A new way to do epidemic modeling

by

Raj Abhijit Dandekar

Submitted to the Department of Civil and Environmental Engineering
and the Center for Computational Science and Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computational Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2022

Author . . . .
Department of Civil and Environmental Engineering
and the Center for Computational Science and Engineering
June 15, 2022

Certified by . . . .
Alan Edelman
Professor of Applied Mathematics
Thesis Supervisor

Certified by . . . .
Chris Rackauckas
Research Affiliate, Department of Applied Mathematics
Thesis Supervisor

Accepted by . . . .
Colette L. Heald
Professor of Civil and Environmental Engineering
Chair, Graduate Program Committee

Accepted by . . . .
Nicolas Hadjiconstantinou
Professor of Mechanical Engineering
Co-Director, Center for Computational Science and Engineering

A new way to do epidemic modeling

by

Raj Abhijit Dandekar

Submitted to the Department of Civil and Environmental Engineering
and the Center for Computational Science and Engineering
on June 15, 2022, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computational Science and Engineering

Abstract

The Coronavirus respiratory disease 2019 originating from the virus SARS-COV-2 led to a global pandemic, leading to more than 500 million confirmed global cases and approximately 6 million deaths in more than 50 countries. Since the outbreak of this pandemic, a number of modeling frameworks have been used to analyze various aspects of the pandemic such as prediction of infected and recovered case counts, hospitalizations, travel restrictions, reopening and non-pharmaceutical interventions. These frameworks can be divided broadly into the following categories: (a) compartment models which are interpretable but cannot capture complex effects and (b) agent based models which can capture varying ranges of complexity; but are generally non interpretable.

In this thesis, we introduce another category for epidemic modelling, which is rooted in Scientific Machine Learning. Scientific Machine Learning (SciML) leverages the interpretability of ODEs with the expressivity of neural networks. We thus aim to retain the interpretability of compartment models along with the complexity of agent based models using the SciML modeling paradigm. Using such a framework, we tackle a wide variety of application based problems including:

- How quarantine control policies shaped the outbreak evolution in different countries around the world.
- Effect of early reopening in the Southern and West Central US states; and how it led to an exponential explosion of infected cases in the USA during the period of June-Aug 2020.
- Virtual Virus spread through Bluetooth tokens; and how it can be used to obtain real time estimates of the pandemic.

Towards the end, we analyze the robustness of the proposed SciML methodology and provide a general set of guidelines for training such models in other domains.

Thesis Supervisor: Alan Edelman
Title: Professor of Applied Mathematics

Thesis Supervisor: Chris Rackauckas
Title: Research Affiliate, Department of Applied Mathematics

Acknowledgments

"Not all those who wander are lost!" was one of my favorite Lord of the Rings quotes when I embarked on my PhD journey. Now when I look back as I write the abstract of the thesis, I realize that I wandered a lot, but was never alone. There were a wide variety of people who constantly supported, motivated, challenged and inspired me.

Starting with my PhD advisors: Chris and Alan, who have created a beautiful culture at the Julia Lab, which allowed me to thrive and reach the end of my PhD. I believe that a new model of leadership is being developed at the Julia Lab, which sets it apart from many other labs at MIT. I call it the "constellation" based leadership where people are given tremendous freedom and responsibility to set off on their own path, make mistakes, learn and grow. Such constellation based leadership eventually made me an independent researcher, start a new field in epidemiological modeling and interact with top researchers, industries and start-ups in this field. I thank Chris and Alan for creating this culture, and set the ball rolling for many more directions in which the Julia Lab will continue to grow.

Secondly, this challenging journey would not have been possible without the support of my family members: Aai, Baba, Radhu, Rajat and the little one, my niece Ira. They have stood by me like a rock in the toughest of my moments. This support eventually allowed me to get through the difficult and challenging times.

I have made some friendships which I will cherish for the lifetime. Sreedath Panat and Arun Krishnadas are like family now, and our home at Marney St was a second home away from home. End of an era. Yash Dixit, who I've known for more than a decade now: thank you for being the best roommate I could have asked for. I cherish the countless movies we have seen together and the discussions that followed. Sungkwon Lee, I lived through the toughest times of my PhD experience with you. I still think you truly understand me. Thank you. Without close friendships, life (read PhD) can sometimes seem really hard. I was lucky to have these friendships and all others which I may have missed here.

Onwards and Upwards! I hope you enjoy reading the thesis as much as I did writing it.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and thesis outline

Traditionally, epidemiological models are divided into 2 major categories. The first category of models are standard compartmental models frequently used in the task of epidemic modeling, such as SIR (Susceptible-Infected-Removed) and SEIR (Susceptible-Exposed-Infected-Recovered) models [63, 82]. These models are typically governed by a set of ordinary differential equations which makes them interpretable. Two major assumptions of this class of models are (a) homogeneity of the population and (b) the law of mass action, which states that the rate of change of compartment population at the next time step is proportional to the compartment population at the current time step [2]. These assumptions make these models weaker than the second category of models: agent-based modeling which can simulate arbitrary magnitudes of heterogeneity and complexity (see [40] and references therein). Since agent-based models are not governed by any scientific structure such as ODEs, their results are typically not as transparent and not as interpretible when compared to compartment-based models.

In this thesis, we introduce a third category of models which are based on Scientific Machine Learning (SciML). Apart from those demonstrated in the present thesis, there have been very few studies aimed at applying SciML methods in the field of epidemiology. In this thesis, we demonstrate novel, yet fundamental models of epidemiology based on Scientific Machine Learning. These models are rooted in interpretable ordinary differential equations, which are augmented with neural network modules. In the second chapter, we introduce the Quarantine SIR model (QSIR) of epidemiology. This model starts with the traditional SIR (Susceptible-Infected-Recovered) model of epidemiology and augments it with a neural network component for the quarantine strength. We show that the QSIR model is not only more expressive than the SIR model, it is also highly interpretable. We show that the QSIR model can be used as a reliable diagnostic tool to estimate the real time quarantine strength evolution in almost all countries of the world, which were affected by Covid-19.

Subsequently, we realized that the QSIR model can also be used to model state reopening scenarios (by using quarantine strength as a proxy to the consequences of reopening). This realization helped us tackle the question: How many Covid-19 infections would have been saved had the Southern and West-Central US states not reopened early during June - Aug 2020? The answer to this question is discussed in Chapter 3.

To further test the SciML modeling paradigm, we ventured into the field of "virtual virus spread" in a collaboration with a team from University of Auckland and Cornell University. The idea to solve the lag problem in reporting of Covid-19 cases: usually it takes a week or two for an infection count to be part of the case load statistic, after the infection occurs. The question was: "can we reduce this lag to zero?". We eventually demonstrated on simulated data that virtual virus spread through Bluetooth tokens can be potentially be used to obtain real time estimates of the Covid-19 infection case count. This project was called the "Safe Blues Project". As will be seen in Chapter 4, at the heart of the Safe Blues Project is a SciML framework: ordinary differential equations augmented with neural network modules.

We soon discovered that SciML frameworks can be applied to a wide range of epidemiological applications. However, 2 questions remained: (a) what is the general procedure for training these models? , (b) how robust are these models? and (c) can we perform uncertainty quantification through these models? In Chapter 5, we delve into the methodology for training SciML models. Although the set of guidelines provided in this chapter are for specific examples, we hope that these guidelines serve as an inspiration for researchers studying a broader set of applications. [131] showed that Neural Ordinary Differential Equations (Neural ODEs) which are the heart of SciML frameworks, consistently outperform non linear and classical linear methods. Their experiments also showed that Neural ODEs are less sensitive to hyperparameters. In this study, we found that, to obtain the same level of predictive performance, linear models require a much larger set of parameters to train compared to the SciML frameworks which use neural networks. Even then, we could not achieve the same forecasting ability as we did with using neural networks.

In Chapter 6, we introduce a new framework for quantifying uncertainty associated with SciML model predictions. This framework combines Neural Ordinary Differential Equations (Neural ODEs) [20] with efficient Bayesian inference frameworks like the No-U-Turn MCMC sampler (NUTS) [67] which is an extension of the Hamiltonian Monte Carlo Algorithm, and Stochastic Gradient Markov Chain Monte Carlo (SGMCMC) methods like Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) [21] and Stochastic Gradient Langevin Descent (SGLD) [164]. We call this framework as Bayesian Neural Ordinary Differential Equations.

Finally, in Chapter 7 we conclude with the implications of this thesis, and scope for further research.

Chapter 2

The Quarantine SIR (QSIR) model of epidemiology

2.1 Summary

This thesis begins with a new modelling framework, in which, we have developed a globally applicable diagnostic Covid-19 model by augmenting the classical SIR epidemiological model with a neural network module. Our model does not rely upon previous epidemics like SARS/MERS and all parameters are optimized via machine learning algorithms employed on publicly available Covid-19 data. The model decomposes the contributions to the infection timeseries to analyze and compare the role of quarantine control policies employed in highly affected regions of Europe, North America, South America and Asia in controlling the spread of the virus. For all continents considered, our results show a generally strong correlation between strengthening of the quarantine controls as learnt by the model and actions taken by the regions' respective governments. Additionally, we have hosted our quarantine diagnosis results for the top 70 affected countries worldwide, on a public platform.

2.2 Introduction

The Coronavirus respiratory disease 2019 originating from the virus “SARS-CoV-2" [18, 17] led to a global pandemic, leading to $12,552,765$ confirmed global cases in more than 200 countries by July 12, 2020 [165]. As the disease began to spread beyond its apparent origin in Wuhan, the responses of local and national governments varied considerably. The evolution of infections has been similarly diverse, in some cases appearing to be contained and in others reaching catastrophic proportions. In Hubei province itself, starting at the end of January, more than 10 million residents were quarantined by shutting down public transport systems, train and airport stations, and imposing police controls on pedestrian traffic. Subsequently, similar policies were applied nation-wide in China. By the end of March, the rate of infections was reportedly receding [28].

By the end of February 2020, the virus began to spread in Europe, with Italy employing extraordinary quarantine measures starting 11 March 2020. France enacted strict quarantine measures beginning 17 March followed later by UK on 23 March; whereas no measures were enforced in Sweden [44]. South Korea, Iran and Spain experienced acute initial increases, but then adopted drastic generalized quarantine. In the United States, the first infections were detected in Washington State as early as 20th January 2020 [68] and now it is being reported that the virus had been circulating undetected in New York City as early as mid-February [16]. Federal, state and city government responses were comparatively delayed and variable, with most states having stay at home orders [44] declared by the end of March. In South America, Brazil, Chile and Peru are the highest affected countries as of 12 July and they employed differing quarantine policies [139]. Brazil’s first case was reported in the last week of February and the country went into a state of partial quarantine on 24 March. Chile declared a state of disaster for 90 days in the first week of March, and the military was deployed to enforce quarantine measures. In Peru, a nationwide curfew was employed much later, on March 19.

Given the available Covid-19 data for the infected case count by country and world-wide, it is seen that the infection growth curve also showed significantly diverse behaviour globally. In some countries, the infected case count peaked within a month and showed a subsequent decline, while in certain other countries, it was seen to increase for much longer before plateauing. In some of the highly affected countries, the infected count has not yet reached a plateau and the daily active cases continue to increase or remain stagnant as of 12 July 2020. The disparity of the countries' responses is compounded by commensurate disparity in their effectiveness in controlling the severity of infectious spread. This, together with standard challenges in epidemiological modeling and certain unusual features of the disease itself (such as the possibility of individuals to remain asymptomatic yet infectious for up to two weeks) create severe difficulty in interpreting the policies or drawing lessons for future outbreaks.

Here we focus on compartment based modelling, a widely used tool in epidemiology . The earliest version of the compartment model was the SIR (Susceptible-Infected-Recovered) model [82]. Two major assumptions of this class of models are (a) homogeneity and (b) the law of mass action, which states that the rate of change of compartment population at the next time step is proportional to the compartment population at the current time step [2]; hence, compartmental models typically results in a set of coupled ordinary differential equations (ODEs) governing the populations. These simplifying assumptions make the compartment models weaker than the other class of models called agent based models, which are used to simulate autonomous agents and their interactions within a constrained environment (see [40] and references therein for a detailed introduction). Although it is easier to incorporate heterogeneity in agent based models, the significant advantage of compartment modelling is interpretibility. This is because physically meaningful information about the system, such as the reproduction number [159] can be extracted directly from the ODEs. Stochastic variations of compartment based models [76, 1, 30] and Bayesian approaches [39] have also been studied.

For analyzing different aspects of the Covid-19 outbreak, compartment based models which are based on the SEIR (Susceptible-Exposed-Infected-Recovered) framework have been used widely [138, 93, 107, 102, 123, 162]. From such studies, it is seen that although increasing the number of compartments results in more realistic behaviour, the model then becomes less identifiable; *i.e.* it becomes progressively more difficult to uniquely determine parameters from the data [143]. For example, while analyzing the Covid-19 outbreak for Wuhan, China, it has been shown in a recent study [143] that the large number of parameters in the SEIR models makes it less reliable than the simpler SIR models.

To deal with the aforementioned disparity between government responses and outcomes to the Covid-19 pandemic, several models studied the effect of quarantine/lockdown measures on the evolution of the disease [143, 93, 162, 123, 147]. Existing models generally

- lack independent estimation: using parameters based on prior knowledge of SARS/MERS coronavirus epidemiology and not derived independently from the Covid-19 data or parameters like rate of detection, nature of government response fixed prior to running the model; or
- lack global applicability: they are not implemented on a global scale; or
- lack interpretibility, as we defined it earlier.

In this chapter, we propose a globally scalable, interpretable compartment based model with entirely independent parameter estimation through a novel approach: augmenting a first principles-derived epidemiological model with a data-driven module, implemented as a neural network. Prior approaches of functional quantification through data involve probabilistic methods like variational inference [65, 163, 80, 66, 134, 84, 141, 135, 108, 64] and Variational Gaussian Processes [158], which do not incorporate knowledge of the ordinary differential equations governing the system under consideration. We leverage our model to quantify the quarantine strengths and analyze and compare the role of quarantine control policies employed to control the virus effective reproduction number [71, 138, 154, 94, 167, 87, 38] in the European, North American, South American and Asian continents. In the SEIR model[37, 145, 151], the population is divided into the susceptible S, exposed E, infected I and recovered R groups, and their relative growths and competition are represented as a set of coupled ordinary differential equations; whereas the simpler SIR model does not account for the exposed population E. These models cannot capture the large-scale effects of more granular interactions, such as the population's response to social distancing and quarantine policies. However, a major assumption of these models is that the rate of transitions between population states is fixed. In our approach, we relax this assumption by estimating the time-dependent quarantine effect on virus exposure as a neural network informs the infected variable I in the SIR model. This trained model thus decomposes the effects and the neural network encodes information about the quarantine strength function in the locale where the model is trained.

In general, neural networks with arbitrary activation functions are universal approximators [27, 69, 152]. Unbounded activation functions, in particular, such as the rectified linear unit (ReLU) has been known to be effective in approximating nonlinear functions with a finite set of parameters [48, 50, 29]. Thus, a neural network solution is attractive to approximate quarantine effects in combination with analytical epidemiological models. The downside is that the presence of the neural network term as a component of the ODEs results in limited interpretability. The recently emerging field of Scientific Machine Learning [8] exploits conservation principles within a universal differential equation [129], SIR in our case, to mitigate overfitting and other related machine learning risks.

In the present work, the neural network is trained from publicly available infection and population data for Covid-19 for a specific region under study; results for which are provided in Section 3 followed by a discussion in Section 4. Details of the model estimation procedure and parameter inference are presented in Section 5 (the Experimental Procedures section).

2.3 Results

2.3.1 Standard SIR model

The classic SIR epidemiological model is a standard tool for basic analysis concerning the outbreak of epidemics. In this model, the entire population is divided into three sub-populations: susceptible S; infected I; and recovered R. The sub-populations' evolution is governed by the following system of three coupled nonlinear ordinary differential equations

$$\frac{\mathrm{d}S(t)}{\mathrm{d}t} = -\frac{\beta\, S(t)\, I(t)}{N} \tag{2.1}$$

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = \frac{\beta\, S(t)\, I(t)}{N} - \gamma I(t) \tag{2.2}$$

$$\frac{\mathrm{d}R(t)}{\mathrm{d}t} = \gamma I(t). \tag{2.3}$$

Here, β and γ are the infection and recovery rates, respectively, and are assumed to be constant in time. The total population $N = S(t) + I(t) + R(t)$ is seen to remain constant as well; that is, births and deaths (unrelated to the disease) are neglected. The recovered population is to be interpreted as those who can no longer infect others; so it also includes individuals deceased due to the infection. The possibility of recovered individuals to become reinfected is accounted for by SEIS models [105], but we do not use this model here, as the negligibly few reinfection cases for Covid-19 have been recorded as of now [REFS]. The reproduction number R_t in the SEIR and SIR models is defined as

$$R_t = \frac{\beta}{\gamma}. \tag{2.4}$$

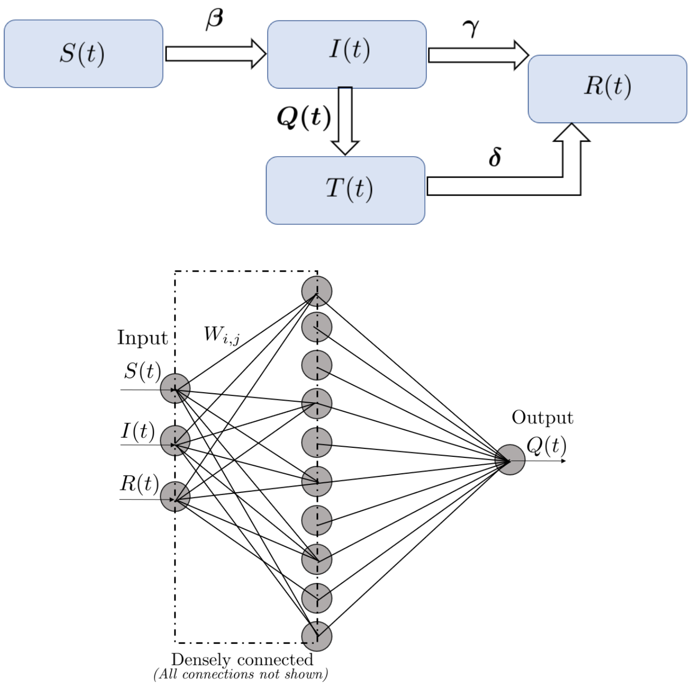


Figure 2-1: [Illustration of the QSIR Model and neural network architecture] (a) Schematic of the augmented QSIR model considered in the present study. (b) Schematic of the neural network architecture used to learn the quarantine strength function $Q(t)$. Here $T(t)$ represents the quarantined infected population prescribed by the quarantine strength rate $Q(t)$.

An important assumption of the SIR models is homogeneous mixing among the subpopulations. Therefore, this model cannot account for social distancing or social network effects. Additionally the model assumes uniform susceptibility and disease progress for every individual; and that no spreading occurs through animals or other non-human means. Alternatively, the SIR model may be interpreted as quantifying the statistical expectations on the respective mean populations, while deviations from the model's assumptions contribute to statistical fluctuations around the mean.

2.3.2 Augmented QSIR model

To study the effect of quarantine control globally, we start with the SIR epidemiological model. Figure 2-1a shows the schematic of the modified SIR model, the QSIR model, which we consider. We augment the SIR model by introducing a time varying quarantine strength rate term $Q(t)$ and a quarantined population $T(t)$, which is prevented from having any further contact with the susceptible population. Thus, the term $I(t)$ denotes the infected population still having contact with the susceptibles, as done in the standard SIR model; while the term $T(t)$ denotes the infected population who are effectively quarantined and isolated. Further we introduce an additional recovery rate δ which quantifies the rate of recovery of the quarantined population. Thus, we can write an expression for the quarantined infected population $T(t)$ as

$$\frac{dT(t)}{dt} = Q(t)I(t) - \delta T(t) \tag{2.5}$$

Based on the modified model, we define a Covid spread parameter in a similar way to the reproduction number defined in the SIR model (2.4) as

$$C_p(t) = \frac{\beta}{\gamma + \delta + Q(t)}. \tag{2.6}$$

$C_p > 1$ indicates that infections are being introduced into the population at a higher rate than they are being removed, leading to rapid spread of the disease. On the other hand, $C_p < 1$ indicates that the Covid spread has been brought under control in the region of consideration. Since $Q(t)$ does not follow from first principles and is highly dependent on local quarantine policies, we devised a neural network-based approach to approximate it.

Recently, it has been shown that neural networks can be used as function approximators to recover unknown constitutive relationships in a system of coupled ordinary differential equations [129, 128]. Following this principle, we represent $Q(t)$ as a n layer-deep neural network with weights $W_1, W_2 \dots W_n$, activation function r and the input vector $U = (S(t), I(t), R(t))$ as

$$Q(t) = r\left(W_n r\left(W_{n-1} \dots r\left(W_1 U\right)\right)\right) \equiv \mathrm{NN}(W, U) \tag{2.7}$$

For the implementation, we choose a $n = 2$-layer densely connected neural network with 10 units in the hidden layer and the ReLU activation function. This choice was because we found sigmoidal activation functions to stagnate. The final model is described by a total of 54 tunable parameters. The neural network architecture schematic is shown in figure 2-1b. The governing coupled ordinary differential equations for the QSIR model are

$$\frac{\mathrm{d}S(t)}{\mathrm{d}t} = -\frac{\beta\, S(t)\, I(t)}{N} \tag{2.8}$$

$$\begin{aligned}\frac{\mathrm{d}I(t)}{\mathrm{d}t} &= \frac{\beta\, S(t)\, I(t)}{N} - \big(\gamma + Q(t)\big)\, I(t)\\ &= \frac{\beta\, S(t)\, I(t)}{N} - \big(\gamma + \mathrm{NN}(W, U)\big)\, I(t)\end{aligned} \tag{2.9}$$

$$\frac{\mathrm{d}R(t)}{\mathrm{d}t} = \gamma I(t) + \delta T(t) \tag{2.10}$$

$$\frac{\mathrm{d}T(t)}{\mathrm{d}t} = Q(t)\, I(t) - \delta T(t) = \mathrm{NN}(W, U)\, I(t) - \delta T(t). \tag{2.11}$$

More details about the model initialization and parameter estimation methods is given in the Experimental Procedures section.In all cases considered below, we trained the model using data starting from the dates when the 500th infection was recorded in each region and up to June 1 2020.

2.3.3 Interpretation of $Q(t)$

$Q(t)$ denotes the rate at which infected persons are effectively quarantined and isolated from the remaining population, and thus gives composite information about (a) the effective testing rate of the infected population as the disease progressed and (b) the intensity of the enforced quarantine as a function of time. To understand the nature of evolution of $Q(t)$, we look at the time point when $Q(t)$ approximately shows an inflection point or a sudden increase in $Q(t)$. An inflection point in $Q(t)$ indicates the time when the rate of increase of $Q(t)$ i.e $dQ(t)/dt$ was at its peak while a sudden increase corresponds to a sudden intensification of quarantine policies employed in the region under consideration.

Introduction of $Q(t)$ in the SIR model has a similar effect as that of having a time varying decreasing contact rate $\beta(t)$ within the population; which would simulate a lockdown situation. As a result, although $Q(t)$ denotes infected population quarantine, the way it is introduced in our augmented SIR model enables our model to capture broad level population lockdown effects; without burdening regression with additional parameters. We demonstrate this ability of our model in the results of the subsequent sections.

Further, we define the quarantine efficiency, Q_{eff} as the increase in $Q(t)$ within a month following the detection of the 500th infected case in the region under consideration. Thus

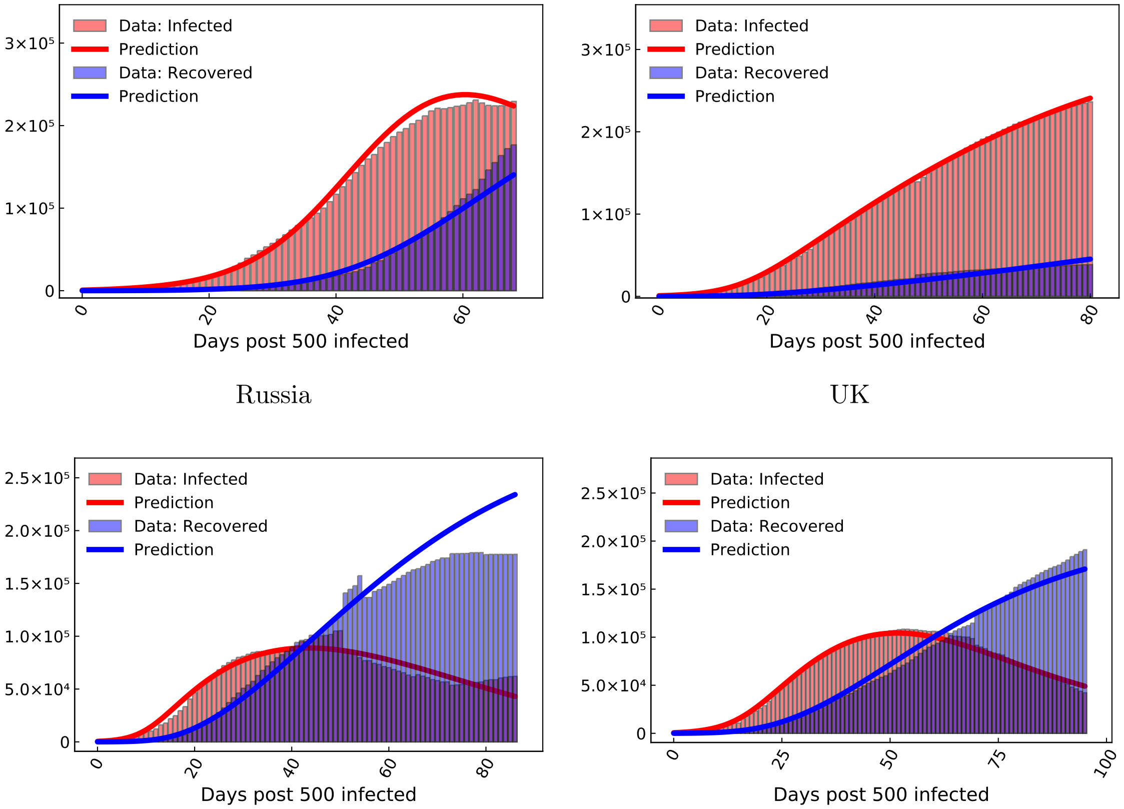
$$Q_{\text{eff}} = Q(30) - Q(1) \tag{2.12}$$

The magnitude of Q_{eff} shows how rapidly the infected individuals were prevented from coming into contact with the susceptibles in the first month following the detection of the 500th infected case; and is indicative of the quarantine responsiveness: the testing and tracing protocols to identify and isolate infected individuals.

2.3.4 Europe

Figure 2-2 shows the comparison of the model-estimated infected and recovered case counts with actual Covid-19 data for the highest affected European countries as of 1 June 2020, namely: Russia, UK, Spain and Italy, in that order. We find that irrespective of a small set of optimized parameters (note that the contact rate β and the recovery rate γ are fixed, and not functions of time), a reasonably good match is seen in all four cases.

Figure 2-3 shows the evolution of the neural network learnt quarantine strength function $Q(t)$ for the considered European nations. Inflection points in $Q(t)$ are seen



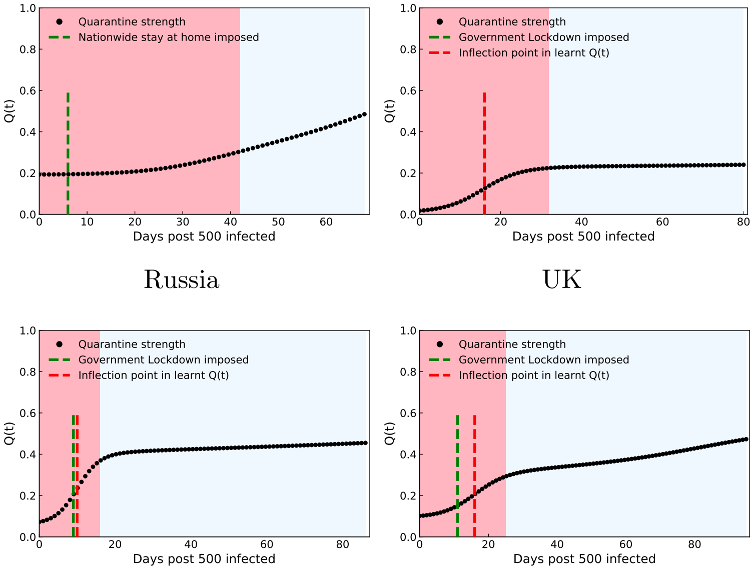
Russia

UK

Spain

Italy

Figure 2-2: [Europe: Infected and Recovered Covid-19 case count evolution] COVID-19 infected and recovered evolution compared with our neural network augmented model prediction in the highest affected European countries as of June 1, 2020.



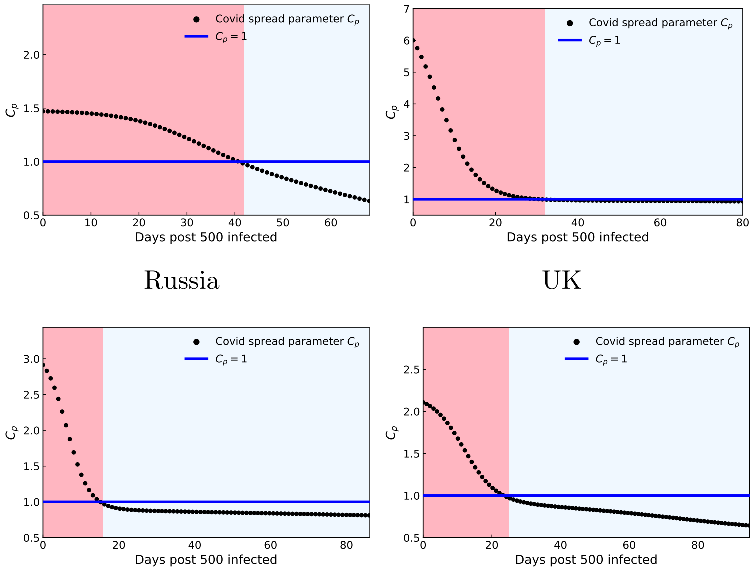
Russia

UK

Spain

Italy

Figure 2-3: [Europe: Quarantine strength evolution in response to Covid-19] Quarantine strength $Q(t)$ learned by the neural network in the highest affected European countries as of June 1, 2020. The transition from the red to blue shaded region indicates the Covid spread parameter of value $C_p < 1$ leading to halting of the infection spread. The green dashed line indicates the time when quarantine measures were implemented in the region under consideration, which generally matches well with an inflection point seen in the $Q(t)$ plot denoted by the red dashed line. For regions in which a clear inflection or ramp up point is not seen (Russia), the red dashed line is not shown.



Russia

UK

Spain

Italy

Figure 2-4: [Europe: Covid spread parameter evolution in response to Covid-19] Control of COVID-19 quantified by the Covid spread parameter evolution in the highest affected European countries as of June 1, 2020. The transition from the red to blue shaded region indicates $C_p < 1$ leading to halting of the infection spread.

for UK, Spain and Italy at 14, 10 and 16 days, respectively, post detection of the 500th case i.e on 23th March, 15th March and 14th March, respectively. This is in good agreement with nationwide quarantine imposed on 25th March, 14th March and 9th March in UK, Spain and Italy, respectively [44, 79, 59].

Figure 2-16a shows the comparison of the contact rate β, quarantine efficiency as defined in the beginning of this subsection and the recovery rate γ. It should be noted that the contact and recovery rates are assumed to be constant in our model, in the duration spanning the detection of the 500th infected case and June 1st, 2020. The average contact rate in Spain and Italy is seen to be higher than Russia and UK over the considered duration of $2-3$ months, possibly because Russia and UK were affected relatively late by the virus, which gave sufficient time for the enforcement strict social distancing protocols prior to widespread outbreak. For Spain and Italy, the quarantine efficiency and also the recovery rate are generally higher than for Russia and UK, possibly indicating more efficient testing, isolation and quarantine;

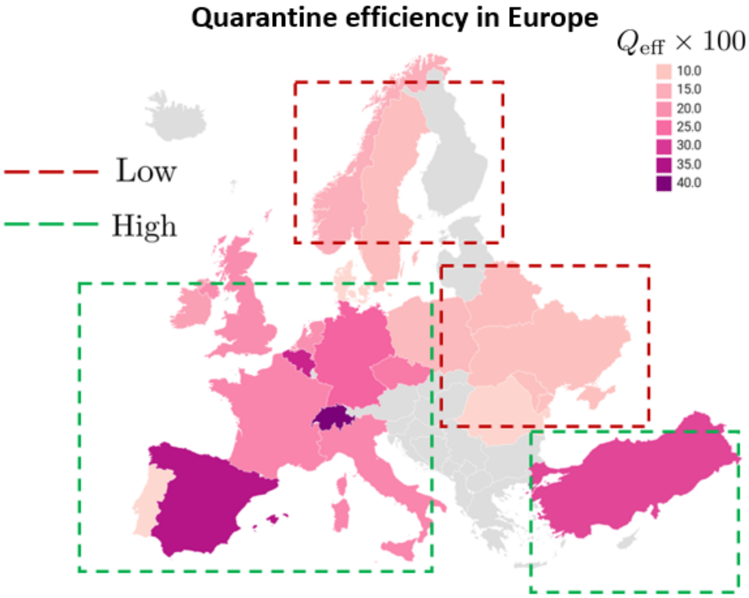


Figure 2-5: [Europe: Quarantine efficiency heatmap] (a) Quarantine efficiency, Q_{eff} defined in (12) for the 23 highest affected European countries. Note that Q_{eff} is indicative of the quarantine responsiveness: the testing and tracing protocols to identify and isolate infected individuals. Map also shows the demarcation between countries with a high Q_{eff} shown by a green dotted line and those with a low Q_{eff} shown by a red dotted line.

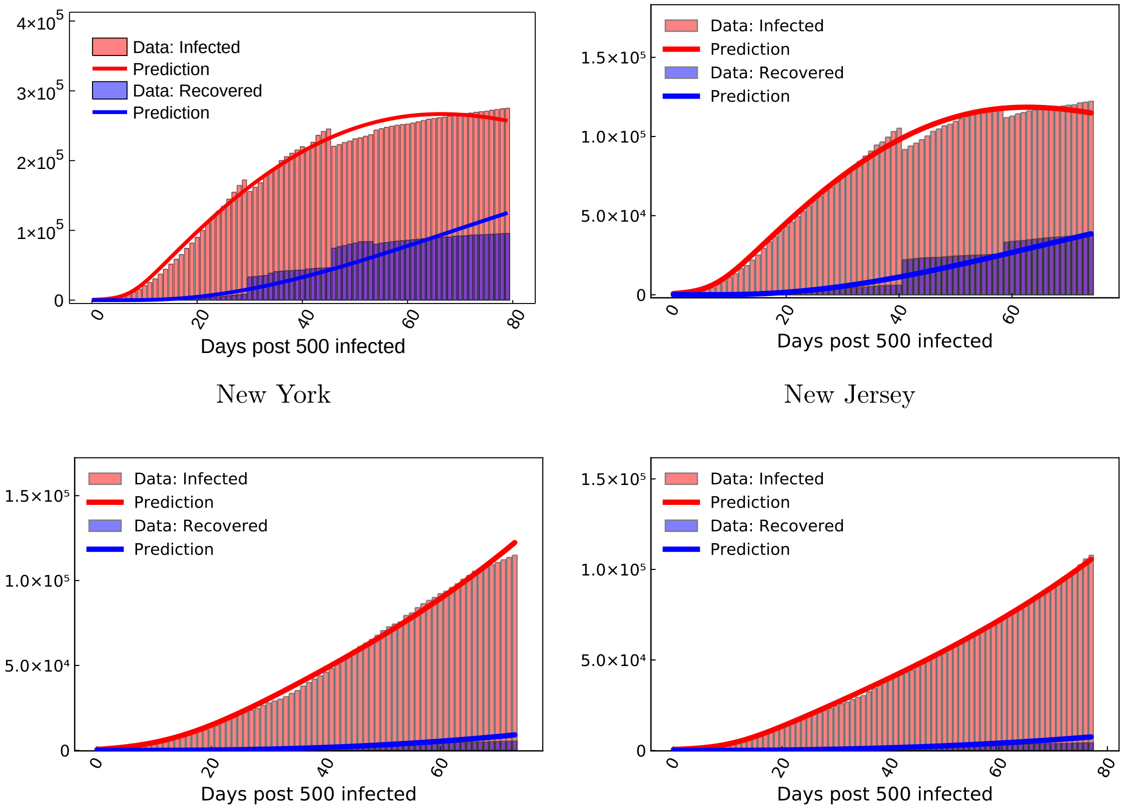
and hospital practices in Spain and Italy. This agrees well with the ineffectiveness of testing, contact tracing and quarantine practices seen in UK [62]. Although the social distancing strength also varied with time, we do not focus on that aspect in the present study, and will be the subject of future studies. A higher quarantine efficiency combined with a higher recovery rate led Spain and Italy to bring down the Covid spread parameter (defined in (2.6)), C_p from > 1 to < 1 in $16, 25$ days. respectively, as compared to 32 days for UK and 42 days for Russia (figure 2-4).

Quarantine efficiency map for Europe

Figure 2-5 shows Q_{eff} for the 23 highest affected European countries. We can see that Q_{eff} in the western European regions is generally higher than eastern Europe. This can be attributed to the strong quarantine response measures implemented in western countries like Spain, Italy, Germany, France after the rise of infections seen first in Italy and Spain [34]. Although countries like Switzerland and Turkey didn't enforce a strict quarantine response as compared to their west European counterparts, they were generally successful in halting the infection count before reaching catastrophic proportions, due to strong testing and tracing protocols [113, 58]. Subsequently, these countries also managed to identify potentially infected individuals and prevented them from coming into contact with susceptibles, giving them a high Q_{eff} score as seen in figure 2-5. In contrast, our study also manages to identify countries like Sweden which had very limited quarantine measures [51]; with a low Q_{eff} score as seen in figure 2-5. This strengthens the validity of our model in diagnosing information about the effectiveness of quarantine and isolation protocols in different countries; which agree well with the actual protocols seen in these countries.

2.3.5 USA

Figure 2-6 shows reasonably good match between the model-estimated infected and recovered case counts with actual Covid-19 data for the highest affected North American states (including states from Mexico, the United States, and Canada) as of 1 June 2020, namely: New York, New Jersey, Illinois and California. $Q(t)$ for New York and New Jersey show a ramp up point immediately in the week following the detection of the 500th case in these regions, *i.e.* on 19 March for New York and on 24 March for New Jersey (figure 2-7). This matches well with the actual dates: 22 March in New York and 21 March in New Jersey when stay at home orders and isolation measures were enforced in these states. A relatively slower rise of $Q(t)$ is seen for Illinois while California showing a ramp up post a week after detection of the 500th case. Although no significant difference is seen in the mean contact and recovery rates between the different US states, the quarantine efficiency in New York and New Jersey is seen to be significantly higher than that of Illinois and California (figure 2-16b), indicating the effectiveness of the rapidly deployed quarantine interventions in New York and New Jersey [98]. Owing to the high quarantine efficiency in New York and New Jersey, these states were able to bring down the Covid spread parameter, C_p to less than 1 in 19 days (figure 2-8). On the other hand, although Illinois and California reached close to $C_p = 1$ after the 30 day and 20 day mark respectively, C_p still remained greater



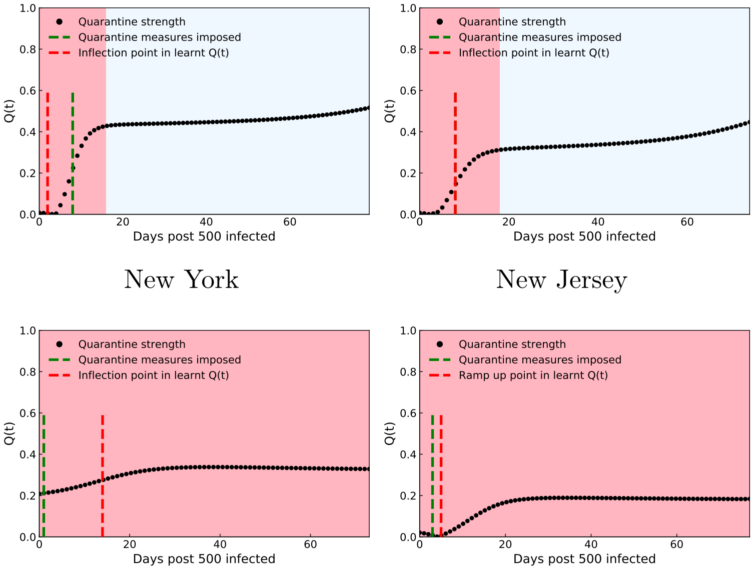
New York

New Jersey

Illinois

California

Figure 2-6: [USA: Infected and Recovered Covid-19 case count evolution] COVID-19 infected and recovered evolution compared with our neural network augmented model prediction in the highest affected USA states as of June 1, 2020.



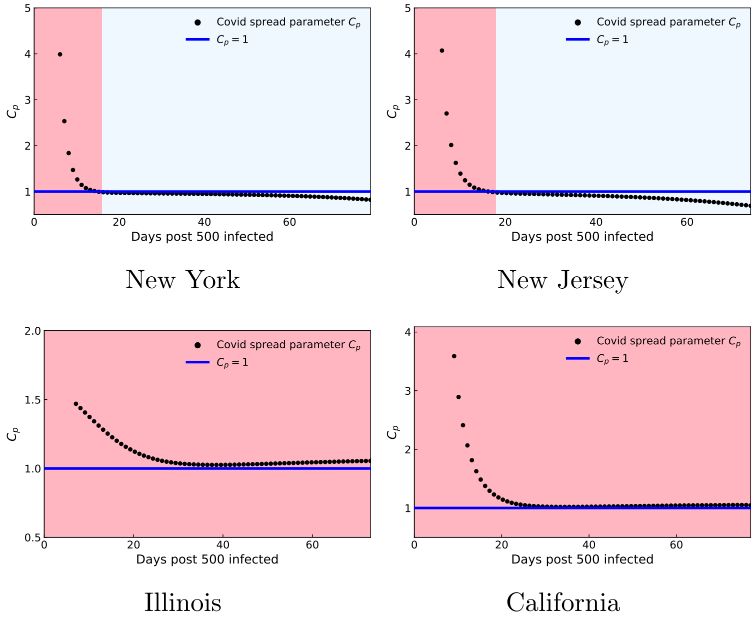
New York

New Jersey

Illinois

California

Figure 2-7: [USA: Quarantine strength evolution in response to Covid-19] Quarantine strength $Q(t)$ learned by the neural network in the highest affected USA states as of June 1, 2020. The transition from the red to blue shaded region indicates the Covid spread parameter of value $C_p < 1$ leading to halting of the infection spread. The green dashed line indicates the time when quarantine measures were implemented in the region under consideration, which generally matches well with an inflection point (for New York, New Jersey and Illinois) or a ramp up point (California) seen in the $Q(t)$ plot denoted by the red dashed line.



New York

New Jersey

Illinois

California

Figure 2-8: [USA: Covid spread parameter evolution in response to Covid-19] Control of COVID-19 quantified by the Covid spread parameter evolution in the highest affected USA states as of June 1, 2020. The transition from the red to blue shaded region indicates $C_p < 1$ leading to halting of the infection spread.

than 1 (figure 2-8), indicating that these states were still in the danger zone as of June 1, 2020. An important caveat to this result is the reporting of the recovered data.

Comparing with Europe, the recovery rates seen in North America are significantly lower (figures 16a,b). It should be noted that accurate reporting of recovery rates is likely to play a major role in this apparent difference. In our study, the recovered data include individuals who cannot further transmit infection; and thus includes treated patients who are currently in a healthy state and also individuals who died due to the virus. Since quantification of deaths can be done in a robust manner, the death data is generally reported more accurately. However, there is no clear definition for quantifying the number of people who transitioned from infected to healthy. As a result, accurate and timely reporting of recovered data is seen to have a significant variation between countries, under reporting of the recovered data being a common practice. Since the effective reproduction number calculation depends on the recovered case count, accurate data regarding the recovered count is vital to as-

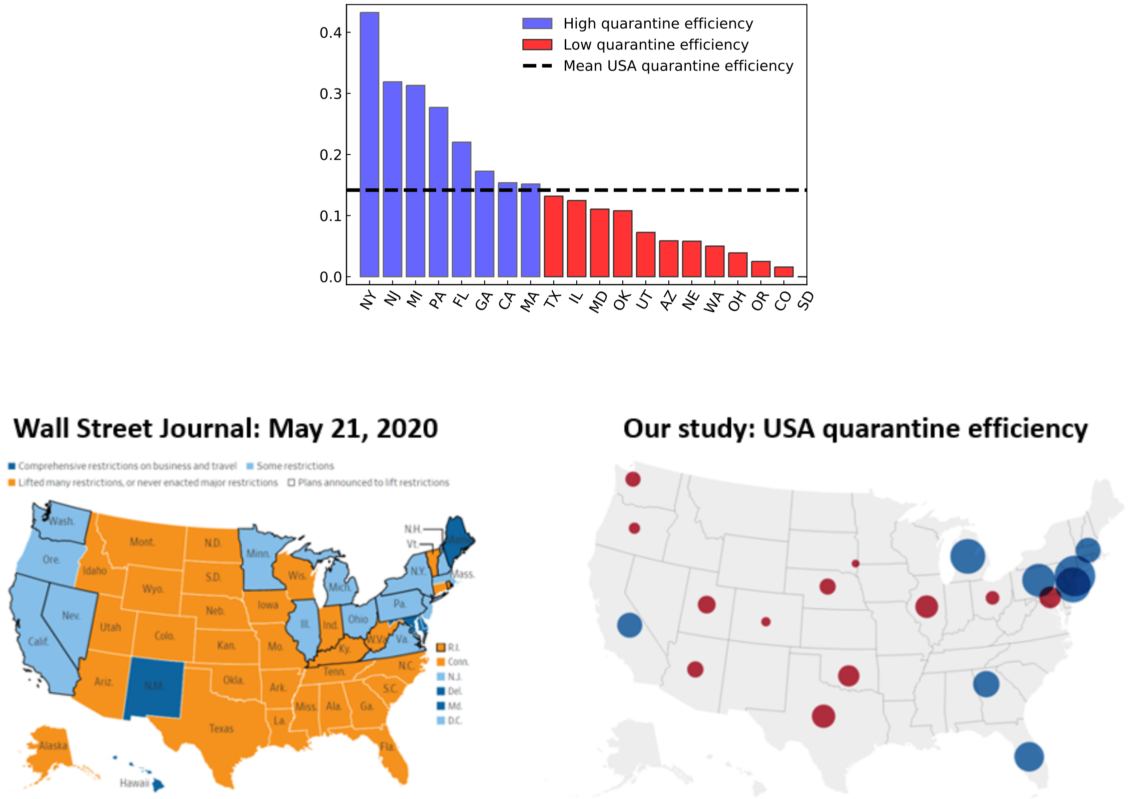


Figure 2-9: [USA: Quarantine efficiency heatmap and its comparison with ground truth data] (a) Quarantine efficiency, Q_{eff} defined in (12) for 20 major USA states. Note that Q_{eff} is indicative of the quarantine responsiveness: the testing and tracing protocols to identify and isolate infected individuals. (b) Comparison between a report published in the Wall Street Journal on May 21 [43] and the quarantine efficiency magnitude in our study. A generally strong correlation is seen between the magnitude of quarantine efficiency in our study and the level of restrictions actually imposed in different USA states.

sess whether the infection has been curtailed in a particular region or not. Thus, our results strongly indicate the need for each country to follow a particular metric for estimating the recovered count robustly, which is vital for data driven assessment of the pandemic spread.

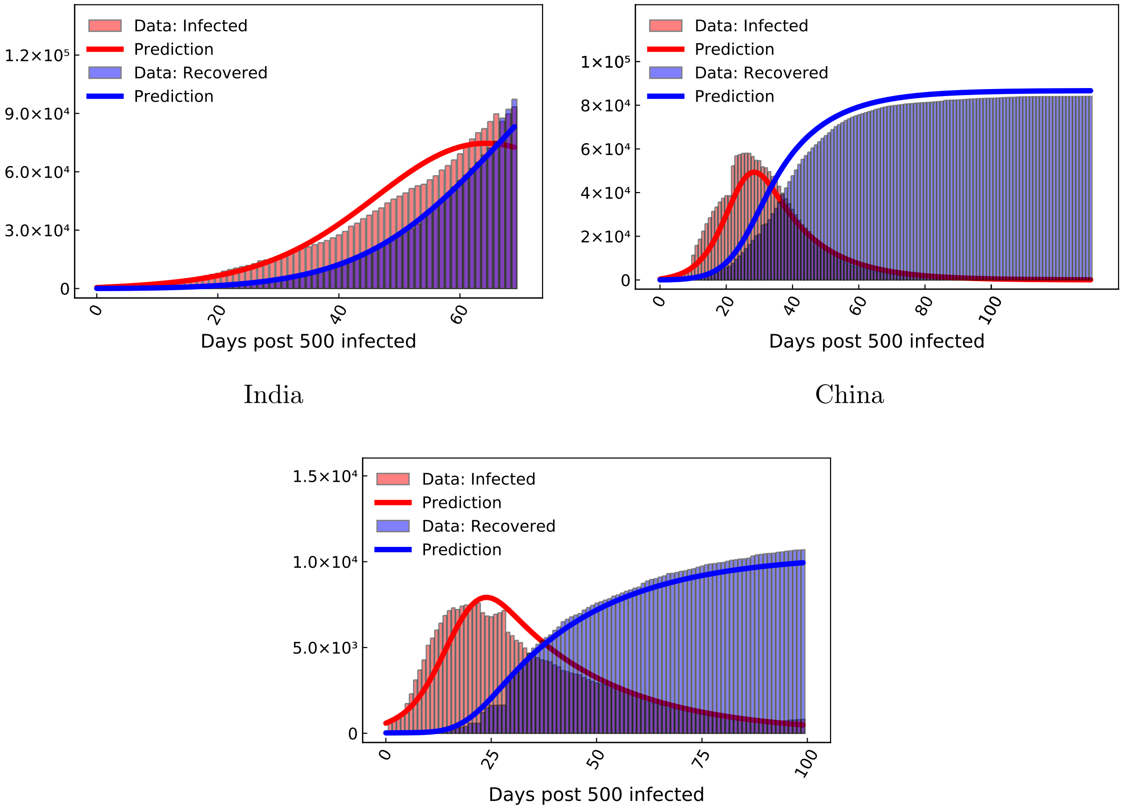
Quarantine efficiency map for the USA

Figure 2-9a shows the quarantine efficiency for 20 major US states spanning the whole country. Figure 2-9b shows the comparison between a report published in the Wall

Street Journal on May 21 highlighting USA states based on the quarantine measure strength [43], and the quarantine efficiency magnitude in our study. The size of the circles represent the magnitude of the quarantine efficiency. The blue color indicates the states for which the quarantine efficiency was greater than the mean quarantine efficiency across all US states, while those in red indicate the opposite. Our results indicate that the north-eastern and western states were much more responsive in implementing rapid quarantine measures in the month following early detection; as compared to the southern and central states. This matches the on-ground situation as indicated by a generally strong correlation is seen between the red circles in our study (states with lower quarantine efficiency) and the yellow regions seen in in the Wall Street Journal report [43] (states with reduced imposition of restrictions) and between the blue circles in our study (states with higher quarantine efficiency) and the blue regions seen in the Wall Street Journal report [43] (states with generally higher level of restrictions). This strengthens the validity of our approach in which the quarantine efficiency is recovered through a trained neural network rooted in fundamental epidemiological equations.

2.3.6 Asia

Figure 2-10 shows reasonably good match between the model-estimated infected and recovered case count with actual Covid-19 data for the highest affected Asian countries as of 1 June 2020, namely: India, China and South Korea. $Q(t)$ shows a rapid ramp up in China and South Korea (figure 2-11) which agrees well with cusps in government interventions which took place in the weeks leading to and after the end of January [28] and February [112] for China and South Korea respectively. On the other hand, a slow build up of $Q(t)$ is seen for India, with no significant ramp up. This is reflected in the quarantine efficiency comparison (figure 2-16c), which is much higher for China and South Korea compared to India. South Korea shows a significantly lower contact rate than its Asian counterparts, indicating strongly enforced and followed social distancing protocols [156]. No significant difference in the recovery rate is observed between the Asian countries. Owing to the high quarantine efficiency in China and a

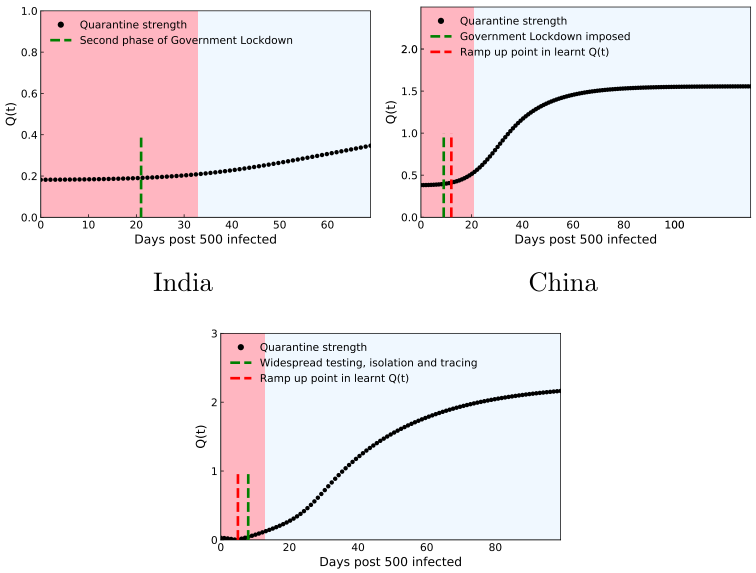


India

China

South Korea

Figure 2-10: [Asia: Infected and Recovered Covid-19 case count evolution] COVID-19 infected and recovered evolution compared with our neural network augmented model prediction in the highest affected Asian countries as of June 1, 2020.

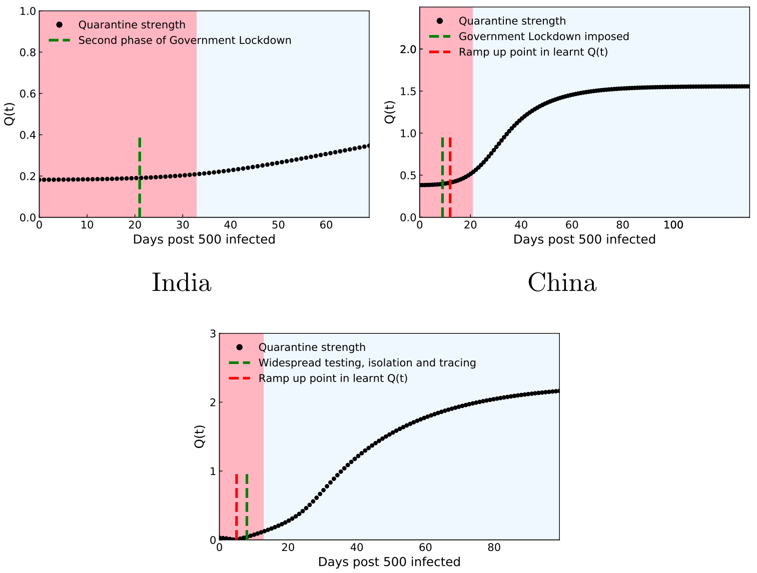


India

China

South Korea

Figure 2-11: [Asia: Quarantine strength evolution in response to Covid-19] Quarantine strength $Q(t)$ learnt by the neural network in the highest affected Asian countries as of June 1, 2020. The transition from the red to blue shaded region indicates the Covid spread parameter of value $C_p < 1$ leading to halting of the infection spread. The green dashed line indicates the time when quarantine measures were implemented in the region under consideration, which generally matches well with a ramp up point seen in the $Q(t)$ plot denoted by the red dashed line. For regions in which a clear inflection or ramp up point is not seen (India), the red dashed line is not shown.

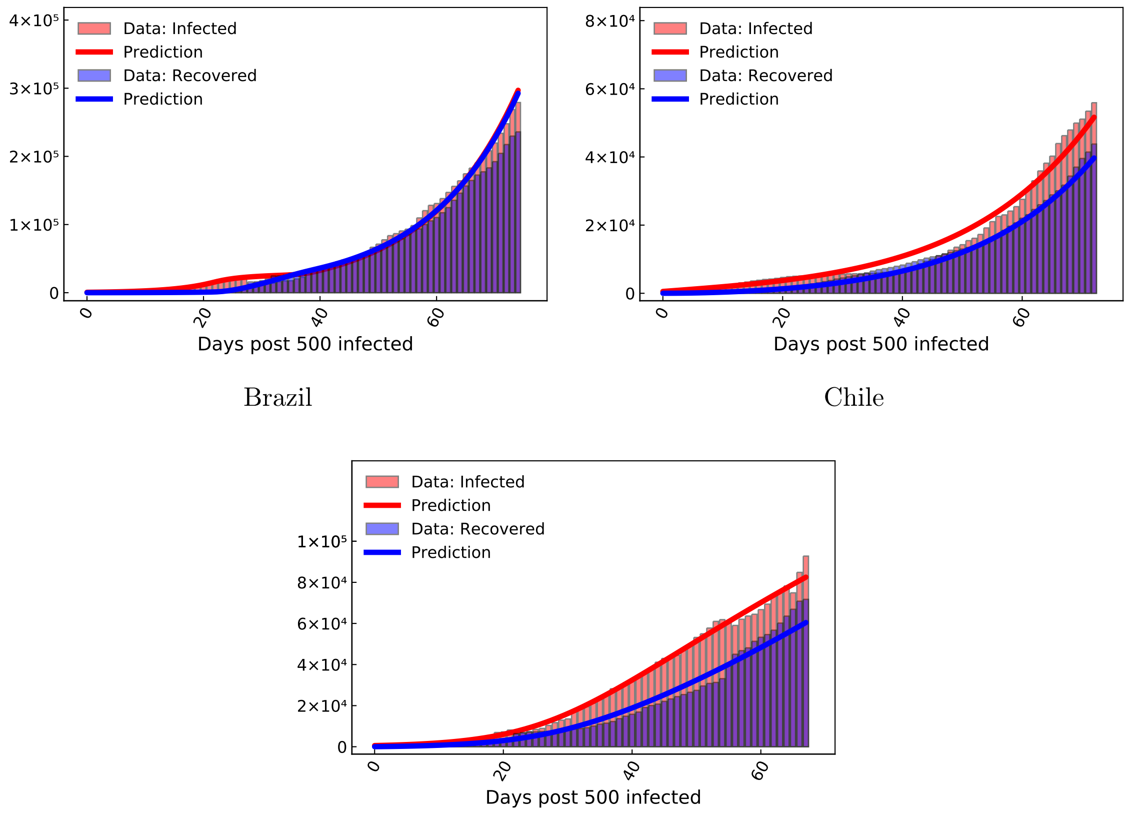


India

China

South Korea

Figure 2-12: [Asia: Quarantine strength evolution in response to Covid-19] Quarantine strength $Q(t)$ learnt by the neural network in the highest affected Asian countries as of June 1, 2020. The transition from the red to blue shaded region indicates the Covid spread parameter of value $C_p < 1$ leading to halting of the infection spread. The green dashed line indicates the time when quarantine measures were implemented in the region under consideration, which generally matches well with a ramp up point seen in the $Q(t)$ plot denoted by the red dashed line. For regions in which a clear inflection or ramp up point is not seen (India), the red dashed line is not shown.



Brazil

Chile

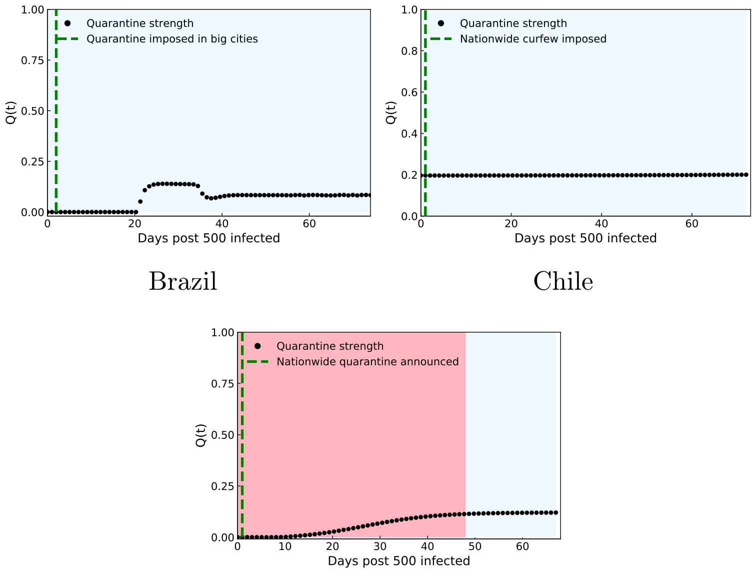
Peru

Figure 2-13: [South America: Infected and Recovered Covid-19 case count evolution] COVID-19 infected and recovered evolution compared with our neural network augmented model prediction in the highest affected South American countries as of June 1, 2020.

high quarantine efficiency coupled with strongly enforced social distancing in South Korea, these countries were able to bring down the Covid spread parameter C_p from > 1 to < 1 in 21 and 13 days respectively, while it took 33 days in India (figure 2-12).

2.3.7 South America

Figure 2-13 shows reasonably good match between the model-estimated infected and recovered case count with actual Covid-19 data for the highest affected South American countries as of 1 June 2020, namely: Brazil, Chile and Peru. For Brazil, $Q(t)$ is seen to be approximately constant ≈ 0 initially with a ramp up around the 20 day mark; after which $Q(t)$ is seen to stagnate (figure 2-14a). The key difference between the Covid progression in Brazil compared to other nations is that the infected and the

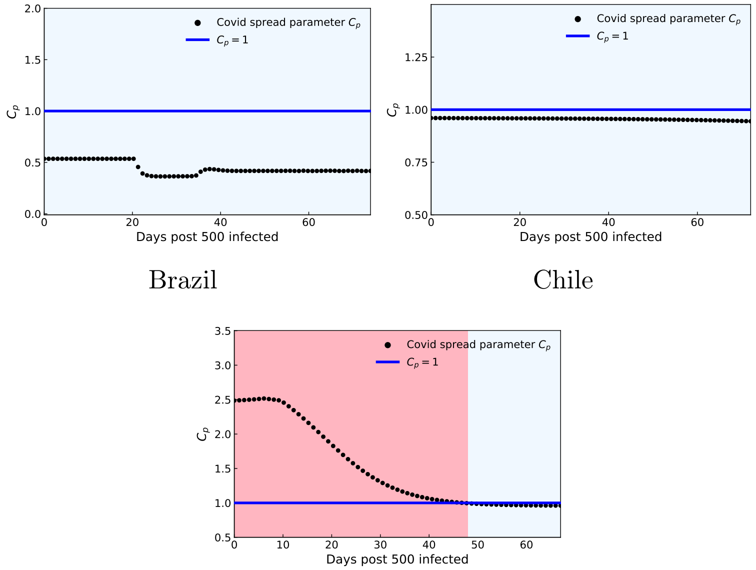


Brazil

Chile

Peru

Figure 2-14: [South America: Quarantine strength evolution in response to Covid-19] Quarantine strength $Q(t)$ learnt by the neural network in the highest affected South American countries as of June 1, 2020. The transition from the red to blue shaded region indicates the Covid spread parameter of value $C_p < 1$ leading to halting of the infection spread. The green dotted line indicates the time when quarantine measures were implemented in the region under consideration.

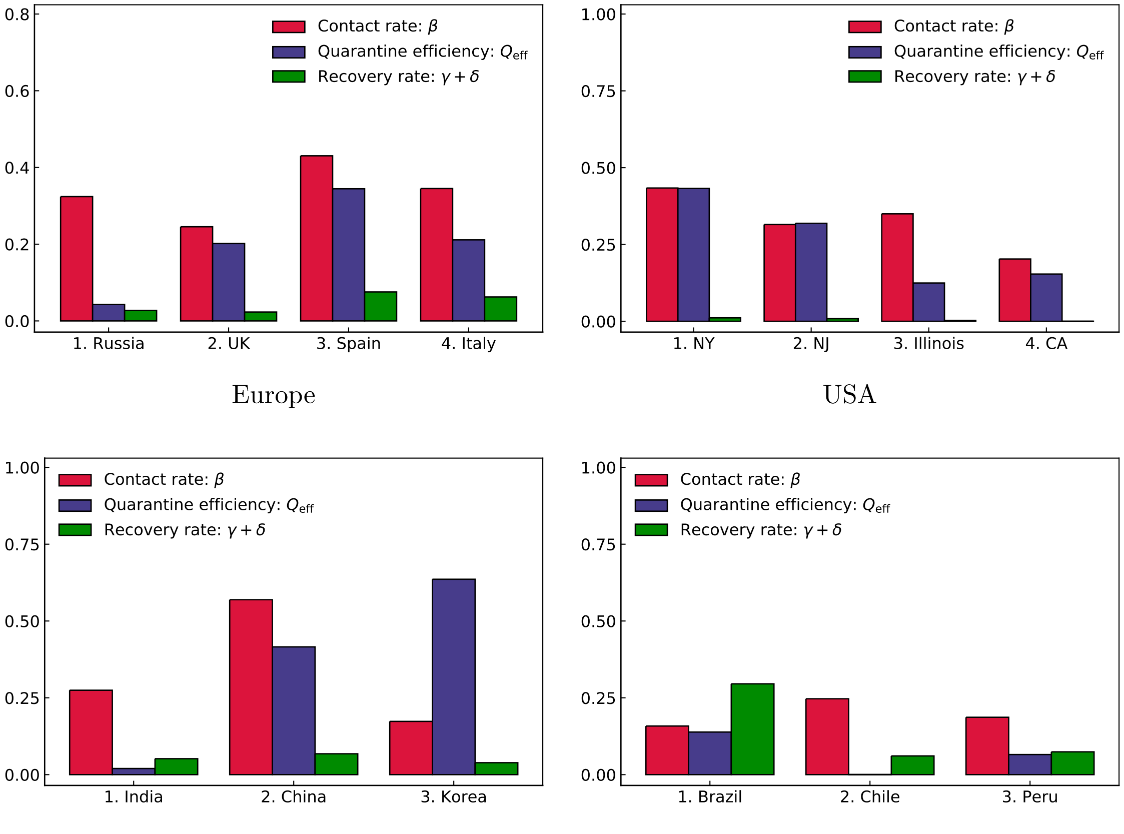


Brazil

Chile

Peru

Figure 2-15: [South America: Covid spread parameter evolution in response to Covid-19] Control of COVID-19 quantified by the Covid spread parameter evolution in the highest affected South American countries as of June 1, 2020. The transition from the red to blue shaded region indicates $C_p < 1$ leading to halting of the infection spread.



Europe

USA

Asia

South America

Figure 2-16: [Covid-19 spread and subsequent response of majorly affected continents and countries therein] Global comparison of infection, recovery rates and quarantine efficiency.

recovered (recovered healthy + dead in our study) count is very close to one another as the disease progressed (figure 2-13). Owing to this, as the disease progressed, the new infected people introduced in the population were balanced by the infected people removed from the population, either by being healthy or deceased. This higher recovery rate combined with a generally low quarantine efficiency and contact rate (figure 2-16d) manifests itself in the Covid spread parameter for Brazil to be < 1 for almost the entire duration of the disease progression (figure 2-15a). For Chile, $Q(t)$ is almost constant for the entire duration considered (figure 2-14b). Thus, although government regulations were imposed swiftly following the initial detection of the virus, leading to a high initial magnitude of $Q(t)$, the government imposition became subsequently relaxed. This may be attributed to several political and social factors outside the scope of the present study [144]. Even for Chile, the infected and recovered count remain close to each other compared to other nations. A generally high quarantine magnitude coupled with a moderate recovery rate (figure 2-16d) leads to C_p being < 1 for the entire duration of disease progression (figure 2-15b). In Peru, $Q(t)$ shows a very slow build up (figure 2-14c) with a very low magnitude. Also, the recovered count is lower than the infected count compared to its South American counterparts (figure 2-13c). A low quarantine efficiency coupled with a low recovery rate (figure 2-16d) leads Peru to be in the danger zone ($C_p > 1$) for 48 days post detection of the 500th case (figure 2-15c).

2.4 Discussion

Our model captures the infected and recovered counts for highly affected countries in Europe, North America, Asia and South America reasonably well, and is thus globally applicable. Along with capturing the evolution of infected and recovered data, the novel machine learning aided epidemiological approach allows us to extract valuable information regarding the quarantine policies, the evolution of Covid spread parameter C_p, the mean contact rate (social distancing effectiveness), and the recovery rate.

Thus, it becomes possible to compare across different countries, with the model serving as an important diagnostic tool.

Our results show a generally strong correlation between strengthening of the quarantine controls, *i.e.* increasing $Q(t)$ as learnt by the neural network model; actions taken by the regions' respective governments; and decrease of the Covid spread parameter C_p for all continents considered in the present study.

Based on the Covid-19 data collected (details in the Materials and Methods section), we note that accurate and timely reporting of recovered data is seen to have a significant variation between countries; with under reporting of the recovered data being a common practice. In the North American countries, for example, the recovered data are significantly lower than its European and Asian counterparts. Thus, our results strongly indicate the need for each country to follow a particular metric for estimating the recovered count robustly, which is vital for data driven assessment of the pandemic spread.

The key highlights of our model are: (a) it is highly interpretable with few free parameters rooted in an epidemiological model, (b) its reliance on only Covid-19 data and not on previous epidemics and (c) it is highly flexible and adaptable to different compartmental modelling assumptions. In particular, our method can be readily extended to more complex compartmental models including hospitalization rates, testing rate and distinction between symptomatic and asymptomatic individuals. Thus, the methodology presented in the present study can be readily adapted to any province, state or country globally; making it a potentially useful tool for policy makers in event of future outbreaks or a relapse in the current one.

Finally, we have hosted our quarantine diagnosis results for the top 70 affected countries worldwide on a public platform (https://covid19ml.org/ or https://rajdandekar.github.io/COVID-QuarantineStrength/), which can be used for informed decision making by public health officials and researchers alike. We believe that such a publicly available global tool will be of significant value for researchers who want to study the correlation between the quarantine strength evolution in a particular region with a wide range of metrics spanning from mortality rate to socio-economic landscape impact of Covid-19 in that region.

Currently, our model lacks forecasting abilities. In order to do robust forecasting based on prior data available, the model needs to be further augmented through coupling with real-time metrics parameterizing social distancing, e.g. the publicly available Apple mobility data [4]. This could be the subject of future studies.

2.5 Experimental Procedures

Data and Code Availability

Data for the infected and recovered case count in all regions was obtained from the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University. All code files are available at https://github.com/RajDandekar/MIT-Global-COVID-Modelling-Project-1. All results are publicly hosted at https://covid19ml.org/ or https://rajdandekar.github.io/COVID-QuarantineStrength/.

2.5.1 Augmented QSIR Model: Initial Conditions

The starting point $t = 0$ for each simulation was the day at which 500 infected cases were crossed, *i.e.* $I_0 \approx 500$. The number of susceptible individuals was assumed to be equal to the population of the considered region. Also, in all simulations, the number of recovered individuals was initialized from data at $t = 0$ as defined above. The quarantined population $T(t)$ is initialized to a small number $T(t = 0) \approx 10$.

2.5.2 Augmented QSIR Model: Parameter estimation

The time resolved data for the infected, I_{data} and recovered, R_{data} for each locale considered is obtained from the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University. The neural network-augmented SIR ODE system was trained by minimizing the mean square error loss function

$$L_{\text{NN}}(W,\beta,\gamma,\delta) = ||\log(I(t)+T(t)) - \log(I_{\text{data}}(t))||^2 + ||\log(R(t)) - \log(R_{\text{data}}(t))||^2 \tag{2.13}$$

that includes the neural network's weights W. For most of the regions under consideration, W,β,γ,δ were optimized by minimizing the loss function given in (2.13). Minimization was employed using local adjoint sensitivity analysis [15, 128] following a similar procedure outlined in a recent study [129] with the ADAM optimizer [83] with a learning rate of 0.01. The iterations required for convergence varied based on the region considered and generally ranged from $40,000 - 100,000$. For regions with a low recovered count: all US states and UK, we employed a two stage optimization procedure to find the optimal W,β,γ,δ. In the first stage, (2.13) was minimized. For the second stage, we fix the optimal γ,δ found in the first stage to optimize for the remaining parameters: W,β based on the loss function defined just on the infected count as $L(W,\beta) = ||\log(I(t)+T(t)) - \log(I_{\text{data}}(t))||^2$. In the second stage, we don't include the recovered count $R(t)$ in the loss function, since $R(t)$ depends on γ,δ which have already been optimized in the first stage. By placing more emphasis on minimizing the infected count, such a two stage procedure leads to much more accurate model estimates; when the recovered data count is low. The iterations required for convergence in both stages varied based on the region considered and generally ranged from $30,000 - 100,000$.

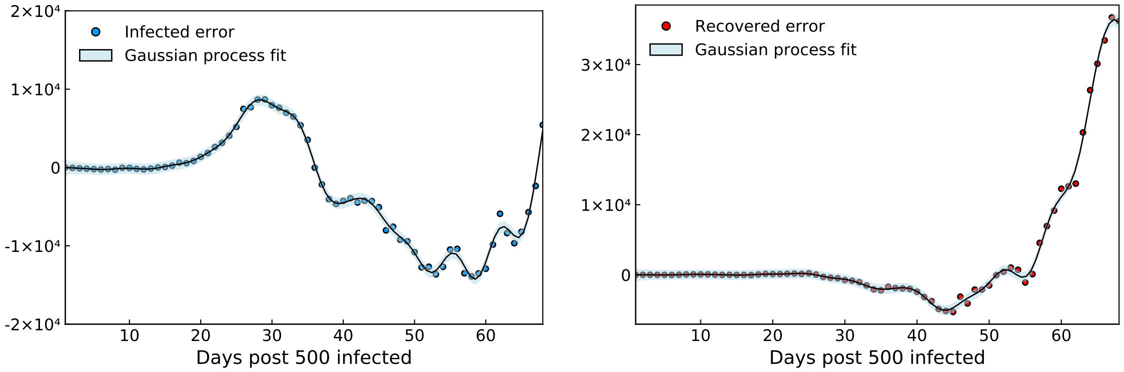


Figure 2-17: [Gaussian Process Residue Regression Model] Gaussian Process residue model fitted to (a) the infected case count and (b) the recovered case count for Russia.

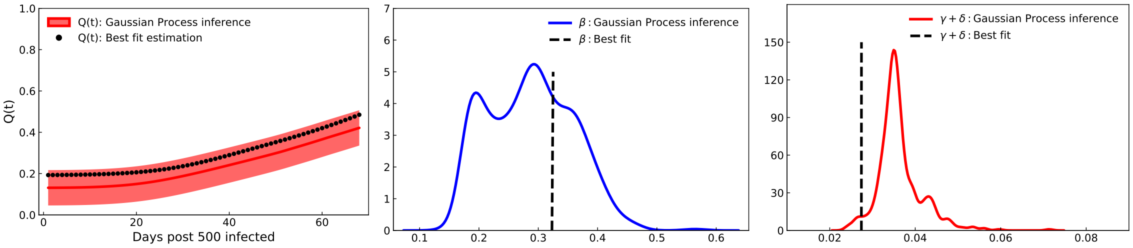


Figure 2-18: [Parameter Inference to demonstrate robustness of QSIR model recovered parameters] Inferred parameters for 500 realizations of the Gaussian process residue model superimposed on the best fit model prediction applied to Russia and shown for (a) the quarantine strength function $Q(t)$, (b) the contact rate β and the recovery rate $\gamma + \delta$. A total of 30 million iterations were performed on the MIT Supercloud cluster to generate parameter histograms for one country.

2.5.3 Parameter Inference: Gaussian Process residue model

In order to validate the robustness of the model and the uniqueness of the parameters recovered by the model, we consider a Gaussian Process residue model for uncertainty quantification. Gaussian Processes have emerged as a useful tool for regression, classification, clustering and uncertainty quantification [137, 166]. Gaussian Process regression can be viewed as a Bayesian inference problem where we want to recover the posterior for the regression function which best approximates the training data. The novelty of such an approach stems from using a the prior probability distribution over a function space rather than from a finite parametric system. Each realization of such a function is a multivariate normal distribution, which allows for exact estimation of the posterior distribution. The covariance underlying the function space distribution is specified by the kernel function. The kernel function affects the shape and noise of the resulting posterior distribution.

In the present study, we fit a Gaussian Process regression model between the error resulting from the best fit model (described in Section 3.2 and optimized using the method described in Section 5.2) and the data. For the prior over the function space, we use a mean of zero and variance described by a Squared Exponential Kernel with a lengthscale of 1 and a significantly high signal standard deviation of $O(10^4)$ which allows for noisy estimates of the posterior. Such a fitted model for the infected and recovered case count for Russia is shown in figure 2-17. It should be noted the recovered optimal posterior is not a deterministic function, but a distribution over function spaces. Subsequently, we sample 500 error residues from this model and superimpose them on the best fit predictions to simulate 500 samples of the infected and recovered case count data. Finally, we apply our model described in Section 3.2 and optimized using the method described in Section 5.2 for these samples.

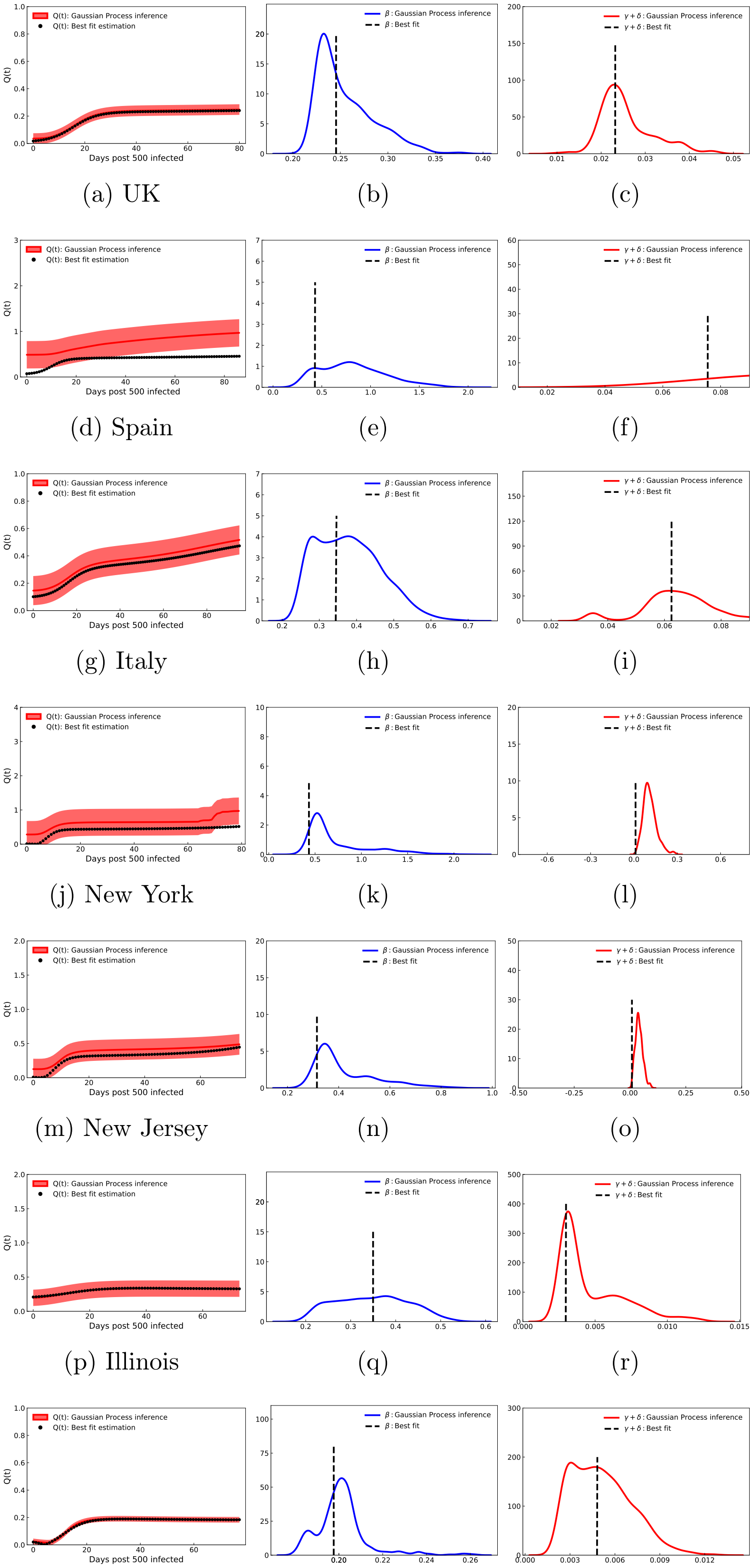
Figure 2-18 shows inferred parameters for 500 realizations of the Gaussian process residue model superimposed on the best fit model prediction applied to Russia and shown for (a) the quarantine strength function $Q(t)$, (b) the contact rate β and (c) the recovery rate $\gamma + \delta$. It can be seen that for all realizations, $Q(t)$ is seen to follow a similar behaviour, which lies close to the best fit model prediction. In addition, the inferred histograms for the contact rate β and the recovery rate $\gamma + \delta$ show a peak which is close to the best fit model prediction. This further validates the robustness of the model and strengthens the uniqueness of the parameters recovered by the model. Similar figures for all other countries are shown in the Supplementary Information.

A total of 30 million iterations were performed on the MIT Supercloud cluster to generate parameter histograms for one country.

2.6 Supplementary Information

2.6.1 Gaussian Process Residue Model for all regions

Figures 3-10, 2-20 shows inferred parameters for 500 realizations of the Gaussian process residue model superimposed on the best fit model prediction applied to Russia and shown for the quarantine strength function $Q(t)$, the contact rate β and the recovery rate $\gamma+\delta$ for highly affected regions of Europe, USA, Asia and South America till 1 June 2020; considered in the present study.



(a) UK (b) (c)

(d) Spain (e) (f)

(g) Italy (h) (i)

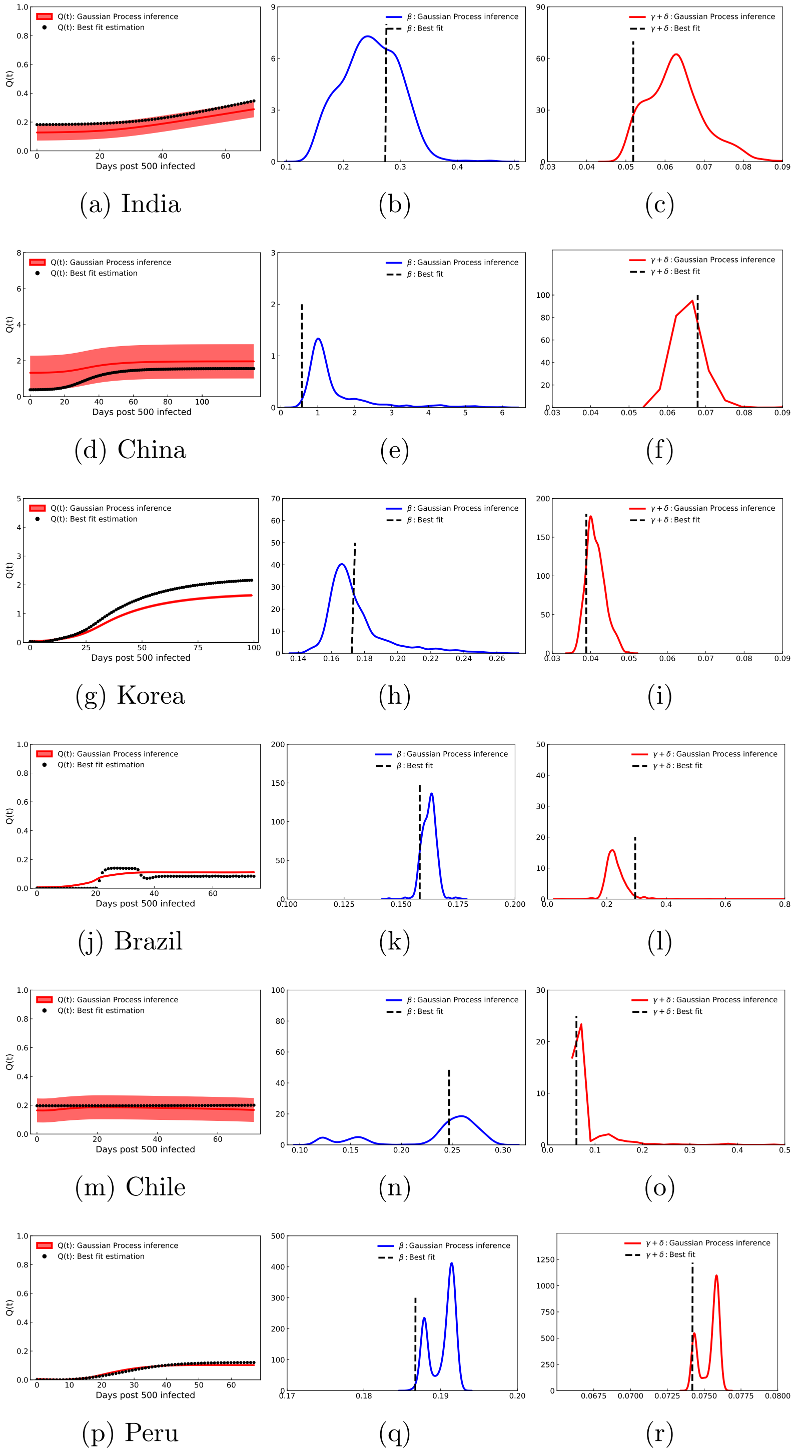
(j) New York (k) (l)

(m) New Jersey (n) (o)

(p) Illinois (q) (r)

(s) California (t) (u)

Figure 2-19: [Parameter Inference for Europe and USA] Inferred parameters for 500 realizations of the Gaussian process residue model superimposed on the best fit model prediction and shown for the quarantine strength function $Q(t)$ (left column), the contact rate β (middle column) and the recovery rate $\gamma + \delta$ (right column) for highly affected regions of Europe and USA till 1 June 2020; considered in the present study. A total of 30 million iterations were performed on the MIT Supercloud cluster to generate parameter histograms for each



(a) India (b) (c)

(d) China (e) (f)

(g) Korea (h) (i)

(j) Brazil (k) (l)

(m) Chile (n) (o)

(p) Peru (q) (r)

Figure 2-20: [Parameter Inference for Asia and South America] Inferred parameters for 500 realizations of the Gaussian process residue model superimposed on the best fit model prediction and shown for the quarantine strength function $Q(t)$ (left column), the contact rate β (middle column) and the recovery rate $\gamma+\delta$ (right column) for highly affected regions of Asia and South America till 1 June 2020; considered in the present study. A total of 30 million iterations were performed on the MIT Supercloud cluster to generate parameter histograms for each region.

Chapter 3

Effects of delayed reopening in Southern and West Central USA

3.1 Summary

In the wake of the rapid surge in the Covid-19 infected cases seen in Southern and West-Central USA in the period of June-July 2020, there was an urgent need to develop robust, data-driven models to quantify the effect which early reopening had on the infected case count increase. In particular, it was imperative to address the question: How many infected cases could have been prevented, had the worst affected states not reopened early? In this chapter, we use the QSIR model developed in the previous chapter, to show that the upsurge in the infected cases seen in these states is strongly co-related with a drop in the quarantine/lockdown strength diagnosed by our model. Further, our results demonstrate that in the event of a stricter lockdown without early reopening, the number of active infected cases recorded on 14 July 2020 could have been reduced by more than 40% in all states considered, with the actual number of infections reduced being more than $100,000$ for the states of Florida and Texas. As we continue our fight against Covid-19, our proposed model can be used as a valuable asset to simulate the effect of several reopening strategies on the infected count evolution; for any region under consideration.

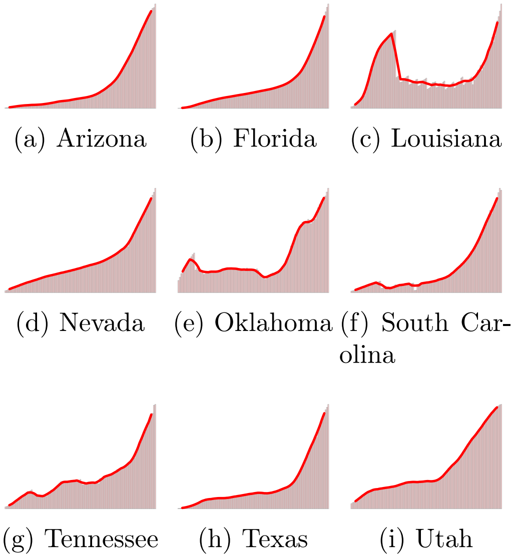

(a) Arizona
(b) Florida
(c) Louisiana
(d) Nevada
(e) Oklahoma
(f) South Carolina
(g) Tennessee
(h) Texas
(i) Utah

Figure 3-1: Active infected cases over time as of July 14, 2020, shown with a 7-day moving average, for the Southern and West-Central states considered in the present study.

3.2 Background

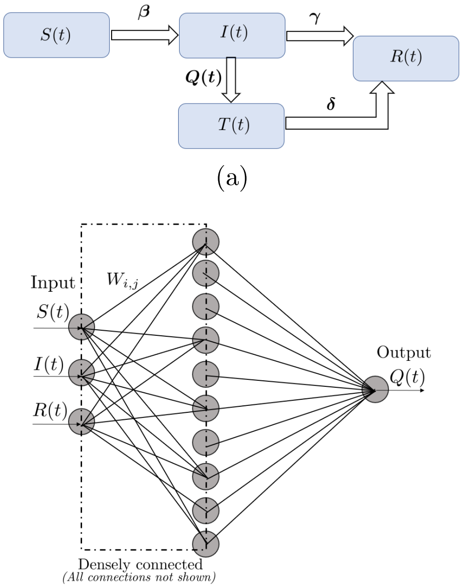
Since the second week of June 2020, a second surge of Covid-19 was seen in the United States [100], with rapidly increasing daily infected cases, hospitalization rates and death rates [10, 86]. Initially driven by disastrous situations in the states of Arizona, South Carolina, Texas, Florida and Georgia [100], the surge in cases was also later seen in several other Southern and West-Central states [41]. This surge can be seen in figure 3-1 which shows the active infected cases over time as of July 14, 2020 with a 7-day moving average for 9 states. States which reopened early show a generally strong co-relation with the rise in the infected cases over the 3-month period from late April to mid July 2020 [41]. For example, states which opened before May 15 showed daily infected case increments of: Florida (1393 %), Arizona (858 %), South Carolina (999 %), Alabama (547 %), Oklahoma (477 %), Tennessee (279 %), Georgia (245 %), Mississippi (215 %), Nevada (697 %), Texas (680 %) and Utah (287 %); while states which reopened after May 29 showed values of: Michigan (16 %), Pennsylvania (−26 %), New York (−52 %), New Jersey (−32 %) and Illinois (−54 %). Thus, although early reopening seems to be co-related to the second surge of cases

Table 3.1: Reopening details for different states considered in the present study

State	Reopening date	Reopening details
1. Arizona	May 15	June 17: Mask regulations strengthened, June 29: Partial reversal of reopening
2. Florida	May 4	June 3: Phase 2 of reopening
3. Louisiana	May 15	June 5: Phase 2 of reopening
4. Nevada	May 9	May 26: Phase 2 of reopening
5. Oklahoma	April 24	May 15: Phase 2 of reopening, June 1: Phase 3 of reopening
6. South Carolina	May 4	May 4: Stay at home order lifted, further facilities reopened till May 18
7. Tennessee	April 30	May 22: Phase 2 of reopening.
8. Texas	May 1	May 18: Phase 2 of reopening, June 3: Phase 3 of reopening
9. Utah	May 1	May 1: Gradual reopening

seen in the USA, there is a need for robust, data-driven quantification of the effect of early reopening on the growth of infected count data. More importantly, it is of utmost importance to answer the question: How many infected cases could have been prevented, had the worst affected states not reopened early?

In an effort to address this question, we have developed a machine learning-aided epidemiological model (the QSIR model). The novelty of our model arises from the fact that it allows us to decompose the contribution of quarantine/lockdown strength evolution to the infected data timeseries for the region under consideration. This enables us to simulate the effect of varying quarantine strength evolutions and hence varying reopening strategies on the infected count data. We define reopening as beginning when a state allows its stay-at-home order to expire, or, in the case of states that never issued a stay-at-home order, when a state first starts allowing non-essential businesses, such as dine-in restaurants and hair salons, to reopen [91, 35]. The reopening details for the states considered in the study are shown in table 3.1. Considering nine US states which showed a significant surge in cases since the last month, we demonstrate that our model shows a drop in the quarantine strength evolution when these states were reopened. Furthermore, we show that maintaining a strict lockdown without early reopening would have led to about $500,000$ fewer



(a)

(b)

Figure 3-2: (a) Schematic of the augmented QSIR model considered in the present study. (b) Schematic of the neural network architecture used to learn the quarantine strength function $Q(t)$.

infected cases in all these states combined.

3.3 Methods

3.3.1 QSIR Model

Standard SIR model

The SIR (Susceptible - Infected - Recovered) is governed by the following set of ODEs

$$\frac{dS}{dt} = -\frac{\beta\, S(t)\, I(t)}{N} \tag{3.1}$$

$$\frac{dI}{dt} = \frac{\beta\, S(t)\, I(t)}{N} - \gamma I(t) \tag{3.2}$$

$$\frac{dR}{dt} = \gamma I(t). \tag{3.3}$$

where β, γ are the contact and recovery rates respectively. We use this framework as our baseline model to be augmented with an neural network module. We do not consider the possibility of recovered individuals being reinfected [105]. We also do not consider the waning of immunity associated with Covid-19 as discovered in recent studies [22].

QSIR model: ODE formulation (same as Chapter 1, reader can skip)

The QSIR ODE model formulation is similar to the one studied previously [31], and is briefly explained in this section. The equations governing the QSIR model are as follows

$$
\begin{aligned}
\frac{dS}{dt} &= -\frac{\beta\, S(t)\, I(t)}{N} && (3.4)\\
\frac{dI}{dt} &= \frac{\beta\, S(t)\, I(t)}{N} - (\gamma + Q(t))\, I(t) \\
&= \frac{\beta\, S(t)\, I(t)}{N} - (\gamma + \mathrm{NN}(W, U))\, I(t) && (3.5)\\
\frac{dR}{dt} &= \gamma I(t) + \delta T(t) && (3.6)\\
\frac{dT}{dt} &= Q(t)\, I(t) - \delta T(t) = \mathrm{NN}(W, U)\, I(t) - \delta T(t). && (3.7)
\end{aligned}
$$

The SIR model is augmented by introducing a time varying quarantine strength rate term $Q(t)$ represented by a neural network [129] and a quarantined population $T(t)$, which is prevented from having any further contact with the susceptible population. Thus, the term $I(t)$ denotes the active infected population (Actively infected = Cumulative infected - Recovered) still having contact with the susceptibles, as done in the standard SIR model, while the term $T(t)$ denotes the infected population who are effectively quarantined and isolated.

Augmented QSIR Model: Initial Conditions

The starting point $t = 0$ for each simulation was the day at which 500 infected cases was crossed, *i.e.* $I_0 \approx 500$. The number of susceptible individuals was assumed to be equal to the population of the considered region. Also, in all simulations, the number of recovered individuals was initialized from data at $t = 0$ as defined above.

The quarantined population $T(t)$ is initialized to a small number $T(t = 0) \approx 10$.

Augmented QSIR Model: Parameter estimation

The data for the infected, recovered case counts was obtained from the publicly maintained repository by the Center for Systems Science and Engineering at Johns Hopkins University. The loss function is defined as

$$\begin{aligned} L_{\text{NN}}(W, \beta, \gamma, \delta) = & \\ & ||\log(I(t) + T(t)) - \log(I_{\text{data}}(t))||^2 \\ & + ||\log(R(t)) - \log(R_{\text{data}}(t))||^2 \end{aligned} \tag{3.8}$$

Parameter optimization for W, β, γ, δ was performed by minimizing the loss function defined in Equation 3.8 using the approach employed in prior studies [31, 15, 128] using an ADAM optimizer [83] with a learning rate of 0.01. For most of the states under consideration, W, β, γ, δ were optimized by minimizing the loss function given in (3.8). For states with a low recovered count: Arizona, Florida, Nevada and Texas, we employed a two stage optimization procedure to find the optimal W, β, γ, δ. In the first stage, (3.8) was minimized. For the second stage, we fix the optimal γ, δ found in the first stage to optimize for the remaining parameters: W, β based on the loss function defined just on the infected count as $L(W, \beta) = ||\log(I(t) + T(t)) - \log(I_{\text{data}}(t))||^2$. Such an approach was found to be optimal for analyzing low recovered count data in previous studies [31].

In all states considered in the present study, we trained the model using data starting from the dates when the 500th infection was recorded in each region and up to July 14, 2020. For each state considered, $Q(t)$ denotes the rate at which infected persons are effectively quarantined and isolated from the remaining population, and thus gives composite information about (a) the effective testing rate of the infected population as the disease progresses and (b) the intensity of the enforced quarantine as a function of time.

This QSIR ODE framework applied on the infected and recovered data is used to estimate the quarantine strength function Q(t) in a particular state as shown in the first and second columns of figure 3-3.

QSIR Model: SDE formulation (same as Chapter 1, reader can skip)

The ODE modelling framework described above is a deterministic approach to model transfer of species (here: people) from one compartment to another through different reaction channels. Such a deterministic approach ignores any random fluctuations during species transfer from one compartment to the other. To include such stochastic effects and thus get a measure of the model uncertainty, we note that the augmented SIR framework derives from the chemical master equation which descibes the time evolution of the probability of such a system of interacting species to be in a given state at a given time (details in Supplementary Information). Although the chemical master equation cannot be solved analytically, under certain conditions, it can be distilled down to a stochastic differential equation (SDE) which captures the fluctuations in species transfer as random walks. Such an SDE, also known as the Chemical Langevin Equation, is thus based on the underlying ODE framework (macroscopic picture) and also includes stochastic effects reminiscent of microscopic modelling. In fact, in the Supplementary Information, we show that the microscopic simulation, macroscopic ODE formulation and the Chemical Langevin Equation (which acts as a bridge between the two) are all equivalent to each other.

The equivalent stochastic formulation or the Chemical Langevin equation for the augmented SIR model is

$$dS = -\left[\frac{\beta\, S(t)\, I(t)}{N}\right] dt - \sqrt{\left[\frac{\beta\, S(t)\, I(t)}{N}\right]} dW_1(t) \tag{3.9}$$

$$\begin{aligned} dI = &\left[\frac{\beta\, S(t)\, I(t)}{N} - \gamma I(t) - Q(t) I(t)\right] dt \\ &+ \sqrt{\frac{\beta\, S(t)\, I(t)}{N}} dW_1(t) - \sqrt{\gamma I(t)} dW_2(t) - \sqrt{Q(t) I(t)} dW_3(t) \end{aligned} \tag{3.10}$$

$$dR = [\gamma I(t) + \delta T(t)]\, dt + \sqrt{\gamma I(t)} dW_2(t) + \sqrt{\delta T(t)} dW_4(t) \tag{3.11}$$

$$dT = [Q(t)\, I(t) - \delta T(t)]\, dt + \sqrt{Q(t) I(t)} dW_3(t) - \sqrt{\delta T(t)} dW_4(t) \tag{3.12}$$

In (3.9), $W_i(t) \sim N(0,t)$ is a normally distributed random variable with mean zero and variance t or $dW_i(t) \sim N(0,dt)$. It should also be noted that each $W_i(t)$ represents an independent Brownian motion. The simulations were performed using the Catalyst.jl software in Julia using the LambaEM algorithm based on [130]. 1000 trajectories were simulated for each state.

This QSIR SDE framework along with the simulated quarantine functions for no reopening is used to predict the new infected case count and hence estimate the reduction in the number of infected cases under the simulated no-reopening quarantine function. The results are shown as 5% and 95% quantiles in the third column of figure 3-3.

Mean Absolute Percentage Error

The Mean Absolute Percentage Error (MAPE) is defined as

$$\begin{aligned} &\text{MAPE} = \\ &\qquad \frac{100}{N} * \sum \frac{[I(t) + T(t) + R(t)] - [I_{\text{data}}(t) + R_{\text{data}}(t)]}{[I_{\text{data}}(t) + R_{\text{data}}(t)]} \end{aligned} \tag{3.13}$$

where N is the number of observations.

3.4 Results

The first stage of our analysis is using our model [31], called the QSIR model to diagnose the underlying quarantine strength evolution $Q(t)$ in the regions under consideration. By applying the QSIR model to more than 70 countries globally, we have established the validity of $Q(t)$ in accurately diagnosing the on-the-ground quarantine situation in majorly affected European, South American and Asian countries [31]. A slow growth of $Q(t)$ without a significant increase indicates relaxed quarantine policies, a sharp transition point in $Q(t)$ is indicative of a sudden ramp-up of quarantine measures, and an inflection point corresponds to the time when the quarantine response was the most rapid in the region under consideration. The results of our model applied globally to all continents are hosted publicly at covid19ml.org.

In this study, to perform the quarantine diagnosis to analyze the implications of delayed reopening, we applied the QSIR model to 9 US states which showed a significant surge in the infected case count in the last month: Arizona, Florida, Louisiana, Nevada, Oklahoma, South Carolina, Tennessee, Texas and Utah. Figure 3-3 shows representative results for Arizona, Nevada, South Carolina and Tennessee. The plots for the remaining states are provided in the Supplementary Information. Figures 3-3 a, d, g, j show the comparison of the infected and recovered count estimated by our model with the actual data. A reasonable agreement is seen for all states, with the model being able to capture the rise in infections seen in the tail end of the time-series. The QSIR model details are provided in the Methods section; Mean Absolute Percentage Error (MAPE) values for the model along with the epochs required for convergence for each state are provided in Supplementary Information.

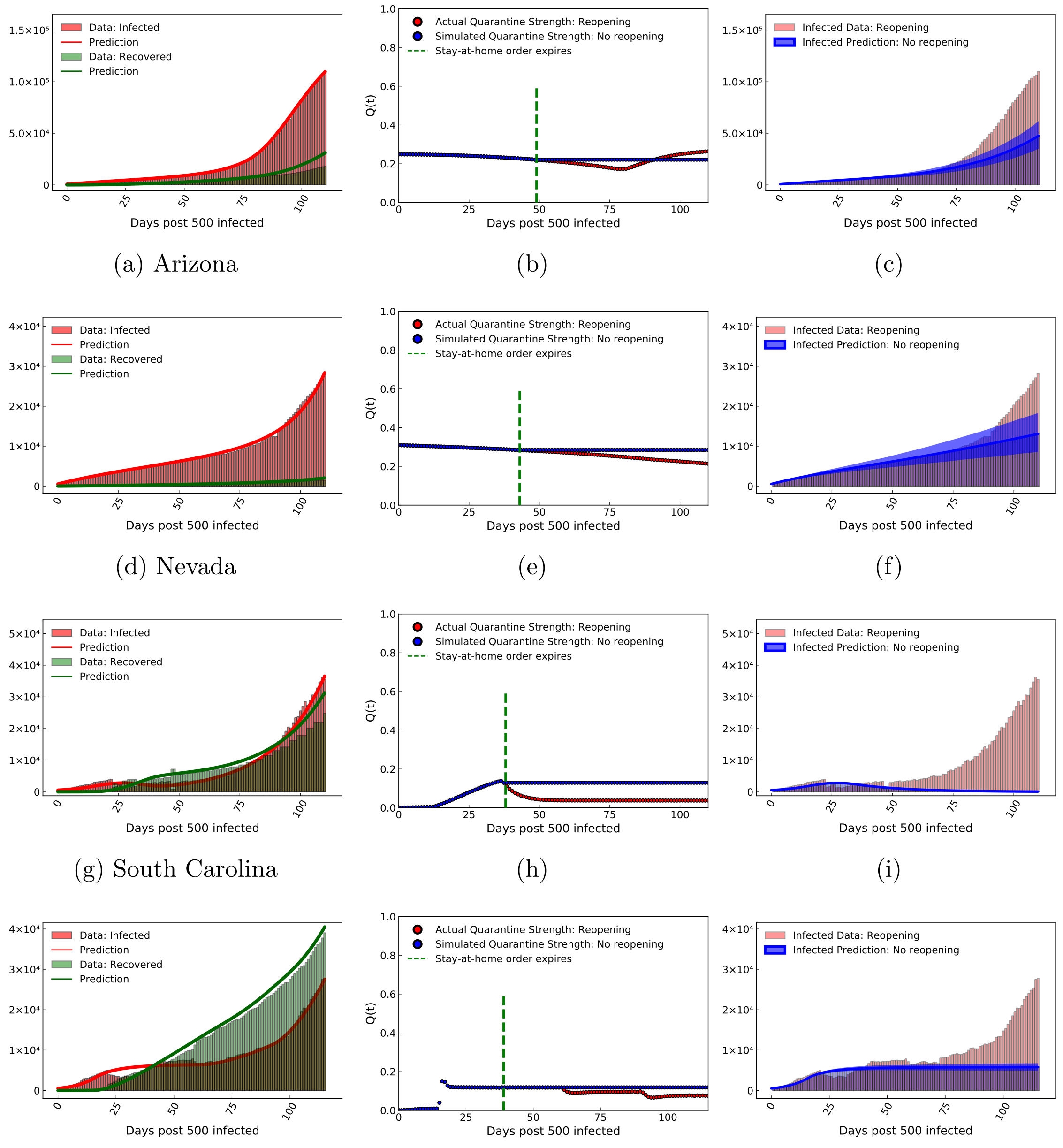
Figures 3-3 b, e, h, k show the quarantine strength evolution $Q(t)$ as learnt by the neural network module, which shows a decline whose starting point corresponds well to the time when these states began reopening, as seen from table 3.1 and the green dotted line in the figures 3-3 b, e, h, k. In some states, the decline in $Q(t)$ starts later than the reopening date; possibly corresponding to the Phase 2 or Phase 3 of reopening (table 3.1) or because of the time delay for population level changes to be seen in the infected count evolution, after reopening. $Q(t)$ trained by our model shows a significant drop after early reopening in all Southern and West-Central states that showed a surge in cases last month; whereas the North-Eastern states of New York, New Jersey and Illinois, which reopened late and showed no surge in infections, did not show a drop in $Q(t)$ (Table 3.2 and figures in Supplementary Information). Thus, the upsurge in the infected cases seen in these states is strongly co-related with a drop in the quarantine/lockdown strength $Q(t)$ diagnosed by our model. This is indicative of two things: (a) the Southern and West-Central states reopened early, which led to a relaxed imposition of quarantine/lockdown measures in these states and consequently a surge in infections was seen, and (b) the North-Eastern states of New York, New Jersey and Illinois reopened late, and even after reopening, a relatively low contact rate was maintained amongst the population, leading to a relatively high magnitude of the imposed quarantine strength, which prevented a surge of infections in these states.

After confirming that our model is able to accurately depict the co-relation between the surge in infections and early reopening in these states through the diagnosed $Q(t)$, we proceed to the second stage of our analysis. In the second stage, we use the diagnosed $Q(t)$ to address the question: How many infected cases would have been reduced, had the worst affected states not reopened early? To answer this question, we simulate the "no-reopening" strategy by assuming that $Q(t)$ is maintained at the value it was before reopening, without decreasing. This simulated $Q(t)$ is shown in Figures 3-3 b, e, h, k. The flexibility of our model allows us to run our model with this simulated $Q(t)$ for all states considered. To quantify the aleatory uncertainty resulting from random fluctuations in the model, we utilized the chemical Langevin equation extension to the QSIR model whose definition and justification is described in the Methods and Supplemental Information section. This allows us to estimate bootstrapped confidence intervals resulting from 1000 simulations of such a stochastic

Table 3.2: Drop in quarantine strength function, $Q(t)$ after reopening as discovered by our trained model. $Q(t)$ trained by our model shows a significant drop for all Southern and West-Central states which showed a surge in cases from reopening; whereas the North-Eastern states which showed no surge don't see a drop in $Q(t)$.

State	Reopening date	% increase in daily cases since reopening	Maximum % decrease in $Q(t)$ after reopening
1. Arizona	May 15	+858	+22
2. Florida	May 4	+1393	+10
3. Louisiana	May 15	+193	+30
4. Nevada	May 9	+697	+25
5. Oklahoma	April 24	+477	+29
6. South Carolina	May 4	+999	+71
7. Tennessee	April 30	+279	+44
8. Texas	May 1	+680	+29
9. Utah	May 1	+287	+39
10. New York	May 29	−52	−45
11. New Jersey	June 9	−32	−60
12. Illinois	May 29	−54	−8

model, and thus quantify the effect of such a "no-reopening policy" on the epidemic spread. The infected count evolution for the simulated $Q(t)$ without reopening is shown in Figures 3-3 c, f, i, l (5% and 95% quantiles are shown). We can see that, for all these states, instead of seeing a spike in infections, we would have seen a plateau in the infected case count evolution. The number and the percentage of infected cases that would have been prevented by July 14 had these states not reopened are shown in Table 3-3. It is evident that the number of infections could have been reduced by more than 40% in all states considered, with the actual number of infections reduced being more than $100,000$ for the states of Florida and Texas. Even the less populated states of Louisiana, South Carolina and Tennessee show mean infected case reduction values of $44\%, 84\%$ and 47% respectively, which correspond to $36,000, 51,000$, and $31,000$ infected cases reduced.



(a) Arizona (b) (c)

(d) Nevada (e) (f)

(g) South Carolina (h) (i)

(j) Tennessee (k) (l)

Figure 3-3: For the states of Arizona, Nevada, South Carolina and Tennessee, figure shows: (a, d, g, j) Model recovery of infected and recovered case count as of 14 July, 2020. (b, e, h, k) Quarantine strength function as discovered by our trained model (with reopening). This is shown along with the quarantine strength function which we use to simulate strict quarantine without reopening after stay-at-home order was imposed. (c, f, i, l) Estimated infected count if strict quarantine and lockdown measures were followed without reopening (5% and 95% quantiles are shown) as compared to the values corresponding to the actual early reopening scenario.

Table 3.3: Infected count reduction by 14 July, 2020, if states had not reopened early, as estimated by our model.

State	% decrease (5% - 95% quantiles)	Mean % decrease	Case reduction	Mean case case reduction
1. Arizona	35 – 62	49	44000 – 79000	63000
2. Florida	20 – 75	49	57000 – 218000	144000
3. Louisiana	37 – 50	44	31000 – 41000	36000
4. Nevada	32 – 68	51	10000 – 20000	15000
5. Oklahoma	46 – 69	58	10000 – 15000	13000
6. South Carolina	83 – 86	84	50000 – 52000	51000
7. Tennessee	41 – 53	47	27000 – 36000	31000
8. Texas	41 – 51	46	115000 – 143000	129000
9. Utah	35 – 47	41	11000 – 14000	12000

3.5 Conclusion

In this study, we have developed a novel methodology to quantify the effect of early reopening on the infected case count surge seen during the period of June-July 2020. We have proposed a machine learning model, called the QSIR model, rooted firmly in fundamental epidemiology principles which has the following attributes: (a) it is highly interpretable with few free parameters rooted in an epidemiological model, (b) it relies on only Covid-19 data and not on previous epidemics and (c) it can decompose the infected timeseries data to reveal the quarantine strength/policy variation, $Q(t)$, in the region under consideration. To demonstrate the validity of our model in capturing the actual quarantine policy evolution in a particular region, the model has been applied to 70 countries globally. The quarantine strength behaviour learnt from the model accurately mimics the on-the-ground situation in majorly affected European, South American and Asian continents. The results for this global analysis are hosted at covid19ml.org [31].

After confirming our belief in the model through a global analysis, we apply the model to the Southern and West-Central US states which have shown a massive surge in Covid-19 infected cases since June 2020. We demonstrate that the $Q(t)$ extracted by our model shows a significant drop in value for the Southern and West-Central states which reopened early and showed a surge in infections. The time at which $Q(t)$ starts to decline generally agrees well with the reopening date for the states considered. Since the decline in $Q(t)$ is strongly co-related to the surge of infections and also the reopening date for states which reopened early, we can then simulate the effect of "no-reopening" by maintaining the $Q(t)$ at a constant level after reopening, instead of declining. We show that maintaining a steady imposition of quarantine/lockdown control would have played a massive role in bringing down the infected count by more than 40% in all states considered, with the infections reduced reaching more than $100,000$ for the states of Florida and Texas.

We have proposed a novel machine learning methodology, rooted in fundamental epidemiological models; which is able to recover the real time quarantine strength evolution for any region under consideration. As the pandemic evolves and we continue our fight against Covid-19; and for future outbreaks, our globally applicable methodology can be a valuable asset for researchers and policy makers to simulate several reopening strategies, counterfactual scenarios and analyze their impact on the infected count evolution. Our findings highlight that as we continue the fight against Covid-19, it is imperative to reduce the contact between susceptible and infected individuals in public places by formulating robust safety guidelines. Such guidelines implemented and maintained in the affected states would ensure a high level of quarantine strength associated with that state and can prevent a future surge or wave in the Covid-19 infected count timeseries.

The results of our model should be taken in the context of its assumptions. Ideally, one needs to consider the shifting US testing policies for the time period under consideration. Since the testing efforts did not show a significant increase during and after the reopening in the US states in the time period considered within the present study [57, 142] and we did not want to burden our model with additional parameters to fit; testing compartments have not been included in the present study. Additionally, several studies in literature [111, 132, 89, 153] have attempted to incorporate underreporting of infected/recovered cases in their modelling paradigm. Most of these studies use previously known estimates of testing data, serology data or Infection-Fatality-Rate(IFR). In these studies involving multiple parameters, a number of parameters are assumed to fixed at the start of the simulation from prior studies. These parameters include and are not limited to: time between onset of infections and symptoms, transmission duration, rate at which hospitalized patients recover [153], mean duration from symptom onset to recovery [111] or even the IFR ratio [111]. A second class of studies uses antibody testing from collected serum samples to estimate the actual number of infected cases [60].

As the pandemic unfolds and starts spreading, the first information available is the number of infected, recovered and deaths (for example: the Johns Hopkins public repository for Covid-19 tracking). Unless we have serum sample data information or we can confidently rely on prior studies for assessment of certain parameters, accurate information of the underreporting factor is difficult to obtain in real time. One of the goals of the present modelling methodology is to assist researchers and policy makers with quarantine diagnosis information in real time, with no reliance on parameters derived from prior studies.

Finally, the model is based on the SIR framework, which assumes a constant, age-independent contact and recovery rate between the infected and susceptible populations. Additionally, we do not consider the spatial heterogeneity in the infected count within a particular state and assume the governing dynamics to be only time-dependent. Consideration of these second-order aspects would further refine the model and would be the subject of future studies.

3.6 Supplementary Information

Model-diagnosed quarantine strength for North-Eastern US states

Figure 3-4 shows the application of the model to the north-eastern states of New York, New Jersey and Illinois along with the diagnosed quarantine strength function $Q(t)$ for these states. These states do not show a decline in $Q(t)$. This corresponds well to the delayed reopening and generally stronger quarantine measures employed in the North-Eastern US states. Since $Q(t)$ does not decrease, these states did not show a surge in infections starting June 2020, unlike their Southern and West-Central counterparts. The difference in these results between the North-Eastern and Southern, West-Central states indicates two things: (a) it strengthens the validity of our proposed model in capturing the real-time reopening scenario in different states through the evolution of the diagnosed $Q(t)$, and, more importantly, (b) it further validates the role played by early reopening in reducing $Q(t)$ and subsequently leading to a surge of new infected cases in the Southern and West-Central US states.

Impact of early reopening on the states of Louisiana, Florida, Oklahoma, Texas and Utah

Figure 3-5, 3-6 implements a similar analysis to study the effect of early reopening for the states of Louisiana, Nevada, Oklahoma, Texas and Utah, as done for the states of Arizona, Nevada, South Carolina and Tennessee. Similar to the states considered in the main text, we see that all of these states show a decline in $Q(t)$ starting around the time when these states were reopened. If these states were not reopened early, a large number of infections would have been reduced as demonstrated in Table 1 of the main text.

Equivalence between the ODE model and the Chemical Langevin SDE model

This analysis heavily borrows from the pioneering work done by Gillespie [46]. In this section, we will establish that the deterministic ODE model and the stochastic Chemical Langevin equation originate from a common expression: the chemical master equation [99], and are closely linked to one another. Following is the notation we will use, in accordance with [46] We consider N compartments: $S_1, S_2 \dots S_N$ and R reaction channels: $R_1, R_2 \dots R_M$ in a fixed volume Ω. In our case, we have $N = 4$ (S, I, R, T) compartments and $R = 4$ reaction channels. We denote the dynamical state of the system at any time t as $X(t) = (X_1(t), X_2(t) \dots X_N(t))$ where

- $X_i(t)$: total number of S_i molecules (in our case: individuals) in the system.
- Propensity function $a_j(x)dt$: probability that a reaction R_j will occur somewhere in Ω in the next time interval [t, t+dt] for $j = 1, 2 \dots M$.
- State change vector ν_j whose ith component is defined by $\nu_{j,i}$: change in the number of S_i molecules produced by one R_j reaction for $i = 1, 2 \dots N$, $j = 1, 2 \dots M$. In our case $\nu_{j,i} = \pm 1$.

From the definition of $a_j(x)dt$, we can write the probability of the system being in state x at time $t + dt$ (we take the sum of all mutually exclusive ways either through one reaction or no reaction in [t, t+dt]):

$$P(x, t + dt|x_0, t_0) = P(x, t|x_0, t_0) \left[1 - \sum_{j=1}^{M} a_j(x)dt \right] + \sum_{j=1}^{M} \left[P(x - \nu_j, t|x_0, t_0) a_j(x - \nu_j)dt \right], \tag{3.14}$$

Taking the limit of (3.14) as dt -> 0 leads to the **chemical master equation**

$$\frac{\partial P(x, t|x_0, t_0)}{\partial t} = \sum_{j=1}^{M} \left[a_j(x - \nu_j) P(x - \nu_j, t|x_0, t_0) - a_j(x) P(x, t|x_0, t_0) \right] \tag{3.15}$$

.

Macroscopic picture: Deterministic model relation to the chemical master equation:

Multiplying the chemical master equation (3.15) by x_i and summing over all x, we obtain for the mean of $X_i(t)$

$$\frac{d\langle X_i(t)\rangle}{dt} = \sum_{j=1}^{M} \nu_{ji}\langle a_j(X(t))\rangle \qquad (i = 1, 2 \ldots N) \tag{3.16}$$

Thus, whenever fluctuations are not important, the species populations evolve deterministically according to the following set of ordinary differential equations

$$\frac{dX_i(t)}{dt} = \sum_{j=1}^{M} \nu_{ji} a_j(X(t)) \qquad (i = 1, 2 \ldots N) \tag{3.17}$$

(3.17) is the basis for the classical SIR epidemiological equations, and we see how they evolve from the chemical master equation (3.15).

$$\begin{aligned}
\frac{dS}{dt} &= -\frac{\beta\, S(t)\, I(t)}{N} && (3.18)\\
\frac{dI}{dt} &= \frac{\beta\, S(t)\, I(t)}{N} - (\gamma + Q(t))\, I(t) &&\\
&= \frac{\beta\, S(t)\, I(t)}{N} - (\gamma + \mathrm{NN}(W, U))\, I(t) && (3.19)\\
\frac{dR}{dt} &= \gamma I(t) + \delta T(t) && (3.20)\\
\frac{dT}{dt} &= Q(t)\, I(t) - \delta T(t) = \mathrm{NN}(W, U)\, I(t) - \delta T(t). && (3.21)
\end{aligned}$$

The ODE system used in the present study shown in (5-8), is of the form (3.17).

Microscopic picture: Stochastic Simulation Algorithm and its relation to the master equation:

Another consequence of the master equation (3.15) is the existence and form of the next-reaction density function $p(\tau, j|x, t)$, which is defined as

- $p(\tau, j|x, t)d\tau$ = probability that given $X(t) = x$, the next reaction in Ω will occur in $[t + \tau, t + \tau + d\tau]$, and will be an R_j reaction

Since $\sum_j a_j(x)dt$ is the probability that some reaction occurs in the time interval dt, the probability that a time interval τ is spent without any reaction occuring is given by the exponential distribution: $\mathrm{Exp}(\sum_j a_j(x)\tau)$. Thus, we obtain for $p(\tau, j|x, t)$

$$p(\tau, j|x, t) = a_j(x)\mathrm{Exp}\left(\sum_{k=1}^{M} a_k(x)\tau\right) \qquad (0 \le \tau < \infty; j = 1, 2 \ldots M) \tag{3.22}$$

(3.22) is the basis for the stochastic simulation algorithm in which Monte-Carlo techniques are used to construct unbiased realizations of the process $X(t)$. A typical algorithm for stochastic simulation of this kind, is the Gillespie Algorithm [45] which can be viewed as a discrete space continuous time Markov jump process, with exponentially distributed jump times.

Chemical Langevin Equation: Bridging the gap between macroscopic and microscopic models:

Let the state of the system $X(t)$ at the current time t be x_t. Let $K_j(x_t, \tau)$ be the number of R_j reactions that occur in the time interval [t, t+dt]. Thus, the number of S_i molecules in the system at time $t + \tau$ will be

$$X_i(t + \tau) = x_{ti} + \sum_{j=1}^{M} K_j(x_t, \tau)\nu_{ji} \qquad (i = 1, 2 \ldots N) \tag{3.23}$$

[46] approximated K_j by imposing the following conditions

- **Condition 1: No propensity function change** This condition requires τ to be small enough so that none of the propensity functions $a_j(x)$ change noticeably. The propensity functions then satisfy

$$a_j(X(t')) \approx a_j(x_t) \qquad \forall t \in [t, t+\tau], \forall j \in [1, M] \tag{3.24}$$

Due to this condition, $K_j(x_t, \tau)$ will be a statistically independent Poisson random variable $P_j(a_j(x_t), \tau)$. Thus (3.23) simplifies to

$$X_i(t+\tau) = x_{ti} + \sum_{j=1}^{M} \nu_{ji} P_j(a_j(x_t), \tau) \qquad (i = 1, 2 \ldots N) \tag{3.25}$$

- **Condition 2: Large number of reaction occurrences:** This condition requires τ to be large enough so that the expected number of occurrences of each reaction channel R_j in $[t, t+\tau]$ is much larger than 1. Thus

$$\langle P_j(a_j(x_t), \tau) \rangle = a_j(x_t)\tau \gg 1, \qquad \forall j \in [1.M]. \tag{3.26}$$

This condition enables us to approximate each Poisson variable $P_j(a_j(x_t), \tau)$ by a normal random variable with the same mean and variance.

Thus, (3.25) further simplifies to

$$X_i(t+\tau) = x_{ti} + \sum_{j=1}^{M} \nu_{ji} N_j(a_j(x_t)\tau, a_j(x_t)\tau) \qquad (i = 1, 2 \ldots N) \tag{3.27}$$

where $N(m, \sigma^2)$ denotes the normal random variable with mean m and variance σ^2. Using $N(m, \sigma^2) = m + \sigma N(0, 1)$, denoting the time interval τ by dt and the unit normal random variable $N_j(0, 1)$ as $N_j(t)$, we obtain

$$X_i(t+dt) = X_i(t) + \sum_{j=1}^{M} \nu_{ji} a_j(X(t))dt + \sum_{j=1}^{M} \nu_{ji} a_j^{1/2}(X(t)) N_j(t)(dt)^{1/2} \qquad (i = 1, 2 \ldots N) \tag{3.28}$$

(3.28) can be written as a stochastic differential equation as

$$\frac{dX_i(t)}{dt} = \sum_{j=1}^{M} \nu_{ji} a_j(X(t)) + \sum_{j=1}^{M} \nu_{ji} a_j^{1/2}(X(t))\Gamma_j(t) \tag{3.29}$$

where $\Gamma_j(t)$ are temporally uncorrelated, statistically independent Gaussian white noise processes.

(3.29) is the Langevin equation, and it derives from the master equation provided that Condition 1 and Condition 2 are satisfied.

The Langevin equation (3.29) form of the ODE system (5-8) leads to the stochastic differential equation used in the current study

$$dS = -\left[\frac{\beta\, S(t)\, I(t)}{N}\right] dt - \sqrt{\left[\frac{\beta\, S(t)\, I(t)}{N}\right]} dW_1(t) \tag{3.30}$$

$$dI = \left[\frac{\beta\, S(t)\, I(t)}{N} - \gamma I(t) - Q(t) I(t)\right] dt + \sqrt{\frac{\beta\, S(t)\, I(t)}{N}} dW_1(t) - \sqrt{\gamma I(t)} dW_2(t) - \sqrt{Q(t) I(t)} dW_3(t) \tag{3.31}$$

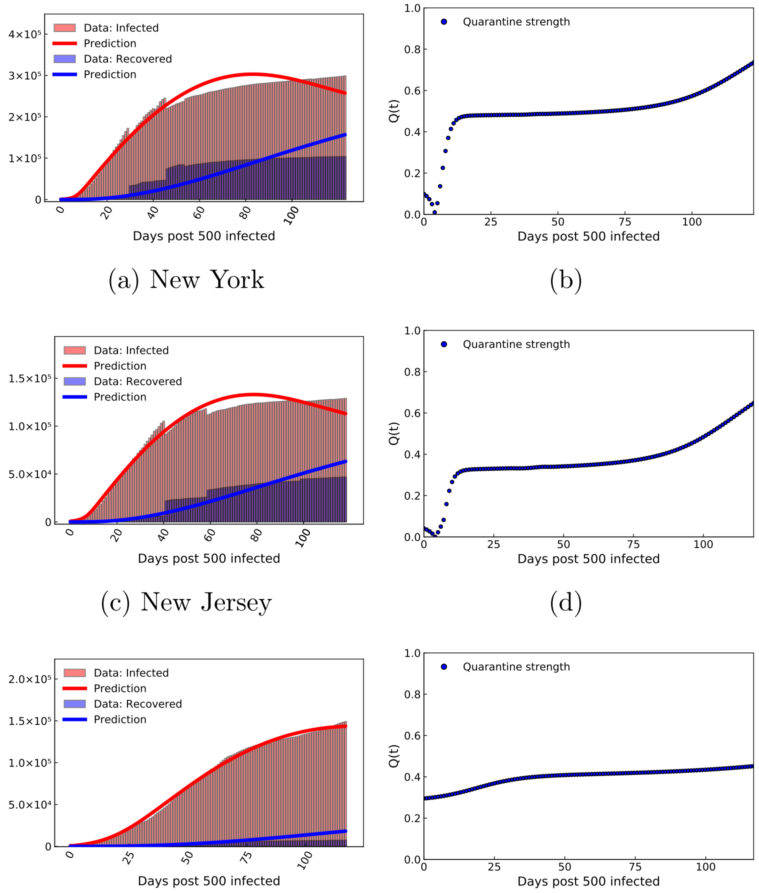
$$dR = \left[\gamma I(t) + \delta T(t)\right] dt + \sqrt{\gamma I(t)} dW_2(t) + \sqrt{\delta T(t)} dW_4(t) \tag{3.32}$$

$$dT = \left[Q(t)\, I(t) - \delta T(t)\right] dt + \sqrt{Q(t) I(t)} dW_3(t) - \sqrt{\delta T(t)} dW_4(t) \tag{3.33}$$

In (3.30), $W_i(t) \sim N(0,t)$ is a normally distributed random variable with mean zero and variance t or $dW_i(t) \sim N(0, dt)$. It should also be noted that each $W_i(t)$ represents an independent Brownian motion.

Comparison of the macroscopic, microscopic and Langevin SDE model for our study

Figure 3-7a shows that the microscopic Stochastic Simulation Gillespie Algorithm and the ODE model presented in Equation (6-9) in the main text show a good agreement with each other. Figure 3-7b shows the comparison of the Chemical Langevin SDE model shown in (3.30) ran for 1000 trajectories and the ODE model; which also show a good agreement. Thus, we have shown the equivalence between the microscopic, macroscopic and the Chemical Langevin model for our study. This equivalence allows



(a) New York

(b)

(c) New Jersey

(d)

(e) Illinois

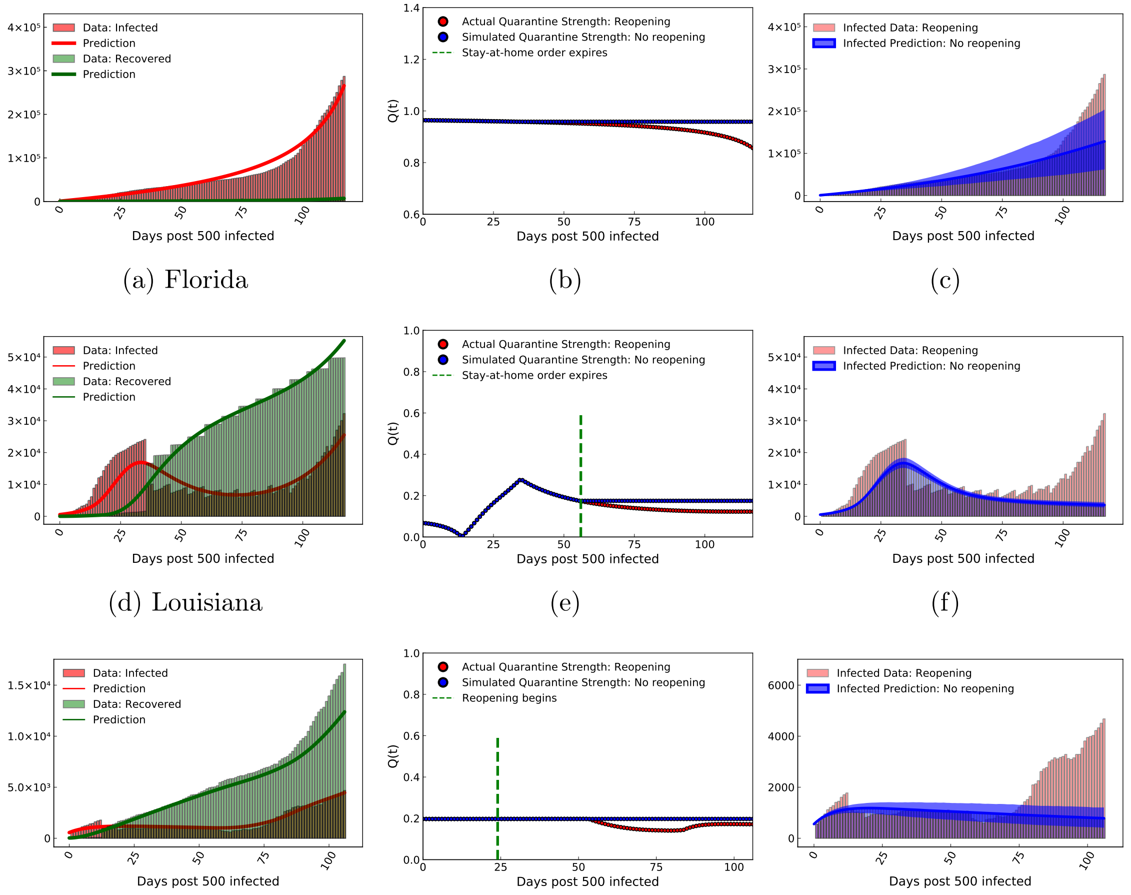
(f)

Figure 3-4: For the states of New York, New Jersey and Illinois, figure shows: (a, c, e) Model recovery of infected and recovered case count trained until 14 July, 2020. (b, d, f) Quarantine strength function as discovered by our trained model

us to add fluctuating components to the standard deterministic SIR model as shown in (3.30) and quantify the uncertainty resulting from these fluctuations.

Model specifications for each state

Table 3.4 shows the Model Mean Absolute Percentage Error (MAPE), epochs needed for convergence and number of parameters optimized for the different states considered.



(a) Florida

(b)

(c)

(d) Louisiana

(e)

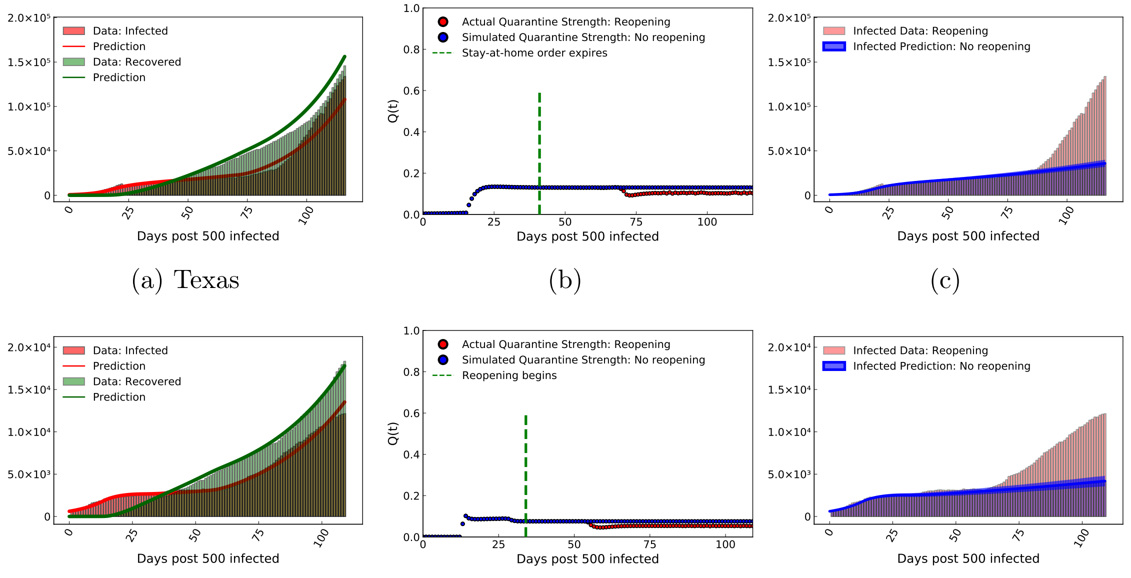
(f)

(g) Oklahoma

(h)

(i)

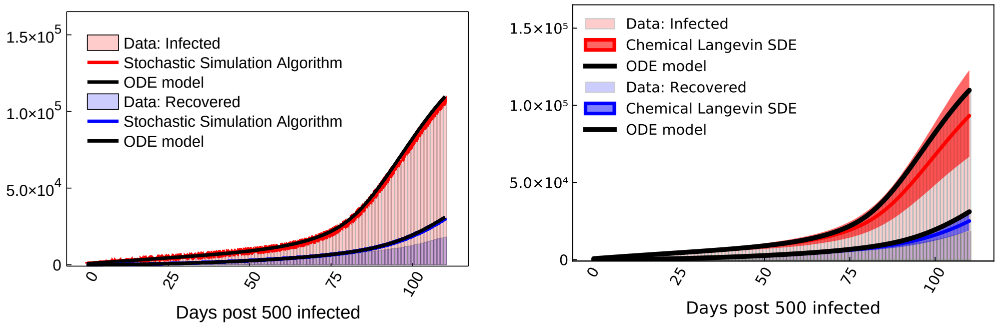
Figure 3-5: For the states of Louisiana, Nevada and Oklahoma: (a, d, g) Model recovery of infected and recovered case count as of 14 July, 2020. (b, e, h) Quarantine strength function as discovered by our trained model (with reopening). This is shown along with the quarantine strength function which we use to simulate strict quarantine without reopening after stay-at-home order was imposed. (c, f, i) Estimated infected count if strict quarantine and lockdown measures were followed without reopening as compared to the values corresponding to the actual early reopening scenario.



(a) Texas (b) (c)

(d) Utah (e) (f)

Figure 3-6: For the states of Texas and Utah: (a, d) Model recovery of infected and recovered case count as of 14 July, 2020. (b, e) Quarantine strength function as discovered by our trained model (with reopening). This is shown along with the quarantine strength function which we use to simulate strict quarantine without reopening after stay-at-home order was imposed. (c, f) Estimated infected count if strict quarantine and lockdown measures were followed without reopening as compared to the values corresponding to the actual early reopening scenario.



(a) Arizona (b)

Figure 3-7: (a) Comparison of the microscopic Stochastic Simulation Gillespie Algorithm and the ODE model presented in Equation (6-9) in the main text. (b) Comparison of the Chemical Langevin SDE model shown in (3.30) ran for 1000 trajectories ($5\% and 95\%$ quantiles are shown) and the ODE model.

Table 3.4: Mean Absolute Percentage Error (MAPE) values are shown along with the number of epochs required for and the number of parameters optimized, for all states considered.

State	Model MAPE	Epochs	Parameters optimized
1. Arizona	5.4%	10^5	54
2. Florida	18.7%	10^5	54
3. Louisiana	12%	12^5	54
4. Nevada	3.14%	18^5	54
5. Oklahoma	7.9%	12^5	54
6. South Carolina	11.7%	12^5	54
7. Tennessee	6.9%	12^5	54
8. Texas	10.4%	24^5	54
9. Utah	3.79%	12^5	54

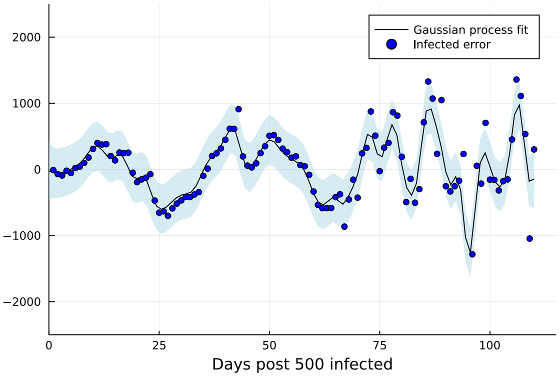


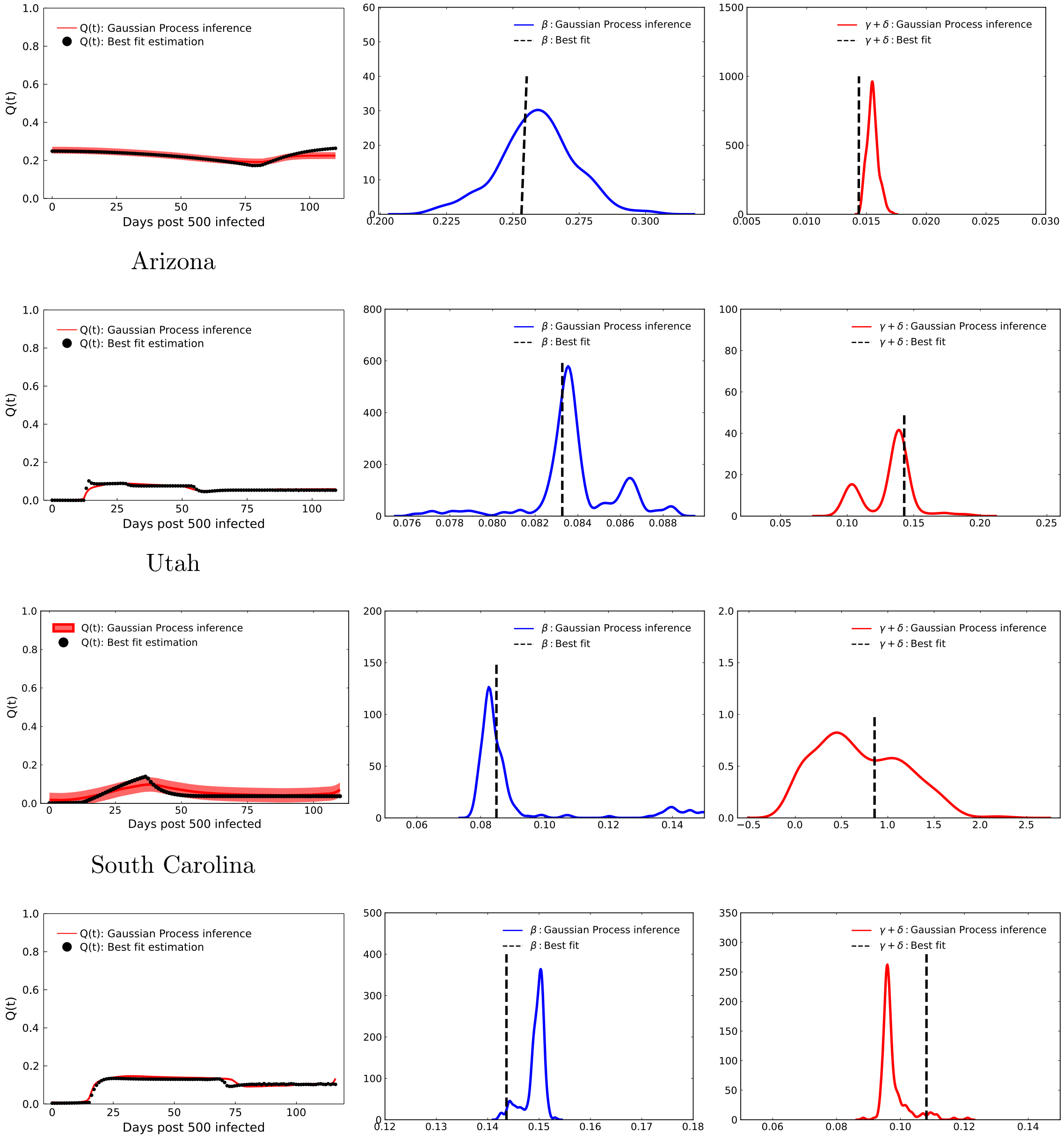
Figure 3-8: [Gaussian Process Residue Regression Model] Gaussian Process residue model fitted to the infected case count shown for Arizona.

Parameter Inference: Gaussian Process Residue Model

In order to validate the robustness of the model and the uniqueness of the parameters recovered by the model, we consider a Gaussian Process residue model for uncertainty quantification. Gaussian Processes have emerged as a useful tool for regression, classification, clustering and uncertainty quantification [137, 166].

In the present study, we fit a Gaussian Process regression model between the error resulting from the best fit model and the infected data. For the prior over the function space, we use a mean of zero and variance described by a Squared Exponential Kernel with a lengthscale of 1 and a significantly high signal standard deviation of $O(10^4)$ which allows for noisy estimates of the posterior. Such a fitted model for the infected count for a region under consideration (Arizona), is shown below in figure 3-8. Subsequently, we sample 500 error residues from this model and superimpose them on the best fit predictions to simulate 500 samples of the infected case count data. Finally, we apply our model described on these 500 samples of data, and recover the parameters $Q(t), \beta, \gamma, \delta$ from each of them.

Figures 3-9, 3-10 shows the inferred parameters for 500 realizations of the Gaussian process residue model superimposed on the best fit model prediction applied to all states considered, and shown for (a) the quarantine strength function $Q(t)$, (b) the contact rate β and (c) the recovery rate $\gamma + \delta$. It can be seen that for all realizations, $Q(t)$ is seen to follow a similar behaviour, which lies close to the best fit model prediction. In addition, the inferred histograms for the contact rate β and the recovery rate $\gamma + \delta$ show a peak which is close to the best fit model prediction. This further validates the robustness of the model for other regions considered and strengthens the uniqueness of the parameters recovered by the model. A total of 12 million iterations (60000 iterations for each realization of the Gaussian process residue model × 500 realizations) were performed on the MIT Supercloud cluster to generate parameter histograms for each state considered.



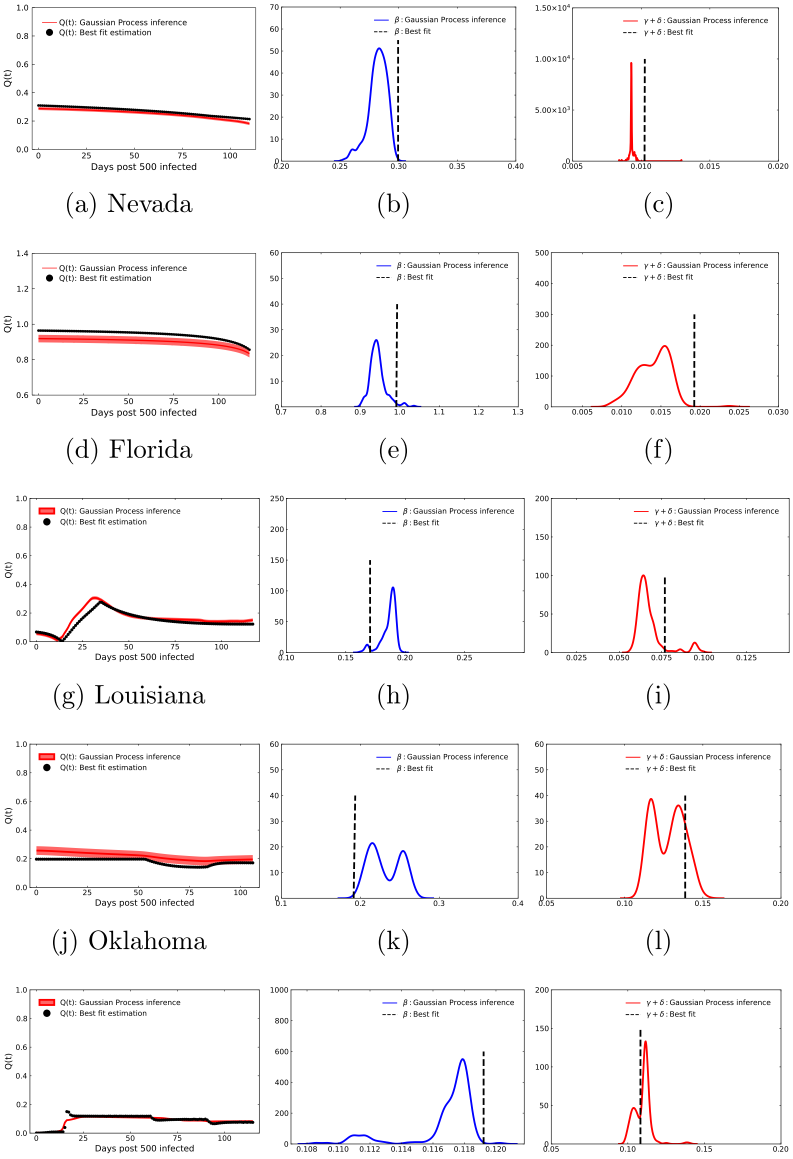
Arizona

Utah

South Carolina

Texas

Figure 3-9: [Parameter Inference for US states] Inferred parameters for 500 realizations of the Gaussian process residue model superimposed on the best fit model prediction applied to the region considered for demonstration, and shown for (a) the quarantine strength function $Q(t)$, (b) the contact rate β and the recovery rate $\gamma+\delta$. A total of 12 million iterations were performed on the MIT Supercloud cluster to generate parameter histograms for one state.



(a) Nevada (b) (c)

(d) Florida (e) (f)

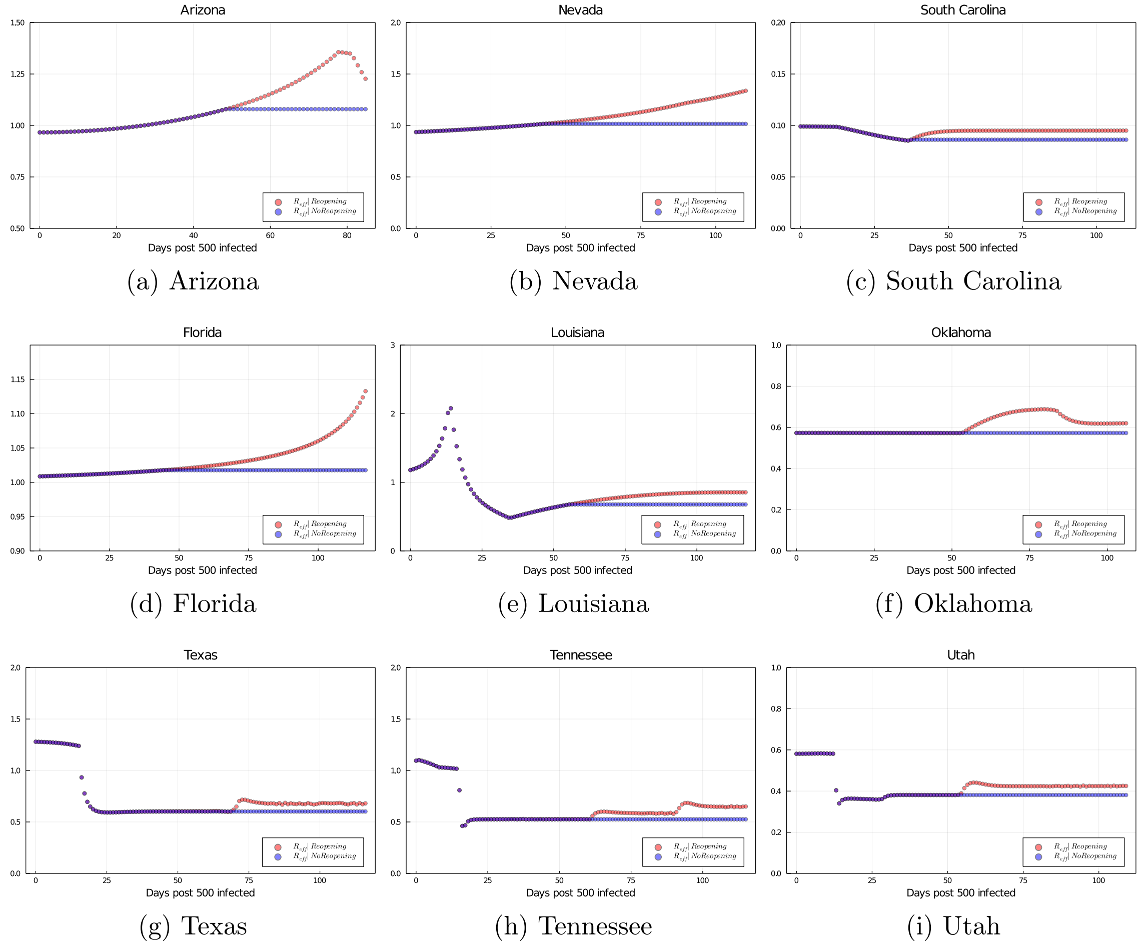
(g) Louisiana (h) (i)

(j) Oklahoma (k) (l)

(m) Tennessee (n) (o)

Figure 3-10: [Parameter Inference for US states] Inferred parameters for 500 realizations of the Gaussian process residue model superimposed on the best fit model prediction and shown for the quarantine strength function $Q(t)$ (left column), the contact rate β (middle column) and the recovery rate $\gamma+\delta$ (right column) for the US states considered in the present study. A total of 12 million iterations were performed on the MIT Supercloud cluster to generate parameter histograms for each region.

3.7 Effective reproduction number comparison across states



(a) Arizona (b) Nevada (c) South Carolina

(d) Florida (e) Louisiana (f) Oklahoma

(g) Texas (h) Tennessee (i) Utah

Figure 3-11: The comparison of the effective reproduction number, as defined in Equation 3.34 with and without reopening, shown for all US states considered in the present study

We estimate an equivalent of the effective reproduction number $R_{eff}(t)$ [110] for all states, with and without reopening. In a prior study of the QSIR model [31], the authors have defined a Covid spread parameter, C_p equivalent to the effective reproduction number to be the following

$$C_p(t) = \frac{\beta}{Q(t) + \gamma + \delta} \tag{3.34}$$

Region	C_p range (Our study)	R_{eff} range (Ref. [57])
Arizona	1.1 – 1.4	1.15 – 1.3
Florida	1 – 1.1	1.07 – 1.64
Nevada	1.05 – 1.5	1.19 – 1.5
Louisiana	0.7 – 1	0.88 – 1.62
Texas	0.6 – 0.7	1.08 – 1, 3
Tennessee	0.5 – 0.65	0.97 – 1.07

Table 3.5: C_p and R_{eff} value ranges from reopening till one month post that, for 6 states considered in our study; lie close to each other.

We have used this definition of the Covid spread parameter $C_p(t)$, which is equivalent to the effective reproduction number $R_{eff}(t)$ in the context of the QSIR model. Figure 3-11 shows the comparison of the Covid spread parameter, as defined in Equation 3.34 with and without reopening, shown for all US states considered in the present study. For all the states, we can see that without reopening, a diminished effective reproduction number is seen, indicating moving in the right direction of halting the infection spread.

To further validate the Covid spread parameter variation and its relation to the effective reproduction number, we compare the variation in C_p to the R_{eff} obtained through a prominent Covid-19 forecasting model used by the CDC, USA [57, 142]. For all of the 9 states which we considered, the time at which an upsurge is seen in C_p due to early reopening corresponds very well to the exact time at which an upsurge is seen in R_{eff} [57, 142]. In addition, the C_p and R_{eff} value ranges from reopening till one month post that; for the states of Arizona, Nevada, Louisiana, Florida, Texas and Tennessee lie close to each other, as mentioned in Table 3.5. This further validates the results of our study and the quantitative metrics derived therein.

Chapter 4

The SafeBlues project (Virtual Virus Spread)

4.1 Summary

After the first 3 chapters, we now show how scientific machine learning can be used in a novel application: virtual virus spread through Bluetooth tokens. Following is the general premise of the SafeBlues project:

How do fine modifications to social distancing measures really affect COVID-19 spread? A major problem for health authorities is that we do not know.

In an imaginary world, we might develop a harmless biological virus that spreads just like COVID-19, but is traceable via a cheap and reliable diagnosis. By introducing such an imaginary virus into the population and observing how it spreads, we would have a way of learning about COVID-19 because the benign virus would respond to population behaviour and social distancing measures in a similar manner. Such a benign biological virus does not exist. Instead, we propose a safe and privacy-preserving digital alternative.

Our solution is to mimic the benign virus by passing virtual tokens between electronic devices when they move into close proximity. As Bluetooth transmission is the most likely method used for such inter-device communication, and as our suggested

“virtual viruses” do not harm individuals’ software or intrude on privacy, we call these *Safe Blues*.

In contrast to many app-based methods that inform individuals or governments about actual COVID-19 patients or hazards, Safe Blues does not provide information about individuals’ locations or contacts. Hence the privacy concerns associated with Safe Blues are much lower than other methods. However, from the point of view of data collection, Safe Blues has two major advantages:

- Data about the spread of Safe Blues is uploaded to a central server in real time, which can give authorities a more up-to-date picture in comparison to actual COVID-19 data, which is only available retrospectively.
- Sampling of Safe Blues data is not biased by being applied only to people who have shown symptoms or who have come into contact with known positive cases.

These features mean that there would be real statistical value in introducing Safe Blues. In the medium term and end game of COVID-19, information from Safe Blues could aid health authorities to make informed decisions with respect to social distancing and other measures.

In this study we outline the general principles of Safe Blues and we illustrate how Safe Blues data together with neural networks may be used to infer characteristics of the progress of the COVID-19 pandemic in real time. Further information is on the Safe Blues website: https://safeblues.org/

4.2 Introduction

As the Covid-19 pandemic was evolving, there was a dire need for timely information about the spread of the COVID-19 virus. The idea of using contact-tracing mobile device apps to help in this endeavour received a considerable amount of attention [9, 5, 24, 36].

There are essentially two classes of information that are provided by such apps:

- information about the actual people that an infected person has met, and
- data that can form a basis for statistical inference and control of the epidemic.

All the contact-tracing apps that we are aware of seem to have collection of the first type of information as their primary purpose. This is not surprising. It is clearly of paramount clinical importance to identify infected people, both for treatment purposes and to prevent them from spreading COVID-19. However, as the epidemic progresses and the conversation turns to the best way of relaxing government controls, it is also very important for decision makers to understand how the epidemic behaves in the whole population. At the moment data exists only about individuals who have come to the attention of authorities because they have been infected or there has been some reason to think that they are at a high risk of infection.

In this chapter, we propose a framework that is aimed exclusively at collecting the second type of information mentioned above. Compared to other frameworks, it has the advantages of

- providing population wide aggregated data in real time,
- tracking the way that the epidemic might be progressing in parts of the population that do not come to the attention of authorities, and
- being less intrusive from a privacy point of view.

In general, the spread of a virus depends on both its biological properties and the behavioural properties of the population. Biological properties of COVID-19 have been studied since the start of the outbreak [169]. On the other hand, population behaviour is changing rapidly due to unprecedented social distancing measures and is hard to observe and to predict. As a consequence, achieving tight real-time estimates of time-varying parameters such as $R_{\text{eff}}(t)$, the expected number of individuals infected by an infectious person is a difficult task [124].

In an imaginary world, we might develop a harmless biological virus that spreads just like COVID-19, but is traceable via a cheap and reliable diagnosis. By spreading such an imaginary virus throughout the population, the spread of COVID-19 could be easily estimated because the benign virus would respond to population behaviour and social distancing measures in a similar manner. Such a benign biological virus does not exist. Instead, we propose a safe and privacy-preserving digital alternative that we call *Safe Blues*.

The Safe Blues method would use Bluetooth signals similarly to the suite of existing and emerging contact tracing frameworks Proximity [36], Blue Trace [9] and the Privacy-Preserving Contact Tracing framework currently being developed by Apple and Google [5]. However, in contrast to these frameworks, Safe Blues does not record information about individuals and their interactions. Instead, it will help understand population wide dynamics in a privacy-preserving manner.

The Safe Blues idea is that mobile devices mimic virus spread via the safe exchange of Bluetooth signals. Then, aggregated counts are reported to a server without recording private information. By periodically creating various strands of Safe Blues and repeatedly spreading them through the (mobile device) population, our analysis of the signals will help to obtain aggregate estimates of population contact. The result will be a real-time estimate of the effect of any social distancing rules that are put in place. Further, when retrospective information about COVID-19 case numbers becomes available, it can be combined with Safe Blues data to train sophisticated machine-learning procedures to estimate COVID-19 infection numbers in real-time.

As an example, in Figure4-1 we present an illustration of the path of one fictitious epidemic where government interventions are taken to curb the spread of the disease. The figure also presents the paths of recorded Safe Blues activity. Decision makers only receive data on the real epidemic lagged by 15 days, but the Safe Blues information is received in real time. The epidemic begins with a proportion of 1.25% infectives, along with 50 different Safe Blues Strands that have been initialised with similar proportions of infectives.

After an initially rapid spread of the epidemic, social distancing regulations are quickly tightened, and after two weeks, they are fixed to prevent the vast majority of social contact. As these rules go into effect, the proportion of infectives begins a slow but steady decline. All the while, Safe Blues Strands are being simulated on mobile

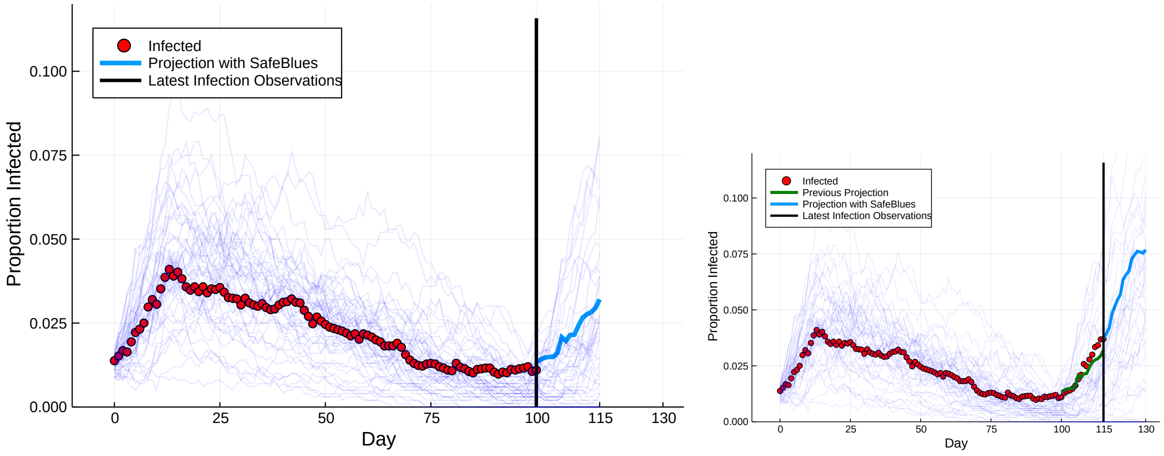


Figure 4-1: Safe Blues presents a near real-time estimate of the potential for virus spread: During days 100–115, Safe Blues activity is observed to rise and thus helps predict a rise in COVID-19 cases. (a) Projection during days 100 – 115 based on Safe Blues real time information. (b) Assessing the quality of the projection 15 days later.

devices. As a consequence, the proportion of Safe Blues infections mirrors the decline in real infectives, driven by a corresponding reduction in physical proximity between Safe Blues-enabled devices.

After 100 days since the start, and months of a promising decline in case numbers, the social distancing rules are mostly lifted. The first plot presents the view at 115 days: about two weeks past this change. At this point, only data up to day 100 is observed, while Safe Blues information is observed in real-time up until day 115. These two pieces of information have also been used to compute a live estimate of the epidemic, shown as a light blue curve in Figure 4-1a. This estimate shows that the epidemic is again on the rise, and that the reversal of social distancing rules may have been too early or too aggressive.

The second plot shows the situation 15 days later, after which the lagged dynamics of the epidemic have again been observed. Once more, the projection shows a strong uptick of infections. Compared to the projection computed earlier, one can see that the Safe Blues system mimics the dynamics and interaction patterns of individuals with high accuracy to yield valuable information on the real-time behaviour of the epidemic. This reflects the reality that we are observing with the COVID-19 epidemic: a long incubation period and mild symptoms at the start of infection mean that diagnoses are delayed by weeks and real-time information is almost non-existent. In this way, the Safe Blues framework may provide unique, invaluable visibility into the current state of the epidemic and a powerful tool for early detection of subsequent waves or outbreaks.

In addition to such real-time early warning predictions, Safe Blues has the potential for more. At the moment, when a government adjusts social distancing directives, it is not clear what the effect on population behaviour is and, even if adherence is immediate, time will elapse before the effect of the measures on the spread of COVID-19 becomes observable. In particular, this makes it difficult to set social distancing measures in a way that balances objectives while keeping $R_{\text{eff}}(t)$ sufficiently low such that the epidemic does not take off again. This is an important consideration towards the end game of COVID-19. The Safe Blues idea will produce a faster feedback loop because indicators of social contact can be observed in a more timely fashion.

If Safe Blues is implemented in the early or middle stages of the COVID-19 pandemic, information obtained during that period will be beneficial for later decision making, especially when dealing with efforts to eliminate second or third waves of infections. The initial analysis of this study indicates that Safe Blues data may be useful for future projections of the epidemic. For this, we rely on statistical machine learning methods, mixed with solid principles of epidemiological modelling. As COVID-19 progresses, the estimates will become more and more precise. A consequence is that within three to six months of deployment of Safe Blues, estimates of $R_{\text{eff}}(t)$ for COVID-19 could be tighter than they currently are. Similarly, the effect of various forms of social distancing government directives on $R_{\text{eff}}(t)$ will be better understood. As an end result, governments will have a better grasp of how to optimally “flatten the curve” while keeping the economy as active as possible.

Structure of this study: In Section 4.3.1 we describe the motivation and underlying principles of Safe Blues. We also discuss the level of privacy that Safe Blues offers (Section4.3.1) and the potential for integration with current contact tracing apps (Section4.3.1). In Section4.3.2 we illustrate the power of Safe Blues for estimation, future projections of the epidemic, and control. For this we present two methods that we call Deep Safe Blues and Dynamic Deep Safe Blues. We then conclude in Section4.4. Further information is also available on the Safe Blues website: https://safeblues.org/.

This study also contains an extensive set of appendices. In Appendix4.5.1 we overview the software and the protocols of communication needed for Safe Blues. In Appendix4.5.2 we spell out some of the details of the machine learning methods employed for estimation and projection. In Appendix4.5.3 we describe the three test bed models that were used to evaluate the potential power of Safe Blues.

4.3 Results

4.3.1 Underlying principles

In the modelling of epidemics, the effective reproduction number $R_{\text{eff}}(t)$ is the central quantity that determines how the epidemic grows or diminishes. This quantity is defined as the average number of individuals infected by each sick individual at time t. For COVID-19, early estimates indicate that, without significant control measures being in place, $R_{\text{eff}}(t)$ lies in the range of 2 – 4 [95]. However, $R_{\text{eff}}(t)$ depends on a combination of biological and behavioural factors. Some key biological factors include the propensity of the pathogen to infiltrate human hosts, the duration of the disease, and the susceptibility of different age groups. Some key behavioural factors include personal hygiene practices, hand shaking practices and, importantly, the proportion of time that individuals are in physical contact or close proximity. The biological factors tend to be uncontrollable and, with the exception of weather effects, may be assumed to remain constant as long as significant virus mutation does not occur. However, the behavioural factors are controllable, at least to some extent. Indeed the suites of social distancing measures imposed in over 150 countries during the first few months of 2020 are attempts to control the behavioural component of $R_{\text{eff}}(t)$ [73].

Such social distancing measures, some of which are outlined in the impactful report [38], have been introduced to slow down the spread of COVID-19. Nevertheless, at this early stage, it is very difficult to quantify the effect that any particular social distancing measure is having on $R_{\text{eff}}(t)$ and the dynamics of the pandemic.

Such lack of quantifiability is problematic because all models attempting to aid policy makers by projecting the course of the epidemic require an estimate of $R_{\text{eff}}(t)$. This is often obtained by modelling that attempts to quantify the level of human to human interaction either at broad scales [124, 104, 109] or at finer scales [38, 19, 155]. A notable recent attempt to include such a quantification [77] used survey sampling of the UK population to estimate that $R_{\text{eff}}(t)$ shifted from around 2.6 prior to lockdown, to around 0.62 after lockdown, which occurred in March 2020. While impressive, such questionnaire-based surveys are difficult to execute and are not able to yield real-time estimates of $R_{\text{eff}}(t)$. Other attempts at measuring $R_{\text{eff}}(t)$ [90] use up-to-date counts data such as the now famous dashboard by the CSSE at Johns Hopkins [33]. However, in such cases the problem is that reported "live data" about COVID-19 is based only on confirmed tested cases and does not consider the large number of asymptomatic cases or untested cases that must exist. Better estimates become known only retrospectively, after the pandemic has progressed.

At this time (April-May 2020) a considerable number of countries still have a significant amount of infection, and many have imposed a suite of social distancing measures that together seem to be having the effect of reducing $R_{\text{eff}}(t)$. However, it is not clear which individual measures are driving this. In the second half of 2020 and onwards, getting more precise estimates of effectiveness will be very important. Many governments will grapple with the optimal way of lifting (and at times reinstating) social distance measures as they attempt to balance economic revival with health considerations. For this, having fine-tuned live estimates of $R_{\text{eff}}(t)$ and related quantities is of paramount importance. Safe Blues is designed to serve as a tool that can help in this arena. If it is introduced at around May-July 2020, then after a training and calibration period, by September 2020 and onwards the tool has the potential to serve as an aid to policy makers when they are considering the adjustment of social

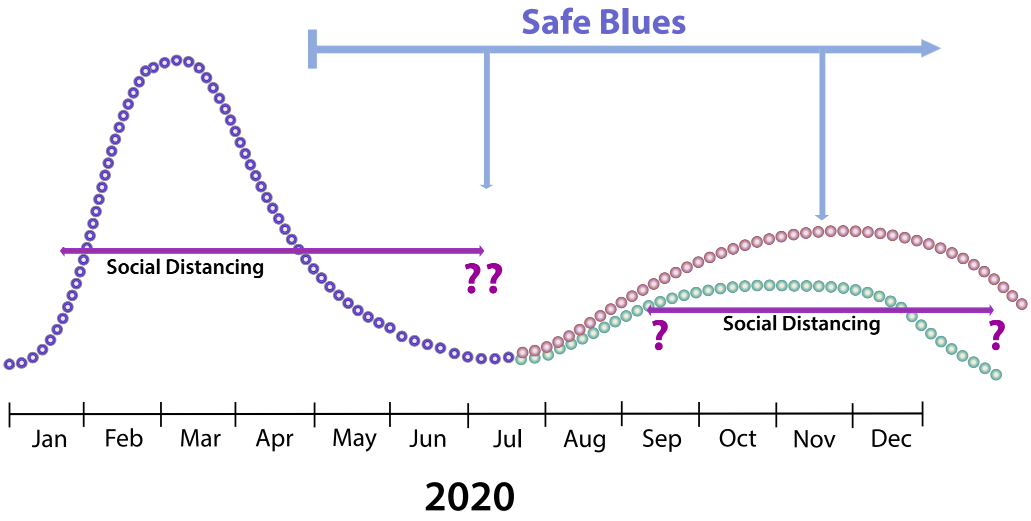


Figure 4-2: An illustrative timeline of the epidemic in a particular region. Introduction of Safe Blues at around May-July may imply that by September meaningful insights about the epidemic can be obtained. This can help to inform social distancing policy in the second half of 2020.

distancing directives. Figure4-2 presents a simple illustration of such a timeline.

How it Works

The key idea of Safe Blues is to obtain real-time estimates of gross population engagement dynamics in a safe and privacy-preserving manner. In real time, Safe Blues data can be processed to yield estimates of COVID-19 $R_{\text{eff}}(t)$ as well as other parameters that can be used to inform future epidemiological models.

The system works by having personal mobile devices take part in an ongoing *safe real-time virus spread simulation* where, by means of Bluetooth signals, the time that individuals spend in close proximity is a key driving factor. This is done in a way that does not compromise individual privacy, does not cause any risk to human health, and does not introduce any risk to individual software or hardware. See Figure4-3 for a schematic illustration of the Safe Blues system.

A *Safe Blue Strand* is a virtual token that circulates and replicates between the mobile devices of individuals using dynamics designed to reflect the transmission of an actual biological virus but without any threat to safety, software, or privacy. Similar to a biological virus, the harmless Strand is counted as “active” for a finite duration of time in each mobile device that is “infected”. During that time, if the mobile device is in close proximity to another device, there is a chance for the Strand to “spread” to the neighbouring device. Similarly, if the mobile device is in relative isolation, the Strand is not likely to spread and the mobile device will eventually “recover”. Further, as with actual biological viruses, Strands can have an incubation period during which they are not infective.

By allowing multiple types of Strands, or multiple Strands, to “infect” the mobile device population, the “epidemics” of the Strands respond to social mobility and social distancing measures in a similar (but not identical) way to COVID-19 response. However, in contrast to COVID-19, the number of devices “infected” by Strands can be measured in real time.

The Safe Blues system will periodically “inject” such Strands into the mobile host population and obtain real-time counts of the number of “infected” hosts for each Strand. While the population dynamics of each Strand will not directly resemble the dynamics of COVID-19, the underlying driving mechanism of close physical social interaction will be shared by both the real biological virus and the harmless Strands. Hence, we expect the course of Strand “epidemics” to be coupled with the course of the COVID-19 epidemic, especially under varying social distance measures.

The mechanism of communication between devices is short-range Bluetooth. This is similar to the communication protocol used by many emerging contact tracing apps (see Section4.3.1 below). In such a setting, as individuals spend time in close proximity, the propagation of these virtual Strands has a higher tendency to succeed and spread. Conversely, as individuals maintain a higher level of social distancing, the Safe Blues Strands are less likely to spread. Unlike systems that use contact tracing apps, the Safe Blues system is oblivious to the actual health status of specific individuals. For example, in Figure4-3, some individuals are infected by COVID-19 (red) while others are not (green). However Safe Blues is not aware of and does not

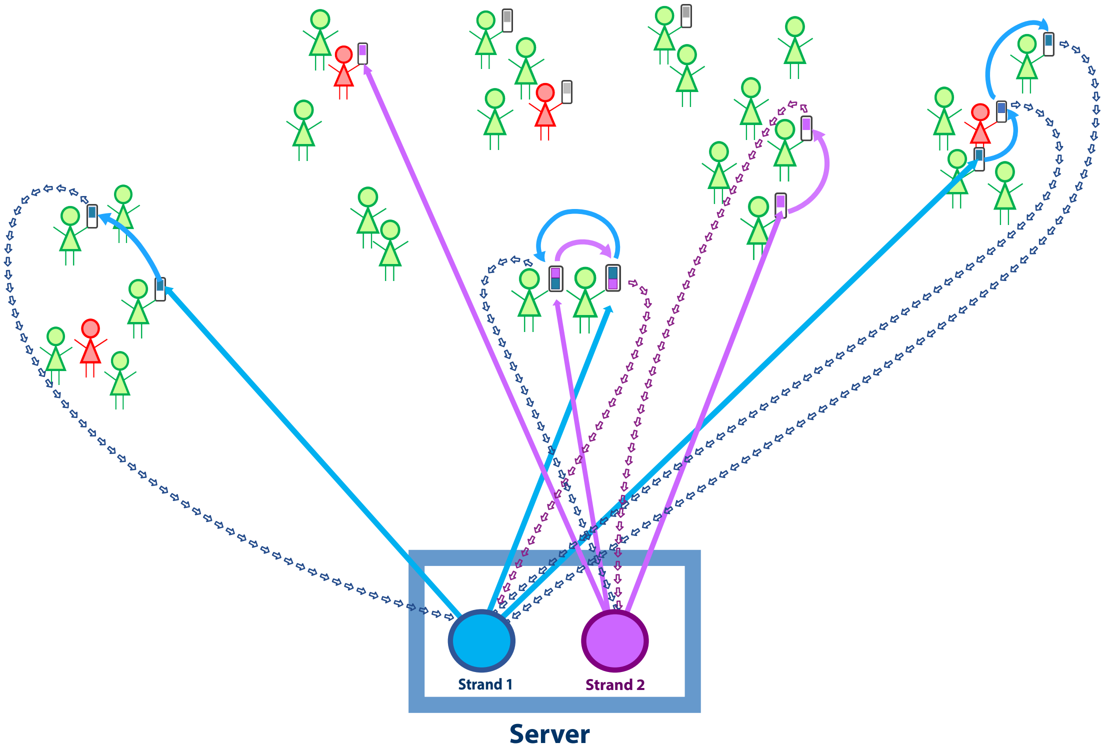


Figure 4-3: Individuals of the population with Safe Blues enabled devices take part in spreading Safe Blues Strands. Marked individuals support the Safe Blues system by carrying devices with Safe Blues software. COVID-19 infected individuals are in red and others are in green. The Safe Blues system operates independently of the COVID-19 status of individuals.

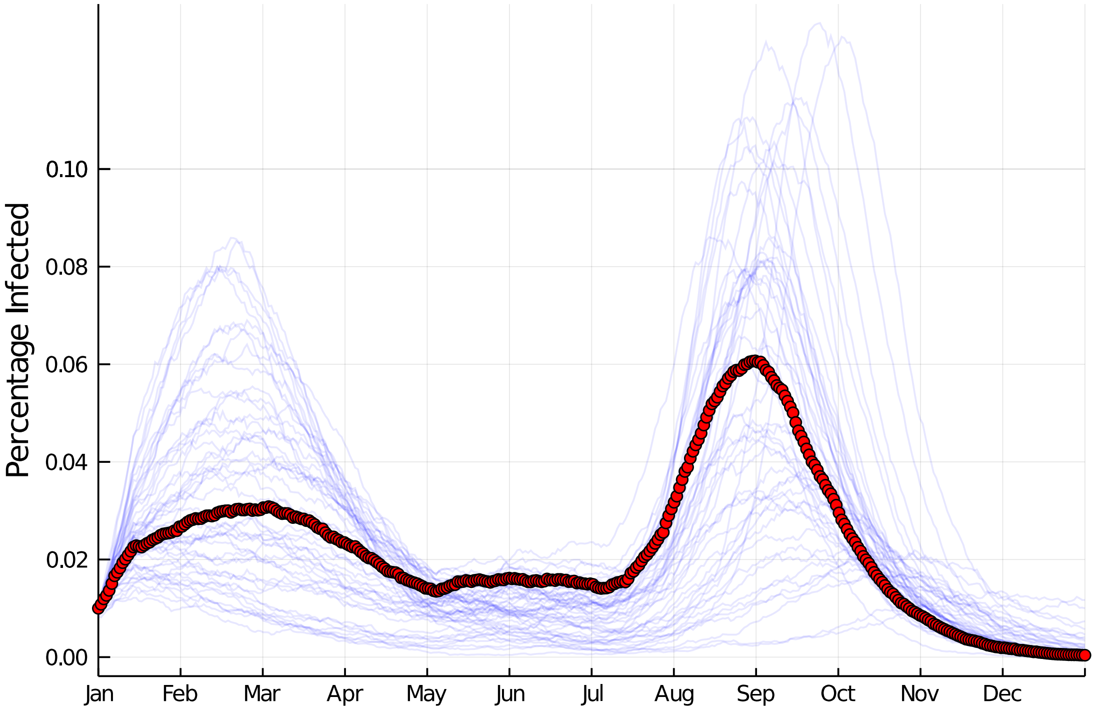


Figure 4-4: A potential course of the epidemic (number infected) with associated Safe Blues Strands. Social distancing measures modify the course of the epidemic and in the process influence Safe Blues.

need this private information. Similarly, some individuals participate in Safe Blues (as signified by a grey "mobile device") and others do not. Clearly some level of population participation is required, but Safe Blues does not require all individuals to participate.

Figure4-4 illustrates the outcome from a simulation of the COVID-19 epidemic in parallel to multiple Strand trajectories. Specific details about our simulation test beds are given in Appendix4.5.3. The key point illustrated in Figure4-4 is that multiple Strands can co-exist in parallel to the COVID-19 epidemic and that they are influenced by social interaction and social distancing in a similar way to COVID-19.

Further Details

Each Strand is uniquely identified by an integer s. The Strand has associated start time t_{start} at which the server *seeds* the Strand with the participating mobile devices.

As this occurs, each device independently chooses whether to get "infected" by the strand, with probability p_0, or ignores the seeding otherwise. Thus devices generate random outcomes that affect the (Safe Blues Strand) epidemic. Hence at t_{start}, if there are N^B mobile devices with Safe Blues enabled, there will be approximately $p_0 \times N^B$ devices infected with the specific Strand.

As an individual together with their Safe Blues-enabled device physically nears another individual with such a device, Bluetooth is used to transmit active Strands from the infected individual to the other. This communication involves a short-range distance measurement denoted by D. The duration of the physical proximity, denoted by τ, is also estimated. Then the probability of infection is given by a function $I_s(D, \tau)$, specific to Strand s. Although no Strand will behave exactly like COVID-19, statistical analysis will later reveal which choices of the function $I_s(D, \tau)$ yield better fits to COVID-19 data.

Once a device is infected by the Strand, it enters a random *incubation period.* During that period the device "carries" the Strand but cannot infect other devices. Once the incubation period is complete, the device becomes "infective" and starts to infect other devices that are in close proximity. This occurs until the device has "recovered".

Appendix4.5.1 provides further details on the suggested system architecture from a software engineering point of view.

On the Penetration Proportion and Strand Parameters: We use N to denote the number of individuals in a population. In determining the parameters of a Strand, the *penetration proportion* $\eta = N^B/N$ plays a key role. Assume that at a given point in time, COVID-19 has an estimated infection rate of β and removal rate of γ. We can then set the infection rate and recovery rate of strand s to roughly follow

$$\beta_s \approx \frac{\beta}{\eta} \qquad \text{and} \qquad \gamma_s \approx \gamma, \tag{4.1}$$

where the approximation is due to the lack of exact knowledge about γ and β, as well as due to the desire for some heterogeneity between strands. That is, we wish for the strands to have different attributes, yet they should roughly reflect the behaviour of COVID-19. The choices of the parameters β_s and γ_s can then be incorporated into actual operational strand parameters such as $I_s(\cdot,\cdot)$ and the infection period distribution, with details appearing in Table4.2. For the incubation period, we may want to set the strand parameters to be lower than typical COVID-19 parameters, for quicker response time in measurements.

The motivation for Equation(4.1) comes from basic epidemiological considerations appearing in SIR models. As an example, consider the difference equations associated with ModelI (see Appendix4.5.3). In this case, if one decreases the population size by a factor of η^{-1}, then achieving similar epidemic behaviour (on the smaller population) can be achieved by setting β_s and γ_s as in Equation(4.1).

As for the needed penetration level of Safe Blues, the larger a population is the smaller the penetration proportion η can be while still generating useful data. Numerical experiments suggest that in a relatively small population of $N = 100,000$ individuals an η between 0.1 and 0.2 already gives high-quality data. We hypothesise that with N in the order of millions a penetration level η between 0.05 and 0.1 may be sufficient for a successful estimation using Safe Blues. It is important to note that the required penetration level for Safe Blues is much lower than for contact-tracing apps to be successful.

Privacy with Safe Blues

Implicit in the design of Safe Blues is an important privacy feature: no individual interaction information or any other private information is shared between devices or between a device and the database. The entire protocol runs without associating long-term identifiers with users and no user can ever know the identity of any other user. In fact, users need not share their location, name, number, identity, health or infection status, movement patterns, or any other type of personal or identifying information with the app. Similarly, the devices do not share anything else between each other than the Strands with which they are currently infected. A specification of the information transmitted between Safe Blues devices and the server is in Appendix4.5.1.

This is in contrast to contact-tracing apps that raise more serious concerns about personal privacy, even when engineered using novel privacy-protecting methods. Fundamentally, this is because the goal of any contact-tracing app is to observe relationships between individual people through their interactions, whereas the goal of Safe Blues is to collect only aggregate simulated epidemic signals. Hence, Safe Blues can be implemented in a way that preserves privacy to a greater extent than any currently-proposed or implemented contact-tracing solution [9, 36, 11, 136]. Thus a proper implementation of Safe Blues can provide stronger privacy guarantees than apps attempting to mitigate COVID-19 through contact-tracing methods.

Nonetheless, there are some issues that need to be addressed. One is the case of an adversary choosing a rare Strand and infecting a user with that Strand, then tracking the spread of that particular Strand to attempt to track the user. In extreme cases where only very few users are affected by a Strand, a third party may be able to achieve some sort of de-identification. This, however, can be mostly subverted by making sure that the seeding probability of each Strand is sufficiently large to make Strands common. Note also that to transmit a rare Strand, the adversary would need to be within several meters of the target for an extended period of time. The fact that the seeding has to occur on each device makes sure that an impostor of the Database cannot perform a similar attack.

In large deployments, governments and health authorities implementing Safe Blues may wish to gain fine-tuned information regarding the geographic patterns and potential spread of the virus. For this, they could add a local identifier in the app. However, such an identifier should refer only to a general region such as a country, state, or a major city. This is also approximately the same level of granularity at which users can be trivially tracked through their internet address, and a myriad of other well-established techniques.

Finally, there may be concern that each device must connect back to a central

Database each day. This is a valid concern, but must be understood in the context of modern apps, where most users already accept regular connection back to servers. This is a widespread practice in almost every app, for various reasons including automatic updates and bug reporting. With any such connection there is some privacy leakage through internet addresses and other identifiers of traffic patterns. However, with Safe Blues, the Hosts do not share any long-term identifiers with the server, and there is no way for a server to tag a user across multiple database pushes. This makes it impossible to perform tracking or de-identification beyond that provided by, for example, knowledge of any website that a user visits.

Potential for Integration with Existing and Emerging Apps

To the best of our knowledge, the Safe Blues system significantly differs from all existing and emerging apps dealing with COVID-19. Contact-tracing apps are concerned with individuals and interactions, whereas the Safe Blues system estimates general population dynamics. Nevertheless, in terms of the software and hardware infrastructure, there is room to embed the Safe Blue protocol within contact-tracing apps in a simple manner. Alternatively, one may consider implementing a stand-alone Safe Blues app, in which case "marketing" the app through a trusted organisation is essential to gain a user base, similar to any other app of this kind.

At the time of writing, multiple countries and organisations have already implemented contact-tracing apps using Bluetooth, with several additional countries rapidly developing their own solutions [24, 36]. This includes the recently published white paper associated with the Singaporean app TraceTogether [9]. Several of the organisations developing these apps have also made the source code available.

We examined the source code of some of the apps and observed that implementing the Safe Blues protocol within them would generally be a straightforward extension to the existing work. In Table4.1 we list current contact-tracing apps that use Bluetooth where, if not stated otherwise, an app is available for both Android and iOS. At the moment, many of these apps require running in the foreground. However, Apple and Google are rapidly developing APIs (Application Program Interfaces) for con-

Country	App Name and reference	Source Code	Comments
Australia	CovidSafe [7]	Not released	A clone of TraceTogether [53]
Czech Republic	eRouska [116]	[117]	
Germany	ITO [74]	N/A	Only on Android at this time
India	Aarogyasetu App [106]	N/A	Also uses GPS
Israel	Hamagen [119]	[118]	Also uses GPS
North Macedonia	StopKorona! [120]	N/A	
Poland	ProteGO! [115]	[114]	Not available at this time
Singapore	TraceTogether [53]	OpenTrace [52]	BlueTrace specification [9]

Table 4.1: Current contact-tracing apps that use Bluetooth.

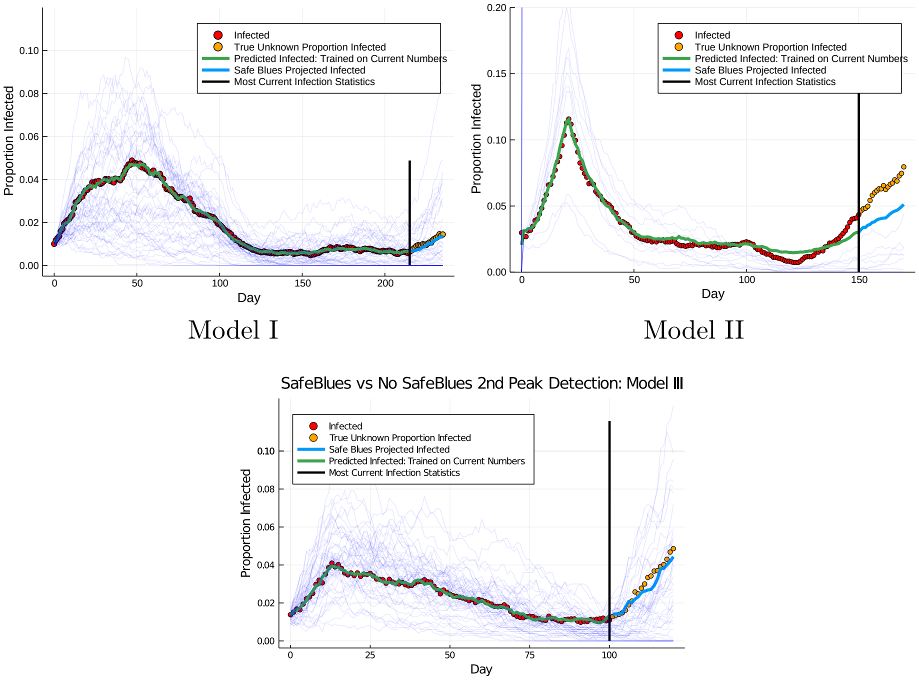
tact tracing [5]. These APIs will allow government-supported apps to use Bluetooth communication in the background.

4.3.2 Using Safe Blues for Accurate COVID-19 Projections

While contact tracing apps can allow public health researchers to identify specific individuals who might possibly be infected, the focus on the population of known infected individuals leads to a heavily-biased sample. This does not aid in estimating the overall spread of the infection. We now illustrate how our alternative approach via Safe Blues is able to give sufficient information for estimating the real-time spread of the disease and allows for estimating the potential effects of social distancing measures.

Our methodology relies on tools from machine learning, including Deep Neural Networks [49]. We also rely on methods for fitting UODEs (Universal Ordinary Differential Equations) [129]. While these are advanced mathematical and statistical tools, using our methodology is simple from a user perspective.

The basic setup follows the paradigm presented in Figure4-1 where information of COVID-19 is available up to a certain point, after which only Safe Blues information is available, typically with a delay in the order of two weeks. This represents the fact that COVID-19 information is not present in real time, in contrast to Safe Blues. The relative magnitude of historic social distancing measures also is available as input for projections (for example “full lock down”, “partial lock down”, etc.).



Model I

Model II

Model III

Figure 4-5: Deep Safe Blues: Safe Blues detection of a second wave applied to data generated from Models I, II, and III. The proportion of infected individuals is only known until the vertical black lines. After that point, only Safe Blues information is available. Nevertheless, Deep Safe Blues (trained up to the black line) is able to accurately predict a second wave of COVID-19 attack.

Our analysis aims to illustrate the predictive power that can be gained by utilising Safe Blues information. For this we created three models that aim to mimic the true spread of an epidemic while taking into account the interaction between individuals via mobility or other means. A detailed description of the models is in Appendix4.5.3. The purpose of these models is not to create a detailed representation of COVID-19 spread, but is rather to supply test beds for our projection methodology. Importantly, the projection methods that we present here do not rely on the specific form of these models. In fact, these specifics are not needed for understanding how Safe Blues can be used for projection and estimation. All that is important is that each model simulates a COVID-19 epidemic, subject to time-varying levels of social distancing, and captures the Safe Blues dynamics, also affected by the same social distancing measures. We label the models as Model I, Model II, and Model III, and use a single simulation run for each model. The details of the parameters used for the simulation runs are also in Appendix4.5.3.

Our goal is to develop methods for using Safe Blues information for a variety of tasks. These include:

1. Early warning of a rise towards a "second wave".
2. Understanding the effect of various social distancing regimes on $R_{\text{eff}}(t)$.
3. Designing optimal control policies for fine tuning social distancing measures towards the end game of COVID-19.
4. Projecting the course of the epidemic in the medium and long run.
5. Estimating the proportion of asymptomatic carriers of COVID-19.
6. Computing uncertainty bounds for projections.
7. Optimally choosing parameters and timing for newly created Safe Blues Strands.

We now demonstrate the potential power of Safe Blues for (1) and (2) above. (1) is achieved using a neural network model that we call *Deep Safe Blues*. (2) is achieved using a universally fitted ODE model that we call *Dynamic Deep Safe Blues*. We overview the results of these predictive methods and leave analysis of methodology dealing with (3)–(7) for future work.

Deep Safe Blues: Early Detection of a Second Wave

We created a deep neural network model that can be trained in real time based on the ensemble of Safe Blues strands and historical COVID-19 information. We call this *Deep Safe Blues*. It is able to accurately detect the start of a trend towards a second peak in the number of infected individuals a significant time before such data is available. Figure4-5 demonstrates the application of Deep Safe Blues for simulation traces from Models I, II, and III. Importantly, the same neural network architecture was used for all three simulation Models.

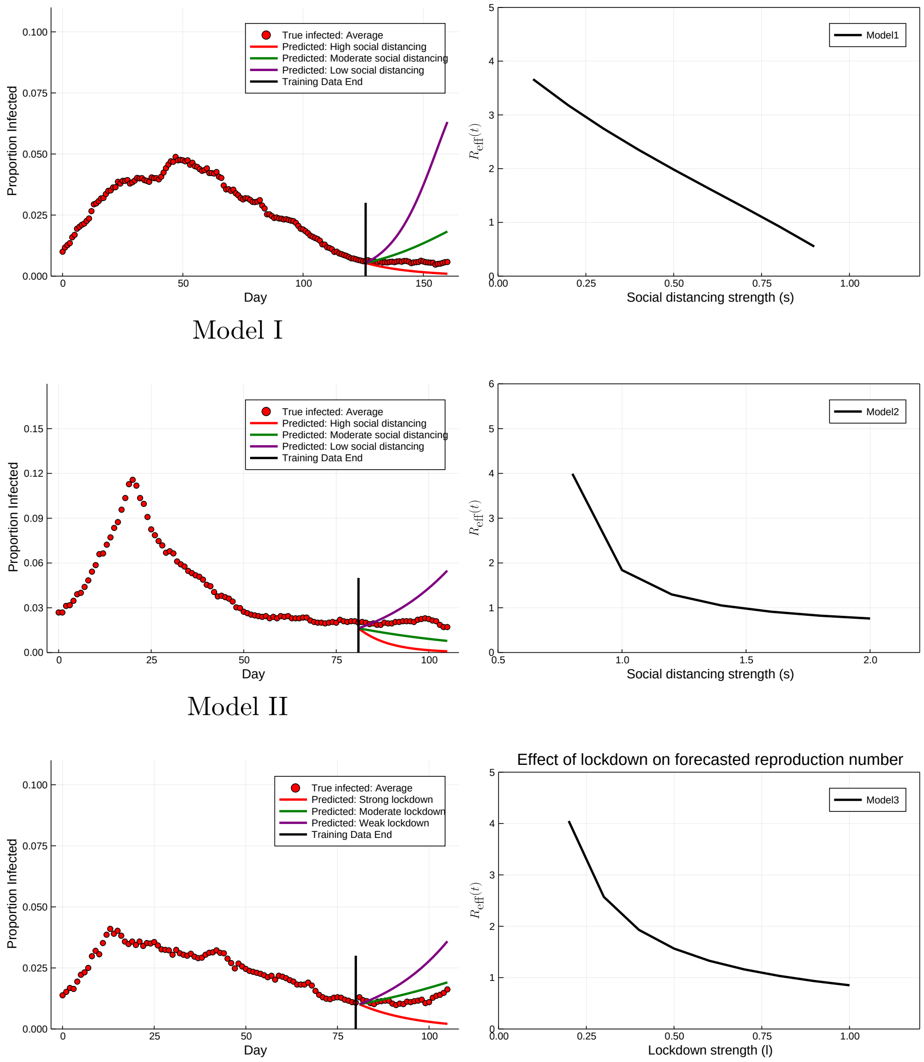
Our results demonstrate the strength of Safe Blues for early detection of a second wave and showcase that the auxiliary information provided by Safe Blues Strands can be valuable for detecting the start of a second peak. This can enable public health officials to respond during the essential early period before infection estimates can be updated.

A full specification of the fitting methodology is provided in Appendix4.5.2.

Dynamic Deep Safe Blues: Policy Projection

In addition to being a tool for estimating the current number of infected individuals before such data is available, Safe Blues can also help estimate the potential effect of policy decisions. For this we developed *Dynamic Deep Safe Blues* which is a tool for projecting R_{eff} as a function of future levels of social distancing. Figure4-6 demonstrates data-driven projections of R_{eff} under various policy levels. Such results can be used as input to decision makers for helping to determine the levels necessary to contain the outbreak and ensure that exponential growth into a second peak does not occur.

The implementation of Dynamic Deep Safe Blues is based on the training of a UODE [129] using the extra information provided by the Safe Blues Strands. This



Model I

Model II

Model III

Figure 4-6: Title: Demonstration of policy projection and refinement using Dynamic Deep Safe Blues on Models I, II, and III. For each of these models, we predict the effect of the social distancing strength on $R_{\mathrm{eff}}(t_0)$ where t_0 is the end of the training period (figures on the right). We also demonstrate potential near future trajectories as a function of policy decision (figures on the left).

approach mixes neural networks into epidemiological models in order to directly learn how policy decisions effect the spread of Safe Blues and the actual infection. Together, this allows for utilising the hidden information obtained from the simulated strands to quantify the effectiveness of social distancing approaches and determine the policies required to prevent further disease outbreak.

4.4 Discussion

This study has presented a framework and method that can aid in estimation and control in epidemics, specifically COVID-19. To the best of our knowledge this type of framework is fundamentally different from existing solutions and other suggestions that have appeared in the literature [122].

The machine learning principles and analysis that we used in this study appear to be robust enough to yield immediate value from collected Safe Blues signals. We have named these *Deep Safe Blues* and *Dynamic Deep Safe Blues.* Nevertheless, there remain open questions requiring further investigation. These include designing optimal control policies for fine tuning social distancing measures, projecting the course of the epidemic in the medium and long run, estimating the proportion of asymptomatic carriers of COVID-19, computing uncertainty bounds for projections, and optimally choosing parameters and timing for newly created Safe Blues Strands.

We also mention a potential extra benefit of the Safe Blues idea. In a second or third generation of apps, one may consider presenting individual users with an up to date count of how many strands of Safe Blues their devices are infected with. One may envision that this will enable users to get a feel for the level of social distancing that they are practising and to stay socially responsible as advised by government.

We believe that Safe Blues can have significant impact in the end game of COVID-19 as it may support governments in making optimal decisions with respect to adjustments of social distancing measures. Safe Blues can be easily implemented as a layer within a current contact-tracing app, or alternatively be hosted by other

COVID-19 related apps. Appendix4.5.1 provides an accessible description of what such an implementation requires. Due to the critical nature of COVID-19, our team at https://safeblues.org/ is willing and able to advise and help with such implementations and with the analysis of collected Safe Blues data.

4.5 Experimental Procedures

4.5.1 Software and Protocol Overview

We now describe a simple and straightforward protocol for the Safe Blues system, involving “Hosts”, “Strands”, a “Database”, and a “Controller” as its basic entities. The Hosts are mobile devices carried by individuals that run Safe Blues enabled software, such as an iOS or Android mobile phone. These Hosts attempt to infect each other with Safe Blues Strands when in physical proximity, akin to how the individuals themselves might infect each other with a real virus when in physical proximity. The central Database exposes a restful API that accepts and aggregates infection reports from Hosts and provides an endpoint to download an updated list of all current Strands along with their parameters. Additionally, the Database tracks the spread of each Strand through aggregate state counts, and allows interested parties to download aggregate time-series data on the spread of Strands. Finally, the Controller is a person charged with introducing new Strands into the system in response to machine learning and forecasting needs.

Hosts do not themselves need to be uniquely identified, however an implementation may wish to include a short-lived identifier that is regenerated regularly and can be used in conjunction with information about the source of the infection reports to guard against bad actors filling the database with false reports. The notion of a locale, such as a country or state may be implemented for the purposes of more geographically fine-grained detail on Safe Blues spread. We omit such details here and present only a basic design assuming a single-locale deployment.

The main activities of the distributed system are implemented by hosts and denoted with upper-case letters, these include `PULL-STRANDS` and `PUSH-INFECTION-REPORT` for interacting with the Database; `SHARE-LIST`, `DISCOVER-NEIGHBORS-LIST` and `DISCOVER-NEIGHBORS-` for interacting with other Hosts via Bluetooth. Finally, the activities `UPDATE-AFTER-MEET-NEIGHBOUR` and `PERIODIC-UPDATE` update the Strand states within a Host. The activities are listed in Table4.5 and explained in detail later. Since a Host may be turned off, may encounter connection issues, or its timely operation may be affected in many other ways, we assume a "best effort" schedule for performing these actions. This means that if a Host misses an activity, it ought to perform it as soon as it resumes operation.

Strands

Each Strand, uniquely identified by a `strandID`, is defined by an immutable list of parameters presented in Table4.2. For each Host, a given Strand acts as a state machine with four infection states: *SUSCEPTIBLE*, *INCUBATING*, *INFECTED*, or *REMOVED*. When a new Strand is pulled from the Database as described later, the Host is either seeded with the Strand with probability `seedingProbability` in which case that Strand starts in state *INFECTED*, or the Host is not seeded, and the Strand starts in state *SUSCEPTIBLE*. After this initial seeding, a Host may "catch" the Strand from another Host with some probability at any time between `startTime` and `endTime` given that the other Host is in close proximity and its state for that Strand is *INFECTED*. This happens at each encounter with an infected host according to a probability determined by the `infectionProbabilityMap` that depends on the mean distance and total duration of that encounter. If the Host is to catch the Strand, then that Host enters the *INCUBATING* state for a random length of time, determined by `incubationPeriodDistribution`. Note that a Host that is seeded with a Strand does not undertake this incubation period. Regardless of how a Host transitioned into the *INFECTED* state, it stays in that state for a random time determined by the `infectiousPeriodDistribution`. Finally, after being in state *INFECTED* for this amount of time, the Host transitions into *REMOVED* state and stops interacting with that Strand. The mechanics of these transitions are defined in detail in later sections.

The Database

The Database exists to distribute the Strand parameters to Hosts and to track and aggregate the spread of each Strand. The Host pulls an updated list of Strands from the Database daily in an activity denoted `PULL-STRANDS` in order to discover newly

Parameter	Name	Value Type
s	`strandID`	Unique identifier.
t_{start}	`startTime`	Date and time at which the Strand begins.
t_{end}	`endTime`	Date and time at which the Strand ends.
p_0	`seedingProbability`	Seeding probability.
$I(D, \tau)$	`infectionProbabilityMap`	Mapping infection probability as distance (cm) and exposure (min).
F_I	`incubationPeriodDistribution`	The distribution of the incubation period in hours.
F_R	`infectiousPeriodDistribution`	The distribution of the infection period in hours.

Table 4.2: The parameters that define a single Strand.

Parameter	Meaning
`currentIncubatingStrands`	List of current Strands with state *INCUBATING* on the Host.
`currentInfectedStrands`	List of current Strands with state *INFECTED* on the Host.
`currentRemovedStrands`	List of current Strands with state *REMOVED* on the Host.

Table 4.3: The daily information pushed to the Database from each Host via the `PUSH-INFECTION-REPORT` activity.

introduced Strands. We describe this and other Host activities in the next subsection. The implementation should begin advertising an upcoming Strand at least one day before its `startTime`, to guarantee full propagation of its parameters prior to the start of its spread. Additionally, an implementation may wish to version Strand definitions or distribute changes to Strand definitions to reduce traffic.

The Host pushes a report of its current state for each Strand (in one batch) to the Database on a daily basis in an activity called `PUSH-INFECTION-REPORT`. In a `PUSH-INFECTION-REPORT` message, the Host sends to the Database three lists containing the set of Strands whose state is currently *INCUBATING*, *INFECTED*, and *REMOVED*, respectively. An implementation may wish to randomly distribute these pushes throughout the day to avoid burst traffic to its servers.

After collecting daily pushes from Hosts, the system aggregates the individual Host states to create `totalHostsSusceptible`, `totalHostsIncubating`, `totalHostsInfected`, and `totalHostsRemoved` for each strand. This update is done in a straightforward manner based on the collected history of state information as described in Table4.3. This server side aggregation uses a best effort discipline to deal with missing values

Parameter	Meaning
`strandID`	The unique identifier of the Strand.
`date`	The date.
`totalHosts`	An estimate of the total number of participating Hosts.
`totalHostsSusceptible`	The estimated number of Hosts in state *SUSCEPTIBLE* for `strandID` on `date`.
`totalHostsIncubating`	The number of Hosts in state *INCUBATING* for `strandID` on `date`.
`totalHostsInfected`	The number of Hosts in state *INFECTED* for `strandID` on `date`.
`totalHostsRemoved`	The number of Hosts in state *REMOVED* for `strandID` on `date`.

Table 4.4: Aggregate information publically available via the Database.

and other inconsistencies. A total estimate of the number of participating Hosts is presented via `totalHosts` (this number varies over time based on Safe Blues usage). This aggregate time series data is exposed for visualisation and data-analysis purposes in an open manner where for each `date` and active `strandID` for that day, the fields in Table4.4 are made available.

Hosts

The Safe Blues protocol runs on Hosts and communicates over Bluetooth, either in a stand alone Safe Blues app, or possibly within other COVID-19 related apps that already utilise Bluetooth communication (see Section4.3.1). The Bluetooth specification defines many roles through its own state machine to facilitate communication, of which the Advertiser and Scanner roles are of importance for this discussion. An Advertiser continually broadcasts short advertising packets on pre-defined advertising frequencies, whereas a Scanner reads those packets being broadcast along with a small set of metadata such as the Advertiser address and a Received Signal Strength Indicator (RSSI). Most full featured Bluetooth devices can perform both roles simultaneously. The Bluetooth specification [125] further allows devices to pair with each other and establish sessions. In this discussion, we assume a model of Bluetooth using Advertisers and Scanners without any session creation, though it is partially incomplete due to implementation constraints, and an actual implementation might need to use slightly different interfaces, or simulate this model via Bluetooth sessions.

Activity	Meaning
PULL-STRANDS	Update list of Strands from Database. See Table4.2.
PUSH-INFECTION-REPORT	Report Host's state for each Strand to the Database. See Table4.3.
SHARE-LIST	Advertise list of Strands in state *INFECTED* to Hosts in proximity. See Table4.7.
DISCOVER-NEIGHBORS-LIST	Receive and process a SHARE-LIST from another Host as a result of Scanning.
DISCOVER-NEIGHBORS-PROXIMITY	Estimate contactDuration and averageDistance of neighbor Host during Scanning.
UPDATE-AFTER-MEET-NEIGHBOUR	Update local Strand state after a Host leaves proximity. See Table4.6.
PERIODIC-UPDATE	Update local state information periodically (hourly). See Table4.6.

Table 4.5: Activities carried out by Hosts.

Discussion of practical issues and one way of implementing this has been reported earlier [9].

The list of activities carried out by a Host is described in Table4.5. The remainder of this subsection outlines how these activities are carried out, together with the associated data.

The local state information stored and updated by the Host is describe in Table4.6. A tempID is used to distinguish one Host from another in a small spatio-temporal window, allowing Hosts to correlate several Advertising packets to one Host, which in turn allows for computing the total duration of close contact. The tempID may be a random Bluetooth address and ought to be changed regularly, such as on each hour. The currentInfectedStrands is the list of Strands with state *INFECTED* for the Host. Infection may occur via "seeding" or via communication with neighbouring Hosts. For each such Strand there is an infection end time listed in strandInfectionEnd and once infection has ended, the Strand moves to *REMOVED* state and is removed from currentInfectedStrands and updated to currentRemovedStrands (see the PERIODIC-UPDATE activity description below). The currentIncubatingStrands is a list of Strands that the Host has acquired via interaction with neighbouring Hosts and are still in the "incubation period". These Strands have a corresponding end time

Parameter	Meaning
`tempID`	A temporary unique identifier for the Host.
`currentIncubatingStrands`	A list of the Strands which are in state *INCUBATING* for the Host.
`currentInfectedStrands`	A list of the Strands which are in state *INFECTED* for the Host.
`currentRemovedStrands`	A list of the Strands which are in state *REMOVED* for the Host.
`strandIncubationEnd`	A list of times for which each Strand in the *INCUBATING* state will change to *INFECTED* state.
`strandInfectionEnd`	A list of times for which each Strand in the *INFECTED* state will change to *REMOVED* state.

Table 4.6: Local state information of a Host.

in the `strandInfectionEnd` list. The Host need not maintain a list of *SUSCEPTIBLE* Strands, as those are precisely the ones not in `currentIncubatingStrands`, `currentInfectedStrands`, or `currentRemovedStrands`.

The `PULL-STRANDS` activity in the Host discovers new Strands in the Database. When a new Strand is received from the Database, a "seeding" event occurs, whereby for each Strand pulled for the first time, the Host randomly decides whether to become infected by the strand or not based on the probability `seedingProbability`. If the outcome is to become infected, then infection begins at `startTime`. Even if such a seeding infection doesn't take place, the parameter information about the Strand should be retained because the Host may be infected by another Host later on.

When two or more Hosts are in physical proximity, each Host uses a "best effort" method to execute `SHARE-LIST`, `DISCOVER-NEIGHBORS-LIST` and `DISCOVER-NEIGHBORS-PROXIMITY`. The exact manner in which these activities are executed is implementation specific and may depend on the host operating system, on the exact app in which the protocol is implemented, and on hardware considerations. One possibility is for an implementation to encode the state of each Strand in a bit field, where each Strand has a fixed position in the field specified as one of its parameters; this allows for a space-efficient transfer of Strand states and an efficient way of checking which Strands need to be updated.

The `SHARE-LIST` activity Advertises the information in Table4.7 from one Host to another Host in physical proximity. On the receiving side, the `DISCOVER-NEIGHBORS-LIST`

Parameter	Meaning
`tempID`	A temporary and unique identifier identifying the Host.
`currentInfectedStrands`	A list of the Strands with which host is currently infected with.

Table 4.7: Information shared by a Host during physical proximity.

activity is executed when this information received through Scanning. One possibility is for each Host to be always be Advertising their `SHARE-LIST` and simultaneously Scanning incoming messages while in the background. The `DISCOVER-NEIGHBORS-PROXIMITY` activity supplies the receiving Host with the best effort `contactDuration` and `averageDistance` values. To aid in the distance estimation, the implementation may wish to include a transmitter power or device model identifier in the Advertising message, which could be combined with the RSSI or precomputed calibration data to get a highly accurate estimate of distance. Best effort is also used to estimate when such a physical interaction is complete. At that point the `UPDATE-AFTER-MEET-NEIGHBOUR` activity is carried out.

The `UPDATE-AFTER-MEET-NEIGHBOUR` activity is responsible for potentially "infecting" the receiving Host with each of the Strands that the sending Host is infected with but which the receiving Host is susceptible to. For each such Strand, there is an independent random outcome where `infectionProbabilityMap` is used with the estimated `contactDuration` and `averageDistance` as inputs to determine the probability of infection. If the resulting outcome is that the receiving Host should become infected then the state of the Strand in the Host is set to *INCUBATING*. This is reflected by updating the `currentIncubatingStrands` to include the given strand. Further, the `strandIncubationEnd` is appended with an entry for that specific Strand with the value set to the current time plus a random variate randomly generated from the `incubationPeriodDistribution`. In any case the `endTime` of the Strand needs to be respected and once passed, all activity regarding the Strand expires, and the Host may remove all information about the Strand once it has pushed its final state to the Database.

The `PERIODIC-UPDATE` activity ought to run on an hourly basis on Hosts and is designed to update the Strand state for each Strand as follows:

- *INCUBATING* → *INFECTED*: If the current time is larger than the `strandIncubationEnd` time for a given Strand, the Strand state is changed from *INCUBATING* to state *INFECTED*.

- *INFECTED* → *REMOVED*: If the current time is larger than the `strandInfectionEnd` time for a given Strand, the Strand state is changed from *INFECTED* to state *REMOVED*.

As such, this activity ensures that Strands progress through the "course of the disease" on Hosts. If the Host is unable to perform a `PERIODIC-UPDATE` activity due to any reason, it should perform this activity before any other activity, as to guarantee that Hosts follow the prescribed state schedule.

4.5.2 Safe Blues Projection Methodology

This appendix describes the details used in Section4.3.2.

Deep Safe Blues

We used the same Neural Network (NN) architecture for all three models. Real-time projections of estimated infected populations were generated by training a NN on the $B_{t,s}$, the ensemble of Safe Blues infection strands at time t, to predict I_t, the number of infected individuals at time t. The neural network $I_t = \text{NN}(B_{t,s})$ was a feed-forward neural network with two hidden layers of size 64 and tanh as the activation functions. Note that the size of the layers for ModelII were reduced to prevent overfitting given the significantly reduced number of Safe Blues strands. For ModelI, the data was trained on the time span $t \in [0, 215]$. For ModelII, the data was trained on the time span $t \in [0, 150]$. For ModelIII, the data was trained on the time span $t \in [0, 100]$. Each time the ADAM optimiser from `Flux.jl` [72] in `Julia` [12] with adaptivity parameter 0.01 was used for $2,000$ iterations with a loss function being the sum squared error.

Dynamic Deep Safe Blues

The universal ODE [129] trained a variant of the SIR model

$$
\begin{aligned}
S' &= -C\beta(p)\delta SI, & \tilde{S}' &= -C\beta(p)\tilde{S}\tilde{I}, \\
I' &= C\beta(p)\delta SI - \gamma(p)\delta_\gamma I, & \tilde{I}' &= C\beta(p)\tilde{S}\tilde{I} - \gamma(p)\tilde{I}, \\
R' &= \gamma(p)\delta_\gamma I, & \tilde{R}' &= \gamma(p)\tilde{I}.
\end{aligned}
$$

Here $C = 0.00004$ is a scaling constant, while $\beta(p)$ and $\gamma(p)$ are policy-dependent functions represented by neural networks. The parameters δ and δ_γ are coupling constants used to establish a relationship between the average of the Safe Blues strands (the tilde variables) to the original infection. The neural networks had 2 hidden layers of size 16 with tanh activation functions and a final abs to ensure that the outputted values were positive without imposing bounds on the parameters. The neural networks and coupling constants were determined by minimising the Euclidian loss between the true infected and I, and between the mean Safe Blues infected and $\tilde{I}$. The parameters were optimised using `DiffEqFlux.jl` [128] with the `Tsit5` adaptive Runge-Kutta method from `DifferentialEquations.jl` [126]. The optimisation was done in two passes, first with the ADAM optimiser from `Flux.jl` [72] with adaptivity parameter 0.001.

The fitting validations for each of the models are shown in Figure4-7.

Additional Remarks About Estimation

In reality, the limited availability of COVID-19 information may not be as simplistic as having a fixed delay. Nevertheless, in general the farther back we move in time, the more knowledge we have about the state of COVID-19. In practice, Safe Blues were not available at the start of the COVID-19 pandemic. See for example Figure4-2 for a possible time line that incorporates Safe Blues in the fight against COVID-19. Nevertheless, for the purposes of the analysis in this section we assume availability of Safe Blues from onset. Adaptations to more realistic scenarios are possible.

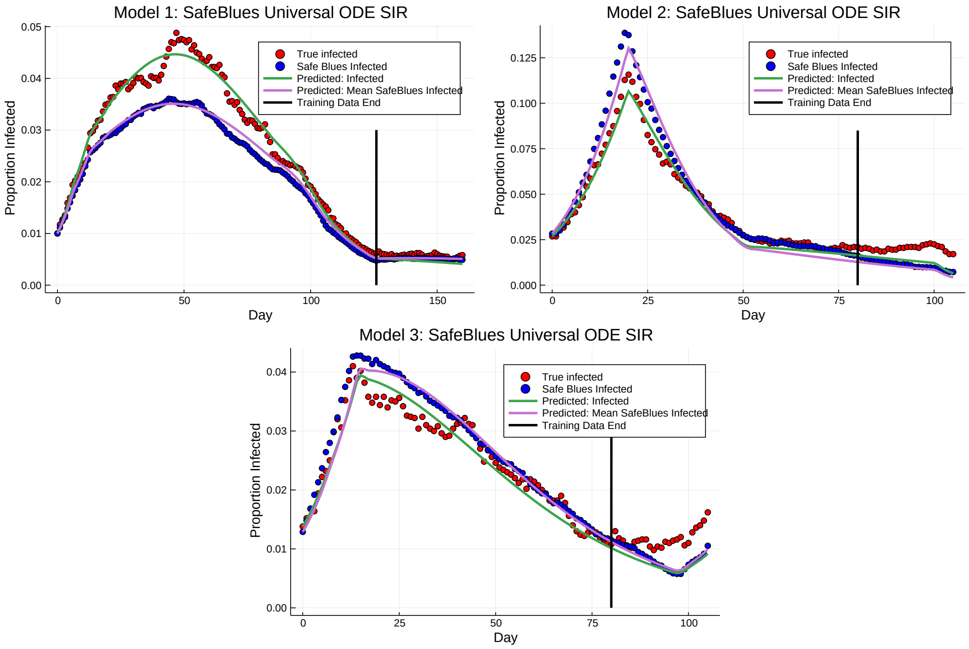


Figure 4-7: Fitting validations for the UODE models. Shown are the fits of the UODE models and their respective extrapolations.

The number of Safe Blues enabled users in our simulation runs was generally much smaller than is expected in reality. As a consequence, the variability of the Safe Blue infections was much higher than is expected in reality. This in turn reduces the predictive power of Deep Safe Blues and Dynamic Deep Safe Blues. However, this loss of predictive power is due to our limited simulation budget and can potentially be improved upon in an actual Safe Blues implementation.

4.5.3 The Test Bed Models

Here we outline the three different models that we use to test the effectiveness of deploying Safe Blues for projecting the spread of COVID-19. They are

ModelI: a discrete-time stochastic SIR model.

ModelII: a continuous-time stochastic SIR Model with migration.

ModelIII: a spatial movement model with location attraction.

Each of these models features a population comprising N individuals. Some of these individuals have Safe Blues-enabled devices, while others do not. At each point in time, the state of an individual registers whether they are susceptible, infected, or removed with respect to the actual virus (COVID-19). If an individual has a Safe Blues enabled-device, then the state also registers for every Safe Blues strand whether they are susceptible, infected, or removed.

The three models differ in their complexity and how they capture individual proximity. However, regardless of the model, individual proximity drives both the COVID-19 spread and the Safe Blues spread in a coupled manner, because both COVID-19 and Safe Blues only spread when individuals are in close proximity. This roughly approximates what one may expect to happen in a real scenario. Importantly, all three models allow for time-varying parameters that enforce social distancing, which in turn affects both COVID-19 spread and Safe Blues spread by changing how much time individuals spend in close proximity of each other.

ModelI is a very simple and stylised model that serves as a sanity check. One of its appealing features is that converges to the well-known SIR difference equations as the population size N becomes large, which makes this model well suited as a first test bed. ModelII incorporates several social and spatial features that are ignored in the first model. In particular, the second model has a spatial component (people have to be in the same place at the same time for virus transmission to occur) as well as a notion of social levels (people have a home where they meet a selected number of other people, a work place where they may meet a larger number of people, et cetera). ModelIII is a spatial model in which individuals move randomly in two-dimensional space. Its distinguishing feature is that it has a notion of centrality: although individuals move around randomly, they are biased towards visiting places that are important for them, such as their home and the supermarket. This creates a form of clustering that is not present in the first two models.

Model I: A Simple Stochastic SIR Model with Invitations

The deterministic discrete-time SIR epidemic model is characterised by the difference equations

$$\begin{aligned} \Delta S_{t+1} &= -\beta S_t I_t, \\ \Delta I_{t+1} &= \beta S_t I_t - \gamma I_t, \\ \Delta R_{t+1} &= \gamma I_t, \end{aligned} \tag{4.2}$$

where $\Delta S_{t+1} = S_{t+1} - S_t$, and similarly for ΔI_{t+1} and ΔR_{t+1}. The parameter β captures the rate of infection, while the parameter γ captures the rate of removal. Given an initial condition, the solution to these difference equations can be considered as the limit of the following simple stochastic epidemic model in discrete time.

Consider a homogeneous population of size N. At time t there are S_t susceptible, I_t infected, and R_t removed individuals. Each individual x invites a fixed number c of randomly chosen individuals to meet. If individual x invites and meets individual y, then x transmits the disease to y with infection probability $p = \beta/c$ if x has status infected and y has status susceptible. After all meetings have taken place, the individuals update their status for time $t+1$. A susceptible individual y becomes infected if the disease has been transmitted to y during one of the meetings. An infected individual x gets removed with removal probability γ. The fraction of susceptible, infected, and removed individuals converges to the solution of the difference equations (4.2) if the population size N becomes large.

Including Safe Blues strands is straightforward. We simply seed the mobile devices of a number of Safe Blues users with strands. Then the spread of the Safe Blues is similar to the spreading of the actual virus. The coupling between the Safe Blues and the actual virus arises because both Safe Blues and the actual virus can be only transmitted during meetings between individuals. A schematic illustration of this process is provided in Figure4-8.

To integrate social distancing in the original model, we consider the number of invitations c per individual as a random variable instead of a fixed number and make

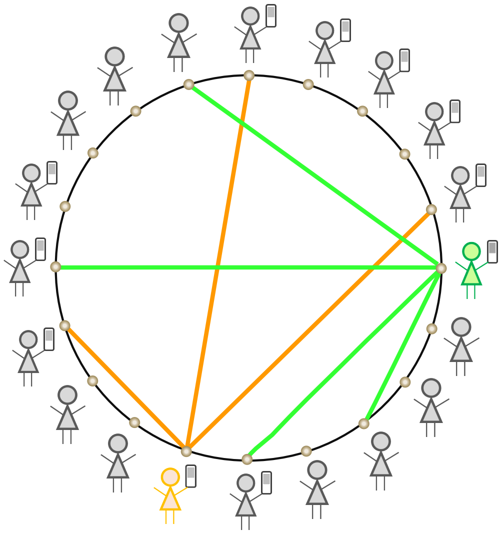


Figure 4-8: At every time point, each of the N individuals selects a random number of other individuals to invite and this implies physical proximity. In this case orange and green individuals make invitations.

the mean number of invitations time-dependent. Specifically, at time t individual x invites $c_{t,x}$ randomly chosen people to meet, where $c_{t,x}$ is an independent random variable having a (truncated) Poisson distribution with mean m_t. This means that the total number of meetings at day t is given by $\sum_{x=1}^{N} c_{t,x} \approx N m_t$. In this case the limit of the system is characterised by the difference equations (4.2) with β replaced by $\beta_t = p\, m_t$, where the infection parameter p is given as a model parameter.

Parameters used for the simulation run: We used the following model parameters to generate the data sets for the projection and policy evaluation experiments. Time consists of 366 days representing the year 2020. The population size $N = 10^4$ and a fraction 0.2 of the population has a Safe Blues enabled device. The infection probability of the biological virus is $p = 0.04$, while the corresponding removal probability $\gamma = 0.1$. We introduce 50 different Safe Blues strands, with strand s having infection probability p_s given as a point in the equidistant grid from $0.75(p/0.2)$ to $1.25(p/0.2)$. The removal probability γ_s for strand s is set equal to γ. The epidemics for the true virus and the 50 Safe Blues strands start at day 1 with a fraction 0.01 of infected individuals in the relevant (sub)population, while the others are susceptible. The number of invitations at day t for individual x follows a (truncated) Poisson distribution with mean parameter m_t. We use the following values for m_t to incorporate time-varying social distancing measures. The model was simulated using the `Numpy` library [121] in Python3.

Time Range (in days)	m_t
1–7	5
8–14	4
15–126	3
127–210	4
211–217	5
217–366	6

Table 4.8: The social distancing parameters for Model I.

Model II: A Stochastic SIR Model with Migration

For this model we consider a complete binary tree of depth k as in Figure4-9 where $k = 3$. Such a tree has 2^k leaves and $n = 2^{k+1} - 1$ nodes (including the leaves). There are $N = 2^k$ individuals and each of them is associated with a unique leaf. Every individual has a unique path between their leaf and the root, where the path consists of $k+1$ nodes (including the leaf and the root). At any point of time, every individual is located at one of the nodes on the path between its unique leaf and the root. A consequence is that individuals may be isolated with certainty in their leaf, or alternatively may be in the root or one of the other nodes of the tree where there is a possibility for them to be in physical proximity with other individuals. Thus the tree structure provides a spatial component (individuals are located at nodes) as well as a social component (individuals can meet specific groups of individuals only in specific parts of the tree).

We say that the root is at distance k and the leaves are at distance 0. The movement of individuals occurs in continuous time and is in unit steps, either increasing distance by 1 or decreasing it by 1. Then for any distance $i = 0, \ldots, k$, there is possibility to have up to 2^i individuals in the node. Hence, the farther away (towards the

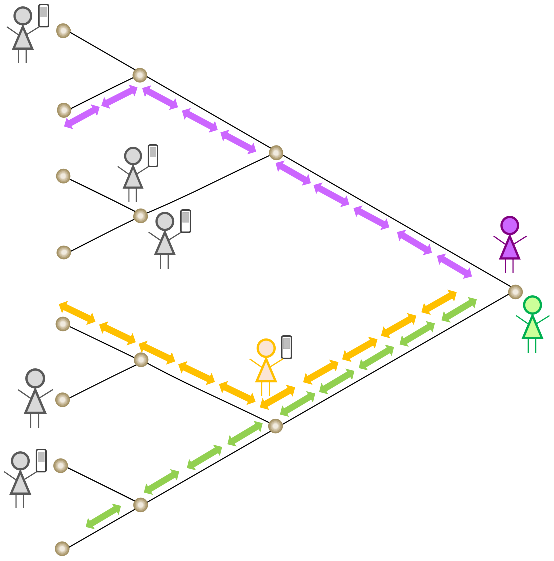


Figure 4-9: All $N = 2^k$ individuals traverse a binary tree between their private leaf and the root. At any node infection follows a continuous-time stochastic SIR model between the individuals present.

root) that an individual travels, the larger the probability of having other individuals in physical proximity. This setup naturally yields a social distancing mechanism: we may enforce the individuals spend (on average) more time close to their leaves.

Both individual mobility and the epidemic dynamics (including Safe Blues dynamics) are governed by continuous-time Markov Chains [30]. The mobility of each individual along the leaf-root path follows a birth-death process on $i = 0, \dots, k$, with all of the 2^k birth-death processes being independent. The birth rate is λ (constant for each distance level) and the death rates are $\mu_i = \mu i$ (linearly increasing with the proximity to the root). Social distancing is enforced by increasing μ, which causes individuals to spend more time near or at their leaf nodes.

Individuals' health state is subject to change via the standard SIR dynamics at each node in the tree. Specifically, if (at given node at a given time) there are ℓ individuals of which ℓ_S are susceptible, ℓ_I are infected and ℓ_R are removed (with $\ell = \ell_S + \ell_I + \ell_R$), then the rate of infecting other individuals at that node is $\beta_C \ell_S \ell_I$ (with the subscript C standing for COVID-19). Further, the rate of transitions from having ℓ_I to $\ell_I - 1$ is $\gamma_C \ell_I$. Upon infection (removal), a random susceptible (infected) individual present at the node is selected for infection (removal).

In a similar manner to the COVID-19 dynamics, the individuals with Safe Blues enabled devices are subject to SIR dynamics for Safe Blues strands. Each Safe Blues strand is indexed by a unique integer s. The infection rate for strand s is β_s and the corresponding removal rate is γ_s. The dynamics are similar to the COVID-19 dynamics described above, except that at a given node only individuals with Safe Blues enabled devices take part.

Parameters used for the simulation run: We used a standard Doob-Gillespie simulation algorithm [85, Chapter 10] to simulate the (time-varying) continuous-time Markov Chain of this model on the time range $[0, 366]$. This was for the case of $N = 2,048$ individuals ($k = 11$). The penetration proportion was $\eta = 0.5$ and thus $N^B = 1,024$.

The infection rate of COVID-19 was $\beta_C = 0.015$ and the removal rate was $\gamma_C = 0.1$. We simulated 10 Safe Blues Strands each with $\gamma_s = 0.1$ and $\beta_s = U_s \beta_C / \eta$ where U_s were pre generated i.i.d. uniform random variables on the range $[0.5, 1.5]$. The initial infection proportion of both COVID-19 and Safe Blues strands was at 0.03.

For mobility within the tree we used $\lambda = 0.9$ and μ_t was a time-varying rate as specified in Table4.9.

Time Range	μ_t
[0,20)	0.9
[20,50)	1.5
[50,100)	1.1
[100,120)	2.1
[120,200)	0.9
[200,220)	2.2
[220,300)	0.8
[300,366]	1.4

Table 4.9: The social distancing parameters for Model II.

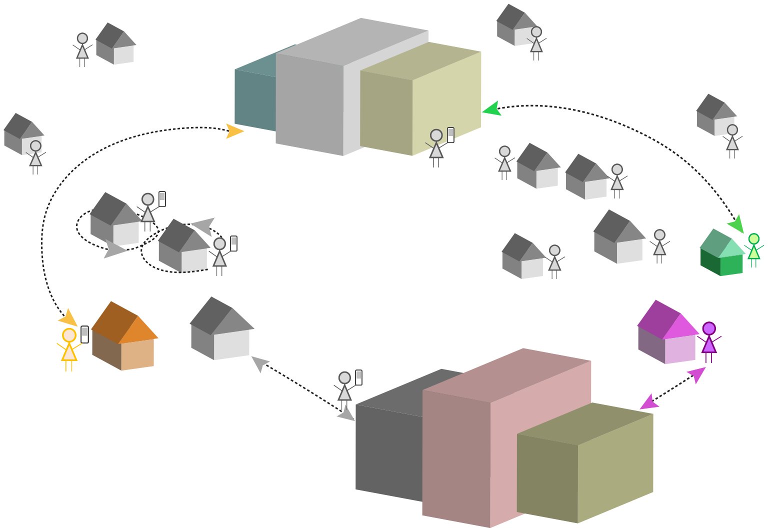


Figure 4-10: A spatial model where each individual diffuses either around their base or around a center.

Model III: A Spatial Agent Model with Centrality

This model is based on N individuals moving on the Euclidian plane and it captures both spatial and social aspects of the interactions between individuals. If two individuals are in close proximity and one of them is infected with COVID-19, it can be transmitted to the other individual with a certain infection probability. The Safe Blue strands are transmitted in a similar way.

For each individual there is a unique fixed *base* (home), located at a fixed point on the plane. There are also commercial/social *centers* that attract individuals, also located at fixed points on the plane. Individuals can visit these centers each day independent of others. Individuals can have social interactions with neighbours near their homes, or at the centers during their visit. The spatial movement of each individual is captured using biased random walks.

The model evolves over time units of days, yet within each day there are finer small discrete time units in which individuals make small steps on the plane, always gravitating towards a fixed point which is either their *base* or a *center*. This gravitation is modelled using a biased random walk which is described later. In addition to the small discrete time steps, individuals may also make quick (immediate) transitions swapping their gravitational point of attraction from *base* to a *center* and vise versa. Whenever swapping occurs the new location around the destination (*base* or *center*) is chosen as a random point near the destination. This models quick transport (e.g. driving) between home and commercial/social centers. See Figure4-10 for an illustration of the model.

The biased random walk that models the gravitation of individuals towards their unique *base* or towards a *center* depending if they are currently marked as "being at base", or "being at a center" is executed by taking steps in a direction as follows. For individual x, consider the angle, θ_x, between the individual and the attraction point (*base* or *center*). Then for some fixed parameter κ, we generate a random angle on $[-\pi, \pi]$, following the Von Mises distribution with density

$$f(\theta|\theta_x) = \frac{1}{2\pi I_0(\kappa)} e^{\kappa(\theta-\theta_x)},$$

where $I_0(\kappa)$ is the modified Bessel function of order 0. Then a step with an exponential distribution having a small step size with time dependent mean δ_t, is taken in the direction specified by the random angle.

The switching of the gravitational center is done as follows. On each day t, an individual spends time around (gravitating towards) the *center* during a time frame of length w_t, which is selected uniformly and independently over the day. During the remaining time of the day the individuals spend time around (gravitating towards) their base. The choice of which center to move to is randomly selected proportionally to the Euclidean distance between the person's current location and the center's location. Hence people generally move to the center closest to their base, but not always.

Social distancing is enforced by modifying w_t and δ_t over time. When w_t is low, individuals spend more time near their base and are less likely to meet others, while with w_t large, individuals spend more time at centers and more social interaction is likely to occur. Further when social distancing is reduced (or increased) w_t we also reduce (or increase) δ_t for individuals currently at base. This implies that when social distancing is enforced, individuals are closer to home and when social distancing is relaxed, more interaction occurs.

In each time step during which an infected individual has another individual with a proximity of less than r distance units, the other individual may be infected with probability p_C for COVID-19 and probability p_s for Safe Blues strand s. On each day, the probability of removing an infected individual is γ_C for COVID-19 and γ_s for Safe Blues strand s similarly to the previous models.

At the onset of the simulation, the base locations are selected randomly and are fixed for the duration of the simulation. Further centers have fixed locations.

Parameters used for the simulation run: The simulation run that we created for experimentation had $N = 5,000$ individuals of which $N^B = 1,000$ had Safe Blues enabled devices (hence $\eta = 0.2$). The locations of the bases were generated at onset using a mixture of two bivariate normal distributions with means $(25,0)$ and $(0,0)$. The respective covariance matrices were,

$$\begin{bmatrix} 100 & 0 \\ 0 & 80 \end{bmatrix} \qquad \text{and} \qquad \begin{bmatrix} 50 & 0 \\ 0 & 100 \end{bmatrix}.$$

The mixture weighting between the distributions is at 0.3 and 0.7 respectively. There were 2 centers located at $(10,15)$ and $(10,-15)$.

The simulation was run for 366 days where within each day there were 14 basic time steps. The parameter of the Von Misses distribution was set at $\kappa = 6$. The basic step size mean when not under social distancing was at $\delta_t = 2$. The proximity radius is $r = 0.0085$. The COVID-19 infection probability was set at $p_C = 0.04$ and the recovery/removal probability was at $\gamma_C = 0.1/14$. At onset a proportion of 0.0125 of the population was infected (uniformly initialized). Further, when individuals swap between a *center* and their *base*, they are located with a uniform angle around the destination and a distance that is exponentially distributed with mean 0.1.

For Safe Blues there were 50 strands denoted $s = 1,\dots,50$, all with the same

Time Range (in days)	lockdown strength l_t	w_t (in basic steps)	δ_t (used for base users)
1 – 7	0.2	10	1.62
8 – 14	0.3	9	1.43
15 – 98	0.8	3	0.48
99 – 126	0.0	12	2.0
127 – 210	0.3	9	1.43
211 – 217	0.2	10	1.62
218 – 366	0.0	12	2.0

Table 4.10: The social distancing parameters for Model III as functions of the lockdown strength, l_t, defined by $w_t = \lfloor 2l_t + 12(1 - l_t) \rfloor$ and $\delta_t = 0.1l_t + 2(1 - l_t)$ (used only for base users), where $\lfloor x \rfloor$ denotes the integer part of x.

parameters. The strands were released at onset $(t_{\text{start}} = 0)$. The removal probability was taken to be identical to COVID-19, and the infection probability was taken to be $p_s = p_C/\eta$. This rule follows the general guideline in Equation(4.1).

The parameters affecting social distancing, w_t and δ_t (used for base users), are expressed as functions of lockdown strength l_t that takes values in the interval $[0, 1]$, where $l_t = 0$ indicates no lockdown and $l_t = 1$ indicates highest possible lockdown. The details are summarized in Table4.10 below. The model was simulated using the `Numpy` library [121] in Python 3.

Chapter 5

Training and robustness of SciML models

After covering majorly application based studies in the previous 4 chapters, we return to the basics in this chapter. We look at the methodology of training scientific machine learning models and how to make them robust. Although the set of guidelines provided in this chapter are for specific examples, we hope that these guidelines serve as an inspiration for researchers studying a broader set of applications.

5.1 Introduction

Scientific Machine Learning (SciML)methods lie at the intersection of machine learning and scientific computing [8]. Many SciML epidemiological models use a methodology of integrating neural networks in ODEs or PDEs; called Universal Differential Equations (UDEs) [129]. This methodology has been shown to robustly perform estimation and forecasting tasks in a wide range of physical systems such as photonics [101], cancer therapy [171], climate modeling [133], quantum control [149] and chemical reaction systems [78]. There remain a number of unanswered questions when developing an epidemiological UDE model such as: (a) Can UDEs be used to replace complex compartment models with simpler graphical assumptions? (b) How robust is the UDE training procedure to varying neural network initializations? (c) Is symbolic recovery of missing epidemiological mechanisms possible through UDEs? (d) How much data is required to train UDEs? and (e) Do UDEs offer a significant advantage to fully non-mechanistic Neural Ordinary Differential Equations (Neural ODEs)? [20].

In this chapter, we study a cadre of epidemic models to give empirical answers to the above questions. The paper is structured as follows: In Section 5.2, we consider data generated by a complex compartment model and demonstrate UDE model prediction and symbolic recovery on this data. We also show the nuts and bolts of hyperparameter optimization required to make a UDE methodology robust to the neural network initialization. In Section 5.3, we introduce a framework called QSIR (Quarantine-Susceptible-Infected-Recovered) which is a UDE framework described in [31, 32]. Using the QSIR UDE, we demonstrate the a clear prediction performance advantage of UDEs compared to Neural ODEs. In Section **??**, we analyze data requirements in training a QSIR framework before concluding in section 5.4.

5.2 UDEs with simple graphical assumptions

In general, neural networks with arbitrary nonlinear activation functions are universal approximators [69, 27, 152]. UDEs which stand for universal approximators in differential equations, using approximators such as neural networks to augment scientific modeling frameworks like ordinary differential equations and partial differential equations (ODEs and PDEs). UDEs are shown to recover missing terms in governing equations, accelerate model simulation and accurately forecasting beyond training data; in wide range of physical and biological settings [129]. To train UDEs and optimize the neural network weights, one needs to take derivatives of functions of the ODE solution. For this, the adjoint equations are employed [20, 129] using the

DifferentialEquations.jl solver suite ([126]) in the Julia programming language [12].

To demonstrate the robustness of the UDE modeling paradigm in epidemiological frameworks, we consider data generated through the 5 compartment SIRHD (Susceptible-Infected-Recovered-Hospitalized-Dead) epidemiological model shown in equations 2.1 - 2.5.

$$\frac{\mathrm{d}S(t)}{\mathrm{d}t} = -\beta \frac{S(t)\,I(t)}{N} \tag{5.1}$$

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = \beta \frac{S(t)\,I(t)}{N} - \gamma_I I(t) - \mu_I I(t) - \rho I(t) \tag{5.2}$$

$$\frac{\mathrm{d}H(t)}{\mathrm{d}t} = \rho I(t) - \gamma_H H(t) - \mu_H H(t) \tag{5.3}$$

$$\frac{\mathrm{d}R(t)}{\mathrm{d}t} = \gamma_I I(t) + \gamma_H H(t) \tag{5.4}$$

$$\frac{\mathrm{d}D(t)}{\mathrm{d}t} = \mu_I I(t) + \mu_H H(t) \tag{5.5}$$

We will replace each structural component of this ODE with a neural network component. The resulting UDE model is shown in equations 2.6 - 2.10.

$$\frac{\mathrm{d}S(t)}{\mathrm{d}t} = -NN_{SI} \tag{5.6}$$

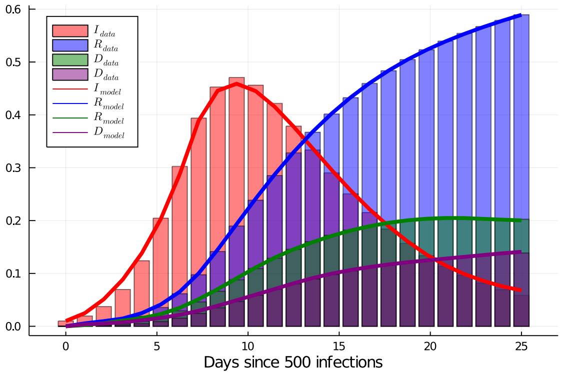
$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = NN_{SI} - NN_{IR} - NN_{ID} - NN_{IH} \tag{5.7}$$

$$\frac{\mathrm{d}R(t)}{\mathrm{d}t} = NN_{IR} + NN_{HR} \tag{5.8}$$

$$\frac{\mathrm{d}H(t)}{\mathrm{d}t} = NN_{IH} - NN_{HR} - NN_{HD} \tag{5.9}$$

$$\frac{\mathrm{d}D(t)}{\mathrm{d}t} = NN_{ID} + NN_{HD} \tag{5.10}$$

This form of a UDE does not include prior known mechanistic knowledge, but does include graphical assumptions in the architecture. For example, this declares



(a)

Figure 5-1: Comparison of the data obtained by training the neural networks in Equations (21-25) with the neural network architectures given by Table 3, with ground truth data.

that susceptible can only become infected, and death can only come from infected and hospitalized. These graphical assumptions also enforce conservation of population in the solution of the equations, as every negative term has a corresponding positive term making the sum of fluxes always zero. The underlying question is, is this level of prior knowledge sufficient to improve the training over standard knowledge-free learning scenarios?

All neural networks are trained using the ADAM optimizer with a step size of 10^{-4} with the relu activation function. Each neural network has 1 hidden layer and 10 units in each layer. Figure 5-1 shows the comparison of the SIRHD data with the optimized UDE model prediction. A good match is observed. Subsequently, we used a sparse regression technique called Sequential Thresholded Ridge Regression (STRRidge) algorithm [14] on each neural network output to reconstruct the missing dynamical equations for each neural network. STRRidge algorithm has a tunable sparsity parameter λ to control the sparsity of the obtained dominant terms. We varied this parameter from 0.0001 to 0.1 for each neural network, and choose the λ with the the lowest positive AIC score (Akaike Information Criteria) [13, 161] which strives to minimize the model error as well as its complexity. The recovered equations are shown in table 5.1. We can see a reliable symbolic recovery for each neural network.

	Actual Equations	STRRidge Active terms	STRRidge Equations	Minimum AICC
NN_{SI}	0.85 S I	1: SI	0.82 S I	22
NN_{IR}	0.1 I	1: I	0.097 I	32
NN_{ID}	0.05 I	1: I	0.022 I	22
NN_{IH}	0.025 I	1: I	0.049 I	28
NN_{HR}	0.02 H	1: H	0.018 H	33
NN_{HD}	0.002 H	1: H	0.002 H + 0.0005	45

Table 5.1: Recovered symbolic form of each of the neural networks using the Sequential Thresholded Ridge Regression (STRRidge) algorithm applied to the UDE model on the SIRHD data.

5.3 Retaining more structure in UDE modeling: the QSIR framework

Next, we look at another UDE model called the QSIR (Quarantine-Susceptible-Infected-Recovered) framework [31]. This model augments the standard SIR module through addition of a neural network for a region-based nonlinear quarrentine model. This makes the resultant model much more powerful and expressive than the standard SIR framework. Following are the equations governing this model

$$\frac{\mathrm{d}S(t)}{\mathrm{d}t} = -\beta \frac{S(t)\, I(t)}{N} \tag{5.11}$$

$$\begin{aligned}\frac{\mathrm{d}I(t)}{\mathrm{d}t} &= \beta \frac{S(t)\, I(t)}{N} - \big(\gamma + Q(t)\big)\, I(t) \\ &= \beta \frac{S(t)\, I(t)}{N} - \big(\gamma + \mathrm{NN}(W, I)\big)\, I(t) \end{aligned} \tag{5.12}$$

$$\frac{\mathrm{d}R(t)}{\mathrm{d}t} = \gamma I(t) + \delta T(t) \tag{5.13}$$

$$\frac{\mathrm{d}T(t)}{\mathrm{d}t} = Q(t)\, I(t) - \delta T(t) = \mathrm{NN}(W, I)\, I(t) - \delta T(t). \tag{5.14}$$

In the QSIR model, the term $I(t)$ denotes the infected population still having contact with the susceptibles, as done in the standard SIR model; while the term $T(t)$ denotes the infected population who are effectively quarantined and isolated. Further

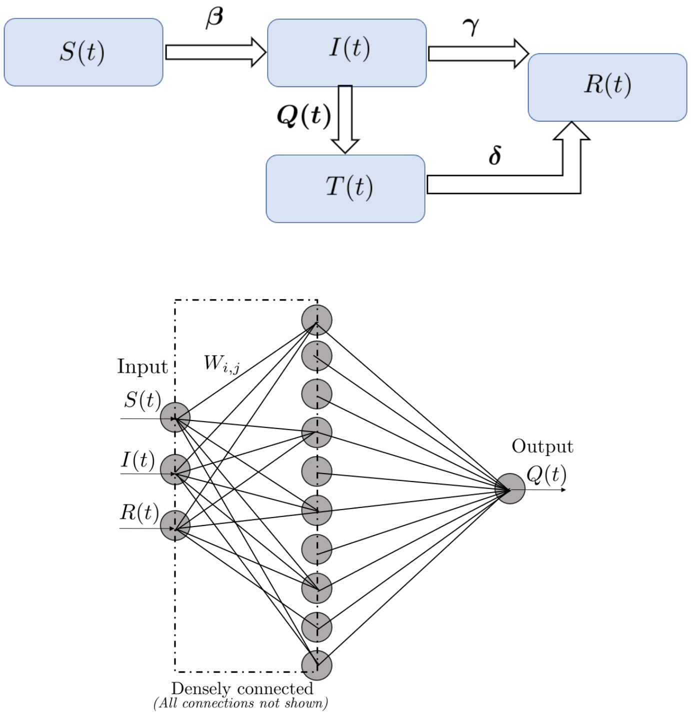


Figure 5-2: [Illustration of the QSIR Model and neural network architecture] (a) Schematic of the augmented QSIR model considered in the present study. (b) Schematic of the neural network architecture used to learn the quarantine strength function $Q(t)$. Here $T(t)$ represents the quarantined infected population prescribed by the quarantine strength rate $Q(t)$.

we introduce an additional recovery rate δ which quantifies the rate of recovery of the quarantined population. Thus, we can write an expression for the quarantined infected population $T(t)$ as

$$\frac{dT(t)}{dt} = Q(t)I(t) - \delta T(t) \tag{5.15}$$

We represent $Q(t)$ as a n layer-deep neural network with weights $W_1, W_2 \ldots W_n$, activation function r and the input vector $U = (S(t), I(t), R(t))$ as

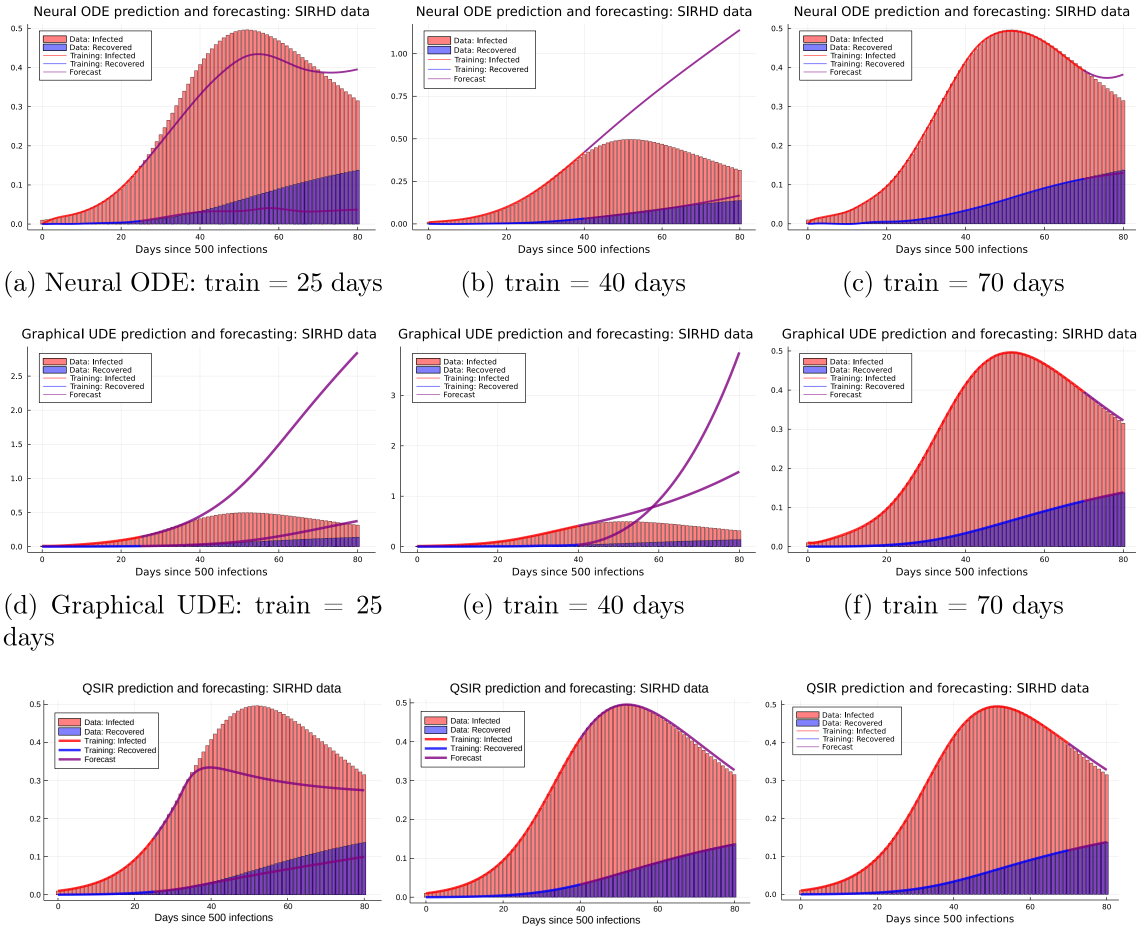
$$Q(t) = r\left(W_n r\left(W_{n-1} \ldots r\left(W_1 U\right)\right)\right) \equiv \mathrm{NN}(W, U) \tag{5.16}$$

For the actual implementation, we choose a $n = 2$-layer densely connected neural network with 10 units in the hidden layer and the ReLU activation function. This choice was because we found sigmoidal activation functions to stagnate. The final model is described by a total of 54 tunable parameters. The neural network architecture schematic is shown in figure 5-2b. More details about the model initialization and parameter estimation methods is given in [31].

We will subsequently train the QSIR model on the SIRHD data which is generated from equations 2.1 - 2.5.

5.3.1 Retaining structure improves forecasting abilities

Figure 5-3 shows the prediction and forecasting capability of a plain Neural ODE model [20], graphical UDE model (shown in equations 2.6 - 2.10 and the QSIR model considered in this section). The underlying data for all three models is generated by the 5 compartment SIRHD model given in equations 2.1 - 2.5. The QSIR model retains much more physical structure than the plain Neural ODE and the graphical UDE model which do not have information governing the underlying interactions. We notice that due to encoding a greater physical structure, the QSIR model is able to forecast well on the least amount of training data, compared to the other two models.



(a) Neural ODE: train = 25 days

(b) train = 40 days

(c) train = 70 days

(d) Graphical UDE: train = 25 days

(e) train = 40 days

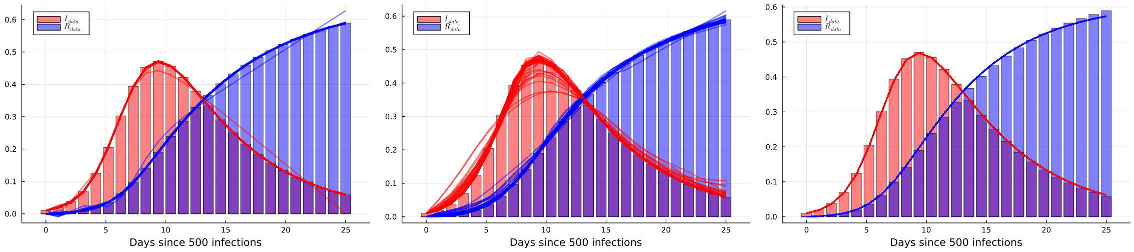
(f) train = 70 days

(g) QSIR UDE: train = 25 days

(h) train = 40 days

(i) train = 70 days

Figure 5-3: Figure shows the prediction and forecasting performance for (a, b, c): the neural ODE model, (d, e, f): the graphical UDE model and (g, h, i): the QSIR UDE model. Data for all models is generated by the 5 compartment model shown in equations 2.1 - 2.5.



(a) Neural ODE (b) Graphical UDE (c) QSIR UDE

Figure 5-4: Figure shows the comparison of the data generated by the 5 compartment SIRHD model in equations 2.1-2.5 with the model predictions for 100 random initializations for (a) the Neural ODE model, (b) the Graphical UDE model and (c) the QSIR UDE model. Number of iterations for each initialization = 15000.

Thus, using just a 4 compartment model (QSIR), augmentation with a neural network module enables us to capture the data generated by the 5 compartment SIRHD model and also make accurate forecasting predictions.

In Section 7.2 of the Appendix, we show that the QSIR model has reliable prediction and forecasting capabilities even on a more complex 9 compartment model data.

5.3.2 Robustness of the models

Figure 5-4 shows the comparison of the data generated by the 5 compartment SIRHD model in equations 2.1-2.5 with the model predictions for 100 random initializations for (a) the Neural ODE model, (b) the Graphical UDE model and (c) the QSIR UDE model. The optimal hyperparameters for each model were obtained by a careful hyperparameter optimization procedure, illustrated in the Appendix 7.1 for the Graphical UDE framework. A similar optimization procedure was performed on the Neural ODE model and the QSIR UDE model. We can see that when the parameters of the models are optimized, all 100 trajectories of the models converge; implying the robustness of the models to the random initializations of the neural network.

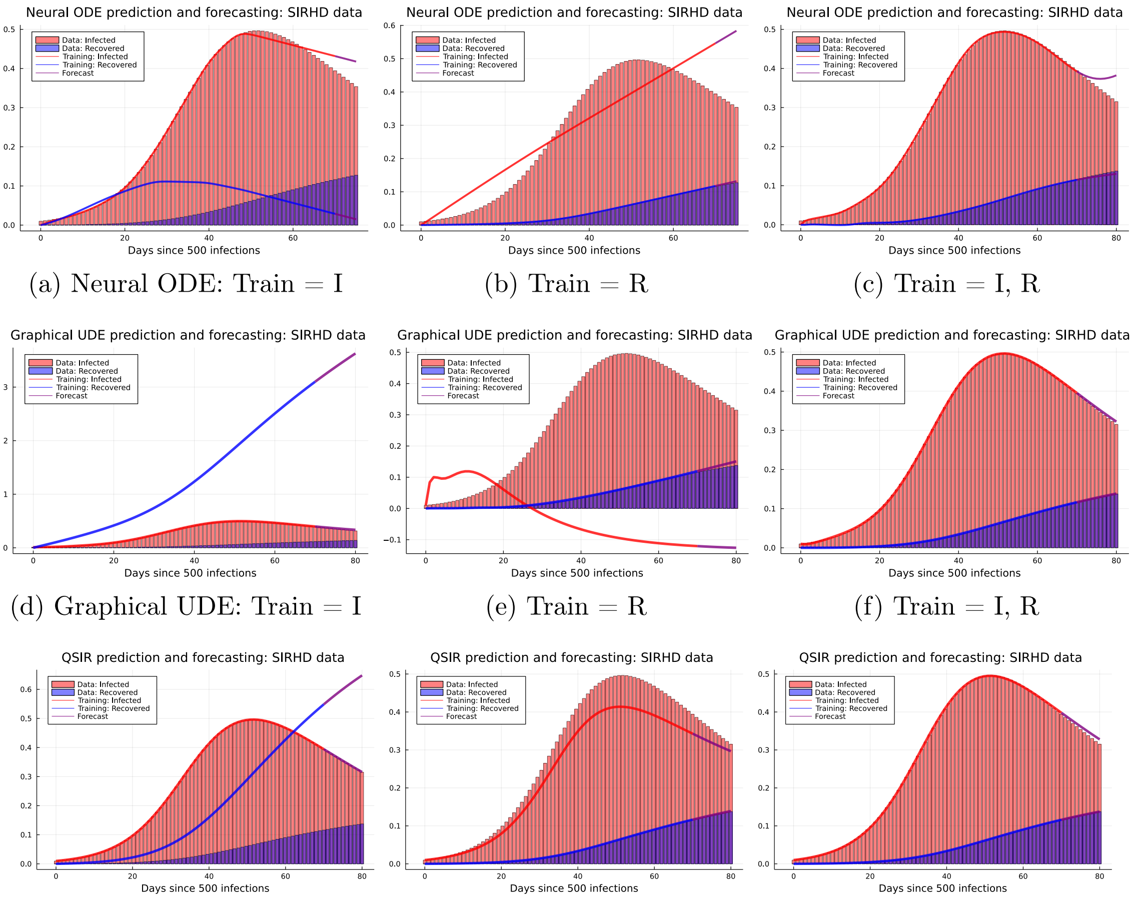
5.3.3 Training with subsets of data

Figure 5-5 shows the effect of training subsets of data on the prediction and forecasting performance for (a, b, c): the neural ODE model, (d, e, f): the graphical UDE model and (g, h, i): the QSIR UDE model. Data for all models is generated by the 5 compartment SIRHD model shown in equations 2.1 - 2.5. We can see that for the Neural ODE and the Graphical UDE model which impose minimal physical structure, when the training data only consisted of the infected (I) compartment or the recovered (R) compartment, the model prediction and forecasting is poor for the remaining compartment. Since no physics is embedded, the remain compartment values can be very unreasonable (very high or even negative). The QSIR model does slightly better than the other models, when trained with limited information. Since the QSIR models embeds a lot of physical structure, it prevents unreasonable values for the compartment population and generally captures the trend of the compartment on which it is not trained on (see figures 5-5 g, h).

We now take a deeper look at the QSIR model in terms of: (a) Training the QSIR model with multiple subsets of observables and (b) parameter identifiability and robustness.

5.4 Conclusion

In conclusion, the study demonstrates the application of Scientific Machine Learning (SciML) methods in the field of epidemiology. We show that SciML based UDE models can be used as surrogates for complex compartment models in which one or more of the compartments can be replaced by neural networks. Not only does this increase the expressivity of the models, but symbolic regression techniques also make these models interpretable. To obtain the optimal performance from a UDE model, we demonstrate the importance of performed a detailed hyperparameter optimization. The most important parameters in an UDE training procedure are found to be:



(a) Neural ODE: Train = I

(b) Train = R

(c) Train = I, R

(d) Graphical UDE: Train = I

(e) Train = R

(f) Train = I, R

(g) QSIR UDE: Train = I

(h) Train = R

(i) Train = I, R

Figure 5-5: Figure shows the effect of training subsets of data on the prediction and forecasting performance for (a, b, c): the neural ODE model, (d, e, f): the graphical UDE model and (g, h, i): the QSIR UDE model. Data for all models is generated by the 5 compartment model shown in equations 2.1 - 2.5.

(a) Optimizer: step size, (b) ODE solver: stiff/non-stiff, relative tolerance, adjoint method, (c) architecture: number of hidden layers, nodes and activation functions of the neural networks.

UDE models in which one or more compartments are replaced by neural networks can be used to capture missing model dynamics. One such model (the QSIR model) has been demonstrated in the study to capture the effects of quarantine which are otherwise missed by the simple SIR model. We subsequently show that the structure embedded in UDE models makes them much more robust than Neural ODEs at reliable forecasting.

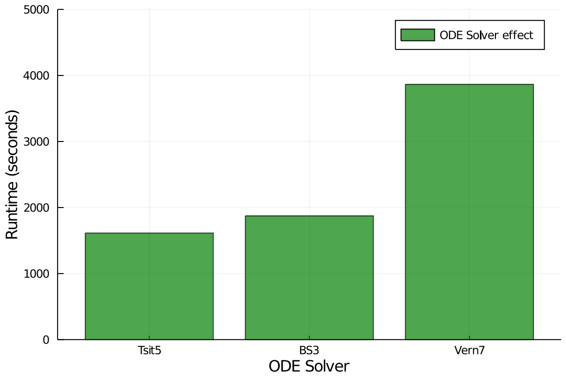
Encoding uncertainty quantification in Neural ODEs/UDEs modeling and estimation can be done if Neural ODEs/UDEs are coupled with high fidelity Bayesian estimation frameworks. This is explored in Chapter 6.

5.5 Appendix

5.5.1 Hyperparameter optimization and robustness analysis of the UDE framework

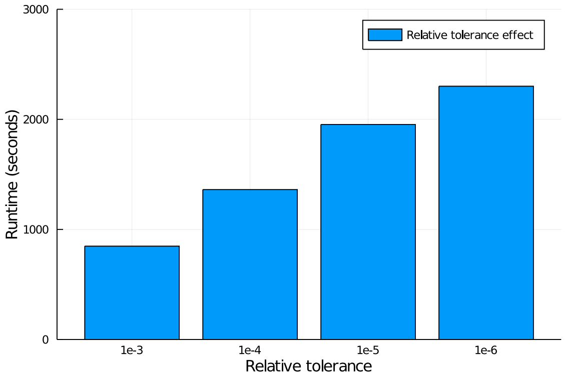
The following details the hyperparameter optimization process for the graphical UDE. We note that these results are not generalizable to all types of equations.

We tested the following activation functions for 100 random UDE initializations: (a) tanh, (b) relu, (c) sigmoid and (d) leakyrelu. relu and tanh were the activation functions which led to more than 70% converging simulations of all initilizations tested. The remaining activation functions were found to be unreliable, potentially because of instabilities with the initial neural networks over a long integration time. Dampening the coefficients or multiple shooting could be used to improve the robust-



(a)

Figure 5-6: Timing performance by varying the ODE Solvers. A relative tolerance of $1e-4$ with the InterpolatingAdjoint method was used.

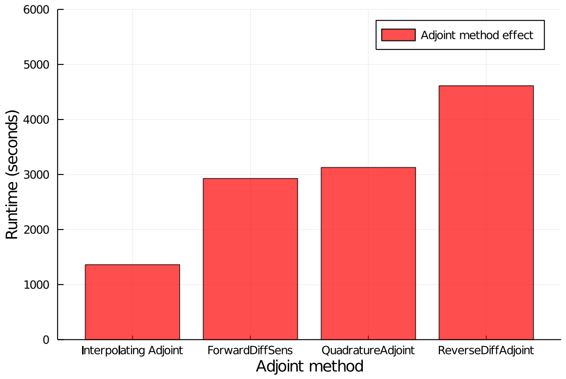


(a)

Figure 5-7: Timing performance by varying the relative tolerance. The adjoint method used was InterpolatingAdjoint() with Tsit5() ODE solver.

ness in those cases, though here we simply chose to use the relu. We simulated an ADAM optimizer with the following step size variations: (a) $1e-2$, (b) $1e-3$, (c) $1e-4$ and (d) $1e-5$. It was found that stepsizes of $1e-2, 1e-3$ lead to exploding gradient issues which leads to losses of $O(1e11)$. The optimal training hyperparameter found which gave fast performance was found to be $1e-4$.

Another factor which has a major influence on the UDE performance is the ODE solver. We tested 3 non-stiff solvers: Tsit5, BS3 and Vern7 which are generally considered fast and accurate in differential equations literature [126]. All of these solvers led to similar training accuracy, with Tsit5 and BS3 showing the most optimal timing

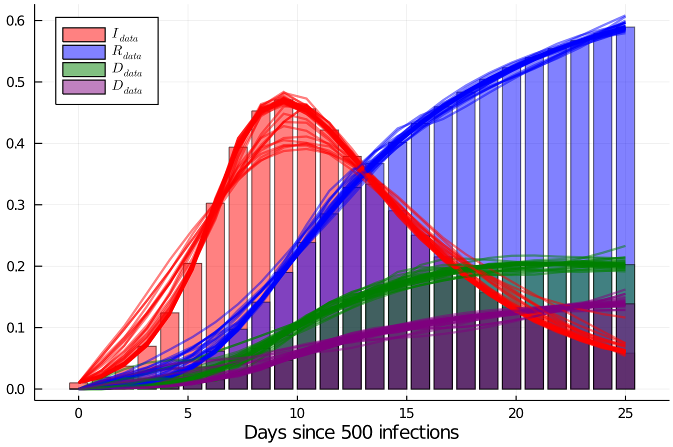


(a)

Figure 5-8: Timing performance by varying the adjoint methods. A relative tolerance of $1e-4$ with Tsit5() ODE solver was used.

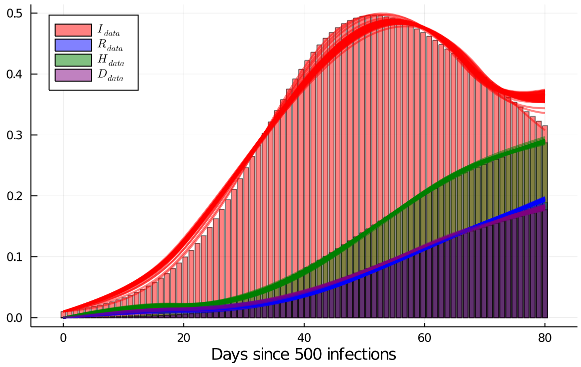
performance as seen in figure 5-6. We also compared performance of stiff solvers like Rosenbrock23(). TRBDF2(). These solvers were an order of magnitude slower than the non-stiff solvers, indicating the equation was non-stiff. In adaptive time stepping methods, the solver relative tolerance controls the errors for the state variables and hence governs the accuracy of the solution. We simulated the Tsit5 ODE solver with the following relative tolerance variations: (a) $1e-3$, (b) $1e-4$, (c) $1e-5$ and (d) $1e-6$. The UDE solution accuracy was seen to be low for a relative tolerance value of $1e-3$ but a good match similar to figure 5-1 was seen for the lower values. However, as seen in figure 5-7, the computational time increases linearly with the relative tolerance. Thus, the value of relative tolerance which balances computational speed with solution accuracy was found to be $1e-4$.

The choice of adjoint method plays a crucial role during backpropagation, when calculating the derivative of an ODE solver with respect to a loss function. Methods vary in stability and memory usage, and depending on the problem to solve, the adjoint method should be chosen accordingly. In this study, we implemented InterpolatingAdjoint,ForwardDiffSensitivity, QuadratureAdjoint and ReverseDiffAdjoint adjoint methods [129]; and tested the timing and training performance. Good training accuracy was seen for all adjoint methods, with the minimum computational time seen for the InterpolatingAdjoint method (figure 5-8. This make the Interpo-



(a)

Figure 5-9: Comparison of the SIRHD data with optimized model prediction for 100 random initializations. Number of iterations for each initialization = 15000.



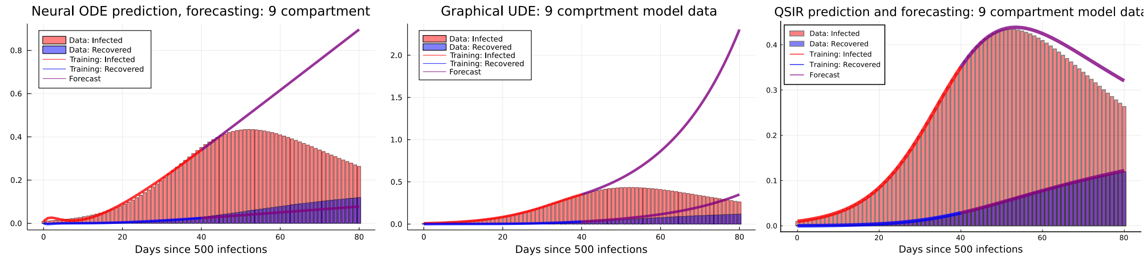
(a)

Figure 5-10: Comparison of the SIRHD data with optimized model prediction for 100 random initializations; shown for a much larger dataset.

latingAdjoint an ideal choice for our study. Note however that this result is not generalizable and other models will have different behaviors.

For the optimized parameters described above, figure 5-9 shows the comparison of the underlying data with the optimized model prediction for 100 random initializations. We can see that most of the UDE trajectories show a good match with the training data. About 85 of the 100 runs converge to a very low loss.

Subsequently, we tested these hyperparameters and how they control the UDE



(a) (b) (c)

Figure 5-11: Figures show the comparison between the prediction and forecasting ability of (a) a plain Neural ODE model, (b) the graphical UDE model shown in (8.11 - 8.20) and (c) the QSIR model shown in (3.1-3.4); for data generated by the 9 compartment model.

training on a three times larger dataset than the one shown in figure 5-9. Again we observed that these hyperparameters play a crucial role in the training process and once they are optimized, a large number of converging trajectories are observed. The converging trajectories are shown in figure 5-10. Compared to figure 5-9, a large number of iterations (120000 compared to 15000) are needed for convergence.

Thus, in this section, we have detailed the hyperparameters which need to be taken into consideration for optimizing an UDE training process. Based on the test case under consideration, the optimal values of these hyperparameters maybe different than those presented here; but its crucial to exhaustively consider all parameters to effectively train a UDE model.

5.5.2 QSIR framework applied to a 9 compartment model

In order to test the robustness of the QSIR framework for prediction and forecasting, we consider a 9 compartment model as follows

$$\frac{\mathrm{d}S(t)}{\mathrm{d}t} = -\frac{\tau_{SI}\, S(t)\, I(t)}{N} \tag{5.17}$$

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = \frac{\tau_{SI}\, S(t)\, I(t)}{N} - \tau_{IR}I(t) - \tau_{ID}I(t) - \tau_{IH}I(t) - \tau_{A}I(t) - \tau_{B}I(t) - \tau_{C}I(t) - \tau_{E}I(t) \tag{5.18}$$

$$\frac{\mathrm{d}R(t)}{\mathrm{d}t} = \tau_{IR}I(t) \tag{5.19}$$

$$\frac{\mathrm{d}H(t)}{\mathrm{d}t} = \tau_{IH}I(t) \tag{5.20}$$

$$\frac{\mathrm{d}H(t)}{\mathrm{d}t} = \tau_{ID}I(t) \tag{5.21}$$

$$\frac{\mathrm{d}A(t)}{\mathrm{d}t} = \tau_{A}I(t) \tag{5.22}$$

$$\frac{\mathrm{d}B(t)}{\mathrm{d}t} = \tau_{B}I(t) \tag{5.23}$$

$$\frac{\mathrm{d}C(t)}{\mathrm{d}t} = \tau_{C}I(t) \tag{5.24}$$

$$\frac{\mathrm{d}E(t)}{\mathrm{d}t} = \tau_{E}I(t) \tag{5.25}$$

$$\tag{5.26}$$

We generate data from the above model and the QSIR framework for prediction and forecasting. Similar to the 5 compartment model, the graphical UDE form the 9 compartment model is

$$\frac{\mathrm{d}S(t)}{\mathrm{d}t} = -NN_{SI} \tag{5.27}$$

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = NN_{SI} - NN_{IR} - NN_{ID} - NN_{IH} - NN_{IA} - NN_{IB} - NN_{IC} - NN_{IE} \tag{5.28}$$

$$\frac{\mathrm{d}R(t)}{\mathrm{d}t} = NN_{IR} \tag{5.29}$$

$$\frac{\mathrm{d}H(t)}{\mathrm{d}t} = NN_{IH} \tag{5.30}$$

$$\frac{\mathrm{d}H(t)}{\mathrm{d}t} = NN_{ID} \tag{5.31}$$

$$\frac{\mathrm{d}A(t)}{\mathrm{d}t} = NN_{IA} \tag{5.32}$$

$$\frac{\mathrm{d}B(t)}{\mathrm{d}t} = NN_{IB} \tag{5.33}$$

$$\frac{\mathrm{d}C(t)}{\mathrm{d}t} = NN_{IC} \tag{5.34}$$

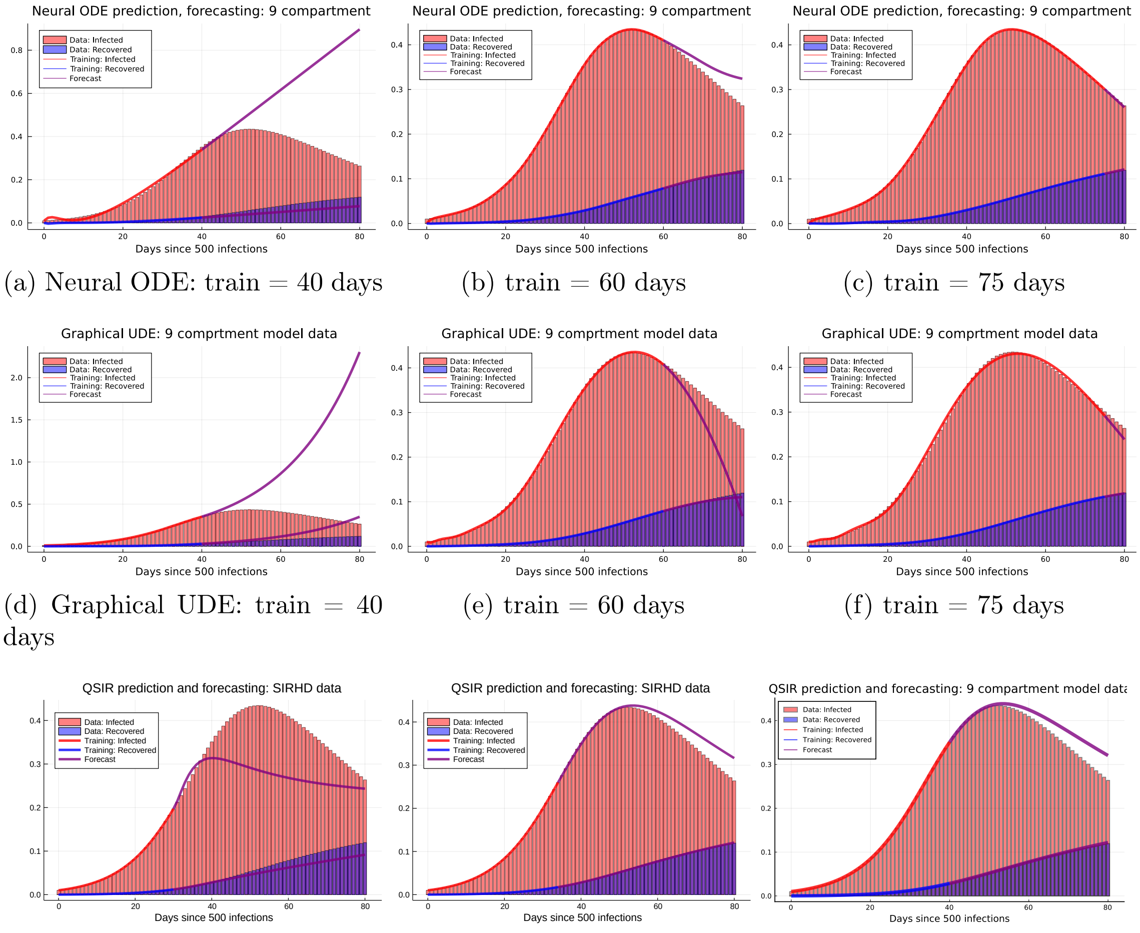
$$\frac{\mathrm{d}E(t)}{\mathrm{d}t} = NN_{IE} \tag{5.35}$$

$$\tag{5.36}$$

Figure 5-11 shows that even for data generated by a 9 compartment model, the QSIR model shows reliable prediction and forecasting capabilities; compared to the a plain Neural ODE [20] and the graphical UDE model (equations 8.11 - 8.20).

Figure 5-12 shows the data requirements for reliable forecasting comparison between a plain Neural ODE model, a graphical UDE model and a QSIR UDE model; with data being generated by the 9 compartment model shown in equations 7.11 - 7.20. We notice that due to encoding a greater physical structure, the QSIR model is able to forecast well on the least amount of training data, compared to the other two models.

Figures 5-11 and 5-12 indicate the reliability of the QSIR UDE for efficient prediction and forecasting on data generated by complex compartment models.



(a) Neural ODE: train = 40 days

(b) train = 60 days

(c) train = 75 days

(d) Graphical UDE: train = 40 days

(e) train = 60 days

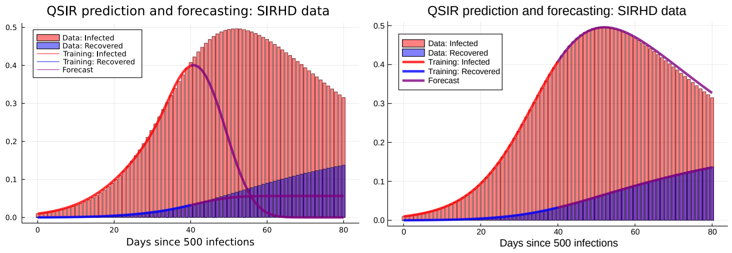
(f) train = 75 days

(g) QSIR UDE: train = 30 days

(h) train = 34 days

(i) train = 40 days

Figure 5-12: Figure shows the prediction and forecasting performance for (a, b, c): the neural ODE model, (d, e, f): the graphical UDE model and (g, h, i): the QSIR UDE model. Data for all models is generated by the 9 compartment model shown in equations 7.11 - 7.20.



(a) Polynomial model QSIR (b) Neural network QSIR

Figure 5-13: Figure shows the prediction and forecasting performance of (a) Polynomial model and (b) Neural network QSIR UDE

5.5.3 Comparison with polynomial models

What if polynomial models are used to represent the quarantined population in the QSIR model? We consider a polynomial model with the same number of parameters as a 2 layer neural network with 10 hidden units which we use in the QSIR framework. Figure 5-13 shows that the forecasting performance for the polynomial model is much poorer than the neural network based model. This is in accordance with [131], who showed that Neural Ordinary Differential Equations (Neural ODEs) which are the heart of SciML frameworks, consistently outperform non linear and classical linear methods. Even by increasing the polynomial complexity to include higher order terms, we could not get the forecasting performance achieved by a simple 2 layer neural network with 10 hidden units. This indicates that performing optimization procedures is easier for neural network based frameworks, as compared to polynomial based frameworks, possibly due to non-smooth loss function landscapes encountered while training the latter [146].

Chapter 6

Error Quantification in SciML models

In the last study of this thesis, we look at uncertainty quantification in scientific machine learning (SciML) models. We look at this through the lens of two popular SciML frameworks: Neural ODEs [20] and Universal Differential Equations (UDEs) [129].

6.1 Introduction

The underlying scientific laws describing the physical world around us are often prescribed in terms of ordinary differential equations (ODEs). Recently, Neural Ordinary Differential Equations [20] has emerged as a powerful framework for modeling physical simulations without explicitly defining the ODEs governing the system, but learning them via machine learning. By noticing that in the limit of infinite layers, a ResNet module [61] behaves as a continuous time ODE, Neural ODEs allow the coupling of neural networks as expressive function transformations, and powerful purpose built ODE solvers. While [20] explored a number of applications of the Neural ODE framework, their success in a Bayesian inference framework remains unexplored.

Simultaneously, there has been an emergence of efficient Bayesian inference methods suited for high-dimensional parameter systems, such as the No-U-Turn MCMC sampler (NUTS) [67] which is an extension of the Hamiltonian Monte Carlo Algorithm, and Stochastic Gradient Markov Chain Monte Carlo (SGMCMC) methods like Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) [21] and Stochastic Gradient Langevin Descent (SGLD) [164].

A number of works in literature explored the use of Bayesian methods to infer parameters of systems defined by ODEs [96, 160, 47, 70] and others used Bayesian methods to infer parameters of neural network models, e.g. [81, 97, 75]. Bayesian neural networks in particular has been an active area of research for a while. The readers are referred to the excellent recent tutorial by [81] for an overview of recent advances in the field. However, this prompts the question: "Can Bayesian learning frameworks be integrated with Neural ODE's to robustly quantify the uncertainty in the weights of a Neural ODE?"

In an effort to address this question, we demonstrate and compare the integration of Neural ODEs with the following methods of Bayesian Inference: (a) The No-U-Turn MCMC sampler (NUTS), (b) Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) and (c) Stochastic Langevin Gradient Descent (SGLD). We present successful results on classical physical systems and on standard machine learning datasets (using GPU acceleration); especially on the standard MNIST dataset, we achieve a test ensemble accuracy of 98.5% on 10000 images. This is a performance competitive with current state-of-the-art image classification methods, which meanwhile lack our method's ability to quantify the confidence in its predictions. Subsequently, we premiere the integration of Bayesian Neural ODEs with variational inference, the predictive power of which improves with the introduction of normalizing flow.

Finally, advancing from learning a physical system's differential equations via Bayesian Neural ODEs, we consider the problem of recovering missing terms from a dynamical system using universal differential equations (UDEs) [129]. Using the Preconditioned SGLD variation of SGLD, we demonstrate the predictive success of Bayesian UDEs on (a) a predator-prey model and (b) the epidemiological model of COVID-19 spread. Through this, we present a viable method for the probabilistic quantification of epistemic uncertainties via a hybrid machine-learning and mechanistic-model-based technique.

Our approach differs from that of [3] who mainly looked at integration of Bayesian methods with Neural SDE's, and not Neural ODEs describing physical systems or large scale deep learning datasets like the MNIST dataset, which we consider here.

In this study, we used the Julia differentiable programming stack [127] to compose the Julia differential equation solvers [126] with the Turing probabilistic programming language [42, 168]. The study was performed without modifications to the underlying libraries due to the composability afforded by the differentiable programming stack.

6.2 Results

We illustrate the robustness of the Bayesian Neural ODE framework through the following case studies:

Case study 1: Spiral ODE

The Spiral ODE model is prescribed by the following system of equations:

$$\frac{du_1}{dt} = -\alpha u_1^3 + \beta u_2^3 \tag{6.1}$$

$$\frac{du_2}{dt} = -\beta u_1^3 - \alpha u_2^3 \tag{6.2}$$

Case study 2: Lotka-Volterra ODE

The Lotka-Volterra predator-prey model is prescribed by the following system of equations:

$$\frac{du_1}{dt} = -\alpha u_1 - \beta u_1 u_2 \tag{6.3}$$

$$\frac{du_2}{dt} = -\delta u_2 + \gamma u_1 u_2 \tag{6.4}$$

6.2.1 Bayesian Neural ODE: NUTS Sampler

The No-U-Turn-Sampler (NUTS) is an extension of the Hamiltonian Monte Carlo (HMC) algorithm. Through a recursive algorithm, NUTS automatically determines when the sampler should stop an iteration, and thus prevents the need to specify user defined parameters, like the number of steps L. In addition, through a dual averaging algorithm, NUTS adapts the step size ϵ throughout the sampling process.
We define the parameters of the d dimensional Neural ODE by θ. The action of the Neural ODE on an input value u_0 generates an output $\tilde{Y} = \text{NNODE}_{\theta}(u_0)$. The input data is denoted by $\hat{Y}$. The loss function, L is defined as

$$L(\theta) = \sum_{i=1}^{d} ||\hat{Y}_i - \tilde{Y}_i||^2 \tag{6.5}$$

The model variables θ and the momentum variables r are drawn from the joint distribution

$$p(\theta, r) \propto \exp[\mathcal{L}(\theta) - \frac{1}{2}r.r] \tag{6.6}$$

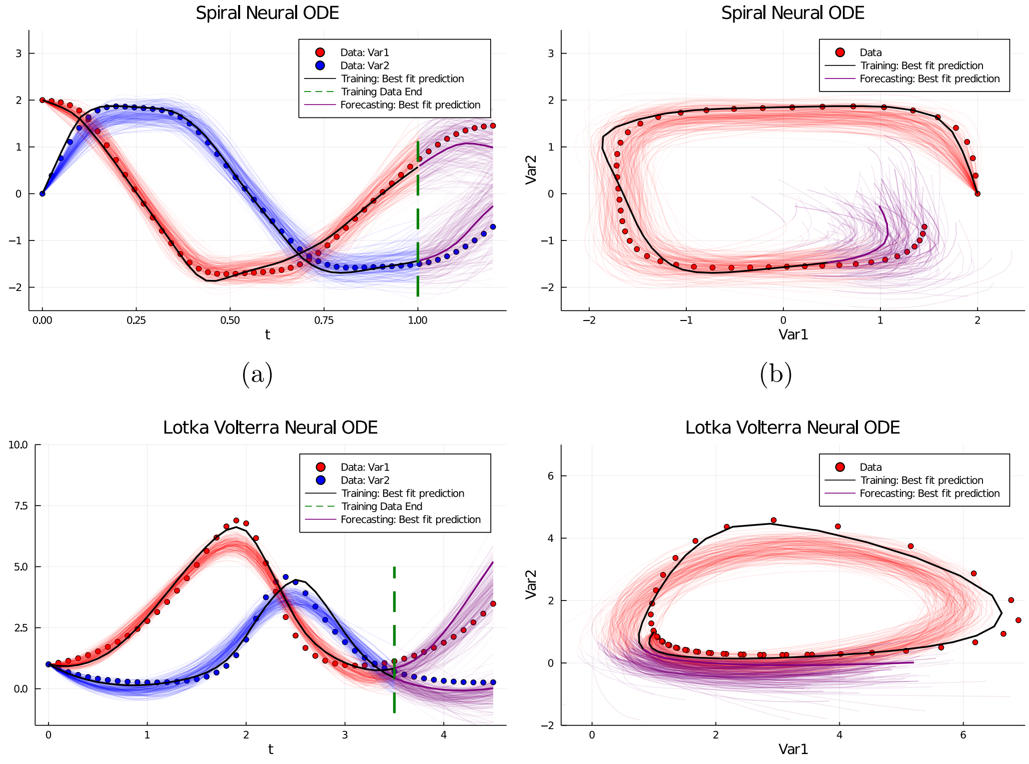
where $\mathcal{L}$ is the logarithm of the joint density of θ. In terms of a physical analogy, if θ denotes a particle's position, then $\mathcal{L}$ can also be viewed as the negative of the potential energy function and $\frac{1}{2}r.r$ denotes the kinetic energy of the particle.

In the Bayesian Neural ODE framework, we define $\mathcal{L}(\theta)$ as

$$\mathcal{L}(\theta) = -\sum_{i=1}^{d} ||\hat{Y}_i - \tilde{Y}_i||^2 - \theta.\theta \tag{6.7}$$

The $\theta.\theta$ term indicates the use of Gaussian priors. We adapt the step size of the leapfrog integrator using Nesterov's dual averaging algorithm [67] with δ as the target acceptance rate. Finally, we define the number of warmup samples as n_w and the number of posterior samples collected as n_p.

We apply the Bayesian Neural ODE framework outlined above to case studies 1 and 2 given in Equations 1-4. For case study 1, we use $\alpha = 0.1, \beta = 2$. In the NUTS algorithm, we use $\delta = 0.45, n_w = 1000, n_s = 500$. For case study 2, we use



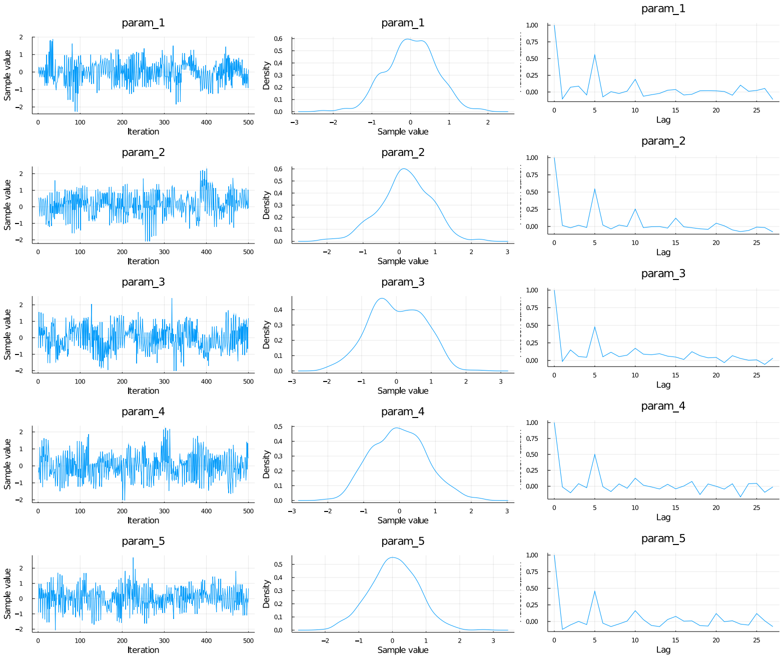
(a)

(b)

(c)

(d)

Figure 6-1: Comparison of the Bayesian Neural ODE: NUTS prediction and estimation compared with data for (a,b) Case study 1 and (c,d) Case study 2.



(a) (b)

Figure 6-2: For the Spiral ODE example (Equations 1-2), using the NUTS framework, figure shows: (a) Trace plots and Density plots of the posterior and (b) auto-corelation plot for the first 5 parameters.

$\alpha = 1.5, \beta = 1, \gamma = 3, \delta = 1$; with $\delta = 0.45, n_w = 500, n_s = 1000$. 2 layers with 50 units in each layer and tanh activation function was used as the neural ODE architecture for both examples.

From figure 6-1, we can see that the Bayesian Neural ODE: NUTS prediction and forecasting for both case studies outlined in Equations 1-4 are consistent with the ground truth data.

Figure 6-2a showing the posterior density and trace plots for the first 5 parameters of the Spiral ODE example, shows that the samples are well mixed. The quick decay seen in the auto-corelation plot shown in figure 6-2b also indicates a fast mixing Markov chain. This is also confirmed by the effective sample size extracted for the posterior chain of 500 samples, which shows values of $362, 470, 134, 509, 661$ for the first 5 parameters. Similar well mixed plots are seen for all parameters, but are not

Table 6.1: Spiral ODE: Effect of NUTS acceptance ratio and Neural ODE architecture. Number of warmup samples, $n_w = 500$ and number of posterior samples, $n_s = 500$ for all cases shown. The minimum loss value obtain is similar in all cases shown.

δ	Units	Layers	Time (s)
0.45	5	2	480
0.45	10	2	900
0.45	50	2	2100
0.45	100	2	3900
0.45	10	3	3300
0.45	10	4	10200
0.65	50	2	3400
0.85	50	2	7900
0.95	50	2	8600

shown here for the sake of brevity.

Table 6.1 shows the effect of NUTS acceptance ratio and Neural ODE architecture on the Bayesian Neural ODE performance for the Spiral ODE example. The minimum loss value obtain is similar in all cases shown. Thus, we see that even the smallest neural architecture with 2 layers and 5 units in each layer gives the optimal loss performance, and with a considerably better timing performance. Among different NUTS acceptance ratios (δ) tested, the best timing performance is given by the lowest acceptance ratio, $\delta = 0.45$.

6.2.2 Bayesian Neural ODE: SGHMC

The Stochastic Gradient Hamiltonian Monte Carlo Sampler (SGHMC) is a method that combines HMC's effective state space exploration with stochastic gradient methods' computational efficiencies [21]. SGHMC injects friction to the "momentum" auxiliary variables that parameterize the target distribution's Hamiltonian dynamics.

The Hamiltonian function $H(\theta, r) = U(\theta) + \frac{1}{2} r^T M^{-1} r$ measures the total "energy" of a system with position variables θ and momentum variables r. The potential energy function is given by $U = -\sum_{x \in \mathcal{D}} \log p(x|\theta) - \log p(\theta)$; mass matrix M and r define the kinetic energy term.

To sample from the posterior distribution $p(\theta|\mathcal{D})$, HMC considers generating samples from the joint distribution $\pi(\theta, r) \propto \exp(H(\theta, r))$, proposing samples according to the Hamiltonian dynamics:

$$d\theta = M^{-1} r dt \tag{6.8}$$

$$dr = -\nabla U(\theta) dt \tag{6.9}$$

Here, SGHMC adds a "friction" term to the momentum update. In practice, we consider $\nabla \tilde{U}$, a noisy estimate of ∇U. Similarly, $\hat{B}$ is defined as an estimate of $B(\theta) = \frac{1}{2}\epsilon V(\theta)$, the diffusion matrix contributed by the covariance of the stochastic gradient noise $V(\theta)$. SGHMC additionally introduces a user-specified friction term $C \geq \hat{B}$. In total, by defining $v = \epsilon M^{-1} r$, $\eta = \epsilon^2 M^{-1}$, $\alpha = \epsilon M^{-1} C$, and $\hat{\beta} = \epsilon M^{-1} \hat{B}$ for stepsize ϵ, SGHMC iteratively updates the variables for sampling according to:

$$\theta_t := \theta_t + \Delta\theta_t \tag{6.10}$$

$$v_t := v_t + \Delta v_t \tag{6.11}$$

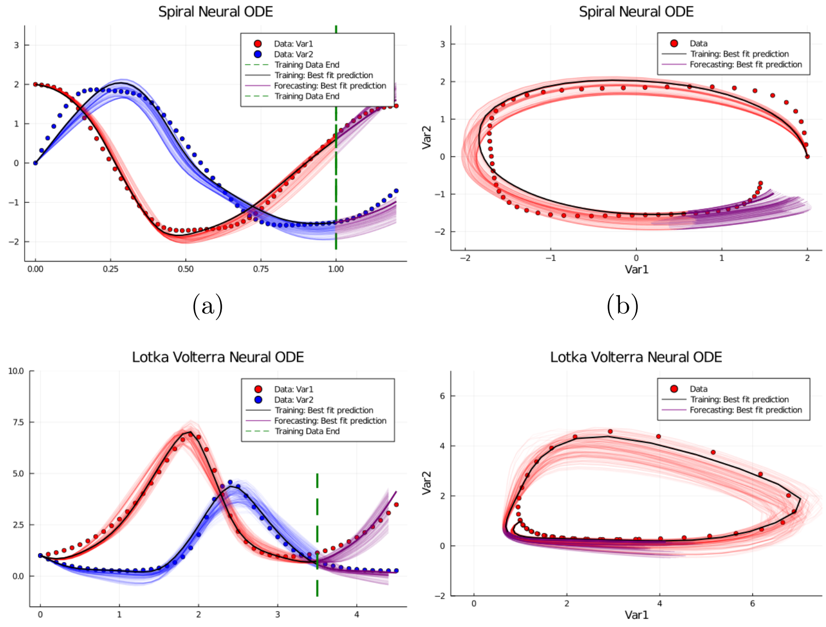
$$\Delta\theta_t := v \tag{6.12}$$

$$\Delta v_t := -\eta \nabla \tilde{U}(x) - \alpha v + \mathcal{N}(0, 2(\alpha - \hat{\beta})\eta) \tag{6.13}$$

Naturally, η corresponds to the learning rate and α the momentum decay.

We apply the Bayesian Neural ODE framework outlined above to case studies 1 and 2 given in Equations 1-4. For case study 1, we use $\alpha = 1$, $\beta = 1$. In the SGHMC algorithm, we use $\eta = 1.5^{-6}$ and $\alpha = 0.07$ and draw 2500 posterior samples. We define a prior distribution centered at the MAP point with a standard deviation of 0.4. For case study 2, we use $\alpha = 1.5$, $\beta = 1.0$, $\gamma = 3.0$, and $\delta = 1.0$; we draw 350 samples with SGHMC using hyperparameters $\eta = 7.0^{-6}$ and $\alpha = 0.07$. We define a prior distribution centered at the MAP point with a standard deviation of 1.0. For both case studies, the neural ODE architecture consists of 2 layers with tanh activation function; there are 50 units in each layer for case study 1, and 10 units for case study 2.

Figure 6-3 illustrates the consistency of Bayesian Neural ODE SGHMC's prediction and forecasting with ground truth data, for both case studies (Equations 1-4).



(a)

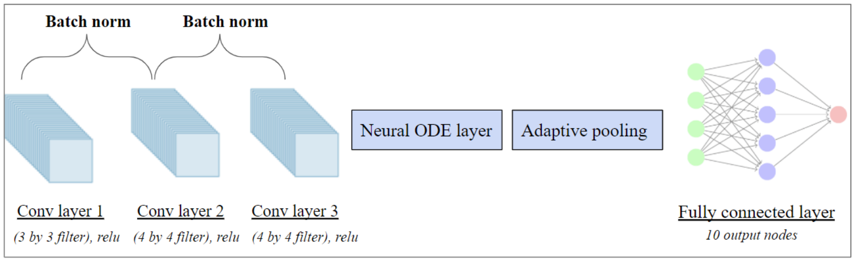
(b)

(c)

(d)

Figure 6-3: Comparison of Bayesian Neural ODE SGHMC's prediction and forecasting against ground truth data for (a,b) case study 1 and (c,d) case study 2.

SGHMC on the MNIST dataset



(a)

Figure 6-4: Neural network architecture used for the image classification task on MNIST. The Neural ODE contains two convolutional layers. The network has 208010 parameters in total. This architecture was combined with the SGHMC method to lead to a Bayesian Neural ODE object which can be used for image classification.

We now apply Bayesian Neural ODE with SGHMC to a image classification task on the MNIST dataset. A ResNet [61] layer behaves as a continuous time ODE at the limit of infinite layers. Given this natural analogy, we here implement ODE layers in place of the residual layers used in classic image recognition architectures.

Specifically, we design three convolutional layers interspersed with Batch Normal-

Table 6.2: Performance on MNIST using the Bayesian Neural ODE: SGHMC approach is outlined in the present study. Here, 310 posterior samples for each image in a test set of 10,000 images is considered. The best fit test error represents the mean of the number of erroneous predictions in all the posterior samples (310) for all images (10000).

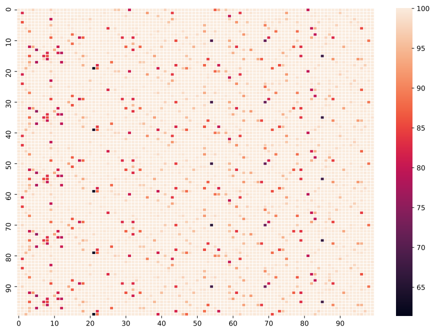
	Error estimates?	Neural ODE?	Best fit test error	Reference test error
RK-Net	No	Yes	0.47 %	Chen (2018)
ODE-Net	No	Yes	0.42 %	Chen (2018)
Bayesian Alex-Net	Yes	No	1 %	Shridhar (2019)
Bayesian LeNet-5	Yes	No	2 %	Shridhar (2019)
Bayesian Neural ODE (ensemble)	Yes	Yes	0.78 %	Our study

ization layers: the initial layer has a 3×3 filter and the next two convolutional layers each have a 4×4 filter (stride 2×2, padding 1×1, ReLU activations). A Neural ODE layer with two convolutional layers with 3×3 filter (padding 0, ReLU activations) is appended to act as a residual layer. Finally, we add an adapative average pooling layer and a fully connected layer consisting of 10 neurons (one per class). The full architecture is visualized in figure 6-4.

Given the Neural ODE architecture, we initialize SGHMC with decay schedule of $\epsilon_t = \eta . t^{-\gamma}$ and parameters $\gamma = 0.01$, $\eta = 0.5$ and $\alpha = 0.1$, execute for 2530 iterations, and sample the last 310 parameter updates. We find that tempering—scaling the standard deviation of the added noise in SGHMC by a constant [92]—significantly reduces time for convergence. Our experiment uses 10^4 as the tempering constant. SGHMC is more computationally expensive, requiring ten epochs through the training dataset, than a simple MAP estimation; a trial MAP optimization with ADAM yields a 98.7% test accuracy after a single-epoch run.

The test set consists of all 10,000 images in the MNIST dataset. Each cell in the heatmap of figure 6-5 represents the percentage of correct predictions out of 310 posterior samples on a single image. 91.8% of these cells have more than 99% confidence.

Table 6.2 shows the performance of our Bayesian approach on the MNIST data. Out of the 310 posterior samples for each image in the test set of 10,000 images considered, the best fit test error represents the mean of the number of erroneous predictions in all samples for all images. In our study, we have obtained a test



(a)

Figure 6-5: Bayesian Neural ODE with SGHMC is applied to the MNIST dataset. Each cell in this figure represents the percentage of correct predictions out of 310 posterior samples on a single image. Results for the entire test set of $10,000$ images is visualized here as a 100×100 heatmap

ensemble accuracy of 99.22 %, which is performance competitive with current state-of-the-art image classification methods.

From table 6.2, we note that previous architectures for MNIST analysis either have a Neural ODE architecture without error estimates [20] or do not incorporate a Neural ODE for error estimation [150]. Incorporated Neural ODEs in our approach, we not only demonstrate a classification performance competitive with current state-of-the-art image classification methods; but also quantify the confidence of our prediction.

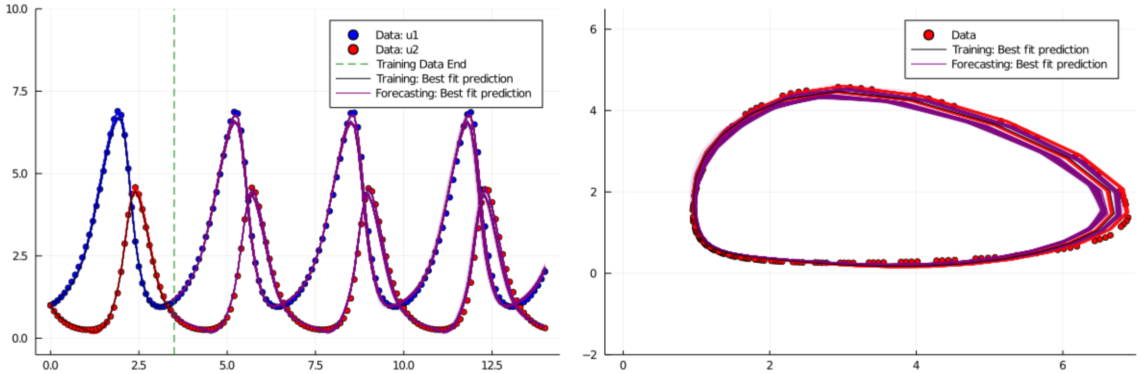
6.2.3 Bayesian Neural ODE: SGLD

Stochastic Gradient Langevin Dynamics (SGLD) is an adaptation, designed to sample from the posterior as the iterations increase, of the usual stochastic gradient descent algorithm. In each iteration, we update our vector θ of parameters according to the rule

$$\theta_t := \theta_t - \Delta\theta_t \tag{6.14}$$

$$\Delta\theta_t := \frac{\epsilon_t}{2}\left(\nabla \log p(\theta_t) + \frac{1}{n}\sum_{i=1}^{n} \nabla \log p(\mathcal{D}_n|\theta)\right) + \eta_t \tag{6.15}$$

$$\eta_t \sim \mathcal{N}(0, \epsilon_t) \tag{6.16}$$



(a) (b)

Figure 6-6: Comparison of the Bayesian Neural ODE: SGLD estimation and data for the Lotka Volterra ODE case study shown as (a) time series plots and (b) contour plots.

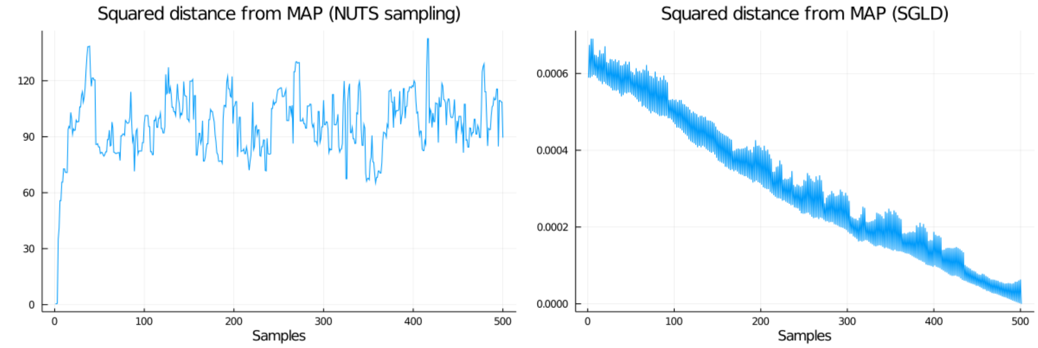
where $\mathcal{D}_n$ are the minibatches the training dataset $\mathcal{D}$ has been split into. $p(\mathcal{D}_n|\theta)$ is the likelihood, whose logarithm is equivalent to the loss function, and $p(\theta_t)$ is any priors, also known as regularisation terms, imposed onto the parameters θ. The stepsizes ϵ_t must follow a decaying scheme which satisfies the conditions [164]:

$$\sum_{t=1}^{\infty}\epsilon_t = \infty \qquad \sum_{t=1}^{\infty}\epsilon_t^2 < \infty \tag{6.17}$$

in this article we have chose a polynomial decaying scheme $\epsilon_t = a(b+t)^{-\gamma}$ with a, b, and γ as tuneable hyperparameters.

The update scheme for θ is composed of two stages. In the first stage, where the approximate gradient dominates, we approach the regions with higher mass probability. During the second phase, instead of allowing θ to converge to a single value, it walks randomly with a predominantly Gaussian noise since the gradient is $\mathcal{O}(\epsilon_t)$ and the Gaussian noise is $\mathcal{O}(\sqrt{\epsilon_t})$. It is on this second stage where it is theoretically guaranteed to converge to the posterior, and hence, we may use this stage to sample parameters from the posterior.

We now apply SGLD on the Lotka Volterra system in Equations 3-4. The Lotka Volterra system used in this case has the same parameters as the one used for NUTS. Again, we apply SGLD with 45000 iterations and sampled the last 2000 updates. The hyperparameters used were $a = 0.0025, b = 0.05, \gamma = 0.35$. The neural ODE



(a) (b)

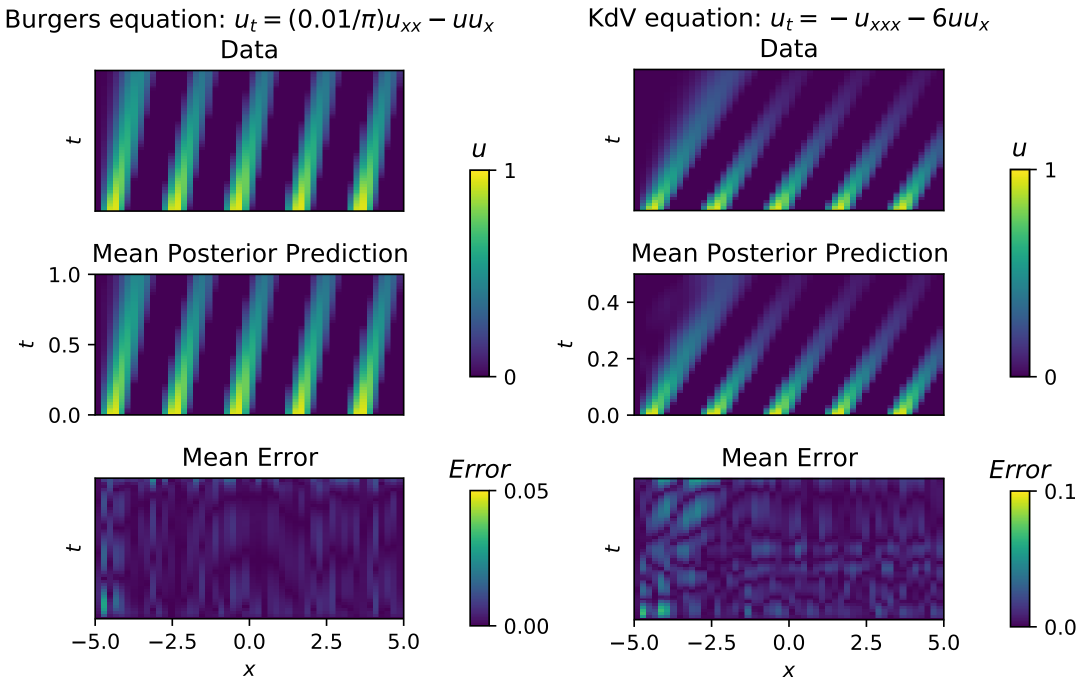
Figure 6-7: (a) Figure shows that for the NUTS sample initialized at the MAP point, the sampler quickly jumps away from the MAP point and never returns back. (b) Figure shows that for the SGLD sampler initialized at the MAP, all posteriors samplers are close to the MAP point.

architecture was again 2 layers with 50 neurons and tanh activation. The algorithm took approximately 679 seconds to run.

From figure 6-6 we notice a good fit on the training dataset. The Bayesian Neural ODE trained using the SGLD approach has accurately captured the periodicity of the system; and is seen to generalize for a much longer duration than the NUTS sampler.

Comparison between SGLD and NUTS

Through figures 6-1 and 6-6, we note that SGLD generally has a better mean prediction accuracy than NUTS. This can be attributed to the non-convexity/multi-modality of the likelihood function where the MAP point is likely to be in a region with low probability mass, leading the NUTS sampler which uses non-stochastic gradients to not "find" the region surrounding the MAP point in the time of the sampling. The use of stochastic gradients in SGLD seems to have led to a sample much closer to the MAP point and with a much lower mean prediction error. This difference between NUTS and SGLD is illustrated in figure 6-7, where the distance of the posterior samples from the MAP point is shown. Figure 6-7a shows that for the NUTS sample initialized at the MAP point, the sampler quickly jumps away from the MAP point and never returns back. Figure 6-7b shows that for the SGLD sampler, all posteriors samplers are much closer to the MAP point, than the NUTS sampler.



(a) (b)

Figure 6-8: Figure shows the comparison between the mean posterior prediction for the trained Neural PDE with the true data, and also the resulting error for (a) Example 1: Burgers' equation and (b) Example 2: KdV equation

Application to Neural PDE's

The SGLD methodology can be readily extended for uncertainty quantification analysis of a Neural PDE. The input vector which varies in space at the initial time $(u(x,0))$ can be propagated through a neural network architecture NNODE with weights p, and reformulated as a ODE according to

$$\frac{\partial u(x,t)}{\partial t} = \text{NNODE}(u(x,0),p) \tag{6.18}$$

The solution of this ODE, $\hat{u}$ will be an $n \times m$ vector where n is the number of time points discretization and m is the number of space points discretization. Using an L2 loss function based on the difference between $\hat{u}$ and the true data $\bar{u}$, the SGLD methodology outlined in section 2.3 can be applied in a similar manner to provide the posterior of the neural network parameters p, and thus enable uncertainty quantification in the PDE solution.

We illustrate the applicability of SGLD to PDEs using the examples below

Example 1: Burgers' equation

The Burgers' equation is a PDE which shows in wide range of fields including fluid dynamics, gas dynamics and acoustics and is given by

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \frac{0.01}{\pi}\frac{\mathrm{d}^2u}{\mathrm{d}x^2} - u\frac{\mathrm{d}u}{\mathrm{d}x} \tag{6.19}$$

Example 2: Korteweg–De Vries (KdV) equation

The KdV equation is a PDE which shows up in the field of shallow water analysis and is given by

$$\frac{\mathrm{d}u}{\mathrm{d}t} = -\frac{\mathrm{d}^3u}{\mathrm{d}x^3} - 6u\frac{\mathrm{d}u}{\mathrm{d}x} \tag{6.20}$$

To generate the true data for these examples, the $x \times t$ space was divided into an $n \times n$ grid with $n = 51$ for both examples. The PDE discretization was performed using the method of lines and then solved using an adapative ODE solver in Julia.

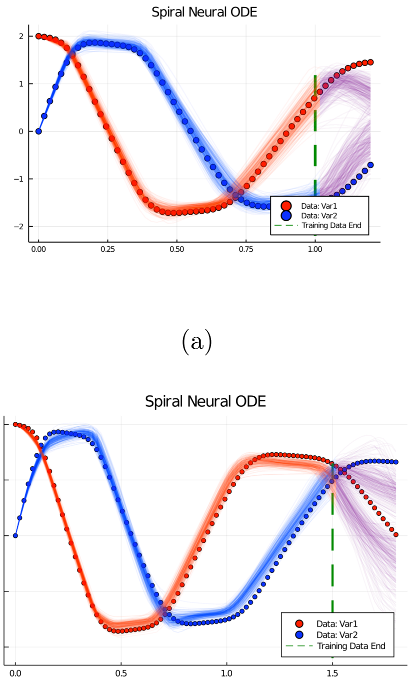
The input vector to the neural PDE architectures in both cases was of size $n \times 1 = 51 \times 1$. We used 2 layers with 10 neurons in each layer and the relu activation function. Propagation of the input vector to this neural network led to an output of size $n \times n = 51 \times 51$.

We ran the SGLD algorithm with 40000 iterations and sampled the last 600 updates. The hyperparameters used were $a = 0.001, b = 0.15, \gamma = 0.05$ for both examples.

Since the SGLD method leads to sampling from the true parameter posterior, we can compare the mean posterior prediction for the trained Neural PDE with the true data and also the resulting error. Figures 6-8a, b shows this comparison and a reasonable agreement is seen. Through this method, we also get estimates of the error in the mean posterior prediction, and not just a deterministic prediction.

The extension of the SGLD method to PDE's further strengthens its validity as a useful Bayesian Neural ODE/PDE method.

6.2.4 Bayesian Neural ODE: Variational Inference



(a)

(b)

Figure 6-9: For the Spiral ODE example (Equations 1-2), figure shows the retrodiction plots for (a) Variational Inference framework used in the present study and (b) Variational Inference integrated with Normalizing Flow. We can see that integration with normalizing flows used shows marginal improvement over plain Variational Inference with mean field approximation.

Since the last decade, there has been explosive interest in applying variational inference (VI) to text analysis, generative image modeling, and physical/chemical systems analysis [140]. VI, an optimization-based approach, generally approximates the posterior much faster than traditional MCMC methods. Variational inference was traditionally used for learning in graphical models (e.g. the sigmoid belief network) [80, 148]. However, there have been significant expansions in the methodology and application domains of VI; examples include stochastic variational inference [66] on large-scale data analysis and black-box variational inference's [134] usage beyond conditionally conjugate models. Through the use of normalizing flows [140], implicit distributions [103, 157] and importance-weighted variational autoencoders [25], the posterior-approximating variational inference family was made more expressive and thus powerful.

Here, we aim to address whether variational inference methods can be integrated into, and thus further expand, our Bayesian Neural ODEs framework. We use the Turing.jl interface in Julia [42]. Initially, we define a multivariate normal distribution as the posterior family approximating the true posterior with a mean-field approximation. For maximizing the expected lower bound, we use ADVI (Automatic Differentiation Variational Inference) [88], with 10 samples per step and 5000 as the upper bound on the number of gradient steps. Figure 6-9a shows the poor forecasting performance of the Bayesian Neural ODE: Variational Inference framework applied to the Spiral ODE example, compared to HMC methods explored above (figures 6-1, 6-3).

Integration of Normalizing Flows

One possible reason for the poor forecasting performance of the VI + Neural ODE framework could be due to the posterior family not being expressive enough. To further explore the effects of a powerful posterior family, we look at normalizing flows.

Through a chain of invertible mappings, normalizing flows transform an initially simple posterior family distribution (such as the multivariate normal in the above example) into an arbitrary complex distribution [140]. Considering an initial random variable z with a distribution prescribed by $q(z)$. A bijective mapping function f with inverse g, when applied to z results in a new variable z' with distribution given by

$$q'(z') = q(z)|\det\frac{\partial g}{\partial z'}| = q(z)|\det\frac{\partial f}{\partial z}|^{-1} \tag{6.21}$$

In their original work, [140] demonstrated two types of normalizing flows: planar layer and radial layer. In this study, we have employed the planar layer. The planar layer, parametrized by u, w, b is given by the invertible function

$$f(z) = z + uh(w^T z + b) \tag{6.22}$$

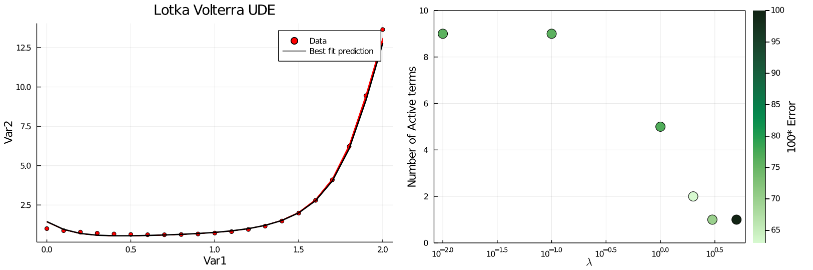
where h is a differentiable element-wise non-linearity. Thus, applying a sequence of maps f_k to an initial density leads to the transformed variable and corresponding density as

$$\begin{aligned} z_k &= f_k \circ f_{k-1} \circ \ldots \circ f_1(z) \\ \ln q_k(z_k) &= \ln q_0(z) - \sum_{k=1}^{K} \ln|1 + u_k^T \psi_k z_{k-1}| \end{aligned} \tag{6.23}$$

where $\psi(z) = h'(w^T z + b)w$. Application of a planar flow to a standard Gaussian distribution leads to flexible expansions/contractions along hyperplanes and thus makes the base distribution a lot more expressive.

In the present study, the base distribution was initialized as a multivariate normal distribution with mean as a randomly initialized vector with length governed by the number of parameters of the neural ODE. We transformed this base distribution using a composition of 2 such planar layers described in Equation (6.22); with z being the parameters of the neural ODE.

Figure 6-9b shows that through the inclusion of normalizing flows, the estimation performance of the Bayesian Neural ODE: Variational Inference object is marginally better compared to figure 6-9a. For the Variational Inference experiment we used During experiments with different configurations such as Neural Network size, time span of solution and initial weights for the Neural Network used as well for the Normalizing Flow layers had considerable effect on the training performance and forecasting capability of the model. There are quite a lot of open questions that need to be addressed to ensure suitability of Variational Inference with Neural ODEs and will be part of future work.



(a) (b)

Figure 6-10: Bayesian Neural UDE estimation is demonstrated for the Lotka Volterra example with a missing term as shown in (6.24); using the PSGLD approach. (a) Comparison of the recovered missing term and the actual term and (b) Sparsity plot using the STRRidge algorithm. The highlighted box shows the optimal point which gives the sparsest solution (1 term) with a low error. This plot is seen to be the same for 100 trajectories considered in the sampling phase.

6.2.5 Bayesian Neural UDE: SGLD and PSGLD

In this section, we aim to demonstrate a viable method for the probabilistic quantification of epistemic uncertainties via a hybrid machine-learning and mechanistic-model-based technique.

More efficient training of deep neural networks is achieved by using a preconditioned matrix $G(\theta)$ in the gradient update step of Equation 15 of the SGLD method [92]. Combining this with an adaptive step size method like RMSprop leads to much faster sampling than the standard SGLD approach, as outlined in the Preconditioned SGLD with RMSprop algorithm by [92]. We demonstrate the use of this algorithm in this section, where standard SGLD failed to converge to the true posterior.

Application to a predator-prey model

As outlined by [129], universal differential equations (UDE's) can be used to recover missing terms of governing equations describing dynamical systems. As an example, we look at the Lotka Volterra system with a missing term denoted by $M(u_1, u_2)$ in the first variable derivative as

$$\frac{du_1}{dt} = -\alpha u_1 - M(u_1, u_2) \tag{6.24}$$

$$\frac{du_2}{dt} = -\delta u_2 + \gamma u_1 u_2 \tag{6.25}$$

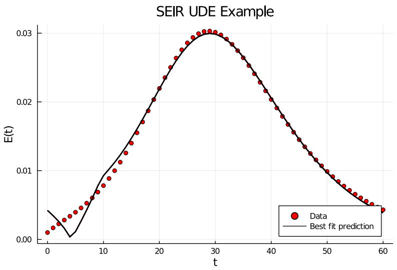
$$\tag{6.26}$$

Table 6.3: Bayesian Neural UDE: Recovery of dominant terms for the Lotka Volterra example, as the sparsity parameter λ is varied. Highlighted row shows the sparsity parameter with the lowest positive AIC score.

λ	Number of Active terms	Dominant terms	Error	Mean AIC score	% sampled
0.01	9	u_1^2, u_2^2, u_1u_2 $u_1^2u_2^2, u_1^2u_2, u_2^2u_1$ u_1u_2, const	0.765	40.4	100
0.1	9	u_1^2, u_2^2, u_1u_2 $u_1^2u_2^2, u_1^2u_2, u_2^2u_1$ u_1u_2, const	0.764	35	100
1	5	u_1^2, u_2^2, u_2 $u_1^2u_2, u_1u_2$	0.764	21.6	100
2	2	$u_1^2u_2, u_1u_2$	0.634	7.2	100
3	**1**	u_1u_2	**0.7**	**4.1**	**100**
5	1	$u_1^2u_2$	2.49	-1	100

Using the PSGLD method outlined in [129], we trained $M_\theta(u_1, u_2)$ as a neural network to optimize the weights θ; and recover the missing term time series. We sampled from the last 100 updates of the converged sampler. Figure 6-10a shows that the recovered time series $M_\theta(u_1, u_2)$ from 100 trajectories matches very well with the actual term $M_\theta(u_1, u_2) = u_1u_2$. These optimized parameter space for all 100 trajectories lies very close to each other, indicating that the PSGLD method indeed converges and then subsequently samples closer to the true posterior.

Subsequently, we used a sparse regression technique called Sequential Thresholded Ridge Regression (STRRidge) algorithm [14] on the neural network output to reconstruct the missing dynamical equations for 100 trajectories of the sampled parameter space. The STRRidge algorithm has a tunable sparsity parameter λ to control the sparsity of the obtained dominant terms. Optimally, we would want the sparsest



(a)

Figure 6-11: Bayesian Neural UDE estimation is demonstrated for the SEIR example with a missing term as shown in (6.32); using the PSGLD approach. (a) Comparison of the recovered missing term and the actual term shown for 100 trajectories considered in the sampling phase.

solution with the least possible error.

Figure 6-10b shows the variation of the number of terms recovered by the STR-Ridge algorithm with the sparsity parameter, λ; which shows a decreasing trend as expected. The colorbar indicates the error between the sparse recovered solution and the neural network output $M_\theta(u_1, u_2)$. It can be seen that the highlighted box indicates the optimal point which has the sparsest solution (1 term) with a very low error. This point also corresponds to the lowest positive AIC score (Akaike Information Criteria) [13, 161] which strives to minimize the model error as well as its complexity (shown in Table 6.3). Along with showing the AIC score as function of the sparsity parameter λ, table 6.3 also shows the dominant terms as the sparsity parameter λ is varied.

This optimal solution has the quadratic form $\sim u_1 u_2$, for all 100 trajectories indicating that Bayesian Neural UDE approach recovers the correct solution for all sampled trajectories. Out of the 100 models sampled for the optimal sparsity parameter, the model with the lowest AIC score was found to be $M(u_1, u_2) = 0.96 u_1 u_2$, which is very close to the true solution $M(u_1, u_2) = u_1 u_2$.

Application to an epidemiology model

As another example to test our approach, we consider a SEIR epidemiological model. The SEIR is a compartment based model which models transfer of population between four compartments: Susceptible, Exposed, Infected and Recovered using the following set of equations:

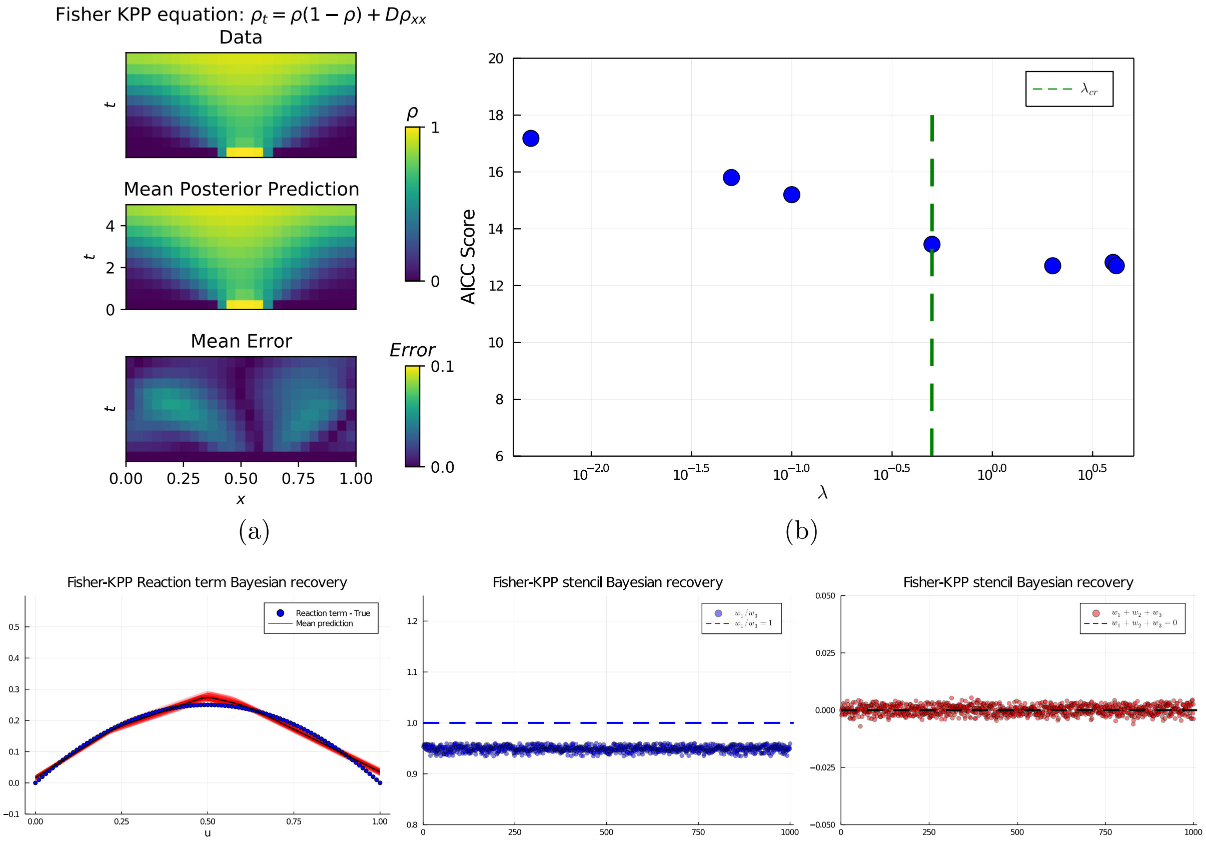
$$\frac{dS}{dt} = -\beta SI \tag{6.27}$$
$$\frac{dE}{dt} = \beta SI - \sigma E \tag{6.28}$$
$$\frac{dI}{dt} = \sigma E - \gamma I \tag{6.29}$$
$$\frac{dR}{dt} = \gamma I \tag{6.30}$$
$$\tag{6.31}$$

In Equation 6.27, we will try to infer the exposure term using Bayesian Neural UDE: PSGLD approach. Thus, we will try to learn $M(E, I) = \sigma E$ in the following system of equations

$$\frac{dS}{dt} = -\beta SI \tag{6.32}$$
$$\frac{dE}{dt} = \beta SI - M(E, I) \tag{6.33}$$
$$\frac{dI}{dt} = \sigma E - \gamma I \tag{6.34}$$
$$\frac{dR}{dt} = \gamma I \tag{6.35}$$
$$\tag{6.36}$$

Using the PSGLD method outlined in [129], we trained $M_\theta(E, I)$ as a neural network to optimize the weights θ; and recover the missing term time series. We sampled from the last 100 updates of the converged sampler. Figure 6-11a shows that the recovered time series $M_\theta(E, I)$ from 100 trajectories matches very well with the actual term $M_\theta(E, I) = \sigma E$.

Subsequently, we applied the STRRidge algorithm to recover the symbolic equations for the missing terms, with the sparsity parameter λ ranging from $0.005 - 0.5$. The model for which the lowest AIC score was obtained, was found to contain just



(a) (b)

(c) (d) (e)

Figure 6-12: Bayesian Neural UDE estimation is demonstrated for the Fisher-KPP PDE example with a missing term as shown in (6.38); using the PSGLD approach. Figures show comparison between: (a) Training data and the mean of 500 recovered posterior solutions, (c) True reaction term and the posterior recovered term, (d, e) Posterior recovered weights for the convolutional filter and the canonical stencil $[1, -2, 1]$ for the one-dimensional Laplacian. (b) The variation of the sparsity parameter λ in the STRRidge algorithm, with the obtained AIC score

one dominant term (E) and with the symbolic form $M(E, I) = 0.099E$ compared to the ground truth data of $M(E, I) = 0.1E$, for all 100 trajectories sampled.

6.2.6 Application to PDE's: Wave propagation

To illustrate the probabilistic system identification using Bayesian Neural UDEs, we consider a spatio-temporal system governed by the one-dimensional Fisher-KPP PDE

$$\frac{\partial \rho}{\partial t} = r\rho(1-\rho) + D\frac{\partial^2 \rho}{\partial x^2}, \tag{6.37}$$

Table 6.4: Bayesian Neural UDE: Recovery of the quadratic reaction term for the Fisher-KPP equation. Results are shown for the sparsity parameter $\lambda = \lambda_{cr} = 0.5$ for which the AIC score begins to show a plateau (figure 6-12b). Results are shown for 1000 posterior samples

λ_{cr}	Number of	Dominant terms	% of
	Active terms		samples
0.5	2	ρ, ρ^2	73
0.5	3	ρ, ρ^2, ρ^3	27

with $x \in [0,1]$, $t \in [0,T]$, and periodic boundary condition $\rho(0,t) = \rho(1,t)$. Here ρ represents population density of a species, r is the local growth rate and D is the diffusion coefficient. Such reaction-diffusion equations appear in diverse physical, chemical and biological problems [56]. To learn the generated data, we define the UPDE:

$$\rho_t = \mathrm{NN}_\theta(\rho) + \hat{D}\,\mathrm{CNN}(\rho), \tag{6.38}$$

where NN_θ is a neural network representing the local growth term. The derivative operator is approximated as a convolutional neural network CNN, a learnable arbitrary representation of a stencil while treating the coefficient $\hat{D}$ as an unknown parameter fit simultaneously with the neural network weights. We encode in the loss function extra constraints to ensure the learned equation is physically realizable, i.e. the derivative stencil must be conservative (the coefficients sum to zero).

We trained the neural network NN_θ and the convolutional neural network NN_θ using the PSGLD method outlined above. We sample from the last 1000 updates of the converged posterior. Figure 6-12a shows that the mean of the posterior recovered solutions shows a good match with the training data. Figure 6-12c shows the bayesian recovery of the reaction term, parameterized by NN_θ, which shows a quadratic form for all 1000 posterior samples with mean value lying close to the true quadratic reaction term. Figure 6-12d, e shows the posterior recovered convolutional filter $[w_1, w_2, w_3]$. From these two plots, we see that the posterior for these weights lies very close to the canonical stencil $[1, -2, 1]$ for the one-dimensional Laplacian.

Subsequently, we applied the STRRidge algorithm to recover the symbolic equations for the reaction term $r\rho(1-\rho)$, with the sparsity parameter λ ranging from $O(1e-3)-O(1)$. The variation of λ with the obtained AIC score is shown in figure 6-12b. We see the critical λ for which the AIC score starts showing a plateau is marked as $\lambda_{cr} = 0.5$ and shown as a dotted line in figure 6-12b. We used the STRRidge algorithm using this value of the sparsity parameter λ for 1000 recovered posterior samples. In table 6.4, we see that 73% of these posterior samples show the dominant symbolic terms to be ρ, ρ_2, which matches the true quadratic reaction term form. For the remaining 27%, the terms recovered are seen to be ρ, ρ^2, ρ^3.

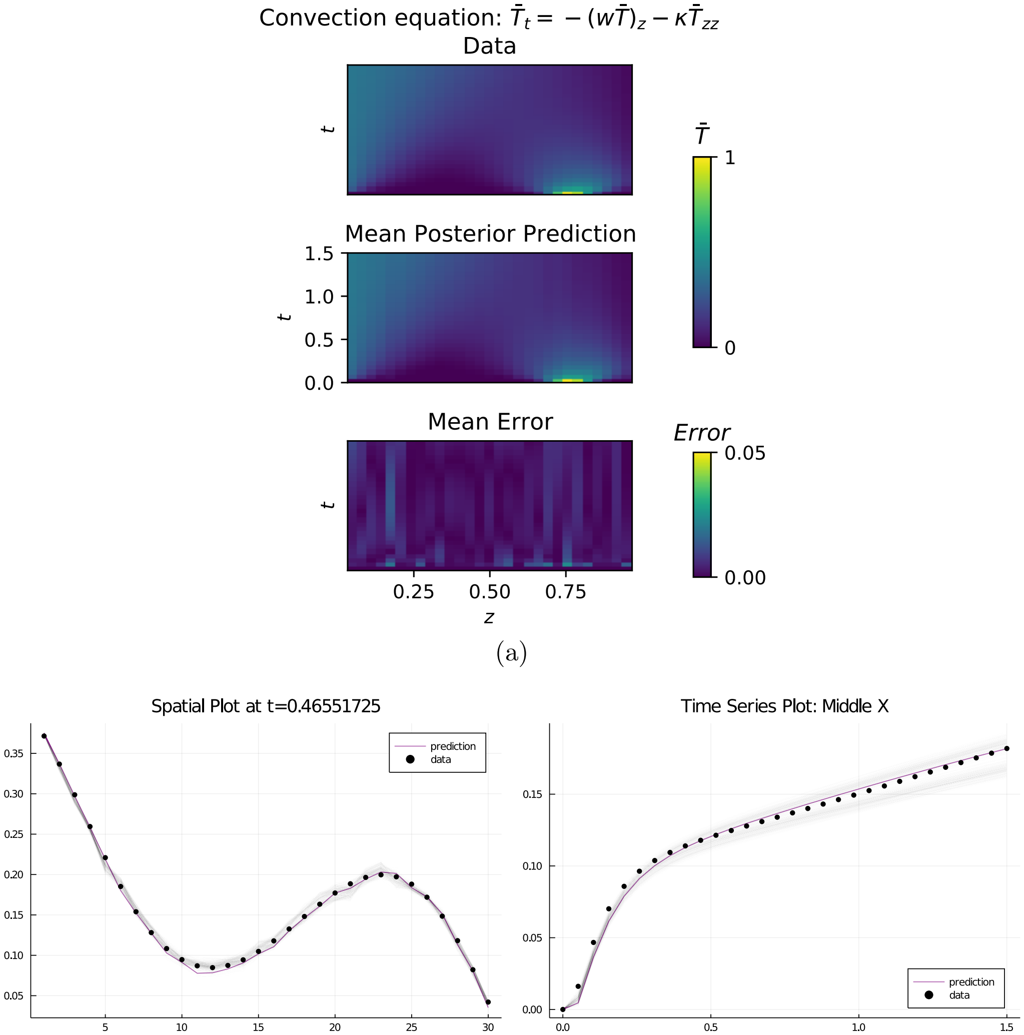
6.2.7 Application to PDE's: Climate models

As an example of directly accelerating existing scientific workflows, we focus on the Boussinesq equations [26]. The Boussinesq equations are a system of 3+1-dimensional partial differential equations acquired through simplifying assumptions on the incompressible Navier-Stokes equations, represented by the system:

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0, \\ \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} &= -\nabla p + \nu \nabla^2 \mathbf{u} + b\hat{z}, \\ \frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T &= \kappa \nabla^2 T, \end{aligned} \tag{6.39}$$

where $\mathbf{u} = (u, v, w)$ is the fluid velocity, p is the kinematic pressure, ν is the kinematic viscosity, κ is the thermal diffusivity, T is the temperature, and b is the fluid buoyancy. We assume that density and temperature are related by a linear equation of state so that the buoyancy b is only a function $b = \alpha g T$ where α is the thermal expansion coefficient and g is the acceleration due to gravity.

This system is commonly used in climate modeling, especially for modeling the ocean [54, 26] in a multi-scale model that approximates these equations by averaging out the horizontal dynamics $\overline{T}(z,t) = \iint T(x,y,z,t)\,dx\,dy$ in individual boxes. The resulting approximation is a local advection-diffusion equation describing the evolution of the horizontally-averaged temperature $\overline{T}$:



(a)

(b)

(c)

Figure 6-13: Bayesian Neural UDE estimation using the PSGLD approach. is demonstrated for the on Eddy Model parametrizations used in Climate models. Figures show comparison between: (a) Training data and the mean of 500 recovered posterior solutions, (b) Temporal slice of the training data and the recovered posterior and (c) Spatial slice of the training data and the recovered posterior.

$$\frac{\partial \overline{T}}{\partial t} + \frac{\partial \overline{wT}}{\partial z} = \kappa \frac{\partial^2 \overline{T}}{\partial z^2}. \tag{6.40}$$

This one-dimensional approximating system is not closed since $\overline{wT}$ is unknown. Common practice closes the system by manually determining an approximating $\overline{wT}$ from ad-hoc models, physical reasoning, and scaling laws. However, we can utilize a UDE-automated approach to learn such an approximation from data. Let

$$\overline{wT} = U_\theta \left(P, \overline{T}, \frac{\partial \overline{T}}{\partial z} \right) \tag{6.41}$$

where P are the physical parameters of the Boussinesq equation at different regimes of the ocean, such as the amount of surface heating or the strength of the surface winds [55]. We can accurately capture the non-locality of the convection in this term by making the UDE a high-dimensional neural network.

Data was generated from the diffusion-advection equations using the missing function $\overline{wT} = \cos(\sin(T^3)) + \sin(\cos(T^2))$. Similar to the above examples, we can train this neural network for the $\overline{wT}$ term using the PSGLD approach and obtain bayesian estimates of the recovered solution using the weight posteriors of this neural network.

Figure 6-13a shows that the mean of the posterior recovered solutions shows a good match with the training data. Figure 6-13b, c shows the temporal and spatial slices of the training data respectively, compared with the bayesian recovered solutions for 500 samples. We can see that we can not only recover the correct spatial and temporal variations in the temperature data but also capture the uncertainty associated with the predictions.

Thus, the model discovery for both the predator-prey example and the epidemiological model is robust to uncertainty. The Bayesian Neural UDE framework is thus an added arsenal to the recently demonstrated methods on bayesian system identification [6, 170]. Future work will further identify the relationship between model uncertainties and probabilistic automated discovery.

6.3 Conclusion and Future Work

We have shown that Bayesian learning frameworks can be integrated with Neural ODE's to quantify the uncertainty in the weights of a Neural ODE, using three sampling methods: NUTS, SGHMC and SGLD. Bayesian Neural ODEs with SGLD sampling is seen to provide better prediction accuracy and less bias from the MAP than NUTS sampling, possibly due to non-convexity/multi-modality of the likelihood function where the MAP point is likely to be in a region with low probability mass. However, a better understanding of why the two algorithms differ and how they compare to other algorithms is needed.

In addition, using a novel architecture which integrates convolution layers and Neural ODEs with the SGHMC framework; we demonstrate a test ensemble accuracy of 99.22% which is comparable with the accuracy of state-of-the-art image classification methods.

Subsequently, for the first time, we demonstrate the integration of Neural ODEs with Variational Inference. We show that when variational inference is combined with normalizing flows, it leads to a good prediction and estimation performance on physical systems; thus leading to a potentially powerful Bayesian Neural ODE object.

Finally, considering the problem of recovering missing terms from a dynamical system using universal differential equations (UDEs); we demonstrate the Bayesian recovery of missing terms from dynamical systems for (a) a predator-prey model and (b) an epidemiological model.

Currently, it is observed that Bayesian learning of Neural ODEs is computationally expensive for large datasets. Therefore, more work is needed to evaluate, understand and improve the convergence of various approximate Bayesian inference and MCMC algorithms in the context of Neural ODEs. Another research direction we plan to delve into further is Bayesian Neural SDE's and their applicability to physical systems and large scale machine learning datasets.

6.4 Code Availability

All codes for SGHMC, Bayesian Neural UDE are publicly available at `https://github.com/RajDandeka` and codes for the Variational Inference Neural ODE object is available at `https://github.com/mohamed` `BayesNeuralODE.jl`

6.5 Acknowledgements

This material is based upon work supported by the National Science Foundation under grant no. OAC-1835443, grant no. SII-2029670, grant no. ECCS-2029670, grant no. OAC-2103804, and grant no. PHY-2021825. We also gratefully acknowledge the U.S. Agency for International Development through Penn State for grant no. S002283-USAID. The information, data, or work presented herein was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0001211 and DE-AR0001222. We also gratefully acknowledge the U.S. Agency for International Development through Penn State for grant no. S002283-USAID. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. This material was supported by The Research Council of Norway and Equinor ASA through Research Council project "308817 - Digital wells for optimal production and drainage". Research was sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-

19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Chapter 7

Concluding remarks

Machine learning (ML) has brought about a revolution in the fields of image processing, robotics, self driving cars and security systems. Typically, these systems require a large amount of data to train the ML models on. Along with the requirements for large amounts of data, these models are typically non interpretable and physically intractable. Scientific Machine Learning (SciML) methodology enables training models using less amounts of data. These models are also highly interpretable in nature. They leverage the expressivity of neural networks with the interpretibilty of scientific structures like Ordinary Differential Equations (ODEs) and Partial Differential Equations (PDEs).

In this thesis, we have presented the application of these SciML frameworks to the field of epidemiology. The SciML models which we have developed, build on the fundamental epidemiological models and augment them with neural network modules. This augmentation not only makes these models expressive, but also highly interpretable in nature. We demonstrate that such models can be used for a wide range of applications including quarantine diagnosis, effect of early reopening in infection count evolution, virtual virus spread through Bluetooth tokens. These models are highly flexible in nature, enabling researchers and policy makers to adapt and extend these models to different applications. The appeal of these new models have led to collaborations with national labs, research universities and Silicon Valley startups who have used/ are using these models.

The development of this new methodology comes at a time when the world is grappling with the devastating effects of the Covid-19 pandemic. Since the outbreak of this pandemic, a number of modeling frameworks have been used to analyze various aspects of the pandemic such as prediction of infected and recovered case counts [31], hospitalizations [102], travel restrictions [23], reopening [32] and non-pharmaceutical interventions [38]. We hope that our proposed methodology can be another tool for researchers, policy makers and industries; as they learn new insights from this pandemic and as we all prepare for future waves of such a pandemic.

7.1 Future work

For the future work, there are several avenues which we wish to explore as follows:

- Design a graphical user interface (GUI) for the SciML models developed in this thesis.
- Look at implementing delay differential equations (DDEs) in association with the QSIR model to capture the delay seen in the diagnosis.
- Improve the Bayesian Neural ODE framework by introducing additional methods like Variational Inference (VI) with/ without normalizing flows.
- Extend the SciML modeling paradigm to Partial Differential Equations (PDEs) to capture population migration between cities etc.
- More interpretability such as bifurcation plots, system transition to chaos etc can be added as tools to the SciML modeling paradigm

Bibliography

[1] Linda JS Allen and Amy M Burgin. Comparison of deterministic and stochastic sis and sir models. *Department of Mathematics and Statistics Technical Report Series*, pages 98–003, 1998.

[2] Roy M Anderson, B Anderson, and Robert M May. Infectious diseases of humans: dynamics and control. Oxford university press, 1992.

[3] Look Andreas and Melih Kandemir. Differential bayesian neural nets. *arXiv preprint arXiv:1912.00796*, 2019.

[4] Apple. Mobility trend report. `https://www.apple.com/covid19/mobility`, 2020.

[5] Google Inc Apple Inc. Privacy-preserving contact tracing. `https://www.apple.com/covid19/contacttracing`, April 2020.

[6] Steven Atkinson. Bayesian hidden physics models: Uncertainty quantification for discovery of nonlinear partial differential operators from data. *arXiv preprint arXiv:2006.04228*, 2020.

[7] Department of Health Australian Government. CovidSafe. `https://www.health.gov.au/resources/apps-and-tools/covidsafe-app`, 2020.

[8] N. Baker, F. Alexander, T. Bremer, A. Hagberg, Y. Kevrekidis, H. Najm, M. Parashar, A. Patra, J. Sethian, S. Wild, et al. Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence. USDOE Washington, 2019.

[9] J. Bay, Joel Kek, Alvin Tan, Chai Sheng Hau, Lai Yongquan, Janice Tan, and Tang Anh Quy. BlueTrace: A privacy-preserving protocol for community-driven contact tracing across borders. `https://bluetrace.io/`, April 2020. Government Technology Agency, Singapore.

[10] Kim Bellware, Derek Hawkins, Hannah Knowles, Hannah Denham, Meryl Kornfield, Michael Brice-Saddler, Marisa Iati, and Joshua Partlow. Coronavirus death toll in u.s. increases as hospitals in hot spot states are overwhelmed. *The Washington Post*, 2020.

[11] Alex Berke, Michiel Bakker, Praneeth Vepakomma, Ramesh Raskar, Kent Larson, and AlexSandy' Pentland. Assessing disease exposure risk with location histories and protecting privacy: A cryptographic approach in response to a global pandemic. arXiv:2003.14412, April 2020.

[12] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.

[13] Hamparsum Bozdogan. Model selection and akaike's information criterion (aic): The general theory and its analytical extensions. *Psychometrika*, 52(3):345–370, 1987.

[14] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[15] Y. Cao, S. Li, L. Petzold, and R. Serban. Adjoint sensitivity analysis for differential-algebraic equations: The adjoint dae system and its numerical solution. SIAM journal on scientific computing 24, 1076-1089, 2003.

[16] B. Carey and J. Glanz. Hidden Outbreaks Spread Through U.S. Cities Far Earlier Than Americans Knew, Estimates Say. The New York Times. `https://www.nytimes.com/2020/04/23/us/coronavirus-early-outbreaks-cities.html`, 2020.

[17] CDC. Coronavirus Disease 2019 (COVID-19) Situation Summary, 3 March 2020, 2020.

[18] Jasper Fuk-Woo Chan, Shuofeng Yuan, Kin-Hang Kok, Kelvin Kai-Wang To, Hin Chu, Jin Yang, Fanfan Xing, Jieling Liu, Cyril Chik-Yan Yip, Rosana Wing-Shan Poon, et al. A familial cluster of pneumonia associated with the 2019 novel coronavirus indicating person-to-person transmission: a study of a family cluster. The Lancet 395, 514-523, 2020.

[19] Sheryl Chang, Nathan Harding, Cameron Zachreson, Oliver Cliff, and Mikhail Prokopenko. Modelling transmission and control of the COVID-19 pandemic in Australia. arXiv:2003.10218, April 2020.

[20] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.

[21] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691. PMLR, 2014.

[22] Wan Ni Chia, Feng Zhu, Sean Wei Xiang Ong, Barnaby Edward Young, Siew-Wai Fong, Nina Le Bert, Chee Wah Tan, Charles Tiu, Jinyan Zhang, Seow Yen

Tan, et al. Dynamics of sars-cov-2 neutralising antibody responses and duration of immunity: a longitudinal study, The Lancet Microbe, 2021.

[23] Matteo Chinazzi, Jessica T Davis, Marco Ajelli, Corrado Gioannini, Maria Litvinova, Stefano Merler, Ana Pastore y Piontti, Kunpeng Mu, Luca Rossi, Kaiyuan Sun, et al. The effect of travel restrictions on the spread of the 2019 novel coronavirus (covid-19) outbreak. Science, 2020.

[24] Hyunghoon Cho, Daphne Ippolito, and Yun William Yu. Contact tracing mobile apps for COVID-19: Privacy considerations and related trade-offs. arXiv:2003.11511, March 2020.

[25] Chris Cremer, Quaid Morris, and David Duvenaud. Reinterpreting importance-weighted autoencoders. *arXiv preprint arXiv:1704.02916*, 2017.

[26] Benoit Cushman-Roisin and Jean-Marie Beckers. Chapter 4 - equations governing geophysical flows. In Benoit Cushman-Roisin and Jean-Marie Beckers, editors, *Introduction to Geophysical Fluid Dynamics*, volume 101 of *International Geophysics*, pages 99 – 129. Academic Press, 2011.

[27] G Cybenko et al. Approximation by superposition of sigmoidal functions. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

[28] D. Cyranoski. What china's coronavirus response can teach the rest of the world. Nature, 2020.

[29] George E Dahl, Tara N Sainath, and Geoffrey E Hinton. Improving deep neural networks for LVCSR using rectified linear units and dropout. IEEE Acoustics, Speech and Signal Processing, 8609-8613, 2013.

[30] Daryl J Daley and Joe Gani. *Epidemic modelling: an introduction.* Number 15. Cambridge University Press, 2001.

[31] Raj Dandekar, Chris Rackauckas, and George Barbastathis. A machine learning-aided global diagnostic and comparative tool to assess effect of quarantine control in covid-19 spread. Patterns 1, 100145, 2020.

[32] Raj Dandekar, Emma Wang, George Barbastathis, and Chris Rackauckas. Implications of delayed reopening in controlling the covid-19 surge in southern and west-central usa. *Health Data Science*, 2021, 2021.

[33] Ensheng Dong, Hongru Du, and Lauren Gardner. An interactive web-based dashboard to track COVID-19 in real time. The Lancet infectious diseases 20, 533-534, 2020.

[34] DW. Coronavirus: What are the lockdown measures across europe?. DW. `https://www.dw.com/en/coronavirus-what-are-the-lockdown-measures-across-europe/a-52905137`, 2020.

[35] Alaa Elassar. This is where each state is during its phased reopening. *Cable News Network*, 2020.

[36] Marco Faggian, Michele Urbani, and Luca Zanotto. Proximity: a recipe to break the outbreak. arXiv:2003.10222, April 2020.

[37] H. Fang, J. Chen, and J. Hu. Modelling the sars epidemic by a lattice-based monte-carlo simulation. IEEE 27, 7470-7473, 2006.

[38] Neil Ferguson, Daniel Laydon, Gemma Nedjati Gilani, Natsuko Imai, Kylie Ainslie, Marc Baguelin, Sangeeta Bhatia, Adhiratha Boonyasiri, ZULMA Cucunuba Perez, Gina Cuomo-Dannenburg, et al. Report 9: Impact of non-pharmaceutical interventions (NPIs) to reduce COVID-19 mortality and healthcare demand. March 2020.

[39] Jonathan Fintzi, Xiang Cui, Jon Wakefield, and Vladimir N Minin. Efficient data augmentation for fitting stochastic epidemic models to prevalence data. *Journal of Computational and Graphical Statistics*, 26(4):918–929, 2017.

[40] Shannon Gallagher and JSM Baltimore. Comparing compartment and agent-based models. CMU, 2017.

[41] Lazaro Gamio. How coronavirus cases have risen since states reopened. *The New York Times*, 2020.

[42] Hong Ge, Kai Xu, and Zoubin Ghahramani. Turing: a language for flexible probabilistic inference. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, pages 1682–1690, 2018.

[43] J. Gershman. A Guide to State Coronavirus Reopenings and Lockdowns. The Wall Street Journal, May 21, 2020.

[44] E. Gibney. Whose coronavirus strategy worked best? Scientists hunt most effective policies. Nature News. https://www.nature.com/articles/d41586-020-01248-1, 2020.

[45] Daniel T Gillespie. Exact stochastic simulation of coupled chemical reactions, The journal of physical chemistry 81, 2340-2361, 1977.

[46] Daniel T Gillespie. The chemical langevin equation. The Journal of Chemical Physics 113, 297-306., 2000.

[47] Mark Girolami. Bayesian inference for differential equations. *Theoretical Computer Science*, 408(1):4 – 16, 2008. Computational Methods in Systems Biology.

[48] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. Proc. 14th International Conference on Artificial Intelligence and Statistics, 315-323, 2011.

[49] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning, 2016. (MIT press).

[50] Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. 30th int. conf. mach. learn., 1319-1327, 2013.

[51] S. Goodman. Sweden Has Become the World's Cautionary Tale. The New York Times. https://www.nytimes.com/2020/07/07/business/sweden-economy-coronavirus.html, 2020.

[52] Singapore Government Technology Agency. OpenTrace on GitHub: open source reference implementation of BlueTrace. https://github.com/OpenTrace-community, April 2020.

[53] Singapore Government Technology Agency. TraceTogether contact tracing app. https://www.tracetogether.gov.sg/, March 2020.

[54] Stephen M Griffies and Alistair J Adcroft. Formulating the equations of ocean models. 2008.

[55] Stephen M. Griffies, Michael Levy, Alistair J. Adcroft, Gokhan Danabasoglu, Robert W. Hallberg, Doug Jacobsen, William Large, , and Todd Ringler. Theory and Numerics of the Community Ocean Vertical Mixing (CVMix) Project. Technical report, 2015. Draft from March 9, 2015. 98 + v pages.

[56] Peter Grindrod. *The theory and applications of reaction-diffusion equations: patterns and waves*. Clarendon Press, 1996.

[57] Y. Gu. COVID-19 Projections Using Machine Learning. https://covid19-projections.com/, 2021.

[58] O. Guerin. Coronavirus: How Turkey took control of covid-19 emergency. BBC News. https://www.bbc.com/news/world-europe-52831017, 2020.

[59] C. Harlan and S. Pitrelli. Italy extends coronavirus lockdown to entire country, imposing restrictions on 60 million people. The Washington Post, 2020.

[60] Fiona P Havers, Carrie Reed, Travis Lim, Joel M Montgomery, John D Klena, Aron J Hall, Alicia M Fry, Deborah L Cannon, Cheng-Feng Chiang, Aridth Gibbons, et al. Seroprevalence of antibodies to sars-cov-2 in 10 sites in the United States, march 23-may 12, 2020, JAMA internal medicine 180, 1576 - 1586, 2020.

[61] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[62] T. Helm, E. Graham-Harrison, and R. Mckie. How did Britain get its coronavirus response so wrong? Guardian. `https://www.cnn.com/2020/05/29/americas/coronavirus-costa-rica-brazil-analysis-intl/index.html`, 2020.

[63] Herbert W Hethcote. The mathematics of infectious diseases. *SIAM review*, 42(4):599–653, 2000.

[64] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. neural computation 18, 1527-1554, 2006.

[65] Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. proceedings of the sixth annual conference on Computational learning theory, 5-13, 1993.

[66] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(5), 2013.

[67] Matthew D Hoffman and Andrew Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

[68] Michelle L Holshue, Chas DeBolt, Scott Lindquist, Kathy H Lofy, John Wiesman, Hollianne Bruce, Christopher Spitters, Keith Ericson, Sara Wilkerson, Ahmet Tural, et al. First case of 2019 novel coronavirus in the united states. New England Journal of Medicine, 2020.

[69] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[70] Hanwen Huang, Andreas Handel, and Xiao Song. A bayesian approach to estimate parameters of ordinary differential equation. *Computational Statistics*, 35(3):1481–1499, Sep 2020.

[71] Natsuko Imai, Anne Cori, Ilaria Dorigatti, Marc Baguelin, Christl A Donnelly, Steven Riley, and Neil M Ferguson. Report 3: transmissibility of 2019-nCov. Imperial College London, 2020.

[72] Michael Innes, Elliot Saba, Keno Fischer, Dhairya Gandhi, Marco Concetto Rudilosso, Neethu Mariya Joy, Tejan Karmali, Avik Pal, and Viral Shah. Fashionable modelling with flux. CoRR abs/1811.01457. `http://arxiv.org/abs/1811.01457`, 2018.

[73] International Monetary Fund. Policy Tracker: Policy responses to COVID-19. `https://www.imf.org/en/Topics/imf-and-covid19/Policy-Responses-to-COVID-19`, April 2020.

[74] ITO. ito: contact tracing app. `https://www.ito-app.org/`, 2020.

[75] Pavel Izmailov, Wesley J. Maddox, Polina Kirichenko, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Subspace inference for bayesian deep learning. *CoRR*, abs/1907.07504, 2019.

[76] John A Jacquez and Philip O'Neill. Reproduction numbers and thresholds in stochastic epidemic models i. homogeneous populations. *Mathematical Biosciences*, 107(2):161–186, 1991.

[77] Christopher I Jarvis, Kevin Van Zandvoort, Amy Gimma, Kiesha Prem, Petra Klepac, G James Rubin, and W John Edmunds. Quantifying the impact of physical distance measures on the transmission of COVID-19 in the UK. BMC Medicine 18, 1-10, 2020.

[78] Weiqi Ji and Sili Deng. Autonomous discovery of unknown reaction pathways from data by chemical reaction neural network. *The Journal of Physical Chemistry A*, 125(4):1082–1092, 2021.

[79] S. Jones. Spain orders nationwide lockdown to battle coronavirus. The Guardian, March 14, 2020.

[80] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[81] Laurent Valentin Jospin, Wray L. Buntine, F. Boussaid, Hamid Laga, and M. Bennamoun. Hands-on bayesian neural networks - a tutorial for deep learning users. *ArXiv*, abs/2007.06823, 2020.

[82] William Ogilvy Kermack and Anderson G McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, 115(772):700–721, 1927.

[83] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. `arXivpreprintarXiv:1412.6980`, 2014.

[84] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. `https://arxiv.org/abs/1312.6114`, 2013.

[85] Hayden Klok and Yoni Nazarathy. Statistics with Julia: Fundamentals for Data Science, Machine Learning and Artificial Intelligence, DRAFT. `https://people.smp.uq.edu.au/YoniNazarathy/julia-stats/StatisticsWithJulia.pdf`, 2020.

[86] H. Knowles, J. Wagner, H. Shaban, A. Taylor, K. Copeland, C. Buckner, M. Kornfield, and C. Itkowitz. Seven states report highest coronavirus hospitalizations since pandemic began. *The Washington Post*, 2020.

[87] Adam J Kucharski, Timothy W Russell, Charlie Diamond, Yang Liu, John Edmunds, Sebastian Funk, Rosalind M Eggo, Fiona Sun, Mark Jit, James D Munday, et al. Early dynamics of transmission and control of covid-19: a mathematical modelling study. The Lancet Infectious Diseases, 2020.

[88] Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M Blei. Automatic differentiation variational inference. *The Journal of Machine Learning Research*, 18(1):430–474, 2017.

[89] Hien Lau, Tanja Khosrawipour, Piotr Kocbach, Hirohito Ichii, Jacek Bania, and Veria Khosrawipour. Evaluating the massive underreporting and undertesting of covid-19 cases in multiple global epicenters, Pulmonology 27, 110 - 115, 2021.

[90] Marc Lavielle. Modelling some COVID-19 data. `http://webpopix.org/covidix19.html`, April 2020.

[91] Jasmine C Lee, Sarah Mervosh, Yuriria Avila, Barbara Harvey, and Alex Leeds Matthew. See how all 50 states are reopening (and closing again). *The New York Times*, 2020.

[92] Chunyuan Li, Changyou Chen, David Carlson, and Lawrence Carin. Preconditioned stochastic gradient langevin dynamics for deep neural networks. *arXiv preprint arXiv:1512.07666*, 2015.

[93] M. L. Li, H. T. Bouardi, O. S. Lami, T. A. Trikalinos, N. K. Trichakis, and D. Bertsimas. Forecasting covid-19 and analyzing the effect of government interventions. `https://doi.org/10.1101/2020.06.23.20138693`, 2020.

[94] Qun Li, Xuhua Guan, Peng Wu, Xiaoye Wang, Lei Zhou, Yeqing Tong, Ruiqi Ren, Kathy SM Leung, Eric HY Lau, Jessica Y Wong, et al. Early transmission dynamics in Wuhan, China, of novel coronavirus–infected pneumonia. New England Journal of Medicine, 2020.

[95] Ying Liu, Albert Gayle, Annelies Wilder-Smith, and Joacim Rocklöv. The reproductive number of COVID-19 is higher compared to SARS coronavirus. Journal of Travel Medicine, 2020.

[96] David J. Lunn, Nicky Best, Andrew Thomas, Jon Wakefield, and David Spiegelhalter. Bayesian analysis of population pk/pd models: General concepts and software. *Journal of Pharmacokinetics and Pharmacodynamics*, 29(3):271–307, Jun 2002.

[97] Wesley Maddox, Timur Garipov, Pavel Izmailov, Dmitry P. Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *CoRR*, abs/1902.02476, 2019.

[98] C. Maxouris. These states have some of the most drastic restrictions to combat the spread of coronavirus. CNN. `https://www.cnn.com/2020/03/17/us/states-measures-coronavirus-spread/index.html`, 2020.

[99] Donald A McQuarrie. Stochastic approach to chemical kinetics. Cambridge University Press 4, 413-478, 1967.

[100] Robinson Meyer and Alexis C. Madrigal. A devastating new stage of the pandemic. *The Atlantic*, 2020.

[101] Momchil Minkov, Ian AD Williamson, Lucio C Andreani, Dario Gerace, Beicheng Lou, Alex Y Song, Tyler W Hughes, and Shanhui Fan. Inverse design of photonic crystals through automatic differentiation. *Acs Photonics*, 7(7):1729–1741, 2020.

[102] Seyed M Moghadas, Affan Shoukat, Meagan C Fitzpatrick, Chad R Wells, Pratha Sah, Abhishek Pandey, Jeffrey D Sachs, Zheng Wang, Lauren A Meyers, Burton H Singer, et al. Projecting hospital utilization during the covid-19 outbreaks in the united states. *Proceedings of the National Academy of Sciences*, 117(16):9122–9126, 2020.

[103] Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.

[104] Robert Moss, J Wood, D Brown, Freya Shearer, Andrew Black, Allen Cheng, James McCaw, and Jodie McVernon. Modelling the impact of COVID-19 in Australia to inform transmission reducing measures and health system preparedness. `https://www.doherty.edu.au/news-events/news/covid-19-modelling-papers`, April 2020. Doherty Institute.

[105] B. Mukhopadhyay and R. Bhattacharyya. Analysis of a spatially extended nonlinear seis epidemic model with distinct incidence for exposed and infectives. Nonlinear Analysis: Real World Applications 9, 585-598, 2008.

[106] Government of India National Informatics Centre, Ministry of Electronics & Information Technology. Aarogya Setu: contact tracing app. `https://www.mygov.in/aarogya-setu-app/`, 2020.

[107] Faical Ndairou, Ivan Area, Juan J Nieto, and Delfim FM Torres. Mathematical modeling of covid-19 transmission dynamics with a case study of wuhan. Chaos, Solitons & Fractals, 2020.

[108] University of Toronto Neal, Radford M. Department of Computer Science. Learning stochastic feedforward networks, 1990.

[109] Zoltan Neufeld and Hamid Khataee. Targeted adaptive isolation strategy for COVID-19 pandemic. medrxiv. doi:10.1101/2020.03.23.20041897, 2020.

[110] Hiroshi Nishiura and Gerardo Chowell. The effective reproduction number as a prelude to statistical estimation of time-dependent epidemic trends. In *Mathematical and statistical estimation approaches in epidemiology*, pages 103–121. Springer, 2009.

[111] Jungsik Noh and Gaudenz Danuser. Estimation of the fraction of covid-19 infected people in Us states and countries worldwide, PLOS ONE, e0246772, 2021.

[112] D. Normille. Coronavirus cases have dropped sharply in south korea. what's the secret to its success?. science. https://www.sciencemag.org/news/2020/03/coronavirus-cases-have-dropped-sharply-south-korea-whats-secret-its-success, 2020.

[113] C. O'Dea. What Switzerland did right in the battle against coronavirus. MarketWatch, June 15, 2020.

[114] Polish Ministry of Digital Affairs. ProteGO app on GitHub. https://github.com/ProteGO-app, April 2020.

[115] Polish Ministry of Digital Affairs. ProteGO contact tracing app. https://www.gov.pl/web/cyfryzacja/zycie-po-kwarantannie--przetestuj-protego, April 2020.

[116] Czech Ministry of Health. eRouška app: Fighting against COVID-19 via privacy-first Bluetooth tracing. https://erouska.cz/, 2020.

[117] Czech Ministry of Health. eRouška app on GitHub. https://github.com/covid19cz, 2020.

[118] Israel's Ministry of Health. HaMagen app on GitHub. https://github.com/MohGovIL/hamagen-react-native, 2020.

[119] Israel's Ministry of Health. HaMagen: contact tracing app. https://govextra.gov.il/ministry-of-health/hamagen-app/download-en/, 2020.

[120] Ministry of Health of the Republic of North Macedonia. StopKorona!: tracing of Coronavirus exposure and protection from COVID-19. https://stop.koronavirus.gov.mk/en, April 2020.

[121] Travis E Oliphant. Guide to numpy, 2015. (CreateSpace Independent Publishing Platform).

[122] Nuria Oliver, Emmanuel Letouzé, Harald Sterly, Sébastien Delataille, Marco De Nadai, Bruno Lepri, Renaud Lambiotte, Richard Benjamins, Ciro Cattuto, Vittoria Colizza, et al. Mobile phone data and COVID-19: Missing an opportunity? arXiv:2003.12347, March 2020.

[123] Kiesha Prem, Yang Liu, Timothy W Russell, Adam J Kucharski, Rosalind M Eggo, Nicholas Davies, Stefan Flasche, Samuel Clifford, Carl AB Pearson, James D Munday, et al. The effect of control strategies to reduce social mixing on outcomes of the covid-19 epidemic in wuhan, china: a modelling study. The Lancet Public Health, 2020.

[124] David Price, Freya Shearer, Michael Meehan, Emma McBryde, Nick Golding, Jodie McVernon, and James McCaw. Estimating the case detection rate and temporal variation in transmission of COVID-19 in Australia. https://www.doherty.edu.au/news-events/news/covid-19-technical-modelling-report-and-press-conference, April 2020. Doherty Institute.

[125] Bluetooth SIG Proprietary. Bluetooth Core Specification version 4.2. https://www.bluetooth.com/specifications/bluetooth-core-specification/, December 2014.

[126] C Rackauckas and Q Nie. DifferentialEquations.jl - A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia. The Journal of Open Research Software 5(1), 2017.

[127] Christopher Rackauckas, Alan Edelman, Keno Fischer, Mike Innes, Elliot Saba, Viral B Shah, and Will Tebbutt. Generalized physics-informed learning through language-wide differentiable programming. In *AAAI Spring Symposium: MLPS*, 2020.

[128] Christopher Rackauckas, Mike Innes, Yingbo Ma, Jesse Bettencourt, Lyndon White, and Vaibhav Dixit. Diffeqflux.jl - A Julia Library for Neural Differential Equations. CoRR abs/1902.02376. http://arxiv.org/abs/1902.02376, 2019.

[129] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, and Ali Ramadhan. Universal Differential Equations for Scientific Machine Learning. arXiv:2001.04385, January 2020.

[130] Christopher Rackauckas and Qing Nie. Adaptive methods for stochastic differential equations via natural embeddings and rejection sampling with memory. Discrete and continuous dynamical systems, Series B 22, 2017.

[131] Aowabin Rahman, Ján Drgoňa, Aaron Tuor, and Jan Strube. Neural ordinary differential equations for nonlinear system identification. *arXiv preprint arXiv:2203.00120*, 2022.

[132] Hazhir Rahmandad, Tse Yang Lim, and John Sterman. Behavioral dynamics of covid-19: Estimating under-reporting, multiple waves, and adherence fatigue across 92 nations, System Dynamics Review 92, 2021.

[133] Ali Ramadhan, John Marshall, Andre Souza, Gregory LeClaire Wagner, Manvitha Ponnapati, and Christopher Rackauckas. Capturing missing physics in climate model parameterizations using neural differential equations. *arXiv preprint arXiv:2010.12559*, 2020.

[134] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822. PMLR, 2014.

[135] Rajesh Ranganath, Linpeng Tang, Laurent Charlin, and David Blei. Deep exponential families. Artificial Intelligence and Statistics, 762-771, 2015.

[136] Ramesh Raskar, Isabel Schunemann, Rachel Barbar, Kristen Vilcans, Jim Gray, Praneeth Vepakomma, Suraj Kapa, Andrea Nuzzo, Rajiv Gupta, Alex Berke, et al. Apps gone rogue: Maintaining personal privacy in an epidemic. arXiv:2003.08567, March 2020.

[137] Carl Edward Rasmussen. Gaussian processes in machine learning. Summer School on Machine Learning. 63-71, 2003.

[138] Jonathan M Read, Jessica RE Bridgen, Derek AT Cummings, Antonia Ho, and Chris P Jewell. Novel coronavirus 2019-ncov: early estimation of epidemiological parameters and epidemic predictions. , 2020.

[139] Report. Coronavirus in Latin America: What governments are doing to stop the spread. Global Americans. `https://theglobalamericans.org/2020/03/coronavirus-in-latin-america/`, 2020.

[140] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538. PMLR, 2015.

[141] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. `https://arxiv.org/abs/1401.4082`, 2014.

[142] S. Roberts. Lessons from the pandemic's superstar data scientist, Youyang Gu. MIT Technology Review. `https://www.technologyreview.com/2021/04/27/1023657/lessons-from-the-pandemics-superstar-data-scientist-youyang-gu/`, 2021.

[143] Weston C Roda, Marie B Varughese, Donglin Han, and Michael Y Li. Why is it difficult to accurately predict the covid-19 epidemic? Infectious Disease Modelling, 2020.

[144] R. Romo. Politics and poverty hinder Covid-19 response in Latin America. CNN, May 29. `https://www.cnn.com/2020/05/29/americas/coronavirus-costa-rica-brazil-analysis-intl/index.html`, 2020.

[145] Masaya M Saito, Seiya Imoto, Rui Yamaguchi, Hiroki Sato, Haruka Nakada, Masahiro Kami, Satoru Miyano, and Tomoyuki Higuchi. Extension and verification of the seir model on the 2009 influenza a (h1n1) pandemic in japan. Mathematical biosciences 246, 47-54, 2013.

[146] Karthik Abinav Sankararaman, Soham De, Zheng Xu, W Ronny Huang, and Tom Goldstein. The impact of neural network overparameterization on gradient confusion and stochastic gradient descent. In *International conference on machine learning*, pages 8469–8479. PMLR, 2020.

[147] Kankan Sarkar, Subhas Khajanchi, and Juan J Nieto. Modeling and forecasting the covid-19 pandemic in india. Chaos, Solitons & Fractals, 2020.

[148] Lawrence K Saul, Tommi Jaakkola, and Michael I Jordan. Mean field theory for sigmoid belief networks. *Journal of artificial intelligence research*, 4:61–76, 1996.

[149] Frank Schäfer, Michal Kloc, Christoph Bruder, and Niels Lörch. A differentiable programming method for quantum control. *Machine Learning: Science and Technology*, 1(3):035009, 2020.

[150] Kumar Shridhar, Felix Laumann, and Marcus Liwicki. A comprehensive guide to bayesian convolutional neural network with variational inference. *arXiv preprint arXiv:1901.02731*, 2019.

[151] A. Smirnova, L. deCamp, and G. Chowell. Forecasting epidemics through nonparametric estimation of time-dependent transmission rates using the seir model. Bulletin of mathematical biology 81, 4343-4365, 2019.

[152] Sho Sonoda and Noboru Murata. Neural network with unbounded activation functions is universal approximator. *Applied and Computational Harmonic Analysis*, 43(2):233–268, 2017.

[153] Rahul Subramanian, Qixin He, and Mercedes Pascual. Quantifying asymptomatic infection and transmission of covid-19 in new york city using observed cases, serology, and testing capacity, Proceedings of the National Academy of Sciences 118, 2021.

[154] Biao Tang, Xia Wang, Qian Li, Nicola Luigi Bragazzi, Sanyi Tang, Yanni Xiao, and Jianhong Wu. Estimation of the transmission risk of the 2019-nCov and its implication for public health interventions. Journal of Clinical Medicine 9, 462, 2020.

[155] IISc-TIFR Covid-19 City-Scale Simulation Team. City-Scale Epidemic Simulator on Github. `https://cni-iisc.github.io/epidemic-simulator/`, 2020.

[156] D. Thompson. What's Behind South Korea's COVID-19 Exceptionalism?. The Atlantic. `https://www.theatlantic.com/ideas/archive/2020/05/whats-south-koreas-secret/611215/`, 2020.

[157] Michalis K Titsias. Learning model reparametrizations: Implicit variational inference by fitting mcmc distributions. *arXiv preprint arXiv:1708.01529*, 2017.

[158] Dustin Tran, Rajesh Ranganath, and David M Blei. The variational gaussian process. `https://arxiv.org/abs/1511.06499`, 2015.

[159] Pauline van den Driessche. Reproduction numbers of infectious disease models. Infectious Disease Modelling 2, 288-303, 2017.

[160] Udo von Toussaint. Bayesian inference in physics. *Rev. Mod. Phys.*, 83:943–999, Sep 2011.

[161] Eric-Jan Wagenmakers and Simon Farrell. Aic model selection using akaike weights. *Psychonomic bulletin & review*, 11(1):192–196, 2004.

[162] Chaolong Wang, Li Liu, Xingjie Hao, Huan Guo, Qi Wang, Jiao Huang, Na He, Hongjie Yu, Xihong Lin, An Pan, et al. Evolving epidemiology and impact of non-pharmaceutical interventions on the outbreak of coronavirus disease 2019 in wuhan, china. `https://www.medrxiv.org/content/10.1101/2020.03.03.20030593v1`, 2020.

[163] Steve R Waterhouse, David MacKay, and Anthony J Robinson. Bayesian methods for mixtures of experts. advances in neural information processing systems, 351-357, 1996.

[164] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011.

[165] WHO. Coronavirus disease 2019 (COVID-19) Situation Report - 174, 12 July 2020, 2020.

[166] Christopher KI Williams and Carl Edward Rasmussen. Gaussian processes for machine learning. Mit Press. 2, 2006.

[167] Joseph T Wu, Kathy Leung, and Gabriel M Leung. Nowcasting and forecasting the potential domestic and international spread of the 2019-nCov outbreak originating in Wuhan, China: a modelling study. The Lancet 395, 689-697, 2020.

[168] Kai Xu, Hong Ge, Will Tebbutt, Mohamed Tarek, Martin Trapp, and Zoubin Ghahramani. Advancedhmc. jl: A robust, modular and efficient implementation of advanced hmc algorithms. 2019.

[169] Zhe Xu, Lei Shi, Yijin Wang, Jiyuan Zhang, Lei Huang, Chao Zhang, Shuhong Liu, Peng Zhao, Hongxia Liu, Li Zhu, Yanhong Tai, Prof Changqing Bai, Tingting Gao, Jinwen Song, Peng Xia, Jinghui Dong, and Prof Jingmin Zhao. Pathological findings of COVID-19 associated with acute respiratory distress syndrome. The Lancet respiratory medicine 8, 420-422, 2020.

[170] Yibo Yang, Mohamed Aziz Bhouri, and Paris Perdikaris. Bayesian differential programming for robust systems identification under uncertainty. *Proceedings of the Royal Society A*, 476(2243):20200290, 2020.

[171] Bo Yuan, Ciyue Shen, Augustin Luna, Anil Korkut, Debora S Marks, John Ingraham, and Chris Sander. Cellbox: interpretable machine learning for perturbation biology with application to the design of cancer combination therapy. *Cell systems*, 12(2):128–140, 2021.