Nonlinear Dyn (2023) 111:1987–2007
https://doi.org/10.1007/s11071-022-08182-z

ORIGINAL PAPER

Data-driven identification of partial differential equations for multi-physics systems using stochastic optimization

Jinwoo Im · Felipe P. J. de Barros · Sami F. Masri

Received: 13 July 2022 / Accepted: 12 December 2022 / Published online: 21 December 2022

Abstract We develop a stochastic optimization framework to identify governing equations in multi-physics systems. The proposed approach discovers partial differential equations (PDEs) by combining user's prior knowledge of the underlying physics of a target system and its observed data. The technique relies on evolutionary processes to randomly generate PDEs and stochastically optimize their structure and coefficients to the data by exploring the infinite model space. Furthermore, the method captures the spatiotemporal dynamics of physical system by direct evaluation of the candidate PDEs under physical constraints. To achieve significant computational speedup, the proposed stochastic optimization method relies on a series of novel modifications. These consist of the incorporation of a multi-purpose loss function into the parallel fitness test and bloat control techniques into the evolutionary processes. As such, these innovations lead to a significant improvement of the effectiveness and computational efficiency in identifying PDEs from given data. We demonstrate the applicability of methodology in two illustrative examples: the nonlinear Burgers' equation and the linear/nonlinear advection-dispersion equation. The exact PDEs are successfully identified, even with significant data noise, and captures the underlying physics of the target system. Through series of simulations, we assess the accuracy, robustness, and limitations of the proposed approach. In particular, we show the impact of key dimensionless groups (that accounts for the competition between various physical phenomena) in controlling the accuracy of the identification process. This work shows that developed identification method is a promising effective and robust gray box modeling tool for identifying PDEs.

Keywords System identification · Multi-physics · Stochastic optimization · Genetic programming · Governing equations · Mathematical-Physics

J. Im · F. P. J. de Barros (✉)· S. F. Masri
Sonny Astani Department of Civil and Environmental Engineering, University of Southern California, 3620 South Vermont Ave., Los Angeles, CA 90089, USA
e-mail: fbarros@usc.edu

J. Im
e-mail: jinwooim@usc.edu

S. F. Masri
e-mail: masri@usc.edu

1 Introduction

System identification methods aim to represent a physical system of concern with a mathematical model on the basis of its input/output data [5,17,28]. It plays a crucial role in predicting the systems' behavior and enhancing user's comprehension of underlying physics. Recent advances of sensing technology and computing power enable system identification methods to capture more complex multi-physics systems [21]. By virtue of their effectiveness and versatile applicability, state-of-the-art identification approaches are actively being developed [16].

Modeling approaches can be divided into three broad categories: white, gray, and black box modeling [21]. Firstly, the "white box" modeling utilizes underlying physics of systems to establish models. Despite its high accuracy of prediction, this type of modeling has limitations in term of solely relying on the user's understanding of the underlying system. Next, the "black box" modeling is purely driven by data. By employing distinctive mathematical forms (e.g., neural networks [4], Gaussian processes [13], and more examples in [1]), this modeling approach is effectively able to predict system responses without any prior assumptions about the underlying physics. A series of examples related to the discovery of physical relationships from experimental data can be found in the literature, see [28]. However, since the black box modeling is purely data-driven, it is neither able to reveal the physics behind systems nor provide reliable estimates beyond the range of training data. Lastly, the "gray box" modeling attempts to leverage both user's prior physical knowledge about systems and data to build models. This mathematical modeling approach provides a way to overcome the limitations of the white and black box modeling; the prior knowledge of systems needed to develop models is minimized, and the models embedded with system dynamics increase reliability of extrapolation. Given these advantages, many gray box approaches are being proposed. These consist of state-space approaches [6,19], sparse approximation [5,15,25,26], Gaussian processes [22], and neural networks [7,8,12,18,23,27,30]. One noticeable example is SINDy [26] that utilizes the sparse approximation to discover governing equations from observed data. Although this method shows its capability in finding many canonical models, it is limited to simple equation forms constructed by a linear combination of the candidate terms.

Genetic programming (GP) is one of the available gray box tools for modeling physical systems. Inspired by Darwin's theory of evolution, GP exercises evolutionary processes (e.g., crossover and mutation) to optimize programs in a stochastic manner for a specific task [14]. Programs of GP are designed to perform a specific task and to be decomposed into genetic information [24]: a vector of parameters for function optimization, an expression tree for symbolic regression, or a decision tree for decision making. In the GP process, a population of programs is randomly initialized as the first generation, and their fitness is tested according to the performance at a certain task. Stochastic evolutionary processes are then applied to the programs in accordance with their fitness in consecutive generations. As a result, the programs become optimized to perform the task, with simple and complex forms.

GP-based identification approaches have been considered effective in performing symbolic regressions, ordinary differential equations (ODEs) and reduced-order models for nonlinear systems in mechanics (see [21] and references therein). In the context of applied mechanics, a few works [2,3] have improved GP-based identification methods by expanding the component library for expression trees (e.g., derivative variables and discontinuous functions). These developments offer insight for system dynamics and are fairly accurate in extrapolation. Despite this progress, current GP applications are still limited due to two main issues. Firstly, GP-based methods have a large amount of hyperparameters that need to be tuned, which makes the generalization of the methods for a set of identification problems difficult. Secondly, an attempt to generalize the method for a variety of identification problems has a potential to delay convergence time considerably. These shortcomings were addressed by [11] in the context of mechanical and environmental systems described by non-linear ODEs.

Given that many multi-physics systems are governed by partial differential equations (PDEs), there is need to further expand GP-based approaches to address physical systems that vary in space and time. Our work aims to fill this gap by developing a generalized GP-based stochastic optimization method to identify PDEs. The novel features of the proposed framework are as follows: (1) the incorporation of a multi-purpose loss function and stochastic sampling in the parallel fitness test and (2) the inclusion of bloat control techniques within the evolutionary processes. As it will be shown, these significant modifications prevent potential drawbacks of GP (i.e., issues of many hyperparameters and slow convergence rate) and lead to computationally efficient and highly reproducible identification results. We demonstrate the performance of the methodology in two canonical PDEs: Burgers' equation (nonlinear PDE) and the advection-dispersion (or convection-diffusion) equation (linear/nonlinear PDE). The accuracy and robustness of the identification procedure are investigated under different noise levels and system characteristics. Since the method provides a variety of candidate PDEs for a target system in the final result, the identified PDEs serve as possible descriptions of complex system dynamics to improve user's understanding, while the best one is used as the system model to predict (or extrapolate) future responses.

This paper is organized as follows. First, the general problem statement is given in Sect. 2. Details pertaining to the novel identification method are presented in Sect. 3. This section will provide a road map for users who wish to tailor this approach for their own application. Illustrative examples are given in Sect. 4, and their discussion is followed in Sect. 5. Lastly, the implications of this study and future research are described in Sect. 6.

2 Problem statement

A multi-physics system is considered under limited prior knowledge. A multi-dimensional quantity of interest in the system is denoted by $\mathbf{r}(\mathbf{x}, t)$ where $\mathbf{x}$ and t correspond to the d-dimensional Cartesian space and time. Such a quantity can be the speed of a wave in shallow water, solute concentration in an environmental medium, the temperature distribution in a heterogeneous material, or atomic dispersion in a metallic glass (see more examples in Fig. 1a). For the purpose of this work, we consider a one-dimensional quantity, $r(x, t)$, in a one-dimensional spatial domain x. We represent an external excitation by $f(x, t)$ which will impact the spatiotemporal dynamics of $r(x, t)$ (see Fig. 1b). We assume that the measurements are available at discrete points of space and time domains, $f(x_i, t_j)$ and $r(x_i, t_j)$, $(i = 1, 2, \ldots, n_x$ and $j = 1, 2, \ldots, n_t$ where n_x and n_t are the numbers of data points for space and time domains, respectively).

On the basis of the measurements, the system model, namely $\mathcal{M}$, can be established to describe the system dynamics of $r(x_i, t_j)$ subject to $f(x_i, t_j)$, in the form of a PDE (see Fig. 1b–d). Since the model response is an approximation of the (mathematical) system response under the same excitation $f(x_i, t_j)$, it is denoted by $\hat{r}(x_i, t_j)$ in the system model:

$$\mathcal{M}\left(\hat{r}, \frac{\partial \hat{r}}{\partial t}, \frac{\partial \hat{r}}{\partial x}, \frac{\partial^2 \hat{r}}{\partial x^2}, \ldots; f(x_i, t_j)\right) = 0. \tag{1}$$

The main goal of this work is to identify the system model $\mathcal{M}$, in the infinite function space $\mathbb{M}$, which minimizes the error norm (i.e., ϵ_{res}) of a suitable model fidelity model, e.g., the normalized root-mean-square error between the values of the system response measurements and the corresponding model response, namely $r(x_i, t_j)$ and $\hat{r}(x_i, t_j)$:

$$\min_{\mathcal{M}} \left\{ \epsilon_{\text{res}} \middle| \mathcal{M}\left(\hat{r}, \frac{\partial \hat{r}}{\partial t}, \frac{\partial \hat{r}}{\partial x}, \frac{\partial^2 \hat{r}}{\partial x^2}, \ldots; f(x_i, t_j)\right) = 0 \right\}$$
$$\text{with } \mathcal{M} \in \mathbb{M} \tag{2}$$

where ϵ_{res} is defined as:

$$\epsilon_{\text{res}} = \sqrt{\frac{\sum_i \sum_j \left[\hat{r}(x_i, t_j) - r(x_i, t_j)\right]^2}{\sum_i \sum_j \left[r(x_i, t_j)\right]^2}}. \tag{3}$$

Thus, the normalized error is the ratio of the Euclidean L^2-norm of the deviation between the reference and estimated responses, $\left\|\hat{r}(x_i, t_j) - r(x_i, t_j)\right\|_2$, divided by the norm of the ensemble of measurements, $\left\|r(x_i, t_j)\right\|_2$.

In order to "discover" a PDE, as the system model $\mathcal{M}$, from a data set of $f(x_i, t_j)$ and $r(x_i, t_j)$ under limited prior knowledge of the system, a general framework of system identification is established by utilizing stochastic optimization (e.g., GP). The proposed methodology is implemented by modifying the GP algorithm reported in [11]. Details follow in Sect. 3.

3 Methodology

3.1 Genetic programming for system identification

A general system identification approach utilizing GP is developed to discover a PDE from a given data set as a model for a multi-physics system. Details of the GP method are found in [14]. Its code package, named Genetic Programming for System Identification (GPSI), is written in Python (version 3.7.4) with the *SymPy* library (version 1.7.1) and is available in Github (https://github.com/Jinwoousc/GPSI). In the following, we provide a series of steps along with a brief synopsis of the methodology employed in GPSI, in line with Fig. 2:

1. Determine a quantity of interest (i.e., $r(x, t)$) in a target multi-physics system, and measure (or sample) both the excitation $f(x_i, t_j)$ $(i = 1, 2, \ldots, n_x$ and $j = 1, 2, \ldots, n_t)$ and the corresponding system response $r(x_i, t_j)$.
2. According to the user's prior knowledge of the physical system, select mathematical components, e.g., basis functions such as the Heaviside step function

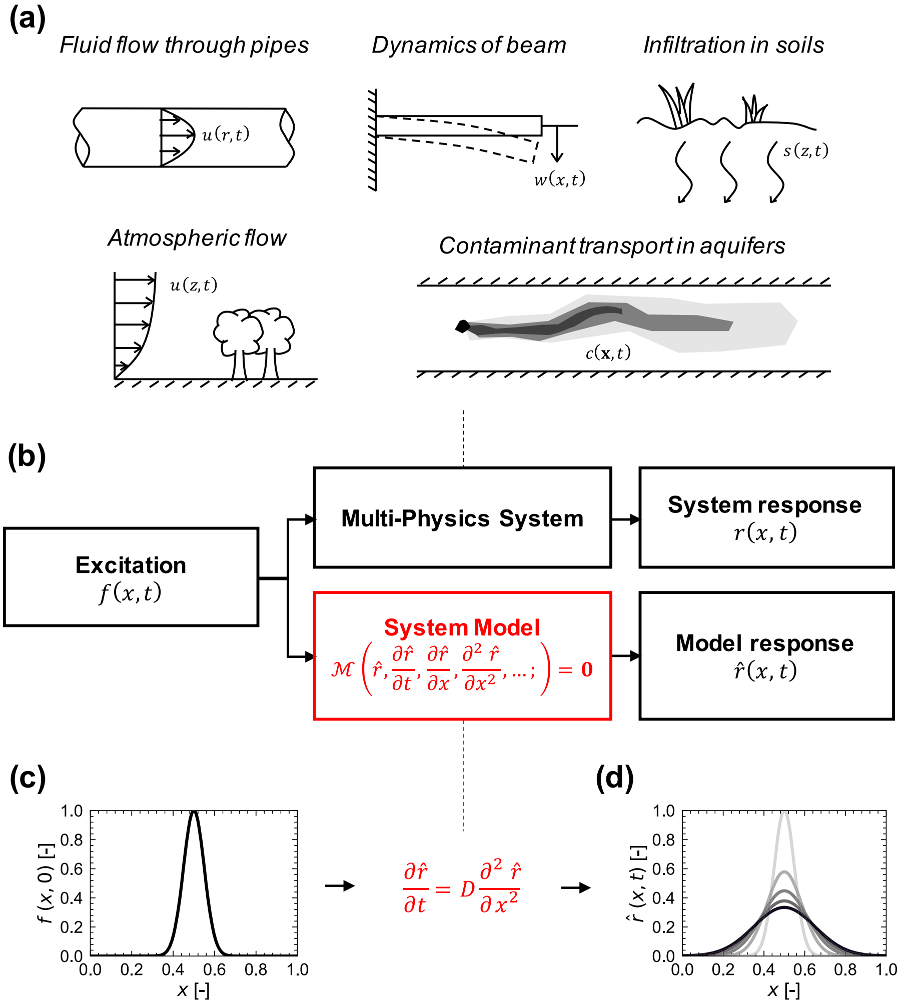

Fig. 1 Schematic illustration of the general modeling problem; **a** multi-physics systems encountered in various science and engineering domains; **b** modeling approach representing system dynamics with a partial differential equation (PDE); **c** external excitation utilized as an initial condition of a PDE; and **d** evaluation of a quantity of interest over time by numerically solving the PDE

and error function (see the mathematical expression library box in Fig. 2), which will postulate the expression "library" to be used for candidate PDEs $\mathcal{M}_k$ ($k = 1, 2, \ldots, n_M$ where n_M is the number of the PDEs).

3. Prepare the GP simulation sets by selecting the loss function components (i.e., the model response error, the equation residual error, or the model complexity penalty) for the fitness test depending on the data quality of $f(x_i, t_j)$ and $r(x_i, t_j)$ (refer to Sect. 3.2 for details) and by arranging the rest of the hyper-parameters of GPSI simulation, mainly focusing on the complex penalty coefficient.
4. Randomly generate initial PDEs by employing the form of a binary expression tree (see the example of a candidate PDE in Fig. 2).
5. Run the GPSI simulation that repeats the fitness test and the evolutionary processes (see the corresponding boxes in Fig. 2) to update the population of PDEs in consecutive generations. Details related to the evolutionary processes are described in Sect. 3.3.
6. Collect the PDEs from the last population with which the stop criterion of the simulation is satisfied, and determine the system model $\mathcal{M}$ among the identified PDEs, by considering their ϵ_{res} (see Eq. 3), model simplicity, and interpretation revealing the system dynamics.
7. Test other simulation sets with different settings from Step 2 on the basis of the updated prior knowledge of the target multi-physics system.

The proposed approach firstly arranges the mathematical components for candidate PDEs and the other

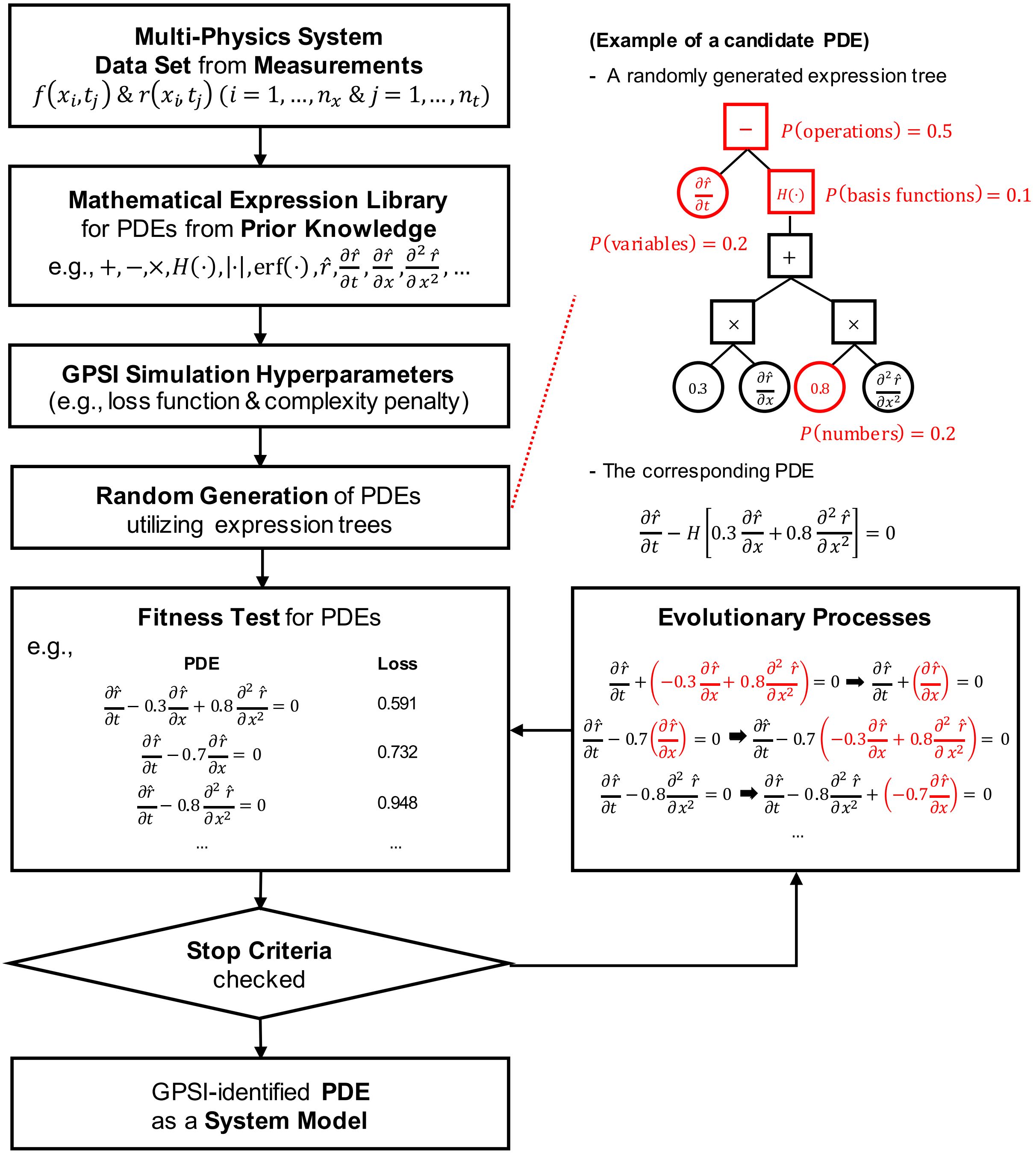
Multi-Physics System
Data Set from Measurements
$f(x_i,t_j)$ & $r(x_i,t_j)$ $(i=1,\ldots,n_x$ & $j=1,\ldots,n_t)$
Mathematical Expression Library
for PDEs from Prior Knowledge
e.g., $+,-,\times,H(\cdot),|\cdot|,\mathrm{erf}(\cdot),\hat{r},\frac{\partial\hat{r}}{\partial t},\frac{\partial\hat{r}}{\partial x},\frac{\partial^2\hat{r}}{\partial x^2},\ldots$
GPSI Simulation Hyperparameters
(e.g., loss function & complexity penalty)
Random Generation of PDEs
utilizing expression trees
Fitness Test for PDEs
e.g.,
PDE
Loss
$\frac{\partial\hat{r}}{\partial t}-0.3\frac{\partial\hat{r}}{\partial x}+0.8\frac{\partial^2\hat{r}}{\partial x^2}=0$
0.591
$\frac{\partial\hat{r}}{\partial t}-0.7\frac{\partial\hat{r}}{\partial x}=0$
0.732
$\frac{\partial\hat{r}}{\partial t}-0.8\frac{\partial^2\hat{r}}{\partial x^2}=0$
0.948
...
...
Stop Criteria
checked
GPSI-identified PDE
as a System Model
(Example of a candidate PDE)
- A randomly generated expression tree
$-$
$P(\text{operations})=0.5$
$\frac{\partial\hat{r}}{\partial t}$
$H(\cdot)$
$P(\text{basis functions})=0.1$
$P(\text{variables})=0.2$
$+$
$\times$
$\times$
0.3
$\frac{\partial\hat{r}}{\partial x}$
0.8
$\frac{\partial^2\hat{r}}{\partial x^2}$
$P(\text{numbers})=0.2$
- The corresponding PDE
$\frac{\partial\hat{r}}{\partial t}-H\left[0.3\frac{\partial\hat{r}}{\partial x}+0.8\frac{\partial^2\hat{r}}{\partial x^2}\right]=0$
Evolutionary Processes
$\frac{\partial\hat{r}}{\partial t}+\left(-0.3\frac{\partial\hat{r}}{\partial x}+0.8\frac{\partial^2\hat{r}}{\partial x^2}\right)=0 \Rightarrow \frac{\partial\hat{r}}{\partial t}+\left(\frac{\partial\hat{r}}{\partial x}\right)=0$
$\frac{\partial\hat{r}}{\partial t}-0.7\left(\frac{\partial\hat{r}}{\partial x}\right)=0 \Rightarrow \frac{\partial\hat{r}}{\partial t}-0.7\left(-0.3\frac{\partial\hat{r}}{\partial x}+0.8\frac{\partial^2\hat{r}}{\partial x^2}\right)=0$
$\frac{\partial\hat{r}}{\partial t}-0.8\frac{\partial^2\hat{r}}{\partial x^2}=0 \Rightarrow \frac{\partial\hat{r}}{\partial t}-0.8\frac{\partial^2\hat{r}}{\partial x^2}+\left(-0.7\frac{\partial\hat{r}}{\partial x}\right)=0$
...

Fig. 2 Workflow chart for the genetic programming for system identification (GPSI) approach proposed; in the mathematical expression library box, $H(\cdot)$, $|\cdot|$ and $\mathrm{erf}(\cdot)$ represent the Heaviside step function, absolute value and error function, respectively; Details related to the evolutionary processes are exhibited in Fig. 3

hyperparameters by utilizing the user's prior knowledge of the target system (Steps 1–3). Next, stochastic optimization in the infinite function space is performed through GP for the randomly generated PDEs to minimize ϵ_{res} (see Eq. 2) (Steps 4–5). Finally, the system model $\mathcal{M}$ is selected among the optimized PDEs, which gives users an acceptable estimate (or prediction) of the system responses under different excitations, and the understanding of the system dynamics (Step 6–7). To implement this method, we develop our own algorithm, GPSI, by enhancing the GP-based identification algorithm for ODE problems in the previous study [11]. We refer to this study [11] for the detailed description of each component in the algorithm and its default parameters.

In this work, additional explanation for the method is categorized and presented mainly for two specific questions: How do we evaluate the plausibility of candidate PDEs for the system? How do we improve the fidelity of the PDEs for the system?

3.2 Fitness test

A fitness test is carried out in parallel to evaluate the suitability of each candidate PDE to a given data set, at every generation. The stochastic optimization utilized here (i.e., the evolutionary processes) is performed on the basis of fitness assigned to the candidate PDEs. In this sense, adopting a proper evaluation function for the fitness test is a key element of the proposed method. GPSI significantly improves its effectiveness and robustness in identifying a system model and its computational efficiency by employing the multi-objective loss function $\mathcal{L}$, defined by:

$$\mathcal{L} = \lambda_1 \epsilon_{\text{res}} + \lambda_2 \epsilon_{\text{eq}} + \lambda_3 p, \tag{4}$$

where λ_w ($w = 1, 2,$ and 3) is the weight of each term bound to the range [0,1] (a binary number, 0 or 1, is recommended for λ_1 and λ_2 to minimize work for tuning parameters; the details follows in the rest of this Section), ϵ_{res} is given in Eq. 3, and p is the number of nodes in the binary expression tree to which a PDE is converted. Here, ϵ_{eq} is the normalized root-mean-square error of the equation residual when $\mathcal{M}$ is rearranged with respect to $\frac{\partial \hat{r}}{\partial t}$ and substituted with the system measurements. It is defined as follows:

$$\epsilon_{\text{eq}} = \sqrt{\frac{\sum_i \sum_j \left\{ \frac{\partial \hat{r}}{\partial t} - \frac{\partial r}{\partial t} \right\}^2}{\sum_i \sum_j \left\{ \frac{\partial r}{\partial t} \right\}^2}}. \tag{5}$$

The objective of GPSI is to identify the PDE that has the minimum $\mathcal{L}$, which results in providing the system model. The loss function $\mathcal{L}$ (Eq. 4) consists of the model response error (i.e., ϵ_{res}), the equation residual error (i.e., ϵ_{eq}), and the complexity penalty of a PDE (i.e., p) with their own weights ($\lambda_1, \lambda_2,$ and λ_3). In the following, we provide details regarding the equation for $\mathcal{L}$. These details will assist users to select proper components and weights in $\mathcal{L}$.

- The model response error, ϵ_{res} (see Eq. 3), is the root-mean-square error between the model response $\hat{r}(x_i, t_j)$ and the system response $r(x_i, t_j)$, which is then normalized by the root-mean-square of $r(x_i, t_j)$. Users can determine whether it is considered in $\mathcal{L}$ or not, by assigning 0 or 1 to λ_1. Here, the model response is computed by solving a candidate PDE $\mathcal{M}_k$ ($k = 1, 2, \ldots, n_M$) with a suitable numerical method, such as the Runge–Kutta fourth-order method, under the measured excitation $f(x_i, t_j)$. In the integration process, initial and boundary conditions need to be posed. They come from physical constraints of a target system, or the observed data, $f(x_i, t_j)$, is incorporated into initial or boundary conditions. If a PDE does not reflect the system dynamics, ϵ_{res} is being rapidly accumulated from the initial state during the integration process. Thus, through this type of error, it is more likely to identify a PDE revealing the system dynamics than the case when only the equation residual error, ϵ_{eq}, is used in $\mathcal{L}$. Regarding the data quality needed, $f(x_i, t_j)$ and well-posed initial and boundary conditions are the only requirement to generate $\hat{r}(x_i, t_j)$ and then to compute ϵ_{res}. Thus, even if the spatial or temporal resolution of $r(x_i, t_j)$ is limited, ϵ_{res} is able to be evaluated. For the data size of $r(x_i, t_j)$, it is desirable for $r(x_i, t_j)$ to have a relatively long time domain for the accumulation of the response error during the integration process.
- The equation residual error, ϵ_{eq} (see Eq. 5), is the root-mean-square error between the time derivatives of the model response $\frac{\partial \hat{r}}{\partial t}$ and the system response $\frac{\partial r}{\partial t}$, which is then normalized by the root-mean-square of $\frac{\partial r}{\partial t}$. The time derivative values of the model response are computed by rearranging

a candidate PDE with respect to $\frac{\partial \hat{r}}{\partial t}$ and substituting the system measurements in it, while those of the system response are from the measurements or differentiation. Users can select whether it is included in $\mathcal{L}$ or not, by assigning 0 or 1 to λ_2. This type of error requires a higher data quality to be computed than the model response error, ϵ_{res}, because $r(x_i, t_j)$ and its all differential values, such as $\frac{\partial r}{\partial t}$, $\frac{\partial r}{\partial x}$, $\frac{\partial^2 r}{\partial x^2}$, etc. need to be substituted into a candidate PDE. Thus, ϵ_{eq} is considered in $\mathcal{L}$ (i.e., $\lambda_2 = 1$) only if all the data (i.e., $r(x_i, t_j)$ and its all differential values) are directly obtained from the measurements, or the data quality of $r(x_i, t_j)$ is appropriate to compute its differential terms with regard to data noise and resolution. In such a case, the associated computational cost is much cheaper than the one when ϵ_{res} is considered in $\mathcal{L}$.

- The model complexity penalty, p, is included in $\mathcal{L}$ (see Eq. 4) to prevent an overfitting issue. Since a candidate PDE is constructed from a binary expression tree, the number of nodes in the expression tree is an effective indicator representing the model complexity. Since there is a noticeable trade-off between the model complexity and the errors (i.e., ϵ_{res} and ϵ_{eq}) as a function of λ_3, it is the important hyperparameter in GPSI which needs to be investigated with preliminary trials. Besides from the model complexity and the errors, the reproducibility of GPSI reduces and its convergence time is extended as λ_3 decreases. Thus, it is recommended that users try a high value (e.g., 1×10^{-1}) first, and then decrease its magnitude of the order to find a proper value of λ_3.
- In this work, the computational efficiency of $\mathcal{L}$ has been significantly enhanced by incorporating a random sampling method [20,29] for ϵ_{res} and ϵ_{eq} (see Eqs. 3 and 5). For the fitness test at each generation, a certain number of small batches in the same size are randomly sampled from the system measurements, $r(x_i, t_j)$, and used to calculate the errors. The number and size of random small batches are denoted by n_b and n_s, respectively. One random small batch, namely $b_l(x_i, t_m)$ ($l = 1, 2, \ldots, n_b$), has the same spatial data points with $r(x_i, t_j)$ ($i = 1, 2, \ldots, n_x$) but has a subsequence ($m = 1, 2, \ldots, n_s$) of the original time sequence ($j = 1, 2, \ldots, n_t$). This subsequence is sampled by selecting a random starting moment from the original time sequence, and extracting the following subsequence with the size of n_s. By utilizing this random sampling method, the data use for the fitness test at each generation is considerably reduced (e.g., 10 or 20%), which directly leads to a lower computational cost for $\mathcal{L}$.

The computational efficiency of GPSI is further improved by utilizing advantages of both error types (i.e., ϵ_{res} and ϵ_{eq}) for $\mathcal{L}$. The GPSI simulation is carried out with three different phases of $\mathcal{L}$ in series. The first two phases only use the model response and the model complexity penalty (i.e., $(\lambda_1, \lambda_2, \lambda_3) = (1, 0, 0.01)$) to capture PDEs that are likely to reflect system dynamics in the early generation. The complexity penalty coefficient is determined by preliminary trials with the range of [0.1, 0.0001]. In the first phase, the GPSI simulation runs with one random small batch having 10% of the data, and stops when the change of the minimum loss function value is less than 1×10^{-3} in 100 consecutive generations. Candidate PDEs are thereby sorted out to have a similar response with the system response within a short time domain at a very early generation. In the second phase, by increasing the size of the random small batch to 20% of the data and the stop criterion to 200 consecutive generations, the candidate PDEs have more generations to evolve into the models predicting the system responses in a longer time domain. In the last phase, the equation residual error is adopted (i.e., $(\lambda_1, \lambda_2, \lambda_3) = (0, 1, 0.01)$) to expedite the convergence. While the model response error needs the maximum data size in one small batch for the error accumulation during the integration process, the equation residual error does not. Thus, ten random small batches with 2% of the data each are used to reduce data biases, which results in using 20% of the data in total for $\mathcal{L}$. The stop criterion of the last phase is that the minimum loss function value decreases less than 1×10^{-3} in 300 consecutive generations, or the GPSI simulation stops regardless of the phases when the number of generations reaches 1000.

3.3 Candidate PDEs and their evolution

GP is an algorithm that utilizes programs to perform a specific task, converts them into genetic information, and applies stochastic evolutionary processes to enhance the programs in consecutive generations [14]. In GPSI, the programs are candidate PDEs. In order to efficiently address candidate PDEs, binary expression trees are employed. Converting the mathematical components into nodes and terminals of the binary expression trees, candidate PDEs can be converted back-and-forth to the corresponding binary expression trees (see together the example of a candidate PDE in Fig. 2 and the conversion of a PDE into an expression tree in Fig. 3).

To initialize candidate PDEs through binary expression trees, several properties for binary expression trees need to be arranged on the basis of the user's prior knowledge. Above all, it is most important to list mathematical components including operations (e.g., addition, subtraction, and multiplication), basis functions (e.g., Heaviside step function, absolute value function, sign function, exponential function, logarithmic function, error function, sine function and cosine function), variables (e.g., $\hat{r}$, $\frac{\partial \hat{r}}{\partial t}$, $\frac{\partial \hat{r}}{\partial x}$, $\frac{\partial^2 \hat{r}}{\partial x^2}$), and numbers (i.e., real numbers randomly chosen between 0 and 1 which will be optimized afterward). Expression trees are randomly generated by assigning the arranged mathematical components into their nodes and terminals, as depicted in the example of a candidate PDE in Fig. 2. This random generation is controlled by the selected maximum tree level (i.e., the maximum number counting from the initial node to the last terminal, e.g., 5), the probabilistic weights for the component types (i.e., $\{$operations, functions, variables, numbers$\} = \{0.5, 0.1, 0.2, 0.2\}$), and the expression tree growing method (i.e., Ramped half-and-half method). Once these properties are set up, a large ensemble of 100 candidate PDEs ($\mathcal{M}_k$ where $k = 1, 2, \ldots, n_M; n_M = 100$) are generated in a stochastic manner for the first population. All the hyperparameters relevant to the expression trees were tested and determined in the previous study [11], which allow testing a variety of expression trees while maintaining a viable computational cost.

The randomly initialized candidate PDEs, $\mathcal{M}_k$, are then optimized with evolutionary processes to produce the system model. Since the evolutionary processes (i.e., crossover and mutation) involve single or multiple expression trees, they are applied to the population of the candidate PDEs and yield a new population for the next generation. By repeating the evolutionary processes in consecutive generations until the stop criteria (refer to Sect. 3.2) are satisfied, the system model can be identified from the best suitable PDE in the last population. Additionally, in order to prevent a memory problem in the computational engine (computer) and enhance computational efficiency, several techniques for bloat control (i.e., simplifying expression trees, limiting the number of tree nodes, and removing repetition in the population) are incorporated at the end of the evolutionary processes. By integrating the evolutionary processes and the bloat control techniques into one evolution module, the framework for GPSI becomes simplified with a higher computational efficiency.

Specific description of the evolutionary processes is presented in Fig. 3. Firstly, loss function values (Eq. 4) are assigned to the candidate PDEs, $\mathcal{M}_k$, in the fitness test beforehand. The evolution starts by the representation in which the PDEs having a top 10% of the loss function values are kept intact for the next generation (see the box of current generation in Fig. 3). The rest of the PDEs for the next generation are produced by the crossover or the mutation. For these evolutionary processes, one PDE is firstly selected from the tournament selection method which randomly selects two PDEs in the current population and keeps the one having a better loss function value [14]. This selected PDE, in the form of the expression tree, goes through the crossover and the mutation (see the arrow of evolutionary processes in Fig. 3), based on their probabilities, i.e., 0.8 and 0.2, respectively. The crossover randomly picks the node of the expression tree converted from the selected PDE and exchanges with another randomly picked node of the expression tree from another selected PDE. The mutation randomly picks the node of the expression tree from the selected PDE and exchanges it with a randomly generated expression tree.

At the end of the evolutionary processes, several bloat control techniques are carried out (see the arrow of bloat control in Fig. 3). Firstly, the modified expression tree is rearranged by performing mathematical operations and simplifying expressions. Next, two conditions are checked. The first one is whether or not the number of nodes in the expression tree is over the maximum node number, e.g., 31. This maximum number of nodes occurs in a fully grown expression tree with the tree level 6. The second condition is whether the PDE from the expression tree has the same structure of the PDEs in the current population. When the expression tree satisfies any of these two conditions, the evolutionary processes repeat from the selection step to guarantee all the expression trees for PDEs are simplified, not too complicated, and unique (see the box of next generation in Fig. 3).

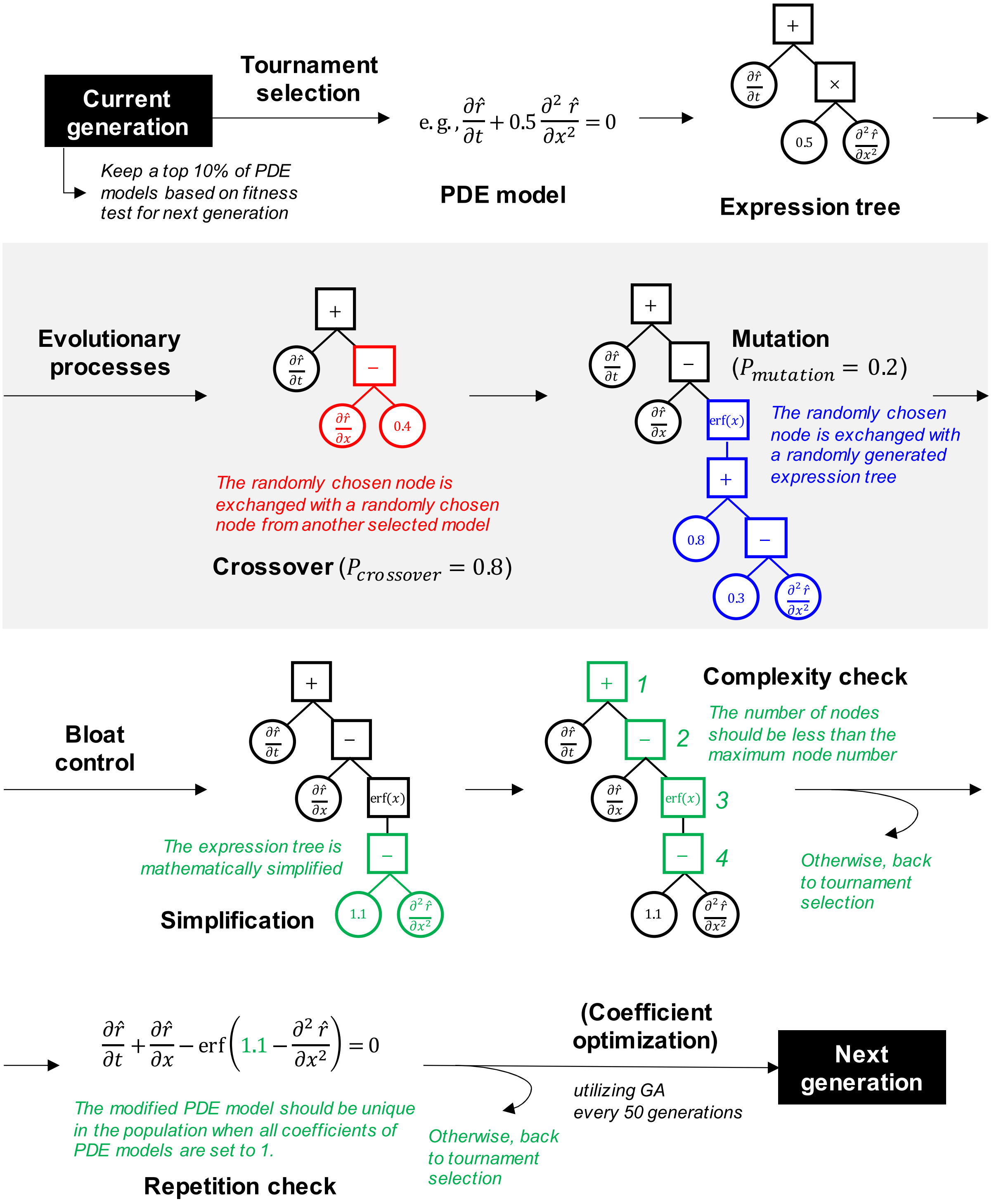
Current generation
Tournament selection
Keep a top 10% of PDE models based on fitness test for next generation
e. g., $\frac{\partial \hat{r}}{\partial t} + 0.5 \frac{\partial^2 \hat{r}}{\partial x^2} = 0$
PDE model
+
×
0.5
Expression tree
Evolutionary processes
−
0.4
The randomly chosen node is exchanged with a randomly chosen node from another selected model
Crossover ($P_{crossover} = 0.8$)
Mutation ($P_{mutation} = 0.2$)
erf(x)
0.8
0.3
The randomly chosen node is exchanged with a randomly generated expression tree
Bloat control
The expression tree is mathematically simplified
1.1
Simplification
Complexity check
1
2
3
4
The number of nodes should be less than the maximum node number
Otherwise, back to tournament selection
$\frac{\partial \hat{r}}{\partial t} + \frac{\partial \hat{r}}{\partial x} - \mathrm{erf}\left(1.1 - \frac{\partial^2 \hat{r}}{\partial x^2}\right) = 0$
The modified PDE model should be unique in the population when all coefficients of PDE models are set to 1.
Repetition check
(Coefficient optimization)
utilizing GA every 50 generations
Otherwise, back to tournament selection
Next generation

Fig. 3 Representative snapshot of the flowchart to implement the evolutionary processes of candidate PDEs utilizing their expression trees

In addition to the evolution of the structure of the candidate PDEs, the optimization of their coefficients are performed separately once in every 50 generations (see the last arrow in Fig. 3). A genetic algorithm [9] is employed instead of gradient-based optimization methods because the library of basis functions for PDEs includes discontinuous functions such as Heaviside step function. The genetic algorithm is applied to each PDE separately. It utilizes the coefficient vector extracted from the PDE. Candidate coefficient vectors are generated by multiplying random real numbers from the uniform distribution with lower and upper bounds 0.05 and 20 into each element of the original vector, respectively, to thoroughly search optimum coefficient values around the original coefficient values. The range of the uniform distribution was tested and determined in the previous study [11]. The population of the candidate coefficient vectors then evolves in consecutive generations in similar fashion as the evolution of the crossover and mutation in GP. In consideration of the computational cost, the coefficient optimization is applied to the PDEs having a top 30% of the loss function values. The user-selectable stop criteria for the coefficient optimization are: the change of the minimum loss function value is selected, e.g., as less than 1×10^{-3} in 30 consecutive generations, or the number of generations reached is 300.

4 Illustrative examples

The proposed stochastic identification method, GPSI, is now demonstrated in two applications. Two cases employ canonical PDEs, i.e., the Burgers' equation and the advection-dispersion equation (ADE). Training data for these cases are prepared by solving theoretical reference models and adding Gaussian noises (to simulate unavoidable measurement noise pollution). The results show that the identification performance relies on noise levels and system characteristic numbers. The system characteristic numbers represent the contribution of each term in the reference models. All GPSI simulations were performed with the use of parallel processing on high-performance computing systems [10]. The processing unit has Intel Xeon 4116 dodeca-core whose CPU clock is 2.10 GHz, and 94 GB RAM.

4.1 Burgers' equation

Burgers' equation is a nonlinear PDE which is expressed in the following non-dimensional form:

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + \frac{1}{\mathrm{Re}}\frac{\partial^2 u}{\partial x^2}, \tag{6}$$

where u is the dimensionless fluid longitudinal velocity and Re is the Reynolds number, which describes the system characteristics through the ratio of inertial forces to viscous forces within the fluid. Here, x denotes the dimensionless coordinate system and t is the dimensionless time.

In order to test different system characteristics, three different regimes of Re (i.e., 20, 100, and 500) are used. The excitation for the reference system model (see Eq. 6) is stated as the initial condition characterized by a Gaussian pulse:

$$f(x,t) = \begin{cases} A\exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right), & \text{if } \ t = 0; \\ 0, & \text{if } \ t > 0, \end{cases} \tag{7}$$

where μ is the central location of the Gaussian pulse, σ is the standard deviation of the pulse from the central location, and A is the coefficient to normalize the peak of the pulse to 1. Regarding boundary conditions, as described in Sect. 3.2, any type of boundary conditions are possible to be adopted. The boundary conditions can be set up from observed data or prior knowledge about a target system. In this example, periodic boundary conditions are employed to exclude any boundary effects of the finite space domain and to present as a general example.

For the training and validation data, the space and time domains are defined with proper step sizes, i.e., $x \in [0, 1]$ with $\Delta x = 1/128$ and $t \in [0, 1]$ with $\Delta t = 1 \times 10^{-4}$. The system response $u(x_i, t_j)$ ($i = 1, 2, \ldots, n_x$ and $j = 1, 2, \ldots, n_t$; $n_x = 128$ and $n_t = 1 \times 10^4$) and its derivatives (i.e., $\frac{\partial u}{\partial t}$, $\frac{\partial u}{\partial x}$, and $\frac{\partial^2 u}{\partial x^2}$) are computed by solving Eq. 6 under $f(x_i, t_j)$. The values of μ and σ for $f(x_i, t_j)$ are suitably selected to exhibit system characteristics within the space and time domains: 0.2 and 0.05 for the training data, and 0.4 and 0.1 for the validation data, respectively. For the data noise, a zero-mean Gaussian noise is embedded only into the training data with different levels (i.e., ε = 1, 5, and 10%). Accordingly, nine identification cases are prepared in total with the Burgers' equation as functions of Re and ε. For the hyperparameters of GPSI, the default values mentioned in Sects. 3.2 and 3.3 are used.

Under the consideration that GPSI performs a stochastic optimization, each identification case has 10 trials with different random seeds, and one PDE is selected as the system model among the PDEs in all the trials in the consideration of ϵ_{res} with the validation data (see Eq. 2) and model complexity.

The identified PDEs and their ϵ_{res} with training and validation data (namely training ϵ_{res} and validation ϵ_{res}, respectively, hereafter) are presented in Table 1 for different Re and ε. In all the identification cases, the true form of the Burgers' equation is discovered with low training and validation ϵ_{res}. As an example, one identification case with Re $=$ 100 and $\varepsilon = 5\%$ is shown in Fig. 4. The loss function values and their composition are exhibited over generations in Fig. 4a. The values of the validation ϵ_{res} (Eq. 3) are plotted together as the reference. On the bottom side of Fig. 4, model response surfaces (Fig. 4b–e) and error surfaces (see Fig. 4f–i) are displayed for the reference generations, such as the first generation and each session end. Over the generation, the validation ϵ_{res} decreases although it has some fluctuations due to the stochastic sampling. While the model response surface converges to the reference model responses, the biased error surface becomes reduced and homogeneous in both space and time.

Table 1 reveals that the example case discovers the true Burgers' equation (compare Case 5 with the reference case in Table 1). The estimated Reynolds number, $\widehat{\text{Re}}$, extracted from the model (i.e., 100) is the same as the true ratio of inertial forces to viscous forces within the fluid (i.e., 100). Since the identified PDE has the same structure as the reference model with the accurate coefficients, its responses are well-matched with the training and validation data as exhibited in Fig. 5. Its training and validation ϵ_{res} are 5.0 and 0.0%, respectively.

In addition to predicting the behavior of the target system (i.e., the validation test), the identified PDE allows users to enhance understanding of the system dynamics. The example model (see Case 5 in Table 1) is further analyzed by superposing the contributions of its terms (i.e., $1.000\hat{u}\frac{\partial \hat{u}}{\partial x}$ and $0.010\frac{\partial^2 \hat{u}}{\partial x^2}$) and their respective responses in Fig. 6. On the left side (Fig. 6a), the values of each term in the identified model are presented over the space domain when the validation excitation (i.e., the black line; see Eq. 7) is given. Figure 6b shows the response simulated with each term in the identified model separately under the same excitation for the short period of time (i.e., $0 < t < 0.15$), which reveals the dynamics of each term respectively. The advection term (i.e., $1.000\hat{u}\frac{\partial \hat{u}}{\partial x}$) has a major contribution to the temporal change of $\hat{u}$ (see the blue and red lines in Fig. 6a), and it moves the pulse forward with the ratio of $\hat{u}$, leading to the nonlinear advection (see the blue line in Fig. 6b). The diffusion term (i.e., $0.010\frac{\partial^2 \hat{u}}{\partial x^2}$) describes the smooth dispersion as shown in the green lines in Fig. 6a, b. Its contribution is less than the advection term, resulting in the final model response (see the red line in Fig. 6b). As a result, the inspection of the identified PDE reveals that the system dynamics consists of nonlinear advection and linear dispersion.

Although all the identification cases discover the reference model within 10 trials, each trial in one identification case shows a different result in identifying PDEs. In order to evaluate the reproducibility of GPSI, the average performance of 10 trials is investigated for each identification case by calculating the probability of discovering the reference PDE (i.e., the Burgers' equation), the average validation ϵ_{res}, and the average convergence time. The average performance as functions of Re and ε are summarized via the contours depicted in Fig. 7. On the left side (Fig. 7a), the contour exhibits the probability of identifying the reference PDE, namely the model probability. This contour represents the diagram of the identified PDE, presenting its structure together. The model probability stays similar as to ε and decreases as the higher Re is used. When Re $=$ 500, the chance to lose the dispersion term in the identified PDEs becomes high, leading to the models having linear advection or degradation. On the right side (see Fig. 7b, c, respectively), the average validation ϵ_{res} and the average convergence time are displayed. They are as well as affected more by Re than by ε. As Re is higher, the validation ϵ_{res} and the convergence time increase. Consequently, the performance of GPSI is robust to the data noise (i.e., ε) and shows a good reproducibility as long as the difference of the contributions that the terms in the model make is less than 500 times.

4.2 Advection–dispersion equation

The advection–dispersion equation (ADE), which shares the same mathematical form as the convection-diffusion equation, is used as the second example of multi-

Table 1 GPSI-identified PDEs and their ϵ_{res} from the training and validation data depending on different Re and ε; $\hat{u}$ is the fluid longitudinal velocity estimated from the identified model; all coefficients are rounded to three decimal places

Case	Re	ε (%)	Identified PDE	ϵ_{res} (%) Training	Validation
Ref.	–	–	$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + \frac{1}{\text{Re}}\frac{\partial^2 u}{\partial x^2}$	–	–
1	20	1	$\frac{\partial \hat{u}}{\partial t} = -1.000\hat{u}\frac{\partial \hat{u}}{\partial x} + 0.050\frac{\partial^2 \hat{u}}{\partial x^2}$	1.0	0.0
2	20	5	$\frac{\partial \hat{u}}{\partial t} = -0.999\hat{u}\frac{\partial \hat{u}}{\partial x} + 0.050\frac{\partial^2 \hat{u}}{\partial x^2}$	5.0	0.2
3	20	10	$\frac{\partial \hat{u}}{\partial t} = -1.000\hat{u}\frac{\partial \hat{u}}{\partial x} + 0.049\frac{\partial^2 \hat{u}}{\partial x^2}$	10.0	0.3
4	100	1	$\frac{\partial \hat{u}}{\partial t} = -1.000\hat{u}\frac{\partial \hat{u}}{\partial x} + 0.010\frac{\partial^2 \hat{u}}{\partial x^2}$	1.0	0.0
5	100	5	$\frac{\partial \hat{u}}{\partial t} = -1.000\hat{u}\frac{\partial \hat{u}}{\partial x} + 0.010\frac{\partial^2 \hat{u}}{\partial x^2}$	5.0	0.0
6	100	10	$\frac{\partial \hat{u}}{\partial t} = -1.000\hat{u}\frac{\partial \hat{u}}{\partial x} + 0.010\frac{\partial^2 \hat{u}}{\partial x^2}$	10.0	0.1
7	500	1	$\frac{\partial \hat{u}}{\partial t} = -1.000\hat{u}\frac{\partial \hat{u}}{\partial x} + 0.002\frac{\partial^2 \hat{u}}{\partial x^2}$	1.0	0.0
8	500	5	$\frac{\partial \hat{u}}{\partial t} = -0.999\hat{u}\frac{\partial \hat{u}}{\partial x} + 0.002\frac{\partial^2 \hat{u}}{\partial x^2}$	5.1	0.2
9	500	10	$\frac{\partial \hat{u}}{\partial t} = -1.006\hat{u}\frac{\partial \hat{u}}{\partial x} + 0.002\frac{\partial^2 \hat{u}}{\partial x^2}$	10.4	1.4

physics system models. The ADE describes solute transport with linear advection and dispersion terms. In an one-dimensional space, the dimensionless form of the ADE is written as:

$$\frac{\partial c}{\partial t} = -\frac{\partial c}{\partial x} + \frac{1}{\text{Pe}}\frac{\partial^2 c}{\partial x^2}, \tag{8}$$

where c is the solute concentration at dimensionless location x and time t and Pe is the Péclet number which represents the ratio of the advection to the dispersion time scales. Pe dictates whether the system dynamics is advection-dominated or dispersion-dominated.

Three different values of Pe are selected (i.e., 20, 100, and 500). The initial and boundary conditions, i.e., the excitation (see Eq. 7) and the periodic boundary conditions, and the GPSI setting are the same as those for the Burgers' equation discussed above (refer to Sect. 4.1). Given the similarity in the set-up, GPSI is able to address a class of PDE problems without tailoring many hyperparameters for each target system (that demonstrating the robustness of the procedure under discussion). The training and validation data are generated in a similar manner. Therefore, nine identification cases, from 3 Pe values and 3 noise levels (i.e., ε = 1, 5, and 10%), are used in total.

The identified PDEs and their training and validation ϵ_{res} are tabulated as functions of Pe and ε in Table 2. The results of GPSI for the ADE are very similar to the Burgers equation cases. In all the identification cases, the PDE responses converge to the training data, and their error surfaces are reduced and homogeneous over generations in the same way as the Burgers equation cases (refer to Fig. 4). At the end, the true form of the ADE is discovered, regardless of Pe and ε with sightly different coefficients, presenting low training and validation ϵ_{res}. For instance, one identification case with Pe $=$ 100 and ε $=$ 5% shows that the identified model's responses are in agreement with the training and validation data (Fig. 8). Its training and validation ϵ_{res} are 5.0 and 0.0%, respectively. The estimated Péclet number (i.e., 100) extracted from the model is identical to the one in the reference system (i.e., 100).

Close inspection of the PDE, identified with Pe $=$ 100 and ε $=$ 5% (see Case 5 in Table 2), reveals that there are both linear advection and dispersion in the system dynamics. The contribution of each term is compared by superposing all the terms (i.e., $1.000\frac{\partial \hat{c}}{\partial x}$ and $0.010\frac{\partial^2 \hat{c}}{\partial x^2}$) in Fig. 9. On the left side (see Fig. 9a), given the validation excitation (i.e., the black line), the contribution of the advection term (i.e., the blue line) is larger than the one of the dispersion term (i.e., the green line) to the temporal evolution of $\hat{c}$ (i.e., the red line). On the right side (see Fig. 9b), the short-term ($0 < t < 0.15$) responses simulated from the advection and dispersion terms, respectively, are presented. As a result, the final response of the identified PDE (i.e., the red line) exhibits the advection-dominated transport with relatively small dispersion, which corresponds with the system dynamics.

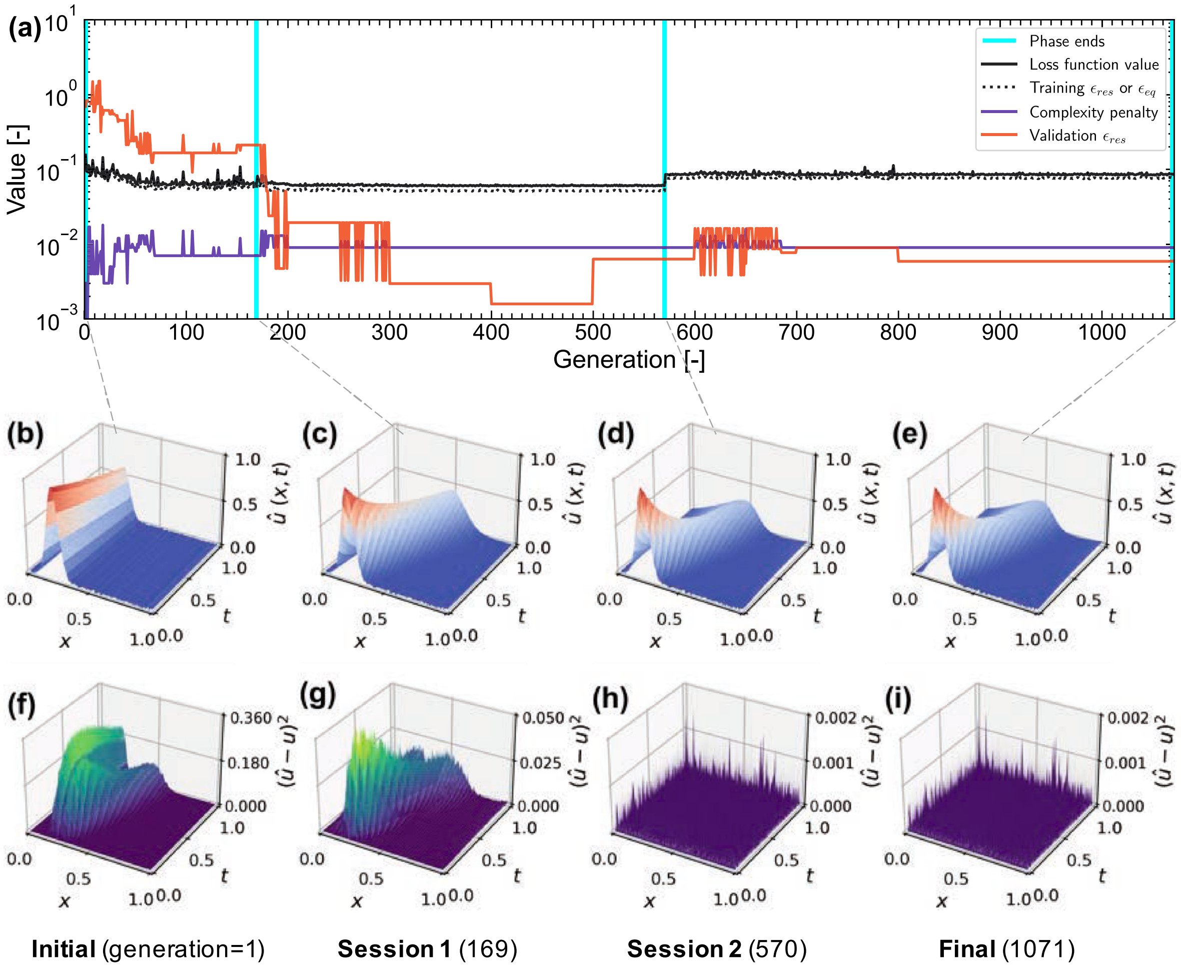

Fig. 4 GPSI simulation for the Burgers' equation with Re = 100 and $\varepsilon = 5\%$; **a** the loss function composition and the validation ϵ_{res} as a function of the generation; **b**–**e** the PDE response surfaces at the initial generation (i.e., **b**) and each $\mathcal{L}$ phase end (i.e., **c**–**e**); and **f**–**i** the corresponding error surfaces; note that, for clarity, the amplitude levels displayed in **f**–**i** are not the same

The performance of GPSI in discovering the governing physical equation is summarized through the contour plots in Fig. 10. The probability of discovering the reference PDE, the average validation ϵ_{res}, and the average convergence time are displayed as to different Pe and ε in Fig. 10a–c, respectively. In most of the cases, the model probability remains high, and the average validation ϵ_{res} and the average convergence time stay low. When both Pe and ε increase to 500 and 10%, respectively, the model probability decreases, and the average validation ϵ_{res} and the average convergence time increase. Compared to the performance for the Burgers' equation, the overall performance is better in all the cases (see Fig. 10 in comparison with Fig. 7).

In addition to the identification cases for the linear ADE, one extra case is examined to test the performance of identification method. Here, we wish to investigate if the approach is able to discover a nonlinear ADE. Equation 8 is modified to have a nonlinear dispersion coefficient (or a nonlinear diffusion coefficient):

$$\begin{aligned}\frac{\partial c}{\partial t} &= -\frac{\partial c}{\partial x} + \frac{1}{\mathrm{Pe}}\frac{\partial}{\partial x}\left(c\frac{\partial c}{\partial x}\right)\\ &= -\frac{\partial c}{\partial x} + \frac{1}{\mathrm{Pe}}\left(\frac{\partial c}{\partial x}\right)^2 + \frac{1}{\mathrm{Pe}}c\frac{\partial^2 c}{\partial x^2}. \qquad (9)\end{aligned}$$

The training data are generated using Eq. 9 with Pe = 20 and $\varepsilon = 1\%$. The use of the the hyperparam-

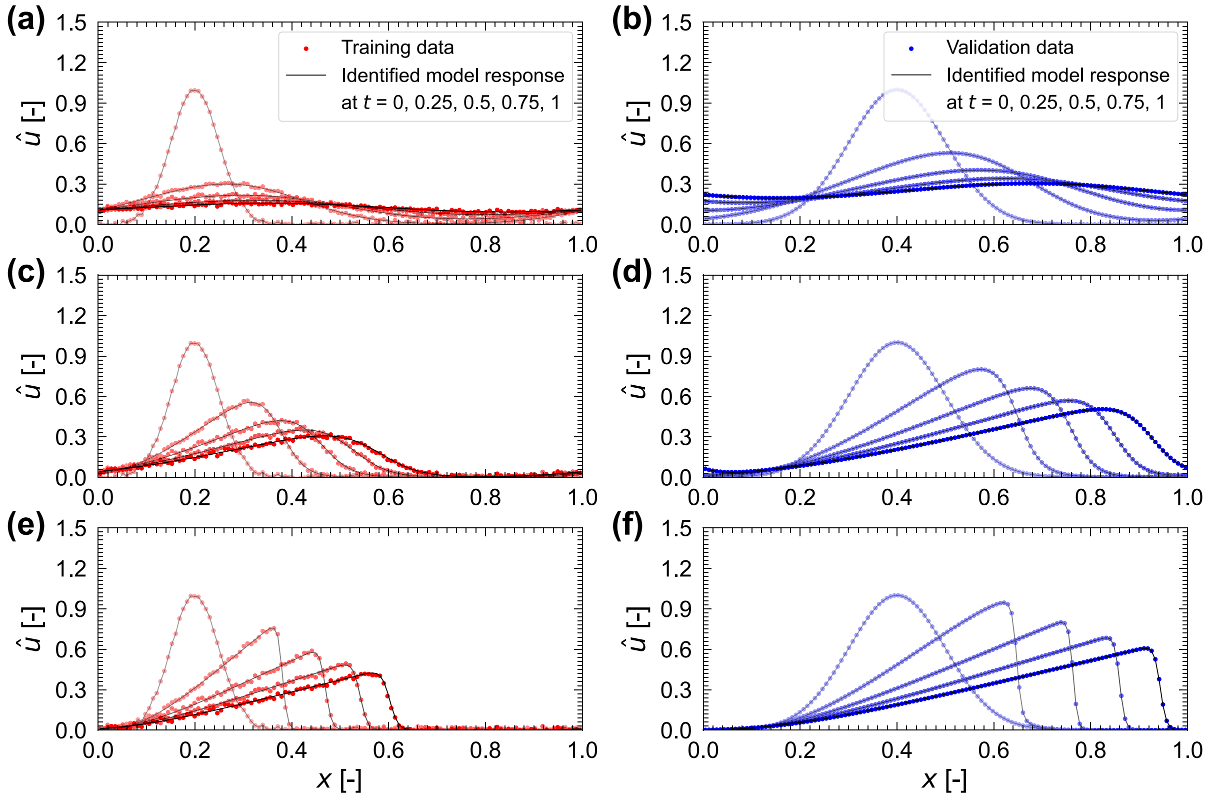

Fig. 5 Identified PDE responses with the training (red dots) and validation data (blue dots) from identification cases for the Burgers' equation; **a** and **b** correspond to Case 2 in Table 1 with the data from Re $=$ 20 and $\varepsilon = 5\%$; **c** and **d** correspond to Case 5 in Table 1 with the data from Re $=$ 100 and $\varepsilon = 5\%$; **e** and **f** correspond to Case 8 in Table 1 with the data from Re $=$ 500 and $\varepsilon = 5\%$. (Color figure online)

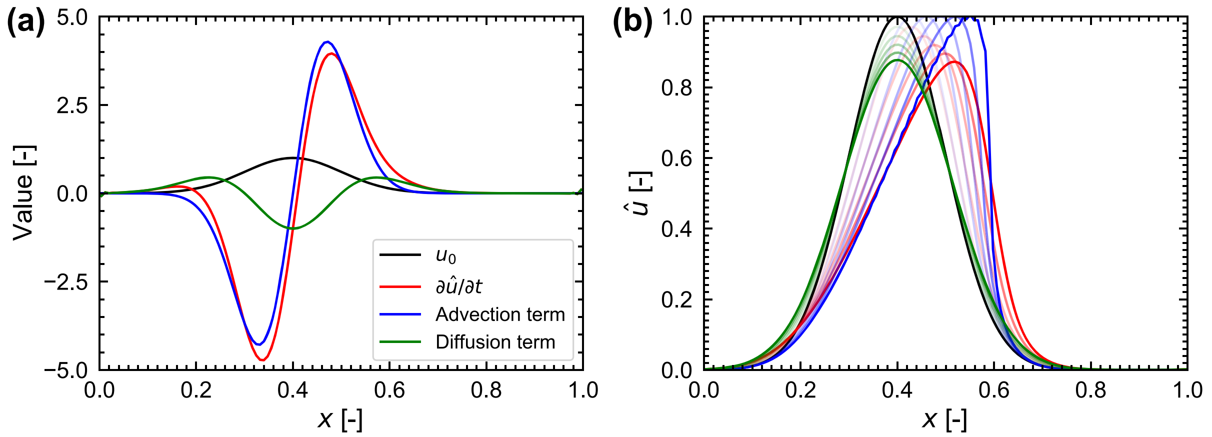

Fig. 6 Contribution of each term in the identified PDE (see Case 5 in Table 1) from one identification case for the Burgers' equation using Re $=$ 100 and $\varepsilon = 5\%$; the advection and diffusion terms are $1.000\hat{u}\frac{\partial \hat{u}}{\partial x}$ and $0.010\frac{\partial^2 \hat{u}}{\partial x^2}$, respectively; **a** the values of the components in the model under the given validation excitation u_0; **b** the responses from the components for a short time domain $(0 < t < 0.15)$; the incremental shades of the colors represent the responses at different time moments $(t = 0.03, 0.06, \ldots, 0.15)$. (Color figure online)

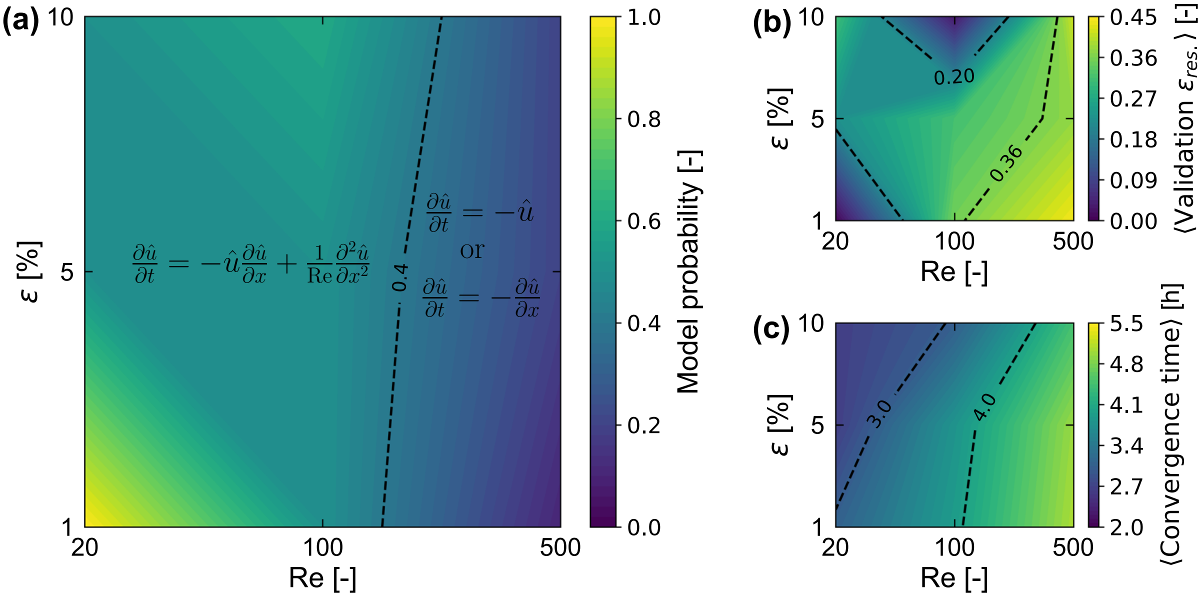

Fig. 7 Average performance of GPSI in identifying the reference PDE (i.e., the Burgers' equation) depending on Re and ε; **a** the phase diagram of the identified PDEs on the probability of discovering the reference model; **b** the average validation ϵ_{res}; and **c** the average convergence time

eters is similar to those employed for the linear ADE case (with minor modifications). The modifications are needed in order to guarantee the numerical stability of the complex equation form (i.e., $\Delta t = 1 \times 10^{-5}$) and to simplify the fitness test (i.e., $(\lambda_1, \lambda_2, \lambda_3) = (0, 1, 0.0001)$ and 50% of the data use in one batch). In Fig. 11, the training (red dots) and validation data (blue dots) exhibit the nonlinear transport behavior in both space and time, leading to bell-shaped curves to convex ones. As a result of the identification, the exact form of the nonlinear ADE is discovered with slightly different coefficients (Eq. 10), presenting 2.6 and 4.0% for the response errors with training and validation data, respectively:

$$\frac{\partial c}{\partial t} = -\frac{\partial c}{\partial x} + 0.05\left(\frac{\partial c}{\partial x}\right)^2 + 0.047c\frac{\partial^2 c}{\partial x^2}. \qquad (10)$$

5 Discussion

5.1 GPSI for the class of PDE problems

The illustrative examples show that GPSI is an effective and robust stochastic method to discover PDEs from a given data set. In the two examples (Sects. 4.1 and 4.2), there are high chances that the identified PDEs are the same as the reference models (i.e., 45 and 98% for the Burgers' equation and the ADE, respectively) within the range of the noise levels (i.e., 1–10%) and characteristic numbers (i.e., 20–500 of Re and Pe) considered. . Thus, the identified models remain highly accurate in predicting system responses in the validation test. Since all the results were obtained without adjusting the hyperparameters of GPSI to each example, the approach is hereby shown to be appropriate for the class of PDE identification problems under discussion.

Table 2 GPSI-identified PDEs and their ϵ_{res} from the training and validation data depending on different Pe and ε; $\hat{c}$ is the solute concentration estimated from the identified model; all coefficients are rounded to three decimal places

Case	Pe	ε (%)	Identified PDE	ϵ_{res} (%) Training	Validation
Ref.	–	–	$\frac{\partial c}{\partial t} = -\frac{\partial c}{\partial x} + \frac{1}{\text{Pe}}\frac{\partial^2 c}{\partial x^2}$	–	–
1	20	1	$\frac{\partial \hat{c}}{\partial t} = -1.000\frac{\partial \hat{c}}{\partial x} + 0.050\frac{\partial^2 \hat{c}}{\partial x^2}$	1.0	0.0
2	20	5	$\frac{\partial \hat{c}}{\partial t} = -1.000\frac{\partial \hat{c}}{\partial x} + 0.050\frac{\partial^2 \hat{c}}{\partial x^2}$	5.0	0.2
3	20	10	$\frac{\partial \hat{c}}{\partial t} = -1.000\frac{\partial \hat{c}}{\partial x} + 0.049\frac{\partial^2 \hat{c}}{\partial x^2}$	10.0	0.0
4	100	1	$\frac{\partial \hat{c}}{\partial t} = -1.000\frac{\partial \hat{c}}{\partial x} + 0.010\frac{\partial^2 \hat{c}}{\partial x^2}$	1.0	0.0
5	100	5	$\frac{\partial \hat{c}}{\partial t} = -1.000\frac{\partial \hat{c}}{\partial x} + 0.010\frac{\partial^2 \hat{c}}{\partial x^2}$	5.0	0.0
6	100	10	$\frac{\partial \hat{c}}{\partial t} = -1.000\frac{\partial \hat{c}}{\partial x} + 0.010\frac{\partial^2 \hat{c}}{\partial x^2}$	10.0	0.0
7	500	1	$\frac{\partial \hat{c}}{\partial t} = -1.000\frac{\partial \hat{c}}{\partial x} + 0.002\frac{\partial^2 \hat{c}}{\partial x^2}$	1.0	0.0
8	500	5	$\frac{\partial \hat{c}}{\partial t} = -1.000\frac{\partial \hat{c}}{\partial x} + 0.002\frac{\partial^2 \hat{c}}{\partial x^2}$	5.1	0.1
9	500	10	$\frac{\partial \hat{c}}{\partial t} = -1.001\frac{\partial \hat{c}}{\partial x} + 0.002\frac{\partial^2 \hat{c}}{\partial x^2}$	10.2	0.1

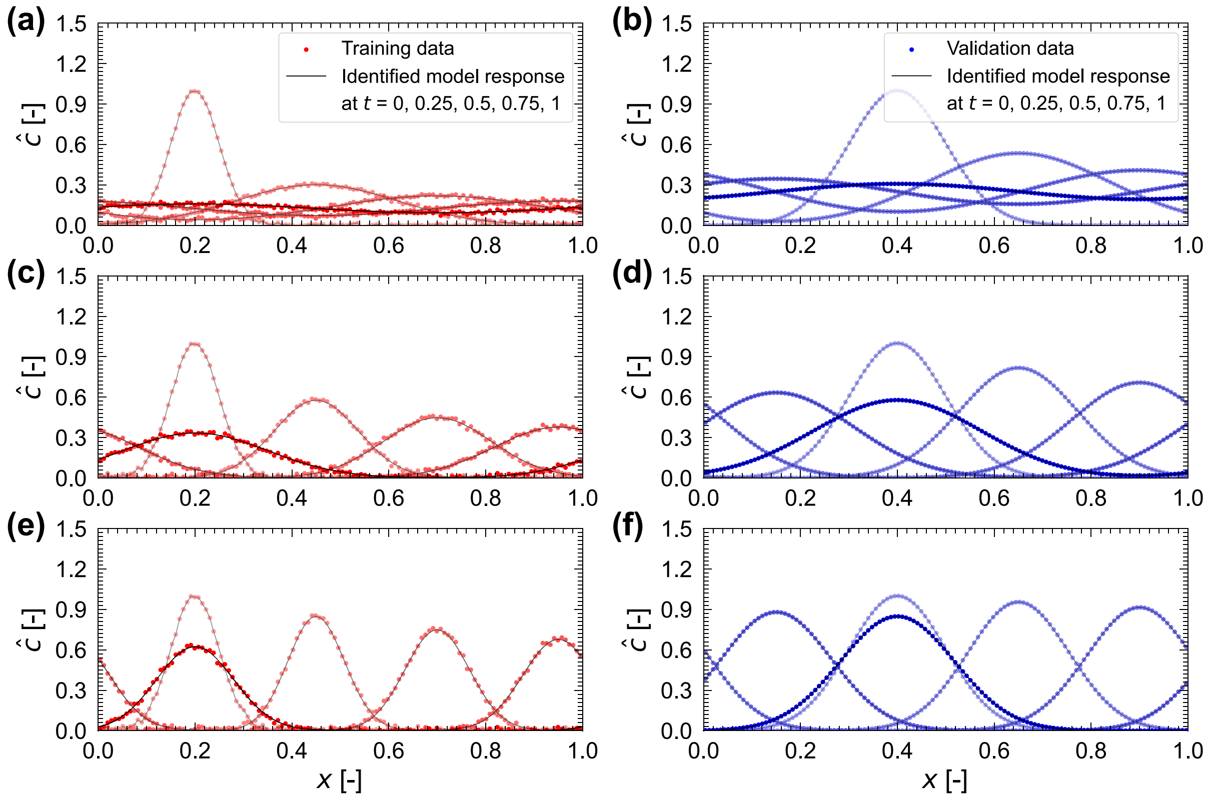

Fig. 8 Identified PDE responses with the training (red dots) and validation data (blue dots) from identification cases for the ADE; **a** and **b** correspond to Case 2 in Table 2 with the data from Pe = 20 and ε = 5%; **c** and **d** correspond to Case 4 in Table 2 with the data from Pe = 100 and ε = 5%; **e** and **f** correspond to Case 8 in Table 2 with the data from Pe = 500 and ε = 5%. (Color figure online)

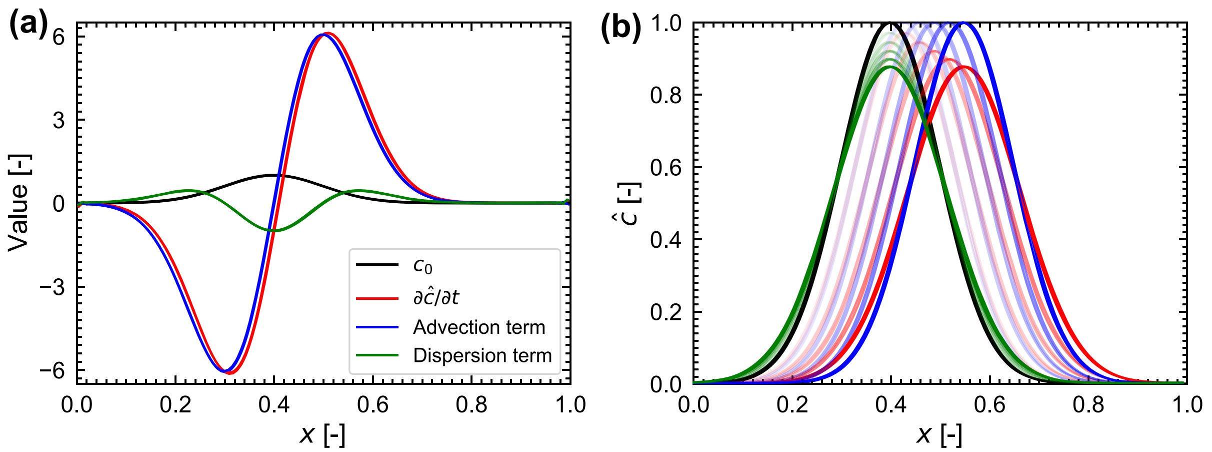

Fig. 9 Contribution of each term in the identified PDE (see Case 5 in Table 2) from one identification case for the ADE using $\mathrm{Re} = 100$ and $\varepsilon = 5\%$; the advection and dispersion terms are $1.000\frac{\partial \hat{c}}{\partial x}$ and $0.010\frac{\partial^2 \hat{c}}{\partial x^2}$, respectively; **a** the values of the components in the model under the given validation excitation c_0; **b** the responses from the components for a short time domain ($0 < t < 0.15$); the incremental shades of the colors represent the responses at different time moments ($t = 0.03, 0.06, \ldots, 0.15$)

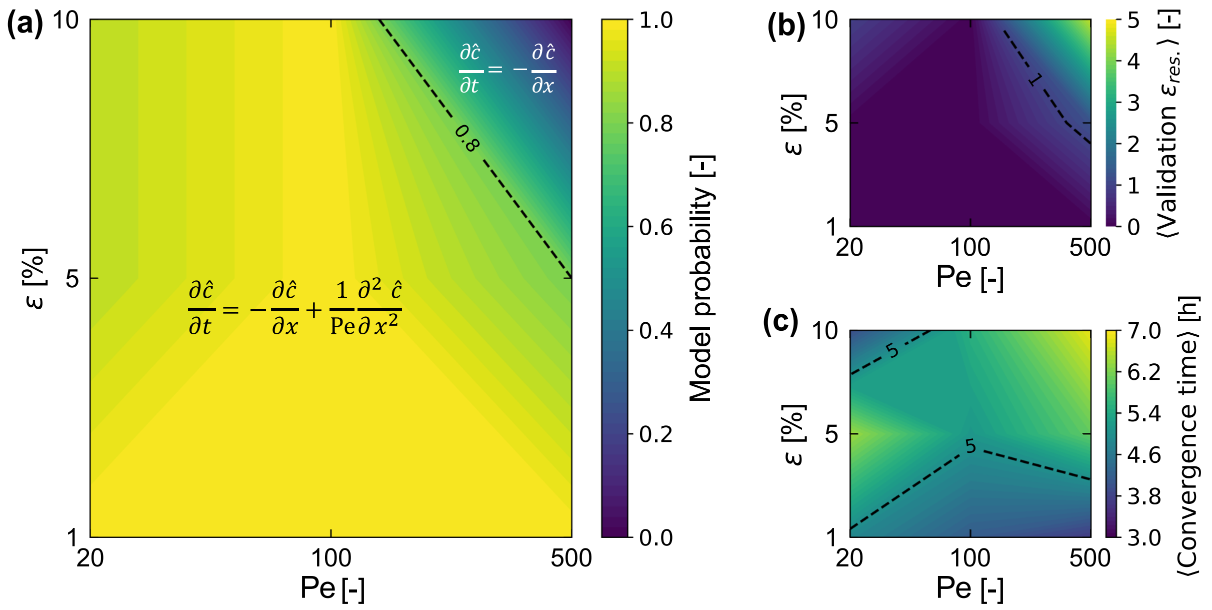

Fig. 10 Average performance of GPSI in identifying the reference PDE (i.e., the ADE) as to Pe and ε; **a** the phase diagram of the identified PDEs on the probability of discovering the reference model; **b** the average validation ϵ_{res}; and **c** the average convergence time

Although GPSI is a model-free identification method, and its hyperparameters do not require repetitive tuning for each new identification problem, there is a room to incorporate user's prior knowledge about a target system to adjust the tuning parameters. Since candidate PDEs are constructed on the basis of expression trees, it is important to select proper mathematical components in the expression library as building blocks. This can be achieved by relying on prior knowledge of simpler (yet similar) physical systems. For instance, some components such as Heaviside step and absolute functions can be employed or removed in the library according to the

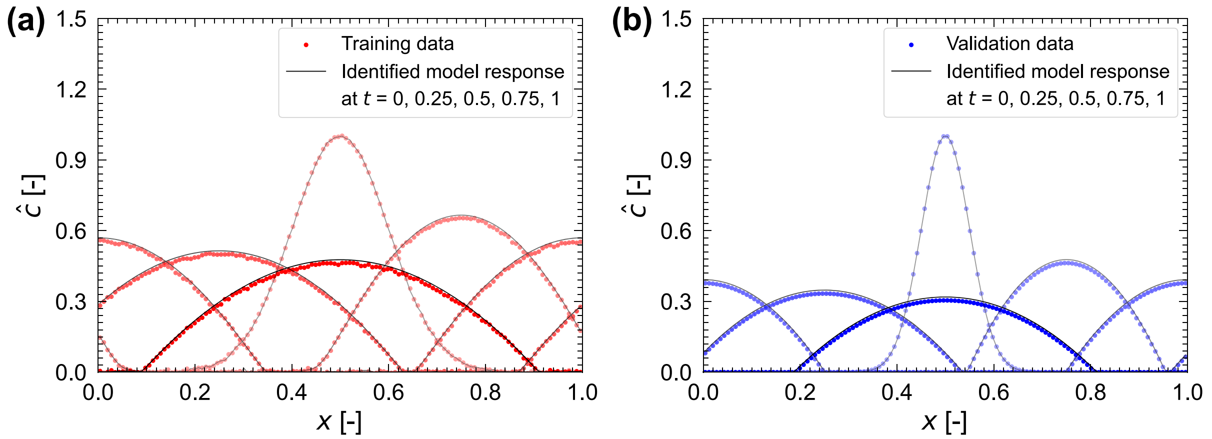

Fig. 11 Identified PDE responses with the training (red dots) and validation data (blue dots) from identification cases for the nonlinear ADE; both initial states are at the center, $x = 0.5$, with the peak level, $c = 1$, and they move to the right beyond the periodic boundary conditions; **a** and **b** correspond to Eq. 10 with the data obtained by setting Pe $= 20$ and $\varepsilon = 1\%$. (Color figure online)

prior domain expertise. Since the size of the expression library determines the size of the candidate PDE space needed to be explored, it is desirable that the expression library does not include too many components. As the expression library is restricted to necessary components, it is more likely to identify the system model with a short convergence time. The complexity penalty coefficient, λ_3, determines how much complexity users allow for PDEs. This is the most important hyperparameter. When λ_3 is too high, a limited number of PDEs would be tested, generating poor validation results. As λ_3 decreases, the candidate PDE space inflates rapidly, which is prone to overfitting and local minima issues. Thus, it is recommended that users try a high value of λ_3 first, and later decrease it by examining the results as the identification process is evolving. Otherwise, several automatic techniques for λ_3, e.g., a dynamic weight over generation or in the evolutionary processes, can be considered to further minimize work for tuning parameters.

5.2 Advantages and limitations

GPSI has many advantages in system identification. Most of all, an identified system PDE is likely to exhibit a low validation error, even if validation data are “moderately” out of the range of the training data. This is because there is a high chance that an identified PDE captures the current system dynamics. In the cases of the Burgers’ equation and the ADEs, GPSI identified the exact linear and nonlinear terms with their accurate estimates for Re and Pe. The performance of GPSI is robust as well. The likelihood of discovering the reference models (i.e., the Burgers’ equation and the ADE) remains high up to 10% of the noise level in training data. Therefore, GPSI is a powerful system identification method to capture the optimum PDE to model the observed data.

In efforts to capture the dynamics of multi-physics systems, many identification approaches have been developed and investigated. For instance, Gaussian process approaches [13,22] and neural network approaches [4,8,12,23,30] show that they successfully built their models yielding low errors from training and test data by capturing system dynamics. However, these approaches have limitation in offering an easily interpretable mathematical model without the prior knowledge of model structure. In this respect, the interpretability of an identified PDE through GPSI is a considerable advantage, which enhances the understanding of the system dynamics by providing its various plausible descriptions. This could be especially useful in the identification cases where the data measured from an unknown system are the only available resource, i.e., users have no reference models.

In terms of discovering an interpretable model, e.g., a PDE, that governs system dynamics, there is considerable progress being made in sparse approximation methods [25,26] and neural network methods [7,18,27]. However, to capture the complex dynamics of multi-physics systems, the strategy of GPSI offers advantages compared to these methods. Firstly, by constructing candidate models on the basis of random expression trees and mixing them with the stochastic evolutionary processes, infinite combinations of mathematical expressions are able to appear and be tested. Next, the optimization process is fundamentally discrete because the candidate models basically keep being "newly" generated through the evolutionary processes. This discrete process is likely to allow more randomness in the training process and more properly addressing discontinuous basis functions, such as Heaviside step function, than gradient-based optimization methods. For these reasons, the strategy of GPSI seems to be considered more suitable for optimization in an infinite function space.

A distinctive advantage of GPSI is that it remains applicable in identifying a PDE with limited data, although it is fundamentally a data-driven identification method. This advantage is especially significant when measurements are limited. The excitation data and the system description are used to pose initial and boundary conditions. These conditions are the only requirement to compute model responses, some of which are compared with limited measurements. Thus, compared to other data-driven methods, GPSI is an attractive identification method when dealing with limited data. Note that other approaches, such as the ones based on sparse regression (see for example SINDy [26]), have the capacity to discover governing equations from observed data. However, noticeable benefits of GP-based approaches such as the one proposed in this work is on its capability to identify complex and irregular equations forms (e.g., a nonlinear ADE; see Eqs. 9 and 10) and its persistent applicability under limited measurements.

Another advantage of GPSI is that it provides multiple candidate PDE models as many as the number of the population is set, e.g., one hundred of PDEs. While Sect. 4 shows the reference models that are used to generate training data and to confirm the identified PDEs, a situation where there is "no reference model" for data will be a very interesting case. Several identified PDEs could be selected for potential governing equations. Interpreting these PDEs and presenting them with their model probability will provide different perspectives on system physics. In addition, multiple PDEs have potential to be used together for system response probability or for model closure problem.

It is commonly known that GP-based methods have *two major drawbacks*. Firstly, a large amount of hyperparameters need to be tuned for each problem. GPSI resolves this issue by improving its reproducibility, which persists in the changes of the hyperparameters. This reproducibility is attributed to the efficient handling of expression trees during the evolutionary processes with bloat control techniques. Consequently, GPSI is able to address the class of PDE identification problems with the default hyperparameters established in the previous study [11], and further refined in this study. Secondly, the convergence time in a standard GP application can be excessively long to be used. In order to overcome this issue, GPSI utilizes the effective multi-purpose loss function, and employs stochastic sampling. Additionally, parallel computing is implemented to run GPSI simulation for the fitness test. As a result, for the examples investigated in this work, GPSI was performed with a relatively short convergence time, even though the basis function library expands to include many functions (i.e., Heaviside step function, absolute value function, sign function, exponential function, logarithmic function, error function, sine function and cosine function).

6 Summary

This paper presents a novel system identification approach (GPSI) that utilizes stochastic optimization, in conjunction with available data sets, to discover the optimum PDE candidates that match the data and to reveal system physics embedded in the data. GPSI aims to discover the system PDEs by utilizing little prior knowledge about a target system and its measurements (i.e., input/output data). Computational speedup is achieved by incorporating a series of novel steps, such as: (1) a multi-purpose loss function and stochastic sampling into the parallel fitness test as well as (2) bloat control techniques into the evolutionary processes. The results reported in this work show that the algorithm is computationally feasible and allows to identify both linear and nonlinear PDEs.

We illustrate the accuracy and robustness of the proposed approach in two canonical PDEs. The first PDE is the nonlinear Burgers' equation which is of relevance to the fluid dynamics community. The second PDE analyzed is the advection–dispersion equation (also known as the convection-diffusion equation in the heat transfer community) and it is widely used in environmental and chemical engineering to analyze the spatiotemporal dynamics of solute mass transport. Illustrative examples demonstrate that GPSI successfully identified the nonlinear Burgers' equation and linear/nonlinear advection–dispersion equations with significant data noise (i.e., up to 10%). The results demonstrate that new technique is effective and robust to discover PDEs from data without the need for the user to select a parametric phenomenological model. Finally, we show the physics of a given system controls the accuracy of the discovery of the governing equation. We illustrate how the accuracy of the identification process varies according to key dimensionless groups such as the Reynolds and Peclet numbers. The framework provides a promising new approach for identifying governing equations as well as physical laws in parameterized spatiotemporal systems.

Acknowledgements Jinwoo Im acknowledges the Kwanjeong Educational Foundation (17AmB25D) for financial support.

Author contributions All authors contributed to the study conception and design. Material preparation and coding were performed by JI. Analysis of the results were performed by JI, FPJdB and SM. The first draft of the manuscript was written by JI and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Kwanjeong Educational Foundation (17AmB25D) for financial support.

Data availability All the result data presented in the paper are available together with the code in Github (http://github.com/Jinwoousc/GPSI).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Code availibility Genetic Programming for System Identification (GPSI), is written in Python (version 3.7.4) with the *SymPy* library (version 1.7.1) and is available in Github (https://github.com/Jinwoousc/GPSI).

References

1. Blasch, E., Ravela, S., Aved, A.: Handbook of Dynamic Data Driven Applications Systems. Springer, Cham (2018)
2. Bolourchi, A., Masri, S.F., Aldraihem, O.J.: Development and application of computational intelligence approaches for the identification of complex nonlinear systems. Nonlinear Dyn. **79**(2), 765–786 (2015)
3. Bolourchi, A., Masri, S.F., Aldraihem, O.J.: Studies into computational intelligence and evolutionary approaches for model-free identification of hysteretic systems. Comput.-Aided Civ. Infrastruct. Eng. **30**(5), 330–346 (2015)
4. Brewick, P.T., Masri, S.F.: An evaluation of data-driven identification strategies for complex nonlinear dynamic systems. Nonlinear Dyn. **85**(2), 1297–1318 (2016)
5. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Natl. Acad. Sci. **113**(15), 3932–3937 (2016)
6. Chatzi, E.N., Smyth, A.W., Masri, S.F.: Experimental application of on-line parametric identification for nonlinear hysteretic systems with model uncertainty. Struct. Saf. **32**(5), 326–337 (2010)
7. Cranmer, M., Sanchez-Gonzalez, A., Battaglia, P., et al. Discovering symbolic models from deep learning with inductive biases (2020). arXiv:2006.11287
8. Dal Santo, N., Deparis, S., Pegolotti, L.: Data driven approximation of parametrized PDEs by reduced basis and neural networks. J. Comput. Phys. **416**(109), 550 (2020)
9. Holland, J.H., et al.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. MIT press, Cambridge (1992)
10. CARC (Center for Advanced Research Computing): Computation for the work described in this paper was supported by the University of Southern California's CARC (carc.usc.edu) (2021) paper was supported by the University of Southern California's CARC (carc.usc.edu) (2021)
11. Im, J., Rizzo, C.B., de Barros, F.P.J., et al.: Application of genetic programming for model-free identification of nonlinear multi-physics systems. Nonlinear Dyn. **104**, 1–20 (2021)
12. Jagtap, A.D., Kharazmi, E., Karniadakis, G.E.: Conservative physics-informed neural networks on discrete domains for conservation laws: applications to forward and inverse problems. Comput. Methods Appl. Mech. Eng. **365**(113), 028 (2020)
13. Kocijan, J., Girard, A., Banko, B., et al.: Dynamic systems identification with gaussian processes. Math. Comput. Model. Dyn. Syst. **11**(4), 411–424 (2005)
14. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection, vol. 1. MIT Press, Cambridge (1992)
15. Lai, Z., Nagarajaiah, S.: Sparse structural system identification method for nonlinear dynamic systems with hysteresis/inelastic behavior. Mech. Syst. Signal Process. **117**, 813–842 (2019)
16. Liu, W.K., Karniadakis, G., Tang, S., et al.: A computational mechanics special issue on: data-driven modeling and simulation-theory, methods, and applications. Comput. Mech. **64**, 275–277 (2019)
17. Ljung, L.: System Identification. Wiley Encyclopedia of Electrical and Electronics Engineering, pp. 1–19. Wiley, New York (1999)

18. Long, Z., Lu, Y., Dong, B.: PDE-Net 2.0: learning PDEs from data with a numeric-symbolic hybrid deep network. J. Comput. Phys. **399**(108), 925 (2019)
19. Noël, J.P., Esfahani, A.F., Kerschen, G., et al.: A nonlinear state-space approach to hysteresis identification. Mech. Syst. Signal Process. **84**, 171–184 (2017)
20. Panteleev, A.V., Lobanov, A.V.: Mini-batch adaptive random search method for the parametric identification of dynamic systems. Autom. Remote Control **81**(11), 2026–2045 (2020)
21. Quaranta, G., Lacarbonara, W., Masri, S.F.: A review on computational intelligence for identification of nonlinear dynamical systems. Nonlinear Dyn. **99**, 1–53 (2020)
22. Raissi, M., Karniadakis, G.E.: Hidden physics models: machine learning of nonlinear partial differential equations. J. Comput. Phys. **357**, 125–141 (2018)
23. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. **378**, 686–707 (2019)
24. Ribeiro, B., Albrecht, R.F., Dobnikar, A., et al.: Adaptive and natural computing algorithms. In: Proceedings of the International Conference in Coimbra, Portugal. Springer (2005)
25. Rudy, S., Alla, A., Brunton, S.L., et al.: Data-driven identification of parametric partial differential equations. SIAM J. Appl. Dyn. Syst. **18**(2), 643–660 (2019)
26. Rudy, S.H., Brunton, S.L., Proctor, J.L., et al.: Data-driven discovery of partial differential equations. Sci. Adv. **3**(4), e1602614 (2017)
27. Sahoo, S. S., Lampert, C. H., Martius, G.: Learning equations for extrapolation and control (2018). arXiv:1806.07259
28. Schmidt, M., Lipson, H.: Distilling free-form natural laws from experimental data. Science **324**(5923), 81–85 (2009)
29. Schneider, J., Kirkpatrick, S.: Stochastic Optimization. Springer, Berlin (2007)
30. Tartakovsky, A.M., Marrero, C.O., Perdikaris, P., et al.: Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems. Water Resour. Res. **56**(5), e2019WR026731 (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.