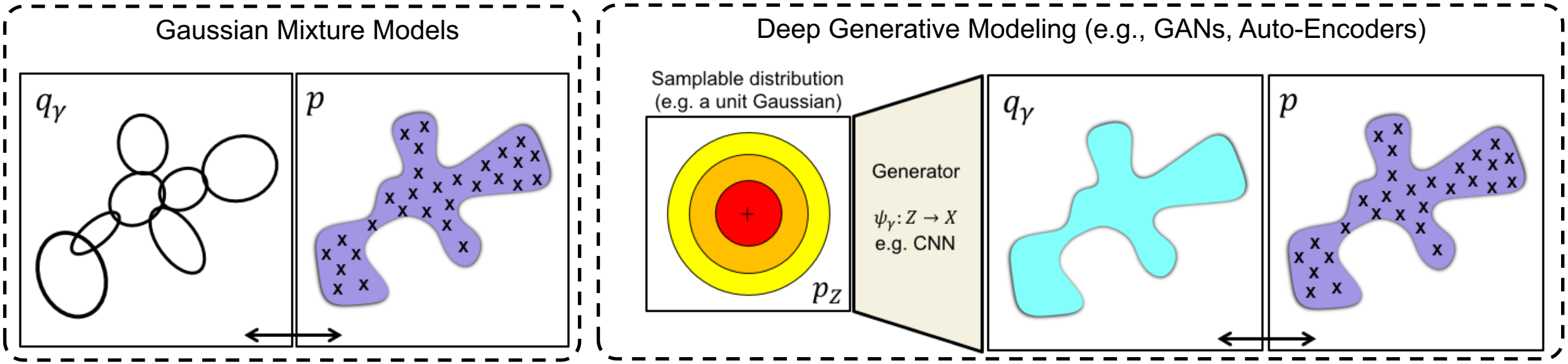
Applications of Transport-Based Methods in Machine Learning

Soheil Kolouri, Ph.D.
skolouri@hrl.com

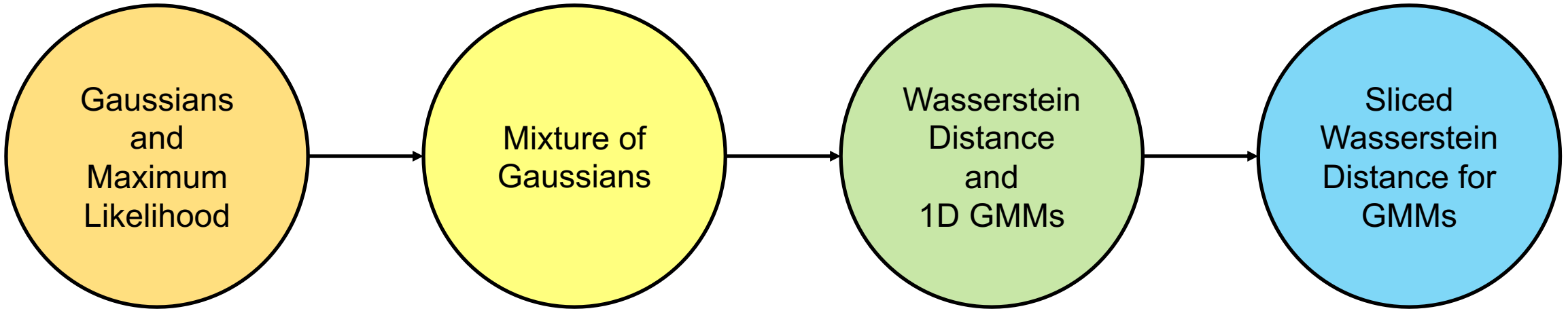
HRL Laboratories, LLC
3011 Malibu Canyon Road
Malibu, CA 90265

Applications of Transport-Based Methods in Machine Learning

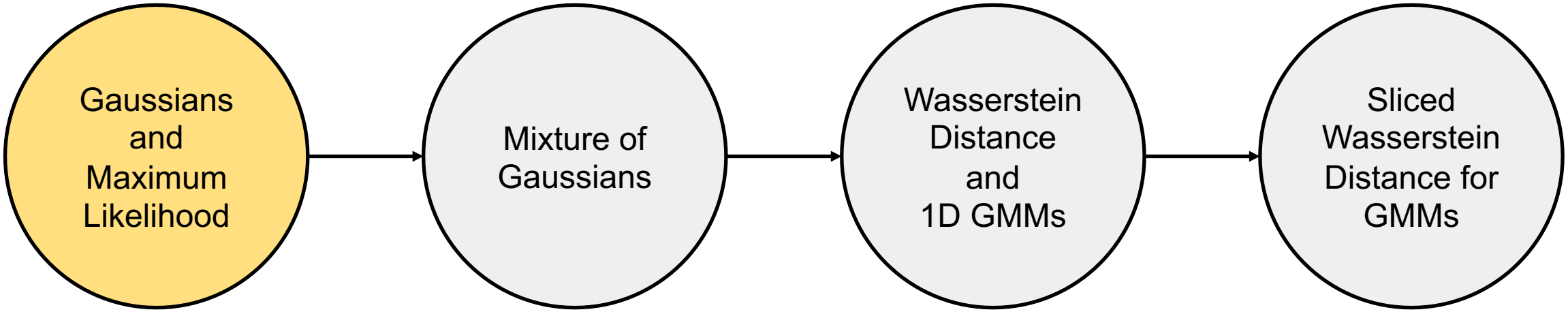


- What are Gaussian Mixture Models (GMMs)?
  - Why should one care?
- What is the EM algorithm?
  - What are the challenges?
- How can we use transport-based distances to estimate GMMs?

Road Map of the Class



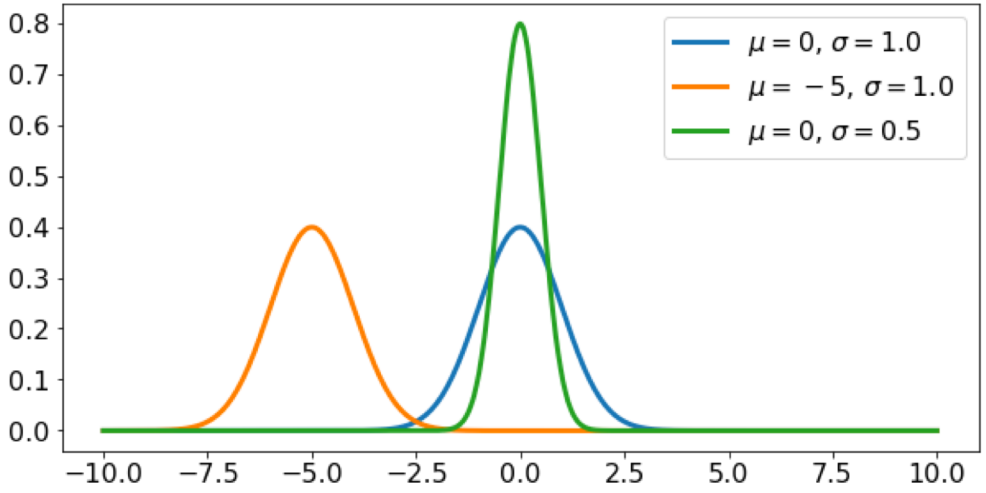
Gaussians and Maximum Likelihood



Gaussian Distribution and Maximum Likelihood

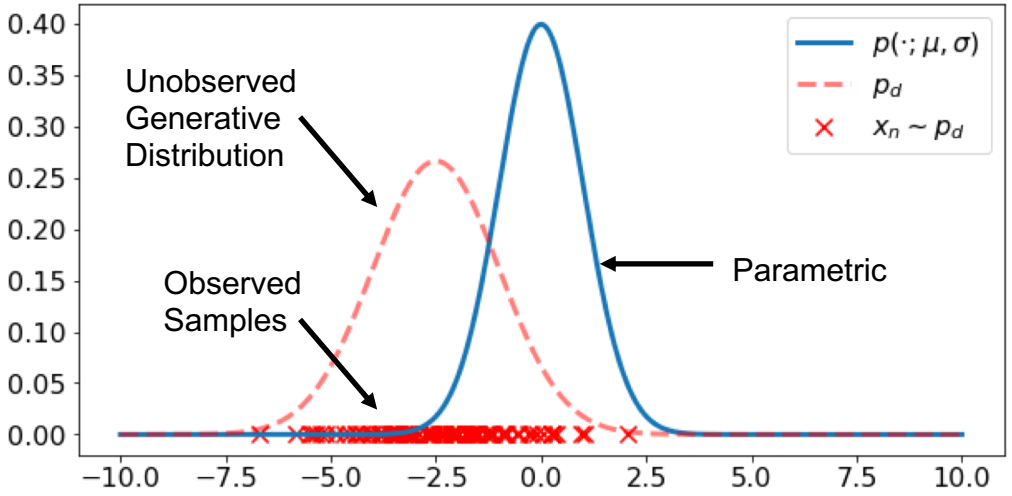
Gaussian Distribution:

$$p(x;\mu,\sigma) = N(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



Assume we have N i.i.d samples from a Gaussian distribution, $\{x_n \sim p_d\}_{n=1}^{N}$.

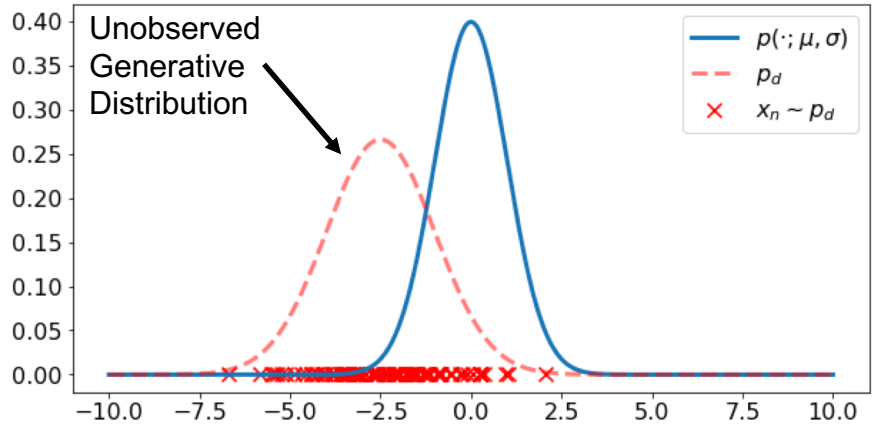
- **How do we estimate the mean and standard deviation from the observed samples?**



Gaussian Distribution and Maximum Likelihood

1- Maximum Likelihood:

$$\max_{\mu,\sigma} p(x_1, \dots, x_N; \mu, \sigma) = \prod_{n=1}^{N} p(x_n; \mu, \sigma)$$



2 - Maximum Log-Likelihood:

$$\max_{\mu,\sigma} \log\left(\prod_{n=1}^{N} p(x_n; \mu, \sigma)\right) = \sum_{n=1}^{N} \log(p(x_n; \mu, \sigma))$$

$$p(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \Rightarrow$$

$$\log\big(p(x; \mu, \sigma)\big) = -\frac{\log(2\pi\sigma^2)}{2} - \frac{(x-\mu)^2}{2\sigma^2}$$

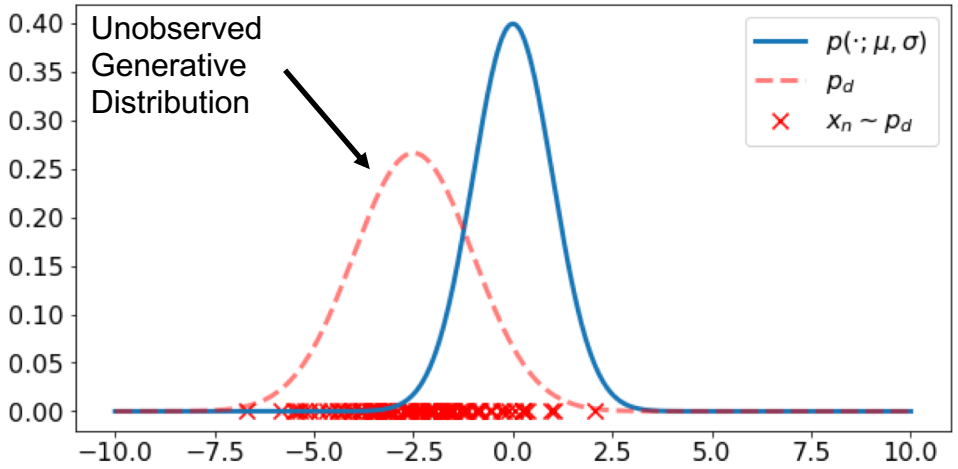
3- Minimizing Negative Log-Likelihood:

$$\min_{\mu,\sigma} \underbrace{\sum_{n=1}^{N} \frac{\log(2\pi\sigma^2)}{2} + \frac{(x_n-\mu)^2}{2\sigma^2}}_{\mathrm{L}} \longrightarrow$$

$$\begin{cases} \dfrac{\partial L}{\partial \mu} = 0 \Rightarrow \mu_* = \dfrac{1}{N}\displaystyle\sum_{n=1}^{N} x_n \\ \dfrac{\partial L}{\partial \sigma} = 0 \Rightarrow \sigma_*^2 = \dfrac{1}{N}\displaystyle\sum_{n=1}^{N} (x_n - \mu^*)^2 \end{cases}$$

Gaussian Distribution and Maximum Likelihood

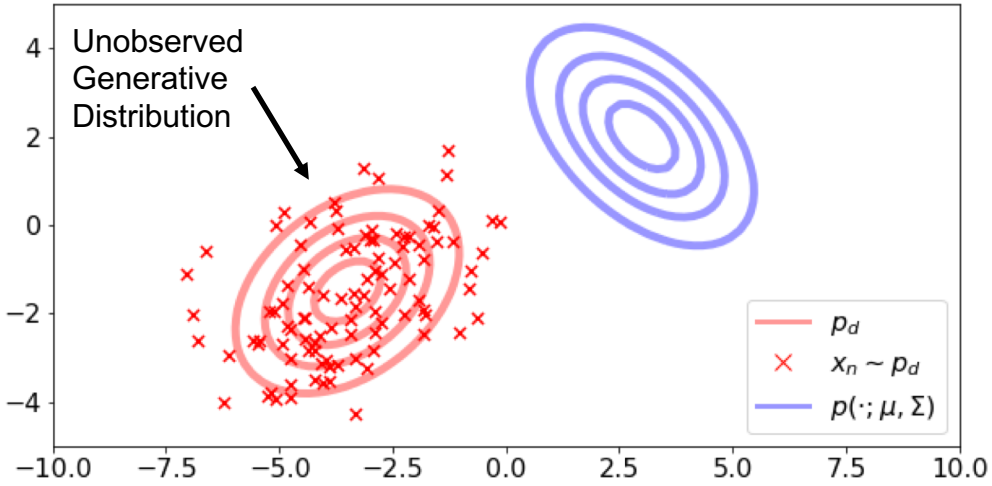
$$p(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



$$\frac{\partial L}{\partial \mu} = 0 \Rightarrow \mu_* = \frac{1}{N}\sum_{n=1}^{N} x_n$$

$$\frac{\partial L}{\partial \sigma} = 0 \Rightarrow \sigma_*^2 = \frac{1}{N}\sum_{n=1}^{N} (x_n - \mu^*)^2$$

$$p(x;\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} e^{-\frac{(x-\mu)^T\Sigma^{-1}(x-\mu)}{2}}$$



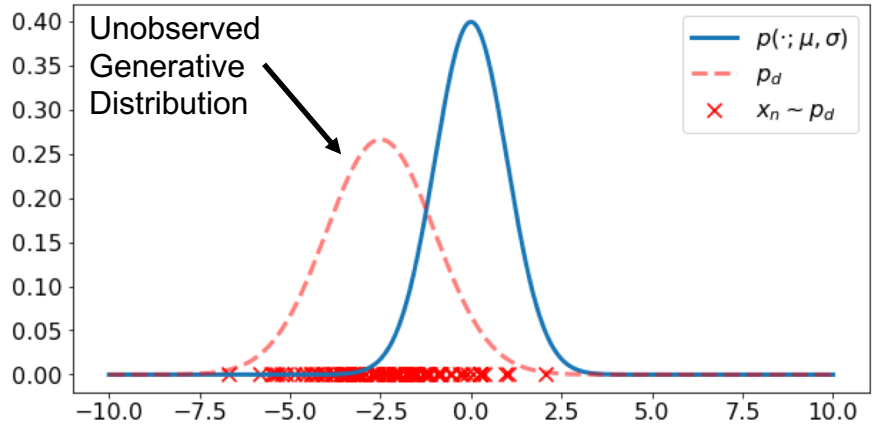
$$\frac{\partial L}{\partial \mu} = 0 \Rightarrow \mu_* = \frac{1}{N}\sum_{n=1}^{N} x_n$$

$$\frac{\partial L}{\partial \Sigma} = 0 \Rightarrow \Sigma_* = \frac{1}{N}\sum_{n=1}^{N} (x_n - \mu^*)(x_n - \mu^*)^T$$

An Alternative View of Maximum Log-Likelihood

Monte-Carlo Approximation

$$\int_{\mathbb{R}} p_d(x) \log\big(p(x;\mu,\sigma)\big)\, dx \approx \sum_{n=1}^{N} \log(p(x_n;\mu,\sigma))$$



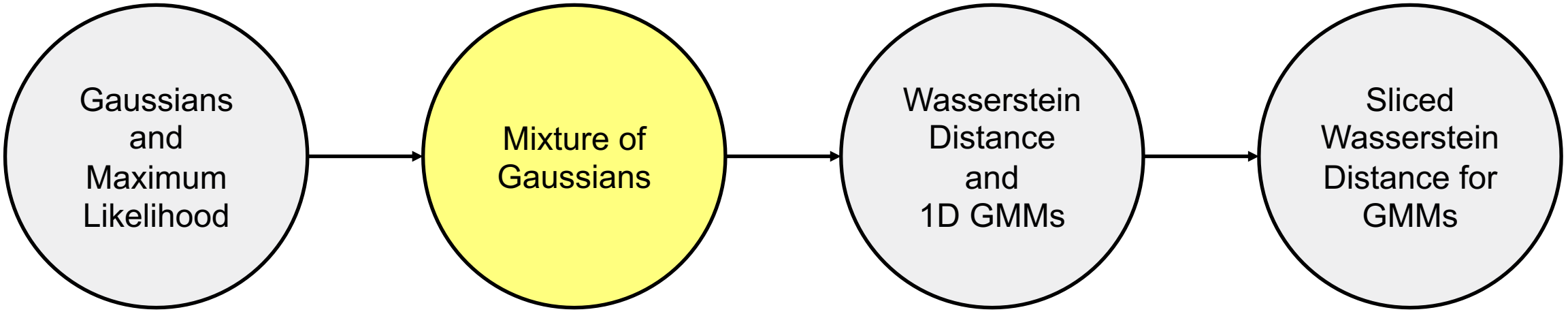
$$\max_{\mu,\sigma} \int_{\mathbb{R}} p_d(x) \log\big(p(x;\mu,\sigma)\big)\, dx = \max_{\mu,\sigma} \int_{\mathbb{R}} p_d(x) \log\big(p(x;\mu,\sigma)\big)\, dx - \int_{\mathbb{R}} p_d(x) \log\big(p_d(x)\big)\, dx$$

$$= \max_{\mu,\sigma} \int_{\mathbb{R}} p_d(x) \log\left(\frac{p(x;\mu,\sigma)}{p_d(x)}\right) dx = \min_{\mu,\sigma} \int_{\mathbb{R}} p_d(x) \log\left(\frac{p_d(x)}{p(x;\mu,\sigma)}\right) dx$$

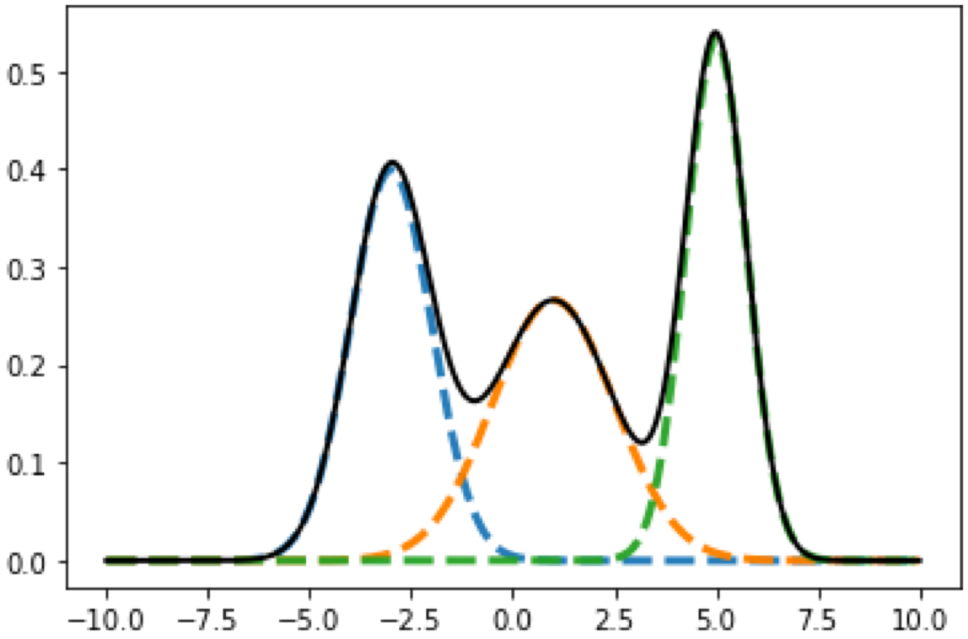
$$= \min_{\mu,\sigma} D_{KL}(p_d \,||\, p(\cdot;\mu,\sigma))$$

Maximizing log-likelihood is equivalent to minimizing the Kullback-Leibler (KL) divergence between the generative distribution and the parametric distribution.

Mixture of Gaussians



Mixture of Gaussians



$$p\left(x; [(\alpha_k, \mu_k, \sigma_k)]_{k=1}^{K}\right) = \sum_{k=1}^{K} \alpha_k N(x; \mu_k, \sigma_k) = \sum_{k=1}^{K} \frac{\alpha_k}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}$$

Where $\alpha_k \geq 0$, $\sum_{k=1}^{K} \alpha_k = 1$, and $\sigma_k \geq 0$.

Assume we have N i.i.d samples from a mixture of Gaussians distribution, $\{x_n \sim p_d\}_{n=1}^{N}$.

- **How do we estimate the parameters of the mixture from the observed samples?**

Log-Likelihood:

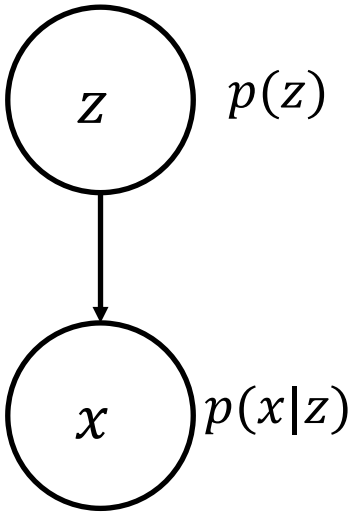
$$\log\left(\prod_{n=1}^{N}\left(\sum_{k=1}^{K} \alpha_k N(x_n; \mu_k, \sigma_k)\right)\right)$$

$$= \sum_{n=1}^{N} \log\left(\sum_{k=1}^{K} \alpha_k N(x_n; \mu_k, \sigma_k)\right)$$

Expectation Maximization: Introducing the Latent Variable

If, for each sample, we knew which Gaussian it is sampled from the problem would have been solved!
Meaning that we could have solved K maximum log-likelihoods to estimate parameters of each Gaussian independent of the others!

$z = [z_1, \dots, z_K], z_k \in \{0,1\}$



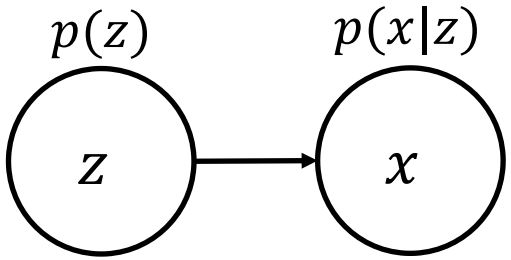
$$p(z) = \prod_{k=1}^{K} \alpha_k^{z_k}, \qquad p(z_k = 1) = \alpha_k$$

$$p(x|z_k = 1) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}$$

$$p(x) = \sum_{k=1}^{K} p(x|z_k = 1)p(z_k = 1) = \sum_{k=1}^{K} \frac{\alpha_k}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}$$

Expectation Maximization: Derivations

$$\frac{d}{d\theta}\log p(x) = \frac{d}{d\theta}\log\left(\sum_z p(z,x)\right) = \frac{\frac{d}{d\theta}\sum_z p(z,x)}{\sum_{z'} p(z',x)} = \frac{\sum_z \frac{d}{d\theta} p(z,x)}{\sum_{z'} p(z',x)}$$



$$\frac{\sum_z p(z,x)\frac{\frac{d}{d\theta}p(z,x)}{p(z,x)}}{\sum_{z'} p(z',x)} = \sum_z \left(\frac{p(z,x)}{\sum_{z'} p(z',x)}\right)\frac{d}{d\theta}\log\big(p(z,x)\big) = \sum_z p(z|x)\frac{d}{d\theta}\log\big(p(z,x)\big)$$

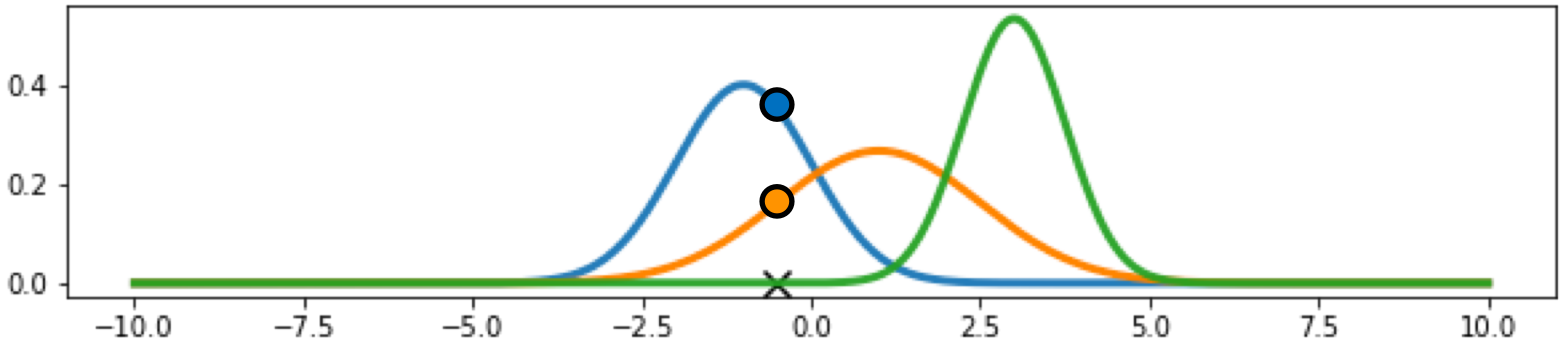
$$\sum_z p(z|x)\frac{d}{d\theta}\log\big(p(x|z)p(z)\big) = \sum_z p(z|x)\frac{d}{d\theta}\log\big(p(x|z)\big) + \sum_z p(z|x)\frac{d}{d\theta}\log\big(p(z)\big)$$

$$p(z|x) = \frac{\mathrm{p(x|z)p(z)}}{\sum_{z'} p(x|z')p(z')} \longrightarrow$$ Is the soft assignment of data x to each Gaussian.

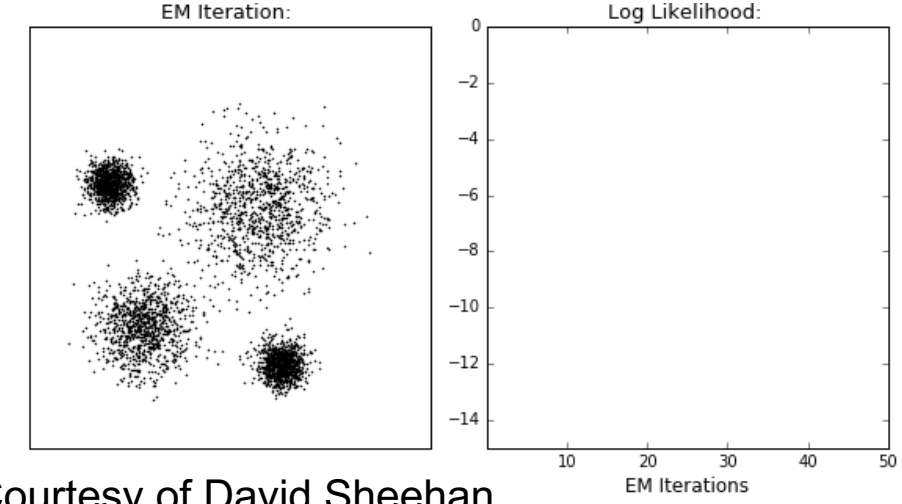
Expectation Maximization: Algorithm

- **E**xpectation Step: for fixed parameters $[(\alpha_k, \mu_k, \sigma_k)]_{k=1}^{K}$ compute $\mathrm{r}_n^k = p(z_k = 1|x_n)$ for each sample

$$\mathrm{r}_n^k = \frac{\alpha_k N(x_n; \mu_k, \sigma_k)}{\sum_i \alpha_i N(x_n; \mu_i, \sigma_i)}$$



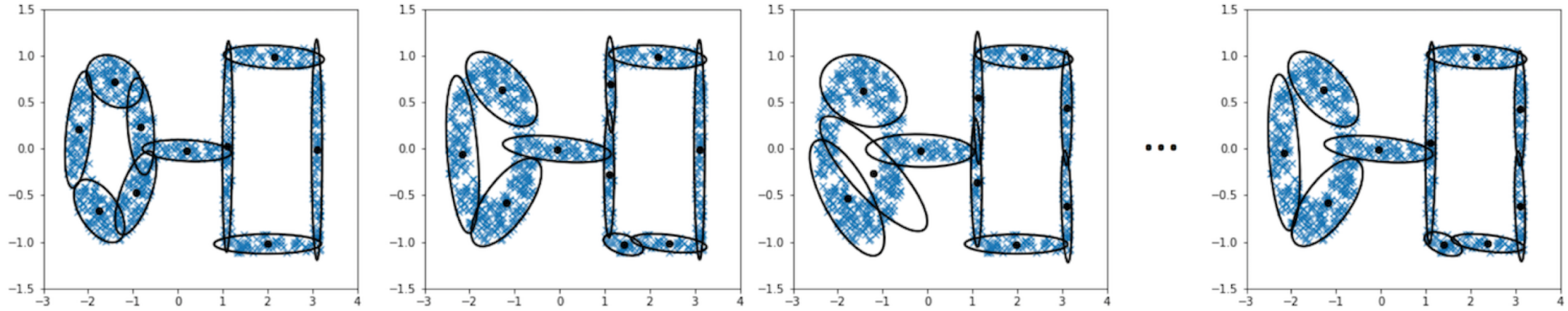
- **M**aximization Step: for fixed r_n^k solve the maximum log-likelihood to obtain optimal parameters:
  - Mixture Coefficients: $N_k = \sum_n r_n^k, \quad \alpha_k = \frac{N_k}{N}$
  - Means: $\mu_k = \frac{1}{N_k} \sum_n r_n^k x_n$
  - Variances: $\sigma_k^2 = \frac{1}{N_k} \sum_n r_n^k (x_n - \mu_k)^2$

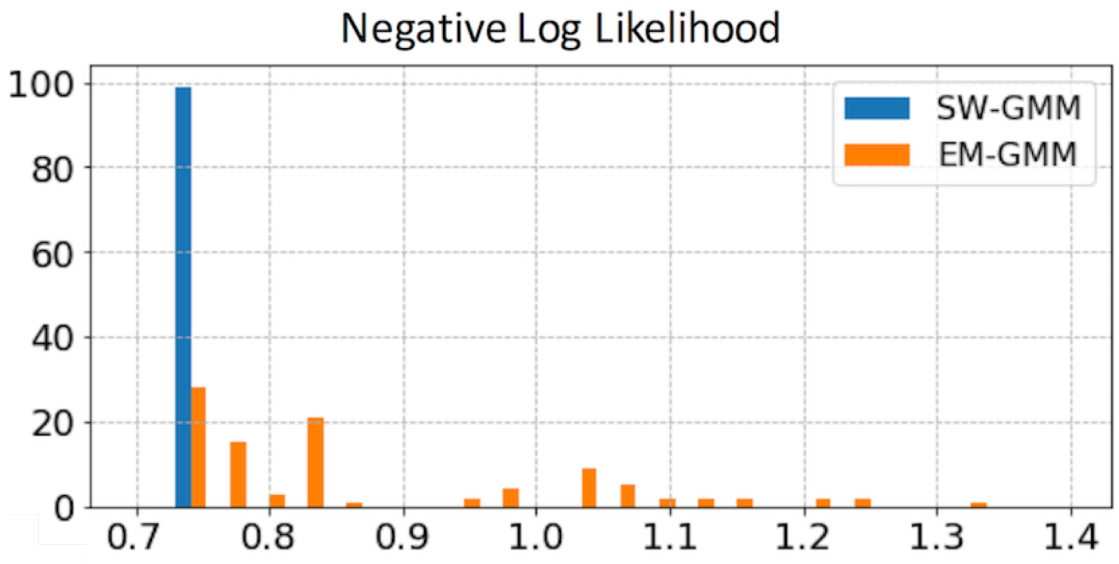


Courtesy of David Sheehan

Expectation Maximization: Sensitivity to Initialization

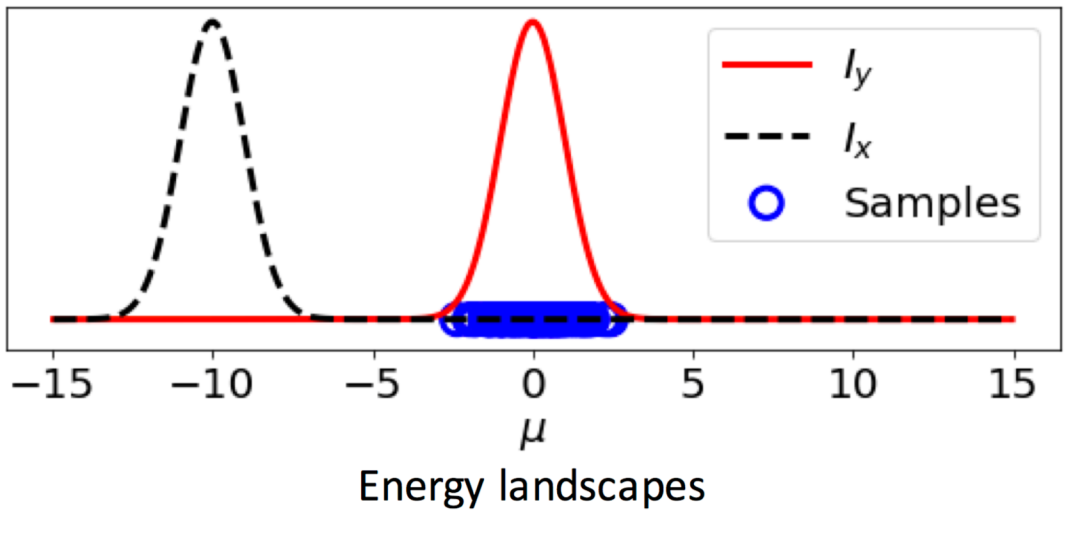
EM-GMM



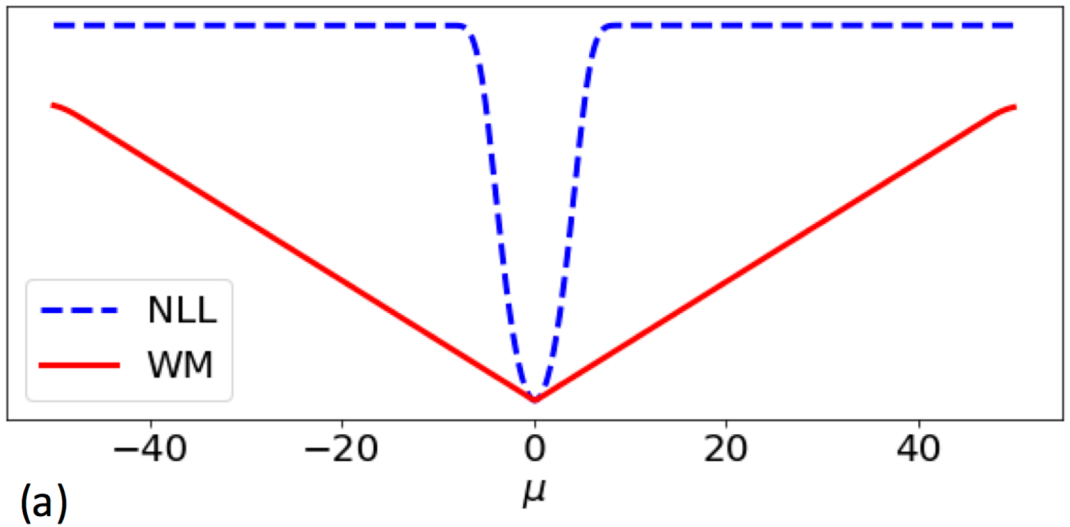


- EM is sensitive to the choice of initial parameters (Non-convex problem!)
- Various methods have been developed for a better initialization of GMMs:
  - Pick means sequentially to provide good coverage of data
  - Initialize with K-means
  - etc.

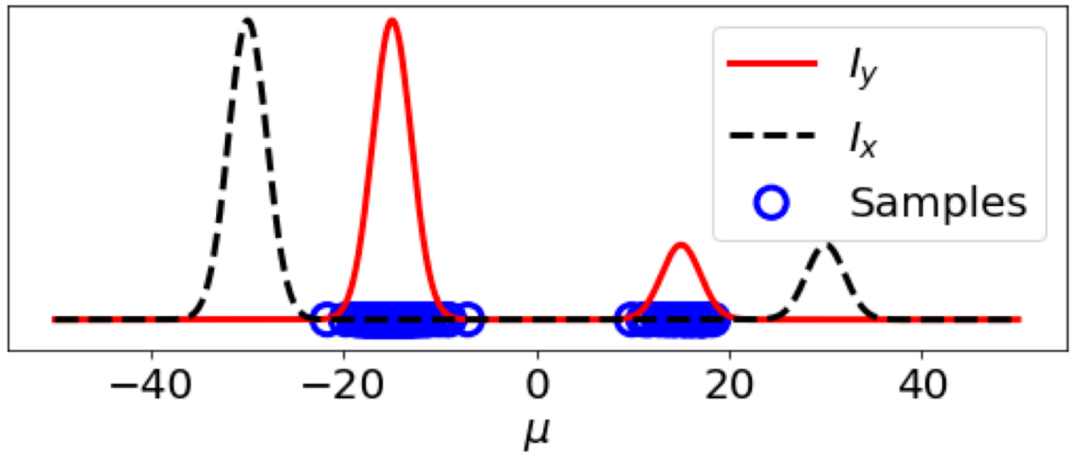
Wasserstein Distance vs. KL Divergence (or NLL)



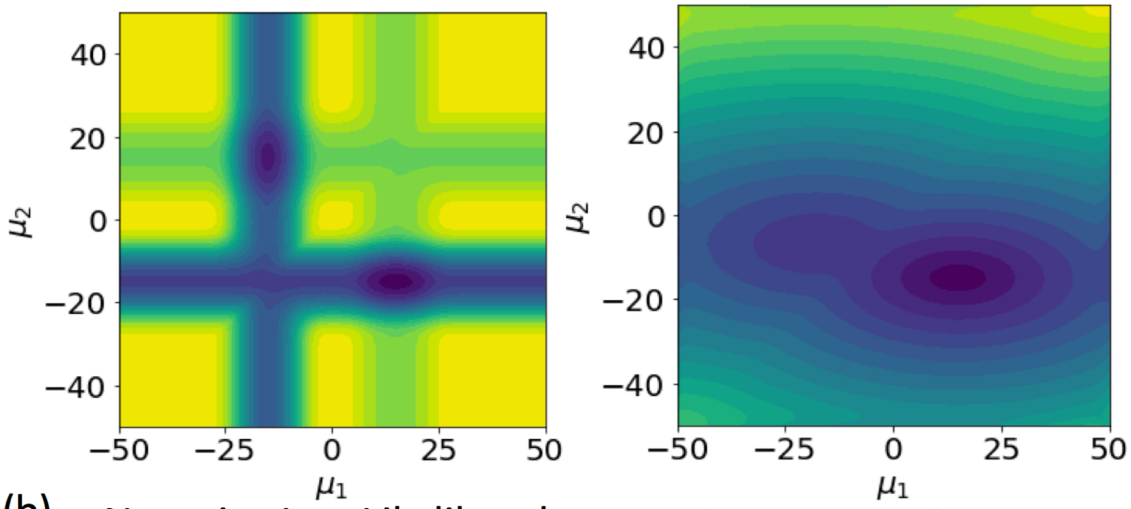
Energy landscapes



(a)



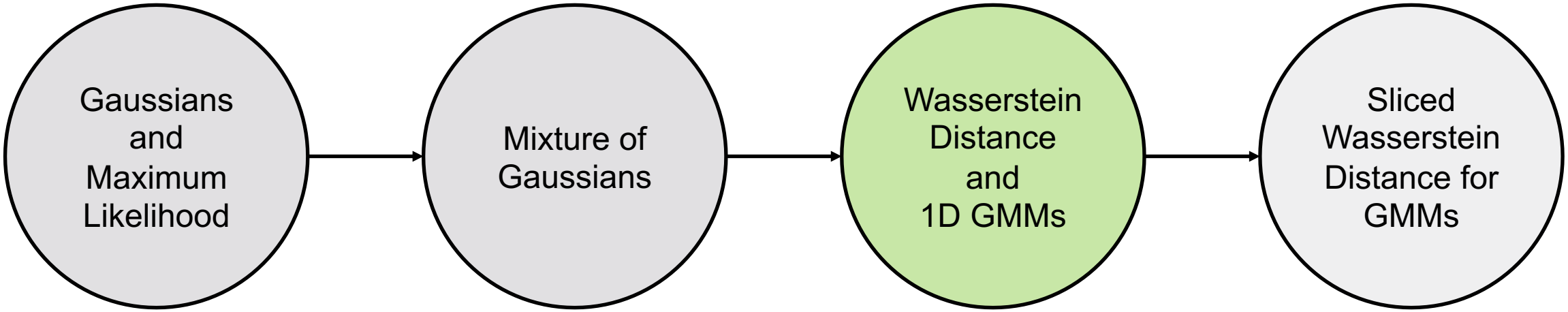
Energy landscapes



(b) Negative Log-Likelihood

2-Wasserstein Means

Wasserstein Distance and 1D GMMs



Wasserstein Distance

Wasserstein: The (p=2)-Wasserstein distance between p_X and q_X is defined as:

$$W_2(q_X, p_X) = \min_f \left(\int_X \|f(x) - x\|_2^2 p_X(x) dx\right)^{\frac{1}{2}},$$
$$\text{s.t. } \det\left(Df(x)\right) q_X\left(f(x)\right) = p_X(x)$$

which is the minimum expected transportation that is needed to morph p_X into q_X.

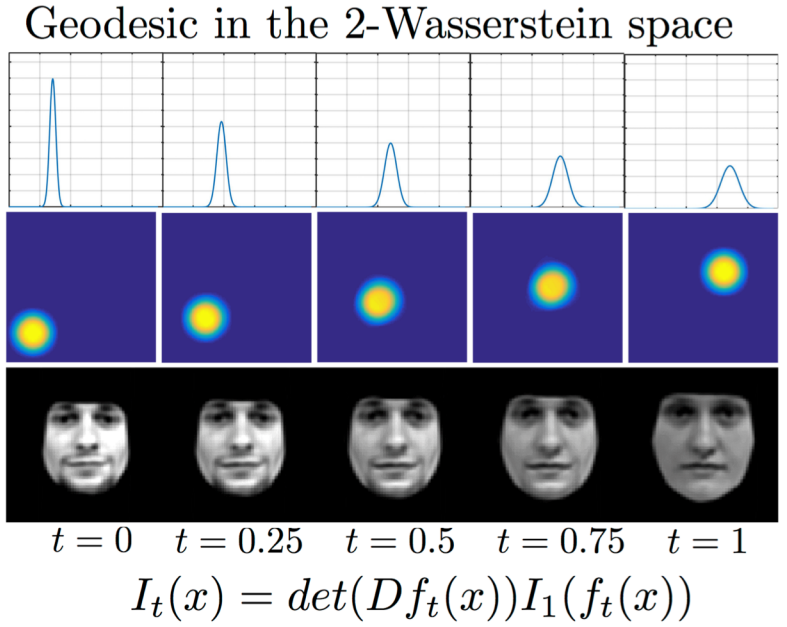
One-dimensional: In one-dimension the Wasserstein distance has a closed form solution:

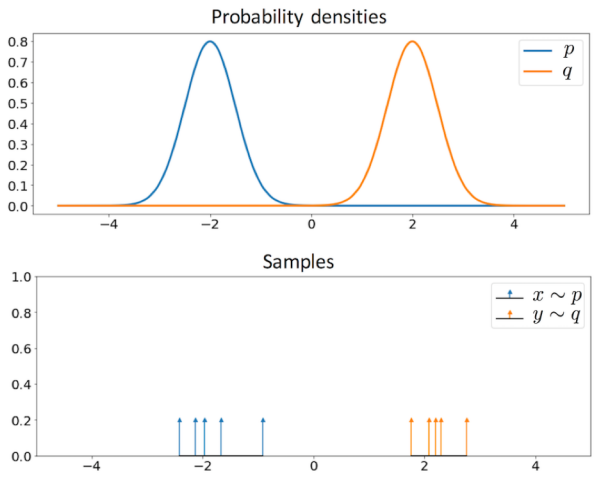
$$W_2(q_X, p_X) = \left(\int_0^1 \left\|P_X^{-1}(\tau) - Q_X^{-1}(\tau)\right\|_2^2 d\tau\right)^{\frac{1}{2}}$$

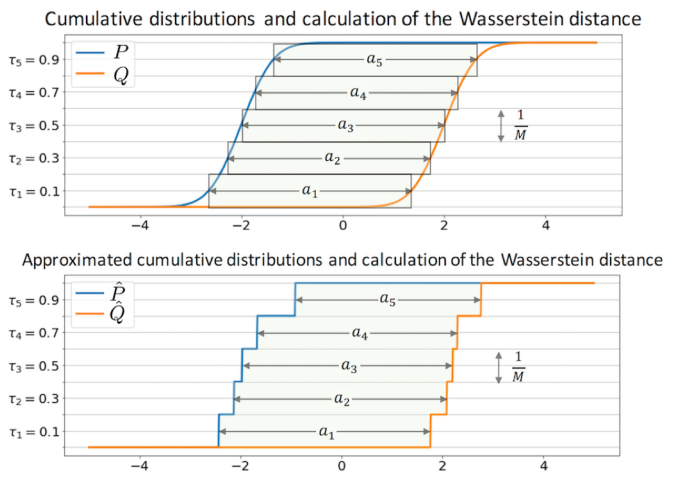
where P_X and Q_X are cumulative distribution functions corresponding to p_X and q_X.

Optimal transport map in one-dimension:

$$f(x) = P_X^{-1}(Q_X(x))$$







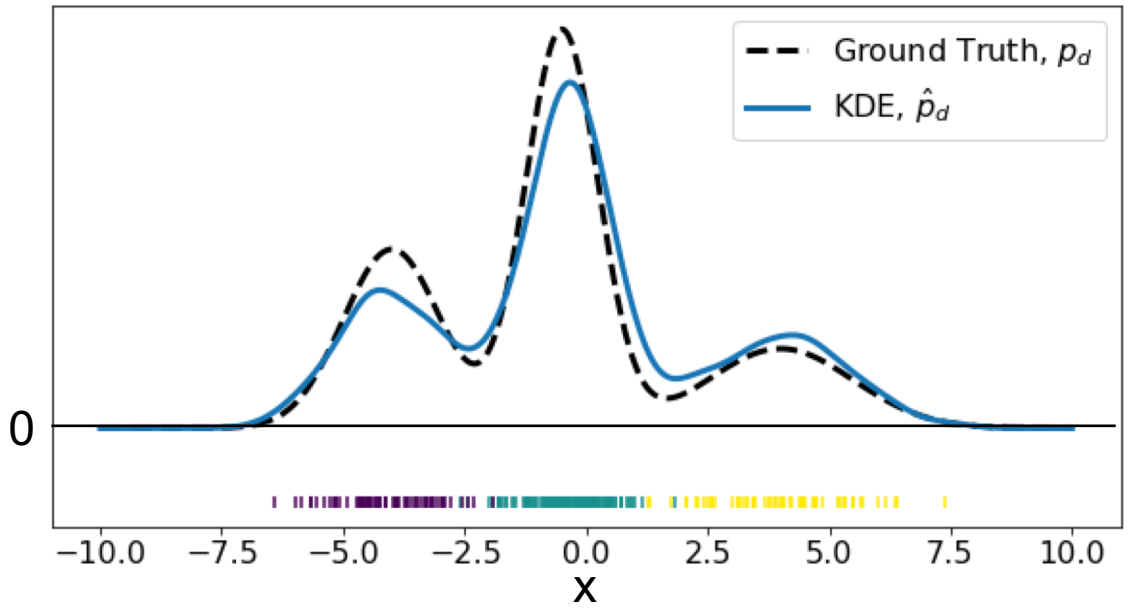
Wasserstein for Estimating GMMs in 1D

$$\hat{p}_d(x) = \frac{1}{N}\sum_{n=1}^{N}\phi(x - x_n), \qquad p(x) = \sum_{k=1}^{K}\alpha_k N(x;\mu_k,\sigma_k)$$

$$\min_{[\alpha_k,\mu_k,\sigma_k]_k} W_2^2(p,\hat{p}_d) = \int_{\mathbb{R}} |f(x) - x|^2 p(x)dx$$

$$= \int_{\mathbb{R}} u(x)^2 p(x)dx$$

$$= \sum_k \alpha_k \int_{\mathbb{R}} u(x)^2\, N(x;\mu_k,\sigma_k)\, dx$$



EM-Like Strategy

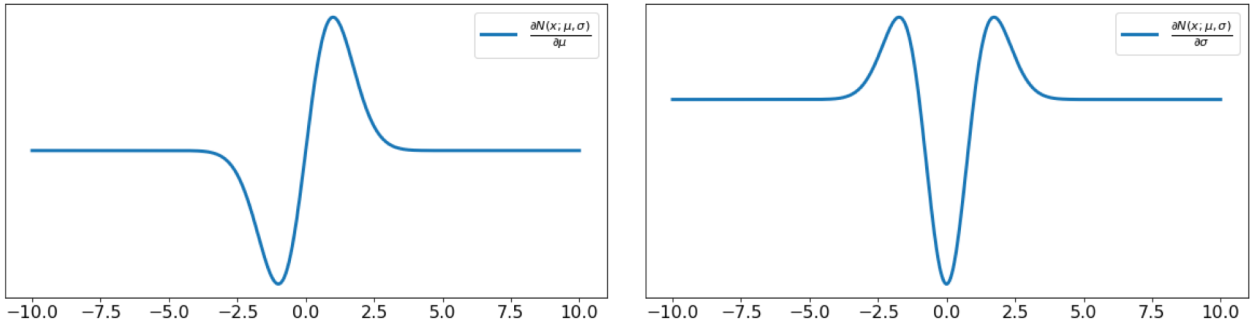
1. For fixed parameters $[(\alpha_k,\mu_k,\sigma_k)]_{k=1}^K$
   - Compute: $u(x)$

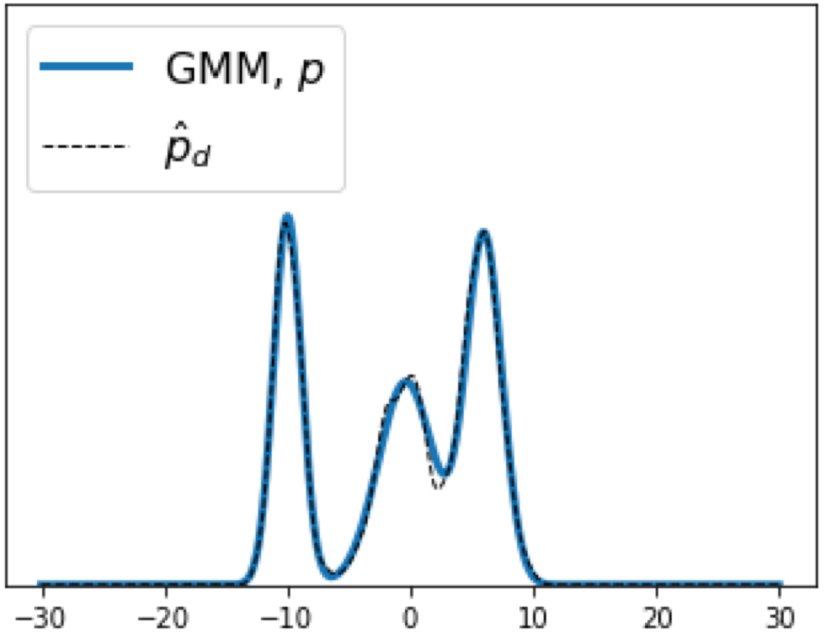
2. For fixed $u(x)$ update parameters:
   - $\mu_k^{t+1} = \mu_k^t - h\left(\alpha_k \int_{\mathbb{R}} u(x)^2 \frac{dN(x;\mu_k,\sigma_k)}{d\mu_k} dx\right)$
   - $\sigma_k^{t+1} = \sigma_k^t - h\left(\alpha_k \int_{\mathbb{R}} u(x)^2 \frac{dN(x;\mu_k,\sigma_k)}{d\sigma_k} dx\right)$
   - $\alpha_k^{t+1} = \alpha_k^t - h\left(\int_{\mathbb{R}} u(x)^2\, N(x;\mu_k,\sigma_k)\, dx\right)$

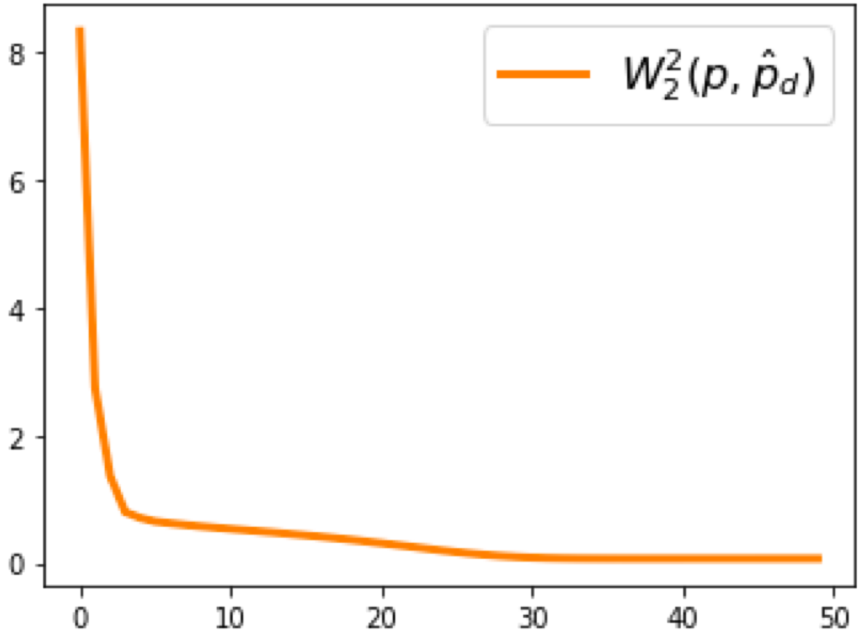
Wasserstein for Estimating GMMs in 1D: Derivative Terms

$$\frac{dN(x;\mu_k,\sigma_k)}{d\mu_k} = \frac{x-\mu_k}{\sigma_k^2} N(x;\mu_k,\sigma_k)$$

$$\frac{dN(x;\mu_k,\sigma_k)}{d\sigma_k} = \frac{1}{\sigma_k}\left(\frac{(x-\mu_k)^2}{\sigma_k^2} - 1\right) N(x;\mu_k,\sigma_k)$$

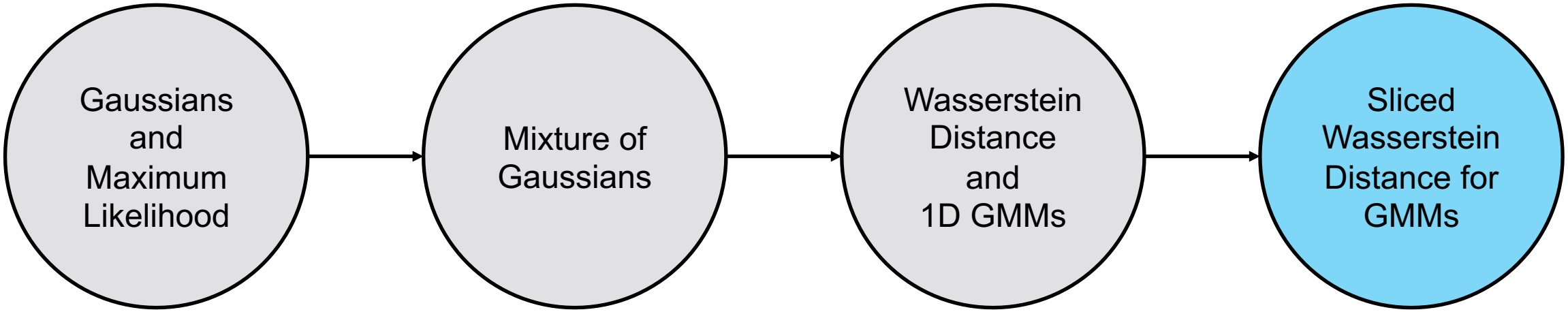




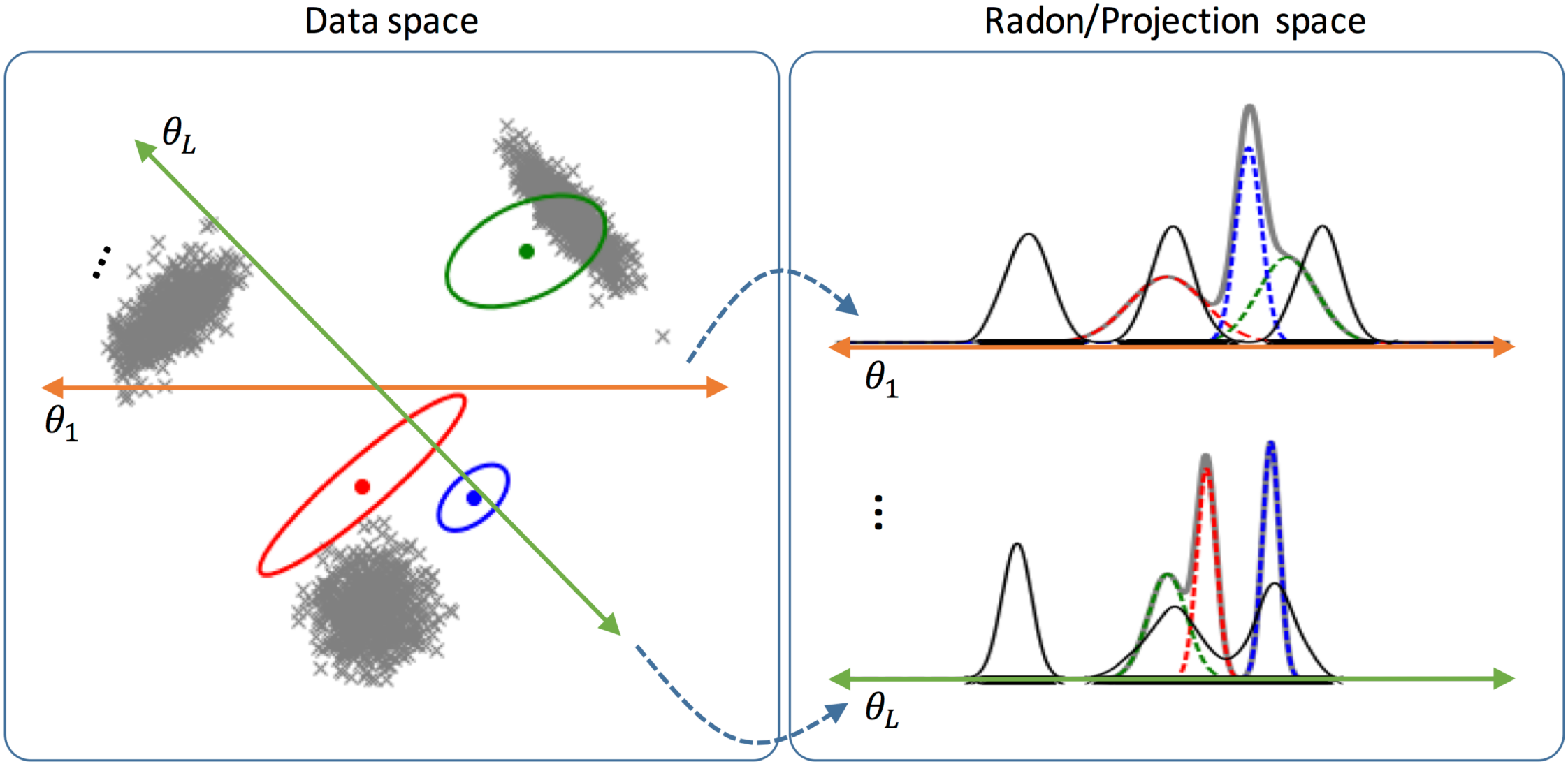


[Link to Code]

Sliced Wasserstein Distance for GMMs



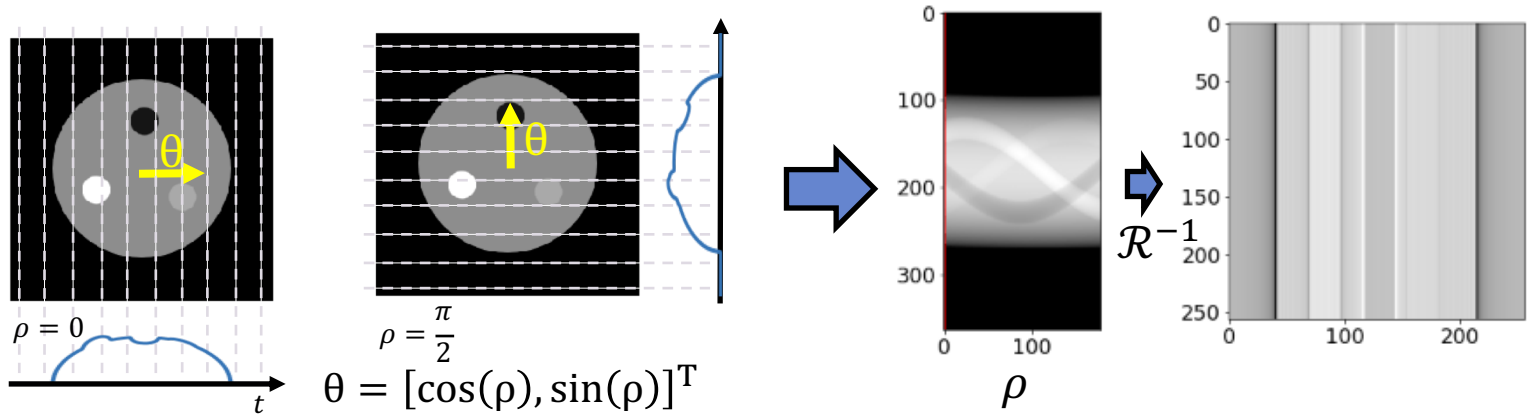
- How About Higher Dimensional Distributions? - **Slice Them!**



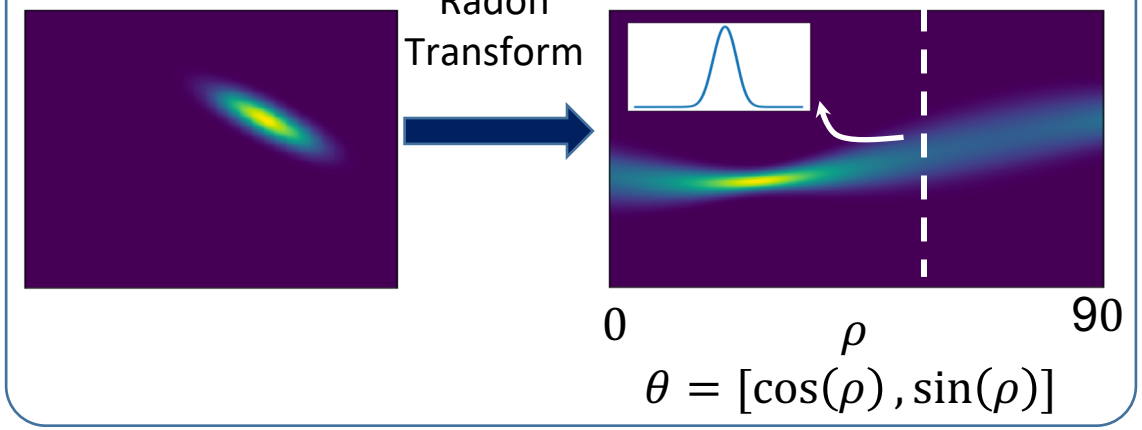
Radon Transform and Slicing High-Dimensional Distributions

A slice of distribution p, with respect to the unit vector θ, is its pushforward with function $g(x) = x \cdot \theta$:

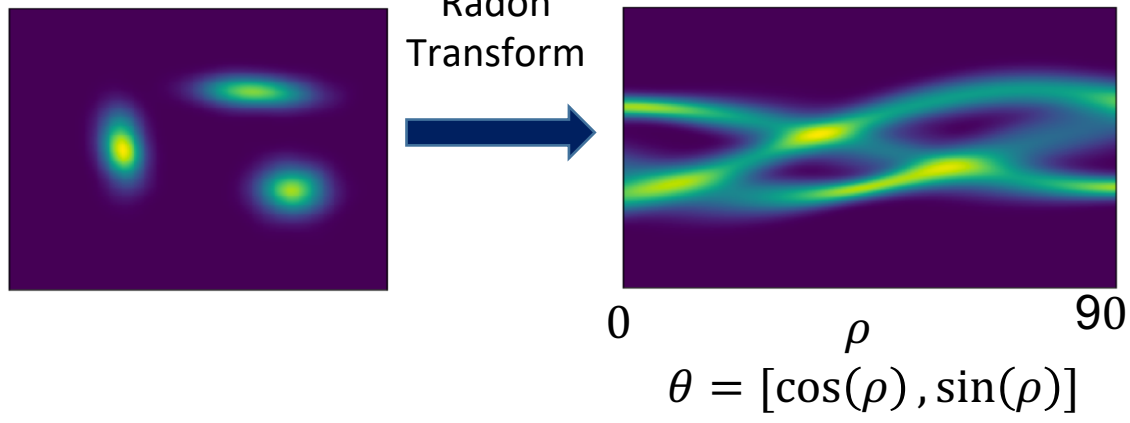
$$\mathcal{R}p(t;\theta) = \int_{\mathbb{R}^d} p(x)\delta(t - x \cdot \theta)dx$$



$p(\cdot) = \mathcal{N}_d(\mu, \Sigma)$ $\quad$ $\mathcal{R}p(\cdot,\theta) = \mathcal{N}_1(\theta^T\mu, \theta^T\Sigma\theta)$



$p(\cdot) = \sum_i p_k \mathcal{N}_d(\mu_k, \Sigma_k)$ $\quad$ $\mathcal{R}p(\cdot,\theta) = \sum_k \alpha_k \mathcal{N}_1(\theta^T\mu_k, \theta^T\Sigma_k\theta)$



Slicing an Empirical Distribution

$$\hat{p}_d(x) = \frac{1}{N}\sum_{n=1}^{N} \phi(x - x_n)$$ Empirical Distribution

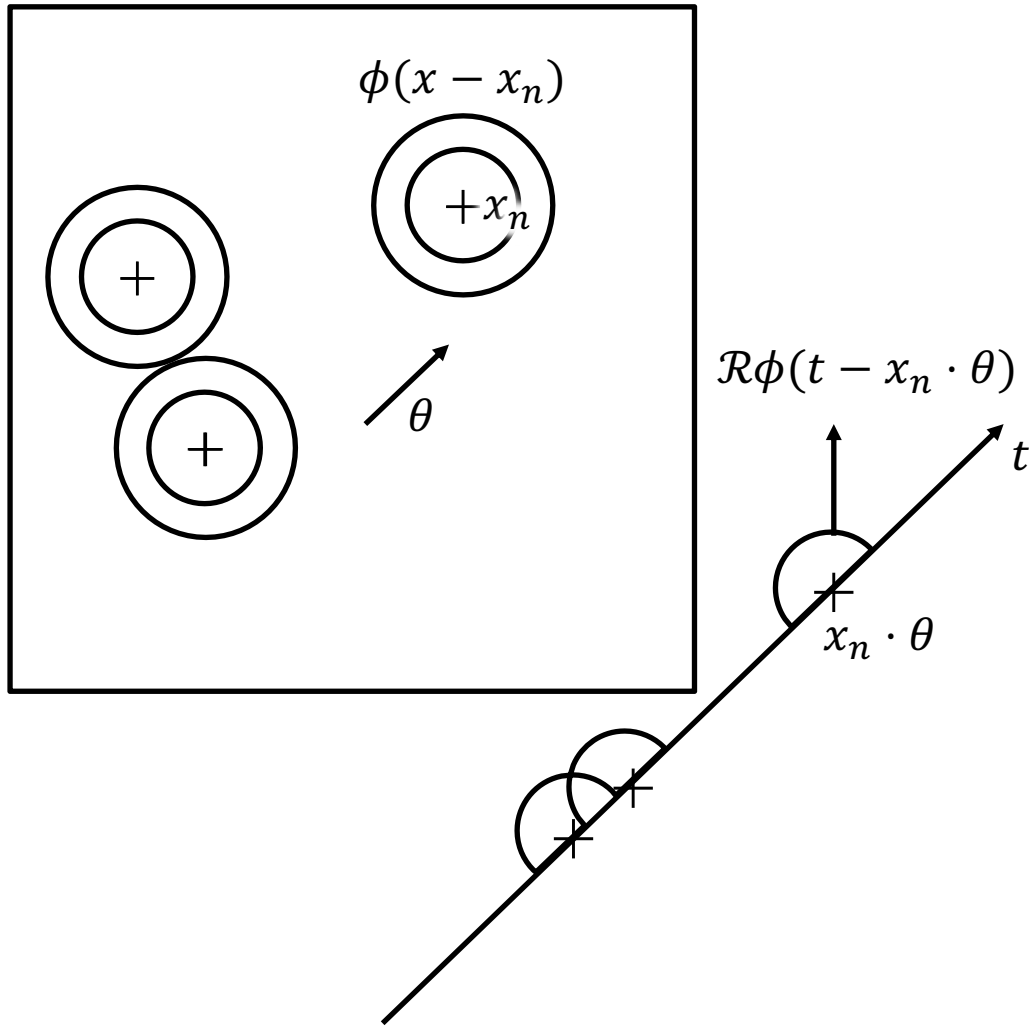
$$\mathcal{R}\hat{p}_d(t;\theta) = \int_{\mathbb{R}^d} \hat{p}(x)\delta(t - x\cdot\theta)dx$$

$$= \int_{\mathbb{R}^d} \frac{1}{N}\sum_{n=1}^{N} \phi(x - x_n)\,\delta(t - x\cdot\theta)dx$$

$$= \frac{1}{N}\sum_{n=1}^{N} \int_{\mathbb{R}^d} \phi(x - x_n)\delta(t - x\cdot\theta)dx$$

$$= \frac{1}{N}\sum_{n=1}^{N} \int_{\mathbb{R}^d} \phi(z)\delta(t - x_n\cdot\theta - z\cdot\theta)dz$$

$$= \frac{1}{N}\sum_{n=1}^{N} \mathcal{R}\phi(t - x_n\cdot\theta;\theta)$$



Sliced Wasserstein Distance for Learning GMMs

Sliced Wasserstein distance between two d-dimensional distributions:

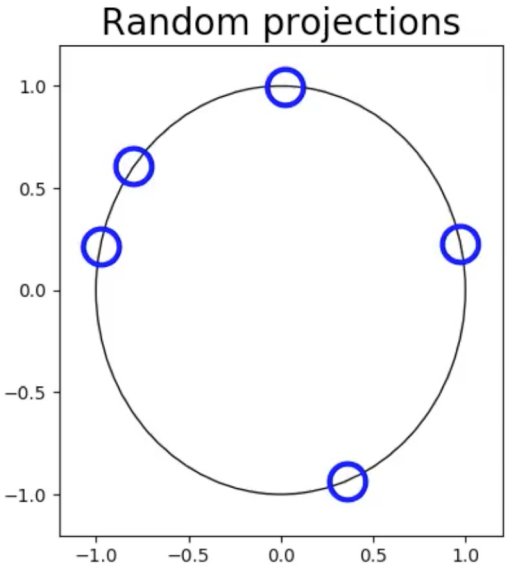
$$SW_2^2(p, \hat{p}_d) = \int_{\mathbb{S}^{d-1}} W_2^2(\mathcal{R}p(\cdot;\theta), \mathcal{R}\hat{p}_d(\cdot;\theta))d\theta$$

$$\min_{[\alpha_k,\mu_k,\sigma_k]_k} SW_2^2(p, \hat{p}_d) = \int_{\mathbb{S}^{d-1}} \int_{\mathbb{R}} |f_\theta(t) - t|^2 \mathcal{R}p(t;\theta)dt\, d\theta$$

where $f_\theta(t) = \mathcal{R}\hat{P}_d^{-1}(\mathcal{R}P(t;\theta);\theta)$

$$= \sum_k \alpha_k \int_{\mathbb{S}^{d-1}} \int_{\mathbb{R}} u_\theta(t)^2 \mathcal{N}_1(\theta^T\mu_k, \theta^T\Sigma_k\theta)dt\, d\theta$$

$$\approx \frac{1}{L}\sum_{l=1}^{L}\sum_k \alpha_k \int_{\mathbb{R}} u_\theta(t)^2 \mathcal{N}_1(\theta_l^T\mu_k, \theta_l^T\Sigma_k\theta_l)dt$$



$L = 5$

Sliced Wasserstein Distance for Learning GMMs

EM-Like Strategy

1. For fixed parameters $[(\alpha_k, \mu_k, \sigma_k)]_{k=1}^{K}$

- Generate L random samples from unit sphere: $[\theta_l]_{l=1}^{L}$
- compute: $\left[u_{\theta_l}(t)\right]_{l=1}^{L}$

2. For fixed $\left[u_{\theta_l}(t)\right]_{l=1}^{L}$ update parameters:

- $\mu_k^{t+1} = \mu_k^t - h\left(\frac{\alpha_k}{L}\sum_{l=1}^{L}\int_{\mathbb{R}} u_{\theta_l}(t)^2 \frac{d\mathcal{N}_1(\theta_l^T\mu_k, \theta_l^T\Sigma_k\theta_l)}{d\mu_k} dt\right)$
- $\Sigma_k^{t+1} = \Sigma_k^t - h\left(\frac{\alpha_k}{L}\sum_{l=1}^{L}\int_{\mathbb{R}} u_{\theta_l}(t)^2 \frac{d\mathcal{N}_1(\theta_l^T\mu_k, \theta_l^T\Sigma_k\theta_l)}{d\Sigma_k} dt\right)$
- $\alpha_k^{t+1} = \alpha_k^t - h\left(\frac{1}{L}\sum_{l=1}^{L}\int_{\mathbb{R}} u_{\theta_l}(t)^2 \mathcal{N}_1(\theta_l^T\mu_k, \theta_l^T\Sigma_k\theta_l) dt\right)$

$$\frac{d\mathcal{N}_1(\theta_l^T\mu_k, \theta_l^T\Sigma_k\theta_l)}{d\mu_k} = \left(\frac{t-\theta_l^T\mu_k}{\theta_l^T\Sigma_k\theta_l}\right)\mathcal{N}_1(\theta_l^T\mu_k, \theta_l^T\Sigma_k\theta_l)\theta_l$$

$$\frac{d\mathcal{N}_1(\theta_l^T\mu_k, \theta_l^T\Sigma_k\theta_l)}{d\Sigma_k} = \frac{1}{\sqrt{\theta_l^T\Sigma_k\theta_l}}\left(\frac{\left(t-\theta_l^T\mu_k\right)^2}{\theta_l^T\Sigma_k\theta_l} - 1\right)\mathcal{N}_1(\theta_l^T\mu_k, \theta_l^T\Sigma_k\theta_l)(\theta_l\theta_l^T)$$

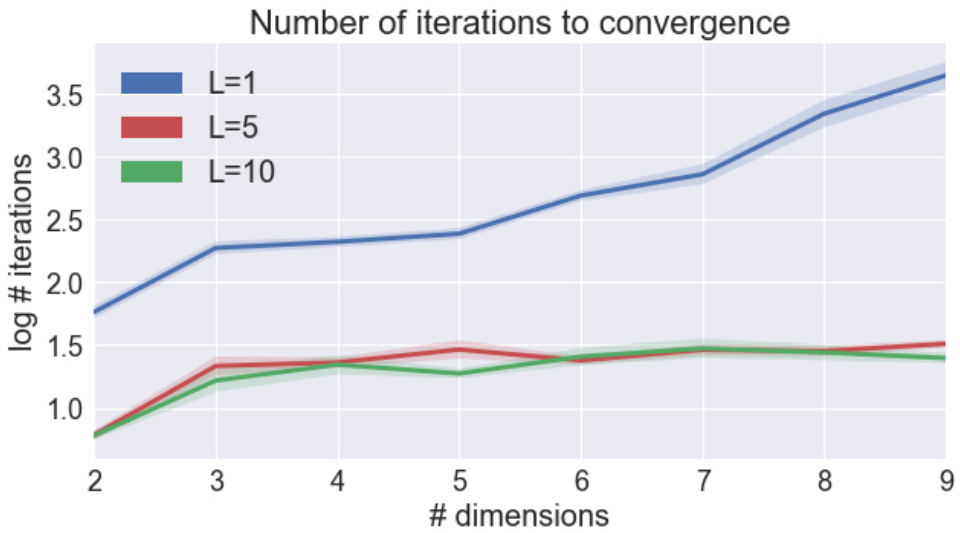
How Many Slices Do We Need?

Algorithm:

Input: Data $X = \{x_1, \dots, x_N \in \mathbb{R}^d\}$, number of components, number of random slices per iteration, L.

Initialize GMM parameters.

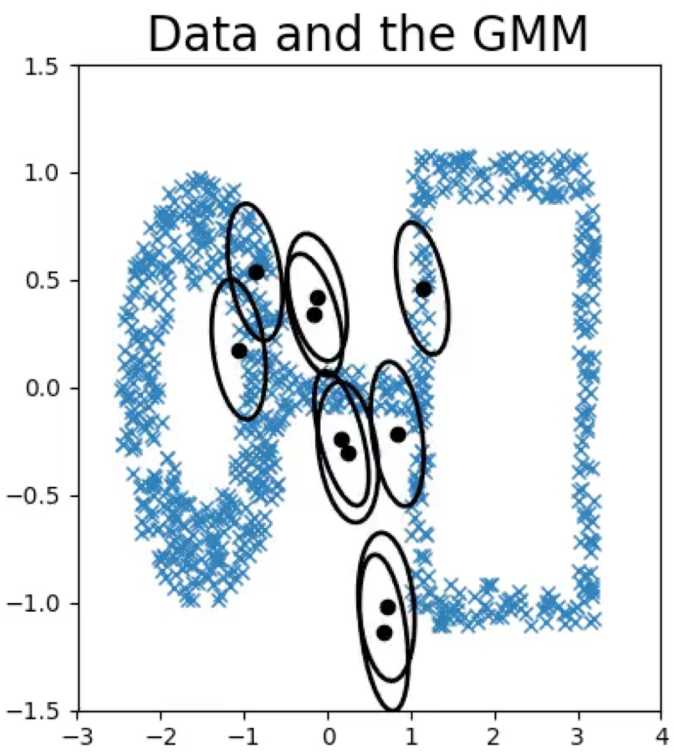
1. Generate L random samples from $\mathbb{S}^{d-1}$, $\Theta = \{\theta_1, \dots, \theta_L\}$.
2. For fixed GMM parameters, slice the GMM with respect to Θ and calculate the optimal transport maps between slices $\mathcal{R}p(\cdot, \theta_l)$ and $\mathcal{R}\hat{p}_{\mathrm{d}}(\cdot, \theta_l)$: $f_{\theta_l}(t) = \mathcal{R}\hat{P}_d^{-1}(\mathcal{R}P(t; \theta); \theta)$
3. Update the GMM parameters using the calculated transport maps, $f_{\theta_l}(t)$.
4. If not converged then go to step 1.

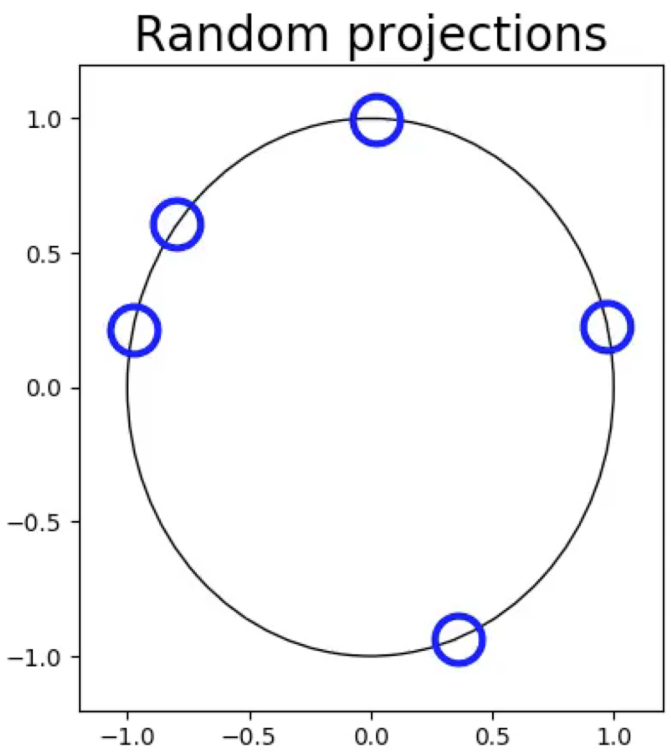


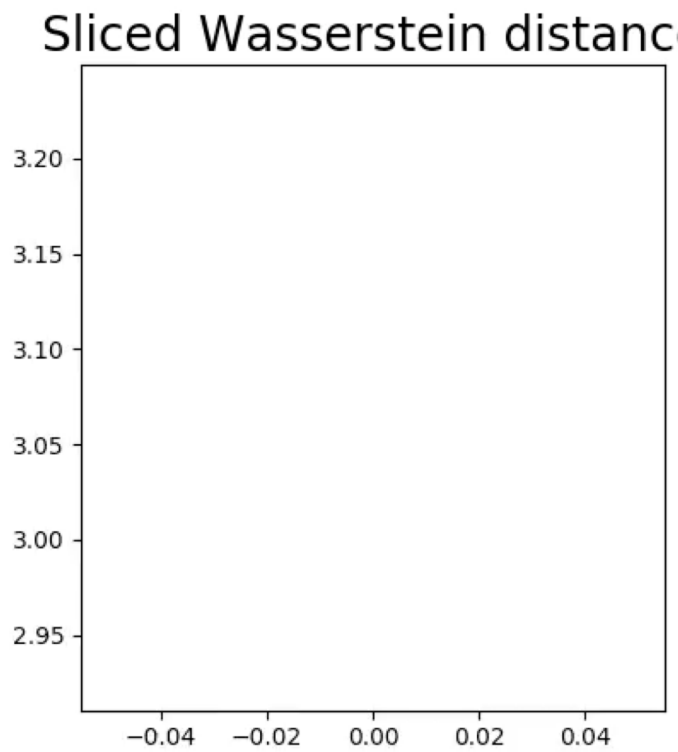
Convergence analysis of SW-GMM

As the dimensionality of the data increases more slices are needed for faster convergence.

Sliced Wasserstein Distance for Learning GMMs

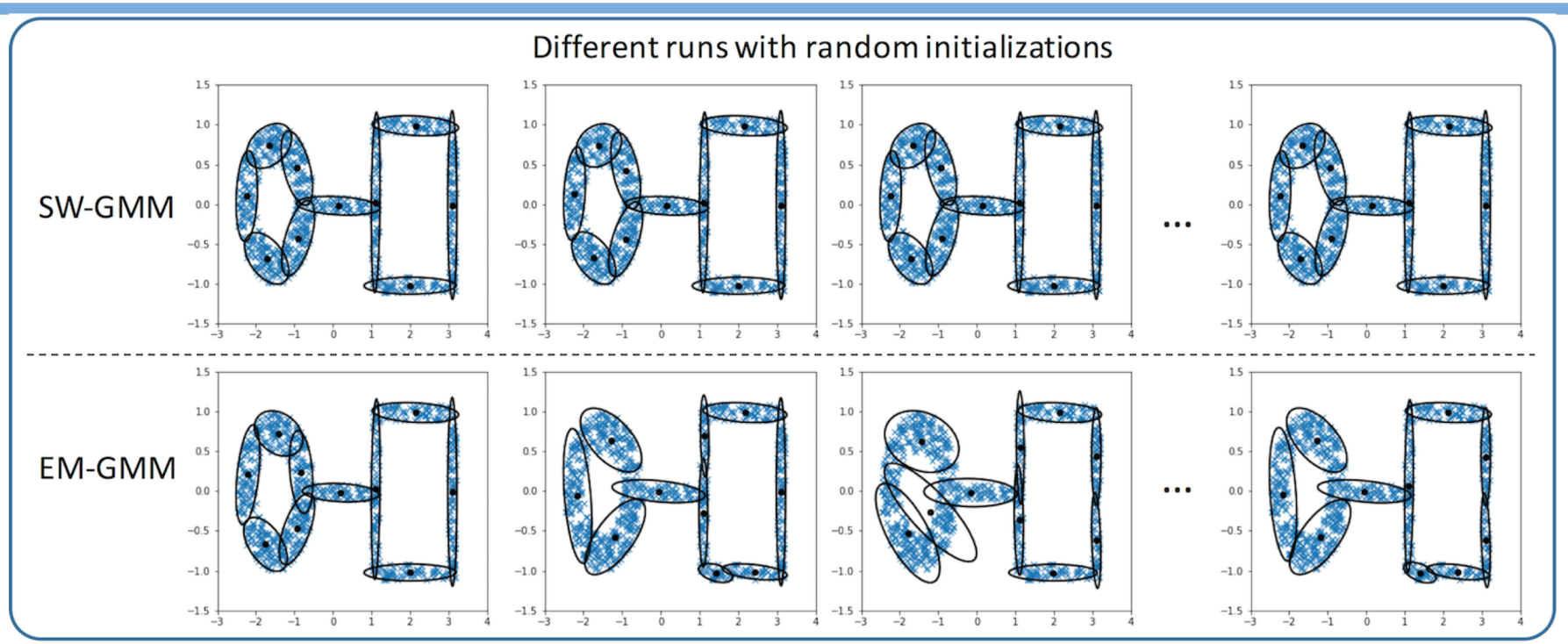


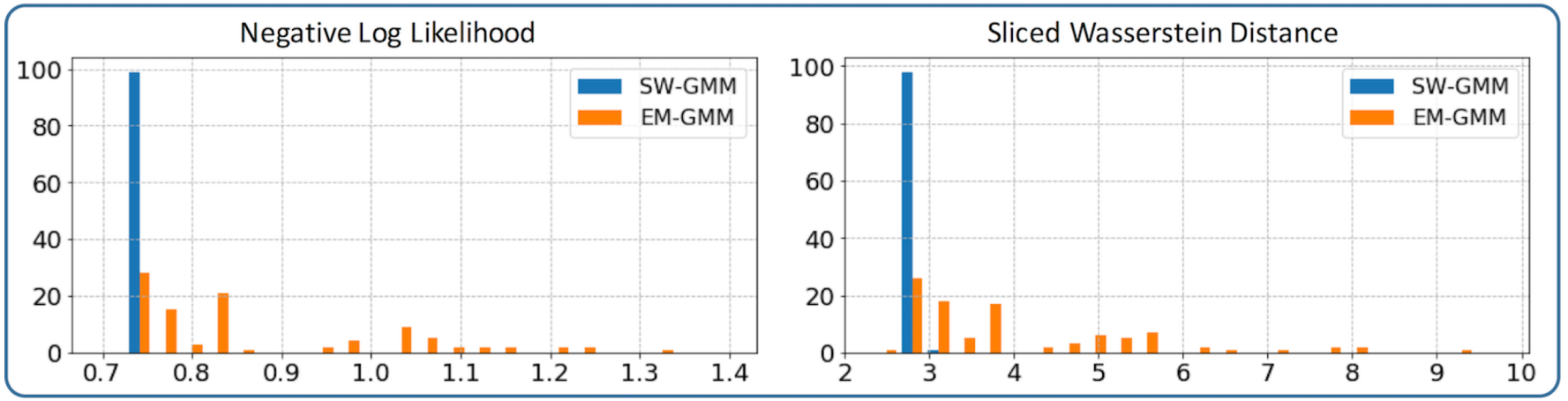




[Link to Code]

SW-GMM vs. EM-GMM





- What are Gaussian Mixture Models (GMMs)?
  - Why should one care?
- What is the EM algorithm?
  - What are the challenges?
- How can we use transport-based distances to estimate GMMs?