

Score-Based Models

LECTURE 13

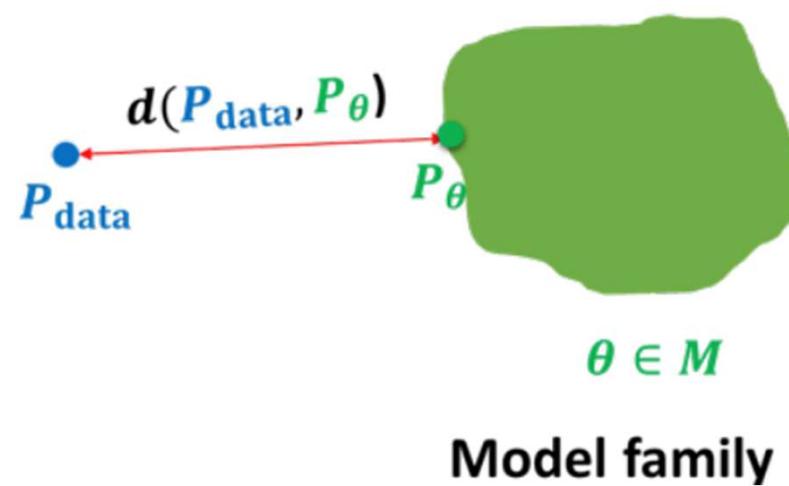
CS236: Deep Generative Models

Stanford University

Summary



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



Stanford University

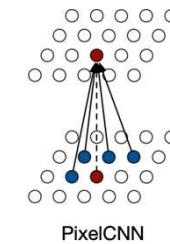
How to represent probability distributions?

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

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

- Autoregressive models

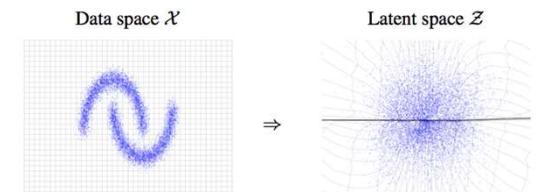
$$p_{\theta}(\mathbf{x}) = \prod_{i=1}^d p_{\theta}(\mathbf{x}_i \mid \mathbf{x}_{<i})$$



PixelCNN

- Flow models

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



Stanford University

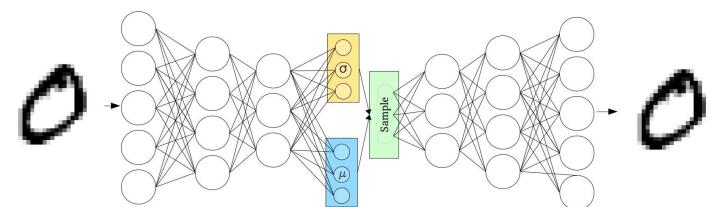
How to represent probability distributions?

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

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

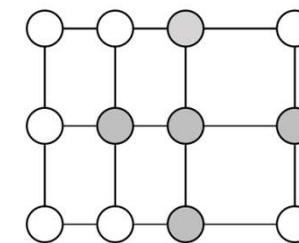
- Variational autoencoders

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



- Energy-based models

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



Stanford University

How to represent probability distributions?

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

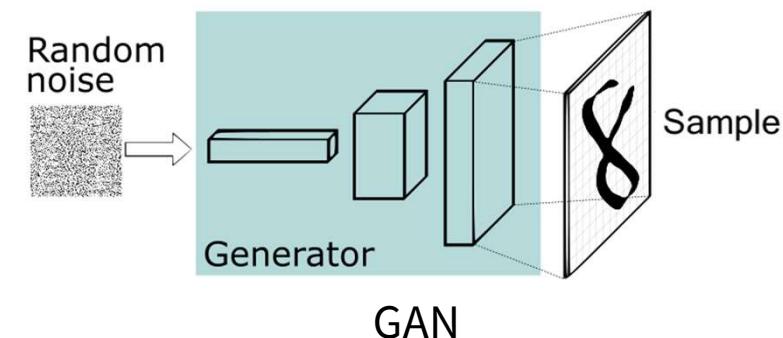
$$p(\mathbf{x})$$

- 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)

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



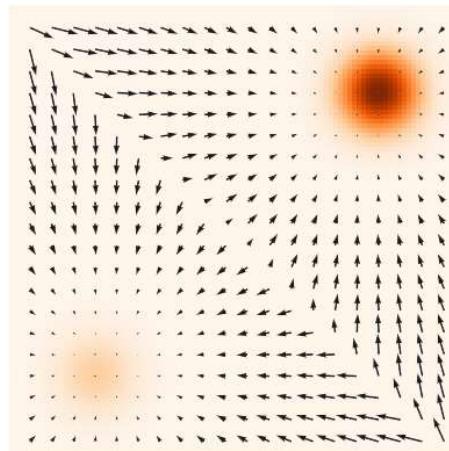
How to represent probability distributions?

- 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

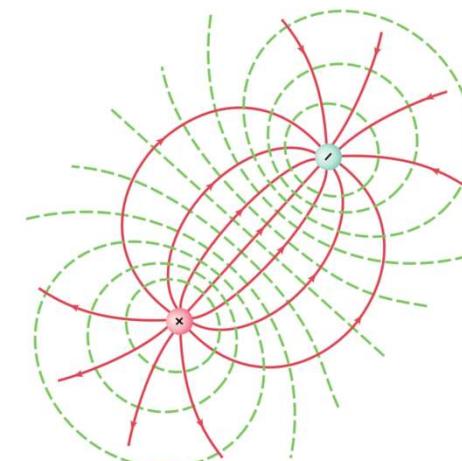
How to represent probability distributions?

- 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)

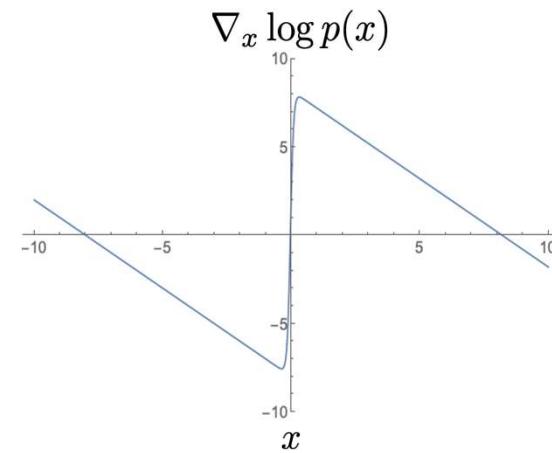
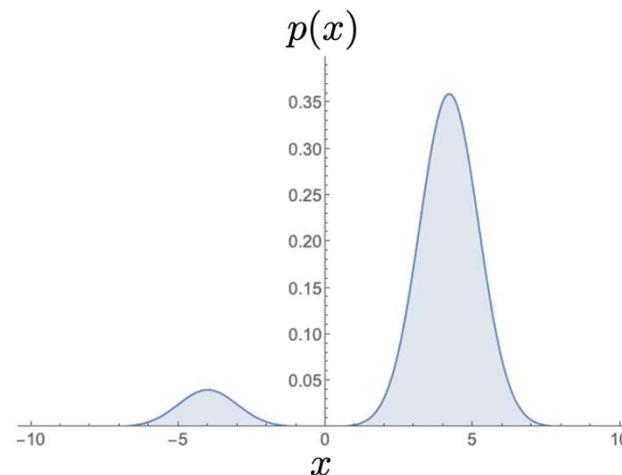


(Electrical potentials and fields)
Stanford University

How to represent probability distributions?

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

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

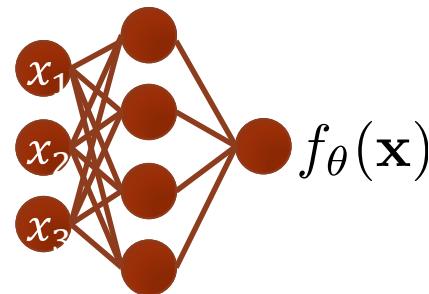


Recap on energy-based models

- 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

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

- Requires iterative sampling during training

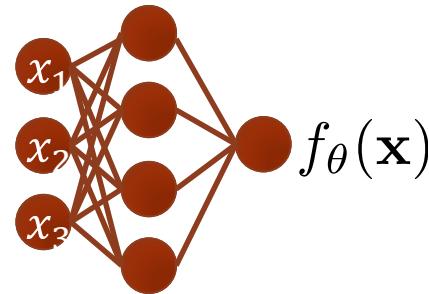
Stanford University

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

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

Score matching for training EBMs

- Score function of EBMs

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

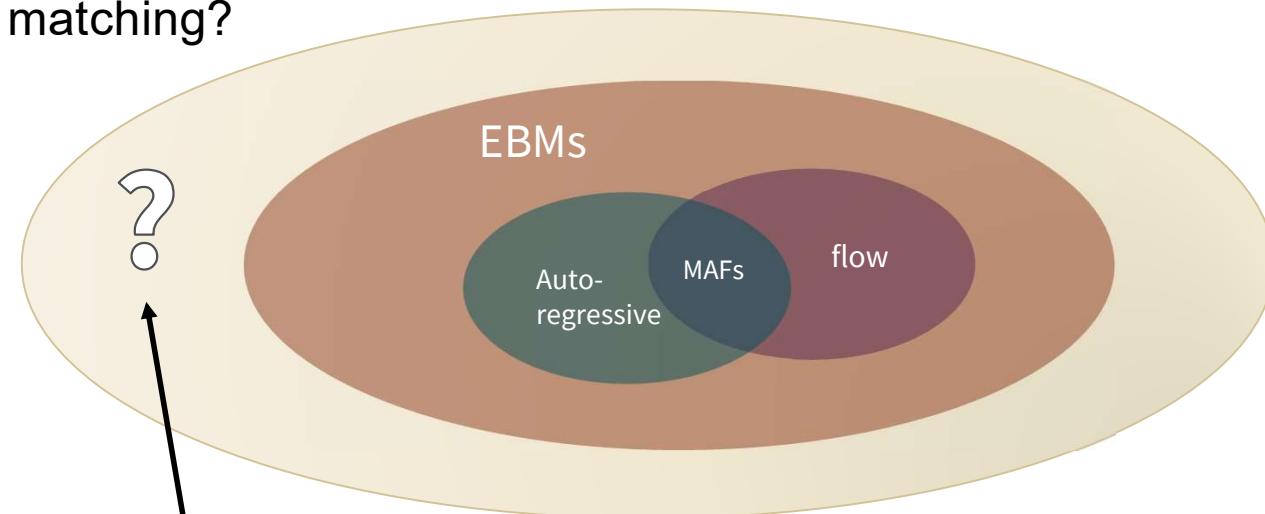
- Score matching for EBMs:

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

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

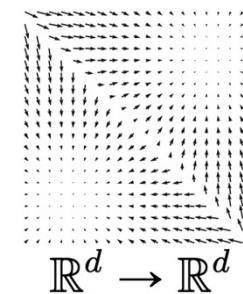
Score-based models

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



- Score-based model

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



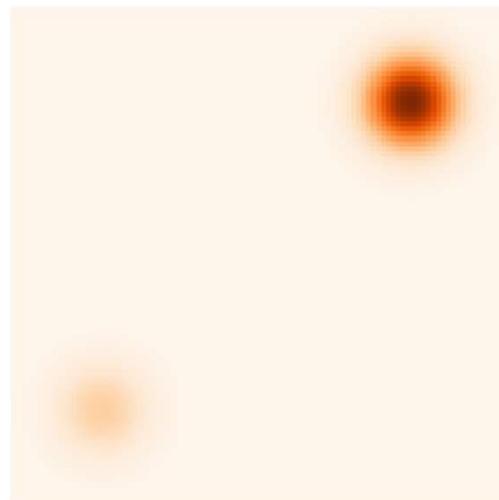
Directly model
the vector field
of gradients!

Stanford University

Score estimation by training score-based models

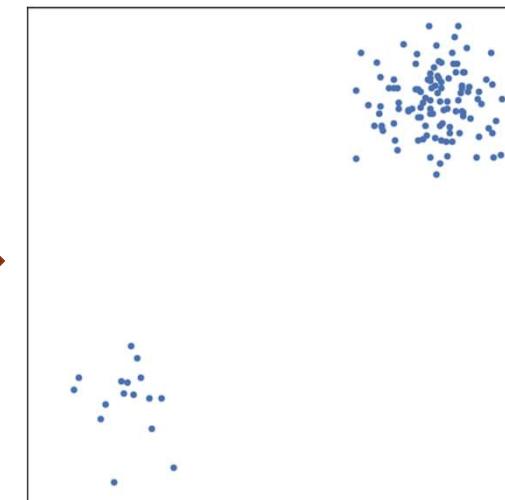
Probability density

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



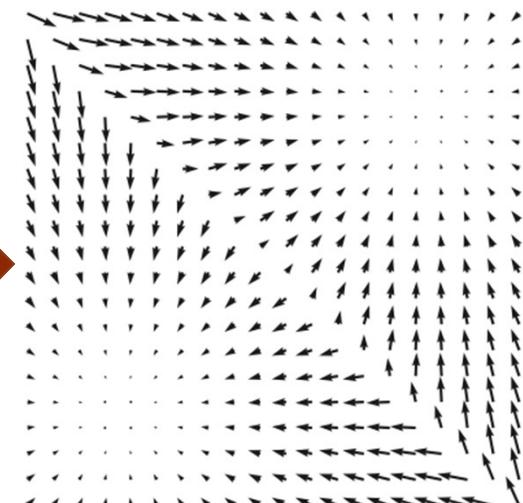
i.i.d. samples

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



Score function

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



Stanford University

Score estimation by training score-based models

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



Score estimation by training score-based models

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

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

(Fisher divergence)

- **Score matching:**

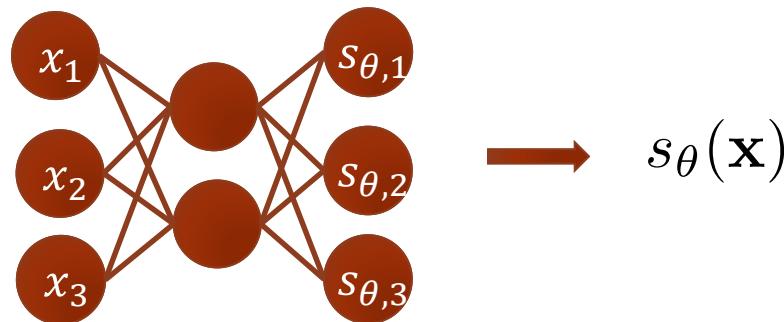
$$E_{\mathbf{x} \sim p_{\text{data}}} \left[\frac{1}{2} \|\mathbf{s}_{\theta}(\mathbf{x})\|_2^2 + \text{tr} \left(\underbrace{\nabla_{\mathbf{x}} \mathbf{s}_{\theta}(\mathbf{x})}_{\text{Jacobian of } \mathbf{s}_{\theta}(\mathbf{x})} \right) \right]$$

- **Requirements:**

- The score model must be efficient to evaluate.
- Do we need the score model to be a proper score function (i.e., gradient of a scalar “energy” function)?

Score matching is not scalable

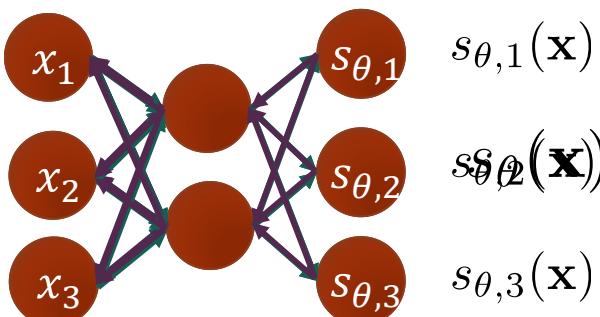
- Deep neural networks as more expressive score models



Score Matching
is not Scalable!

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

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



$O(d)$ Backprops!

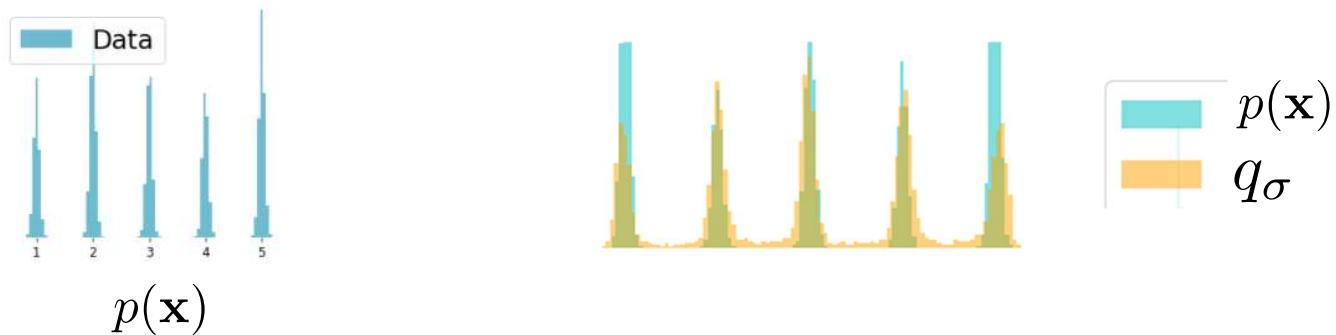
$$\nabla_{\mathbf{x}} s_{\theta}(\mathbf{x}) = \begin{pmatrix} \frac{\partial s_{\theta,1}(\mathbf{x})}{\partial x_1} & \frac{\partial s_{\theta,1}(\mathbf{x})}{\partial x_2} & \frac{\partial s_{\theta,1}(\mathbf{x})}{\partial x_3} \\ \frac{\partial s_{\theta,2}(\mathbf{x})}{\partial x_1} & \frac{\partial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

Stanford University

Denoising Score Matching (Vincent, 2011)

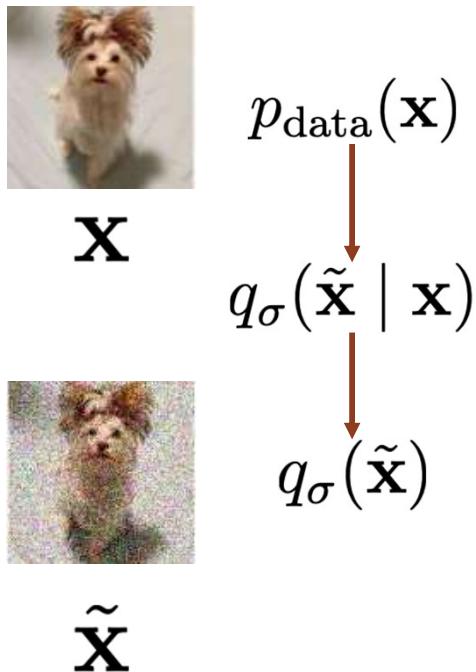
- Consider the perturbed distribution

$$q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = \mathcal{N}(\mathbf{x}; \sigma^2 I) \quad q_\sigma(\tilde{\mathbf{x}}) = \int p(\mathbf{x}) q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) d\mathbf{x}$$



- Score estimation for $\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})$ is easier
- If the noise level is small, this is a good approximation $q_\sigma(\tilde{\mathbf{x}}) \approx p(\tilde{\mathbf{x}})$

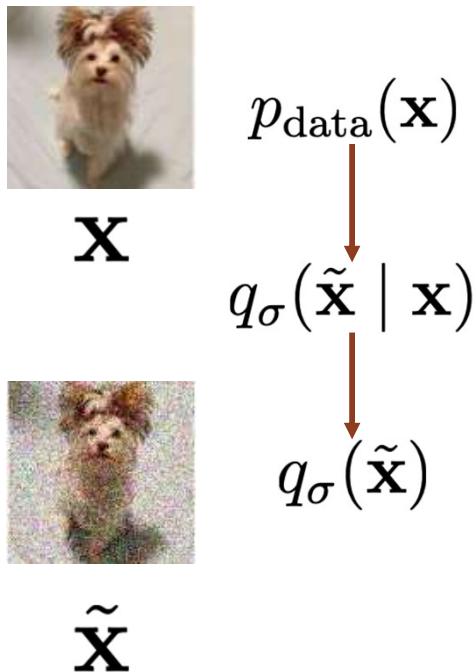
Denoising score matching



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

$$\begin{aligned} & \frac{1}{2} E_{\tilde{\mathbf{x}} \sim q_\sigma} [\|\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}) - \mathbf{s}_\theta(\tilde{\mathbf{x}})\|_2^2] \\ &= \frac{1}{2} \int q_\sigma(\tilde{\mathbf{x}}) \|\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}) - \mathbf{s}_\theta(\tilde{\mathbf{x}})\|_2^2 d\tilde{\mathbf{x}} \\ &= \frac{1}{2} \int q_\sigma(\tilde{\mathbf{x}}) \|\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})\|_2^2 d\tilde{\mathbf{x}} + \frac{1}{2} \int q_\sigma(\tilde{\mathbf{x}}) \|\mathbf{s}_\theta(\tilde{\mathbf{x}})\|_2^2 d\tilde{\mathbf{x}} \\ &\quad - \int q_\sigma(\tilde{\mathbf{x}}) \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})^\top \mathbf{s}_\theta(\tilde{\mathbf{x}}) d\tilde{\mathbf{x}} \\ &= \text{const.} + \frac{1}{2} E_{\tilde{\mathbf{x}} \sim q_\sigma} [\|\mathbf{s}_\theta(\tilde{\mathbf{x}})\|_2^2] - \int q_\sigma(\tilde{\mathbf{x}}) \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})^\top \mathbf{s}_\theta(\tilde{\mathbf{x}}) d\tilde{\mathbf{x}} \end{aligned}$$

Denoising score matching



- Denoising score matching

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

Stanford University

Denoising score matching



X



~X

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

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

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

- Denoising score matching

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

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

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

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

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

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

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

Stanford University

Denoising score matching

- Estimate the score of a noise-perturbed distribution

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

- $\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} | \mathbf{x})$ is easy to compute
 - $q_\sigma(\tilde{\mathbf{x}} | \mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}} | \mathbf{x}, \sigma^2 \mathbf{I})$
 - $\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} | \mathbf{x}) = -\frac{\tilde{\mathbf{x}} - \mathbf{x}}{\sigma^2}$
- **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 $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \sim p_{\text{data}}(\mathbf{x})$
- Sample a minibatch of perturbed datapoints $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_n\} \sim q_\sigma(\tilde{\mathbf{x}})$
$$\tilde{\mathbf{x}}_i \sim q_\sigma(\tilde{\mathbf{x}}_i \mid \mathbf{x}_i)$$
- Estimate the denoising score matching loss with empirical means

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

- If Gaussian perturbation

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

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

Denoising Score Matching (Vincent, 2011)

- Consider the perturbed distribution

$$q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = \mathcal{N}(\mathbf{x}; \sigma^2 I) \quad q_\sigma(\tilde{\mathbf{x}}) = \int p(\mathbf{x}) q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) d\mathbf{x}$$

- Score estimation for $\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})$ is easier

$$\text{Score matching} \quad \frac{1}{2} \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})} [\|\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}) - s_\theta(\tilde{\mathbf{x}})\|_2^2]$$

$$\text{denoising} \quad = \frac{1}{2} \mathbb{E}_{p(\mathbf{x})} \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})} [\left\| \frac{1}{\sigma^2} (\mathbf{x} - \tilde{\mathbf{x}}) \right\|_2^2 - s_\theta(\tilde{\mathbf{x}})] + \text{const.}$$



\mathbf{x}

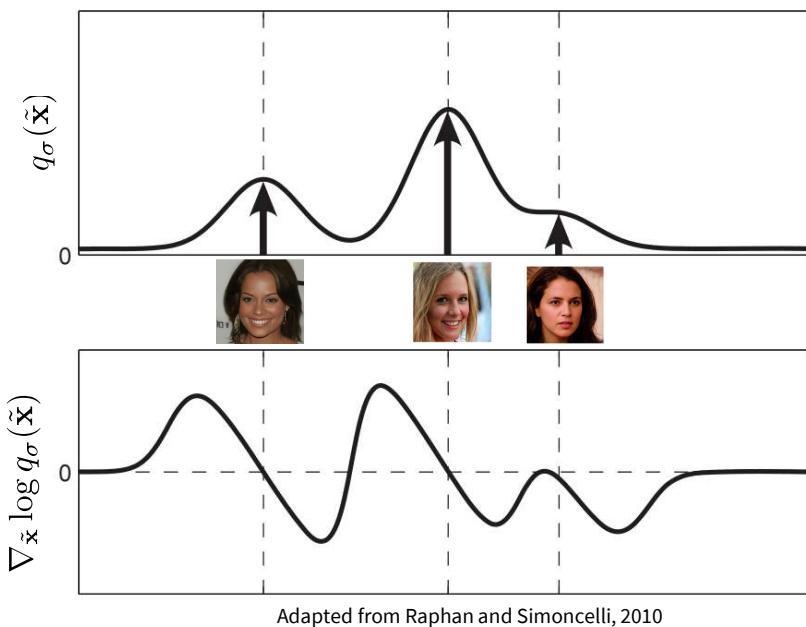
$\tilde{\mathbf{x}} = \mathbf{x} + \text{noise}$

$s_\theta(\tilde{\mathbf{x}})$ tries to estimate the noise that was added to produce $\tilde{\mathbf{x}}$

Tweedie's formula

- Score estimation for $\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})$ is equivalent to denoising

$$\frac{1}{2} \mathbb{E}_{p(\mathbf{x})} \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})} [\| \frac{1}{\sigma^2} (\mathbf{x} - \tilde{\mathbf{x}}) - s_\theta(\tilde{\mathbf{x}}) \|_2^2] + \text{const.}$$



Tweedie's formula

Optimal denoising strategy is to follow the gradient (score):

$$\tilde{\mathbf{x}} + \sigma^2 \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})$$

Stanford University

Tweedie's formula and denoising score matching

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

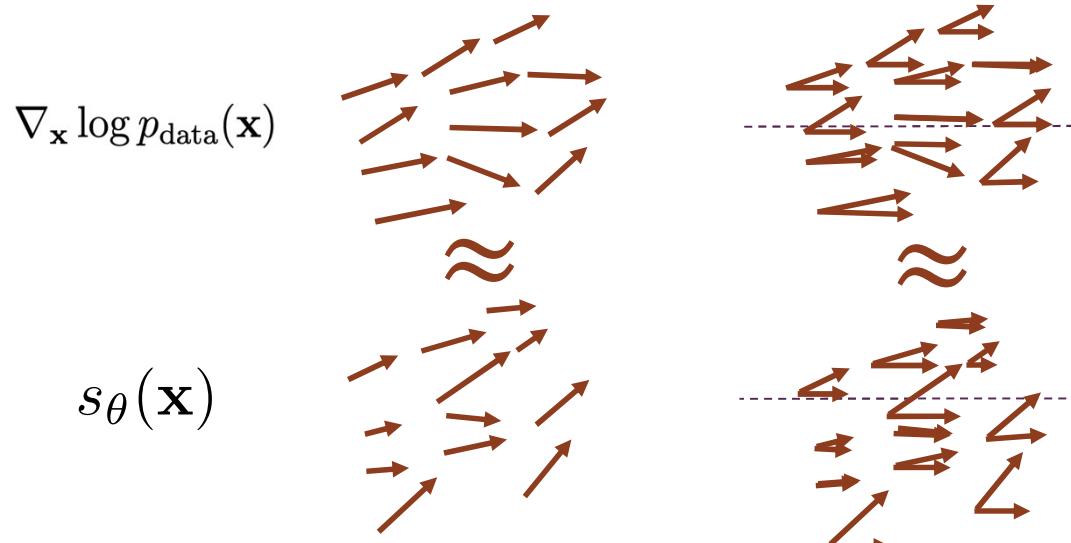
$$p(\mathbf{x} \mid \tilde{\mathbf{x}}) \propto p(\mathbf{x}) q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})$$

- Recall that
- Tweedie's formula:

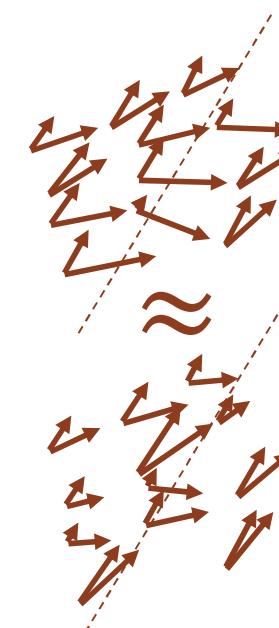
$$\begin{aligned} E_{\mathbf{x} \sim p(\mathbf{x} \mid \tilde{\mathbf{x}})}[\mathbf{x}] &= \tilde{\mathbf{x}} + \sigma^2 \nabla_{\mathbf{x}} \log q_\sigma(\tilde{\mathbf{x}}) \\ &\approx \tilde{\mathbf{x}} + \sigma^2 \mathbf{s}_\theta(\tilde{\mathbf{x}}) \end{aligned}$$

Sliced score matching

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



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



Stanford University

Sliced score matching

- **Objective:** Sliced Fisher Divergence

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

- **Integration by parts**

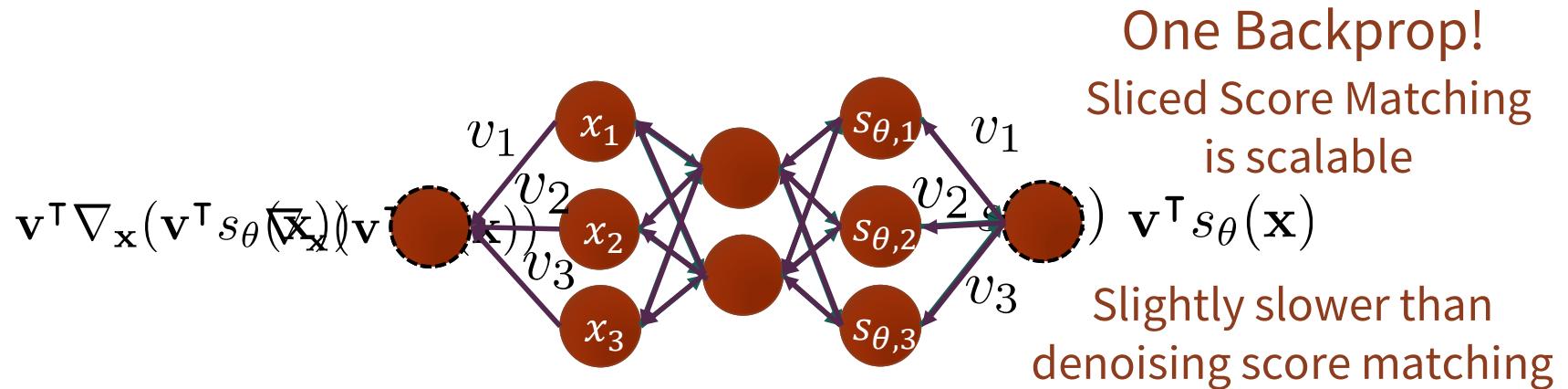
$$E_{\mathbf{v} \sim p_{\mathbf{v}}} E_{\mathbf{x} \sim p_{\text{data}}} \left[\mathbf{v}^T \nabla_{\mathbf{x}} \mathbf{s}_{\theta}(\mathbf{x}) \mathbf{v} + \frac{1}{2} (\mathbf{v}^T \mathbf{s}_{\theta}(\mathbf{x}))^2 \right]$$

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

Sliced Score Matching

Computing Jacobian-vector products is scalable

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



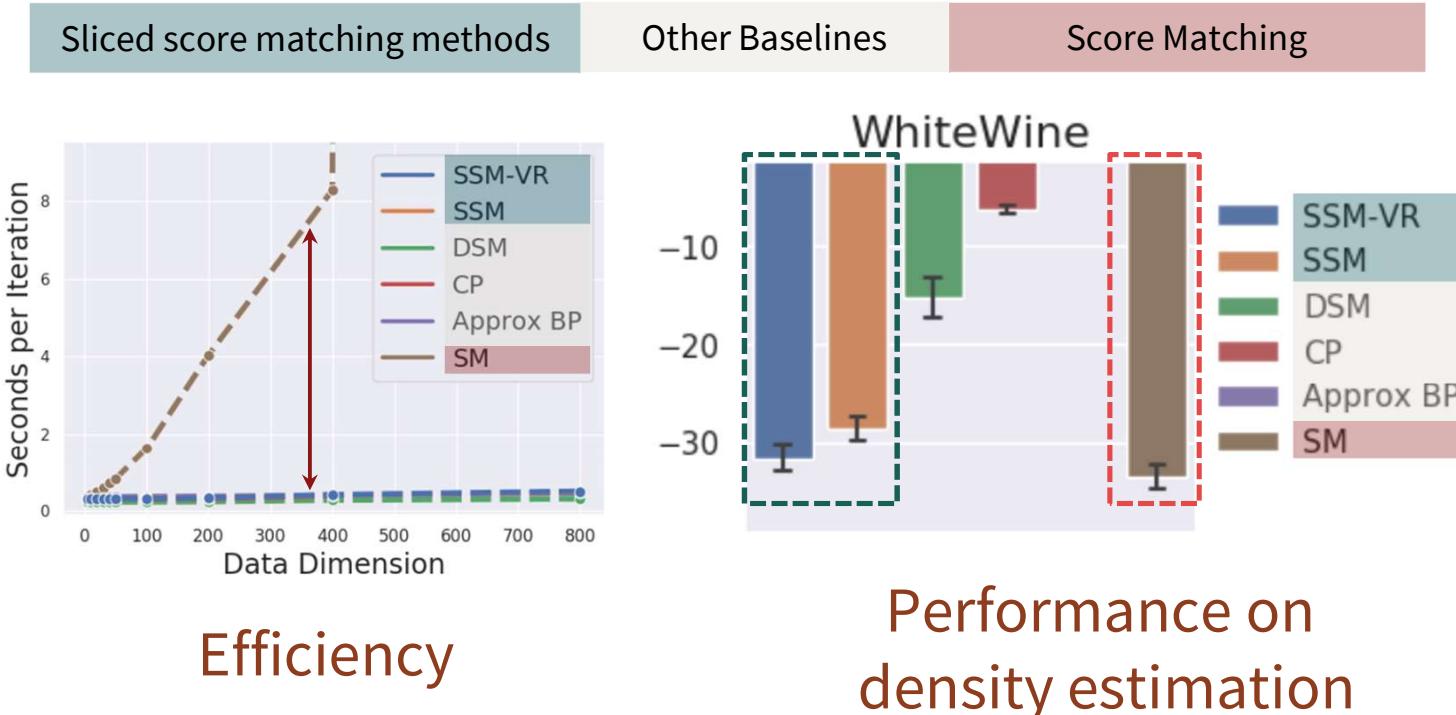
Sliced score matching

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

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

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

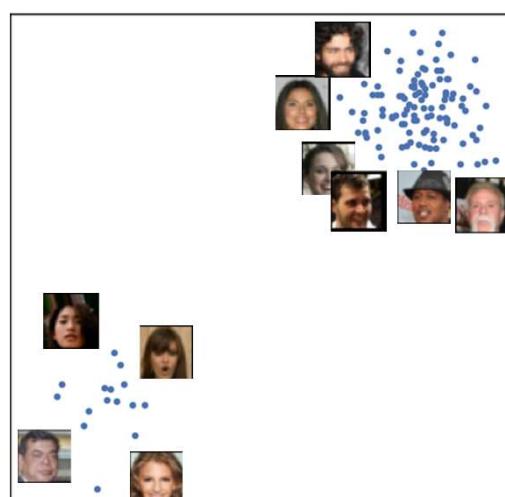
Experimental results for sliced score matching



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

Stanford University

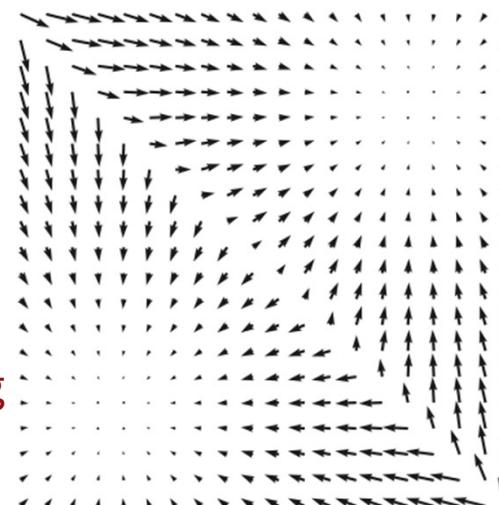
Score-based generative modeling



Data samples

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

Score
Matching



Scores

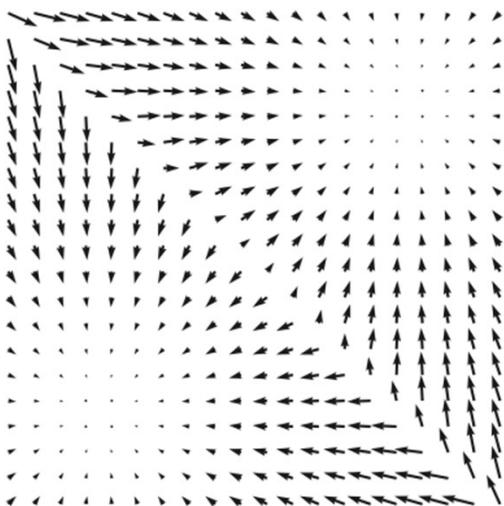
$$s_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$$



New samples

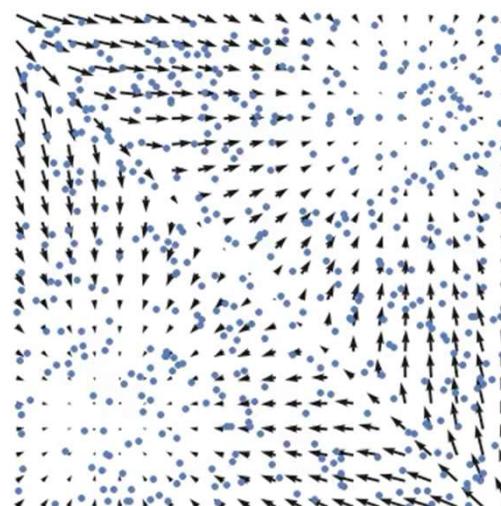
Stanford University

From scores to samples: Langevin MCMC



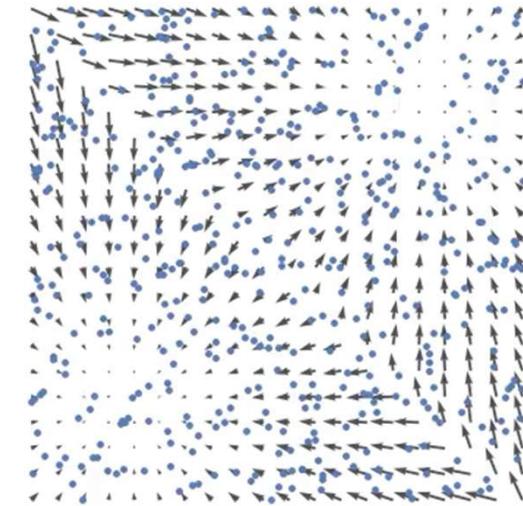
Scores

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



Follow the scores

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



Follow noisy scores:
Langevin MCMC

$$\begin{aligned}\mathbf{z}_t &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \tilde{\mathbf{x}}_{t+1} &\leftarrow \tilde{\mathbf{x}}_t + \frac{\epsilon}{2} \mathbf{s}_\theta(\tilde{\mathbf{x}}_t) + \sqrt{\epsilon} \mathbf{z}_t\end{aligned}$$

Stanford University

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 $t \leftarrow 1, 2, \dots, T$

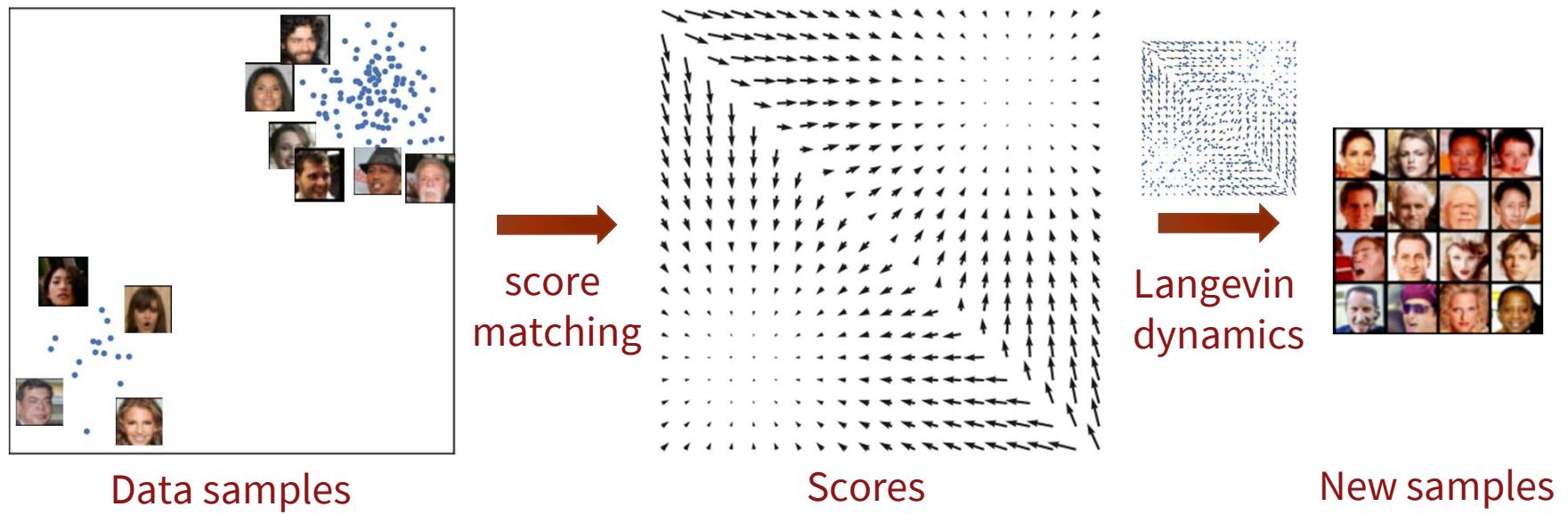
$$\mathbf{z}^t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

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

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

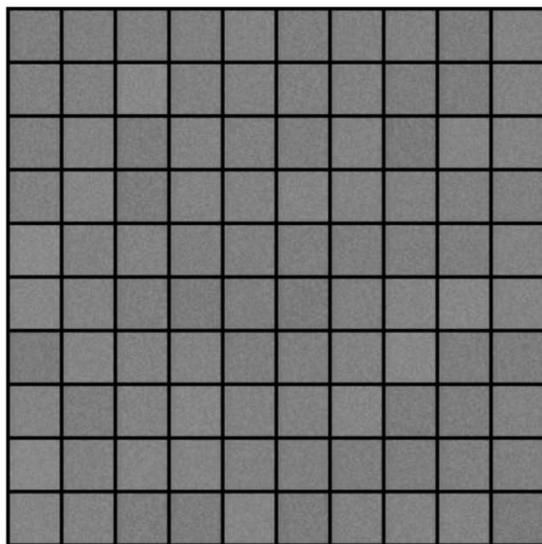
$$s_{\theta}(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p(\mathbf{x})$$

Score-based generative modeling

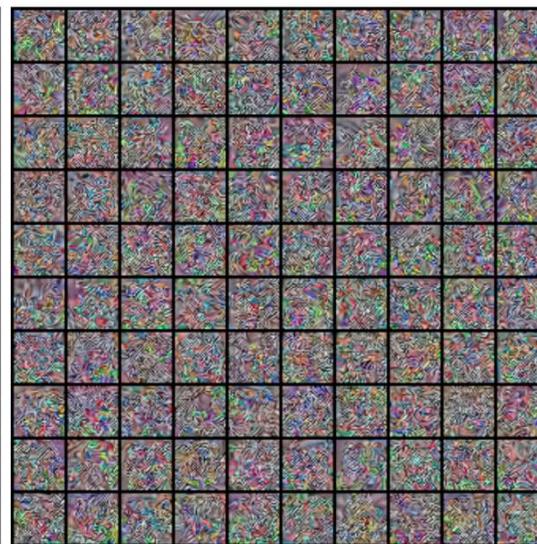


Stanford University

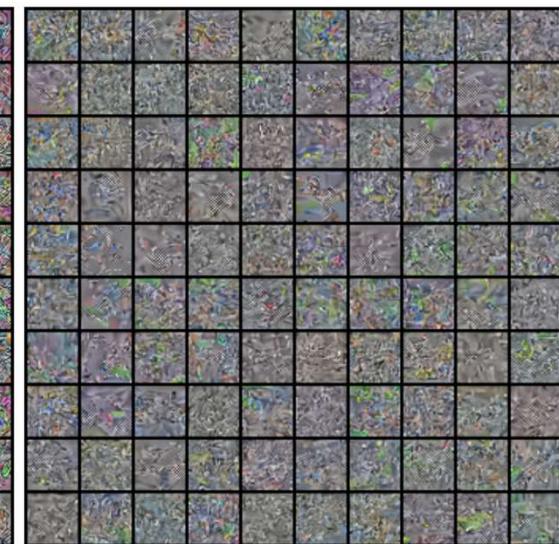
Score-based generative modeling: results



(a) MNIST



(b) CelebA

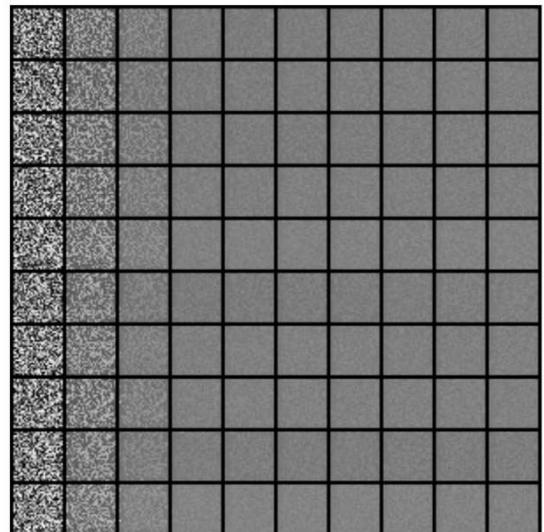


(c) CIFAR-10

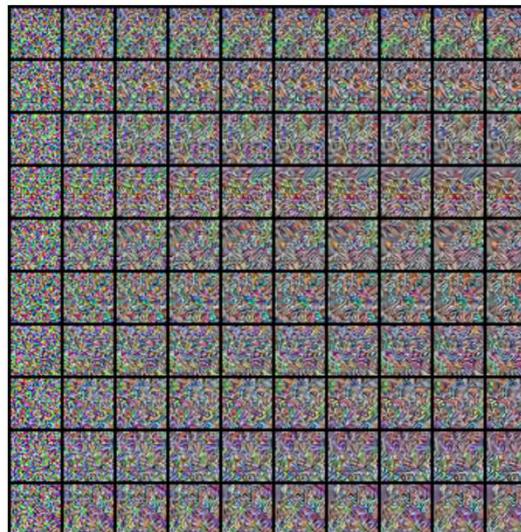
Final samples

Stanford University

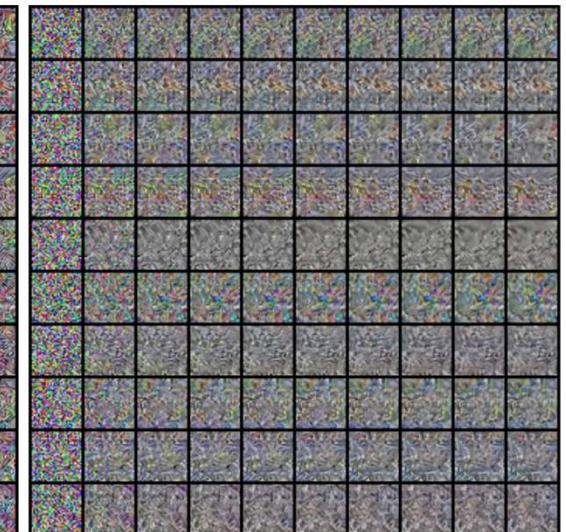
Score-based generative modeling: results



(a) MNIST



(b) CelebA



(c) CIFAR-10

Langevin sampling process

Stanford University

Pitfall 1: manifold hypothesis

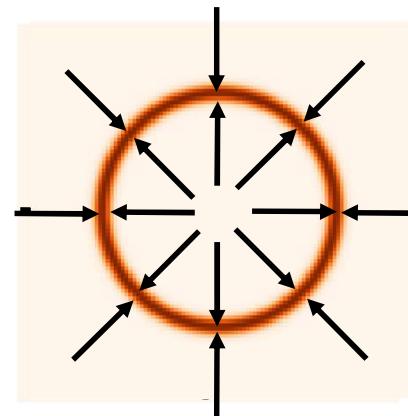
- Manifold hypothesis.



- Data score is undefined.

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

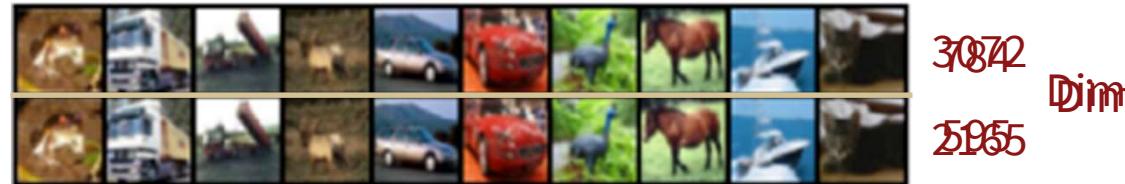
A large red X is drawn over this equation.



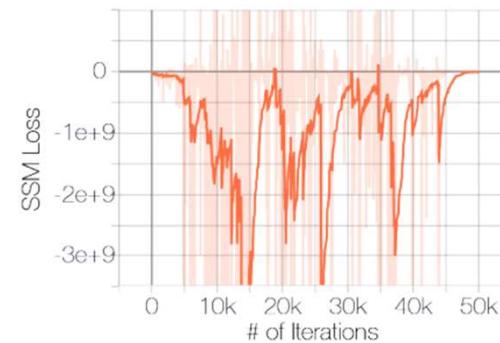
Stanford University

Pitfall 1: manifold hypothesis

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

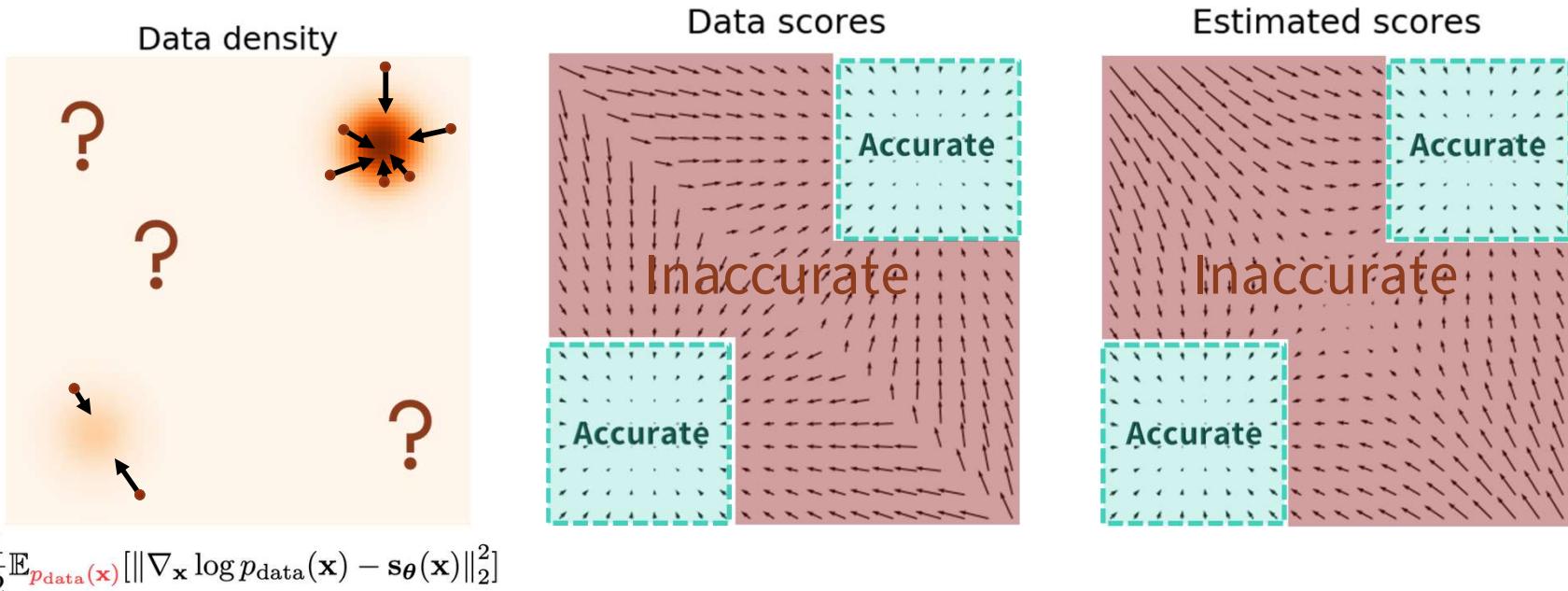


- Score estimation on CIFAR-10.



Stanford University

Challenge in low data density regions



**Langevin MCMC will have trouble
exploring low density regions**

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

Stanford University

Pitfall 3: slow mixing of Langevin dynamics between data modes

- Suppose the data distribution has two modes with disjoint supports:

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

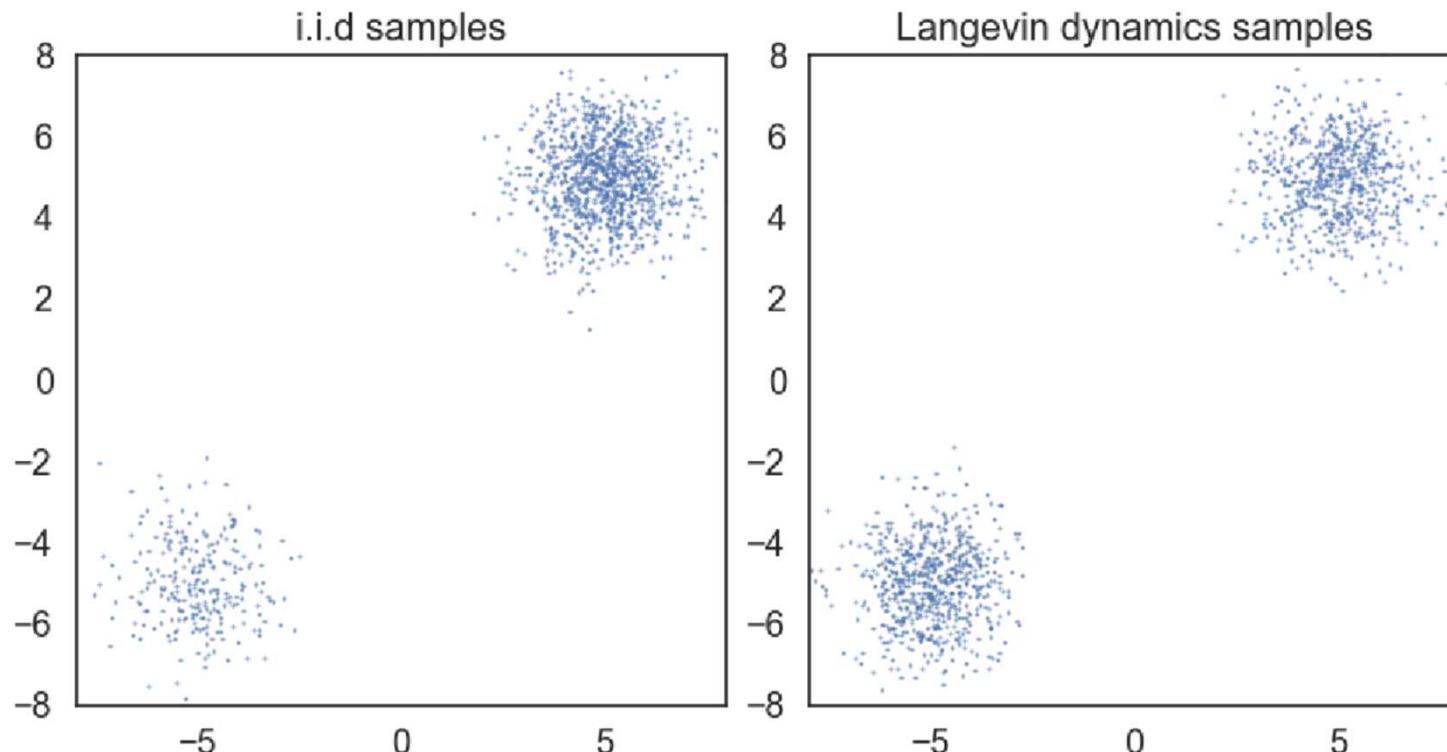
$$\mathcal{A} \cap \mathcal{B} = \emptyset \quad p_{\text{data}}(\mathbf{x}) = \begin{cases} \pi p_1(\mathbf{x}), & \mathbf{x} \in \mathcal{A} \\ (1 - \pi)p_2(\mathbf{x}), & \mathbf{x} \in \mathcal{B} \end{cases}$$

- Data score function:

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

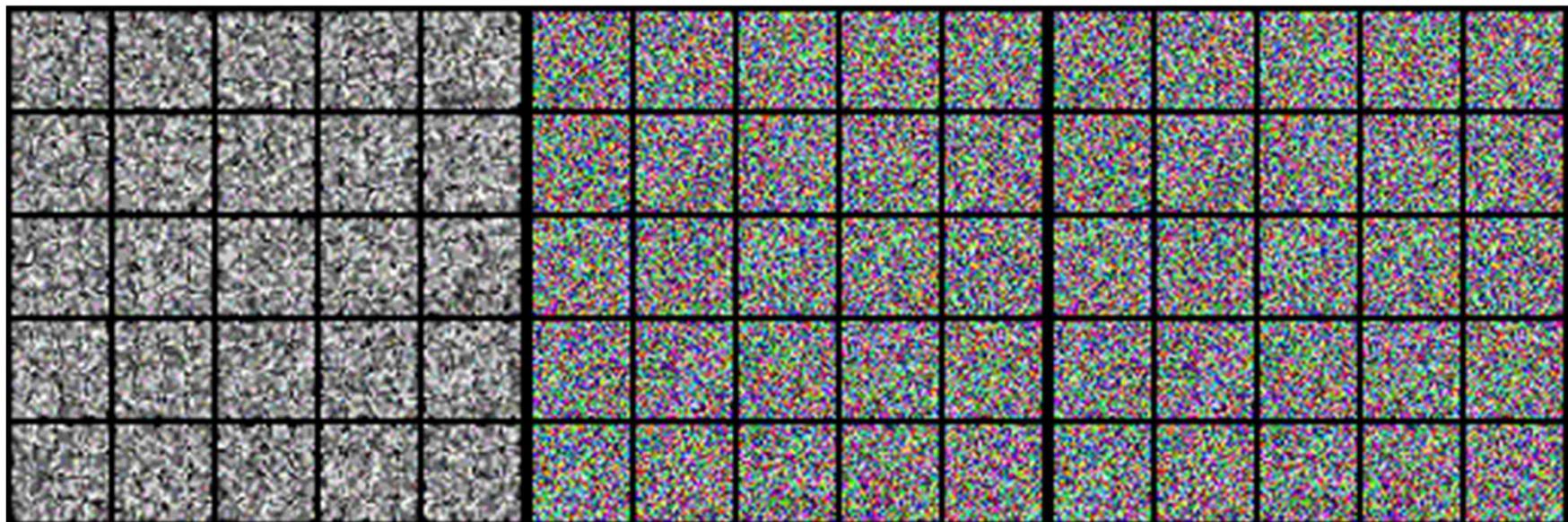
- The score function has no dependence on the mode weighting π at all!
- Langevin sampling will not reflect π

Pitfall 3: slow mixing of Langevin dynamics between data modes



Stanford University

After fixing these pitfalls (next lecture)



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

Stanford University