Accelerating Bayesian inference with physics-governed likelihoods using deep learning

(Alternative Format Thesis)

Deveney, Teo

Award date:
2022

Awarding institution:
University of Bath

Link to publication

Alternative formats
If you require this document in an alternative format, please contact: openaccess@bath.ac.uk

Copyright of this thesis rests with the author. Access is subject to the above licence, if given. If no licence is specified above, original content in this thesis is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC-ND 4.0) Licence (https://creativecommons.org/licenses/by-nc-nd/4.0/). Any third-party copyright material present remains the property of its respective owner(s) and is licensed under its existing terms.

Take down policy
If you consider content within Bath’s Research Portal to be in breach of UK law, please contact: openaccess@bath.ac.uk with the details. Your claim will be investigated and, where appropriate, the item will be removed from public view as soon as possible.
Accelerating Bayesian inference with physics-governed likelihoods using deep learning

submitted by

Teo James Deveney

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Mathematical Sciences

June 2022
COPYRIGHT NOTICE

Attention is drawn to the fact that copyright of this thesis rests with the author and copyright of any previously published materials included may rest with third parties. A copy of this thesis has been supplied on condition that anyone who consults it understands that they must not copy it or use material from it except as licensed, permitted by law or with the consent of the author or other copyright owners, as applicable.

DECLARATION OF ANY PREVIOUS SUBMISSION OF THE WORK

The material presented here for examination for the award of a higher degree by research has not been incorporated into a submission for another degree.

Signature of Author .................................................................

Teo James Deveney

DECLARATION OF AUTHORSHIP

I am the author of this thesis, and the work described therein was carried out by myself personally in collaboration with my supervisors.

Chapter 2 is reproduced from a submitted manuscript.

Chapter 3 is reproduced from one submitted manuscript and one published manuscript.

Chapter 4 is reproduced from a draft manuscript.

Signature of Author .................................................................

Teo James Deveney
ACKNOWLEDGEMENTS

I thank my supervisors, Tony and Eike, for providing the opportunity to carry out research in this area, and for their guidance and patience as I explored the various avenues that resulted in this research. Thank you also to my collaborator Hui, whose input formed a valuable part of this work.

To the friends I made studying towards this thesis. The times that we shared enriched my time in Bath and gave me experiences that I will treasure. Thank you for all the memories, support, and companionship.

Above all, thank you to my Mum and Dad, my sister Alex, and my Grandma, Grandad, and Nana for the continued love and support they’ve shown throughout my entire life.
Abstract

This thesis compiles three papers at various stages of publication consisting of novel algorithms and research on the topic of deep learning accelerated physics based Bayesian inference. Additionally included is an application paper written in collaboration with colleagues in the Department of Engineering where some methodology developed in this thesis is applied to heat transfer data from an experimental compressor cavity rig.

Before presenting the first of these papers, we begin in Chapter 1 with an introduction and overview of the themes of the thesis. This overview includes a description of Bayesian inference and its solution methods, an outline of traditional solution methods for PDEs and integral equations, a brief introduction to neural networks and their training, and some established approaches applied to solve PDE-based Bayesian inverse problems. Much of the material covered here will be revisited where appropriate in subsequent chapters, and so Chapter 1 is not intended to be overly comprehensive. Instead this chapter serves as a gentle introduction to these topics, providing both a natural framework through which the subsequent chapters can be viewed, as well as a reference of more established approaches for PDE-based Bayesian inversion to which our deep learning accelerations can be compared. The remaining chapters follow organically from this starting point. The statistical models considered in each chapter are ordered in increasing levels of complexity, and the contents focus on how deep learning surrogates can accelerate Bayesian inference in these varied settings. Numerical examples are plentiful throughout and Python implementations are publicly available where indicated.

Chapter 2 of this report contains a paper [38] which is currently under review. In this chapter we describe a deep learning approach to efficiently perform Bayesian inference in partial differential equation (PDE) and integral equation models over potentially high-dimensional parameter spaces. We review some deep learning approaches to approximate the solutions to PDEs, and introduce a new neural network approach to approximating the solutions of Fredholm and Volterra integral equations of the first and second kind. These algorithms work by formulating appropriate loss functions such that the solutions of these equations are the minimisers of the corresponding optimisation problems. We then extend these algorithms to approximate parametric “surrogate” solutions of PDEs and integral equations. This deep learning approach allows the efficient approximation of parametric solutions in significantly higher dimensions than is possible using classical techniques. Since the approximated solutions are very cheap
to evaluate, the solutions of Bayesian inverse problems become tractable using Markov chain Monte Carlo. Our method is applied to two real-world examples; these include Bayesian inference of PDE and integral equation parameters in an electrochemical example, and Bayesian inference of a heat-transfer function with applications in aviation.

Chapter 3 comprises two papers born from a collaboration with the Department of Mechanical Engineering. The main part of this chapter presents a paper that significantly extends the techniques introduced previously, in order to solve a spatio-temporal extension of the heat-transfer problem introduced in Chapter 2. In this highly ill-posed inverse problem we seek to understand the heat flux behaviour in rapidly rotating discs within engine cavities, the results of which will be used in practice with experimental data to inform future engine design. Due to this demanding application, the developments in this paper focus on ensuring that our inference is efficient and reliable despite the black-box nature of deep learning. We do this by constructing appropriate Gaussian process priors, developing a training procedure that efficiently achieves more accurate approximations by adapting to the desired posterior distribution, using delayed-acceptance sampling to ensure the accuracy of the posteriors we achieve are mathematically guaranteed, and using surrogate accelerated Hamiltonian proposals to achieve these samples within a reasonable time-frame. The method is then applied in a simulation study to verify its effectiveness, before illustrating the results using real data from a multi-cavity compressor rig. Additionally included as Subchapter 3A, is an application paper written in conjunction with colleagues in the Department of Mechanical Engineering. This is a more detailed investigation of the experimental setting and how the inferences resulting from our method will affect engine design in the future.

In Chapter 4 we present a manuscript in preparation, containing some preliminary results that explore a more flexible application of deep learning surrogates. Here, neural networks are used as physical features in a spatio-temporal statistical model of pollution levels. We are motivated by the situation in Ulaanbaatar, Mongolia, which has dangerous pollution levels that fluctuate according to daily, weekly, and annual patterns. Despite this, the number of pollution sensors in the city is low, leading to large uncertainties dominating our inference if modern spatio-temporal statistical models are fit (e.g. regression with an autoregressive-Matérn Gaussian field). To overcome this, we use physical knowledge — represented by a deep learning surrogate of advection-diffusion dynamics with uncertain parameterisation — as the mean in a hierarchical spatio-temporal Gaussian process model. This approach heavily weights the prior of the model towards advection-diffusion dynamics, and its inclusion within a more flex-
ible Gaussian process model allows any behaviour that is not fully described by the physical equation to be resolved. In this chapter we perform an exploratory analysis of the atmospheric data from Ulaanbaatar, describe the statistical and physical models that we base our analysis on, and then introduce a physics-informed statistical model that incorporates aspects of both. Using the deep learning surrogate to represent the PDE, we fit all parameters (both PDE and Gaussian process parameters) within this model simultaneously, by sampling from their joint posterior distribution using Hamiltonian Monte Carlo. Finally we marginalise these parameters to form a predictive posterior field over the city that fully accounts the physical and statistical uncertainty and their interdependencies. This chapter then concludes with some notes on possible extensions to this topic, including the description of a mass-controlled neural network which allows us to systematically control the integral of our surrogate approximation, as well as several ways that the physical and statistical model could be developed in future.
3.2 Problem specification

3.3 Methodology

3.3.1 Prior approximation

3.3.2 Deep surrogate approximation

3.3.3 Approximating the posterior using delayed-acceptance HMC

3.4 Numerical results

3.4.1 Implementation details

3.4.2 Simulation study

3.4.3 Experimental data study

3.5 Closing remarks

Subchapter 3A

Use of Bayesian statistics to calculate transient heat fluxes on compressor discs

3A.1 Introduction

3A.2 Buoyancy-induced and stratified flow in compressor cavities

3A.3 Bath compressor cavity rig

3A.4 Modelling of transient heat transfer in compressor discs

3A.5 Bayesian approach

3A.5.1 Likelihood function

3A.5.2 Prior distribution

3A.5.3 Posterior distribution

3A.5.4 Solution technique

3A.6 Calculation of transient heat fluxes on compressor discs

3A.6.1 Application of Bayesian method to simulated temperature data

3A.6.2 Transient heat fluxes in a closed rotating cavity

3A.7 Conclusions and future work
Chapter 4

PDE informed statistical modelling of spatio-temporal pollution

4.1 Introduction .......................................................... 160
4.2 Ulaanbaatar setting ................................................. 162
4.3 Statistical pollution models ........................................ 167
   4.3.1 Spatial models ............................................... 167
   4.3.2 Spatio-temporal models .................................... 170
   4.3.3 Bayesian inverse problem approach ....................... 174
4.4 Physical pollution models ......................................... 176
   4.4.1 Solving the physical model with deep learning .......... 180
4.5 Physics-informed statistical approach ......................... 181
   4.5.1 Modelling assumptions and methodology ................. 182
      4.5.1.1 Statistical model .................................... 182
      4.5.1.2 PDE assumptions .................................. 183
   4.5.2 Simulated data ............................................... 185
   4.5.3 Ulaanbaatar data ............................................ 189
4.6 Conclusions ....................................................... 192
4.7 Future research opportunities ................................. 193
   4.7.1 Mass controlled neural network .......................... 193
   4.7.2 More complex modelling assumptions .................... 196
4.8 Appendix .......................................................... 198

Chapter 5

Conclusion .............................................................. 201
5.1 Summary ............................................................ 201
5.2 Closing remarks .................................................. 203
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>True underlying Gaussian curve plotted in dashed line, and noisy data samples plotted as points.</td>
<td>25</td>
</tr>
<tr>
<td>1-2</td>
<td>Bayesian and maximum likelihood estimates of the underlying Gaussian curve using the data.</td>
<td>27</td>
</tr>
<tr>
<td>1-3</td>
<td>Left: 5 point stencil used in finite differences. Right: Triangular elements of a finite element discretisation.</td>
<td>29</td>
</tr>
<tr>
<td>1-4</td>
<td>Architecture of a fully-connected feedforward neural network mapping a 3-dimensional input to a 1-dimensional output through a pair of 5-dimensional hidden layers.</td>
<td>33</td>
</tr>
<tr>
<td>2-1</td>
<td>Comparison of neural network approximations and midpoint quadrature for the voltammetry example in Section 2.6.1. The top row shows the currents $I(t; E^0)$ approximated by each method for different values of $E^0$. The error in the bottom row is the difference between the neural networks and the reference solution obtained by solving the integral equation with midpoint quadrature.</td>
<td>77</td>
</tr>
<tr>
<td>2-2</td>
<td>Top: Simulated data and fitted solutions using the mean of the posterior. The true standard deviations $\sigma$ and oxidation potential $E^0$ used to generate the Gaussian noise are given at the top of each column. Bottom: Approximate posterior distributions and 95% credible intervals computed using the PDE deep surrogate model. The true values of $E^0$ used to generate the data are marked by vertical lines.</td>
<td>78</td>
</tr>
<tr>
<td>2-3</td>
<td>Top: Simulated data and fitted solutions using the posterior mean. The true oxidation potential $E^0$ and standard deviations $\sigma$ used to generate the Gaussian noise are given at the top of each column. Bottom: Posterior distributions and 95% credible intervals computed using the integral equation deep surrogate. The true value of $E^0$ used to generate the data are marked by vertical lines.</td>
<td>79</td>
</tr>
</tbody>
</table>
2-4 Left: Fitted heat profile using the inferred Biot number. Right: Inferred $\tilde{Bi}(x; \theta)$ and 95% credible interval compared to the true Biot number. 82

2-5 Right: $\tilde{Bi}(x; \theta)$ inferred by minimising (2.6.18) compared to true Biot number. Left: Fitted heat profile using the inferred Biot number. 83

3-1 Illustration of the adaptive training scheme. At iteration $n$ the surrogate is trained locally by minimising the loss (3.3.10) with local measure $\pi^\alpha = \pi^\alpha_n$ as depicted by the grey measures. The gradient $\nabla_\alpha \tilde{u}$ of the locally trained surrogate is then used to update to $\alpha_{n+1}$ by gradient ascent. Upon convergence to the MAP estimate $\alpha^*$, the surrogate is trained over the Laplace approximation to the posterior shown in red (by minimising (3.3.10) with $\pi^\alpha = \pi^\alpha_{\text{Laplace}}$). The surrogate is then ready to begin sampling. 101

3-2 Left: Chebyshev approximation to $K_x(x, x')$ and $K_t(t, t')$. Right: Error between the true and approximated kernel. 110

3-3 Top: A random instance of $Bi(t, x)$. Bottom: Corresponding PDE solution and data locations. 112

3-4 Top: An example of a $Bi(t, x)$ that is physically unrealistic but achieves a good fit to the data shown in Figure 3-3. Middle: At discrete times the spatial profiles of this $Bi(t, x)$ are shown in blue. The profiles of the true $Bi(t, x)$ are plotted in red. The green curves are alternate examples of $Bi(t, x)$ that achieve a similar fit. Bottom: Temporal residuals between the data and solution fitted using this $Bi(t, x)$, demonstrating that the unrealistic $Bi(t, x)$ produces a good fit. 113

3-5 Laplace estimate of Biot number compared to the true value. 114

3-6 Spatial profiles of the 95% credible intervals of surrogate-only adaptive HMC, MALA, and RWMH at various time points. The red curves are the spatial profiles of the true Biot number used to simulate the data. 117

3-7 Spatial profiles of the 95% credible intervals of surrogate-only adaptive HMC, delayed-acceptance adaptive HMC, surrogate-only general HMC, delayed-acceptance general HMC. The red curves are the spatial profiles of the true Biot number used to simulate the data. 118

3-8 Comparison of trace plots of delayed-acceptance HMC, MALA, and RWMH and their 50 step moving average. The lengths of these chains is chosen so that the time taken to achieve each sample is the same. 119

3-9 Comparison of sample densities of different schemes for the Chebyshev coefficient that is cubic in space and constant in time. 119

14
3-10 Boundary and initial conditions for the experimental data computed using cubic spline regression. .................................................. 120

3-11 The inferred Biot number and 95% credible intervals at six time points. 121

3-12 Experimental data at various time points overlaid on the PDE solution using the inferred Bi(t, x). The subplots below each panel are the residuals for the corresponding plot. .............................................. 122

3A-1 A cross-section through a high pressure aero-engine compressor. (Owen, J. M., Tang, H., and Lock, G. D., Aerospace, 5(1), 32, 2018; licensed under a Creative Commons Attribution (CC BY) license.) ......... 131

3A-2 Steady-state disc temperatures and heat fluxes at Reφ = 7.0 × 10^5 (data from [75]). (a) Non-dimensional disc temperatures. (b) Heat fluxes. ................................................................. 135

3A-3 Simplified representation of buoyancy-induced flow inside a closed rotating cavity, showing the r-φ plane. ...................................................... 136

3A-4 Diagram of the compressor cavity rig at Bath. ...................... 136

3A-5 Instrumented closed cavity showing the locations of thermocouples. 137

3A-6 True heat flux distribution ...................................................... 143

3A-7 Initial and boundary conditions for simulated data. (a) Boundary conditions at x = 1 and xa = 0.5. (b) Initial condition at Fo= 0. ... 144

3A-8 Simulated disc temperature data and calculated temperatures from the Bayesian model. Symbols: simulated data; solid lines: calculated temperatures from the Bayesian model. (a) Temperature data histories at different radial locations. (b) Radial data distributions at different temporal locations. .............................................. 145

3A-9 Calculated heat fluxes (φ(Fo,x)) from simulated data using the Bayesian model. (a) Calculated heat fluxes using the simulated data. (b) Standard deviation of the calculated heat fluxes. (c) Difference between the calculated and true heat flux distributions. ......................... 146

3A-10 Transient temperatures and non-dimensional parameters for Reφ = 1.4 × 10^6. Symbols: experimental data; red solid lines: calculated temperatures from the Bayesian model. (a) Temperature data histories at different radial locations. (b) Radial data distributions at different temporal locations. (c) Non-dimensional parameters. ....... 148
Transient temperatures and non-dimensional parameters for \( \text{Re}_\phi = 0.7, 2.1 \) and \( 2.7 \times 10^6 \) (left, middle, and right, respectively). Symbols: experimental data; red solid lines: calculated temperatures from the Bayesian model. (a) Transient temperatures at \( \text{Re}_\phi = 0.7 \times 10^6 \). (b) Transient temperatures at \( \text{Re}_\phi = 2.1 \times 10^6 \). (c) Transient temperatures at \( \text{Re}_\phi = 2.7 \times 10^6 \). (d) Non-dimensional parameters at \( \text{Re}_\phi = 0.7 \times 10^6 \). (e) Non-dimensional parameters at \( \text{Re}_\phi = 2.1 \times 10^6 \). (f) Non-dimensional parameters at \( \text{Re}_\phi = 2.7 \times 10^6 \).

Calculated heat fluxes and standard deviations from experimental data at \( \text{Re}_\phi = 1.4 \times 10^6 \) using the Bayesian model. (a) Heat flux. (b) Standard deviation.

Calculated heat fluxes (curves) at different Fo and 95% confidence intervals (shaded regions) for four \( \text{Re}_\phi \) numbers. (a) \( \text{Re}_\phi = 1.4 \times 10^6 \). (b) \( \text{Re}_\phi = 1.4 \times 10^6 \). (c) \( \text{Re}_\phi = 1.4 \times 10^6 \). (d) \( \text{Re}_\phi = 1.4 \times 10^6 \).

Locations of air pollution monitoring stations in Ulaanbaatar....

Daily average PM10 levels (in \( \mu g/m^3 \)) and temperatures (°C) in 2018.

Weekly trends in PM10 data (\( \mu g/m^3 \)).

Daily trends in PM10 data (\( \mu g/m^3 \)).

Weekly trends in PM10 data (\( \mu g/m^3 \)).

Hourly posterior mean (in \( \mu g/m^3 \)) of the autoregressive Matérn Gaussian field applied to the raw Ulaanbaatar data for November 2018.

Spatial uncertainty field (\( \mu g/m^3 \)) showing the width of the 95% credible interval of the posterior autoregressive Matérn Gaussian field applied to the raw Ulaanbaatar data for November 2018.

Simulated data spatio-temporal predictions. Left: Posterior mean of \( \hat{u}(t, x; \theta^u) \). Centre: Posterior mean of \( \eta(t, x; \theta^\eta) \). Right: Posterior hybrid pollution prediction (in \( \mu g/m^3 \)).

95% credible interval widths associated with simulated data study (\( \mu g/m^3 \)). Left: Hybrid model uncertainty. Centre: Physical model uncertainty. Right: Provided for comparison, the uncertainty of a GP spatial post-processing approach.

Real data spatio-temporal predictions. Left: Posterior mean of \( \hat{u}(t, x; \theta^u) \). Centre: Posterior mean of \( \eta(t, x; \theta^\eta) \). Right: Posterior hybrid pollution prediction (in \( \mu g/m^3 \)).

95% credible interval uncertainties associated with the real data (\( \mu g/m^3 \)).
4-12 Full 24 hour predictions from the hybrid model, corresponding to Figure 4-10 .................................................. 198

4-13 Hourly predictions from the hybrid model using data taken from days when the daily average temperature was between $-5^\circ C$ and $0^\circ C$ ... 199
Chapter 1

Introduction and preliminaries

Statistical machine learning methodologies have seen significant development in recent years. Spurred on by the advent of faster computation and efficient optimisation algorithms, such methods have achieved state of the art results in many tasks where an abundance of data is available. In a perfect world, we would have unlimited quantities of noise-free data for every problem; such an ideal scenario would mean the key to solving all of our problems would lie entirely in analysing this data. Unfortunately the collection of data is not always trivial, and there still exist many applications where the available data alone is insufficient to solve the problem. Dealing with such problems is a popular area of research; the general consensus being that the incorporation of prior knowledge is the best way to regularise the analysis of such data. Prior knowledge can come in several forms; be it an awareness of the underlying physics, or an understanding of the system parameters gained from previous observations. Bayesian statistical models with physics-governed likelihoods provide an elegant means of incorporating these sources of prior information, and the efficient computation of their solutions using deep learning is the focus of this thesis.

The Bayesian approach to statistical inference allows us to use data to perform full uncertainty quantification on any feature (parameters, functions, hypotheses, etc.) of a statistical model [52]. In a Bayesian setting, practitioners can incorporate prior information about the model features into their inference, and fully analyse the posterior distribution of any quantity of interest by using the prior to condition this quantity on available data. Historically this approach was viewed unfavourably by many, due in part to a lack of computational resources to perform Bayesian calculations, but also due to philosophical beliefs that the features of a statistical model should be fixed and,
only the data uncertain. Despite this, the past few decades have seen Bayesian methods
grow vastly in popularity, and algorithms such as Markov chain Monte Carlo (MCMC)
have made Bayesian approaches tractable in many settings. Today Bayesian methods
are applied to highly complex models, with applications extending far beyond statistics
into fields such as finance [121], engineering [163] and social science [57].

In this thesis, we investigate how deep learning surrogates that emulate physical phe-
nomena can be integrated into a physics-governed Bayesian framework in order to
accelerate posterior computations. We consider the issue of using MCMC for Bayesian
parameter inference and spatio-temporal prediction in settings where physical knowl-
edge is encoded as either a partial differential equation (PDE), or integral equation. In
particular, we focus on the case where the statistical model has a likelihood function
that is dependent on the solution of one of these equations. This is indeed the case
when we want to infer model parameters using data, as in a traditional Bayesian in-
verse problem [147], however more general predictive contexts will also be explored by
combining physical models with statistical terms.

Currently, the main challenge with applying Bayesian methods in this setting is that
the cost of repeated likelihood computations can be high. This is because the likelihood
typically depends on an evaluation of the physical model, and so numerical solution
algorithms are needed [81]. In many cases these algorithms are expensive to execute,
and this bottleneck often renders vanilla MCMC intractable if applied with established
solvers such as finite differences. The situation is reminiscent of that several decades
ago for more traditional statistical models. Then, a lack of computing power was the
major hurdle to the implementation of Bayesian methodology, even for simple paramet-
ric statistical models that lack physical terms. Today, most inverse problems involving
expensive numerical solvers are treated how statistical models were treated then, that
is as optimisation problems over a parameter set. These methods can estimate pa-
rameters more quickly than MCMC sampling, but provide little, if any, measure of
uncertainty for these estimates. Bayesian posterior approximation techniques such as
variational Bayes are sometimes used if uncertainty quantification is desired [18, 102].
These are also framed as optimisation problems, this time over a predefined class of
probability distributions, and can form a tractable middle ground in many cases. How-
ever in general the true posterior will not lie within this predefined class, meaning the
reliability of these methods is debatable.

The approach that we take in this thesis is to use deep learning to construct sur-
rogate models to represent the solutions to physical models. These surrogates can be trained to approximate the solutions to a wide range of differential and integral equations, and once training is complete they can be evaluated and differentiated rapidly using automatic differentiation. Having access to such a rapid solver of differential and integral equations allows us access to the entire repertoire of methods from parametric statistics with these physical models. In particular for Bayesian inference our focus will be on how this allows MCMC to be applied efficiently.

The remainder of this introduction proceeds as follows. In Section 1.1 we introduce a general form of statistical model that we will consider in this thesis, and outline the Bayesian approach to inference in these models along with common solution methods. Section 1.2 uses examples to highlight why the Bayesian approach is beneficial over its frequentist counterpart. Section 1.3 introduces the physical models that are considered in this thesis, followed by a brief description of commonly applied numerical methods for their solutions. In Section 1.4 we present some preliminary knowledge on deep learning that is assumed throughout the thesis. Section 1.5 of this chapter introduces how the previous topics interact by introducing Bayesian inverse problems governed by a PDE forward model. In this section we outline some established approaches to solving these problems and highlight why most common approaches are costly, inaccurate, or both. Then, in Section 1.6 we give an overview of modern accelerated MCMC approaches that can be applied to make sampling based approaches to these problems more tractable. Finally, having established the required context, this introduction ends with a summary of the contributions of this thesis and an outline of the remaining chapters.

1.1 The Bayesian approach to statistics

Let us assume that we have a dataset of observations \( \{ \hat{z}_i \}_{i=1:N} \) from some process that we seek to understand, and associated to each data point we have a vector of covariates \( \{ \hat{x}_i \}_{i=1:N} \). The covariates \( \hat{x}_i \) can be thought of as explanatory variables for \( \hat{z}_i \), and the mechanisms by which the \( \hat{x}_i \) explain the \( \hat{z}_i \) is encoded by a statistical model with unknown variables \( \alpha \), which in general can include scalar and vector valued parameters, as well as non-parametric terms such as Gaussian processes. A common notation for this relationship that can apply to all examples in this thesis is

\[
\hat{z}_i = f(\hat{x}_i; \alpha) + \epsilon_i. \tag{1.1.1}
\]
We will take the common convention that the $\epsilon_i$ are i.i.d, and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, where $\sigma$ may be known or unknown. The model (1.1.1) therefore specifies that each $\hat{z}_i$ is independently normally distributed with mean $f(\hat{x}_i; \alpha)$ and variance $\sigma^2$. Assuming for instance that $\sigma$ is unknown and combining all the unknowns into a single variable $\theta := \{\alpha, \sigma\} \in \Theta$, the probability density function of the data conditional on $\theta$ being known is the likelihood function $p(\hat{z}|\theta, \hat{x})$.

The goal in a statistical inference scheme is to find $\theta$ such that the model ‘fits’ the data. Once the model has been fit, $f(\cdot)$ can be evaluated at out-of-data locations in order to predict new outcomes, and the terms in $\theta$ that we infer may give us a deeper insight into the governing process. What it means to fit a model has several interpretations, some more of which will be elaborated on in Section 1.2. From the Bayesian perspective we view $\theta$ as uncertain and assign a distribution $p(\theta)$ to these variables that represents our beliefs about their value prior to observing any data. Fitting the model then consists of using Bayes rule to condition this prior distribution on available data, thus producing the posterior distribution

$$p(\theta|\hat{z}, \hat{x}) = \frac{p(\hat{z}|\theta, \hat{x})p(\theta)}{\int p(\hat{z}|\theta, \hat{x})d\theta}.$$  (1.1.2)

The posterior distribution represents an updated belief about the parameters after having observed the data, and since it is a distribution it naturally provides a form of uncertainty quantification that takes into account both our prior beliefs and the observations. Computing this posterior distribution is the main goal in a Bayesian inference problem, and once complete we can perform uncertainty quantification for any quantity of interest by marginalising the posterior from the relevant joint distribution. A commonly sought after example of this is the predictive posterior distribution of $z$ corresponding to out-of-data covariates $x$, which can be computed as

$$p(z|x, \hat{z}, \hat{x}) = \int p(z|\theta, x)p(\theta|\hat{z}, \hat{x})d\theta.$$  (1.1.3)

Using (1.1.2) we are able to write down the posterior PDF analytically up to an unknown constant of proportionality (the denominator is in general an intractable integral in multiple dimensions). Yet this relatively complete picture of the functional form of the posterior is of limited use in practice, as we are typically interested in statistics of the posterior such as the mean and covariance rather than the PDF itself. Moreover, the computation of these statistics, as well as the marginalisations required to compute the posterior distribution of quantities of interest such as (1.1.3), require integrals.
over the parameters. These integrals are intractable by direct means in general, and therefore approximations must be made. Currently, two broad approaches to this end dominate the literature.

The first are sampling based methods such as Markov chain Monte Carlo \cite{52}. In this approach we construct a Markov chain $\theta_1, \theta_2, \ldots, \theta_K \in \Theta$ such that the empirical density of the states visited by this chain converges in distribution to the posterior (1.1.2) as $K \to \infty$. A description of how this is achieved as this is given in Section 1.6.1, though we highlight now that the key usefulness of this approach is that it allows integrals such as (1.1.3) to be tackled by Monte Carlo integration using the sample points

$$p(z|x, \hat{z}, \hat{x}) \approx \frac{1}{K} \sum_{i=1}^{K} p(z|\theta_i, x). \quad (1.1.4)$$

The advantages of MCMC sampling schemes are accuracy and flexibility. We have long had proof that MCMC based methods converge to the true posterior distribution, and such methods can in principle be applied to sample from any distribution that we are able to evaluate up to some constant of proportionality. The biggest challenge facing MCMC however is efficiency. This is because the rate of convergence of the empirical density of the MCMC sample to the true posterior depends on factors such as the dimension and complexity of the distribution, as well as manual choices such as the transition kernel which can be tricky to calibrate. Furthermore, the sampling speed itself scales linearly with the time taken to evaluate the distribution, which can introduce a huge cost in settings with expensive likelihood such as those considered throughout this work.

The second broad approach is approximation based methods, such as variational inference \cite{102} and Laplace approximations \cite{134}. These methods construct approximations to the posterior quantity of interest

$$q(\theta) \approx p(\theta|\hat{z}, \hat{x}). \quad (1.1.5)$$

These approximations typically come from a family of distributions $q \in Q$ consisting of one or more well understood distributions that are amenable to marginalisation or sampling. The final approximation to the posterior is the $q^* \in Q$ that best satisfies the criteria of the method being applied. For example in variational inference $Q$ is typically a family of factorisable distributions and we minimise the Kullback–Leibler
divergence, giving the estimate
\[ q^* = \arg\min_{q \in Q} \int_\Theta q(\theta) \log \left( \frac{q(\theta)}{p(\theta|\hat{z}, \hat{x})} \right) d\theta, \]  
(1.1.6)

whereas, in the Laplace approximation \( Q \) is the set of multivariate Gaussian distributions and we choose the mean and covariance such that \( q^* \sim \mathcal{N}(\mu^*, \Sigma^*) \), where
\[ \mu^* = \arg\max_{\theta \in \Theta} p(\theta|\hat{z}, \hat{x}), \quad \Sigma^* = \text{Hess}^{-1}_\theta \{ -\log p(\theta^*|\hat{z}, \hat{x}) \}. \]  
(1.1.7)

These approximations are usually optimised using iterative procedures such as gradient descent, and typically converge more quickly than MCMC based methods [16]. The main drawback however is that this approach cannot converge to the true posterior in general, as the true posterior generally does not lie in \( Q \) and thus there is a systematic error [138]. Furthermore, without knowledge of the true posterior it is difficult to assess whether the best approximation in \( Q \) is accurate. For example if the true posterior is non-factorisable and bi-modal, neither of these approaches will represent this behaviour; however since we are unlikely to know these properties of the posterior a-priori we would likely proceed with this inaccurate estimate. Approximation based methods are therefore commonly applied in situations where sampling cannot be performed efficiently, such as problems with very high-dimensional posteriors or those with expensive likelihoods to evaluate [81].

1.2 Why take the Bayesian approach?

We have outlined the principles underpinning the Bayesian approach to statistical inference and the two most common approaches to applying it. We now demonstrate why we would want to take the Bayesian approach by applying it to a 1-dimensional polynomial regression toy problem. We also implement a commonly applied frequentist alternative, maximum-likelihood estimation, as a comparison. Our focus here is to show how the Bayesian estimate differs from the frequentist estimate and highlight why this approach is desirable in many cases. Computational considerations can be ignored here as both methods can be implemented analytically, however the computational cost of a Bayesian analysis can often be prohibitive as we will see in subsequent sections.

Suppose we have the following data, generated by sampling noisy data from a Gaussian curve at 32 locations.
Let us consider the statistical model (1.1.1) with polynomial estimate

\[ f(x; \alpha) = \sum_{d=0}^{D} \alpha_d x^d = \phi(x)^T \alpha, \tag{1.2.1} \]

where \( \phi(x) = [1, x, x^2, \ldots, x^D]^T \). To simplify the calculations we assume that \( \sigma \) is known, so that the logarithm of the likelihood function is given by

\[ \log p(z|x, \alpha) = -\frac{N}{2} \log(2\pi \sigma^2) - \frac{1}{2\sigma^2} \sum_{n=1}^{N} (z_n - f(x_n; \alpha))^2. \tag{1.2.2} \]

The maximum likelihood estimator is the maximiser of this expression and therefore given by the coefficients that achieve the least cumulative squared error. Introducing \( \Phi \in \mathbb{R}^{N \times D+1} \) such that \( \Phi_{n,d} = x_n^{d-1} \), this can be re-written as the linear least squares problem

\[ \alpha_{ML} = \text{argmin}_{\alpha \in \Theta} \| z - \Phi \alpha \|_2^2. \tag{1.2.3} \]

The optimal coefficients can be obtained by setting the gradient of (1.2.3) equal to
zero, yielding the solution to the well known normal equations

\[ \alpha_{ML} = (\Phi^\top \Phi)^{-1} \Phi^\top z. \]  

(1.2.4)

As a comparison, in our Bayesian approach we assign a multivariate normal prior distribution to the coefficients with mean vector zero and diagonal covariance \( \beta^2 I \), so the log-prior distribution is

\[ \log p(\alpha) = -\frac{D}{2} \log(2\pi \beta^2) - \frac{1}{2\beta^2} \sum_{d=0}^{D} (\alpha_d)^2. \]  

(1.2.5)

The log-posterior distribution is therefore

\[ \log p(\alpha|x, z) = -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (z_n - f(x_n; \alpha))^2 - \frac{1}{2\beta^2} \sum_{d=0}^{D} (\alpha_d)^2 + \text{const}, \]  

(1.2.6)

where we have gathered the terms that are constant in \( \alpha \) into const. With some manipulation of (1.2.6) it can be shown that the posterior is also Gaussian in \( \alpha \), a property known as self-conjugacy of the Gaussian distribution, and so the mean of the posterior is the MAP estimate

\[ \mu^* = \arg\min_{\alpha \in \Theta} \left( \|z - \Phi \alpha\|_2^2 + \frac{\sigma^2}{\beta^2} \|\alpha\|_2^2 \right). \]  

(1.2.7)

Setting the gradient of this expression equal to zero we can derive

\[ \mu^* = (\Phi^\top \Phi + \frac{\sigma^2}{\beta^2} I)^{-1} \Phi^\top z, \]  

(1.2.8)

and the variance of the posterior distribution is given by

\[ \Sigma^* = \sigma^2 \left( \Phi^\top \Phi + \frac{\sigma^2}{\beta^2} I \right)^{-1}. \]  

(1.2.9)

Comparing the objective function for maximum-likelihood (1.2.3) to the MAP objective (1.2.7) reveals the first benefit of the Bayesian approach. That is the additional \( \frac{\sigma^2}{\beta^2} \|\alpha\|_2^2 \) in (1.2.7) encourages us to find a coefficients with lower absolute value, leading to simpler ‘regularised’ solutions that adhere to the principles of Occam’s razor while still obeying the data. If one has existing knowledge of the process underlying the data, then an alternative prior mean and covariance can be chosen to achieve regularisation towards this belief. In the absence of such knowledge however, the above-stated
prior provides a convenient and objective form of regularisation, and one that often arises in the solutions of ill-posed inverse problems, statistics, and machine learning, often with different names such as Tikhonov regularisation [88] or ridge regression [68, 69]. The Bayesian formulation therefore benefits from the same behaviour that motivates the use of these methods. The effect of this is shown in Figure 1-2 where the Bayesian estimate shows better agreement with the underlying function, despite maximum-likelihood achieving a closer agreement with the data.

Figure 1-2: Bayesian and maximum likelihood estimates of the underlying Gaussian curve using the data.

Beyond being an effective regulariser, the second major benefit of the Bayesian approach is that its probabilistic construction provides a natural and complete form of uncertainty quantification in the form of the posterior distribution [52]. This can be seen as the blue band in Figure 1-2, which represents the 95% credible interval of the predictive posterior distribution. In this specific example we used self-conjugacy of the Gaussian distribution to obtain a Gaussian posterior, however in general the posterior can be any probability distribution and thus the posterior can capture complex probabilistic structure such as correlated, skewed, nonlinear, or heavy-tailed behaviour. This is not the case in maximum-likelihood estimation, where only a point estimate is returned directly. And while Bayesian approaches derive their uncertainty estimates by consistently applying the laws of conditional probability, frequentist methods generally calibrate their uncertainties using additional (potentially inconsistent) assumptions such as normality, which can restrict the expressivity presented in the structure of the uncertainty.
1.3 Differential and integral equations and their solutions

Many processes of interest, ranging from weather patterns\cite{34} and tectonic movements\cite{146}, to cellular diffusion\cite{49} and electrical impedance\cite{106}, and even financial markets\cite{4} can be modelled by partial differential equations or integral equations. Throughout this thesis, we are interested in applying Bayesian methodology to estimate quantities of interest from data related to physical processes such as these, therefore we must be able to solve these equations. For both types of equation, the common solution methods reduce to solving linear equations of the form

\begin{equation}
Au = b.
\end{equation}

(1.3.1)

Here the vector \( u \in \mathbb{R}^J \) is the discretisation of a function that approximates the solution to the equation, and the form of the matrix \( A \) and vector \( b \) are dictated by the equation and the method being applied to solve it. The importance of this is that the computational cost of solving a PDE or integral equation is therefore related to the cost of solving this system. For a general dense matrix \( A \in \mathbb{R}^{J \times J} \), the worst case scenario is that this cost is \( O(J^3) \), though in practice the matrices that arise from numerical solvers often have sparse structures that can be exploited to reduce this cost, in optimal cases to \( O(J) \). The number of degrees of freedom \( J \) required is highly problem-dependent, and in particular it typically grows exponentially with the dimension of the PDE or integral equation being solved as the following examples will illustrate. Hence, despite the availability of solvers that scale efficiently with \( J \), the computational burden associated with inverting large matrices remains a major hurdle to the application of statistical methods with large scale mathematical models.

To illustrate why the most widely used numerical solvers result in linear equations of the form\[1.3.1\] we will briefly describe the most common approaches. Let us take Poisson’s boundary value problem on the 2-dimensional square domain \( \Omega = [0,1] \times [0,1] \) as an exemplar PDE

\[\Delta u(x) = g(x), \quad x \in \Omega,\]

(1.3.2)

with boundary conditions \( u(\partial \Omega) = 0 \).

In one common approach, the finite differences scheme\[61\], this domain is discretised using a rectangular mesh of points \( \{ x_{i,j} : i, j = 0, \ldots, K \} \), where \( x_{i,j} = [ih, jh] \) with
$h = 1/K$. The differential operator is then approximated numerically using adjacent mesh points, resulting in the Laplacian being replaced with the second-order central differences operator, giving the discretised equation

$$\frac{u(x_{n+1,m}) + u(x_{n-1,m}) + u(x_{n,m+1}) + u(x_{n,m-1}) - 4u(x_{n,m})}{h^2} = g(x_{n,m}), \quad (1.3.3)$$

for $n, m = 1, 2, \ldots, K - 1$. This defines a set of $(K - 1)^2$ linear equations to be solved together with $u(x_{0,n}) = u(x_{K,n}) = u(x_{n,0}) = u(x_{n,K}) = 0$, $n = 0, 1, \ldots, K$. Thus we have a linear equation of the form $(1.3.1)$ with dimension $J = (K + 1)^2$ whose solution approximates the PDE solution at the set of mesh points, which can then be interpolated to give a global solution. Note that in our two-dimensional example, the number of degrees of freedom $J = (K + 1)^2 = O(K^2)$ results from using $K + 1$ discretisation points in each dimension. Applying finite differences at the same resolution to the general $d$-dimensional Poisson boundary value therefore results in $O(K^d)$ degrees of freedom, thus each additional dimension increases the size of the linear system $(1.3.1)$ by an order of magnitude.

Figure 1-3: Left: 5 point stencil used in finite differences. Right: Triangular elements of a finite element discretisation.

The other mainstream method for solving PDEs is the finite element method [61]. This method solves the PDE $(1.3.2)$ based on its weak formulation, which is the problem of
finding \( u \in H^1_0(\Omega) \), such that for any \( v \in H^1_0(\Omega) \) we have,

\[
\int_{\Omega} g(x)v(x)dx = \int_{\Omega} \Delta u(x)v(x)dx = -\int_{\Omega} \nabla u(x)\nabla v(x)dx.
\]

(1.3.4)

Here the Sobolev space \( H^1_0(\Omega) \) is the space of functions that disappear on the boundary, and have finite \( L^2 \)-norm in the zeroth and first derivative. This weak formulation has a unique solution for any functions \( g, v \in H^1_0(\Omega) \), however to approximate the solution to this problem numerically these function are replaced with functions from a finite dimensional subspace \( V \subset H^1_0(\Omega) \) written in terms of basis functions \( v_n \) as

\[
u(x) \approx \sum_{n=1}^{J} u_n v_n(x) \in V,
\]

\[
g(x) \approx \sum_{n=1}^{J} g_n v_n(x) \in V.
\]

(1.3.5)

In general \( u, g \notin V \) so this step defines an approximation as denoted in (1.3.5). The coefficients \( u_i \) are unknowns that will be solved for, while the \( g_i \) are chosen such that the series in (1.3.5) approximates \( g(x) \). Letting \( v(x) = v_j(x) \) in (1.3.4) for \( j = 1, \ldots, J \) results in the finite dimensional approximation to the weak formulation

\[
-\sum_{n=1}^{J} u_n \int_{\Omega} \nabla v_n(x)\nabla v_j(x)dx = \sum_{n=1}^{J} g_n \int_{\Omega} v_n(x)v_j(x)dx.
\]

(1.3.6)

The integrals involved in (1.3.6) are inner products that can be made simple to evaluate, most commonly by subdividing the domain into a mesh of triangular elements and using a piecewise polynomial basis on each element (as in traditional finite elements), or sometimes by using orthogonal functions such as a Fourier basis (as applied in spectral methods). Assuming that we can evaluate these inner products, solving the approximated weak formulation again reduces to solving a set of linear equations of the form (1.3.1) for \( u = [u_1, \ldots, u_J]^T \). In comparison to finite differences, finite element methods offer more flexible discretisations that allow for non-rectangular domains and mesh generation schemes that can more densely target particular regions to improve local accuracy. Similarly to the finite difference scheme however, the number of degrees of freedom \( J \) in finite elements generally increases exponentially as dimensions are added to the PDE. The reason for this is the same as in finite differences, that is because the mesh must be extended into these higher dimensions at a consistent resolution.
Finite differences and finite elements are the most popular methods used for PDEs, and most other common approaches such as the finite volume method can also be shown to result in a similar system of linear equations. These approaches are all easily extended to higher dimensions, however to retain the same accuracy the number of degrees of freedom $J$ must typically grow by an order of magnitude for each additional dimension, making the solution of (1.3.1) significantly more expensive [61]. Furthermore, if the PDE has a component with a temporal derivative (resulting in the forced heat equation in this case), it is typically discretised over the temporal domain by a finite difference approximation and a linear equation of the form (1.3.1) is solved at each time step. Thus, in this setting the computational cost is multiplied by the size of the temporal discretisation.

When it comes to the numerical solutions of integral equations, the standard approach is to approximate the integral term using quadrature [32]. Take for example the Fredholm integral equation of the second kind

$$u(t) = g(t) + \lambda \int_a^b K(t, s)u(s)ds.$$  

(1.3.7)

Here the function $g$ and kernel $K$ are known, and $u$ is the unknown solution. To solve this equation the integral term can be approximated by a quadrature rule. For example Gauss-Legendre quadrature with nodes $\{t_1, \ldots, t_J\}$ and weights $\{w_1, \ldots, w_J\}$ leads to an equation of the form

$$u(t_j) = g(t_j) + \lambda \sum_{i=1}^{J} w_i K(t_j, t_i)u(t_i),$$  

(1.3.8)

which must be satisfied for $j = 1, \ldots, J$. This is again a set of linear equations that can be written in the form (1.3.1). Alternative integral equations such as first kind and Volterra equations can be handled similarly, and variations to this approach usually amount to a change in the quadrature rule that is applied. As with the PDE solvers one can generalise this approach to higher dimensions by applying multidimensional quadrature rules. The number of quadrature nodes in such rules grows exponentially with the dimension of the function that is to be integrated, and therefore once again so does the size of the linear equation (1.3.1) which must be solved in these cases.

Overall these methods are very effective and their use is widespread across a multitude of fields such as epidemiology [113] and thermodynamics [72]. The linear discretisation
that is present in each approach has clear theoretical benefits but substantial practical drawbacks. On one hand it allowed us to develop a rigorous theory of the accuracy of these methods [153], as well as leverage existing algorithms designed for the fast solutions of such systems. On the other hand, linear schemes such as these suffer the curse of dimensionality, meaning some problems of interest are too expensive to tackle. An example of this is the 7-dimensional PDE for the density of particles governed by Langevin dynamics in 3-dimensions often applied to model atmospheric dispersion. Assuming $K + 1$ discretisation points in each dimension of a finite difference scheme, this PDE requires us to solve for $J = (K + 1)^7$ variables. Solver costs such as this impact the efficiency of Bayesian problems involving these equations; we discuss this in further detail after introducing some deep learning preliminaries.

1.4 Deep learning preliminaries

1.4.1 Neural networks

Many problems can be characterised by the need to compute relationships between variables. A neural network is a parameterised function approximator commonly used to represent these relationships. However, contrary to classical function approximation methods which are typically constructed linearly from basis functions of the inputs, a neural network is constructed by composing multiple nested nonlinearities and applying these to its inputs. This approach to function approximation has been proven to have remarkable approximation properties in high dimensions [8, 60], and shown state of the art performance in many applications [89, 94]. Deep learning is concerned with the study and application of neural networks.

More concretely, feedforward neural networks are functions composed of neurons that are arranged into sequential layers connected by edges. During the evaluation of a neural network for some input vector (known as a forward pass), the values of the first layer (the input layer) are set to equal the input vector; this then sequentially defines the values of the intermediate layers (hidden layers) and the final layer (the network outputs). The mechanisms by which the inputs define the hidden layers and outputs are controlled by the parameters of the neural network, and during training these parameters are estimated by applying gradient descent to optimise some objective. The architecture of a neural network is defined by the specific arrangement of layers and edges in the network, as well as constraints on the parameters of the network. Perhaps
the simplest architecture is the fully-connected feedforward neural network, where each neuron is connected to every neuron in the subsequent layer. This is depicted in Figure 1-4.

Figure 1-4: Architecture of a fully-connected feedforward neural network mapping a 3-dimensional input to a 1-dimensional output through a pair of 5-dimensional hidden layers.

To make this description more precise, let \( n_i \) represent the number of neurons in layer \( i \) and consider inputs \( \{x_{0,1}, \ldots, x_{0,n_0}\} \). In the fully-connected feedforward architecture, the values \( \{x_{i,1}, \ldots, x_{i,n_i}\} \) from layer \( i \) are propagated forward into neurons in the next layer by \( n_{i+1} \) mappings of the form

\[
x_{i+1,k} = \phi_i \left( \sum_{j=1}^{n_i} \beta_{k,j}^i x_{i,j} + \beta_{k,0}^i \right).
\]

That is, they undergo a linear transformation defined by the neural network parameters \( \beta_{k,j}^i \), followed by the application of the function \( \phi_i \), which is typically a nonlinear activation function. There are \( n_{i+1} \) of these mappings at layer \( n_i \), therefore these form
system of equations that can be written

\[ x_{i+1} = \phi_i(B_i x_i + \beta_i), \quad (1.4.2) \]

where the matrix of weights \( B_i \) and vector of biases \( \beta_i \) are constructed accordingly and the activation function \( \phi_i \) acts elementwise on the vector.

Beyond the fully-connected feedforward architecture, a breadth of different architectures have been developed that are adapted to different purposes, such as convolutional neural networks for image data [89], recurrent neural networks for sequential data [59], and auto-encoders for dimensionality reduction [67] to name a few. Relations between some neural network architectures and continuous mathematical systems have also been drawn, for example residual neural networks are related to dynamical systems [27], and convolutional networks are related to PDEs [136].

For the purposes of this thesis we implement all examples using the standard fully-connected feedforward architecture. Specifically, we always use hyperbolic tangent activation functions at all layers, with the exception of the output layer where the identity function is used. The number of nodes and layers used for each problem is specified where appropriate. This architecture was chosen to provide reassurance that the methods described are effective without exhaustive network prototyping. Of course, more specialised architectures could also be developed which would likely lead to further improvements, but this is not our focus. For brevity of notation, we will signify that a function is a neural network by a hat without explicitly referencing its weights and biases. For example, we frequently write \( \hat{u}(x) \) to denote a neural network approximation to the solution \( u(x) \) of a PDE.

1.4.2 Training and automatic differentiation

To establish a suitable set of weights and biases for a given task a neural network must be trained. The standard approach to this is to formulate the objective of the neural network in terms of a loss function, and then use gradient descent to seek the weights and biases that cause the network to minimise this loss function. For example, if the purpose of the neural network is to interpolate data points, the squared distance between the network output and data might be chosen as the loss function. The minimisation problem is generally a non-convex optimisation problem, often with millions of degrees of freedom, and a large evaluation cost at each step if large datasets are
To account for these factors a variety of strategies have been developed to accelerate gradient descent. Momentum-based gradient descent methods are widely applied \cite{00}, as well as methods which adaptively determine the gradient descent step size \cite{02}. These approaches improve upon vanilla gradient descent whilst only requiring first order derivative information. Additionally, second-order methods such as BFGS \cite{03} are sometimes used for modest model sizes. Stochastic gradient descent is another extension \cite{05}, introduced to account for the cost associated with processing large datasets. In this approach, each gradient descent iteration is applied to a random subset of the data, thus reducing the burden of processing the entire dataset each iteration. Combinations of these approaches have also been developed, most notably the Adam algorithm \cite{06}, which combines all of these features, is the most common training algorithm applied today.

A prerequisite to applying any of the algorithms mentioned is the ability to compute derivatives of the loss function with respect to the weights and biases of the neural network, which is achieved using automatic differentiation. For brevity we do not include an in-depth discussion of automatic differentiation (as can be found in \cite{12,13}). The general idea is to view a computer program that maps inputs to outputs as a sequence of nested function compositions applied to those inputs. In general, each computational operation involved can be described by simple mathematical operations, and so through this viewpoint the program defines a mathematical function, that is differentiated with respect to its inputs through the recursive application of the chain rule. Several user-friendly implementations of this approach now exist \cite{07,08,10,14}, which are optimised to take advantage of parallel hardware in order to flexibly and efficiently compute derivatives arising in complex computer programs.

Once trained, a feedforward neural network defines a mathematical function that can be evaluated anywhere, with parameters optimised to minimise the chosen loss function. A trained network can therefore be thought of approximately as a functional minimiser, the behaviour of which depends on the loss function. In this work we apply loss functions that embed PDE and integral equations dynamics, and through this approach we are able to efficiently approximate their parametric solutions and apply them to accelerate Bayesian inference without using external numerical solvers or labelled training data. The remainder of this introductory chapter motivates the requirement for accelerated Bayesian inference methods in the presence of physics-governed likelihoods, and provides a survey of classical methods and recent developments to achieving this.
1.5 Bayesian inverse problems

An inverse problem can be defined as using observations from some process to infer the underlying properties that caused these observations to occur. In contrast to the previous section where, for example, the function $g(x)$ was known in Poisson’s equation (1.3.2) and we sought to solve the forward problem for $u(x)$, the inverse problem might have us seeking to infer $g(x)$ given (possibly sparse and noisy) measurements from $u(x)$. Using this equation as an example once more, we can naturally describe this inverse problem as a statistical inference problem

$$\hat{z}_i = u(\hat{x}_i; \alpha_g) + \epsilon_i.$$ (1.5.1)

Here $z_i \in \mathbb{R}$ are noisy observations of the solution at locations $x_i \in \Omega$, the parameter $\alpha_g \in \Theta$ determines the unknown function $g = g(x; \alpha_g)$, and $u(x; \alpha_g)$ is the solution of the equation provided this $g$. It is theoretically well defined to consider Bayesian inference for function valued parameter $g$ directly, by using appropriate function space norms in the prior and likelihood as described in [137], however for practical computation this function must be discretised, and so $\alpha_g$ can be interpreted as the parameters of a finite dimensional discretisation of $g$.

Of course (1.5.1) is just a specific example of (1.1.1), and inverse problems in general can technically just be thought of as parameter inference problems of this form. A Bayesian inverse problem is therefore simply the Bayesian inference problem applied to (1.5.1) as described in Section 1.1. The key differentiating factor that often causes one to speak of inverse problems over statistical inference is the solutions dependency on the evaluation of nontrivial forward problems. Solving these forward problems using numerical schemes is often computationally intensive, and useful information that is usually implicit in simple parametric statistical settings is often missing from these solution techniques, such as gradients of the solution with respect to the parameters required to find a maximum-likelihood or MAP estimate. As a result, many of the tools applied in parametric statistics are too expensive to apply directly for Bayesian inverse problems. Continuing with the Poisson equation example (1.5.1), here we give a flavour of the dominant approaches to overcoming this limitation.
1.5.1 Approximation based approaches

Methods that approximate the posterior using the optimal distribution $q^*$ from some parameterised class of distributions $Q$ are common for Bayesian inverse problems [36, 147]. These methods are mostly implemented using gradient based optimisation just as we described in Section 1.1, however computing these gradients is significantly more challenging in the case of inverse problems. To see why suppose we seek to produce a Laplace approximation for Poisson’s problem, and that we define $g$ to be a polynomial with $P$ terms and coefficients $\alpha_g$, each with independent normal prior distribution. As shown in Section 1.2 the log posterior distribution is

$$\log p(\alpha_g|\mathbf{x}, \mathbf{z}) = -\frac{1}{2\sigma^2} \| \mathbf{z} - u(\mathbf{x}; \alpha_g) \|_2^2 - \frac{1}{2\beta^2} \| \alpha_g \|_2^2 + \text{const.} \quad (1.5.2)$$

To find the Laplace approximation we first need to optimise this with respect to $\alpha_g$. Defining the Jacobian $\nabla_{\alpha} u(\mathbf{x}; \alpha_g) \in \mathbb{R}^{P \times N}$ such that $\nabla_{\alpha} u(\mathbf{x}; \alpha_g)_{i,j} = \frac{\partial u(\mathbf{x}_j; \alpha_g)}{\partial \alpha_i}$, we require the gradients

$$\nabla_{\alpha} \log p(\alpha_g|\mathbf{x}, \mathbf{z}) = \frac{1}{\sigma^2} \nabla_{\alpha} u(\mathbf{x}; \alpha_g) (\mathbf{z} - u(\mathbf{x}; \alpha_g)) - \frac{1}{\beta^2} \alpha_g. \quad (1.5.3)$$

Standard numerical solvers do not return $\nabla_{\alpha} u$ so applying this is nontrivial. We can approximate the $i^{th}$ component of $\nabla_{\alpha} u$ using a finite difference approximation $(\nabla_{\alpha} u)_i \approx \frac{1}{\delta}(u(\mathbf{x}, \alpha_g + \delta \mathbf{e}_i) - u(\mathbf{x}, \alpha_g))$, thus a gradient computation can be produced by performing $P + 1$ PDE solves. This approach is applied often in practice since it is simple to understand and can be implemented using the same solver that is used for the PDE.

In many cases of interest however, the number of parameters is large and a finite difference approximation to $\nabla_{\alpha} u$ is impractical. In these cases a related PDE known as the adjoint equation is a useful tool. Using the solution of the adjoint PDE we can write the gradients $\nabla_{\alpha} u$. A thorough derivation of how the adjoint PDE arises can be found in texts on sensitivity analysis and optimal control [23, 77]. A simple way to derive it for our Poisson example is to consider the PDE after linear discretisation, so that it can be written as in (1.3.1). The solution map over the discretisation points is defined by the linear transformation $\mathbf{b} \mapsto S\mathbf{b}$ where $S = A^{-1}$, and $\mathbf{b} = \mathbf{b}(\alpha_g)$ is determined by $g$. Let us also assume that the data $\mathbf{z}$ is defined at all discretisation
points. The log-posterior and its gradient can be written

$$\log p(\alpha | x, z) = -\frac{1}{2\sigma^2} \| z - Sb \|^2 + \frac{1}{2} \| \alpha \|^2 + \text{const} \quad (1.5.4)$$

$$\nabla_\alpha \log p(\alpha | x, z) = \sigma^{-2} \nabla_\alpha b S (z - Sb) - \frac{1}{\beta^2} \alpha. \quad (1.5.5)$$

If we write $w = S(z - Sb)$ then

$$S^{-1}w = z - Sb, \quad (1.5.6)$$

or in terms of $A$ we have

$$Aw = z - u. \quad (1.5.7)$$

Because $A$ is an arbitrary linear discretisation of the Laplacian, we can interpret $w$ as the discretisation of a function $w$ which solves

$$\Delta w(x, \alpha_g) = z(x) - u(x, \alpha_g). \quad (1.5.8)$$

Equation (1.5.8) is known as the adjoint equation for the Poisson problem.

Thus to compute the gradient (1.5.5) at $\alpha_g$ we must first solve Poisson’s equation (1.3.2) with these parameters, then use the result of this to solve the adjoint equation (1.5.8) for $w$. The gradient can then be constructed as

$$\nabla_\alpha \log p(\alpha | x, z) = \sigma^{-2} (\nabla_\alpha b) w - \frac{1}{\beta^2} \alpha. \quad (1.5.9)$$

This approach to computing the posterior derivatives requires just two PDE solves, thus providing a significant speed-up when compared to the finite differences approximation. Here we derived the equation for a specific case, however the adjoint method can be applied with various PDEs and objective functions, making it a useful tool for a range of PDE constrained optimisation problems. When applied to optimise approximations to the posterior, the adjoint method can attain tractable Bayesian estimates, though these are still limited by the shortcomings associated with approximation based approaches.
1.5.2 Sampling based approaches

Approaches that apply MCMC while using numerical solvers to evaluate the likelihood during each iteration are sometimes applied in practice. These methods work in principle. They can be made arbitrarily accurate, and if the Markov chain has a simple transition kernel then no gradients of the solution with respect to the parameters are required. Unfortunately this approach is only viable for smaller problems with inexpensive solvers and few parameters, as thousands of iterations typically are required for a sufficiently large sample and the efficiency of MCMC with simpler transition kernels is poor as the number of parameters increases. A common approach to overcome this is to incorporate a surrogate function capable of cheaply emulating the forward problem. Here we introduce established classical and data driven methods used to construct suitable surrogates, and outline a simple approach through which they can effectively accelerate sampling. In section 1.6 we then provide a more detailed description of MCMC and survey various advanced acceleration methods, some of which utilise such surrogates in more sophisticated ways.

The stochastic Galerkin projection \[55, 161\] is a mathematically elegant means of constructing a surrogate. In this method the solution is written in terms of stochastic coefficients, yielding a polynomial chaos expansion

\[
    u(x; \alpha_g) \approx K \sum_{k=0}^{K} \phi_k(\alpha_g) u_k(x).
\]  

(1.5.10)

Here we interpret \(\alpha_g\) as a random variable with PDF \(P(\alpha_g)\) supported over the parameter space \(\Theta\), and \(\phi_k\) are polynomials that are orthogonal with respect to the inner product defined by expectation with respect to \(P(\alpha_g)\). The \(u_k\) can be thought of as the unknown (function valued) polynomial coefficients that make (1.5.10) approximate the parametric PDE solution. Decomposing \(g(x, \alpha_g)\) similarly and substituting into Poisson’s equation (1.3.2) yields a random PDE, then taking the inner product of this PDE against \(\phi_n\) gives

\[
    \sum_{k=0}^{K} \Delta u_k(x) \int_{\Theta} \phi_k(\alpha_g) \phi_n(\alpha_g) dP(\alpha_g) = \sum_{k=0}^{K} g_k(x) \int_{\Theta} \phi_k(\alpha_g) \phi_n(\alpha_g) dP(\alpha_g). 
\]  

(1.5.11)

The integrals in this expression are once again inner products that are one when \(n = k\) and zero otherwise. The result of the Galerkin projection is therefore a system of \(K + 1\) deterministic PDEs with solutions \(u_k\) that can be solved for using standard numerical
methods. In general these PDEs will be coupled, thus the linear system to be solved requires the inversion of a \((K + 1)J \times (K + 1)J\) matrix, although in this example we see a decoupling so the \(K + 1\) PDEs can be solved independently. Once solved \((1.5.10)\) gives an approximation to the solution that converges to true solution in the \(L^2\)-norm induced by \(P(\alpha_g)\). This solution is defined over \(\Omega \times \Theta\), thus once computed it can be quickly evaluated for different parameter values. This surrogate is then substituted in place of the full model during the likelihood evaluation step during MCMC. As the evaluation of the forward model is typically the dominant cost involved in solving a Bayesian inverse problem using MCMC, this can significantly accelerate sampling. Of course the accuracy of the inference depends on the surrogate accuracy, and a more accurate surrogate is typically more expensive to produce and evaluate. Consideration of the cost/accuracy trade-off is therefore important when using surrogates in this way.

The other main method of generating surrogates is more data driven. In this approach a set \(\alpha^i_g \in \Theta\) of \(N\) parameters are chosen, and the corresponding PDE is solved numerically for each, giving an ensemble of solutions \(u(x; \alpha^i_g)\). The surrogate is then formed by interpolating the ensemble using a suitably flexible function approximator \(\hat{u}(x; \alpha_g) : \Omega \times \Theta \to \mathbb{R}\). Various strategies exist to perform this interpolation which we won’t exhaust, a common choice (see for example [15, 85]) is to define \(\hat{u}(x; \alpha_g)\) to be a neural network trained to minimise

\[
\sum_{i=1}^{N} \| \hat{u}(x; \alpha^i_g) - u(x; \alpha^i_g) \|_{L^2(\Omega)}^2.
\]  

(1.5.12)

Once again this approximated solution is defined over \(\Omega \times \Theta\), thus it can be quickly evaluated for different parameters to accelerate MCMC. The main limitation of this approach is the requirement to solve the PDE numerically a large number of times in order to generate the data needed to construct the surrogate. This is particularly restrictive if the number of parameters is large, since assuming uniform spacing between the \(\alpha^i_g\), the number of solves required to fill \(\Theta\) grows exponentially with its dimension.
1.6 Review of Markov chain Monte Carlo and accelerations

Markov chain Monte Carlo (MCMC) is a technique used to draw samples from a distribution that is known up to a constant of proportionality. Assuming we can do this, the empirical density of any quantity of interest defined by the sample points converges weakly to the true distribution over that quantity as the sample size increases. MCMC is both simple to understand and implement, yet robust enough to be applied to a large set of scenarios while retaining theoretical guarantees. These properties have propelled MCMC methods into use across a wide range of disciplines. Unfortunately, in some cases of interest the theoretical convergence of MCMC to the posterior can be difficult to attain in practice due to computational costs. Here we present a brief description of MCMC and outline some recent advances used to accelerate its implementation.

1.6.1 MCMC description

As previously mentioned, it is often beneficial in Bayesian settings to sample from the posterior distribution \( p(\theta | \hat{x}, \hat{z}) \) over the parameters. Let us consider a stochastic process \( \theta_1, \theta_2, \theta_3, \ldots \) taking values in the parameter space \( \Theta \). This process is a Markov chain if it satisfies the Markov property

\[
k(\theta_{n+1} | \theta_n, \ldots, \theta_1) = k(\theta_{n+1} | \theta_n),
\]

where \( k(\theta_{n+1} | \theta_n) \) is the transition kernel of the process (the probability density given \( \theta_n \), of transitioning from \( \theta_n \) to \( \theta_{n+1} \)). This property states that \( \theta_{n+1} \) given \( \theta_n \) is independent of all other previous states. The stationary distribution of a Markov chain is a probability measure \( \psi \) such that \( \psi(\theta) \geq 0 \ \forall \theta \in \Theta, \int_{\Theta} d\psi(\theta) = 1 \) and \( \int_B d\psi(\theta) = \int_{\Theta} g(B|\theta)d\psi(\theta) \) for any \( B \subset \Theta \). Here the left hand side is the probability of \( B \), whereas the right side is the probability of transitioning into \( B \) from \( \psi \) in one transition. Their equality implies that \( \psi \) remains constant after transitioning.

Markov chain Monte Carlo methods are a group of algorithms for sampling from a distribution known up to an arbitrary constant of proportionality. This is achieved by simulating stationary ergodic Markov chains for which the target distribution is their stationary distribution. The states visited by the process then form a sample from this distribution and the empirical density of these states converges in distribution to the
A sufficient condition for constructing an appropriate Markov chain is detailed balance, which states that
\[ k(\theta' | \theta) \psi(\theta) = k(\theta | \theta') \psi(\theta') \]
holds for any \( \theta, \theta' \in \Theta \). It can be shown that if \((k, \psi)\) satisfies detailed balance, then \(\psi\) is the stationary distribution of the process. In MCMC methods we construct transition kernels that achieve detailed balance for a given distribution. In particular if we choose the target distribution as the posterior \(\psi(\theta) = p(\theta | \hat{x}, \hat{z})\) then this allows us to draw the required samples. One of the most common approaches to constructing appropriate transition kernels, and the prominent approach of this thesis, is the Metropolis–Hastings algorithm [65, 104].

In the Metropolis–Hastings algorithm we choose a proposal density \(q(\theta_{\text{prop}} | \theta)\) describing the probability of proposing a transition from \(\theta\) to \(\theta_{\text{prop}}\). This proposal is then accepted with probability
\[
A_M(\theta_{\text{prop}}, \theta) = \min \left\{ 1, \frac{q(\theta | \theta_{\text{prop}}) \psi(\theta_{\text{prop}})}{q(\theta_{\text{prop}} | \theta) \psi(\theta)} \right\}.
\]
(1.6.2)

It is readily verifiable that detailed balance is preserved with respect to \(\psi\) if the transitions are carried out this way. The full algorithm can be written as:

**Algorithm 1: Metropolis-Hastings**

Choose initial \(\theta_1 \in \Theta\);

for \(i = 1, 2, \ldots, N_{\text{samples}}\) do

| Propose \(\theta_{\text{prop}} \sim q(\theta_{\text{prop}} | \theta_i)\);

| Sample \(u \sim U(0, 1)\);

| if \(u \leq A_M(\theta_{\text{prop}}, \theta_i)\); then

| | set \(\theta_{i+1} := \theta_{\text{prop}}\) (accept);

| else

| | set \(\theta_{i+1} := \theta_i\) (reject);

end

end

return \(\theta = (\theta_1, \theta_2, \ldots, \theta_{N_{\text{samples}}})\)

Though simple in concept, there are a number of intricacies associated with applying Metropolis–Hastings that must be accounted for. For example, it is important to consider that entries in the Markov chain are typically correlated to subsequent entries, which increases the variance of quantity of interest estimates made from finite samples. Consider the Markov chain for the quantity of interest \(f_i = f(\theta_i)\) where \(\theta_1, \theta_2, \ldots, \theta_{N_{\text{samples}}}\) is an MCMC sample from \(\psi(\theta)\). A function representing how correlated states in the chain are as the number of transitions separating them varies is
the autocorrelation function
\[ \rho_f(\tau) = \frac{\text{Cov}(f_{n+\tau}, f_n)}{\text{Var}(f_n)}. \] (1.6.3)

The values of this autocorrelation function heavily impact the accuracy of estimates derived from MCMC samples. To see why this happens, suppose we estimate the posterior mean of \( f \) using MCMC samples by applying the empirical mean formula
\[ \mathbb{E}_{\psi(\theta)}[f(\theta)] \approx \frac{1}{N_{\text{samples}}} \sum_{n=1}^{N_{\text{samples}}} f_i. \] (1.6.4)

The estimator on the right of (1.6.4) is a random variable with mean \( \mathbb{E}_{\psi(\theta)}[f(\theta)] \) and variance
\[ \frac{\tau_f}{N_{\text{samples}}} \text{Var}_{\psi(\theta)}(f(\theta)), \] (1.6.5)

where \( \tau_f \) is the integrated autocorrelation time of the Markov chain \( f_1, f_2, \ldots, f_{N_{\text{samples}}} \)
\[ \tau_f = 1 + 2 \sum_{\tau=1}^{N_{\text{samples}}} \rho_f(\tau). \] (1.6.6)

If subsequent samples are independent of one another, then there is no autocorrelation \( (\rho_f(\tau) = 0 \text{ for } \tau \geq 1) \), and so the variance (1.6.5) is the usual variance of a Monte Carlo estimator of the mean based on i.i.d samples. However as Metropolis–Hastings samples are generally correlated, \( \tau_f > 1 \) is typical, resulting in higher variance quantity of interest estimates and lower accuracy. The effective sample size is given by \( N_{\text{samples}}/\tau_f \). This represents the number of i.i.d samples that would be required to produce an estimator with the same variance as the estimator produced by the MCMC sample [54].

In practice the autocorrelation, integrated autocorrelation time, and effective sample size, are estimated empirically from the MCMC samples. These estimates are then used to guide diagnostics. More detailed information on these estimations and how they are used is given in [160].

Another intricacy associated with MCMC is that the initial state of the Markov chain must be manually specified. This is typically done without detailed knowledge of the true posterior, and so this initial state is often unrepresentative of a sample from it.
Coupled with the fact that subsequent states in the chain are often correlated, this can result in an early sampling phase where the Markov chain remains biased towards the initial state as it transitions towards the target distribution. To account for this initial “burn-in” phase, these early samples are discarded from the overall sample, thus removing the bias towards the initial state and leaving a more representative sample from the target distribution.

One possible way to quantify the efficiency of an MCMC scheme is to measure its effective sample size per second. The remainder of this section is devoted to introducing some breakthroughs and recent works that have been applied to optimise this. We cover two major sources of computational cost in MCMC: the number of iterations required, and the average cost of executing an iteration. To motivate the first, consider two Markov chains with the same stationary distribution that undergo the same number of transitions. On average, the Markov chain whose auto-correlation function decays faster produces a larger effective sample size, resulting in a more accurate prediction. In Metropolis–Hastings, the proposal distribution \( q(\theta_{\text{prop}}|\theta) \) is the key factor influencing the auto-correlation of a Markov chain. Improving this proposal distribution has been the focus of much research, and we introduce some breakthroughs and recent developments in this area in Section 1.6.3. Alternatively, if we consider two MCMC schemes with the same target distribution and proposal distribution, then the chain that is run for more iterations will typically produce a larger effective sample size. This has motivated the development of reduced cost methods which allow more iterations to be performed within a fixed time. We outline some recent work in this direction for expensive likelihoods in Section 1.6.2.

1.6.2 Reducing the average cost of an iteration

1.6.2.1 Approximate likelihood models

In Algorithm 1, the likelihood is evaluated every iteration within the computation of the acceptance probability. As previously highlighted this is particularly costly in scenarios when the evaluation of this likelihood involves an expensive numerical solver. Cheaper approximations to this likelihood are often employed to reduce this cost. Take for example a surrogate approximation \( \hat{u}(x; \alpha) \) of the true parametric forward map \( u(x; \alpha) \) constructed using an approach described in Section 1.5.2. Replacing the full model with this surrogate in the likelihood induces a surrogate posterior
\( \hat{\psi}(\boldsymbol{\theta}) \approx \psi(\boldsymbol{\theta}) \) that can be evaluated more quickly than the full model. Alternatively, instead of approximating the forward solution, some approaches opt to construct a surrogate that emulates the log-likelihood or the unnormalised target distribution itself as a function of \( \boldsymbol{\theta} \). This is still sufficient for performing MCMC, and is often simpler, as it only requires that we interpolate over real valued quantities rather than potentially high-dimensional solution fields.

Many approaches exist to construct appropriate surrogates. Recently stochastic Galerkin expansions were applied in [137] to infer heat transfer coefficients, though being a linear decomposition of the parametric solution, this approach suffers the curse of dimensionality and is thus restricted to lower-dimensional problems. Deep neural network surrogates have been trained to interpolate solution data generated by numerical solvers, for example in [15, 85]. These techniques are effective, though a significant limitation is the requirement to numerically evaluate the forward model many times in order to generate data for training. Recent neural network approaches have overcome this limitation by training using a physics-informed objective, which encourages the satisfaction of physical equations without labelled data [38].

Further optimisations to the surrogate approach have been explored in the form of adaptive schemes. These approaches construct the surrogate online during MCMC by using the posterior samples to guide local refinements to the surrogate on the fly. In local approximation MCMC [33], an ensemble of samples from the posterior are generated using MCMC, and their associated forward model evaluated. These are then re-used in subsequent MCMC iterations where the likelihoods of out-of-sample proposals are approximated using local elements of the ensemble. If, for a proposed MCMC sample, there are no existing samples in the local region of the parameter space, then the forward map is evaluated for this proposal and it is added to the ensemble. This constitutes a refinement of the surrogate. If, as the surrogate is refined it converges pointwise to the full model, then this approach to MCMC converges to the true target distribution. The efficiency of this approach therefore depends mostly on the number of refinements (and therefore full numerical solves) needed to construct an accurate surrogate.

1.6.2.2 Multi-step acceptance rules

Some common methods of constructing surrogates do not converge to the full model under refinement, for example a neural network surrogate with a finite number of pa-
rameters cannot in general match an arbitrary solution everywhere, regardless of how long it is trained. Such an approximate model can still prove valuable in MCMC while maintaining detailed balance with respect to the posterior induced by the full model through the incorporation of a multi-step acceptance rule. These multi-step methods use a surrogate informed proposal distribution to reduce the number of full model evaluations that are executed, thus accelerating MCMC.

To describe these methods, we denote the posterior distribution induced by the surrogate by \( \hat{\psi}(\theta) \), and the true posterior by \( \psi(\theta) \). In a multi-step acceptance rule scheme we sample from \( \hat{\psi}(\theta) \) using Metropolis–Hastings with proposal distribution given by

\[
q(\theta_{prop}|\theta) = A^*(\theta_{prop}, \theta)q^*(\theta_{prop}|\theta) + \left(1 - \int A^*(\theta_{prop}, \theta)q^*(\theta_{prop}|\theta)\,d\theta_{prop}\right)\delta(\theta - \theta_{prop}).
\]

Here \( q^*(\theta_{prop}, \theta) \) is some specified proposal distribution, and \( A^*(\theta_{prop}, \theta) \) is the corresponding acceptance probability applied using \( \hat{\psi}(\theta) \).

A proposal from this distribution is naturally produced as the subsequent state of a nested MCMC iteration executed using the cheaper surrogate posterior \( \hat{\psi}(\theta) \). In this nested iteration, a proposal is made from \( q^*(\theta_{prop}, \theta) \) and accepted with standard acceptance criteria \( [1.6.2] \) using \( \hat{\psi}(\theta) \). The resulting state is a sample from \( [1.6.7] \) which acts as an MCMC proposal for the posterior \( \psi(\theta) \) induced by the full model. This is then accepted or rejected as usual using Metropolis acceptance probability \( [1.6.2] \) applied with \( \psi(\theta) \). Crucially, if the outcome of the nested iteration is rejection then the proposal to \( \psi(\theta) \) is the same as the current state, therefore the full model is not required to be evaluated as acceptance is guaranteed. Often in MCMC the proportion of rejected proposals is very high, and so the cheap rejection of many of these proposals in the nested iteration can provide a significant cost saving while still maintaining convergence to the true posterior.

There are a number of variations to this procedure. The method as it is described here is known as the delayed-acceptance method \( [29] \). This can be applied in scenarios where the surrogate is dependent on the current state of the Markov chain. If the surrogate approximation is not state-dependent the delayed-acceptance scheme coincides with an earlier multi-step acceptance scheme known as the surrogate transition method
In this case the acceptance criteria for $\psi(\theta)$ reduces to
\[
A(\theta_{\text{prop}}, \theta) = \min \left\{ 1, \frac{\psi(\theta_{\text{prop}}) \hat{\psi}(\theta)}{\psi(\theta) \hat{\psi}(\theta_{\text{prop}})} \right\}.
\] (1.6.8)

One can verify this by substituting (1.6.7) into the standard Metropolis acceptance criteria (1.6.2) and simplifying (see for example [46] for the full proof). Further extensions to these multi-step methods allow for n-step surrogate proposals, which attempt to decorrelate the proposed samples of $\psi(\theta)$ by carrying out multiple cheap MCMC transitions using $\hat{\psi}(\theta)$ between each full model evaluation. Multi-level approaches have also been proposed [41] that recursively apply a hierarchy of model refinements in order to achieve the variance reduction in estimates of posterior quantities of interest that is associated with multi-level Monte Carlo approaches [56].

1.6.3 Reducing the required number of iterations

1.6.3.1 MALA and HMC

As previously mentioned, the proposal distribution $q(\theta_{\text{prop}}|\theta)$ in Metropolis–Hastings is crucial in determining how correlated nearby MCMC samples are, and as a consequence $q(\theta_{\text{prop}}|\theta)$ determines the number of iterations that are required to achieve accurate posterior quantity of interest estimates.

Perhaps the simplest choice of proposal distribution is employed in the random walk Metropolis–Hastings (RWMH) scheme, where $q(\theta_{\text{prop}}|\theta)$ is set independently of the target distribution, for example to a multivariate normal distribution centred at $\theta$ with fixed covariance. This turns out to be inefficient in most cases of interest as the random nature of these proposals results in a strong chance that they correspond to lower probability parameters under the target distribution. This results in a large number of rejections or low variance proposals that give Markov chains with high auto-correlation.

To improve upon this, techniques which use information from the target distribution have been developed. The guiding principle in these schemes is to target regions of higher probability according to $\psi(\theta)$. Notably, techniques which allow $q(\theta_{\text{prop}}|\theta)$ to depend on the gradients of $\psi(\theta)$ have been shown to dramatically improve upon random walk based proposals. To convey this idea we outline the Metropolis adjusted Langevin (MALA) and Hamiltonian Monte Carlo (HMC) methods. These methods
(particularly HMC and its extensions \[14 \hspace{1pt} 111\]) have been implemented into widely used packages \[29 \hspace{1pt} 139\] for Bayesian analysis in industrial and academic environments. An implementation demonstrating the relative effectiveness of RWMH, MALA and HMC is given in Chapter 3.

MALA

The proposal strategy in MALA is motivated by the Langevin dynamics with stationary distribution $\psi(\theta)$. These dynamics can be written

$$d\theta_t = \nabla \log \psi(\theta_t) dt + \sqrt{2} dW_t.$$  \hspace{1pt} (1.6.9)

To generate proposals using these dynamics the Euler–Maruyama discretisation of these dynamics is applied

$$\theta_{n+1} = \theta_n + \tau \nabla \log \psi(\theta_n) + \sqrt{2\tau} \xi_n,$$  \hspace{1pt} (1.6.10)

where $\tau$ is a discretisation length-scale and $\xi_n \sim \mathcal{N}(0, I)$. In other words the proposal distribution is $q(\theta_{\text{prop}}|\theta) = \mathcal{M}V\mathcal{N}(\theta + \tau \nabla \log \psi(\theta), 2\tau I)$. From this we can see that the mean of the proposal distribution is located in the direction of increasing probability, thus MALA proposals target areas of higher probability which results in a higher acceptance probability. As the Euler–Maruyama integrator is an inexact approximation to the Langevin dynamics, this proposal is followed by the standard Metropolis acceptance criteria to ensure that detailed balance is retained with respect to $\psi(\theta)$. In comparison to RWMH, MALA has demonstrated orders of magnitude improvements in sampling efficiency. Generalisations of this approach include preconditioned MALA \[58\], which can adapt to non-isotropic target distributions through a linear transformation, and Riemannian manifold MALA \[58\] which does so locally by allowing this preconditioning to depend on the current state.

HMC

In Hamiltonian Monte Carlo, proposals arise as a result of Hamiltonian trajectories governed by the target distribution. To give a brief overview, in HMC each parameter $\theta_i$ is viewed as a positional coordinate of a Hamiltonian trajectory, and appended with a corresponding momentum variable $p_i$. The target density $\psi(\theta)$ is also augmented with momentum variables, resulting in an expanded target distribution $\psi(\theta, p) = \psi(p|\theta)\psi(\theta)$
over phase space. To generate a HMC proposal from this expanded target distribution, Hamilton’s equations

\[
\begin{align*}
\frac{d\theta_i}{dt} &= \frac{\partial H}{\partial p_i}, \\
\frac{dp_i}{dt} &= \frac{\partial H}{\partial \theta_i}.
\end{align*}
\]

are simulated numerically using symplectic integrators, with the Hamiltonian function set according to the target distribution as

\[
H(\theta, p) = -\log \psi(\theta) - \log \psi(p|\theta) = U(\theta) + K(p|\theta). \tag{1.6.11}
\]

(1.6.12)

Here the potential energy is the negative log of the target \(U(\theta) = -\log \psi(\theta)\), and \(K(p|\theta) = -\log \psi(p|\theta)\) represents the kinetic energy of the system. The terminal position of this simulation constitutes a proposal, which is then assessed as usual by the Metropolis acceptance criteria. The momentum coordinates of the sample can then be discarded to leave a sample from \(\psi(\theta)\), since the original target is the marginal of the expanded target \(\psi(\theta) = \int \psi(\theta, p)dp\).

In a sense HMC can be thought of as an extension of MALA, since HMC reduces to MALA when the Hamiltonian trajectories are simulated using a single integration step. However HMC proposals generated using multiple integration steps give rise to longer Hamiltonian trajectories governed by the target distribution. These proposals can cover large distances in phase space, resulting in a significant decorrelation of proposals when compared to RWMH or MALA, and thus more efficient sampling is achieved. Extensions and generalisations of this method also include a Riemannian manifold variant \[58\], as well as the widely used ‘No-U-turn’ sampler which reduces redundant simulation effort by applying an appropriate early termination criteria to the Hamiltonian trajectories \[71\].

1.6.3.2 Optimal transport map proposals

Approaches to efficiently sample from the posterior distribution have also been proposed using optimal transport. Most commonly these have been based on the transportation of samples between a reference measure \(\psi_{Ref}(r)\) (e.g. a standard normal distribution) and a target measure \(\psi(\theta)\) by means of an invertible transport map \(T : \mathbb{R}^M \rightarrow \mathbb{R}^M\). The
The purpose of this map is to enforce that \( \psi_{\text{Ref}}(T^{-1}(A)) = \psi(A) \) is satisfied for any Borel set \( A \in \mathbb{R}^M \) (i.e. the pushforward of the reference measure is the target, \( T#\psi_{\text{Ref}} = \psi \)), and this is achieved approximately by solving an appropriate optimisation problem, for example the minimisation of KL divergence between the target and transported reference \([17, 97]\). Once this map is constructed, sampling from the target follows by drawing samples from the reference and mapping them through the transport map.

In optimal transport map proposals MCMC \([117]\), transport maps are used to produce more efficient Metropolis–Hastings proposals for challenging target distributions. The idea here is to construct a transport map from the target distribution \( \psi \) to a pushforward distribution \( T#\psi \) that approximates some simple reference distribution \( (T#\psi \approx \psi_{\text{Ref}}) \). In this way, more efficient Metropolis–Hastings samples can be produced on the target measure by proposing on \( T#\psi \) (which is closer to a reference that is chosen to be easy to sample efficiently from) and mapping those proposals to the target, thus ensuring detailed balance of \( \psi \). The KL divergence objective used to construct the transport map is estimated using samples from the target, and the approximation of the transport map becomes more accurate as more samples are drawn. Contrary to the previous approaches, this method does not require gradients of the target distribution if gradient-free proposals are applied to \( T#\psi \). However, gradient based proposals can be applied here to improve efficiency if the corresponding gradients of the target measure are available.
1.7 Outline and key contributions of the thesis

In this introduction we have given a brief overview of Bayesian inference, indicated why it is a beneficial approach to pursue, and described some methods by which it can be performed. We have described common solution approaches for PDEs and integral equations, and pointed out how it is inefficient to naively apply these within methods such as MCMC and gradient based optimisation. After introducing some deep learning preliminaries, some established methods for accelerating Bayesian inference for PDE problems beyond this naive approach were also presented. We reiterate that this section is not designed to be exhaustive, and acknowledge that there exist countless other approaches to solving Bayesian problems with physical likelihoods. The purpose of this section is to introduce the concepts needed to motivate this work, and to describe some established methods so that they can be compared to the techniques we develop in the subsequent chapters. To put our contributions in context with the approaches previously described; our methods reduce the cost of an MCMC iteration by using deep learning surrogates to cheaply emulate physical likelihoods, and guarantee the accuracy of our inferences by utilising these surrogates within multi-step acceptance schemes. To reduce the surrogate construction cost, an adaptive approach to training these surrogates online based on existing MCMC samples is described. This significantly outperforms the analogous non-adaptive approach in training time and approximation accuracy. Finally, our approach reduces the number of required iterations by applying gradient based MCMC proposal schemes. In particular, we note that our approach has the distinct advantages of being efficient to train over high dimensional parameter spaces without any training data or external solution algorithms, and that our deep learning approach is particularly amenable to efficient gradient based MCMC proposals in settings with physics-governed likelihoods. This is because our approach uses automatic differentiation of the surrogate to compute the relevant gradients, whereas more traditional approaches would require numerical solutions of the adjoint equation. Having introduced the above-stated background material, we now state the contributions of this thesis in more detail.

In Chapter 2 we introduce the deep surrogate model, which is a neural network that emulates the solution of a PDE given the inputs $x$ and parameters $\theta$ as arguments. This network is trained in an unsupervised manner, by extending the training regime described in the Deep Galerkin Method into the parameter space, thus no additional numerical solves are required. The surrogate is then utilised within an MCMC scheme to sample from the Bayesian posterior distribution. To the authors’ knowledge
this work is the first time that deep learning surrogates of this type have been applied to solve Bayesian inverse problems. A second contribution of the chapter is the introduction of an unsupervised deep learning solution method for integral equations. This method is also extended into the parameter space thus constructing surrogate models, which are then utilised to sample from the Bayesian posterior distribution associated with integral equation models.

Chapter 3 is focused on an inverse problem where spatio-temporal heat fluxes are sought from temperature measurements in compressor cavities [39, 148]. This topic was motivated by colleagues in the Department of Mechanical Engineering, who collected data using an experimental compressor cavity rig. The contribution concerning this application is a full Bayesian uncertainty quantification of the spatio-temporal heat fluxes in a range of real experimental settings. Accuracy is extremely important for this application, and so mathematically the contributions of this chapter focus on extending the framework of Chapter 2 such that we can specify interpretable priors and achieve efficient posterior samples with guaranteed accuracy. To achieve this we describe how to apply the deep surrogate method with Gaussian process priors, use differentiability of neural networks to apply gradient based proposals for more efficient MCMC, increase the accuracy and efficiency of the surrogate through an adaptive training approach that weights the training measure by the posterior probability, and guarantee the accuracy of the posterior using a delayed-acceptance scheme with a more traditional solver. These factors are all combined in a scheme that leverages the efficiency of the surrogate to make decorrelated proposals, while the error bounds associated with more traditional solvers guarantee the accuracy of the sample. A range of numerical results are given to validate the effectiveness of this approach and convergence of the posterior error to zero is derived in the Hellinger metric.

In Chapter 4 we present some preliminary results where the deep surrogate methodology is applied to an air pollution prediction problem. In this problem, neural network approximations to pollution transport equations are included as physical features in a Bayesian spatio-temporal statistical model. Some air pollution data from Ulaanbaatar, Mongolia, is presented, and we produce a posterior distribution jointly describing the air pollution in the city in space and time. The contributions of this chapter are a full Bayesian analysis of this problem, that uses a custom Hamiltonian Monte Carlo sampler to simultaneously fit the physical model and an additive Gaussian process term. This is in contrast to common approaches applied in spatial statistics, where physical models are typically fit in advance, and a statistical treatment of the uncertainties and
the regression of unresolved behaviour is performed as a separate post-processing step. The computation of a joint posterior distribution over the physical model and Gaussian process terms ensures that both sources of uncertainty and their dependence on one another is captured. A further contribution lies in the data that is being analysed. This air pollution data is sparsely located, and purely data-driven approaches to understanding it give poor results. Our physics-informed approach sheds more light on the behaviour of the pollution in Ulaanbaatar, including systematic details such as how the pollution source intensity varies with space and time at different temperatures. We also describe some ideas that could be implemented in order to enhance this approach in future, including the description of a neural network construction that guarantees that the mass of the PDE solver is controlled appropriately, and the incorporation of further meteorological factors into the model such as a direct dependence on temperature or variable wind.

We conclude in Chapter 5 by providing a summary and discussion of the results and contributions of the thesis, before finally ending with some closing remarks.
Chapter 2

A deep surrogate approach to efficient Bayesian inversion in PDE and integral equation models

This chapter is reproduced from the authors’ manuscript currently under review [38].

Here, we introduce a deep learning approach for conducting Bayesian inference for PDEs and integral equations. This method uses an unsupervised training approach to construct neural network approximations to parametric PDEs and integral equations. These approximations are used as surrogate models, which can be evaluated rapidly, and therefore leveraged in Bayesian inverse uncertainty quantification settings. We train these models and demonstrate their effectiveness in simulation studies covering PDE and integral equation examples.

There are two major contributions of this chapter. The first is the development of an unsupervised deep learning solution method for parametric integral equations. The second is the use of deep learning surrogates to solve Bayesian inverse problems, which to the authors’ knowledge, is the first application of unsupervised deep learning surrogate methodology to this setting. We demonstrate that the surrogate models developed in this chapter result in a speed-up of several orders of magnitude when compared with traditional discretisation schemes, while maintaining comparable accuracy.
This declaration concerns the article entitled:

A deep surrogate approach to efficient Bayesian inversion in PDE and integral equation models.

<table>
<thead>
<tr>
<th>Publication status (tick one)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft manuscript</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Publication details (reference)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Copyright status (tick the appropriate statement)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I hold the copyright for this material ✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Candidate’s contribution to the paper (provide details, and also indicate as a percentage)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The candidate predominantly executed the…</td>
</tr>
<tr>
<td>Formulation of ideas:</td>
</tr>
<tr>
<td>100%. The author of this thesis developed the project concept independently.</td>
</tr>
<tr>
<td>Design of methodology:</td>
</tr>
<tr>
<td>100%. The author of this thesis devised the methodology.</td>
</tr>
<tr>
<td>Experimental work:</td>
</tr>
<tr>
<td>100%. All code was developed and implemented by the author of this thesis.</td>
</tr>
<tr>
<td>Presentation of data in journal format:</td>
</tr>
<tr>
<td>90%. The author of this thesis produced all results and drafts of the manuscript. Feedback was provided by all other authors.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statement from Candidate</th>
</tr>
</thead>
<tbody>
<tr>
<td>This paper reports on original research I conducted during the period of my Higher Degree by Research candidature.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Signed</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teo Deveney</td>
<td>05/06/2022</td>
</tr>
</tbody>
</table>
Abstract

We investigate a deep learning approach to efficiently perform Bayesian inference in partial differential equation (PDE) and integral equation models over potentially high-dimensional parameter spaces. The contributions of this paper are two-fold; the first is the introduction of a neural network approach to approximating the solutions of Fredholm and Volterra integral equations of the first and second kind. The second is the development of a new, efficient deep learning-based method for Bayesian inversion applied to problems that can be described by PDEs or integral equations. To achieve this we introduce a surrogate model, and demonstrate how this allows efficient sampling from a Bayesian posterior distribution in which the likelihood depends on the solutions of PDEs or integral equations. Our method relies on the direct approximation of parametric solutions by neural networks, without need of traditional numerical solves. This deep learning approach allows the accurate and efficient approximation of parametric solutions in significantly higher dimensions than is possible using classical discretisation schemes. Since the approximated solutions can be cheaply evaluated, the solutions of Bayesian inverse problems over large parameter spaces are efficient using Markov chain Monte Carlo. We demonstrate the performance of our method using two real-world examples; these include Bayesian inference in the PDE and integral equation case for an example from electrochemistry, and Bayesian inference of a function-valued heat-transfer parameter with applications in aviation.
2.1 Introduction

Deep learning methodologies have seen significant development in the past few decades. The advent of efficient optimisation algorithms has led to impressive results in many high-dimensional tasks, particularly in settings where there is an abundance of data. Recently there has been much focus on the use of deep learning to address mathematical challenges involving differential equations. In the short time of just a few years, this has led to much progress, both in algorithmic ingenuity and theoretical understanding of neural network approaches to specific mathematical problems. For example, we now have proof that deep neural networks can overcome the curse of dimensionality in the approximation of the solutions of certain classes of PDEs [11, 60, 78], and efficient algorithms capable of generating these approximations have been identified [44, 64, 125, 144]. These methods have already found applications in a range of areas such as fluid dynamics [126] and financial mathematics [10, 164]. Further exploration of these methods is well underway, with applications including optimal control problems using the Hamilton-Jacobi-Bellman equation [74, 109], and PDE-based regularisation in regression tasks for which the governing laws are partially known [108]. In forward uncertainty quantification, surrogate models and probabilistic generative models have been implemented to capture response uncertainty of PDEs with random terms [107, 165].

While deep learning approximations for PDE-based problems have received a lot of attention, integral equations [124] are less well investigated, despite also having widespread applications in areas such as radiative transfer, viscoelasticity, and electrochemistry. This paper extends previous work on the solution of PDE problems to integral equations, by introducing a new method for solving integral equations using neural networks. Furthermore, existing approaches for PDEs focus mostly on the solution of these equations and forward uncertainty quantification. The second novelty of this paper is the application of deep learning to Bayesian inverse problems for which the likelihood function depends on the solution of a PDE or integral equation. Our method is based on the direct approximation to the solution of the parametric forward problem by deep surrogate models. In both the PDE and integral equation case, our surrogates are analytic, easily differentiable, and allow for rapid approximate evaluations of the solution for different model parameters without need of further numerical solves. This makes the full library of existing parametric statistics techniques accessible for these physics-based problems without modification. Using these surrogates we apply Markov chain Monte Carlo (MCMC) to rapidly sample from the Bayesian posterior distribution of uncertain
parameters given data and a prior distribution. This is a novel application of parametric forward solvers based on deep neural networks, which achieves a significant speed-up when compared to a more traditional finite difference MCMC approach, without sacrificing accuracy. Since deep neural networks overcome the curse of dimensionality, our approach is efficient and accurate for high-dimensional problems or problems with a large parameter space, for which the repeated approximation of the forward problem becomes computationally intractable using more traditional discretisation schemes.

Multiple alternative approaches have been devised to improve the efficiency of MCMC methods for inverse problems, such as the use of inexpensive coarse scale models as pre-conditioners for the fine scale model \cite{45, 46}, and multilevel techniques \cite{41}. These methods can reduce the number of expensive numerical solves of the fine scale problem that result in rejected proposals in cases where the solutions at the coarser levels are closely correlated to the true solution, however a large number of fine scale solves are still required for sufficient new proposals to be accepted. Surrogates based on approximations to the parametric forward problem have also been employed. A common approach is to use a generalised polynomial chaos representation constructed by a stochastic Galerkin projection \cite{55, 161}. In this scheme a orthogonal polynomial decomposition is used to approximate the parametric solution, however this becomes intractable in higher dimensions due to the curse of dimensionality of the polynomial approximation to the solution. For this reason, the approach has been restricted to applications with few model parameters \cite{66, 137}. Collocation based surrogates have been applied in cases where the number of parameters is larger (though typically less than 10). In this case, more expensive traditional discretisation schemes are employed to generate an ensemble of solutions for a range of parameter values, and these are interpolated to construct the surrogate \cite{101, 103}. This approach requires a large number of expensive solves, and relies on interpolants which may not satisfy the required equations in complicated regions of the domain \cite{165}. While all of these methods have all been successful in accelerating MCMC for various applications, each is restricted either by the dimensionality of the problems that they can be applied to, or the accuracy of the surrogates that they construct. Our approach overcomes these issues by using a deep learning approximation that is accurate in high dimensions and rapid to evaluate.

The key contributions of this paper are as follows:

- We introduce a new deep learning approach for the solution of parametric Fredholm and Volterra integral equations of the first and second kind. By capturing
the dependence on model parameters, our method provides an efficient approximation of the parametric forward problem for high-dimensional integral equation parameter spaces.

- We describe how neural network surrogate models for PDE and integral equation based problems can be used to rapidly sample from the posterior distribution in a Bayesian setting. In particular, we argue that our approach is efficient even if the problem is high-dimensional or depends on a large number of parameters.

- By considering a specific example from electrochemistry, we demonstrate that our deep surrogate integral-equation solvers can be used to infer parameters in a Bayesian setting with an accuracy that is comparable to current PDE-based deep learning methods.

- To demonstrate that the method can efficiently perform Bayesian inversion in a high-dimensional parameter space, we apply our deep surrogate solver to infer a parametric function in the PDE for heat transfer in aircraft turbines.

This paper is organised as follows: In Section 2.2, we describe the general form of the PDE and integral equation models, and define the deterministic and parametric forward problem that will be considered. Here we also introduce the statistical model that form the basis of the Bayesian inverse problem [147]. In Section 2.3, we review the physics-informed neural network [127] and the deep Galerkin method [144] for solving the PDE forward problem, and describe an extension of these methods to the accurate approximation of the parametric forward problem by a deep surrogate model. In Section 2.4, we introduce a novel deep learning method for the solution of integral equations and outline its extension to parametric integral equation problems using a deep surrogate model. In Section 2.5, we explain how to apply a MCMC scheme that uses deep surrogate model evaluations to significantly reduce the computational cost of sampling from the posterior distribution. Finally in Section 2.6 we demonstrate how the deep surrogate method can accelerate Bayesian inference by applying our method to two problems with real applications in electrochemistry and heat transfer in aircraft turbines. These examples cover PDE-based problems and integral equations, with inferred parameters including scalar values and functions. Python code is available to reproduce all of the presented examples [37].
2.2 Problem specification

We begin by defining the general form of the PDE and integral equation problems that will be investigated in this paper. This includes a description of the deterministic and parametric forward problems for these models, as well as the inverse problem corresponding to these models.

2.2.1 Forward problem

First consider the parameterised scalar-valued PDE,

\[ N(x, u(x; \theta); \theta_N) = h(x; \theta_h), \quad x \in \Omega, \theta \in \Theta, \] \hspace{1cm} (2.2.1)

with boundary conditions defined by

\[ u(x; \theta) = b(x; \theta_b), \quad x \in \partial \Omega, \theta \in \Theta. \] \hspace{1cm} (2.2.2)

Here \( \Theta \subset \mathbb{R}^p \) denotes the parameter space and \( \theta = (\theta_N, \theta_h, \theta_b) \in \Theta \) is a parameter vector. The input space is denoted \( \Omega \subset \mathbb{R}^d \), where \( d \) is the dimension of the PDE domain \( \Omega \). We assume that \( N \) is a known (possibly non-linear) differential operator parameterised by \( \theta_N \), and \( h, b \) are known functions parameterised by \( \theta_h \) and \( \theta_b \). If \( \theta \) is fixed this is a deterministic PDE, and the deterministic forward problem is to approximate the \( d \)-dimensional function \( u_\theta(x) : \Omega \rightarrow \mathbb{R} \) satisfying (2.2.1) given \( \theta \). Here we have used \( u_\theta(x) \) in place of \( u(x; \theta) \) as the solution to (2.2.1) to signify that the parameters \( \theta \) are fixed, and therefore not inputs of the solution. This convention is extended to all functions and operators throughout this work, whereby \( \theta \) is placed into their subscript when it is fixed. The parametric forward problem is to produce a function capable of returning the solution corresponding to any parameters \( \theta \in \Theta \), that is to approximate the \( d+p \)-dimensional function \( u(x; \theta) : \Omega \times \Theta \rightarrow \mathbb{R} \) satisfying (2.2.1).

In this work, we also consider non-homogenous integral equations of the first kind

\[ 0 = v(x; \theta_v) + \int_a^{b(x)} k(x, y; \theta_k)u(y; \theta)dy, \quad x \in \Omega, \theta \in \Theta, \] \hspace{1cm} (2.2.3)
and the second kind

\[ u(x; \theta) = v(x; \theta_v) + \int_a^{b(x)} k(x, y; \theta_k) u(y; \theta) dy, \quad x \in \Omega, \theta \in \Theta. \] (2.2.4)

Again \( \Theta \subset \mathbb{R}^p \) denotes the parameter space and \( \theta = (\theta_v, \theta_k) \in \Theta \) is a parameter vector. The input space is \( \Omega = [a, b^*] \) for some \( a < b^* \). We assume \( v(x; \theta_v) \) is a given function parameterised by \( \theta_v \), and \( k(x, y; \theta_k) \) is a known kernel parameterised by \( \theta_k \). The deterministic forward problem is to approximate the function \( u_\theta(x) : \Omega \to \mathbb{R} \) satisfying the integral equation for a fixed \( \theta \). The parametric forward problem is to approximate \( p + 1 \)-dimensional function \( u(x; \theta) : \Omega \times \Theta \to \mathbb{R} \) satisfying the equation. Our deep learning algorithm can solve the parametric forward problems (2.2.3) and (2.2.4) for both Fredholm equations \( (b(x) = b^*) \), and Volterra equations \( (b(x) = x) \).

### 2.2.2 Inverse problem

The goal of the inverse problem for both the PDE and integral equation is to estimate \( \theta \) given data \( (\hat{x}_i, \hat{z}_i)_{i=1:M} \) observed from some real physical process. This data consists of input data \( \hat{x}_i \in \Omega \) and measured responses \( \hat{z}_i \in \mathbb{R} \) corresponding to those inputs. We model the data as the solution of the equation with an additional error term

\[ \hat{z}_i = u(\hat{x}_i; \theta) + \epsilon_i, \quad i = 1, \ldots, M. \] (2.2.5)

This is similar to the common scenario in parametric statistics

\[ \hat{z}_i = f(\hat{x}_i; \theta) + \epsilon_i, \quad i = 1, \ldots, M. \] (2.2.6)

In statistical applications \( f(x; \theta) \) is typically a carefully constructed parametric function, designed using specialist knowledge to take input variables \( x_i \) and produce outputs \( z_i \) which model the mean behaviour of the system. In our case this function is the parametric solution \( u(\hat{x}_i; \theta) \), therefore it behaves according to the laws imposed by the PDE or integral equation. The last term in (2.2.5) is a random variable which accounts for deviation of the data from the model. A standard choice for the \( \epsilon_i \), which we will adopt, is to assume unbiased i.i.d. Gaussian deviations of the form \( \epsilon_i = N(0, \sigma^2) \), where \( \sigma^2 \) is an unknown parameter.

The objective in the inverse problem is the same as a statistical regression problem, that is to infer the unknown parameters \( \{\theta, \sigma^2\} \) such that the model describes the data.
We note that using our deep surrogate methodology, the inverse problem can be solved with any parametric regression scheme from statistics without modification, since the deep surrogate model provides a functional representation of the solution $u(x, \theta)$ that can be rapidly evaluated and differentiated with respect to the parameters $\theta$. In this work we focus on applying MCMC methods from Bayesian parametric regression to solve Bayesian inverse problems for which the forward problem can be described by a PDE and integral equation. This provides a natural form of inverse uncertainty quantification in the form of a posterior probability distribution over all parameters of interest, but has proven too expensive to be implemented fully in the majority of cases with standard numerical discretisation techniques such as finite differences or quadrature methods.

2.3 Review of deep learning techniques for PDEs

Neural networks are capable of accurately approximating high-dimensional functions using significantly fewer parameters than traditional function approximations based on linear basis expansions. For multiple classes of PDEs it can be proven that the number of parameters required to approximate the solution to a fixed accuracy grows at most polynomially with the dimension of the problem \cite{11, 50, 78}, in contrast to exponentially for techniques like finite differences and finite elements. Because of this, deep learning techniques have shown great success in the solutions of PDE forward problems with dimensions of up to $d = 200$ \cite{44, 64, 125, 144}, which would have been intractable to solve with traditional discretisation approaches. Here we briefly review the physics-informed neural network and deep Galerkin methods of approximating deterministic PDE solutions by neural networks. We also describe an extension of these methods to approximate the solution to the parametric forward problem which will be used in our Bayesian approach to inverse problems.

2.3.1 Deterministic forward problem

We first consider the neural network approximation of the deterministic PDE forward problem defined by (2.2.1)-(2.2.2). Recent work in deep learning for PDEs \cite{127, 144} has uncovered a simple method designed to solve such equations over a closed domain $\Omega$. Let $\pi^d$, and $\pi^b$ be measures with supports $\text{supp}(\pi^d) = \Omega$ and $\text{supp}(\pi^b) = \partial \Omega$. Under the assumption that the PDE is well-posed, the approach in \cite{127, 144} exploits the fact
that for a fixed $\theta \in \Theta$ and $\nu_1, \nu_2 > 0$ the quantity
\[
\nu_1 \| N_\theta(u_\theta(x), x) - h_\theta(x) \|^2_{L_2(\Omega, \pi^d)} + \nu_2 \| u_\theta(x) - b_\theta(x) \|^2_{L_2(\partial \Omega, \pi^b)}.
\]

is non-negative, and zero only when $u_\theta(x)$ is the PDE solution. Here the subscript $L_2(D, \mu)$ denotes the $L_2$-norm over domain $D$ with respect to the measure $\mu$
\[
\| f(x) \|^2_{L_2(D, \mu)} = \int_D |f(x)|^2 d\mu.
\]

The universal approximation theorem [35] states that neural networks are dense in the space of continuous functions. It follows that if a solution exists and is continuous, then it can be approximated arbitrarily well by a neural network. The deep learning methods described in [127, 144] use neural networks to approximate the solutions of PDEs. Denoting the neural network approximation as $\hat{u}_\theta(x) \approx u_\theta(x)$, these algorithms take inspiration from (2.3.1) by using some variant of the gradient descent scheme to update the neural network parameters to minimise the loss function
\[
\frac{\nu_1}{N} \sum_{n=1}^{N} (N_\theta(\hat{u}_\theta(x^n), x^n) - h_\theta(x^n))^2 + \frac{\nu_2}{J} \sum_{j=1}^{J} (\hat{u}_\theta(y^j) - b_\theta(y^j))^2.
\]

Here $x^n \in \Omega$, $y^j \in \partial \Omega$ are collocation points. If these points are distributed according to the measures $\pi^d, \pi^b$ then (2.3.3) is a Monte Carlo estimate of (2.3.1), and the network is trained to approximate the solution to (2.2.1) by minimising this estimate with respect to the network parameters. The coefficients $\nu_1, \nu_2$ are interpreted as positive weights which can be adjusted to reflect the relative importance of the terms in (2.3.3). The minimisation algorithm depends on gradients, which are computed via back-propagation using automatic differentiation. Automatic differentiation is applied to compute both the descent direction of the network parameters, and to apply the differential operator $N$ to the neural network at the collocation points. These computations can be routinely implemented using automatic differentiation software such as TensorFlow [1].

If the collocation points are constant throughout training, we arrive at the physics-informed neural network method in [127]. This method is based on a fixed mesh, which requires a sufficiently fine resolution in order to yield accurate results, since the neural network must interpolate the solution between training points. In higher-dimensional spaces, this becomes prohibitively expensive since the number of collocation points required to saturate $\Omega$ grows exponentially with the PDE dimension $d$. 
In contrast, if the collocation points are randomly sampled from $\pi^d, \pi^b$ before each gradient descent iteration, the minimisation of (2.3.1) results in the deep Galerkin method (DGM) described in [144]. This mesh-free approach has proven efficient at accurately approximating high-dimensional solutions, since it allows for substantially fewer collocation points to be used at each iteration. By re-sampling before each iteration, it relies on a mini-batch stochastic gradient descent approach in order to adequately cover the domain. In [144] this approach was used to solve a PDE with dimension $d = 200$. An outline of this algorithm is:

**Algorithm 2**: DGM solver for the deterministic forward PDE problem

1. Randomly initialise a neural network approximation of the solution $\hat{u}_\theta(x) : \mathbb{R}^d \to \mathbb{R}$
2. Draw i.i.d. random samples $(x^n)_{n=1:N} \sim \pi^d, (y^j)_{j=1:J} \sim \pi^b$
3. Update the neural network parameters by taking one gradient descent step to reduce the loss function (2.3.3)
4. Repeat 2-3 until convergence

Upon convergence, this approach will approximate the solution to the PDE for a fixed choice of parameters $\theta$. In our implementations we use the relative change in the moving average of the estimated loss function (2.3.3) as convergence criteria.

### 2.3.2 Parametric forward problem

In order to adapt the deep Galerkin approach to the parametric forward problem we must introduce an additional measure $\pi^\theta$ with support $\text{supp}(\pi^\theta) = \Theta$, and use this to extend the loss function (2.3.1) so that the minimisation also spans the parameter space. The extended loss function is

$$
\nu_1 \left\| N(u(x; \theta), x; \theta_N) - h(x; \theta_h) \right\|_{L_2(\Omega \times \Theta, \pi^d \otimes \pi^\theta)}^2 + \nu_2 \left\| u(x; \theta) - b(x; \theta_b) \right\|_{L_2(\partial \Omega \times \Theta, \pi^b \otimes \pi^\theta)}^2.
$$

(2.3.4)

In this extended loss function the $L_2$-norm is taken over the joint domain $\Omega \times \Theta$ with respect to the product measure $(\pi^d \otimes \pi^\theta)(x, \theta) = \pi^d(x)\pi^\theta(\theta)$. Denoting the neural network approximation of the parametric solution by $\hat{u}(x; \theta)$, we now apply stochastic gradient descent to minimise (2.3.4), by sampling $x^n$ from $\pi^d$, $y^j$ from $\pi^b$, and $\theta^n, \phi^j$. 

65
from $\pi^\theta$ at each iteration and minimising the Monte Carlo estimate given by

$$\frac{L_1}{N} \sum_{n=1}^{N} (N(\hat{u}(x^n; \theta^n), x^n; \theta^n_h) - h(x^n; \theta^n_h))^2 + \frac{L_2}{J} \sum_{j=1}^{J} (\hat{u}(y^j; \phi^j) - b(y^j; \phi^j))^2. \quad (2.3.5)$$

The algorithm for the parametric forward problem is therefore:

**Algorithm 3:** DL solver for the parametric forward PDE problem

1. Randomly initialise a neural network approximation of the parametric solution, $\hat{u}(x; \theta) : \mathbb{R}^{d+p} \to \mathbb{R}$
2. Draw i.i.d. samples $(x^n)_{n=1:N} \sim \pi^d, (y^j)_{j=1:J} \sim \pi^b$,
   $(\theta^n_N, \theta^n_h, \theta^n_b)_{n=1:N} \sim \pi^\theta, (\phi^j_N, \phi^j_h, \phi^j_b)_{j=1:J} \sim \pi^\theta$
3. Update the neural network parameters by taking one gradient descent step to reduce loss function $(2.3.5)$
4. Repeat 2-3 until convergence

We call the resulting neural network $\hat{u}(x, \theta) \approx u(x, \theta)$ a *deep surrogate model* for the parametric PDE. This method of approximating parametric solutions is efficient even over high dimensional parameter spaces where standard discretisations would exhibit exponential growth in computational requirement. In Section 2.6 we implement this approach to a heat transfer PDE applied in aviation, and an electrochemistry PDE problem. The results demonstrate the accuracy and efficiency of the deep surrogate model for rapid solution evaluations over a range of parameter values.

### 2.4 A deep learning approach for integral equations

We now extend the deep learning approaches described in the previous section to the solution of integral equations, which so far have not been discussed in the literature. Our novel methodology allows the computation of approximate solutions to the parametric forward problems for these models.
2.4.1 Deterministic forward problem

Motivated by (2.2.3) and (2.2.4), we begin by defining the function
\[
    w_\theta(x, y) := \int_a^y k_\theta(x, \gamma) u_\theta(\gamma) d\gamma.
\]  
(2.4.1)

Using this, for fixed parameters \(\theta\), the non-homogeneous integral equation of the second kind (2.2.4) can be written as
\[
    u_\theta(x) = v_\theta(x) + w_\theta(x, b(x)), \quad x \in [a, b^*].
\]  
(2.4.2)

We seek to solve (2.4.2) by approximating the solution \(u_\theta(x)\) by a neural network and minimising the residual norm of (2.4.2) in a similar fashion to the PDE based problem discussed in Section 2.3. For integral equations, an added complication arises since \(w_\theta(x, b(x))\) is an integral involving the unknown solution \(u_\theta(x)\). Differentiating (2.4.1) with respect to \(y\) shows that \(w_\theta(x, y)\) solves the initial value problem
\[
    \frac{\partial w_\theta}{\partial y}(x, y) = k_\theta(x, y) u_\theta(y), \quad x \in [a, b^*], \ y \in [a, b(x)],
\]  
\[
    w_\theta(x, a) = 0.
\]  
(2.4.3)

Hence if \(u_\theta(x), w_\theta(x, y)\) satisfy both (2.4.2) and (2.4.3), then \(u_\theta(x)\) is a solution to the integral equation. Proceeding similarly to the PDE-based problem, we define the domain \(\Omega = \{ x, y : x \in [a, b^*], y \in [a, b(x)] \}\) and let \(\pi^d\) be a measure such that supp(\(\pi^d\)) = \(\Omega\). Assuming the integral equation (2.2.4) is well-posed, then for \(\nu_1, \nu_2, \nu_3 > 0\), \(u_\theta(x)\) solves (2.2.4) if \(u_\theta(x)\) and \(w_\theta(x, y)\) are functions such that the following quantity is zero
\[
    \nu_1 \left\| \frac{\partial w_\theta}{\partial y}(x, y) - k_\theta(x, y) u_\theta(y) \right\|^2_{L^2(\Omega, \pi^d)} + \nu_2 \| w_\theta(x, a) \|^2_{L^2(\Omega, \pi^d)}
\]  
\[
    + \nu_3 \| u_\theta(x) - v_\theta(x) - w_\theta(x, b(x)) \|^2_{L^2(\Omega, \pi^d)}.
\]  
(2.4.4)

Here the first two terms being equal to zero ensures that (2.4.3) is satisfied, while the third term accounts for (2.4.2). This motivates the use of two neural networks to approximate the functions \(u_\theta(x), w_\theta(x, y)\). We use \(\hat{u}_\theta(x) \approx u_\theta(x)\) to approximate the solution, and an integrator network \(\hat{w}_\theta(x, y) \approx w_\theta(x, y)\) to approximate the integral term. These networks are trained to approximately satisfy (2.4.2) and (2.4.3) by
minimising a loss function of the form
\[
\frac{1}{N} \sum_{n=1}^{N} \left[ \nu_1 \left( \frac{\partial \hat{u}_\theta(x^n, y^n)}{\partial y} - k_\theta(x^n, y^n) \hat{u}_\theta(y^n) \right)^2 + \nu_2 \hat{w}_\theta(x^n, a)^2 \right. \\
\left. + \nu_3 (\hat{u}_\theta(x^n) - v_\theta(x^n) - \hat{w}_\theta(x^n, b(x^n)))^2 \right],
\]
(2.4.5)

with respect to the neural network parameters. Similarly to the PDE-based problem, (2.4.5) is an unbiased Monte Carlo estimate of (2.4.4) if the collocation points \((x^n, y^n) \in \Omega\) are distributed according to \(\pi_d\). A mini-batch stochastic gradient descent algorithm to train a neural network approximation of the solution to integral equations of the second kind is then:

**Algorithm 4: DL solver for the deterministic integral equation forward problem**

1. Initialise a neural network approximation of the solution, \(\hat{u}_\theta(x) : \mathbb{R} \rightarrow \mathbb{R}\)
2. Initialise an integrator network, \(\hat{w}_\theta(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}\)
3. Draw i.i.d. samples \((x^n, y^n)_{n=1}^{N} \sim \pi_d\)
4. Jointly update both neural networks by taking one gradient descent step to reduce the loss function (2.4.5)
5. Repeat 3-4 until convergence

The algorithm for solving integral equations of the first kind in (2.2.3) is similar, though in this case (2.4.2) is replaced by

\[
0 = v_\theta(x) + w_\theta(x, b(x)),
\]
(2.4.6)

and the corresponding loss function becomes
\[
\nu_1 \left\| \frac{\partial \hat{u}_\theta(x, y)}{\partial y} - k_\theta(x, y) \hat{u}_\theta(y) \right\|_{L_2(\Omega, \pi^d)}^2 + \nu_2 \left\| \hat{w}_\theta(x, a) \right\|_{L_2(\Omega, \pi^d)}^2 \\
+ \nu_3 \left\| \hat{v}(x) + \hat{w}_\theta(x, b(x)) \right\|_{L_2(\Omega, \pi^d)}^2.
\]
(2.4.7)

Again, a Monte Carlo estimate can be constructed by using suitable random collocation points, and this can be reduced using stochastic gradient descent with respect to the parameters of a neural network to approximate the solution to the forward problem.
2.4.2 Parametric forward problem

Following the same approach as in Section 2.3.2, we can also extend the integral equation solver to the solution of the parametric integral equation forward problem. For this, we introduce a measure $\pi^\theta$ with support $\text{supp}(\pi^\theta) = \Theta$, and extend the loss function (2.4.4) over this measure. For integral equations of the second kind the resulting loss function is

$$
\nu_1 \left\| \frac{\partial w}{\partial y}(x, y; \theta) - k(x, y; \theta_k) u(y; \theta) \right\|^2_{L^2(\Omega \times \Theta, \pi^d \otimes \pi^\theta)} + \nu_2 \| w(x, a; \theta) \|^2_{L^2(\Omega \times \Theta, \pi^d \otimes \pi^\theta)}
$$

$$
+ \nu_3 \| u(x; \theta) - v(x; \theta_v) - w(x, b(x); \theta) \|^2_{L^2(\Omega \times \Theta, \pi^d \otimes \pi^\theta)}. \tag{2.4.8}
$$

A neural network approximation to the solution $\hat{u}(x; \theta) : \mathbb{R}^{p+1} \to \mathbb{R}$ can then be substituted into this expression and the corresponding Monte Carlo estimate minimised with respect to the neural network parameters using stochastic gradient descent. Like in the deterministic case, parametric integral equations of the first kind can also be solved by extending (2.4.7) analogously. A demonstration of the algorithm applied in this case is given in Section 2.6, where it is compared to an accurate midpoint quadrature solution and a deep neural network approximation to an equivalent PDE formulation of the problem.

2.5 Deep surrogate approach for Bayesian inference

In the previous sections, we argued that neural networks allow the efficient approximate solution of high-dimensional parameterised problems that can be formulated as PDEs or integral equations. We now show how this can be applied to Bayesian inference. Our approach uses a deep surrogate model to approximate the solution to the parametric forward problem $u(x; \theta)$ over the joint domain $\Omega \times \Theta$. Since the deep surrogate model is efficiently evaluated for any input values $x \in \Omega$ and model parameters $\theta \in \Theta$, it is inserted into a Markov chain Monte Carlo (MCMC) algorithm to quickly draw samples from the posterior distribution. We focus on the Metropolis–Hastings sampler for brevity. However a significant advantage is that our parametric approximation is easily differentiable with respect to the model parameters $\theta$ using automatic differentiation. This allows the possibility to apply more efficient gradient based sampling methods such as Hamiltonian Monte Carlo [13].
2.5.1 Bayesian inference

Bayesian inference is a means of inferring a distribution over a set of parameters $\tilde{\theta}$ by conditioning on an observed dataset $\tilde{z}$. In this approach the posterior probability of the parameters is calculated based on a prior distribution, a likelihood function, and some observed data. The Bayesian approach allows us to make inferences about model parameters through analysis of the posterior distribution

$$p(\tilde{\theta}|\tilde{z}) = \frac{p(\tilde{z}|\tilde{\theta})p(\tilde{\theta})}{\int p(\tilde{z}|\tilde{\theta})p(\tilde{\theta})d\tilde{\theta}}.$$  \hspace{1cm} (2.5.1)

Here the prior distribution, with density $p(\tilde{\theta})$, is chosen to describe one’s existing knowledge about the parameters prior to data being observed. The likelihood, $p(\tilde{z}|\tilde{\theta})$ is the conditional density of the data $\tilde{z}$ given parameters $\tilde{\theta}$ according to a statistical model. If we assume that measurement errors on the data are normally distributed with variance $\sigma^2$ as in (2.2.5), then the unknown parameters are $\tilde{\theta} = \{\theta, \sigma^2\} \in \Theta \times \mathbb{R}^+$, and the likelihood function is

$$p(\tilde{z}|\tilde{\theta}) = \frac{1}{(2\pi\sigma^2)^M/2}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{M}(\hat{z}_i - u(\hat{x}_i; \theta))^2\right).$$ \hspace{1cm} (2.5.2)

The product of the prior and likelihood is proportional to the posterior density, $p(\tilde{\theta}|\tilde{z})$. This is the joint distribution of the parameters $\tilde{\theta}$ conditional on the data $\tilde{z}$, and represents an updated belief about the about the distribution of the unknown parameters $\tilde{\theta}$ based on the data. From this distribution estimates of the parameter values, as well as any uncertainties and dependencies can be derived.

In general the exact functional form of the posterior cannot be directly computed and thus must be approximated. A popular method in statistics and data science is to apply a MCMC scheme to sample from the posterior distribution over the parameters $\tilde{\theta}$ [145]. One of the simplest and most widely used MCMC schemes is the Metropolis–Hastings algorithm [104]. In this scheme a Markov chain is constructed by proposing successive states from a proposal distribution $q(\tilde{\theta}_{\text{prop}}|\tilde{\theta})$ which can depend on the current state $\tilde{\theta}$. In this scheme it can be shown that if the proposals are accepted with probability

$$A(\tilde{\theta}_{\text{prop}}, \tilde{\theta}) = \min\left(1, \frac{q(\tilde{\theta}_{\text{prop}}|\tilde{\theta})p(\tilde{z}|\tilde{\theta}_{\text{prop}})p(\tilde{\theta}_{\text{prop}})}{q(\tilde{\theta}|\tilde{\theta}_{\text{prop}})p(\tilde{z}|\tilde{\theta})p(\tilde{\theta})}\right),$$ \hspace{1cm} (2.5.3)
then \( p(\tilde{\theta}, \tilde{z}) \) is the stationary distribution of the Markov chain. Assuming mild conditions on \( q(\cdot|\cdot) \) to ensure ergodicity (a sufficient condition is \( q(\tilde{\theta}|\tilde{\theta}_{\text{prop}}) > 0 \ \forall \ \tilde{\theta}, \tilde{\theta}_{\text{prop}} \in \Theta \times \mathbb{R}^+ \)), the states visited by the Markov chain form a sample from the posterior distribution. The Metropolis–Hastings algorithm can be written:

**Algorithm 5: Metropolis–Hastings**

Choose initial \( \tilde{\theta}_0 \in \Theta \times \mathbb{R}^+ \);

for \( i = 0, 1, \ldots, N_{\text{samples}} \) do

    Propose \( \tilde{\theta}_{\text{prop}} \sim q(\tilde{\theta}_{\text{prop}}|\tilde{\theta}_i) \);
    Sample \( u \sim U(0, 1) \);
    if \( u \leq A(\tilde{\theta}_{\text{prop}}, \tilde{\theta}_i) \) then
        set \( \tilde{\theta}_{i+1} := \tilde{\theta}_{\text{prop}} \) (accept);
    else
        set \( \tilde{\theta}_{i+1} := \tilde{\theta}_i \) (reject);
    end

end

Return \( \tilde{\theta} = (\tilde{\theta}_0, \tilde{\theta}_1, \ldots, \tilde{\theta}_{N_{\text{samples}}}) \)

The empirical distribution of samples generated in this way converges in distribution to the true posterior distribution \([2.5.1]\) as the number of samples increases. To ensure a representative sample in practice, some proportion of the samples are removed from the start of the sample due to the 'burn-in' period. These represent the time taken for the Markov chain to transition towards the typical set of the posterior distribution and are typically a poor representation of the true posterior. Since subsequent samples in the Markov chain are correlated, it is typically subsampled to obtain approximately uncorrelated samples.

Although MCMC methods are widely used in applied statistics, their application to large-scale, physics-based Bayesian inverse problems remains very challenging. This is because in each iteration of Algorithm 5 the prior density and likelihood function must be evaluated for the proposed parameters to calculate the acceptance probability \([2.5.3]\). The prior density is chosen in advance and is easily evaluated for any parameter values. However evaluating the likelihood function \([2.5.2]\) requires the solution \( u(\hat{x}_i; \theta) \) to the PDE or integral equation. This is a key computational bottleneck since a numerical forward solve must be performed at each MCMC iteration. Furthermore a large number of MCMC iterations are required to produce a sufficient an accurate approximation to the posterior. This repeated execution of expensive numerical schemes
has so far restricted the MCMC approach to simple models where numerical solvers are not too expensive.

To overcome these limitations, we apply the deep surrogate model to Bayesian inference by introducing the surrogate into the Metropolis–Hastings algorithm. Specifically, we approximate \( u(\hat{x}_i, \theta) \) by the surrogate \( \hat{u}(\hat{x}_i, \theta) \) during the likelihood calculations in the MCMC scheme, thus replacing a numerical solve at each iteration with a cheap evaluation of a known analytic function. Our proposed surrogate model provides a computationally tractable approach to accurately approximate the parametric solutions to PDE and integral equations over the high-dimensional space \( \Omega \times \Theta \). Since the surrogates can be evaluated rapidly for any parameter values, they are a powerful tool for Bayesian sampling problems. Our numerical results in Section 2.6 demonstrate that this allows for fast posterior sampling when compared to solvers based on grid-based discretisations. Additionally, the separation of the approximation of the parametric solution from the MCMC sampling means that once the surrogate is constructed, it can be stored and used to do inference with different data sets without the need to re-solve the PDE or integral equation.

The next section explores two examples which demonstrate that neural networks allow the accurate and computationally efficient approximation of parametric solutions. The numerical results confirm that this approach significantly accelerates Bayesian inference with an MCMC sampler.

### 2.6 Examples

Here we apply our method to an electrochemistry Bayesian inverse problem that can be represented both as an integral equation and a PDE. We compare our parametric integral equation solver to the deep learning solver for parametric PDEs and an accurate quadrature scheme. Additionally, we apply our methodology to the inference of a function valued parameter used when modelling heat transfer in aircraft compressor turbines. This example demonstrates our approach is capable for high dimensional problems by sampling from the posterior distribution of the parameters of a reduced basis representation of the function of interest. All examples were implemented in TensorFlow using a laptop with a 6 core 3.9GHz CPU and a mobile RTX 2080 GPU.
2.6.1 Voltammetry

Voltammetry is an experimental technique used in electrochemistry to infer the oxidation potential of a chemical. Our description of this problem is based on the construction given in [9]. A standard experiment is to apply a potential to an electrode in an electrochemical cell. This causes electron transfer which is measured as an electrical current. This current manifests as the rate of reaction taking place at the electrode surface, which is related to the rate of change in concentration of the chemical species. Using measurements of the current we can deduce information about the chemical system. Specifically, the inverse problem we will consider is to infer the formal potential $E^0$ of the chemical, which is related to the amount of energy that is required to stimulate a reaction. This experiment can be modelled by a PDE or an integral equation, and therefore it allows us to compare the new deep surrogate approach for integral equations to existing PDE-based techniques.

2.6.1.1 PDE approach

We first model the system by a 1-dimensional PDE which describes the transport of chemicals through diffusion

$$\frac{\partial C_s}{\partial t} = D_s \frac{\partial^2 C_s}{\partial x^2}. \quad (2.6.1)$$

Here $x \in [0, \infty)$ is the distance from the electrode surface, $C_s(x,t)$ represents the concentration of the chemical species $s$, and $D_s$ is the known scalar diffusion coefficient of species $s$. We seek the solution for all locations $x$ and times $t$ such that $x = (x,t) \in \Omega = [0, \infty) \times [0, \infty)$.

The experiment we will consider involves 2 chemical species, $A$ and $B$. It begins with only chemical $A$ present over the spatial domain at time $t = 0$. We assume that the oxidation reaction $A \rightarrow B + e^-$ (the conversion of $A$ to $B$ through the loss of an electron) takes place at the electrode surface at a rate dependent on the intensity of the applied potential. For simplicity, we assume that $A$ and $B$ share the same constant diffusion coefficient, then after non-dimensionalising appropriately we have a system of
equations

\[ \frac{\partial C_a}{\partial t} = \frac{\partial^2 C_a}{\partial x^2}, \quad (x,t) \in \Omega, \quad (2.6.2) \]

\[ \frac{\partial C_b}{\partial t} = \frac{\partial^2 C_b}{\partial x^2}, \quad (x,t) \in \Omega. \quad (2.6.3) \]

Here \( C_a \) and \( C_b \) represent the concentrations of chemicals \( A \) and \( B \). We assume local conservation of matter, so that

\[ C_a(x,t) + C_b(x,t) = 1, \quad (x,t) \in \Omega, \quad (2.6.4) \]

therefore it is sufficient to solve for just \( C_a(x,t) \). Initially, we assume that only chemical \( A \) is present, so the initial condition is

\[ C_a(x,0) = 1, \quad x \in [0, \infty). \quad (2.6.5) \]

We impose far field boundary conditions for all \( t \in [0, \infty) \)

\[ C_a(x,t) \to 1, \quad x \to \infty. \quad (2.6.6) \]

The final boundary condition at \( x = 0 \) depends on the potential at the electrode. In linear sweep voltammetry a linearly increasing current of the form

\[ E(t) = E_{start} + t, \quad (2.6.7) \]

is applied. The boundary condition is then

\[ C_a(0,t) = \frac{1}{1 + e^{E(t) - E_0}}, \quad t > 0. \quad (2.6.8) \]

The current is the rate of reaction given by the gradient at the electrode surface

\[ I(t) = \frac{\partial C_a}{\partial x} \bigg|_{x=0}. \quad (2.6.9) \]

The forward problem is to produce this current as a function of the applied potential \( E(t) \). The inverse problem is to infer the formal potential \( E^0 \) of the chemical, given noisy current measurements from experiments.
2.6.1.2 Integral equation approach

Linear sweep voltammetry can also be described by an integral equation by taking the Laplace transform of the PDE problem. Letting \( \tilde{C}_a(x,s) = \mathcal{L}(C_a(x,t)) \) denote the Laplace transform of \( C_a(x,t) \) with respect to \( t \), it can be shown that

\[
\tilde{C}_a(x,s) = c_1(s)e^{-\sqrt{s}x} + \frac{1}{s},
\]

(2.6.10)
for some function \( c_1(s) \). To calculate the current we take the Laplace transform of (2.6.9), giving

\[
\tilde{I}(s) = \frac{\partial \tilde{C}_a}{\partial x} \bigg|_{x=0},
\]

(2.6.11)
where \( \tilde{I}(s) = \mathcal{L}(I(t)) \) denotes the Laplace transform of \( I(t) \). Differentiating (2.6.10) with respect to \( x \), and inserting into (2.6.11) with \( x = 0 \) we deduce

\[
c_1(s) = -\frac{\tilde{I}(s)}{\sqrt{s}},
\]

(2.6.12)
and therefore

\[
\tilde{C}_a(0,s) = -\frac{\tilde{I}(s)}{\sqrt{s}} + \frac{1}{s}.
\]

(2.6.13)

To recover \( C_a(0,t) \) from (2.6.13) we apply the inverse Laplace transform to \( \tilde{C}_a(0,s) \) using the convolution theorem and identities: \( \mathcal{L}^{-1}\left(\frac{1}{\sqrt{s}}\right) = \frac{1}{\sqrt{\pi t}}, \mathcal{L}^{-1}\left(\frac{1}{s}\right) = 1 \). This allows us to express the concentration at the electrode as an integral over the current,

\[
C_a(0,t) = 1 - \frac{1}{\sqrt{\pi}} \int_0^t I(\tau) \cdot \frac{1}{\sqrt{t-\tau}} d\tau.
\]

(2.6.14)
Together with the boundary condition (2.6.8) this gives rise to the following Volterra integral equation of the first kind

\[
\frac{\sqrt{\pi}}{1 + e^{-(E(t)-E^0)}} = \int_0^t \frac{I(\tau)}{\sqrt{t-\tau}} d\tau.
\]

(2.6.15)
The solution \( I(t) \) of this integral equation is the same as the gradient defined in (2.6.9) of the solution of the PDE problem (2.6.2, 2.6.5, 2.6.6, 2.6.8). The objective here is the same as before: we wish to infer the formal potential \( E^0 \) given measurements of the time-dependent current \( I(t) \) from experimentation.
The deep surrogate models described in Section 2.3 and 2.4 are used to approximate parametric solutions to both the PDE and integral equation formulation of the problems described in Sections 2.6.1.1 and 2.6.1.2 respectively. We use $E_{\text{start}} = -15$ in (2.6.7), and work with the truncated domain $\bar{\Omega} = [0,200] \times [0,25]$. The neural networks were constructed and trained using TensorFlow [1].

For the PDE problem, we applied the methods described in Section 2.3 to solve the parametric PDE problem using a fully connected neural network with 4 hidden layers of 45 neurons and tanh activation functions. The approximation $\hat{I}(t; E_0)$ to the current $I(a; E_0)$ defined in (2.6.9) is recovered by computing the gradient $\partial_x \hat{C}_a(0,t; E_0)$ at the spatial boundary of the surrogate approximation using automatic differentiation. For the integral equation approach described in Section 2.6.1.2 we approximate the parametric solution $I(t; E_0)$ of (2.6.15) for $t \in [0,25]$ using the methods described in Section 2.6.4. The approximation to the solution and the integrator network were represented by separate fully connected neural networks, each with tanh activation functions and 4 hidden layers containing 45 neurons. The training time for the networks in both cases was under 5 minutes.

As reference solutions $I(t; E_0)$ for each potential $E_0$ we numerically solve the integral equation by approximating the integral in (2.6.15) by a midpoint quadrature rule and solving the resulting system of linear equations. To achieve a high accuracy we use a very small grid spacing in the temporal direction of $\Delta t \approx 5 \times 10^{-4}$. These reference solutions are used as reliable approximations of the true solutions and compared to the deep surrogate approach. For $E_0 = \{-4,0,4\}$, Figure 2-1 compares these reference solutions to the parametric solutions achieved by the surrogates. In each case the neural network solutions are close to the references. In particular, we see that the deep learning integral equation solver has comparable accuracy to the established deep learning PDE solver.
Figure 2-1: Comparison of neural network approximations and midpoint quadrature for the voltammetry example in Section 2.6.1. The top row shows the currents $I(t; E^0)$ approximated by each method for different values of $E^0$. The error in the bottom row is the difference between the neural networks and the reference solution obtained by solving the integral equation with midpoint quadrature.

Evaluating both the integral equation and PDE based surrogates is significantly faster than performing quadrature or finite difference at a resolution that achieves the same accuracy. For the integral equation surrogate we are able to perform 178,194 evaluations every second at 100 data locations. In comparison, applying midpoint quadrature with a resolution that achieves the same accuracy as the surrogate we can solve the integral equation problem just 229 times a second using optimised forward substitution methods to solve the lower triangular system. Our PDE surrogate model is slightly slower, achieving 171,943 evaluations at the data points each second, due to requirement to automatically differentiate the approximation to the PDE at the spatial boundary. Solving the 2-dimensional PDE by applying Crank–Nicolson finite differences with sparse linear algebra methods takes 1.51 seconds for a single solve at the same accuracy as the surrogate. Hence in both cases we see a massive a speed-up of 3-6 orders of magnitude using the surrogate models over traditional discretisation schemes.
Having solved the forward problem, we now consider the Bayesian inverse problem. We generate synthetic data for the Bayesian inference task by randomly adding noise to the reference solutions at 100 random time points before sampling. To study the effect of the noise-level of the data on the posterior distributions for $E^0$, Gaussian noise with different variances was used for $E^0 = \{-4, 0, 4\}$. A uniform $U(-6, 6)$ prior distribution was placed on $E^0$, and a $U(0, 3)$ prior distribution was placed on $\sigma$. The neural network surrogate solutions were then used in a Metropolis–Hastings sampler \cite{104} to approximate the joint posterior distribution. Figure 2-2 shows the simulated data (first row) together with the approximated marginal posterior distributions for $E^0$ obtained using the PDE-based surrogate model (second row). For the integral equation based surrogate the results are presented in the same format in Figure 2-3. The approximated posterior distributions of $\sigma$ also gave estimates consistent with the true noise levels, however these results are omitted.

Figure 2-2: Top: Simulated data and fitted solutions using the mean of the posterior. The true standard deviations $\sigma$ and oxidation potential $E^0$ used to generate the Gaussian noise are given at the top of each column. Bottom: Approximate posterior distributions and 95% credible intervals computed using the PDE deep surrogate model. The true values of $E^0$ used to generate the data are marked by vertical lines.
For both the PDE and integral equation based surrogates, 500,000 Metropolis–Hastings iterations were used to sample from the posterior, and kernel density estimation was used to visualise the distributions. In each case, the means of the MCMC samples are close to the true parameter used to generate the data, and the width of the credible intervals increases for noisier data as we would expect. The shape and width of the posterior distributions for the integral equation surrogate are consistent with the PDE based surrogate, though we note some slight random variation in their estimates due to the use of different randomly generated noise.

Figure 2-3: Top: Simulated data and fitted solutions using the posterior mean. The true oxidation potential $E^0$ and standard deviations $\sigma$ used to generate the Gaussian noise are given at the top of each column. Bottom: Posterior distributions and 95% credible intervals computed using the integral equation deep surrogate. The true value of $E^0$ used to generate the data are marked by vertical lines.
2.6.2 Inferring the non-constant Biot number in rotating discs

Heat transfer in rotating-disc systems encountered in aircraft turbines comprises two mechanisms; heat diffusion in the solid disc and convection of fluid particles. The steady state behaviour of this process can be modelled using the one-dimensional Fin equation [73], which only considers variations in the radial direction. The relative effects of diffusion and convection are quantified by a spatially varying parameter known as the Biot number. Estimating the Biot number is of interest in aviation applications in order to predict material expansion in aircraft turbines.

Using suitable units, we consider a rotating disc with inner and outer radii $a$ and $1$ respectively, leading to the domain $\Omega = [a, 1]$. In this case, the Fin equation with constant Dirichlet boundary conditions is

$$\frac{d^2 u}{dx^2} + \frac{1}{x} \frac{du}{dx} - Bi \ u = 0, \quad x \in \Omega,$$

$$u(a) = u_a,$$

$$u(1) = u_1.$$  \hspace{1cm} (2.6.16)

Here $u = u(x)$ is a non-dimensional temperature, $Bi = Bi(x)$ is the Biot number, $x$ represents radial distance from the centre of rotation, and $u_a, u_1 \in \mathbb{R}$ are fixed Dirichlet boundary conditions. Analytical solutions to (2.6.16) are intractable when the Biot number varies over the domain, however it can still be solved numerically in this case. The inverse problem we consider here is to infer the functional parameter $Bi(x)$, given noisy measurements of the temperature profile $u(x)$ at a set of points $x \in \Omega$.

In principle $Bi(x)$ can be determined numerically if the radial distribution of the temperature $u(x)$ is known, however this is an ill-posed inverse problem where very small uncertainties in the temperature measurements can create large uncertainties in the computed Biot number. This feature of the inverse problem has caused early curve fitting attempts based mean squared error minimisation to infer curves with unphysical behaviour [3, 151]. Similar issues arise if the physics-informed deep learning approach for inverse problems [127] is applied directly as Figure 2-5 below shows. A maximum a posteriori approach (MAP) to the inverse problem of inferring $Bi(x)$ from $u(x)$ indicated that a Bayesian approach can be robust to this sensitivity and allow for reliable inferences [151]. Although the MAP approach gives us an estimate of $Bi(x)$, it does not reliably quantify uncertainties in this Biot number. To overcome this limitation we apply the deep surrogate method to approximate the posterior distribution of $Bi(x)$.  

80
2.6.2.1 Results

The parameter $Bi(x)$ which is to be inferred lies in the infinite-dimensional space of all continuous function. To make the problem computationally tractable, we consider Biot numbers in the finite dimensional subspace of polynomials of degree up to 15. More specifically, we write the Biot number as a finite sum of monomials

$$\widetilde{Bi}(x; \theta) = \sum_{n=0}^{15} \theta_n x^n, \quad x \in \Omega, \ \theta \in \Theta. \quad (2.6.17)$$

We then use a neural network to generate an approximate parametric solution $\hat{u}(x; \theta)$ of the PDE, which depends on the position $x \in \Omega = [a, 1]$, and the polynomial coefficients $\theta \in \Theta \subset \mathbb{R}^{16}$ that define $\widetilde{Bi}(x; \theta)$.

Using this representation we approximate the solution $u(x; \theta)$ of the parametric forward problem by minimising (2.3.5). To represent $\hat{u}(x; \theta)$ we used a fully connected neural network with a 17-dimensional input layer with arguments $x \in \mathbb{R}$ and $\theta \in \mathbb{R}^{16}$, 4 hidden layers with 45 neurons per layer using tanh activation functions. The output layer of the network $\hat{u}(x; \theta)$ approximates the solution $u(x; \theta)$. A $U(0, 1)$ prior was placed on $\sigma$, and normal priors were placed on the polynomial coefficients of the form $\theta_n \sim N(0, V_n)$, where $V_0 = 20$ and $V_n = 20/2^{n-1}$ $n = 1, 2, \ldots, 15$. The decay in the width of the Gaussian priors enforces the smoothness of the functions in the posterior, reducing the ill-posedness of the problem and preventing the non-physical estimates. The training time for the parametric forward problem over this domain was 14 minutes. Once trained, the surrogate can be evaluated 587304 times each second at 30 data locations. This performance is expected since this network has the same architecture as the surrogate used in the voltammetry integral equation example, however in this example there are 70% fewer data points leading to an increase in speed of over 3 times.

Using the trained neural network approximation $\hat{u}(x; \theta)$ in a Metropolis–Hastings sampler we then sample from the posterior obtained with synthetic data. This data is generated by using finite differences to solve (2.6.16) with $Bi(x) = 18e^{x-0.3}$, this non-polynomial is typical of real world data for this scenario (see e.g. [151]). We add Gaussian noise with standard deviation $\sigma = 0.003$ to the solution at 30 equidistant points as shown on the left of Figure 2-4. Overlaid on this plot is the solution fitted by solving (2.6.16) using the $\widetilde{Bi}(x; \theta)$ estimated using the mean of the posterior sample. The right side of Figure 2-4 shows the estimated $\widetilde{Bi}(x; \theta)$ (blue) and its 95% credible interval compared to the true $Bi(x)$ (red). The plot demonstrates that the true Biot
number lies within the 95% credible region, and that the fitted solution matches the data well.

![Figure 2-4: Left: Fitted heat profile using the inferred Biot number. Right: Inferred \(\tilde{Bi}(x; \theta)\) and 95% credible interval compared to the true Biot number.](image)

As a comparison, we implement the physics-informed deep learning approach described in [127]. For this we again represent the Biot number as the sum of monomials given in (2.6.17) and construct the augmented loss function

$$
\nu_1 \frac{\delta}{N} \sum_{n=1}^{N} (\mathcal{N}_\theta(\hat{u}_\theta(x^n), x^n) - h_\theta(x^n))^2
$$

$$
+ \nu_2 \frac{\delta}{J} \sum_{j=1}^{J} (\hat{u}_\theta(y^j) - b_\theta(y^j))^2 + \nu_3 \frac{\delta}{M} \sum_{i=1}^{M} (\hat{u}_\theta(\hat{x}_i) - \hat{z}_i)^2.
$$

The addition of the final term in (2.6.18) encourages the optimiser to minimise the squared difference of the learned solution from the data, this approach has been shown to work well with very large amounts of simulated data. This loss function is minimised with respect to both the parameters of the neural network, and the polynomial coefficients \(\theta\) simultaneously. Figure 2-5 shows the estimate achieved by applying this scheme with the same data used in Figure 2-4. The neural network was trained until convergence, with the weighting coefficients in (2.6.18) set to \(\nu_1 = \nu_2 = \nu_3 = 1\).
Here we see that the fitted solution is close to the data, however the inferred Biot number does not closely match its true value, and would clearly lie outside the 95% credible interval computed by our Bayesian approach (see Figure 2-4). Here we have only 30 noisy data points, and so the minimiser of (2.6.18) must balance the accuracy of the PDE and the interpolation of the data. This risks converging to inaccurate solutions at the expense of better interpolation of the data. In Figure 2-5 the impact of this behaviour on inference is exacerbated due to the ill-posedness of the inverse problem. The method proposed in [127] also produces no uncertainty estimates over this parameter giving us no indication of confidence in the estimates. This demonstrates that for ill-posed inverse problems it is advantageous to compute an accurate surrogate solution first, and that by using Bayesian methodology suitable regularisation and uncertainty quantification can be achieved.
2.7 Conclusion

In this paper we propose a new deep learning approach for the solutions of integral equations. Our method uses two neural networks, one to approximate the solution of the integral equation, and another to approximate the integral term of these equations. Both networks are trained simultaneously by using mini-batch stochastic gradient descent to minimise a loss function designed such that its minimiser solves the equation. Using this algorithm and existing approaches for PDE problems we described how to construct deep surrogate models to approximate the parametric PDE and integral equation forward problems, by extending the sampling domain to include the parameter space of the problem. Numerical evidence illustrates the accuracy of the parametric integral equation solver when compared to a quadrature based solver. We then applied our surrogate models to sample from the posterior distribution of PDE and integral equation based Bayesian inverse problems. We demonstrated that our surrogate models can be evaluated rapidly, achieving a sampling acceleration of several orders of magnitude at the same accuracy when compared to more traditional discretisation schemes. The examples that we considered in this paper show that empirical posterior distributions of the model parameters achieved by applying this scheme are consistent with the true parameters.

In contrast to traditional grid-based methods, for which the computational complexity grows exponentially with the dimension, the deep surrogate model is efficient at approximating solutions to high dimensional parametric forward problems. Additionally, since our method is trained to satisfy the equations directly over the entire parameter space, it approximates the solution more accurately than surrogate models based on interpolation. Our framework is flexible enough to be applied to a wide range of PDEs and integral equations, and extensions to higher dimension integral equations, problems with Neumann boundary conditions, or systems of equations, are possible using the principles described. Once training is complete the deep surrogate model can be evaluated rapidly, and in parallel, making deep surrogates ideal candidates to accelerate MCMC schemes for Bayesian inverse problems.

Upcoming research on this topic will investigate how deep surrogate models can be applied in more complex and varied scenarios. A key benefit of the surrogate over traditional methods is its differentiability with respect to the PDE or integral equation model parameters. Future work will utilise this to develop bespoke samplers using gradient based transition kernels and delayed acceptance criteria that mathematically
guarantees the accuracy of the posterior samples in large-scale problems. Other possible
extension include adaptive surrogate construction and the interaction of physics (rep-
resented by deep surrogate models) with applied statistical problems in environmental
sciences.
Chapter 3

Deep surrogate accelerated delayed-acceptance HMC for Bayesian inference of spatio-temporal heat fluxes in rotating disc systems.

The work presented in this chapter was undertaken as part of a collaboration with the Department of Mechanical Engineering. Here we present two papers. The first paper develops the mathematical and computational methodology for the project, and is reproduced from the authors’ submitted manuscript [39]. The second paper, which focuses on the experimental setting and practical applications of this method, is reproduced from the authors’ published manuscript in collaboration with colleagues from the Department of Mechanical Engineering [148].

In Chapter 2, we demonstrated a deep learning approach for performing Bayesian inference for PDEs. Within this chapter we illustrated this approach with a 1-dimensional boundary value problem for the heat transfer parameter known as the Biot number. In Chapter 3, we significantly expand upon the techniques developed in Chapter 2 in order to solve a spatio-temporal extension of the Biot number problem. The results of this work can be applied in industrial settings to improve aircraft engine design.

There are major contributions of this work relating both to the methodology and the
application of interest. Due to the intricacy of the industrial application, our methodo-
logical improvements focus on developing the reliability, efficiency and interpretability
of the method introduced in Chapter 2. To this end, we describe an approach to spec-
ifying interpretable priors over the Biot number by approximating a Gaussian process,
implement an adaptive surrogate training algorithm that attains more accurate approx-
imations in a reduced time, and then ensure that accurate posterior samples can be
generated efficiently through the use of a delayed-acceptance scheme with a Hamilto-
nian Monte Carlo proposal distribution. The contribution to the application is the full
Bayesian uncertainty quantification of the spatio-temporal heat fluxes across a range of
real world experimental settings using temperature measurements from a compressor
cavity rig.
This declaration concerns the article entitled:

Deep surrogate accelerated delayed-acceptance HMC for Bayesian inference of spatio-temporal heat fluxes in rotating disc systems.

<table>
<thead>
<tr>
<th>Publication status (tick one)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft manuscript</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Publication details (reference)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Copyright status (tick the appropriate statement)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I hold the copyright for this material</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Candidate’s contribution to the paper (provide details, and also indicate as a percentage)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The candidate predominantly executed the…</td>
</tr>
<tr>
<td>Formulation of ideas: 70%. The author of this thesis developed the project concept in conjunction with other authors and colleagues in the Department of Mechanical Engineering.</td>
</tr>
<tr>
<td>Design of methodology: 90%. The author of this thesis designed the full methodological framework. Suggestions from other authors were considered and expanded upon as appropriate as part of this.</td>
</tr>
<tr>
<td>Experimental work: 100%. All code was developed and implemented by the author of this thesis.</td>
</tr>
<tr>
<td>Presentation of data in journal format: 90%. The author of this thesis produced all results and drafts of the manuscript. Feedback was provided by all other authors.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statement from Candidate</th>
</tr>
</thead>
<tbody>
<tr>
<td>This paper reports on original research I conducted during the period of my Higher Degree by Research candidature.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Signed</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teo Deveney</td>
<td>05/06/2022</td>
</tr>
</tbody>
</table>
This declaration concerns the article entitled:

Use of Bayesian statistics to calculate transient heat fluxes on compressor discs.

<table>
<thead>
<tr>
<th>Publication status (tick one)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft manuscript</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Publication details (reference)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Copyright status (tick the appropriate statement)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I hold the copyright for this material ✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Candidate’s contribution to the paper (provide details, and also indicate as a percentage)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The candidate considerably contributed to...</td>
</tr>
<tr>
<td>Formulation of ideas: 30%. The author of this thesis proposed the mathematical method to solve the problem posed by co-authors in the Department of Mechanical Engineering.</td>
</tr>
<tr>
<td>Design of methodology: 50%. The author of this thesis designed the mathematical framework used to solve the problem provided by co-authors. The other authors designed the experimental methodology and data collection, etc.</td>
</tr>
<tr>
<td>Experimental work: 50%. All code was developed and implemented by the author of this thesis. Experimental rig work was carried out by authors in the Department of Mathematical Engineering.</td>
</tr>
<tr>
<td>Presentation of data in journal format: 30%. The author of this thesis produced the numerical results for this manuscript. The author of this thesis also provided contributions to section 3A.4 and wrote section 3A.5 (“Bayesian approach”) of the manuscript.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statement from Candidate</th>
</tr>
</thead>
<tbody>
<tr>
<td>This paper reports on original research I conducted during the period of my Higher Degree by Research candidature.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Signed</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teo Deveney</td>
<td>05/06/2022</td>
</tr>
</tbody>
</table>
Abstract

We study the Bayesian inverse problem of inferring the Biot number, a spatio-temporal heat-flux parameter in a PDE model. This is an ill-posed problem where standard optimisation yields unphysical inferences. We introduce a training scheme that uses temperature data to adaptively train a neural-network surrogate simulating the parametric forward model. By simultaneously identifying an approximate posterior distribution over the Biot number, and weighting a physics-informed training loss according to this, our approach approximates forward and inverse solution together without any need for external solves. Using a random Chebyshev series, we outline how to approximate a Gaussian process prior, and using the surrogate we apply Hamiltonian Monte Carlo (HMC) to sample from the posterior distribution. We derive convergence of the surrogate posterior to the true posterior distribution in the Hellinger metric as our adaptive loss approaches zero. Additionally, we describe how this surrogate-accelerated HMC approach can be combined with traditional PDE solvers in a delayed-acceptance scheme to a-priori control the posterior accuracy. This overcomes a major limitation of deep learning-based surrogate approaches, which do not achieve guaranteed accuracy a-priori due to their non-convex training. Biot number calculations are involved in turbo-machinery design, which is safety critical and highly regulated, therefore it is important that our results have such mathematical guarantees. Our approach achieves fast mixing in high-dimensional parameter spaces, whilst retaining the convergence guarantees of a traditional PDE solver, and without the burden of evaluating this solver for proposals that are likely to be rejected. A range of numerical results are given using real and simulated data that compare adaptive and general training regimes and various gradient-based Markov chain Monte Carlo (MCMC) sampling methods.
3.1 Introduction

The disc temperature distribution in compressor cavities is a fundamental quantity of interest for aerospace engineers due to its effect on material expansion. In order to improve engine design, engineers are interested in simulating the temperature evolution over time, leading to an urgent requirement for accurate physical models of heat transfer inside the engine cavity. Appropriate parameterisations of these models in a transient setting are currently not well understood, and therefore experimental data can be extremely valuable in aiding our knowledge of the parameters. One very important parameter in such models is the Biot number, a function which can vary over space and time, that dictates the relative effects of convection and conduction on heat transfer.

Previous work has been carried out to infer the Biot number from temperature measurements in a stationary setting. In [3], polynomial curves were fit to data using least squares. This approach, while yielding a good fit, lacks appropriate regularisation which often leads to physically implausible inferences with large oscillations unless restricted to very low degree polynomials. This effect is highlighted in [151], where a Bayesian regularisation method is instead proposed based on maximum a posteriori (MAP) estimation over a spatial discretisation of the Biot number. This approach is shown to yield physical results and a local estimate of the uncertainty is achieved using a Laplace approximation based on the Hessian of the log-posterior. Using this Laplace approach, the full posterior distribution is not returned, meaning the uncertainty estimate may be unreliable. Furthermore, the increase in number of degrees of freedom of the Biot number coupled with the higher complexity of the PDE solve in the spatio-temporal setting pursued in this work, ensure that the extension of the approach of [151] to this case is computationally intractable. This is because it is reliant on numerically calculating the gradient of the posterior with respect to each degree of freedom of the discretised Biot number. More efficient approaches to achieve this are possible by using the adjoint PDE to solve the PDE-constrained optimisation problem, though a large number of numerical solves is still required in this case, and the local estimate of the associated uncertainty returned is insufficient as our numerical results reveal.

In this work, we overcome the limitations imposed by traditional numerical schemes in the spatio-temporal setting, by using deep learning to develop a Bayesian methodology capable of approximating the full posterior distribution of the Biot number. Our ap-
proach is to represent the parametric forward map by a neural network, thus greatly accelerating the simulation and differentiation of the PDE model. To attain this map efficiently, we design an adaptive training scheme, based on minimising the squared PDE-residual over a measure that approximates the true posterior over the parameters. The relationship between our adaptive training loss function value and the accuracy of the posterior approximation induced by this surrogate is analysed in the Hellinger metric. We demonstrate that this restriction to the posterior measure vastly improves the approximation speed and accuracy when compared to a general parametric approximation over a wider parameter space. Given our approximate forward map, we use the Hamiltonian Monte Carlo (HMC) sampling scheme \[14\] to generate proposal samples from the posterior distribution. Using this method, we are able to perform a full Bayesian analysis of the posterior distribution in minutes. Our results on simulated data show that a fully Bayesian approach is justified, as it provides a more accurate quantification of uncertainty than the Laplace approximation, which gives overly confident results for this problem.

PDE surrogates based on the approximation of parametric solutions by neural networks have been applied with impressive results previously \[38, 162, 165\]. However, due to the non-convex nature of the training procedure, these approaches suffer from an inability to guarantee the accuracy of the approximations that are attained. To overcome this, we additionally propose delayed-acceptance as part of our MCMC scheme \[29\]. In this delayed-acceptance HMC scheme, proposals which pass the initial surrogate-based Metropolis acceptance criteria are passed to a secondary acceptance criteria dependent on a Crank–Nicolson (CN) solver. The delayed-acceptance criteria is chosen such that the stationary distribution of the Markov chain satisfies detailed balance according to the likelihood induced by the CN solver. This approach, while slower than relying solely on the deep learning surrogate, ensures that the computation time dedicated to CN is being used optimally, since CN is only executed for proposals that have passed initial acceptance criteria and therefore have a high probability of acceptance, and successive proposals are decorrelated by the Hamiltonian proposal distribution. Furthermore, the CN solver is a well-studied space-time discretisation of the PDE solution with a rich convergence theory and quantifiable error \[153\]. As a result, we obtain a posterior sample that has the accuracy and convergence guarantees associated with a CN solver, but at a significantly lower computational cost than is possible by using CN in a typical Metropolis–Hastings sampler.

The remainder of this work proceeds as follows. In Section \[3.2\] we fully outline the
Bayesian inverse problem for the Biot number that we consider throughout this work. Section 3.3 describes our methodology in various parts, beginning with how to specify a Gaussian process prior for the Biot number within a deep-learning surrogate, proceeded by a description of our adaptive training scheme, then the deep-surrogate-accelerated delayed-acceptance HMC sampling. In Section 3.4, we apply this methodology to simulated and real experimental data. Our experiments compare the efficiency and accuracy of our adaptively trained surrogate to a surrogate trained over a more general set of parameters, and quantifies the statistical accuracy of different sampling schemes in terms of the effective sample size (ESS) obtained. These experiments are carried out with and without the delayed-acceptance step, and the accuracy of the various sampling schemes and “surrogate-only” approaches (without delayed-acceptance) are visualised through posterior density plots.

### 3.2 Problem specification

We consider heat transfer in rotating disc systems. To model the disc, we make an axisymmetric assumption in one spatial dimension representing radial location, and consider the evolution of the heat profile over time. After non-dimensionalisation, an appropriate PDE for the temperature of the disc is the transient fin equation

\[
\frac{c_0}{\partial t} u(t, x) = c_1 \frac{\partial^2 u}{\partial x^2}(t, x) + \frac{c_2}{x} \frac{\partial u}{\partial x}(t, x) - Bi(t, x) u(t, x), \quad t \in [0, T], x \in [a, b],
\]

for \(0 < a < b\) and fixed parameters \(c_0, c_1, c_2 > 0\). To this equation, we prescribe Dirichlet boundary conditions

\[
\begin{aligned}
  u(t, a) &= u_a(t), & t \in [0, T], \\
  u(t, b) &= u_b(t), & t \in [0, T],
\end{aligned}
\]

and an initial condition

\[
  u(0, x) = u_0(x), \quad x \in [a, b].
\]

Given data \(\{(\hat{t}_n, \hat{x}_n, \hat{z}_n) : n = 1, 2, \ldots, N\}\), our goal is to perform Bayesian uncertainty quantification for the spatio-temporal parameter \(Bi(t, x)\) in the transient fin equation known as the Biot number. Here \((\hat{t}_n, \hat{x}_n)\) represent space-time coordinates and \(\hat{z}_n\) represent temperature measurements corresponding to these coordinates.
We assume the data is related to the PDE through the statistical model

\[ \hat{z}_n = u(\hat{t}_n, \hat{x}_n) + \epsilon_n, \quad n = 1, \ldots, N, \]  

where \( u(\cdot, \cdot) \) is the solution to \( (3.2.1, 3.2.2, 3.2.3) \) and the noise terms \( \epsilon_n \sim N(0, \sigma^2_\epsilon) \) are i.i.d. Gaussian random variables with unknown standard deviation \( \sigma_\epsilon \). Under this model, the probability of observing the data for fixed parameters \( Bi(t, x), \sigma_\epsilon \) is given by the likelihood function

\[ p(\hat{z} | \hat{t}, \hat{x}, Bi, \sigma_\epsilon) = \frac{1}{(2\pi\sigma^2_\epsilon)^{N/2}} \exp \left( -\frac{1}{2\sigma^2_\epsilon} \sum_{n=1}^{N} (\hat{z}_n - u(\hat{t}_n, \hat{x}_n))^2 \right). \]  

(3.2.5)

To perform Bayesian inference, we must define a prior distribution \( p(Bi, \sigma_\epsilon) \) over the unknown parameters \( Bi(t, x) \) and \( \sigma_\epsilon \). The choice of prior distribution is important since it can be used as a regulariser to ensure that we recover physically meaningful results. Here we will assume that \( Bi(t, x) \) and \( \sigma_\epsilon \) are independent in the prior distribution, so that \( p(Bi, \sigma_\epsilon) = p_B(Bi)p_\sigma(\sigma_\epsilon) \) and we define \( p_\sigma(\sigma_\epsilon) \) to be a Gamma distribution. For \( p_B(Bi) \), we require the prior distribution to be defined over the space of 2-dimensional functions, and for this we choose the popular option of assigning a Gaussian process prior

\[ Bi \sim GP(\mu(\cdot), K(\cdot, \cdot)). \]  

(3.2.6)

Here the mean function \( \mu(\cdot) \) and covariance kernel \( K(\cdot, \cdot) \) are manually specified such that they represent our prior beliefs about the behaviour of the function. In this work, we discretise the Gaussian process by a finite random-series expansion and the infinite-dimensional prior \( p_B(Bi) \) is represented explicitly by a density over finitely many discretisation parameters.

Having decided the prior distribution and likelihood \( (3.2.5) \), an application of Bayes rule gives us the posterior distribution

\[ p(Bi, \sigma_\epsilon | \hat{t}, \hat{x}, \hat{z}) = \frac{p(Bi, \sigma_\epsilon)p(\hat{z} | \hat{t}, \hat{x}, Bi, \sigma_\epsilon)}{p(\hat{z} | \hat{t}, \hat{x})} \propto p(Bi, \sigma_\epsilon)p(\hat{z} | \hat{t}, \hat{x}, Bi, \sigma_\epsilon). \]  

(3.2.7)

Approximating the posterior distribution is the aim of the Bayesian inverse problem. This measure represents the full conditional distribution of all unknown parameters given the observed data and prior information. Analysis of this distribution provides a
robust and natural form of uncertainty quantification over the parameters, whilst also providing a complete picture of any correlations, skews, or heavy tails present in the distribution that might be important when assessing the results in practice.

3.3 Methodology

Our methodology broadly consists of 3 stages:

1. Represent $Bi(t, x)$ by a series expansion and compute the distribution over the coefficients such that the series approximates a Gaussian process prior.

2. Train a deep learning surrogate model to approximate the parametric forward problem on an appropriate measure over the series coefficients.

3. Apply HMC within a Crank–Nicolson delayed-acceptance scheme to accurately sample from the posterior distribution using the computed prior distribution and deep learning surrogate.

Here we describe each of these steps in detail, outlining various options and their relative benefits in practice.

3.3.1 Prior approximation

We represent the functional parameter $Bi(t, x)$ through the Chebyshev expansion

$$
\hat{Bi}(t, x) = \sum_{i=1}^{M} \alpha_i T_i(t, x).
$$

(3.3.1)

In this expression the $T_i(t, x)$ are two dimensional Chebyshev polynomial basis functions. The maximum degree of these Chebyshev basis functions is $D$, therefore the number of basis terms is $M = \frac{1}{2}(D + 1)(D + 2)$, where for each $k, l \in \{0, 1, \ldots, D\}$ such that $k + l \leq D$ there is a unique $i \in \{1, \ldots, M\}$ such that $T_i(t, x) = T_k(t)T_l(x)$. $T_k(t), T_l(x)$ are the shifted Chebyshev polynomials of the first kind onto the domains $t \in [0, T]$ and $x \in [a, b]$ respectively. The coefficients $\alpha \in \mathbb{R}^M$ are parameters that we wish to infer using data, and will be given as inputs to the deep learning surrogate model. Given this representation of $Bi(t, x)$ as a linear basis expansion, we seek to define a meaningful prior distribution over the coefficients of this expansion. The
Gaussian process class of distributions is a natural target, as it allows us to define prior distributions over functions in a way that is easily interpretable through the specification of the mean function and covariance kernel. We ensure that \( \hat{B}_i(t, x) \) is itself a Gaussian process by assigning a multivariate normal prior over the coefficients

\[
\alpha \sim \text{MVN}(m, \Sigma), \quad (3.3.2)
\]

where \( m \in \mathbb{R}^M \) and \( \Sigma \in \mathbb{R}^{M \times M} \) are the mean vector and covariance matrix.

The mean function of \( \hat{B}_i(t_j, x_j) \) is straightforward to calculate as

\[
\hat{\mu}(t, x) = \sum_{i=1}^{M} m_i T_i(t, x), \quad (3.3.3)
\]

and the covariance function is

\[
\hat{K}([t, x], [t', x']) = \sum_{i,j=1}^{M} T_i(t, x) \Sigma_{i,j} T_j(t', x'). \quad (3.3.4)
\]

We can then approximate a Gaussian process using (3.3.1) by approximating the desired deterministic mean and covariance using (3.3.3, 3.3.4). More explicitly, suppose that we wish to approximate the Gaussian process \( B_i(t, x) \sim \mathcal{GP}(\mu(t, x), K([t, x], [t', x'])) \); we can compute the mean vector \((m_1, \ldots, m_M)\) such that

\[
\sum_{i=1}^{M} m_i T_i(t, x) \approx \mu(t, x), \quad (3.3.5)
\]

and the covariance matrix entries \( (\Sigma_{1,1}, \Sigma_{1,2}, \ldots, \Sigma_{M,M}) \) such that

\[
\sum_{i,j=1}^{M} T_i(t, x) \Sigma_{i,j} T_j(t', x') \approx K([t, x], [t', x']). \quad (3.3.6)
\]

The computation of \( m \) and \( \Sigma \) are function approximation problems that are achieved by interpolation on Chebyshev nodes. This is guaranteed to be close to the optimal polynomial in the maximum norm for any continuous kernel function, and is computationally efficient due to its reliance only on interpolation \[155\]. Another approach that we could apply is the Karhunen–Loève (KL) expansion \[99\]. In the KL expansion the basis functions and the distribution of the coefficients are solutions to an eigenvalue problem, and the corresponding approximation is the best linear approximation to the
Gaussian field in terms of the mean-squared error. We note that while the KL expansion defines a more accurate approximation to the GP, it is inconvenient in practice as it requires us to train a new surrogate with different basis functions each time we change the prior distribution. By instead taking the approach described above, we require only one surrogate based on Chebyshev polynomials, and if we wish to change the prior we need only recompute mean vector and covariance matrix of the random vector \( \alpha \).

The resultant distribution of \( \hat{B}(t, x) \) is a Gaussian process with mean function and covariance kernel approximately equal to those of the original Gaussian process \( B(t, x) \), however it is expressed entirely by a multivariate normal distribution over the coefficients \( p(\alpha) \), with mean and covariance according to (3.3.2). We can subsequently use this as the prior distribution for our method. The corresponding posterior distribution is therefore

\[
p(\alpha, \sigma^2 | \hat{t}, \hat{x}, \hat{z}) = \frac{p(\alpha, \sigma^2) p(\hat{z}|\hat{t}, \hat{x}, \alpha, \sigma^2)}{p(\hat{z}|\hat{t}, \hat{x})} \propto p(\alpha, \sigma^2) p(\hat{z}|\hat{t}, \hat{x}, \alpha, \sigma^2),
\]

(3.3.7)

where \( p(\alpha, \sigma^2) = p_\alpha(\alpha)p_\sigma(\sigma^2) \), and \( p(\hat{z}|\hat{t}, \hat{x}, \alpha, \sigma^2) \) is the likelihood function

\[
p(\hat{z}|\hat{t}, \hat{x}, \alpha, \sigma^2) = \frac{1}{(2\pi \sigma^2)^{N/2}} \exp \left( -\frac{1}{2 \sigma^2} \sum_{n=1}^{N} (\hat{z}_n - u(\hat{t}_n, \hat{x}_n, \alpha))^2 \right).
\]

(3.3.8)

Here \( u(\hat{t}, \hat{x}, \alpha) \) is the parametric solution to the PDE over the coefficients \( \alpha \), which will be approximated by a deep surrogate model in our method.

### 3.3.2 Deep surrogate approximation

Now we turn to approximating the parametric solution to the fin equation (3.2.1) over the parameters \( \alpha \). Let us define the interior PDE domain \( \Omega = [0, T] \times [a, b] \), and associate to it a positive measure \( \pi^\Omega \). Similarly define the boundary domain \( \partial \Omega = [0, T] \times \{a, b\} \cup \{0\} \times [a, b] \) with positive measure \( \pi^b \), and note that the boundary and initial conditions (3.2.2, 3.2.3) can be combined into a single Dirichlet condition

\[
u(t, x) = u_{BC}(t, x), \quad (t, x) \in \partial \Omega.
\]

To construct the deep surrogate model, we approximate the parametric solution to (3.2.1) using a neural network \( \hat{u}(t, x, \alpha) : \Omega \times \mathbb{R}^M \to \mathbb{R} \). We associate a positive measure
πα on \( \mathbb{R}^M \) to these coefficients, and train the network using stochastic gradient descent to minimise the loss function

\[
\text{Loss} = \int_{\mathbb{R}^M} F(\alpha) \, d\pi^\alpha(\alpha),
\]

\[
F(\alpha) = \nu_1 \| \mathcal{L} \hat{u}(\cdot, \cdot, \alpha) - b(\cdot) \|_{L_2(\Omega, \pi)}^2
+ \nu_2 \| \hat{u}(\cdot, \cdot, \alpha) - u_{BC}(\cdot, \cdot, \alpha) \|_{L_2(\partial \Omega, \pi^b)}^2,
\]

(3.3.10)

for some parameters \( \nu_1, \nu_2 > 0 \). Here \( b = 0 \), and \( \mathcal{L} \hat{u} \) is the differential operator of the fin equation applied to the surrogate

\[
\mathcal{L} \hat{u}(t, x, \alpha) = \frac{\partial^2 \hat{u}}{\partial x^2}(t, x, \alpha) + \frac{1}{x} \frac{\partial \hat{u}}{\partial x}(t, x, \alpha) - \hat{B} \hat{u}(t, x, \alpha) - \frac{\partial \hat{u}}{\partial t}(t, x, \alpha).
\]

(3.3.11)

We use the subscript \( L_2(C, \mu) \) notation to denote the \( L_2 \)-norm over domain \( C \) with respect to the measure \( \mu \)

\[
\| h \|_{L_2(C, \mu)}^2 = \int_C |h(x)|^2 d\mu(x).
\]

(3.3.12)

The integrals in (3.3.10) are intractable, so in practice this minimisation is implemented by drawing randomised collocation points \((t, x, \alpha)\) from the interior measure \( \pi \otimes \pi^\alpha \) and boundary points \((t_b, x_b, \alpha_b)\) from \( \pi^b \otimes \pi^\alpha \), then minimising the Monte Carlo approximation of the integral in (3.3.12) induced by these points. This reduction is achieved by taking a gradient descent step, and after each step a new random sample of collocation points is drawn and the process repeated. This approach is an extension of the Deep Galerkin Method first introduced for static parameter values in [144].

We will take \( \pi \) and \( \pi^b \) to be uniform on \( \Omega \) and \( \partial \Omega \) respectively throughout this work. The parameter measure \( \pi^\alpha \) however is a key ingredient whose specification can have a significant impact on both the accuracy of the surrogate and the efficiency of its training. It has been shown that the number of neural network parameters required to accurately approximate the parametric solution to several classes of PDE depends only on the intrinsic dimension of the solution manifold [90]. This is in contrast to a dependency on the dimensionality of PDE parameters themselves, meaning solutions approximated using this method are capable of overcoming the curse of dimensionality associated with large numbers of parameters if the solution over the training domain has a low intrinsic dimension. The measure \( \pi^\alpha \) influences the intrinsic dimensionality of the solution manifold and so its specification is important.
Previous work has set $\pi^\alpha$ to be a uniform distribution on a compact subset of $\mathbb{R}^M$ [38], producing a general surrogate spanning a relatively large parameter space. This has the advantage that the surrogate need only be trained once, resulting in an analytic function that can be stored and applied to various datasets, however this surrogate will typically be expensive to train and less accurate than a more focused measure. In Section 3.4.2 we demonstrate that this is the case in our setting, and that using a general surrogate leads to inaccurate inferences for this problem. An alternative approach is to set $\pi^\alpha$ to be the posterior distribution. By definition, this measure weights the training across parameter space exactly as we require, and since this is concentrated around parameters that achieve a good fit to the data the corresponding solution manifold will have a lower intrinsic dimension. Of course this approach assumes that we know the posterior distribution a-priori, which is not the case in practice, and thus requires an adaptive training regime that trains the surrogate and approximates the posterior simultaneously.

**Data adaptive training**

Our approach to training the surrogate seeks to set the parameter measure $\pi^\alpha$ in the loss function (3.3.10) to be the posterior distribution (3.3.7). This is ultimately achieved by using MCMC samples from the posterior as training points for the surrogate. However, this approach raises the issue of how to initially generate these samples given that we require a trained surrogate to begin MCMC. Our solution is to first train the surrogate over the Laplace approximation to the posterior distribution

$$\pi^{\alpha}_{\text{Laplace}} = \mathcal{MVN}(\alpha^*, \text{Hess}_{\alpha}^{-1} \{- \log p(\alpha^*, \sigma^*|\hat{t}, \hat{x}, \hat{z})\}),$$  

(3.3.13)

where $\alpha^*, \sigma^*$ are the MAP estimates

$$(\alpha^*, \sigma^*_t) = \arg\max \{ \log p(\alpha, \sigma_t|\hat{t}, \hat{x}, \hat{z})\}.$$  

(3.3.14)

The MAP optimisation is carried out by beginning with some initial estimate $(\alpha_0, \sigma_{0t})$ and iteratively updating this estimate using gradient ascent. To approximate the gradients to the objective function (3.3.14) at iteration $n$, the deep surrogate approximation is updated to approximate the solution over a local radius surrounding $\alpha_n$, and automatically differentiated. These gradients are applied to update to $\alpha_{n+1}$, and the process is repeated. The size of the local radius and the step size is reduced as we approach the MAP estimate in a manner similar to trust region optimisation (see the
appendix for more detail). Given the MAP estimate \((\alpha^*, \sigma^*_\epsilon)\) and the corresponding locally trained surrogate \(\hat{u}^*(t, x, \alpha)\), the covariance of the Laplace approximation to the posterior distribution is readily available by automatic differentiation, thus we next train a surrogate over this approximation by using samples from the Laplace approximation as training points. This initial surrogate is then utilised within an MCMC scheme to begin sampling from the posterior distribution.

Figure 3-1: Illustration of the adaptive training scheme. At iteration \(n\) the surrogate is trained locally by minimising the loss (3.3.10) with local measure \(\pi^\alpha = \pi^\alpha_n\) as depicted by the grey measures. The gradient \(\nabla_\alpha \hat{u}\) of the locally trained surrogate is then used to update to \(\alpha_{n+1}\) by gradient ascent. Upon convergence to the MAP estimate \(\alpha^*\), the surrogate is trained over the Laplace approximation to the posterior shown in red (by minimising (3.3.10) with \(\pi^\alpha = \pi^\alpha_{\text{Laplace}}\)). The surrogate is then ready to begin sampling.

Although the Laplace approximation provides a reasonable estimate of the posterior distribution for many applications, it may be inaccurate if the true posterior deviates too far from a Gaussian. To account for this and ensure that the training measure \(\pi^\alpha\) is as close as possible to the posterior distribution, we continue training the surrogate online using posterior samples from the MCMC scheme as training points for the loss function (3.3.10). This additional refinement of the approximation is interleaved with the MCMC sampler during a warm-up period, after which the surrogate is fixed and used to sample from the posterior distribution.
We can understand the relation between the training loss function and the accuracy of the posterior distribution induced by the adaptively trained surrogate in the Hellinger metric as follows. Let us denote the true posterior by \( \psi(\theta) \) and let \( \psi_0(\theta) \) be the prior distribution so that

\[
d\psi = \frac{1}{Z} \exp(-\Phi(\theta)) d\psi_0(\theta).
\] (3.3.15)

Here \( \theta = (\alpha, \sigma_\epsilon) \), \( \Phi(\theta) = \frac{1}{2\sigma^2} \sum_{n=1}^{N} (\hat{z}_n - u(\hat{t}_n, \hat{x}_n, \alpha))^2 \) and \( Z \) is a normalisation constant. Denote the approximated posterior induced by substituting the surrogate into the likelihood by \( \hat{\psi}(\theta) \), and its associated potential by \( \hat{\Phi}(\theta) \).

Let \( \frac{d\psi}{d\psi_0} = f \) and \( \frac{d\hat{\psi}}{d\psi_0} = \hat{f} \) be Radon–Nikodym derivatives with respect to \( \psi_0 \). Then the squared Hellinger distance between \( \psi \) and \( \hat{\psi} \) is

\[
H^2(\psi, \hat{\psi}) = \frac{1}{2} \int (\sqrt{f(\theta)} - \sqrt{\hat{f}(\theta)})^2 d\psi_0(\theta)
= \frac{1}{2} \int (\sqrt{f(\theta)/\hat{f}(\theta)} - 1)^2 d\hat{\psi}(\theta).
\] (3.3.16)

Assuming \( \Phi(\theta) \approx \hat{\Phi}(\theta) \) and \( Z \approx \hat{Z} \), we can approximate

\[
H^2(\psi, \hat{\psi}) = \frac{1}{2} \int \left( \sqrt{\frac{Z}{\hat{Z}}} \exp \left( \frac{1}{2} (\Phi(\theta) - \hat{\Phi}(\theta)) \right) - 1 \right)^2 d\hat{\psi}(\theta)
\approx \frac{1}{8} \int \left( \Phi(\theta) - \hat{\Phi}(\theta) \right)^2 d\hat{\psi}(\theta).
\] (3.3.17)

By assuming that \( u \) and \( \hat{u} \) are bounded uniformly, we have for some \( M > 0 \) that

\[
|\Phi(\theta) - \hat{\Phi}(\theta)| = \left| \frac{1}{2\sigma^2} \sum_{n=1}^{N} (\hat{u}(\hat{t}_n, \hat{x}_n, \alpha) - u(\hat{t}_n, \hat{x}_n, \alpha))(\hat{u}(\hat{t}_n, \hat{x}_n, \alpha) + u(\hat{t}_n, \hat{x}_n, \alpha) - 2\hat{z}_n) \right|
\leq \frac{M}{2\sigma^2} \sum_{n=1}^{N} |(\hat{u}(\hat{t}_n, \hat{x}_n, \alpha) - u(\hat{t}_n, \hat{x}_n, \alpha))|,
\] (3.3.18)

using \((a - c)^2 - (b - c)^2 = (a - b)(a + b - 2c)\). So applying Jensen’s inequality we have that the Hellinger distance approximately satisfies

\[
H^2(\psi, \hat{\psi}) \leq \frac{NM^2}{32} \int \frac{1}{\sigma^2} \sum_{n=1}^{N} (\hat{u}(\hat{t}_n, \hat{x}_n, \alpha) - u(\hat{t}_n, \hat{x}_n, \alpha))^2 d\hat{\psi}(\theta).
\] (3.3.19)
Now, \( \hat{u} \) satisfies a second-order PDE similar to (3.2.1) with perturbed initial and boundary data and an extra forcing term (by substituting \( \hat{u} \) into (3.2.1)). We may apply \( L^2 \)-stability theory for second-order parabolic equations (e.g., [130]), to determine that,

\[
\frac{1}{N} \sum_{n=1}^{N} (\hat{u}(\hat{t}_n, \hat{x}_n, \alpha) - u(\hat{t}_n, \hat{x}_n, \alpha))^2 \approx \frac{1}{A} \| \hat{u}(\cdot, \cdot, \alpha) - u(\cdot, \cdot, \alpha) \|_{L^2(\Omega)}^2 \leq F(\alpha),
\]

(3.3.20)

where \( F \) is the size of the perturbed boundary data and extra forcing term as defined in (3.3.10) with appropriate weighting \( \nu_1, \nu_2 \). Here we have approximated \( \frac{1}{N} \sum_{n=1}^{N} (\hat{u}(\hat{t}_n, \hat{x}_n, \alpha) - u(\hat{t}_n, \hat{x}_n, \alpha))^2 \) by \( \frac{1}{A} \| \hat{u} - u \|_{L^2(\Omega)}^2 \), where \( A \) is the area of \( \Omega \), which is reasonable in our settings where data is collected at regular space-time intervals. Then, for some \( C > 0 \),

\[
H^2(\psi, \hat{\psi}) \leq C \int \frac{1}{\sigma_\epsilon^4} F(\alpha) d\hat{\psi}(\alpha, \sigma_\epsilon).
\]

(3.3.21)

The right-hand side corresponds to the training loss function (3.3.10) when the prior on the hyper-parameter \( \sigma_\epsilon \) is trivial (\( \sigma_\epsilon \) is treated as known) and describes the appropriate generalisation of (3.3.10) to non-trivial priors on \( \sigma_\epsilon \). When evaluated with MCMC samples from the surrogate posterior, therefore, it provides an approximate bound on the posterior error in the Hellinger metric. A rigorous treatment that quantifies the effect of the \( \Phi(\theta) \approx \hat{\Phi}(\theta) \) and \( Z \approx \hat{Z} \) assumptions can also be achieved following the methodology in Chapter 4 of [147].

### 3.3.3 Approximating the posterior using delayed-acceptance HMC

The main aim in a Bayesian inverse problem is to infer the posterior distribution. Most commonly for PDE-based problems these approaches will be based upon optimisation of parameterised approximations to the target distribution. Such methods — which include variational techniques and Laplace approximations — have the advantage of being numerically tractable using traditional techniques for PDEs. However this approximation to the target distribution is based upon assumptions made about its shape a-priori. Alternatively, Markov chain Monte Carlo methods empirically approximate the target by sampling directly from it. The resulting approximations can be made arbitrarily accurate by the law of large numbers, but require a large number of decorrelated samples. This requirement has restricted the application of MCMC to PDE-based inverse problems as each MCMC iteration requires the PDE to be solved numerically, which is an expensive undertaking.
The major advantages of deep neural network approximations to the parametric solutions of PDEs is that they can efficiently be evaluated and differentiated with respect to the PDE parameters using automatic differentiation within dedicated software such as TensorFlow [1]. This ease of manipulation of the deep surrogate model affords us great flexibility in how we approximate the posterior distribution. For example, one can quickly establish a Laplace approximation as described in Section 3.3.2, or use their differentiability to efficiently implement gradient-based MCMC schemes such as Hamiltonian Monte Carlo (HMC). Conversely, a current disadvantage of deep learning-based PDE solvers is a lack of convergence guarantees. Despite the remarkable approximation capabilities of neural networks for many classes of PDEs, our ability to realise these approximations typically relies on non-convex optimisation using stochastic gradient descent, and as a result we are currently unable to guarantee the accuracy of deep learning surrogates.

In the following we introduce the sampling scheme that we will apply to infer \( \hat{B}i(t, x) \).

Our scheme efficiently combines the gradient-based proposal distributions defined by the Hamiltonian Monte Carlo method with the guaranteed accuracy of a Crank–Nicolson solver using a delayed-acceptance verification step. We begin by describing Metropolis–Hastings MCMC, and then highlight how a Hamiltonian delayed-acceptance proposal distribution can be incorporated to improve sampling efficiency and accuracy.

**Metropolis–Hastings Markov chain Monte Carlo**

A Markov chain is a discrete-time stochastic process \( X_0, X_1, X_2, \ldots \) such that \( X_{n+1} \) given \( X_n \) is independent of all other previous states. In our application we wish to sample from the posterior distribution over the parameters \( \theta = (\alpha, \sigma_\epsilon) \), accordingly let us consider a stochastic process \( \theta_0, \theta_1, \theta_2, \ldots \) taking values in the parameter space \( \Theta \subset \mathbb{R}^M \times \mathbb{R}^+ \). If the process satisfies the Markov property

\[
g(\theta_{n+1}|\theta_n, \ldots, \theta_0) = g(\theta_{n+1}|\theta_n),
\]

then it is a Markov chain, and \( g(\theta_{n+1}|\theta_n) \) is the transition kernel of the process (the probability density given \( \theta_n \), of transitioning from \( \theta_n \) to \( \theta_{n+1} \)). The stationary distribution of a Markov chain is a probability measure \( \psi \) such that \( \psi(\theta) \geq 0 \quad \forall \theta \in \Theta, \quad \int_{\Theta} d\psi(\theta) = 1 \) and \( \int_B d\psi(\theta) = \int_{\Theta} g(B|\theta)d\psi(\theta) \) for any \( B \subset \Theta \). Here the left hand side is the probability of \( B \), whereas the right side is the probability that transi-
tions into $B$ from $\psi$ in one time step. Their equality implies that $\psi$ remains constant after transitioning.

Markov chain Monte Carlo methods are a group of algorithms for sampling from a distribution known up to an arbitrary constant of proportionality. This is achieved by simulating ergodic Markov chains for which the target distribution is their stationary distribution. The states visited by the process then form a sample from this distribution and the empirical density of these states converges in distribution to the target. A sufficient condition for constructing an appropriate Markov chain is detailed balance, which states that $g(\theta'|\theta)\psi(\theta) = g(\theta|\theta')\psi(\theta')$ holds for any $\theta, \theta' \in \Theta$. It can be shown that if $(g, \psi)$ satisfies detailed balance, then $\psi$ is the stationary distribution of the process. In MCMC methods we construct transition kernels that achieve detailed balance for a given distribution. In particular if we choose the target distribution as $\psi(\theta) = p(\theta|\hat{t}, \hat{x}, \hat{z})$ defined in (3.3.7) then this allows us to sample from the posterior. One of the simplest approaches to the construction of appropriate transition kernels is the Metropolis–Hastings (MH) algorithm.

In the Metropolis–Hastings algorithm we choose a proposal density $q(\theta_{\text{prop}}|\theta)$ describing the probability of proposing a transition from $\theta$ to $\theta_{\text{prop}}$. This proposal is then accepted with probability

$$A_M(\theta_{\text{prop}}, \theta) = \min\left\{1, \frac{q(\theta|\theta_{\text{prop}})\psi(\theta_{\text{prop}})}{q(\theta_{\text{prop}}|\theta)\psi(\theta)}\right\}.$$  (3.3.23)

It is readily verifiable that detailed balance is preserved with respect to $\psi$ if the transitions are carried out this way. The full algorithm can be written as:

\begin{algorithm}[H]
\caption{Metropolis-Hastings}
Choose initial $\theta_0 \in \Theta$;
\For{$i = 0, 1, \ldots, N_{\text{samples}}$ do}
    Propose $\theta_{\text{prop}} \sim q(\theta_{\text{prop}}|\theta_i)$;
    Sample $u \sim U(0,1)$;
    \If{$u \leq A_M(\theta_{\text{prop}}, \theta_i)$}{
        set $\theta_{i+1} := \theta_{\text{prop}}$ (accept);
    }\Else{
        set $\theta_{i+1} := \theta_i$ (reject);
    }
\EndFor
return $\theta = (\theta_0, \theta_1, \ldots, \theta_{N_{\text{samples}}})$
\end{algorithm}

105
The choice of proposal distribution $q(\theta_{\text{prop}}|\theta)$ is key to the success of MH. An efficient proposal distribution will yield samples that are not heavily correlated, resulting in a sample that estimates the posterior with a lower statistical error. For a sample generated using some proposal distribution, the effective sample size (ESS) quantifies the efficiency of the proposal by returning the number of i.i.d samples required to reach the same level of precision \[54\]. In practice simple symmetric proposal distributions such as the multivariate Gaussian distribution centred at $\theta$ are commonly used. This random walk Metropolis–Hastings (RWMH) scheme has the advantage of simplicity, as it requires just one PDE solve per iteration and results in an acceptance probability that is independent of the proposal distribution. Unfortunately exploration strategies based on symmetric random walks such as this typically result in Markov chains that have a low acceptance rate and highly correlated samples in higher dimensions, leading to a low ESS and slow convergence to the stationary distribution.

Alternative approaches — such as Hamiltonian Monte Carlo (HMC) and Metropolis Adjusted Langevin (MALA) — use gradients of the target distribution to improve the efficiency of MCMC. For brevity we reserve a detailed treatment of HMC to the appendix. Put briefly, in HMC we define a Hamiltonian function using the target distribution by

$$H(\theta, p) = -\log \psi(\theta) - \log \psi(p|\theta)$$

$$= U(\theta) + K(p|\theta). \quad (3.3.24)$$

Here $U(\theta) = -\log \psi(\theta)$ represents the potential energy and $K(p|\theta) = -\log \psi(p|\theta)$ represents the kinetic energy of the system, where $\psi(p|\theta)$ is a user specified distribution over some auxiliary momentum variables. Proposals are then generated by using symplectic integrators to simulate random trajectories from the corresponding Hamiltonian system. This has the effect of significantly reducing the autocorrelation of the chain. It is impractical to perform HMC using traditional PDE solvers since $\nabla_{\theta} \psi(\theta)$ is required at each step of the symplectic integrator, leading to tens or hundreds of PDE solves being performed for each MCMC sample. MALA, by comparison, simulates the Langevin dynamics of a process that has stationary distribution $\psi(\theta)$, and can be shown to be equivalent to performing HMC with trajectories simulated over just one time step. MALA therefore represents a more tractable intermediary between random walk and Hamiltonian proposals, requiring just two gradients per iteration which can be calculated using adjoint PDE methods. This yields some improvement over symmetric
random walk proposals, however we demonstrate that it is significantly outperformed by HMC, thus justifying the need for efficient HMC methods for PDEs.

**Delayed-acceptance HMC**

With deep surrogate models the burden of repeated differentiation of the PDE solution is alleviated by automatic differentiation, therefore HMC can be performed efficiently. Unfortunately the accuracy of deep surrogate solutions cannot be mathematically guaranteed as is the case with traditional numerical solvers such as Crank–Nicolson. This inaccuracy has the potential to bias the posterior sample, leading to unreliable estimates that cannot safely be used in turbine design. The delayed-acceptance HMC sampler that we describe here overcomes this by performing Metropolis–Hastings using a Crank–Nicolson solver with a proposal distribution constructed such that:

1. HMC proposals based on the deep learning surrogate are utilised for efficient exploration of the posterior distribution.
2. The Crank–Nicolson solver is only executed if the proposal is accepted according to the regular HMC acceptance criteria based on the surrogate.

The delayed-acceptance method was first described in [29]. To describe our adaptation of this method, let us first denote the posterior distribution induced by substituting the deep surrogate model \( \hat{u}(\hat{t}_n, \hat{x}_n, \alpha) \) into the likelihood (3.3.8) by \( \hat{\psi}(\theta) \), and similarly denote by \( \tilde{\psi}(\theta) \) the posterior induced by the Crank–Nicolson solution \( \tilde{u}(\hat{t}_n, \hat{x}_n, \alpha) \).

In the delayed-acceptance HMC scheme we sample from \( \tilde{\psi}(\theta) \) using Metropolis–Hastings with proposal distribution given by

\[
q(\theta_{\text{prop}}|\theta) = A_H(\theta_{\text{prop}}, \theta)q_H(\theta_{\text{prop}}|\theta) \\
+ \left(1 - \int (A_H(\theta_{\text{prop}}, \theta)q_H(\theta_{\text{prop}}|\theta)) d\theta_{\text{prop}}\right) \delta(\theta - \theta_{\text{prop}}).
\]

(3.3.25)

Here \( q_H(\theta_{\text{prop}}, \theta) \) and \( A_H(\theta_{\text{prop}}, \theta) \) are the HMC proposal distribution and acceptance probability applied using \( \hat{\psi}(\theta) \). The first term in (3.3.25) is therefore the density of HMC proposals that are accepted, while the second term assigns the remaining probability to the chain remaining in place at \( \theta \).

It is natural to execute this proposal in two stages. During the first, a regular HMC
proposal is made with standard acceptance criteria \(3.5.6\), this acts as preliminary screening by the surrogate. If rejected by this criteria, the Markov chain remains in the same state and the Crank–Nicolson solver is not executed. If accepted, the proposal is passed to a secondary acceptance criteria with acceptance probability

\[
A(\theta_{\text{prop}}, \theta) = \min \left\{ 1, \frac{\tilde{\psi}(\theta_{\text{prop}}) \hat{\psi}(\theta)}{\hat{\psi}(\theta) \tilde{\psi}(\theta_{\text{prop}})} \right\}.
\]  

(3.3.26)

This additional acceptance criteria ensures detailed balance is achieved with respect to \(\tilde{\psi}\), thus the posterior is accurate to the level of the Crank–Nicolson solver. One can verify this by substituting \(3.3.25\) into the standard Metropolis acceptance criteria \(3.3.23\) and simplifying (see \[46\] for the full calculation).

The sampling efficiency and preliminary screening offered in the delayed-acceptance HMC algorithm ensures we are able to maximise the utility of the Crank–Nicolson solves. This allows us to tractably approximate the posterior by MCMC with the guaranteed accuracy that is critical in practice. There is of course some additional computational effort in comparison to relying purely on the surrogate, which we quantify in our simulation study below, as well as comparing to the performance of sampling based purely on surrogates.

### 3.4 Numerical results

The remainder of this work is dedicated to demonstrating deep surrogate accelerated delayed-acceptance HMC on real and simulated data. We first outline the specifics of our implementation, detailing the choice of prior, network architecture, training strategy and HMC calibration. Using simulated data we then demonstrate the speed and accuracy of our approach in comparison to alternative methods, before concluding this section with some results obtained using data generated by an experimental rig. All of the results and timings presented were produced in TensorFlow using a machine with a mobile RTX 2080 and a 6 core 3.9Ghz processor.
3.4.1 Implementation details

Prior distribution

We assign a Gaussian process prior distribution to $Bi(t, x)$ with mean equal to zero $\mu(t, x) = 0$, and a separable covariance kernel in space and time

$$K([t, x], [t', x']) = \sigma^2 K_x(x, x')K_t(t, t').$$ \hspace{1cm} (3.4.1)

For the spatial covariance $K_x(x, x')$ we adopt the twice differentiable Matérn kernel

$$K_x(x, x') = \left(1 + \frac{|x - x'|}{\rho_x} + \frac{|x - x'|^2}{3\rho_x^2}\right) \exp\left(-\frac{|x - x'|}{\rho_x}\right).$$ \hspace{1cm} (3.4.2)

Here we choose a length scale of $\rho_x = b - a = 0.7$. For the temporal covariance we use the squared exponential kernel

$$K_t(t, t') = \exp\left(-\frac{|t - t'|^2}{2\rho_t^2}\right).$$ \hspace{1cm} (3.4.3)

Here we choose $\rho_t = 900$, representing a temporal correlation length scale of 15 minutes. The marginal standard deviation of the prior distribution is $\sigma = 100$. Using (3.3.5, 3.3.6) the corresponding prior distribution over the coefficients is $\alpha \sim MVN(0, \Sigma)$ where $\Sigma$ is chosen such that

$$\sum_{i,j=1}^{M} T_i(t, x)\Sigma_{i,j}T_j(t', x') \approx K([t, x], [t', x']).$$ \hspace{1cm} (3.4.4)

We resolve this expression using interpolation on the grid of Chebyshev nodes. The accuracy of this approximation is visualised for each of the spatial and temporal kernels in Figure 3-2.
Network architecture

We use a fully-connected feed-forward neural network consisting of 4 hidden layers of 256 neurons with tanh activation functions to approximate the parametric solution. This deep learning surrogate $\hat{u} : \mathbb{R}^{68} \rightarrow \mathbb{R}$ takes as inputs the coordinates $(t, x) \in \mathbb{R}^2$, and the Chebyshev polynomial coefficients $\alpha \in \mathbb{R}^{66}$, and returns an approximation to the corresponding PDE solution at those coordinates.
Training

To train the deep learning surrogate we minimise the loss function (3.3.10) using stochastic gradient descent until convergence. We assign uniform measures $\pi \sim \text{Unif}(\Omega)$ to the interior domain, and $\pi^b \sim \text{Unif}(\partial \Omega)$ to the boundary domain. For the parameter space measure we compare two approaches. First is an adaptive surrogate computed using the data adaptive training scheme described in Section 3.3.2, and this is compared to a general surrogate trained over $\pi^\alpha = \text{Unif}(A)$ where $A$ is the general parameter space $A = \bigcup_{i+j\leq 10} A_{i,j}$, where $A_{i,j} = [-80/2^k, 80/2^k]$ for $k = \max(i + j - 3, 0)$. This set is broad enough to cover all of the parameter values that we reasonably expect to infer, whilst also being prevented from becoming unnecessarily complex by the decay in the coefficients of higher degree terms.

HMC calibration

The surrogate is used to draw 20,000 MCMC samples from the posterior distribution. The first 10,000 of these constitute a warm-up period during which hyperparameters of the MCMC scheme are automatically determined. In this period the step size of a leap-frog integrator of the Hamiltonian dynamics is calibrated such that an acceptance rate of about 65% is achieved. Furthermore the HMC mass matrix (as defined in the appendix) is updated such that it is proportional to the inverse of the empirical covariance of the existing sample during this period. Both of these choices are generally regarded as optimal for HMC. After the warm-up period these hyperparameters are fixed, and delayed-acceptance HMC is used with a Crank–Nicolson solver to draw a further 10,000 samples from the posterior distribution. As comparisons we also perform RWMH and MALA. Similarly to our HMC implementation we use the warm up period to calibrate their proposal distribution/preconditioning matrix, and adapt the step size to target the corresponding optimal acceptance rates of 23.4% and 57.4% respectively.
3.4.2 Simulation study

To validate our method we first provide a study using simulated data. For this we consider the PDE (3.2.1) on the domain \((t, x) \in \Omega = [0, 3600] \times [0.3, 1]\) with \(c_0 = 35,000, c_1 = 1, c_2 = 1\), and initial and boundary conditions

\[
\begin{align*}
    u(0, x) &= x, & x &\in [0.3, 1], \\
    u(t, 0.3) &= 0.3, & x &\in [0.3, 1], \\
    u(t, 1) &= 1, & t &\in [0, 3600].
\end{align*}
\]  

(3.4.5)

We produce noisy data from this PDE by solving the equation for instances of \(Bi(t, x)\) using a high resolution Crank-Nicolson solver and adding Gaussian white noise to data subsampled from the solution at 152 locations as illustrated in Figure 3-3.

Our aim in the Bayesian inverse problem is to sample from the posterior distribution of \(\alpha\) conditional on this data. The instances of \(Bi(t, x)\) that we illustrate are direct
samples from a grid-based Gaussian process discretisation, that is the true $Bi(t, x)$ are not polynomials in these examples. The ill-posedness of this problem implies that there exist a wide range of Biot numbers that achieve a good fit to the data (see Figure 3-4 for some examples of this variability for the data in Figure 3-3). It is therefore unreasonable to think that we should be able to infer $Bi(t, x)$ with a very high degree of certainty. Instead what we seek is a quantification of this uncertainty that uses our prior distribution to restrict our inference to physically reasonable results.

Figure 3-4: Top: An example of a $Bi(t, x)$ that is physically unrealistic but achieves a good fit to the data shown in Figure 3-3. Middle: At discrete times the spatial profiles of this $Bi(t, x)$ are shown in blue. The profiles of the true $Bi(t, x)$ are plotted in red. The green curves are alternate examples of $Bi(t, x)$ that achieve a similar fit. Bottom: Temporal residuals between the data and solution fitted using this $Bi(t, x)$, demonstrating that the unrealistic $Bi(t, x)$ produces a good fit.
To train the adaptive surrogate we begin by finding the Laplace estimate. As shown in Figure 3-5, this provides a reasonable starting point for inferring the Biot number, however at discrete times the spatial profile reveals that the true Biot number used to generate the data is often significantly outside the 95% credible interval. This behaviour is observed consistently with different instances of $Bi(t, x)$ and so we conclude that the Laplace approximation is over-certain for this problem.

![Figure 3-5: Laplace estimate of Biot number compared to the true value.](image)

This poor quantification of the uncertainty is not unexpected, as the Laplace approximation is a local estimate based only on the posterior curvature at the MAP estimate. In particular, Figure 3-5 justifies the requirement for MCMC methods to quantify the uncertainty more reliably. In our experiments we carry out RWMH, MALA, and HMC using the general and adaptive surrogates. After a warm up period each of
these schemes are run for 10,000 iterations, both in a surrogate-only setting, and with delayed-acceptance using Crank–Nicolson. In each case, Table 3.1 shows the accuracy of the surrogate, time taken in seconds for training and sampling, as well as the effective sample size and cost (measured as the time per effective sample). In this table the timings, ESS, and costs given are the averages over runs with 5 different datasets, and the $L^1$ errors correspond to the averaged difference between the surrogate and a high resolution Crank–Nicolson solver at the MAP parameter values.

<table>
<thead>
<tr>
<th>Surrogate type</th>
<th>Proposal</th>
<th>Time</th>
<th>ESS</th>
<th>Cost</th>
<th>Time</th>
<th>ESS</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive</td>
<td>RWMH</td>
<td>70.76</td>
<td>32.90</td>
<td>2.151</td>
<td>673.62</td>
<td>20.28</td>
<td>33.216</td>
</tr>
<tr>
<td></td>
<td>MALA</td>
<td>110.84</td>
<td>427.34</td>
<td>0.259</td>
<td>799.51</td>
<td>364.97</td>
<td>2.191</td>
</tr>
<tr>
<td></td>
<td>HMC</td>
<td>379.96</td>
<td>11263.20</td>
<td>0.034</td>
<td>1337.94</td>
<td>6497.04</td>
<td>0.206</td>
</tr>
<tr>
<td>General</td>
<td>RWMH</td>
<td>68.71</td>
<td>36.69</td>
<td>1.873</td>
<td>665.35</td>
<td>23.79</td>
<td>27.978</td>
</tr>
<tr>
<td></td>
<td>MALA</td>
<td>109.32</td>
<td>372.30</td>
<td>0.294</td>
<td>797.28</td>
<td>26.85</td>
<td>29.694</td>
</tr>
<tr>
<td></td>
<td>HMC</td>
<td>389.28</td>
<td>10831.66</td>
<td>0.036</td>
<td>1359.60</td>
<td>8.25</td>
<td>74.499</td>
</tr>
</tbody>
</table>

Table 3.1: Average timings, accuracy, and ESS achieved by each surrogate type after running 10,000 iterations of each sampling scheme.

Beginning from a randomly initialised network, the time taken to train the general surrogate is 4 hours and 25 minutes, which is significantly longer than the 15 minutes taken to train the adaptive surrogate. In principle the general surrogate has the advantage of being applicable to multiple datasets once trained, however the larger error of this surrogate prevents it from being a reliable option. This is because proposals generated using the surrogate are almost always rejected by the secondary acceptance criteria in delayed-acceptance due to the discrepancy between the surrogate and the CN solution, furthermore Figure 3-9 shows that the inferences obtained by the general surrogate without delayed-acceptance are inaccurate. If available however, a general surrogate can significantly accelerate the training of an adaptive surrogate if used as the network’s initialisation. This acceleration can even apply if the general surrogate was trained to solve a version of the PDE with different boundary conditions and coefficients $\{c_0, c_1, c_2\}$; for example it takes 171 seconds on average to train an adaptive surrogate to the same level of accuracy when initialised by a general surrogate trained to solve the PDE defined for real data in Section 3.4.3 with boundary conditions shown in Figure 3-10.

Comparing sampling schemes we see that RWMH has the highest cost, followed by MALA. HMC significantly outperforms these methods despite taking longer to perform each iteration, obtaining an ESS that is slightly larger than the number of samples in
the surrogate-only setting. The introduction of the additional acceptance criteria in delayed-acceptance lowers the ESS since it reduces number of transitions that take place. Moreover, the introduction of delayed-acceptance decreases the average speed of each iteration due to the CN solver being executed. It is therefore notable that the cost of DA-HMC is still lower than surrogate-only RWMH or MALA, highlighting that the optimisation of the proposal strategy is a significant factor that, if made efficient, can outweigh numerical solver inefficiencies when performing PDE-based MCMC sampling. As a comparison, if we were to apply these sampling schemes solely using a CN solver the estimated cost is 30.40 for RWMH, 4.68 for MALA, and 8.97 for HMC. These estimates are based on the cumulative cost attributable to solving the forward and adjoint PDEs required to calculate the solutions and gradients required for each MCMC scheme, assuming that the ESS achieved is the same as the corresponding adaptive surrogate-only sampler. Although these are somewhat conservative estimates that discard any computational effort in MCMC that is not a direct result of solving the PDEs, this approach still has a cost that is over 20 times higher than our delayed-acceptance HMC scheme (which converges to the same posterior), and 130 times higher than our surrogate-only approach (which converges to a surrogate induced approximation).

To investigate the accuracy of the posteriors achieved using each method we use delayed-acceptance HMC sampling as a baseline, as this is provably accurate and achieves a sufficiently large ESS. Figure 3-6 compares the credible intervals achieved by each proposal distribution in the surrogate-only setting. Figure 3-7 compares the credible intervals achieved with and without delayed-acceptance using adaptive and general surrogates. Example trace plots for the coefficient $\alpha_3$, to the term that is cubic in space and constant in time are shown in Figure 3-8, these visualise the samples achieved by each delayed-acceptance scheme over a fixed runtime. Finally in Figure 3-9 for the same coefficient, we show how these differences in proposal distribution and surrogate choice affect the posterior accuracy by comparing the densities of their samples. These results show that in our experiments the adaptive surrogate HMC approach without delayed-acceptance achieves results that are very close to the true posterior with delayed-acceptance, albeit this behaviour is not mathematically guaranteed a-priori. All other approaches perform poorly in comparison. For RWMH and MALA with an adaptive surrogate this is due to their lower ESS. For the general surrogate the lower accuracy results in a sampling bias in the surrogate-only setting, resulting in $Bi(t,x)$ being underestimated in comparison to the true posterior. Moreover for the general surrogate, when coupled with delayed-acceptance it exhibits an extremely high rejection rate resulting in a poor Monte Carlo estimate of the posterior.
Figure 3-6: Spatial profiles of the 95% credible intervals of surrogate-only adaptive HMC, MALA, and RWMH at various time points. The red curves are the spatial profiles of the true Biot number used to simulate the data.
Figure 3-7: Spatial profiles of the 95% credible intervals of surrogate-only adaptive HMC, delayed-acceptance adaptive HMC, surrogate-only general HMC, delayed-acceptance general HMC. The red curves are the spatial profiles of the true Biot number used to simulate the data.
Figure 3-8: Comparison of trace plots of delayed-acceptance HMC, MALA, and RWMH and their 50 step moving average. The lengths of these chains is chosen so that the time taken to achieve each sample is the same.

Figure 3-9: Comparison of sample densities of different schemes for the Chebyshev coefficient that is cubic in space and constant in time.
3.4.3 Experimental data study

A current aim of engineers in the area of turbo-machinery is to develop well-informed models of the temporal heat transfer within engine cavities. This will allow time-dependent heat transfer scenarios such as take-off and landing of aircrafts to be better understood, and the resulting models will inform design optimisations that will improve engine efficiency and reduce emissions. A key method used to develop the required understanding is through the analysis of experimental data from a range of scenarios. In this section, we apply our methodology to infer the Biot number evolution using real data from a simple experimental setting of a compressor disc rotating in a compressor cavity rig. Detailed information about the experimental setting is given in [148].

![Graphs showing the evolution of Biot number with time and radius](image)

Figure 3-10: Boundary and initial conditions for the experimental data computed using cubic spline regression.

In this experiment a disc spins at 6000RPM in a closed cavity. At the outer radius $b = 0.228$ metres, the disc is heated to 91°C and maintained at that temperature for
At the inner radius \( a = 0.133 \) cool air flows through the cavity. After 500 seconds the heater is turned off and the disc continues to rotate while its temperature is monitored in 30 second increments at 18 radial locations for a further 1900 seconds. The parameters of the fin equation (3.2.1) corresponding to this setting are \( c_0 = 362, 319, c_1 = 1, c_2 = 1 \). The boundary conditions are chosen so that they agree with the data at the inner and outer radius as shown in Figure 3-10.

For the prior distribution we apply the same separable Gaussian process as in the simulation study (3.4.1). We set the spatial length scale to \( \rho_x = b - a = 0.095 \), as this is consistent with previous studies on the corresponding 1-dimensional stationary heat transfer problem described in [151], and the temporal length scale is \( \rho_t = 400 \). The inferred Biot number at various time points is shown in Figure 3-11.

As expected our results show that there is very little change in \( Bi(t, x) \) at the time points prior to the heater being turned off at 500 seconds, and after this we observe the transient behaviour of \( Bi(t, x) \) as the disc cools. To validate the fit of the inferred

Figure 3-11: The inferred Biot number and 95% credible intervals at six time points.

As expected our results show that there is very little change in \( Bi(t, x) \) at the time points prior to the heater being turned off at 500 seconds, and after this we observe the transient behaviour of \( Bi(t, x) \) as the disc cools. To validate the fit of the inferred
$Bi(t, x)$, we substitute it into the governing PDE and solve using a high resolution CN scheme to obtain the predicted temperature. Figure 3-12 visualises these predicted temperatures and overlays the measured temperatures at various time points to demonstrate a good agreement between the data and PDE solution.

![Figure 3-12: Experimental data at various time points overlaid on the PDE solution using the inferred $Bi(t, x)$. The subplots below each panel are the residuals for the corresponding plot.](image)

A detailed physical interpretation of these results is outside the scope of this paper. Heuristically speaking, at $t = 0$ the large positive $Bi(t, x)$ at the outer radius ($x = 0.228$) represents the transfer of heat from the disc to the air at the outer radius where the disc is heated. This heated air then circulates and heats the disc at the cooler inner radius, thus producing the negative $Bi(t, x)$ that we observe in this region. When the heater is turned off at $t = 500$ the disc cools and heat transfer slows, resulting in the decay in $Bi(t, x)$ shown. This demonstrates that the results of our analysis shown in Figure 3-11 are physically interpretable, and as shown in Figure 3-12 this result achieves...
a very good agreement with the experimental data when used to solve the PDE. In future we will apply this methodology in a range of more complex experimental settings, such as experiments with variable frequencies that mimic the engine activity at various stages of a flight.

3.5 Closing remarks

In this work we have developed a fully Bayesian approach to solving the PDE inverse problem for the unknown spatio-temporal Biot number. Our approach augments the general deep surrogate method for parametric PDEs by implementing a novel training scheme based on solving the PDE only over the approximated posterior distribution. To achieve this, we have described how to apply our deep learning approach to quickly obtain a Laplace estimate. By comparing this to the true posterior we have demonstrated that the Laplace approximation gives reasonable estimates of the Biot number, though it underestimates the overall uncertainty of this estimate. To sample from the posterior in an accurate and efficient manner, we applied a novel deep surrogate-based delayed-acceptance HMC scheme. This scheme utilises the fast evaluation and differentiation of the deep learning surrogate to make decorrelated proposals based on HMC trajectories and applies delayed-acceptance acceptance criteria to mathematically guarantee the accuracy of our posterior with respect to a Crank–Nicolson solver.

Since Biot number calculations are involved in the design of turbo-machinery, it is important that the results of our inferences have quantifiable accuracy. The delayed-acceptance step ensures sampling accuracy is consistent with a Crank–Nicolson solver, and our results demonstrate that the delayed-acceptance HMC scheme efficiently achieves a large effective sample size, thus ensuring a low statistical error. Our scheme achieves this accuracy while maintaining a lower cost per effective sample than Metropolis–Hastings schemes based on random walk or Langevin proposals, even if these schemes sacrifice the sampling accuracy of the delayed-acceptance step and rely solely on rapid surrogate evaluations. We also showed that the higher accuracy of our adaptively trained surrogate combined with HMC resulted in accurate posterior samples at a fraction of the cost. Our methodology is generalisable to other PDEs and applications, therefore in settings where approximate accuracy is sufficient the potential exists to apply this approach for substantial computational gains.
Appendix

MAP estimation using local deep surrogates

To train the Laplace surrogate we must first compute the MAP estimate for the posterior in (3.3.7), that is we are interested in finding

$$\left(\alpha^*, \sigma^*\right) = \text{argmax}\ \left\{\log p(\alpha, \sigma | \hat{t}, \hat{x}, \hat{z})\right\}. \quad (3.5.1)$$

Starting with an initial guess \((\alpha_0, \sigma_0)\), we solve this optimisation problem in an approach that is similar in concept to trust region optimisation. More specifically, we train a local surrogate \(\hat{u}(t, x, \alpha)\) at \(\alpha_0\) by minimising the loss function (3.3.10) with \(\pi_\alpha = \pi_{\alpha_0}\), where \(\pi_{\alpha_0}\) is a probability measure with mass focused locally around \(\alpha_0\).

When the local surrogate is accurate, as judged by achieving a sufficiently low loss function value, we substitute this into the log-posterior (3.5.1) and take a gradient ascent step with respect to \((\alpha_0, \sigma_0)\) to maximise the resulting expression. This process repeats in an alternating manner, whereby at iteration \(n\) the network is trained over \(\pi^{\alpha_n}\) to ensure local accuracy at \(\alpha_n\) and another gradient ascent step is performed to maximise the posterior. Upon convergence the MAP estimate is the final value of \(\alpha_n\), and the Hessian matrix used in the Laplace approximation covariance is constructed by automatic differentiation of the log-posterior at \(\alpha_n\).

In our implementation we choose \(\pi^{\alpha_n} \sim \text{MVN}(\alpha_n, \Lambda^n)\) for the local measure, where \(\Lambda\) is a diagonal matrix. For the variance of this distribution we choose \(\Lambda^n_{i,i} = \lambda_n/2^k\), where \(k\) is the degree of the polynomial term corresponding to the \(i^{th}\) position. The decay in the variance of \(\pi^{\alpha_n}\) is manually set, starting with \(\lambda_0 = 20\) and ending with \(\lambda_{end} = 0.5\). This local surrogate is significantly cheaper to compute than the general surrogate, taking only a few seconds to reduce the loss function to below the level achieved by training a general surrogate for 15 hours, and ultimately achieving a significantly higher accuracy in a fraction of the time.

Hamiltonian Monte Carlo

Hamiltonian Monte Carlo \cite{14} is an MCMC approach which uses the gradients of the target distribution to generate efficient transitions. A benefit to using HMC proposals over simpler proposals methods such as those based on Gaussian symmetric random
walks is that HMC significantly reduces the correlation between successive sampled states while maintaining a high acceptance rate, meaning that far fewer MCMC iterations are needed to approximate the target probability distribution to a given tolerance level. This sampling efficiency is achieved by simulating random trajectories of a carefully designed Hamiltonian system to propose successive states, whilst maintaining detailed balance with respect to the target distribution. Another benefit of HMC is short burn-in periods, since these Hamiltonian proposals target regions of higher probability. [13, 110] have noted that the HMC approach is particularly beneficial in Bayesian problems.

Consider the target distribution \( \psi(\theta) \), where \( \theta \in \mathbb{R}^{M+1} \), and suppose we want to simulate a chain of samples \( \theta_0, \theta_1, \theta_2, \ldots \) from this distribution. In HMC, we expand the target density into a \( 2(M+1) \)-dimensional distribution \( \psi(\theta, p) = \psi(p|\theta)\psi(\theta) \) over a phase space described in terms of a position variable in the parameter space \( \theta \in \Theta \) and an auxiliary momentum variable \( p \in \mathbb{R}^{M+1} \). We then use Hamiltonian dynamics to sample from this density, noting that the target distribution is the marginal \( \psi(\theta) = \int_{\mathbb{R}^{M+1}} \psi(\theta, p)dp \) which is estimated by considering only the positional coordinates of the sample. Hamilton’s equations are given by the coupled ordinary differential equations

\[
\frac{d\theta_i}{dt} = \frac{\partial H}{\partial p_i}, \\
\frac{dp_i}{dt} = -\frac{\partial H}{\partial \theta_i}.
\]  (3.5.2)

Here, \( \theta_i \) and \( p_i \) represent the \( i \)th component of the position and momentum vectors, and \( H \) is the Hamiltonian. For more details on Hamiltonian systems, see, for example [91]. In HMC, we define the Hamiltonian by decomposing the phase space distribution

\[
H(\theta, p) = -\log \psi(p, \theta) \\
= -\log \psi(\theta) - \log \psi(p|\theta) \\
= U(\theta) + K(p|\theta).
\]  (3.5.3)

Here \( U(\theta) = -\log \psi(\theta) \) represents the potential energy and \( K(p|\theta) = -\log \psi(p|\theta) \) represents the kinetic energy of the system. For our application we define the conditional momentum distribution as \( \psi(p|\theta) \sim \text{MVN}(0, M) \) for some symmetric positive definite matrix \( M \), then the Hamiltonian takes the form

\[
H(\theta, p) = -\ln(\psi(\theta)) + \frac{1}{2}p^TM^{-1}p.
\]  (3.5.4)
At the beginning of each Monte Carlo iteration the initial momentum is sampled from $\psi(p|\theta)$. Using this initial momentum the Hamiltonian dynamics are simulated starting at the current state of the position variable of the Markov Chain. These simulations then follow contours of constant energy in phase space over a predefined time horizon. The matrix $M$ (known as the mass matrix due to its physical interpretation) determines the initial speed and directions of the trajectories, and therefore has a large impact on the proposals. We specify this matrix adaptively within a warm-up phase prior to sampling as outlined in Section 3.4.1.

In order to use the Hamiltonian dynamics within the proposal distribution for a Markov Chain we require a mechanism for simulating these trajectories. This is achieved using the leapfrog integrator, which conserves the energy in the system, leading to accurate simulations relative to non-symplectic solvers. Given a step size $\delta$, a single leapfrog iteration of the system (3.5.2) is given by the steps:

\begin{align*}
    p_i(t + \delta/2) &= p_i(t) - (\delta/2) \frac{\partial U}{\partial \theta_i(t)} , \\
    \theta_i(t + \delta) &= \theta_i(t) + \delta \frac{\partial K}{\partial p_i(t + \delta/2)} , \\
    p_i(t + \delta) &= p_i(t + \delta/2) - (\delta/2) \frac{\partial U}{\partial \theta_i(t + \delta)} .
\end{align*}

(3.5.5)

Multiple of these steps can be carried out in order to integrate over longer time horizons, and different step sizes can be used depending on the accuracy required. The state of the position and momentum variables at the end of the simulation are used as proposed state variables $\theta_{\text{prop}}$ and $p_{\text{prop}}$. Finally the Metropolis-Hastings acceptance criteria is employed (on the phase space distribution) to ensure detailed balance is maintained. The acceptance probability in this case is

\[ A_H(\theta_{\text{prop}}, p_{\text{prop}}, \theta, p) = \min \left\{ 1, \frac{\psi(p_{\text{prop}}|\theta_{\text{prop}})\psi(\theta_{\text{prop}})}{\psi(p|\theta)\psi(\theta)} \right\} . \]  

(3.5.6)

Combining these steps, the HMC algorithm for drawing $N_{\text{samples}}$ samples from a target distribution can be written as:
Algorithm 7: Hamiltonian Monte Carlo

Set initial position state $\theta_0 \in \Theta$.

for $i = 0, 1, \ldots, N_{\text{samples}}$ do

    Sample momentum $p_i \sim \text{MVN}(0, M)$.
    Simulate Hamiltonian with initial state $(\theta_i, p_i)$ using leapfrog algorithm to obtain proposed states $(\theta_{\text{prop}}, p_{\text{prop}})$.

    Sample $u \sim U(0, 1)$;
    if $u \leq A_H(\theta_{\text{prop}}, p_{\text{prop}}, \theta, p)$; then
    | set $\theta_{i+1} := \theta_{\text{prop}}$ (accept);
    else
    | set $\theta_{i+1} := \theta_i$ (reject);
    end

end

return $\theta = (\theta_0, \theta_1, \ldots, \theta_{N_{\text{samples}}})$
Subchapter 3A

Use of Bayesian statistics to calculate transient heat fluxes on compressor discs

Abstract

Future gas turbine engines require improved understanding of the heat transfer between compressor discs and air in compressor cavities under transient operating conditions. Calculation of transient heat fluxes from temperature measurements on compressor discs is a typical ill-posed inverse problem where small uncertainties of measurements can lead to large uncertainties of the calculated fluxes. This paper develops a Bayesian model for the heat flux to reduce the adverse nature of the problem by using a Gaussian prior distribution with Matérn covariance. To efficiently find the maximum a posterior (MAP), a neural network was used to solve the heat equation for compressor discs for any choice of parameters, allowing fast evaluation of the solution to the forward model for any heat flux of interest. The power of the Bayesian model is first demonstrated using numerically-simulated data. Subsequently, the model is used to calculate heat fluxes from measurements of transient temperature collected from the Compressor Cavity Rig at the University of Bath. During these transient tests, the periphery of the rotating compressor disc was initially heated to a steady-state condition and then cooled rapidly by the ambient air. The fluxes for four transient cycles were calculated, with the operating range of $7.0 \times 10^5 < \text{Re}_\phi < 2.8 \times 10^6$, $0.0 < \beta \Delta T < 0.15$ and $0.13 < \chi$, where $\text{Re}_\phi$, $\beta \Delta T$ and $\chi$ are the rotational Reynolds number, the buoyancy...
parameter and the compressibility parameter, respectively. The results show that, for all four cases, the flow and the heat transfer in the closed cavity were initially dominated by buoyancy effects, with heat transferred from the disc to the cavity air in the outer region and reversed in the inner region. The initial heat fluxes at \( \text{Re}_\phi = 2.1 \times 10^6 \) were higher than those at \( \text{Re}_\phi = 2.8 \times 10^6 \) owing to a compressibility effect. During the cooling transient, for cases with \( \text{Re}_\phi \leq 1.4 \times 10^6 \), the magnitudes of the heat fluxes gradually decreased and eventually reached virtually zero. This indicated that the flow was first governed by the buoyancy effects and then became stratified. At \( \text{Re}_\phi = 2.8 \times 10^6 \), where the rotational speed was at its maximum, buoyancy-induced flow dominated the entirety of the transient process due to significant frictional heating at the periphery of the rig. The calculated fluxes present evidence for future theoretical and computational modelling of transient disc heat transfer, and the Bayesian model provides guidance for transient temperature data analysis.
3A.1 Introduction

Efficiency and reliability of gas turbine engines are highly dependent on blade tip clearances in compressors. For the next generation of aero-engines to meet environmental targets, overall pressure ratios will be increased from 50:1 in state-of-art engines to 70:1. The added compression will decrease the height of blades in high-pressure compressors and will require much smaller tip clearances to maintain high efficiency and reliability. Figure 3A-1 shows the structure of a typical aero-engine high pressure compressor. The tip clearance is governed by the relative expansion of the casing and the rotor. Under transient operating conditions, the thermal growth of the rotor responds slower than that of the casing owing to its larger thermal inertia. This leads to a minimum tip clearance during the restart of a hot engine, which constrains the reduction of clearances and hence an optimisation of engine efficiency. To improve clearance control technologies for the next generation of aircraft engines, it is vital to accurately predict the transient thermal growth of the compressor rotor. The thermal growth is determined by the distribution of the disc temperature, which is governed by the transient heat transfer on disc surfaces.

Figure 3A-1: A cross-section through a high pressure aero-engine compressor. (Owen, J. M., Tang, H., and Lock, G. D., Aerospace, 5(1), 32, 2018; licensed under a Creative Commons Attribution (CC BY) license.)

Accurate and fast predictive methods are important for engine clearance decisions using combined thermo-mechanical models across the full operating flight envelope. Under cruise conditions, the shroud is hotter than the throughflow (see Figure 3A-1) and buoyancy-induced flow can occur in the open or closed rotating cavities formed by the discs and shroud. However, at other points in the flight cycle the shroud can become
colder than the cob, and this suppresses buoyancy effects; here stratified flow may form
with heat transfer governed by conduction rather than convection. This large reduction
of heat transfer, which could occur during an engine transient, would have a significant
effect on the temperature of the compressor discs.

Compressor cavities rigs have been built to simulate disc heat transfer under different
operating conditions, with disc temperatures measured using embedded thermocouples.
For example, the Universities of Sussex [2, 7, 105] and Bath [75, 76, 100] and TU
Dresden [40, 62, 63] have all used K-type thermocouples in titanium discs to measure
radial distributions of temperature. Given an appropriate model of heat transfer, heat
fluxes can be calculated from these measurements. This is a typical ill-posed inverse
heat transfer problem where small uncertainties in temperature measurements lead to
large uncertainties in heat fluxes.

Conventionally, temperatures are fitted and then differentiated to compute the fluxes.
In [2, 62, 63, 105, 120], polynomial or exponential functions were used to fit steady-
state temperature measurements and then differentiated to evaluate the fluxes. This
process can introduce extra uncertainty from the fitting models and result in unrealistic
heat flux distributions [151]. Bayesian statistics has been proven to be effective in
reducing uncertainties of inverse heat transfer results, given reliable prior information
of the unknown parameters [82, 114, 157]. Tang et al. [150, 151] and Deveney et al.
[38] used Bayesian models together with prior density functions to calculate steady-
state heat fluxes from steady temperature measurements, and reliable fluxes as well
as their confidence intervals were provided. In this paper, a Bayesian model is built
for the transient heat fluxes and is applied to transient temperature measurements
collected from the Compressor Cavity Rig at the University of Bath. Following Deveney
et al. [39], a deep-learning surrogate for the forward model is computed to enable
rapid evaluation of the forward model for any parameters of interest. This gives a
representation of the parametric solution as a deep neural network, which is easily
differentiated, providing gradients for computation of the MAP estimate by gradient-
based descent methods. The calculated fluxes provide insight related to the transient
flow and heat transfer in compressor cavities, and they will be used to validate and
inform future theoretical and computational models. The method developed here will
provide guidelines for data analysis of future experiments on transient disc heat transfer
and have practical significance to the designers of aero-engines.

Section 3A.2 reviews research on flow and heat transfer in compressor cavities. Section
3A.3 introduces the Compressor Cavity Rig at the University of Bath, including the
disc geometry and instrumentation. Section 3A.4 presents a mathematical model of the
disc heat transfer and Section 3A.5 explains the Bayesian model. In Section 3A.6, the
Bayesian model is applied to simulated temperature data and is validated through the
comparison between the calculated and the true heat flux distributions. The method
is then applied to transient temperature measurements from the Bath rig, and the
transient flow and heat transfer in compressor cavities illustrated by the inverse solution
are discussed. Section 3A.7 states the conclusions and suggestions for future work.

3A.2 Buoyancy-induced and stratified flow in compressor
cavities

Alexious [2] and Atkins [6] presented measurements of transient disc temperature mea-
surements using the multi-cavity compressor rigs at the University of Sussex. The
transient behaviour of temperature was discussed, but the heat fluxes were not calcu-
lated. Though research on transient heat transfer on compressor discs is limited, there
are a broad range of experimental, theoretical, and computational studies on steady-
state heat transfer in different compressor cavities. As shown in Figure 3A-1, there
are axial gaps between adjacent disc cobs and these are referred to as open cavities.
In some compressors, the axial gaps do not exist, leaving closed spaces between the
shroud, the two adjacent disc and a hub. These configurations, often found in indus-
trial compressors, are referred to as closed cavities. This section focuses on a review
of flow and heat transfer in closed cavities (one of which is used in this research). A
broad review of research on buoyancy-induced flow in compressor cavities was given by
Owen and Long [115].

The well-known non-dimensional parameter to determine the steady-state flow and
heat transfer in closed compressor cavities is the Grashof number $Gr$, which can be
defined as

$$Gr = Re_\phi^2 \beta \Delta T,$$

where $Re_\phi$ and $\beta \Delta T$ are the rotational Reynolds number and the buoyancy parameter,
respectively. They are defined as

$$Re_\phi = \frac{\Omega r_s^2}{\nu},$$

$$\beta \Delta T = \frac{T_s - T_h}{(T_s + T_h)/2},$$

$$Gr = Re_\phi^2 \beta \Delta T,$$

$$Re_\phi = \frac{\Omega r_s^2}{\nu},$$

$$\beta \Delta T = \frac{T_s - T_h}{(T_s + T_h)/2},$$

133
where \( r_s \) is the outer radius of the cavity and \( T_s \) and \( T_h \) are the shroud and hub temperatures, respectively.

Bohn et al. [19] measured the steady-state heat transfer in three closed compressor cavities (denoted as cavities A, B, and C), with \( Gr < 4.0 \times 10^{12} \) and \( Re_\phi < 9.0 \times 10^6 \). In these cavities, the discs are thermally insulated and the shroud were heated using a copper ring. The flow structure and heat transfer in cavity B was investigated by Pitz et al. [119] and Gao et al. [50] using high fidelity Computational Fluid Dynamics (CFD) models. The simulated results showed four pairs of cyclonic and anti-cyclonic vortices which were induced by the buoyancy effects. It was shown that circumstantially and time averaged cavity air temperatures were almost constant with radius. Tang and Owen [151] used reduced-order methods to reproduce the heat transfer and general good agreement between experiment and theory was reached. In the model, they derived an equation to calculate the radial distribution of the core temperature. Note that in this paper the core temperature denotes the circumferentially-averaged air temperature (outside of boundary layers) in the cavity. It was shown that the core temperature increases as the radius increases and this temperature rise is enhanced when \( Re_\phi \) is increased. The effect was referred to as the compressibility effect.

At the University of Bath, a closed cavity with non-insulated discs was tested under steady-state conditions [75], with \( Gr < 10^{12} \) and \( Re_\phi < 2.7 \times 10^6 \). A steady-state Bayesian model was used to calculate the disc heat fluxes. Figure 3A-2 shows the radial distributions of measured disc temperatures and derived heat fluxes. Heat was transferred to the cavity air at high radial locations and into the discs at low radial locations. Importantly, the radial location where the disc heat flux was zero was consistent for all cases; this radius indicates where the core and disc temperatures are equal. It was found that this radius was consistent at \( r/b = 0.7 \) under different steady-state operating conditions. A plume model was derived to predict the steady-state disc temperatures and heat fluxes [149]. As shown in Figure 3A-3, the model assumed heat was transferred from the hot shroud to the cold hub via radial plumes between cyclonic and anti-cyclonic vortices. The model also assumed conductive laminar Ekman layers on disc surfaces, which can be used to determine the heat transfer coefficient \( h \) for the disc heat transfer. The heat transfer coefficient is defined as

\[
h = \frac{q}{T - T_{\text{core}}},
\]

where \( q \) is the disc heat flux, \( T \) the disc temperature, and \( T_{\text{core}} \) the core air temperature. For conductive Ekman layers, \( h \) is mainly determined by \( Re_\phi \). Hence for a fixed \( Re_\phi \),
the flux is solely dependent on the temperature difference between the disc and the core. Consequently, the prediction of the radial distribution of the core temperature is critical to the agreement between theoretical and experimental heat fluxes. As discussed in [149], the core temperature is heavily influenced by the compressibility effect, which is governed by the compressibility parameter $\chi$:

$$\chi = \frac{\Omega^2 r_s^2}{\gamma R(T_s - T_h)}.$$  \hspace{1cm} (3A.2.5)

It was predicted that as $\chi$ increases the radial rise of the non-dimensional core temperature increases, which decreases the disc heat fluxes. There is a critical $\chi$ where the core temperature becomes higher than the shroud temperature, the buoyancy-induced flow structures disappear and the flow becomes stratified. The model was validated by experimental data presented in Lock et al. [98]. Both the theory and experiment concluded that the steady-state disc temperatures and fluxes are governed by three non-dimensional parameters, $Re_\phi$, $\beta \Delta T$ and $\chi$. The model predicted that for a cavity with $r_s/r_h = 0.5$ and adiabatic discs the critical $\chi$ is 6.7, and the experiments showed that at $\chi = 6.6$ there were no buoyancy-induced structures, and the disc heat flux became virtually zero.

Figure 3A-2: Steady-state disc temperatures and heat fluxes at $Re_\phi = 7.0 \times 10^5$ (data from [75]). (a) Non-dimensional disc temperatures. (b) Heat fluxes.
3A.3 Bath compressor cavity rig

Figure 3A-4 shows a diagram of the Compressor Cavity Rig in the Turbomachinery Research Centre at the University of Bath. This rig was designed to investigate heat transfer in typical high pressure compressor cavities. The data collected from the rig are used to inform and validate theoretical models that can be used by aero-engine designers. The test section consists of three rotating cavities with the central one instrumented. The test section is driven using a supercritical design by a 30 kW motor and a maximum rotational speed of 8000 rpm. The shroud of the rotor is heated through a 12 kW radiant heater. A 7.5 kW extraction unit provides cooling air flow through the compressor discs for a maximum flow rate of 0.15 kg/s. More details about the rig design and capabilities are given in [100].

Here the instrumented rotating cavity is closed using aluminium attachments near the cobs. A detailed diagram is shown in Figure 3A-5. The back surfaces of the instru-
mented discs are insulated using a low-conductivity foam (Rohacell with a thermal conductivity of 0.03 W/mK). The maximum shroud temperature was limited to 100°C considering the safe temperature range of the foam material. Both upstream and downstream discs are made of titanium alloy (Ti-5AL-4V, grade 5, annealed) and heated symmetrically. The thermal properties of the disc material at T = 293K are listed in Table 3A.1. K-type thermocouples are embedded in the disc to measure the transient disc temperatures at a frequency of 10 Hz. The upstream and downstream discs feature a distribution of 27 thermocouples on diaphragms and their locations are shown in Figure 3A-5. As the discs are symmetrically heated, there is no significant temperature difference between the two surfaces. In this paper, the measurements on both discs are used. The inner and outer radii of the closed cavity are 109 mm and 240 mm, and the inner and outer radii, and thickness of the diaphragm section are $a = 0.124$ m, $b = 0.236$ m and $t_d = 0.008$ m, respectively. To avoid the effect of fillet radius on the disc heat fluxes and temperatures, only the temperature measurements on the diaphragm section are used in this study.

Figure 3A-5: Instrumented closed cavity showing the locations of thermocouples.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thermal Conductivity</td>
<td>6.6 W/(m K)</td>
</tr>
<tr>
<td>Specific Heat</td>
<td>542 J/(kg K)</td>
</tr>
<tr>
<td>Density</td>
<td>$4.42 \times 10^3$ kg/m$^3$</td>
</tr>
<tr>
<td>Thermal Diffusivity</td>
<td>$2.76 \times 10^{-6}$ m$^2$/s</td>
</tr>
</tbody>
</table>

Table 3A.1: Thermal properties of Ti-6Al-4V at 293K, interpolated from data in [158].
3A.4 Modelling of transient heat transfer in compressor discs

As the thickness of the disc is relatively small, the heat transfer in the disc can be modelled using the transient annular fin equation, which can be written as

\[
\frac{1}{\alpha} \frac{\partial T(t,r)}{\partial t} = \frac{\partial^2 T(t,r)}{\partial r^2} + \frac{1}{r} \frac{\partial T(t,r)}{\partial r} - \frac{2}{k t_d} q(t,r),
\]

(3A.4.1)

where \( T(t,r) \) and \( q(t,r) \) are the transient disc temperature and heat flux, respectively. Other symbols are defined in the nomenclature. Note that in (3A.4.1), \( t_d \) should be twice the actual disc thickness due to the insulated disc back surfaces. To solve Equation (3A.4.1), initial conditions at \( t = 0 \) and boundary conditions at \( r = a \) and \( r = b \) are required. These conditions can be written as

\[
T(0,r) = T_0(r),
\]

(3A.4.2)

\[
T(t,a) = T_a(t),
\]

(3A.4.3)

\[
T(t,b) = T_b(t).
\]

(3A.4.4)

Equations (3A.4.1-3A.4.4) can be written in a non-dimensional form as

\[
\frac{\partial \theta(Fo,x)}{\partial Fo} = \frac{\partial^2 \theta(Fo,x)}{\partial x^2} + \frac{1}{x} \frac{\partial \theta(Fo,x)}{\partial x} - \phi(Fo,x),
\]

(3A.4.5)

\[
\theta(0,x) = \theta_0(x),
\]

(3A.4.6)

\[
\theta(Fo,x_a) = \theta_a(Fo),
\]

(3A.4.7)

\[
\theta(Fo,x_b) = \theta_b(Fo).
\]

(3A.4.8)

where the non-dimensional time, radius, temperature and heat flux are defined as

\[
x = \frac{r}{b},
\]

(3A.4.9)

\[
Fo = \frac{\alpha t}{b^2},
\]

(3A.4.10)

\[
\theta = \frac{T - T_{ref}}{T_{b,0} - T_{ref}},
\]

(3A.4.11)

\[
\phi = \frac{2b^2 q}{k t_d(T_{b,0} - T_{ref})},
\]

(3A.4.12)
where the subscript ‘ref’ denotes values in the cooling flow.

When the transient heat flux $\phi(Fo,x)$ is given, Equation (3A.4.5) can be solved to calculate the transient disc temperature $\theta(Fo,x)$. This is referred to as the forward problem. However, in this study, $\phi(Fo,x)$ is unknown and needs to be inferred from temperature measurements. To do this, $\phi$ is iteratively updated, and Equation (3A.4.5) is solved for each iteration until good agreement between the calculated and measured temperatures is reached. Temperature measurements are taken at a limited number of radial locations with fixed data acquisition frequency and experimental error. The $i^{th}$ temperature measurement is denoted by $\Theta(Fo_i,x_i)$, for $1 \leq i \leq N_r \times N_t$ where $N_r$ is the number of thermocouples on the two disc diaphragm sections and $N_t$ is the number of temporal points. Determining $\phi(Fo,x)$ from $\Theta(Fo_i,x_i)$ is an ill-posed inverse problem where small uncertainties in $\Theta(Fo_i,x_i)$ can cause large uncertainties in the heat fluxes $\phi(Fo,x)$. Therefore, regularisation is required to reduce oscillatory behaviour and increase the accuracy of $\phi(Fo,x)$. This can be achieved through a Gaussian prior density function with a covariance that encourages smoothly evolving behaviour in the Bayesian inference which is discussed in detail in Section 3A.5.

Solving this inverse problem can be time consuming hence it is important to carefully define the variable space for $\phi(Fo,x)$. A 2-dimensional Chebyshev polynomial function is used to represent the unknown distribution of $\phi(Fo,x)$. The maximum combined orders of the polynomial is set to be $N_p = 10$, which results in $N_c = 66$ coefficients for the polynomial function. Therefore,

$$\phi(Fo,x) = \sum_{j=1}^{N_r} c_j f_j(Fo,x) = \sum_{n=0}^{N_p} \sum_{m=0}^{N_p-n} c(2N_p+3-n)n/2+m+1 G_n(Fo) G_m(x), \quad (3A.4.13)$$

where $G_k$ is the $k^{th}$ degree shifted Chebyshev polynomials of the first kind and $f_j(2N_p+3-n)n/2+m+1(Fo,x) = G_n(Fo) G_m(x)$, and the $c_j$ are the Chebyshev coefficients which determine the shape of $\phi$.

To improve the computing efficiency of solving Equation (3A.4.5), the deep learning methodology of [39] was applied. In this approach a six-layer neural network was trained to approximate the solution to the Equation (3A.4.5) over a range of heat fluxes. Specific details of how the neural network is constructed and trained to solve the Partial Differential Equation (PDE) are given in [39]. By applying automatic differentiation to this neural network approximation, the time required to compute the gradients of the PDE solution with respect to the flux is reduced by 99.6%. This is in comparison to a traditional finite difference approximation of these gradients, which would require the
equation to be solved \( N_c + 1 \) times. The flux \( \phi(Fo, x) \) was then estimated by applying these gradients to optimise the Bayesian posterior distribution as described in Section 3A.5. After training, the maximum relative temperature difference between the neural network and a high resolution Crank-Nicolson finite difference scheme for the estimated \( \phi(Fo, x) \) is less than 0.3%.

### 3A.5 Bayesian approach

The transient heat flux is inferred from temperature measurements using a Bayesian statistical method. In the Bayesian approach, the unknown heat flux was estimated by combining a prior distribution over the unknown function with experimental data using a forward model of the physical process (Equation (3A.4.5)). This approach provides a principled quantification of uncertainties in our estimated flux in the form of a posterior probability distribution over the transient heat flux. Additionally, the Bayesian approach incorporates prior understanding of the flux into the inference, allowing physically sensible constraints to be enforced, such as a smooth evolution of the heat flux over space and time.

Bayesian inference can be stated mathematically for the inverse problem in the following form as:

\[
p(\phi|\Theta) = \frac{p(\Theta|\phi)p(\phi)}{\int p(\Theta|\phi)p(\phi)d\phi},
\]

where \( p(\phi) \) is the prior probability density function of the non-dimensional heat flux \( \phi \), \( p(\Theta|\phi) \) is the conditional probability density function (or the likelihood function) of the temperature measurements \( \Theta \) given the heat flux \( \phi(Fo, x) \), and \( p(\phi|\Theta) \) is the posterior probability density function of the flux conditional on the measurements \( \Theta \). The integral \( \int p(\Theta|\phi)p(\phi)d\phi \) can be treated as a constant.

As discussed in Section 3A.4, \( \phi(Fo, x) \) is represented using a 2D Chebyshev polynomial with shape determined by the coefficients \( c \) in Equation (3A.4.13); hence Equation (3A.5.1) can be written in terms of these coefficients

\[
p(c|\Theta) = \frac{p(\Theta|c)P(c)}{\int p(\Theta|c)p(c)dc}.
\]

140
### 3A.5.1 Likelihood function

The measured temperature data is assumed to be related to the temperature calculated from the forward model according to the statistical model

$$\Theta(t_i, x_i) = \theta(t_i, x_i) + \epsilon_i, \text{ or } \Theta_i = \theta_i + \epsilon_i.$$  \hfill (3A.5.3)

Here $\theta$ is the solution to the forward problem (Equation (3A.4.5)), and the $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ are independent normal random variables representing sensor measurement error. According to this model the likelihood function is given by

$$p(\Theta|c) = \frac{1}{(2\pi \sigma^2)^{N_t \times N_r/2}} \exp \left( - \sum_{i=1}^{N_t \times N_r} \frac{(\Theta_i - \theta_i(c))^2}{2\sigma^2} \right).$$  \hfill (3A.5.4)

This represents the probability of observing the data $\Theta_i$ for the given heat flux distribution $\phi(Fo,x)$ induced by the coefficients $c$.

### 3A.5.2 Prior distribution

The prior distribution is an integral part of the inference as it has a large influence on the behaviour of the inferred heat flux. In this setting it is desirable to infer a smooth heat flux as this behaviour is more physically expected. To achieve this the popular class of mean zero Gaussian process priors in space and time $\phi \sim \mathcal{GP}(0,K)$ was selected, where $K$ is a separable covariance kernel of the form

$$K([Fo_1, x_1], [Fo_2, x_2]) = \sigma^2 K_{Fo}(Fo_1, Fo_2) K_{x}(x_1, x_2).$$  \hfill (3A.5.5)

The spatial correlation structure is defined by the twice differentiable Matérn covariance kernel

$$K_{x}(x_1, x_2) = \left( 1 + \frac{|x_1 - x_2|}{\rho_x} + \frac{|x_1 - x_2|^2}{3\rho_x^2} \right) \exp \left( -\frac{|x_1 - x_2|}{\rho_x} \right).$$  \hfill (3A.5.6)

The temporal correlation structure is given by the squared exponential kernel

$$K_{Fo}(Fo_1, Fo_2) = \exp \left( -\frac{|Fo_1 - Fo_2|^2}{2\rho_{Fo}^2} \right).$$  \hfill (3A.5.7)

The length-scales $\rho_{Fo} = 0.053$ and $\rho_x = x_b - x_a$ represent the temporal and spatial length-scales, and $\sigma = 3.75$ is the standard deviation of the amplitude of $\phi(Fo,x)$.
according to the prior distribution. This Gaussian process is approximated to a high
degree of accuracy as shown in \[39\] using the Chebyshev polynomial by choosing a zero
mean multivariate normal distribution over the coefficients \(c\), with covariance matrix
\(E\) such that

\[
K([F_{o1}, x_1], [F_{o2}, x_2]) \approx \sum_{m=1}^{N_c} f_m(F_{o1}, x_1) \sum_{n=1}^{N_c} E_{mn} f_n(F_{o2}, x_2).
\] (3A.5.8)

The resulting prior density function for the coefficients \(c\) is therefore

\[
p(c) = \frac{1}{(2\pi)^{N_c/2}|E|} \exp \left( -\frac{c^T E^{-1} c}{2} \right).
\] (3A.5.9)

### 3A.5.3 Posterior distribution

The output of the Bayesian method is the posterior distribution. This is the con-
tditional distribution of \(\phi(F_o, x)\) after taking into account the temperature measurements.
Using the prior distribution (3A.5.9) and likelihood function (3A.5.4), the posterior dis-
tribution \(p(c|\Theta)\) is given by Equation (3A.5.1). It follows

\[
p(c|\Theta) \propto p(\Theta|c) p(c) = \frac{1}{(2\pi\sigma^2)^{N_t \times N_r/2}} \exp \left( -\frac{N_t \times N_r}{2\sigma^2} \sum_{i=1}^{N_t \times N_r} (\Theta_i - \theta_i)^2 \right) \times
\frac{1}{(2\pi)^{N_c/2}|E|} \exp \left( -\frac{c^T E^{-1} c}{2} \right).
\] (3A.5.10)

### 3A.5.4 Solution technique

To approximate the posterior distribution gradient descent was applied using the neural
network to find the flux that has the highest posterior probability; this is known as the
Maximum a Posteriori (MAP) estimate \[81\]. As noted above, automatic differentia-
tion of the neural network allows these gradients to be computed with minimal cost, whereas
a more traditional finite differences approximation to the gradient would require \(N_c + 1\)
PDE solves. MAP estimation is equivalent to minimising the negative logarithm of the
posterior:

\[ L = N_t \times N_r \left( \frac{1}{2} \ln 2\pi + \ln \sigma \right) + \sum_{i=1}^{N_t \times N_r} \frac{(\Theta_i - \theta_i)^2}{2\sigma^2} + \frac{N_c}{2} \ln 2\pi + \ln |E| + \frac{1}{2} c^T E^{-1} c. \]  (3A.5.11)

Once the MAP estimate is obtained, the posterior distribution can then be explored using Markov Chain Monte Carlo methods (MCMC) resulting in a full probability distribution over the heat flux. As shown in [39], the MAP method can provide reliable estimation of the heat flux with a relatively low computational cost, while the full posterior density function gives an accurate representation of the overall uncertainty of this estimate.

3A.6 Calculation of transient heat fluxes on compressor discs

3A.6.1 Application of Bayesian method to simulated temperature data

To validate the Bayesian model, it is applied to simulated temperature measurements using a given heat flux distribution, which is referred to as the ‘true’ heat flux distribution and is denoted as \( \phi_{\text{true}} \). The true heat flux distribution is set as

\[ \phi_{\text{true}} = 4 \sin \left( \frac{x - (x_b + x_a)/2}{x_b - x_a} \pi \right) e^{-50F_o} + 1, \]  (3A.6.1)

and is plotted in Figure 3A-6.
Given initial and boundary conditions, the true disc temperature, \( \theta_{\text{true}}(F_{o_i}, x_i) \), can be calculated from solving Equation (3A.4.5). Here \( F_{o_i} \) and \( x_i \) represent temporal and radial locations of the simulated data. 101 uniformly distributed temporal locations between \( 0 < F_{o} < 0.16 \) and 21 uniformly distributed radial locations between \( 0.5 < x < 1 \) are used (\( N_t = 101 \) and \( N_r = 21 \)). The initial disc temperatures are assumed to be those with pure conduction within the disc. The temperature at the inner radius is fixed at 0, and at the outer radius is assumed to decrease exponentially. The assumed initial conditions at \( F_{o} = 0 \) and boundary conditions at \( x_i = 0.5 \) and \( x_i = 1 \) are plotted in Figure 3A-7 and listed below:

\[
\begin{align*}
\theta(0, x) &= \frac{\ln(x/x_a)}{\ln(1/x_a)}, \\
\theta(F_{o}, x_a) &= 0, \\
\theta(F_{o}, 1) &= e^{-11F_{o}}.
\end{align*}
\] (3A.6.2) (3A.6.3) (3A.6.4)

Figure 3A-7: Initial and boundary conditions for simulated data. (a) Boundary conditions at \( x = 1 \) and \( x_a = 0.5 \). (b) Initial condition at \( F_{o} = 0 \).

The true disc temperature \( \theta_{\text{true}}(F_{o_i}, x_i) \) from Equation (3A.4.5) can be either determined using conventional finite difference methods or trained neural networks. As discussed in Section 3A.4, the neural network method can be more efficient than and as accurate as the finite difference methods, for example, the Crank-Nicolson method. Independent random normal noise \( \epsilon_i \sim \mathcal{N}(0, 0.0036^2) \) is added to each \( \theta_{\text{true}}(F_{o_i}, x_i) \) to
generate simulated data. Hence

\[ \Theta(t_i, x_i) = \theta_{true}(t_i, x_i) + \epsilon_i \text{ or } \Theta_i = \theta_{true,i} + \epsilon_i, \]  

(3A.6.5)

The simulated temperature data at different radii and different temperature locations are shown in Figure 3A-8. Using the Bayesian method described in Section 3A.5, transient heat fluxes \( \phi(Fo, x) \) can be determined from \( \Theta_i \). The distributions of the calculated heat fluxes and their standard deviations are shown in Figure 3A-9 (a) and (b), respectively. The calculated disc temperatures using the fluxes are plotted in Figure 3A-9, showing good agreement with the simulated data. The calculated heat flux from the Bayesian model is then compared to \( \phi_{true} \) in Figure 3A-9(c). The difference between the calculated and true heat flux and the standard deviation increase near the boundaries, as a result of the temperatures being less sensitive to heat fluxes near the boundaries. The maximum standard deviation in Figure 3A-9 (b) is 0.55 and the maximum magnitude of the difference in Figure 3A-9 (b) is 0.85. It can be shown that the difference is less than 11% relative to the range of the true heat flux distribution and within the calculated 95% confidence internals, which demonstrates the accuracy of the Bayesian model (Note that the 95% confidence intervals are treated as twice the magnitude of the standard deviation.).

![Figure 3A-8](image)

Figure 3A-8: Simulated disc temperature data and calculated temperatures from the Bayesian model. Symbols: simulated data; solid lines: calculated temperatures from the Bayesian model. (a) Temperature data histories at different radial locations. (b) Radial data distributions at different temporal locations.
Figure 3A-9: Calculated heat fluxes ($\phi(Fo, x)$) from simulated data using the Bayesian model. (a) Calculated heat fluxes using the simulated data. (b) Standard deviation of the calculated heat fluxes. (c) Difference between the calculated and true heat flux distributions.
### 3A.6.2 Transient heat fluxes in a closed rotating cavity

In this section, the Bayesian model is applied to transient temperatures measured from the Compressor Cavity Rig at the University of Bath. The transient disc heat fluxes are determined from the temperature data using the Bayesian model. The distributions of the temperature measurements and the calculated heat fluxes from four transient tests are discussed.

Transient tests were conducted at four different rotational Reynolds numbers. Prior to the transient phase, the cavity was heated to a steady-state condition. The transient phase started at $Fo = 0$ when the heating to the shroud was shut off and ended at $Fo = 0.16$. Table 3A.2 lists the key initial nondimensional parameters for these cases. Though temperature measurements were recorded at a frequency of 10 Hz, due to the limit of the computer memory, a time step of $\Delta Fo \approx 0.0016$ (equivalent to $\Delta t = 30\,s$) was used. Temperatures at different radial locations on the two disc diaphragms were measured using 27 K-type thermocouples ($N_r = 27$).

<table>
<thead>
<tr>
<th>Case No.</th>
<th>$Re_\phi$</th>
<th>$\beta\Delta T_0$</th>
<th>$Gr_0$</th>
<th>$\chi_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$7.0 \times 10^5$</td>
<td>0.15</td>
<td>$8.4 \times 10^{10}$</td>
<td>0.14</td>
</tr>
<tr>
<td>B</td>
<td>$1.4 \times 10^6$</td>
<td>0.15</td>
<td>$3.3 \times 10^{11}$</td>
<td>0.55</td>
</tr>
<tr>
<td>C</td>
<td>$2.1 \times 10^6$</td>
<td>0.15</td>
<td>$7.4 \times 10^{11}$</td>
<td>1.2</td>
</tr>
<tr>
<td>D</td>
<td>$2.7 \times 10^6$</td>
<td>0.15</td>
<td>$1.3 \times 10^{12}$</td>
<td>2.2</td>
</tr>
</tbody>
</table>

Table 3A.2: Non-dimensional parameters for four transient tests on the Compressor Cavity Rig at the University of Bath

Figure 3A-10 (a) and (b) shows the measured temperatures at different radial and temporal locations at $Re_\phi = 1.4 \times 10^6$ (case B). The radial distribution on the two discs is symmetrical as the heating is symmetric and the cavity closed. The heat fluxes on the two discs are assumed to be equal and data from both discs are used to determine the radial and temporal distributions of the flux. The steady-state temperature distribution at $Fo = 0$ is consistent with experimental data presented in [98], showing a maximum temperature at $x = x_b = 1$. As the transient progressed, the disc temperature at $r = b$ decreased, and, due to the high thermal inertia of the disc material, the temperatures at lower radii reduced less rapidly. The radial locations with the highest temperatures decreased as $Fo$ increases. The discs were gradually cooled to a virtually isothermal condition. Figure 3A-10(c) shows the history of the non-dimensional parameters $Re_\phi$, $\beta\Delta T$ and $Gr$ and $\chi$: as the transient progressed, $\beta\Delta T$ decreased and
$Re_\phi$ was kept constant, leading to a decrease of Gr and an increase of $\chi$.

Figure 3A-10: Transient temperatures and non-dimensional parameters for $Re_\phi = 1.4 \times 10^6$. Symbols: experimental data; red solid lines: calculated temperatures from the Bayesian model. (a) Temperature data histories at different radial locations. (b) Radial data distributions at different temporal locations. (c) Non-dimensional parameters.

Figure 3A-11 shows the transient temperature and the variation of non-dimensional parameters for the other three $Re_\phi$ (cases A, C and D). The temperature distributions for case A ($Re_\phi = 7 \times 10^5$) were similar to those for case B. However, at $Re_\phi > 2.1 \times 10^6$ (cases C and D) the discs remained heated at the end of the transient. This is because at high rotational speeds, the frictional heating at the periphery of the rig was significant, with heat flux to the shroud even without the power of the heater. The final $\beta \Delta T$ at $Re_\phi = 2.8 \times 10^6$ is higher than that at $Re_\phi = 2.1 \times 10^6$, showing a stronger influence of frictional heating at higher rotational speed.
Figure 3A-11: Transient temperatures and non-dimensional parameters for $Re_\phi = 0.7, 2.1$ and $2.7 \times 10^6$ (left, middle, and right, respectively). Symbols: experimental data; red solid lines: calculated temperatures from the Bayesian model. (a) Transient temperatures at $Re_\phi = 0.7 \times 10^6$. (b) Transient temperatures at $Re_\phi = 2.1 \times 10^6$. (c) Transient temperatures at $Re_\phi = 2.7 \times 10^6$. (d) Non-dimensional parameters at $Re_\phi = 0.7 \times 10^6$. (e) Non-dimensional parameters at $Re_\phi = 2.1 \times 10^6$. (f) Non-dimensional parameters at $Re_\phi = 2.7 \times 10^6$.

The Bayesian model has been applied to the temperature measurements in Figures 3A-10 and 3A-11 to determine the transient heat transfer on disc surfaces. Measured temperatures at $Fo = 0, x = x_a = 0.55$ and $x = x_b = 1$ were fitted and used as the boundary conditions to solve Equation (3A.4.5). As discussed in Sections 3A.4 and 3A.5, different instances of the heat fluxes were evaluated until both a good agreement with temperature measurements and a smooth flux distribution were achieved. This was attained by maximising the posterior density function in Equation (3A.5.10). Figures 3A-10 and 3A-11 also presented the agreement between the calculated and mea-
sured temperatures at different radial and temporal locations for all four tests. The standard deviations between the measured \( \Theta \) and the calculated \( \theta \) were about \( 4 \times 10^{-3} \).

Figure 3A-12 shows contours of the calculated fluxes and their standard deviations at \( \text{Re}_\phi = 1.4 \times 10^6 \) (case B). The maximum standard deviation of the calculated heat fluxes is 0.36. At \( \text{Fo} < 0.022 \), positive heat fluxes are achieved at higher radii (heat is transferred from the discs to the cavity air), showing higher temperature in the discs compared to the core. At lower radii, the fluxes are negative, hence the disc temperature is lower than the core temperature. This is consistent with the steady-state buoyancy-induced heat transfer measured in [75, 98]. The radial location \( x^* \) where \( \phi = 0 \) indicates the radius where the disc temperature is equal to the core temperature. Figure 3A-12 (a) shows that at \( \text{Fo} < 0.022 \), \( x^* \) decreases as \( \text{Fo} \) increases, leading to a decrease in \( \theta_{x^*} \) (the disc temperature at \( x^* \)) and hence the core temperature. As \( \text{Fo} \) approaches 0.16, the heat flux gradually decreases to virtually zero, which implies that the flow becomes stratified.

Figure 3A-12: Calculated heat fluxes and standard deviations from experimental data at \( \text{Re}_\phi = 1.4 \times 10^6 \) using the Bayesian model. (a) Heat flux. (b) Standard deviation.

Figure 3A-13 (b) shows the radial distributions of the heat fluxes and their 95% confidence intervals at different \( \text{Fo} \) numbers for case B. At \( \text{Fo} = 0 \), the steady-state disc
heat flux reaches zero at $x^* = 0.7$, which is consistent with experiments in [75]. At $Fo = 0.02$, the radial location is close to $x_a$, showing a dramatic decrease in the core temperature. Though the disc temperature at $x_b$ decreases dramatically, the disc heat flux $\phi_b$ at $Fo = 0.02$ remains at a value of 4. This implies a constant temperature difference between the core and the disc at $x = x_b$, which suggests that the response speed of the core is similar to that of the outer part of the disc and much faster than that of the inner part. The fluxes are positive at all radii when $Fo > 0.04$, which suggests that the core temperature becomes lower than the temperature of the disc.

Figure 3A-13: Calculated heat fluxes (curves) at different $Fo$ and 95% confidence intervals (shaded regions) for four $Re_\phi$ numbers. (a) $Re_\phi = 1.4 \times 10^6$. (b) $Re_\phi = 1.4 \times 10^6$. (c) $Re_\phi = 1.4 \times 10^6$. (d) $Re_\phi = 1.4 \times 10^6$.

Figure 3A-13 also presents the calculated heat flux distributions for cases A, C and
D. The fluxes $\phi$ at $Fo = 0$ stay almost constant from case A to case C and decrease at case D, owing to the increase of the compressibility parameter $\chi_0$ (see Table 3A.2). The calculated heat fluxes for cases A and C show similar behaviour to that for case B: the flow gradually moves away from the buoyancy-induced regime and becomes stratified. For case C, as the shroud is influenced by frictional heating, the compressibility parameter $\chi$ at higher $Fo$ is lower than those for cases A and B. Therefore, it is likely that the flow is in transition to buoyancy-induced flow at $Fo = 0.16$. For case D, at $Fo = 0.16$, the heat flux becomes positive in the outer region and negative in the inner region, showing the characteristics of the buoyancy-induced flow. The maximum heat flux is less than that at $Fo = 0$ due to the decrease of $\beta\Delta T$.

3A.7 Conclusions and future work

This paper developed a Bayesian model to calculate heat fluxes from transient temperature measurements on compressor discs, which is a typical ill-conditioned inverse heat transfer problem. The heat transfer from compressor discs was modelled using a transient annular fin equation. The fin equation defined the forward model in the Bayesian framework, and was used to calculate transient disc temperatures for different instances of transient heat fluxes. A deep neural network was employed to reduce the computational cost of evaluating and differentiating the PDE solution. The spatio-temporal distribution of the heat fluxes were represented using a two-dimensional Chebyshev polynomial, and a Gaussian prior probability distribution over this heat flux with Matérn covariance structure was used to reduce the ill-posedness of the problem. The optimal heat flux distribution, which is referred to as the inverse solution of the fin equation, was estimated through the maximisation of the posterior density function.

The Bayesian model was applied to simulated temperature data. The simulated data was generated by adding noise to ‘true’ disc temperatures, which were calculated by solving the forward problem using a known flux distribution and fixed boundary conditions. The Bayesian model used the noisy data to provide the spatio-temporal distributions of the heat flux and an associated confidence interval. The differences between the calculated and the ‘true’ flux distribution were less than 11%. These differences in fluxes are within the calculated 95% confidence intervals. Therefore, the reliability and accuracy of the Bayesian model was verified.

The model was then applied to four sets of transient temperature data collected from the Compressor Cavity Rig at the University of Bath. The instrumented compressor cavity
was closed with two adjacent co-rotating discs, a shroud and a hub. The operating range of the rig was $7.0 \times 10^5 < \text{Re}_\phi < 2.8 \times 10^6$, $0.0 < \beta \Delta T < 0.15$ and $0.13 < \chi$. The compressor discs were made of titanium alloy and 27 thermocouples on both disc diaphragms were used to measure the disc temperatures. The four transient tests were conducted at four different rotational Reynolds numbers, each started from a steady-state condition. The tests ended with a Fourier number of 0.16.

The transient temperature distributions were discussed. For the two cases with $\text{Re}_\phi < 1.4 \times 10^6$ (cases A and B), the outer part of the disc initially experienced a steep rate of cooling with the entire disc gradually cooling to virtually isothermal conditions. For the two tests conducted at higher $\text{Re}_\phi$ (cases C and D), the outer part of the disc remained heated throughout the transient due to significant frictional heating at high rotational speed.

The distributions of the calculated heat fluxes were then discussed. For all four cases, initially the heat fluxes were positive in the outer part and negative in the inner part, which was consistent with the heat transfer characteristics under buoyancy-induced flow. The flux at $x = x_b$ and $\text{Fo} = 0$ decreased from case C to case D owing to the enhancement of the compressibility effect. The radial location with zero heat flux, $x^*$, reduced as $\text{Fo}$ increased, showing a reduction of the core temperature. The response of the reduction was similar to that of the temperatures at the outer radii. For the two cases at $\text{Re}_\phi < 1.4 \times 10^6$, the disc heat flux gradually decreased towards zero, showing characteristics of stratified flow. For case C, as the disc was heated by frictional heating, the flow tends to converge to buoyancy-induced regime at $\text{Fo} = 0.16$. For case D, the frictional heating was sufficiently strong to retain buoyancy-induced flow across the entirety of the transient process, which is indicated by the positive and negative fluxes in the outer and inner regions.

A theoretical model of transient heat transfer in closed compressor cavities will be developed using the measured temperatures and the calculated fluxes. The data will be used to validate the model and other computational models. The methods developed here will be applied to future temperature data analysis in gas turbine research.
Nomenclature

\[ a \] Inner radius of the disc diaphragm [m]
\[ b \] Outer radius of the disc diaphragm [m]
\[ c \] Coefficients of the Chebyshev polynomial function
\[ E \] Covariance matrix for \( c \)
\[ f \] Term in the Chebyshev polynomial function
\[ F_0 \] Fourier number \( (= \alpha t/b^2) \)
\[ G \] Chebyshev polynomial of the first kind
\[ Gr \] Grashof number \( (= \text{Re} \phi^2 \beta \Delta T) \)
\[ h \] Heat transfer coefficient \( [W/(Km^2)] \) \( (= q/(T - T_{core})) \)
\[ k \] Thermal conductivity of the disc material \( [W/(Km)] \)
\[ K \] Covariance matrix kernel for heat fluxes
\[ N \] Total number of entries
\[ p \] Probability density function
\[ q \] Heat flux \( [W/m^2] \)
\[ r \] Radius [m]
\[ R \] Gas constant for air \( [J/(K kg)] \)
\[ \text{Re}_\phi \] Rotational Reynolds number \( (= \Omega r_s^2/\nu) \)
\[ t \] Time [s]
\[ t_d \] Thickness of disc diaphragms [m]
\[ T \] Disc temperature [K]
\[ x \] Non-dimensional radius \( (= r/b) \)

Greek letters

\[ \alpha \] Thermal diffusivity of the disc material \( [m^2/s] \)
\[ \beta \] Thermal expansion coefficient \( [1/K] \) \( (= 2/(T_s + T_h)) \)
\[ \chi \] Compressibility parameter \( (= \Omega^2 b^2/(\gamma R(T_s - T_h))) \)
\[ \Delta T \] Temperature difference [K] \( (T_s - T_h) \)
\[ \Gamma \] Specific heat ratio of air
\[ \Omega \] Rotational speed of discs [rad/s]
\[ \phi \] Non-dimensional heat flux \( (= 2b^2q/[kt_d(B_{b,0} - T_{ref})]) \)
\[ \rho \] Length scale used in \( K \)
\[ \nu \] Kinematic viscosity of air \( [m^2/s] \)
\[ \theta \] Calculated non-dimensional disc temperature
\[ \Theta \] Measured non-dimensional disc temperature
Subscripts

\( a \) values at \( r = a \)
\( b \) values at \( r = b \)
\( c \) values for Chebyshev polynomial coefficients
\( \text{core} \) values in the cavity core
\( \text{Fo} \) values for Fourier numbers
\( h \) values at hub
\( i \) index for temperature measurements
\( j \) index for two-dimensional Chebyshev polynomials
\( k \) index for first kind Chebyshev polynomials
\( m \) index for one-dimensional Chebyshev polynomials in terms of \( \text{Fo} \)
\( n \) index for one-dimensional Chebyshev polynomials in terms of \( x \)
\( p \) order of Chebyshev polynomial functions
\( r \) values for radius
\( s \) values at shroud
\( t \) values for time
\( \text{true} \) true value for simulated data
\( x \) values for non-dimensional radius

Acknowledgments

The authors wish to thank Edmund Harberd for his work on data collection. We appreciate the technical conversations with and the experimental support from Dr Carl Sangan, Dr Oliver Pountney, Dr James Scobie and Dr Richard Jackson in the Turbomachinery Research Centre at the University of Bath. This work is dedicated to Professor Mike Owen, who although no longer with us, continues to inspire with his passion for research.

Data Availability Statement

The data that supports the findings of this study are available within the article.
Chapter 4

PDE informed statistical modelling of spatio-temporal pollution

This chapter contains a manuscript in preparation. Here we investigate how the deep surrogate methodology, as described in Chapters 2 and 3, can be applied to the spatio-temporal statistical modelling setting by injecting physical information into models that are often more data-driven. In particular, motivated by air pollution in Ulaanbaatar, we explore the situation of making more reliable inferences when spatial data coverage is sparse.

There are two key contributions of this chapter. The first is the development of a full Bayesian analysis of the air pollution prediction problem, which uses a custom Hamiltonian Monte Carlo sampler to fit a hybrid model consisting of physical and Gaussian process terms. This contrasts with typical statistical approaches to physical problems, where physical models are solved in advance, then incorporated into statistical methodology. Instead, we ensure that the fit and uncertainty derived from both the physical and Gaussian process components, and their interdependence, is accurately captured by jointly inferring their posterior distribution. Another contribution is the application to sparse data problems. In Ulaanbaatar, for example, we had access to pollution data at only 9 spatial locations. Through the incorporation of a physical model into a spatio-temporal statistical model, we demonstrate a more principled understanding of the behaviour of pollution in the city than a purely data-driven approach.
This declaration concerns the article entitled:

PDE informed statistical modelling of spatio-temporal pollution.

<table>
<thead>
<tr>
<th>Publication status (tick one)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft manuscript ✓</td>
<td>Submitted</td>
</tr>
</tbody>
</table>

Publication details (reference)

Copyright status (tick the appropriate statement)
I hold the copyright for this material ✓ Copyright is retained by the publisher, but I have been given permission to replicate the material here

Candidate’s contribution to the paper (provide details, and also indicate as a percentage)
The candidate predominantly executed the…

Formulation of ideas:
90%. The author of this thesis developed the project concept independently following a research visit to Mongolia. This idea was expanded on during discussions with all other authors.

Design of methodology:
100%. The author of this thesis developed the hybrid physical-statistical model underpinning this work, also developing the framework for fitting this model.

Experimental work:
100%. All code was developed and implemented by the author of this thesis.

Presentation of data in journal format:
90%. The author of this thesis produced all results and drafts of the manuscript. Feedback was provided by all other authors.

Statement from Candidate
This paper reports on original research I conducted during the period of my Higher Degree by Research candidature.

Signed
Teo Deveney  
Date 05/06/2022
Abstract

We present an approach where the deep surrogate methodology developed in [38] is applied to air pollution prediction. As a motivating example we consider the case of air pollution in Ulaanbaatar, Mongolia, which is one of the world’s most polluted capital cities. Despite these high pollution concentrations, there is relatively little data available to analyse the spatio-temporal pollution field. In this work we adopt a physically-informed Bayesian hierarchical modelling approach, by including a neural network approximation to a parameterised pollution transport PDE as the mean function in a spatio-temporal Gaussian process (GP) model. The parameters of this PDE are treated as hyperparameters of the GP, and are inferred from data jointly with covariance hyperparameters defining the correlation structure of spatio-temporal kernel. A fully Bayesian analysis of this model is implemented, which uses a custom Hamiltonian Monte Carlo scheme to simultaneously sample from the posterior distribution of these physical and statistical terms. We produce a posterior spatio-temporal field describing the air pollution in the city, which can be disaggregated into physical and statistical components with interdependence described by the inferred posterior. In addition, after identifying a relationship between temperature and pollution concentration in Ulaanbaatar, we investigate how this posterior field varies when applied to data aggregated using dates with differing temperatures. A significant contribution of this work lies in the data that we are able to analyse. This air pollution data is sparsely distributed, and we demonstrate that our approach gives physically principled estimates and uncertainties. We show that this allows us to make more generalisable inferences in comparison to data-driven GP regression approaches, both applied to the raw data, or as a post-processing step to residual data after subtracting a physical model. Finally, we conclude with some ideas that could be implemented in order to enhance this approach in future, including the description of a neural network construction that guarantees that mass of the PDE solver is controlled appropriately, and the incorporation of more complex modelling assumptions such as an approach that could be applied to explicitly infer the effect of temperature fluctuations.
4.1 Introduction

Statistical modelling of air pollution is an active field with important applications for public health and climate change [22, 31]. Research in this topic is often centred around data relating to specific regions (see for example [24, 30, 93]), meaning that the results can be acted on locally, for example by informing government policies. Here, we are motivated by developing a physics-informed approach to understand the spatio-temporal structure of the daily air pollution movement in Ulaanbaatar, the capital city of Mongolia [5]. In Ulaanbaatar, pollution levels are often attributed to the daily schedules of citizens, leading them to fluctuate over short timescales. The concentrations of these pollutants are recorded at a small number of monitoring stations each hour, and from this data it can be observed that there are large increases in pollution concentrations at certain times of day. These spikes in pollution can have considerable health implications for young children and vulnerable groups, thus it is vital to develop spatio-temporal statistical models that will reliably capture the evolution of pollution concentrations over the course of the day in the city.

The field of spatio-temporal statistical modelling has undergone dramatic change in recent years. As we gather larger datasets, a requirement has arisen for algorithms that can scale favourably with this data, leading to a focus on scalable Gaussian process (GP) regression techniques. This focus has led to various breakthroughs such as the use of stochastic FEM solutions to construct sparse GP discretisations by solving an associated SPDE [95, 134], random Fourier features to define reduced order Monte Carlo approximations to the covariance kernel [123], and sparse variational inference methods based on summarising the data using a relatively small number optimised artificial ‘inducing’ data points [154]. In our setting however, we are faced with the often overlooked contrasting issue of deriving reliable inferences from relatively little data. More specifically, in our setting there are large distances between some monitoring stations in Ulaanbaatar, resulting in uncertainty dominating a large proportion of the posterior Gaussian field if state of the art Gaussian process spatio-temporal modelling methods are applied directly to the data. To account for this, systematic considerations of the data generating process must be taken into account. One such example of this, that we develop in this work, is the inclusion of physical models governing pollution dispersion.

A common statistical approach to incorporating physics is to first subtract the output of a physical model from the raw data, before applying a statistical scheme such
as GP regression to the residual data as a spatial post-processing step (for example, [51, 130]), which corrects for effects not captured by the physical model. Assuming the physical model can capture the large-scale pollution behaviour, this approach can vastly reduce the uncertainty when compared to GP regression on the raw data, however it assumes that the physical model output is deterministic. In reality, the output of the physical model is likely to be uncertain, and the spatial and temporal correlations of this uncertainty are likely to be more reflective of the underlying physical process than those produced by distance-based kernel methods as applied in GP regression. As we discuss in this work, the spatial post-processing approach overlooks this source of uncertainty, leading to inferences that are overly confident in comparison to approaches that account for uncertainty in the physical model.

Conversely, an alternative approach is to model the movement of pollution using a physical model with unknown parameters. This problem can then be cast as a Bayesian inverse problem [147], where uncertainty in these parameters can be quantified using Bayesian inference, thus producing a posterior pollution field governed entirely by the physical modelling assumptions. This approach ensures the inferred pollution and its uncertainty obeys the underlying physical model; however this is a heavy restriction, as in reality behaviour often presents in the data that are not resolved by the physical model. Examples of such factors might include topographical considerations or localised turbulence patterns. Unfortunately these factors complicate the models, resulting in a requirement for more expensive numerical solvers that typically make the Bayesian inverse problem approach intractable. Bayesian model averaging [70] is an alternative approach that can utilise expensive model simulations if an ensemble of such simulations are available, though typically this approach is restricted by a limited ensemble (for example [122] applies this approach in a meteorological setting using just 5 ensemble members).

In this work we instead propose a hybrid model that involves both physical and statistical aspects. In our approach we approximate the large-scale physical behaviour by a pollution transport PDE, and include a Gaussian process in addition to represent any spatio-temporal processes that are present in the data but unresolved by the large-scale PDE behaviour. We then apply a Bayesian approach, in which the posterior distribution over the PDE solution and Gaussian field are inferred jointly using Hamiltonian Monte Carlo (HMC). The resulting combined posterior pollution field therefore accounts for both physical and data-driven sources of uncertainty. To implement this model in a computationally efficient manner, we train a deep learning surrogate to
solve the pollution transport model over a range of parameters and use this surrogate within the HMC sampler. Unlike more traditional solvers such as finite differences, this surrogate solution can be evaluated and differentiated rapidly with respect to the PDE parameters, thus allowing us to efficiently simulate the proposal trajectories involved in HMC. Our method is then applied to learn about the hourly pollution evolution over the course of a day, using both simulated data and the Ulaanbaatar air pollution sensor network. In particular we demonstrate that we can recover information about the time and location of pollution sources, and produce a posterior space-time field that can be used to estimate the local pollution throughout the day. This posterior field is shown to have significantly reduced uncertainty in comparison to the purely data-driven GP regression approach, in particular at larger distances from the data locations.

This work is organised as follows. We begin by describing the Ulaanbaatar setting in detail. Here we take a first look at the data and highlight some daily and seasonal patterns which are used to inform our physical modelling assumptions. Following this, in Sections 4.3 and 4.4 we introduce the methods we will use to analyse the data; these include Gaussian process models often used in spatial statistics, and the Langevin dynamics and advection-diffusion physical models that are commonly applied to represent the movement of pollution. The results achieved by fitting some of these models to the raw data are also demonstrated. In Section 4.5 we outline our new hybrid approach in detail and present some results demonstrating the advantages of this approach using both real and simulated data. Finally we conclude this work by outlining future directions that could be explored to further enhance this approach. These include the description of a mass-controlled neural network algorithm, which can guarantee that the integral of the deep learning surrogate is consistent with the true solution (i.e. mass is conserved appropriately), as well as extensions to the modelling assumptions that would allow us to better understand the relationship between pollution and meteorological factors such as temperature and wind velocity.

4.2 Ulaanbaatar setting

We will motivate our methodology throughout using the example of air pollution in Ulaanbaatar. As the monitoring network gathering air pollution data is sparse, we believe the proposed method combining a physical model for air pollution dynamics along with a Gaussian process capturing additional variability in the data will improve the predictions of air pollution concentrations in space and time compared to a more
data-driven statistical approach.

Ulaanbaatar, the capital of Mongolia, is amongst the most polluted capital cities in the world. The high levels of pollution are driven mainly by the domestic burning of coal, which releases particulate matter and other pollutants into the atmosphere [5]. This stands in contrast to other highly polluted cities, where industry and vehicles tend to be the largest contributors. Because of this, the situation in Ulaanbaatar shows some unique characteristics. In this section we introduce the available data, and do some visual investigation into factors that might aid our physical modelling assumptions.

The data available is sparse in space; including hourly concentration readings at just 9 locations for the pollutants PM2.5, PM10, NO2 and SO2. Table 4.1 shows the completeness of the data for these stations and Figure 4-1 shows the geographical locations of the stations in longitude-latitude coordinates on a map of Ulaanbaatar. These stations are operated by various organisations and the completeness of the data varies accordingly. The data provided in Table 4.1 shows the proportion of data available for each pollutant at each of the nine stations for the years 2015-2018. Additional data about the atmospheric conditions, including temperature, wind speed and direction, relative humidity and air pressure, is also available. These values are provided for the entire domain for the period 2014 to 2018.

<table>
<thead>
<tr>
<th>Station</th>
<th>SO2</th>
<th>NO2</th>
<th>PM10</th>
<th>PM2.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baruun 4 zam</td>
<td>82.6%</td>
<td>68.8%</td>
<td>86.1%</td>
<td>83.1%</td>
</tr>
<tr>
<td>Bukhiin urguu</td>
<td>82.4%</td>
<td>81.3%</td>
<td>82.5%</td>
<td>68.7%</td>
</tr>
<tr>
<td>100 Ail</td>
<td>85.2%</td>
<td>85.5%</td>
<td>85.4%</td>
<td>0%</td>
</tr>
<tr>
<td>Mongol gazar</td>
<td>86.5%</td>
<td>71%</td>
<td>81.7%</td>
<td>0%</td>
</tr>
<tr>
<td>Tolgoit</td>
<td>77.3%</td>
<td>84.1%</td>
<td>85.8%</td>
<td>85.8%</td>
</tr>
<tr>
<td>Televiz</td>
<td>77.7%</td>
<td>96.7%</td>
<td>92.5%</td>
<td>92.5%</td>
</tr>
<tr>
<td>Amgalan</td>
<td>90.6%</td>
<td>90.5%</td>
<td>85.1%</td>
<td>85.1%</td>
</tr>
<tr>
<td>Nisekh</td>
<td>81.8%</td>
<td>89.3%</td>
<td>86.2%</td>
<td>86.2%</td>
</tr>
<tr>
<td>Bayankhoshuu</td>
<td>86.7%</td>
<td>0%</td>
<td>83.0%</td>
<td>82.5%</td>
</tr>
</tbody>
</table>

Table 4.1: The percentage of complete records of pollutants at Ulaanbaatar monitoring stations.

We will focus on the concentrations of the pollutants PM2.5 and PM10, which are measured in $\mu g/m^3$. These are fine particles which are inhaled deep into the lungs, leading to a variety of respiratory issues and adverse pregnancy outcomes [3]. PM2.5
particles are finer and more dangerous, since they can penetrate more deeply into the lungs. Unfortunately, a significant portion of this data is missing. To overcome this, we instead elect to focus on PM10 concentrations which are highly correlated to PM2.5 (with a correlation coefficient of 0.74 amongst the available data). Missingness in the PM10 data is presumed to be missing at random for the analysis that follows, and is interpolated in time. To ensure that interpolation in time was reasonable (i.e. that we are not interpolating over large periods of hours/days), we subset the data and use months where there are minimal time gaps. A potential piece of further analysis would be to test the missing at random assumption and investigate alternative approaches should the data instead be missing not at random.

The districts in the north and west of the city contain densely populated areas known as ger areas. In these districts people primarily live in gers, which are yurt-like structures that are commonly heated entirely by coal burning stoves. As a result, the concentrations of PM10 tend to be higher at monitoring stations which are closer to these areas. Evidence to this effect is presented in Figure 4-2, as we can see that the regression line for Bayanhoshuu attains the highest value, while Televiz, 100 ail and Tolgoit all

![Figure 4-1: Locations of air pollution monitoring stations in Ulaanbaatar.](image-url)
show above average PM10 levels. These stations are all located in the west or north side of the city, and so we expect our inferred pollution field to be higher in these areas. Figure 4.2 shows that the PM10 concentrations are negatively correlated with the temperature, and the logarithmic scale suggests that this relationship is log-linear. In Section 4.7 we indicate how this can inform our methodology in order to establish the relationship between the temperature and the amplitude of the pollution sources.

Figure 4-2: Daily average PM10 levels (in $\mu g/m^3$) and temperatures (°C) in 2018.

Figure 4-3 displays the hourly pollution across each week in November and December 2018. For brevity, this is presented for three of these stations, but the daily pattern demonstrated is consistent across all locations. Overlaid is a regression line which is computed using a locally averaged polynomial regression fitting technique. These figures indicate a daily repeating pattern to the pollution which consists of a double peak each day, typically one just before midday and one in the evening. The timings of this behaviour are consistent across the stations, however the amplitudes of the two peaks seem to vary depending on location.
We can take a closer look at the daily pattern by overlaying the hourly pollution levels for each day in November and December 2018. This is presented for a further 3 stations in Figure 4-4. Overlaid on this is a regression line (again computed using locally averaged polynomial regression) which shows the pollution begins to increase at around 7am, peaking close to midday and reducing, before increasing again at 5pm and reaching a peak at midnight.

Figure 4-5 shows similar information to Figure 4-4 for the remaining stations, however in this plot we display only days when the average temperature was between $-10^\circ$C and $-5^\circ$C or between $0^\circ$C and $5^\circ$C. This figure shows that the same daily cycle remains despite these temperature differences, and we see that our previous observation that warmer temperatures are correlated with lower levels of pollution remains true on average during all hours of the day.
In this introduction to the available data we have identified some key characteristics of pollution in Ulaanbaatar. This analysis has been simple, however the information obtained will be invaluable in informing the spatio-temporal modelling that follows in the remainder of this work. For example Figures 4-3, 4-4, 4-5 allowed us to see the hourly evolution of pollution concentrations, and therefore we will ensure that the parameterisation of our physical model in Section 4.5 allows for the double peaked daily structure revealed in these plots. This work focuses mostly on the spatio-temporal evolution of pollution, however motivated by temperature trends in Figures 4-2, 4-5 we apply our method to data taken from periods with differing temperatures in Section 4.8 to demonstrate the differences in concentrations. Furthermore, in Section 4.7 we describe how our method could be extended to directly model the relationship between temperature and pollution source intensity. Our goal from here will be to outline a physics-informed Bayesian methodology to estimate the average spatio-temporal evolution of pollution concentrations over the course of a day.

4.3 Statistical pollution models

4.3.1 Spatial models

One of the standard approaches to spatial analysis of pollution is to apply stationary Gaussian process regression [116, 129] (often called kriging in geostatistical settings) or an extension of this [17, 141]. For the purpose of exposition we begin by outlining this setting in detail, as the method of computing the posterior field is generalisable to the spatio-temporal and physics-informed models that follow. A stationary GP regression
The model can be written

\[ y_i = \eta(x_i; \theta^n) + \epsilon_i. \]  

(4.3.1)

Here \([x_1, \ldots, x_M] = X\) are the longitude-latitude coordinates of the pollution sensor, \([y_1, \ldots, y_M] = Y\) are the pollution readings at those locations, and \(\epsilon_i \sim \mathcal{N}(0, \sigma^2_\epsilon)\) are i.i.d variables representing random deviations from the underlying field due to factors such as sensor measurement error. The spatial field \(\eta(\cdot)\) is an unknown function which is inferred using the data; it is endowed with a Gaussian process prior distribution \(\mathcal{GP}(0, K_x(\cdot, \cdot; \theta^n))\). A commonly used covariance kernel, that we apply in this work, is the Matérn covariance

\[
K_x(x, x' | \theta^n) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left( \frac{\sqrt{2\nu \|x - x'\|}}{\rho_x} \right)^\nu \frac{\Gamma(\nu)_{\sqrt{2\nu \|x - x'\|}}}{\nu K_{\nu} \left( \sqrt{2\nu \|x - x'\|} \right)}. 
\]

(4.3.2)

Here \(\Gamma(\cdot)\) is the gamma function and \(K_\nu(\cdot)\) is the modified Bessel function of the second kind. The constants \(\theta^n = \{\sigma^2, \rho_x, \nu\}\) are hyperparameters of the covariance, which may themselves have prior distributions, or may be fixed to regularise the predictions. The hyperparameters consist of the marginal standard deviation of the field \(\sigma^2\), the spatial correlation length scale \(\rho_x\), and a smoothness parameter \(\nu\) that controls the differentiability of the field.

Given the model \(4.3.1\) and the data \(X, Y\), one typically wants to recover the posterior distribution of the pollution level \([y'_1, \ldots, y'_K] = Y'\), at new locations \([x'_1, \ldots, x'_K] = X'\), conditional on the data. Achieving this means we can evaluate the posterior latent field \(\eta(x; \theta^n)\) conditional on the data for any \(x\) of interest; thus we can reconstruct the full posterior field. According to the model \(4.3.1\), the pollution levels \([y_1, \ldots, y_M, y'_1, \ldots, y'_K]\) have a multivariate Gaussian distribution with the covariance matrix

\[
\Sigma = \begin{bmatrix}
K_x(X, X | \theta^n) + \sigma^2_\epsilon I_M & K_x(X, X' | \theta^n) \\
K_x(X', X | \theta^n) & K_x(X', X' | \theta^n)
\end{bmatrix}.
\]

(4.3.3)

Here \(K_x(X, X' | \theta^n)\) denotes the covariance matrix where \(K_x(X, X' | \theta^n)_{ij} = K_x(x_i, x'_j | \theta^n)\). For fixed Matérn hyperparameters \(\theta^n\) and noise level \(\sigma^2\), the posterior distribution of \(Y'\) conditional on \(X, Y, X'\) is the \(\text{MVN}(\mu', \Sigma')\) distribution

\[
p(Y' | X, Y, X', \theta^n, \sigma^2) = \frac{1}{2\pi \Sigma'}^{-\frac{1}{2}} e^{-\frac{1}{2}(Y' - \mu')^T \Sigma'^{-1} (Y' - \mu')},
\]

(4.3.4)
where the mean and covariance are
\[
\mu' = K_x(X', X|\theta^o)[K_x(X, X|\theta^o) + \sigma^2 \epsilon I_M]^{-1} Y^\top,
\]
\[
\Sigma' = K_x(X', X'|\theta^o) - K_x(X', X|\theta^o)[K_x(X, X|\theta^o) + \sigma^2 \epsilon I_M]^{-1} K_x(X, X'|\theta^o).
\]

Computing \( \mu', \Sigma' \) directly is straightforward using linear algebra software, though it comes with an \( \mathcal{O}(M^3) \) computational cost for \( M \) data points due to the inversion of \( K_x(X, X|\theta^o) + \sigma^2 \epsilon I_M \), which is a dense \( M \times M \) matrix in general. Otherwise, reduced dimension approximations to this such as [95] can be applied to reduce this cost.

If the Matérn hyperparameters \( \theta^o \) and noise level \( \sigma^2 \epsilon \) are unknown then we can assign priors to them and include them in the inference. In this case the model (4.3.1) is a hierarchical model [52, 53], in which the Gaussian process prior distribution over the latent field depends on hyperparameters which themselves have prior distributions. Inference in these models first involves computing the posterior \( p(\theta^o, \sigma^2 \epsilon | X, Y) \) over the hyperparameters. The distribution of the posterior latent field at out-of-sample locations \( X' \) is then achieved by marginalising over these hyperparameters, giving
\[
p(Y'|X, Y, X') = \int \int p(Y'|X, Y, X', \theta^o, \sigma^2 \epsilon) p(\theta^o, \sigma^2 \epsilon | X, Y) d\theta^o d\sigma^2 \epsilon.
\]

An approximation of \( p(\theta^o, \sigma^2 \epsilon | X, Y) \) can be obtained using an MCMC scheme such as Hamiltonian Monte Carlo (HMC) [13], which uses the priors over \( \theta^o \) and \( \sigma^2 \epsilon \), and the likelihood over the data, which by (4.3.1) is the \( \text{MVN}(0, K_x(X, X|\theta^o) + \sigma^2 \epsilon I_M) \) density evaluated at \([y_1, \ldots, y_M] \). Once HMC is complete and we have samples \([\theta^o_1, \ldots, \theta^o_{N_{\text{samples}}}], [\sigma^2 \epsilon_1, \ldots, \sigma^2 \epsilon_{N_{\text{samples}}}] \) from \( p(\theta^o, \sigma^2 \epsilon | X, Y) \), the marginalised predictive posterior field (4.3.6) can be approximated as
\[
p(Y'|X, Y, X') \approx \frac{1}{N_{\text{samples}}} \sum_{n=1}^{N_{\text{samples}}} p(Y'|X, Y, X', \theta^o_n, \sigma^2 \epsilon_n),
\]
where each entry in this sum is the posterior PDF (4.3.4) obtained by applying (4.3.5) to compute the mean and covariance of the posterior Gaussian latent field corresponding to each \( \theta^o \) and \( \sigma^2 \epsilon \) in the HMC sample. The ensemble (4.3.7) can be viewed as a Gaussian mixture model [16, 131], though it is not Gaussian itself in general. For visualisation purposes we are interested in the mean and variance of (4.3.7), which are
given by

\[ E[Y'] = \frac{1}{N_{\text{samples}}} \sum_{n=1}^{N_{\text{samples}}} \mu'_n, \quad (4.3.8) \]

\[ \text{Var}(Y') = \frac{1}{N_{\text{samples}}} \sum_{n=1}^{N_{\text{samples}}} (\sigma'_n)^2 + \left( \frac{1}{N_{\text{samples}}} \sum_{n=1}^{N_{\text{samples}}} (\mu'_n)^2 - \left( \frac{1}{N_{\text{samples}}} \sum_{n=1}^{N_{\text{samples}}} \mu'_n \right)^2 \right). \]

Here \( \mu'_n \) is the GP posterior mean conditional on hyperparameters \( \theta_n^2, \sigma^2 \epsilon_n \) (according to (4.3.5)), and \( (\sigma'_n)^2 \) is the corresponding variance, consisting of the diagonal elements of \( \Sigma_n' \). The interpretation of (4.3.8) is intuitive, as it simply says that the overall mean of (4.3.7) is the average of the sampled posterior means corresponding to each hyperparameter, while its variance is the average of the posterior variances plus the variance over the posterior means. As (4.3.7) is an approximation of (4.3.6), it follows that (4.3.8) is a Monte Carlo approximation to the mean and variance of the posterior Gaussian field (4.3.6) after hyperparameter marginalisation.

It is worth noting that in the vast majority of practical applications the hyperparameters are assumed fixed. This has a significantly lower computational cost, as the conditional mean and covariance (4.3.5) need only be computed once. In these cases the hyperparameters are either chosen manually to enforce a certain correlation structure, or more commonly optimised a-priori using maximum-likelihood or MAP. This optimisation is usually sufficient to achieve good results and the full approximation of (4.3.6) is not generally recommended due to its cost. Nonetheless we describe this approach to marginalising the hyperparameters now, as we will see that it generalises to the inference and marginalisation of the physical processes which will be utilised in subsequent models, which will allow us to achieve a more robust quantification of the overall uncertainty.

### 4.3.2 Spatio-temporal models

The previous model (4.3.1) is stationary in time, which is insufficient if we are interested in temporal changes in pollution, and so an extension is required. A spatio-temporal model can similarly be written

\[ y_i = \eta(t_i, \mathbf{x}_i; \theta^n) + \epsilon_i. \quad (4.3.9) \]
In this case the Gaussian field $\eta(t, x; \theta_0)$ evolves in space and time with hyperparameters $\theta_0$, and the $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$ are identical to the stationary model described previously. We assume again that data is recorded at $M$ locations, and that this occurs at $T$ time points. Again $Y = [y_1, \ldots, y_{TM}]$ represents the recorded pollution concentrations, and $t = [t_1, \ldots, t_{TM}], X = [x_1, \ldots, x_{TM}]$ are the corresponding temporal and spatial coordinates of those readings respectively. In general this Cartesian product assumption on the structure of the data is not required and $t, X$ can be arbitrary coordinates in time and space, however this data structure is very common (indeed our data is structured this way) and can allow us to save significant computational effort through specific choice of covariance kernels.

In principle, any covariance kernel $K([t, x], [t', x]; \theta_0)$ describing spatio-temporal correlations can be applied for the Gaussian field, and the procedure for computing the posterior latent field given this kernel is analogous to the stationary case described in Section 4.3.1. However the covariance matrix is now $K([t, X], [t, X]; \theta_0) \in \mathbb{R}^{TM \times TM}$ and so the computational requirement is $O(M^3 T^3)$, a costly increase proportional to the cube of the number of time points. A simplification that can be used to accelerate this is to apply a space-time separable covariance structure \cite{95}. To do this we define a temporal covariance that is independent of the spatial covariance; for example in this work we apply an autoregressive covariance structure \cite{143} to the temporal evolution of the field, leading to a correlation between common points at different times of

$$K_t(t, t') = \rho_{t}^{\lfloor t-t' \rfloor}, \quad \rho_t \in [0, 1]. \quad (4.3.10)$$

Here $\rho_t$ is a temporal length scale hyperparameter representing the correlation between subsequent time points, which can be fixed or inferred along with the other hyperparameters. Denoting the temporal covariance matrix over the $T$ time points by $K(t, t)$, where $K(t, t)_{i,j} = K_t(i, j)$, the full spatio-temporal covariance matrix is then given by the Kronecker product

$$K([t, X], [t, X]; \theta_0) = K_t(t, t) \otimes K_x(X, X). \quad (4.3.11)$$

Here $\theta_0 = \{\sigma_x^2, \rho_x, \nu, \rho_t\}$ is the combined set of hyperparameters of the spatial and temporal kernels. To invert the matrix (4.3.11) we need only invert the two factors and compute their Kronecker product, and by calculating the singular value decomposition of each factor we can invert $K([t, X], [t, X]; \theta_0) + \sigma_x^2 I_M$ as required to compute the temporal analogue of the posterior mean and covariance written in (4.3.5). Thus the computational cost of working with this model is reduced to $O(M^3 + T^3)$. As
a consequence of this simplification, separable models such as (4.3.9) with covariance (4.3.11) are unable to fully represent non-trivial space-time correlations in the data. In most cases this is a worthwhile trade-off that has little impact on the overall prediction.

Applying this model to the raw Ulaanbaatar data from November 2018 (using the Matérn spatial and autoregressive temporal covariance kernels), we display the mean and standard deviation, as defined in 4.3.8 of the hourly posterior pollution levels in the city after marginalisation of the hyperparameters.

Figure 4-6: Hourly posterior mean (in $\mu g/m^3$) of the autoregressive Matérn Gaussian field applied to the raw Ulaanbaatar data for November 2018.
Figure 4-6 shows gives us a general idea of the mean pollution behaviour over the course of the day. From left to right, the first row in this figure shows the posterior mean pollution level at 00:00am, 01:00am, 02:00am, 03:00am, the second row from 04:00am to 07:00am, and so on. We plot the city map in the first panel to give an idea of locations, but omit this in the remaining panels so that the features of the prediction are unobscured. To ensure consistency with the physics-based approaches that follow, the Gaussian field is fitted to a standardised dataset, with longitudinal co-ordinates mapped to lie within $[-0.6, 0.6]$, latitudes scaled proportionally and centred at zero, and pollution values divided by a factor of 10. The smoothness hyperparameter for this field is fixed to $\nu = 2.5$. The remaining hyperparameters are assigned priors $\sigma \sim \Gamma(2, 1)$, $\sigma_\epsilon \sim \Gamma(4, 4)$, $\rho_x \sim \Gamma(2, 4)$, $\rho_t \sim \beta(3, 3)$ (where $\Gamma(\text{shape, rate})$ is the gamma distribution, and $\beta(a, b)$ is the beta distribution), then inferred using HMC and marginalised as described in Section 4.3.1. These priors are chosen to respect the ranges in the data (for example the largest distance between mapped data locations is 1.23 so the prior for $\rho_x$ assigns the majority of its mass to values below this), but are relatively uninformative otherwise.

The white circles in Figure 4-6 represent the data locations, and the fill colour of these circles is the mean pollution concentration observed in the data for that location at that time of the day. As we would expect, the hourly fluctuations are consistent across the city, with the amplitude of the pollution appearing to be higher in the northern and western regions. We see that the GP prediction is consistent with the data (as the coloured circles match the surrounding field), and that these predictions generalise to some local region around the data. Unfortunately we also observe significant behaviour that is not physically expected, such as the prediction of zero pollution in the south, where there is no data, and the disjointedness of the pollution patches in the early afternoon displayed in row 4. A further issue is that the uncertainty of these predictions is high away from the data locations as shown in Figure 4-7.
Figure 4-7: Spatial uncertainty field ($\mu g/m^3$) showing the width of the 95% credible interval of the posterior autoregressive Matérn Gaussian field applied to the raw Ulaanbaatar data for November 2018.

This figure is consistent at each time point so we only show a single spatial field. As we would expect the predictions are more certain near the stations, but this uncertainty increases quickly as the prediction moves away from these locations. As a result uncertainty dominates the inference over most of the domain, and uncertainties of over $\pm 150 \mu g/m^3$ are common even in densely populated areas. This sheds some light on the unphysical behaviour, as the GP prediction is attracted back towards its prior mean value of zero in areas of high uncertainty. This is a simple approach based only on spatio-temporal information, and predictive improvements could likely be made by investigating the impact of physical behaviour or additional covariates such as the meteorological conditions. We next investigate how an approach to prediction and uncertainty quantification based purely on physical assumptions differs from this approach.

4.3.3 Bayesian inverse problem approach

A direct approach to incorporating physics into the inference would be to replace the Gaussian field in (4.3.9) with a physical model $u(t, \mathbf{x}; \theta^u)$ with uncertain parameters,
and compute the posterior predictive distribution of this model conditional on the data. In this approach we assume the model is related to the data through the statistical relationship

\[ y_i = u(t_i, x_i; \theta^u) + \epsilon_i. \]  

(4.3.12)

Here \( u(t_i, x_i; \theta^u) \) represents the solution to some differential equation describing atmospheric dispersion, \( \epsilon_i \sim N(0, \sigma^2_\epsilon) \) are i.i.d Gaussian noise, and we treat the physical model parameters \( \theta^u \) as unknowns that can depend on the data \( \{t_i, x_i, y_i\} \), as well as relevant data about the meteorological conditions. Given appropriate priors over \( \theta^u \) and \( \sigma^2_\epsilon \), approximating the posterior distribution \( p(\theta^u, \sigma^2_\epsilon | t, X, Y) \) over these parameters is a Bayesian inverse problem [147], which can be achieved using MCMC in a similar manner to that used to sample the GP hyperparameters previously. The likelihood in this case is the \( \text{MVN}(u(t, X; \theta^u), \sigma^2_\epsilon I_{TM}) \) density evaluated at \( Y \). The unknown latent field is governed by the PDE solution, therefore the prediction at out-of-sample coordinates \( t', X' \), is \( Y' = u(t', X'; \theta^u) \), which has posterior predictive distribution

\[ p(Y'|t, X, Y, t', X') = \int \int p(u(t', X'; \theta^u)|t', X') p(\theta^u, \sigma^2_\epsilon | t, X, Y) d\theta^u d\sigma^2_\epsilon. \]  

(4.3.13)

Given an MCMC sample \( [\theta^u_1, \ldots, \theta^u_{N_{\text{samples}}}] \), this marginalisation of the physical parameters can be approximated similarly to how the hyperparameters were marginalised in (4.3.7) for the stationary GP case. More usefully its empirical mean, variance, and correlation structure can be calculated directly by evaluating \( u(t', X'; \theta^u_n) \) for each \( \theta^u_n \) in the sample and using the appropriate formula for the empirical statistics. For a given \( \theta^u \) the PDE model is deterministic. This is unlike the GP models previously described, for which \( \theta^\eta \) defines the covariance structure of a GP prior distribution. As a result (4.3.12) is not a hierarchical model, and therefore no further conditioning steps are required.

The precise construction of the physical model determining \( u(t, x; \theta^u) \) can have a large impact on the quality of the predictions. Simpler models that describe larger scale advective and diffusive movement provide a means to model the dominant behaviour, however more specific patterns in the data such as turbulence may require more complicated models in order to be accurately represented. Similarly to high-dimensional statistical models, highly parameterised physical models can represent more complex behaviour, but are more likely to over-fit without effective priors. Furthermore, higher dimensional posterior distributions require more samples in order to achieve the same
accuracy (see for example the Dvoretzky-Kiefer-Wolfowitz inequality \[43\] for the relation between empirical approximation accuracy and density dimension in the case of i.i.d. samples), therefore MCMC applied to models with high dimensional $\theta^u$ also converge more slowly to the true posterior distribution.

Balancing these factors of accuracy and efficiency in conjunction with prior distribution is a difficult task. Due to computational constraints, the most common choice applied in Bayesian approaches is to use a simple model. An example of a simple model applied to the Ulaanbaatar data is the advection-diffusion dynamics \[28\]. We visualise the predictions obtained by applying the Bayesian inverse problem approach with advection-diffusion dynamics in Section \[4.5.3\] as they are produced as part of the hybrid approach that we introduce in Section \[4.5\]. The mean prediction attributable to this Bayesian inverse problem approach can be seen in the first column of Figure \[4.10\] and the uncertainty associated with this prediction is shown in the middle panel in Figure \[4.11\]. The physical modelling assumptions are outlined in Sections \[4.4\] and \[4.5.1\]. The methodology used to produce those figures differs slightly from that written here (correlated noise is assumed as described in Section \[4.5\]), however the result is visually identical to that achieved in the current set-up so we omit additional figures here for brevity.

In Section \[4.4\] we describe commonly applied models of pollutant movement which can be used as the mean function $u(t, x; \theta^u)$ underpinning this Bayesian inverse problem approach. We begin this discussion by describing the Langevin dynamics \[92\], which are based upon physical principles implied by Newton’s second law, and used in industrial atmospheric dispersion settings \[79, 80\]. In particular we highlight how the more realistic Langevin dynamics can be computationally costly to work with, before explaining how these result in the simpler Brownian dynamics in the limit of low particle mass. Using these Brownian dynamics we derive the advection-diffusion model that our method will be based upon.

### 4.4 Physical pollution models

Pollution concentrations are determined by the density of pollutant particles in the air. A physically accurate description of particle movement is the Langevin dynamics. In the Langevin dynamics the positions and velocities of particles in a turbulent flow are modelled by applying Newton’s second law with random forces representing turbulence,
and friction. This leads to the set of coupled governing SDEs

\begin{align*}
    dV_t &= (-\lambda(t, X_t; \theta^u))V_t + \nabla E(t, X_t; \theta^u))dt + \sqrt{2\Gamma(t, X_t; \theta^u)}dW_t, \\
    dX_t &= (V_t + \mu(t, X_t; \theta^u))dt.
\end{align*}

(4.4.1)

Here \(W_t\) is a two-dimensional Brownian motion, and the state variables \((X_t, V_t)\) represent the position and velocity of a single particle at time \(t\). The \(\lambda(t, X_t; \theta^u)\) term determines the dissipation rate of the particle’s momentum away from the mean flow, \(E(t, X_t; \theta^u)\) is a particle interaction potential, \(\Gamma(t, X_t; \theta^u)\) is the volatility of the velocity, and \(\mu(t, X_t; \theta^u)\) is the overall advection field.

The Langevin dynamics (4.4.1) are capable of modelling a wide class of turbulent behaviour, and are often applied to dispersion modelling in industrial settings [79, 80].

Assuming some initial distribution \(u(0, x, v; \theta^u)\) over the position-velocity field is given, the density of an infinite number of particles following the Langevin dynamics (in position-velocity space) is provided by the solution to the Fokker–Planck equation

\begin{align*}
    \frac{\partial u}{\partial t}(t, x, v; \theta^u) &= \nabla_v \cdot \left[ (\lambda(t, x; \theta^u)v - \nabla_x E(t, x; \theta^u))u(t, x, v; \theta^u) \right] \\
    &\quad - \nabla_x \cdot \left[ (v + \mu(t, x; \theta^u))u(t, x, v; \theta^u) \right] + \nabla_v \cdot \left[ \nabla_v (\Gamma(t, x; \theta^u))u(t, x, v; \theta^u) \right].
\end{align*}

(4.4.2)

Unfortunately, this equation is difficult to work with directly. This is primarily because, in addition to time, it has two velocity variables and two position variables, meaning it is a five-dimensional PDE. As a result, mesh-based solvers such as finite differences cannot be applied accurately due to computational constraints. In addition to this, the quantity of interest is the pollution concentration over the domain, so even if we could compute a solution to (4.4.2) we would still requires the marginalisation of the velocity variables from this

\[ u(t, x; \theta^u) = \int_{\mathbb{R}^2} u(t, x, v; \theta^u) dv. \]

(4.4.3)

This means an additional two-dimensional integral must be computed at every space-time coordinate where the concentration is desired. Consequently, a Monte Carlo approach to solving (4.4.2) is typically preferred, that consists of simulating a large number of particles using time stepping methods such as Euler–Maruyama [87], and analysing their empirical density. This solution approach requires significant computational resources to be performed accurately and is difficult to assimilate data into.
A simpler SDE for particle movement that is related to the Langevin dynamics is the Brownian dynamics. The Brownian dynamics can be derived from the Langevin dynamics in the so called ‘diffusion limit’, which is reached as particle mass converges to zero \[132\]. To show this, we investigate the scaled limit of infinite dissipation, interaction potential, and velocity volatility. Defining the scaling relationship

\[
\lambda(t, x; \theta^u) = \hat{\lambda} A(t, x; \theta^u),
\]

where \( \hat{\lambda} \in \mathbb{R} \), the Langevin dynamics \[(4.4.1)\] can be rewritten

\[
\frac{1}{\hat{\lambda}} dV_t = \left( -A(t, X_t; \theta^u) V_t + \frac{\nabla E(t, X_t; \theta^u)}{\lambda} \right) dt + \sqrt{\frac{2\Gamma(t, X_t; \theta^u)}{\lambda^2}} dW_t. \tag{4.4.5}
\]

Integrating both sides of this gives

\[
\frac{1}{\hat{\lambda}} (V_t - V_0) = \int_0^t \left( -A(s, X_s; \theta^u) V_s + \frac{\nabla E(s, X_s; \theta^u)}{\lambda} \right) ds + \int_0^t \sqrt{\frac{2\Gamma(s, X_s; \theta^u)}{\lambda^2}} dW_s. \tag{4.4.6}
\]

Next we let the dissipation \( \hat{\lambda} \to \infty \), volatility of the velocity \( \Gamma(t, x; \theta^u) \to \infty \), and interaction potential \( \nabla E(t, x; \theta^u) \to \infty \), with scaling such that

\[
\frac{\Gamma(t, x; \theta^u)}{\lambda^2} \to D(t, x; \theta^u) A^2(t, x; \theta^u),
\]

\[
\frac{A \nabla E(t, x; \theta^u)}{\lambda} \to \nabla D(t, x; \theta^u) A(t, x; \theta^u). \tag{4.4.7}
\]

Then in the limit we have

\[
\int_0^t A(s, X_s; \theta^u)V_s ds = \int_0^t A(s, X_s; \theta^u) \nabla D(s, X_s; \theta^u) ds + \int_0^t A(s, X_s; \theta^u) \sqrt{2D(s, X_s; \theta^u)} dW_s. \tag{4.4.8}
\]

Substituting \[(4.4.1)\] into \[(4.4.8)\] gives

\[
\int_0^t A(s, X_s; \theta^u) dX_s = \int_0^t A(s, X_s; \theta^u) (\mu(s, X_s; \theta^u) + \nabla D(s, X_s; \theta^u)) ds + \int_0^t A(s, X_s; \theta^u) \sqrt{2D(s, X_s; \theta^u)} dW_s. \tag{4.4.9}
\]

178
Differentiating with respect to time reduces this to the SDE

$$dX_t = \left(\mu(t, X_t; \theta^u) + \nabla D(t, X_t; \theta^u)\right)dt + \sqrt{2D(t, X_t; \theta^u)}dW_t. \quad (4.4.10)$$

Equation (4.4.10) is known as the Brownian dynamics. Physically, $\lambda(t, x; \theta^u), \nabla E(t, x; \theta^u)$, and $\Gamma(t, x; \theta^u)$ are deterministic and random components of the acceleration. Therefore according to Newton’s second law, the assumption that these terms tend to infinity is equivalent to the particles becoming massless (assuming that the maximum force that can be applied to the particles is bounded). The consequence of this simplification is that particles cannot gain momentum and consequently paths that involve large deviations from the mean flow become more unlikely under Brownian dynamics. The dominant behaviour of the two approaches is similar however, making Brownian dynamics a reasonable simplification.

The Fokker–Planck equation for the distribution of particles following the Brownian dynamics is given by

$$\frac{\partial u}{\partial t}(t, x; \theta^u) = -\nabla \cdot (\mu(t, x; \theta^u)u(t, x; \theta^u)) + \nabla \cdot (D(t, x; \theta^u)\nabla u(t, x; \theta^u)). \quad (4.4.11)$$

This equation is more commonly known as the advection-diffusion equation, and is now three-dimensional, as velocity is not modelled. Again, $\mu(t, X_t; \theta^u)$ is the overall advection field, and the diffusion rate $D(t, x; \theta^u) > 0$ is the rate at which the pollution spreads away from areas of high concentration. To make this model more realistic in our setting, we must also include a source term $F(t, x; \theta^u)$ to represent the production of pollutants within the domain, and a decay term $-\alpha(t, x; \theta^u)u(t, x; \theta^u)$ to represent the proportional removal of pollutants due to factors such as gravitational settling of particles. This results in the final form of our advection-diffusion model becoming

$$\frac{\partial u}{\partial t}(t, x; \theta^u) = -\nabla \cdot (\mu(t, x; \theta^u)u(t, x; \theta^u)) + \nabla \cdot (D(t, x; \theta^u)\nabla u(t, x; \theta^u)) + F(t, x; \theta^u) - \alpha(t, x; \theta^u)u(t, x; \theta^u). \quad (4.4.12)$$

Variants of (4.4.12) are commonly applied to model a variety of scenarios which show advective and diffusive properties, such as heat transfer [25], coastal engineering [84] and fluid dynamics [142]. For computational tractability we fix the domain $(t, x) \in [0, T] \times \Omega$, where $T > 0$ is a finite time horizon and $\Omega \subset \mathbb{R}^2$ is the spatial domain. We also will assume an initial pollution concentration of

$$u(0, x; \theta^u) = g(x; \theta^u), \quad x \in \Omega, \quad (4.4.13)$$
and reflective (zero-flux) boundary conditions

$$\hat{n} \cdot (-\mu(t, x; \theta^u)u(t, x; \theta^u) + D(t, x; \theta^u)\nabla u(t, x; \theta^u)) = 0, \quad x \in \partial\Omega,$$

(4.4.14)

for $t \in [0, T]$, where $\hat{n}$ is the unit normal vector to the boundary. We will assume $g(x; \theta^u) > 0$, $F(t, x; \theta^u) > 0$, and $\alpha(t, x; \theta^u) > 0$ so that this PDE has finite positive solutions. The inputs $\theta^u \in \Theta$ represent parameterisations of the drift, diffusion, source, decay, and initial condition, and thus the solution $u(t, x; \theta^u)$ depends on their value.

The parametric solution to (4.4.12, 4.4.13, 4.4.14) is the map $u : \mathbb{R}^3 \times \Theta \to \mathbb{R}$ satisfying the equation for all $\theta^u \in \Theta$.

By integrating (4.4.12) and using the divergence theorem we see that the rate at which the total volume of pollutants changes is

$$\frac{\partial}{\partial t} \int_{\Omega} u(t, x; \theta^u) dx = \int_{\Omega} \nabla \cdot (-\mu(t, x; \theta^u)u(t, x; \theta^u) + D(t, x; \theta^u)\nabla u(t, x; \theta^u)) dx$$

$$+ \int_{\partial\Omega} (F(t, x; \theta^u) - \alpha(t, x; \theta^u)u(t, x; \theta^u)) \cdot \hat{n} dx$$

$$= \int_{\partial\Omega} (F(t, x; \theta^u) - \alpha(t, x; \theta^u)u(t, x; \theta^u)) \cdot \hat{n} dx$$

$$+ \int_{\Omega} (F(t, x; \theta^u) - \alpha(t, x; \theta^u)u(t, x; \theta^u)) dx$$

$$= \int_{\Omega} (F(t, x; \theta^u) - \alpha(t, x; \theta^u)u(t, x; \theta^u)) dx.$$

(4.4.15)

Hence the amplitude of the source term $F(t, x; \theta^u)$ determines how quickly the volume of pollutants in the system grows, and $\alpha(t, x; \theta^u)$ dictates the rate at which this decays proportionally to the current pollution level. Mass is clearly conserved if these terms are zero. This information will be used in Section 4.7 to motivate a mass controlled neural network approach capable of solving (4.4.12, 4.4.13, 4.4.14). However first we describe a simpler deep learning algorithm to solving the parametric PDE, which will subsequently be applied within the physics-informed statistical model in Section 4.5.

### 4.4.1 Solving the physical model with deep learning

In order to work with the physical model efficiently we propose to approximate the solution to the parametric PDE (4.4.12, 4.4.13, 4.4.14) by a neural network. The method used to train this neural network is a parametric extension to the Deep Galerkin method [144], and is outlined in [38]. Once trained, this network allows us to query the
solution to the PDE and its gradients with respect to the PDE parameters, at a cost that is far lower than that required by implementing more traditional mesh-based numerical solvers. To train this approximation we define a neural network approximation to the solution $\hat{u}(t, x; \theta^u)$. Letting $L$ denote the differential operator in (4.4.12), so that

$$L u(t, x; \theta^u) = \frac{\partial u}{\partial t}(t, x; \theta^u) + \nabla \cdot (\mu(t, x; \theta^u)u(t, x; \theta^u)) - \nabla \cdot (D(t, x; \theta^u)\nabla u(t, x; \theta^u)) + \alpha(t, x; \theta^u)u(t, x; \theta^u),$$

we approximate the solution to the PDE by minimising the PDE residuals in the $L^2$-norm in the interior and boundary domains, leading to the loss function

$$\| L \hat{u}(t, x; \theta^u) - F(t, x; \theta^u) \|_{L^2(\pi^\Omega, \pi^\Theta)}^2 + \| u(0, x; \theta^u) - g(x; \theta^u) \|_{L^2(\pi^0, \pi^\Theta)}^2 + \| -\mu(t, x; \theta^u)\hat{u}(t, x; \theta^u) + D(t, x; \theta^u)\nabla \hat{u}(t, x; \theta^u) \|_{L^2(\pi^\Omega, \pi^\Theta)}^2.$$
advection-diffusion model. The uncertainty in this mean therefore reflects the uncertainty of the physical parameters of the advection-diffusion model, which is derived by conditioning on the data. The uncertainty in the covariance structure arises through its hyperparameters as described in Section 4.3.1. Since the mean of this GP prior is governed by the advection-diffusion dynamics, the large scale physical behaviour of pollution movement is accounted for prior to the GP posterior itself being computed. By using the inferred GP prior mean and covariance, and then computing the corresponding GP posterior given data, we additionally capture the features in the data that are unresolved by the PDE. This is achieved whilst achieving a higher level of certainty than a typical zero mean prior would provide as in Figures 4-6–4-7. Moreover, since the unknown PDE terms and covariance hyperparameters are approximately marginalised using MCMC, we return a posterior predictive field which is also marginalised over these variables. Thus our output provides a full quantification of the physical and statistical uncertainty, complete with marginalisation over the posterior of the physical model and GP covariance structure. In this section we detail how this approach can be tractably implemented by using deep surrogate models to efficiently resolve the physical features of this model. Following this we present the results achieved by applying the method to simulated data and the data from Ulaanbaatar.

4.5.1 Modelling assumptions and methodology

4.5.1.1 Statistical model

The physics-informed statistical model can be stated as

\[ y_i = \hat{u}(t_i, x_i; \theta^u) + \eta(t_i, x_i; \theta^\eta) + \epsilon_i. \]  

(4.5.1)

Here \( \theta^u \) is a parameterisation of the advection-diffusion equation, which determines the initial condition \( g(x, \theta^u) \), drift function \( \mu(t, x; \theta^u) \), diffusion coefficient \( D(t, x; \theta^u) \), source term \( F(t, x; \theta^u) \), and decay rate \( \alpha(t, x; \theta^u) \). Given these parameters \( \hat{u}(t, x; \theta^u) \) is the corresponding neural network approximation of the PDE solution for any \( t \in [0, T] \) and \( x \in \Omega \). The second term \( \eta(t_i, x_i; \theta^\eta) \) is a spatio-temporal Gaussian process with mean zero and separable autoregressive Matérn covariance kernel

\[ K([t, x], [t', x']); \theta^\eta) = K_t(t, t') K_x(x, x'). \]

As before the hyperparameters \( \theta^\eta \) of this kernel determine its correlation structure. The last term \( \epsilon_i \sim \mathcal{N}(0, \sigma^2_{\epsilon}) \) are i.i.d. normal random variables of unknown variance \( \sigma^2_{\epsilon} \).
Let \( \theta = [\theta^u, \theta^v, \sigma^2] \), then given \( \theta \) and pollution concentration data \( Y \), with corresponding data for the time of day \( t \) and location \( X \), the model (4.5.1) specifies a multivariate normal likelihood over the data

\[
Y \sim MVN \left( \hat{u}(t; X, \theta^u), K([t, X], [t, X] | \theta^v) + \sigma^2 I_M \right),
\]

where \( K([t, X], [t, X] | \theta^v) \in \mathbb{R}^{TM \times TM} \) denotes the covariance matrix of the GP evaluated at the data locations, that is \( K([t, X], [t, X] | \theta^v)_{i,j} = K([t_i, x_i], [t_j, x_j] | \theta^v) = K_{t}(t_i, t_j)K_{x}(x_i, x_j) \). For the covariance hyperparameters we apply the same gamma priors to \( \sigma^2, \rho_x, \sigma^2_\epsilon \), and the same beta prior to \( \rho_t \) as we applied in Section 4.3.2.

Given the posterior distribution \( p(\theta | X, Y) \), the predictive posterior distribution is

\[
p(Y' | X, Y, X', \theta) = \int p(Y' | X, Y, X', \theta) p(\theta | X, Y) d\theta.
\]

Here \( p(Y' | X, Y, X', \theta) \) is the GP posterior given \( \theta \), that is \( MVN(\mu', \Sigma') \) with

\[
\mu' = \hat{u}(t', X' ; \theta^v) + K([t', X'] | \theta^v)[K([t, X], [t, X] | \theta^v) + \sigma^2 I_M]^{-1}(Y - \hat{u}(t, X ; \theta^v))^	op,
\]

\[
\Sigma' = K([t', X'] | \theta^v) - K([t', X'] | \theta^v)[K([t, X], [t, X] | \theta^v) + \sigma^2 I_M]^{-1}K([t, X], [t', X'] | \theta^v).
\]

The posterior field (4.5.3) after marginalisation of the physical model and GP hyperparameter is approximated by Monte Carlo integration in the same manner as described for the stationary GP model in (4.3.7), resulting in a Gaussian mixture model from which the mean and variance are approximated.

### 4.5.1.2 PDE assumptions

Our aim with the PDE model is to capture the large scale movements of pollution, therefore in order to reduce the dimensionality of the problem and prevent overfitting we make some simplifying assumptions in the advection-diffusion equation (4.4.12). Firstly we assume that \( \alpha(t, x ; \theta^u) = \alpha \), \( D(t, x ; \theta^v) = D \), \( \mu(t, x ; \theta^u) = \mu \) are all constant in space and time. These parameters are treated as unknown and their values are sampled during the MCMC. Their prior distributions are \( \alpha \sim N(0, 0.2) \), \( D \sim \Gamma(4,3) \), and \( \mu \sim MVN(0, 0.2I_2) \).

183
We represent the initial pollution concentration by a radial basis approximation

\[ g(x; \theta^u) = \sum_{n=1}^{N_0} \theta^0_n \varphi(\| x - x^n \|). \]  

(4.5.5)

Here \([x^1, \ldots, x^{N_0}]\) are the centres of each radial basis function, which are arranged in a regular equispaced grid. The coefficients \(\theta^0_n\), which determine the local amplitude of the initial pollution field, are treated as unknowns that are inferred. These coefficients are defined by \(\theta^0_n = 15(\tanh(\gamma^0) + 1)\), where \(\gamma^0\) has multivariate normal prior with mean vector -1 and covariance matrix \(K_x(x^i, x^j)\) (where \(K_x(\cdot, \cdot)\) is the Matérn covariance with fixed hyperparameters \(\nu = 2.5, \sigma = 1, \rho_x = 0.3\)). This choice of prior ensures that the initial pollution field is positive, and the spatial covariance ensures it varies smoothly over the city. Behaviourally, this prior favours a low initial pollution field, however it provides enough freedom for the coefficients in this field to increase to a maximum of 30 where required by the data to achieve a good fit.

In Figures 4-4–4-5 we observed that pollution levels rose twice per day, therefore for the source \(F(t, x; \theta^u)\) we use this information to parameterise two source fields in a similar way to the initial condition

\[ F_1(x; \theta^u) = \sum_{n=1}^{N_F} \theta_{F1}^n \varphi(\| x - x^n \|), \]  

(4.5.6)

\[ F_2(x; \theta^u) = \sum_{n=1}^{N_F} \theta_{F2}^n \varphi(\| x - x^n \|). \]  

(4.5.7)

The overall parameterisation of the source in space and time is then

\[ F(t, x; \theta^u) = \phi(\| t - \theta_{1t}^1 \|; \theta_{1en}^1) F_1(x; \theta^u) + \phi(\| t - \theta_{2t}^1 \|; \theta_{2en}^1) F_2(x; \theta^u). \]  

(4.5.8)

Here (4.5.6) and (4.5.7) represent the spatial distribution of the pollution sources. The priors of their parameters \(\theta_{F1}\) and \(\theta_{F2}\) are the same tanh transformed multivariate normal prior used for the initial condition \(\theta^0\). The temporal functions \(\phi(\| t - \theta_{it}^1 \|; \theta_{iten}^1)\) are again radial basis functions centred at \(\theta_{it}^1\), with width defined by \(\theta_{iten}^1\). The timings of the two sources are therefore determined by \(\theta_{it}^1\), and the duration that each of these sources emits pollution by \(\theta_{iten}^1\). The full source function \(F(t, x; \theta^u)\), can therefore capture the source distribution of the two peaks shown in the data, and estimate the times at which they occur and for how long. For the study on simulated data, Gamma prior distributions \(\theta_{1t}^1 \sim \Gamma(2, 2)\) and \(\theta_{2t}^1 \sim \Gamma(12, 2)\) are applied, and for the real data these
are \( \theta_1 \sim \Gamma(5,1) \) and \( \theta_2 \sim \Gamma(20,1) \). These priors have separated means 1 and 6 in the simulation study, and 5 and 20 for the real data, to encourage pollution production in the physical model to occur around the two distinct times observed in the data. The parameters \( \theta_{1\text{len}}, \theta_{2\text{len}} \) governing how long pollution is produced during each peak have \( \Gamma(3,1) \) priors. For the implementations presented in this work we use Gaussian PDFs for the radial basis functions \( \varphi(\|x-c_x\|) \) and \( \phi(\|t-c_t\|; s_t^2) \), where \( c_x, c_t \) are the centres of the corresponding Gaussian distributions, and \( s_t^2 \) are their variances.

Given the parameterisations described, the full set of unknowns in the PDE that we infer is \( \theta_u = [D, \alpha, \mu, \theta^0, \theta^{F_1}, \theta^{F_2}, \theta_1, \theta_2, \theta_{1\text{len}}, \theta_{2\text{len}}] \). We train the deep surrogate model \( \hat{u}(t,x; \theta_u) \) to approximate a solution to the advection-diffusion equation (4.4.12) over this set of parameters by minimising the loss function (4.4.17) with \( \pi^\Omega, \pi^\partial^\Omega, \pi^\Theta \) set to be uniform distributions. This surrogate allows us to efficiently approximate and differentiate the PDE solution for any combination of these parameter values. We utilise this efficiency by substituting the surrogate into the HMC sampler to sample from the joint posterior distribution of \( \theta = [\theta_u, \theta^\eta, \sigma^2] \) as required to fit (4.5.1).

### 4.5.2 Simulated data

We begin by demonstrating this approach using simulated data, which we construct to be similar in structure to the measured data from Ulaanbaatar. We will work in the domain \( x = \Omega = [-1,1] \times [-1,1], t \in [0,7] \). To generate synthetic data we fix a set of parameters \( \theta = [\theta_u, \theta^\eta, \sigma^2] \) and sample from the model (4.5.1). More specifically we solve the advection-diffusion equation for \( \theta_u \), and add this solution to a sample from the Gaussian process \( \eta(t,x; \theta^\eta) \) at times \( t \in \{0,1,\ldots,7\} \). This combined field is then subsampled at 10 random spatial locations, and 10 random instances of i.i.d. Gaussian noise is added to each these samples, resulting in 10 observations at each space-time location. The result of applying our method to this data is visualised in Figure (4-8).
Figure 4-8: Simulated data spatio-temporal predictions. Left: Posterior mean of \( \hat{u}(t, x; \theta^u) \). Centre: Posterior mean of \( \eta(t, x; \theta^\eta) \). Right: Posterior hybrid pollution prediction (in \( \mu g/m^3 \)).
The rows of this figure correspond to increasing time points from $t = 0$ in row 1, to $t = 7$ in row 8. The fields plotted in the first column of Figure 4-8 shows the mean of the posterior PDE prediction, the last column is mean of the GP posterior after marginalisation of the PDE and covariance parameters. The central column can be considered the GP part of the hybrid model, in the sense of being the mean obtained after subtracting the corresponding PDE solution from each GP posterior sample (thus this column is the posterior mean of $\eta(t, x; \theta^\eta)$ in (4.5.1)). The central column therefore amounts to the mean of a GP based correction to the PDE part which is marginalised with respect to all PDE parameters and GP hyperparameters. The coloured circles in the first and last columns represent the locations and pollution concentration of the data in the same way as Figure 4-6.

We see that the PDE part of this solution is able to capture the large scale features of this data, including a good agreement with the initial pollution concentration and the successful identification of sources in the northeast and later in the southeast. It is evident on close inspection that the PDE part is unable to resolve all of the behaviour however, and this can be seen most clearly in the bottom row where the highest measurement is underestimated by the PDE. In the right hand column we observe that the hybrid model is just a small perturbation away from the PDE based model, therefore this prediction is also mostly physics governed. However the additional flexibility endowed upon the hybrid model by the GP has ensured that the hard-to-resolve features are now accounted for, and we attain a good agreement with the data at all spatio-temporal locations. For the final time point $t = 7$, the uncertainties associated with this model are shown in Figure 4-9.

Figure 4-9: 95% credible interval widths associated with simulated data study ($\mu g/m^3$). Left: Hybrid model uncertainty. Centre: Physical model uncertainty. Right: Provided for comparison, the uncertainty of a GP spatial post-processing approach.
The first plot shows the overall uncertainty associated with this estimate by applying the spatio-temporal analogue to the variance formula (4.3.8), where in this case the PDE parameters are treated as additional hyperparameters. This can be related to the second image, which shows the uncertainty of the PDE, specifically the 95% credible interval width of the posterior over the PDE solution. It is notable that while this physical uncertainty does not contribute to the overall uncertainty additively (the PDEs are the prior means, thus this figure shows the uncertainty over the mean of the GP prior), its effects are clearly reflected in the uncertainty of the hybrid model. In fact, in the hybrid model we see that the uncertainty behaves similarly to a pure Gaussian process approach close to the data where information is more abundant and the likelihood dominates the model, however in further away regions the posterior reverts back to the prior, which is dominated by the uncertainty over the physical model.

As a comparison, the final figure is the uncertainty derived solely from a spatial output post-processing approach as in \cite{51}. This is the uncertainty derived by subtracting a deterministic physical prediction (in this case the posterior mean PDE) from the data and applying GP regression to the residual data. In comparison to the full hybrid model we see that the physical uncertainty observed in the PDE posterior is no longer evident in this case. Furthermore, the residual data to which this GP model was fitted has smaller amplitude, thus a smaller posterior variance is inferred from this model. As a result, this model is overly certain further away from the data when compared with our hybrid approach.

In cases where a physical model prediction is provided with a measure of its uncertainty, this spatial post-processing approach can be improved in order to reflect this uncertainty by simply treating the spatio-temporal Gaussian process and physical process as independent random variables (as done in \cite{159} for example). This is a conservative approach, as it results in the overall uncertainty being derived from a direct summation of the variances of the physical and GP parts. In our setting for example, this would be equivalent to stating that the overall variance is the sum of the variances corresponding to the middle and right hand panels in Figure (4-9). On the other hand the joint modelling in our approach considers their correlations, which allows our approach to automatically detect that we can be confident of our estimates in areas with significant data irrespective of physical uncertainty, while still utilising this physical uncertainty in a statistically consistent way in areas where data is sparse.
Finally, ensemble methods such as [12] can achieve simultaneous consideration of physical and spatial-temporal statistical uncertainty through Bayesian model averaging of the ensemble combined with spatial GP post-processing. In this setting, the marginalisation of physical models is achieved through a weighted average over the ensemble members. These approaches are therefore restricted by the available ensemble, whereas the deep learning approximation underlying our approach allows us to fully marginalise the entire class of physical models that we are working with.

4.5.3 Ulaanbaatar data

We now apply the method to real pollution data. Since the pollution levels were shown to be more severe in colder weather we will focus on data from November 2018. The mean daily average temperature for this month was $-8.9^\circ$C, and 80% of the days in this month had an average temperature of less than $-5^\circ$C. Given that there are 30 days in November, the information available to us at each monitoring station therefore includes 30 observations of the pollution concentration for each hour of the day. Before inference, the data is standardised identically to the example in Section 4.3.2. The results are visualised in Figure 4-10 in the same format as Figure 4-8. The rows of this figure represent 3 hour increments throughout the day, with row 1 showing the fields at 1:00am, row 2 showing 4:00am, and so on until row 8 which corresponds to 10:00pm. The full 24 hour hybrid model predictions in the same format as Figure 4-6 can additionally be found in Figure 4-12 of the appendix. In Figure 4-10 we see similar behaviour to our simulation study, where the PDE model in the left hand column successfully captures the general structure of the pollution, but is unable to resolve the finer details required to achieve a full agreement with the data. In the right hand column we see that our hybrid model is able to correct for these difference and achieve a close agreement with the data.
Figure 4-10: Real data spatio-temporal predictions. Left: Posterior mean of $\hat{u}(t, x; \theta^u)$. Centre: Posterior mean of $\eta(t, x; \theta^\eta)$. Right: Posterior hybrid pollution prediction (in $\mu g/m^3$).
In order to provide a comparison across temperatures, Figure 4-13 of the appendix demonstrates the results obtained by applying this model to data taken from slightly warmer days (with daily average temperature between $-5^\circ$C and $0^\circ$C). These results show behaviourally similar results to those shown in Figure 4-10 however with lower amplitude which reflects the negative correlation between temperature and pollution. An interesting feature of these predictions is the presence of a large pollution source originating in the north. Despite the lack of local data in this region, the location of this inferred source coincides with a dense population of ger dwellings [5]. This adds credibility to our inference, as such dwellings are commonly considered large drivers of pollution concentrations. In comparison to the GP-only predictions shown in Figure 4-6 it is clear that unphysical behaviour is largely avoided, as the hybrid prediction is clearly governed mostly by the PDE model, and therefore physical assumptions. The uncertainties attached to this inference when the pollution peaks at $t = 19$ are shown below.

Figure 4-11: 95% credible interval uncertainties associated with the real data ($\mu g/m^3$).

These figures are analogous to Figure 4-11 for this dataset, and we observe similar features here as we did in the simulation study. In comparison to the GP-only approach shown in Figure 4-7 we see a drastic reduction in overall uncertainty away from the data, particularly in areas where the physical model is confident. This physically principled quantification of the uncertainty allows us to be more confident of our predictions over a much larger region than the GP-only prediction.
4.6 Conclusions

In this work we presented an approach where the deep surrogate methodology developed in [38] is applied to an air pollution prediction problem. By including a neural network approximation to the parametric advection-diffusion PDE we were able to tractably incorporate appropriate physical dynamics into a Bayesian hierarchical spatio-temporal Gaussian process model. Using both synthetic data and real air pollution data from Ulaanbaatar, Mongolia, we jointly inferred advection-diffusion parameters and GP hyperparameters using a custom Hamiltonian Monte Carlo sampler, and used this to produce a posterior pollution field that composes physical and data-driven behaviour.

A significant contribution of our approach is that it ensures that sources of uncertainty derived from both the PDE and Gaussian process, and their dependence on one another is captured through the computation of a joint posterior distribution over the physical model and Gaussian process parameters. This contrasts with the standard statistical approaches of evaluating physical models in advance, then combining these outputs with statistical post-processing terms to capture unresolved behaviour [51, 140]. Additionally, as the data presented in this paper is sparsely distributed, we demonstrated that purely data-driven approaches to understanding it will give highly uncertain results in comparison to our approach. Thus, using our physics-informed approach, we are able to obtain greater understanding of the spatio-temporal behaviour of pollution in Ulaanbaatar.

The full Bayesian analysis which we implement here utilises a bespoke Hamiltonian Monte Carlo sampler which leverages the fast evaluation and differentiation of the deep surrogate model in order to sample from the posterior distribution. We note that while the specific implementation of the deep surrogate methodology was not the major focus of this work, its inclusion is a fundamental ingredient without which this analysis would not be possible. In fact, the use of deep learning surrogate methodology invokes new possibilities for research in many fields where physical model evaluation was previously a bottleneck. This work provides an example beyond the typical domains of forward and inverse uncertainty quantification by applying it to a spatio-temporal statistical setting, and we are confident that in future such approaches will find applications in a much wider range of applied settings.
4.7 Future research opportunities

We now present two possible extensions to the approach outlined in this work. In Section 4.7.1 we describe an approach to guarantee mass conservation within the PDE solver by using a neural network to construct a flexible function approximator that has controllable integral over a closed domain. This mass controlled neural network can then be trained to solve the advection-diffusion equation similarly to a regular neural network. Preliminary testing with this approach has shown promising results in terms of PDE solution accuracy and speed of convergence. A second avenue for future research that we discuss in Section 4.7.2 involves incorporating more complex statistical and physical modelling assumptions into our framework. In particular, we discuss an extension to our implementation that would allow us to model the impact of temperature on the prediction by influencing the rate of pollution production in the physical model.

4.7.1 Mass controlled neural network

The PDE models involved in this work obey known mass conservation properties. As an extension to the deep learning solution approach described in Section 4.4.1 we propose a mass-controlled neural network that can increase the accuracy of approximation to equations such as (4.4.12) where the definite integral of the solution over the domain is known a-priori. This is a novel method of composing a neural network approximation, which allows us to precisely control the integral of the approximation over the spatial domain, and thus these approximations do not have to learn factors such as mass conservation from the equation. To maintain tractability, we seek an approximation to (4.4.12, 4.4.13, 4.4.14) on the compact domain $[0, T] \times \Omega$, where $\Omega$ is a rectangular domain which covers the city. First consider the case where there is no source or decay, so (4.4.12) is a mass conserving system. The solution can be written as

$$u(t, x; \theta^u) = u(0, x; \theta^u) + \nabla_x \cdot f(t, x; \theta^u), \quad x \in \Omega, t \in [0, T], \quad (4.7.1)$$
for some unknown field \( f(t, x; \theta^u) \) that depends on the PDE parameters \( \theta^u \). Using the divergence theorem, the integral of the solution is therefore

\[
\int_\Omega u(t, x; \theta^u) dx = \int_\Omega u(0, x; \theta^u) dx + \int_\Omega \nabla \cdot f(t, x; \theta^u) dx \\
= \int_\Omega u(0, x; \theta^u) dx + \int_{\partial\Omega} f(t, x; \theta^u) \cdot \hat{n} dx.
\] (4.7.2)

If \( f(t, x; \theta^u) = 0 \) for \( x \in \partial\Omega \), then the final term in (4.7.2) is zero and \( u(t, x; \theta^u) \) has the same integral as \( u(0, x; \theta^u) \). To translate this into a deep learning algorithm, we decompose the unknown function \( f(t, x; \theta^u) \) into two factors and write the deep learning approximation as

\[
\hat{u}(t, x; \theta^u) = u(0, x; \theta^u) + \nabla \cdot (p(t, x; \theta^u)q(x)), \quad x \in \Omega, t \in [0, T],
\] (4.7.3)

Here \( p(t, x; \theta^u) \) is a neural network with a two-dimensional output, and \( q(x) \) is a function with \( q(x) > 0 \) for \( x \in \Omega \), and \( q(x) = 0 \) for \( x \in \partial\Omega \). Since (4.7.3) is guaranteed to conserve mass, we can use this to learn a mass conserving approximation to the parameteric PDE solution of (4.4.12) by substituting (4.7.3) into the loss function (4.4.17). Preliminary experimentation with this idea, where \( f(t, x) \) is a fully connected neural network, shows a reduction in the loss function (4.4.17) by a factor of 30 in comparison to approaches achieved by approximating \( u(t, x; \theta^u) \) by a fully connected neural network directly.

In the case of non-zero source terms \( F(t, x; \theta^u) \) we can integrate (4.4.15) with respect to time to show that the amount of mass in the system is

\[
U(t; \theta^u) = \int_\Omega u(t, x; \theta^u) dx = \int_\Omega u(0, x; \theta^u) dx + \int_0^t \int_\Omega F(s, x; \theta^u) dx ds,
\] (4.7.4)

so the neural network approximation can be defined as

\[
\hat{u}(t, x; \theta^u) = M(t, x; \theta^u) + \nabla \cdot (p(t, x; \theta^u)q(x)),
\] (4.7.5)

for any function \( M(t, x; \theta^u) \) with \( \int_\Omega M(t, x; \theta^u) dx = U(t; \theta^u) \). For example \( M(t, x; \theta^u) \) could simply be the constant in space \( U(t; \theta^u) / \text{Area}(\Omega) \), or we could set

\[
M(t, x; \theta^u) = u(0, x; \theta^u) + \int_0^t F(s, x; \theta^u) ds.
\] (4.7.6)
Most importantly, computing $M(t, x; \theta^u)$ does not require us to solve the PDE, and for many choices of $F(s, x; \theta^u)$ the spatial and temporal integrals required to evaluate $M(t, x; \theta^u)$ are readily available (as in our case where $F(t, x; \theta^u)$ is comprised of Gaussian radial basis functions with known cumulative densities). Therefore $M(t, x; \theta^u)$, and consequently this neural network approximation, can be specified directly.

If, in addition to $F(t, x; \theta^u) \neq 0$, the decay rate $\alpha(t, x, \theta^u)$ is also non-zero we have

$$U(t; \theta^u) = \int_{\Omega} u(0, x) dx + \int_0^t \int_{\Omega} \left( F(s, x; \theta^u) - \alpha(s, x; \theta^u) u(s, x; \theta^u) \right) dx ds. \quad (4.7.7)$$

The mass in the system therefore satisfies

$$\frac{dU}{dt}(t; \theta^u) = \int_{\Omega} F(t, x; \theta^u) dx - \int_{\Omega} \alpha(t, x; \theta^u) u(t, x; \theta^u) dx,$$

$$U(0; \theta^u) = \int_{\Omega} u(0, x; \theta^u) dx.$$

Now if we again assume $\alpha(t, x, \theta^u) = \alpha$ is constant, then this becomes the first order ODE

$$\frac{dU}{dt}(t; \theta^u) = \int_{\Omega} F(t, x; \theta^u) dx - \alpha U(t; \theta^u), \quad (4.7.8)$$

which has the solution

$$U(t; \theta^u) = e^{-\alpha t} \left( \int_0^t \int_{\Omega} e^{\alpha s} F(s, x; \theta^u) dx ds + U(0; \theta^u) \right). \quad (4.7.9)$$

So the corresponding mass controlled neural network approximation is $M(t, x; \theta^u)$, where we can, for example, set

$$M(t, x; \theta^u) = e^{-\alpha t} \left( u(0, x; \theta^u) + \int_0^t e^{\alpha s} F(s, x; \theta^u) ds \right). \quad (4.7.10)$$

Here we have again that $M(t, x; \theta^u)$ requires only integrals of the initial conditions and $F(t, x; \theta^u)$, so a solution of the PDE is not needed. And using the radial basis approach described $M(t, x; \theta^u)$ can be specified directly. This neural network approximation can then be trained to solve the parametric PDE by substituting it into the deep surrogate loss function. The resulting approximations are systematically constrained to integrate to the same quantity as the true solution to the problem.
4.7.2 More complex modelling assumptions

There is scope to develop both the physical and statistical modelling assumptions in order to better understand the factors that are driving the evolution of pollution in UB.

A key point of interest which was touched upon during the exploratory data analysis in Section 4.2, is learning more about the relationship between the pollution levels and meteorological conditions in UB. In particular we highlighted that temperature is negatively correlated with pollution, but other factors such as wind direction might also be valuable predictors. Focusing on temperature as an example, future work could incorporate temperature data into the statistical and physical modelling. For example we could specify the model

\[ y_i = u(t_i, x_i; c_i, \theta^u) + \eta(t_i, x_i, c_i; \theta^\eta) + \epsilon_i, \]  

(4.7.11)

where \( c_i \) is the temperature data in Celsius corresponding to \( t_i, x_i, y_i \). This information can then be assimilated into the advection-diffusion model through the source term. Figure 4-2 shows that the PM10 concentrations vary exponentially with temperature, while Figure 4-13 in the appendix suggests that the overall spatial structure of the pollution is preserved as temperature varies. Therefore a sensible parameterisation of the source would be to assume the same spatial radial basis construction and, for example, assuming the amplitude of this source has exponential scaling with temperature through the parameterised form

\[ F_i(x, c; \theta^u) = e^{\beta_0 + \beta_1 c} \sum_{n=1}^{N_F} \theta^u_n F^i_n \phi(\|x - x^n\|). \]  

(4.7.12)

The exact scale of this impact, which is determined by \( \beta_0 \) and \( \beta_1 \) could then be inferred along with the other parameters and hyperparameters.

A difficulty with this approach however is that that the Gaussian field \( \eta(t, x, c; \theta^\eta) \) must now be defined over time, space, and temperature, to allow us to represent how the data deviates from the physical model differently at different temperatures. However unlike \( t \) and \( X \), the temperature \( c \) does not have a Cartesian product structure, and so Kronecker product models with separable covariances are not applicable. This greatly increases the matrix dimensions that must be inverted, thus reduced dimension approaches to GP regression such as [154] may be required to implement this.
If changes are made to the statistical model, we can implement a model comparison procedure to compare the pointwise out-of-sample prediction accuracy. These methods include leave-one-out cross-validation (LOO) and the widely applicable information criterion (WAIC), further details of their implementation can be seen in [156]. A residual analysis could also be conducted to evaluate the goodness-of-fit for any given model.

Extensions to the physical modelling could include considering more flexible parameterisations of the advection-diffusion model. In addition to the temperature based source terms described, additions could involve models which allow variable wind velocities, spatially dependent diffusion rates, topological considerations, or a vertical dimension in the advection-diffusion PDE. Even further, one could consider adding source and decay terms to the Fokker–Planck equation of the Langevin dynamics (4.4.2) and training a deep learning surrogate similar to the one implemented in this work for the advection-diffusion model. The densities of the additional velocity variables present in the Langevin dynamics could be marginalised by from this solution using a second neural that integrates the first in a similar manner to our deep learning integral equation solution approach.
4.8 Appendix

Figure 4-12: Full 24 hour predictions from the hybrid model, corresponding to Figure 4-10.
Figure 4-13: Hourly predictions from the hybrid model using data taken from days when the daily average temperature was between $-5^\circ C$ and $0^\circ C$. 
Chapter 5

Conclusion

5.1 Summary

The overall aim of this thesis was to develop the deep surrogate methodology and explore its applications as a tool for a variety of Bayesian problems requiring the solution of expensive physical models.

There are a number of contributions in this thesis. Chapter 2 demonstrates a novel application of deep learning surrogates based on unsupervised training over parametric physical equations to Bayesian inverse problems. In addition to parametric PDEs, this chapter introduced an unsupervised deep learning solution method for parametric integral equations, which is an area that has currently received little attention. The effectiveness of both techniques for Bayesian inverse problem was then demonstrated by being used within a Markov chain Monte Carlo sampler to infer the Bayesian posterior distribution in a number of simulated examples.

Chapter 3 contributes significant extensions to the techniques described in Chapter 2. These extensions can be applied to considerably increase the efficiency and accuracy of both the surrogate training and posterior sampling, as well as improving the interpretability of the priors that can be applied. The novel contributions that allow us to achieve this include a training scheme that adaptively trains the surrogate over the desired posterior distribution, and a sampling scheme that combines the efficiency of Hamiltonian Monte Carlo using deep learning surrogates with the robustness and accuracy associated with traditional finite difference schemes. Another contribution is the application of this approach to real data gathered from a range of experimental
settings using a compressor cavity rig. We were able to perform a full Bayesian uncertainty quantification of the spatio-temporal heat fluxes for this data and outline the practical impact of this.

Finally, Chapter 4 demonstrates the novel application of deep surrogate methodology in spatio-temporal statistical settings. In this chapter we use a neural network approximation to a parameterised atmospheric dispersion PDE to model the mean function of a Bayesian hierarchical spatio-temporal Gaussian process model. In addition to the specification of this model, we contribute the algorithmic methodology that allows us to marginalise all physical parameters and Gaussian process hyperparameters, thus providing a fully Bayesian estimate of the spatio-temporal pollution field and its uncertainty. Using data from Ulaanbaatar, Mongolia, we demonstrated that this method provides more physics-driven predictions than the corresponding standard statistical approach, giving more generalisable predictions in this sparse data setting.

As with any other method, our approach has its limitations. In Chapter 3 we highlighted that the acceleration provided by the deep surrogate approach comes at the cost of the theoretical guarantees associated with more traditional solvers. The delayed-acceptance approach with finite differences described in this chapter provides an intermediary, that effectively leverages the efficiency of the surrogate while retaining these guarantees, however this is still significantly less efficient than the surrogate only approach as we showed. Furthermore, the surrogate models relied upon in Chapters 2 and 4 of this work require offline training over the parametric domain prior to inference being performed. As shown in Chapter 3 this training can be expensive, and while the adaptively trained surrogate approach proposed in this chapter overcame this issue for the problem it was applied to, it is sometimes more desirable to train a general surrogate so that it can be applied quickly to a large number of data sets without further adaptation. In these cases the large training cost in unavoidable, thus is the trade-off for the accelerated inferences provided by the surrogate once trained.

Despite these factors, this thesis has shown that the deep surrogate methodology, and the various approaches to its training and incorporation into broader algorithms, provide a promising approach to significantly accelerate problems that require the repeated evaluation of physical models. In particular, the flexibility of this approach for Bayesian problems has been illustrated by both a functional parameter inference setting where accuracy is of utmost importance, and a spatio-temporal statistical predictive model for which a fully Bayesian approach incorporating physics would otherwise be impossible
with current techniques. Despite the differing nature of these problems we showed that the deep surrogate approach could be seamlessly incorporated for substantial efficiency gain. In future it is intended that this work will be applied to larger-scale applications, for example surrogate models could be applied to accelerate expensive components of large-scale models of climate and hydrology, allowing for more efficient evaluations and assimilation of data into such models. In addition to this, we hope that the availability of physical surrogate models that can be efficiently evaluated much like any other parametric function will lead to closer collaboration between statistical and physical scientists. The investigation of this relationship, and the opportunities it presents is a further area for exploration.

5.2 Closing remarks

Deep learning has firmly established itself as one of the most promising avenues forward in our ambition to tackle problems of increasing complexity. Its impact extends into our everyday lives and the possibilities afforded by deep learning are the focal point of interest of an ever-growing population of applied and theoretical scientists. In applied mathematics there has been a huge influx of interest into this area. In fact, today the literature investigating the application of deep learning in mathematics is orders of magnitude larger than it was when the ideas for this thesis were conceived. This body of work is a part of this literature, which focuses on the impact that deep learning surrogate methodology can have on statistical problems. The results obtained within this work provide examples of how novel approaches based on deep learning can contribute value to engineers and statisticians, and the methodology underpinning these examples can be applied to benefit research in many communities.

The study of interactions between deep learning and differential/integral equations is a young field, and we are still in the very early stages of investigation. As time passes these methods will likely develop and mature, gaining theoretical grounding and robustness as they do. There is room for optimism, given the impressive results that are already being demonstrated, that the outcomes of this research effort will be transformative across the numerical sciences.
Bibliography


two-stage Markov chain Monte Carlo method for dynamic data integration. Water
Resources Research, 411, 12 2005.

803, 01 2006.


[49] E. Friedmann. PDE/ODE modeling and simulation to determine the role of

flow in a sealed rapidly rotating disc cavity. Int. J. Heat Mass Transf., 147:11886,
2020.

weather field forecasting: The geostatistical output perturbation method. Journal


[53] A. Gelman, P.D.S.A. Gelman, and J. Hill. Data Analysis Using Regression and


[56] M. B. Giles. Multilevel Monte Carlo path simulation. Operations research,

CRC Press, 2014.


[74] D. Izzo, E. Öztürk, and M. Märtens. Interplanetary transfers via deep representations of the optimal policy and/or of the value function, 07 2019.


211


[132] H. C. Rodean. *Stochastic Lagrangian Models of Turbulent Diffusion*. Meteoro-


[134] H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent
Gaussian models by using integrated nested Laplace approximations. *Journal of
the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):319–392,
2009.


Bayesian inversion for the determination of the thermal diffusivity of a material.


[141] G. Shaddick and J. V. Zidek. *Spatio-temporal methods in environmental epidemi-


