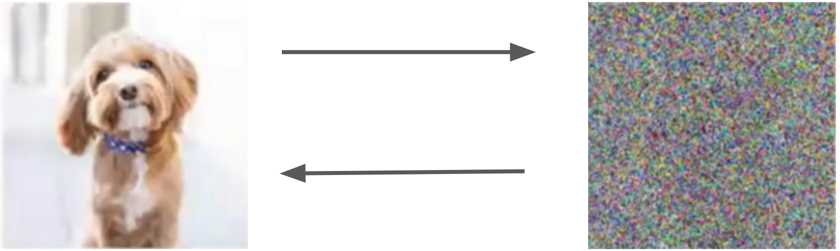
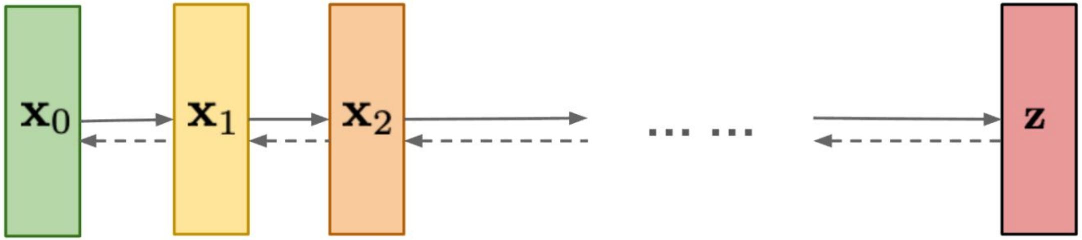
Brief introduction to Diffusion Model And Score-based generative Models

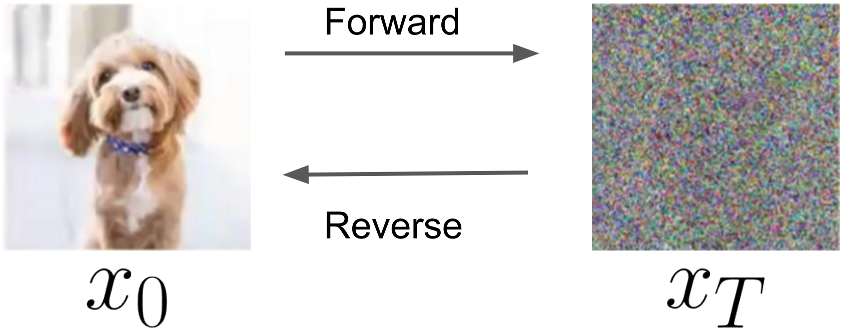
Diffusion Model-idea



Diffusion Model-idea

Diffusion models:
Gradually add Gaussian noise and then reverse





Diffusion Model-Forward Process

$$x_0 \rightarrow x_1 \rightarrow \cdots \rightarrow x_T$$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$

$$\boxed{q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})}$$

β_t Is the variance at time t. hyperparameter

$\beta_1 < \beta_2 < \cdots < \beta_T \quad \beta_t \in (0, 1)$

Diffusion Model-Forward Process

$$x_0 \rightarrow x_1 \rightarrow \cdots \rightarrow x_T$$

1. $T \rightarrow \infty,\ q(\mathbf{x}_T | x_0) \approx \mathcal{N}(0, \mathbf{I})$

2. $q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I})$

$$\text{Let } \alpha_t = 1 - \beta_t \text{ and } \bar{\alpha}_t = \prod_{i=1}^{T} \alpha_i$$

$$\boxed{q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)\mathbf{I})}$$

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\mathbf{z}_{t-1} \\
&= \sqrt{\alpha_t \alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t \alpha_{t-1}}\bar{\mathbf{z}}_{t-2} \\
&= \dots \\
&= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{z}
\end{aligned}$$

;where $\mathbf{z}_{t-1}, \mathbf{z}_{t-2}, \dots \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

;where $\bar{\mathbf{z}}_{t-2}$ merges two Gaussians (*).

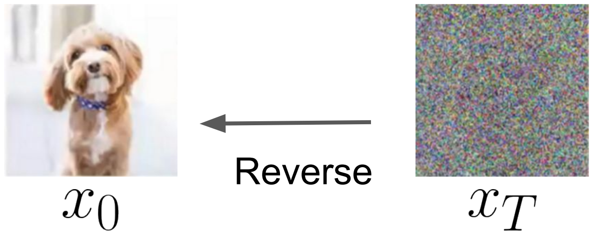
Diffusion Model-Reverse Process

$$x_0 \leftarrow x_1 \leftarrow \cdots \leftarrow x_T$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

$$p(x_T) = \mathcal{N}(x_T; 0, \mathbf{I})$$



Diffusion Model-Objective function

$$p_\theta(x_0) = \int p_\theta(x_{0:T})dx_{1:T}$$ Intractable.

We can view $x_1, x_2, \ldots, x_T$ as latent variable and x_0 as observed variable.

ELBO for VAE:

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p_\theta(z))$$

ELBO for Diffusion Models:

$$\log p_\theta(x_0) \geq \mathbb{E}_{q(x_{1:T}|x_0)}[\log p_\theta(x_0|x_{1:T})] - D_{KL}(q(x_{1:T}|x_0)||p_\theta(x_{1:T}))$$

$$\boxed{q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})}$$

Diffusion Model-Objective function

$$\log p_\theta(x_0) \geq \mathbb{E}_{q(x_{1:T}|x_0)}[\log p_\theta(x_0|x_{1:T})] - D_{KL}(q(x_{1:T}|x_0)||p_\theta(x_{1:T}))$$

$$= \mathbb{E}_{q(x_{1:T}|x_0)}[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}]$$

$$\boxed{= \mathbb{E}_{q(x_{1:T}|x_0)}[\log p_\theta(x_T) + \sum_{t\geq 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}]}$$

Sample pair of x_t-1 and x_t

$$\boxed{= \mathbb{E}_{q(x_{1:T}|x_0)}[-\sum_t D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))] - C}$$

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}$$

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t\mathbf{I})$$

Diffusion Model-Objective function

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

$$\boxed{\mathbb{E}_{q(x_{1:T}|x_0)}[-\sum_t D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))]}$$

$$= \mathbb{E}_{\mathbf{x}_0,\epsilon,t}\left[||\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)||^2\right]$$

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

$$x_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_{\mathbf{t}} \quad \epsilon_t \sim \mathcal{N}(0, \mathbf{I})$$

$$\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_{\mathbf{t}}\right) \qquad \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right)$$

$$\boxed{\text{loss} = \mathbb{E}_{x_0,\epsilon,t}[||\epsilon_t - \epsilon_\theta(x_t, t)||^2]}$$

“DDPM”
2015

“Diffusion Models Beat GANs on Image Synthesis”

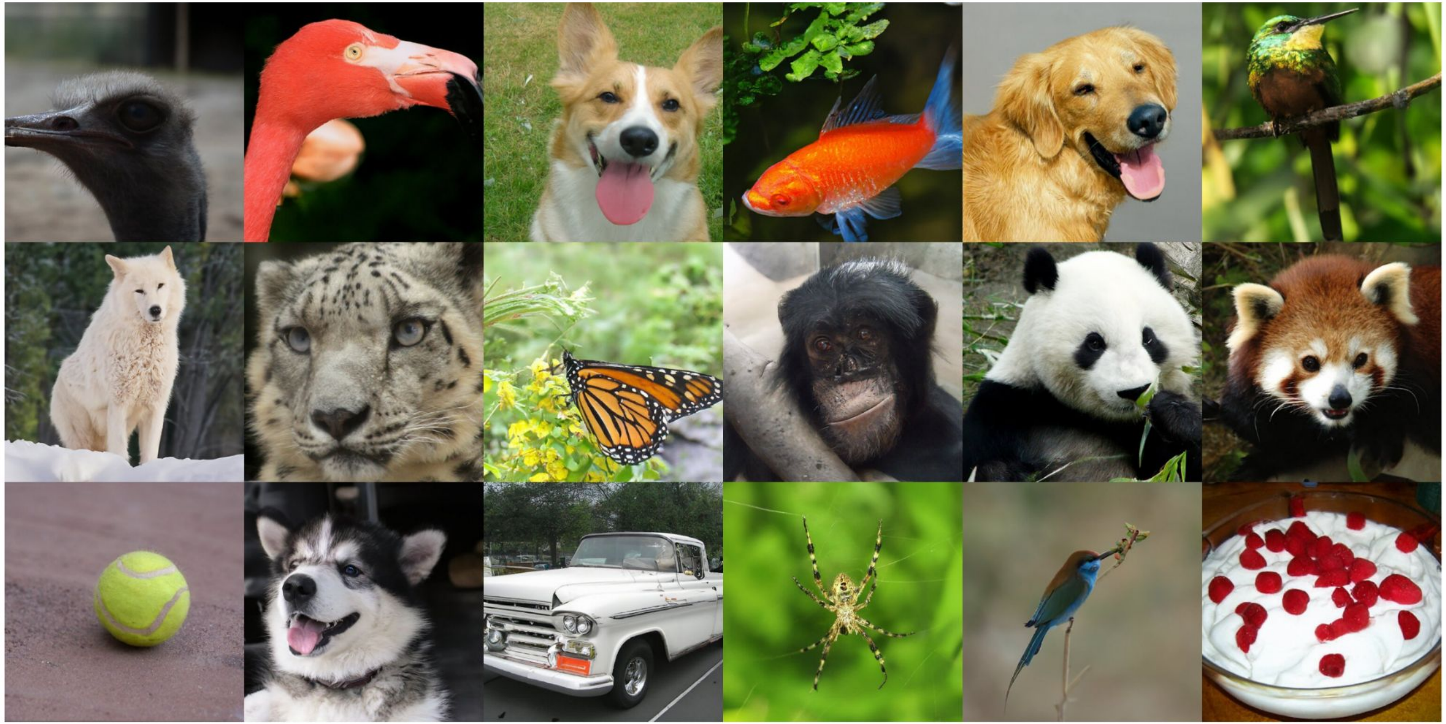
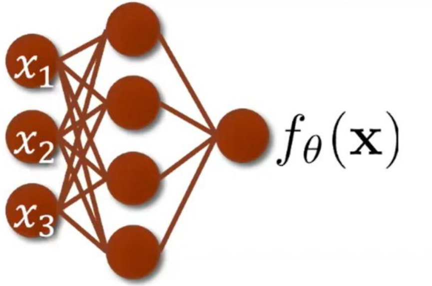


Figure 1: Selected samples from our best ImageNet 512×512 model (FID 3.85)

Score-based generative model

- Deep Energy-Based models (EBMs)

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



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

$$Z_{\boldsymbol{\theta}} = \int e^{f_{\boldsymbol{\theta}}(\mathbf{x})} \, \mathrm{d}\mathbf{x}$$

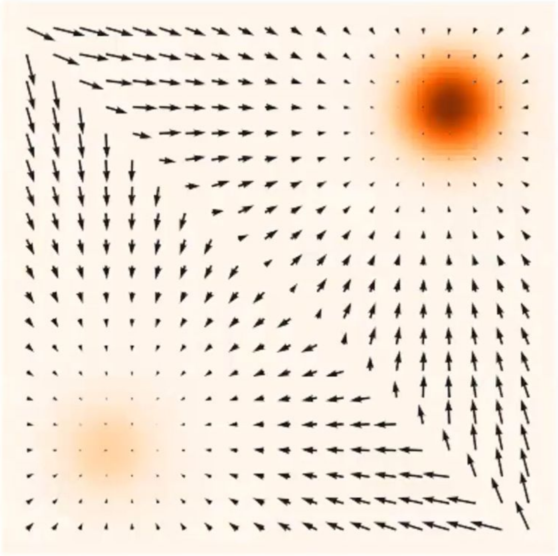
- **Cons**: Learning parameter θ via maximum likelihood (MLE) is hard

$$\mathbb{E}_{p_{\text{data}}}[-\log p_\theta(\mathbf{x})] = \mathbb{E}_{p_{\text{data}}}[\log f_\theta(\mathbf{x}) - \cancel{\log Z_\theta}]$$

Score-based generative model

The gradient of a probability density w.r.t. the input dimensions

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



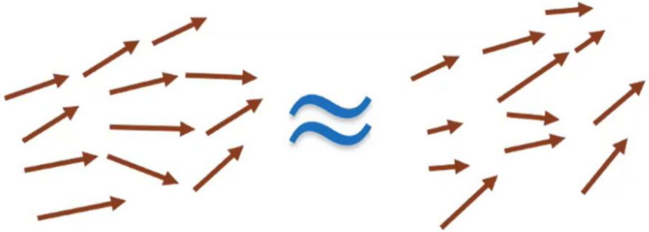
Score vs. density function

Score-based generative model

- Score does not depend on the partition function

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

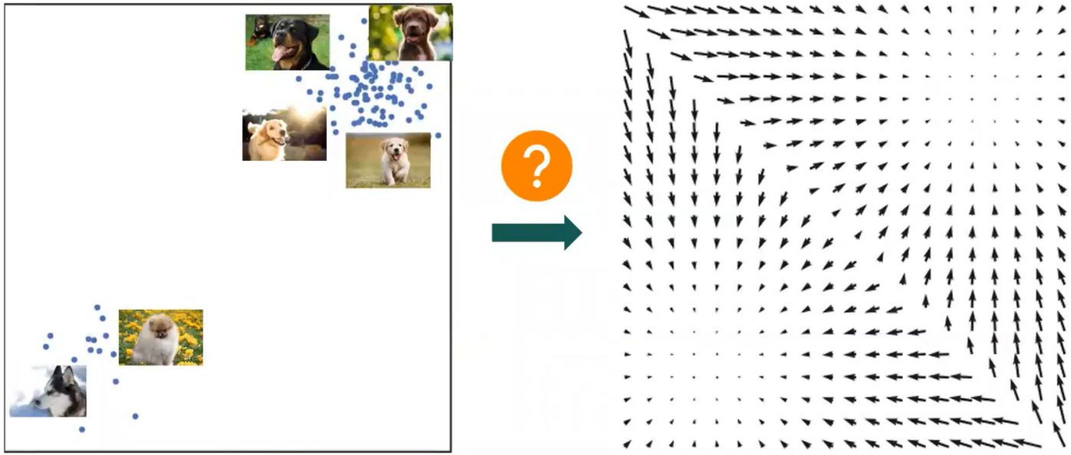
Score Model: $s_\theta(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}^d$

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

- **Idea**: learn θ by fitting $\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x})$ to $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$

Score-based generative model

Score models can be estimated from data



Training data

$$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \overset{\text{i.i.d.}}{\sim} p_{\text{data}}(\mathbf{x})$$

Score function

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

Score-based generative model

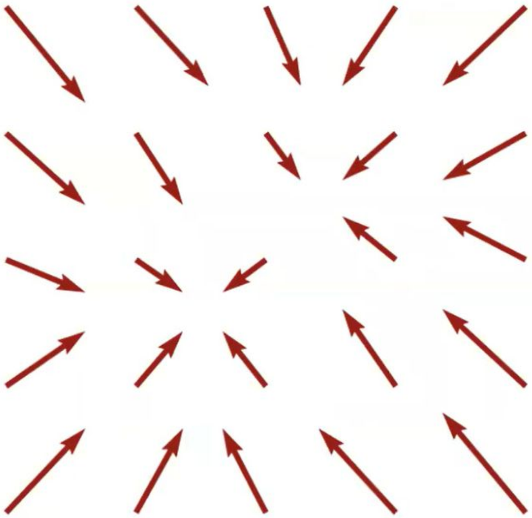
Given: $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \stackrel{\text{i.i.d.}}{\sim} p_{\text{data}}(\mathbf{x})$

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

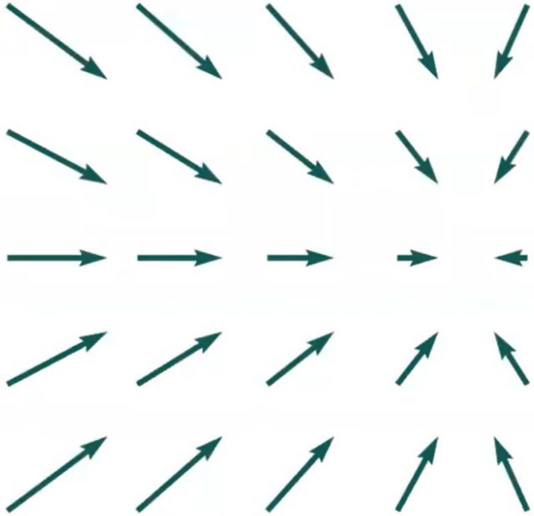
Score Model: $s_{\boldsymbol{\theta}}(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^d \approx \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$

Objective: How to compare two vector fields of scores?

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

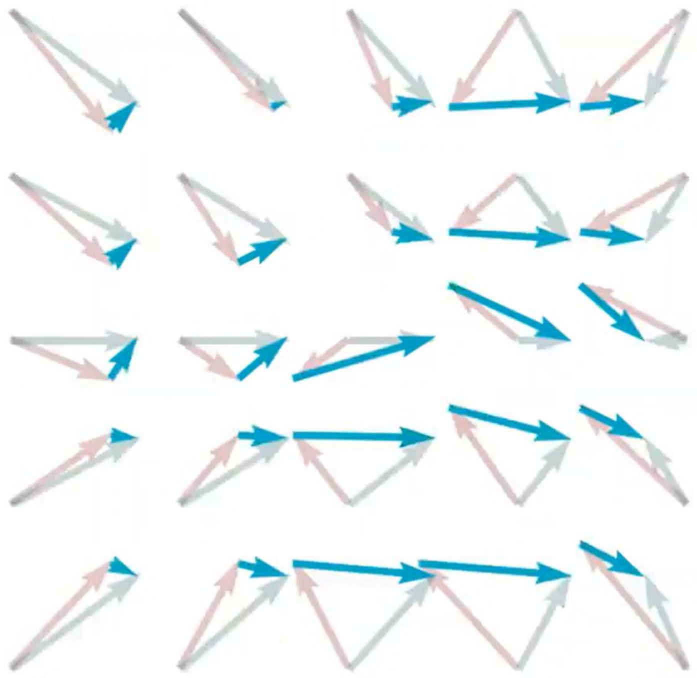


$s_{\boldsymbol{\theta}}(\mathbf{x})$



Score-based generative model

Objective: How to compare two vector fields of scores?



Score-based generative model:score matching

- Average Euclidean distance over the whole space.

$$\frac{1}{2}\mathbb{E}_{p(\mathbf{x})}[||\nabla_{\mathbf{x}} \log p(\mathbf{x}) - s_\theta(\mathbf{x})||_2^2]$$

Can approximate the
expectation by MC
From the training set.

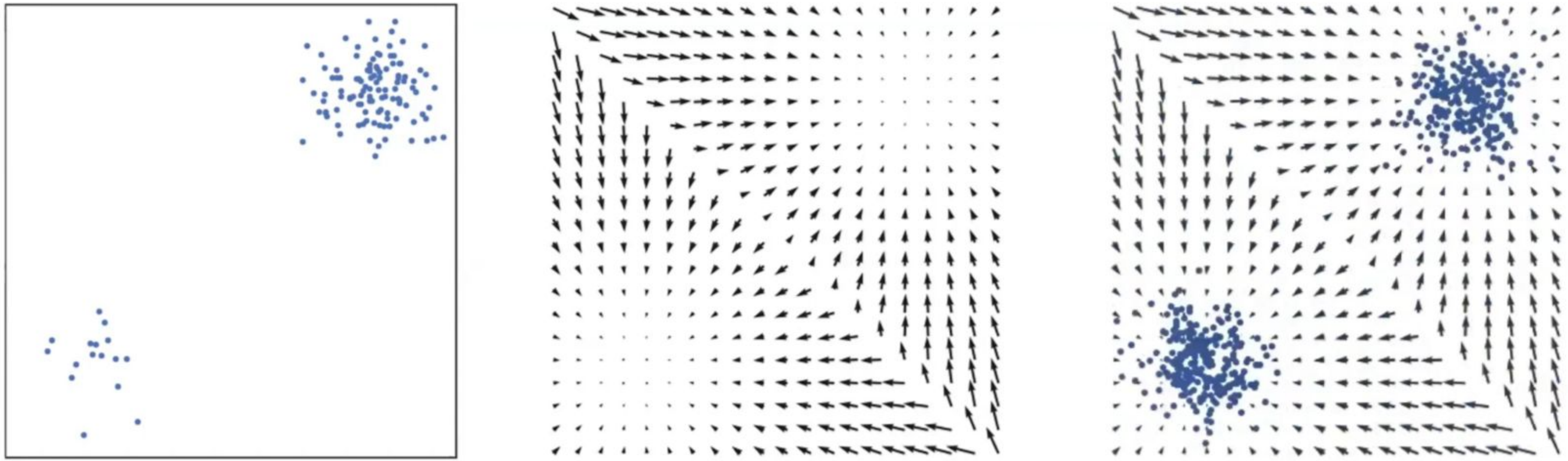
(Fisher divergence)

- Integration by parts

$$\mathbb{E}_{p(\mathbf{x})}\left[\frac{1}{2}\|s_\theta(\mathbf{x})\|_2^2 + \mathrm{tr}(\underbrace{\nabla_{\mathbf{x}} s_\theta(\mathbf{x})}_{\text{Jacobian of } s_\theta(\mathbf{x})})\right]$$

Score Matching
Hyvarinen (2005)

Score-based generative model: generation



Samples Score Estimation Langevin dynamics

Score-based generative model: generation

Langevin dynamics sampling

- Sample from $p(\mathbf{x})$ using only the score $\nabla_{\mathbf{x}} \log p(\mathbf{x})$
- Initialize $\tilde{\mathbf{x}}_0 \sim \pi(\mathbf{x})$
- Repeat for $t \leftarrow 1, 2, \cdots, T$

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

Little gaussian noise.

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

Take a step in the direction of the gradient

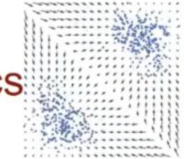
Score-based generative model:

$$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \overset{\text{i.i.d.}}{\sim} p_{\text{data}}(\mathbf{x})$$

Score Matching

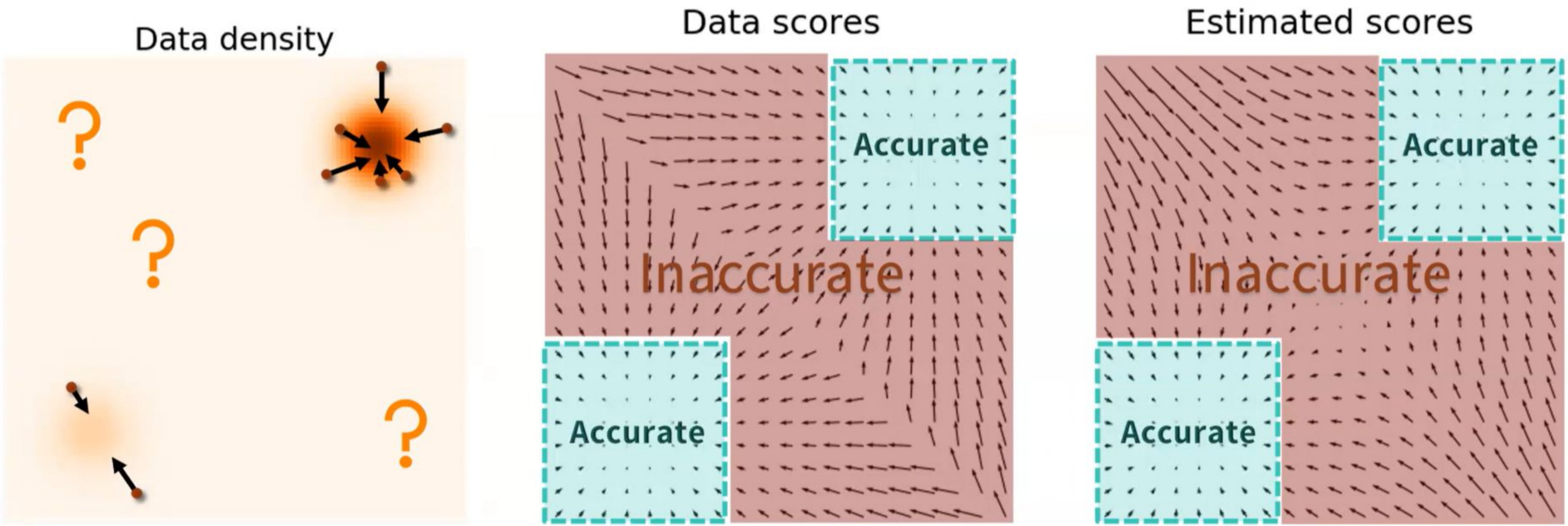
$$\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$$

Langevin dynamics



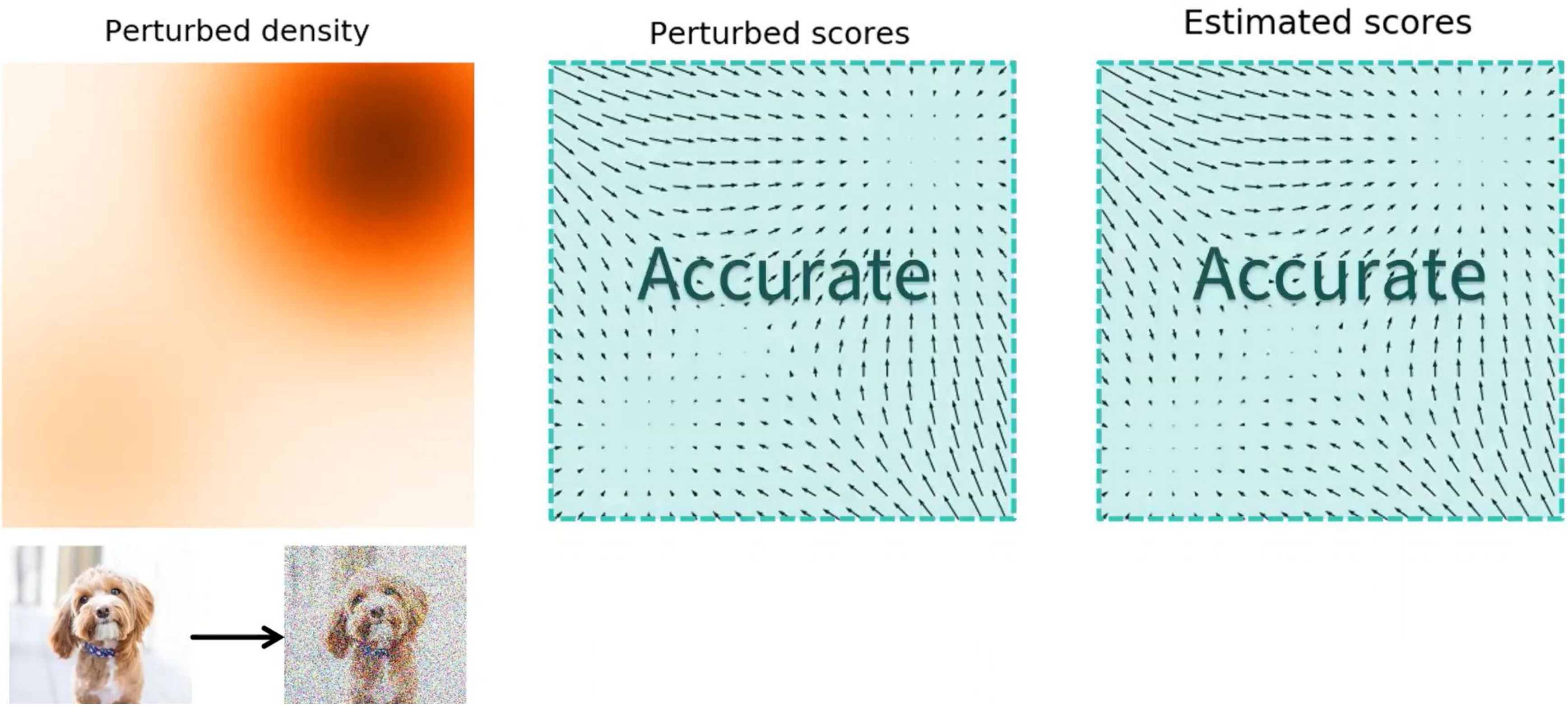
Samples

Score-based generative model

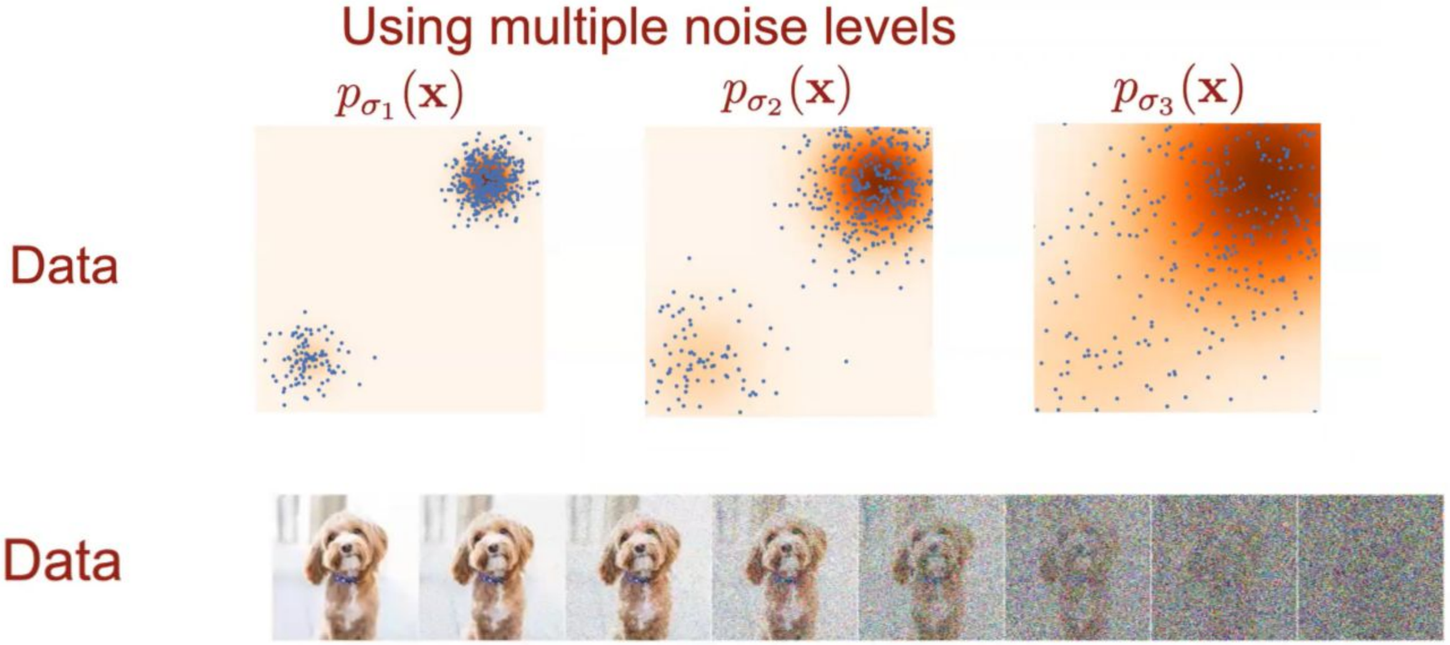


$$\frac{1}{2}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}[\|\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x})\|_2^2]$$

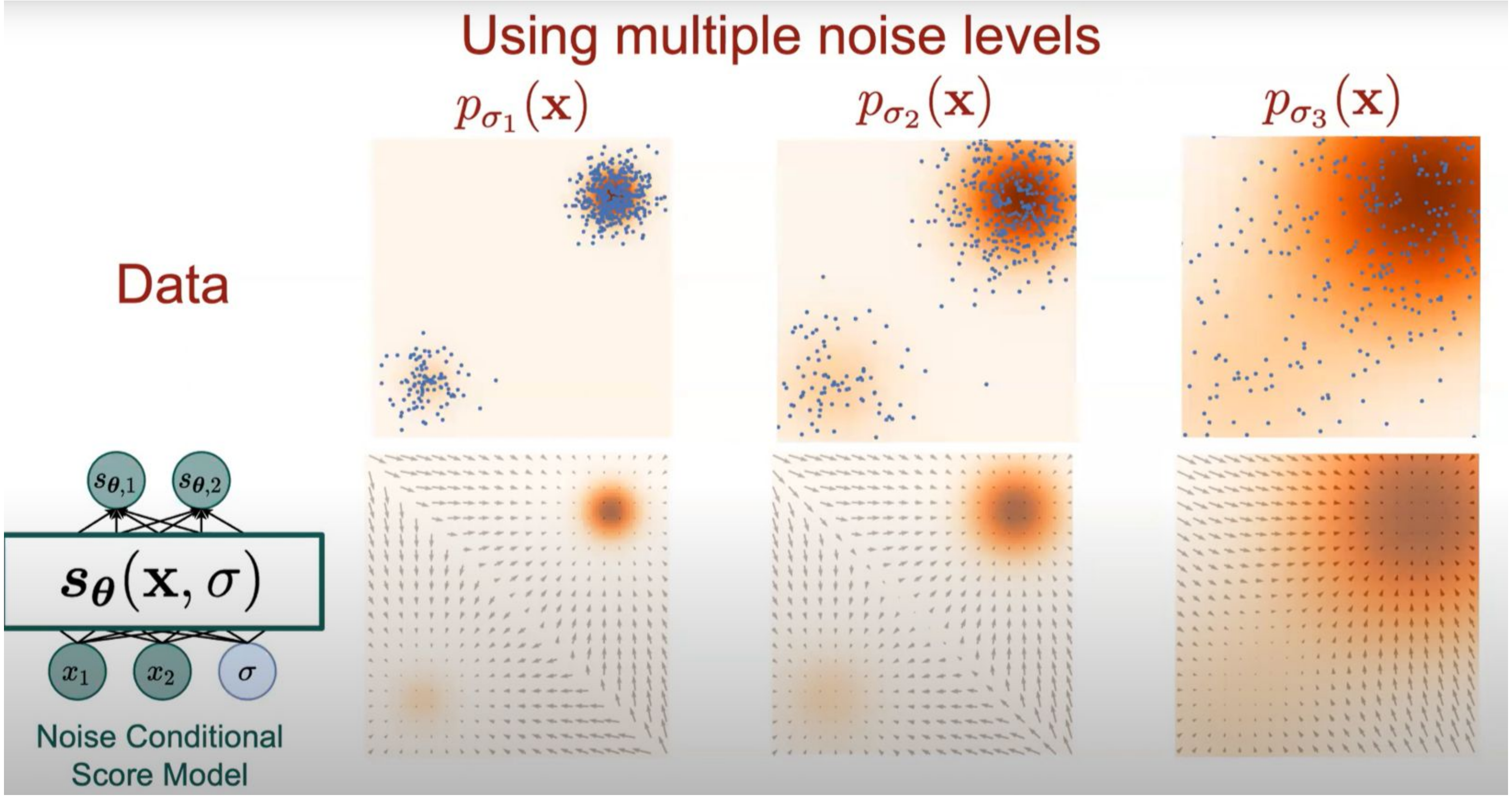
Score-based generative model



Score-based generative model



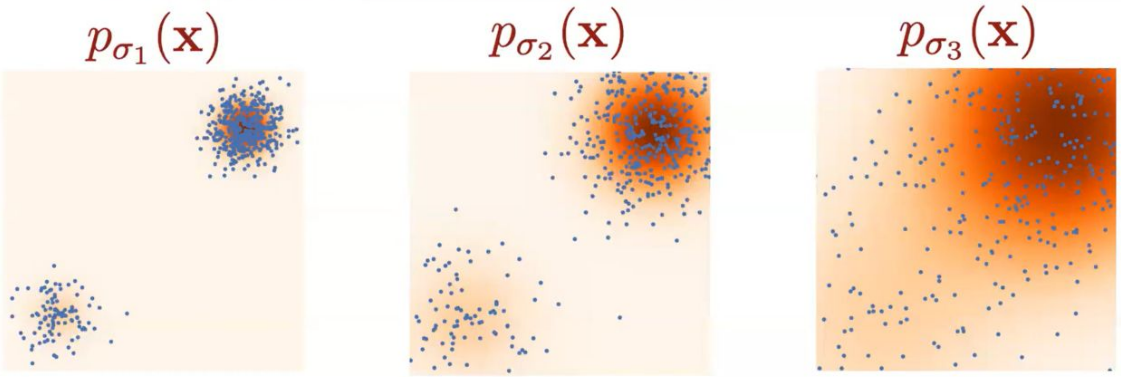
Score-based generative model

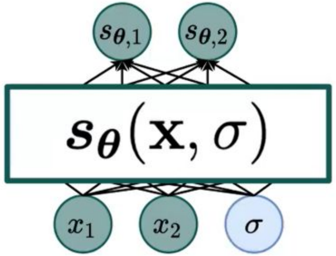


Score-based generative model

Using multiple noise levels

Data





Noise Conditional Score Model

$$\frac{1}{N}\sum_{i=1}^{N}\overbrace{\lambda(\sigma_i)}^{\text{Positive weighting function}}\underbrace{\mathbb{E}_{p_{\sigma_i}(\mathbf{x})}[\|\nabla_{\mathbf{x}}\log p_{\sigma_i}(\mathbf{x}) - \boldsymbol{s}_\theta(\mathbf{x},\sigma_i)\|_2^2]}_{\text{Score matching loss}}$$

Score-based generative model

$$J_{\mathrm{D}}(\theta) = \mathbb{E}_{p^{\sigma}_{\mathrm{data}}(\tilde{\mathbf{x}})}\left[\left|\left| \mathbf{s}_\theta(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log p^{\sigma}_{\mathrm{data}}(\tilde{\mathbf{x}}) \right|\right|_2^2 \right]$$

$$J_{\mathrm{D}}(\theta) = \mathbb{E}_{p^{\sigma}_{\mathrm{data}}(\tilde{\mathbf{x}},\mathbf{x})}\left[\left|\left| \mathbf{s}_\theta(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log p^{\sigma}_{\mathcal{N}}(\tilde{\mathbf{x}}|\mathbf{x}) \right|\right|_2^2 \right]$$

$$\nabla_{\tilde{\mathbf{x}}} \log p^{\sigma}_{\mathcal{N}}(\tilde{\mathbf{x}}|\mathbf{x}) = -\frac{1}{\sigma^2}(\tilde{\mathbf{x}} - \mathbf{x})$$

Reference

https://lilianweng.github.io/posts/2021-07-11-diffusion-models/#nice

https://yang-song.github.io/blog/2021/score/