THE UNIVERSITY *of* EDINBURGH

Stochastic modelling and inference of ocean transport

Martin Thomas Brolly

Doctor of Philosophy
University of Edinburgh
July, 2023

Declaration

I declare that this thesis was composed by myself and that the work contained therein is my own, except where explicitly stated otherwise in the text.

(Martin Thomas Brolly)

Acknowledgements

I owe a debt of gratitude to all those who supported me in completing my PhD and in getting there in the first place.

I thank my primary supervisor, Jacques Vanneste, for his inspiring mentorship. I am grateful for everything that he has taught me. I also thank my other supervisors, Aretha Teckentrup and James Maddison, for their advice and support. It has been a privilege to learn from all of you.

I would like to thank my family for supporting my education. The freedom to study is a luxury that I do not take for granted.

Finally, I thank Karolina for making these years so much sweeter.

Martin Thomas Brolly was supported by the EPSRC Centre for Doctoral Training in Mathematical Modelling, Analysis and Computation (MAC-MIGS) funded by the UK Engineering and Physical Sciences Research Council (grant EP/S023291/1), the University of Edinburgh and Heriot-Watt University.

Dla Karoliny.

Lay summary

The motion of the ocean is notoriously difficult to observe. While satellites capture only the large-scale surface behaviour, a unique view is obtained from satellite-tracked drifting buoys known as drifters. These devices are released at sea and transported by ocean currents, experiencing the full complexity of ocean transport. The meandering trajectories that they follow can be studied to infer the statistics of ocean dynamics. This thesis is devoted to methods for carrying out this inference. A particular focus is on quantifying the uncertainty in our inferences. The distribution of drifters throughout the oceans is highly nonuniform, meaning that in some areas we have much less information than we need to make inferences with confidence.

The first problem we consider is model comparison. Models of ocean transport are typically designed based on physical intuition, and are not unique. Hence, a method is needed to choose between them. We apply a method known as Bayesian model comparison to assess the relative performance of existing models of transport in an idealised model of turbulence. Our comparison is based on simulated trajectory data. We show that on different timescales the preference of models switches, with a simpler model preferred at longer timescales. We demonstrate the method in an idealised setting, but it can be applied also to more complex problems, including models of ocean drifters.

We then develop a novel model of drifter motion based on probabilistic neural networks and informed by observations of the real ocean. Our model predicts the probability of the future position of a fluid particle released at the ocean surface. It outperforms existing models and reproduces clustering behaviour found in previous studies on so-called garbage patches. It also provides a convenient means of estimating a range of dynamical statistics from sparse drifter trajectories.

Finally, we turn to the subject of uncertainty quantification for probabilistic neural networks. Our aim is to assess the suitability of so-called Bayesian neural networks to problems in ocean transport. The construction of Bayesian neural networks is challenging, and state-of-the-art methods compromise on rigour in favour of computational efficiency. We find that current methods are not sufficiently robust when applied to our drifter model.

Abstract

Inference of ocean dynamical properties from observations requires a suite of statistical tools. In this thesis we assemble and develop a selection of useful methods for oceanographic inference problems. Our work is centred around the modelling of ocean transport. We consider Lagrangian observations, including those obtained from surface drifters. We adopt a Bayesian approach which offers a coherent framework for diagnosing and predicting ocean transport and enables principled uncertainty quantification. We also emphasise the role of stochastic models.

We begin with the problem of comparing stochastic models on the basis of observations. We apply Bayesian model comparison to classical stochastic differential equation models of turbulent dispersion given trajectory data generated by simulation of particles in an idealised forced–dissipative model of two-dimensional turbulence. We discuss how model preference is quantifiably sensitive to the timescale on which the models are applied. The method is widely applicable and accounts for uncertainty in model parameters.

We then consider purely data-driven models for particle dynamics. In particular we build a probabilistic neural network model of the single-particle transition density given observations from the Global Drifter Program. The transition density model can be used either to emulate surface transport, by modelling trajectories as a discrete-time Markov process, or to estimate spatially-varying dynamical statistics including diffusivity. As is standard for probabilistic neural networks we train our model to maximise the likelihood of data. The model outperforms existing stochastic models, as assessed by skill scores for probabilistic forecasts, and is better able to deal with non-uniform data than standard methods.

A weakness of our transition density model is that, since it is trained by maximum likelihood rather than Bayesian inference, its predictions come without uncertainty quantification. This is especially concerning in regions where little data is available and point estimates of statistics such as diffusivity cannot be trusted. With this motivation we discuss state-of-the-art methods in approximate Bayesian inference and their effectiveness in building Bayesian neural networks. We highlight deficiencies in current methods and identify the key challenges in providing uncertainty quantification with neural network models. We illustrate these issues both in a simple one-dimensional problem and in a Bayesian version of our transition density model.

Contents

Chapter 1

Introduction

The state of the Earth as we know it is sustained by the dynamics of the oceans. By way of their turbulent motions, they circulate and disperse heat, carbon, nutrients and more throughout the world in vast quantities. This redistribution of heat and material regulates the planet in crucial ways. But these dynamics and the transport processes they induce are subtle. They involve complex interactions between motions that span disparate scales in space and time. The ocean thus stands as a prototypical chaotic system. It is due to this, along with practical concerns such as understanding the climate, that the ocean wins the attention of many mathematicians.

Another facet of the ocean complicates matters further: it is dark. Not only to our eyes, but also to the satellites and other devices that probe the atmosphere, the ocean is opaque, so that only its surface may be observed remotely. Thus, to study the ocean at depth and at scales smaller than the resolution of satellite imagery, oceanographers are forced to rely on limited in situ observations.

With observations limited we turn to modelling the oceans, both to uncover the mechanisms that drive their evolution, and to predict their future state. While the equations that govern fluid motion (the Navier–Stokes equations) are known, we face two critical issues. Namely, (i) that the solution to these equations is highly sensitive to the prescribed initial state, which we cannot measure with sufficient accuracy, and (ii) that the solution is incredibly expensive to compute. With the best computers available today the highest resolution ocean and climate models available fail to resolve the full complexity of ocean dynamics.

Given limited modelling capabilities and sparse data we have two options. We can dispense of the Navier–Stokes equations and employ so-called diagnostic models, informed by physical intuition and the observations we have, to obtain a simplified description — we refer to this process as surrogate modelling. Alternatively, we can use observational data to learn corrections to the coarsely resolved ocean models which solve the Navier–Stokes equations approximately — this process is known to oceanographers as parameterisation and to others as reduced order modelling or closure modelling. Often these two approaches are linked in that diagnostic models inform the development of parameterisations. For both problems matters are compli-

cated further when data is incomplete, indirect or corrupted by noise. Fundamentally, both of these tasks, when observational data are used, are statistical inference problems. More precisely the development of a surrogate model, or a closure, is an inverse problem.

The goal of an inverse problem is to infer from observational data something about the system that generates it. A classical example is the problem of inferring the shape of a drum from the sound it makes; others are the various forms of medical and geophysical imaging. The study of inverse problems is a field in its own right and occupies many mathematicians and statisticians, who ask questions like: when does an inverse problem have a (unique) solution? Or, how should data be collected in order to learn as much as possible about the system we are interested in? When the system in question is complicated, the tools required to solve inverse problems become increasingly sophisticated. Efficient algorithms and surrogate models with high accuracy need to be developed.

In solving inverse problems a classical approach is to construct a best guess at the answer that is considered the most likely solution. However, there is increasing acknowledgement that in any inference procedure we should quantify the uncertainty in the solution. In most cases a finite number of observations contains finite information, and does not constrain the solution to a unique value. A range of solutions can remain plausible. Bayesian statistics provides a language for dealing with uncertainty in this context, whereby probabilities are assigned to the range of possible solutions according to how plausible they are. In this thesis we bring a Bayesian approach to the topic of surrogate modelling of ocean transport. Ocean modelling problems have many of the features that make for challenging inverse problems. We present a range of advanced methods that we believe can support the oceanographer in making inferences from observations.

The first topic we consider is model comparison. Given a set of candidate models and a set of observations, we discuss how a model can be chosen on the basis that it is most plausible in light of all available information. A solution is generically provided by Bayesian statistics, namely Bayesian model comparison, which amounts to assigning probabilities to models. In Chapter 2 we apply this method to compare stochastic differential equation models of the dispersion of fluid particles given trajectory data generated through simulations of idealised two-dimensional turbulence. We show that the relative performance of these surrogate models is sensitive to the timescale of interest. By demonstrating the effectiveness of Bayesian model comparison in a canonical fluid dynamical problem, we show that this method, rarely employed in a dynamical context, is applicable to a range of problems in fluid dynamics and oceanography. This work was published as Brolly et al. (2022).

The models considered in Chapter 2, although guided by physical intuition, ultimately depend on statistical assumptions. As part of a trend towards increasing acceptance of purely data-driven modelling in physical problems, there has been a surge in the use of advanced statistical models for a range of surrogate and reduced-order modelling tasks in the Earth sciences. In particular, the use of models based on

artificial neural networks has become widespread. Born out of fields such as computer vision and speech recognition, neural networks have proliferated to become widely used in a large range of fields. The popularity of neural networks is due to their strong predictive performance in a large range of regression and classification problems. In Chapter 3 we construct a probabilistic neural network model of the single-particle transition density of near-surface ocean dynamics. The essence of our model is that displacements are modelled by a Gaussian mixture model, whose parameters are functions of initial position, and these functions are represented by a neural network. The parameters of our model are chosen to maximise the likelihood of trajectory data collected by satellite-tracked drifting buoys known as drifters. We find that our model not only outperforms simpler surrogate models, but also provides a convenient means of estimating simultaneously a range of dynamical statistics such as diffusivity from highly nonuniform data. This work is published as Brolly (2023).

In deviating from the Bayesian paradigm and adopting instead a maximum likelihood approach, the results in Chapter 3 lack uncertainty quantification. In Chapter 4 we consider a Bayesian version of our neural network model. The application of Bayesian inference to neural networks is a challenge at the leading edge of machine learning research. The challenge is due primarily to the large number of parameters. The sampling methods typically used for Bayesian inference procedures fail in this regime and advanced methods for approximate inference are needed. We discuss this challenge and others, first in the context of a synthetic problem, and then in the case of the transition density model. We show that current state-of-the-art methods fail to meet the standards expected for inference in the scientific domain.

In Chapter 5 we give our concluding remarks and discuss possible directions of future research.

1.1 Notation and formalism

We give here some clarifying comments on the notation and formalism used throughout the thesis.

Random variables, measures and densities

For ease of notation we generally avoid the need to distinguish a random variable from the argument of its probability density function. For example, we will write $p(X)$ to denote the value of the probability density function of a random variable X at the value X (overloading notation), rather than the more explicit notation $p_X(x)$ where the subscript indicates the measure the density relates to and the argument (now with its own distinct label x) indicates the value at which the density is evaluated. Similarly we will write $\mathbb{E}[f(X)]$ to denote the expectation of a function of a random variable X with respect to the obvious measure, rather than the more explicit $\mathbb{E}_X[f(X)]$. One exception is in the proof of Lemma 1, where the more explicit notation is required.

Formal assumptions

We will regularly take for granted the existence of derivatives, integrals (including expectations), and probability densities with respect to the Lebesgue measure. Where any such objects are referenced in the text, their existence is tacitly assumed. Integrals are, where appropriate, further assumed to be finite. These assumptions are uncontroversial in the context of ocean transport modelling, but may not be in other areas where the methods considered here may be applied.

Chapter 2

Bayesian comparison of stochastic models of dispersion

2.1 Introduction

Since Taylor introduced the notion of turbulent diffusion in the 1920s (Taylor 1922), a wide variety of stochastic models have been proposed to represent the dynamics of particles in turbulent flows (e.g. Thomson 1987, Rodean 1996, Majda & Kramer 1999, Berloff & McWilliams 2002). The Brownian dynamics used by Taylor models Lagrangian velocities as white noise processes and is a good approximation only on sufficiently long time scales. More complex models incorporate temporal and/or spatial correlation (e.g. Griffa 1996, Pasquero et al. 2001, Lilly et al. 2017). For example, Langevin dynamics incorporate autocorrelation in Lagrangian velocities by representing them as Ornstein–Uhlenbeck processes (Uhlenbeck & Ornstein 1930). It is in general unclear when such additional complexity leads to improved predictions rather than to overfitting. Given the increased difficulty and cost of implementing more complex models, a method for comparing the performance of competing stochastic models for particle dynamics is needed.

To this end, we propose a data-driven approach: we apply *Bayesian model comparison* (BMC) (Jaynes 2003, Kass & Raftery 1995, MacKay 2003), which assigns probabilities to competing models based on their ability to explain observed data. We focus on the comparison between the Brownian and Langevin models for particles in two-dimensional homogeneous isotropic turbulence, with data that consists of sequences of particle positions obtained from simulated Lagrangian trajectories. While this set-up is highly idealised, the methodology developed is applicable to more complex flows and models of particle dynamics.

Model comparison is complicated by two issues: (i) proposed models typically contain a number of parameters whose values are uncertain, and (ii) a measure of model suitability is required, balancing accuracy and complexity. The natural language for this problem is then that of decision theory (see e.g. Bernardo & Smith (1994) and Robert (2007) for an overview of decision problems under uncertainty);

however, several philosophical issues therein, such as the choice of utility function and its subjectivity, can be avoided by adopting the ready-made approach of BMC. BMC and the related technique Bayesian model averaging are gaining popularity in many applied fields (Mark et al. 2018, Min et al. 2007, Carson et al. 2018, Mann 2011). In this paper, we demonstrate the potential of BMC by comparing the Brownian and Langevin models of dispersion in two-dimensional turbulence. This provides a simple illustration of the BMC methodology while addressing a problem of interest: dispersion in two-dimensional turbulence has received much attention as a paradigm for transport and mixing in stratified, planetary-scale geophysical flows (Provenzale et al. 1995), and can be modelled with stochastic processes (e.g. Pasquero et al. 2001, Lilly et al. 2017).

The paper is structured as follows. We introduce the Brownian and Langevin models in §2.2 and review the BMC method in §2.3. In §2.4 we show how this method can be applied to discrete particle trajectory data; we also show results of a test case, where the data are generated by the Langevin model itself. In §2.5 we apply BMC to data from direct numerical simulations of two-dimensional turbulence. In §2.6 we give our conclusions on the method.

2.2 Models and data

2.2.1 Brownian and Langevin models

The models of interest are the *Brownian model*, which for passive particles in homogeneous and isotropic turbulence is given by

$$\mathrm{d}\boldsymbol{X} = \sqrt{2\kappa}\,\mathrm{d}\boldsymbol{W}, \tag{2.1}$$

with $\kappa > 0$, and the *Langevin model*, which, under the same conditions, is given by

$$\mathrm{d}\boldsymbol{X} = \boldsymbol{U}\,\mathrm{d}t, \tag{2.2a}$$

$$\mathrm{d}\boldsymbol{U} = -\gamma\boldsymbol{U}\,\mathrm{d}t + \gamma\sqrt{2k}\,\mathrm{d}\boldsymbol{W}, \tag{2.2b}$$

with γ, $k > 0$, and where, in both cases, $\boldsymbol{W}$ is a vector composed of independent Brownian motions. We denote the models by $\mathcal{M}_B(\kappa)$ and $\mathcal{M}_L(\gamma, k)$.

We note some important characteristics of the two models. The Brownian model involves particle position, $\boldsymbol{X}$, as its only component, which evolves as a scaled d-dimensional Brownian motion, where d is the number of spatial dimensions. This implies that particle velocity evolves as a white noise process. The model has one parameter, the diffusivity κ. The Brownian model aims to represent dynamics which are effectively diffusive on the timescales of interest, as opposed to molecular diffusion alone. The validity of (2.1) is typically justified by arguments involving strong assumptions of scale separation between mean flows and small-scale fluctuations which rarely hold in applications (Majda & Kramer 1999, Berloff & McWilliams 2002).

The Langevin model, by contrast, involves two components, particle position and particle velocity, $(\boldsymbol{X}, \boldsymbol{U})$. The velocity component evolves according to a mean-zero Ornstein–Uhlenbeck process, and position results from time integration of this velocity. The model has two parameters, γ and k, where γ^{-1} is a Lagrangian velocity decorrelation time and k characterises the strength of Gaussian velocity fluctuations. The Brownian and Langevin models are the first two members of a hierarchy of Markovian models involving an increasing number of time derivatives of the position (Berloff & McWilliams 2002). All such models should be understood as surrogate models, aiming to capture emergent statistics, rather than physical models derived from the laws of motion.

In practice, the Brownian model is favoured over the Langevin model for its simplicity as well as for the practical virtue of having a smaller, more-easily-explored, one-dimensional parameter space. Note that if these models are to be implemented in the limit of continuous concentrations of particles then it is their corresponding Fokker–Planck equations which must be solved — this means solving partial differential equations in $d + 1$ or $2d + 1$ dimensions, respectively.

Both the Brownian and Langevin model can be extended to account for spatial anisotropy, inhomogeneity and the presence of a mean flow, at the cost of increasing the dimension of their parameter spaces; full details are given in Berloff & McWilliams (2002). Brownian and Langevin dynamics underlie the so-called random displacement and random flight models used for dispersion in the atmospheric boundary layer (Esler & Ramli 2017), and have been applied to the simulation of ocean transport, as models of mixing in the horizontal (Berloff & McWilliams 2002), vertical (Onink et al. 2022), and on neutral surfaces (Reijnders et al. 2022). Ying et al. (2019) showed how Bayesian parameter inference can be applied to the Brownian model in the inhomogeneous setting using Lagrangian trajectory data. We restrict attention to isotropic turbulence in this work for simplicity, noting that the methods demonstrated below are equally applicable in the more general case.

2.2.2 Data

For our comparison we consider trajectory data of the form

$$\left\{\left(\boldsymbol{X}_0^{(p)}, \cdots, \boldsymbol{X}_{N_\tau}^{(p)}\right) : p \in \left\{1, \cdots, N_p\right\}\right\}, \tag{2.3}$$

where $\boldsymbol{X}_n^{(p)}$ is the position of particle p at time $t = n\tau$. In words, we observe the positions of a set of N_p particles at $N_\tau + 1$ times separated by uniform time intervals of length τ, which we refer to as the sampling time. The performance of the models depends crucially on τ. Since both models are uncorrelated in space, we can rewrite the observations as the set of displacements

$$\Delta\mathcal{X}_\tau = \left\{\left(\Delta\boldsymbol{X}_0^{(p)}, \cdots, \Delta\boldsymbol{X}_{N_\tau-1}^{(p)}\right) : p \in \left\{1, \cdots, N_p\right\}\right\}, \tag{2.4}$$

where $\Delta \boldsymbol{X}_n^{(p)} = \boldsymbol{X}_{n+1}^{(p)} - \boldsymbol{X}_n^{(p)}$.

In §2.4 we consider the case that the trajectory data are generated by Langevin dynamics, while in §2.5 we compare the Brownian and Langevin models given data from direct numerical simulations of a forced-dissipative model of stationary, isotropic two-dimensional turbulence. In both cases we consider observations without noise.

2.3 Methods

In this work we appeal to the Bayesian interpretation of probability and statistics. This means that probabilities reflect levels of plausibility in light of all available information. In particular, we deal with uncertainty in both the parameters of each model and the models themselves by assigning probabilities to them. We outline this procedure in §§2.3.1 and 2.3.2.

2.3.1 Parameter inference

The goal of parameter inference is to infer the values of the parameters $\boldsymbol{\theta} \in \Theta$ of a statistical model, say $\mathcal{M}(\boldsymbol{\theta})$, given observational data $\mathcal{D}$. A model is characterised completely by its likelihood function $p(\,\cdot \mid \mathcal{M}(\boldsymbol{\theta}))$ which denotes the probability (density) of observations under $\mathcal{M}(\boldsymbol{\theta})$. Bayesian inference requires the specification of one's belief prior to observations through a prior distribution $p(\boldsymbol{\theta} \mid \mathcal{M})$. One can then invoke Bayes' Theorem, (2.5), to update this belief in light of the observations. This results in a posterior distribution

$$\overbrace{p(\boldsymbol{\theta} \mid \mathcal{D}, \mathcal{M})}^{\text{Posterior}} = \frac{\overbrace{p(\mathcal{D} \mid \mathcal{M}(\boldsymbol{\theta}))}^{\text{Likelihood}} \overbrace{p(\boldsymbol{\theta} \mid \mathcal{M})}^{\text{Prior}}}{\underbrace{p(\mathcal{D} \mid \mathcal{M})}_{\text{Evidence}}}, \tag{2.5}$$

which denotes the probability (density) of each $\boldsymbol{\theta} \in \Theta$ given observations and prior knowledge (Jeffreys 1983). The posterior fully describes the uncertainty in the inferred parameters, in our case $\boldsymbol{\theta} = \kappa$ or $\boldsymbol{\theta} = (k, \gamma)$. In applications where point estimates of the parameters are required, these can be taken as e.g. the mean or mode of the posterior.

2.3.2 Model inference

Beyond parameter inference we can also make inferences when the model itself, $\mathcal{M}$, is considered unknown. However, in order to meaningfully assign probabilities to models we must assume that the set of models under consideration, $M = \{\mathcal{M}_i\}_{i=1}^{N_m}$, includes all plausibly true models. That is, for any $\mathcal{M}^* \notin M$, $p(\mathcal{M}^*) = 0$. This is known as the $\mathcal{M}$-closed regime (see Chapter 6 of Bernardo & Smith (1994) or Clyde

& Iversen (2013)). In situations where all models under consideration are known to be false this assumption appears dubious; however, we note that the same fallacy is committed in Bayesian parameter inference when we assign probabilities to the parameters of a parametric model which we know is imperfect, i.e. false. In the $\mathcal{M}$-closed regime one assigns prior probabilities to models such that $\sum_{i=1}^{N_m} p(\mathcal{M}_i) = 1$. This allows us to again invoke Bayes' Theorem in the form

$$p(\mathcal{M} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \mathcal{M})\, p(\mathcal{M})}{p(\mathcal{D})}. \tag{2.6}$$

If $\mathcal{M}_i$ is parametric with parameters $\boldsymbol{\theta}_i \in \Theta_i$, $p(\mathcal{D} \mid \mathcal{M}_i)$ is given by

$$p(\mathcal{D} \mid \mathcal{M}_i) = \int_{\Theta_i} p(\mathcal{D} \mid \mathcal{M}_i(\boldsymbol{\theta}_i))\, p(\boldsymbol{\theta}_i \mid \mathcal{M}_i)\, \mathrm{d}\boldsymbol{\theta}_i, \tag{2.7}$$

which is known as the model evidence (or marginal likelihood, or model likelihood) of $\mathcal{M}_i$.

An important property of the evidence is that it accounts for parameter uncertainty. Considering the likelihood as a score of model performance given some fixed parameter values, the evidence can be viewed as an expectation of that score with respect to the prior measure on parameters. In this way the evidence favours models where observations are highly probable for the range of parameter values considered plausible a priori. In particular, this means that a model with many parameters which achieves a very high value of the likelihood only for a narrow range of parameter values which could not be predicted a priori is not likely to attain a higher value of the evidence than a model with fewer parameters whose values are better constrained by prior information. This apparent penalty is usually quantified by the so-called *Occam (or Ockham) factor*, named in reference to Occam's razor,

$$\mathrm{Occam}_i = p(\mathcal{D} \mid \mathcal{M}_i) \,/\, p(\mathcal{D} \mid \mathcal{M}_i(\boldsymbol{\theta}_i^*)) \in [0,\, 1], \tag{2.8}$$

where $\boldsymbol{\theta}_i^*$ is the posterior mode of $\boldsymbol{\theta}_i$ (Jaynes 2003, MacKay 2003).

Given two models, $\{\mathcal{M}_0,\ \mathcal{M}_1\}$, a test statistic for the hypotheses

$$\begin{cases} \mathcal{H}_0 : & \mathcal{M}_0 \text{ is the true model,} \\ \mathcal{H}_1 : & \mathcal{M}_1 \text{ is the true model,} \end{cases}$$

is given by the Bayes factor (Kass & Raftery 1995),

$$K_{1,0} = \frac{p(\mathcal{D} \mid \mathcal{M}_1)}{p(\mathcal{D} \mid \mathcal{M}_0)}, \tag{2.9}$$

where a large value of $K_{1,0}$ represents statistical evidence against $\mathcal{H}_0$.

The log-evidence is exactly equal to the log score (Gneiting & Raftery 2007), also known as the ignorance score (Bernardo 1979, Bröcker & Smith 2007), for probabilis-

tic forecasts. Therefore, the log Bayes factor can be understood as a difference of scores for probabilistic models. Merits of the log score have been appreciated since at least the 1950s (Good 1952), including its intimate connection with information theory (Roulston & Smith 2002, Du 2021). This interpretation of the Bayes factor does not rely on the assumption of the $\mathcal{M}$-closed regime. In what follows we use the Bayes factor to compare the Brownian and Langevin models.

A useful approximation for the evidence (2.7) is given by Laplace's method: a Gaussian approximation of the unnormalised posterior, $p_u(\boldsymbol{\theta}) = p(\mathcal{D} \mid \mathcal{M}(\boldsymbol{\theta}))\, p(\boldsymbol{\theta} \mid \mathcal{M})$, is obtained from a quadratic expansion of $\ln p_u$ about the posterior mode, $\boldsymbol{\theta}^*$,

$$\ln\left(p_u(\boldsymbol{\theta})\right) \approx \ln\left(p_u(\boldsymbol{\theta}^*)\right) - \frac{1}{2}\left(\boldsymbol{\theta} - \boldsymbol{\theta}^*\right)^{\mathrm{T}} J \left(\boldsymbol{\theta} - \boldsymbol{\theta}^*\right), \tag{2.10}$$

where

$$J_{ij} = -\frac{\partial^2}{\partial\boldsymbol{\theta}_i \partial\boldsymbol{\theta}_j} \ln p_u(\boldsymbol{\theta})\bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}. \tag{2.11}$$

Taking an exponential of (2.10) we recognise that we have approximated $p_u(\boldsymbol{\theta})$ with the probability density function (up to a known normalisation) of a Gaussian random variable with mean $\boldsymbol{\theta}^*$ and covariance J^{-1}, so (2.7) becomes

$$\underbrace{p(\mathcal{D} \mid \mathcal{M}_i)}_{\text{Evidence}} \approx \underbrace{p(\mathcal{D} \mid \mathcal{M}_i(\boldsymbol{\theta}_i^*))}_{\text{Maximum likelihood}} \times \underbrace{p(\boldsymbol{\theta}_i^* \mid \mathcal{M}_i)\left(\det(J \,/\, 2\pi)\right)^{-\frac{1}{2}}}_{\text{Occam factor}}. \tag{2.12}$$

This approximation is accurate for a large number of data points $N_p \times N_\tau$ where a Bernstein–von Mises theorem can be shown to hold, guaranteeing asymptotic normality of the posterior measure (Vaart 1998).

We highlight that a model's evidence is sensitive to the prior distribution on the parameters, $p(\boldsymbol{\theta} \mid \mathcal{M})$. This is entirely in the spirit of Bayesian statistics in that a parametric model accompanied with the prior uncertainty on its parameters constitutes a single, complete hypothesis for explaining observations. The evidence for a model is less when the mass of prior probability on parameters is less concentrated on those values for which the likelihood is largest.

2.3.3 Alternative methods for model comparison

In the Bayesian framework BMC is the natural choice of method for model comparison. However, alternative methods could be used. A particularly popular frequentist approach is to score models based on the Akaike information criterion (AIC) (Akaike 1998)

$$\mathrm{AIC}(\mathcal{M}_i) = 2\dim(\boldsymbol{\theta}_i) - 2\ln p(\mathcal{D} \mid \mathcal{M}_i(\widehat{\boldsymbol{\theta}}_i)), \tag{2.13}$$

where $\widehat{\boldsymbol{\theta}}_i$ is the value of $\boldsymbol{\theta}_i$ which maximises the likelihood. Lower values of AIC are preferred, meaning that a model is rewarded for fitting the data well with its maximum likelihood parameter values and is penalised according to the number of unknown parameters. Another commonly used measure of model performance is the Bayesian information criterion (BIC) (Schwarz 1978)

$$\mathrm{BIC}(\mathcal{M}_i) = \dim(\boldsymbol{\theta}_i) \ln N - 2 \ln p(\mathcal{D} \mid \mathcal{M}_i(\widehat{\boldsymbol{\theta}}_i)), \tag{2.14}$$

where N is the sample size. The AIC and BIC both provide a simple means of comparing models, including models of Lagrangian motion (Sykulski et al. 2017), and in some cases will be easier to compute than the model evidence. However, their definitions are justified by asymptotic arguments valid only in the limit of infinite data. In contrast, the model evidence is valid outside of this regime and accounts properly for parameter uncertainty. Note, in particular, that, despite its name, the BIC does not take into account the prior distribution of parameters.

Yet another means of comparing models is with a likelihood ratio test (Vuong 1989, Sykulski et al. 2016). However, the interpretation of the likelihood ratio relies on its asymptotic distribution (typically a χ^2 distribution). Hence, this approach, too, applies formally only in the large data regime.

In this work we restrict our attention solely to BMC, which we favour for its wider applicability and careful accounting of parameter uncertainty.

2.4 Results

In this section we provide details on how BMC can be performed for the Brownian and Langevin model and consider data generated by the Langevin model. We derive the likelihood function for each model, discuss prior distributions for parameters, and the practicalities of inference calculations.

Before we compute the Bayes factor for the Langevin and Brownian models $\mathcal{M}_L$ and $\mathcal{M}_B$, we infer the parameters of both models using a range of datasets with varying sampling time, τ, to establish when each model is *sampling-time consistent* — we say a model is sampling-time consistent when inferred parameter values are stable over a range of τ. We emphasise that sampling-time consistency does not imply a model is good, but is certainly a desirable property when one wishes to use a model for extrapolation, e.g. for unobserved values of τ.

Justifications for the Brownian model apply formally only in the large-time limit; we are, therefore, interested in establishing a minimum timescale for the sampling-time consistency of the Brownian model, and further establishing whether the Langevin model, given that it includes time correlation, is sampling-time consistent on shorter timescales.

Note that in the large-time limit, that is, for $t \gg \gamma^{-1}$, the Langevin dynamics are asymptotically diffusive: for $\gamma \to \infty$, the Langevin equations (2.2) reduce to (Pavliotis

2014)

$$\mathrm{d}\boldsymbol{X} = \sqrt{2k}\,\mathrm{d}\boldsymbol{W}. \tag{2.15}$$

To see this, note the solution (A.4), in which both terms involving γ vanish, leaving

$$\boldsymbol{X}(t) = \boldsymbol{X}(0) + \sqrt{2k}\boldsymbol{W}(t), \tag{2.16}$$

as required. This fact is important when comparing the models, and we return to it later.

2.4.1 Likelihoods

We can derive explicit expressions for the probability of data of the form of $\Delta\mathcal{X}_\tau$ under $\mathcal{M}_B(\kappa)$ and $\mathcal{M}_L(\gamma,\, k)$ by using their transition probabilities. The position increments for $\mathcal{M}_B(\kappa)$ satisfy

$$\boldsymbol{X}(t+\tau) - \boldsymbol{X}(t) \sim \mathcal{N}\left(0,\, 2\kappa\tau\mathbb{I}\right), \tag{2.17}$$

where $\mathcal{N}(\mu,\, C)$ is the d-dimensional Gaussian distribution with mean μ and covariance matrix C, and $\mathbb{I}$ is the $d \times d$ identity matrix. Further, distinct increments are independent under $\mathcal{M}_B(\kappa)$. Therefore, the desired probability is

$$p\left(\Delta\mathcal{X}_\tau \mid \mathcal{M}_B(\kappa)\right) = \prod_{p=1}^{N_p}\prod_{n=0}^{N_\tau-1}\prod_{i=1}^{d}\rho_{\mathcal{N}}\left(\Delta X_{n,i}^{(p)};\, 0,\, 2\kappa\tau\right), \tag{2.18}$$

where i indexes spatial dimension and $\rho_{\mathcal{N}}(\boldsymbol{x}\,;\,\boldsymbol{\mu}, C)$ is the probability density at $\boldsymbol{x}$ of the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu},\, C)$.

The corresponding likelihood for the Langevin model is shown in Appendix A.1 to be

$$p(\Delta\mathcal{X}_\tau \mid \mathcal{M}_L(\gamma,k)) = \prod_{p=1}^{N_p}\prod_{i=1}^{d}\rho_{\mathcal{N}}\left((\Delta X_{0,i}^{(p)},\, \cdots\, \Delta X_{N_\tau-1,i}^{(p)})^{\mathrm{T}};\, \mathbf{0},\, S\right), \tag{2.19}$$

where S is the symmetric Toeplitz matrix with

$$S_{ij} = \begin{cases} 2k\tau(1-\varphi(\gamma\tau)) & \text{if } m = 0 \\ k\gamma\tau^2\varphi^2(\gamma\tau)e^{-(m-1)\gamma\tau} & \text{if } m > 0 \end{cases}, \tag{2.20}$$

$$\varphi(x) = \frac{1-e^{-x}}{x}, \tag{2.21}$$

and $m = |i-j|$.

2.4.2 Prior distributions

It is necessary, both for parameter and model inference, to specify a prior distribution for each of the parameters, κ, γ, and k. For a given flow we can appeal to scaling considerations to assign a prior mean to each parameter, derived from characteristic scales. Once such prior means are prescribed, the maximum entropy principle, along with positivity and independence of the parameters motivates a choice of corresponding exponential distributions as priors (Jaynes 2003, Cover & Thomas 2006). That is, for a parameter $\theta > 0$ with prior mean μ, the distribution with maximum entropy is the exponential distribution $\mathrm{Exp}(\lambda)$ with rate $\lambda = 1 / \mu$. We use this prescription for our choice of prior.

2.4.3 Inference numerics

The computations we perform for Bayesian parameter inference are: (i) an optimisation procedure to find the posterior mode, $\boldsymbol{\theta}^*$, and (ii) a single evaluation of the Hessian of the log-posterior distribution at $\boldsymbol{\theta}^*$, $-J$ in (2.11), which we can use to estimate the posterior variance by a Gaussian approximation as in (2.10). We have analytical expressions for the likelihood and prior for both models, so we can easily evaluate the negative log unnormalised posterior, $f(\boldsymbol{\theta}) = -\ln p_u(\boldsymbol{\theta} \mid \mathcal{D})$, in each case; we find $\boldsymbol{\theta}^*$ by minimising $f(\boldsymbol{\theta})$ using the `SciPy` function `optimize.minimize()`.

In the case of the Brownian model derivatives of $f(\boldsymbol{\theta})$ are easily derived analytically, so we use the `L-BFGS-B` routine which exploits gradient information and allows for the specification of lower bound constraints to enforce positivity (Zhu et al. 1997). In the case of the Langevin model calculation of derivatives of the posterior is nontrivial because the likelihood (2.19) is a complicated function. For this reason we use the gradient-free `Nelder-Mead` (Nelder & Mead 1965) routine rather than `L-BFGS-B`. We evaluate J^{-1} approximately using a fourth-order central difference approximation for the log-likelihood.

No further computations are required for BMC if the Laplace's method approximation for the evidence in (2.12) is used.

2.4.4 Test case: Langevin data

As a test case and to build intuition, we first consider trajectory data generated by the Langevin model with $d = 3$. In this case, one of the two candidate models is the true model. We generate the data by simulating the Langevin SDE (2.2) exactly, drawing initial velocities from the stationary distribution $\boldsymbol{U} \mid \mathcal{M}_L(\gamma, k) \sim \mathcal{N}(\boldsymbol{0}, \gamma k \mathbb{I})$, and using the transition probabilities (A.1); velocity data are discarded to construct the dataset of position increments $\Delta\mathcal{X}_\tau$.

We set $\gamma = k = 1$, fix $N_p = 100$ and $N_\tau = 10$, and perform Bayesian parameter inference and model comparison with a series of independently generated datasets with $\tau \in [10^{-2}, 10^2]$. We set fixed priors γ, k, $\kappa \sim \mathrm{Exp}(1)$.

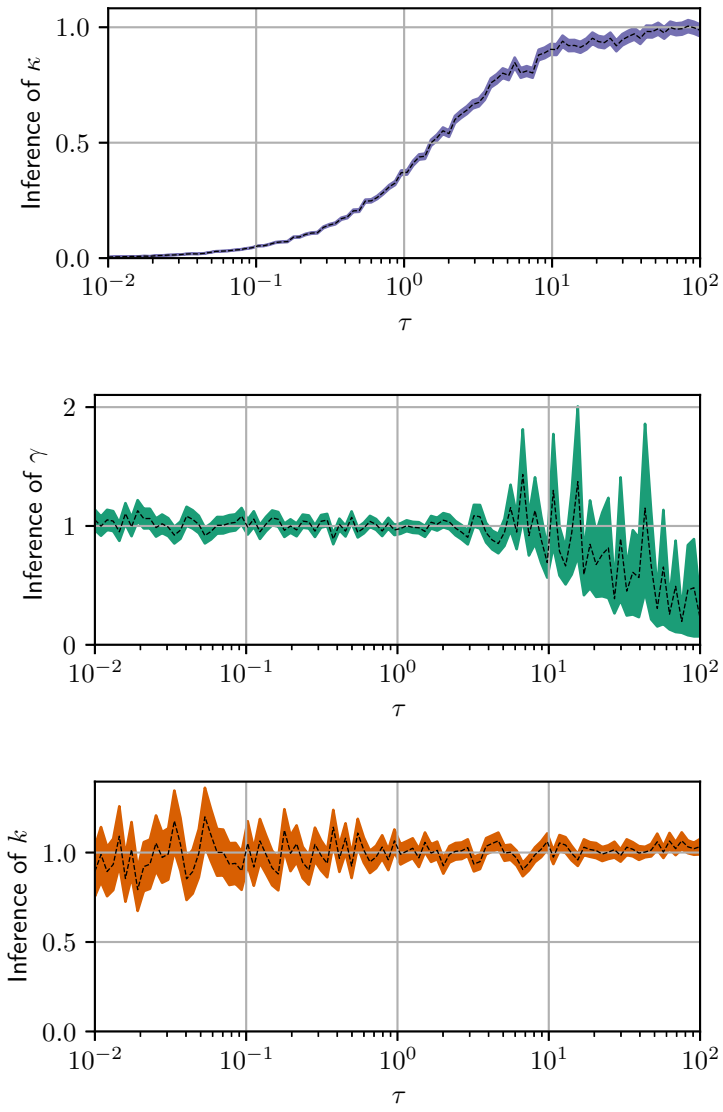


Figure 2.1: Parameter inference for the Brownian and Langevin models as a function of observation interval, τ, for data from the Langevin model in three spatial dimensions — the true value of each parameter is 1. Dashed lines indicate posterior mode estimates, $\theta^* = \kappa$ (top), γ^* (middle) and k^* (bottom); shaded areas show $\theta^* \pm \mathrm{SD}(\theta \mid \Delta X_\tau)$. Each inference is made with a fixed volume of data: $N_p = 100$ and $N_\tau = 10$.

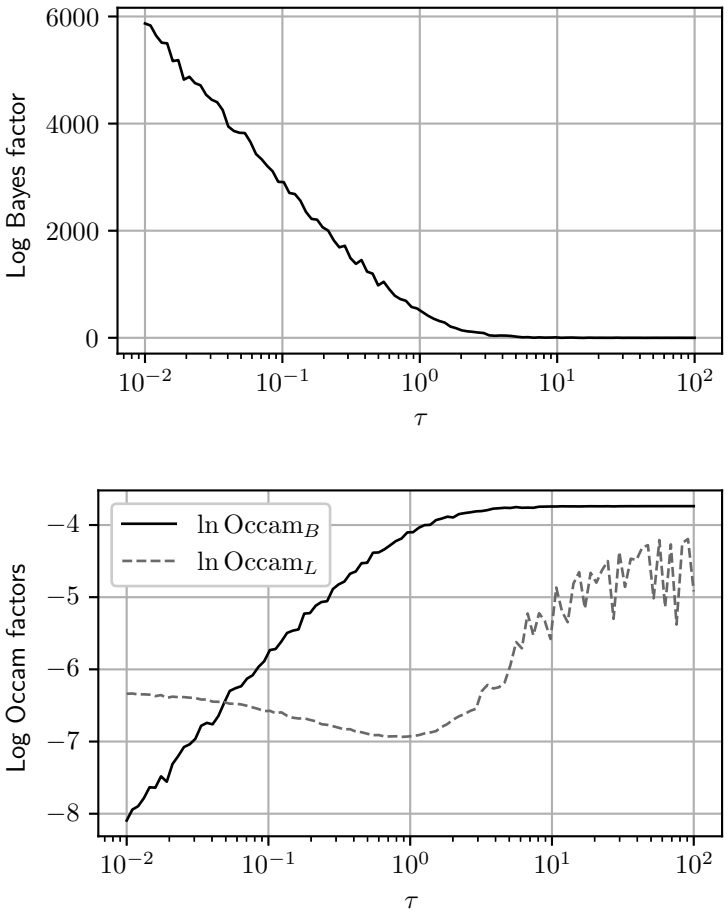


Figure 2.2: Log Bayes factors, $\ln K_{L,B}$, and corresponding log Occam factors, as a function of τ, given the same data used for Figure 2.1.

Figure 2.1 shows the results of the parameter inference. Note that both Langevin parameters are well identified until, at sufficiently large τ, the error of the posterior mode estimate of γ grows along with the posterior standard deviation of γ. This is a manifestation of the diffusive limit (2.15) of the Langevin dynamics, wherein $\boldsymbol{X}(t + \Delta t) - \boldsymbol{X}(t) \sim \mathcal{N}(\boldsymbol{0}, 2k\Delta t\mathbb{I})$ is independent of γ. Unsurprisingly, then, $\Delta\mathcal{X}_\tau$ is less informative about γ when τ is large.

The diffusivity κ of the Brownian model is sampling-time consistent only when τ is sufficiently large, i.e. in the diffusive limit of the Langevin dynamics, when $\kappa \approx k$. The inaccuracy of posterior mode estimates of κ at small τ is expected as it is known that the inference of diffusivities from discrete trajectory data is sensitive to sampling time (Cotter & Pavliotis 2009). We note that $\gamma^{-1} = 1$ is the decorrelation time for this data so that the timescales at which this limiting behaviour is observed, $\tau \gtrsim 10$, are indeed large.

Note that the posterior mode estimates of γ eventually decay to zero as τ increases; since, as observed, $\Delta\mathcal{X}_\tau$ becomes less informative about γ with increasing τ, the contribution of the prior information to the posterior becomes dominant over the contribution from the likelihood — the consequence of this is that the posterior mode tends to the prior mode, which is zero since we take $\gamma \sim \mathrm{Exp}(1)$.

Figure 2.2 shows the log Bayes factors found using Laplace's method for the evidences. We see that for a significant range of τ the Langevin model is preferred, indicated by large positive values of $\ln K_{L,B}$, but its dominance diminishes as τ increases until the diffusive limit is reached, at which point values of $|\ln K_{L,B}| < 1$ are typical, indicating insubstantial preference for either model.

Also shown in Figure 2.2 are the corresponding log Occam factors. Occam factors for the Brownian model are approximately constant once τ is sufficiently large, while the Occam factors for the Langevin model increase at large τ in line with a broadening posterior. This is indicative of decreased sensitivity to choice of parameters, specifically γ, whose value becomes less critical for explaining dynamics on large timescales.

2.5 Application to two-dimensional turbulence

In this section we report an application of BMC to particle trajectories in a model of stationary, isotropic two-dimensional (2D) turbulence.

2.5.1 Forced-dissipative model

We consider a forced–dissipative model of isotropic 2D turbulence in an incompressible fluid governed by the vorticity equation (Vallis 2017)

$$\frac{\partial\zeta}{\partial t} + (\boldsymbol{u}\cdot\nabla)\zeta = F + D, \tag{2.22}$$

$$\text{domain} = [0,\ 2\pi]^2; \quad \text{resolution} = 1024 \times 1024 \text{ grid points};$$
$$\text{timestep size} = 2.5 \times 10^{-4}; \quad k_F = 64; \quad A_F = 8.9 \times 10^8; \quad A_{\text{lsf}} = 1.$$

Table 2.1: Flow configuration parameter values for simulations of the 2D turbulence model.

where ζ is the vertical vorticity and F and D represent forcing and dissipation, respectively. The particular forcing used is an additive homogeneous and isotropic white Gaussian noise concentrated in a specified range of wavenumber centred about a forcing wavenumber, k_F. In particular, following Scott (2007), we have that, at each timestep, the Fourier transform of F satisfies

$$\text{Re}\left(\hat{F}(\boldsymbol{k})\right) \overset{d}{=} \text{Im}\left(\hat{F}(\boldsymbol{k})\right) \sim \mathcal{N}\left(0, \frac{A_F \mathcal{F}_F(|\boldsymbol{k}|)}{2\pi|\boldsymbol{k}|}\right), \tag{2.23}$$

where A_F is the forcing amplitude, and $\mathcal{F}_F(|\boldsymbol{k}|) = 1$ for $||\boldsymbol{k}| - k_F| \leq 2$ and $\mathcal{F}_F = 0$ otherwise.

Two dissipation mechanisms are included: (i) small-scale dissipation implemented with a scale-selective exponential cut-off filter (see Arbic & Flierl (2003) for details and justification), and (ii) large-scale friction (aka hypodiffusion), so that total dissipation is given by

$$D = A_{\text{lsf}}\psi + \text{ssd}, \tag{2.24}$$

where ssd denotes the small-scale dissipation.

Equation (2.22) is solved in a periodic domain, $[0,\ 2\pi]^2$, using a standard pseudospectral solver, at a resolution of 1024×1024 gridpoints, with the third-order Adams–Bashforth timestepping scheme. The complete set of flow configuration parameter values for our simulations are given in Table 2.1.

The model is initialised with a random, Gaussian field with prescribed mean energy spectrum and is run until the total energy,

$$E(t) \coloneqq \frac{1}{2}\int |\boldsymbol{u}(\boldsymbol{x}, t)|^2 \,\mathrm{d}x\,\mathrm{d}y, \tag{2.25}$$

appears to reach a statistically stationary state; this amounted to a spin-up time of approximately 6800 eddy turnover times, where the eddy turnover time is estimated by

$$\tau_\zeta = 2\pi \,/\, \sqrt{Z}, \tag{2.26}$$

and Z is the total enstrophy,

$$Z \coloneqq \frac{1}{2}\int \zeta^2 \,\mathrm{d}x\,\mathrm{d}y. \tag{2.27}$$

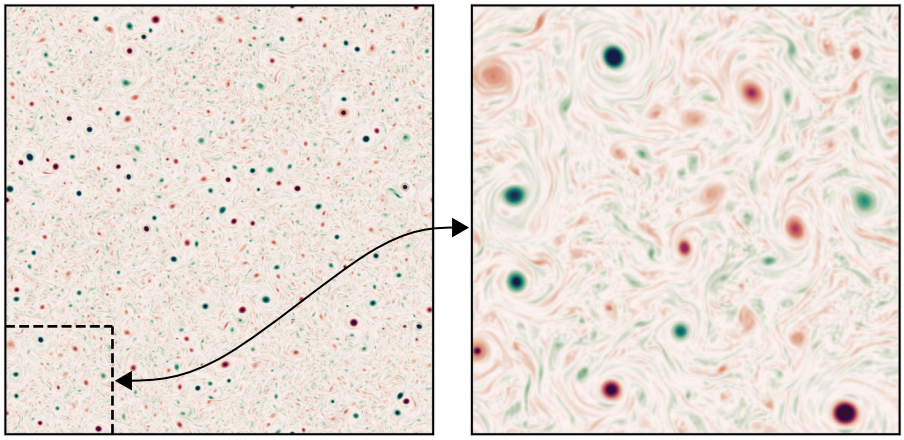


Figure 2.3: Snapshot of the vorticity field in the forced-dissipative model at stationarity showing $x, y \in [0,\ 2\pi]$ (left) and $x, y \in [0,\ \pi\ /\ 2]$ (right).

Figure 2.3 shows a snapshot of the vorticity field at the end of the spin-up process. Enstrophy is concentrated in a population of coherent vortices whose scale is set by the forcing scale, k_F^{-1}. Figure 2.4 shows the isotropic energy spectrum calculated at the same instant. A power law of approximately k^{-2} is observed at wavenumbers between the peak wavenumber at $k \approx 6$ and the forcing wavenumber at $k \approx 64$ (indicated with a vertical line). A second power law of approximately $k^{-3.5}$ is seen at wavenumbers between the forcing scale and the dissipation range. Large-scale friction prevents the indefinite accumulation of energy at the largest scales, while continued forcing prevents energy from concentrating exclusively around a peak wavenumber at late times, and, by inputting enstrophy at a moderate scale, sustains a lively population of vortices.

2.5.2 Particle numerics

After spin-up, a set of 1000 passive tracer particles are evolved in the flow of the forced-dissipative model for approximately another 6800 eddy turnover times; this is done using bilinear interpolation of the velocity field and the fourth-order Runge–Kutta time-stepping scheme. Particles are seeded at initial positions chosen uniformly at random in the domain.

Figure 2.5 shows a subset of the trajectory data generated. Some particles follow highly oscillatory paths, while others do not, depending on whether they are seeded in the interior of a coherent vortex or in the background turbulence.

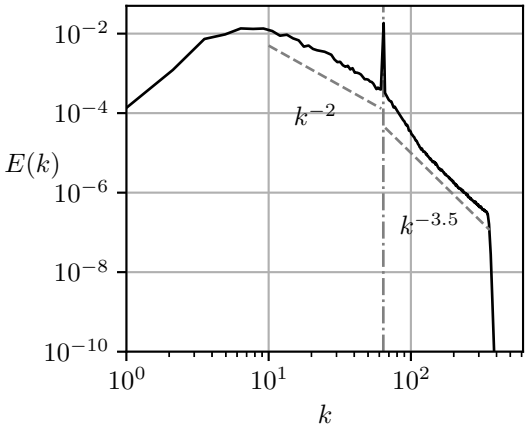


Figure 2.4: Snapshot of the isotropic energy spectrum in the forced–dissipative model at stationarity.

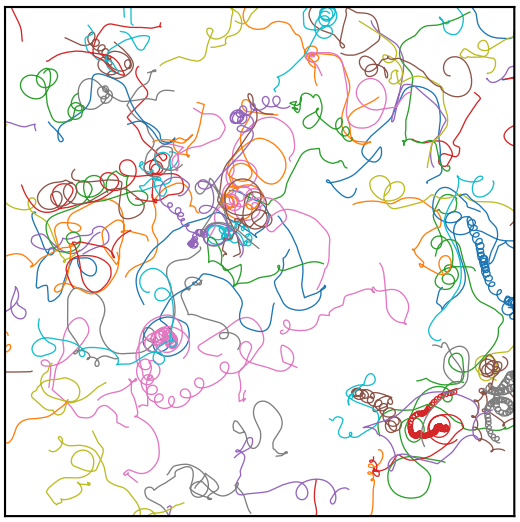


Figure 2.5: Trajectories of 100 passive particles advected in the the forced–dissipative model, shown as recorded over a period of $100\tau_\zeta$ with a different colour for each trajectory.

2.5.3 Diagnostics

To illustrate the dynamics that we parameterise with the stochastic models we show two diagnostics commonly used in Lagrangian analyses (Pasquero et al. 2001, van Sebille et al. 2018), namely, the Lagrangian velocity autocovariance function (LVAF), defined in the isotropic case as

$$r(\tau) = \langle U^{(p)}(t+\tau)U^{(p)}(t)\rangle, \tag{2.28}$$

where the angled-brackets denote the average over t and particles p, and the absolute diffusivity

$$\kappa_{\mathrm{abs}}(\tau) = \frac{\left\langle \left(\Delta X^{(p)}(\tau)\right)^2 \right\rangle}{2\tau}. \tag{2.29}$$

Figure 2.6 shows the LVAF as estimated from the simulated particle trajectory data. The corresponding LVAF of the Brownian model is a delta function at zero, since velocity is implicitly represented as a white-noise process, while the LVAF of the Langevin model, which represents particle velocity as an Ornstein–Uhlenbeck process, is

$$r_{\mathrm{OU}}(\tau\,;\,\gamma,\,k) = k\gamma\exp(-\gamma|\tau|). \tag{2.30}$$

In contrast, the estimated LVAF of the forced–dissipative model not only shows finite decorrelation time but is noticeably sub-exponential.

Figure 2.7 shows the estimated absolute diffusivity. In line with the asymptotic laws described in Taylor (1922) the absolute diffusivity is linear at small τ and constant at large τ, corresponding to the ballistic and diffusive regimes, respectively. The absolute diffusivity of the Brownian model is constant, while that of the Langevin model is

$$\kappa_{\mathrm{OU}}(\tau) = k\left(1-\varphi(\gamma\tau)\right), \tag{2.31}$$

which is linear at small τ and constant at large τ.

Qualitatively, from comparing these diagnostics with those of the stochastic models it is clear that on sufficiently large times (in the diffusive regime) the Brownian model is valid; in particular, the LVAF is well-approximated by a delta function at large-times, and correspondingly the absolute diffusivity is constant. On timescales shorter than the diffusive regime the LVAF of the observed trajectories may be better approximated by that of the Langevin model; however, the quality of this approximation is in general unclear a priori. It could be tempting to estimate γ by fitting the LVAF, using e.g. a least-squares method, but this approach would not correctly deal with uncertainty in parameters.

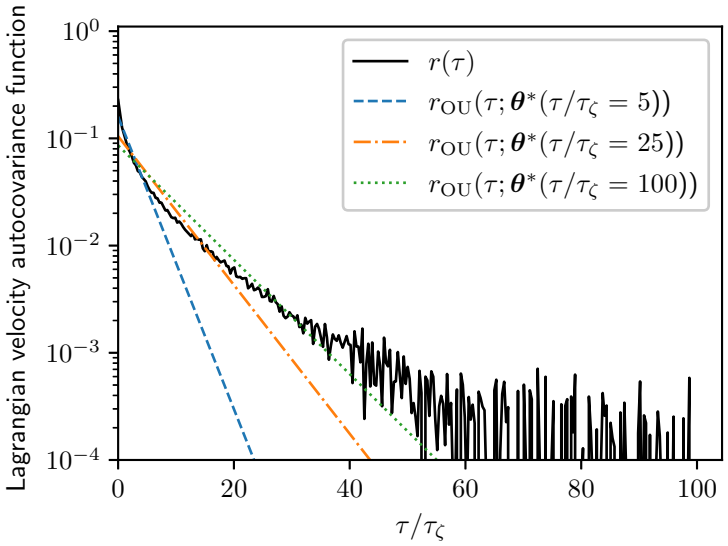


Figure 2.6: LVAF $r(\tau)$ for the forced–dissipative model, as estimated from the full set of 1000 simulated particle trajectories. The LVAF of the Langevin model $r_{\mathrm{OU}}(\tau)$ is also shown using MAP estimates (discussed below) $\boldsymbol{\theta}^* = (\gamma^*, k^*)$ derived from datasets with $\tau = (5,\ 25,\ 100)\tau_\zeta$, respectively (see Figure 2.8).

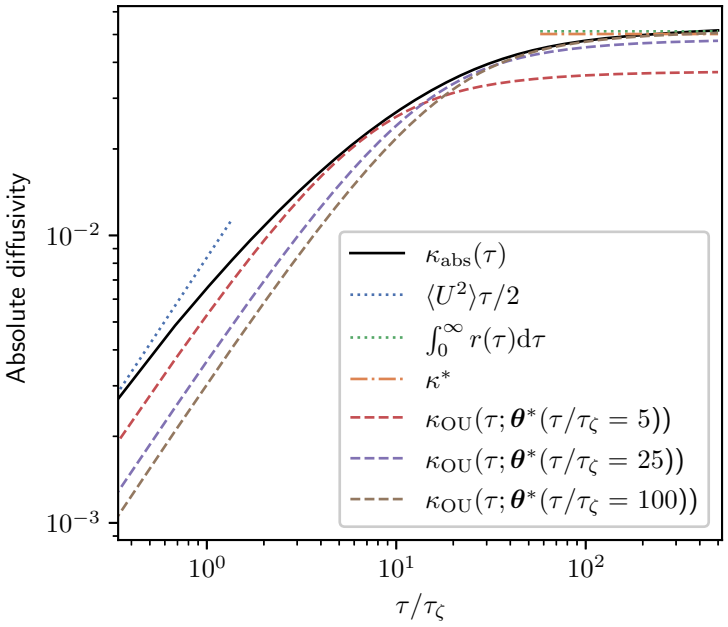


Figure 2.7: Absolute diffusivity, $\kappa_{\mathrm{abs}}(\tau)$, for the forced–dissipative model, as estimated from the full set of 1000 simulated particle trajectories. A MAP estimate κ^* is shown, along with two asymptotic laws: $\kappa_{\mathrm{abs}}(\tau)$ = linear (ballistic regime), and $\kappa_{\mathrm{abs}}(\tau)$ = const. (diffusive regime).

2.5.4 Parameter inference and BMC

We now apply the parameter inference and BMC procedures demonstrated in the test case of §2.4.4. By subsampling the results of our particle simulations we generate datasets with $N_p = 1000$, $N_\tau = 25$, and a set of sampling times τ in the range $[\tau_\zeta, 250\tau_\zeta]$.

Prior means for the parameters are derived from τ_ζ and the root-mean-square velocity $u_{\mathrm{RMS}} = \sqrt{2E}$, where E is the mean energy: as discussed in §2.4.2 we take

$$\mathbb{E}[\kappa] = \mathbb{E}[k] = u_{\mathrm{RMS}}^2 \tau_\zeta, \tag{2.32a}$$

$$\mathbb{E}[\gamma] = \tau_\zeta^{-1}, \tag{2.32b}$$

and use the corresponding exponential distributions as priors.

The results of the parameter inference are shown in Figure 2.8. The Brownian model is sampling-time consistent for $\tau \gtrsim 150\tau_\zeta$, with a posterior mode that differs by 40% from the scaling estimate used as prior mean. The long time required for sampling-time consistency is in line with the expected validity of the Brownian model in the long-time limit. In this limit κ^* agrees very well with Taylor's 1922 theoretical prediction, $\kappa = \int_0^\infty r(\tau)\,\mathrm{d}\tau$ (see Figure 2.7).

The Langevin model is roughly sampling-time consistent from much smaller values of τ, say $\tau \gtrsim 50\tau_\zeta$. This suggests that there is a range of sampling times, roughly $50\tau_\zeta \lesssim \tau \lesssim 150\tau_\zeta$, where the Langevin model is potentially useful but the Brownian model is not. The BMC analysis below sheds further light on this. However, there is noticeable decay in the MAP estimates of γ with increasing τ — this is likely a reflection of the sub-exponential nature of the true LVAF. In Figure 2.6 we plot the Langevin LVAF given parameters inferred with data of various τ, where the decay in estimates of γ corresponds to a shallowing of the Langevin LVAF. In Figure 2.7 we plot the absolute diffusivity of the Langiven model κ_{OU} using the same parameter estimates as in Figure 2.6. The absolute diffusivity is best approximated at a timescale matching the sampling time of the data. The posterior mode of γ, when roughly stable, is almost an order of magnitude smaller than the scaling estimate in (2.32), indicating that particle dynamics decorrelate slower than might be predicted by a naive dimensional analysis based on the enstrophy alone. In particular, the inferred value of γ corresponds to a decorrelation time of about 8 eddy turnover times. As in the test case of §2.4.4 the posterior standard deviation of γ grows with τ as the diffusive limit is reached and the particle dynamics become insensitive to γ. It is interesting to note that the Langevin diffusivity k is estimated consistently for sampling times much shorter than those required to estimate the Brownian diffusivity κ even though their values are identical when the Brownian model represents the dispersion well. This suggests that carrying out Bayesian inference of the Langevin model might provide a means to estimate the Brownian diffusivity when data is not available over the long, diffusive time scales that are required a priori. This may generalise to other flows only when the Langevin model is a reasonable approximation — inference of k is unlikely to be sampling time

consistent if inference of γ is not, for example, due to the LVAF of the flow of interest being very far from exponential. We emphasise that the inference results just described are largely insensitive to specification of the prior.

The results of the BMC for the turbulence model data are shown in Figure 2.9. The picture is similar to that in the test case of §2.4.4, in that the Bayes factor favours the Langevin model for shorter timescales, but with diminishing strength as τ is increased, until, at timescales corresponding to convergence of the Brownian diffusivity, the value of the log Bayes factor becomes small enough that the two models cannot be meaningfully discriminated.

Also shown in Figure 2.9 are the corresponding log Occam factors. For τ large enough that the Brownian model is sampling-time consistent, its Occam factor is approximately constant and larger than that of the Langevin model. As in the test case in §2.4.4, the Occam factor for the Langevin model increases towards that of the Brownian model at large τ when the particle dynamics are sufficiently decorrelated that the likelihood is less sensitive to the value of γ, albeit more slowly, owing to the more slowly decaying LVAF of the turbulent dynamics. The difference in log Occam factors is much smaller than the difference in the corresponding maximum log-likelihoods for all but the largest values of τ, which explains why the Bayes factor mainly favours the Langevin model.

In summary, these results indicate that while the Brownian model is adequate on sufficiently large timescales($\tau \gtrsim 150\tau_\zeta$), the Langevin model can explain better the dynamics of tracer particles in the turbulence model of §2.5.1 on shorter timescales ($50\tau_\zeta \lesssim \tau \lesssim 150\tau_\zeta$). On time scales $\tau \gtrsim 150\tau_\zeta$ the two models are indistinguishable in their performance, so that in this regime the Brownian model should be favoured in practice as a more parsimonious description.

2.6 Conclusions

We have demonstrated the application of BMC to a problem of interest in fluid dynamics, and shown that we can compare the performance of competing stochastic models of particle dynamics given discrete trajectory data alone while accounting for parameter uncertainty. In particular, we found that the Langevin model is preferred over the Brownian model for describing particle dynamics in a model of two-dimensional turbulence on a range of timescales, but that on sufficiently large timescales the two models perform equally well.

The broad conclusion of the BMC, then, is that the additional complexity of the Langevin model, associated with the presence of an additional parameter, is justified: its better capability to explain the data, as quantified by the maximum likelihood, overwhelms the penalty for complexity quantified by the Occam factor. We stress, however, that this conclusion does not take into account the computational cost involved if the models are used for predictions.

The application of the BMC method to other problems is limited by the feasibility of the calculation of the model evidence. Specifically, BMC inherits the usual chal-

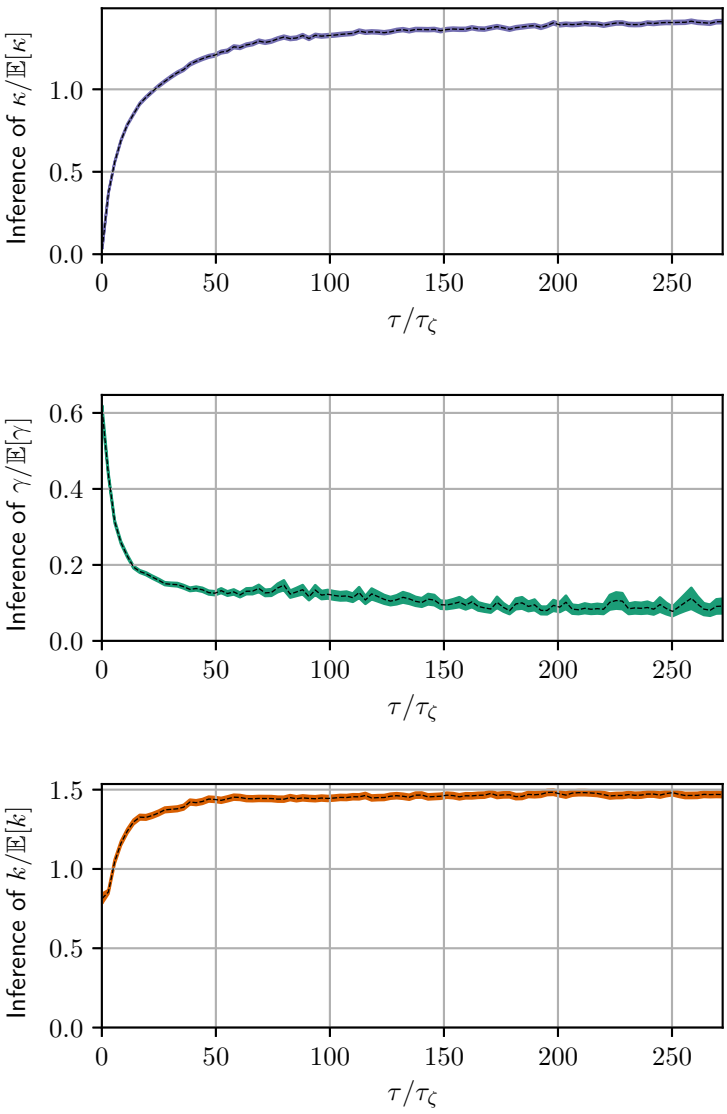


Figure 2.8: Parameter inference for the Brownian and Langevin models as a function of observation interval, τ, for data from the two-dimensional turbulence model. Dashed lines indicate posterior mode estimates, θ^*, normalised with respect to prior means, and shaded areas are $\theta^* \pm \mathrm{SD}(\theta \mid \Delta\mathcal{X}_\tau)$. Each inference is made with a fixed volume of data: $N_p = 1000$ and $N_\tau = 25$.

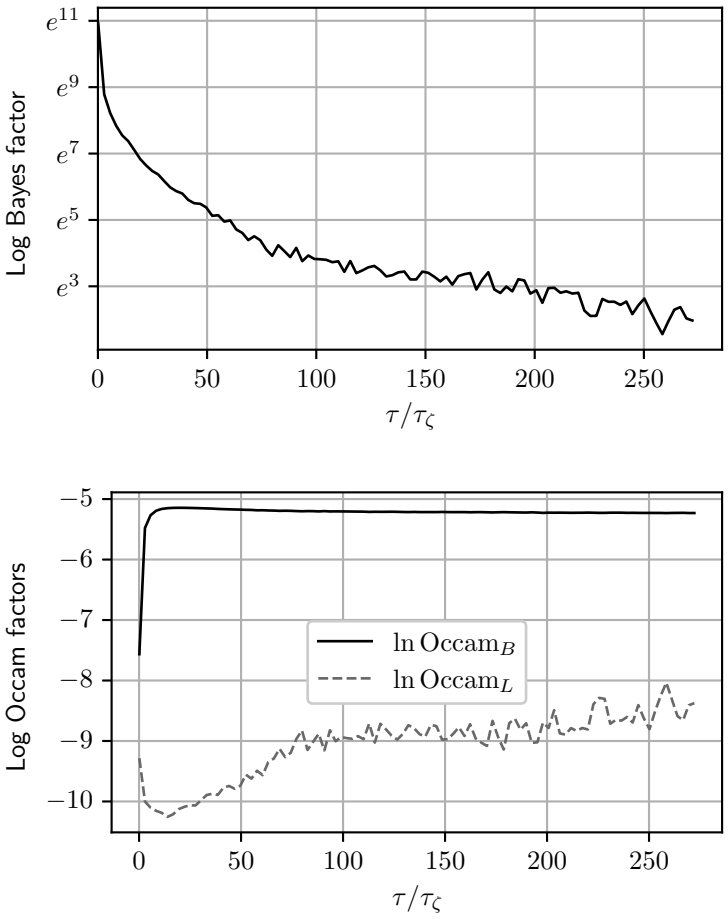


Figure 2.9: Log Bayes factors, $\ln K_{L,B}$, and corresponding log Occam factors, as a function of τ, given the same data used for Figure 2.8.

lenges of Bayesian and likelihood-based inference procedures, namely that the likelihood can be intractable or expensive to compute for complex models — the Brownian and Langevin models considered here, as linear stochastic differential equations, are very simple examples whose likelihoods could be computed analytically — alternative models which are nonlinear, have higher dimension, or have more complicated correlation structure will likely have intractable likelihoods. For example, for spatially inhomogeneous flows, such as in the atmosphere or oceans, nonlinear models arise with spatially-varying (and hence high-dimensional) parameters. Fortunately, the collection of methods referred to as approximate Bayesian computation have been developed to deal with this problem. For example, Carson et al. (2018) used the SMC^2 ('sequential Monte Carlo squared') algorithm to compare SDE models of glacial–interglacial cycles with intractable likelihoods. Calculation of the likelihood can also be more difficult when observations are noisy. For the Brownian and Langevin models it is straightforward to account for additive Gaussian noise. However, in other cases, the likelihood of noisy observations is often intractable and approximate methods are again required (see, e.g., Reich & Cotter 2015).

There is the further issue of performing the integration required to obtain the evidence as in (2.7). When the posterior is sufficiently Gaussian-like, i.e. peaked around a single mode, Laplace's method can be very accurate (Kass & Raftery 1995) as well as cheap, however, this requires (at least an approximation to) the Hessian of the log-posterior at its mode. Aside from Laplace's method, Krog & Lomholt (2017) and Thapa et al. (2018) have implemented the nested sampling algorithm of Skilling (2004) to calculate the evidence in similar analyses, while the line of work by Hannart et al. (2016), Carrassi et al. (2017) and Metref et al. (2019) has sought to perform model evidence estimation using ensemble-based data assimilation methods originally designed for state estimation in the context of incomplete, noisy observations of high-dimensional dynamical systems.

While BMC inevitably comes with computational challenges in complex problems, there are many cases where it can feasibly be applied, and, where it cannot, it should serve as a useful theoretical starting point, with alternative methods measured by how well their conclusions agree with those of BMC.

Data availability statement. The code required to reproduce the results in this chapter is available at doi.org/10.5281/zenodo.5820320 and the trajectory data generated with the 2D turbulence model is available at doi.org/10.7488/ds/3267.

Chapter 3

Inferring ocean transport statistics with probabilistic neural networks

3.1 Introduction

The motion of turbulent fluids can be characterised usefully by dynamical statistics such as dispersion, energy spectra and velocity structure functions (e.g., Batchelor 1953, Monin & Yaglom 1971). In oceanography much effort has been directed towards inferring such statistics from observations (e.g., LaCasce 2008, van Sebille et al. 2018). In many cases, these inference tasks can be related to problems in conditional probability density estimation. For example, estimating single-particle dispersion is related to estimating the conditional density

$$p\left(\boldsymbol{X}(t+\tau) - \boldsymbol{X}(t) \mid \boldsymbol{X}(t),\, t,\, \tau\right), \tag{3.1}$$

where $\boldsymbol{X}(t)$ is the position of a particle at time t, in that the dispersion is the variance of this distribution. Similarly, the velocity structure functions are moments of the conditional density

$$p\left(\boldsymbol{u}(\boldsymbol{x}_1,\, t) - \boldsymbol{u}(\boldsymbol{x}_2,\, t) \mid \boldsymbol{x}_1,\, \boldsymbol{x}_2,\, t\right), \tag{3.2}$$

where $\boldsymbol{u}(\boldsymbol{x},\, t)$ is the fluid velocity at position $\boldsymbol{x}$. By estimating full conditional densities like (3.1) and (3.2), it is possible to estimate simultaneously a number of related statistics. For instance, (3.1) describes entirely the single-particle displacement statistics, while (3.2) encodes velocity structure functions of all orders, providing two-point Eulerian velocity statistics. It is no surprise, then, that estimating these conditional densities accurately is a nontrivial task.

In this work we consider a particular tool for conditional density estimation, the mixture density network (MDN) (Bishop 1994), and test its performance in learning fluid statistics from observations. MDNs are machine learning models, which combine artificial neural networks with probabilistic mixture models to represent conditional densities (Bishop 2006). Their use has increased rapidly in recent years with

applications in a variety of fields for a range of reduced order modelling and emulation tasks, including surrogate modelling of fluid flow (Maulik et al. 2020), parameterisation of subgrid momentum forcing in ocean models (Guillaumin & Zanna 2021), emulation of complex stochastic models in epidemiology (Davis et al. 2020) and multiscale models of chemical reaction networks (Bortolussi & Palmieri 2018), and subgrid scale closures in large eddy simulations of turbulent combustion (Shin et al. 2021).

We focus on learning the single-particle transition density (3.1) in the ocean near-surface using Lagrangian trajectory data collected as part of the Global Drifter Program (Lumpkin & Centurioni 2019). A model of the transition density provides, at every point in the ocean, a probabilistic forecast for drifter displacements from that location. We show that the MDN model outperforms existing stochastic models of drifter dynamics based on Ulam's method (Ulam 1960, Froyland 2001), as well as another simple benchmark model, and eliminates the difficulty of designing appropriate discretisations of space needed for such models.

From the transition density it is possible to derive estimates of a range of single-particle statistics. As examples, we provide maps of the mean displacement over four days as a function of initial position $\boldsymbol{X}_0$, as well as the lateral diffusivity. The transition density produces highly non-Gaussian statistics in some regions. By calculating the Kullback–Leibler divergence between our full model and a simplified Gaussian model, we quantify and map non-Gaussianity in drifter displacements.

The MDN model also provides the basis for a discrete-time Markov process model of drifter dynamics, offering a continuous space alternative to Markov chain models which have been used in numerous studies (Maximenko et al. 2012, van Sebille et al. 2012, Miron et al. 2017, 2021). We perform a global simulation of drifters for a period of ten years with initial positions given on a uniform grid, and reproduce the 'garbage patches' in subtropical gyres seen in previous studies.

The article is structured as follows. In §3.2 we discuss conditional density estimation and the estimation of conditional statistics. In §3.3 we introduce MDNs. In §3.4 we describe the MDN model of the single-particle transition density from drifter observations. We compare its performance with alternative models, present derived single-particle statistics and simulate the clustering of drifters in subtropical gyres. In §3.5 we conclude and suggest further problems where MDNs may be a useful tool.

3.2 Conditional modelling

While the aim of regression is to model $\mathbb{E}[\boldsymbol{Y} \mid \boldsymbol{X}]$, where $\boldsymbol{X}$ and $\boldsymbol{Y}$ are random variables, conditional modelling (or conditional density estimation, CDE) is the task of inferring the full conditional probability density $p(\boldsymbol{Y} \mid \boldsymbol{X})$ [1]. By modelling conditional densities, rather than just conditional means, we incorporate information about the variability of $\boldsymbol{Y} \mid \boldsymbol{X}$; more than this, conditional models can capture skewness, excess kurtosis and multimodality. This comprehensive description of conditional

[1] We restrict attention to the case of continuous random variables.

statistics is valuable in applications where single point-estimates are insufficient due to inherent variability, and where there is interest in non-Gaussian statistics, including those associated with rare events. Conditional models can be used in two ways: (i) as stochastic surrogate models (or emulators), and (ii) as a tool for estimating conditional statistics.

Parametric conditional models (such as MDNs) assume that, for each possible value of $\boldsymbol{X}$, the distribution of $\boldsymbol{Y} \mid \boldsymbol{X}$ belongs to a certain family of parametric distributions, i.e.

$$p(\boldsymbol{Y} \mid \boldsymbol{X}) = \rho(\boldsymbol{Y}\,;\,\boldsymbol{\theta}(\boldsymbol{X})), \tag{3.3}$$

where $\rho(\,\cdot\,;\,\boldsymbol{\theta})$ is the probability density corresponding to a family of distributions parameterised by $\boldsymbol{\theta}$. In this case, not only must the form of ρ be chosen, but the dependence on the conditioned variable must also be modelled by some representation of $\boldsymbol{\theta}(\boldsymbol{X})$.

3.2.1 Estimating conditional statistics

Given data $\{\boldsymbol{X}_i, \boldsymbol{Y}_i\}$, a standard approach to estimating conditional statistics $\mathbb{E}[\boldsymbol{f}(\boldsymbol{Y}) \mid \boldsymbol{X}]$ is to first discretise (or 'bin') in $\boldsymbol{X}$ and produce local estimates $\widehat{\mathbb{E}[\boldsymbol{f}(\boldsymbol{Y})]}(\tilde{\boldsymbol{X}})$ for each value of the discretised variable $\tilde{\boldsymbol{X}}$, typically by Monte Carlo estimation, such that

$$\widehat{\mathbb{E}[\boldsymbol{f}(\boldsymbol{Y})]}(\tilde{\boldsymbol{X}}) \coloneqq \frac{\sum_i \boldsymbol{f}(\boldsymbol{Y}_i)\,\mathbb{1}_B(\boldsymbol{X}_i)}{\sum_i \mathbb{1}_B(\boldsymbol{X}_i)}, \tag{3.4}$$

where $\mathbb{1}_B$ is the indicator function of B, the set of values of $\boldsymbol{X}$ whose discretised value is $\tilde{\boldsymbol{X}}$. For estimates (3.4) to be useful, one must design a suitable discretisation of the domain of $\boldsymbol{X}$, which balances the need to choose a fine enough discretisation to resolve details in $\boldsymbol{X}$ with the need to take sufficiently large bins to have enough data for these estimates to have reasonably small variance. This can be especially challenging when data is sparse, or when the density of data is highly inhomogeneous.

Conditional modelling offers an alternative approach wherein one first constructs a model of the conditional density, as in (3.3), that is continuous in both $\boldsymbol{X}$ and $\boldsymbol{Y}$, then computes estimates

$$\mathbb{E}_{\mathcal{M}}[\boldsymbol{f}(\boldsymbol{Y}) \mid \boldsymbol{X}] \coloneqq \int \boldsymbol{f}(\boldsymbol{Y})\,\rho(\boldsymbol{Y}\,;\,\boldsymbol{\theta}(\boldsymbol{X}))\,\mathrm{d}\boldsymbol{Y} \tag{3.5}$$

for as many statistics as desired at any value of $\boldsymbol{X}$ in the domain, without the need to revisit the raw data. In some cases the expectations $\mathbb{E}_{\mathcal{M}}$ can be calculated using a closed-form expression. Where no such expression is known, the expectation can be computed by numerical integration or a Monte Carlo method. Since these calculations rely only on evaluating the modelled conditional density, or sampling from it, they are not limited by sparsity of data. Also, for a given $\boldsymbol{X}^*$, estimates of the form (3.4)

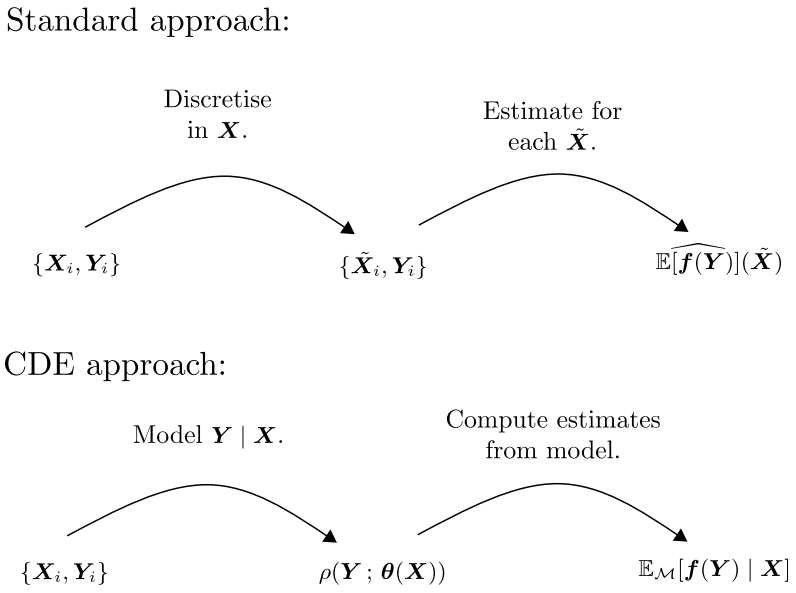


Figure 3.1: Estimating conditional statistics by a standard approach versus by first constructing a model for the conditional density.

are informed only by observations in the same bin as $\boldsymbol{X}^*$, whereas in a conditional model, all observations are used to fit $\rho(\boldsymbol{Y}\,;\,\boldsymbol{\theta}(\boldsymbol{X}^*))$. The schematic in Figure 3.1 contrasts the standard approach and the conditional modelling approaches.

3.3 Mixture density networks

A mixture density network (Bishop 1994, 2006) is a conditional model where an artificial neural network is employed to represent the function $\boldsymbol{\theta}(\boldsymbol{X})$ in (3.3) and the parametric form $\rho(\,\cdot\,;\,\boldsymbol{\theta})$ corresponds to a mixture distribution. The density of a general mixture distribution is

$$\rho(\,\cdot\,;\,\boldsymbol{\theta}) = \sum_{i=1}^{N_c} \alpha_i\, \rho_i(\,\cdot\,;\,\boldsymbol{\theta}_i), \tag{3.6}$$

where N_c is the number of components in the mixture, the i^{th} component has density $\rho_i(\,\cdot\,;\,\boldsymbol{\theta}_i)$ with parameters $\boldsymbol{\theta}_i$, $\boldsymbol{\theta} = [(\alpha_1,\ \boldsymbol{\theta}_1),\ \cdots,\ (\alpha_{N_c},\ \boldsymbol{\theta}_{N_c})]$ and the α_i are

component weights subject to the constraint

$$\sum_{i=1}^{N_c} \alpha_i = 1. \tag{3.7}$$

Commonly, the component densities ρ_i are chosen from the same family and, in particular, Gaussian, but components can be chosen differently. In the Gaussian case, the $\boldsymbol{\theta}_i$ are conditional means and covariances.

The neural network representation of $\boldsymbol{\theta}(\,\cdot\,)$ is itself parametric with parameters $\boldsymbol{w}$; hence, MDNs model $p(\boldsymbol{Y} \mid \boldsymbol{X})$ with $\rho(\boldsymbol{Y}\,;\,\boldsymbol{\theta}(\boldsymbol{X}\,;\,\boldsymbol{w}))$. The network can have any architecture, but that of a multilayer perceptron (Rumelhart et al. 1985) (also known as a fully connected multilayer feedforward neural network) with nonlinear activation functions is common — in this case $\boldsymbol{w}$ consists of the weights and biases.

A natural loss function for conditional models, which quantifies how well they fit data, is the negative (conditional) log likelihood of observations $\mathcal{D} = \{\boldsymbol{X}_i, \boldsymbol{Y}_i\}$ under the model. In MDNs this is

$$\mathcal{L}(\boldsymbol{w}\,;\,\mathcal{D}) = \sum_i -\log \rho\left(\boldsymbol{Y}_i\,;\,\boldsymbol{\theta}(\boldsymbol{X}_i\,;\,\boldsymbol{w})\right). \tag{3.8}$$

Training an MDN then amounts to finding optimal values for the neural network's parameters

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}}\ \mathcal{L}(\boldsymbol{w}\,;\,\mathcal{D}). \tag{3.9}$$

Minimising the negative log likelihood is equivalent to maximising the log likelihood of training data, also referred to as the log score in probabilistic forecasting (Bernardo 1979, Gneiting & Raftery 2007, Bröcker & Smith 2007). Maximum likelihood estimation in this context differs from the more familiar setting of fitting an unconditional model for $p(\boldsymbol{Y})$ given observed data $\{\boldsymbol{Y}_i\}$ — here, there is generically only one observed value of $\boldsymbol{Y} \mid \boldsymbol{X}$ corresponding to each observed value of $\boldsymbol{X}$, and for most values of $\boldsymbol{X}$ there are no observations at all. It is clear, then, that, for each value of $\boldsymbol{X}$, we are certainly not in the large-data regime that would allow one to invoke asymptotic properties of maximum likelihood estimates. The quality of parametric conditional models (3.3) depends critically on how well $\boldsymbol{\theta}(\boldsymbol{X}\,;\,\boldsymbol{w}^*)$ represents how the distribution of $\boldsymbol{Y} \mid \boldsymbol{X}$ varies with $\boldsymbol{X}$. In particular, since MDNs employ a neural network to model $\boldsymbol{\theta}(\boldsymbol{X})$, and neural networks are highly flexible models, it is common for MDNs to exhibit poor generalisation unless regularisation techniques are used. In the following section we employ a widely used regularisation technique known as early stopping (see e.g. Prechelt (2012)), wherein a small proportion of training data (referred to as the test set) are not used to inform steps in the optimisation scheme, but are instead used to track the evolution of an estimate of the model's generalisation error (the value of the loss function evaluated on data outside the training set). The guiding heuristic is that it is typical for the generalisation error of neural networks to

reach a minimum as training progresses before increasing due to overfitting — early stopping is a strategy where one terminates model training when the generalisation error is believed to have reached this minimum. Details of our implementation are given in the following section.

3.4 Application to single-particle statistics of the ocean near-surface

In this section we present an MDN model of the single-particle transition density (3.1) of ocean surface drifting buoys (drifters). The model's parameters are inferred from trajectory data collected as part of the Global Drifter Program (Lumpkin & Centurioni 2019).

3.4.1 Data

We use the Global Drifter Program quality-controlled 6-hour interpolated dataset, which includes positions (latitude and longitude) and sea-surface temperatures. Drifter velocity estimates are also provided, though these are obtained by simple finite-differencing of position measurements. Position measurements are obtained from satellite fixes which are nonuniform in time and subject to error. The raw measurements are treated according to the procedure of Hansen & Poulain (1996), which involves the removal of suspected spurious values and interpolation to regular 6-hour intervals. The interpolation method, which is a form of kriging (Hansen & Herman 1989), assumes contamination by an uncorrelated zero-mean noise and makes assumptions about the structure functions of the discretised position process. We leave as a caveat to our results that this preprocessing of the data could be questioned and proceed taking the interpolated data as our ground-truth. Only position observations are used in our modelling. Figure 3.2 shows how many observed displacements are recorded per squared kilometre in each 1° latitude $\times$1° longitude square. These data were recorded between 1989 and 2021 and include a total of 23893 drifter trajectories. We split the data in two parts, by selecting approximately half (11946) of the drifter trajectories at random to use for creating the model and set the remaining data aside for validation. The overall dataset contains over 18 million observations of 6-hour displacements.

In section 3.4.3 we perform a model comparison. Skill scores are computed for the full training and validation datasets with global coverage, as well as for three restricted regions, labelled A, B, and C, shown in Figure 3.3, having extents 20–50° W, 30–50° N; 145–175° E, 20–40° N; and 110–130° W, 10° S–10° N.

3.4.2 Model

The transition density is not modelled in the most general form. Instead, we (i) consider, at first, a fixed value of the time-lag τ, so that the transition density may be

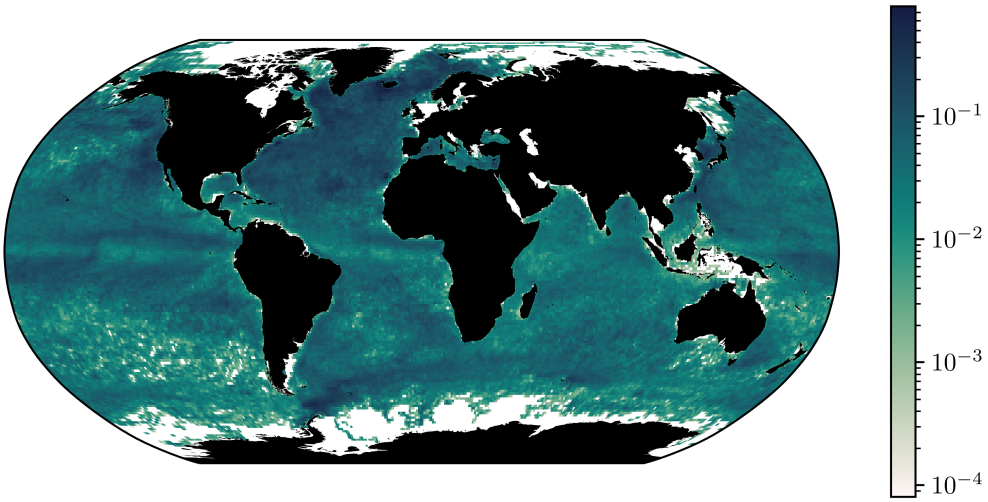


Figure 3.2: Count of drifter observations per squared kilometre.

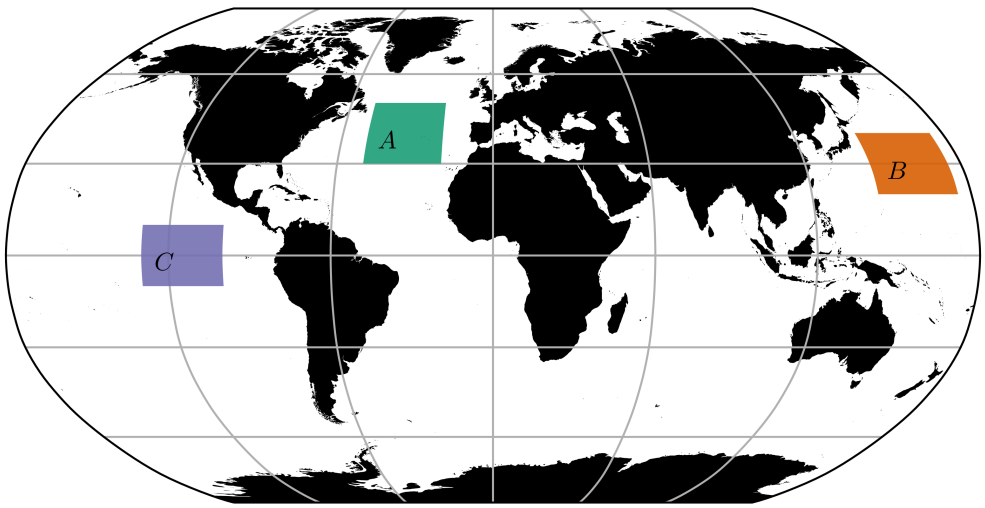


Figure 3.3: Regions considered for model comparison in section 3.4.3.

written

$$p(\boldsymbol{X}_{n+1} \mid \boldsymbol{X}_n), \tag{3.10}$$

where $\boldsymbol{X}_n = \boldsymbol{X}(t_0 + n\tau)$, and (ii) assume the process $\boldsymbol{X}(t)$ is time-homogeneous, such that (3.1) is independent of the initial time t, (3.10) is independent of n and

$$p(\Delta\boldsymbol{X} \mid \boldsymbol{X}_0) \tag{3.11}$$

represents the same information as (3.10), where $\Delta\boldsymbol{X}$ is the displacement of a drifter from its position at the previous timestep, denoted $\boldsymbol{X}_0$. By assuming time-homogeneity we neglect the effects of seasonality and low-frequency variability in ocean dynamics. If, additionally, an assumption of Markovianity is made, then (3.10) is enough to construct a discrete-time Markov process model $(\boldsymbol{X}_n)$ for drifter position (Pavliotis 2014). For a Markov assumption to be accurate, the discretisation timescale τ must be chosen appropriately. We choose a timescale of 4 days on the basis that the Lagrangian velocity decorrelation time (or integral timescale) at the surface was previously estimated from drifters to be approximately 2-3 days in all four ocean basins (Rupolo 2007).

We model (3.11) using an MDN — see the schematic in Figure 3.4. The model takes as input $\boldsymbol{X}_0$, given in longitude–latitude coordinates, and its output is a Gaussian mixture distribution with $N_c = 32$ mixture components modelling $\Delta\boldsymbol{X} \mid \boldsymbol{X}_0$, also in degrees of longitude and latitude from $\boldsymbol{X}_0$. The neural network part of the model thus encodes

$$\boldsymbol{\theta}(\cdot) = \left\{\alpha_i(\cdot),\ \boldsymbol{\mu}_i(\cdot),\ \mathbf{C}_i(\cdot)\right\}_{i=1}^{N_c} \tag{3.12}$$

such that

$$p(\Delta\boldsymbol{X} \mid \boldsymbol{X}_0) = \sum_{i=1}^{N_c} \alpha_i(\boldsymbol{X}_0) \det\left(2\pi\mathbf{C}_i\left(\boldsymbol{X}_0\right)\right)^{-\frac{1}{2}} \times \exp\left[-\frac{1}{2}\left(\Delta\boldsymbol{X} - \boldsymbol{\mu}_i\left(\boldsymbol{X}_0\right)\right)^{\mathrm{T}} \mathbf{C}_i^{-1}\left(\boldsymbol{X}_0\right)\ \left(\Delta\boldsymbol{X} - \boldsymbol{\mu}_i\left(\boldsymbol{X}_0\right)\right)\right], \tag{3.13}$$

where $\boldsymbol{\mu}_i$ and $\mathbf{C}_i$ are the mean vector and covariance matrix of mixture component i. The number of mixture components is a hyperparameter which could be optimised. We chose $N_c = 32$ on the basis that 32 component mixtures were found to be sufficiently expressive in trial experiments with MDNs.

The architecture chosen for the neural network is the standard multilayer perceptron, with six hidden (i.e. interior) layers. The first four hidden layers have 256 neurons and the remaining two have 512. The activation function $\tanh x$ is applied to each of the hidden layers. Thus, the activity of hidden layer i, is

$$\boldsymbol{h}_i = \tanh\left(W_i\boldsymbol{h}_{i-1} + \boldsymbol{b}_i\right) \tag{3.14}$$

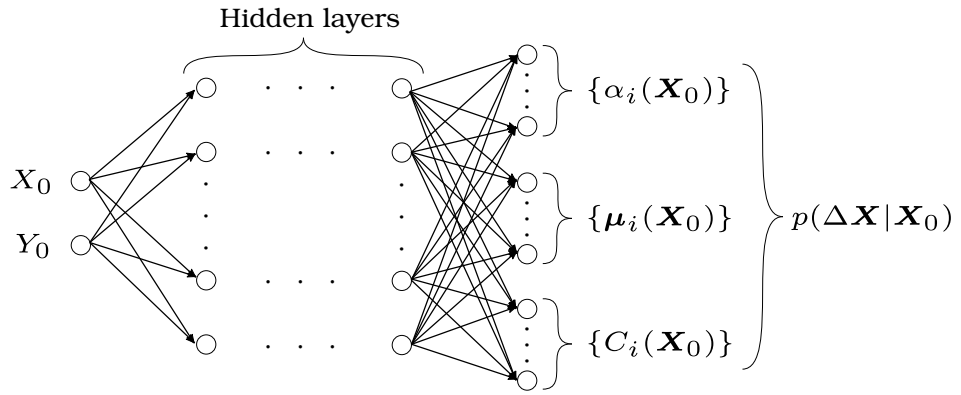


Figure 3.4: Schematic of the MDN model of the single-particle transition density of drifters.

for $i > 1$, and

$$\boldsymbol{h}_1 = \tanh\left(W_1\boldsymbol{X}_0 + \boldsymbol{b}_1\right). \tag{3.15}$$

Here, $\boldsymbol{W}_i \in \mathbb{R}^{d_i \times d_{i-1}}$ and $\boldsymbol{b}_i \in \mathbb{R}^{d_i}$ are the weight and bias parameters corresponding to the i^{th} layer, having d_i neurons. Note that $\boldsymbol{w} = \{\boldsymbol{W}_i, \boldsymbol{b}_i\}$. The final layer has custom activation functions designed to enforce the natural constraints on the components of $\boldsymbol{\theta}$. In particular, the softmax activation function $a_{\text{sm}}(\boldsymbol{x}) = \exp(\boldsymbol{x}) \,/\, \sum_i \exp(x_i)$ is applied to the neural network outputs which correspond to the mixture component weights, $\boldsymbol{\alpha}$, to ensure that these are positive and satisfy the constraint (3.7). Each covariance matrix $\mathbf{C}_i$ is represented by the components of a lower triangular Cholesky factor — positivity of the diagonal elements is enforced by taking an exponential. When $N_c = 32$ we have $\dim(\boldsymbol{\theta}) = 192$, and the total number of neural network parameters, i.e. weights and biases, is $\dim(\boldsymbol{w}) = 690,880$. We train the model by minimising the negative log likelihood loss function (3.8) using the Adam algorithm (Kingma & Ba 2015). We note that the number of widths of hidden layers are further hyperparameters which we have chosen after experimentation with test problems. We do not attempt to find optimal values for these in this work.

As is common in machine learning, we standardise the data before training (LeCun et al. 2012), that is we transform both the input data, $\{\boldsymbol{X}_{0i}\}$, and output data, $\{\Delta\boldsymbol{X}_i\}$, separately, by subtracting the mean of the training data and dividing each component by its standard deviation in the training data, so that each component of the transformed data has zero mean and unit variance. While theoretical justifications for this practice are lacking or unsatisfactory, we found that it did improve noticeably the numerical stability of the optimisation procedure. In any case, the transformation that we apply is invertible, although care must be taken to correctly invert the rescaling of the transition density. For example, if we denote the standardised variables by $\widetilde{\boldsymbol{X}_0}$ and $\widetilde{\Delta\boldsymbol{X}}$, then the model approximates $p(\widetilde{\Delta\boldsymbol{X}} \mid \widetilde{\boldsymbol{X}_0})$, and we can recover the

transition density with the correct units as

$$p(\Delta \boldsymbol{X} \mid \boldsymbol{X}_0) = \frac{p\left(\widetilde{\Delta \boldsymbol{X}} \mid \widetilde{\boldsymbol{X}_0}\right)}{\widehat{\mathrm{std}}(\Delta X)\ \widehat{\mathrm{std}}(\Delta Y)}, \tag{3.16}$$

where $\widehat{\mathrm{std}}(\,\cdot\,)$ denotes the sample standard deviation among the training data.

One aspect of neural networks that is particularly relevant to the problem at hand, is that they struggle to represent periodic functions (Liu et al. 2020). Given that we operate in longitude–latitude coordinates, a model of the transition density ought to be periodic in longitude. However, since the neural network model receives the initial position $\boldsymbol{X}_0$ as simply a vector in $\mathbb{R}^2$, the concept of a spherical domain is not built in to the representation. Indeed, the MDN model produces discontinuities in $p(\Delta \boldsymbol{X} \mid \boldsymbol{X}_0)$ at the dateline due to model error on either side. To improve continuity at the dateline we employ a crude technique, wherein we replicate the data twice, once shifted by 360° longitude west, and once shifted by 360° east.

The model is implemented in Python using TensorFlow (TensorFlow Developers 2021) and TensorFlow Probability (TensorFlow Probability Developers 2021) and trained using four NVIDIA Tesla V100 16GB GPUs in parallel. Of the 50% of data used to construct the model, 90%, again chosen randomly, was used to inform iterations of the optimisation procedure, and 10% was used for early stopping — we refer to these portions of the data as the training and test sets, respectively. Training took approximately 90 minutes. The evolution during training of the loss function on training and test sets is shown in Figure 3.5 as a function of epoch. An epoch is the number of iterations taken for all data to be used once in the Adam algorithm. The stopping criterion used for the optimisation, an example of early stopping, was that the test loss had not decreased since 50 epochs previous.

3.4.3 Model evaluation and comparison

Since the MDN model is probabilistic, its performance should be assessed using skill scores for probabilistic forecasts, as opposed to performance metrics commonly used for deterministic models, such as the mean squared error. As discussed above, minimising the negative log likelihood is equivalent to maximising the log score, since this is exactly the log likelihood. The log score has attractive properties, namely strict propriety (Bröcker & Smith 2007) and locality (Du 2021). Indeed it is the only smooth local strictly proper scoring rule for continuous variables up to affine transformation (Bernardo 1979). A scoring rule is strictly proper if its expectation (with respect to data) is maximised uniquely by the correct/perfect model (assuming it exists). A scoring rule is local if it is a function only of the value of the forecast probability distribution evaluated at the observed data, and does not depend for example on other features of the forecast distribution, such as its shape. For validation purposes we can compute the log score on our validation data set. However, while the value of the log score can be easily interpreted in the case of forecasts of discrete/categorical variables,

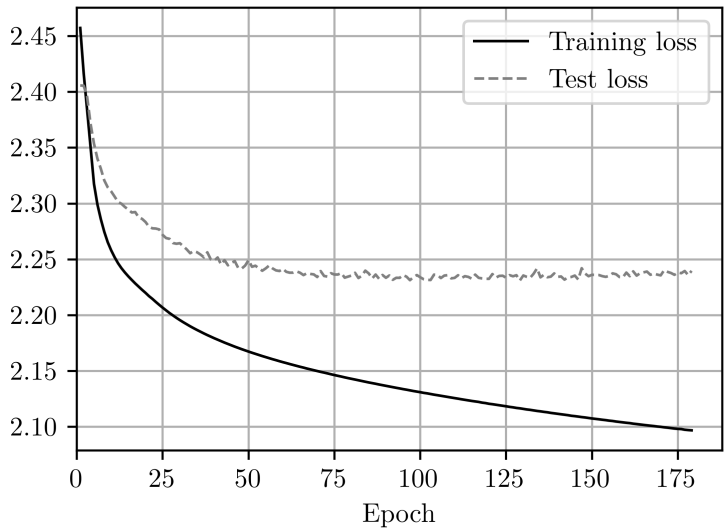


Figure 3.5: Evolution of the training and test loss in the MDN model during optimisation. The loss shown is the mean negative log likelihood per datapoint (i.e. a normalised form of (3.8)) in terms of the standardised variables $\widetilde{\boldsymbol{X}_0}$ and $\widetilde{\Delta \boldsymbol{X}}$.

its value in the case of continuous variables is not immediately meaningful, since it refers to probability density, which has dimensions inverse to the area of its support, meaning that the scale of the log score is problem-dependent. On the other hand, the log score can be more easily interpreted when used as a relative score between models — in particular, the mean difference of log scores between models reflects the average additional probability the first model places on observed outcomes compared to the other model, measured in units of information, nats (or shannons when $\log_2$ is used in the definition of the score). The difference of log scores is invariant under smooth transformations of the forecast variable (Du 2021); this means, in particular, that differences in log scores are unaffected by a change of units. Thus, in order to evaluate the MDN model we compare it with alternative models. We describe two alternative models, one used extensively in the literature, and one proposed here as a simple but reasonable alternative. We also compare with a simplified version of the MDN model, which features only one mixture component, i.e. for which $N_c = 1$. The log score of all models is computed on both training and validation data to assess relative performance.

Transition matrix model

Previous work (Maximenko et al. 2012, van Sebille et al. 2012, Miron et al. 2017, 2021), modelled drifter dynamics with a discrete-time Markov chain using Ulam's method (Ulam 1960, Froyland 2001). This requires to discretise space into bins $\{B_i\}$

and estimate the transition matrix

$$P_{ij} = \mathbb{P}(\boldsymbol{X}_{n+1} \in B_j \mid \boldsymbol{X}_n \in B_i), \tag{3.17}$$

which is the discrete analogue of the transition density (3.10). Indeed the primary difference between a Markov chain model and our Markov process model is that ours is continuous in space. The elements of the transition matrix are usually estimated by the standard approach sketched in Figure 3.1, where we have $\boldsymbol{Y} = \boldsymbol{X}_{n+1}$ and $f(\boldsymbol{Y}) = \mathbb{1}_{B_j}(\boldsymbol{X}_{n+1})$ — this corresponds to the maximum-likelihood estimate for each P_{ij} and, hence, maximises the log score on the training dataset. Note that the transition matrix can be used to construct a corresponding transition density which is piecewise constant on gridcells in $\boldsymbol{X}_n$ and $\boldsymbol{X}_{n+1}$ via

$$p(\Delta\boldsymbol{X} \mid \boldsymbol{X}_0) = \frac{P_{ij}}{A(B_j)}, \quad \text{when } \boldsymbol{X}_0 \in B_i,\ \boldsymbol{X}_0 + \Delta\boldsymbol{X} \in B_j, \tag{3.18}$$

where $A(B_j)$ is the area[2] of B_j. This is important for allowing comparison with models which are continuous in space.

An advantage of Markov chain models is that analysis of their long time behaviour is straightforward — the left and right eigenvectors of the transition matrix can be studied to identify almost-invariant sets, as in Miron et al. (2017). This has been called the eigenvector method (Froyland et al. 2014). The extension of this analysis to the continuous-space setting using our model, which we leave for future work, requires the calculation of eigenfunctions of the relevant Perron–Frobenius operator $\mathcal{P}$, which acts on probability density functions to evolve them forward in time, such that

$$p(\boldsymbol{X}_{n+1}) = \mathcal{P}(p(\boldsymbol{X}_n)) \tag{3.19}$$

$$:= \int_\Omega p(\boldsymbol{X}_n)\, p(\boldsymbol{X}_{n+1} \mid \boldsymbol{X}_n)\, \mathrm{d}\boldsymbol{X}_n. \tag{3.20}$$

Alternatively, it is worth noting that the MDN model can be used to construct a transition matrix, by numerical integration of the transition density, that is by computing numerically

$$P_{ij} = \int_{B_i} \int_{B_j} p(\boldsymbol{X}_{n+1} \mid \boldsymbol{X}_n)\, \mathrm{d}\boldsymbol{X}_{n+1}\, \mathrm{d}\boldsymbol{X}_n. \tag{3.21}$$

In Figure 3.6 we show the log transition density $\log p(\Delta\boldsymbol{X} \mid \boldsymbol{X}_0)$ derived from the transition matrix model via (3.18) for two different initial positions $\boldsymbol{X}_0$. The first is located within the core of the Gulf Stream at 34.85° N, 74.50° W, and the second is just outside the Gulf stream at 33.67° N, 72.55° W. Notice that in each case the support of the density is the set of grid cells to which transitions were observed in the training

[2]For consistency with the transition density as given by the MDN model, these areas must be calculated in terms of the same variables, i.e. degrees longitude by degrees latitude.

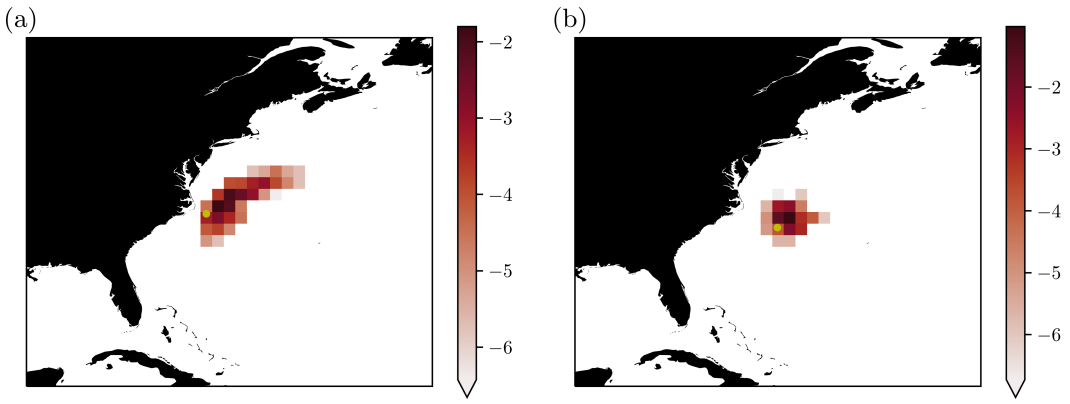


Figure 3.6: Maps of the log transition probability density function, $\log p(\Delta \boldsymbol{X} \mid \boldsymbol{X}_0)$, for initial positions, $\boldsymbol{X}_0$, (a) in the Gulf Stream (34.85° N, 74.50° W), and (b) adjacent to the Gulf Stream (33.67° N, 72.55° W), derived from the transition matrix model with $\tau = 4$ days via (3.18). Yellow dots indicate $\boldsymbol{X}_0$.

data. In other words, transitions to other grid cells have probability zero under the model. We return to this point in section 3.4.3.

Gaussian transitions with gridded parameters (GTGP)

A simple model for the transition density (3.11) is that, given initial positions $\boldsymbol{X}_0$, transitions are conditionally Gaussian with conditional mean and covariance given by functions of $\boldsymbol{X}_0$ which are piecewise constant on grid cells, i.e.

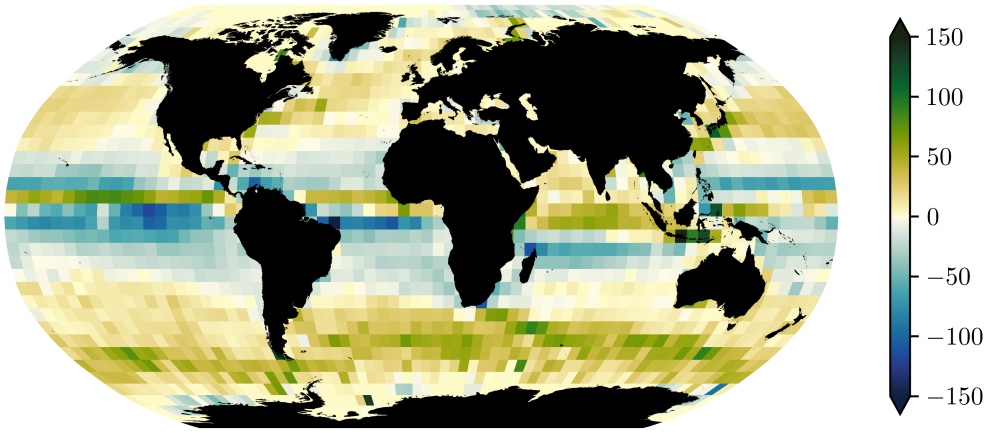
$$\Delta \boldsymbol{X} \mid \boldsymbol{X}_0 \sim \mathcal{N}\left(\boldsymbol{\mu}\left(\boldsymbol{X}_0\right), \mathbf{C}\left(\boldsymbol{X}_0\right)\right), \tag{3.22}$$

with $\boldsymbol{\mu}\left(\boldsymbol{X}_0\right)$ and $\mathbf{C}\left(\boldsymbol{X}_0\right)$ piecewise constant in $\boldsymbol{X}_0$. The parameters $\boldsymbol{\mu}$ and $\mathbf{C}$ are estimated by sorting the observations into bins and computing the sample mean and sample covariance for each bin. The sample mean is the maximum likelihood estimate of $\boldsymbol{\mu}$, while the sample covariance differs from the maximum likelihood estimate of $\mathbf{C}$ only by a factor of $\frac{N-1}{N} \approx 1$, where N is the number of training data in the given bin. Hence, the parameter estimates used are very close to those which maximise the log score on training data.

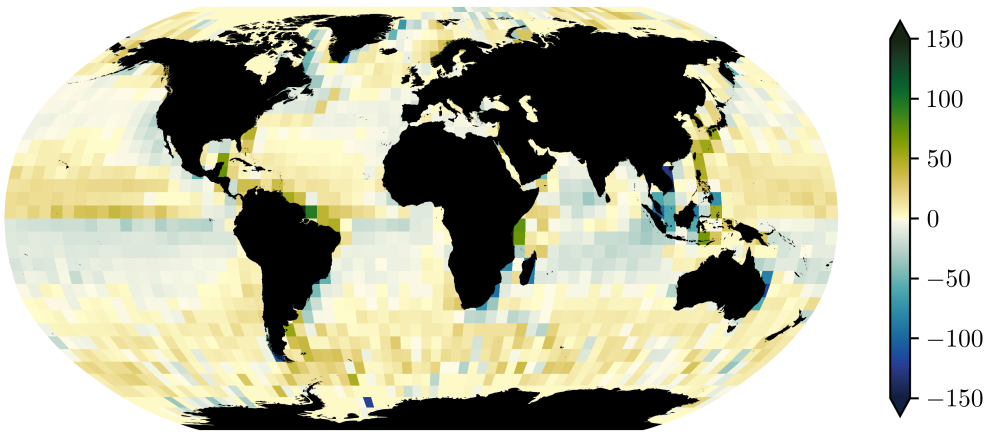
Figure 3.7 shows the mean of displacements from a GTGP model with a regular $1° \times 1°$ longitude–latitude grid and $\tau = 4$ days, as a function of initial position.

Model scores

We compute skill scores for the full training and validation datasets with global coverage, and for regions A, B, and C. For both the transition matrix and GTGP models it is necessary to choose a spatial discretisation; herein we consider only square latitude–longitude grids, so that the only parameter to be chosen is the grid cell side length.



(a) Mean of 4-day zonal displacement (km).



(b) Mean of 4-day meridional displacement (km).

Figure 3.7: Mean of displacements from the GTGP model, with $\tau = 4$ days, as a function of initial position.

The choice of grid cell size affects the performance of these models. If a relatively high resolution discretisation is used, the models attain relatively high scores in training, but generalise poorly, as reflected in poor scores on validation data. In the case of the GTGP model, an issue arises when validation data falls in grid cells not visited by drifters in the training set, since sample means and covariances cannot be estimated in bins where data is absent. As a simple solution, we set the value of $\boldsymbol{\mu}$ and $\mathbf{C}$ on unvisited grid cells equal to a global (or regional) estimate. A flaw of the transition matrix model, with the transition matrix estimated as discussed above, is that validation data can have zero probability under the model, and hence achieve a log score of minus infinity. This situation is avoided by taking sufficiently large grid cells, but this leads to exceptionally low scores. On the other hand, a validation score can be computed with smaller grid cells if one is prepared to simply discard validation data which have zero probability under the model. This seems overly generous, as the transition matrix model will be scored increasingly highly as the grid cell size is reduced to zero and an increasing number of the validation data are neglected. As a compromise, we fix the grid cell size for the transition matrix model to $1^\circ \times 1^\circ$, the resolution used in some previous studies (van Sebille et al. 2012) where the transition matrix model was used, and discard validation data with zero probability — the proportion of validation data discarded was 7% globally, and 2%, 9% and 4% in regions A, B and C, respectively. For the GTGP model the grid cell size was optimised to maximise validation scores using grid search cross validation: a common procedure which amounts to trying a range of values of a model hyperparameter (in this case the grid cell size) and choosing the value which optimises the validation score. The optimal grid cell size found ranged from 1.1° in region A to 5° globally. The scores are presented in table 3.1. In all regions the MDN models outperform the alternatives, with the 32-component model achieving slightly higher scores than the single-component model. Note that the scores reported happen to be negative — this is not by convention, but instead reflects that log probability densities are often negative. A higher score is a better score.

Although latitude–longitude grids have been the most commonly used discretisation for binning drifter observations (van Sebille et al. 2012, Maximenko et al. 2012, McAdam & van Sebille 2018, Miron et al. 2017), alternative methods of binning have been employed in some works. In particular, Koszalka et al. (2011) used a nearest-neighbour clustering algorithm to group drifter data when estimating mean velocity and diffusivity — this can be understood as discretising the ocean according to the corresponding Voronoi diagram — leading to more evenly sampled bins; alternatively, O'Malley et al. (2021) used a hexagonal mesh, leading to more evenly sized bins compared to standard latitude–longitude bins, whose areas decrease with latitude. While it is likely that a more carefully chosen discretisation would lead to improved scores for the TM and GTGP models, the problem of undersampled bins and transitions can be expected to persist.

		TM	GTGP	MDN1	MDN32
Global:	Training	−1.61	−1.20	−1.07	−1.02
	Validation	−1.74	−1.30	−1.14	−1.11
A:	Training	−1.78	−1.25	−1.35	−1.30
	Validation	−1.89	−1.42	−1.40	−1.35
B:	Training	−1.95	−1.90	−1.91	−1.85
	Validation	−2.16	−2.01	−2.01	−1.96
C:	Training	−1.93	−1.59	−1.63	−1.56
	Validation	−2.00	−1.66	−1.61	−1.57

Table 3.1: Training and validation scores for the transition matrix and GTGP models, as well as the single-component MDN and full 32-component MDN models, in each of the regions considered (see the map in Figure 3.3). The scores are the mean log score (i.e. mean log likelihood) per datapoint calculated in terms of the variables $\boldsymbol{X}_0$ and $\Delta\boldsymbol{X}$ in their original degrees longitude/latitude units.

3.4.4 Results

Once trained, the model can be used in at least two ways: (i) to derive estimates of single-particle displacement statistics, and (ii) to simulate drifter trajectories. However, we first examine the transition density directly. In Figure 3.8 we show the log transition density $\log p(\Delta\boldsymbol{X} \mid \boldsymbol{X}_0)$ for two different initial positions $\boldsymbol{X}_0$. In the first case, where $\boldsymbol{X}_0$ is located within the core of the Gulf Stream at 34.85° N, 74.50° W, the transition density is strongly non-Gaussian, with contours extending roughly to the south and northeast, showing the influence of the Gulf Stream on drifters. In the second case, where $\boldsymbol{X}_0$ is just outside the Gulf stream at 33.67° N, 72.55° W, the transition density is closer to Gaussian.

In order to quantify the extent to which the transition density deviates from being Gaussian, and how this varies from one region of the ocean to another, we computed the Kullback–Leibler (KL) divergence[3] of the single-component MDN model, which is Gaussian, from the full 32-component model as a function of initial position. The result is shown in Figure 3.9. Note that, since a closed-form expression for the KL divergence between two Gaussian mixtures is not known (Cui & Datcu 2015), we provide simple Monte Carlo estimates based on 5000 samples at each of the vertices of a $1° \times 1°$ grid. Where the KL divergence is zero, the two models agree exactly, indicating that displacements are Gaussian. The larger the KL divergence is, the greater the disagreement between the models, and the further from Gaussian the full model is. As a point of reference for interpreting the magnitude of the KL divergence, note that the if $Z_0 \sim \mathcal{N}(m_0, 1)$ and $Z_1 \sim \mathcal{N}(m_1, 1)$, then, writing their pdfs as p_0 and p_1, $D_{\mathrm{KL}}(p_1 \parallel p_0) = (m_1 - m_0)^2$. Non-Gaussianity of displacements is likely due primarily

[3]The KL divergence of p from q, also known as the relative entropy, defined $D_{\mathrm{KL}}(q \parallel p) = \int q(x) \log \frac{q(x)}{p(x)} \, \mathrm{d}x$, is a measure of the divergence of a probability density p from a reference probability density q — often interpreted as the amount of information lost when p is used to approximate q.

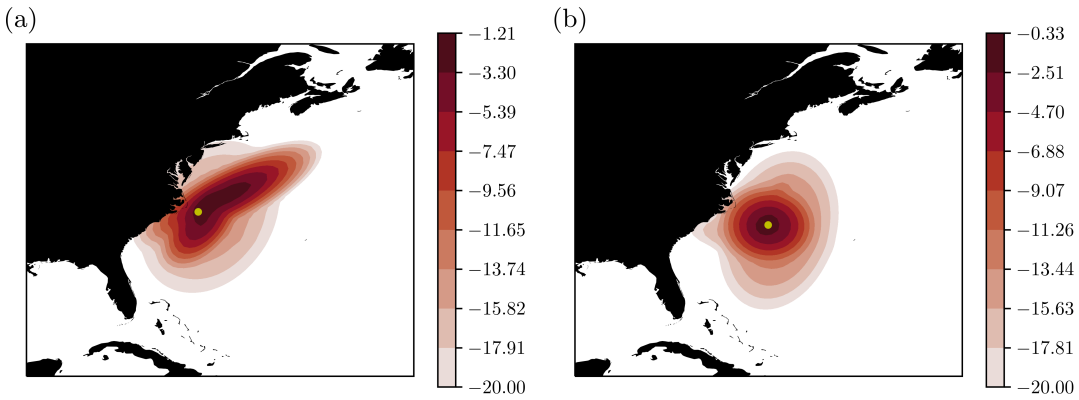


Figure 3.8: Maps of the log transition density, $\log p(\Delta \boldsymbol{X} \mid \boldsymbol{X}_0)$, for initial positions, $\boldsymbol{X}_0$, (a) in the Gulf Stream (34.85° N, 74.50° W), and (b) adjacent to the Gulf Stream (33.67° N, 72.55° W), derived from the MDN model with $\tau = 4$ days. Yellow dots indicate $\boldsymbol{X}_0$.

to inhomogeneity of ocean velocities — drifters can explore a range of flow statistics as they move, and the convolved effects of these are reflected in observed displacements. An alternative explanation is that the underlying velocity field is non-Gaussian — evidence of non-Gaussian velocities in the North Atlantic has been presented by Bracco et al. (2000) and LaCasce (2005) on the basis of observations from both subsurface current meters and subsurface floats.

As can be seen in Figure 3.8, the model assigns nonzero probability to drifter displacements intersecting land. This is unavoidable given that the support of the assumed parametric form, that of a Gaussian mixture, extends to infinity; moreover, this may not be entirely spurious, given that some drifters do run aground. In 2012 Lumpkin et al. (2012) reevaluated drifter data to study the causes of drifter deaths. They concluded that approximately 27% of drifter deaths were due to running aground, with a further 10% being picked up by humans, and the remainder failing due to internal faults. Outside of coastal regions this issue is unlikely to have a strong effect on the estimates of displacement statistics considered in section 3.4.4. The implications for drifter simulations are discussed further in section 3.4.4.

Displacement statistics

In this section we present maps of single-particle statistics derived from the model. As a first example, we show the mean of displacements over the 4-day time increment of our model. We further provide global estimates of lateral diffusivity.

Figure 3.10 shows the mean of drifter displacements as a function of initial position. While the output of the model is in longitude–latitude coordinates (λ, ϕ), we apply a simple conversion to kilometres based on a local tangent-plane approximation

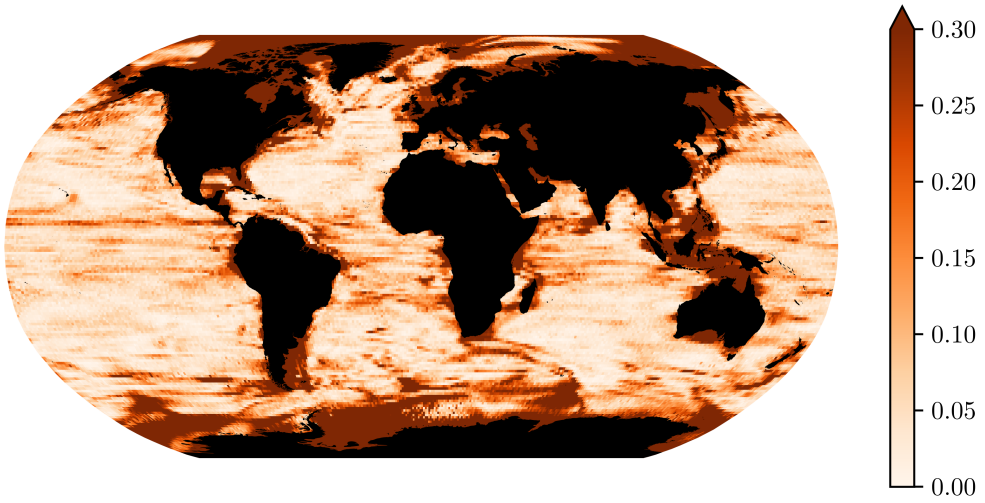


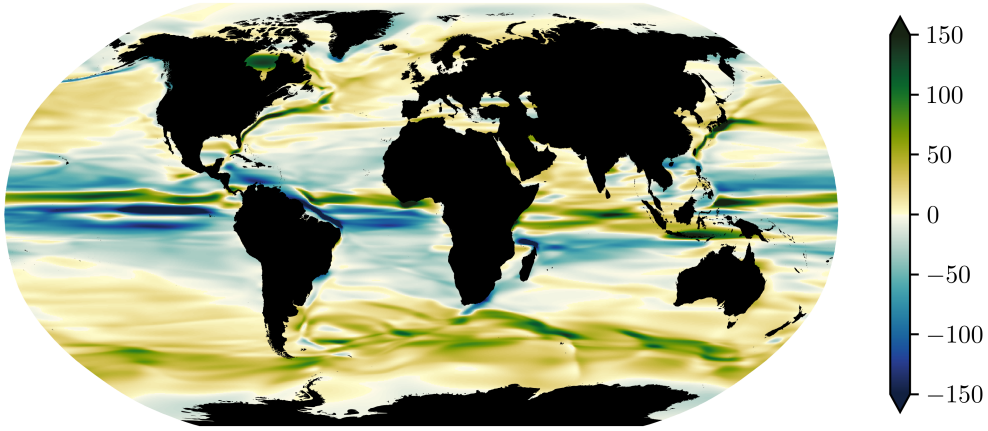
Figure 3.9: Kullback–Leibler divergence of the single-component MDN model from the full 32-component MDN model, as a function of initial position. Larger values indicate stronger deviations from Gaussianity in displacements.

$$\Delta X = R\,\Delta\phi \tag{3.23a}$$
$$\Delta Y = R\,\Delta\lambda\,\cos\phi_0, \tag{3.23b}$$

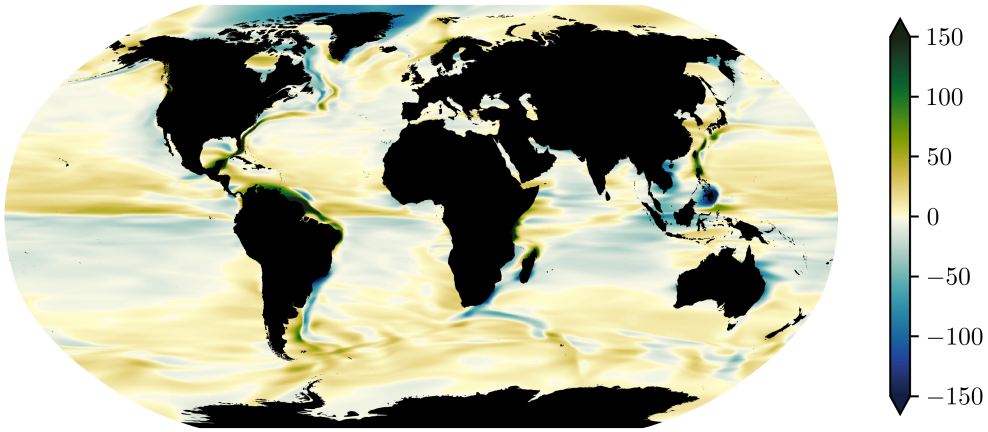
where R is the radius of the Earth at the equator. The imprint of several features of the surface dynamics, such as the western boundary currents and equatorial (counter) currents, is clear.

For the sake of comparison with previous work, we consider the estimation of lateral diffusivity from our model, though we emphasise that by modelling the full transition density, we provide a more accurate description of Lagrangian statistics than can be captured by the familiar advective-diffusive model of dispersion put forward by Davis (1987, 1991). The estimation of ocean diffusivity by various methods has been the subject of numerous papers (Oh et al. 2000, Zhurbas & Oh 2003, 2004, Klocker et al. 2012, Abernathey & Marshall 2013, Klocker & Abernathey 2014, Ying et al. 2019). The estimation of diffusivity from drifter displacements is straightforward only when there exists a suitable sampling time, which is larger than the time for drifter velocities to decorrelate, i.e. for drifter motion to become diffusive, and such that the scale of drifter displacements over that time scale is small relative to spatial variations in the diffusivity. In this case a simple estimate of the lateral diffusivity tensor $\mathbf{K}(\boldsymbol{x})$ is

$$\mathbf{K}(\boldsymbol{x}) = \frac{1}{2\,\tau}\,\mathrm{Cov}(\Delta\boldsymbol{X} \mid \boldsymbol{X}_0 = \boldsymbol{x}), \tag{3.24}$$



(a) Mean of 4-day zonal displacement (km).



(b) Mean of 4-day meridional displacement (km).

Figure 3.10: Mean of displacements from the MDN model, with $\tau = 4$ days, as a function of initial position.

where τ is the suitably chosen time scale, and the conditional covariance is estimated by either one of the approaches sketched in Figure 3.1. Unfortunately, such a time scale may not exist in the ocean, and, if it does exist, it likely varies in space, making its determination difficult. This challenge has been borne out in previous studies (LaCasce et al. 2014, Zhurbas et al. 2014). Oh et al. (2000) proposed a method to circumvent the issues created by inhomogeneity. They proposed to isolate the cross-flow component of the displacement covariance, identified by the minor principal component (the smaller eigenvalue of displacement covariance), and use this to provide a scalar estimate of diffusivity, since the cross-flow component is less affected by shear in the mean flow. In Figure 3.11(a) we provide a similar estimate, derived from the MDN model with $\tau = 4$ days,

$$K(\boldsymbol{x}) = \frac{1}{2\,\tau}\,\lambda_2(\boldsymbol{x}), \tag{3.25}$$

where $\lambda_2(\boldsymbol{x})$ is the smallest eigenvalue of

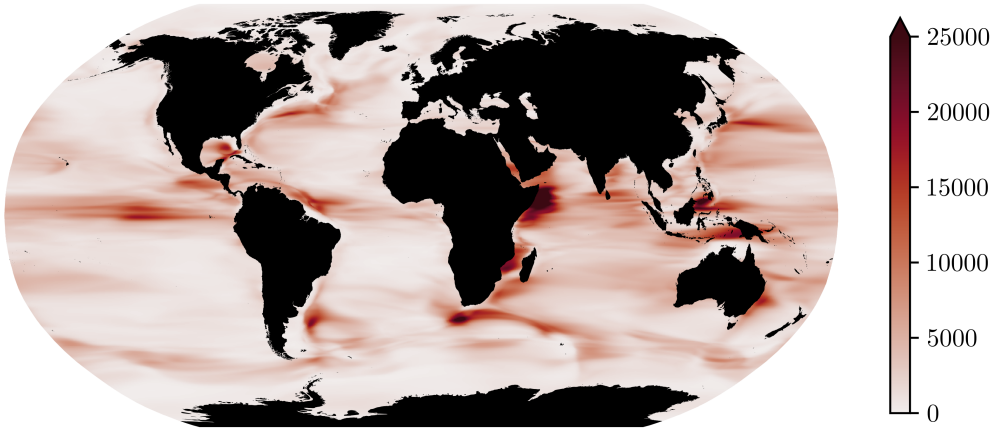
$$\mathrm{Cov}(\Delta \boldsymbol{X} \mid \boldsymbol{X}_0 = \boldsymbol{x}) = \sum_i \alpha_i \mathbf{C}_i + \sum_i \left[\alpha_i \left(\boldsymbol{\mu}_i - \sum_i \alpha_i \boldsymbol{\mu}_i \right) \left(\boldsymbol{\mu}_i - \sum_i \alpha_i \boldsymbol{\mu}_i \right)^T \right]. \tag{3.26}$$

The result agrees very well with estimates provided by Zhurbas & Oh (2004) for the Atlantic and Pacific oceans. Figure 3.11(b) shows the difference of estimates of the form (3.25) with $\tau = 14$ days and $\tau = 4$ days, respectively. In many areas, the diffusivity estimates are slightly amplified by taking a larger time lag τ, with greater differences visible in some particularly energetic regions; however, the effect is indeed much weaker than that observed with analogous along-flow diffusivity estimates derived from the largest eigenvalue of the displacement covariance matrix.

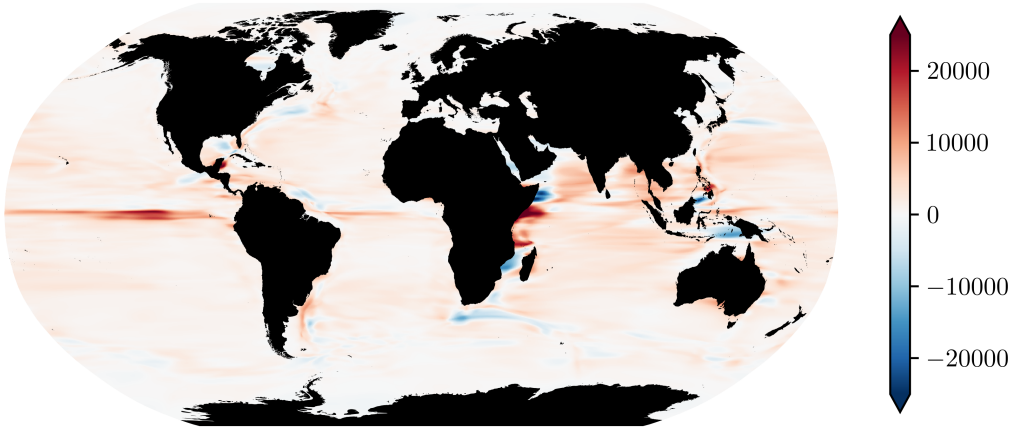
Leaving aside the challenges of estimating diffusivity from displacements, which are common to all methods, we highlight as this point some advantages to our approach. Using the MDN model, trained with maximum likelihood and effectively regularised by the use of early-stopping, removes the difficulty of tuning the resolution of bins. Instead, the effective resolution of our mean displacement and diffusivity estimates is set automatically by the resolution of the data, and is free to vary optimally in space. This allows us to produce at once global estimates, which resolve well-sampled flow features very well and are forgiving in regions where data is relatively sparse, with the exception of very high-latitude regions, where there is simply no data to constrain the model.

Drifter simulations

In this section we demonstrate the simulation of drifter trajectories using the MDN model as the basis for a discrete time Markov process model. In a discrete-time setting, assuming Markovianity means assuming that $p(\boldsymbol{X}_{n+1} \mid \boldsymbol{X}_n, \boldsymbol{X}_{n-1}, \cdots) = p(\boldsymbol{X}_{n+1} \mid \boldsymbol{X}_n)$. In this case, sampling trajectories amounts to repeatedly sampling displace-



(a) Scalar estimate of lateral diffusivity, $K^{(4)}$ ($m^2 s^{-1}$), derived from the MDN model with $\tau = 4$ days.



(b) Difference of scalar estimates of lateral diffusivity, $K^{(14)} - K^{(4)}$, derived from MDN models with $\tau = 14$ days and $\tau = 4$ days, respectively.

Figure 3.11: Global estimate of lateral diffusivity derived from the MDN model of the transition density.

ments in sequence according to the transition density, since, given the current position, displacements are statistically independent of previous positions.

A complication of simulating drifters in this way is that, for reasons discussed above, drifters can hit land. In this work we do not attempt to model the beaching of drifters, since it is not clear that the Global Drifter Program dataset contains sufficient reliable information — in particular, it remains a challenge to determine whether drifters have run aground or not (Lumpkin et al. 2012). To exclude the possibility of running aground in our drifter simulations we implement a simple rejection sampling scheme, wherein displacements sampled from the transition density which would bring a drifter on land are rejected, and a new displacement is sampled until a displacement which keeps the drifter in the ocean is drawn. This amounts to sampling according to the conditional density $p(\Delta \boldsymbol{X} \mid \boldsymbol{X}_0, \boldsymbol{X}_0 + \Delta \boldsymbol{X} \notin \text{land})$, and is equivalent to the standard practice when using transition matrix models of restricting the domain considered to the ocean and normalising probability estimates correspondingly. To determine whether a proposed new position is on land, we check intersection with a 110m-resolution land mask.

We simulated the evolution of a set of drifters initialised on the vertices of a $2^\circ \times 2^\circ$ grid for a period of 10 years. Note that the evolution of each drifter is simulated independently. This means that multi-particle statistics that would characterise the joint evolution of drifters released simultaneously in the ocean are not represented and, in particular, that the current model is not appropriate for simulating the release of a cloud of tracer particles on short time scales; however, it can be expected to represent the behaviour of drifters or buoyant tracers over large spatial and temporal scales. Similar experiments, carried out by Maximenko et al. (2012) and van Sebille et al. (2012) using transition matrix models trained on Global Drifter Program data, studied the clustering of simulated drifters due to near-surface convergence and the formation of so-called garbage patches, including the North Pacific Garbage Patch (Moore et al. 2001) and others corresponding to the other subtropical ocean gyres. The simulations of van Sebille et al. (2012) showed a further cluster in the Barents Sea which formed only after several decades.

The results of our model simulation are largely in agreement with these previous studies. The distribution of the simulated drifters is shown in Figure 3.12 at the beginning of the simulation and after one, three, and ten years of evolution under the MDN model. After one year the drifters have become relatively sparse in equatorial regions. After three years clusters in the subtropical gyres have begun to appear, and after ten years, these are very well defined. Smaller clusters are also seen to appear, notably in the North Sea and in the seas south of Papua, as well as in some high latitude regions including along the west coast of Greenland and off Antarctica around 100 – 130° E.. Validating these clusters, that is, assessing whether marine debris is likely to accumulate in these areas, is difficult, because in situ observations remain sparse (Ryan et al. 2009). It may be that the dynamics in these regions, which are poorly sampled by GDP drifters, are simply underresolved by the MDN model, leading to spurious convergence zones. We note, for example, in Figure 3.10 that mean displacements do

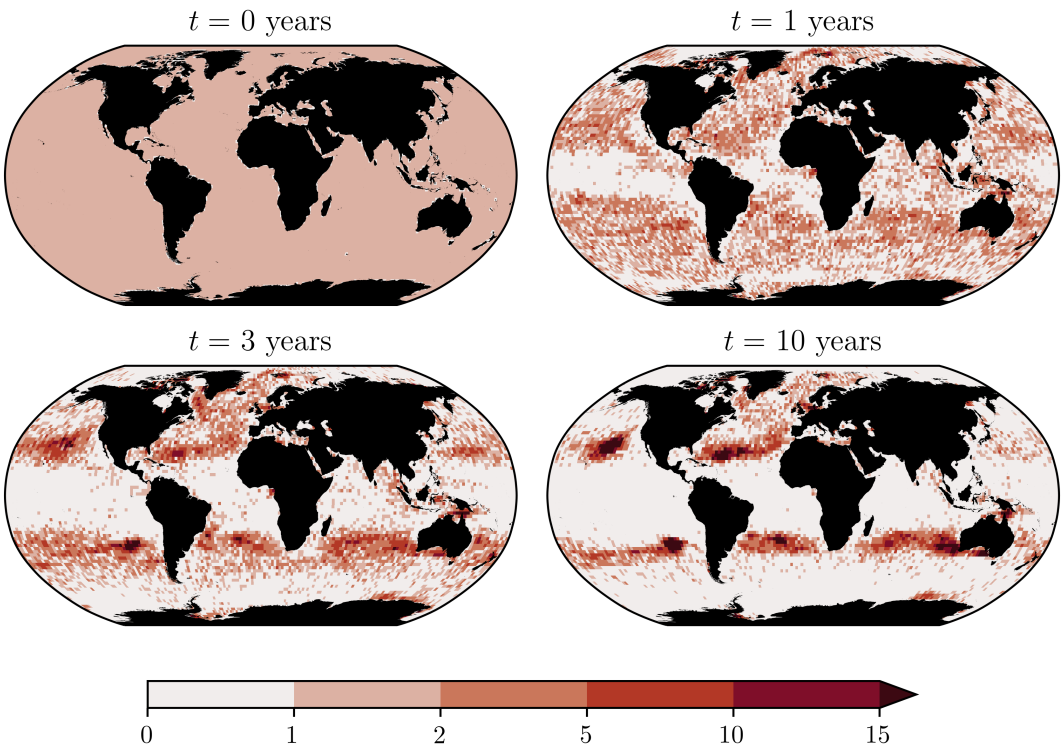


Figure 3.12: Histograms of simulated drifters initially and after one, three, and ten years of evolution under the MDN model, respectively.

not appear to represent the detail of known currents in the southern portion of the North Sea, which is not visited by drifters in the GDP data (see Figure 3.2). In general, as is true for any data-driven model, caution should be exercised when interpreting model outputs in regions where data is lacking.

3.5 Conclusions

This work demonstrates the use of conditional density estimation, and, in particular, stochastic neural networks, in a fluid dynamical problem, namely that of diagnosing single-particle statistics from trajectory data. We show how such probabilistic models are useful both as emulators, and as an indirect means of estimating conditional statistics. By operating in the framework of probabilistic modelling we are able to appeal to the extensive literature on statistical inference, probabilistic forecasting, model comparison and validation, and thereby avoid ad hoc choices of loss functions and performance metrics. Our model is compared, using a probabilistic scoring rule, to alternative models, including a Markov chain model used extensively in the literature, and is shown to outperform these, both globally and in three specific regions.

By modelling the single-particle transition density of surface drifters, we gain estimates of a range of conditional statistics simultaneously, which capture the occurrence of strongly non-Gaussian statistics in some areas of the ocean. We provide global maps of mean displacement and lateral diffusivity, but emphasise that these examples provide only a limited summary of the information contained in the transition density; further statistics, including higher moments of displacements can readily be computed from our model. Interpreted as the basis for a discrete-time Markov process, our model is also used to simulate the evolution of a set of drifters seeded globally on a uniform grid, and shows the emergence of clusters of drifters in the subtropical gyres, in agreement with previous work on the formation of garbage patches.

The approach espoused in this work is equally applicable to other problems in fluid dynamics and oceanography. One example is the estimation of structure functions from either Eulerian or Lagrangian velocity data. Another is the estimation of multi-particle statistics, such as relative dispersion, via modelling of multi-particle transition densities. Yet another is the learning of stochastic parameterisations in climate/atmosphere/ocean models. Guillaumin & Zanna (2021) made progress on the parameterisation of subgrid momentum forcing in an ocean model with a single-component MDN model, but the approach is applicable more broadly, e.g. to the parameterisation of subgrid transport.

In this work we have largely neglected the need to quantify uncertainty in model parameters and to incorporate prior knowledge in our modelling. These needs would be met by a Bayesian approach, where, instead of estimating parameters by maximum likelihood, we apply Bayesian inference to obtain posterior distributions on parameters, which account for prior knowledge. Indeed, all of the results presented herein would benefit from uncertainty quantification. In the case of conditional statistics, a Bayesian approach would, e.g., allow to identify where there is not enough data

to inform reliable estimates of lateral diffusivity; and in general, incorporating prior knowledge may help to regularise our model of the transition density, so that, in the case of drifter simulations, spurious convergence zones can be avoided. The application of Bayesian inference to MDNs remains challenging, but we consider this an important future direction.

Data availability statement. The code required to reproduce the results in this chapter is available at doi.org/10.5281/zenodo.7737161, along with the trained MDN model and a Jupyter Notebook which demonstrates its use. The processed GDP data used and drifter simulation data are available at doi.org/10.7488/ds/3821.

Chapter 4

Quantifying uncertainty in ocean dynamical statistics with Bayesian neural networks

4.1 Introduction

In Chapter 3 we saw an example of the use of a probabilistic neural network model. There we trained the model using maximum likelihood estimation and took the resulting parameter estimates as our best guess. We then proceeded to study outputs of the model and take these as a proxy for information about the real ocean. But how confident should we be that our model is representative of reality? Should we trust its predictions? In order to answer these questions we are forced to acknowledge and quantify our uncertainty in model parameters, and perhaps also in model choice. To this end we again appeal to a Bayesian formalism, as in Chapter 2. Chapter 4 focusses on Bayesian neural networks (BNNs) which are probabilistic neural networks whose parameters are estimated using Bayesian inference — or more accurately, using various approximations to Bayesian inference. Our aim is to assess the suitability of state-of-the-art methods in Bayesian machine learning to problems in ocean transport modelling. We consider this a natural extension of the work in Chapter 3. The issues raised here are also important for the use of Bayesian neural networks more broadly in the natural sciences and beyond. We identify the primary challenges hindering progress with BNNs as the specification of meaningful prior distributions on network parameters and the development of methods for approximate Bayesian inference which are both accurate and affordable.

4.2 Bayesian neural networks

To fix ideas consider the multilayer perceptron (MLP) described in Section 3.4.2. The MLP, as the prototypical neural network architecture, is the basis for many advanced

machine learning models. Owing to their large number of parameters, neural networks are incredibly flexible models. The redundancy that comes with such over-parameterised models means that they can be continually improved as more data become available. Indeed it is in the large-data regime that such models are most useful. This is the context in which such models were originally proposed and where they continue to enjoy reputed success. However, in the natural sciences data is often not so abundant. Moreover, in a scientific context, the cost associated with making inaccurate predictions may well be higher. Central to building models that provide trustworthy predictions in the scientific domain is uncertainty quantification. Leaving aside the issue of model uncertainty, unknown parameters cannot usually be inferred with certainty from finite data. In Chapter 2 we thought of model parameters as physically meaningful quantities, such that our knowledge of their values was of interest in and of itself. The parameters of neural network models have no such status. Instead we are interested only in model predictions and the uncertainty thereon. Still, in order to quantify uncertainty on model predictions we must quantify uncertainty on neural network weights and biases.

To be explicit consider a neural network model $\mathcal{M}_{\text{NN}}$ with parameters $\boldsymbol{w}$ taking input $\boldsymbol{X}$. We denote the network's output by $\boldsymbol{f}_{\text{NN}}(\boldsymbol{X}\,;\,\boldsymbol{w})$ where $\boldsymbol{f}_{\text{NN}}(\,\cdot\,;\,\boldsymbol{w}) : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^{d_{\text{out}}}$ and d_{in} and d_{out} are the dimension of the neural networks input and output, respectively. A common view of such a model is that it can be trained to approximate deterministic functions. Indeed, so-called universal approximation theorems state that an MLP with non-polynomial activation functions can approximate any continuous function supported on a compact subspace of $\mathbb{R}^d$ arbitrarily well (in the sense of the supremum norm) provided that the network is sufficiently wide (Leshno et al. 1993) or sufficiently deep (Kidger & Lyons 2020). The training that leads to such approximations typically involves minimising (with respect to $\boldsymbol{w}$) a mean-squared-error (MSE) loss function

$$\mathcal{L}_{\text{MSE}}(\boldsymbol{w}\,;\,\mathcal{D}) = \sum_{i=1}^{N} (\boldsymbol{f}_{\text{NN}}(\boldsymbol{X}_i\,;\,\boldsymbol{w}) - \boldsymbol{Y}_i)^2 \tag{4.1}$$

given observations $\mathcal{D} = (\boldsymbol{X}_i,\, \boldsymbol{Y}_i)_{i=1}^N$ where $\boldsymbol{Y}_i$ represents an observation of $\boldsymbol{f}(\boldsymbol{X}_i)$. The choice of the MSE loss function can be justified merely as an arbitrary choice of vector norm (L^2) or by invoking a model of the observation process,

$$\boldsymbol{Y}_i = \boldsymbol{f}(\boldsymbol{X}_i) + \boldsymbol{\eta}_i \tag{4.2}$$

where $\boldsymbol{\eta}_i \overset{iid}{\sim} \mathcal{N}(\boldsymbol{0}, \mathbb{I})$, since in this case the MSE loss is proportional to the negative conditional log likelihood, $-\log p(\boldsymbol{Y}_i \mid \boldsymbol{X}_i)$, and its minimiser is equivalent to a maximum likelihood estimate of the parameters $\boldsymbol{w}$. These two justifications are suggestive of two different approaches to understanding neural networks — the first, an approximation-theoretic approach, which treats neural networks as deterministic models whose parameters are inferred from noisy data, and the second, a statistical

approach, where, for whatever reason, we may model $\boldsymbol{Y} \mid \boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{X}), \mathbb{I})$ and use $\boldsymbol{f}_{\text{NN}}(\,\cdot\,;\,\boldsymbol{w})$ to represent $\boldsymbol{\mu}(\,\cdot\,)$. This second view is easily generalised to arbitrary parametric conditional models

$$p(\boldsymbol{Y} \mid \boldsymbol{X}) = \rho(\boldsymbol{Y}\,;\,\boldsymbol{\theta}(\boldsymbol{X})), \tag{4.3}$$

where $\boldsymbol{f}_{\text{NN}}(\,\cdot\,;\,\boldsymbol{w})$ can be used to represent $\boldsymbol{\theta}(\,\cdot\,)$; that is, where there exists a function $\boldsymbol{h} : \mathbb{R}^{d_{\text{out}}} \to \mathbb{R}^{\dim(\boldsymbol{\theta})}$ such that $\boldsymbol{h}(\boldsymbol{f}_{\text{NN}}(\,\cdot\,;\,\boldsymbol{w})) = \boldsymbol{\theta}(\,\cdot\,)$, and hence,

$$p(\boldsymbol{Y} \mid \boldsymbol{X}) = \rho(\boldsymbol{Y}\,;\,\boldsymbol{h}(\boldsymbol{f}_{\text{NN}}(\boldsymbol{X}))). \tag{4.4}$$

We refer to this class of model as probabilistic neural networks, of which the mixture density network model used in Chapter 3 is an example — in that case $\boldsymbol{X} = \boldsymbol{X}_0$ and $\boldsymbol{\theta}$ represents the parameters of the Gaussian mixture distribution. In the general case the MSE loss is replaced by the negative conditional log likelihood loss function

$$\mathcal{L}_{\text{NLL}}(\boldsymbol{w}\,;\,\mathcal{D}) = \sum_{i=1}^{N} -\log \rho(\boldsymbol{Y}_i\,;\,\boldsymbol{\theta}(\boldsymbol{X}_i)), \tag{4.5}$$

denoted simply as $\mathcal{L}$ in Chapter 3.

The statistical view of neural networks provides a clear framework for questions of uncertainty. In particular it permits a Bayesian approach, whence arise Bayesian neural networks (MacKay 1995, Neal 1996), the name given to probabilistic neural networks whose parameters have been inferred by a procedure approximating that of Bayesian inference. What motivates BNNs is the desire to quantify the uncertainty in the predictions of probabilistic neural networks. Thus our goal is to obtain, by way of a posterior distribution on network parameters $p(\boldsymbol{w} \mid \mathcal{D})$, some representation of the posterior uncertainty on $\boldsymbol{\theta}(\boldsymbol{X})$, and hence on $\boldsymbol{Y} \mid \boldsymbol{X}$. The following section discusses the difficulty of this task and current approaches.

4.3 Approximate inference

Given a probabilistic neural network $\mathcal{M}_{\text{NN}}$ and a prior on network weights $p(\boldsymbol{w} \mid \mathcal{M}_{\text{NN}})$ performing Bayesian inference amounts to computing the posterior distribution

$$\overbrace{p(\boldsymbol{w} \mid \mathcal{D}, \mathcal{M}_{\text{NN}})}^{\text{Posterior}} = \frac{\overbrace{p(\mathcal{D} \mid \boldsymbol{w}, \mathcal{M}_{\text{NN}})}^{\text{Likelihood}}\ \overbrace{p(\boldsymbol{w} \mid \mathcal{M}_{\text{NN}})}^{\text{Prior}}}{\underbrace{\int_{W} p(\mathcal{D} \mid \boldsymbol{w}, \mathcal{M}_{\text{NN}})\, p(\boldsymbol{w} \mid \mathcal{M}_{\text{NN}})\, \mathrm{d}\boldsymbol{w}}_{\text{Evidence}}}. \tag{4.6}$$

To be more explicit about the fact that we are interested in conditional modelling we may write

$$p(\boldsymbol{w} \mid \{\boldsymbol{X}_i, \boldsymbol{Y}_i\}_{i=1}^N, \mathcal{M}_{\text{NN}}) \tag{4.7a}$$

$$= \frac{p(\{\boldsymbol{X}_i, \boldsymbol{Y}_i\}_{i=1}^N \mid \boldsymbol{w}, \mathcal{M}_{\text{NN}})\, p(\boldsymbol{w} \mid \mathcal{M}_{\text{NN}})}{p(\{\boldsymbol{X}_i, \boldsymbol{Y}_i\}_{i=1}^N \mid \mathcal{M}_{\text{NN}})} \tag{4.7b}$$

$$= \frac{p(\{\boldsymbol{Y}_i\}_{i=1}^N \mid \{\boldsymbol{X}_i\}_{i=1}^N, \boldsymbol{w}, \mathcal{M}_{\text{NN}})\, p(\{\boldsymbol{X}_i\}_{i=1}^N \mid \boldsymbol{w}, \mathcal{M}_{\text{NN}})\, p(\boldsymbol{w} \mid \mathcal{M}_{\text{NN}})}{p(\{\boldsymbol{Y}_i\}_{i=1}^N \mid \{\boldsymbol{X}_i\}_{i=1}^N, \mathcal{M}_{\text{NN}})\, p(\{\boldsymbol{X}_i\}_{i=1}^N \mid \mathcal{M}_{\text{NN}})} \tag{4.7c}$$

$$= \frac{p(\{\boldsymbol{Y}_i\}_{i=1}^N \mid \{\boldsymbol{X}_i\}_{i=1}^N, \boldsymbol{w}, \mathcal{M}_{\text{NN}})\, p(\boldsymbol{w} \mid \mathcal{M}_{\text{NN}})}{p(\{\boldsymbol{Y}_i\}_{i=1}^N \mid \{\boldsymbol{X}_i\}_{i=1}^N, \mathcal{M}_{\text{NN}})}, \tag{4.7d}$$

where the cancellation leading to (4.7d) is valid only upon making the modelling assumption that $\boldsymbol{w}$ is independent of $\boldsymbol{X}$. Assuming data are conditionally independent, in the sense that $p(\boldsymbol{Y}_i \mid \boldsymbol{X}_i, \boldsymbol{X}_j, \boldsymbol{Y}_j, \boldsymbol{w}, \mathcal{M}_{\text{NN}}) = p(\boldsymbol{Y}_i \mid \boldsymbol{X}_i, \boldsymbol{w}, \mathcal{M}_{\text{NN}})$ for $j \neq i$, we can then write

$$p(\boldsymbol{w} \mid \{\boldsymbol{X}_i, \boldsymbol{Y}_i\}_{i=1}^N, \mathcal{M}_{\text{NN}}) = \frac{\prod_{i=1}^N p(\boldsymbol{Y}_i \mid \boldsymbol{X}_i, \boldsymbol{w}, \mathcal{M}_{\text{NN}})\, p(\boldsymbol{w} \mid \mathcal{M}_{\text{NN}})}{\prod_{i=1}^N p(\boldsymbol{Y}_i \mid \boldsymbol{X}_i, \mathcal{M}_{\text{NN}})}. \tag{4.8}$$

Comparing (4.8) with (4.6) we see that the likelihood is replaced with the conditional likelihood and the evidence similarly replaced by a conditional form. Although the conditional likelihood and prior terms can be evaluated easily, the evidence cannot, since it is an intractable integral over the (usually high-dimensional) space of network parameters. This is a standard challenge in Bayesian inference with complex models, both for conditional and marginal models, and it is one of the main obstacles to providing credible uncertainty quantification for neural network-based models. In order to make progress approximate inference algorithms are needed. In what follows we briefly review several techniques used to study the posterior distribution of neural networks: namely maximum a posteriori estimation, Laplace's method, Markov chain Monte Carlo and variational Bayesian inference.

4.3.1 Maximum a posteriori estimation and Laplace's method

As discussed in Section 2.3, a commonly used point estimate for parameters in a Bayesian setting is the posterior mode, or maximum a posteriori (MAP) estimate. This approach has been used to infer parameters of neural networks (MacKay 1995), in which case the MAP estimate is

$$\boldsymbol{w}^* = \arg\max_{\boldsymbol{w}} p(\boldsymbol{w} \mid \{\boldsymbol{X}_i, \boldsymbol{Y}_i\}_{i=1}^N, \mathcal{M}_{\text{NN}}) \tag{4.9a}$$

$$= \arg\max_{\boldsymbol{w}} \prod_{i=1}^N p(\boldsymbol{Y}_i \mid \boldsymbol{X}_i, \boldsymbol{w}, \mathcal{M}_{\text{NN}})\, p(\boldsymbol{w} \mid \mathcal{M}_{\text{NN}}). \tag{4.9b}$$

An advantage of adopting MAP estimates is that this approach avoids entirely the need to compute/estimate the evidence, which, since it is independent of $\boldsymbol{w}$, does not affect the optimisation. MAP estimation takes into account prior information but comes with little extra computational cost compared to maximum likelihood estimation, assuming the prior can be easily computed. However, MAP estimates alone do not satisfactorily characterise the posterior, since they, as point estimates, do not represent posterior uncertainty.

Laplace's method (also described in Section 2.3) offers a simple means of obtaining uncertainty quantification by making a Gaussian approximation for the posterior centred at its mode, enabling a simple estimate of the posterior covariance. One may worry that such a simple approximation will yield an inaccurate summary of the posterior distribution given that in neural networks with many parameters the posterior may be very far from Gaussian, having many local maxima. Indeed commonly used gradient-based optimisation schemes will find one such maximum and it is not clear if the neighbourhood of this maximum is representative of the posterior uncertainty. Moreover, the naive calculation of the hessian of the log of the unnormalised posterior required to apply Laplace's method (recall (2.12)) is expensive in the setting of neural networks. It is for these reasons that Laplace's method is largely unused in this context, in favour of more advanced methods. However, some have argued that Laplace's method can be applied in such models and that it is competitive with other methods (Daxberger et al. 2021).

4.3.2 Markov chain Monte Carlo

Beyond MAP estimates and Laplace's method a standard approach to posterior inference is sampling. From samples of the posterior one can produce estimates of the posterior mean and variance. Sampling can also highlight deviations from Gaussianity, such as kurtosis or multimodality, which cannot be identified with Laplace's method. Sampling is most commonly performed using Markov chain Monte Carlo (MCMC) methods (Robert & Casella. 2004, Gelman et al. 2013). Here we provide only the briefest summary of MCMC methods. The latter constitute a diverse range of sampling algorithms that are used frequently for tasks such as parameter estimation and rare event sampling. They serve as an indispensable tool in many areas of statistical research and applications.

The defining feature of MCMC algorithms is that they allow to generate samples of a random variable whose pdf is known only up to an unknown constant by constructing and simulating a Markov chain whose invariant measure is the desired probability measure. In the case of posterior inference for BNNs, we aim to sample realisations of $\boldsymbol{w}$ distributed according to its posterior distribution, with density $p(\boldsymbol{w} \mid \mathcal{D}, \mathcal{M}_{\mathrm{NN}}) \propto p(\mathcal{D} \mid \boldsymbol{w}, \mathcal{M}_{\mathrm{NN}})\, p(\boldsymbol{w} \mid \mathcal{M}_{\mathrm{NN}})$, without knowing the relevant normalising constant, the model evidence $p(\mathcal{D} \mid \mathcal{M}_{\mathrm{NN}})$.

Many MCMC algorithms can be considered variants or specific cases of the well-known Metropolis–Hastings algorithm (Hastings 1970, Robert 2015). These include

the random walk Metropolis algorithm (Metropolis et al. 1953), the Gibbs sampler (Geman & Geman 1984) and the Metropolis-adjusted Langevin algorithm (MALA) (Roberts & Tweedie 1996). In the general case the Metropolis–Hastings algorithm constructs a Markov chain $\boldsymbol{w}_n$ by defining a Markov transition density which combines a so-called proposal density $q(\tilde{\boldsymbol{w}}_{n+1} \mid \boldsymbol{w}_n)$ with an acceptance probability $\alpha(\tilde{\boldsymbol{w}}_{n+1} \mid \boldsymbol{w}_n)$. In particular,

$$\begin{aligned} p(\boldsymbol{w}_{n+1} \mid \boldsymbol{w}_n) = \alpha(\boldsymbol{w}_{n+1} \mid \boldsymbol{w}_n)\, q(\boldsymbol{w}_{n+1} \mid \boldsymbol{w}_n) \\ + \delta_{\boldsymbol{w}_n}(\boldsymbol{w}_{n+1}) \int (1 - \alpha(\boldsymbol{w}_{n+1} \mid \boldsymbol{w}_n))\, q(\boldsymbol{w}_{n+1} \mid \boldsymbol{w}_n)\, \mathrm{d}\boldsymbol{w}_{n+1}, \end{aligned} \tag{4.10}$$

where $\delta_{\boldsymbol{w}_n}$ denotes the Dirac measure in $\boldsymbol{w}_n$ and

$$\alpha(\tilde{\boldsymbol{w}}_{n+1} \mid \boldsymbol{w}_n) = \min\left\{1, \frac{p(\tilde{\boldsymbol{w}}_{n+1} \mid \mathcal{D}, \mathcal{M}_{\mathrm{NN}})\, q(\boldsymbol{w}_n \mid \tilde{\boldsymbol{w}}_{n+1})}{p(\boldsymbol{w}_n \mid \mathcal{D}, \mathcal{M}_{\mathrm{NN}})\, q(\tilde{\boldsymbol{w}}_{n+1} \mid \boldsymbol{w}_n)}\right\}. \tag{4.11}$$

In practice the Markov chain is simulated by first sampling a proposal $\tilde{\boldsymbol{w}}_{n+1}$ for the next state, then accepting that proposal with probability $\alpha(\tilde{\boldsymbol{w}}_{n+1} \mid \boldsymbol{w}_n)$ or rejecting it and taking $\boldsymbol{w}_{n+1} = \boldsymbol{w}_n$ with probability $1 - \alpha(\tilde{\boldsymbol{w}}_{n+1} \mid \boldsymbol{w}_n)$. The form of the acceptance probability ensures that the chain converges to the desired invariant distribution — in this case the posterior. The states $\{\boldsymbol{w}_n\}$ simulated can be taken as samples from the posterior, however, since Markov chains generally exhibit autocorrelation, these samples are correlated. In particular samples are not independent of the initial state $\boldsymbol{w}_0$. Typically only samples $\boldsymbol{w}_n$ for n greater than some chosen value are used, since the distribution of these samples will typically be closer to the invariant distribution of the Markov chain (Brooks et al. 2011). Earlier samples are discarded and referred to as burn-in samples. It is also common to subsample the trajectory, in order to discard highly correlated consecutive samples.

An MCMC method is successful if the samples obtained from it faithfully represent the posterior distribution. Typically this means we want simulated trajectories to explore well regions of parameter space which have relatively high posterior probability, although in certain cases we may care also about the tails of the posterior distribution, such as in the study of rare event statistics. The performance of the Metropolis–Hastings algorithm is highly sensitive to the choice of proposal density q. The random walk Metropolis, Gibbs sampler and MALA methods each correspond to different prescriptions for q. For example, the random walk Metropolis algorithm proposes states $\tilde{\boldsymbol{w}}_{n+1} = \boldsymbol{w}_n + \boldsymbol{\varepsilon}_n$ where $\boldsymbol{\varepsilon}_n \sim \mathcal{N}(\boldsymbol{0}, C_p)$ independently for each n and C_p is a covariance matrix to be chosen. In contrast MALA generates proposals by solving a discretised form of a Langevin-type equation whose invariant measure is (by construction) the posterior measure. Using the simple Euler–Maruyama discretisation (Pavliotis 2014) this leads to

$$\tilde{\boldsymbol{w}}_{n+1} = \boldsymbol{w}_n - h\nabla_{\boldsymbol{w}} \log p(\boldsymbol{w}_n \mid \mathcal{D}) + \sqrt{2h}\, \boldsymbol{\varepsilon}_n, \tag{4.12}$$

where h is a fixed stepsize and the gradient term can be evaluated without knowledge of the evidence since $\nabla_{\boldsymbol{w}} \log p(\boldsymbol{w}_n \mid \mathcal{D}) = \nabla_{\boldsymbol{w}} [\log p(\mathcal{D} \mid \boldsymbol{w}_n) + \log p(\boldsymbol{w}_n)]$. Note that although the continuous Langevin equation has the desired invariant measure, the error caused by the discretisation which leads to (4.12) introduces a bias, which is corrected by applying the Metropolis–Hastings accept/reject step. The corresponding algorithm where this correction is not applied is known as the Unadjusted Langevin algorithm (ULA) and is sometimes used despite the bias in order to reduce computational cost (Roberts & Tweedie 1996, Durmus & Moulines 2019). Yet another flavour of MCMC algorithm is Hamiltonian Monte Carlo (HMC) (see e.g. Brooks et al. 2011, Chapter 5). In HMC proposals are generated by solving discretely a Hamiltonian system whose potential energy is equal to the negative log posterior. The standard choice of discretisation is the Leapfrog method. A proposal is drawn by solving the system for L timesteps with stepsize ε_h. L and ε_h are parameters which must be tuned to achieve effective sampling. A popular variant of HMC known as the No-U-Turn Sampler (NUTS) (Hoffman et al. 2014) aims to automate tuning of L during the burn-in phase guided by a heuristic of ceasing iteration once trajectories are expected to retrace themselves, maximising the spread the proposals. This is paired with schemes for tuning ε_h based on the dual averaging algorithm of Nesterov (Nesterov 2009). The result is a fully-automatic sampling algorithm which is implemented in several widely used statistical software packages.

The utility of the MCMC methods mentioned above is however limited by the deterioration of their performance as the dimension of the space to be sampled increases. An important failure of, for example, the random walk Metropolis algorithm, is that with a fixed covariance of proposals, C_p, the average acceptance probability tends to decrease as the dimension of $\boldsymbol{w}$ increases, leading to poor exploration of the parameter space. In other words, to maintain a desirable average acceptance probability with standard MCMC methods as the problem dimension increases, it is usually necessary to scale the average proposal stepsize. This is a problem for BNNs, since neural networks regularly have millions, or even billions, of parameters. However, MCMC methods which do not suffer from acceptance probability degeneracy have been proposed as simple variations on existing algorithms. In particular, Cotter et al. (2013) proposed the preconditioned Crank–Nicolson (pCN) and the preconditioned Crank–Nicolson Langevin (pCNL) algorithms as a solution. Hairer et al. (2014) showed that, in the case of sampling posteriors with densities absolutely continuous with respect to a Gaussian reference density (which may or may not be the prior), the pCN algorithm possesses a dimension-independent spectral gap property, indicating that the convergence rate of this method is independent of the dimension of the parameter space to be sampled. Despite this promising progress in developing efficient MCMC methods, their implementation in large neural network models is still prohibitively expensive and there remains a demand for cheaper approximate methods. It is for this reason that variational Bayesian inference has become an increasingly popular method for approximate inference and arguably the go-to method for building Bayesian neural networks.

4.3.3 Variational Bayesian inference

Variational Bayesian inference aims to approximate the posterior distribution by assuming it belongs to (or at least can be approximated by a member of) a certain parametric family of distributions, say with density $q(\boldsymbol{w};\, \boldsymbol{\xi})$, and minimising with respect to $\boldsymbol{\xi}$ a loss function which quantifies the difference between q and the true posterior. We call q the surrogate posterior and $\boldsymbol{\xi}$ the variational parameters. The loss function that is commonly used is based on a quantity known as the evidence lower bound (ELBO),

$$\mathrm{ELBO}(\boldsymbol{\xi}) = \mathbb{E}_Q \left[\log p(\boldsymbol{w},\, \mathcal{D} \mid \mathcal{M}_{\mathrm{NN}}) - \log q(\boldsymbol{w};\, \boldsymbol{\xi})\right], \tag{4.13}$$

where $\mathbb{E}_Q$ denotes expectation with respect to the surrogate posterior. The ELBO is motivated by noting

$$\begin{aligned}
&D_{\mathrm{KL}}(q(\boldsymbol{w};\, \boldsymbol{\xi}) \parallel p(\boldsymbol{w} \mid \mathcal{D},\, \mathcal{M}_{\mathrm{NN}})) && \text{(4.14a)}\\
&\quad= \int q(\boldsymbol{w};\, \boldsymbol{\xi})\, \log \frac{q(\boldsymbol{w};\, \boldsymbol{\xi})}{p\left(\boldsymbol{w} \mid \mathcal{D},\, \mathcal{M}_{\mathrm{NN}}\right)}\, \mathrm{d}\boldsymbol{w} && \text{(4.14b)}\\
&\quad= \int q(\boldsymbol{w};\, \boldsymbol{\xi})\, \left[\log \frac{q(\boldsymbol{w};\, \boldsymbol{\xi})}{p\left(\boldsymbol{w},\, \mathcal{D} \mid \mathcal{M}_{\mathrm{NN}}\right)} + \log p\left(\mathcal{D} \mid \mathcal{M}_{\mathrm{NN}}\right)\right] \mathrm{d}\boldsymbol{w} && \text{(4.14c)}\\
&\quad= \int q(\boldsymbol{w};\, \boldsymbol{\xi})\, \left[\log q(\boldsymbol{w};\, \boldsymbol{\xi}) - \log p\left(\boldsymbol{w},\, \mathcal{D} \mid \mathcal{M}_{\mathrm{NN}}\right)\right] \mathrm{d}\boldsymbol{w} + \log p(\mathcal{D} \mid \mathcal{M}_{\mathrm{NN}}) && \text{(4.14d)}\\
&\quad= \mathbb{E}_Q \left[\log q(\boldsymbol{w};\, \boldsymbol{\xi}) - \log p(\boldsymbol{w},\, \mathcal{D} \mid \mathcal{M}_{\mathrm{NN}})\right] + \log p(\mathcal{D} \mid \mathcal{M}_{\mathrm{NN}}). && \text{(4.14e)}
\end{aligned}$$

Thus,

$$\arg\min_{\boldsymbol{\xi}}\ D_{\mathrm{KL}}(q(\boldsymbol{w};\, \boldsymbol{\xi}) \parallel p\left(\boldsymbol{w} \mid \mathcal{D},\, \mathcal{M}_{\mathrm{NN}}\right)) = \arg\max_{\boldsymbol{\xi}}\ \mathrm{ELBO}(\boldsymbol{\xi}). \tag{4.15}$$

However, we highlight that the order of arguments of the KL divergence in (4.14a) is opposite to what we would like[1]. The name evidence lower bound comes from the fact that, since $p(\mathcal{D} \mid \mathcal{M}_{\mathrm{NN}})$ is the model evidence, and the KL divergence is non-negative, the ELBO provides a lower bound of the evidence. Notice that the ELBO can be rewritten

$$\begin{aligned}
\mathrm{ELBO}(\boldsymbol{\xi}) &= \mathbb{E}_Q \left[\log p(\boldsymbol{w},\, \mathcal{D} \mid \mathcal{M}_{\mathrm{NN}}) - \log q(\boldsymbol{w};\, \boldsymbol{\xi})\right] && \text{(4.16a)}\\
&= \mathbb{E}_Q \left[\log p(\boldsymbol{w} \mid \mathcal{M}_{\mathrm{NN}})\right] + \mathbb{E}_Q \left[\log p(\mathcal{D} \mid \boldsymbol{w},\, \mathcal{M}_{\mathrm{NN}})\right] - \mathbb{E}_Q \left[\log q(\boldsymbol{w};\, \boldsymbol{\xi})\right] && \text{(4.16b)}\\
&= \mathbb{E}_Q \left[\log p(\mathcal{D} \mid \boldsymbol{w},\, \mathcal{M}_{\mathrm{NN}})\right] - D_{\mathrm{KL}}(q(\boldsymbol{w}\,;\, \boldsymbol{\xi}) \parallel p(\boldsymbol{w} \mid \mathcal{M}_{\mathrm{NN}})), && \text{(4.16c)}
\end{aligned}$$

[1]We would anticipate minimising $D_{\mathrm{KL}}(p(\boldsymbol{w} \mid \mathcal{D},\, \mathcal{M}_{\mathrm{NN}}) \parallel q(\boldsymbol{w};\, \boldsymbol{\xi}))$, but this is typically intractable since it involves an expectation with respect to the unknown posterior distribution. Heuristically, one hopes that (4.14a) is a good approximation to the KL divergence with opposite order of arguments. In any case, this is the objective that is used in practice.

leading to the negative ELBO loss function used in practice,

$$-\text{ELBO}(\boldsymbol{\xi}) = -\mathbb{E}_{Q}\left[\log p(\mathcal{D} \mid \boldsymbol{w},\, \mathcal{M}_{\text{NN}})\right] + D_{\text{KL}}(q(\boldsymbol{w};\, \boldsymbol{\xi}) \parallel p(\boldsymbol{w} \mid \mathcal{M}_{\text{NN}})). \tag{4.17}$$

In fact, this is the form used for unconditional (or marginal, or generative) models. In the conditional modelling case we can note that

$$\begin{aligned} p(\mathcal{D} \mid \boldsymbol{w},\, \mathcal{M}_{\text{NN}}) &= p(\{\boldsymbol{X}_i,\, \boldsymbol{Y}_i\}_{i=1}^{N} \mid \boldsymbol{w},\, \mathcal{M}_{\text{NN}}) && \text{(4.18a)} \\ &= p(\{\boldsymbol{Y}_i\}_{i=1}^{N} \mid \{\boldsymbol{X}_i\}_{i=1}^{N},\, \boldsymbol{w},\, \mathcal{M}_{\text{NN}})\; p(\{\boldsymbol{X}_i\}_{i=1}^{N} \mid \boldsymbol{w},\, \mathcal{M}_{\text{NN}}), && \text{(4.18b)} \end{aligned}$$

where the latter term is, by assumption, independent of $\boldsymbol{w}$ and hence independent of $\boldsymbol{\xi}$. Thus, we may equivalently minimise the following loss function

$$\mathcal{L}_{\text{ELBO}}(\boldsymbol{\xi}) = -\mathbb{E}_{Q}\left[\sum_{i=1}^{N} \log \rho(\boldsymbol{Y}_i\,;\, \boldsymbol{\theta}(\boldsymbol{X}_i\,;\, \boldsymbol{w}))\right] + D_{\text{KL}}(q(\boldsymbol{w}\,;\, \boldsymbol{\xi}) \parallel p(\boldsymbol{w} \mid \mathcal{M}_{\text{NN}})). \tag{4.19}$$

The first term in the loss function (4.19) is the posterior expected conditional log likelihood, and quantifies how well the surrogate posterior fits data. The second term is the KL divergence between the prior and the surrogate posterior and enforces a degree of consistency with prior knowledge. The tradeoff between these terms mirrors the balance between likelihood and prior seen in Bayes' theorem (4.6). Both terms in (4.19) may be intractable for a given model, prior and family of surrogate posteriors. The development of estimators for the gradient of these terms has been the subject of recent research (Kingma & Welling 2014, Wen et al. 2018, Roeder et al. 2017), given that gradient estimates are required to perform stochastic gradient-based optimisation for the solution of the variational problem (4.15).

Often $q(\boldsymbol{w}\,;\, \boldsymbol{\xi})$ and $p(\boldsymbol{w} \mid \mathcal{M}_{\text{NN}})$ are specified such that the KL divergence in (4.19), and its derivative with respect to $\boldsymbol{\xi}$ can be computed analytically, e.g. by asserting that these densities are both Gaussian. However, it remains to estimate the gradient of the first term which is of the form $\nabla_{\boldsymbol{\xi}}\mathbb{E}_q[f(\boldsymbol{w})]$. Assuming the required regularity an unbiased estimator of this gradient can be constructed by rewriting the gradient as an expectation as follows

$$\begin{aligned} \nabla_{\boldsymbol{\xi}}\mathbb{E}_q[f(\boldsymbol{w})] &= \nabla_{\boldsymbol{\xi}} \int f(\boldsymbol{w})\, q(\boldsymbol{w}\,;\, \boldsymbol{\xi})\, \mathrm{d}\boldsymbol{w} && \text{(4.20)} \\ &= \int f(\boldsymbol{w})\, \nabla_{\boldsymbol{\xi}} q(\boldsymbol{w}\,;\, \boldsymbol{\xi})\, \mathrm{d}\boldsymbol{w} && \text{(4.21)} \\ &= \int f(\boldsymbol{w})\, q(\boldsymbol{w}\,;\, \boldsymbol{\xi})\, \nabla_{\boldsymbol{\xi}} \ln q(\boldsymbol{w}\,;\, \boldsymbol{\xi})\, \mathrm{d}\boldsymbol{w} && \text{(4.22)} \\ &= \mathbb{E}_q\left[f(\boldsymbol{w})\, \nabla_{\boldsymbol{\xi}} \ln q(\boldsymbol{w}\,;\, \boldsymbol{\xi})\right] && \text{(4.23)} \end{aligned}$$

such that a Monte Carlo estimate can be used based on samples of the surrogate

posterior. In particular,

$$\nabla_{\boldsymbol{\xi}} \mathbb{E}_q[f(\boldsymbol{w})] \approx \frac{1}{N_s} \sum_{s=1}^{N_s} f\left(\boldsymbol{w}^{(s)}\right) \nabla_{\boldsymbol{\xi}} \ln q\left(\boldsymbol{w}^{(s)}\,;\,\boldsymbol{\xi}\right), \tag{4.24}$$

where $\boldsymbol{w}^{(s)}$ are independently sampled with density $q(\boldsymbol{w}\,;\,\boldsymbol{\xi})$. However, this estimator has been found to have high variance (Paisley et al. 2012).

An alternative estimate, given by the so-called reparameterisation trick (Kingma & Welling 2014, Rezende et al. 2014), is more commonly used. For many choices of the surrogate posterior, the random variable $\boldsymbol{w}^{(i)} \sim q(\boldsymbol{w}^{(i)}\,;\,\boldsymbol{\xi})$ can be written as a deterministic differentiable function of $\boldsymbol{\xi}$ and an auxiliary random variable $\boldsymbol{\eta}_{\text{aux}}$, say $\boldsymbol{w}^{(i)} = \boldsymbol{g}(\boldsymbol{\xi}, \boldsymbol{\eta}^{(i)}_{\text{aux}})$. For example, in the case of a surrogate posterior which is Gaussian with diagonal covariance matrix we can write $\boldsymbol{w}^{(i)} = \boldsymbol{\mu}_{\boldsymbol{\xi}} + \boldsymbol{\sigma}_{\boldsymbol{\xi}} \cdot \boldsymbol{\eta}^{(i)}_{\text{aux}}$ with $\boldsymbol{\eta}^{(i)}_{\text{aux}}$ distributed as a standard multivariate Gaussian and $\boldsymbol{\xi} = \{\boldsymbol{\mu}_{\boldsymbol{\xi}}, \boldsymbol{\sigma}_{\boldsymbol{\xi}}\}$ the relevant variational parameters. The utility of this rewriting is that gradients with respect to $\boldsymbol{\xi}$ of Monte Carlo estimates of expectations with respect to $q(\boldsymbol{w}^{(i)}\,;\,\boldsymbol{\xi})$ can be defined and computed. In particular, the first term in (4.19) can be estimated by Monte Carlo and we can compute the gradient of this estimate with respect to $\boldsymbol{\xi}$. This provides an estimate of the gradient of (4.19)

$$\begin{aligned}\nabla_{\boldsymbol{\xi}} \mathcal{L}_{\text{ELBO}}(\boldsymbol{\xi}) \approx \nabla_{\boldsymbol{\xi}} \sum_{s=1}^{N_s} \left(\sum_{i=1}^{N} \log \rho(\boldsymbol{Y}_i\,;\,\boldsymbol{\theta}(\boldsymbol{X}_i\,;\,\boldsymbol{w}^{(s)})) \right) \\ + \nabla_{\boldsymbol{\xi}} D_{\text{KL}}(q(\boldsymbol{w}\,;\,\boldsymbol{\xi}) \parallel p(\boldsymbol{w} \mid \mathcal{M}_{\text{NN}})).\end{aligned} \tag{4.25}$$

This estimated gradient can then be used in a gradient descent-type optimisation procedure. Usually this estimation is combined with mini-batching, where at each iteration of the optimisation scheme the loss over the full dataset is estimated by the loss over a random subset of the data; in our case this is

$$\begin{aligned}\nabla_{\boldsymbol{\xi}} \mathcal{L}_{\text{ELBO}}(\boldsymbol{\xi}) \approx \nabla_{\boldsymbol{\xi}} \sum_{s=1}^{N_s} \left(\frac{N}{B} \sum_{i=1}^{B} \log \rho(\boldsymbol{Y}_i\,;\,\boldsymbol{\theta}(\boldsymbol{X}_i\,;\,\boldsymbol{w}^{(s)})) \right) \\ + \nabla_{\boldsymbol{\xi}} D_{\text{KL}}(q(\boldsymbol{w}\,;\,\boldsymbol{\xi}) \parallel p(\boldsymbol{w} \mid \mathcal{M}_{\text{NN}})),\end{aligned} \tag{4.26}$$

where B is the size of the minibatch and each $\{\boldsymbol{X}_i, \boldsymbol{Y}_i\}$ is a data pair drawn uniformly at random from the dataset without replacement. Descending with this gradient then corresponds to a form of stochastic gradient descent (SGD) and is known as stochastic gradient variational Bayes (SGVB) (Kingma & Welling 2014). Variants of SGD such as Adam (Kingma & Ba 2015) can also be applied. In this case, the gradient estimate features two controllable parameters, N_s and B, which can be chosen to achieve a desired variance at an acceptable cost. In experiments Kingma & Welling (2014) found that the number of samples in the Monte Carlo estimate can be set to $N_s = 1$ provided that the batch size B is large enough. Several variants of the gradient estimator in SGVB

have been proposed (Kingma et al. 2015, Wen et al. 2018, Roeder et al. 2017) which exploit various canonical variance reduction techniques for Monte Carlo methods, including control variates and antithetic sampling (Kroese et al. 2011).

SGVB and its variants represent the state of the art in variational Bayesian inference for neural networks. However, theoretical guarantees for such methods are sparse and, as with much of machine learning, practice remains guided to some extent by heuristics and by experience: methods are judged by their empirical performance. Obvious questions include:

(i) How flexible does the surrogate posterior family need to be to contain a 'good' approximation of the true posterior?

(ii) Assuming the surrogate posterior family contains 'good' approximations, under what conditions will SGVB converge to one, in what sense, and at what rate?

A recent paper by Sharma et al. (2023) suggests a route to an answer to (i). The authors combine universal approximation theorems for neural networks with a result from probability theory to argue (roughly) that, under certain assumptions, neural networks with only m independent random parameters can approximate an arbitrary conditional distribution, say $\tilde{\boldsymbol{Y}} \mid \tilde{\boldsymbol{X}}$, where m is the dimension of $\tilde{\boldsymbol{Y}}$. This is an appealing notion because it suggests particularly simple prescriptions for forms of surrogate posterior satisfying (i). For instance, one could have independent fixed Gaussian distributions on m network parameters, and a Dirac delta function (to be optimised) on all other parameters — in other words most network parameters could take a deterministic value. Adopting such a surrogate posterior could reduce significantly the cost of sampling (and hence also training) Bayesian neural networks in comparison with more flexible surrogate families, since each realisation of the network weights would require the generation of only m Gaussian samples, as opposed to the number of parameters in the network, as in the diagonal Gaussian case. They appeal to a result known variously as either noise outsourcing (Austin 2015, lemma 3.1) or, simply, transfer (Kallenberg 1997, theorem 5.10).

Theorem 1 (Transfer, or noise outsourcing)**.** *Suppose S and T are random variables taking values in standard Borel spaces $\mathcal{S}$ and $\mathcal{T}$. Then there exist a random variable $\eta \sim U(0,\, 1)$ and a measurable function $g : \mathcal{S} \times [0,\, 1] \to \mathcal{T}$ such that $(S,\, T) = (S,\, g(T,\, \eta))$ almost surely.*

Further, since η is independent of S, we have that $g(S,\, \eta) \overset{d}{=} T \mid S$. We call such a function g a conditional generator function of T given S. Intuitively, this says that the variability in $T \mid S$ can be represented by a function of S and a uniform random variable η which is independent of S and known as a randomisation variable. An equivalent result holds for η a standard Gaussian. We emphasise that, while η is a scalar random variable, the statement holds for very general $\mathcal{S}$ and $\mathcal{T}$; in particular, S and T can take values in subsets of high-dimensional Euclidean spaces, as is most relevant to point (i).

The concept of conditional generator functions allows to phrase the task of modelling conditional distributions as approximating a deterministic function (the function g in Theorem 1). Since formal results exist regarding the ability of neural networks of various architectures to approximate functions, it appears natural to combine a version of Theorem 1 with a universal approximation theorem for neural networks in order to claim that surrogate posteriors for neural networks can approximate true posteriors. However, basic universal approximation theorems for neural networks apply only to continuous functions, and the function f in Theorem 1 may not be continuous. Sharma et al. (2023) argue that there is more likely to exist a continuous generator function based on m independent Gaussian random variables; in particular, one can imagine constructing such a function via the inverse cumulative density functions corresponding to the conditional distributions $\{\tilde{\boldsymbol{Y}}_j \mid \tilde{\boldsymbol{X}}, \tilde{\boldsymbol{Y}}_1, \cdots, \tilde{\boldsymbol{Y}}_{j-1}\}_{j=1}^m$, provided these exist and are continuous — this is the strategy of inverse transform sampling (Robert & Casella. 2004), but here we employ a neural network to represent the unknown inverse cumulative density functions. Ultimately, in their work, Sharma et al. (2023) are forced to assume an appropriate continuous generator function exists. This leaves a major gap in their result.

To the best of our knowledge, theoretical results regarding question (ii) are essentially lacking, and progress by way of empirical investigations is hampered by the lack of a ground truth for comparison, since it is currently prohibitively expensive to perform thorough MCMC sampling of the posterior of large neural network models. Questions (i) and (ii) point to two ways in which variational Bayesian inference for neural networks can fail — without answers to both of these questions, the method comes without guarantees. Hence, there is a clear need for progress on these issues.

Aside from the challenge of constructing satisfactory approximate inference methods, there is also the issue of specifying priors for neural networks. Indeed this is the other major challenge for building truly Bayesian neural networks and is the subject of the following section.

4.4 Priors on neural networks

Bayesian inference requires the specification of a prior on model parameters, $p(\boldsymbol{w} \mid \mathcal{M}_{\text{NN}})$. In practice, it is difficult to assign a meaningful prior, since the values of network parameters are not easily interpreted in terms of domain knowledge. Commonly a fairly arbitrary prior is applied which is Gaussian with zero mean and identity covariance matrix, i.e.

$$\boldsymbol{w} \mid \mathcal{M}_{\text{NN}} \sim \mathcal{N}(\boldsymbol{0}, \mathbb{I}), \tag{4.27}$$

where $\mathbb{I}$ is the $\dim(\boldsymbol{w}) \times \dim(\boldsymbol{w})$ identity matrix. We refer to this as the default prior. A common justification for the default prior is that it encourages sparsity by regularising parameter value (Jospin et al. 2022). However, this prior is not well understood. Note that, since neural networks are functions of their inputs, a value of $\boldsymbol{w}$ corre-

sponds to a function $\boldsymbol{f}_{\mathrm{NN}}(\,\cdot\,;\,\boldsymbol{w})$, and a measure on $\boldsymbol{w}$ corresponds to a measure on the space of functions that the network can represent, $\left\{\boldsymbol{f}_{\mathrm{NN}}(\,\cdot\,;\,\boldsymbol{w}) : \boldsymbol{w} \in \mathbb{R}^{\dim(\boldsymbol{w})}\right\}$. This measure can be characterised by its moments. For instance, its mean and autocovariance functions are

$$\mathbb{E}\left[\boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{X})\right] = \int_W \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{X}\,;\,\boldsymbol{w})\, p(\boldsymbol{w} \mid \mathcal{M}_{\mathrm{NN}})\,\mathrm{d}\boldsymbol{w} \tag{4.28}$$

and

$$\begin{aligned}\operatorname{Cov}\left(\boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{X}),\, \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{X'})\right) = \int_W &\left(\boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{X}\,;\,\boldsymbol{w}) - \mathbb{E}[\boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{X})]\right) \\ &\left(\boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{X'}\,;\,\boldsymbol{w}) - \mathbb{E}[\boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{X'})]\right)\, p(\boldsymbol{w} \mid \mathcal{M}_{\mathrm{NN}})\,\mathrm{d}\boldsymbol{w}.\end{aligned} \tag{4.29}$$

We refer to such a function-space prior as the pushforward of the corresponding weight-space prior. The pushforward of the default prior (4.27) for (nonlinear) MLPs is intractable, but it can be seen, simply by sampling from the prior, that it has several undesirable characteristics. In Figures 4.1 and 4.2 we present samples from the pushforward of the default prior for MLPs with scalar input X and output f_{NN} and with two and six hidden layers, respectively, each with two different choices of nonlinear activation function $a(\,\cdot\,)$, namely $a(x) = \tanh x$, as used in Chapter 3, and $a(x) = \mathrm{ReLU}(x) \coloneqq \max(0, x)$, another very popular choice of activation function for neural networks. These figures highlight two features of the default prior: i) that it has implicit scales and is hence not invariant to affine transformation of the data, and ii) that the asymptotic behaviour of f_{NN} is degenerate — in particular, with $a(x) = \tanh x$ all samples tend to constant and with $a(x) = \mathrm{ReLU}(x)$ all samples tend to linear as $|X| \to \infty$. In both cases nontrivial fluctuations in f_{NN} are seen only in a narrow interval around zero. The effect of increasing network depth is to allow more complex fluctuations in a slightly enlarged interval. Note that, since $\boldsymbol{\theta}(\,\cdot\,) = \boldsymbol{h}(\boldsymbol{f}_{\mathrm{NN}}(\,\cdot\,;\,\boldsymbol{w}))$, the default prior on $\boldsymbol{w}$ implies a prior on $\boldsymbol{\theta}(\,\cdot\,)$ given by the pushforward under $\boldsymbol{h} \circ \boldsymbol{f}_{\mathrm{NN}}$. Hence, the implied prior on $\boldsymbol{\theta}(\,\cdot\,)$ depends also on $\boldsymbol{h}$, which further complicates its interpretation.

Figures 4.3 and 4.4 show the mean and variance of 1000 samples for each configuration. Here we see again that the choice of activation function has a significant effect on the pushforward of the default prior. In particular, although the mean function is approximately zero in both cases, the standard deviation of f_{NN} is near constant with $a(x) = \tanh x$ and grows linearly in $|X|$ with $a(x) = \mathrm{ReLU}(x)$. The rate of growth is seen to increase by more than two orders of magnitude with six hidden layers relative to the two hidden layers case.

A further issue with the default prior is that the implied prior autocovariance function is highly uncontrolled and sensitive to network architecture. In particular,

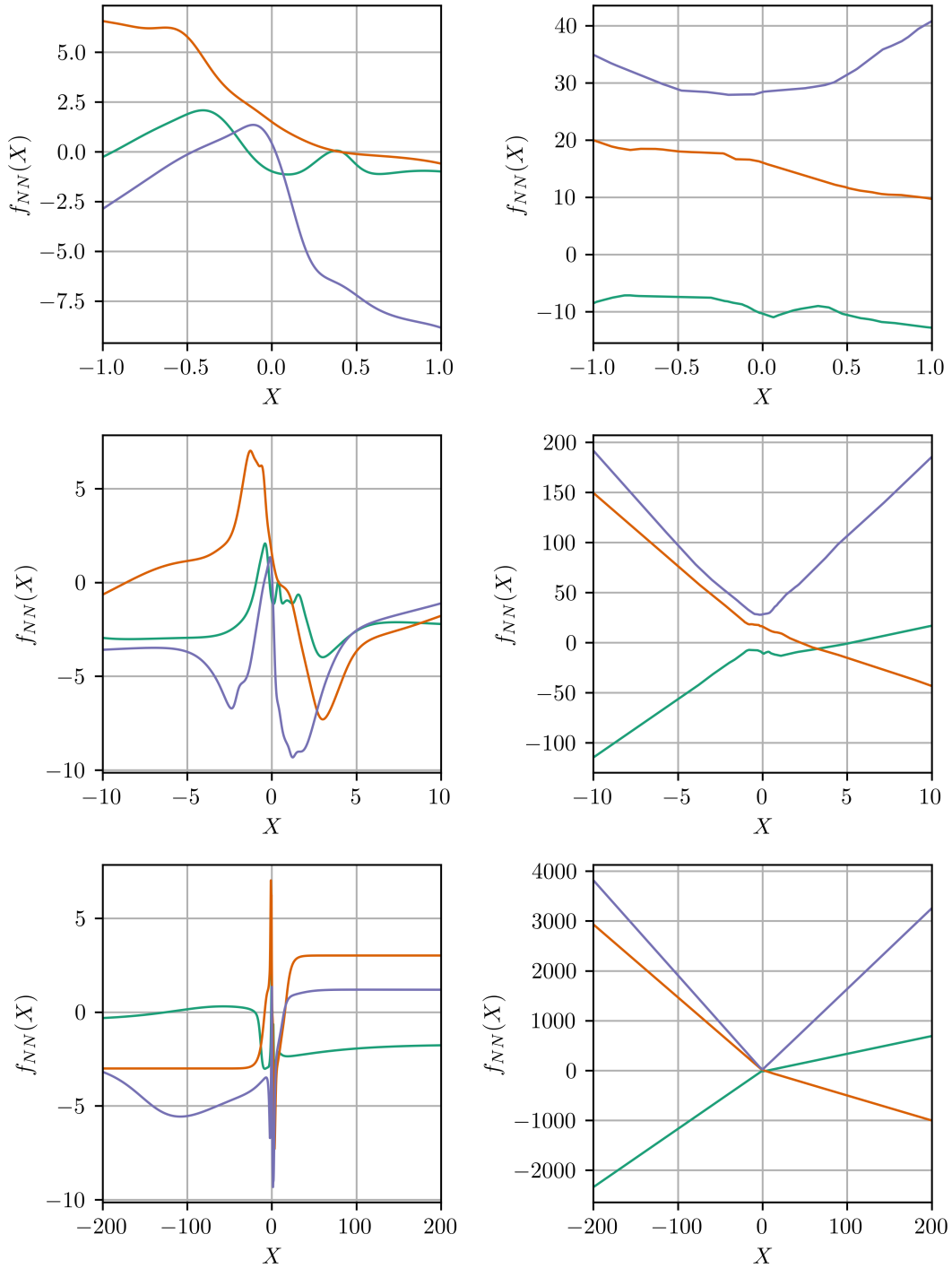


Figure 4.1: Realisations of $f_{\text{NN}}(\,\cdot\,)$ with two hidden layers, each having 32 neurons, with $a(x) = \tanh x$ (left), and with $a(x) = \text{ReLU}(x)$ (right), shown for increasingly wide ranges of input.

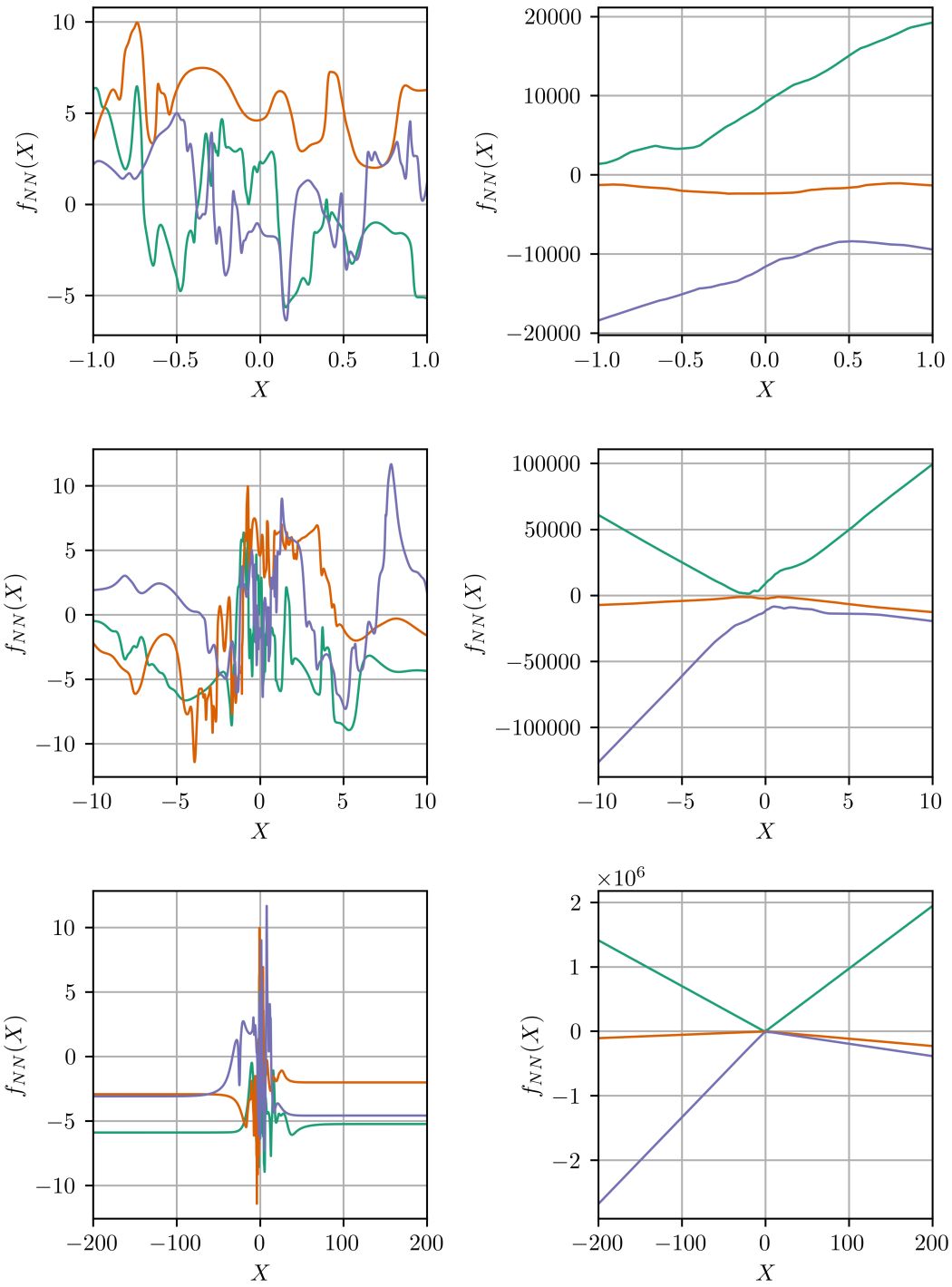


Figure 4.2: Same as Figure 4.1 but with six hidden layers.

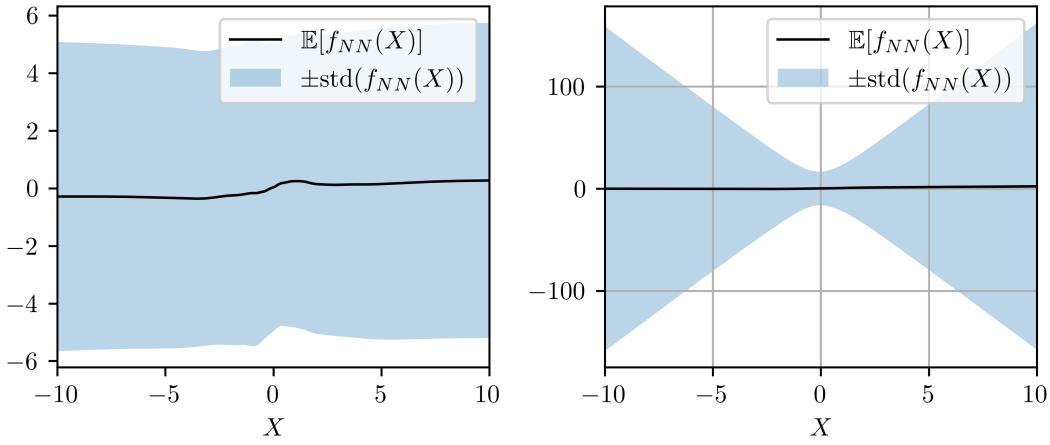


Figure 4.3: Sample mean and standard deviation of 1000 samples of $f_{\text{NN}}(\,\cdot\,)$ with two hidden layers, each having 32 neurons, with $a(x) = \tanh x$ (left), and with $a(x) = \text{ReLU}(x)$ (right).

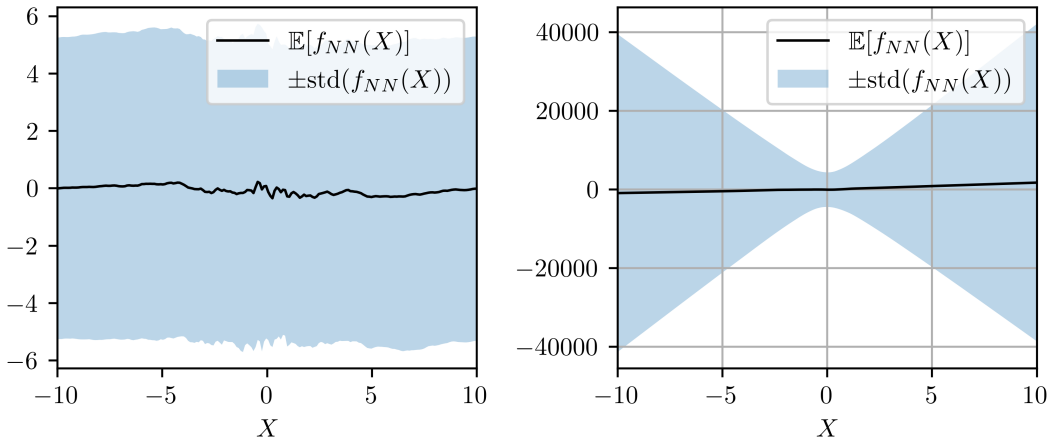


Figure 4.4: Same as Figure 4.3 but with six hidden layers.

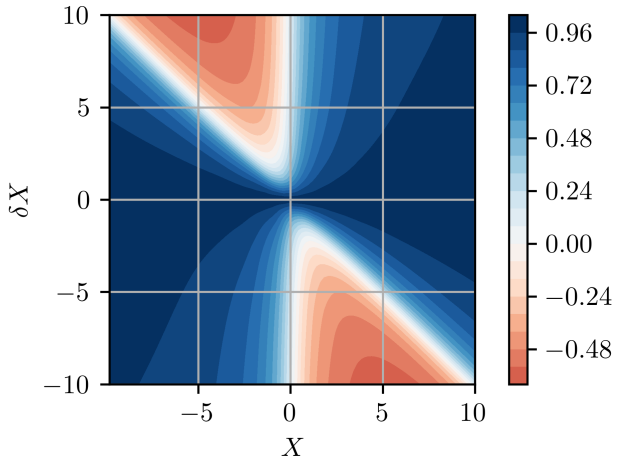


Figure 4.5: Estimated autocorrelation function of f_{NN}, $R_f(X, \delta X)$, with two hidden layers, each having 32 neurons, and $a(x) = \tanh x$.

f_{NN} is not stationary. We define the nonstationary autocorrelation function of f_{NN}

$$R_f(X, \delta X) := \operatorname{Corr}(f_{\mathrm{NN}}(X), f_{\mathrm{NN}}(X + \delta X)) \tag{4.30a}$$

$$= \frac{\operatorname{Cov}(f_{\mathrm{NN}}(X), f_{\mathrm{NN}}(X + \delta X))}{\operatorname{std}(f_{\mathrm{NN}}(X)) \operatorname{std}(f_{\mathrm{NN}}(X + \delta X))}. \tag{4.30b}$$

Figures 4.5, 4.6, 4.7 and 4.8 show estimates of R_f for the same configurations as above. These are computed in each case as simple empirical averages based on 2000 samples of f_{NN}. It is clear that the autocorrelation, aside from being nonstationary, is also asymmetric in δX. However, it appears that $R_f(X, \delta X) = R_f(-X, -\delta X)$ in each case.

Overall, the default prior is highly unsatisfactory for several reasons:

(i) its pushforward is severely affected by changes in network configuration, including depth and choice of activation function;

(ii) it is not stationary, and hence not invariant to simple transformations of the data; and

(iii) it has degenerate asymptotic behaviour which depends on the choice of activation function.

The default prior is, therefore, not suitable as a generic choice for modelling with Bayesian neural networks.

Recently, attempts have been made to construct priors for neural networks whose pushforwards approximate Gaussian process priors on $\boldsymbol{f}_{\mathrm{NN}}$ (Flam-Shepherd et al.

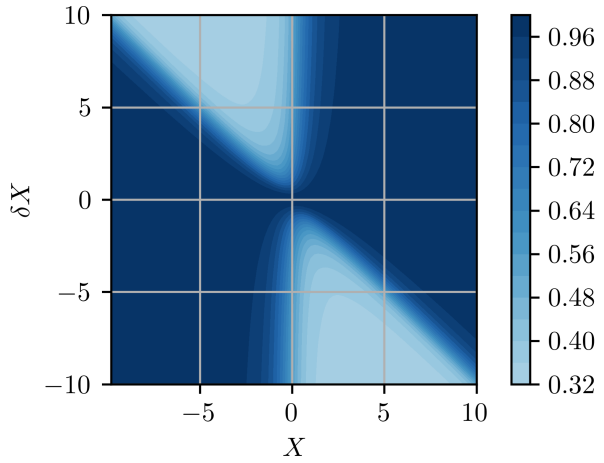


Figure 4.6: Same as Figure 4.5 but with $a(x) = \text{ReLU}(x)$.

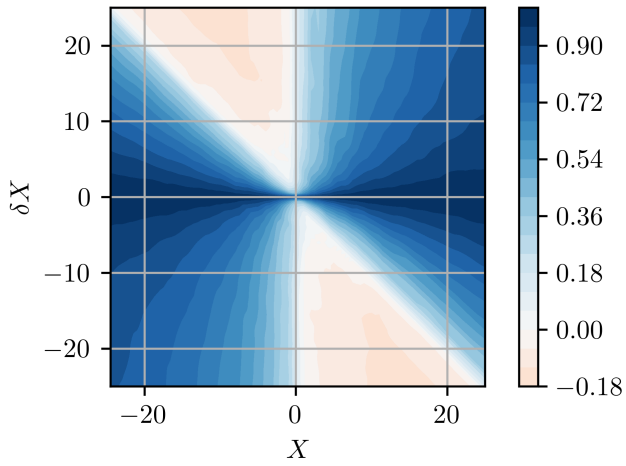


Figure 4.7: Same as Figure 4.5 but with six hidden layers.

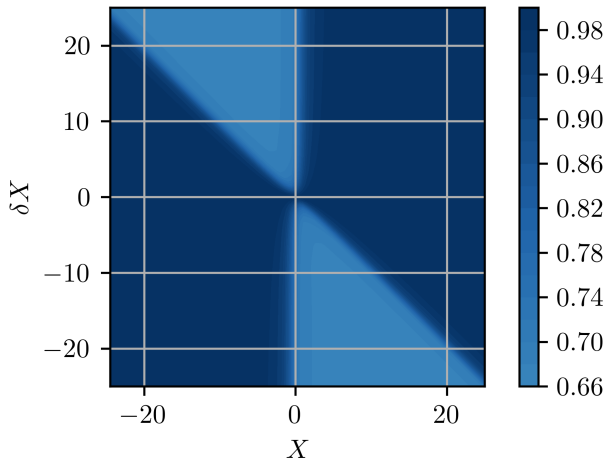


Figure 4.8: Same as Figure 4.5 but with $a(x) = \mathrm{ReLU}(x)$ and six hidden layers.

2017, Tran et al. 2022). Gaussian process priors, as used in Gaussian process regression (see e.g. Williams & Rasmussen 2006), are more interpretable than the pushforwards of standard priors on $\boldsymbol{w}$ and are better understood in general. In order to approximate a Gaussian process prior one can introduce a family of possible weight-space priors, say with density $\varrho(\boldsymbol{w}\,;\,\beta)$ and parameters β, and try to minimise a measure of the discrepancy between the pushforward of members of that family, which we denote $\varrho(\boldsymbol{f}_{\mathrm{NN}}\,;\,\boldsymbol{\beta})$, and a desired Gaussian process $p_{\mathrm{GP}}(\boldsymbol{f}_{\mathrm{NN}})$. This results in a variational optimisation problem which mirrors that of finding an optimal surrogate posterior in variational Bayesian inference. In particular, given a suitable divergence (possibly a distance) D between densities, we might try to find

$$\arg\min_{\boldsymbol{\beta}} \; D(\varrho(\boldsymbol{f}_{\mathrm{NN}}\,;\,\boldsymbol{\beta}) \parallel p_{\mathrm{GP}}(\boldsymbol{f}_{\mathrm{NN}})). \tag{4.31}$$

Flam-Shepherd et al. (2017) proposed to follow this approach with $D = D_{\mathrm{KL}}$. Note that, while this is formally a KL divergence between measures on function spaces, a finite dimensional view of $\boldsymbol{f}_{\mathrm{NN}}$ is taken in practice. Rather than treat the infinite dimensional KL divergence, Flam-Shepherd et al. (2017) restrict attention to a finite number of $\boldsymbol{X}$ values sampled randomly according to a chosen density $p(\boldsymbol{X})$. In par-

ticular, taking this finite-dimensional view we may write

$$D_{\mathrm{KL}}(\varrho(\boldsymbol{f}_{\mathrm{NN}}\,;\,\boldsymbol{\beta}) \parallel p_{\mathrm{GP}}(\boldsymbol{f}_{\mathrm{NN}})) = \int \varrho(\boldsymbol{f}_{\mathrm{NN}}\,;\,\boldsymbol{\beta}) \log\left[\frac{\varrho(\boldsymbol{f}_{\mathrm{NN}}\,;\,\boldsymbol{\beta})}{p_{\mathrm{GP}}(\boldsymbol{f}_{\mathrm{NN}})}\right] \mathrm{d}\boldsymbol{f}_{\mathrm{NN}} \tag{4.32a}$$

$$= \int \varrho(\boldsymbol{f}_{\mathrm{NN}}\,;\,\boldsymbol{\beta}) \log \varrho(\boldsymbol{f}_{\mathrm{NN}}\,;\,\boldsymbol{\beta})\, \mathrm{d}\boldsymbol{f}_{\mathrm{NN}} - \int \varrho(\boldsymbol{f}_{\mathrm{NN}}\,;\,\boldsymbol{\beta}) \log p_{\mathrm{GP}}(\boldsymbol{f}_{\mathrm{NN}})\, \mathrm{d}\boldsymbol{f}_{\mathrm{NN}}. \tag{4.32b}$$

The first term in (4.32b) is the (differential) entropy[2] of $\boldsymbol{f}_{\mathrm{NN}}$ with respect to the density $\varrho(\boldsymbol{f}_{\mathrm{NN}}\,;\,\boldsymbol{\beta})$. This term is intractable. Indeed, the density $\varrho(\boldsymbol{f}_{\mathrm{NN}}\,;\,\boldsymbol{\beta})$ is not known in closed form. Thus, the entropy term can only be estimated by sampling. Estimating entropy from samples is a notoriously difficult problem. Flam-Shepherd et al. (2017) compute an estimate of the entropy based on moment-matching, i.e. they compute an empirical mean and variance of $\boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{X})$ (pointwise at each chosen value of $\boldsymbol{X}$) from samples, and take the average of the entropies of the corresponding Gaussian distributions.

The second term in (4.32b), which is a cross-entropy, is more tractable, because the density of the Gaussian process $p_{\mathrm{GP}}(\boldsymbol{f}_{\mathrm{NN}})$ can be evaluated analytically (after restricting to a finite number of $\boldsymbol{X}$ values). The authors ultimately report that results were unsatisfactory. Instead they suggest to neglect the entropy term completely and instead implement an early-stopping scheme, on the basis that the entropy term acts as a form of regularisation for the optimised prior. They report that results are improved relative to the moment-matching method, but overall the success of the approach is very limited. Experiments are carried out for a one-dimensional toy problem and with Gaussian process priors with various covariance kernels. In some cases samples of the resulting BNN prior resemble those of the corresponding Gaussian process, but in other cases they do not. Evaluation of the learned priors does not go beyond visual inspection of samples.

Tran et al. (2022) considered minimising a Wasserstein distance (e.g. Villani 2009) between the BNN and Gaussian process priors instead of the Kullback–Leibler divergence.

Definition 1 (Wasserstein distances)**.** *Let $(\mathcal{X},\, d)$ be a Polish metric space, and let $p \in [1,\, \infty)$. For any two probability measures $\mu,\, \nu$ on $\mathcal{X}$, the Wasserstein distance of order p between μ and ν is*

$$W_p(\mu,\, \nu) = \left(\inf_{\pi \in \Pi(\mu,\, \nu)} \int_{\mathcal{X}\times\mathcal{X}} d(x,\, y)^p \,\mathrm{d}\pi(x,\, y)\right)^{1/p}, \tag{4.33}$$

where $\Pi(\mu,\, \nu)$ is the set of all joint probability measures on $\mathcal{X} \times \mathcal{X}$ whose marginals are

[2]Recall that the differential entropy of a continuous random variable $\boldsymbol{X}$ with pdf $p(\boldsymbol{X})$ is $h(\boldsymbol{X}) = -\int p(\boldsymbol{X}) \log p(\boldsymbol{X})\, \mathrm{d}\boldsymbol{X}$.

μ and ν.

In particular, they considered the Wasserstein distance of order one, W_1, with d the Euclidean distance, i.e.

$$W_1(\mu,\, \nu) = \inf_{\pi \in \Pi(\mu,\, \nu)} \int_{\mathcal{X}\times\mathcal{X}} \|x - y\|\, \pi(x,\, y)\, \mathrm{d}x\mathrm{d}y. \tag{4.34}$$

Except in a few specific cases, such as when μ and ν are both Gaussian, computing the infimum in (4.34) is intractable, in analytical terms and by optimisation. However, as a classical result of optimal transport, (4.34) admits a dual form,

$$W_1(\mu,\, \nu) = \sup_{\|\vartheta\|_L \leq 1} \left[\int \vartheta(x)\, \pi(x)\, \mathrm{d}x - \int \vartheta(y)\, \nu(y)\, \mathrm{d}y \right] \tag{4.35a}$$

$$= \sup_{\|\vartheta\|_L \leq 1} \mathbb{E}_\pi[\vartheta(x)] - \mathbb{E}_\nu[\vartheta(x)], \tag{4.35b}$$

where $\vartheta : \mathcal{X} \to \mathbb{R}$ is 1-Lipschitz continuous. Tran et al. (2022) proposed to compute the form (4.35b) by parameterising $\vartheta(\,\cdot\,)$ with an auxiliary neural network. A regularisation term based on the gradient of $\vartheta(\,\cdot\,)$ with respect to its argument is used to enforce that $\vartheta(\,\cdot\,)$ is 1-Lipschitz. As in the KL divergence approach, a discrete set of $\boldsymbol{X}$ values is chosen to reduce the problem to a tractable finite-dimensional one — in this case some are drawn from training data and some are sampled uniformly in the domain. The resulting algorithm involves alternating steps of maximising with respect to the parameters of $\vartheta(\,\cdot\,)$ (i.e. the parameters of the auxiliary neural network) in order to estimate the Wasserstein distance, and minimising with respect to $\boldsymbol{\beta}$ (the variational parameters of the BNN prior) in order to minimise the Wasserstein distance. An attractive feature of this construction is that the objective (4.35b) does not require knowledge of either ones of the prior densities, corresponding to π and ν. It is sufficient to be able to sample from these measures and estimate the expectations in (4.35b) by Monte Carlo sampling. A consequence is that the method is neither limited to considering Gaussian BNN priors nor to Gaussian process target priors. This allowed the authors to consider alternative more complex families of BNN priors including hierarchical Gaussian distributions and priors constructed through normalising flows (Tabak & Vanden-Eijnden 2010, Tabak & Turner 2013). They found, based on visual inspection of prior samples, that the resulting optimised BNN priors reflected the target Gaussian process prior fairly well, particularly in the case of the more flexible normalising flow-based BNN prior. The estimated Wasserstein distance naturally provides a means of quantifying the accuracy of the prior fit. In a one-dimensional example they saw promising convergence of their algorithm in W_1, with the final value of W_1 obtained with the normalising flow-based prior an order of magnitude lower than those obtained with the other prior families. They employ BNNs in a range of higher-dimensional test problems and find that BNNs with Gaussian process-emulating priors outperform those with the default prior. However, their optimised priors are not evaluated directly in these higher-dimensional

cases. While this work demonstrates that the W_1 distance can be used to construct BNN priors which approximate Gaussian processes, the accuracy of this approximation (particularly in high-dimensional problems) is not thoroughly investigated. It is expected that the accuracy of these optimised priors as approximations to Gaussian processes is sensitive to a number of factors including:

- properties of the target Gaussian process prior, such as the form of its autocovariance function;
- the family of BNN priors over which the optimisation takes place, $\varrho(\boldsymbol{f}_{\mathrm{NN}}\,;\,\boldsymbol{\beta})$;
- the architecture of the neural network, including its depth, width, and activation functions;
- the fidelity of the representation of $\vartheta(\,\cdot\,)$;
- the finite set of $\boldsymbol{X}$ values chosen to construct a finite-dimensional optimisation scheme.

In general, more research is needed to build a better understanding of this method and its properties, but this study provides some hope for the problem of defining meaningful priors for neural networks. Until this problem is solved, the uncertainty quantification provided by BNNs in applications (including ours) cannot be considered a genuine application of Bayesian inference.

In the following section we apply methods of approximate inference to study the posteriors of neural networks given the default prior.

4.5 Posteriors on neural networks

4.5.1 Experiments with synthetic data

This section considers a simple one-dimensional test problem based on synthetically generated data, in order to illustrate the effects of the challenges discussed above (namely approximate inference and prior specification) on the posterior distribution $p(\boldsymbol{w} \mid \mathcal{D},\, \mathcal{M}_{\mathrm{NN}})$ and its pushforward onto $\boldsymbol{\theta}(\,\cdot\,)$, denoted $p(\boldsymbol{\theta}(\,\cdot\,) \mid \mathcal{D},\, \mathcal{M}_{\mathrm{NN}})$. Data $\mathcal{D} = \{X_i,\, Y_i\}_{i=1}^{N}$ is generated such that

$$X_i \sim \mathcal{N}(0,\, 1) \tag{4.36a}$$

$$Y_i \mid X_i \sim \mathcal{N}(\sin(4X_i),\, 0.01), \tag{4.36b}$$

independently for each i. The distribution of the X_i is chosen Gaussian in order that the data are inhomogeneous in X — we anticipate greater posterior uncertainty for values of X further from where there are data. We further discard any $X_i \in (-0.3,\, 0.3)$ to create a distinct gap in the data. The particularly simple form of $Y \mid X$, which is

Gaussian and such that $\mathrm{Var}(Y \mid X)$ is independent of X, is chosen to allow comparison with standard Gaussian process regression (Williams & Rasmussen 2006) — typically in Gaussian process regression one assumes $Y \mid X \sim \mathcal{N}(f_{\mathrm{GP}}(X),\, \varepsilon_{\mathrm{GP}})$ with $\varepsilon_{\mathrm{GP}}$ assumed known and a Gaussian process prior assigned to f_{GP}; given observations a Gaussian process posterior follows for f_{GP}, whose mean and covariance function can be computed exactly at specific values of X (Williams & Rasmussen 2006). We consider particularly simple probabilistic neural networks which have a single hidden layer with 32 neurons and a nonlinear activation function. The conditional variance is taken as known such that the network output f_{NN} is one-dimensional, encoding only the conditional mean. That is, we have $\theta(\,\cdot\,) = \mu(\,\cdot\,)$, h is the identity function and

$$Y \mid X,\, \boldsymbol{w},\, \mathcal{M}_{\mathrm{NN}} \sim \mathcal{N}(f_{\mathrm{NN}}(X\,;\,\boldsymbol{w}),\, 0.01). \tag{4.37}$$

As a benchmark we first show a Gaussian process regression solution. Figure 4.9 shows the posterior of $\mu(X)$. The Gaussian process prior used is zero mean with a so-called Gaussian (or squared-exponential) autocovariance function

$$C_{\mathrm{GP}}(X_1, X_2) = \sigma_c^2 \exp\left(-\frac{|X_1 - X_2|^2}{2l_c^2}\right), \tag{4.38}$$

where σ_c and l_c are hyperparameters. We set $\varepsilon_{\mathrm{GP}} = 0.1$ so that the relevant likelihood (given f_{GP}, or correspondingly f_{NN}) coincides with that of the neural network model. That is, we have

$$Y \mid X,\, \boldsymbol{w},\, \mathcal{M}_{\mathrm{GP}} \sim \mathcal{N}(f_{\mathrm{GP}}(X),\, 0.01). \tag{4.39}$$

The hyperparameters σ_c and l_c, and hence also the prior, are optimised to maximise the marginal likelihood of the data, i.e. we set

$$(\sigma_c^*,\, l_c^*) = \operatorname*{arg\,max}_{\sigma_c,\, l_c} \sum_{i=1}^{N} p(Y_i \mid X_i, \sigma_c, l_c, \mathcal{M}_{\mathrm{GP}}). \tag{4.40}$$

This procedure is known as empirical Bayes (Berger 1985). The posterior mean of $\mu(X)$ as shown in Figure 4.9 fits its true value well where there is sufficient data, and the posterior variance, as expected, is smallest near clusters of data, and grows large outside the range of observations. Thus, for this simple problem we conclude that Gaussian process regression provides a satisfactory result. Gaussian process regression is, however, in its basic form, limited to the case of stationary Gaussian process priors (i.e. where $C_{\mathrm{GP}}(X_1, X_2) = C_{\mathrm{GP}}(|X_1 - X_2|)$) and likelihoods with constant variance (i.e. $\varepsilon_{\mathrm{GP}}$ independent of X). A second limitation of basic Gaussian process regression is its $O(N^3)$ cost, due to a matrix inversion involved in computing the posterior covariance. In contrast Bayesian neural networks permit general parametric distributions as likelihoods and their training cost scales linearly with N. BNNs will be successful if

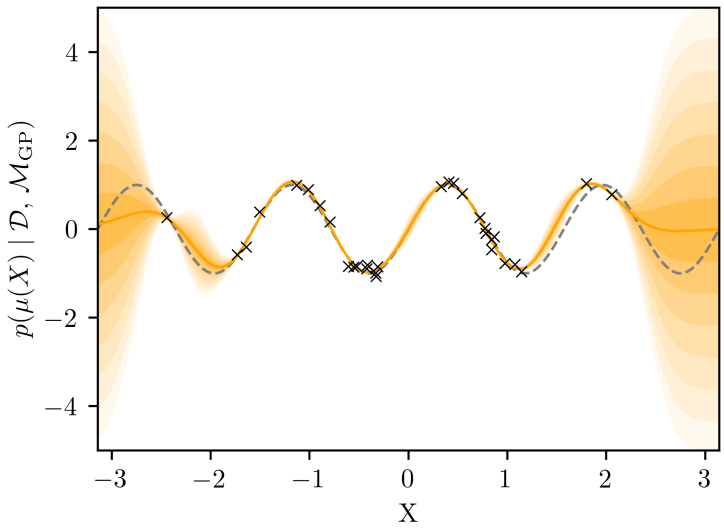


Figure 4.9: Gaussian process posterior $p(\mu(X) \mid \mathcal{D}, \mathcal{M}_{\text{GP}})$ given synthetic data and the squared-exponential autocovariance function. The grey dashed line is the true $\mu(X)$, the solid orange line is $\mathbb{E}[\mu(X) \mid \mathcal{D}, \mathcal{M}_{\text{GP}}]$, and the shaded regions show $\mathbb{E}[\mu(X) \mid \mathcal{D}, \mathcal{M}_{\text{GP}}] \pm j \,\text{Std}(\mu(X) \mid \mathcal{D}, \mathcal{M}_{\text{GP}})$ for $j \in \{1, \ldots, 7\}$. Black crosses indicate the data $\{X_i, Y_i\}_{i=1}^N$.

they can be implemented such that their results share the desirable features of Gaussian process regression but overcome its limitations. Here we look at the posterior of BNNs given Gaussian priors on the weights. We implement both MCMC sampling and variational Bayesian inference in Python using the TensorFlow (TensorFlow Developers 2021) and TensorFlow Probability (TensorFlow Probability Developers 2021) libraries.

We use the NUTS Hamiltonian Monte Carlo algorithm to sample the posterior of the simple probabilistic neural network described above. The network has a total of 97 parameters (64 weights and 33 biases). As priors for these we take the default prior, but with increased prior variance for the biases. In particular, we take $\boldsymbol{W} \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$ and $\boldsymbol{b} \sim \mathcal{N}(\mathbf{0}, 10^6\,\mathbb{I})$. This more diffuse prior for the biases was found empirically to lead to improved results. Three Markov chains were simulated from independent initial states drawn from the prior, each for 5000 burn-in steps and generating 5000 samples. It is known that achieving convergence (in distribution) of MCMC samples to the posterior is extremely challenging. This is due to the high-dimensionality of the parameter space (especially in large networks) and non-convexity of the posterior. Symmetries in the definition of the neural network, in particular combinatorially many weight permutation symmetries, lead to nonidentifiability. In practice this means that MCMC algorithms fail to explore the many modes of the posterior. It has been suggested, however, that while MCMC sampling fails to converge in terms of the posterior of the weights, the same algorithms may converge in terms of the posterior

on f_{NN}, or in this problem the posterior on $\mu(\,\cdot\,)$ (Izmailov et al. 2021, Papamarkou et al. 2022). That is to say, the pushforward of individual samples $f_{\mathrm{NN}}(X\,;\,\boldsymbol{w}_i)$ may converge in distribution to the pushforward of the posterior distribution of $\boldsymbol{w}$ even when samples $\boldsymbol{w}_i$ have not converged not converged to the posterior distribution of $\boldsymbol{w}$. This raises the possibility of accurate posterior prediction despite MCMC failing to converge at the level of the weights and biases. A statistic commonly used to detect a lack of convergence in MCMC sampling is the potential scale reduction factor (PSRF) (Gelman & Rubin 1992, Brooks & Gelman 1998), usually denoted $\hat{R}$. The PSRF relies on the simulation of several chains from independently sampled initial states — by comparing how well estimates of the posterior variance based on samples from each of these chains agree, $\hat{R}$ can indicate that chains are sampling from different measures, and in particular, that they have not converged. For a parameter ς, with ς_{jt} denoting the t^{th} of n samples in chain j, the PSRF is defined by

$$B/n = \frac{1}{m-1}\sum_{j=1}^{m}(\overline{\varsigma}_j - \overline{\overline{\varsigma}})^2 \tag{4.41a}$$

$$W = \frac{1}{m(n-1)}\sum_{j=1}^{m}\sum_{t=1}^{n}(\varsigma_{jt} - \overline{\varsigma}_j)^2 \tag{4.41b}$$

$$\hat{\sigma}_+^2 = \frac{n-1}{n}W + \frac{B}{n} \tag{4.41c}$$

$$\hat{R} = \frac{m+1}{m}\frac{\hat{\sigma}_+^2}{W} - \frac{n-1}{mn}, \tag{4.41d}$$

where $\overline{\varsigma}_j = \frac{1}{n}\sum_{t=1}^{n}\varsigma_{jt}$ and $\overline{\overline{\varsigma}} = \frac{1}{mn}\sum_{j=1}^{m}\sum_{t=1}^{n}\varsigma_{jt}$. Except in degenerate cases, $\hat{R} \geq 1$. A value of $\hat{R}$ close to 1 indicates that chains are in agreement, while a value much larger than 1 indicates a lack of convergence. A value of $\hat{R}$ close to 1 is thus a necessary but not a sufficient condition for convergence. A value of 1.1 is regularly cited as a cut-off value, but smaller values have also been recommended (Vats & Knudson 2021). From our samples we compute the PSRF for each component of $\boldsymbol{w}$. We find that for 75% of the weights $\hat{R} < 1.1$ while the same was true for only 6% of the biases. Figure 4.10 shows normalised histograms of a single weight $\boldsymbol{W}_{1_{1,1}}$ and a single bias $\boldsymbol{b}_{1_1}$ of the hidden layer based on samples from each chain — in this case we use the activation function $a(x) = \tanh x$. For $\boldsymbol{W}_{1_{1,1}}$ we have $\hat{R} = 1.01$ and the three histograms are fairly similar, while for $\boldsymbol{b}_{1_1}$ we have $\hat{R} = 11.48$ and histograms are much more diverse and appear to exhibit multimodality. This seemingly poor inference of the biases is suppressed only partially by taking a narrower prior for the biases. Overall, the inference of $\boldsymbol{w}$ by MCMC is, as expected, not convincing. Still, it is plausible that different chains sample distinct but equivalent high-probability regions of parameter space which differ only in the labelling of the parameters. The PSRF can also be calculated for functions of $\boldsymbol{w}$; in particular, we can calculate $\hat{R}$ for $\mu(X\,;\,\boldsymbol{w}) = f_{\mathrm{NN}}(X\,;\,\boldsymbol{w})$ at specific values of X based on samples $\boldsymbol{w}_i$. We compute $\hat{R}$ for $\mu(X\,;\,\boldsymbol{w})$ at 200 uniformly-spaced values of X in the range $[-\pi,\,\pi]$ and find that all

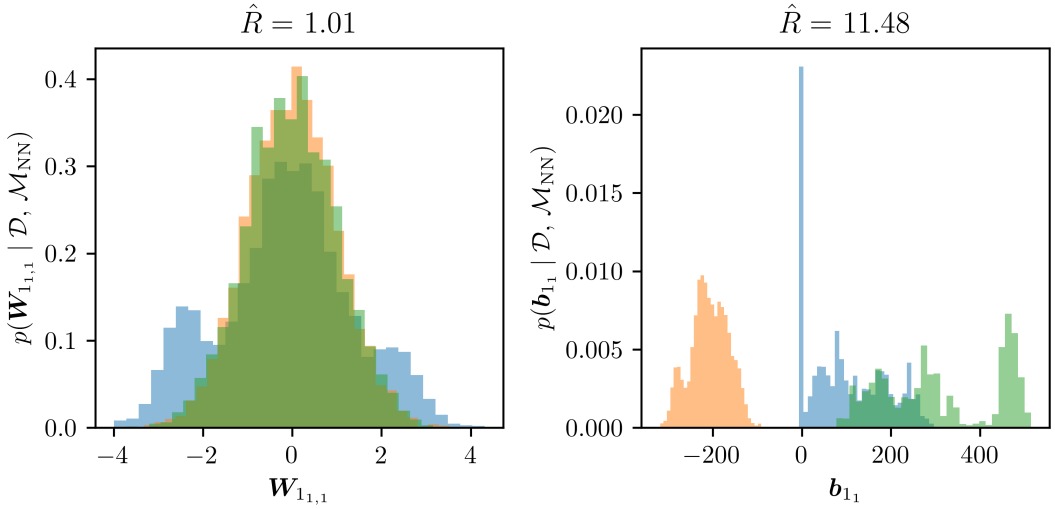


Figure 4.10: Histograms of MCMC samples of the posterior of example neural network parameters $\boldsymbol{W}_{1,1}$ and $\boldsymbol{b}_{1_1}$ given synthetic data and with $a(x) = \tanh x$. Each histogram corresponds to samples from one of the three chains. For each parameter the corresponding value of $\hat{R}$ is indicated.

values are close to 1, with the largest value being 1.016. Figure 4.11 shows normalised histograms of MCMC samples of $\mu(X = 0\,;\, \boldsymbol{w})$ and $\mu(X = \pi\,;\, \boldsymbol{w})$. Again, a histogram is shown for each chain, but in this case the histograms are in very close agreement, even for $X = \pi$ at which the largest value of $\hat{R} = 1.016$ was observed.

Figures 4.12, 4.13, 4.14 show, for different choices of activation function, a representation of the posterior of $p(\mu(X) \mid \mathcal{D}, \mathcal{M}_{\text{NN}})$, derived from the combined samples of all three chains, as a function of X. We find that reproducing these plots using samples from one chain at a time yields results which are essentially indistinguishable, further supporting the hypothesis that while MCMC samples do not converge to the posterior of $\boldsymbol{w}$, effective samples $\mu_i(X\,;\, \boldsymbol{w}) = f_{\text{NN}}(X\,;\, \boldsymbol{w}_i)$ do in fact convergence to the relevant posterior. Moreover, the results share the desirable properties of the Gaussian process regression solution for all choices of activation function considered: $a(x) = \tanh x$, $a(x) = \text{ReLU}(x)$, and $a(x) = \text{SiLU}(x)$, the sigmoid linear unit, defined $\text{SiLU}(x) := \frac{x}{1+e^{-x}}$. Despite the use of the default prior, and despite the non-convergence of MCMC samples of $\boldsymbol{w}$, these BNNs appear a useful model with useful uncertainty quantification.

Figure 4.15 shows, for $a(x) = \tanh x$, the result of a variational Bayesian approach to the same problem as described in Section 4.3.3. The surrogate posterior used is a diagonal Gaussian distribution on $\boldsymbol{w}$, so that $\boldsymbol{\xi}$ consists of a mean and a variance for each weight and bias. This is the de facto standard choice for variational Bayesian neural networks. To optimise $\boldsymbol{\xi}$ we use the Adam algorithm (Kingma & Ba 2015) along with the gradient estimate (4.25) setting $N_s = 1$. After optimising $\boldsymbol{\xi}$ the surrogate posterior is readily sampled to produce an ensemble of values of $\boldsymbol{w}$, and correspondingly

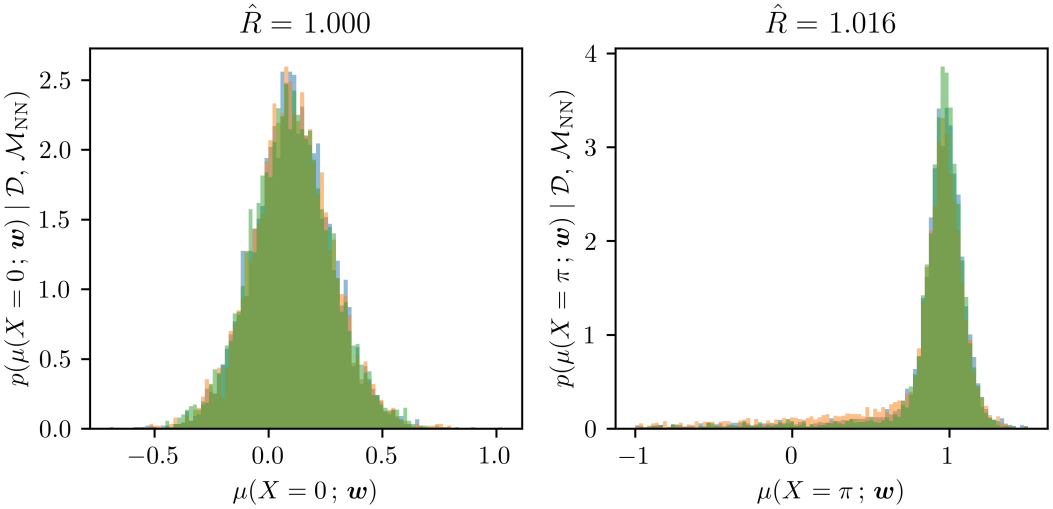


Figure 4.11: Histograms of MCMC samples of the posterior of $\mu(X = 0\,;\, \boldsymbol{w})$ and $\mu(X = \pi\,;\, \boldsymbol{w})$ given synthetic data and with $a(x) = \tanh x$. Each histogram corresponds to samples from one of the three chains. For each parameter the corresponding value of $\hat{R}$ is indicated.

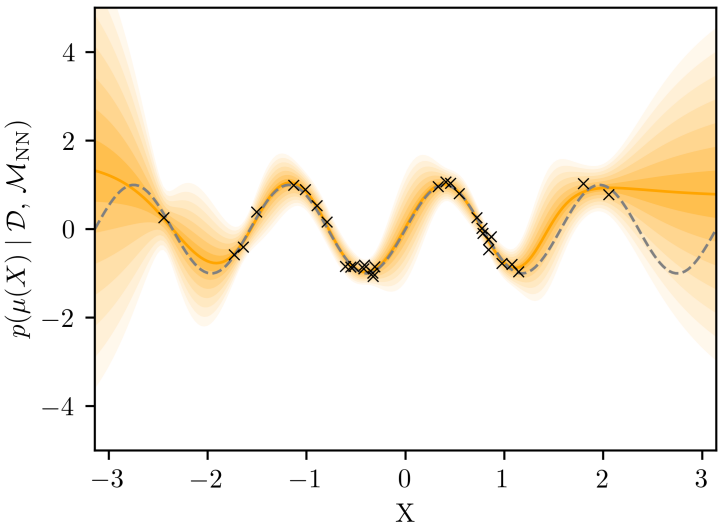


Figure 4.12: Neural network posterior $p(\mu(X) \mid \mathcal{D},\ \mathcal{M}_{\mathrm{NN}})$, derived from MCMC samples of the posterior, given synthetic data with $a(x) = \tanh x$. The grey dashed line is the true $\mu(X)$, the solid orange line is $\widehat{\mathbb{E}}[\mu(X) \mid \mathcal{D},\ \mathcal{M}_{\mathrm{NN}}]$, and the shaded regions show $\widehat{\mathbb{E}}[\mu(X) \mid \mathcal{D},\ \mathcal{M}_{\mathrm{NN}}] \pm j\,\widehat{\mathrm{Std}}(\mu(X) \mid \mathcal{D},\ \mathcal{M}_{\mathrm{NN}})$ for $j \in \{1, \ldots, 7\}$. Black crosses indicate the data $\{X_i,\ Y_i\}_{i=1}^{N}$.

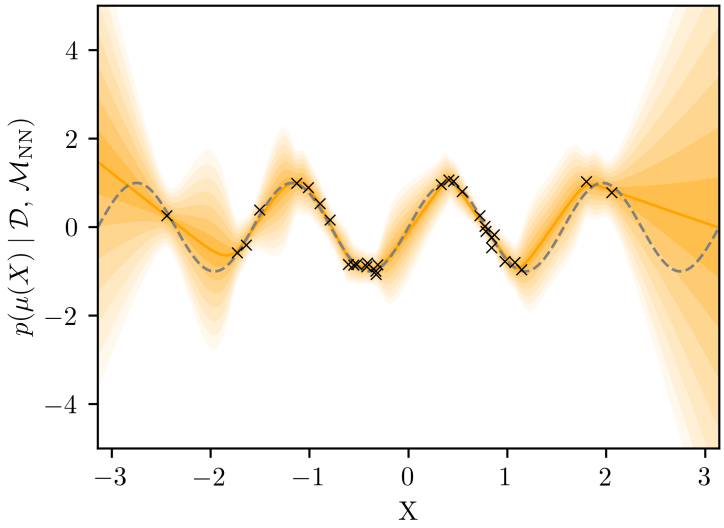


Figure 4.13: Same as Figure 4.12 but with $a(x) = \text{ReLU}(x)$.

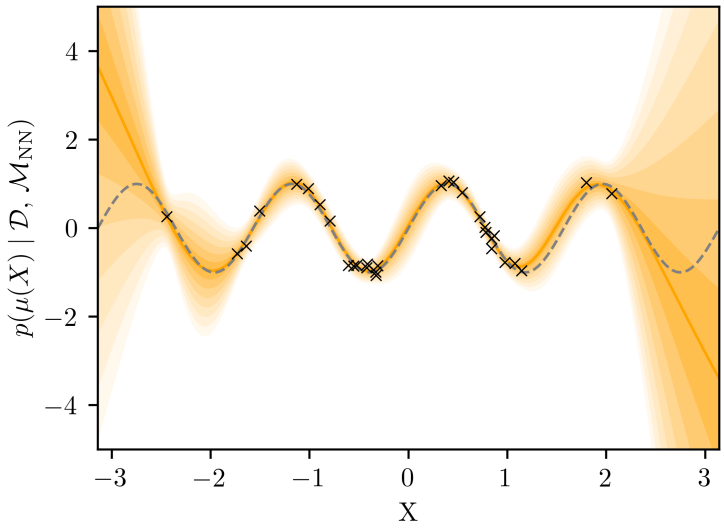


Figure 4.14: Same as Figure 4.12 but with $a(x) = \text{SiLU}(x)$.

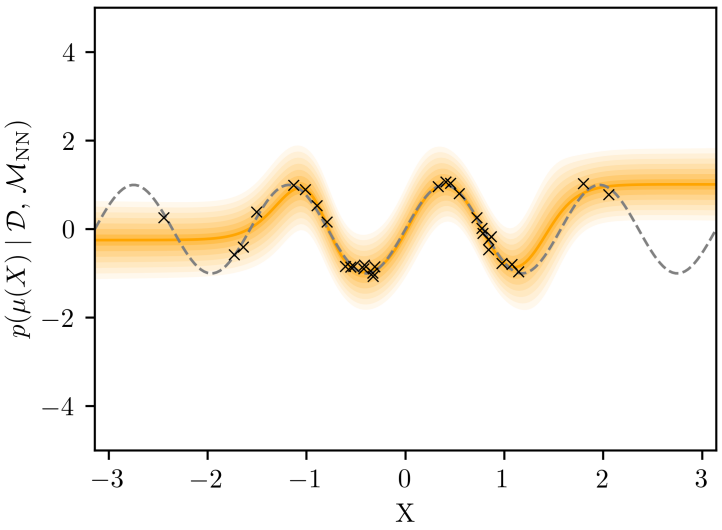


Figure 4.15: Variational Bayesian neural network posterior $p(\mu(X) \mid \mathcal{D}, \mathcal{M}_{\mathrm{NN}})$, based on 100 samples of the surrogate posterior, given synthetic data with $a(x) = \tanh x$. The grey dashed line is the true $\mu(X)$, the solid orange line is $\widehat{\mathbb{E}}[\mu(X) \mid \mathcal{D}, \mathcal{M}_{\mathrm{NN}}]$, and the shaded regions show $\widehat{\mathbb{E}}[\mu(X) \mid \mathcal{D}, \mathcal{M}_{\mathrm{NN}}] \pm j\,\widehat{\mathrm{Std}}(\mu(X) \mid \mathcal{D}, \mathcal{M}_{\mathrm{NN}})$ for $j \in \{1, \ldots, 7\}$. Black crosses indicate the data $\{X_i,\ Y_i\}_{i=1}^{N}$.

$\mu(X)$. We generate 100 samples to estimate the (surrogate) posterior mean and standard deviation of $\mu(X)$. The result, shown in Figure 4.15, is not satisfactory as the posterior variance of $\mu(X)$ appears almost constant in X and the fit to data points is poor for $|X| \gtrsim \pi/2$. We found that increasing N_s did not improve results. The poor performance of variational Bayesian inference in this problem is likely due to the small size of the network. Variational Bayesian inference may be more accurate for larger models, given that greater numbers of weights and biases correspond to more flexible surrogate posterior families.

In the following section we return to the MDN model of Chapter 3 with the aim of providing uncertainty quantification for the results presented there, and an example of the use of BNNs in an oceanographic context.

4.5.2 Application to drifter data

To apply approximate inference methods to the MDN model deployed in Chapter 3 is more challenging than for the test problem considered above. The difficulty lies in scale — both in terms of the number of model parameters and the number of data. While for the test problem there were 97 parameters and only 31 datapoints, the MDN model has 690,880 parameters and 9,181,955 datapoints. In the test problem we saw that MCMC led to better inference than with variational Bayesian inference, but for the drifter transition density problem MCMC is prohibitively expensive.

We trialled the use of the NUTS HMC sampler, as used for the test problem, on the MDN model with the drifter dataset. In order to reduce the computational burden of sampling, we implemented a reduced model by decreasing the number of neurons per layer from 256–512 to just 32, retaining the same depth with 6 hidden layers, and considering only one mixture component (i.e. $N_c = 1$), resulting in a model with $\dim(\boldsymbol{w}) = 5574$. This configuration was chosen to obtain the greatest reduction in $\dim(\boldsymbol{w})$ while maintaining enough flexibility for the model to represent the bulk behaviour captured by the full model — this was assessed by training various smaller models with maximum likelihood estimation and inspecting the results. Even for the reduced model we found that the stepsize required to obtain an adequate acceptance rate was unacceptably small, in the sense that sampling was not efficient enough to be practical. This is not surprising given the high dimensionality of parameter space.

Researchers have employed a number of strategies to reduce the cost of MCMC for large BNNs, but these come at the cost of introducing bias in sampling. One increasingly popular strategy is the use of stochastic gradient MCMC (Chen et al. 2014, Wenzel et al. 2020), where, in samplers such as HMC and MALA, the gradient of the unnormalised log posterior used to calculate proposals is replaced with a gradient estimate computed using only a subset (or minibatch) of the data. The use of stochastic gradients in these samplers biases the stationary distribution to which samples converge. Another strategy is to skip the Metropolis–Hastings accept/reject step in MCMC samplers, again introducing bias in sampling, sacrificing accuracy for computational efficiency. Izmailov et al. (2021) implemented full-batch HMC but without the Metropolis–Hastings correction for benchmark problems is image classification and sentiment analysis; this came with an extreme computational cost and required parallelisation of computations over hundreds of tensor processing units (TPUs). Since careful implementation and testing of stochastic gradient MCMC algorithms is beyond the scope of this work, and full-batch MCMC is prohibitively expensive, we are unable to proceed with MCMC for the transition density problem. Instead we implement variational Bayesian inference.

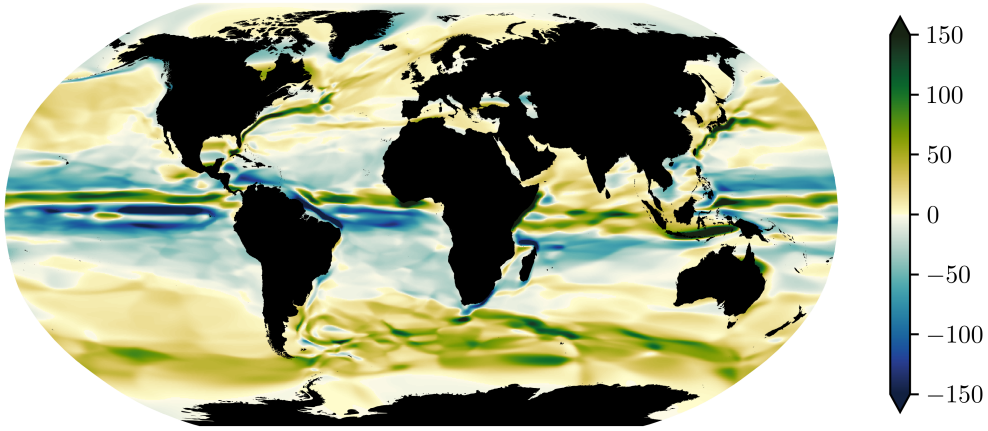
To implement variational Bayesian inference for the MDN model we must choose a prior and surrogate posterior. As prior we take the default prior, and we use again a Gaussian surrogate posterior with diagonal covariance. To optimise $\boldsymbol{\xi}$ we use the Adam algorithm (Kingma & Ba 2015) along with the minibatch-based gradient estimate (4.26) setting $N_s = 1$ and $B = 8192$ (the same batch size as used when training the MDN model in Chapter 3 with maximum likelihood). Since the cost of implementing this approach is not much greater than training with maximum likelihood, we are able to use the same network architecture as in Chapter 3 without reductions in the number of neurons and keeping $N_c = 32$. The model considered here differs from that in Chapter 3 only in the choice of activation function — we found that $a(x) = \mathrm{ReLU}(x)$ led to better results than $a(x) = \tanh x$. In contrast we found that switching to $a(x) = \mathrm{ReLU}(x)$ had little effect on the results of MDN models trained with maximum likelihood. As in Chapter 3 we standardise data and make use of early stopping. Training took approximately 2 hours and 10 minutes.

Figure 4.16 shows the posterior mean (estimated from 100 realisations of $\boldsymbol{w}$) of mean displacement over four days in the zonal and meridional directions, which does not differ significantly from the corresponding maximum likelihood estimate shown in Figure 3.10. Figure 4.17 shows the corresponding posterior standard deviations. High values of posterior standard deviation are seen to generally correspond to areas where the gradient in the posterior mean is large, such as along western boundary currents and at the equator. In particular, we see regions of high uncertainty at the edges of the Gulf Stream and the Brazil, Agulhas and Kuroshio currents, with lower posterior standard deviation in their cores.

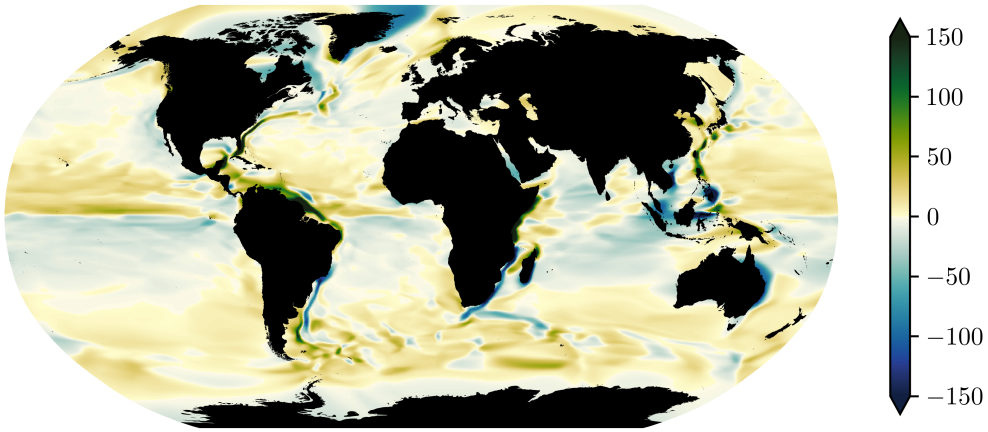
Figure 4.18 shows both the posterior mean (panel (a)) and the posterior coefficient of variation[3] (panel (b)) of the lateral diffusivity estimate (3.25), again estimated from 100 samples of $\boldsymbol{w}$. The posterior mean estimate agrees well with the corresponding maximum likelihood estimate shown in Figure 3.11. The coefficient of variation, as a nondimensional measure of uncertainty, allows to meaningfully assess how posterior uncertainty varies with position $\boldsymbol{X}_0$. The relevant map in Figure 4.18(b) shows complex spatial variation. It is difficult to determine how much of this variation is representative of the true posterior distribution and how much is an artefact of error in the surrogate posterior. The surrogate posterior is sure to exhibit some error due to the combined effects of (i) the approximation made in adopting a surrogate posterior family, and (ii) challenges in optimising the loss function (4.19) which is generically highly non-convex and whose value can only be estimated. Indeed, we saw clearly such error in the test problem and Figure 4.15. However, there is reason to believe that, since the network used here is much larger than that considered for the test problem, and since the surrogate posterior used here is correspondingly more flexible, the first source of surrogate posterior error may be dominated by the second. Moreover, complex spatial structure in uncertainty is not surprising given that the distribution of drifter observations is highly nonuniform. We see, as expected, that levels of uncertainty are relatively high in high-latitude regions. On the whole, however, uncertainty appears low. Even in high latitude regions where observations are most sparse, such as those parts of the Arctic Ocean which are persistently covered by sea ice, the posterior coefficient of variation of lateral diffusivity is of order unity. Ying et al. (2019) applied Bayesian inference to estimate diffusivity from Lagrangian data, but since this was in the context of an idealised quasigeostrophic ocean model their results are not directly comparable.

Previous studies have made use of bootstrap methods (Efron 1979) when estimating diffusivity from Lagrangian data. Bootstrapping refers to a class of frequentist methods for estimating the standard error of an estimator and is based on resampling of data; the standard error can be considered roughly the frequentist analogue of the posterior standard deviation. Both Griesel et al. (2010) and Klocker et al. (2012) applied bootstrapping when estimating diffusivity from Lagrangian float data simulated at depth in models of the Southern Ocean. Roach et al. (2018) applied bootstrapping

[3]The coefficient of variation is the ratio of the standard deviation and the mean. It provides a nondimensional measure of uncertainty.

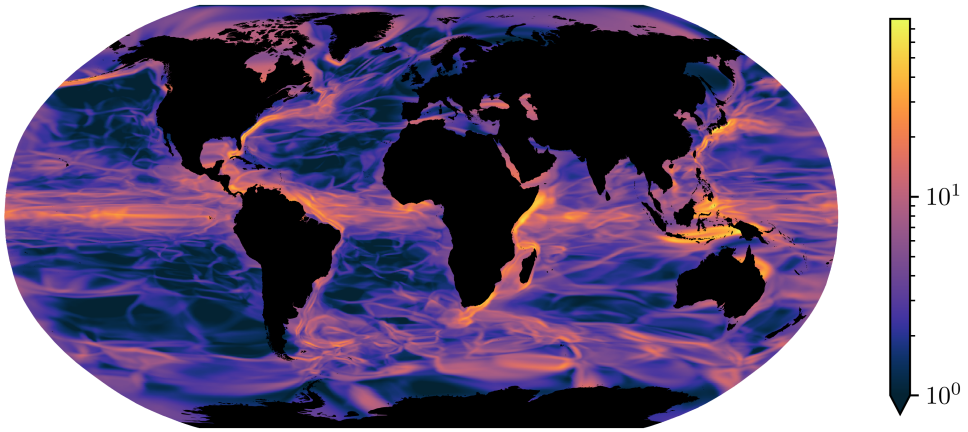


(a) Posterior mean of mean of 4-day zonal displacement (km).

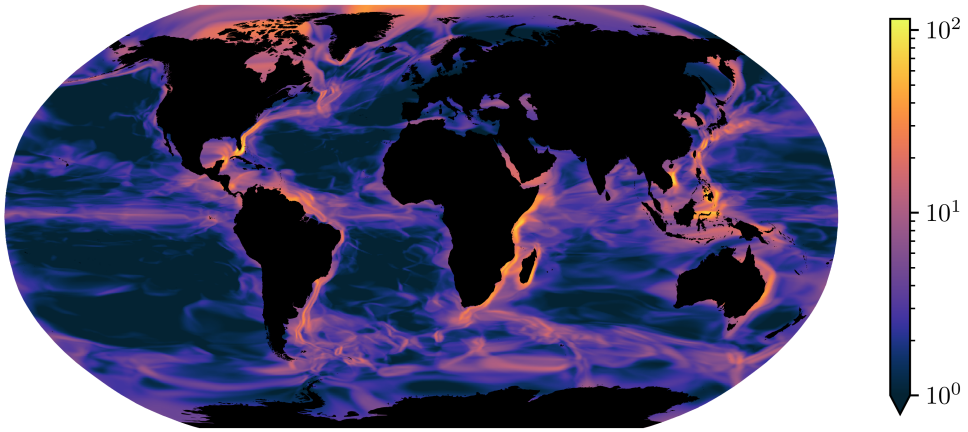


(b) Posterior mean of mean of 4-day meridional displacement (km).

Figure 4.16: Posterior mean of mean of displacements from the variational Bayesian MDN model, with $\tau = 4$ days, as a function of initial position.



(a) Posterior standard deviation of mean of 4-day zonal displacement (km).



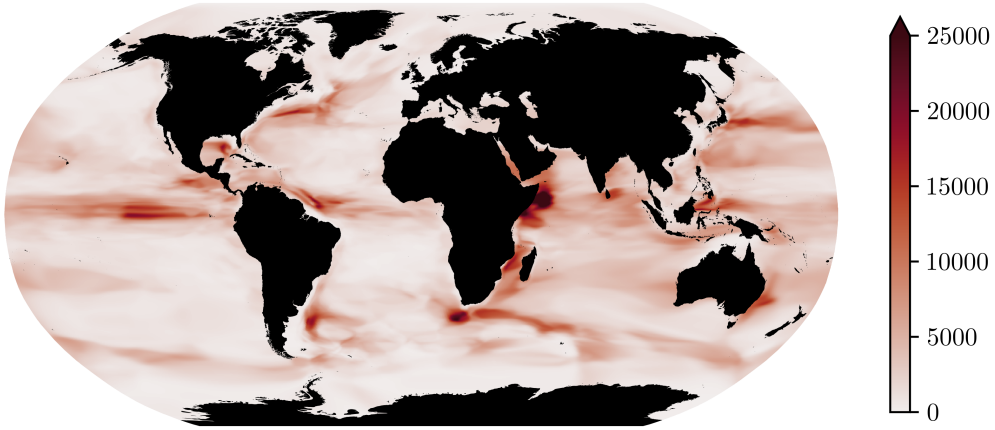
(b) Posterior standard deviation of mean of 4-day meridional displacement (km).

Figure 4.17: Posterior standard deviation of mean of displacements from the variational Bayesian MDN model, with $\tau = 4$ days, as a function of initial position.

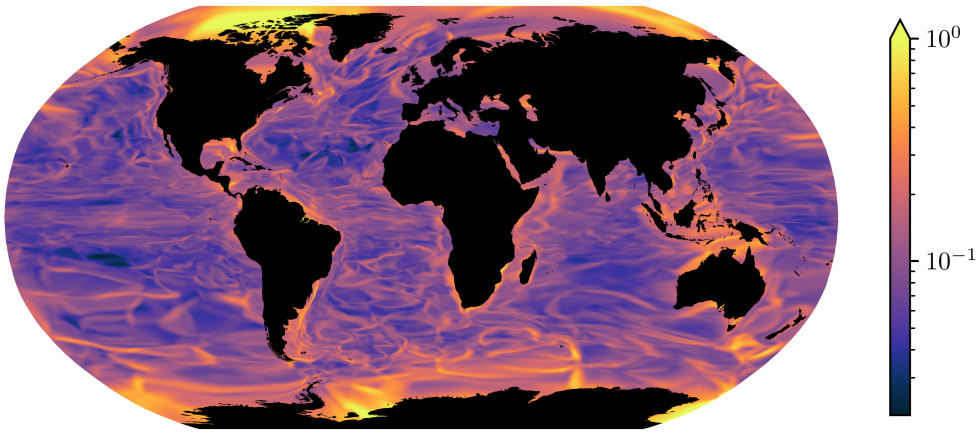
to diffusivity estimates both at the near-surface using Global Drifter Program data and at depth using data from Argo floats (Wong et al. 2020). For near-surface lateral diffusivity they reported estimated standard error values less than 30% in most areas, however, no estimates were reported for the poorest sampled areas. While our estimate differs somewhat from that of Roach et al. (2018) (likely due to their use of a significantly longer time-lag τ, between 10 and 50 days, while we take $\tau = 4$ days), their assessment of uncertainty is broadly compatible with ours. While there are noticeable local differences in our maps, these may be primarily due to the choice of time-lag, and the range of values of posterior coefficient of variation is similar to the range of values of normalised standard error they report. We highlight that bootstrapping is not feasible for estimators derived from complex models such as the MDN, since this would require retraining the neural network many times with different subsets of the data at considerable cost.

The posterior mean and coefficient of variation provide a convenient but limited summary of the posterior distribution. By sampling the surrogate posterior we can obtain a comprehensive description of our state of knowledge. As an example, we show in Figure 4.19 a histogram of 1000 posterior samples of the lateral diffusivity estimate $K(\boldsymbol{X}_0^\dagger)$, as defined in (3.25), where $\boldsymbol{X}_0^\dagger$ is the point in the Gulf Stream shown in Figure 3.8(a). In principle any posterior statistic can be estimated.

Overall, variational Bayesian inference appears to have been effective for this problem. However, there is reason to be sceptical of these results. Namely, (i) we have used the default prior, despite its known deficiencies, and (ii) the accuracy of the surrogate posterior is essentially unknown and cannot easily be tested against MCMC. Regarding (i), recall that the default prior is highly sensitive to the choice of activation function, as discussed in section 4.4. In Figure 4.20 we show again the posterior mean and coefficient of variation of our diffusivity estimate, but with $a(x) = \tanh x$. Although the posterior mean estimate is not significantly changed, the coefficient of variation map is markedly different. In this case the spatial patterns seen are less convincingly physical and suggest that the choice of activation function has an unintended effect on the surrogate posterior. The location of the global maximum of uncertainty is consistent in both cases (in the Arctic Ocean, North of the Beaufort Sea), but its value is less than half with $a(x) = \tanh x$ than with $a(x) = \mathrm{ReLU}(x)$. This discrepancy is not surprising given that changing the activation function changes significantly the pushforward of the prior distribution, and will affect the true posterior distribution, but could additionally be due to the effect of the activation function on the loss function and the local optimum identified by stochastic gradient variational Bayesian inference. For further comparison, Figure 4.21 shows a histogram of 1000 posterior samples of $K(\boldsymbol{X}_0^\dagger)$ with $a(x) = \tanh x$. Relative to Figure 4.19 we see that the mean appears similar but the variance is slightly larger.



(a) Posterior mean of lateral diffusivity estimate ($m^2 s^{-1}$).



(b) Posterior coefficient of variation of lateral diffusivity estimate.

Figure 4.18: Posterior mean and posterior coefficient of variation of lateral diffusivity estimate from the variational Bayesian MDN model, with $\tau = 4$ days, as a function of initial position.

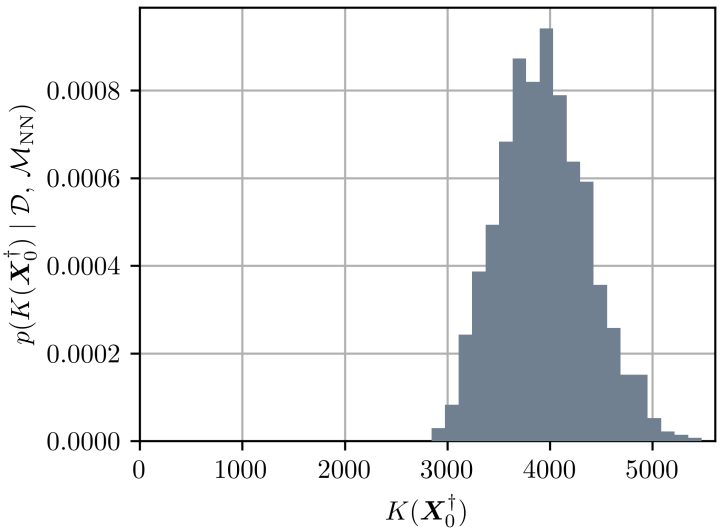
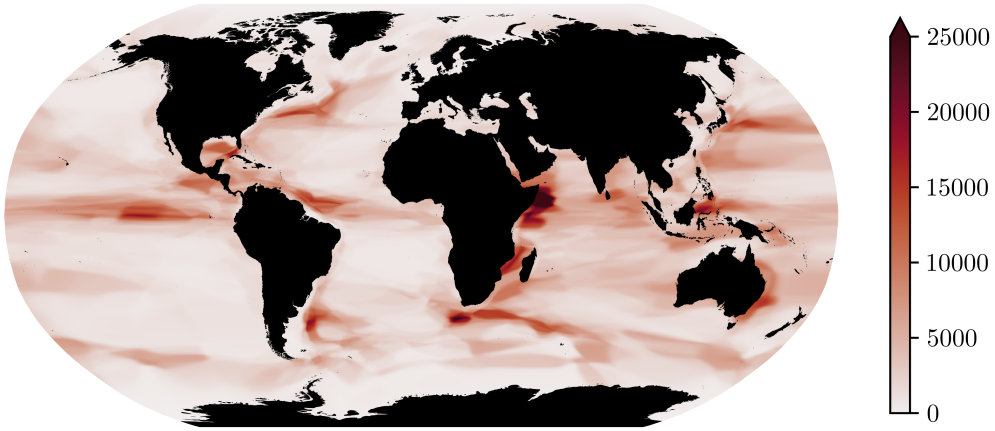


Figure 4.19: Histogram of samples of the lateral diffusivity estimate $K(\boldsymbol{X}_0^\dagger)$ where $\boldsymbol{X}_0^\dagger$ is the point in the Gulf Stream shown in Figure 3.8(a).

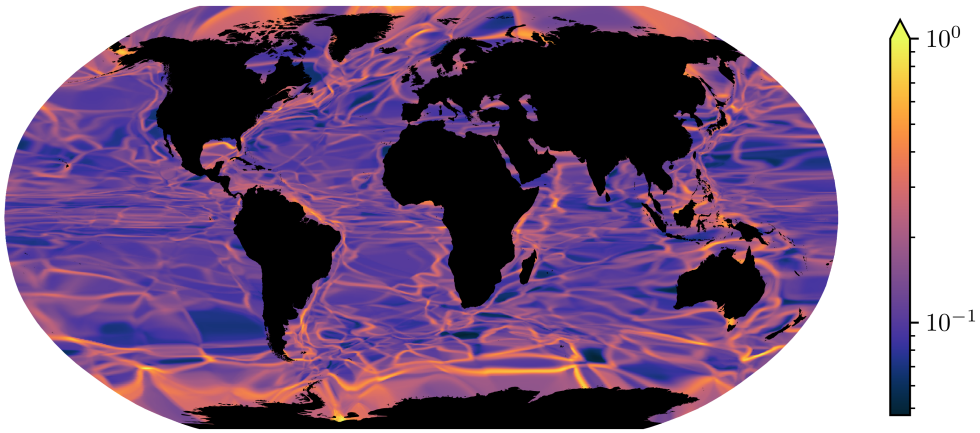
4.6 Conclusions

This chapter explored state-of-the-art methods in Bayesian machine learning and their applicability to scientific problems. We discussed the key difficulties in implementing BNNs, namely the specification of meaningful prior distributions on network parameters and the design of accurate approximate inference methods. If genuinely Bayesian neural networks can be constructed, it is unlikely that off-the-shelf methods will suffice, which invoke the default prior and stochastic gradient variational Bayesian inference. These methods provide only a coarse approximation to Bayesian inference. The prior distributions of Tran et al. (2022) which approximate desired Gaussian processes may provide a way forward for the first hurdle, but affordable accurate posterior sampling/approximation via MCMC or improved variational Bayesian methods appears out of reach for large neural networks for now.

We applied standard methods to the problem addressed in Chapter 3. While the results appear satisfactory at a superficial level, they are not robust to simple changes to the neural network configuration such as the choice of activation function, and hence should be received with healthy scepticism. We hope that by drawing attention to these deficiencies, we will raise awareness within the scientific community to the risks of applying Bayesian neural networks with off-the-shelf methods. Such methods are already being applied to Earth system modelling (Luo et al. 2022) and to the analysis of ocean data (Clare et al. 2022), despite the problems we discuss and without drawing attention to them. For instance, both of these works consider only the default prior and neglect to assess the sensitivity of their results to this choice. As machine learning methods are further integrated into scientific research it is crucial



(a) Posterior mean of lateral diffusivity estimate.



(b) Posterior coefficient of variation of lateral diffusivity estimate.

Figure 4.20: Same as Figure 4.18 but with $a(x) = \tanh x$.

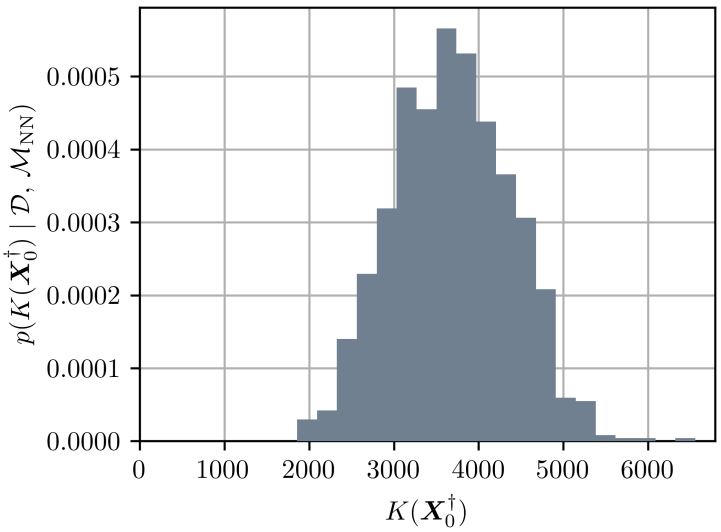


Figure 4.21: Same as Figure 4.19 but with $a(x) = \tanh x$.

that their weaknesses are understood and made transparent. We should beware that the standards expected of inference in scientific contexts may well be more stringent than those in other fields.

Chapter 5

Conclusions and future work

5.1 Conclusions

This thesis developed a Bayesian approach to the stochastic modelling of ocean transport. In Chapter 1 we outlined our view of diagnostic modelling and parameterisation of ocean transport as inverse problems and motivated the development of advanced methods for their solution.

In Chapter 2 we introduced Bayesian model comparison and used it to compare the classical Brownian and Langevin models of single particle dispersion in two-dimensional turbulence. After first illustrating the method with a test case involving synthetic data generated from the Langevin model, we generated trajectory data using simulations of a forced–dissipative model. We demonstrated the sensitivity of model choice to the sampling interval of the data, concluding that while the Langevin model is favoured on short timescales, the Brownian model performs equally well on larger timescales, corresponding intuitively to an asymptotically diffusive regime in the particle dynamics. Our analysis further dealt systematically with uncertainty in model parameters and illuminated issues of identifiability in different timescale regimes. This work was published as Brolly et al. (2022).

In Chapter 3 we developed a model of the transition density of near-surface ocean dynamics using a probabilistic neural network. Our model approximates the transition density with a Gaussian mixture distribution whose parameters are functions of longitude and latitude represented by a neural network trained to maximise the likelihood of data. As data we considered trajectories of satellite-tracked drifting buoys. This model is useful in two ways, firstly as an indirect means of estimating spatially-varying statistics such as diffusivity, defined as functionals of the transition density, and secondly as the basis for a discrete-time Markov process surrogate model of drifter dynamics. The latter use was demonstrated by simulations of clustering in ocean gyres. Our modelling approach provides a convenient framework for inference and modelling based on sparse nonuniform data, such as that collected by drifters. This work was published as Brolly (2023).

In Chapter 4 we considered the use of Bayesian neural networks to allow the

model of Chapter 3 to be incorporated into the Bayesian framework. We discussed the challenges in applying Bayesian inference to neural network models and described the various methods of approximate inference that can be applied to approximate Bayesian inference. We also discussed the difficulty of constructing meaningful prior distributions on neural network parameters and showed that the common choice of a Gaussian distribution with identity covariance matrix is not fit for purpose. Through experiments with a test problem we highlighted the deficiencies of state-of-the-art methods including variational Bayesian inference. We applied variational Bayesian inference to the model of Chapter 3, using the commonly used Gaussian prior distribution, and found that the results, although acceptable at a superficial level, were not robust to seemingly benign modelling choices, such as the choice of activation function. We conclude that, while neural network-based models can be useful, as demonstrated in Chapter 3, deficiencies in current methods for applying Bayesian inference to large neural networks mean that it is not yet possible to quantify robustly the uncertainty in their predictions.

5.2 Future work

The methods discussed in this thesis could be applied to a variety of other problems. The application of BMC in Chapter 2 could be used to compare more sophisticated stochastic models such as the Matérn process model proposed by Lilly et al. (2017), which was found to provide an excellent fit to similar data from isotropic 2D turbulence. The method could also be applied to data from statistically inhomogeneous flows. In particular, there exist inhomogeneous versions of the Brownian and Langevin models considered in that chapter, where parameters such as diffusivity vary in space. Ying et al. (2019) showed that Bayesian inference can be applied to the inhomogeneous Brownian model with Lagrangian data using MCMC — in this case a version of Laplace's method based on the sample mean and variance of the same posterior samples could be used to give a crude approximation to the evidence, or a more sophisticated method such as the nested sampling algorithm of Skilling (2004) could be employed for greater accuracy. Beyond surrogate models BMC could also be used to compare stochastic parameterisations of subgrid processes in turbulence or Earth system modelling.

The transition density modelling of Chapter 3 could be applied to trajectory data collected at depth in the ocean by Argo floats (Wong et al. 2020) to infer statistics of transport at depth. Modelled transition densities could also be used to estimate travel times and most likely paths of marine debris, as well as to study the stability of garbage patches. These issues have previously been studied with transition matrices (O'Malley et al. 2021, Miron et al. 2021), but our MDN approach could provide an alternative method. MDNs can also be used to model other distributions relating to dynamics. For instance, as previously mentioned, velocity structure functions are moments of the conditional distribution with density (3.2). In a stationary, isotropic

flow this reduces to

$$p\left(\boldsymbol{u}(\boldsymbol{x}_1,\, t) - \boldsymbol{u}(\boldsymbol{x}_2,\, t) \mid \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|\right). \tag{5.1}$$

Structure functions estimated from surface drifters have been used to show the existence of a dual cascade of kinetic energy in the ocean and to study its seasonality (Balwada et al. 2022). The conditional density (5.1) is a natural candidate for modelling with mixture density networks. While studies like that of Balwada et al. (2022) typically assume isotropy and discretise in $r = \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|$ to allow estimation of structure functions, MDNs may enable anisotropic analyses and avoid discretisation. MDNs could also be used to build stochastic parameterisations. Consider the conditional density

$$p\left(\text{subgrid forcing} \mid \text{coarse model state}\right). \tag{5.2}$$

Where relevant training data can be generated, for example from a library of simulation data, MDNs could be employed to learn (5.2). A stochastic parameterisation could then be provided by sampling the resulting distribution. An example of this has already been provided by (Guillaumin & Zanna 2021), but parameterisations for many other processes could be built in this way.

As highlighted in Chapter 4, there remain many open questions around the implementation of Bayesian neural networks. Since the specification of priors is a leading issue, for small and large networks, further research along the lines of Tran et al. (2022), who proposed an algorithm for learning neural network priors which approximate Gaussian processes, could be an important step forward. More work is needed to assess the sensitivity of their algorithm's performance to the various factors listed in Section 4.4. The other leading issue, posterior sampling, could also benefit from further research. In our test problem we found that MCMC was surprisingly effective in sampling the posterior predictive distribution despite a clear lack of convergence at the level of the posterior on the network parameters, $\boldsymbol{w}$. A better understanding of this behaviour would aid the design of effective sampling methods. There are open questions, too, around variational Bayesian inference for neural networks. The accuracy of the surrogate posterior depends on the flexibility of surrogate posterior family used and the effective maximisation of the ELBO. It would be helpful to establish which of these sources of error dominates in practice. Custom algorithms may be needed to find satisfactory solutions to the challenging nonconvex optimisation problem of ELBO maximisation. Ultimately, where BNNs can be implemented properly, they should be assessed using BMC. This may well come with its own challenges, but should remain a goal nonetheless.

Appendix A

Appendix to Chapter 2

A.1 Langevin likelihood for position observations

To derive $p(\Delta\mathcal{X}_\tau \mid \mathcal{M}_L(\gamma, k))$ we simplify notation by recognising that all particles are independent under $\mathcal{M}_L$ and that dynamics in each spatial dimension are independent. We therefore need only calculate $p(\Delta\mathcal{X}_\tau \mid \mathcal{M}_L(\gamma, k))$ in the one-dimensional, single-particle case. We proceed by: (i) showing that the joint process of particle position and velocity is an order-one vector autoregressive process, or VAR(1) process, and hence, has a Gaussian likelihood, (ii) calculating the mean and covariance for a sequence of joint position–velocity observations, and (iii) marginalising this likelihood to find $p(\Delta\mathcal{X}_\tau \mid \mathcal{M}_L(\gamma, k))$.

It can be shown that for the one-dimensional Langevin equation

$$\boldsymbol{Y}_n \mid \boldsymbol{Y}_{n-1} \sim \mathcal{N}\left(\begin{pmatrix} U_n\varphi(\gamma\tau)\tau \\ U_n e^{-\gamma\tau} \end{pmatrix}, C\right), \tag{A.1}$$

where $\boldsymbol{Y}_n \coloneqq (\Delta X_n, U_{n+1})^{\mathrm{T}}$ and

$$C_{11} = 2k\tau\left(1 - 2\varphi(\gamma\tau) + \varphi(2\gamma\tau)\right), \tag{A.2a}$$
$$C_{12} = C_{21} = k(\varphi(\gamma\tau)\gamma\tau)^2, \tag{A.2b}$$
$$C_{22} = 2k\gamma^2\tau\varphi(2\gamma\tau). \tag{A.2c}$$

This follows from the well-known solution of the Ornstein–Uhlenbeck process,

$$U(t) = U(0)e^{-\gamma t} + \gamma\sqrt{2k}\int_0^t e^{-\gamma(t-t')}\,\mathrm{d}W(t') \tag{A.3}$$

and the corresponding solution for the position,

$$X(t) = X(0) + \int_0^t U(t')\,\mathrm{d}t' \tag{A.4a}$$

$$= X(0) + U(0)\varphi(\gamma t)t - \sqrt{2k}\int_0^t e^{-\gamma(t-t')}\,\mathrm{d}W(t') + \sqrt{2k}\,W(t). \tag{A.4b}$$

Therefore, we can write the Langevin model in the time-discretised form

$$\boldsymbol{Y}_n = A\boldsymbol{Y}_{n-1} + \boldsymbol{\varepsilon}_n, \tag{A.5}$$

where

$$A = \begin{pmatrix} 0 & \varphi(\gamma\tau)\tau \\ 0 & e^{-\gamma\tau} \end{pmatrix}, \tag{A.6}$$

and $\boldsymbol{\varepsilon}_n$ is a mean-zero white-noise process with covariance matrix C.

The discrete process (A.5) has the form of a VAR(1) process. Furthermore, Y_n is stationary with mean and stationary variance

$$\boldsymbol{\mu} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},\ V = \begin{pmatrix} 2k\tau(1-\varphi(\gamma\tau)) & k\varphi(\gamma\tau)\gamma\tau \\ k\varphi(\gamma\tau)\gamma\tau & k\gamma \end{pmatrix}. \tag{A.7}$$

To see this, note that the marginal distribution of $U(t)$ at any time is given by the stationary distribution of the Ornstein–Uhlenbeck process,

$$U(t) \sim \mathcal{N}(0, k\gamma), \tag{A.8}$$

which gives μ_2 and V_{22}. Using (A.1), (A.8) and Lemma 1 yields μ_1 and V_{11}. Finally, $V_{12} = V_{21}$ can be calculated using (A.1) and the law of total covariance — specifically,

$$\mathrm{Cov}(\Delta X_n, U_{n+1}) = \mathbb{E}\left[\mathrm{Cov}(\Delta X_n, U_{n+1} \mid U_n)\right] + \mathrm{Cov}\left(\mathbb{E}\left[\Delta X_n \mid U_n\right], \mathbb{E}\left[U_{n+1} \mid U_n\right]\right) \tag{A.9a}$$

$$= C_{12} + \mathrm{Cov}\left(U_n\varphi(\gamma\tau)\tau,\ U_n e^{-\gamma\tau}\right) \tag{A.9b}$$

$$= C_{12} + \varphi(\gamma\tau)\tau e^{-\gamma\tau}\mathrm{Var}(U_n) \tag{A.9c}$$

$$= k\varphi(\gamma\tau)\gamma\tau, \tag{A.9d}$$

recalling that $\mathrm{Var}(U_n) = k\gamma$.

The autocovariance of $\boldsymbol{Y}_n$ is defined as

$$G(m) \coloneqq \mathbb{E}[(\boldsymbol{Y}_n - \boldsymbol{\mu})(\boldsymbol{Y}_{n-m} - \boldsymbol{\mu})^{\mathrm{T}}], \tag{A.10}$$

where $m \in \mathbb{Z}$. Notice that $G(0)$ is the stationary variance of $\boldsymbol{Y}_n$. Postmultiplying (A.5)

by $\boldsymbol{Y}_{n-m}^{\mathrm{T}}$ and taking expectations gives

$$\mathbb{E}\left[\boldsymbol{Y}_n \boldsymbol{Y}_{n-m}^{\mathrm{T}}\right] = A\,\mathbb{E}\left[\boldsymbol{Y}_{n-1} \boldsymbol{Y}_{n-m}^{\mathrm{T}}\right] + \mathbb{E}\left[\boldsymbol{\varepsilon}_n \boldsymbol{Y}_{n-m}^{\mathrm{T}}\right]. \tag{A.11}$$

Thus, for $m > 0$, since $\boldsymbol{Y}_{n-m}$ is independent of $\boldsymbol{\varepsilon}_n$,

$$G(m) = A\,G(m-1) \tag{A.12}$$

Therefore, $G(m)$ can be calculated recursively for $m > 0$ as

$$G(m) = A^m\,G(0) \tag{A.13a}$$
$$= A^m\,V. \tag{A.13b}$$

Note that (A.12) is an instance of a Yule–Walker equation (Lütkepohl 2007, pp. 26-27).

Thus, the joint distribution of a sequence of observations $\{\boldsymbol{Y}_n : n \in \{0, \cdots, N_\tau - 1\}\}$ is given by

$$\begin{pmatrix} \boldsymbol{Y}_0 \\ \boldsymbol{Y}_1 \\ \vdots \\ \boldsymbol{Y}_{N_\tau - 1} \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} G(0) & G(1) & \cdots & G(N_\tau - 1) \\ G(1) & \ddots & \ddots & \vdots \\ \vdots & \ddots & & \\ & & & G(1) \\ G(N_\tau - 1) & & G(1) & G(0) \end{pmatrix}\right). \tag{A.14}$$

Marginalising (A.14) for the distribution of $(\Delta X_0, \cdots, \Delta X_{N_\tau - 1})^{\mathrm{T}}$ we find

$$\begin{pmatrix} \Delta X_0 \\ \Delta X_1 \\ \vdots \\ \Delta X_{N_\tau - 1} \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} G_{11}(0) & G_{11}(1) & \cdots & G_{11}(N_\tau - 1) \\ G_{11}(1) & \ddots & \ddots & \vdots \\ \vdots & \ddots & & \\ & & & G_{11}(1) \\ G_{11}(N_\tau - 1) & & G_{11}(1) & G_{11}(0) \end{pmatrix}\right). \tag{A.15}$$

Using (A.7) and (A.12) it is easy to see that for $m \geq 1$

$$G(m) = \begin{pmatrix} k\gamma\tau^2\varphi^2(\gamma\tau)e^{-(m-1)\gamma\tau} & k\gamma\tau\varphi(\gamma\tau)e^{-(m-1)\gamma\tau} \\ k\gamma\tau\varphi(\gamma\tau)e^{-m\gamma\tau} & k\gamma e^{-m\gamma\tau} \end{pmatrix}. \tag{A.16}$$

Hence, in particular,

$$G_{11}(m) = k\gamma\tau^2\varphi^2(\gamma\tau)e^{-(m-1)\gamma\tau}. \tag{A.17}$$

The likelihood $p(\Delta X_0, \cdots, \Delta X_{N_\tau - 1})$ is determined by (A.15) and (A.17).

A.2 Lemmas

Lemma 1. *If* $X \sim \mathcal{N}(\mu, \sigma^2)$ *and* $Y \mid X \sim \mathcal{N}(aX, \tau^2)$*, then* $Y \sim \mathcal{N}(a\mu, a^2\sigma^2 + \tau^2)$.

Proof. Consider the moment generating function of Y, $M_Y(t)$. Recall that for a Gaussian random variable such as X the moment generating function is

$$M_X(t) \coloneqq \mathbb{E}_X\left[e^{Xt}\right] = e^{\mu t + \sigma^2 t^2/2}; \tag{A.18}$$

similarly, since $aX \sim \mathcal{N}\left(a\mu, a^2\sigma^2\right)$,

$$M_{aX}(t) \coloneqq \mathbb{E}_X\left[e^{aXt}\right] = e^{a\mu t + a^2\sigma^2 t^2/2}. \tag{A.19}$$

Now,

$$\begin{aligned} M_Y(t) &= \mathbb{E}_Y\left[e^{Yt}\right] && \text{(A.20a)}\\ &= \mathbb{E}_X\left[\mathbb{E}_{Y|X}\left[e^{Yt}\right]\right] && \text{(A.20b)}\\ &= \mathbb{E}_X\left[e^{aXt + \tau^2 t^2/2}\right] && \text{(A.20c)}\\ &= e^{\tau^2 t^2/2}\mathbb{E}_X\left[e^{aXt}\right] = e^{(a\mu)t + (a^2\sigma^2 + \tau^2)t^2/2}, && \text{(A.20d)} \end{aligned}$$

which we can recognise as the moment generating function of a Gaussian random variable with mean $a\mu$ and variance $a^2\sigma^2 + \tau^2$. □

Bibliography

Abernathey, R. P. & Marshall, J. (2013), 'Global surface eddy diffusivities derived from satellite altimetry', *Journal of Geophysical Research: Oceans* **118**(2), 901–916.

Akaike, H. (1998), *Information Theory and an Extension of the Maximum Likelihood Principle*, Springer, pp. 199–213.

Arbic, B. K. & Flierl, G. R. (2003), 'Coherent vortices and kinetic energy ribbons in asymptotic, quasi two-dimensional f-plane turbulence', *Physics of fluids* **15**(8), 2177–2189.

Austin, T. (2015), 'Exchangeable random measures', *Annales de l'IHP Probabilités et statistiques* **51**(3), 842–861.

Balwada, D., Xie, J.-H., Marino, R. & Feraco, F. (2022), 'Direct observational evidence of an oceanic dual kinetic energy cascade and its seasonality', *Science Advances* **8**(41), eabq2566.

Batchelor, G. K. (1953), *The theory of homogeneous turbulence*, Cambridge University Press.

Berger, J. O. (1985), *Statistical decision theory and Bayesian analysis*, second edn, Springer.

Berloff, P. S. & McWilliams, J. C. (2002), 'Material transport in oceanic gyres. Part II: Hierarchy of stochastic models', *Journal of Physical Oceanography* **32**(3), 797–830.

Bernardo, J. M. (1979), 'Expected information as expected utility', *The Annals of Statistics* **7**(3), 686 – 690.

Bernardo, J. M. & Smith, A. F. M. (1994), *Bayesian theory*, Wiley.

Bishop, C. M. (1994), Mixture density networks, Technical Report NCRG/94/004, Aston University.

Bishop, C. M. (2006), *Pattern Recognition and Machine Learning*, Springer.

Bortolussi, L. & Palmieri, L. (2018), Deep Abstractions of Chemical Reaction Networks, *in* M. Češka & D. Šafránek, eds, 'Computational Methods in Systems Biology', Springer, pp. 21–38.

Bracco, A., LaCasce, J. H. & Provenzale, A. (2000), 'Velocity probability density functions for oceanic floats', *Journal of Physical Oceanography* **30**(3), 461 – 474.

Bröcker, J. & Smith, L. A. (2007), 'Scoring probabilistic forecasts: The importance of being proper', *Weather and Forecasting* **22**(2), 382–388.

Brolly, M. T. (2023), 'Inferring ocean transport statistics with probabilistic neural networks', *Journal of Advances in Modeling Earth Systems* **15**(6), e2023MS003718.

Brolly, M. T., Maddison, J. R., Teckentrup, A. L. & Vanneste, J. (2022), 'Bayesian comparison of stochastic models of dispersion', *Journal of Fluid Mechanics* **944**, A2.

Brooks, S., Gelman, A., Jones, G. & Meng, X.-L. (2011), *Handbook of Markov Chain Monte Carlo*, CRC Press.

Brooks, S. P. & Gelman, A. (1998), 'General methods for monitoring convergence of iterative simulations', *Journal of Computational and Graphical Statistics* **7**(4), 434–455.

Carrassi, A., Bocquet, M., Hannart, A. & Ghil, M. (2017), 'Estimating model evidence using data assimilation', *Quarterly Journal of the Royal Meteorological Society* **143**(703), 866–880.

Carson, J., Crucifix, M., Preston, S. & Wilkinson, R. D. (2018), 'Bayesian model selection for the glacial–interglacial cycle', *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **67**(1), 25–54.

Chen, T., Fox, E. & Guestrin, C. (2014), Stochastic Gradient Hamiltonian Monte Carlo, *in* 'International conference on machine learning', pp. 1683–1691.

Clare, M. C. A., Sonnewald, M., Lguensat, R., Deshayes, J. & Balaji, V. (2022), 'Explainable artificial intelligence for bayesian neural networks: Toward trustworthy predictions of ocean dynamics', *Journal of Advances in Modeling Earth Systems* **14**(11), e2022MS003162.

Clyde, M. & Iversen, E. S. (2013), Bayesian model averaging in the M-open framework, *in* 'Bayesian Theory and Applications', Oxford University Press.

Cotter, C. & Pavliotis, G. A. (2009), 'Estimating eddy diffusivities from noisy Lagrangian observations', *Communications in Mathematical Sciences* **7**(4), 805 – 838.

Cotter, S. L., Roberts, G. O., Stuart, A. M. & White, D. (2013), 'MCMC Methods for Functions: Modifying Old Algorithms to Make Them Faster', *Statistical Science* **28**(3), 424 – 446.

Cover, T. M. & Thomas, J. A. (2006), *Elements of Information Theory*, John Wiley and Sons, Inc.

Cui, S. & Datcu, M. (2015), Comparison of Kullback–Leibler divergence approximation methods between gaussian mixture models for satellite image retrieval, *in* '2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)', pp. 3719–3722.

Davis, C. N., Hollingsworth, T. D., Caudron, Q. & Irvine, M. A. (2020), 'The use of mixture density networks in the emulation of complex epidemiological individual-based models', *PLoS computational biology* **16**(3), e1006869.

Davis, R. E. (1987), 'Modeling eddy transport of passive tracers', *Journal of Marine Research* **45**(3), 635–666.

Davis, R. E. (1991), 'Observing the general circulation with floats', *Deep Sea Research Part A. Oceanographic Research Papers* **38**, S531–S571.

Daxberger, E., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M. & Hennig, P. (2021), Laplace redux - effortless Bayesian deep learning, *in* 'Advances in Neural Information Processing Systems', Vol. 34, pp. 20089–20103.

Du, H. (2021), 'Beyond strictly proper scoring rules: The importance of being local', *Weather and Forecasting* **36**(2), 457–468.

Durmus, A. & Moulines, É. (2019), 'High-dimensional Bayesian inference via the unadjusted Langevin algorithm', *Bernoulli* **25**(4A), 2854 – 2882.

Efron, B. (1979), 'Bootstrap Methods: Another Look at the Jackknife', *The Annals of Statistics* **7**(1), 1 – 26.

Esler, J. G. & Ramli, H. M. (2017), 'Shear dispersion in the turbulent atmospheric boundary layer', *Quarterly Journal of the Royal Meteorological Society* **143**(705), 1721–1733.

Flam-Shepherd, D., Requeima, J. & Duvenaud, D. (2017), 'Mapping Gaussian Process Priors to Bayesian Neural Networks', *NeurIPS workshop on Bayesian Deep Learning* .

Froyland, G. (2001), Extracting dynamical behavior via Markov models, *in* 'Nonlinear dynamics and statistics', Springer, pp. 281–321.

Froyland, G., Stuart, R. M. & van Sebille, E. (2014), 'How well-connected is the surface of the global ocean?', *Chaos: An Interdisciplinary Journal of Nonlinear Science* **24**(3), 033126.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A. & Rubin, D. B. (2013), *Bayesian Data Analysis*, CRC Press.

Gelman, A. & Rubin, D. B. (1992), 'Inference from iterative simulation using multiple sequences', *Statistical Science* **7**(4), 457–472.

Geman, S. & Geman, D. (1984), 'Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**, 721–741.

Gneiting, T. & Raftery, A. E. (2007), 'Strictly proper scoring rules, prediction, and estimation', *Journal of the American Statistical Association* **102**(477), 359–378.

Good, I. J. (1952), 'Rational decisions', *Journal of the Royal Statistical Society: Series B (Methodological)* **14**(1), 107–114.

Griesel, A., Gille, S. T., Sprintall, J., McClean, J. L., LaCasce, J. H. & Maltrud, M. E. (2010), 'Isopycnal diffusivities in the antarctic circumpolar current inferred from lagrangian floats in an eddying model', *Journal of Geophysical Research: Oceans* **115**(C6).

Griffa, A. (1996), *Applications of stochastic particle models to oceanographic problems*, Birkhäuser, pp. 113–140.

Guillaumin, A. P. & Zanna, L. (2021), 'Stochastic-Deep Learning Parameterization of Ocean Momentum Forcing', *Journal of Advances in Modeling Earth Systems* **13**(9), e2021MS002534.

Hairer, M., Stuart, A. M. & Vollmer, S. J. (2014), 'Spectral gaps for a Metropolis–Hastings algorithm in infinite dimensions', *The Annals of Applied Probability* **24**(6), 2455 – 2490.

Hannart, A., Carrassi, A., Bocquet, M., Ghil, M., Naveau, P., Pulido, M., Ruiz, J. & Tandeo, P. (2016), 'DADA: data assimilation for the detection and attribution of weather and climate-related events', *Climatic Change* **136**(2), 155–174.

Hansen, D. V. & Herman, A. (1989), 'Temporal Sampling Requirements for Surface Drifting Buoys in the Tropical Pacific', *Journal of Atmospheric and Oceanic Technology* **6**(4), 599–607.

Hansen, D. V. & Poulain, P.-M. (1996), 'Quality Control and Interpolations of WOCE-TOGA Drifter Data', *Journal of Atmospheric and Oceanic Technology* **13**(4), 900–909.

Hastings, W. K. (1970), 'Monte Carlo sampling methods using Markov chains and their applications', *Biometrika* **57**(1), 97–109.

Hoffman, M. D., Gelman, A. et al. (2014), 'The No-U-Turn Sampler: adaptively setting path lengths in Hamiltonian Monte Carlo', *J. Mach. Learn. Res.* **15**(1), 1593–1623.

Izmailov, P., Vikram, S., Hoffman, M. D. & Wilson, A. G. G. (2021), What are Bayesian neural network posteriors really like?, *in* 'International conference on machine learning', pp. 4629–4640.

Jaynes, E. T. (2003), *Probability Theory: The Logic of Science*, Cambridge University Press.

Jeffreys, H. (1983), *Theory of Probability*, Clarendon Press.

Jospin, L. V., Laga, H., Boussaid, F., Buntine, W. & Bennamoun, M. (2022), 'Hands-on bayesian neural networks—a tutorial for deep learning users', *IEEE Computational Intelligence Magazine* **17**(2), 29–48.

Kallenberg, O. (1997), *Foundations of Modern Probability*, Springer.

Kass, R. E. & Raftery, A. E. (1995), 'Bayes Factors', *Journal of the American Statistical Association* **90**(430), 773–795.

Kidger, P. & Lyons, T. (2020), Universal approximation with deep narrow networks, *in* 'Proceedings of Thirty Third Conference on Learning Theory', PMLR, pp. 2306–2327.

Kingma, D. P. & Ba, J. (2015), 'Adam: A Method for Stochastic Optimization', *arXiv:1412.6980* .

Kingma, D. P., Salimans, T. & Welling, M. (2015), 'Variational dropout and the local reparameterization trick', *Advances in neural information processing systems* **28**.

Kingma, D. P. & Welling, M. (2014), Auto-encoding variational bayes, *in* '2nd International Conference on Learning Representations'.

Klocker, A. & Abernathey, R. (2014), 'Global patterns of mesoscale eddy properties and diffusivities', *Journal of Physical Oceanography* **44**(3), 1030–1046.

Klocker, A., Ferrari, R., LaCasce, J. H. & Merrifield, S. T. (2012), 'Reconciling float-based and tracer-based estimates of lateral diffusivities', *Journal of Marine Research* **70**(4), 569–602.

Koszalka, I., LaCasce, J., Andersson, M., Orvik, K. & Mauritzen, C. (2011), 'Surface circulation in the Nordic seas from clustered drifters', *Deep Sea Research Part I: Oceanographic Research Papers* **58**(4), 468–485.

Kroese, D. P., Taimre, T. & Botev, Z. I. (2011), *Handbook of Monte Carlo Methods*, Wiley.

Krog, J. & Lomholt, M. A. (2017), 'Bayesian inference with information content model check for Langevin equations', *Physical Review E* **96**(6), 062106.

LaCasce, J. (2005), 'Eulerian and Lagrangian velocity distributions in the North Atlantic', *Journal of physical oceanography* **35**(12), 2327–2336.

LaCasce, J. (2008), 'Statistics from Lagrangian observations', *Progress in Oceanography* **77**(1), 1–29.

LaCasce, J. H., Ferrari, R., Marshall, J., Tulloch, R., Balwada, D. & Speer, K. (2014), 'Float-derived isopycnal diffusivities in the dimes experiment', *Journal of Physical Oceanography* **44**(2), 764 – 780.

LeCun, Y. A., Bottou, L., Orr, G. B. & Müller, K.-R. (2012), *Efficient BackProp*, Springer, pp. 9–48.

Leshno, M., Lin, V. Y., Pinkus, A. & Schocken, S. (1993), 'Multilayer feedforward networks with a nonpolynomial activation function can approximate any function', *Neural Networks* **6**(6), 861–867.

Lilly, J. M., Sykulski, A. M., Early, J. J. & Olhede, S. C. (2017), 'Fractional Brownian motion, the Matérn process, and stochastic modeling of turbulent dispersion', *Nonlinear Processes in Geophysics* **24**(3), 481–514.

Liu, Z., Hartwig, T. & Ueda, M. (2020), 'Neural networks fail to learn periodic functions and how to fix it', *arXiv:2006.08195* .

Lumpkin, R. & Centurioni, L. (2019), 'Dataset: Global Drifter Program quality-controlled 6-hour interpolated data from ocean surface drifting buoys'. Accessed on 12/04/2022.

Lumpkin, R., Maximenko, N. & Pazos, M. (2012), 'Evaluating where and why drifters die', *Journal of Atmospheric and Oceanic Technology* **29**(2), 300 – 308.

Luo, X., Nadiga, B. T., Park, J. H., Ren, Y., Xu, W. & Yoo, S. (2022), 'A bayesian deep learning approach to near-term climate prediction', *Journal of Advances in Modeling Earth Systems* **14**(10), e2022MS003058.

Lütkepohl, H. (2007), *New Introduction to Multiple Time Series Analysis*, Springer.

MacKay, D. J. C. (1995), 'Probable networks and plausible predictions-a review of practical bayesian methods for supervised neural networks', *Network: computation in neural systems* **6**(3), 469.

MacKay, D. J. C. (2003), *Information theory, inference, and learning algorithms*, Cambridge University Press.

Majda, A. J. & Kramer, P. R. (1999), 'Simplified models for turbulent diffusion: Theory, numerical modelling, and physical phenomena', *Physics Reports* **314**(4), 237–574.

Mann, R. P. (2011), 'Bayesian inference for identifying interaction rules in moving animal groups', *PLOS ONE* **6**(8), 1–10.

Mark, C., Metzner, C., Lautscham, L., Strissel, P. L., Strick, R. & Fabry, B. (2018), 'Bayesian model selection for complex dynamic systems', *Nature communications* **9**(1), 1–12.

Maulik, R., Fukami, K., Ramachandra, N., Fukagata, K. & Taira, K. (2020), 'Probabilistic neural networks for fluid flow surrogate modeling and data recovery', *Physical Review Fluids* **5**(10), 104401.

Maximenko, N., Hafner, J. & Niiler, P. (2012), 'Pathways of marine debris derived from trajectories of Lagrangian drifters', *Marine Pollution Bulletin* **65**(1), 51–62.

McAdam, R. & van Sebille, E. (2018), 'Surface connectivity and interocean exchanges from drifter-based transition matrices', *Journal of Geophysical Research: Oceans* **123**(1), 514–532.

Metref, S., Hannart, A., Ruiz, J., Bocquet, M., Carrassi, A. & Ghil, M. (2019), 'Estimating model evidence using ensemble-based data assimilation with localization – the model selection problem', *Quarterly Journal of the Royal Meteorological Society* **145**(721), 1571–1588.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. (1953), 'Equation of state calculations by fast computing machines', *The Journal of Chemical Physics* **21**(6), 1087–1092.

Min, S.-K., Simonis, D. & Hense, A. (2007), 'Probabilistic climate change predictions applying Bayesian model averaging', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **365**(1857), 2103–2116.

Miron, P., Beron-Vera, F. J., Helfmann, L. & Koltai, P. (2021), 'Transition paths of marine debris and the stability of the garbage patches', *Chaos: An Interdisciplinary Journal of Nonlinear Science* **31**(3), 033101.

Miron, P., Beron-Vera, F. J., Olascoaga, M. J., Sheinbaum, J., Pérez-Brunius, P. & Froyland, G. (2017), 'Lagrangian dynamical geography of the Gulf of Mexico', *Nature Scientific Reports* **7**(1), 1–12.

Monin, A. S. & Yaglom, A. M. (1971), *Statistical Fluid Mechanics: Mechanics of Turbulence*, Vol. 1-2, MIT Press.

Moore, C., Moore, S., Leecaster, M. & Weisberg, S. (2001), 'A comparison of plastic and plankton in the north pacific central gyre', *Marine Pollution Bulletin* **42**(12), 1297–1300.

Neal, R. M. (1996), *Bayesian learning for neural networks*, Springer.

Nelder, J. A. & Mead, R. (1965), 'A simplex method for function minimization', *The computer journal* **7**(4), 308–313.

Nesterov, Y. (2009), 'Primal-dual subgradient methods for convex problems', *Mathematical programming* **120**(1), 221–259.

Oh, I. S., Zhurbas, V. & Park, W. (2000), 'Estimating horizontal diffusivity in the east sea (sea of japan) and the northwest pacific from satellite-tracked drifter data', *Journal of Geophysical Research: Oceans* **105**(C3), 6483–6492.

Onink, V., van Sebille, E. & Laufkötter, C. (2022), 'Empirical Lagrangian parametrization for wind-driven mixing of buoyant particles at the ocean surface', *Geoscientific Model Development* **15**(5), 1995–2012.

O'Malley, M., Sykulski, A. M., Laso-Jadart, R. & Madoui, M.-A. (2021), 'Estimating the travel time and the most likely path from Lagrangian drifters', *Journal of Atmospheric and Oceanic Technology* **38**(5), 1059 – 1073.

Paisley, J., Blei, D. M. & Jordan, M. I. (2012), Variational bayesian inference with stochastic search, *in* 'Proceedings of the 29th International Coference on International Conference on Machine Learning', p. 1363–1370.

Papamarkou, T., Hinkle, J., Young, M. T. & Womble, D. (2022), 'Challenges in Markov Chain Monte Carlo for Bayesian Neural Networks', *Statistical Science* **37**(3), 425–442.

Pasquero, C., Provenzale, A. & Babiano, A. (2001), 'Parameterization of dispersion in two-dimensional turbulence', *Journal of Fluid Mechanics* **439**, 279–303.

Pavliotis, G. A. (2014), *Stochastic processes and applications: diffusion processes, the Fokker–Planck and Langevin equations*, Springer.

Prechelt, L. (2012), *Early Stopping — But When?*, Springer, pp. 53–67.

Provenzale, A., Babiano, A. & Villone, B. (1995), 'Single-particle trajectories in two-dimensional turbulence', *Chaos, solitons and fractals* **5**(10), 2055–2071.

Reich, S. & Cotter, C. (2015), *Probabilistic Forecasting and Bayesian Data Assimilation*, Cambridge University Press.

Reijnders, D., Deleersnijder, E. & van Sebille, E. (2022), 'Simulating Lagrangian Subgrid-Scale Dispersion on Neutral Surfaces in the Ocean', *Journal of Advances in Modeling Earth Systems* **14**(2), e2021MS002850.

Rezende, D. J., Mohamed, S. & Wierstra, D. (2014), Stochastic backpropagation and approximate inference in deep generative models, *in* 'International conference on machine learning', pp. 1278–1286.

Roach, C. J., Balwada, D. & Speer, K. (2018), 'Global observations of horizontal mixing from argo float and surface drifter trajectories', *Journal of Geophysical Research: Oceans* **123**(7), 4560–4575.

Robert, C. P. (2007), *The Bayesian choice: From Decision-Theoretic Foundations to Computational Implementation*, second edn, Springer.

Robert, C. P. (2015), *The Metropolis–Hastings Algorithm*, Wiley, pp. 1–15.

Robert, C. P. & Casella., G. (2004), *Monte Carlo statistical methods*, second edn, Springer.

Roberts, G. O. & Tweedie, R. L. (1996), 'Exponential convergence of Langevin distributions and their discrete approximations', *Bernoulli* **2**(4), 341–363.

Rodean, H. C. (1996), *Stochastic Lagrangian models of turbulent diffusion*, Springer.

Roeder, G., Wu, Y. & Duvenaud, D. K. (2017), 'Sticking the landing: Simple, lower-variance gradient estimators for variational inference', *Advances in Neural Information Processing Systems* **30**.

Roulston, M. S. & Smith, L. A. (2002), 'Evaluating probabilistic forecasts using information theory', *Monthly Weather Review* **130**(6), 1653–1660.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1985), Learning internal representations by error propagation, Technical report, UCSD Institute for Cognitive Science.

Rupolo, V. (2007), 'A Lagrangian-Based Approach for Determining Trajectories Taxonomy and Turbulence Regimes', *Journal of Physical Oceanography* **37**(6), 1584 – 1609.

Ryan, P. G., Moore, C. J., Van Franeker, J. A. & Moloney, C. L. (2009), 'Monitoring the abundance of plastic debris in the marine environment', *Philosophical Transactions of the Royal Society B: Biological Sciences* **364**(1526), 1999–2012.

Schwarz, G. (1978), 'Estimating the dimension of a model', *The Annals of Statistics* **6**(2), 461–464.

Scott, R. K. (2007), 'Nonrobustness of the two-dimensional turbulent inverse cascade', *Physical Review E* **75**, 046301.

Sharma, M., Farquhar, S., Nalisnick, E. & Rainforth, T. (2023), Do Bayesian Neural Networks Need To Be Fully Stochastic?, *in* 'International Conference on Artificial Intelligence and Statistics', pp. 7694–7722.

Shin, J., Ge, Y., Lampmann, A. & Pfitzner, M. (2021), 'A data-driven subgrid scale model in Large Eddy Simulation of turbulent premixed combustion', *Combustion and Flame* **231**, 111486.

Skilling, J. (2004), 'Nested Sampling', *AIP Conference Proceedings* **735**(1), 395–405.

Sykulski, A. M., Olhede, S. C., Lilly, J. M. & Danioux, E. (2016), 'Lagrangian time series models for ocean surface drifter trajectories', *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **65**(1), 29–50.

Sykulski, A. M., Olhede, S. C., Lilly, J. M. & Early, J. J. (2017), 'Frequency-domain stochastic modeling of stationary bivariate or complex-valued signals', *IEEE Transactions on Signal Processing* **65**(12), 3136–3151.

Tabak, E. G. & Turner, C. V. (2013), 'A family of nonparametric density estimation algorithms', *Communications on Pure and Applied Mathematics* **66**(2), 145–164.

Tabak, E. G. & Vanden-Eijnden, E. (2010), 'Density estimation by dual ascent of the log-likelihood', *Communications in Mathematical Sciences* **8**(1), 217 – 233.

Taylor, G. I. (1922), 'Diffusion by continuous movements', *Proceedings of the London Mathematical Society* **s2-20**(1), 196–212.

TensorFlow Developers (2021), 'Tensorflow'. v2.7.0, doi.

TensorFlow Probability Developers (2021), 'Tensorflow probability'. v0.15.0, GitHub Release.

Thapa, S., Lomholt, M. A., Krog, J., Cherstvy, A. G. & Metzler, R. (2018), 'Bayesian analysis of single-particle tracking data using the nested-sampling algorithm: maximum-likelihood model selection applied to stochastic-diffusivity data', *Physical Chemistry Chemical Physics* **20**(46), 29018–29037.

Thomson, D. J. (1987), 'Criteria for the selection of stochastic models of particle trajectories in turbulent flows', *Journal of Fluid Mechanics* **180**, 529–556.

Tran, B.-H., Rossi, S., Milios, D. & Filippone, M. (2022), 'All You Need is a Good Functional Prior for Bayesian Deep Learning', *Journal of Machine Learning Research* **23**(74), 1–56.

Uhlenbeck, G. E. & Ornstein, L. S. (1930), 'On the theory of the Brownian motion', *Physical Review* **36**, 823–841.

Ulam, S. M. (1960), *A collection of mathematical problems*, Interscience Publishers.

Vaart, A. W. v. d. (1998), *Asymptotic Statistics*, Cambridge University Press.

Vallis, G. K. (2017), *Atmospheric and Oceanic Fluid Dynamics: Fundamentals and Large-Scale Circulation*, 2 edn, Cambridge University Press.

van Sebille, E., England, M. H. & Froyland, G. (2012), 'Origin, dynamics and evolution of ocean garbage patches from observed surface drifters', *Environmental Research Letters* **7**(4), 044040.

van Sebille, E. et al. (2018), 'Lagrangian ocean analysis: Fundamentals and practices', *Ocean Modelling* **121**, 49–75.

Vats, D. & Knudson, C. (2021), 'Revisiting the Gelman–Rubin Diagnostic', *Statistical Science* **36**(4), 518–529.

Villani, C. (2009), *Optimal Transport: Old and New*, Springer.

Vuong, Q. H. (1989), 'Likelihood ratio tests for model selection and non-nested hypotheses', *Econometrica* **57**(2), 307–333.

Wen, Y., Vicol, P., Ba, J., Tran, D. & Grosse, R. (2018), Flipout: Efficient Pseudo-Independent Weight Perturbations on Mini-Batches, *in* 'International Conference on Learning Representations'.

Wenzel, F., Roth, K., Veeling, B. S., Świątkowski, J., Tran, L., Mandt, S., Snoek, J., Salimans, T., Jenatton, R. & Nowozin, S. (2020), How good is the Bayes posterior in deep neural networks really?, *in* 'Proceedings of the 37th International Conference on Machine Learning', pp. 10248–10259.

Williams, C. K. & Rasmussen, C. E. (2006), *Gaussian processes for machine learning*, MIT press.

Wong, A. P., Wijffels, S. E., Riser, S. C., Pouliquen, S., Hosoda, S., Roemmich, D., Gilson, J., Johnson, G. C., Martini, K., Murphy, D. J. et al. (2020), 'Argo data 1999–2019: Two million temperature-salinity profiles and subsurface velocity observations from a global array of profiling floats', *Frontiers in Marine Science* **7**, 700.

Ying, Y. K., Maddison, J. R. & Vanneste, J. (2019), 'Bayesian inference of ocean diffusivity from Lagrangian trajectory data', *Ocean Modelling* **140**, 101401.

Zhu, C., Byrd, R. H., Lu, P. & Nocedal, J. (1997), 'Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization', *ACM Transactions on mathematical software (TOMS)* **23**(4), 550–560.

Zhurbas, V., Lyzhkov, D. & Kuzmina, N. (2014), 'Drifter-derived estimates of lateral eddy diffusivity in the world ocean with emphasis on the indian ocean and problems of parameterisation', *Deep Sea Research Part I: Oceanographic Research Papers* **83**, 1–11.

Zhurbas, V. & Oh, I. S. (2003), 'Lateral diffusivity and Lagrangian scales in the Pacific Ocean as derived from drifter data', *Journal of Geophysical Research: Oceans* **108**(C5).

Zhurbas, V. & Oh, I. S. (2004), 'Drifter-derived maps of lateral diffusivity in the Pacific and Atlantic Oceans in relation to surface circulation patterns', *Journal of Geophysical Research: Oceans* **109**(C5).