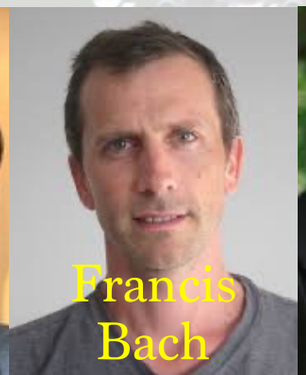
Scaling Optimal Transport to High Dimensional Learning

Gabriel Peyré

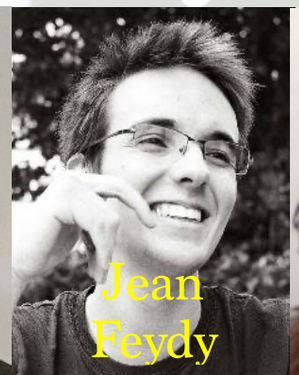
Joint work with:


Shun'ichi Amari

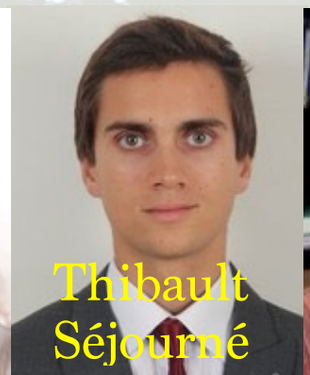

Francis Bach


Lénaic Chizat


Marco Cuturi


Jean Feydy

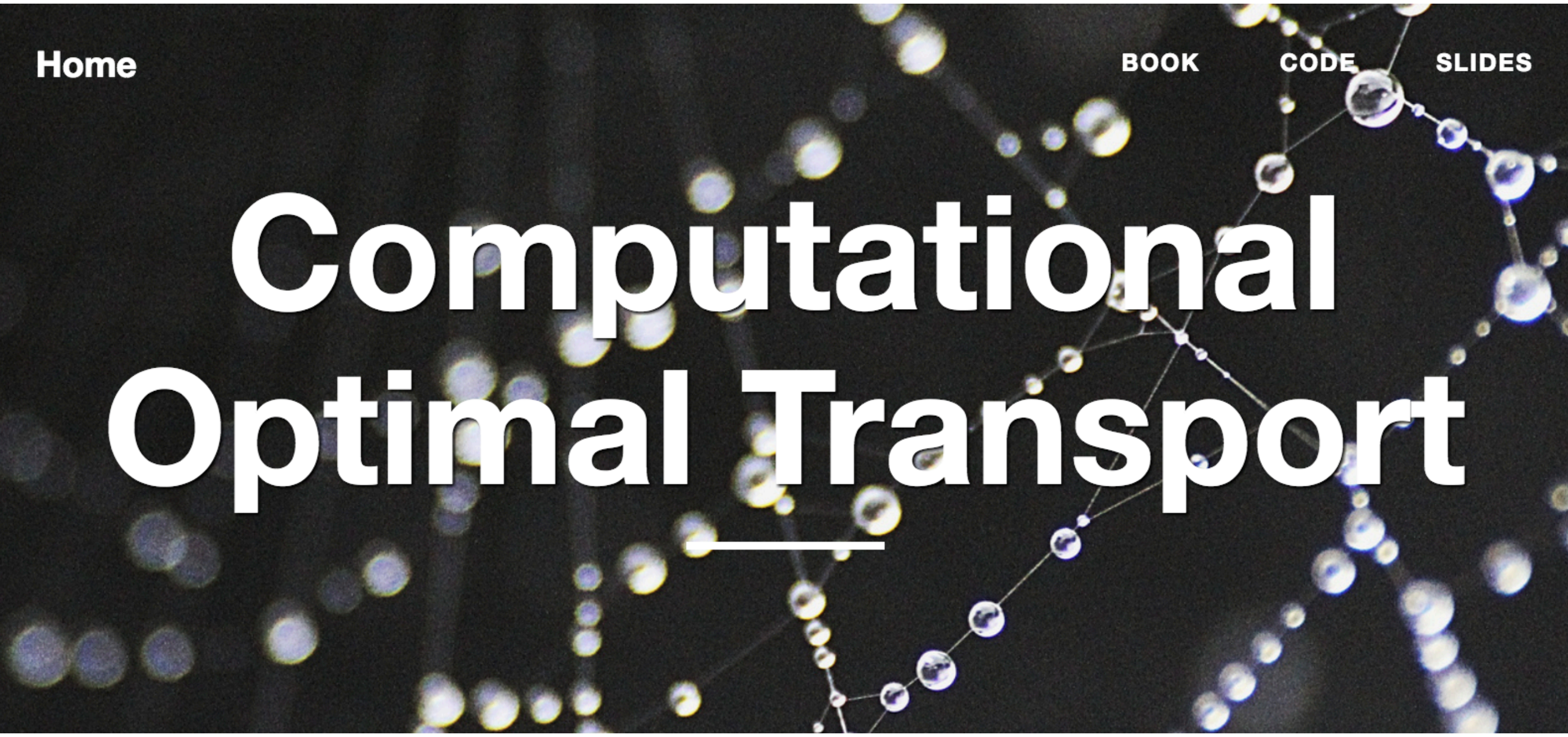

Aude Genevay


Thibault Séjourné


Alain Trouvé


François-Xavier Vialard

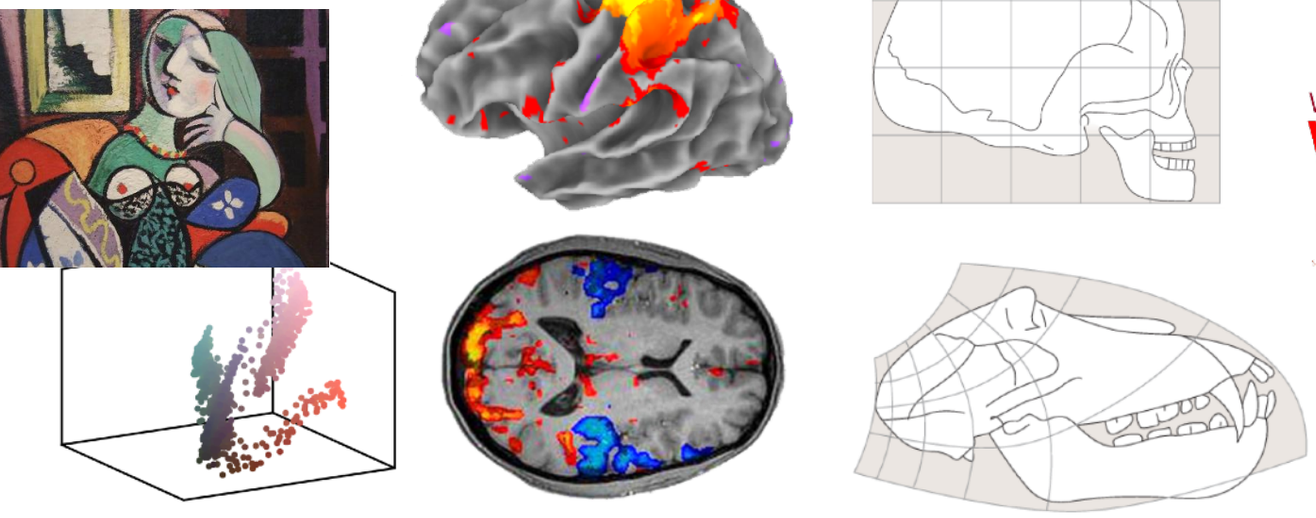
https://optimaltransport.github.io



Probability Distributions in Data Sciences

Probability distributions and histograms

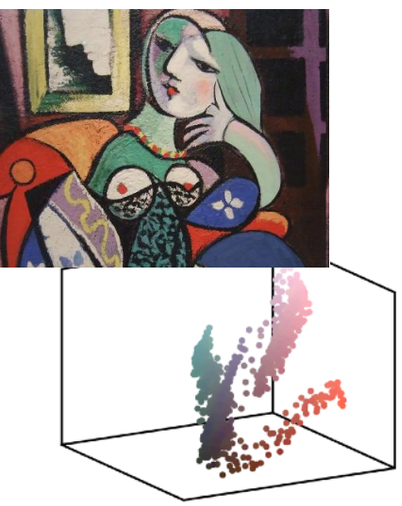
$\rightarrow$ images, vision, graphics and machine learning, . . .

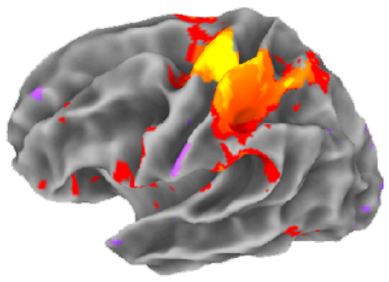


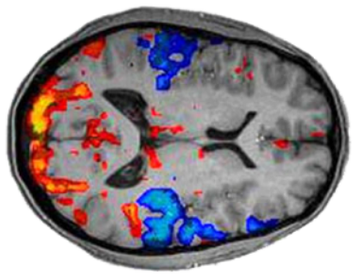
Probability Distributions in Data Sciences

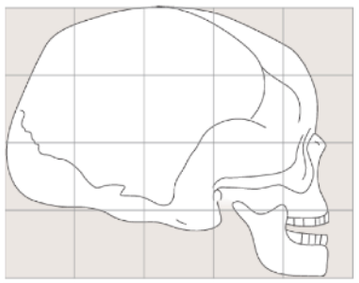
Probability distributions and histograms

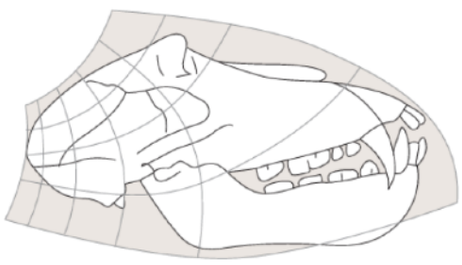
$\rightarrow$ images, vision, graphics and machine learning, . . .

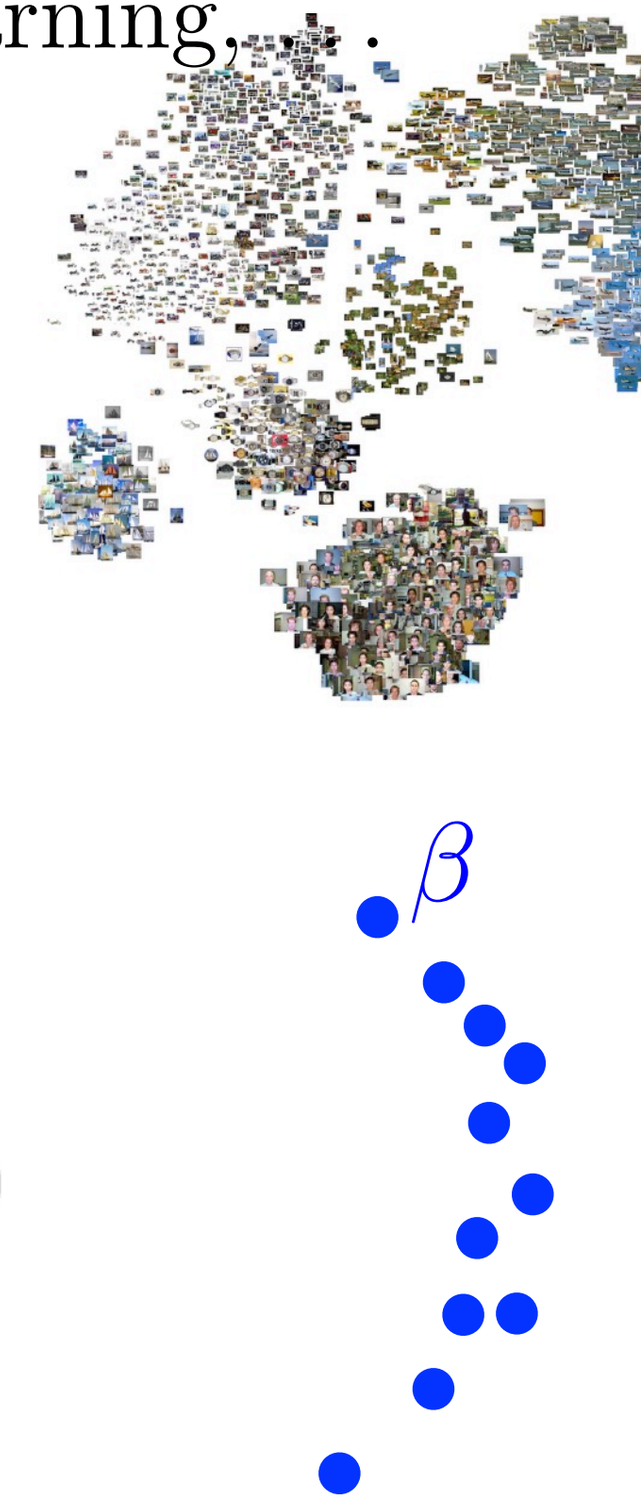








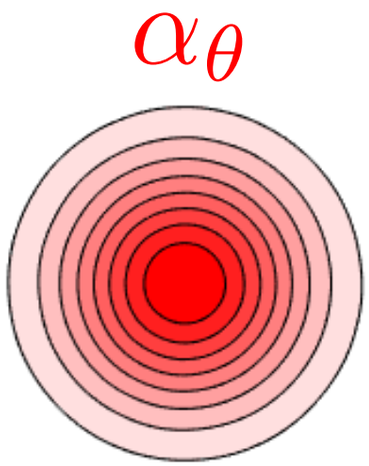




Unsupervised learning

Observations: $\beta \stackrel{\text{def.}}{=} \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}$

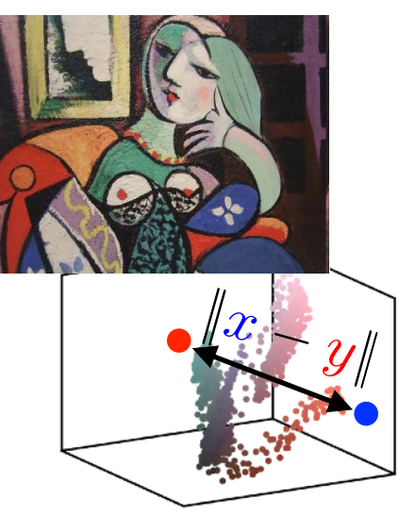
Parametric model: $\theta \mapsto \alpha_\theta$

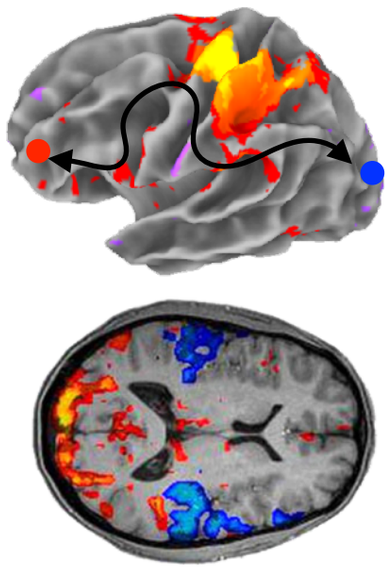


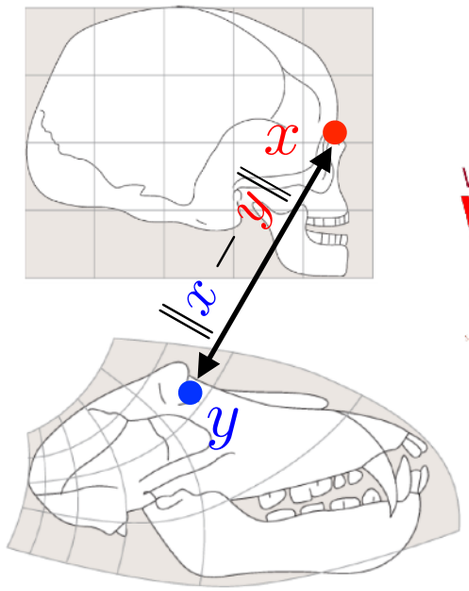
Probability Distributions in Data Sciences

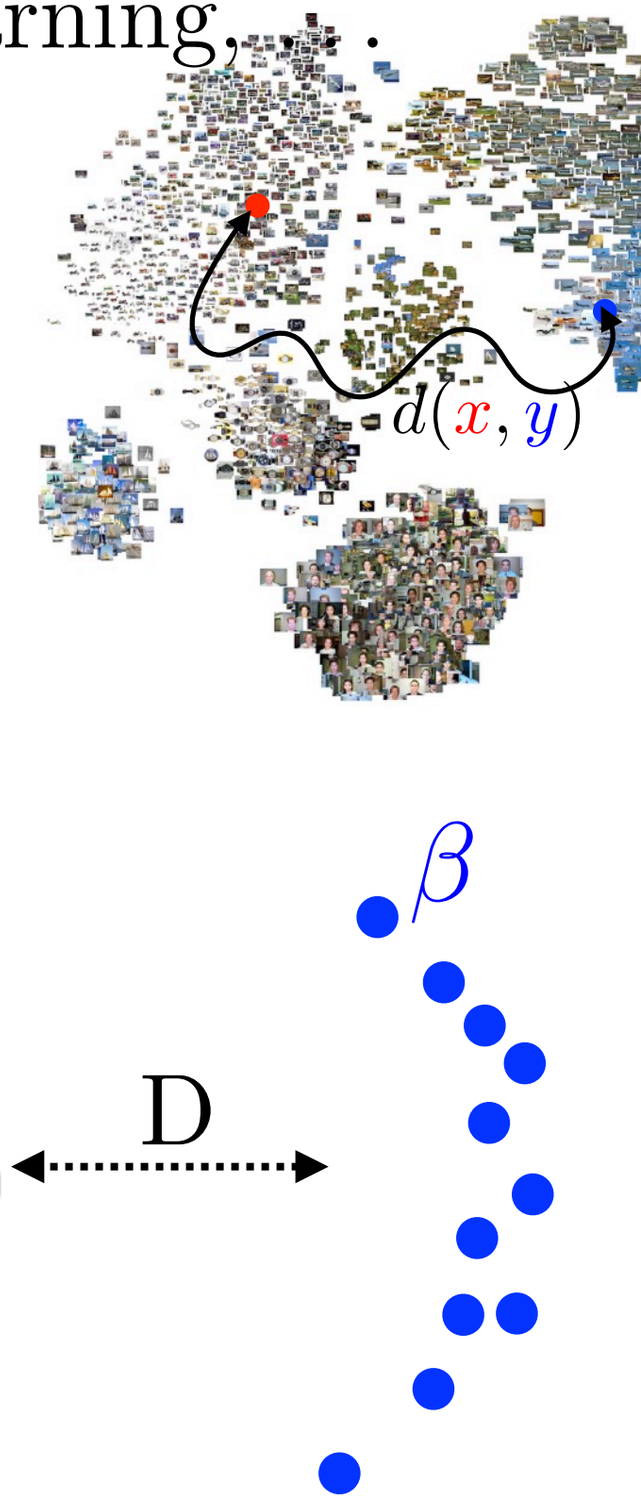
Probability distributions and histograms

$\rightarrow$ images, vision, graphics and machine learning, . . .









Unsupervised learning

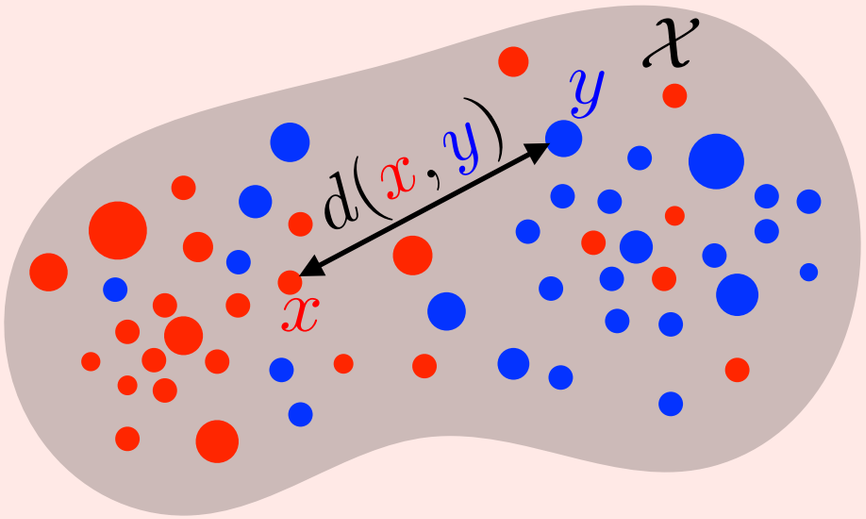
Observations: $\beta \stackrel{\text{def.}}{=} \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}$

Parametric model: $\theta \mapsto \alpha_\theta$

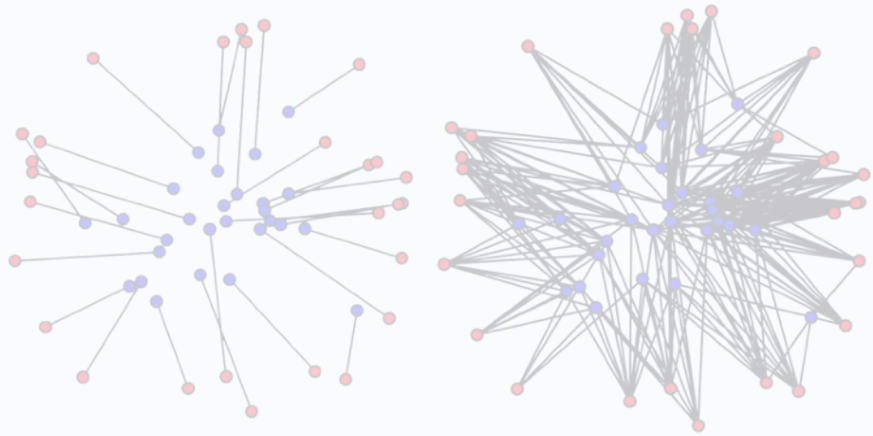
Density fitting: $\min_\theta \mathrm{D}(\alpha_\theta, \beta)$

$\rightarrow$ takes into account a metric d.

1. Optimal Transport

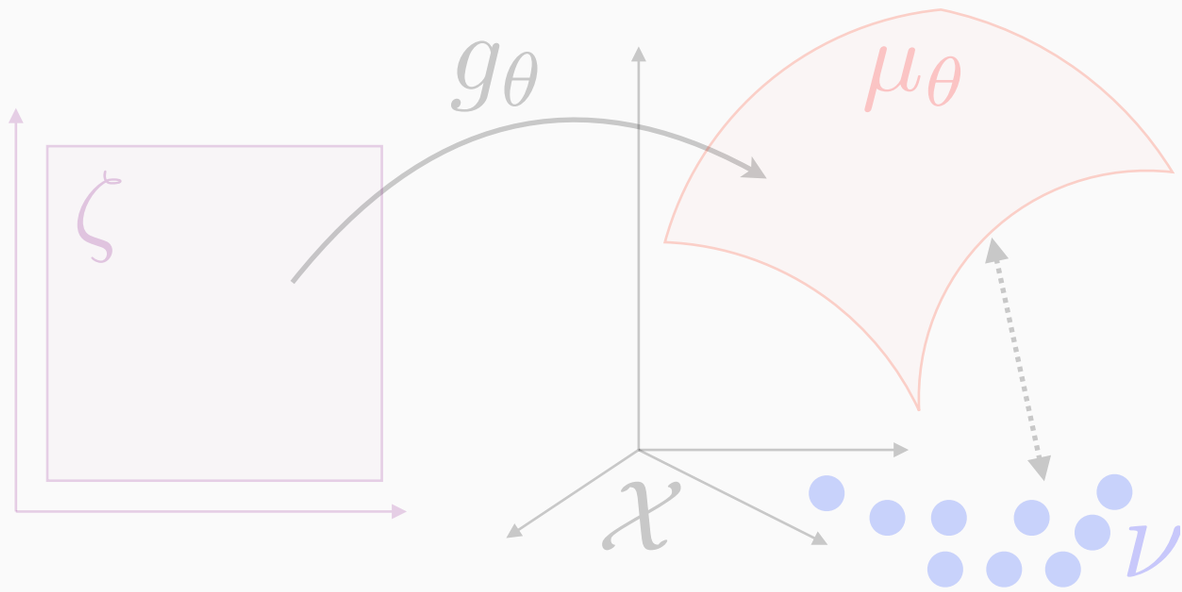


2. Entropic Regularization



3. Sinkhorn Divergences

4. Application to Generative Models

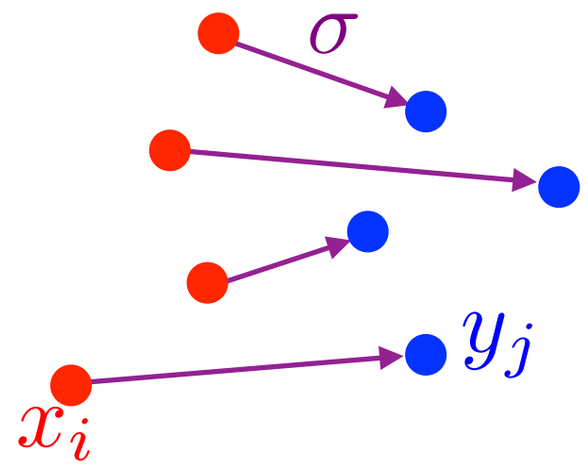


Monge's Problem

Points $(x_i)_i$, $(y_j)_j$

Permutation:

$$\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$$

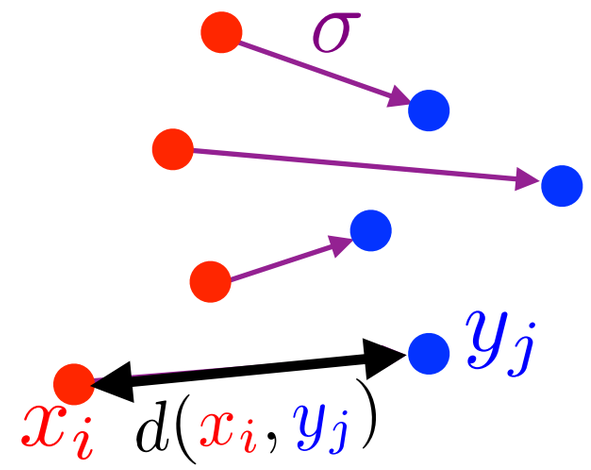


Monge's Problem

Points $(x_i)_i$, $(y_j)_j$

Permutation:

$$\sigma : \{1, \dots, n\} \to \{1, \dots, n\}$$



Monge optimal matching: $\min_{\sigma} \sum_{i=1}^{n} d(x_i, y_{\sigma(i)})$

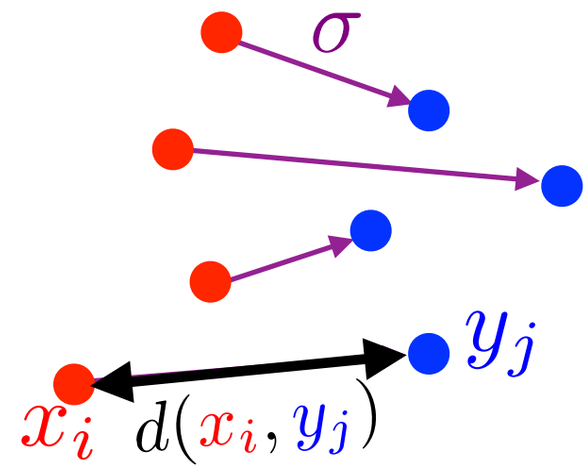
[Monge 1784]

Monge's Problem

Points $(x_i)_i$, $(y_j)_j$

Permutation:

$$\sigma : \{1, \dots, n\} \to \{1, \dots, n\}$$



Monge optimal matching: $\min_{\sigma} \sum_{i=1}^{n} d(x_i, y_{\sigma(i)})$

[Monge 1784]

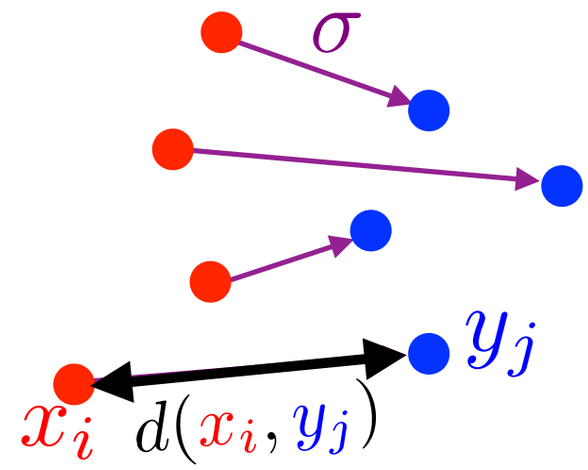
$\to$ Seems intractable: $n!$ possibilities.

Monge's Problem

Points $(x_i)_i$, $(y_j)_j$

Permutation:

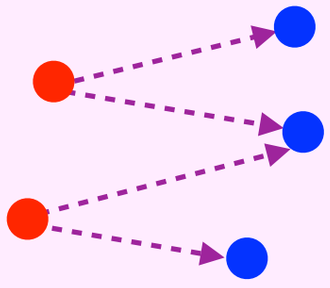
$$\sigma : \{1, \dots, n\} \to \{1, \dots, n\}$$



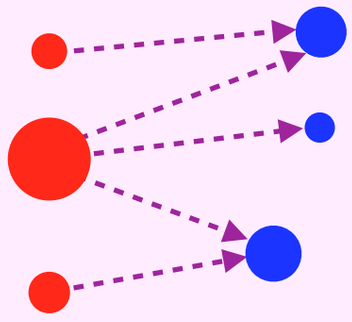
Monge optimal matching: $\min_{\sigma} \sum_{i=1}^{n} d(x_i, y_{\sigma(i)})$

[Monge 1784]

$\to$ Seems intractable: $n!$ possibilities.



Different # points?



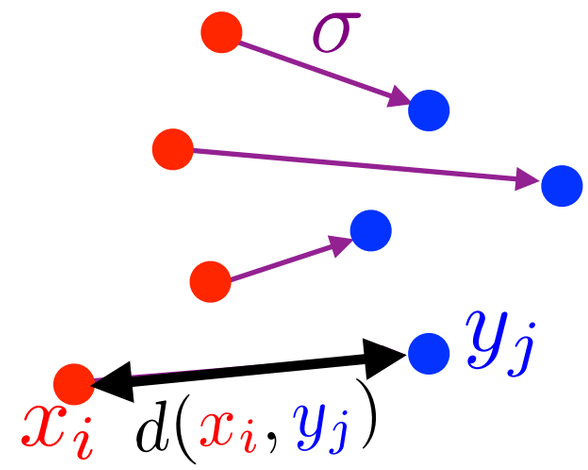
Weights?

Monge's Problem

Points $(x_i)_i$, $(y_j)_j$

Permutation:

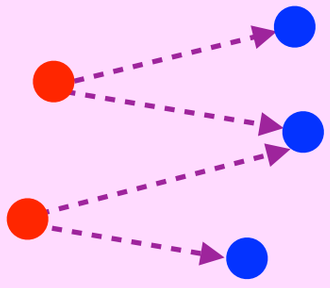
$$\sigma : \{1, \dots, n\} \to \{1, \dots, n\}$$



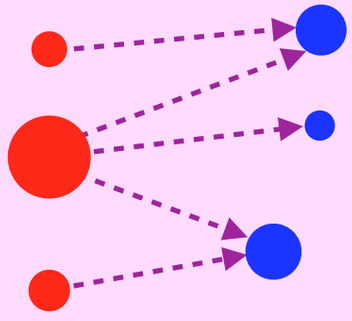
Monge optimal matching: $\min_{\sigma} \sum_{i=1}^{n} d(x_i, y_{\sigma(i)})$

[Monge 1784]

$\to$ Seems intractable: $n!$ possibilities.



Different # points?



Weights?

$\Longrightarrow$

"Relax"

points $\to$ mass

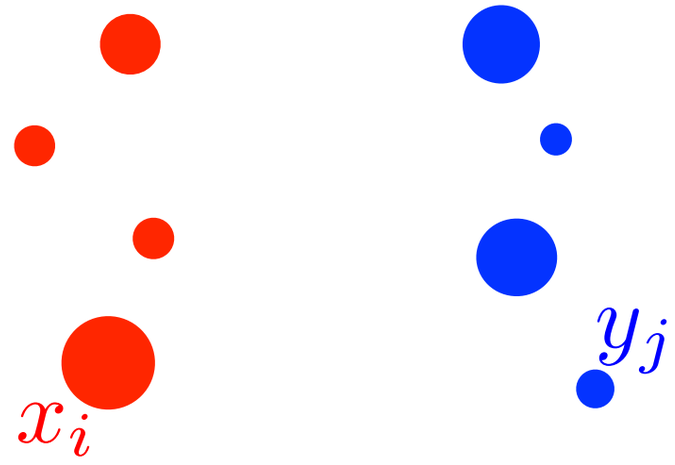
permutation $\to$ coupling

Kantorovitch's Formulation

Discrete distributions:

$$\alpha = \sum_{i=1}^{n} \mathbf{a}_i \delta_{x_i}$$

$$\beta = \sum_{j=1}^{m} \mathbf{b}_j \delta_{y_j}$$



Points $(x_i)_i$, $(y_j)_j$

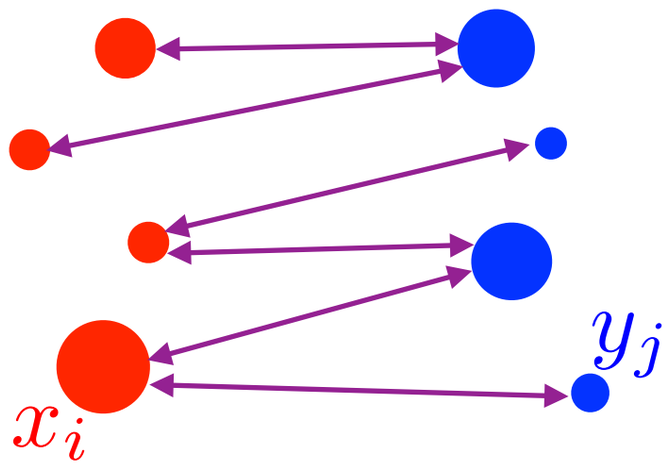
Weights $\mathbf{a}_i \geqslant 0$, $\mathbf{b}_j \geqslant 0$.

$\sum_{i=1}^{n} \mathbf{a}_i = \sum_{j=1}^{m} \mathbf{b}_j = 1$

Kantorovitch's Formulation

Discrete distributions: $\alpha = \sum_{i=1}^{n} \mathbf{a}_i \delta_{x_i}$

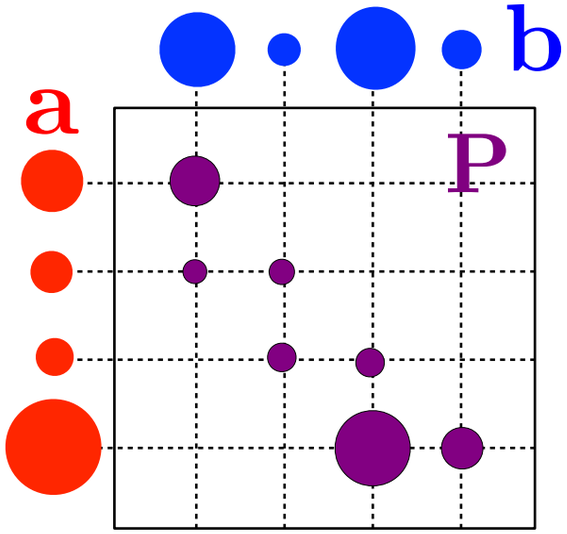
$\beta = \sum_{j=1}^{m} \mathbf{b}_j \delta_{y_j}$



Points $(x_i)_i$, $(y_j)_j$

Weights $\mathbf{a}_i \geqslant 0$, $\mathbf{b}_j \geqslant 0$.

$\sum_{i=1}^{n} \mathbf{a}_i = \sum_{j=1}^{m} \mathbf{b}_j = 1$



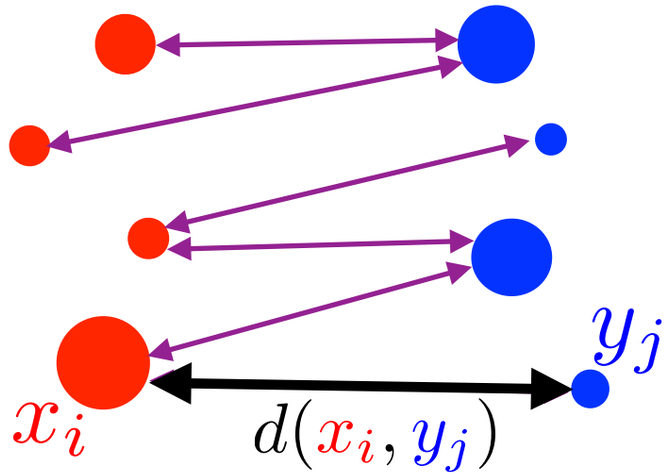
Couplings: $\sum_j \mathbf{P}_{i,j} = \mathbf{a}_i$ $\sum_i \mathbf{P}_{i,j} = \mathbf{b}_j$

$\mathbf{P} \geqslant 0$, $\mathbf{P}\mathbb{1}_m = \mathbf{a}$, $\mathbf{P}^\top \mathbb{1}_n = \mathbf{b}$

Kantorovitch's Formulation

Discrete distributions: $\alpha = \sum_{i=1}^{n} \mathbf{a}_i \delta_{x_i}$

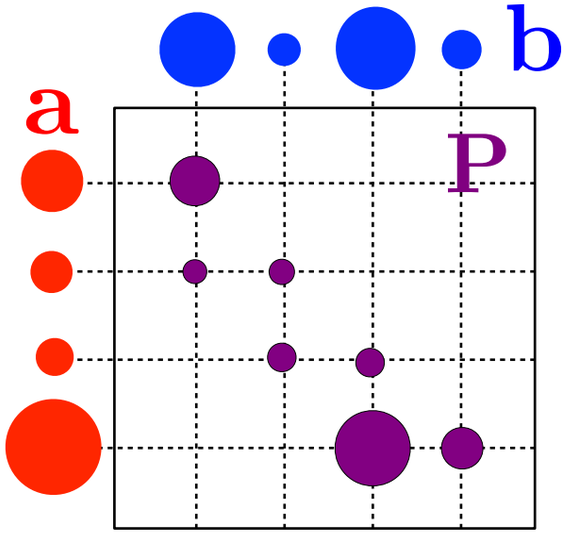
$\beta = \sum_{j=1}^{m} \mathbf{b}_j \delta_{y_j}$



Points $(x_i)_i$, $(y_j)_j$

Weights $\mathbf{a}_i \geqslant 0$, $\mathbf{b}_j \geqslant 0$.

$\sum_{i=1}^{n} \mathbf{a}_i = \sum_{j=1}^{m} \mathbf{b}_j = 1$



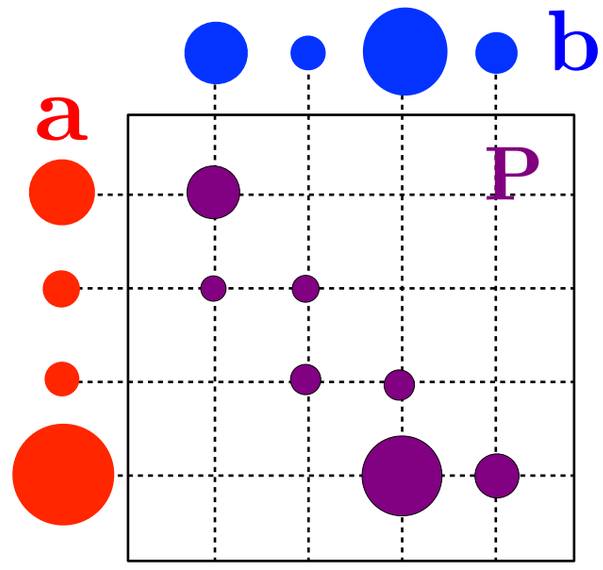
Couplings: $\sum_j \mathbf{P}_{i,j} = \mathbf{a}_i$ $\sum_i \mathbf{P}_{i,j} = \mathbf{b}_j$

[Kantorovich 1942]

$$\min_{\mathbf{P}} \left\{ \sum_{i,j} d(x_i, y_j)^p \mathbf{P}_{i,j} \;;\; \mathbf{P} \geqslant 0, \mathbf{P}\mathbb{1}_m = \mathbf{a}, \mathbf{P}^\top \mathbb{1}_n = \mathbf{b} \right\}$$

Optimal Transport Distances

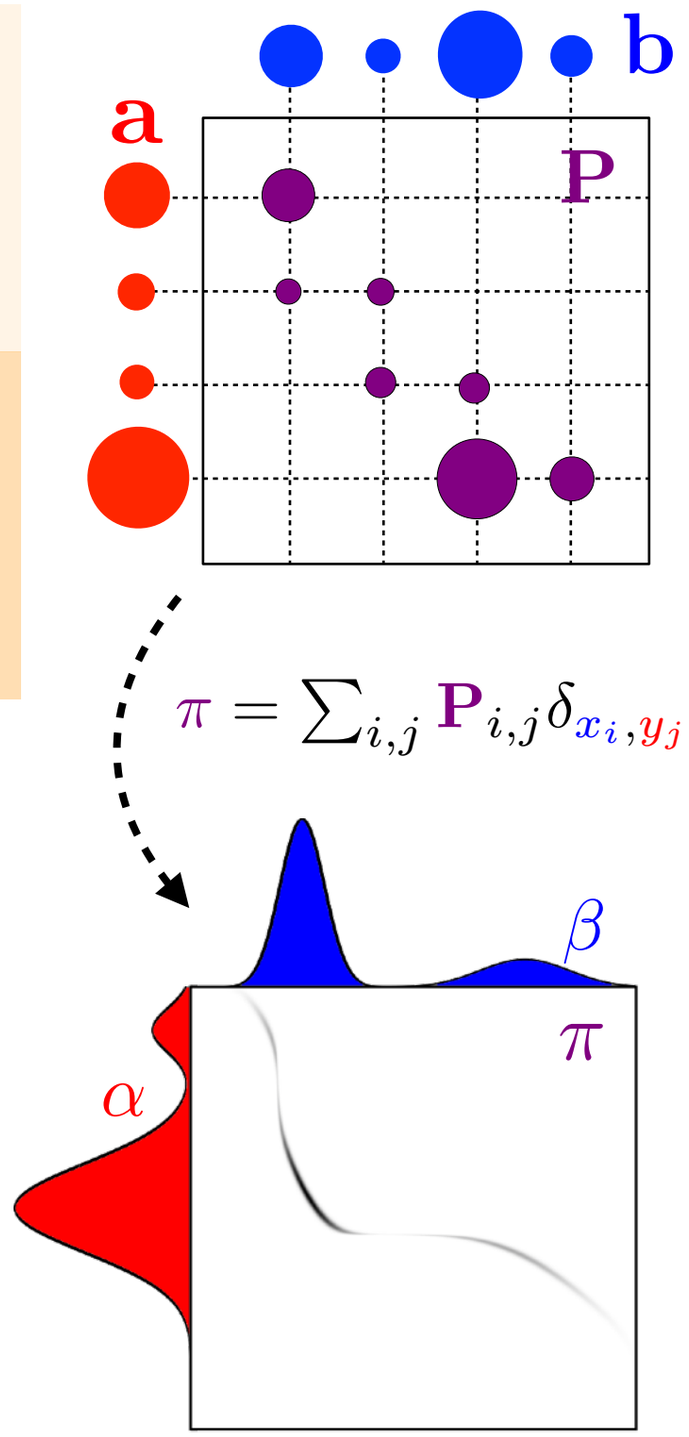
$$\mathrm{W}_p(\alpha,\beta) \overset{\text{def.}}{=} \left(\min_{\mathbf{P}\mathbb{1}=\mathbf{a},\mathbf{P}^\top\mathbb{1}=\mathbf{b}} \sum_{i,j} d(x_i,y_j)^p \mathbf{P}_{i,j} \right)^{\frac{1}{p}}$$



Optimal Transport Distances

$$\mathrm{W}_p(\alpha,\beta) \overset{\text{def.}}{=} \left(\min_{\mathbf{P}\mathbb{1}=\mathbf{a},\mathbf{P}^\top\mathbb{1}=\mathbf{b}} \sum_{i,j} d(x_i,y_j)^p \mathbf{P}_{i,j} \right)^{\frac{1}{p}}$$

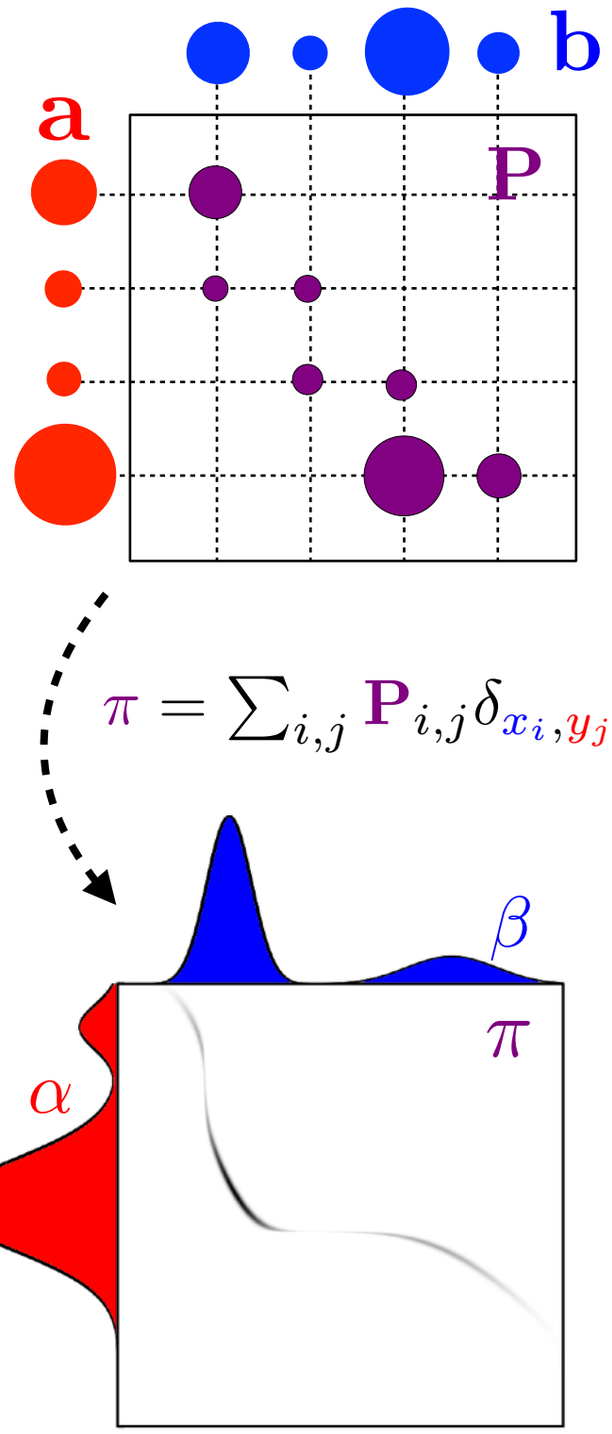
$$\min_{\pi\in\mathcal{M}_+^1(\mathcal{X}^2)} \left\{ \int_{\mathcal{X}^2} d(x,y)^p \mathrm{d}\pi(x,y) \;;\; \pi_1=\alpha, \pi_2=\beta \right\}$$



Optimal Transport Distances

$$\mathrm{W}_p(\alpha,\beta) \overset{\text{def.}}{=} \left(\min_{\mathbf{P}\mathbb{1}=\mathbf{a},\mathbf{P}^\top\mathbb{1}=\mathbf{b}} \sum_{i,j} d(x_i,y_j)^p \mathbf{P}_{i,j} \right)^{\frac{1}{p}}$$

$$\min_{\pi\in\mathcal{M}_+^1(\mathcal{X}^2)} \left\{ \int_{\mathcal{X}^2} d(x,y)^p \mathrm{d}\pi(x,y) \;;\; \pi_1=\alpha, \pi_2=\beta \right\}$$



Convergence in law: $\alpha_n \rightharpoonup \beta$

$\Leftrightarrow \quad \forall f \in \mathcal{C}(\mathcal{X}),\ \int_{\mathcal{X}} f \mathrm{d}\alpha_n \to \int_{\mathcal{X}} f \mathrm{d}\beta$

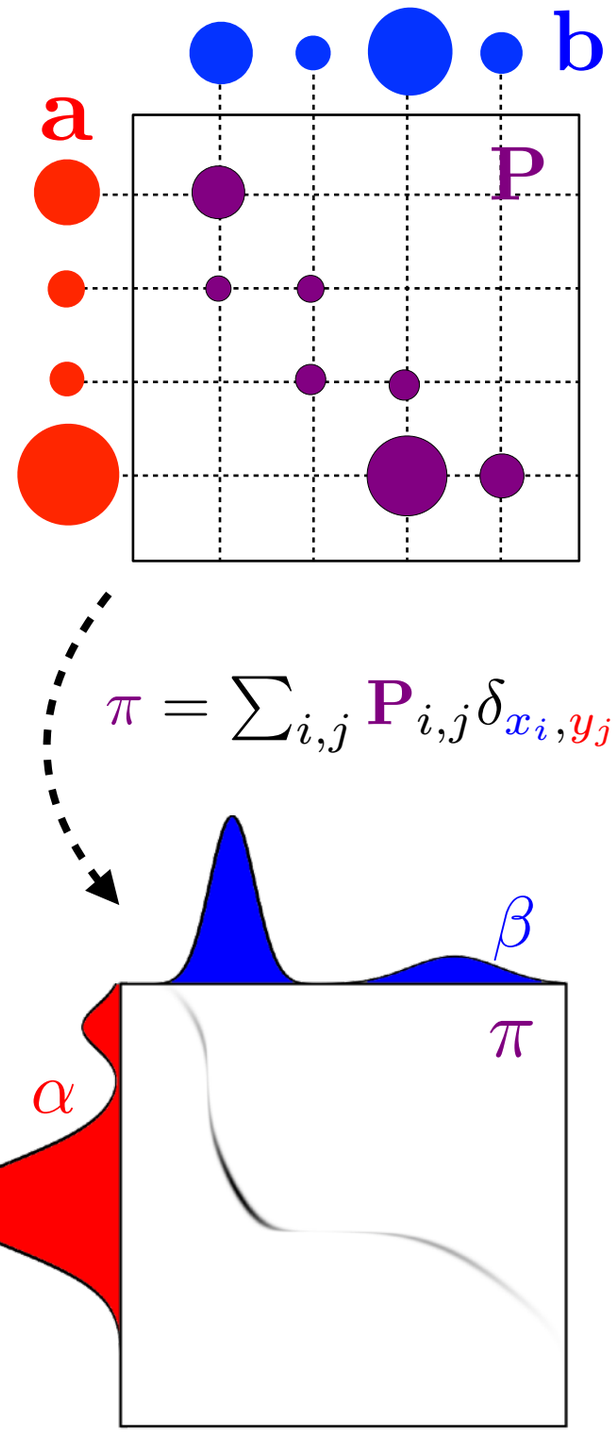
$\delta_{x_1} \quad \delta_{x_2} \quad \delta_{x_3} \quad \dots \quad \delta_x$

$\|\delta_{x_n} - \delta_x\|_1 = 2$ **vs.** $\mathrm{W}_p(\delta_{x_n},\delta_x) = d(x_n,x)$

Optimal Transport Distances

$$\mathrm{W}_p(\alpha,\beta) \overset{\text{def.}}{=} \left(\min_{\mathbf{P}\mathbb{1}=\mathbf{a},\mathbf{P}^\top\mathbb{1}=\mathbf{b}} \sum_{i,j} d(x_i,y_j)^p \mathbf{P}_{i,j} \right)^{\frac{1}{p}}$$

$$\min_{\pi\in\mathcal{M}_+^1(\mathcal{X}^2)} \left\{ \int_{\mathcal{X}^2} d(x,y)^p \mathrm{d}\pi(x,y) \;;\; \pi_1=\alpha, \pi_2=\beta \right\}$$



Convergence in law: $\alpha_n \rightharpoonup \beta$

$\Leftrightarrow \quad \forall\, f \in \mathcal{C}(\mathcal{X}),\ \int_{\mathcal{X}} f \mathrm{d}\alpha_n \to \int_{\mathcal{X}} f \mathrm{d}\beta$

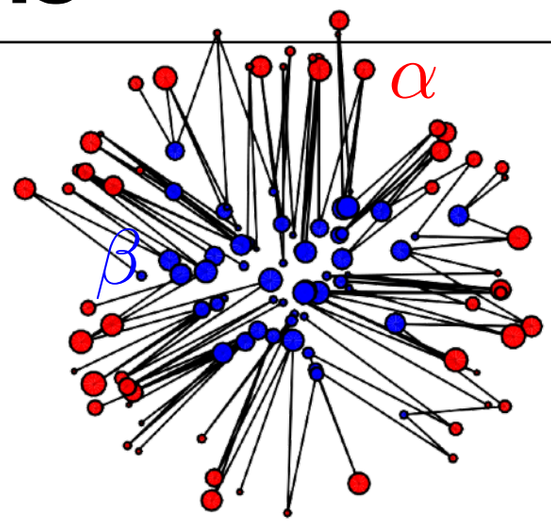
$\delta_{x_1} \quad \delta_{x_2} \quad \delta_{x_3} \quad \dots \quad \delta_x$

$\|\delta_{x_n} - \delta_x\|_1 = 2$ **vs.** $\mathrm{W}_p(\delta_{x_n},\delta_x) = d(x_n,x)$

Theorem: W_p is a distance and $\alpha_n \rightharpoonup \beta \Leftrightarrow \mathrm{W}_p(\alpha_n,\beta) \to 0$

Algorithms

Linear programming: $O(n^3 \log(n)^2)$

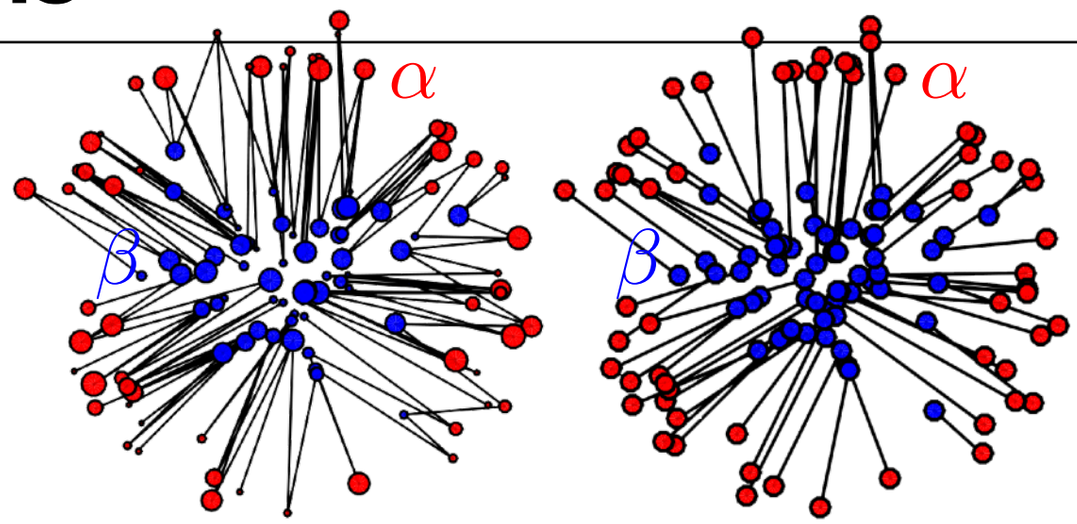


Algorithms

Linear programming: $O(n^3 \log(n)^2)$

Hungarian/Auction: $O(n^3)$

$\alpha = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}$ $\beta = \frac{1}{n}\sum_{j=1}^{n} \delta_{y_j}$



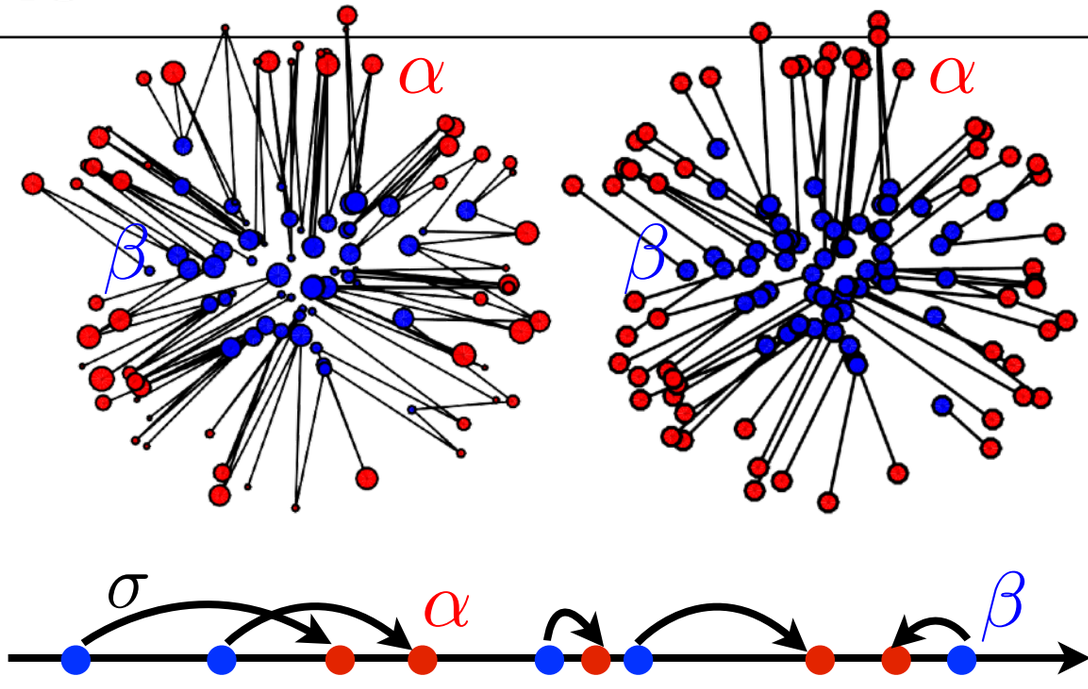
Algorithms

Linear programming: $O(n^3 \log(n)^2)$

Hungarian/Auction: $O(n^3)$

$\alpha = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}$ $\beta = \frac{1}{n}\sum_{j=1}^{n} \delta_{y_j}$

1-D case: sorting $O(n \log(n))$.

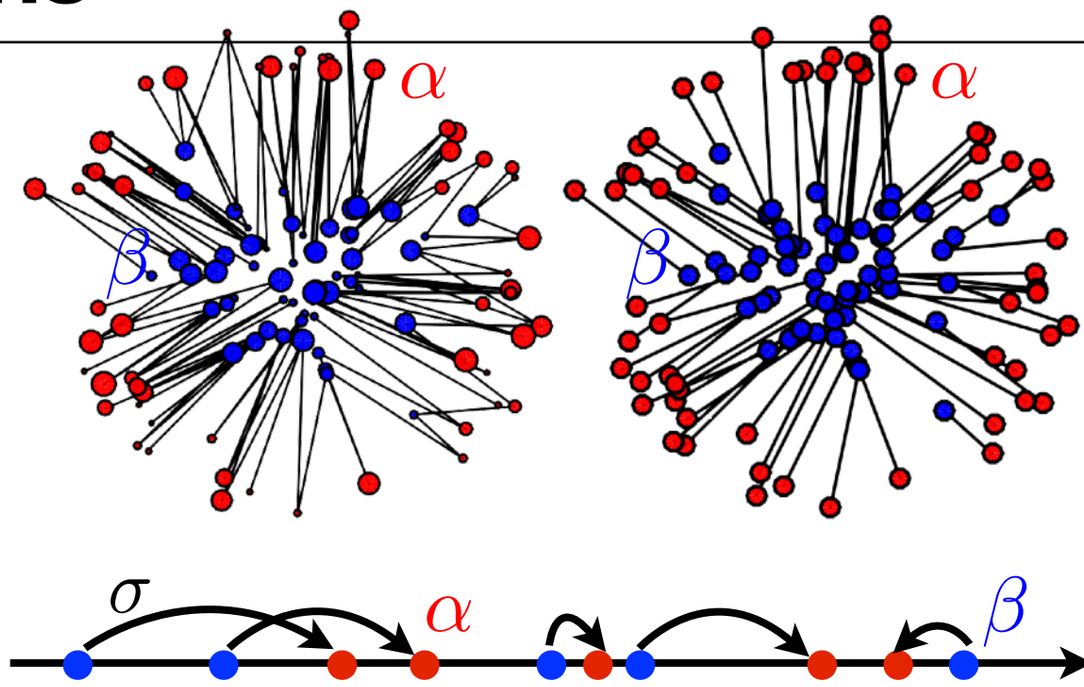


Algorithms

Linear programming: $O(n^3 \log(n)^2)$

Hungarian/Auction: $O(n^3)$

$\alpha = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$ $\beta = \frac{1}{n}\sum_{j=1}^n \delta_{y_j}$



1-D case: sorting $O(n \log(n))$.

$p = 1$
$d = \|\cdot\|$

$$W_1(\alpha, \beta) = \min_{\mathrm{div}(u)=\alpha-\beta} \int \|u(x)\| \mathrm{d}x$$

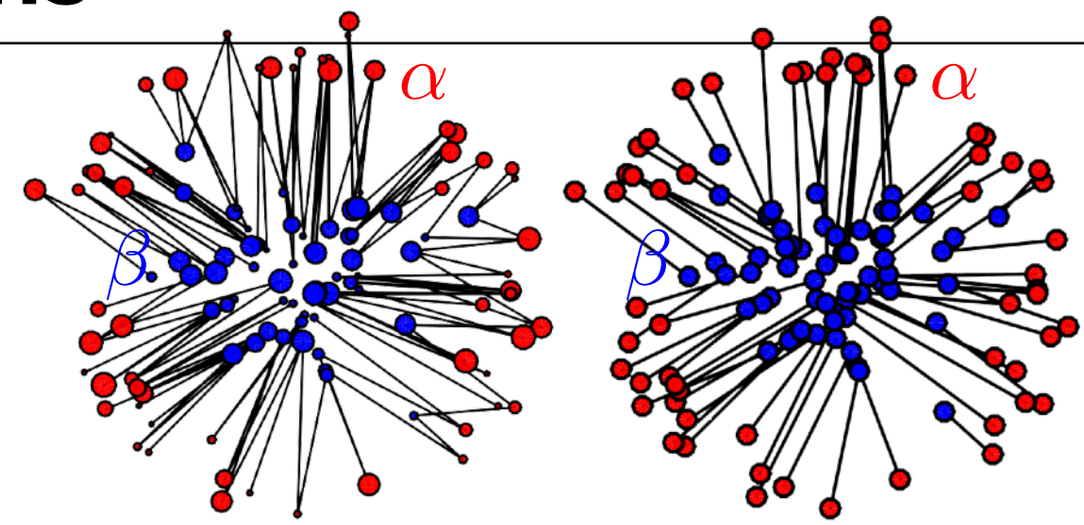
$\rightarrow$ min-cost flow, on graphs $O(n^2 \log(n))$.

Algorithms

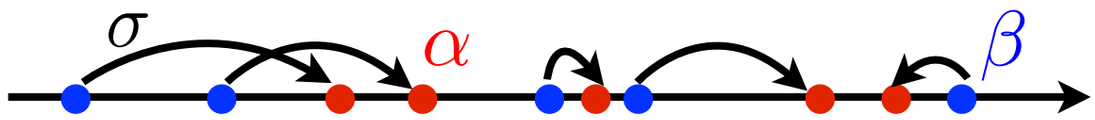
Linear programming: $O(n^3 \log(n)^2)$

Hungarian/Auction: $O(n^3)$

$\alpha = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$ $\beta = \frac{1}{n}\sum_{j=1}^n \delta_{y_j}$



1-D case: sorting $O(n \log(n))$.



$p = 1$
$d = \|\cdot\|$ $\quad W_1(\alpha,\beta) = \min_{\mathrm{div}(u)=\alpha-\beta} \int \|u(x)\| \mathrm{d}x$

$\rightarrow$ min-cost flow, on graphs $O(n^2 \log(n))$.

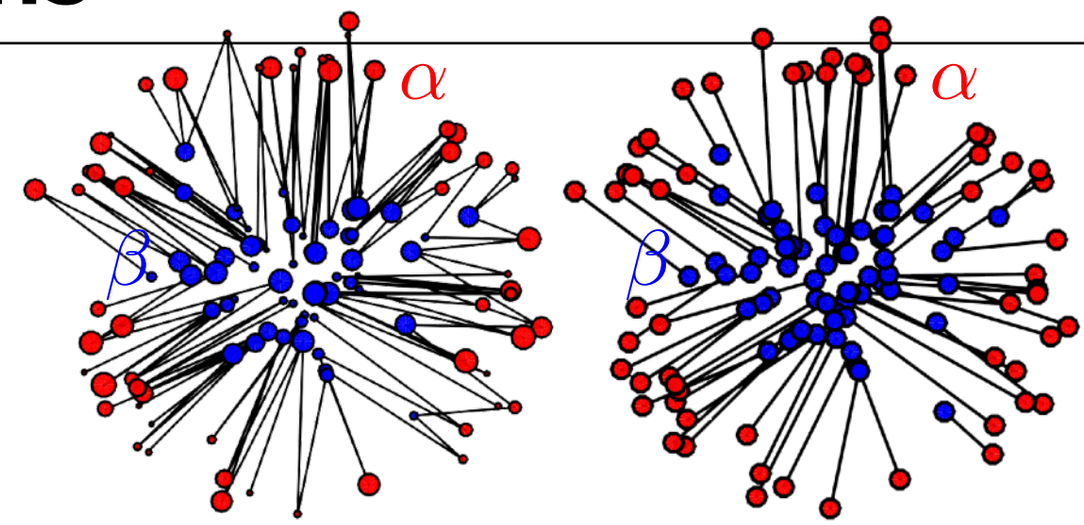
Monge-Ampère/Benamou-Brenier, $d = \|\cdot\|_2^2$.

Algorithms

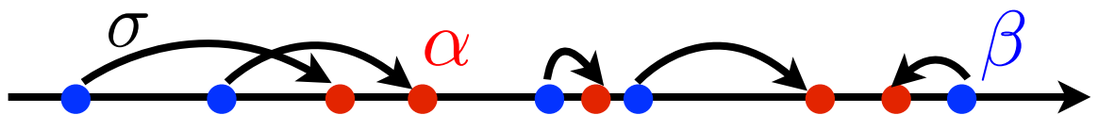
Linear programming: $O(n^3 \log(n)^2)$

Hungarian/Auction: $O(n^3)$

$\alpha = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$ $\beta = \frac{1}{n}\sum_{j=1}^n \delta_{y_j}$



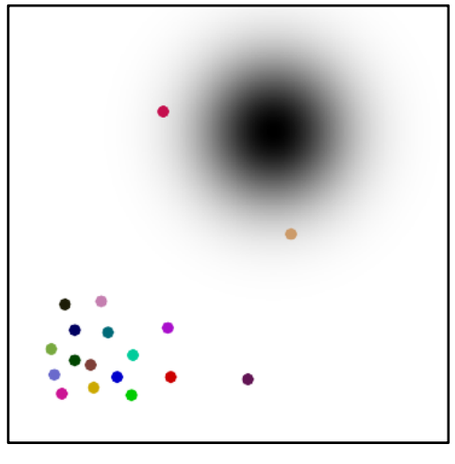
1-D case: sorting $O(n \log(n))$.



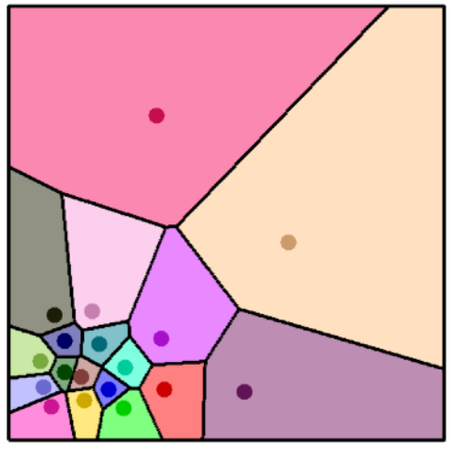
$p = 1$
$d = \|\cdot\|$ $\quad W_1(\alpha,\beta) = \min_{\mathrm{div}(u)=\alpha-\beta} \int \|u(x)\| \mathrm{d}x$

$\rightarrow$ min-cost flow, on graphs $O(n^2 \log(n))$.

Monge-Ampère/Benamou-Brenier, $d = \|\cdot\|_2^2$.



Semi-discrete: Laguerre cells, $d = \|\cdot\|_2^2$.
[Merigot 2013]

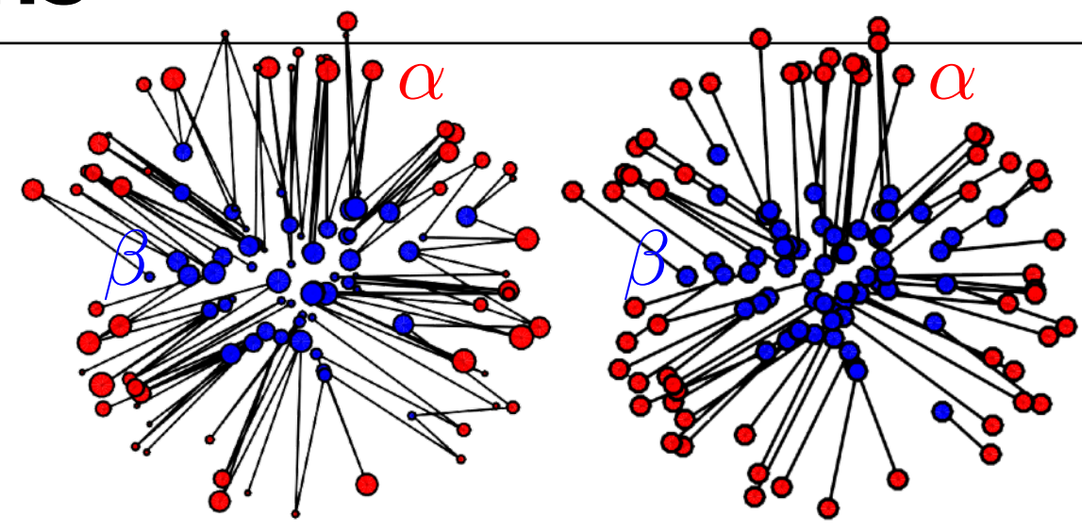


Algorithms

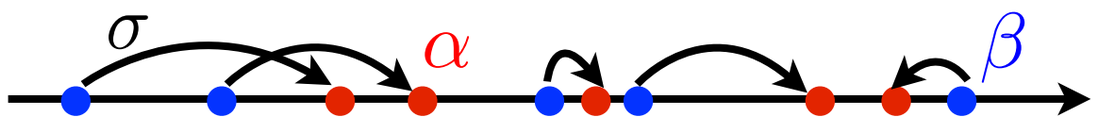
Linear programming: $O(n^3 \log(n)^2)$

Hungarian/Auction: $O(n^3)$

$\alpha = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$ $\beta = \frac{1}{n}\sum_{j=1}^n \delta_{y_j}$



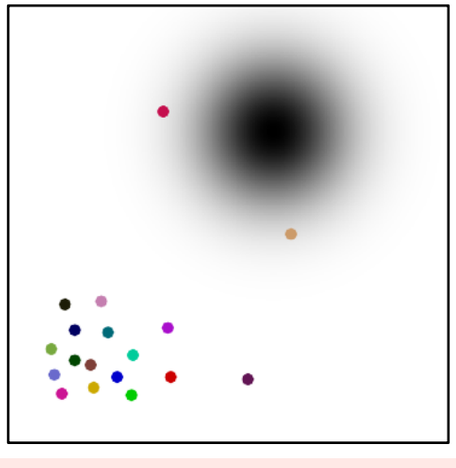
1-D case: sorting $O(n \log(n))$.



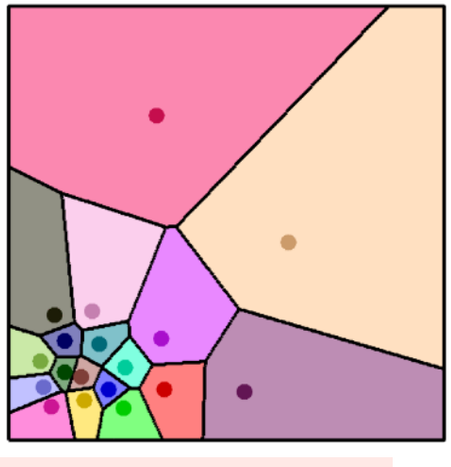
$p = 1$
$d = \|\cdot\|$ $W_1(\alpha,\beta) = \min_{\mathrm{div}(u)=\alpha-\beta} \int \|u(x)\| \mathrm{d}x$

$\rightarrow$ min-cost flow, on graphs $O(n^2 \log(n))$.

Monge-Ampère/Benamou-Brenier, $d = \|\cdot\|_2^2$.

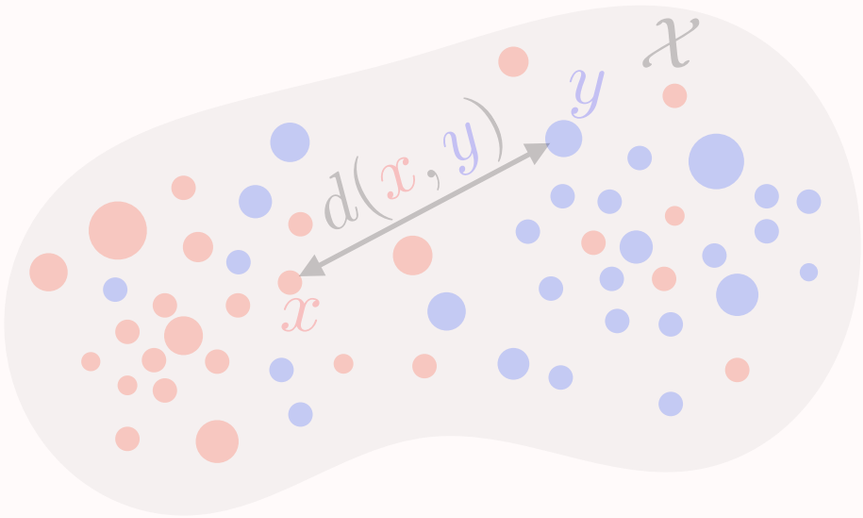


Semi-discrete: Laguerre cells, $d = \|\cdot\|_2^2$.
[Merigot 2013]

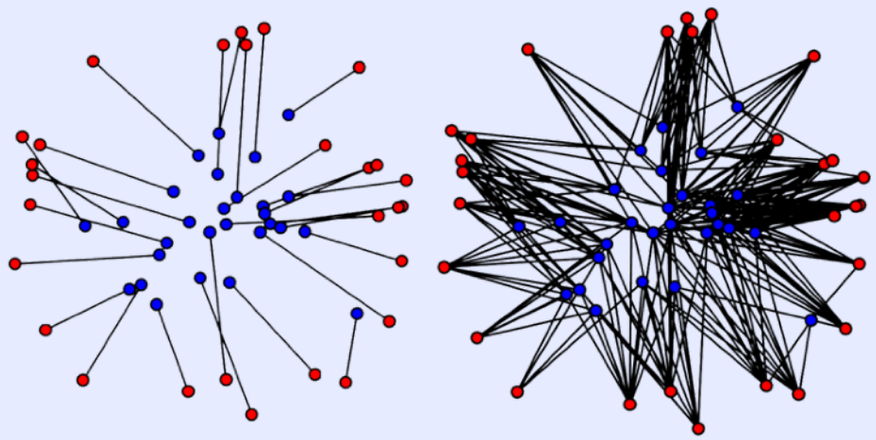


Need for fast approximate algorithms for generic $d(x,y)$.

1. Optimal Transport

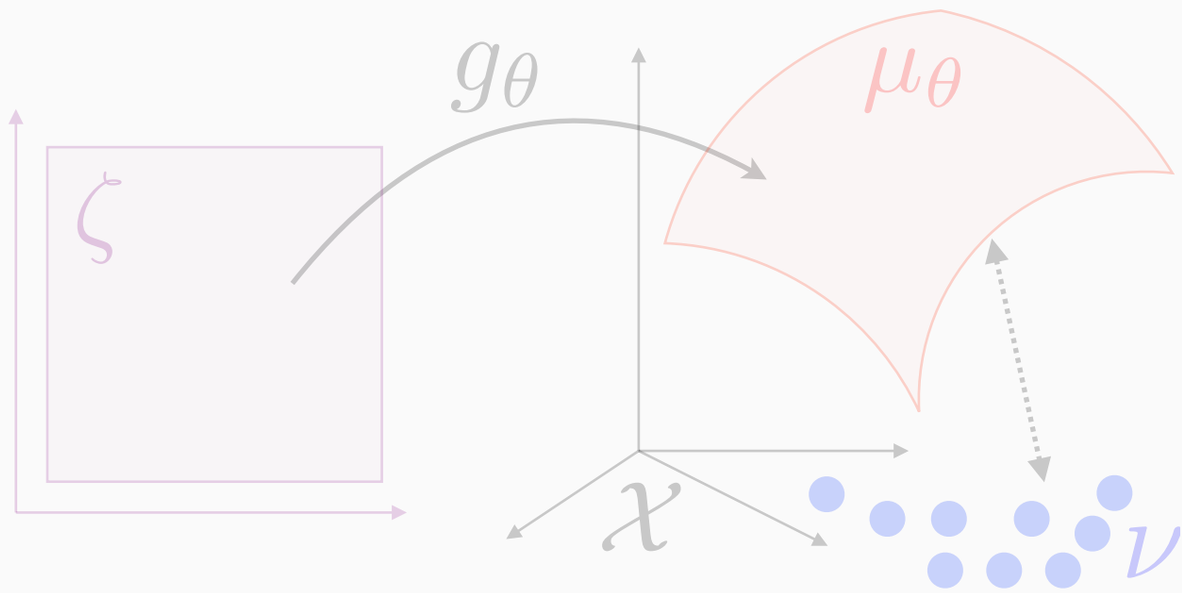


2. Entropic Regularization



3. Sinkhorn Divergences

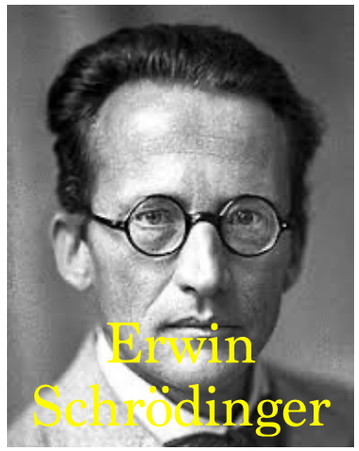
4. Application to Generative Models



Entropic Regularization

Schrödinger's problem: [1931]

$$\min_{\mathbf{P}\mathbb{1}=\mathbf{a},\mathbf{P}^\top\mathbb{1}=\mathbf{b}} \sum_{i,j} d(x_i, y_j)^p \mathbf{P}_{i,j} + \varepsilon \mathbf{P}_{i,j} \log(\mathbf{P}_{i,j})$$

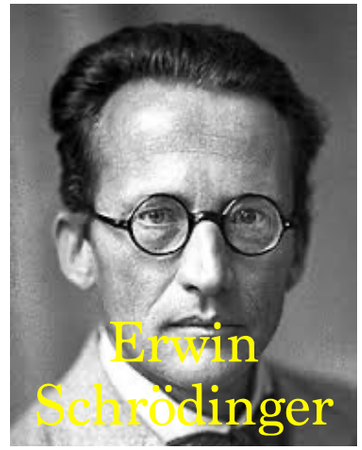


$$\pi = \sum_{i,j} \mathbf{P}_{i,j} \delta_{x_i, y_j}$$

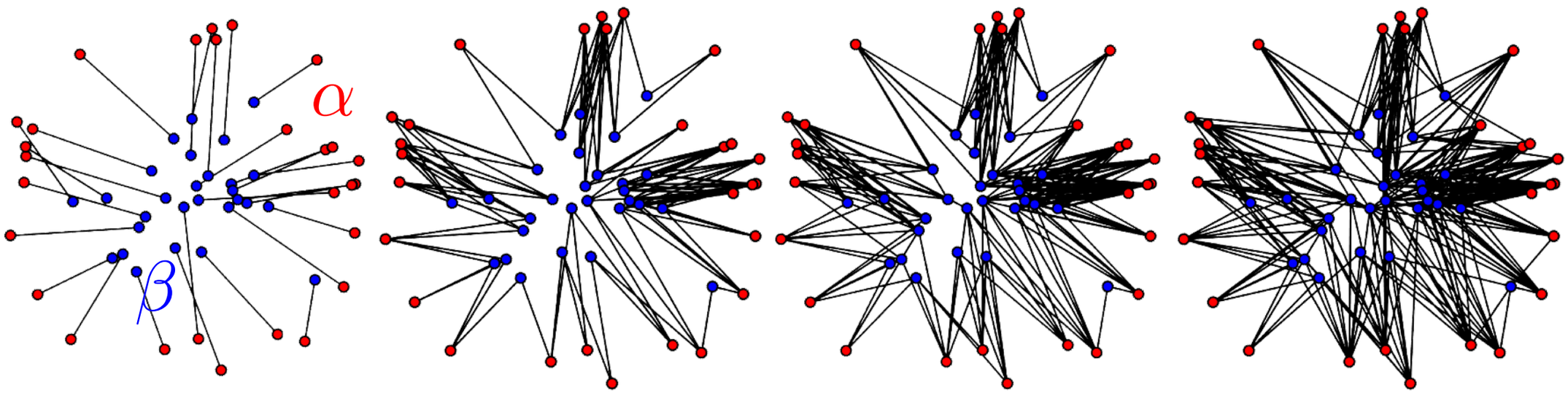
Entropic Regularization

Schrödinger's problem: [1931]

$$\min_{\mathbf{P}\mathbb{1}=\mathbf{a},\mathbf{P}^\top\mathbb{1}=\mathbf{b}} \sum_{i,j} d(x_i,y_j)^p \mathbf{P}_{i,j} + \varepsilon \mathbf{P}_{i,j}\log(\mathbf{P}_{i,j})$$



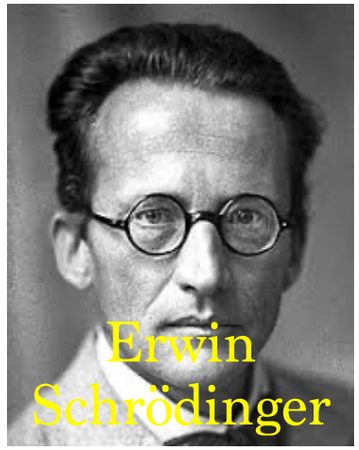
$$\pi = \sum_{i,j} \mathbf{P}_{i,j}\delta_{x_i,y_j}$$



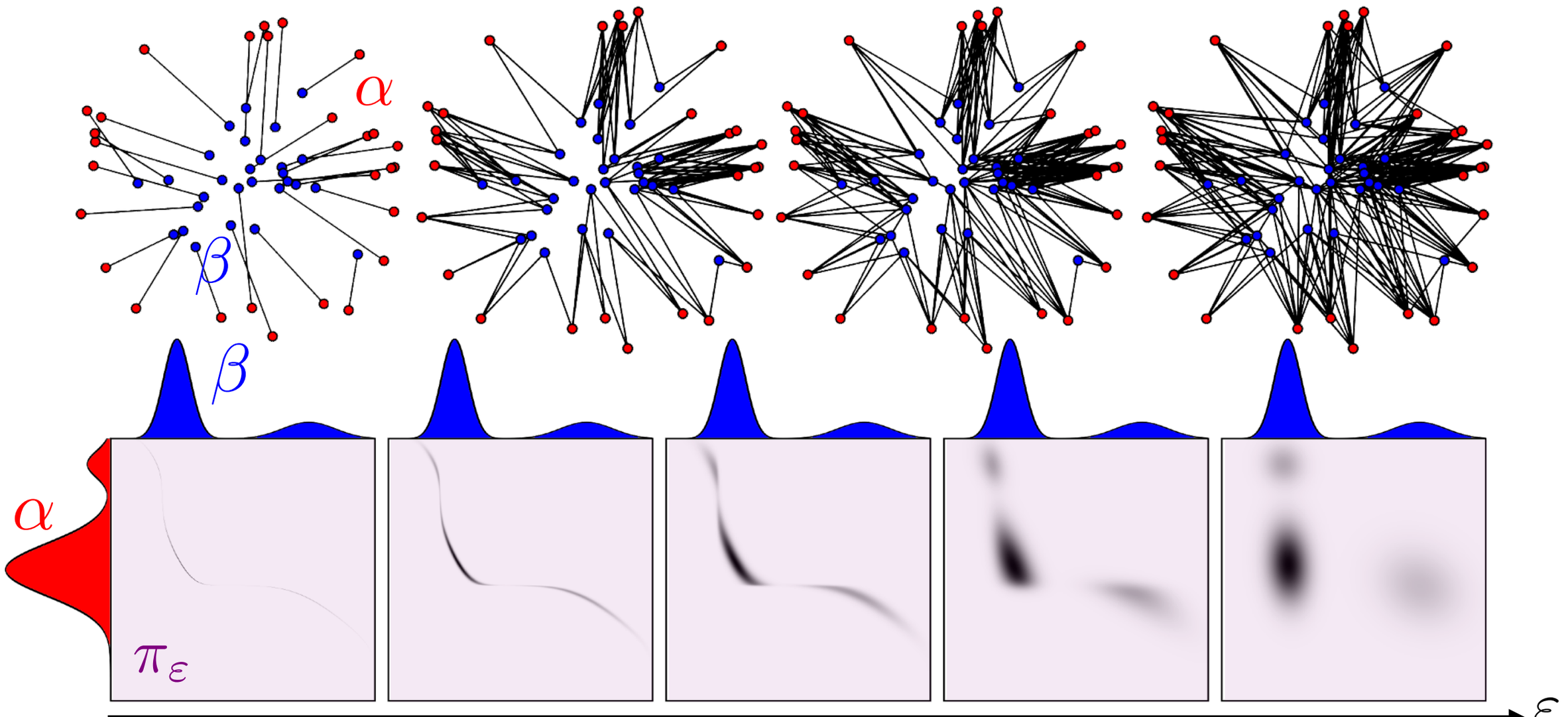
Entropic Regularization

Schrödinger's problem: [1931]

$$\min_{\mathbf{P}\mathbb{1}=\mathbf{a},\mathbf{P}^\top\mathbb{1}=\mathbf{b}} \sum_{i,j} d(x_i,y_j)^p \mathbf{P}_{i,j} + \varepsilon \mathbf{P}_{i,j}\log(\mathbf{P}_{i,j})$$



$$\pi = \sum_{i,j} \mathbf{P}_{i,j}\delta_{x_i,y_j}$$



Sinkhorn's Algorithm

$$\min_{\mathbf{P}} \left\{ \sum_{i,j} d(x_i, y_j)^p \mathbf{P}_{i,j} + \varepsilon \mathbf{P}_{i,j} \log(\mathbf{P}_{i,j}) \; ; \; \mathbf{P}\mathbb{1} = \mathbf{a}, \mathbf{P}^\top \mathbb{1} = \mathbf{b} \right\}$$

Proposition:

$$\mathbf{P} \text{ solution} \Leftrightarrow \begin{cases} \mathbf{P}\mathbb{1} = \mathbf{a}, \mathbf{P}^\top \mathbb{1} = \mathbf{b} \text{ and} \\ \exists \mathbf{u}, \mathbf{v}, \;\; \mathbf{P}_{i,j} = \mathbf{u}_i \, \mathbf{K}_{i,j} \, \mathbf{v}_j \end{cases}$$

$$\mathbf{K}_{i,j} \overset{\text{def.}}{=} e^{-\frac{d(x_i, y_j)^p}{\varepsilon}}$$

Sinkhorn's Algorithm

$$\min_{\mathbf{P}} \left\{ \sum_{i,j} d(x_i, y_j)^p \mathbf{P}_{i,j} + \varepsilon \mathbf{P}_{i,j} \log(\mathbf{P}_{i,j}) \;;\; \mathbf{P}\mathbb{1} = \mathbf{a}, \mathbf{P}^\top \mathbb{1} = \mathbf{b} \right\}$$

Proposition:

$$\mathbf{P} \text{ solution} \Leftrightarrow \begin{cases} \mathbf{P}\mathbb{1} = \mathbf{a}, \mathbf{P}^\top \mathbb{1} = \mathbf{b} \text{ and} \\ \exists \mathbf{u}, \mathbf{v}, \;\; \mathbf{P}_{i,j} = \mathbf{u}_i \, \mathbf{K}_{i,j} \, \mathbf{v}_j \end{cases}$$

$$\mathbf{K}_{i,j} \overset{\text{def.}}{=} e^{-\frac{d(x_i, y_j)^p}{\varepsilon}}$$

$$\mathbf{P} = \operatorname{diag}(\mathbf{u}) \mathbf{K} \operatorname{diag}(\mathbf{v}) \quad \Longrightarrow \quad \mathbf{a} = \mathbf{P}\mathbb{1} = \operatorname{diag}(\mathbf{u})(\mathbf{K}\mathbf{v}) = \mathbf{u} \odot (\mathbf{K}\mathbf{v})$$

Sinkhorn's Algorithm

$$\min_{\mathbf{P}} \left\{ \sum_{i,j} d(x_i, y_j)^p \mathbf{P}_{i,j} + \varepsilon \mathbf{P}_{i,j} \log(\mathbf{P}_{i,j}) \;;\; \mathbf{P}\mathbb{1} = \mathbf{a}, \mathbf{P}^\top \mathbb{1} = \mathbf{b} \right\}$$

Proposition:

$$\mathbf{P} \text{ solution} \Leftrightarrow \begin{cases} \mathbf{P}\mathbb{1} = \mathbf{a}, \mathbf{P}^\top \mathbb{1} = \mathbf{b} \text{ and} \\ \exists \mathbf{u}, \mathbf{v}, \;\; \mathbf{P}_{i,j} = \mathbf{u}_i \, \mathbf{K}_{i,j} \, \mathbf{v}_j \end{cases}$$

$$\mathbf{K}_{i,j} \overset{\text{def.}}{=} e^{-\frac{d(x_i, y_j)^p}{\varepsilon}}$$

$$\mathbf{P} = \mathrm{diag}(\mathbf{u}) \mathbf{K} \, \mathrm{diag}(\mathbf{v}) \quad \Longrightarrow \quad \mathbf{a} = \mathbf{P}\mathbb{1} = \mathrm{diag}(\mathbf{u})(\mathbf{K}\mathbf{v}) = \mathbf{u} \odot (\mathbf{K}\mathbf{v})$$

Row constraint: $\mathbf{u} \odot (\mathbf{K}\mathbf{v}) = \mathbf{a}$

Col. constraint: $\mathbf{v} \odot (\mathbf{K}^\top \mathbf{u}) = \mathbf{b}$

Sinkhorn's Algorithm

$$\min_{\mathbf{P}} \left\{ \sum_{i,j} d(x_i, y_j)^p \mathbf{P}_{i,j} + \varepsilon \mathbf{P}_{i,j} \log(\mathbf{P}_{i,j}) \;;\; \mathbf{P}\mathbb{1} = \mathbf{a}, \mathbf{P}^\top \mathbb{1} = \mathbf{b} \right\}$$

Proposition:

$$\mathbf{P} \text{ solution} \Leftrightarrow \begin{cases} \mathbf{P}\mathbb{1} = \mathbf{a}, \mathbf{P}^\top \mathbb{1} = \mathbf{b} \text{ and} \\ \exists \mathbf{u}, \mathbf{v}, \;\; \mathbf{P}_{i,j} = \mathbf{u}_i \, \mathbf{K}_{i,j} \, \mathbf{v}_j \end{cases}$$

$$\mathbf{K}_{i,j} \overset{\text{def.}}{=} e^{-\frac{d(x_i, y_j)^p}{\varepsilon}}$$

$$\mathbf{P} = \mathrm{diag}(\mathbf{u}) \mathbf{K} \, \mathrm{diag}(\mathbf{v}) \quad \Longrightarrow \quad \mathbf{a} = \mathbf{P}\mathbb{1} = \mathrm{diag}(\mathbf{u})(\mathbf{K}\mathbf{v}) = \mathbf{u} \odot (\mathbf{K}\mathbf{v})$$

Row constraint: $\mathbf{u} \odot (\mathbf{K}\mathbf{v}) = \mathbf{a}$

Col. constraint: $\mathbf{v} \odot (\mathbf{K}^\top \mathbf{u}) = \mathbf{b}$

Sinkhorn iterations:

$$\mathbf{u} \leftarrow \frac{\mathbf{a}}{\mathbf{K}\mathbf{v}} \qquad \mathbf{v} \leftarrow \frac{\mathbf{b}}{\mathbf{K}^\top \mathbf{u}}$$

Theorem: [Sinkhorn 1964] $(\mathbf{u}, \mathbf{v})$ converges.

Sinkhorn's Algorithm

$$\min_{\mathbf{P}} \left\{ \sum_{i,j} d(x_i, y_j)^p \mathbf{P}_{i,j} + \varepsilon \mathbf{P}_{i,j} \log(\mathbf{P}_{i,j}) \;;\; \mathbf{P}\mathbb{1} = \mathbf{a}, \mathbf{P}^\top \mathbb{1} = \mathbf{b} \right\}$$

Proposition:

$$\mathbf{P} \text{ solution} \Leftrightarrow \begin{cases} \mathbf{P}\mathbb{1} = \mathbf{a}, \mathbf{P}^\top \mathbb{1} = \mathbf{b} \text{ and} \\ \exists \mathbf{u}, \mathbf{v}, \;\; \mathbf{P}_{i,j} = \mathbf{u}_i \, \mathbf{K}_{i,j} \, \mathbf{v}_j \end{cases}$$

$$\mathbf{K}_{i,j} \overset{\text{def.}}{=} e^{-\frac{d(x_i, y_j)^p}{\varepsilon}}$$

$$\mathbf{P} = \mathrm{diag}(\mathbf{u}) \mathbf{K} \, \mathrm{diag}(\mathbf{v}) \quad \Longrightarrow \quad \mathbf{a} = \mathbf{P}\mathbb{1} = \mathrm{diag}(\mathbf{u})(\mathbf{K}\mathbf{v}) = \mathbf{u} \odot (\mathbf{K}\mathbf{v})$$

Row constraint: $\mathbf{u} \odot (\mathbf{K}\mathbf{v}) = \mathbf{a}$

Col. constraint: $\mathbf{v} \odot (\mathbf{K}^\top \mathbf{u}) = \mathbf{b}$

Sinkhorn iterations: $\mathbf{u} \leftarrow \dfrac{\mathbf{a}}{\mathbf{K}\mathbf{v}}$ $\mathbf{v} \leftarrow \dfrac{\mathbf{b}}{\mathbf{K}^\top \mathbf{u}}$

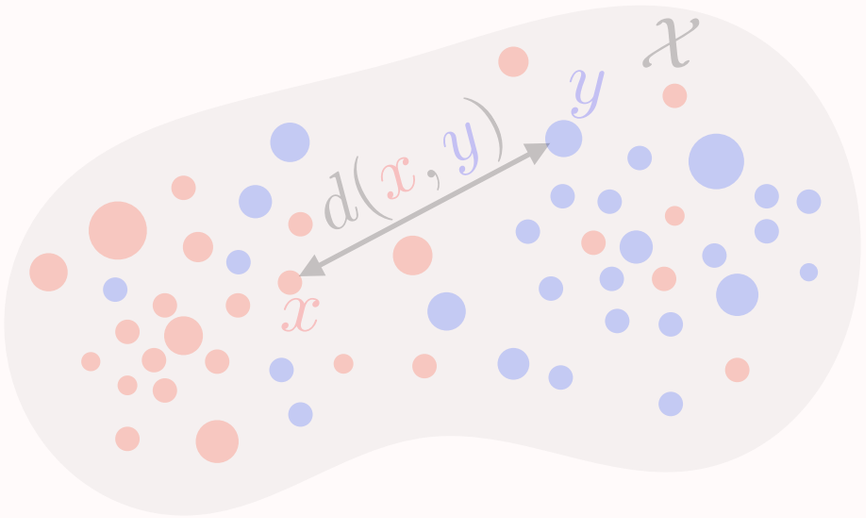
Theorem: [Sinkhorn 1964] $(\mathbf{u}, \mathbf{v})$ converges.

Matrix/vector multiplications: $\rightarrow O(n^2/\varepsilon^2)$ complexity.

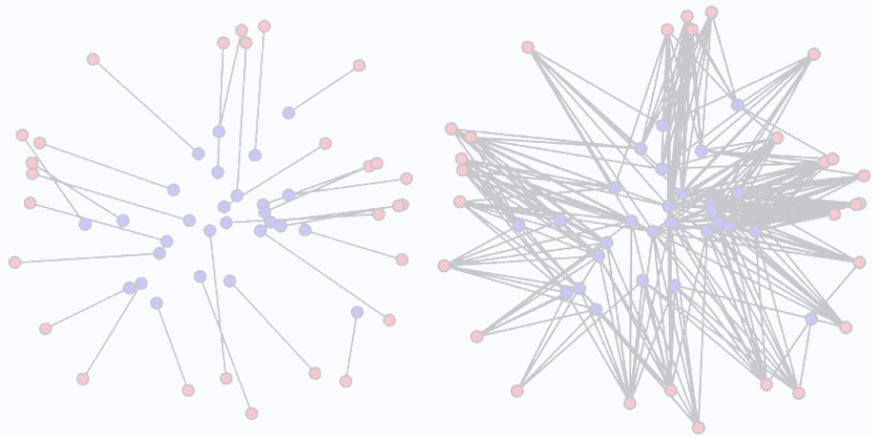
$\rightarrow$ Parallelizable on GPUs.

$\rightarrow$ Convolution on regular grids, separable kernels.

1. Optimal Transport

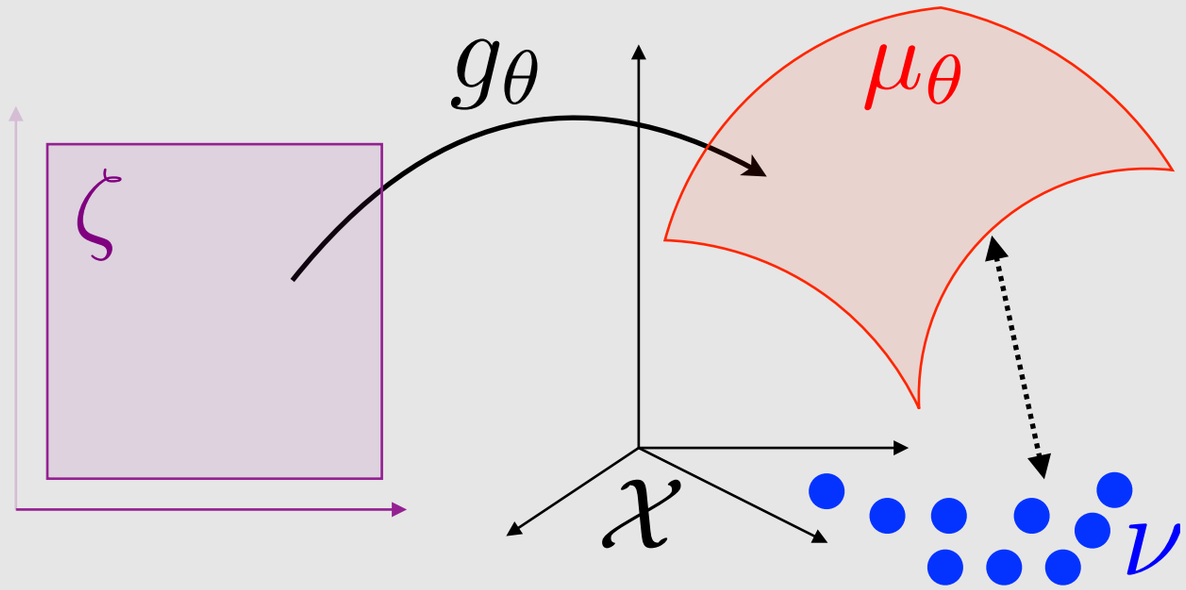


2. Entropic Regularization



3. Sinkhorn Divergences

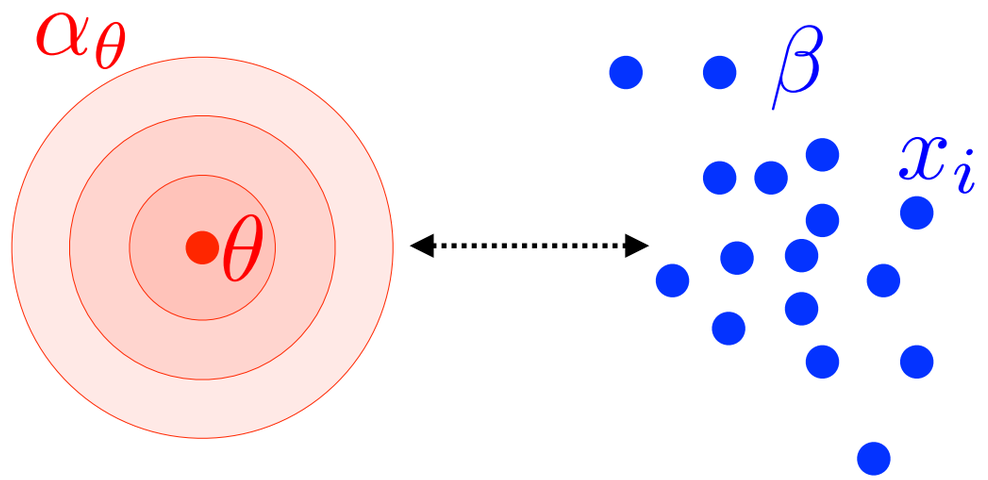
4. Application to Generative Models



Density Fitting and Generative Models

$$\textit{Observations: } \beta \overset{\text{def.}}{=} \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}$$

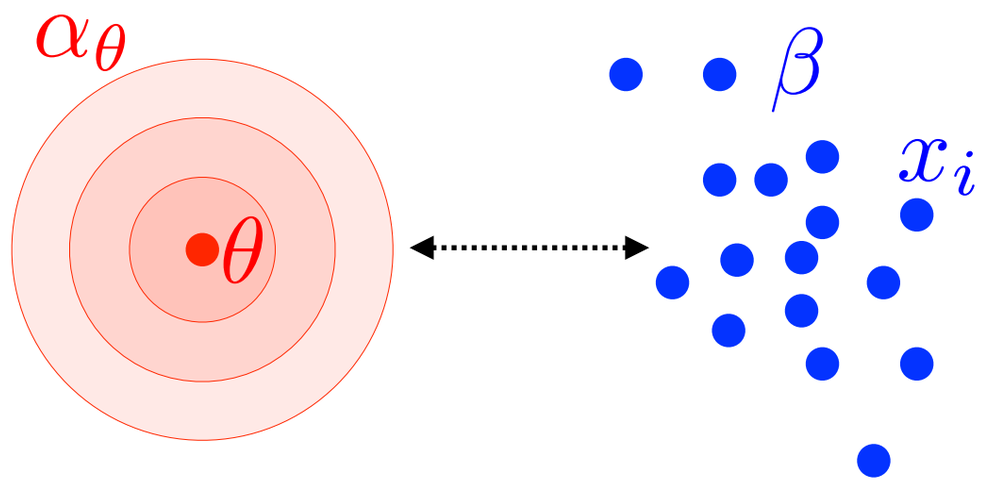
$$\textit{Parametric model: } \theta \mapsto \alpha_\theta$$



Density Fitting and Generative Models

Observations: $\beta \overset{\text{def.}}{=} \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}$

Parametric model: $\theta \mapsto \alpha_\theta$



Density fitting: $\mathrm{d}\alpha_\theta(x) = \rho_\theta(x)\mathrm{d}x$

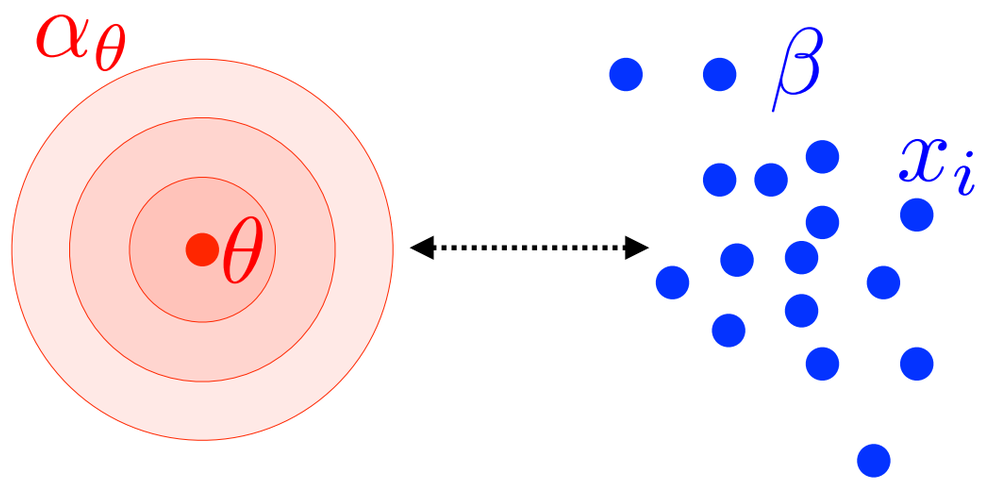
$$\min_\theta \; -\sum_i \log(\rho_\theta(x_i)) \overset{n\to+\infty}{\longrightarrow} \mathrm{KL}(\beta|\alpha_\theta)$$

Maximum likelihood (MLE)

Density Fitting and Generative Models

Observations: $\beta \overset{\text{def.}}{=} \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}$

Parametric model: $\theta \mapsto \alpha_\theta$



Density fitting: $\mathrm{d}\alpha_\theta(x) = \rho_\theta(x)\mathrm{d}x$

$$\min_\theta \; -\sum_i \log(\rho_\theta(x_i)) \overset{n\to+\infty}{\longrightarrow} \mathrm{KL}(\beta|\alpha_\theta)$$

Maximum likelihood (MLE)

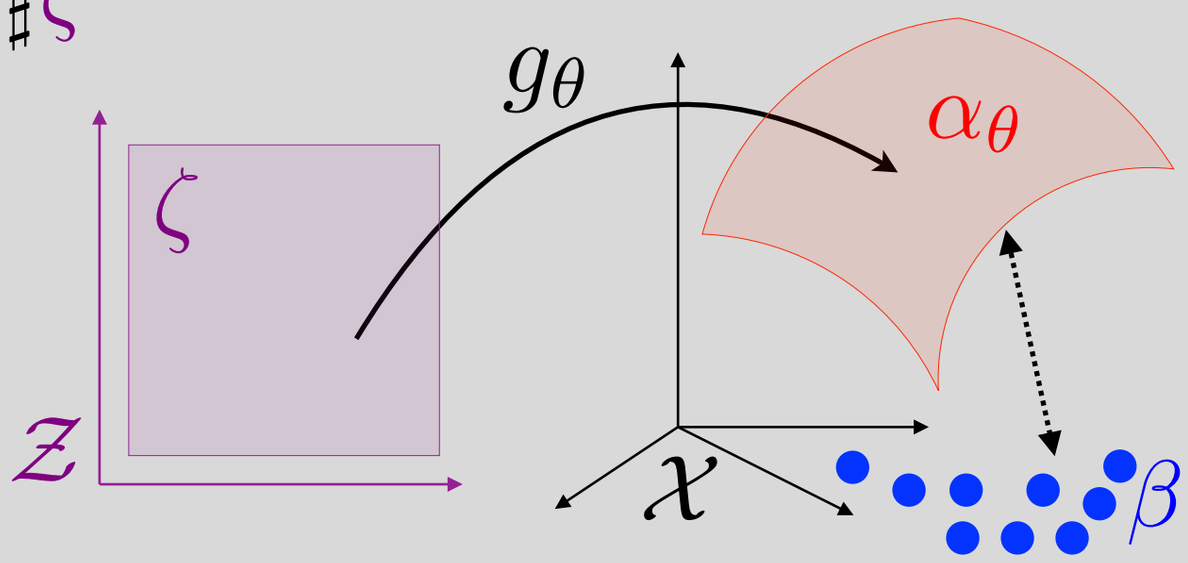
Generative model fit: $\alpha_\theta = g_{\theta,\sharp}\zeta$

$$\mathrm{KL}(\beta|\alpha_\theta) = +\infty$$

$\to$ MLE undefined.

$\to$ Need a weaker metric.

$$\min_\theta \overline{\mathrm{W}}^p_{\varepsilon,p}(\alpha_\theta, \beta)$$

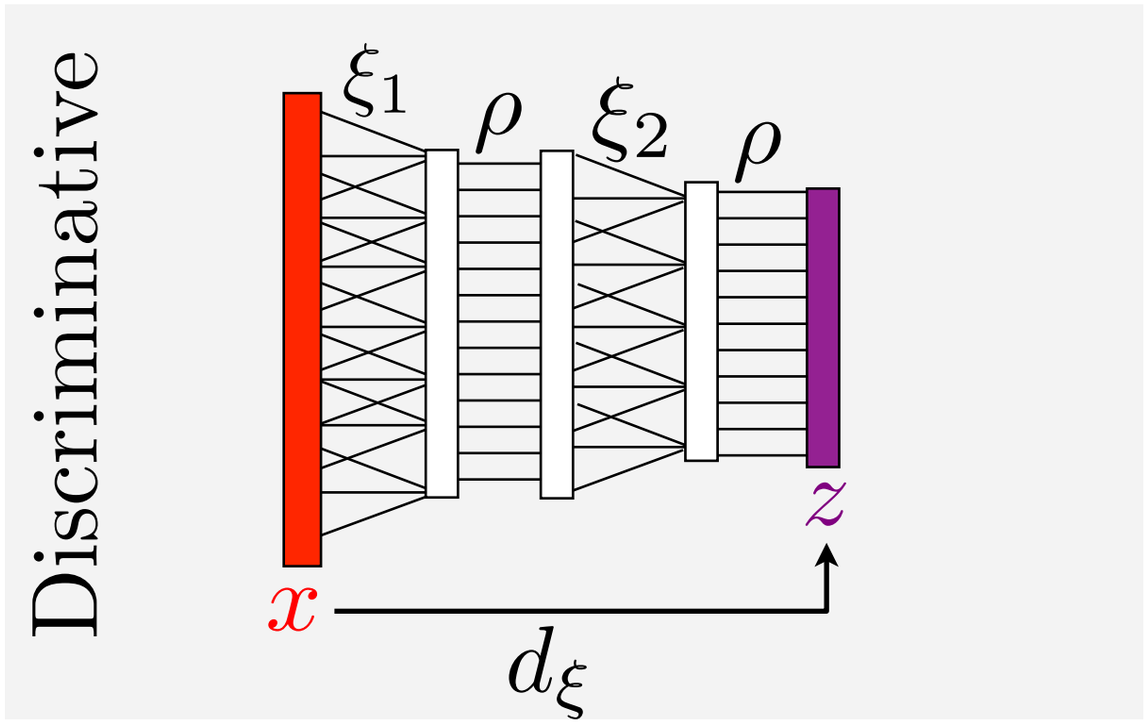


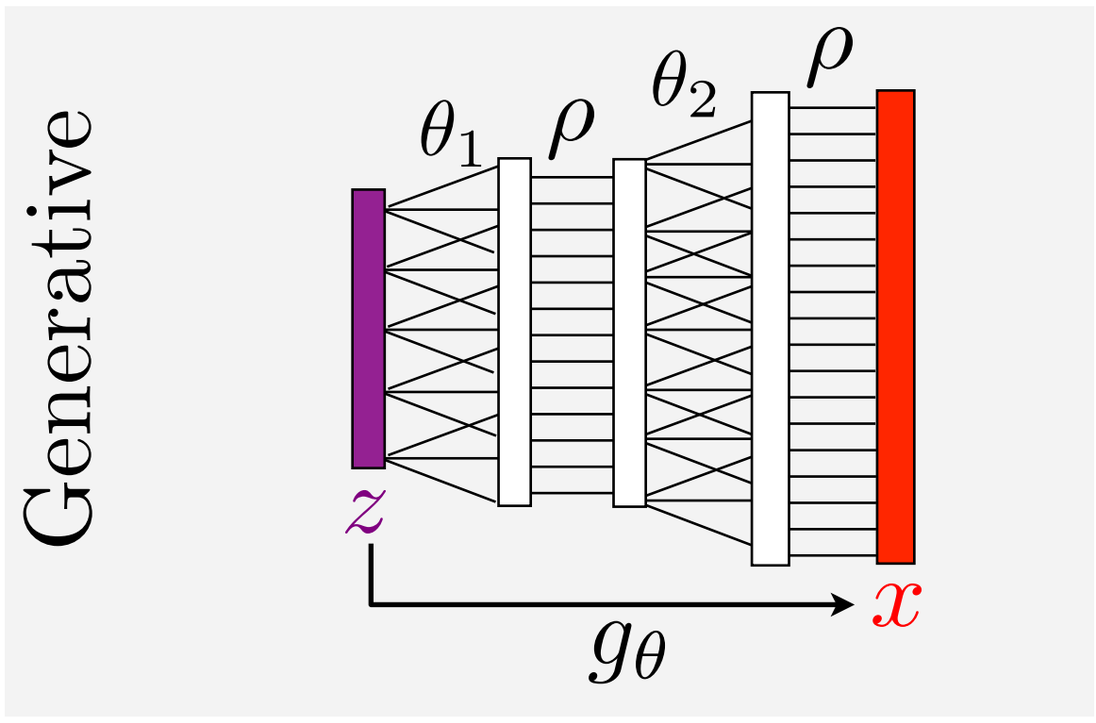
Deep Discriminative vs Generative Models

Deep networks:

$$d_\xi(x) = \rho(\xi_K(\dots\rho(\xi_2(\rho(\xi_1(x)\dots)$$

$$g_\theta(z) = \rho(\theta_K(\dots\rho(\theta_2(\rho(\theta_1(z)\dots)$$



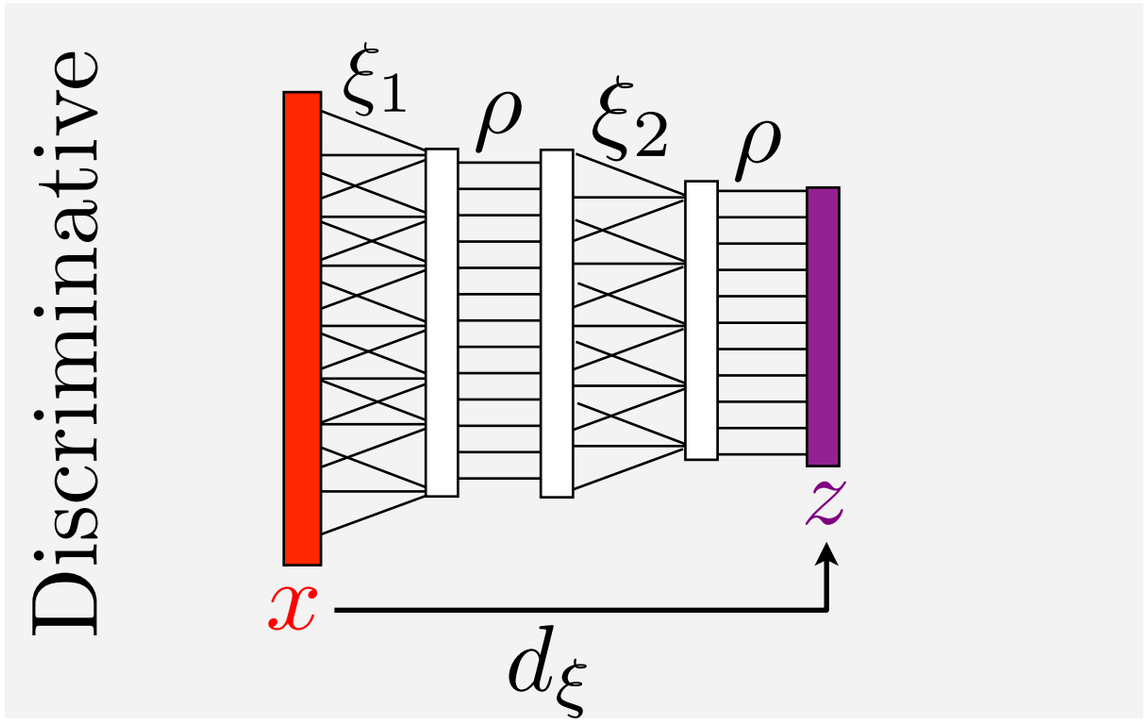


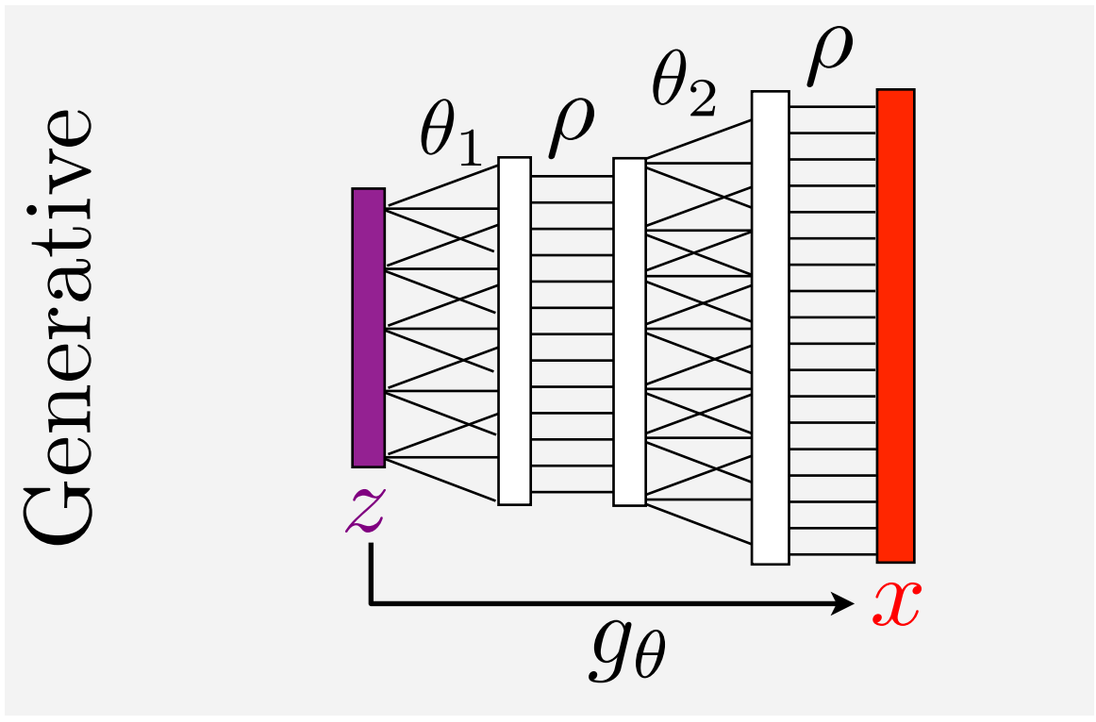
Deep Discriminative vs Generative Models

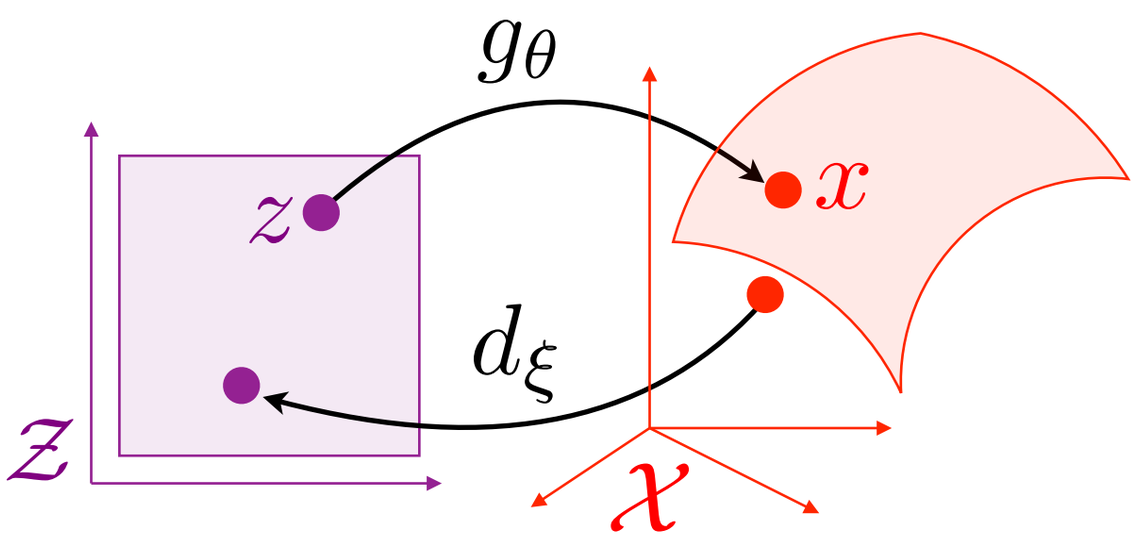
Deep networks:

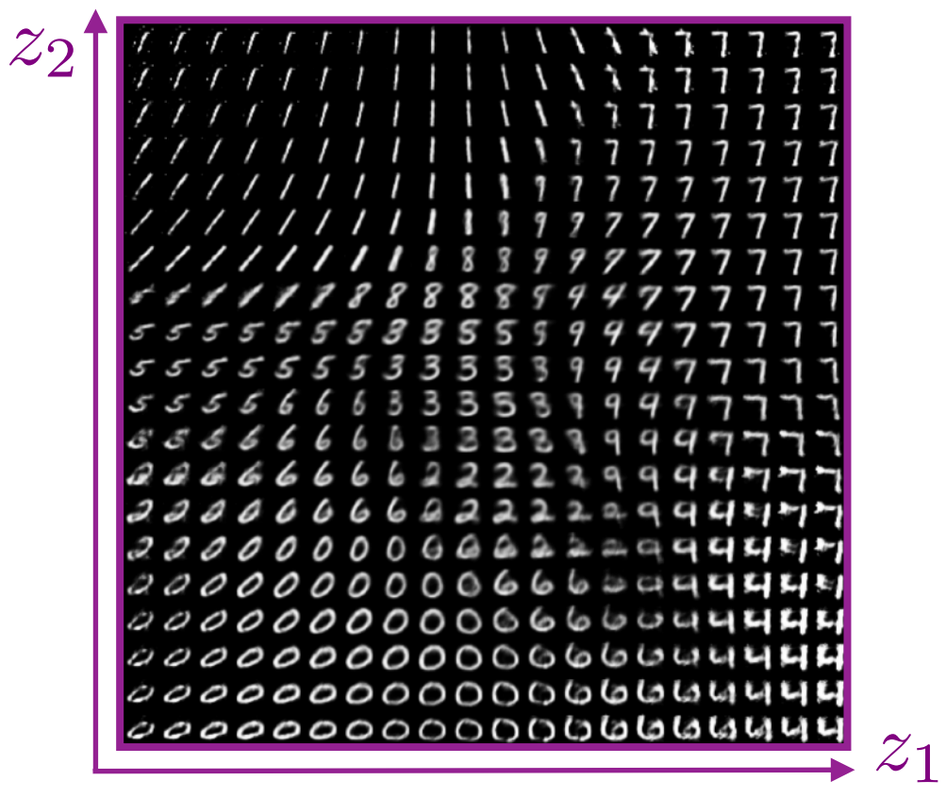
$$d_\xi(x) = \rho(\xi_K(\dots\rho(\xi_2(\rho(\xi_1(x)\dots)$$

$$g_\theta(z) = \rho(\theta_K(\dots\rho(\theta_2(\rho(\theta_1(z)\dots)$$





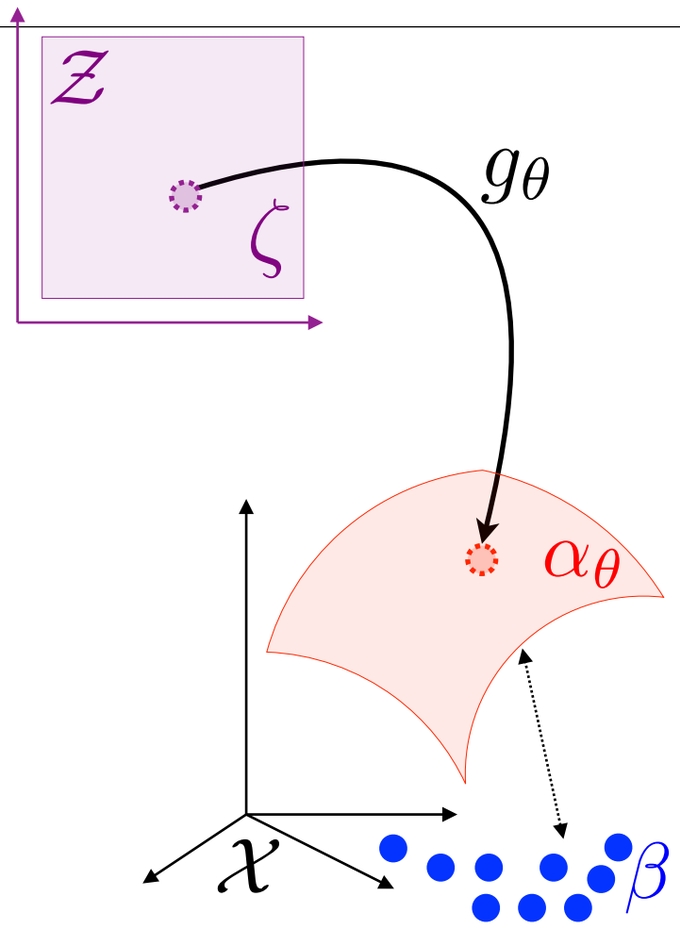




Examples of Images Generation

Inputs β

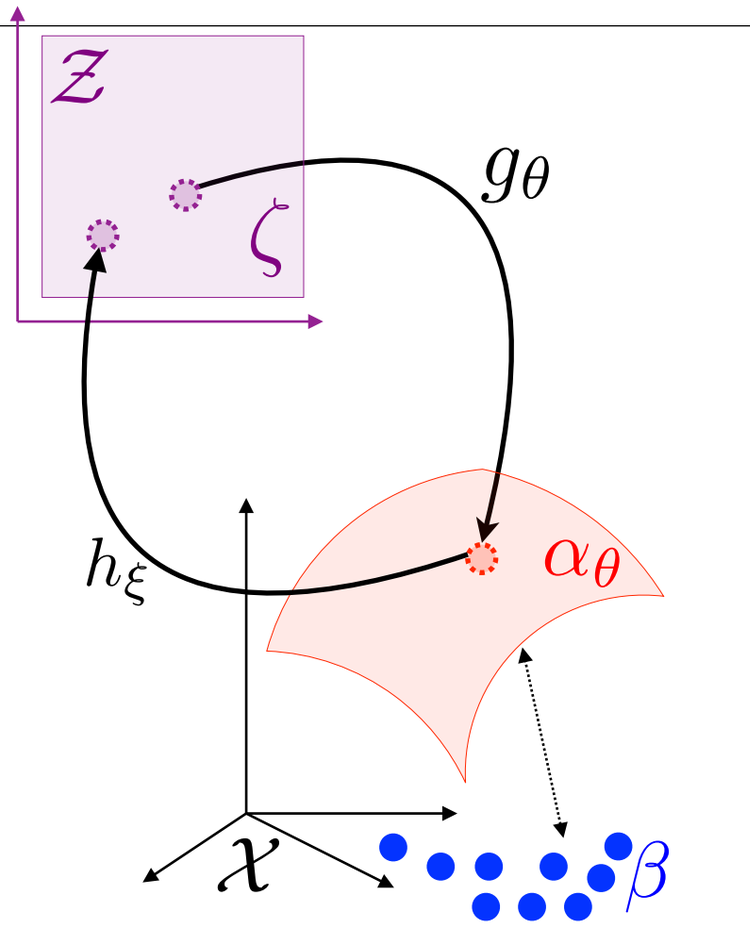
Generated α_θ

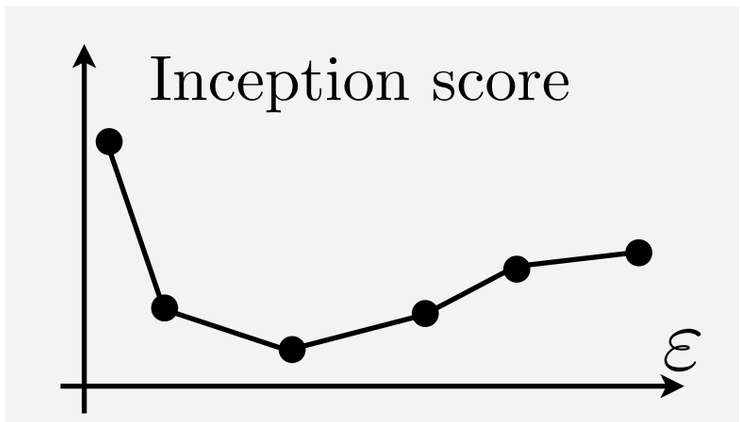


Examples of Images Generation

Inputs β

Generated α_θ





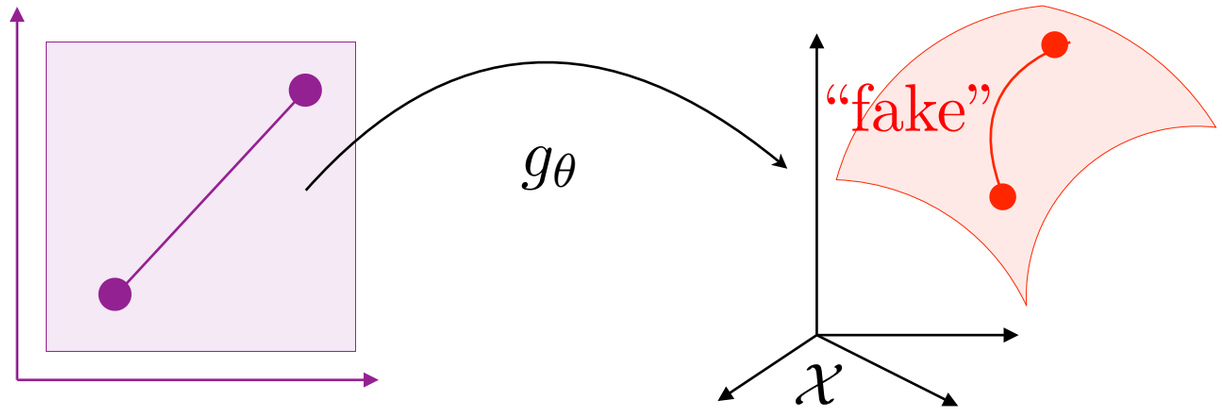
$\rightarrow$ Need to learn the metric $d(x, y) = \|h_\xi(x) - h_\xi(y)\|$ (GANs)

$\rightarrow$ Influence of ε?

$\rightarrow$ Performance evaluation of generative models is an open problem.

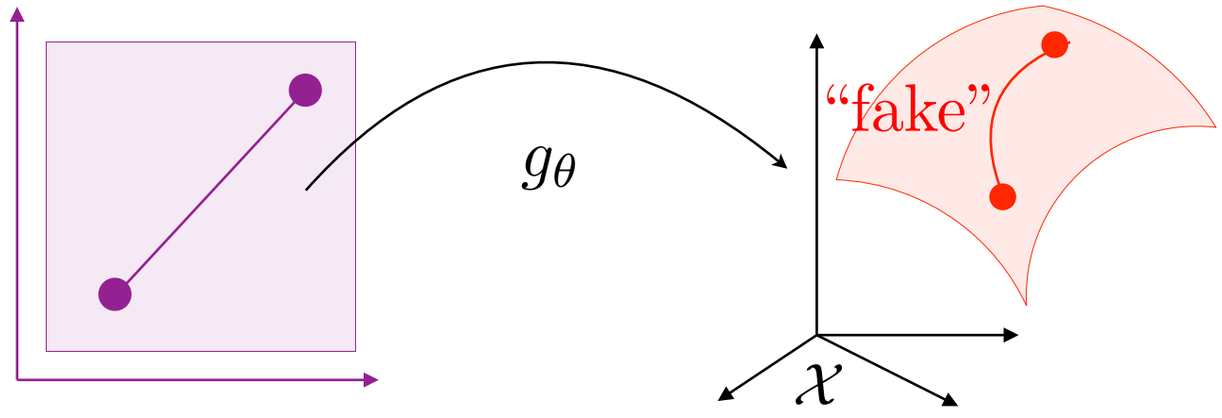
Progressive Growing of GANs for Improved Quality, Stability, and Variation

Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen, ICLR 2018



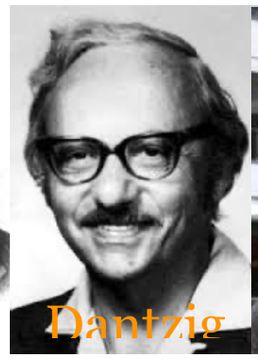
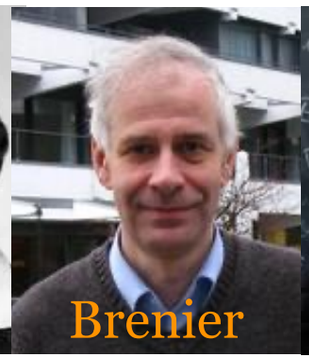
Progressive Growing of GANs for Improved Quality, Stability, and Variation

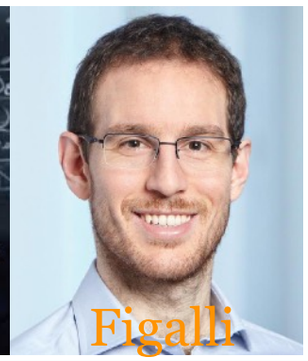
Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen, ICLR 2018



Conclusion



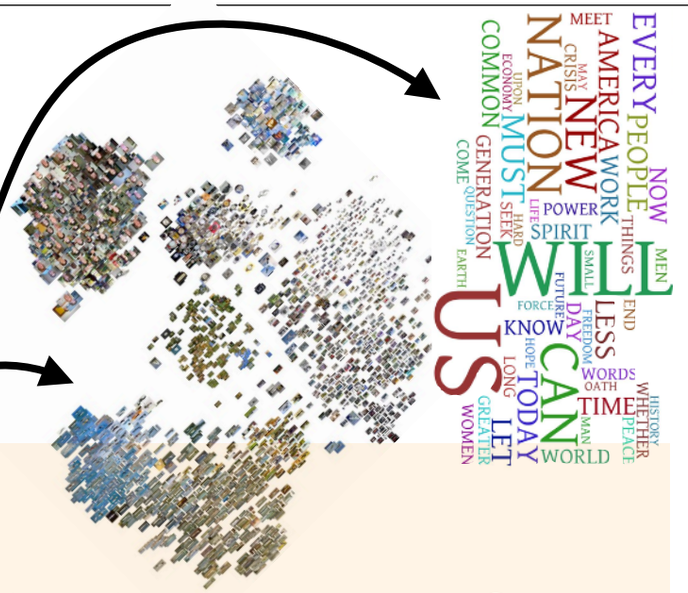


Toward high-dimensional OT:

$\rightarrow$ Scalable geometrical loss functions in high dimension?

$\rightarrow$ Performance quality measures for unsupervised learning?

Conclusion



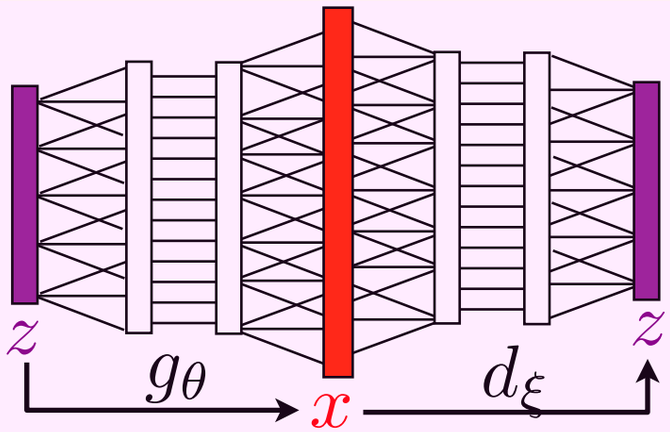
Toward high-dimensional OT:

→ Scalable geometrical loss functions in high dimension?

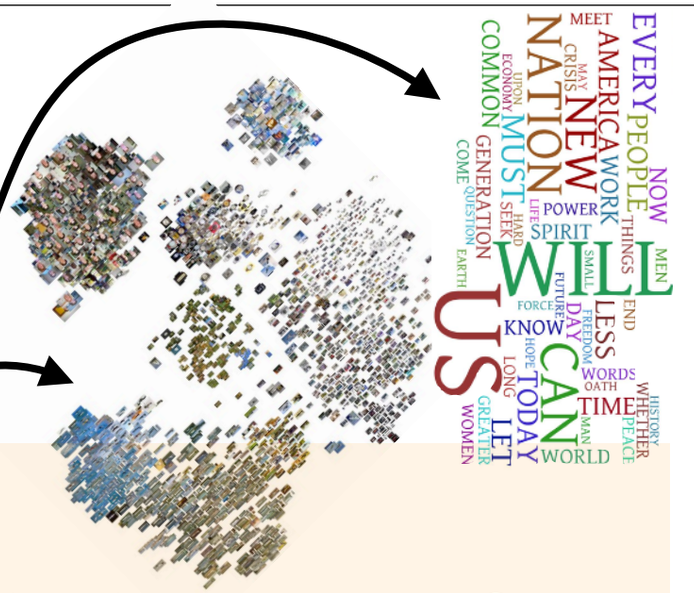
→ Performance quality measures for unsupervised learning?

Metric learning for OT:

→ Adversarial training to leverage multi scale priors?



Conclusion



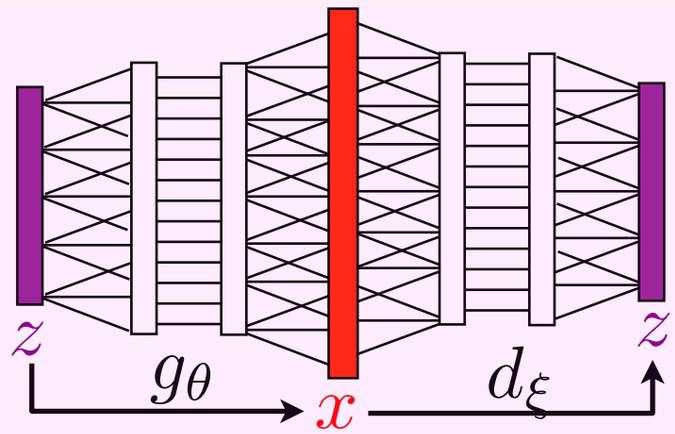
Toward high-dimensional OT:

→ Scalable geometrical loss functions in high dimension?

→ Performance quality measures for unsupervised learning?

Metric learning for OT:

→ Adversarial training to leverage multi scale priors?



Beyond comparing measures:

→ Learning for surfaces, graphs, metric spaces?

→ Using Gromov-Wasserstein geometry?