

# Deep Generative Models

## Lecture 8

Roman Isachenko



2025, Spring

## Recap of previous lecture

### Frechet Inception Distance (FID)

In case of Normal distributions  $\pi(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_\pi, \boldsymbol{\Sigma}_\pi)$ ,  
 $p(\mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ :

$$\begin{aligned}\text{FID}(\pi, p) &= W_2^2(\pi, p) = \inf_{\gamma \in \Gamma(\pi, p)} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \gamma} \|\mathbf{x} - \mathbf{y}\|^2 \\ &= \|\boldsymbol{\mu}_\pi - \boldsymbol{\mu}_p\|^2 + \text{tr} \left[ \boldsymbol{\Sigma}_\pi + \boldsymbol{\Sigma}_p - 2 \left( \boldsymbol{\Sigma}_\pi^{1/2} \boldsymbol{\Sigma}_p \boldsymbol{\Sigma}_\pi^{1/2} \right)^{1/2} \right]\end{aligned}$$

- ▶ Needs a large sample size for evaluation.
- ▶ Calculation of FID is slow.
- ▶ High dependence on the pretrained classification model.
- ▶ Uses the normality assumption!

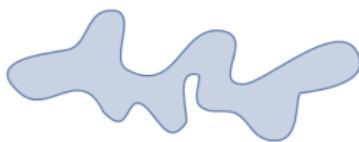
## Recap of previous lecture

- ▶  $\mathcal{S}_\pi = \{\mathbf{x}_i\}_{i=1}^n \sim \pi(\mathbf{x})$  – real samples;
- ▶  $\mathcal{S}_p = \{\mathbf{x}_i\}_{i=1}^n \sim p(\mathbf{x}|\theta)$  – generated samples.

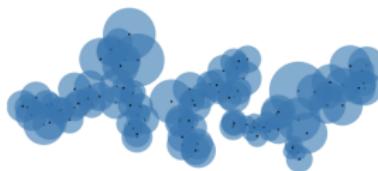
Define binary function:

$$\mathbb{I}(\mathbf{x}, \mathcal{S}) = \begin{cases} 1, & \text{if exists } \mathbf{x}' \in \mathcal{S} : \|\mathbf{x} - \mathbf{x}'\|_2 \leq \|\mathbf{x}' - \text{NN}_k(\mathbf{x}', \mathcal{S})\|_2; \\ 0, & \text{otherwise.} \end{cases}$$

$$\text{Precision}(\mathcal{S}_\pi, \mathcal{S}_p) = \frac{1}{n} \sum_{\mathbf{x} \in \mathcal{S}_p} \mathbb{I}(\mathbf{x}, \mathcal{S}_\pi); \quad \text{Recall}(\mathcal{S}_\pi, \mathcal{S}_p) = \frac{1}{n} \sum_{\mathbf{x} \in \mathcal{S}_\pi} \mathbb{I}(\mathbf{x}, \mathcal{S}_p).$$



(a) True manifold

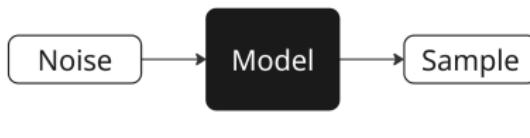


(b) Approx. manifold

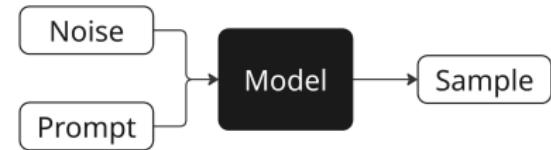
Embed the samples using the pretrained network (as for FID).

# Recap of previous lecture

## Unconditional model

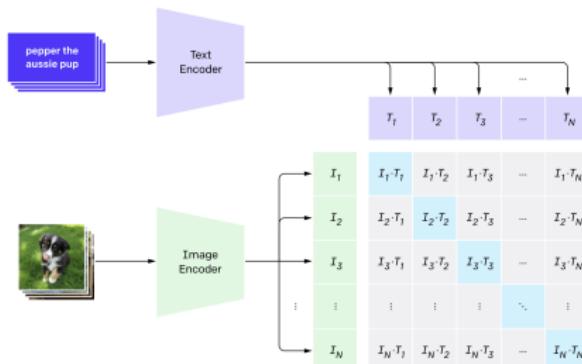


## Conditional model



We need the way to measure not only generated image quality, but also its relevance to the prompt.

## CLIP score



## Recap of previous lecture

- ▶ There is no perfect automated metric.
- ▶ The best way to evaluate the generative model is to make human evaluation.
- ▶ It is essential to evaluate different aspects.

## Human Evaluation

Аспект	Yandex ART 2.0	Mj 6.1	Mj 6	Ideogram	Recraft	Google Imagen3	Dall-E 3	FLUX	SBER Kandi3.1
Релевантность	<b>0,59</b>	<b>0,58</b>	<b>0,63</b>	<b>0,45</b>	<b>0,51</b>	<b>0,50</b>	<b>0,50</b>	<b>0,54</b>	<b>0,75</b>
Эстетика	<b>0,49</b>	<b>0,55</b>	<b>0,55</b>	<b>0,51</b>	<b>0,51</b>	<b>0,61</b>	<b>0,61</b>	<b>0,54</b>	<b>0,59</b>
Комплексность	<b>0,44</b>	<b>0,73</b>	<b>0,70</b>	<b>0,68</b>	<b>0,76</b>	<b>0,75</b>	<b>0,75</b>	<b>0,71</b>	<b>0,74</b>
Дефектность	<b>0,69</b>	<b>0,57</b>	<b>0,68</b>	<b>0,55</b>	<b>0,59</b>	<b>0,63</b>	<b>0,63</b>	<b>0,50</b>	<b>0,75</b>
Предпочтение	<b>0,66</b>	<b>0,60</b>	<b>0,69</b>	<b>0,49</b>	<b>0,54</b>	<b>0,63</b>	<b>0,63</b>	<b>0,51</b>	<b>0,84</b>

# Recap of previous lecture

## Langevin dynamic

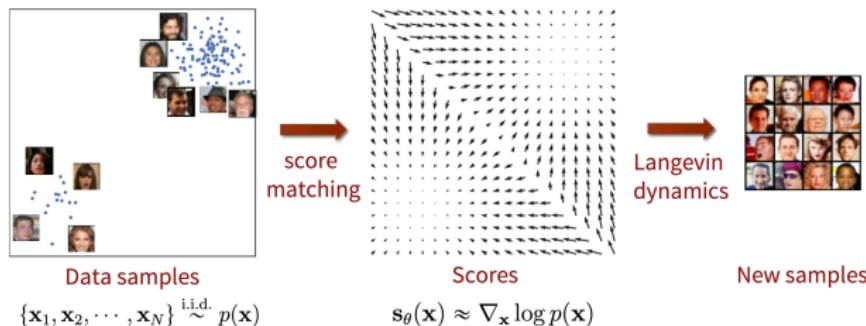
$$\mathbf{x}_{I+1} = \mathbf{x}_I + \frac{\eta}{2} \cdot \nabla_{\mathbf{x}_I} \log p(\mathbf{x}_I | \boldsymbol{\theta}) + \sqrt{\eta} \cdot \boldsymbol{\epsilon}_I, \quad \boldsymbol{\epsilon}_I \sim \mathcal{N}(0, \mathbf{I}).$$

## Fisher divergence

$$D_F(\pi, p) = \frac{1}{2} \mathbb{E}_\pi \| \nabla_{\mathbf{x}} \log p(\mathbf{x} | \boldsymbol{\theta}) - \nabla_{\mathbf{x}} \log \pi(\mathbf{x}) \|_2^2 \rightarrow \min_{\boldsymbol{\theta}}$$

## Score function

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



# Outline

## 1. Score matching

Denoising score matching

Noise Conditioned Score Network (NCSN)

## 2. Forward gaussian diffusion process

## 3. Denoising score matching for diffusion

## 4. Reverse Gaussian diffusion process

# Outline

## 1. Score matching

Denoising score matching

Noise Conditioned Score Network (NCSN)

## 2. Forward gaussian diffusion process

## 3. Denoising score matching for diffusion

## 4. Reverse Gaussian diffusion process

# Outline

## 1. Score matching

Denoising score matching

Noise Conditioned Score Network (NCSN)

## 2. Forward gaussian diffusion process

## 3. Denoising score matching for diffusion

## 4. Reverse Gaussian diffusion process

## Denoising score matching

Let perturb original data  $\mathbf{x} \sim \pi(\mathbf{x})$  by random normal noise

$$\mathbf{x}_\sigma = \mathbf{x} + \sigma \cdot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), \quad q(\mathbf{x}_\sigma | \mathbf{x}) = \mathcal{N}(\mathbf{x}, \sigma^2 \cdot \mathbf{I})$$

$$q(\mathbf{x}_\sigma) = \int q(\mathbf{x}_\sigma | \mathbf{x}) \pi(\mathbf{x}) d\mathbf{x}.$$

### Assumption

The solution of

$$\frac{1}{2} \mathbb{E}_{q(\mathbf{x}_\sigma)} \| \mathbf{s}_{\theta, \sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma) \|_2^2 \rightarrow \min_{\theta}$$

satisfies  $\mathbf{s}_{\theta, \sigma}(\mathbf{x}_\sigma) \approx \mathbf{s}_{\theta, 0}(\mathbf{x}_0) = \mathbf{s}_\theta(\mathbf{x})$  if  $\sigma$  is small enough.

- ▶ The score function of the noised data is almost the same as the score function of the original data.
- ▶ Score function  $\mathbf{s}_{\theta, \sigma}(\mathbf{x}_\sigma)$  parametrized by  $\sigma$ .
- ▶ **Note:** We don't know  $q(\mathbf{x}_\sigma)$ , just like  $\pi(\mathbf{x})$ .

# Denoising score matching

## Theorem

$$\begin{aligned}\mathbb{E}_{q(\mathbf{x}_\sigma)} \|\mathbf{s}_{\theta,\sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma)\|_2^2 &= \\ &= \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_\sigma|\mathbf{x})} \|\mathbf{s}_{\theta,\sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma|\mathbf{x})\|_2^2 + \text{const}(\theta)\end{aligned}$$

## Gradient of the noise kernel

$$\mathbf{x}_\sigma = \mathbf{x} + \sigma \cdot \boldsymbol{\epsilon}, \quad q(\mathbf{x}_\sigma|\mathbf{x}) = \mathcal{N}(\mathbf{x}, \sigma^2 \cdot \mathbf{I})$$

$$\nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma|\mathbf{x}) = -\frac{\mathbf{x}_\sigma - \mathbf{x}}{\sigma^2} = -\frac{\boldsymbol{\epsilon}}{\sigma}$$

- ▶ The RHS does not need to compute  $\nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma)$  and even  $\nabla_{\mathbf{x}_\sigma} \log \pi(\mathbf{x}_\sigma)$ .
- ▶  $\mathbf{s}_{\theta,\sigma}(\mathbf{x}_\sigma)$  tries to **denoise** the noised samples  $\mathbf{x}_\sigma$ .

# Denoising score matching

## Theorem

$$\begin{aligned} \mathbb{E}_{q(\mathbf{x}_\sigma)} \underbrace{\left\| \mathbf{s}_{\theta,\sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma) \right\|_2^2}_{h(\mathbf{x}_\sigma)} &= \\ &= \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_\sigma|\mathbf{x})} \left\| \mathbf{s}_{\theta,\sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma|\mathbf{x}) \right\|_2^2 + \text{const}(\theta) \end{aligned}$$

## Proof

$$\begin{aligned} \mathbb{E}_{q(\mathbf{x}_\sigma)} h(\mathbf{x}_\sigma) &= \int q(\mathbf{x}_\sigma) h(\mathbf{x}_\sigma) d\mathbf{x}_\sigma = \\ &= \int \left( \int q(\mathbf{x}_\sigma|\mathbf{x}) \pi(\mathbf{x}) d\mathbf{x} \right) h(\mathbf{x}_\sigma) d\mathbf{x}_\sigma = \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_\sigma|\mathbf{x})} h(\mathbf{x}_\sigma) \end{aligned}$$

$$\begin{aligned} \mathbb{E}_{q(\mathbf{x}_\sigma)} \left\| \mathbf{s}_{\theta,\sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma) \right\|_2^2 &= \\ &= \mathbb{E}_{q(\mathbf{x}_\sigma)} \left[ \left\| \mathbf{s}_{\theta,\sigma}(\mathbf{x}_\sigma) \right\|^2 + \underbrace{\left\| \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma) \right\|_2^2}_{\text{const}(\theta)} - 2 \mathbf{s}_{\theta,\sigma}^T(\mathbf{x}_\sigma) \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma) \right] \end{aligned}$$

# Denoising score matching

## Theorem

$$\begin{aligned}\mathbb{E}_{q(\mathbf{x}_\sigma)} \|\mathbf{s}_{\theta,\sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma)\|_2^2 &= \\ &= \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_\sigma|\mathbf{x})} \|\mathbf{s}_{\theta,\sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma|\mathbf{x})\|_2^2 + \text{const}(\theta)\end{aligned}$$

## Proof (continued)

$$\begin{aligned}\mathbb{E}_{q(\mathbf{x}_\sigma)} [\mathbf{s}_{\theta,\sigma}^T(\mathbf{x}_\sigma) \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma)] &= \int q(\mathbf{x}_\sigma) \left[ \mathbf{s}_{\theta,\sigma}^T(\mathbf{x}_\sigma) \frac{\nabla_{\mathbf{x}_\sigma} q(\mathbf{x}_\sigma)}{q(\mathbf{x}_\sigma)} \right] d\mathbf{x}_\sigma = \\ &= \int \left[ \mathbf{s}_{\theta,\sigma}^T(\mathbf{x}_\sigma) \nabla_{\mathbf{x}_\sigma} \left( \int q(\mathbf{x}_\sigma|\mathbf{x}) \pi(\mathbf{x}) d\mathbf{x} \right) \right] d\mathbf{x}_\sigma = \\ &= \int \int \pi(\mathbf{x}) [\mathbf{s}_{\theta,\sigma}^T(\mathbf{x}_\sigma) \nabla_{\mathbf{x}_\sigma} q(\mathbf{x}_\sigma|\mathbf{x})] d\mathbf{x}_\sigma d\mathbf{x} = \\ &= \int \int \pi(\mathbf{x}) q(\mathbf{x}_\sigma|\mathbf{x}) [\mathbf{s}_{\theta,\sigma}^T(\mathbf{x}_\sigma) \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma|\mathbf{x})] d\mathbf{x}_\sigma d\mathbf{x} = \\ &= \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_\sigma|\mathbf{x})} [\mathbf{s}_{\theta,\sigma}^T(\mathbf{x}_\sigma) \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma|\mathbf{x})]\end{aligned}$$

# Denoising score matching

## Theorem

$$\begin{aligned} \mathbb{E}_{q(\mathbf{x}_\sigma)} \underbrace{\left\| \mathbf{s}_{\theta, \sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma) \right\|_2^2}_{h(\mathbf{x}_\sigma)} &= \\ &= \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_\sigma | \mathbf{x})} \left\| \mathbf{s}_{\theta, \sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma | \mathbf{x}) \right\|_2^2 + \text{const}(\theta) \end{aligned}$$

## Proof (continued)

$$\mathbb{E}_{q(\mathbf{x}_\sigma)} h(\mathbf{x}_\sigma) = \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_\sigma | \mathbf{x})} h(\mathbf{x}_\sigma) d\mathbf{x}_\sigma$$

$$\mathbb{E}_{q(\mathbf{x}_\sigma)} \left[ \mathbf{s}_{\theta, \sigma}^T(\mathbf{x}_\sigma) \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma) \right] = \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_\sigma | \mathbf{x})} \left[ \mathbf{s}_{\theta, \sigma}^T(\mathbf{x}_\sigma) \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma | \mathbf{x}) \right]$$

$$\begin{aligned} \mathbb{E}_{q(\mathbf{x}_\sigma)} \left\| \mathbf{s}_{\theta, \sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma) \right\|_2^2 &= \\ &= \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_\sigma | \mathbf{x})} \left[ \left\| \mathbf{s}_{\theta, \sigma}(\mathbf{x}_\sigma) \right\|^2 - 2 \mathbf{s}_{\theta, \sigma}^T(\mathbf{x}_\sigma) \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma | \mathbf{x}) \right] + \text{const}(\theta) \\ &= \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_\sigma | \mathbf{x})} \left\| \mathbf{s}_{\theta, \sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma | \mathbf{x}) \right\|_2^2 + \text{const}(\theta) \end{aligned}$$

# Denoising score matching

Initial objective:

$$\mathbb{E}_{\pi(\mathbf{x})} \left\| \mathbf{s}_\theta(\mathbf{x}) - \nabla_{\mathbf{x}} \log \pi(\mathbf{x}) \right\|_2^2 \rightarrow \min_{\theta}$$

Noised objective:

$$\mathbb{E}_{q(\mathbf{x}_\sigma)} \left\| \mathbf{s}_{\theta,\sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}} \log q(\mathbf{x}_\sigma) \right\|_2^2 \rightarrow \min_{\theta}$$

This is equivalent to denoising task

$$\mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_\sigma|\mathbf{x})} \left\| \mathbf{s}_{\theta,\sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma|\mathbf{x}) \right\|_2^2 \rightarrow \min_{\theta}$$

$$\mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{\mathcal{N}(0, \mathbf{I})} \left\| \mathbf{s}_{\theta,\sigma}(\mathbf{x} + \sigma \cdot \boldsymbol{\epsilon}) + \frac{\boldsymbol{\epsilon}}{\sigma} \right\|_2^2 \rightarrow \min_{\theta}$$

## Langevin dynamic

$$\mathbf{x}_{I+1} = \mathbf{x}_I + \frac{\eta}{2} \cdot \mathbf{s}_{\theta,\sigma}(\mathbf{x}_I) + \sqrt{\eta} \cdot \boldsymbol{\epsilon}_I, \quad \boldsymbol{\epsilon}_I \sim \mathcal{N}(0, \mathbf{I}).$$

# Outline

## 1. Score matching

Denoising score matching

Noise Conditioned Score Network (NCSN)

## 2. Forward gaussian diffusion process

## 3. Denoising score matching for diffusion

## 4. Reverse Gaussian diffusion process

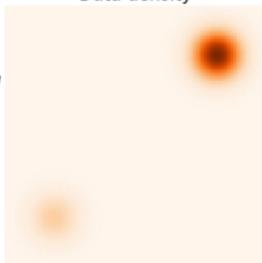
# Denoising score matching

$$\mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{\mathcal{N}(0, \mathbf{I})} \left\| \mathbf{s}_{\theta, \sigma}(\mathbf{x} + \sigma \cdot \boldsymbol{\epsilon}) + \frac{\boldsymbol{\epsilon}}{\sigma} \right\|_2^2 \rightarrow \min_{\theta}$$

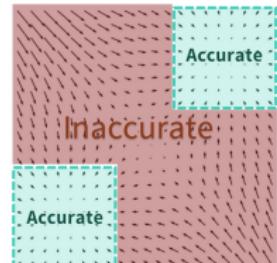
$$\mathbf{x}_{l+1} = \mathbf{x}_l + \frac{\eta}{2} \cdot \mathbf{s}_{\theta, \sigma}(\mathbf{x}_l) + \sqrt{\eta} \cdot \boldsymbol{\epsilon}_l$$

- ▶ If  $\sigma$  is **small**,  $\mathbf{s}_{\theta, \sigma}(\mathbf{x})$  is not accurate and Langevin dynamics will fail to jump between modes.
- ▶ If  $\sigma$  is **large**, it is good for low-density regions, but we will learn too corrupted distribution.

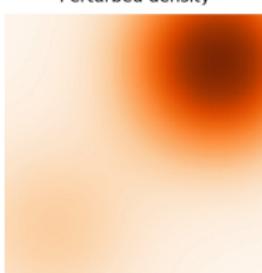
Data density



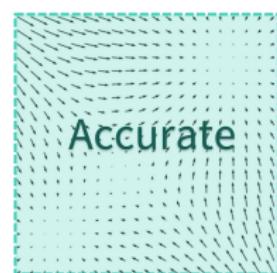
Estimated scores



Perturbed density



Estimated scores

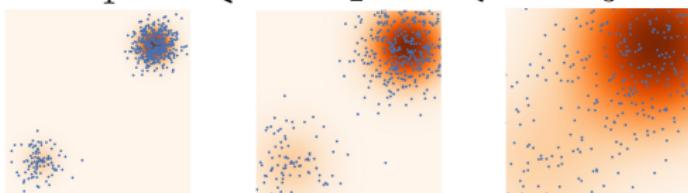


# Noise Conditioned Score Network (NCSN)

- ▶ Define the sequence of the noise levels:  $\sigma_1 < \sigma_2 < \dots < \sigma_T$ .
- ▶ Perturb the original data with the different noise levels to obtain  $\mathbf{x}_t = \mathbf{x} + \sigma_t \cdot \epsilon$ ,  $\mathbf{x}_t \sim q(\mathbf{x}_t)$ .
- ▶ Choose  $\sigma_1, \sigma_T$  such that:

$$q(\mathbf{x}_1) \approx \pi(\mathbf{x}), \quad q(\mathbf{x}_T) \approx \mathcal{N}(0, \sigma_T^2 \cdot \mathbf{I}).$$

$$\sigma_1 \quad < \quad \sigma_2 \quad < \quad \sigma_3$$



# Noise Conditioned Score Network (NCSN)

Train the denoising score function  $s_{\theta, \sigma_t}(\mathbf{x}_t)$  for each noise level  $\sigma_t$  using unified weighted objective:

$$\sum_{t=1}^T \sigma_t^2 \cdot \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})} \|s_{\theta, \sigma_t}(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{x})\|_2^2 \rightarrow \min_{\theta}$$

Here  $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{x}) = -\frac{\mathbf{x}_t - \mathbf{x}}{\sigma_t^2} = -\frac{\epsilon}{\sigma_t}$ .

## Training

1. Get the sample  $\mathbf{x}_0 \sim \pi(\mathbf{x})$ .
2. Sample noise level  $t \sim U\{1, T\}$  and the noise  $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ .
3. Get noisy image  $\mathbf{x}_t = \mathbf{x}_0 + \sigma_t \cdot \epsilon$ .
4. Compute loss  $\mathcal{L} = \sigma_t^2 \cdot \|s_{\theta, \sigma_t}(\mathbf{x}_t) + \frac{\epsilon}{\sigma_t}\|^2$ .

How to sample from this model?

# Noise Conditioned Score Network (NCSN)

## Sampling (annealed Langevin dynamics)

- ▶ Sample  $\mathbf{x}_0 \sim \mathcal{N}(0, \sigma_T^2 \cdot \mathbf{I}) \approx q(\mathbf{x}_T)$ .
- ▶ Apply  $L$  steps of Langevin dynamic

$$\mathbf{x}_l = \mathbf{x}_{l-1} + \frac{\eta_t}{2} \cdot \mathbf{s}_{\theta, \sigma_t}(\mathbf{x}_{l-1}) + \sqrt{\eta_t} \cdot \boldsymbol{\epsilon}_l.$$

- ▶ Update  $\mathbf{x}_0 := \mathbf{x}_L$  and choose the next  $\sigma_t$ .



# Outline

## 1. Score matching

Denoising score matching

Noise Conditioned Score Network (NCSN)

## 2. Forward gaussian diffusion process

## 3. Denoising score matching for diffusion

## 4. Reverse Gaussian diffusion process

## Forward gaussian diffusion process

Let  $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$ ,  $\beta_t \ll 1$ . Define the Markov chain

$$\mathbf{x}_t = \sqrt{1 - \beta_t} \cdot \mathbf{x}_{t-1} + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon}_t, \quad \text{where } \boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I});$$

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

## Langevin dynamics

$$\mathbf{x}_{I+1} = \mathbf{x}_I + \frac{\eta}{2} \cdot \nabla_{\mathbf{x}_I} \log p(\mathbf{x}_I | \boldsymbol{\theta}) + \sqrt{\eta} \cdot \boldsymbol{\epsilon}_I, \quad \boldsymbol{\epsilon}_I \sim \mathcal{N}(0, \mathbf{I}).$$

$$\begin{aligned} \mathbf{x}_t &= \sqrt{1 - \beta_t} \cdot \mathbf{x}_{t-1} + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon}_t \approx \left(1 - \frac{\beta_t}{2}\right) \cdot \mathbf{x}_{t-1} + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon}_t = \\ &= \mathbf{x}_{t-1} + \frac{\beta_t}{2} \cdot (-\mathbf{x}_{t-1}) + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon}_t \end{aligned}$$

►  $\beta_t = \eta$

►  $\nabla_{\mathbf{x}_{t-1}} \log p(\mathbf{x}_{t-1} | \boldsymbol{\theta}) = -\mathbf{x}_{t-1} = \nabla_{\mathbf{x}_{t-1}} \log \mathcal{N}(0, \mathbf{I})$

## Forward gaussian diffusion process

$$\mathbf{x}_t = \sqrt{1 - \beta_t} \cdot \mathbf{x}_{t-1} + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon}_t, \quad \text{where } \boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I});$$
$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\sqrt{1 - \beta_t} \cdot \mathbf{x}_{t-1}, \beta_t \cdot \mathbf{I}).$$

### Statement 1

Let denote  $\alpha_t = 1 - \beta_t$  and  $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s = \prod_{s=1}^t (1 - \beta_s)$ . Then

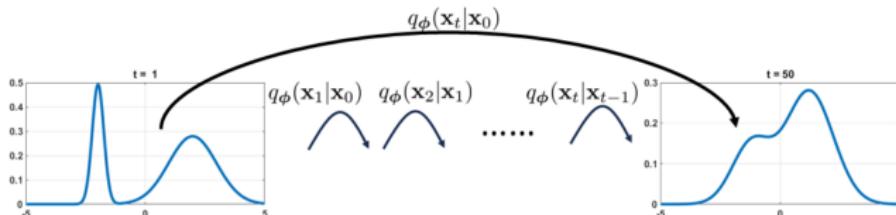
$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0, (1 - \bar{\alpha}_t) \cdot \mathbf{I})$$

We are able to sample from any timestamp using only  $\mathbf{x}_0$ !

$$\begin{aligned} \mathbf{x}_t &= \sqrt{\alpha_t} \cdot \mathbf{x}_{t-1} + \sqrt{1 - \alpha_t} \cdot \boldsymbol{\epsilon}_t = \\ &= \sqrt{\alpha_t} (\sqrt{\alpha_{t-1}} \cdot \mathbf{x}_{t-2} + \sqrt{1 - \alpha_{t-1}} \cdot \boldsymbol{\epsilon}_{t-1}) + \sqrt{1 - \alpha_t} \cdot \boldsymbol{\epsilon}_t = \\ &= \sqrt{\alpha_t \alpha_{t-1}} \cdot \mathbf{x}_{t-2} + (\sqrt{\alpha_t (1 - \alpha_{t-1})} \cdot \boldsymbol{\epsilon}_{t-1} + \sqrt{1 - \alpha_t} \cdot \boldsymbol{\epsilon}_t) = \\ &= \sqrt{\alpha_t \alpha_{t-1}} \cdot \mathbf{x}_{t-2} + \sqrt{1 - \alpha_t \alpha_{t-1}} \cdot \boldsymbol{\epsilon}'_t = \\ &= \dots = \sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \cdot \boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}). \end{aligned}$$

# Forward gaussian diffusion process

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N} \left( \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I} \right); \quad q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N} \left( \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I} \right).$$



## Statement 2

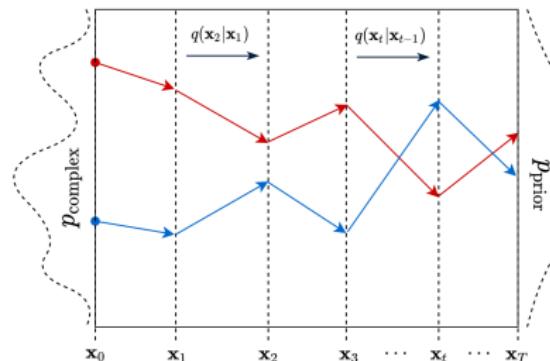
Applying the Markov chain to samples from any  $\pi(\mathbf{x})$  we will get  $\mathbf{x}_\infty \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, \mathbf{I})$ . Here  $p_\infty(\mathbf{x})$  is a **stationary** and **limiting** distribution:

$$p_\infty(\mathbf{x}) = \int q(\mathbf{x} | \mathbf{x}') p_\infty(\mathbf{x}') d\mathbf{x}'.$$

$$p_\infty(\mathbf{x}) = \int q(\mathbf{x}_\infty | \mathbf{x}_0) \pi(\mathbf{x}_0) d\mathbf{x}_0 \approx \mathcal{N}(0, \mathbf{I}) \int \pi(\mathbf{x}_0) d\mathbf{x}_0 = \mathcal{N}(0, \mathbf{I})$$

# Forward gaussian diffusion process

**Diffusion** refers to the flow of particles from high-density regions towards low-density regions.



1.  $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$ ;
2.  $\mathbf{x}_t = \sqrt{1 - \beta_t} \cdot \mathbf{x}_{t-1} + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon}$ , where  $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ ,  $t \geq 1$ ;
3.  $\mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, \mathbf{I})$ , where  $T \gg 1$ .

If we are able to invert this process, we will get the way to sample  $\mathbf{x} \sim \pi(\mathbf{x})$  using noise samples  $p_\infty(\mathbf{x}) = \mathcal{N}(0, \mathbf{I})$ .

Now our goal is to revert this process.

# Outline

## 1. Score matching

Denoising score matching

Noise Conditioned Score Network (NCSN)

## 2. Forward gaussian diffusion process

## 3. Denoising score matching for diffusion

## 4. Reverse Gaussian diffusion process

## Denoising score matching

### NCSN

$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0, \sigma_t^2 \cdot \mathbf{I}), \quad q(\mathbf{x}_1) \approx \pi(\mathbf{x}), \quad q(\mathbf{x}_T) \approx \mathcal{N}(0, \sigma_T^2 \cdot \mathbf{I}).$$

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t | \mathbf{x}) = -\frac{\mathbf{x}_t - \mathbf{x}}{\sigma_t^2}$$

### Gaussian diffusion

$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0, (1 - \bar{\alpha}_t) \cdot \mathbf{I}), \quad q(\mathbf{x}_1) \approx \pi(\mathbf{x}), \quad q(\mathbf{x}_T) \approx \mathcal{N}(0, \mathbf{I})$$

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t | \mathbf{x}_0) = -\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0}{1 - \bar{\alpha}_t}$$

### Theorem (denoising score matching)

$$\begin{aligned} \mathbb{E}_{q(\mathbf{x}_t)} \|\mathbf{s}_{\theta,t}(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t)\|_2^2 &= \\ &= \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_t | \mathbf{x})} \|\mathbf{s}_{\theta,t}(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t | \mathbf{x})\|_2^2 + \text{const}(\theta) \end{aligned}$$

**Note:** We are able to apply NCSN approach with annealed Langevin dynamics to get diffusion denoising model.

# Outline

## 1. Score matching

Denoising score matching

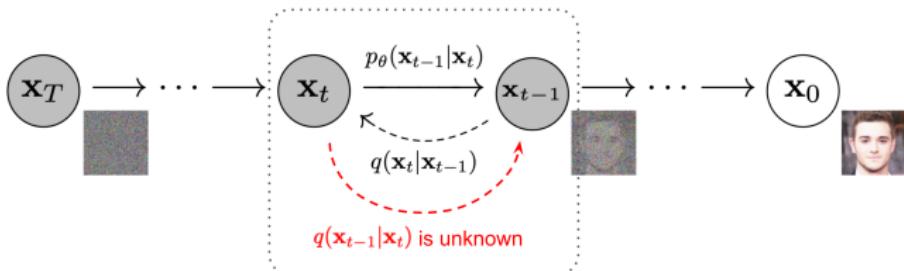
Noise Conditioned Score Network (NCSN)

## 2. Forward gaussian diffusion process

## 3. Denoising score matching for diffusion

## 4. Reverse Gaussian diffusion process

# Reverse Gaussian diffusion process



Forward process

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

Reverse process

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

$q(\mathbf{x}_{t-1})$ ,  $q(\mathbf{x}_t)$  are intractable:

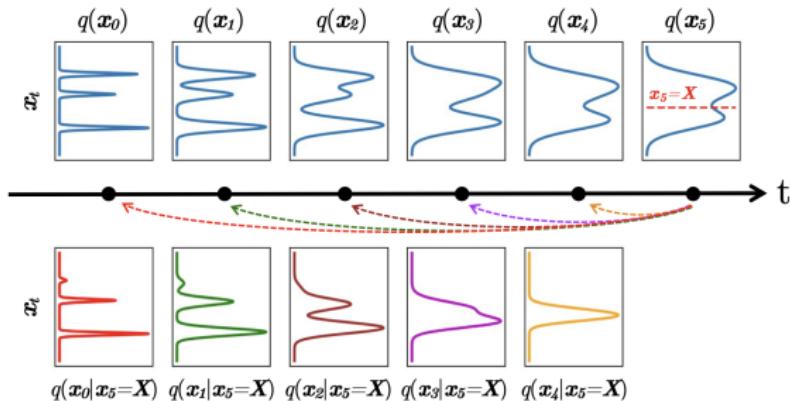
$$q(\mathbf{x}_t) = \int q(\mathbf{x}_t|\mathbf{x}_0)\pi(\mathbf{x}_0)d\mathbf{x}_0$$

# Reverse Gaussian diffusion process

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

Theorem (Feller, 1949)

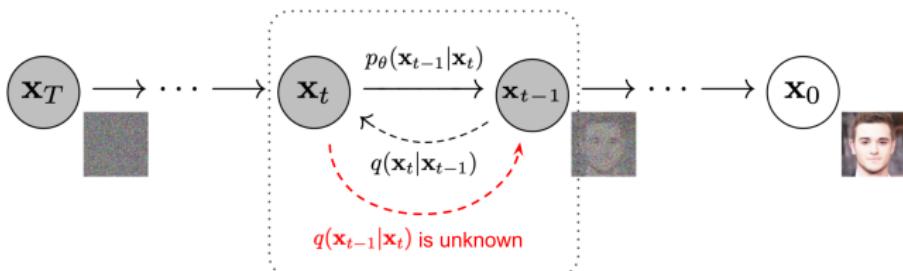
If  $\beta_t$  is small enough,  $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$  will be Gaussian (that is why diffusion needs  $T \approx 1000$  steps to converge).



Feller W. On the theory of stochastic processes, with particular reference to applications, 1949

Xiao Z., Kreis K., Vahdat A. Tackling the generative learning trilemma with denoising diffusion GANs, 2021

# Reverse Gaussian diffusion process (ancestral sampling)



Let define the reverse process:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t) \approx p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) = \mathcal{N}(\mu_{\boldsymbol{\theta}, t}(\mathbf{x}_t), \sigma_{\boldsymbol{\theta}, t}^2(\mathbf{x}_t))$$

Feller theorem shows that it is a reasonable assumption.

Forward process

$$1. \quad \mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x});$$

$$2. \quad \mathbf{x}_t = \sqrt{1 - \beta_t} \cdot \mathbf{x}_{t-1} + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon};$$

$$3. \quad \mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, \mathbf{I}).$$

Reverse process

$$1. \quad \mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, \mathbf{I});$$

$$2. \quad \mathbf{x}_{t-1} = \boldsymbol{\sigma}_{\boldsymbol{\theta}, t}(\mathbf{x}_t) \cdot \boldsymbol{\epsilon} + \mu_{\boldsymbol{\theta}, t}(\mathbf{x}_t);$$

$$3. \quad \mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x}).$$

**Note:** The forward process does not have any learnable parameters!

## Summary

- ▶ Denoising score matching minimizes the Fisher divergence on noisy samples. It allows to estimate the Fisher divergence using samples.
- ▶ Noise conditioned score network uses multiple noise levels and annealed Langevin dynamics to fit the score function and sample from the model.
- ▶ Gaussian diffusion process is a Markov chain that injects special form of Gaussian noise to the samples.
- ▶ Denoising score matching with Langevin dynamics is applicable to Gaussian diffusion process.
- ▶ Reverse process allows to recover the real samples from the noise, but it is intractable.