# Denoising Diffusion Probabilistic Models

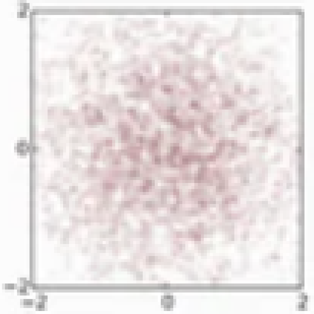
Jonathan Ho, Ajay Jain, Pieter Abbeel

BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

NeurIPS 2020

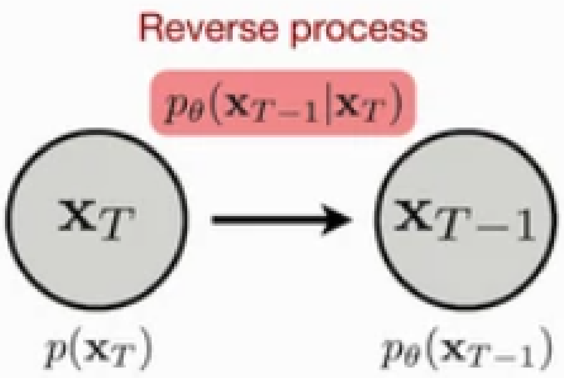
# Diffusion probabilistic models (Sohl-Dickstein et al 2015)

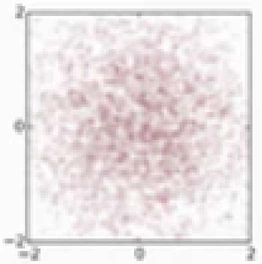

$p(\mathbf{x}_T)$



Noise distribution

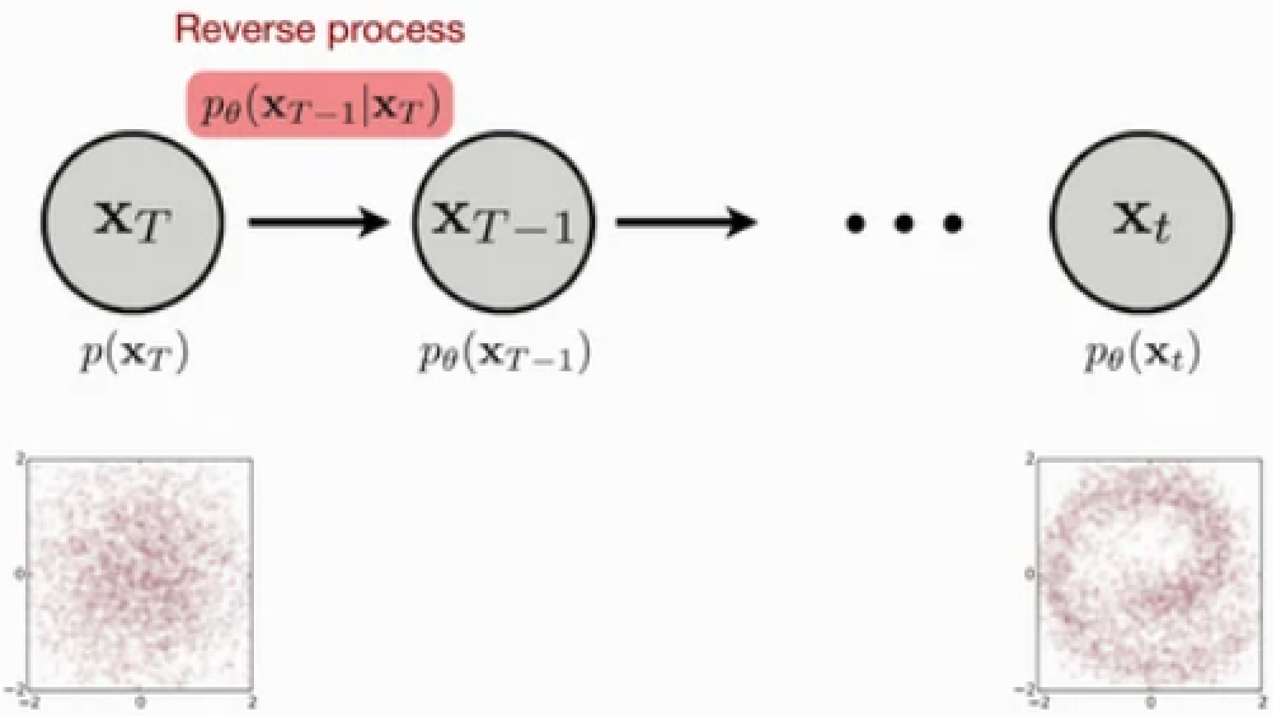
# Diffusion probabilistic models (Sohl-Dickstein et al 2015)



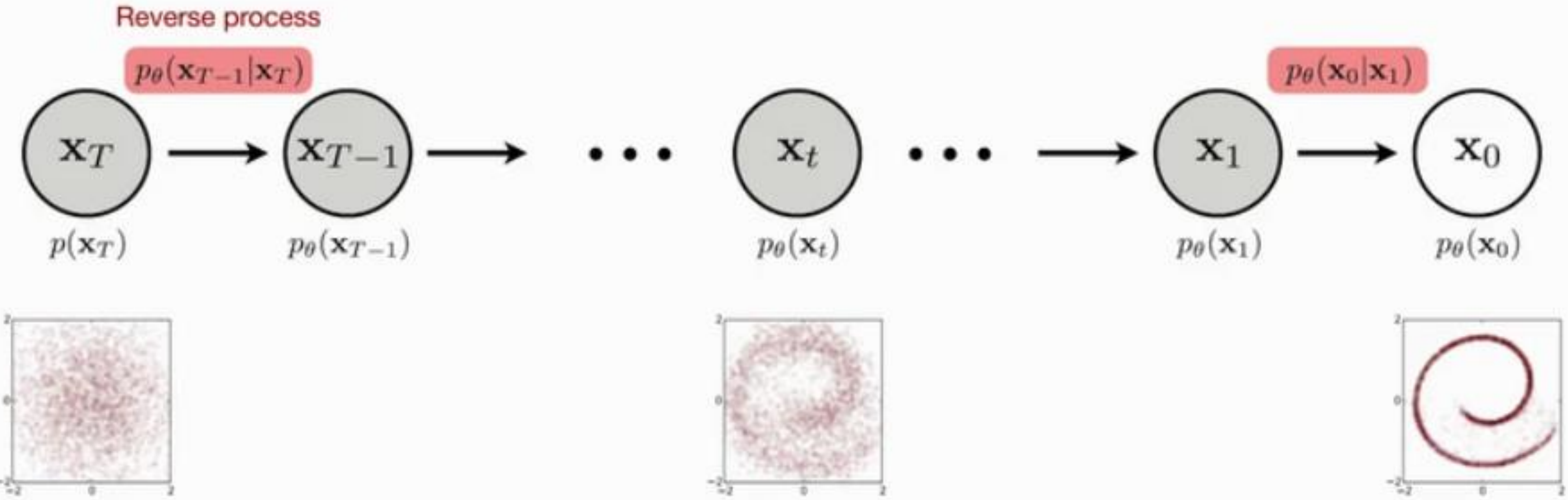
Noise distribution

# Diffusion probabilistic models (Sohl-Dickstein et al 2015)




Noise distribution

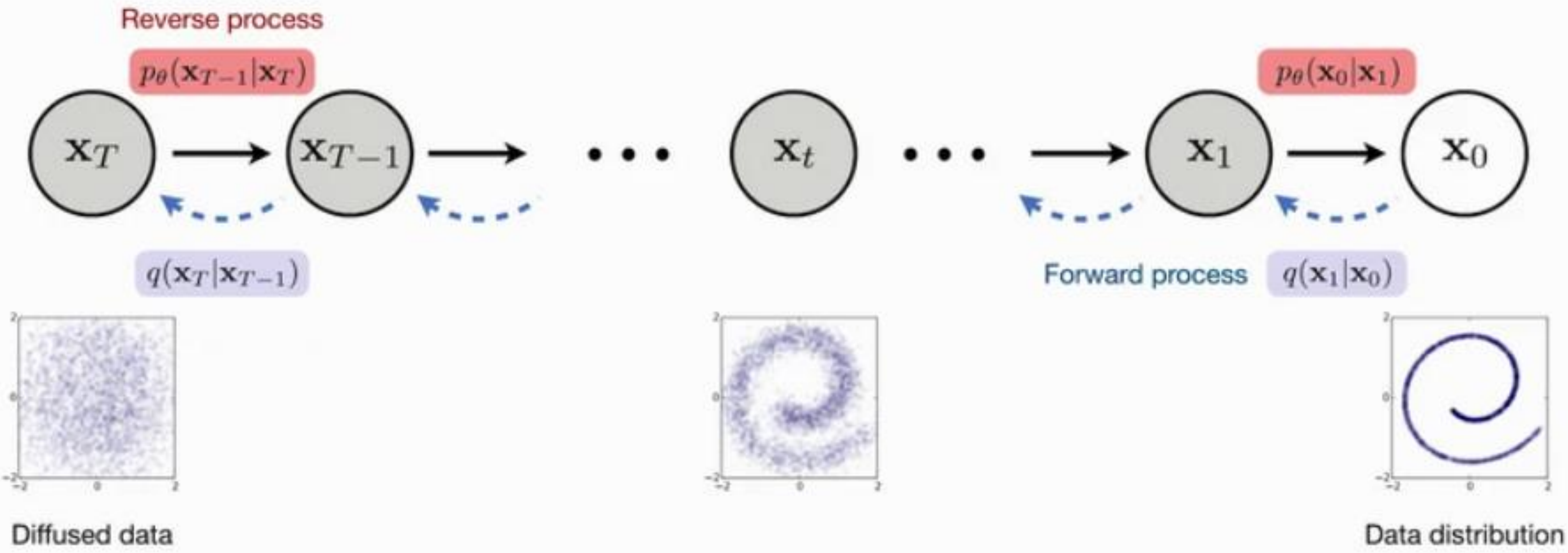
# Diffusion probabilistic models (Sohl-Dickstein et al 2015)




Noise distribution

Samples

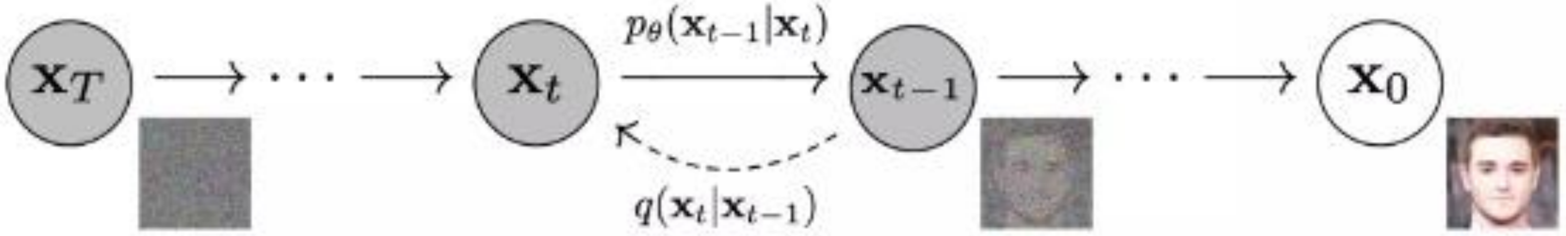
# Diffusion probabilistic models (Sohl-Dickstein et al 2015)



# Diffusion Probabilistic Model

- **Diffusion model aims to learn the reverse of noise generation procedure**
  - **Forward step:** (Iteratively) Add noise to the original sample

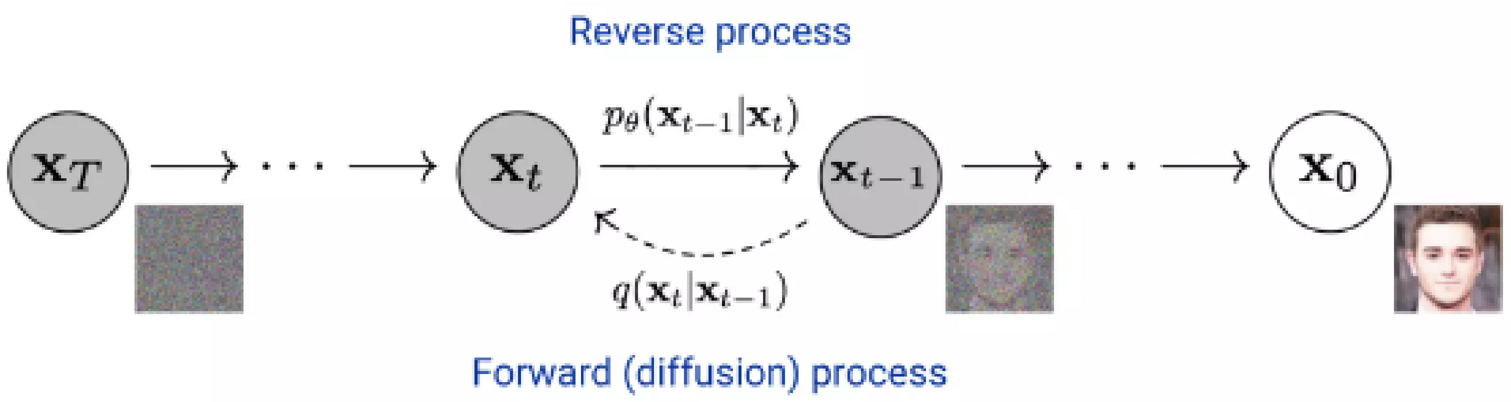
    → The sample $x_0$ converges to the complete noise $x_T$ (e.g., $\sim \mathcal{N}(0, I)$)



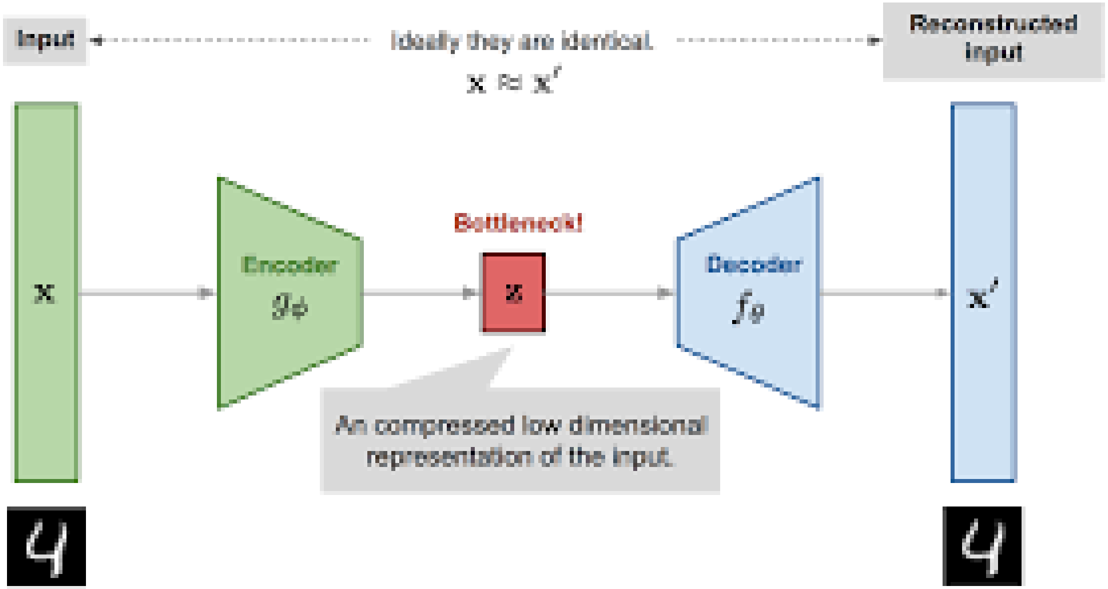

Forward (diffusion) process

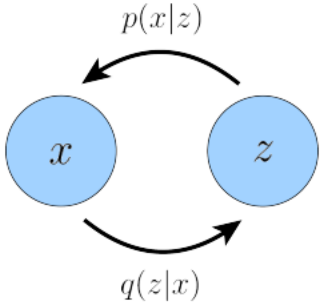
# Diffusion Probabilistic Model

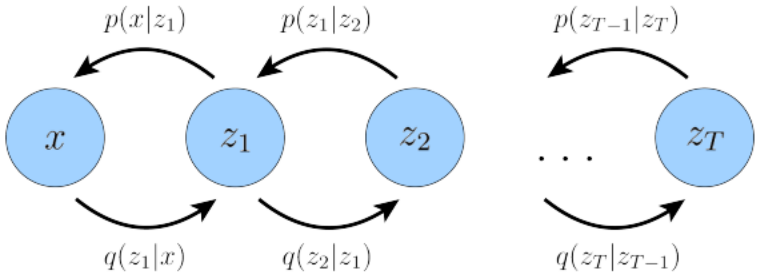
- **Diffusion model aims to learn the reverse of noise generation procedure**
  - **Forward step:** (Iteratively) Add noise to the original sample
    → The sample $x_0$ converges to the complete noise $x_T$ (e.g., $\sim \mathcal{N}(0, I)$)
  - **Reverse step:** Recover the original sample from the noise
    → Note that it is the "generation" procedure



# Connection with VAE Models



# 5.2 What is a Markov Chain?

- One special type of discrete-time is called a Markov Chain.
- **Definition:** A discrete-time stochastic process is a **Markov chain** if, for $t = 0,1,2\ldots$ and all states
  $P(\mathbf{X}_{t+1} = i_{t+1}|\mathbf{X}_t = i_t, \mathbf{X}_{t-1}=i_{t-1},\ldots,\mathbf{X}_1=i_1, \mathbf{X}_0=i_0)$
  $=P(\mathbf{X}_{t+1}=i_{t+1}|\mathbf{X}_t = i_t)$
- Essentially this says that the probability distribution of the state at time $t$+1 depends on the state at time $t(i_t)$ and does not depend on the states the chain passed through on the way to $i_t$ at time $t$.

# Loss

**VLB loss** $$\mathbb{E}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_q\left[L_T + \sum_{t>1} D_{\mathrm{KL}}\left(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)\right) + L_0\right]$$

# Loss

**VLB loss** $\mathbb{E}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_q\left[L_T + \sum_{t>1} D_{\mathrm{KL}}\left(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)\right) + L_0\right]$

# Loss

**VLB loss** $\mathbb{E}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_q\left[L_T + \sum_{t>1} D_{\mathrm{KL}}\left(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)\right) + L_0\right]$

$$\downarrow$$

**DSM loss** $\text{constant} * \|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|^2$

# Loss

**VLB loss** $\mathbb{E}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_q\left[L_T + \sum_{t>1} D_{\mathrm{KL}}\left(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \,\|\, p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)\right) + L_0\right]$

$$\downarrow$$

**DSM loss** $\cancel{\text{constant}} * \|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|^2$

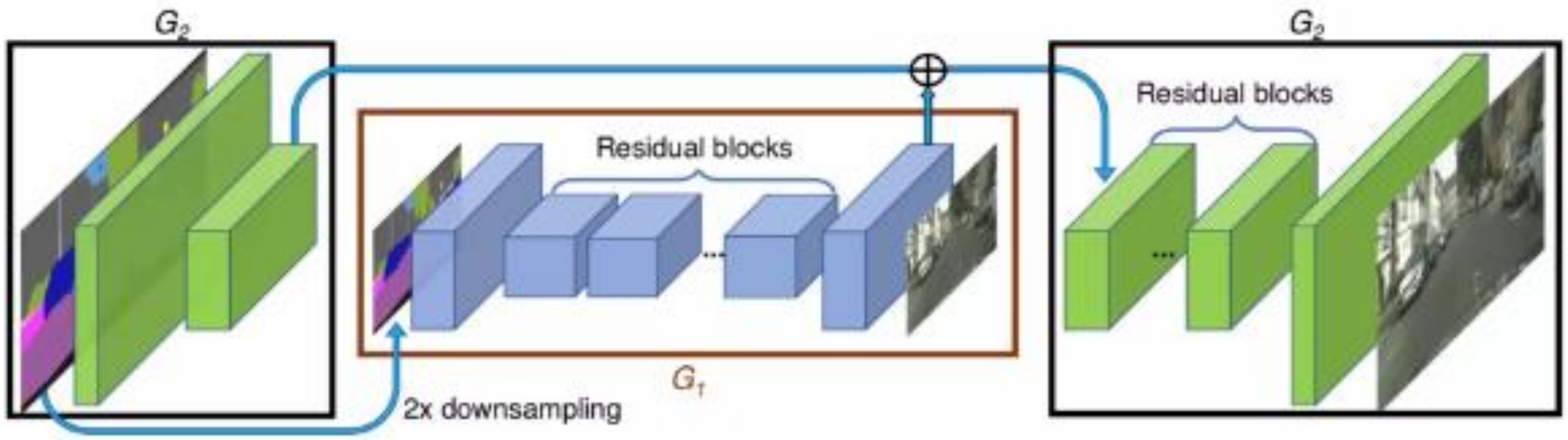
**Algorithm 1** Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2$$
6: **until** converged

From variational inference to denoising score matching

# Diffusion Probabilistic Model

- **Diffusion model aims to learn the reverse of noise generation procedure**
  - **Network:** Use the image-to-image translation (e.g., U-Net) architectures
    - Recall that input is $\mathbf{x}_t$ and output is $\mathbf{x}_{t-1}$, both are images
    - It is expensive since both input and output are high-dimensional
    - Note that the denoiser $\mu_\theta(\mathbf{x}_t, t)$ shares weights, but conditioned by step $t$



# Sampling

Shows that Langevin dynamics is the natural sampler for DSM

**Algorithm 2** Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

- **Forward Process:**
  - The forward process adds noise to the data $x_0 \sim q(x_0)$, for T time steps.

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)I)$$
$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$$

  where $\alpha_1, ...., \alpha_t$ is the variance schedule.
  - We can sample $x_t$ at any time step t with

$$q(x_t|x0) = \mathcal{N}(x_t; \sqrt{\tilde{\alpha_t}}, (1-\tilde{\alpha_t})I)$$
$$\tilde{\alpha_t} = \prod_{s=1}^{t} \alpha_s$$

$$\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon} \quad \text{with } \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})$$
$$\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon} \quad \text{with } \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})$$

$$\begin{aligned}\boldsymbol{x}_t &= \sqrt{\prod_{i=1}^{t}\alpha_i}\boldsymbol{x}_0 + \sqrt{1-\prod_{i=1}^{t}\alpha_i}\boldsymbol{\epsilon}_0 \\ &= \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0 \\ &\sim \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I})\end{aligned}$$

# Reverse process

– The reverse process removes noise starting at $p(x_T) = \mathcal{N}(x_T; 0, I)$ for T time steps.

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

$$p_\theta(x_{0:T}) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha_t}}}\epsilon_\theta(x_t, t))$$

$\theta$ are the parameters we train.

$$\begin{aligned} q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) &= \frac{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0)q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \\ &= \frac{\mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I})\mathcal{N}(\boldsymbol{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0, (1-\bar{\alpha}_{t-1})\mathbf{I})}{\mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I})} \end{aligned}$$

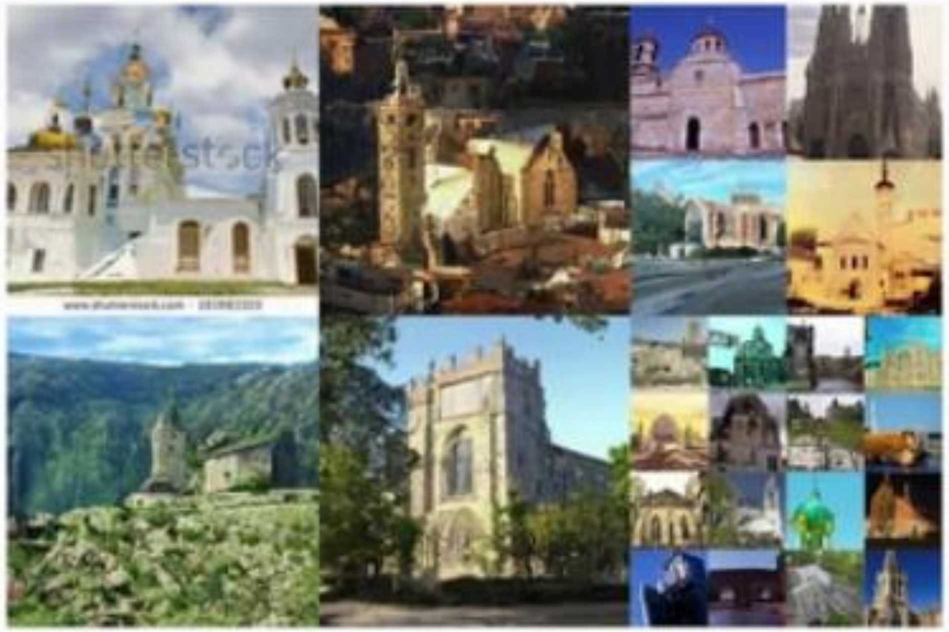
$$\propto \mathcal{N}(\boldsymbol{x}_{t-1}; \underbrace{\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t}}_{\mu_q(\boldsymbol{x}_t, \boldsymbol{x}_0)}, \underbrace{\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{I}}_{\boldsymbol{\Sigma}_q(t)})$$

# Diffusion models can generate high quality samples

CelebA-HQ 256x256

CIFAR-10 FID = 3.17 (SOTA)

# Diffusion models can generate high quality samples



LSUN 256x256 Church FID = 7.89

LSUN 256x256 Bedroom FID = 4.90

# Diffusion Model is All We Need?

- **Trilemma of generative models: Quality vs. Diversity vs. Speed**
  - Diffusion model produces diverse and high-quality samples, but generations is slow