Generative Models with Latent Diffusions

Xinda Wu

Born 2st August 1989 in Shantou, China

1st November 2020

Master's Thesis Mathematics

Advisor: Prof. Dr. Andreas Eberle

Second Advisor: Prof. Dr. Anton Bovier

Mathematisches Institut

Mathematisch-Naturwissenschaftliche Fakultät der

Rheinischen Friedrich-Wilhelms-Universität Bonn

Abstract

This thesis is divided into 4 chapters and it investigates the recently developed gernerative models with latent diffusions [35], which is also known as neural stochastic differential equations [36].

Chapter 1 gives a brief introduction to probabilistic generative models. I will explain what the generative models are, and show how they differ from heir counterpart in machine learning, the discriminative models. And then it follows a short review on the prototype model that my study begins with, the deep latent Gasussian model (variational auto-encoder) [15, 29]. Also, we review the framework of variational inference, which is used as the essential method for model learning in this thesis. The background ingredients in this chapter provide inspirations for the study of diffusion model in later chapters.

Chapter 2 starts considering the continuous-time limit of the deep latent Gaussion models. Through the lens of stochastic control, we inspect different theoretical problems of this generalized model, including the inference problem, sampling problem, and expressivity. This part will provide solid theoretical guarantees.

Chapter 3 concerns the problem of trainability. We consider the training methods based on gradient computation and it can be reduced to considering the problem of sensitivity analysis. We discuss and compare different optional approaches and introduce a scalable method for gradient computations, the stochastic adjoint sensitivity [22].

Chapter 4 contains a summary and outlines possible extensions to this thesis.

Notation

The Euclidean norm of a vector $x \in \mathbb{R}$ will be denoted by $\|x\|$. The d-dimensional Euclidean ball of radius R centered at the origin will be denoted by $\mathsf{B}^d(R)$. A function $g : \mathbb{R}^d \times [0,1] \to \mathbb{R}$ is of class $C^{2,1}$ if it is twice continuously differentiable in the space variable $x \in \mathbb{R}^d$ and once continuously differentiable in the time variable $t \in [0,1]$. The function g is of class $C_b^{\infty,1}$ if it is bounded, smooth in the space variable and once continuously differentiable in the time variable. The standard Gaussian measure on $\mathbb{R}^d$ will be denoted by γ_d. The Euclidean heat semigroup Q_t, $t \geq 0$, acts on measurable functions $f : \mathbb{R}^d \to \mathbb{R}$ as follows:

$$Q_t f(x) := \int_{\mathbb{R}^d} f(x + \sqrt{t}z)\gamma_d(dz) = \mathbf{E}[f(x + \sqrt{t}Z)], \;\; Z \sim \gamma_d. \tag{1}$$

For two probability measures μ and ν on the same probability space, the Kullback-Leibler divergence from ν to μ is,

$$D(\mu\|\nu) = \mathbf{E}_\mu \left[\log \frac{d\mu}{d\nu}\right]. \tag{2}$$

Let ψ be a Orlicz function (i.e. $\psi : \mathbb{R}_+ \to \mathbb{R}_+$, *convex*, *continuous*, and *nondecreasing* with $\psi(0) = 0$, $\psi(x) \to \infty$ as $x \to \infty$), the Orlicz norm of a real value random variable U on the probability space $(\Omega, \Sigma, \mathbf{P})$ is,

$$\|U\|_\psi := \inf\{t > 0 : \mathbf{E}\psi(|U|/t) \leq 1\}. \tag{3}$$

When $\psi(x) = x^p$ for $p \geq 1$, the Orlicz norm is the classical L^p norm $\|\cdot\|_{L^p}$. When $\psi(x) = e^{x^2} - 1$, the resulting Orlicz norm is denoted by $\|\cdot\|_{\psi_2}$.

Contents

Abstract **1**

Notation **2**

1 Introduction **5**

1.1 Generative Models 5

1.2 Variational Inference 6

1.2.1 Posterior Approximation 6

1.2.2 Maximal Likelihood Estimation 8

1.2.3 Implementation 8

1.3 Deep Latent Gaussian Model 9

2 Generative Models with Latent Diffusions **12**

2.1 A Stochastic Control Problem 13

2.2 Variational Upper-bound 16

2.3 Exact Sampling 17

2.4 Expressivity of the Föllmer Drift 19

2.4.1 Uniform Approximation of the Heat Semigroup 22

2.4.2 Uniform Approximation of the Föllmer Drift 23

2.4.3 Upper-bound of the Approximation Error 26

3 Trainability: Scalable Gradients for Neural SDEs **28**

3.1 The Problem of Sensitivity Analysis 28

3.2 Backward Calculus 30

3.3 Stochastic Adjoint Sensitivity 32

4 Conclusion **35**

4.1 Summary and Discussion 35

A Designs of the Variational Distribution q_ϕ **37**

A.1 Mean-field Approximation 37

A.2 Normalizing Flows 38

B Auxiliary Material **42**

B.1 Results from the Theory of Empirical Process 42

B.2 Approximate Elementary Operations by Neural Nets 48

Bibliography **49**

Chapter 1

Introduction

In recent decades, due to tremendous increase in computing power since the 1990s, to train large neural networks in a reasonable amount of time turns to be possible, and the area of *deep learning* have begun to develop rapidly.

The *artificial neural networks* (ANNs) are at the very core of deep learning, which are machine learning models inspired by the networks of biological neurons found in animal brains. They were first introduced in 1943 by the neurophysiologist Warren McCulloch and the mathematician Walter Pitts [23]. They presented a simplified computational model of how biological neurons might work together in animal brains to perform complex computations. Since then many different architectures are invented, such as multilayer perceptron [31, 32], convolutional neural net [9, 19], variational auto-encoder [15, 29]. They are versatile, scalable, and of exponential expressive power, making them ideal to tackle large and highly complex machine learning tasks. These models, in general, fall into two categories, *generative* or *discriminative*.

The purpose of this thesis is to provide a detailed treatment of the class of generative model *variational auto-encoder* and the theory of its continuous-time limit under the framework of variational inference.

This chapter begins with a general introduction to the field of probabilistic generative models. We will look at what it means to say that a model is *generative*, and compare it with *discriminative* modelling. Then in the second section we review the essential material of variational inference, to see how this approach could address both the problems of posterior inference and maximal likelihood estimation in model learning. At last, we consider the details of variational auto-encoder in this framework.

1.1 Generative Models

A generative model can be broadly defined as a probabilistic model that describes how datasets are generated, and by sampling from this model we can generate new data. Its counterpart, *discriminative model* are used for classification and regression. Let us look at an example. Suppose we have a data set of painting, some by Van Gogh and some by others. With enough data we can train a discriminative model to predict if a given painting was painted by Van Gogh, while a trained generative model will attempt to generate a similar painting in Van Gogh's style. One key difference is that when performing the discriminative modellong, each observation in the training data set has a label and the modelling in fact learns a function that maps a input to the label set. On the other hand, generative modelling usually performs with unlabeled dataset.

Using mathematical notation, discriminaive modelling estimates $p(x|y)$, the probability of a label x given observation y. And generative model estimates the $p(y)$, the probability of observing observation y.

In the framework of generative model, we assume that the observations have been generated according to some unknown distribution p_{data} and the purpose of the generative model p_{model} is to try to mimic p_{data}. We say that we are impressed by p_{model} if it can generate samples that appear to have been generated by p_{data} and it should not simply reproduce those observations that we have already seen. More specifically, this is often done by defining a parametric family of densities $\{p_\theta\}_\theta$ and finding the one that maximize the likelihood on the given data $\{y_i\}_{i=1}^n$, that is, to solve the problem

$$\max_\theta \frac{1}{n} \sum_{i=1}^n \log p_\theta(y_i).$$

where $p_\theta(y)$ is interpreted as the plausibility of θ given the data y. In the following section, we review the approaches for solving inference problems.

1.2 Variational Inference

1.2.1 Posterior Approximation

In Bayesian inference, it gives a simple recipe for learning from data. Given a set of unknown latent variables x or parameters that are of interest, we specify a prior distribution $p(x)$ with the knowledge we have about x before any observation. Then we quantify how the observation data y relates to x by specifying a likelihood function $p(y|x)$. Finally, by Bayes' rule

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

it gives the posterior distribution, which quantifies what we can know about x after we have observation y.

Although this recipe is simple, the computation of marginal likelihood $p(y) = \int p(y|x)p(x)dx$ is typically intractable at least due to the following two reasons:

- The integration that $p(y)$ involves could have no closed-form solution;
- The integration could be computationaly intractable due to the large number of latent variables with complicated distributions.

Therefore we want to resort to approximation methods. Two widely used methods on this purpose include Markov Chain Monte Carlo and variational inference. The former has the advantage of being non-parametric and asymptotically exact while the later has the advantage of being faster in most of the cases. Here we have a short review on variational inference, since throughout this thesis we use it as our model learning approach.

The method of variational inference assumes that we can use a variational distribution $q_\phi(x|y)$ to approximate the true posterior, measuring the difference between them by Kullback-Leibler divergence and then this changes the inference problem into an optimization problem:

$$\min_\phi D(q_\phi(x|y)||p_\theta(x|y)) \tag{1.1}$$

where we assume they are parametized by the parameter vectors ϕ and θ respectively. The reason we use $D(q_\phi||p_\theta)$ rather than $D(p_\theta||q_\phi)$ is that in the former case the averages are taken using the tractable q_ϕ distribution rather than the intractable p_θ distribution and also in this way it avoids the case when p_θ is near zero in large region, which leads to failures of the measure.

Because the true posterior is unknown, we cannot evaluate (1.1) directly. Here it comes the trick. By definition of Kullback-Leibler divergence (2) we derive the following equation

$$\begin{aligned} D(q_\phi(x|y)||p_\theta(x|y)) &= \int q_\phi(x|y)\log\frac{q_\phi(x|y)}{p_\theta(x|y)} \\ &= \mathbf{E}_{q_\phi(x|y)}\log\frac{q_\phi(x|y)p_\theta(y)}{p_\theta(x|y)p_\theta(y)} \\ &= \mathbf{E}_{q_\phi(x|y)}\log\frac{q_\phi(x|y)}{p_\theta(x,y)} + \mathbf{E}_{q_\phi(x|y)}[\log p_\theta(y)] \\ &= \mathbf{F}_{\phi,\theta}(y) + \log p_\theta(y) \end{aligned} \tag{1.2}$$

with

$$\mathbf{F}_{\phi,\theta}(y) := D(q_\phi(x|y)||p_\theta(x)) - \mathbf{E}_{q_\phi(x|y)}[\log p_\theta(y|x)] \tag{1.3}$$

which is refered to as *variational free energy* (equivalently, *evidence lower bound* i.e. *ELBO* $:= -\mathbf{F}_{\phi,\theta}$). Since $D(q_\phi(x|y)||p_\theta(x|y)) \geq 0$, and $\log p_\theta(y)$ is a constant, then the optimization problem (1.1) is equivalent to

$$\min_\phi \mathbf{F}_{\phi,\theta}(y) \tag{1.4}$$

From here, in order to construct the approximate posterior q_ϕ such that it is as close as possible to the true posterior, one can work with the free energy (1.3) instead of trying to evaluate (1.1) directly.

1.2.2 Maximal Likelihood Estimation

The maximal likelihood estimation concerns the marginal likelihood $p_\theta(y)$, and the following optimization problem

$$\max_\theta p_\theta(y) \tag{1.5}$$

where $p_\theta(y)$ measures the probability of θ given some observed data y. By the Jensen's inequality, we can derive the following upper-bound,

$$\begin{aligned}
-\log p_\theta(y) &= -\log \int p_\theta(x,y)dx \\
&= -\log \int \frac{p_\theta(x,y)q_\phi(x|y)}{q_\phi(x|y)}dx \\
&= -\log \mathbf{E}_{q_\phi(x|y)}\left[\frac{p_\theta(x,y)}{q_\phi(x|y)}\right] \\
&\overset{\text{Jensen}}{\leq} -\mathbf{E}_{q_\phi(x|y)}\left[\log\frac{p_\theta(x,y)}{q_\phi(x|y)}\right] \\
&= -\mathbf{E}_{q_\phi(x|y)}\left[\log\frac{p_\theta(y|x)p_\theta(x)}{q_\phi(x|y)}\right] \\
&= \underbrace{D(q_\phi(x|y)||p_\theta(x)) - \mathbf{E}_{q_\phi(x|y)}[\log p_\theta(y|x)]}_{\mathbf{F}_{\phi,\theta}(y):=}
\end{aligned} \tag{1.6}$$

The equality in (1.6) can be attained when q_ϕ is the true posterior, i.e. $q_\phi(x|y) = p_\theta(x|y)$. And then the optimization problem (1.5) is equivalent to

$$\min_{\phi,\theta} \mathbf{F}_{\phi,\theta}(y) \tag{1.7}$$

It is different here from the former case of posterior approximation by optimizing over both sets of parameters ϕ, θ.

1.2.3 Implementation

There are mainly two problems that must be addressed to successfully use variational inference. The first is the efficient computation of the gradient of free energy with respect to parameters. To address this problem, it usually uses some specially designed Monte Carlo gradient estimates according to the specific models. The second problem is to choose a family of variational distributions $\{q_\phi\}$ with rich expressiveness as well as being computationally feasible, which would hopefully to be able to well approximate or even recover the true posteriors. Classical methods include mean-field approximation [25], mixture models [14], which share some common limitations that either have poor performance on approximating the true posterior or have the computational complexity unscalable. Recently, some promising method called *normalizing flow* [30] is introduced, which is designed to effectively overcome these shortcomings. We have included a short review on this in Appendix A. As follows, we summarize the general steps of variational inference in Algorithm 1.

```
Algorithm 1: Learning with Variational Inference
Input: parameters $\phi$ for variational distributions, $\theta$ for the
       generative model (both initialized randomly).
while the free energy $F_{\phi,\theta}$ not converged do
    $y \leftarrow$ {Get mini-batch}
    compute the variational distribution $q_\phi$
    sample $x \sim q_\phi(\cdot)$
    compute the free energy $F_{\phi,\theta}(y) \approx F_{\phi,\theta}(x, y)$
    $\Delta\theta \propto -\nabla_\theta F_{\phi,\theta}$
    $\Delta\phi \propto -\nabla_\phi F_{\phi,\theta}$
end
```

In the following section, we study a specific class of generative model, the deep latent Gaussian model and see how the framework of variational inference could address the inference problem in model learning.

1.3 Deep Latent Gaussian Model

The deep latent Gaussian model (DLGM) [15, 29] are designed based on the idea of *representation learning*: Instead of trying to model the high-dimensional sample space directly, it should describe each observation in the training set using some low-dimensional *latent* space and then learn a mapping function that can take a point in the latent space and map it to a point in the original domain. Each point in the latent space is the *representation* of some high-dimensional observation.

In other words, it is designed to exact features of a given data manifold, which will be learned by the parameters in the latent object. Then one can use this model to generate similar samples. It has many applications. For instance, dimentional reduction in data processing, denoicing, missing data imputation. Moreover, some advanced techniques such as DeepFake using two DLGMs to play face swap. When this model is trained using the variational inference method, it is also refer to as *variational auto-encoder*.

Mathematically, this model is defined as follows. The latent variables $X_0, ..., X_k$ and the observed variable Y are generated recursively according to

$$
\begin{aligned}
X_0 &= Z_0 \\
X_i &= X_{i-1} + b_i(X_{i-1}) + \sigma_i Z_i, \quad i = 1, ..., k \\
Y &\sim p(\cdot|X_k)
\end{aligned}
\tag{1.8}
$$

where $Z_i \overset{i.i.d}{\sim} \mathcal{N}(0, I_d)$ in $\mathbb{R}^d$ and $b_i : \mathbb{R}^d \to \mathbb{R}^d$ parametric nonlinear transformations, $\sigma_i \in \mathbb{R}^d \times \mathbb{R}^d$ a sequence of matrices, $p(\cdot|\cdot)$ the observation likelihood. In practice, b_i would be implemented by different types of neural nets according to its usage. In this thesis, we assume the neural nets that we use are the multi-layer perceptrons (MLP), i.e. the fully connected feedforward neural networks

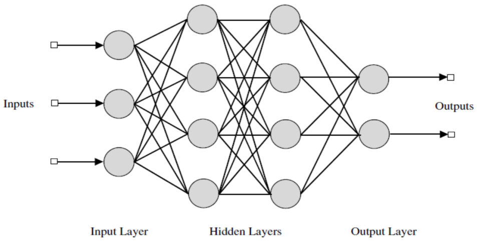

Figure 1.1: MLPs

$$Y = b(X) := \varphi_h(W_h^\intercal \varphi_{h-1}(W_{h-1}^\intercal ... \varphi_1(W_1^\intercal X)...))$$

where you have the input $X \in \mathbb{R}^n$, and the output $Y \in \mathbb{R}^m$. The connections in the graph are the weight parameters which are encoded into the weight matrices W. And in the nodes, it uses the non-linear activations $\varphi : \mathbb{R} \to \mathbb{R}$. The input propagates forward by iteratively applying the weight matrices and non-linear activations.

It is worth mentioning that the formulation of DLGM (1.8) generalises a number of well known models. For example, when it has only one layer of latent variables and uses a linear mapping $b(\cdot)$, it recover *factor analysis* [2], while more general mappings allow for *non-linear factor analysis* [20].

Denote all the parameters in model (1.8) by θ. The underlying generative process can be captured by the joint densiy,

$$p_\theta(y, x_0, ..., x_k) = p(y|x_k) \prod_{i=1}^{k} p(x_i|x_{i-1}) p(x_0) \tag{1.9}$$

which factorizes over the hidden variables. The main object of the inference problem for this model is the marginal likelihood $p_\theta(y)$, obtained by integrating out all the latent variables. However, this integration is typically intractable. Thus, one instead wants to use the method of variational inference that we have been talking about in the last section. By (1.6), we have

$$-\log p_\theta(y) \leq \underbrace{D(\underbrace{q_\phi(x_0,...,x_k|y)}_{\text{app.post.(encoder)}}\,||\,\underbrace{p_\theta(x_0,..,x_k)}_{\text{prior}})}_{\text{regulariser}} \underbrace{-\mathbf{E}_{q_\phi}[\log \quad \underbrace{p_\theta(y|x_k)}_{\text{likelihood(decoder)}} \quad]}_{\text{reconstruction error}} \tag{1.10}$$

where the right-hand side consists of two terms: the first term is the Kullback-Leibler divergence between the approximate posterior and the prior, acting as a regulariser, while the second term is a reconstruction error. Then for the model training, it learns the parameters ϕ jointly with θ by minimizing this upper-bound.

We refer to the distribution $q_\phi(x|y)$ as a *recognition model*, whose design is independent of the generative model $p_\theta(y|x)$ (some widely used methods to construct this approximate posterior is are given in Appendix A). Besides, we say that the generative model would have better representative power when the number of latent variables increases and it would be the best when it goes to infinity. However, by this way it would also get infinitely many parameters as well, which is not implementable. And here it is the point that motivates us to investigate the continuous-time limit of this model and regard it as a stochastic process. We will see later that the variational upper-bound has natural counterpart in the context of the model with latent diffusions.

Chapter 2

Generative Models with Latent Diffusions

In this chapter, we consider the continuous-time limit of DLGM (1.8), the latent object becomes a d-dimensional Itô diffusion

$$dX_t = b(X_t, t)dt + \sigma(X_t, t)dW_t, \ t \in [0, 1]$$

with latent space $\mathbb{W} = C([0, 1]; \mathbb{R}^d)$, the space of continuous path $w : [0, 1] \to \mathbb{R}^d$. The observation variable is generated conditionally on X_1 i.e. $Y \sim p(\cdot|X_1)$. The joint density in this case becomes

$$p_\theta(dy, dw) = p_\theta(y|X_1(w))\nu_w(dw)dy$$

where ν_w is the Wiener measure on $\mathbb{W}$. Considering this model has several benefits:

- Memory efficiency. The representative power of DLGM will generally increase along with the increase of the number of latent variables. However, in the meanwhile the number of parameters also increases largely which makes it computationally prohibitive. By consider the continuous-time limit, it overcomes this limitation.
- Better representation of given data in the latent object. This diffusion limit corresponds to the case when DLGM has infinitely many latent variables, implying that it should possess the best representative performance.
- It could enable the use of SDE solvers, which we already have in many years' development for both efficiency and accuracy.

Let $(\Omega, \mathcal{F}.\{\mathcal{F}_t\}, \mathbf{P})$ be a probability space with complete, right-continuous filtration $\{\mathcal{F}_t\}$ and let $W = \{W_t\}$ be a standard d-dimensional Brownian motion adapted to $\{\mathcal{F}_t\}$. We focus on the special case where $\sigma \equiv I_d$,

$$dX_t = b(X_t, t; \theta)dt + dW_t, \qquad t \in [0, 1]; \ X_0 = x_0 \tag{2.1}$$

and $b : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$ is implemented by feedforward neural nets, with θ the weight parameters and here we use the term *neural SDE* to refer to the Itô processes with the drift and the diffusion coefficient function implemented by neural nets. Moreover, we assume that b sufficiently well behaves, such as being *bounded, Lipschitz continuous*, and admitting a unique strong solution and a transition kernel $\kappa_{s,t} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ for $0 \leq s \leq t \leq 1$. We will see later in the study of sampling problem (Section 2.3) that the setting $\sigma \equiv I_d$ is sufficient to guarantee a rich representative power. In the following sections, we would like to show how the theory of stochastic control provides a unified viewpoint on different problems concerning this generative model.

2.1 A Stochastic Control Problem

Consider the following stochastic control problem. Define the controlled diffusion process $X^u = \{X^u_t\}_{t\in[0,1]}$ as follows

$$dX^u_t = (b(X^u_t, t; \theta) + u(X^u_t, t; \phi))dt + dW_t, \quad t \in [0, 1]; X^u_0 = x_0 \tag{2.2}$$

where $u : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$ is an *admissible control* which means it satisfies:

1. u_t is $\sigma\{W_t\}$-measurable;
2. (2.2) admits a strong solution on $[0, 1]$;
3. $\mathbf{E}\left[\int_0^1 \|u_s\|^2 ds\right] < \infty$.

Both b and u are parametric and assumed to be implemented by neural nets. We say that the diffusion X^u is controlled by u. Then for given function $g : \mathbb{R} \to (0, \infty)$, we define the following *cost-to-go* function in the form of variational free energy (1.3)

$$J^u(x, t) := D(\mathbf{P}^u||\mathbf{P}^0) - \mathbf{E}[\log g(X^u_1)|X^u_t = x] \tag{2.3}$$

where $\mathbf{P}^0 := \mathrm{Law}(X_{[t,1]})$ and $\mathbf{P}^u := \mathrm{Law}(X^u_{[t,1]})$. The function g indicates some terminal condition. The Kullback-Leibler divergence on the right-hand side can be interpreted as the control cost, and its computation looks forbidding, as it involves the integration with respect to the probability measure on the path space $\mathbb{W}$. A simplification is allowed from the Girsanov's formula ([26], p.142): any probability measure $\boldsymbol{\mu}$ on $\mathbb{W}$ absolutely continuous with respect to the Wiener measure $\boldsymbol{\nu}$ (i.e. $\boldsymbol{\mu} \ll \boldsymbol{\nu}$) corresponds to the addition of a drift term to the basic Wiener process. And since the prior process X and the controlled diffusion X^u differ by a change of drift, we have the following Radon-Nikodym derivative

$$\frac{d\mathbf{P}^u}{d\mathbf{P}^0} = \exp\left(\int_t^1 u_s^\intercal dW_s + \frac{1}{2}\int_t^1 \|u_s\|^2 ds\right) \tag{2.4}$$

where $u_t^\intercal dW_t := \sum_{i=1}^d u_t^{(i)} dW_t^{(i)}$ with $u_\cdot^{(i)}$ and $dW_\cdot^{(i)}$ denoting i-th coordinate of u and W. Then we calculate the Kullback-Leibler divergence

$$D(\mathbf{P}^u || \mathbf{P}^0) = \mathbf{E}_{\mathbf{P^u}} \left[\log \frac{d\mathbf{P}^u}{d\mathbf{P}^0} \right] = \mathbf{E} \left[\frac{1}{2} \int_t^1 \|u_s\|^2 ds \right] \tag{2.5}$$

and we have the following *Girsanov representation*

$$J^u(x,t) := \mathbf{E} \left[\frac{1}{2} \int_t^1 \|u_s\|^2 ds - \log g(X_1^u) \middle| X_t^u = x \right] \tag{2.6}$$

Now, our goal is to find the *value function* $v : \mathbb{R}^d \times [0,1] \to \mathbb{R}_+$, defined by

$$v(x,t) := \inf_u J^u(x,t); \quad v(\cdot,1) = -\log g(\cdot) \tag{2.7}$$

and the optimal control u^* such that u^* s.t. $J^{u^*}(x,t) = v(x,t)$ for all x and t.

To solve the problem (2.3), first we consider the following intuitive:

> *Bellman's principle of optimality* [3]: An optimal policy has the property that whatever the initial state and initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

With this principle, we can derive the corresponding *Bellman's equation* associated to the above control problem and by solving this partial differential equation we can finally find the solution for the value function. The following theorem shows us the result.

Theorem 2.1.1. *Consider the above control problem, then the value function is given by*

$$v(x,t) = -\log \mathbf{E}[g(X_1)|X_t = x], \tag{2.8}$$

the optimal control is given by

$$u^*(x,t) = -\nabla v(x,t) = \nabla \log \mathbf{E}[g(X_1)|X_t = x], \tag{2.9}$$

the corresponding controlled diffusion X^{u^*} has the transition density

$$\kappa_{s,t}^*(x,y) = \kappa_{s,t}(x,y) \exp(v(x,s) - v(y,t)), \tag{2.10}$$

where $\kappa_{s,t}(\cdot)$ is the transition density of uncontrolled process.

Proof. Since the last term $-\log g(\cdot)$ in the cost-to-go function (2.3) gives a preset value at the final state, i.e. $J^u(x,1) = -\log g(x)$, we want to apply the principle of optimality and work out backward. Suppose we have known the optimal policy backward up to the moment $t + dt$, and we want to extend it on $[t, t+dt]$. Assume that $v(x,t)$ is the induced value function.

By the principle of optimality, the value function at time t is equal to the minimization over the sum of its value at time $t + dt$ and the 'energy cost' incurred in this time increment. Therefore, from t to $t+dt$, by (2.6) we have

$$v(x,t) = \min_u \left\{ v(x, t+dt) + \mathbf{E}\left[\frac{1}{2} \int_t^{t+dt} \|u_s\|^2 ds \right] \right\}$$

Then we use Itô formula to expand $v(x, t+dt)$ and let $dt \to 0$. We obtain the *Bellman's equation* associated to the control problem (2.3)

$$\frac{\partial v}{\partial t} + \mathcal{L}_t v = -\min_{\alpha \in \mathbb{R}^d} \left\{ \alpha^\intercal \nabla v + \frac{1}{2}\|\alpha\|^2 \right\}; \quad v(\cdot, 1) = -\log g(\cdot) \tag{2.11}$$

where $\mathcal{L}_t$ is the generator of the uncontrolled process (2.1)

$$\mathcal{L}_t h(x,t) := b(x,t)^\intercal \nabla h(x,t) + \frac{1}{2} \operatorname{tr} \nabla^2 h(x,t)$$

for any function $h \in C^{2,1}(\mathbb{R}^d \times [0,1])$. We want to solve the PDE (2.11) for the value function. This can be reduced to the following Cauchy problem

$$\frac{\partial v}{\partial t} + \mathcal{L}_t v = \frac{1}{2}\|\nabla v\|^2 \text{ on } \mathbb{R}^d \times [0,1]; \quad v(\cdot, 1) = -\log g(\cdot) \tag{2.12}$$

With simple computation, we can verify that (2.12) is solved by the negative logarithmic transformation of space-time harmonic function h, which satisfies

$$\frac{\partial h}{\partial t} + \mathcal{L}_t h = 0 \text{ on } \mathbb{R}^d \times [0,1]; \quad h(\cdot, 1) = g(\cdot) \tag{2.13}$$

By Feynman-Kac formula ([16], Thm.24.1), $h(x,t) = \mathbf{E}[g(X_1)|X_t = x]$ is a solution of (2.13). We then obtain the value function

$$v(x,t) = -\log \mathbf{E}[g(X_1)|X_t = x]$$

By (2.11) and (2.12), we have

$$\frac{1}{2}\|\nabla v\|^2 = -\min_{\alpha \in \mathbb{R}^d} \left\{ \alpha^\intercal \nabla v + \frac{1}{2}\|\alpha\|^2 \right\}, \tag{2.14}$$

where the optimizer is given by $\alpha^* = -\nabla v$ which implies that the optimal control we are looking for is $u^*(x,t) = -\nabla v(x,t)$.

To derive the transition density of the controlled diffusion, note that the optimal controlled diffusion X^{u^*} and the prior process X differ from each other by a change of the additional drift u^*, then by the Girsanov's formula ([26], p.142), we have the Radon-Nikodym derivative $\kappa^*_{s,t}/\kappa_{s,t}$

$$\frac{\kappa^*_{s,t}(x,y)}{\kappa_{s,t}(x,y)} = \exp\left(\int_s^t u_r^{*\intercal} dW_r + \frac{1}{2}\int_s^t \|u_r^*\|^2 dr \right) \tag{2.15}$$

On the other hand, by Itô's formula and (2.12),

$$
\begin{aligned}
v(x,s)-v(y,t) =& \log h(y,t)-\log h(x,s)\\
=& \int_s^t \left[\frac{\partial(\log h)}{\partial r}+\nabla(\log h)^{\intercal}(b+u^*)+\frac{1}{2}\operatorname{tr}\nabla^2(\log h)\right]dr\\
&+\int_s^t \nabla(\log h)^{\intercal}dW_r\\
=& -\int_s^t\left[\frac{\partial v}{\partial r}+\mathcal{L}_r v+\nabla v^{\intercal}u^*\right]dr-\int_s^t\nabla v^{\intercal}dW_r\\
\overset{2.12}{=}& \frac{1}{2}\int_s^t\|u_r^*\|^2dr+\int_s^t u_r^{*\intercal}dW_r
\end{aligned}
$$

Therefore, with (2.15) we have

$$
\kappa^*_{s,t}(x,y)=\kappa_{s,t}(x,y)\exp(v(x,s)-v(y,t)).
$$

■

From the results above, we obtain the following variational principle, which will be applied repeatedly in following sections.

Corollary 2.1.1 (*entropy inequality*)**.** *For any control $u\in U$,*

$$
-\log\mathbf{E}[g(X_1)|X_0=x]\le D(\mathbf{P^u}||\mathbf{P^0})-\mathbf{E}[\log g(X_1^u)|X_0^u=x] \tag{2.16}
$$

Proof. On the right-hand side, it is the value function which is always less than or equal to the cost-to-go function. ■

2.2 Variational Upper-bound

Now we consider the problem of variational inference concerning the model (2.1) and derive the corresponding variational upper-bound as the form (1.6). Suppose we are given n observations $y_1,...,y_n$, and wish to upper-bound the negative log-likelihood

$$
L_n:=-\frac{1}{n}\sum_{i=1}^n\log\mathbf{E}\left[p_\theta(y_i|X_1)\right]
$$

where $\{X_t\}$ is the diffusion process in (2.1). Consider the control problem (2.3) with $g(\cdot)=p(y|\cdot)$ to be the observation likelihood for some fix y. Then by the entropy inequality (2.16), any admissible control u will give rise to an upper-bound as follows

$$
\begin{aligned}
-\log\mathbf{E}[p_\theta(y|X_1)|X_0=x]\le& D(\mathbf{P^u}||\mathbf{P^0})-\mathbf{E}[\log p_\theta(y|X_1^u)|X_0^u=x]\\
=&\mathbf{E}\left[\frac{1}{2}\int_0^1\|u_s\|^2ds-\log p_\theta(y|X_1^u)\middle|X_0^u=x\right]:=\mathbf{F}^u(y;\phi,\theta)
\end{aligned}
\tag{2.17}
$$

Compare it with the one derived for DLGM in (1.10)

$$-\log p_\theta(y) \leq \underbrace{D(q_\phi(x_0, ..., x_k|y)||p_\theta(x_0, .., x_k)) - \mathbf{E}_{q_\phi}[\log p_\theta(y|x_k)]}_{\mathbf{F}_{\phi,\theta}(y)}$$

we see they are structurally identical. One notable difference between these two settings is that here the only degree of freedom we need in (2.17) is the additive drift u which affects the mean, whereas in the DLGM case we have to optimize both over the mean and the covariance matrix in (1.8).

2.3 Exact Sampling

The problem of exact sampling traces back to the *Schrödinger bridge problem* in 1931 [33]. The objective is to construct a process $\{X_t^u\}_{t\in[0,1]}$ such that X_1^u has a given distribution. We formulate the problem as follows: Given a target distribution μ, and the prior process $\{X_t\}_{t\in[0,1]}$ with $X_1 \sim \nu$. What would be the appropriate control u, such that $X_1^u \sim \mu$? Moreover, among all these admissible controls, which one would take the *minimum energy*? Here by the term *energy* we refer to the control cost in (2.3). That is, when do we have

$$D(\mathbf{P}^u||\mathbf{P}^0) \to \min?$$

On the other hand, from the perspective of variational inference (refer to the upper-bound (2.17) in last section), this problem can also be regarded as: what is the optimal posterior process X^* w.r.t given a target distribution μ?

This problem can be naturally addressed by the results of the control problem (2.3) that we have considered. We have the following theorem.

Theorem 2.3.1. *Given a target distribution μ and a prior process $\{X_t\}_{t\in[0,1]}$ as (2.1) with $X_0 = 0$, $X_1 \sim \nu$, and $\mu \ll \nu$. Let the terminal condition function g in the control problem (2.3) to be $f := d\mu/d\nu$, then*

$$X_1^* \sim \mu$$

and the optimal control u^* has the minimal energy

$$D(\mu||\nu) \leq D(\mathbf{P^u}||\mathbf{P^0}) \tag{2.18}$$

among all admissible controls u that induce the target distribution μ at $t = 1$.

Proof. First we show that $X_1^* \sim \mu$,

$$\begin{aligned}\mathbb{P}[X_1^* \in A] &= \int_A \kappa_{0,1}^*(0,y)dy \\ &= \int_A \kappa_{0,1}(0,y)\exp(v(0,0) - v(y,1))dy \\ &= \int_A f\, d\nu \\ &= \mu(A).\end{aligned}$$

Now we recall the entropy inequality (2.16), let $g = f$

$$-\log \mathbf{E}[f(X_1)|X_0 = 0] \leq D(\mathbf{P^u}||\mathbf{P^0}) - \mathbf{E}[\log f(X_1^u)|X_0^u = 0].$$

Consider any admissible control u with $X_0^u = 0$ and the property that $X_1^u \sim \mu$, then on the left-hand side of the inequality we have

$$\mathbf{E}[f(X_1)|X_0 = 0] = \mathbf{E}_\nu\left[\frac{d\mu}{d\nu}\right] = 1,$$

and on the right-hand side

$$\mathbf{E}[\log f(X_1^u)|X_0^u = 0] = \mathbf{E}_\mu\left[\log\frac{d\mu}{d\nu}\right] = D(\mu||\nu).$$

This implies

$$D(\mathbf{P^u}||\mathbf{P^0}) \geq D(\mu||\nu),$$

with equality if and only if $u = u^*$ and we have the minimal energy

$$D(\mu||\nu) = \min_u \frac{1}{2}\mathbf{E}\left[\int_0^1 \|u_s\|^2 ds\right].$$

■

In the special case when the prior process is the Wiener process, i.e. $b(x,t) \equiv 0$ and $X_1 \sim \gamma_d$, we can compute the value function and optimal control explicitly:

- compute the value function

$$\begin{aligned}v(x,t) &= -\log \mathbf{E}[g(X_1)|X_t = x] = -\log \mathbf{E}[f(W_1)|X_t = x] \\ &= -\log\left((2\pi(1-t))^{-d/2}\int_t^1 f(y)\exp(-\frac{\|y-x\|^2}{2(1-t)})dy\right) \\ &= -\log Q_{1-t}f(x)\end{aligned}$$

- compute the optimal control

$$u^*(x,t) = -\nabla v(x,t) = \underbrace{\nabla \log Q_{1-t} f(x)}_{\text{Föllmer drift}} \tag{2.19}$$

where Q denotes the Euclidean heat semigroup (1), and (2.19) is refered to as the *Föllmer drift*, following [21]. By the above theorem, the Föllmer drift has the minimal energy among all admissible controls that allow the sampling $X_1^u \sim \mu$ at $t = 1$ and this minimal energy it takes is exactly $D(\mu||\gamma_d)$.

2.4 Expressivity of the Föllmer Drift

After considering the exact sampling problem, we have examined the richness of this model in the aspect of representative power. We can generate samples according to a given arbitrary target distribution which is absolutely continuous to Gaussian distribution, and theoretically the optimal control with minimal energy we can make use of is the Föllmer drift. However, in practice, the Föllmer drift usually has no closed-form solution and we cannot play the exact sampling. For this reason, we want to resort to some approximate sampling methods. In this section, we look into the expressivity of the Föllmer drift, that is, to consider the problem: Could the Föllmer drift be well approximated by neural nets?

To observe the formula of the Föllmer drift (2.19), we have some ideas in mind. We want to

- replace $Q_t f(x)$ by Monte Carlo estimate
- replace $f(\cdot)$ by a neural net approximation $\hat{f}(\cdot;\theta)$
- replace the elementary operations (i.e. gradients, multiplications, reciprocals) by neural net approximations

$$\begin{aligned}\nabla \log Q_{1-t} f(x) &\approx \nabla \log \left\{ \frac{1}{N} \sum_{n=1}^{N} \hat{f}(x + \sqrt{1-t} z_n; \theta) \right\} \\ &= \frac{\sum_{n=1}^{N} \nabla \hat{f}(x + \sqrt{1-t} z_n; \theta)}{\sum_{n=1}^{N} \hat{f}(x + \sqrt{1-t} z_n; \theta)}\end{aligned}$$

Based on this, it leads to a sequence of assumptions. First we want the derivative f to be Lipschitz continuous.

Assumption 2.4.1 (*target distribution*)**.** *The function f is differentiable, both f and ∇f are L-Lipschitz, and there exits a constant $c \in (0,1]$ such that $f \geq c$ everywhere.*

This assumption would guarantee the following regularity for the Föllmer drift.

Lemma 2.4.1 (*regularity of the Föllmer drift*)**.** *With assumption 2.4.1, the Föllmer drift $b(x,t) = \nabla \log Q_{1-t} f(x)$ is bounded in the Euclidean norm,*

$$\|\nabla \log Q_{1-t} f(x)\| \leq \frac{L}{c} \tag{2.20}$$

for $x \in \mathbb{R}^d$, $t \in [0,1]$, and L the maximum of the Lipschitz constants of f and ∇f. And also, it is Lipschitz with Lipschitz constant $L/c + L^2/c^2$,

$$\|b(x,t) - b(x',t)\| \leq \left(\frac{L}{c} + \frac{L^2}{c^2}\right) \|x - x'\|. \tag{2.21}$$

Proof. For differentiable and Lipschitz $f : \mathbb{R}^d \to \mathbb{R}$, the heat semigroup and gradient operator commute ([37], Cor.2.2.8), i.e. $\nabla Q_t f = Q_t \nabla f$. Since f is Lipschitz with Lipschtz constant L, we have $\|\nabla f\| \leq L$ for all x, which implies $\|\nabla Q_t f(x)\| \leq L$. And also, by assumption $f(x) \geq c$, we have $Q_t f(x) \geq c$ for all $t \geq 0$ and $x \in \mathbb{R}^d$. Therefore, for any $x \in \mathbb{R}^d$ and $t \in [0,1]$

$$\|b(x,t)\| = \left\| \frac{\nabla Q_{1-t} f(x)}{Q_{1-t} f(x)} \right\| \leq \frac{L}{c}.$$

Furthermore, by the assumption that ∇f is Lipschitz, we have the term $\nabla Q_{1-t} f(x)$ also being Lipschitz

$$\|\nabla Q_t f(x) - \nabla Q_t f(x')\| \leq L \|x - x'\|,$$

for x, $x' \in \mathbb{R}^d$ and $t \in [0,1]$. Thus we have

$$\begin{aligned} \|b(x,t) - b(x',t)\| &= \left\| \frac{\nabla Q_{1-t} f(x)}{Q_{1-t} f(x)} - \frac{\nabla Q_{1-t} f(x')}{Q_{1-t} f(x')} \right\| \\ &\leq \frac{\|\nabla Q_t f(x) - \nabla Q_t f(x')\|}{Q_{1-t} f(x')} + \|b(x,t)\| \cdot \frac{|Q_t f(x) - Q_t f(x')|}{Q_{1-t} f(x')} \\ &\leq \left(\frac{L}{c} + \frac{L^2}{c^2}\right) \|x - x'\|. \end{aligned}$$

■

Next, we want to assume that the activation functions σ we use are differentiable and *universal*, which means that any univariate Lipschitz function on a bounded interval can be approximated arbitrarily well by a 2-layer feedforward neural net.

Assumption 2.4.2 (*differentiable and universal activation*)**.** *The activation function σ is differentiable. Moreover, it is universal: For any L-Lipschitz* *function $h : \mathbb{R} \to \mathbb{R}$ which is constant outside the interval $[-R, R]$, and for any $\delta > 0$, there exists real numbers $c_\sigma, a, \{(\alpha_i, \beta_i, \gamma_i)\}_{i=1}^m$, where $c_\sigma > 0$ depending on σ and $m \le c_\sigma \frac{RL}{\delta}$, such that the function*

$$\tilde{h}(x) = a + \sum_{i=1}^{m} \alpha_i \sigma(\beta_i x + \gamma_i) \tag{2.22}$$

satisfies

$$\sup_{x \in \mathbb{R}} |\tilde{h}(x) - h(x)| \le \delta.$$

Remark. *We refer to the functions of the form (2.22) as 2-layer (one input layer and one hidden layer) networks of the size $m + 1$. This assumption of universal property of activation functions goes back to Hornik's universal approximation theorem [13], which states that feedforward neural nets with only one hidden layer using arbitrary squashing functions (i.e. function $\sigma : \mathbb{R} \to [0, 1]$, non-decreasing, $\lim_{x \to \infty} \sigma(x) = 1$, and $\lim_{x \to -\infty} \sigma(x) = 0$) are capable of approximating any Borel measurable function from one finite dimensional space to another to any desired degree of accuracy, provided sufficiently many hidden units are available. For instance, the sigmoidal functions satisfy this property [6]. The rectified linear unit (ReLU) activation function $x \mapsto x \vee 0$ also satisfies the universal property, though it is not differentiable at 0. We can replace it by the softplus function $x \mapsto \log(1 + e^{cx})$, which will approximate ReLU when $c > 0$ increases.*

The last one, we want to make the assumption regarding the approximability of f by neural nets: Both f and ∇f can be efficiently approximated by a neural net on any compact subset of $\mathbb{R}^d$.

Assumption 2.4.3 (*convenient approximation by neural nets*)**.** *For any $R > 0$ and $\epsilon > 0$, there exists a l-layer feed forward neural net $\hat{f}$ of size (number of nodes) s, such that*

$$\sup_{x \in \mathsf{B}^d(R)} |f(x) - \hat{f}(x)| \le \epsilon \quad \text{and} \quad \sup_{x \in \mathsf{B}^d(R)} \|\nabla f(x) - \nabla \hat{f}(x)\| \le \epsilon$$

and $l, s \le poly(1/\epsilon, d, L, R)$.

With these assumptions, we state the following result.

Theorem 2.4.1 (*expressivity of the Föllmer drift*)**.** *Suppose assumptions A1-A3 are in force. Let L denote the maximum of Lipschitz constants of f and ∇f. Then for any $0 < \epsilon < 16L^2/c^2$, there exists a neural net $\hat{v} : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$ with size polynomial in $1/\epsilon, d, L, c, 1/c$, such that the activation function of each neuron is an element of the set $\{\sigma, \sigma', ReLU\}$ and the following holds: If $\{\hat{X}_t\}_{t \in [0,1]}$ is the diffusion process governed by the Itô SDE*

$$d\hat{X}_t = \hat{b}(\hat{X}_t, t)dt + dW_t, \quad \hat{X}_0 = 0,$$

with the drift $\hat{b}(x,t) = \hat{v}(x, \sqrt{1-t})$*, then* $\hat{\mu} :=$ *Law*$(\hat{X}_1)$ *satisfies*

$$D(\mu || \hat{\mu}) \leq \epsilon.$$

Proof. The proof of this theorem is divided into 3 steps:

- The first step is to prove that heat semigroup $Q_t f(x)$ can be approximated by a finite sum of the form $\frac{1}{N}\sum_{n\leq N} f(x+\sqrt{t}z_n)$, uniformly for $x \in \mathsf{B}^d(R)$ and all $t \in [0,1]$ with $z_1, ..., z_N \in \mathsf{B}^d(\mathcal{O}(\sqrt{d}\log N))$. See Section 2.4.1.
- The second step is to show that f can be approximated by neural nets. And further more, the Föllmer drift $\nabla \log Q_{1-t} f(x)$ can be approximated by a neural net using $\sigma, \sigma', \mathrm{ReLU}$ as activation functions. This is given in Section 2.4.2.
- Finally in Section 2.4.3, we show that the error resulted from above approximation can be upper-bounded using the Girsanov formula.

■

Remark. *All the preliminaries and auxiliary lemmas for this proof are included in Appendix B.*

2.4.1 Uniform Approximation of the Heat Semigroup

Theorem 2.4.2. *For* $\epsilon > 0$*,* $R > 0$*, there exist* $N =$ *poly*$(1/\epsilon, d, L, R)$ *points* $z_1, ..., z_N \in \mathbb{R}^d$ *bounded by*

$$\max_{n\leq N} \|z_n\| \leq 8\sqrt{(d+6)\log N}$$

such that the following holds:

$$\sup_{x\in\mathsf{B}^d(R)} \sup_{t\in[0,1]} \left| \frac{1}{N}\sum_{n=1}^{N} f(x+z_n) - Q_t f(x) \right| \leq \epsilon$$

$$\sup_{x\in\mathsf{B}^d(R)} \sup_{t\in[0,1]} \left\| \frac{1}{N}\sum_{n=1}^{N} \nabla f(x+\sqrt{t}z_n) - \nabla Q_t f(x) \right\| \leq \epsilon$$

Proof. We will use probabilistic method. Let $Z_1, ..., Z_N$ be i.i.d. copies of $Z \sim \gamma_d$. Given $R > 0$, $\epsilon > 0$, define the following events:

$$E_1 := \left\{ \max_{n\leq N} \|z_n\| > 8\sqrt{(d+6)\log N} \right\}$$

$$E_2 := \left\{ \sup_{x\in\mathsf{B}^d(R)} \sup_{t\in[0,1]} \left| \frac{1}{N}\sum_{n=1}^{N} f(x+z_n) - Q_t f(x) \right| > \epsilon \right\}$$

$$E_3 := \left\{ \max_{i\in[d]} \sup_{x\in\mathsf{B}^d(R)} \sup_{t\in[0,1]} \left| \frac{1}{N}\sum_{n=1}^{N} \partial_i f(x+\sqrt{t}z_n) - \partial_i Q_t f(x) \right| > \epsilon \right\}$$

We want to show

$$\mathbf{P}(\neg E_1 \cap \neg E_2 \cap \neg E_3) > 0$$

or equivalently

$$\mathbf{P}(E_1 \cup E_2 \cup E_3) < 1$$

which implies that there exists a realization of $z_1, ..., z_N$ verifying Theorem 2.4.2.

Let $U = \|Z\|$. By Lemma B.1.4, we have $\|U\|_{\psi_2} \leq \sqrt{d} + \sqrt{6}$. Then by the maximal inequality in Lemma B.5, the term $U_N^* := \max_{n \leq N} U_n$ satisfies

$$\|U_N^*\|_{\psi_2} \leq \sqrt{32(d+6)\log N}$$

Follow from Lemma B.1.1, using the tail bound estimate,

$$\mathbf{P}(E_0) \leq \mathbf{P}\left(U_N^* \geq \sqrt{2}\|U_N^*\|_{\psi_2}\right) \leq \frac{1}{e^2 - 1} \leq \frac{1}{4}$$

Moreover, by the Assumption 2.4.3, both f and its gradient ∇f are L-Lipschitz. And the heat semigroup commutes with the gradient operator, $\partial_i Q_t f = Q_t \partial_i f$ ([37], Cor.2.2.8). We then use Lemma B.1.8 with $C > 0$, $\gamma = \log 4(d+1)$ and choose

$$N = \left\lceil \left(\frac{C\sqrt{d}}{\epsilon} \cdot L\left((R \vee 1) + \sqrt{d} + \sqrt{6}\right) \cdot \left(16\sqrt{6\pi R d} + 5\sqrt{\log 4(d+1)}\right)\right)^2 \right\rceil$$

it gives

$$\mathbf{P}(E_2 \cup E_3) \leq \frac{1}{4}$$

therefore

$$\mathbf{P}(E_1 \cup E_2 \cup E_3) \leq \frac{1}{2}.$$

■

2.4.2 Uniform Approximation of the Föllmer Drift

Theorem 2.4.3. *Let $0 < \epsilon < 4L/c$ and $R > 0$, then there exists a neural net $\hat{v} : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ of size polynomial in $1/\epsilon, d, L, R, c, 1/c$, such that the activation function of each neuron is an element of the set $\{\sigma, \sigma', ReLU\}$, and the following holds*

$$\sup_{x \in \mathsf{B}^d(R)} \sup_{t \in [0,1]} \left\| \hat{v}(x, \sqrt{t}) - \nabla \log Q_t f(x) \right\| \leq \epsilon$$

and

$$\max_{i\in[d]}\sup_{x\in\mathbb{R}^d}\sup_{t\in[0,1]}|\hat{v}_i(x,\sqrt{t})|\leq\frac{2L}{c}.$$

Proof. By Theorem 2.4.2, the heat semigroup $Q_tf(x)$ can be approximated by a finite sum of the form $\varphi := N^{-1}\sum_{n=1}^{N} f(x+tz_n)$. For $\delta > 0$, there exists N =poly$(1/\delta, d, L, R)$ points $z_1, ..., z_N \in \mathbb{R}^d$ bounded by $\max_{n\leq N}\|z_n\| \leq 8\sqrt{(d+6)}$, and the following holds

$$\begin{aligned}\sup_{x\in\mathsf{B}^d(R)}\sup_{t\in[0,1]}|\varphi(x,\sqrt{t})-Q_tf(x)|\leq\delta\\ \sup_{x\in\mathsf{B}^d(R)}\sup_{t\in[0,1]}|\nabla\varphi(x,\sqrt{t})-\nabla Q_tf(x)|\leq\delta\end{aligned}\tag{2.23}$$

Moreover, by Assumption 2.4.3, there exists a feedforward neural net $\hat{f}$ of the size less than poly$(1/\delta, d, L, R)$, which approximates f and ∇f to accuracy of δ on the ball $\mathsf{B}^d(R+R_N)$

$$\sup_{x\in\mathsf{B}^d(R+R_N)}|f(x)-\hat{f}(x)|\leq\delta\ \ \textit{and}\ \ \sup_{x\in\mathsf{B}^d(R+R_N)}\|\nabla f(x)-\nabla\hat{f}(x)\|\leq\delta.\tag{2.24}$$

where $R_N = \max_{n\leq N}\|z_n\|$.

Now we want to use the function of the form

$$\hat{\varphi}:\mathbb{R}^d\times[0,1]\to\mathbb{R},\ \hat{\varphi}(x,t):=\frac{1}{N}\sum_{n=1}^{N}\hat{f}(x+tz_n)$$

to approximate the Föllmer drift. By (2.23) and (2.24), we have

$$\begin{aligned}&\sup_{x\in\mathsf{B}^d(R)}\sup_{t\in[0,1]}|\hat{\varphi}(x,\sqrt{t})-Q_tf(x)|\\ &\leq\sup_{x\in\mathsf{B}^d(R)}\sup_{t\in[0,1]}|\hat{\varphi}(x,\sqrt{t})-\varphi(x,\sqrt{t})|+\sup_{x\in\mathsf{B}^d(R)}\sup_{t\in[0,1]}|\varphi(x,\sqrt{t})-Q_tf(x)|\\ &\leq\sup_{x\in\mathsf{B}^d(R+R_N)}|\hat{f}(x)-f(x)|+\sup_{x\in\mathsf{B}^d(R)}\sup_{t\in[0,1]}|\varphi(x,\sqrt{t})-Q_tf(x)|\leq2\delta\end{aligned}\tag{2.25}$$

and also

$$\begin{aligned}&\sup_{x\in\mathsf{B}^d(R)}\sup_{t\in[0,1]}\|\nabla\hat{\varphi}(x,\sqrt{t})-\nabla Q_tf(x)\|\\ &\leq\sup_{x\in\mathsf{B}^d(R)}\sup_{t\in[0,1]}\|\nabla\hat{\varphi}(x,\sqrt{t})-\nabla\varphi(x,\sqrt{t})\|+\sup_{x\in\mathsf{B}^d(R)}\sup_{t\in[0,1]}\|\nabla\varphi(x,\sqrt{t})-\nabla Q_tf(x)\|\\ &\leq\sup_{x\in\mathsf{B}^d(R+R_N)}\|\nabla\hat{f}(x)-\nabla f(x)\|+\sup_{x\in\mathsf{B}^d(R)}\sup_{t\in[0,1]}\|\nabla\varphi(x,\sqrt{t})-\nabla Q_tf(x)\|\leq2\delta\end{aligned}\tag{2.26}$$

By Assumption 2.4.1, f is L-Lipschitz and bounded below by $c \in (0,1]$. Therefore, we know

$$\begin{aligned} c \le Q_t f(x) &\le L(\|x\| + \mathbf{E}\|Z\|) + f(0) \\ &\le L(\|x\| + \sqrt{d}) + f(0) \end{aligned} \tag{2.27}$$

for any $x \in \mathbb{R}^d$ and $t \in [0,1]$. Choose $\delta = \frac{c^2\epsilon}{16L}$ and ϵ small enough such that $\delta \le \frac{c}{4}$. Then by (2.25) and (2.27) we obtain the following bound on $\mathsf{B}^d \times [0,1]$,

$$\frac{c}{2} \le \hat{\varphi}(x, \sqrt{t}) \le L(R + \sqrt{d}) + f(0) + \frac{c}{2} \tag{2.28}$$

Without loss of generality, we assume $L \ge 1$. Then by (2.25), (2.26), (2.28) and (2.20) in Lemma 2.4.1, we have the Föllmer drift $\nabla \log Q_t f(x)$ approximated by $\nabla \log \hat{\varphi}(x, \sqrt{t})$ on $\mathsf{B}^d(R) \times [0,1]$,

$$\begin{aligned} &\|\nabla \log \hat{\varphi}(x, \sqrt{t}) - \nabla \log Q_t f(x)\| = \left\| \frac{\nabla \hat{\varphi}(x, \sqrt{t})}{\hat{\varphi}(x, \sqrt{t})} - \frac{\nabla Q_t f(x)}{Q_t f(x)} \right\| \\ &\le \frac{1}{\hat{\varphi}(x, \sqrt{t})} \|\nabla \hat{\varphi}(x, \sqrt{t}) - \nabla Q_t f(x)\| + \left\| \frac{\nabla Q_t f(x)}{Q_t f(x)} \right\| \frac{|\hat{\varphi}(x, \sqrt{t} - Q_t f(x)|}{\hat{\varphi}(x, \sqrt{t})} \\ &\le \frac{2L}{c} \cdot 2\delta + \frac{L}{c} \cdot \frac{2}{c} \cdot 2\delta \\ &\le \frac{\epsilon}{2}. \end{aligned} \tag{2.29}$$

It remains to show we can approximate $\nabla \log \hat{\varphi}(x, \sqrt{t})$ by a feedforward neural net to the accuracy of $\epsilon/2$. Notice that elementary operations such as multiplication, reciprocal, and gradient can be approximated by neural nets (see Appendix B.2). We represent $\nabla \log \hat{\varphi}(x, \sqrt{t})$ as a composition of these operations and then approximate each step by a neural net.

Consider the following computation graph:

1. Compute $a = \hat{\varphi}(x, \sqrt{t})$

↓

2. Compute $b_i = \partial_i \hat{\varphi}(x, \sqrt{t})$

↓

3. Compute $r = 1/a$

↓

4. Compute $v_i = r b_i$

In step 1, for given x and $\sqrt{t}$, $a = \hat{\varphi}$ can be computed by a neural net with activation function σ of size $\mathrm{poly}(1/\epsilon, d, L, R)$ and depth $\mathrm{poly}(1/\epsilon, d, L, R)$ according to the Assumption 2.4.3. In step 2, by the cheap gradient principle

(Lemma B.2.3), the coordinate of the gradient $\nabla\hat{\varphi}$ can be computed by a neural net of size poly$(1/\epsilon, d, L, R)$ using the elements of $\{\sigma, \sigma'\}$ as activation functions in each neuron. In step 3, since $a \in [c/2, L(R+\sqrt{d})+f(0)+c/2]$, by Lemma B.2.2, the reciprocal $r = 1/a$ can be approximated to the accuracy $\epsilon/(4L\sqrt{d})$ by a 2-layer neural net with activation function σ of size

$$\mathcal{O}\left(\frac{4}{c^2}\cdot\Big(L(R+\sqrt{d})+f(0)+c/2\Big)\cdot\frac{4L\sqrt{d}}{\epsilon}\right) \leq \text{poly}(1/\epsilon, d, L, R, c, 1/c)$$

We denote the resulting approximation in step 3 by $\hat{r}$, which satisfies $|\hat{r}| \leq 2/c + \epsilon/(4L\sqrt{d}) \leq 4/c$. And since f is assumed to be L-Lipschitz, we $|b_i| \leq 2L$. Therefore, in step 4, by the Lemma B.2.1, we can approximate the product $\hat{r}b_i$ to the accuracy $\epsilon/4\sqrt{d}$ by a 2-layer neural net with activation function σ of size

$$\mathcal{O}\left((4/c \vee 2L)^2\cdot\frac{4\sqrt{d}}{\epsilon}\right) \leq \text{poly}(1/\epsilon, d, L, 1/c)$$

We need to estimate the extra error resulted from the approximation in both step 3 and step 4

$$|\hat{v}_i - v_i| \leq |\hat{v}_i - \hat{r}_i b_i| + |\hat{r}_i b_i - r_i b_i| \leq \frac{\epsilon}{2\sqrt{d}} \tag{2.30}$$

Therefore, the vector $v = (v_1, ..., v_d)$ can be $\epsilon/2$-approximated by $\tilde{v}(x, \sqrt{t}) : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$, $\tilde{v} := (\hat{v}_1, ..., \hat{v}_d)$,which is a vector-valued feedforward neural net of size poly$(1/\epsilon, d, L, R, c, 1/c)$. Overall, by (2.29) and (2.30) we have

$$\begin{aligned}&\left\|\tilde{v}(x,\sqrt{t}) - \nabla\log Q_t f(x)\right\|\\ &\qquad\leq \|\tilde{v}(x,\sqrt{t}) - v(x,\sqrt{t})\| + \|v(x,\sqrt{t}) - \nabla\log Q_t f(x)\| \leq \epsilon\end{aligned}$$

Finally, since $|\tilde{v}_i| \leq 2L/c$ on $\mathsf{B}^d(R)\times[0,1]$, we let

$$\hat{v}_i(x,\sqrt{t}) := \Big(\tilde{v}_i(x,\sqrt{t}) \vee (-2L/c)\Big) \wedge (2L/c)$$

which takes value in $[-2L/c, 2L/c]$ and coincides with $\tilde{v}_i$ on $\mathsf{B}^d(R)\times[0,1]$. The operation $\vee$ and $\wedge$ can be implemented by the ReLU neuron.

■

2.4.3 Upper-bound of the Approximation Error

In this part, by using the Girsanov formula, we want to upper-bound the error that resulted from the neural net approximation of the Föllmer Drift.

From Theorem 2.4.3, there exists a neural net $\hat{v} : \mathbb{R}^d \to \mathbb{R}^d$ satisfying

$$\sup_{x \in \mathsf{B}^d(R)} \sup_{t \in [0,1]} \left\| \hat{v}(x, \sqrt{t}) - \nabla \log Q_t f(x) \right\| \leq \epsilon \tag{2.31}$$

and

$$\max_{i \in [d]} \sup_{x \in \mathbb{R}^d} \sup_{t \in [0,1]} |\hat{v}_i(x, \sqrt{t})| \leq \frac{2L}{c}. \tag{2.32}$$

For $\boldsymbol{\mu}$:=Law($X_{[0,1]}$) and $\hat{\boldsymbol{\mu}}$:=Law($\hat{X}_{[0,1]}$), by the Girsanov formula

$$\begin{aligned} D(\boldsymbol{\mu}||\hat{\boldsymbol{\mu}}) &= \frac{1}{2} \mathbf{E} \int_0^1 \|b(X_t, t) - \hat{b}(X_t, t)\|^2 dt \\ &\overset{Fubini}{=} \frac{1}{2} \int_0^1 \mathbf{E}\|b(X_t, t) - \hat{b}(X_t, t)\|^2 dt \end{aligned}$$

where in the second equality we use the Fubini's theorem, since both $b(X_t, t)$ and $\hat{b}(X_t, t)$ are bounded by Lemma 2.4.1 and (2.32).
Next, we divide the integral into two parts

$$\begin{aligned} &\mathbf{E}\|b(X_t, t) - \hat{b}(X_t, t)\|^2 \\ =& \underbrace{\mathbf{E}\left[\|b(X_t, t) - \hat{b}(X_t, t)\|^2 \cdot \mathbf{1}_{\{X_t \in \mathsf{B}^d(R)\}}\right]}_{T_1 :=} + \underbrace{\mathbf{E}\left[\|b(X_t, t) - \hat{b}(X_t, t)\|^2 \cdot \mathbf{1}_{\{X_t \notin \mathsf{B}^d(R)\}}\right]}_{T_2 :=} \end{aligned}$$

We see that $T_1 \leq \epsilon$ by (2.31). It is left to estimate T_2. Since $\|b(x, t)\| \leq L/c$ by Lemma 2.4.1, use the triangle inequality we have

$$\|X_t\| \leq \|W_t\| + Lt/c$$

for any t. Now we see that the process $\|W_t\| + Lt/c$ is a non-negative submartingale, then by Doob's maximal inequality

$$\mathbf{P}\left(\sup_{t \in [0,1]} \|X_t\| \geq R\right) \leq \frac{\mathbf{E}[\,\|W_1\| + L/c\,]}{R} \leq \frac{\sqrt{d} + L/c}{R}$$

Therefore

$$T_2 = \mathbf{E}\|b(X_t, t) - \hat{b}(X_t, t)\|^2 \cdot \mathbf{P}\left(X_t \notin \mathsf{B}^d(R)\right) \leq \frac{9dL^2}{c^2} \frac{\sqrt{d} + L/c}{R}$$

If we choose R large enough, it will guarantee $T_2 < \epsilon$ and obtain $D(\boldsymbol{\mu}||\hat{\boldsymbol{\mu}}) \leq \epsilon$. Then by the data processing inequality [4], $D(\mu||\hat{\mu}) \leq D(\boldsymbol{\mu}||\hat{\boldsymbol{\mu}}) \leq \epsilon$. We finish the proof of Theorem 2.4.1.

Chapter 3

Trainability: Scalable Gradients for Neural SDEs

The main goal of this chapter is to discuss the trainability of the latent diffusion generative model that we have introduced. In last chapter, we explore its properties in different aspects. In the discussion on the problem of exact sampling, we have known that the model has sufficient richness in representative power. And then we examine its expressivity, that is, the possibility that its optimal control drift can be well approximated by a feedforward neural net. And it turns out this is theoretically feasible with some proper assumptions. Except for these, we also know at the beginning that the model learning can actually operate using the method of variational inference by computing the gradients of the variational free energy with respect to parameters and then updating these parameters according to gradients until the free energy converges. Thus here it remains us the problem: in order to enable the use of some widely used model learning methods, such as the stochastic gradient decent, can we efficiently compute these gradients? and second, how scalable is it of our algorithm? We formulate this as a problem of sensitivity analysis which was introduced by Gobet and Munos in 2005 [10].

3.1 The Problem of Sensitivity Analysis

Consider the following problem: given a d-dimensional Itô process,

$$dX_t^\alpha = b(X_t^\alpha, t; \alpha)dt + \sigma(X_t^\alpha, t; \alpha)dW_t; \quad t \in [0, T] \tag{3.1}$$

where α is a n_α-dimensional parameter, and a function $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ is given. We want to compute the gradient of the expectation

$$J(\alpha) := \mathbf{E}[\mathcal{L}(X_T^\alpha)]$$

with respect to α.

Remark. *In the context of variational inference for Neural SDE,* $\alpha = (\phi, \theta)$*,* $\mathcal{L}(\cdot) = \mathbf{F}^u(y; \phi, \theta)(\cdot)$

There are several possible approaches to compute this gradient. The simplest one could be the *resampling method* (or *finite difference*). One computes different values of $J(\alpha)$ for some close values of the parameter α then compute the difference to approximate the derivative

$$\frac{\partial J}{\partial \alpha_i} \approx \frac{J(\alpha + \epsilon e_i) - J(\alpha)}{\epsilon}$$

However, this method is not only costly when the dimension of the parameters is large, but also causes numerical errors, such as the round-off error and the truncation error.

Another option is the so-called *likelihood method* or *score method* [10] in which the gradient is rewritten as

$$\nabla_\alpha J = \mathbf{E}[\mathcal{L}(X_T^\alpha) H]$$

for some random variable H. Nevertheless, H usually depends on the density of X_T^α which is difficult to compute, especially in the context of neural SDE, where we have less assumptions on coefficient functions of the diffusion process.

The *path-wise* method is as well a possible solution, which interchanges the gradient operator ∇ and the expectation

$$\nabla_\alpha J = \nabla_\alpha \mathbf{E}[\mathcal{L}(X_T^\alpha)] = \mathbf{E}[\nabla \mathcal{L}(X_T^\alpha) \nabla_\alpha X_T^\alpha]$$

and we need to compute the state sensitivity

$$\frac{\partial X_t}{\partial \alpha_i} = \int_0^t \left(\frac{\partial b_s}{\partial \alpha_i} + \frac{\partial b_s}{\partial x} \frac{\partial X_s}{\partial \alpha_i} \right) ds + \sum_{l=1}^m \int_0^t \left(\frac{\partial \sigma_{s,l}}{\partial \alpha_i} + \frac{\partial \sigma_{s,l}}{\partial x} \frac{\partial X_s}{\partial \alpha_i} \right) dW_s^l \quad (3.2)$$

The limitation of this method is that the function $\mathcal{L}$ has to be smooth and also the compuation of $\nabla_\alpha X_t$ is not numerically scalable, causing high computational complexity when the dimension of parameter increases. Note that in our setting, the coefficient functions b and σ in the neural SDE are implemented by neural nets, therefore the dimension of the parameter would not be a small number.

After these attempts from the above methods, we want to seek for an alternative, which can generally overcome the drawbacks that we have mentioned. In the following, we introduce the *stochastic adjoint sensitivity method.*

The general idea of the *adjoint method* goes back to Pontryagin's work in 1962 [24]: instead of using the forward sensitivity as (3.2) for gradient computation, one tries to construct a backward SDE of the so-called *adjoint state* whose solution would be the objective gradient we want. We present this method here from the more modern perspective of *standard back-propagation*, which is short and easy to follow. Note that in the special case when it is deterministic, i.e. diffusion function $\sigma = 0$, this is an instance of the adjoint method which was derived for the *Neural ODE* framework [7].

The *standard back-propagation* [28] is a popular method used in training feedforward neural nets for supervised learning. It states that the gradient of the observation (or loss function) $\mathcal{L}$ with respect to the hidden state h_t is dependent on the gradient from next layer h_{t+1} by the chain rule

$$\frac{d\mathcal{L}}{dh_t} = \frac{d\mathcal{L}}{dh_{t+1}} \frac{dh_{t+1}}{dh_t} \tag{3.3}$$

This equation generally describes how the gradient is propagating backward. Inspired by this, we outline the steps we will take in the adjoint method:

1. First, we define the adjoint state to be $a_t = d\mathcal{L}/dx_t$, with x_t to be some state of the diffusion process (3.1);

2. Derive a backward dynamics for the adjoint state which shows how the gradient propagate backward. In other words, we want to derive a formula for the the dynamics of dh_{t+1}/dh_t;

3. Treating the parameters as an additional part of the augmented state;

4. By integrating backward, we finally obtain the total gradient.

The purpose to consider this method lies in the reason that it can immensely reduce computational complexity when it is compared to the path-wise method. Since the adjoint state we make use of does not depend on the state sensitivity (3.2) as well as the parameters, it avoids to compute the Jacobian and reduces the complexity of one gradient evaluation from $n_\alpha + 1$ systems to only 2 systems, which implies that state dimension involved goes from $n_x(n_\alpha + 1)$ to $3n_x + n_\alpha$. Therefore, it would be scalable when the dimension of parameter increases largely. In the next sections, we explore the details of the above steps when this adjoint method is applied to the sensitivity problem (3.1).

3.2 Backward Calculus

We collect some preliminaries on backward calculus. We follow the treatment of Kunita [18]. Let $\{\mathcal{F}_{s,t}\}_{s\leq t;\, s,t\in\mathbb{T}}$ be the two-sided filtration

$$\mathcal{F}_{s,t} := \sigma(W_u - W_v : s \leq u < v \leq t),\ s, t \in \mathbb{T}(:= [0, T])$$

which can be adjusted to be a forward or backward filtration.

Definition 3.2.1 (*backward Stratonovich integral*)**.** The *backward Wiener process* $\{\check{W}_t\}_{t\in\mathbb{T}}$ is defined as $\check{W}_t = W_t - W_T$ for $t \in \mathbb{T}$, which is adapted to the backward filtration $\{\mathcal{F}_{t,T}\}_{t\in\mathbb{T}}$. Then for continuous semimartingale $\check{Y}_t$ which is adapted to the backward filtration, we define the *backward Stratonovich integral* as

$$\int_s^T \check{Y}_t \circ d\check{W}_t = \lim_{|\Pi|\to 0} \sum_{k=1}^{N} \frac{1}{2} \left(\check{Y}_{t_k} + \check{Y}_{t_{k-1}}\right) \left(\check{W}_{t_{k-1}} - \check{W}_{t_k}\right)$$

where $\Pi = \{0 = t_N < ... < t_0 = T\}$ is a backward partition on $[0, T]$ and the limit is considered in the L^2 sense.

Remark. *The Stratonovich and Itô integrals differ by a term of finite variation. The reason we consider Stratonovich integral here is that the way it is defined results in the fact that the classical chain rule of ordinary calculus holds, which would be easier to manipulate.*

Consider the Stratonovich stochastic differential equation

$$Z_T = z_0 + \int_0^T b(Z_t, t)dt + \sum_{i=1}^{m} \int_0^T \sigma_i(Z_t, t) \circ dW_t^{(i)} \tag{3.4}$$

We assume b, $\sigma \in C_b^{\infty,1}$, so that the SDE has unique strong solution. Denote by $\Phi_{s,t}(z) := Z_t^{s,z}$ the solution at time t when the process started at z at time s.

Definition 3.2.2 (*stochastic flow*)**.** Given a realization of the Wiener process, the collection $\mathcal{S} = \{\Phi_{s,t}\}_{s\leq t;s,t\in\mathbb{T}}$ of continuous maps from $\mathbb{R}^d$ to itself is called the *stocahstic flow* generated by the SDE (3.4), which satisfies the flow property

$$\Phi_{s,u}(z) = \Phi_{t,u} \circ \Phi_{s,t}(z)\,, \quad s < t < u,\, z \in \mathbb{R}^d$$

By the Theorem 3.7.1 in [18], we know that each $\Phi_{s,t}$ is a smooth diffeomorphism $\mathbb{R}^d \to \mathbb{R}^d$. And the *backward flow* $\check{\Psi}_{s,t} := \Phi_{s,t}^{-1}$ satisfies the following backward SDE:

$$\check{\Psi}_{s,t}(z) = z - \int_s^t b(\check{\Psi}_{s,t}(z), u)du - \sum_{i=1}^{m} \int_s^t \sigma_i(\check{\Psi}_{s,t}(z), u) \circ d\check{W}_u^{(i)} \tag{3.5}$$

for $z \in \mathbb{R}^d$, $s, t \in \mathbb{T}$, $s \leq t$. (3.4) and (3.5) differ by a negative sign. This symmetry is resulted from the use of Stratonovich integral.

3.3 Stochastic Adjoint Sensitivity

With the preliminaries, now we want to derive the backward dynamic which prescribes the back-propagation of the gradient. First of all, recall the equation $\frac{d\mathcal{L}}{dh_t} = \frac{d\mathcal{L}}{dh_{t+1}}\frac{dh_{t+1}}{dh_t}$ from standard back-propagation, we consider the quantity

$$A_{s,t}(z) := \nabla(\mathcal{L}(\Phi_{s,t}(z))) = \nabla\mathcal{L}(\Phi_{s,t}(z))\nabla\Phi_{s,t}(z)$$

and let the adjoint state to be

$$\check{A}_{s,T}(z) := A_{s,T}(\check{\Psi}_{s,T}(z)) = \underbrace{\nabla\mathcal{L}(z)}_{\check{A}_T}\underbrace{\nabla\Phi_{s,T}(\check{\Psi}_{s,T}(z))}_{\partial Z_T/\partial Z_s} \tag{3.6}$$

Note that $\nabla\mathcal{L}(z)$ is the terminal value of $\check{A}_{s,T}(z)$ at $s = T$. And the term $\nabla\Phi_{s,T}(\check{\Psi}_{s,T}(z))$ is actually the state derivative $\partial Z_T/\partial Z_s$. Definition of $\check{A}_{s,T}(z)$ can be interpreted as the gradient of the observation $\mathcal{L}$ with respect to the initial state $z \in \mathbb{R}^d$ when we consider the diffusion process goes from time s to T.

Let $J_{s,t}(z) := \partial Z_s/\partial Z_t = \nabla\check{\Psi}_{s,t}(z)$ for $0 \leq s \leq t \leq T$. Taking the gradient with respect to z on both sides of (3.5) we have

$$J_{s,t}(z) = I_d - \int_s^t \nabla b(\check{\Psi}_{r,t}(z), r)J_{r,t}(z)dr - \sum_{i=1}^m \int_s^t \nabla\sigma_i(\check{\Psi}_{r,t}(z), r)J_{r,t}(z)\circ d\check{W}_r^{(i)} \tag{3.7}$$

Then we can derive the dynamics of $\partial Z_t/\partial Z_s$ as the inverse of $J_{s,t}(z)$. Let $K_{s,t}(z) := J_{s,t}(z)^{-1} = \nabla\Phi_{s,t}(\check{\Psi}_{s,t}(z))$. By the Stratonovich version of Itô's formula and (3.7), it satisfies

$$K_{s,t}(z) = I_d + \int_s^t K_{r,t}(z)\nabla b(\check{\Psi}_{r,t}(z), r)dr + \sum_{i=1}^m \int_s^t K_{r,t}(z)\nabla\sigma_i(\check{\Psi}_{r,t}(z), r)\circ d\check{W}_r^{(i)} \tag{3.8}$$

Now from (3.6), we have $\check{A}_{s,T}(z) = \check{A}_T K_{s,T}(z)$ and it gives

$$\begin{aligned}\check{A}_{s,T}(z) = \nabla\mathcal{L}(z) &+ \int_s^T \check{A}_{r,T}(z)^\intercal\nabla b(\check{\Psi}_{r,T}(z), r)dr \\ &+ \sum_{i=1}^m \int_s^T \check{A}_{r,T}(z)^\intercal\nabla\sigma_i(\check{\Psi}_{r,T}(z), r)\circ d\check{W}_r^{(i)}\end{aligned} \tag{3.9}$$

This is exactly the backward dynamics for the adjoint state that we have been looking for.

In the final step, we want to extend this result to give the gradient with respect to parameters of the drift and diffusion function by treating them as an additional part of the augmented state whose dynamics has zero drift and diffusion. Therefore, now we consider augmented state $Y_t := (Z_t, \alpha)$, which satisfies a Stratonovich SDE with the drift function $\tilde{b}(y,t) = (b(z,t), \mathbf{0}_{n_\alpha})$ and the diffusion function $\tilde{\sigma_i}(y,t) = (\sigma_i(y,t), \mathbf{0}_{n_\alpha})$. Denote by $\check{A}^\alpha$ the adjoint state with respect to the parameters, and write the backward SDE for augmented adjoint $\check{A}^y := (\check{A}^z, \check{A}^\alpha)$ separately, we get the total gradient with respect to parameters

$$\begin{aligned}\check{A}^\alpha_{s,T}(z) = \nabla_\alpha \mathcal{L}(z) + \int_s^T \check{A}^z_{r,T}(z)^\intercal \nabla_\alpha b(\check{\Psi}_{r,T}(z), r) dr \\ + \sum_{i=1}^m \int_s^T \check{A}^z_{r,T}(z)^\intercal \nabla_\alpha \sigma_i(\check{\Psi}_{r,T}(z), r) \circ d\check{W}^{(i)}_r \qquad (3.10)\end{aligned}$$

The Stratonovich integral gives a nice expression for the backward dynamics due to its symmetry. These results can also be converted to the version of Itô integral. So far, we have finished the main steps of the stochastic adjoint method. To design the algorithm, first we need to solve the forward dynamics (3.4), then using the terminal state and the same Wiener sample path, we run the backward system that consists of (3.5), (3.9) and (3.10). Algorithm 2 summarizes this procedure.

Algorithm 2: Stochastic Ajoint Sensitivity (Stratonovich)

```
Input: parameters $\alpha$, start time $t_0$, stop time $t_1$, final state $z_{t_1}$,
       observation gradient $\partial\mathcal{L}/z_{t_1}$, drift function $b(z,t,\alpha)$,
       diffusion function $\sigma(z,t,\alpha)$, Wiener process sample path
       $w(t)$.
def augmented drift $\bar{b}(z_t, a_t, t, \alpha)$:
    return $[-b(z_t, -t, \alpha),\ a_t^\intercal \partial b/\partial z,\ a_t^\intercal \partial b/\partial \alpha]$
def augmented diffusion $\bar{\sigma}(z_t, a_t, t, \alpha)$:
    return $[-\sigma_i(z_t, -t, \alpha),\ a_t^\intercal \partial \sigma_i/\partial z,\ a_t^\intercal \partial \sigma_i/\partial \alpha]$
def replicated noise $\bar{w}(t)$:
    return $[-w(-t),\ -w(-t),\ -w(-t)]$
$\begin{bmatrix} z_{t_0} \\ \partial\mathcal{L}/\partial z_{t_0} \\ \partial\mathcal{L}/\partial\alpha \end{bmatrix} = \texttt{SDESolver}\left(\begin{bmatrix} z_{t_1} \\ \partial\mathcal{L}/\partial z_{t_1} \\ \mathbf{0}_{n_\alpha} \end{bmatrix}, \bar{b}, \bar{\sigma}, \bar{w}, -t_1, -t_0\right)$
return $\partial\mathcal{L}/\partial z_{t_0},\ \partial\mathcal{L}/\partial\alpha$
```

There are different numerical solvers that we could use to compute the solutions for above SDEs. For example, the Euler-Maruyama and the Milstein method are two possible options. In contrast to some existing approaches, such as the path-wise method, this method has nearly the same time and memory complexity as simply solving the SDE. Table 3.1 shows the asymptotic complexity comparison.

In practice, to successfully implement this algorithm it still remains a technical issue that one needs to design some special data structure such that

Table 3.1: L denotes the numbers of steps in the SDE solving. n_α is the dimension of the parameter, and d is the dimension of the system state.

Method	Memory	Time
Path-wise Forward Sensitivity	$\mathcal{O}(1)$	$\mathcal{O}(L \cdot (n_\alpha + d))$
Adjoint Sensitivity	$\mathcal{O}(1)$	$\mathcal{O}(L)$

the same Wiener sample path in the forward pass can be easily queried in the backward pass. Just to simply store the the Brownian motion increments, however, will cause a large memory computation as well as disable using the adaptive time stepping integrators. Therefore, here it requires a more sophisticate construction of the data structure being used. We will not delve into the details of these techniques here, which will be beyond the scope of this thesis.

Chapter 4

Conclusion

4.1 Summary and Discussion

In general, this generative model with latent diffusion we have studied allows irregularly-sampled time series modelling. That is, it can be used to model the time series with non-uniform time intervals that actually occur in many applications. Moreover, the time-continuous structure in its latent object permits it a high representative power, since it works as a generative model with this latent object of continuous depth. In other words, infinitely many hidden layers are configured together in the latent diffusion. Especially, when examining the sampling problem, we see that in theory, even the setting of the diffusion function is simplified (set to be identity matrix), this model would still be able to generate samples according to some arbitrarily given target distribution that is absolutely continuous with respect to the Gaussian distribution. This shows from the side how the representative power of this model is. As for the Theorem 2.4.1, on the one hand, it shows the expressivity of the Föllmer drift, that is, we can use neural nets to approximate this drift term and it provides a theoretical guarantee that we can resort to approximate method and numerically compute the models with latent diffusions. On the other hand, this theorem also gives us an idea that deep neural nets can always be applied to provide the terms of non-linearity in probabilistic models, which is much more convenient and efficient than some other existing approaches. At the end, we discuss different possible methods to compute the gradient of variational free energy in the context of neural SDE, and we introduce the method of stochastic adjoint sensitivity, which is memory efficient and scales well with the parameter dimension. This actually opens up a broad set of opportunities for fitting any differentiable SDE models. Derivative models and Greeks' computation in finance is one of the instances. However, one possible drawback is that, it requires the use of SDE solvers, which may not be working in some circumstance due to the underlying property of the stochastic differential equation, such as the stiffness.

Finally, another possible issue worth mentioning is that, through out this thesis we use the Kullback-Leibler divergence to measure the distance between two probability distributions. This would be problematic when we are dealing with high dimensional distributions that are supported by low dimensional manifolds, because in this case two distributions may not have intersection at all, and the divergence will be simply undefined. Therefore, a further survey could be of our interest is to consider and compare different ways to define a measure distance, which would also have significant impact on convergence of sequence of probability distributions.

Appendix A

Designs of the Variational Distribution q_ϕ

The choice of the approximate posterior is one of the core problems in variational inference, which generally require to employ simple families of posterior approximation in order to allow for efficient inference and in the meanwhile richness of the expressiveness from this approximation should also be guaranteed. We start by considering some classical method that has been widely used before, the *mean-field approximation.*

A.1 Mean-field Approximation

This method assumes the latent variables to be mutually independent and the distribution q_ϕ factorizes as follows

$$q_\phi(x_0, ..., x_k) = \prod_{i=0}^{k} q_{\phi_i}(x_i)$$

then the optimization over the parameters ϕ_i of marginal distribution is simplified. This factorized version of approximate posterior in variational inference corresponds to the approximation framework developed in physics called *mean-field theory* [25].

Though this assumption leads to a few nice properties, it also inccures several problems by its nature. For instance, it cannot capture the correlations between the variables, and also fails to approximate a multi-modal true posteriors. Besides, another drawback is that the variance is mostly underestimated [34].

There are efforts trying to improve this approach. One is to parametrize the correlations between the latent variables. But this will substantially increase the complexity when the number of latent variables increases largely.

Another improvement is to introduce mixture models [14]

$$q_{mix} = \sum_i \alpha_i q_{mf}^{(i)}$$

where each component distribution $q_{mf}^{(i)}$ is a factorized distribution, and α_n the mixing proportions. This could be potentially a powerful alternative, however again it introduces a potential problem of limiting the scability of variational inference since it requires to evaluate the log-likelihood and its gradients for each component distribution per parameter update, which is computationally expensive.

For these reasons, we want to look for alternatives such that the prescribed family for q_ϕ is expressive enough with scalable complexity. The following method of normalizing flow has surprising simplicity and overcomes the above limitations.

A.2 Normalizing Flows

The flow method in [30] is introduced by Rezende in 2015, whose basic idea is to construct complex distributions by transforming a initial probability distribution through a series invertible mappings. By repeatedly applying the rule of change of variables, the initial distribution 'flows' through this sequence of mappings. Then at the end of this sequence we can obtain a valid distribution. For this reason it is referred to as *normalizing flow*. It provides a modified variational upper bound with additional terms that only add linear complexity.

For invertible, smooth mapping $f : \mathbb{R}^d \to \mathbb{R}^d$, random variable x with distribution $q(x)$, by the chage of variable theorem, the resulting variable $x' = f(x)$ has a distribution

$$q(x') = q(x) \left| \det \frac{\partial f^{-1}}{\partial x'} \right| = q(x) \left| \det \frac{\partial f}{\partial x} \right|^{-1}. \tag{A.1}$$

We want to compose a sequence of simple mappings by applying (A.1) successively and then construct an arbitrarily complex distribution.

Definition A.2.1 (*normalizing flow*)**.** For invertible, smooth mappings $f_1, ..., f_K : \mathbb{R}^d \to \mathbb{R}^d$, density $q_0(x_0)$ of random variable x_0 , then the density q_K obtained by successively transforming the initial random variable with a chain of K transformations is

$$x_K = f_K \circ ... \circ f_1(x_0)$$

$$\ln q_K(x_K) = \ln q_0(x_0) - \sum_{i=1}^{K} \ln \left| \det \frac{\partial f_i}{\partial x_{i-1}} \right| \tag{A.2}$$

the path traversed by the random variables $x_i = f_i(x_{i-1})$ with initial distribution $q_0(x_0)$ is called the *flow* and the path formed by the successive distributions q_i is a *normalizing flow.*

Remark. *A interesting property of this flow, refer to as the law of the unconscious statistician (LOTUS), is that the expectation w.r.t. q_K can be computed without explicitly knowing q_K. That is, for any function g,*

$$\mathbf{E}_{q_K}[g(x_K)] = \mathbf{E}_{q_0}[g(f_K \circ ... \circ f_1(x_0))] \tag{A.3}$$

The effect of flows can be thought of as a sequence of expansions and contractions on the initial density and by specifying the class of invertible transformations we can define different type of flows. Two often used flows are the *planar flow* and the *radial flow*, which have efficient mechanism for computing.

Definition A.2.2 (*planar flow*)**.** Consider the family of transformations of the form:

$$f(x) = x + \mathbf{u}h(\mathbf{w}^\intercal x + b) \tag{A.4}$$

where $\mathbf{w} \in \mathbb{R}^d$, $\mathbf{u} \in \mathbb{R}^d$, $b \in \mathbb{R}$ are parameters and $h : \mathbb{R} \to \mathbb{R}$ is a smooth, element-wise nonlinear function. A flow implemented by transformations of this form is called a *planar flow.*

Remark. *The flow above defined by the transformation modifies the initial density by applying a series of contraction and expansion perpendicular to the hyperplane $w^\intercal x + b = 0$. Then the log-det-Jacobian in (A.2) that corresponds to this flow would be*

$$\left|\det \frac{\partial f}{\partial x}\right| = |\det(\boldsymbol{I} + \boldsymbol{u}\psi(x)^\intercal)| = |1 + \boldsymbol{u}^\intercal \psi(x)|,$$

with $\psi(x) = h'(\boldsymbol{w}^\intercal x + b)\boldsymbol{w}$. Therefore, from (A.2) we can conclude that the density obtained by this flow is

$$\ln q_K(x_K) = \ln q_0(x_0) - \sum_{i=1}^{K} \ln |1 + \boldsymbol{u}_i^\intercal \psi_i(x_{i-1})|. \tag{A.5}$$

Furthermore, by (A.3) and (A.5), the flow-based free energy (1.3) in this case would be

$$\begin{aligned}
\mathbf{F}_{\phi,\theta}(y) :&= D(q_\phi(x|y)||p_\theta(x)) - \mathbf{E}_{q_\phi(x|y)}[\log p_\theta(y|x)] \\
&= \mathbf{E}_{q_\phi(x|y)}[\log q_\phi(x|y) - \log p_\theta(x, y)] \\
&= \mathbf{E}_{q_0(x_0)}[\ln q_K(x_K) - \log p_\theta(x_K, y)] \\
&= \mathbf{E}_{q_0(x_0)}[\ln q_0(x_0)] - \mathbf{E}_{q_0(x_0)}[\log p_\theta(x_K, y)] \\
&\quad - \mathbf{E}_{q_0(x_0)}\left[\sum_{i=1}^{K} \ln |1 + \boldsymbol{u}_i^\intercal \psi_i(x_{i-1})|\right].
\end{aligned} \tag{A.6}$$

The functions of the form (A.4) are not always invertible. It depends on the non-linearity and parameters chosen. For example, if we use $h(x) = \tanh(x)$*, then a sufficient condition that makes it invertible is that* $\boldsymbol{w}^\intercal \boldsymbol{u} \geq -1$.

Similarly, we can define the *radial flow* considering the following transformation

$$f(x) = x + \frac{\beta}{\alpha + r}(x - x_0) \tag{A.7}$$

where $r = |x - x_0|$, $\alpha \in \mathbb{R}^+$, $\beta \in \mathbb{R}$. This family applies radial contractions and expansions around the reference point x_0. The corresponding expression of free energy for this case can be derived as (A.6).

Here we give another example of flow structures, the *volume-preserving flows*, which was actually developed before the the general concept of normalizing flows, serving as a prototype of the flow method. It has special design such that its Jacobian determinant is equal to one while still allowing for rich posterior distributions. The *Non-linear Independent Components Estimation (NICE)* developed by Dinh et al. [8] is an instance of volume-preserving flows. Considering the transformation of the form

$$f(x) = (x_A, x_B + h_\lambda(x_A)) \tag{A.8}$$

where $x = (x_A, x_B)$ is an arbitrary partitioning of the vector x (for example, $x = [x_A = x_{1:n}, x_B = x_{n+1:d}]$) and h_λ is a neural net with parameter λ. Its inverse is easy to compute, given by $g(x') = (x'_A, x'_B - h_\lambda(x'_A))$. This design results in a Jacobian that has a zero upper triangular part and all the diagonal elements are 1's and hence the determinant to be 1. Thus the volume of elementary unit in the sample space is preserved after the transformation. This immensely simplifies the expression of the change of the log-density

$$\ln q_K(f_K \circ f_{K-1} \circ ... \circ f_1(x_0)) = \ln q_0(x_0)$$

However, in order to build transformations capable of mixing all components of the initial random variable x_0, such flows must alternate between different partitionings of x_k. This requires us to introduce extra mechanisms for mixing the components of x, which cannot be done automatically and obviously it would lead to a drawback in model learning efficiency. In general, the convergence rate of this flow will be slower than the planar and radial flows, though it could as well achieve similar results in the sense of asymptotic convergence.

Alternative option of transformations could be used and different variants of flows were developed these recent years. For instance, In [7], it also takes a further step to consider the continuous-time limit of normalizing flows, and it turns out the continuous normalizing flow simplifies the computation and outperform the discrete normalizing flow in convergence rate of the loss function. For latest development on normalizing flows, refer to [27], which provides a comprehensive review.

Appendix B

Auxiliary Material

Here we gather the preliminaries for the proof of Theorem 2.4.1.

B.1 Results from the Theory of Empirical Process

First we recall the concepts of *Orlicz norm* and *Orlicz space* [38].

Definition B.1.1. The Orlicz exponential norm of order 2 for a real-valued random variable U on the probability space $(\Omega.\Sigma, \mathbf{P})$ is defined as

$$\|U\|_{\psi_2} := \inf\{t > 0 : \mathbf{E}\psi_2(|U|/t) \leq 1\}.$$

where $\psi_2(x) = e^{x^2} - 1$. The corresponding Orlicz space $L_{\psi_2} = L_{\psi_2}(\Omega, \Sigma, \mathbf{P})$ consists of all random variables U with finite Orlicz norm

$$L_{\psi_2} := \{U : \|U\|_{\psi_2} < \infty\}$$

Note that the random variables in L_{ψ_2} are sub-gaussian and the ψ_2 Orlicz norm dominates the L^2 norm, i.e. $\|U\|_{\psi_2} \geq \|U\|_{L^2}$.

The following lemma allows us to use the Orlicz norm to estimate the tail of a distribution.

Lemma B.1.1 (*tail bound for random variable in L_{ψ_2}*)**.** *For $U \in L_{\psi_2}$, $t > 0$,*

$$\mathbf{P}(|U| \geq t\|U\|_{\psi_2}) \leq \frac{1}{e^{t^2} - 1}. \tag{B.1}$$

Proof. By Markov's inequality

$$\mathbf{P}(|U|/\|U\|_{\psi_2} \geq t) \leq \frac{\mathbf{E}[\psi_2(|U|/\|U\|_{\psi_2})]}{\psi_2(t)} \leq \frac{1}{e^{t^2} - 1}.$$

■

Conversely, when we have the tail bound, it can be used to estimate the Orlicz norm.

Lemma B.1.2 (*estimate for $\|\cdot\|_{\psi_2}$*)**.** *For $U \in L_{\psi_2}$ with*

$$\mathbf{P}(|U| \geq t) \leq K\exp(-Ct^2),$$

for $t > 0$, constants K, C. If $C > t^{-2}$, then

$$\|U\|_{\psi_2} \leq \left(\frac{K+1}{C}\right)^{1/2}. \tag{B.2}$$

Proof. Let $s = t^{-2}$, by Fubini's theorem

$$\begin{aligned}
\mathbf{E}\psi_2(|U|/t) &= \mathbf{E}\left[e^{sU^2} - 1\right] = \mathbf{E}\int_0^{U^2} se^{sr}dr \\
&= \mathbf{E}\int_0^{\infty} \mathbf{1}_{\{U^2 \geq r\}}\, se^{sr}dr \\
&\overset{Fubini}{=} \int_0^{\infty} \mathbf{P}(|U| \geq r^{1/2})se^{sr}dr \\
&\leq \int_0^{\infty} Kse^{(s-C)r}dr \\
&= Ks/(C-s)
\end{aligned}$$

This implies

$$\begin{aligned}
&\mathbf{E}\psi_2(|U|/t) \leq 1 \\
\Leftrightarrow\ & t = s^{-1/2} \geq ((K+1)/C)^{1/2} \\
\Rightarrow\ & \|U\|_{\psi_2} \leq ((K+1)/C)^{1/2}
\end{aligned}$$

■

Lemma B.1.3 (*Gaussian concentration inequality*)**.** *Let $Z \sim \gamma_d$, $f : \mathbb{R}^d \to \mathbb{R}$ an L-Lipschitz function. Then for $t > 0$,*

$$\mathbf{P}(f(X) - \mathbf{E}f(X) \geq t) \leq e^{-t^2/2L^2}. \tag{B.3}$$

Proof. Theorem 5.6 in [5]. ■

Lemma B.1.4. *Let $U = \|Z\|$, where $Z \sim \gamma_d$. Then*

$$\|U\|_{\psi_2} \leq \sqrt{d} + \sqrt{6} \tag{B.4}$$

Proof. Using the triangle inequality, we have

$$\|U\|_{\psi_2} \leq \mathbf{E}U + \|U - \mathbf{E}U\|_{\psi_2},$$

where $\mathbf{E}U \leq \|U\|_{L_2} = \sqrt{d}$ by Jensen's inequality. Moreover, the function $f(\cdot) = \|\cdot\|$, $\mathbb{R}^d \to \mathbb{R}^+$ is 1-Lipschitz. Apply (B.3), we obtain

$$\mathbf{P}(|\|X\| - \mathbf{E}\|X\|| \geq t) \leq 2e^{-t^2/2}.$$

which implies $\||\|X\| - \mathbf{E}\|X\||\|_{\psi_2} \leq \sqrt{6}$ by (B.2). Therefore we have

$$\|U\|_{\psi_2} \leq \sqrt{d} + \sqrt{6}.$$

■

Maximal inequalities allow us to bound probabilities involving suprema of random variables, for both the law of large numbers and the central limit theorem.

Lemma B.1.5 (*maximal inequality*)**.** *Let $U_1, ..., U_N$, $N \geq 2$, be a collection of random variables in L_{ψ_2}. Then we have the following maximal inequality:*

$$\left\|\max_{j\leq N} |U_j|\right\|_{\psi_2} \leq 4\sqrt{\log N} \max_{j\leq N} \|U_j\|_{\psi_2}. \tag{B.5}$$

Proof. Without loss of generality, we assume $\psi_2(x)\psi_2(y) \leq \psi_2(cxy)$, for $x, y \geq 1$ and some constant c. Thus, for any $2x \geq y \geq 1$

$$\psi_2(x/y) \leq \psi_2(cx)/\psi_2(y)$$

Now for constant $C > 0$, and $y \geq 1$

$$\begin{aligned}
&\max_j \psi_2\left(\frac{|U_j|}{Cy}\right) \\
\leq &\max_j \left(\frac{\psi_2(c|U_j|/C)}{\psi_2(y)} \cdot \mathbf{1}_{\{\frac{|U_j|}{Cy} \geq \frac{1}{2}\}} + \psi_2\left(\frac{|U_j|}{Cy}\right) \cdot \mathbf{1}_{\{\frac{|U_j|}{Cy} < \frac{1}{2}\}}\right) \\
\leq &\max_j \frac{\psi_2(c|U_j|/C)}{\psi_2(y)} + \psi_2\left(\frac{1}{2}\right) \\
\leq &\sum_j \frac{\psi_2(c|U_j|/C)}{\psi_2(y)} + \psi_2\left(\frac{1}{2}\right)
\end{aligned}$$

Let $C = c \max_{\{j\leq N\}} \|U_j\|_{\psi_2}$,

$$\begin{aligned}
&\mathbf{E}\psi_2\left(\frac{\max_j |U_j|}{Cy}\right) \leq \mathbf{E} \max_j \psi_2\left(\frac{|U_j|}{Cy}\right) \\
\leq &\sum_j \mathbf{E}\left[\frac{\psi_2(|U_j|/\max_j \|U_j\|_{\psi_2})}{\psi_2(y)}\right] + \psi_2\left(\frac{1}{2}\right) \\
\leq &\frac{N}{\psi_2(y)} + \psi_2\left(\frac{1}{2}\right)
\end{aligned}$$

Since $\psi_2(1/2) \leq 1/2$, this implies that $\mathbf{E}\psi_2\left(\max_j |U_j|/Cy\right) \leq 1$ is equivalent to

$$Cy \geq \psi_2^{-1}(2N) \cdot c \max_j \|U_j\|_{\psi_2}$$

Therefore, let $c = 2$ we have

$$\left\| \max_j |U_j| \right\|_{\psi_2} \leq 4\sqrt{\log N} \max_j \|U_j\|_{\psi_2}.$$

■

Besides, we also want to use some results from the *entropy methods.*

Definition B.1.2 (*envelope*)**.** Let $\mathcal{G}$ be a class of real-valued functions on some measurable space Z. A positive function: $F : \mathsf{Z} \to \mathbb{R}_+$ is called an *envelope* of $\mathcal{G}$ if $|g(z) \leq F(z)|$ for every $g \in \mathcal{G}$ and $z \in \mathsf{Z}$.

Definition B.1.3 (*covering number, Koltchinskii-Pollard ϵ-entropy*)**.** Let $Z_1, ..., Z_N$ be i.i.d. random elements of Z with probability law P and denote by P_N the corresponding empirical distribution, i.e. $P_N(A) = \frac{1}{N}\sum_{n\leq N} \mathbf{1}_{\{Z_n \in A\}}$ for all measurable set $A \subset \mathsf{Z}$. Denote the expectations by

$$Pg := \mathbf{E}_P[g(Z)]\,, \;\; P_N g := \mathbf{E}_{P_N}[g(Z)] = \frac{1}{N}\sum_{n\leq N} g(Z_n)$$

then we define the quantity

$$\|P_N - P\|_{\mathcal{G}} := \sup_{g\in\mathcal{G}} |P_N g - Pg| \tag{B.6}$$

which is by definition a random variable on Z. The $L(Q)$ *covering number of* $\mathcal{G}$ with respect to a probability measure Q in Z is defined by

$$N(\mathcal{G}, L^2(Q), \epsilon) := \min\{K : \exists f_1, ..., f_K \in L^2(Q) \\ \text{such that } \sup_{g\in\mathcal{G}} \min_{k\leq K} \|g - f_k\|_{L^2(Q)} \leq \epsilon\}$$

And the *Koltchinskii-Pollard ϵ-entropy of* $\mathcal{G}$ is given by

$$H(\mathcal{G}, F, \epsilon) := \sup_Q \sqrt{\log 2N(\mathcal{G}, L^2(Q), \epsilon\|F\|_{L^2(Q)})}$$

with the supremum over the measures Q that are supprted on finiely many points of Z.

We are interested in the properties of (B.6), and it has the following bound on expectation and a concentration inequality.

Lemma B.1.6 (*the bound for* $\mathbf{E}\|P_N - P\|_{\mathcal{G}}$). *Let* $\mathcal{G}$ *be a class of functions containing* 0, *such that*

$$J(\mathcal{G}, F) := \int_0^\infty H(\mathcal{G}, F, \epsilon) d\epsilon < \infty.$$

Let $Z_1, ..., Z_N$ *be i.i.d. copies of a random variable* Z *on* Z *with probability law* P, *such that* $F \in L^2(P)$. *Then*

$$\mathbf{E}\|P_N - P\|_{\mathcal{G}} \le \frac{8\sqrt{2}J(\mathcal{G}, F)\|F\|_{L^2(P)}}{\sqrt{N}}$$

Proof. Theorem 3.5.4 in [12]. ■

Lemma B.1.7 (*Adamczak's concentration inequality*). *Let* $\mathcal{G}$ *be a class of real-valued functions on* Z *with envelope* F. *Then there exists a constant* $C > 0$, *such that for any* $\gamma > 0$,

$$\mathbf{P}\left\{\|P_N - P\|_{\mathcal{G}} \le C\left[\mathbf{E}\|P_N - P\|_{\mathcal{G}} + \sigma_P(\mathcal{G})\sqrt{\frac{\gamma}{N}} + \left\|\max_{n \le N} F(Z_n)\right\|_{\psi_2} \frac{\sqrt{\gamma}}{N}\right]\right\} \le e^{-\gamma}$$

with $\sigma_P^2(\mathcal{G}) := \sup_{g \in \mathcal{G}}(Pg^2 - (Pg)^2)$.

Proof. Section 2.3 in [17], and theorem 4 in [1]. ■

With all these preliminaries, we show the following result, which will be used in the proof of Theorem 2.4.1.

Lemma B.1.8. *Let* $g : \mathbb{R}^d \to \mathbb{R}$ *be L-Lipschitz with respect to the Euclidean norm. Let* $Z_1, ..., Z_N$ *be i.i.d. copies of a d-dimensional random vector* Z, *such that* $U := \|Z\|$ *has finite* ψ_2 *norm. Then there exists a constant* $C > 0$, *such that for any* $\gamma > 0$,

$$\begin{aligned}\sup_{x \in \mathsf{B}^d(R)} \sup_{t \in [0,1]} &\left|\frac{1}{N}\sum_{n=1}^N g(x + \sqrt{t}Z_i) - \mathbf{E}[g(x + \sqrt{t}Z)]\right| \\ &\le C\left[\frac{16L\sqrt{6\pi Rd}(R \vee 1 + \|U\|_{\psi_2})}{\sqrt{N}} + 5L((R \vee 1) + \|U\|_{\psi_2})\sqrt{\frac{\gamma}{N}}\right] \quad \text{(B.7)}\end{aligned}$$

with probability at least $1 - e^{-\gamma}$.

Proof. Let P denote the probability law of Z. Let $g_{x,t}(z) := g(x + \sqrt{t}z)$ for $x \in \mathbb{R}^d$ and $t \ge 0$, and $\overline{g}_{x,t} := g_{x,t} - g_{0,0}$, where $g_{0,0} = g(0)$. Then we introduce a function class $\mathcal{G} := \{\overline{g}_{x,t} : x \in \mathsf{B}^d(R), t \in [0,1]\}$ and the empirical process supremum

$$\|P_N - P\|_{\mathcal{G}} = \sup_{x \in \mathsf{B}^d(R)} \sup_{t \in [0,1]} |P_N\overline{g}_{x,t} - P\overline{g}_{x,t}|.$$

Since $\overline{g}$ is L-Lipschitz, for $z \in \mathbb{R}^d, x \in \mathsf{B}^d(R), t \in [0,1]$, we have

$$|\overline{g}_{x,t}(z)| \leq |g(x+\sqrt{t}z) - g(0)| \leq L\|x+\sqrt{t}z\| \leq L((R \vee 1) + \|z\|).$$

Let $F(z) := L((R \vee 1) + \|z\|)$. Since $\|\cdot\|_{L^2} \leq \|\cdot\|_{\psi_2}$, the fuction $F \in L^2(P)$. We see that F is a square-integrable envelope of $\mathcal{G}$.

Moreover, for any probability measure Q supported on finitely many points in $\mathbb{R}^d$ and for all $x, x' \in \mathsf{B}^d(R)$ and $t, t' \in [0,2]$,

$$\|\overline{g}_{x,t} - \overline{g}_{x',t'}\|_{L^2(Q)} \leq \|F\|_{L^2(Q)} \cdot (\|x-x'\| + |t-t'|^{1/2}).$$

Then we can estimate the $L^2(Q)$ covering number of (G) by

$$N(\mathcal{G}, L^2(Q), \epsilon\|F\|_{L^2(Q)}) \leq N(\mathsf{B}^d(R), \|\cdot\|, \epsilon/2) \cdot N([0,1], |\cdot|, \epsilon^2/4).$$

Using standard volumetric estimates on the covering number, we obtain the following bound on the Koltchinskii-Pollard entropy of $\mathcal{G}$:

$$H(\mathcal{G}, F, \epsilon) \leq \left(4d \log \frac{2\sqrt{3R}}{\epsilon}\right) \vee 0$$

therefore

$$J(\mathcal{G}, F) = \int_0^\infty H(\mathcal{G}, F, \epsilon) d\epsilon \leq 2\sqrt{3\pi Rd}.$$

Then by Lemma B.1.6, we have

$$\begin{aligned}
\mathbf{E}\|P_N - P\|_{\mathcal{G}} &\leq \frac{8\sqrt{2}J(\mathcal{G},F)\|F\|_{L^2(P)}}{\sqrt{N}} \\
&= \frac{16L\sqrt{6\pi Rd}((R \vee 1) + \|U\|_{L^2(P)}}{\sqrt{N}} \\
&= \frac{16L\sqrt{6\pi Rd}((R \vee 1) + \|U\|_{\psi_2}}{\sqrt{N}}. \qquad \text{(B.8)}
\end{aligned}$$

Also, we have

$$\begin{aligned}
\sigma_P(\mathcal{G}) &\leq \|F\|_{L^2(P)} \leq \|F\|_{\psi_2} = \|L((R \vee 1) + U)\|_{\psi_2} \\
&\leq L((R \vee 1) + \|U\|_{\psi_2}) \qquad \text{(B.9)}
\end{aligned}$$

and by the maximal inequality (B.5)

$$\begin{aligned}
\left\|\max_{j \leq N} F(Z_j)\right\|_{\psi_2} &= L\left\|(R \vee 1) + \max_{j \leq N} U_j\right\|_{\psi_2} \\
&\leq L(R \vee 1) + 4L\sqrt{\log N}\|U\|_{\psi_2}. \qquad \text{(B.10)}
\end{aligned}$$

Finally, we combine (B.8), (B.9), (B.10) with Lemma B.1.7 and obtain (B.7).

■

B.2 Approximate Elementary Operations by Neural Nets

Lemma B.2.1 (*approximating the multiplications*)**.** *With Assumption 2.4.2 on activation functions, for any $M > 0$ and any $\delta > 0$, there exists a $2-$layer neural net $g : \mathbb{R}^2 \to \mathbb{R}$ of size $m \le 4c_\sigma M^2/\delta + 1$, such that*

$$\sup_{x,y\in[-M,M]} |g(x,y) - xy| \le \delta.$$

Proof. Consider the function

$$x \mapsto x^2 \wedge M^2$$

which is $2M$-Lipschitz and constant outside $[-M, M]$. With Assumption 2.4.2, it guarantees the existence of a 2-layer neural net $g_0 : \mathbb{R} \to \mathbb{R}$, with the size (number of nodes) $m \le 2c_\sigma \frac{M^2}{\delta}$ satisfying $|g_0(x) - x^2| \le 2\delta$ for all $x \in [-M, M]$. Then by the polarization identity $4xy = (x + y)^2 + (x - y)^2$, we have the desired approximation $g : \mathbb{R}^2 \to \mathbb{R}$ given by

$$g(x,y) = \frac{1}{4}(g_0(x+y) + g_0(x-y))$$

which is also a 2-layer feed forward neural net and of the size $m \le 4c_\sigma \frac{M^2}{\delta} + 1$, satisfying

$$|g(x,y) - xy| \le \delta$$

for all $x, y \in [-M, M]$.

■

Lemma B.2.2 (*approximating the reciprocals*)**.** *With Assumption 2.4.2 on activation functions, for any $0 < a \le b < \infty$ and any $\delta > 0$, there exsits a 2-layer neural net $q : \mathbb{R} \to \mathbb{R}$ of size $m \le c_\sigma \frac{b}{a^2\delta} + 1$, such that*

$$\sup_{x\in[a,b]} \left| q(x) - \frac{1}{x} \right| \le \delta$$

Proof. Consider the function

$$x \mapsto \frac{1}{a}\mathbf{1}_{\{x<a\}} + \frac{1}{x}\mathbf{1}_{\{a\le x\le b\}} + \frac{1}{b}\mathbf{1}_{\{x>b\}}$$

which is $(1/a^2)$-Lipschitz and constant outside $[a, b]$. The existence of the desired approximation q then follows immediately from the Assumption 2.4.2. ■

Lemma B.2.3 (*cheap gradient principle*)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be implementable by a neural net of size m (number of nodes) and depth l (number of layers), with differentiable activation functions $\sigma : \mathbb{R} \to \mathbb{R}$. Then each coordinate of the gradient ∇f can be computed by a neural net that has size $\mathcal{O}(m + l)$, with activation function belonging to the set $\{\sigma, \sigma'\}$ in each neuron.*

Proof. Section 3.3 in [11]. ■

Bibliography

[1] R. Adamczak. *A Tail Inequality for Suprema of Unbounded Empirical Processes with Applications to Markov Chains.* Electronic Journal of Probability, 13:1000-1034, 2008.

[2] D. J. Bartholomew, M. Knott, I. Moustaki. *Latent Variable Models and Factor Analysis: A Unified Approach.* John Wiley & Sons, ISBN: 978-0-470-97192-5, 2011.

[3] R. Bellman. *Dynamic Programming.* Dover, ISBN: 0-486-42809-5, 1957.

[4] N. J. Beaudry, R. Rennner. *An Intuitive Proof of Data Processing Inequality.* Quantum Information and Computation, 2012.

[5] S. Boucheron. *Concentration Inequalities: A Nonasymptotic Theory of Independence.* Oxford University Press, 2013.

[6] G. Cybenko. *Approximation by Superpositions of a Sigmoidal Function.* Mathematics of Control, Signals and Systems 2, 303–314, 1989.

[7] R. Chen, Y. Rubanova, J. Bettencourt, D. Duvenaud. *Neural Ordinary Differential Equations.* Neural Information Processing Systems, 2018.

[8] L. Dinh, D. Krueger, Y. Bengio. *NICE: Non-linear Independent Components Estimation.* ICLR Workshop, 2015.

[9] K. Fukushima. *Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position.* Biological Cybernetics 36: 193-202, 1980.

[10] E. Gobet, R. Munos. *Sensitivity Analysis Using Itô–Malliavin Calculus and Martingales, and Application to Stochastic Optimal Control.* SIAM Journal on Control and Optimization 43(5):1676-1713, 2005.

[11] A. Griewank, A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation.* SIAM, 2nd edition, 2008.

[12] E. Giné, R. Nickl. *Mathematical Foundations of Infinite-Dimensional Statistical Models.* Cambridge University Press, 2016.

[13] K. Hornik, M. Stinchcombe, H. White. *Multilayer Feedforward Networks are Universal Approximators.* Neural Networks, Vol.2, pp.359-366, 1989.

[14] T. S. Jaakkola, M. I. Jordan. *Improving the Mean Field Approximation via the Use of Mixture Distributions.* Learning in graphical models, pp. 163–173. 1998.

[15] D. P. Kingma, M. Welling. *Auto-Encoding Variational Bayes.* International Conference on Learning Representations, 2014.

[16] O. Kallenberg. *Foundations of Modern Probability.* Springer, 2002.

[17] V. Koltchinskii. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems.* Springer, 2011.

[18] H. Kunita. *Stochastic Flows and Jump-Diffusions.* Springer, 2019.

[19] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner. *GradientBased Learning Applied to Document Recognition.* Proceedings of the IEEE 86, No.11: 2278-2324, 1998.

[20] H. Lappalainen, A. Honkela. *Bayesian Non-Linear Independent Component Analysis by Multi-Layer Perceptrons.* Springer, London, ISBN: 978-1-85233-263-1, 2000.

[21] J. Lehec. *Representation formula for the entropy and functional inequalities.* Annales de l'Institut Henri Poincaré, Probabilités et Statistiques, Volume 49, Number 3, 885-899, 2013.

[22] X. Li, T. K. Leonard, R. Chen, D. Duvenaud. *Scalable Gradients and Variational Inference for Stochastic Differential Equation.* Proceedings of The 2nd Symposium on Advances in Approximate Bayesian Inference, PMLR 118:1-28, 2020.

[23] W. S. McCulloch, W. Pitts. *A Logical Calculus of the Ideas Immanent in Nervous Activity.* The Bulletin of Mathematical Biology 5, no.4: 115-113, 1943.

[24] L. S. Pontryagin, E. Mishchenko, V. Boltyanskii, R. Gamkrelidze. *The Mathematical Theory of Optimal Processes.* John Wiley & Sons, 1962.

[25] G. Parisi. *Statistical Field Theory.* Addison-Wesley: Redwood City, 1988.

[26] P. E. Protter. *Stochastic Integration and Differential Equations.* Springer, 2005.

[27] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, B. Lakshminarayanan. *Normalizing Flows for Probabilistic Modeling and Inference.* arXiv preprint arXiv:1912.02762, 2019.

[28] D. Rumelhart, G. Hinton, R. Williams. *Learning Representations by Back-propagating Errors.* Nature 323, 533–536, 1986.

[29] D. J. Rezende, S. Mohamed, D. Wierstra. *Stochastic Backpropagation and Approximate Inference in Deep Generative Models.* Proceedings of the 31st International Conference on Machine Learning, PMLR 32(2):1278-1286, 2014.

[30] D. J. Rezende, S. Mohamed. *Variational Inference with Normalizing Flows.* Proceedings of the 32nd International Conference on Machine Learning, PMLR 37:1530-1538, 2015.

[31] F. Rosenblatt. *The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain.* Psychological Review, Vol.65, No.6, 1958.

[32] D. E. Rumelhart, G. E. Hinton, R. J. Williams. *Learning Internal Representations by Error Propagation.* Defense Technical Information Center technical report, 1985.

[33] E. Schrödinger. *Über die Umkehrung der Naturgesetze.* Sitzung ber Preuss. Akad. Wissen., Berlin Phys. Math., 144, 1931.

[34] R. E. Turner, M. Sahani1. *Two Problems with Variational Expectation Maximisation for Time-series Models.* Cambridge University Press, 2011.

[35] B. Tzen, M. Raginsky. *Theoretical Guarantees for Sampling and Inference in Generative Models with Latent Diffusions.* Proceeings of the Conference on Learning Theory, 2019.

[36] B. Tzen, M. Raginsky. *Neural Stochastic Differential Equations: Deep Latent Gaussian Models in the Diffusion Limit.* arXiv preprint arXiv:1905.09883, 2019.

[37] D. W. Stroock. *An Introduction to Partial Differential Equations for Probabilists.* Cambridge University Press, 2008.

[38] R. Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science.* Cambridge University Press, 2018.