# Deep Generative Models: Diffusion Models

Fall Semester 2024

René Vidal

Director of the Center for Innovation in Data Engineering and Science (IDEAS)

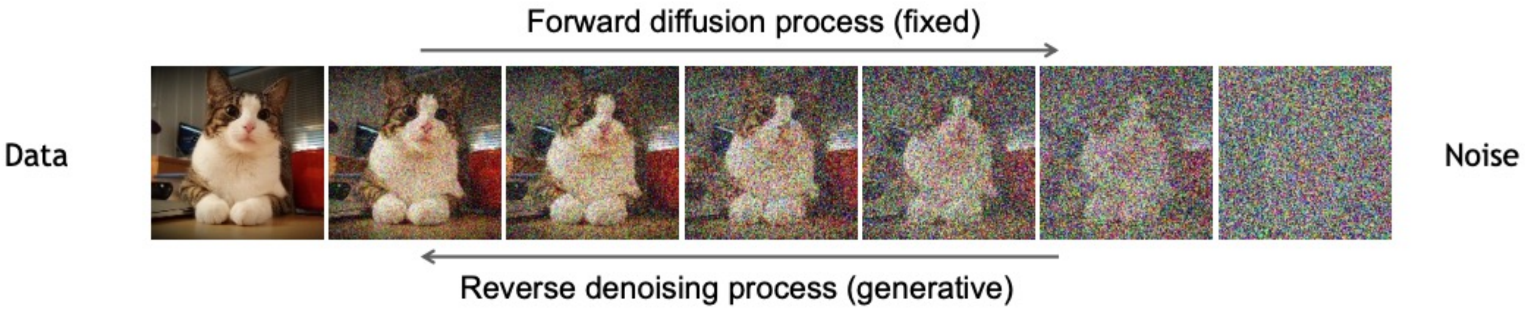
Rachleff University Professor, University of Pennsylvania
Amazon Scholar & Chief Scientist at NORCE

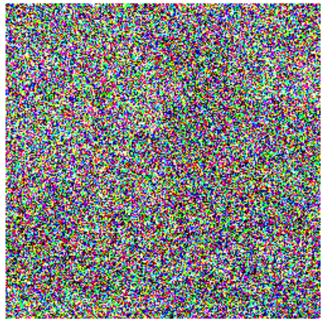
# Outline

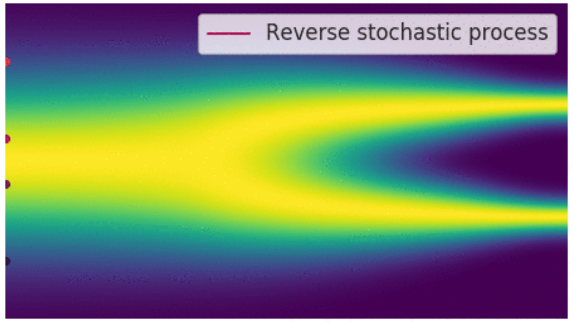
- **Denoising Diffusion Probabilistic Models**
- Conditional Diffusion Models: Stable Diffusion, ControlNet, VideoFusion

# Denoising Diffusion Probabilistic Models (DDPM)

- A denoising diffusion probabilistic model is a parameterized Markov chain trained using variational inference to produce samples matching the data after finite time.
- DDPM learns to reverse a forward diffusion process. The forward process gradually adds gaussian noise to the data until signal (i.e., the image) is destroyed. The reverse process predicts how to denoise.

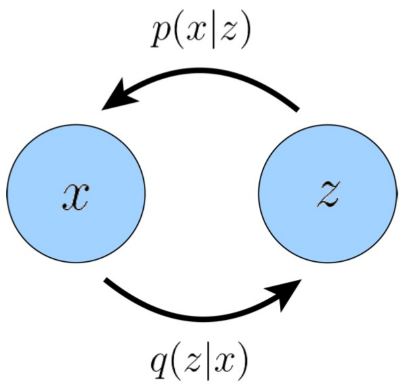

Forward diffusion process (fixed)
Data
Noise
Reverse denoising process (generative)

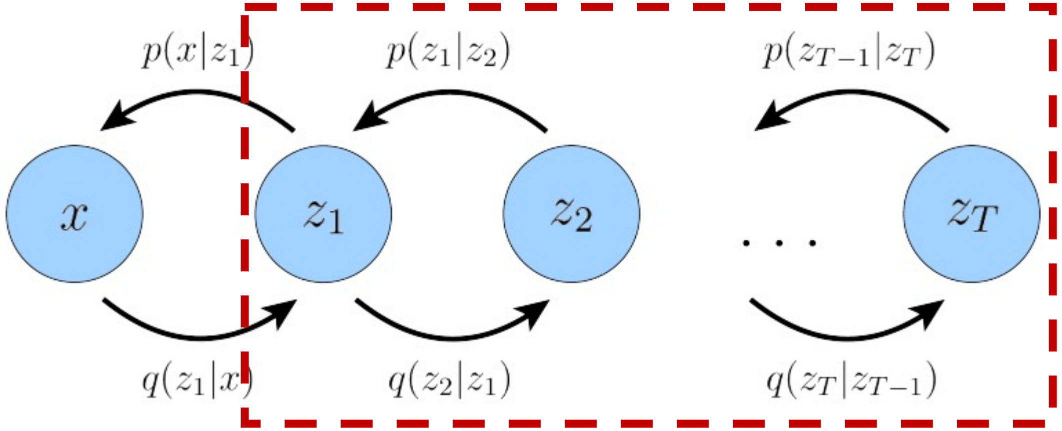




# Denoising Diffusion Probabilistic Models (DDPM)

- We can view DDPM as a Markovian Hierarchical Variational Autoencoder (MHVAE) with $T$ hierarchical latents $z = z_{1:T} = \{z_t\}_{t=1}^{T}$ modeled by a Markov chain where each latent $z_t$ is generated only from the previous latent $z_{t+1}$.





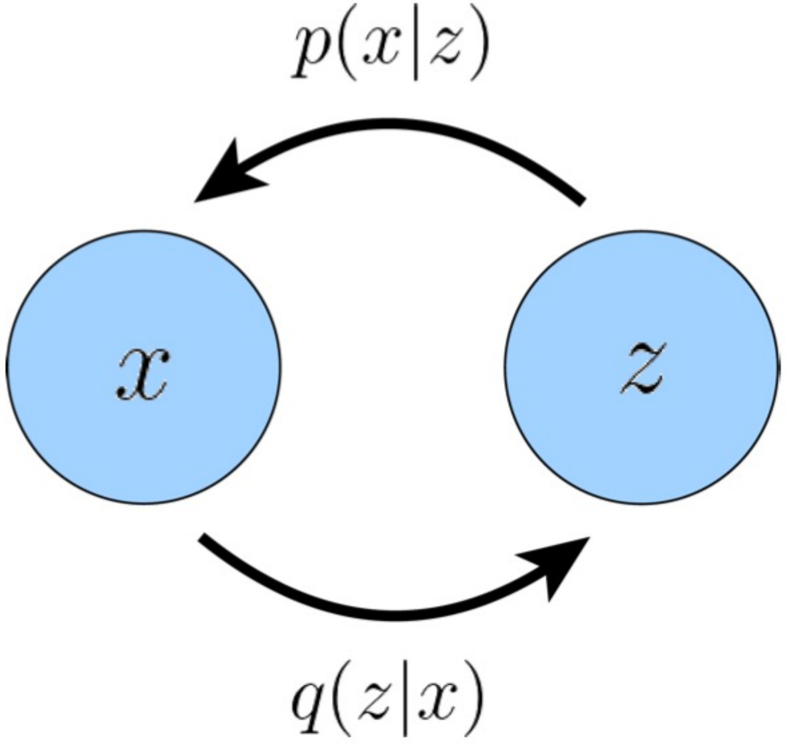
$z = z_{1:T} = \{z_t\}_{t=1}^{T}$

- What is the VAE encoder $q(z \mid x)$ of DDPM?
- What is the VAE decoder $p(x \mid z)$ of DDPM?
- What is the ELBO of DDPM?

# Let us Recall the Training Objective of the VAE

- Evidence Lower Bound (ELBO)

$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right]$$

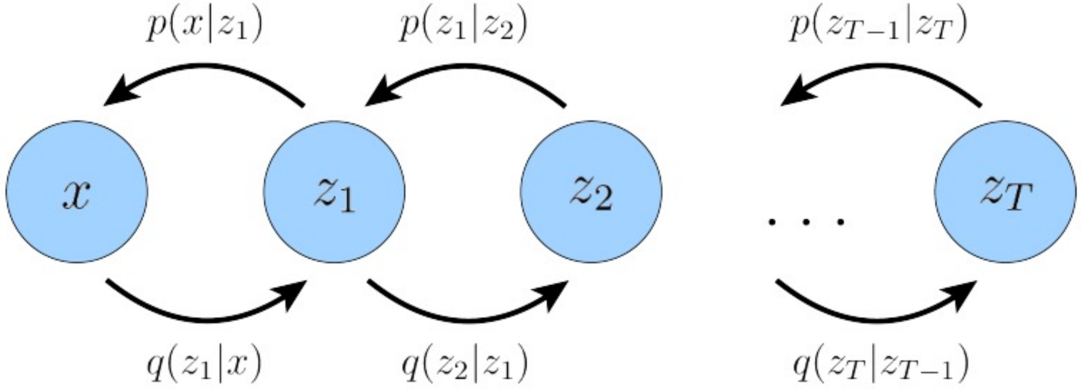


- Decomposition of the ELBO

$$\begin{aligned}
\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right] &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right] + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right] \\
&= \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z}))}_{\text{prior matching term}}
\end{aligned}$$

# MHVAE: the Latent Variable is Autoregressive

- A MHVAE is a VAE whose encoder and decoder are autoregressive models:



$$p(x, z_{1:T}) = p(z_T) p_\theta(x \mid z_1) \prod_{t=2}^{T} p_\theta(z_{t-1} \mid z_t)$$

$$q_\phi(z_{1:T} \mid x) = q_\phi(z_1 \mid x) \prod_{t=2}^{T} q_\phi(z_t \mid z_{t-1})$$

- Given this joint distribution and posterior, we can further rewrite the ELBO for MHVAE (details see next page):

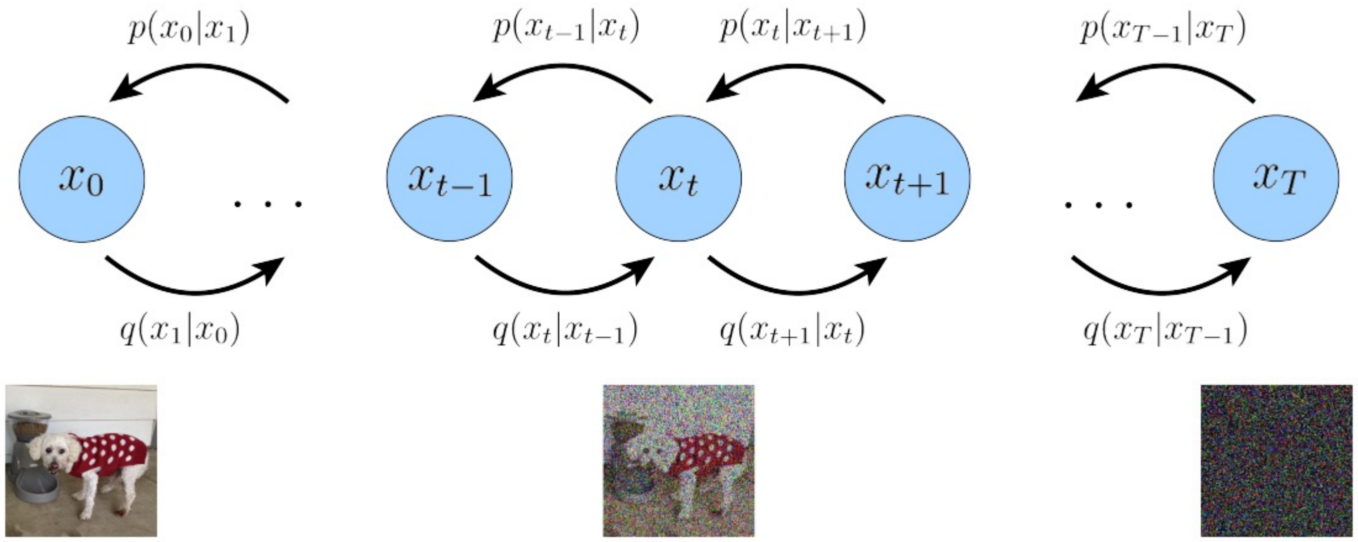
$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}_{1:T})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{x})}\right] = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{z}_T) p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}_1) \prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_1|\boldsymbol{x}) \prod_{t=2}^{T} q_{\boldsymbol{\phi}}(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})}\right]$$

# Denoising Diffusion Probabilistic Models (DDPM)

- **A DDPM is an MHVAE**: $x_0 = x$ is the data and $x_{1:T} = z_{1:T}$ is the latent variable
- All latent variables have the same dimension as the dimension of the data
- The structure of the encoder $q(\, x_{1:T} \mid x_0 \,) = \prod_{t=1}^{T} q(x_t \mid x_{t-1})$ is not learned, but it is pre-specified as a linear Gaussian model

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

- The parameter $\alpha_t$ is chosen such that $x_T \sim \mathcal{N}(x_T; 0, I)$ is a standard Gaussian



# The Forward Process of DDPM

Given the formulation of a single noising step

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\, \epsilon,\ \epsilon \sim \mathcal{N}(\epsilon; 0, I),$$

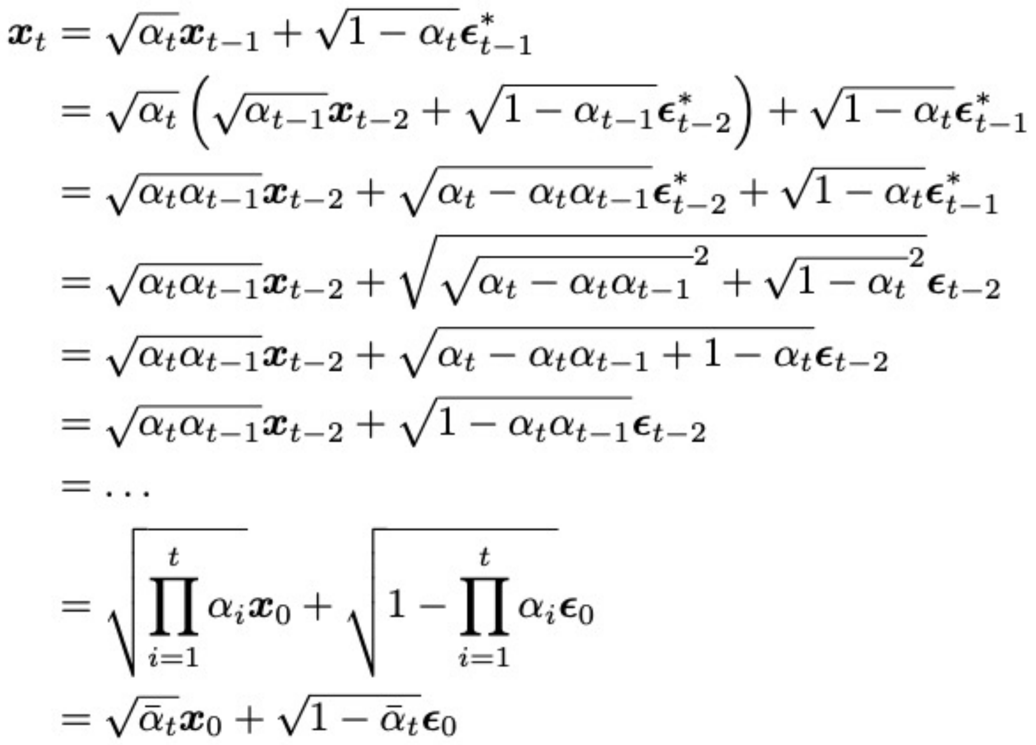
we can recursively derive the closed form for arbitrary noising steps

$$x_t = \sqrt{\overline{\alpha_t}} x_0 + \sqrt{1-\overline{\alpha_t}}\, \epsilon, \epsilon \sim \mathcal{N}(\epsilon; 0, I)$$

- That is

$$x_0 = \frac{x_t - \sqrt{1-\overline{\alpha_t}}\epsilon_0}{\sqrt{\overline{\alpha_t}}}$$

- We will use this for the reparameterization trick later.



$$\begin{aligned}
\boldsymbol{x}_t &= \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}^*_{t-1} \\
&= \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}^*_{t-2}\right) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}^*_{t-1} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{\alpha_t - \alpha_t\alpha_{t-1}}\boldsymbol{\epsilon}^*_{t-2} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}^*_{t-1} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{\sqrt{\alpha_t - \alpha_t\alpha_{t-1}}^2 + \sqrt{1-\alpha_t}^2}\boldsymbol{\epsilon}_{t-2} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{\alpha_t - \alpha_t\alpha_{t-1} + 1 - \alpha_t}\boldsymbol{\epsilon}_{t-2} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{1 - \alpha_t\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2} \\
&= \ldots \\
&= \sqrt{\prod_{i=1}^{t}\alpha_i}\boldsymbol{x}_0 + \sqrt{1 - \prod_{i=1}^{t}\alpha_i}\boldsymbol{\epsilon}_0 \\
&= \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0
\end{aligned}$$

# The Forward Process of DDPM

Note that, due to the Markov assumption of DDPM, the next phase is only conditioned on the previous adjacent phase. We can rewrite the encoder transitions using the trick:

$$
\begin{aligned}
& q(x_t \mid x_{t-1}) \\
&= q(x_t \mid x_{t-1}, x_0) \\
&= \frac{q(x_{t-1} | x_t, x_0) q(x_t | x_0)}{q(x_{t-1} | x_0)}
\end{aligned}
$$

This is because the extra conditioning term is superfluous and does not affect the conditional distribution.

We will use this derivation when rewriting the ELBO for DDPM.

# The Reverse Process of DDPM

Given assumptions of DDPM, we rewrite the joint distribution of a MHVAE to write the joint distribution for DDPM as the product of decoder transitions (reverse process):

$$p(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t) \quad \text{where} \quad p(x_T) = \mathcal{N}(x_T; 0, I).$$

The ELBO for DDPM is (details elaborated in next page):

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}_{1:T})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{x})}\right] = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{z}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}_1)\prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_1|\boldsymbol{x})\prod_{t=2}^{T} q_{\boldsymbol{\phi}}(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})}\right]$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_T|\boldsymbol{x}_{T-1})\prod_{t=1}^{T-1} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right]$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\prod_{t=1}^{T-1} p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})}{q(\boldsymbol{x}_T|\boldsymbol{x}_{T-1})\prod_{t=1}^{T-1} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right]$$

# Interpretation of the ELBO for Diffusion Model

$$\log p(\boldsymbol{x}) = \cdots$$

$$= \underbrace{E_{q(x_1|x_0)}[\log p_\theta(x_0 \mid x_1)]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q(x_T \mid x_0)|p(x_T))}_{\text{prior matching term}} - \sum_{t=2}^{T} \underbrace{E_{q(x_t|x_0)}[D_{\mathrm{KL}}(q(x_{t-1} \mid x_t, x_0) \,||\, p_\theta(x_{t-1} \mid x_t))]}_{\text{denoising matching term}}$$

$E_{q(x_1|x_0)}[\log p_\theta(x_0 \mid x_1)]$ can be interpreted as a reconstruction term; like its analogue in the ELBO of a vanilla VAE, this term can be approximated and optimized using a Monte Carlo estimate.

$D_{\mathrm{KL}}(q(x_T \mid x_0)|p(x_T))$ represents how close the distribution of the final noisified input is to the standard Gaussian prior. It has no trainable parameters, and is also equal to zero under our assumptions.

$E_{q(x_t|x_0)}[D_{\mathrm{KL}}(q(x_{t-1} \mid x_t, x_0) \,||\, p_\theta(x_{t-1} \mid x_t))]$ is a denoising matching term. We learn desired denoising transition step $p_\theta(x_{t-1} \mid x_t)$ as an approximation to tractable, ground-truth denoising transition step $q(x_{t-1} \mid x_t, x_0)$.

# ELBO for a DDPM: Denoising Matching Term

- To compute the third term, we need $$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0)q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}$$
- Letting $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, recall that

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

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

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

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

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

# ELBO for a DDPM: Training Objective

$$\boxed{\sigma_q^2(t) = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}}$$

- Recall KL divergence for Gaussians

$$D_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_x,\boldsymbol{\Sigma}_x) \parallel \mathcal{N}(\boldsymbol{y};\boldsymbol{\mu}_y,\boldsymbol{\Sigma}_y)) = \frac{1}{2}\left[\cancel{\log\frac{|\boldsymbol{\Sigma}_y|}{|\boldsymbol{\Sigma}_x|}} - d + \mathrm{tr}(\cancel{\boldsymbol{\Sigma}_y^{-1}\boldsymbol{\Sigma}_x}) + (\boldsymbol{\mu}_y-\boldsymbol{\mu}_x)^T\boldsymbol{\Sigma}_y^{-1}(\boldsymbol{\mu}_y-\boldsymbol{\mu}_x)\right]$$

- Choose variance of p to match exactly variance of q

$$D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)) = D_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{x}_{t-1};\boldsymbol{\mu}_q,\boldsymbol{\Sigma}_q(t)) \parallel \mathcal{N}(\boldsymbol{x}_{t-1};\boldsymbol{\mu}_{\boldsymbol{\theta}},\boldsymbol{\Sigma}_q(t)))$$

$$= \frac{1}{2\sigma_q^2(t)}\left[\|\boldsymbol{\mu}_{\boldsymbol{\theta}}-\boldsymbol{\mu}_q\|_2^2\right] = \boxed{\frac{1}{2\sigma_q^2(t)}\frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2}\left[\|\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)-\boldsymbol{x}_0\|_2^2\right]}$$

- Choose mean of p to match form of mean of q

$$\boxed{\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)}{1-\bar{\alpha}_t}}$$

$$\boxed{\boldsymbol{\mu}_q(\boldsymbol{x}_t,\boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t}}$$

# ELBO for a DDPM: Training Objective

- What is $\hat{x}_\theta(x_t, t)$? Neural network that seeks to predict $x_0$ from noisy image $x_t$

$$D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)) = \frac{1}{2\sigma_q^2(t)} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2} \left[\|\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{x}_0\|_2^2\right]$$

- Therefore, optimizing a DDPM boils down to learning a neural network to predict the original ground truth image from an arbitrarily noisified version of it.

- Furthermore, minimizing the sum across all noise levels can be approximated by minimizing the expectation over all timesteps, which can then be optimized using stochastic samples over timesteps.

$$\arg\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim U\{2,T\}} \left[\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \left[D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))\right]\right]$$

# ELBO for a DDPM: Training Objective

- ELBO objective can be further rewritten as:

$$
\begin{aligned}
\frac{1}{2\sigma_q^2(t)}\frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2}\left[\left\|\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)-\boldsymbol{x}_0\right\|_2^2\right] &= \frac{1}{2\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}}\frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2}\left[\left\|\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)-\boldsymbol{x}_0\right\|_2^2\right] \\
&= \frac{1}{2}\frac{\bar{\alpha}_{t-1}(1-\alpha_t)}{(1-\bar{\alpha}_{t-1})(1-\bar{\alpha}_t)}\left[\left\|\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)-\boldsymbol{x}_0\right\|_2^2\right] \\
&= \frac{1}{2}\frac{\bar{\alpha}_{t-1}-\bar{\alpha}_t}{(1-\bar{\alpha}_{t-1})(1-\bar{\alpha}_t)}\left[\left\|\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)-\boldsymbol{x}_0\right\|_2^2\right] \\
&= \frac{1}{2}\left(\frac{\bar{\alpha}_{t-1}}{1-\bar{\alpha}_{t-1}}-\frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}\right)\left[\left\|\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)-\boldsymbol{x}_0\right\|_2^2\right] \\
&= \frac{1}{2}\left(\mathrm{SNR}(t-1)-\mathrm{SNR}(t)\right)\left[\left\|\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)-\boldsymbol{x}_0\right\|_2^2\right]
\end{aligned}
$$

$$\boxed{\mathrm{SNR}=\frac{\mu^2}{\sigma^2}\quad \mathrm{SNR}(t)=\frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}}$$

- Can train a network to predict $\mathrm{SNR}(t)=\exp(-\omega_{\boldsymbol{\eta}}(t)) \therefore 1-\bar{\alpha}_t=\mathrm{sigmoid}(\omega_{\boldsymbol{\eta}}(t))$

# Reparameterization as an Alternative Form for ELBO

- Plugging our previous finding $x_0 = \frac{x_t - \sqrt{1-\overline{\alpha_t}}\epsilon_0}{\sqrt{\overline{\alpha_t}}}$ into the denoising transition mean $\mu_q(x_t, x_0)$, we have:

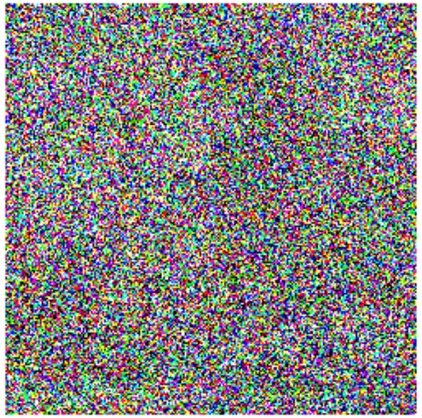
$$
\begin{aligned}
\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) &= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t} \\
&= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0}{\sqrt{\bar{\alpha}_t}}}{1-\bar{\alpha}_t} \\
&= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + (1-\alpha_t)\frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0}{\sqrt{\alpha_t}}}{1-\bar{\alpha}_t} \\
&= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t}{1-\bar{\alpha}_t} + \frac{(1-\alpha_t)\boldsymbol{x}_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}} - \frac{(1-\alpha_t)\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}} \\
&= \left(\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} + \frac{1-\alpha_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}}\right)\boldsymbol{x}_t - \frac{(1-\alpha_t)\sqrt{1-\bar{\alpha}_t}}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}}\boldsymbol{\epsilon}_0 \\
&= \left(\frac{\alpha_t(1-\bar{\alpha}_{t-1})}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}} + \frac{1-\alpha_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}}\right)\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_0 \\
&= \frac{\alpha_t - \bar{\alpha}_t + 1 - \alpha_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_0 \\
&= \frac{1-\bar{\alpha}_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_0 \\
&= \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \boxed{\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_0}
\end{aligned}
$$

- This inspires us to approximate the denoising transition mean as choosing the mean of p to match q:

$$
\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \boxed{\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}
$$

# ELBO for a DDPM: Noise Predictor

- The model predicts the noise to be removed in each step (i.e., denoising) by optimizing denoising matching term. This reduces to minimizing the difference between the means of the two distributions:



$$
\begin{aligned}
&\arg\min_{\boldsymbol{\theta}} D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)) \\
&= \arg\min_{\boldsymbol{\theta}} D_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{x}_{t-1};\boldsymbol{\mu}_q,\boldsymbol{\Sigma}_q(t)) \parallel \mathcal{N}(\boldsymbol{x}_{t-1};\boldsymbol{\mu}_{\boldsymbol{\theta}},\boldsymbol{\Sigma}_q(t))) \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)}\left[\left\|\frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t) - \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t + \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_0\right\|_2^2\right] \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)}\left[\left\|\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_0 - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)\right\|_2^2\right] \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)}\left[\left\|\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}(\boldsymbol{\epsilon}_0 - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t))\right\|_2^2\right] \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)}\frac{(1-\alpha_t)^2}{(1-\bar{\alpha}_t)\alpha_t}\left[\left\|\boldsymbol{\epsilon}_0 - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)\right\|_2^2\right]
\end{aligned}
$$

**Algorithm 1** Training

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

# Sampling from Diffusion Model

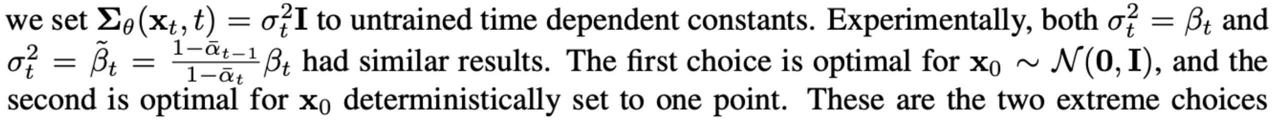
- The complete sampling procedure, as we have described, iteratively executes the denoising process from a Gaussian initialization $x_T$.

Recall: $q(\,x_{t-1} \mid x_t, x_0\,) = N(x_{t-1}; \underline{\mu_q(x_t, x_0)}, \underbrace{\Sigma_q(t)}_{})$

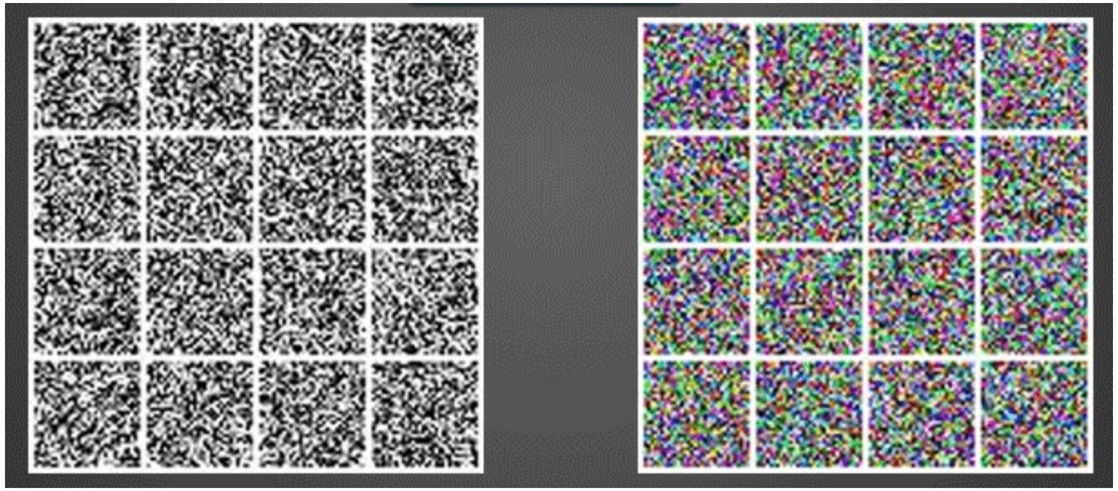
As we have derived:

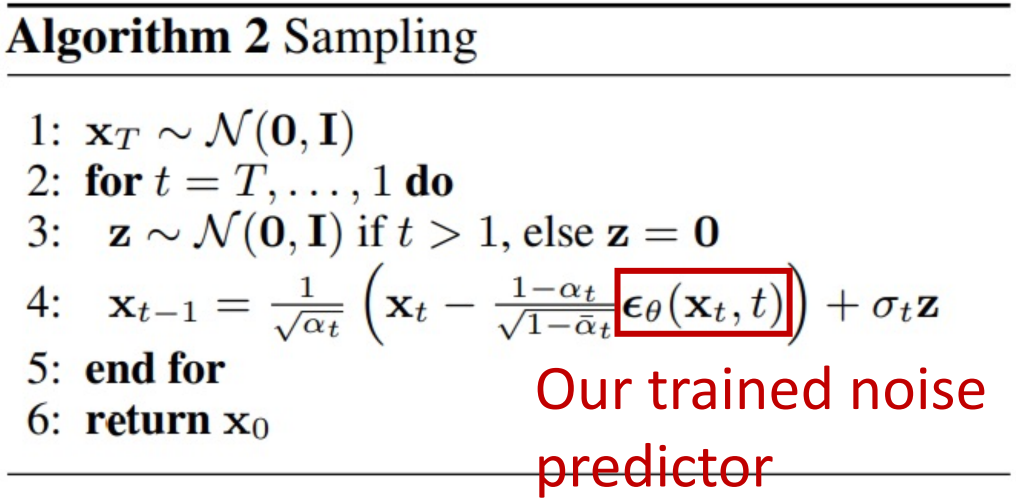
$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$$

The variance is a scheduled constant:



we set $\mathbf{\Sigma}_\theta(\mathbf{x}_t, t) = \sigma_t^2\mathbf{I}$ to untrained time dependent constants. Experimentally, both $\sigma_t^2 = \beta_t$ and $\sigma_t^2 = \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$ had similar results. The first choice is optimal for $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and the second is optimal for $\mathbf{x}_0$ deterministically set to one point. These are the two extreme choices





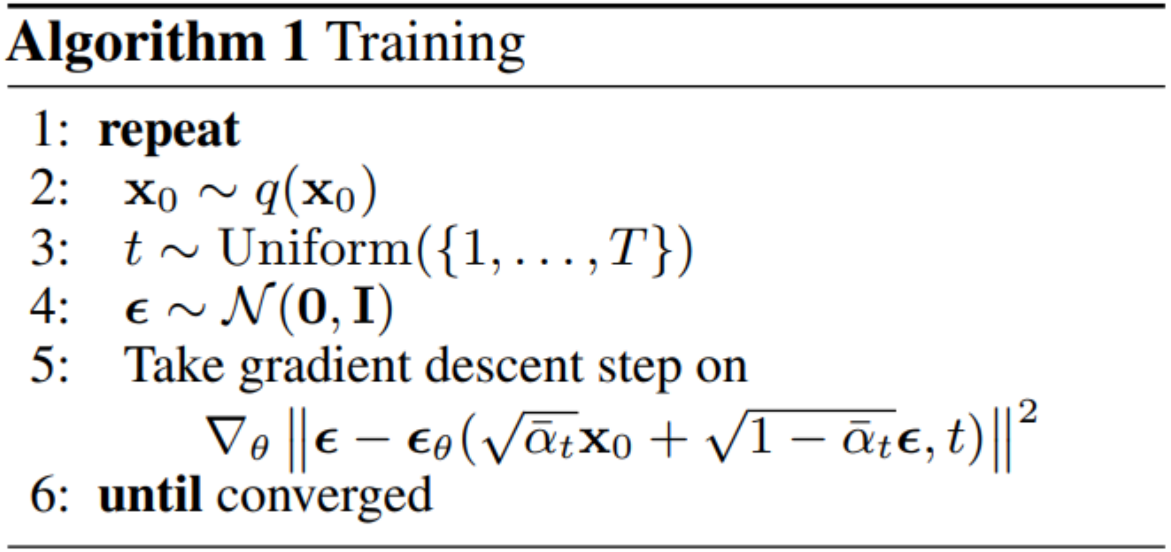
**Algorithm 2** Sampling

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

Our trained noise predictor

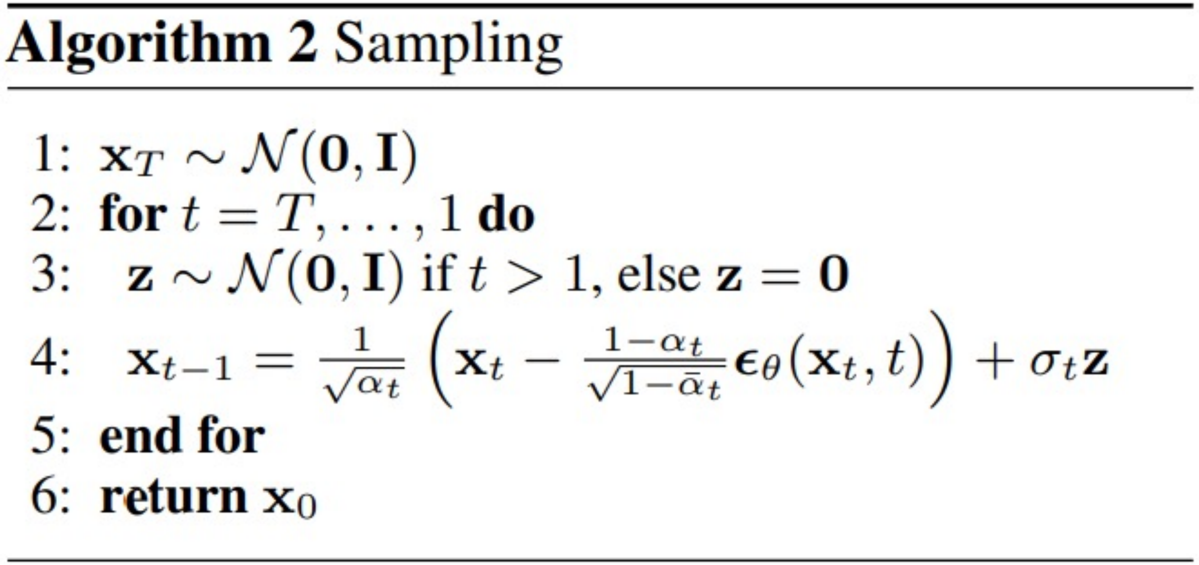
# Summary

- Our key idea is to find a way to learn the reverse process.
- Give a (corrupted) image and its current time step, the neural network predicts the noise for next reverse time step.



**Algorithm 1** Training

```
repeat
    $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
    $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
    $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    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
```

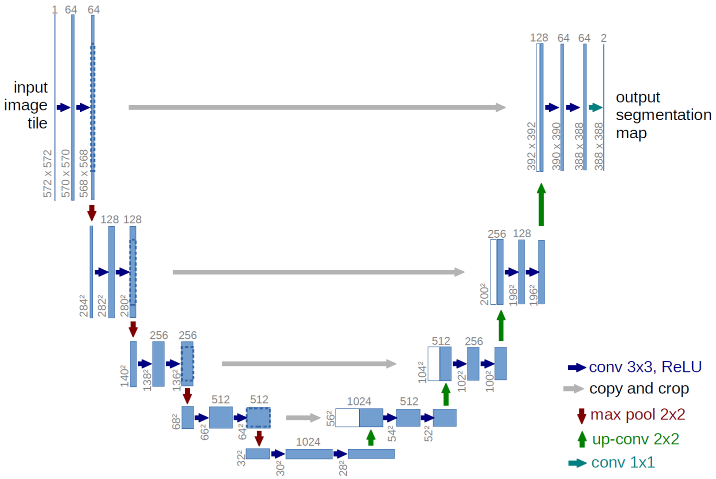


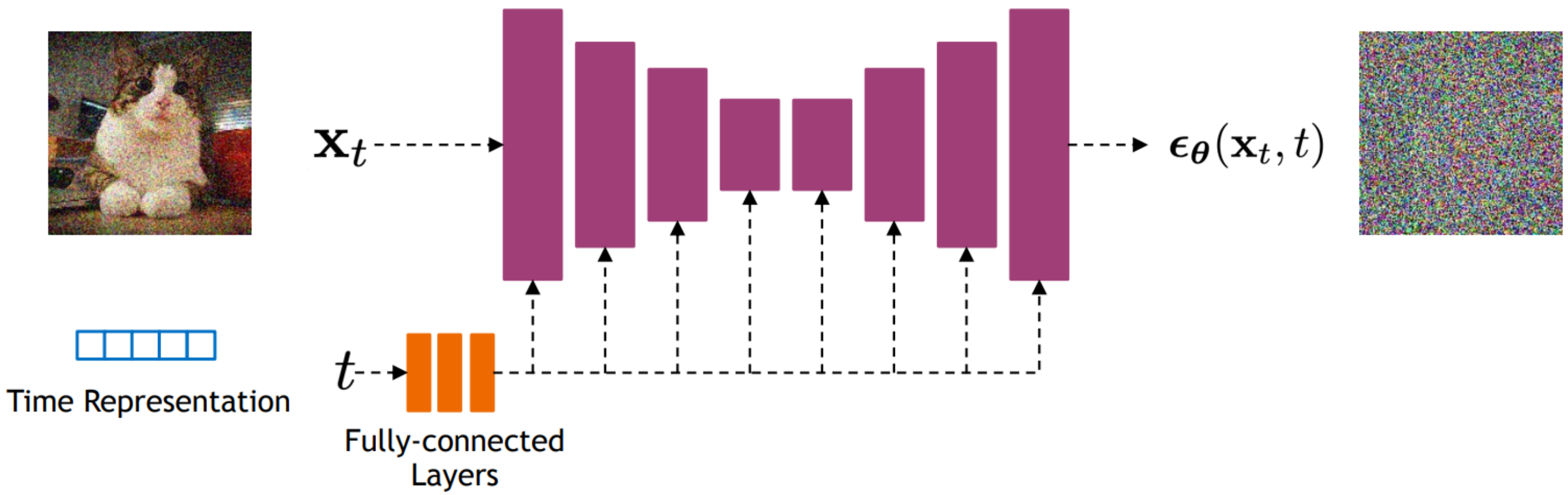
**Algorithm 2** Sampling

```
$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ 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 $\mathbf{x}_0$
```

# Implementation

- DDPM often uses U-Net with residual connection and self-attention layers to represent $\epsilon_\theta(\mathbf{x}_t, t)$ .
- Time representation is conditioned in the U-Net as sinusoidal positional embeddings or random Fourier features.
- Given a (corrupted) image and its current time step, the U-Net predicts the noise for next reverse time step.





# Implementation

- Samples of DDPM