William Dalheim

Generative AI through Latent Modeling: The Theoretical Foundations of Diffusion Models

Master's thesis in Computer Science
Supervisor: Inga Strümke
Co-supervisor: Helge Langseth
June 2023

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

William Dalheim

Generative AI through Latent Modeling: The Theoretical Foundations of Diffusion Models

Master's thesis in Computer Science
Supervisor: Inga Strümke
Co-supervisor: Helge Langseth
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

In recent years, Diffusion Models (DMs) have undergone rapid advancements, surpassing renowned deep generative models on image synthesis, and demonstrating that photorealism is achievable with text-to-image applications. Inspired by thermodynamics, DMs learn to gradually recover data that has undergone diffusion. Starting from pure noise, coherent data can be synthesized.

The impressive performance and perceived novelty of DMs have led to several contributions in the field. Found to be the most influential work is the Denoising Diffusion Probabilistic Model (DDPM) framework, serving as the basis for many improvements implemented in today's large-scale DMs. Second to that, the Denoising Diffusion Implicit Model (DDIM) is a reformulation of the sampling procedure that allows for deterministic and more efficient generation.

As with other deep generative models, DMs lack interpretability on the rationale that affects data synthesis. However, the mapping between noise and the data distribution learned by DDIM has been found to align with the optimal transport map. In essence, changes to individual dimensions on the path from pure noise to clean data are minimized by the model, giving a certain degree of predictability in the data generation process.

Experiments on both low- and high-dimensional data show that optimal transport serves as a valuable interpretation of the behavior of DMs, enabling control of the sampling process. This allows for manipulating certain features through a scheme this thesis terms *latent manipulation*. In one experiment, groups of latent dimensions are translated toward specific color intensities that coincide with the desired feature, increasing the likelihood for it to appear. Through another experiment, latent information is copied from one sample to another for a targeted feature to emerge. *Latent manipulation* allows local and global contexts to interweave and does not require retraining, making it a more flexible alternative to existing editing methods, such as inpainting.

Sammendrag

I løpet av de siste årene har diffusjonsmodeller (DMer) gjennomgått en rekke forbedringer, som har ført til bedre resultater enn andre anerkjente dype generative modeller innen bildesyntese. Modellene har også demonstrert at syntetisert fotorealisme er oppnåelig gjennom tekst-til-bilde anvendelser. DMer tar inspirasjon fra termodynamikk, ved at de lærer seg å gradvis gjenopprette data som er påvirket av støy. Når denne prosessen startes fra komplett støy, kan realistiske data bli syntetisert.

De imponerende resultatene har ført til flere bidrag til feltet. “Denoising Diffusion Probabilistic Model" (DDPM) anses å være det mest innflytelsesrike rammeverket, og danner grunnlaget for mange av forbedringene man ser i dagens storskala DMer. Videre er “Denoising Diffusion Implicit Model" (DDIM) en omformulering av genereringsprosessen, som tillater deterministisk og mer effektiv generering av data.

I likhet med andre dype generative modeller, er det en mangel på verktøy til bruk i tolkning av DMer for å avdekke logikken som inngår i generering av data. Imidlertid har det blitt oppdaget at DDIM lærer en transformasjon mellom støy og data som samsvarer med den optimale transportruten. I praksis betyr det at modellen forsøker å minimere endringene i de individuelle dimensjonene på veien fra støy til data, noe som gir en viss forutsigbarhet i genereringsprosessen.

Eksperimenter på både lav- og høydimensjonale data antyder at optimal transport gir verdifull innsikt i hvordan DMer fungerer, og tillater bedre kontroll over genereringsprosessen. Dette muliggjør manipulasjon av bestemte egenskaper ved dataen gjennom en metode denne masteroppgaven kaller “latent manipulasjon". I et eksperiment blir grupper av latente dimensjoner flyttet mot bestemte fargeintensiteter som samsvarer med den ønskede egenskapen, for å øke sannsynligheten for at den oppstår. I et annet eksperiment kopieres latent informasjon fra et datapunkt til et annet for å få frem en bestemt egenskap. “Latent manipulasjon" muligjør en flyt mellom lokale og globale kontekster, og krever ikke finjustering eller trening av en ny modell. Dette gjør det til et fleksibelt alternativ til eksisterende redigeringsmetoder, som for eksempel “inpainting".

Preface

This thesis is written at the Department of Computer Science at the Norwegian University of Science and Technology.

I show my gratitude to my supervisor Inga Strümke and my co-supervisor Helge Langseth. Throughout the demanding process of working on this thesis, their invaluable guidance has been a great motivating factor. Thank you for all the interesting discussions.

To my amazing parents, thank you for patiently enduring my enthusiastic ramblings, and comforting me during moments of frustration.

William Dalheim
Trondheim, 12th June 2023

Contents

Abstract i

Sammendrag ii

Preface iii

List of Figures viii

List of Tables ix

Acronyms xi

Notation xiii

1 Introduction 1
- 1.1 Background and Motivation . . . 1
- 1.2 Goals and Research Questions . . . 3
- 1.3 Thesis Structure . . . 4

2 Background Theory 7
- 2.1 Random variables and distributions . . . 7
- 2.2 Gaussian Distribution . . . 9
- 2.3 Likelihood . . . 11
- 2.4 KL Divergence . . . 12
- 2.5 Markov Chains . . . 13
- 2.6 Deep Learning . . . 14

3 State of the Art 15
- 3.1 Deep Generative Modeling . . . 15
  - 3.1.1 Learning latent representations . . . 16
  - 3.1.2 The Autoencoder: A naïve Deep Generative Model . . . 17
- 3.2 Variational inference . . . 18
  - 3.2.1 The Evidence Lower Bound . . . 19
  - 3.2.2 The Variational Autoencoder . . . 19
- 3.3 Diffusion Models . . . 22
- 3.4 Denoising Diffusion Probabilistic Models . . . 23
  - 3.4.1 Forward process . . . 24
  - 3.4.2 Reverse process . . . 26

3.4.3 The variance schedule 26
3.4.4 Determining the objective 27
3.4.5 Training 35
3.4.6 Sampling 35
3.5 Denoising Diffusion Implicit Models 38
3.5.1 Alternative forward processes 38
3.5.2 Sampling 40
3.6 Optimal Transport 42

4 Architecture and Implementation 45
4.1 Architecture 45
4.2 Data sets 47
4.3 Models 47

5 Experiments and Results 49
5.1 Experimental Plan 49
5.2 Experimental Setup 49
5.3 Experiments on Optimal Transport: Low dimensionality 50
Experiment group L.1 Minimization of Euclidean distance 50
Experiment group L.2 Density transportation 52
5.4 Experiments on Optimal Transport: High dimensionality 57
Experiment group H.1 Identical initial latents 57
Experiment group H.2 Latent manipulation 60

6 Discussion and Conclusion 67
6.1 Discussion 67
6.2 Contributions 69
6.3 Future Work 69

Bibliography 73

Appendices 79

A Marginalizing irrelevant variables 81

B DDIM ELBO derivation 83

C DDPM sampling with no noise 85

D Additional mappings for Experiment L.2c 87

E Additional samples for Experiment H.2a 89

List of Figures

1.1 Diffusion process 1
1.2 Samples from Midjourney 2

2.1 Gaussian PDF 10
2.2 KL divergence between two Gaussians 13
2.3 Example Markov chain 13
2.4 Multilayer Perceptron 14

3.1 A simple representation learning model 16
3.2 A simple deep generative model 17
3.3 Reparameterization technique 20
3.4 Simple VAE architecture 22
3.5 Diffusion Model Markov chain 23
3.6 Components of the variance schedule 27
3.7 DDPM subsamples from the generative chain 37
3.8 One-dimensional sampling using a fixed latent 37
3.9 Predicting data from latents 38
3.10 Deterministic sampling trajectories 41
3.11 Optimal transport in one dimension 42
3.12 Optimal transport maps 44

4.1 The U-Net architecture 46

5.1 Effect of number of sampling steps on DDIM 50
5.2 Shortest paths found in symmetric and asymmetric mixtures of Gaussians 51
5.3 DDIM transporting densities between the prior and a shifted prior 53
5.4 DDIM transporting densities between the prior and a symmetric mixture of Gaussians 54
5.5 DDIM transporting densities from the right half of the prior 55
5.6 Extending CDF-mapping to higher dimensions 56
5.7 DDIM mapping orthogonal subsets of the prior to the data distribution. . 57
5.8 Samples from fixed latents by two separately trained models on the same data set 58
5.9 Samples from fixed latents by two separately trained models on different data sets 59
5.10 Modifying the intensity of latents 61
5.11 Modifying the intensity of latents along each color channel 62

5.12 Decreasing the intensity of specific latent dimensions 63
5.13 Increasing the intensity of specific latent dimensions 64
5.14 Transferring features between two samples 65
5.15 Additional examples of transferring features between samples 65

C.1 DDPM sampling with no noise 85

D.1 Additional CDF and DDIM mappings 87

E.1 Additional samples for intensity modification experiment 89

List of Tables

4.1 Training hyperparameters for models trained on high-dimensional data. . . 48
4.2 Training hyperparameters for models trained on low-dimensional data. . . 48
4.3 Variance schedule parameters. 48
4.4 Usage of models in experiments and figures. 48

Acronyms

AI Artificial Intelligence.

CDF Cumulative Distribution Function.

DDIM Denoising Diffusion Implicit Model.

DDPM Denoising Diffusion Probabilistic Model.

DM Diffusion Model.

EDF Empirical Distribution Function.

ELBO Evidence Lower Bound.

GAN Generative Adversarial Network.

ML Machine Learning.

NCSN Noise Conditional Score Network.

PDF Probability Density Function.

ReLU Rectified Linear Unit.

SOTA State of the Art.

VAE Variational AutoEncoder.

Notation

x	scalar
$\boldsymbol{x}$	vector/matrix/tensor
f	function
$\mathcal{X}$	data space
$\mathcal{Z}$	latent space
$\mathcal{Y}$	output space
$\mathcal{D}$	data set
$\boldsymbol{x}^{(i)}$	i'th instance of a data set
x	scalar random variable
$\mathbf{x}$	vector-valued random variable
Ω	sample space
$p(\mathrm{x}),\ p(\mathbf{x})$	probability distribution over a random variable
$\mathbf{x} \sim p$	random variable follows the distribution p
$p_{\text{data}}(\mathbf{x}),\ q(\mathbf{x}_0)$	the underlying distribution that a data set follows
$\mathbb{E}_{p(\mathbf{x})}[f]$	expectation of f with respect to $p(\mathbf{x})$
$p(x),\ p(\boldsymbol{x})$	probability density function
$D_{\mathrm{KL}}(p \mid\mid q)$	KL divergence between distributions p and q
$\boldsymbol{\theta}$	parameters
$f_{\boldsymbol{\theta}}$	function parameterized by $\boldsymbol{\theta}$
$p_{\boldsymbol{\theta}}$	probability distribution parameterized by $\boldsymbol{\theta}$

Chapter 1

Introduction

Diffusion Models (DMs) have over the past few years demonstrated its superiority as a state of the art (SOTA) deep generative model, surpassing the performance of generative adversarial networks (GANs) on image synthesis (Dhariwal and Nichol, 2021). This novel and flexible generative model has garnered significant attention in Machine Learning (ML) research, as well as in the media. Section 1.1 aims to provide an introduction to DMs, emphasizing the unique attributes that set them apart from other generative models. Following that, section 1.2 presents the goals and research questions that serve as the foundation for the research conducted in this thesis towards learning about and understanding DMs. Lastly, an overview of the thesis structure is given in section 1.3.

1.1 Background and Motivation

From thermodynamics, diffusion refers to the process where particles flow from an area of high concentration to an area of lower concentration until they reach an equilibrium (Cengel and Boles, 2005). The particles start at a state where they form a distinct structure. Over time, it is slowly broken down into a more uniform and dispersed configuration where entropy is increased.

Figure 1.1 demonstrates the dynamics of a probability distribution (left panel) that undergoes diffusion. This distribution is analogous to groups of highly concentrated particles, as it has a complex structure. As time passes, the complexity of the distribution vanishes. Particles that were once separated by low-density regions, become intermixed

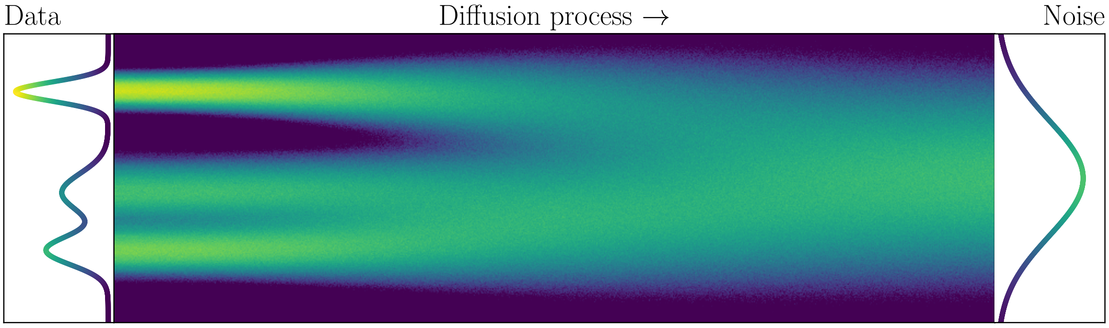


Figure 1.1: A diffusion process depicted on a complex probability distribution.

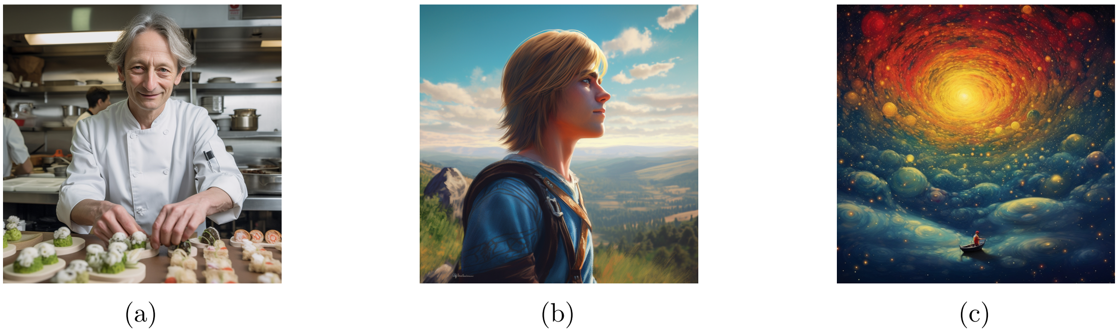


Figure 1.2: Samples from Midjourney. **(a)** “Geoffrey Hinton as a sushi chef". **(b)** “Link from The Legend of Zelda, Breath of the Wild, photorealistic, hyperdetailed, landscape, vibrant, sunny". **(c)** “The Gaussian Galaxy".

(middle panel). Towards the end, a simpler distribution (right panel) is obtained, representing the equilibrium. What was once clean data has become pure noise.

DMs learn to reverse the diffusion process (Sohl-Dickstein et al., 2015), allowing data to be recovered from noise corruption. The middle panel of fig. 1.1 depicts the intermediate states a one-dimensional DM learns to model. When sufficiently trained, the DM can accurately predict the previous state of any partially diffused data point. To synthesize new data that fits with the data distribution, the model starts from pure noise and iteratively *denoises* it until realistic data emerge.

The excellence of DMs has best been demonstrated in text-conditional image generation. Due to their helpfulness in creative workflows such as image manipulation and asset generation, a multitude of large-scale models has become available to the public as commercial services. At the time of writing, Midjourney[1] represents the pinnacle of DMs, capable of producing images nearly indistinguishable from reality. Figure 1.2 shows three samples, demonstrating its ability to synthesize well-known concepts with flexibility in terms of style.

As the performance of deep generative models improves, the necessity to understand their reasoning grows larger. However, being part of unsupervised learning, the unavailability of labels in the training data makes them more challenging to interpret than supervised ones. Renowned deep generative models such as variational autoencoders (VAEs) and GANs are designed to learn compressed encodings of the training data, thereby capturing the underlying independent data characteristics in the latent space. These are called generative factors. Ideally, the latent dimensions become disentangled, such that each is in control of its own independent generative factor (Bengio et al., 2014). While forcing the model to generalize over the training data, it has the added benefit of providing an interpretative scheme. Studying how the output changes when adjusting individual latent dimensions gives insight into the rationale that data generation undergoes.

The notion of latents for DMs differs from that of VAEs and GANs. By convention,

[1] Link to home-page: midjourney.com

the latent space refers to the space where fully corrupted data reside, which is not inside any intermediate layers of the neural network (Ho et al., 2020; Song et al., 2022). This is an effect of the iterative approach that goes into generating data. Consequently, the latent space has the same number of dimensions as the data space. When dealing with images of high resolutions, the pursuit of disentangled factors loses its significance as the model is discouraged from learning dimension-independent encodings. DMs must therefore be treated differently.

Some recent efforts have been made to understand DMs and interpret the latent space to gain control of the generation process. Yang et al. (2023b) aim to construct a DM-compatible disentanglement framework that enables control over independent generative factors. Kwon et al. (2022b) adjust the intermediate network activations of a DM to impel it towards generating specific features. The research by Tang et al. (2022) focuses on analyzing the open-sourced text-conditional model known as Stable Diffusion (Rombach et al., 2022). They study the relationship between the text encoder and the image-generating layers to create pixel-based heat maps for each token in the text condition. Their results bring clarity to how the model distributes conceptual knowledge from the text prompt into the output image.

Each of the aforementioned approaches has one disadvantage; they either rely on training new models on top of the DM to achieve control, or on extensions of the minimal unconditional DM. For the former, they do not explain its inherent behavior and relationship with the latent space. Arguably, if they were to be used for interpretations, the supplementing models that are trained by Yang et al. (2023b) and Kwon et al. (2022b) would also be in need of interpretation. For the latter, the core mechanisms of a DM are altered.

On the other hand, Khrulkov et al. (2022) present a noteworthy observation on how the transformation from noise to data appears to coincide with the optimal transport map. Furthermore, their studies do not rely on extensions of basal DMs, except for one that enables deterministic sampling. Nevertheless, this extension has become widely accepted as a fundamental aspect of DMs. In addition to going deep into the theory of DMs, this thesis also aims to build upon the findings of Khrulkov et al. (2022) to uncover suitable interpretations.

1.2 Goals and Research Questions

Guiding the research for this thesis are two goals and three research questions.

Goal 1 *Provide an accessible and thorough survey of the theoretical foundations surrounding DMs.*

DMs are relatively new and currently receiving a substantial amount of attention in the ML field. Many researchers strive to find generalizations and improvements, some potentially overstating their contributions to the theoretical foundation. Moreover, the research domain is possibly influenced by commercial entities pursuing improvements tailored for entertainment purposes.

Research question 1 *What scholarly works contribute to defining the theoretical foundations of DMs?*

Due to the rapid advancement and inherent complexity, there are likely numerous separate contributions that have led to the perceived foundation. Some may offer formulations that bring out desirable properties, while others may establish mathematical frameworks.

Gaining knowledge about the foundational theory has the potential to widen the possibilities of finding suitable interpretations. This thesis not only aims to learn about the theory, but also what mechanisms drive the behavior of a DM.

Goal 2 *Uncover meaningful interpretations of DMs and their latent space.*

As mentioned in section 1.1, the iterative nature of the generation process gives rise to a latent space independent of any intermediate layers in the network architecture. Furthermore, typical DMs architectures do not enforce dimensionality reductions comparable to that of VAEs and GANs. Methods such as disentanglement analysis are therefore incompatible, calling the need for new techniques.

Research question 2 *Can insight be gained regarding the behavior of DMs when studied from the perspective of optimal transport?*

Intuition on optimal transport arises from observing its practical applications in low dimensions, such as the distribution of goods from factories to retail stores. In higher dimensions, the theory becomes more abstract and creating compact visual representations is a challenge. Therefore, experiments designed from the perspective of optimal transport must be conducted to reveal behavioral elements of a DM, with the focus of constructing a bridge allowing the flow of low-dimensional intuition to higher dimensions.

Research question 3 *Can optimal transport be utilized to encourage the generation of specific features?*

Knowledge attained on the relationship between optimal transport and the generative process of a DM may be leveraged to gain control of features that appear in samples. Such a control scheme can potentially aid in designing interpretations that align with disentanglement in VAEs and GANs.

1.3 Thesis Structure

Chapter 1 is the introduction, where research motivation and goals are presented. Following is chapter 2, background theory, where mostly necessary elements from probability theory are provided. The SOTA surrounding DMs and relevant preliminaries are presented in chapter 3. Details on implementation and architecture are given in chapter 4, highlighting relevant models trained for use in this thesis. These are put to use in the experiments presented in chapter 5, with discussions of the results. Chapter 6 concludes the thesis, mentioning some valuable directions for future work.

Some figures presented throughout this thesis are made using results from trained DMs. Table 4.4 in section 4.3 provides an overview of where models are used. The code accompanying this thesis is available at `github.com/willdalh/diffusion-ot`.

Chapter 2

Background Theory

The purpose of this chapter is to establish the notation and present the mathematical theory that DMs are based on. Most of the theory revolves around probability theory, and its presentation is primarily based on the works of Bishop (2006), Wasserman (2004) and Goodfellow et al. (2016). Section 2.6 is based on Goodfellow et al. (2016). Section 2.2 is an extensive rework of a similar section from Dalheim (2022).

2.1 Random variables and distributions

Probability theory deals with random variables, for example, x, which describe uncertain quantities that take on different values from a set Ω, called the sample space. This thesis focuses on continuous distributions, describing random variables whose sample space is an uncountable set of real numbers. The height of a random subject in a population of humans is an example of a continuous random variable, as the measured quantity falls on the real number line $\mathbb{R}$. The tendency of x to take on specific values is defined by a probability distribution $p(\mathrm{x})$, described through the relationship $\mathrm{x} \sim p(\mathrm{x})$. A specific realization of the random variable is a value $x \in \Omega$. In the case where the measure is a vector quantity, it is type-faced in bold, such that $\mathbf{x} \sim p(\mathbf{x})$ is a multivariate random variable following the distribution $p(\mathbf{x})$. $\mathbf{x} = (\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n)$ is then a collection of univariate random variables with the possibility of them being correlated.

Throughout this thesis, the word distribution will be used repeatedly. In some cases, the random variable x will not be stated as it is implied by the presence of a distribution. For ease of notation, often only the letter denoting the distribution will be stated, for example, p. When several distributions are presented by their letters only, it is assumed that they define a distribution of the same random variable. Additionally, vectors generalize collections of scalars and give no restriction on the number of elements. This implies that a collection can be of one element, representing a single scalar. Some of the theory will be presented with scalar values x when necessary, while in most cases it will be generalized to vectors $\boldsymbol{x}$.

The relationship between two random variables $\mathbf{x}$ and $\mathbf{y}$ is defined through the two fundamental rules of probability theory. The first is the sum rule, whose name originates from discrete random variables where the integral is replaced by a sum. It is given by

$$p(\mathbf{x}) = \int_{\Omega_{\mathbf{y}}} p(\mathbf{x}, \mathbf{y}) \; d\mathbf{y}, \tag{2.1}$$

where $p(\mathbf{x}, \mathbf{y})$ is the joint distribution defining the probability for pairs of values occurring. The process of integrating the joint distribution over a set of random variables to get a distribution with less variables is called *marginalization*, hence the resulting distribution $p(\mathbf{x})$ is often referred to as the marginal distribution over $\mathbf{x}$. The second rule is the product rule,

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x}), \tag{2.2}$$

where $p(\mathbf{y}|\mathbf{x})$ is the conditional probability of $\mathbf{y}$ given $\mathbf{x}$. It defines how $\mathbf{y}$ behaves given observations of $\mathbf{x}$. The product rule is symmetric, meaning that $p(\mathbf{x}, \mathbf{y})$ can also be calculated through $p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$. Equating the two expressions for the joint distribution and solving for either of the conditionals give rise to Bayes' rule

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{x})} \Leftrightarrow \; p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})}, \tag{2.3}$$

which has an important role in variational inference, helping to infer properties about unknown variables with knowledge about the observed ones. Therefore, Bayes' rule will return in section 3.2 where each distribution part of the equation is labeled for that context.

The probability density function (PDF), here denoted $p(\boldsymbol{x})$, is a continuous function defined over the sample space of a continuous random variable. For it to be a true probability density, it must satisfy

$$p(\boldsymbol{x}) \geq 0, \tag{2.4}$$

$$\int_{\Omega} p(\boldsymbol{x}) \; d\boldsymbol{x} = 1, \tag{2.5}$$

being the requirements of non-negativity and integrability over the whole sample space that sums up to a total density of 1. Instead of talking about probabilities of specific observations $\mathbf{x}$ occurring, it is more relevant to state the probability of the quantity falling inside some subspace $\mathcal{A} \subseteq \Omega$. The probability is given by the definite integral

$$p(\boldsymbol{x} \in \mathcal{A}) = \int_{\mathcal{A}} p(\boldsymbol{x}) \; d\boldsymbol{x}.$$

The probability that the random variable takes on a value on the interval $\langle -\boldsymbol{\infty}, \boldsymbol{a}]$ is given by

$$F(\boldsymbol{a}) = p(\mathbf{x} \leq \boldsymbol{a}) = \int_{-\boldsymbol{\infty}}^{\boldsymbol{a}} p(\boldsymbol{x}) d\boldsymbol{x},$$

where the cumulative distribution function (CDF) F of $\mathbf{x}$ is evaluated at the point $\boldsymbol{a}$. The CDFs relevant for this thesis are monotonically increasing and continuous, implying the existence of an inverse. For univariate distributions, the inverse CDF, denoted $F^{-1}(q)$, is called the quantile function defined for $q \in \langle 0, 1\rangle$ mapping cumulative probabilities back to the sample space, such that $F(x) = q$.

The expectation of a random variable states the value that it is centered around. Consider a data set $\mathcal{D} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{n}$ of n samples from a random variable $\mathbf{x}$. An assumption often made about the data set is that its elements are independently sampled and identically distributed, a property often abbreviated as i.i.d. This means that each $\boldsymbol{x}^{(i)}$ was sampled from the same distribution $p(\mathbf{x})$, and in isolation from the others. From the data set, the sample average of the random variable is found through

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}^{(i)}.$$

On the limit $n \to \infty$, the sample average converges to the expected value. In the presence of the PDF $p(\boldsymbol{x})$ for $\mathbf{x}$, the true mean is calculated by weighting all samples with their corresponding probabilities and integrating over the sample space, giving

$$\mathbb{E}[\mathbf{x}] = \int_{\Omega} p(\boldsymbol{x})\ \boldsymbol{x}\ d\boldsymbol{x}.$$

The expected value can also be computed for transformations of the random variable, defined through a function $f(\boldsymbol{x})$ such that

$$\mathbb{E}_{p(\mathbf{x})}[f] = \int_{\Omega} p(\boldsymbol{x}) f(\boldsymbol{x})\ d\boldsymbol{x}, \tag{2.6}$$

where the subscript of $\mathbb{E}[\cdot]$ denotes which distribution is used to weight the function. This subscript also specifies where samples $\boldsymbol{x}$ originate from, proving its usefulness when encountering expectations taken over functions consisting of multiple random variables.

A conditional expectation is taken with respect to a conditional distribution, expressed as

$$\mathbb{E}_{p(\mathbf{x}|\boldsymbol{y})}[f|\boldsymbol{y}] = \int_{\Omega} p(\boldsymbol{x}|\boldsymbol{y}) f(\boldsymbol{x})\ d\boldsymbol{x}, \tag{2.7}$$

where $\boldsymbol{y}$ is a realization of one or multiple random variables.

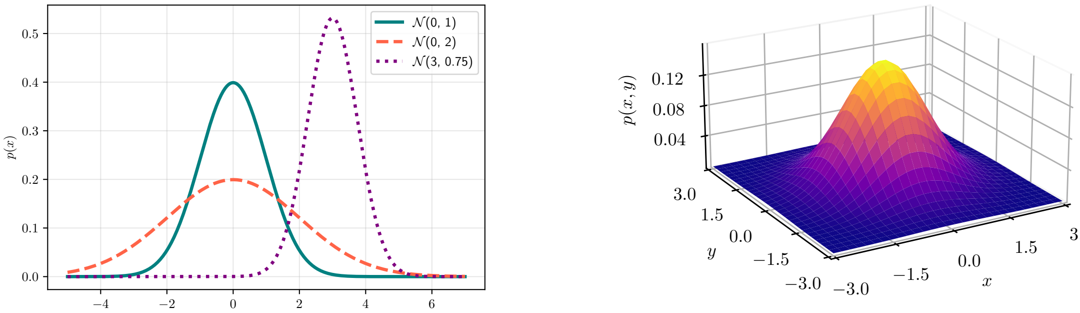
The variance $\mathrm{Var}_{p(\mathbf{x})}[f(\boldsymbol{x})]$ is the expected, or average, squared deviation of a function from the mean of the same function. Formally, it is given by

$$\mathrm{Var}_{p(\mathbf{x})}[f(\boldsymbol{x})] = \mathbb{E}_{p(\mathbf{x})}\left[(f(\mathbf{x}) - \mathbb{E}_{p(\mathbf{x})}[f(\mathbf{x})])^2\right]. \tag{2.8}$$

When $f(\boldsymbol{x}) = \boldsymbol{x}$, the quantity is the variance of the random variable $\mathbf{x}$. The variance and standard deviation are useful as they provide information about how uncertain a distribution is.

2.2 Gaussian Distribution

The Gaussian distribution, also called the normal distribution, plays an important role in generative modeling, as it is often used as prior knowledge on some complex structures to be modeled. A univariate Gaussian is defined by two scalar parameters, the mean and the variance, denoted by μ and σ^2 respectively. When $\mu = 0$ and $\sigma^2 = 1$, the Gaussian is



(a)

(b)

Figure 2.1: Gaussian PDFs. **(a)** Three univariate Gaussians with different parameters. **(b)** Standard bivariate Gaussian.

origin-centered and known as a standard normal distribution or standard Gaussian. The PDF of the Gaussian is given by

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}, \tag{2.9}$$

plotted for three different sets of parameters in fig. 2.1a. The PDF of a Gaussian is commonly written as $p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$, and will be used when a compact representation is necessary.

A random variable distributed according to a Gaussian distribution is written as $\mathrm{x} \sim \mathcal{N}(\mu, \sigma^2)$. In some cases, the sampling variable will be emphasized by writing the Gaussian as $\mathcal{N}(\mathrm{x}; \mu, \sigma^2)$.

For easier notation and decomposition of stochasticity, a reparameterization can be performed on $\mathcal{N}(\mu, \sigma^2)$ that extracts the parameters μ and σ^2. The stochasticity is then introduced through the variable ϵ following the standard normal distribution, as follows

$$\mathrm{x} = \mu + \sigma \cdot \epsilon \quad \text{where } \epsilon \sim \mathcal{N}(0, 1). \tag{2.10}$$

Given n independent Gaussians, such that $\mathrm{x}_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, the sum over all random variables is distributed according to

$$\sum^{n} \mathrm{x}_i \sim \mathcal{N}\left(\sum^{n} \mu_i,\ \sum^{n} \sigma_i^2\right). \tag{2.11}$$

Extending the Gaussian distribution to multiple dimensions requires modifying the parameters into vectors and matrices. In d dimensions, the mean is a vector denoted by $\boldsymbol{\mu} \in \mathbb{R}^d$, and the variances are given by the covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ specifying the dependence between dimensions. The PDF for a multivariate Gaussian is given by

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma})}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\intercal \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}, \tag{2.12}$$

where det(·) is the determinant of a square matrix, and $\intercal$ represents the transpose operation. Figure 2.1b visualizes the PDF for a standard bivariate Gaussian.

Similar to the univariate case, a vector random variable distributed as a Gaussian is indicated through as $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The reparameterization in equation 2.10 applies in the multivariate case,

$$\mathbf{x} = \boldsymbol{\mu} + \boldsymbol{\Sigma} \times \boldsymbol{\epsilon} \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{2.13}$$

When $\boldsymbol{\Sigma} = \mathbf{I}\boldsymbol{\beta}$ for some variance vector $\boldsymbol{\beta}$, the covariance matrix is diagonal, meaning that all dimensions are independent. In this thesis, most focus will be on multivariate Gaussians that have diagonal covariance matrices where the elements of $\boldsymbol{\sigma}$ are all equal. This simplifies to $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$, where σ^2 is a scalar denoting the shared variance across all dimensions. Equation (2.12) consequently simplifies to

$$\begin{aligned} p(\boldsymbol{x}) &= \frac{1}{\sqrt{(2\pi)^d \det(\sigma^2\mathbf{I})}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\intercal(\sigma^2\mathbf{I})^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \\ &= \frac{1}{\sqrt{(2\pi\sigma^2)^d}} e^{-\frac{1}{2\sigma^2}(\boldsymbol{x}-\boldsymbol{\mu})^\intercal(\boldsymbol{x}-\boldsymbol{\mu})}. \end{aligned} \tag{2.14}$$

Under the same assumptions, given n independent multivariate Gaussians such that $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$, the sum is given by

$$\sum^n \mathbf{x}_i \sim \mathcal{N}\left(\sum^n \boldsymbol{\mu}_i, \ \left(\sum^n \sigma_i^2\right)\mathbf{I}\right). \tag{2.15}$$

2.3 Likelihood

Assume a distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$, for example a Gaussian, where $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the parameters. For this scenario, the parameters are known, meaning that the PDF $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ can be evaluated. Further, assume a generated i.i.d. data set $\mathcal{D} = \{\boldsymbol{x}^{(i)}\}_{i=1}^n$ suspected to not originate from $p_{\boldsymbol{\theta}}(\mathbf{x})$. The concept of likelihood is useful for testing this. The PDF evaluates a sample to a value that, relative to other inputs, signifies how likely it is under the distribution. As such, high values of the PDF indicate that $\boldsymbol{x}$-values are more likely, and similarly, low values indicate unlikely samples. This is realized through the likelihood function $L(\boldsymbol{\theta}|\boldsymbol{x})$, which measures how well the parameters of the distribution fit with a specific data point $\boldsymbol{x}$ using the PDF $p_{\boldsymbol{\theta}}(\boldsymbol{x})$. A likelihood estimate $L(\boldsymbol{\theta}|\mathcal{D})$ is the total likelihood that the data set falls under the distribution. Since $\mathcal{D}$ is i.i.d., the likelihood estimate is the product of all $\boldsymbol{x}^{(i)}$, each evaluated by the PDF, giving

$$L(\boldsymbol{\theta}|\mathcal{D}) = L\left(\boldsymbol{\theta}|\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)}\right) \tag{2.16}$$

$$= p_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(1)}\right) \cdot p_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(2)}\right) \cdot \ldots \cdot p_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(n)}\right) \tag{2.17}$$

$$= \prod_{i=1}^n p_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right). \tag{2.18}$$

A computational problem related to the evaluation of eq. (2.18) is that floating point arithmetic has a limit in terms of precision. This can potentially lead to challenges for the numerical representation of the likelihood estimate if it approaches a small value. A solution is to work with the log-likelihood $\log L(\boldsymbol{\theta}|\mathcal{D})$ instead. The logarithm is a monotonically increasing function, meaning that the optimum of $L(\boldsymbol{\theta}|\mathcal{D})$ is located at the same parameters as for $\log L(\boldsymbol{\theta}|\mathcal{D})$. The log-likelihood estimate becomes

$$\begin{aligned}\log L(\boldsymbol{\theta}|\mathcal{D}) &= \log \left(p_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(1)}\right) \cdot p_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(2)}\right) \cdot \ldots \cdot p_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(n)}\right)\right) \\ &= \sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right). \qquad (2.19)\end{aligned}$$

2.4 KL Divergence

The Kullback-Leibler (KL) divergence is a similarity measure between two probability distributions. Given two distributions p and q, it is calculated as

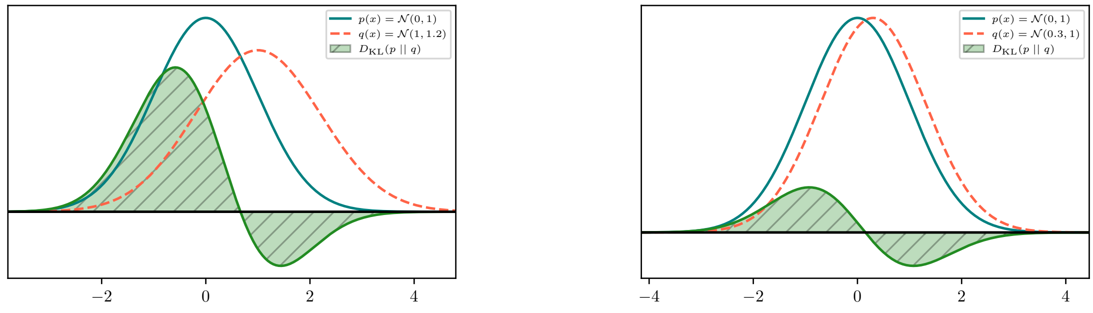
$$\begin{aligned}D_{\mathrm{KL}}(p \mid\mid q) &= \int_{-\infty}^{\infty} p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} d\boldsymbol{x} \qquad (2.20) \\ &= \mathbb{E}_{p(\mathbf{x})}\left[\log \frac{p(\mathbf{x})}{q(\mathbf{x})}\right]. \qquad (2.21)\end{aligned}$$

A property of the KL divergence that will become useful when approximating the likelihood estimate later, is its non-negativity. The lowest possible value is zero, which is the case when p and q are identical.

Given the PDFs of p and q, it is not immediately intuitive why eq. (2.21) is a similarity measure. At its core lies the log-fraction $\log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}$. Due to the quotient logarithmic identity, this corresponds to the difference $\log p(\boldsymbol{x}) - \log q(\boldsymbol{x})$. In order to evaluate this expression to a single value, the expectation is calculated with respect to the distribution in the numerator. As such, the focus will be on samples from one of the distributions. The KL divergence is therefore not symmetric, meaning that $D_{\mathrm{KL}}(p \mid\mid q) \neq D_{\mathrm{KL}}(q \mid\mid p)$. Satisfying the triangle equality is a desirable property for similarity measures. It is expressed as $\Delta(p, r) \leq \Delta(p, q) + \Delta(q, r)$, where p, r, and q are probability distributions, and Δ is a similarity measure. The KL divergence does not satisfy this property, indicating that it is not considered a distance measure.

Figure 2.2 visualizes the KL divergence as the shaded area for two Gaussians with varying degrees of similarity. The left figure displays two Gaussians with different variances and means offset by 1 unit, thus giving a high KL divergence. In contrast, the right figure depicts two Gaussians of higher similarity, resulting in a lower KL divergence.

For the case when the distributions are multivariate Gaussians, meaning $p(\mathbf{x})$ and $q(\mathbf{x})$ with means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, covariance matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$, the KL divergence (Duchi, 2014)



(a)

(b)

Figure 2.2: The KL divergence (shaded area) visualized for two arbitrary univariate Gaussians (blue and red). The green curve plots the integrand in eq. (2.20). **(a)** large divergence. **(b)** smaller divergence.

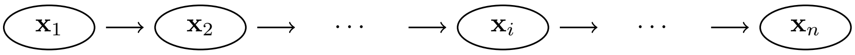


Figure 2.3: A Markov chain of n random variables, modeling how the same entity changes over time.

is given by

$$\begin{aligned} D_{\mathrm{KL}}(q \mid\mid p) &= D_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)) \mid\mid \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)) \\ &= \frac{1}{2}\left[\mathrm{tr}(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1) - d + (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^\intercal \boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) + \log\left(\frac{\det \boldsymbol{\Sigma}_2}{\det \boldsymbol{\Sigma}_1}\right)\right], \end{aligned} \tag{2.22}$$

where $\mathrm{tr}(\cdot)$ is the trace-operation summing the elements along the diagonal and d is the dimensionality of $\mathbf{x}$.

2.5 Markov Chains

A Markov chain models how a dynamic system moves between states. It consists of random variables forming a sequence of events $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, where $\mathbf{x}_n$ denotes either the current state in the case of unbounded chains or the end for bounded chains. The random variables quantify the same entity at separate instances of time. An important characteristic of Markov chains, named the Markov property, is that each variable is only conditioned on the previous variable. A system satisfying

$$p(\mathbf{x}_{i+1} | \mathbf{x}_{1:i}) = p(\mathbf{x}_{i+1} | \mathbf{x}_i) \tag{2.23}$$

retains the Markov property, where $\mathbf{x}_{1:i}$ denotes the set of variables $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_i\}$. The stochastic movement between the states is specified by the conditional distribution $p(\mathbf{x}_{i+1} | \mathbf{x}_i)$, which provides probabilities of the proceeding variable taking on some value based on the current one.

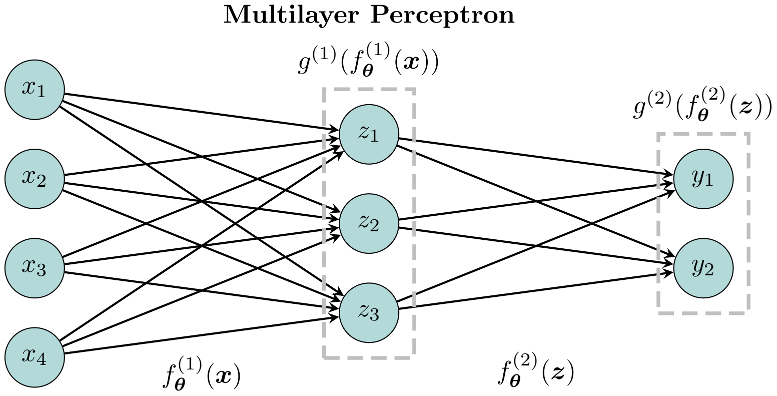

Multilayer Perceptron
$g^{(1)}(f_{\boldsymbol{\theta}}^{(1)}(\boldsymbol{x}))$
$g^{(2)}(f_{\boldsymbol{\theta}}^{(2)}(\boldsymbol{z}))$
x_1
x_2
x_3
x_4
z_1
z_2
z_3
y_1
y_2
$f_{\boldsymbol{\theta}}^{(1)}(\boldsymbol{x})$
$f_{\boldsymbol{\theta}}^{(2)}(\boldsymbol{z})$

Figure 2.4: Multilayer Perceptron with two layers $f_{\boldsymbol{\theta}}^{(1)}$ and $f_{\boldsymbol{\theta}}^{(2)}$. $g^{(1)}$ and $g^{(2)}$ are activation functions.

A Markov chain can be considered a special case of a Bayesian network where the variables appear chronologically, as seen in fig. 2.3. The figure illustrates how each random variable is only conditioned on the previous one, indicated by the arrow connecting them. A specific sequence of realizations of the random variables, $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$, is termed a trajectory. The first variable $\mathbf{x}_1$ follows some initial distribution $p(\mathbf{x}_1)$ that is sampled from to start the chain.

2.6 Deep Learning

Deep neural networks are functions $f_{\boldsymbol{\theta}}$ built up of several layers of neurons connected by weight parameters $\boldsymbol{\theta}$. They define a mapping $\hat{\boldsymbol{y}} = f_{\boldsymbol{\theta}}(\boldsymbol{x})$, where $\boldsymbol{x}$ is an input vector and $\hat{\boldsymbol{y}}$ is the output vector. Figure 2.4 depicts a simple neural network architecture called the multilayer perceptron. In this case, it has two layers $f_{\boldsymbol{\theta}}^{(1)}$ and $f_{\boldsymbol{\theta}}^{(2)}$. Following each transformation is an activation function g, as highlighted by the dashed boxes around the middle and output neurons. These introduce non-linearity into $f_{\boldsymbol{\theta}}$, and are typically not learnable. The neurons represent the values that result from each intermediate function.

With a data set $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^{n}$ of data points $\boldsymbol{x}^{(i)}$ and target values $\boldsymbol{y}^{(i)}$, the model's performance is evaluated through a loss function L. The loss function compares the predictions $\hat{\boldsymbol{y}}$ with the ground truth target values $\boldsymbol{y}$. A low value of the loss over a data set indicates that the model is familiar with the underlying relation between $\boldsymbol{x}$ and $\boldsymbol{y}$. The gradient $\nabla_{\boldsymbol{\theta}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})$ can be used to modify the parameters such that the loss function is minimized. This process is called gradient descent.

Chapter 3

State of the Art

This chapter provides a thorough survey of the theoretical underpinnings of modern DMs. It starts with deep generative modeling in section 3.1 before proceeding to variational inference in section 3.2. These sections serve as a catalyst for the subsequent theoretical discussion on DMs given in sections 3.3 to 3.5, followed by section 3.6 connecting them to optimal transport.

3.1 Deep Generative Modeling

Generative modeling differs from discriminative modeling in what distribution the model is trained to learn. Data sets used in ML commonly consist of data $\boldsymbol{x}$ containing a set of features, and target values $\boldsymbol{y}$ categorizing the data or relating it to some other space. Models trained to infer $\boldsymbol{y}$ from $\boldsymbol{x}$ do so-called supervised learning, learning the distribution $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ by optimizing the parameters $\boldsymbol{\theta}$. Generative modeling, on the other hand, aims to learn $p_{\boldsymbol{\theta}}(\mathbf{x})$, which represents a distribution over the data itself (Ng and Jordan, 2002). There are two main ways of modeling $p_{\boldsymbol{\theta}}(\mathbf{x})$ (Doersch, 2021): Either, the model can take the data $\boldsymbol{x}$ as input, and output the likelihood of said data belonging to a learned distribution. Or, one can train a model on input data $\boldsymbol{x}$ to generate new data points from the learned distribution. This thesis focuses on the latter.

Modeling $p_{\boldsymbol{\theta}}(\mathbf{x})$ is part of unsupervised learning, as the model learns to understand the underlying structure of the data without explicitly knowing what it represents (Kingma and Welling, 2019). For clarity, learning $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y})$ still constitutes a generative model, but now having some overlap with supervised learning. This is called a conditional generative model, synthesizing data $\boldsymbol{x}$ that fits with the queried label $\boldsymbol{y}$. For example, the images in fig. 1.2 were generated by Midjourney, a text-conditional generative model.

Any statistical model that can synthesize data is generative, an example being a Gaussian. However, when statistical models incorporate neural networks, they are specifically known as deep generative models. Architectures such as VAEs and GANs place significant importance on the concept of latent representations. The following sections aim to provide a thorough explanation of the role of latent representations in ML. The word *latent* refers to mainly two concepts, the first being a data representation not directly observed in the training data set, a topic for discussion in this section. The latter regards them as random variables and will be discussed in section 3.2.

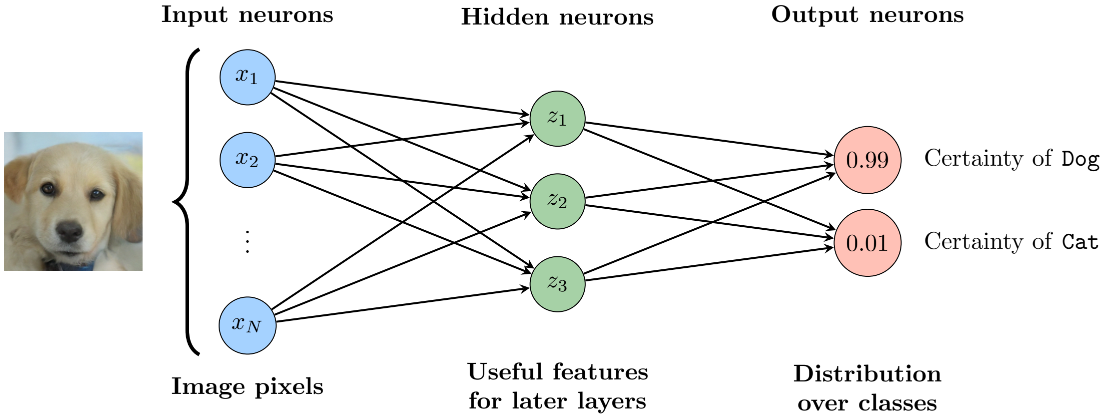


Figure 3.1: A simplified representation learning model where the linear layers successively reduce the dimensionality to obtain class probabilities. The hidden layer outputs unobserved data. Activation functions are omitted.

3.1.1 Learning latent representations

Consider the discriminative animal classifier in fig. 3.1. The first and last set of neurons have dimensionality corresponding to that of the data set, the first being the pixels of the input image $\boldsymbol{x} \in \mathcal{X}$, and the last being the class assignment $\boldsymbol{y} \in \mathcal{Y}$. The latent representation $\boldsymbol{z} \in \mathcal{Z}$ in the middle of the network has values not present in the data set, but inferred by the model through training. The layer between the input neurons and this unobserved representation is therefore often referred to as a *hidden* layer (Goodfellow et al., 2016).

The model has optimized $\boldsymbol{z}$ such that features relevant for determining $\boldsymbol{y}$ from $\boldsymbol{x}$ are encoded in $\boldsymbol{z}$, ultimately minimizing the loss. As such, parts of the network constitute a representation learning model. At the end of a forward pass, a vector of conditional probabilities over all classes given the input is obtained. This representation can be regarded as a highly compressed version of the input image, representing only the degree to which it resembles each class. This means that the final representation is not useful for describing features specific to each class, although it solves the goal of the classifier. For some tasks, the representation is desired to be less reduced and capture necessary factors such as fur color, the shape of the nose, and age.

Representation learning models have some connection with data compression algorithms. Huffman coding compresses data sequences by identifying the frequency of symbols, utilizing these when creating an alternative representation. Symbols that occur infrequently are assigned long binary codes, while common symbols are identified by shorter ones. The data is then compressed by substituting the original symbols with these binary codes, leading to a compact version (Li et al., 2021). While Huffman coding is a lossless compression algorithm, meaning that the original input can be reconstructed with perfect accuracy, this is rarely the case for representation learning models. The reason is that features found not to contribute towards minimizing the loss function will be discarded

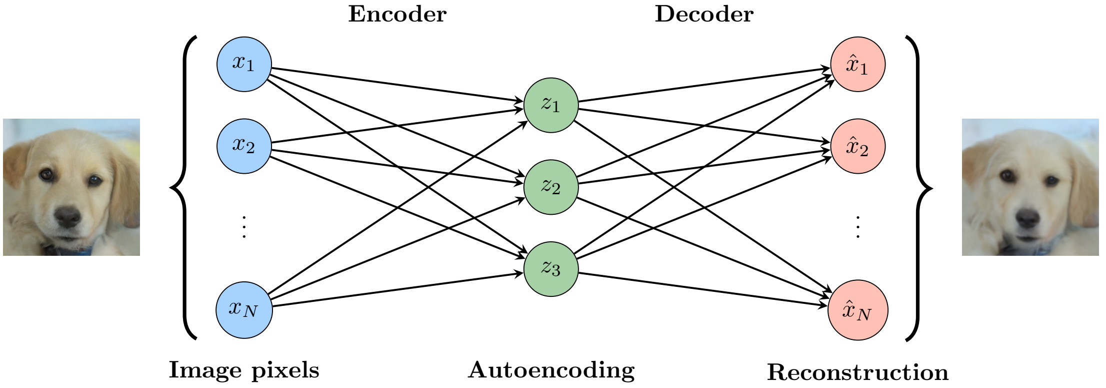


Figure 3.2: A simple deep generative model called the autoencoder. The encoder takes in data $\boldsymbol{x}$, and produces latents $\boldsymbol{z}$ from which the decoder attempts to reconstruct the input data. The decoder can act as a generative model by using random latent vectors $\boldsymbol{z}$ as input.

in the transformation from the input to the latent representation. Taking the forward pass in fig. 3.1 as an example, the background color has little correlation with the animal type, so one can expect the hidden layer not to extract this information.

Figure 3.1 illustrates the forward pass of a discriminative model, yet this thesis is focused on deep generative models. A relevant question is what input data such an architecture should take. Within the image data set of cats and dogs, there are many dependencies between high-level features (Doersch, 2021). For example, a Siamese cat does not have the ears of a Golden Retriever. To relieve the model of having to decide upon what features to generate in the middle of a forward pass, they are decided through an external process prior to the model being queried. This is achieved by first sampling a random vector in some latent space $\mathcal{Z}$ understood by the model, and using it as input.

3.1.2 The Autoencoder: A naïve Deep Generative Model

With some minor modifications, the network architecture in fig. 3.1 can become a generative model. The hidden layer extracts relevant information for the proceeding layer to use, making it advisable to retain it as part of the architecture. A generative model is required to output vectors in the same space as the data being modeled. For this case, instead of outputting the class probabilities, the model should output an image. Figure 3.2 reflects these changes, now taking some data as input and reconstructing it. This architecture is referred to as an autoencoder (Goodfellow et al., 2016). The hidden layer acts as a bottleneck, where the model must carefully choose what information to preserve. While background information was not relevant for the classifier, it is necessary for the generative model to reconstruct those parts. The first component of the model is referred to as an *encoder*. The layers going from the latent space to the generated data are referred to as a *decoder*.

Training an autoencoder is performed by minimizing the mean squared error across the pixels between data samples and their corresponding reconstructions. Upon completion, the decoder can be repurposed as a generative model, mapping randomly sampled vectors $\boldsymbol{z}$ to generated outputs instead of reconstructions. Unfortunately, interpreting these latent inputs is challenging. While the decoder will induce progressive changes in the final output from adjustments on the dimensions of $\boldsymbol{z}$, there is little statistical knowledge about them. It raises the issue of determining the distribution from which to sample $\boldsymbol{z}$. With the described naïve implementation, there are many factors that could affect how the model perceives the latent space, like training data skewness, initial weights, and mini batch sampling order. The VAE, a topic of discussion in section 3.2.2, overcomes this obstacle by forcing a distribution on the latents. Here, the more sophisticated perspective on latents emerges, regarding them as random variables.

3.2 Variational inference

In this section, the focus is on modeling complex probability distributions, describing data on the level of high-quality images such as human faces or animals. Simple distributions are not sufficient by themselves to describe such data. For example, the covariance matrix of a single multivariate Gaussian does not provide enough flexibility in defining the complex dependencies that exist between pixels. Furthermore, the outlined Gaussian has only one maximum, meaning it lacks the capacity to represent multi-modal data, such as cats and dogs together. It is clear that more sophisticated methods must be used.

The goal is to find a distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$ that most closely matches the true data distribution $p_{\text{data}}(\mathbf{x})$ from where the data set originates from. To obtain flexibility, the learned distribution will be in the form of a neural network with parameters $\boldsymbol{\theta}$. After having discussed the autoencoder in section 3.1.2, introducing latent representations seems promising. For that, the posterior distribution $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ is required to infer latents from observed data. Turning to Bayes' rule, eq. (2.3) states that $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x})}$. While useful for seeing how the latent variables are related to the observed variables, it does not allow for a tractable solution for the posterior due to $p_{\boldsymbol{\theta}}(\mathbf{x})$ being in the denominator. Before addressing this issue, the following is an overview of each distribution present in Bayes' rule:

***The posterior* $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$:** The distribution needed to infer the underlying structure of the data. It defines how the latent variable is structured given the observed variables.

***The likelihood* $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$:** Defines how the data is generated from some latent. Can be thought of how likely a data point is in accordance to the latents.

***The evidence* $p_{\boldsymbol{\theta}}(\mathbf{x})$:** Oftentimes called the marginal likelihood, it is the distribution over the observed data. The end goal is to model this distribution.

***The prior* $p_{\boldsymbol{\theta}}(\mathbf{z})$:** Some knowledge that is forced upon the latent variable, regardless of what the data set of observations contains.

Variational inference deals with approximating the posterior by redefining the problem of inferring $\mathbf{z}$ in the presence of $\mathbf{x}$ as an optimization problem (Goodfellow et al., 2016;

Bishop, 2006). For this approach, a surrogate posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is introduced, defined by the parameters $\boldsymbol{\phi}$, not relying on $\boldsymbol{\theta}$. This approximate distribution is tractable and allows for estimating samples $\boldsymbol{z}$ given data $\boldsymbol{x}$.

3.2.1 The Evidence Lower Bound

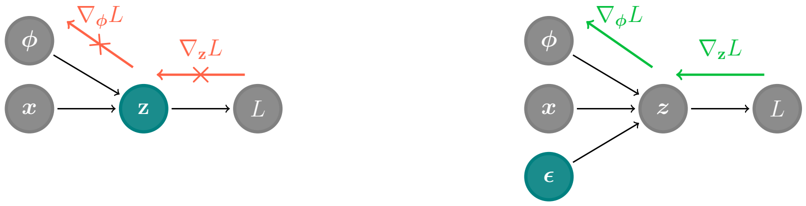
To incorporate the approximate posterior into an optimization scheme (Kingma and Welling, 2022), the likelihood function is rewritten as:

$$
\begin{aligned}
\log L(\boldsymbol{\theta}|\boldsymbol{x}) &= \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \\
&= \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z} && \Bigg| \int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z} = 1 \\
&= \int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \cdot (\log p_{\boldsymbol{\theta}}(\boldsymbol{x})) d\boldsymbol{z} \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x})\right] && \Bigg| \text{ Using eq. (2.7)} \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\boldsymbol{x})}\right] && \Bigg| \text{ Using eq. (2.2)} \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \mathbf{z}) q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\boldsymbol{x}) q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})}\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})}\right] + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})} \left[\log \frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\boldsymbol{x})}\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})}\right] + D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x}) \mid\mid p_{\boldsymbol{\theta}}(\mathbf{z}|\boldsymbol{x})) \qquad (3.1) \\
&\geq \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})}\right] \qquad (3.2)
\end{aligned}
$$

The KL divergence in eq. (3.1) compares the approximate posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ with the ground truth posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$. The latter is not available, making the term challenging to evaluate. Taking advantage of its non-negativity, it is removed in eq. (3.2), turning the expression into a variational *lower bound* on the log-likelihood of the evidence. The term on the right-hand side is therefore called the Evidence Lower Bound (ELBO). Importantly, the left-hand side of this lower bound does not depend on the parameters $\boldsymbol{\phi}$ subject for optimization. In other words, it is a constant. In eq. (3.1), the two terms add to this constant value. As such, maximizing the ELBO is equivalent to minimizing the KL divergence.

3.2.2 The Variational Autoencoder

The VAE was introduced by Kingma and Welling (2022), applying methods from variational inference to develop a probabilistic model that learns latent representations of data through a bottlenecking scheme. Although the name suggests that the VAE might



(a) (b)

Figure 3.3: The reparameterization technique shown in a simplified computation graph. The figure is inspired by Kingma and Welling (2019). **(a)** The stochastic vector $\mathbf{z}$ blocks the loss gradient from reaching $\boldsymbol{\phi}$. **(b)** Extracting the stochastic component $\boldsymbol{\epsilon}$ from $\boldsymbol{z}$ permits gradient flow.

be an extension of autoencoders, the mathematical foundation differs greatly. Still, their connection is relevant for this thesis due to their similarity in terms of architecture. The connection also demonstrates the perspective of latent representations as random variables.

To optimize $p_{\boldsymbol{\theta}}(\mathbf{x})$, Kingma and Welling (2022) maximize the ELBO from eq. (3.2), but in its current form, it is not immediately clear how. To see exactly what constraints need to be fulfilled for a VAE, the ELBO can be decomposed into

$$\begin{aligned}\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})}\right] &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\mathbf{z})p(\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})}\right] \\ &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\mathbf{z})\right] - \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})}\left[\log \frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})}{p(\mathbf{z})}\right] \\ &= \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\mathbf{z})\right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x}) \mid\mid p(\mathbf{z}))}_{\text{prior matching term}} . \end{aligned} \tag{3.3}$$

What has emerged through the reconstruction term is a decoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ measuring the likelihood of $\boldsymbol{x}$ appearing from a latent $\boldsymbol{z}$. The reconstruction term is completed by sampling from the posterior (the encoder) given a data sample $\boldsymbol{x}$ and measuring the ability of the decoder to reconstruct it, typically through a mean squared error. The prior matching term ensures that the encoder generates latent values that fit with the prior distribution. In the case of VAEs, the prior $p(\mathbf{z})$ is typically set to a standard Gaussian. As a result, the prior has no learnable parameters, hence no $\boldsymbol{\theta}$ in the subscript.

In contrast to the autoencoder discussed in section 3.1.2, reconstructing the input data is now a stochastic procedure since the components are represented by distributions (Kingma and Welling, 2019). This makes the encoder a function that is challenging to differentiate. Figure 3.3a shows a simplified view of the computation graph where the stochastic vector (teal node) blocks the gradient of the loss function L from reaching the parameters $\boldsymbol{\phi}$ that should be optimized. The solution, as shown in fig. 3.3b, is to extract the stochastic component from $\boldsymbol{z}$ and define it as its own random variable $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ not parameterized by learnable parameters. The latent variable is now a transformation on the input data $\boldsymbol{x}$ and $\boldsymbol{\epsilon}$ using a function $g_{\boldsymbol{\phi}}$ such that $\mathbf{z} \sim g_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{\epsilon})$ (Goodfellow et al., 2016).

The reparameterization technique has a more mathematical rationalization (Kingma and Welling, 2019). The PDF of $q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})$ is not available, making it infeasible to evaluate the ELBO in eq. (3.3) analytically. As such, sample averaging is used to approximate the expectation. To express the challenge that arises when performing gradient descent on this expectation, assume a function $f_{\boldsymbol{\phi}}(\boldsymbol{z})$ and a distribution $p_{\boldsymbol{\phi}}(\mathbf{z})$ over the random variable $\mathbf{z}$. Both the function and the distribution are parameterized by $\boldsymbol{\phi}$. Computing the gradient with respect to $\boldsymbol{\phi}$ of an expectation taken with respect to a distribution dependent on the same $\boldsymbol{\phi}$ leads to a problem:

$$
\begin{aligned}
\nabla_{\boldsymbol{\phi}} \mathbb{E}_{p_{\boldsymbol{\phi}}(\mathbf{z})}[f_{\boldsymbol{\phi}}(\mathbf{z})] &= \nabla_{\boldsymbol{\phi}} \int p_{\boldsymbol{\phi}}(\boldsymbol{z}) \cdot f_{\boldsymbol{\phi}}(\boldsymbol{z}) \ d\boldsymbol{z} \\
&= \int \nabla_{\boldsymbol{\phi}} (p_{\boldsymbol{\phi}}(\boldsymbol{z}) \cdot f_{\boldsymbol{\phi}}(\boldsymbol{z})) \ d\boldsymbol{z} && (3.4) \\
&= \int \nabla_{\boldsymbol{\phi}} p_{\boldsymbol{\phi}}(\boldsymbol{z}) \cdot f_{\boldsymbol{\phi}}(\boldsymbol{z}) \ d\boldsymbol{z} + \int p_{\boldsymbol{\phi}}(\boldsymbol{z}) \cdot \nabla_{\boldsymbol{\phi}} f_{\boldsymbol{\phi}}(\boldsymbol{z}) \ d\boldsymbol{z} && (3.5) \\
&= \int \nabla_{\boldsymbol{\phi}} p_{\boldsymbol{\phi}}(\boldsymbol{z}) \cdot f_{\boldsymbol{\phi}}(\boldsymbol{z}) \ d\boldsymbol{z} + \mathbb{E}_{p_{\boldsymbol{\phi}}(\mathbf{z})}[\nabla_{\boldsymbol{\phi}} f_{\boldsymbol{\phi}}(\boldsymbol{z})]. && (3.6)
\end{aligned}
$$

In eq. (3.4), the Leibniz integral rule is used to change the order of the integral and the gradient operation. Subsequently, the product rule of differentiation is used in eq. (3.5) to find the gradient of the product of the two functions. Neither of the functions inside the integral in the first term in eq. (3.6) are necessarily PDFs, making sample averaging the gradient of $\mathbb{E}_{p_{\boldsymbol{\phi}}(\mathbf{z})}[f_{\boldsymbol{\phi}}(\mathbf{z})]$ inaccurate. Using a function $g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}) \sim p_{\boldsymbol{\phi}}(\mathbf{z})$ such that $\boldsymbol{z} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon})$ is determined deterministically, the expectation can be formulated to not be taken with respect to a distribution that is dependent on the parameters $\boldsymbol{\phi}$ that are subject to optimization. It holds that

$$
\begin{aligned}
\nabla_{\boldsymbol{\phi}} \mathbb{E}_{p_{\boldsymbol{\phi}}(\mathbf{z})}[f_{\boldsymbol{\phi}}(\mathbf{z})] &= \nabla_{\boldsymbol{\phi}} \mathbb{E}_{p(\boldsymbol{\epsilon})}[f_{\boldsymbol{\phi}}(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}))] \\
&= \mathbb{E}_{p(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\phi}} f_{\boldsymbol{\phi}}(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}))], && (3.7)
\end{aligned}
$$

and consequently, the parameters can be optimized through stochastic gradient descent, where the gradients are estimated with sample averaging. In eq. (3.3), the expectations are part of the loss function that is minimized during training. Using the reparameterization technique as shown in eq. (3.7) ensures that the parameters can be optimized reliably.

The final element is to fit the reparameterization technique with the network architecture. The computation graph in fig. 3.3b leaves out some details on how $\boldsymbol{x}$ is transformed and combined with $\boldsymbol{\epsilon}$. A reasonable approach, as the prior is a Gaussian, is to have the encoder output parameters forming a Gaussian distribution, as shown in fig. 3.4. Essentially, the encoder is now the stochastic function $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x}))$. The prior matching term then acts as a regularization on the outputted parameters, ensuring that they model a standard Gaussian. Analogously to the simple autoencoder, new data can be generated by using a random vector in the latent space $\mathcal{Z}$ as input to the decoder. Fortunately, the distribution from which to sample this is now explicitly defined, being the prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$.

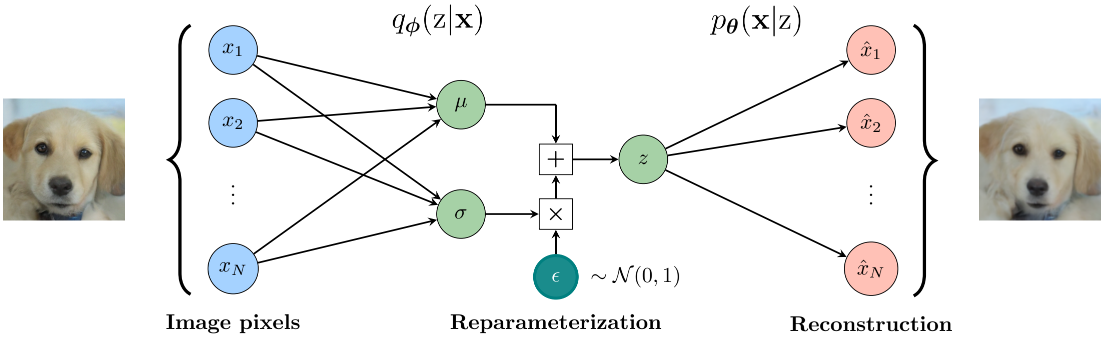


Figure 3.4: A simple VAE architecture utilizing the reparameterization technique to allow gradient flow to the encoder. The latent space is simplified to one dimension to highlight the mechanism of the reparameterization technique.

For a cleaner presentation, the displayed VAE in fig. 3.4 has only one dimension in the latent space. Typically, there are multiple dimensions, with a pair of μ and σ accompanying each. An important convention illustrated in the figure, is that despite the decoder being a stochastic function in eq. (3.3), it is usually implemented as a deterministic mapping from a latent $\boldsymbol{z}$ to data $\boldsymbol{x}$.

3.3 Diffusion Models

Sohl-Dickstein et al. (2015) had the idea of training a generative model by gradually recovering data from noise corruption. Their goal was to develop a flexible method for modeling complex probability distributions that are computationally tractable to learn and perform sampling from. Inspired by diffusion processes from thermodynamics, data is iteratively diffused with varying amounts of Gaussian noise until all information is lost. The model is taught to restore data by reversing the diffusion process, making up a generative model that is capable of synthesizing new data by starting from pure Gaussian noise.

Formally, a DM works on a set of random variables $\mathbf{x}_{0:T}$ making up a Markov chain. $q(\mathbf{x}_0)$ denotes the initial distribution over clean data $\mathbf{x}_0$, assumed to be where data set samples $\boldsymbol{x}_0$ originate from. Previously, $p_{\text{data}}(\mathbf{x})$ was used to denote this distribution, but with the appearance of a Markov chain, it is more reasonable to denote it as $q(\mathbf{x}_0)$. The random variables following $\mathbf{x}_0$ in the chain, $\mathbf{x}_{1:T}$, are samples perturbed with Gaussian noise. From the perspective of DMs, all of these are addressed as latent variables. The noising process, called the forward process, involves starting from a clean data set instance $\boldsymbol{x}_0$, and gradually perturbing it to $\boldsymbol{x}_T$ over T time steps, arriving at pure Gaussian noise. The transition distribution at any time step t is denoted by $q(\mathbf{x}_t|\mathbf{x}_{t-1})$, and is fixed, meaning it contains no learnable parameters. It represents a stochastic function of how much noisier $\mathbf{x}_t$ will get compared to $\mathbf{x}_{t-1}$. The reverse of this transition, $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, involves removing noise from the perturbed instances. Unfortunately, transitioning in

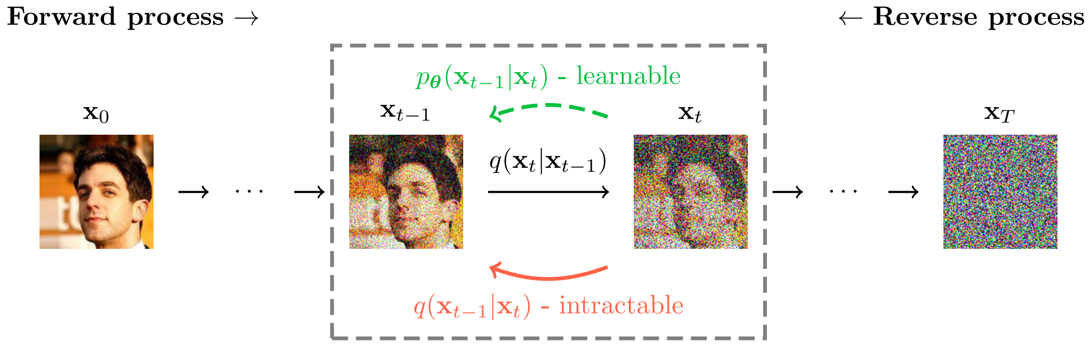

Forward process →
← Reverse process
$p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$ - learnable
$\mathbf{x}_0$
$\mathbf{x}_{t-1}$
$\mathbf{x}_t$
$\mathbf{x}_T$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ - intractable

Figure 3.5: The forward and reverse process of a DM depicted on an instance from CelebA. The variables $\mathbf{x}_{0:T}$ form a Markov chain with gradual amounts of noise added for increasing t.

this direction is analytically intractable. The solution is to approximate it using an auxiliary distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$ parameterized by neural network parameters $\boldsymbol{\theta}$. This is a stochastic function computing the distribution of a less noised sample $\mathbf{x}_{t-1}$ given a noisier sample $\mathbf{x}_t$.

An example trajectory of the Markov chain is shown in fig. 3.5 to accentuate the dynamics of the diffusion process, where an instance $\boldsymbol{x}_0$ from the CelebA data set is gradually noised. The gray box highlights the interplay between two arbitrary neighboring latents $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$. Seen from left to right, the figure reflects diffusion, where information gradually vanishes and entropy is increased. Conversely, from right to left, it depicts a generative process where small amounts of noise are removed at each time step to reveal realistic data. Although there are multiple latents present, a network evaluation conditions on one latent $\mathbf{x}_t$ at a time.

Figure 3.5 illustrates an important detail about DMs, namely that all latents share the same dimensionality with the input data. In contrast to the VAE, there is no bottleneck forcing the model to extract the most important features into the latent space. The latent space instead resides outside of the model architecture. As such, the notion of latents differs from VAEs.

3.4 Denoising Diffusion Probabilistic Models

Building upon the works of Sohl-Dickstein et al. (2015), the Denoising Diffusion Probabilistic Model (DDPM) framework was introduced by Ho et al. (2020), highlighting important implementation details and providing insight on the potential such models have in synthesizing high-quality images. Since then, this framework has become the foundation that several influential papers have based their improvements on. The following examples motivate this claim. Song et al. (2022) study a trade-off between sample quality and efficiency by only taking into account a subset of the chained variables $\mathbf{x}_{1:T-1}$, essentially disregarding many of the latents when generating data. Dhariwal and Nichol (2021) and Ho and Salimans (2022) achieve conditional sampling with their respective methods called classifier-based guidance and classifier-free guidance. The first utilizes knowledge from an independently trained classifier to guide the noise-removal process towards desired modes of the target distribution. Interestingly, this approach is compatible with any existing unconditional DM. The second incorporates the class knowledge into the model during training, ending up with a class-conditional model. Saharia et al. (2022) and Rombach et al. (2022) demonstrate the superior performance of DMs when conditioning on text descriptions and training them on large data sets, obtaining a model that is capable of visualizing a diverse range of objects and scenes. Additionally, Rombach et al. (2022) turns the DDPM framework to the latent space of a VAE, in order to reduce computational requirements for sampling.

On the path of gaining a solid understanding of the mechanisms behind DMs, the mentioned articles demonstrate that the DDPM framework is a good starting point. Therefore it will be used as the basis for this thesis as well. Fortunately, much of the notation and terminology is used consistently throughout the literature, meaning that ideas not covered here can be grasped by readers interested in further reading. In the following discussions, DDPM refers to the base framework, while DM refers to the family of generative models encompassing DDPM and those extending it.

3.4.1 Forward process

The forward process is a fixed Markov chain over T time steps. As previously mentioned, it starts with data $\mathbf{x}_0 \sim q(\mathbf{x}_0)$, which it iteratively corrupts with Gaussian noise up to $\mathbf{x}_T$, becoming indistinguishable from pure noise. $\mathbf{x}_t$ denotes an intermediate latent, being a blend of pure data and noise. A variance vector $\boldsymbol{\beta} \in \langle \mathbf{0}, \mathbf{1} \rangle$ of scalars $\beta_1, \beta_2, \ldots, \beta_T$ defines how much noise is introduced when transitioning from a state to the next. Specifically, β_t determines the amount of noise added to latent $\mathbf{x}_{t-1}$ to produce $\mathbf{x}_t$. In the context of DMs, all details surrounding $\boldsymbol{\beta}$ are referred to as the *variance schedule*. Throughout the literature, several ways of specifying this schedule are presented. Among the simpler ones, as used by Ho et al. (2020), is to define the endpoints and interpolate β_t linearly between these, hence the name *linear schedule*. Understanding the effects of the variance schedule is crucial for understanding the system as a whole, thus more details will be given in section 3.4.3.

One step of the forward process is defined as the conditional Gaussian distribution

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \tag{3.8}$$

being a stochastic linear transformation on $\mathbf{x}_{t-1}$. Due to the Markov property, the joint conditional on the starting point $\mathbf{x}_0$ is given by

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}). \tag{3.9}$$

To more explicitly show the linear transformation, the reparameterization in eq. (2.13) can be used to rewrite the Markov transition as

$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{3.10}$$

Ho et al. (2020) highlight a property of the forward process utilized to compute the outcome of the iterative process from 0 to t in closed form, leading to a drastic reduction in complexity during training. This is possible due to $q(\mathbf{x}_t|\mathbf{x}_0)$ also being a Gaussian, which can be proved by starting from eq. (3.10) and substituting in solutions from previous time steps until $\mathbf{x}_0$ explicitly appears as a factor in the mean. For this purpose, each $\boldsymbol{\epsilon}$ is distinguished using the relative time step from $\mathbf{x}_t$ in the subscript. For algebraic relief, a counterpart to the variance schedule, $\alpha_t = 1 - \beta_t$, is used to show that

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}_{t-1} && \Big|\ \text{See eq. (3.10)} \\
&= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \\
&= \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2}) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} && \Big|\ \mathbf{x}_{t-1} \text{ is eq. (3.10) with } t \text{ shifted by } -1 \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\alpha_t - \alpha_t\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} && (3.11) \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{(\sqrt{\alpha_t - \alpha_t\alpha_{t-1}})^2 + (\sqrt{1-\alpha_t})^2}\bar{\boldsymbol{\epsilon}}_{t-2} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\alpha_t - \alpha_t\alpha_{t-1} + 1 - \alpha_t}\bar{\boldsymbol{\epsilon}}_{t-2} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1 - \alpha_t\alpha_{t-1}}\bar{\boldsymbol{\epsilon}}_{t-2} && (3.12) \\
&= \ldots \\
&= \sqrt{\alpha_t\alpha_{t-1}\cdots\alpha_{t-k}}\mathbf{x}_{t-k} + \sqrt{1 - \alpha_t\alpha_{t-2}\cdots\alpha_{t-k}}\bar{\boldsymbol{\epsilon}}_{t-k} && (3.13) \\
&= \sqrt{\alpha_t\alpha_{t-1}\cdots\alpha_{t-k}}\mathbf{x}_{t-k} + (\mathbf{0} + \sqrt{1 - \alpha_t\alpha_{t-2}\cdots\alpha_{t-k}}\bar{\boldsymbol{\epsilon}}_{t-k}) && (3.14) \\
&= \ldots \\
&= \sqrt{\alpha_t\alpha_{t-1}\cdots\alpha_2\alpha_1}\mathbf{x}_0 + \sqrt{1 - \alpha_t\alpha_{t-1}\cdots\alpha_2\alpha_1}\bar{\boldsymbol{\epsilon}}_0. && (3.15)
\end{aligned}$$

In eq. (3.11), $\sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}$ is distributed by $\mathcal{N}(\mathbf{0}, (1-\alpha_t)\mathbf{I})$, and $\sqrt{\alpha_t - \alpha_t\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2}$ by $\mathcal{N}(\mathbf{0}, (\alpha_t - \alpha_t\alpha_{t-1})\mathbf{I})$. From eq. (2.15), their sum gives the distribution $\mathcal{N}(\mathbf{0}, (1-\alpha_t\alpha_{t-1})\mathbf{I})$, which $\sqrt{1-\alpha_t\alpha_{t-1}}\bar{\boldsymbol{\epsilon}}_{t-2}$ in eq. (3.12) follows (Luo, 2022).

Inserting $\mathbf{x}_{t-k}$, as seen in eq. (3.13), it is revealed that $\sqrt{1-\alpha_t\alpha_{t-1}\cdots\alpha_{t-k}}\bar{\boldsymbol{\epsilon}}_{t-k}$ is distributed in accordance with $\mathcal{N}(\mathbf{0}, (1-\alpha_t\alpha_{t-1}\cdots\alpha_{t-k})\mathbf{I})$ for $k = 2, \ldots, t$. Using the reparameterization technique (eq. (2.13)), eq. (3.14) shows that $\bar{\boldsymbol{\epsilon}}_{t-k}$ follows a standard Gaussian.

This proves that $q(\mathbf{x}_t|\mathbf{x}_0)$ is also a Gaussian, as seen in eq. (3.15), where the mean is proportional to the square root of the cumulative product of α_t, and the standard deviation is scaled by the square root of the complement of the same cumulative product. From this, Ho et al. (2020) define $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, used in the closed form expression for the forward process at an arbitrary time step t,

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{3.16}$$

giving the multistep transition distribution

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}). \tag{3.17}$$

The scaling factor $\bar{\alpha}_T$ for the mean in eq. (3.16) when computing $\mathbf{x}_T$ is made up of several multiplications of non-negative numbers restricted to the range $\langle 0, 1 \rangle$, resulting in a small number. Stated mathematically, $\lim_{T\to\infty} \bar{\alpha}_T = 0$. As a consequence,

$$\begin{aligned} q(\mathbf{x}_T|\mathbf{x}_0) &\approx \mathcal{N}(\mathbf{x}_T; \sqrt{0}\mathbf{x}_0, (1-0)\mathbf{I}) \\ &= \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I}), \\ &= q(\mathbf{x}_T), \end{aligned} \tag{3.18}$$

which converges to a standard Gaussian independent of the conditioning variable $\mathbf{x}_0$ under ideal conditions of the variance schedule and large T.

3.4.2 Reverse process

Section 3.4.1 demonstrated the corruption of data through a Markov chain, with the transition distribution being a conditional Gaussian $q(\mathbf{x}_t|\mathbf{x}_{t-1})$. To obtain a generative model following the distribution $q(\mathbf{x}_0)$, the Markov transition must be reversed, meaning going from $\mathbf{x}_t$ to $\mathbf{x}_{t-1}$. As previously mentioned, $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ does not have a tractable solution. A remedy is to approximate this transition with a neural network that defines $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$.

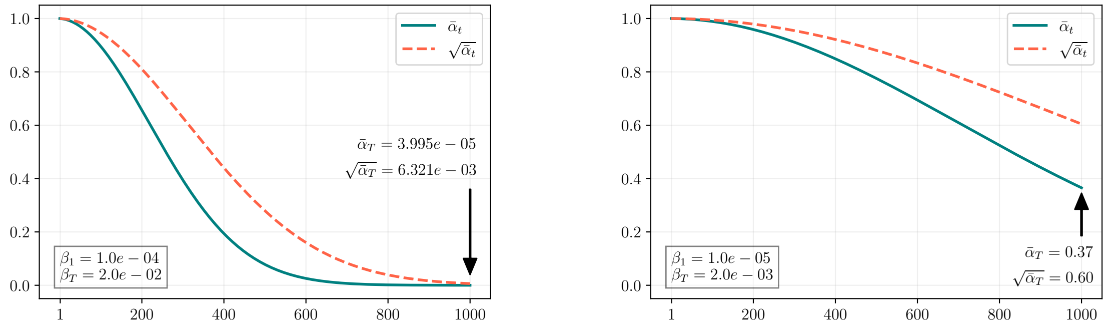
To reverse the chain and sample new data, the initial distribution $p(\mathbf{x}_T)$ must be defined. Equation (3.18) states that $q(\mathbf{x}_T)$ is reminiscent of a standard Gaussian. As such, Ho et al. (2020) define the prior as $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. Similar to the VAE, this distribution is not parameterized by any learnable parameters, hence no subscript. The joint distribution of the reverse process is then

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t), \tag{3.19}$$

utilizing the fact that the Markov property is also relevant in the reverse direction. The joint defines a Markov chain going from $\mathbf{x}_T$ to $\mathbf{x}_0$, often termed the generative or denoising process. Synthesizing new data begins by sampling a $\boldsymbol{x}_T$ from $p(\mathbf{x}_T)$, and iteratively denoising it by sampling from $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$ until $\boldsymbol{x}_0$ is obtained.

3.4.3 The variance schedule

Both β_t and $\bar{\alpha}_t$ are used to determine the amount of noise to be added, but while the former facilitates the local change between two neighboring latents, the latter determines the difference between a latent $\mathbf{x}_t$ and clean data $\mathbf{x}_0$. Ho et al. (2020) use $\beta_1 = 10^{-4}$ and $\beta_T = 0.02$, linearly interpolating $\beta_{2:T-1}$ between these for $T = 1000$ time steps. Figure 3.6a shows how $\bar{\alpha}_t$ behaves under these conditions, together with its square root used in eq. (3.16). $\sqrt{\bar{\alpha}_t}$ illustrates the level of cleanliness in the data over time. For



(a) (b)

Figure 3.6: $\bar{\alpha}_t$ and $\sqrt{\bar{\alpha}_t}$ plotted for all time steps 1 through T for a **(a)** viable schedule and **(b)** an unviable schedule.

example, $\mathbf{x}_1$ will be almost indistinguishable from $\mathbf{x}_0$ as $\sqrt{\bar{\alpha}_1} \approx 1$. The plot also highlights the desirable property that $\mathbf{x}_T$ will be approximately distributed by a standard Gaussian since $\sqrt{\bar{\alpha}_T}$ converges to 0.

A critical concern has been left out of the discussion up until now. Diffusing data encompasses a simple linear transformation with Gaussian noise added (eq. (3.10)). Is the denoising operation expected to follow the same distributional family? On the macroscopic level with large step sizes, this is not the case. Looking at a shape undergoing diffusion, one cannot expect the reverse operation to have the same simple form as the noising procedure. The variance schedule has the role of alleviating this concern: If β_t is set to a large value, $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ will deviate too much, making the task of translating between them difficult for the model. Conversely, a small value means they are *almost* identically distributed. In the limit of small β_t, the reverse transition has the same functional form as the forward transition (Sohl-Dickstein et al., 2015; Feller, 1949).

Subsequently, the chosen value for T must be balanced together with β_t for the variance schedule to be ideal. Figure 3.6b illustrates this requirement. Although smaller values are chosen for β_t compared to fig. 3.6a, it leads to an unviable schedule since $q(\mathbf{x}_T|\mathbf{x}_0)$ fails to reach a standard Gaussian. A possible fix to this problematic schedule is to increase T, letting $\bar{\alpha}_T$ and $\sqrt{\bar{\alpha}_t}$ converge to 0.

3.4.4 Determining the objective

Ho et al. (2020) derive the training objective from the ELBO. To make it clear why this lower bound applies to this case, the same procedure is followed as in section 3.2.1, now replacing the joint $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ with the generative process $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})$, and the posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ with the forward process $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$. Starting from the marginal log-likelihood, it holds that

$$\begin{align}
\log L(\boldsymbol{\theta}|\boldsymbol{x}_0) &= \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0) \notag \\
&= \int q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) \cdot (\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0)) d\boldsymbol{x}_{1:T} \notag \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0)\right] && \Big| \text{ Using eq. (2.7)} \notag \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})}{p_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\right] && \Bigg| \text{ Using eq. (2.2)} \notag \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}{p_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}|\boldsymbol{x}_0) q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\right] \notag \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\right] + \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}{p_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\right] \notag \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\right] + D_{\mathrm{KL}}(q(\mathbf{x}_{1:T}|\boldsymbol{x}_0) \mid\mid p_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}|\boldsymbol{x}_0)) \tag{3.20} \\
&\geq \mathbb{E}_{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\right]. \tag{3.21}
\end{align}$$

Similarly to the VAE, the KL term in eq. (3.20) measures the variation between two posteriors, one unobtainable through analytical means. The solution is again to construct a variational lower bound on the log-likelihood of the data. This ELBO must also be decomposed in order to arrive at the conditions that must be met when maximizing it.

3.4.4.1 Decomposing the ELBO

The Markov property states that $\mathbf{x}_t$ is only based on $\mathbf{x}_{t-1}$ during the forward process. Including $\mathbf{x}_0$ as a condition leads to no changes in the mean function of the forward transition, meaning that $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)$. This does not change the fact that the reverse transition must be approximated, since the ground truth $\mathbf{x}_0$ is not available when sampling. Using Bayes' rule (eq. (2.3)), the forward transition conditioned on $\mathbf{x}_0$ can be formulated as

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) = \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}. \tag{3.22}$$

When decomposing the ELBO, this reformulation will lead to an expectation representing the training objective taken with regards to fewer variables (Luo, 2022). With these tools, the ELBO becomes

$$
\begin{aligned}
&\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0)\\
&\geq \mathbb{E}_{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\right]\\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1},\boldsymbol{x}_0)}\right]\\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\mathbf{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\mathbf{x}_1)\prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_1|\boldsymbol{x}_0)\prod_{t=2}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] && (3.23)\\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\mathbf{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\mathbf{x}_1)\prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_1|\boldsymbol{x}_0)\prod_{t=2}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1},\boldsymbol{x}_0)}\right]\\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\mathbf{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\boldsymbol{x}_0)} + \log \prod_{t=2}^{T} \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1},\boldsymbol{x}_0)}\right]\\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\mathbf{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\boldsymbol{x}_0)} + \log \prod_{t=2}^{T} \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\boldsymbol{x}_0)q(\mathbf{x}_t|\boldsymbol{x}_0)}{q(\mathbf{x}_{t-1}|\boldsymbol{x}_0)}}\right] && (3.24)\\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\mathbf{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\boldsymbol{x}_0)} + \underbrace{\log \prod_{t=2}^{T} \frac{q(\mathbf{x}_{t-1}|\boldsymbol{x}_0)}{q(\mathbf{x}_t|\boldsymbol{x}_0)}}_{\text{Telescoping product}} + \log \prod_{t=2}^{T} \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\boldsymbol{x}_0)}\right] && (3.25)\\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\mathbf{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\mathbf{x}_1)}{\cancel{q(\mathbf{x}_1|\boldsymbol{x}_0)}} + \log \frac{\cancel{q(\mathbf{x}_1|\boldsymbol{x}_0)}}{q(\mathbf{x}_T|\boldsymbol{x}_0)} + \log \prod_{t=2}^{T} \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\boldsymbol{x}_0)}\right]\\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\mathbf{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_T|\boldsymbol{x}_0)} + \sum_{t=2}^{T} \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\boldsymbol{x}_0)}\right]\\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\mathbf{x}_1)] + \mathbb{E}_{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\boldsymbol{x}_0)}\right] + \sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\boldsymbol{x}_0)}\right]\\
&= \mathbb{E}_{q(\mathbf{x}_1|\boldsymbol{x}_0)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\mathbf{x}_1)] + \mathbb{E}_{q(\mathbf{x}_T|\boldsymbol{x}_0)}\left[\log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\boldsymbol{x}_0)}\right]\\
&\qquad\qquad + \sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_t|\boldsymbol{x}_0)}\left[\mathbb{E}_{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\boldsymbol{x}_0)}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\boldsymbol{x}_0)}\right]\right] && (3.26)\\
&= \underbrace{\mathbb{E}_{q(\mathbf{x}_1|\boldsymbol{x}_0)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\mathbf{x}_1)]}_{\text{reconstruction term } \mathcal{L}_0} - \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\boldsymbol{x}_0)\;||\;p(\mathbf{x}_T))}_{\text{prior matching term } \mathcal{L}_T}\\
&\qquad\qquad - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\mathbf{x}_t|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\boldsymbol{x}_0)\;||\;p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)\right]}_{\text{denoising matching term } \mathcal{L}_{t-1}}. && (3.27)
\end{aligned}
$$

In eq. (3.25), a product of fractions has been extracted which constitutes a telescoping product, evaluating to the term $\frac{q(\mathbf{x}_1|\boldsymbol{x}_0)}{q(\mathbf{x}_T|\boldsymbol{x}_0)}$ when expanded. Equation (3.26) applies a reduction of variables in the distributions that the expectations are being taken with respect to. Proof and further details around this is given in appendix A. The three terms in eq. (3.27) that make up a lower bound of the log-likelihood of the observed data have their own interpretations and names:

Reconstruction term ($\mathcal{L}_0$): This term emerged in eq. (3.23) when changing the initialization of the products from $t = 1$ to $t = 2$. It differs from the other two by not measuring any similarity through the KL divergence. Instead, it is an expectation over the transition from $\mathbf{x}_1$ to $\boldsymbol{x}_0$. When synthesizing new data, the $\mathcal{L}_0$ term is satisfied by not adding noise correction during the last sample step.

Prior matching term ($\mathcal{L}_T$): This term measures the similarity between $q(\mathbf{x}_T|\boldsymbol{x}_0)$ and $p(\mathbf{x}_T)$. Equation (3.18) made it clear that the former converges towards a standard Gaussian. In section 3.4.2, it was decided that $p(\mathbf{x}_T)$ is equal to a standard Gaussian. Therefore, under the desired conditions of the variance schedule, $\mathcal{L}_T$ will be constant and can be disregarded while optimizing the model.

Denoising matching term ($\mathcal{L}_{t-1}$): Part of the set $\mathcal{L}_{1:T-1}$, this term measures the ability of the approximative distribution to match the ground truth denoising step from t to $t-1$. The ground truth denoising step $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{x}_0)$ is tractable due to the condition on $\boldsymbol{x}_0$.

3.4.4.2 Deriving the form for the ground truth denoising step

The ground truth denoising step $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ in $\mathcal{L}_{t-1}$ is a conditional Gaussian in the limit of infinitesimal step sizes. To evaluate it, its PDF must be found. Bayes' rule reveals which distributions are needed to solve for the reverse transition:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \tag{3.28}$$

Expressions for the factors in eq. (3.28) are found using the formulas presented in section 3.4.1:

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

Equation (2.14) states that the PDF for a multivariate Gaussian with scaled diagonal covariance matrix $\sigma^2\mathbf{I}$ is given by $p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi\sigma^2)^d}} e^{-\frac{1}{2\sigma^2}(\boldsymbol{x}-\boldsymbol{\mu})^\intercal(\boldsymbol{x}-\boldsymbol{\mu})}$. The exponent contains both the variance scaling factor σ^2 and the mean $\boldsymbol{\mu}$. As such, the exponent of the product of density functions in eq. (3.28) can be studied to determine the functional form of the ground truth denoising step. In the following derivation, the notation $\boldsymbol{x}^2$ is used as shorthand to denote the inner product $\boldsymbol{x}^\intercal\boldsymbol{x}$, where $\boldsymbol{x}$ is an expression that encompasses one or multiple vectors. Additionally, for two arbitrary vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ such that $\boldsymbol{x}^\intercal\boldsymbol{x} = (\boldsymbol{a}-\boldsymbol{b})^\intercal(\boldsymbol{a}-\boldsymbol{b})$, its expansion $\boldsymbol{a}^\intercal\boldsymbol{a} - \boldsymbol{a}^\intercal\boldsymbol{b} - \boldsymbol{b}^\intercal\boldsymbol{a} + \boldsymbol{b}^\intercal\boldsymbol{b}$ will be written as $\boldsymbol{a}^2 - 2\boldsymbol{a}\boldsymbol{b} + \boldsymbol{b}^2$. It holds that

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

whose PDF is proportional to

$$\begin{aligned}
&\propto \exp\left\{-\left[\frac{(\boldsymbol{x}_t-\sqrt{\alpha_t}\boldsymbol{x}_{t-1})^2}{2(1-\alpha_t)}+\frac{(\boldsymbol{x}_{t-1}-\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0)^2}{2(1-\bar{\alpha}_{t-1})}-\frac{(\boldsymbol{x}_t-\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0)^2}{2(1-\bar{\alpha}_t)}\right]\right\}\\
&=\exp\left\{-\frac{1}{2}\left[\frac{\boldsymbol{x}_t^2-2\sqrt{\alpha_t}\boldsymbol{x}_t\boldsymbol{x}_{t-1}+\alpha_t\boldsymbol{x}_{t-1}^2}{1-\alpha_t}+\frac{\boldsymbol{x}_{t-1}^2-2\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_{t-1}\boldsymbol{x}_0+\bar{\alpha}_{t-1}\boldsymbol{x}_0^2}{1-\bar{\alpha}_{t-1}}-\frac{(\boldsymbol{x}_t-\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0)^2}{2(1-\bar{\alpha}_t)}\right]\right\}\\
&=\exp\left\{-\frac{1}{2}\left[\frac{\alpha_t\boldsymbol{x}_{t-1}^2}{1-\alpha_t}+\frac{\boldsymbol{x}_{t-1}^2}{1-\bar{\alpha}_{t-1}}-\frac{2\sqrt{\alpha_t}\boldsymbol{x}_t\boldsymbol{x}_{t-1}}{1-\alpha_t}-\frac{2\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_{t-1}\boldsymbol{x}_0}{1-\bar{\alpha}_{t-1}}\right.\right.\\
&\qquad\qquad\left.\left.+\frac{\boldsymbol{x}_t^2}{1-\alpha_t}+\frac{\bar{\alpha}_{t-1}\boldsymbol{x}_0^2}{1-\bar{\alpha}_{t-1}}-\frac{(\boldsymbol{x}_t-\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0)^2}{2(1-\bar{\alpha}_t)}\right]\right\}.
\end{aligned} \tag{3.29}$$

The last three terms in eq. (3.29) are not dependent on the sample variable $\boldsymbol{x}_{t-1}$. Since the exponent in eq. (2.14) is a square, these are proportional to $\boldsymbol{\mu}^\intercal\boldsymbol{\mu}$. The mean resides in the remaining part of the square, $\boldsymbol{x}^\intercal\boldsymbol{x}-\boldsymbol{x}^\intercal\boldsymbol{\mu}-\boldsymbol{\mu}^\intercal\boldsymbol{x}$, so the three terms in eq. (3.29) can be set aside. Their sum will be denoted by $C(\boldsymbol{x}_0,\boldsymbol{x}_t,t)$. Further simplifying eq. (3.29) gives

$$\begin{aligned}
&=\exp\left\{-\frac{1}{2}\left[\left(\frac{\alpha_t}{1-\alpha_t}+\frac{1}{1-\bar{\alpha}_{t-1}}\right)\boldsymbol{x}_{t-1}^2-2\left(\frac{\sqrt{\alpha_t}\boldsymbol{x}_t}{1-\alpha_t}+\frac{\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0}{1-\bar{\alpha}_{t-1}}\right)\boldsymbol{x}_{t-1}+C(\boldsymbol{x}_0,\boldsymbol{x}_t,t)\right]\right\}\\
&=\exp\left\{-\frac{1}{2}\left[\left(\frac{\alpha_t(1-\bar{\alpha}_{t-1})+1-\alpha_t}{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}\right)\boldsymbol{x}_{t-1}^2-2\left(\frac{\sqrt{\alpha_t}\boldsymbol{x}_t}{1-\alpha_t}+\frac{\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0}{1-\bar{\alpha}_{t-1}}\right)\boldsymbol{x}_{t-1}+C(\boldsymbol{x}_0,\boldsymbol{x}_t,t)\right]\right\}\\
&=\exp\left\{-\frac{1}{2}\left[\left(\frac{1-\bar{\alpha}_t}{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}\right)\boldsymbol{x}_{t-1}^2-2\left(\frac{\sqrt{\alpha_t}\boldsymbol{x}_t}{1-\alpha_t}+\frac{\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0}{1-\bar{\alpha}_{t-1}}\right)\boldsymbol{x}_{t-1}+C(\boldsymbol{x}_0,\boldsymbol{x}_t,t)\right]\right\}.
\end{aligned} \tag{3.30}$$

Here, $\alpha_t\bar{\alpha}_{t-1}=\bar{\alpha}_t$ is used. For compactness, $\tilde{\beta}_t=\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}$ is employed when continuing from eq. (3.30) to show that

$$\begin{aligned}
&=\exp\left\{-\frac{1}{2\tilde{\beta}_t}\left[\boldsymbol{x}_{t-1}^2-2\frac{\left(\frac{\sqrt{\alpha_t}\boldsymbol{x}_t}{1-\alpha_t}+\frac{\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0}{1-\bar{\alpha}_{t-1}}\right)}{\left(\frac{1-\bar{\alpha}_t}{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}\right)}\boldsymbol{x}_{t-1}+\tilde{\beta}_tC(\boldsymbol{x}_0,\boldsymbol{x}_t,t)\right]\right\}\\
&=\exp\left\{-\frac{1}{2\tilde{\beta}_t}\left[\boldsymbol{x}_{t-1}^2-2\frac{\left(\frac{\sqrt{\alpha_t}\boldsymbol{x}_t}{1-\alpha_t}+\frac{\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0}{1-\bar{\alpha}_{t-1}}\right)(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{(1-\bar{\alpha}_t)}\boldsymbol{x}_{t-1}+\tilde{\beta}_tC(\boldsymbol{x}_0,\boldsymbol{x}_t,t)\right]\right\}\\
&=\exp\left\{-\frac{1}{2\tilde{\beta}_t}\left[\boldsymbol{x}_{t-1}^2-2\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t+\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{(1-\bar{\alpha}_t)}\boldsymbol{x}_{t-1}+\tilde{\beta}_tC(\boldsymbol{x}_0,\boldsymbol{x}_t,t)\right]\right\}
\end{aligned} \tag{3.31}$$

The expression is now reminiscent of the expansion of the square $(\boldsymbol{x}-\boldsymbol{\mu})^\intercal(\boldsymbol{x}-\boldsymbol{\mu})=\boldsymbol{x}^\intercal\boldsymbol{x}-\boldsymbol{x}^\intercal\boldsymbol{\mu}-\boldsymbol{\mu}^\intercal\boldsymbol{x}-\boldsymbol{\mu}^\intercal\boldsymbol{\mu}$ in the exponent of eq. (2.14). $\boldsymbol{x}^\intercal\boldsymbol{x}$ coincides with $\boldsymbol{x}_{t-1}^2$, being the sampling variable. Similarly, $-2\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t+\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{(1-\bar{\alpha}_t)}\boldsymbol{x}_{t-1}$ coincides with $-\boldsymbol{x}^\intercal\boldsymbol{\mu}-\boldsymbol{\mu}^\intercal\boldsymbol{x}$, being a mix of the sampling variable and the conditioning variables.

The term $\tilde{\beta}_t C(\boldsymbol{x}_0, \boldsymbol{x}_t, t)$ is used to complete the square, and ensures that $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ integrates to 1, a requirement specified for PDFs in eq. (2.5). Equation (3.31) is therefore proportional to the PDF of

$$\begin{aligned}
&\propto \mathcal{N}\left(\mathbf{x}_{t-1}; \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\mathbf{x}_0}{1-\bar{\alpha}_t}, \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{I}\right) \\
&= \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\mathbf{\Sigma}}_t).
\end{aligned} \tag{3.32}$$

In conclusion, the ground truth reverse transition is a conditional Gaussian with mean and variance

$$\tilde{\boldsymbol{\mu}}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t} \tag{3.33}$$

$$\tilde{\mathbf{\Sigma}}_t = \tilde{\beta}_t\mathbf{I} = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{I}. \tag{3.34}$$

The goal is for the learnable reverse transition to maintain as much similarity with the ground truth reverse transition, making it necessary to model it as a conditional Gaussian. Ho et al. (2020) decide for the variance to not be parameterized, yielding the one shown in eq. (3.34). That leaves a learnable mean function $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ conditioning on $\mathbf{x}_t$ and t, as the ground truth $\mathbf{x}_0$ is not available after training. This results in $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \tilde{\mathbf{\Sigma}}_t)$. To obtain the form Ho et al. (2020) arrive at, one can instead parameterize a $\hat{\mathbf{x}}_{0,\boldsymbol{\theta}}(\mathbf{x}_t, t)$ as a replacement for the mean (Luo, 2022), such that

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{\boldsymbol{x}}_{0,\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{1-\bar{\alpha}_t}. \tag{3.35}$$

Having a network that predicts clean data given an arbitrary latent sounds intriguing, as it gives insight into where the denoising procedure is headed. This idea deserves more thought and will be revisited in section 3.5. Returning to the problem of deriving a parameterization: There exists another formulation for the mean. Recalling the closed form solution for the forward process, eq. (3.16), $\boldsymbol{x}_0$ can be expressed as

$$\begin{aligned}
\boldsymbol{x}_t &= \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \quad \text{where} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \qquad \mid \text{Repetition of eq. (3.16)} \\
\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 &= \boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \\
\boldsymbol{x}_0 &= \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}}{\sqrt{\bar{\alpha}_t}}.
\end{aligned} \tag{3.36}$$

Interestingly yet expected, the linear noising transformation is invertible, meaning $\boldsymbol{x}_0$ can be computed from a latent, its corresponding time step, and the noise that was added. When training on a data set of $\boldsymbol{x}_0$, both $\boldsymbol{x}_t$ and $\boldsymbol{\epsilon}$ are known. Outside of training, only $\boldsymbol{x}_t$ is known. As a consequence, learning $\hat{\boldsymbol{x}}_{0,\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ would mean implicitly learning the noise $\boldsymbol{\epsilon}$ that influenced an $\boldsymbol{x}_0$. Utilizing this insight, the focus can be shifted from predicting pure data towards learning the noise, expressed mathematically as

$$\hat{\boldsymbol{x}}_{0,\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{\sqrt{\bar{\alpha}_t}}. \tag{3.37}$$

Inserting this equation into eq. (3.35), the expression for the predicted mean in the reverse transition can be formulated as

$$
\begin{aligned}
\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) &= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)}{\sqrt{\bar{\alpha}_t}}}{1-\bar{\alpha}_t} \\
&= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t}{1-\bar{\alpha}_t} + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)(\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t))}{(1-\bar{\alpha}_t)\sqrt{\bar{\alpha}_t}} \quad \Bigg| \quad \frac{\sqrt{\bar{\alpha}_{t-1}}}{\sqrt{\bar{\alpha}_t}} = \frac{1}{\sqrt{\alpha_t}} \\
&= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t}{1-\bar{\alpha}_t} + \frac{(1-\alpha_t)(\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t))}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}} \\
&= \frac{\alpha_t(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}} + \frac{(1-\alpha_t)\boldsymbol{x}_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}} - \frac{(1-\alpha_t)\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}} \quad \Bigg| \quad \alpha_t\bar{\alpha}_{t-1} = \bar{\alpha}_t \\
&= \frac{\alpha_t - \bar{\alpha}_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}}\boldsymbol{x}_t + \frac{1-\alpha_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t) \\
&= \frac{\alpha_t - \bar{\alpha}_t + 1 - \alpha_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t) \\
&= \frac{1-\bar{\alpha}_t}{(1-\bar{\alpha}_t)\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t) \\
&= \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)\right),
\end{aligned} \tag{3.38}
$$

which will be of great relevance to the sampling algorithm. More details surrounding this will be given in section 3.4.6.

So far, parameterizations for three separate variables, all of which are used to model the reverse transition, have been derived. They are linearly related in the presence of any $\boldsymbol{x}_t$, meaning that any can be calculated given one of them and the time step t. Ho et al. (2020) derive the loss function with respect to $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$, and will therefore be mainly attended to in this thesis. It should be noted that in some cases, there is still interest in calculating the predicted mean or a prediction of $\boldsymbol{x}_0$. Appropriately, $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ and $\hat{\boldsymbol{x}}_{0,\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ will be used as functions that at their core employs $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$. With these forms in mind, the loss function can be derived.

3.4.4.3 Deriving the loss function

Knowing that both $q(\mathbf{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ and $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ are multivariate Gaussians, the KL divergence term in $\mathcal{L}_{t-1}$ can be simplified using eq. (2.22). Additionally, the distributions share the variance $\tilde{\boldsymbol{\Sigma}}_t$, making the KL divergence more convenient to evaluate. For compactness, the shorthands $\tilde{\boldsymbol{\mu}}$ and $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ are defined for $\tilde{\boldsymbol{\mu}}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathbb{E}_q[\mathbf{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0]$ and

$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \mathbb{E}_{p_{\boldsymbol{\theta}}}[\mathbf{x}_{t-1}|\boldsymbol{x}_t]$ respectively. It follows that

$$\begin{aligned}
\mathcal{L}_{t-1} &= \mathbb{E}_{q(\mathbf{x}_t|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{x}_0) \mid\mid p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t))\right] \\
&= \mathbb{E}_{q(\mathbf{x}_t|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}(\mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t, \tilde{\boldsymbol{\Sigma}}_t) \mid\mid \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}, \tilde{\boldsymbol{\Sigma}}_t))\right] \\
&= \mathbb{E}_{q(\mathbf{x}_t|\boldsymbol{x}_0)}\left[\frac{1}{2}\left[\operatorname{tr}(\tilde{\boldsymbol{\Sigma}}_t^{-1}\tilde{\boldsymbol{\Sigma}}_t) - d + (\boldsymbol{\mu}_{\boldsymbol{\theta}} - \tilde{\boldsymbol{\mu}}_t)^{\mathsf{T}}\tilde{\boldsymbol{\Sigma}}_t^{-1}(\boldsymbol{\mu}_{\boldsymbol{\theta}} - \tilde{\boldsymbol{\mu}}_t) + \log\left(\frac{|\tilde{\boldsymbol{\Sigma}}_t|}{|\tilde{\boldsymbol{\Sigma}}_t|}\right)\right]\right] \\
&= \mathbb{E}_{q(\mathbf{x}_t|\boldsymbol{x}_0)}\left[\frac{1}{2}\left[\operatorname{tr}(\mathbf{I}) - d + (\boldsymbol{\mu}_{\boldsymbol{\theta}} - \tilde{\boldsymbol{\mu}}_t)^{\mathsf{T}}(\tilde{\beta}_t\mathbf{I})^{-1}(\boldsymbol{\mu}_{\boldsymbol{\theta}} - \tilde{\boldsymbol{\mu}}_t) + \log 1\right]\right] \\
&= \mathbb{E}_{q(\mathbf{x}_t|\boldsymbol{x}_0)}\left[\frac{1}{2\tilde{\beta}_t}\left[d - d + (\boldsymbol{\mu}_{\boldsymbol{\theta}} - \tilde{\boldsymbol{\mu}}_t)^{\mathsf{T}}(\boldsymbol{\mu}_{\boldsymbol{\theta}} - \tilde{\boldsymbol{\mu}}_t)\right]\right] \\
&= \mathbb{E}_{q(\mathbf{x}_t|\boldsymbol{x}_0)}\left[\frac{1}{2\tilde{\beta}_t}\left[(\boldsymbol{\mu}_{\boldsymbol{\theta}} - \tilde{\boldsymbol{\mu}}_t)^{\mathsf{T}}(\boldsymbol{\mu}_{\boldsymbol{\theta}} - \tilde{\boldsymbol{\mu}}_t)\right]\right] \\
&= \mathbb{E}_{q(\mathbf{x}_t|\boldsymbol{x}_0)}\left[\frac{1}{2\tilde{\beta}_t}\left[\|\boldsymbol{\mu}_{\boldsymbol{\theta}} - \tilde{\boldsymbol{\mu}}_t\|_2^2\right]\right],
\end{aligned} \tag{3.39}$$

being the squared L2-norm between the predicted mean and the actual mean. Previously it was shown that $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ learns to predict a transformed $\boldsymbol{x}_0$, allowing the reformulation

$$\begin{aligned}
\mathcal{L}_{t-1} &= \mathbb{E}_{q(\mathbf{x}_t|\boldsymbol{x}_0)}\left[\frac{1}{2\tilde{\beta}_t}\left[\|\boldsymbol{\mu}_{\boldsymbol{\theta}} - \tilde{\boldsymbol{\mu}}_t\|_2^2\right]\right] \\
&= \mathbb{E}_{q(\mathbf{x}_t|\boldsymbol{x}_0)}\left[\frac{1}{2\tilde{\beta}_t}\left[\left\|\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{\boldsymbol{x}}_{0,\boldsymbol{\theta}}(\mathbf{x}_t, t)}{1-\bar{\alpha}_t}\right.\right.\right. \\
&\qquad\qquad\left.\left.\left. - \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t}\right\|_2^2\right]\right] \\
&= \mathbb{E}_{q(\mathbf{x}_t|\boldsymbol{x}_0)}\left[\frac{1}{2\tilde{\beta}_t}\left[\left\|\frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{\boldsymbol{x}}_{0,\boldsymbol{\theta}}(\mathbf{x}_t, t) - \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t}\right\|_2^2\right]\right] \\
&= \mathbb{E}_{q(\mathbf{x}_t|\boldsymbol{x}_0)}\left[\frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{2\tilde{\beta}_t(1-\bar{\alpha}_t)^2}\left[\|\hat{\boldsymbol{x}}_{0,\boldsymbol{\theta}}(\mathbf{x}_t, t) - \boldsymbol{x}_0\|_2^2\right]\right].
\end{aligned} \tag{3.40}$$

The objective is identical to measuring the dissimilarity between the predicted pure data and the pure data itself.

Equation (3.37) states that $\hat{\boldsymbol{x}}_{0,\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{\sqrt{\bar{\alpha}_t}}$, suggesting a formulation of the objective comparing the noise and predicted noise:

$$\begin{aligned}\mathcal{L}_{t-1} &= \mathbb{E}_{q(\mathbf{x}_t|\boldsymbol{x}_0)}\left[\frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{2\tilde{\beta}_t(1-\bar{\alpha}_t)^2}\left[\|\hat{\boldsymbol{x}}_{0,\boldsymbol{\theta}}(\mathbf{x}_t,t)-\boldsymbol{x}_0\|_2^2\right]\right]\\ &= \mathbb{E}_{q(\mathbf{x}_t|\boldsymbol{x}_0)}\left[\frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{2\tilde{\beta}_t(1-\bar{\alpha}_t)^2}\left[\left\|\frac{\mathbf{x}_t-\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)}{\sqrt{\bar{\alpha}_t}}-\frac{\mathbf{x}_t-\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}}{\sqrt{\bar{\alpha}_t}}\right\|_2^2\right]\right]\\ &= \mathbb{E}_{q(\mathbf{x}_t|\boldsymbol{x}_0)}\left[\frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2(1-\bar{\alpha}_t)}{2\tilde{\beta}_t(1-\bar{\alpha}_t)^2}\left[\left\|\frac{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)-\boldsymbol{\epsilon}}{\sqrt{\bar{\alpha}_t}}\right\|_2^2\right]\right] \qquad \left|\ \frac{\bar{\alpha}_{t-1}}{(\sqrt{\bar{\alpha}_t})^2}=\frac{1}{\alpha_t}\right.\\ &= \mathbb{E}_{q(\mathbf{x}_t|\boldsymbol{x}_0)}\left[\frac{(1-\alpha_t)^2}{2\tilde{\beta}_t(1-\bar{\alpha}_t)\alpha_t}\left[\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)-\boldsymbol{\epsilon}\|_2^2\right]\right]. \qquad (3.41)\end{aligned}$$

Effectively, the denoising matching term $\mathcal{L}_{t-1}$ compares the predicted and ground truth noise through the squared L2 norm. The function $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ will for the remainder of this thesis be referred to as the noise predictor.

Ho et al. (2020) go one step further in the simplifications of the learning objective by replacing the scaling factor residing outside the L2-norm in eq. (3.41) with the scalar value 1. Although there is no theoretical motivation for this, the authors found it to improve sampling quality through empirical studies. They refer to this re-weighted objective as $\mathcal{L}_{\text{simple}}$, while this thesis denotes it as

$$\mathcal{L}_{t-1}^{1} = \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}\left[1\left[\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)-\boldsymbol{\epsilon}\|_2^2\right]\right]. \qquad (3.42)$$

3.4.5 Training

The noise predictor $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ serves as the foundation for the mean function in $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$, predicting the noise that distinguishes $\mathbf{x}_{t-1}$ from $\mathbf{x}_t$. To obtain a useful model $p_{\boldsymbol{\theta}}(\mathbf{x}_0)$, it must be trained towards this purpose. $\mathcal{L}_0$ and $\mathcal{L}_T$ are irrelevant during training, as the former must be satisfied during sampling, and the latter is already satisfied under the assumptions of an ideal variance schedule. The training algorithm will thus focus on $\mathcal{L}_{t-1}$. Recalling from eq. (3.21), to minimize the divergence between the forward and modeled posterior, the ELBO must be maximized. $\mathcal{L}_{1:T-1}$ contributes a negative value, totalling multiple KL divergences in eq. (3.27). This means each $\mathcal{L}_{t-1}$ should be minimized.

Algorithm 1 shows the general approach of training a DDPM (Ho et al., 2020). From eq. (3.17), $\mathbf{x}_t$ is sampled directly instead of doing the forward process iteratively from 1 through t. $\mathcal{L}_{t-1}$ depends only on the data-conditioned marginal $q(\mathbf{x}_t|\mathbf{x}_0)$, meaning it can be optimized independently from all other denoising matching terms. Ho et al. (2020) implements the squared L2-norm as the mean-squared error over vector elements.

3.4.6 Sampling

Once the noise predictor $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ is trained, the Markov chain can be reversed. The general procedure for sampling using a DDPM is presented in algorithm 2. $\mathcal{L}_0$ is satisfied

Algorithm 1 Training procedure of DDPMs

```
1: Input: Noise predictor $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$, data set distribution $q(\mathbf{x}_0)$
2: repeat
3:     $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
4:     $t \sim \text{Uniform}(\{1, ..., T\})$
5:     $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
6:     $\mathbf{x}_t \leftarrow \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}$
7:     Update $\boldsymbol{\theta}$ on $\nabla_{\boldsymbol{\theta}} \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t,\ t)\|_2^2$
8: until converged
```

by setting the noise correction to $\mathbf{0}$ at $t = 1$. Line 6 then becomes reminiscent of computing the expectation of $p_{\boldsymbol{\theta}}(\mathbf{x}_0|\mathbf{x}_1)$. Ho et al. (2020) found the choice of either $\tilde{\beta}_t = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}$ or β_t as the variance to give similar results, hence the only justification for this thesis choosing the latter being its simplicity. Figure 3.7 visualizes subsets of four sampled trajectories from the generative chain $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})$. The latent $\boldsymbol{x}_T$ is presented in the leftmost column, and the generated $\boldsymbol{x}_0$ is farthest to the right. The samples are generated with one of the DMs trained for this thesis. Details on all models are given in section 4.3.

Algorithm 2 Sampling procedure for DDPMs

```
1: Input: Noise predictor $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$
2: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
3: for $t = T, ..., 1$ do
4:     $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ if $t > 1$, else $\mathbf{z} \leftarrow \mathbf{0}$
5:     $\boldsymbol{\mu}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\right)$        eq. (3.38)
6:     $\mathbf{x}_{t-1} \leftarrow \boldsymbol{\mu}_{t-1} + \sqrt{\beta_t}\mathbf{z}$
7: end for
8: return $\mathbf{x}_0$
```

A limitation of algorithm 2 is that generating a single sample requires T evaluations of the noise predictor. The models used in figs. 3.7 and 3.8 are trained with $T = 1000$, corresponding to 1000 neural network evaluations. Compared to VAEs and GANs, needing only one forward pass, it is clear such a generative model is much less efficient.

From the discussion about the deep generative models and the VAE in sections 3.1.1 and 3.2.2, the importance of deterministic mappings between the latents $\mathbf{z}$ and observed data $\mathbf{x}$ was discussed. In the case of DDPMs, a starting latent $\boldsymbol{x}_T$ will not map to the same $\boldsymbol{x}_0$ because of the noise vector $\boldsymbol{z}$ introduced in line 4 in algorithm 2. Since the generative process of the DDPM reverses the forward process, which is a Markov chain with stochastic transitions, this is no surprise.

Figure 3.8 illustrates the significant impact of the stochastic component when sampling from a DDPM where the target distribution is a univariate mixture of Gaussians (depicted on the right). The latent $x_T = 0$ is fixed for several runs through lines 3 - 8 in algorithm 2.

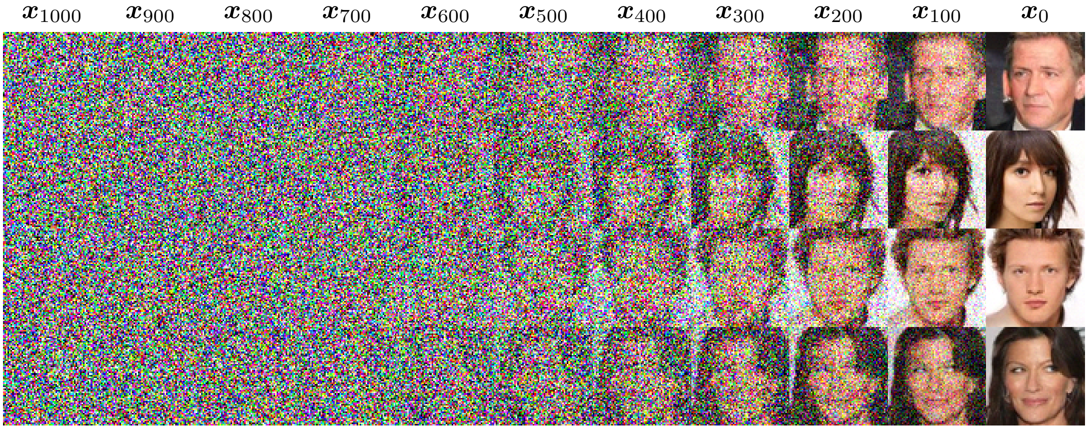


Figure 3.7: Subsamples from the generative process of a DDPM trained on the CelebA data set downscaled to 64×64 with $T = 1000$. $\boldsymbol{x}_{1000}$ is the latent sampled from the prior $p(\mathbf{x}_T)$ and $\boldsymbol{x}_0$ is the generated data. A subset of the latent states $\boldsymbol{x}_t$ for $t \in [1, 1000]$ is shown.

Plotted are the sampled trajectories $\boldsymbol{x}_{0:T-1}$. The figure effectively demonstrates that, from the perspective of a DDPM, the latents $\mathbf{x}_T$ contain little to no information about the resulting $\mathbf{x}_0$. One single latent is capable of covering the whole target distribution. This can be regarded as a shortcoming for a generative model, as it weakens its interpretability, especially for this thesis aiming to find suitable interpretations of the latent space. Fortunately, there exists a solution that alleviates this, as well as the problem of requiring T network evaluations.

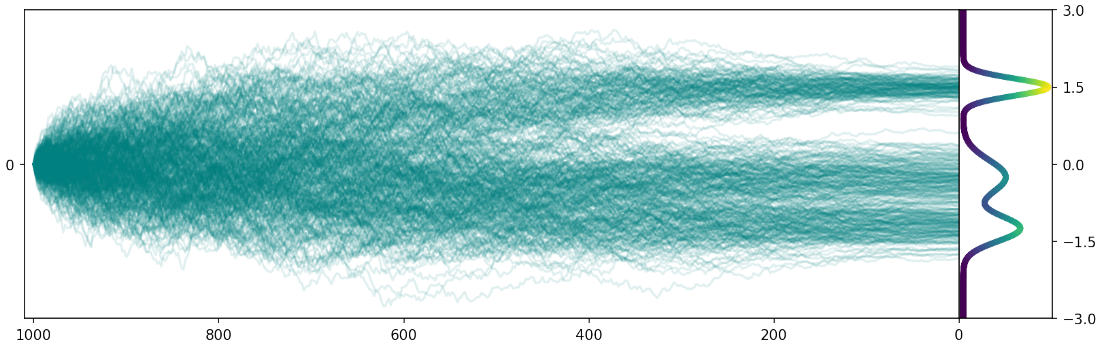


Figure 3.8: 400 sampling trajectories from a one-dimensional DDPM starting from the same latent $x_T = 0$.

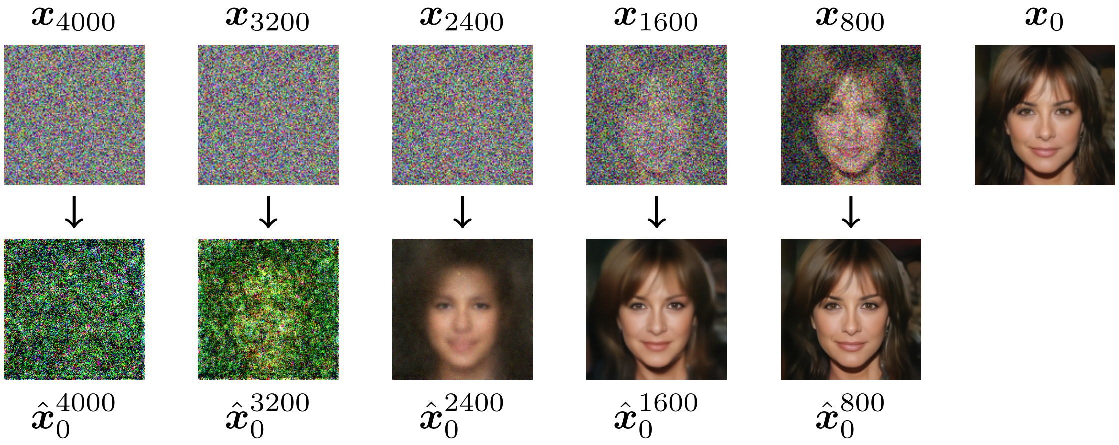

$\boldsymbol{x}_{4000}$
$\boldsymbol{x}_{3200}$
$\boldsymbol{x}_{2400}$
$\boldsymbol{x}_{1600}$
$\boldsymbol{x}_{800}$
$\boldsymbol{x}_{0}$
$\hat{\boldsymbol{x}}_0^{4000}$
$\hat{\boldsymbol{x}}_0^{3200}$
$\hat{\boldsymbol{x}}_0^{2400}$
$\hat{\boldsymbol{x}}_0^{1600}$
$\hat{\boldsymbol{x}}_0^{800}$

Figure 3.9: Using $\hat{\boldsymbol{x}}_{0,\boldsymbol{\theta}}$ from eq. (3.37) to predict where the reverse process is headed. $\hat{\boldsymbol{x}}_0^t$ is used to denote $\hat{\boldsymbol{x}}_{0,\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$. The model is trained with $T = 4000$.

3.5 Denoising Diffusion Implicit Models

While the noise predictor takes part in modeling the reverse of a stochastic diffusion process, it is not responsible for introducing noise when sampling. Remember that the noise predictor serves as a building block for the mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ in the Gaussian reverse $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \beta_t \mathbf{I})$. Therefore, it is reasonable to conclude that the noise predictor has a deterministic stance on exactly what data should emerge from a latent $\boldsymbol{x}_t$. This fact was previously stated mathematically in eq. (3.37), which is a formula for predicting $\boldsymbol{x}_0$ from anywhere in the chain. It is important to emphasize that this formula is not meant as a replacement for the sampling algorithm given for DDPMs. This claim is further strengthened by fig. 3.9, where some predictions from a subset of the latents are shown for a model trained with $T = 4000$. Predictions done for high values of t lack low-level details, appearing blurry and obscure. The more pure data is revealed in the latent, the easier the task becomes. Interestingly, $\hat{\boldsymbol{x}}_{0,\boldsymbol{\theta}}(\boldsymbol{x}_{800}, 800)$ strongly resembles the final sample $\boldsymbol{x}_0$. The figure demonstrates the importance of iteratively refining the data during sampling, giving the model the ability to correct previous mistakes and add detail.

Denoising Diffusion Implicit Models (DDIMs) were introduced by Song et al. (2022) as a direct follow-up to Ho et al. (2020), with their main goal being to increase sampling efficiency while retaining as much output quality as possible. To achieve this, they exploit the ability of the noise predictor to do deterministic predictions from latents to data. This way, they also achieve fully deterministic sampling trajectories.

3.5.1 Alternative forward processes

DDPMs learn the reverse of the forward process, which itself is stochastic. Song et al. (2022) therefore suggest rethinking the forward process to achieve deterministic and faster sampling. Importantly, they are motivated to find a formulation that is compatible with a noise predictor trained through the DDPM framework.

A notable observation, is that the DDPM objective, $\mathcal{L}^1_{t-1}$, is solely dependent on the conditional marginal $q(\mathbf{x}_t|\mathbf{x}_0)$ as seen in eq. (3.42). This distribution is utilized in line 6 from algorithm 1 to generate the training data for the noise predictor. As long as its functional forms stay the same after the reformulation of the forward process, there will be no requirement for retraining the noise predictor, and the same objective can be utilized for training. They reformulate the forward process as

$$q_{\boldsymbol{\sigma}}(\mathbf{x}_{1:T}|\mathbf{x}_0) = q_{\boldsymbol{\sigma}}(\mathbf{x}_T|\mathbf{x}_0)\prod_{t=2}^{T} q_{\boldsymbol{\sigma}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0), \tag{3.43}$$

where $\boldsymbol{\sigma}$ is a vector of elements $\sigma_1, \sigma_2, \dots, \sigma_T$, indexing a family $\mathcal{Q}$ of forward processes over $\mathbf{x}_{0:T}$. Although $q_{\boldsymbol{\sigma}}$ bears similarity with the notation for the learnable distribution $p_{\boldsymbol{\theta}}$, $\boldsymbol{\sigma}$ does not represent learnable parameters, and $q_{\boldsymbol{\sigma}}$ remains a fixed noising process. The role of $\boldsymbol{\sigma}$ will become clear when DDIM sampling is presented.

Recall that for the DDPM framework, Ho et al. (2020) defined $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) = q(\mathbf{x}_t|\mathbf{x}_{t-1})$ and used Bayes' rule to find the reverse $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$. Song et al. (2022) take a contrasting approach by defining the reverse transition directly as

$$q_{\boldsymbol{\sigma}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}, \sigma_t^2\mathbf{I}\right), \tag{3.44}$$

where the mean function is chosen such that $q_{\boldsymbol{\sigma}}(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}) = q(\mathbf{x}_t|\mathbf{x}_0)$. As such, it matches the conditional marginal for DDPMs. Although the forward process $q_{\boldsymbol{\sigma}}(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)$ can be found with Bayes' rule, $q_{\boldsymbol{\sigma}}(\mathbf{x}_t|\mathbf{x}_0)$ is the only requirement for the training objective, and its presence makes the derivation of $q_{\boldsymbol{\sigma}}(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)$ unnecessary. The stochasticity of the reverse transition is controlled by σ_t. As $\sigma_t \to 0$, the distribution becomes degenerate and given any $\mathbf{x}_t$ and $\mathbf{x}_0$, $\mathbf{x}_{t-1}$ is fully deterministic.

The objective for DDIMs is determined from the ELBO (Song et al., 2022), and can be decomposed. The derivation differs only slightly from the DDPM case, and leads to the same decomposition, except that the new forward process is indexed by $\boldsymbol{\sigma}$. The full derivation is given in appendix B, and shows that

$$\begin{aligned}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0) &\geq \mathbb{E}_{q_{\boldsymbol{\sigma}}(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\log\frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})}{q_{\boldsymbol{\sigma}}(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\right] \\ &= \underbrace{\mathbb{E}_{q_{\boldsymbol{\sigma}}(\mathbf{x}_1|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\mathbf{x}_1)\right]}_{\text{reconstruction term } \mathcal{J}_0} - \underbrace{D_{\mathrm{KL}}(q_{\boldsymbol{\sigma}}(\mathbf{x}_T|\boldsymbol{x}_0) \mid\mid p(\mathbf{x}_T))}_{\text{prior matching term } \mathcal{J}_T} \\ &\quad - \sum_{t=2}^{T}\underbrace{\mathbb{E}_{q_{\boldsymbol{\sigma}}(\mathbf{x}_t|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}(q_{\boldsymbol{\sigma}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{x}_0) \mid\mid p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t))\right]}_{\text{denoising matching term } \mathcal{J}_{t-1}}\end{aligned} \tag{3.45}$$

Similarly to the DDPM case, there is a reconstruction term $\mathcal{J}_0$ that is satisfied by computing $\boldsymbol{x}_0$ noiselessly from $\boldsymbol{x}_1$ during sampling. The prior matching term $\mathcal{J}_T$ is satisfied with an ideal variance schedule. Lastly, the denoising matching term is a sum over terms similar to $\mathcal{L}_{t-1}$, hence it is referred to as $\mathcal{J}_{t-1}$. The form of $q_{\boldsymbol{\sigma}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ is known (eq. (3.44)), making it again necessary to assign an identical form to $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$. As a consequence, simplifying the KL divergence in $\mathcal{J}_{t-1}$ when the variance is shared, leads to a scaled L2-norm between the mean functions

$$\begin{aligned}\mathcal{J}_{t-1} &= \mathbb{E}_{q_{\boldsymbol{\sigma}}(\mathbf{x}_t|\mathbf{x}_0)}\left[D_{\mathrm{KL}}\left(q_{\boldsymbol{\sigma}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)\ ||\ p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)\right)\right] \\ &= \mathbb{E}_{q_{\boldsymbol{\sigma}}(\mathbf{x}_t|\mathbf{x}_0)} = \left[\frac{1}{2\sigma_t^2}\left[\|\boldsymbol{\mu}_{\boldsymbol{\theta}}^{\sigma} - \boldsymbol{\mu}_t^{\sigma}\|_2^2\right]\right], \end{aligned} \tag{3.46}$$

where σ in the superscript highlights the alternative formulation proposed by Song et al. (2022). With identical forms, the two means will further simplify to a comparison between the predicted and ground truth noise, with a different scaling factor compared to DDPM. Song et al. (2022) show that the reformulated $\mathcal{J}_{t-1} = \mathcal{L}_{t-1}^{\gamma} + C$, for some values of γ and a constant C, where γ denotes the reweighting of the original $\mathcal{L}_{t-1}$ objective. When performing gradient operations on this objective, the constant C is disregarded. Recall from eq. (3.42) that Ho et al. (2020) define $\gamma = 1$ when training DDPMs. Fascinatingly, for this value of γ, it holds that a model optimizing the DDPM objective (eq. (3.42)), will also optimize the DDIM objective (eq. (3.46)). The consequence is that the DDPM training algorithm, algorithm 1, can be used to train the noise predictor but enables rethinking in light of eq. (3.44) how to use it. This is discussed next.

3.5.2 Sampling

The variance $\boldsymbol{\sigma}$ is a function of a scalar parameter $\eta \geq 0$ (Song et al., 2022), shown through

$$\sigma_t(\eta) = \eta\sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}}\sqrt{1-\alpha_t}. \tag{3.47}$$

When $\eta = 0$, sampling becomes fully deterministic. For the case of $\eta = 1$, the authors state that the forward process becomes stochastic, and the generative process $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})$ resolves to a DDPM. Although used as a term for a deterministic DMs, DDIM also generalizes to DDPM because of this property. Conveniently, this generalization still allows for accelerated sampling, meaning efficient stochastic sampling is enabled. Using $\eta \in \langle 0, 1\rangle$ resolves to stochastic outputs, but allows for a degree of consistency and variation from when $\eta = 0$. This case is not used for this thesis.

Song et al. (2022) point out an additional effect of the $\mathcal{L}_{t-1}^{1}$ objective not relying on an iterative scheme, but instead only on $q(\mathbf{x}_t|\mathbf{x}_0)$. It enables the forward process to also be defined on a subset $\{\mathbf{x}_{\tau_1}, \dots, \mathbf{x}_{\tau_n}\}$ of the latent variables, where $\tau_{1:n}$ is any increasing sequence of values from the set $\{0, \dots, T\}$. In a similar fashion where data $\mathbf{x}_0$ can be predicted from $\mathbf{x}_t$, the reformulation of the reverse transition allows for predicting a latent $\mathbf{x}_{\tau_{i-1}}$ from $\mathbf{x}_{\tau_i}$. Their empirical studies indicate that DDIM allows for generating data through 20 to 100 steps without decreasing the quality of the generated sample. This is a 10× to 50× speed-up compared to the base DDPM algorithm, which typically

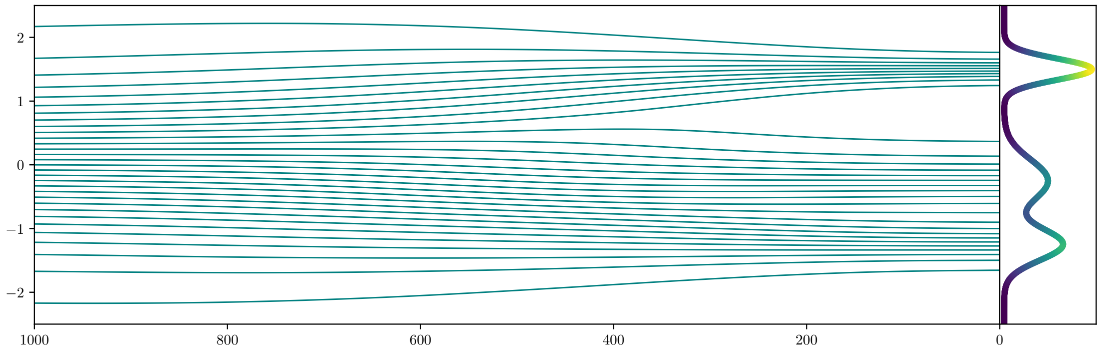


Figure 3.10: DDIM trajectories when $\eta = 0$.

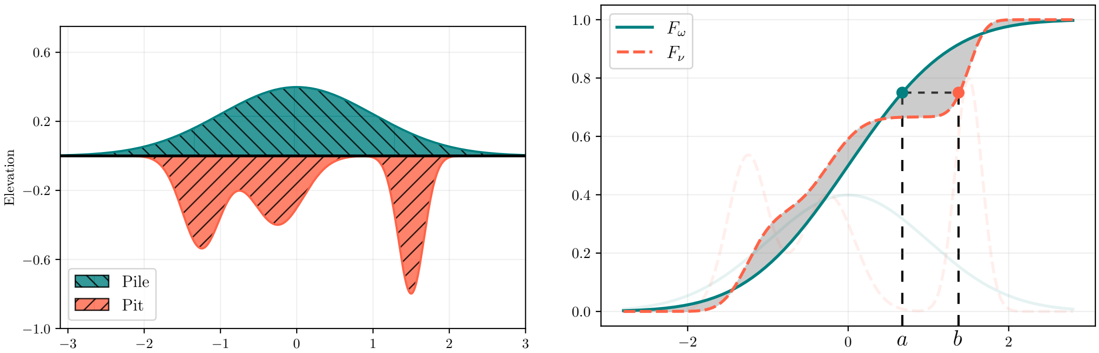
uses $T = 1000$. Algorithm 3 illustrates DDIM sampling, where the variance vector $\boldsymbol{\sigma}$ is controlled by η through eq. (3.47).

Algorithm 3 Sampling procedure for DDIM

```
**Input:** Noise predictor $\boldsymbol{\epsilon_\theta}$, stochasticity parameter $\eta$, decreasing sequence of time steps $T = \tau_n > \tau_{n-1} > \cdots > \tau_1 = 0$
$\mathbf{x}_{\tau_n} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
**for** $i = n, \ldots, 2$ **do**
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ if $i = 2$, else $\mathbf{z} \leftarrow \mathbf{0}$
    $\sigma_{\tau_i} \leftarrow \eta \sqrt{\frac{1-\bar{\alpha}_{\tau_{i-1}}}{1-\bar{\alpha}_{\tau_i}}} \sqrt{1 - \alpha_{\tau_i}}$
    $\hat{\boldsymbol{\epsilon}} \leftarrow \boldsymbol{\epsilon_\theta}(\mathbf{x}_{\tau_i}, \tau_i)$
    $\hat{\mathbf{x}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_{\tau_i}}} (\mathbf{x}_{\tau_i} - \sqrt{1 - \bar{\alpha}_{\tau_i}} \hat{\boldsymbol{\epsilon}})$
    $\hat{\boldsymbol{\epsilon}} \leftarrow \sqrt{1 - \bar{\alpha}_{\tau_{i-1}} - \sigma^2_{\tau_i}} \cdot \hat{\boldsymbol{\epsilon}}$ if $i = 2$, else $\hat{\boldsymbol{\epsilon}} \leftarrow \mathbf{0}$
    $\mathbf{x}_{\tau_{i-1}} \leftarrow \sqrt{\bar{\alpha}_{\tau_{i-1}}} \hat{\mathbf{x}}_0 + \hat{\boldsymbol{\epsilon}} + \sigma_{\tau_i} \mathbf{z}$
**end for**
**return** $\mathbf{x}_0$
```

Song et al. (2022) state that "samples are uniquely determined from latent variables", meaning that deterministic DDIM sampling is an injective function. Figure 3.10 visualizes some trajectories that emerge when sampling with $\eta = 0$ on one-dimensional data. The starting latents $\boldsymbol{x}_T$ are positioned evenly in Gaussian space, and their mappings manifest that DDIM generates a distribution that aligns with the target distribution. Furthermore, no trajectories intersect. If two trajectories were to intersect at time step t, the deterministic property would ensure that they were mapped to the same point x_0, confirming the injective property.

The reformulation of the mean function in the denoising transition $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$ makes the model able to perform denoising even in the case where the variance is set to 0, as is the case when $\eta = 0$. If attempted to set the variance to 0 during DDPM sampling, it would fail to cover the target distribution. Appendix C demonstrates this case when



(a)

(b)

Figure 3.11: **(a)** A dirt pile (teal) and multiple dirt pits (red). **(b)** Optimal transport mapping using the CDF and its inverse.

sampling from the same latents as in fig. 3.10.

The non-intersecting DDIM trajectories in the one-dimensional case imply that the order of latents coincides with the order of data. In practical terms, the model performs the minimum amount of change to each latent on its path to the target distribution. Khrulkov et al. (2022) study this exact ability, tying the DDIM map to the optimal transport map.

3.6 Optimal Transport

Consider some flat terrain represented by a one-dimensional line at elevation $y(x) = 0$, as illustrated in fig. 3.11a. On this field are some dirt pits (red) that need to be filled. Behind it is a dirt pile (teal), assumed to have a total area equal to the empty area in the pits. What is the most efficient way to distribute the dirt from the pile into the pits? The elevation indicates differences in probability mass, or density, at specific points. Points of higher density are more expensive to transport, so optimally, the distance moved for these should be minimized.

Optimal transport is a field in mathematics concerned with finding the most efficient way of distributing mass from one probability distribution to another (Villani, 2008). There exist many transport plans, and most of them are inefficient. Optimality is achieved if it minimizes a total cost C. The cost can be influenced by several factors, however, this thesis only considers cost functions related to distance metrics, referred to as $d(\boldsymbol{a}, \boldsymbol{b})$ where $\boldsymbol{a}$ is a point in the sample space of the first distribution, and $\boldsymbol{b}$ for the second.

The concept is best demonstrated in one dimension as there exists an elegant approach using standard tools from probability theory. Assume two univariate distributions ω and ν with their respective PDFs $\omega(x)$ and $\nu(x)$, and CDFs $F_\omega(x)$ and $F_\nu(x)$, with sample spaces Ω_ω and Ω_ν. This situation can be viewed through the dirt pit analogy, and the task remains to distribute density from one to the other, in this case from ω to ν. As such, the two distributions in fig. 3.11a are suitable reference points.

At the median point x'_ω where $F_\omega(x'_\omega) = 0.5$, ω is divided into two halves. This means that its PDF has an equal amount of density on the left as on the right of x'_ω. Similarly, the median point x'_ν of ν gives an equivalent scenario. By intuition, density on the left of x'_ω should be transported to the corresponding left side of x'_ν. If done the other way around, meaning density in the left partition of ω is moved to the right partition of ν, a greater distance would be traveled for both groups of densities. In the discrete case with only two quantiles, this illustrates the optimal transport map.

Consider the case where the distributions are divided into four quantiles. Again, each quantile has an equal amount of density contained within it. Starting from the first quantile of ω, referred to as $Q1_\omega$, its optimal transportation is to $Q1_\nu$. Regarding this as a recursive problem, there remain three quantiles to transport, and $Q2_\omega$ is best matched with $Q2_\nu$. As such, optimal transport in the one-dimensional case preserves the order of quantiles.

This approach applies to the general case with n quantiles, visualized in fig. 3.11b. Since each quantile in both distributions encompasses an equivalent amount of mass, it follows that $F_\omega(a) = F_\nu(b)$ where a and b represent the matching quantile boundary in terms of order. The point that a maps to is obtained through the CDF of ω and the quantile function of ν, such that

$$b = F_\nu^{-1}(F_\omega(a)). \tag{3.48}$$

The distance traversed is calculated as $|b - a|$ for that specific point. In one dimension, the total transportation cost can be calculated either from the viewpoint of the CDF, or the quantile function

$$C(\omega, \nu) = \int_{\Omega_\omega} |F_\omega(x) - F_\nu(x)| dx = \int_0^1 \left| F_\omega^{-1}(q) - F_\nu^{-1}(q) \right| dq, \tag{3.49}$$

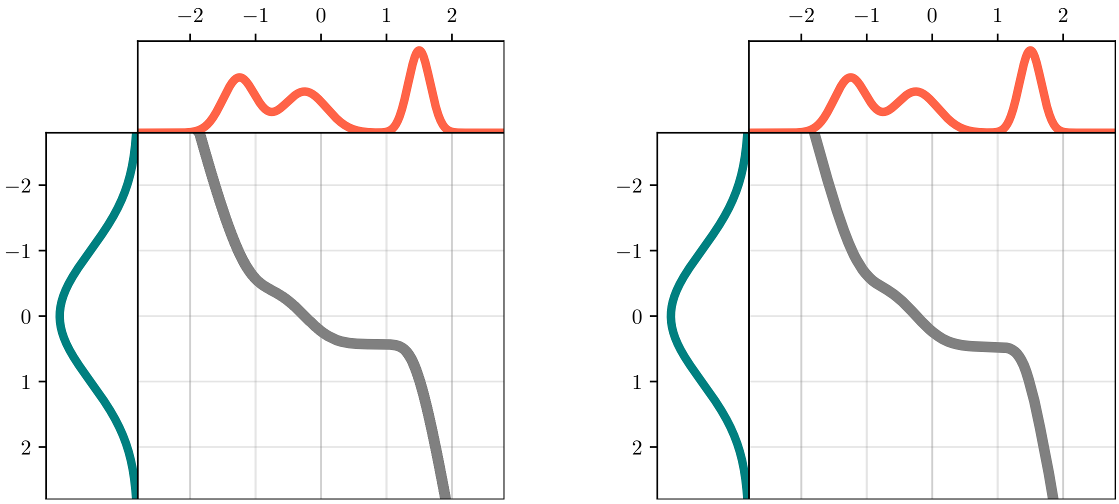
where dq is an infinitesimal quantile, and assuming $\Omega_\omega = \Omega_\nu$.

Figure 3.12a visualizes the optimal transport map defined through eq. (3.48), as a continuous function for all points in the sample space. More specifically, this is the Monge-formulation, defining an injective function $M : \Omega_\omega \to \Omega_\nu$, mapping atomic points between two distributions (Peyré and Cuturi, 2020).

The optimal transport problem is generalized through the Wasserstein distance, whose minimization results in the optimal transport map. It is often called the Earth Mover's distance since the dirt pit analogy is commonly used when exemplifying it. For the n-dimensional case, it is given by

$$\mathcal{W}_p(\omega, \nu) = \min_M \left(\int_{\Omega_\omega} d(\boldsymbol{x}, M(\boldsymbol{x}))^p \; \omega(\boldsymbol{x}) \; d\boldsymbol{x} \right)^{\frac{1}{p}}, \tag{3.50}$$

where p defines the Wasserstein order. Equation (3.49) is then the $\mathcal{W}_1$ distance, with the cost function being the absolute difference between two sample points (Peyré and Cuturi, 2020).



(a)

(b)

Figure 3.12: Optimal transport maps **(a)** found through analytical means, and **(b)** learned by DDIM.

Khrulkov et al. (2022) suggest that DDIM learns the optimal transport map between the prior and the data distribution it trains on by indirectly minimizing the $\mathcal{W}_2$ distance. They conducted experiments in 2, 3, and 7 dimensions, comparing the Wasserstein distance calculated from DDIM with the empirical Wasserstein distance found using the `Python Optimal Transport` library. For all cases, they obtain errors at the level of machine precision. To support their claim, fig. 3.12b shows the DDIM map when trained on the distribution in the upper panel. The prior is depicted in the left panel. The two maps presented in fig. 3.12 are highly similar.

Chapter 4

Architecture and Implementation

This chapter provides details about the data sets used for training and the implementation of the models in this thesis. Section 4.1 explains the typical neural network architecture used for DMs modeling images. Section 4.2 provides an overview of the data sets used. Section 4.3 details the hyperparameters for the models used in the experiments, which also include those utilized for figures presented outside the experiments. A common theme for all sections is the separation between low- and high-dimensional data, where *low* refers to data in one or two dimensions, and *high* refers to images.

4.1 Architecture

In regards to the network architecture, the only requirement for the noise predictor $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ is that the dimensionality of the output vector is the same as the input $\boldsymbol{x}_t$. Consequently, it allows for flexibility when designing the layers that fall between. The network also requires information about which time step t the input $\boldsymbol{x}_t$ is corrupted at, such that $\mathbf{x}_{t-1}$ is correctly modeled.

Ho et al. (2020) employ the U-Net for their experiments, an architecture originally used to segment biomedical images (Ronneberger et al., 2015). As the name suggests, its structure forms the letter 'U'. From the start, the input to the neural network is incrementally reduced along the spatial dimensions through a series of down-sampling blocks. Simultaneously, the number of channels is increased. The bottom of the 'U' – corresponding to the middle layer – represents a bottleneck, where the spatial dimensions reach a minimum and the channel dimension a maximum. Following the bottleneck is a series of up-sampling blocks, essentially mirroring the down-sampling part of the network. Importantly, the U-Net has skip connections between the down-sampling and up-sampling of the same spatial resolutions across the bottleneck. The first convolution in each up-sampling block therefore works on a concatenation of the current activation and the one from the corresponding down-sampling block. The number of down-sampling blocks in the U-Net specifies the number of *levels*.

An illustration of the original architecture is presented in fig. 4.1. Although the specific details shown in the figure do not perfectly align with the implementation used in this thesis, it serves as a point of reference.

To incorporate knowledge about the time step into the U-net, Ho et al. (2020) propose the use of sinusoidal embeddings as first introduced by Vaswani et al. (2017) for use in the

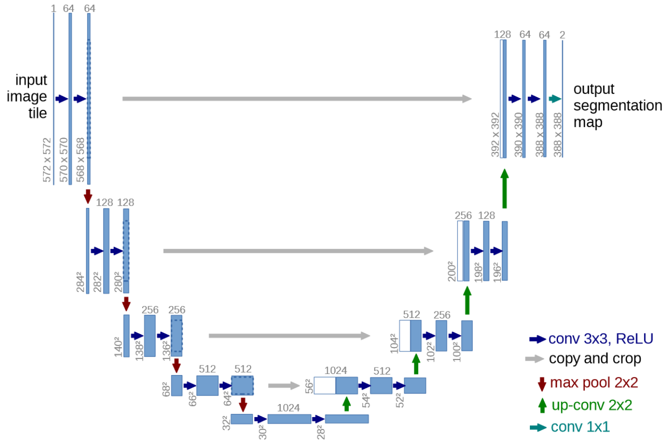


Figure 4.1: The U-Net as illustrated by Ronneberger et al. (2015).

Transformer architecture to specify token positions. The work done for this thesis used a U-Net architecture inspired by dome272 on GitHub[1]. Here, the sinusoidal embeddings undergo a learnable transformation before they are added to the final activation of each down- and up-sampling block. Ho et al. (2020) employ self-attention layers on some blocks. However, empirical studies done in this thesis found it not to be necessary, and it is therefore not implemented.

The U-Net is only relevant for high-dimensional data such as images. When modeling low-dimensional data, a simple multi-layer perceptron was found to be sufficient. The time step is fed as an additional input neuron to the first layer. All models use the same architecture, where the linear layers are followed by Leaky Rectified Linear Unit (ReLU) activation functions.

As mentioned in section 3.4.3, the variance schedule proposed by Ho et al. (2020) is a linear interpolation between $\beta_1 = 10^{-4}$ and $\beta_T = 0.02$. This approach has been criticized for not balancing noise and data well (Nichol and Dhariwal, 2021). As such, the field has seen many variants. Actively used for this thesis is one commonly known as the *scaled linear* schedule used by Rombach et al. (2022) for the first public release of Stable Diffusion[2]. This schedule interpolates linearly between the square root of the endpoints before squaring the whole sequence. Given input endpoints β_1 and β_T, $\tilde{\boldsymbol{\beta}}$ is a linearly increasing sequence going from $\sqrt{\beta_1}$ to $\sqrt{\beta_T}$. To obtain the variance schedule, each element is squared such that $\boldsymbol{\beta} = \tilde{\boldsymbol{\beta}} \odot \tilde{\boldsymbol{\beta}}$, where $\odot$ is element-wise multiplication. Consequently, $\bar{\boldsymbol{\alpha}}$ becomes less steep and data is corrupted at a more stable rate.

[1] Link to the repository: github.com/dome272/Diffusion-Models-pytorch

[2] Link to code snippet: github.com/CompVis/stable-diffusion/ldm/modules/diffusionmodules/util.py

4.2 Data sets

In the case of low dimensions, data sets are dynamically generated at the initiation of each training session. To obtain complex distributions, such as the S-curve, the `datasets` module from scikit-learn (Pedregosa et al., 2011) is utilized. Furthermore, the `distributions` module from PyTorch (Paszke et al., 2019) is employed for generating mixtures of Gaussians in arbitrary dimensions. Exact generation procedures are provided in the `get_datasets`-module available in this thesis's repository.

For high dimensions, only computer vision data sets a used. Following is a list of the data sets, with details on who collected them and where they were downloaded from:

Animal Faces High Quality (AFHQ): Consisting of 16 130 images of cats, dogs, and wild felines, AFHQ was collected by Choi et al. (2020). A download script available on their project repository was utilized[3].

CelebFaces Attributes (CelebA): The CelebA data set consists of 202 599 human faces and is collected by Liu et al. (2015). It was downloaded through the `datasets` module from TorchVision[4].

CelebA High Quality (CelebA-HQ): As the name suggests, CelebA-HQ is an upscaled version of CelebA created by Karras et al. (2018). Appendix C of their publication gives details on how it was generated. CelebA-HQ contains the 30 000 best upscalings at a resolution of 1024 × 1024. A re-sampled edition to 256 × 256 available on Kaggle[5] was used for this thesis.

4.3 Models

This section aims to present an overview of the models used in this thesis. The inclusion of hyperparameter details makes it valuable for readers interested in training their own DM. Training of models operating on low-dimensional data was conducted on a mid-range laptop from 2018. These training sessions typically achieved acceptable performance after around 10 minutes, but were trained more to ensure stability. For high dimensional data, models were trained on the IDUN cluster, maintained by the Norwegian University of Science and Technology (Själander et al., 2019). Training duration for these models was on the magnitude of weeks, not exceeding one month.

Tables 4.1 and 4.2 display training hyperparameters for high- and low-dimensional models, respectively. The arrows in the tables indicate modifications made during training. The majority of hyperparameter choices are based on the experiments by Ho et al. (2020). Table 4.3 specifies the parameters used for the variance schedule, and table 4.4 provides an overview of where in this thesis the models have been used. Details on parameters for the U-Net, as well as instructions for training the models, are provided in this thesis's repository at `github.com/willdalh/diffusion-ot`.

[3] Link to the download script: github.com/clovaai/stargan-v2/blob/master/download.sh

[4] Link to the module: pytorch.org/vision/stable/datasets.html

[5] Data set link: kaggle.com/datasets/badasstechie/celebahq-resized-256x256

Model name	Data set	Epochs	Batch size	LR
Celeb256	CelebA-HQ	4540	$32 \rightarrow 64$	$2 \cdot 10^{-5}$
Celeb64	CelebA	320	128	$0.8 \cdot 10^{-4}$
AFHQ256	AFHQ	4500	32	$0.0002 \rightarrow 2 \cdot 10^{-5} \rightarrow 4 \cdot 10^{-6}$
AFHQ256Exp1	AFHQ	1100	16	$0.8 \cdot 10^{-4}$
AFHQ256Exp2	AFHQ	1820	32	$0.8 \cdot 10^{-4}$

Table 4.1: Training hyperparameters for models trained on high-dimensional data.

Model name	Data set	Epochs	Batch size	LR
Low1DMix	Mixture of Gaussians	90	1024	0.0003
Low2DSymMix	Mixture of Gaussians	75	1024	0.0003
Low2DSymMix	Mixture of Gaussians	70	1024	0.001
Low2DUnimodal	Gaussian	20	1024	0.003
Low2DBimodal	Mixture of Gaussians	75	1024	0.001
Low2DSCurve	Transformed S-curve	35	1024	0.0003

Table 4.2: Training hyperparameters for models trained on low-dimensional data.

Model name(s)	Schedule	T	β_1	β_T
Celeb256 and AFHQ256	Scaled linear	4000	0.00085	0.012
Celeb64	Linear	1000	0.0001	0.02
AFHQ256Exp(1|2)	Linear	4000	0.0001	0.02
Low1DMix	Scaled linear	1000	$2.5 \cdot 10^{-5}$	0.005
Low2DSymMix, Low2DAsymMix, Low2DUnimodal, and Low2DBimodal	Scaled linear	1000	0.0001	0.02
Low2DSCurve	Linear	1000	0.0001	0.02

Table 4.3: Variance schedule parameters.

Model name(s)	Usage references
Celeb256	Experiment group H.2, figs. 3.9 and 5.1, and appendix E
Celeb64	Figure 3.7
AFHQ256	Exp. H.1b and figs. 3.1, 3.2 and 3.4
AFHQ256Exp(1|2)	Exp. H.1a
Low1DMix	Figures 3.8, 3.10 and 3.12b. Variance schedule used for fig. 1.1
Low2DSymMix	Exp. L.1a
Low2DASymMix	Exp. L.1b
Low2DUnimodal	Exp. L.2a
Low2DBimodal	Exp. L.2b
Low2DSCurve	Exp. L.2c and appendix D

Table 4.4: Usage of models in experiments and figures.

Chapter 5

Experiments and Results

Through section 3.6, it was established that sufficient training leads to an alignment between the DDIM map and the optimal transport map from the prior to the target distribution. This chapter investigates the implications of this empirical finding for various data sets. Section 5.1 describes the prerequisites needed for performing the experiments, while section 5.2 provides details on the experimental setup and the assumptions made for the experiments.

5.1 Experimental Plan

DMs are compatible with any data representable in vector form, regardless of the size. To study how optimal transport relates to DMs, a reasonable plan is to separate experiments based on their dimensionality since it determines how results are best presented and what understanding can be formed. In lower dimensions, observations are more verifiable from theory and can provide insight and, potentially, intuition on how optimal transport is transferable to higher dimensions. Additionally, experiments that are of interest in higher dimensions may not be relevant in lower dimensions, and vice versa. An example is studying the effect of using the same initial latent across models, where the information richness in high-dimensional data allows for more discussion, compared to a two-dimensional point.

5.2 Experimental Setup

As stated in section 3.5, DDIM is a generalization over both stochastic and deterministic sampling. As it provides accelerated sampling for both cases, drastically reducing necessary computation resources, it will be used exclusively for all experiments. Song et al. (2022) state that the case when $\eta = 1$ is equivalent to DDPM sampling, meaning all observations and conclusions on accelerated DDPM sampling applies to the original case. For clarity, when used in the context of generating data, DDIM refers to accelerated deterministic sampling with $\eta = 0$, and DDPM refers to accelerated stochastic sampling with $\eta = 1$.

This leaves a question on the number of time steps to be used. Song et al. (2022) provided insight into how the number of sampling steps affects output quality. They mostly observed it to affect the level of detail, where larger latent skips lead to blurry results.

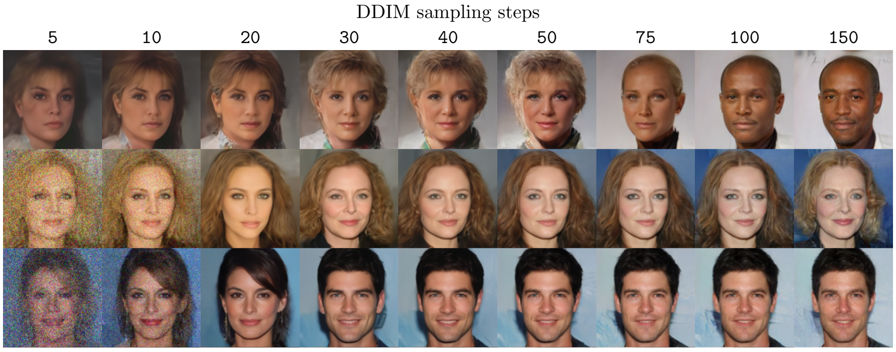


Figure 5.1: DDIM output for fixed latents (rows) with different number of sampling steps (columns).

For this thesis, empirical studies uncovered additional unwanted effects when adjusting the step size. Figure 5.1 demonstrates how output varies depending on the number of sampling steps. The same latent maps to different outputs, where even significant features like gender are altered. This poses a threat to the desirable determinism. To cope with this, the results presented in each atomic experiment use the same number of time steps, typically in the 20-40 range.

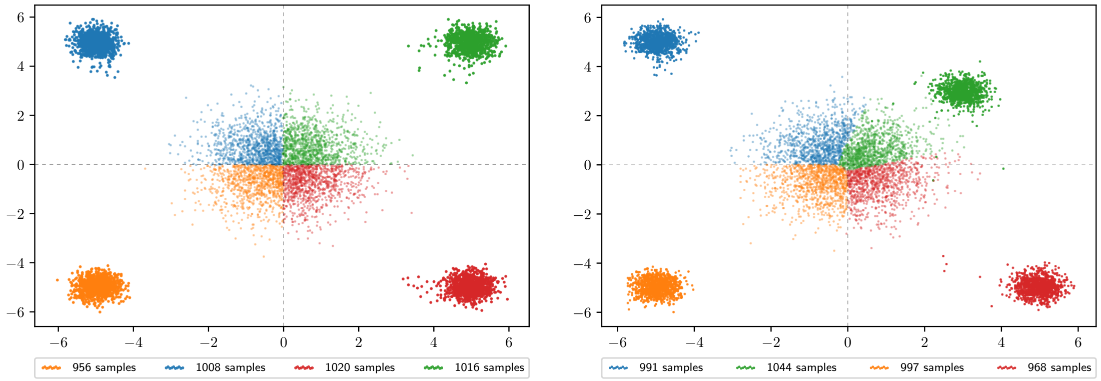
To presentably separate the experiments, a prefix will be used on the numbering system to denote whether they are done on low- or high-dimensional data. For the former, the letter ‘L’ is used, all contained in section 5.3. Correspondingly, the letter ‘H’ denotes experiments on high dimensionality and are presented in section 5.4. Each section contains two groups of experiments. For instance, L.1 refers to a group of experiments on low-dimensional data. Contained within is for example L.1a. Table 4.4 specifies which models are used for the experiments.

5.3 Experiments on Optimal Transport: Low dimensionality

Experiment group L.1 studies how a model must balance finding shortest paths and distributing probability mass correctly, followed by experiment group L.2, studying the transformations that make up the optimal transport map.

Experiment group L.1 - Minimization of Euclidean distance

An important aspect of optimal transport is moving mass in the least costly manner, in this case measured as the smallest total Euclidean distance, determined through empirical studies by Khrulkov et al. (2022). To study this in isolation, a suitable target distribution is one where there is little ambiguity regarding how mass should be dispersed from the prior across all modes.



(a)

(b)

Figure 5.2: Samples from the prior (middle cluster) and DDIM (four outer clusters). The assigned colors denote from where in the prior the samples emerged. Shown is a **(a)** symmetric target and an **(b)** asymmetric target.

Experiment L.1a - Symmetric mixture of Gaussians

This experiment uses a model trained on a data distribution that is symmetric along both its two dimensions. More specifically, it is a bivariate mixture of Gaussians with four modes positioned in their own quadrant, sharing the covariance matrix $\mathbf{\Sigma} = 0.1 \cdot \mathbf{I}$.

Hypothesis 1. *Samples in each mode will originate from latents positioned in the same quadrant.*

Backing the hypothesis is the fact that the prior is centered around the origin, such that an equal amount of latents are expected to reside in each quadrant. Figure 5.2a shows five clusters of points. The largest one and located in the middle is the prior. The remaining four clusters make up the symmetric mixture of Gaussians, that is, the target distribution. The color assigned to each point signifies from where in the prior it originates. For example, the red points in the prior were transported to the red points in the target distribution.

Upon immediate observation, it becomes evident that the colors are separated by the faint gray lines representing the x- and y-axis, supporting hypothesis 1. There are some instances where points appear in unexpected quadrants, which can be argued to be a consequence of suboptimal training. It should be noted that there is no interplay between latents during the sampling process. Each point is sampled in an i.i.d. manner, meaning that, for example, the presence of blue points in the latents does not affect where the orange points will end up. This experiment simultaneously verifies from the sample counts given in the figure that DDIM is roughly able to distribute the points evenly from the prior to the data distribution.

Experiment L.1b - Asymmetric mixture of Gaussians

The results from the symmetric setup in exp. L.1a align with the intuitive notion of the shortest path from any given latent to the target distribution. An interesting case of study is how a model behaves when the target distribution is asymmetric, leading to the following hypothesis:

Hypothesis 2. *The model will learn an optimal transport map that leverages the closeness of the upper right mode.*

To investigate this, a model is trained on a variant of the data set where the mode in the upper right quadrant is positioned closer to the origin. Results are presented in fig. 5.2b. Although hard to distinguish, this model distributes the latents in accordance with a different plan than in exp. L.1a. It exploits the closeness of data density by reaching deeper into the center of the prior when gathering latents to transport to the upper right mode. This way, the model is able to move points from the densest area with a lower cost, reinforcing hypothesis 2.

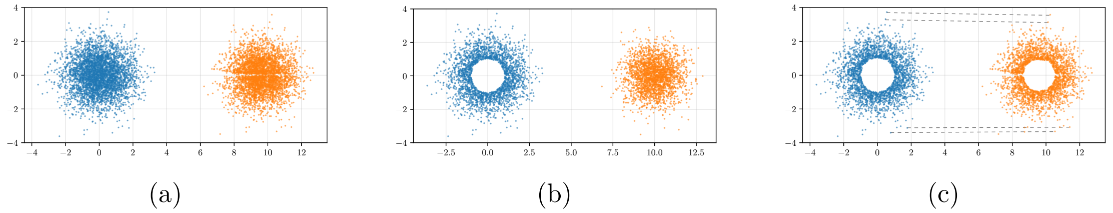
An asymmetric target distribution leads to an asymmetric division in the prior, no longer following the faint gray lines. Upon initial observation, it appears that the model tends to overlook paths that are short. For example, certain red latents situated in the upper right quadrant are closer to the green mode. Despite this, the model has erroneously assigned those to the mode in the lower right quadrant. Recalling from fig. 3.10, DDIM was shown to model a distribution that, to its fullest extent, follows the target distribution. This property is observed in this experiment by how it strives for all modes to receive an equal number of points since they are all of equal density. The model has not fully attained this, as indicated by the imbalanced sample counts. Again, this outcome could be attributed to insufficient training conditions. Now that the model transports more latents from the center of the prior to the upper right mode, it is expected that some prior outliers in the upper right quadrant become available for other modes.

Evaluation

Exp. L.1a verifies that DDIM satisfies the property of learning the shortest path, matching the intuitive notion when done on the symmetric setup. Experiments L.1a and L.1b uncover another important aspect of optimal transport, showing how DDIM must balance minimizing Euclidean distance and distributing latents correctly. The optimal transport map learned in each experiment depicts different sets of transformations, a topic that will be explored further in experiment group L.2.

Experiment group L.2 - Density transportation

This set of experiments studies the transformations that make up the optimal transport map, achieved by masking parts of the prior to observe their destination.



(a) (b) (c)

Figure 5.3: An x-shifted variant of the prior (orange) generated by a DM from the prior (blue). **(a)** DDIM samples from the prior. **(b)** DDPM samples from outliers. **(c)** DDIM matches densities such that outliers become outliers. Some trajectories are shown.

Experiment L.2a - Shifted prior

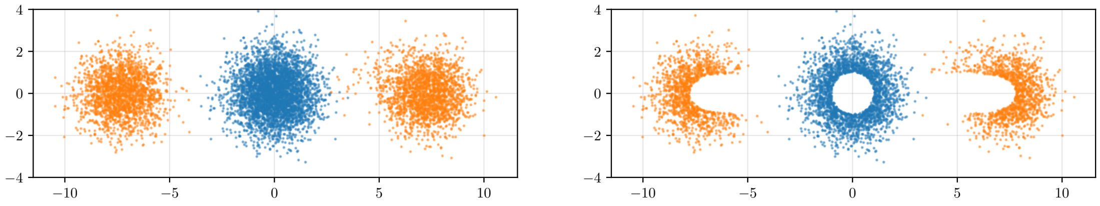
While not resulting in any exciting sample perceived in isolation, one can train a DM on a shifted variant of the prior. This is particularly useful as it provides insight into how certain densities are transported, becoming easier to analyze when the densities are highly similar.

Hypothesis 3. *The DDIM optimal transport map will be a translation when trained on a shifted prior.*

The hypothesis is based on the fact that the target distribution and the prior have the same relative densities, but are positioned differently. This experiment uses a model trained on a bivariate standard Gaussian with shifted mean of 10 units along the x-axis. The prior and target are depicted in fig. 5.3a, being the left and right clusters respectively.

Figure 3.8 demonstrated that DDPM is stochastic to the degree that a single latent is capable of covering the entire target distribution when sampled from multiple times. The same effect presents itself in fig. 5.3b, where outliers of the prior further than one standard deviation from the mean are sampled from, still resulting in full coverage of the target distribution. This information serves as a prerequisite for interpreting the next case. When sampling on the same subset of latents using DDIM, as shown in fig. 5.3c, the densities are fully matched, to the extent that one can notice direct mappings between certain points upon close inspection. Some trajectories are highlighted to support this.

The results are better analyzed after reflecting on what the optimal transport map would be. The mean of the target distribution is located to the right of the prior, but at the same height. Since the cost function is related to the Euclidean distance, a reasonable plan is to keep the y-coordinate fixed throughout transportation. As a consequence, outliers located at the highest point in the prior will not be moved to the area of the greatest density. If this was the plan, one would have to compensate by filling outliers in the target distribution with points originating from denser areas in the prior. As such, data is only transported along the x-axis, verifying the hypothesis. The outliers from the prior that was used as latents are transported to become outliers in the target distribution. The transformation observed is affine, preserving parallel lines. It also preserves distances, a statement that supports hypothesis 3.



(a)

(b)

Figure 5.4: A bivariate symmetric mixture of Gaussians (orange) generated from the prior (blue) by DDIM. **(a)** The complete distribution. **(b)** Outliers sampled.

Experiment L.2b - Symmetric bimodal mixture of Gaussians

The transformation demonstrated in exp. L.2a raises questions about the behavior of a model when applied on a multimodal Gaussian. Concretely, the target distribution in this experiment is a bimodal Gaussian where the modes enclose the prior on the x-axis. This bivariate mixture is illustrated by the orange points in fig. 5.4a along with the blue points for the prior.

Hypothesis 4. *An outlier in the prior does not necessarily map to an outlier in the target distribution.*

In exp. L.2a, densities in the target distribution match exactly with the prior's densities. This will not be the case for this experiment, since the model will have to distribute the prior between the two Gaussians. This setup is symmetric, and knowledge from experiment group L.1 can be applied to infer a division on the prior along the vertical line $x(y) = 0$, separating the latents based on what mode they will be transported to. For consistency, again only latents that lie at least one standard deviation away from the mean are sampled from. The results are shown in fig. 5.4b.

The pattern that emerges when sampling from outliers is different from the one in fig. 5.3c, indicating a non-linear map and strengthens hypothesis 4. Figure 5.5a provides a closer look at the pattern for the right mode. All orange points originate from latents with positive x-values, a fact that proves useful when attempting to explain how the densities are transported. The right half of the prior can, for the purpose of this discussion, be regarded as a new distribution seen in isolation from the left half. Consequently, this cluster of points is no longer reminiscent of a Gaussian, and the densities no longer match. A high-density area in the full prior is not bounded by the same coordinates as a high-density area in the right half. Points residing in the center of the full prior are outliers in the right half, explaining why the pattern of omission includes outliers in the data distribution.

For the figures discussed up until now, the omission pattern was achieved by removing specific points from the latents. To better make sense of where high-density areas in the target were mapped from, a better perspective is from the data distribution. The orange points in fig. 5.5b are outliers in the right mode distanced one standard deviation

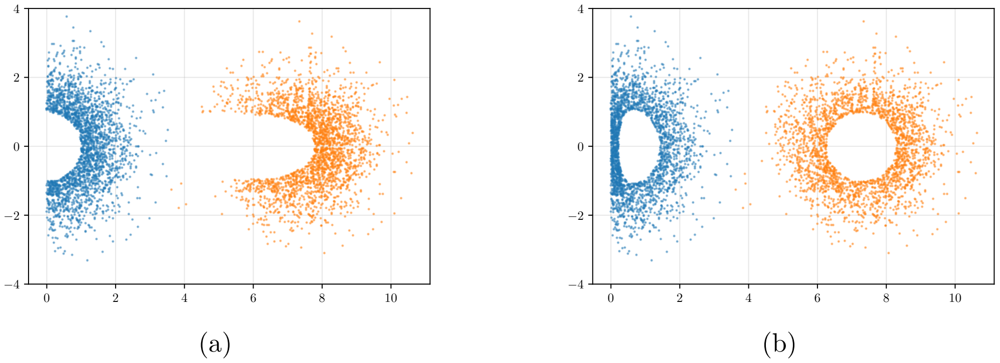


Figure 5.5: Samples from the right half of the prior. Shown are latent and data pairs, with points one standard deviation away from the center of **(a)** the latents and **(b)** the target.

away from the sampled cluster center. Paired with the data points are the corresponding latents (in blue) they were mapped from. This gives rise to a new pattern in the right half of the prior, better reflecting where the area of highest density is located when considered in isolation.

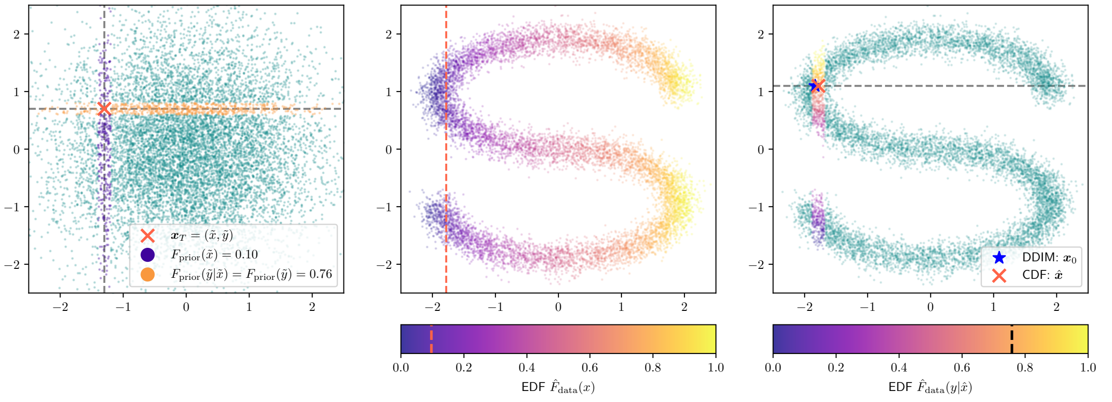
Experiment L.2c - Extending CDF-mapping to higher dimensions

Section 3.6 gave an overview of how the CDF relates to the optimal transport map between two univariate distributions. Specifically, fig. 3.11b illustrated how accumulated probability mass is preserved throughout transportation, due to non-intersecting trajectories. Now, the question arises: Is this intuitive approach applicable to higher dimensions? This is a difficult problem due to no evident way of ordering quantiles in two dimensions. However, acknowledging that the prior has no internal dependencies serves as motivation for mapping accumulated probability mass across each dimension independently.

Hypothesis 5. *The CDF approach extends to higher dimensions for DMs due to the independence between the random variables making up the prior.*

To make this experiment more reflect high-dimensional data, a target distribution is used where the CDF is not explicitly defined.

Figure 5.6a shows the prior distribution, along with an arbitrary latent $\boldsymbol{x}_T = (\tilde{x}, \tilde{y})$ located where the marginal $F_{\text{prior}}(\tilde{x}) = 0.10$ and $F_{\text{prior}}(\tilde{y}) = 0.76$. Notably, $F_{\text{prior}}(x)$ is constant along the marked vertical line, and $F_{\text{prior}}(y)$ along the horizontal line. The target distribution, represented by several points, is shown in fig. 5.6b. The color map indicates a point's relative position along the x-axis, giving rise to the marginal empirical distribution function (EDF) $\hat{F}_{\text{data}}(x)$. To find the optimal transport mapping along the x-axis, recall that preservation of accumulated probability mass implies $F_{\text{prior}}(\tilde{x}) = F_{\text{data}}(\hat{x})$, where $\hat{x}$ is the mapping of interest. This value is obtained by indexing the sorted list with the integer $\lfloor n \cdot F_{\text{prior}}(\tilde{x}) \rfloor$, where n is the number of data points.



(a) (b) (c)

Figure 5.6: Extending CDF-mapping to higher dimensions. **(a)** Latent positioned at where the independent marginal CDFs intersect. **(b)** Target distribution sorted to obtain the marginal EDF along the x-axis. **(c)** Tolerable points sorted to obtain the marginal EDF along the y-axis. × is the CDF-mapping, and ★ is a DDIM-mapping.

Knowing $\hat{x}$, points within the interval $[\hat{x} - \delta, \hat{x} + \delta]$ are suitable for the conditional EDF $\hat{F}_{\text{data}}(y|\hat{x})$, where δ is the tolerance for deviating from the optimal mapping along the x-axis. Figure 5.6c displays the distribution again, now highlighting the m extracted points sorted along the y-axis with $\delta = 0.1$. Accordingly, the optimal transport mapping from $\tilde{y}$ is found by indexing $\lfloor m \cdot F_{\text{prior}}(\tilde{y}) \rfloor$, yielding $\hat{y}$. Marked with the cross, is the CDF-mapped point $\hat{\boldsymbol{x}} = (\hat{x}, \hat{y})$. As a verification of the result, the star marks the point where DDIM maps the latent $\boldsymbol{x}_T$, matching the CDF-mapping and consequently supporting hypothesis 5. Additional CDF and DDIM mappings are provided in appendix D.

Under the assumption that DDIMs learn the optimal transport map, the following conjecture can act as an interpretative framework for how they map noise to data in higher dimensions:

Conjecture 1. *For a high-dimensional latent space, the DDIM optimal transport map approximately preserves accumulated probability mass for each dimension in the prior.*

To more explicitly observe the interplay between the dimensions throughout transportation, fig. 5.7 shows how two orthogonal masks in the prior are mapped to the target distribution by DDIM. The horizontal mask is bent to follow the local curvature, similar to the vertical mask, which is also broken up to take into account the gaps. The figure demonstrates how moving a latent along one axis while keeping the rest constant will propagate changes to the other dimensions in the data distribution. For instance, if the vertical line was positioned further towards negative x-values, keeping the y-value of the prior intersection constant, the resulting target intersection would obtain a lower y-value. In other words, the latent space is entangled.

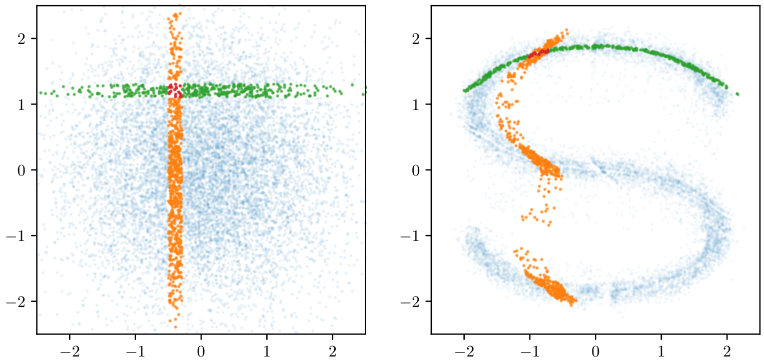


Figure 5.7: DDIM mapping orthogonal subsets of the prior to the data distribution.

Evaluation

The learned transformations depend on the target distribution. Exp. L.2a shows that it can be affine, and exp. L.2b shows that it is capable of being more complex than that. These led up to exp. L.2c, studying how CDF-mappings extend to higher dimensions. The more dimensions, the more sparse the data sets become. For instance, the Euclidean distance between two high-quality images is typically larger than between two points in the S-curve. As such, the tolerance δ is required to be large. Moreover, the experiment does not investigate the amplification of error as dimensionality increases. It is anticipated that the error associated with the CDF-method will grow larger as more dimensions are involved. Consequently, conjecture 1 should not be drawn conclusions from, but rather be viewed as a means of providing additional perspective.

5.4 Experiments on Optimal Transport: High dimensionality

As seen in section 5.3, experiments on optimal transport in lower dimensions can be illustrated compactly and sometimes verified solely by visual inspection. Now, we are also interested in observing the consequences of optimal transport when applied to data of higher dimensions. Exploring the transformations demonstrated in experiment group L.2 can prove challenging since clusters of high-dimensional data are difficult to visualize without resolving to dimensionality reduction methods, potentially leading to loss of valuable information. Consequently, the following experiments are limited to studying a relatively small number of samples in isolation.

Experiment group H.1 - Identical initial latents

The study on high-dimensional data starts by conducting experiments where identical latents $\boldsymbol{x}_T$ are used as input for various models. Employing the same latent for different

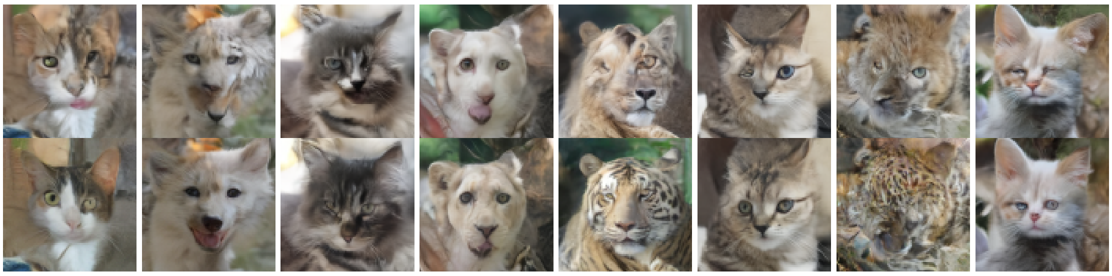


Figure 5.8: Samples from two separately trained models (rows), AFHQ256Exp1 and AFHQ256Exp2, using the same fixed latents (columns). The models are trained on the same data set.

purposes can lead to several points of reference to collectively assess multiple models' abilities to efficiently transport noise to the data distribution.

Experiment H.1a - Identical data set

We start the analysis by sampling from two models that have been trained separately on the same data set. While the training data is identical, the instances are fed into the models at random, meaning no seeding is used. To make this experiment more conclusive, the models are distinguished by their number of parameters and their training time.

Hypothesis 6. *Fixed latent inputs to two different models trained on the same data set will produce data with high pixel similarities.*

Due to the fact that DDIM is deterministic and learns the optimal transport map, it is anticipated that there will be high similarities in the images produced by the two models. Results from denoising the same set of latents are shown in fig. 5.8. The first model (top row) is a U-Net of 4 levels made up of 22 million parameters and was trained for 1100 epochs. The second model (bottom row) is also a U-Net, but of 5 levels and containing 89 million parameters trained for 1820 epochs.

Without further context, one could believe that the two rows in fig. 5.8 are meant to highlight how output quality improves throughout training a single DM. Although each column pair contains samples from different models, high-level features such as background color, fur color, and animal type are similar. Oftentimes, even low-level features like the position of eyes and fur pattern are similar. Some samples appear obscure due to limited training time and sub-optimal choices of hyperparameters. Interestingly, the two separately trained models have created similar representations of the latent space, and one can expect them to converge further towards a mutual representation given more training resources. This can be understood from the perspective of optimal transport, as the models were on the path of learning the minimum amount of change required for each pixel channel to make them collectively fit the target distribution. In conclusion, the results support hypothesis 6.

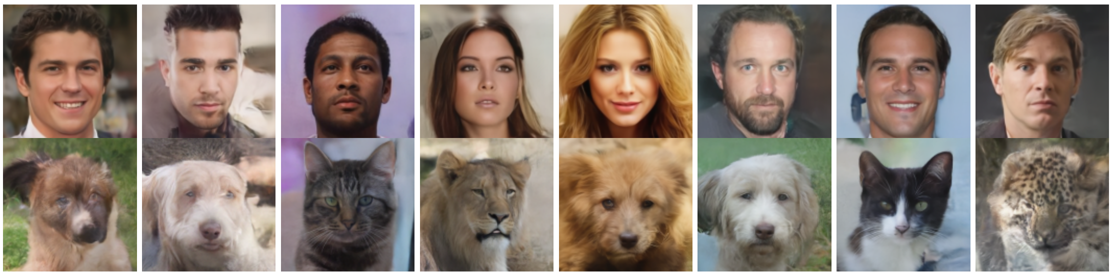


Figure 5.9: Samples from two separately trained models (rows), Celeb256 and AFHQ256, using the same fixed latents (columns). The models are trained on different data sets.

Experiment H.1b - Different data sets

In exp. H.1a, the models are trained on the same data set, leading to the convergence of similar transport maps. Khrulkov et al. (2022) conducted an experiment involving fixed latent inputs to models trained on disparate data sets. However, they limited the experiment to comparing outputs within the same overall class. For instance, they compared cats with dogs, where the shared class was *animal*, and portraits of humans with paintings of humans, with *human* as the shared class. In addition to verifying their results, this experiment serves as an extension by comparing model outputs with highly contrasting data sets.

Hypothesis 7. *Two models trained on different data sets will output pixel-similar images with fixed latent input.*

The top row in fig. 5.9 displays samples from Celeb256 and the bottom row from AFHQ256 with fixed latent inputs. As expected, the pairs do not exhibit similarities to the degree as seen in exp. H.1a. However, features like background and color tint are shared. In some cases, even low-level features appear to be shared, but upon closer inspection are groups of pixels that have gotten similar colors. For example, the background for the human sample in the second column from the right has some darker area positioned below the left ear. The animal model inferred this area to be a darker patch of fur on the cat. These features are similar only on the pixel level, making hypothesis 7 valid.

Evaluation

The findings from experiments H.1a and H.1b are undeniably intriguing. By feeding the same latent across different models, the impact of optimal transport becomes apparent. In the case of different data sets, the modalities in the two target distributions are arguably dissimilar. Still, there is some degree of pixel similarity, an example being the patch of fur on the cat and the background graphic for the man in the second column from the right. This suggests that there is some overlap in the high-dimensional PDFs. Although it is difficult to picture such PDFs, conjecture 1 supports the suggestion, such that the marginal CDFs matches between the samples from each model.

An example of a feature not consistently overlapping in the high-dimensional PDFs is the background color. This stems from a bias in the data set: As animals are commonly photographed in the wild, the background tends to have greener values as compared to human portraits. This can be observed in the first and sixth column pairs from the right in fig. 5.8.

Experiment group H.2 - Latent manipulation

As mentioned in chapter 1, when exploring the latent space of a VAE or a GAN, one typically traverses a single dimension while keeping the rest static to observe how the sample changes. This is doable as there is a strict bottleneck on the latent space, which forces the models to capture the necessary characteristics into independent generative factors. Due to the high dimensionality of DMs, independent factors are not expected to be captured by each dimension in $\mathbf{x}_T$. As such, the traversal approach is not relevant. This group of experiments looks into alternative ways of exploring the latent space, ideally hoping for an intuitive interpretation.

Experiment H.2a - Intensity modification

When comparing model outputs from fixed latents in experiment group H.1, a common observation was correlation on color balances. This aspect can be explored further by observing how a sample is altered when the latent is modified by a constant value across channel dimensions.

Hypothesis 8. *Brighter latents lead to brighter images, and darker latents lead to darker images.*

Specifically, this experiment focuses on uncovering if the intensity of an image can be changed to bring out samples where desired lighting conditions are met. For the setup, various latents $\tilde{\boldsymbol{x}}_T = \boldsymbol{x}_T + \lambda \mathbf{1}$ are sampled from, where λ is a constant scalar value linearly interpolated between $[-0.02, 0.02]$. The denoised latents $\tilde{\boldsymbol{x}}_0$ are shown in fig. 5.10. From the middle column, where $\lambda = 0$, the intensities of the latents are decreased towards the left, and increased going to the right. Interestingly, not only are lighting conditions altered, but also features like ethnicity and hair color. Additionally, some $\tilde{\boldsymbol{x}}_0$ portray men at $\lambda = 0$, while increased intensities change the portrayed subjects to women. From the endpoints, it can be argued that the samples will become increasingly obscure with increasing values of λ values. The more unlikely $\tilde{\boldsymbol{x}}_T$ is under the prior, the more difficult it will be to map it to the target distribution.

An overall observation is that linear changes in the latent do not correspond to linear changes in data. Although the requirement of dimmed or brightened images is met, the model adapts itself to latent changes by targeting modes in the underlying distribution of closer vicinity. The third row exemplifies this as λ approaches negative values. With darker areas, the model becomes attracted to features that are reasonable under the new conditions, in this case, suiting the subject with dark sunglasses. The three last panels in the top row, going towards positive values of λ, are examples where the model must

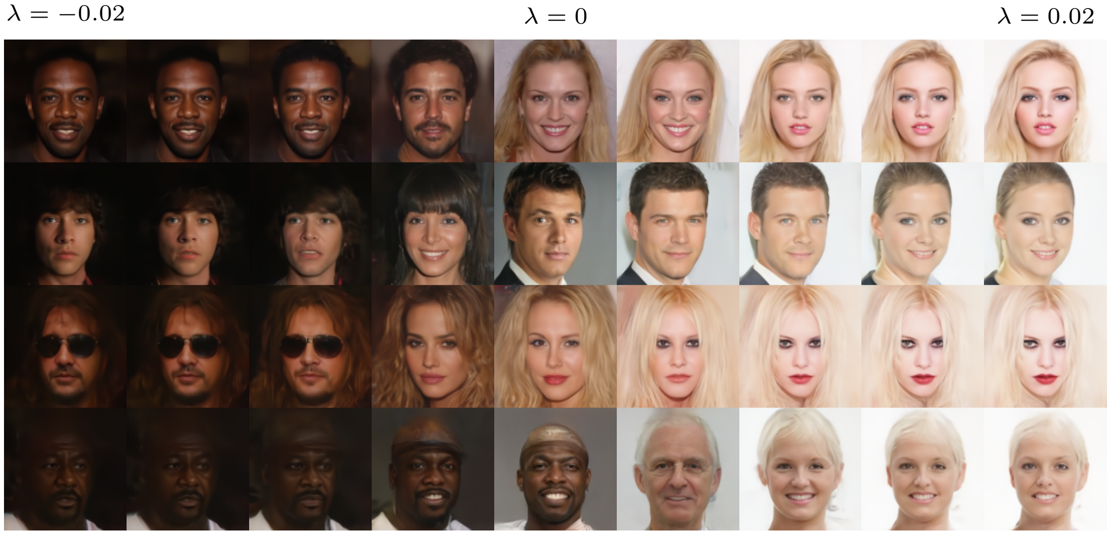


Figure 5.10: Four instances of $\tilde{\boldsymbol{x}}_0$ generated from $\tilde{\boldsymbol{x}}_T = \boldsymbol{x}_T + \lambda\mathbf{1}$ where λ is a scalar value.

overlook the aspect of finding the shortest distance to still match the target distribution. As the latent is brightened, more makeup is applied. Interestingly, this leads to darker eyeshadow. A possible explanation is that lighter skin and the amount of visible makeup are positively correlated features in the data set.

For completeness, a study on how the model adapts to changes along separate color channels is provided. Each row in fig. 5.11 contains samples from $\tilde{\boldsymbol{x}}_T = \boldsymbol{x}_T + \lambda\boldsymbol{m}$, where $\boldsymbol{m}$ is a mask targeting a color channel, and matches the dimensionality of $\boldsymbol{x}_T$.

The results indicate that the model can be forced to generate data that is highly unlikely under the target distribution. It is reasonable to assume that the data set does not contain a multitude of data points with color tints as extreme as the ones in the left-most column of fig. 5.11. Although the latents are translated in a way that makes them less likely under the prior, the presented images are still coherent and visually plausible. Both results propose that hypothesis 8 holds.

This experiment has been restricted to studying global modifications to latents, meaning all pixels in the same channel have undergone the same changes. Still, they provide one important finding: While exp. H.1b indicated color correlations across model outputs from fixed inputs, this experiment shows that the latents themselves also partake in this correlation. This insight enables a certain degree of predictability on the overall visuals of the sample prior to querying the model.

Experiment H.2b - Partial intensity modification

As pointed out in exp. H.2a, global dimming leads to sunglasses appearing. Is it possible to encourage the appearance of this feature without large changes in the values of irrelevant pixels residing outside the area where sunglasses are expected to appear? Recalling from exp. L.2a, DDIM will minimize changes to latent dimensions that already align with the

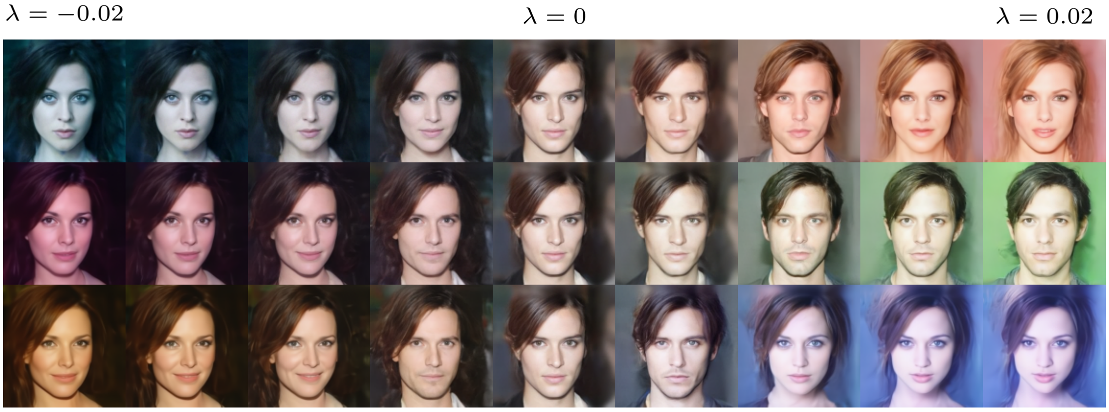


Figure 5.11: $\tilde{\boldsymbol{x}}_0$ generated from $\tilde{\boldsymbol{x}}_T = \boldsymbol{x}_T + \lambda\boldsymbol{m}$, where $\boldsymbol{m}$ masks a color channel and λ is a scalar value. Each row uses a different $\boldsymbol{m}$ to target the R, G, and B channel respectively.

target distribution. In the high-dimensional case, it is possible that a dark area in the latent is more likely to map to sunglasses due to its dark appearance. This motivates an experiment where specific groups of pixels are targeted in an attempt to bring out desired features.

Hypothesis 9. *Modifying a subset of latent dimensions will enforce both local and global changes.*

Global changes are hypothesized to emerge due to the effect observed in two dimensions in fig. 5.7, where DDIM entangles the independent latent dimensions during transportation to the target distribution. To answer the hypothesis, a mask $\boldsymbol{m}$ will be used to target and modify specific areas in the latents such that $\tilde{\boldsymbol{x}}_T = \boldsymbol{x}_T + \lambda\boldsymbol{m}$. In the case where specific results are referenced, λ and its value will be specified in superscript.

Figure 5.12a demonstrates the effect of translating a subset of latent dimensions, specifically the region around the eyes, towards negative values. The mask $\boldsymbol{m}$ is visualized as the red box on the sample $\tilde{\boldsymbol{x}}_0^{\lambda=0}$. Not only do two of the subjects obtain sunglasses, but there is also a multitude of changes to other areas positioned in latent areas that were not modified. The rows are addressed in order.

As λ decreases for the top row, the overall skin color becomes darker while the background color stays constant. Notably, many of the facial expressions undergo changes. When the sunglasses appear, the smile loses some of its strength, and the nasolabial folds diminish. While the jowl appears with a constant level of darkness throughout all values for λ, the last panel depicts this area with a texture reminiscent of a beard. The second row shows some unwanted effects of the outlined approach. From $\tilde{\boldsymbol{x}}_0^{\lambda=-0.011}$ and onwards, the gender is changed. In the last two panels, the model inferred that darker areas around the eyes in the latent maps to a shadow. To reason with this, it changes the hair into a hat. Despite faint lines, one could argue that the last panel displays some form of eyewear. The last row depicts a similar story to that

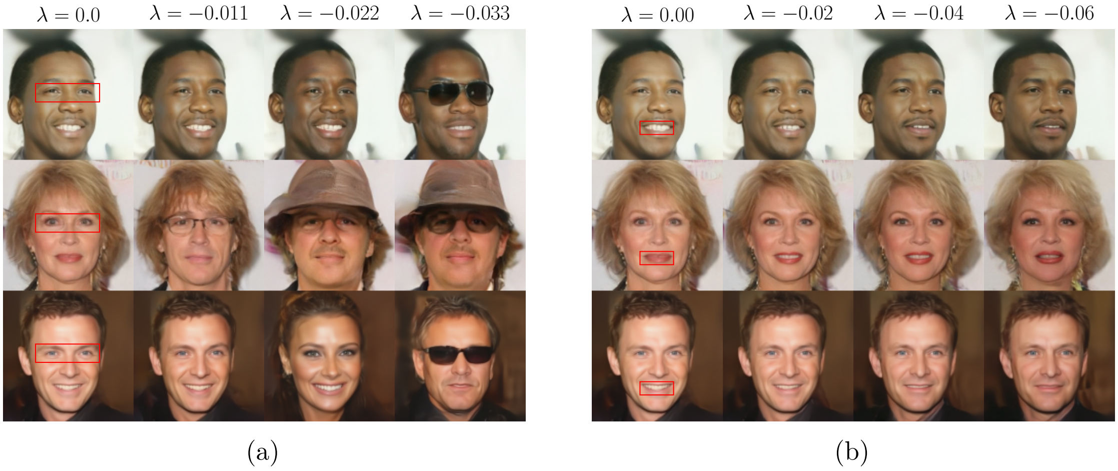
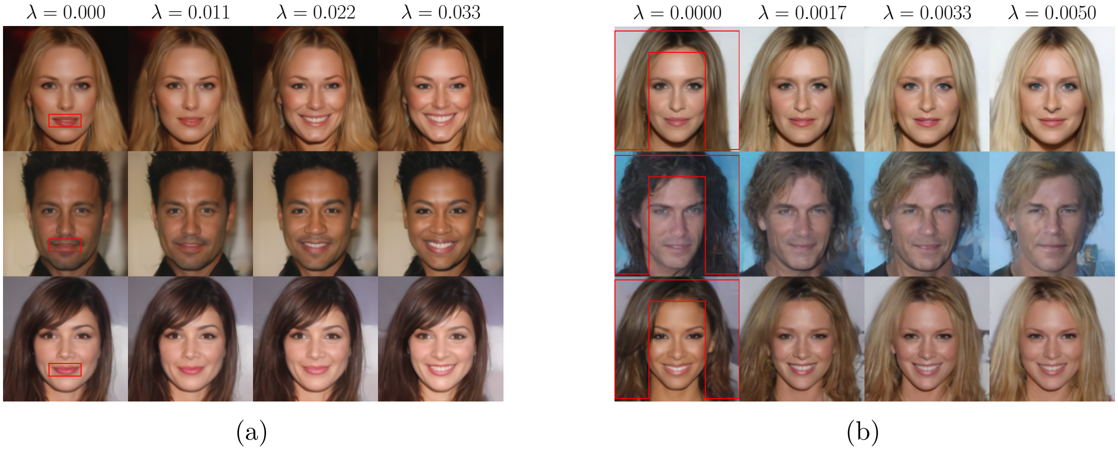


Figure 5.12: Decreasing the intensity of specific latent dimensions. The images are sampled from $\tilde{\boldsymbol{x}}_T = \boldsymbol{x}_T + \lambda \boldsymbol{m}$, where $\boldsymbol{m}$ is a mask for the red box targeting **(a)** the periorbital region and **(b)** the mouth.

of the first, where both expressions and colors are changed to match the presence of sunglasses. Unexpectedly, at $\tilde{\boldsymbol{x}}_0^{\lambda=-0.022}$, the subject temporarily switches gender. The model seems to have found it feasible to map the darker areas to eyes with eyeshadow applied. A reasonable explanation is that most subjects in the data set wearing makeup portrays women. Note that the mask is positioned at the same coordinates for all rows. Although the heads have different orientations, the model is still able to rotate the eyewear correctly.

The same experiment of decreasing the intensity is done with another latent mask, shown in fig. 5.12b, now around the mouth. In the top and bottom rows, the teeth become less visible as the subject smiles less. Again, facial expressions and skin tones are modified as a result of the local change. The eyes, which were once gleaming, now appear more fatigued. Notably, also age has seemed to change. For the middle row, $\tilde{\boldsymbol{x}}_0^{\lambda=0}$ does not display any teeth. Still, darkening the latent parts mapping to the mouth resulted in an open mouth. This area in the last panel, $\tilde{\boldsymbol{x}}_0^{\lambda=-0.06}$, still has an overall darker appearance, now with a stronger color on the lips.

Until now, only negative translations on latent groups have been studied. Some features may require groups to be shifted in the positive direction to appear. The results in fig. 5.13 are obtained through the same procedure as the ones shown in the previous figure, employing a mask on specific areas, with the difference being that λ is increased. In fig. 5.13a, the masked area is the mouth. As λ goes from 0 to 0.033, all subjects show teeth. Except for the middle row, the same person is depicted. A possible explanation for the gender alteration could be attributed to the influence on latent areas associated with dark facial hair when the mouth is targeted. As these latent pixels are increased in brightness, they lose their representation as incipient mustache growth. Once again, the change in global facial expressions reflects the emerged feature, examples being raised



(a)

(b)

Figure 5.13: Increasing the intensity of specific latent dimensions. The images are sampled from $\tilde{\boldsymbol{x}}_T = \boldsymbol{x}_T + \lambda \boldsymbol{m}$, where $\boldsymbol{m}$ is a mask for the red box targeting **(a)** the mouth and **(b)** the hair.

eyebrows, more defined jaws, and gleaming eyes.

Figure 5.13b shows results obtained by increasing λ when the mask covers the hair. For all subjects, the hair becomes brighter with small adjustments. Since the mask avoids most of the face, there are few changes in facial expressions. However, the skin tones are changed, and the woman in the bottom row even switches ethnicity. In conclusion, hypothesis 9 is valid.

Experiment H.2c - Feature transfer through latent transfer

The results from exp. H.2b suggest that groups of latent pixels encode features that will appear at the same coordinates, with some interplay with global factors. Can this knowledge be leveraged to transfer features between samples?

Hypothesis 10. *By selectively copying dimensions from one latent to another, the corresponding area in the mapping of the latter incorporates the feature present in the mapping of the former.*

The approach used in this experiment is termed feature transfer through latent transfer and can be seen as one specific approach of doing partial latent manipulation.

The two samples $\boldsymbol{x}_0^1$ (left image) and $\boldsymbol{x}_0^2$ (middle image) shown in fig. 5.14 are generated from independent latents $\boldsymbol{x}_T^1$ and $\boldsymbol{x}_T^2$. The first subject smiles, while the second does not. This feature is of interest and is therefore highlighted with the red box. In an attempt to transfer it to the second subject, the area in the latent corresponding to the area bounded by the red box is copied from $\boldsymbol{x}_T^1$ to the corresponding position in $\boldsymbol{x}_T^2$. With a mask $\boldsymbol{m}$ targeting the enclosed area, a modified latent is given as $\tilde{\boldsymbol{x}}_T^2 = \boldsymbol{m} \odot \boldsymbol{x}_T^1 + (\mathbf{1} - \boldsymbol{m}) \odot \boldsymbol{x}_T^2$, where $\odot$ is element-wise multiplication between two vectors. This modified latent maps to the sample $\tilde{\boldsymbol{x}}_0^2$ (right image).

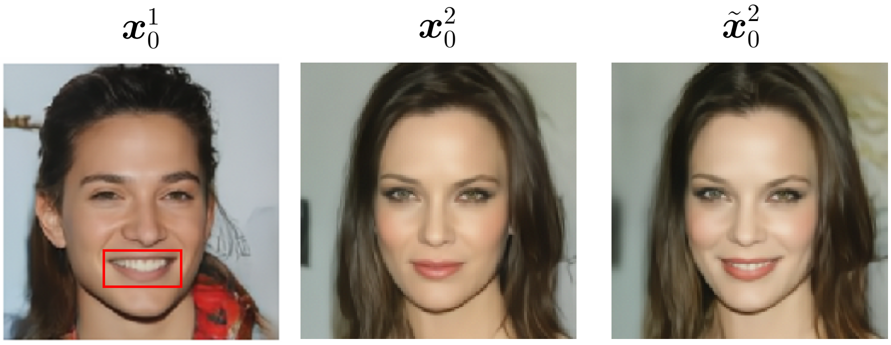
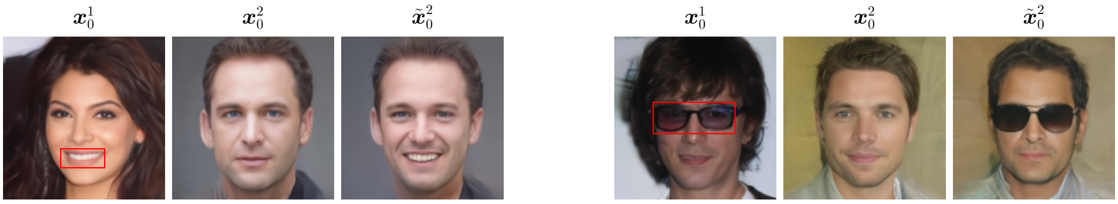


Figure 5.14: Transferring a feature from $\boldsymbol{x}_0^1$ to $\boldsymbol{x}_0^2$ by copying the masked latent pixels (highlighted by the red box) from $\boldsymbol{x}_T^1$ into $\boldsymbol{x}_T^2$, resulting in $\tilde{\boldsymbol{x}}_T^2$, from which $\tilde{\boldsymbol{x}}_0^2$ is sampled.



(a)

(b)

Figure 5.15: Additional examples of performing feature transfer through latent transfer, with the feature being **(a)** the smile and **(b)** sunglasses.

The targeted feature was successfully transferred. Also shown in exp. H.2b, the model is capable of utilizing the new information to make it sensible in regards to the data distribution. Humans that smile typically reflect this facial expression in other areas as well, as seen by how the cheeks are more tightened and the nose is more stretched horizontally. When one part of the latent suggests a smile, the model does not restrict itself to using that knowledge for the corresponding area only, but allows this information to flow out to other areas as well. More concretely, local changes lead to global effects.

This experiment demonstrates another important property, which is that the *exact same* smile is not pasted, but a variation of it that fits with the global characteristics of the original subject. The lip color of $\boldsymbol{x}_0^2$ is more saturated compared to $\boldsymbol{x}_0^1$. Even after the feature transfer, the lip color stays the same. This means that surrounding and untouched latent pixels encode information that affects the area around the mouth, and that the model allows for information to flow from surrounding areas to local areas. As such, global information affects local areas. These results are remarkable and strengthen hypothesis 10 since the feature is incorporated in both a global and local manner.

Figure 5.15 presents two additional cases of this experiment, where the desired feature is also successfully transferred and made to fit with the surrounding context. For example, the subject in fig. 5.15a appears younger and has narrowed eyes when smiling. In fig. 5.15b, the subject appears older when the sunglasses are copied over. Similar to fig. 5.14, global factors also affect the targeted area, as seen by the difference between the source sunglasses and the transferred sunglasses.

From the perspective of optimal transport, the best assumption is that the group of latent pixels was positioned close to a group of pixels rendering the specific feature. Take fig. 5.15b as an example: Although the values in the red box in the corresponding latent were randomly sampled, they happened to be less intense and therefore lead to dark sunglasses.

Evaluation

Exp. H.2a demonstrated how it is possible to manipulate the color composition of images. Since a latent is sampled from a multivariate standard Gaussians, the mean of the dimensions is expected to be 0, which has the highest relative likelihood. When the latent dimensions are shifted by a constant value, the mean is expected to be that scalar shift, consequently becoming less likely. In the context of the CDF, the latents are shifted away from the median value, potentially towards more unlikely modes. Using conjecture 1 to state that the marginal CDFs are matched is a plausible explanation for why DDIM is forced to map unlikely latents to unlikely data.

For partial intensity modification in exp. H.2b, local changes lead to global effects. This has some impractical consequences as, occasionally, unforeseen and undesirable features may appear. Sometimes it even fails to deliver the intended feature. An example is the middle row in fig. 5.12a, where a hat emerges, while the sunglasses are too faint. Moreover, the interval for λ is defined over varied magnitudes, suggesting that numerous conditions must be met for the model to behave deliberately. For each desired feature, this range must be optimized manually. On top of that, despite working for one specific sample, there is no guarantee for it to work with another.

Transferring features through latent transfer, as shown in exp. H.2c, is also a method that cannot be consistently relied upon. If two samples diverge in terms of overall lighting conditions, an area considered dark in the first sample may not be perceived as dark in the other, despite them having matching numerical values. When sampling, the model uses the surrounding latent pixels to determine specific pixel values. If the surrounding lighting conditions are relatively dim compared to the area around the eyes, the threshold for adding sunglasses will be increased. This is due to the fact that sunglasses in photographs tend to be highly contrasting in terms of color and brightness compared to other areas. In conclusion, features are more likely to be transferred if two samples are balanced color-wise.

The data sets used have been edited to ensure consistency across instances. For example, eyes are positioned at roughly the same y-coordinates in the CelebA data set. If trained on more general data where such consistency is not present, it is necessary to identify regions manually for each sample when performing partial latent manipulation.

Chapter 6

Discussion and Conclusion

DMs mark an exciting direction for data synthesis with its flexible techniques and superior performance compared to other deep generative models. Discovering tools to interpret and comprehend the mechanisms of DMs will prove vital for understanding their limitations and facilitating improvements. This chapter concludes the thesis with section 6.1 summarizing all findings, and section 6.2 stating the contributions that have been made. Following these is section 6.3, discussing possible directions for future work.

6.1 Discussion

The results from exp. H.1a give rise to a remarkable interpretation of the latent space. Referring to an entity as *optimal*, implies the global optimum. In the case of finding an optimal transport map, its connection to fundamental mathematical laws renders the solution dependent solely on the underlying data distribution. As such, one can think of the mapping from latents to data as an intrinsic aspect of nature, existing independently of the discovery and implementation of any DDIM. Since the noise predictor bases itself on this determinism, one can argue that its loss space has fewer local minima, making convergence more efficient and reliable. This can point towards an explanation as to why DMs have achieved such impressive performance.

Through the experiments on latent manipulation, many of the features that appear can be pointed toward biases in the data set. For instance, in exp. H.2a, all subjects become women as latent brightness is increased (with additional samples in appendix E). Furthermore, when subjects undergo their change to wearing sunglasses in exp. H.2b, their skin tone becomes tawnier. On top of that, as subjects are forced to smile less, their age seems to increase. Common for these observations, is that they coincide well with how the world operates. Female celebrities appear in brighter environments during photoshoots, to emphasize youthfulness. People wearing sunglasses have been exposed to the sun, leading to browner skin. Older celebrities often find photoshoots more tedious than exciting, and thus exhibit a more serious demeanor that results in weak or no smiles. These observations should also be explained from the perspective of how the model adapts its mappings when the latents become less likely. As was seen in exp. H.2a when modifying the color intensity of the latents, the model can be forced to produce images that are highly unlikely in the data set. The incidental features may be an artifact of the model working on increasingly unlikely latents. Still, the color-tinted images may reflect well how humans appear in colorful environments. As such, using a DM as a tool for feature exploration on large-scale data sets can prove valuable when attempting to uncover biases.

Sohl-Dickstein et al. (2015) experimented with a method called inpainting. It allows modifying regions of an existing image $\boldsymbol{x}_0$ by sampling as usual, but at each step, the area desired to be kept is replaced by the corresponding area sampled from $q(\mathbf{x}_t|\boldsymbol{x}_0)$. They found the model successful at filling in regions such that they seem relevant to the static parts. Nichol et al. (2022) mentioned some weaknesses of this approach when using text conditions to guide the inpainting process. They observed unwanted artifacts that led them to fine-tune the DM with a mask channel as additional input. Another disadvantage is that the model is limited to information flow from global to local contexts only. When painting in an object, the overall surrounding context will not reflect the added information. Partial latent manipulation, proposed in experiments H.2b and H.2c, alleviates both problems. It does not require retraining, as it bases itself on the deterministic and optimal mappings between latents and data. It also allows for information to flow out from the masked area and affect global features, resulting in data that aligns more with the underlying data distribution. An example is that overall facial expressions are altered as a consequence of modifications in the area around the mouth.

The partial latent manipulation method bears some resemblance to how other deep generative models are explored. For DMs, each latent dimension does not have an explicit responsibility for its mapping, but is instead heavily influenced by all other dimensions. As such, they are better studied in groups. To some extent, a group of latent dimensions can be regarded as its own generative factor, although they are far from being disentangled. This is a weak interpretation considering the vast number of permutations from the set of dimensions. Even so, with DMs being in its infancy in the ML field, the interpretation may not be far off. The method that it bases itself on, partial latent manipulation, has a theoretical connection to optimal transport. It reveals how features occur in tandem with each other, occasionally uncovering data set biases. As demonstrated, DDIM was also able to produce coherent images in between feature encouragements, further strengthening this interpretation. Finally, it does not require any supplemental model, instead rooting itself to the capabilities provided by the base framework, DDPM. Although weak compared to VAEs, partial latent manipulation still justifies itself as a powerful tool for gaining insight into the model's reasoning process, and how it perceives the latent space.

There is still some limitation regarding the exploration of the latent space with the methods proposed. Some features are more abstract and not concretely tied to textural details. For example, there are few evident areas that can be modified to guarantee a specific gender. This challenge is emphasized by the two middle samples, $\tilde{\boldsymbol{x}}_0^{\lambda=-0.011}$ and $\tilde{\boldsymbol{x}}_0^{\lambda=-0.022}$, in the bottom row of fig. 5.12. Both give the same visual impression and undeniably have a high pixel similarity, which can be attributed to the close proximity of their respective latents. Nonetheless, they depict different genders. Because of the similarity in the latents, it gives an indication the model exhibits highly sensitive and impulsive decision-making. It is also reasonable to assume that there are other areas that, when modified, could lead to a similar outcome. This behavior is difficult to gain control over, suggesting that there is still much work to be done on understanding the reasoning process that DDIM uses for data generation.

6.2 Contributions

The first goal of this thesis was to provide the essential theory surrounding DMs as a survey. Importantly, it was intended to be accessible, meaning some preliminary theory had to be included. This survey was given in chapter 3. The first research question guided the survey in discerning what scholarly works contributed the most to the theoretical foundations of DMs. Sections 3.3 to 3.5 highlight the work by Sohl-Dickstein et al. (2015), Ho et al. (2020), and Song et al. (2022) respectively.

The second goal was to uncover meaningful interpretations of the latent space of DMs, attempted through the experiments in chapter 5. The second research question inquired about the possibility of gaining insight into the behavior of DMs on the basis of their connection to optimal transport. Starting with low-dimensional data, the experiments verified the connection, showing how optimal transport affects deterministic sampling. These lead to the interpretative framework defined through conjecture 1, the usefulness of which was apparent when analyzing the results on high-dimensional data. Overall, the experiments consistently support the notion that the behavior of DMs can be adequately explained by the principles of optimal transport.

The last research question probed the plausibility of harnessing optimal transport to gain control of features that are generated. All results in experiment group H.2 demonstrate that this is possible, whether it is to modify color conditions, or to bring forth features that are color-consistent such as smiles and sunglasses.

The determinism offered by DDIM has garnered attention in the research field. However, in the commercial landscape, DDIM is merely seen as *one* way to sample data, lost in the crowd of numerous other algorithms aiming to improve efficiency, sampling quality, or to impose specific styles. This thesis has illuminated the broader value of DDIM beyond research, demonstrating its potential as a tool for data editing and data set bias identification. An example is partial latent manipulation as an alternative to inpainting. Like inpainting, it allows the global context to affect the masked area. However, it also allows global effects to occur from local enforcement.

6.3 Future Work

Within the field, DMs have been subject to alternative mathematical formulations, generalizations, and applications in other domains. However, during the research for this thesis, it was necessary to focus on specific aspects and refrain from the many breakthroughs that emerged along the way. This leaves a multitude of directions to pursue when solidifying the understanding of DMs and gaining insight into their perception of the latent space.

While Khrulkov et al. (2022) suggest that DDIM minimizes the Wasserstein distance between the prior and the target distribution, Kwon et al. (2022a) propose that DDPMs in general minimizes the Wasserstein distance between the target and the synthesized distribution. These perspectives describe two different scenarios. The first studies the sampling process, while the second focuses on the training process. In the $\mathcal{L}_{t-1}$ objective, the KL divergence is minimized between the reverse transitions $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ and $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$. This similarity measure lacks certain desirable properties, as discussed in section 2.4. What Kwon et al. (2022a) imply, is that hidden beneath the ELBO objective is a concurrent minimization of the Wasserstein distance between $q(\mathbf{x}_0)$ and $p_{\boldsymbol{\theta}}(\mathbf{x}_0)$. In the context of GANs, modifying the loss function to obtain what are termed Wasserstein GANs was found to improve stability and alleviate mode collapse (Arjovsky et al., 2017), which is a common problem. This possibly explains why DDPMs are so robust out of the box. Nonetheless, if two instances of Wasserstein minimization occur in the DDPM framework, it is worth studying the connection between them. Both describe a relationship that DMs have with optimal transport theory.

Quantile matching is a great interpretation for DDIM in one dimension. However, when extending this concept to two dimensions, as attempted in exp. L.2c, the notion of quantile order becomes nonsensical. To develop a more robust and accurate interpretative model than conjecture 1, it may prove beneficial to delve deeper into optimal transport theory in the search for more established theoretical constructs compatible with DMs.

Prior to the work by Ho et al. (2020), Song and Ermon (2020) proposed noise conditional score networks (NCSNs). Because of their notable resemblance to DMs, they have since been generalized into a unified framework based on differential equations (Song et al., 2021). Principally, the DMs outlined in this thesis are discrete solutions to differential equations, and its theoretical foundation may possess tools that have value towards interpretation, regardless of whether they are related to optimal transport.

The idea of guiding DMs has been instrumental to their success, especially when applied to text-conditioning tasks. At the core, these methods alter the unconditional direction the noise predictor proposes, forcing it to point more towards a desired mode. In their study of optimal transport, Khrulkov et al. (2022), touched upon the effect of fixing latents while varying the class condition for a DM trained on ImageNet. Their results (fig. 5) show images with shared high-level features, demonstrating how DDIM is able to design optimal transport maps for each class that bears high similarities. Although guiding was out of scope for this thesis, it is an area deserving of attention. The experiments in section 5.3 on low-dimensional data would certainly be more complete if the concept of guiding was incorporated. Examples of interesting studies would be to examine how the prior is divided when conditioning on specific modes, or how densities are transported when only one of multiple modes is targeted. Such experiments would be an extension of the already existing analytic solutions to optimal transport problems.

Subsequent to the work by Khrulkov et al. (2022), parameters for the text-conditional model, Stable Diffusion, were made public. This model allows for interesting experiments, such as studying how it reacts to a prompt suggesting a particular dominant color while latent pixels are more positive for another color. With such a general and high-capacity model, there are many aspects worth exploring.

Exp. H.1a showed that two models trained on matching data sets converged to a similar optimal transport map. The results spark curiosity about how this applies to large-scale models. Stable Diffusion is the de facto open-source text-conditional DM, with few established alternatives, except fine-tuned versions on specific visual styles. New and separate models will inevitably be released to the public, opening up the possibility of comparing their DDIM maps. An interesting case of study is to input the same text prompt and latent $\boldsymbol{x}_T$, and observe the similarities between the DDIM samples. This approach can potentially provide an unbiased benchmark for DMs. Since the sampling conditions are identical for both models, their performance can be compared by assessing whose samples are of greater quality. A requirement is that both models target a similar data distribution. For example, comparing two models where one is trained on medical images and the second on natural images serves little value except for examining local pixel similarities.

Latent manipulation holds potential in data editing tasks where a certain feature, and a global reflection of said feature, are desired. However, the experiments demonstrated situations where the global factors were excessively modified. A potential reason is the infeasible masks employed. In most cases, the mask covered additional areas that magnified the unwanted extra features. To alleviate this problem, a more sophisticated segmentation tool could be utilized to target the relevant areas more precisely.

Section 6.1 compared partial latent manipulation to inpainting. For the usual inpainting approach, the model is free to fill in whatever it sees fit since the area being filled in is initialized with random noise. A fact not taken advantage of in this thesis, is that the noise which separates $\boldsymbol{x}_T$ and $\boldsymbol{x}_0$ is known. This allows samples from the degenerate distribution $q(\mathbf{x}_t|\boldsymbol{x}_T, \boldsymbol{x}_0)$ to replace parts of the latent undergoing denoising, giving the ability to restrict certain areas from changing. Such an extension would make it bear more resemblance to inpainting, except that the area being inpainted is more controlled. Out of scope for this thesis is a technique called DDIM inversion, where data is mapped back to the latent space, effectively enabling encodings of real data. Latent manipulation coupled with DDIM inversion could prove itself as a powerful technique for generating controlled variations of the same image.

This thesis was limited to studying DMs in the domain of Computer Vision. Intriguingly, they have also shown potential in other domains such as decision-making (Ajay et al., 2022), protein design (Watson et al., 2022), audio synthesis, and natural language generation to name a few (Yang et al., 2023a). Although data is nevertheless generalized to vector form, each domain poses unique constraints, precipitating interesting interpretations on how DDIM relates the latent space to the target distribution.

Bibliography

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is Conditional Generative Modeling all you need for Decision-Making?, December 2022. URL http://arxiv.org/abs/2211.15657. arXiv:2211.15657 [cs].

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN, December 2017. URL http://arxiv.org/abs/1701.07875. arXiv:1701.07875 [cs, stat].

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives, April 2014. URL http://arxiv.org/abs/1206.5538. arXiv:1206.5538 [cs].

Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Information Science and Statistics Ser. Springer, August 2006. ISBN 978-0-387-31073-2. doi:10.1007/b9479810.1007/978-0-387-45528-0.

Yunis A. Cengel and Michael A. Boles. *Thermodynamics: An Engineering Approach.* McGraw-Hill Series in Mechanical Engineering. McGraw-Hill Education, 5 edition, June 2005. ISBN 978-0-07-288495-1. doi:10.1604/9780072884951.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse Image Synthesis for Multiple Domains, April 2020. URL http://arxiv.org/abs/1912.01865. arXiv:1912.01865 [cs].

William Dalheim. A young Scandinavian researcher looking for concepts inside diffusion models. Unpublished, December 2022.

Prafulla Dhariwal and Alex Nichol. Diffusion Models Beat GANs on Image Synthesis, June 2021. URL http://arxiv.org/abs/2105.05233. arXiv:2105.05233 [cs, stat].

Carl Doersch. Tutorial on Variational Autoencoders, January 2021. URL http://arxiv.org/abs/1606.05908. arXiv:1606.05908 [cs, stat].

John Duchi. Derivations for Linear Algebra and Optimization, 2014.

William Feller. On the Theory of Stochastic Processes, with Particular Reference to Applications. 1949.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016. URL https://www.deeplearningbook.org.

Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance, July 2022. URL http://arxiv.org/abs/2207.12598. arXiv:2207.12598 [cs].

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models, December 2020. URL http://arxiv.org/abs/2006.11239. arXiv:2006.11239 [cs, stat].

Inga Strümke. Lecture Notes in Probabilistic Diffusion Models, April 2023.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation, February 2018. URL http://arxiv.org/abs/1710.10196. arXiv:1710.10196 [cs, stat].

Valentin Khrulkov, Gleb Ryzhakov, Andrei Chertkov, and Ivan Oseledets. Understanding DDPM Latent Codes Through Optimal Transport, December 2022. URL http://arxiv.org/abs/2202.07477. arXiv:2202.07477 [cs, math, stat].

Diederik P. Kingma and Max Welling. An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. ISSN 1935-8237, 1935-8245. doi:10.1561/2200000056. URL http://arxiv.org/abs/1906.02691. arXiv:1906.02691 [cs, stat].

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes, December 2022. URL http://arxiv.org/abs/1312.6114. arXiv:1312.6114 [cs, stat].

Dohyun Kwon, Ying Fan, and Kangwook Lee. Score-based Generative Modeling Secretly Minimizes the Wasserstein Distance, December 2022a. URL http://arxiv.org/abs/2212.06359. arXiv:2212.06359 [cs, math].

Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion Models already have a Semantic Latent Space, October 2022b. URL http://arxiv.org/abs/2210.10960. arXiv:2210.10960 [cs].

Ze-Nian Li, Mark S. Drew, and Jiangchuan Liu. *Fundamentals of Multimedia*. Texts in Computer Science Ser. Springer, February 2021. ISBN 978-3-030-62123-0. doi:10.1007/978-3-030-62124-7.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Calvin Luo. Understanding Diffusion Models: A Unified Perspective, August 2022. URL http://arxiv.org/abs/2208.11970. arXiv:2208.11970 [cs].

Andrew Ng and Michael Jordan. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and Naive Bayes. *Advances in Neural Information Processing System*, 2002. URL https://ai.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf.

Alex Nichol and Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models, February 2021. URL http://arxiv.org/abs/2102.09672. arXiv:2102.09672 [cs, stat].

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models, March 2022. URL http://arxiv.org/abs/2112.10741. arXiv:2112.10741 [cs].

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Gabriel Peyré and Marco Cuturi. Computational Optimal Transport, March 2020. URL http://arxiv.org/abs/1803.00567. arXiv:1803.00567 [stat].

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models, April 2022. URL http://arxiv.org/abs/2112.10752. arXiv:2112.10752 [cs].

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation, May 2015. URL http://arxiv.org/abs/1505.04597. arXiv:1505.04597 [cs].

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding, May 2022. URL http://arxiv.org/abs/2205.11487. arXiv:2205.11487 [cs].

Magnus Själander, Magnus Jahre, Gunnar Tufte, and Nico Reissmann. EPIC: An Energy-Efficient, High-Performance GPGPU Computing Research Infrastructure, 2019. URL https://arxiv.org/abs/1912.05848. _eprint: 1912.05848.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics, November 2015. URL http://arxiv.org/abs/1503.03585. arXiv:1503.03585 [cond-mat, q-bio, stat] version: 8.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models, June 2022. URL http://arxiv.org/abs/2010.02502. arXiv:2010.02502 [cs].

Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution, October 2020. URL http://arxiv.org/abs/1907.05600. arXiv:1907.05600 [cs, stat].

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations, February 2021. URL http://arxiv.org/abs/2011.13456. arXiv:2011.13456 [cs, stat].

Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Ture. What the DAAM: Interpreting Stable Diffusion Using Cross Attention, December 2022. URL http://arxiv.org/abs/2210.04885. arXiv:2210.04885 [cs].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, December 2017. URL http://arxiv.org/abs/1706.03762. arXiv:1706.03762 [cs].

C. Villani. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-71050-9. URL https://books.google.no/books?id=hV8o5R7_5tkC.

Larry A. Wasserman. *All of Statistics*. Springer Texts in Statistics Ser. Springer, September 2004. ISBN 978-0-387-40272-7. doi:10.1007/b1229010.1007/978-0-387-21736-9. Publication Title: A Concise Course in Statistical Inference.

Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models, December 2022. URL https://www.biorxiv.org/content/10.1101/2022.12.09.519842v1. Pages: 2022.12.09.519842 Section: New Results.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion Models: A Comprehensive Survey of Methods and Applications, March 2023a. URL http://arxiv.org/abs/2209.00796. arXiv:2209.00796 [cs].

Tao Yang, Yuwang Wang, Yan Lv, and Nanning Zheng. DisDiff: Unsupervised Disentanglement of Diffusion Probabilistic Models, January 2023b. URL http://arxiv.org/abs/2301.13721. arXiv:2301.13721 [cs].

Appendices

Appendix A

Marginalizing irrelevant variables

The following proof is heavily inspired by Inga Strümke (2023). Assume we are interested in the expectation of a function $f(\mathbf{x}_{a:b})$ with respect to a distribution $q(\mathbf{x}_{0:T})$ where $1 \leq a \leq b \leq T$, meaning that $\mathbf{x}_{a:b}$ defines a subset chain of the full Markov chain from 0 to T. We show that

$$\begin{aligned}
\mathbb{E}_{q(\mathbf{x}_{0:T})}[f(\mathbf{x}_{a:b})] &= \int_{\mathbf{x}_{0:T}} q(\mathbf{x}_{0:T}) \cdot f(\mathbf{x}_{a:b}) d\mathbf{x}_{0:T} \\
&= \int_{\mathbf{x}_{0:T}} q(\mathbf{x}_{0:a-1,b+1:T} | \mathbf{x}_{a:b}) \cdot q(\mathbf{x}_{a:b}) \cdot f(\mathbf{x}_{a:b}) \ d\mathbf{x}_{0:T} \\
&= \int_{\mathbf{x}_{a:b}} q(\mathbf{x}_{a:b}) \cdot f(\mathbf{x}_{a:b}) \underbrace{\int_{\mathbf{x}_{0:a-1,b+1:T}} q(\mathbf{x}_{0:a-1,b+1:T} | \mathbf{x}_{a:b}) \ d\mathbf{x}_{0:a-1,b+1:T}}_{\text{Inner integral equals 1}} \ d\mathbf{x}_{a:b} \\
&= \int_{\mathbf{x}_{a:b}} q(\mathbf{x}_{a:b}) \cdot f(\mathbf{x}_{a:b}) \ d\mathbf{x}_{a:b} \\
&= \mathbb{E}_{q(\mathbf{x}_{a:b})}[f(\mathbf{x}_{a:b})],
\end{aligned} \tag{A.1}$$

ultimately reducing itself to the expectation over the same function but with respect to a distribution defined over the relevant variables. From the chain rule of probability, we have that two random variables $\mathbf{x}$ and $\mathbf{y}$ for an arbitrary distribution p, are related through $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}|\mathbf{y}) \cdot p(\mathbf{y})$. In eq. (A.1), we can regard $\mathbf{x}_{0:T}$ as the pair of subsets $(\mathbf{x}, \mathbf{y})$, where $\mathbf{x}$ represents $\mathbf{x}_{0:a-1,b+1:T}$ and $\mathbf{y}$ represents the remaining $\mathbf{x}_{a:b}$.

Appendix B

DDIM ELBO derivation

$$
\begin{aligned}
\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0) &\geq \mathbb{E}_{q_{\boldsymbol{\sigma}}(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})}{q_{\boldsymbol{\sigma}}(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\sigma}}(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) - \log q_{\boldsymbol{\sigma}}(\mathbf{x}_{1:T}|\boldsymbol{x}_0)\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\sigma}}(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\underbrace{\log p(\mathbf{x}_T) + \sum_{t=1}^{T} \log p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}_{\text{From eq. (3.19)}}\right. \\
&\qquad\qquad\qquad\qquad \left. - \underbrace{\left(\log q_{\boldsymbol{\sigma}}(\mathbf{x}_T|\boldsymbol{x}_0) + \sum_{t=2}^{T} \log q_{\boldsymbol{\sigma}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{x}_0)\right)}_{\text{From eq. (3.43)}}\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\sigma}}(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\mathbf{x}_1) + \log p(\mathbf{x}_T) - \log q_{\boldsymbol{\sigma}}(\mathbf{x}_T|\boldsymbol{x}_0) + \sum_{t=2}^{T} \log p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)\right. \\
&\qquad\qquad\qquad\qquad\qquad\qquad \left. - \sum_{t=2}^{T} \log q_{\boldsymbol{\sigma}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{x}_0)\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\sigma}}(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\mathbf{x}_1) + \log \frac{p(\mathbf{x}_T)}{q_{\boldsymbol{\sigma}}(\mathbf{x}_T|\boldsymbol{x}_0)} + \sum_{t=2}^{T} \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q_{\boldsymbol{\sigma}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{x}_0)}\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\sigma}}(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\mathbf{x}_1)\right] + \mathbb{E}_{q_{\boldsymbol{\sigma}}(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\mathbf{x}_T)}{q_{\boldsymbol{\sigma}}(\mathbf{x}_T|\boldsymbol{x}_0)}\right] \\
&\qquad\qquad\qquad\qquad + \mathbb{E}_{q_{\boldsymbol{\sigma}}(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\sum_{t=2}^{T} \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q_{\boldsymbol{\sigma}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{x}_0)}\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\sigma}}(\mathbf{x}_1|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\mathbf{x}_1)\right] + \mathbb{E}_{q_{\boldsymbol{\sigma}}(\mathbf{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\mathbf{x}_T)}{q_{\boldsymbol{\sigma}}(\mathbf{x}_T|\boldsymbol{x}_0)}\right] \\
&\qquad\qquad + \sum_{t=2}^{T} \mathbb{E}_{q_{\boldsymbol{\sigma}}(\mathbf{x}_t|\boldsymbol{x}_0)}\left[\mathbb{E}_{q_{\boldsymbol{\sigma}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{x}_0)}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q_{\boldsymbol{\sigma}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{x}_0)}\right]\right] \\
&= \underbrace{\mathbb{E}_{q_{\boldsymbol{\sigma}}(\mathbf{x}_1|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\mathbf{x}_1)\right]}_{\text{reconstruction term } \mathcal{J}_0} - \underbrace{D_{\mathrm{KL}}(q_{\boldsymbol{\sigma}}(\mathbf{x}_T|\boldsymbol{x}_0) \mid\mid p(\mathbf{x}_T))}_{\text{prior matching term } \mathcal{J}_T} \\
&\qquad - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q_{\boldsymbol{\sigma}}(\mathbf{x}_t|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}(q_{\boldsymbol{\sigma}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{x}_0) \mid\mid p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t))\right]}_{\text{denoising matching term } \mathcal{J}_{t-1}}
\end{aligned} \tag{B.1}
$$

Appendix C

DDPM sampling with no noise

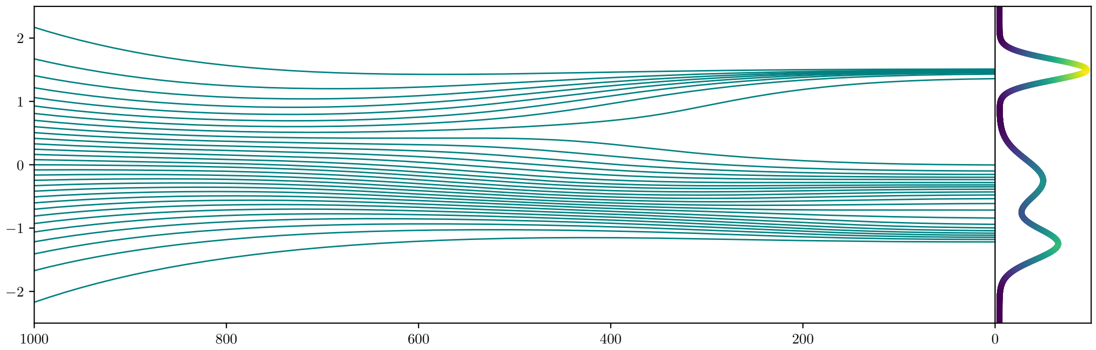


Figure C.1: Sampling from latents evenly positioned in Gaussian space with algorithm 2 and setting the noise vector $\boldsymbol{z}$ to $\mathbf{0}$ for all steps. The consequence is that DDPM fails to cover the entire target distribution.

Appendix D

Additional mappings for Experiment L.2c

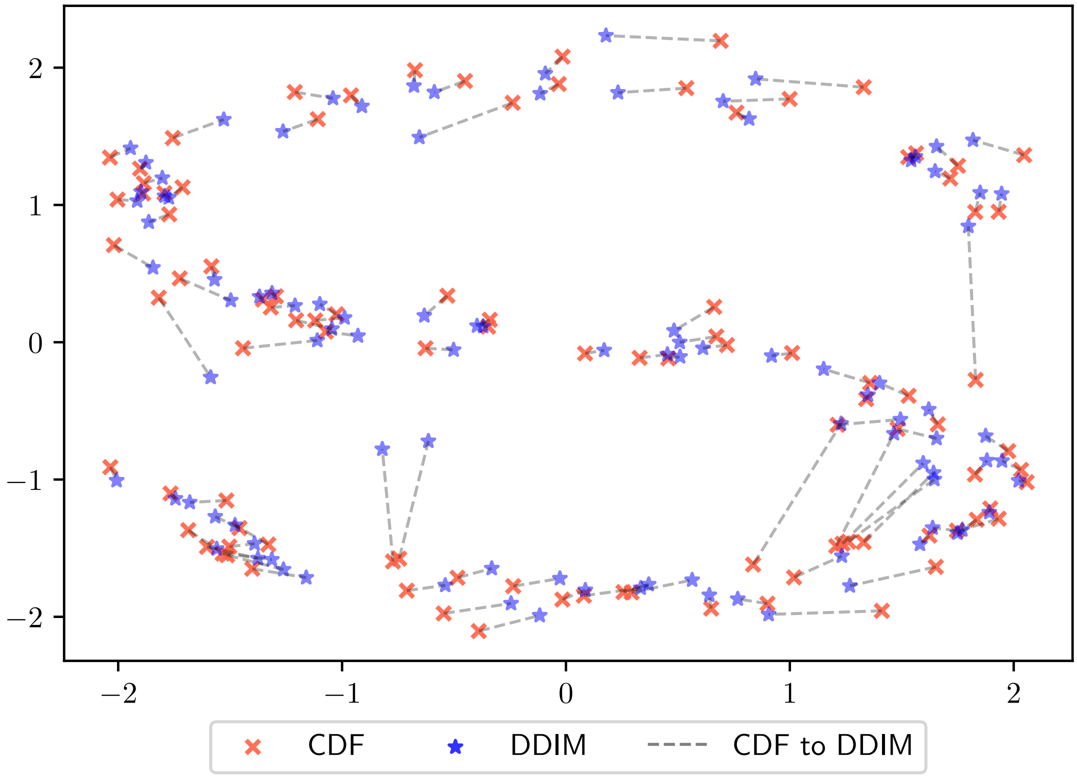


Figure D.1: Additional mappings for exp. L.2c. Crosses are CDF-mappings from latents (not shown), and stars are DDIM-mappings from the same latents. Crosses and stars are connected to highlight that they originated from the same latent point.

Appendix E

Additional samples for Experiment H.2a

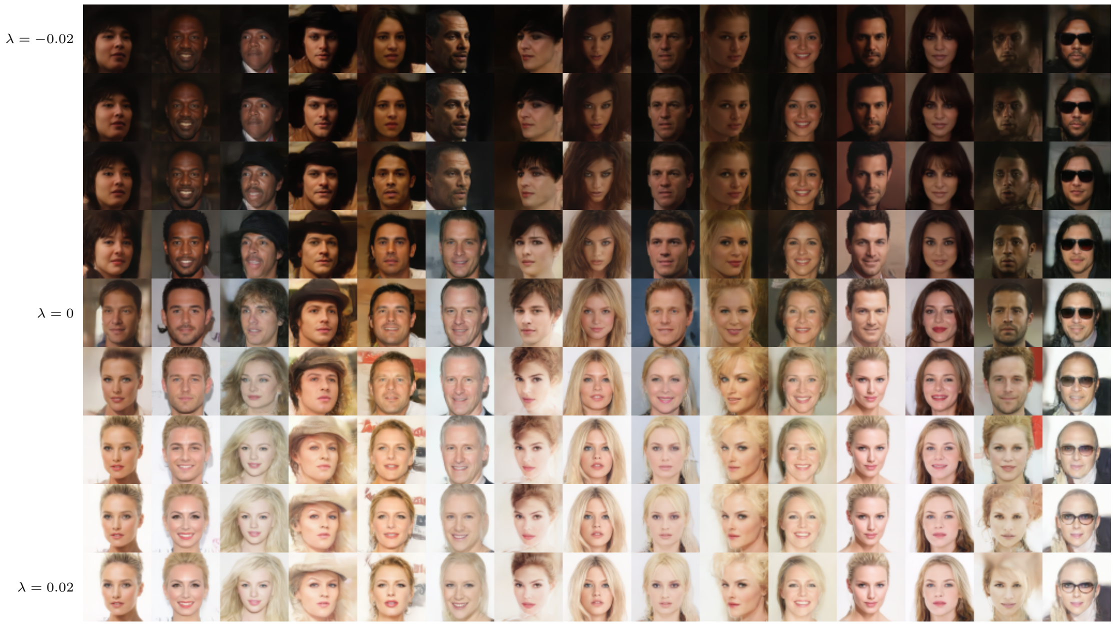


Figure E.1: Additional samples for exp. H.2a.

NTNU

Norwegian University of
Science and Technology