

# Deep Generative Models

## Lecture 14

Roman Isachenko



Autumn, 2022

## Recap of previous lecture

### Continuous normalizing flows

$$\frac{d \log p(\mathbf{z}(t), t)}{dt} = -\text{tr} \left( \frac{\partial f(\mathbf{z}(t), \boldsymbol{\theta})}{\partial \mathbf{z}(t)} \right).$$

Forward transform + log-density

$$\begin{bmatrix} \mathbf{x} \\ \log p(\mathbf{x}|\boldsymbol{\theta}) \end{bmatrix} = \begin{bmatrix} \mathbf{z} \\ \log p(\mathbf{z}) \end{bmatrix} + \int_{t_0}^{t_1} \begin{bmatrix} f(\mathbf{z}(t), \boldsymbol{\theta}) \\ -\text{tr} \left( \frac{\partial f(\mathbf{z}(t), \boldsymbol{\theta})}{\partial \mathbf{z}(t)} \right) \end{bmatrix} dt.$$

Hutchinson's trace estimator

$$\begin{aligned} \log p(\mathbf{z}(t_1)) &= \log p(\mathbf{z}(t_0)) - \int_{t_0}^{t_1} \text{tr} \left( \frac{\partial f(\mathbf{z}(t), \boldsymbol{\theta})}{\partial \mathbf{z}(t)} \right) dt = \\ &= \log p(\mathbf{z}(t_0)) - \mathbb{E}_{p(\epsilon)} \int_{t_0}^{t_1} \left[ \epsilon^T \frac{\partial f}{\partial \mathbf{z}} \epsilon \right] dt. \end{aligned}$$

## Recap of previous lecture

### SDE basics

Let define stochastic process  $\mathbf{x}(t)$  with initial condition  $\mathbf{x}(0) \sim p_0(\mathbf{x})$ :

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w},$$

where  $\mathbf{w}(t)$  is the standard Wiener process (Brownian motion)

$$\mathbf{w}(t) - \mathbf{w}(s) \sim \mathcal{N}(0, (t-s)\mathbf{I}), \quad d\mathbf{w} = \boldsymbol{\epsilon} \cdot \sqrt{dt}, \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}).$$

### Langevin dynamics

Let  $\mathbf{x}_0$  be a random vector. Then under mild regularity conditions for small enough  $\eta$  samples from the following dynamics

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

will come from  $p(\mathbf{x} | \theta)$ .

The density  $p(\mathbf{x} | \theta)$  is a **stationary** distribution for the Langevin SDE.

# Outline

1. Noise conditioned score network

2. Diffusion models

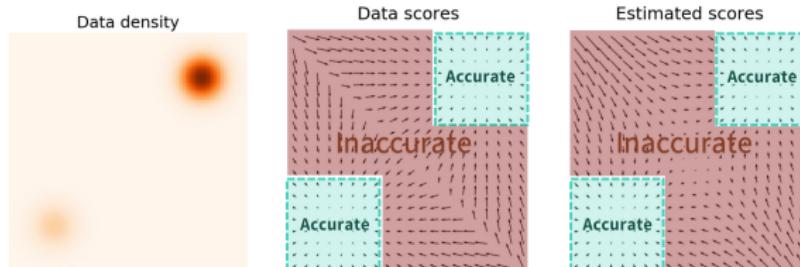
# Outline

1. Noise conditioned score network

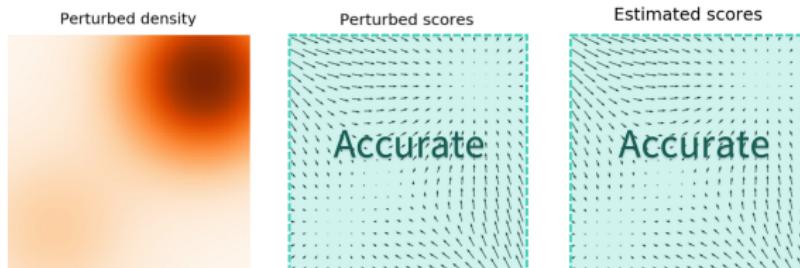
2. Diffusion models

# Denoising score matching

- If  $\sigma$  is **small**, the score function is not accurate and Langevin dynamics will probably fail to jump between modes.



- If  $\sigma$  is **large**, it is good for low-density regions and multimodal distributions, but we will learn too corrupted distribution.

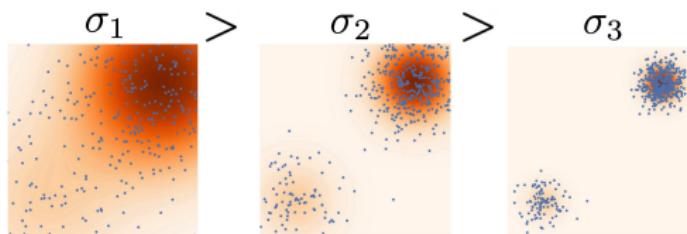


## Noise conditioned score network

- ▶ Define the sequence of noise levels:  $\sigma_1 > \sigma_2 > \dots > \sigma_L$ .
- ▶ Perturb the original data with the different noise level to get  $\pi(\mathbf{x}|\sigma_1), \dots, \pi(\mathbf{x}|\sigma_L)$ .
- ▶ Train denoised score function  $\mathbf{s}(\mathbf{x}, \theta, \sigma)$  for each noise level:

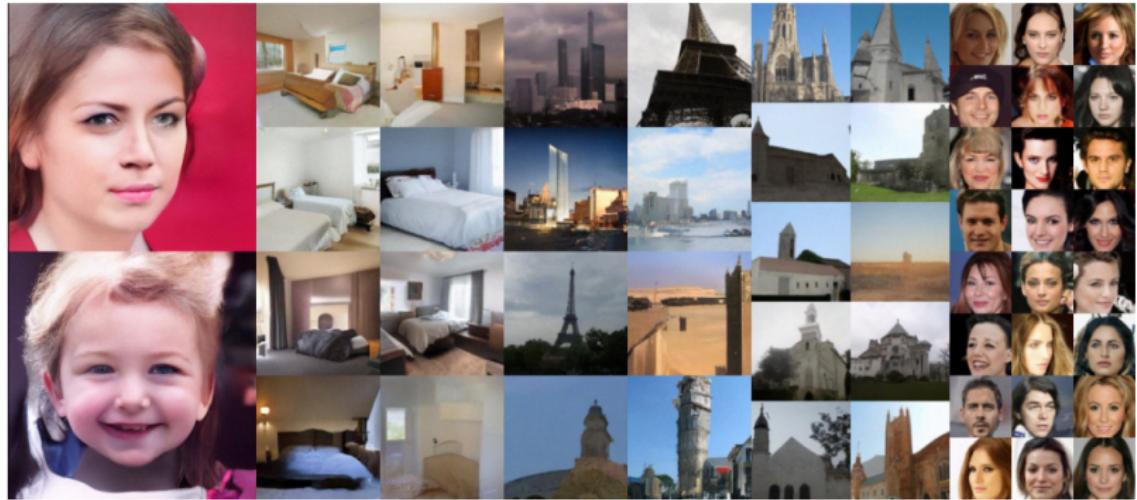
$$\sum_{l=1}^L \sigma_l^2 \mathbb{E}_{\pi(\mathbf{x}')} \mathbb{E}_{p(\mathbf{x}|\mathbf{x}', \sigma_l)} \| \mathbf{s}(\mathbf{x}, \theta, \sigma_l) - \nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{x}', \sigma_l) \|_2^2 \rightarrow \min_{\theta}$$

- ▶ Sample from **annealed** Langevin dynamics (for  $l = 1, \dots, L$ ).



# Noise conditioned score network

## Samples



# Outline

1. Noise conditioned score network

2. Diffusion models

## Forward diffusion process

Let  $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$ ,  $\beta \in (0, 1)$ . Define the Markov chain

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

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

### Statement

Applying the Markov chain to samples from any  $\pi(\mathbf{x})$  we will get  $\mathbf{x}_\infty \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, 1)$ . Here  $p_\infty(\mathbf{x})$  is a **stationary** distribution:

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

### Statement

Denote  $\alpha_t = 1 - \beta_t$ ,  $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$ . Then

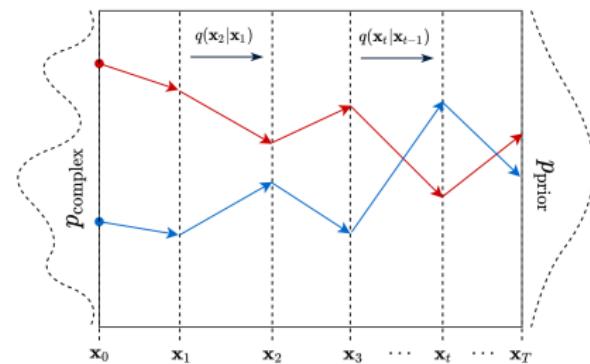
$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$$

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

We could sample from any timestamp using only  $\mathbf{x}_0$ .

# Forward diffusion process

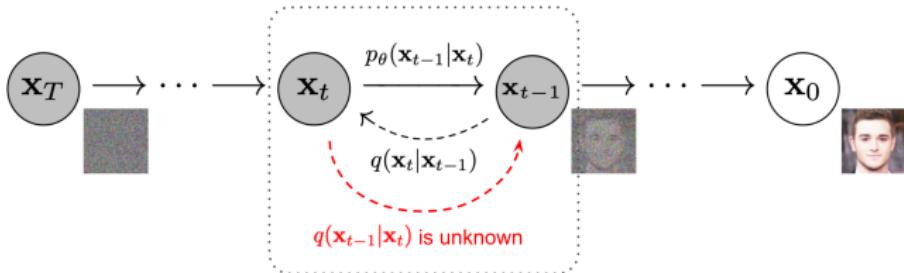
**Diffusion** refers to the flow of particles from high-density regions towards low-density regions.



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

Now our goal is to revert this process.

# Reverse diffusion process



Let define the reverse process

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \theta) = \mathcal{N}(\mathbf{x}_{t-1}|\mu(\mathbf{x}_t, \theta, t), \sigma^2(\mathbf{x}_t, \theta, t))$$

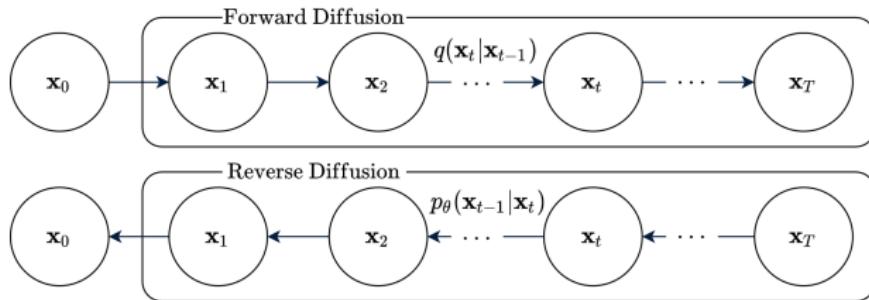
Forward process

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

Reverse process

1.  $\mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, 1);$
2.  $\mathbf{x}_{t-1} = \sigma(\mathbf{x}_t, \theta, t) \cdot \mathbf{x}_t + \mu(\mathbf{x}_t, \theta, t);$
3.  $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x});$

# Diffusion model



- ▶ Let treat  $\mathbf{z} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$  as a latent variable.
- ▶ Variational posterior distribution

$$q(\mathbf{z} | \mathbf{x}) = q(\mathbf{x}_1, \dots, \mathbf{x}_T | \mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t | \mathbf{x}_{t-1}).$$

- ▶ Probabilistic model

$$p(\mathbf{x}, \mathbf{z} | \boldsymbol{\theta}) = p(\mathbf{x} | \mathbf{z}, \boldsymbol{\theta})p(\mathbf{z} | \boldsymbol{\theta})$$

- ▶ Generative distribution and prior

$$p(\mathbf{x} | \mathbf{z}, \boldsymbol{\theta}) = p(\mathbf{x}_0 | \mathbf{x}_1, \boldsymbol{\theta}); \quad p(\mathbf{z} | \boldsymbol{\theta}) = \prod_{t=2}^T p(\mathbf{x}_{t-1} | \mathbf{x}_t, \boldsymbol{\theta})$$

# Diffusion model

## ELBO

$$\log p(\mathbf{x}|\theta) \geq \int q(\mathbf{z}|\mathbf{x}) \frac{p(\mathbf{x}, \mathbf{z}|\theta)}{q(\mathbf{z}|\mathbf{x})} d\mathbf{z} = \mathcal{L}(q, \theta) \rightarrow \max_{q, \theta}$$

## Statement

$$\begin{aligned}\mathcal{L}(q, \theta) &= \mathbb{E}_{q(\mathbf{x}_1, \dots, \mathbf{x}_T | \mathbf{x}_0)} \frac{p(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_T | \theta)}{q(\mathbf{x}_1, \dots, \mathbf{x}_T | \mathbf{x}_0)} = \\ &= \mathbb{E}_q \left[ \underbrace{KL(q(\mathbf{x}_T | \mathbf{x}_0) || p(\mathbf{x}_T))}_{\mathcal{L}_0} + \sum_{t=2}^T \underbrace{KL(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) || p(\mathbf{x}_{t-1} | \mathbf{x}_t, \theta))}_{\mathcal{L}_t} - \right. \\ &\quad \left. - \log p(\mathbf{x}_0 | \mathbf{x}_1, \theta) \right]\end{aligned}$$

- ▶ First term is constant (KL between two standard normals).
- ▶ Third term is a decoder distribution (could be AR model or discretized distribution (like mixture of logistics)).

# Diffusion model

$$\mathcal{L}_t = KL(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) || p(\mathbf{x}_{t-1} | \mathbf{x}_t, \theta)),$$

Here

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

$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0)$  and  $\tilde{\beta}_t$  have analytical formulas (we omit it) and both dependent on  $\beta_t$ .

- ▶ Assume  $\sigma^2(\mathbf{x}_t, \theta, t) = \tilde{\beta}_t \mathbf{I}$  (reminder:  
 $p(\mathbf{x}_{t-1} | \mathbf{x}_t, \theta) = \mathcal{N}(\mathbf{x}_{t-1} | \mu(\mathbf{x}_t, \theta, t), \sigma^2(\mathbf{x}_t, \theta, t))$ ).
- ▶ Use KL formula for normal distributions.
- ▶ Use the fact  $\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}$

$$\begin{aligned}\mathcal{L}_t &= \mathbb{E}_{\boldsymbol{\epsilon}} \left[ \frac{1}{2\tilde{\beta}_t} \| \tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) - \mu(\mathbf{x}_t, \theta, t) \|^2 \right] = \\ &= \mathbb{E}_{\boldsymbol{\epsilon}} \left[ \frac{1}{2\tilde{\beta}_t} \left\| \frac{1}{\sqrt{\bar{\alpha}_t}} \left( \mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon} \right) - \mu(\mathbf{x}_t, \theta, t) \right\|^2 \right]\end{aligned}$$

# Diffusion model

## Reparametrization

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

## KL term

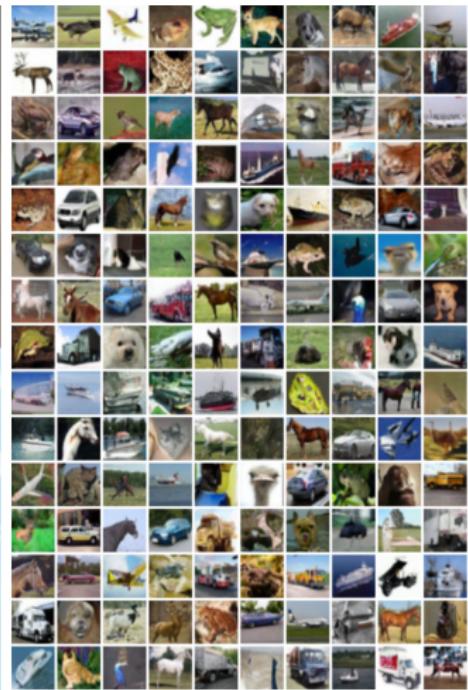
$$\begin{aligned} \mathcal{L}_t &= \mathbb{E}_{\epsilon} \left[ \frac{1}{2\tilde{\beta}_t} \left\| \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon \right) - \mu(\mathbf{x}_t, \boldsymbol{\theta}, t) \right\|^2 \right] \\ &= \mathbb{E}_{\epsilon} \left[ \frac{\beta_t^2}{2\tilde{\beta}_t \alpha_t (1 - \bar{\alpha}_t)} \|\epsilon - \epsilon(\mathbf{x}_t, \boldsymbol{\theta}, t)\|^2 \right] \end{aligned}$$

## Noise conditioned score network

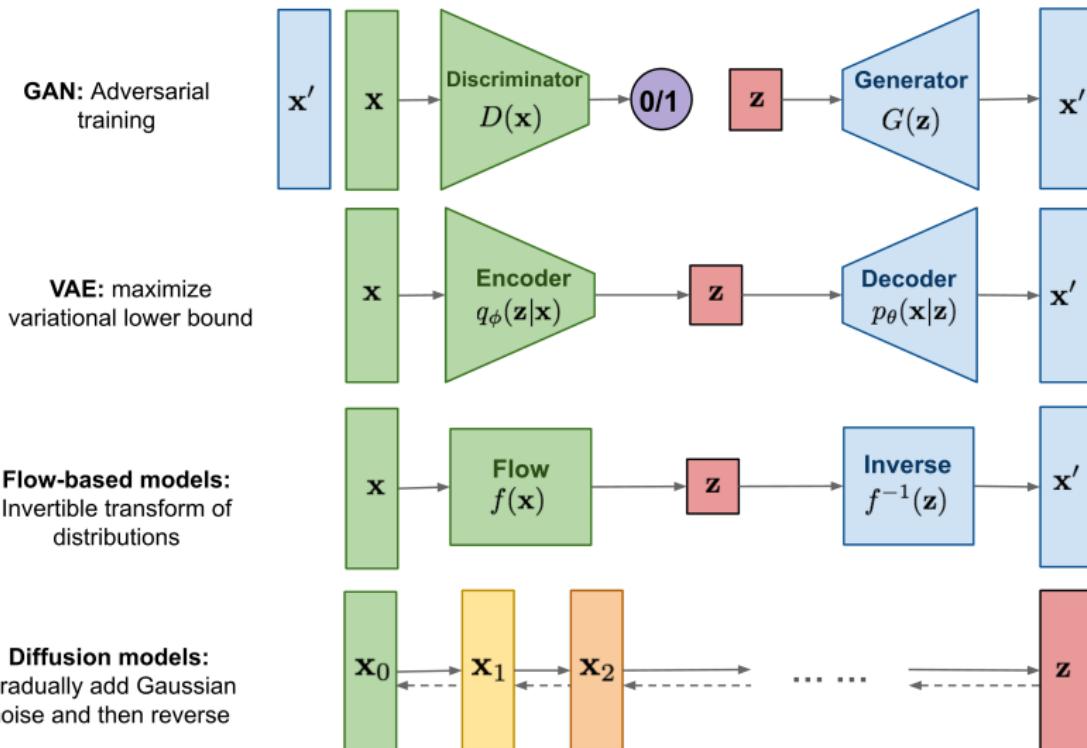
$$\mathbb{E}_{p(\mathbf{x}|\mathbf{x}', \sigma_I)} \|\mathbf{s}(\mathbf{x}, \boldsymbol{\theta}, \sigma_I) - \nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{x}', \sigma_I)\|_2^2 \rightarrow \min_{\boldsymbol{\theta}}$$

# Denoising diffusion probabilistic model

## Samples



# The poorest course overview :)



## Summary

- ▶ Noise conditioned score network uses multiple noise levels and annealed Langevin dynamics to fit score function.
- ▶ Gaussian diffusion process is a Markov chain that inject Gaussian noise.
- ▶ Diffusion model is a VAE model which revert gaussian diffusion process using variational inference.
- ▶ Objective of diffusion model is closely related to the noise conditioned score network.