UNIT 5

Diffusion Models

Johannes Brandstetter
Institute for Machine Learning

JOHANNES KEPLER
UNIVERSITY LINZ

Institute for
Machine Learning

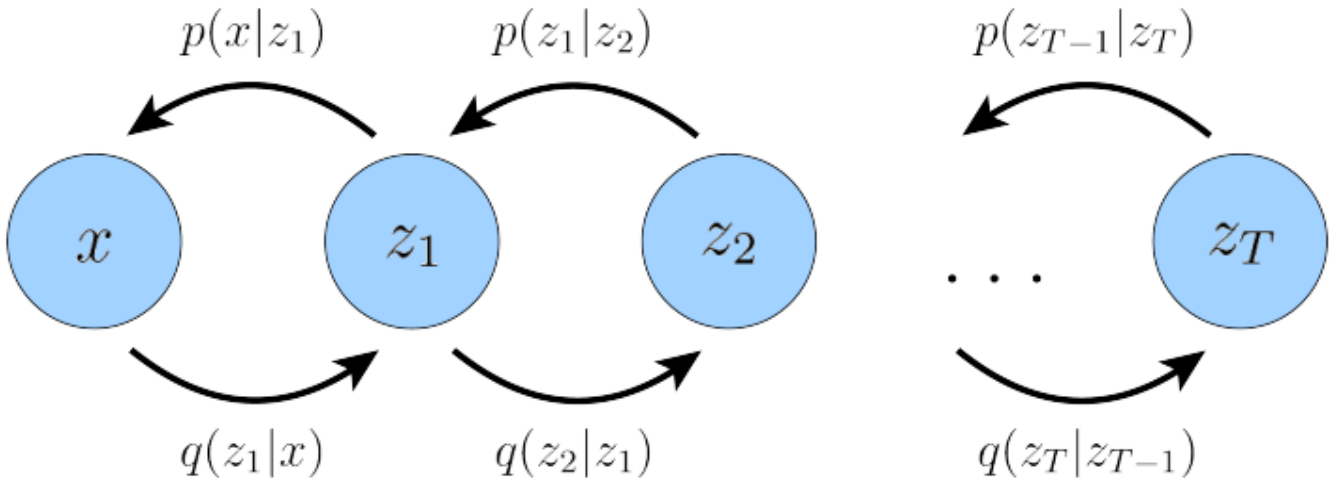
Copyright statement:

Evidence Lower Bound (ELBO)

$$
\begin{aligned}
\log p(\boldsymbol{x}) &= \log p(\boldsymbol{x}) \times \int q_\phi(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z} \\
&= \int \log p(\boldsymbol{x}) \times q_\phi(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z} \\
&= \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \Big[\log p(\boldsymbol{x}) \Big] \\
&= \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{p(\boldsymbol{z}|\boldsymbol{x})} \right] \\
&= \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{p(\boldsymbol{z}|\boldsymbol{x})} \times \frac{q_\phi(\boldsymbol{z}|\boldsymbol{x})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \right] \\
&= \underbrace{\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \right]}_{\text{ELBO}} + \underbrace{\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \left[\log \frac{q_\phi(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z}|\boldsymbol{x})} \right]}_{\mathbb{D}_{\mathsf{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) || p(\boldsymbol{z}|\boldsymbol{x}))}
\end{aligned}
$$

Hierarchical VAE

- HVAE extends to multiple hierarchies over latent variables
- Latent variables are interpreted as generated from other higher-level, i.e., more abstract latents
- Special case: **Markovian HVAE** $\Rightarrow$ generative process is a Markov chain



Markovian HVAE

- The generative process is modeled as a Markov chain: each latent $\boldsymbol{z}_t$ is generated only from the previous latent $\boldsymbol{z}_{t+1}$
- Joint distribution:

$$p(\boldsymbol{x}, \boldsymbol{z}_{1:T}) = p(\boldsymbol{z}_T) p_\theta(\boldsymbol{x}|\boldsymbol{z}_1) \prod_{t=2}^{T} p_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t)$$

- The encoder needs to model:

$$q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{x}) = q_\phi(\boldsymbol{z}_1|\boldsymbol{x}) \prod_{t=2}^{T} q_\phi(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})$$

ELBO of Hierarchical VAE

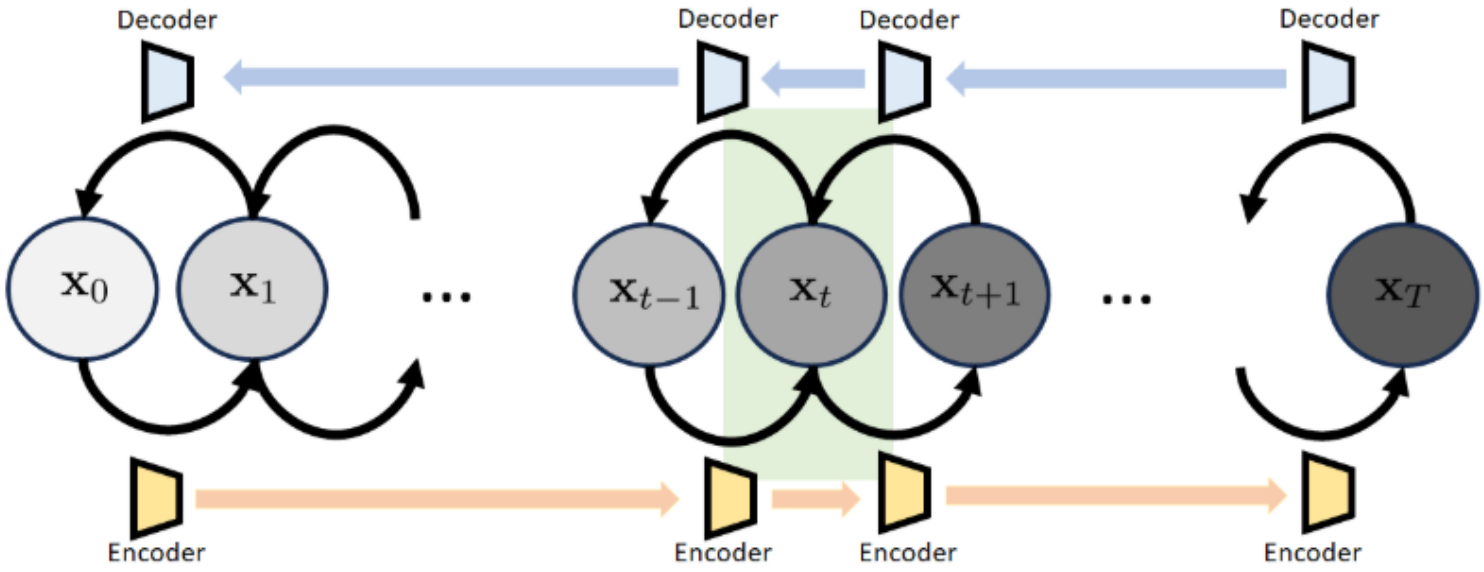
$$
\begin{aligned}
\log p(\boldsymbol{x}) &= \log p(\boldsymbol{x}) \times \int q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{x}) d\boldsymbol{z}_{1:T} \\
&= \int \log p(\boldsymbol{x}) \times q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{x}) d\boldsymbol{z}_{1:T} \\
&= \mathbb{E}_{q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{x})}\big[\log p(\boldsymbol{x})\big] \\
&= \mathbb{E}_{q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}_{1:T})}{p(\boldsymbol{z}_{1:T}|\boldsymbol{x})}\right] \\
&= \mathbb{E}_{q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}_{1:T})}{p(\boldsymbol{z}_{1:T}|\boldsymbol{x})} \times \frac{q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{x})}{q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{x})}\right] \\
&= \underbrace{\mathbb{E}_{q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}_{1:T})}{q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{x})}\right]}_{\text{ELBO}} + \underbrace{\mathbb{E}_{q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{x})}\left[\log \frac{q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{x})}{p(\boldsymbol{z}_{1:T}|\boldsymbol{x})}\right]}_{\mathbb{D}_{\mathsf{KL}}(q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{x})||p(\boldsymbol{z}_{1:T}|\boldsymbol{x}))}
\end{aligned}
$$

From HVAE to Diffusion

- The latent dimension is exactly equal to the data dimension
- The structure of the latent encoder at each timestep is not learned; it is pre-defined as a linear Gaussian model. In other words, it is a Gaussian distribution **centered around the output of the previous timestep**
- The Gaussian parameters of the latent encoders vary over time in such a way that the distribution of the latent at **final timestep T is a standard Gaussian**

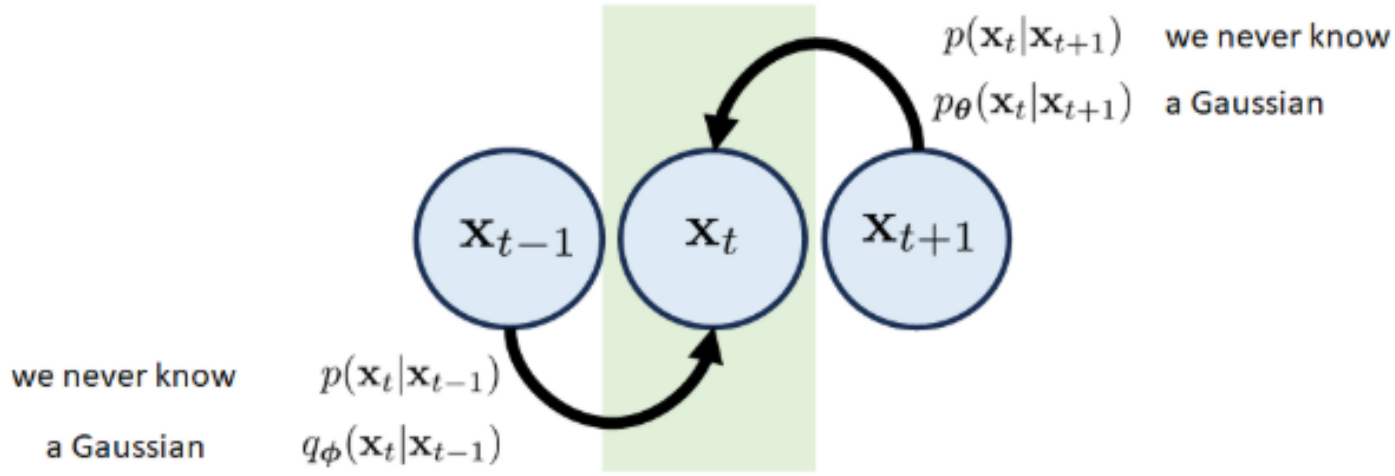
Diffusion through the lense of VAEs

- The input image is $\boldsymbol{x}_0$ and the white noise is $\boldsymbol{x}_T$
- The intermediate states $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{T-1}$ are latent variables
- The **forward transition** from $\boldsymbol{x}_{t-1}$ to $\boldsymbol{x}_t$ is analogous to the encoder, the **backward transition** from $\boldsymbol{x}_t$ to $\boldsymbol{x}_{t-1}$ is analogous to the decoder
- Input and output dimensions of the encoder/decoder are identical



Transition blocks

- Holds for $t = 1, \ldots, T-1$
- Just like in a VAE, the encoder (forward) transition distribution $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ is never accessible
  - We approximate it by a Gaussian $q_\phi(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$
- backward (decoder) transition goes from $\boldsymbol{x}_{t+1}$ to $\boldsymbol{x}_t$
  - We approximate it by another Gaussian $p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})$, but this time we need to estimate the (means of the) distribution by a neural net



Initial and final block

- **Initial block**: Since the diffusion process starts at $\boldsymbol{x}_0$, we only need to worry about the backward transition $p(\boldsymbol{x}_0|\boldsymbol{x}_1)$
  - Since $p(\boldsymbol{x}_0|\boldsymbol{x}_1)$ is never accessible, we approximate it by a Gaussian $p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)$, where we again estimate the mean by a neural network
- **Final block**: There is only a forward transition from $\boldsymbol{x}_{T-1}$ to $\boldsymbol{x}_T$, which is approximated by a Gaussian $q_\phi(\boldsymbol{x}_T|\boldsymbol{x}_{T-1})$

Transition distribution

- In a denoising diffusion probabilistic model, the transition distribution $q_\phi(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ is defined as:

$$q_\phi(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \overset{\text{def}}{=} \mathcal{N}(\sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I})$$

- To proof this transition distribution, we start with two scalars $a, b \in \mathbb{R}$, and define the transition distribution as:

$$q_\phi(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(a\boldsymbol{x}_{t-1}, b^2\mathbf{I})$$

  The equivalent sampling step is $\boldsymbol{x}_t = a\boldsymbol{x}_{t-1} + b\boldsymbol{\epsilon}_{t-1}$

- For a random variable $X \sim \mathcal{N}(\mu, \sigma^2)$, drawing X from this distribution is equivalent to $X = \mu + \sigma\eta$ where $\eta \sim \mathcal{N}(0, 1)$

Transition distribution

$$
\begin{aligned}
\boldsymbol{x}_t &= a\boldsymbol{x}_{t-1} + b\boldsymbol{\epsilon}_{t-1} \\
&= a(a\boldsymbol{x}_{t-2} + b\boldsymbol{\epsilon}_{t-2}) + b\boldsymbol{\epsilon}_{t-1} \\
&= a^2\boldsymbol{x}_{t-2} + ab\boldsymbol{\epsilon}_{t-2} + b\boldsymbol{\epsilon}_{t-1} \\
&\vdots \\
&= a^t\boldsymbol{x}_0 + b\underbrace{\left[\boldsymbol{\epsilon}_{t-1} + a\boldsymbol{\epsilon}_{t-2} + a^2\boldsymbol{\epsilon}_{t-3} + \ldots + a^{t-1}\boldsymbol{\epsilon}_0\right]}_{=\boldsymbol{w}_t}
\end{aligned}
$$

$\mathbb{E}[\boldsymbol{x}_t] = a^t\mathbb{E}[\boldsymbol{x}_0] \Rightarrow$ for $a < 1$, $\mathbb{E}[\boldsymbol{x}_t]$ goes towards zero for $t \to \infty$

$$
\begin{aligned}
\mathsf{Cov}[\boldsymbol{x}_t] &= 0 + \mathsf{Cov}(\boldsymbol{w}_t) \\
&= b^2\left(\mathsf{Cov}(\boldsymbol{\epsilon}_{t-1}) + a^2\mathsf{Cov}(\boldsymbol{\epsilon}_{t-2}) + \ldots + (a^{t-1})^2\mathsf{Cov}(\boldsymbol{\epsilon}_0)\right)\mathbf{I} \\
&= b^2\frac{1-a^{2t}}{1-a^2}\mathbf{I}
\end{aligned}
$$

Transition distribution

- We want $\underset{t\to\infty}{\text{Cov}}[\boldsymbol{x}_t] = \frac{b^2}{1-a^2}\mathbf{I} = \mathbf{I}$
  - I.e., the distribution $\text{Cov}[\boldsymbol{x}_t]$ should approach $\mathcal{N}(0, \mathbf{I})$
- Setting $a = \sqrt{\alpha}$ this yields $b = \sqrt{1-\alpha}$
- This gives:

$$\boldsymbol{x}_t = \sqrt{\alpha}\boldsymbol{x}_{t-1} + \sqrt{1-\alpha}\boldsymbol{\epsilon}_{t-1}$$

$$q_\phi(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\sqrt{\alpha}\boldsymbol{x}_{t-1}, (1-\alpha)\mathbf{I})$$

- If a scheduler is used, α is replaced by α_t

Conditional distribution

$$
\begin{aligned}
\boldsymbol{x}_t &= \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \\
&= \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2}\right) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \underbrace{\sqrt{\alpha_t}\sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}}_{\boldsymbol{w}_1}
\end{aligned}
$$

- Sum of two Gaussians remains a Gaussian $\Rightarrow$ we just need to re-calculate its new covariance: $\text{Cov}(\boldsymbol{w}_1) = \left[(\sqrt{\alpha_t}\sqrt{1-\alpha_{t-1}})^2 + (\sqrt{1-\alpha_t})^2\right]\mathbf{I} = \left[1-\alpha_t\alpha_{t-1}\right]\mathbf{I}$

Conditional distribution

- Therefore:

$$\begin{aligned}\boldsymbol{x}_t &= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2} \\ &= \sqrt{\alpha_t\alpha_{t-1}\alpha_{t-2}}\boldsymbol{x}_{t-3} + \sqrt{1-\alpha_t\alpha_{t-1}\alpha_{t-2}}\boldsymbol{\epsilon}_{t-3} \\ &\vdots \\ &= \sqrt{\prod_{i=1}^{t}\alpha_i}\boldsymbol{x}_0 + \sqrt{1-\prod_{i=1}^{t}\alpha_i}\boldsymbol{\epsilon}_0\end{aligned}$$

- Consequently, for $\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i$:

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0 \ ,$$

- The distribution $q_\phi(\boldsymbol{x}_t|\boldsymbol{x}_0)$ can be written as:

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

Evidence lower bound

$$\log p(\boldsymbol{x}_0) = \log \int p(\boldsymbol{x}_{0:T}) d\boldsymbol{x}_{1:T}$$

$$= \log \int p(\boldsymbol{x}_{0:T}) \frac{q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}{q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} d\boldsymbol{x}_{1:T}$$

$$= \log \int q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) \frac{p(\boldsymbol{x}_{0:T})}{q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} d\boldsymbol{x}_{1:T}$$

$$= \log \mathbb{E}_{q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\frac{p(\boldsymbol{x}_{0:T})}{q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \right] \geq \mathbb{E}_{q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p(\boldsymbol{x}_{0:T})}{q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \right]$$

$$\log p(\boldsymbol{x}_0) \geq \mathbb{E}_{q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p(\boldsymbol{x}_{0:T})}{q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \right]$$

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

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

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

Evidence lower bound

$$\mathbb{E}_{q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p(\boldsymbol{x}_0|\boldsymbol{x}_1)}{q_\phi(\boldsymbol{x}_T|\boldsymbol{x}_{T-1})}\right] = \underbrace{\mathbb{E}_{q_\phi(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p(\boldsymbol{x}_0|\boldsymbol{x}_1)\right]}_{\text{reconstruction}}$$

$$\underbrace{- \mathbb{E}_{q_\phi(\boldsymbol{x}_T,\boldsymbol{x}_{T-1}|\boldsymbol{x}_0)}\left[\mathbb{D}_{\mathsf{KL}}\left(q_\phi(\boldsymbol{x}_T|\boldsymbol{x}_{T-1})||p(\boldsymbol{x}_T)\right)\right]}_{\text{prior matching}}$$

$$\mathbb{E}_{q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{\prod_{t=1}^{T-1} p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})}{\prod_{t=1}^{T-1} q_\phi(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right] = \sum_{t=1}^{T-1} \mathbb{E}_{q_\phi(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})}{q_\phi(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right]$$

$$= \sum_{t=1}^{T-1} \mathbb{E}_{q_\phi(\boldsymbol{x}_{t-1},\boldsymbol{x}_t,\boldsymbol{x}_{t+1}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})}{q_\phi(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right]$$

$$= \underbrace{\sum_{t=1}^{T-1} \mathbb{E}_{q_\phi(\boldsymbol{x}_{t-1},\boldsymbol{x}_{t+1}|\boldsymbol{x}_0)}\left[\mathbb{D}_{\mathsf{KL}}\left(q_\phi(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})||p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})\right)\right]}_{\text{Consistency}}$$

- We replace $p(\boldsymbol{x}_0|\boldsymbol{x}_1)$ with a neural network $p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)$ and $p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})$ with a neural network $p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})$

Bayes' theorem on consistency term

- Problem: $q_\phi(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ and $p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})$ run in opposite directions
- Solution via Bayes' theorem:

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

- We add a conditioning on $\boldsymbol{x}_0$:

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

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

Bayes' theorem on consistency term

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

- Minimizing this distribution (without proof) yields the mean of the distribution, setting the second derivative to zero yields the standard deviation:

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

$$\boldsymbol{\Sigma}_q(t) = \frac{(1-\alpha_t)(1-\sqrt{\hat{\alpha}_{t-1}})}{1-\bar{\alpha}_t}\mathbf{I}$$

- $q_\phi(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ is completely characterized by $\boldsymbol{x}_t$ and $\boldsymbol{x}_0$. No NN required to estimate the mean and variance!

New ELBO

$$\text{ELBO}_{\phi,\theta}(\boldsymbol{x}) = \underbrace{\mathbb{E}_{q_\phi(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)\right]}_{\text{reconstruction}} - \underbrace{\mathbb{D}_{\text{KL}}\left(q_\phi(\boldsymbol{x}_T|\boldsymbol{x}_{T-1})||p(\boldsymbol{x}_T)\right)}_{\text{new prior matching}}$$
$$- \underbrace{\sum_{t=2}^{T} \mathbb{E}_{q_{\phi(\boldsymbol{x}_t|\boldsymbol{x}_0)}}\left[\mathbb{D}_{\text{KL}}\left(q_\phi(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\right)\right]}_{\text{new consistency}}$$

- **Reconstruction**: we are maximizing the likelihood
- **Prior matching**: nothing to train
- **Consistency**: Instead of asking a forward transition to match with a reverse transition, we use q_ϕ to construct a reverse transition and use it to match with p_θ

Consistency term

$$q_\phi(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\Big(\underbrace{\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)}_{\text{known}}, \underbrace{\sigma_q^2(t)\mathbf{I}}_{\text{known}} \Big)$$

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}\Big(\underbrace{\boldsymbol{\mu}_\theta(\boldsymbol{x}_t)}_{\text{not known}}, \underbrace{\sigma_q^2(t)\mathbf{I}}_{\text{known}} \Big)$$

- The KL divergence is simplified to:

$$\begin{aligned}
&\mathbb{D}_{\text{KL}}\left(q_\phi(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\right) \\
&= \mathbb{D}_{\text{KL}}\Big(\mathcal{N}\big(\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0), \sigma_q^2(t)\mathbf{I}\big)||\mathcal{N}\big(\boldsymbol{\mu}_\theta(\boldsymbol{x}_t), \sigma_q^2(t)\mathbf{I}\big)\Big) \\
&= \frac{1}{2\sigma_q^2(t)}\|\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) - \boldsymbol{\mu}_\theta(\boldsymbol{x}_t)\|^2
\end{aligned}$$

ELBO for training

- We need to find a network that minimizes: $\frac{1}{2\sigma_q^2(t)}\|\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) - \boldsymbol{\mu}_\theta(\boldsymbol{x}_t)\|^2$, where

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

- To make it more convenient, we choose:

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

- This yields:

$$\begin{aligned}
&\frac{1}{2\sigma_q^2(t)}\|\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) - \boldsymbol{\mu}_\theta(\boldsymbol{x}_t)\|^2 \\
&= \frac{1}{2\sigma_q^2(t)}\frac{(1-\alpha_t)^2\bar{\alpha}_{t-1}}{(1-\bar{\alpha}_t)^2}\|\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t) - \boldsymbol{x}_0\|^2
\end{aligned}$$

ELBO for training

- For DDPM training, we use:

$$\text{ELBO}_\theta(\boldsymbol{x}) = \underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)\right]}_{\text{reconstruction}}$$

$$- \underbrace{\sum_{t=2}^{T} \mathbb{E}_{q_{\phi(\boldsymbol{x}_t|\boldsymbol{x}_0)}}\left[\mathbb{D}_{\text{KL}}\left(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\right)\right]}_{\text{consistency}}$$

$$= \underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)\right]}_{\text{reconstruction}}$$

$$- \sum_{t=2}^{T} \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2 \bar{\alpha}_{t-1}}{(1-\bar{\alpha}_t)^2} \left\|\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t) - \boldsymbol{x}_0\right\|^2$$

- We use:

$$\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1) = \log \mathcal{N}\left(\boldsymbol{\mu}_\theta(\boldsymbol{x}_1), \sigma_q^2(1)\mathbf{I}\right) \propto -\frac{1}{2\sigma_q^2(1)} \left\|\boldsymbol{\mu}_\theta(\boldsymbol{x}_1) - \boldsymbol{x}_0\right\|^2$$

$$= -\frac{1}{2\sigma_q^2(1)} \left\|\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_1) - \boldsymbol{x}_0\right\|^2$$

ELBO for training

- Using $\alpha_0 = 1$ and $\hat{\alpha}_1 = \alpha_1$, we obtain:

$$\text{ELBO}_\theta = -\sum_{t=1}^{T} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \left[\frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2 \bar{\alpha}_{t-1}}{(1-\bar{\alpha}_t)^2} \|\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t) - \boldsymbol{x}_0\|^2 \right]$$

- Therefore, the training objective minimizes:

$$\theta^* = \text{argmin} \sum_{t=1}^{T} \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2 \bar{\alpha}_{t-1}}{(1-\bar{\alpha}_t)^2} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \|\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t) - \boldsymbol{x}_0\|^2$$

- Ignoring the constants and expectations, the objective is:

$$\underset{\theta}{\text{argmin}} \|\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t) - \boldsymbol{x}_0\|$$

DDPM training

- $\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}$: We are not denoising any random noisy image. Instead, we the noisy image is fully defined as: $\boldsymbol{x}_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$
- $\frac{1}{\sigma_q^2(t)} \frac{(1-\alpha_t)^2 \bar{\alpha}_{t-1}}{(1-\bar{\alpha}_t)^2}$: controls the relative emphasis on each denoising loss. However, only minor impact in practice, usually dropped
- $\sum_{t=1}^{T}$: the summation can be replaced by a uniform distribution $t \sim \text{Uniform}[1, T]$

DDPM training

- For every image $\boldsymbol{x}_0$ in your training set:
  - Repeat the following steps until convergence
  - Pick a random time stamp $t \sim \text{Uniform}[1, T]$
  - Draw a sample $\boldsymbol{x}_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$
  - Take gradient descent step on:

$$\nabla_\theta \|\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t) - \boldsymbol{x}_0\|^2$$

DDPM inference

- We start with white noise vector $\boldsymbol{x}_T \sim \mathcal{N}(0, \mathbf{I})$
- For $t = T, T-1, \ldots, 1$:
  - Calculate $\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t)$ using our trained denoiser
  - Update $\boldsymbol{x}_t$ following:

$$\begin{aligned}\boldsymbol{x}_{t-1} &\sim p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) \\ &= \frac{(1-\bar{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\bar{\alpha}_t}\boldsymbol{x}_t + \frac{(1-\alpha_t)\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t}\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t) + \sigma_q(t)\boldsymbol{z}\end{aligned}$$

  - $\boldsymbol{z} \sim \mathcal{N}(0, \mathbf{I})$
  - Remember, we chose:

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

DDPM training - based on noise prediction

- Reformulate diffusion to predict the noise instead of the signal:

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0 \Rightarrow \boldsymbol{x}_0 = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0}{\sqrt{\bar{\alpha}_t}}$$

- Substituting into

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

gives (after several steps):

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

- Thus, we design our mean estimator to match this form:

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

DDPM training - based on noise prediction

- In Ho et al, the new training objective becomes:

$$\text{ELBO}_\theta = -\sum_{t=1}^{T} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \left[\frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2 \bar{\alpha}_{t-1}}{(1-\bar{\alpha}_t)^2} \|\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{\epsilon}_t) - \boldsymbol{\epsilon}_0\|^2 \right]$$

- For every image $\boldsymbol{x}_0$ in your training set:
  - Repeat the following steps until convergence
  - Pick a random time stamp $t \sim \text{Uniform}[1, T]$
  - Draw a sample $\boldsymbol{x}_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$
  - Take gradient descent step on:

$$\nabla_\theta \|\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t) - \boldsymbol{\epsilon}_0\|^2$$

DDPM inference - based on noise prediction

- The inference step can be derived through:

$$\begin{aligned}\boldsymbol{x}_{t-1} \sim p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) &= \mathcal{N}(\boldsymbol{\mu}_\theta(\boldsymbol{x}_t), \sigma_q^2(t)\mathbf{I}) \\ &= \boldsymbol{\mu}_\theta(\boldsymbol{x}_t) + \sigma_q^2(t)\boldsymbol{z} \\ &= \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t) + \sigma_q(t)\boldsymbol{z}\end{aligned}$$

- We start with white noise vector $\boldsymbol{x}_T \sim \mathcal{N}(0, \mathbf{I})$
- For $t = T, T-1, \ldots, 1$:
  - Calculate $\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t)$ using our trained denoiser
  - Update $\boldsymbol{x}_t$ following:

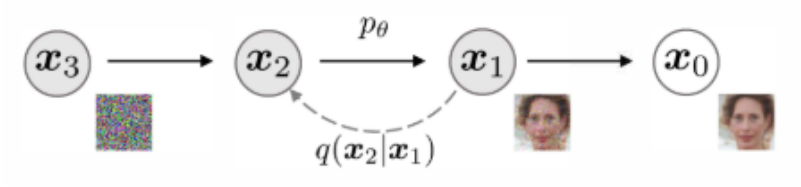
$$\begin{aligned}\boldsymbol{x}_{t-1} &\sim p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) \\ &= \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t) + \sigma_q(t)\boldsymbol{z}\end{aligned}$$

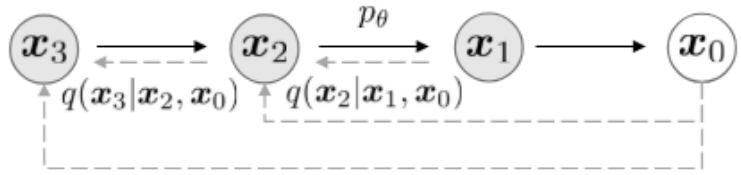
DDPMs are slow

- One of the most prevalent drawbacks of DDPM is that they need a large number of iterations to generate a reasonably good looking samples.
- If the reverse diffusion process intrinsically requires many steps to converge, then it will take us many denoising steps. Therefore, to speed up the computing, it is necessary to reduce the number of iterations.

Generalization of DDPMs

- Generalization of DDPMs via a non-Markovian forward process





- Song et al propose a family of backward processes:

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\frac{\boldsymbol{x}_t - \alpha_t\boldsymbol{x}_0}{\sqrt{1-\alpha_t}}, \sigma_t^2\mathbf{I}\right)$$

- σ_t modulates the stochasticity of the process:
  - If $\sigma_t = \sqrt{\frac{(1-\alpha_{t-1})}{(1-\alpha_t)}}\sqrt{1-\frac{\alpha_t}{\alpha_{t-1}}}$, we obtain the original DDPM
  - If $\sigma_t = 0$, the forward process becomes Markovian and the backward process becomes deterministic; the same original noise leads to the same image

Derivation of DDIMs (sketch)

- The original DDPM transition probabilities are:

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

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

- One important observation is that the transition probability $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ follows a Markov chain
- The Markov property ensures that the system is memoryless, i.e., once we know $\boldsymbol{x}_{t-1}$, we know $\boldsymbol{x}_t$
- Downside: a Markov chain can take many steps to converge

Derivation of DDIMs (sketch)

- Choice: $\alpha_t = \dfrac{\alpha_t}{\alpha_{t-1}}$
- I.e., $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}\left(\boldsymbol{x}_t|\sqrt{\frac{\alpha_t}{\alpha_{t-1}}}\boldsymbol{x}_{t-1}, (1-\frac{\alpha_t}{\alpha_{t-1}})\mathbf{I}\right)$
- The product simplifies to $\bar{\alpha}_t = \prod_{i=1}^{t} \frac{\alpha_i}{\alpha_{i-1}} = \alpha_t$
- Thus, $q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{x}_t|\sqrt{\alpha_t}\boldsymbol{x}_0, (1-\alpha_t)\mathbf{I}\right)$
- $\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_0 + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}$ and $\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}}\boldsymbol{x}_0 + \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}$
- $\Rightarrow \boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}}\boldsymbol{x}_0 + \sqrt{1-\alpha_{t-1}}\left(\frac{\boldsymbol{x}_t - \sqrt{\alpha_t}\boldsymbol{x}_0}{\sqrt{1-\alpha_t}}\right)$
- We have replaced the Gaussian by an estimate

Derivation of DDIMs (sketch)

- Given:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\alpha_t}\boldsymbol{x}_0, (1-\alpha_t)\mathbf{I}\right)$$

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_0 + \sqrt{1-\alpha_{t-1}}\left(\frac{\boldsymbol{x}_t - \sqrt{\alpha_t}\boldsymbol{x}_0}{\sqrt{1-\alpha_t}}\right), \sigma_t^2\mathbf{I}\right)$$

- Can we ensure that

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_0, (1-\alpha_{t-1})\mathbf{I}\right)?$$

- The solution is a family of backward processes:

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\frac{\boldsymbol{x}_t - \alpha_t\boldsymbol{x}_0}{\sqrt{1-\alpha_t}}, \sigma_t^2\mathbf{I}\right)$$

DDIM

- Special case when $\sigma_t = 0$ for all t
- The forward process becomes deterministic given $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_0$, except for $t = 1$
- Additionally, no random noise σ_q in the reverse process
- The resulting model becomes an **implicit probabilistic model**, where samples are generated from latent variables with a fixed procedure (from $\boldsymbol{x}_T$ to $\boldsymbol{x}_0$)
- This is named **denoising diffusion implicit model** (DDIM), because it is an implicit probabilistic model trained with the DDPM objective (despite the forward process no longer being a diffusion)

Generalized training objective

- Training objectives are equivalent for any value of σ_t:

$$\mathbb{E}_{\boldsymbol{x}_0, q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \left[\mathbb{D}_{\mathsf{KL}} \left(q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) || p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) \right) \right]$$
$$= \sqrt{\alpha_t}\boldsymbol{x}_0 + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_0 \left[\frac{\|\boldsymbol{\epsilon}_0 - \hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t)\|^2}{2\sigma_t^2 \alpha_t} \right]$$

- A model trained for the original DDPM process can be used for any process of the family

Generalized sampling method

- Sampling becomes:

$$\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}} \left(\frac{\boldsymbol{x}_t - \sqrt{1-\alpha_t}\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t)}{\sqrt{\alpha_t}} \right) + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t) + \sigma_t \boldsymbol{\epsilon}_t$$

- Different choices of σ_t result in different generative processes, all for the same model $\boldsymbol{\epsilon}_\theta$
- **Accelerated generative processes**: we define the forward process not on all the latent variables $\boldsymbol{x}_{1:T}$, but only on a subset $\{\boldsymbol{x}_{\tau_1}, \dots, \boldsymbol{x}_{\tau_S}\}$, where τ is an increasing sub-sequence of $[1, \dots, T]$ of length S
  - $q(\boldsymbol{x}_{\tau_i}|\boldsymbol{x}_0) = \mathcal{N}(\sqrt{\alpha_{\tau_i}}\boldsymbol{x}_0, (1-\alpha_{\tau_i})\mathbf{I})$ matches the "marginals"
  - This achievies significant speed-ups when the length of the sampling trajectory is much smaller than T

Relevance to neural ODEs

- We can rewrite the sampling to:

$$\frac{\boldsymbol{x}_{t-\Delta t}}{\sqrt{\alpha_{t-\Delta t}}} = \frac{\boldsymbol{x}_t}{\sqrt{\alpha_t}} + \left(\sqrt{\frac{1-\alpha_{t-\Delta t}}{\alpha_{t-\Delta t}}} - \sqrt{\frac{1-\alpha_t}{\alpha_t}} \right) \hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t)$$

- With enough discretization steps, we can reverse the generation process (going from $t = 0$ to T)
  - This encodes $\boldsymbol{x}_0$ to $\boldsymbol{x}_T$ and simulates the reverse of the ODE
  - We can use DDIM to obtain encodings of the observations (as the form of $\boldsymbol{x}_T$)
  - Useful for other downstream applications that require latent representations of a model

Literature

- Important papers:
  - First 2015 diffusion paper
  - Jonathan Ho - DDPM
  - Jiaming Song - DDIM
  - Latent diffusion
  - Hoogeboom - look at the appendix Step-by-Step Diffusion: An Elementary Tutorial