The Maths Behind Denoising Diffusion Probabilistic Models

Giovani Tavares
giovanitavares@outlook.com

University of Sao Paulo — November 15, 2025

1 Motivation

If you read my last post on Denoising Diffusion Probabilistic Models (DDPMs) where we derived the training and sampling algorithms for this type of image generation models, you might have noticed that the original DDPM [1] does not let one generate an image $\mathbf{x}$ from a specific class $\mathbf{y}$.

Hence, even though that we might have trained a model useful for generating images of clothes from the Fashion MNIST Dataset [3], the generated samples classes output in the sampling algorithm were random *by design*.

In this post we are interested in deriving the training and sampling algorithm for an implementation of **Classifier-Free Diffusion Guidance** [2] for DDPMs. This type of guidance lets us use diffusion models to generate images from specific classes, i.e., with them we can parametrize the sampling algorithm to generate images from specific classes.

2 Recap: What are DDPMs?

Denoising Diffusion Probabilistic Models (DDPMs) are models capable of predicting *noise* from a noisy input. By using such prediction, a sampling algorithm can be used to remove the noise from the input which results in a denoised output.

DDPMs are made of two processes: a **forward** and a **reversion** process. The former is responsible for gradually adding noise to a image by sampling from a normal distribution according to a Markov Chain. The latter removes added noise by sampling from another normal distribution. In simple terms, the training of DDPMs involve learning the reversion process' distribution's parameters.

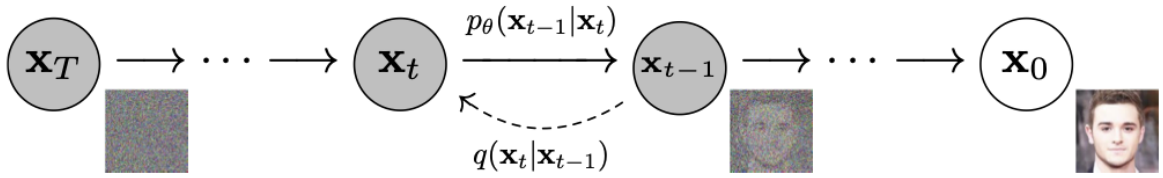

Figure 1: Image extracted from the original DDPM paperl: "*Denoising Diffusion Probabilistic Models*" [**?**]

3 What is Classifier-Free Diffusion Guidance

Classifier-Free Diffusion Guidance [2] was introduced in 2022 by Google Research.

3.1 Forward Process q

The process of adding noise to an input image ($\mathbf{x_0}$) is a Markov Chain that generates a noisier image x_t from a less noisy image $\mathbf{x_{t-1}}$. Hence, $\mathbf{x_t}$ represents the result of adding noise to $\mathbf{x_{t-1}}$ by transitioning it **once** in the Markov Chain.

From the original DDPM paper, we know that in the forward process, a noisy version $\mathbf{x_t}$ of an image $\mathbf{x_0}$ is produced by sampling from the following distribution with $\mathbf{t} > \mathbf{0}$.

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

$$\bar{\alpha_t} := \prod_{s=1}^{t} a_s \tag{2}$$

$$q(\mathbf{x_t}|\mathbf{x_0}) := \mathcal{N}(\mathbf{x_t}; \sqrt{\bar{\alpha_t}} \times \mathbf{x_0}; (\mathbf{1} - \bar{\alpha_t})\mathbf{I}) \tag{3}$$

As According to the original DDPM paper, the **Variance Schedule** $\beta_1, ..., \beta_T$ sequence that defines the noisy images distribution are held constant.

3.2 Reverse Process p_θ

The Reverse Process is a Markov Chain with a Standard Gaussian initial state and Gaussian transition distribution p_θ parametrized by mean μ_θ and variance Σ_θ. **The core idea of the chain is that the transitions remove each a little bit of the noise from the initial state up until the noise-free state $\mathbf{x_0}$.**

Definition 3.1 (Reverse Process Transitions)**.** *The Reverse Process is a Markov Chain with the following transitions:*

$$p(x_T) = \mathcal{N}(x_T; 0; 1) \tag{4}$$

$$p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \tag{5}$$

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t); \Sigma_\theta(x_t, t)) \tag{6}$$

What we ultimately want is to have a $\mathbf{x}_0$ that is as likely as possible. We can use the standard rule of probability to obtain a marginalization of $p(\mathbf{x}_0)$ using the latent variables $\mathbf{x_{1:T}}$

Definition 3.2 (Reverse Process Prior)**.** *Ideally, the Reverse Process would let us sample from:*

$$p(\mathbf{x_0}) = \int p(\mathbf{x_0}, \mathbf{x_{1:T}}) \mathbf{d_{1:T}} \tag{7}$$

$$\tag{8}$$

From the defition above, we see that $p(\mathbf{x}_0)$ is very complex due to its **multidimensionality**, which makes it intractable. That is why in DDPMs, $p(\mathbf{x}_0)$ is never computed directly, but instead its lower bound.

3.2.1 Evidence Lower Bound / ELBO

The Evidence Lower Bound is a tight lower bound that limits $log(p(\mathbf{x_0}))$ from below. Hence, when $p(\mathbf{x_0})$ is intractable as in the case of DDPMs, one can always maximize such lower bound as a means to ensure that $log(p(\mathbf{x_0}))$ is as large as possible. Such lower bound is often called **ELBO** and will be demonstrated using two different approaches: the **Jensen's Inequality** and the **KL Divergence**.

Definition 3.3 (Evidence Lower Bound - Jensen's Inequality)**.** *Let's use the Rule Of Total Probability to find*

a lower bound for the log-likelihood function.

$$log[p_\theta(\mathbf{x_0})] = log \int_{\mathbf{x_{1:T}}} p(\mathbf{x_0}, \mathbf{x_{1:T}}) d\mathbf{x_{1:T}} \tag{9}$$

$$log[p_\theta(\mathbf{x_0})] = log \int_{\mathbf{x_{1:T}}} p(\mathbf{x_0}, \mathbf{x_{1:T}}) \frac{q(\mathbf{x_{1:T}}|\mathbf{x_0})}{q(\mathbf{x_{1:T}}|\mathbf{x_0})} d\mathbf{x_{1:T}} \tag{10}$$

$$log[p_\theta(\mathbf{x_0})] = log(\mathbb{E}_q \Big[\frac{p(\mathbf{x_{0:T}})}{q(\mathbf{x_{1:T}}|\mathbf{x_0})} \Big]) \tag{11}$$

the Jensen's inequality tells us (12)

$$f(\mathbb{E}(\mathbf{X})) \geq \mathbb{E}(f(\mathbf{X}))) \tag{13}$$

for any concave function f. (14)

log is concave, hence: (15)

$$log[p_\theta(\mathbf{x_0})] \geq \mathbb{E}_q \Big[log \frac{p(\mathbf{x_{0:T}})}{q(\mathbf{x_{1:T}}|\mathbf{x_0})} \Big] \tag{16}$$

If we define the Evidence Lower Bound L as: (17)

$$L := \mathbb{E}_q \Big[-log \frac{p(\mathbf{x_{0:T}})}{q(\mathbf{x_{1:T}}|\mathbf{x_0})} \Big] \tag{18}$$

$$\implies -log[p_\theta(\mathbf{x_0})] \leq L \tag{19}$$

Definition 3.4 (Evidence Lower Bound - KL Divergence)**.** *In order to reverse the forward process, we need that the forward process' distribution $q(\mathbf{x_{1:T}}|\mathbf{x_0})$ is as close to $p(\mathbf{x_{1:T}}|\mathbf{x_0})$ as possible. We can use the Kullback-Leibler (KL) divergence between q and p ($\mathbf{D_{KL}}$) to evaluate their difference as find a bound to $log[p_\theta(\mathbf{x}_0)]$.*

$$\mathbf{D_{KL}}\big[q(\mathbf{x_{1:T}}|\mathbf{x_0})||p(\mathbf{x_{1:T}}|\mathbf{x_0})\big] := \mathbb{E}_q\big[log(q(\mathbf{x_{1:T}}|\mathbf{x_0}) - log(p(\mathbf{x_{1:T}}|\mathbf{x_0}))\big] \tag{20}$$

using the Bayes' Rule we can write (21)

$$p(\mathbf{x_{1:T}}|\mathbf{x_0}) = \frac{p(\mathbf{x_{1:T}}, \mathbf{x_0})}{p(\mathbf{x_0})} \tag{22}$$

$$\implies \mathbf{D_{KL}}\big[q(\mathbf{x_{1:T}}|\mathbf{x_0})||p(\mathbf{x_{1:T}}|\mathbf{x_0})\big] = \mathbb{E}_q\big[log(q(\mathbf{x_{1:T}}|\mathbf{x_0}) - log(p(\mathbf{x_{1:T}}|\mathbf{x_0})) + log(p(\mathbf{x_0}))\big] \tag{23}$$

the prior of the latent variables does not depend on q (24)

$$\mathbf{D_{KL}}\big[q(\mathbf{x_{1:T}}|\mathbf{x_0})||p(\mathbf{x_{1:T}}|\mathbf{x_0})\big] = \mathbb{E}_q\big[log(q(\mathbf{x_{1:T}}|\mathbf{x_0}) - log(p(\mathbf{x_{1:T}}, \mathbf{x_0}))\big] + log(p(\mathbf{x_0})) \tag{25}$$

$$\implies log(p(\mathbf{x_0})) = \mathbf{D_{KL}}\big[q(\mathbf{x_{1:T}}|\mathbf{x_0})||p(\mathbf{x_{1:T}}|\mathbf{x_0})\big] - \mathbb{E}_q\big[log(q(\mathbf{x_{1:T}}|\mathbf{x_0}) - log(p(\mathbf{x_{1:T}}, \mathbf{x_0}))\big] \tag{26}$$

$$log(p(\mathbf{x_0})) = \mathbf{D_{KL}}\big[q(\mathbf{x_{1:T}}|\mathbf{x_0})||p(\mathbf{x_{1:T}}|\mathbf{x_0})\big] + \mathbb{E}_q\big[log(p(\mathbf{x_{1:T}}, \mathbf{x_0}) - log(q(\mathbf{x_{1:T}}|\mathbf{x_0}))\big] \tag{27}$$

but $\mathbf{D_{KL}}\big[q(\mathbf{x_{1:T}}|\mathbf{x_0})||p(\mathbf{x_{1:T}}|\mathbf{x_0})\big] \geq 0$ (28)

$$\implies -log(p(\mathbf{x_0})) \leq \mathbb{E}_q\big[-log \frac{p(\mathbf{x_{0:T}})}{q(\mathbf{x_{1:T}}|\mathbf{x_0})}\big] \tag{29}$$

If we define the Evidence Lower Bound L as: (30)

$$L := \mathbb{E}_q \Big[-log \frac{p(\mathbf{x_{0:T}})}{q(\mathbf{x_{1:T}}|\mathbf{x_0})} \Big] \tag{31}$$

$$\implies -log[p(\mathbf{x_0})] \leq L \tag{32}$$

We have found upper bound L for the negative log-likelihood function that can be maximided in the forward process' training.

3.2.2 Noise Predictor Training

In order to use L as the loss function in our training, further algebraic manipulation must be performed

Definition 3.5 (Noise Predictor's Loss Derivation)**.** *In order to build the Noise Predictor's loss function, we need to remember that both forward and reverse processes are Markov Chains and use this fact to manipulate*

L.

$$L = \mathbb{E}_q\Big[- log\frac{p(\mathbf{x_{0:T}})}{q(\mathbf{x_{1:T}}|\mathbf{x_0})}\Big] \tag{33}$$

The forward and reverse processes are Markov Chains, so (34)

$$\mathbb{E}_q\Big[- log\frac{p(\mathbf{x_{0:T}})}{q(\mathbf{x_{1:T}}|\mathbf{x_0})}\Big] = \mathbb{E}_q\Big[log\frac{p(\mathbf{x_T})\prod_{t=1}^{T} p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})}{q(\mathbf{x_1}|\mathbf{x_0})\prod_{t=2}^{T} q(\mathbf{x_t}|\mathbf{x_{t-1}},\mathbf{x_0})}\Big] \tag{35}$$

$$= \mathbb{E}_q\Big[- log\frac{p(\mathbf{x_T})\prod_{t=1}^{T} p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})}{q(\mathbf{x_T}|\mathbf{x_0})\prod_{t=2}^{T} q(\mathbf{x_{t-1}}|\mathbf{x_t},\mathbf{x_0})}\Big] \tag{36}$$

$$= \mathbb{E}_q\Big[- log\frac{p(\mathbf{x_T})p_\theta(\mathbf{x_0}|\mathbf{x_1})\prod_{t=2}^{T} p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})}{q(\mathbf{x_T}|\mathbf{x_0})\prod_{t=2}^{T} q(\mathbf{x_{t-1}}|\mathbf{x_t},\mathbf{x_0})}\Big] \tag{37}$$

$$= \mathbb{E}_q\Big[- log\frac{p(\mathbf{x_T})}{q(\mathbf{x_T}|\mathbf{x_0})}\Big] - logp_\theta(\mathbf{x_0}|\mathbf{x_1}) + \sum_{t=2}^{T}\mathbb{E}_q\Big[- log\frac{p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})}{q(\mathbf{x_{t-1}}|\mathbf{x_t},\mathbf{x_0})}\Big] \tag{38}$$

$$= \mathbb{E}_q\Big[- log\frac{p(\mathbf{x_T})}{q(\mathbf{x_T}|\mathbf{x_0})}\Big] - logp_\theta(\mathbf{x_0}|\mathbf{x_1}) - \mathbb{E}_q\Big[\sum_{t=2}^{T}\mathbf{D_{KL}}\big[q(\mathbf{x_{t-1}}|\mathbf{x_t},\mathbf{x_0})||p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t}))\big]\Big] \tag{39}$$

We see that the first term is parameter free, because $p(\mathbf{x_T})$ is fixed and defined as a Gaussian, while $q(\mathbf{x_T}|\mathbf{x_0})$ is also Gaussian from the definition of the forward process. Hence, we are left with the second and third terms from L.

More specifically, we can conclude that maximizing ELBO (L) is equivalent to minimizing the KL-Divergence between $q(\mathbf{x_{t-1}}|\mathbf{x_t},\mathbf{x_0})$ and $p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})$

We know that both distributions are Gaussians, which makes computing the KL Divergence between them easier if we know their mean and variance. We will begin by calculating such moments for $q(\mathbf{x_{t-1}}|\mathbf{x_t},\mathbf{x_0})$.

$$q(\mathbf{x_{t-1}}|\mathbf{x_t},\mathbf{x_0}) = \frac{q(\mathbf{x_t}|\mathbf{x_{t-1}},\mathbf{x_0})q(\mathbf{x_{t-1}}|\mathbf{x_0})}{q(\mathbf{x_t}|\mathbf{x_0})} \tag{40}$$

we know the q distribution from Definition **??**, hence (41)

$q(\mathbf{x_{t-1}}|\mathbf{x_t},\mathbf{x_0})$ is a product of known Gaussians over another known Gaussian that lets us define (42)

$$\mu_q(\mathbf{x_t},\mathbf{x_0}) = \frac{(1-\bar{\alpha}_{t-1})\sqrt{\alpha_t}\mathbf{x_t} + (1-\alpha_t)\sqrt{\bar{\alpha}_{t-1}}\mathbf{x_0}}{(1-\bar{\alpha}_t)} \tag{43}$$

$$\Sigma_q(t) = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{I} \tag{44}$$

$$\implies q(\mathbf{x_{t-1}}|\mathbf{x_t},\mathbf{x_0}) = \mathcal{N}(\mathbf{x_{t-1}};\mu_q(\mathbf{x_t},\mathbf{x_0});\Sigma_q(t)) \tag{45}$$

We have just defined the $q(\mathbf{x_{t-1}}|\mathbf{x_t},\mathbf{x_0})$ distribution. Now let's move on to the $p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})$ distribution.

$$p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t}) = \mathcal{N}(\mathbf{x_{t-1}};\mu_\theta(\mathbf{x_t});\Sigma_\theta(t)) \tag{46}$$

the reverse process variance is defined as the ground truth variance of the forward process: (47)

$$\Sigma_\theta(t) = \Sigma_q(t) \tag{48}$$

we are only left with the distribution's mean μ_θ (49)

$$p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t}) = \mathcal{N}(\mathbf{x_{t-1}};\mu_\theta(\mathbf{x_t});\Sigma_q(t)) \tag{50}$$

Equation 50 makes it much easier to calculate $\mathbf{D_{KL}}\big[q(\mathbf{x_{t-1}}|\mathbf{x_t},\mathbf{x_0})||p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t}))\big]$:

Now we know we are trying to compute the KL Divergence between two Gaussians with the exact same variance.

For that, there is the following result that arives from the definition of such divergence

$$d_1(x) = \mathcal{N}(\mu_1, \sigma^2) \tag{51}$$

$$d_2(x) = \mathcal{N}(\mu_2, \sigma^2) \tag{52}$$

The KL divergence $D_{KL}(d_1|d_2)$ is given by: (53)

$$D_{KL}(d_1|d_2) = \frac{(\mu_1 - \mu_2)^2}{2\sigma^2} \tag{54}$$

Hence, (55)

$$\mathbf{D_{KL}}\big[q(\mathbf{x_{t-1}}|\mathbf{x_t}, \mathbf{x_0})||p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t}))\big] = \mathbf{D_{KL}}\big(\mathcal{N}(\mathbf{x_{t-1}}; \mu_q(\mathbf{x_t}, \mathbf{x_0}); \Sigma_q(t)), \mathcal{N}(\mathbf{x_{t-1}}; \mu_\theta(\mathbf{x_t}); \Sigma_q(t))\big) \tag{56}$$

$$= \frac{1 - \bar{\alpha}_t}{2(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})}\big|\big|(\mu_q - \mu_\theta)_2^2\big|\big| \tag{57}$$

As our goal is to minimize the KL Divergence, from equation 59 we see that such goal comes down to basically minimizing the difference $\mu_q - \mu_\theta$, i.e.,

We just need to minimize the difference between the means of the reverse and forward processes' distributions. We need to define the reverse process' distribution mean (μ_θ) prediction by taking a look at the forward process' one (μ_q).

$$\mu_q(\mathbf{x_t}, \mathbf{x_0}) = \frac{(1 - \bar{\alpha}_{t-1})\sqrt{\alpha_t}\mathbf{x_t} + (1 - \alpha_t)\sqrt{\bar{\alpha}_{t-1}}\mathbf{x_0}}{(1 - \bar{\alpha}_t)} \tag{58}$$

we can define the prediction (59)

$$\hat{\mu}_q(\mathbf{x_t}, \mathbf{x_0}) = \mu_\theta(\mathbf{x_t}) \tag{60}$$

$$= \frac{(1 - \bar{\alpha}_{t-1})\sqrt{\alpha_t}\mathbf{x_t} + (1 - \alpha_t)\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_\theta}{(1 - \bar{\alpha}_t)} \tag{61}$$

In equation 63 we see that we are using our reverse process model's prediction $\mathbf{x}_\theta$ in the prediction of its distribution's mean, which let's us rewrite $\big|\big|(\mu_\theta - \mu_q)_2^2\big|\big|$ in terms of $\mathbf{x_0}$ and $\mathbf{x}_\theta$ which leves us with the following for the KL Divergence:

$$\mathbf{D_{KL}}\big(\mathcal{N}(\mathbf{x_{t-1}}; \mu_\theta(\mathbf{x_t}); \Sigma_q(t)), \mathcal{N}(\mathbf{x_{t-1}}; \mu_q(\mathbf{x_t}, \mathbf{x_0}); \Sigma_q(t))\big) = \frac{(1 - \bar{\alpha}_t)(\bar{\alpha}_{t-1})}{2(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})}\big|\big|(\mathbf{x}_\theta - \mathbf{x_0})_2^2\big|\big| \tag{62}$$

The author's of the DDPM paper mention that equation 64 can be used as the loss function to train the reverse process model. On the other hand, we now that the forward process actually predict the noise that was added to an input $\mathbf{x_t}$ intead of predicting $\mathbf{x}_\theta$ directly. This means that the loss function must account for the error prediction somehow. This is achieved by further analysing $\mathbf{x_0}$ and $\mathbf{x}_\theta$ and remembering how $\mathbf{x}_\theta$ was defined in the forward process.

$$q(x_t|x_0) := \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} \times x_0; (1 - \bar{\alpha}_t)\mathbb{I}) \tag{63}$$

which let's us write (64)

$$\mathbf{x_t} = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon \tag{65}$$

$$\implies \mathbf{x_0} = \frac{\mathbf{x_t} - \sqrt{1 - \bar{\alpha}_t}\epsilon}{\sqrt{\bar{\alpha}_t}} \tag{66}$$

for a Standard Gaussian Noise ϵ. (67)

We can now define our prediction $\hat{\epsilon} = \epsilon_\theta$ (68)

$$\mathbf{x}_\theta = \frac{\mathbf{x_t} - \sqrt{1 - \bar{\alpha}_t}\epsilon_\theta}{\sqrt{\bar{\alpha}_t}} \tag{69}$$

$$\implies \frac{(1 - \bar{\alpha}_t)(\bar{\alpha}_{t-1})}{2(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})}\big|\big|(\mathbf{x}_\theta - \mathbf{x_0})_2^2\big|\big| = \frac{(1 - \bar{\alpha}_t)(\bar{\alpha}_{t-1})}{2(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})}\frac{(1 - \alpha_t)^2}{(1 - \bar{\alpha}_t)\alpha_t}\big|\big|(\epsilon_\theta - \epsilon)_2^2\big|\big| \tag{70}$$

Even though equation 72 could be used directly as the loss function for the noise predictor, the DDPM paper authors mention that optimizing $\big|\big|(\epsilon_\theta - \epsilon)_2^2\big|\big|$ without the scaling factor with the cumulative noise α_t is enough. Hence, we have finally defined a function to be minimized for the noise predictor training and hence write its algorithm.

Algorithm 1: Noise Predictor Training

repeat
$\mathbf{x_0} \sim \mathbf{q}(\mathbf{x_0})$ ▷ Sample image from training set
$\mathbf{x_0} \sim \mathbf{Uniform}(\{\mathbf{1}, \ldots, \mathbf{T}\})$ ▷ Sample the step of the Forward Process Markov Chain
$\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Sample standard gaussian noise to be added to the input
$\mathbf{x_t} = \sqrt{\bar{\alpha}_\mathbf{t}}\mathbf{x_0} + \sqrt{\mathbf{1} - \bar{\alpha}_\mathbf{t}}\epsilon$ ▷ Forward Process/ Generating Noisy Image
Take Gradient Descent Step on $\nabla_\theta(||\epsilon - \epsilon_\theta(\mathbf{x_t}, \mathbf{t})||)$
until converged

3.2.3 Sampling Algorithm Derivation

Now that we have defined a way to predict an input image $\mathbf{x_t}$'s noise, we need a way to use such prediction to reconstruct the original de-noised image $\mathbf{x_0}$, i.e., we need a way to sample from $p(\mathbf{x_0})$. To do so, let's recall how we have defined the $p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})$ distribution.

$$p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t}) = \mathcal{N}(\mathbf{x_{t-1}}; \mu_\theta(\mathbf{x_t}); \Sigma_q(t)) \tag{71}$$

$$\mu_\theta(\mathbf{x_t}) = \frac{(1-\bar{\alpha}_{t-1})\sqrt{\alpha_t}\mathbf{x_t} + (1-\alpha_t)\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_\theta}{(1-\bar{\alpha}_t)} \tag{72}$$

$$\mathbf{x}_\theta = \frac{\mathbf{x_t} - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta}{\sqrt{\bar{\alpha}_t}} \tag{73}$$

$$\implies \mu_\theta(\mathbf{x_t}) = \frac{\mathbf{x_t}}{\sqrt{\alpha_t}} - \frac{(1-\alpha_t)(\sqrt{1-\bar{\alpha}_t})}{(1-\bar{\alpha}_t)(\sqrt{\alpha_t})}\epsilon_\theta = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x_t} - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta\right) \tag{74}$$

$$\Sigma_\theta(t) = \Sigma_q(t) = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{I} \tag{75}$$

We now have defined $\mathbf{x_{t-1}}$'s mean and variance given $\mathbf{x_t}$ which let's us write it as (76)

$$\mathbf{x_{t-1}} = \mu_\theta(\mathbf{x_t}) + \sqrt{\Sigma_\theta(t)}\mathbf{z} \tag{77}$$

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{78}$$

We have now a way to generate $\mathbf{x_{t-1}}$ from $\mathbf{x_t}$. This means that if we have $\mathbf{x_1}$ we can generate $\mathbf{x_0}$. This let's us finally define our sampling algorithm:

Algorithm 2: Sampling

repeat
$\mathbf{x_T} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Sample random noisy image
$\mathbf{T} \sim \mathbf{Uniform}(\{\mathbf{1}, \ldots, \mathbf{1000}\})$ ▷ Sample random length of the Denoising Chain. Max chain size of 1000 was set arbitrarily
for $\mathbf{t} = \mathbf{T}, \ldots, \mathbf{1}$ **do**
$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$ else $\mathbf{z} = \mathbf{0}$
$\mathbf{x_{t-1}} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x_t} - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x_t}, \mathbf{t})\right) + \sqrt{\Sigma_\theta(t)}\mathbf{z}$ ▷ Sampling $\mathbf{x_{t-1}}$ from $p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})$
end for
return $\mathbf{x_0}$

3.3 Recap

We have studied both the forward and reverse process that make up DDPMs. We have seen that good samples (or images) are generated by maximing the reverse process' likelihood $logp_\theta(\mathbf{x_0})$ lower bound, the Evidence Lower Bound, ELBO, or simply L.

We have rewritten ELBO in a way that its maximization turns out to be dual with minimizing the Kullback-Leibler (KL) divergence between $p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})$ and $q(\mathbf{x_{t-1}}|\mathbf{x_t}, \mathbf{x_0})$. Such KL-Divergence minimization was then translated to a noise predictor training. After the predictor training algorithm was defined,

we have shown how to use such prediction to sample from $p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})$ and finally reconstructing $\mathbf{x_0}$, the original denoised image.

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020.

[2] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022.

[3] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.