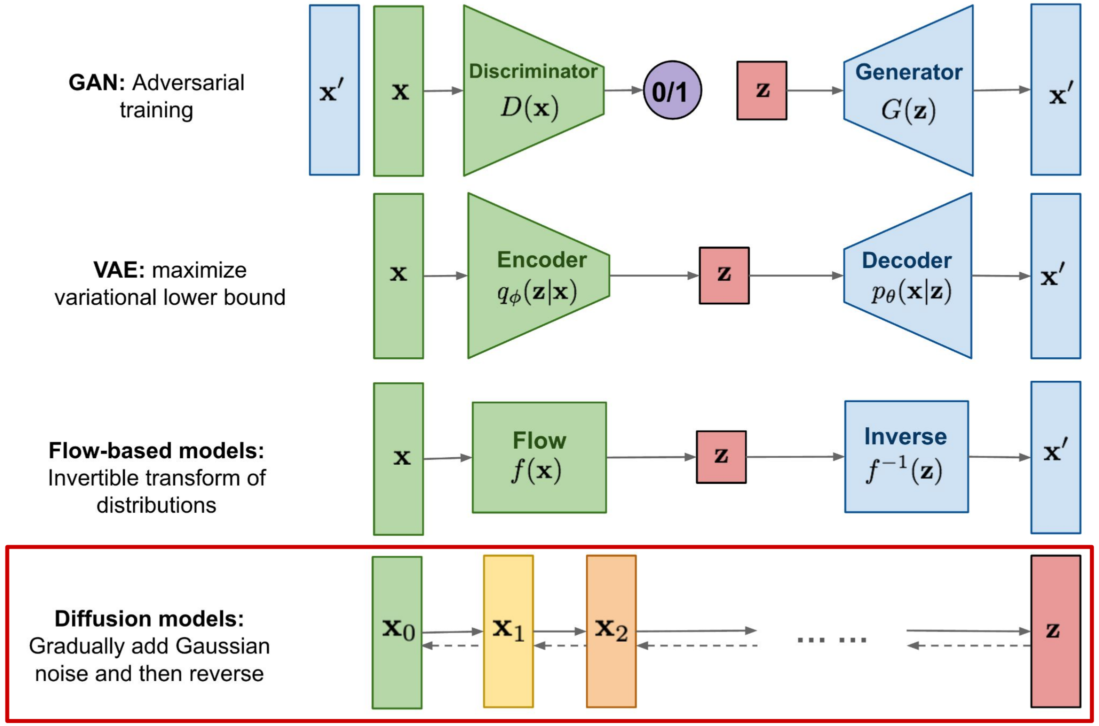
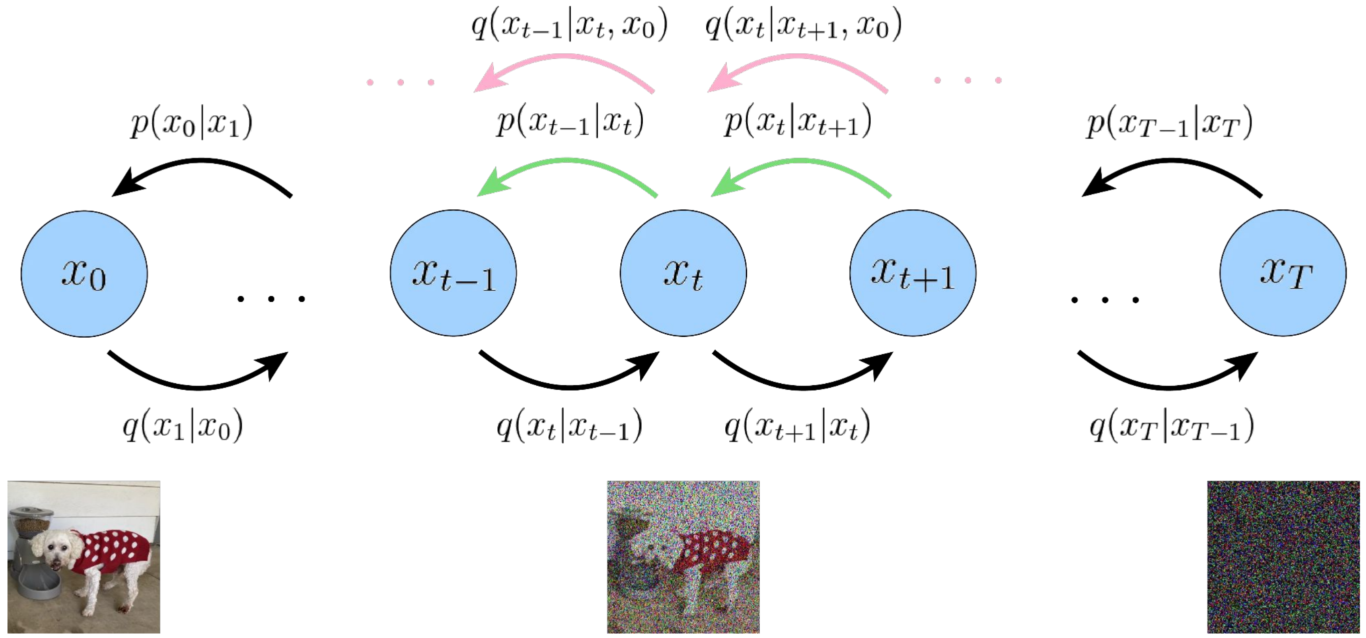
Lab 13: Diffusion

Diffusion Overview

- Diffusion models are a type of **generative model**
  - Estimate $P(x \mid z)$ to then deduce $P(z \mid x)$
- Diffusion models as **stacked VAEs**
- Forward process (noising) + reverse process (denoising)
- Train a neural network by estimating the **noise added at each time step**



Diffusion Overview



Forward process $(x_0 \rightarrow x_T)$: gradually noise images according to posterior q

Reverse process $(x_T \rightarrow x_0)$: gradually denoise images according to p

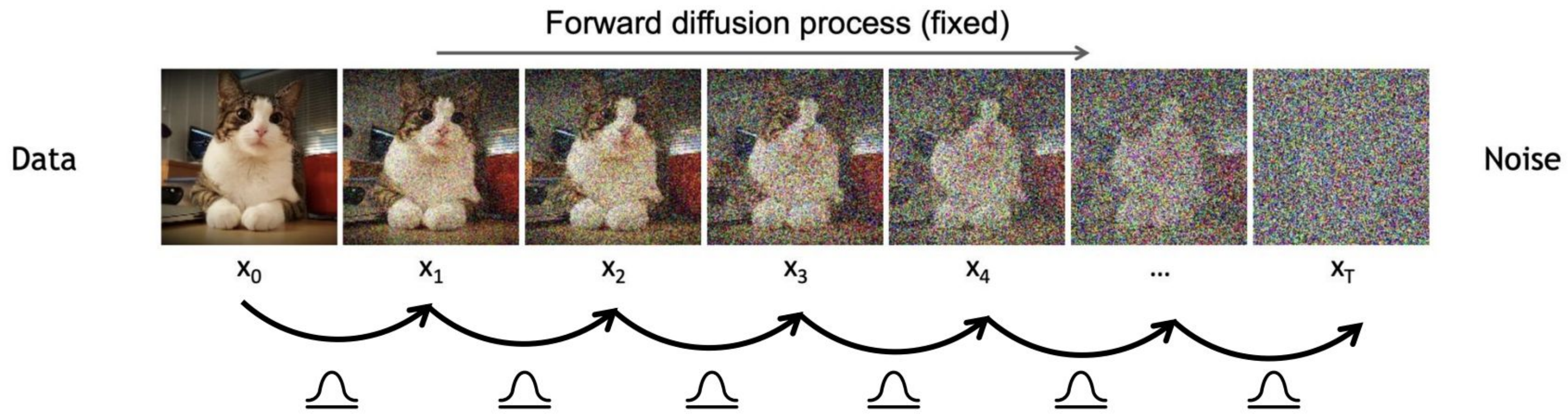
Forward Process

Forward process is usually a fixed Markov chain with transitions q

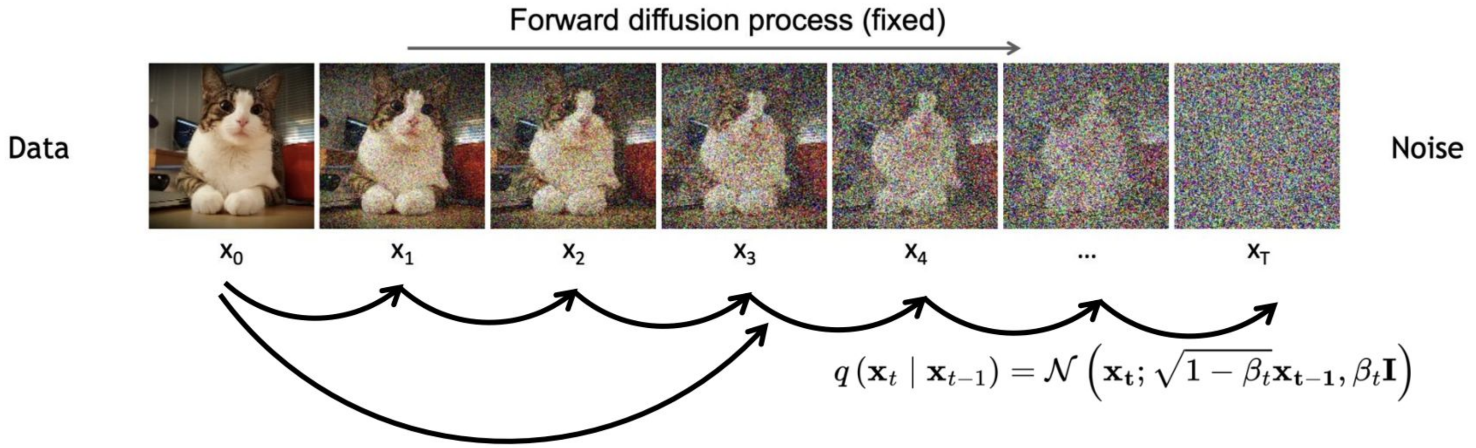
- Gradually add Gaussian noise according to schedule β_t

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

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



Forward Process



$$q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)=\mathcal{N}\left(\mathbf{x_t}; \sqrt{1-\beta_t}\mathbf{x_{t-1}}, \beta_t \mathbf{I}\right)$$

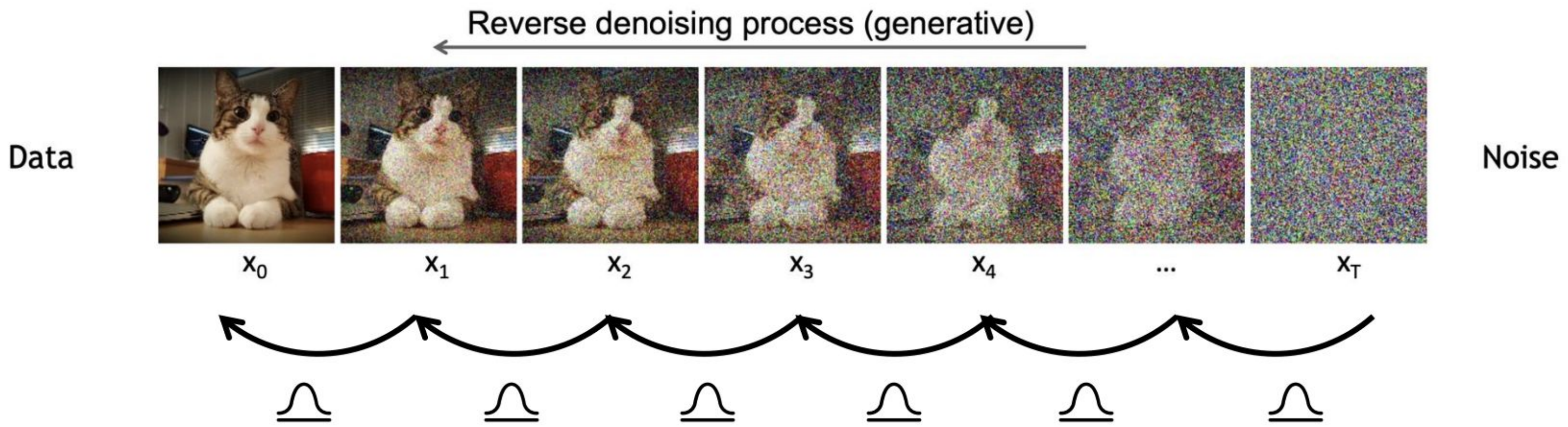
Because this is a Markov chain, we can sample x_t at any timestep t given x_0 in closed form

$$q\left(\mathbf{x}_t \mid \mathbf{x}_0\right)=\mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right)\right) \qquad \bar{\alpha}_t=\prod_{s=1}^{t}(1-\beta_s)$$

$$\mathbf{x}_t=\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{(1-\bar{\alpha}_t)}\epsilon \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$q\left(\mathbf{x}_T \mid \mathbf{x}_0\right) \approx \mathcal{N}\left(\mathbf{x}_T; \mathbf{0}, \mathbf{I}\right))$$

Reverse Process

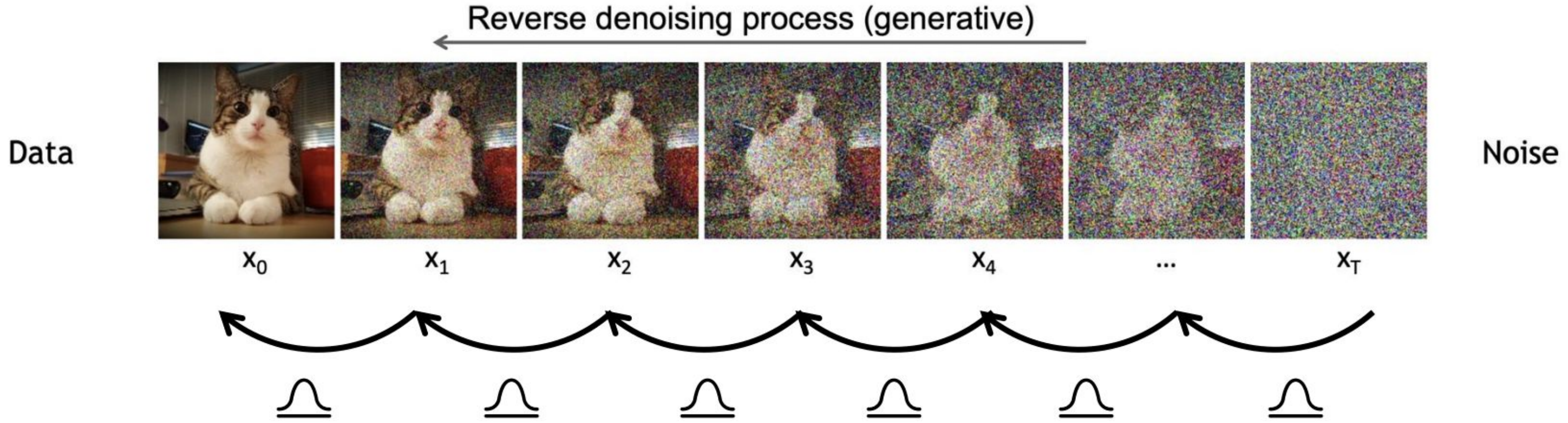


We can sample x_T from the standard normal distribution, and then iteratively sample x_{t-1} from $q(x_{t-1} \mid x_t)$

Problem: $q(x_{t-1} \mid x_t)$ is not tractable

Solution: We can approximate this with a Gaussian distribution $p_\theta(x_{t-1} \mid x_t)$

Reverse Process

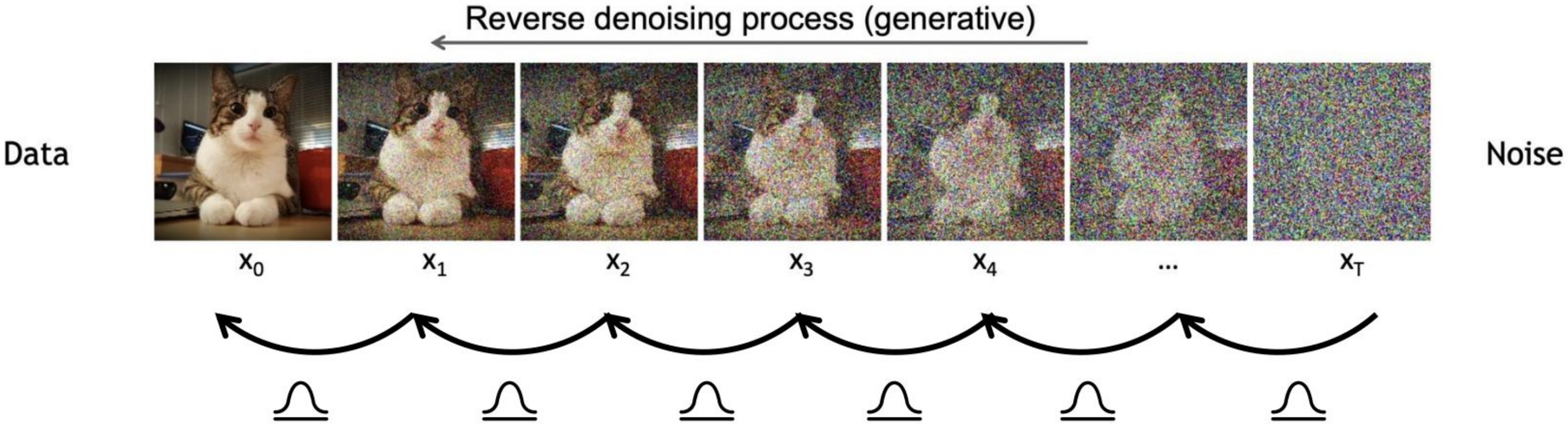


Reverse process is also a Markov chain, but with learned transitions p

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

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

Reverse Process



We want to predict the mean and std of the added Gaussian noise

$$p\left(\mathbf{x}_T\right)=\mathcal{N}\left(\mathbf{x}_T ; \mathbf{0}, \mathbf{I}\right)$$

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

$$\boxed{p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)=\mathcal{N}\left(\mathbf{x}_{t-1} ; \mu_\theta\left(\mathbf{x}_t, t\right), \sigma_t^2 \mathbf{I}\right)}$$

Training the Reverse Process

- Train using negative ELBO, which can be rewritten as

$$\mathbb{E}_{q(\mathbf{x}_0)}\left[-\log p_\theta\left(\mathbf{x}_0\right)\right] \leq \mathbb{E}_{q(\mathbf{x}_0)q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[-\log \frac{p_\theta\left(\mathbf{x}_{0:T}\right)}{q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right)}\right] =: L$$

$$L = \mathbb{E}_q[\underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_T \mid \mathbf{x}_0\right) \| p\left(\mathbf{x}_T\right)\right)}_{L_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)\right)}_{L_{t-1}} \underbrace{- \log p_\theta\left(\mathbf{x}_0 \mid \mathbf{x}_1\right)}_{L_0}]$$

Constant, ignore

Only need to care about this term!

Constant, ignore

Training the Reverse Process

$$D_{\mathrm{KL}}\left(q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)\right)$$

Gaussian (can be proved)

Gaussian

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

$$\text{where } \tilde{\mu}_t\left(\mathbf{x}_t, \mathbf{x}_0\right):=\frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1-\bar{\alpha}_t} \mathbf{x}_0+\frac{\sqrt{1-\beta_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t} \mathbf{x}_t \text{ and } \tilde{\beta}_t:=\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t$$

- Because both are Gaussians, we can use the KL divergence formula for two Gaussian distributions

Training the Reverse Process

- KL divergence between Gaussians

$$L_{t-1}=D_{\mathrm{KL}}\left(q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)\right)=\mathbb{E}_q\left[\frac{1}{2 \sigma_t^2}\left\|\tilde{\mu}_t\left(\mathbf{x}_t, \mathbf{x}_0\right)-\mu_\theta\left(\mathbf{x}_t, t\right)\right\|^2\right]+C$$

- Want to train μ_θ to predict $\tilde{\mu}_t$

$$\mathbf{x}_t=\sqrt{\bar{\alpha}_t} \mathbf{x}_0+\sqrt{\left(1-\bar{\alpha}_t\right)} \epsilon$$

$$\tilde{\mu}_t\left(\mathbf{x}_t, \mathbf{x}_0\right)=\frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon\right)$$

- Instead, we can reparametrize so that we predict the noise ϵ given x_t and t

$$\mu_\theta\left(\mathbf{x}_t, t\right)=\frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \boxed{\epsilon_\theta\left(\mathbf{x}_t, t\right)}\right)$$

Training the Reverse Process

- New Objective

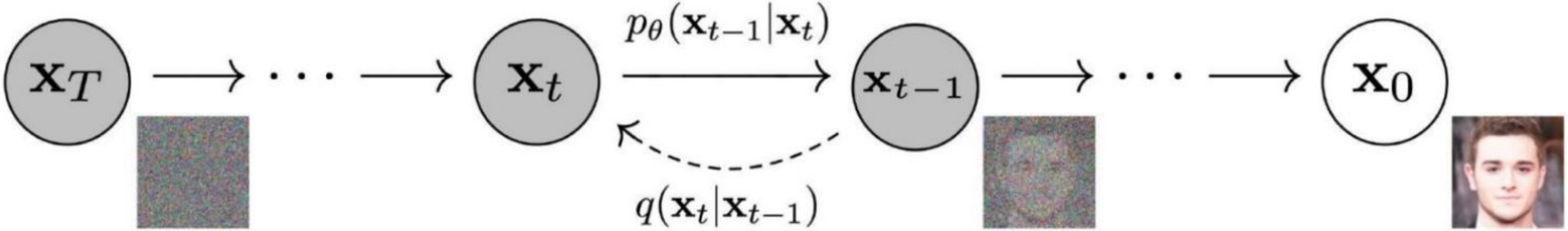
$$L_{t-1} = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\frac{\beta_t^2}{2\sigma_t^2 (1-\beta_t)(1-\bar{\alpha}_t)} \| \epsilon - \epsilon_\theta(\underbrace{\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon}_{\mathbf{x}_t}, t) \|^2 \right] + C$$

- Can be further simplified to

$$L_{\text{simple}} = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t \sim \mathcal{U}(1,T)} \left[\| \epsilon - \epsilon_\theta(\underbrace{\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon}_{\mathbf{x}_t}, t) \|^2 \right]$$

Deriving Diffusion: Sampling from posterior q

- Variance schedule: $\boldsymbol{\beta}_t$ // big $\boldsymbol{\beta}_t \rightarrow$ add a lot of noise at timestep t
- Define $\boldsymbol{\alpha}_t = \mathbf{1} - \boldsymbol{\beta}_t$
- $q(x_t|x_{t-1}) = N(x_t|\mu = \sqrt{\alpha_t}x_{t-1},\ \sigma^2 = \boldsymbol{\beta}_t I)$
- To summarize…
  - It's really easy to sample corrupted images x_t for training given a real image x_0
  - $x_t = \sqrt{\alpha_t \ldots \alpha_1}\, x_0 + \sqrt{\mathbf{1} - \alpha_t \ldots \alpha_1}\, z$
  - Basically just adding zero-centered noise to x_0 **(but the constants are important!)**



DDPM (Denoising Diffusion Probabilistic Model)

Algorithm 1 Training

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

Algorithm 2 Sampling

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

Let’s Try Diffusion!