Deep Generative Models

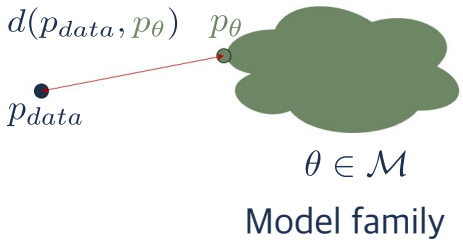
12. Score-Based Models

- 국가수리과학연구소 산업수학혁신센터 김민중

Summary

- **Representation**: how do we model the joint distribution of many random variables?
  - Need compact representation
- **Learning**: what is the right way to compare probability distributions?

$\mathbf{x}_i \sim p_{data}$
$i = 1, 2, ..., N$



- **Inference**: how do we invert the generation process

Representation

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

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

  - Autoregressive Models

$$p_\theta(\boldsymbol{x}) = \prod_{i=1}^{d} p_\theta(x_i|\boldsymbol{x}_{<i})$$

  - Variational Autoencoders

$$p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x}, \boldsymbol{z})d\boldsymbol{z} = \int p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$$

Representation

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

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

  - Flow Models

$$p_X(\boldsymbol{x};\theta) = p_Z\left(\boldsymbol{f}_\theta^{-1}(\boldsymbol{x})\right)\left|\det\left(\frac{\partial \boldsymbol{f}_\theta^{-1}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)\right|$$

  - Energy-based models

$$p_\theta(\boldsymbol{x}) = \frac{1}{Z(\theta)}\exp\big(f_\theta(\boldsymbol{x})\big)$$

Representation

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

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

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

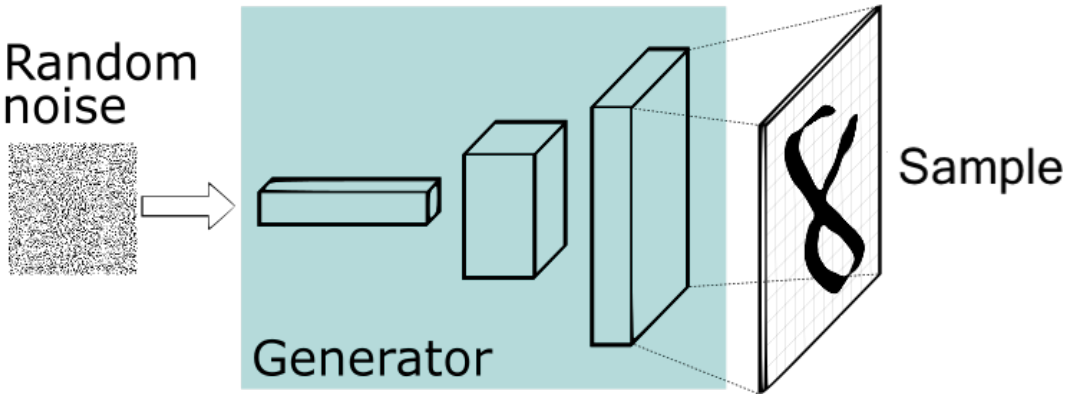
- Cons
  - Special architectures or surrogate losses to deal with intractable partition functions

Implicit generative model

- Generative adversarial networks (GANs)

$$z \sim p(z)$$
$$x = G_\theta(z)$$

- A two-player minimax game between a generator and a discriminator(two sample test)



GAN

Generative adversarial networks

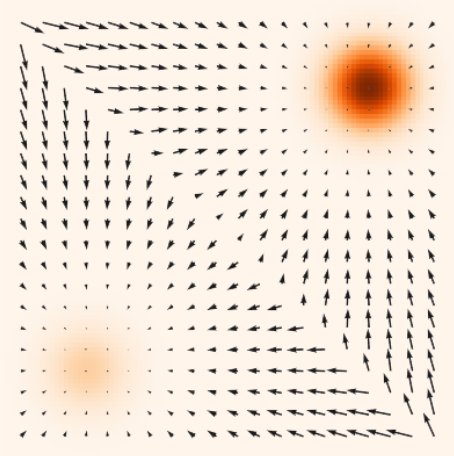
- Pros
  - Samples typically have better quality

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

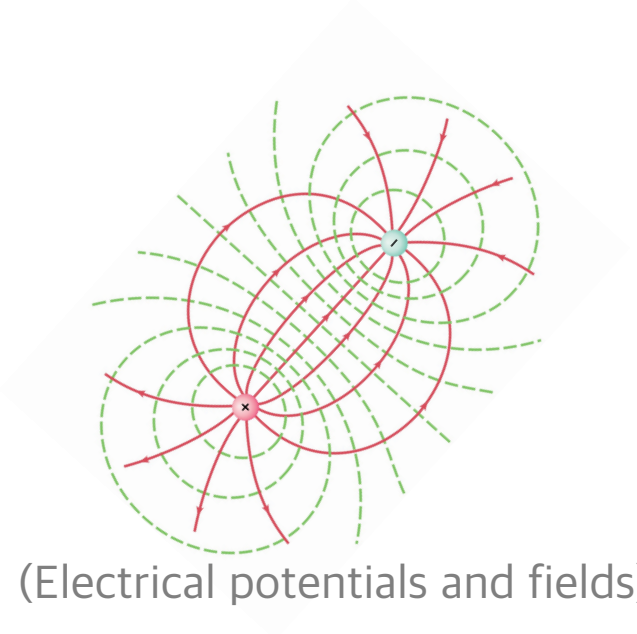
Score functions

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

$$\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})$$



(pdf and score)

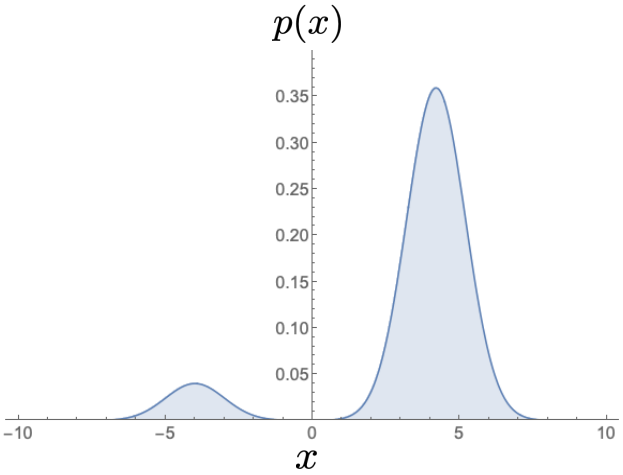


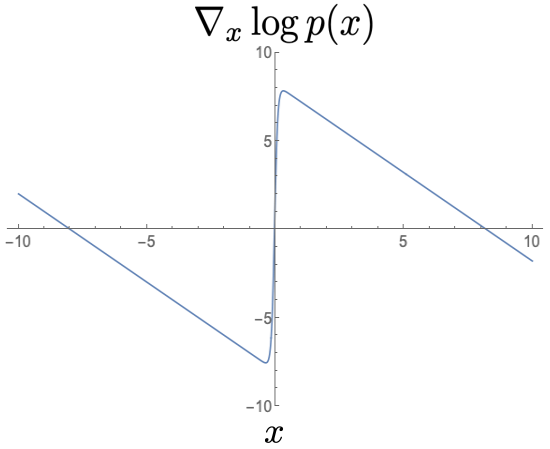
(Electrical potentials and fields)

Score functions

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

$$\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})$$

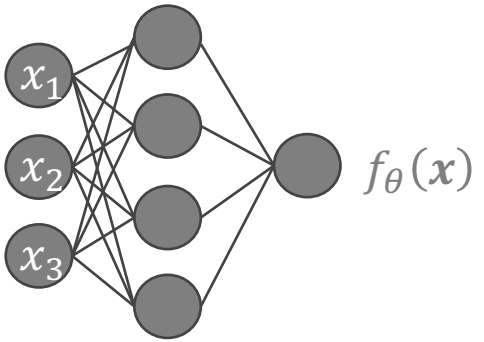




Recap. of Energy-based model

- Energy-based models: $\frac{1}{Z(\theta)}\exp\left(f_\theta(\boldsymbol{x})\right)$
  - $Z(\theta)$ is intractable, so no access to likelihood
  - Comparing the probability of two points is easy

$$\frac{p_\theta(\boldsymbol{x})}{p_\theta(\boldsymbol{x}')} = \exp\left(f_\theta(\boldsymbol{x}) - f_\theta(\boldsymbol{x}')\right)$$



- Maximum likelihood training:

$$\max_\theta [f_\theta(\boldsymbol{x}_{train}) - \log Z(\theta)]$$

  - Contrastive divergence:

$$\nabla_\theta f_\theta(\boldsymbol{x}_{train}) - \nabla_\theta \log Z(\theta) \approx \nabla_\theta f_\theta(\boldsymbol{x}_{train}) - \nabla_\theta f_\theta\left(\boldsymbol{x}_{sample}\right)$$

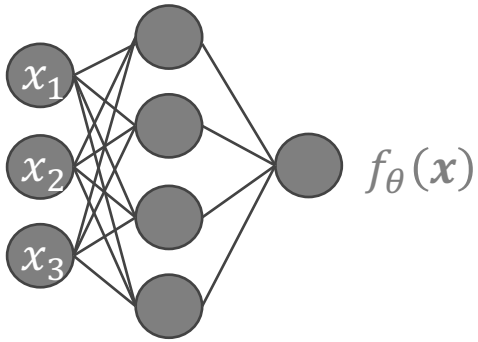
  - where $\boldsymbol{x}_{sample} \sim p_\theta(\boldsymbol{x}) = \frac{\exp\left(f_\theta(\boldsymbol{x})\right)}{Z(\theta)}$

- Requires iterative sampling during training

Recap. of Energy-based model

- Energy-based models

$$p_\theta(\boldsymbol{x}) = \frac{1}{Z(\theta)} \exp\big(f_\theta(\boldsymbol{x})\big)$$
$$f_\theta(\boldsymbol{x}) \in \mathbb{R}$$



- Minimizing Fisher divergence

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

  - Score matching

$$\frac{1}{2} E_{\boldsymbol{x}\sim p_{data}}[\, \|\nabla_{\boldsymbol{x}} \log p_{data}(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x})\|_2^2]$$
$$= E_{\boldsymbol{x}\sim p_{data}}\left[\frac{1}{2}\|\nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x})\|_2^2 + tr(\nabla_{\boldsymbol{x}}^2 \log p_\theta(\boldsymbol{x}))\right] + const.$$

Score matching for training EBMs

- Score function of EBMs

$$\nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} f_\theta(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log Z(\theta) = \nabla_{\boldsymbol{x}} f_\theta(\boldsymbol{x})$$

- Score matching for EBMs

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

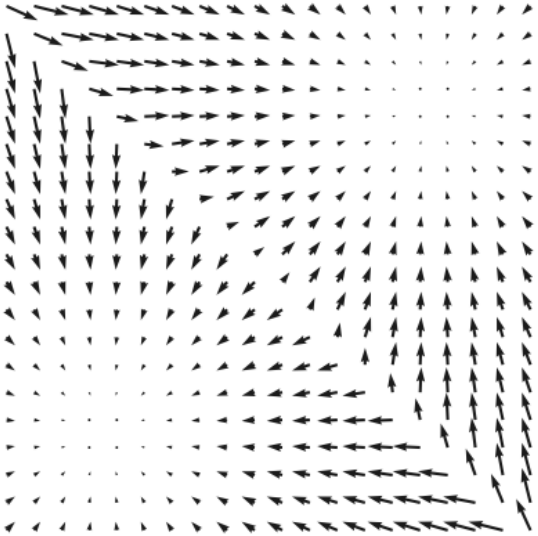
$$= E_{\boldsymbol{x}\sim p_{data}}\left[\frac{1}{2}\|\nabla_{\boldsymbol{x}} f_\theta(\boldsymbol{x})\|_2^2 + tr\left(\nabla_{\boldsymbol{x}}^2 f_\theta(\boldsymbol{x})\right)\right]$$

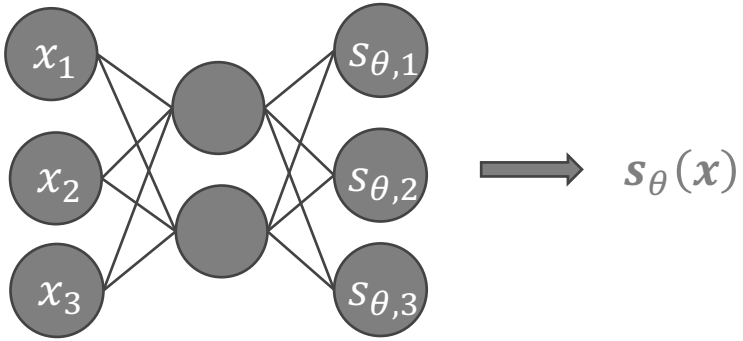
Score-based models

- Directly model the vector field of gradients

$$\boldsymbol{s}_\theta(\boldsymbol{x}): \mathbb{R}^{\mathrm{d}} \rightarrow \mathbb{R}^d$$

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





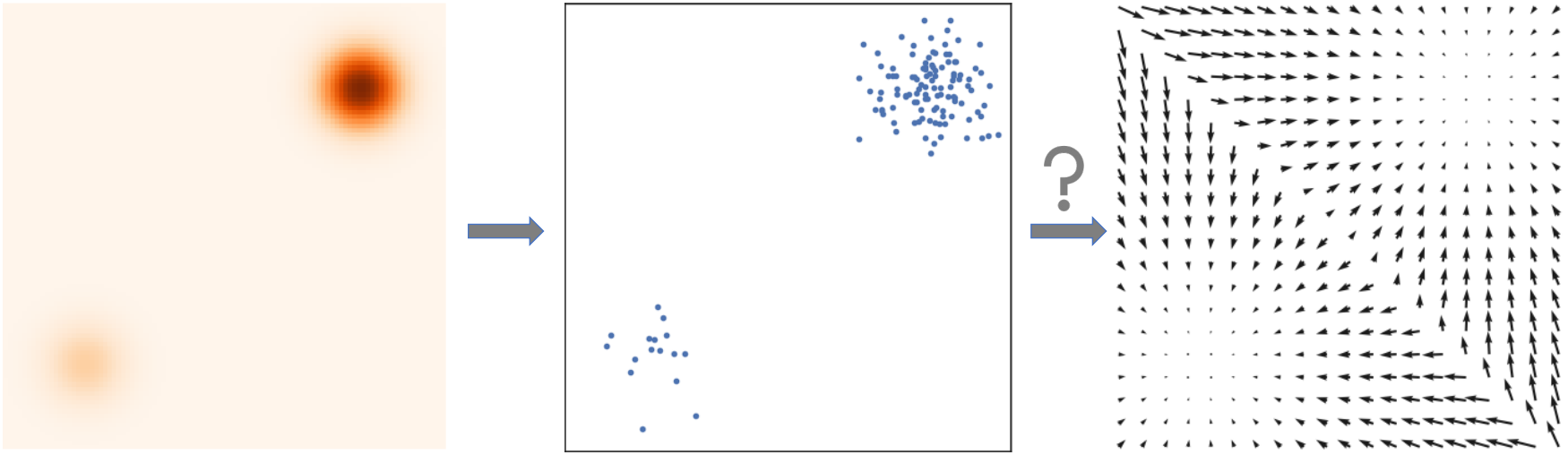
Framework for score estimation

- Score estimation by training score-based models

Probability density
$p_{data}(\boldsymbol{x})$

i.i.d. samples
$\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}$

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



Score estimation by training score-based models

- **Given**: i.i.d. samples $\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(N)}\} \sim p_{data}(\boldsymbol{x})$
- **Task**: Estimating the score $\nabla_{\boldsymbol{x}} \log p_{data}(\boldsymbol{x})$
- **Score model**: A learnable vector valued function

$$\boldsymbol{s}_\theta(\boldsymbol{x}): \mathbb{R}^{\mathrm{d}} \to \mathbb{R}^d$$

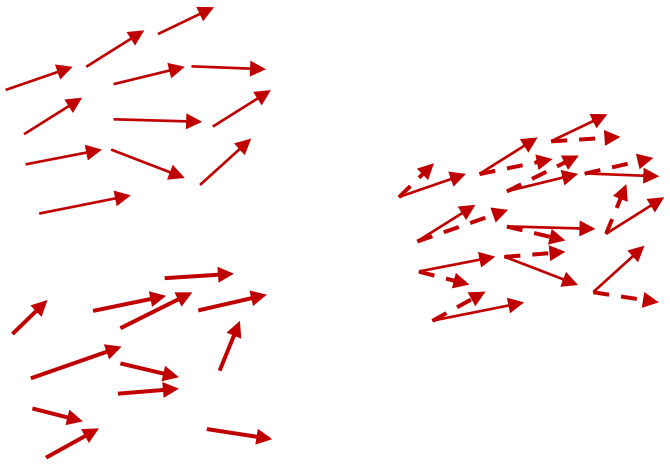
- **Goal**

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

- How to compare two vector fields of scores?

$\nabla_{\boldsymbol{x}} \log p_{data}(\boldsymbol{x})$

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



Average Euclidean distance over the space

Score estimation by training score-based models

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

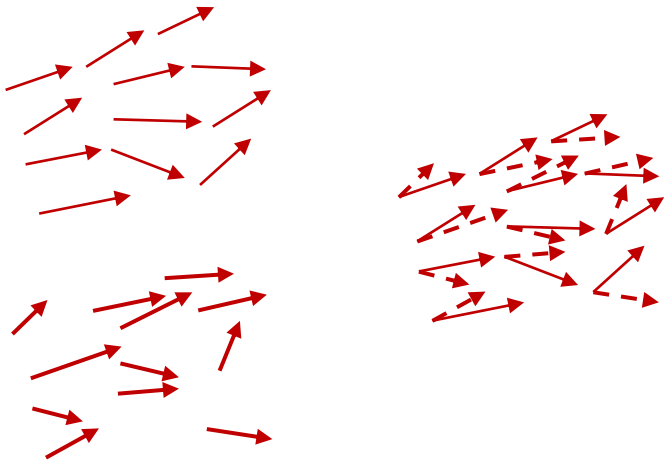
$$\frac{1}{2} E_{x \sim p_{data}} [\, \| \nabla_x \log p_{data}(x) - s_\theta(x) \|_2^2]$$

- **Score matching**

$$E_{x \sim p_{data}} \left[\frac{1}{2} \| s_\theta(x) \|_2^2 + tr\left(\nabla_x s_\theta(x) \right) \right]$$

$\nabla_x \log p_{data}(x)$

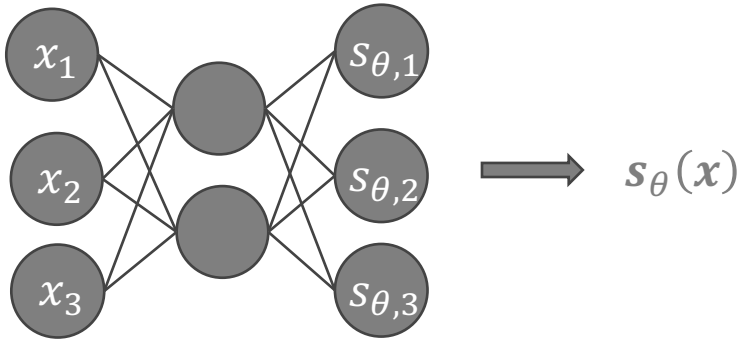
$s_\theta(x)$



Average Euclidean distance over the space

Score matching is not scalable

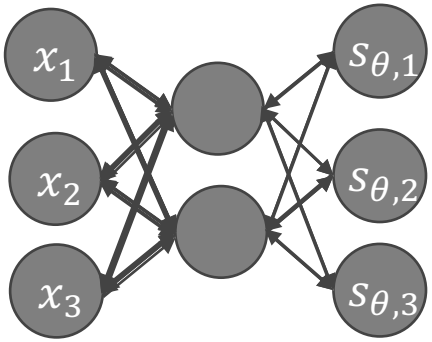
- Deep neural networks as more expressive score models



Not scalable!

- Need to compute $\|\boldsymbol{s}_\theta(\boldsymbol{x})\|_2^2$ and $\mathrm{tr}\left(\nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta(\boldsymbol{x})\right)$

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



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

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

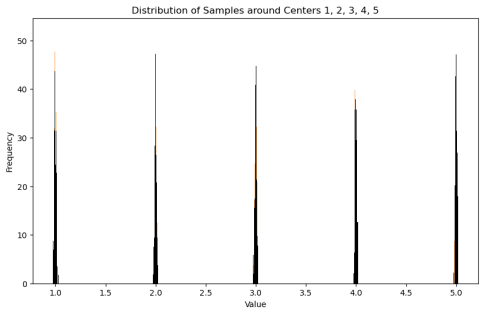
Scalable approaches to score estimation

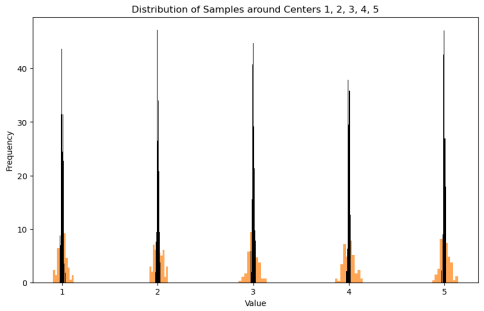
- Denoising score matching (Vincent, 2011)
- Sliced score matching (Song & Garg, 2019)
- Need to eliminate the trace of Jacobian term in the training objective

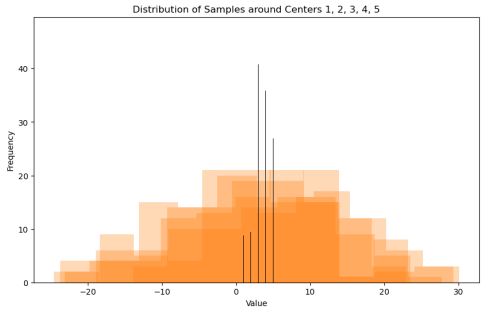
Denoising score matching (Vincent, 2011)

- Consider the perturbed distribution

$$q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) \coloneqq N(\tilde{\boldsymbol{x}}|\boldsymbol{x}, \sigma^2\boldsymbol{I}), \qquad q_\sigma(\tilde{\boldsymbol{x}}) = \int p_{data}(\boldsymbol{x}) q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) d\boldsymbol{x}$$





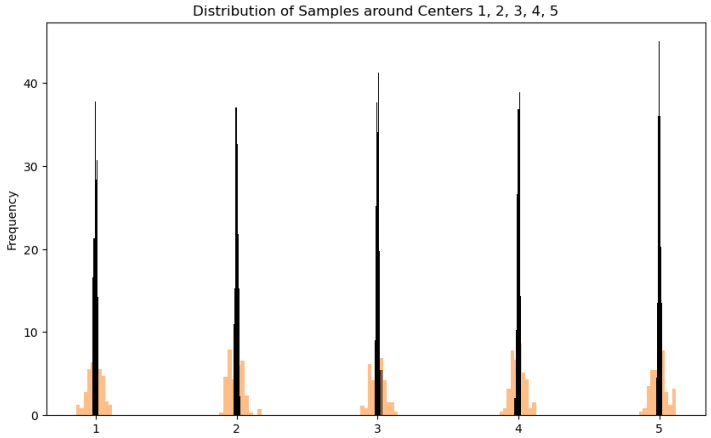


$p_{data}(\boldsymbol{x})$ $q_\sigma(\boldsymbol{x})$

Denoising score matching (Vincent, 2011)

- Consider the perturbed distribution

$$q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) \coloneqq N(\tilde{\boldsymbol{x}}|\boldsymbol{x}, \sigma^2 \boldsymbol{I}), \qquad q_\sigma(\tilde{\boldsymbol{x}}) = \int p_{data}(\boldsymbol{x}) q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) d\boldsymbol{x}$$



$p_{data}(\boldsymbol{x})$

$q_\sigma(\boldsymbol{x})$

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

Denoising score matching

- Matching the score of a noise-perturbed distribution
- Denoising score matching: $s_\theta(\tilde{\boldsymbol{x}}) \approx \nabla_{\tilde{x}} \log q_\sigma(\tilde{\boldsymbol{x}})$

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

- However, $q_\sigma(\tilde{\boldsymbol{x}}) = \int p_{data}(\boldsymbol{x}) q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) d\boldsymbol{x}$ is unknown

Denoising score matching

- Fisher divergence

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

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

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

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

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

Denoising score matching

$$\frac{1}{2}\int q_\sigma(\tilde{\boldsymbol{x}})\|\boldsymbol{s}_\theta(\tilde{\boldsymbol{x}})\|_2^2 d\tilde{\boldsymbol{x}} = \frac{1}{2}\int\int p_{data}(\boldsymbol{x})q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})\|\boldsymbol{s}_\theta(\tilde{\boldsymbol{x}})\|_2^2\, d\tilde{\boldsymbol{x}}d\boldsymbol{x}$$

$$= \frac{1}{2}E_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}E_{\tilde{\boldsymbol{x}}\sim q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})}[\|\boldsymbol{s}_\theta(\tilde{\boldsymbol{x}})\|_2^2]$$

Denoising score matching

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

$$= -\int \nabla_{\tilde{\boldsymbol{x}}}\left(\int p_{data}(\boldsymbol{x})q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})d\boldsymbol{x}\right)^T \boldsymbol{s}_\theta(\tilde{\boldsymbol{x}})d\tilde{\boldsymbol{x}}$$

$$= -\int \left(\int p_{data}(\boldsymbol{x})\nabla_{\tilde{\boldsymbol{x}}} q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})d\boldsymbol{x}\right)^T \boldsymbol{s}_\theta(\tilde{\boldsymbol{x}})d\tilde{\boldsymbol{x}}$$

$$= -\int \left(\int p_{data}(\boldsymbol{x})q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})\nabla_{\tilde{\boldsymbol{x}}}\log q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})\,d\boldsymbol{x}\right)^T \boldsymbol{s}_\theta(\tilde{\boldsymbol{x}})d\tilde{\boldsymbol{x}}$$

$$= -\int\int p_{data}(\boldsymbol{x})q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})\nabla_{\tilde{\boldsymbol{x}}}\log q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})^T\,\boldsymbol{s}_\theta(\tilde{\boldsymbol{x}})d\tilde{\boldsymbol{x}}\,d\boldsymbol{x}$$

Denoising score matching

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

$$= -\int\int p_{data}(\boldsymbol{x})q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})\nabla_{\tilde{\boldsymbol{x}}}\log q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})^T\,\boldsymbol{s}_\theta(\tilde{\boldsymbol{x}})d\tilde{\boldsymbol{x}}\,d\boldsymbol{x}$$

$$= -E_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}E_{\tilde{\boldsymbol{x}}\sim q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})}[\nabla_{\tilde{\boldsymbol{x}}}\log q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})^T\,\boldsymbol{s}_\theta(\tilde{\boldsymbol{x}})]$$

Denoising score matching

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

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

$$- E_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}E_{\tilde{\boldsymbol{x}}\sim q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})}[\nabla_{\tilde{\boldsymbol{x}}}\log q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})^T\,\boldsymbol{s}_\theta(\tilde{\boldsymbol{x}})]$$

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

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

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

Denoising score matching

- Estimate the score of a noise-perturbed distribution

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

$$\Leftrightarrow \min_{\theta} E_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})} E_{\tilde{\boldsymbol{x}}\sim q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})}[\|\nabla_{\tilde{\boldsymbol{x}}}\log q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) - \boldsymbol{s}_\theta(\tilde{\boldsymbol{x}})\|_2^2]$$

- $\nabla_{\tilde{\boldsymbol{x}}}\log q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})$ is easy to compute

$$q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) = N(\tilde{\boldsymbol{x}}|\boldsymbol{x}, \sigma^2\boldsymbol{I}), \qquad \nabla_{\tilde{\boldsymbol{x}}}\log q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) = -\frac{1}{\sigma^2}(\tilde{\boldsymbol{x}} - \boldsymbol{x})$$

- **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)

Training for denoising score matching

- Sample a minibatch of datapoints $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(n)}$ from p_{data}
- Sample a minibatch of perturbed datapoints $\tilde{\boldsymbol{x}}^{(1)}, \tilde{\boldsymbol{x}}^{(2)}, \cdots, \tilde{\boldsymbol{x}}^{(n)}$ from $q_\sigma(\tilde{\boldsymbol{x}})$

$$\tilde{\boldsymbol{x}}^{(i)} \sim q_\sigma\left(\tilde{\boldsymbol{x}}^{(i)} | \boldsymbol{x}^{(i)}\right) = N\left(\tilde{\boldsymbol{x}}^{(i)} | \boldsymbol{x}^{(i)}, \sigma^2 \boldsymbol{I}\right)$$

- Estimate the denoising score matching loss with empirical means

$$\frac{1}{2n} \sum_{i=1}^{n} \left\| \nabla_{\tilde{x}} \log q_\sigma\left(\tilde{\boldsymbol{x}}^{(i)} | \boldsymbol{x}^{(i)}\right) - \boldsymbol{s}_\theta\left(\tilde{\boldsymbol{x}}^{(i)}\right) \right\|$$

- I.e.,

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

- Stochastic gradient descent
- Need to choose a small σ

Pitfall of denoising score matching

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

$$E_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})} E_{\tilde{\boldsymbol{x}}\sim q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})}\left[\left\|\frac{1}{\sigma^2}(\tilde{\boldsymbol{x}}-\boldsymbol{x})+\boldsymbol{s}_\theta(\tilde{\boldsymbol{x}})\right\|_2^2\right]$$

$$= E_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})} E_{\boldsymbol{z}\sim N(\boldsymbol{0},\boldsymbol{I})}\left[\left\|\frac{1}{\sigma}\boldsymbol{z}+\boldsymbol{s}_\theta(\boldsymbol{x}+\sigma\boldsymbol{z})\right\|_2^2\right]$$

$$= E_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})} E_{\boldsymbol{z}\sim N(\boldsymbol{0},\boldsymbol{I})}\left[\|\boldsymbol{s}_\theta(\boldsymbol{x}+\sigma\boldsymbol{z})\|_2^2+2\boldsymbol{s}_\theta(\boldsymbol{x}+\sigma\boldsymbol{z})^T\frac{\boldsymbol{z}}{\sigma}+\frac{\|\boldsymbol{z}\|_2^2}{\sigma^2}\right]$$

- If we choose very small $\sigma \to 0$

$$\mathrm{Var}\left(\frac{\boldsymbol{z}}{\sigma}\right) \to \infty,$$

$$E_{\boldsymbol{z}\sim N(\boldsymbol{0},\boldsymbol{I})}\left[\frac{\|\boldsymbol{z}\|_2^2}{\sigma^2}\right] \to \infty$$

Denoising score matching

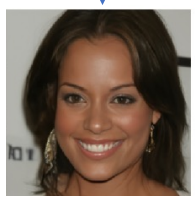
- Consider the perturbed distribution

$$q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) \coloneqq N(\tilde{\boldsymbol{x}}|\boldsymbol{x}, \sigma^2\boldsymbol{I}), \qquad q_\sigma(\tilde{\boldsymbol{x}}) = \int p_{data}(\boldsymbol{x}) q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) d\boldsymbol{x}$$

- Score estimation for $\nabla_{\tilde{\boldsymbol{x}}} \log q_\sigma(\tilde{\boldsymbol{x}})$
- Score matching

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

$$\Leftrightarrow \min_\theta E_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})} E_{\tilde{\boldsymbol{x}}\sim q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})} \left[\left\| \frac{1}{\sigma^2}(\tilde{\boldsymbol{x}} - \boldsymbol{x}) + \boldsymbol{s}_\theta(\tilde{\boldsymbol{x}}) \right\|_2^2 \right]$$



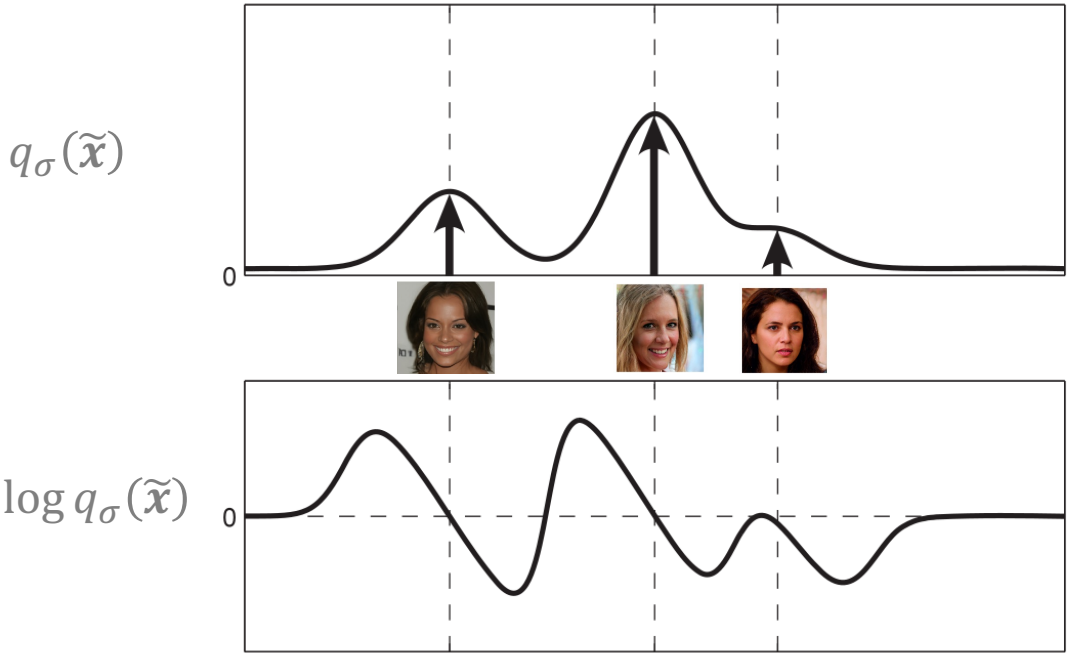
- $\boldsymbol{s}_\theta(\tilde{\boldsymbol{x}})$ tries to estimate the noise that was added to produce $\tilde{\boldsymbol{x}}$

Tweedie's formula

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

$$E_{x \sim p_{data}(x)} E_{\tilde{x} \sim q_\sigma(\tilde{x}|x)} \left[\left\| \frac{1}{\sigma^2} (\tilde{x} - x) + s_\theta(\tilde{x}) \right\|_2^2 \right]$$

- **Tweedie's formula**: Optimal denoising strategy is to follow the gradient (score):



Tweedie's formula and denoising score matching

- Denoising score matching is suitable for optimal denoising
- Given $p_{data}(\boldsymbol{x})$ and $q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})$, we can define the posterior $p(\boldsymbol{x}|\tilde{\boldsymbol{x}})$ with Bayes' rule

$$p(\boldsymbol{x}|\tilde{\boldsymbol{x}}) \propto p_{data}(\boldsymbol{x}) q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})$$

- Recall that

$$q_\sigma(\tilde{\boldsymbol{x}}) = \int p_{data}(\boldsymbol{x}) q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) d\boldsymbol{x}$$

- Tweedie's formula:

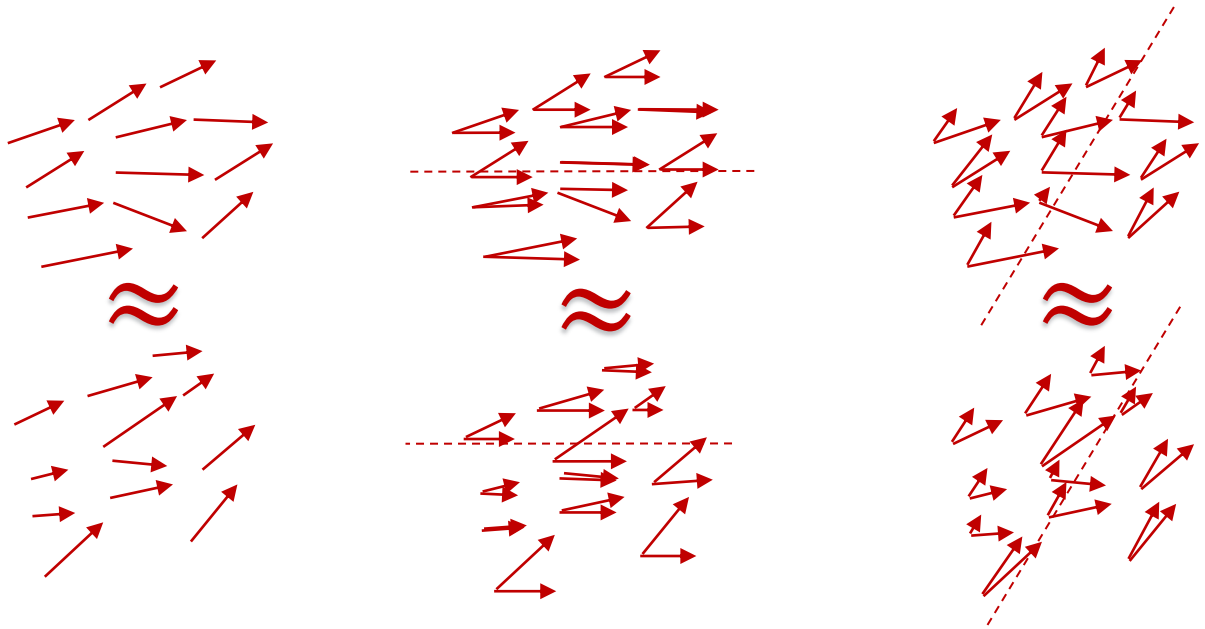
$$E_{x \sim p(\boldsymbol{x}|\tilde{\boldsymbol{x}})}[\boldsymbol{x}] = \tilde{\boldsymbol{x}} + \sigma^2 \nabla_x \log q_\sigma(\tilde{\boldsymbol{x}}) \approx \tilde{\boldsymbol{x}} + \sigma^2 s_\theta(\tilde{\boldsymbol{x}})$$

Sliced score matching

- One dimensional problems is easier
- Consider projections onto random directions

$\nabla_x \log p_{data}(\boldsymbol{x})$

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



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

Sliced score matching

- **Objective**: Sliced Fisher Divergence

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

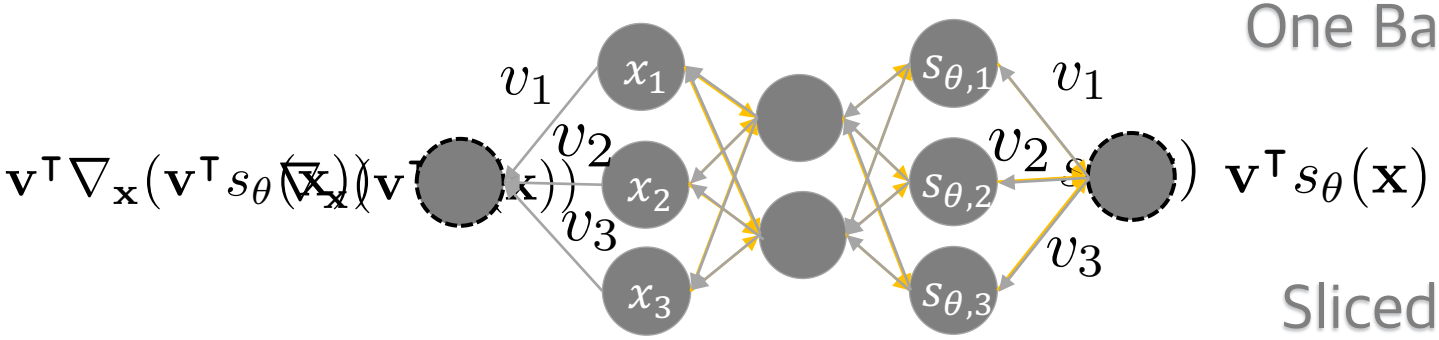
- Integration by parts

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

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

Computing Jacobian-vector products is scalable

$$v^T \nabla_x s_\theta(x) v = v^T \nabla_x \left(v^T s_\theta(x) \right)$$



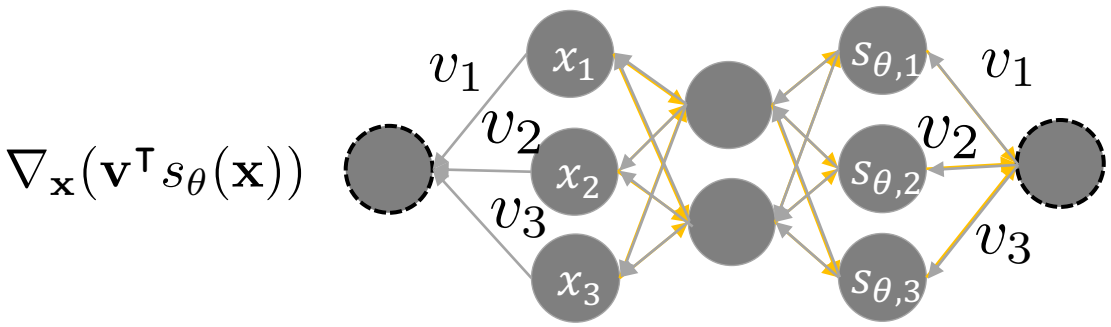
One Backprop!

Sliced Score Matching is scalable

- Slightly slower than denoising score matching

Computing Jacobian-vector products is scalable

$$v^T \nabla_x s_\theta(x) v = v^T \nabla_x \left(v^T s_\theta(x) \right)$$



One Backprop!

Sliced Score Matching is scalable

- Slightly slower than denoising score matching

Training of sliced score matching

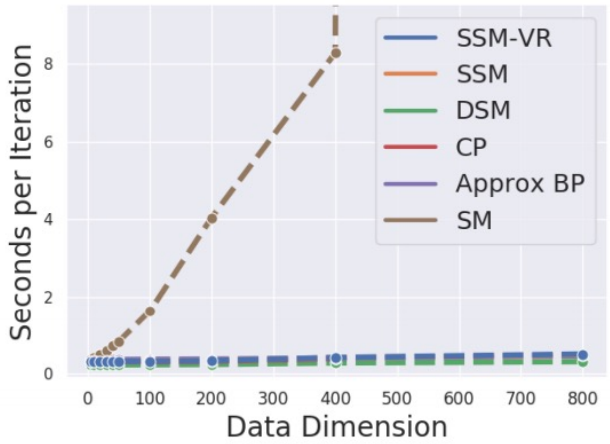
- Sample a minibatch of datapoints $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(n)}$ from p_{data}
- Sample a minibatch of projection directions $\boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}, \cdots, \boldsymbol{v}^{(n)}$ from $p_{\boldsymbol{v}}$
- Estimate the sliced score matching loss with empirical means

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

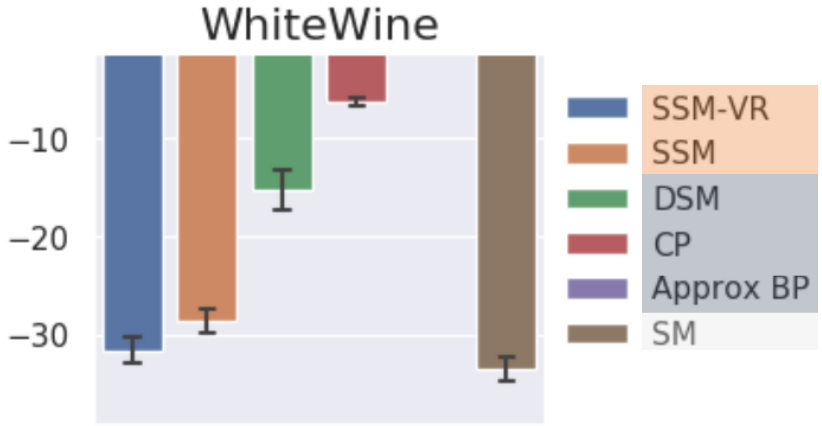
- The projection distribution is typically Gaussian or Rademacher
- Stochastic gradient descent
- Can use more projections per datapoint to boost performance

Experimental results for sliced score matching

Sliced score matching methods | Other Baselines | Score Matching



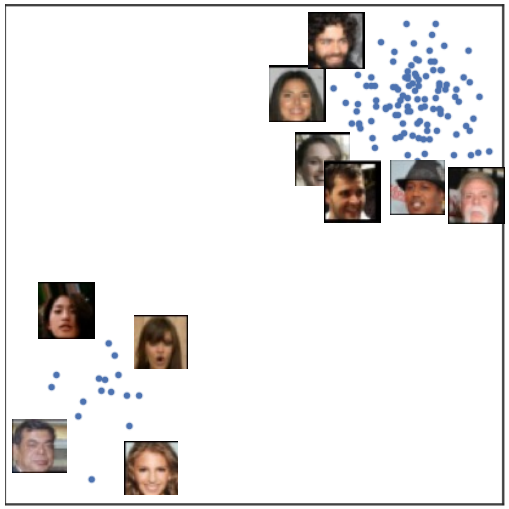
Efficiency



Performance on density estimation

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

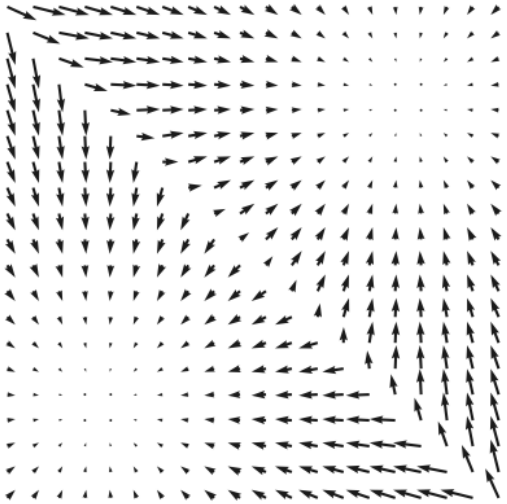
Score-based generative modeling



Data samples

$\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}$

Score Matching



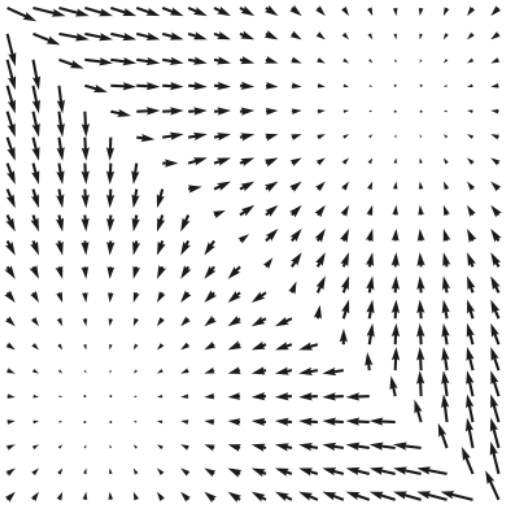
Scores function

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

?

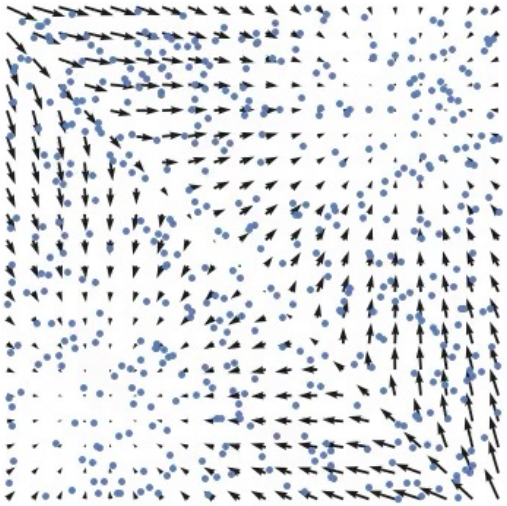
New samples

From scores to samples: Langevin MCMC



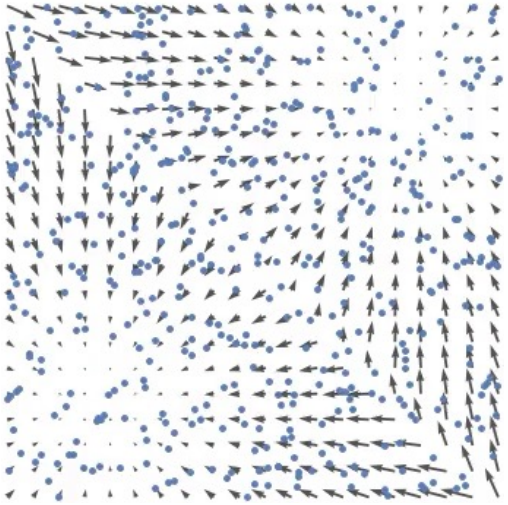
Scores

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



Follow the scores

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t + \frac{\epsilon}{2}\boldsymbol{s}_\theta(\boldsymbol{x})$$



Follow noisy scores: Langevin MCMC

$$\boldsymbol{z} \sim N(\boldsymbol{0}, \boldsymbol{I})$$

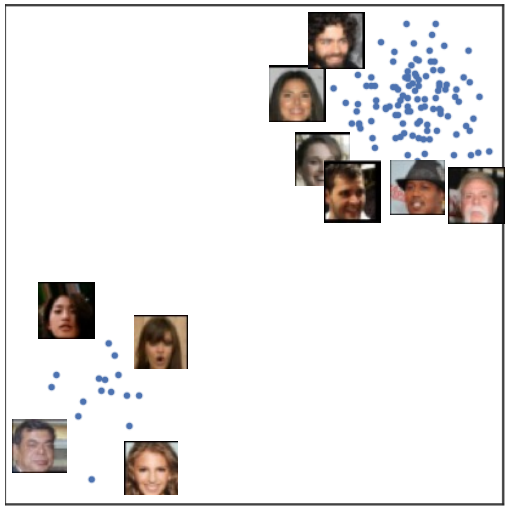
$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t + \frac{\epsilon}{2}\boldsymbol{s}_\theta(\boldsymbol{x}) + \sqrt{\epsilon}\boldsymbol{z}$$

Langevin dynamics sampling

- Sample from $p_{data}(\boldsymbol{x})$ using only the scores $\nabla_{\boldsymbol{x}} \log p_{data}(\boldsymbol{x})$
  - Initialize $\boldsymbol{x}^0 \sim \pi(\boldsymbol{x})$
  - Repeat for $t = 0, \ldots, T-1$
    - $\boldsymbol{z} \sim N(0, I)$
    - $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t + \frac{\epsilon}{2} \nabla_{\boldsymbol{x}} \log p_{data}(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}^t} + \sqrt{\epsilon}\boldsymbol{z}$
  - If $\epsilon \to 0$ and $T \to \infty$, then we have $\boldsymbol{x}^T$ converges to a sample from p_{data}
- Langevin dynamics + score estimation

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

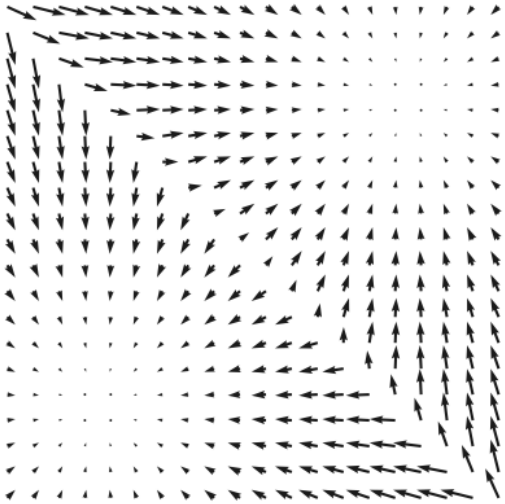
Score-based generative modeling



Data samples

$\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}$

Score Matching



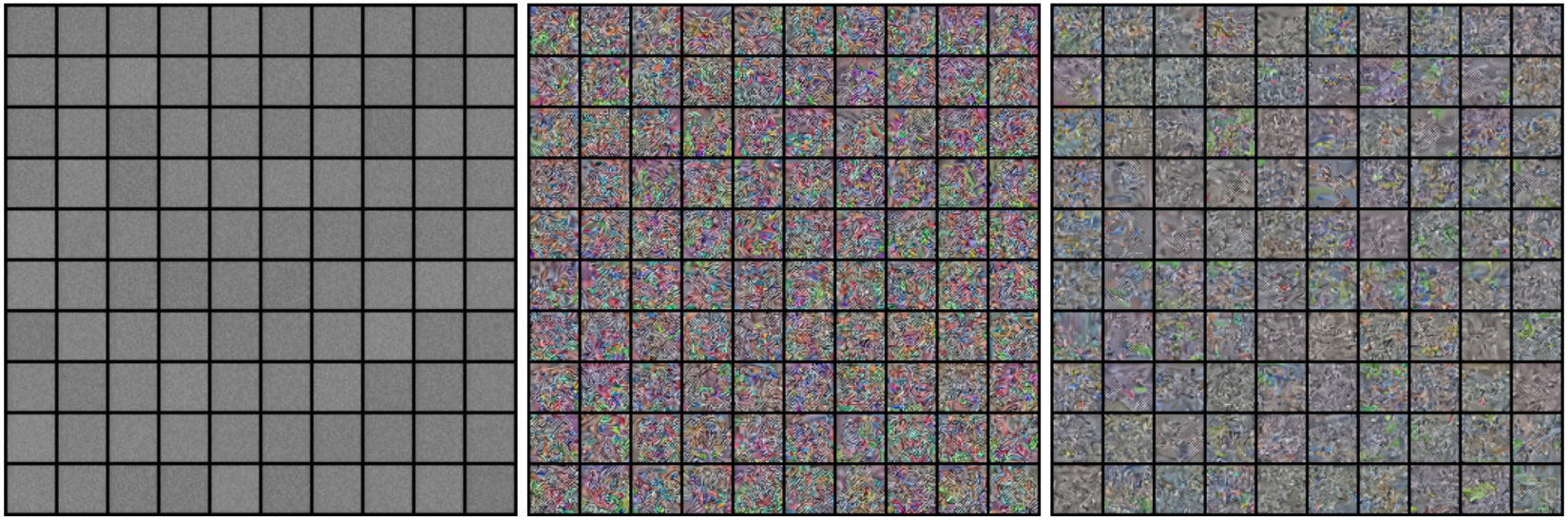
Scores function

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

Langevin dynamics

New samples

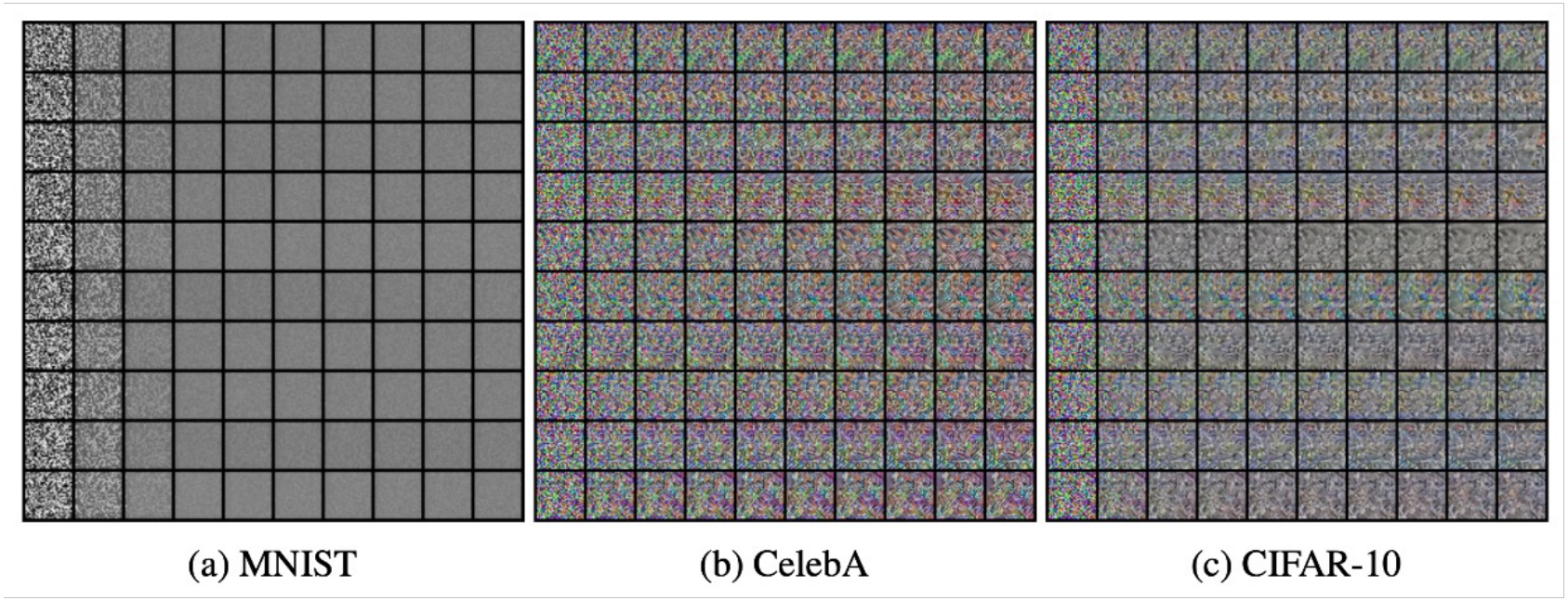
Score-based generative modeling: results



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

Final samples

Score-based generative modeling: results

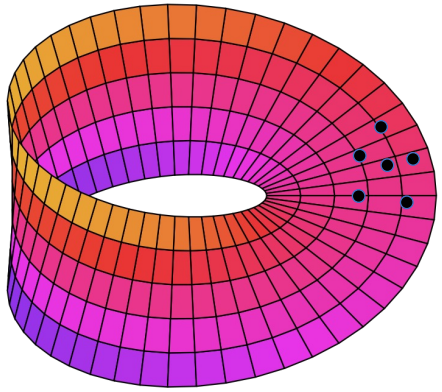


(a) MNIST (b) CelebA (c) CIFAR-10

Langevin sampling process

Pitfall 1: manifold hypothesis

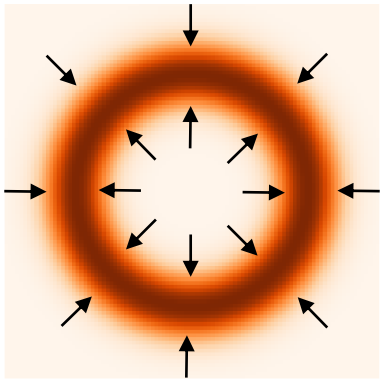
- Manifold hypothesis



Data points

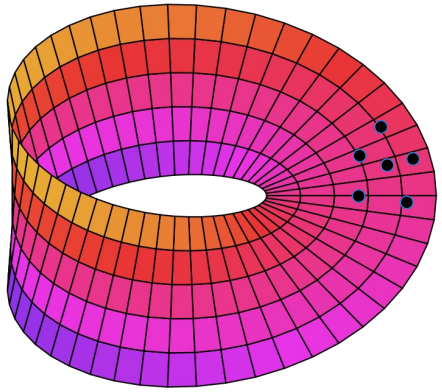
- Data score is undefined

$$\nabla_{\boldsymbol{x}} \log p_{data}(\boldsymbol{x})$$



Pitfall 1: manifold hypothesis

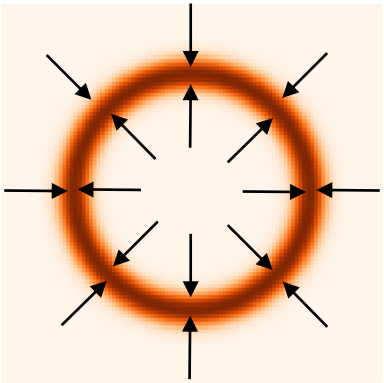
- Manifold hypothesis



Data points

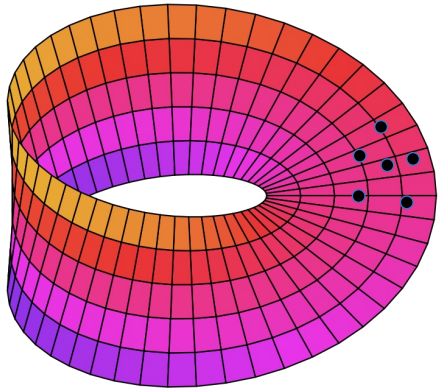
- Data score is undefined

$$\nabla_{\boldsymbol{x}} \log p_{data}(\boldsymbol{x})$$



Pitfall 1: manifold hypothesis

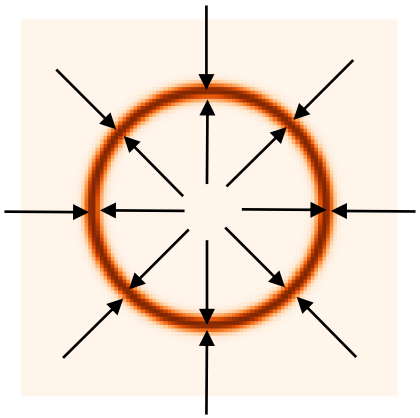
- Manifold hypothesis



Data points

- Data score is undefined

$$\nabla_{\boldsymbol{x}} \log p_{data}(\boldsymbol{x})$$

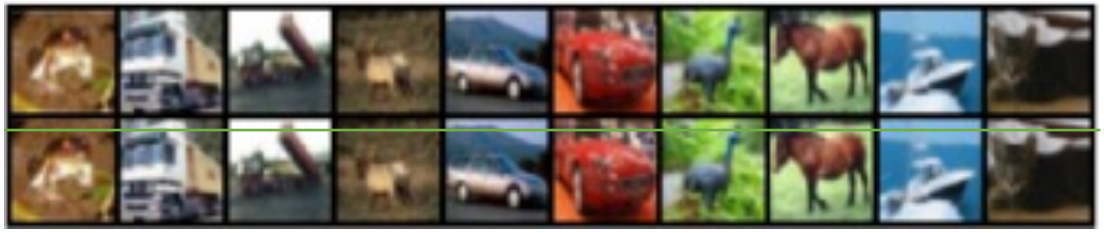


Pitfall 1: manifold hypothesis

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

784 Dim

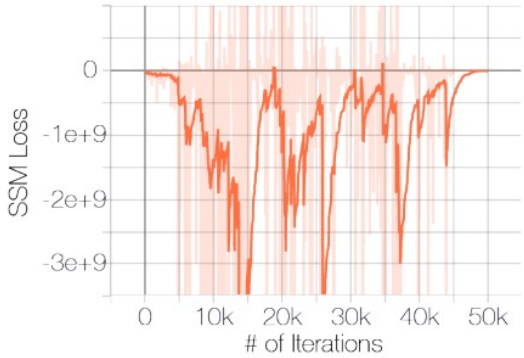
595



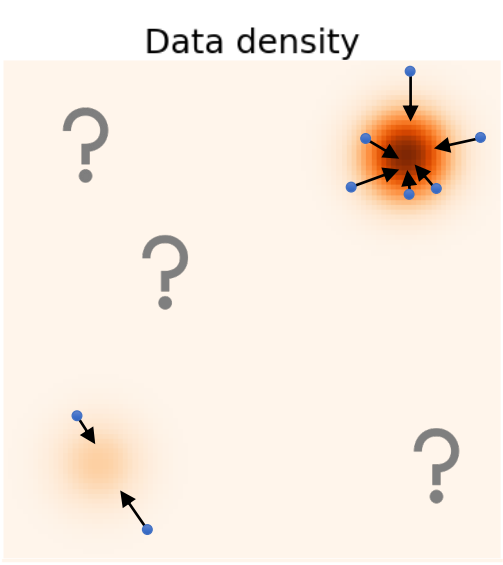
3072 Dim

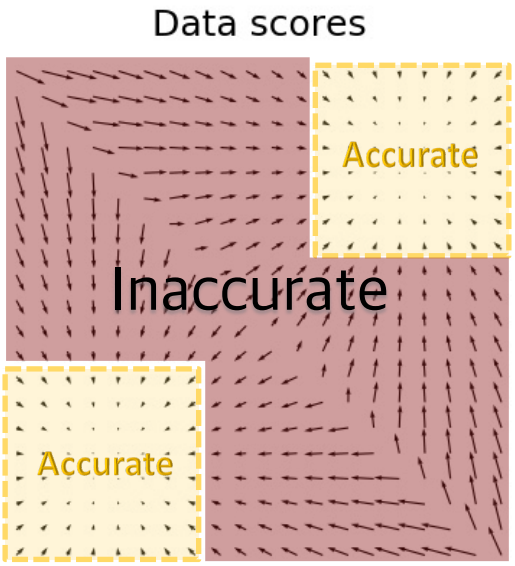
2165

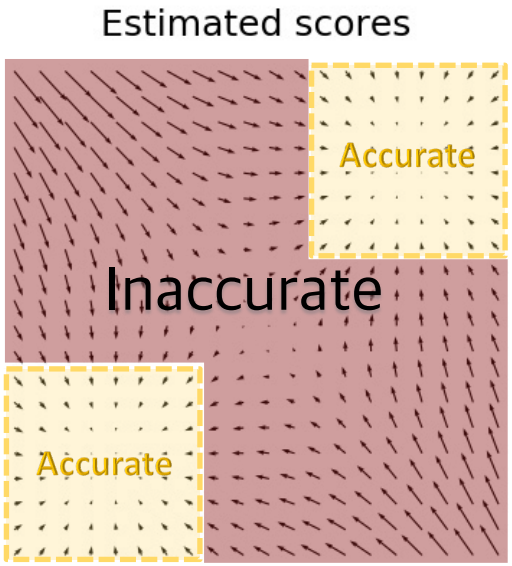
- Score estimation on CIFAR-10



Pitfall 2: 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 modes with disjoint supports:

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

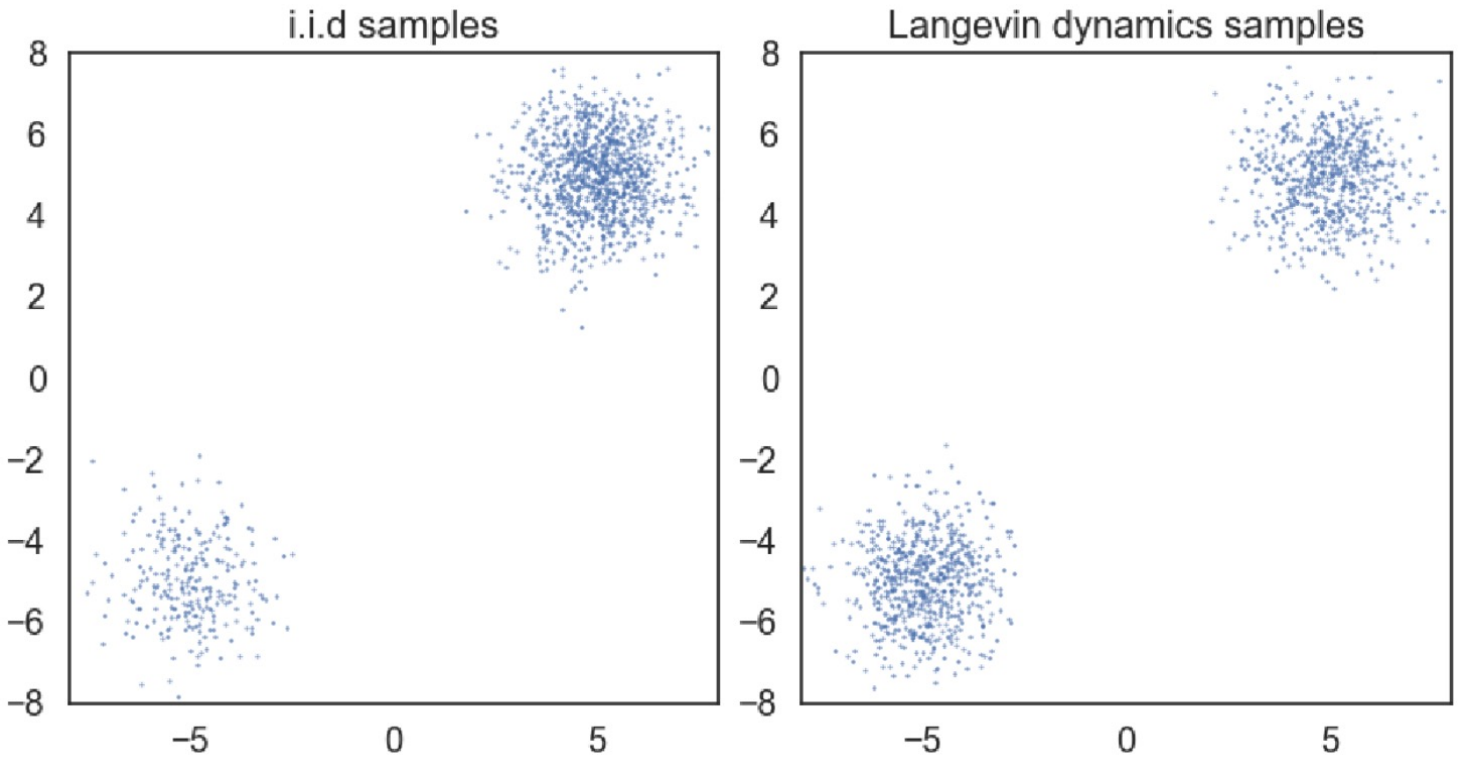
$$A \cap B = \emptyset, \qquad p_{data}(x) = \begin{cases} \pi p_1(\boldsymbol{x}) & \boldsymbol{x} \in A \\ (1-\pi)p_2(\boldsymbol{x}) & \boldsymbol{x} \in B \end{cases}$$

- Data score function:

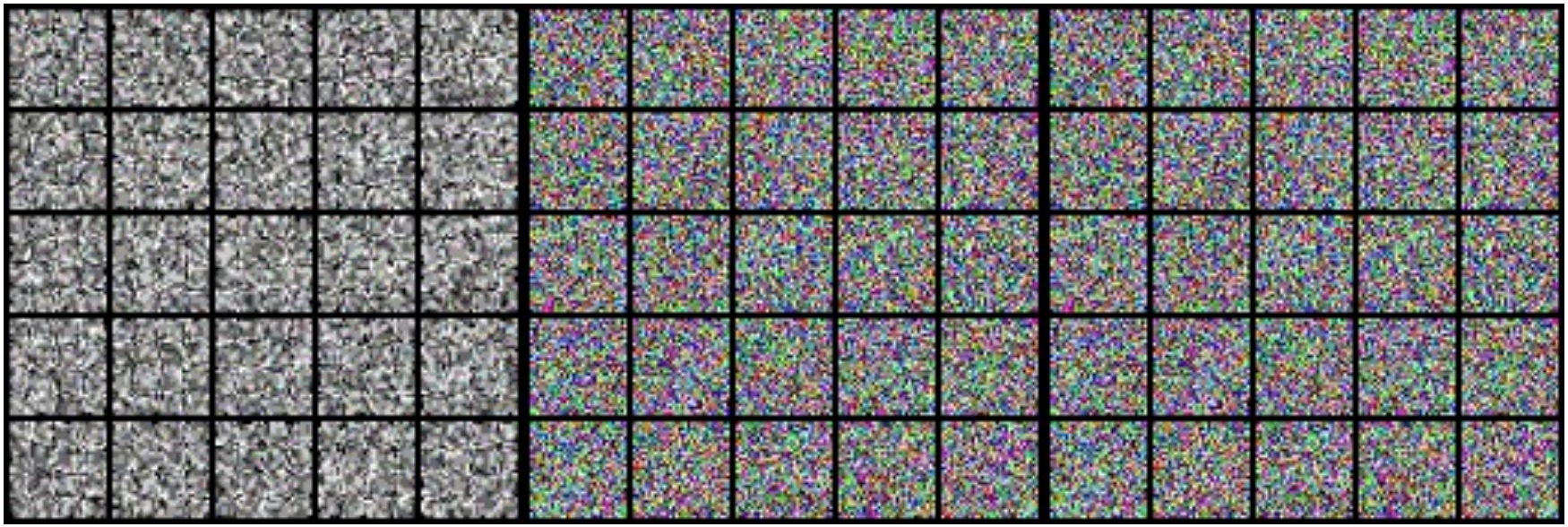
$$\nabla_{\boldsymbol{x}} \log p_{data}(\boldsymbol{x}) = \begin{cases} \nabla_{\boldsymbol{x}} \log p_1(\boldsymbol{x}) & \boldsymbol{x} \in A \\ \nabla_{\boldsymbol{x}} \log p_2(\boldsymbol{x}) & \boldsymbol{x} \in B \end{cases}$$

- The score function has no dependence on π
- Langevin sampling will not reflect π

Pitfall 3: slow mixing of Langevin dynamics between data modes



After fixing pitfalls



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

Thanks