

Diffusion Models

Andreas Bagge & Gustav Rørhauge

October 24, 2023

1 Introduction

Diffusion models are a class of generative models that aim to learn the latent structure of complex data, such as images. These latent structures are underlying structures that are used in the generative process behind the data. Generative models are models that can generate new data similar to the data on which they are trained. They have many potential applications, such as data augmentation, image synthesis, video generation, molecule design, and text-to-image generation. However, generative modeling is also a very challenging task, as it requires capturing the high-dimensional and multimodal distribution of natural data.

Diffusion models are based on the idea of reversing a diffusion process, which is a stochastic process that gradually adds noise to the data until it reaches a predefined noise level. The diffusion process can be seen as a way of destroying the information in the data while preserving some of its statistical properties. By learning to reverse this process, diffusion models can recover the original data from the noisy data, and thus generate new data from pure noise.

2 The forward process

The diffusion model can be interpreted as a two-part system: A forward and a backward process. The forward process steadily noisifies the images in a Markovian chain, where a series of $t = 0 \dots T$ timesteps following a noising schedule transforms the image into new images that more and more closely resemble pure Gaussian noise. The forward process can be described as:

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

Where β is a fixed schedule parameter (although it can be learned). Simply put, β_t is just a scalar value pertaining to the timestep t .

Imagine that you have some image that is then flattened into an n -dimensional vector:

$$\mathbf{x}_t = [\mathbf{x}_{t,0}, \mathbf{x}_{t,1}, \dots, \mathbf{x}_{t,n}]$$

such that n is the number of pixels in the image and such that $\mathbf{x}_{t,l}$ is the l 'th pixel in the image. Then, a slightly more noisy version of \mathbf{x}_t can be calculated by sampling a new pixel-value using $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ and the parameterization trick for each pixel in \mathbf{x}_t :

$$\mathbf{x}_{t+1} = \sqrt{1 - \beta_t}\mathbf{x}_t + \beta_t\mathbf{I}$$

Such that each new pixel-value, $\mathbf{x}_{t+1,l}$ is drawn from a normal distribution with mean $\sqrt{1 - \beta_t}\mathbf{x}_{t,l}$ and standard deviation β_t .

Obviously, there are many ways to make data more and more noisy and this is just one of them. However, picking the forward process to follow a Gaussian distribution will, as usual, give some very nice properties later on. Each image in the Markov chain depends only on the prior image. So, given

\mathbf{x}_0 , our ground truth image, the following images can be written in a sequence as:

$$\begin{aligned}
\mathbf{x}_0 &\sim q(\mathbf{x}_0) \\
\mathbf{x}_1 &\sim \mathcal{N}(\sqrt{1-\beta_1}\mathbf{x}_0, \beta_1\mathbf{I}) \\
\mathbf{x}_2 &\sim \mathcal{N}(\sqrt{1-\beta_2}\mathbf{x}_1, \beta_2\mathbf{I}) \\
&\vdots \\
\mathbf{x}_{T-1} &\sim \mathcal{N}(\sqrt{1-\beta_{T-1}}\mathbf{x}_{T-2}, \beta_{T-1}\mathbf{I}) \\
\mathbf{x}_T &\sim \mathcal{N}(\sqrt{1-\beta_T}\mathbf{x}_{T-1}, \beta_T\mathbf{I})
\end{aligned}$$

Where $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ is drawn from the true distribution of the data, i.e. it is an image from our dataset. Even though the forward process is totally fixed, it is important to understand that it is not deterministic; we don't know the exact values of $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T$, only their distributions.

So, to obtain \mathbf{x}_t , we have to fist obtain $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{t-1}$. This can be a costly affair, especially for large T and for large data. However, due to our choice of noise scheduling, this can be circumvented by the "repeated reparameterization trick". Instead of sampling t times to obtain \mathbf{x}_t , it is possible to go from an input image \mathbf{x}_0 directly to any timestep in the forward process: First, we'll introduce α_t :

$$\alpha_t = 1 - \beta_t \tag{2}$$

$$\tag{3}$$

Using α_t , the forward process can be rewritten as:

$$\mathbf{x}_t \sim \mathcal{N}(\sqrt{\alpha_t}\mathbf{x}_{t-1}, (1 - \alpha_t)\mathbf{I})$$

Sampling from a normal distribution (reparameterization trick)

Given:

$$p(x) = \mathcal{N}(\mu, \sigma^2)$$

A sample \mathbf{x} from p can be drawn by calculating:

$$x = \mu + \sigma\epsilon, \quad \epsilon \sim \mathcal{N}(0, 1)$$

Using the reparameterization trick, the next step in the forward process can also be expressed as:

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1 - \alpha_t}\epsilon_t$$

\mathbf{x}_{t-1} can be rewritten in the same way:

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1 - \alpha_{t-1}}\epsilon_{t-1}$$

Where ϵ_t and ϵ_{t-1} are independent and identically distributed. Combining the two expressions above we obtain:

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1 - \alpha_{t-1}}\epsilon_{t-1}) + \sqrt{1 - \alpha_t}\epsilon_t \\
&= \sqrt{\alpha_t}\sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \underbrace{\sqrt{\alpha_t}\sqrt{1 - \alpha_{t-1}}\epsilon_{t-1} + \sqrt{1 - \alpha_t}\epsilon_t}_{\text{Sum of Gaussians}} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\alpha_t - \alpha_{t-1}\alpha_t}\epsilon_{t-1} + \sqrt{1 - \alpha_t}\epsilon_t \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\alpha_t - \alpha_{t-1}\alpha_t}\epsilon_{t-1} + \sqrt{1 - \alpha_t}\epsilon_t \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1 - \alpha_t}\epsilon^*
\end{aligned}$$

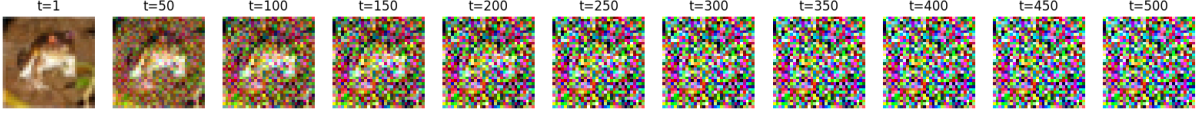


Figure 1: Gradually more noisy versions of the original \mathbf{x}_0 . Here, each image have been sampled from the distribution pertaining to the timestep t

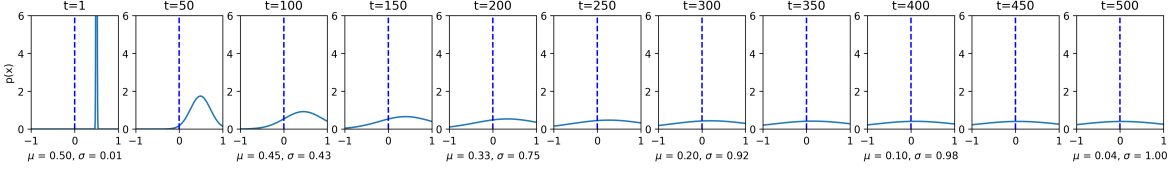


Figure 2: The distribution for the pixel value for each timestep given $\mathbf{x}_0 = 0.5$. Clearly, the distribution approaches a standard normal distribution.

Sum of centered Gaussians (i.i.d)

We'll now use the definition of a sum of centered Gaussians. The mean and standard deviation of the resulting Gaussian distribution is defined as follows:

$$X \sim \mathcal{N}(0, \sigma_x^2) \quad (4)$$

$$Y \sim \mathcal{N}(0, \sigma_y^2) \quad (5)$$

$$X + Y \sim \mathcal{N}(0, \sigma_x^2 + \sigma_y^2) \quad (6)$$

Given that X and Y are not correlated.

Now, the above can be repeated by writing the expression for \mathbf{x}_{t-2} using the reparameterization trick. This can be done all the way down to \mathbf{x}_0 by when we stop. The expression for \mathbf{x}_t will then be:

$$\begin{aligned} \mathbf{x}_t &= \sqrt{\alpha_t \alpha_{t-1} \dots \alpha_1} \mathbf{x}_0 + \sqrt{1 - \alpha_t \alpha_{t-1} \dots \alpha_1} \epsilon^* \\ &= \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon^* \\ q(\mathbf{x}_t | \mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_t, \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I}) \end{aligned}$$

Where $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$. We can visualize what happens with an image when running it through the encoder by sampling from the distribution pertaining to each timestep. For example, if we let $\mathbf{x}_0 = \text{frog}$, we can obtain the noisy images visualized in figure 1. Another way of interpreting what exactly happens in the forward process is to look at the resulting distribution of some pixel value after each timestep. This change in distribution can be seen in figure 2. Evidently, the distribution of the pixel gets shifted closer and closer towards a standard normal distribution. This also explains why the images become more and more noisy; they carry less and less information from the original image and come closer and closer to Gaussian noise.

2.1 Other types of noise schedules

The purpose of the noise scheduler is to gradually add more and more noise to the original picture such that the final image resembles Gaussian noise. Remember how we parameterized each image in the chain of noisy images:

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

So, given that we want \mathbf{x}_t to resemble \mathbf{x}_0 for small $t \approx 0$, and we want \mathbf{x}_t to resemble $\mathcal{N}(0, 1)$ for big $t \approx T$, it should be obvious that we want to model $\bar{\alpha}_t$ such that $\bar{\alpha}_1 \approx 1$ and $\bar{\alpha}_T \approx 0$, and such that

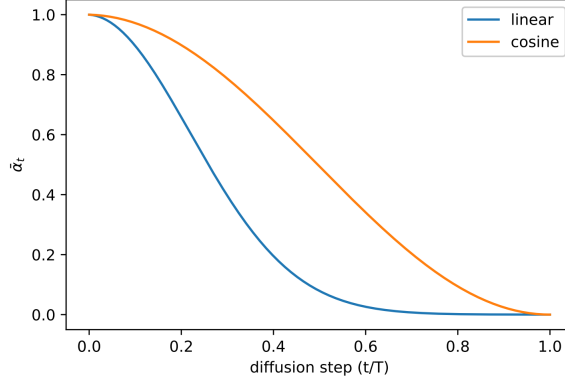


Figure 3: Comparison of the behavior of $\bar{\alpha}_t$ depending on the type of noise schedule. Obtained from: [Improved Denoising Diffusion Probabilistic Models](#)

$\bar{\alpha}_t$ is monotonously decreasing on the range from $t = 1 \dots T$. This is also exactly the case using the linear noise schedule used in the chapter above. But obviously, there are other ways of defining the noise schedule such that $\bar{\alpha}_t$ obtains the sought-after behavior. For example, in [Improved Denoising Diffusion Probabilistic Models](#) they define $\bar{\alpha}_t$ to follow a cosine wave defined as:

$$\bar{\alpha}_t = \frac{f(t)}{f(0)}, \quad f(t) = \cos \left(\frac{t/T + s}{1 + s} \cdot \frac{\pi}{2} \right)^2$$

The authors discover that this noise schedule performs better than the "traditional" linear noise schedule. A comparison between the behavior of linear and cosine noise schedules can be seen in figure 3.

2.2 What is our goal?

Until now, we have formulated the forward process of our model. That is, we have formulated a way of gradually producing more and more noisy images originating from some ground truth image called \mathbf{x}_0 such that the final image resembles simple Gaussian noise. Our goal is now to go backward and try to reproduce the original image. In other words, in the forward process, we modeled the next image based on the prior, such that we had $q(\mathbf{x}_t|\mathbf{x}_{t-1})$. Now, we want to model the prior image based on the next one; we want to obtain the distribution of $p(\mathbf{x}_{t-1}|\mathbf{x}_t)$. Just like in the forward process, we would also this backward process to produce new images in a discrete manner. The difference is now that we want the images to become less and less noisy. Also, like in the forward process, we actually do not want to learn the image pertaining to each timestep in the backward process but rather the distribution of the image. One question is how to describe this backward process, another is how to optimize it. To do this, we need to use maximum likelihood estimation.

3 The ELBO

Imagine that we have some data denoted \mathbf{x} . These pictures are our "ground truth" meaning we interpret them as having no noise. In the following, we will specifically denote the data \mathbf{x} as \mathbf{x}_0 when the characteristic of "no noise" is important. In other words, they are the 0'th element in our Markov chain before any noise is added. As usual, we wish to model the distribution of these data, $p(\mathbf{x}) = p(\mathbf{x}_0)$. One way of doing this is by marginalization:

$$p(\mathbf{x}_0) = \int \int \dots \int p(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_t) d\mathbf{x}_1 d\mathbf{x}_2 \dots d\mathbf{x}_t$$

To make this notation prettier, we will introduce $\mathbf{x}_{0:T} = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_T)$. Also, we will simply write one integral which implicitly contains the others. Then, we get:

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

We now want to manipulate the above expression until we get to some expression that can be evaluated on a computer and that somehow involves the mentioned backward process that we are interested in. To do this, we do a trick: we multiply by $\frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} = 1$:

$$p(\mathbf{x}_0) = \int p(\mathbf{x}_{0:T}) \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} d\mathbf{x}_{1:T}$$

Expectation value

The expectation value is defined as:

$$\mathbb{E}_{p(x)}[f(x)] = \int p(x)f(x) dx$$

Where p is a probability density function over x and f is some function of x .

If we let $p = q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ and $f = \frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}$, then:

$$p(\mathbf{x}_0) = \int p(\mathbf{x}_{0:T}) \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} d\mathbf{x}_{1:T} = \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right]$$

In a second, we'll take the logarithm on both sides of the above equation. Logarithms usually make expressions easier to deal with since products can be split into sums. Also, in our case, it opens the possibility of using Jensen's Inequality to simplify things further:

Jensens Inequality

Jensen's inequality says that:

$$\log(\mathbb{E}[\mathbf{x}]) \geq \mathbb{E}[\log(\mathbf{x})]$$

Jensen's equality sometimes helps get a lower bound on expressions that are otherwise intractable.

We take the logarithm on both sides and apply Jensen's inequality:

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

Great! Now, let's expand the expressions in the fraction:

$$\begin{aligned} p(\mathbf{x}_{0:T}) &= p(\mathbf{x}_0, \mathbf{x}_1 \cdots \mathbf{x}_T) \\ &= p(\mathbf{x}_0|\mathbf{x}_{1:T})p(\mathbf{x}_1|\mathbf{x}_{2:T}) \cdots p(\mathbf{x}_{T-1}|\mathbf{x}_T)p(\mathbf{x}_T) \\ &= p(\mathbf{x}_0|\mathbf{x}_1)p(\mathbf{x}_1|\mathbf{x}_2) \cdots p(\mathbf{x}_{T-1}|\mathbf{x}_T)p(\mathbf{x}_T) \\ &= p(\mathbf{x}_T) \prod_{t=1}^T p(\mathbf{x}_{t-1}|\mathbf{x}_t) \end{aligned} \tag{7}$$

We can rewrite the probabilities on the form $p(\mathbf{x}_t|\mathbf{x}_{(t+1):T})$ as $p(\mathbf{x}_t|\mathbf{x}_{t+1})$ due to our assumptions, that is the that the distribution of some noisy image \mathbf{x}_t ONLY depends on the prior image \mathbf{x}_{t+1} .

Now, let us look at the denominator:

$$\begin{aligned}
q(\mathbf{x}_{1:T}|\mathbf{x}_0) &= \frac{q(\mathbf{x}_{1:T}, \mathbf{x}_0)}{q(\mathbf{x}_0)} \\
&= \frac{q(\mathbf{x}_0, \mathbf{x}_1 \cdots \mathbf{x}_T)}{q(\mathbf{x}_0)} \\
&= \frac{q(\mathbf{x}_T|\mathbf{x}_{(T-1):0})q(\mathbf{x}_{T-1}|\mathbf{x}_{(T-2):0}) \cdots q(\mathbf{x}_1|\mathbf{x}_0)q(\mathbf{x}_0)}{q(\mathbf{x}_0)} \\
&= \frac{q(\mathbf{x}_T|\mathbf{x}_{T-1})q(\mathbf{x}_{T-1}|\mathbf{x}_{T-2}) \cdots q(\mathbf{x}_1|\mathbf{x}_0)q(\mathbf{x}_0)}{q(\mathbf{x}_0)} \\
&= q(\mathbf{x}_T|\mathbf{x}_{T-1})q(\mathbf{x}_{T-1}|\mathbf{x}_{T-2}) \cdots q(\mathbf{x}_1|\mathbf{x}_0) \\
&= \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})
\end{aligned} \tag{8}$$

Once again, we used the properties of the Markov chain to simplify the above. We'll insert these new expressions in the fraction:

$$\begin{aligned}
\log(p(\mathbf{x}_0)) &\geq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \left(\frac{p(\mathbf{x}_T) \prod_{t=1}^T p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{\prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right) \right] \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \left(\frac{p(\mathbf{x}_T)p(\mathbf{x}_0|\mathbf{x}_1) \prod_{t=2}^T p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_T|\mathbf{x}_{T-1}) \prod_{t=1}^{T-1} q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right) \right] \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \left(\frac{p(\mathbf{x}_T)p(\mathbf{x}_0|\mathbf{x}_1) \prod_{t=1}^{T-1} p(\mathbf{x}_t|\mathbf{x}_{t+1})}{q(\mathbf{x}_T|\mathbf{x}_{T-1}) \prod_{t=1}^{T-1} q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right) \right] \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{p(\mathbf{x}_T)p(\mathbf{x}_0|\mathbf{x}_1)}{p(\mathbf{x}_T|\mathbf{x}_{T-1})} \right] + \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \prod_{t=1}^{T-1} \frac{p(\mathbf{x}_t|\mathbf{x}_{t+1})}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} [\log p(\mathbf{x}_0|\mathbf{x}_1)] + \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_{T-1})} \right] + \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\sum_{t=1}^{T-1} \log \frac{p(\mathbf{x}_t|\mathbf{x}_{t+1})}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] \\
&= \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} [\log p(\mathbf{x}_0|\mathbf{x}_1)] + \mathbb{E}_{q(\mathbf{x}_{T-1}, \mathbf{x}_T|\mathbf{x}_0)} \left[\log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_{T-1})} \right] + \mathbb{E}_{q(\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{x}_0)} \sum_{t=1}^{T-1} \left[\log \frac{p(\mathbf{x}_t|\mathbf{x}_{t+1})}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right]
\end{aligned} \tag{9}$$

$$\begin{aligned}
\log(p(\mathbf{x}_0)) &\geq \underbrace{\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} [\log p(\mathbf{x}_0|\mathbf{x}_1)]}_{\text{reconstruction term}} - \underbrace{\mathbb{E}_{q(\mathbf{x}_{T-1}|\mathbf{x}_0)} [D_{KL}(q(\mathbf{x}_T|\mathbf{x}_{T-1})||p(\mathbf{x}_T))]}_{\text{prior matching term}} \\
&\quad - \sum_{t=1}^{T-1} \underbrace{\mathbb{E}_{q(\mathbf{x}_{t-1}, \mathbf{x}_{t+1}|\mathbf{x}_0)} [D_{KL}(q(\mathbf{x}_t|\mathbf{x}_{t-1})||p(\mathbf{x}_t|\mathbf{x}_{t+1}))]}_{\text{consistency term}}
\end{aligned}$$

Now, realize that the following must hold, once again due to the Markov chain:

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

Properties of Markov chain

Due to the Markov chain, the following is true:

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, a) = p(\mathbf{x}_{t-1}|\mathbf{x}_t) \quad \text{and} \quad q(\mathbf{x}_t|\mathbf{x}_{t-1}, a) = q(\mathbf{x}_t|\mathbf{x}_{t-1})$$

For any a .

That is, that every \mathbf{x}_t only depends on \mathbf{x}_{t-1} (as described above). Therefore, we can condition on whatever variables we want. Let us insert this and get:

$$\begin{aligned}
\log(p(\mathbf{x}_0)) &\geq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \left(\frac{p(\mathbf{x}_T)p(\mathbf{x}_0|\mathbf{x}_1) \prod_{t=2}^T p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_1|\mathbf{x}_0) \prod_{t=2}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)} \right) \right] \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \left(\frac{p(\mathbf{x}_T)p(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} \prod_{t=2}^T \frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)} \right) \right] \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \left(\frac{p(\mathbf{x}_T)p(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} \right) + \log \left(\prod_{t=2}^T \frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)} \right) \right]
\end{aligned} \tag{10}$$

We can rewrite $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)$ using Bayes:

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

This is also inserted into the expression. Thereafter, we can go on algebra-autopilot:

$$\begin{aligned}
\log(p(\mathbf{x}_0)) &\geq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \left(\frac{p(\mathbf{x}_T)p(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} \right) + \log \left(\prod_{t=2}^T \frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}} \right) \right] \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \left(\frac{p(\mathbf{x}_T)p(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} \right) + \log \left(\prod_{t=2}^T \frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t)q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_0)} \right) \right] \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \left(\frac{p(\mathbf{x}_T)p(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} \right) + \log \left(\prod_{t=2}^T \frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right) + \log \left(\prod_{t=2}^T \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \right) \right] \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \left(\frac{p(\mathbf{x}_T)p(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} \right) + \log \left(\prod_{t=2}^T \frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right) + \log \left(\frac{q(\mathbf{x}_1|\mathbf{x}_0)}{q(\mathbf{x}_T|\mathbf{x}_0)} \right) \right] \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \left(\frac{p(\mathbf{x}_T)p(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_T|\mathbf{x}_0)} \right) + \log \left(\prod_{t=2}^T \frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right) \right] \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log(p(\mathbf{x}_0|\mathbf{x}_1)) + \log \left(\frac{q(\mathbf{x}_T)}{p(\mathbf{x}_T|\mathbf{x}_0)} \right) + \log \left(\prod_{t=2}^T \frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right) \right] \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} [\log(p(\mathbf{x}_0|\mathbf{x}_1))] + \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \left(\frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} \right) \right] + \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \left(\prod_{t=2}^T \frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right) \right] \\
&= \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} [\log(p(\mathbf{x}_0|\mathbf{x}_1))] + \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \left[\log \left(\frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} \right) \right] + \sum_{t=2}^T \mathbb{E}_{q(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{x}_0)} \left[\log \left(\frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right) \right]
\end{aligned} \tag{11}$$

Please note how the expectation in the first term changes from $\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}(\dots)$ to $\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)}(\dots)$, and similar for the other expectations. This can be derived from the fact that the function inside the expectation value only contains a subset of the conditional events.

Expectation value over a subset of conditional events

Conditional events can be ignored in the distribution involved in an expectation value if the said conditional events aren't included in the expression inside the expectation value. This can formally be described as:

$$\begin{aligned}
 \mathbb{E}_{q(a,b|c)}(f(a)) &= \int \int q(a,b|c) f(a) da db \\
 &= \int f(a) \int q(a,b|c) db da \\
 &= \int f(a) q(a|c) da \\
 &= \mathbb{E}_{q(a|c)}(f(a))
 \end{aligned} \tag{12}$$

Kullback-Leibler divergence

The Kullback-Leibler divergence (often shortened as KL-divergence) is a kind of similarity measure between distributions. A low Kullback-Leibler divergence means that the distributions are similar to each other. The KL-divergence is defined as:

$$D_{KL}(p(x)||q(x)) = - \int p(x) \log \left(\frac{q(x)}{p(x)} \right) dx = -\mathbb{E} \left[\log \left(\frac{q(x)}{p(x)} \right) \right]$$

We'll apply the KL-divergence to the expression above:

$$\log(p(\mathbf{x}_0)) \geq \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)}[\log(p(\mathbf{x}_0|\mathbf{x}_1))] - \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} [D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T))] + \sum_{t=2}^T \mathbb{E}_{q(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{x}_0)} \left[\log \left(\frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right) \right]$$

The very last term in the above expression can also be rewritten using the KL-divergence. This is not totally trivial though, and it requires that we rewrite the expectation value as the initial integral it arose from:

$$\begin{aligned}
 \mathbb{E}_{q(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{x}_0)} \left[\log \left(\frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right) \right] &= \int \int q(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{x}_0) \log \left(\frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right) d\mathbf{x}_t d\mathbf{x}_{t-1} \\
 &= \int \int q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) q(\mathbf{x}_t|\mathbf{x}_0) \log \left(\frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right) d\mathbf{x}_t d\mathbf{x}_{t-1} \\
 &= \int q(\mathbf{x}_t|\mathbf{x}_0) \int q(\mathbf{x}_{t-1}|\mathbf{x}_0) \log \left(\frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right) d\mathbf{x}_{t-1} d\mathbf{x}_t \\
 &= - \int q(\mathbf{x}_t|\mathbf{x}_0) D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p(\mathbf{x}_{t-1}|\mathbf{x}_t)) d\mathbf{x}_t \\
 &= -\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} [D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p(\mathbf{x}_{t-1}|\mathbf{x}_t))]
 \end{aligned} \tag{13}$$

Now, we can write our final lower bound for the logarithm of $p(\mathbf{x}_0)$. This expression is also called the evidence lower bound, shortened ELBO:

$$\begin{aligned}
 \log(p(\mathbf{x}_0)) &\geq \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)}[\log(p(\mathbf{x}_0|\mathbf{x}_1))] - \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} [D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T))] \\
 &\quad - \sum_{t=2}^T \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} [D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p(\mathbf{x}_{t-1}|\mathbf{x}_t))]
 \end{aligned}$$

Now, this is the expression for the ELBO, which we want to maximize. Oftentimes, we put a negative sign in front and the problem therefore becomes a minimization problem.

4 Interpreting the ELBO

Let us look at the different terms in the ELBO. First, let us name the terms appropriately:

$$\log(p(\mathbf{x}_0)) \geq \underbrace{\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)}[\log(p_\theta(\mathbf{x}_0|\mathbf{x}_1))]}_{L_0} - \underbrace{\mathbb{E}_{q(\mathbf{x}_T|\mathbf{x}_0)}[D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T))]}_{L_T} - \underbrace{\sum_{t=2}^T \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}[D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))]}_{L_{t-1}}$$

- L_0 is the reconstruction error. It examines whether the ground truth image \mathbf{x}_0 scores high in the distribution resulting from the second to last layer in the backward diffusion process, \mathbf{x}_1 .
- L_T is the "prior matching term". It examines whether the last layer in the forward diffusion process matches the prior. We haven't chosen a prior yet, but considering that we have chosen the diffusion process to go towards a standard normal, it would be natural to choose $p(\mathbf{x}_T) \sim \mathcal{N}(0, 1)$. This term has no trainable parameters and can therefore be ignored when training the model since it won't affect the gradients.
- L_{t-1} is the denoising matching term. It examines whether our learnable denoising transition step $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ matches the ground truth denoising step $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$. To minimize this term, the two distributions should be as close to each other as possible therefore minimizing the KL divergence between them. The question is: how do we parameterize these distributions?

4.1 Why can we ignore L_T

Let's discuss the term L_T . We will not allocate any learnable parameters to this term and in practice the importance of this will be negligible, assuming we intelligently choose our prior. It will not impact our model's choice of the optimal parameter values, θ^* . To be precise, we could choose to define $p(\mathbf{x}_T)$ as a standard, normal Gaussian. And since we have chosen our diffusion schedule such that \mathbf{x}_T approximately ends up looking like a standard Gaussian, this term will virtually be equal to zero anyways.

In order to convince you, and ourselves, that this term is virtually equal to zero during training as seen in [Denoising Diffusion Probabilistic Models](#), we have done some experimentation with the noising schedule and prior. It is important that the chosen prior distribution $p(\mathbf{x}_T)$ and the final distribution in the noising process are very similar, since our model is trying to work backwards from this prior. If these are too dissimilar, there is a dissonance between the stopping point of the forward process and the starting point of the backward process, thus making the learning problem infinitely harder. In order to investigate this we took an image (here from MNIST) and ran it through the forward process with $\beta = (0.0001, 0.02)$ and $T = 500$ and compared the resulting distribution to $p(\mathbf{x}_T)$ throughout, see figure 4. We use this to argue that satisfactory asymptotic behaviour is present and we can safely disregard this term in the ELBO. A visual representation of the noisified input image can be seen in 1.

4.2 Deriving the distributions in L_{t-1}

As mentioned earlier, we can minimize L_{t-1} if we let $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$. We are going to call $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ the "ground-truth denoising transition step"; this makes sense since it has access to the original image \mathbf{x}_0 . However, obviously, it is not possible to completely match our learnable denoising step with the ground-truth denoising step, simply because the ground-truth step has access to the ground-truth picture \mathbf{x}_0 and our learnable step doesn't. To overcome this problem, we will

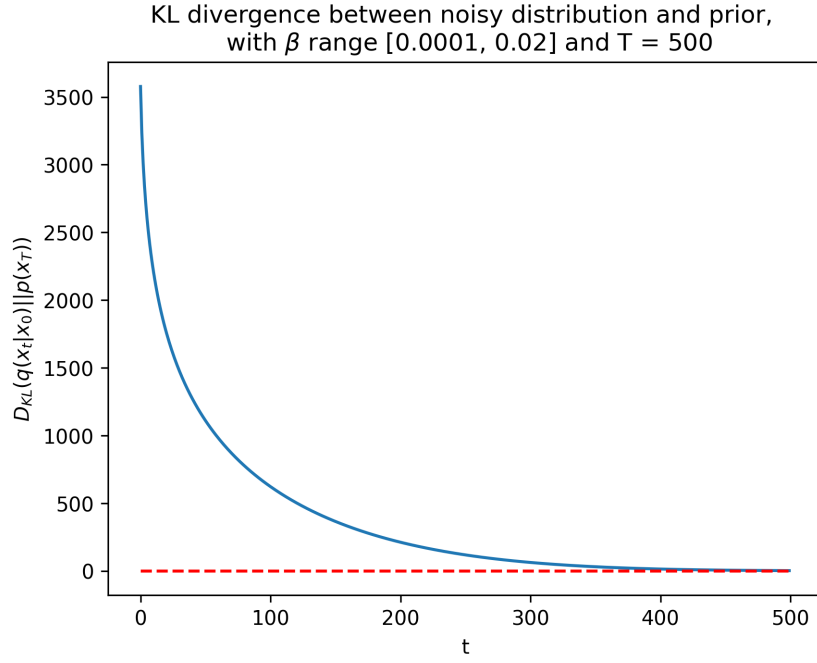


Figure 4: Comparison of KL divergences for each timestep.

derive the actual distribution of the ground-truth step using Bayes:

$$\begin{aligned}
q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) &= \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \\
&= \frac{\mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I})\mathcal{N}(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0, (1-\bar{\alpha}_{t-1})\mathbf{I})}{\mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})} \\
&\propto \exp \left\{ -\frac{1}{2} \left[\frac{(\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{(1-\alpha_t)} + \frac{(\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0)^2}{(1-\bar{\alpha}_{t-1})} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{(1-\bar{\alpha}_t)} \right] \right\} \\
&= \exp \left\{ -\frac{1}{2} \left[\frac{\mathbf{x}_t^2 + \alpha_t\mathbf{x}_{t-1}^2 - 2\mathbf{x}_t\sqrt{\alpha_t}\mathbf{x}_{t-1}}{(1-\alpha_t)} + \frac{\mathbf{x}_{t-1}^2 + \bar{\alpha}_{t-1}\mathbf{x}_0^2 - 2\mathbf{x}_{t-1}\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0}{(1-\bar{\alpha}_{t-1})} - \frac{\mathbf{x}_t^2 + \bar{\alpha}_t\mathbf{x}_0^2 - 2\mathbf{x}_t\sqrt{\bar{\alpha}_t}\mathbf{x}_0}{(1-\bar{\alpha}_t)} \right] \right\} \\
&= \exp \left\{ -\frac{1}{2} \left[\frac{\mathbf{x}_t^2}{(1-\alpha_t)} + \frac{\bar{\alpha}_{t-1}\mathbf{x}_0^2}{(1-\bar{\alpha}_t)} + \frac{\alpha_t\mathbf{x}_{t-1}^2 - 2\mathbf{x}_t\sqrt{\alpha_t}\mathbf{x}_{t-1}}{(1-\alpha_t)} + \frac{\mathbf{x}_{t-1}^2 - 2\mathbf{x}_{t-1}\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0}{(1-\bar{\alpha}_{t-1})} - \frac{\mathbf{x}_t^2 + \bar{\alpha}_t\mathbf{x}_0^2 - 2\mathbf{x}_t\sqrt{\bar{\alpha}_t}\mathbf{x}_0}{(1-\bar{\alpha}_t)} \right] \right\} \\
&= \exp \left\{ -\frac{1}{2} \left[\frac{\alpha_t\mathbf{x}_{t-1}^2 - 2\mathbf{x}_t\sqrt{\alpha_t}\mathbf{x}_{t-1}}{(1-\alpha_t)} + \frac{\mathbf{x}_{t-1}^2 - 2\mathbf{x}_{t-1}\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0}{(1-\bar{\alpha}_{t-1})} + C(\mathbf{x}_t, \mathbf{x}_0) \right] \right\} \\
&= \exp \left\{ -\frac{1}{2} \left[\frac{\alpha_t\mathbf{x}_{t-1}^2}{(1-\alpha_t)} - \frac{2\mathbf{x}_t\sqrt{\alpha_t}\mathbf{x}_{t-1}}{(1-\alpha_t)} + \frac{\mathbf{x}_{t-1}^2}{(1-\bar{\alpha}_{t-1})} - \frac{2\mathbf{x}_{t-1}\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0}{(1-\bar{\alpha}_{t-1})} + C(\mathbf{x}_t, \mathbf{x}_0) \right] \right\} \\
&= \exp \left\{ -\frac{1}{2} \left[\left(\frac{\alpha_t}{(1-\alpha_t)} + \frac{1}{(1-\bar{\alpha}_{t-1})} \right) \mathbf{x}_{t-1}^2 - 2 \left(\frac{\mathbf{x}_t\sqrt{\alpha_t}}{(1-\alpha_t)} + \frac{\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0}{(1-\bar{\alpha}_{t-1})} \right) \mathbf{x}_{t-1} \right] - \frac{1}{2}C(\mathbf{x}_t, \mathbf{x}_0) \right\} \\
&= \exp \left\{ -\frac{1}{2} \left[\left(\frac{\alpha_t(1-\bar{\alpha}_{t-1}) + 1 - \alpha_t}{(1-\alpha_t)(1-\bar{\alpha}_{t-1})} \right) \mathbf{x}_{t-1}^2 - 2 \left(\frac{\mathbf{x}_t\sqrt{\alpha_t}}{(1-\alpha_t)} + \frac{\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0}{(1-\bar{\alpha}_{t-1})} \right) \mathbf{x}_{t-1} \right] - \frac{1}{2}C(\mathbf{x}_t, \mathbf{x}_0) \right\} \\
&= \exp \left\{ -\frac{1}{2} \left[\left(\frac{(1-\bar{\alpha}_t)}{(1-\alpha_t)(1-\bar{\alpha}_{t-1})} \right) \mathbf{x}_{t-1}^2 - 2 \left(\frac{\mathbf{x}_t\sqrt{\alpha_t}}{(1-\alpha_t)} + \frac{\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0}{(1-\bar{\alpha}_{t-1})} \right) \mathbf{x}_{t-1} \right] - \frac{1}{2}C(\mathbf{x}_t, \mathbf{x}_0) \right\} \\
&= \exp \left\{ -\frac{1}{2} \left[\left(\frac{(1-\bar{\alpha}_t)}{(1-\alpha_t)(1-\bar{\alpha}_{t-1})} \right) \mathbf{x}_{t-1}^2 - 2 \left(\frac{\mathbf{x}_t\sqrt{\alpha_t}}{(1-\alpha_t)} + \frac{\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0}{(1-\bar{\alpha}_{t-1})} \right) \mathbf{x}_{t-1} \right] - \frac{1}{2}C(\mathbf{x}_t, \mathbf{x}_0) \right\}
\end{aligned} \tag{14}$$

$$\begin{aligned}
&= \exp \left\{ -\frac{1}{2} \left(\frac{(1-\bar{\alpha}_t)}{(1-\alpha_t)(1-\bar{\alpha}_{t-1})} \right) \left[\mathbf{x}_{t-1}^2 - 2 \frac{\frac{\mathbf{x}_t \sqrt{\alpha_t}}{(1-\alpha_t)} + \frac{\sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0}{(1-\bar{\alpha}_{t-1})}}{\left(\frac{(1-\bar{\alpha}_t)}{(1-\alpha_t)(1-\bar{\alpha}_{t-1})} \right)} \mathbf{x}_{t-1} \right] - \frac{1}{2} C(\mathbf{x}_t, \mathbf{x}_0) \right\} \\
&= \exp \left\{ -\frac{1}{2} \left(\frac{(1-\bar{\alpha}_t)}{(1-\alpha_t)(1-\bar{\alpha}_{t-1})} \right) \left[\mathbf{x}_{t-1}^2 - 2 \frac{\left(\frac{\mathbf{x}_t \sqrt{\alpha_t}}{(1-\alpha_t)} + \frac{\sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0}{(1-\bar{\alpha}_{t-1})} \right) (1-\alpha_t)(1-\bar{\alpha}_{t-1})}{(1-\bar{\alpha}_t)} \mathbf{x}_{t-1} \right] - \frac{1}{2} C(\mathbf{x}_t, \mathbf{x}_0) \right\} \\
&= \exp \left\{ -\frac{1}{2} \left(\frac{1}{\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{(1-\bar{\alpha}_t)}} \right) \left[\mathbf{x}_{t-1}^2 - 2 \left(\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\mathbf{x}_0}{(1-\bar{\alpha}_t)} \right) \mathbf{x}_{t-1} \right] - \frac{1}{2} C(\mathbf{x}_t, \mathbf{x}_0) \right\} \\
&\propto \mathcal{N} \left(\mathbf{x}_{t-1}; \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\mathbf{x}_0}{1-\bar{\alpha}_t}, \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \mathbf{I} \right)
\end{aligned} \tag{15}$$

In the second to last line, we implicitly include $C(\mathbf{x}_t, \mathbf{x}_0)$ and can therefore complete the square (the square is on the form $(x_{t-1} - \mu_q)^2$, so the curious reader will just have to confirm that $\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} C(\mathbf{x}_t, \mathbf{x}_0)$ is actually equal to μ_q^2).

Therefore, given some noisy picture \mathbf{x}_t , we can derive the distribution of the slightly less noisy picture \mathbf{x}_{t-1} and sample from this distribution. To make things shorter, we will define the following mean and variance:

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

To make the following derivations more readable, we'll denote $\Sigma_q(t) = \sigma_q^2(t) \mathbf{I}$, such that $\sigma_q^2(t) = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}$. Our goal is to match our learnable denoising step with the ground-truth denoising step. Since the ground-truth denoising step have been shown to follow a Gaussian distribution, we can likewise parameterize the learnable step as a Gaussian in almost the same way, and we can name the mean and variance of the this distributions appropriately. That is:

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

Here, we can realize that the "true" variance, $\Sigma_q(t) = \sigma_q^2(t)$, is a function of only t and therefore we reuse this variance in the learnable transition step!

Likewise, we would also like to use the true mean, $\mu_q(\mathbf{x}_t, t)$. However, since this mean depends on \mathbf{x}_0 , we can not use it directly (remember, we will not have access to \mathbf{x}_0 in our learnable denoising step, only to \mathbf{x}_t !). However, perhaps we can somehow approximate \mathbf{x}_0 on the basis of the available information. In particular, we could approximate \mathbf{x}_0 via a neural network, using \mathbf{x}_t and t as the inputs. In other words, we can write:

$$\mu_\theta(\mathbf{x}_t, t) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{\mathbf{x}}_0(\mathbf{x}_t, t)}{1-\bar{\alpha}_t}$$

Therefore, the learnable denoising step has been parameterized to follow a Gaussian distribution with parameters $\mu_\theta(\mathbf{x}_t, t)$ and $\Sigma_q(t)$:

$$p(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N} \left(\mathbf{x}_{t-1}; \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{\mathbf{x}}_0(\mathbf{x}_t, t)}{1-\bar{\alpha}_t}, \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \mathbf{I} \right)$$

Now that we have parameterized the two distributions $p(\mathbf{x}_{t-1} | \mathbf{x}_t)$ and $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)$, we can insert these distributions into the formula for the KL-divergence and calculate the term L_{t-1} . And since both distributions are normal distributions, the KL-divergence turns out to be extremely easy to calculate.

KL-divergence for multivariate Gaussians

Given two multivariate normal distributions such that:

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mu_x, \Sigma_x) \quad q(\mathbf{x}) = \mathcal{N}(\mathbf{y}; \mu_y, \Sigma_y)$$

Then the KL-divergence can be calculated as:

$$D_{KL}(p(\mathbf{x})||q(\mathbf{x})) = \frac{1}{2} \left[\log \frac{|\Sigma_y|}{|\Sigma_x|} - d + \text{tr}(\Sigma_y^{-1} \Sigma_x) + (\mu_y - \mu_x)^T \Sigma_y^{-1} (\mu_y - \mu_x) \right]$$

Let us substitute in our distributions (to make things more readable we'll write $\mu_q = \mu_q(\mathbf{x}_t, t)$ and $\mu_\theta = \mu_\theta(\mathbf{x}_t, t)$):

$$\begin{aligned} D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p(\mathbf{x}_{t-1}|\mathbf{x}_t)) &= D_{KL}(\mathcal{N}(\mathbf{x}_{t-1}; \mu_q, \Sigma_q(t))||\mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta, \Sigma_q(t))) \\ &= \frac{1}{2} \left[\log \frac{|\Sigma_q(t)|}{|\Sigma_q(t)|} - d + \text{tr}(\Sigma_q(t)^{-1} \Sigma_q(t)) + (\mu_\theta - \mu_q)^T \Sigma_q(t)^{-1} (\mu_\theta - \mu_q) \right] \\ &= \frac{1}{2} [\log(1) - d + d + (\mu_\theta - \mu_q)^T \Sigma_q(t)^{-1} (\mu_\theta - \mu_q)] \\ &= \frac{1}{2} [(\mu_\theta - \mu_q)^T \sigma_q^2(t)^{-1} \mathbf{I} (\mu_\theta - \mu_q)] \\ &= \frac{1}{2\sigma_q^2(t)} [(\mu_\theta - \mu_q)^T \mathbf{I} (\mu_\theta - \mu_q)] \\ &= \frac{1}{2\sigma_q^2(t)} \|\mu_\theta - \mu_q\|_2^2 \end{aligned} \tag{16}$$

So, the KL-divergence boils down to being some factor, which depends on t , times the squared distance between the means of the two distributions. This should make somewhat good sense considering that the covariance matrices were identical.

Therefore, the term L_{t-1} can be rewritten as:

$$L_{t-1} = \sum_{t=2}^T \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \left[\frac{1}{2\sigma_q^2(t)} \|\mu_\theta(\mathbf{x}_t, t) - \mu_q(\mathbf{x}_t, t)\|_2^2 \right]$$

4.3 Deriving an expression for L_0

Now let's look at L_0 :

$$L_0 = \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} [\log(p_\theta(\mathbf{x}_0|\mathbf{x}_1))]$$

Remembering how we chose to parameterize the distribution p_θ , we can rewrite the above:

$$L_0 = \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \left[C - \frac{1}{2\sigma_q^2(1)} \|\mu_\theta(\mathbf{x}_1, 1) - \mathbf{x}_0\|_2^2 \right] = C - \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \left[\frac{1}{2\sigma_q^2(1)} \|\mu_\theta(\mathbf{x}_1, 1) - \mathbf{x}_0\|_2^2 \right]$$

Where C is some constant that doesn't have any parameters.

4.4 Combining the expressions

Let's combine what we have so far: we have expressions for L_0 and L_{t-1} and we can therefore rewrite the ELBO:

$$\begin{aligned} \log(p(\mathbf{x}_0)) &\geq L_0 - L_T - L_{t-1} \\ &= C - L_T - \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \left[\frac{1}{2\sigma_q^2(1)} \|\mu_\theta(\mathbf{x}_1, 1) - \mathbf{x}_0\|_2^2 \right] - \sum_{t=2}^T \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \left[\frac{1}{2\sigma_q^2(t)} \|\mu_\theta(\mathbf{x}_t, t) - \mu_q(\mathbf{x}_t, t)\|_2^2 \right] \\ &= C - L_T - \sum_{t=1}^T \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \left[\frac{1}{2\sigma_q^2(t)} \|\mu_\theta(\mathbf{x}_t, t) - \mu_q(\mathbf{x}_t, t)\|_2^2 \right] \end{aligned} \tag{17}$$

We can extend the summation sign by realizing that $\mu_q(\mathbf{x}_1, 1) = \mathbf{x}_0$ (this can be shown algebraically or simply by remembering the purpose of the ground truth denoising step). Since C and L_T contain no learnable parameters, we are only interested in minimizing the last term. Therefore, in the end, we end up with the following optimization problem:

$$\theta^* = \arg \min_{\theta} \sum_{t=1}^T \frac{1}{2\sigma_q^2(t)} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} [\|\mu_{\theta}(\mathbf{x}_t, t) - \mu_q(\mathbf{x}_t, t)\|_2^2]$$

4.5 Minimizing the objective

Now, we want to minimize the objective function to maximize the ELBO. Therefore, let's look at how we can minimize each of the terms inside the summation sign:

$$\begin{aligned} & \arg \min_{\theta} \frac{1}{2\sigma_q^2(t)} \|\mu_{\theta} - \mu_y\|_2^2 \\ &= \frac{1}{2\sigma_q^2(t)} \arg \min_{\theta} \left\| \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)\hat{\mathbf{x}}_0(\mathbf{x}_t, t)}{1 - \bar{\alpha}_t} - \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)\mathbf{x}_0}{1 - \bar{\alpha}_t} \right\|_2^2 \\ &= \frac{1}{2\sigma_q^2(t)} \arg \min_{\theta} \left\| \frac{\sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)(\hat{\mathbf{x}}_0(\mathbf{x}_t, t) - \mathbf{x}_0)}{1 - \bar{\alpha}_t} \right\|_2^2 \\ &= \frac{1}{2\sigma_q^2(t)} \frac{\bar{\alpha}_{t-1}(1 - \alpha_t)^2}{(1 - \bar{\alpha}_t)^2} \arg \min_{\theta} \|\hat{\mathbf{x}}_0(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2 \end{aligned}$$

So, in the end, it turns out that we can optimize the ELBO by minimizing the squared distance between the real image \mathbf{x}_0 and our best guess of the real image $\hat{\mathbf{x}}_0$!

And we can actually take this one step further. We can rewrite the expression for \mathbf{x}_t :

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1 - \alpha_t}\epsilon^* \quad \Rightarrow \quad \mathbf{x}_0 = \frac{\mathbf{x}_t - \sqrt{1 - \alpha_t}\epsilon^*}{\sqrt{\alpha_t}}$$

We can insert this into the expression for μ_q :

$$\begin{aligned} \mu_q &= \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)\mathbf{x}_0}{1 - \bar{\alpha}_t} \\ &= \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t) \frac{\mathbf{x}_t - \sqrt{1 - \alpha_t}\epsilon^*}{\sqrt{\alpha_t}}}{1 - \bar{\alpha}_t} \\ &= \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\mathbf{x}_t + (1 - \alpha_t) \frac{\mathbf{x}_t - \sqrt{1 - \alpha_t}\epsilon^*}{\sqrt{\alpha_t}}}{1 - \bar{\alpha}_t} \\ &= \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\mathbf{x}_t}{(1 - \bar{\alpha}_t)} + \frac{(1 - \alpha_t)\mathbf{x}_t}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} - \frac{(1 - \alpha_t)\sqrt{1 - \alpha_t}\epsilon_0^*}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} \\ &= \left(\frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\mathbf{x}_t}{(1 - \bar{\alpha}_t)} + \frac{(1 - \alpha_t)\mathbf{x}_t}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} \right) \mathbf{x}_t - \frac{(1 - \alpha_t)\sqrt{1 - \alpha_t}\epsilon_0^*}{(1 - \bar{\alpha}_t)\sqrt{\alpha_t}} \\ &= \text{regne, regne...} \\ &= \frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}\sqrt{\alpha_t}}\epsilon^* \end{aligned}$$

Likewise, since we want to parameterize our forward and backward process in the same way, we can rewrite our expression for μ_{θ} in the same way. However, since the expression no longer contains $\bar{\mathbf{x}}_0$ we instead choose to predict the noise ϵ^* . We will call this expression $\hat{\epsilon}_{\theta}(\mathbf{x}_t, t)$

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

We can insert these new expression for μ_q and μ_θ into the KL-divergence:

$$D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p(\mathbf{x}_{t-1}|\mathbf{x}_t)) \quad (18)$$

$$= \frac{1}{2\sigma_q^2(t)} \|\mu_\theta - \mu_y\|_2^2 \quad (19)$$

$$= \frac{1}{2\sigma_q^2(t)} \left\| \left(\frac{1}{\sqrt{\alpha_t}} \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}} \hat{\epsilon}_\theta(\mathbf{x}_t, t) \right) - \left(\frac{1}{\sqrt{\alpha_t}} \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}} \epsilon^* \right) \right\|_2^2 \quad (20)$$

$$= \frac{1}{2\sigma_q^2(t)} \left\| \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}} \epsilon^* - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}} \hat{\epsilon}_\theta(\mathbf{x}_t, t) \right\|_2^2 \quad (21)$$

$$= \frac{1}{2\sigma_q^2(t)} \left\| \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}} (\epsilon^* - \hat{\epsilon}_\theta(\mathbf{x}_t, t)) \right\|_2^2 \quad (22)$$

$$= \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2}{1-\bar{\alpha}_t\alpha_t} \|\epsilon^* - \hat{\epsilon}_\theta(\mathbf{x}_t, t)\|_2^2 \quad (23)$$

$$(24)$$

Therefore, in order to optimize the KL-divergence, we could also choose to minimize the squared distance between the noise added to the picture and our predicted noise. Empirically, this approach has been found to sample pictures of higher quality. Some of these results can be seen in [Denoising Diffusion Probabilistic Models](#). In this paper, they also argue that the loss function can be simplified even further by ignoring the factor in front of the KL-divergence, therefore obtaining the "simple loss function":

$$\mathcal{L}_{simple} = \|\epsilon^* - \hat{\epsilon}_\theta(\mathbf{x}_t, t)\|_2^2$$

The idea behind this loss function is that we do not weigh the different timesteps differently. This should also make good intuitive sense: if the model comes arbitrarily close to predicting the correct noise for all timesteps, then, in theory, the model would be perfect at predicting the distribution of the prior latent layer.

5 The different loss functions

To sum up, we have derived three different loss functions that can be minimized in order to train a diffusion model. We have:

The full negative ELBO:

$$\begin{aligned} \mathcal{L}_{ELBO} = & -\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} [\log(p_\theta(\mathbf{x}_0|\mathbf{x}_1))] + \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} [D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T))] \\ & + \sum_{t=2}^T \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} [D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))] \end{aligned}$$

The loss optimized for finding the ground truth:

$$\mathcal{L}_{\hat{\mathbf{x}}_0} = \|\mathbf{x}_0 - \hat{\mathbf{x}}_0(\mathbf{x}_t, t)\|_2^2$$

And the loss optimized for finding the noise added to the ground truth (which we'll call the simple loss):

$$\mathcal{L}_{simple} = \|\epsilon^* - \hat{\epsilon}_\theta(\mathbf{x}_t, t)\|_2^2$$

Each of these losses have their own ups- and downsides. In the following, we'll focus on \mathcal{L}_{simple} since it turns out, empirically, that this method produces the best results when sampling new images. We'll also try to investigate why this is the case.

5.1 Training and sampling

Here comes probably the most exciting part. How do we set up and train our diffusion model and how do we sample new pictures from it? Here, we can once again draw inspiration from [Denoising Diffusion Probabilistic Models](#) and their algorithms. However, it would also be a good idea to go through some of the expressions that we have derived so far to get an intuitive understanding of what the sampling process actually does.

Remember, that we can generate arbitrarily noisy images from \mathbf{x}_0 using our forward process $q(\mathbf{x}_t|\mathbf{x}_{t-1})$. But when sampling we do not have access to \mathbf{x}_0 ; instead we want to generate some \mathbf{x}_0 . We can do this using our backward process $p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})$. Still, we need to start somewhere in order to begin sampling, that is, we need some \mathbf{x}_t for $t \in [1, T]$. The problem is still that we don't know any \mathbf{x}_t or their distribution. However, earlier we showed that the distribution of $p(\mathbf{x}_T)$ will approximately follow a normal Gaussian simply due to how the forward process was defined. We can use this fact to sample $\mathbf{x}_T \sim p(\mathbf{x}_T) \approx \mathcal{N}(\mathbf{0}, \mathbf{I})$. Now that we have \mathbf{x}_T we can sample \mathbf{x}_{t-1} . From \mathbf{x}_{t-1} we can sample \mathbf{x}_{t-2} and so on all the way down to \mathbf{x}_0 using $p(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mu_\theta, \sigma_q^2(t))$ (here, we can both use $\mu_\theta(\hat{\epsilon}_\theta(\mathbf{x}_t, t))$ and $\mu_\theta(\bar{\mathbf{x}}_0(\mathbf{x}_t, t))$). However, when we get to $p(\mathbf{x}_0|\mathbf{x}_1)$ we choose to let $\mathbf{x}_0 = \mu_\theta(\mathbf{x}_1)$ - we don't sample from the distribution but take the mean. This is done to avoid unnecessary noise added by the random process in the sampling procedure. In the end, we can sum up the sampling process as an algorithm:

Sampling algorithm

```

1:  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t$  from  $T$  downto 1 do
3:    $\mathbf{x}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}} \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}\sqrt{\alpha_t}} \hat{\epsilon}_\theta(\mathbf{x}_t, t)$ 
4:   if  $t > 0$  then
5:      $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
6:      $\mathbf{x}_{t-1} \leftarrow \mathbf{x}_{t-1} + \mathbf{z} \sigma_q(t)$ 
7:   end if
8: end for
9: return  $\mathbf{x}_0$ 

```

Likewise, we can write an algorithm for training the model. Assuming that we are using the simple loss function, the algorithm for training the model can be written as:

Training algorithm

```

1: repeat
2:    $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ 
3:    $t \sim \text{Uniform}(1, \dots, T)$ 
4:    $\epsilon^* \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
5:    $\mathbf{x}_t \leftarrow \sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1-\alpha_t} \epsilon^*$ 
6:   Take gradient descent step on  $\nabla_\theta ||\epsilon^* - \hat{\epsilon}_\theta(\mathbf{x}_t, t)||_2^2$ 
7: until convergence

```

References