Universidad de los Andes Colombia | Facultad de Ciencias Departamento de Física

Image Generation With Langevin Dynamics

Bachelor Thesis

by

David Leonardo Almanza Márquez

Departamento de Física, Universidad de los Andes
Bogotá D.C, Colombia

Supervisor
Gabriel Téllez

24/05/2024

Abstract

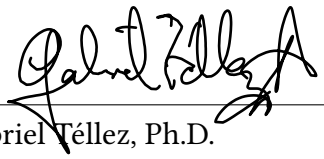
In the field of machine learning, Diffusion Probabilistic Models have emerged as a prominent category of generative models. Their main objective is to learn a diffusion process that describes the probability distribution of a given dataset. The essence of diffusion-based generative models finds its roots in statistical physics, where diffusion is modeled by describing the Brownian motion of particles through a physical system. Inspired by these concepts, diffusion generative models adopt diffusion as a central process. By understanding how data diffuse in an abstract space, these models capture inherent patterns and generate synthetic data that reflects the underlying structure of real datasets.

This study undertakes a comparative analysis between Denoising Diffusion Probabilistic Models and statistical physics. It was discovered that the "diffusion" process of data, as elucidated in the seminal paper by Jo et al. (2020), can be effectively explained using a specialized version of the Langevin Equation. Building upon this understanding and leveraging the work of Song et al. on score-based generative modeling through stochastic differential equations, we expanded the framework of DPMs by examining continuous time diffusion processes, as opposed to the discrete time framework presented by Jo et al. This continuous time perspective provided deeper insights into the physical principles underlying DPMs. Furthermore, it facilitated new observations and advancements in the development of DPMs, enhancing our comprehension and application of these models.

Finally, a diffusion model was developed for two-dimensional and three-dimensional physical systems. This facilitated the learning of various out-of-equilibrium particle position distributions, thereby providing empirical validation for the efficacy of DPMs. Additionally, a diffusion model was implemented to generate 64x64 pixel images of flowers, further demonstrating the practical utility and versatility of these models. This implementation underscored the potential of DPMs in generating high-quality synthetic data across different types of datasets.

Keywords: Diffusion Probabilistic Models, Machine Learning, Statistical Physics, Langevin Equation, Synthetic Data Generation.

Title:	Image Generation With Langevin Dynamics
Author:	David Leonardo Almanza Márquez
Advisor:	Gabriel Téllez
Study programme:	Física
Institution:	Universidad de los Andes - Departamento de Física
Year:	2024
Pages:	43



Gabriel Téllez, Ph.D.
Professor
Departamento de Física - Universidad de los Andes

Acknowledgements

I would like to express my deepest gratitude to everyone who supported me throughout the development of this thesis. My heartfelt thanks go to my family—my parents and my brother—for their unwavering support and encouragement. I am also profoundly grateful to the Statistical Physics group and to Professor Gabriel Téllez. This thesis would not have been possible without their invaluable guidance and assistance.

Contents

Chapter 1

Introduction

The concept of Diffusion Probabilistic Models (DPM) originated in 2015, thanks to the pioneering work of Jascha et al. in the influential paper "Deep Unsupervised Learning using Nonequilibrium Thermodynamics" [1]. In this seminal document, Sohl-Dickstein introduces an innovative approach for unsupervised learning of probability distributions by applying principles of nonequilibrium thermodynamics. The foundation of DPM lies in constructing Markov chains that transform a simple probability distribution into the desired probability distribution for the data through small perturbations. This concept provides a unique and powerful perspective for modeling the intrinsic complexity of datasets. Subsequent research, such as the work titled "Denoising Diffusion Probabilistic Models" [2], has expanded and refined these principles, solidifying DPM as a crucial tool in the landscape of machine learning and image generation.

More recent advancements in Diffusion Probabilistic Models have significantly enhanced their efficiency, training stability, and application scope. Innovations like DiffuSSM[3], which integrates State Space Models to reduce reliance on attention mechanisms, have improved computational efficiency. Advances in score-based generative modeling and the development of techniques for disentangled representations, such as DisDiff[4], have further stabilized training and increased control over generated outputs. DPMs have also been applied successfully in domains like text-to-image synthesis and medical imaging, demonstrating their versatility and potential for generating high-quality, coherent images across various contexts.

1.1 Research Problem

Consider the task of generating highly complex data, such as images belonging to a specific category, like cats. Despite our visual familiarity with the appearance

of a cat image, the precise characteristics of the underlying probability distribution governing these images remain elusive. Consequently, sampling a new image $x_{new} \sim P_{data}(x)$ presents a challenge. This challenge comes from the intrinsic complexity of the probability distribution P_{data}. The high dimensionality of the distribution (proportional to the number of pixels in the image) and the multitude of features, poses, and environmental contexts in which cats can be depicted contributes to the complexity and the ambiguity surrounding the formulation of this distribution.

Furthermore, it is worth highlighting that, in order to learn about this probability distribution, we only have access to a finite subset of representative image samples x such that $x \sim P_{data}(x)$. These images constitute our sole source of observable information to tackle the challenge of learning the target probability distribution. The problem of modeling the intricate probability distribution of cat images from a limited set of instances arises the need for sophisticated approaches. In this context, Diffusion Generative Models, which explore the application of particle diffusion processes to effectively allow sampling from complex distributions, become relevant.

Chapter 2

Denoising Diffusion Probabilistic Models

The process of learning the target probability distribution involves two major components: the *Forward Diffusion Process* and the *Backward Diffusion Process*. In the forward process, the primary objective is the gradual and controlled destruction of the data structure. To achieve this goal, the main strategy involves selecting images $x_0 \sim q(x_0)$ from our dataset, and then iteratively introducing Gaussian noise into them. This process is repeated until, after a sufficient number T of iterations, the dataset has transformed into pure Gaussian noise, and thus, our images become $x_T \sim q(x_T)$, with $\lim_{T\to\infty} q(x_T) = \mathcal{N}(0, I)$, i.e., a Gaussian distribution centered at the origin with covariance matrix I: pure Gaussian noise.

The second stage, associated with the generation of new data, is denominated the Backward Diffusion Process. During this phase, the premise is to reverse the Forward Diffusion Process, thereby enabling a stochastic process in which Gaussian noise is iteratively removed from an initial image x_T, which, as its subscript T suggests, initially consists of pure Gaussian noise. This iterative procedure is carried out until, after exactly the same number of iterations as it took for the Forward Process, we obtain a completely new image x_0, distributed according to our target distribution $q(x_0)$. Figure 2.1 provides a more visual explanation of the Forward an Backward Diffusion Processes.

2.1 Forward Diffusion Process

We start with our set of data-points x_0, aiming to learn the underlying probability distribution $q(x_0)$. Remember, the objective of the forward diffusion process is

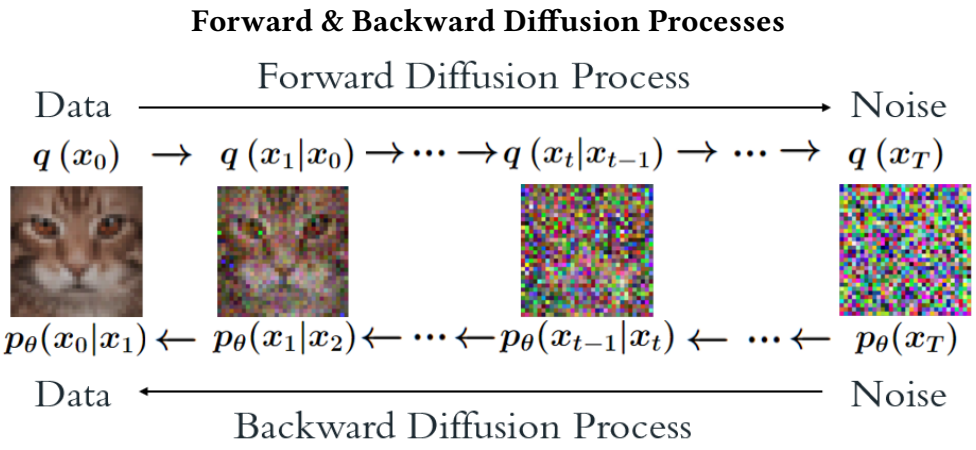


Figure 2.1: Illustration of the Forward Diffusion Process and the Backward Diffusion Process. In the Forward process, as we progress through time t, the original structure of the image is gradually destroyed due to the addition of Gaussian noise at each time step. At $t = T$, the resulting image is pure Gaussian noise. In the case of the Backward Process, the aim is to reverse the diffusion process, thereby generating a completely new image from Gaussian noise.

to destroy data through long enough until at iteration step $t = T$ it converges to pure Gaussian noise. The Forward diffusion process step for an element x_{t-1} is defined in [2] as follows:

$$q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t I), \tag{2.1}$$

where $q(x_t|x_{t-1})$ is the conditional probability distribution of going from x_{t-1} to x_t, $\mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t I)$ represents a Gaussian distribution with mean $\mu_q(t, x_{t-1}) = \sqrt{1-\beta_t}x_{t-1}$ and covariance matrix $\Sigma(t) = \beta_t I$. The subscript t denotes the iteration step the Forward Diffusion Process is presently undergoing. Consequently, this process conforms to a Markovian nature since the probability distribution of the dataset at time t relies solely on time $t-1$. β_t serves as a parameter dictating the diffusion at each time step. We may interpret this parameter as the degree of noise introduced to the image, considering the process has undergone t steps. This parameter was originally scheduled in [2] to grow lineally from $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$.

Up to this point, it may result hard for the reader to comprehend why is this called a *Diffusion* process. To establish a connection to statistical physics, let's reconsider x_0, previously construed as an image, as the position of a particle in space. With this reinterpretation, $q(x_0)$ then represents the positions distribution of particles at time $t = 0$. In this analogy, our aim is to comprehend the

position of an image x_0 in probability space as the position of a particle at time $t = 0$ within a physical system characterized by certain attributes, which, as we shall explore, are determined by the functional form of the forward step.

Equation (2.1) describes the Brownian motion of the particle through the system:

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon, \tag{2.2}$$

where $\epsilon \sim \mathcal{N}(0, I)$. So, the position of the particle at time t depends on its immediately previous position x_{t-1}, scaled by a factor $\sqrt{1-\beta_t}$, plus some random noise ϵ, which is scaled by a factor contingent upon β_t.

Assuming that β_t is sufficiently small, we can approximate $(1-\beta_t)^{1/2} \approx 1 - \frac{1}{2}\beta_t + O(\beta_t^2)$. This approximation yields

$$x_t - x_{t-1} = -\frac{1}{2}\beta_t\, x_{t-1} + \sqrt{\beta_t}\,\epsilon. \tag{2.3}$$

Aiming for a continuous limit, we take $\beta_t \to \beta(t)dt$ and $\sqrt{dt}\epsilon \to dW$, a Wiener process, which yields the following stochastic differential equation:

$$dx = -\frac{1}{2}\beta(t)\, x\, dt + \sqrt{\beta(t)}dW. \tag{2.4}$$

A Wiener process W_t is a stochastic process $W_t \in \mathbb{R}$, $t \in [0, \infty)$ that fulfills a set of conditions [5]. In particular, we are interested in the condition

- For every $t > 0$, the future increments $W_{t+\Delta t} - W_t$ are Gaussian $W_{t+\Delta t} - W_t \sim \mathcal{N}(0, \Delta t)$.

The difference of a Wiener process $W_{t+dt} - W_t$ is normally distributed and of magnitude $\sqrt{\Delta t}$. If we take an infinitesimally small Δt, then $dW_t = W_{t+dt} - W_t$ and the aforementioned condition suggests

$$\langle dW_t \rangle = 0 \;\;\&\;\; \langle dW_t^2 \rangle = dt. \tag{2.5}$$

Condition which will prove useful later in Chapter 3, when we try to find a more general formulation for the Diffusion Processes.

Our new objective is to demonstrate that the Diffusion process (2.4) corresponds to the behavior of a harmonic oscillator ($F(x) = -kx$) with a mass m in a fluid where the damping constant is γ, as described by the Langevin Equation:

$$m\frac{d^2x}{dt^2} = F(x) - \gamma\frac{dx}{dt} + \eta(t). \tag{2.6}$$

Where $\eta(t)$ is a random force that accounts for random collisions with other particles in the fluid.

If we consider the system in the special case of overdamped dynamics, where $d^2x/dt^2 = 0$, then, at equilibrium temperature T, the particle's trajectory is governed by the overdamped Langevin Equation [6]:

$$\frac{dx}{dt} = -\frac{k}{\gamma}x + \frac{\sqrt{k_BT}}{\gamma}\epsilon, \tag{2.7}$$

being k_B the Boltzmann constant and T the temperature of the system. By defining $k = \beta(t)$, $\gamma = 1$ and $k_BT = \beta(t)$, we recover equation (2.4). This establishes a compelling link between the diffusion process inherent in DMPs and the dynamics of a particle undergoing overdamped motion in a fluid.

The equilibrium positions distribution $q(x_T)$ for such a diffusion process is characterized by the Boltzmann distribution:

$$q(x_T) \propto e^{\frac{-U(x)}{k_BT'}} = e^{\frac{-1/2kx^2}{k_BT'}}, \tag{2.8}$$

this T' being the temperature at final time T. Utilizing the earlier definitions for k_BT and k, we deduce

$$q(x_T) \propto e^{-1/2x^2} = \mathcal{N}(0, I), \tag{2.9}$$

result that aligns with the equilibrium distribution anticipated for the image diffusion process, earlier in this text.

This way, we notice that running the Forward Diffusion Process in an image x_0 is equivalent to simulating Brownian motion under certain conditions, imbued with physical significance, for the initial position x_0 of the image within probability space. Thus, learning the target distribution $q(x_0)$ implies, in our analogy, learning an out-of-equilibrium particle positions distribution $q(x_0)$. Figure 2.2 gives a general idea of the parallel we have built in this section of the text.

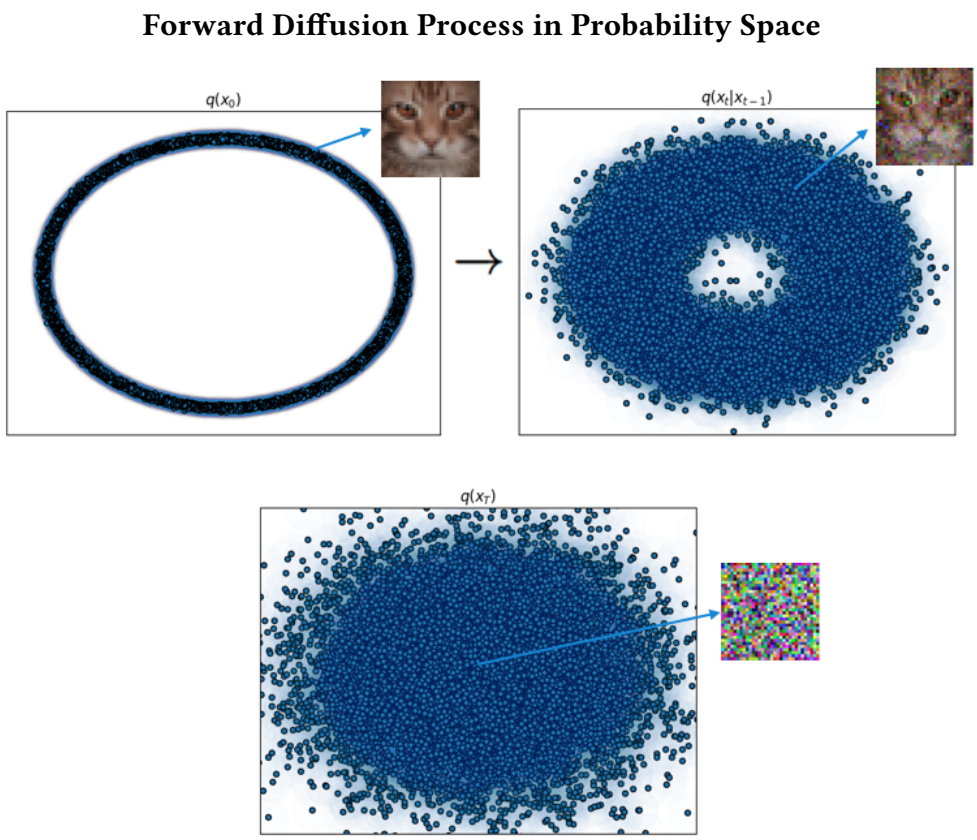


Figure 2.2: Simplified Illustration of the Forward Diffusion Process for the entire dataset, depicted in probability space. Here, we assume the original data distribution $q(x_0)$ forms a circular shape. The blue particles, each representing an image in the dataset, undergo a diffusion process from $q(x_0)$, an out-of-equilibrium distribution, towards a Gaussian equilibrium distribution $q(x_T)$, once a sufficient amount of time T has elapsed.

2.1.1 Forward Process Re-parametrization

Due to the Markov chain structure inherent in the Forward Diffusion process (2.1), we can introduce the following reparametrization to simplify the expression:

$$\alpha_t := 1 - \beta_t, \tag{2.10}$$

where α_t can be interpreted as a measure of how much of the original image remains intact at time t (Recall that β_t increases linearly, and hence, α_t decreases

likewise). Then, if we define

$$\overline{\alpha}_t = \prod_{i=1}^{t} \alpha_i, \tag{2.11}$$

with this, we can write the distribution $q(x_t|x_0)$ as follows:

$$q(x_t|x_0) = \mathcal{N}\left(\sqrt{\overline{\alpha}_t}x_0, (1-\overline{\alpha}_t)I\right). \tag{2.12}$$

It is important to note that, although this reparametrization may lack a direct physical interpretation for $\overline{\alpha}_t$, it provides us with the advantage of being able to reach the diffusion time t directly from time 0:

$$x_t = \sqrt{\overline{\alpha}_t}x_0 + \sqrt{1-\overline{\alpha}_t}\epsilon. \tag{2.13}$$

This new parametrization will prove beneficial later on, particularly in the definition of the Backward Diffusion Process. For the time being, remember the necessity of sampling **only one** ϵ.

2.2 Backward Diffusion Process

Having explained the Forward Diffusion Process $q(x_t|x_{t-1})$ and its connection to physics, along with a comprehension of its functioning and purpose, we can now delve into the discussion of the backward diffusion process. As mentioned above, this process ($q(x_{t-1}|x_t)$) is the inverse of the forward diffusion process, and its primary objective is to facilitate sampling, starting from Gaussian noise x_T, to generate new images x_0 distributed according to the target distribution $q(x_0)$. **This** is the process we aim to parameterize. However, it is worth noting that this process, in order to reverse diffusion, traverses backward in time ($x_t \rightarrow x_{t-1}$). However, in the preceding section, we constructed a process that progresses forward in time ($x_{t-1} \rightarrow x_t$).

Based on the structure of the Forward Process, we can infer certain characteristics of the Backward Process. The key insight lies in the fact that when the step size Δt of the Forward Process is sufficiently small, we can safely assume that the functional form of the Backward Process is the same as that of the Forward Process: a Gaussian. In this context, $q(x_{t-1}|x_t)$ serves as a posterior probability

distribution, computable using the Bayes rule if it is conditioned on x_0 [2]. With this in mind, we define

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(\tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I) \tag{2.14}$$

$$\text{with } \tilde{\mu}_t(x_t, x_0) := \frac{\sqrt{\overline{\alpha}_t}\beta_t}{1-\overline{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})}{1-\overline{\alpha}_t}x_t \;\&\; \tilde{\beta}_t := \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t. \tag{2.15}$$

In order to learn the series of distributions involved in the Backward Diffusion Process, we opt to parameterize a function $p_\theta(x_{t-1}|x_t)$ with the aim of approximating, to the best of our data's capability, the posterior distributions $q(x_{t-1}|x_t)$. (It is crucial to recall that we lack precise knowledge of the q distributions; instead, we possess datapoints or particles x distributed according to q).

Once again, given the functional form in which the diffusion process occurs, we define the parameterization of $p_\theta(x_{t-1}|x_t)$ as follows:

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \tag{2.16}$$

It is important to note that parameterizing $p_\theta(x_{t-1}|x_t)$ entails parameterizing both the mean μ_θ and the covariance Σ_θ of a series of Gaussian distributions. In other words, we need to learn the set of parameters θ^* that minimizes the discrepancy between the distributions p_θ and q for each time t. This minimization is performed with respect to the Kullback-Leibler divergence (D_{KL}), a measure of the difference between the two probability distributions p_θ and q. The loss function to minimize is expressed as:

$$\mathbb{E}_q[D_{KL}(q(x_T|x_0)||p(x_T)) + \sum_{t>1} D_{KL}(q(x_{t-1}|x_t, x_0)||p(x_{t-1}|x_t)) - \log p_\theta(x_0|x_1)]. \tag{2.17}$$

This explanation provides a heuristic to understand the loss function. Minimizing this function allows us to learn the parameters θ^* that describe the diffusion process optimally.

Let's delve deeper into the parameterization of $p_\theta(x_{t-1}|x_t)$ for $1 < t \leq T$. First, note that, thanks to the schedule defined for our noise β_t in the forward diffusion process, a convenient approximation is $\Sigma_\theta(x_t, t) = \beta_t I$ [2]. This implies that we do not need to parameterize the covariance matrix, as its form is known beforehand. With this in mind, we rewrite

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t, t), \beta_t I). \tag{2.18}$$

As both distributions are assumed to have the same covariance matrix, the only aspect left to reduce the difference between the two probability distributions is to minimize the difference between their means μ. With this in mind, the loss function, specifically for time $t-1$, becomes:

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\beta_t} \|\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)\|^2 \right] \tag{2.19}$$

Resuming our analogy in terms of statistical physics, we are seeking the parameters θ^* that minimize the difference between the particles' actual average position $\tilde{\mu}$ and the average position of our parameterized distribution μ_θ (see Figure 2.3). This process is repeated for each time t in the diffusion chain.

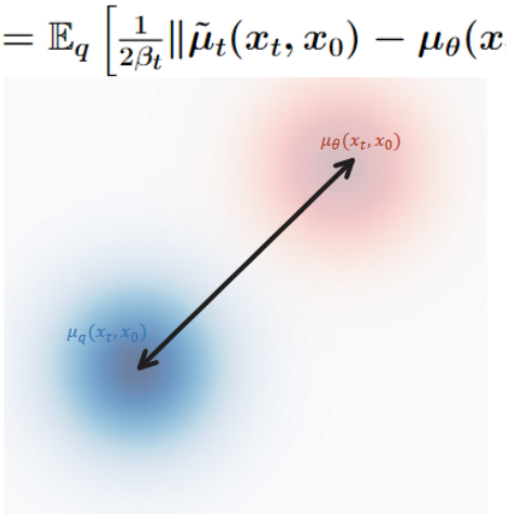


Figure 2.3: Representation of the optimization process. The figure illustrates the minimization of the difference between the actual average position $\tilde{\mu}$ and the average position of the parameterized distribution μ_θ. This procedure is repeated for each time step t in the diffusion chain.

We can continue simplifying the loss function (2.19). To do this, we first solve for x_0 from (2.13), obtaining:

$$x_0 = \frac{1}{\sqrt{\overline{\alpha}_t}}\left(x_t - \sqrt{1-\overline{\alpha}_t}\epsilon\right), \tag{2.20}$$

we then substitute this expression into (2.19), resulting in:

$$L_{t-1} = \mathbb{E}_{x_0,\epsilon}\left[\frac{1}{2\beta_t}\left\|\tilde{\mu}_t\left(x_t(x_0,\epsilon), \frac{1}{\sqrt{\overline{\alpha}_t}}(x_t(x_0,\epsilon) - \sqrt{1-\overline{\alpha}_t}\epsilon)\right) - \mu_\theta(x_t(x_0,\epsilon),t)\right\|^2\right]. \tag{2.21}$$

We use the definition of $\tilde{\mu}_t$ (2.15), and simplify:

$$L_{t-1} = \mathbb{E}_{x_0,\epsilon}\left[\frac{1}{2\beta_t}\left\|\frac{1}{\sqrt{\alpha_t}}\left(x_t(x_0,\epsilon) - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon\right) - \mu_\theta(x_t(x_0,\epsilon),t)\right\|^2\right], \tag{2.22}$$

Thus, we have proved that μ_θ must predict $\frac{1}{\sqrt{\alpha_t}}\left(x_t(x_0,\epsilon) - \frac{\beta_t}{\sqrt{1-\overline{\alpha}t}}\epsilon\right)$. The only unknown there is ϵ. We then choose the parameterization for μ_θ as:

$$\mu_\theta(x_t,t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t(x_0,\epsilon) - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta(x_t,t)\right), \tag{2.23}$$

And we note that we are attempting to predict ϵ. We will do this by parameterizing the function ϵ_θ. The final loss formula is then:

$$L_{t-1} = \mathbb{E}_{x_0,\epsilon}\left[\frac{\beta_t}{2\alpha_t(1-\overline{\alpha}_t)}\left\|\epsilon - \epsilon_\theta(\sqrt{\overline{\alpha}_t}x_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, t)\right\|^2\right]. \tag{2.24}$$

This equation tells us that we are attempting to predict the noise ϵ (see eq. 2.13) present in the image x_t, given that t steps have passed in the diffusion chain. To fulfill this task, we have x_0, the noise-free image.

Backward Process Summary and Training Algorithm

In this section, we first proposed a framework to reverse the Forward Diffusion Process. By accomplishing this, we would establish a Backward Diffusion Process enabling us to generate new images $x_0 \sim q(x_0)$. To achieve this, we initially

randomly sample a fully Gaussian noise image x_T and traverse a series of distributions $q(x_{t-1}|x_t)$ to eliminate the noise supposedly present in x_t, yielding a less noisy image x_{t-1}. To parameterize the distributions $q(x_{t-1}|x_t)$, we introduced $p_\theta(x_{t-1}|x_t)$ and formulated a loss function aimed at minimizing the discrepancy between the distributions across all time steps t. Ultimately, we demonstrated that to adequately parametrize p_θ, we only need to predict ϵ in the equation (2.13). To accomplish this, we propose training a Deep Neural Network to serve as ϵ_θ. With these components in place, the training algorithm can be summarized as follows.

1. **Repeat until convergence**
2. Take an image $x_0 \sim q(x_0)$
3. Sample a random time $t \sim \text{Uniform}(1, 2, ..., T)$
4. Sample $\epsilon \sim \mathcal{N}(0, I)$
5. Perform gradient descent on $\nabla_\theta \left| \epsilon - \epsilon_\theta \left(\sqrt{\overline{\alpha}_t} x_0 + \sqrt{1 - \overline{\alpha}_t} \epsilon, t \right) \right|^2$

2.3 Sampling Process

Until now, in our efforts to derive a set of distributions $p_\theta(x_0)$ expected to facilitate the creation of new images x_0 that accurately reflect the actual distribution $q(x_0)$ has led us to formulate a function ϵ_θ for estimating the noise within an image. This section will focus on examining the technique we intend to use to achieve our goal of new data generation: *Langevin Dynamics Sampling*. Before detailing its specific application within the context of Probabilistic Diffusion Models, we will provide a general overview of the method. Subsequently, we will examine how this approach is integrated into our research, highlighting the specific considerations necessary within the framework of probabilistic diffusion models.

2.3.1 Langevin Dynamics Sampling

We revisit the over-damped regime of the Langevin Equation. Similarly to how we previously modeled forward diffusion in probability space using the Langevin Equation for the non-equilibrium image distribution $q(x_0)$, we now seek to reverse the diffusion process from the equilibrium state $q(x_T)$ back to $q(x_0)$ utilizing Langevin dynamics as well. As we will explore, our selection of this method

was not solely because the forward process involves diffusion; there are multiple reasons why we opted to employ this dynamic for sampling, which we will elaborate on.

It has been noted multiple times that the real distributions $q(x_t)$ are inherently complex. By parametrizing $p_\theta(x_t)$ closely to $q(x_t)$ through the optimization process given by the training algorithm, it is expected for p_θ to exhibit certain flaws. Figure 2.4 displays an actual data distribution p_{Data} on the left, and its parametrized distribution p_θ on the right. Such discrepancies can cause issues in data generation, as an image in process of generation might diffuse towards these areas of probability density that correspond to estimation inaccuracies.

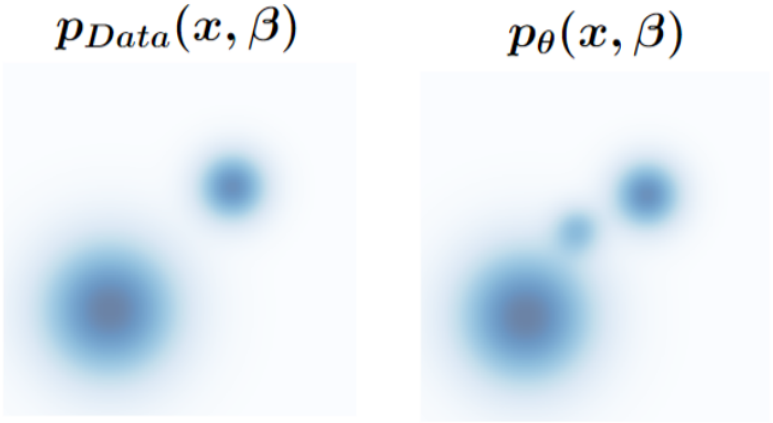


Figure 2.4: On the left, a representation of the actual distribution of the data p_{Data}. On the right, the achieved parameterized distribution, p_θ. For the aproximated distribution, there is a third well, associated to error of estimation.

An advantage of employing the Langevin equation becomes apparent: Should the image during generation settle into an estimation error *local minima potential well*, the stochastic force's perturbations enable the image to escape this minimum, ideally not a profound one. Consequently, the image can proceed along its path until it stabilizes at a global minimum. Ultimately, this process results in an image $x \sim q(x_0)$.

To grasp another advantage of employing the *Langevin Sampling* technique, reconsider the sample creation process with no stochastic force. Imagine that the image has reached a global minimum. By the conclusion of the sampling process, in the absence of a stochastic force, the image finds itself at the lowest point of the well. Subsequently, we attempt to produce another image, only to discover at the process's end that it is the same as the first. Both images, being

at the bottom of the well, are identical. This is suboptimal, as our objective is to produce a unique image with each iteration. Nevertheless, this is achievable due to the fluctuations brought about by the stochastic force. Even when the image reaches the well, it does not settle at precisely the same spot every time, enabling the generation of distinct images on each occasion.

Specifically, in the context of Diffusion Probabilistic Models[2], the version of the discretized Langevin equation followed for backward diffusion is

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sqrt{\beta_t}z, \tag{2.25}$$

with $z \sim \mathcal{N}(0, I)$. In this equation, the part with ϵ_θ resembles a step provoked by a deterministic force going against the noise in probability space, and $\sqrt{\beta_t}z$ a step corresponding to a stochastic force.

Let us also remember from (2.23) that

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t(x_0, \epsilon) - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta(x_t, t)\right),$$

and, therefore, the left hand side of the backward diffusion Langevin equation corresponds to a displacement proportional to the average position $\mu_\theta(x_t, t)$ of the images at the next time in the inverse diffusion chain, $t-1$. This iterative procedure is performed from time $t = T$ until $t = 0$. When $t = 0$, we have a new image $x \sim q(x_0)$.

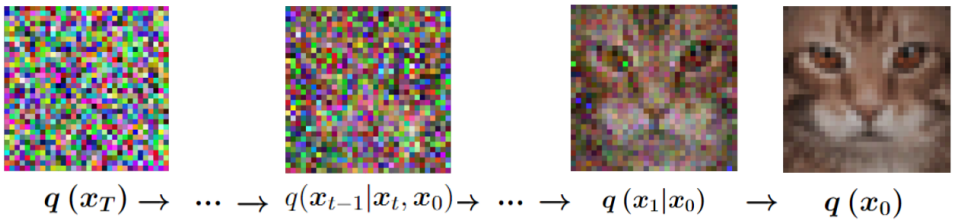


Figure 2.5: Representation of the process of generating new images. We start with a Gaussian noise image x_T, and follow the discretized Langevin equation (2.25) until converging on an image $x_0 \sim q(x_0)$.

Sampling Algorithm

The algorithm followed in order to generate new samples $x_0 \sim q(x_0)$ is as follows:

1. Take a pure Gaussian noise image $x_T \sim \mathcal{N}(0, I)$
2. **for** $t = T, T-1, ..., 1$ **:**
3. $\quad z \sim \mathcal{N}(0, I)$
4. $\quad x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sqrt{\beta_t} z$
5. **End for**
6. **return** x_0.

In this section, we have developed a comprehensive sampling process based on Langevin Dynamics to effectively generate new data that mirrors the original distribution. By leveraging the stochastic and deterministic components of the Langevin equation, we have addressed the challenges associated with estimation inaccuracies and ensured the generation of unique images at each iteration providing a robust framework for data generation.

Chapter 3

A Generalized Framework Through Stochastic Differential Equations

The technique of generating data from noise extends beyond the methodologies presented by [1, 2] and the Diffusion Probabilistic Models examined in this thesis. In other notable works, such as the paper by Song and Ermon, *Generative Modeling by Estimating Gradients of the Data Distribution* [7], a similar approach is employed where the sampling of new images is also achieved using Langevin Dynamics. This chapter delves into the broader concept of generative modeling through stochastic differential equations (SDEs), presenting it as a generalization of the Diffusion Probabilistic Models.

Song et al. in [8] propose a generalization of the diffusion probabilistic modeling method, exploring Forward and Backward processes in continuous time. They demonstrate that the Forward Diffusion Process of an image in probability space can be described by the more general stochastic differential equation (SDE)

$$dx = f(x,t)dt + g(t)dW, \tag{3.1}$$

where $f(x,t)dt$ refers to the deterministic component of the diffusion process, $g(t)dW$ denotes the stochastic part, and dW is a standard Wiener process differential (2.5). This formulation aligns with the Langevin equation in the overdamped regime, reinforcing its relevance in modeling diffusion processes.

Corresponding to this forward SDE is a reverse SDE capable of reversing the diffusion process, as described in [9],

$$dx = [f(x,t) - g(t)^2 \nabla_x \log p_t(x)]dt + g(t)dW. \tag{3.2}$$

In this equation, $p_t(x)$ represents a continuum of distributions that evolve over time, describing the diffusion process. Notably, dt is a **negative** time-step and

dW is a Wiener process that flows backward in time, from T to 0.

By comparing Equation (3.1) with (2.4), it becomes evident that $f(x,t) = -\frac{1}{2}\beta(t)\,x$, representing the force associated with a harmonic potential well, and $g(t) = \sqrt{\beta(t)}$, representing the stochastic force accounting for random collisions. Consequently, to reverse the diffusion process using the Reverse SDE (3.2), we require $\nabla_x \log p_t(x)$, which is known in the field of machine learning as the *Score Function*. This necessitates the parametrization of a series of distributions to compute their scores, thereby enabling the reversal of the diffusion process and the creation of new data samples.

To understand the concept of the score function and its physical meaning within the context of Diffusion Probabilistic Models (DPMs), let us simplify by considering a one-dimensional distribution. According to Equation (2.1), we have:

$$q(x_t|x_{t-1}) = \frac{1}{Z}\exp\frac{-\left(x_t - x_{t-1}\sqrt{1-\beta_t}\right)^2}{2\beta_t}, \tag{3.3}$$

where Z is the normalization factor. The score function of the distribution at time t is defined as the gradient of the log-probability with respect to x_t

$$\nabla_x \log q(x_t|x_{t-1}) = \frac{d}{dx_t}\left[-\frac{\left(x_t - x_{t-1}\sqrt{1-\beta_t}\right)^2}{2\beta_t}\right]$$

Carrying out the differentiation, we get:

$$\nabla_x \log q(x_t|x_{t-1}) = -\frac{\left(x_t - x_{t-1}\sqrt{1-\beta_t}\right)}{\beta_t}.$$

Using the definition of x_t given by equation (2.2)

$$x_t = \sqrt{1-\beta_t}, x_{t-1} + \sqrt{\beta_t}, \epsilon,$$

we can substitute x_t into the score function

$$\nabla_x \log q(x_t|x_{t-1}) = -\frac{\left(\sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon - x_{t-1}\sqrt{1-\beta_t}\right)}{\beta_t}.$$

Simplifying the expression, we obtain

$$\nabla_x \log q(x_t|x_{t-1}) = -\frac{\epsilon}{\sqrt{\beta_t}}. \tag{3.4}$$

We conclude that the score function is proportional to the noise vector ϵ scaled by a factor of $\frac{1}{\sqrt{\beta_t}}$. This noise vector ϵ is the same one that we proposed to predict using a deep neural network, ϵ_θ, in Chapter 2. Hence, unbeknownst to us, we have been learning the score function of the diffusion process all along. This realization bridges the gap between the stochastic differential equation framework and the practical implementation of DPMs, showcasing how the learned noise predictions directly correspond to the essential component that is the score function required for the reverse diffusion process.

It is interesting to note that these score functions are designed to create vector fields that point in the direction of increasing probability density. In Figure 3.1, we first see an arbitrary distribution $p(x)$, along with data points sampled from this distribution, $x \sim p(x)$. The corresponding score function is shown on the right side of the figure. The plot of the score function represents a vector field that indicates the direction of the steepest ascent of $p(x)$. This means that each vector in the field points towards regions where the probability density increases the most rapidly, effectively guiding the sampling process to generate data points that are more likely to occur under the true distribution $p(x)$.

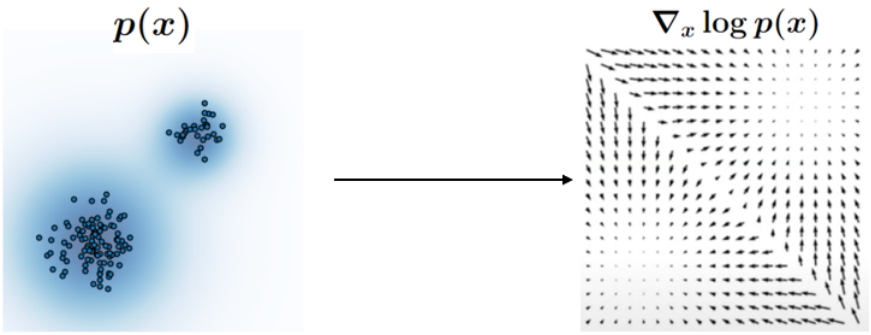


Figure 3.1: On the left, an arbitrary distribution $p(x)$ is depicted along with data points $x \sim p(x)$. On the right, the corresponding score function is illustrated. The plot of the score function represents a vector field indicating the direction of the greatest ascent of the distribution $p(x)$.

It is evident that the sampling equation (2.25) resembles a discretization of the reverse diffusion equation (3.2). During the sampling process, the image iteratively takes steps, guided by a force that directs it towards a distribution evolving over time, $p_t(x)$. At time $t = 0$, we expect the image to be located at a position $x \sim p_0(x)$, the target distribution.

3.0.1 Probability Flow & Deterministic Sampling

In relation to the diffusion process (3.1) governed by an SDE, our objective is to identify a corresponding deterministic process governed by an ODE that exhibits a probability distribution that evolves identically over time. Essentially, we aim for both processes to exhibit the same series of probability distributions $\{p_t(x)\}_{t=0}^{T}$. To achieve this, Song et al. in [8] employ the Fokker-Planck equation linked to the forward diffusion process (3.1) to examine the temporal evolution of the probability density. Following certain simplifications, a fully deterministic process emerges, enabling us to establish a method for deterministically sampling new images.

Fokker-Planck Equation

The Fokker-Planck equation is a partial differential equation that describes the temporal evolution of a probability density function. Before delving into the specifics, let us first establish a relation between the Langevin equation and the Fokker-Planck equation. To do this, we consider the Langevin equation in its more general form than (3.1)

$$dx = f(x,t)dt + G(x,t)dW, \tag{3.5}$$

where the stochastic part can now also depend on x (just for generality of the proof purposes, we will go back to G depending only on t later). Let us also consider an arbitrary function $u(x,t)$ differentiable twice on x and once on t. Then, according to Itô's lemma, a differential du is given by

$$du(x,t) = \frac{\partial u}{\partial t}dt + \frac{\partial u}{\partial x}dx + \frac{1\partial^2 u}{2\partial x^2}dx^2. \tag{3.6}$$

We substitute with (3.5), yielding

$$= \left(\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x}f(x,t)\right)dt + \frac{\partial u}{\partial x}G(x,t)dW + \frac{\partial^2 u}{\partial x^2}f(x,t)G(x,t)dt\ dW + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}G(x,t)^2dW^2, \tag{3.7}$$

where dt^2 is assumed to tend to 0. Now, we take the expectation of the entire expression, and given the conditions of the expectations of the Wiener process differential (2.5), we find

$$\langle du\rangle = \left(\langle\frac{\partial u}{\partial t}\rangle + \langle\frac{\partial u}{\partial x}f(x,t)\rangle\right)dt + \frac{1}{2}\langle\frac{\partial^2 u}{\partial x^2}G(x,t)^2\rangle dt \tag{3.8}$$

$$\frac{d}{dt}\langle u\rangle = \langle\frac{\partial u}{\partial t}\rangle + \langle\frac{\partial u}{\partial x}f(x,t)\rangle + \frac{1}{2}\langle\frac{\partial^2 u}{\partial x^2}G(x,t)^2\rangle. \tag{3.9}$$

On the left hand side of the equation we have

$$\frac{d}{dt}\langle f\rangle = \frac{d}{dt}\int_{-\infty}^{\infty} f(x,t)p(x,t)dx = \int_{-\infty}^{\infty}\frac{\partial u}{\partial t}p(x,t)dx + \int_{-\infty}^{\infty} u(x,t)\frac{\partial p}{\partial x}dx, \tag{3.10}$$

and on the right hand side

$$= \int_{-\infty}^{\infty}\frac{\partial u}{\partial t}p(x,t)dx + \int_{-\infty}^{\infty}\frac{\partial u}{\partial x}f(x,t)p(x,t)dx + \frac{1}{2}\int_{-\infty}^{\infty}\frac{\partial^2 u}{\partial x^2}G(x,t)^2p(x,t)dx. \tag{3.11}$$

Combining these two equations yields

$$\int_{-\infty}^{\infty} u(x,t)\frac{\partial p}{\partial x}dx = \int_{-\infty}^{\infty}\frac{\partial u}{\partial x}f(x,t)p(x,t)dx + \frac{1}{2}\int_{-\infty}^{\infty}\frac{\partial^2 u}{\partial x^2}G(x,t)^2p(x,t)dx, \tag{3.12}$$

and integrating by parts the right hand side of the equation, we deduce

$$= -\int_{-\infty}^{\infty} u(x,t)\frac{\partial}{\partial x}\left(f(x,t)p(x,t)\right)dx + \frac{1}{2}\int_{-\infty}^{\infty} u(x,t)\frac{\partial^2}{\partial x^2}\left(G(x,t)^2p(x,t)\right)dx. \tag{3.13}$$

Finally, we take $u(x,t) = 1$, and conclude with the Fokker-Planck equation for the one dimension case, as proved in [5].

$$\frac{\partial p(x,t)}{\partial t} = -\frac{\partial}{\partial x}\left(f(x,t)p(x,t)\right) + \frac{1}{2}\frac{\partial^2}{\partial x^2}\left(G(x,t)^2p(x,t)\right). \tag{3.14}$$

In a framework of higher dimensions, like the one describing probabilistic diffusion for images, where the number of dimensions is proportional to the number of pixels in the image, the Fokker-Planck equation takes the form

$$\frac{\partial p(x,t)}{\partial t} = -\sum_{i=1}^{d}\frac{\partial}{\partial x_i}[f_i(x,t)p(x,t)] + \frac{1}{2}\sum_{i=1}^{d}\frac{\partial}{\partial x_i}\left[\sum_{j=1}^{d}\frac{\partial}{\partial x_j}\left[\sum_{k=1}^{d}G_{ik}(x,t)G_{jk}(x,t)p(x,t)\right]\right]. \tag{3.15}$$

The core of the method to allow deterministic sampling described in [8] is in noting that the hard right part of the equation, making reference to the diffusion coefficient G, can be rewritten

$$\frac{1}{2}\sum_{i=1}^{d}\frac{\partial}{\partial x_i}\left[\sum_{j=1}^{d}\frac{\partial}{\partial x_j}\left[\sum_{k=1}^{d}G_{ik}(x,t)G_{jk}(x,t)p(x,t)\right]\right]$$

$$= \frac{1}{2}\sum_{i=1}^{d}\frac{\partial}{\partial x_i}\left[p(x,t)\nabla\cdot\left(G(x,t)G(x,t)^\top\right) + p(x,t)G(x,t)G(x,t)^\top\nabla_x \log p(x,t)\right] \tag{3.16}$$

Substituting this expression into 3.15 yields

$$\begin{aligned}\frac{\partial p(x,t)}{\partial t} &= -\sum_{i=1}^{d}\frac{\partial}{\partial x_i}([f_i(x,t)p(x,t)] \\ -\frac{1}{2}p(x,t)\left[\nabla\cdot\left(G(x,t)G(x,t)^\top\right) + G(x,t)G(x,t)^\top\nabla_x \log p(x,t)\right])\end{aligned} \tag{3.17}$$

$$\frac{\partial p(x,t)}{\partial t} = -\sum_{i=1}^{d}\frac{\partial}{\partial x_i}\left[\tilde{f}_i(x,t)p(x,t)\right], \tag{3.18}$$

with

$$\tilde{f}_i(x,t) = f(x,t) - \frac{1}{2}\left[\nabla\cdot\left(G(x,t)G(x,t)^\top\right) + G(x,t)G(x,t)^\top\nabla_x \log p(x,t)\right]. \tag{3.19}$$

Equation (3.18) is only the continuity equation for the probability flux[1]. Here, $\tilde{f}_i(x,t)$ denotes a modified drift term, incorporating both the deterministic force $f(x,t)$ and additional terms involving the gradient of the diffusion matrix $G(x,t)$ and the score function. This formulation allows us to interpret (3.18) as a new Fokker-Planck equation corresponding to a process with a deterministic force $\tilde{f}_i(x,t)$ and no stochastic force ($\tilde{G}(x,t) = 0$). In this scenario, the associated Langevin equation simplifies to:

$$dx = \tilde{f}_i(x,t)dt + \tilde{G}(x,t)dW = \tilde{f}_i(x,t)dt. \tag{3.20}$$

There is no stochasticity in this new parametrization. Starting from a stochastic process (3.5), we have found a *deterministic* process (3.20) that shares the same set of probability distributions $\{p(x,t)\}_{t=0}^{T}$.

Let us revisit the assumption of a drift coefficient in the diffusion process SDE being solely dependent on time, like in equation (3.1). Under this premise, where $\nabla\cdot G(t)G(t)^\top = 0$, the resultant deterministic process (3.20) assumes a simplified structure:

$$dx = \left(f(x,t) - \frac{1}{2}G(t)G(t)^\top\nabla_x \log p(x,t)\right)dt. \tag{3.21}$$

[1]For a more in-depth derivation, see [8], Appendix D.1

The remaining task involves numerically integrating this equation in reverse time for sampling purposes, succeeded by the discretization of the obtained equation. Particularly within the domain of Diffusion Probabilistic Models, the deterministic sampling equation [8, 10] is formulated as:

$$x_i = \left(2 - \sqrt{1 - \beta_{i+1}}\right) x_{i+1} + \frac{1}{2}\beta_{i+1} \nabla_x \log p(x, t). \tag{3.22}$$

By inspecting the Fokker-Planck equation's evolution of probability density over time, we transition from a stochastic process (3.5) to its deterministic counterpart (3.20). Notably, this deterministic process is described by the same sequence of probability distributions $\{p(x,t)\}_{t=0}^{T}$. Thus allowing us to sample images $x_0 \sim q(x_0)$ in a deterministic way.

Chapter 4

Results

4.1 2D and 3D model implementations

In this section, we present the results obtained from the application of diffusion probabilistic models in a two-dimensional (and then a three-dimensional) setting. The purpose of this demonstration is to provide a tangible illustration of the theoretical concepts and methodologies explained throughout the previous chapters of this thesis. Through a comprehensive analysis of the generated outcomes, we aim to validate the efficacy and applicability of diffusion generative models in capturing complex probability distributions and generating realistic data samples.

The presented results are derived from a meticulously designed two-dimensional diffusion model, made to encapsulate the essence of the diffusion processes and their associated probabilistic backgrounds. By leveraging this model, we explore various aspects of diffusion dynamics, including forward and backward diffusion processes, parameterization strategies, and sampling techniques.

Throughout this section, we analyze the obtained results, and additionally, we compare the generated samples with ground-truth data distributions, offering a comprehensive evaluation of the model's performance in faithfully representing underlying probability distributions.

Overall, the results presented here serve to give us a taste of the potential of diffusion probabilistic models as powerful tools for data generation and distribution modeling, paving the way for their application in domains we will be exploring later, like the aforementioned image generation.

4.1.1 2D Model Implementation

We begin our exploration by defining the initial distribution $q(x_0)$, which encapsulates the underlying probability distribution we aim to learn. In this demonstration, we simplify our initial distribution to a two-dimensional circular shape with a slight width, representing a circumference of particles. The choice of a circular distribution allows for a straightforward illustration of the diffusion processes and facilitates the visualization of the model's behavior. Within this distribution, we place a total of 100,000 particles, each representing a datapoint in our dataset.

Forward Process

The parameterization of the forward diffusion process is a critical aspect of our model implementation, as it dictates the manner in which the original data distribution is progressively transformed into a Gaussian distribution over time. Unlike the linear noise schedule utilized in previous studies [2], we adopt a cosine schedule for noise annealing, as proposed by [11]. This schedule has been demonstrated to induce a slower diffusion process, which in turn facilitates more effective and efficient learning by the model.

The cosine schedule is meticulously designed to modulate the noise level throughout the diffusion process, ensuring a gradual and controlled transformation of the data distribution. By slowing down the diffusion process, the model can better capture the underlying structure of the data distribution, leading to enhanced learning and more accurate generation of new data samples.

In our implementation, we employ a total of $T = 50$ timesteps for the diffusion process. This choice of timestep duration strikes a balance between computational efficiency and model performance, allowing for an adequate exploration of the diffusion dynamics while ensuring timely convergence of the learning process.

Backward Process

In the backward diffusion process, our objective is to revert the transformations applied during the forward diffusion process, thereby generating new data samples from Gaussian noise. As the theoretical framework established earlier, we employ a deep neural network ϵ_θ to predict the noise ϵ required to transition from the initial state x_0 to a given state x_t, as outlined in Equation 2.13.

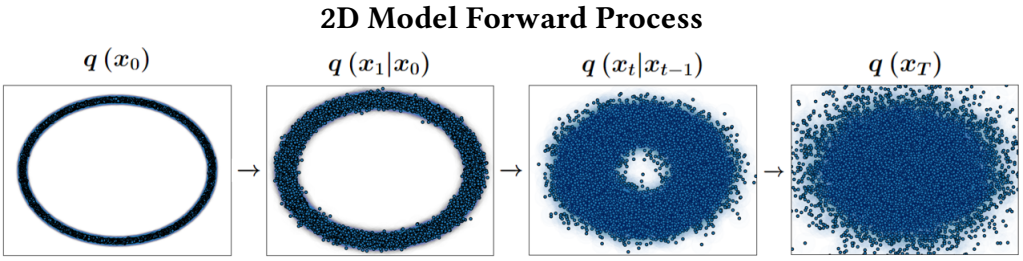


Figure 4.1: Depiction of the Forward Diffusion Process for the set of 100,000 particles. After a sufficient amount of time, the initial out-of-equilibrium distribution $q(x_0)$ transforms into a Gaussian distribution $q(x_T)$

To parameterize ϵ_θ, we opt for a deep neural network architecture designed to the specific requirements of our task. This neural network takes as input two numbers, representing the coordinates of the particle within the two-dimensional space. Subsequently, it outputs the x and y components of the noise ϵ.

The chosen neural network architecture comprises four hidden layers, each consisting of 64 neurons, resulting in a total of 256 neurons across all hidden layers. This architecture is designed to strike a balance between model complexity and computational efficiency, enabling the neural network to learn intricate mappings between input coordinates and noise components effectively.

By leveraging this parameterization approach, we aim to equip our model with the capability to accurately predict the noise required for backward diffusion, thereby enabling the generation of data samples from Gaussian noise. Next, we delve into the training methodology employed to optimize the parameters of ϵ_θ, making possible its effective integration into the backward diffusion process.

For the training of the neural network ϵ_θ, we employ a training regimen aimed at optimizing its parameters to accurately predict the noise required for backward diffusion. The training process spans a total of 100 epochs, ensuring sufficient iterations for the network to converge to an optimal solution.

During each epoch, training samples are fed (according to the training algorithm shown earlier) to the network in batches to facilitate efficient computation and parameter updates. Specifically, we utilize a batch size of 2048, enabling the neural network to process multiple data points simultaneously.

By iteratively exposing the neural network to training data and adjusting its parameters through backpropagation, we aim to minimize the discrepancy

between the predicted noise components and the true noise required for backward diffusion.

With the model successfully trained, we proceed to sample a total of 10,000 particles from the learned distribution $p_\theta(x_0)$. These samples provide valuable insights into the nature of the approximated distribution and allow us to assess the effectiveness of the model in capturing the underlying data distribution.

Upon visual inspection of the scatter plot depicting the 10,000 sampled particles (refer to Figure 4.2), we observe that the majority of particles are distributed in a manner closely resembling the initial distribution $q(x_0)$. However, upon closer examination, we also identify a small subset of particles that deviate from this distribution.

The presence of these outliers suggests potential areas where the model may exhibit limitations or areas for improvement. By analyzing these deviations, we can gain valuable insights into the model's performance and identify potential avenues for refinement or optimization.

Overall, the sampling results provide valuable validation of the model's efficacy in approximating the target distribution $q(x_0)$.

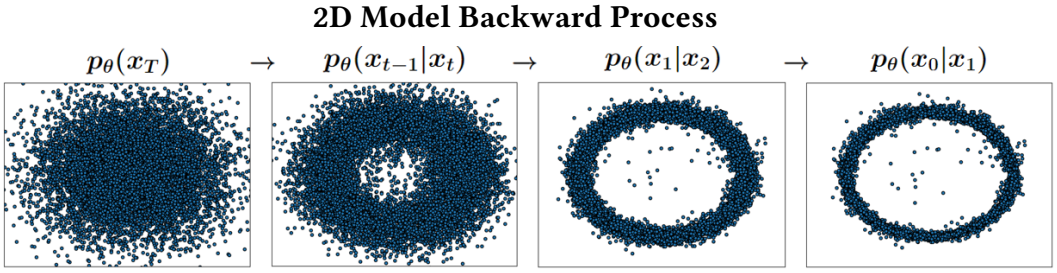


Figure 4.2: Depiction of the Backward Diffusion Process for a set of 10,000 particles. Their initial positions are sampled from the Gaussian distribution $x_T = q(x_T)$. After following the backward process, the majority of particles closely resemble the initial distribution $q(x_0)$, with some deviations observed in a small subset of particles.

4.1.2 3D Model Implementation

In this subsection, we present the results of extending our diffusion model to three dimensions. The hyperparameters used in this model largely remain consistent with those of the 2D model, with adjustments made to accommodate the

increased dimensionality. Notably, the architecture of the neural network is modified to handle three-dimensional inputs and outputs, and one additional hidden layer is introduced to enhance the model's capacity for capturing complex relationships within the data.

The neural network architecture for the 3D particle model features five hidden layers, each comprising 64 neurons. This increased depth enables the model to effectively capture intricate patterns and dependencies within the data, facilitating more accurate predictions of the diffusion process. Additionally, adjustments are made to the input and output layers to accommodate three-dimensional data representations, ensuring compatibility with the model's architecture.

To evaluate the performance of the 3D particle model, we first examine the initial distribution $q(x_0)$, representing the spatial distribution of particles at the start of the diffusion process. This distribution provides valuable insights into the spatial characteristics of the data and serves as a reference point for assessing the quality of the generated samples.

Next, we analyze the equilibrium distribution $q(x_T)$, which represents the spatial distribution of particles after the completion of the diffusion process. This distribution offers insights into the long-term behavior of the diffusion process and provides a benchmark for evaluating the fidelity of the generated samples.

Finally, we can visually compare the predicted distribution $p_\theta(x_0)$ generated by the trained model against the ground truth distribution $q(x_0)$.

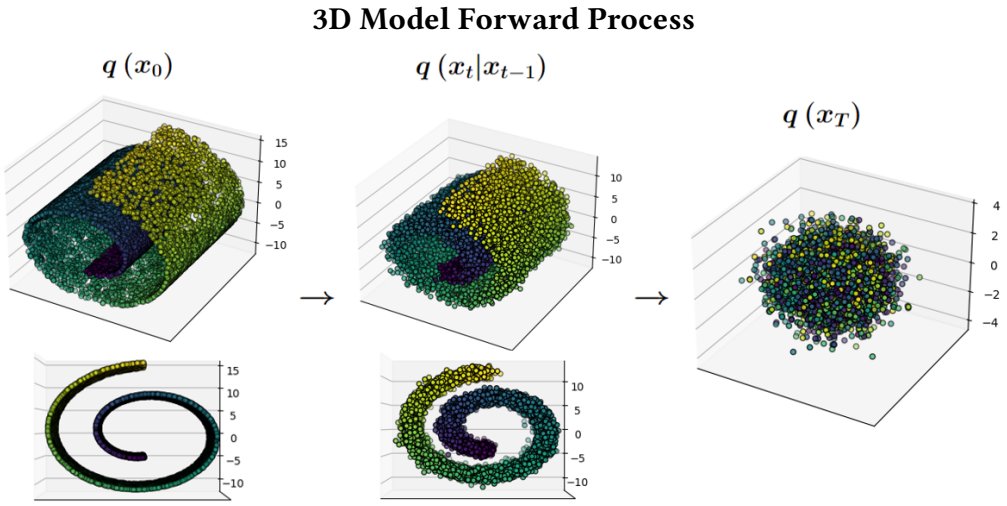


Figure 4.3: Depiction of the Forward Diffusion Process, now for a 3D distribution.

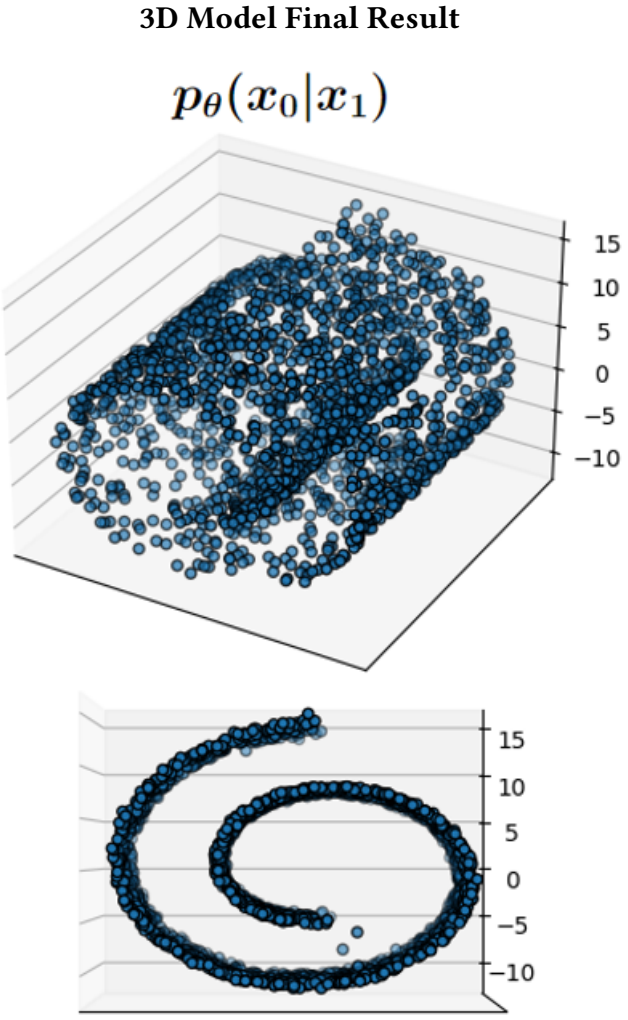


Figure 4.4: Scatter plot of the final learned $p_\theta(x_0)$ distribution. Once again, the majority of particles closely resemble the initial distribution $q(x_0)$.

4.2 Images Model Implementation

This implementation is based on the work described in the paper authored by Jo. et al [2] and Chapter 2 of this thesis. The implementation employs the U-Net architecture to model the Flowers102 dataset, which comprises images from 102 different flower categories. Although the dataset includes category labels, the model does not utilize them during training. Instead, it learns to generate images that combine attributes from all categories, resulting in a diverse and

generalized representation of the dataset. We resize the images to 64x64 pixels to ensure computational efficiency while preserving sufficient detail.

U-net Convolutional Network

The U-Net architecture, initially designed for biomedical image segmentation, is characterized by its symmetric contracting and expanding pathways, shaped like the letter "U". This design allows for the extraction of intricate features from images while preserving spatial information. The down-sampling path, akin to zooming out on an image, involves the use of convolutional layers followed by pooling layers to reduce spatial dimensions. The bottleneck, located at the center of the network, plays a crucial role in consolidating information from the down-sampling path while maintaining sufficient context for subsequent processing. The up-sampling path, akin to zooming in on an image, reconstructs the high-resolution output using transposed convolutional layers, also known as deconvolution layers, to gradually restore spatial dimensions while retaining relevant features.

In addition to the standard U-Net architecture, self-attention mechanisms are incorporated to enhance feature representation. Self-attention enables the model to focus on relevant regions of the image, akin to selectively sharpening certain aspects while blurring others. This attention mechanism aids in capturing long-range dependencies and improving the model's ability to generate coherent and realistic images.

The implementation incorporates a linear noise schedule for the diffusion process, as proposed in [2]. This schedule begins with an initial diffusion coefficient, $\beta_0 = 10^{-4}$, and linearly increases it to a final value, $\beta_T = 0.02$, over a specified number 300 of diffusion steps. This gradual increase in noise enables the model to progressively learn to denoise images and generate samples that closely resemble the underlying data distribution.

The training process is facilitated by the stochastic gradient descent algorithm, specifically the AdamW optimizer available in the pytorch library, which adjusts the model's parameters based on gradients computed from mini-batches of 8 images from the training data.

The learning rate at which the optimizer updates the model's parameteras is of 3×10^{-4} and was chosen based on empirical testing to achieve stable convergence. The number of passes through the entire dataset during training, or in this case the number of epochs, is 30. The training process is conducted on

a personal NVIDIA RTX 3060 GPU with 12GB of memory, leveraging its parallel processing capabilities to accelerate computation. The final results for both stochastic and deterministic sampling processes can be seen in the subsequent figures:

Stochastic Sampling Results

Figure 4.5: Results of the generation of 64 images for the stochastic sampling method (2.25).

Stochastic Sampling Timeline

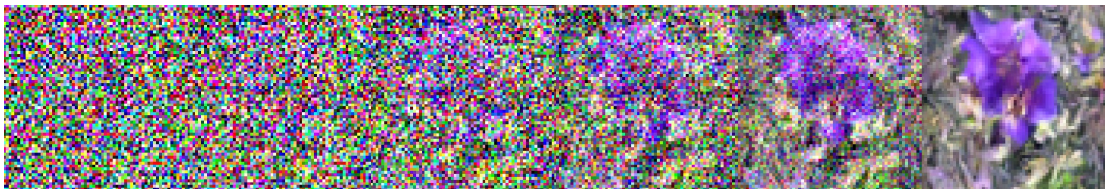


Figure 4.6: Timeline of the generation process of an image with the stochastic sampling method.

Deterministic Sampling Results

Figure 4.7: Results of generating 64 images using the deterministic sampling method (3.22). While this sampling method does not perform as effectively as its stochastic counterpart, resulting in crisper images, it still produces coherent outputs.

Deterministic Sampling Timeline

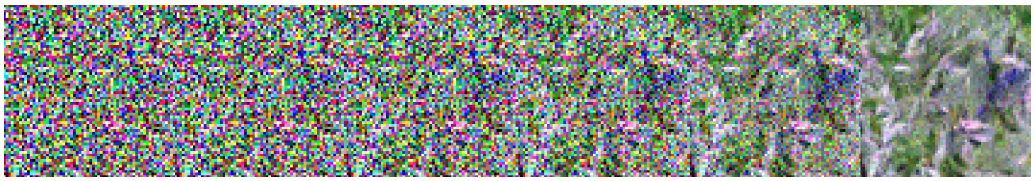


Figure 4.8: Timeline of the generation process of an image with the deterministic sampling method.

Chapter 5

Conclusions

In this thesis, we have thoroughly explored Diffusion Probabilistic Models (DPM) and their application in generating synthetic images using Langevin dynamics. DPMs, inspired by principles of statistical physics, allow for the modeling of diffusion processes to capture intrinsic patterns in complex datasets. The research focused on comparing probabilistic diffusion models with statistical physics, highlighting how the diffusion process can be effectively explained using a specialized version of the Langevin equation. This approach was extended to continuous-time diffusion processes, providing a deeper understanding of the physical principles underlying DPMs.

During the development of this research, a detailed analysis was conducted on how DPMs utilize the concept of diffusion to learn the probability distribution of a dataset. In particular, the relationship between the diffusion process in DPMs and the Brownian motion described by the Langevin equation was studied. It was found that the data diffusion process can be modeled as the movement of particles in a physical system, where particles undergo small Gaussian perturbations that, over time, transform the original data distribution into a pure Gaussian noise distribution. This analogy allowed for the establishment of a solid theoretical framework that not only validates the use of diffusion models in machine learning but also provides a basis for understanding how these models capture and generate synthetic data that reflects the underlying structure of the original datasets.

Chapter 3 significantly expands the theoretical framework of Diffusion Probabilistic Models by introducing a generalized formulation through Stochastic Differential Equations (SDE). This section delves into the idea that the diffusion process can be described in continuous time, offering a more flexible and detailed understanding of these models' behavior. The Fokker-Planck equation is used

to connect the stochastic formulation with a deterministic equivalent, thereby allowing the use of deterministic methods to sample new images while maintaining the evolution of the probability density.

The concept of the score function, essential for reversing the diffusion process, is also explored, along with how it can be parameterized using deep neural networks to predict the necessary perturbations. This generalized framework not only broadens the applicability of DPMs to a wider range of problems but also enhances the understanding of synthetic data sampling, underscoring its potential for future applications in data generation, physical system simulation, and probabilistic modeling.

The implementation of diffusion models in two-dimensional and three-dimensional systems allowed for the learning of various out-of-equilibrium particle position distributions. This approach not only provided empirical validation of the efficacy of DPMs but also demonstrated these models' ability to learn and generate complex distributions. Applying these models to image generation, a diffusion model was developed to create 64x64 pixel images of flowers. The results obtained showed that DPMs could generate high-quality synthetic data that faithfully reflects the structure of the original datasets.

In terms of major contributions, this work demonstrated that the data diffusion process could be interpreted using the Langevin equation, establishing a clear parallel between the behavior of particles in physical systems and the evolution of data distributions in probabilistic space. This comparison not only validates the use of diffusion models in machine learning but also provides a solid theoretical foundation for future studies and applications. The implementation of diffusion models in both two-dimensional and three-dimensional physical systems allowed for the learning of out-of-equilibrium particle position distributions, providing empirical validation for the efficacy of DPMs. These models' ability to learn and generate complex distributions underscores their potential in various scientific and technological applications.

In conclusion, this thesis has significantly advanced the understanding and application of Diffusion Probabilistic Models in generating synthetic data. By integrating concepts from statistical physics with advanced machine learning techniques, we have demonstrated that DPMs are not only theoretically sound but also practically effective in generating high-quality images. These findings establish a solid foundation for future research and applications, opening new avenues for developing more advanced and efficient generative models. The continuous interaction between theoretical physics and machine learning promises to

provide valuable insights and powerful tools for addressing complex challenges in science and technology.

References

Articles, proceedings and theses

1 J. Sohl-Dickstein and E. A. Weiss:
'Deep unsupervised learning using nonequilibrium thermodynamics.',
Cornell University (2015).

2 J. Ho, A. J. Jain and P. Abbeel:
'Denoising diffusion probabilistic models',
Cornell University (2020).

3 J. N. Yan, J. Gu and A. M. Rush:
Diffusion models without attention,
2023.

4 T. Yang, Y. Wang, Y. Lv and N. Zheng:
DISDIFF: Unsupervised Disentanglement of Diffusion Probabilistic models,
2023.

5 D. Walter:
'Fokker-planck and langevin equation',
Institut für Theoretische Physik (2021).

6 R. Pathria and P. Beale:
'Statistical Mechanics, 3rd edn.',
Contemporary physics **52**, 619–620 (2011).

7 Y. Song and S. Ermon:
Generative modeling by estimating gradients of the data distribution,
2020.

8 J. S.-D. Yang Song:
'Score-based generative modeling through stochastic differential equations.',
International Conference on Learning Representations. (2021).

9 B. D. Anderson:
'Reverse-time diffusion equation models',
Stochastic Processes and their Applications **12**, 313–326 (1982).

10 C. M. Jiaming Song and S. Ermon:
'Denoising diffusion implicit models',
Cornell University (2020).

11 P. D. Alex Nichol:
'Improved denoising diffusion probabilistic models',
Cornell University (2021).