M. MONGE.

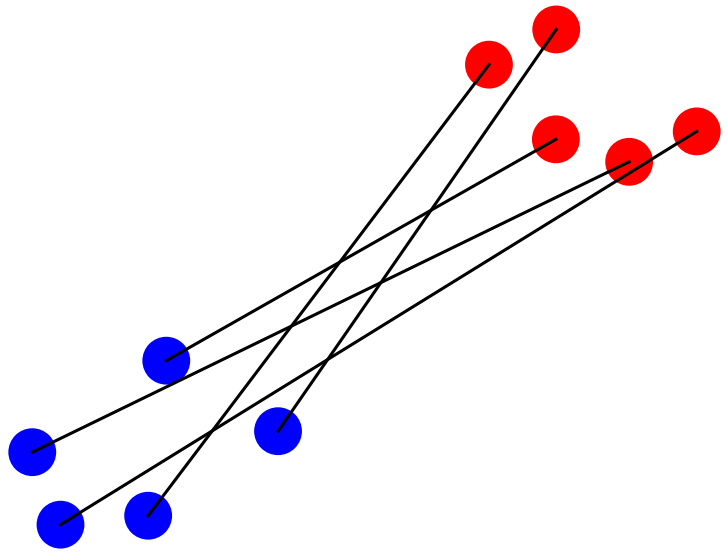


The Monge Problem (1781)

666 · MÉMOIRES DE L'ACADÉMIE ROYALE

MÉMOIRE
SUR LA
THÉORIE DES DÉBLAIS
ET DES REMBLAIS.

Par M. MONGE.

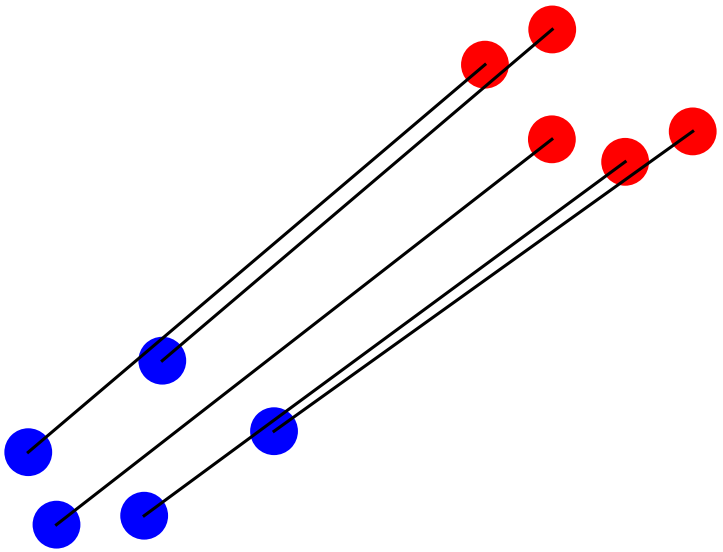


The Monge Problem (1781)

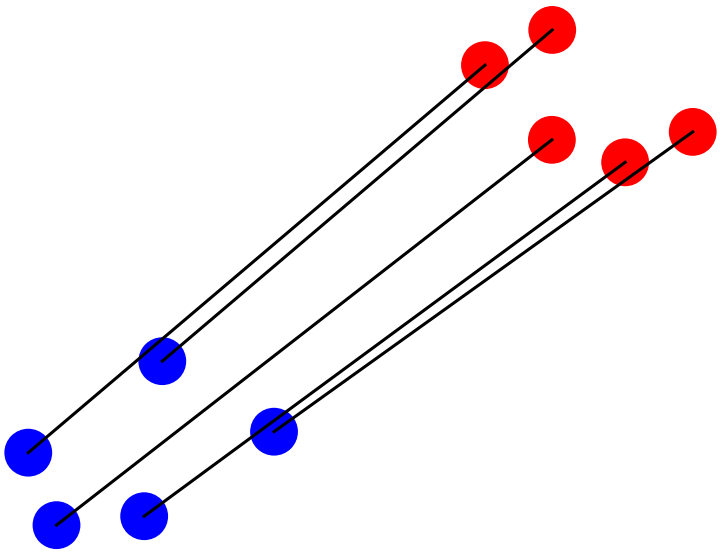
666 · MÉMOIRES DE L'ACADÉMIE ROYALE

MÉMOIRE
SUR LA
THÉORIE DES DÉBLAIS
ET DES REMBLAIS.

Par M. MONGE.



The Monge Problem (1781)



For $\mu_n = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$, $\nu_n = \frac{1}{n}\sum_{i=1}^n \delta_{y_i}$,

$$\mathrm{M}_c(\mu_n, \nu_n) = \inf_{\sigma \in S_n} \frac{1}{n}\sum_{i=1}^n c(x_i, y_{\sigma(i)})$$

$$= \inf_{\frac{1}{n}\sum_{i=1}^n \delta_{T(x_i)} = \frac{1}{n}\sum_{i=1}^n \delta_{y_i}} \frac{1}{n}\sum_{i=1}^n c\big(x_i, T(x_i)\big),$$

with $T : \{x_1, \ldots, x_n\} \to \{y_1, \ldots, y_n\}$ bijection.

The Monge Problem (1781)

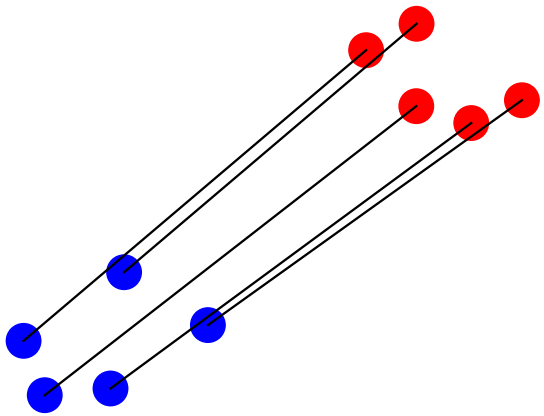
Monge Problem (Monge, 1781)

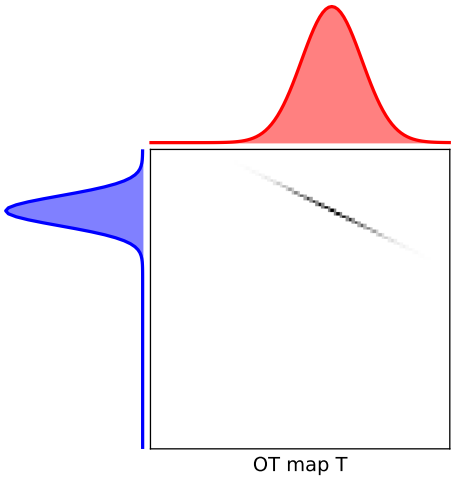
Let $\mu, \nu \in \mathcal{P}(\mathbb{R}^d)$, $c : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$,

$$\mathrm{M}_c(\mu, \nu) = \inf_{T_\# \mu = \nu} \int c\big(x, T(x)\big)\, \mathrm{d}\mu(x),$$

where $T : \mathbb{R}^d \to \mathbb{R}^d$ and $T_\# \mu = \mu \circ T^{-1}$.

If $T_\# \mu = \nu$, then $X \sim \mu \implies T(X) \sim \nu$.



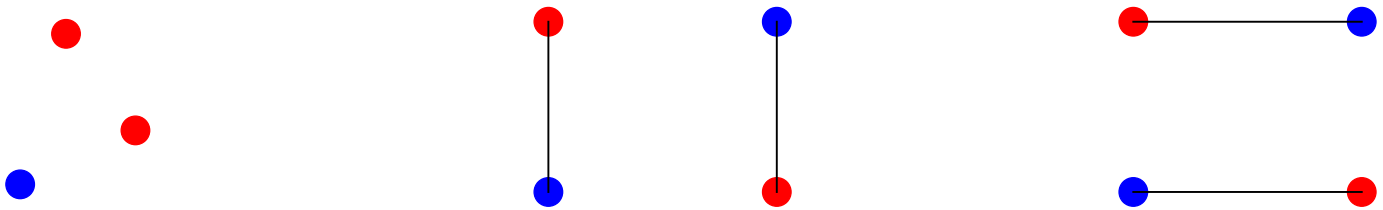


The Monge Problem (1781)

Monge problem: hard to solve

Constraints $T_{\#}\mu = \nu$ hard to satisfy:

- Might be empty, *e.g.* with $\mu = \delta_x$, $\nu = \frac{1}{2}\delta_{y_1} + \frac{1}{2}\delta_{y_2}$
- Or optimal T might be not unique

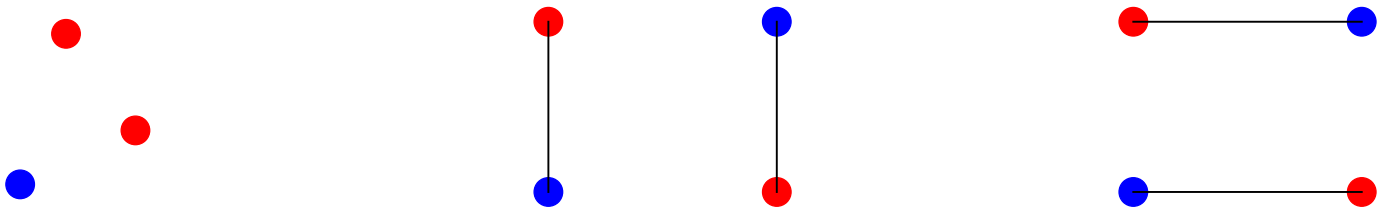


The Monge Problem (1781)

Monge problem: hard to solve

Constraints $T_{\#}\mu = \nu$ hard to satisfy:

- Might be empty, *e.g.* with $\mu = \delta_x$, $\nu = \frac{1}{2}\delta_{y_1} + \frac{1}{2}\delta_{y_2}$
- Or optimal T might be not unique



$\rightarrow$ Kantorovich problem

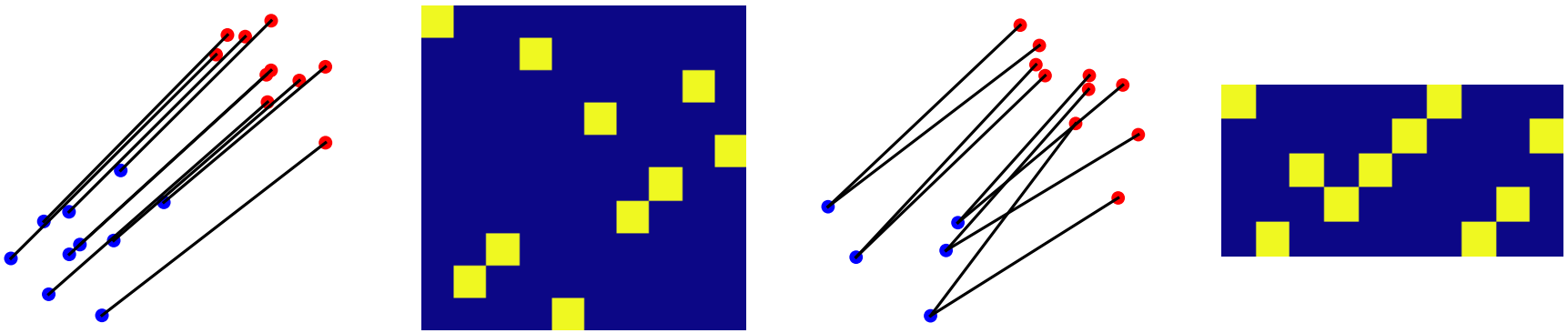
The Kantorovich Problem

Kantorovich Problem

Let $\mu, \nu \in \mathcal{P}_2(\mathbb{R}^d)$, $c : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$,

$$\mathrm{OT}_c(\mu, \nu) = \inf_{\gamma \in \Pi(\mu,\nu)} \int c(x, y) \, \mathrm{d}\gamma(x, y),$$

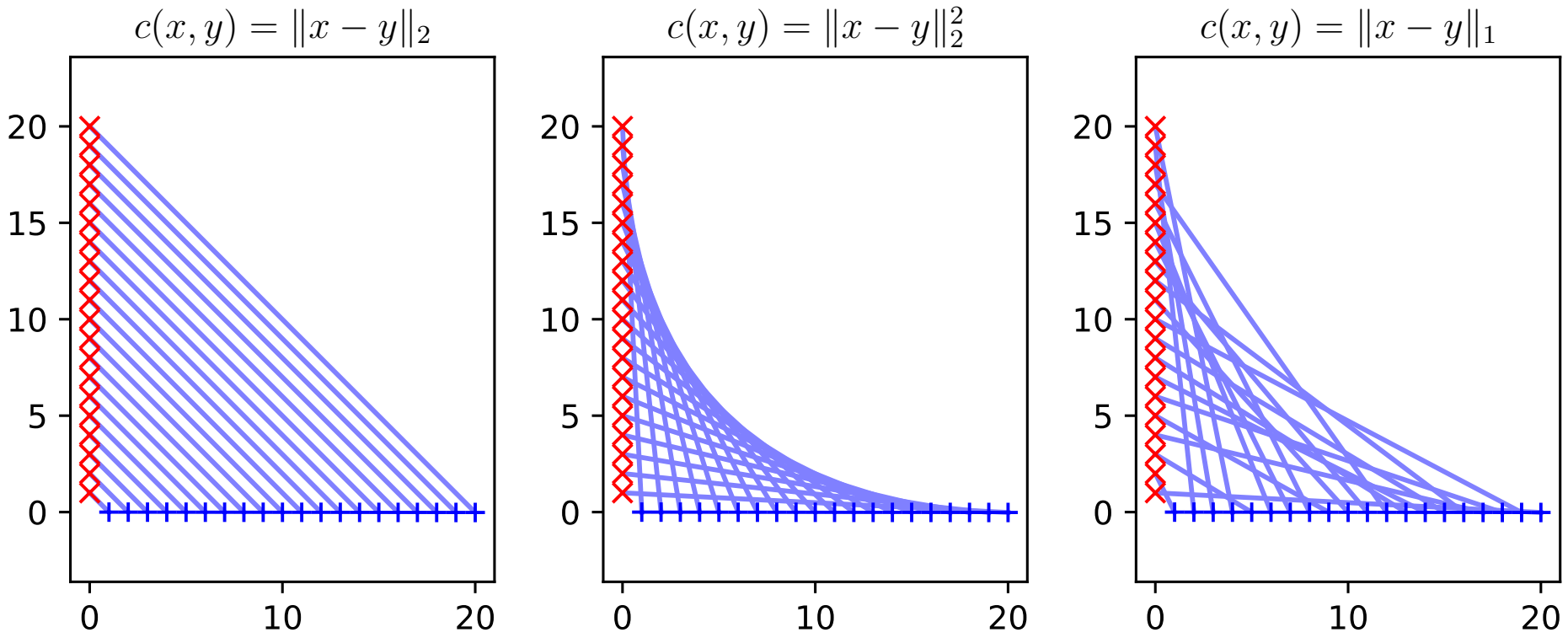
$$\Pi(\mu, \nu) = \left\{\gamma \in \mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d),\ \forall A \in \mathcal{B}(\mathbb{R}^d),\ \gamma(A \times \mathbb{R}^d) = \mu(A),\ \gamma(\mathbb{R}^d \times A) = \nu(A)\right\}$$



- Is always well defined (as $\mu \otimes \nu \in \Pi(\mu, \nu)$)
- $\gamma = (\mathrm{Id}, T)_{\#}\mu \in \Pi(\mu, \nu) \implies \mathrm{OT}_c(\mu, \nu) \leq \mathrm{M}_c(\mu, \nu)$
- $\mathrm{OT}_c(\mu, \nu) = \mathrm{M}_c(\mu, \nu)$ whenever OT maps exist (*e.g.* between $\mu_n = \nu_n$)

Influence of the cost

Different costs $\Longrightarrow$ different optimal couplings



From POT

The Wasserstein Distance

Wasserstein Distance

Let $\mu, \nu \in \mathcal{P}_2(\mathbb{R}^d)$, $c(x, y) = \|x - y\|_2^2$ for all $x, y \in \mathbb{R}^d$,

$$\mathrm{W}_2^2(\mu, \nu) = \inf_{\gamma \in \Pi(\mu, \nu)} \int \|x - y\|_2^2 \, \mathrm{d}\gamma(x, y)$$

Properties:

- W_2 distance
- $\mathrm{W}_2(\delta_x, \delta_y) = \|x - y\|_2$
- $(\mathcal{P}_2(\mathbb{R}^d), \mathrm{W}_2)$ has a Riemannian structure

Condition to have a deterministic coupling, *i.e.* $\gamma = (\mathrm{Id}, T)_\# \mu$ with $T_\# \mu = \nu$ where $\forall A \in \mathcal{B}(\mathbb{R}^d)$, $T_\# \mu(A) = \mu\big(T^{-1}(A)\big)$: **Brenier's theorem** (Brenier, 1991)

$\mu \ll \mathrm{Leb} \implies$ Optimal coupling γ^* unique and $\gamma^* = (\mathrm{Id}, \nabla\varphi)_\# \mu$ with φ convex

Table of Contents

Solving the OT Problem

Let $x_1, \dots, x_n, y_1, \dots, y_n \in \mathbb{R}^d$, $\alpha, \beta \in \Sigma_n$, $\mu = \sum_{i=1}^n \alpha_i \delta_{x_i}$, $\nu = \sum_{i=1}^n \beta_i \delta_{y_i}$,

$$\mathrm{W}_2^2(\mu, \nu) = \min_{P \in \mathbb{R}_+^{n \times n},\ P\mathbb{1}_n = \alpha,\ P^T \mathbb{1}_n = \beta} \langle C, P \rangle_F \quad \text{with} \quad C = \left(\|x_i - y_j\|_2^2 \right)_{i,j}$$

Solving the OT Problem

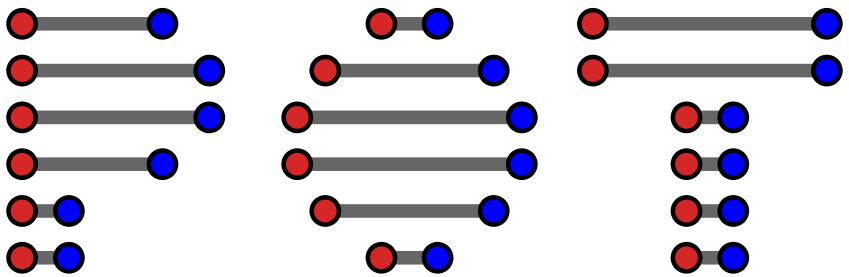
Let $x_1, \dots, x_n, y_1, \dots, y_n \in \mathbb{R}^d$, $\alpha, \beta \in \Sigma_n$, $\mu = \sum_{i=1}^n \alpha_i \delta_{x_i}$, $\nu = \sum_{i=1}^n \beta_i \delta_{y_i}$,

$$\mathrm{W}_2^2(\mu, \nu) = \min_{P \in \mathbb{R}_+^{n \times n},\ P\mathbb{1}_n = \alpha,\ P^T\mathbb{1}_n = \beta} \langle C, P \rangle_F \quad \text{with} \quad C = \left(\|x_i - y_j\|_2^2\right)_{i,j}$$

Computational Complexity (Pele and Werman, 2009)

Numerical computation: **Linear program** in $O(n^3 \log n)$

Implemented efficiently *e.g.* in POT (Flamary et al., 2021) but still costly



Solving the OT Problem

Let $x_1, \dots, x_n, y_1, \dots, y_n \in \mathbb{R}^d$, $\alpha, \beta \in \Sigma_n$, $\mu = \sum_{i=1}^n \alpha_i \delta_{x_i}$, $\nu = \sum_{i=1}^n \beta_i \delta_{y_i}$,

$$\mathrm{W}_2^2(\mu, \nu) = \min_{P \in \mathbb{R}_+^{n \times n},\ P\mathbb{1}_n = \alpha,\ P^T\mathbb{1}_n = \beta} \langle C, P \rangle_F \quad \text{with} \quad C = \left(\|x_i - y_j\|_2^2\right)_{i,j}$$

Computational Complexity (Pele and Werman, 2009)

Numerical computation: **Linear program** in $O(n^3 \log n)$

Sample Complexity (Boissard and Le Gouic, 2014)

For $\mu, \nu \in \mathcal{P}_2(\mathbb{R}^d)$, $x_1, \dots, x_n \sim \mu$, $y_1, \dots, y_n \sim \nu$, $\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$ and $\hat{\nu}_n = \frac{1}{n}\sum_{i=1}^n \delta_{y_i}$,

$$\mathbb{E}\left[\left|\mathrm{W}_2(\hat{\mu}_n, \hat{\nu}_n) - \mathrm{W}_2(\mu, \nu)\right|\right] = O(n^{-1/d})$$

$\rightarrow$ **curse of dimensionality**

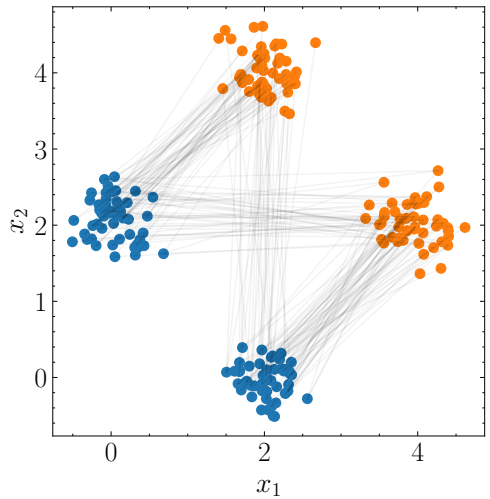
Minibatch OT (Fatras et al., 2021)

$$\mathrm{MBOT}_K(\mu,\nu) = \mathbb{E}_{X\sim\mu^{\otimes K},Y\sim\nu^{\otimes K}}\left[\mathrm{OT}_c\left(\frac{1}{K}\sum_{k=1}^{K}\delta_{x_k},\frac{1}{K}\sum_{k=1}^{K}\delta_{y_k}\right)\right]$$

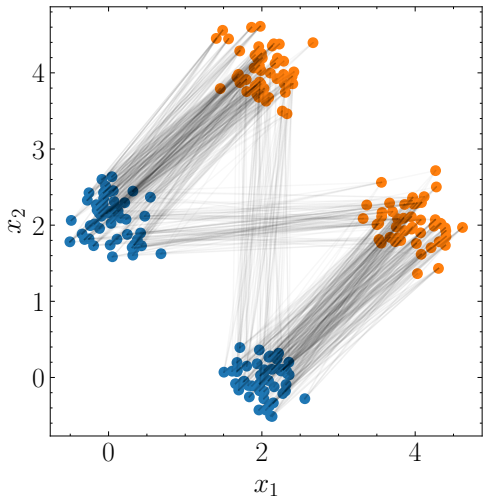
$$\approx \frac{1}{L}\sum_{\ell=1}^{L}\mathrm{OT}_c\left(\frac{1}{K}\sum_{k=1}^{K}\delta_{x_k^{(\ell)}},\frac{1}{K}\sum_{k=1}^{K}\delta_{y_k^{\ell}}\right),$$

where $X=(x_1,\ldots,x_K)$, $Y=(y_1,\ldots,y_K)$.

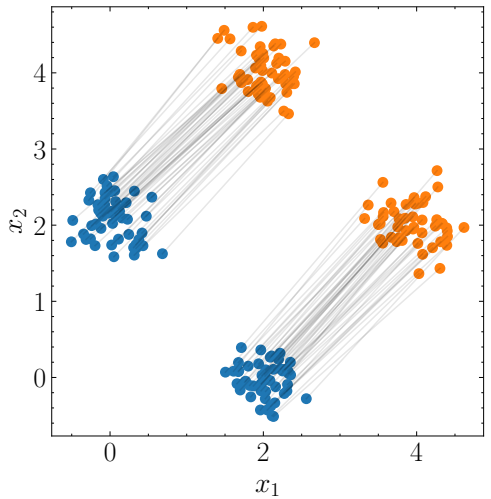
Properties: Computational complexity in $O(LK^3\log K)$, OT plans less sparse



$K=2$



$K=10$



$K=n=100$

Figure from (Montesuma et al., 2024)

Entropic Regularization (Cuturi, 2013)

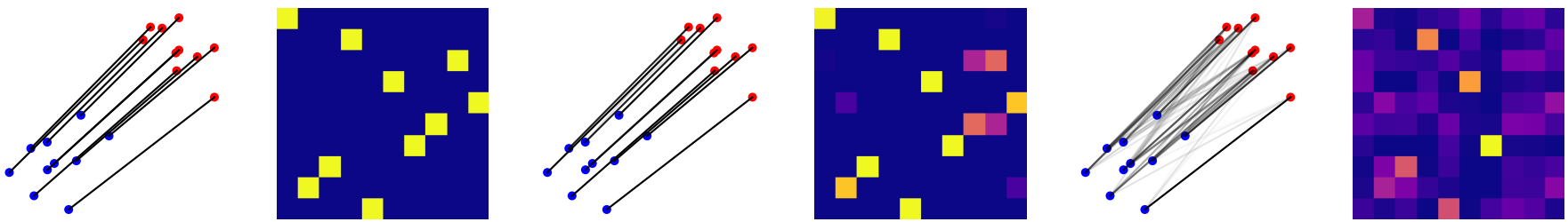
Let $\mu, \nu \in \mathcal{P}_2(\mathbb{R}^d)$,

$$\mathrm{OT}_{c,\varepsilon}(\mu,\nu) = \inf_{\gamma \in \Pi(\mu,\nu)} \int c(x,y)\ \mathrm{d}\gamma(x,y) + \varepsilon \mathcal{H}(\gamma)$$

where $\mathcal{H}(\gamma) = \int \log\big(\gamma(x,y)\big)\ \mathrm{d}\gamma(x,y)$ is the negative entropy.

Properties:

- Smooth and strictly convex problem $\rightarrow$ **unique solution**
- Solved in $O(n^2 \log n/\varepsilon)$ with the **Sinkhorn** algorithm
- Converge to $\mathrm{OT}_c(\mu,\nu)$ as $\varepsilon \rightarrow 0$
- Lose sparsity of the coupling



OT_c $\quad$ $\mathrm{OT}_{c,\varepsilon}$, $\varepsilon = 0.1$ $\quad$ $\mathrm{OT}_{c,\varepsilon}$ $\varepsilon = 1$

1D OT Problem

Let $\mu, \nu \in \mathcal{P}_2(\mathbb{R})$,

- Cumulative distribution function:

$$\forall t \in \mathbb{R},\ F_\mu(t) = \mu(]-\infty, t]) = \int \mathbb{1}_{]-\infty,t]}(x)\ \mathrm{d}\mu(x)$$

- Quantile function:

$$\forall u \in [0,1],\ F_\mu^{-1}(u) = \inf\left\{x \in \mathbb{R},\ F_\mu(x) \geq u\right\}$$

1D Wasserstein Distance

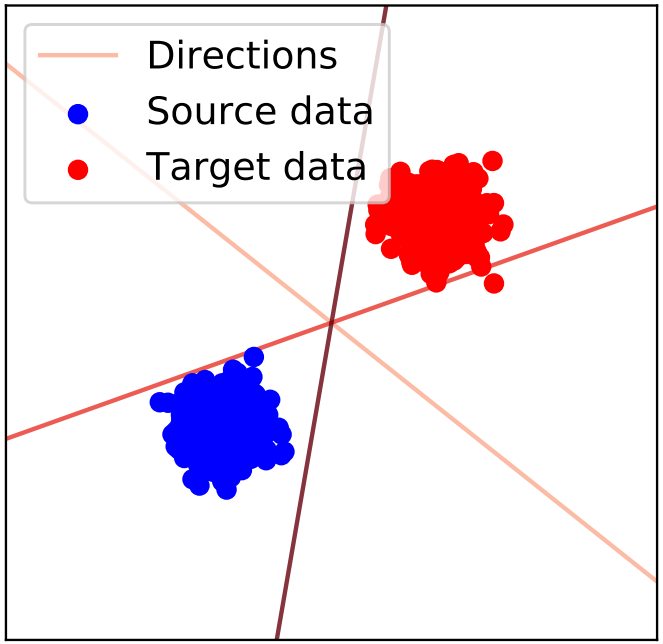
$$\mathrm{W}_2^2(\mu,\nu) = \int_0^1 \left|F_\mu^{-1}(u) - F_\nu^{-1}(u)\right|^2 \mathrm{d}u = \left\|F_\mu^{-1} - F_\nu^{-1}\right\|^2_{L^2([0,1])}$$

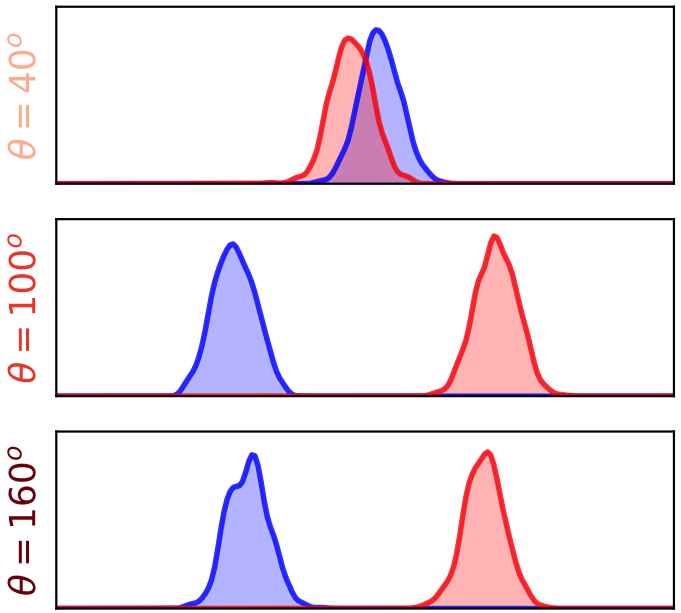
Let $x_1 < \cdots < x_n,\ y_1 < \cdots < y_n$, $\mu = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$, $\nu = \frac{1}{n}\sum_{i=1}^n \delta_{y_i}$,

$$\mathrm{W}_2^2(\mu,\nu) = \frac{1}{n}\sum_{i=1}^n (x_i - y_i)^2$$

$\rightarrow O(n \log n)$

Sliced-Wasserstein Distance





Definition (Sliced-Wasserstein (Rabin et al., 2011))

Let $\mu, \nu \in \mathcal{P}_2(\mathbb{R}^d)$,

$$\mathrm{SW}_2^2(\mu, \nu) = \int_{S^{d-1}} \mathrm{W}_2^2(P^\theta_\# \mu, P^\theta_\# \nu) \, \mathrm{d}\lambda(\theta),$$

where $P^\theta(x) = \langle x, \theta \rangle$, λ uniform measure on S^{d-1}.

Properties of the Sliced-Wasserstein Distance

Let $x_1, \dots, x_n, y_1, \dots, y_n \in \mathbb{R}^d$, $\alpha, \beta \in \Sigma_n$, $\mu = \sum_{i=1}^n \alpha_i \delta_{x_i}$, $\nu = \sum_{i=1}^n \beta_i \delta_{y_i}$.

Approximation via Monte-Carlo:

$$\widehat{\mathrm{SW}}_{2,L}^2(\mu, \nu) = \frac{1}{L} \sum_{\ell=1}^{L} \mathrm{W}_2^2(P_{\#}^{\theta_\ell} \mu, P_{\#}^{\theta_\ell} \nu),$$

$\theta_1, \dots, \theta_L \sim \lambda$.

Properties:

- Computational complexity: $O(Ln \log n + Lnd)$
- Sample complexity: independent of the dimension (Nadjahi et al., 2020)
- SW_2 distance (Bonnotte, 2013)
- Topologically equivalent to the Wasserstein distance (Nadjahi et al., 2019), *i.e.* $\lim_{n \to \infty} \mathrm{SW}_2^2(\mu_n, \mu) = 0 \iff \lim_{n \to \infty} \mathrm{W}_2^2(\mu_n, \mu) = 0.$

Summary

- Couplings between measures
- Distance
- Takes into account the geometry of the underlying space

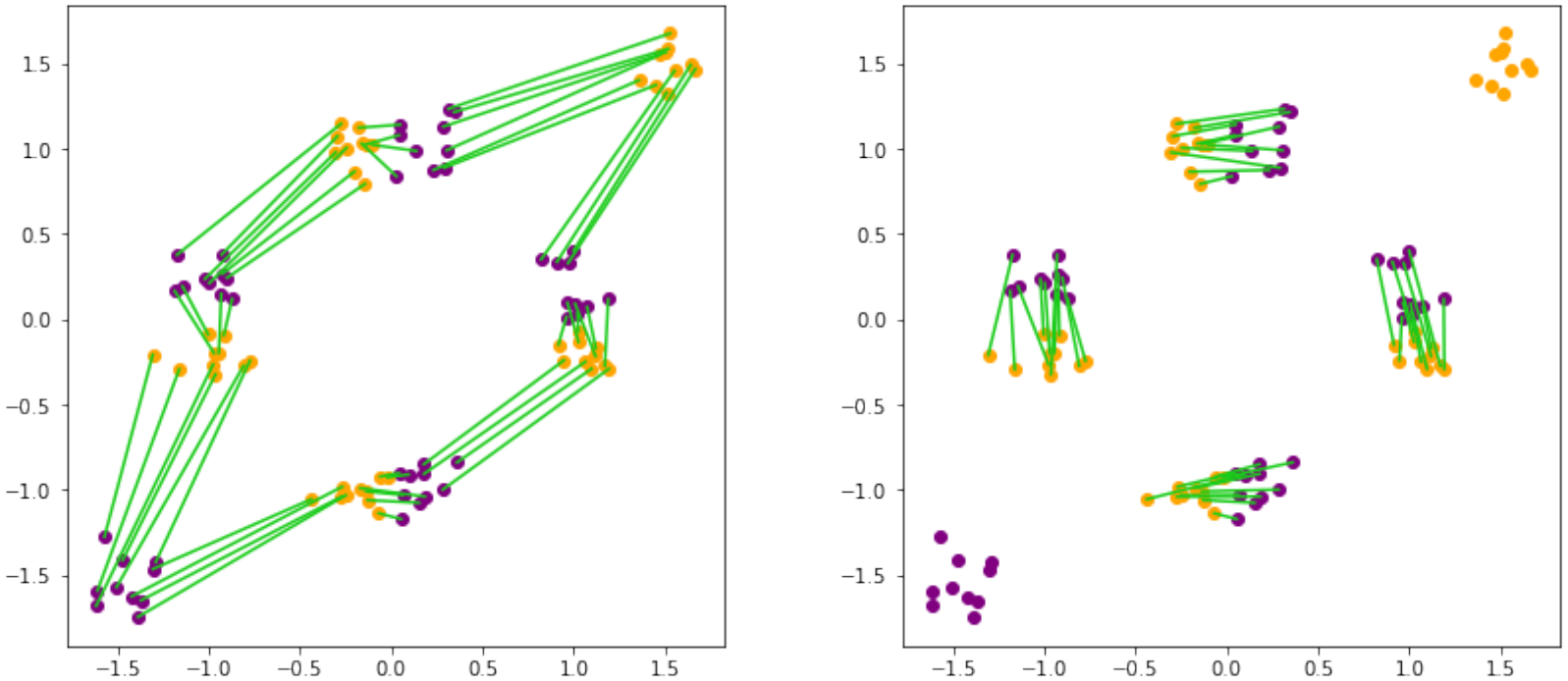
- Costly to compute
- Curse of dimensionality
- Alternatives/approximations come with pro and cons

$\rightarrow$ many variants to alleviate the cons

Extensions

Problems not solved by the OT problem in its original formulation:

- Outliers $\rightarrow$ Unbalanced and Partial OT (Séjourné et al., 2023)

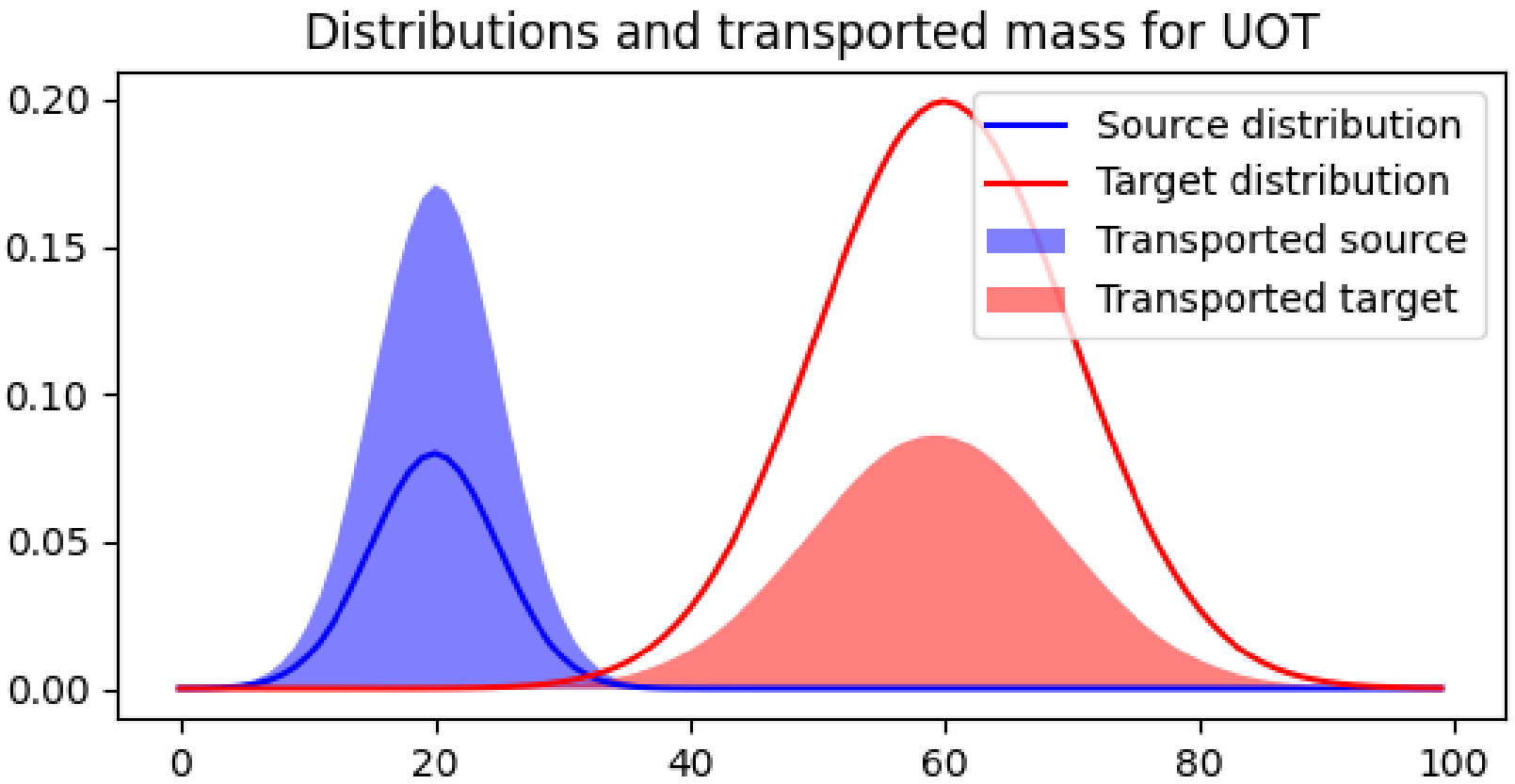


From (Séjourné et al., 2023)

Extensions

Problems not solved by the OT problem in its original formulation:

- Outliers $\rightarrow$ Unbalanced and Partial OT (Séjourné et al., 2023)
- Comparing positive measures $\rightarrow$ Unbalanced and Partial OT

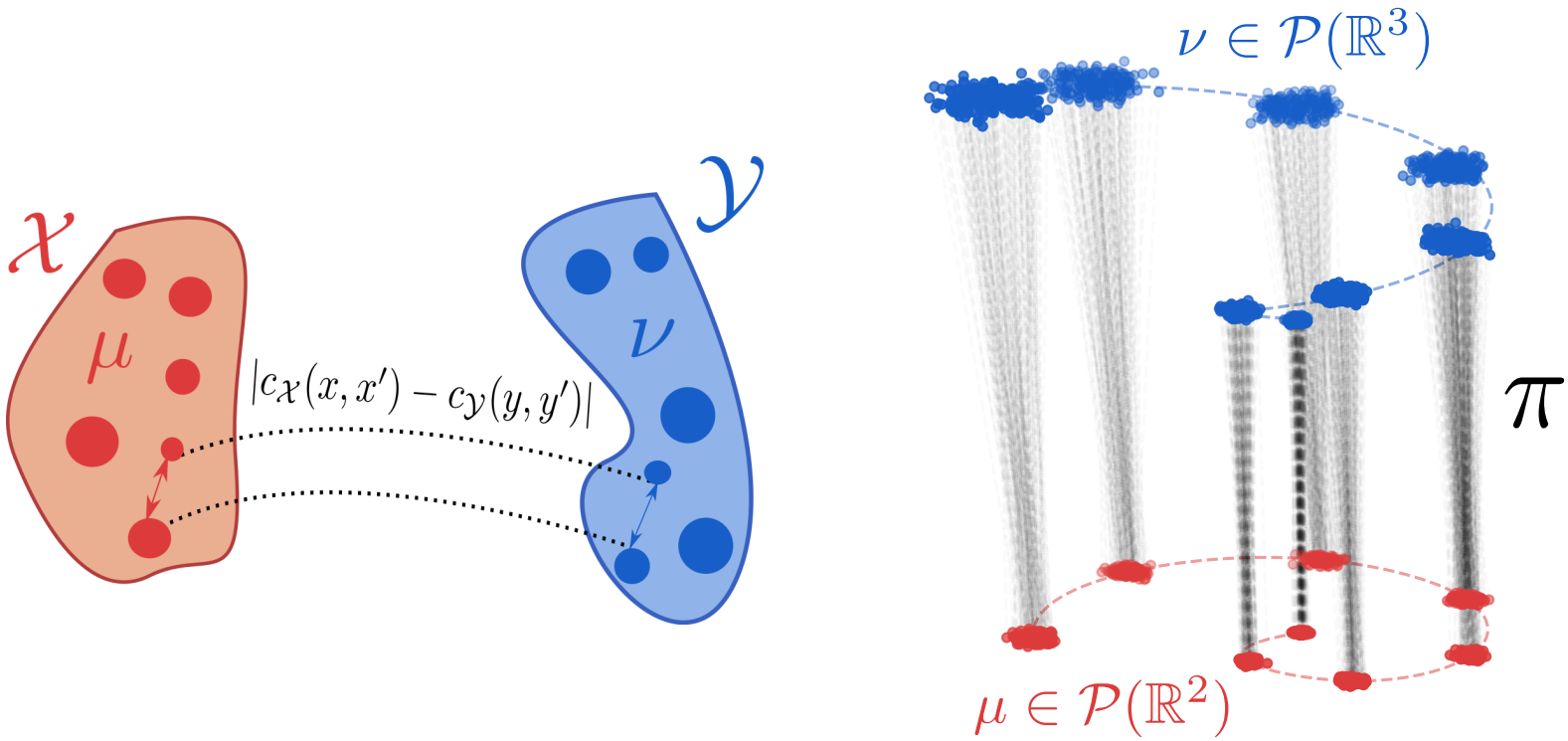


From POT

Extensions

Problems not solved by the OT problem in its original formulation:

- Outliers $\rightarrow$ Unbalanced and Partial OT (Séjourné et al., 2023)
- Comparing positive measures $\rightarrow$ Unbalanced and Partial OT
- Comparing distributions on incomparable spaces $\rightarrow$ Gromov-Wasserstein



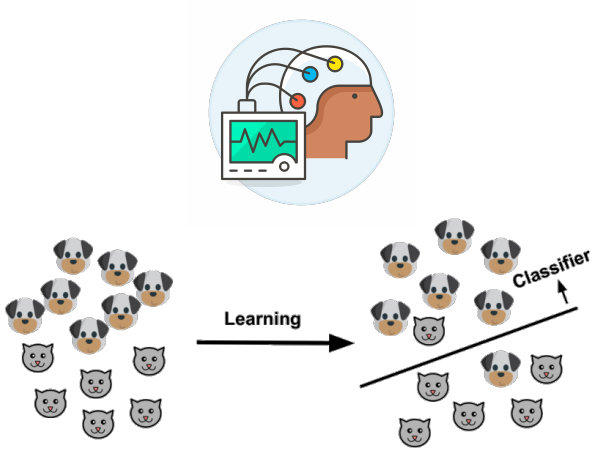
From (Vayer, 2020)

Table of Contents

Applications of OT in ML

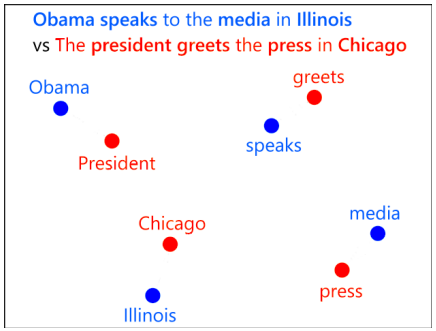
Wide range of applications:

- Supervised learning (Frogner et al., 2015)
- Domain Adaptation (Courty et al., 2016)
- Generative Modeling (Arjovsky et al., 2017)
- Biology (Schiebinger et al., 2019)
- Neuroscience (Bonet et al., 2023)
- Transformers (Sander et al., 2022)



Wide range of data types:

- Images
- Datasets (Alvarez-Melis and Fusi, 2020)
- Documents (Kusner et al., 2015)
- Genes (Bellazzi et al., 2021)
- Graphs (Vayer et al., 2019)



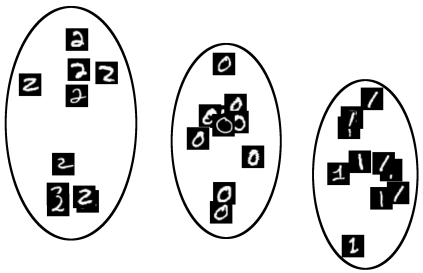
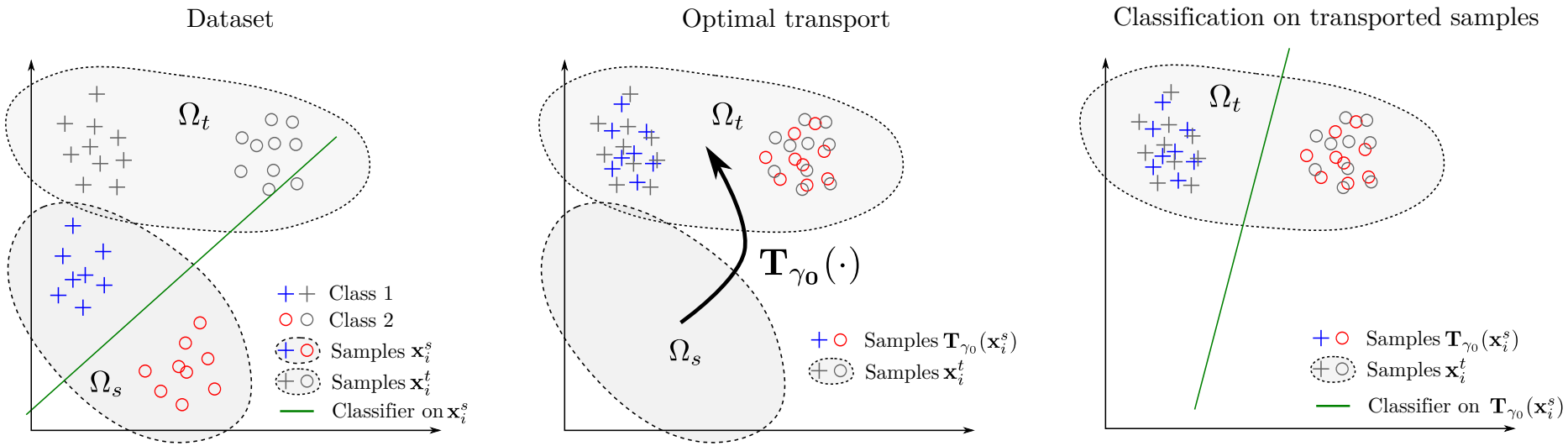


Table of Contents

Domain Adaptation with Optimal Transport



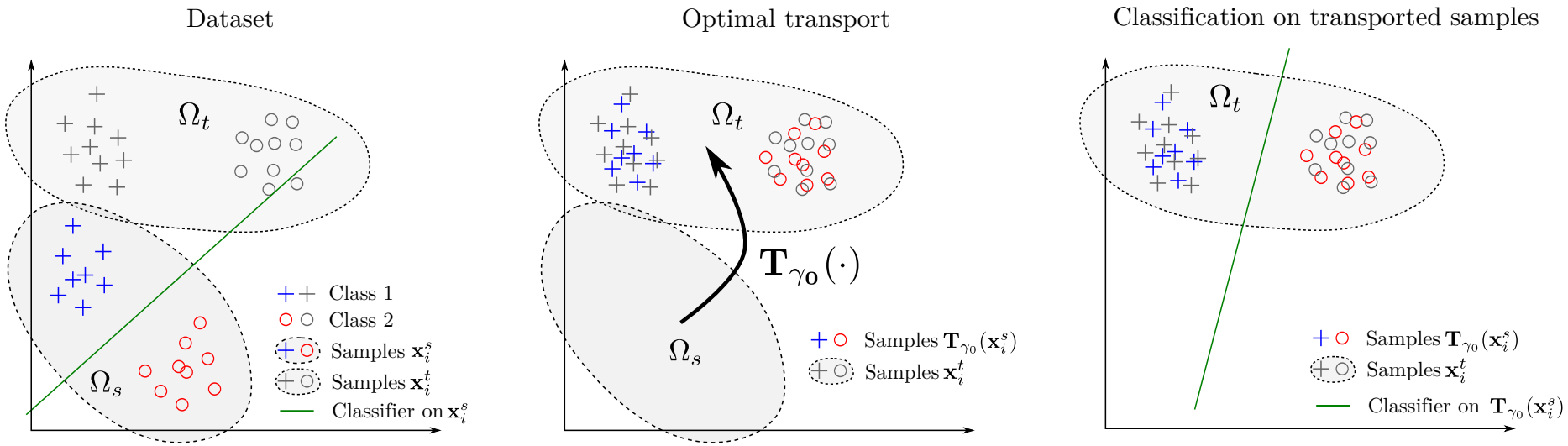
From (Courty et al., 2016)

- Labeled source dataset $\Omega_s = (x_i^s, y_i^s)_i$, $\mu_s = \frac{1}{n}\sum_{i=1}^n \delta_{x_i^s}$
- Unlabeled target dataset $\Omega_t = (x_j^t)_j$, $\mu_t = \frac{1}{m}\sum_{j=1}^m \delta_{x_j^t}$

Solution from (Courty et al., 2016):

1. Find the OT plan $\gamma \in \operatorname{argmin}_{\gamma \in \Pi(\mu_s, \mu_t)} \int \|x - y\|_2^2 \, \mathrm{d}\gamma(x, y)$
2. Define a barycentric map: $T_\gamma(x_i^s) = \operatorname{argmin}_x \sum_{j=1}^m \|x - x_j^t\|_2^2 \gamma_{ij}$
3. Train a classifier on $(T_\gamma(x_i^s), y_i^s)_i$

Domain Adaptation with Optimal Transport



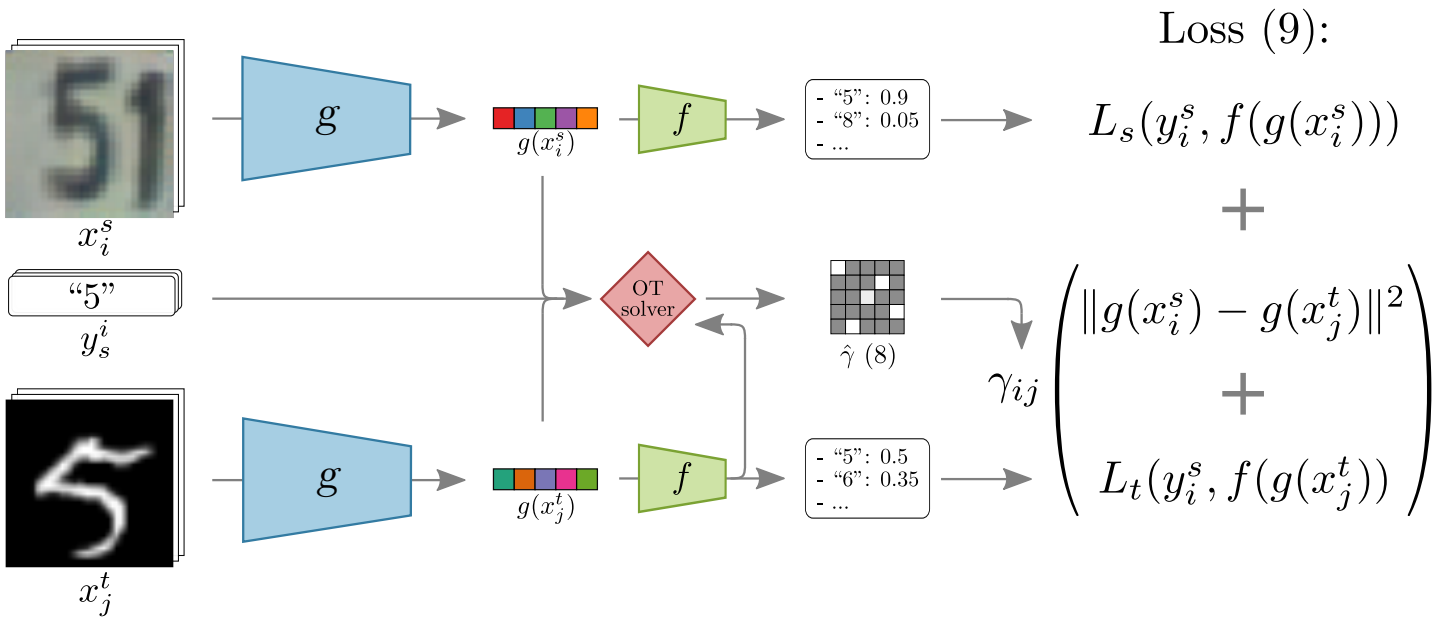
From (Courty et al., 2016)

- Labeled source dataset $\Omega_s = (x_i^s, y_i^s)_i$, $\mu_s = \frac{1}{n}\sum_{i=1}^n \delta_{x_i^s}$
- Unlabeled target dataset $\Omega_t = (x_j^t)_j$, $\mu_t = \frac{1}{m}\sum_{j=1}^m \delta_{x_j^t}$

Solution from (Courty et al., 2016):

1. Find the OT plan $\gamma \in \operatorname{argmin}_{\gamma \in \Pi(\mu_s, \mu_t)} \int \|x - y\|_2^2 \, \mathrm{d}\gamma(x, y) + \mathcal{R}_c(\gamma)$
2. Define a barycentric map: $T_\gamma(x_i^s) = \operatorname{argmin}_x \sum_{j=1}^m \|x - x_j^t\|_2^2 \gamma_{ij}$
3. Train a classifier on $(T_\gamma(x_i^s), y_i^s)_i$

DeepJDOT



From (Damodaran et al., 2018)

Idea: Learn suitable embedding, and align joint distribution of features and labels.

Table of Contents

Generative Modeling

Setting:

- $\nu \in \mathcal{P}_2(\mathrm{X})$: Target measure (*e.g.* distribution of images)
- P_Z a standard measure on a latent space Z

Goal of generative modeling: learn a map $g : \mathrm{Z} \to \mathrm{X}$ such that $g_{\#} P_Z \approx \nu$.

Remark: For $\mathrm{Z} = \mathrm{X} = \mathbb{R}^d$ and $P_Z \ll \mathrm{Leb}$, such a g exists by Brenier's theorem.

Generative Modeling

Setting:

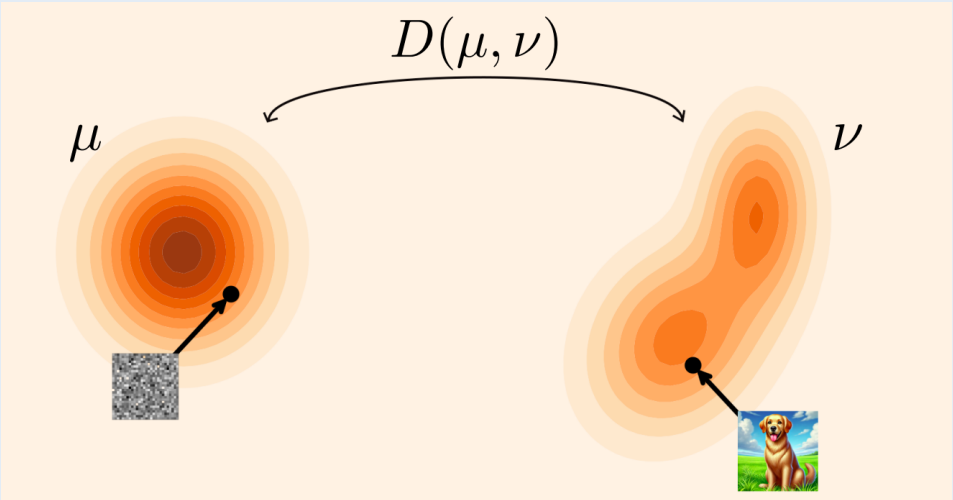
- $\nu \in \mathcal{P}_2(\mathrm{X})$: Target measure (*e.g.* distribution of images)
- P_Z a standard measure on a latent space Z

Goal of generative modeling: learn a map $g : \mathrm{Z} \to \mathrm{X}$ such that $g_{\#} P_Z \approx \nu$.

Remark: For $\mathrm{Z} = \mathrm{X} = \mathbb{R}^d$ and $P_Z \ll \mathrm{Leb}$, such a g exists by Brenier's theorem.

Solutions:

- Solve $\min_g \; D(g_{\#} P_Z, \nu)$ for D a divergence (GANs, VAEs, NFs...)



Generative Modeling

Setting:

- $\nu \in \mathcal{P}_2(\mathrm{X})$: Target measure (*e.g.* distribution of images)
- P_Z a standard measure on a latent space Z

Goal of generative modeling: learn a map $g : \mathrm{Z} \to \mathrm{X}$ such that $g_{\#} P_Z \approx \nu$.

Remark: For $\mathrm{Z} = \mathrm{X} = \mathbb{R}^d$ and $P_Z \ll \mathrm{Leb}$, such a g exists by Brenier's theorem.

Solutions:

- Solve $\min_g \; D(g_{\#} P_Z, \nu)$ for D a divergence (GANs, VAEs, NFs...)
- Learn the OT map, *i.e.*

$$g \in \underset{T,\; T_{\#} P_Z = \nu}{\operatorname{argmin}} \int \|T(z) - z\|_2^2 \, \mathrm{d}P_Z(z)$$

$\to$ difficulties: enforce $T_{\#} P_Z = \nu$, only access to samples from ν

Generative Modeling

Setting:

- $\nu \in \mathcal{P}_2(\mathrm{X})$: Target measure (*e.g.* distribution of images)
- P_Z a standard measure on a latent space Z

Goal of generative modeling: learn a map $g : \mathrm{Z} \to \mathrm{X}$ such that $g_\# P_Z \approx \nu$.

Remark: For $\mathrm{Z} = \mathrm{X} = \mathbb{R}^d$ and $P_Z \ll \mathrm{Leb}$, such a g exists by Brenier's theorem.

Solutions:

- Solve $\min_g \; D(g_\# P_Z, \nu)$ for D a divergence (GANs, VAEs, NFs...)
- Learn the OT map, *i.e.*

$$g \in \underset{T,\; T_\# P_Z = \nu}{\operatorname{argmin}} \int \|T(z) - z\|_2^2 \, \mathrm{d}P_Z(z)$$

  $\to$ difficulties: enforce $T_\# P_Z = \nu$, only access to samples from ν
- Learn a trajectory going from P_Z to ν (Diffusion, Flow Matching...)

Wasserstein GAN (Arjovsky et al., 2017)

Objective:

$$\min_g \ \mathrm{W}_1(g_{\#}P_Z, \nu)$$

Dual

Let $\mu, \nu \in \mathcal{P}_2(\mathbb{R}^d)$, $c : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$,

$$\begin{aligned} \mathrm{OT}_c(\mu, \nu) &= \inf_{\gamma \in \Pi(\mu,\nu)} \int c(x, y) \ \mathrm{d}\gamma(x, y) \\ &= \sup_{f \oplus g \leq c} \int f \mathrm{d}\mu + \int g \mathrm{d}\nu, \end{aligned}$$

where $f \oplus g \leq c$ means that for all x, y, $f(x) + g(y) \leq c(x, y)$.

In the particular case where $c(x, y) = \|x - y\|_2$,

$$\mathrm{W}_1(\mu, \nu) = \sup_{f \in \mathrm{Lip}_1} \int f \mathrm{d}(\mu - \nu).$$

Wasserstein GAN (Arjovsky et al., 2017)

$$\mathrm{W}_1(\mu,\nu) = \sup_{f\in \mathrm{Lip}_1} \int f \mathrm{d}(\mu-\nu).$$

Wasserstein GANs:

- Parametrize a function g_θ with a neural network
- Solve: $\min_\theta \ \mathrm{W}_1\big((g_\theta)_\# P_Z, \nu\big)$
- Use the dual and a 1-Lipschitz parametric critic function $f_\xi : \mathrm{X} \to \mathbb{R}$:

$$\inf_\theta \sup_\xi \int f_\xi\big(g_\theta(z)\big)\ \mathrm{d}P_Z(z) - \int f_\xi(x)\ \mathrm{d}\nu(x)$$

- Learn a 1-Lipschitz Neural network
- Bilevel optimization
- Often instable

Optimizing Divergences in $(\mathcal{P}_2(\mathbb{R}^d), \mathrm{W}_2)$

Goal:

$$\min_{\mu} \ \mathcal{F}(\mu) := D(\mu, \nu)$$

for D a divergence (*e.g.* $D = \mathrm{KL}$)

Definition (Wasserstein Gradient Flow (WGF))

A Wasserstein gradient flow is a continuous curve $\mu : [0, T] \to \mathcal{P}_2(\mathbb{R}^d)$ which follows the steepest descent direction to minimize $\mathcal{F}$ in $(\mathcal{P}_2(\mathbb{R}^d), \mathrm{W}_2)$.

Optimizing Divergences in $(\mathcal{P}_2(\mathbb{R}^d), \mathrm{W}_2)$

Goal:

$$\min_{\mu} \ \mathcal{F}(\mu) := D(\mu, \nu)$$

for D a divergence (*e.g.* $D = \mathrm{KL}$)

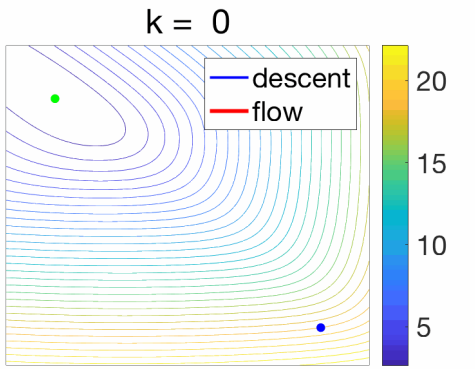
Definition (Wasserstein Gradient Flow (WGF))

A Wasserstein gradient flow is a continuous curve $\mu : [0, T] \to \mathcal{P}_2(\mathbb{R}^d)$ which follows the steepest descent direction to minimize $\mathcal{F}$ in $(\mathcal{P}_2(\mathbb{R}^d), \mathrm{W}_2)$.

Analogous to Euclidean Gradient Flows to minimize $f : \mathbb{R}^d \to \mathbb{R}$:

$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = -\nabla f(x_t)$$

$\to$ Discretization in time: Gradient descent



From (Bach, 2020)

Optimizing Divergences in $(\mathcal{P}_2(\mathbb{R}^d), \mathrm{W}_2)$

Goal:

$$\min_{\mu} \ \mathcal{F}(\mu) := D(\mu, \nu)$$

for D a divergence (*e.g.* $D = \mathrm{KL}$)

Definition (Wasserstein Gradient Flow (WGF))

A Wasserstein gradient flow is a continuous curve $\mu : [0, T] \to \mathcal{P}_2(\mathbb{R}^d)$ which follows the steepest descent direction to minimize $\mathcal{F}$ in $(\mathcal{P}_2(\mathbb{R}^d), \mathrm{W}_2)$.

	Gradient Flows on $\mathbb{R}^d$	WGF
Characterization	ODE $\frac{\mathrm{d}x_t}{\mathrm{d}t} = -\nabla f(x_t)$	PDE (Continuity Equation) $\partial_t \mu_t - \mathrm{div}\big(\mu_t \nabla_{\mathrm{W}_2} \mathcal{F}(\mu_t)\big) = 0$
Discretization (Forward) in time $\forall k \geq 0,$	Gradient Descent $x_{k+1} = x_k - \tau \nabla f(x_k)$	Wasserstein Gradient Descent $\mu_{k+1} = \big(\mathrm{Id} - \tau \nabla_{\mathrm{W}_2} \mathcal{F}(\mu_k)\big)_{\#} \mu_k$
Discretization (Backward) in time $\forall k \geq 0,$	Proximal Point Algorithm $x_{k+1} = \mathrm{argmin}_x \ \frac{1}{2\tau}\|x - x_k\|_2^2 + f(x)$	JKO scheme $\mu_{k+1} = \mathrm{argmin}_\mu \ \frac{1}{2\tau}\mathrm{W}_2^2(\mu, \mu_k) + \mathcal{F}(\mu)$

JKO Scheme

Endowing $\mathcal{P}_2(\mathbb{R}^d)$ with W_2, JKO scheme (Jordan et al., 1998):

$$\forall k \geq 0,\ \mu_{k+1}^{\tau} \in \underset{\mu \in \mathcal{P}_2(\mathbb{R}^d)}{\operatorname{argmin}} \ \frac{1}{2\tau}\mathrm{W}_2^2(\mu, \mu_k^{\tau}) + \mathcal{F}(\mu)$$

$\rightarrow$ solve this problem with neural networks

JKO Scheme

Endowing $\mathcal{P}_2(\mathbb{R}^d)$ with W_2, JKO scheme (Jordan et al., 1998):

$$\forall k \geq 0,\ \mu_{k+1}^\tau \in \operatorname*{argmin}_{\mu \in \mathcal{P}_2(\mathbb{R}^d)} \frac{1}{2\tau}\mathrm{W}_2^2(\mu, \mu_k^\tau) + \mathcal{F}(\mu)$$

$\rightarrow$ solve this problem with neural networks

- JKO-ICNN (Mokrov et al., 2021; Alvarez-Melis et al., 2022)

$$\begin{cases} u_{k+1}^\tau = \operatorname{argmin}_{u \in \mathrm{CVX}} \ \frac{1}{2\tau} \int \|\nabla u(x) - x\|_2^2 \ \mathrm{d}\mu_k^\tau(x) + \mathcal{F}\big((\nabla u)_\# \mu_k^\tau\big) \\ \mu_{k+1}^\tau = (\nabla u_{k+1}^\tau)_\# \mu_k^\tau \end{cases}$$

Drawback: use ICNNs, $O(k^2)$ gradient evaluations at each step

JKO Scheme

Endowing $\mathcal{P}_2(\mathbb{R}^d)$ with W_2, JKO scheme (Jordan et al., 1998):

$$\forall k \geq 0,\ \mu_{k+1}^\tau \in \operatorname*{argmin}_{\mu \in \mathcal{P}_2(\mathbb{R}^d)} \frac{1}{2\tau}\mathrm{W}_2^2(\mu, \mu_k^\tau) + \mathcal{F}(\mu)$$

$\rightarrow$ solve this problem with neural networks

- JKO-ICNN (Mokrov et al., 2021; Alvarez-Melis et al., 2022)

$$\begin{cases} u_{k+1}^\tau = \operatorname{argmin}_{u \in \mathrm{CVX}} \ \frac{1}{2\tau} \int \|\nabla u(x) - x\|_2^2 \ \mathrm{d}\mu_k^\tau(x) + \mathcal{F}\big((\nabla u)_\# \mu_k^\tau\big) \\ \mu_{k+1}^\tau = (\nabla u_{k+1}^\tau)_\# \mu_k^\tau \end{cases}$$

  Drawback: use ICNNs, $O(k^2)$ gradient evaluations at each step

- (Fan et al., 2022; Choi et al., 2024): parametrize the Monge map, *i.e.*

$$T_{k+1}^\tau = \operatorname*{argmin}_T \ \frac{1}{2\tau} \int \|T(x) - x\|_2^2 \ \mathrm{d}\mu_k^\tau(x) + \mathcal{F}(T_\# \mu_k^\tau), \quad \mu_{k+1}^\tau = (T_{k+1}^\tau)_\# \mu_k^\tau$$

  For $\mathcal{F}(\mu) = \mathrm{KL}(\mu||\nu)$, use the variational formulation.

Learning the OT Map

Learning the OT map, *i.e.*

$$T \in \operatorname*{argmin}_{T_\# P_Z = \nu} \int \|T(z) - z\|_2^2 \, \mathrm{d}P_Z(z)$$

- Leveraging the dual, Makkuva et al. (2020) solve

$$\mathrm{W}_2^2(P_Z, \nu) = \sup_{f \in \mathrm{CVX}(\nu)} \inf_{g \in \mathrm{CVX}(P_Z)} - \int f \mathrm{d}\nu - \int \left(\langle f(z), \nabla g(z) \rangle - f(\nabla g(z)) \right) \mathrm{d}P_Z(z),$$

and use $T = \nabla g$. f, g are parametrized by ICNNs (Amos et al., 2017)
$\rightarrow$ minimax problem, use ICNNs

Learning the OT Map

Learning the OT map, *i.e.*

$$T \in \underset{T_\# P_Z = \nu}{\operatorname{argmin}} \int \|T(z) - z\|_2^2 \, \mathrm{d}P_Z(z)$$

- Leveraging the dual, Makkuva et al. (2020) solve

$$\mathrm{W}_2^2(P_Z, \nu) = \sup_{f \in \mathrm{CVX}(\nu)} \inf_{g \in \mathrm{CVX}(P_Z)} - \int f \mathrm{d}\nu - \int \big(\langle f(z), \nabla g(z)\rangle - f(\nabla g(z))\big) \, \mathrm{d}P_Z(z),$$

  and use $T = \nabla g$. f, g are parametrized by ICNNs (Amos et al., 2017)
  $\rightarrow$ minimax problem, use ICNNs

- **Monge Gap** (Uscidda and Cuturi, 2023):

$$\min_\theta \; \mathcal{D}\big((T_\theta)_\# P_Z, \nu\big) + \mathcal{M}_c^{P_Z}(T_\theta),$$

  where

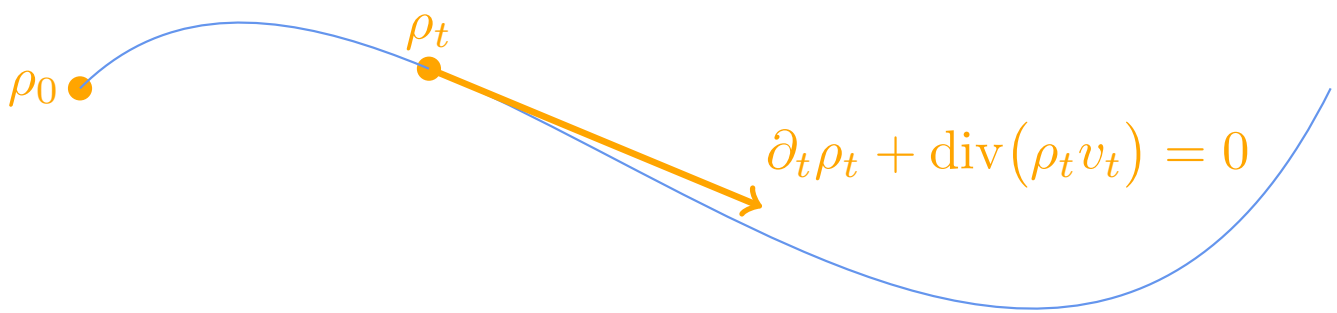
$$\mathcal{M}_c^\rho(T) = \int c\big(x, T(x)\big) \mathrm{d}\rho(x) - \mathrm{OT}_c(\rho, T_\# \rho) \geq 0.$$

  $\rightarrow$ learn OT map between P_Z and ν for any c.

Continuity equation

For any (absolutely continuous) curve $t \mapsto \rho_t \in \mathcal{P}_2(\mathbb{R}^d)$, there exists for *a.e.* $t \in [0,1]$ a velocity field $v_t : \mathbb{R}^d \to \mathbb{R}^d \in L^2(\rho_t)$ such that (ρ_t, v_t) satisfies (weakly) the **continuity equation**

$$\partial_t \rho_t + \operatorname{div}(\rho_t v_t) = 0.$$

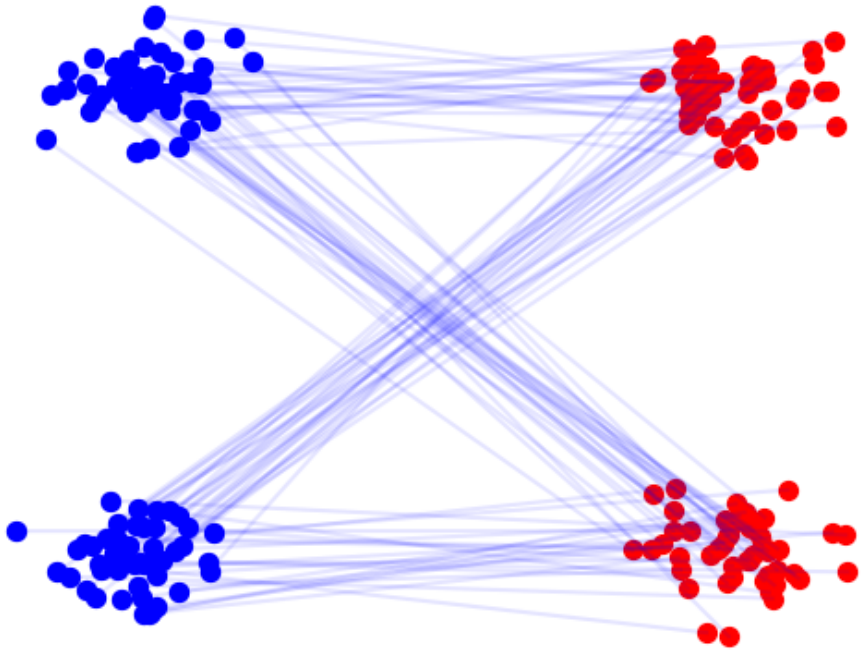


Particles: $x_t \sim \rho_t \iff \frac{\mathrm{d}x_t}{\mathrm{d}t} = v_t(x_t).$

Continuity equation

For any (absolutely continuous) curve $t \mapsto \rho_t \in \mathcal{P}_2(\mathbb{R}^d)$, there exists for *a.e.* $t \in [0,1]$ a velocity field $v_t : \mathbb{R}^d \to \mathbb{R}^d \in L^2(\rho_t)$ such that (ρ_t, v_t) satisfies (weakly) the **continuity equation**

$$\partial_t \rho_t + \operatorname{div}(\rho_t v_t) = 0.$$



Dynamic formulation of OT

Benamou-Brenier formulation (Benamou and Brenier, 2000)

Let $\mu, \nu \in \mathcal{P}_2(\mathbb{R}^d)$,

$$\mathrm{W}_2^2(\mu, \nu) = \inf_{\rho, v} \int_0^1 \int_{\mathbb{R}^d} \|v_t(x)\|_2^2 \, \mathrm{d}\rho_t(x)\mathrm{d}t$$

subject to $\partial_t \rho_t + \mathrm{div}(\rho_t v_t) = 0, \ \rho_0 = \mu, \ \rho_1 = \nu.$

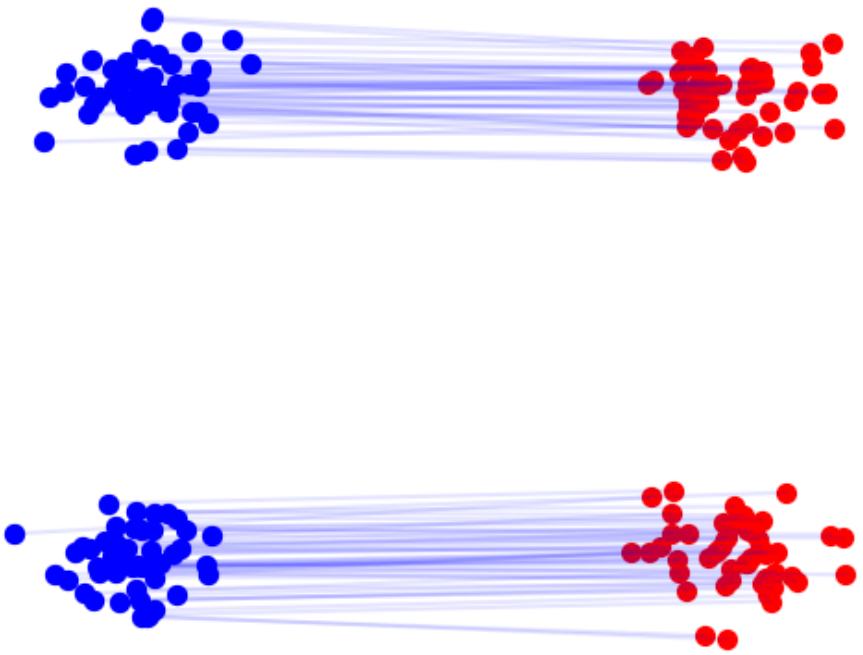
Dynamic formulation of OT

Benamou-Brenier formulation (Benamou and Brenier, 2000)

Let $\mu, \nu \in \mathcal{P}_2(\mathbb{R}^d)$,

$$\mathrm{W}_2^2(\mu, \nu) = \inf_{\rho, v} \int_0^1 \int_{\mathbb{R}^d} \|v_t(x)\|_2^2 \, \mathrm{d}\rho_t(x)\mathrm{d}t$$

subject to $\partial_t \rho_t + \mathrm{div}(\rho_t v_t) = 0, \ \rho_0 = \mu, \ \rho_1 = \nu.$



Flow Matchings (Lipman et al., 2022)

Idea: learn v_t by a regression problem

Ingredients:

- An interpolating curve between $x_0 \sim \mu, x_1 \sim \nu$: $x_t = I_t(x_0, x_1)$ *e.g.*

$$\forall t \in [0,1],\ I_t(x_0, x_1) = x_t = (1-t)x_0 + tx_1$$

- A coupling $\gamma \in \Pi(\mu, \nu) \rightarrow$ defines $\mu_t = (I_t)_{\#}\gamma$ curve interpolating between μ_0 and μ_1
- Learn v_θ such that $\frac{\mathrm{d}x_t}{\mathrm{d}t} = v_\theta(t, x_t)$
- After training, solve the ODE $\frac{\mathrm{d}x_t}{\mathrm{d}t} = v_\theta(t, x_t)$ to sample from ν

Flow Matchings (Lipman et al., 2022)

Idea: learn v_t by a regression problem

Ingredients:

- An interpolating curve between $x_0 \sim \mu, x_1 \sim \nu$: $x_t = I_t(x_0, x_1)$ *e.g.*

$$\forall t \in [0,1],\ I_t(x_0, x_1) = x_t = (1-t)x_0 + tx_1$$

- A coupling $\gamma \in \Pi(\mu, \nu) \rightarrow$ defines $\mu_t = (I_t)_{\#}\gamma$ curve interpolating between μ_0 and μ_1
- Learn v_θ such that $\frac{\mathrm{d}x_t}{\mathrm{d}t} = v_\theta(t, x_t)$
- After training, solve the ODE $\frac{\mathrm{d}x_t}{\mathrm{d}t} = v_\theta(t, x_t)$ to sample from ν

Loss:

$$\mathcal{L}(\theta) = \int_0^1 \int \left\| v_\theta\big(t, I_t(x_0, x_1)\big) - \frac{\mathrm{d}}{\mathrm{d}t} I_t(x_0, x_1) \right\|_2^2 \mathrm{d}\gamma(x_0, x_1)\mathrm{d}t$$

Flow Matchings (Lipman et al., 2022)

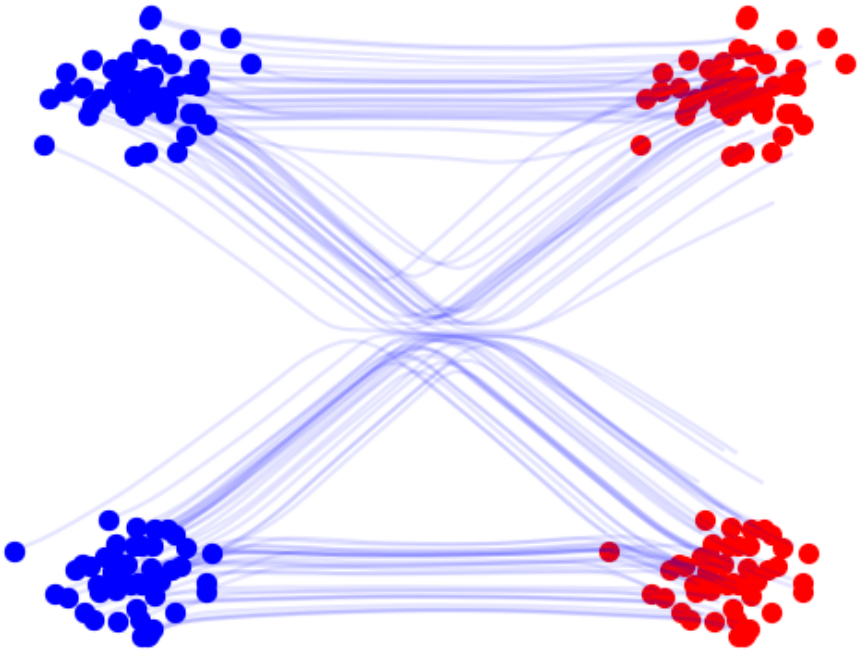
Simplest setting: $\gamma = \mu \otimes \nu \in \Pi(\mu, \nu)$, $I_t(x_0, x_1) = (1-t)x_0 + tx_1$:

$$\mathcal{L}(\theta) = \int_0^1 \int \| v_t\big((1-t)x_0 + tx_1\big) - \big(x_1 - x_0\big) \|_2^2 \, \mathrm{d}\mu(x_0)\mathrm{d}\nu(x_1)\mathrm{d}t,$$

Flow Matchings (Lipman et al., 2022)

Simplest setting: $\gamma = \mu \otimes \nu \in \Pi(\mu, \nu)$, $I_t(x_0, x_1) = (1-t)x_0 + tx_1$:

$$\mathcal{L}(\theta) = \int_0^1 \int \| v_t\big((1-t)x_0 + tx_1\big) - \big(x_1 - x_0\big) \|_2^2 \, \mathrm{d}\mu(x_0)\mathrm{d}\nu(x_1)\mathrm{d}t,$$



Rectified Flows (Liu et al., 2022)

Trajectories: not straight $\rightarrow$ ODE longer to solve

Rectified Flows:

- Perform Flow matching
- Denote $T_1(x_0) = x_0 + \int_0^1 v_\theta(t, x_t)\mathrm{d}t$
- Perform Flow matching using $\gamma = (\mathrm{Id}, T_1)_{\#}\mu \in \Pi(\mu, \nu)$

$\rightarrow$ Reduce the OT cost and get straighter trajectories

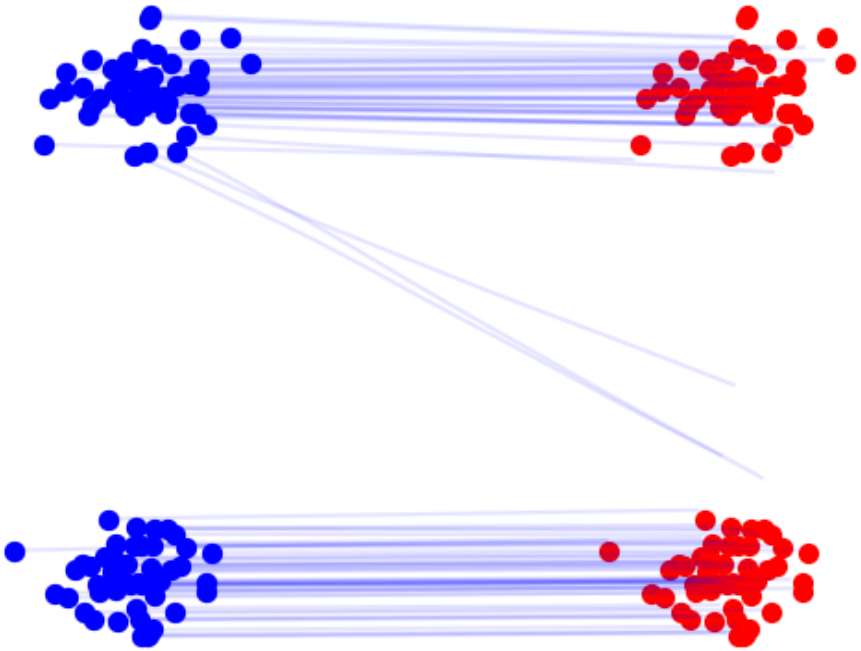
Rectified Flows (Liu et al., 2022)

Trajectories: not straight $\rightarrow$ ODE longer to solve

Rectified Flows:

- Perform Flow matching
- Denote $T_1(x_0) = x_0 + \int_0^1 v_\theta(t, x_t)\mathrm{d}t$
- Perform Flow matching using $\gamma = (\mathrm{Id}, T_1)_{\#}\mu \in \Pi(\mu, \nu)$

$\rightarrow$ Reduce the OT cost and get straighter trajectories



Flow Matchings with OT Couplings (Tong et al., 2023; Pooladian et al., 2023)

Use OT couplings $\gamma \in \Pi_o(\mu, \nu)$, $I_t(x_0, x_1) = (1-t)x_0 + tx_1$:

$$\mathcal{L}(\theta) = \int_0^1 \int \|v_t\big((1-t)x_0 + tx_1\big) - \big(x_1 - x_0\big)\|_2^2 \, \mathrm{d}\gamma(x_0, x_1)\mathrm{d}t.$$

- If too much samples: use minibatch OT (Tong et al., 2023; Pooladian et al., 2023)

Flow Matchings with OT Couplings (Tong et al., 2023; Pooladian et al., 2023)

Use OT couplings $\gamma \in \Pi_o(\mu, \nu)$, $I_t(x_0, x_1) = (1-t)x_0 + tx_1$:

$$\mathcal{L}(\theta) = \int_0^1 \int \|v_t\big((1-t)x_0 + tx_1\big) - \big(x_1 - x_0\big)\|_2^2 \, \mathrm{d}\gamma(x_0, x_1)\mathrm{d}t.$$

- If too much samples: use minibatch OT (Tong et al., 2023; Pooladian et al., 2023)
- Klein et al. (2025): Use big batchs with Sinkhorn on multiple GPUs

ImageNet-32					ImageNet-64				
NFE → $n \downarrow$	4	8	16	Adaptive 115 ± 1	NFE → $n \downarrow$	4	8	16	Adaptive 269 ± 1
I-FM	66.4	24.3	12.1	5.55	I-FM	80.1	37.0	19.5	9.32
2048	38.2	16.8	10.0	5.89	4096	50.3	25.0	15.8	9.39
65536	33.1	15.1	9.28	4.88	32768	48.8	24.6	15.7	9.08
524288	**31.5**	**14.8**	**9.19**	**4.85**	131072	**46.9**	**23.9**	**15.4**	**8.99**

Table 1: FID for models trained across different OT batch sizes. We use the best checkpoint (w.r.t FID at `Dopri5`) for each model, restricting results to the setting where the relative `epsilon` value $\varepsilon = 0.1$ for ease of presentation (more detailed results can be seen in the plots of Figure 5).

From (Klein et al., 2025)

Flow Matchings with OT Couplings (Tong et al., 2023; Pooladian et al., 2023)

Use OT couplings $\gamma \in \Pi_o(\mu, \nu)$, $I_t(x_0, x_1) = (1-t)x_0 + tx_1$:

$$\mathcal{L}(\theta) = \int_0^1 \int \|v_t\big((1-t)x_0 + tx_1\big) - \big(x_1 - x_0\big)\|_2^2 \, \mathrm{d}\gamma(x_0, x_1)\mathrm{d}t.$$

- If too much samples: use minibatch OT (Tong et al., 2023; Pooladian et al., 2023)
- Klein et al. (2025): Use big batchs with Sinkhorn on multiple GPUs
- Mousavi-Hosseini et al. (2025): Pre-compute semi-discrete OT plan between $\mu = p_Z$ and ν

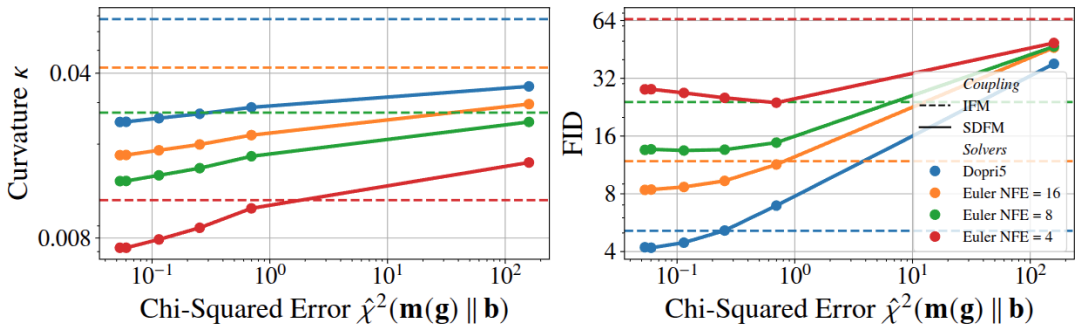


Figure 3 Better SD Potential Estimation = Better Curvature and FID. On ImgN32, convergence of dual potential $\mathbf{g}$ *vs.* SD-FM ($\varepsilon = 0$) curvature and FID; I-FM is shown as lines. Note that curvatures of different solvers are computed on different trajectories, hence they are not comparable.

From (Mousavi-Hosseini et al., 2025)

Thank you for your attention!

References I

David Alvarez-Melis and Nicolo Fusi. Geometric dataset distances via optimal transport. *Advances in Neural Information Processing Systems*, 33: 21428–21439, 2020.

David Alvarez-Melis, Yair Schiff, and Youssef Mroueh. Optimizing functionals on the space of probabilities with input convex neural networks. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *International Conference on Machine Learning*, pages 146–155. PMLR, 2017.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

Francis Bach. Effortless optimization through gradient flows, 2020. URL https://francisbach.com/gradient-flows/.

Riccardo Bellazzi, Andrea Codegoni, Stefano Gualandi, Giovanna Nicora, and Eleonora Vercesi. The gene mover's distance: Single-cell similarity via optimal transport. *arXiv preprint arXiv:2102.01218*, 2021.

References II

Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84 (3):375–393, 2000.

Emmanuel Boissard and Thibaut Le Gouic. On the mean speed of convergence of empirical and occupation measures in wasserstein distance. In *Annales de l'IHP Probabilités et statistiques*, volume 50, pages 539–563, 2014.

Clément Bonet, Benoît Malézieux, Alain Rakotomamonjy, Lucas Drumetz, Thomas Moreau, Matthieu Kowalski, and Nicolas Courty. Sliced-Wasserstein on Symmetric Positive Definite Matrices for M/EEG Signals. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2777–2805. PMLR, 23–29 Jul 2023.

Clément Bonet, Nicolas Courty, François Septier, and Lucas Drumetz. Efficient gradient flows in sliced-wasserstein space. *Transactions on Machine Learning Research*, 2022.

Nicolas Bonnotte. *Unidimensional and evolution methods for optimal transportation*. PhD thesis, Paris 11, 2013.

References III

Yann Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics*, 44(4):375–417, 1991.

Jaemoo Choi, Jaewoong Choi, and Myungjoo Kang. Scalable wasserstein gradient flow for generative modeling through unbalanced optimal transport. *arXiv preprint arXiv:2402.05443*, 2024.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2016.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 447–463, 2018.

References IV

Jiaojiao Fan, Qinsheng Zhang, Amirhossein Taghvaei, and Yongxin Chen. Variational Wasserstein gradient flow. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 6185–6215. PMLR, 17–23 Jul 2022.

Kilian Fatras, Younes Zine, Szymon Majewski, Rémi Flamary, Rémi Gribonval, and Nicolas Courty. Minibatch optimal transport distances; analysis and applications. *arXiv preprint arXiv:2101.01792*, 2021.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al. Pot: Python optimal transport. *The Journal of Machine Learning Research*, 22(1):3571–3578, 2021.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. *Advances in neural information processing systems*, 28, 2015.

Anil Goyal. *Learning a Multiview Weighted Majority Vote Classifier: Using PAC-Bayesian Theory and Boosting*. PhD thesis, Lyon, 2018.

References V

Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1): 1–17, 1998.

Michal Klein, Alireza Mousavi-Hosseini, Stephen Zhang, and Marco Cuturi. On fitting flow models with large sinkhorn couplings. *arXiv preprint arXiv:2506.05526*, 2025.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR, 2015.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Ashok Makkuva, Amirhossein Taghvaei, Sewoong Oh, and Jason Lee. Optimal transport mapping via input convex neural networks. In *International Conference on Machine Learning*, pages 6672–6681. PMLR, 2020.

References VI

Petr Mokrov, Alexander Korotin, Lingxiao Li, Aude Genevay, Justin M Solomon, and Evgeny Burnaev. Large-scale wasserstein gradient flows. *Advances in Neural Information Processing Systems*, 34:15243–15256, 2021.

Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.*, pages 666–704, 1781.

Eduardo Fernandes Montesuma, Fred Maurice Ngole Mboula, and Antoine Souloumiac. Recent advances in optimal transport for machine learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Alireza Mousavi-Hosseini, Stephen Y Zhang, Michal Klein, and Marco Cuturi. Flow matching with semidiscrete couplings. *arXiv preprint arXiv:2509.25519*, 2025.

Kimia Nadjahi, Alain Durmus, Umut Simsekli, and Roland Badeau. Asymptotic guarantees for learning generative models with the sliced-wasserstein distance. *Advances in Neural Information Processing Systems*, 32, 2019.

Kimia Nadjahi, Alain Durmus, Lénaïc Chizat, Soheil Kolouri, Shahin Shahrampour, and Umut Simsekli. Statistical and topological properties of sliced probability divergences. *Advances in Neural Information Processing Systems*, 33: 20802–20812, 2020.

References VII

Ofir Pele and Michael Werman. Fast and robust earth mover's distances. In *2009 IEEE 12th international conference on computer vision*, pages 460–467. IEEE, 2009.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky TQ Chen. Multisample flow matching: Straightening flows with minibatch couplings. *arXiv preprint arXiv:2304.14772*, 2023.

Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 435–446. Springer, 2011.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

References VIII

Michael E Sander, Pierre Ablin, Mathieu Blondel, and Gabriel Peyré. Sinkformers: Transformers with doubly stochastic attention. In *International Conference on Artificial Intelligence and Statistics*, pages 3515–3530. PMLR, 2022.

Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943, 2019.

Thibault Séjourné, Gabriel Peyré, and François-Xavier Vialard. Unbalanced optimal transport, from theory to numerics. *Handbook of Numerical Analysis*, 24:407–471, 2023.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

Théo Uscidda and Marco Cuturi. The monge gap: A regularizer to learn all transport maps. In *International Conference on Machine Learning*, pages 34709–34733. PMLR, 2023.

References IX

Titouan Vayer. A contribution to optimal transport on incomparable spaces. *arXiv preprint arXiv:2011.04447*, 2020.

Titouan Vayer, Nicolas Courty, Romain Tavenard, and Rémi Flamary. Optimal transport for structured data with application on graphs. In *International Conference on Machine Learning*, pages 6275–6284. PMLR, 2019.