Available online at www.sciencedirect.com
ScienceDirect

Comput. Methods Appl. Mech. Engrg. 402 (2022) 115346

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Bayesian Physics Informed Neural Networks for real-world nonlinear dynamical systems

Kevin Linka[a], Amelie Schäfer[b], Xuhui Meng[c], Zongren Zou[c], George Em Karniadakis[c], Ellen Kuhl[b,*]

[a] *Institute of Continuum and Material Mechanics, Hamburg University of Technology, Hamburg, Germany*
[b] *Department of Mechanical Engineering, Stanford University, Stanford, CA, United States*
[c] *Division of Applied Mathematics, Brown University, Providence, RI, United States*

Available online 15 July 2022

Dataset link: https://github.com/LivingMatterLab/xPINNs

Abstract

Understanding real-world dynamical phenomena remains a challenging task. Across various scientific disciplines, machine learning has advanced as the go-to technology to analyze nonlinear dynamical systems, identify patterns in big data, and make decision around them. Neural networks are now consistently used as universal function approximators for data with underlying mechanisms that are incompletely understood or exceedingly complex. However, neural networks alone ignore the fundamental laws of physics and often fail to make plausible predictions. Here we integrate data, physics, and uncertainties by combining neural networks, physics informed modeling, and Bayesian inference to improve the predictive potential of traditional neural network models. We embed the physical model of a damped harmonic oscillator into a fully-connected feed-forward neural network to explore a simple and illustrative model system, the outbreak dynamics of COVID-19. Our Physics Informed Neural Networks seamlessly integrate data and physics, robustly solve forward and inverse problems, and perform well for both interpolation and extrapolation, even for a small amount of noisy and incomplete data. At only minor additional cost, they self-adaptively learn the weighting between data and physics. They can serve as priors in a Bayesian Inference, and provide credible intervals for uncertainty quantification. Our study reveals the inherent advantages and disadvantages of Neural Networks, Bayesian Inference, and a combination of both and provides valuable guidelines for model selection. While we have only demonstrated these different approaches for the simple model problem of a seasonal endemic infectious disease, we anticipate that the underlying concepts and trends generalize to more complex disease conditions and, more broadly, to a wide variety of nonlinear dynamical systems. Our source code and examples are available at https://github.com/LivingMatterLab/xPINNs.
© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Keywords: Dynamical systems; Machine learning; Neural Networks; Physics Informed Neural Networks; Bayesian Inference; Bayesian Neural Networks

* Corresponding author.
E-mail address: ekuhl@stanford.edu (E. Kuhl).

https://doi.org/10.1016/j.cma.2022.115346
0045-7825/© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Motivation

Modeling and predicting the behavior of complex nonlinear dynamical systems remains a challenging scientific task [1]. Numerous scientists are addressing this challenge by collecting more observational data than ever before; however, more often than not without a clear picture how to sort, analyze, and understand these enormous amounts of information [2]. Throughout the past two decades, machine learning has advanced as the go-to technology to analyze big data and explore the massive design spaces associated with them [3]. Simply put, machine learning is an incredibly powerful strategy to make data-driven recommendations and decisions based on the input data alone. As such, it is has become an indispensable technology for image or speech recognition, medical diagnostics, or self-driving cars. However, traditional machine learning ignores the fundamental laws of physics, and, as a consequence, performs well at fitting observations, but often fails to make consistent and plausible predictions. In the engineering science community, especially in disciplines that are traditionally not data-rich, these limitations have raised the question how we can advance machine learning and build in our prior knowledge to respect underlying physical principles.

The trend of embedding physics into machine learning has recently drawn tremendous attention in various scientific applications [4]. As engineering scientists, we have been modeling complex nonlinear dynamical systems for hundreds of years and we have developed a reasonable quantitative knowledge about physical parameters, boundary conditions, and constraints [5]. Physics-informed machine learning now allows us to use this prior knowledge when training machine learning tools [3]. This not only makes the training more efficient, but also more accurate and robust, especially when working with missing, noisy, or sparse real-life data [6]. A powerful and effective strategy to seamlessly integrate data and physics are Physics Informed Neural Networks [7]. Physics Informed Neural Networks introduce a learning bias by directly embedding the physics into the loss function of a neural network [8]. This makes the neural network more robust, especially in the presence of sparse or imperfect data, and can provide more accurate and physically consistent predictions, even outside the training window [2]. For example, recent studies have successfully applied Physics Informed Neural Networks to study the complex outbreak dynamics of COVID-19 [9,10] by integrating advance epidemiology models into deep neural networks [11,12]. However, the success of these methods depends crucially on the amount of available data and the complexity of the system itself. For the COVID-19 pandemic, we now know that predictions typically fail beyond a two-week window [13,14], and that predicting outbreak dynamics beyond this range can have devastating consequences [15]. It seems intuitive to ask how we can quantify uncertainties to estimate the reliability of our models and to build confidence in our model predictions [16].

Real-world data are inherently stochastic, noisy, and incomplete, and naturally contain aleatoric uncertainty [17]. This is why we should never just look at the plain data. Instead, we should always interpret data in the context of models that allow us to quantifying uncertainties [18]. A recently proposed promising strategy is to interpret Physics Informed Neural Networks within the framework of Bayesian statistics [6]. This approach combines a Bayesian Neural Network with a Physics Informed Neural Network to create prior distributions and uses Hamilton Monte Carlo methods to estimate posterior distributions [19]. Uncertainty quantification in the form of credible intervals is a natural and important by-product of such a Bayesian analysis [20]. Throughout the past decades, Bayesian statistics have undeniably gained massive popularity [21] and have stimulated the birth to an entirely new field, probabilistic programming [22]. Combined with Physics Informed Neural Networks, Bayesian statistics perform well for both forward and inverse problems including Poisson problems, flow problems, phase separation problems, and reaction–diffusion problems [19]. However, most of these applications train their network on synthetic data from a known solution with overlaid noise [23]. This motivates the question how well Bayesian Physics Informed Neural Networks perform on real-word data from a nonlinear dynamical system for which the underlying physics are not entirely known, with both aleatoric and epistemic uncertainty.

The objective of our study is to illustrate the performance of Neural Networks, Bayesian Inference, and a combination of both using real-world data. However, rather than showcasing these methods on real big data, we focus on a simple data set that is easy to understand, interpret, and reproduce: the reported daily number of new COVID-19 cases worldwide [24] throughout the year of 2021 [25]. We identify a suitable, yet simple mathematical model with an analytical solution that allows us to characterize the data. Again, our objective is not to develop the most rigorous physical model to explain each and every feature of the outbreak dynamics of a global pandemic [26]. Instead, we adopt simple physical model, a damped harmonic oscillator, to characterize the long-term dynamics of COVID-19 on a global scale. We integrate the case data and the oscillator model using Neural Networks and

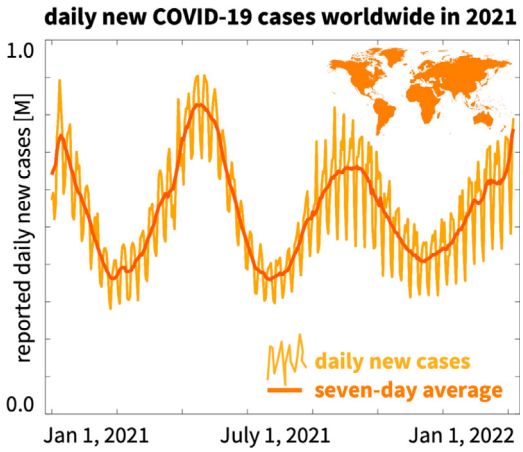

Fig. 1. Data - Daily new COVID-19 cases worldwide in 2021. Thin fluctuating line illustrates the reported daily new cases of COVID-19 worldwide; thick smooth line illustrates the seven-day moving average of the daily new cases, from January 1, 2021 to December 31, 2021.

Bayesian Inference and compare their model equations, parameters, training, predictions, and uncertainties. We envision that this study sheds light on the performance of different cutting-edge machine learning tools, highlights their advantages and disadvantages, and helps engineering scientists to select the best method to analyze real-world nonlinear dynamical systems.

2. Model problem

2.1. Data - Daily new COVID-19 cases

Fig. 1 illustrates our real-world data to compare different machine learning approaches. As example, we use the number of new COVID-19 cases worldwide [27] throughout the entire year of 2021, which we draw from a public database [25]. The thin fluctuating line shows the raw data, the reported daily new cases worldwide from January 1, 2021 to December 31, 2021. Global reporting started on January 22, 2020 with a total number of cases of 557 [28]. On January 1, 2021, at the beginning of our analysis window, the number of newly reported cases worldwide was 572,602 bringing the total number of cases up to 84,332,767. On December 31, 2021, at the end of our window, the total number of cases was 288,631,129. Altogether, the reporting window saw three pronounced waves. The first wave had a minimum number of new cases of 281,223 on day 46, February 16, 2021, and a maximum number of 905,378 on day 118, April 29, 2021, as we can confirm from the global minimum and maximum in Fig. 1. The second wave had a minimum number of new cases of 296,808 on day 172, June 22, 2021, and a maximum number of 819,336 cases on day 221, August 10, 2021, about four months after. To eliminate noise, reporting uncertainties, and weekday–weekend fluctuations, data analysts typically average the reported case numbers across a seven-day window [29]. The thick smooth line illustrates the seven-day moving average $\hat{x}(t)$ of the daily new cases. In the following sections, we will use both the raw data and their seven-day moving average as an example to illustrate the potential of Neural Networks, Bayesian Inference, and both methods combined.

2.2. Neural Network modeling - Fully-connected feed-forward Neural Network

To model the data in Fig. 1 without any prior knowledge of the underlying physics, we use a neural network that approximates the model solution $\mathbf{x}$, the reported daily new COVID-19 cases, by taking the time coordinates $\mathbf{t}$ as input. We adopt one of the simplest examples of neural networks, a fully-connected feed-forward neural network composed of multiple hidden layers. We denote the hidden variables of the kth hidden layer as $\mathbf{z}_k$ and summarize

the neural network, here for two hidden layers, as

$$\begin{aligned} \mathbf{z}_0 &= \mathbf{t} \\ \mathbf{z}_1 &= \sigma(\mathbf{W}_1\,\mathbf{z}_0 + \mathbf{b}_1) \\ \mathbf{z}_2 &= \sigma(\mathbf{W}_2\,\mathbf{z}_1 + \mathbf{b}_2) \\ \mathbf{z}_3 &= \mathbf{W}_3\,\mathbf{z}_2 + \mathbf{b}_3 \end{aligned} \tag{1}$$

where the output of the last layer is used to approximate the true solution, $\mathbf{x} \approx \mathbf{z}_3$. Here, $\sigma(\circ)$ denotes the non-linear activation function for which we select a hyperbolic tangent function, $\sigma(\circ) = \tanh(\circ)$, and $\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\}$ are the trainable network parameters, the weight matrix $\mathbf{W}_k$ and the bias vector $\mathbf{b}_k$ of the kth layer of the network. Throughout this project, for illustrative purposes, we consider a fully-connected feed-forward network with one input, the time t, two hidden layers with 32 nodes each, and one output, the daily new cases $x(t)$. For this type of network, $\mathbf{W}_1 \in \mathbb{R}^{1\times 32}$, $\mathbf{W}_2 \in \mathbb{R}^{32\times 32}$, $\mathbf{W}_3 \in \mathbb{R}^{32\times 1}$, and $\mathbf{b}_1 \in \mathbb{R}^{32}$, $\mathbf{b}_2 \in \mathbb{R}^{32}$, $\mathbf{b}_3 \in \mathbb{R}^{1}$, resulting in $32 + 32 \times 32 + 32 = 1088$ weights and $32 + 32 + 1 = 65$ biases.

2.3. Physic Informed modeling - Damped harmonic oscillator

To model the data in Fig. 1 using a physics-based model, we consider a general parameterized second order differential equation,

$$r(t, x, \dot{x}, \ddot{x}, \boldsymbol{\vartheta}) = 0 \tag{2}$$

where $t \in [\,0, T\,]$ is the time, $x(t)$ is the solution with initial conditions $x(t_0) = x_0$ and $\dot{x}(t_0) = \dot{x}_0$, r is the residual as a function of x and its first and second time derivatives $\dot{x}$ and $\ddot{x}$, and $\boldsymbol{\vartheta}$ are the physics parameters. We now specify the general Eq. (2) for the model problem of a damped harmonic oscillator, which is governed by Newton's second law, the balance of forces, $m\,\ddot{x} + c\,\dot{x} + k\,x = 0$, or, divided by the mass m,

$$r = \ddot{x} + \frac{c}{m}\,\dot{x} + \frac{k}{m}\,x = 0 \qquad \text{or} \qquad r = \ddot{x} + 2\,\zeta\,\omega_0\,\dot{x} + \omega_0^2\,x = 0\,. \tag{3}$$

We can parameterize the damped harmonic oscillator equation in terms of the mass m, the viscous damping coefficient c, and the stiffness k, or, equivalently, in terms of the damping ratio $\zeta = \delta/\omega_0 = c/(2\sqrt{mk})$, the damping $\delta = c/(2m)$, and the angular frequency $\omega_0 = \sqrt{k/m}$. Eqs. ((3).1) and ((3).2) are homogeneous linear differential equations of second order with constant coefficients. For solutions to this type of equations, we can make an exponential ansatz of the form $x(t) = C\,\exp(\lambda\,t)$, such that $\dot{x}(t) = \lambda\,C\,\exp(\lambda\,t)$ and $\ddot{x}(t) = \lambda^2 C\,\exp(\lambda\,t)$. Inserting this ansatz into Eqs. ((3).1) and ((3).2) yields their characteristic equations,

$$\lambda^2 + \frac{c}{m}\,\lambda + \frac{k}{m} = 0 \qquad \text{or} \qquad \lambda^2 + 2\,\delta\,\lambda + \omega_0^2 = 0\,. \tag{4}$$

These equations are quadratic equations in normal form with two solutions for λ,

$$\lambda_{1,2} = -\frac{c}{2m} \pm \sqrt{\left(\frac{c}{2m}\right)^2 - \frac{k}{m}} \qquad \text{or} \qquad \lambda_{1,2} = -\delta \pm \sqrt{\delta^2 - \omega_0^2}\,. \tag{5}$$

From Eq. ((5).2) we conclude that the damping ratio $\zeta = \delta/\omega_0$, the ratio between damping δ and frequency ω_0, determines the type of the solution and with it the behavior of the system: A damped harmonic oscillator can be overdamped for $\zeta > 1$ and $\delta > \omega_0$ with two different real valued solutions for λ; critically damped for $\zeta = 1$ and $\delta = \omega_0$ with two identical real valued solutions for λ, or underdamped for $\zeta < 1$ and $\delta < \omega_0$ with two conjugate complex solutions for λ. Here we consider the underdamped case. The analytical solution of an underdamped harmonic oscillator describes an exponential decay of the oscillation,

$$x(t) = 2A_0\,\cos(\omega\,t + \phi)\,\exp(-\delta\,t)\,, \tag{6}$$

where A_0 is the amplitude, $\omega = \sqrt{\omega_0^2 - \delta^2}$ is the natural frequency, and $\phi = \arctan(-\delta/\omega)$ is the phase angle.

Fig. 2 illustrates the dynamics of a damped harmonic oscillator. The solid line highlights the analytical solution $m\,\ddot{x} + c\,\dot{x} + k(x - x_0) = 0$ or $\ddot{x} + 2\,\zeta\omega_0\,\dot{x} + \omega_0^2\,(x - x_0) = 0$, the dashed lines highlight the exponential decay in amplitude, $x_0 \pm A_0\,\exp(-\delta t)$, where A_0 is the amplitude, $\delta = c/(2m)$ is the damping, $\omega_0 = \sqrt{k/m}$ is the frequency associated with the period $T = 2\pi/\omega_0$, and x_0 is the offset. We fix the value of the mass to $m = 1$ and collectively summarize the remaining physics parameters in the parameter vector $\boldsymbol{\vartheta} = \{c, k, x_0\}$.

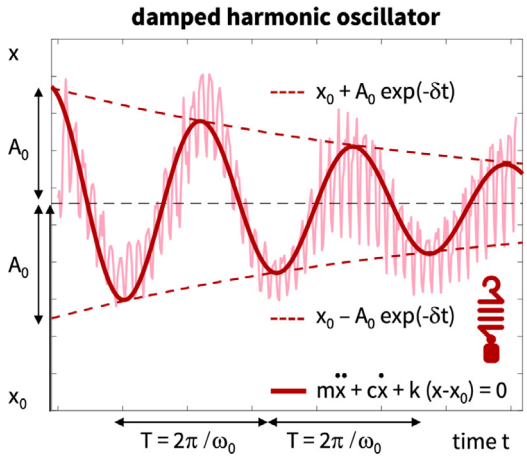

Fig. 2. Physics - Damped harmonic oscillator. Characteristic exponential decay of the oscillation in terms of the period $T = 2\pi/\omega_0$ with frequency $\omega_0 = \sqrt{k/m}$, damping $\delta = c/(2m)$, offset x_0, and amplitude A_0. Solid lines illustrate the analytical solution $m\,\ddot{x}+c\,\dot{x}+k(x-x_0) = 0$ or $\ddot{x} + 2\,\zeta\omega_0\,\dot{x} + \omega_0^2\,(x - x_0) = 0$, dashed lines illustrate the exponential decay in amplitude $x_0 \pm A_0\,\exp(-\delta t)$ around the offset x_0.

2.4. Machine Learning - Integrating data and physics

To create learning machines that seamlessly integrate the data from Fig. 1, the physics from Fig. 2, or both, we combine Neural Network modeling from Section 2.2 and Physics Informed modeling from Section 2.3 and learn the underlying network and physics parameters. To quantify the uncertainty of the learned parameters, and with it the quality of the model, we can integrate each of these methods into a Bayesian Inference.

Fig. 3 motivates the use of classical Neural Networks, Physics Informed Neural Networks, and classical Bayesian Inference before we describe these methods in detail in Sections 3 and 4. The thin yellow lines indicate daily new case data, the thick orange lines are their seven-day average, the blue dots are the training data, here for 225 out of 365 days, the dashed red lines are the model simulations, and light blue areas are the credible intervals of the model prediction. The left column uses the thin yellow lines of the daily new case data, and the right column uses the thick orange lines of their seven-day average for training the models.

Neural Networks in the top row approximate the training data of the first 225 days well, but fail to predict the behavior outside the training window of the remaining 140 days. The neural network model simply continues the learned trend. Since day 225 corresponds to a plateau with a horizontal tangent, the neural network model predicts a flatline with constant case numbers for the remaining part of the year. This trend is similar when trained on both the daily case data in the top left and the seven-day average in the top right.

Physics Informed Neural Networks in the middle row approximate both the training data of the first 225 days and the behavior outside the training window of the remaining 140 days with a good accuracy. Throughout the entire year, the red dashed lines of the model remain close to the orange line of the seven-day case average. The fit of the Physics Informed Neural Network is virtually identical for training on the daily case data in the middle left and on the seven-day average in the middle right.

Bayesian Inference in the bottom row not only approximates both, the training data and the behavior outside the training window, but also provides credible intervals to estimate the quality of the fit. Since the Bayesian Inference uses the same underlying physics as the Physics Informed Neural Network, its red dashed lines also remain close to the orange line of the seven-day case average throughout the entire year. However, the fit with respect to the training data is worse compared to the Physics Informed Neural Network, because we bias the solution towards the physics equations. In contrast to the previous two models in the top and middle rows, the Bayesian Inference displays a clear difference between training on raw versus averaged data. The light blue areas of its credible intervals are notably wider when the model parameters are inferred using the daily case data in the bottom left than when using the seven-day average data in the bottom right. We will now describe details of the underlying equations for Neural Networks in Section 3 and for Bayesian Inference in Section 4.

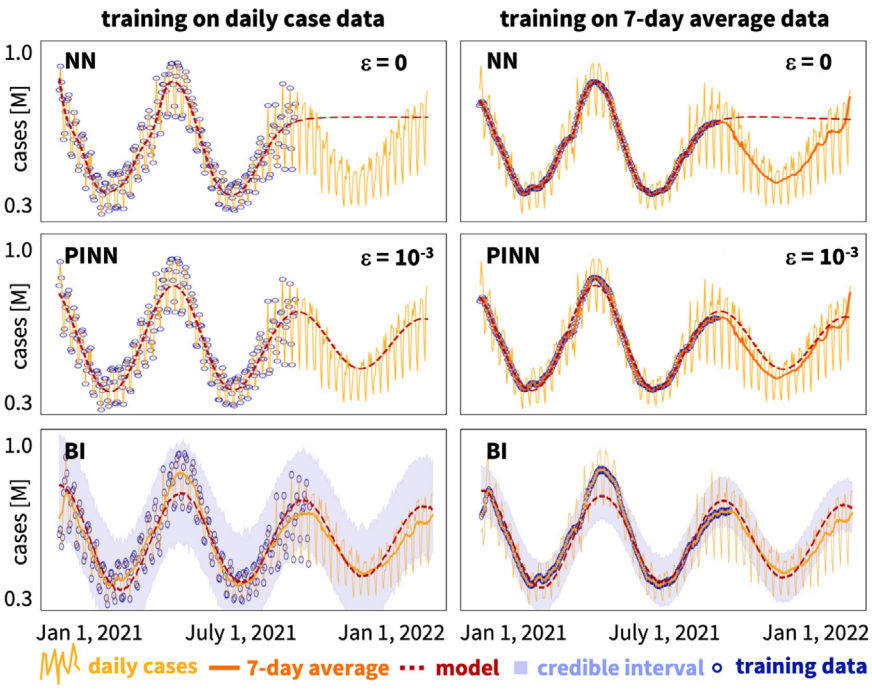

Fig. 3. Machine learning - Integrating data and physics. Neural Networks approximate the training data well, but fail to predict the behavior outside the training window, when trained on both, the daily case data (top left) and the seven-day average (top right). Physics Informed Neural Networks approximate both the training data and the behavior outside the training window, when trained on both, the daily case data (middle left) and the seven-day average data (middle right). Bayesian Inference not only approximates both, the training data and the behavior outside the training window, but also provides credible intervals, which are wider when model parameters are inferred using the daily case data (bottom left) than when using the seven-day average data (bottom right). Thin yellow lines indicate daily new case data, thick orange lines are their seven-day average, dashed red lines are the model simulations, light blue areas are its credible intervals, blue dots are the training data, here for 225 out of 365 days.

3. Neural Network modeling

The objective of Neural Network modeling is to train the neural network (1) such that the model output $\mathbf{x}(\mathbf{t})$ best approximates the data $\hat{\mathbf{x}}$, by minimizing a loss function,

$$L(\boldsymbol{\Theta}\,;t) \rightarrow \min\,, \tag{7}$$

through iteratively updating the set of model parameters $\boldsymbol{\Theta}$. The set of model parameters could consist of the network parameters $\boldsymbol{\theta}$, the physics parameters $\boldsymbol{\vartheta}$, and a weighting coefficient ε,

$$\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \boldsymbol{\vartheta}, \varepsilon\} \qquad \text{with} \qquad \boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\} \qquad \text{and} \qquad \boldsymbol{\vartheta} = \{c, k, x_0\}, \tag{8}$$

where $\mathbf{W}_k$ and $\mathbf{b}_k$ are the network weights and biases from Eq. (1) and c, k, and x_0 are the physical damping, stiffness, and offset from Eq. (3). We specify the loss function (7) and the relevant parameters (8) for classical Neural Networks in Section 3.1, for Physics Informed Neural Networks in Section 3.2, and for Self Adaptive Physics Informed Neural Networks in Section 3.3.

3.1. Neural Networks

The objective of classical Neural Networks is to learn the network parameters $\boldsymbol{\theta}$, without prior knowledge of the underlying physics, by minimizing a loss function (7) that consists of a single term L_{data},

$$L(\boldsymbol{\theta}\,;t) = L_{\text{data}} \rightarrow \min\,. \tag{9}$$

The data loss L_{data} penalizes the error between model $\mathbf{x}(\mathbf{t})$ and data $\hat{\mathbf{x}}$. We define it as the mean square error, the L_2-norm of the difference between model and data, $\| \mathbf{x}(\mathbf{t}) - \hat{\mathbf{x}} \|^2$, divided by the number of training points n_{trn},

$$L_{\text{data}}(\boldsymbol{\theta}\,;t) = \frac{1}{n_{\text{trn}}} \sum_{i=1}^{n_{\text{trn}}} \| \, x(t_i) - \hat{x}_i \, \|^2 \,. \tag{10}$$

We minimize the loss function (9) in terms of the mean square error (10) to optimize the network parameters $\boldsymbol{\theta}$, the network weights and biases $\mathbf{W}_k$ and $\mathbf{b}_k$,

$$\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\} \,, \tag{11}$$

using the ADAM optimizer, an adaptive algorithm for gradient-based first-order optimization.

3.2. Physics Informed Neural Networks

The objective of Physics Informed Neural Networks is to learn both the network parameters $\boldsymbol{\theta}$ and the physics parameters $\boldsymbol{\vartheta}$ by simultaneously training a neural network (1) and solving an underlying physics equation (3). We convert both into a single optimization problem that minimizes the loss function L to learn the parameters $\boldsymbol{\Theta} = \{\, \boldsymbol{\theta}, \boldsymbol{\vartheta} \,\}$,

$$L(\boldsymbol{\Theta}; t) = (\, 1 - \varepsilon \,)\, L_{\text{data}} + \varepsilon \, L_{\text{phys}} \rightarrow \min \,. \tag{12}$$

The first term, the data loss L_{data}, penalizes the error between model $\mathbf{x}(\mathbf{t})$ and data $\hat{\mathbf{x}}$, the second term, the physics loss L_{phys}, penalizes the physics residual $\mathbf{r}$. Since both terms can be of a different order of magnitude, we weight them with the weighting coefficients $(\, 1 - \varepsilon \,)$ and ε. For the first term, we use the mean square error, the L_2-norm of the difference between model and data, $\| \mathbf{x}(\mathbf{t}) - \hat{\mathbf{x}} \|^2$, divided by the number of training points n_{trn} similar to Section 3.1,

$$L_{\text{data}}(\boldsymbol{\Theta}; t) = \frac{1}{n_{\text{trn}}} \sum_{i=1}^{n_{\text{trn}}} \| \, x(t_i) - \hat{x}_i \, \|^2 \,. \tag{13}$$

For the second term, we use the L_2-norm of the residual, $\| \, \mathbf{r} \, \|^2$, divided by the total number of sampling points n_{smp},

$$L_{\text{phys}}(\boldsymbol{\Theta}; t) = \frac{1}{n_{\text{smp}}} \sum_{i=1}^{n_{\text{smp}}} \| \, r(t_i) \, \|^2 = \frac{1}{n_{\text{smp}}} \sum_{i=1}^{n_{\text{smp}}} \| \, \ddot{x}(t_i) + \frac{c}{m} \dot{x}(t_i) + \frac{k}{m} \, (x(t_i) - x_0) \, \|^2 \,. \tag{14}$$

A key step in calculating the physical loss function (14) is to compute the time derivatives of the model solution x, $\dot{x} = \mathrm{d}x/\mathrm{d}t$ and $\ddot{x} = \mathrm{d}\dot{x}/\mathrm{d}t$, which we address using automatic differentiation. Automatic differentiation is part of various deep learning frameworks which makes it convenient for Physics Informed Neural Networks to evaluate derivatives. We minimize the overall loss function (12) to optimize the network parameters $\boldsymbol{\theta}$, the network weights $\mathbf{W}_k$ and biases $\mathbf{b}_k$, and the physics parameters $\boldsymbol{\vartheta}$, the viscous damping c, stiffness k, and offset x_0,

$$\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\} \quad \text{and} \quad \boldsymbol{\vartheta} = \{c, k, x_0\} \,, \tag{15}$$

again using the ADAM optimizer, an adaptive algorithm for gradient-based first-order optimization.

3.3. Self Adaptive Physics Informed Neural Networks

The objective of Self Adaptive Physics Informed Neural Networks is to train a neural network (1) and solve an underlying physics equation (3) similar to Section 3.2, but now, rather than prescribing the weighting coefficient ε between data and physics, we learn it as a function of time t, along with the other parameters, $\boldsymbol{\Theta} = \{\, \boldsymbol{\theta}, \boldsymbol{\vartheta}, \epsilon \,\}$ [30]. We adopt the same loss function L as in Section 3.2,

$$L(\boldsymbol{\Theta}; t) = [\, 1 - \varepsilon(t) \,]\, L_{\text{data}} + \varepsilon(t)\, L_{\text{phys}} \,, \tag{16}$$

in terms of the data loss, the weighted error between model $\mathbf{x}$ and data $\mathbf{x}_{\text{data}}$,

$$L_{\text{data}}(\boldsymbol{\Theta}; t) = \frac{1}{n_{\text{trn}}} \sum_{i=1}^{n_{\text{trn}}} \| \, x(t_i) - \hat{x}_i \, \|^2 \,, \tag{17}$$

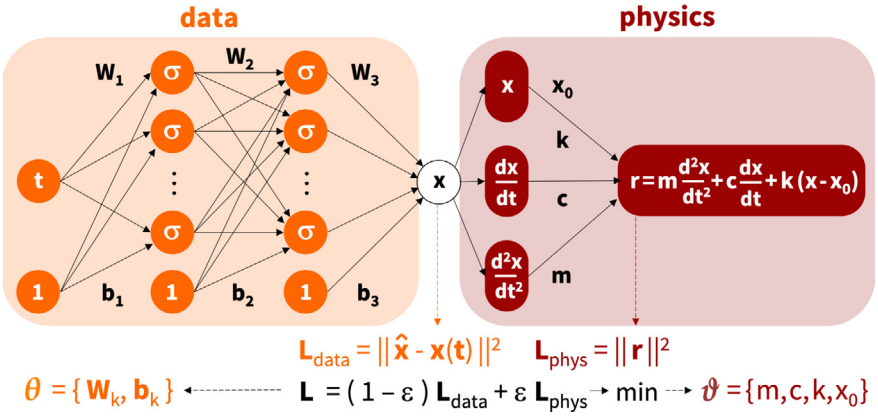

Fig. 4. Neural Network modeling. Neural Networks minimize a loss function $L \to$ min that could consist of two terms, a data loss L_{data}, the error between data and model $\| \hat{\mathbf{x}} - \mathbf{x}(\mathbf{t}) \|$, and a physics loss L_{phys}, the physics residual $\| \mathbf{r} \|$, to learn the network parameters $\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\}$, the physics parameters $\boldsymbol{\vartheta} = \{c, k, x_0\}$, and possibly even the dynamic weighting term $\varepsilon(t)$ between L_{data} and L_{phys}.

and the physics loss, the weighted norm of the physics residual $\mathbf{r}$,

$$L_{\text{phys}}(\boldsymbol{\Theta}; t) = \frac{1}{n_{\text{smp}}} \sum_{i=1}^{n_{\text{smp}}} \| r(t_i) \|^2 = \frac{1}{n_{\text{smp}}} \sum_{i=1}^{n_{\text{smp}}} \| \ddot{x}(t_i) + \frac{c}{m} \dot{x}(t_i) + \frac{k}{m} (x(t_i) - x_0) \|^2 . \tag{18}$$

We minimize the overall loss function (16) to optimize the network parameters, $\mathbf{W}_k$ and $\mathbf{b}_k$, the physics parameters, c, k, and x_0, and the weighting term $\varepsilon(t)$,

$$\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\} \quad \text{and} \quad \boldsymbol{\vartheta} = \{m, c, k, x_0\} \quad \text{and} \quad \varepsilon(t) . \tag{19}$$

The dynamics of the weighting term $\varepsilon(t)$ provide insight into the relative importance of data loss L_{data} and physics loss L_{phys} and into the change of both terms in time.

3.4. Comparison of Neural Network models

Fig. 4 illustrates the features the Neural Network models of this section. The classical Neural Networks from Section 3.1 minimize a loss function, $L \to$ min, that characterizes the data loss L_{data}, the error between data and model $\| \hat{\mathbf{x}} - \mathbf{x}(\mathbf{t}) \|$ shown on the left. The Physics Informed Neural Networks from Section 3.2 build additional physics loss tersm L_{phys} into the loss function L, for example, the physics residual $\| \mathbf{r} \|$ shown on the right. In a forward problem, Neural Networks learn the network parameters $\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\}$. In an inverse problem, Neural Networks learn the network parameters $\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\}$. the physics parameters $\boldsymbol{\vartheta} = \{c, k, x_0\}$. Self Adaptive Physics Informed Neural Networks from Section 3.3 also learn the dynamic weighting term $\varepsilon(t)$ between data and physics loss L_{data} and L_{phys}.

Fig. 5 illustrates the result of a grid search for the Physics Informed Neural Network. Here we show the simulation with one and two hidden layers, left and right columns, with two, four, and eight nodes each, from top to bottom. The number of network weights $\mathbf{W}_k$ and biases $\boldsymbol{b}_k$ increases with increasing number of layers and nodes. The one-hidden-layer models have $n_\Theta = 10$, 16, and 28 parameters and the two-hidden-layer models have $n_\Theta = 16$, 36, and 100 parameters. The six graphs confirm our intuition that the performance of the Physics Informed Neural Network increases with increasing number of hidden layers, from left to right, and increasing number of nodes per layer, from top to bottom. For both models, we also performed simulations with 16 and 32 nodes, but the performance did not increase markedly beyond the eight-node models. Fig. 5 suggests that the one-hidden-layer models perform poorly, independent of the number of nodes. For each additional hidden layer, the number of parameters increases with the number of nodes squared and the risk of overfitting increases. The two-hidden-layer models seem to be a reasonable compromise between underfitting and overfitting: They perform well, even with only eight nodes and $n_\Theta = 100$ parameters. Throughout this manuscript, we use Neural Networks with two hidden layers and 32 nodes.

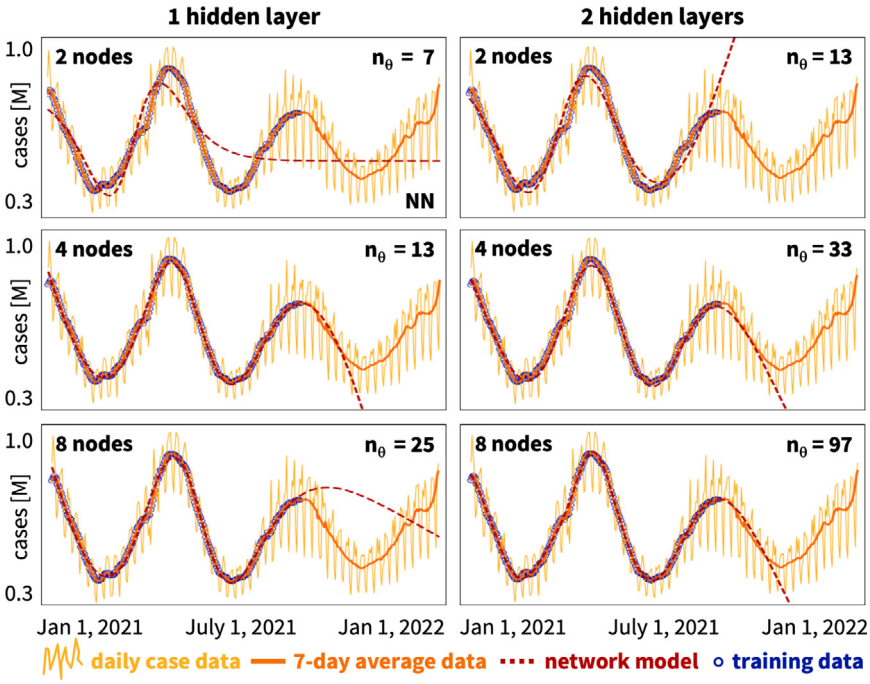

Fig. 5. Neural Network modeling. Grid search for Physics Informed Neural Network. The performance and number of parameters of the Physics Informed Neural Network increases with increasing number of hidden layers (left to right) and increasing number of nodes per layer (top to bottom). For both one and two hidden layers, the performance does not increase markedly beyond eight nodes (bottom row). Throughout this manuscript, we use a Neural Network with two hidden layers and 32 nodes. Thin yellow lines indicate daily new case data, thick orange lines are their seven-day average, dashed red lines are the physics-based model, light blue areas are its credible intervals, blue dots are the training data.

Fig. 6 compares the classical Neural Network with Physics Informed Neural Networks with varying weighting coefficient ε. The thin yellow lines indicate daily new case data, the thick orange lines are their seven-day average, the blue dots are the training data, here for 225 out of 365 days, the dashed red lines are the simulation with the physics-based model. In Physics Informed Neural Networks, the loss function, $L = (1-\varepsilon)\,L_{\text{data}} + \varepsilon\,L_{\text{phys}}$, is a weighted average of the data-based mean square error, L_{data}, and the physics-based residual, L_{phys}. For the special case of $\varepsilon = 0$ in the top left, the loss function degenerates to the loss function of the classical Neural Network $L \rightarrow L_{\text{data}}$ from Section 3.1. The classical Neural Network provides a good approximation of the training data during the first 225 days of the year, but fails to predict the behavior outside the training window. During the remaining 140 days, the classical Neural Network simply continues the learned trend, which, on day 225, suggests a plateau with constant case numbers for the remaining part of the year. For positive weighting coefficients $\varepsilon > 0$, the loss function accounts for both data loss L_{data} and physics loss L_{phys}. With increasing weighting coefficients $\varepsilon = [10^{-\infty}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-0}]$, from top left to bottom right, the influence of the data loss L_{data} decreases and the influence of the physics loss L_{phys} increases. In other words, the smaller the weighting coefficient ε, the better the fit of the data; the larger the weighting coefficient ε, the better the fit of the physics. For the special case of $\varepsilon = 1$ in the bottom right, the loss function degenerates to the loss function of the physical model $L \rightarrow L_{\text{phys}}$ from Section 2.3. The network solves the physics of the damped harmonic oscillator without accounting for any data, and converges to the trivial solution with a zero amplitude.

Fig. 7 quantifies the contributions to the loss function for Physics Informed Neural Networks with weighting coefficients ε varying from 10^{-6} to 10^{-2}. In Physics Informed Neural Networks, the loss function, $L = (1-\varepsilon)\,L_{\text{data}} + \varepsilon\,L_{\text{phys}}$, is a weighted average of the data-based mean square error, L_{data}, summarized in the left panel, and the physics-based residual, L_{phys}, summarized in the right panel. Each panel separately reports the descriptive

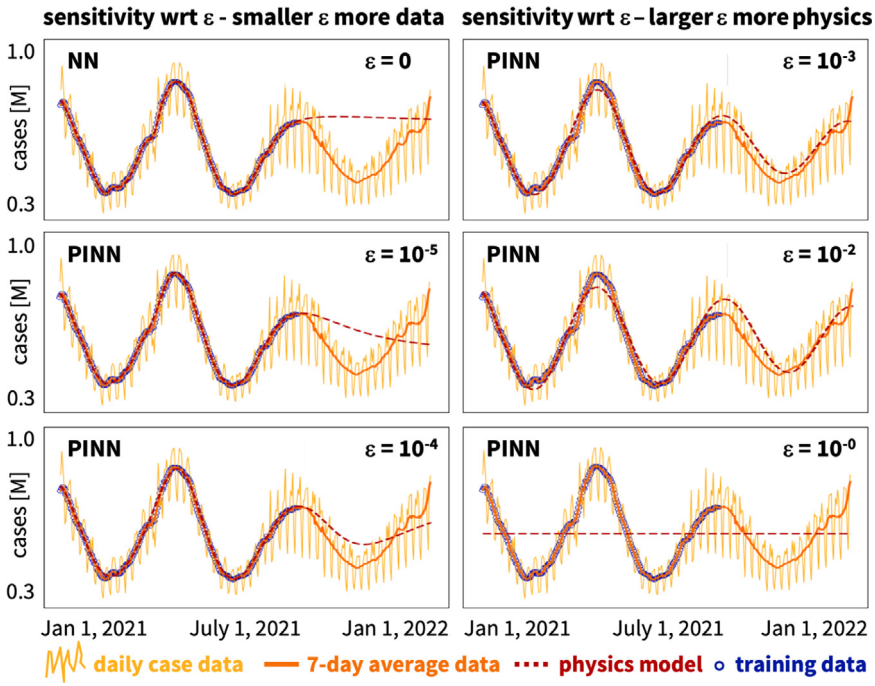

Fig. 6. Neural Network modeling. Classical Neural Network vs Physics Informed Neural Networks with varying weighting coefficients ε. In Physics Informed Neural Networks, the loss function, $L = (1-\varepsilon)\,L_{\text{data}} + \varepsilon\,L_{\text{phys}}$, is a weighted average of the data-based mean square error, L_{data}, and the physics-based residual, L_{phys}. The special case of $\varepsilon = 0$ represents a classical Neural Network (top left). With increasing weighting coefficients, $\varepsilon = [10^{-\infty}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-0}]$, the influence of the data fit decreases and the influence of the physics increases (top left to bottom right). Thin yellow lines indicate daily new case data, thick orange lines are their seven-day average, dashed red lines are the physics-based model, light blue areas are its credible intervals, blue dots are the training data, here for 225 out of 365 days.

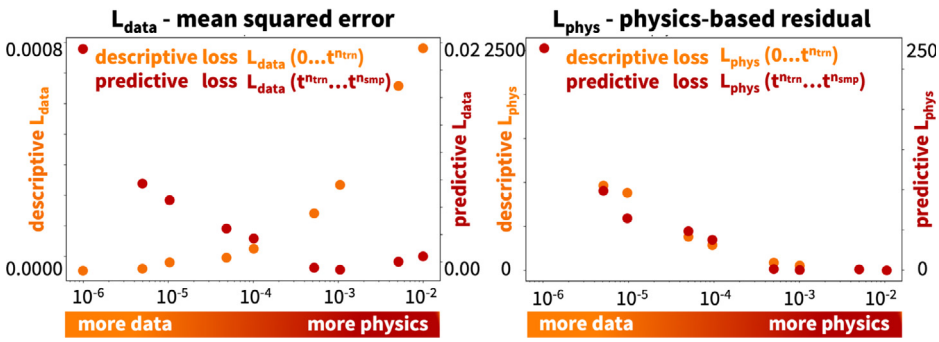

Fig. 7. Neural Network modeling. Sensitivity analysis for Physics Informed Neural Networks with varying weighting coefficients ε. In Physics Informed Neural Networks, the loss function, $L = (1-\varepsilon)\,L_{\text{data}} + \varepsilon\,L_{\text{phys}}$, is a weighted average of the data-based mean square error, L_{data}, and the physics-based residual, L_{phys}. With increasing weighting coefficients ε, the model emphasizes less on the data fit and more on the physics fit; the predictive data loss, descriptive physics loss, and predictive physics loss decrease, while the descriptive data loss increases. Orange dots indicate the descriptive losses L_{data} and L_{phys} for the first 255 days; red dots indicate the predictive losses L_{data} and L_{phys} for the following 140 days.

loss, the sum of the daily errors in the training window as orange dots,

$$L_{\text{data}}(\boldsymbol{\Theta}; t) = \frac{1}{255}\sum_{i=1}^{255} \| x(t_i) - \hat{x}_i \|^2 \qquad \text{and} \qquad L_{\text{phys}}(\boldsymbol{\Theta}; t) = \frac{1}{255}\sum_{i=1}^{255} \| r(t_i) \|^2,$$

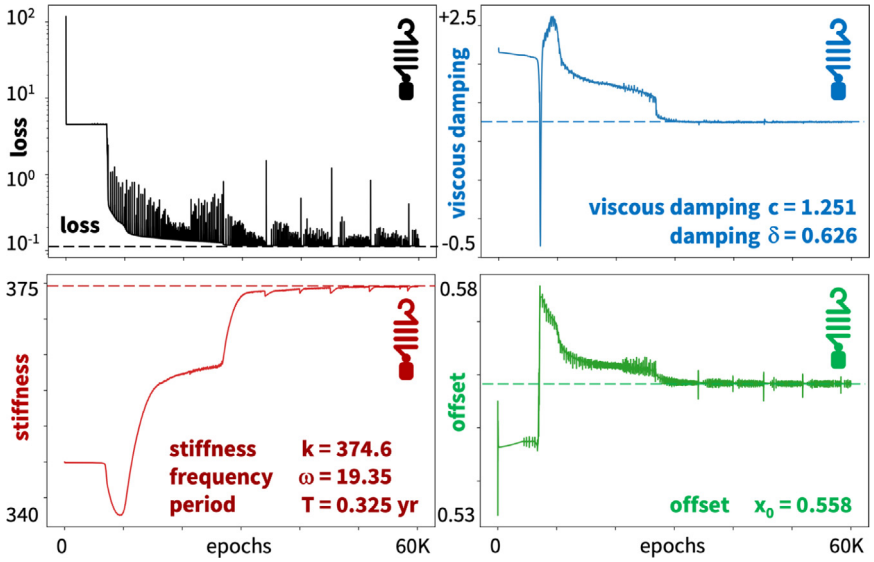

Fig. 8. Physics Informed Neural Network. History of loss function L and physics parameters damping c, stiffness k, and offset x_0. Within 60K epochs, the physics parameters converge towards a viscous damping coefficient of $c = 1.251$, a stiffness of $k = 374.6$, and an offset of $x_0 = 0.558$. This corresponds to a damping of $\delta = 0.626$, a frequency of $\omega = 19.35$, and a period of $T = 0.325$ years. Simulation with weighting coefficient of $\varepsilon = 10^{-3}$ and training window of $n_{\text{trn}} = 225$ days.

and the predictive loss, the sum of the daily errors in the prediction window as red dots,

$$L_{\text{data}}(\boldsymbol{\Theta}; t) = \frac{1}{140} \sum_{i=256}^{365} \| x(t_i) - \hat{x}_i \|^2 \qquad \text{and} \qquad L_{\text{phys}}(\boldsymbol{\Theta}; t) = \frac{1}{140} \sum_{i=256}^{365} \| r(t_i) \|^2.$$

With increasing weighting coefficients ε, the model emphasizes less on the data fit and more on the physics fit. The predictive data loss, descriptive physics loss, and predictive physics loss decrease, while the descriptive data loss increases. The quantitative comparison of the four losses in Fig. 7 agrees with the qualitative observation in Fig. 6: For smaller weighting coefficients ε, the red dashed lines of the model and orange lines of the seven-day case data average are in close proximity during the first 225 days of the descriptive regime, but deviate notably during the remaining 140 days of the predictive regime. For larger weighting coefficients ε, the red and orange lines deviate more during the first 225 days of the descriptive regime, but are in close proximity during the remaining 140 days of the predictive regime. This example highlights the importance of selecting an appropriate weighting coefficient ε, which can bias the result of a Physics Informed Neural Network either towards the data or towards the physics. An ideal weighting coefficient minimizes the sum of all four loss terms. From Fig. 7, we can estimate a reasonable weighting coefficient on the order of $\varepsilon \approx 10^{-3}$. This value is of the same order of magnitude as the learned weighting term $\varepsilon(t)$ with mean and standard deviation of $\varepsilon = 0.007 \pm 0.003$ for the self-adaptive Physics Informed Neural Network from Section 3.3.

Fig. 8 summarizes the history of the loss function $\boldsymbol{L}$ and of the physics parameters $\boldsymbol{\vartheta}$, the damping c, stiffness k, and offset x_0, for a Physics Informed Neural Network with a weighting coefficient of $\varepsilon = 10^{-3}$. For the first 30K epochs, the loss function decreases while the physics parameters are still oscillating without clear trends. After 60K epochs, the physics parameters have converged towards a viscous damping coefficient of $c = 1.251$, a stiffness of $k = 374.6$, and an offset of $x_0 = 0.558$. This corresponds to a damping of $\delta = 0.626$, a frequency of $\omega = 19.35$, and a period of $T = 0.325$ years. This period of approximately four months agrees well with the reported daily new cases in Fig. 1 that display three pronounced waves throughout the year of 2021.

4. Bayesian Inference modeling

The objective of Bayesian Inference is to estimate the posterior probability distribution of a set of the parameters $\boldsymbol{\Theta}$, such that the statistics of the model $\mathbf{x}(\mathbf{t})$ agree with data $\hat{\mathbf{x}}$ [31] and possibly also satisfy the physics $\mathbf{r}$, by

maximizing the prior-weighted likelihood $P(\hat{\mathbf{x}}, \mathbf{r}|\boldsymbol{\Theta})$,

$$P(\boldsymbol{\Theta}|\hat{\mathbf{x}}, \mathbf{r}) = \frac{P(\hat{\mathbf{x}}, \mathbf{r}|\boldsymbol{\Theta})\, P(\boldsymbol{\Theta})}{P(\hat{\mathbf{x}})P(\mathbf{r})} = \frac{P(\hat{\mathbf{x}}|\boldsymbol{\Theta})\, P(\mathbf{r}|\boldsymbol{\Theta})\, P(\boldsymbol{\Theta})}{P(\hat{\mathbf{x}})P(\mathbf{r})} \rightarrow \max\,. \tag{20}$$

Here, $P(\hat{\mathbf{x}}, \mathbf{r}|\boldsymbol{\Theta})$ is the likelihood of the data $\hat{\mathbf{x}}$ and $P(\mathbf{r}|\boldsymbol{\Theta})$ the likelihood of the physics $\mathbf{r}$ for given fixed parameters $\boldsymbol{\Theta}$; $P(\boldsymbol{\Theta})$ is the prior, the probability distribution of the model parameters $\boldsymbol{\Theta}$; $P(\hat{\mathbf{x}})$ and $P(\mathbf{r})$ are the marginal likelihood or evidence; and $P(\boldsymbol{\Theta}|\hat{\mathbf{x}}, \mathbf{r})$ is the posterior, the conditional probability of the parameters $\boldsymbol{\Theta}$ for the given data $\hat{\mathbf{x}}$ and physics $\mathbf{r}$. The first likelihood function $P(\hat{\mathbf{x}}|\boldsymbol{\Theta})$ evaluates the quality of fit between the model output $\mathbf{x}(\mathbf{t})$, here the simulated number of new cases for given parameters $\boldsymbol{\Theta}$, and the observed data $\hat{\mathbf{x}}$, here the reported number of new cases at every day i. For the daily likelihood function $p_i(\hat{x}|\boldsymbol{\Theta})$, at each time point t_i, we select a normal distribution, $\mathcal{N}(\mu, \sigma)$, where the mean μ is the model result $x(t_i)$ for the given parameter set $\boldsymbol{\Theta}$ and the likelihood width, the standard deviation σ, accounts for the observation error in the data $\hat{x}$. The product $\prod_{i=0}^{\mathrm{n}}$ of all daily likelihood functions $p_i(\hat{x}|\boldsymbol{\Theta})$, in our case throughout the $\mathrm{n} = 365$ days of the year 2021, defines the overall likelihood $P(\hat{\mathbf{x}}|\boldsymbol{\Theta})$,

$$P(\hat{\mathbf{x}}|\boldsymbol{\Theta}) = \prod_{i=0}^{\mathrm{n}} p_i(\hat{x}|\boldsymbol{\Theta}) \qquad \text{with} \qquad p_i(\hat{x}|\boldsymbol{\Theta}) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\|\hat{x} - x(t_i)\|^2}{2\sigma^2}\right). \tag{21}$$

The second likelihood function $P(\mathbf{r}|\boldsymbol{\Theta})$ evaluates how accurately the model output $\mathbf{x}(\mathbf{t})$, here the simulated number of new cases for given parameters $\boldsymbol{\Theta}$, satisfies the physics equation $\mathbf{r}(\mathbf{t}) = \mathbf{0}$ at every day i. Again, we define the daily likelihood function $p_i(r|\boldsymbol{\Theta})$, at each time point t_i, through a normal distribution, $\mathcal{N}(\mu, \sigma)$, where the mean μ is the model result of the physics equation $r(x(t_i))$ for the given parameter set $\boldsymbol{\Theta}$ and the likelihood width is the standard deviation σ. The product $\prod_{i=0}^{\mathrm{n}}$ of all daily likelihood functions $p_i(r|\boldsymbol{\Theta})$, defines the overall likelihood $P(\mathbf{r}|\boldsymbol{\Theta})$,

$$P(\mathbf{r}|\boldsymbol{\Theta}) = \prod_{i=0}^{\mathrm{n}} p_i(r|\boldsymbol{\Theta}) \qquad \text{with} \qquad p_i(r|\boldsymbol{\Theta}) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\|r(t_i)\|^2}{2\sigma^2}\right). \tag{22}$$

Similar to the previous section on Neural Network modeling, the set of model parameters could consist of network parameters $\boldsymbol{\theta}$ and physics parameters $\boldsymbol{\vartheta}$,

$$\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \boldsymbol{\vartheta}\} \qquad \text{with} \qquad \boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\} \qquad \text{and} \qquad \boldsymbol{\vartheta} = \{c, k, x_0\}, \tag{23}$$

where $\mathbf{W}_k$ and $\mathbf{b}_k$ are the network weights and biases from Eq. (1) and c, k, and x_0 are the physical damping, stiffness, and offset from Eq. (3). We specify the Bayesian Inference (20) and the prior distributions of its parameters (23) for classical Bayesian Inference in Section 4.1, for Bayesian Neural Networks in Section 4.2, and for Bayesian Physics Informed Neural Networks in Section 4.3.

4.1. Bayesian Inference

The objective of a classical Bayesian Inference is to estimate the posterior parameter distributions $P(\boldsymbol{\vartheta}|\hat{\mathbf{x}})$ of a set of physics parameters $\boldsymbol{\vartheta}$ such that the statistics of a physical model $\mathbf{x}(\mathbf{t})$ agree with data $\hat{\mathbf{x}}$. For the physical model, we assume that the number of daily new cases $x(t)$ follows the physics of a damped harmonic oscillator from Section 2.3, $m\,\ddot{x} + c\,\dot{x} + k\,x = 0$, where m is the mass, c is the viscous damping coefficient, and k is the stiffness. Without loss of generality, we set $m \equiv 1$. We consider the underdamped case, for which the damping ratio is smaller than one, $c^2 < 4\,k$. For this case, following Eq. (6), the analytical solution for the number of daily new cases is

$$x(\boldsymbol{\vartheta}; t) = x_0 + A_0 \cos\left((k - \tfrac{1}{2}c^2)\,t\right) \exp(-\tfrac{1}{2}c\,t)\,. \tag{24}$$

The physics of the number of new cases are uniquely determined by three parameters, $\boldsymbol{\vartheta} = \{c, k, x_0\}$, the viscous damping c, the stiffness k, and the offset x_0. We now use Bayes' theorem, to estimate the posterior probability distribution of the physics parameters $\boldsymbol{\vartheta}$ such that the statistics of the model $\mathbf{x}(t)$ from Eq. (24) agree with the reported case data $\hat{\mathbf{x}}$ in Fig. 1,

$$P(\boldsymbol{\vartheta}|\hat{\mathbf{x}}) = \frac{P(\hat{\mathbf{x}}|\boldsymbol{\vartheta})\, P(\boldsymbol{\vartheta})}{P(\hat{\mathbf{x}})}\,. \tag{25}$$

Here $P(\hat{\mathbf{x}}|\boldsymbol{\vartheta})$ is the likelihood, the conditional probability of the data $\hat{\mathbf{x}}$ for given fixed physics parameters $\boldsymbol{\vartheta}$; $P(\boldsymbol{\vartheta})$ is the prior, the probability distribution of the physics parameters $\boldsymbol{\vartheta}$; $P(\hat{\mathbf{x}})$ is the marginal likelihood or evidence; and $P(\boldsymbol{\vartheta}|\hat{\mathbf{x}})$ is the posterior, the conditional probability of the physics parameters $\boldsymbol{\vartheta}$ for the given data $\hat{\mathbf{x}}$. For the likelihood $P(\hat{\mathbf{x}}|\boldsymbol{\vartheta})$, the product of the individual point-wise likelihoods $p_i(\hat{x}|\boldsymbol{\vartheta})$ according to Eq. (33), we select a normal distribution $\mathcal{N}(\mu,\sigma)$, where the mean μ is the model result $x(t_i)$ for the given physics parameters $\boldsymbol{\vartheta}$ and the likelihood width σ takes a half Cauchy distribution, $\sigma = \text{halfCauchy}(\beta = 1.0)$,

$$P(\hat{\mathbf{x}}|\boldsymbol{\vartheta}) = \prod_{i=0}^{\mathrm{n}} p_i(\hat{x}|\boldsymbol{\vartheta}) \qquad \text{with} \qquad p_i(\hat{x}|\boldsymbol{\vartheta}) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\|\hat{x}-x(t_i)\|^2}{2\sigma^2}\right). \tag{26}$$

For the prior probability distributions $P(\boldsymbol{\vartheta})$, we select independent weakly informed priors with log-normal distributions for the three physics parameters, $\boldsymbol{\vartheta} = \{c, k, x_0\}$, the viscous damping c, the stiffness k, and the offset x_0,

$$P(\boldsymbol{\vartheta}) = P(c,k,x_0) \qquad \text{with} \qquad \begin{array}{ll} \log(c) & \sim \mathcal{N}(\mu = \log(2.2), \quad \sigma = 0.5) \\ \log(k) & \sim \mathcal{N}(\mu = \log(350), \ \sigma = 0.5) \\ \log(x_0) & \sim \mathcal{N}(\mu = \log(0.56), \ \sigma = 0.5). \end{array} \tag{27}$$

Using the case data from Fig. 1, we approximately infer the posterior distributions from Bayes theorem (25) by employing a NO-U-Turn sampler [32] implementation of the Python package PyMC3 [33]. We use two chains with the first 2000 samples to tune the sampler and the subsequent 4000 samples to estimate the set of parameters $\boldsymbol{\vartheta}$, and employ a target acceptance rate of 0.85.

4.2. Bayesian Neural Networks

The objective of Bayesian Neural Networks is to estimate the posterior parameter distributions $P(\boldsymbol{\theta}|\hat{\mathbf{x}})$ of a set of network parameters $\boldsymbol{\theta}$, without any prior information about the underlying physics, such that the statistics of the output of the neural network $\mathbf{x}(\mathbf{t})$ agree with data $\hat{\mathbf{x}}$. For the neural network, we use the fully-connected feed-forward neural network composed of multiple hidden layers of Section 2.2 to approximate the output, the number of daily new cases $\mathbf{x}$, from the input, the time coordinates $\mathbf{t}$. We denote the hidden variables of the k_{th} hidden layer as $\mathbf{z}^k$ and summarize the neural network, here for two hidden layers, as

$$\begin{array}{lcl} \mathbf{z}_0 & = & \mathbf{t} \\ \mathbf{z}_1 & = & \sigma\,(\ \mathbf{W}_1\,\mathbf{z}_0 + \mathbf{b}_1\,) \\ \mathbf{z}_2 & = & \sigma\,(\ \mathbf{W}_2\,\mathbf{z}_1 + \mathbf{b}_2\,) \\ \mathbf{z}_3 & = & \mathbf{W}_3\,\mathbf{z}_2 + \mathbf{b}_3\,, \end{array} \tag{28}$$

where the output of the last layer approximates the true solution, $\mathbf{x} \approx \mathbf{z}_3$. Similar to the previous sections, $\sigma\,(\circ)$ denotes the non-linear activation function for which we select a hyperbolic tangent function, $\sigma\,(\circ) = \tanh\,(\circ)$, and $\mathbf{W}_k$ and $\mathbf{b}_k$ are the network weights and biases of the kth layer. We use Bayes' theorem, to estimate the posterior probability distribution of the network parameters $\boldsymbol{\theta}$ such that the statistics of the neural network $\mathbf{x} \approx \mathbf{z}_3$ from Eq. (28) agree with the reported case data $\hat{\mathbf{x}}$ in Fig. 1,

$$P(\boldsymbol{\theta}|\hat{\mathbf{x}}) = \frac{P(\hat{\mathbf{x}}|\boldsymbol{\theta})\,P(\boldsymbol{\theta})}{P(\hat{\mathbf{x}})}\,. \tag{29}$$

Here $P(\hat{\mathbf{x}}|\boldsymbol{\theta})$ is the likelihood, the conditional probability of the data $\hat{\mathbf{x}}$ for given fixed parameters $\boldsymbol{\theta}$; $P(\boldsymbol{\theta})$ is the prior, the probability distribution of the network parameters $\boldsymbol{\theta}$; $P(\hat{\mathbf{x}})$ is the marginal likelihood or evidence; and $P(\boldsymbol{\theta}|\hat{\mathbf{x}})$ is the posterior, the conditional probability of the parameters $\boldsymbol{\theta}$ for the given data $\hat{\mathbf{x}}$. For the likelihood $P(\hat{\mathbf{x}}|\boldsymbol{\theta})$, the product of the individual daily point-wise likelihoods $p_i(\hat{x}|\boldsymbol{\theta})$ according to Eq. (33), we select a normal distribution $\mathcal{N}(\mu,\sigma)$, where the mean μ is the model result $x(t_i)$ for the given network parameters $\boldsymbol{\theta}$ and the likelihood width σ is the standard deviation with $\sigma = 0.05$,

$$P(\hat{\mathbf{x}}|\boldsymbol{\theta}) = \prod_{i=0}^{\mathrm{n}} p_i(\hat{x}|\boldsymbol{\theta}) \qquad \text{with} \qquad p_i(\hat{x}|\boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\|\hat{x}-x(t_i)\|^2}{2\sigma^2}\right). \tag{30}$$

For the prior probability distributions $P(\boldsymbol{\theta})$, we select independent weakly informed priors with normal distributions with a zero-mean for the two sets of network parameters, $\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\}$, the network weights $\mathbf{W}_k$ and biases $\mathbf{b}_k$ [19],

$$P(\boldsymbol{\theta}) = P(\mathbf{W}_k, \mathbf{b}_k) \qquad \text{with} \qquad \begin{aligned} \mathbf{W}_k &\sim \mathcal{N}(\mu = 0.0,\, \sigma = 2.0) \\ \mathbf{b}_k &\sim \mathcal{N}(\mu = 0.0,\, \sigma = 2.0)\,. \end{aligned} \tag{31}$$

Using the case data from Fig. 1, we approximately infer the posterior distributions from Bayes theorem (29) by employing a Hamiltonian Monte Carlo sampling [19] using Tensorflow-Probability [34]. We use the first 3000 samples to tune the sampler and the subsequent 3000 samples to estimate the set of parameters $\boldsymbol{\theta}$.

4.3. Bayesian Physics Informed Neural Networks

The objective of Bayesian Physics Informed Neural Networks is to estimate the posterior parameter distributions $P(\boldsymbol{\Theta}|\hat{\mathbf{x}}, \mathbf{r})$ of both network and physics parameters, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \boldsymbol{\vartheta}\}$, such that the statistics of Physics Informed Neural Network $\mathbf{x}(\mathbf{t})$ agree with data $\hat{\mathbf{x}}$ and satisfy the physics equation $\mathbf{r}$. As such, the Bayesian Physics Informed Neural Network integrates the Physics Informed Neural Network from Section 3.2 into a Bayesian Inference. We use Bayes' theorem, to estimate the posterior probability distribution of the parameters such that the statistics of the neural network $\mathbf{x} \approx \mathbf{z}_3$ from Eq. (1) agree with the reported case data $\hat{\mathbf{x}}$ in Fig. 1,

$$P(\boldsymbol{\Theta}|\hat{\mathbf{x}}, \mathbf{r}) = \frac{P(\hat{\mathbf{x}}, \mathbf{r}|\boldsymbol{\Theta})\, P(\boldsymbol{\Theta})}{P(\hat{\mathbf{x}})\, P(\mathbf{r})} = \frac{P(\hat{\mathbf{x}}|\boldsymbol{\Theta})\, P(\mathbf{r}|\boldsymbol{\Theta})\, P(\boldsymbol{\theta})\, P(\boldsymbol{\vartheta})}{P(\hat{\mathbf{x}})\, P(\mathbf{r})}\,. \tag{32}$$

Here $P(\hat{\mathbf{x}}, \mathbf{r}|\boldsymbol{\Theta}) = P(\hat{\mathbf{x}}|\boldsymbol{\Theta})\, P(\mathbf{r}|\boldsymbol{\Theta})$ are the two likelihood functions, the conditional probabilities of data $\hat{\mathbf{x}}$ and physics $\mathbf{r}$ for given parameters $\boldsymbol{\Theta}$; $P(\boldsymbol{\Theta}) = P(\boldsymbol{\theta})\, P(\boldsymbol{\vartheta})$ are the priors, the probability distribution of the network and model parameters $\boldsymbol{\theta}$ and $\boldsymbol{\vartheta}$; $P(\hat{\mathbf{x}})\, P(\mathbf{r})$ is the marginal likelihood or evidence; and $P(\boldsymbol{\Theta}|\hat{\mathbf{x}}, \mathbf{r})$ is the posterior, the conditional probability of the parameters $\boldsymbol{\Theta}$ for given data $\hat{\mathbf{x}}$ and physics $\mathbf{r}$. The first likelihood function $P(\hat{\mathbf{x}}|\boldsymbol{\Theta})$ evaluates the quality of fit between the model output $\mathbf{x}(\mathbf{t})$ and the observed data $\hat{\mathbf{x}}$ at every day i as a product of the daily likelihood functions $p_i(\hat{x}|\boldsymbol{\Theta})$, for which we select a normal distribution,

$$P(\hat{\mathbf{x}}|\boldsymbol{\Theta}) = \prod_{i=0}^{\mathrm{n}} p_i(\hat{x}|\boldsymbol{\Theta}) \qquad \text{with} \qquad p_i(\hat{x}|\boldsymbol{\Theta}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|\hat{x} - x(t_i)\|^2}{2\sigma^2}\right). \tag{33}$$

The second likelihood function $P(\mathbf{r}|\boldsymbol{\Theta})$ evaluates how accurately the model output $\mathbf{x}(\mathbf{t})$ satisfies the physics equation $\mathbf{r} = \mathbf{0}$ at every day i as a product of the daily likelihood functions $p_i(r|\boldsymbol{\Theta})$, for which we also assume a normal distribution,

$$P(\mathbf{r}|\boldsymbol{\Theta}) = \prod_{i=0}^{\mathrm{n}} p_i(r|\boldsymbol{\Theta}) \qquad \text{with} \qquad p_i(r|\boldsymbol{\Theta}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|r(t_i)\|^2}{2\sigma^2}\right). \tag{34}$$

For the prior probability distributions $P(\boldsymbol{\Theta})$, we select independent weakly informed priors with log-normal distributions for the three physics parameters, $\boldsymbol{\vartheta} = \{c, k, x_0\}$, normal distributions for the network parameters, $\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\}$,

$$P(\boldsymbol{\Theta}) = P(c, k, x_0, \mathbf{W}_k, \mathbf{b}_k) \qquad \text{with} \qquad \begin{aligned} \log\,(c) &\sim \mathcal{N}(\mu = \log(2.2), \quad \sigma = 0.5) \\ \log\,(k) &\sim \mathcal{N}(\mu = \log(350), \quad \sigma = 0.5) \\ \log\,(x_0) &\sim \mathcal{N}(\mu = \log(0.56), \quad \sigma = 0.5) \\ \mathbf{W}_k &\sim \mathcal{N}(\mu = 0.0, \qquad \sigma = 2.0) \\ \mathbf{b}_k &\sim \mathcal{N}(\mu = 0.0, \qquad \sigma = 2.0)\,. \end{aligned} \tag{35}$$

Using the case data from Fig. 1, we approximately infer the posterior distributions from Bayes theorem (29) by employing a Hamiltonian Monte Carlo sampler implementation of Tensorflow-Probability [34]. For the Hamiltonian Monte Carlo sampler, we use 50 leapfrog steps with an initial time step of $\mathrm{d}t = 5 \cdot 10^{-4}$, and set the burn-in steps and total number of samples to 3000. We use the first 3000 samples to tune the sampler and the subsequent 3000 samples to estimate the set of parameters $\boldsymbol{\Theta}$.

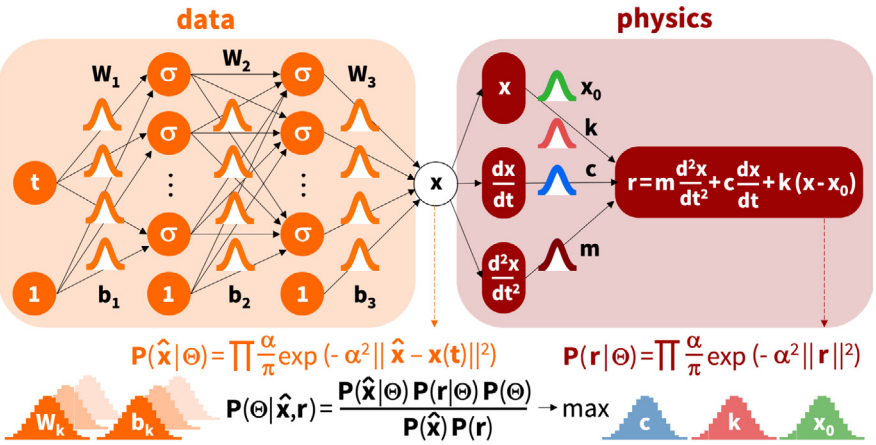

Fig. 9. Bayesian Inference modeling. Bayesian Inference maximizes a prior-weighted likelihood function $P(\boldsymbol{\Theta}|\hat{\mathbf{x}}, \mathbf{r}) \rightarrow$ max that could consist of two terms, the likelihoods $P(\hat{\mathbf{x}}|\boldsymbol{\Theta})$ and $P(\mathbf{r}|\boldsymbol{\Theta})$ of data $\hat{\mathbf{x}}$ and physics $\mathbf{r}$ for given parameters $\boldsymbol{\Theta}$, to infer distributions of the network parameters $\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\}$ and physics parameters $\boldsymbol{\vartheta} = \{c, k, x_0\}$.

4.4. Comparison of Bayesian Inference modeling

Fig. 9 illustrates the features the Bayesian Inference models of this section. The classical Bayesian Inference from Section 4.1 maximizes a prior-weighted likelihood, $P(\hat{\mathbf{x}}|\boldsymbol{\vartheta})$, in terms of the error between data and model $\| \hat{x} - x(t) \|$ to infer distributions of the physics parameters $\boldsymbol{\vartheta}$. The Bayesian Neural Networks from Section 4.2 maximize a prior-weighted likelihood, $P(\hat{\mathbf{x}}|\boldsymbol{\theta})$, in terms of the error between data and model $\| \hat{x} - x(t) \|$ to infer distributions of the network parameters $\boldsymbol{\theta}$. The Bayesian Physics Informed Neural Networks from Section 4.3 maximize a prior-weighted likelihood, $P(\hat{\mathbf{x}}, \mathbf{r}|\boldsymbol{\Theta})$, in terms of the error between data and model $\| \hat{x} - x(t) \|$, shown on the left, and the error of the physics $\| r \|$, shown on the right, to infer the distributions of both network and physics parameters $\boldsymbol{\theta}$ and $\boldsymbol{\vartheta}$.

Fig. 10 illustrates the results of the classical Bayesian Inference for a varying size of the training data set. The thin yellow lines indicate daily new case data, the thick orange lines are their seven-day average, and the blue dots are the training data varying between 150, 175, 200, 225, 250, and 275 days, from top left to bottom right. The dashed red lines represent the physics-based model and the light blue areas are its credible intervals. First and foremost, we note that all six panels display a reasonable agreement between data and model, both inside the training window and beyond. With an increasing size of training data, from top left to bottom right, the error in the descriptive window increases and the error in the predictive window decreases. For example, the first peak of 905,378 cases on day 118, April 29, 2021, is best approximated for the smallest training data set in the top left, while the second peak of 819,336 cases on day 221, August 10, 2021, is best approximated for the largest training data set in the bottom right. In agreement with our expectation, the width of the credible interval, the light blue area, decreases with an increase in training data: The more data we use to train the model, the better its fit.

Fig. 11 illustrates the posterior distributions of the physics parameters $\boldsymbol{\vartheta}$, the damping c, stiffness k, and offset x_0. The Bayesian Inferences yields a viscous damping coefficient of $c = 1.111 \pm 0.126$, a stiffness of $k = 402.4 \pm 1.7$, and an offset of $x_0 = 0.541 \pm 0.002$. This corresponds to a damping of $\delta = 0.556$, a frequency of $\omega = 20.06$, and a period of $T = 0.313$ years. These values compare well to the learned parameters of the Physics Informed Neural Network in Fig. 8, with a viscous damping coefficient of $c = 1.251$, a stiffness of $k = 374.6$, and an offset of $x_0 = 0.558$. However, the Bayesian Inference not only inferred the means of the parameters, but also their posterior distributions, where narrow distributions indicate a higher confidence in the mean values.

Fig. 12 illustrates the posterior distributions of the physics parameters $\boldsymbol{\vartheta}$, the damping c, stiffness k, and offset x_0. The Bayesian Physics Informed Neural Network yields a viscous damping coefficient of $c = 1.312 \pm 0.391$, a stiffness of $k = 319.4 \pm 38.6$, and an offset of $x_0 = 0.571 \pm 0.020$. This corresponds to a damping of $\delta = 0.65$, a frequency of $\omega = 17.87$, and a period of $T = 0.352$ years. Compared to the classical Bayesian Inference in Fig. 11, the damping is 18% larger, the stiffness is 21% smaller, and the offset is 6% larger. Notably, all three parameters

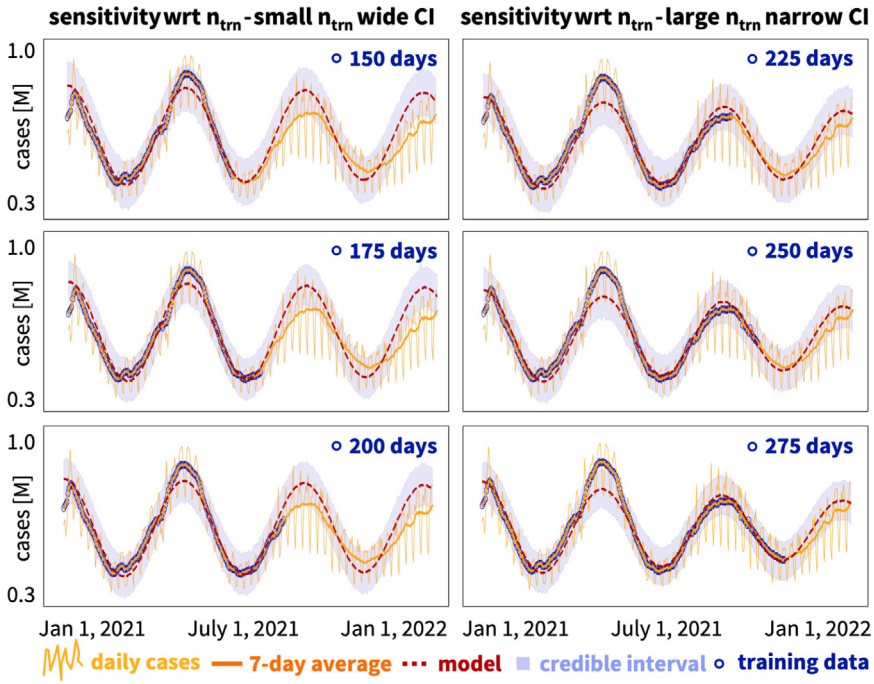

Fig. 10. Bayesian Inference. Sensitivity analysis for varying size of training data set. With increasing training data size of 150, 175, 200, 225, 250, and 275 days, the error in the descriptive window increases, the error in the predictive window decreases, and the width of the credible interval decreases (top left to bottom right). Thin yellow lines indicate daily new case data, thick orange lines are their seven-day average, dashed red lines are the physics-based model, light blue areas are its credible intervals, blue dots are the training data.

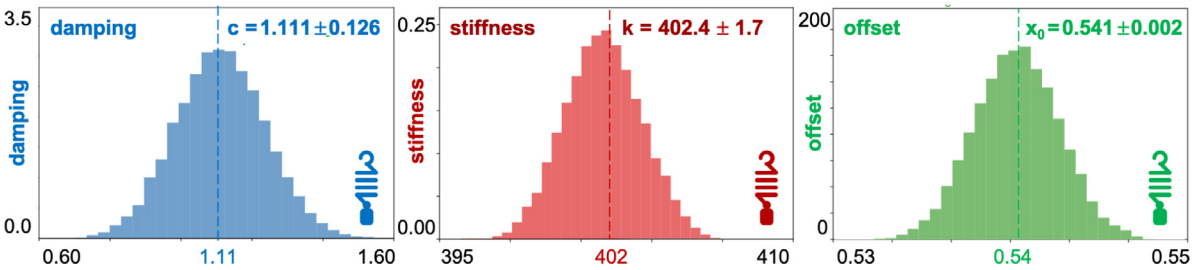

Fig. 11. Bayesian Inference. Posterior distributions of physics parameters damping c, stiffness k, and offset x_0. Inferred viscous damping coefficient of $c = 1.111 \pm 0.126$, stiffness of $k = 402.4 \pm 1.7$, and offset of $x_0 = 0.541 \pm 0.002$. This corresponds to a damping of $\delta = 0.556$, a frequency of $\omega = 20.06$, and a period of $T = 0.313$ years. Simulation with training window of n_{trn} = 225 days.

of the Bayesian Physics Informed Neural Network show wider standard deviations than in the classical Bayesian Inference, the damping by a factor 3, stiffness by a factor 23, and offset by a factor 10. These observations suggest that, for the chosen training window of n_{trn} = 225 days, the Bayesian Physics Informed Neural Network is not only more expensive than the Bayesian Inference, but also less robust.

5. Discussion

Understanding the behavior of nonlinear dynamical systems remains a challenging task. Machine learning has emerged as a powerful technology to provide insight into observational data. A vast variety of learning machines have been developed throughout the past decade, but to the unexperienced user, selecting the right model for a specific task is an overwhelming endeavor. This is amplified by the fact that tutorials or publications typically

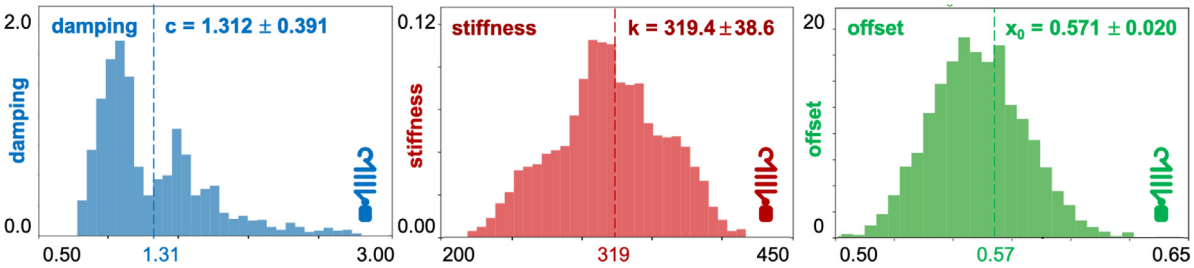

Fig. 12. Bayesian Physics Informed Neural Network. Posterior distributions of physics parameters damping c, stiffness k, and offset x_0. Inferred viscous damping coefficient of $c = 1.312 \pm 0.391$, stiffness of $k = 319.4 \pm 38.6$, and offset of $x_0 = 0.571 \pm 0.020$. This corresponds to a damping of $\delta = 0.65$, a frequency of $\omega = 17.87$, and a period of $T = 0.352$ years. Simulation with training window of $n_{\text{trn}} = 225$ days.

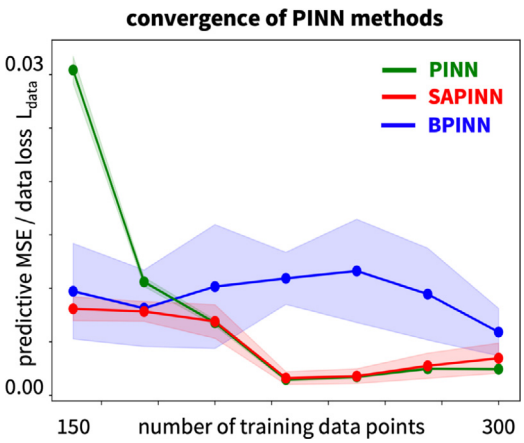

Fig. 13. Comparison of Physics Informed Neural Network models. Convergence with increasing size of training data set. Predictive data loss, L_{data}, as mean squared error in the predictive window from t^{ntrn} to t^{nsmp}, for Physics Informed Neural Network, Self Adaptive Physics Informed Neural Network, and Bayesian Physics Informed Neural Network. For all three methods, the predictive data loss decreases with an increasing number of training data points. The Physics Informed Neural Network performs poorly initially, but converges with increasing training set size. The Self Adaptive Physics Informed Neural Network performs well, even for small training sets. Both methods perform similarly with increasing training set size. The Bayesian Physics Informed Neural Network performs reasonably well for small data sets but only improves marginally with increasing training set size.

introduce new methods using artificially generated synthetic data for which the solution is already known. The objective of this study was to compare different families of models for real-world data, the reported number of COVID-19 cases throughout the year of 2021, using a simple dynamical model, a damped harmonic oscillator. We reviewed the underlying model equations, parameters, and characteristic results of two families of models, Neural Networks and Bayesian Inference, each with three members of increasing complexity.

Both Neural Networks and Bayesian Inference perform equally well on the raw daily case data with noise, reporting uncertainties, and weekday–weekend fluctuations and on their smooth seven-day average, as Fig. 3 confirms. Our learned damping, stiffness, and offset of $c = 1.251$, $k = 374.6$, and $x_0 = 0.558$ for the Physics Informed Neural Network in Section 3.2, Fig. 8, agree well with the inferred means of $c = 1.111$, $k = 402.4$, $x_0 = 0.541$ for the Bayesian Inference in Section 4.1, Fig. 11, and with the combination of both methods of $c = 1.312$, $k = 319.4$, $x_0 = 0.571$ for the Bayesian Physics Informed Neural Network in Section 4.3, Fig. 12. Our study confirms our general intuition that the quality of the fit increases with increasing size of the training data set. For the classical Bayesian Inference in Section 4.1, Fig. 10 shows that the credible intervals become progressively narrower as the size of the training data increases. Fig. 13 compares the convergence of the three Physics Informed Neural Network models of our study, the plain Physics Informed Neural Network from Section 3.2 the Self Adaptive

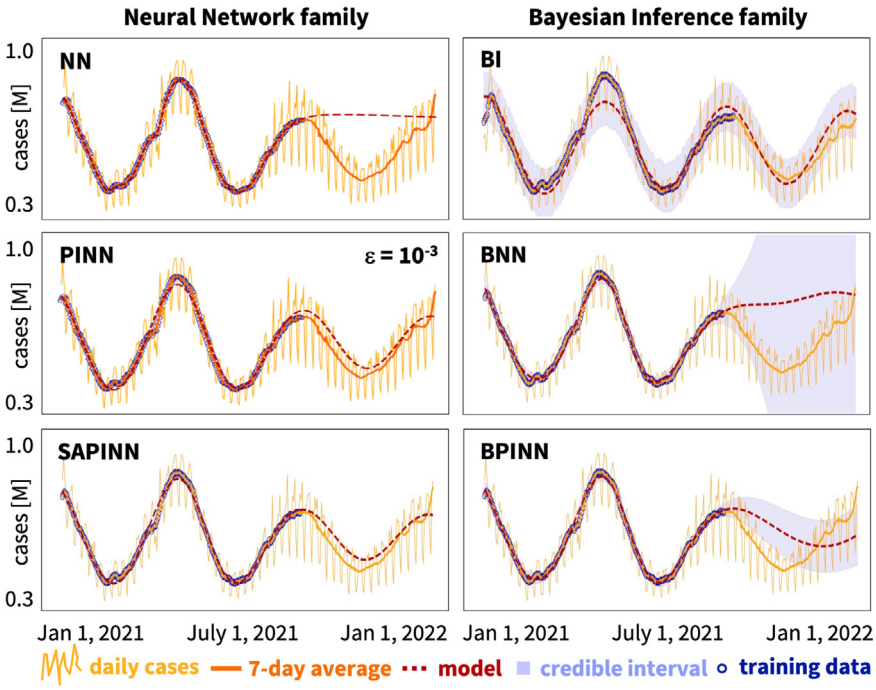

Fig. 14. Comparison of Neural Network and Bayesian Inference models. Neural Networks approximate the training data well, but fail to predict the behavior outside the training window (top left). Physics Informed Neural Networks approximate both the training data and the behavior outside the training window, but are sensitive to the weighting coefficient (middle left). Self Adaptive Physics Informed Neural Networks approximate both the training data and the behavior outside the training window, and simultaneously learn the weighting term $\varepsilon(t)$ (bottom left). Bayesian Inference not only approximates both, the training data and the behavior outside the training window, but also provides credible intervals of the model (top right). Bayesian Neural Networks approximate the training data well, but fail to predict the behavior outside the training window where they generate a wide credible intervals (middle right). Bayesian Physics Informed Neural Networks approximate the training data well, but fail to predict the behavior outside the training window (bottom right). Thin yellow lines indicate daily new case data, thick orange lines are their seven-day average, dashed red lines are the model, light blue areas are its credible intervals, blue dots are the training data.

Physics Informed Neural Network from Section 3.3, and the Bayesian Physics Informed Neural Network from Section 4.3. Clearly, for all three methods, the prediction error decreases with increasing size of the training data set. We close our study with a direct side-by-side comparison of all six methods from Sections 3 and 4 in Fig. 14. Fig. 15 provides a final recommendation to improve the performance of the Bayesian Physics Informed Neural Networks by scaling the physics equation. Table 1 summarizes the results of Figs. 14 and 15 and discusses the advantages and disadvantages of each method.

Neural Networks are a simple and robust tool to fit training data, but have poor predictive potential. Classical Neural Networks have remarkable power to fit a network model to data without any underlying physics. By minimizing a loss function that characterizes the error between model and data, the neural network learns the network parameters $\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\}$, the network weights and biases. For our example of a feed-forward network with one input, two hidden layers with 32 nodes each, and one output, it has 32+32×32+32=1088 weights and 32+32+1=65 biases, resulting in a total of 1153 unknowns. Fig. 14, top left, suggests that even in the absence of any prior physical knowledge, the classical Neural Network provides an excellent approximation of the training data. However, it fails to predict the behavior outside the training window, where it simply continues the linear horizontal trend from the last set of training data points. This becomes particularly clear when the underlying problem is non-monotonic, or, like in our case, even oscillatory. Since classical Neural Networks provide no credible intervals, we have no way of knowing, how poor the predictive potential of the network really is.

Physics Informed Neural Networks integrate data and physics and have a good predictive potential. Physics Informed Neural Networks fit a network model to data, and, at the same time, to a physics-based

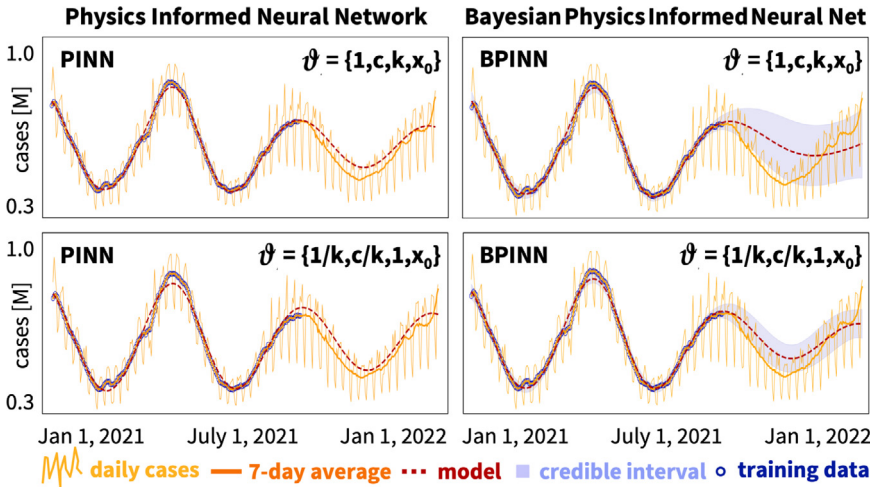

Fig. 15. Comparison of classical and Bayesian Physics Informed Neural Network models for different scaling. Both types of models are sensitive to the scaling of the physics equation, $\ddot{x} + c\,\dot{x} + k(x - x_0)$ (top row) and $\ddot{x}/k + c\,\dot{x}/k + (x - x_0)$ (bottom row). Appropriate scaling improves model performance: The fit between the red dashed lines of the model $x(t)$ and the thick orange lines of the data $\hat{x}$ increases and the light blue credible intervals decrease. Thin yellow lines indicate daily new case data, thick orange lines are their seven-day average, dashed red lines are the model, light blue areas are its credible intervals, blue dots are the training data.

Table 1
Comparison of Neural Network and Bayesian Inference models. Advantages and disadvantages of Neural Networks, Physics Informed Neural Networks, Self Adaptive Physics Informed Neural Networks, Bayesian Inference, Bayesian Neural Networks, and Bayesian Physics Informed Neural Networks in terms of parameters, data fit, predictive potential, and robustness.

Neural Network family	Bayesian Inference family
NN	**BI**
Fits Network without physics, no credible intervals Learns $\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\}$ Good fit of training data, poor predictive potential Simple, robust, and straightforward	Fits Physics model, provides credible intervals Learns distributions of $\boldsymbol{\vartheta} = \{c, k, x_0\}$ Decent fit of training data, good predictive potential Simple, robust, and straightforward
PINN	**BNN**
Fits network and physics model, no credible intervals Learns $\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\}$ and $\boldsymbol{\vartheta} = \{c, k, x_0\}$ Good fit of training data, good predictive potential Sensitive to choice of weighting coefficient ε	Fits network without physics,provides credible intervals Learns distributions of $\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\}$ Good fit of training data, poor predictive potential Credible intervals provide insight into reliable regimes
SAPINN	**BPINN**
Adaptively fits network and physics,no credible intervals Learns $\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\}$ and $\boldsymbol{\vartheta} = \{c, k, x_0\}$ and $\varepsilon(t)$ Good fit of training data, good predictive potential Robust even for small training data set	Fits network and physics, provides credible intervals Learns distributions of $\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\}$ and $\boldsymbol{\vartheta} = \{c, k, x_0\}$ Good fit of training data, moderate predictive potential Most expensive method, requires large training data set

model. If we know the underlying physics, they are an effective tool to constrain a deep learning method to a lower-dimensional manifold and create models that can be trained effectively with a small amount of data. Physics Informed Neural Networks enforce physics via soft penalty constraints. By minimizing a loss function that characterizes the error between model and data and the error in satisfying the physics, they simultaneously learn the network parameters $\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\}$ and the physics parameters $\boldsymbol{\vartheta} = \{c, k, x_0\}$. For our example, the network has 1088 weights and 65 biases, and the physics model has a damping, stiffness, and offset parameter, resulting in a total of 1156 unknowns. Fig. 14, middle left, suggests that the Physics Informed Neural Network approximates both the training data and the behavior outside the training window reasonably well and is equally capable of both interpolation and extrapolation. From Fig. 13 we conclude that Physics Informed Neural Networks

require a relatively large set of training data. For our example, they perform moderately initially, but converge well with increasing training set size. However, from Figs. 6 and 7, we conclude that Physics Informed Neural Networks are sensitive to the weighting coefficient ε that can bias the solution to emphasize on either data or physics. From Fig. 15, we add that they are also sensitive to a scaling of the physics parameters $\boldsymbol{\vartheta}$, which, if done appropriately, can improve the performance of the model. An inherent limitation of plain Physics Informed Neural Networks is that they are not equipped with built-in uncertainty quantification which may restrict their applications, especially in situations with noisy data.

Self Adaptive Physics Informed Neural Networks provide adaptive and robust fits of data and physics. Self Adaptive Physics Informed Neural Networks adaptively fit a network model to data, and, at the same time, to a physics-based model. They inherit all the advantages of Physics Informed Neural Networks and address the limitation of bias between data and physics by introducing the weighting coefficient as independent time-varying unknown. This allows them to perform well, even in regions with steep gradients, while using a smaller number of training epochs. Self Adaptive Physics Informed Neural Networks minimize a loss function that characterizes the error between model and data and between model and physics, and learn the network parameters $\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\}$, the physics parameters $\boldsymbol{\vartheta} = \{c, k, x_0\}$, and the weighting term $\varepsilon(t)$ between network and physics. For our example, the network model has 1088 weights and 65 biases, the physics model has a damping, stiffness, and offset, and the self adaptive model has a weighting coefficient, resulting in a total of 1157 unknowns. Fig. 14, bottom left, confirms that the Self Adaptive Physics Informed Neural Network approximates both the training data and the behavior outside the training window, similar to the regular Physics Informed Neural Network. With only one additional time-varying parameter, the self-adaptively learned weighting term, it performs well even for small training data sets, as Fig. 13 confirms. One caveat is that Self Adaptive Physics Informed Neural Networks involve training with complex loss functions that consist of multiple terms and result in a highly non-convex optimization problem. During training, these terms compete with one another which implies that the training process may not be robust and stable, and does not always converge to a global minimum. Self Adaptive Physics Informed Neural Networks perform best when the equations are well scaled and pre-conditioned and their parameters all lie within the same range of magnitude. While we have not explicitly shown the effect of different pre-conditioning and scaling techniques here, we have observed that the method is sensitive to scaling and could fail to converge when scaled inappropriately.

Bayesian Inference fits a physics based model to data and provides credible intervals. Classical Bayesian Inference is a simple, robust, and stable method that fits a physics based model to data. This implies that we know the underlying physics. In our example, we do not know the exact physics; however, we hypothesized that the data, the daily new cases of COVID-19 worldwide, follow the dynamics of a damped harmonic oscillator. Bayesian Inference then uses Bayes' theorem to infer the distribution of a set of model parameters, $\boldsymbol{\vartheta} = \{c, k, x_0\}$, that best explain the data. For our example, we have assumed log normal distributions for each parameter, with individual means μ and standard deviations σ, introducing a total of $3 \times 2 = 6$ unknowns. Fig. 14, top right, demonstrates that, provided the physics-based model is reasonable, the Bayesian Inference approximates the behavior well throughout the entire time window. Instead of only inferring a point value for each parameter, Bayesian Inference infers parameter distributions and credible intervals that provide valuable insight into the goodness of the fit as we have seen in the sensitivity analysis for varying training data set sizes in Fig. 10: The narrower the credible interval, the better the fit.

Bayesian Neural Networks fit a network model to data and provide credible intervals. Bayesian Neural Networks fit a network model to data without any underlying physics. They use Bayes' theorem to infer the distribution of a set of network parameters, $\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\}$ that best explain the data. For our example, a feed-forward network with one input, two hidden layers with 32 nodes each, and one output, the network has 1088 weights and 65 biases. We have assumed log-normal distributions for each parameter, with individual means μ and standard deviations σ, introducing a total of $1053 \times 2 = 2306$ unknowns. Fig. 14, middle right, shows that the Bayesian Neural Network shares the features of classical Neural Networks and Bayesian Inference. It approximates the training data well, but fails to predict the behavior outside the training window. The wide credible intervals highlight the poor predictive potential of the model and confirm that any prediction a few days beyond the training window generates unreliable results.

Bayesian Physics Informed Neural Networks provide a good fit and prediction, but are sensitive to scaling. Bayesian Physics Informed Neural Networks fit both network and physics, and, in addition, provide

credible intervals that provide insight into the quality of the fit. They use Bayes' theorem to learn the distributions of both network parameters, $\boldsymbol{\theta} = \{\mathbf{W}_k, \mathbf{b}_k\}$ and physics parameters, $\boldsymbol{\vartheta} = \{c, k, x_0\}$, with means μ and standard deviations σ each, resulting in a total of 2312 unknowns. Fig. 14, bottom right, suggests that Bayesian Physics Informed Neural Networks provide a good fit of the training data and have a reasonable predictive potential outside the training window. The narrower credible intervals compared to the Bayesian Neural Network indicate that including physics increases the performance, especially in the prediction window. Bayesian Physics Informed Neural Networks are a powerful tool when used appropriately. They combine the advantages of all methods, and perform best, albeit at a larger computational cost. Similar to plain Physics Informed Neural Networks, their true performance is sensitive to an appropriate scaling and pre-conditioning and to the size of the training data set, as we can conclude from their convergence curve in Fig. 13. However, when scaled appropriately, in our example such that the parameter k associated with the unknown $x(t)$ becomes unity, their performance increases drastically, as we conclude from the narrow credible intervals of the scaling study in Fig. 15.

Taken together, our study has shown that by embedding physical principles into a neural network architecture, we can generate effective models that train well, even with a small amount of data. Physics Informed Neural Networks seamlessly integrate data and physics, robustly solve forward and inverse problems, and perform well for both interpolation and extrapolation. At only minor additional cost, they can self-adaptively learn the weighting between data and physics and smoothly integrate real-world data and physics-based modeling. Combined with Bayesian statistics, Physics Informed Neural Networks can serve as a prior in a general Bayesian Inference, and can generate estimators for posterior distributions that provide valuable insight into uncertainty quantification. Here we have only demonstrated these features for the simple model problem of a seasonal endemic infectious disease, but it is easy to see how the underlying concepts and trends would generalize to more complex disease conditions and, more broadly, to a wide variety of nonlinear dynamical systems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Our source code and examples are available at https://github.com/LivingMatterLab/xPINNs.

Acknowledgments

This work was supported by a DAAD Fellowship, Germany to Kevin Linka, by the MURI/ARO, United States grant W911NF-15-1-0562 to George Karniadakis, and by the Stanford School of Engineering COVID-19 Research and Assistance Fund, United States and Stanford Bio-X IIP seed, United States grant to Ellen Kuhl.

References

[1] G.C.Y. Peng, M. Alber, A. Buganza Tepole, W. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G.E. Karniadakis, W.W. Lytton, P. Perdikaris, L. Petzold, E. Kuhl, Multiscale modeling meets machine learning: What can we learn? Arch. Comput. Methods Eng. 28 (2021) 1017–1037.

[2] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nat. Rev. Phys. 3 (2021) 422–440.

[3] M. Alber, A. Buganza Tepole, W. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G.E. Karniadakis, W.W. Lytton, P. Perdikaris, L. Petzold, E. Kuhl, Integrating machine learning and multiscale modeling: Perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences, Npj Digit. Med. 2 (2019) 115.

[4] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics informed deep learning: Data-driven solutions of nonlinear partial differential equations, 2017, arXiv preprint arXiv:1711.10561.

[5] M. Raissi, P. Perdikaris, G.E. Karniadakis, Multistep neural networks for data-driven discovery of nonlinear dynamical systems, 2018, arXiv preprint arXiv:1801.01236.

[6] Y. Yang, P. Perdikaris, Adversarial uncertainty quantification in physics-informed neural networks, J. Comput. Phys. 394 (2019) 136–152.

[7] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.

[8] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics informed learning machine, 2021, US Patent 10, 963, 540.

[9] M. Cai, G.E. Karniadakis, C. Li, Fractional SEIR model and data-driven predictions of COVID-19 dynamics of omicron variant, 2022, http://dx.doi.org/10.48550/arXiv.2205.11379.
[10] S. Treibert, M. Ehrhardt, An unsupervised physics-informed neural network to model COVID-19 infection and hospitalization scenarios, 2021, Bergische Universität Wuppertal, Preprint BUW-IAMCM 21/35.
[11] E. Kharazmi, M. Cai, X. Zheng, Z. Zhang, G. Lin, G.E. Karniadakis, Identifiability and predictability of integer-and fractional-order epidemiological models using physics-informed neural networks, Nat. Comput. Sci. 1 (2021) 744–753.
[12] S. Zhang, J. Ponce, Z. Zhang, G. Lin, G. Karniadakis, An integrated framework for building trustworthy data-driven epidemiological models: Application to the COVID-19 outbreak in New York City, PLoS Comput. Biol. 17 (2021) e1009334.
[13] M.A. Bhouri, F. Sahli Costabal, H. Wang, K. Linka, M. Peirlinck, E. Kuhl, P. Perdikaris, COVID-19 dynamics across the US: A deep learning study of human mobility and social behavior, Comput. Methods Appl. Mech. Engrg. 382 (2021) 113891.
[14] K. Linka, A. Goriely, E. Kuhl, Global and local mobility as a barometer for COVID-19 dynamics, Biomech. Model. Mechanobiol. 20 (2021) 651–669.
[15] I. Holmdahl, C. Buckee, Wrong but useful–what Covid-19 epidemiologic models can and cannot tell us, N. Engl. J. Med. 383 (2020) 303–305.
[16] X. Meng, H. Babaee, G.E. GE Karniadakis, Multi-fidelity Bayesian neural networks: Algorithms and applications, J. Comput. Phys. 438 (2021) 110361.
[17] J.T. Oden, I. Babuska, D. Faghihi, Predictive computational science: Computer predictions in the presence of uncertainty, in: Encyclopedia of Computational Mechanics, John Wiley & Sons, 2017.
[18] J.T. Oden, R. Moser, O. Ghattas, Computer predictions with quantified uncertainty, SIAM News 43 (9) (2010).
[19] L. Yang, X. Meng, G.E. Karniadakis, B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data, J. Comput. Phys. 425 (2021) 109913.
[20] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vektari, D.B. Rubin, Bayesian Data Analysis, third ed., Chapman and Hall/CRC, 2013.
[21] J.T. Oden, Adaptive multiscale predictive modelling, Acta Numer. 2018 (2018) 353–450.
[22] M. Osvaldo, Bayesian Analysis with Python: Introduction to Statistical Modeling and Probabilistic Programming using PyMC3 and ArviZ, second ed., Packt Publishing, 2018.
[23] X. Meng, L. Yang, Z. Mao, J. del Aguila Ferrandis, G.E. Karniadakis, Learning functional priors and posteriors from data and physics, J. Comput. Phys. 457 (2022) 111073.
[24] K. Linka, M. Peirlinck, E. Kuhl, The reproduction number of COVID-19 and its correlation with public heath interventions, Comput. Mech. 66 (2020) 1035–1050.
[25] Johns Hopkins University, Coronavirus COVID-19 global cases by the center for systems science and engineering, 2021, https://Coronavirus.Jhu.Edu/Map.Html, https://Github.Com/CSSEGISandData/Covid-19. (Assessed 1 May 2022).
[26] P.K. Jha, L. Cao, J.T. Oden, Bayesian-based predictions of COVID-19 evolution in Texas using multispecies mixture-theoretic continuum models, Comput. Mech. 66 (2020) 1055–1068.
[27] M. Peirlinck, K. Linka, F. Sahli Costabal, E. Kuhl, Outbreak dynamics of COVID-19 in China and the United States, Biomech. Model. Mechanobiol. 19 (2020) 2179–2193.
[28] K. Linka, M. Peirlinck, F. Sahli Costabal, E. Kuhl, Outbreak dynamics of COVID-19 in Europe and the effect of travel restrictions, Comput. Methods Biomech. Biomed. Eng. 23 (2020) 710–717.
[29] E. Kuhl, Computational Epidemiology. Data-Driven Modeling of COVID-19, Springer Nature, Switzerland, 2021.
[30] L. McClenny, U. Braga-Neto, Self-adaptive physics informed neural networks using a soft attention mechanism, 2020, http://dx.doi.org/10.48550/arXiv.2009.04544.
[31] T. Bayes, R. Price, An essay towards solving a problem in the doctrine of chances. By the late rev. Mr. Bayes, F.R.S. communicated by Mr. Price, in a letter to John Canton, A.M.F.R.S, Philos. Trans. R. Soc. Lond. 53 (1763) 370–418.
[32] M.D. Hoffman, A. Gelman, The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo, J. Mach. Learn. Res. 15 (2014) 1593–1623.
[33] J. Salvatier, T.V. Wiecki, C. Fonnesbeck, Probabilistic programming in Python using PyMC3, PeerJ Comput. Sci. 2 (2016) e55.
[34] Tensorflow I/O, version 0.15.0, 2020, https://pypi.org/project/tensorflow-io/0.15.0 (Assessed 1 May 2022).