Machine Learning Approaches for The Empirical Schrödinger Bridge Problem

Francisco A. Vargas
Girton College

UNIVERSITY OF CAMBRIDGE

A dissertation submitted to the University of Cambridge in partial fulfilment of the requirements for the degree of Master of Philosophy in Advanced Computer Science

University of Cambridge
Department of Computer Science and Technology
William Gates Building
15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom

Email: fav25@cam.ac.uk

November 24, 2020

Declaration

I Francisco A. Vargas of Girton College, being a candidate for the M.Phil in Advanced Computer Science, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

Total word count: 14950

Signed: Francisco Vargas

Date: 24.06.2020

Acknowledgements

I would like to thank my supervisor, Professor Neil Lawrence, for his guidance throughout this project and for providing extensive feedback for this manuscript. I would like to thank my second supervisor, Professor Austen Lamacraft, for his in-depth tutoring on stochastic differential equations and his assistance in some of the mathematical aspects of this work.

To my parents Julio and Ligia and to my brother Arturo for their eternal support in my education.

I would like to thank Ramona Comănescu, for her support throughout this whole project and the challenging times during lockdown. Finally, I would like to thank Neil Lawrence, Thomas Burger, Ramona Comănescu, Paul Scherer and Kamen Brestnichki for helping with the proofreading of this manuscript.

Machine Learning Approaches for The Empirical Schrödinger Bridge Problem

Abstract

The Schrödinger bridge problem is concerned with finding the most likely stochastic evolution between two probability distributions given a prior/reference stochastic evolution. This problem was posed by Schrödinger (1931, 1932) and solved to a large extent. Problems of this kind, whilst not popular in the machine learning community, have direct applications such as domain adaptation, hypothesis testing, semantic similarity, and others.

Thus, the focus of this thesis is to carry out a preliminary study on computational approaches for estimating the Schrödinger bridge between two distributions, when these distributions are available (or can be available) through samples, as most problems in machine learning are.

Due to the mathematical nature of the problem, this manuscript is also concerned with restating and re-deriving theorems and results that seem to be considered communal knowledge within the mathematical community or hidden in type-written textbooks behind paywalls. Part of the aim of this thesis is to make the mathematical machinery behind these approaches more accessible to a broader audience.

Word count: 14950.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The motion of small particles in a fluid and the value of financial instruments as a function of time have an interesting intersection. These two processes have a large number of underlying factors (e.g. physical forces) that impact their trajectories and constantly change as a function of time. This constant stream of varying forces causes the trajectory of these systems to seem random and makes it infeasible to model with non-probabilistic approaches.

What we call Brownian motion today is a random process. It is one of the simplest physical models used to describe the motion of small particles in a fluid. The earliest mathematical formulations of this process date back to Einstein (1905), where the diffusion-based formulation of Brownian motion was used to describe the motion of pollen particles in a fluid. Many day-to-day processes have an element of Brownian motion within them, thus Brownian motion has become a fundamental mathematical tool when describing the motion of stochastic temporal processes.

A class of Brownian-motion-driven processes that is of particular interest to many areas of applied science is the drift-augmented Brownian motion. In simple terms, this family of processes can be split into two components: a Brownian-motion-driven term and a drift which provides a directional drive to the system, much like a driving force. A particular example process that is

popular in the machine learning and computational neuroscience communities is the Ornstein–Uhlenbeck (OU) process (Doob, 1942). The OU-process is used to model particles undergoing Brownian motion that are subject to friction represented by the drift term. An interesting remark is that the discrete-time analogue of the OU-process is the auto-regressive process of order 1 – AR(1).

This project is concerned with a specific event regarding Brownian-motion-driven processes proposed by Schrödinger (Schrödinger, 1931, 1932). The event postulates a group of particles undergoing Brownian motion in $\mathbb{R}^d$, where we observe their position distributions $\pi_0(\boldsymbol{x})$ at time 0, and then $\pi_1(\boldsymbol{y})$ at time 1. Then, consider the case where $\pi_1(\boldsymbol{y})$ differs significantly from the distribution predicted by Brownian motion, that is

$$\pi_1(\boldsymbol{y}) \neq \int p(\boldsymbol{x}, 0, \boldsymbol{y}, 1)\pi_0(\boldsymbol{x})d\boldsymbol{x},$$

where $p(\boldsymbol{x}, 0, \boldsymbol{y}, 1) = p(\boldsymbol{y}_1|\boldsymbol{x}_0)$ represents the transition density under the Brownian motion. The transition density is the probability of transitioning from $\boldsymbol{x}$ at time 0 to $\boldsymbol{y}$ at time 1. We can model this event as if the particles at time 0 were transported to time 1 in an unlikely manner (a rare event). Out of the many unlikely ways in which this event could have happened, the Schrödinger bridge tells us which one is the most likely. This question turns out to be equivalent to finding a drift-augmented Brownian motion that satisfies the observed distributions and is as close to Brownian motion as possible in an information-theoretic sense.

Rephrasing this in a more machine-learning setting, the Schrödinger bridge is concerned with finding the most likely stochastic process that evolves a distribution $\pi_0(\boldsymbol{x})$ to another distribution $\pi_1(\boldsymbol{y})$, and is in line with a pre-specified Brownian-motion prior.

From an application viewpoint, the Schrödinger bridge provides us with a theoretically-grounded mechanism for mapping between two distributions. When those distributions are only available through samples, we effectively have a classical, unsupervised domain-adaptation problem, as illustrated in

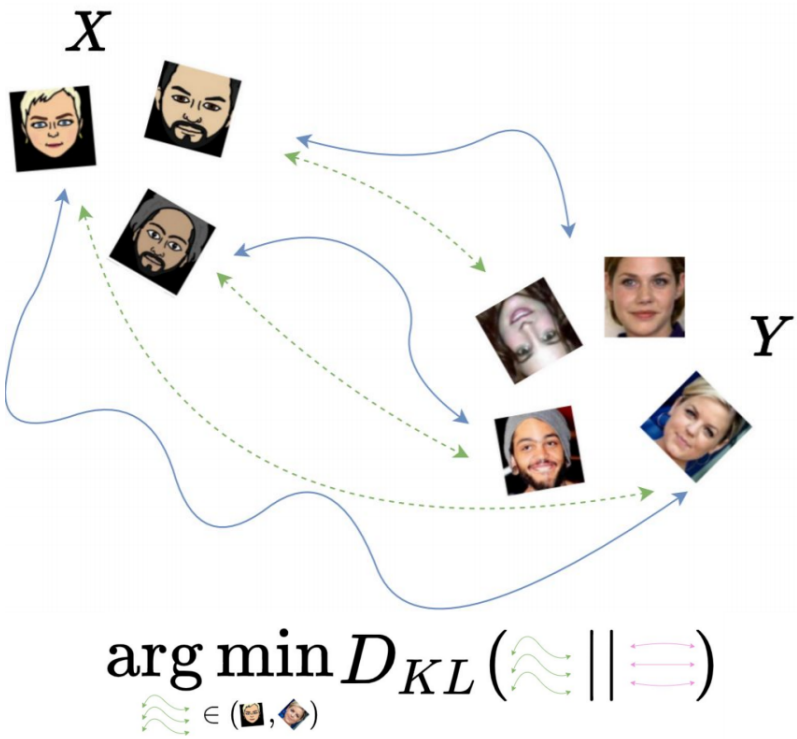

Figure 1.1: Intuitive Illustration of the Schrödinger Bridge.

Figure 1.1. Furthermore, the Schrödinger bridge also provides us with the probability of this stochastic evolution, thus allowing us to compare two datasets/distributions which can be useful for hypothesis testing (Gretton et al., 2012; Ramdas et al., 2017) and semantic similarity (Vargas et al., 2019).

In this project, we explore machine-learning-based approaches to estimating the Schrödinger bridge between two distributions that are available through samples.

Chapter 2

Mathematical Preliminaries

The goal of this chapter is to introduce mathematical notation, concepts and lemmas that we will use throughout this thesis. Whilst many of these lemmas are omitted or taken for granted in the stochastic process literature, we found that some of them were not directly accessible when searched for and not taught in graduate-level measure– and probability-theory courses (Salamon, 2019; Viaclovsky, 2003; Andres, 2019). Therefore, it will be useful to restate and rederive some of these results, in order to make the theory behind Schrödinger bridges more accessible to the computational sciences.

Due to limitations of the Riemann integral, we will be briefly covering a measure-theoretic introduction to probability in order to have the background required to introduce the Lebesgue–Stieltjes integral, which we will be using to express some of the expectations in this thesis. Furthermore, we will also be providing a sketch proof of standard results for decompositions of the Radon-Nikodym derivative which we will employ in decomposing the Kullback–Leibler (KL) divergence.

2.1 Probability Space Formalism

For many of the technical derivations in the methodology we will study, a basic notion of measure and integration is required, thus we will refresh con-

cepts such as σ-algebra, probability measures, measurable functions and the Lebesgue–Stieltjes integral, without going into unnecessary technical detail.

Definition 2.1.1. A probability space is defined as a 3-element tuple $(\Omega, \mathcal{F}, \mathbb{P})$, where:

- Ω is the sample space, i.e. the set of possible outcomes. For example, for a coin toss $\Omega = \{\text{Head}, \text{Tails}\}$.
- The σ-algebra $\mathcal{F} \subseteq 2^{\Omega}$ represents the set of events we may want to consider. Continuing the coin toss example, we may have $\Omega = \{\emptyset, \text{Head}, \text{Tails}, \{\text{Head}, \text{Tails}\}\}$.
- A probability measure $\mathbb{P} : \mathcal{F} \to [0, 1]$ is a function which assigns a number in $[0, 1]$ to any event (set) in the σ-algebra $\mathcal{F}$. The function $\mathbb{P}$ has the following requirements:
  - σ-additive, which means that if $\bigcup_{i=0}^{\infty} A_i \in \mathcal{F}$ and $A_j \cap A_j = \emptyset,\ i \neq j$, then $\mathbb{P}\left(\bigcup_{i=0}^{\infty} A_i\right) = \sum_{i=0}^{\infty} \mathbb{P}(A_i)$.
  - $\mathbb{P}(\Omega) = 1$, which we can read as the sample space summing up to (integrating) to 1. Without this condition $\mathbb{P}$ would be a regular measure.

Note that we want to be able to measure (assign a probability to) each set in $\mathcal{F}$ rather than 2^{Ω}, since the latter is not always possible. In other words, one can construct sets that cannot be measured (or can only be measured by the trivial measure $\lambda(A) = 0, \forall A \subseteq \Omega$, which is not a probability measure). Thus, we require $\mathcal{F}$ to only contain measurable sets and this is what a σ-algebra guarantees.

Definition 2.1.2. A σ-algebra $\mathcal{F} \subseteq 2^{\Omega}$ is a collection of sets satisfying the property:

- $\mathcal{F}$ contains Ω: $\Omega \in \mathcal{F}$.
- $\mathcal{F}$ is closed under complements: if $A \in \mathcal{F}$, then $\Omega \setminus A \in \mathcal{F}$.
- $\mathcal{F}$ is closed under countable union: if $A_i \in \mathcal{F}\ \forall i$, then $\bigcup_i A_i \in \mathcal{F}$.

Note we use the notation $\mathcal{B}(\mathbb{R}^d)$ for the Borel σ-algebra of $\mathbb{R}^d$, which we can think of as the canonical σ-algebra for $\mathbb{R}^d$ – it is the most compact representation of all measurable sets in $\mathbb{R}^d$. This notation can also be extended for subsets of $\mathbb{R}^d$ and more generally to any topological space.

Definition 2.1.3. For a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, a real-valued random variable (vector) $\mathbf{x}(\omega)$ is a function $\mathbf{x} : \Omega \to \mathbb{R}^d$, requiring that $\mathbf{x}(\omega)$ is a measurable function, meaning that the pre-image of $\mathbf{x}(\omega)$ lies within the σ-algebra $\mathcal{F}$:

$$\mathbf{x}^{-1}(B) = \{\omega : \mathbf{x}(\omega) \in B\} \in \mathcal{F}, \quad \forall B \in \mathcal{B}(\mathbb{R}^d).$$

The above formalism of the random variable allows us to assign a numerical representation to outcomes in Ω. The clear advantage is that we can now ask questions such as what is the probability that $\mathbf{x}$ is contained within a set $B \subseteq \mathbb{R}^d$:

$$\mathrm{P}(\mathbf{x}(\omega) \in B) = \mathbb{P}(\{\omega : \mathbf{x}(\omega) \in B\}),$$

and if we consider the more familiar 1D example, we recover the cumulative distribution function (CDF):

$$\mathrm{P}(\mathrm{x}(\omega) \leq r) = \mathbb{P}(\{\omega : \mathrm{x}(\omega) \leq r\}).$$

The random-variable formalism provides us with a more clear connection between the probability measure $\mathbb{P}$ and the familiar CDF. For simplicity in some cases, we may drop the argument ω from the random variable notation (e.g. $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$).

2.1.1 Lebesgue–Stieltjes Integral

Here we offer a pragmatic introduction to the Lebesgue–Stieltjes integral, in the context of a probability space. For a more technical introduction, we point the reader to advanced probability theory or measure and integration

courses (Salamon, 2019; Viaclovsky, 2003; Andres, 2019).

Definition 2.1.4. For a probability measure space $(\Omega, \mathcal{F}, \mathbb{P})$, a measurable function $f : \Omega \to \mathbb{R}$ and $A \in \mathcal{F}$, the Lebesgue–Stieltjes integral

$$\int_A f(\boldsymbol{x}) d\mathbb{P}(\boldsymbol{x}) \tag{2.1}$$

is a Lebesgue integral with respect to the probability measure $\mathbb{P}$.

Whilst we have not yet introduced a precise definition for the Lebesgue integral, we will now illustrate some of its properties that give us a grasp of this seemingly new notation. Expectations in our probability space can be written as

$$\mathbb{E}_{\mathbb{P}}[f(\mathbf{x})] = \int_{\Omega} f(\boldsymbol{x}) d\mathbb{P}(\boldsymbol{x}). \tag{2.2}$$

Let $\mathbf{1}(\mathbf{x} \in A)$ be the indicator function for set A, then

$$\mathbb{E}_{\mathbb{P}}[\mathbf{1}(\mathbf{x} \in A)] = \int_{\Omega} \mathbf{1}(\mathbf{x} \in A) d\mathbb{P}(\boldsymbol{x}) = \int_A d\mathbb{P}(\boldsymbol{x}) = \mathbb{P}(A). \tag{2.3}$$

The above result is a useful example, since it shows how a distribution (probability measure) is defined in terms of the integral. This is effectively the definition of a cumulative density function. When our distribution $\mathbb{P}$ admits a probability density function (PDF) $p(\boldsymbol{x})$, we have the following:

$$\int_{\Omega} f(\boldsymbol{x}) d\mathbb{P}(\boldsymbol{x}) = \int_{\Omega} f(\boldsymbol{x}) p(\boldsymbol{x}) d\lambda(\boldsymbol{x}), \tag{2.4}$$

where λ is the Lesbesgue measure, and we can think of it as the characteristic measure for $\mathbb{R}^d$. For many purposes, we can interpret $\int_{\Omega} f(\boldsymbol{x}) p(\boldsymbol{x}) d\lambda(\boldsymbol{x})$ as the regular Riemann integral and in many cases authors (Williams & Rasmussen, 2006) use $\int_{\Omega} f(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}$ notationally when the integral is with respect to the Lebesgue measure.

An important takeaway is that, whilst connecting the Lebesgue integral to the standard Riemann integral gives us a useful conceptual connection, it is

not always something that can be done. As we will soon see, many random processes do not admit a PDF. In order to be able to compute expectations with respect to these processes, we must adopt the Lebesgue-Stieltjes integral, which is well-defined in these settings, while the standard Riemann integral is not.

2.2 Stochastic Process Formalism

Informally, a stochastic process is a time-dependent random variable $\mathbf{x}(t)$. In other words, it is a random variable whose distribution at any point in time $P_t(\mathbf{x}(t) < r)$ is itself a function of time.

Definition 2.2.1. Given probability space $(\Omega, \mathcal{F}, \mathbb{P})$, a stochastic process is a collection of random variables $\mathbf{x}(\omega, t) : \Omega \times T \to \mathbb{R}$ indexed by T ($T = \mathbb{R}^+$ when representing time), which can be written as

$$\{\mathbf{x}(\omega, t) : t \in T\}.$$

We will adopt the notation $\mathbf{x}(t)$. More commonly in the statistics community, the notation X_t is used to emphasise that T is an index set.

Stochastic processes are not necessarily limited to temporal processes like the examples we have given so far. In fact, one of the most popular stochastic processes in machine learning, the Gaussian process (GP) for regression, was initially devised for a spatial application known as Kriging. In this thesis, we will focus on the temporal case where $T = \mathbb{R}^+$. Furthermore, we will also restrict ourselves to causal processes[1] in that $\mathbf{x}(t)$ only depends on the present and the past. In order to formalise this notion of causality, we require the concept of a filtration:

Definition 2.2.2. A filtration $\mathfrak{F} = (\mathcal{F}_t)_{i \in T}$ on probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is

[1] Causal in the engineering and physics sense, e.g. causal Green's function.

a sequence of indexed sub-σ-algebras of $\mathcal{F}$:

$$\mathcal{F}_s \subseteq \mathcal{F}_t \subseteq \mathcal{F} \quad , \forall s \leq t.$$

We then call the space $(\Omega, \mathcal{F}, \mathfrak{F}, \mathbb{P})$ an $\mathfrak{F}$-filtered probability space.

At a high level, the construction in Definition 2.2.2 simply creates an indexed sequence of events $(\mathcal{F}_t)_{i \in T}$, which now allows us to define processes that only depend on the past and present:

Definition 2.2.3. A stochastic process $\mathbf{x}$ is $\mathcal{F}_t$-adapted if $\mathbf{x}(t)$ is $\mathcal{F}_t$-measurable:

$$\{\omega : \mathbf{x}(\omega, t) \in B\} \in \mathcal{F}_t \quad , \forall t \in T, \, \forall B \in \mathcal{B}(\mathbb{R}^d).$$

2.2.1 Wiener Process

We will provide the definition of a causal Wiener process ($\mathcal{F}_t$-adapted), since it is the type of processes that we will be working with. More generally, Wiener processes do not have to be $\mathcal{F}_t$-adapted.

Definition 2.2.4. An $\mathcal{F}_t$-adapted Wiener process (Brownian motion) is a stochastic process $\boldsymbol{\beta}(t)$ with the following properties:

- $\boldsymbol{\beta}(0) = \mathbf{0}$,
- $\boldsymbol{\beta}(t) - \boldsymbol{\beta}(s) \perp\!\!\!\perp \mathcal{F}_s, \quad \forall s < t$ (independent increments),
- $\boldsymbol{\beta}(t) - \boldsymbol{\beta}(s) \sim \mathcal{N}(\mathbf{0}, (t-s)\mathbb{I}_d), \quad \forall s < t,$
- $\boldsymbol{\beta}(t)$ is continuous in t.

A simpler way of looking at the above definition is to examine what the joint PDF is for a set of observations $\boldsymbol{\beta}(t_1), ..., \boldsymbol{\beta}(t_n)$ under this process:

$$p(\boldsymbol{\beta}(t_1), ..., \boldsymbol{\beta}(t_n)) = \prod_{i=1}^{n-1} \mathcal{N}\left(\boldsymbol{\beta}(t_{n+1}) \middle| \boldsymbol{\beta}(t_n), (t_{n+1} - t_n)\mathbb{I}_d\right).$$

From here, we can see that $\mathcal{N}\left(\boldsymbol{\beta}(t_{n+1}) \middle| \boldsymbol{\beta}(t_n), (t_{n+1} - t_n)\mathbb{I}_d\right)$, meaning the

transition is given by a random increment from $\boldsymbol{\beta}(t_n)$, i.e.

$$\boldsymbol{\beta}(t_{n+1}) = \boldsymbol{\beta}(t_n) + \sqrt{(t_{n+1} - t_n)}\mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_d).$$

Additionally it follows that the Wiener process is also a Gaussian process, more specifically a Gaussian-Markov process parametrised by mean and covariance functions

$$\boldsymbol{m}(t) = \mathbf{0}, \quad \boldsymbol{k}(t, s) = \mathbb{I}_d \min(t, s).$$

2.2.2 Stochastic Integrals

Stochastic integrals are the integrals induced by a stochastic process. Let's first consider the simplest type of stochastic integral, that is

$$\int_a^b \mathbf{x}(t)dt, \tag{2.5}$$

where $\boldsymbol{x}(\omega, t) : \Omega \times T \to \mathbb{R}^d$ is a $\mathcal{F}_t$-adapted stochastic process. This type of integral appears in machine learning, for example through latent-force models (Alvarez et al., 2009, 2013), where the integrand $\mathbf{x}(t)$ is a Gaussian process. The notation in Equation 2.5, whilst compact, may initially seem confusing to the reader, as $\mathbf{x}(t)$ is not a deterministic function and can effectively take on different values at each time step t.

To clarify the above, we can express Equation 2.5 in more detail:

$$\int_a^b \mathbf{x}(\omega, t)dt, \quad \forall\omega.$$

We basically fix ω (i.e. consider a single-sampled random function) and the resulting integral should exist for all ω. Now we can define the above integral

as a Riemann sum:

$$\int_a^b \mathbf{x}(\omega,t)dt = \lim_{n\to\infty}\sum_{i=1}^{n-1}\mathbf{x}(\omega,t_i^*)(t_{i+1}-t_i),$$
$$\text{where } t_1 = a < t_2 < \ldots < t_n = b,\ t_i^* \in [t_i, t_{i+1}]$$

and the convergence of the limit is defined in the mean-square sense:

$$\lim_{n\to\infty}\mathbb{E}\left[\left|\left|\sum_{i=1}^{n-1}\mathbf{x}(\omega,t_i^*)(t_{i+1}-t_i) - \int_a^b \mathbf{x}(\omega,t)\right|\right|^2\right] = 0.$$

Now it remains to discover when the above limit exists:

Theorem 1. *If a stochastic process* $\mathbf{x}(t)$ *has continuous mean and covariance functions* $\boldsymbol{m}(t) = \mathbb{E}[\mathbf{x}(t)]$, $\boldsymbol{k}(t,s) = Cov(\mathbf{x}(t),\mathbf{x}(s))$, *then the limit* $\int_a^b \mathbf{x}(\omega,t)dt$ *exists.*

If Theorem 1 holds true, then analysing the resulting stochastic process produced by $\int_a^b \mathbf{x}(\omega,t)dt$ can be quite simple. For example, in Alvarez et al. (2009), it was sufficient to compute expectations and covariances of the above integral to fully characterise the resulting process.

Itô Integral

considerring integrals with respect to Brownian motion is more difficult:

$$\int_a^b \mathbf{x}(t)d\boldsymbol{\beta}(t). \tag{2.6}$$

First thing to note is that this takes the form of a Stieltjes integral with respect to another function $\boldsymbol{\beta}(t)$ (which in this case is a random function), rather than the domain of integration t. Naively defining this integral as before is problematic, since the limit is no longer well-defined (unique) for

this case:

$$\int_a^b \mathbf{x}(t)d\boldsymbol{\beta}(t) = \lim_{n\to\infty} \sum_{i=0}^{n-1} \mathbf{x}(t_i^*)(\boldsymbol{\beta}(t_{i+1}) - \boldsymbol{\beta}(t_i)). \tag{2.7}$$

For the above limit to exist, we require that the function $\boldsymbol{\beta}(\omega, t)$ has a bounded total variation in t, which does not happen, since Brownian-motion paths do not have bounded total variation. However, if we fix the choice $t_i^* = t_i$, so that

$$\int_a^b \mathbf{x}(t)d\boldsymbol{\beta}(t) = \lim_{n\to\infty} \sum_{i=0}^{n-1} \mathbf{x}(t_i)(\boldsymbol{\beta}(t_{i+1}) - \boldsymbol{\beta}(t_i)), \tag{2.8}$$

it can be shown that this limit will converge in the mean-square sense. The above integral is known as the Itô integral.

2.2.3 Itô Processes and SDEs

For the purpose of this work, Itô processes will be our definition of stochastic differential equations (SDEs) and we will use both terms to refer to the same object.

Definition 2.2.5. For $\mathcal{F}_t$-adapted stochastic processes $\mathbf{b}(t)$ and $\boldsymbol{\sigma}(t)$, an Itô process $\mathbf{x}(t)$ is defined as

$$\mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \mathbf{b}(s)ds + \int_0^t \boldsymbol{\sigma}(s)d\boldsymbol{\beta}(s). \tag{2.9}$$

Equation 2.9 is often notationally simplified to

$$d\mathbf{x}(t) = \mathbf{b}(t)dt + \boldsymbol{\sigma}(t)d\boldsymbol{\beta}(t). \tag{2.10}$$

The process $\mathbf{b}(t)$ is often referred to as the drift, and $\boldsymbol{\sigma}(t)$ as the volatility. The notation in Equation 2.10 is what we typically refer to as a stochastic

differential equation, since if we "divide" both sides by dt we obtain

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{b}(t) + \boldsymbol{\sigma}(t)\boldsymbol{\epsilon}(t),$$

where $\boldsymbol{\epsilon}(t) \sim \mathcal{N}(\mathbf{0}, \delta(k-s)\mathbb{I}_d)$ is white noise, and is regarded as the derivative of Brownian motion (in some sense). Note that the above representation is purely notational, since most stochastic processes are not differentiable (e.g. Brownian motion). Note that SDEs describe the dynamical evolution of a random variable in time, and thus, one may want to ask what the probability density of such random variable is. For Itô processes of the form

$$d\mathbf{x}(t) = \boldsymbol{b}(\mathbf{x}(t), t)dt + \boldsymbol{\sigma}(\mathbf{x}(t), t)d\boldsymbol{\beta}(t), \tag{2.11}$$

where $\boldsymbol{b} : \mathbb{R}^d \times \mathbb{R}^+ \to \mathbb{R}^d$ and $\boldsymbol{\sigma} : \mathbb{R}^d \times \mathbb{R}^+ \to \mathbb{R}^{d\times d}$ are deterministic functions that parametrise the drift and volatility respectively, we can define a partial differential equation (PDE), that describes the evolution of the PDF as a function of time (Särkkä & Solin, 2019):

Definition 2.2.6. For an Itô process following the form of Equation 2.11, the Fokker-Planck (FPK) equation has the form

$$\partial_t p(\boldsymbol{x}, t) = -\nabla \cdot p(\boldsymbol{x}, t)\boldsymbol{b}(\boldsymbol{x}(t), t) + \frac{1}{2}\sum_{ij} \partial^2_{x_i x_j}[\boldsymbol{\sigma}(\boldsymbol{x}(t), t)\boldsymbol{\sigma}(\boldsymbol{x}(t), t)^\top]_{ij}, \tag{2.12}$$

where $p(\boldsymbol{x}, t)$ is the probability density function of the solution of the SDE equation.

The FPK equation provides us with an alternative representation for the solution of SDEs, via a PDE whose solution describes a PDF. The above result is intuitive: given that the SDE describes the dynamic evolution of a random variable, then we should be able to describe the PDF for said random variable at a given point in time, and it seems natural that said PDF can be described with a differential equation. For more rigorous details of the derivation of the FPK equation see Särkkä & Solin (2019).

Itô's Rule

At a high level, Itô's rule is the equivalent to the change-of-variables rule for integration. We will not be going into much technical detail explaining how to arrive to this rule, but we will be restating it here since it is used in some of our results.

Theorem 2. *(Itô's rule) Assume that* $\mathbf{x}(t)$ *is an Itô process and consider an arbitrary scalar function* $f(\mathbf{x}(t), t)$ *of the process. Then, the Itô SDE for* f *is given by*

$$df = \partial_t f dt + \sum_i \partial_{x_i} f dx_i + \frac{1}{2} \sum_{ij} \partial^2_{x_i x_j} f dx_i dx_j, \tag{2.13}$$

provided that the required partial derivatives exist.

Proof. See Øksendal (2003). □

Note that the above is very similar to the standard change-of-variables formula, apart from an extra quadratic term. For more intuition behind this result, see Särkkä & Solin (2019, Chapter 4).

Euler-Mayurama Discretisation

A particular tool, useful when deriving and interpreting Itô processes, is considering their time discretisations. The Euler-Mayurama (EM) discretisation

Algorithm 1: Euler-Mayurama Discretisation

input: $[t_0, t]$, $p(\boldsymbol{x}(t_0))$, Δt, $d\mathbf{x}(t) = \mathbf{b}(\mathbf{x}(t), t) + \boldsymbol{\sigma}(\boldsymbol{x}(t), t) d\boldsymbol{\beta}(t)$

Divide the interval$[t_0, t]$ into steps of size Δt:
$t_0,\ t_0 + \Delta t,\ \ldots,\ t_0 + k\Delta t,\ \ldots,\ t$

$\mathbf{x}(t_0) \sim p(\boldsymbol{x}(t_0))$

for *each step* k **do**

  $\Delta\boldsymbol{\beta}(t_k) \sim \mathcal{N}(\mathbf{0}, \Delta t \mathbb{I}_d)$

  $\mathbf{x}(t_{k+1}) = \mathbf{x}(t_k) + \boldsymbol{b}(\boldsymbol{x}(t_k), t_k)\Delta t + \boldsymbol{\sigma}(\mathbf{x}(t_k), t_k)\Delta\boldsymbol{\beta}(t_k)$

end

return $\{\mathbf{x}(t_k)\}_k$

will be the method we use throughout this work to sample trajectories, in order to compute expectations and other quantities.

2.3 Radon-Nikodym Derivative

As hinted earlier, we will require the Radon-Nikodym (RN) derivative in order to compute the KL divergence between two stochastic processes. We introduce the RN derivative in this section, as well as certain properties required later.

The RN theorem allows us to write a probability measure in terms of an integral with respect to another probability measure.

Theorem 3. *(Radon-Nikodym theorem) Given probability measures $\mathbb{P}$ and $\mathbb{Q}$, defined on the measurable space $(\Omega, \mathcal{F})$, there exists a measurable function $\frac{d\mathbb{P}}{d\mathbb{Q}} : \Omega \to [0, \infty)$, and for any set $A \subseteq \mathcal{F}$:*

$$\mathbb{P}(A) = \int_A \frac{d\mathbb{P}}{d\mathbb{Q}}(\boldsymbol{x}) d\mathbb{Q}(\boldsymbol{x}), \tag{2.14}$$

where the function $\frac{d\mathbb{P}}{d\mathbb{Q}}(\boldsymbol{x})$ is known as the RN-derivative.

A direct consequence of this result is

$$\int_A f(\boldsymbol{x}) d\mathbb{P}(\boldsymbol{x}) = \int_A f(\boldsymbol{x}) \frac{d\mathbb{P}}{d\mathbb{Q}}(\boldsymbol{x}) d\mathbb{Q}(\boldsymbol{x}).$$

The change of measure above is analogous to the trick employed when we do importance sampling (Martino et al., 2017). The RN derivative is effectively the same as the importance sample weights, and it in fact reduces to a ratio of PDFs for the case when the PDFs of the respective distributions are available.

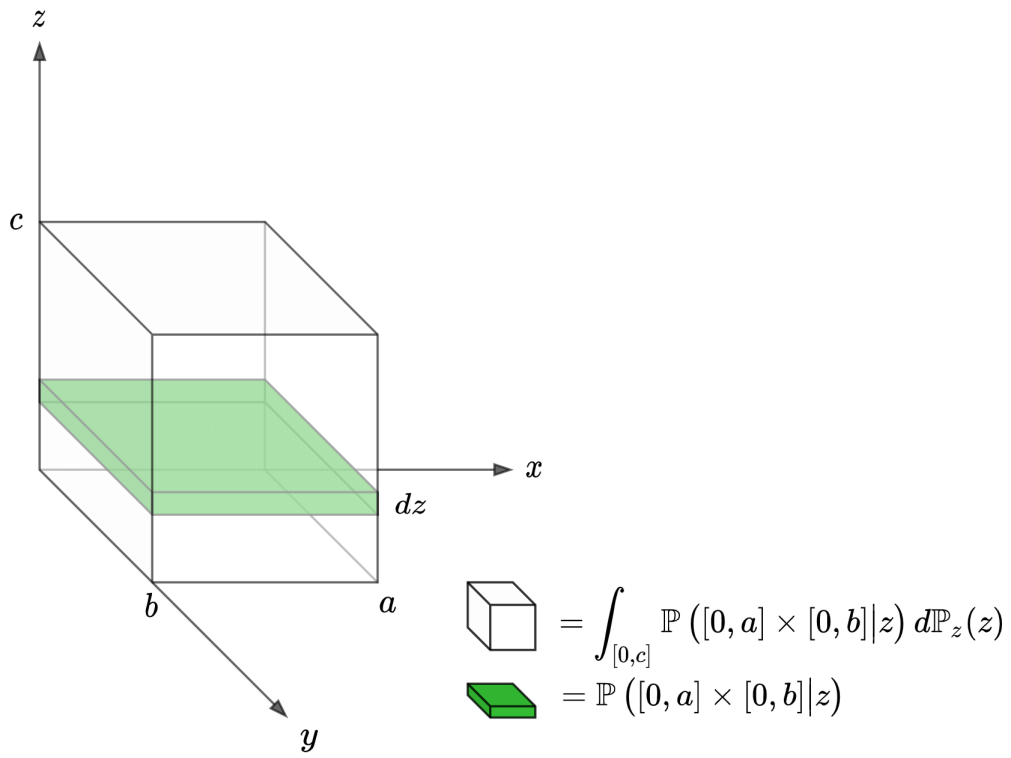

Figure 2.1: This diagram shows that it is possible to construct a conditional measure to calculate the "size" of the green rectangle (whose height tends to 0), however, under the joint measure, such volume evaluates to 0. The Disintegration Theorem provides us with the construction of such conditional measure.

2.3.1 Disintegration Theorem and Conditional Measures

In this section, we will present the Disintegration Theorem in the context of probability measures, which serves as the extension of the product rule to measures that do not admit the traditional product rule. The product rule is an essential rule in machine learning: for example, it is used in factorising the posterior in Bayesian inference (Bayes, 1763). We will use this theorem to present a derivation for the RN-derivative equivalent of the product rule.

Theorem 4. *(Disintegration Theorem for continuous probability measures):*

For a probability space $(Z, \mathcal{B}(Z), \mathbb{P})$*, where* Z *is a product space:* $Z = Z_x \times Z_y$

and

- $Z_x \subseteq \mathbb{R}^d, Z_y \subseteq \mathbb{R}^{d'}$,
- $\pi_i : Z \to Z_i$ *is a measurable function known as the canonical projection operator (i.e.* $\pi_x(z_x, z_y) = z_x$ *and* $\pi_x^{-1}(z_x) = \{y|\pi_x(z_x) = z\}$*),*

there exists a measure $\mathbb{P}_{y|x}(\cdot|\boldsymbol{x})$*, such that*

$$\int_{Z_x \times Z_y} f(\boldsymbol{x}, \boldsymbol{y}) d\mathbb{P}(\boldsymbol{y}) = \int_{Z_x} \int_{Z_y} f(\boldsymbol{x}, \boldsymbol{y}) d\mathbb{P}_{y|x}(\boldsymbol{y}|\boldsymbol{x}) d\mathbb{P}(\pi^{-1}(\boldsymbol{x})), \tag{2.15}$$

where $P_x(\cdot) = \mathbb{P}(\pi^{-1}(\cdot))$ *is a probability measure, typically referred to as a pullback measure, and corresponds to the marginal distribution.*

A direct consequence of the above instance of the disintegration theorem is, with $f(\boldsymbol{x}, \boldsymbol{y}) = \mathbf{1}_{A_x \times A_y}(\boldsymbol{x}, \boldsymbol{y})$,

$$\mathbb{P}(A_x \times A_y) = \int_{A_x} \mathbb{P}(A_y|\boldsymbol{x}) d\mathbb{P}_x(\boldsymbol{x}). \tag{2.16}$$

We can see that, in the context of probability measures, the above is effectively analogous to the product rule.

We now have the required ingredients to show the following:

Lemma 1. *(RN-derivative product rule) Given two probability measures defined on the same product space,* $(Z_x \times Z_y, \mathcal{B}(Z_x \times Z_y), \mathbb{P})$ *and* $(Z_x \times Z_y, \mathcal{B}(Z_x \times Z_y), \mathbb{Q})$*, the Radon–Nikodym derivative* $\frac{d\mathbb{P}}{d\mathbb{Q}}(\boldsymbol{x}, \boldsymbol{y})$ *can be decomposed as*

$$\frac{d\mathbb{P}}{d\mathbb{Q}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{d\mathbb{P}_{y|x}}{d\mathbb{Q}_{y|x}}(\boldsymbol{y}) \frac{d\mathbb{P}_x}{d\mathbb{Q}_x}(\boldsymbol{x}). \tag{2.17}$$

Proof. Starting from

$$\mathbb{P}(A_x \times A_y) = \int_{A_x} \mathbb{P}(A_y|\boldsymbol{x}) d\mathbb{P}_x(\boldsymbol{x}),$$

we apply the Radon-Nikodym theorem to $\mathbb{P}(A_y|\boldsymbol{x})$ and then to P_x:

$$\begin{aligned}\mathbb{P}(A_x \times A_y) &= \int_{A_x}\int_{A_y} \frac{d\mathbb{P}_{y|x}}{d\mathbb{Q}_{y|x}}(\boldsymbol{y}) d\mathbb{Q}_{y|x}(\boldsymbol{y}) d\mathbb{P}_x(\boldsymbol{x}) \\ &= \int_{A_x}\left(\int_{A_y} \frac{d\mathbb{P}_{y|x}}{d\mathbb{Q}_{y|x}}(\boldsymbol{y}) d\mathbb{Q}_{y|x}(\boldsymbol{y})\right) \frac{d\mathbb{P}_x}{d\mathbb{Q}_x}(\boldsymbol{x}) d\mathbb{Q}_x(\boldsymbol{x}) \\ &= \int_{A_x}\int_{A_y} \frac{d\mathbb{P}_x}{d\mathbb{Q}_x}(\boldsymbol{x}) \frac{d\mathbb{P}_{y|x}}{d\mathbb{Q}_{y|x}}(\boldsymbol{y}) d\mathbb{Q}_{y|x}(\boldsymbol{y}) d\mathbb{Q}_x(\boldsymbol{x}).\end{aligned}$$

Now, via the disintegration we have that

$$\int_{A_x\times A_y} \frac{d\mathbb{P}_x}{d\mathbb{Q}_x}(\boldsymbol{x}) \frac{d\mathbb{P}_{y|x}}{d\mathbb{Q}_{y|x}}(\boldsymbol{y}) d\mathbb{Q}(\boldsymbol{x},\boldsymbol{y}) = \int_{A_x}\int_{A_y} \frac{d\mathbb{P}_x}{d\mathbb{Q}_x}(\boldsymbol{x}) \frac{d\mathbb{P}_{y|x}}{d\mathbb{Q}_{y|x}}(\boldsymbol{y}) d\mathbb{Q}_{y|x}(\boldsymbol{y}) d\mathbb{Q}_x(\boldsymbol{x}).$$

Thus, we conclude that

$$\mathbb{P}(A_x \times A_y) = \int_{A_x\times A_y} \frac{d\mathbb{P}_x}{d\mathbb{Q}_x}(\boldsymbol{x}) \frac{d\mathbb{P}_{y|x}}{d\mathbb{Q}_{y|x}}(\boldsymbol{y}) d\mathbb{Q}(\boldsymbol{x},\boldsymbol{y}),$$

which, via the Radon-Nikodym theorem, implies

$$\frac{d\mathbb{P}}{d\mathbb{Q}}(\boldsymbol{x},\boldsymbol{y}) = \frac{d\mathbb{P}_{y|x}}{d\mathbb{Q}_{y|x}}(\boldsymbol{y}) \frac{d\mathbb{P}_x}{d\mathbb{Q}_x}(\boldsymbol{x}).$$

□

The above result is used across a variety of different texts in stochastic processes, nonetheless it has proven difficult to find an original source for it and its derivation. We searched through a variety of graduate courses and books on measure and integration and could not find this result, either stated or derived, thus we decided it would be instructive to provide a sketch proof for it, since we will be using it multiple times.

2.3.2 RN Derivative of Itô Processes

As hinted earlier, Itô processes do not admit a PDF, since they are not absolutely continuous with respect to the Lebesgue measure. However, some

Itô processes are absolutely continuous with respect to one another, and thus we are able to compute their RN derivatives, useful for calculating the KL divergence between the two processes. First, we must introduce the following notation:

Definition 2.3.1. (Path measure) For an Itô process of the form

$$d\mathbf{x}(t) = \mathbf{b}(t) + \boldsymbol{\sigma}(t)d\boldsymbol{\beta}(t),$$

defined in $[0, T]$, we call $\mathbb{P}$ the path measure of the above process, with outcome space $\Omega = C([0, T], \mathbb{R}^d)$, if the distribution $\mathbb{P}$ describes a weak solution to the above SDE[2].

In short, the path measure represents the probability measure associated to the stochastic process specified by the SDE. For example, $\mathbb{W}_\gamma$ is the Wiener measure and represents the path measure of a Wiener process with volatility $\sqrt{\gamma}$ (i.e. $d\mathbf{x}(t) = \sqrt{\gamma}d\boldsymbol{\beta}(t)$). For a more formal introduction to path measures, we require the notion of a *path integral*. We point the reader to Särkkä & Solin (2019); Øksendal (2003) for a more thorough introduction.

We can now present the following theorem (Särkkä & Solin, 2019):

Theorem 5. *(Särkkä & Solin, 2019) Given two Itô processes with the same constant volatility:*

$$\begin{aligned} d\mathbf{x}(t) &= \mathbf{b}_1(t) + \sigma\boldsymbol{\beta}(t), \quad \mathbf{x} = \mathbf{x}_0, \\ d\mathbf{y}(t) &= \mathbf{b}_2(t) + \sigma\boldsymbol{\beta}(t), \quad \mathbf{y} = \mathbf{x}_0, \end{aligned}$$

the RN derivative of their respective path measures $\mathbb{P}, \mathbb{Q}$ is given by

$$\frac{d\mathbb{P}}{d\mathbb{Q}}(\cdot) = \exp\left(-\frac{1}{2\sigma^2}\int_0^t ||\mathbf{b}_1(s) - \mathbf{b}_2(s)||^2 ds + \frac{1}{\sigma^2}\int_0^t (\mathbf{b}_1(s) - \mathbf{b}_2(s))^\top d\boldsymbol{\beta}(s)\right), \tag{2.18}$$

[2]Weak solution is a terminology for a solution of an SDE, that does not take into account an initial value problem and has the freedom of specifying its probability space.

where the type signature of this RN derivative is $\frac{d\mathbb{P}}{d\mathbb{Q}} : C(T, \mathbb{R}^d) \to \mathbb{R}$. For example, if $\mathbf{b}_1(s) = \boldsymbol{b}_1(\mathbf{x}(s), s)$ and $\mathbf{b}_2(s) = \boldsymbol{b}_2(\mathbf{x}(s), s)$ we can write

$$\frac{d\mathbb{P}}{d\mathbb{Q}}(\boldsymbol{x}(t)) = \exp\left(-\frac{1}{2\sigma^2}\int_0^t ||\boldsymbol{b}_1(\boldsymbol{x}(s), s) - \boldsymbol{b}_2(\boldsymbol{x}(s), s)||^2 ds \right. \\ \left. + \frac{1}{\sigma^2}\int_0^t \big(\boldsymbol{b}_1(\boldsymbol{x}(s), s) - \boldsymbol{b}_2(\boldsymbol{x}(s), s)\big)^\top d\boldsymbol{\beta}(s)\right).$$

Note that, in the case where we take the RN derivative with respect to Brownian motion (i.e. $\mathbf{b}_2(t) = 0,\ \sigma = 1$), we have the following expression:

$$\frac{d\mathbb{P}}{d\mathbb{W}}(\cdot) = \exp\left(-\frac{1}{2}\int_0^t ||\mathbf{b}_1(s)||^2 ds + \int_0^t \mathbf{b}_1(s)^\top d\boldsymbol{\beta}(s)\right), \tag{2.19}$$

which is popularly referred to as Girsanov's theorem, as it is one of the main elements in said theorem.

2.4 Summary

So far, we have introduced a set of self-contained mathematical definitions and results. We have briefly motivated the need for these results and some potentially familiar applications to make them more accessible. Now, we will briefly go over the concepts introduced in this chapter and why we will be needing them:

- We have introduced probability measures and defined integration with respect to these measures (i.e. the Lebesgue–Stieltjes integral). We require these formalisms in order to write expectations with respect to stochastic processes which do not admit a formulation as a Riemann integral.
- We have introduced Brownian motion, path measures and stochastic integrals, which are the core tools needed for representing and manipulating SDEs (Itô processes). SDEs are the main object of study in this thesis and a basic understanding is required to follow some of the

algorithms and proofs.

- We have provided a sketch proof of a theorem that allows us to decompose the RN derivative (change of measure ratio) of two probability measures analogous to the product rule of probability. We will be using this to decompose and extremise the KL divergence between two SDEs.
- We have presented how to compute the RN derivative between two SDEs, required when computing the KL divergence between SDEs.

Chapter 3

The Schrödinger Bridge Problem

As originally posed by Schrödinger (1931, 1932), the Schrödinger bridge consists of finding a posterior stochastic evolution between two distributions, that is optimally close to a Brownian-motion prior in a KL sense:

$$\hat{\mathbb{Q}} = \underset{\mathbb{Q}\in\mathcal{D}(\pi_0,\pi_1)}{\arg\min}\, D_{\mathrm{KL}}\left(\mathbb{Q}\middle\|\mathbb{W}\right), \tag{3.1}$$

where $\mathcal{D}(\pi_0, \pi_1)$ represents the set of path measures with marginals π_0 and π_1 and:

$$D_{\mathrm{KL}}\left(\mathbb{Q}\middle\|\mathbb{W}\right) = \mathbb{E}_{\mathbb{Q}}\left[\ln \frac{d\mathbb{Q}}{d\mathbb{W}}\right]. \tag{3.2}$$

To the eye of a probabilistic modeller, this objective may be initially quite confusing, mainly because it is minimising the KL divergence between a "posterior" distribution $\mathbb{Q}$ and a "prior" distribution $\mathbb{W}$, which does not match the usual variational inference objectives that arise from minimising an evidence lower bound (Yang, 2017). In order to give a sound interpretation to the objective in Equation 3.1, we will look into the Schrödinger Bridge's formulation as a rare event, and additionally comment on the maximum-

entropy-like nature of the objective.

3.1 Rare Events and Maximum Entropy

Let our sample space be the space of random functions with the unit interval as pre-image $C([0,1],\mathbb{R}^d)$, meaning a sample represents a function of the form $\boldsymbol{x} : [0,1] :\to \mathbb{R}^d$. We now take a set of i.i.d samples $\{\mathbf{x}_i(t)\}_{i=1}^N$ following standard Brownian motion defined on $C([0,1],\mathbb{R}^d)$. The empirical distribution for $\{\mathbf{x}_i(t)\}_{i=1}^N$ is defined by

$$\hat{\mathbb{W}}(A) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left(\mathbf{x}_i(t) \in A\right), \quad A \in \mathcal{B}(\mathbb{R}^d)^{[0,1]}. \tag{3.3}$$

We then might want to find the probability that the empirical distribution prescribes marginals π_0, π_1 which cannot be attained by Brownian motion:

$$P\left(\hat{\mathbb{W}} \in \mathcal{D}(\pi_0,\pi_1)\right). \tag{3.4}$$

It turns out that a result from the theory of large deviations (Sanov's theorem) allows us to compute an asymptotic expression for such probability (Léonard, 2012):

$$P\left(\hat{\mathbb{W}} \in \mathcal{D}(\pi_0,\pi_1)\right) \sim \exp\left(-N \inf_{\mathbb{Q}\in\mathcal{D}(\pi_0,\pi_1)} D_{\mathrm{KL}}\left(\mathbb{Q}\middle\|\mathbb{W}\right)\right). \tag{3.5}$$

For a more technically thorough introduction, please check Léonard (2013). Note that the exponent for the probability in Equation 3.5 extremises the KL divergence, following the principle of minimum discrimination information by Kullback (1997), which is a generalisation of Edwin Jaynes' maximum entropy principle (Jaynes, 1957, 2003) to continuous distributions. Note one can observe the connection between maximising entropy and minimising KL divergence by considering the discrete setting, where minimising KL divergence with respect to a uniform reference distribution is equivalent to maximising entropy. In simple words, we are selecting a distribution $\hat{\mathbb{Q}}$, subject to

the marginal constraints, that is as close as possible to given prior knowledge $\mathbb{W}$.

We can see how the extremisation in the exponent of Equation 3.5 may be a difficult problem numerically, since it involves two boundary-value constraints, which cannot both be enforced without introducing computational overhead. For example, using the approach of Lagrange multipliers will turn the problem into one of finding a saddle point, rather than the minimisation for which our current tools are better suited.

3.2 Dynamic Formulation

The dynamic formulation of the Schrödinger bridge follows directly from the exponent in the maximum-entropy formulation. The dynamic version of the Schrödinger bridge is written in terms of path measures, that describe the stochastic dynamics defined over the unit interval.

Definition 3.2.1. (Dynamic Schrödinger problem) The dynamic Schrödinger problem is given by

$$\inf_{\mathbb{Q}\in\mathcal{D}(\pi_0,\pi_1)} D_{\mathrm{KL}}\left(\mathbb{Q}\middle\|\mathbb{W}^{\gamma}\right), \tag{3.6}$$

where $\mathbb{Q} \in \mathcal{D}(\pi_0, \pi_1)$ is a path measure with prescribed marginals of π_0, π_1 at times $0, 1$, and $\mathbb{W}_\gamma$ is the Wiener measure with volatility γ (see Definition 2.3.1).

3.2.1 As a Stochastic Control Problem

This formulation of the Schrödinger bridge problem is characterised in terms of extremising a mean-squared-error-styled objective with respect to the drift that describes the SDE for $\mathbb{Q}$. This formulation will allow us to naturally enforce one of the boundary value constraints as an initial value, which will be helpful in the design of an iterative procedure for solving the Schrödinger bridge problem.

The two results presented below are from Pavon & Wakolbinger (1991). For pedagogical reasons, we provide a proof sketch of these two results.

We can represent $\mathbb{Q}$ as a distribution which evolves according to the solution of an SDE of the form

$$d\mathbf{x}(t) = \mathbf{b}^+(t)dt + \sqrt{\gamma}\boldsymbol{\beta}^+(t).$$

Lemma 2. *(Pavon & Wakolbinger, 1991) The KL divergence between $\mathbb{Q}$ and $\mathbb{W}_\gamma$ can be decomposed as*

$$D_{\mathrm{KL}}\left(\mathbb{Q}\middle|\middle|\mathbb{W}^\gamma\right) = D_{\mathrm{KL}}(p_0^{\mathbb{Q}}||p_0^{\mathbb{W}^\gamma}) + \mathbb{E}_{\mathbb{Q}}\left[\int_0^1 \frac{1}{2\gamma}\left|\left|\mathbf{b}^+(t)\right|\right|^2 dt\right]. \tag{3.7}$$

We use the notation $p_t^{\mathcal{X}}$ for the marginal induced by path measure $\mathcal{X}$ at time t. We refer to this expression (Equation 3.7) as the stochastic control formulation of the Schrödinger bridge problem.

Proof. Via the Disintegration Theorem and Theorem 1, we can condition on the endpoint and re-write the RN derivative as

$$\frac{d\mathbb{Q}}{d\mathbb{W}^\gamma} = \frac{p_0^{\mathbb{Q}}}{p_0^{\mathbb{W}^\gamma}} \frac{d\mathbb{Q}_{(0,1]}}{d\mathbb{W}^\gamma_{(0,1]}}\left(\cdot|\mathbf{x}(0) = \boldsymbol{x}\right),$$

where the disintegration $\mathbb{Q}_{(0,1]}\left(\cdot|\mathbf{x}(0) = \boldsymbol{x}\right)$ is a solution to $d\mathbf{x}(t) = \mathbf{b}_t^+ dt + \sqrt{\gamma}\boldsymbol{\beta}^+(t)$. Then, by Theorem 5, we can express the RN derivative in terms of the drift $\mathbf{b}_t^+$:

$$\frac{d\mathbb{Q}}{d\mathbb{W}^\gamma} = \frac{p_0^{\mathbb{Q}}}{p_0^{\mathbb{W}^\gamma}} \exp\left(\int_0^1 \frac{1}{2\gamma}\left|\left|\mathbf{b}^+(t)\right|\right|^2 dt\right).$$

Substituting the above back into the KL divergence completes the result for Theorem 3.7. □

We can equivalently express $\mathbb{Q}$ as the solution to a reverse time diffusion:

$$d\mathbf{x}(t) = \mathbf{b}^-(t)dt + \sqrt{\gamma}\boldsymbol{\beta}^-(t), \tag{3.8}$$

where $\mathbf{x}(t)$ is adapted to the reverse filtration $(\mathcal{F}_i^-)_{i\in T}$, that is $\mathcal{F}_t^- \subseteq \mathcal{F}_s^-,\ s \leq t$[1], and

$$\mathbf{b}^+(t) - \mathbf{b}^-(t) = \gamma \nabla_{\boldsymbol{x}} p(\mathbf{x}(t), t), \tag{3.9}$$

where $p(\mathbf{x}(t), t)$ is the solution to the FPK equation. The above is typically known as Nelson's duality equation and it relates the forward drift to the dual backward drift (Nelson, 1967).

Using reverse diffusion, Pavon & Wakolbinger (1991) decompose the KL divergence as done with the forward diffusion:

Lemma 3. *(Pavon & Wakolbinger, 1991) The KL divergence between $\mathbb{Q}$ and $\mathbb{W}^\gamma$ can be decomposed as*

$$D_{\mathrm{KL}}\left(\mathbb{Q}\middle|\middle|\mathbb{W}^\gamma\right) = D_{\mathrm{KL}}(p_1^{\mathbb{Q}}||p_1^{\mathbb{W}^\gamma}) + \mathbb{E}_{\mathbb{Q}}\left[\int_0^1 \frac{1}{2\gamma}\left|\left|\mathbf{b}^-(t)\right|\right|^2 dt\right]. \tag{3.10}$$

Using the drift-based formulations defined above, Pavon & Wakolbinger (1991) derive alternate (yet equivalent) objectives for the Schrödinger Bridge problem:

- Forward Objective:

$$\begin{aligned}\min_{\mathbb{Q}\in\mathcal{D}(\pi_0,\pi_1)} D_{\mathrm{KL}}\left(\mathbb{Q}\middle|\middle|\mathbb{W}^\gamma\right) &= \min_{\mathbf{b}^+\in\mathcal{B}} \mathbb{E}_{\mathbb{Q}}\left[\int_0^1 \frac{1}{2\gamma}\left|\left|\mathbf{b}^+(t)\right|\right|^2 dt\right], \\ s.t.\quad d\mathbf{x}(t) &= \mathbf{b}^+(t)dt + \sqrt{\gamma}\boldsymbol{\beta}^+(t),\ \ \mathbf{x}(0)\sim\pi_0,\ \ \mathbf{x}(1)\sim\pi_1.\end{aligned} \tag{3.11}$$

[1]This simply means $\mathbf{x}(t)$ is only aware about the future.

- Backward Objective:

$$\min_{\mathbb{Q}\in\mathcal{D}(\pi_0,\pi_1)} D_{\mathrm{KL}}\big(\mathbb{Q}\big|\big|\mathbb{W}^{\gamma}\big) = \min_{\mathbf{b}^{-}\in\mathcal{B}} \mathbb{E}_{\mathbb{Q}}\left[\int_0^1 \frac{1}{2\gamma}\big|\big|\mathbf{b}^{-}(t)\big|\big|^2 dt\right],$$
$$s.t. \quad d\mathbf{x}(t) = \mathbf{b}^{-}(t)dt + \sqrt{\gamma}\boldsymbol{\beta}^{-}(t), \;\; \mathbf{x}(1)\sim\pi_1, \;\; \mathbf{x}(0)\sim\pi_0. \quad (3.12)$$

Where $\mathcal{B}$ represents the space of admissible control signals or equivalently the space of valid drifts. Notice that the conditioning carried out by the disintegration theorem allows us to remove one of the boundary constraints and integrate it as an initial value problem to the objective. This result inspired us the most in the design of an iterative algorithm for solving the Schrödinger bridge numerically.

3.3 Static Formulation

Definition 3.3.1. (Static Schrödinger Problem) The static Schrödinger bridge consists in finding the joint distribution $q(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{D}(\pi_0(\boldsymbol{x}),\pi_1(\boldsymbol{y}))$ which is closest to the Brownian-motion prior subject to marginal constraints, that is

$$\inf_{q(\boldsymbol{x},\boldsymbol{y})} D_{\mathrm{KL}}(q(\boldsymbol{x},\boldsymbol{y})||p^{\mathbb{W}^{\gamma}}(\boldsymbol{x},\boldsymbol{y}))$$
$$s.t. \quad \pi_0(\boldsymbol{x}) = \int q(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{y}, \;\; \pi_1(\boldsymbol{y}) = \int q(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{x}. \quad (3.13)$$

We will now derive this result from the dynamic version. Most surveys and papers on the Schrödinger bridge include some form of this derivation, however they skip several steps which might make it inaccessible.

Theorem 6. *(Föllmer, 1988) The dynamic Schrödinger bridge is solved by*

$$\mathbb{Q}^*(\cdot) = \int \mathbb{W}^{\gamma}(\cdot|\boldsymbol{x},\boldsymbol{y})\, q^*(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{x}d\boldsymbol{y}, \quad (3.14)$$

where q^* is the optimal density that solves the static bridge

$$q^*(\boldsymbol{x},\boldsymbol{y}) = \underset{q(\boldsymbol{x},\boldsymbol{y})}{\arg\inf}\, D_{\mathrm{KL}}(q(\boldsymbol{x},\boldsymbol{y})||p^{\mathbb{W}^\gamma}(\boldsymbol{x},\boldsymbol{y})),$$
$$s.t.\quad \pi_0(\boldsymbol{x}) = \int q(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{y},\quad \pi_1(\boldsymbol{y}) = \int q(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{x},$$

and

$$\mathbb{W}^\gamma\left(\cdot|\boldsymbol{x},\boldsymbol{y}\right) = \mathbb{W}^\gamma_{(0,1)}\left(\cdot|\mathbf{x}(0)=\boldsymbol{x},\mathbf{x}(1)=\boldsymbol{y}\right)$$

is the conditional (disintegration) of the Wiener measure about its endpoints.

Proof. Firstly, we decompose the KL divergence over path measures $D_{\mathrm{KL}}\left(\mathbb{Q}\middle|\middle|\mathbb{W}\right)$ using Lemma 1, and conditioning on the endpoints $\mathbf{x}(0),\mathbf{x}(1)$ we can re-express the RN derivative as

$$\frac{d\mathbb{Q}}{d\mathbb{W}^\gamma} = \frac{q(\boldsymbol{x},\boldsymbol{y})}{p^{\mathbb{W}^\gamma}(\boldsymbol{x},\boldsymbol{y})}\frac{d\mathbb{Q}_{(0,1)}}{d\mathbb{W}^\gamma_{(0,1)}}\left(\cdot|\mathbf{x}(0)=\boldsymbol{x},\mathbf{x}(1)=\boldsymbol{y}\right). \tag{3.15}$$

Let $\mathbb{Q}_{(0,1)}(\cdot|\mathbf{x}(0)=\boldsymbol{x},\mathbf{x}(1)=\boldsymbol{y}) = \mathbb{Q}(\cdot|\boldsymbol{x},\boldsymbol{y})$. Substituting the above decomposition back into the KL divergence and marginalising where possible, we arrive at

$$\begin{aligned} D_{\mathrm{KL}}\left(\mathbb{Q}\middle|\middle|\mathbb{W}^\gamma\right) =& D_{\mathrm{KL}}\left(q(\boldsymbol{x},\boldsymbol{y})\middle|\middle|p^{\mathbb{W}^\gamma}(\boldsymbol{x}.\boldsymbol{y})\right) + \mathbb{E}_{\mathbb{Q}}\left[\frac{d\mathbb{Q}}{d\mathbb{W}^\gamma}\left(\cdot|\boldsymbol{x},\boldsymbol{y}\right)\right] \\ =& D_{\mathrm{KL}}\left(q(\boldsymbol{x},\boldsymbol{y})\middle|\middle|p^{\mathbb{W}^\gamma}(\boldsymbol{x}.\boldsymbol{y})\right) + \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{y})}\left[D_{\mathrm{KL}}\left(\mathbb{Q}(\cdot|\boldsymbol{x},\boldsymbol{y})\middle|\middle|\mathbb{W}^\gamma(\cdot|\boldsymbol{x},\boldsymbol{y})\right)\right]. \end{aligned} \tag{3.16}$$

Notice that the conditional $\mathbb{Q}_{(0,1)}(\cdot|\mathbf{x}(0)=\boldsymbol{x},\mathbf{x}(1)=\boldsymbol{y})$ is not affected by the boundary constraints, thus we can set $\mathbb{Q}(\cdot|\boldsymbol{x},\boldsymbol{y}) = \mathbb{W}^\gamma\left(\cdot|\boldsymbol{x},\boldsymbol{y}\right)$, making the second term 0, and leaving us with the static bridge. It then suffices to reverse the disintegration theorem in order to build up $\mathbb{Q}^*$ from q^* and $\mathbb{Q}^*(\cdot|\boldsymbol{x},\boldsymbol{y})$, completing the proof. □

The earliest references we found for the above result are given by Föllmer (1988), where the decomposition of the KL divergence for two diffusions

(Equation 3.16) is provided. However, we were not able to find a good reference for Lemma 1 and thus we have provided an instructive derivation for it.

3.3.1 As an Entropy-Regularised Optimal Transport Problem

The optimal transport problem in its intuitive form is concerned with the transportation of a distribution into another distribution, subject to cost $c(\boldsymbol{x}, \boldsymbol{y})$. In its discrete histogram-based formulation, it is referred to as the earth mover's distance (EMD) (Levina & Bickel, 2001), since it can be illustrated as moving dirt between two piles of earth. One of the early applications of EMD was in comparing grey-scale images for the development of an image retrieval system.

Definition 3.3.2. Formally, the optimal transport problem is given by the following objective:

$$\inf_{Q \in \mathcal{D}(\pi_0, \pi_1)} \int c(\boldsymbol{x}, \boldsymbol{y}) dQ(\boldsymbol{x}, \boldsymbol{y}),$$

and if Q is absolutely continuous w.r.t. to the Lebesgue measure, we can express the above as

$$\inf_{Q \in \mathcal{D}(\pi_0, \pi_1)} \int \int c(\boldsymbol{x}, \boldsymbol{y}) q(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y}.$$

Furthermore, when $c(\boldsymbol{x}, \boldsymbol{y}) = ||\boldsymbol{x} - \boldsymbol{y}||^2$, the above resembles the square of the Wasserstein distance $\mathcal{W}_2^2(\pi_0, \pi_1)$.

Here, we will present and discuss a very well-studied (Mikami & Thieullen, 2008; Léonard, 2012, 2013; Carlier et al., 2017) connection between the static bridge and the Wasserstein distance. For the Wiener process $\mathbb{W}^\gamma$ prior, we have

$$p^{\mathbb{W}^\gamma}(\boldsymbol{x}, \boldsymbol{y}) = p_0^{\mathbb{W}^\gamma}(\boldsymbol{x}) \mathcal{N}(\boldsymbol{y} | \boldsymbol{x}, \gamma \mathbb{I}_d), \tag{3.17}$$

and the term

$$\int q(\boldsymbol{x},\boldsymbol{y}) \ln p_0^{\mathbb{W}^\gamma}(\boldsymbol{x}) d\boldsymbol{x} d\boldsymbol{y} = \int \pi_0(\boldsymbol{x}) \ln p_0^{\mathbb{W}^\gamma}(\boldsymbol{x}) d\boldsymbol{x}$$

does not depend on q (due to the constraints). Substituting the above into Equation 3.13, we arrive at

$$\begin{aligned} &\inf_{q\in\mathcal{D}(\pi_0,\pi_1)} \int\int -\frac{||\boldsymbol{x}-\boldsymbol{y}||^2}{2\gamma} q(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} + \int\int q(\boldsymbol{x},\boldsymbol{y}) \ln q(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y}, \\ &= \inf_{q\in\mathcal{D}(\pi_0,\pi_1)} \int\int -\frac{||\boldsymbol{x}-\boldsymbol{y}||^2}{2} q(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} - \gamma \mathrm{H}\left(q(\boldsymbol{x},\boldsymbol{y})\right), \end{aligned} \tag{3.18}$$

which is an entropy-regularised optimal mass transport problem (OMT) (Villani, 2003) with a quadratic cost function. Furthermore, as the volatility/noise of the Brownian-motion prior goes to 0 ($\gamma \downarrow 0$), the above quantity converges to $\mathcal{W}_2^2(\pi_0,\pi_1)$ (squared Wasserstein distance in an L_2 metric space).

3.3.2 The Schrödinger System

Following Pavon et al. (2018), the Lagrangian of Equation 3.13 is given by

$$\begin{aligned} \mathcal{L}(q,\lambda,\mu) = &D_{\mathrm{KL}}(q(\boldsymbol{x},\boldsymbol{y})||p^{\mathbb{W}^\gamma}(\boldsymbol{x},\boldsymbol{y})) \\ &+ \int \lambda(\boldsymbol{x}) \left(\int q(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{y} - \pi_0(\boldsymbol{x}) \right) d\boldsymbol{x} \\ &+ \int \mu(\boldsymbol{y}) \left(\int q(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{x} - \pi_1(\boldsymbol{y}) \right) d\boldsymbol{y}. \end{aligned} \tag{3.19}$$

Let $p^{\mathbb{W}^\gamma}(\boldsymbol{x},\boldsymbol{y}) = p_0^{\mathbb{W}^\gamma}(\boldsymbol{x}) p^{\mathbb{W}^\gamma}(\boldsymbol{y}|\boldsymbol{x})$, where $p_0^{\mathbb{W}^\gamma}(\boldsymbol{x})$ is the marginal prior, which we are free to set, and $p^{\mathbb{W}^\gamma}(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{x},\gamma\mathbb{I}_d)$ is the transition density of the prior. Then, we set the functional derivative $\frac{\delta\mathcal{L}}{\delta q(\boldsymbol{x},\boldsymbol{y})}$ to 0 and obtain

$$1 + \ln q(\boldsymbol{x},\boldsymbol{y}) - \ln p^{\mathbb{W}^\gamma}(\boldsymbol{y}|\boldsymbol{x}) - \ln p_0^{\mathbb{W}^\gamma}(\boldsymbol{x}) + \lambda(\boldsymbol{x}) + \mu(\boldsymbol{y}) = 0.$$

Rearranging, we get

$$q^*(\boldsymbol{x},\boldsymbol{y}) = \exp\left(\ln p_0^{\mathbb{W}^\gamma}(\boldsymbol{x}) - \lambda(\boldsymbol{x}) - 1\right) p^{\mathbb{W}^\gamma}(\boldsymbol{y}|\boldsymbol{x}) \exp\left(-\mu(\boldsymbol{y})\right),$$

which we can re-express in terms of the auxiliary potentials $\hat{\phi}_0(\boldsymbol{x}), \phi_1(\boldsymbol{y})$:

$$q^*(\boldsymbol{x},\boldsymbol{y}) = \hat{\phi}_0(\boldsymbol{x}) p^{\mathbb{W}^\gamma}(\boldsymbol{y}|\boldsymbol{x}) \phi_1(\boldsymbol{y}),$$

satisfying

$$\begin{aligned}
\hat{\phi}_0(\boldsymbol{x}) \int \phi_1(\boldsymbol{y}) p^{\mathbb{W}^\gamma}(\boldsymbol{y}|\boldsymbol{x}) d\boldsymbol{y} &= \pi_0(\boldsymbol{x}), \\
\phi_1(\boldsymbol{y}) \int \hat{\phi}_0(\boldsymbol{x}) p^{\mathbb{W}^\gamma}(\boldsymbol{y}|\boldsymbol{x}) d\boldsymbol{x} &= \pi_1(\boldsymbol{y}),
\end{aligned}$$

where we re-label the terms with the integrals to

$$\begin{aligned}
\phi_0(\boldsymbol{x}) &= \int \phi_1(\boldsymbol{y}) p^{\mathbb{W}^\gamma}(\boldsymbol{y}|\boldsymbol{x}) d\boldsymbol{y}, \\
\hat{\phi}_1(\boldsymbol{y}) &= \int \hat{\phi}_0(\boldsymbol{x}) p^{\mathbb{W}^\gamma}(\boldsymbol{y}|\boldsymbol{x}) d\boldsymbol{x}.
\end{aligned}$$

Putting it all together, the following linear functional system is known as the Schrödinger system:

$$\begin{aligned}
\hat{\phi}_0(\boldsymbol{x})\phi_1(\boldsymbol{x}) &= \pi_0(\boldsymbol{x}), \\
\hat{\phi}_1(\boldsymbol{x})\phi_1(\boldsymbol{y}) &= \pi_1(\boldsymbol{y}).
\end{aligned} \tag{3.20}$$

We obtain the distribution $\pi_t^*(\boldsymbol{z})$ (solution to the FPK equation for the optimal Q^*):

$$\pi_t^*(\boldsymbol{z}) = \hat{\phi}_t(\boldsymbol{z})\phi_t(\boldsymbol{z}), \tag{3.21}$$

where

$$\phi_t(\boldsymbol{z}) = \int \phi_1(\boldsymbol{y}(1)) p^{\mathbb{W}^\gamma}(\boldsymbol{y}(1)|\boldsymbol{z}(t)) d\boldsymbol{y}(1),$$
$$\hat{\phi}_t(\boldsymbol{z}) = \int \hat{\phi}_0(\boldsymbol{x}(0)) p^{\mathbb{W}^\gamma}(\boldsymbol{z}(t)|\boldsymbol{x}(0)) d\boldsymbol{x}(0).$$

Note that, by Proposition 3.3 of Pavon & Wakolbinger (1991), the optimal control signal/drift $\mathbf{b}_t^+$ can be recovered from the solution of the Schrödinger system:

$$\mathbf{b}_t^+ = \gamma \nabla \ln \phi_t(\boldsymbol{x}(t)). \tag{3.22}$$

The Schrödinger system is an equivalent formulation to the static Schrödinger problem in terms of the potential functions. It is by no means a method or an algorithm for solving the problem. One can solve the system in closed form, when the integrals in Equation 3.20 admit a simple solution, by guessing for the potentials in a similar way to how we formulate an *ansatz* for simple differential equations.

3.4 Half Bridges

Having only one boundary constraint simplifies the original problem, since it becomes trivial to enforce such constraint numerically. In this section, we will study in more detail such single-constraint problems and illustrate why exactly this is a simpler problem. This construction will be of use, since it will constitute part of the steps in the final algorithms for solving the full bridge problem.

The half-bridge problem, as presented in Pavon et al. (2018), is a simpler variant of the full Schrödinger bridge with only one boundary constraint.

Definition 3.4.1. The forward half bridge is given by

$$\mathbb{Q}^* = \inf_{\mathbb{Q} \in \mathcal{D}(\pi_0, \cdot)} D_{\mathrm{KL}}(\mathbb{Q} || \mathbb{W}^\gamma). \tag{3.23}$$

Theorem 7. *The forward half bridge admits the following solution:*

$$\mathbb{Q}^*\left(A_0 \times A_{(0,1]}\right) = \int_{A_0 \times A_{(0,1]}} \frac{d\pi_0}{dp_0^{\mathbb{W}^\gamma}} d\mathbb{W}^\gamma. \tag{3.24}$$

Proof. Via the disintegration theorem, we have the following decomposition of KL:

$$D_{\mathrm{KL}}(\mathbb{Q}||\mathbb{W}^\gamma) = D_{\mathrm{KL}}(p_0^{\mathbb{Q}}||p_0^{\mathbb{W}}) + \mathbb{E}_p\left[D_{\mathrm{KL}}(\mathbb{Q}(\cdot|\boldsymbol{x})||\mathbb{W}^\gamma(\cdot|\boldsymbol{x}))\right].$$

Thus, via matching terms accordingly, we can construct $\mathbb{Q}^*$ by setting $\mathbb{Q}(\cdot|\boldsymbol{x}) = \mathbb{W}^\gamma(\cdot|\boldsymbol{x})$ and matching the constraints:

$$\mathbb{Q}^*\left(A_0 \times A_{(0,1]}\right) = \int_{A_0 \times A_{(0,1]}} \mathbb{W}^\gamma(A_{(0,1]}|\boldsymbol{x}) d\pi_0(\boldsymbol{x}), \tag{3.25}$$

$$\begin{aligned}\mathbb{Q}^* &= \int_{A_0} \frac{d\pi_0}{dp_0^{\mathbb{W}^\gamma}}(\boldsymbol{x}) \mathbb{W}^\gamma(\cdot|\boldsymbol{x}) dp_0^{\mathbb{W}^\gamma}(\boldsymbol{x}) \\ &= \int_{A_0 \times A_{(0,1]}} \frac{d\pi_0}{dp_0^{\mathbb{W}}}(\boldsymbol{x}) d\mathbb{W}^\gamma. \end{aligned} \tag{3.26}$$

□

Definition 3.4.2. The backward half bridge is given by

$$\mathbb{P}^* = \inf_{\mathbb{P} \in \mathcal{D}(\cdot, \pi_1)} D_{\mathrm{KL}}(\mathbb{P}||\mathbb{W}^\gamma). \tag{3.27}$$

Theorem 8. *The backward half bridge admits the following solution:*

$$\mathbb{P}^*\left(A_{[0,1)} \times A_1\right) = \int_{A_{[0,1)} \times A_1} \frac{d\pi_1}{dp_1^{\mathbb{W}^\gamma}} d\mathbb{W}^\gamma. \tag{3.28}$$

Proof. Same as Theorem 7. □

Note how the main difference between the full and half bridges is that the half bridge admits a closed-form solution in terms of known quantities. Similarly

to the full bridge, the half bridges admit a static formulation:

Definition 3.4.3. The static forward bridge is given by the following objective:

$$\inf_{q(\boldsymbol{x},\boldsymbol{y})} D_{\mathrm{KL}}(q(\boldsymbol{x},\boldsymbol{y})||p^{\mathbb{W}^\gamma}(\boldsymbol{x},\boldsymbol{y})),$$
$$s.t. \quad \pi_0(\boldsymbol{x}) = \int q(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{y}. \tag{3.29}$$

Theorem 9. *The static forward bridge admits the following solution:*

$$q^*(\boldsymbol{x},\boldsymbol{y}) = p^{\mathbb{W}^\gamma}(\boldsymbol{x},\boldsymbol{y})\frac{\pi_0(\boldsymbol{x})}{p^{\mathbb{W}^\gamma}(\boldsymbol{x})}. \tag{3.30}$$

Proof. See Pavon et al. (2018) for the solution of the backward half bridge. Proof is simple and easy to adapt to the forward bridge. □

Definition 3.4.4. The static backward bridge is given by the following objective:

$$\inf_{p(\boldsymbol{x},\boldsymbol{y})} D_{\mathrm{KL}}(p(\boldsymbol{x},\boldsymbol{y})||p^{\mathbb{W}^\gamma}(\boldsymbol{x},\boldsymbol{y})),$$
$$s.t. \quad \pi_1(\boldsymbol{y}) = \int p(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{x}. \tag{3.31}$$

Theorem 10. *The static backward bridge admits the following solution:*

$$p^*(\boldsymbol{x},\boldsymbol{y}) = p^{\mathbb{W}^\gamma}(\boldsymbol{x},\boldsymbol{y})\frac{\pi_1(\boldsymbol{y})}{p^{\mathbb{W}^\gamma}(\boldsymbol{y})}. \tag{3.32}$$

Proof. See Pavon et al. (2018). □

Half bridges are a significantly easier problem than full bridges. Not only do they admit "closed-form" solutions in some sense, they also allow removing constraints by incorporating them as an initial value problem. Going forward, we will see how to build an iterative scheme for solving the full bridge that makes use of solving half bridge sub-problems at each iteration.

3.5 Summary

In this chapter, we have introduced the dynamic and static formulations of the Schrödinger bridge problem, together with the following equivalent formulations:

- A stochastic control formulation that parametrises the dynamic bridge in terms of a drift and allows to incorporate a boundary constraint as an initial value problem.
- A system of functions known as the Schrödinger system, parametrised in terms of potentials. The solution of this system solves the static Schrödinger bridge.

We will be using these alternate formulations in the design of numerical algorithms for solving the Schrödinger bridge problem.

Chapter 4

Iterative Proportional Fitting Procedure

So far we have introduced the Schrödinger bridge problem as well as its simpler half bridge variant. We have also introduced the Schrödinger system and loosely hinted at a potential iterative solution, but we have not yet presented a full solution and discussed its guarantees/properties.

In this chapter, we will study an algorithmic framework known as the Iterative Proportional Fitting Procedure (IPFP) (Csiszár, 1975; Kullback, 1968; Ruschendorf et al., 1995; Cramer, 2000) and describe its usage for solving the Schrödinger bridge. Furthermore, we will formalise a previously made observation that connects Fortet's Iterative scheme (Fortet, 1940) for solving the Schrödinger system with the more general IPFP.

4.1 Fortet's Algorithm

Let us start by introducing Fortet's algorithm, which is maybe the oldest algorithm with a proof of convergence (Fortet, 1940) for solving the Schrödinger system.

Modern adaptations for the proof of convergence for Algorithm 2 can be

Algorithm 2: Fortet's Iterative Procedure

input: $\pi_0(\boldsymbol{x}), \pi_1(\boldsymbol{y}), p(\boldsymbol{y}|\boldsymbol{x})$

Initialise $\phi_0^{(0)}(\boldsymbol{x})$ such that $\phi_0^{(0)}(\boldsymbol{x}) << \pi_0(\boldsymbol{x})$

repeat

  $\hat{\phi}_0^{(i)}(\boldsymbol{x}) := \frac{\pi_0(\boldsymbol{x})}{\phi_0^{(i)}(\boldsymbol{x})}$

  $\hat{\phi}_1^{(i)}(\boldsymbol{y}) := \int p(\boldsymbol{y}|\boldsymbol{x})\hat{\phi}_0^{(i)}(\boldsymbol{x})d\boldsymbol{x}$

  $\phi_1^{(i)}(\boldsymbol{y}) := \frac{\pi_1(\boldsymbol{y})}{\hat{\phi}_1^{(i)}(\boldsymbol{y})}$

  $\phi_0^{(i+1)}(\boldsymbol{x}) := \int p(\boldsymbol{y}|\boldsymbol{x})\phi_1^{(i)}(\boldsymbol{y})d\boldsymbol{y}$

  $i := i + 1$

until *convergence*;

return $\hat{\phi}_0^{(i)}(\boldsymbol{x}), \phi_1^{(i)}(\boldsymbol{y})$

found in (Essid & Pavon, 2019; Chen et al., 2016), however, these proofs are beyond the scope of this thesis. Let us now provide an intuition behind each step in the algorithm:

- $\hat{\phi}_0^{(i)}(\boldsymbol{x}) := \frac{\pi_0(\boldsymbol{x})}{\phi_0^{(i)}(\boldsymbol{x})}$: enforces the marginal/boundary constraint at time 0. That is, it enforces that the product of the factors/potentials matches the marginal distribution π_0.

- $\hat{\phi}_1^{(i)}(\boldsymbol{y}) := \int p(\boldsymbol{y}|\boldsymbol{x})\hat{\phi}_0^{(i)}(\boldsymbol{x})d\boldsymbol{x}$: Now that we have the factors $\phi_0^{(i)}(\boldsymbol{x})\hat{\phi}_0^{(i)}(\boldsymbol{x})$ $:= \pi_0(\boldsymbol{x})$ for the marginal at $t = 0$, we transport/transition them to the marginal at time $t = 1$ by marginalising the current estimate of the joint posterior $\pi_1(\boldsymbol{y}) = \phi_1(\boldsymbol{y}) \int p(\boldsymbol{y}|\boldsymbol{x})\hat{\phi}_0^{(i)}(\boldsymbol{x})d\boldsymbol{x}$.

- $\phi_1^{(i)}(\boldsymbol{y}) := \frac{\pi_1(\boldsymbol{y})}{\hat{\phi}_1^{(i)}(\boldsymbol{y})}$: As with $t = 0$, we enforce the marginal/boundary constraint such that $\phi_1^{(i)}(\boldsymbol{y})\hat{\phi}_1^{(i)}(\boldsymbol{y}) := \pi_1(\boldsymbol{y})$.

- $\phi_0^{(i+1)}(\boldsymbol{x}) := \int p(\boldsymbol{y}|\boldsymbol{x})\phi_1^{(i)}(\boldsymbol{y})d\boldsymbol{y}$: Now that we have enforced the constraint for $t = 1$, we marginalise our current estimate of the joint to move from $\boldsymbol{y}$ to $\boldsymbol{x}$ and repeat.

The potential at time $t = 0$ is initialised with the prior marginal. Then, we iterate the Schrödinger system until reaching a fixed point. The marginal constraints are satisfied by alternating between the $t = 0$ and $t = 1$ marginals.

This leads us to the following observation:

Observation 1. *Fortet's algorithm starting at $t = 1$, rather than $t = 0$, is equivalent to sequentially alternating between solving forward and backward static half bridges.*

Algorithm 3: Alternating half bridges (Kullback (1968) IPFP)

input: $\pi_0(\boldsymbol{x}), \pi_1(\boldsymbol{y}), p(\boldsymbol{y}|\boldsymbol{x})$

Initialise:
$p_1^{\mathbb{W}^\gamma}(\boldsymbol{y})$ such that $p_1^{\mathbb{W}^\gamma}(\boldsymbol{y}) << \pi_1(\boldsymbol{y})$
$q_0^*(\boldsymbol{x}, \boldsymbol{y}) := p^{\mathbb{W}^\gamma}(\boldsymbol{x}, \boldsymbol{y})$
$i = 0$
repeat
  $i := i + 1$
  $p_i^*(\boldsymbol{x}, \boldsymbol{y}) = \inf_{p(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}(\cdot,\pi_1)} D_{\mathrm{KL}}(p(\boldsymbol{x}, \boldsymbol{y})||p_{i-1}^*(\boldsymbol{x}, \boldsymbol{y}))$
  $q_i^*(\boldsymbol{x}, \boldsymbol{y}) = \inf_{q(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}(\pi_0,\cdot)} D_{\mathrm{KL}}(q(\boldsymbol{x}, \boldsymbol{y})||p_i^*(\boldsymbol{x}, \boldsymbol{y}))$
until *convergence*;
return $q_i^*(\boldsymbol{x}, \boldsymbol{y}), p*_i(\boldsymbol{x}, \boldsymbol{y})$

Proof. Consider the first two iterations of Algorithm 3:

Using Theorem 10, the solution to the backward half bridge

$$\inf_{p(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}(\cdot,\pi_1)} D_{\mathrm{KL}}(p(\boldsymbol{x}, \boldsymbol{y})||p^{\mathbb{W}^\gamma}(\boldsymbol{x}, \boldsymbol{y}))$$

is

$$p_0^*(\boldsymbol{x}, \boldsymbol{y}) = p_0^{\mathbb{W}^\gamma}(\boldsymbol{x})p^{\mathbb{W}^\gamma}(\boldsymbol{y}|\boldsymbol{x})\frac{\pi_1(\boldsymbol{y})}{p_1^{\mathbb{W}^\gamma}(\boldsymbol{y})}. \tag{4.1}$$

We set up the following forward bridge:

$$\arg\inf_{q_0(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}(\pi_0,\cdot)} D_{\mathrm{KL}}(q_0(\boldsymbol{x}, \boldsymbol{y})||p_0^*(\boldsymbol{x}, \boldsymbol{y})), \tag{4.2}$$

which, following Theorem 3.29, can be solved by

$$q_0^*(\boldsymbol{x}, \boldsymbol{y}) = p_0^*(\boldsymbol{x}, \boldsymbol{y})\frac{\pi_0(\boldsymbol{x})}{p_0^*(\boldsymbol{x})} \tag{4.3}$$

$$= p^{\mathbb{W}^\gamma}(\boldsymbol{y}|\boldsymbol{x})\frac{\pi_1(\boldsymbol{y})}{p_1^{\mathbb{W}^\gamma}(\boldsymbol{y})}\frac{\pi_0(\boldsymbol{x})}{\int p^{\mathbb{W}^\gamma}(\boldsymbol{y}|\boldsymbol{x})\frac{\pi_1(\boldsymbol{y})}{p_1^{\mathbb{W}^\gamma}(\boldsymbol{y})}d\boldsymbol{y}}. \tag{4.4}$$

If we proceed to the second iteration of this procedure, the backward bridge step will yield

$$p_1^*(\boldsymbol{x},\boldsymbol{y}) = q_0^*(\boldsymbol{x},\boldsymbol{y})\frac{p_1^{\mathbb{W}^\gamma}(\boldsymbol{y})}{\underbrace{\int \frac{p^{\mathbb{W}^\gamma}(\boldsymbol{y}|\boldsymbol{x})\pi_0(\boldsymbol{x})}{\int p^{\mathbb{W}^\gamma}(\boldsymbol{y}|\boldsymbol{x})\frac{\pi_1(\boldsymbol{y})}{p_1^{\mathbb{W}^\gamma}(\boldsymbol{y})}d\boldsymbol{y}}d\boldsymbol{x}}_{\phi^1(\boldsymbol{y})}}, \tag{4.5}$$

and the forward step:

$$q_1*(\boldsymbol{x},\boldsymbol{y}) = p_1^*(\boldsymbol{x},\boldsymbol{y})\frac{\int p^{\mathbb{W}^\gamma}(\boldsymbol{y}|\boldsymbol{x})\frac{\pi_1(\boldsymbol{y})}{p_1^{\mathbb{W}^\gamma}(\boldsymbol{y})}d\boldsymbol{y}}{\int p^{\mathbb{W}^\gamma}(\boldsymbol{y}|\boldsymbol{x})\frac{\pi_1(\boldsymbol{y})}{\phi^1(\boldsymbol{y})}d\boldsymbol{y}}. \tag{4.6}$$

Re-labeling, we get

$$\phi^0(\boldsymbol{y}) = p_1^{\mathbb{W}^\gamma}(\boldsymbol{y}), \tag{4.7}$$

$$\hat{\phi}_0^i(\boldsymbol{x}) = \int p^{\mathbb{W}^\gamma}(\boldsymbol{y}|\boldsymbol{x})\frac{\pi_1(\boldsymbol{y})}{\phi^i(\boldsymbol{y})}d\boldsymbol{y}, \tag{4.8}$$

and we can make an inductive argument on i for the following inductive hypothesis[1]:

$$\phi^{i+1}(\boldsymbol{y}) = \int \frac{p^{\mathbb{W}^\gamma}(\boldsymbol{y}|\boldsymbol{x})\pi_0(\boldsymbol{x})}{\int p^{\mathbb{W}^\gamma}(\boldsymbol{y}|\boldsymbol{x})\frac{\pi_1(\boldsymbol{y})}{\phi^i(\boldsymbol{y})}d\boldsymbol{y}}d\boldsymbol{x}, \tag{4.9}$$

$$p_i^*(\boldsymbol{x},\boldsymbol{y}) = q_{i-1}^*(\boldsymbol{x},\boldsymbol{y})\frac{\phi^{i-1}(\boldsymbol{y})}{\phi^i(\boldsymbol{y})}, \tag{4.10}$$

$$q_i*(\boldsymbol{x},\boldsymbol{y}) = p_i^*(\boldsymbol{x},\boldsymbol{y})\frac{\hat{\phi}_0^{i-1}(\boldsymbol{x})}{\hat{\phi}_0^i(\boldsymbol{x})}. \tag{4.11}$$

[1]Some simple term-cancellation steps have been omitted to reduce verbosity.

We can see that the recurrence in Equation 4.9 is the exact same set of steps performed by Fortet's algorithm, if we were to reverse the time order or start Fortet's algorithm at step 5 and consider the first steps as an initialisation of $\phi_0(\boldsymbol{y})$. □

Whilst in hindsight the above observation may seem simple, we regard it as a contribution, as we have not observed a formal argument made towards this connection. As we will see in the following sections, Observation 1 establishes a link between Fortet's algorithm and algorithms derived from the iterative proportional fitting procedure.

4.2 Kullback's IPFP

The original iterative proportional fitting procedure (IPFP) consists of estimating a normalised contingency table (discrete joint distribution) given prescribed marginals, via some form of information-discrimination/maximum-entropy principle. However, we are interested in the continuous variant of IPFP, which dates back to Kullback (Kullback, 1968).

What we call in this section Kullback's IPFP is in fact Algorithm 3, which we have introduced via its formal connection to Fortet's algorithm. The first complete proof of convergence to Algorithm 3 was provided in Ruschendorf et al. (1995), using information-geometric arguments from Csiszár (1975).

4.3 Generalised IPFP

We call generalised iterative proportional fitting procedure (g-IPFP) the extension of the continuous IPFP, initially proposed by Kullback to a more general setting over path measures, as presented in Cramer (2000); Bernton et al. (2019). The method is identical to the regular IPFP, only that it re-states the problem in terms of probability measures.

Using Theorems 7 and 8, steps 6 and 7 of Algorithm 4 can be expressed in

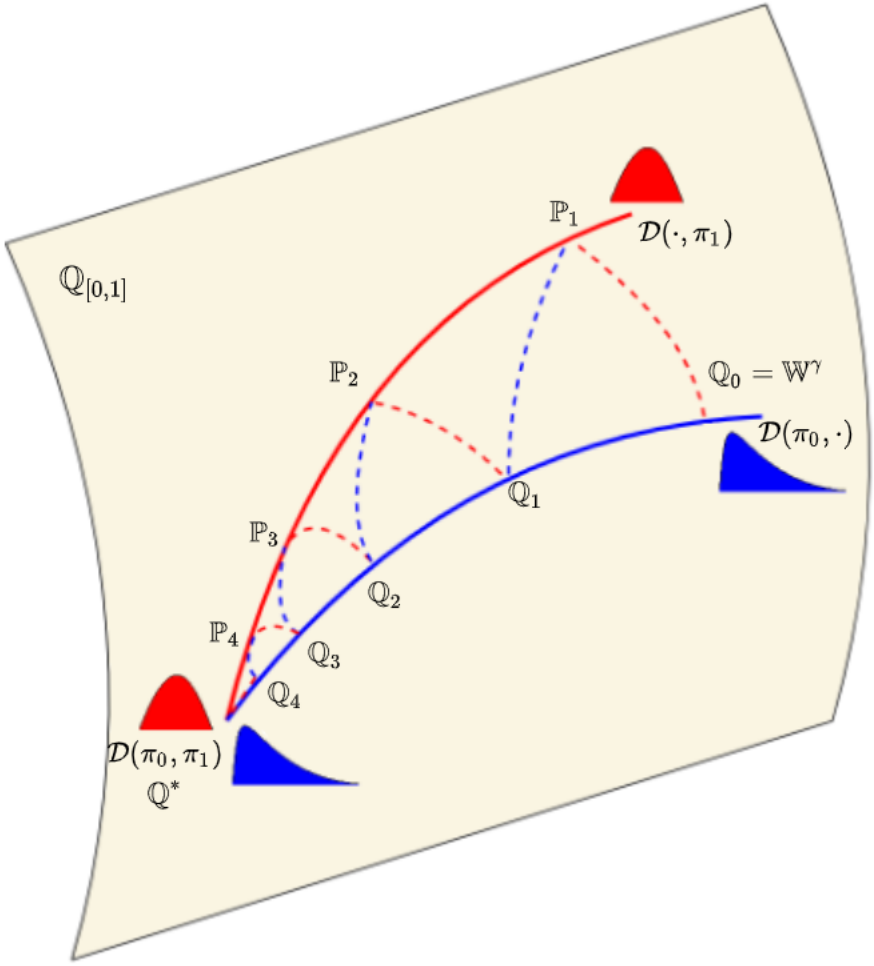

Figure 4.1: Illustration of the iterative proportional fitting procedure, inspired and adapted from Figure 1 in Bernton et al. (2019). The red line represents valid Itô-process posteriors with the terminal constraint $\mathbb{P} \in \mathcal{D}(\cdot, \pi_1)$, and the blue represents valid Itô-process posteriors with the initial constraint $\mathbb{Q} \in \mathcal{D}(\pi_0, \cdot)$. The illustration shows the alternation between the forward and backward steps, until the joint-bridge solution is reached where both constraints are met. Note the sheet represents the space of all valid Itô processes in the interval $[0, 1]$. By valid we mean Itô processes driven by the prior of the form $d\mathbf{x}(t) = \mathbf{b}_t + \gamma d\boldsymbol{\beta}(t)$.

"closed form" in terms of known quantities:

$$\mathbb{P}_i^* \left(A_{[0,1)} \times A_1 \right) = \int_{A_{[0,1)} \times A_1} \frac{d\pi_1}{dp_1^{\mathbb{Q}_{i-1}^*}} d\mathbb{Q}_{i-1}^*, \tag{4.12}$$

Algorithm 4: g-IPFP (Cramer, 2000)

input: $\pi_0(\boldsymbol{x}), \pi_1(\boldsymbol{y}), \mathbb{W}^\gamma$

Initialise:

$\mathbb{Q}_0^* = \mathbb{W}^\gamma$

$i = 0$

repeat

  $i := i + 1$

  $\mathbb{P}_i^* = \inf_{\mathbb{P} \in \mathcal{D}(\cdot, \pi_1)} D_{\mathrm{KL}}(\mathbb{P} || \mathbb{Q}_{i-1}^*)$

  $\mathbb{Q}_i^* = \inf_{\mathbb{Q} \in \mathcal{D}(\pi_0, \cdot)} D_{\mathrm{KL}}(\mathbb{Q} || \mathbb{P}_i^*)$

until *convergence*;

return $\mathbb{Q}_i^*, \mathbb{P}_i^*$

$$\mathbb{Q}_i^* \left(A_0 \times A_{(0,1]} \right) = \int_{A_0 \times A_{(0,1]}} \frac{d\pi_0}{dp_0^{\mathbb{P}_i^*}} d\mathbb{P}_i^*. \tag{4.13}$$

An important thing to note about the above solutions is that while these quantities are written in terms of components that we "know", such components are themselves not available in closed form and require some form of approximation scheme (e.g. Monte Carlo integration) in order to be computed.

Via the disintegration theorem, we can reduce g-IPFP to the standard IPFP, by noting that $\mathbb{Q}_i^*(\cdot|\boldsymbol{x}, \boldsymbol{y}) = \mathbb{P}_i^*(\cdot|\boldsymbol{x}, \boldsymbol{y}) = \mathbb{W}(\cdot|\boldsymbol{x}, \boldsymbol{y})$ remains invariant across iterations, thus we highlight that more than being an algorithm, g-IPFP serves as a framework for the development and design of algorithms aimed at solving the Schrödinger Bridge.

The proof from Ruschendorf et al. (1995) extends naturally to g-IPFP, as mentioned in Bernton et al. (2019). Furthermore, Bernton et al. (2019) provide additional results regarding the convergence rate of g-IPFP.

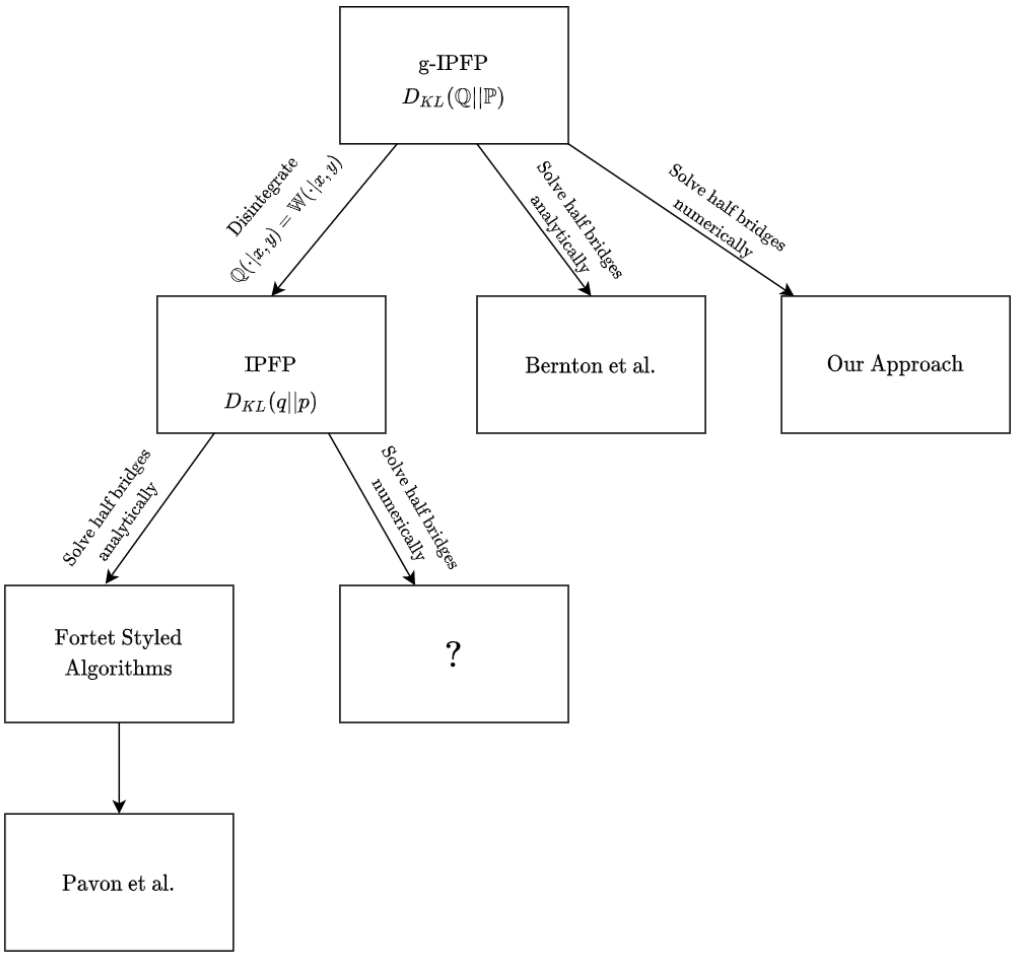

Figure 4.2: Genealogy of IPFP-based algorithms for solving the Schrödinger bridge problem. "?" symbolises an area without much prior research.

Chapter 5

Related Problems

We have presented the Schrödinger bridge, as well as an algorithmic framework for solving it. In this chapter, we will compare the Schrödinger bridge and IPFP to two different methodologies employed in machine learning for mapping between distributions.

5.1 Continuous-Time Stochastic Flows

Imagine that, as a probabilistic modeller, you were given the problem of mapping from one distribution to another. One of the simplest ways of approaching such task is to specify a generative process of the form

$$\begin{aligned} \mathbf{x} &\sim \pi_0, \\ \mathbf{y} &= \mathcal{T}_\theta(\mathbf{x}), \end{aligned}$$

where $\mathcal{T}_\theta(\mathbf{x})$ is a stochastic mapping (i.e. $\mathcal{T}_\theta(\mathbf{x}) = f_\theta(\mathbf{x}) + \boldsymbol{\epsilon}$). Then, to learn the mapping $\mathcal{T}_\theta(\mathbf{x})$, we maximise the marginal likelihood for a set of observations $\{\boldsymbol{y}_i \sim \pi_1\}$:

$$\arg\max_\theta \prod_i p(\boldsymbol{y}_i) = \arg\max_\theta \prod_i \int p_\theta(\boldsymbol{y}_i|\boldsymbol{x}) d\pi_0(\boldsymbol{x}).$$

We can take a further step and model the relationship between the two distributions using an SDE:

$$\begin{aligned} \mathbf{x}(0) &\sim \pi_0, \\ d\mathbf{x}(t) &= \boldsymbol{b}_\theta(\mathbf{x}(t), t) + \sqrt{\gamma} d\boldsymbol{\beta}(t), \\ \mathbf{y} &= f_\theta(\mathbf{x}(1)). \end{aligned}$$

In other words: $\mathbf{y}$ is generated by the solutions to an SDE with initial distribution π_0. We can also interpret the above as the latent object living in path space (i.e. $C([0,1], \mathbb{R}^d)$), as we have no observations for the trajectories, but only observations of its initial and terminal distributions. Similarly to the latent-variable-model example, we aim to maximise an estimate of the marginal likelihood under this generative process. Variants of this approach are explored by (Lähdesmäki & Kaski; Tzen & Raginsky, 2019) as ways of having infinitely-deep neural networks and Gaussian processes.

It is clear that this approach is different to the Schrödinger bridge. Firstly, the generative process requires the observation of pairs $\boldsymbol{x}_i, \boldsymbol{y}_i$ in order to estimate the marginal likelihood. Secondly, the Schrödinger bridge is based on the principle of maximum entropy (Jaynes, 1957), whilst this generative approach is based on maximising a marginal likelihood (ML-II). However, they share the similarity of exploiting stochastic dynamics to relate source and target distributions. Additionally, both have similar notions of prior and posterior dynamics.

5.2 Domain Adaptation and Generative Adversarial Networks (GANs)

In this section, we will motivate and provide some formal arguments towards the following observation:

Observation 2. *Each half-bridge objective in the IPFP algorithm is equivalent to a GAN-like objective with a corresponding cycle-consistency term, up*

to a regularisation term.

5.2.1 Short Introduction to Domain Adaptation with GANs

The goal of a GAN (Goodfellow et al., 2014) is to fit a generative model of the form

$$\begin{aligned}\mathbf{x} &\sim \pi_0,\\ \mathbf{y} &= f_\theta(\mathbf{x}),\end{aligned}$$

in the setting where we can sample from $\pi_0(\boldsymbol{x})$ and $\pi_1(\boldsymbol{y})$ tractably. The above model induces likelihood $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ and marginal $p_\theta(\boldsymbol{y})$, however the marginal is never computed explicitly and, thus, sometimes fitting GANs is referred to as fitting implicit generative models (Mohamed & Lakshminarayanan, 2016).

Using our notation, the GAN-fitting procedure (Goodfellow et al., 2014) takes the following form:

- (Discriminator loss) Estimating the surrogate to later be fitted by model $p_\theta(\boldsymbol{y})$:

$$\beta^* = \arg\min_\beta \left(\alpha\mathbb{E}_{\pi_1(\boldsymbol{y})}[-\ln D_\beta(\boldsymbol{y})] + (1-\alpha)\mathbb{E}_{p_\theta(\boldsymbol{y})}[-\ln(1-D_\beta(\boldsymbol{y}))]\right).$$

- (Generative loss) Fitting model $p_\theta(\boldsymbol{y})$ on the estimated surrogate:

$$\theta^* = \arg\min_\theta -\mathbb{E}_{p_\theta(\boldsymbol{y})}\left[\ln(D_{\hat{\beta^*}}(\boldsymbol{y}))\right].$$

In practice, the generator step (Goodfellow et al., 2014) typically minimises $-\mathbb{E}_{p_\theta(\boldsymbol{y})}[\ln(D(\boldsymbol{y}))]$, which according to Goodfellow et al. (2014) is equivalent to maximising $\mathbb{E}_{p_\theta(\boldsymbol{y})}[\ln(1-D(\boldsymbol{y}))]$.

The task of domain adaptation in GANs is finding a mapping between two

distributions π_0, π_1 that have been observed empirically. This has applications in image and language translation (Zhu et al., 2017; Lample et al., 2017). Typically, in domain adaptation with GANs, the following two processes are fitted in parallel:

$$\begin{aligned} &\mathbf{x} \sim \pi_0, && \mathbf{y} \sim \pi_1, \\ &\mathbf{y} = f_\theta(\mathbf{x}), && \mathbf{x} = g_\phi(\mathbf{y}). \end{aligned}$$

An interesting variation of GANs, used for domain adaptation and known as Cycle-GAN, adds an extra term to the generative loss, called the cycle-consistency loss:

$$\mathbb{E}_{\boldsymbol{x}\sim\pi_0}\left[\left|\left|\boldsymbol{x} - g_\phi\left(f_\theta(\boldsymbol{x})\right)\right|\right|^2\right] + \mathbb{E}_{\boldsymbol{y}\sim\pi_1}\left[\left|\left|\boldsymbol{y} - f_\theta\left(g_\phi(\boldsymbol{y})\right)\right|\right|^2\right].$$

This additional autoencoder loss encourages the transformations from one dataset to another to be inverses of each other. In the next section, we will show how the cycle-consistency term arises naturally from the Schrödinger bridge problem.

5.2.2 Connection to IPFP

Starting from the static Schrödinger bridge:

$$\underset{q(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}(\pi_0,\pi_1)}{\arg\min} -\int q(\boldsymbol{x},\boldsymbol{y}) \ln p^{\mathbb{W}^\gamma}(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} + \int q(\boldsymbol{x},\boldsymbol{y}) \ln q(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y},$$

we consider the backward step in IPFP for the i-th iteration

$$\underset{p(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}(\pi_1)}{\arg\inf} -\int p(\boldsymbol{x},\boldsymbol{y}) \ln q^{i-1}(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} + \int p(\boldsymbol{x},\boldsymbol{y}) \ln p(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y}.$$

We can enforce this constraint using the product rule $p(\boldsymbol{x}, \boldsymbol{y}) = p_\phi(\boldsymbol{x}|\boldsymbol{y})\pi_1(\boldsymbol{y})$, and parametrise $p_\phi(\boldsymbol{x}|\boldsymbol{y})$ with a powerful estimator:

$$\arg\min_\phi -\int p_\phi(\boldsymbol{x}|\boldsymbol{y})\pi_1(\boldsymbol{y}) \ln q^{i-1}(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{x}d\boldsymbol{y} + \int p_\phi(\boldsymbol{x}|\boldsymbol{y})\pi_1(\boldsymbol{y}) \ln p_\phi(\boldsymbol{x}|\boldsymbol{y})\pi_1(y) d\boldsymbol{x}d\boldsymbol{y},$$

$$\arg\min_\phi - \int p_\phi(\boldsymbol{x}|\boldsymbol{y})\pi_1(\boldsymbol{y}) \ln q^{i-1}(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{x}d\boldsymbol{y} + \int p_\phi(\boldsymbol{x}|\boldsymbol{y})\pi_1(\boldsymbol{y}) \ln q_\phi(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x}d\boldsymbol{y}.$$

Following our selected parametrisation, when we pass the first iteration, the product rule yields $q^{i-1}(\boldsymbol{x}, \boldsymbol{y}) = q_\theta^{i-1}(\boldsymbol{y}|\boldsymbol{x})\pi_0(\boldsymbol{x})$, which results in

$$\arg\min_\theta - \int p_\phi(\boldsymbol{x}|\boldsymbol{y})\pi_1(\boldsymbol{y}) \ln q_\theta^{i-1}(\boldsymbol{x}|\boldsymbol{y}) dxdy - \int p_\phi(\boldsymbol{x}|\boldsymbol{y})\pi_1(\boldsymbol{y}) \ln \pi_0(\boldsymbol{x}) d\boldsymbol{x}d\boldsymbol{y} + \int p_\phi(\boldsymbol{x}|\boldsymbol{y})\pi_1(\boldsymbol{y}) \ln p_\phi(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x}d\boldsymbol{y}.$$

Simplifying further, we get

$$\arg\min_\phi -\mathbb{E}_{p_\phi(\boldsymbol{x}|\boldsymbol{y})\pi_1(\boldsymbol{y})} \left[\ln q_\theta^{i-1}(\boldsymbol{y}|\boldsymbol{x})\right] - \mathbb{E}_{p_\phi(\boldsymbol{x})} \left[\ln \pi_0(\boldsymbol{x})\right] - H\left(p_\phi(\boldsymbol{x}|\boldsymbol{y})\pi_1(\boldsymbol{y})\right).$$

Sampling from $p_\phi(\boldsymbol{x}|\boldsymbol{y})\pi_1(y)$ can be achieved via ancestral sampling and we can parametrise $p_\phi(\boldsymbol{x}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{x}|\mu_\phi(\boldsymbol{y}), \sigma_\phi^2(\boldsymbol{y}))$ following Kingma & Welling (2013) such that it is easy to sample $\boldsymbol{x}$ conditioned on $\boldsymbol{y}$. However, we have introduced some bias by parametrising $p_\phi(\boldsymbol{x}|\boldsymbol{y})$ with a class of distributions.

Backward step:

$$\arg\min_\phi -\mathbb{E}_{p_\phi(\boldsymbol{x}|\boldsymbol{y})\pi_1(\boldsymbol{y})} \left[\ln q_\theta^{i-1}(\boldsymbol{y}|\boldsymbol{x})\right] - \mathbb{E}_{p_\phi(\boldsymbol{x})} \left[\ln \pi_0(\boldsymbol{x})\right] - H\left(p_\phi(\boldsymbol{x}|\boldsymbol{y})\pi_1(\boldsymbol{y})\right).$$

Forward step:

$$\arg\min_\theta -\mathbb{E}_{q_\theta(\boldsymbol{y}|\boldsymbol{x})\pi_0(\boldsymbol{x})} \left[\ln p_\phi^i(\boldsymbol{x}|\boldsymbol{y})\right] - \mathbb{E}_{q_\theta(\boldsymbol{y})} \left[\ln \pi_1(\boldsymbol{y})\right] - H\left(q_\theta(\boldsymbol{y}|\boldsymbol{x})\pi_0(\boldsymbol{x})\right).$$

We note that the above objectives are very similar to the Cycle-GAN (Zhu et al., 2017) objective, which gives us a link to a wide variety of successful

empirical methods.

Observation 3. *Using our proposed parametrisation, we can see that the term*

$$\mathbb{E}_{q^i_\theta(\boldsymbol{y}|\boldsymbol{x})\pi_0(\boldsymbol{x})}\left[\ln p^i_\phi(\boldsymbol{x}|\boldsymbol{y})\right] \propto \mathbb{E}_{\pi_0(\boldsymbol{x})\mathcal{N}(\epsilon|\boldsymbol{0},\mathbb{I}_d)}\left[-\frac{1}{2\sigma_x^2}\Big|\Big|\boldsymbol{x}-\mu_\phi\left(\mu_\theta(\boldsymbol{x})+\sigma_y\boldsymbol{\epsilon}\right)\Big|\Big|^2\right],$$

matches the corresponding cycle-consistency loss terms in Cycle-GAN (Zhu et al., 2017) in the $\sigma_y \downarrow 0$ limit.

Proof. For simplicity, let's make the variance function independent of the input $\sigma^2_\theta(\boldsymbol{x}) = \sigma^2_x\mathbb{I}_d$:

$$\mathbb{E}_{q_\theta(\boldsymbol{y}|\boldsymbol{x})\pi_0(\boldsymbol{x})}\left[\ln p^i_\phi(\boldsymbol{x}|\boldsymbol{y})\right] \propto \mathbb{E}_{q_\theta(\boldsymbol{y}|\boldsymbol{x})\pi_0(\boldsymbol{x})}\left[-\frac{1}{2\sigma_x^2}||\boldsymbol{x}-\mu_\phi(\boldsymbol{y}))||^2\right].$$

Applying the reparametrisation trick (Kingma & Welling, 2013) to $p_\theta(\boldsymbol{y}|\boldsymbol{x})\pi_0(\boldsymbol{x})$, we get

$$\begin{aligned}&\boldsymbol{x}\sim\pi_0(\boldsymbol{x}),\\ &\boldsymbol{y}=\mu_\theta(\boldsymbol{x})+\sigma_y\boldsymbol{\epsilon}\quad \boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\mathbb{I}_d).\end{aligned}$$

We can rewrite the cross-term loss as

$$\mathbb{E}_{q_\theta(\boldsymbol{y}|\boldsymbol{x})\pi_0(\boldsymbol{x})}\left[\ln p^i_\phi(\boldsymbol{x}|\boldsymbol{y})\right] \propto \mathbb{E}_{\pi_0(\boldsymbol{x})\mathcal{N}(\epsilon|\boldsymbol{0},\mathbb{I}_d)}\left[-\frac{1}{2\sigma_x^2}\Big|\Big|\boldsymbol{x}-\mu_\phi\left(\mu_\theta(\boldsymbol{x})+\sigma_y\boldsymbol{\epsilon}\right)\Big|\Big|^2\right],$$

where $-\frac{1}{2\sigma_x^2}\Big|\Big|\boldsymbol{x}-\mu_\phi\left(\mu_\theta(\boldsymbol{x})+\sigma_y\boldsymbol{\epsilon}\right)\Big|\Big|^2$ takes the same autoencoder (AE) form as the cycle-consistency loss in Cycle-GAN, with some added noise (they are related asymptotically in the zero-noise limit, just like VAEs and AEs). □

For notational simplicity, we have treated the σ_i terms as constants in the above derivations. Adapting the above argument to their non-constant counterparts is simple and does not add much to Observation 3.

Now, we will focus on the connection between the remaining half bridge terms and the GAN generative loss:

Observation 4. *Up to optimisation constants and for the optimal discriminator D^* for a generator $q_\theta(\boldsymbol{y}|\boldsymbol{x})$ as defined in (Goodfellow et al., 2014; Mohamed & Lakshminarayanan, 2016), we have the following:*

$$-\mathbb{E}_{q_\theta(\boldsymbol{y})}\left[\ln \pi_1(\boldsymbol{y})\right] \propto -\mathbb{E}_{q_\theta(\boldsymbol{y})}\left[\ln(D^*(\boldsymbol{y}))\right] - \mathbb{E}_{q_\theta(\boldsymbol{y})}\left[\ln\left(q_\theta(\boldsymbol{y}) + \pi_1(\boldsymbol{y})\right)\right].$$

In short, we can express the cross-entropy between our parametrised model and the empirical distribution in terms of the optimal discriminator D^* and a further term (which is one of the terms in the Jensen-Shannon Divergence (JSD)).

Proof. Using the ratio expression from Mohamed & Lakshminarayanan (2016):

$$\begin{aligned}
\ln D^*(\boldsymbol{y}) &= \ln\left(\frac{\pi_1(\boldsymbol{y})p(c=1)}{p(\boldsymbol{y})}\right) \\
&= \ln\left(\frac{\pi_1(\boldsymbol{y})p(c=1)}{p(c=1)p(\boldsymbol{y}|c=1) + p(c=0)p(\boldsymbol{y}|c=0)}\right) \\
&= \ln\left(\frac{\pi_1(\boldsymbol{y})\alpha}{\alpha\pi_1(\boldsymbol{y}) + (1-\alpha)p_\theta(\boldsymbol{y})}\right) \\
&\propto \ln\left(\pi_1(\boldsymbol{y})\right) - \ln\left(\alpha\pi_1(\boldsymbol{y}) + (1-\alpha)p_\theta(\boldsymbol{y})\right).
\end{aligned}$$

Considering the case where $\alpha = \frac{1}{2}$,

$$\ln D^*(\boldsymbol{y}) \propto \ln\left(\pi_1(\boldsymbol{y})\right) - \ln\left(q_\theta(\boldsymbol{y}) + \pi_1(\boldsymbol{y})\right),$$

thus

$$\mathbb{E}_{q_\theta(\boldsymbol{y})}\left[\ln D^*(\boldsymbol{y})\right] \propto \mathbb{E}_{q_\theta(\boldsymbol{y})}\left[\ln\left(\pi_1(\boldsymbol{y})\right)\right] - \mathbb{E}_{q_\theta(\boldsymbol{y})}\left[\ln\left(q_\theta(\boldsymbol{y}) + \pi_1(\boldsymbol{y})\right)\right]. \tag{5.1}$$

□

The above observation holds true for an optimal classifier D^* that can per-

fectly distinguish samples from $\pi_\theta(\boldsymbol{y})$ and $\pi_1(\boldsymbol{y})$. Mohamed & Lakshminarayanan (2016) offer further motivations. Therefore, up to a term, IPFP trains two GANs until convergence, going forward and backward in turn.

We explore the terms that differ between our loss and the GAN loss. Removing constants, the terms are

$$\mathbb{E}_{q_\theta(\boldsymbol{y}|\boldsymbol{x})\pi_0(\boldsymbol{x})}\left[\ln q_\theta(\boldsymbol{y}|\boldsymbol{x})\right] \quad \text{vs} \quad \mathbb{E}_{q_\theta(\boldsymbol{y})}\left[\ln\left(q_\theta(\boldsymbol{y}) + \pi_1(\boldsymbol{y})\right)\right]. \tag{5.2}$$

The term $H_\theta(\boldsymbol{y}|\boldsymbol{x}) = \mathbb{E}_{q_\theta(\boldsymbol{y}|\boldsymbol{x})\pi_0(\boldsymbol{x})}\left[\ln q_\theta(\boldsymbol{y}|\boldsymbol{x})\right]$ is a conditional entropy, whilst the term $\mathbb{E}_{q_\theta(\boldsymbol{y})}\left[\ln\left(q_\theta(\boldsymbol{y}) + \pi_1(\boldsymbol{y})\right)\right]$ is one of the terms in the Jensen-Shannon divergence. This is the main difference between GANs and a single forward (or backward) pass of IPFP on the Schrödinger bridge problem, but both prevent the delta collapse of the reverse cross-entropy.

Note than under this parametrisation, the term $H_\theta(\boldsymbol{y}|\boldsymbol{x})$ can be obtained in closed form ($\mathbb{E}_{\pi_0(\boldsymbol{x})}[\ln \sigma_\theta(\boldsymbol{x})]$), and has the clear interpretation of maximising the variance/spread of our mapping.

The above results raise an interesting question: is it possible to train GANs for domain adaptation by alternating half-bridge-styled GAN iterations until convergence as some form of approximate IPFP? The additional term difference from the half bridges removes the theoretical support from this proposed heuristic. Nonetheless, there might still be a deeper relationship that we may have missed between the two.

5.3 Summary

In this chapter we have:

- Highlighted the differences between the Schrödinger bridge problem and learning a stochastic flow between two distributions via maximum likelihood,

- Shown the similarities between GANs for domain adaptation and solving a single half bridge sub-iteration in IPFP.

Chapter 6

Empirical Schrödinger Bridges

Having introduced, motivated and discussed the Schrödinger bridge problem, we will move onto the core section of this thesis, where we propose and discuss numerical implementations of it. We call the empirical Schrödinger bridge the setting of the Schrödinger bridge problem where the marginal/boundary distributions are only available via samples (i.e. their empirical distribution):

$$\hat{\pi}_0(\boldsymbol{x}) = \frac{1}{N}\sum_i \delta(\boldsymbol{x} - \boldsymbol{x}_i), \quad \hat{\pi}_1(\boldsymbol{y}) = \frac{1}{M}\sum_j \delta(\boldsymbol{y} - \boldsymbol{y}_j),$$
$$\boldsymbol{x}_i \sim \pi_0, \quad \boldsymbol{y}_j \sim \pi_1. \tag{6.1}$$

Note that Pavon et al. (2018) refer to this problem as the data-driven Schrödinger bridge. We will discuss different algorithms derived from the g-IPFP framework for solving the Schrödinger bridge in this setting. As this is a fairly new problem of interest in the applied/numerical setting, there is not much accessible and validated prior work available. We will only comment on the work of Pavon et al. (2018) and omit a discussion of Bernton et al. (2019), we reserve this for future work.

Our core contributions in this chapter are:

- The rephrasing of the method by Pavon et al. (2018) as an unnor-

malised likelihood estimation problem, which allows a connection to alternative methods within the machine-learning and applied-statistics communities.

- The proposal of two novel numerical algorithms for solving the empirical Schrödinger bridge problem.

6.1 Maximum Likelihood Approach (Pavon et al., 2018)

The approach proposed in Pavon et al. (2018) is an adaptation of Fortet's algorithm (Algorithm 2) to the empirical setting, and is based on the following unstated observation:

Observation 5. *Steps 3 and 5 of Algorithm 2:*

$$\hat{\phi}_0^{(i)}(\boldsymbol{x}) := \frac{\pi_0(\boldsymbol{x})}{\phi_0^{(i)}(\boldsymbol{x})}, \tag{6.2}$$

$$\phi_1^{(i)}(\boldsymbol{y}) := \frac{\pi_1(\boldsymbol{y})}{\hat{\phi}_1^{(i)}(\boldsymbol{y})} \tag{6.3}$$

can equivalently be stated as a free-form, cross-entropy minimisation problem over the space of probability distributions $\mathcal{H}$:

$$\hat{\phi}_0^{(i)}(\boldsymbol{x}) = \underset{\hat{\phi}_0(\boldsymbol{x})\in\mathcal{H}}{\arg\sup}\, \mathbb{E}_{\pi_0(\boldsymbol{x})}\left[\ln \hat{\phi}_0(\boldsymbol{x})\phi_0^{(i)}(\boldsymbol{x})\right], \tag{6.4}$$

$$\phi_1^{(i)}(\boldsymbol{y}) = \underset{\phi_1(\boldsymbol{y})\in\mathcal{H}}{\arg\sup}\, \mathbb{E}_{\pi_1(\boldsymbol{y})}\left[\ln \phi_1(\boldsymbol{y})\hat{\phi}_1^{(i)}(\boldsymbol{y})\right]. \tag{6.5}$$

Proof. Equation 6.2 implies the following:

$$D_{\mathrm{KL}}\left(\pi_0(\boldsymbol{x})||\hat{\phi}_0^{(i)}(\boldsymbol{x})\phi_0^{(i)}(\boldsymbol{x})\right) = 0, \tag{6.6}$$

which can be restated as

$$\hat{\phi}_0^{(i)}(\boldsymbol{x}) = \underset{\hat{\phi}_0(\boldsymbol{x})\in\mathcal{H}}{\arg\inf}\, D_{\mathrm{KL}}\left(\pi_0(\boldsymbol{x})||\hat{\phi}_0(\boldsymbol{x})\phi_0^{(i)}(\boldsymbol{x})\right). \tag{6.7}$$

By removing the constant-entropy term, we arrive at

$$\hat{\phi}_0^{(i)}(\boldsymbol{x}) = \underset{\hat{\phi}_0(\boldsymbol{x})\in\mathcal{H}}{\arg\inf} -\mathbb{E}_{\pi_0(\boldsymbol{x})}\left[\ln \hat{\phi}_0(\boldsymbol{x})\phi_0^{(i)}(\boldsymbol{x})\right].$$

A similar argument follows for Equation 6.3. □

Note that Equation 6.6 motivates the use of other probability metrics/divergences, provided they share the same property of being minimised when their arguments are equal. Following Observation 5, the essence of the method in Pavon et al. (2018) is parametrising $\hat{\phi}_0(\boldsymbol{x})$, $\phi_1(\boldsymbol{y})$ with a parametric family of positive functions and minimising an empirical estimate of Equation 6.4:

$$\begin{aligned}\hat{\beta}_i^* = \arg\max_{\hat{\beta}} \frac{1}{M}\sum_s \ln \hat{\phi}_0(\boldsymbol{x}_s;\hat{\beta})\phi_0^{(i)}(\boldsymbol{x}_s), \quad \boldsymbol{x}_s \sim \pi_0(\boldsymbol{x}),\\ s.t. \int \hat{\phi}_0(\boldsymbol{x};\hat{\beta})\phi_0^{(i)}(\boldsymbol{x})d\boldsymbol{x} = 1,\end{aligned} \tag{6.8}$$

and

$$\begin{aligned}\beta_i^* = \arg\max_{\beta} \frac{1}{N}\sum_s \ln \phi_1(\boldsymbol{y}_s;\beta)\hat{\phi}_1^{(i)}(\boldsymbol{y}_s), \quad \boldsymbol{y}_s \sim \pi_1(\boldsymbol{y}),\\ s.t. \int \phi_1(\boldsymbol{y};\beta)\hat{\phi}_1^{(i)}(\boldsymbol{y})d\boldsymbol{y} = 1.\end{aligned} \tag{6.9}$$

Furthermore, via the method of Lagrange multipliers, Pavon et al. (2018) show that the above two objectives can be reduced to

$$\hat{\beta}_i^* = \arg\max_{\hat{\beta}} \frac{1}{M}\sum_s \ln \hat{\phi}_0(\boldsymbol{x}_s;\hat{\beta})\phi_0^{(i)}(\boldsymbol{x}_s) - \int \hat{\phi}_0(\boldsymbol{x};\hat{\beta})\phi_0^{(i)}(\boldsymbol{x})d\boldsymbol{x}, \quad \boldsymbol{x}_s \sim \pi_0(\boldsymbol{x}),$$

and

$$\beta_i^* = \arg\max_{\beta} \frac{1}{N} \sum_s \ln \phi_1(\boldsymbol{y}_s;\beta)\hat{\phi}_1^{(i)}(\boldsymbol{y}_s) - \int \phi_1(\boldsymbol{y};\beta)\hat{\phi}_1^{(i)}(\boldsymbol{y})d\boldsymbol{y}, \quad \boldsymbol{y}_s \sim \pi_1(\boldsymbol{y}).$$

Now, all that is left to numerically carry out this approximate variant of Fortet's algorithm is to estimate the propagation terms $\phi_0^{(i)}(\boldsymbol{x}_s), \hat{\phi}_1^{(i)}(\boldsymbol{y}_s)$ and compute the constraint terms $\int \phi_1(\boldsymbol{y};\beta)\hat{\phi}_1^{(i)}(\boldsymbol{y})d\boldsymbol{y}$, $\int \hat{\phi}_0(\boldsymbol{x};\hat{\beta})\phi_0^{(i)}(\boldsymbol{x})d\boldsymbol{x}$, which depend on a non-trivial multidimensional integral.

6.1.1 Importance Sampling Approach by Pavon et al. (2018)

Estimating the terms $\phi_0^{(i)}(\boldsymbol{x}_s), \hat{\phi}_1^{(i)}(\boldsymbol{y}_s)$ can be done through sampling the prior, either directly, in the case of Brownian motion, or via the EM discretisation for more complex Itô-process priors. Pavon et al. (2018) only estimate the term $\hat{\phi}_0^{(i)}(\boldsymbol{x}_s)$:

$$\begin{aligned} \phi_0^{(i)}(\boldsymbol{x}_s) &= \int P(\boldsymbol{y}|\boldsymbol{x}_s)\phi_1^{(i)}(\boldsymbol{y};\beta)d\boldsymbol{y} \\ &\approx \frac{1}{\tilde{M}} \sum_l \phi_1^{(i)}(\tilde{\boldsymbol{y}}_{ls};\beta) \quad \tilde{\boldsymbol{y}}_{ls} \sim P(\boldsymbol{y}|\boldsymbol{x}_s), \end{aligned} \tag{6.10}$$

as we require only one of the propagations in order to compute the normalising constraints $\int \phi_1(\boldsymbol{y};\beta)\hat{\phi}_1^{(i)}(\boldsymbol{y})d\boldsymbol{y}$, $\int \hat{\phi}_0(\boldsymbol{x};\hat{\beta})\phi_0^{(i)}(\boldsymbol{x})d\boldsymbol{x}$, since both integrals are equal to each other.

Pavon et al. (2018) note that the integrals arising from the constraint can be

decomposed in the following convenient order:

$$\int \hat{\phi}_0(\boldsymbol{x};\hat{\beta})\phi_0^{(i)}(\boldsymbol{x})d\boldsymbol{x} = \\ = \int \hat{\phi}_0(\boldsymbol{x};\hat{\beta}) \int P(\boldsymbol{y}|\boldsymbol{x}_s)\phi_1^{(i)}(\boldsymbol{x};\beta)d\boldsymbol{y}d\boldsymbol{x}, \tag{6.11}$$

$$\int \phi_1(\boldsymbol{y};\beta)\hat{\phi}_1^{(i)}(\boldsymbol{y})d\boldsymbol{y} = \\ = \int \hat{\phi}_0(\boldsymbol{x};\hat{\beta}) \int P(\boldsymbol{y}|\boldsymbol{x}_s)\phi_1^{(i)}(\boldsymbol{x};\beta)d\boldsymbol{y}d\boldsymbol{x}. \tag{6.12}$$

Now, all that is required is estimating the outer integrals with respect to $d\boldsymbol{x}$, as we have already provided a Monte Carlo estimate for the $d\boldsymbol{y}$ integral in Equation 6.10. In order to estimate the outer integrals, Pavon et al. (2018) propose to use importance sampling:

$$\int \hat{\phi}_0(\boldsymbol{x};\hat{\beta}) \int P(\boldsymbol{y}|\boldsymbol{x}_s)\phi_1^{(i)}(\boldsymbol{x};\beta)d\boldsymbol{y}d\boldsymbol{x} = \\ = \int \tilde{\pi}_0(\boldsymbol{x}) \frac{\hat{\phi}_0(\boldsymbol{x};\hat{\beta}) \int P(\boldsymbol{y}|\boldsymbol{x}_s)\phi_1^{(i)}(\boldsymbol{x};\beta)d\boldsymbol{y}}{\tilde{\pi}_0(\boldsymbol{x})} d\boldsymbol{x} \\ \approx \frac{1}{\tilde{M}\tilde{N}} \sum_{lk} \frac{\hat{\phi}_0(\boldsymbol{x}_k;\hat{\beta})\phi_1^{(i)}(\tilde{\boldsymbol{y}}_{lk};\beta)}{\tilde{\pi}_0(\boldsymbol{x}_k)}, \quad \boldsymbol{y}_{lk} \sim P(\boldsymbol{y}|\boldsymbol{x}_k),\ \boldsymbol{x}_k \sim \tilde{\pi}_0(\boldsymbol{x}), \tag{6.13}$$

where the authors of Pavon et al. (2018) recommend setting $\tilde{\pi}_0(\boldsymbol{x})$ to a density estimator of $\pi_0(\boldsymbol{x})$, since the integrand $\hat{\phi}_0(\boldsymbol{x};\hat{\beta})\phi_0^{(i)}(\boldsymbol{x})$ will converge to a density estimator for $\pi_0(\boldsymbol{x})$ as we iterate Fortet's algorithm. However, in the early iterations of Fortet's algorithm there is no particular advantage in using this heuristic to initialise the importance sampler. Furthermore, this naive importance sampling scheme does not focus on sampling in areas where the integrand has mass (i.e. near its modes for example). Finally, due to the curse of dimensionality, vanilla importance samplers will not scale to high dimensions.

6.1.2 Alternative Formulation as an Unnormalised Likelihood Problem

Rather than enforcing the normalization constraint via a method of Lagrange multipliers, we can enforce it via the following reparametrisation:

$$\hat{\beta}_i^* = \arg\max_{\hat{\beta}} \frac{1}{M} \sum_s \ln \frac{\hat{\phi}_0(\boldsymbol{x}_s; \hat{\beta}) \phi_0^{(i)}(\boldsymbol{x}_s)}{\int \hat{\phi}_0(\boldsymbol{x}; \hat{\beta}) \phi_0^{(i)}(\boldsymbol{x}) d\boldsymbol{x}}, \quad \boldsymbol{x}_s \sim \pi_0(\boldsymbol{x}),$$

and

$$\beta_i^* = \arg\max_{\beta} \frac{1}{N} \sum_s \ln \frac{\phi_1(\boldsymbol{y}_s; \beta) \hat{\phi}_1^{(i)}(\boldsymbol{y}_s)}{\int \phi_1(\boldsymbol{y}; \beta) \hat{\phi}_1^{(i)}(\boldsymbol{y}) d\boldsymbol{y}}, \quad \boldsymbol{y}_s \sim \pi_1(\boldsymbol{y}),$$

turning the problem into estimating and fitting an unnormalised likelihood (Gutmann & Hyvärinen, 2010). This formulation opens the door to estimation techniques alternate to maximum likelihood, such as noise-contrastive estimation (Gutmann & Hyvärinen, 2010).

A natural follow-up arising from this setup is to parametrise the potentials with (mixtures of) Gaussian distributions. This parametrisation would lead to closed-form expressions for the partition function and, intuitively, mixtures of Gaussians seem flexible enough to fit arbitrary distributions. However, we found that these parametrisations are not flexible enough to fit most distributions. This can be explained by looking at how Fortet's algorithm requires one potential to be the reciprocal of the other times the marginal, which in many cases can lead to having one potential being log-concave, whilst the other is log-convex. This does not hold true for Gaussian potentials. Further details are found in Appendix A.

6.2 Direct Half-Bridge-Drift Estimation with Gaussian Processes

We now present a novel approach to approximately solving the empirical Schrödinger bridge problem by exploiting the "closed-form" expressions of the half bridge problem. Thus, rather than parametrising the measures in the half bridge and solving the optimisation numerically, we seek to directly approximate the measure that extremises the half bridge objective. We do this by using Gaussian processes (Williams & Rasmussen, 2006) to estimate the drift of the trajectories sampled from the optimal half bridge measure.

We start from the following observation, which, in short, tells us how to sample from the optimal half bridge distribution:

Observation 6. *We can parametrise a measure $\mathbb{Q}$ with its drift as the solution to the following SDEs:*

$$d\mathbf{x}^{\pm}(t) = \mathbf{b}^{\pm}(t) + \sqrt{\gamma} d\boldsymbol{\beta}^{\pm}(t),$$
$$\mathbf{x}(0) \sim p_0^{\mathbb{Q}},\ \mathbf{x}(1) \sim p_1^{\mathbb{Q}}.$$

Then, we can sample from the solution to the following half bridges:

$$\mathbb{P}^{*-} = \underset{\mathbb{P} \in \mathcal{D}(\cdot, \pi_1)}{\arg\inf}\ D_{\mathrm{KL}}(\mathbb{P}||\mathbb{Q}),$$
$$\mathbb{P}^{*+} = \underset{\mathbb{P} \in \mathcal{D}(\pi_0, \cdot)}{\arg\inf}\ D_{\mathrm{KL}}(\mathbb{P}||\mathbb{Q}),$$

via simulating trajectories (e.g. using the EM method) following the SDEs

$$d\mathbf{x}^{-}(t) = \mathbf{b}^{-}(t) + \sqrt{\gamma} d\boldsymbol{\beta}^{-}(t), \quad \mathbf{x}(1) \sim \pi_1,$$
$$d\mathbf{x}^{+}(t) = \mathbf{b}^{+}(t) + \sqrt{\gamma} d\boldsymbol{\beta}^{+}(t), \quad \mathbf{x}(0) \sim \pi_0.$$

Paths sampled from the above SDEs will be distributed according to $\mathbb{P}^{*-}$ and $\mathbb{P}^{*+}$ respectively.

Proof. (Sketch) W.l.o.g., the disintegration $\mathbb{Q}(\cdot|\mathbf{x}(0))$ follows the dynamics

$d\mathbf{x}^+(t) = \mathbf{b}^+(t) + \sqrt{\gamma} d\boldsymbol{\beta}^+(t)$. Since the optimal half bridge's disintegration is given by $\mathbb{P}^+(\cdot|\mathbf{x}(0)) = \mathbb{Q}(\cdot|\mathbf{x}(0))$, its dynamics also follow $d\mathbf{x}^+(t) = \mathbf{b}^+(t) + \sqrt{\gamma} d\boldsymbol{\beta}^+(t)$. What is left is to attach the constraint via $\mathbf{x}(0) \sim \pi_0(\mathbf{x}(0))$.

This is simply enforcing the constraints via the product rule (Disintegration Theorem) and then matching the remainder of the unconstrained interval with the disintegration for the reference distribution $\mathbb{Q}$, following Theorems 7 and 8. □

Intuitively, we are performing a cut-and-paste-styled operation by cutting the dynamics of the shortened (unconstrained) time interval and pasting the constraint to it at the corresponding boundary.

6.2.1 Fitting the Drift from Samples

Once we have sampled trajectories from the half bridge solutions

$$\left\{ \left(\boldsymbol{x}_n^+(t_k) \right)_{k=0}^T \right\}_{n=0}^N, \quad \left\{ \left(\boldsymbol{x}_m^-(t_l) \right)_{l=0}^T \right\}_{m=0}^M, \tag{6.14}$$

we apply the methodology from (Ruttor et al., 2013; Batz et al., 2018), in order to infer the optimal half bridge drifts. Note that the method by Ruttor et al. (2013) is specified for inferring the drift from a single sampled trajectory. We adapt this method to work for multiple sampled trajectories.

Using the Euler-Mayurama discretisations, we know that the discretised transition probabilities follow a Gaussian distribution.

For the backward samples, we wish to infer the forward drift:

$$p\left(\left(\boldsymbol{x}_n^+(t_k) \right)_{k=0}^T \middle| \boldsymbol{b}_{\mathbb{P}_i}^+(\cdot,\cdot) \right) \propto$$
$$\prod_{k=0}^{T} \exp\left(-\frac{1}{2\Delta t} || \boldsymbol{x}_n^+(t_k) - \boldsymbol{x}_n^+(t_k - \Delta t) - \Delta t \boldsymbol{b}_{\mathbb{P}_i}^+ \left(\boldsymbol{x}_n^-(t_l - \Delta t), t_k - \Delta t \right) ||^2 \right),$$

and, conversely for forward samples, we wish to infer the backward drift:

$$p\left(\left(\boldsymbol{x}_m^-(t_l)\right)_{l=0}^T \middle| \boldsymbol{b}_{\mathbb{Q}_i}^-(\cdot,\cdot)\right) \propto$$
$$\prod_{l=0}^{T} \exp\left(-\frac{1}{2\Delta t}||\boldsymbol{x}_m^-(t_l-\Delta t) - \boldsymbol{x}_m^-(t_l) + \Delta t \boldsymbol{b}_{\mathbb{Q}_i}^-\left(\boldsymbol{x}_m^-(t_l-\Delta t), t_l-\Delta t\right)||^2\right).$$

Placing Gaussian-process priors on the drift functions $\boldsymbol{b}_{\mathbb{Q}_i}^- \sim \mathcal{GP}$ and $\boldsymbol{b}_{\mathbb{P}_i}^+ \sim \mathcal{GP}$, we arrive at the following multioutput GP regression problems:

$$\frac{\boldsymbol{x}_n^-(t_k) - \boldsymbol{x}_n^-(t_k-\Delta t)}{\Delta t} = \boldsymbol{b}_{\mathbb{P}_i}^+\left(\boldsymbol{x}_n^-(t_k-\Delta t), t_k-\Delta t\right) + \frac{\boldsymbol{\gamma}^{1/2}}{\Delta t}\epsilon,$$
$$\frac{\boldsymbol{x}_m^-(t_l-\Delta t) - \boldsymbol{x}_m^-(t_l)}{\Delta t} = -\boldsymbol{b}_{\mathbb{Q}_i}^-\left(\boldsymbol{x}_m^-(t_l-\Delta t), t_l-\Delta t\right) + \frac{\boldsymbol{\gamma}^{1/2}}{\Delta t}\epsilon,$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_d)$. Following (Ruttor et al., 2013; Batz et al., 2018), we assume the dimensions of the drift function are independent (equivalent to imposing a block-diagonal kernel matrix on a multi-output GP) and thus we fit a separate GP for each dimension, yielding the following predictive mean per dimension d of the drift:

$$[\bar{\boldsymbol{b}}_{\mathbb{P}_i}^+(\boldsymbol{x},t)]_d = \mathrm{vec}\big(\boldsymbol{k}_d^+(\boldsymbol{x}\oplus t)\big)^\top \left(\tilde{\boldsymbol{K}}_d^+ + \frac{\gamma}{\Delta t}\mathbb{I}_{MT}\right)^{-1}\mathrm{vec}(\boldsymbol{Y}_d^+), \tag{6.15}$$
$$[\boldsymbol{b}_{\mathbb{Q}_i}^-(\boldsymbol{x},t)]_d = \mathrm{vec}\big(\boldsymbol{k}_d^-(\boldsymbol{x}\oplus t)\big)^\top \left(\tilde{\boldsymbol{K}}_d^- + \frac{\gamma}{\Delta t}\mathbb{I}_{NT}\right)^{-1}\mathrm{vec}(\boldsymbol{Y}_d^-), \tag{6.16}$$

where

$$[\tilde{\boldsymbol{K}}_d^+]_{m\cdot T+l,m'\cdot T+l'} = [\tilde{\boldsymbol{K}}_d^+]_{mm'll'},\ \ [\boldsymbol{K}_d^+]_{n\cdot T+k,n'\cdot T+k'} = [\boldsymbol{K}_d^+]_{nn'kk'},$$
$$[\boldsymbol{K}_d^+]_{mm'll'} = K_d\big(\boldsymbol{x}_m^-(t_l-\Delta t)\oplus(t_l-\Delta t), \boldsymbol{x}_{m'}^-(t_{l'}-\Delta t)\oplus(t_{l'}-\Delta t)\big),$$
$$[\boldsymbol{K}_d^-]_{nn'kk'} = K_d\big(\boldsymbol{x}_n^+(t_k)\oplus t_k, \boldsymbol{x}_{n'}^+(t_{k'})\oplus t_{k'}\big),$$
$$[\boldsymbol{k}_d^+(\boldsymbol{x}\oplus t)]_{lm} = K_d(\boldsymbol{x}\oplus t, \boldsymbol{x}_m^-(t_l-\Delta t)\oplus(t_l-\Delta t)),$$
$$[\boldsymbol{k}_d^-(\boldsymbol{x}\oplus t)]_{kn} = K_d(\boldsymbol{x}\oplus t, \boldsymbol{x}_n^-(t_k)\oplus(t_k)),$$
$$[\boldsymbol{Y}_d^+]_{lm} = \left[\frac{\boldsymbol{x}_m^-(t_l)-\boldsymbol{x}_m^-(t_l-\Delta t)}{\Delta t}\right]_d, \quad [\boldsymbol{Y}_d^-]_{kn} = \left[\frac{\boldsymbol{x}_n^+(t_k-\Delta t)-\boldsymbol{x}_n^+(t_k)}{\Delta t}\right]_d,$$

$\oplus$ is the concatenation operator, vec is the standard vectorisation operator,

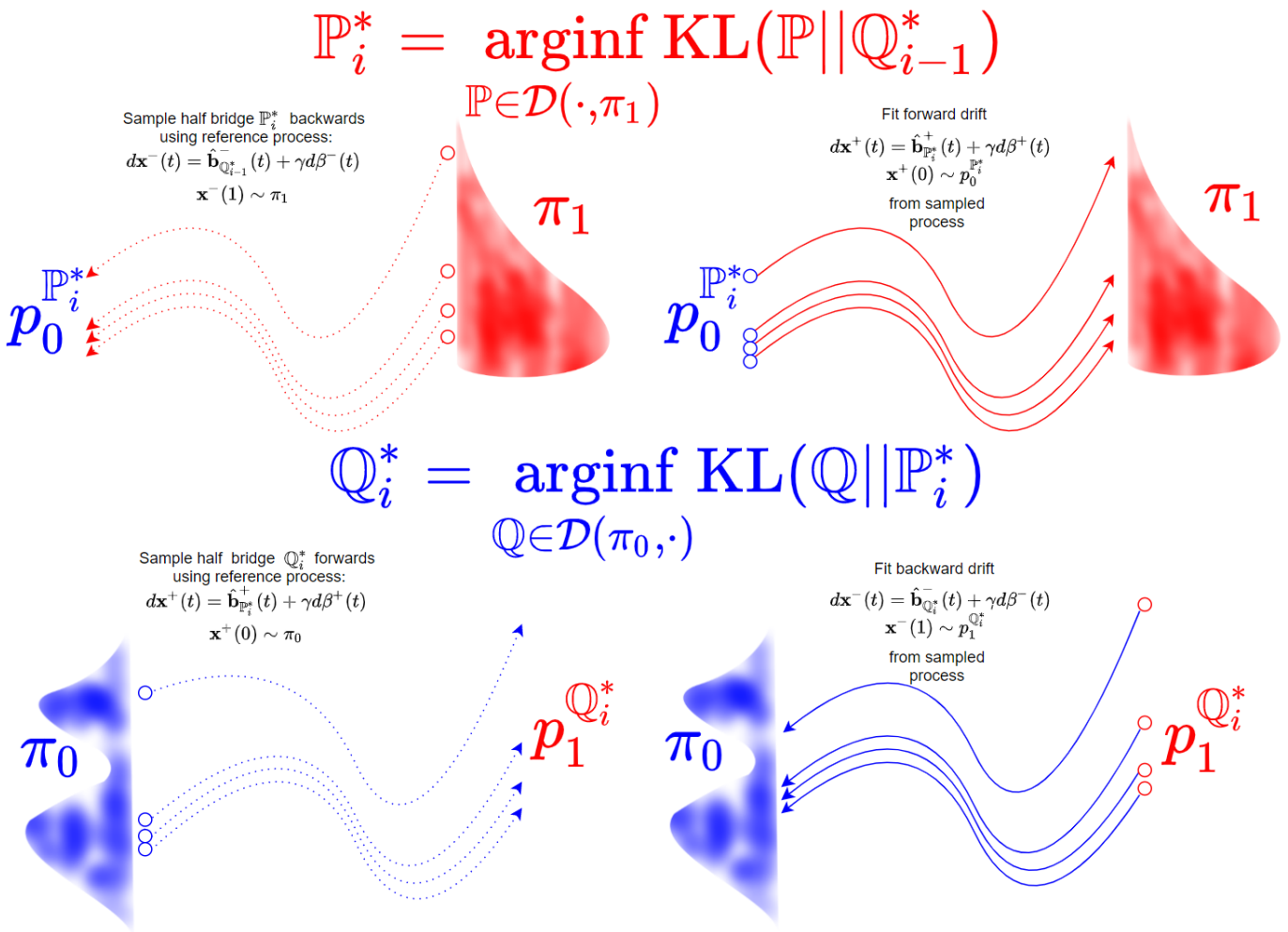

Figure 6.1: Iteration i for the drift-based approximation to g-IPFP. In each iteration, we draw samples from the process in the direction that incorporates the constraint as an initial value, and we learn the drift which simulates the same path measure in the opposite direction to the sampling. With each iteration $\mathbf{p_0^{\mathbb{P}_i^*}}$ and $\mathbf{p_1^{\mathbb{Q}_i^*}}$ get closer to π_0 and π_1 respectively.

and $K_d : \mathbb{R}^{d+1} \times \mathbb{R}^{d+1} \to \mathbb{R}$ is a valid kernel function. In practice, we are not making use of the predictive variances[1]. Instead, we simply use the predictive mean as an estimate of the drift and subsequently use that estimate to perform the EM method.

We now have the relevant ingredients to carry out this flavour of approximate g-IPFP specified in Algorithm 5, where the SDESolve$\Big(-\bar{\boldsymbol{b}}^-_{\mathbb{Q}_{i-1}}, \gamma, \pi_1, \Delta t, M\Big)$ routine generates M trajectories using the Euler-Mayurama method.

Note that we can use Equations 3.7 and 3.10, which allow us to easily compute an estimate of the Schrödinger bridge using the current estimates of the optimal drifts.

Algorithm 5: Approximate g-IPFP with Gaussian processes (GP)

input: $\pi_0(\boldsymbol{x}), \pi_1(\boldsymbol{y}), \mathbb{W}^\gamma$

1 Initialise:
2 $i := 0$
3 $\mathbb{Q}^*_0 := \mathbb{W}^\gamma$
4 Obtain or estimate backward drift of prior:
5 $\bar{\boldsymbol{b}}^-_{\mathbb{Q}_0}(\boldsymbol{x}, t) :=$ ObtainBackwardDrift$(\mathbb{Q}^*_0)$
6 **repeat**
7 $\quad i := i + 1$
8 $\quad \Big\{\big(\boldsymbol{x}^-_m(t_l)\big)^T_{l=0}\Big\}^M_{m=0} :=$ SDESolve$\Big(-\bar{\boldsymbol{b}}^-_{\mathbb{Q}_{i-1}}, \gamma, \pi_1, \Delta t, M\Big)$
9 $\quad \bar{\boldsymbol{b}}^+_{\mathbb{P}_i} :=$ GPDriftFit$\Big(\Big\{\big(\boldsymbol{x}^-_m(t_l)\big)^T_{l=0}\Big\}^M_{m=0}, \gamma,$ forward=True$\Big)$
10 $\quad \Big\{\big(\boldsymbol{x}^+_n(t_k)\big)^T_{k=0}\Big\}^N_{n=0} :=$ SDESolve$\Big(\bar{\boldsymbol{b}}^+_{\mathbb{P}_i}, \gamma, \pi_0, \Delta t, N\Big)$
11 $\quad \bar{\boldsymbol{b}}^-_{\mathbb{Q}_i} :=$ GPDriftFit$\Big(\Big\{\big(\boldsymbol{x}^+_n(t_k)\big)^T_{k=0}\Big\}^N_{n=0}, \gamma,$ forward=False$\Big)$
12 **until** *convergence*;
13 **return** $\bar{\boldsymbol{b}}^-_{\mathbb{Q}_i}, \bar{\boldsymbol{b}}^+_{\mathbb{P}_i}$

Sampling a Reverse-Time Diffusion

A quick note to the reader is that the sign multiplying the drift in line 8 of Algorithm 5 may seem arbitrary. However, we can motivate it by regarding

[1] We are effectively simply doing kernel ridge regression.

an Itô process as the continuous-time limit of discrete-time Markov chains.

We want to consider the case where both backward and forward transitions obey the same joint probability law, that is

$$p^+(\boldsymbol{x}_{t+\Delta t}|\boldsymbol{x}_t)p_t(\boldsymbol{x}_t) = p^-(\boldsymbol{x}_t|\boldsymbol{x}_{t+\Delta t})p_{t+\epsilon}(\boldsymbol{x}_{t+\Delta t}), \tag{6.17}$$

where $p_t(\cdot)$ is the marginal distribution at time t and $p^+(\cdot|\cdot)$, $p^-(\cdot|\cdot)$ respectively are Gaussians of the form

$$p^+(\boldsymbol{x}_{t+\Delta t}|\boldsymbol{x}_t) = \frac{1}{2\pi\Delta t}\exp\left(-\frac{1}{2\Delta t}\left[\boldsymbol{x}_{t+\Delta t} - (\boldsymbol{x}_t + \boldsymbol{b}^+(\boldsymbol{x}_t, t)\Delta t)\right]^2\right),$$
$$p^-(\boldsymbol{x}_t|\boldsymbol{x}_{t+\Delta t}) = \frac{1}{2\pi\Delta t}\exp\left(-\frac{1}{2\Delta t}\left[\boldsymbol{x}_t - \left(\boldsymbol{x}_{t+\Delta t} - \boldsymbol{b}^-(\boldsymbol{x}_{t+\Delta t}, t+\Delta t)\Delta t\right)\right]^2\right).$$

Taking logs on both sides of Equation 6.17 and preserving a linear order in Δt gives

$$(\boldsymbol{b}^+(\boldsymbol{x}_t, t) - \boldsymbol{b}^-(\boldsymbol{x}_{t+\Delta t}, t+\Delta t)(\boldsymbol{x}_{t+\Delta t} - \boldsymbol{x}_t) = \ln p_{t+\Delta t}(\boldsymbol{x}_{t+\Delta t}) - \ln p_t(\boldsymbol{x}_t),$$

which, in the continuous-time limit, arrives to our chosen flavour[2] of Nelson's duality formula (Nelson, 1967):

$$\boldsymbol{b}^+(\boldsymbol{x}_t, t) - \boldsymbol{b}^-(\boldsymbol{x}_t, t) = \nabla \ln p(\boldsymbol{x}_t, t).$$

6.3 Stochastic Control Approach

We would like to carry out a variant of g-IPFP that is based on the stochastic control formulations of the half bridge problems. Using a slight adaptation to Lemma 3, we can rewrite the half bridge steps of g-IPFP as:

[2]Different authors (e.g. Nagasawa (2012)) parametrise the duality formula with different signs, in turn changing the sign of the EM discretisation.

- Backward half bridge (time-reversed dynamics):

$$\mathbf{b}^{-}_{\mathbb{P}_i}(t) = \min_{\mathbf{b}^{-}\in\mathcal{B}} \mathbb{E}_{\mathbb{P}}\left[\int_0^1 \frac{1}{2\gamma}\left|\left|\mathbf{b}^{-}(t) - \mathbf{b}^{-}_{\mathbb{Q}_{i-1}}(t),\right|\right|^2 dt\right],$$
$$s.t. \quad d\mathbf{x}(t) = \mathbf{b}^{-}(t)dt + \sqrt{\gamma}d\boldsymbol{\beta}^{-}(t), \;\; \mathbf{x}(1) \sim \pi_1. \tag{6.18}$$

- Forward half bridge:

$$\mathbf{b}^{+}_{\mathbb{Q}_i}(t) = \min_{\mathbf{b}^{+}\in\mathcal{B}} \mathbb{E}_{\mathbb{Q}}\left[\int_0^1 \frac{1}{2\gamma}\left|\left|\mathbf{b}^{+}(t) - \mathbf{b}^{+}_{\mathbb{P}_i}(t)\right|\right|^2 dt\right],$$
$$s.t. \quad d\mathbf{x}(t) = \mathbf{b}^{+}(t)dt + \sqrt{\gamma}d\boldsymbol{\beta}^{+}(t), \;\; \mathbf{x}(0) \sim \pi_0. \tag{6.19}$$

.

The dual drifts can be obtained using Nelson's duality equations:

$$\mathbf{b}^{-}_{\mathbb{Q}_{i-1}}(t) = \mathbf{b}^{+}_{\mathbb{Q}_{i-1}}(t) - \gamma\nabla_{\boldsymbol{x}} p^{\mathbb{Q}_{i-1}}(\mathbf{x}(t), t),$$
$$\mathbf{b}^{+}_{\mathbb{P}_i}(t) = \mathbf{b}^{-}_{\mathbb{P}_i}(t) + \gamma\nabla_{\boldsymbol{x}} p^{\mathbb{P}_i}(\mathbf{x}(t), t).$$

We express the dynamical evolution of $\mathbb{P}$ in the backward half bridge as a time-reversed process (Pavon & Wakolbinger, 1991; Nelson, 1967), such that the terminal distribution constraint $\mathbf{x}(1) \sim \pi_1$ becomes an initial value problem in reversed time, which is something we know how to sample from.

In theory, we now have all the elements required to carry out g-IPFP under this control formulation. However, the dual drifts require the logarithmic gradient of the solution to their respective Fokker-Plank equations (i.e. $\nabla_{\boldsymbol{x}} p^{\mathbb{P}_i}(\mathbf{x}(t), t)$, $\nabla_{\boldsymbol{x}} p^{\mathbb{Q}_{i-1}}(\mathbf{x}(t), t)$). Estimating $p^{\mathbb{P}_i}(\mathbf{x}(t), t)$ or $p^{\mathbb{Q}_{i-1}}(\mathbf{x}(t), t)$ is very costly, since we will also need to evaluate such estimations for every point in the interval $[0, 1]$ that we use to estimate the integral $\int_0^1 \ldots dt$.

6.3.1 Forward and Backward Diffusions

As mentioned, Equations 6.19 and 6.18 exhibit the computational challenge of estimating $p^{\mathbb{X}}(\mathbf{x}(t), t)$. In order to overcome this, we derive alternate ex-

pressions for Equations 6.19 and 6.18 that do not involve FPK solution terms inside the time integral.

Proposition 1. *The forward half bridge (setting $\gamma = 1$ for simplicity):*

$$\mathbf{b}^+_{\mathbb{Q}_i}(t) = \min_{\mathbf{b}^-_{\mathbb{Q}} \in \mathcal{B}} \mathbb{E}_{\mathbb{Q}} \left[\int_0^1 \frac{1}{2} \left|\left| \mathbf{b}^+_{\mathbb{Q}}(t) - \mathbf{b}^+_{\mathbb{P}_i}(t) \right|\right|^2 dt \right],$$
$$s.t. \quad d\mathbf{x}(t) = \mathbf{b}^+_{\mathbb{Q}}(t)dt + \boldsymbol{\beta}^+(t), \;\; \mathbf{x}(0) \sim \pi_0$$

can be expressed as[3]*:*

$$\mathbf{b}^+_{\mathbb{Q}_i}(t) = \min_{\mathbf{b}^- \in \mathcal{B}} -2\mathbb{E}_{p_1^{\mathbb{Q}}} \left[\ln \pi_1\right] + \mathbb{E}_{\mathbb{Q}} \left[\int_0^1 \left(\frac{1}{2} ||\mathbf{b}^+_{\mathbb{Q}}(t) - \mathbf{b}^-_{\mathbb{P}_i}(t)||^2 - \nabla \cdot \mathbf{b}^-_{\mathbb{P}_i}(t) \right) \right],$$
$$s.t. \quad d\mathbf{x}(t) = \mathbf{b}^+_{\mathbb{Q}}(t)dt + \boldsymbol{\beta}^+(t), \;\; \mathbf{x}(0) \sim \pi_0. \tag{6.20}$$

Proof. We abbreviate terms of the form $p^{\mathbb{X}}(\mathbf{x}(t), t)$ with $p_t^{\mathbb{X}}$ and $\mathbf{b}^{\mathbb{X}}_{\pm}(t)$, using $\mathbf{b}^{\mathbb{X}}_{\pm}$ for compactness. We express the KL in terms of backward drifts (known result):

$$D_{\mathrm{KL}}(\mathbb{Q}||\mathbb{P}_i) = D_{\mathrm{KL}}(p_0^{\mathbb{Q}}||p_0^{\mathbb{P}_i}) + \frac{1}{2}\mathbb{E}_{\mathbb{Q}} \left[\int_0^1 ||\mathbf{b}^+_{\mathbb{Q}} - \mathbf{b}^+_{\mathbb{P}_i}||^2 dt \right]. \tag{6.21}$$

Using Nelson's duality formula $\mathbf{b}^{\mathbb{P}_i}_+(t) = \mathbf{b}^{\mathbb{P}_i}_-(t) + \nabla \ln p_t$:

$$D_{\mathrm{KL}}(\mathbb{Q}||\mathbb{P}_i) = D_{\mathrm{KL}}(p_0^{\mathbb{Q}}||p_0^{\mathbb{P}_i}) + \frac{1}{2}\mathbb{E}_{\mathbb{Q}} \left[\int_0^1 ||\mathbf{b}^+_{\mathbb{Q}} - \mathbf{b}^-_{\mathbb{P}_i} - \nabla \ln p_t^{\mathbb{P}_i}||^2 dt \right]. \tag{6.22}$$

Expanding:

$$D_{\mathrm{KL}}(\mathbb{Q}||\mathbb{P}_i) = D_{\mathrm{KL}}(p_0^{\mathbb{Q}}||p_0^{\mathbb{P}_i}) +$$
$$\frac{1}{2}\mathbb{E}_{\mathbb{Q}} \left[\int_0^1 \left(||\mathbf{b}^+_{\mathbb{Q}} - \mathbf{b}^-_{\mathbb{P}_i}||^2 - 2\nabla \ln p_t^{\mathbb{P}_i} \cdot (\mathbf{b}^+_{\mathbb{Q}} - \mathbf{b}^-_{\mathbb{P}_i}) + \nabla \ln {p_t^{\mathbb{P}_i}}^2 \right) dt \right].$$

[3]We thank Professor Austen Lamacraft for providing the first sketch of this proof, which we have reviewed in detail and iterated on.

Using the chain rule for the second derivative $\Delta \ln p = -\frac{(\nabla p)^2}{p^2} + \frac{\Delta p}{p}$:

$$D_{\mathrm{KL}}(\mathbb{Q}||\mathbb{P}_i) = D_{\mathrm{KL}}(p_0^{\mathbb{Q}}||p_0^{\mathbb{P}_i}) +$$
$$\frac{1}{2}\mathbb{E}_{\mathbb{Q}}\left[\int_0^1 \left(||\mathbf{b}_{\mathbb{Q}}^+ - \mathbf{b}_{\mathbb{P}_i}^-||^2 - 2\nabla \ln p_t^{\mathbb{P}_i}(\mathbf{b}_{\mathbb{Q}}^+ - \mathbf{b}_{\mathbb{P}_i}^-) - \Delta \ln p^{\mathbb{P}_i} + \frac{\nabla p^{\mathbb{P}_i}}{p^{\mathbb{P}_i}}\right) dt\right].$$

Using Itô's rule, we can show the following:

$$\mathbb{E}_{\mathbb{Q}}\left[\ln p_1^{\mathbb{P}_i} - \ln p_0^{\mathbb{P}_i}\right] = \mathbb{E}_{\mathbb{Q}}\left[\int dt \left(\mathbf{b}_{\mathbb{Q}}^+ \cdot \nabla \ln p_t^{\mathbb{P}_i} + \frac{1}{2}\Delta \ln p_t^{\mathbb{P}_i}\right)\right],$$
$$\mathbb{E}_{\mathbb{Q}}\left[\int dt \left(\mathbf{b}_+^{\mathbb{Q}} \cdot \nabla \ln p_t^{\mathbb{P}_i}\right)\right] = \mathbb{E}_{\mathbb{Q}}\left[\ln p_1^{\mathbb{P}_i} - \ln p_0^{\mathbb{P}_i} - \int dt \left(\frac{1}{2}\Delta \ln p_t^{\mathbb{P}_i}\right)\right].$$

The above step is derived using Itô's formula:

$$d \ln p_t^{\mathbb{P}_i} = \mathbf{b}_{\mathbb{Q}}^+ \cdot \nabla \ln p_t^{\mathbb{P}_i} dt + \frac{1}{2}\Delta \ln p_t^{\mathbb{P}_i} dt + \nabla \ln p_t^{\mathbb{P}_i} \cdot d\boldsymbol{\beta}^+(t),$$

which implies (taking expectations on both sides w.r.t. $\mathbb{Q}$ and then dividing by dt. See Equation 5.13, Theorem 5.4 from Särkkä & Solin (2019)):

$$\partial_t \mathbb{E}_{\mathbb{Q}}[\ln p_t^{\mathbb{P}_i}] = \mathbb{E}_{\mathbb{Q}}\left[\mathbf{b}_{\mathbb{Q}}^+ \cdot \nabla \ln p_t^{\mathbb{P}_i} + \frac{1}{2}\Delta \ln p_t^{\mathbb{P}_i}\right].$$

Integrating both sides from 0 to 1 (and applying Fubini) gives the result.

Substituting this back into the KL (the Laplacians cancel out):

$$D_{\mathrm{KL}}(\mathbb{Q}||\mathbb{P}_i) = \mathbb{E}_{\mathbb{Q}}\left[-\ln \frac{p_1^{\mathbb{P}_i}}{p_0^{\mathbb{Q}}}\right]$$
$$+\frac{1}{2}\mathbb{E}_{\mathbb{Q}}\left[\int_0^1 \left(||\mathbf{b}_{\mathbb{Q}}^+ - \mathbf{b}_{\mathbb{P}_i}^-||^2 + 2\nabla \ln p_t^{\mathbb{P}_i} \cdot \mathbf{b}_{\mathbb{P}_i}^- + \frac{\Delta p^{\mathbb{P}_i}}{p^{\mathbb{P}_i}}\right) dt\right].$$

Since $p_t^{\mathbb{P}_i}$ obeys the Fokker-Planck equation (this is the case for diagonal covariance Itô-Processes, otherwise we would have cross terms from the Hes-

sian, see Equation 13.4 in Nelson (1967)[4]):

$$-\partial_t p_t^{\mathbb{P}_i} = \frac{1}{2}\Delta p_t^{\mathbb{P}_i} + \nabla \cdot (\mathbf{b}_-^{\mathbb{P}_i} p_t^{\mathbb{P}_i}).$$

Using the product rule:

$$\begin{aligned} -\partial_t p_t^{\mathbb{P}_i} &= \frac{1}{2}\Delta p_t^{\mathbb{P}_i} + p_t^{\mathbb{P}_i}\nabla \cdot (b_-^{\mathbb{P}_i}) + \mathbf{b}_{\mathbb{P}_i}^- \cdot \nabla p_t^{\mathbb{P}_i}, \\ -\partial_t p_t^{\mathbb{P}_i} - p_t^{\mathbb{P}_i}\nabla \cdot \mathbf{b}_{\mathbb{P}_i}^- &= \frac{1}{2}\Delta p_t^{\mathbb{P}_i} + \mathbf{b}_{\mathbb{P}_i}^- \cdot \nabla p_t^{\mathbb{P}_i}. \end{aligned}$$

Dividing both sides by $p_t^{\mathbb{P}_i}$:

$$-\partial_t \ln p_t^{\mathbb{P}_i} - \nabla \cdot \mathbf{b}_{\mathbb{P}_i}^- = \frac{1}{2}\frac{\Delta p_t^{\mathbb{P}_i}}{p_t^{\mathbb{P}_i}} + \mathbf{b}_{\mathbb{P}_i}^- \cdot \nabla \ln p_t^{\mathbb{P}_i}.$$

Substituting back into the KL we have:

$$D_{\mathrm{KL}}(\mathbb{Q}||\mathbb{P}_i) = \mathbb{E}_{\mathbb{Q}}\left[-\ln \frac{{p_1^{\mathbb{P}_i}}^2}{p_0^{\mathbb{Q}} p_0^{\mathbb{P}_i}}\right] + \frac{1}{2}\mathbb{E}_{\mathbb{Q}}\left[\int_0^1 \left(||\mathbf{b}_{\mathbb{Q}}^+ - \mathbf{b}_{\mathbb{P}_i}^-||^2 - 2\nabla \cdot \mathbf{b}_{\mathbb{P}_i}^-\right) dt\right],$$

We are only interested in terms that depend on $\mathbf{b}_{\mathbb{Q}}^+$ for the optimisation. Using that $p_1^{\mathbb{P}_i} = \pi_1$, which follows as a constraint from the previous iteration, we arrive at:

$$D_{\mathrm{KL}}(\mathbb{Q}||\mathbb{P}_i) \propto -2\mathbb{E}_{p_1^{\mathbb{Q}}}[\ln \pi_1] + \mathbb{E}_{\mathbb{Q}}\left[\int_0^1 \left(\frac{1}{2}||\mathbf{b}_{\mathbb{Q}}^+ - \mathbf{b}_{\mathbb{P}_i}^-||^2 - \nabla \cdot \mathbf{b}_{\mathbb{P}_i}^-\right) dt\right]. \tag{6.23}$$

□

Following the same steps as above, we derive the equivalent result for the backward bridge.

[4]more generally we have $\pm\partial_t p_t^{\mathbb{P}_i} = \frac{1}{2}\Delta p_t^{\mathbb{P}_i} \mp \nabla \cdot (\mathbf{b}_{\pm}^{\mathbb{P}_i} p_t^{\mathbb{P}_i})$

Proposition 2. *The backward half bridge:*

$$\mathbf{b}^-_{\mathbb{P}_i}(t) = \min_{\mathbf{b}^-_{\mathbb{P}} \in \mathcal{B}} \mathbb{E}_{\mathbb{Q}} \left[\int_0^1 \frac{1}{2} \left|\left| \mathbf{b}^-_{\mathbb{P}}(t) - \mathbf{b}^-_{\mathbb{Q}_{i-1}}(t) \right|\right|^2 dt \right],$$
$$s.t. \quad d\mathbf{x}(t) = \mathbf{b}^-_{\mathbb{P}}(t)dt + \boldsymbol{\beta}^-(t), \;\; \mathbf{x}(1) \sim \pi_1,$$

can be expressed as:

$$\mathbf{b}^+_{\mathbb{P}_i}(t) = \min_{\mathbf{b}^-_{\mathbb{P}} \in \mathcal{B}} -2\mathbb{E}_{p_0^{\mathbb{P}}}[\ln \pi_0] + \mathbb{E}_{\mathbb{P}} \left[\int_0^1 \left(\frac{1}{2} ||\mathbf{b}^-_{\mathbb{P}}(t) - \mathbf{b}^+_{\mathbb{Q}_{i-1}}(t)||^2 + \nabla \cdot \mathbf{b}^+_{\mathbb{Q}_{i-1}}(t) \right) \right],$$
$$s.t. \quad d\mathbf{x}(t) = \mathbf{b}^-_{\mathbb{P}}(t)dt + \boldsymbol{\beta}^-(t), \;\; \mathbf{x}(1) \sim \pi_1. \tag{6.24}$$

We now have all the ingredients to carry out g-IPFP using the newly derived half bridge formulations:

- Backward step:

$$\mathbf{b}^+_{\mathbb{P}_i}(t) = \min_{\mathbf{b}^-_{\mathbb{P}} \in \mathcal{B}} -2\mathbb{E}_{p_0^{\mathbb{P}}}[\ln \pi_0] + \mathbb{E}_{\mathbb{P}} \left[\int_0^1 \left(\frac{1}{2} ||\mathbf{b}^-_{\mathbb{P}}(t) - \mathbf{b}^+_{\mathbb{Q}_{i-1}}(t)||^2 + \nabla \cdot \mathbf{b}^+_{\mathbb{Q}_{i-1}}(t) \right) \right],$$
$$s.t. \quad d\mathbf{x}(t) = \mathbf{b}^-_{\mathbb{P}}(t)dt + \sqrt{\gamma}\boldsymbol{\beta}^-(t), \;\; \mathbf{x}(1) \sim \pi_1.$$

- Forward step:

$$\mathbf{b}^+_{\mathbb{Q}_i}(t) = \min_{\mathbf{b}^+ \in \mathcal{B}} -2\mathbb{E}_{p_1^{\mathbb{Q}}}[\ln \pi_1] + \mathbb{E}_{\mathbb{Q}} \left[\int_0^1 \left(\frac{1}{2} ||\mathbf{b}^+_{\mathbb{Q}}(t) - \mathbf{b}^-_{\mathbb{P}_i}(t)||^2 - \nabla \cdot \mathbf{b}^-_{\mathbb{P}_i}(t) \right) \right],$$
$$s.t. \quad d\mathbf{x}(t) = \mathbf{b}^+_{\mathbb{Q}}(t)dt + \sqrt{\gamma}\boldsymbol{\beta}^+(t), \;\; \mathbf{x}(0) \sim \pi_0.$$

Here, $\mathbf{b}^+_{\mathbb{Q}^0}(t)$ is initialised in accordance to the prior (such that $\mathbb{Q}^0 = \mathbb{W}^\gamma$), which in the case of Brownian $\mathbb{W}^\gamma$ motion is zero, that is $\mathbf{b}^+_{\mathbb{Q}_0}(t) = 0$.

We highlight that the newly derived objective has an interpretable decom-

position:

$$\overbrace{-2\mathbb{E}_{p_0^{\mathbb{P}}}\left[\ln \pi_0\right]}^{\textbf{cross entropy/data fit}} + \overbrace{\mathbb{E}_{\mathbb{P}}\left[\int_0^1 \left(\frac{1}{2}||\mathbf{b}_{\mathbb{P}}^{-}(t) - \mathbf{b}_{\mathbb{Q}_{i-1}}^{+}(t)||^2 + \nabla \cdot \mathbf{b}_{\mathbb{Q}_{i-1}}^{+}(t)\right)\right]}^{\textbf{path error}}.$$

The **cross entropy/data fit** term acts as a density estimator encouraging the drift to match its end point distribution, in this case π_0. Meanwhile, the **path error** term enforces the duality of the drifts, as well as making the process akin to the reference prior in the KL sense.

Note that the new terms left to estimate that may seem nontrivial are the cross entropy terms $\mathbb{E}_{p_1^{\mathbb{Q}}}\left[\ln \pi_1\right]$ and $\mathbb{E}_{p_0^{\mathbb{P}}}\left[\ln \pi_0\right]$. We will later discuss how to estimate the cross entropy of two distribution that are only available through samples.

6.3.2 Numerical Implementation

The objectives we have just derived are in general not solvable in closed form for the optimal drift, thus we will require three main approximations:

- A parametrisation of the drift such that we can optimise the objectives approximately using gradient based methods.
- A Monte Carlo approximation to the path integrals arising from the expectations.
- We need to be able to estimate the cross entropy terms as discussed previously.

We proceed to discuss these approximations in more detail.

Numerical Optimization (Differentiable Parametrisation)

In order to optimise the control based variants of the half bridge objectives, we need to parametrise the drift as a differentiable function [5]. We parametrise

[5] We could alternatively parametrise the space of admissible control signals $\mathcal{B}$ as a functional space (i.e. an RKHS (Álvarez et al., 2012)) in which we may be able to carry

the forward and backward drift with neural networks:

$$\begin{aligned}\mathbf{b}_{\mathbb{Q}}^{+}(t) &:= \mathbf{b}_{\phi}^{+}(\boldsymbol{x}(t), t),\\ \mathbf{b}_{\mathbb{P}}^{-}(t) &:= \mathbf{b}_{\theta}^{-}(\boldsymbol{x}(t), t).\end{aligned} \tag{6.25}$$

Rather than optimising over $\mathcal{B}$, the optimisations are carried out in terms of the parameters θ and ϕ respectively. This will also help in the next step, which is estimating the expectations and integrals in the half bridge objective.

We refer to the path measures induced under this parametrisation as $\mathbb{P}_\theta$ and $\mathbb{Q}_\phi$ respectively.

Numerical Integration (Path Error Term)

We can sample from $\mathbb{Q}_\phi$ and $\mathbb{P}_\theta$, whilst enabling the single boundary constraints using the Euler-Mayurama method:

- Forward SDE discretisation for $\mathbb{Q}_\phi$:

$$\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \mathbf{b}_{\phi}^{+}(\boldsymbol{x}_t, t) + \sqrt{\gamma}\boldsymbol{\epsilon}, \quad \mathbf{x}_0 \sim \pi_0, \;\; \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_d).$$

- Backward SDE discretisation for $\mathbb{P}_\theta$:

$$\mathbf{x}_t = \mathbf{x}_{t+\Delta t} - \mathbf{b}_{\phi}^{-}(\boldsymbol{x}_t, t) + \sqrt{\gamma}\boldsymbol{\epsilon}, \quad \mathbf{x}_1 \sim \pi_1, \;\; \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_d).$$

Once we have sampled batches of trajectories $\left\{\left(\boldsymbol{x}_i^{+}(t_k)\right)_{k=0}^{T}\right\}_{n=0}^{N}$ and $\left\{\left(\boldsymbol{x}_j^{-}(t_l)\right)_{l=0}^{T}\right\}_{m=0}^{M}$, we can proceed to estimate the path error term in the objective using the Monte Carlo method:

out minimisation of the half bridges in closed form.

- Backward path error term

$$\mathbb{E}_{\mathbb{P}}\left[\int_0^1 \left(\frac{1}{2}||\mathbf{b}_{\mathbb{P}}^-(\boldsymbol{x}(t),t) - \mathbf{b}_{\mathbb{Q}_{i-1}}^+(\boldsymbol{x}(t),t)||^2 + \nabla\cdot\mathbf{b}_{\mathbb{Q}_{i-1}}^+(\boldsymbol{x}(t),t)\right)\right] \approx \frac{\Delta t}{M}\sum_{m,l}\left(\frac{1}{2}||\mathbf{b}_{\phi}^-(\boldsymbol{x}_m^-(t_l),t_l) - \mathbf{b}_{\theta}^+(\boldsymbol{x}_m^-(t_l),t_l)||^2 + \nabla\cdot\mathbf{b}_{\theta}^+(\boldsymbol{x}_m^-(t_l),t_l)\right). \tag{6.26}$$

- Forward path error term

$$\mathbb{E}_{\mathbb{Q}}\left[\int_0^1 \left(\frac{1}{2}||\mathbf{b}_{\mathbb{Q}}^+(\boldsymbol{x}^-(t),t) - \mathbf{b}_{\mathbb{P}_i}^-(\boldsymbol{x},t)||^2 - \nabla\cdot\mathbf{b}_{\mathbb{P}_i}^-(\boldsymbol{x},t)\right)\right] \approx \frac{\Delta t}{N}\sum_{n,k}\left(\frac{1}{2}||\mathbf{b}_{\theta}^+(\boldsymbol{x}_n^+(t_k),t_k) - \mathbf{b}_{\phi}^-(\boldsymbol{x}_n^+(t_k),t_k)||^2 - \nabla\cdot\mathbf{b}_{\phi}^-(\boldsymbol{x}_n^+(t_k),t_k)\right). \tag{6.27}$$

Cross-entropy Boundary Terms

The densities inside the cross-entropy terms are π_0, π_1, which we only have access to via samples. Thus, we can not apply the standard Monte Carlo method as we can not evaluate the integrands π_0, π_1 at the Monte Carlo samples.

Therefore, we are required to use some form of density estimation for the boundary distributions π_0, π_1, in order to compute a Monte Carlo estimator for the cross entropy boundary terms. The two approximations we consider are non parametric approximations using Kernel Density Estimation and K-Nearest Neighbours.

We need to estimate:

$$\mathbb{E}_{\pi_0^{\mathbb{P}}}[\ln \pi_0] \approx \frac{1}{M}\sum_m^M \ln \hat{\pi}_0(\boldsymbol{x}_m^-(1)),$$

$$\mathbb{E}_{\pi_1^{\mathbb{Q}}}[\ln \pi_1] \approx \frac{1}{N}\sum_n^N \ln \hat{\pi}_1(\boldsymbol{x}_n^+(0)),$$

where $\hat{\pi}_i$ is a KDE based approximation of π_i:

$$\hat{\pi}_0(\boldsymbol{x}) = \frac{1}{N}\sum_{n}^{N} \mathrm{k}_{\boldsymbol{H}_x}(\boldsymbol{x} - \boldsymbol{x}_n),$$
$$\hat{\pi}_1(\boldsymbol{y}) = \frac{1}{M}\sum_{m}^{M} \mathrm{k}_{\boldsymbol{H}_y}(\boldsymbol{y} - \boldsymbol{y}_m),$$

where $\mathrm{k}_{\mathbf{H}}$ is a smooth function called a kernel. In our case, we use the Squared Exponential kernel:

$$\mathrm{k}_{\boldsymbol{H}}(\boldsymbol{x} - \boldsymbol{x}') = (2\pi)^{d/2}|\boldsymbol{H}|^{-1/2}\exp\left(-\frac{1}{2}||\boldsymbol{x} - \boldsymbol{x}'||^2_{\boldsymbol{H}^{-1}}\right),$$

where the bandwidth matrix $\boldsymbol{H}$ is set using Silverman's rule (Silverman, 1986):

$$\boldsymbol{H}_{ij} = \begin{cases} \left(\frac{4}{d+2}\right)^{\frac{2}{d+4}} N^{\frac{-2}{d+4}} \sigma_i^2 & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \tag{6.28}$$

where N is the number of data points used in the KDE estimate. Alternatively, for estimating the densities non-parametrically one can use a K-nearest neighbours basis (Veksler, 2013):

$$\hat{\pi}_0(\boldsymbol{x}) \propto \mathrm{R}_k(\boldsymbol{x}, \{\boldsymbol{x}_n\})^{-d}$$
$$\hat{\pi}_1(\boldsymbol{y}) \propto \mathrm{R}_k(\boldsymbol{y}, \{\boldsymbol{y}_m\})^{-d},$$

where $\mathrm{R}_k(\boldsymbol{x}, \{\boldsymbol{x}_i\})$ is the distance to the k^{th} nearest neighbour of $\boldsymbol{x}$ in $\{\boldsymbol{x}_i\}$. Approximating the density with a KNN basis is the core ingredient for approximating entropy in Singh & Póczos (2016).

6.3.3 Mode Collapse in Reverse KL

The forward KL (e.g. Maximum Likelihood) $\arg\min_{\mathbb{Q}} D_{\mathrm{KL}}(\mathbb{P}||\mathbb{Q})$ blows up when $\mathbb{P}$ tries to assign 0 to regions where $\mathbb{Q}$ has mass. The reverse KL (i.e $\arg\min_{\mathbb{Q}} D_{\mathrm{KL}}(\mathbb{Q}||\mathbb{P})$) can be prone to mode collapse (Zhang et al., 2019).

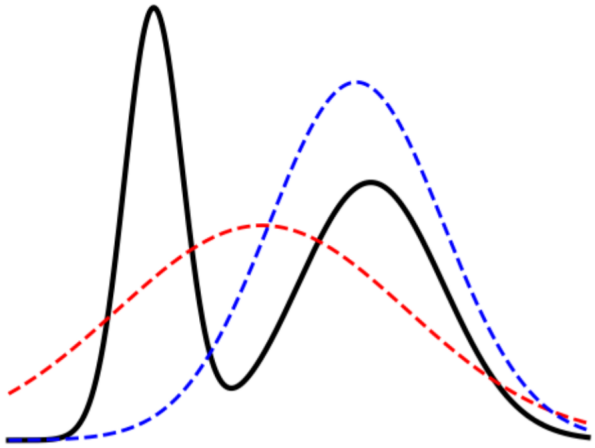

Figure 6.2: Figure 1 from Zhang et al. (2019): Fitting a Gaussian to a mixture of Gaussians (black) by minimizing the forward KL (red) and the reverse KL (blue).

The issue arises from the expectations and minimisations both being taken with respect to $\mathbb{Q}$. Thus we can set the measure $\mathbb{Q}$ to 0 for regions where $\mathbb{P}$ has mass without incurring in an additional penalty since this zeroes out the cost over that region.

As put succinctly in Lawrence (2001):

"In other words, when the approximating distribution is not well matched to the true posterior, the former case is likely to produce a very implausible inference. The latter case will only produce plausible inference, in legal terms it would find the truth, nothing but the truth, but not the whole truth."

Meaning that the reverse KL provides us with the truth (it is precise), yet it does not provide us with the whole truth (i.e. has low recall).

6.4 Conceptual Comparison of Approaches

In this section we briefly define and contrast some of the points of failure of each method based on conceptual observations prior to any experimentation.

- **Copes with Mode Collapse**: This occurs when the optimisation pro-

Table 6.1: Challenges faced by methods. A check mark indicates the method overcomes the respective pitfall.

Method	Copes with Mode Collapse	No Auxiliary Sampling	Handles Distant Support	Scales well in N, M and Δt	No Explicit Density Estimation
Pavon et al. (2018)	✓	✗	✗	✓	✗
Drift Estimation	✓	✓	✗	✗	✓
Stochastic Control	✗	✓	✓	✓	✗

cedure is not penalised for missing a particular mode in the boundary distributions. As a result, the method is prone to missing modes and is unable to map between distributions with a signficant difference in the numbers of modes.

- **No Auxiliary Sampling:** This is concerned with computing integrals with auxiliary distributions (e.g. importance sampling) as done in the method by Pavon et al. (2018).

- **Handles Distant Support:** This issue is concerned with being unable to generate samples that are in the empirical support of π_1 when evolving from π_0 (and vice-versa). This problem is due to the reference process not being able to transport samples from one marginal to areas where there is mass in the other marginal. As a result, this will not give any initial signal to the IPFP algorithm and in lay terms it will not kick off the feedback-loop that refines the constraints in turn.

- **Scales well in N, M and Δt:** By this we mean that the method scales well in the number of data-points, a problem with most non-parametric kernel methods that are $\mathcal{O}((NM\Delta t)^2)$ in memory and $\mathcal{O}((NM\Delta t)^3)$ in computation. Note that there are techniques to overcome this (e.g. sparse variational approaches (Van der Wilk, 2019)).

- **No Explicit Density Estimation:** This applies to methods that require solving an explicit density estimation task at each g-IPFP iteration. Whilst not a major downside, Pavon et al. (2018) mentions that density estimation in high dimensions may be both difficult and

inaccurate, resulting in poor g-IPFP iterations.

We believe that mode collapse and auxiliary sampling are the most impactfull issues when numerically solving the empirical bridge, since there are mitigations for the other issues. We could not find any successful heuristics to improve mode collapse. For auxiliary sampling based approaches as in Pavon et al. (2018), it becomes very difficult to scale them to high dimensions. In particular, one needs to design an approach that is aware of where the mass in the integrand resides, as in Osborne et al. (2012). This is where our proposed methods have the biggest advantage.
The problem of missing the support of the boundary distributions can be addressed by re-scaling the data (i.e. standardise) so that it is at a scale where the Brownian motion prior can cover the support of the boundaries when going from one to the other. Alternatively, increasing the value of γ will also achieve this. See Table 6.1 for a conceptual comparison between the studied methods in this thesis.

6.5 Summary

In this section we studied and proposed different approaches for numerically solving the Schrödinger bridge problem. Since the algorithms used for solving the Schrödinger bridge are derived from the g-IPFP framework, much of the focus in this chapter is placed on how to solve the half bridges numerically, given these are the inner steps of the g-IPFP based algorithms.

- Presented the method by Pavon et al. (2018) for solving the Schrödinger system via a maximum likelihood variant of Fortet's. This method proposes representing the potentials in the Schrödinger system as parametric functions and enforces the steps in Fortet's algorithm using maximum likelihood.

- Introduced a novel direct drift estimation based approach using Gaussian processes. This method proposes to approximate the optimal half bridge path measures directly by fitting a dual drift to trajectories sam-

pled from the optimal half bridge path measure. We use the method by Ruttor et al. (2013) in order to fit the drift of an SDE observed through samples.

- Introduced a novel approach for solving the dynamic half bridges numerically using the stochastic control formulation of the half bridges. In this approach we represent the drift as a parametric function and numerically minimise the half bridge objective with respect to the drifts parameters.
- Presented a conceptual comparison of the 3 discussed approaches , discussing their strengths and weakness.

Chapter 7

Experiments

In this section we explore the performance of our methods for the empirical Schrödinger bridge on toy 1D and 2D datasets. We limit to toy data which we can visually evaluate, due to the early nature of the approaches. This helps gainining a practical understanding of their mechanics and limitations before handling real world data.

Our experiments can be grouped into the following tasks:

- **Unimodal to Unimodal Tests:** In these experiments we evaluate the ability of the algorithms to transition from simple unimodal Gaussian distributions with different means and variances. This is one of the simplest tests and serves as the most basic sanity check for each method.
- **Unimodal to Multimodal Tests:** Some of the methods might suffer from mode collapse. In order to evaluate the ability of a method to fit multimodal boundaries, we set up a simple test of mapping from a unimodal distribution to a multimodal one.
- **Non-Gaussian Distributions Tests:** Here we explore the tasks for mapping from a unimodal Gaussian to a complex non-Gaussian distribution, such as the moons and circles datasets (Pedregosa et al., 2011). Due to time constraints, we only explored this test with the direct drift estimation approach.

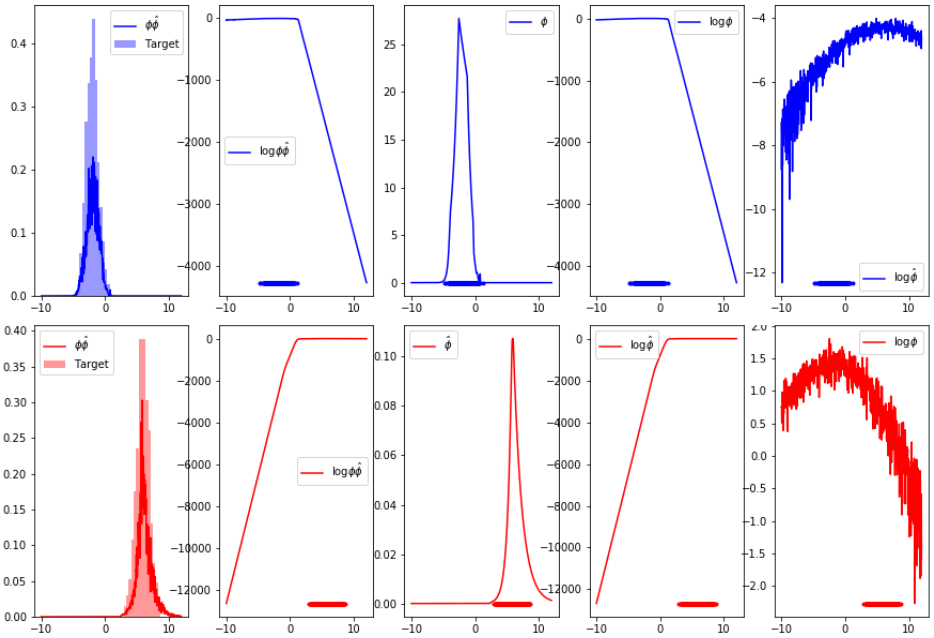

Figure 7.1: Schrödinger Bridge results using the method by Pavon et al. (2018) on unimodal Gaussian 1D data. We included the potentials since it was helpful to illustrate how they compensate each other.

Finally, we select two toy datasets (one unimodal to unimodal and one unimodal to bimodal) which all methods managed to fit visually, and construct a hypothesis test for comparing the methods against each other in a more statistically sound framework. We have open-sourced the code for the experiments and algorithms [1].

7.1 Method by Pavon et al. (2018)

We found Pavon et al. (2018) to be extremely sensitive to the value of γ in the prior $\mathbb{W}^\gamma$: for most values of $\gamma < 5$, the method worked poorly and the boundary distributions would collapse to Dirac delta distributions. In

[1] For our proposed approach the experiments can be found at `https://github.com/AforAnonyMeta/SC_IPFP/tree/master/SC_IPFP/torch/final_notebooks`. For the experiments with the method by Pavon et al. (2018) the notebooks can be found at `https://github.com/AforAnonyMeta/DataDrivenSchrodingerBridge/tree/master/notebooks`

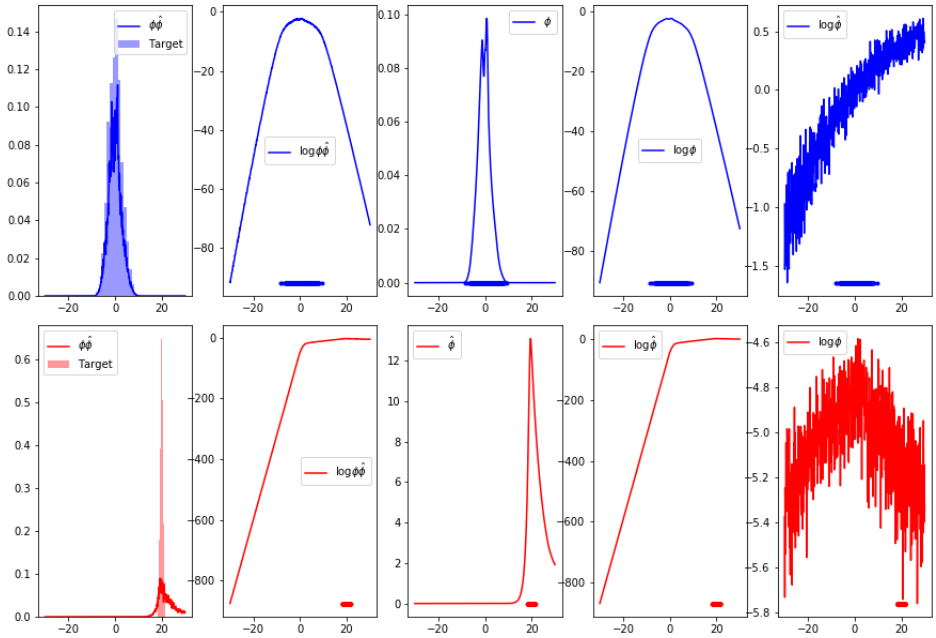

Figure 7.2: Schrödinger Bridge results using the method by Pavon et al. (2018) on unimodal Gaussian 1D data and different variances.

order to overcome this issue, we had to experiment with large values of γ. Alternatively, we found that another heuristic to overcome this issue was to standardise the data. This led us to understand that the issue was due to not being able to cover the support of the boundary distributions when sampling from the transition density of $\mathbb{W}^\gamma$.

We set $\gamma = 1000$ in all the following experiments, unless stated otherwise. We determined this value to be stable enough for us to explore the method. We used 400 Monte Carlo samples for each of the expectations approximated. A single hidden layer neural network (LeCun et al., 2015) with ReLu activations (Glorot et al., 2011) and 500 hidden units is used to parametrise the potentials across all experiments. We used Glorot initialisation (Glorot & Bengio, 2010) to initialize the weights and the optimizer was Adagrad (Duchi et al., 2011). Each example was trained for an outer loop of 250 epochs with 150 inner epochs for the MLE sub-iterations.

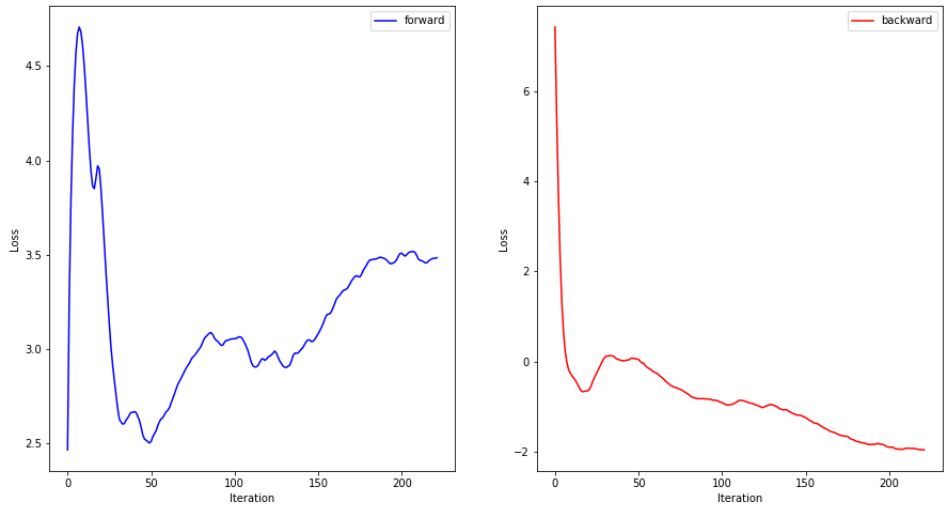

Figure 7.3: Likelihood per epoch for the method by Pavon et al. (2018) applied to unimodal Gaussian 1D data with different variances.

7.1.1 Unimodal Experiments

In this section, we explore transitioning in between normal distributions with different means and variances. We also illustrate how the method fails for small values of γ and correctly diagnose a cause, in addition to proposing heuristics to overcome this issue.

Different Means, Same Variances

In this experiment, we explore the following generative process for the marginals:

$$\pi_0(x) = \mathcal{N}(x; 5, 1)$$
$$\pi_1(y) = \mathcal{N}(y; -2, 1).$$

Results can be found in Figure 7.1, where we visually observe a good fit of the boundary distributions. This experiment serves mostly as a sanity check and to tune γ to a value where the method is stable.

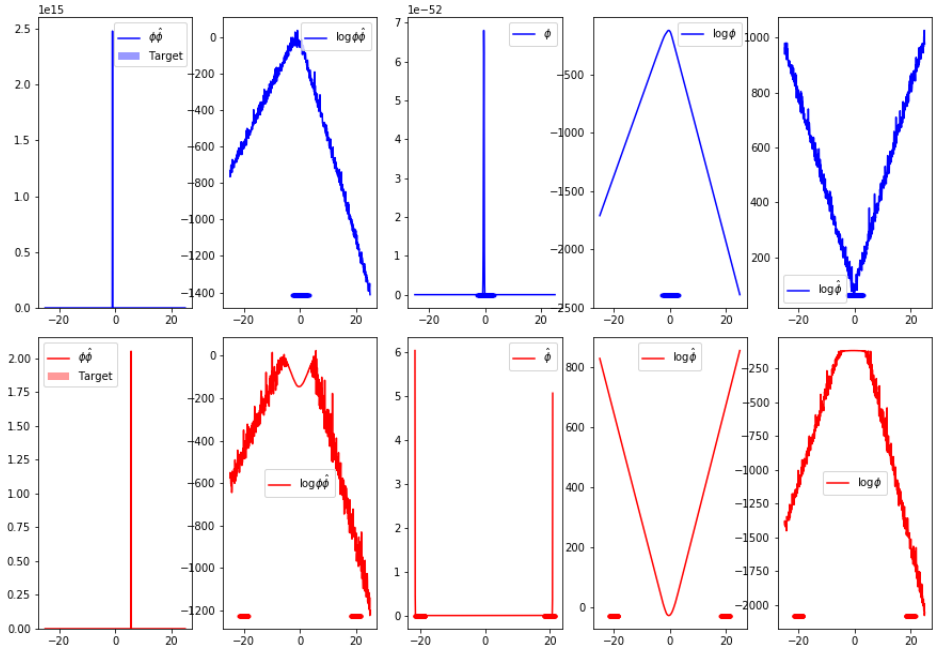

Figure 7.4: Schrödinger Bridge results using the method by Pavon et al. (2018) on unimodal to bimodal Gaussian 1D data. This example illustrates the Dirac delta collapse of the marginals.

Different Means and Variances

We generated a 1D toy dataset using the following process:

$$\pi_0(x) = \mathcal{N}(x; 0, 9)$$
$$\pi_1(y) = \mathcal{N}(y; 20, 0.5^2).$$

Results can be found in Figure 7.2, where we observe a good fit for π_0, yet visually not such a good fit for π_1. Furthermore, the Lagrange constraint integral converged to approximately 0.6 rather than 1, meaning the distribution was not normalised via this approximate Lagrange multipliers based approach. From Figure 7.3 it is not clear if the method fully converged.

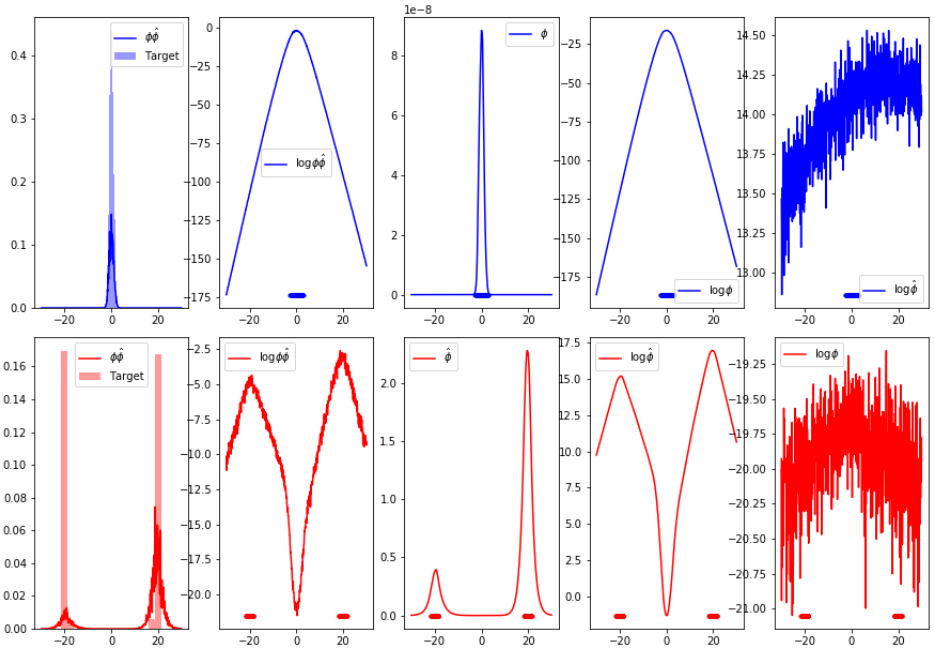

Figure 7.5: Schrödinger Bridge results using the method by Pavon et al. (2018) on unimodal to bimodal Gaussian 1D data.

Delta Collapse

In this section we explore how the method in Pavon et al. (2018) fails for small values of γ. In this case we use the different means, same variance dataset and set $\gamma = 1$. We observed similar behaviour up to $\gamma = 10$ and it was not until $\gamma = 100$ that results became stable. In order to save time and avoid having to tweak γ per experiment, we choose an overly conservative value of $\gamma = 1000$ which we used for all experiments in this section.

As shown in Figure 7.4, the learned marginal distributions have effectively collapsed into a Dirac delta function. We believe this happens due to the transition probability not being able to hit points in the support of π_0 when mapping from π_1 (and vice-versa). The empirical proof backing this is the fact that this only happens for low values of γ and that standardising the data or considering toy examples whose modes lie in the unit cube fixes this issue. We note that in Pavon et al. (2018) they use a value of $\gamma = 2$ and

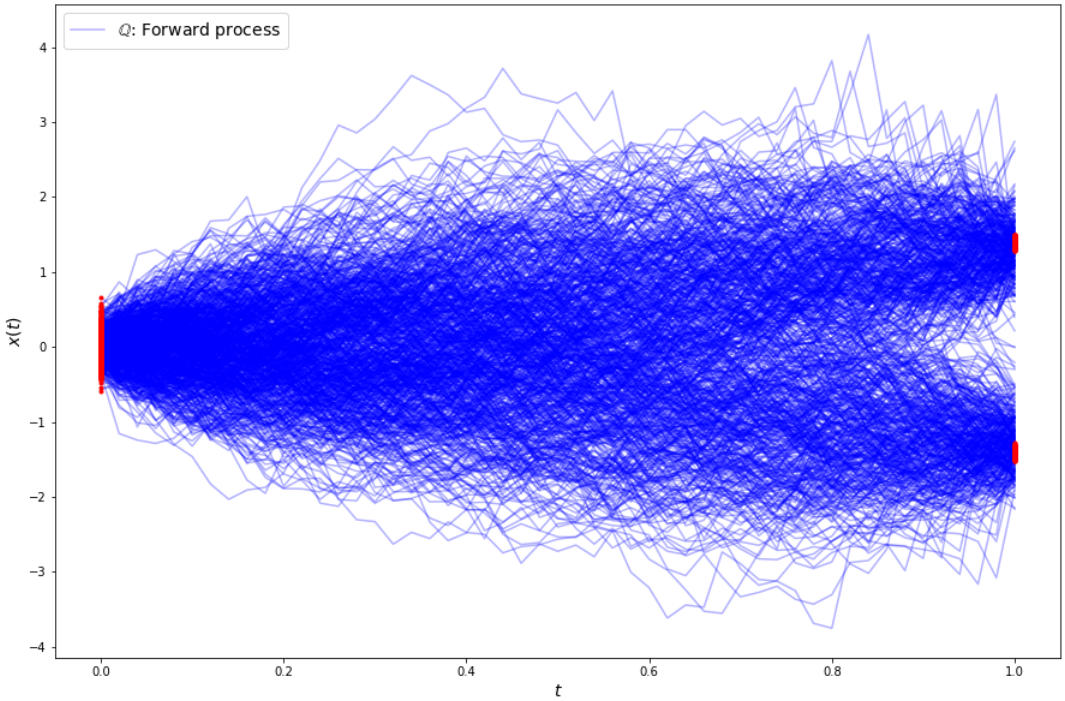

Figure 7.6: Extracted drift from optimal potentials. Marginals highlighted in red for visibility.

furthermore the only 2 examples they have are distributions centered very close to 0, all with means < 3, which is where we observe the cutoff happens empirically.

7.1.2 Multimodal Experiments

Unfortunately, we were unable to obtain sensible results for trimodal data with distinct modes, thus we focused solely on bimodal data where we managed to get this method working.

$$\pi_0(x) = \mathcal{N}(x; 0, 1)$$
$$\pi_1(y) = \frac{1}{2}\mathcal{N}(y; 20, 0.5^2) + \frac{1}{2}\mathcal{N}(y; -20, 0.5^2).$$

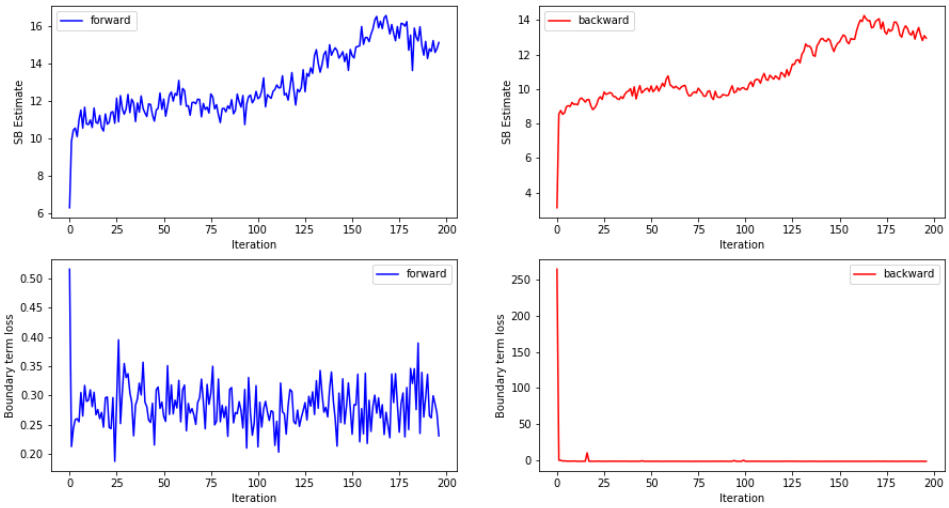

Figure 7.7: Loss per epoch for fitted unimodal Schrödinger (SB) using the DDE method.

As shown in Figure 7.5, the method correctly identifies the modes and in the case of π_1 it correctly identifies the variance. However, for π_0 we can see that the method fits some slightly broader variances than those of the target.

7.1.3 Extracting the Optimal Drift

Using Proposition 3.3 of Pavon & Wakolbinger (1991), we extract the optimal control signal/drift $\mathbf{b}_t^+$ from the potentials fitted in Figure 7.5 by Pavon et al. (2018)'s method using:

$$\mathbf{b}_t^+ = \gamma \nabla \ln \phi_t^*(\boldsymbol{x}(t)).$$

We then generate trajectories using the EM method in order to visualise samples from the fitted path measure. Results can be seen in Figure 7.6. Whilst the trajectories successfully split and manage to match the modes, they do not match the variances of the fitted potentials from Figure 7.5. This suggests that although the method from Pavon et al. (2018) may seem to solve the static bridge and provide some reasonable approximation for

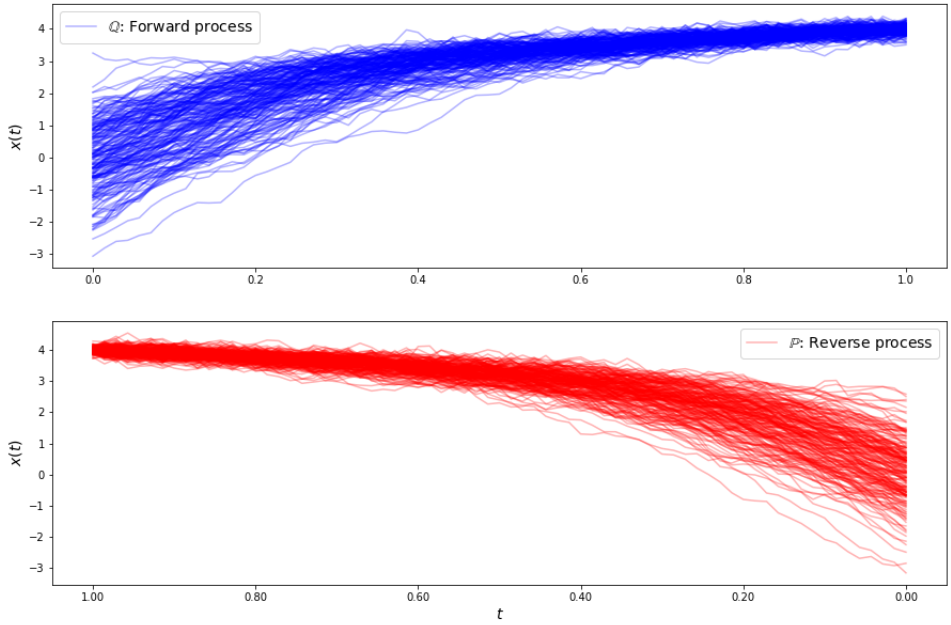

Figure 7.8: Fitted SB trajectories using the DDE approach on unimodal to unimodal 1D data.

the potentials in the Schrödinger system, we are not able to recover a good estimate of the drift from these fitted potentials. This might be due to the inaccuracies arising from the importance sampling used in the method generating an overall poor estimate of the logarithmic derivative. We do note that the method by Pavon et al. (2018) does seem to provide reasonable estimates to the solution of the FPK equation and thus one can still interpolate the fitted distribution at different times.

7.2 Our Approach - Direct Drift Estimation (DDE)

We found this approach to be much less brittle than the method by Pavon et al. (2018) when it came to distributions whose modes were far from the

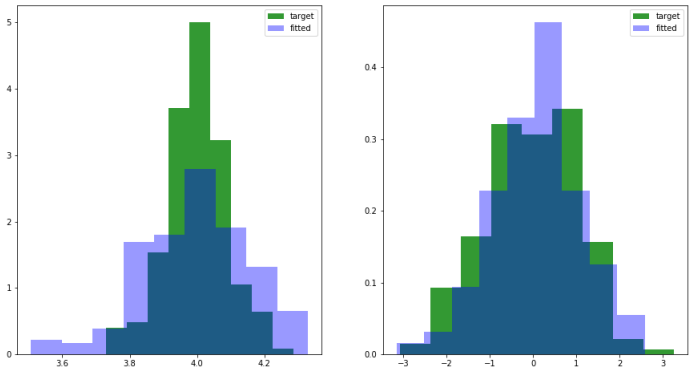

Figure 7.9: Fitted SB marginals using the DDE approach on unimodal to unimodal 1D data.

origin. However, it still suffers from the problem that it cannot fit distributions whose supports are too far from each other. We were empirically able to fit distributions whose means exceeded 4 with a value of $\gamma = 1$. All experiments in this section were performed with $\gamma = 1$ and in some cases we adapted the examples from previous sections to have smaller means in order to accommodate for the issue regarding relative support of the distributions that we mentioned earlier. An important detail is that we did not optimise the hyperparameters of the fitted GP, this was mostly due to time constraints. Due to the typical memory constraints in Gaussian processes, we use 200 datapoints in each toy dataset and 70 time steps for the EM method.

7.2.1 1D Toy Experiments

Unimodal Experiments

Given the success of this method in more complex tasks, we omit the simple experiment of mapping in between different mean but shared variance Gaussian distributions.

Different Means and Variances

We generate this dataset following:

$$\pi_0(x) = \mathcal{N}(x; 0, 1)$$
$$\pi_1(y) = \mathcal{N}(y; 4, 0.1^2).$$

We can observe from Figure 7.8 that the trajectories match the correct evolution of the means and the variances. Inspecting Figure 7.9, we can observe a minor struggle in matching the 0.1 variance for which our approach seems to estimate a slightly higher variance. Overall we can observe a good fit.

Multimodal Experiments

We experiment with mapping in between unimodal and multimodal distributions. We empirically find that this is the only method that can robustly cope with more than 2 modes.

Unimodal to Bimodal

We follow the generative following generative process for generating the toy data:

$$\pi_0(x) = \mathcal{N}(x; 0, 1)$$
$$\pi_1(y) = \frac{1}{2}\mathcal{N}(y; 1.8, 0.6^2) + \frac{1}{2}\mathcal{N}(y; -1.9, 0.6^2).$$

Results can be found in Figures 7.11 and 7.10. We observe mostly a good fit for the unimodal boundary and for the bimodal boundary with the exception of 3 outliers that were transported to an area that should have close to 0 probability mass. We found that these outliers occurred outside of the Schrödinger bridge context and rather due to the method by Ruttor et al. (2013) (used for fitting the drift of SDEs) struggling in matching low variances at the boundaries. It is possible that optimising the GPs hyperparameters may also improve on this.

Unimodal to Trimodal

We generate the 1D toy dataset following the following distribution:

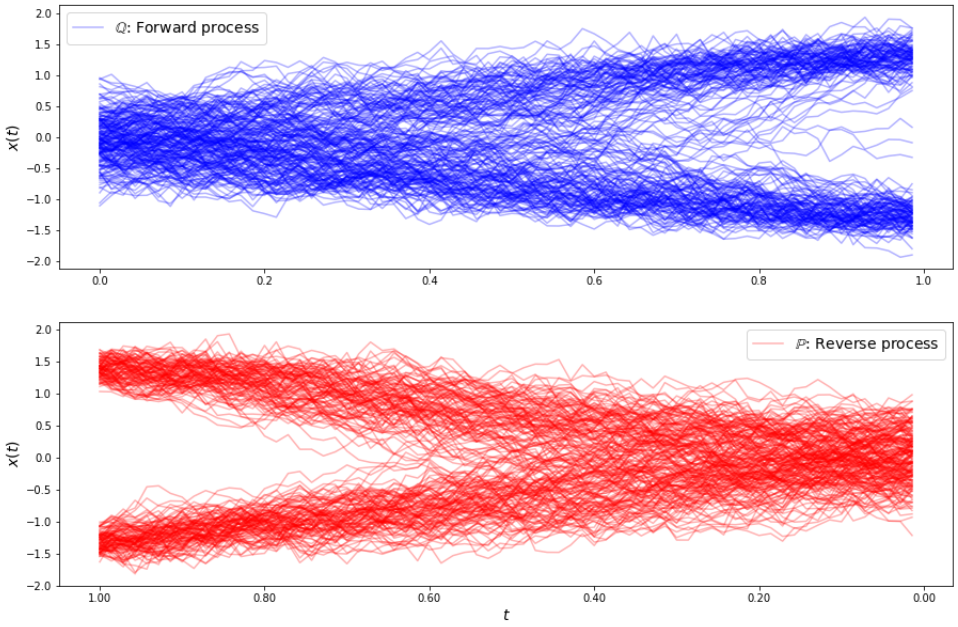

Figure 7.10: Fitted SB trajectories using the DDE method on the unimodal to bimodal 1D dataset.

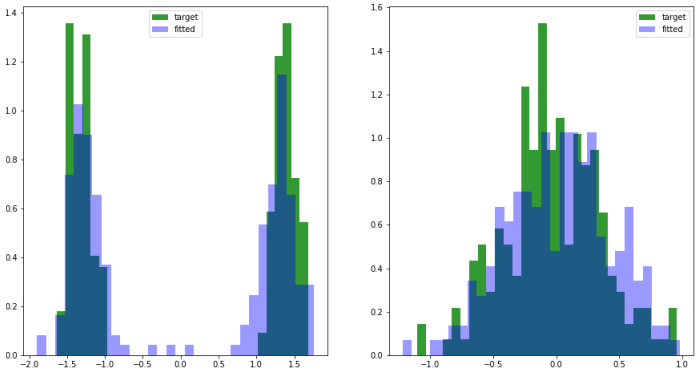

Figure 7.11: Fitted SB marginals using the DDE method on the unimodal to bimodal 1D dataset.

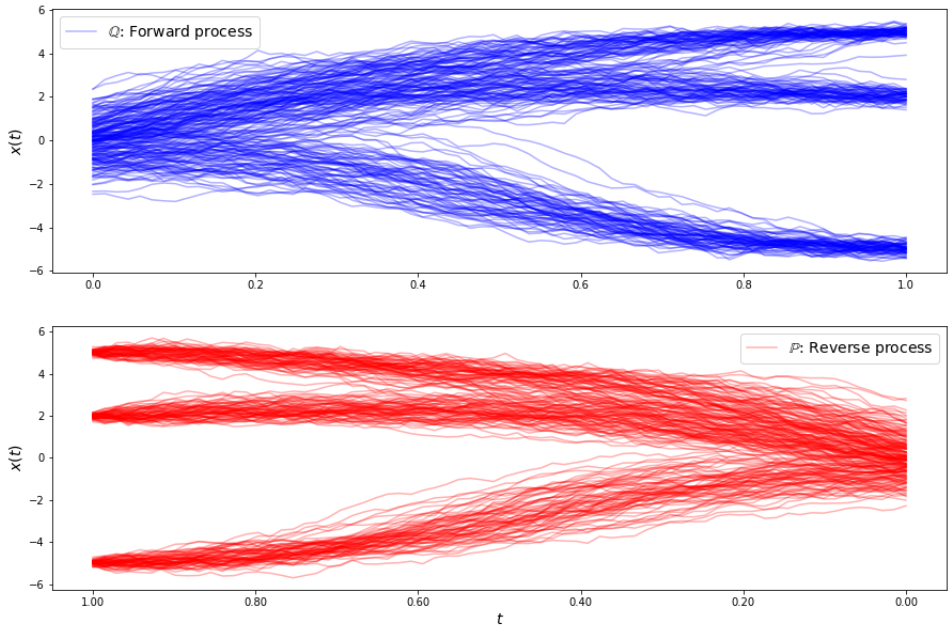

Figure 7.12: Fitted SB trajectories for unimodal to trimodal data using DDE, successfully fitting three modes.

$$\pi_0(x) = \mathcal{N}(x; 0, 1)$$
$$\pi_1(y) = \frac{1}{3}\mathcal{N}(y; 5, 0.5^2) + \frac{1}{3}\mathcal{N}(y; 2, 0.5^2) + \frac{1}{3}\mathcal{N}(y; -5, 0.5^2).$$

We can observe from Figures 7.12, 7.13 that the boundaries distributions are matched nicely both in variance and mean. Furthermore, the trajectories look sound: they have some reminiscence of Brownian motion, follow nice simple paths, and look dual to each other. We also observe that for the two modes that are close to each other there are two outliers in low probability areas.

7.2.2 2D Toy Experiments

Given the success of the DDE method in going beyond splitting two modes, we decided to experiment further and go into 2-dimensions. Due to time

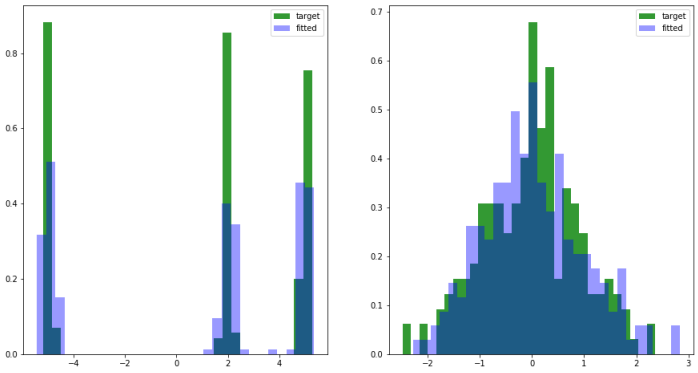

Figure 7.13: Fitted SB trajectories for unimodal to trimodal data using DDE.

constraints, we chose not to carry out 2-dimensional experiments for the method in Pavon et al. (2018), furthermore we were unable to get their method to split beyond 2 modes.

Trimodal Experiment

In this section we consider the 2D extension of the trimodal experiment. We use the following distributions to generate our toy examples:

$$\pi_0(\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x}; -\begin{pmatrix}0.1\\0.1\end{pmatrix}, 0.19^2\,\mathbb{I}_2\right)$$

$$\pi_1(\boldsymbol{y}) = \frac{1}{3}\mathcal{N}\left(\boldsymbol{y}; -\begin{pmatrix}3.3\\0.1\end{pmatrix}, 0.11^2\mathbb{I}_2\right) + \frac{1}{3}\mathcal{N}\left(\boldsymbol{y}; \begin{pmatrix}3.5\\3.5\end{pmatrix}, 0.11^2\mathbb{I}_2\right) + \frac{1}{3}\mathcal{N}\left(\boldsymbol{y}; \begin{pmatrix}0.7\\0.7\end{pmatrix}, 0.11^2\mathbb{I}_2\right).$$

In Figure 7.14 we can see how the trajectories successfully split and merge into 3 modes. In Figure 7.15 we observe that the marginal constraints are enforced to a reasonable amount, matching means quite well, yet struggling partly with the variance of the low variance boundaries.

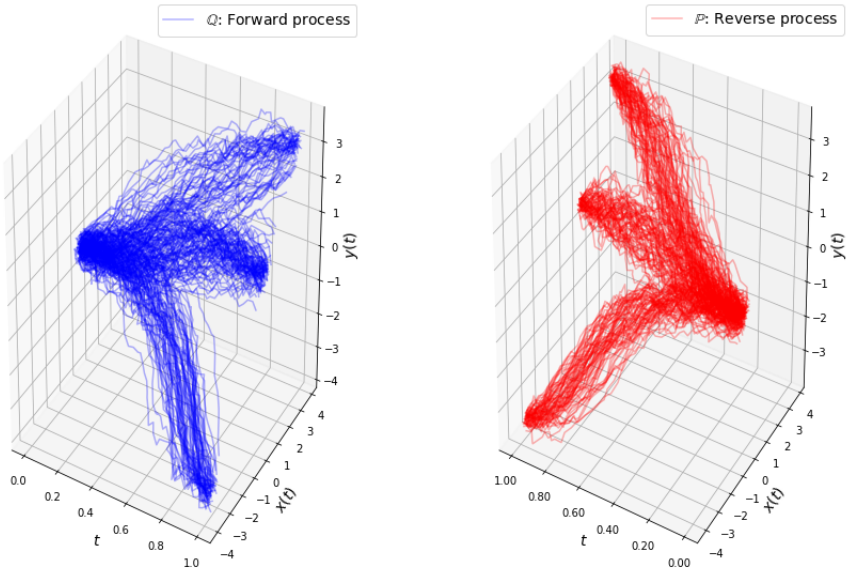

Figure 7.14: Fitted SB trajectories using the DDE method for unimodal to trimodal data.

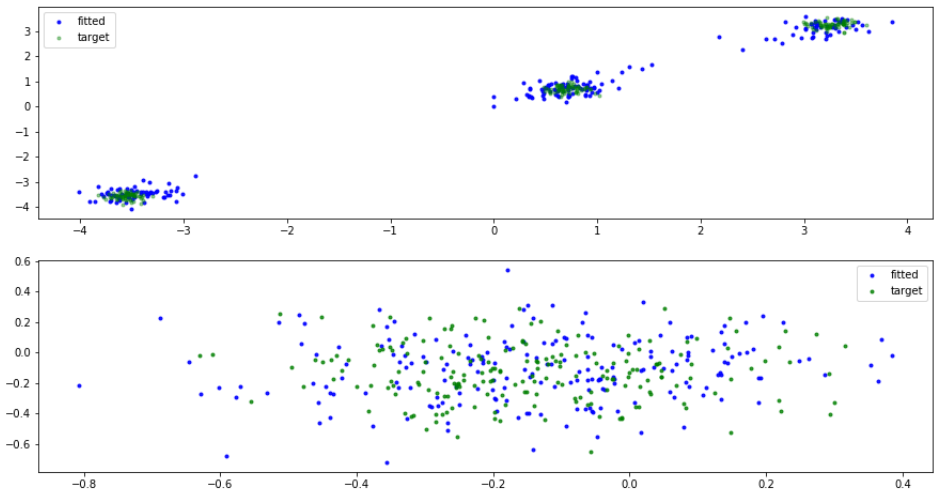

Figure 7.15: Fitted SB marginals using the DDE method for unimodal to trimodal data.

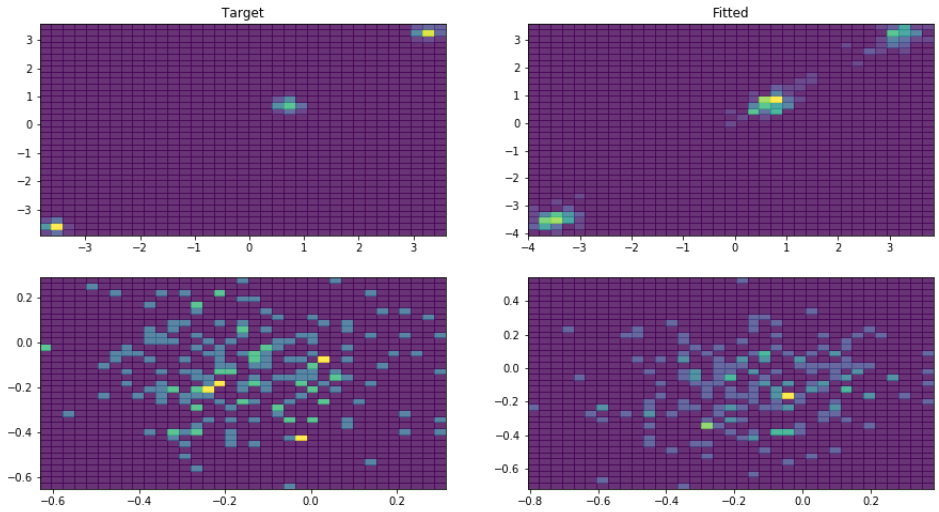

Figure 7.16: 2D histograms for the fitted SB marginals using the DDE method for unimodal to trimodal data. On the left we show the true empirical distribution and on the right we show the empirical distribution learned by our Schrödinger bridge mapping.

Circles Dataset

We used the circles dataset (Pedregosa et al., 2011) which consists of two concentric circles. In this task we consider mapping from a standardised unimodal Gaussian to the Circles data.

$$\pi_0(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \mathbf{0}, \mathbb{I}_2)$$
$$\hat{\pi}_1(\boldsymbol{y}) = \text{GenerateCircles(factor=0.3, radius=3.5, noise=0.03)}.$$

The radius and factor (distance between circles) were picked so the data was still within a close proximity to the origin, whilst making the circles distinct.

We can see from Figure 7.17 that some very structured inner circle-like trajectories are formed going forward and backward with a dual like pattern. Furthermore, from Figures 7.18, 7.19 we can observe a good fit in terms of structure, but the fits of our method are slightly more dispersed, with higher variance. We noticed this could be improved by using more training data and

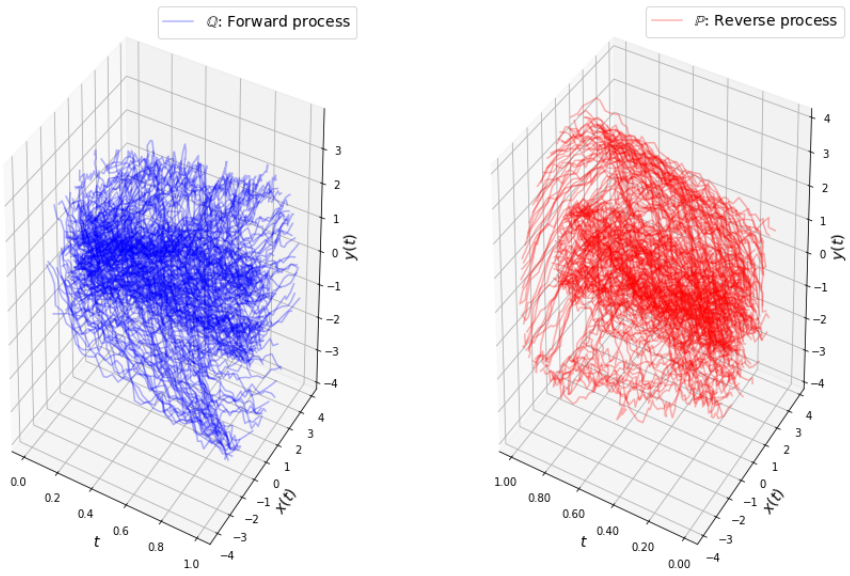

Figure 7.17: Fitted SB trajectories using the DDE method for unimodal to circles data. We can observe the nice and clear concentric circles trajectories arising in the learned bridge.

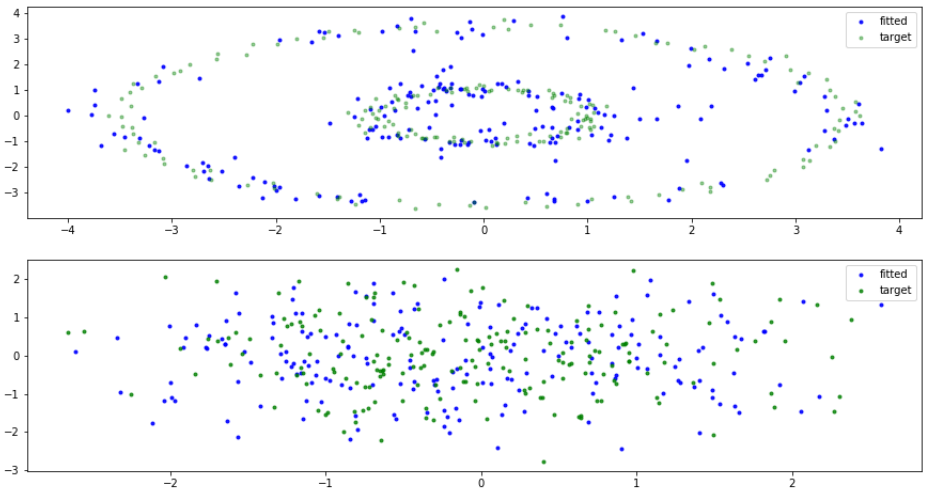

Figure 7.18: Fitted SB Boundary distributions using the DDE method on unimodal to circles data.

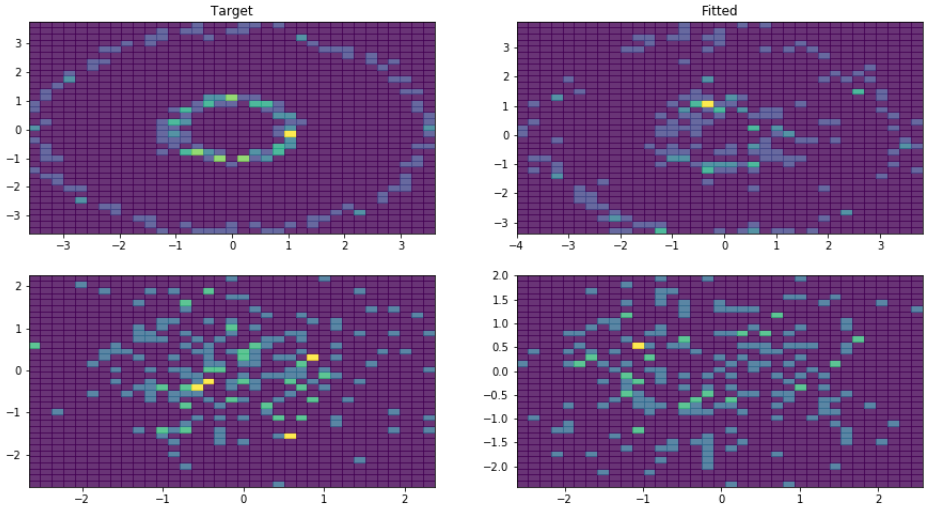

Figure 7.19: 2D histograms for the fitted SB marginals using the DDE method for unimodal to circles data.

tweaking the GP hyper-parameters. Further experiments should be carried out to diagnose and solve this issue.

Moons Dataset

We use the moons dataset from Pedregosa et al. (2011), generated by the following procedure:

$$\pi_0(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \mathbf{0}, \mathbb{I}_2)$$
$$\hat{\pi}_1(\boldsymbol{y}) = \text{GenerateMoons(scale\_x=1.8, scale\_y=3.8, noise=0.03).}$$

Similar to the circles dataset, the scales were picked to give a nice visual scale and a clear separation between the moons, making the task as simple as possible given that we were still in an exploratory stage.

We observe that the DDS method struggles with the moons dataset in Figures 7.22, 7.21. However, it still manages to capture most of the structure in the marginals, except for the higher variances. As with the other experiments, the trajectories (Figure 7.20) have a dual like nature.

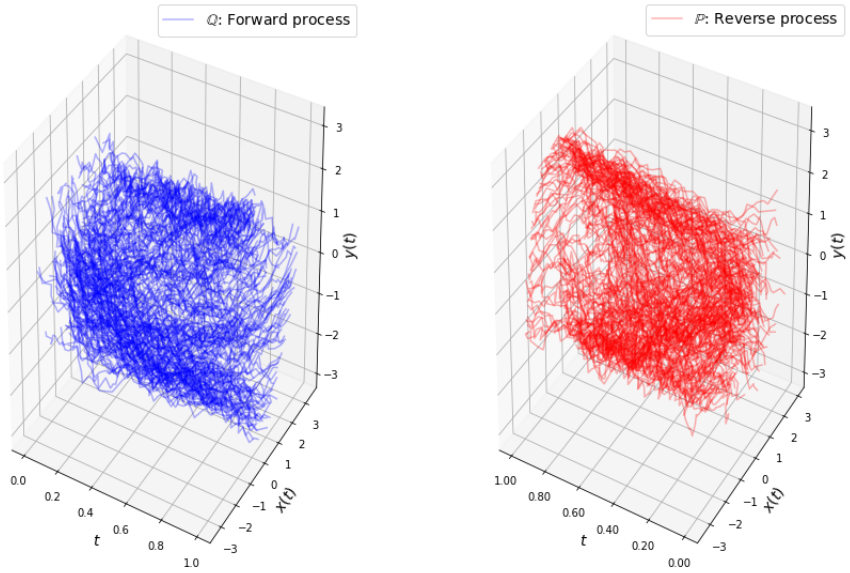

Figure 7.20: Fitted SB trajectories using the DDE method for unimodal to moons data.

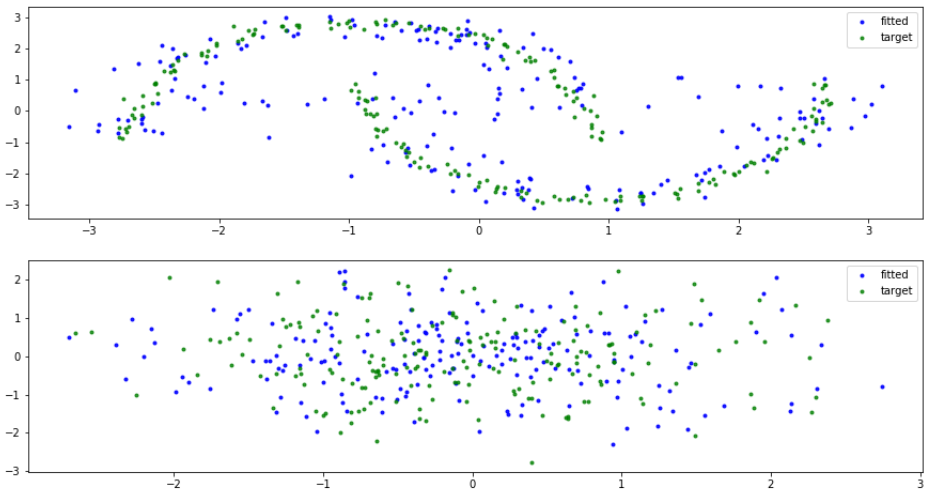

Figure 7.21: Fitted SB Boundary distributions using the DDE method for unimodal to moons data.

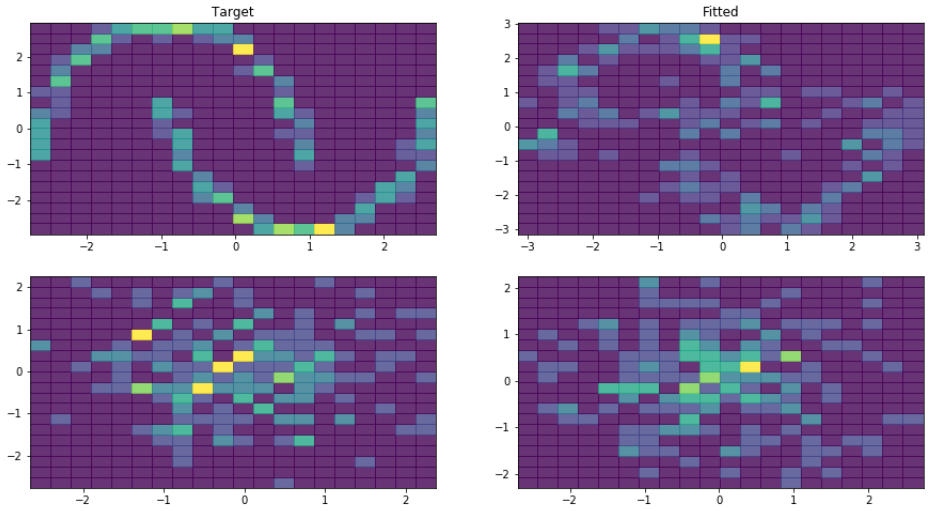

Figure 7.22: 2D histograms for the fitted SB marginals using the DDE method for unimodal to moons data.

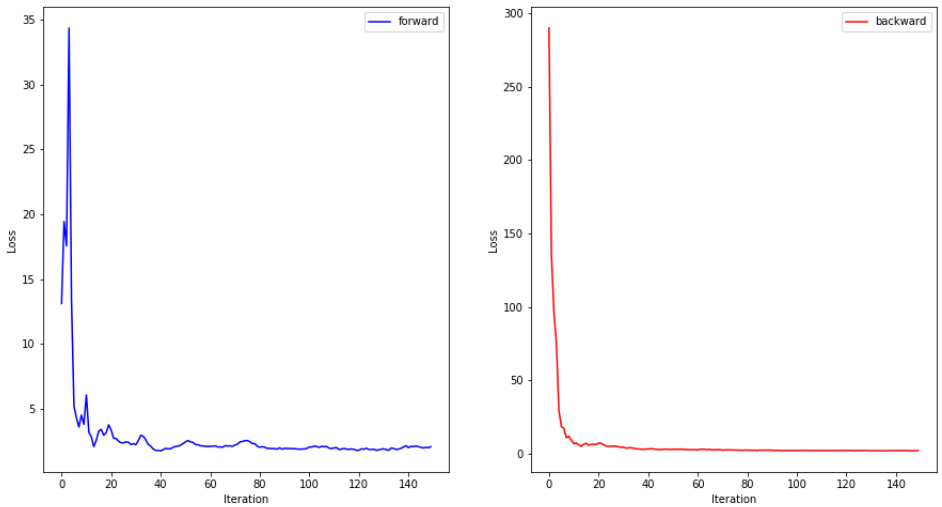

Figure 7.23: Loss per epoch for fitted unimodal SB using tanh activation, single layer and 200 hidden units for SC approach.

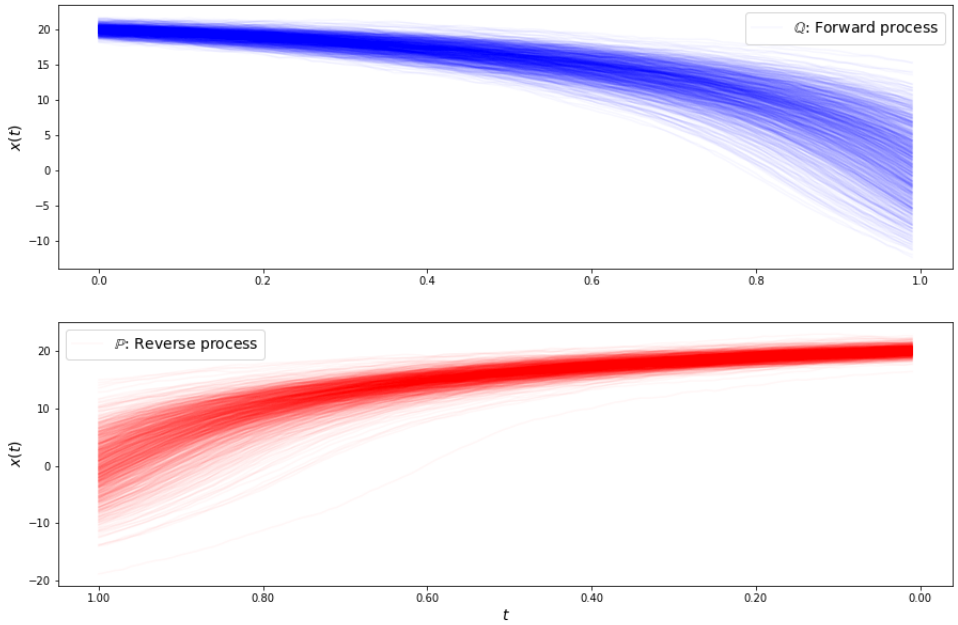

Figure 7.24: Fitted SB trajectories using the SC method with tanh activations, single layer and 200 hidden units, fitted on the unimodal dataset. Note that the trajectories seem much smoother than the ones induced by the direct drift approach. This is mostly a visual effect due to the scale of the y axis being much higher. In other words you will see less noise if you zoom out.

7.3 Stochastic Control (SC) Approach

Throughout this section we employ neural networks (LeCun et al., 2015) to parametrise the drifts in Equation 6.25 and experiment with 3 different types of activations: Tanh, Hard Tanh, ReLu. We varied layer widths and depths in order to obtain good results, without careful regard for overfitting, since, as mentioned, we are merely exploring the capacities and faults of the approaches. Across all experiments we initialise the weights of the neural networks using Glorot initialisation (Glorot & Bengio, 2010) and we used the Adagrad optimiser (Duchi et al., 2011). Due to the KNN based approach to computing cross entropy being unstable, we used the KDE based approach across all experiments. This method allowed using bigger datasets than the previous two methods, thus we used 900 datapoints in each toy dataset, 100

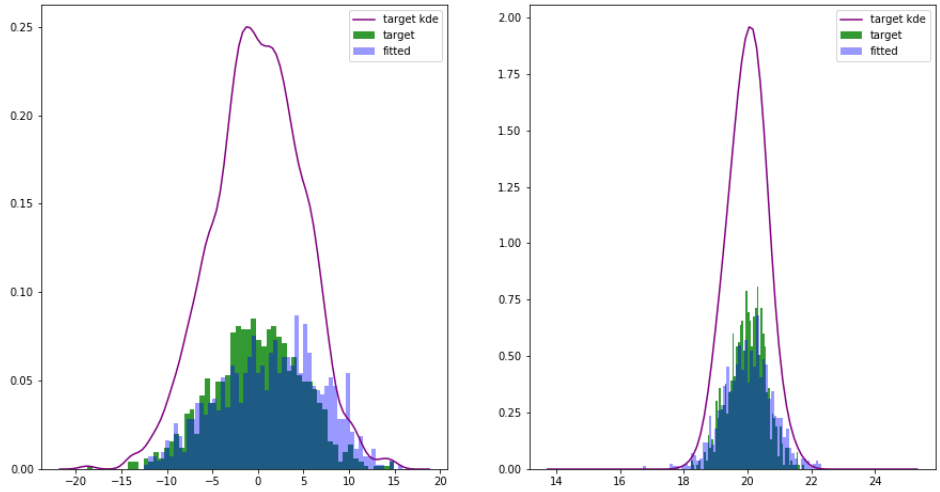

Figure 7.25: Fitted SB boundary distributions using the SC method with tanh activations, single layer and 200 hidden units.

time steps for the EM method and a batch size of 900 (only one batch) for the optimiser.

7.3.1 Unimodal Experiments

For the unimodal experiments we focus on mapping between normal distributions with different means and variances.

Small to Big Variance

Data was generated using:

$$\pi_0(x) = \mathcal{N}(x; 0, 3)$$
$$\pi_1(y) = \mathcal{N}(y; 6, 0.1^2)$$

For these particular experiments, we found very little differences among the trajectories learned by different activations. Results can be found in Figures 7.25 and 7.24, where we can see a good fit for the marginal with low variance, but a worse fit for the high variance distribution where the variance is

matched but the mode is not.

7.3.2 Multimodal Experiments

In this section we followed the generative process:

$$\pi_0(x) = \mathcal{N}(x; 0, 1)$$
$$\pi_1(y) = \frac{1}{2}\mathcal{N}(y; 20, 0.6^2) + \frac{1}{2}\mathcal{N}(y; -20, 0.6^2).$$

We parametrised the drifts with 3 hidden layers, each having 20 hidden units, with the exception of the final experiment and best performing model for which we used 2 hidden layers, each with 200 units. In general, we found that increasing the number of units and layers made the optimisation process more unstable and typically required reducing the initial step size for the optimiser. These experiments are used for investigating how the stochastic control based approach can handle mode collapse.

ReLu

For these experiments, we use the rectified linear-unit (ReLu) (Glorot et al., 2011). We show results of a further experiment with larger means to illustrate the shape of the trajectories induced by this activation. As shown in Figures 7.26 and 7.28, the learned trajectories are more curved (almost parabolic like) than the trajectories induced by tanh based activations. We also observe that whilst being on the right track to matching the means, it fails to match the variance of the bimodal marginals. One positive aspect is that the curves do have a dual like nature.

Tanh

Here we use the hyperbolic-tangent (Tanh) as the activation function. We also experimented with similar smooth activations and obtained similar results. We observe from Figures 7.29 and 7.30 that the method roughly matches the means, with a clear dual nature between the trajectories. However, a strange spike forms where the two modes try to merge. This spike

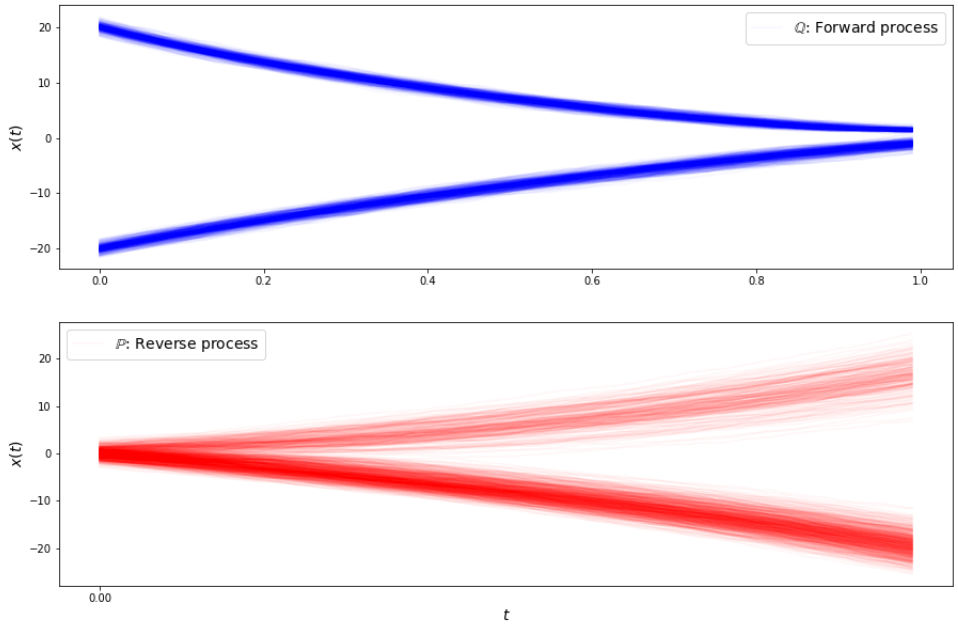

Figure 7.26: Fitted SB trajectories using ReLu activation, 3 hidden layers of 20 hidden units each

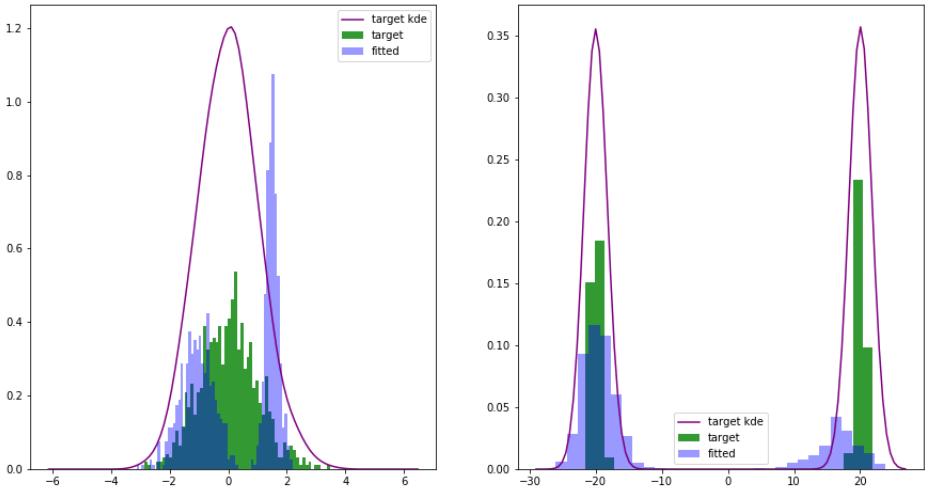

Figure 7.27: Fitted SB boundary distributions using the SC method with ReLu activations, 3 hidden layers of 20 hidden units each.

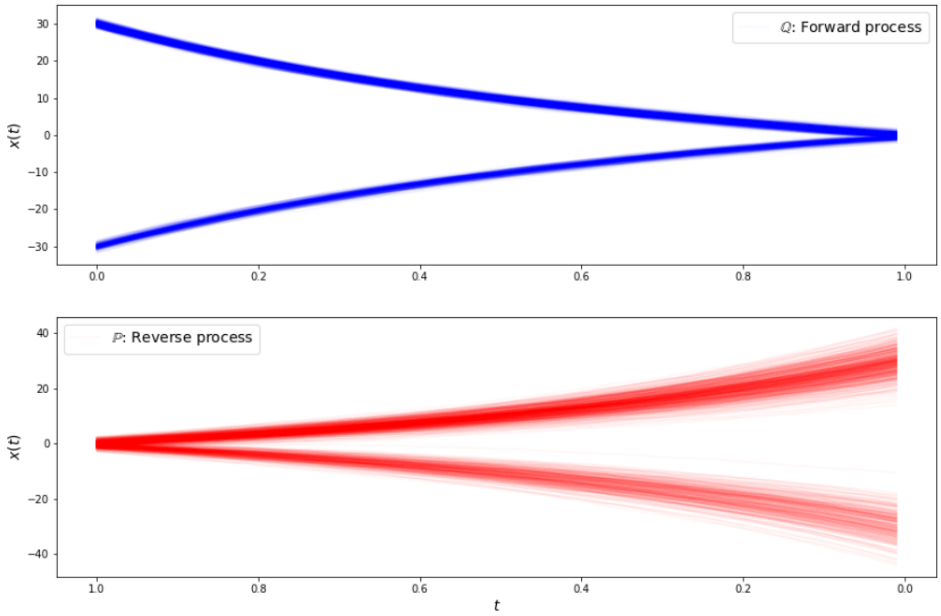

Figure 7.28: Additional bimodal experiment - Fitted SB trajectories using the SC method with ReLu activations, 3 hidden layers of 20 hidden units each.

has a much sharper variance than the actual marginal and seems to be an issue the tanh (and sigmoid and hard-tanh) based activations experience when merging modes. Increasing the number of hidden units improved this effect, however it did not remove it completely.

Hard Tanh

The hard tanh activation is given by:

$$\sigma(x) = \max(-1, \min(1, x)),$$

and it is a piecewise analogue of the tanh function, similar to what the step function is to the logistic sigmoid. We found this function was the most stable when optimising and were able to obtain better results with both deeper and wider networks than with the other activations which became unstable with increased depth and width.

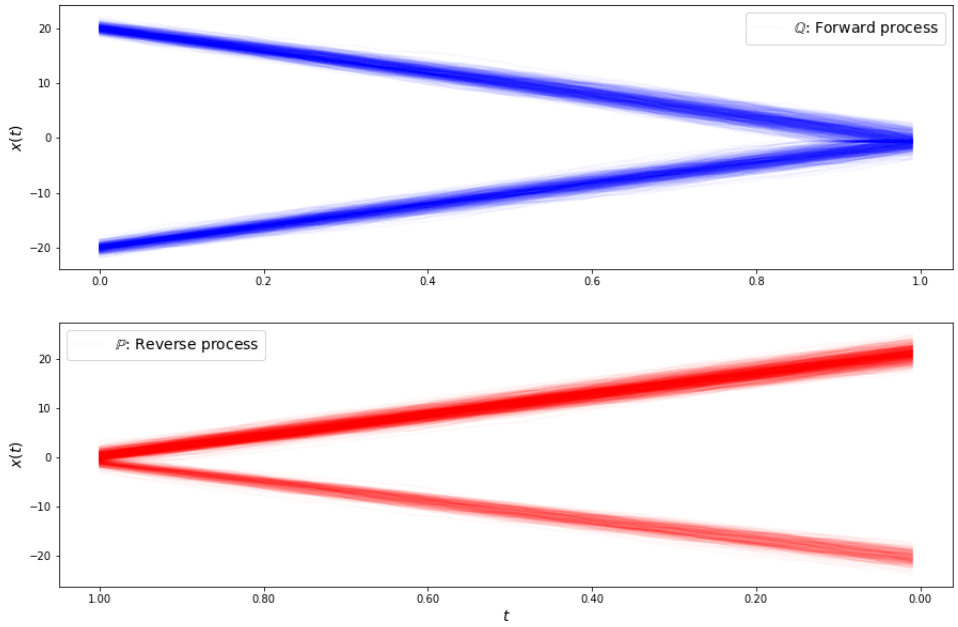

Figure 7.29: Fitted SB trajectories using the SC method with tanh activations, 3 hidden layers of 20 hidden units each.

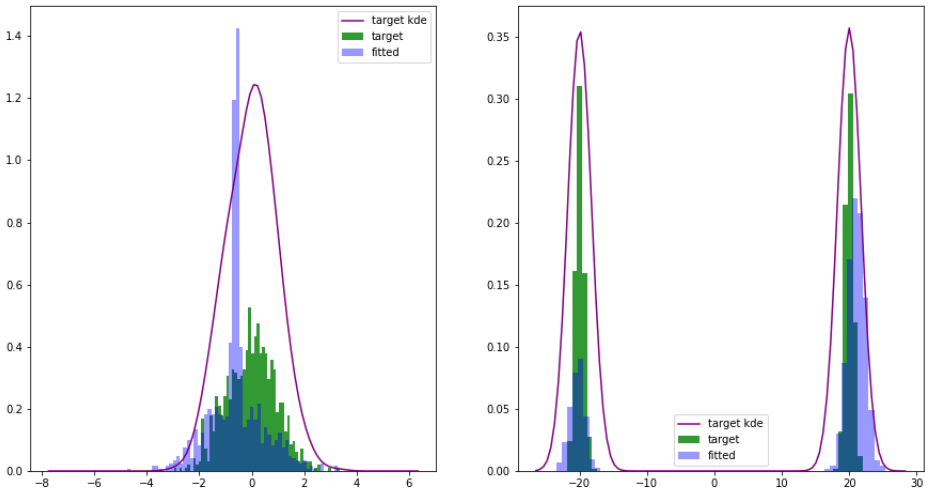

Figure 7.30: Fitted SB boundary distributions using the SC method with tanh activations, 3 hidden layers of 20 hidden units each.

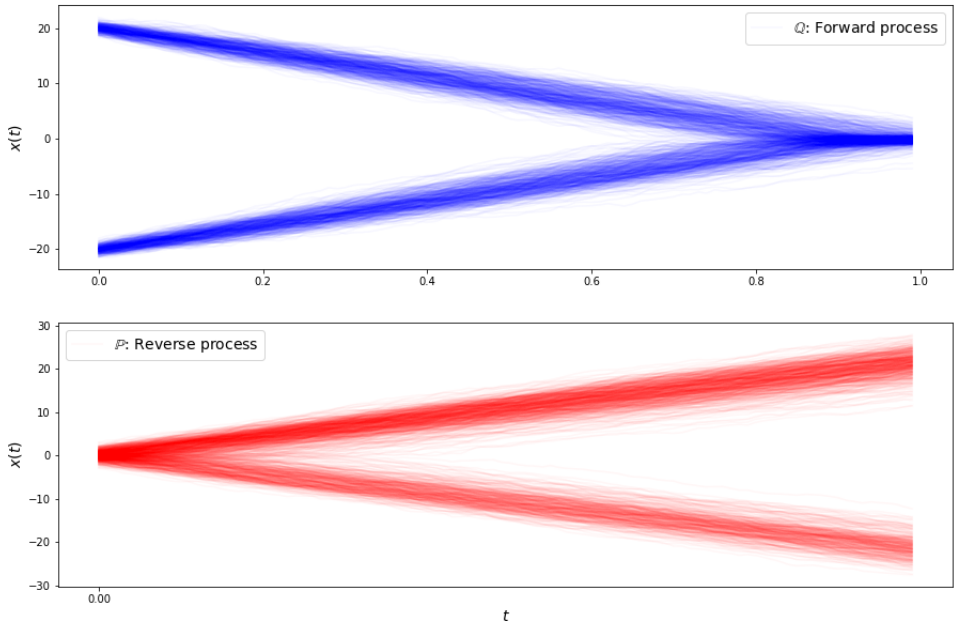

Figure 7.31: Fitted SB trajectories using hard-tanh activation, 2 hidden layers and 200 hidden units per layer.

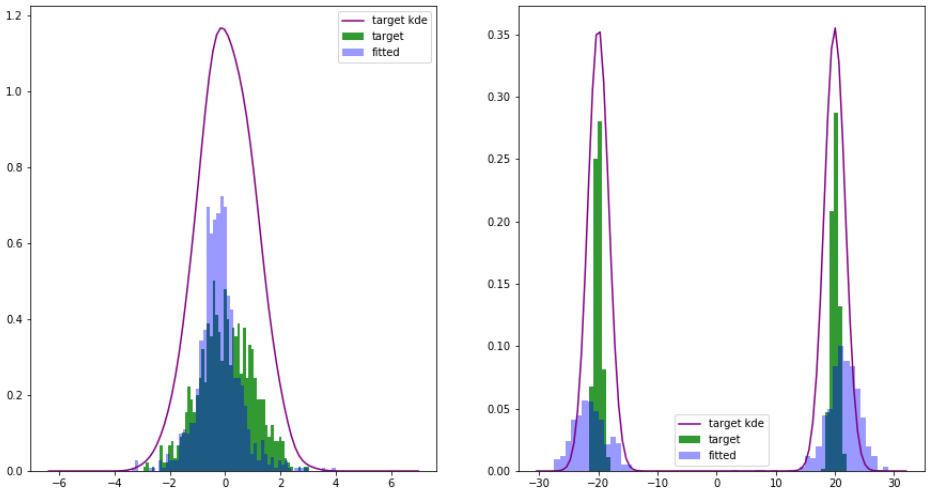

Figure 7.32: Fitted SB boundary distributions using the SC method with hard-tanh activations, 2 hidden layers and 200 hidden units per layer.

As shown in Figures 7.31 and 7.32, the trajectories and results are similar to when using a tanh activation function, however the spike caused when the trajectories merge was less prominent. We believe that increasing the parameters of the network led to this improvement. Unfortunately, we were unable to improve on this issue further. We explored several other activations, with hard-tanh giving the best results when it came to splitting.

An interesting empirical observation is that increasing the depth of the network helped with being able to split modes (which requires learning a piecewise drift), while increasing the width helped with being able to learn mappings from distributions with very different variances. It is possible that learning drifts that are dual to each other, close to Brownian motion (close to 0) and match the boundary distributions is a very challenging task and thus reaches the maximum capacity of the neural network architectures we explored.

Mode Collapse

As mentioned earlier, due to minimising the reverse KL numerically, this method is in theory prone to mode collapse. In practice, mode collapse did occur fairly often, especially when the marginals were not symmetrically positioned relative to each other.

7.4 Comparison of Methods

First, we summarize our findings, highlighting the empirical capacities of the three approaches we trialed. A summary of the results can be found in Table 7.1. The main practical milestones considered when evaluating the methods are:

- **2 Modes:** Being able to split into 2 modes and merge into 1.
- **3 Modes:** Being able to split into 3 modes and merge into one.
- **Non-Gaussian $\boldsymbol{\pi_0, \pi_1}$:** Being able to handle non-Gaussian boundary distributions (moons and circles datasets).

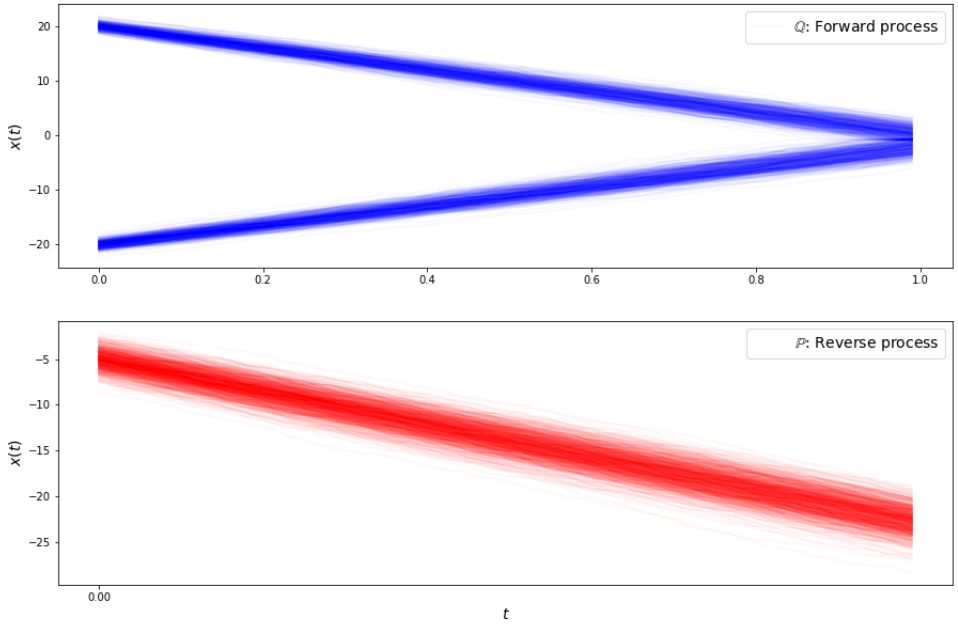

Figure 7.33: Mode collapse example, using the SC method with tanh activations, 3 hidden layers 20 hidden units each.

- **Small Variance:** Being able to match a boundary distribution that has a numerically small variance (e.g. tending towards a point mass).
- **Works at $\gamma = 1$:** Being able to work for a volatility of $\gamma = 1$. We consider it working if it handles at least a radius of 4 between the two distributions, meaning the mode that is most distant to the unimodal distribution has a magnitude of 4. The only method failing this milestone was the method by Pavon et al. (2018) since the method effectively only worked for $\gamma = 1$ when the support of the data was within the unit cube.
- **Distant π_0, π_1:** This is effectively the same milestone as above, however we do not limit the radius to 3. It tests how well the models work when the distributions are very far apart from each other. We found that the method by Pavon et al. (2018) and our direct drift estima-

Table 7.1: Scenarios that each algorithm is able to overcome. The checkmark indicates the method can overcome this task/challenge, whilst a cross indicates it is not able to do so. The exclamation mark indicates in some cases it was able to solve this milestone. The underscores indicates a lack of experiments for the relevant milestone-method pair.

Method	2 Mode Split	3 Mode Split	Non Gaussian π_0, π_1	Small variance	Works at $\gamma = 1$	Distant π_0, π_1
Pavon et al. (2018)	✓	✗	_	✗	✗	✗
Drift Estimation	✓	✓	✓	✗	✓	✗
Stochastic Control	!	✗	_	✗	✓	✓

Table 7.2: Kolmogorov-Smirnov test results on learned boundary distribution. The significance level is set at $\alpha = 0.05$. All methods use $\gamma = 1$ with the exception of the method in Pavon et al. (2018) for which we had to use $\gamma = 100$ for the Unimodal experiment.

Method	Unimodal				Bimodal			
	π_0		π_1		π_0		π_1	
	D_{200}	p-value	D_{200}	p-value	D_{200}	p-value	D_{200}	p-value
Pavon et. al. (2018)	0.150	0.221	0.195	0.001	0.115	0.142	**0.110**	0.178
Direct Drift	**0.070**	0.713	**0.190**	0.001	**0.110**	0.178	0.125	0.088
Stochastic Control	**0.070**	0.713	0.805	$< 10^{-3}$	0.360	$< 10^{-3}$	0.500	10^{-3}

tion approach did not work well in this scenario and required either increasing γ or standardising the data.

Inspecting the summary of findings reported Table 7.1, we can observe that the most robust method is the direct drift estimation approach using GPs. We will now carry out a comparative experiment with a sound hypothesis test set a priori.

Comparative Results

In order to carry out a more quantitative comparison of the methods, we select two of the toy datasets that all methods seem to be able to fit:

$$\pi_0(x) = \mathcal{N}(x; 0, 1)$$
$$\pi_1(y) = \mathcal{N}(y; 4, 0.1^2),$$

and

$$\pi_0(x) = \mathcal{N}(x; 0, 1)$$
$$\pi_1(y) = \frac{1}{2}\mathcal{N}(y; 1.8, 0.6^2) + \frac{1}{2}\mathcal{N}(y; -1.9, 0.6^2).$$

We then perform a Kolmogorov-Smirnov (Kolmogorov-Smirnov et al., 1933) test where the null hypothesis is that the learned marginals from our methods are equal. We set the significance level to its usual setting $\alpha = 0.05$, meaning that if we observe a p-value ≤ 0.05, we reject the null hypothesis that the two distributions are equal. We present results in Table 7.2, displaying both the Kolmogorov test statistic as a goodness of fit measure (the lower the better) across methods and the estimated p-value for the test. We reject the null-hypothesis three times for the stochastic control based method meaning that it had the lowest performance across all methods. We also observe that overall the Direct Drift method and the method by Pavon et al. (2018) are fairly on-par for these two datasets.

Chapter 8

Discussion

Schrödinger bridges have broad applications in physical sciences, but only recently have they sparked interest in the machine learning community. In this work we have aimed to bridge this gap, firstly by trying to present an extensive yet accessible background to the problem, producing proofs for select known results whose original resource we could not find. We have highlighted some of its known connections to the more popular Wasserstein distance and thus we have suggested its potential application in machine learning as a way of comparing distributions. Furthermore, we explored the connection between IPFP and generative adversarial networks, motivating the potential application of using Schrödinger bridges for the task of domain adaptation.

We have learned and illustrated how challenging it is to develop a numerical algorithm for solving the Schrödinger bridge. Part of this challenge comes from having to deal with more than one boundary condition. This is where the g-IPFP framework comes in, allowing us to alternate between subproblems (half bridges) with only one boundary condition until convergence. However, empirically solving these half bridges proved challenging, since they presented harder variants of typical challenges that arise in machine learning, such as numerical integration and density estimation.

We proposed two methods for solving the empirical setting of the Schrödinger bridge and explored them through simple 1D and 2D simulations which allowed us to visually and extensively evaluate the properties of the proposed methods. We compared against the recently proposed method by Pavon et al. (2018) and showed competitive performance across two toy datasets, whilst being more robust to values of γ than the method of (Pavon et al., 2018). In addition, our method was uniquely able to successfully fit a unimodal to trimodal experiment.

At the core of the direct drift estimation approach, we have successfully transformed a complex mathematical problem arising from the physical sciences into a classical machine learning task (regression).

8.1 Summary of Contributions

The key contributions of this thesis are:

- Provided a thorough and accessible introduction to the problem, with motivational examples closer to machine learning as well as intuitive proof sketches for steps that are skipped in the available literature.
- Structured the existing IPFP algorithmic framework and provided formal connections between existing methods, highlighting that they are within the IPFP framework.
- Turned the iterative procedure for solving the Schrödinger bridge into a classical machine learning task at each iteration (i.e. regression). This contribution makes the problem much more accessible to the machine learning community.
- Proposed, implemented and evaluated two new working methods that use common methodology from machine learning. The newly proposed methods overcome some of the conceptual issues in Pavon et al. (2018) (e.g. importance sampling not scaling to high dimensions, instability for low values of γ).

- Using thorough experimentation, we uncovered many of the breaking points of different methods (e.g. mode collapse, sampling between distant boundary distributions), including ours. This is work that was missing from Pavon et al. (2018) and Bernton et al. (2019), thus we provided a holistic view of the methods, helpful for porting the algorithms to practical applications.

8.2 Further Work

Whilst we introduced and evaluated to a great extent the first successful machine learning based approach for solving the Schrödinger bridge, there are still areas of exploration left to pursue further:

- Our best performing method is based on Gaussian processes, which will not scale well to large datasets. Ruttor et al. (2013) further propose a sparse expectation maximisation based algorithm for inferring the drift of SDEs which we did not explore. This could potentially make the DDE method scalable enough for optimising the GP hyperparameters, which we were unable to do due to speed limitations.

- We found that the method by Ruttor et al. (2013) had difficulties in learning drifts whose end point distribution was very sharp (low variance). Optimising the hyperparameters may fix this issue.

- Explore the learning of the drift using a method that is not based on Gaussian processes.

- Some of the unimodal examples admit closed form solutions for the full bridge. We should use this as a ground truth and compute the relative path error to the closed form drift in order to evaluate the methods.

- The direct half bridge drift estimation method we proposed can be adapted to the static Schrödinger bridge. This would be an interesting avenue of research, potentially reducing the computational complexity of the approach.

- We should explore our methods on real world data. One possible task is 2-sample hypothesis tests as in Gretton et al. (2012), where we would use our approach to compare datasets to each other.
- Explore and compare our proposed method with the method in Bernton et al. (2019).

Appendix A

Analysis of Simple Gaussian Like Parametrisation

This appendix illustrates the challenges in parametrising the MLE based objectives proposed in Pavon et al. (2018), such that the partition function can be computed in closed form. Ultimately, we were unable to come up with a parametrisation that had a closed form partition function and was flexible enough to solve the half bridge for multimodal distributions.

A.1 Unimodal Parametrisation

Following the reparametrisation introduced in Section 6.1.2, we consider the case of parametrising the potentials with unimodal Gaussians. This allows for further analysis of the method, since both the normalisation and the updates can be attained in closed form.

Firstly, we offer the counterexample that valid Gaussian like parametrisations for the potentials cannot solve the bridge even for unimodal Gaussian marginals:

Proposition 3. *If we parametrise the potentials as:*

$$\hat{\phi}_0(\boldsymbol{x};\beta) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x), \tag{A.1}$$

$$\phi_1(\boldsymbol{y};\hat{\beta}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y), \tag{A.2}$$

and try to solve the empirical bridge for the marginals:

$$\pi_0(x) = \mathcal{Z}_x^{-1}\mathcal{N}(x, 0, 1),$$
$$\pi_1(y) = \mathcal{Z}_y^{-1}\mathcal{N}(y, 0, 10^2),$$

then the MLE based approximation of Fortet's in Pavon et al. (2018) will not meet the marginal constraints.

Proof. Assuming a prior $\mathbb{W}^\gamma$. Propagating:

$$\phi_0(\boldsymbol{x}; \hat{\beta}) = \mathcal{Z}_x^{-1}\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y + \gamma\mathbb{I}_d),$$
$$\hat{\phi}_1(\boldsymbol{y}; \beta) = \mathcal{Z}_y^{-1}\mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x + \gamma\mathbb{I}_d).$$

Then:

$$\phi_0(\boldsymbol{x}; \hat{\beta})\phi_0(\boldsymbol{x}; \beta) = \mathcal{N}\Big(\boldsymbol{x}\Big|\big(\boldsymbol{\Sigma}_x^{-1} + (\boldsymbol{\Sigma}_y + \gamma\mathbb{I}_d)^{-1}\big)^{-1}\big(\boldsymbol{\Sigma}_x^{-1}\boldsymbol{\mu}_x + (\boldsymbol{\Sigma}_y + \gamma\mathbb{I}_d)^{-1}\boldsymbol{\mu}_y\big)\big),$$
$$\big(\boldsymbol{\Sigma}_x^{-1} + (\boldsymbol{\Sigma}_y + \gamma\mathbb{I}_d)^{-1}\big)^{-1}\Big).$$

Via MLE (holding y-parameters constant) we have:

$$\big(\boldsymbol{\Sigma}_x^{-1} + (\boldsymbol{\Sigma}_y + \gamma\mathbb{I}_d)^{-1}\big)^{-1} = \boldsymbol{S}_x.$$

Yielding update

$$\boldsymbol{\Sigma}_x^{(t)} = (\boldsymbol{S}_x^{-1} - (\boldsymbol{\Sigma}_y^{(t-1)} + \gamma\mathbb{I}_d)^{-1})^{-1},$$

and

$$\big(\boldsymbol{\Sigma}_x + (\boldsymbol{\Sigma}_y + \gamma\mathbb{I}_d)^{-1}\big)^{-1}\big(\boldsymbol{\Sigma}_x^{-1}\boldsymbol{\mu}_x + (\boldsymbol{\Sigma}_y + \gamma\mathbb{I}_d)^{-1}\boldsymbol{\mu}_y\big)\big) = \bar{\boldsymbol{x}}.$$
$$\big(\boldsymbol{\Sigma}_x^{-1}\boldsymbol{\mu}_x + (\boldsymbol{\Sigma}_y + \gamma\mathbb{I}_d)^{-1}\boldsymbol{\mu}_y\big)\big) = \big(\boldsymbol{\Sigma}_x^{-1} + (\boldsymbol{\Sigma}_y + \gamma\mathbb{I}_d)^{-1}\big)\,\bar{\boldsymbol{x}}.$$

Yielding update:

$$\boldsymbol{\mu}_x^{(i)} = \big(\mathbb{I} + \boldsymbol{\Sigma}_x^{(i)}(\boldsymbol{\Sigma}_y^{(i-1)} + \gamma\mathbb{I}_d)^{-1}\big)\,\bar{\boldsymbol{x}} - \boldsymbol{\Sigma}_x^{(i)}(\boldsymbol{\Sigma}_y^{(i-1)} + \gamma\mathbb{I}_d)^{-1}\boldsymbol{\mu}_y^{(i-1)},$$

and similarity for $\boldsymbol{\Sigma}_y, \boldsymbol{\mu}_y$. In 1D (and in terms of the free parameters), the

fitted marginal density variances are:

$$\sigma_y^* = \left(\frac{1}{1+\sigma_x^2} + \frac{1}{\sigma_y^2}\right)^{-1},$$
$$\sigma_x^* = \left(\frac{1}{1+\sigma_y^2} + \frac{1}{\sigma_x^2}\right)^{-1}.$$

If the empirical marginal variances were $1, 10$:

$$1 = \left(\frac{1}{1+\sigma_x^2} + \frac{1}{\sigma_y^2}\right)^{-1},$$
$$10 = \left(\frac{1}{1+\sigma_y^2} + \frac{1}{\sigma_x^2}\right)^{-1}.$$

Then the above system has no valid solutions (can be solved only for negative values of σ_x), by the quadratic formula:

$$0.8\sigma_x^4 + 1.9\sigma_x^2 + 1 = 0. \qquad \lightning$$

□

The above tells us that in order to do a squared exponential paremetrisation for the potentials, we must drop the positive definite constraint. In geometric terms, if one potential is log concave then the other needs to be log convex due to how the variances combine.

We must still enforce an integrability constraint (required to make the inner product of potentials converge):

$$\left(\boldsymbol{\Sigma}_x^{-1} + \boldsymbol{\Sigma}_y^{-1}\right)^{-1} = \boldsymbol{L}\boldsymbol{L}^\top,$$

which implies:

$$\boldsymbol{\Sigma}_x^{-1} + \boldsymbol{\Sigma}_y^{-1} = \boldsymbol{L}\boldsymbol{L}^\top,$$

which can be enforced via a simple reparametrisation (this allows for negative lengthscales):

$$\boldsymbol{\Sigma}_x = (\boldsymbol{L}\boldsymbol{L}^\top - \boldsymbol{\Sigma}_y^{-1})^{-1},$$

where $\boldsymbol{L}$ can be parametrised in terms of an unconstrained vector.

A.2 Mixture of Exponentiated Quadratics

We now consider whether it is possible to extend the exponentiated quadratic parametrisation of the potentials to a mixture of exponentiated quadratics. In order to achieve this parametrisation, we need to reconsider the integrability constraint.

$$\hat{\phi}_0(\boldsymbol{x};\beta) = \sum_{k_x} \pi_{k_x} \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{k_x}, \boldsymbol{\Sigma}_{k_x}) \tag{A.3}$$

$$\phi_1(\boldsymbol{y};\hat{\beta}) = \sum_{k_y} \pi_{k_y} \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_{k_y}, \boldsymbol{\Sigma}_{k_y}) \tag{A.4}$$

The integrability constraint then becomes $\forall k_x$:

$$\bigwedge_{k_y} \left(\boldsymbol{\Sigma}_{k_x} + \boldsymbol{\Sigma}_{k_y} \succ 0\right),$$

which using the eigen-decompositions $\boldsymbol{\Sigma}_{k_x} = \boldsymbol{U}_{k_x}\boldsymbol{D}_{k_x}\boldsymbol{U}_{k_x}^\top$, $\boldsymbol{\Sigma}_{k_y} = \boldsymbol{U}_{k_y}\boldsymbol{D}_{k_y}\boldsymbol{U}_{k_y}^\top$ can be re-expressed as:

$$\lambda_{k_x}^d + \min \lambda_{k_y}^d \geq 0,$$

where d is a specific component / dimension. Thus, we enforce this constraint via the reparametrisation:

$$\lambda_{k_x}^d = \exp(\theta) - \min \lambda_{k_y}^d,$$

where θ is unconstrained. Furthermore, we can parametrise the orthogonal matrices $\boldsymbol{U}_{k_x}$ in terms of unconstrained vectors using Householder reflections. Combining the aforementioned parametrisations, we can enforce the integrability constraint fully via reparametrisations. While these parametrisations are not differentiable everywhere, one can still maximise the liklelihood using the sub-gradient method (Shor, 1991). Unfortunately, this method was unable to work for multimodal distributions, thus we did not pursue this path further.

Bibliography

Alvarez, M., Luengo, D., and Lawrence, N. D. Latent force models. In *Artificial Intelligence and Statistics*, pp. 9–16, 2009.

Álvarez, M. A., Rosasco, L., and Lawrence, N. D. Kernels for vector-valued functions: A review. *Found. Trends Mach. Learn.*, 4(3):195–266, 2012.

Alvarez, M. A., Luengo, D., and Lawrence, N. D. Linear latent force models using Gaussian processes. *IEEE Transactions on pattern analysis and machine intelligence*, 35(11):2693–2705, 2013.

Andres, D. S. Lecture notes in Advanced Probability, Cambridge University, September 2019. URL `http://www.statslab.cam.ac.uk/~sa836/teaching/ap19/\\AdvancedProbability.html`.

Batz, P., Ruttor, A., and Opper, M. Approximate Bayes learning of stochastic differential equations. *Physical Review E*, 98(2):022109, 2018.

Bayes, T. LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S. *Philosophical transactions of the Royal Society of London*, (53):370–418, 1763.

Bernton, E., Heng, J., Doucet, A., and Jacob, P. E. Schrödinger Bridge Samplers. *arXiv preprint arXiv:1912.13170*, 2019.

Carlier, G., Duval, V., Peyré, G., and Schmitzer, B. Convergence of entropic schemes for optimal transport and gradient flows. *SIAM Journal on Mathematical Analysis*, 49(2):1385–1418, 2017.

Chen, Y., Georgiou, T., and Pavon, M. Entropic and displacement interpolation: a computational approach using the Hilbert metric. *SIAM Journal on Applied Mathematics*, 76(6):2375–2396, 2016.

Cramer, E. Probability measures with given marginals and conditionals: I-projections and conditional iterative proportional fitting. *Statistics and*

Decisions-International Journal for Stochastic Methods and Models, 18(3): 311–330, 2000.

Csiszár, I. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pp. 146–158, 1975.

Doob, J. L. The Brownian movement and stochastic equations. *Annals of Mathematics*, pp. 351–369, 1942.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(Jul):2121–2159, 2011.

Einstein, A. On the motion required by the molecular kinetic theory of heat of small particles suspended in a stationary liquid. *Annalen der physik*, 17 (8):549–560, 1905.

Essid, M. and Pavon, M. Traversing the Schrödinger Bridge strait: Robert Fortet's marvelous proof redux. *Journal of Optimization Theory and Applications*, 181(1):23–60, 2019.

Fortet, R. Résolution d'un systeme d'équations de M. Schrödinger. *J. Math. Pure Appl. IX*, 1:83–105, 1940.

Föllmer, H. Random fields and diffusion processes. In *École d'Été de Probabilités de Saint-Flour XV–XVII, 1985–87*, pp. 101–203. Springer, 1988.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

Glorot, X., Bordes, A., and Bengio, Y. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323, 2011.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar): 723–773, 2012.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304, 2010.

Jaynes, E. T. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

Jaynes, E. T. *Probability theory: The logic of science*. Cambridge University press, 2003.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kolmogorov-Smirnov, A., Kolmogorov, A., and Kolmogorov, M. Sulla Determinazione empírica di uma legge di distribuzione. 1933.

Kullback, S. Probability densities with given marginals. *The Annals of Mathematical Statistics*, 39(4):1236–1243, 1968.

Kullback, S. *Information theory and statistics*. Courier Corporation, 1997.

Lähdesmäki, P. H. M. H. H. and Kaski, S. Deep Learning with differential Gaussian process flows.

Lample, G., Conneau, A., Denoyer, L., and Ranzato, M. Unsupervised Machine Translation Using Monolingual Corpora Only. *arXiv preprint arXiv:1711.00043*, 2017.

Lawrence, N. D. *Variational inference in probabilistic models*. PhD thesis, University of Cambridge, 2001.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553): 436–444, 2015.

Léonard, C. From the Schrödinger problem to the Monge–Kantorovich problem. *Journal of Functional Analysis*, 262(4):1879–1920, 2012.

Léonard, C. A survey of the Schrödinger problem and some of its connections with optimal transport. *arXiv preprint arXiv:1308.0215*, 2013.

Levina, E. and Bickel, P. The earth mover's distance is the mallows distance: Some insights from statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pp. 251–256. IEEE, 2001.

Martino, L., Elvira, V., and Louzada, F. Effective sample size for importance sampling based on discrepancy measures. *Signal Processing*, 131:386–401, 2017.

Mikami, T. and Thieullen, M. Optimal transportation problem by stochastic optimal control. *SIAM Journal on Control and Optimization*, 47(3):1127–1139, 2008.

Mohamed, S. and Lakshminarayanan, B. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.

Nagasawa, M. *Stochastic processes in quantum physics*, volume 94. Birkhäuser, 2012.

Nelson, E. *Dynamical theories of Brownian motion*, volume 3. Princeton university press, 1967.

Øksendal, B. Stochastic differential equations. In *Stochastic differential equations*, pp. 65–84. Springer, 2003.

Osborne, M., Garnett, R., Ghahramani, Z., Duvenaud, D. K., Roberts, S. J., and Rasmussen, C. E. Active learning of model evidence using Bayesian quadrature. In *Advances in Neural Information Processing Systems*, pp. 46–54, 2012.

Pavon, M. and Wakolbinger, A. On free energy, stochastic control, and Schrödinger processes. In *Modeling, Estimation and Control of Systems with Uncertainty*, pp. 334–348. Springer, 1991.

Pavon, M., Tabak, E. G., and Trigila, G. The data-driven Schrödinger bridge. *arXiv preprint arXiv:1806.01364*, 2018.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Ramdas, A., Trillos, N. G., and Cuturi, M. On Wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, 19(2):47, 2017.

Ruschendorf, L. et al. Convergence of the iterative proportional fitting procedure. *The Annals of Statistics*, 23(4):1160–1174, 1995.

Ruttor, A., Batz, P., and Opper, M. Approximate Gaussian process inference for the drift function in stochastic differential equations. In *Advances in Neural Information Processing Systems*, pp. 2040–2048, 2013.

Salamon, D. A. Lecture notes in measure and integration, eth, June 2019. URL `https://people.math.ethz.ch/~salamon/PREPRINTS/measure.pdf`.

Särkkä, S. and Solin, A. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.

Schrödinger, E. Uber die umkehrung der naturgesetze, Verlag Akademie der wissenschaften in kommission bei Walter de Gruyter u. *Company, Berlin*, 1931.

Schrödinger, E. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. In *Annales de l'institut Henri Poincaré*, volume 2, pp. 269–310, 1932.

Shor, N. Z. The development of numerical methods for nonsmooth optimization in the USSR. *History of Mathematical Programming. A Collection of Personal Reminiscences*, pp. 135–139, 1991.

Silverman, B. W. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

Singh, S. and Póczos, B. Analysis of k-nearest neighbor distances with application to entropy estimation. *arXiv preprint arXiv:1603.08578*, 2016.

Tzen, B. and Raginsky, M. Neural stochastic differential equations: Deep latent Gaussian models in the diffusion limit. *arXiv preprint arXiv:1905.09883*, 2019.

Van der Wilk, M. *Sparse Gaussian process approximations and applications*. PhD thesis, University of Cambridge, 2019.

Vargas, F., Brestnichki, K., and Hammerla, N. Model Comparison for Semantic Grouping. In *International Conference on Machine Learning*, pp. 6410–6417, 2019.

Veksler, O. Nonparametric density estimation nearest neighbors, KNN, 2013.

Viaclovsky, P. J. Lecture notes in measure and integration, mit, September 2003. URL `https://ocw.mit.edu/courses/mathematics/18-125-measure-and-\\integration-fall-2003/`.

Villani, C. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.

Williams, C. K. and Rasmussen, C. E. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Yang, X. Understanding the variational lower bound, 2017.

Zhang, M., Bird, T., Habib, R., Xu, T., and Barber, D. Variational f-divergence Minimization. *arXiv preprint arXiv:1907.11891*, 2019.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of*

the IEEE international conference on computer vision, pp. 2223–2232, 2017.