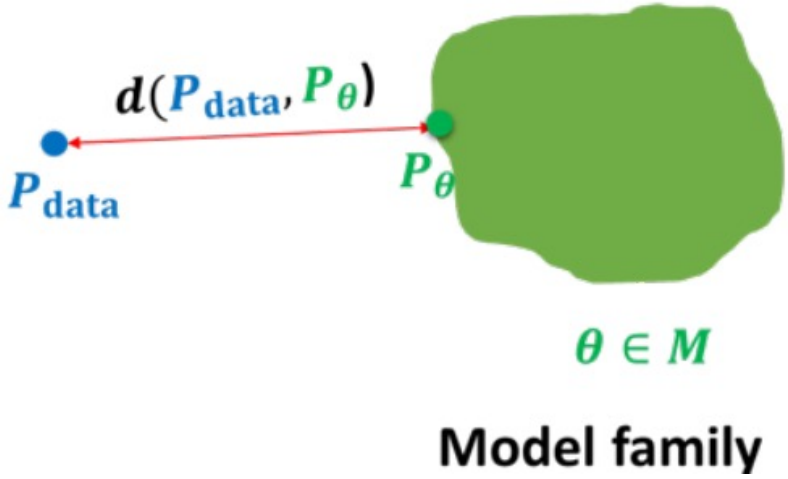
Score-Based Models

LECTURE 13

CS236: Deep Generative Models

Summary

$$x_i \sim P_{\text{data}}$$
$$i = 1, 2, \ldots, n$$



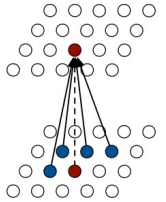
How to represent probability distributions?

- Probability density function (p.d.f.) or probability mass function (p.m.f.)

$$p(\mathbf{x})$$

- Autogressive models

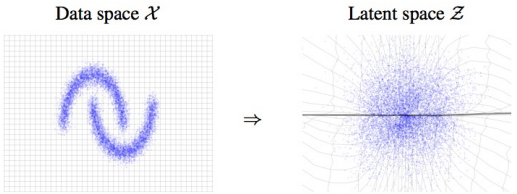
$$p_\theta(\mathbf{x}) = \prod_{i=1}^{d} p_\theta(\mathbf{x}_i \mid \mathbf{x}_{<i})$$



PixelCNN

- Flow models

$$p_\theta(\mathbf{x}) = p(\mathbf{z})\left|\det(J_{f_\theta}(\mathbf{x}))\right|, \quad \mathbf{z} = f_\theta(\mathbf{x})$$



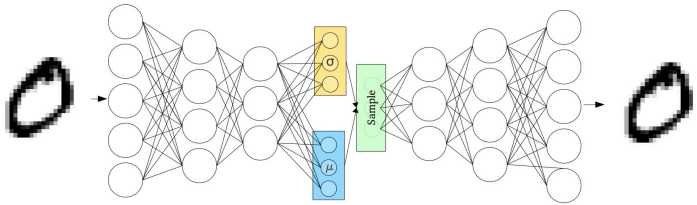
How to represent probability distributions?

- Probability density function (p.d.f.) or probability mass function (p.m.f.)

$$p(\mathbf{x})$$

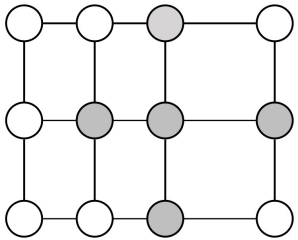
- Variational autoencoders

$$p_\theta(\mathbf{x}) = \int p(\mathbf{z}) p_\theta(\mathbf{x} \mid \mathbf{z}) \, \mathrm{d}\mathbf{z}$$



- Energy-based models

$$p_\theta(\mathbf{x}) = \frac{e^{f_\theta(\mathbf{x})}}{Z(\theta)}$$



How to represent probability distributions?

- Probability density function (p.d.f.) or probability mass function (p.m.f.)

$$p(\mathbf{x})$$

- Pros
  - Maximum likelihood training
  - Principled model comparison via likelihoods

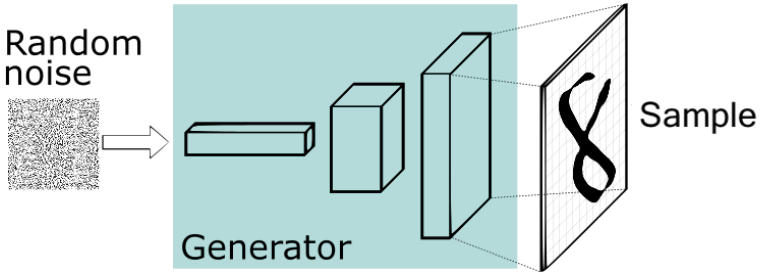
- Cons
  - Special architectures or surrogate losses to deal with intractable partition functions

How to represent probability distributions?

- Sampling process

- Generative adversarial networks (GANs)

$$\mathbf{z} \sim p(\mathbf{z})$$
$$\mathbf{x} = g_\theta(\mathbf{z})$$



GAN

How to represent probability distributions?

- Sampling process

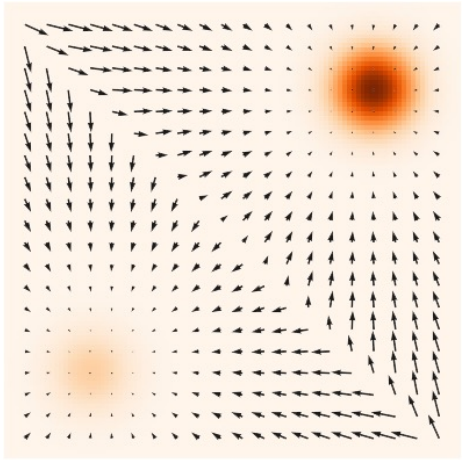
- Pros
  - Samples typically have better quality

- Cons
  - Require adversarial training. Training instability and mode collapse.
  - No principled way to compare different models
  - No principled termination criteria for training

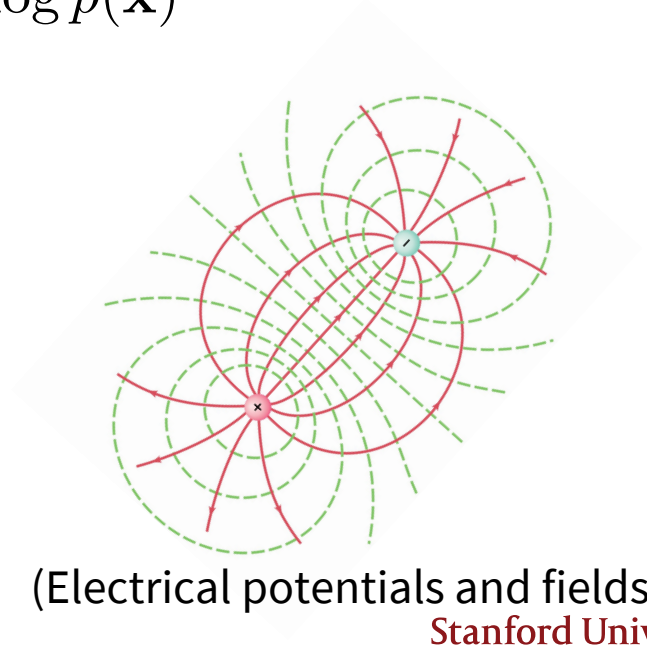
How to represent probability distributions?

- When the pdf is differentiable, we can compute the gradient of a probability density.

Score function $\nabla_{\mathbf{x}} \log p(\mathbf{x})$



(pdf and score)

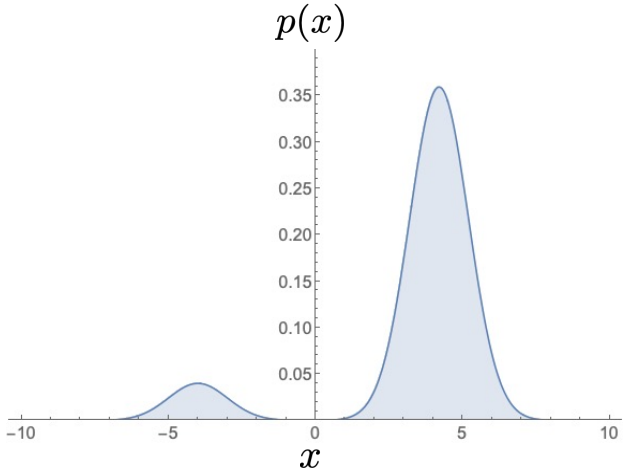


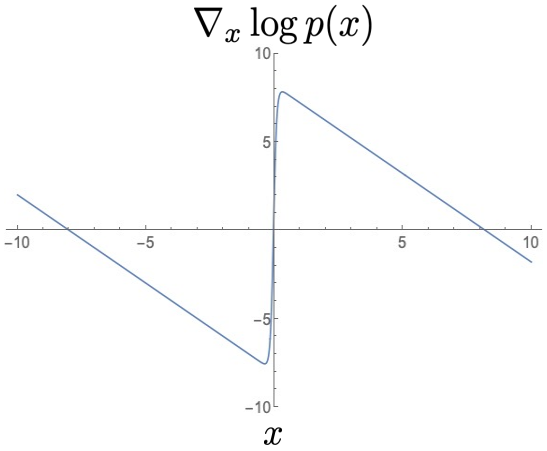
(Electrical potentials and fields)

How to represent probability distributions?

- When the pdf is differentiable, we can compute the gradient of a probability density.

Score function $\nabla_{\mathbf{x}} \log p(\mathbf{x})$



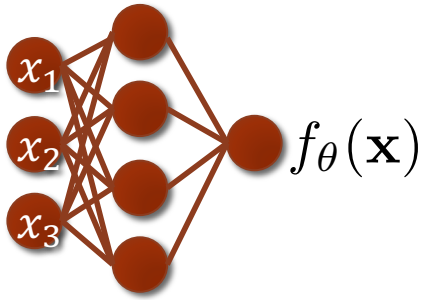


Recap on energy-based models

- Deep Energy-Based models (EBMs)

$$f_\theta(\mathbf{x}) \in \mathbb{R}$$

$$p_\theta(\mathbf{x}) = \frac{e^{f_\theta(\mathbf{x})}}{Z(\theta)}$$



- Maximum likelihood training: $\max_\theta f_\theta(\mathbf{x}_{\text{train}}) - \log Z(\theta)$
  - Contrastive divergence

$$\nabla_\theta f_\theta(\mathbf{x}_{\text{train}}) - \nabla_\theta \log Z(\theta) \approx \nabla_\theta f_\theta(\mathbf{x}_{\text{train}}) - \nabla_\theta f_\theta(\mathbf{x}_{\text{sample}})$$

$$\mathbf{x}_{\text{sample}} \sim p_\theta(\mathbf{x})$$

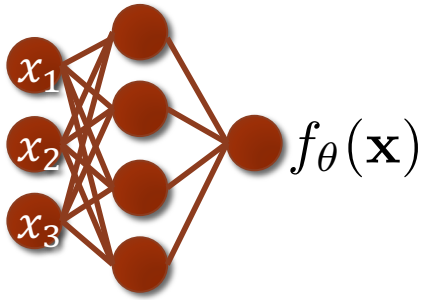
  - Requires iterative sampling during training

Recap on energy-based models

- Deep Energy-Based models (EBMs)

$$f_\theta(\mathbf{x}) \in \mathbb{R}$$

$$p_\theta(\mathbf{x}) = \frac{e^{f_\theta(\mathbf{x})}}{Z(\theta)}$$



- Minimizing Fisher divergence: $\min_\theta \frac{1}{2} E_{\mathbf{x}\sim p_{\text{data}}}[\|\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x})\|_2^2]$
  - Score matching

$$\frac{1}{2} E_{\mathbf{x}\sim p_{\text{data}}}[\|\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x})\|_2^2]$$

$$= E_{\mathbf{x}\sim p_{\text{data}}}\left[\frac{1}{2}\|\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x})\|_2^2 + \text{tr}(\nabla_{\mathbf{x}}^2 \log p_\theta(\mathbf{x}))\right]$$

Score matching for training EBMs

- Score function of EBMs

$$\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) = \nabla_{\mathbf{x}} f_\theta(\mathbf{x}) - \underbrace{\nabla_{\mathbf{x}} \log Z(\theta)}_{=0} = \nabla_{\mathbf{x}} f_\theta(\mathbf{x})$$

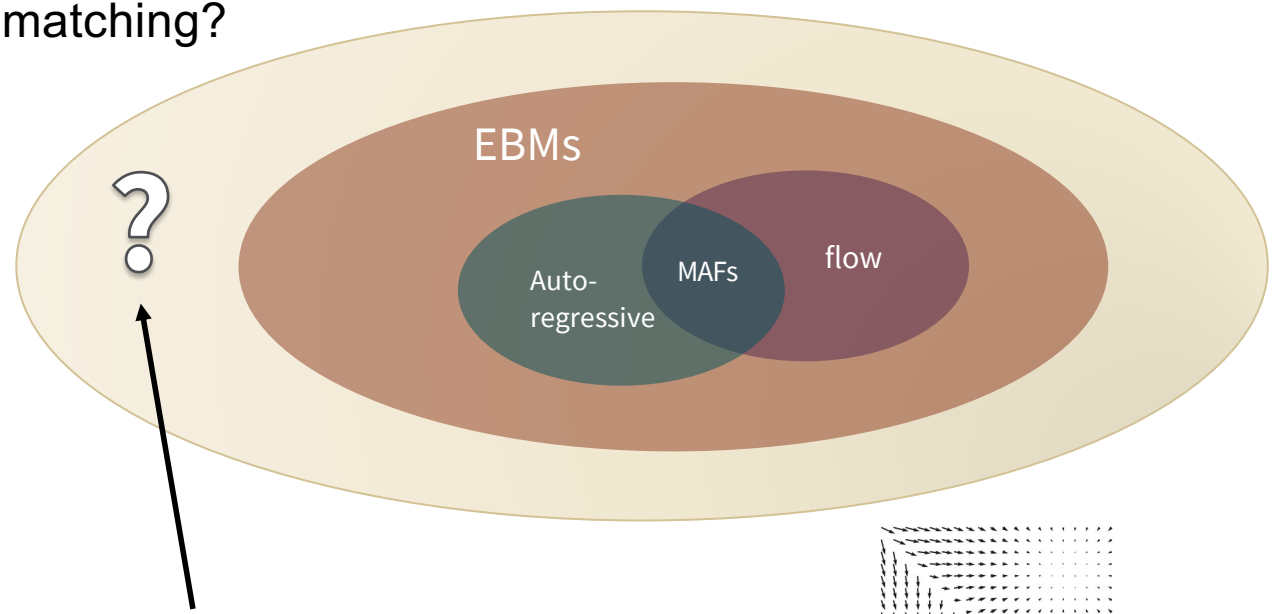
- Score matching for EBMs:

$$E_{\mathbf{x} \sim p_{\text{data}}} \left[\frac{1}{2} \left\| \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) \right\|_2^2 + \text{tr}(\nabla_{\mathbf{x}}^2 \log p_\theta(\mathbf{x})) \right]$$
$$= E_{\mathbf{x} \sim p_{\text{data}}} \left[\frac{1}{2} \left\| \nabla_{\mathbf{x}} f_\theta(\mathbf{x}) \right\|_2^2 + \text{tr}(\nabla_{\mathbf{x}}^2 f_\theta(\mathbf{x})) \right]$$

- Is score matching limited to EBMs?
  - Autoregressive models
  - Normalizing flow models

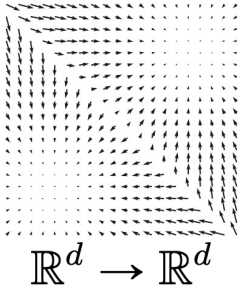
Score-based models

- What's the most general model that can be efficiently trained by score matching?



- Score-based model

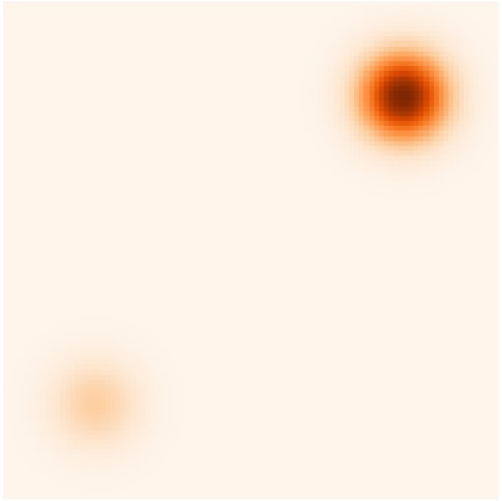
$$s_\theta(\mathbf{x})$$



Score estimation by training score-based models

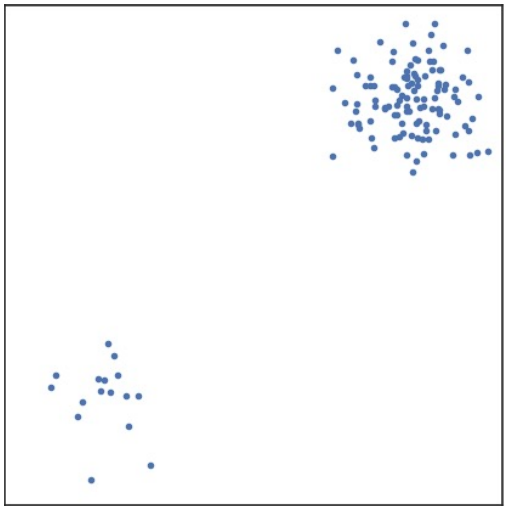
Probability density

$p_{\text{data}}(\mathbf{x})$



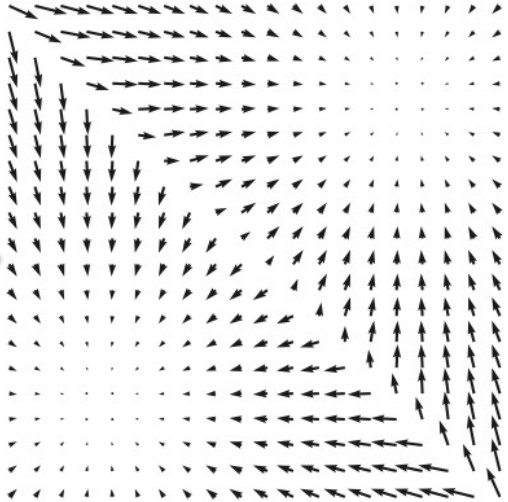
i.i.d. samples

$\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$



Score function

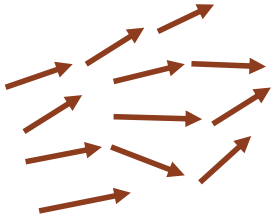
$\boldsymbol{s}_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$



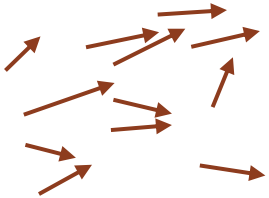
Score estimation by training score-based models

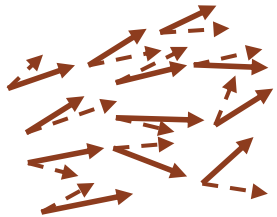
- **Given:** i.i.d. samples $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \overset{\text{i.i.d.}}{\sim} p(\mathbf{x})$
- **Task:** Estimating the score $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$
- **Score Model:** A learnable vector-valued function $s_\theta(\mathbf{x}) : \mathbb{R}^D \to \mathbb{R}^D$
- **Goal:** $s_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$
- How to compare two vector fields of scores?

$\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$



$s_\theta(\mathbf{x})$





Average Euclidean distance over the space

Score estimation by training score-based models

- **Objective:** Average Euclidean distance over the whole space.

$$\frac{1}{2} E_{\mathbf{x} \sim p_{\text{data}}} [\| \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \boldsymbol{s}_\theta(\mathbf{x}) \|_2^2]$$

(Fisher divergence)

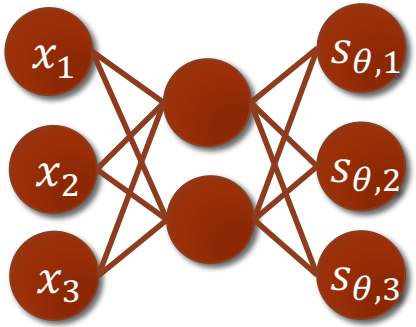
- **Score matching:**

$$E_{\mathbf{x} \sim p_{\text{data}}} \left[\frac{1}{2} \| \boldsymbol{s}_\theta(\mathbf{x}) \|_2^2 + \text{tr}(\underbrace{\nabla_{\mathbf{x}} \boldsymbol{s}_\theta(\mathbf{x})}_{\text{Jacobian of } \boldsymbol{s}_\theta(\mathbf{x})}) \right]$$

- **Requirements:**
  - The score model must be efficient to evaluate.
  - Do we need the score model to be a proper score function?

Score matching is not scalable

- Deep neural networks as more expressive score models

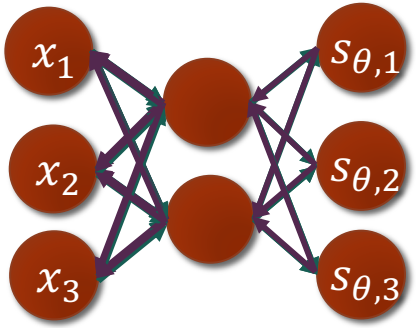


$$s_\theta(\mathbf{x})$$

Score Matching is not Scalable!

- Compute $\|s_\theta(\mathbf{x})\|_2^2$ and $\mathrm{tr}(\nabla_{\mathbf{x}} s_\theta(\mathbf{x}))$

$$\frac{\partial s_{\theta,1}(\mathbf{x})}{\partial x_1} \quad \frac{\partial s_{\theta,2}(\mathbf{x})}{\partial x_2} \quad \frac{\partial s_{\theta,3}(\mathbf{x})}{\partial x_3}$$



$$s_{\theta,1}(\mathbf{x})$$
$$s_{\theta,2}(\mathbf{x})$$
$$s_{\theta,3}(\mathbf{x})$$

$O(D)$ Backprops!

$$\nabla_{\mathbf{x}} s_\theta(\mathbf{x}) = \begin{pmatrix} \frac{\partial s_{\theta,1}(\mathbf{x})}{\partial x_1} & \frac{\partial s_{\theta,1}(\mathbf{x})}{\partial x_2} & \frac{\partial s_{\theta,1}(\mathbf{x})}{\partial x_3} \\ \frac{\partial s_{\theta,2}(\mathbf{x})}{\partial x_1} & \frac{\partial s_{\theta,2}(\mathbf{x})}{\partial x_2} & \frac{\partial s_{\theta,2}(\mathbf{x})}{\partial x_3} \\ \frac{\partial s_{\theta,3}(\mathbf{x})}{\partial x_1} & \frac{\partial s_{\theta,3}(\mathbf{x})}{\partial x_2} & \frac{\partial s_{\theta,3}(\mathbf{x})}{\partial x_3} \end{pmatrix}$$

Denoising score matching

$\mathbf{x}$

$p_{\text{data}}(\mathbf{x})$

$q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})$

$\tilde{\mathbf{x}}$

$q_\sigma(\tilde{\mathbf{x}})$

- Denoising score matching (Vincent 2011): matching the score of a noise-perturbed distribution

$$\frac{1}{2} E_{\tilde{\mathbf{x}} \sim q_\sigma}[\|\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}) - \boldsymbol{s}_\theta(\tilde{\mathbf{x}})\|_2^2]$$

$$= \frac{1}{2} \int q_\sigma(\tilde{\mathbf{x}}) \|\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}) - \boldsymbol{s}_\theta(\tilde{\mathbf{x}})\|_2^2 \, \mathrm{d}\tilde{\mathbf{x}}$$

$$= \frac{1}{2} \int q_\sigma(\tilde{\mathbf{x}}) \|\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})\|_2^2 \, \mathrm{d}\tilde{\mathbf{x}} + \frac{1}{2} \int q_\sigma(\tilde{\mathbf{x}}) \|\boldsymbol{s}_\theta(\tilde{\mathbf{x}})\|_2^2 \, \mathrm{d}\tilde{\mathbf{x}}$$

$$- \int q_\sigma(\tilde{\mathbf{x}}) \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})^\mathsf{T} \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) \, \mathrm{d}\tilde{\mathbf{x}}$$

$$= \text{const.} + \frac{1}{2} E_{\tilde{\mathbf{x}} \sim q_\sigma}[\|\boldsymbol{s}_\theta(\tilde{\mathbf{x}})\|_2^2] - \int q_\sigma(\tilde{\mathbf{x}}) \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})^\mathsf{T} \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) \, \mathrm{d}\tilde{\mathbf{x}}$$

Denoising score matching

$\mathbf{x}$

$p_{\text{data}}(\mathbf{x})$

$q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})$

$\tilde{\mathbf{x}}$

$q_\sigma(\tilde{\mathbf{x}})$

- Denoising score matching

$$
\begin{aligned}
&-\int q_\sigma(\tilde{\mathbf{x}}) \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})^\mathsf{T} \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) \, \mathrm{d}\tilde{\mathbf{x}} \\
&= -\int q_\sigma(\tilde{\mathbf{x}}) \frac{1}{q_\sigma(\tilde{\mathbf{x}})} \nabla_{\tilde{\mathbf{x}}} q_\sigma(\tilde{\mathbf{x}})^\mathsf{T} \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) \, \mathrm{d}\tilde{\mathbf{x}} \\
&= -\int \nabla_{\tilde{\mathbf{x}}} q_\sigma(\tilde{\mathbf{x}})^\mathsf{T} \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) \, \mathrm{d}\tilde{\mathbf{x}} \\
&= -\int \nabla_{\tilde{\mathbf{x}}} \left(\int p_{\text{data}}(\mathbf{x}) q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) \, \mathrm{d}\mathbf{x} \right)^\mathsf{T} \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) \, \mathrm{d}\tilde{\mathbf{x}} \\
&= -\int \left(\int p_{\text{data}}(\mathbf{x}) \nabla_{\tilde{\mathbf{x}}} q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) \, \mathrm{d}\mathbf{x} \right)^\mathsf{T} \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) \, \mathrm{d}\tilde{\mathbf{x}} \\
&= -\int \left(\int p_{\text{data}}(\mathbf{x}) q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) \, \mathrm{d}\mathbf{x} \right)^\mathsf{T} \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) \, \mathrm{d}\tilde{\mathbf{x}} \\
&= -\iint p_{\text{data}}(\mathbf{x}) q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})^\mathsf{T} \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) \, \mathrm{d}\mathbf{x} \, \mathrm{d}\tilde{\mathbf{x}} \\
&= -E_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x}), \tilde{\mathbf{x}} \sim q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})} [\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})^\mathsf{T} \boldsymbol{s}_\theta(\tilde{\mathbf{x}})]
\end{aligned}
$$

Denoising score matching

$\mathbf{x}$

$p_{\text{data}}(\mathbf{x})$

$q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})$

$q_\sigma(\tilde{\mathbf{x}})$

$\tilde{\mathbf{x}}$

- Denoising score matching

$$\frac{1}{2} E_{\tilde{\mathbf{x}} \sim q_\sigma}[\| \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}) - \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) \|_2^2]$$

$$= \text{const.} + \frac{1}{2} E_{\tilde{\mathbf{x}} \sim q_\sigma}[\| \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) \|_2^2] - \int q_\sigma(\tilde{\mathbf{x}}) \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})^\intercal \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) \, \mathrm{d}\tilde{\mathbf{x}}$$

$$= \text{const.} + \frac{1}{2} E_{\tilde{\mathbf{x}} \sim q_\sigma}[\| \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) \|_2^2] - E_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x}), \tilde{\mathbf{x}} \sim q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})}[\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})^\intercal \boldsymbol{s}_\theta(\tilde{\mathbf{x}})]$$

$$= \text{const.} + \frac{1}{2} E_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x}), \tilde{\mathbf{x}} \sim q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})}[\| \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) \|_2^2]$$

$$- \frac{1}{2} E_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x}), \tilde{\mathbf{x}} \sim q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})}[\| \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) \|_2^2]$$

$$= \text{const.} + \frac{1}{2} E_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x}), \tilde{\mathbf{x}} \sim q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})}[\| \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) \|_2^2] + \text{const.}$$

$$= \frac{1}{2} E_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x}), \tilde{\mathbf{x}} \sim q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})}[\| \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) \|_2^2] + \text{const.}$$

Denoising score matching

- Estimate the score of a noise-perturbed distribution

$$\frac{1}{2} E_{\tilde{\mathbf{x}} \sim p_{\text{data}}} [\| \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}) \|_2^2]$$
$$= \frac{1}{2} E_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x}), \tilde{\mathbf{x}} \sim q_\sigma(\tilde{\mathbf{x}} | \mathbf{x})} [\| \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) \|_2^2] + \text{const.}$$

- $\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})$ is easy to compute
  - $q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}} \mid \mathbf{x}, \sigma^2 \boldsymbol{I})$
  - $\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = -\frac{\tilde{\mathbf{x}} - \mathbf{x}}{\sigma^2}$
- **Pros:** efficient to optimize even for very high dimensional data, and useful for optimal denoising.
- **Con:** cannot estimate the score of clean data (noise-free)

Denoising score matching

- Sample a minibatch of datapoints $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\} \sim p_{\text{data}}(\mathbf{x})$
- Sample a minibatch of perturbed datapoints $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \cdots, \tilde{\mathbf{x}}_n\} \sim q_\sigma(\tilde{\mathbf{x}})$

$$\tilde{\mathbf{x}}_i \sim q_\sigma(\tilde{\mathbf{x}}_i \mid \mathbf{x}_i)$$

- Estimate the denoising score matching loss with empirical means

$$\frac{1}{2n}\sum_{i=1}^{n}[\|\boldsymbol{s}_\theta(\tilde{\mathbf{x}}_i) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}_i \mid \mathbf{x}_i)\|_2^2]$$

- If Gaussian perturbation

$$\frac{1}{2n}\sum_{i=1}^{n}\left[\left\|\boldsymbol{s}_\theta(\tilde{\mathbf{x}}_i) + \frac{\tilde{\mathbf{x}}_i - \mathbf{x}_i}{\sigma^2}\right\|_2^2\right]$$

- Stochastic gradient descent
- Need to choose a very small σ!

Pitfall of denoising score matching

- The loss variance will increase drastically as $\sigma \to 0$!
- Denoising score matching loss for Gaussian perturbations

$$\begin{aligned}
&\frac{1}{2} E_{\mathbf{x} \sim p_{\text{data}}} E_{\tilde{\mathbf{x}} \sim q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})} \left[\left\| \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) + \frac{\tilde{\mathbf{x}} - \mathbf{x}}{\sigma^2} \right\|_2^2 \right] \\
&= \frac{1}{2} E_{\mathbf{x} \sim p_{\text{data}}} E_{\mathbf{z} \sim \mathcal{N}(0, \boldsymbol{I})} \left[\left\| \boldsymbol{s}_\theta(\mathbf{x} + \sigma \mathbf{z}) + \frac{\mathbf{z}}{\sigma} \right\|_2^2 \right] \qquad \text{(reparameterization trick)} \\
&= \frac{1}{2} E_{\mathbf{x} \sim p_{\text{data}}} E_{\mathbf{z} \sim \mathcal{N}(0, \boldsymbol{I})} \left[\| \boldsymbol{s}_\theta(\mathbf{x} + \sigma \mathbf{z}) \|_2^2 + 2 \boldsymbol{s}_\theta(\mathbf{x} + \sigma \mathbf{z})^\intercal \frac{\mathbf{z}}{\sigma} + \frac{\|\mathbf{z}\|_2^2}{\sigma^2} \right]
\end{aligned}$$

- If we choose very small σ

$$\text{Var}\left(\frac{\mathbf{z}}{\sigma} \right) \to \infty$$

$$\text{Var}\left(\frac{\|\mathbf{z}\|_2^2}{\sigma^2} \right) \to \infty$$

Tweedie's formula and denoising score matching

- Denoising score matching is suitable for optimal denoising
- Given $p(\mathbf{x}), \quad q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}} \mid \mathbf{x}, \sigma^2 \boldsymbol{I})$, we can define the posterior $p(\mathbf{x} \mid \tilde{\mathbf{x}})$ with Bayes' rule

$$p(\mathbf{x} \mid \tilde{\mathbf{x}}) \propto p(\mathbf{x}) q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})$$

- Recall that

$$q_\sigma(\tilde{\mathbf{x}}) = \int p_{\text{data}}(\mathbf{x}) q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) \, \mathrm{d}\mathbf{x}$$

- Tweedie's formula:

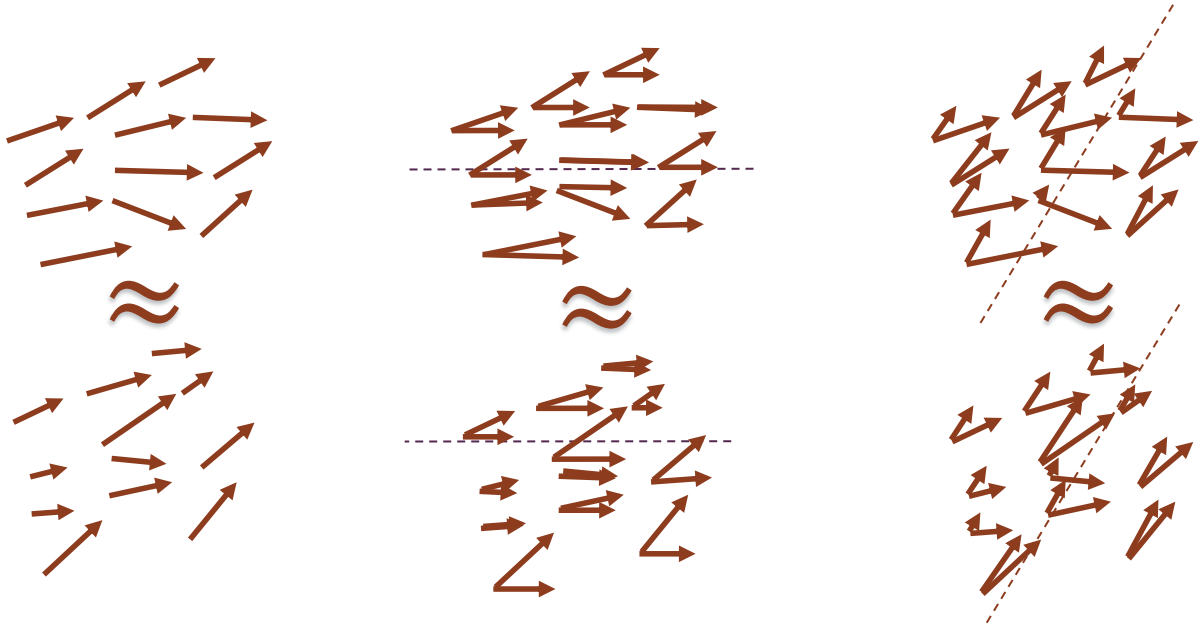
$$\begin{aligned} E_{\mathbf{x} \sim p(\mathbf{x} \mid \tilde{\mathbf{x}})}[\mathbf{x}] &= \tilde{\mathbf{x}} + \sigma^2 \nabla_{\mathbf{x}} \log q_\sigma(\tilde{\mathbf{x}}) \\ &\approx \tilde{\mathbf{x}} + \sigma^2 \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) \end{aligned}$$

Sliced score matching

- One dimensional problems should be easier.
- Consider projections onto random directions.

$\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$

$s_\theta(\mathbf{x})$



Song*, Garg*, Shi, Ermon. "Sliced Score Matching: A Scalable Approach to Density and Score Estimation." UAI 2019.

Sliced score matching

- **Objective:** Sliced Fisher Divergence

$$\frac{1}{2} E_{\mathbf{v}\sim p_{\mathbf{v}}} E_{\mathbf{x}\sim p_{\text{data}}} [(\mathbf{v}^\intercal \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \mathbf{v}^\intercal \boldsymbol{s}_\theta(\mathbf{x}))^2]$$

- **Integration by parts**

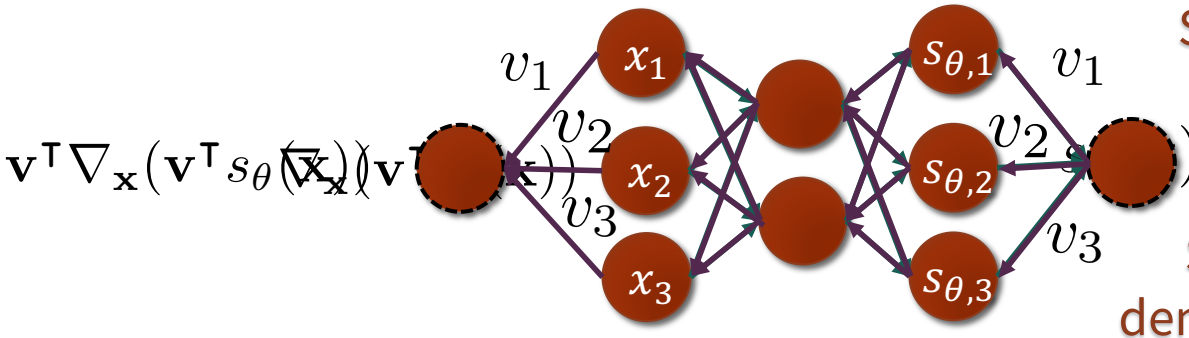
$$E_{\mathbf{v}\sim p_{\mathbf{v}}} E_{\mathbf{x}\sim p_{\text{data}}} \left[\mathbf{v}^\intercal \nabla_{\mathbf{x}} \boldsymbol{s}_\theta(\mathbf{x}) \mathbf{v} + \frac{1}{2} (\mathbf{v}^\intercal \boldsymbol{s}_\theta(\mathbf{x}))^2 \right]$$

$$\begin{pmatrix} v_1 & v_2 & v_3 \end{pmatrix} \begin{pmatrix} \frac{\partial s_{\theta,1}(\mathbf{x})}{\partial x_1} & \frac{\partial s_{\theta,1}(\mathbf{x})}{\partial x_2} & \frac{\partial s_{\theta,1}(\mathbf{x})}{\partial x_3} \\ \frac{\partial s_{\theta,2}(\mathbf{x})}{\partial x_1} & \frac{\partial s_{\theta,2}(\mathbf{x})}{\partial x_2} & \frac{\partial s_{\theta,2}(\mathbf{x})}{\partial x_3} \\ \frac{\partial s_{\theta,3}(\mathbf{x})}{\partial x_1} & \frac{\partial s_{\theta,3}(\mathbf{x})}{\partial x_2} & \frac{\partial s_{\theta,3}(\mathbf{x})}{\partial x_3} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

Sliced Score Matching: fast?

Computing Jacobian-vector products is scalable

$$\mathbf{v}^\intercal \nabla_{\mathbf{x}} s_\theta(\mathbf{x}) \mathbf{v} = \mathbf{v}^\intercal \nabla_{\mathbf{x}} (\mathbf{v}^\intercal s_\theta(\mathbf{x}))$$



One Backprop!

Sliced Score Matching is scalable

Slightly slower than denoising score matching

Sliced score matching

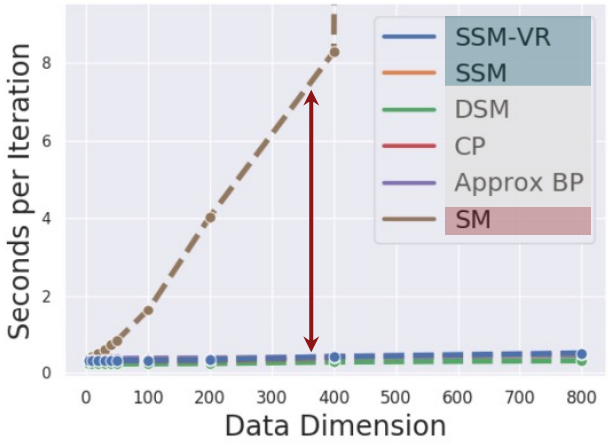
- Sample a minibatch of datapoints $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\} \sim p_{\text{data}}(\mathbf{x})$
- Sample a minibatch of projection directions $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n\} \sim p_{\mathbf{v}}$
- Estimate the sliced score matching loss with empirical means

$$\frac{1}{n}\sum_{i=1}^{n}\left[\mathbf{v}_i^\intercal \nabla_{\mathbf{x}} \boldsymbol{s}_\theta(\mathbf{x}_i)\mathbf{v}_i + \frac{1}{2}(\mathbf{v}_i^\intercal \boldsymbol{s}_\theta(\mathbf{x}_i))^2\right]$$

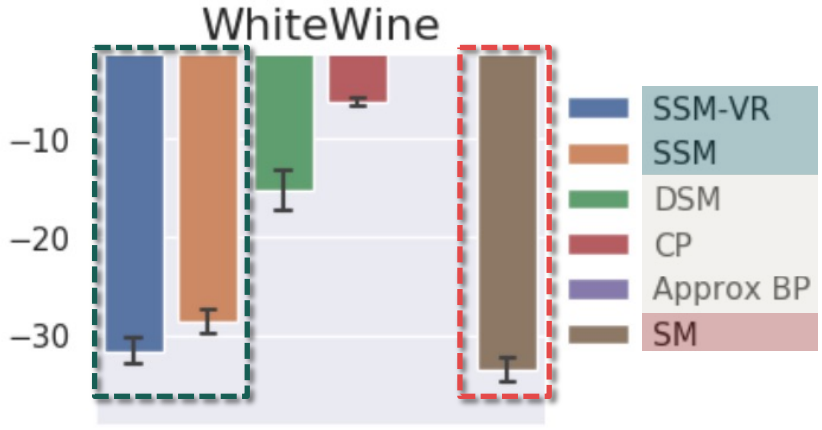
- The perturbation distribution is typically Gaussian or Rademacher
- Stochastic gradient descent
- Can use more projections per datapoint to boost performance

Experimental results for sliced score matching

Sliced score matching methods | Other Baselines | Score Matching



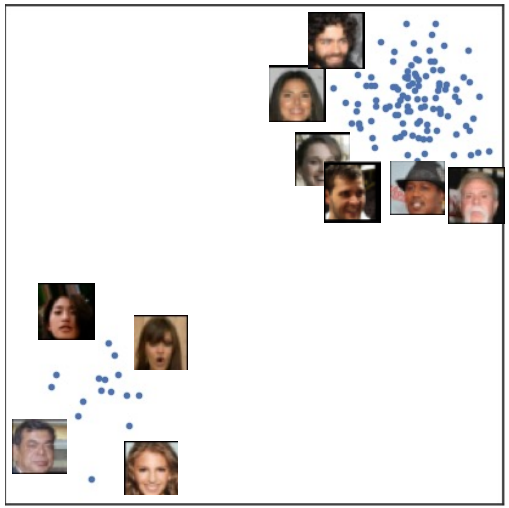
Efficiency



Performance on density estimation

Song*, Garg*, Shi, Ermon. "Sliced Score Matching: A Scalable Approach to Density and Score Estimation." UAI 2019.

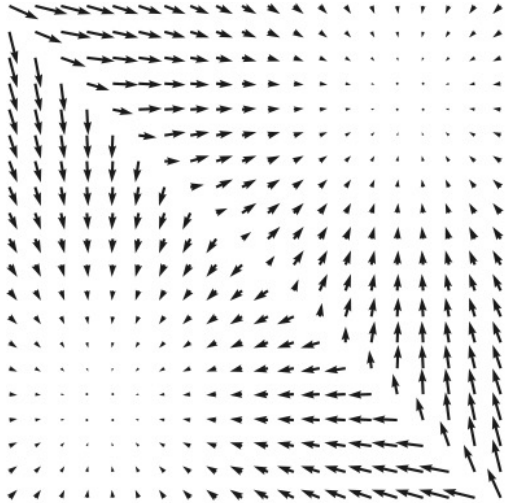
Score-based generative modeling



Data samples

$$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\} \sim p_{\text{data}}(\mathbf{x})$$

Score Matching



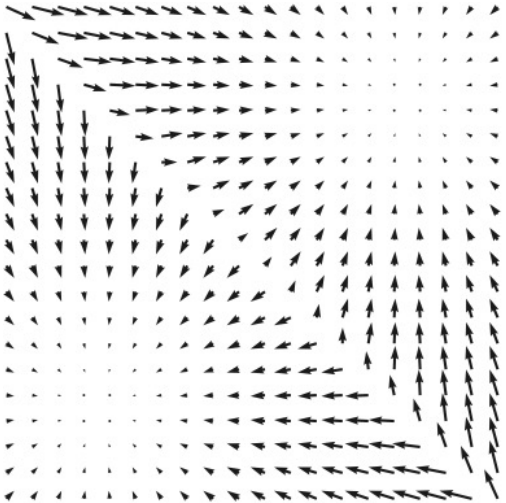
Scores

$$\boldsymbol{s}_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$$

?

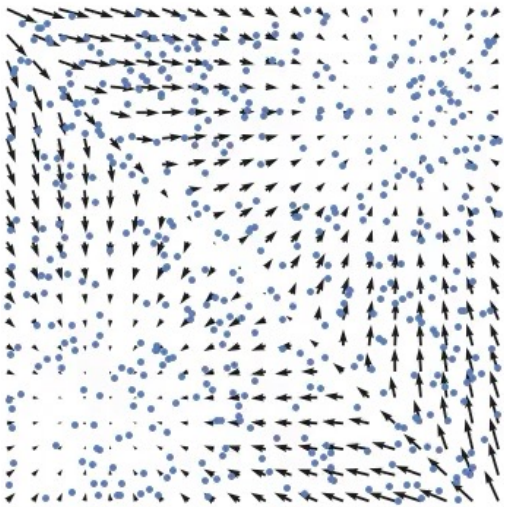
New samples

From scores to samples: Langevin MCMC



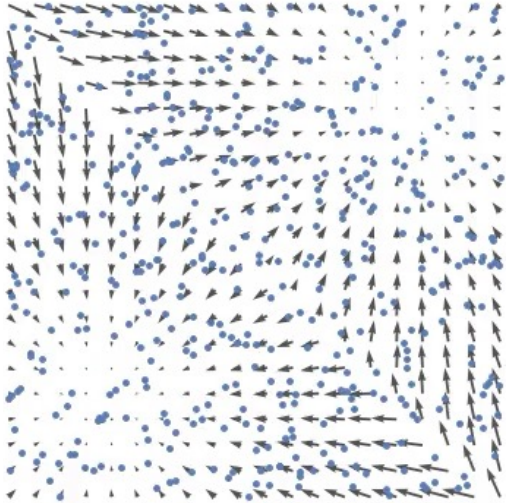
Scores

$\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})$



Follow the scores

$$\tilde{\mathbf{x}}_{t+1} \leftarrow \tilde{\mathbf{x}}_t + \frac{\epsilon}{2}\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_t)$$



Follow noisy scores:
Langevin MCMC

$$\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\tilde{\mathbf{x}}_{t+1} \leftarrow \tilde{\mathbf{x}}_t + \frac{\epsilon}{2}\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_t) + \sqrt{\epsilon}\,\mathbf{z}_t$$

Langevin dynamics sampling

- Sample from $p(\mathbf{x})$ using only the score $\nabla_{\mathbf{x}} \log p(\mathbf{x})$
- Initialize $\mathbf{x}^0 \sim \pi(\mathbf{x})$
- Repeat for $t \leftarrow 1, 2, \cdots, T$

$$\mathbf{z}^t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

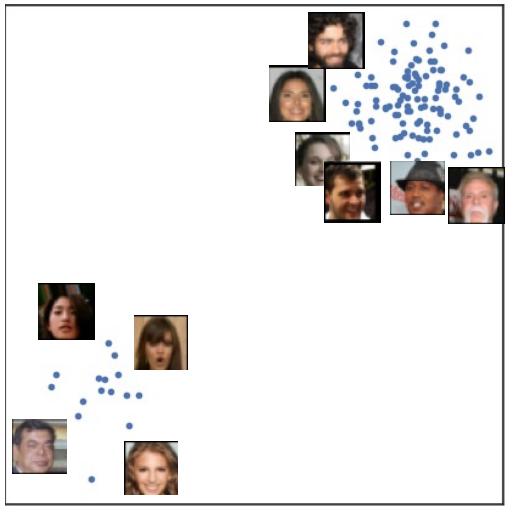
$$\mathbf{x}^t \leftarrow \mathbf{x}^{t-1} + \frac{\epsilon}{2} \nabla_{\mathbf{x}} \log p(\mathbf{x}^{t-1}) + \sqrt{\epsilon}\, \mathbf{z}^t$$

- If $\epsilon \to 0$ and $T \to \infty$, we are guaranteed to have $\mathbf{x}^T \sim p(\mathbf{x})$

- Langevin dynamics + score estimation

$$\boldsymbol{s}_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p(\mathbf{x})$$

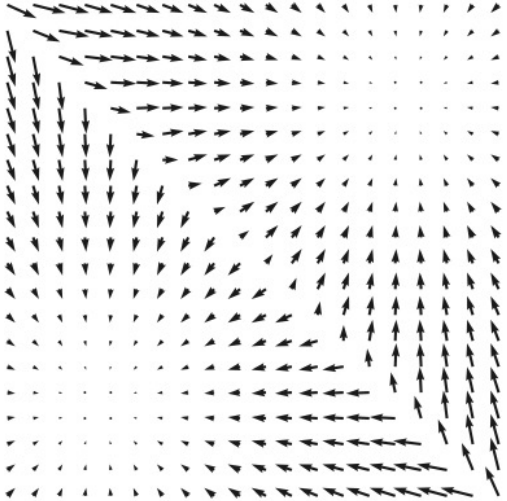
Score-based generative modeling



Data samples

$$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\} \sim p_{\text{data}}(\mathbf{x})$$

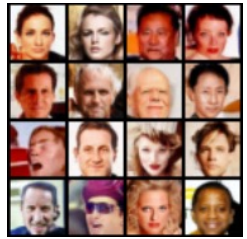
score matching



Scores

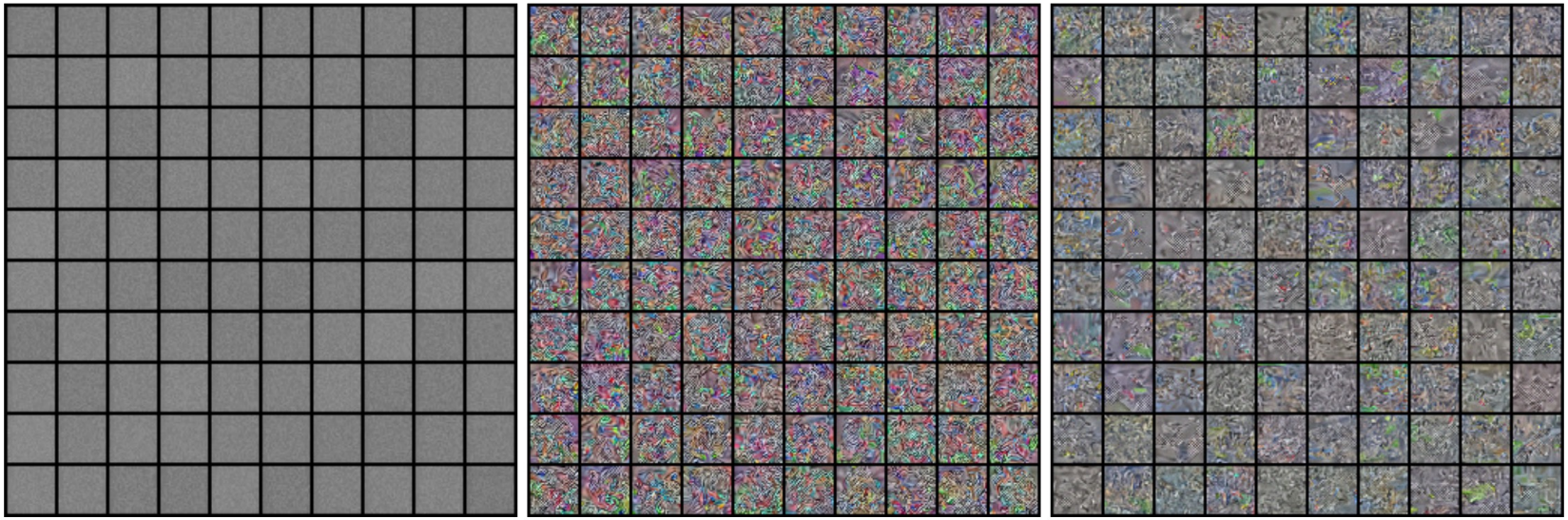
$$\boldsymbol{s}_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$$

Langevin dynamics



New samples

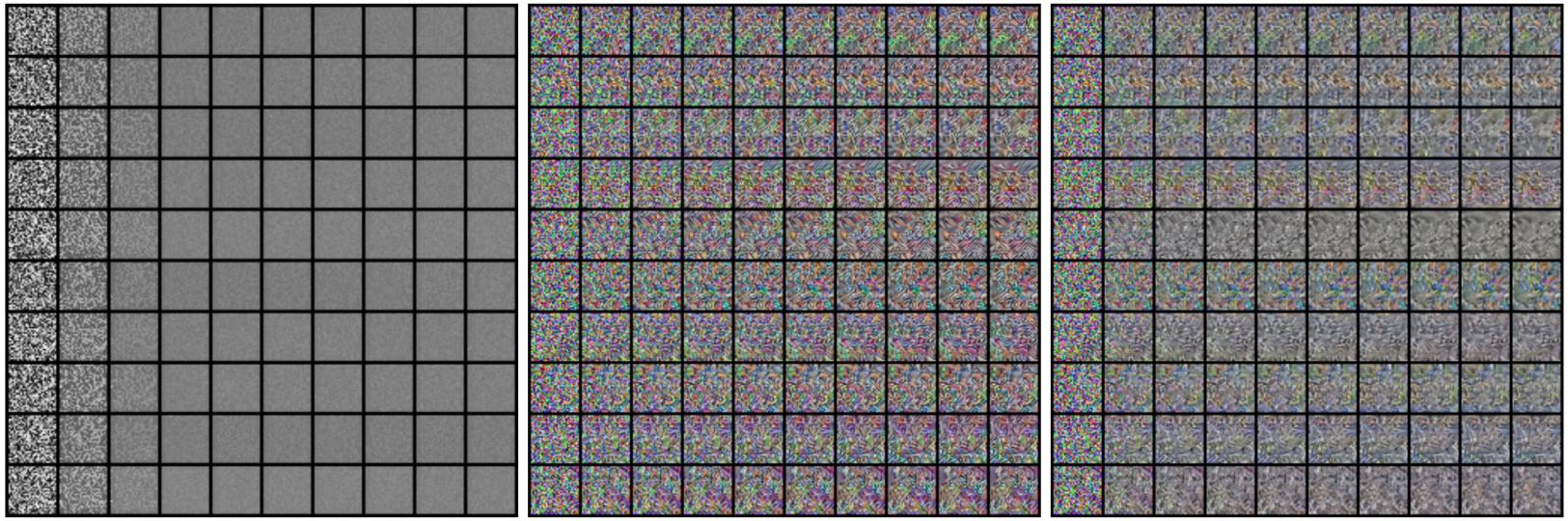
Score-based generative modeling: results



(a) MNIST (b) CelebA (c) CIFAR-10

Final samples

Score-based generative modeling: results

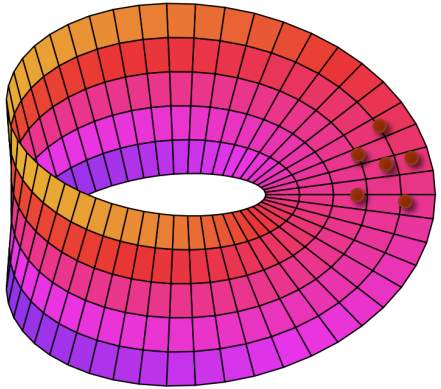


(a) MNIST (b) CelebA (c) CIFAR-10

Langevin sampling process

Pitfall 1: manifold hypothesis

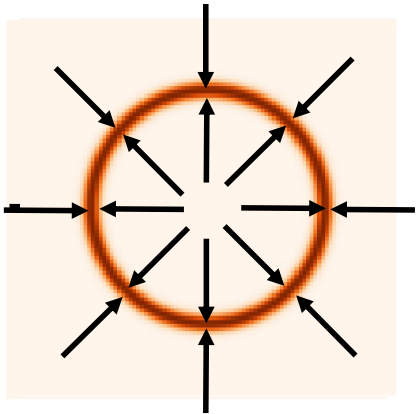
- Manifold hypothesis.



Data points

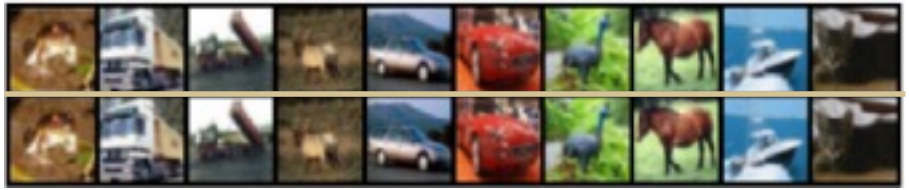
- Data score is undefined.

$$\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$$

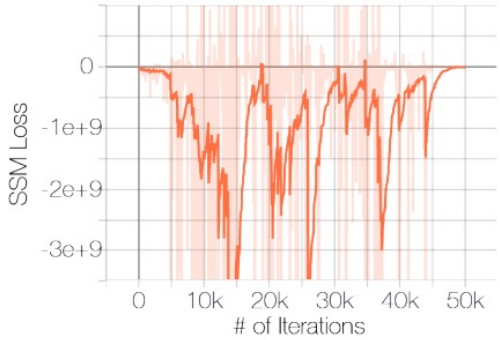


Pitfall 1: manifold hypothesis

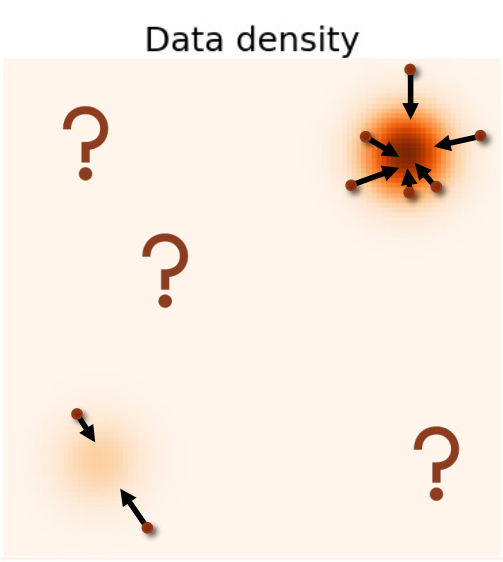
- Fitting the data with a low-dimensional linear manifold (PCA)

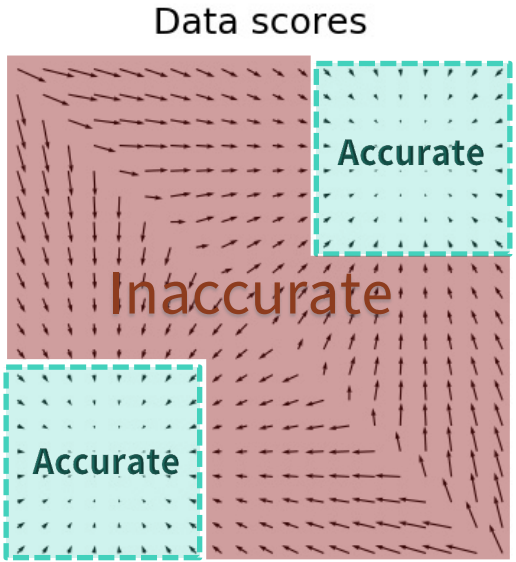


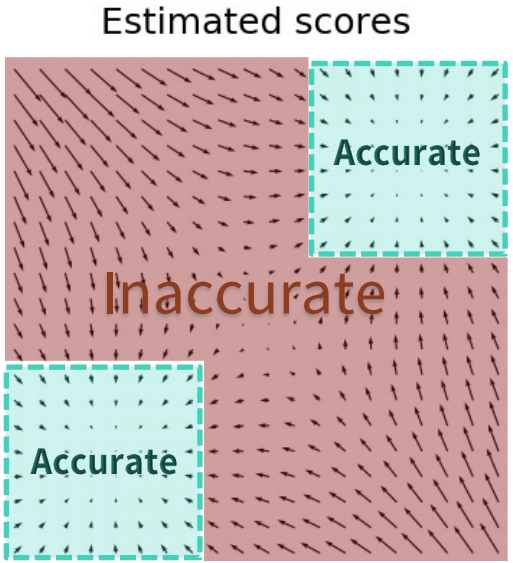
- Score estimation on CIFAR-10.



Challenge in low data density regions







$$\frac{1}{2}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}[\|\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})\|_2^2]$$

Langevin MCMC will have trouble exploring low density regions

Song and Ermon. "Generative Modeling by Estimating Gradients of the Data Distribution." NeurIPS 2019.

Pitfall 3: slow mixing of Langevin dynamics between data modes

- Suppose the data distribution has two disajoint modes:

$$p_{\text{data}}(\mathbf{x}) = \pi p_1(\mathbf{x}) + (1-\pi)p_2(\mathbf{x})$$

$$\mathcal{A} \cap \mathcal{B} = \varnothing \qquad p_{\text{data}}(\mathbf{x}) = \begin{cases} \pi p_1(\mathbf{x}), & \mathbf{x} \in \mathcal{A} \\ (1-\pi)p_2(\mathbf{x}), & \mathbf{x} \in \mathcal{B} \end{cases}$$

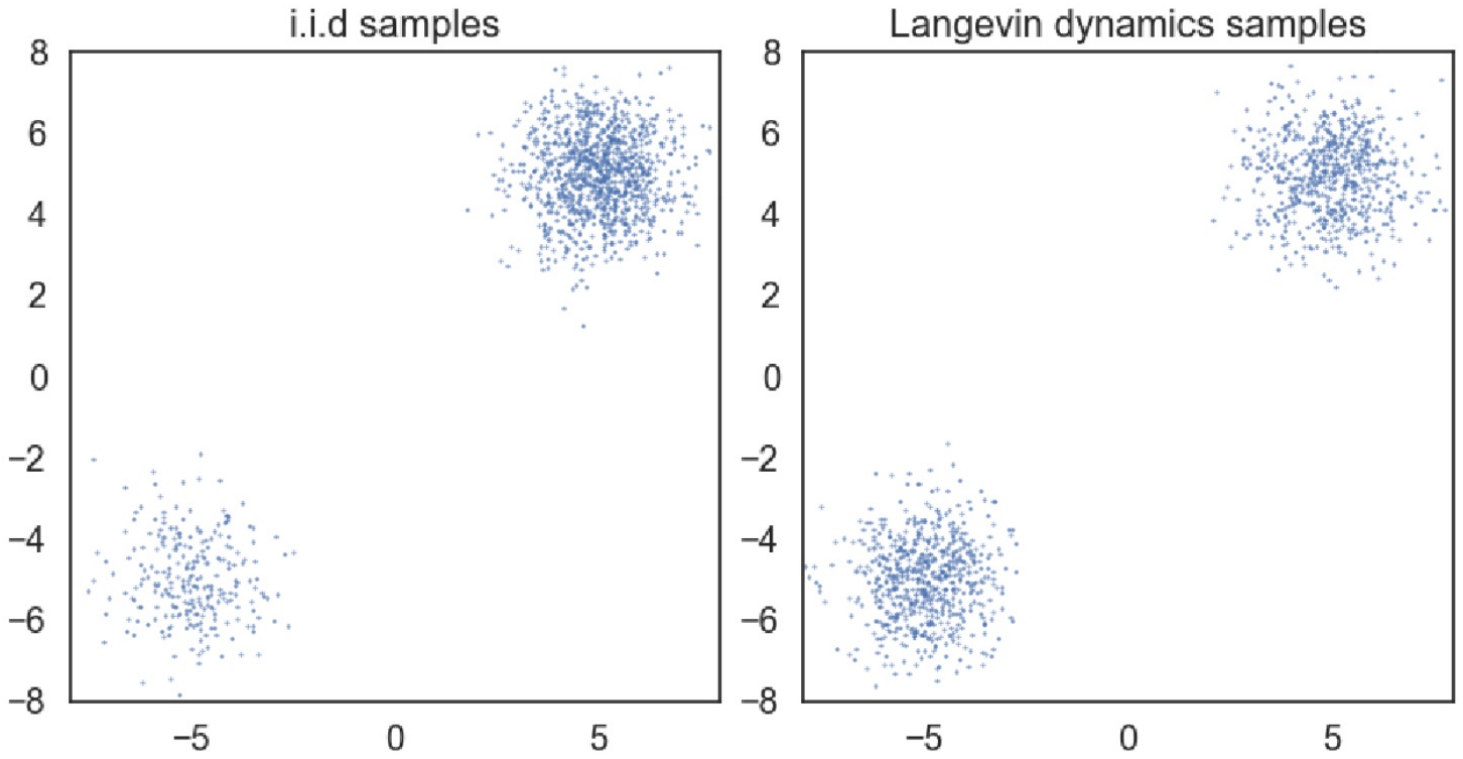
- Data score function:

$$\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) = \begin{cases} \nabla_{\mathbf{x}}[\log \pi + \log p_1(\mathbf{x})], & \mathbf{x} \in \mathcal{A} \\ \nabla_{\mathbf{x}}[\log(1-\pi) + \log p_2(\mathbf{x})], & \mathbf{x} \in \mathcal{B} \end{cases}$$

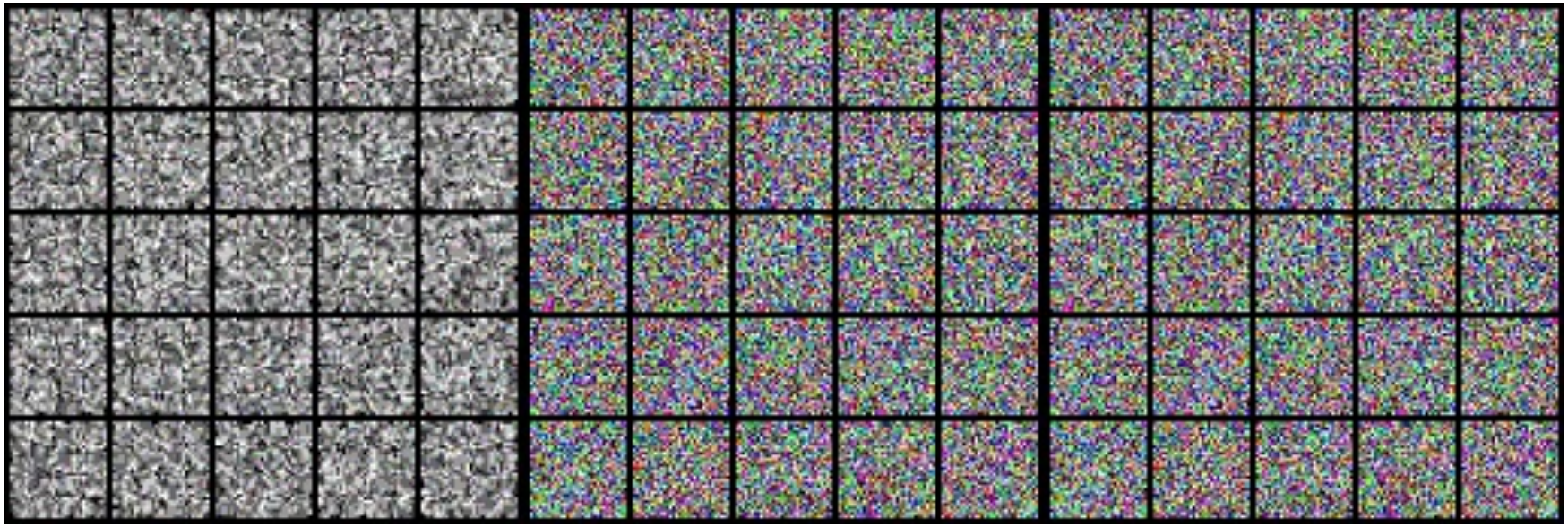
$$= \begin{cases} \nabla_{\mathbf{x}} \log p_1(\mathbf{x}), & \mathbf{x} \in \mathcal{A} \\ \nabla_{\mathbf{x}} \log p_2(\mathbf{x}), & \mathbf{x} \in \mathcal{B} \end{cases}$$

- The score function has no dependence on the mode weighting π at all!
- Langevin sampling will not reflect π

Pitfall 3: slow mixing of Langevin dynamics between data modes

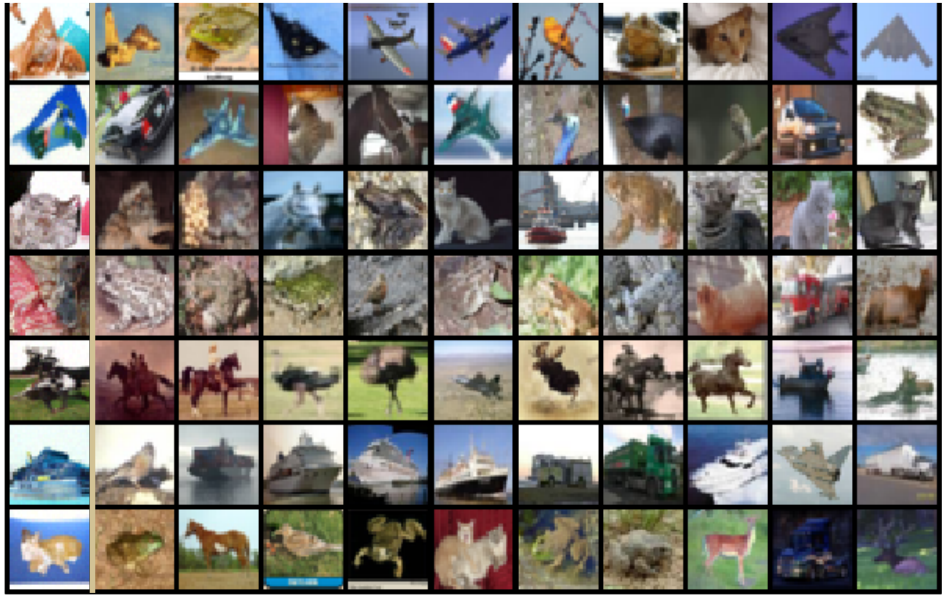


After fixing these pitfalls (next lecture)



Song, Yang, and Stefano Ermon. "Generative Modeling by Estimating Gradients of the Data Distribution." NeurIPS 2019.

Experiments: nearest neighbors



Experiments: inpainting