The Mathematical Foundations of Denoising Diffusion Probabilistic Models

Dimas Tri Kurniawan

dimas.tri01@ui.ac.id

Abstract

Denoising Diffusion Probabilistic Models (DDPM) is one of the latest approaches in image generation that has demonstrated impressive results in producing high-quality images. This model is based on the principle of reverse diffusion, which transforms the original data distribution into a noise distribution and then gradually recovers it back to the original data through a step-by-step denoising process. In this paper, the author aims to delve deeper into the mathematical foundations of DDPM. The author believes that there are still shortcomings in the explanation of the mathematical foundations presented in the DDPM paper. Therefore, the author is interested in reviewing the various formulas used as the foundation of the paper.

1 Introduction

Denoising Diffusion Probabilistic Models (hereafter referred to as Diffusion Models, for brevity) have emerged as one of the latest and most promising methods in probabilistic image and data generation. Since their initial introduction, DDPM has shown significant advancements in the quality of generated images, surpassing other generative methods such as Generative Adversarial Networks (GANs) in terms of stability and the ability to produce highly detailed images. This method operates through a diffusion process, gradually transforming the original data into noise and then attempting to recover the original data from the noise using a probabilistic

Preprint. Under review.

model. This approach offers a new and intriguing way to understand how models can address challenges in data generation, particularly in the context of image processing.

Although the empirical results of DDPM are very promising, most existing literature tends to focus on algorithmic implementation and optimization techniques without providing an in-depth explanation of the mathematical foundations of this model. However, a deeper understanding of the mathematical aspects of DDPM is crucial for enhancing our comprehension of how this model functions, as well as for unlocking its potential applications in various domains, such as data distribution modeling and image reconstruction.

This paper aims to fill this gap by providing a detailed explanation of the mathematical foundations underlying Denoising Diffusion Probabilistic Models. We will discuss how the reverse diffusion process works in a probabilistic context and outline the relevant probability theory, including the role of noise distribution and optimization-based learning in improving model performance. Additionally, we will elaborate on the key challenges in implementing DDPM and explore potential future directions for the development of this model.

By providing deeper insights into the mathematical foundations of DDPM, we hope that readers can gain a more comprehensive understanding of how this model works and how its technological potential can be leveraged in the future.

2 Background

Diffusion models consist of two main processes: the Forward/Diffusion Process and the Reverse Diffusion Process, inspired by nonequilibrium thermodynamics. Our goal is to define a forward (or inference) diffusion process which converts any complex data distribution into a simple, tractable, distribution, and then learn a finite-time reversal of this diffusion process which defines our generative model distribution. We first describe the forward, inference diffusion process. We then show how the reverse diffusion process can be trained and used to evaluate probabilities.

2.1 Forward Process

The forward diffusion process is inspired by the forward trajectory in deep unsupervised learning using Nonequilibrium Thermodynamics. In the forward trajectory, we label the data distribution $q(\mathbf{x}^{(0)})$. The data distribution is gradually converted

into a well behaved (analytically tractable) distribution $\pi(\mathbf{y})$ by repeated application of a Markov diffusion kernel $T_\pi(\mathbf{y}|\mathbf{y}';\beta)$ for $\pi(\mathbf{y})$, where β is the diffusion rate,

$$\pi(\mathbf{y}) = \int T_\pi(\mathbf{y}|\mathbf{y}';\beta)\pi(\mathbf{y}')\,d\mathbf{y}' \tag{1}$$

$$q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}) = T_\pi(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)};\beta_t). \tag{2}$$

The forward trajectory, corresponding to starting at the data distribution and performing T steps of diffusion, is thus

$$q(\mathbf{x}^{(0\ldots T)}) = q(\mathbf{x}^{(0)})\prod_{t=1}^{T} q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}) \tag{3}$$

The forward diffusion follows the forward trajectory, only with few different notations. The forward process/diffusion process is the process in which noise is gradually added to an image until it converges to a Gaussian distribution with a mean of 0 and a variance of 1. In other words, the image will become pure noise after t steps. The forward process is defined by a Markov chain that periodically adds Gaussian noise:

$$q(\mathbf{x}_{0:T}) = q(\mathbf{x}_0)\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{4}$$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \frac{q(\mathbf{x}_0)\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})}{q(\mathbf{x}_0)} \tag{5}$$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) \coloneqq \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{6}$$

Equation (1) means that we want to find the distribution of the image that has been progressively noised, from $\mathbf{x}_1$ to $\mathbf{x}_T$, given the original image $\mathbf{x}_0$. Due to the Markov assumption, the previous images/states are not needed, so the equation should be:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) \coloneqq \frac{q(\mathbf{x}_T|\mathbf{x}_{T-1},\mathbf{x}_{T-2},\mathbf{x}_{T-3},\ldots,\mathbf{x}_0)\; q(\mathbf{x}_{T-1}|\mathbf{x}_{T-2},\mathbf{x}_{T-3},\mathbf{x}_{T-4},\ldots,\mathbf{x}_0)\;\ldots\; q(\mathbf{x}_0)}{q(\mathbf{x}_0)}, \tag{7}$$

, which can be simplified by removing the previous states:

$$\coloneqq \frac{q(\mathbf{x}_T|\mathbf{x}_{T-1}, \cancel{\mathbf{x}_{T-2}}, \cancel{\mathbf{x}_{T-3}}, \cancel{...}, \cancel{\mathbf{x}_0})\ q(\mathbf{x}_{T-1}|\mathbf{x}_{T-2}, \cancel{\mathbf{x}_{T-3}}, \cancel{\mathbf{x}_{T-4}}, \cancel{...}, \cancel{\mathbf{x}_0})\ ...\ \cancel{q(\mathbf{x}_0)}}{\cancel{q(\mathbf{x}_0)}} \tag{8}$$

becomes:

$$\coloneqq \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{9}$$

The forward process gradually adds Gaussian noise based on a linear variance schedule $\beta_1, \beta_2, \beta_3, ..., \beta_t$:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) \coloneqq \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \tag{10}$$

with $\mathbf{x}_t$ as the output, $\sqrt{1-\beta_t}$ as the mean, and β_t as the variance of $\mathbf{x}_t$. Using reparameterization trick $\mathcal{N}(\mu, \sigma^2) = \mu + \sigma\epsilon$, the equation above can be rewritten as:

$$\coloneqq \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\epsilon_{t-1}. \tag{11}$$

Let $\alpha_t \coloneqq 1 - \beta_t$, the equation above can be rewritten as:

$$\coloneqq \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1}. \tag{12}$$

Substitute $\mathbf{x}_{t-1}$:

$$\coloneqq \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\epsilon_{t-2}) + \sqrt{1-\alpha_t}\epsilon_{t-1} \tag{13}$$

$$\coloneqq \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\alpha_t - \alpha_t\alpha_{t-1}}\epsilon_{t-2} + \sqrt{1-\alpha_t}\epsilon_{t-1} \tag{14}$$

If 2 gaussian noises $\mathcal{N}(\mu, \sigma_1^2)$ and $\mathcal{N}(\mu, \sigma_2^2)$ are summed, then the result is $\mathcal{N}(\mu, (\sigma_1^2 + \sigma_2^2))$. Rewrite the last 2 terms:

$$\coloneqq \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\alpha_t - \alpha_t\alpha_{t-1} + 1 - \alpha_t}\epsilon \tag{15}$$

$$\coloneqq \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1 - \alpha_t\alpha_{t-1}}\epsilon \tag{16}$$

Substitute $\mathbf{x}_{t-2}$:

$$\coloneqq \sqrt{\alpha_t \alpha_{t-1} \alpha_{t-2}} \mathbf{x}_{t-3} + \sqrt{1 - \alpha_t \alpha_{t-1} \alpha_{t-2}} \epsilon. \quad (17)$$

Substitute $\mathbf{x}_{t-3}$:

$$\coloneqq \sqrt{\alpha_t \alpha_{t-1} \alpha_{t-2} \alpha_{t-3}} \mathbf{x}_{t-4} + \sqrt{1 - \alpha_t \alpha_{t-1} \alpha_{t-2} \alpha_{t-3}} \epsilon \quad (18)$$

If we continue substituting down to $\mathbf{x}_1$:

$$\coloneqq \sqrt{\alpha_t \alpha_{t-1} \alpha_{t-2} \alpha_{t-3} \ldots \alpha_1} \mathbf{x}_0 + \sqrt{1 - \alpha_t \alpha_{t-1} \alpha_{t-2} \alpha_{t-3} \ldots \alpha_1} \epsilon \quad (19)$$

Let $\bar{\alpha}_t \coloneqq \prod_{t=1}^{T} \alpha_t$, then:

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) \coloneqq \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon \quad (20)$$

Simplifying $q(\mathbf{x}_t | \mathbf{x}_{t-1})$ into Equation (???) makes the formula easier to solve. Instead of iterating from $\mathbf{x}_T$ to $\mathbf{x}_1$, we only need to determine $\bar{\alpha}_t$, and then substitute it into Equation (???).

2.2 Reverse Process

The reverse diffusion process is inspired by the reverse trajectory in deep unsupervised learning using Nonequilibrium Thermodynamics. In the reverse trajectory, the generative distribution will be trained to describe the same trajectory, but in reverse,

$$p(\mathbf{x}^{(T)}) = \pi(\mathbf{x}^{(T)}) \quad (21)$$

$$p(\mathbf{x}^{(0 \ldots T)}) = p(\mathbf{x}^{(T)}) \prod_{t=1}^{T} p(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}) \quad (22)$$

Diffusion models are a latent variable model in the form of $p_\theta(\mathbf{x}_0) \coloneqq \int p_\theta(\mathbf{x}_{0:T}) \, d_{\mathbf{x}_{1:T}}$, where $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_T$ are latents with the same dimensionality as the data $\mathbf{x}_0 \sim q(\mathbf{x}_0)$. The reverse diffusion is very similar to the reverse trajectory. Joint distribution $p_\theta(\mathbf{x}_{0:T})$ is the reverse process:

$$p_\theta(\mathbf{x}_{0:T}) \coloneqq p_\theta(\mathbf{x}_0|\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_T)\ p_\theta(\mathbf{x}_1|\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, ..., \mathbf{x}_T)\ ...\ p(\mathbf{x}_T) \tag{23}$$

Just like the forward process, the reverse process is defined as a Markov chain:

$$\coloneqq p_\theta(\mathbf{x}_0|\mathbf{x}_1, \cancel{\mathbf{x}_2}, \cancel{\mathbf{x}_3}, \cancel{...}, \cancel{\mathbf{x}_T})\ p_\theta(\mathbf{x}_1|\mathbf{x}_2, \cancel{\mathbf{x}_3}, \cancel{\mathbf{x}_4}, \cancel{...}, \cancel{\mathbf{x}_T})\ ...\ p(\mathbf{x}_T), \tag{24}$$

so that we obtain the final equation:

$$p_\theta(\mathbf{x}_{0:T}) \coloneqq p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t). \tag{25}$$

So far, we do not know the true distribution of the reverse process. Therefore, we use an approximation p_θ , which will be learned by the neural network. Specifically for $\mathbf{x}_T$, we do not use the approximation p_θ because we know the true value of $\mathbf{x}_T$.

The reverse process is a Gaussian transition with μ and σ^2 that will be learned later, starting from $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; 0, \mathbf{I})$, which represents pure noise with a mean of 0 and a variance of 1 (identity matrix):

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \coloneqq \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)) \tag{26}$$

However, the authors of the DDPM paper decided to assign a fixed value to σ^2 based on a linear schedule. In other words, σ does not need to be learned in the reverse process.

3 Loss Function

We do not know the distribution of the reverse process. Therefore, we will train the model to approximate the distribution of p as accurately as possible. It turns out that we can find a lower bound using the same approach as in Variational Autoencoder (VAE). To do this, we will study what Latent Variables are, Deep Latent Variable Models, VAE, and their connection to DDPM.

3.1 Latent Variable

A latent variable is a variable that is part of the model but cannot be observed due to its hidden nature. Therefore, a latent variable is not part of the dataset. Typically,

we define **z** as the latent variable and **x** as the observed variable. The marginal distribution of the observed variables $p_\theta(x)$ is given by:

$$p_\theta(x) = \int p_\theta(x, z) dz \tag{27}$$

3.2 Deep Latent Variable Models

Deep Latent Variable Models (DLVM) are latent variable models $p_\theta(x, y)$ whose distribution is parameterized by a neural network. One of the most commonly encountered (and perhaps the simplest) DLVMs is:

$$p_\theta(x, z) = p_\theta(z)\, p_\theta(x|z) \tag{28}$$

3.3 Intractabilities

The main challenge in finding $p_\theta(z|x)$ is that both $p_\theta(x, z)$ and $p_\theta(x)$ are intractable in DLVM, making $p_\theta(z|x)$ intractable as well. Therefore, an approximate inference technique is needed to approximate the posterior $p_\theta(z|x)$. To transform the intractable posterior inference in DLVM into a tractable one, we introduce a parametric inference model $q_\phi(z|x)$. This model is also known as an encoder or recognition model. We use ϕ as the parameter of this inference model, which is also referred to as the variational parameters.

Next, we will optimize the variational parameters ϕ so that:

$$q_\phi(z|x) \approx p_\theta(z|x) \tag{29}$$

Moving forward, this posterior approximation will help us optimize the marginal likelihood.

3.4 VAE

VAE is an example of a DLVM. VAE uses deep neural networks to parameterize the probability distributions that define the latent variable model.

Consider the following DLVM:

$$p_\theta(x) = \int p_\theta(x, z)\, dz \tag{30}$$

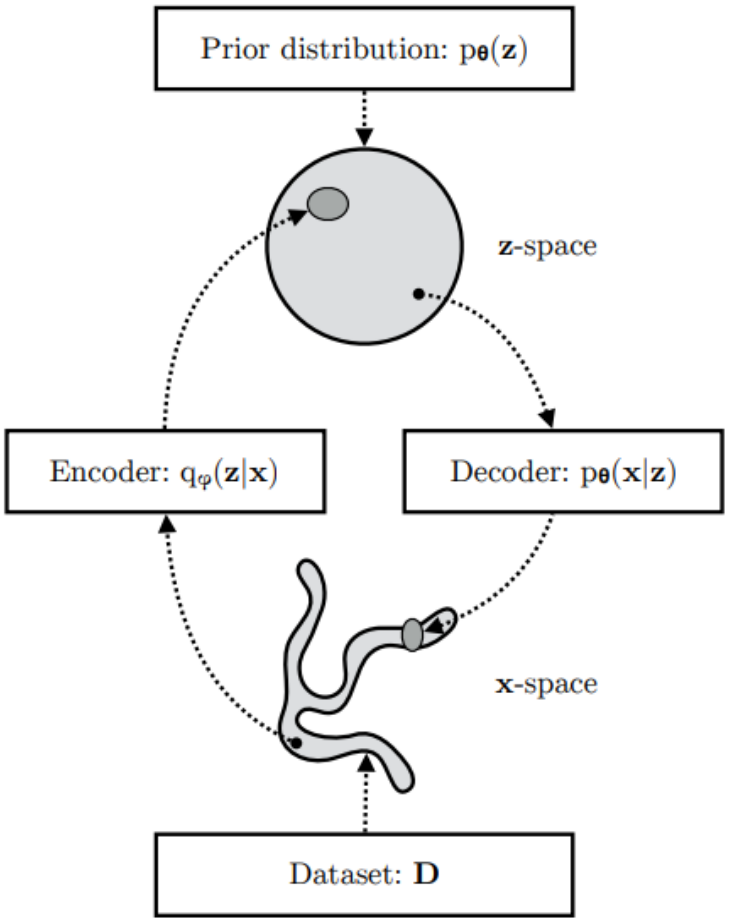

Figure 1: VAE learns a stochastic mapping between the observed x-space , whose empirical distribution $q_D(x)$ is usually complex, and a latent z-space , whose distribution is relatively simple (e.g., a sphere, as shown in the figure). The generative model learns the joint distribution $p_\theta(x, z)$ which is often (but not always) factorized as: $p_\theta(x, z) = p_\theta(z)\, p_\theta(x|z)$, with a prior distribution over latent space $p_\theta(z)$, and a stochastic decoder $p_\theta(x|z)$. The stochastic encoder $q_\phi(z|x)$, also known as the inference model, approximates the true intractable posterior $p_\theta(z|x)$ of the generative model.

For any inference model $q_\phi(z|x)$ and any variational parameters ϕ:

$$p_\theta(x) = \int p_\theta(x, z) \frac{q_\phi(z|x)}{q_\phi(z|x)}\, dz \tag{31}$$

Since $q_\phi(z|x) \approx p_\theta(z|x)$:

$$p_\theta(x) = \int p_\theta(x, z) \frac{q_\phi(z|x)}{p_\theta(z|x)}\, dz \tag{32}$$

$$p_\theta(x) = \mathbb{E}_{q_\phi(z|x)}\left[\frac{p_\theta(x,z)}{p_\theta(z|x)}\right] \tag{33}$$

$$\log p_\theta(x) = \log \mathbb{E}_{q_\phi(z|x)}\left[\frac{p_\theta(x,z)}{p_\theta(z|x)}\right] \tag{34}$$

The logarithm is a concave function. This can be proven because the second derivative of the logarithm is negative. Therefore, the following holds: $\log \mathbb{E}_{q_\phi(z|x)}\left[\frac{p_\theta(x,z)}{p_\theta(z|x)}\right] = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{p_\theta(z|x)}\right]$, so that:

$$\log p_\theta(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{p_\theta(z|x)}\right] \tag{35}$$

$$\log p_\theta(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log \left[\frac{p_\theta(x,z)}{q_\phi(z|x)}\frac{q_\phi(z|x)}{p_\theta(z|x)}\right]\right] \tag{36}$$

$$\log p_\theta(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] + \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{q_\phi(z|x)}{p_\theta(z|x)}\right] \tag{37}$$

The second term in Equation (37) is the Kullback-Leibler (KL) divergence between $q_\phi(z|x)$ and $p_\theta(z|x)$, which is non-negative:

$$D_{KL}(q_\phi(z|x) \parallel p_\theta(z|x)) \geq 0 \tag{38}$$

and equals zero if and only if $q_\phi(z|x)$ matches the true posterior distribution.

The first term in Equation (37) is the variational lower bound, also called the evidence lower bound (ELBO):

$$\mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] \tag{39}$$

Since the KL divergence is non-negative, the ELBO serves as a lower bound on the log-likelihood of the data

$$\mathcal{L}_{\theta,\phi}(x) = \log p_\theta(x) - D_{KL}(q_\phi(z|x) \parallel p_\theta(z|x)) \tag{40}$$

$$\mathcal{L}_{\theta,\phi}(x) \leq \log p_\theta(x), \tag{41}$$

Thus, Equation (37) can be simplified as follows:

$$\log p_\theta(x) = \mathcal{L}_{\theta,\phi}(x) + D_{KL}(q_\phi(z|x) \parallel p_\theta(z|x)) \tag{42}$$

$$\log p_\theta(x) \leq \log p_\theta(x) + D_{KL}(q_\phi(z|x) \parallel p_\theta(z|x)) \tag{43}$$

3.5 DDPM

We can optimize the log-likelihood using a method similar to VAE. Consider the following DLVM:

$$\log p_\theta(x) = \log \int p_\theta(x, z)\, dz \tag{44}$$

$$\log p_\theta(x) = \log \int p_\theta(x, z) \frac{q(z|x)}{q(z|x)}\, dz \tag{45}$$

$$\log p_\theta(x) = \log \mathbb{E}_{q(z|x)} \left[\frac{p_\theta(x, z)}{q(z|x)} \right] \tag{46}$$

The logarithm is a concave function. This can be proven because its second derivative is negative. Therefore, the following holds: $\log \mathbb{E}_{q(z|x)} \left[\frac{p_\theta(x,z)}{q(z|x)} \right] = \mathbb{E}_{q(z|x)} \left[\log \frac{p_\theta(x,z)}{q(z|x)} \right]$, , so as:

$$\log p_\theta(x) \geq \mathbb{E}_{q(z|x)} \left[\log \frac{p_\theta(x, z)}{q(z|x)} \right] \tag{47}$$

We can apply the same approach to diffusion models. The difference is that the latent variables we use are $\mathbf{x}_{1:T}$ and the observed variable is $\mathbf{x}_0$. Additionally, instead of directly optimizing a loss function, we seek the expectation of the loss function.

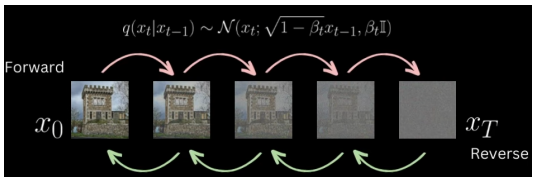

Figure 2: reverse process

Additionally, instead of maximizing the log-likelihood, we minimize the negative log-likelihood. Essentially, both approaches are equivalent. However, since optimizers typically minimize functions, it is more suitable to use the negative log-likelihood. As the name suggests, a loss function is something we aim to minimize as much as possible.

$$\mathbb{E}_{q(\mathbf{x}_0)}[-\log p_\theta(x_0)] = \mathbb{E}_{q(\mathbf{x}_0)}\left[-\log \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right]\right] \tag{48}$$

$$\mathbb{E}_{q(\mathbf{x}_0)}[-\log p_\theta(x_0)] \leq \mathbb{E}_{q(\mathbf{x}_0)}\left[\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right]\right] \tag{49}$$

$$\mathbb{E}_{q(\mathbf{x}_0)}[-\log p_\theta(x_0)] \leq \mathbb{E}_{q(\mathbf{x}_0)}\left[\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right]\right] \tag{50}$$

$$\mathbb{E}_{q(\mathbf{x}_0)}[-\log p_\theta(x_0)] \leq \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] \tag{51}$$

Next, we can optimize the term inside the expectation:

$$\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} = \log \frac{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \tag{52}$$

$q(\mathbf{x}_t|\mathbf{x}_{t-1})$ can be rewritten as $\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t)\ q(\mathbf{x}_t)}{q(\mathbf{x}_{t-1})}$. However, $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ will have high variance because there can be multiple candidates for $\mathbf{x}_t$.

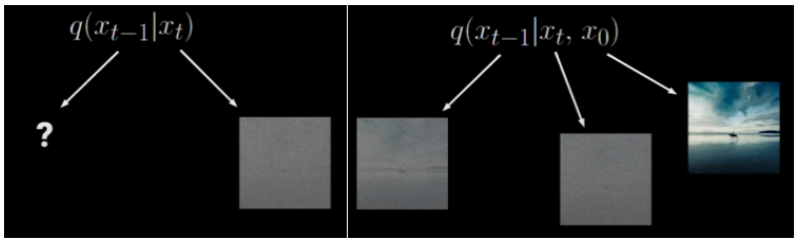

Figure 3: high variance vs. low variance

Therefore, we will introduce the original image $\mathbf{x}_0$ so that the candidates for $\mathbf{x}_{t-1}$ can better align with the original image.

$$\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} = \log \frac{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)}{p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \tag{53}$$

However, there is something odd. If we examine it more closely, the first term in the numerator contains a self-loop! More precisely, $q(\mathbf{x}_1|\mathbf{x}_0, \mathbf{x}_0) = \frac{q(\mathbf{x}_0|\mathbf{x}_1, \mathbf{x}_0)\ q(\mathbf{x}_1|\mathbf{x}_0)}{q(\mathbf{x}_0|\mathbf{x}_0)}$. To address this issue, we need to factor out the first term from the product series. Only then can we add $\mathbf{x}_0$ to the conditional probability in the product series.

$$\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} = \log \frac{\cancel{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)}}{\cancel{p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}} \tag{54}$$

$$\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} = \log \frac{q(\mathbf{x}_1|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)}{p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \tag{55}$$

$$\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} = \log \frac{q(\mathbf{x}_1|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)\ q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)\ p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \tag{56}$$

$$\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} =$$
$$\log \frac{\cancel{q(\mathbf{x}_1|\mathbf{x}_0)}\ q(\mathbf{x}_{T-1}|\mathbf{x}_T, \mathbf{x}_0)\ q(\mathbf{x}_T|\mathbf{x}_0)\ q(\mathbf{x}_{T-2}|\mathbf{x}_{T-1}, \mathbf{x}_0)\ \cancel{q(\mathbf{x}_{T-1}|\mathbf{x}_0)} \ldots q(\mathbf{x}_1|\mathbf{x}_2, \mathbf{x}_0)\ \cancel{q(\mathbf{x}_2|\mathbf{x}_0)}}{\cancel{q(\mathbf{x}_{T-1}|\mathbf{x}_0)}\ q(\mathbf{x}_{T-2}|\mathbf{x}_0) \ldots \cancel{q(\mathbf{x}_2|\mathbf{x}_0)}\ \cancel{q(\mathbf{x}_1|\mathbf{x}_0)}\ p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \tag{57}$$

$$\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} = \log \frac{q(\mathbf{x}_T|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \tag{58}$$

$$\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} = \log \frac{q(\mathbf{x}_T|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p(\mathbf{x}_T)\, p_\theta(\mathbf{x}_0|\mathbf{x}_1) \prod_{t=2}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \tag{59}$$

$$\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} = \log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p(\mathbf{x}_T)} + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \tag{60}$$

Reinsert $\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}$ into the loss function:

$$\mathbb{E}_{q(\mathbf{x}_0)}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p(\mathbf{x}_T)} + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\right]. \tag{61}$$

Convert each term in the loss function into the form of KL (Kullback–Leibler) divergence:

$$\mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p(\mathbf{x}_T)} + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\right] \tag{62}$$

$$= \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T)) + \sum_{t=2}^{T} D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) - \log p_\theta(\mathbf{x}_0\right. \tag{63}$$

We get:

$$L = \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T)) + \sum_{t=2}^{T} D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) - \log p_\theta(\right. \tag{64}$$

with

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}), \tag{65}$$

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t, \tag{66}$$

$$\tilde{\beta}_t \coloneqq \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}}\beta_t. \tag{67}$$

We can decompose L into:

$$L_T = D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T)). \tag{68}$$

$$L_{t-1} = D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)). \tag{69}$$

$$L_0 = \log p_\theta(\mathbf{x}_0|\mathbf{x}_1). \tag{70}$$

3.6 Forward process and L_T

In fact, β_t can be learned through reparameterization. However, the value of β_t is kept constant, meaning that q has no learnable parameters. This makes L_T constant during training and therefore can be ignored.

3.7 Reverse process and $L_{1:T-1}$

We aim to learn the mean and variance of the reverse diffusion process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t))$, where $1 < t < T$. Here, the authors decided not to learn the variance and instead set it as a constant

$$\Sigma_\theta(\mathbf{x}_t, t) = \sigma_t^2 \mathbf{I} \tag{71}$$

, so that

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I}) \tag{72}$$

To represent the mean $\mu_\theta(\mathbf{x}_t, t)$, the DDPM authors parameterized it based on the analysis of L_t, namely $D_{KL}(q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}))$, using the mean square error:

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \|\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) - \mu_\theta(\mathbf{x}_t, t)\|^2 \right] + C, \tag{73}$$

with C being a constant that does not depend on θ. Substitute $\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0)$ into (12):

$$L_{t-1} = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t - \mu_\theta(\mathbf{x}_t, t)\|^2\right] + C, \tag{74}$$

In equation (2), we have seen how to find $\mathbf{x}_t$ given $\mathbf{x}_0$. We can use that equation to find $\mathbf{x}_0$:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{75}$$

$$\iff \mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon) \tag{76}$$

Substitute $\mathbf{x}_0$ into (14):

$$L_{t-1} = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon) + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t - \mu_\theta(\mathbf{x}_t, t)\|^2\right] + C \tag{77}$$

$$= \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon) - \mu_\theta(\mathbf{x}_t, t)\|^2\right] + C \tag{78}$$

The authors of DDPM reparameterize x_t to predict the noise in the reverse diffusion process:

$$= \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t(\mathbf{x}_0, \epsilon) - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon) - \mu_\theta(\mathbf{x}_t(\mathbf{x}_0, \epsilon), t)\|^2\right] + C, \tag{79}$$

with $\mathbf{x}_t(\mathbf{x}_0, \epsilon) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I})$. Equation (10) shows that μ_θ must predict $\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t(\mathbf{x}_0, \epsilon) - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon)$, given **t**. Since **t** is already included as an input, we can parameterize it as follows:

$$\mu_\theta(\mathbf{x}_t, t) = \bar{\mu}_t(\mathbf{x}_t, \mathbf{x}_0((x)_t, \epsilon_\theta)) \tag{80}$$

$$= \bar{\mu}_t(\mathbf{x}_t, \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(\mathbf{x}_t))) \tag{81}$$

$$= \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)) \tag{82}$$

Substitute Equation (16) into Equation (15):

$$L_{t-1} = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon) - \mu_\theta(\mathbf{x}_t, t)\|^2\right] + C \tag{83}$$

$$= \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon) - \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t))\|^2\right] + C \tag{84}$$

$$= \mathbb{E}_q\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2\right] + C \tag{85}$$

After conducting experiments, the DDPM authors found that removing the first term in (17) during training improves sample quality and makes implementation easier.

$$L_{simple}(\theta) \coloneqq \mathbb{E}_q\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2\right] \tag{86}$$

Recall that $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t))$, we can sample $\mathbf{x}_{t-1}$ using the reparameterization trick $\mathcal{N}(\mu, \sigma^2) = \mu + \sigma\epsilon$:

$$\mathbf{x}_{t-1} = \mu_\theta(\mathbf{x}_t, t) + \Sigma_\theta(\mathbf{x}_t, t) + \epsilon \tag{87}$$

$$= \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)) + \sqrt{\beta_t}\epsilon \tag{88}$$

4 Data Scaling