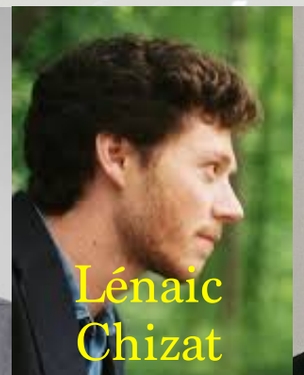
Scaling Optimal Transport to High Dimensional Learning

Gabriel Peyré

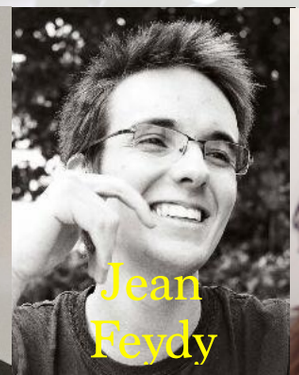
Joint works with:


Shun'ichi Amari

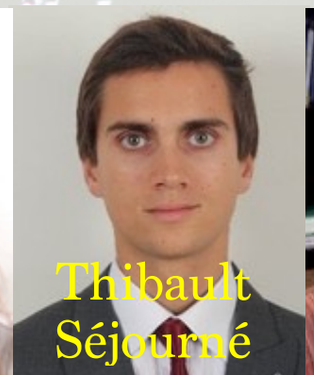

Francis Bach


Lénaic Chizat


Marco Cuturi


Jean Feydy

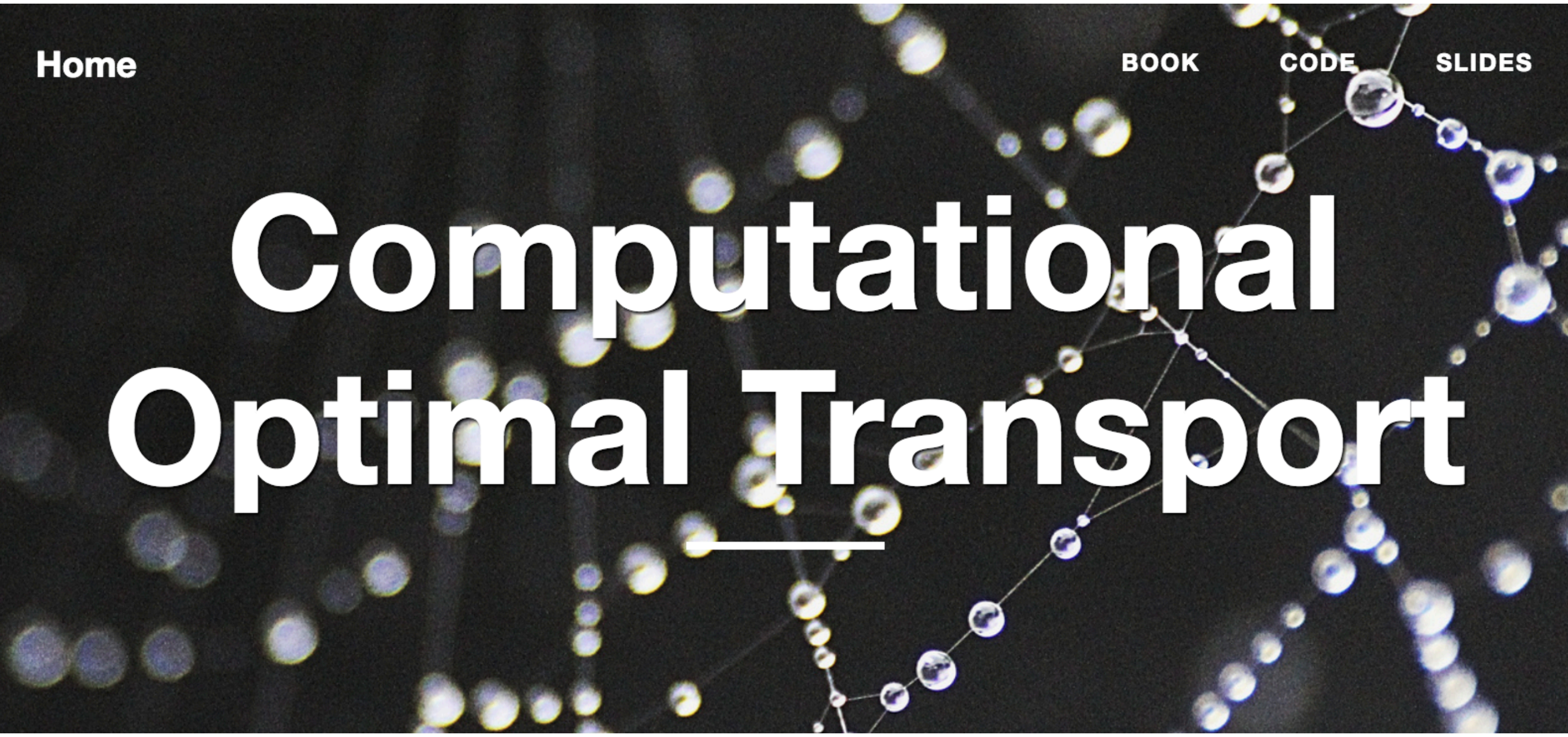

Aude Genevay


Thibault Séjourné


Alain Trouvé


François-Xavier Vialard

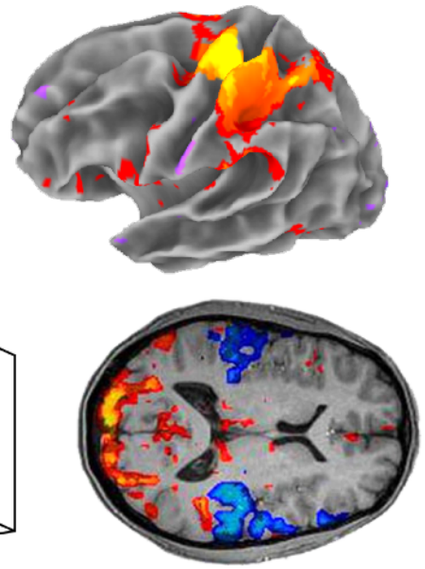
https://optimaltransport.github.io

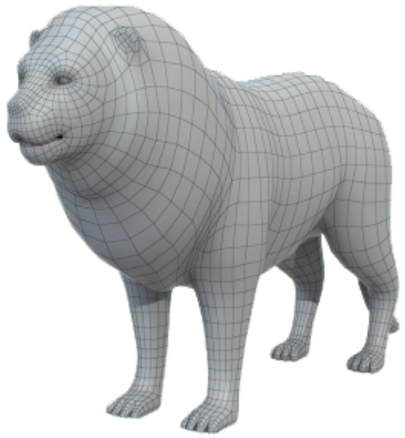


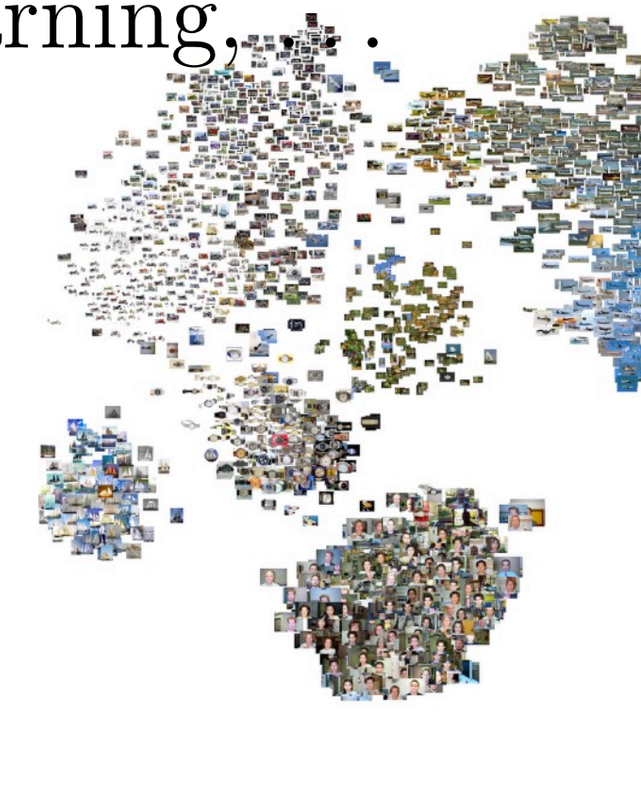
Probability Distributions in Data Sciences

Probability distributions and histograms

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



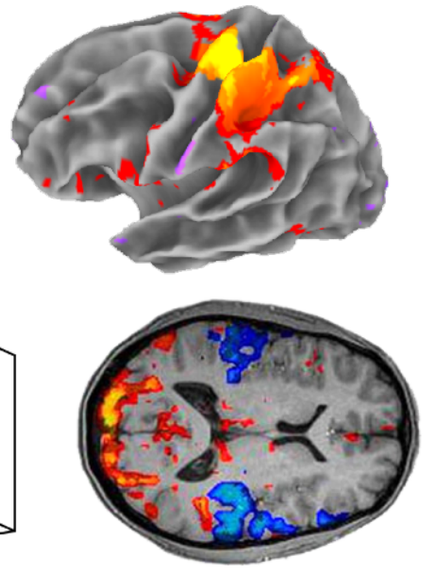


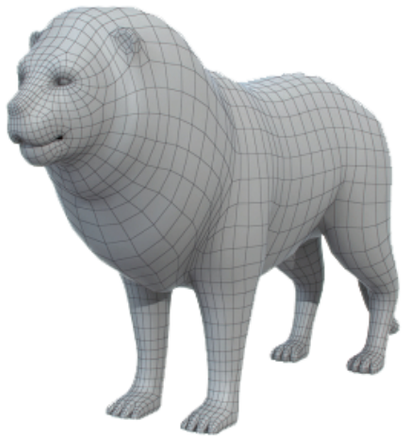


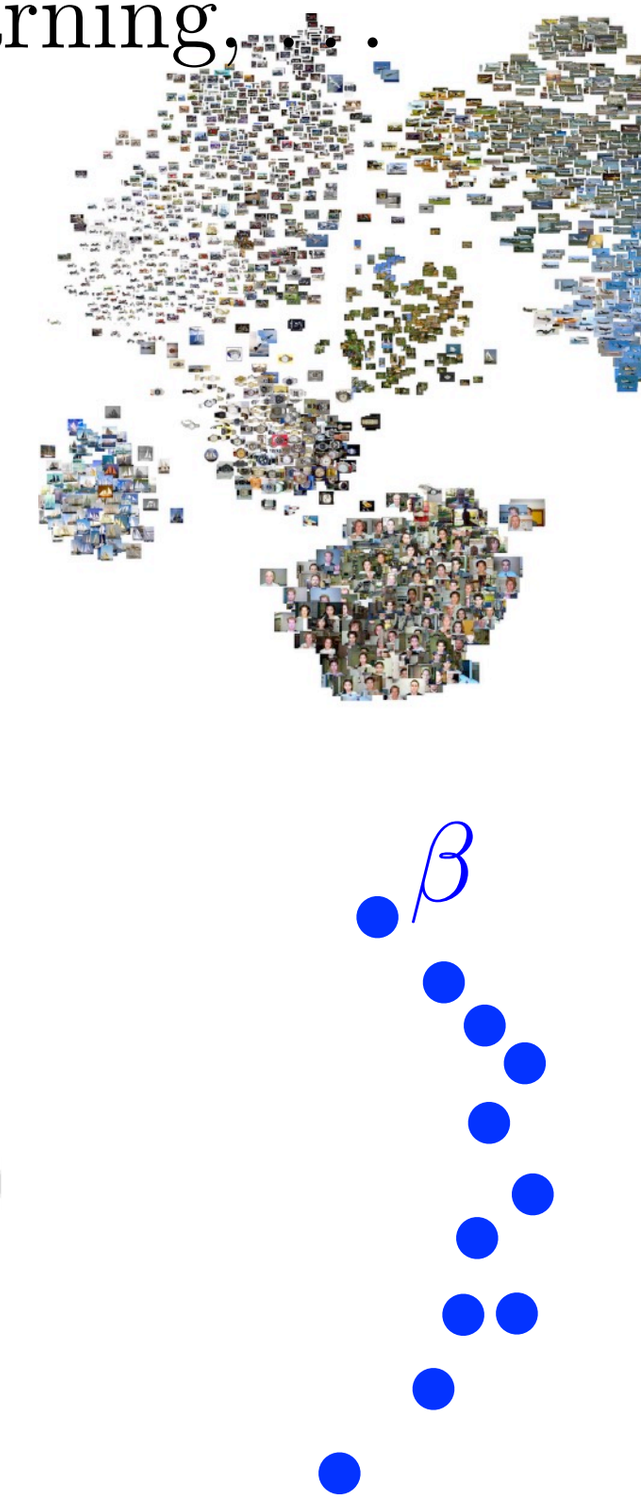
Probability Distributions in Data Sciences

Probability distributions and histograms

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



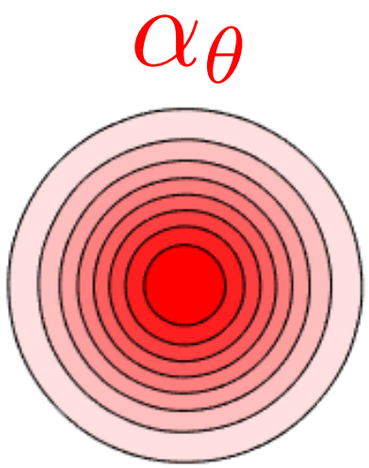




Unsupervised learning

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

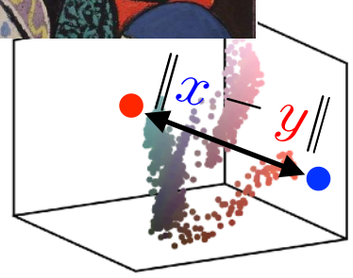
Parametric model: $\theta \mapsto \alpha_\theta$

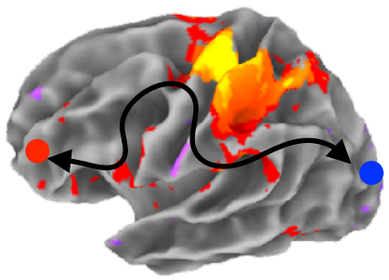


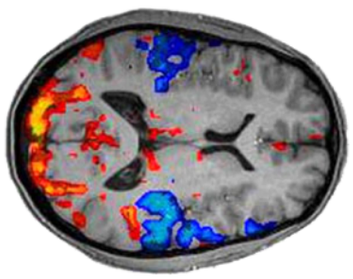
Probability Distributions in Data Sciences

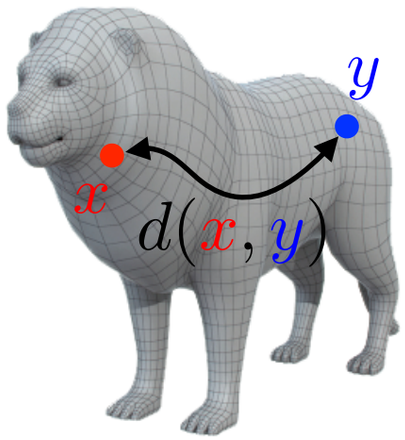
Probability distributions and histograms

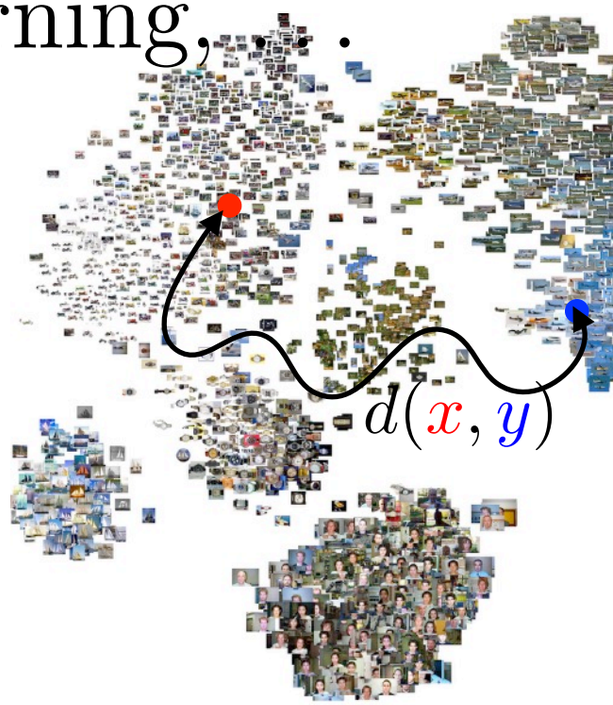
$\rightarrow$ images, vision, graphics and machine learning, . . .











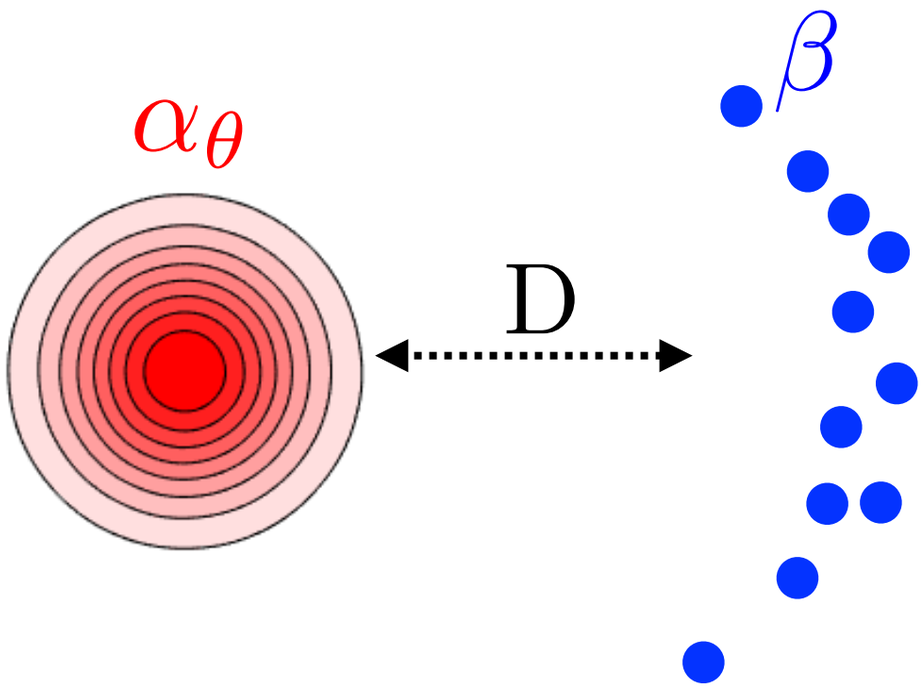
Unsupervised learning

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

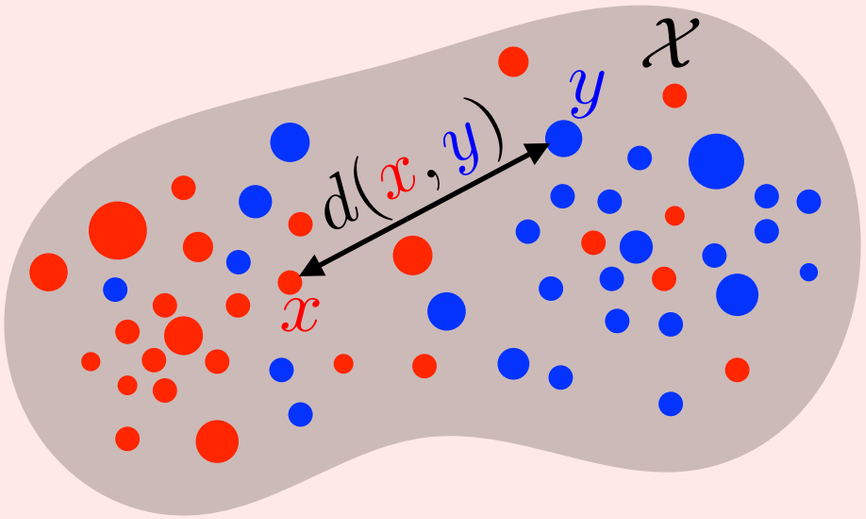
Parametric model: $\theta \mapsto \alpha_\theta$

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

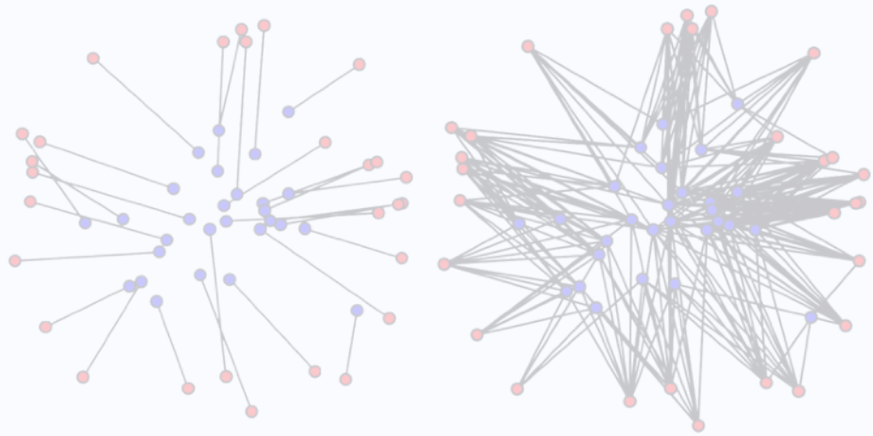
$\rightarrow$ takes into account a metric d.



1. Optimal Transport

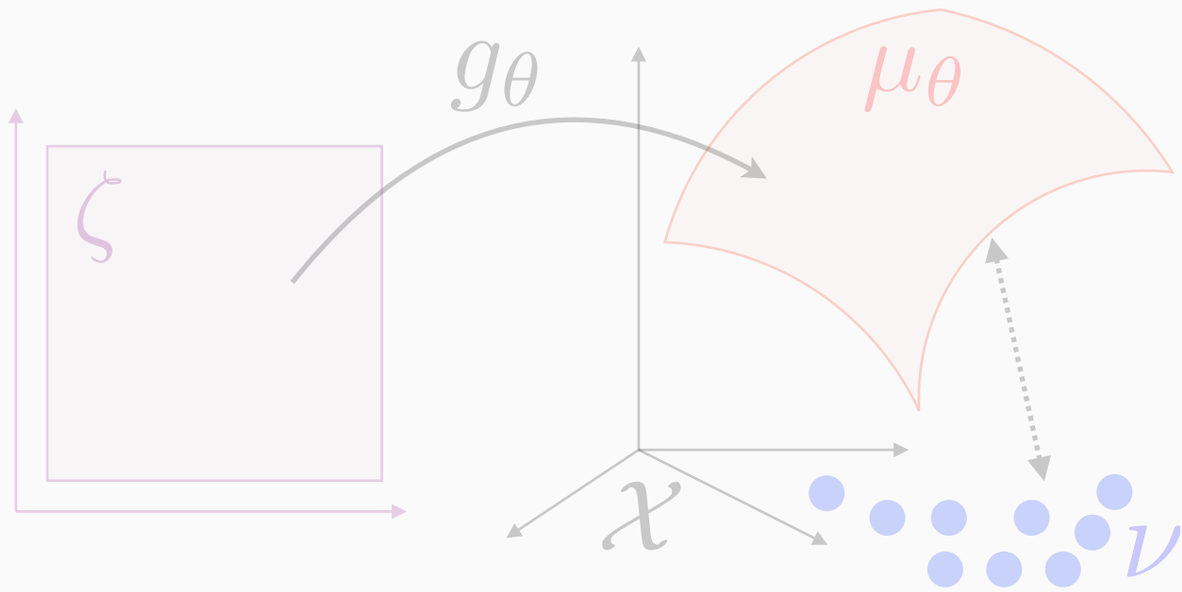


2. Entropic Regularization



3. Sinkhorn Divergences

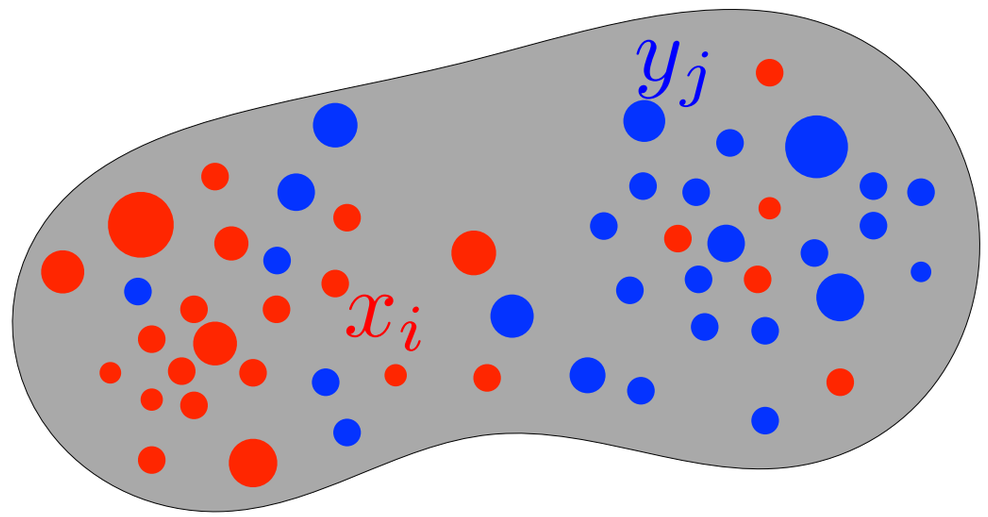
4. Application to Generative Models



Kantorovitch's Formulation

Input distributions

$\alpha = \sum_{i=1}^{n} \mathbf{a}_i \delta_{x_i} \quad \beta = \sum_{j=1}^{m} \mathbf{b}_j \delta_{y_j}$



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

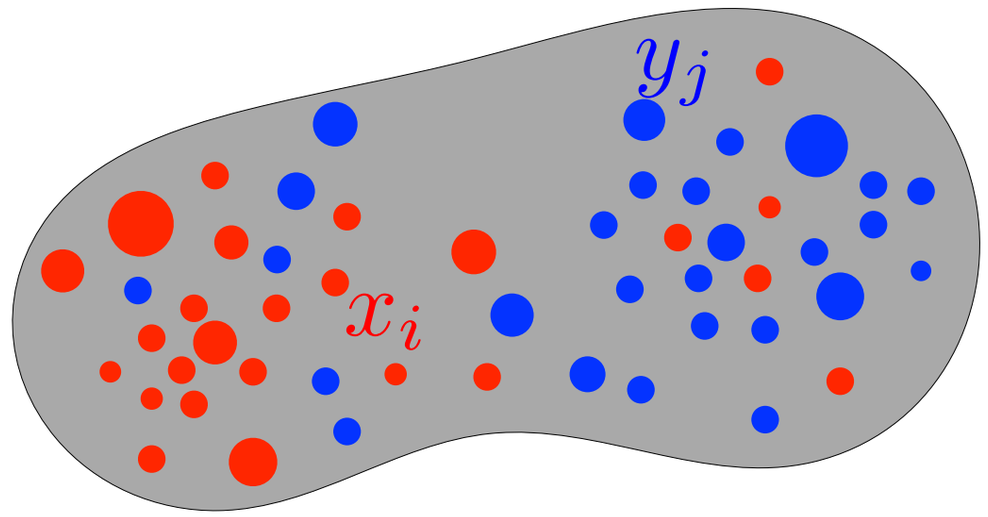
Weights $\mathbf{a}_i \geqslant 0$, $\mathbf{b}_j \geqslant 0$.

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

Kantorovitch's Formulation

Input distributions

$$\alpha = \sum_{i=1}^n \mathbf{a}_i \delta_{x_i} \quad \beta = \sum_{j=1}^m \mathbf{b}_j \delta_{y_j}$$



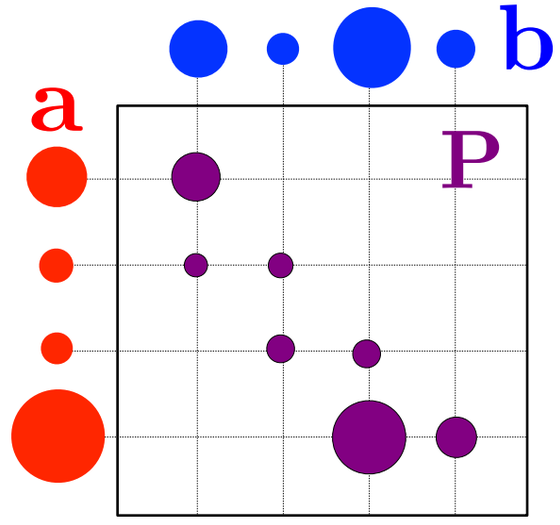
Points $(x_i)_i$, $(y_j)_j$

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:

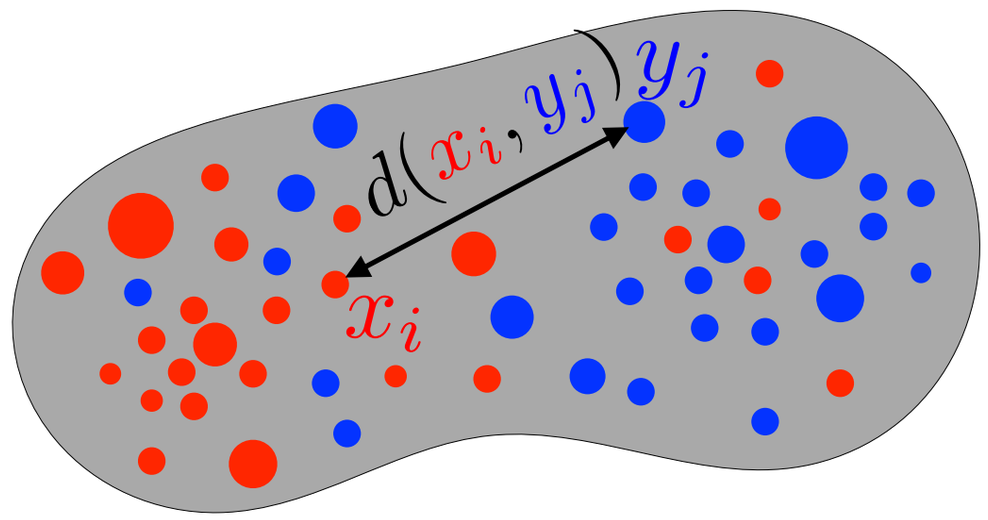
$$\mathbf{U}(\mathbf{a}, \mathbf{b}) \stackrel{\text{def.}}{=} \left\{ \mathbf{P} \in \mathbb{R}_+^{n \times m} \;;\; \mathbf{P}\mathbb{1}_m = \mathbf{a}, \mathbf{P}^\top \mathbb{1}_n = \mathbf{b} \right\}$$



Kantorovitch's Formulation

Input distributions

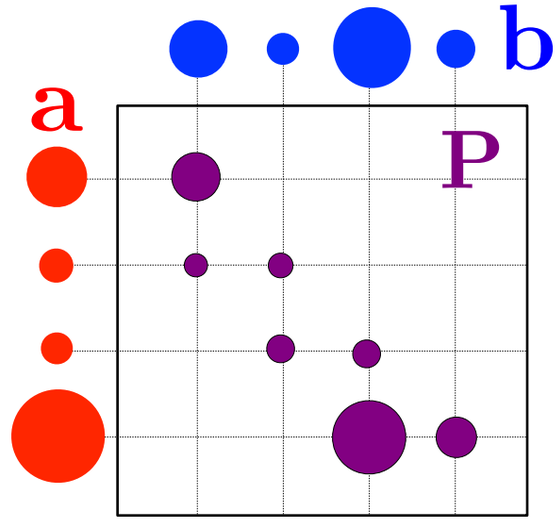
$$\alpha = \sum_{i=1}^n \mathbf{a}_i \delta_{x_i} \quad \beta = \sum_{j=1}^m \mathbf{b}_j \delta_{y_j}$$



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

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

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

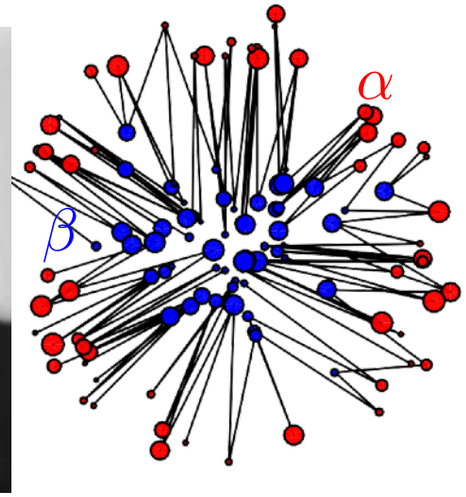


Couplings:

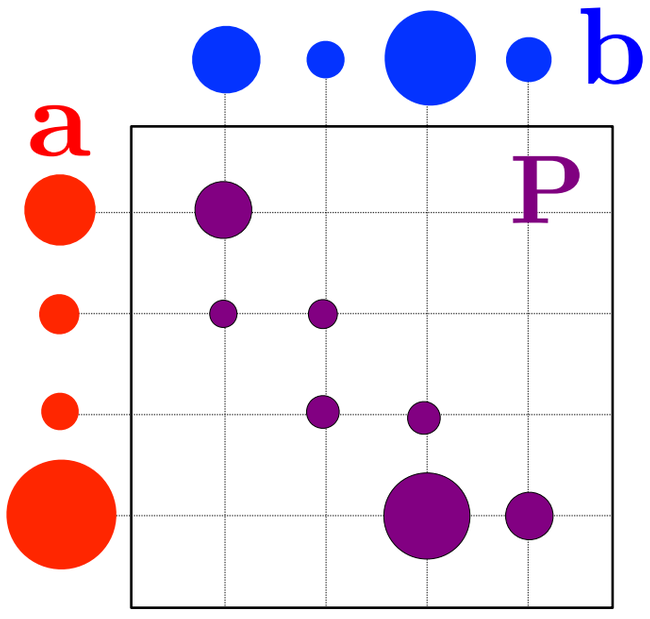
$$\mathbf{U}(\mathbf{a}, \mathbf{b}) \stackrel{\text{def.}}{=} \left\{ \mathbf{P} \in \mathbb{R}_+^{n \times m} \;;\; \mathbf{P}\mathbb{1}_m = \mathbf{a}, \mathbf{P}^\top \mathbb{1}_n = \mathbf{b} \right\}$$

[Kantorovich 1942]

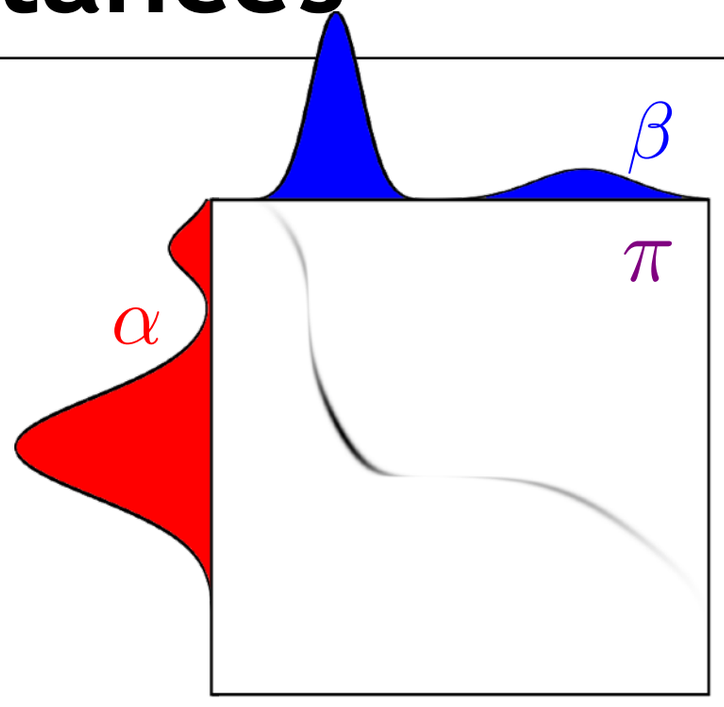
$$\min \left\{ \sum_{i,j} d(x_i, y_j)^p \mathbf{P}_{i,j} \;;\; \mathbf{P} \in \mathbf{U}(\mathbf{a}, \mathbf{b}) \right\}$$



Optimal Transport Distances

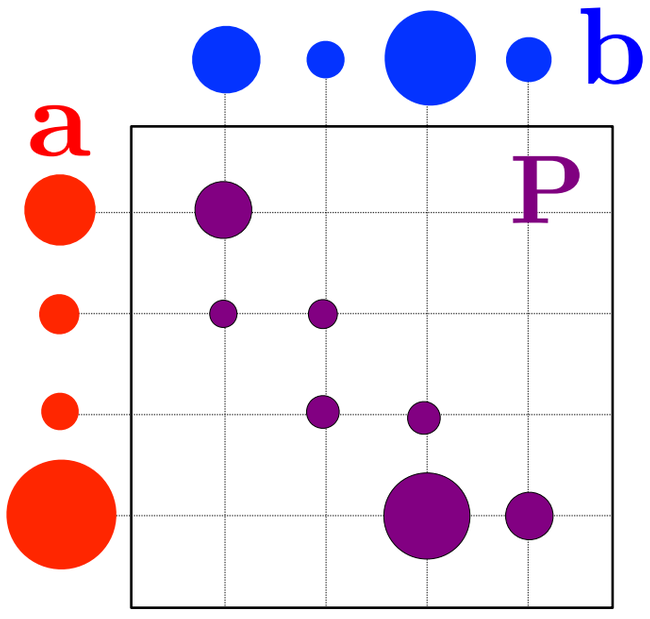


$$\pi = \sum_{i,j} \mathbf{P}_{i,j} \delta_{x_i, y_j}$$

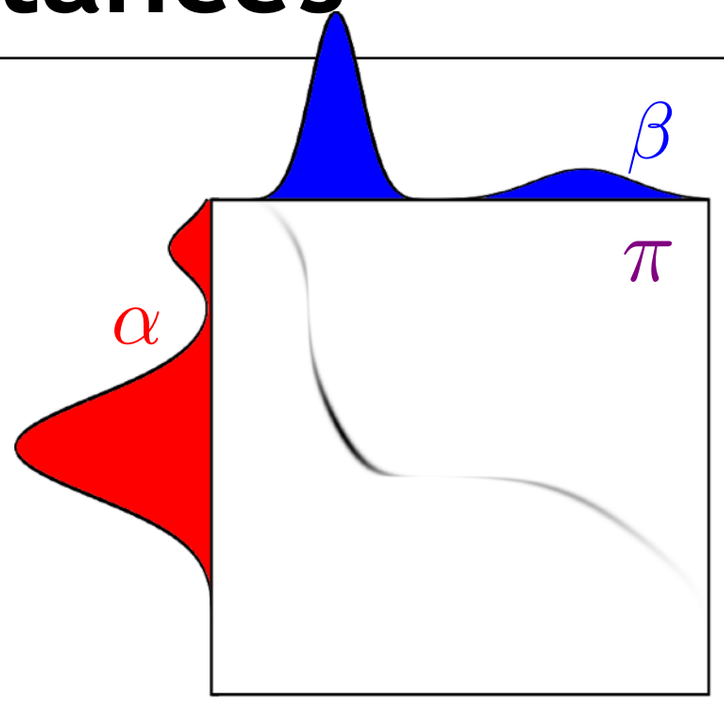


$$\mathrm{W}_p(\alpha,\beta)^p \stackrel{\text{def.}}{=} \min_{\pi \in \mathcal{M}_+^1(\mathcal{X}^2)} \left\{ \int_{\mathcal{X}^2} d(x,y)^p \mathrm{d}\pi(x,y) \;;\; \pi_1 = \alpha, \pi_2 = \beta \right\}$$

Optimal Transport Distances



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



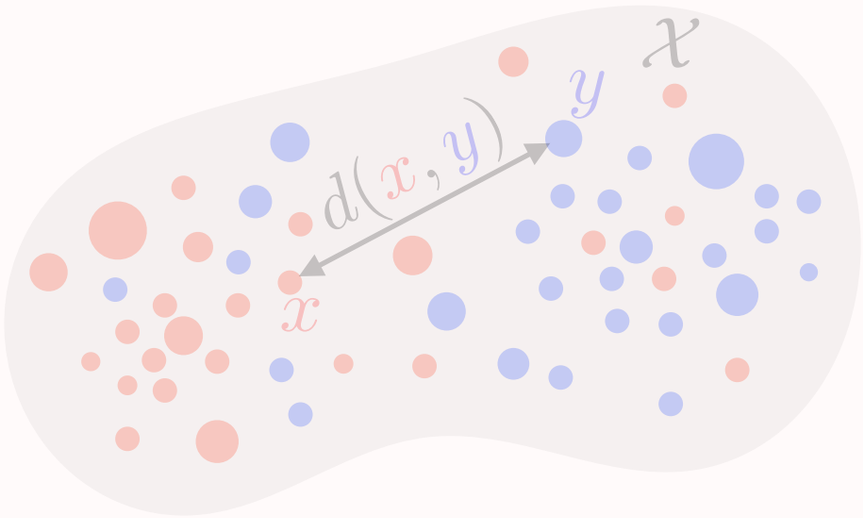
$$\mathrm{W}_p(\alpha,\beta)^p \stackrel{\text{def.}}{=} \min_{\pi \in \mathcal{M}_+^1(\mathcal{X}^2)} \left\{ \int_{\mathcal{X}^2} d(x,y)^p \mathrm{d}\pi(x,y) \;;\; \pi_1 = \alpha, \pi_2 = \beta \right\}$$

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

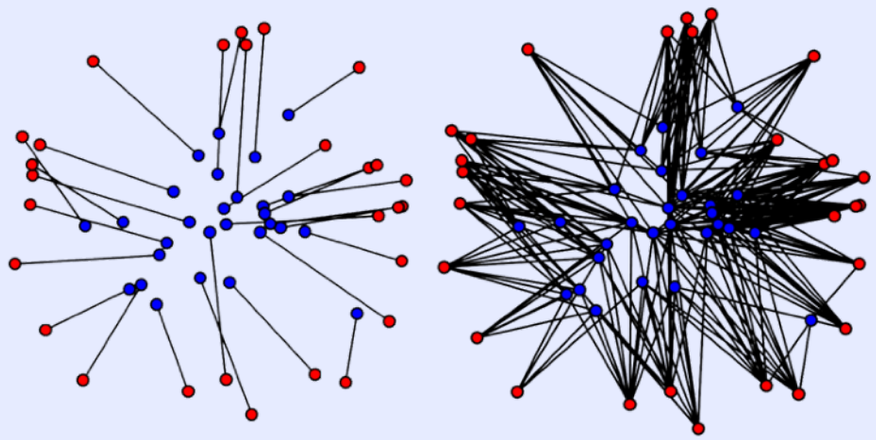
Weak* (aka in law) convergence: $\alpha_n \rightharpoonup \beta \;\Leftrightarrow\; \begin{array}{c} \forall\, f \in \mathcal{C}(\mathcal{X}), \\ \int_{\mathcal{X}} f \mathrm{d}\alpha_n \to \int_{\mathcal{X}} f \mathrm{d}\beta \end{array}$

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

1. Optimal Transport

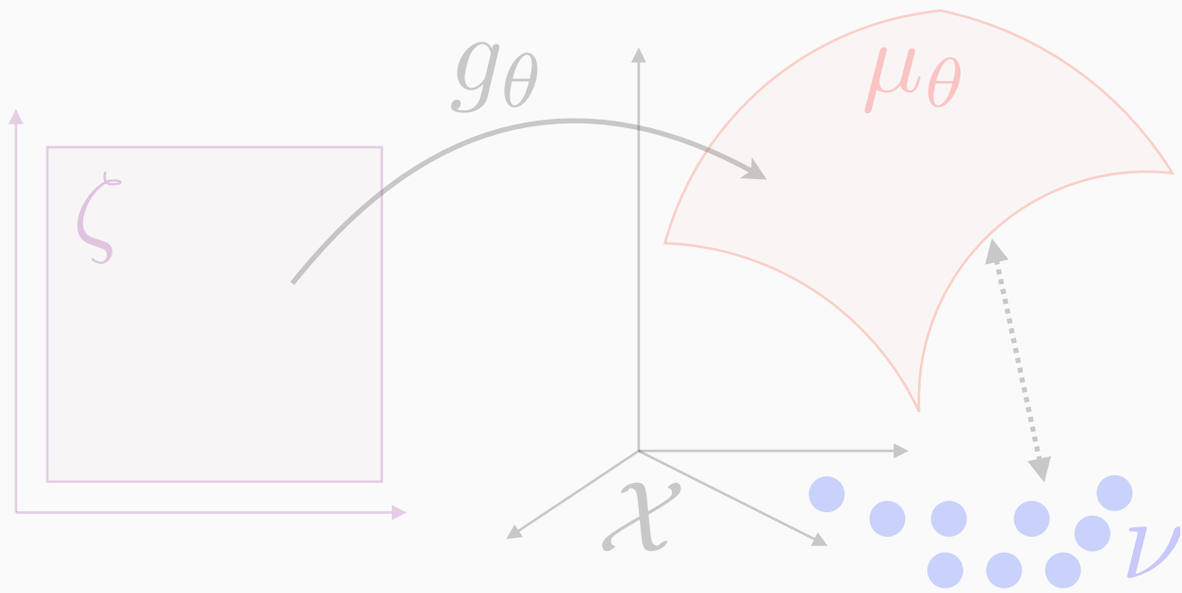


2. Entropic Regularization



3. Sinkhorn Divergences

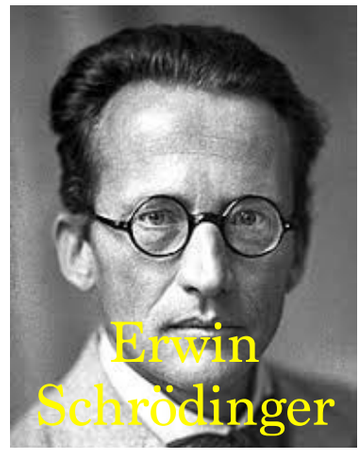
4. Application to Generative Models



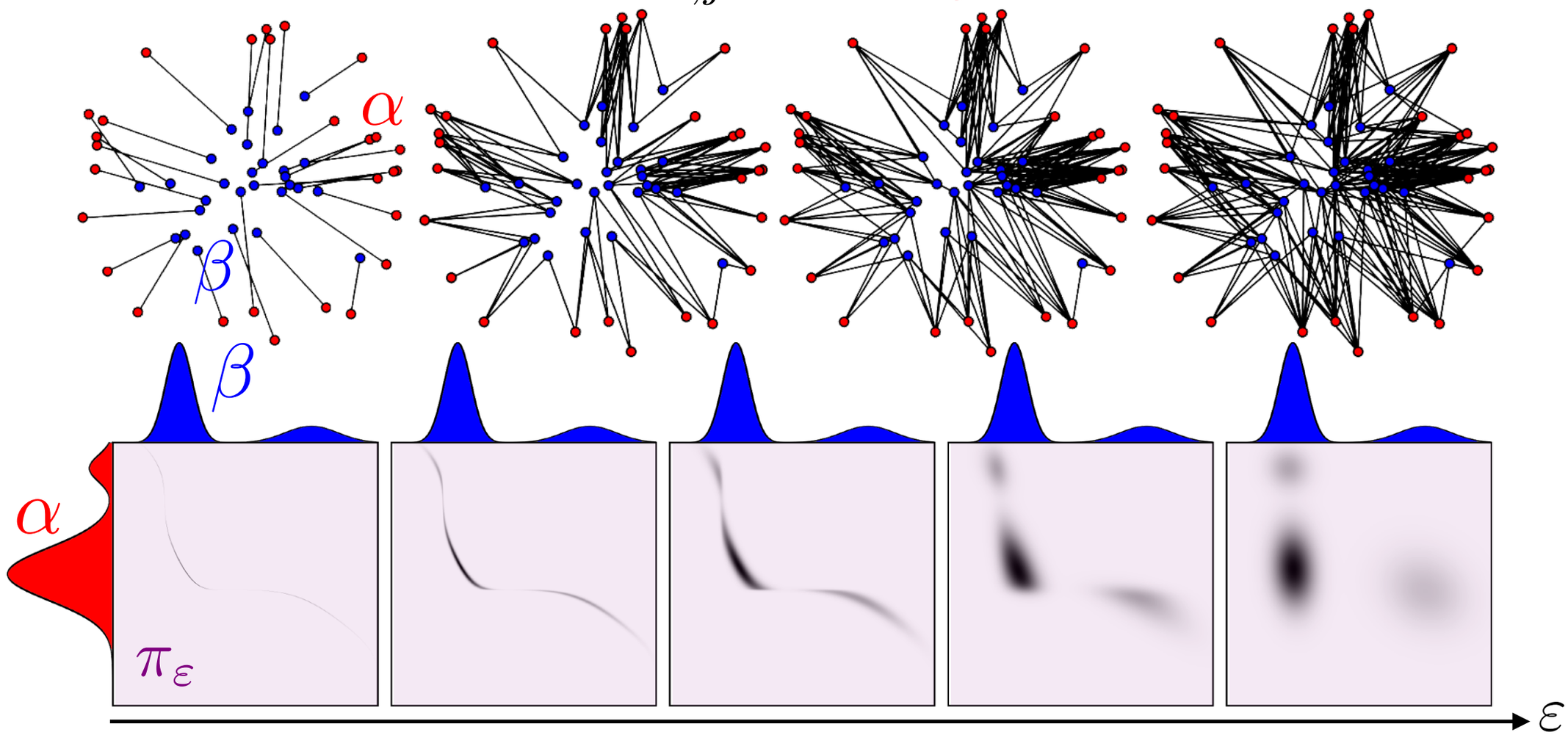
Entropic Regularization

Schrödinger's problem: [1931]

$$\min_{\mathbf{P}\in\mathbf{U}(\mathbf{a},\mathbf{b})} \sum_{i,j} d(x_i, y_j)^p \mathbf{P}_{i,j} + \varepsilon \mathbf{P}_{i,j} \log\left(\frac{\mathbf{P}_{i,j}}{\mathbf{a}_i \mathbf{b}_j}\right)$$



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



Sinkhorn's Algorithm

$$\min_{\mathbf{P} \in \mathbf{U}(\mathbf{a},\mathbf{b})} \sum_{i,j} d(x_i, y_j)^p \mathbf{P}_{i,j} + \varepsilon \mathbf{P}_{i,j} \log\left(\frac{\mathbf{P}_{i,j}}{\mathbf{a}_i \mathbf{b}_j}\right)$$

Proposition: $\quad \mathbf{P}_{i,j} = \mathbf{u}_i \, \mathbf{K}_{i,j} \, \mathbf{v}_j \qquad \mathbf{K}_{i,j} \overset{\text{def.}}{=} e^{-\frac{d(x_i,y_j)^p}{\varepsilon}}$

Sinkhorn's Algorithm

$$\min_{\mathbf{P} \in \mathbf{U}(\mathbf{a},\mathbf{b})} \sum_{i,j} d(x_i, y_j)^p \mathbf{P}_{i,j} + \varepsilon \mathbf{P}_{i,j} \log\left(\frac{\mathbf{P}_{i,j}}{\mathbf{a}_i \mathbf{b}_j}\right)$$

Proposition: $\quad \mathbf{P}_{i,j} = \mathbf{u}_i \, \mathbf{K}_{i,j} \, \mathbf{v}_j \qquad \mathbf{K}_{i,j} \overset{\text{def.}}{=} e^{-\frac{d(x_i,y_j)^p}{\varepsilon}}$

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

$$\min_{\mathbf{P}\in\mathbf{U}(\mathbf{a},\mathbf{b})} \sum_{i,j} d(x_i,y_j)^p \mathbf{P}_{i,j} + \varepsilon \mathbf{P}_{i,j} \log\left(\frac{\mathbf{P}_{i,j}}{\mathbf{a}_i \mathbf{b}_j}\right)$$

Proposition: $\mathbf{P}_{i,j} = \mathbf{u}_i \, \mathbf{K}_{i,j} \, \mathbf{v}_j$ $\qquad \mathbf{K}_{i,j} \overset{\text{def.}}{=} e^{-\frac{d(x_i,y_j)^p}{\varepsilon}}$

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

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

Sinkhorn's Algorithm

$$\min_{\mathbf{P}\in\mathbf{U}(\mathbf{a},\mathbf{b})} \sum_{i,j} d(x_i,y_j)^p \mathbf{P}_{i,j} + \varepsilon \mathbf{P}_{i,j} \log\left(\frac{\mathbf{P}_{i,j}}{\mathbf{a}_i \mathbf{b}_j}\right)$$

Proposition: $\mathbf{P}_{i,j} = \mathbf{u}_i \, \mathbf{K}_{i,j} \, \mathbf{v}_j$ $\qquad \mathbf{K}_{i,j} \overset{\text{def.}}{=} e^{-\frac{d(x_i,y_j)^p}{\varepsilon}}$

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

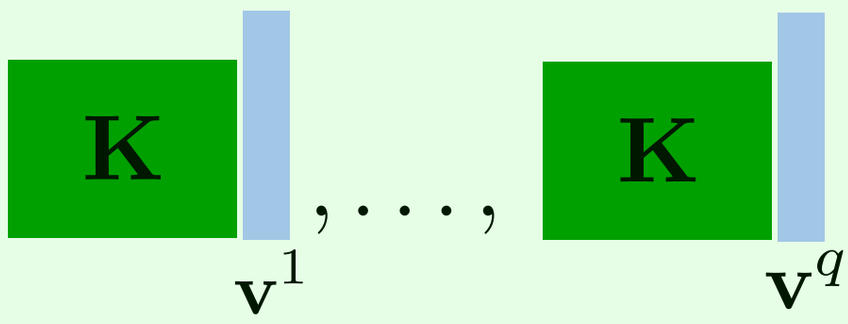
Col. constraint: $\mathbf{v} \odot (\mathbf{K}^\top \mathbf{u}) = \mathbf{b}$

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

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

Only matrix/vector multiplications.

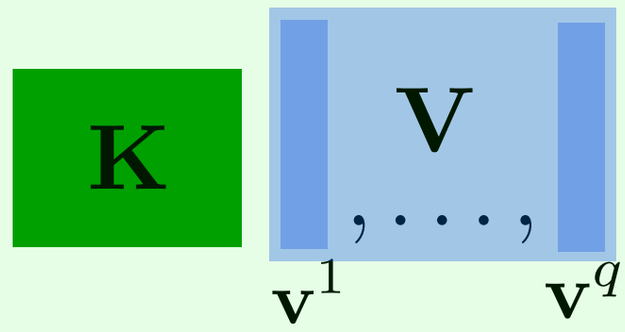
Matrix-vectors



parallelization

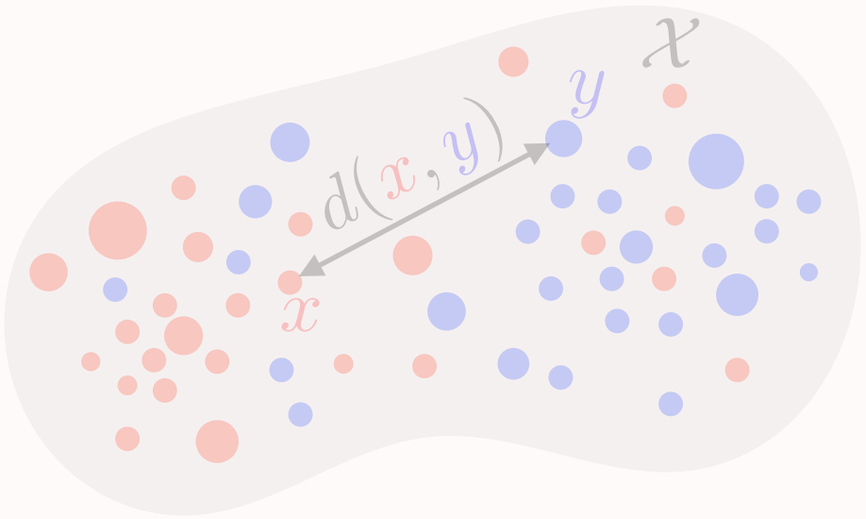
GPU

Matrix-matrix

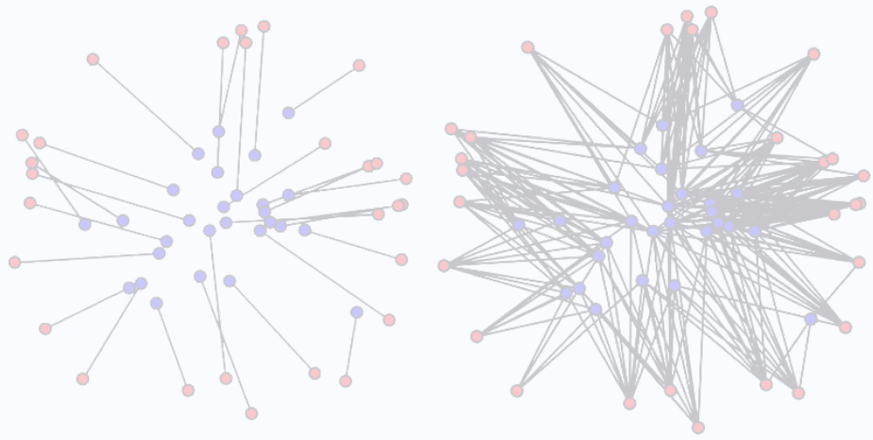


$\rightarrow$ Convolution on regular grids, separable kernels.

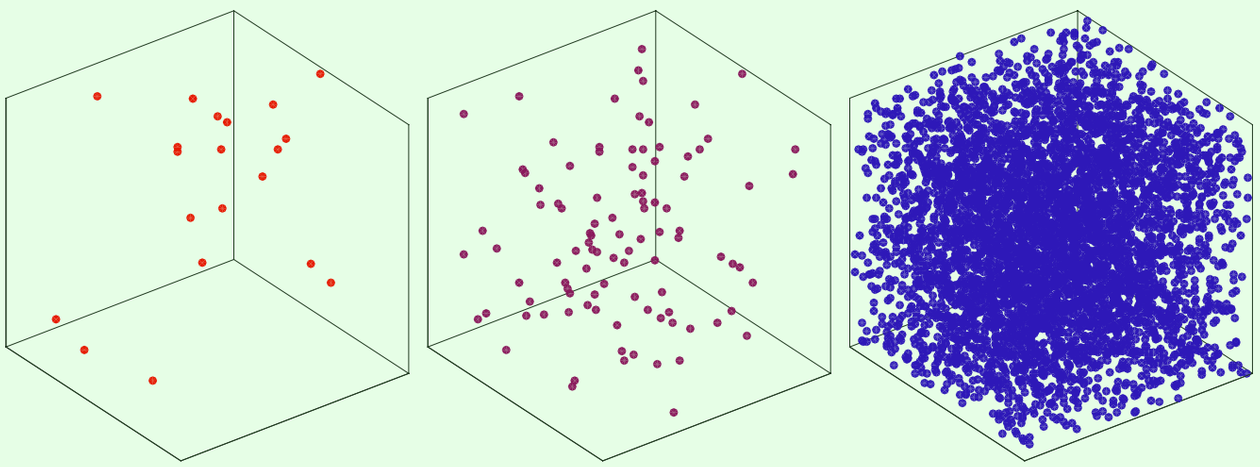
1. Optimal Transport



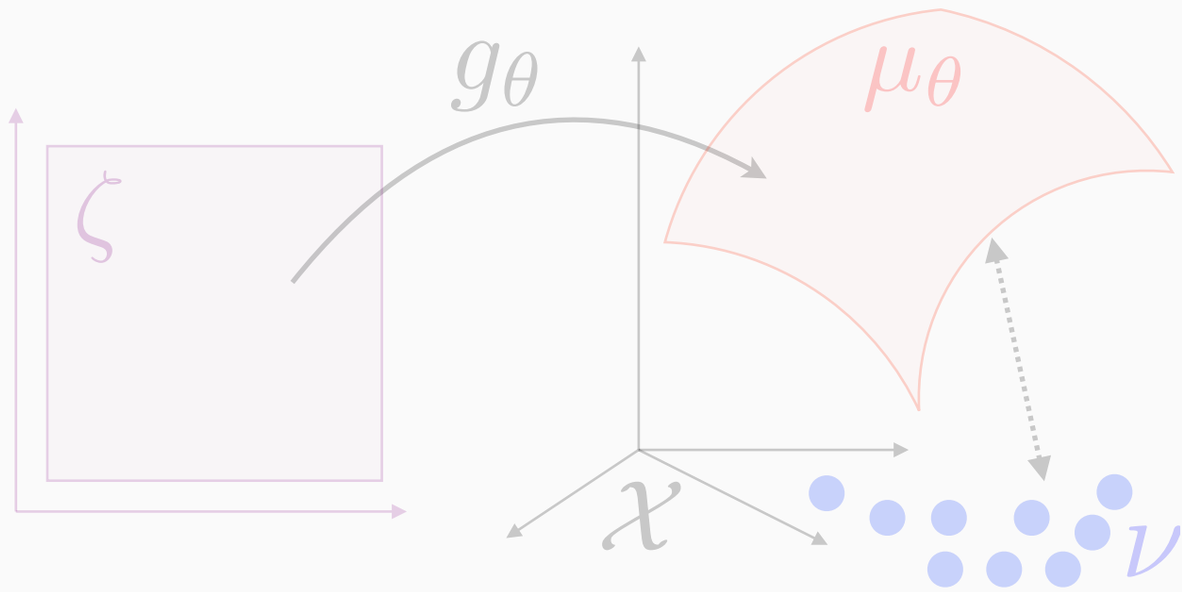
2. Entropic Regularization



3. Sinkhorn Divergences



4. Application to Generative Models

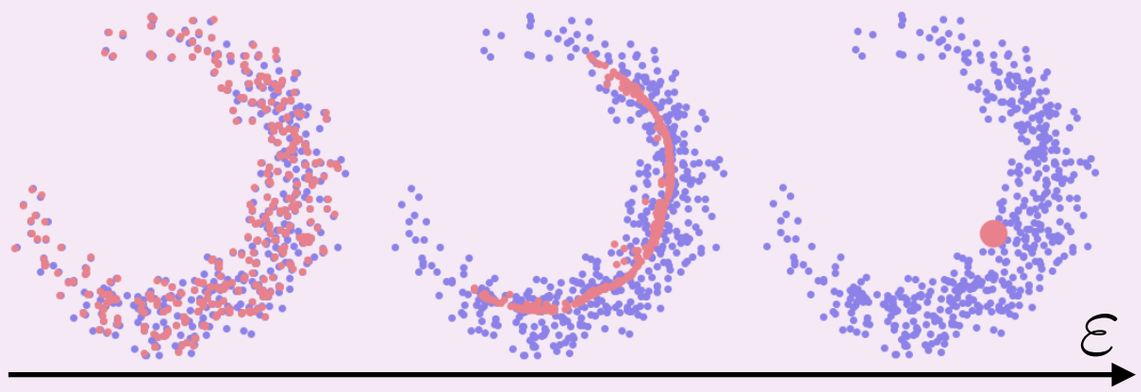


Sinkhorn Divergences

$$\mathrm{W}^p_{\varepsilon,p}(\alpha,\beta) \overset{\text{def.}}{=} \min_{\pi_1=\alpha,\pi_2=\beta} \int_{\mathcal{X}^2} d^p(x,y)\mathrm{d}\pi(x,y) + \varepsilon \mathrm{KL}(\pi|\xi)$$

Problem: $\mathrm{W}_\varepsilon(\alpha,\alpha) \neq 0$

$$\min_{\alpha} \mathrm{W}^p_{\varepsilon,p}(\alpha,\beta)$$

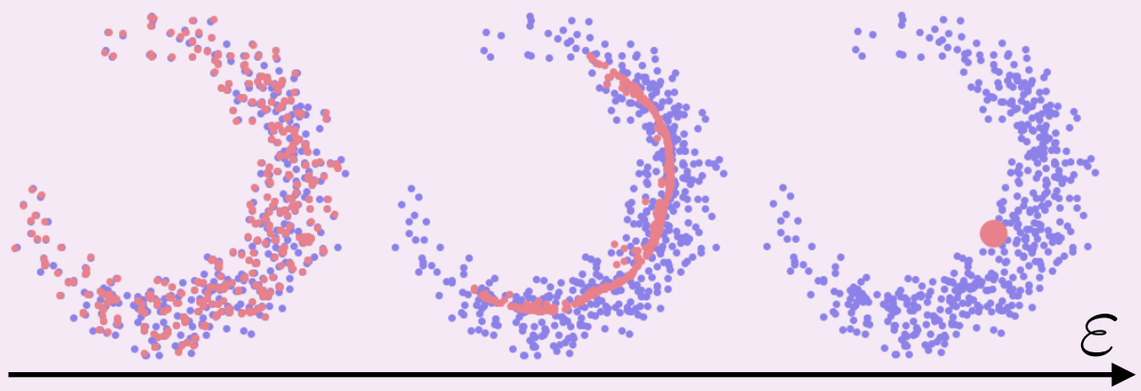


Sinkhorn Divergences

$$\mathrm{W}^p_{\varepsilon,p}(\alpha,\beta) \stackrel{\text{def.}}{=} \min_{\pi_1=\alpha,\pi_2=\beta} \int_{\mathcal{X}^2} d^p(x,y)\mathrm{d}\pi(x,y) + \varepsilon \mathrm{KL}(\pi|\xi)$$

Problem: $\mathrm{W}_\varepsilon(\alpha,\alpha) \neq 0$

$$\min_{\alpha} \mathrm{W}^p_{\varepsilon,p}(\alpha,\beta)$$



$$\overline{\mathrm{W}}^p_{p,\varepsilon}(\alpha,\beta) \stackrel{\text{def.}}{=} \mathrm{W}^p_{p,\varepsilon}(\alpha,\beta) - \tfrac{1}{2}\mathrm{W}^p_{p,\varepsilon}(\alpha,\alpha) - \tfrac{1}{2}\mathrm{W}^p_{p,\varepsilon}(\beta,\beta)$$

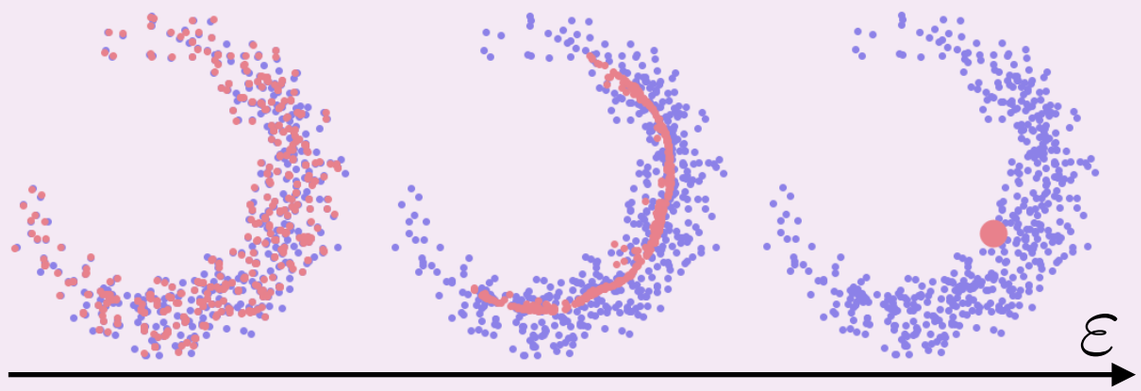
[Ramdas, García Trillos, Cuturi, 2017]

Sinkhorn Divergences

$$\mathrm{W}^p_{\varepsilon,p}(\alpha,\beta) \stackrel{\text{def.}}{=} \min_{\pi_1=\alpha,\pi_2=\beta} \int_{\mathcal{X}^2} d^p(x,y)\mathrm{d}\pi(x,y) + \varepsilon \mathrm{KL}(\pi|\xi)$$

Problem: $\mathrm{W}_\varepsilon(\alpha,\alpha) \neq 0$

$$\min_\alpha \mathrm{W}^p_{\varepsilon,p}(\alpha,\beta)$$



$$\overline{\mathrm{W}}^p_{p,\varepsilon}(\alpha,\beta) \stackrel{\text{def.}}{=} \mathrm{W}^p_{p,\varepsilon}(\alpha,\beta) - \tfrac{1}{2}\mathrm{W}^p_{p,\varepsilon}(\alpha,\alpha) - \tfrac{1}{2}\mathrm{W}^p_{p,\varepsilon}(\beta,\beta)$$

[Ramdas, García Trillos, Cuturi, 2017]

Theorem: $\mathrm{W}^p_p(\alpha,\beta) \xleftarrow{\varepsilon \to 0} \overline{\mathrm{W}}^p_{\varepsilon,p}(\alpha,\beta) \xrightarrow{\varepsilon \to +\infty} \|\alpha-\beta\|^2_{-d^p}$

[Léonard 2012] [Carlier et al 2017]

[Ramdas, García Trillos, Cuturi, 2017]

Kernel norms (MMD): $\|\xi\|^2_{-d^p} \stackrel{\text{def.}}{=} -\int_{\mathcal{X}^2} d(x,y)^p \mathrm{d}\xi(x)\mathrm{d}\xi(y)$

Proposition: $\|\cdot\|_{-\|\cdot\|^p}$ is a norm for $0 < p < 2$.

Sinkhorn Divergences

$$\overline{\mathrm{W}}^p_{p,\varepsilon}(\alpha,\beta) \overset{\text{def.}}{=} \mathrm{W}^p_{p,\varepsilon}(\alpha,\beta) - \tfrac{1}{2}\mathrm{W}^p_{p,\varepsilon}(\alpha,\alpha) - \tfrac{1}{2}\mathrm{W}^p_{p,\varepsilon}(\beta,\beta)$$

concave (under $\mathrm{W}^p_{p,\varepsilon}(\alpha,\alpha)$) concave (under $\mathrm{W}^p_{p,\varepsilon}(\beta,\beta)$)

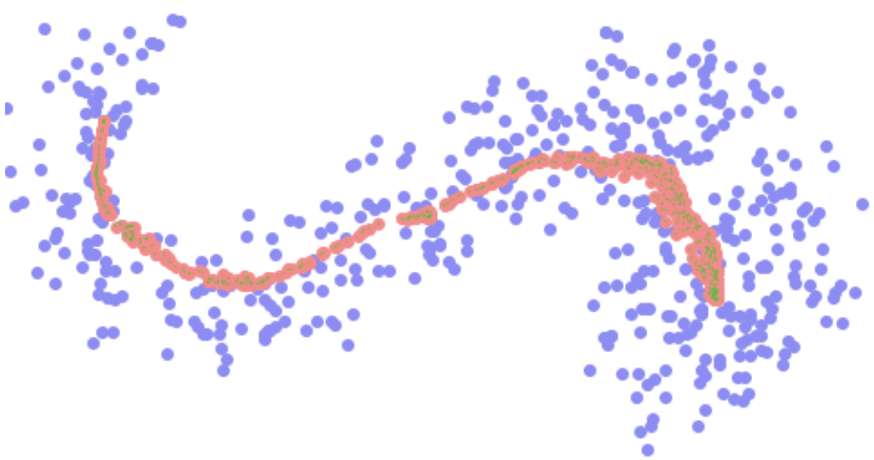
Theorem: [Feydy, Séjourné, P, Vialard, Trouvé, Amari 2018]

If $e^{-\frac{d^p}{\varepsilon}}$ is positive:

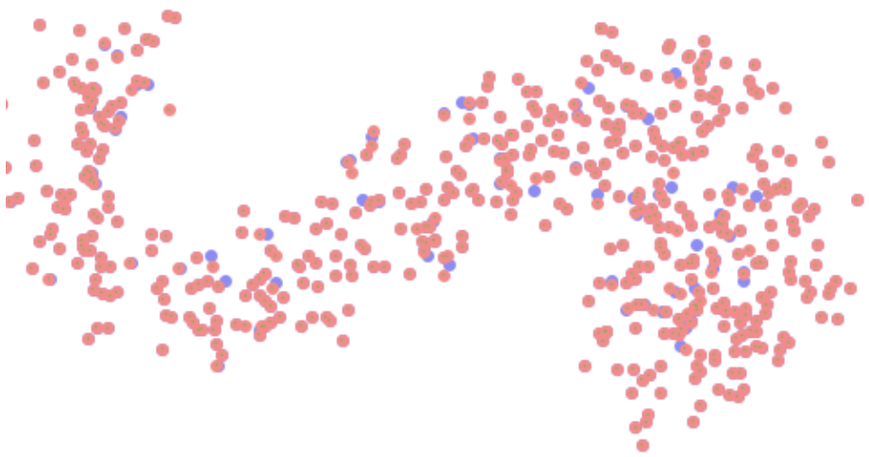
$$\overline{\mathrm{W}}_{\varepsilon,p} \geqslant 0 \text{ and } \overline{\mathrm{W}}^p_{\varepsilon,p}(\cdot,\beta) \text{ is convex.}$$

$$\overline{\mathrm{W}}_{\varepsilon,p}(\alpha_n,\beta) \to 0 \iff \alpha_n \overset{\text{weak}*}{\rightharpoonup} \beta$$

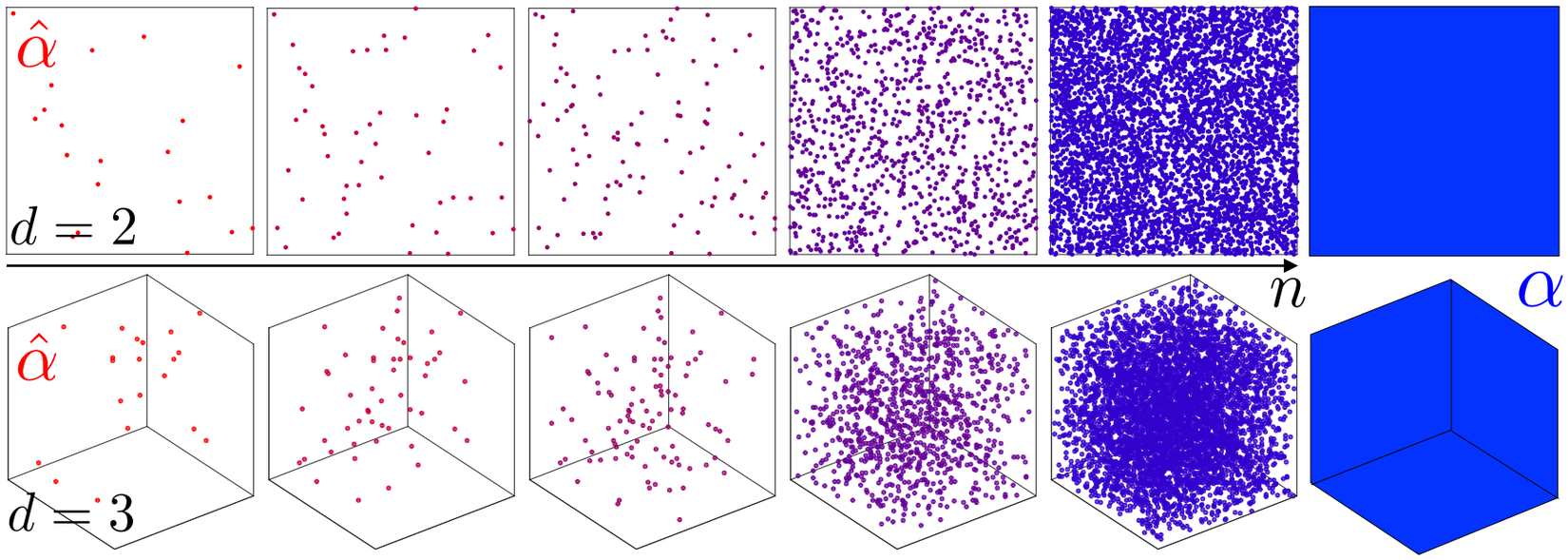
$\min_\alpha \mathrm{W}^p_{\varepsilon,p}(\alpha,\beta)$

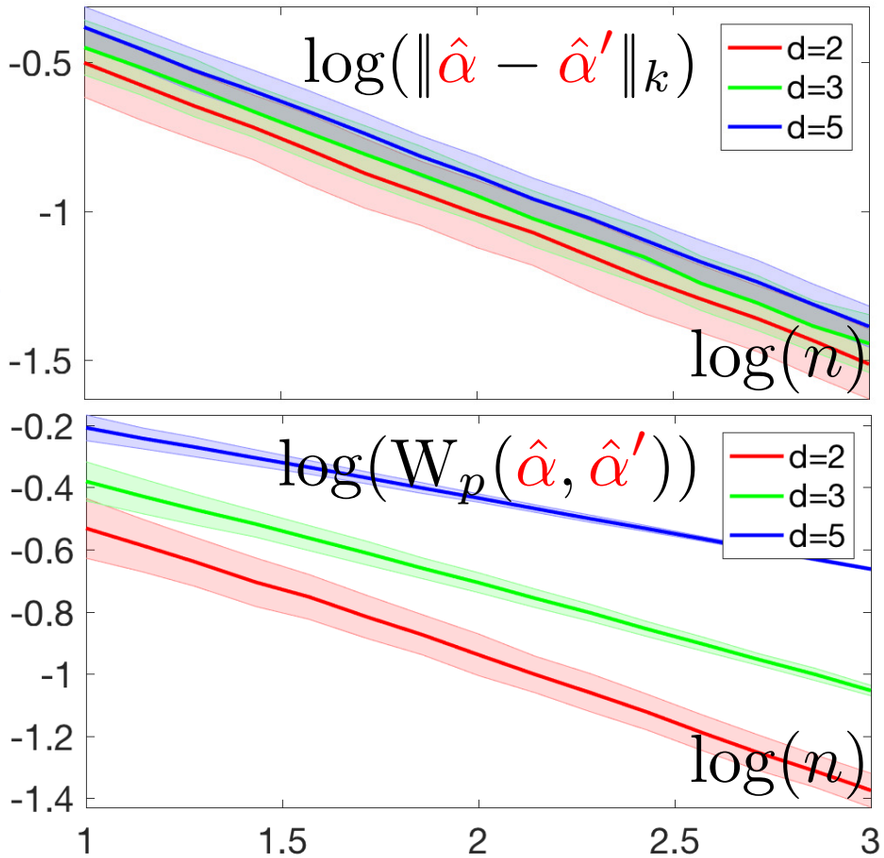


$\min_\alpha \overline{\mathrm{W}}^p_{\varepsilon,p}(\alpha,\beta)$



Sample Complexity





Theorem:

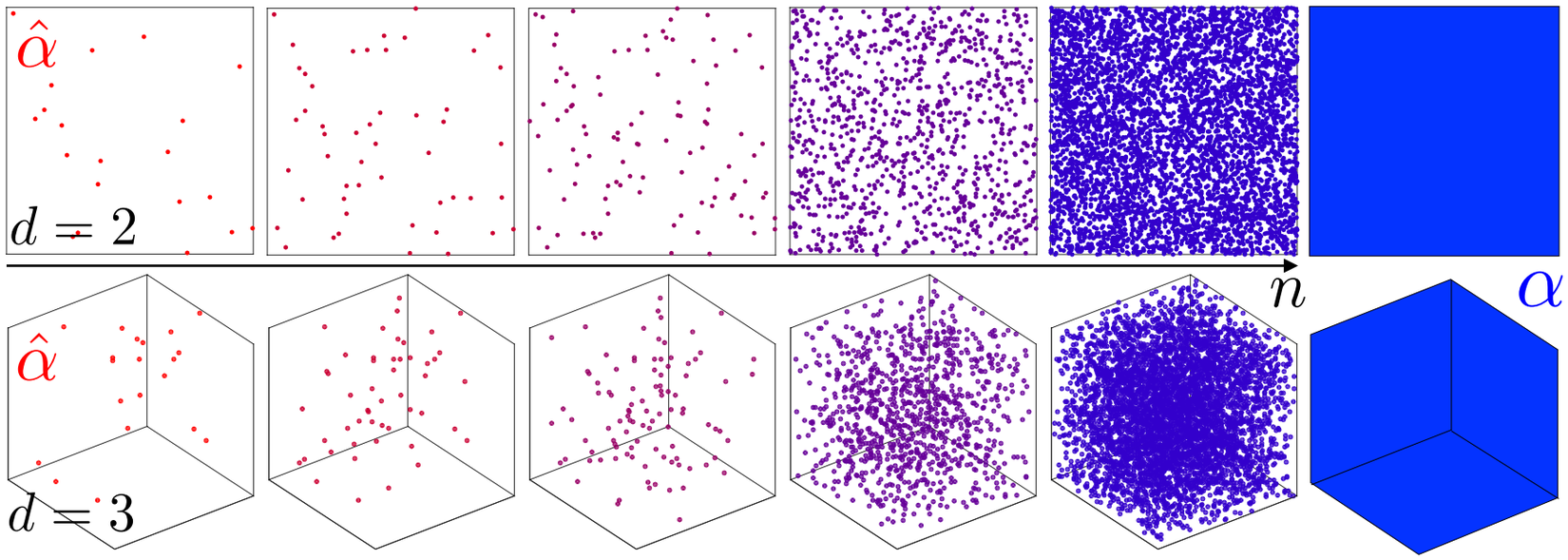
$$\mathbb{E}(|\mathrm{W}_p(\hat{\alpha}, \hat{\beta}) - \mathrm{W}_p(\alpha, \beta)|) = O(n^{-\frac{1}{d}})$$

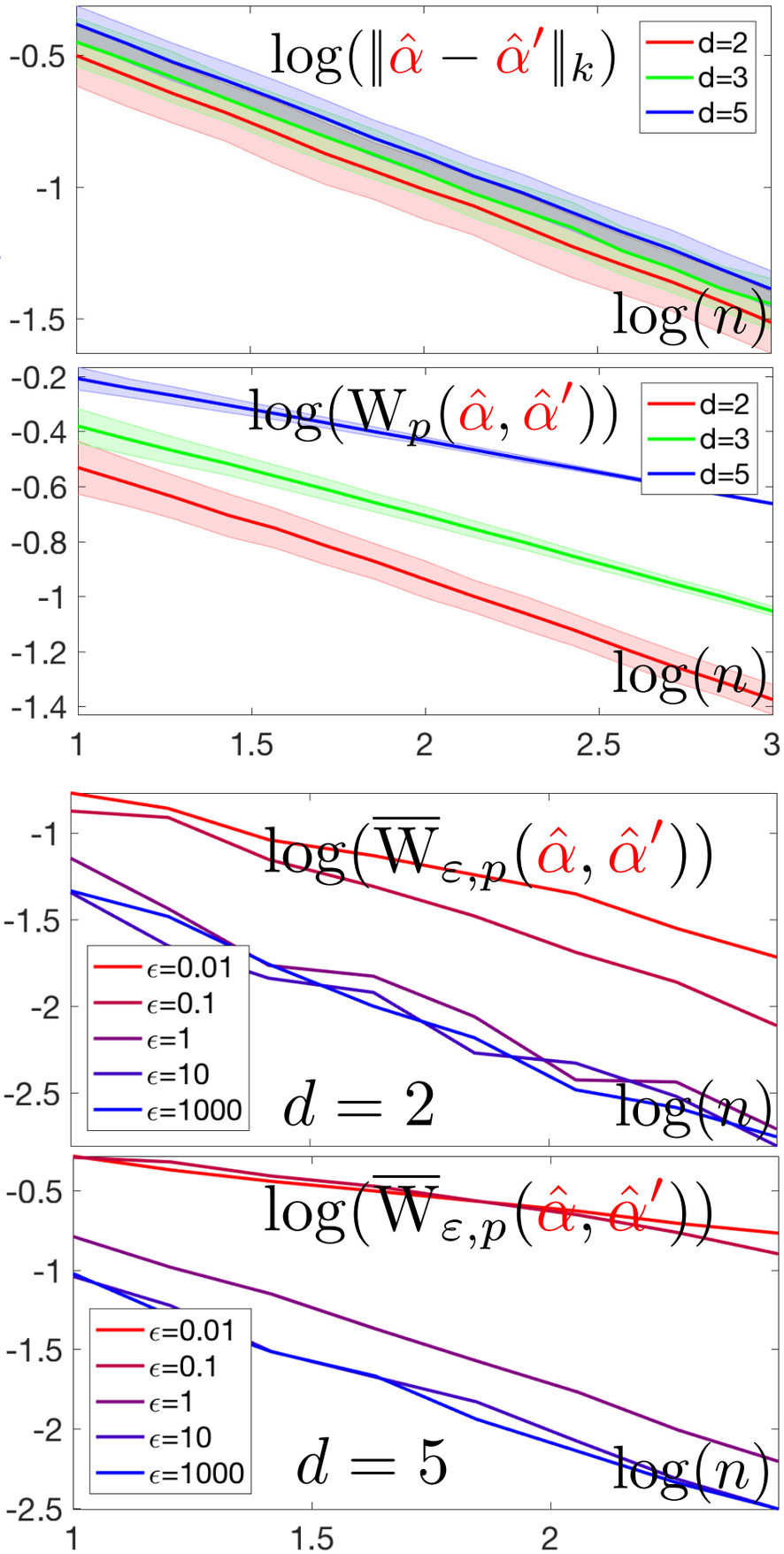
$$\mathbb{E}(|\|\hat{\alpha} - \hat{\beta}\|_k - \|\alpha - \beta\|_k|) = O(n^{-\frac{1}{2}})$$

Optimal transport: suffers from curse of dimensionality.

$\rightarrow$ Adapt to support dimensionality [Weed, Bach 2017]

Sample Complexity





Theorem:

$$\mathbb{E}(|\mathrm{W}_p(\hat{\alpha}, \hat{\beta}) - \mathrm{W}_p(\alpha, \beta)|) = O(n^{-\frac{1}{d}})$$

$$\mathbb{E}(|\|\hat{\alpha} - \hat{\beta}\|_k - \|\alpha - \beta\|_k|) = O(n^{-\frac{1}{2}})$$

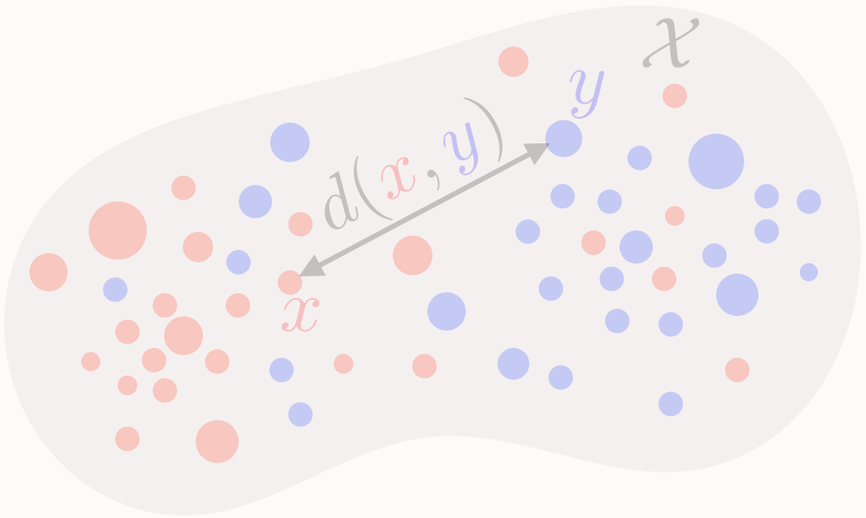
Optimal transport: suffers from curse of dimensionality.

$\rightarrow$ Adapt to support dimensionality [Weed, Bach 2017]

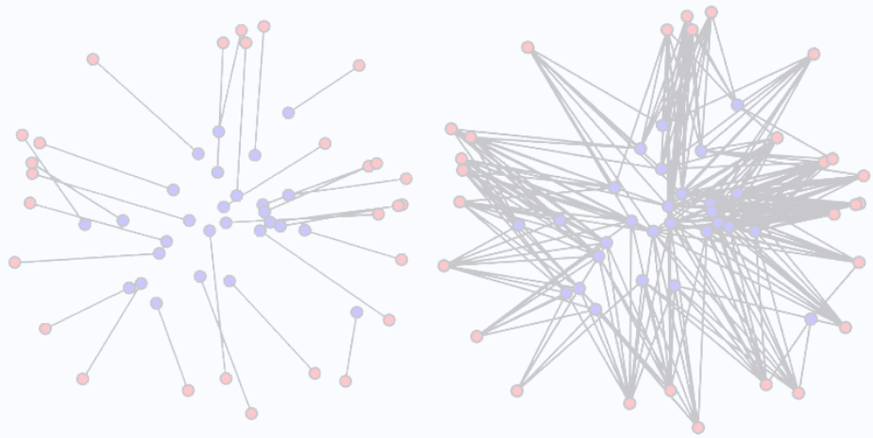
Theorem: [Genevay, Bach, P, Cuturi]

$$\mathbb{E}(|\overline{\mathrm{W}}_{\varepsilon,p}(\hat{\alpha}, \hat{\beta}) - \overline{\mathrm{W}}_{\varepsilon,p}(\alpha, \beta)|) = O(\varepsilon^{-\frac{d}{2}} n^{-\frac{1}{2}})$$

1. Optimal Transport

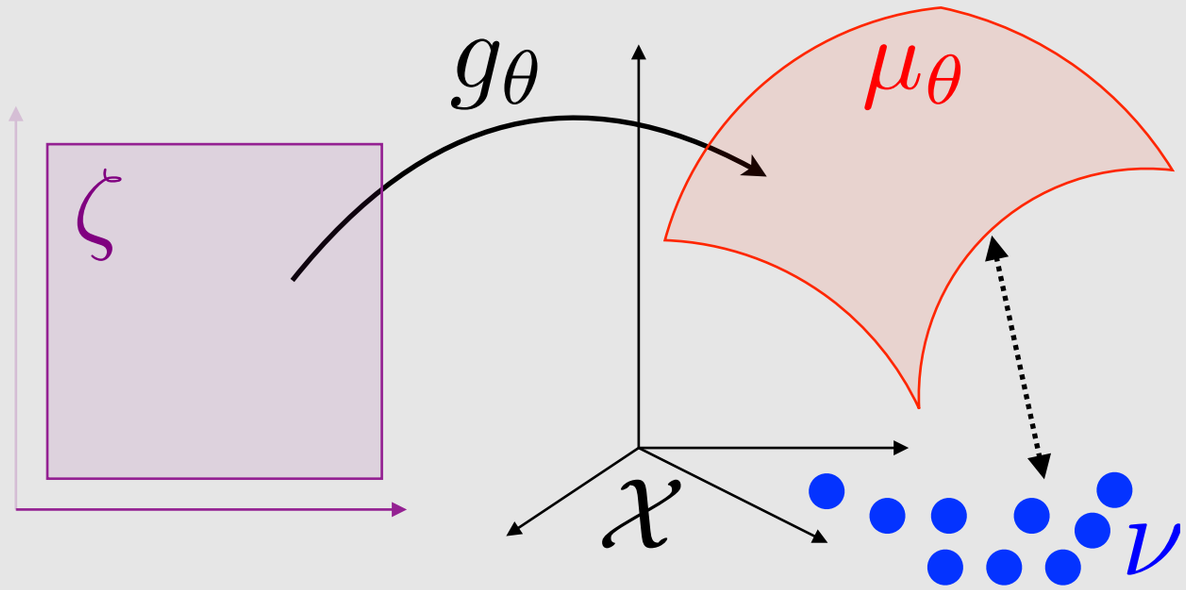


2. Entropic Regularization



3. Sinkhorn Divergences

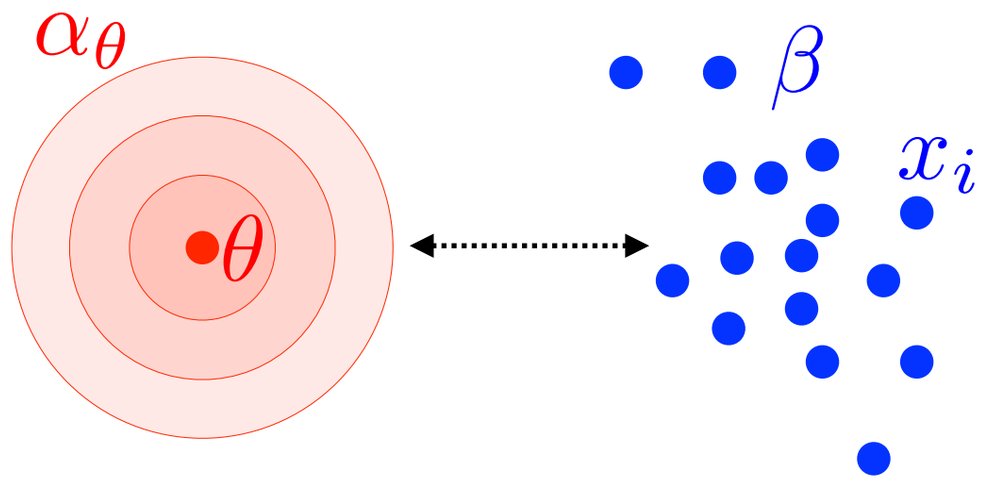
4. Application to Generative Models



Density Fitting and Generative Models

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

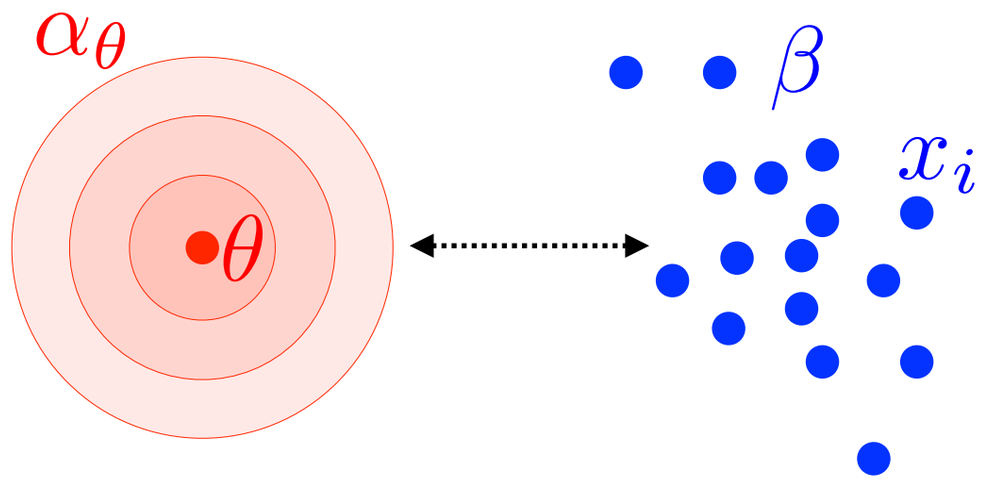
$$\textit{Parametric model: } \theta \mapsto \alpha_\theta$$



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

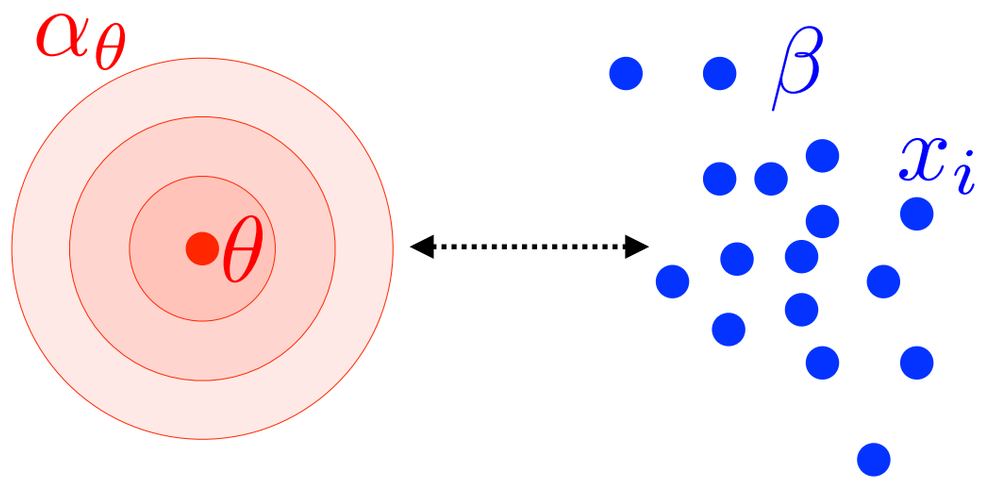
$$\min_\theta \widehat{\mathrm{KL}}(\alpha_\theta|\beta) \overset{\text{def.}}{=} -\sum_i \log(\rho_\theta(x_i))$$

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

$$\min_\theta \widehat{\mathrm{KL}}(\alpha_\theta|\beta) \overset{\text{def.}}{=} -\sum_i \log(\rho_\theta(x_i))$$

Maximum likelihood (MLE)

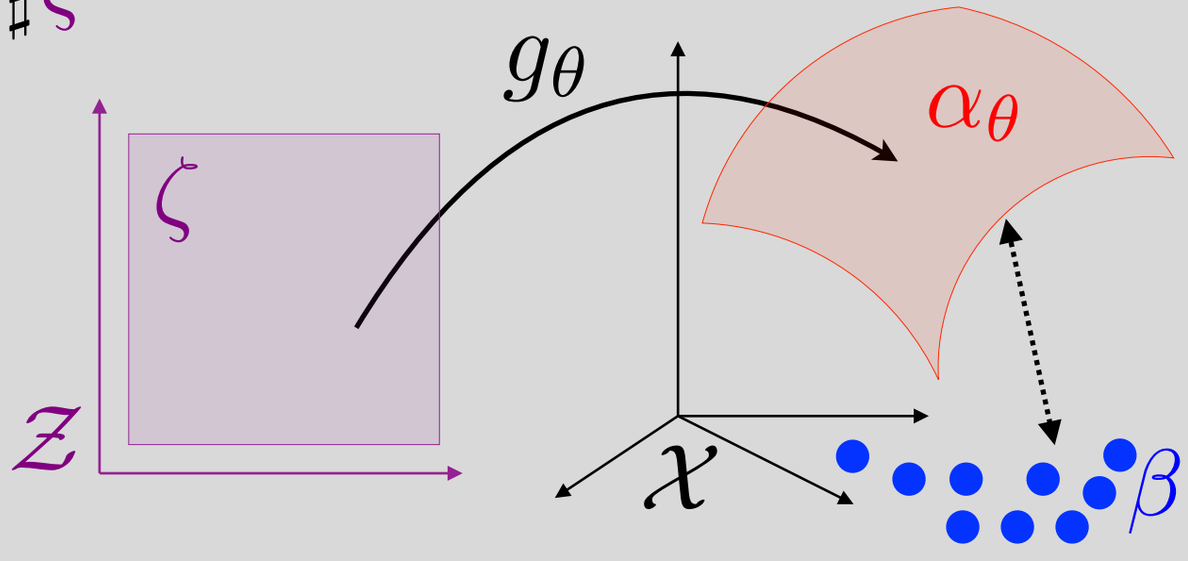
Generative model fit: $\alpha_\theta = g_{\theta,\sharp}\zeta$

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

$\rightarrow$ MLE undefined.

$\rightarrow$ Need a weaker metric.

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

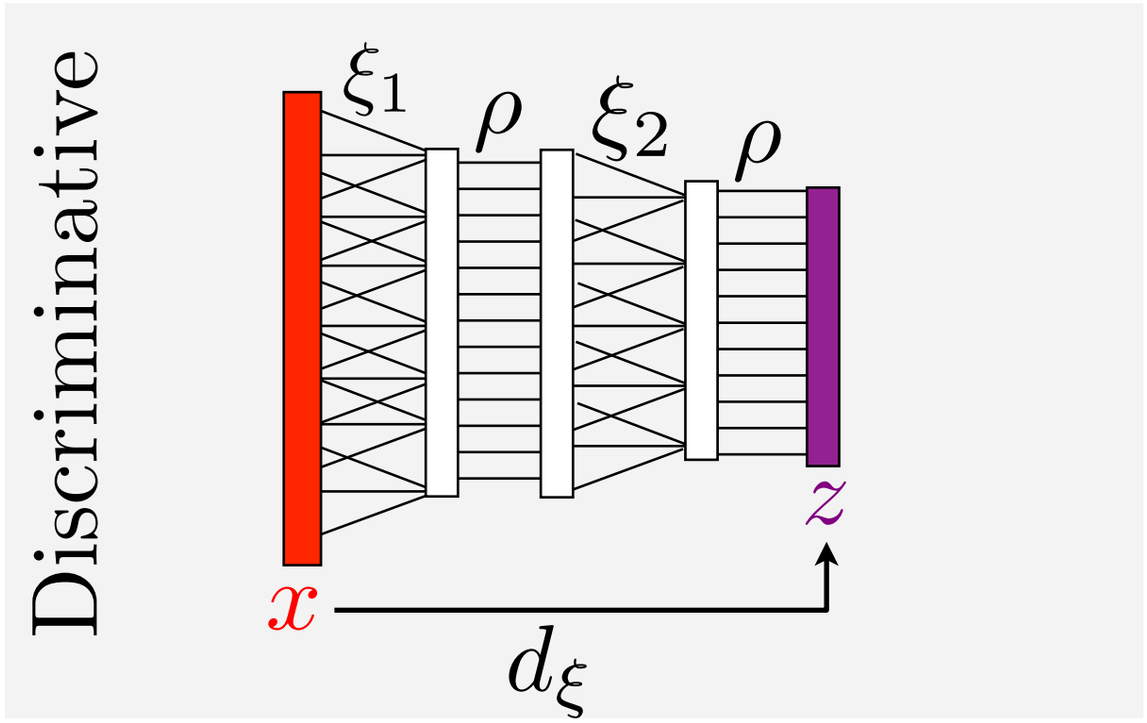


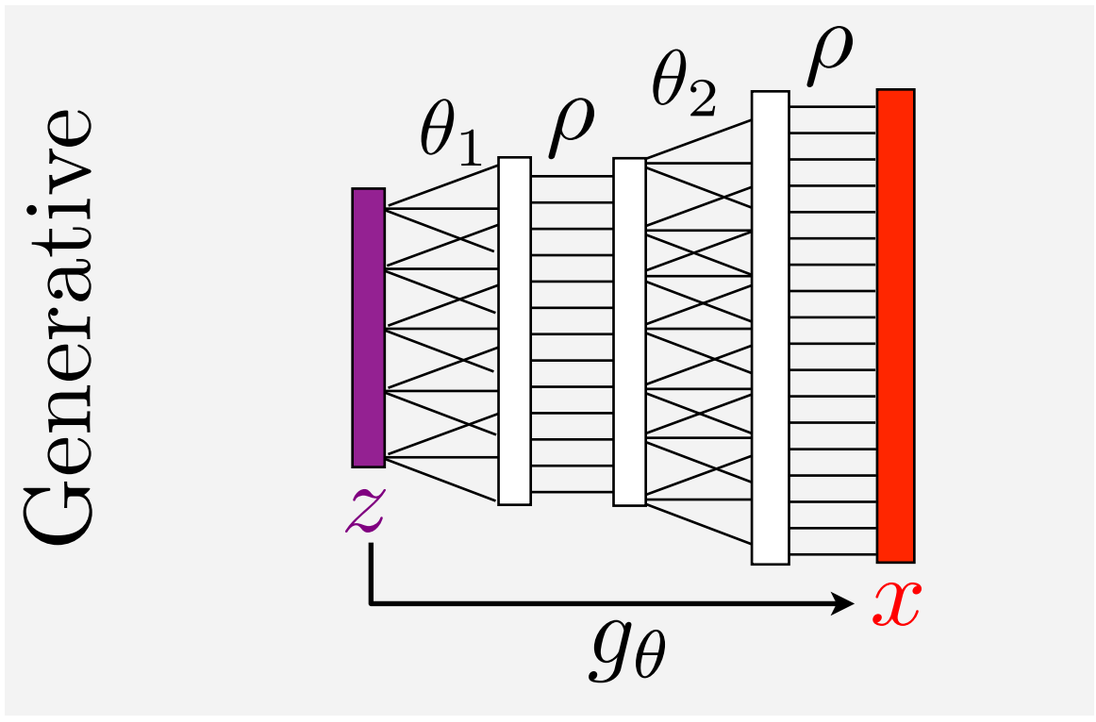
Deep Discriminative vs Generative Models

Deep networks:

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

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

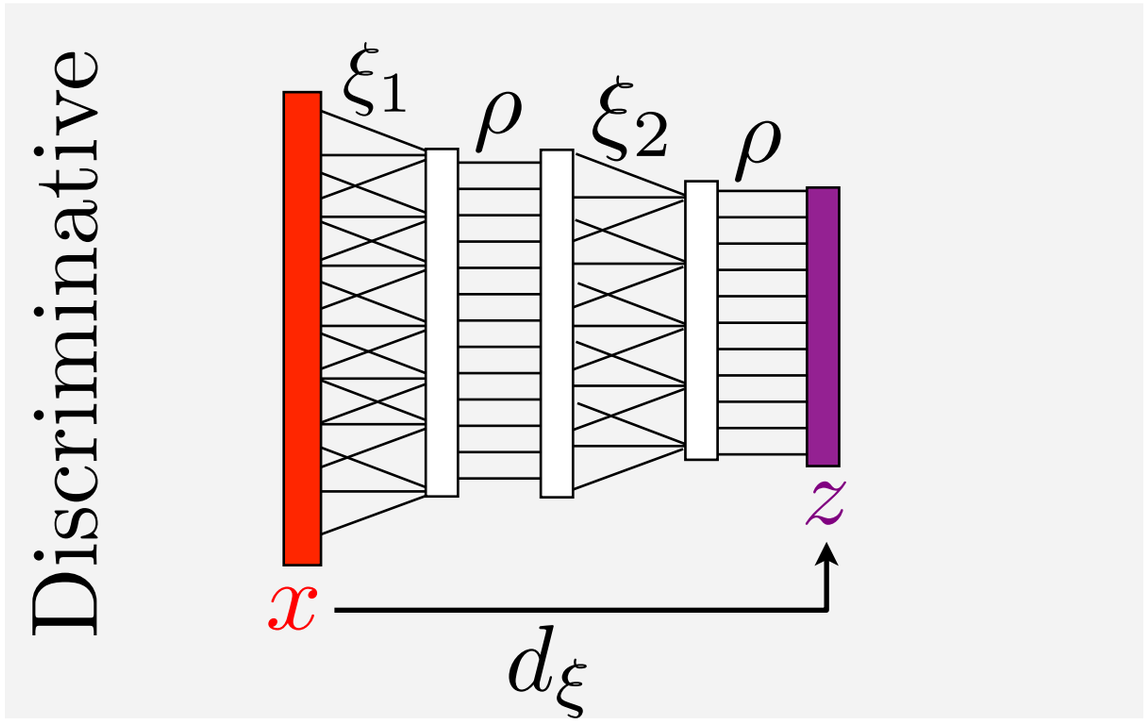


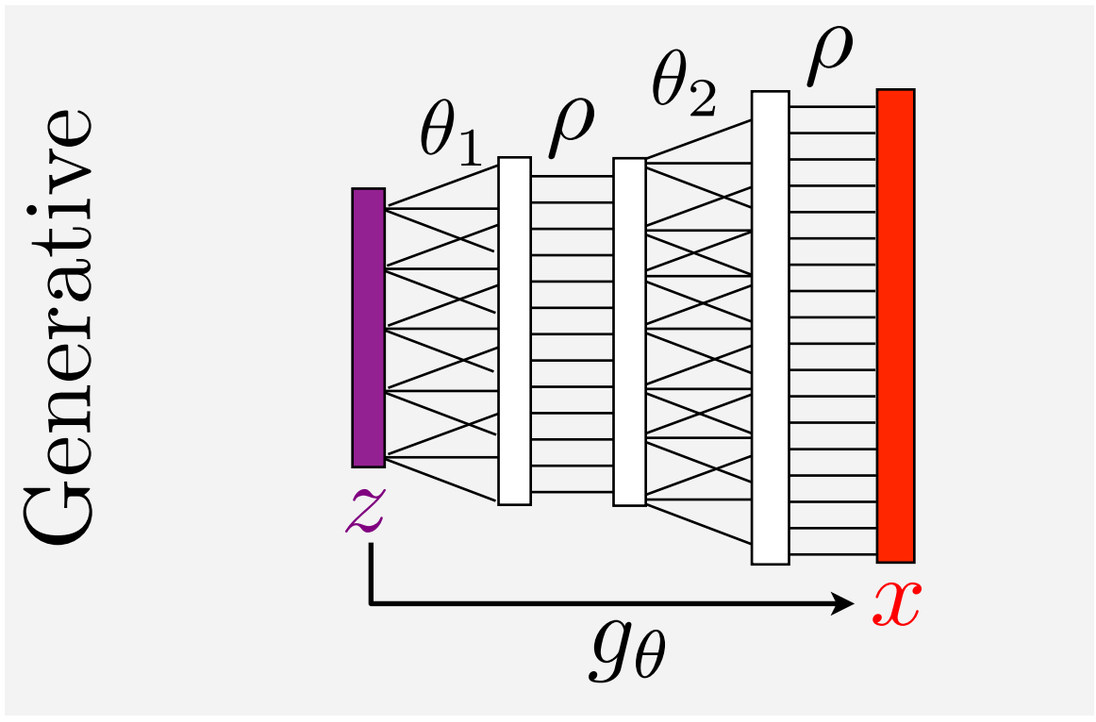


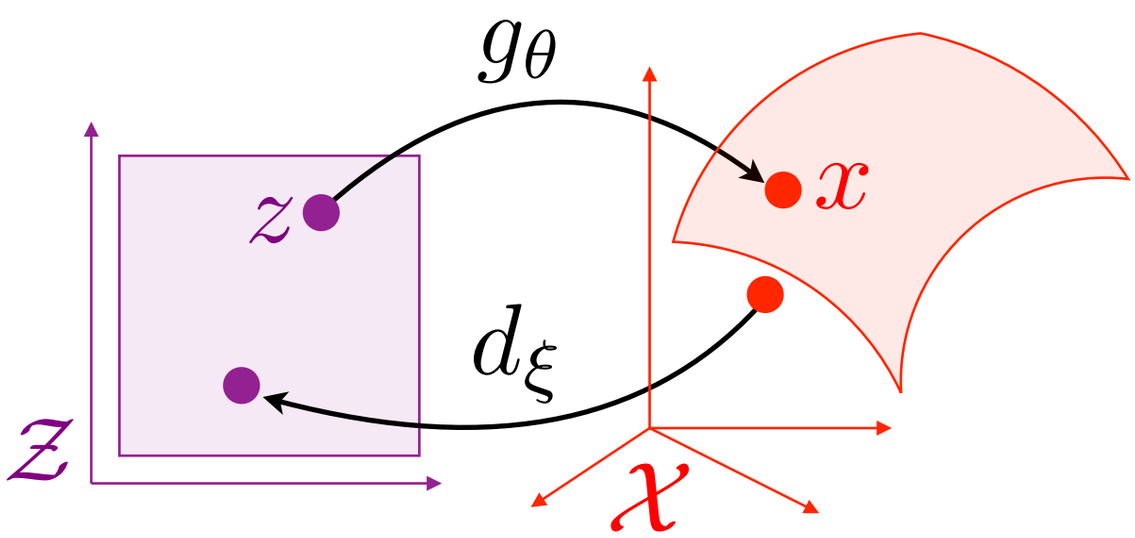
Deep Discriminative vs Generative Models

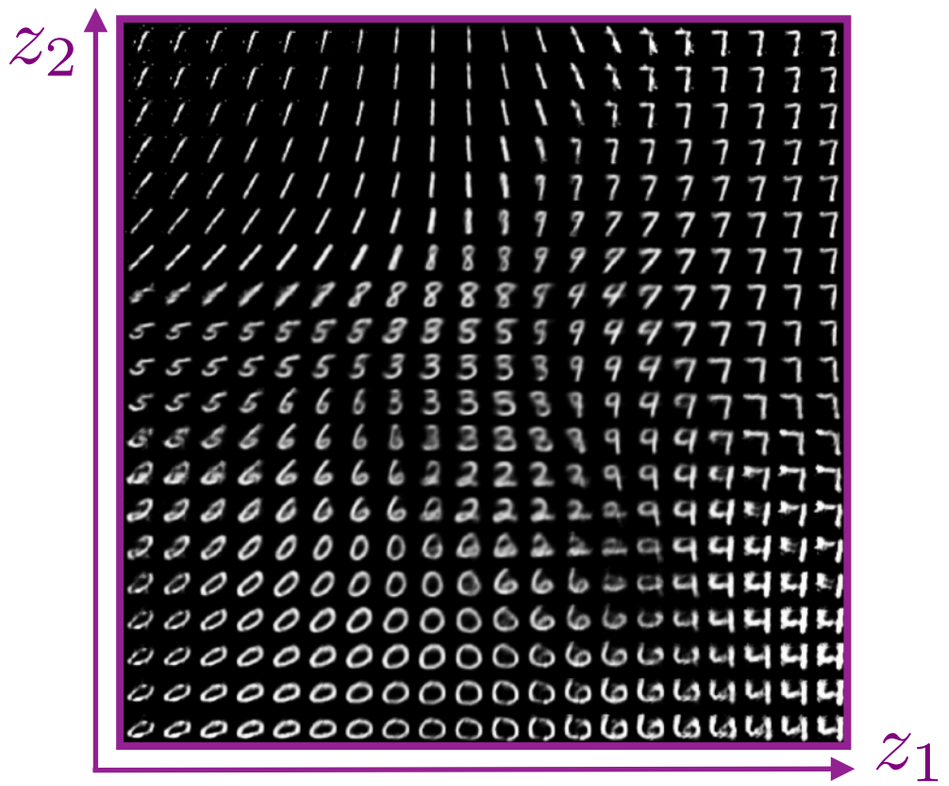
Deep networks:

$$d_\xi(x) = \rho(\xi_K(\dots\rho(\xi_2(\rho(\xi_1(x)\dots)$$
$$g_\theta(z) = \rho(\theta_K(\dots\rho(\theta_2(\rho(\theta_1(z)\dots)$$

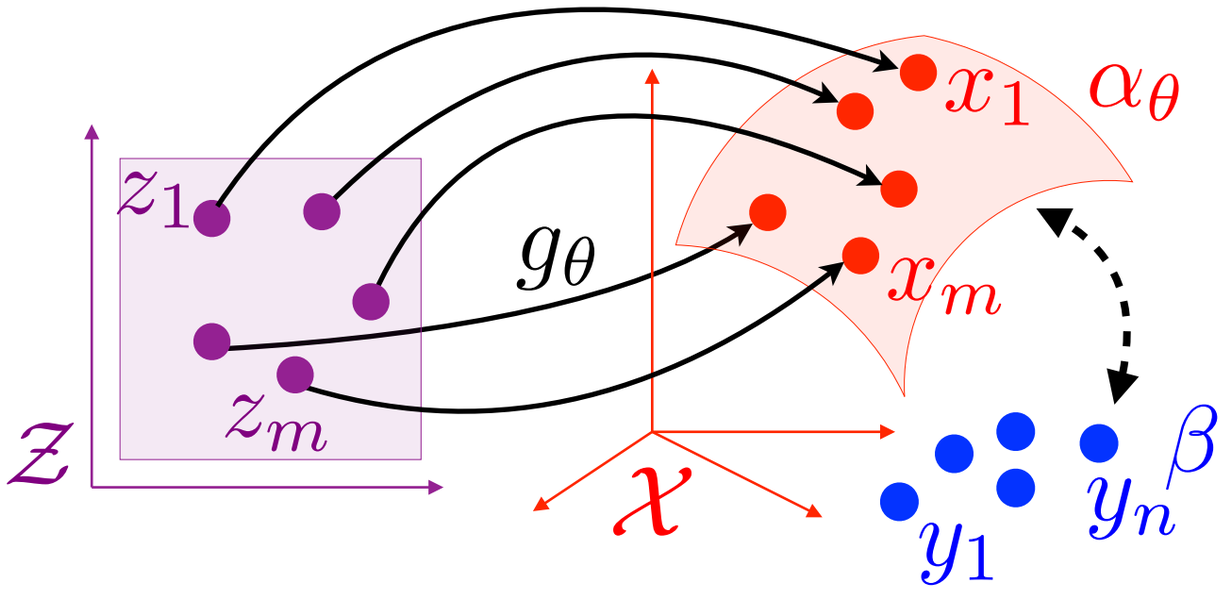








Training Architecture



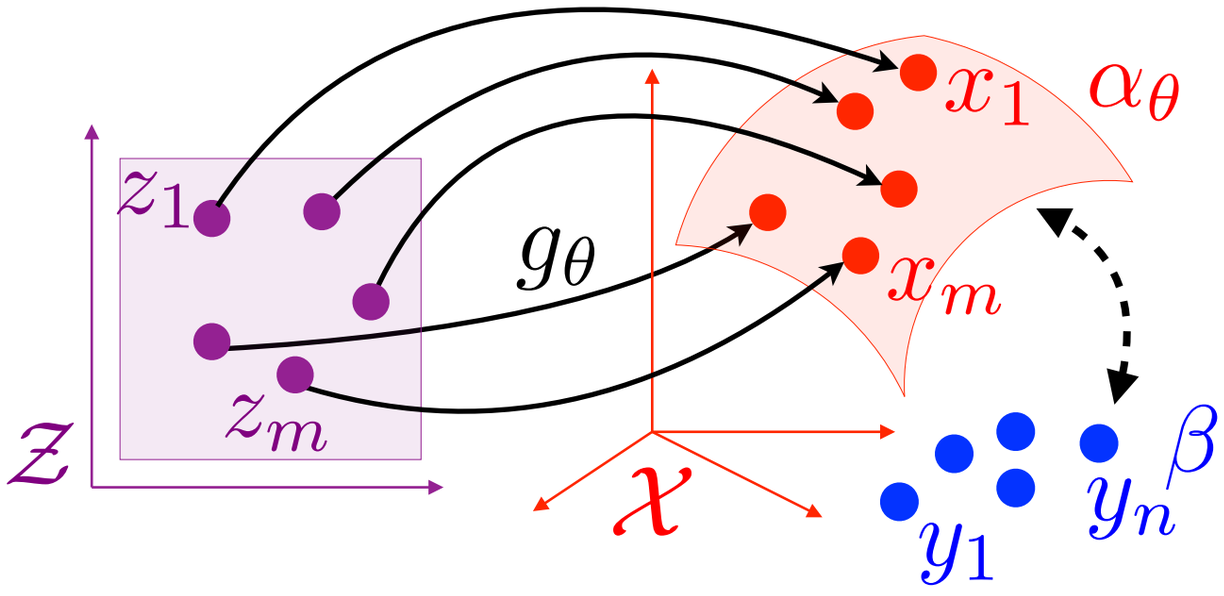
$$\min_\theta \mathcal{E}(\theta) \overset{\text{def.}}{=} \overline{\mathrm{W}}^p_{\varepsilon,p}(\alpha_\theta, \beta)$$

Stochastic gradient descent

$$\theta \leftarrow \theta - \tau \nabla \hat{\mathcal{E}}(\theta)$$

$$\hat{\mathcal{E}}(\theta) \overset{\text{def.}}{=} \overline{\mathrm{W}}^p_{\varepsilon,p}(\tfrac{1}{m}\textstyle\sum_i \delta_{g_\theta(z_i)}, \beta)$$

Training Architecture

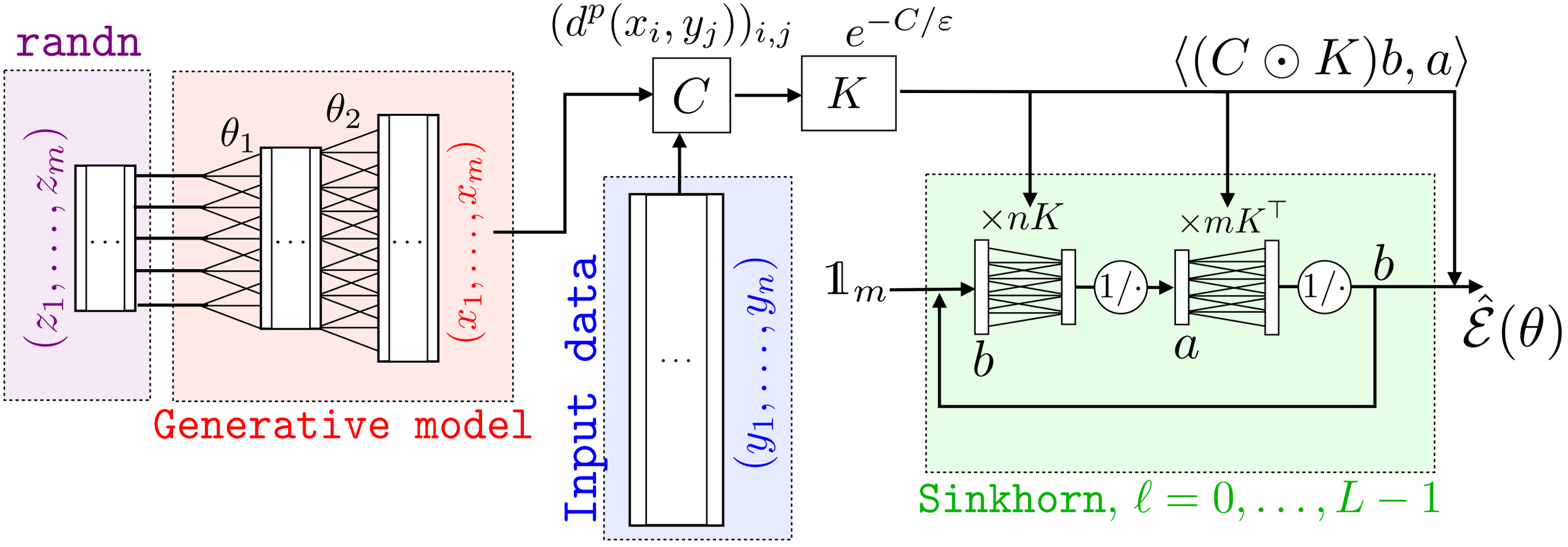


$$\min_\theta \mathcal{E}(\theta) \overset{\text{def.}}{=} \overline{\mathrm{W}}^p_{\varepsilon,p}(\alpha_\theta, \beta)$$

Stochastic gradient descent

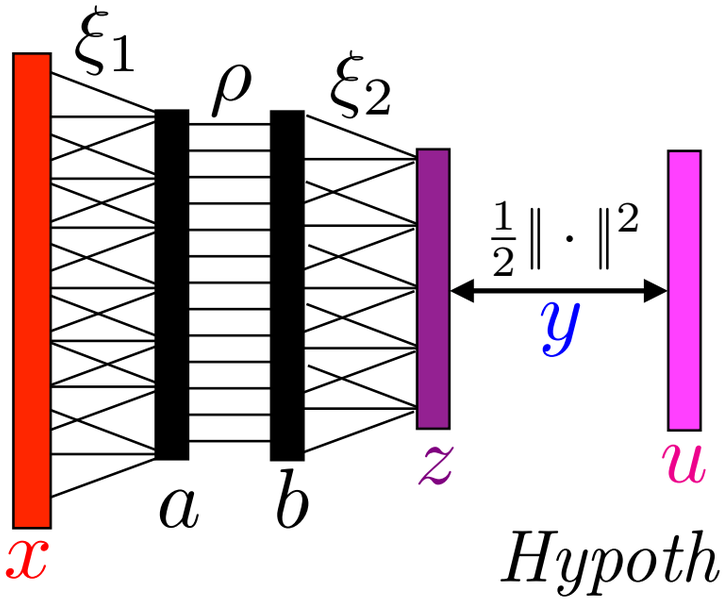
$$\theta \leftarrow \theta - \tau \nabla \hat{\mathcal{E}}(\theta)$$

$$\hat{\mathcal{E}}(\theta) \overset{\text{def.}}{=} \overline{\mathrm{W}}^p_{\varepsilon,p}(\tfrac{1}{m}\textstyle\sum_i \delta_{g_\theta(z_i)}, \beta)$$



Automatic Differentiation

Setup: $\mathcal{E} : \mathbb{R}^n \to \mathbb{R}$ computable in K operations.



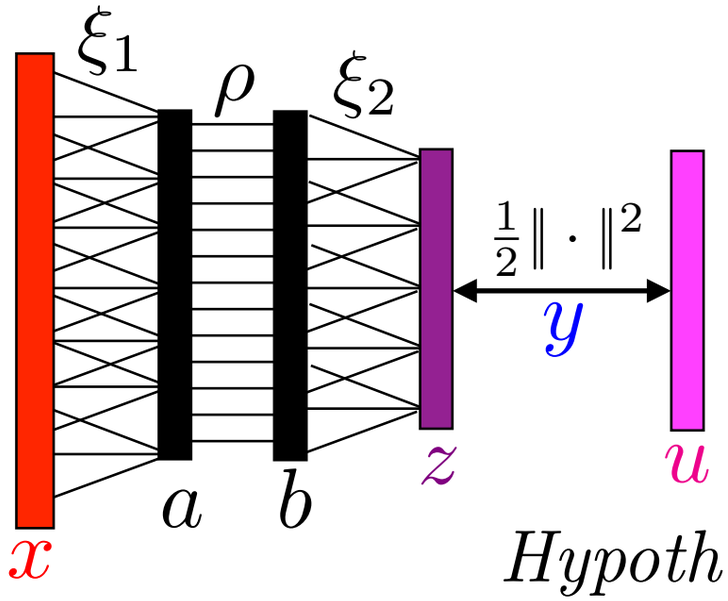

```
function y = E(x)
 a = xi1*x
 b = rho(a)
 z = xi2*b
 y = 1/2*norm(z-u)^2
```

Hypothesis: elementary operations $(a \times b, \log(a), \sqrt{a} \ldots)$ and their derivatives cost $O(1)$.

Question: What is the complexity of computing $\nabla\mathcal{E} : \mathbb{R}^n \to \mathbb{R}^n$?

Automatic Differentiation

Setup: $\mathcal{E} : \mathbb{R}^n \to \mathbb{R}$ computable in K operations.




```
function y = E(x)
 a = xi1*x
 b = rho(a)
 z = xi2*b
 y = 1/2*norm(z-u)^2
```

Hypothesis: elementary operations $(a \times b, \log(a), \sqrt{a} \ldots)$ and their derivatives cost $O(1)$.

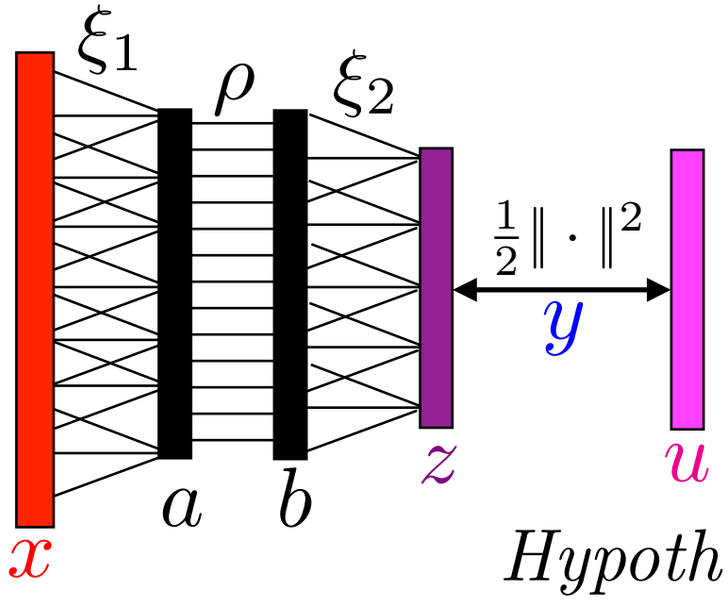
Question: What is the complexity of computing $\nabla\mathcal{E} : \mathbb{R}^n \to \mathbb{R}^n$?

Finite differences: $\nabla\mathcal{E}(\theta) \approx \frac{1}{\varepsilon}(\mathcal{E}(\theta + \varepsilon\delta_1) - \mathcal{E}(\theta), \ldots \mathcal{E}(\theta + \varepsilon\delta_n) - \mathcal{E}(\theta))$

$K(n+1)$ operations, intractable for large n.

Automatic Differentiation

Setup: $\mathcal{E} : \mathbb{R}^n \to \mathbb{R}$ computable in K operations.




```
function y = E(x)
 a = xi1*x
 b = rho(a)
 z = xi2*b
 y = 1/2*norm(z-u)^2
```

```
function dx = nablaE(x)
 dz = z-u
 db = xi2'*dz
 da = diag(dphi(a)) * db
 dx = xi1'*da
```

Hypothesis: elementary operations $(a \times b, \log(a), \sqrt{a} \ldots)$ and their derivatives cost $O(1)$.

Question: What is the complexity of computing $\nabla\mathcal{E} : \mathbb{R}^n \to \mathbb{R}^n$?

Finite differences: $\nabla\mathcal{E}(\theta) \approx \frac{1}{\varepsilon}(\mathcal{E}(\theta + \varepsilon\delta_1) - \mathcal{E}(\theta), \ldots \mathcal{E}(\theta + \varepsilon\delta_n) - \mathcal{E}(\theta))$

$K(n+1)$ operations, intractable for large n.

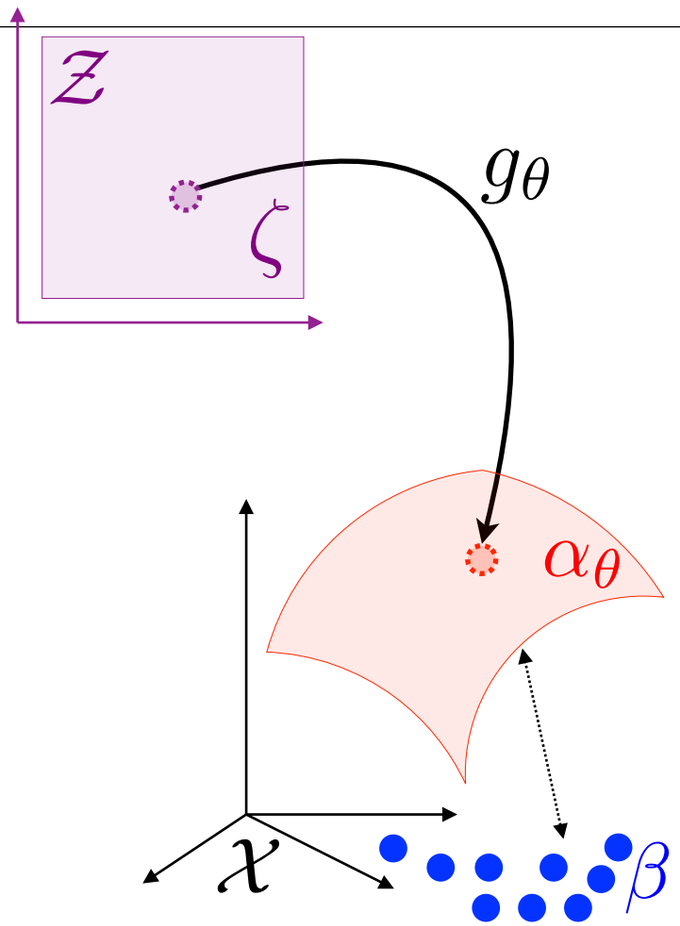
Theorem: there is an algorithm to compute $\nabla\mathcal{E}$ in $O(K)$ operations. [Seppo Linnainmaa, 1970]

This algorithm is reverse mode automatic differentiation

Examples of Images Generation

Inputs β

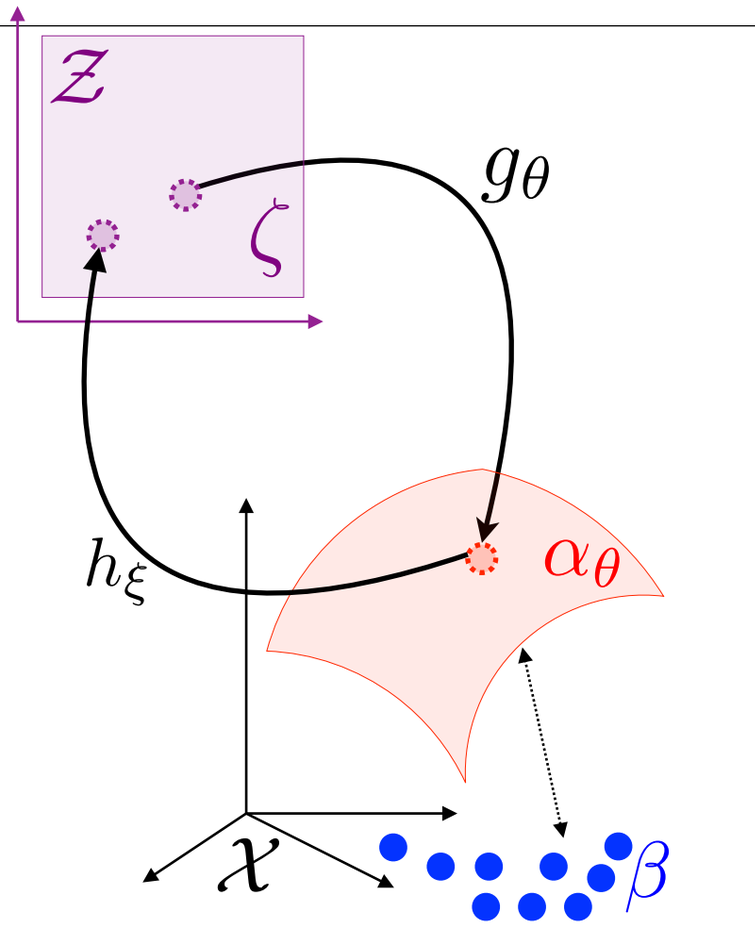
Generated α_θ

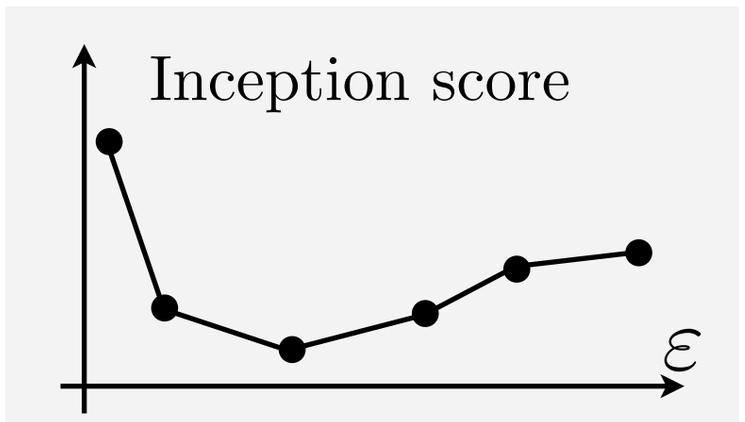


Examples of Images Generation

Inputs β

Generated α_θ



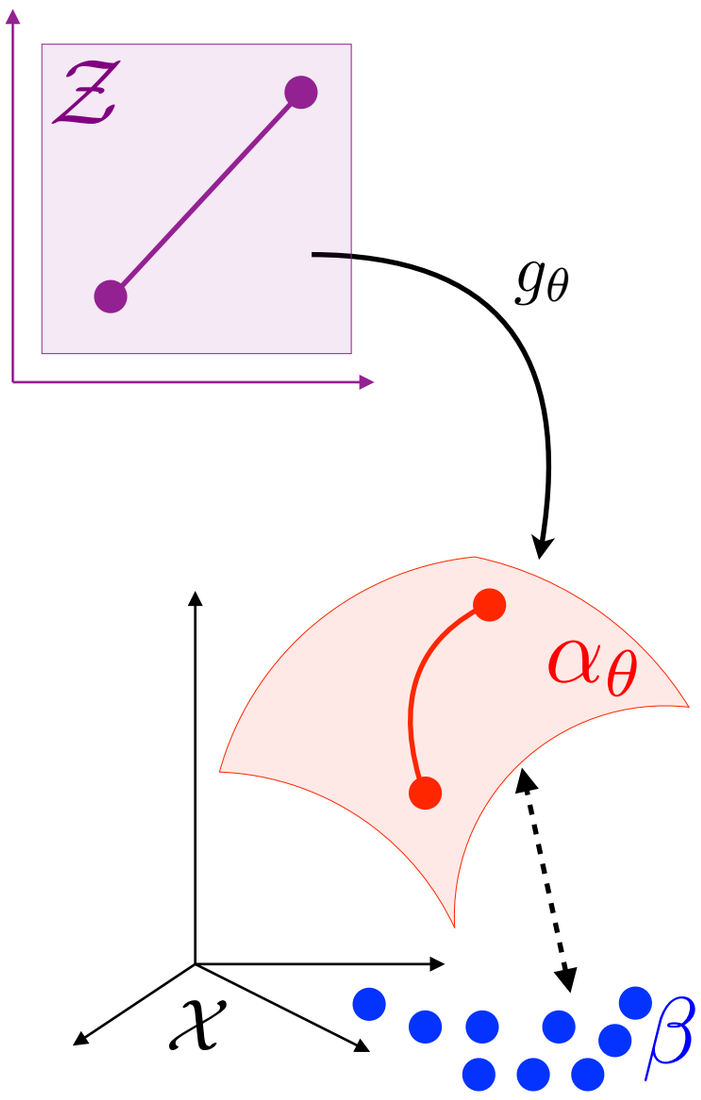


$\rightarrow$ Need to learn the metric $d(x, y) = \|h_\xi(x) - h_\xi(y)\|$ (GANs)

$\rightarrow$ Influence of ε?

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

Generative Adversarial Networks



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

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

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

Open Problems

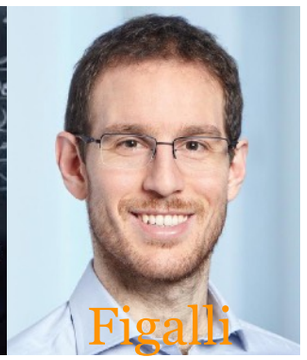

Monge

Kantorovic


Dantzig


Brenier


Villani

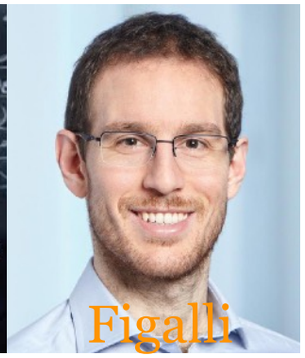

Figalli

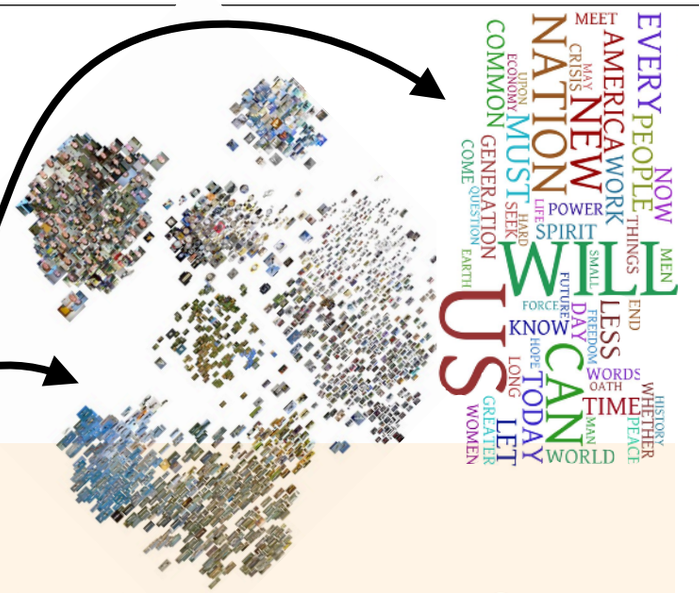
Toward high-dimensional OT:

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

$\rightarrow$ Performance quality measures for unsupervised learning?

Open Problems





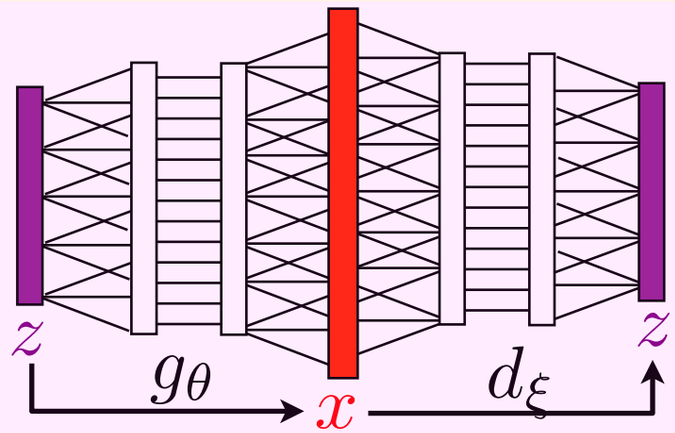
Toward high-dimensional OT:

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

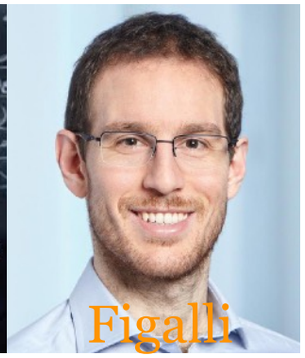
$\rightarrow$ Performance quality measures for unsupervised learning?

Metric learning for OT:

$\rightarrow$ Adversarial training to leverage multi scale priors?



Open Problems



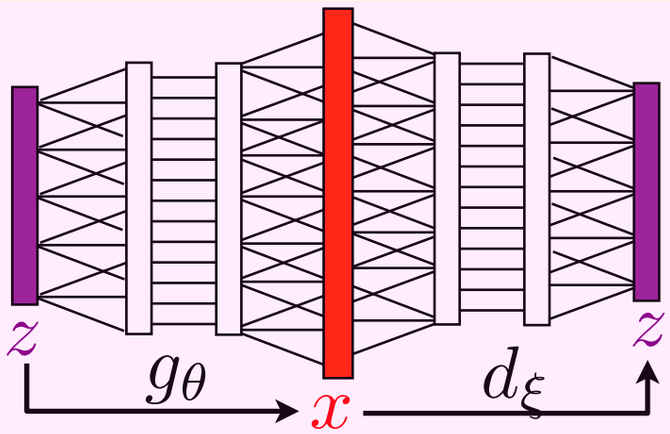
Toward high-dimensional OT:

→ 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?