nature COMMUNICATIONS

ARTICLE

https://doi.org/10.1038/s41467-021-26434-1 OPEN

Physics-informed learning of governing equations from scarce data

Zhao Chen[1], Yang Liu[2✉] & Hao Sun[3,4,5✉]

Harnessing data to discover the underlying governing laws or equations that describe the behavior of complex physical systems can significantly advance our modeling, simulation and understanding of such systems in various science and engineering disciplines. This work introduces a novel approach called physics-informed neural network with sparse regression to discover governing partial differential equations from scarce and noisy data for nonlinear spatiotemporal systems. In particular, this discovery approach seamlessly integrates the strengths of deep neural networks for rich representation learning, physics embedding, automatic differentiation and sparse regression to approximate the solution of system variables, compute essential derivatives, as well as identify the key derivative terms and parameters that form the structure and explicit expression of the equations. The efficacy and robustness of this method are demonstrated, both numerically and experimentally, on discovering a variety of partial differential equation systems with different levels of data scarcity and noise accounting for different initial/boundary conditions. The resulting computational framework shows the potential for closed-form model discovery in practical applications where large and accurate datasets are intractable to capture.

[1] Department of Civil and Environmental Engineering, Northeastern University, Boston, MA 02115, USA. [2] Department of Mechanical and Industrial Engineering, Northeastern University, Boston, MA 02115, USA. [3] Gaoling School of Artificial Intelligence, Renmin University of China, 100872 Beijing, China. [4] Beijing Key Laboratory of Big Data Management and Analysis Methods, 100872 Beijing, China. [5] Department of Civil and Environmental Engineering, MIT, Cambridge, MA 02139, USA. ✉email: yang1.liu@northeastern.edu; haosun@ruc.edu.cn

Current practices on modeling of complex dynamical systems have been mostly rooted in the use of ordinary and/or partial differential equations (ODEs, PDEs) that govern the system behaviors. These governing equations are conventionally obtained from rigorous first principles such as the conservation laws or knowledge-based phenomenological derivations. However, there remain many real-world complex systems underexplored, whose analytical descriptions are undiscovered and parsimonious closed forms of governing equations are unclear or partially unknown. Luckily, observational datasets become increasingly rich and offer an alternative of distilling the underlying equations from data. Harnessing data to uncover the governing laws or equations can significantly advance and transform our modeling, simulation, and understanding of complex physical systems in various science and engineering disciplines. For example, obtaining mathematical equations that govern the evolution of sea ice from observational data (e.g., satellite remote sensing images) brings distinct benefits for better understanding and predicting the growth, melt and movement of the Arctic ice pack. Distilling an explicit formulation from field sensing data (e.g., Doppler radar recordings) will accelerate more accurate prediction of weather and climate patterns. Recently, advances in machine learning theories, computational capacity, and data availability kindle significant enthusiasm and efforts towards data-driven discovery of physical laws and governing equations[1–13].

Pioneering contributions by Bongard and Lipson[1] and Schmidt and Lipson[2] leveraged stratified symbolic regression and genetic programming to successfully distil the underlying differential equations that govern nonlinear system dynamics from data. However, this elegant approach does not scale up well with the dimensionality of the system, is computationally expensive, and might suffer from overfitting issues. Recently, an impressive breakthrough made by Brunton et al.[5] leads to an innovative sparsity-promoting approach called sparse identification of nonlinear dynamics (SINDy), which selects dominant candidate functions from a high-dimensional nonlinear function space based on sparse regression to uncover parsimonious governing equations, ODEs in particular. The sparsity was achieved by a sequential threshold ridge regression (STRidge) algorithm which recursively determines the sparse solution subjected to hard thresholds[5,6]. Such an approach is capable of balancing the complexity and accuracy of identified models and thus results in parsimony. SINDy has drawn tremendous attention in the past few years, leading to variant algorithms with applications to identify projected low-dimensional surrogate models in the form of first-order ODEs, alternatively with linear embedding[8,10], for a wide range of nonlinear dynamical systems, such as fluid flows[14,15], structural systems[16,17], biological and chemical systems[18–20], active matter[21], predictive control of nonlinear dynamics[22], multi-time-scale systems[23], a predator–prey system[24], and stochastic processes[25], just naming a few among many others. There are also a number of other extensions of SINDy that discover implicit dynamics[18,26], incorporate physics constraints[14], and embed random sampling to improve the robustness to noise for sparse discovery of high-dimensional dynamics[27]. The convergence and error estimate analyses[28] theoretically sustain the family of SINDy approaches.

The sparsity-promoting paradigm has been later extended for the data-driven discovery of spatiotemporal systems governed by PDEs, e.g., the PDE-FIND algorithm[6,7], where the library of candidate functions is augmented by incorporating spatial partial derivative terms. This method has been further investigated or improved to, for example, obtain parametric PDEs from data[29], discover PDEs enhanced by Bayesian inference[30] and gene expression programming[31], identify diffusion and Navier-Stokes equations based on molecular simulation[32], and learn PDEs for biological transport models[33]. Nevertheless, a critical bottleneck of the SINDy framework, especially for the data-driven discovery of PDEs, lies in its strong dependence on both quality and quantity of the measurement data, since numerical differentiation is required to compute the derivatives in order to construct governing equation(s). Especially, the use of finite difference or filtering to calculate derivatives leads to a pivotal challenge that reduces the algorithm robustness. This specially limits the applicability of SINDy in its present form to scenarios given highly incomplete, scarce and noisy data. It is notable that variational system identification[9] shows satisfactory robustness of calculating derivatives based on isogeometric analysis for discovering the weak form of PDEs. However, such an approach doesn't scale down well with respect to the fidelity of available data. Another work[34] shows that weak formulation can significantly improve the discovery robustness against noise, but requires careful design of test functions, which is intractable for high-dimensional spatiotemporal systems.

Automatic differentiation[35] is well-posed to address the above issue, which has been proven successful in physics-informed neural networks (PINN) for forward and inverse analyses of nonlinear PDEs[36–40]. In particular, the deep neural network (DNN) is used to approximate the solution constrained by both the PDE(s) and a small amount of available data. PINN has attracted increasing attention for tackling in a wide range of scientific problems such as fluid flows[39,40], vortex-induced vibrations[41], cardiovascular systems[42], among many others, when the explicit form of PDEs is known. Recently, the important work by Raissi[43] introduced a deep hidden physics model for data-driven modeling of spatiotemporal dynamics based on sparse data, where the unknown underlying physics characterized by possible PDE terms is weakly imposed and implicitly learned by an auxiliary neural network. Nevertheless, the resulting model is still a "black box" and lacks sufficient interpretability since the closed-form governing equations cannot be uncovered. Latest studies[44,45] show the potential of using DNNs and automatic differentiation to obtain closed-form PDEs, from noisy data, in a constrained search space with a predefined library of PDE terms; yet, false-positive identification occurs due to the use of less rigorous sparse regression along with DNN training. In fact, simultaneously optimizing the DNN parameters and sparse PDE coefficients, while accurately enforcing sparsity, is non-trivial and remains a significant challenge in closed-form PDE discovery.

To this end, we leverage these advances and leap beyond to present a novel PINN-SR method (i.e., PINN with sparse regression), possessing salient features of interpretability and generalizability, to discover governing PDEs of nonlinear spatiotemporal systems from scarce and noisy data. Our approach integrates the strengths of DNNs for rich representation learning, automatic differentiation for accurate derivative calculation as well as ℓ_0 sparse regression to tackle the fundamental limitation of existing methods that scale poorly with data noise and scarcity. In particular, the paper involves two methodological contributions: (1) a "root-branch" network, constrained by unified underlying physics, that is capable of dealing with a small number of multi-datasets coming from different initial/boundary conditions, and (2) a simple, yet effective, alternating direction training strategy for optimization of heterogeneous parameters, i.e., DNN trainable parameters and sparse PDE coefficients. The efficacy and robustness of our method are demonstrated on a variety of PDE systems, based on both numerical and experimental datasets.

Results

PINN with sparse regression for PDE discovery. We consider a multi-dimensional spatiotemporal system whose governing

equations can be described by a set of nonlinear, coupled, parameterized PDEs in the general form given by

$$\mathbf{u}_t + \mathcal{F}\left[\mathbf{u}, \mathbf{u}^2, \ldots, \nabla_{\mathbf{x}}\mathbf{u}, \nabla_{\mathbf{x}}^2\mathbf{u}, \nabla_{\mathbf{x}}\mathbf{u}\cdot\mathbf{u}, \ldots; \boldsymbol{\lambda}\right] = \mathbf{p} \quad (1)$$

where $\mathbf{u} = \mathbf{u}(\mathbf{x}, t) \in \mathbb{R}^{1\times n}$ is the multi-dimensional latent solution (dimension $= n$) while $\mathbf{u}_t$ is the first-order time derivative term; $t \in [0, T]$ denotes time and $\mathbf{x} \in \Omega$ specifies the space; $\mathcal{F}[\cdot]$ is a complex nonlinear functional of $\mathbf{u}$ and its spatial derivatives, parameterized by $\boldsymbol{\lambda}$; ∇ is the gradient operator with respect to $\mathbf{x}$; $\mathbf{p} = \mathbf{p}(\mathbf{x}, t)$ is the source term (note that, in many common cases, $\mathbf{p} = \mathbf{0}$ represents no source input to the system). The PDEs are also subjected to initial and boundary conditions (I/BCs), if known, denoted by $\mathcal{I}[\mathbf{x} \in \Omega, t = 0; \mathbf{u}, \mathbf{u}_t] = 0$ and $\mathcal{B}[\mathbf{x} \in \partial\Omega; \mathbf{u}, \nabla_{\mathbf{x}}\mathbf{u}] = 0$. For systems that obey Newton's second law of motion (e.g., $\mathbf{u}_{tt}$ in wave equations), the governing PDEs can be written in a state-space form of Eq. (1) by defining $\mathbf{v} = \{\mathbf{u}\mathbf{u}_t\}$ as the solution variable. Our objective is to find the closed form of $\mathcal{F}[\cdot]$ from available spatiotemporal measurements which are assumed to be incomplete, scarce and noisy commonly seen in real-world applications (e.g., when data capture is very costly or the data itself is sparse in nature). We assume that the physical law is governed by only a few important terms which can be selected from a large-space library of candidate functions, where sparse regression can be applied[5–7]. Inherent in this assumption leads to a reformulation of Eq. (1) in the following (assuming zero or unknown source for simplicity):

$$\mathbf{u}_t = \boldsymbol{\phi}\boldsymbol{\Lambda} \quad (2)$$

Here, $\boldsymbol{\phi} = \boldsymbol{\phi}(\mathbf{u}) \in \mathbb{R}^{1\times s}$ is an extensive library of symbolic functions consisting of many candidate terms, e.g., constant, polynomial, and trigonometric terms with respect to each spatial dimension[6,7], assembled in a row vector given by $\boldsymbol{\phi} = \{1, \mathbf{u}, \mathbf{u}^2, \ldots, \mathbf{u}_x, \mathbf{u}_y, \ldots, \mathbf{u}^3 \odot \mathbf{u}_{xy}, \ldots, \sin(\mathbf{u}), \ldots\}$, where $\odot$ represents the element-wise Hadamard product; s denotes the total number of candidate terms in the library; the subscripts in the context of $\{x, y, z\}$ depict the derivatives; $\boldsymbol{\Lambda} \in \mathbb{R}^{s\times n}$ is the sparse coefficient matrix (only the active candidate terms in $\boldsymbol{\phi}$ have non-zero values), e.g., $\boldsymbol{\Lambda} = [\boldsymbol{\lambda}^u\, \boldsymbol{\lambda}^v\, \boldsymbol{\lambda}^w] \in \mathbb{R}^{s\times 3}$ for $\mathbf{u} = \{u, v, w\}$. If there is an unknown source input, the candidate functions for $\mathbf{p}$ can also be incorporated into $\boldsymbol{\phi}$ for discovery (see Supplementary Note 3.1). Thus, the discovery problem can then be stated as: given the spatiotemporal measurement data $\mathcal{D}_u$, find sparse $\boldsymbol{\Lambda}$ such that Eq. (2) holds.

We present a new PINN-SR paradigm to simultaneously model the system response and identify the parsimonious closed form of the governing PDE(s). The innovative algorithm architecture of this method is shown in Fig. 1, where datasets sampled from two different I/BC scenarios are considered: (1) one dataset from a single I/BC and (2) $r \geq 2$ independent datasets from multiple I/BCs. For the case of single dataset, we interpret the latent solution $\mathbf{u}$ by a DNN (denoted by $\mathcal{N}$), namely, $\mathbf{u}^\theta = \mathbf{u}(\mathbf{x}, t; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represents the DNN trainable parameters including weights and biases, as shown in Fig. 1a. When multiple independent datasets are available, a "root-branch" DNN depicted in Fig. 1b is designed to approximate the latent solutions $\mathbf{u}_i$ $(i = 1, \ldots, r)$ corresponding to different I/BCs, viz., $\mathbf{u}_i^\theta = \mathbf{u}\left(\mathbf{x}, t; \boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(i)}\right)$, where $\boldsymbol{\theta}^{(0)}$ and $\boldsymbol{\theta}^{(i)}$ denote the trainable parameters of the root layers $\mathcal{N}^{(0)}$ and the branch layers $\mathcal{N}^{(i)}$, respectively. Noteworthy, the I/BCs are unnecessarily either known a priori or measured since the measurement data already reflects the specific I/BC (e.g., there exists a one-to-one mapping between the I/BC and the PDE solution). The DNN essentially plays a role as a nonlinear functional to approximate the latent solution with the data loss function $\mathcal{L}_d(\boldsymbol{\theta}; \mathcal{D}_u)$. With automatic differentiation where derivatives on $\mathbf{u}$ are evaluated at machine precision, the library of candidate functions $\boldsymbol{\phi}^\theta$ can be computed from the DNN. For the case of multiple independent datasets, the libraries $\boldsymbol{\phi}^{(i)}$ resulted from the branch nets are concatenated to build $\boldsymbol{\phi}^\theta$ for constructing the unified governing PDE(s). Thus, the sparse representation of the reconstructed PDE(s) can be written in a residual form, namely, $\mathcal{R}^\theta := \mathbf{u}_t^\theta - \boldsymbol{\phi}^\theta\boldsymbol{\Lambda} \to \mathbf{0}$, where $\mathcal{R}^\theta \in \mathbb{R}^{1\times n}$ denotes the PDE residuals. The basic concept is to adapt both the DNN trainable parameters $\boldsymbol{\theta}$ and the PDE coefficients $\boldsymbol{\Lambda}$ such that the neural network can fit the measurement data while satisfying the constraints defined by the underlying PDE(s). The PDE residuals will be evaluated on a large number of collocation points $\mathcal{D}_c = \{\mathbf{x}_i, t_i\}_{i=1}^{N_c}$, randomly sampled in the spatiotemporal space, leading to the residual physics loss function $\mathcal{L}_p(\boldsymbol{\theta}, \boldsymbol{\Lambda}; \mathcal{D}_c)$. When multiple I/BCs are considered, the measurement data and the collocation points will be stacked when calculating the data loss and the physics loss (based on a unified physics residual formulation $\mathcal{R}^\theta \to \mathbf{0}$).

The total loss function for training the overall PINN-SR network is thus composed of the data loss $\mathcal{L}_d$, the residual physics loss $\mathcal{L}_p$ and a regularization term, expressed as

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\Lambda}; \mathcal{D}_u, \mathcal{D}_c) = \mathcal{L}_d(\boldsymbol{\theta}; \mathcal{D}_u) + \alpha\mathcal{L}_p(\boldsymbol{\theta}, \boldsymbol{\Lambda}; \mathcal{D}_c) + \beta \parallel \boldsymbol{\Lambda}\parallel_0 \quad (3)$$

where α is the relative weighting of the residual physics loss function; β is the regularization parameter; $\|\cdot\|_0$ represents the ℓ_0 norm. Optimizing the total loss function can produce a DNN that can not only predict the data-driven full-field system response, but also uncover the parsimonious closed-form PDE(s), i.e., $\{\boldsymbol{\theta}^\star, \boldsymbol{\Lambda}^\star\} := \arg\min_{\{\boldsymbol{\theta},\boldsymbol{\Lambda}\}}\left[\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\Lambda}; \mathcal{D}_u, \mathcal{D}_c)\right]$, where $\{\boldsymbol{\theta}^\star, \boldsymbol{\Lambda}^\star\}$ denote the optimal set of parameters. Noteworthy, the total loss function has an implicit complex form, and thus, directly solving the optimization problem is highly intractable since the ℓ_0 regularization makes this problem *np*-hard. To address this challenge, we present an alternating direction optimization (ADO) algorithm that divides the overall optimization problem into a set of tractable subproblems to sequentially optimize the trainable parameters, as shown in Fig. 1c. Pre-training of PINN-SR is conducted before running the ADO algorithm for discovery, by simply replacing $\|\boldsymbol{\Lambda}\|_0$ in Eq. (3) with $\|\boldsymbol{\Lambda}\|_1$ where brute-force gradient-based optimization for both $\boldsymbol{\theta}$ and $\boldsymbol{\Lambda}$ becomes applicable. The ℓ_1-regularized pre-training can accelerate the convergence of ADO by providing an admissible "initial guess". More detailed formulation and algorithm description are found in Methods and Supplementary Note 1.

The synergy of DNN and sparse regression results in the following outcome: the DNN provides accurate modeling of the latent solution, its derivatives and possible candidate function terms as a basis for constructing the governing PDE(s), while the sparsely represented PDE(s) in turn constraints the DNN modeling and projects correct candidate functions, eventually turning the measured system into closed-form PDE(s).

Discovery of benchmark PDEs with single dataset. We observe the efficacy and robustness of our methodology on a group of canonical PDEs used to represent a wide range of physical systems with nonlinear, periodic and/or chaotic behaviors. In particular, we discover the closed forms of Burgers', Kuramoto–Sivashinsky (KS), nonlinear Schrödinger, Navier–Stokes (NS), and λ-ω reaction–diffusion (RD) equations from scarce and noisy time-series measurements recorded by a number of sensors at fixed locations (data are polluted with Gaussian white noise) from a single I/BC. Results are presented in Table 1, Fig. 2 and Supplementary Note 2.1, which show quite accurate discovery and demonstrate satisfactory performance of the proposed method and its robustness to measurement data scarcity and noise. We also

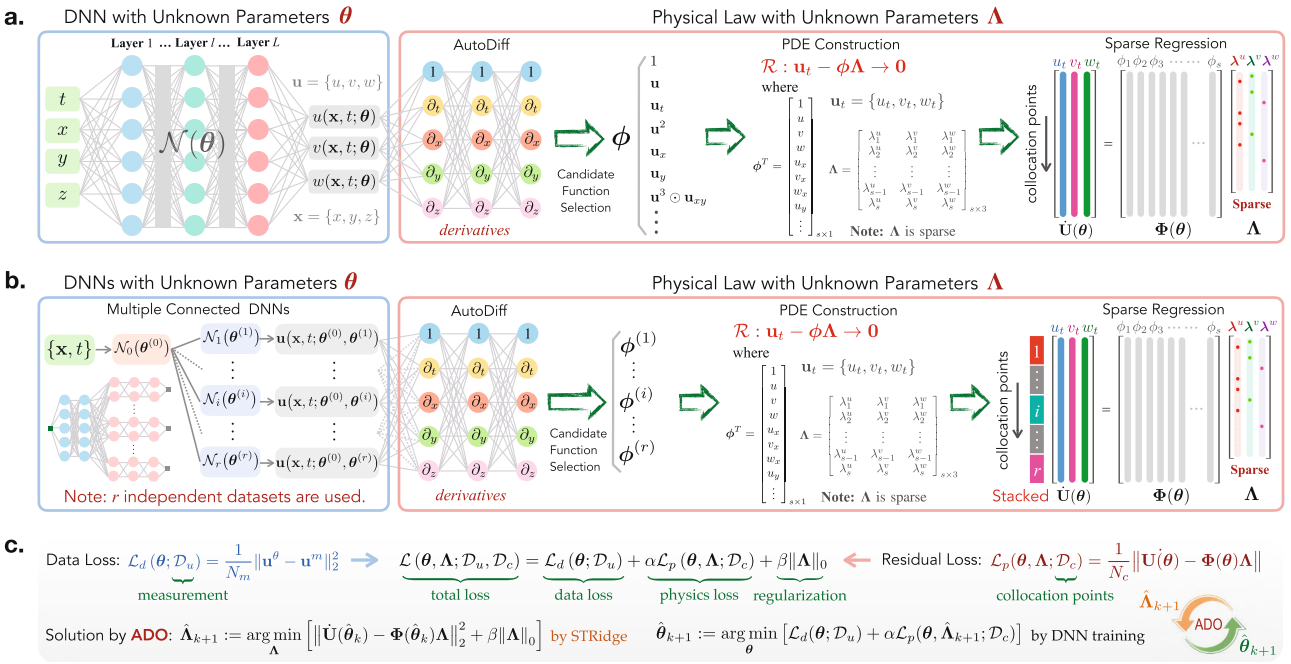

Fig. 1 Schematic architecture of the framework of PINN-SR for data-driven discovery of PDE(s). a the network for one dataset from a single I/BC, **b** the "root-branch" network for $r \geq 2$ independent datasets from multiple I/BCs, and **c** schematic for training the networks based on alternating direction optimization. The network consists of two components: a DNN governed by the trainable parameters $\boldsymbol{\theta}$, which maps the spatiotemporal coordinates $\{\mathbf{x}, t\}$ to the latent solution $\mathbf{u} = \{u, v, w\}$, and the physical law described by a set of nonlinear PDEs, which are formed by the derivative candidate functions $\boldsymbol{\phi}$ parameterized by the unknown sparse coefficients $\boldsymbol{\Lambda}$. Note that, for the case of multiple independent datasets, the libraries $\boldsymbol{\phi}^{(i)}$ are concatenated to build $\boldsymbol{\phi}$ for constructing the unified governing PDE(s). The total loss function $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\Lambda}; \mathcal{D}_u, \mathcal{D}_c)$ is composed of the data loss $\mathcal{L}_d(\boldsymbol{\theta}, \mathcal{D}_u)$, the physics loss $\alpha\mathcal{L}_p(\boldsymbol{\theta}, \boldsymbol{\Lambda}; \mathcal{D}_c)$, and the ℓ_0 regularization term $\beta\|\boldsymbol{\Lambda}\|_0$ that promotes the sparsity. Here, α and β denote the relative weighting of the loss functions, while $\mathcal{D}_u$ and $\mathcal{D}_c$ represent the measurement data and collocation samples respectively. Note that the physics loss, in a residual form, is only evaluated on the spatiotemporal collocation samples. The colored dots in the sparse coefficients matrix (or vector) on the right denote non-zero values. Simultaneous optimization of the unknown parameters $\{\boldsymbol{\theta}, \boldsymbol{\Lambda}\}$ leads to both the trained DNN for inference of the data-driven full-field solution and the discovered parsimonious closed-form PDEs.

Table 1 Summary of the PINN-SR discovery results in the context of accuracy for a range of canonical models.

PDE name	Err. (N-0%)	Err. (N-1%)	Err. (N-10%)	Description of data discretization
Burgers'	0.01 ± 0.01%	0.19 ± 0.11%	0.88 ± 0.03%	$x \in [-8, 8]_{\tilde{n}=256}$, $t \in [0, 10]_{\tilde{n}=101}$, sub. 3.19%
KS	0.07 ± 0.01%	0.61 ± 0.04%	0.94 ± 0.05%	$x \in [0, 100]_{\tilde{n}=1024}$, $t \in [0, 100]_{\tilde{n}=251}$, sub. 12.6%%
Schrödinger	0.09 ± 0.04%	0.65 ± 0.29%	0.08 ± 0.03%	$x \in [-4.5, 4.5]_{\tilde{n}=512}$, $t \in [0, \pi]_{\tilde{n}=501}$, sub. 37.5%
NS	0.66 ± 0.72%	0.86 ± 0.63%	1.22 ± 0.69%	$x \in [0, 9]_{\tilde{n}=449}$, $y \in [-2, 2]_{\tilde{n}=199}$, $t \in [0, 30]_{\tilde{n}=151}$, sub. 0.22%
λ-ω RD	0.07 ± 0.08%	0.25 ± 0.30%	1.84 ± 1.48%	$x, y \in [-10, 10]_{\tilde{n}=256}$, $t \in [0, 10]_{\tilde{n}=201}$, sub. 0.29%

The error is defined as the average relative error of the identified non-zero coefficients w.r.t. the ground truth. The percentage values in the parentheses denote the noise levels (e.g., noise free 0%, 1% and 10%) and the subscript $\tilde{n}$ represents the number of discretization. Our method is also compared with SINDy (the PDE-FIND approach presented in ref. [6]) as illustrated in Supplementary Table 1. It is noted that much less measurement data polluted with a higher level of noise are used in our discovery. Gaussian white noise is added to the synthetic response with the noise level defined as the root-mean-square ratio between the noise and the exact solution.

extensively compare our method with SINDy considering different levels of data scarcity and noise (summarized in Supplementary Note 2.2 and Supplementary Table 1).

Burgers' Equation: We first consider a dissipative system with the dynamics governed by a 1D viscous Burgers' equation expressed as $u_t = -uu_x + \nu u_{xx}$, where ν (equal to 0.1) denotes the diffusion coefficient. The equation describes the decaying stationary viscous shock of a system after a finite period of time, commonly found in simplified fluid mechanics, nonlinear acoustics and gas dynamics. We test the PINN-SR approach on the recorded traveling shock waves from the solution to Burgers' equation subjected to a Gaussian initial condition. In particular, 10 sensors are randomly placed at fixed locations among the 256 spatial grids and record the wave for 101 time steps, leading to 3.19% of the dataset used in ref. [6]. A full description of the dataset, design of the library of candidate functions (16 terms) and model training is given in Supplementary Note 2.1.1. Figure 2a shows the discovered Burgers' equation for a dataset with 10% noise. The evolution of the coefficients $\boldsymbol{\Lambda} \in \mathbb{R}^{16\times 1}$ illustrates robust convergence to the ground truth (error about 0.88%), resulting in accurate discovery. The trained PINN-SR properly reproduces the dynamical response from noisy measurements (e.g., the full-field ℓ_2 prediction error is 1.32%) as shown in Supplementary Fig. 1. The ADO algorithm converges only after the first alternating iteration and shows capacity to recover the correct sparsity pattern of the PDE. We also discover the Burgers' equation with an unknown/unmeasured source $\sin(x)\sin(t)$, given scarce u-measurement with 10% noise. When discovering the underlying governing equation, the source should be considered and reconstructed concurrently. In this case, we incorporate 14 source candidate functions, composed of

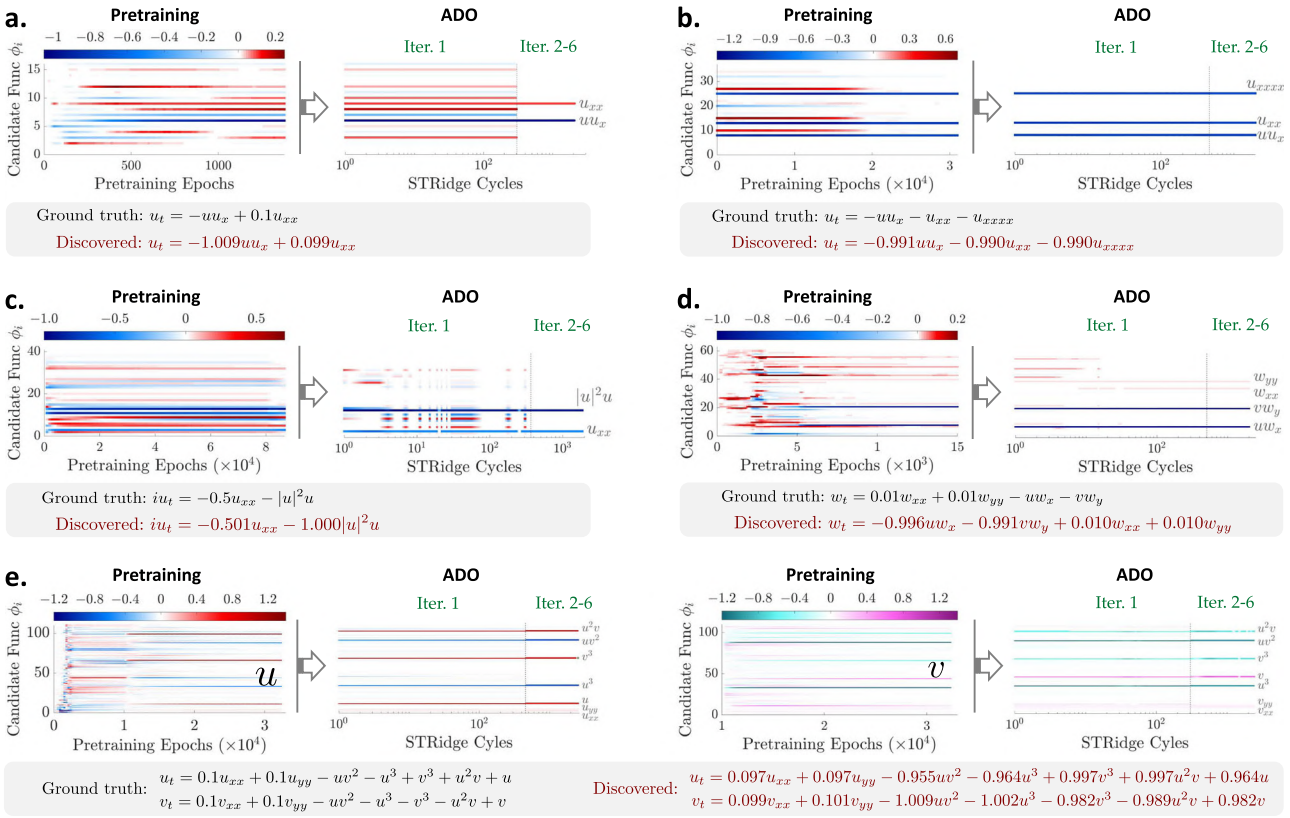

Fig. 2 Discovery of selected benchmark PDEs for sparsely sampled measurement data with 10% noise. a Discovered Burgers' equation: evolution of the sparse coefficients $\boldsymbol{\Lambda} \in \mathbb{R}^{16\times 1}$ for 16 candidate functions $\boldsymbol{\phi} \in \mathbb{R}^{1\times 16}$ used to form the PDE, where the color represents the coefficient value. **b** Discovered KS equation: Evolution of the sparse coefficients $\boldsymbol{\Lambda} \in \mathbb{R}^{36\times 1}$ for 36 candidate functions $\boldsymbol{\phi} \in \mathbb{R}^{1\times 36}$. **c** Discovered nonlinear Schrödinger equation: evolution of the sparse coefficients $\boldsymbol{\Lambda} \in \mathbb{R}^{40\times 1}$ for the candidate functions $\boldsymbol{\phi} \in \mathbb{R}^{1\times 40}$. **d** Discovered NS equation: evolution of the sparse coefficients $\boldsymbol{\Lambda} \in \mathbb{R}^{60\times 1}$ for 60 candidate functions $\boldsymbol{\phi} \in \mathbb{R}^{1\times 60}$. **e** Discovered RD equations: evolution of the sparse coefficients $\boldsymbol{\lambda}^u \in \mathbb{R}^{110\times 1}$ and $\boldsymbol{\lambda}^v \in \mathbb{R}^{110\times 1}$ ($\boldsymbol{\Lambda} = [\boldsymbol{\lambda}^u \boldsymbol{\lambda}^v]$) for 110 candidate functions $\boldsymbol{\phi} \in \mathbb{R}^{1\times 110}$ used to reconstruct the *u*-equation and the *v*-equation, respectively.

$\{\sin(t), \sin(x), \cos(t), \cos(x)\}$ and their combination, into the aforementioned library, resulting in a total of 30 candidate terms for simultaneous discovery of the PDE and reconstruction of the unknown source. The corresponding discovery result is summarized in Supplementary Fig. 12, which includes the discovered equation and source function, the evolution of sparse coefficients $\boldsymbol{\Lambda} \in \mathbb{R}^{30\times 1}$, and the predicted full-field response. It turns out that both PDE and source terms along with their coefficients are well identified. Nevertheless, if the source is very complex with its general expression or form completely unknown, distinct challenges arise when designing the source candidate functions. This may require an extraordinarily large-space library to retain diversifying representations, and thus pose additional computational complexity for accurate discovery of the PDEs. More discussions are presented in Supplementary Note 3.1.

Kuramoto–Sivashinsky (KS) Equation: Another dissipative system with intrinsic instabilities is considered, governed by the 1D Kuramoto-Sivashinsky (KS) equation $u_t = -uu_x - u_{xx} - u_{xxxx}$, where the reverse diffusion term $-u_{xx}$ leads to the disruptive behavior while the fourth-order derivative u_{xxxx} introduces chaotic patterns as shown in Supplementary Fig. 2, making it an ideal test problem for equation discovery. The KS equation is widely used to model the instabilities in laminar flame fronts and dissipative trapped-ion modes among others. We randomly choose 320 points as fixed sensors and record the wave response for 101 time steps, resulting in 12.6% of the dataset used in ref. [6]. A total of 36 candidate functions are employed to construct the underlying PDE. Detail description of this example is found in Supplementary Note 2.1.2. It is notable that the chaotic behavior poses significant challenges in approximating the full-field spatiotemporal derivatives, especially the high-order u_{xxxx}, from poorly measured data for discovery of such a PDE. Existing methods (e.g., the family of SINDy methods[6,7]) eventually fail in this case given very coarse and noisy measurements. Nevertheless, PINN-SR successfully distils the closed form of the KS equation from subsampled sparse data with 10% noise, shown in Fig. 2b. The evolution of the coefficients $\boldsymbol{\Lambda} \in \mathbb{R}^{36\times 1}$ in Fig. 2b illustrates that both the candidate terms and the corresponding coefficients are correctly identified (close to the original parameters; error around 0.94%) within a few ADO iterations. The predicted full-field wave by the trained PINN-SR also coincides with the exact solution at a relative ℓ_2 error of 2.14% (Supplementary Fig. 2).

Nonlinear Schrödinger equation: In the third example, we discover the nonlinear Schrödinger equation, $iu_t = -0.5u_{xx} - |u|^2u$, where u is a complex field variable. This well-known equation is widely used in modeling the propagation of light in nonlinear optical fibers, Bose-Einstein condensates, Langmuir waves in hot plasmas, and so on. We take 37.5% subsamples (e.g., randomly selected from the spatial grids) of the dataset as shown in Table 1 to construct the PDE using 40 candidate functions $\boldsymbol{\phi} \in \mathbb{R}^{1\times 40}$. Since the function is complex-valued, we model separately the real part (u_R) and the imaginary part (u_I) of the solution in the output of the DNN, assemble them to obtain the complex solution $u = u_R + iu_I$, and construct the complex-valued candidate functions for PDE discovery. To avoid complex

gradients in optimization, we use the modulus $|u|$, instead of the ℓ_2 norm shown in Eq. (5), for the residual physics loss $\mathcal{L}_p$ (see Supplementary Note 2.1.3 for more details). Figure 2c shows the discovered Schrödinger equation for the case of 10% noise. The evolution history of the sparse coefficients $\boldsymbol{\Lambda} \in \mathbb{R}^{40\times 1}$ clearly shows the convergence to the actual values (Fig. 2c; error about 0.08%) resulting in accurate closed-form identification of the PDE, while the reconstructed full-field response, for both real and imaginary parts, matches well the exact solution with a slight relative ℓ_2 error of 0.26% (Supplementary Fig. 3).

Navier-Stokes (NS) Equation: We consider a 2D fluid flow passing a circular cylinder with the local rotation dynamics governed by the well-known Navier-Stokes vorticity equation $w_t = -(\mathbf{u}\cdot\nabla)w + \nu\nabla^2 w$, where w is the spatiotemporally variant vorticity, $\mathbf{u} = \{u, v\}$ denotes the fluid velocities, and ν is the kinematic viscosity ($\nu = 0.01$ at Reynolds number 100). We leverage the open simulation data[6] and subsample a dataset of the flow response $\{u, v, w\}$ at 500 spatial locations randomly picked within the indicated region in Supplementary Fig. S4, which record time series for 60 time steps. The resulting dataset is only 10% of that used in ref. [6]. A comprehensive discussion of this example is found in Supplementary Note 2.1.4. Figure 2d summarizes the result of the discovered NS equation for a dataset with 10% noise. It is encouraging that the uncovered PDE expression is almost identical to the ground truth, for both the derivative terms and their coefficients, even under 10% noise corruption. The coefficients $\boldsymbol{\Lambda} \in \mathbb{R}^{60\times 1}$, corresponding to 60 candidate functions $\boldsymbol{\phi} \in \mathbb{R}^{1\times 60}$, converge very quickly to the correct values with precise sparsity right after the first ADO iteration (Fig. 2d). The vorticity patterns and magnitudes are also well predicted as indicated by the snapshot (at $t = 23.8$) shown in Supplementary Fig. 5 (the full-field ℓ_2 error for all snapshots is about 2.58%). This example provides a compelling test case for the proposed PINN-SR approach which is capable of discovering the closed-form NS equation with scarce and noisy data.

Reaction–diffusion (RD) equations: The examples above are mostly low-dimensional models with limited complexity. We herein consider a λ-ω reaction–diffusion (RD) system in a 2D domain with the pattern forming behavior governed by two coupled PDEs: $u_t = 0.1\nabla^2 u + \lambda(g)u - \omega(g)v$ and $v_t = 0.1\nabla^2 v + \omega(g)u + \lambda(g)v$, where u and v are the two field variables, $g = u^2 + v^2$, $\omega = -g^2$, and $\lambda = 1 - g^2$. The RD equations exhibit a wide range of behaviors including wave-like phenomena and self-organized patterns found in chemical and biological systems. The particular RD equations considered here display spiral waves subjected to periodic boundary conditions. Full details on the dataset, selection of candidate functions and hyper-parameter setup of the PINN-SR model are given in Supplementary Note 2.1.5. Fig. 2e shows the evolution of the sparse coefficients $\boldsymbol{\lambda}^u, \boldsymbol{\lambda}^v \in \mathbb{R}^{110\times 1}$ for 110 candidate functions $\boldsymbol{\phi} \in \mathbb{R}^{1\times 110}$, given a dataset with 10% noise. Both the sparse terms and the associated coefficients are precisely identified to form the the closed-form equations (as depicted in Fig. 2e). Due to the complexity of the PDEs and the high dimension, slightly more epochs are required in ADO to retain reliable convergence. The predicted response snapshots (e.g., at $t = 2.95$) by the trained PINN-SR in Supplementary Fig. 6 are close to the ground truth. This example shows especially the great ability and robustness of our method for discovering governing PDEs for high-dimensional systems from highly noisy data.

Discovery of PDEs with multiple independent datasets. To demonstrate the "root-branch" network presented in Fig. 1b for the discovery of PDE(s) based on multiple independent datasets sampled under different I/BCs, we consider (1) the 1D Burgers' equation with light viscosity that exhibits a shock behavior, and (2) a 2D Fitzhugh–Nagumo (FN) type reaction–diffusion system that describes activator-inhibitor neuron activities excited by external stimulus. The measurement data are sparsely sampled (e.g., time series or snapshots) with 10% noise under three different I/BCs. Note that the I/BCs are unnecessarily either measured or known a priori since the measurements already reflect the specific I/BC which holds uniquely one-to-one mapping to the system response. The discovery results are discussed as follows.

Burgers' equation with shock behavior: In this example, we test the previously discussed Burgers' equation with a small diffusion/viscosity parameter ($\nu = 0.01/\pi \approx 0.0032$) based on datasets generated by imposing three different I/BCs. Such a small coefficient creates shock formation in a compact area with sharp gradient (see Fig. 3c) that could challenge the DNN's approximation ability and thus affect the discovery. The three initial and Dirichlet boundary conditions include

$$\text{I/BC 1: } u(x,0) = -\sin(\pi x),\ u(-1,t) = u(1,t) = 0$$
$$\text{I/BC 2: } u(x,0) = \mathcal{G}(x),\ u(-1,t) = u(1,t) = 0$$
$$\text{I/BC 3: } u(x,0) = -x^3,\ u(-1,t) = 1,\ u(1,t) = -1$$

where $\mathcal{G}$ denotes a Gaussian function. Although the measurement datasets for different I/BCs exhibit completely distinct system responses, they obey the same underlying PDE, namely, $u_t = -uu_x + 0.0032u_{xx}$. For all I/BCs, we assume that there are 30 sensors randomly deployed in space ($x \in [-1, 1]$) measuring the wave traveling (e.g., u) for 500 time instants ($t \in [0, 1]$). A denser sensor grid is needed herein, compared with the previous Burgers' example, in order to capture the shock behaviors. Figure 3a shows some of the measurements recorded by six typical sensors under 10% noise. A three-branch network ($r = 3$) shown in Fig. 1b is used for discovery. The full description of the dataset, the library of candidate functions (16 terms) and model training is given in Supplementary Note 2.3.1. Figure 3b depicts the evolution of the coefficients ($\boldsymbol{\Lambda} \in \mathbb{R}^{16\times 1}$) of candidate functions, where the correct terms in the library (uu_x and u_{xx}) are successfully distilled while other redundant terms are eliminated (e.g., hardly thresholded to zero) by ADO. The coefficients of the active terms are accurately identified as well (in particular the small viscosity parameter that leads to shock formation, e.g., 0.0039). The discovered PDE reads $u_t = -1.002uu_x + 0.0032u_{xx}$. Figure 3c, d shows the predicted responses and errors for three I/BC cases, with a stacked full-field ℓ_2 error of 0.65%.

FitzHugh–Nagumo (FN) reaction–diffusion system: We consider the FitzHugh–Nagumo (FN) type reaction–diffusion system, in a 2D domain $\Omega = [0, 150] \times [0, 150]$ with periodic boundary conditions, whose governing equations are expressed by two coupled PDEs: $u_t = \gamma_u\Delta u + u - u^3 - v + \alpha$ and $v_t = \gamma_v\Delta v + \beta(u - v)$. Here, u and v represent two interactive components/matters (e.g., biological), $\gamma_u = 1$ and $\gamma_v = 100$ are diffusion coefficients, $\alpha = 0.01$ and $\beta = 0.25$ are the coefficients for reaction terms, and Δ is the Laplacian operator. The FN equations are commonly used to describe biological neuron activities excited by external stimulus (α), which exhibit an activator-inhibitor system because one equation boosts the production of both components while the other equation dissipates their new growth. Three random fields are taken as initial conditions to generate three independent datasets for discovery, each of which consists of 31 low-resolution snapshots (projected into a 31×31 grid) downsampled from the high-fidelity simulation under a 10% noise condition (see Supplementary Fig. 8). We assume the diffusion terms (Δu and Δv) are known in the PDEs, whose coefficients (γ_u and γ_v) yet need to be identified. A library with 72 candidate

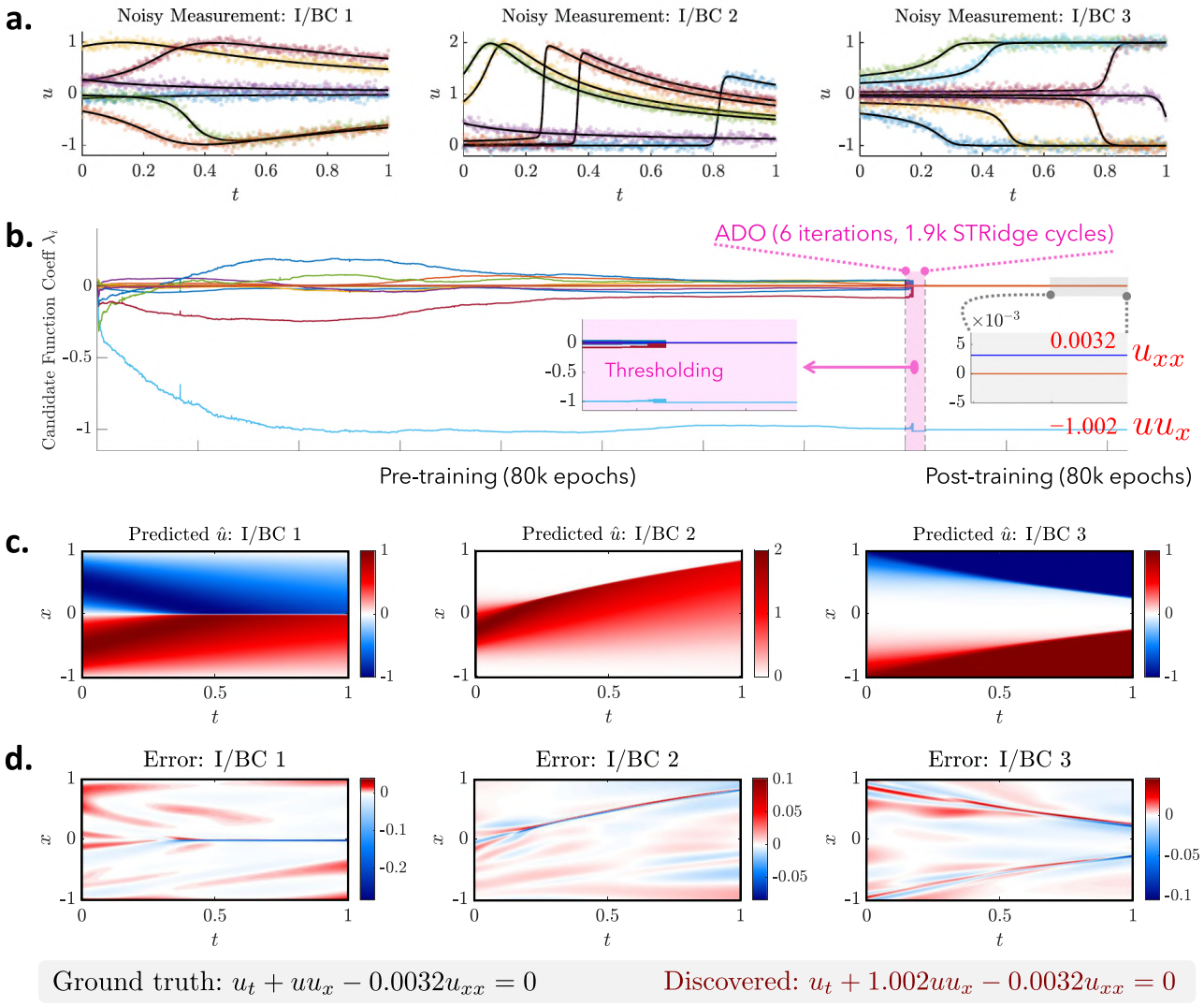

Fig. 3 Discovered Burgers' equation with small viscosity based on datasets sampled under three I/BCs with 10% noise. a Visualization of noisy measurements for the three datasets. Note that there are 30 sensors and only a few are illustrated in this figure. **b** Evolution of the sparse coefficients $\boldsymbol{\Lambda} \in \mathbb{R}^{16\times 1}$ for 16 candidate functions $\boldsymbol{\phi} \in \mathbb{R}^{1\times 16}$ used to construct the PDE, where the color represents the coefficient value. The correct terms (uu_x and u_{xx}) and their coefficients are successfully identified while other redundant terms are eliminated by ADO. **c**, **d** The predicted responses and errors for three I/BC cases. The ground truth is not listed herein since the visualization is almost indistinguishable from the prediction (see Supplementary Fig. 7). The relative full-field ℓ_2 error of the stacked prediction is 0.65%.

functions ($\boldsymbol{\phi} \in \mathbb{R}^{1\times 72}$) is designed for discovery of the coupled PDEs (in particular, the nonlinear reaction terms). Similar to the previous example, a root-branch network shown in Fig. 1b is employed for discovery. More description of the data generation, the specific candidate functions and model training can be found in Supplementary Note 2.3.2. Figure 4a, b depicts the evolution of the sparse coefficients $\boldsymbol{\lambda}^u, \boldsymbol{\lambda}^v \in \mathbb{R}^{72\times 1}$ for 72 candidate functions. The pre-training step provides a redundant projection of the system onto 72 candidates; however, minor candidates are pruned out right after the first ADO iteration. The rest ADO iterations continue to refine all the trainable parameters including $\boldsymbol{\theta}$, $\boldsymbol{\lambda}^u$ and $\boldsymbol{\lambda}^v$. The finally discovered PDEs are listed in Fig. 4 in comparison with the ground truth. It is seen that the form of the PDEs is precisely uncovered with all correct active terms (including the unknown external stimulus in the first equation). The corresponding identified coefficients are generally close to the ground truth except the diffusion coefficient for v (i.e., γ_v) which seems to be a less sensitive parameter according to our test. It should be noted that, given very scarce and noisy measurement datasets in this example, the "root-branch" DNN is faced with challenges to accurately model the solutions with sharp propagating fronts (see Fig. 4c). The less accurate solution approximation by DNN then affects the discovery precision. This issue can be naturally alleviated by increasing the spatiotemporal measurement resolution (even still under fairly large noise pollution, e.g., 10%). Nevertheless, the exact form of the PDEs is successfully discovered in this challenging example, which is deemed more important since the coefficients can be further tuned/calibrated when additional data arrives. Figure 4c shows typical snapshots of the predicted u and v components, the ground truth reference and the error distributions for one unmeasured time instance ($t = 18.72$). The stacked full-field ℓ_2 error is 5.02%.

Experimental discovery of cell migration and proliferation. The last example is placed to demonstrate the proposed approach for discovering a governing PDE that describes cell migration and proliferation, based on the sparse and noisy experimental data collected from in vitro cell migration (scratch) assays[46]. The 1D

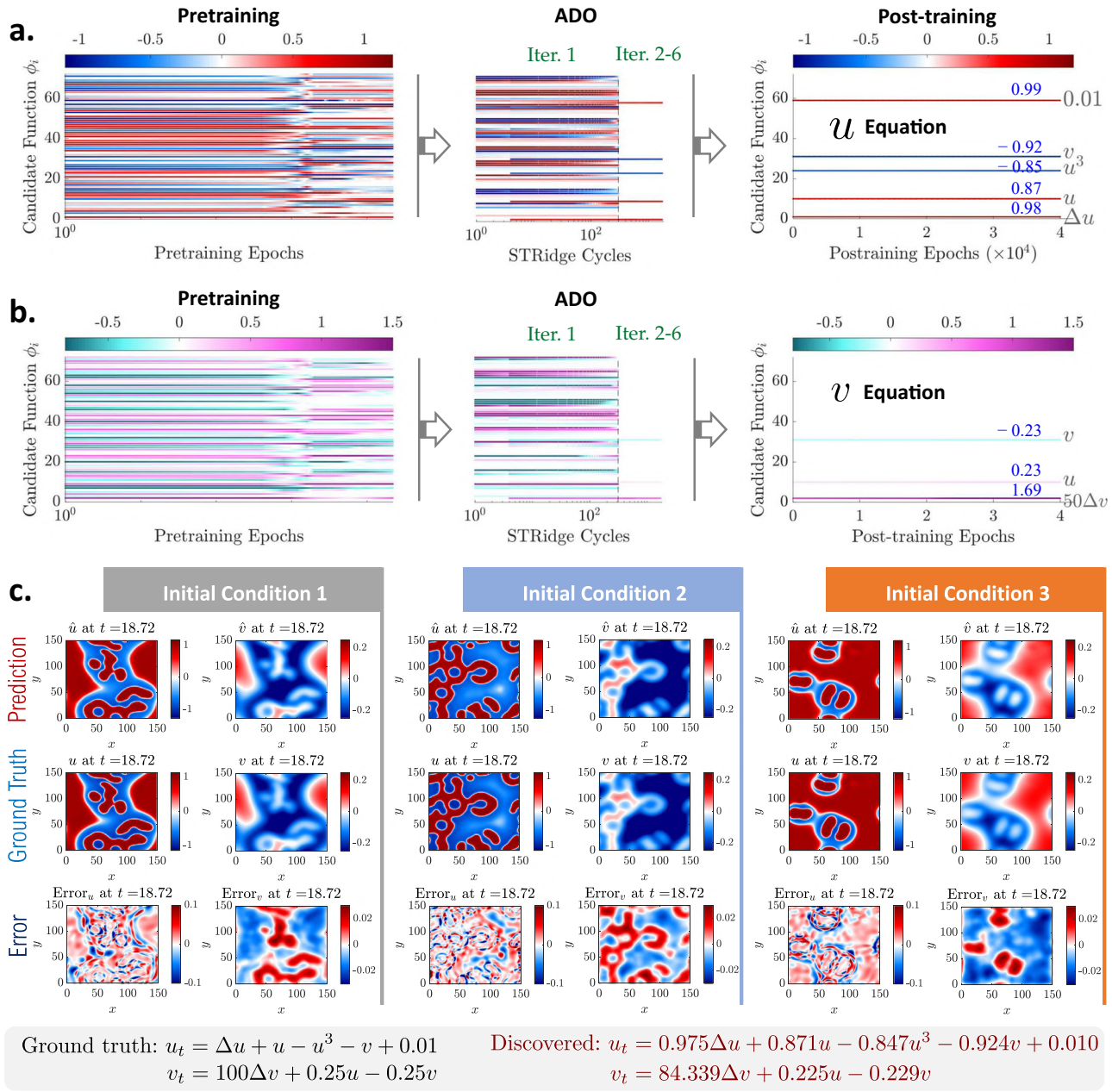

Fig. 4 Discovered Fitzhugh–Nagumo equations based on data sampled under three initial conditions (ICs) with 10% noise. a Evolution of the sparse coefficients $\boldsymbol{\lambda}_u \in \mathbb{R}^{72\times 1}$ for 72 candidate functions used to construct the first PDE (u-equation), where the color represents the coefficient value. **b** Evolution of the sparse coefficients $\boldsymbol{\lambda}_v \in \mathbb{R}^{72\times 1}$ for the second PDE (v-equation). For visualization purpose, we re-scale the identified coefficients of the constant stimulus term "1" in the u-equation by multiplying 100 and of the diffusion term Δv in the v-equation by dividing 50. **c** Snapshots of predicted response, ground truth and error distributions for all three ICs at an unmeasured time instance ($t = 18.72$). The relative ℓ_2 error for the predicted full-field response (stacked u and v) is 5.02%.

cell density distributions at different time instants (0 h, 12 h, 24 h, 36 h, 48 h) were extracted from high-resolution imaging via image segmentation and cell counting. A series of assays were performed under different initial cell densities (e.g., the total number of cells spans from 10,000 to 20,000 following the designated initial distribution in the test well shown in Supplementary Fig. 10 at $t = 0$h). A more detailed description of the experiment setup and datasets can be found in ref. [46]. Our objective herein is to uncover a parsimonious PDE for modeling the dynamics of cell density $\rho(x, t)$. Here, we consider four scenarios with the initial number of cells ranging from 14,000, 16,000, 18,000 to 20,000. We take the mean of the test data from three identically-prepared experimental replicates for each scenario (see Supplementary Fig. 10) to train our model shown in Fig. 1a for PDE discovery. Given our prior knowledge that the cell dynamics can be described by a diffusion (migration) and reaction (proliferation) process, we assume the PDE holds the form of $\rho_t = \gamma\rho_{xx} + \mathcal{F}(\rho)$, where γ is the unknown diffusion coefficient and $\mathcal{F}$ denotes the underlying nonlinear reaction functional. We use 8 additional candidate terms (e.g., $\{1, \rho, \rho^2, \rho^3, \rho_x, \rho\rho_x, \rho^2\rho_x, \rho^3\rho_x\}$) to reconstruct $\mathcal{F}$, whose coefficients are sparse. Hence, the total number of trainable coefficients remains 9 (e.g., $\boldsymbol{\Lambda} \in \mathbb{R}^{9\times 1}$). We believe incorporating our domain-specific prior knowledge is reasonable and should be encouraged in interpretable model discovery,

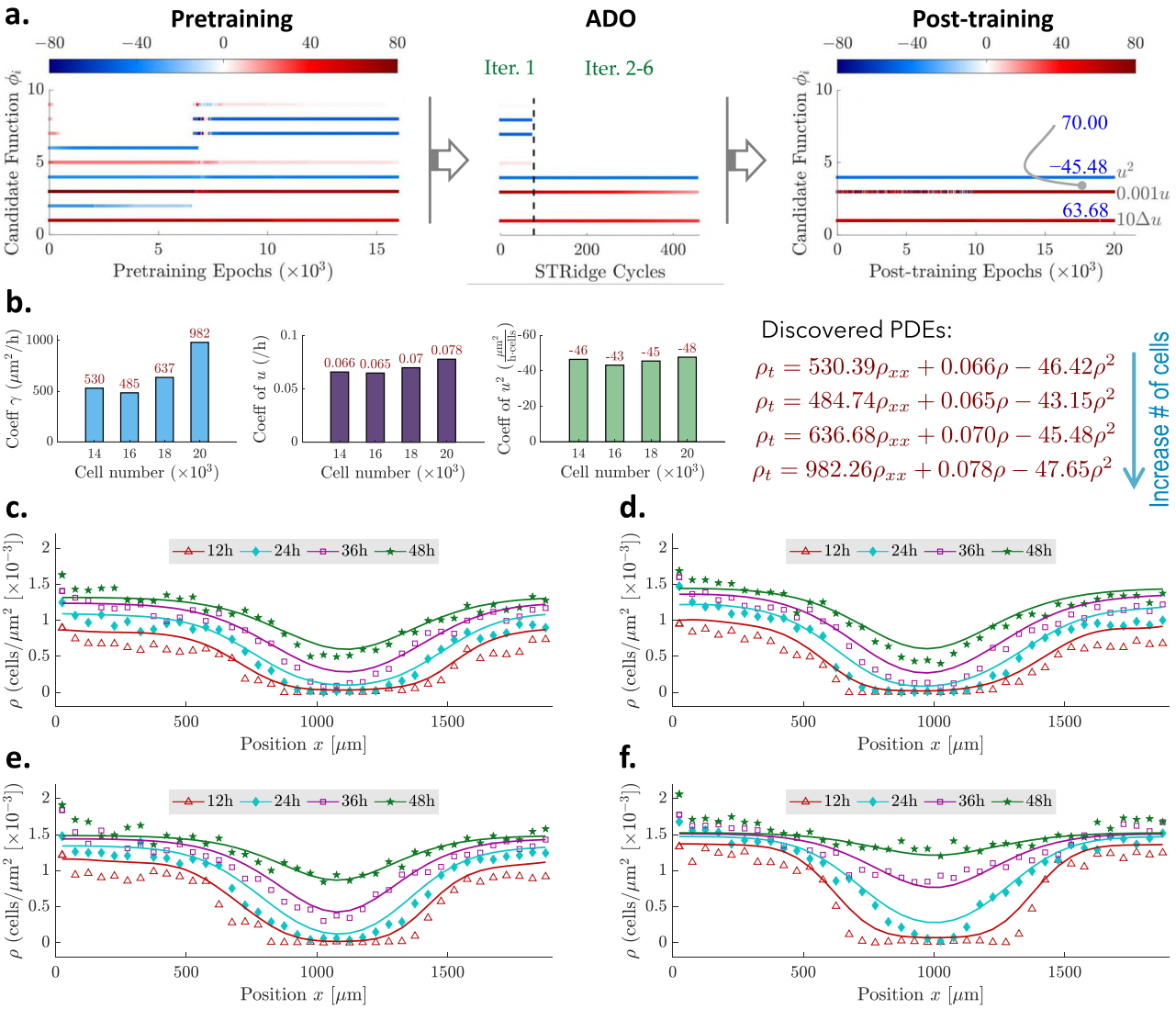

Fig. 5 Discovery result for cell migration and proliferation. a Example evolution of the sparse coefficients $\mathbf{\Lambda} \in \mathbb{R}^{9\times 1}$ for 9 candidate functions used to construct the underlying PDE for the case of 18,000 cells. The diffusion and reaction coefficients for Δu and u are re-scaled for visualization purpose. **b** Discovered active terms $\{\Delta\rho, \rho, \rho^2\}$, their coefficients and the corresponding PDEs for 14,000, 16,000, 18,000 and 20,000 cells, respectively. **c**, **f** Simulated cell densities at different time instants based on the discovered PDEs for 14,000, 16,000, 18,000 and 20,000 cells, respectively, where the measurement at 0h is used as the initial condition while $\rho_x(x=0,t) = \rho_x(x=1900,t) = 0$ is employed as the Neumann boundary condition. The simulation result is represented by solid curves while the markers denote the measurement data.

which could help improve our solution confidence when available data is very sparse and noisy (e.g., in this example). Other details on the PINN-SR model setting and training can be found in Supplementary Note 2.4.

Figure 5a shows the evolution of 9 coefficients for the example case of 18,000 cells, where redundant candidate terms are pruned right after the first ADO iteration via hard thresholding of the corresponding coefficients to zero. The next ADO iterations followed by post-tuning refine the coefficients of active terms for final reconstruction of the PDE. Figure 5b depicts the identified active term coefficients and the corresponding PDEs for different quantities of cells, sharing a unified form of $\rho_t = \gamma\rho_{xx} + \lambda_1\rho + \lambda_2\rho^2$ which exactly matches the famous Fisher-Kolmogorov model[47]. The rates of migration (diffusion) and proliferation (reaction) generally increase along with the number of cells, as seen from the identified coefficients in Fig. 5b. With the discovered PDEs, we simulate/predict the evolution of cell densities at different time instants (12h, 24h, 36h and 48h) presented in Fig. 5c–f, where the measurement at 0h is used as the initial condition while $\rho_x(x=0,t) = \rho_x(x=1900,t) = 0$ is employed as the Neumann boundary condition. The satisfactory agreement between the prediction and the measurement provides a clear validation of our discovered PDEs. It is noted that the extremely scarce and noisy experimental datasets unfortunately pose intractable challenge for any other existing methods (e.g., SINDy[5,6]) to produce a reasonable discovery. This experimental example further demonstrates the strength and capacity of the proposed methodology in regard to handling high level of data scarcity and noise for PDE discovery.

Discussion

In summary, we have presented a novel deep learning method for discovering physical laws, in particular parsimonious closed-form PDE(s), from scarce and noisy data (commonly seen in scientific investigations and real-world applications) for multi-dimensional

nonlinear spatiotemporal systems. This approach combines the strengths of DNNs for rich representation learning of nonlinear functions, automatic differentiation for accurate derivative calculation as well as ℓ_0 sparse regression to tackle the fundamental limitation faced by existing sparsity-promoting methods that scale poorly with respect to data noise and scarcity. The use of collocation points (having no correlation with the measurement data) can render the proposed framework tolerable to scarce and noisy measurements, making the DNN for PDE solution approximation generalizable (see Supplementary Note 3.3). The special network architecture design is able to account for multiple independent datasets sampled under different initial/boundary conditions. An alternating direction optimization strategy is proposed to simultaneously train the DNN and determine the optimal sparse coefficients of selected candidate terms for reconstructing the PDE(s). The synergy of DNN and sparse PDE representation results in the following outcome: the DNN provides accurate modeling of the solution and its derivatives as a basis for constructing the governing equation(s), while the sparsely represented PDE(s) in turn informs and constraints the DNN which makes it generalizable and further enhances the discovery. The overall approach is rooted in a comprehensive integration of bottom-up (data-driven) and top-down (physics-informed) processes for scientific discovery, with fusion of physics-informed deep learning, sparse regression and optimization. We demonstrate this method on a number of dynamical systems exhibiting nonlinear spatiotemporal behaviors (e.g., chaotic, shock, propagating front, etc.) governed by multi-dimensional PDEs based on either single or multiple datasets, numerically or experimentally. Results highlight that the approach is capable of accurately discovering the exact form of the governing equation(s), even in an information-poor space where the multi-dimensional measurements are scarce and noisy. The proposed method also maintains satisfactory robustness against different types of noises (Gaussian and non-Gaussian; see Supplementary Note 3.4) for PDE discovery.

There still remain some potential limitations associated with the present PINN-SR framework for physical law discovery. Firstly, we have to admit that the computational cost of PINN-SR is much higher compared with the state-of-the-art SINDy method, primarily due to the time-consuming DNN training (see Supplementary Note 2.2). However, the critical bottleneck of SINDy lies in its requirement of large high-quality (clean) structured measurement data, owing to its use of numerical differentiation, which poses critical limitation of SINDy in practical applications where data is sparse and noisy (e.g., the experimental data in the cell migration and proliferation example). There is obviously a trade-off between computational efficiency and need of high-quality data. Another limitation is that, although the fully connected DNN used in this work has advantage of analytical approximation of the PDE derivatives via automatic differentiation, directly applying it to model the solution of higher dimensional systems (such as long/short-term response evolution in a 3D domain) results in computational bottleneck and optimization challenges, e.g., due to the need for a vast number of collocation points to maintain satisfactory accuracy. Advances in discrete DNNs with spatiotemporal discretization (e.g., the convolutional long short-term memory network (ConvLSTM)[48] or similar) have the potential to help resolve this challenge, which will be demonstrated in our future work. In addition, the "root-branch" scheme might suffer from scalability issues when a large number of independent datasets sampled under various I/BCs are available, resulting in many branches of the network for PDE solution approximation. The number of DNN trainable variables, the requirement of collocation points for retaining solution accuracy, and thus the computing memory, will grow in general linearly with the number of independent datasets (e.g., $\mathcal{O}(r)$). Nevertheless, this issue can be potentially well resolved by multi-GPU parallelization. Ideally, if the I/BCs are known a priori and can be parameterized under the condition that large and diverse datasets are available, a parametric DNN learning scheme[39] or neural operator learning[49,50] could be developed into the proposed PINN-SR for parametric PDE solution approximation that accounts for different I/BCs. Nevertheless, we emphasize that the assumption of large datasets is out of the scope of our present study, since this requirement is generally hard to meet in equation discovery related applications where data is commonly scarce.

The current version of PINN-SR is inapplicable to the scenario where the PDE coefficients are variant (e.g., time and/or space dependent). However, given PINN's ability of identifying varying coefficients of PDEs[51], PINN-SR can be naturally extended to discover the closed form of PDEs where the varying coefficients are separately modeled and identified. Moreover, PINN is not good at modeling system with chaotic behaviors or sharp propagating wave fronts, primarily due to the way of its solution field approximation with global basis. This limitation is particularly evident when the labeled data is missing (e.g., solving PDEs given I/BCs[52]) or when the model form is unknown (e.g., data-driven modeling with constrained by hidden physics[43]). However, such a limitation can be apparently alleviated, when the labeled data is relatively rich and a clear PDE model is explicitly given (e.g., the library-based model). Nevertheless, the learned full-field response still possesses errors in the propagating wave fronts if the training data is sparse and noisy. Although these errors did not affect much the discovered PDE structure, they result in less accurate identification of PDE coefficients. A network with local basis support might help resolve this issue. Lastly, while PINN-SR relies on a pre-defined library of candidate terms, designing a priori inclusive but not unnecessarily large library remains a difficult task (see more details in Methods). Combining expression trees[53] or symbolic neural networks[54] with PINN and automatic differentiation has the potential to break the limitation of library-based methods for PDE discovery under sparse and noisy data conditions.

Several other aspects (including optimal placement of sensors, convergence history, parametric study on the network size, list of hyper-parameters used in the examples, and other limitations of the method) are further discussed in Supplementary Note 3.5–3.9.

Methods

The innovations of this work are built upon seamless integration of the strengths of deep neural networks for rich representation learning, physics embedding, automatic differentiation and sparse regression to (1) approximate the solution of system variables, (2) compute essential derivatives, as well as (3) identify the key derivative terms and parameters that form the structure and explicit expression of the PDE(s). The technical contributions include: (1) a "root-branch" network, constrained by unified underlying physics, that is capable of dealing with a small number of multi-datasets coming from different I/BCs, and (2) a simple, yet effective, multi-step training strategy for optimization of heterogeneous parameters. The resulting approach is able to deal with scarce/sparse and highly noisy measurement data while accounting for different initial/boundary conditions. The key method components are discussed below.

Network architecture. The proposed network architectures of PINN-SR are shown in Figs. 1a, b that respectively deal with single-I/BC dataset and multiple-I/BC (r) independent datasets. The latent solution $\mathbf{u}$ is interpreted by a dense (fully connected) DNN shown in Fig. 1a, namely, $\mathbf{u}^\theta = \mathbf{u}(\mathbf{x}, t; \boldsymbol{\theta})$, for the case of single dataset, while a "root-branch" dense DNN depicted in Fig. 1b is designed to approximate the latent solutions $\mathbf{u}_i$ $(i = 1, \ldots, r)$ corresponding to different I/BCs, viz., $\mathbf{u}_i^\theta = \mathbf{u}\left(\mathbf{x}, t; \boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(i)}\right)$, for multiple independent datasets. Here, $\boldsymbol{\theta}$'s denote the DNN trainable parameters. The DNNs take the spatiotemporal domain coordinates $\{\mathbf{x}, t\}$ as input followed by multiple fully connected feedforward hidden layers (each layer has dozens of nodes). We use the hyperbolic tangent (tanh) or sine (sin) as the universal activation function thanks to their strength for high-order differentiation and unbiased estimation for both positive and negative values. The sin function is used when the system response exhibits periodic patterns. The output

later is based on linear activation for universal magnitude mapping. When multiple datasets are available, e.g. sampled from different I/BCs, domain coordinates are input to the "root" net (shared hidden layers), followed by r "branch" nets (individual hidden layers) that predict system response corresponding to each I/BC or dataset. The "root" learns the common patterns across all datasets (e.g., the homogeneous part of the solution) while the "branches" learn specific details determined by each I/BC for each independent dataset (e.g., the causality attributed by a specific I/BC). The resulting "root-branch" network, constrained by unified underlying physics, is capable of accounting for different I/BCs. Such an architecture integrates information from different measurements at the expense of larger computational efforts and produces solution approximations satisfying a unified physics (e.g., governing PDE(s)), which essentially strengthens PINN to perform multi-source data-driven modeling. The DNNs essentially play a role as a nonlinear functional to approximate the latent solution.

The DNN is connected to the physical law (reconstruction of PDE(s)) through a automatic differentiator where derivatives on $\mathbf{u}$'s are evaluated at machine precision. The library of candidate functions $\boldsymbol{\phi}^{\boldsymbol{\theta}}$ can be computed from the DNNs. For the case of multiple independent datasets, the libraries $\boldsymbol{\phi}^{(i)}$ resulted from the "branch" nets are concatenated to build one unified $\boldsymbol{\phi}^{\boldsymbol{\theta}}$. If there is unknown source input, the candidate functions for $\mathbf{p}$ can also be incorporated into the library for discovery. The sparse representation of the reconstructed PDE(s) is then expressed in a residual form: $\mathcal{R}^{\boldsymbol{\theta}} := \mathbf{u}_t^{\boldsymbol{\theta}} - \boldsymbol{\phi}^{\boldsymbol{\theta}}\boldsymbol{\Lambda} \to \mathbf{0}$ s.t. $\boldsymbol{\Lambda} \in \mathcal{S}$, where $\mathcal{R}^{\boldsymbol{\theta}} \in \mathbb{R}^{1\times n}$ denotes the PDE residuals, $\mathcal{S}$ represents the sparsity constraint set, and n is the dimension of the system variable (e.g., $\mathbf{u} \in \mathbb{R}^{1\times n}$). Thus, the overall network architecture consists of heterogeneous trainable variables, namely, DNN parameters $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta \times 1}$ and PDE coefficients $\boldsymbol{\Lambda} \in \mathcal{S} \subset \mathbb{R}^{s\times n}$, where n_θ denotes the number of DNN trainable parameters and $n_\theta \gg sn$.

Physics-constrained sparsity-regularized loss function. The physics-constrained sparsity-regularized loss function, expressed in Eq. (3), is composed of three components, the data loss $\mathcal{L}_d$, the residual physics loss $\mathcal{L}_p$ and a sparsity regularization term imposed on $\boldsymbol{\Lambda}$. The data loss function reads

$$\mathcal{L}_d(\boldsymbol{\theta}; \mathcal{D}_u) = \frac{1}{N_m}\left\|\mathbf{u}^{\boldsymbol{\theta}} - \mathbf{u}^m\right\|_2^2 \tag{4}$$

where $\mathbf{u}^m$ is the measurement data, $\mathbf{u}^{\boldsymbol{\theta}}$ is the corresponding DNN-approximated solution, N_m is the total number of data points, and $\|\cdot\|_2$ denotes the Frobenius norm. The responses are stacked when multiple datasets are available, e.g., $\mathbf{u}^m = \{\mathbf{u}_1^m, \ldots, \mathbf{u}_r^m\}$ and $\mathbf{u}^{\boldsymbol{\theta}} = \{\mathbf{u}_1^{\boldsymbol{\theta}}, \ldots, \mathbf{u}_r^{\boldsymbol{\theta}}\}$, where $r \geq 2$, as shown in Fig. 1b. The PDE residuals $\mathcal{R}^{\boldsymbol{\theta}}$ are evaluated on a large number of randomly sampled collocation points $\mathcal{D}_c$, and used to form the residual physics loss function given by

$$\mathcal{L}_p(\boldsymbol{\theta}, \boldsymbol{\Lambda}; \mathcal{D}_c) = \frac{1}{N_c}\left\|\dot{\mathbf{U}}(\boldsymbol{\theta}) - \boldsymbol{\Phi}(\boldsymbol{\theta})\boldsymbol{\Lambda}\right\|_2^2 \tag{5}$$

where $\dot{\mathbf{U}}$ and $\boldsymbol{\Phi}$ denote respectively the discretization of the first-order time derivative term and the library of candidate functions evaluated on the collocation points; N_c is the total number of spatiotemporal collocation points. For the case of multiple datasets, $\dot{\mathbf{U}}$ and $\boldsymbol{\Phi}$ are concatenated over the index of different I/BCs to ensure the identical physical law (in particular, the governing PDE(s)) is imposed, as depicted in Fig. 1b. Note that $\mathcal{L}_d$ ensures that the DNN accurately interpret the latent solution of the PDE(s) via fitting the data, while $\mathcal{L}_p$ generalizes and provides constraints for the DNN through reconstructing the closed form of the PDE(s). The ℓ_0 regularization term in Eq. (3) promotes the sparsity of the coefficients $\boldsymbol{\Lambda}$ for sparse representation of the PDE(s).

Alternating direction optimization. A brute-force training of the network via solving the optimization problem defined in Eq. (3) is highly intractable since the ℓ_0 regularization makes this problem np-hard. Though relaxation of the ℓ_0 term by the less rigorous ℓ_1 regularization improves the well-posedness and enables the optimization in a continuous space, false-positive identification occurs where accurate sparsity of the PDE coefficients cannot be realized[44,45]. To address this challenge, we present an alternating direction optimization (ADO) algorithm that divides the overall optimization problem into a set of tractable subproblems to sequentially optimize $\boldsymbol{\theta}$ and $\boldsymbol{\Lambda}$ within a few alternating iterations (denoted by k), namely,

$$\boldsymbol{\Lambda}_{k+1}^\star := \arg\min_{\boldsymbol{\Lambda}}[\|\dot{\mathbf{U}}(\boldsymbol{\theta}_k^\star) - \boldsymbol{\Phi}(\boldsymbol{\theta}_k^\star)\boldsymbol{\Lambda}\|_2^2 + \beta\|\boldsymbol{\Lambda}\|_0] \tag{6a}$$

$$\boldsymbol{\theta}_{k+1}^\star := \arg\min_{\boldsymbol{\theta}}[\mathcal{L}_d(\boldsymbol{\theta}; \mathcal{D}_u) + \alpha\mathcal{L}_p(\boldsymbol{\theta}, \boldsymbol{\Lambda}_{k+1}^\star; \mathcal{D}_c)] \tag{6b}$$

The fundamental concept of the ADO algorithm is similar to (or can be regarded as a simplified version of) the alternating direction methods of multipliers[55]. In each alternating iteration $k+1$, the sparse PDE coefficients $\boldsymbol{\Lambda}$ in Eq. (6a) are updated (denoted by $\boldsymbol{\Lambda}_{k+1}^\star$) via STRidge (a sequential thresholding regression process that serves as a proxy for ℓ_0 regularization[5,6]), based on the DNN parameters from the previous iteration (e.g., $\boldsymbol{\theta}_k^\star$). The convergence analysis of STRidge can be found in ref. [28]. The DNN parameters $\boldsymbol{\theta}$ in the current iteration are then updated (denoted by $\boldsymbol{\theta}_{k+1}^\star$) through a standard DNN training algorithm (in particular, the combined Adam[56] + L-BFGS[57] optimizer), taking $\boldsymbol{\Lambda}_{k+1}^\star$ as known. Note that a sufficient number of epochs should be used when training the network in order to achieve satisfactory solution accuracy of $\boldsymbol{\theta}_{k+1}^\star$. The alternations between the sub-optimal solutions will lead to a high-quality optimization solution satisfying global convergence. The ADO sequence converges q-linearly (see Theorem 1 below), where q stands for "quotient". Detailed theoretical analysis of generalized alternating optimization can be found in ref. [58]. It is noteworthy that the Adam optimizer plays a role for global search while the L-BFGS optimizer takes responsibility of fine tuning in a local solution region. The learning rate of Adam ranges from 10^{-5} to 10^{-3} in the test examples. The algorithm design of ADO as well as the implementation details and specifications are given in Supplementary Algorithm 1, Algorithm 2 and Note 1.1.

Theorem 1. Let $\boldsymbol{\Theta}^\star = \{\boldsymbol{\theta}^\star, \boldsymbol{\Lambda}^\star\}$ be a local minimizer of the total loss function $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\Lambda}; \mathcal{D}_u, \mathcal{D}_c) : \mathbb{R}^\eta \mapsto \mathbb{R}$ and let $\mathcal{L}$ be strictly convex in a neighborhood $\mathfrak{N}(\boldsymbol{\Theta}^\star, \boldsymbol{\delta})$, where η denotes the number of trainable parameters. We choose $\mathbf{0} < \boldsymbol{\epsilon} \leq \boldsymbol{\delta}$ so that $\mathcal{L}$ is strictly convex on $\mathfrak{N}(\boldsymbol{\Theta}^\star, \boldsymbol{\epsilon})$. If $\mathbf{y} = \{\boldsymbol{\theta}, \boldsymbol{\Lambda}^\star\} \in \mathfrak{N}(\boldsymbol{\Theta}^\star, \boldsymbol{\epsilon})$ and $\boldsymbol{\theta}^*$ locally minimizes $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\Lambda}^\star; \mathcal{D}_u, \mathcal{D}_c)$, then $\boldsymbol{\theta}^*$ is the unique global minimizer. This is also applicable to $\boldsymbol{\Lambda}^*$. For any admissible initial solution $\boldsymbol{\Theta}_0 \in \mathfrak{N}(\boldsymbol{\Theta}^\star, \boldsymbol{\epsilon})$, the corresponding ADO iteration sequence converges to $\boldsymbol{\Theta}^\star$ q-linearly in theory. The actual convergence rate depends on the error propagation in each ADO iteration.

Pre-training of PINN-SR is conducted before running the ADO algorithm for discovery, by simply replacing $\|\boldsymbol{\Lambda}\|_0$ in Eq. (3) with $\|\boldsymbol{\Lambda}\|_1$ where brute-force gradient-based optimization (e.g., Adam + L-BFGS) for both $\boldsymbol{\theta}$ and $\boldsymbol{\Lambda}$ becomes applicable, namely,

$$\{\boldsymbol{\theta}^\star, \boldsymbol{\Lambda}^\star\} = \arg\min_{\{\boldsymbol{\theta}, \boldsymbol{\Lambda}\}}\{\mathcal{L}_d(\boldsymbol{\theta}; \mathcal{D}_u) + \alpha\mathcal{L}_p(\boldsymbol{\theta}, \boldsymbol{\Lambda}; \mathcal{D}_c) + \gamma\|\boldsymbol{\Lambda}\|_1\} \tag{7}$$

where γ denotes the ℓ_1 regularization parameter. The ℓ_1-regularized pre-training can accelerate the convergence of ADO by providing an admissible "initial guess". During pre-training, the DNN learns the physics patterns underlying the sparse and noisy data, weakly constrained by the regression formulation of governing PDEs. Post-training (or post-tuning) is also applicable, which can be applied after the closed form (structure) of the PDE(s) is uncovered. This can be done by training the DNN along with the identification of the discovered non-zero coefficients, viz.,

$$\{\boldsymbol{\theta}^\star, \boldsymbol{\Lambda}^\star\} = \arg\min_{\{\boldsymbol{\theta}, \boldsymbol{\Lambda}\}}\{\mathcal{L}_d(\boldsymbol{\theta}; \mathcal{D}_u) + \alpha\mathcal{L}_p(\boldsymbol{\theta}, \boldsymbol{\Lambda}; \mathcal{D}_c)\} \tag{8}$$

where the initialization of the unknown parameters $\{\boldsymbol{\theta}, \boldsymbol{\Lambda}\}$ can be inherited from the ADO result. The post-training step is completely optional since the ADO method can already provide a high-quality solution as shown in the test examples. Nevertheless, the post-training could add additional discovery accuracy through fine tuning.

It is worthwhile to mention that the underlying intuition of multi-step training has been widely used and justified effective in the deep learning community, in particular, for DNN compression[59,60] (e.g., network pre-training, weights pruning, and post-training). The proposed training strategy is similar to this commonly used procedure. The heuristic justification of the proposed 3-step training strategy reads: the pre-training phyase learns a good PDE solution approximator, ADO uncovers the parsimonious PDE structure, while the post-training stage fine-tunes the coefficients of the discovered PDE structure.

Selection of hyper-parameters. A proper selection of hyper-parameters (e.g., α, β, γ and those required by Supplementary Algorithm 1 and Algorithm 2) guarantees the success of the proposed method for PDE discovery. In this study, the hyper-parameters are selected following the heuristically consistent criteria below.

- α: This hyper-parameter balances the loss contributions from data and physics regularization for network training, which can be generally estimated based on the scale ratio between the measured response $\mathbf{u}^m$ and its temporal derivative $\mathbf{u}_t^m$ (estimated/approximated by finite difference). In particular, the magnitude of α is set to be similar to the deviation ratio between $\mathbf{u}^m$ and $\mathbf{u}_t^m$, namely, $\alpha \sim r_\sigma = \left[\sigma(\mathbf{u}^m)/\sigma(\mathbf{u}_t^m)\right]^2$. Note that, to facilitate the PDE solution approximation highlighting the measurement data, we generally reduce the value of α in the pre-training stage by several times (e.g., 2–10) to relax the physics constraint. In ADO and post-training, the value of α is increased (e.g., $\alpha \sim r_\sigma$) to enhance the discovery of the PDE structure and fine tuning of PDE coefficients. However, we also find exception such as the λ-ω equations, where α in both pre-training and ADO stages should be set greater than the scaled ratio. It is likely due to the high resemblance between $\mathbf{u}$ and $\mathbf{v}$ in the spiral pattern, which can be alleviated if datasets from diverse IB/Cs are included in the measurements. Nevertheless, we have to mention that how to select this hyper-parameter is a common and critical open question in the PINN community.
- β: This hyper-parameter is the coefficient of the ℓ_0 regularizer on $\boldsymbol{\Lambda}$ for the physics regression used in STRidge, which helps adaptively adjust the threshold tolerance in Supplementary Algorithm 1. We propose a Pareto front analysis strategy to estimate the value of β in order to best balance the physics loss and the equation sparsity. We first construct the sparse regression problem (see Eq. (6a)) solved by STRidge, where $\dot{\mathbf{U}}$ and $\boldsymbol{\Phi}$ are

evaluated based on the pre-trained DNN (with the trained network parameters denoted by $\boldsymbol{\theta}_0$). A grid search for β is then performed to obtain the graphical representation of the Pareto set (e.g., $\mathcal{L}_p(\boldsymbol{\theta}_0, \boldsymbol{\Lambda}; \mathcal{D}_c)$ vs. $\|\boldsymbol{\Lambda}\|_0$). The optimal range of β can then be determined (see Supplementary Note 3.8). To avoid scaling issues, we further define $\beta = \kappa \mathcal{L}_p(\boldsymbol{\theta}_0, \boldsymbol{\Lambda}_0; \mathcal{D}_c)$, where κ is an auxiliary scaling variable determined by the Pareto front analysis and $\boldsymbol{\Lambda}_0$ denotes the pre-trained PDE coefficients.

- γ: This hyper-parameter used in pre-training (i.e., coefficient of the ℓ_1 regularizer) is set to be a small value, e.g., 1×10^{-7}. Our parametric study showed that this parameter is less important, which can also be set as zero although a small γ helps weakly promote the coefficient sparsity for PDE candidate terms.
- $n_{\max}$: Based on our extensive tests, it is observed that the correct PDE structure can always be found within the first couple of ADO iterations. Hence, a safe value of 5–10 for the maximum number of ADO iterations ($n_{\max}$) will be sufficient to ensure convergence, e.g., we set $n_{\max}$ as 6 in all examples.

Other hyper-parameters (e.g., number of epochs, number of STRidge iterations, and the threshold increment in STRidge) used to activate Supplementary Algorithm 1 and Algorithm 2 are further discussed in detail in Supplementary Note 1.1 and 3.8.

Initialization of trainable variables. Initiation of the heterogeneous trainable variables remains different. Specifically, the DNN weights are initialized based on Xavier Initialization, while the sparse PDE coefficients are uninformatively initialized either as zero or by uniformly sampling in $[-1, 1]$.

Selection of candidate functions. The library of candidate functions is a significant component in PINN-SR. Designing a a priori inclusive but not unnecessarily large library is a difficult task. On one hand, we prefer to make the candidate library as diverse as possible. On the other hand, balancing the increasing theoretical and computational complexity is crucial for applications. We believe that a specialized library hinged by our domain-specific knowledge and statistical experience can constrain the search space and reduce the complexity of PDE discovery. Although the higher the dimension of the library is, the more likely the exact terms will be uncovered from data. Nevertheless, a highly large-scale library (e.g., the number of components on the order of magnitude of $\geq 10^3$), essentially approximated by the DNN, is very likely to be rank deficient and have poor conditioning, in addition to the growing theoretical complexity and computational burden. Balancing these concerns and finding mathematical principles based on domain-specific knowledge to establish an efficient candidate library remain an open problem. Moreover, failing to include essential candidate functions will lead to false-positive discovery of parsimonious closed form of PDEs, despite that a "best-of-fit" form can be found (see Supplementary Note 3.2). Since the majority of well-known first-order PDEs with respect to time can be represented by linear combination of several active linear/nonlinear terms, we try to include as many as possible commonly seen terms following polynomial basis in this study.

Data availability

All the used datasets in this study are available on GitHub at https://github.com/isds-neu/EQDiscovery.

Code availability

All the source codes to reproduce the results in this study are available on GitHub at https://github.com/isds-neu/EQDiscovery.

Received: 1 July 2021; Accepted: 4 October 2021;
Published online: 21 October 2021

References

1. Bongard, J. & Lipson, H. Automated reverse engineering of nonlinear dynamical systems. *Proc. Natl Acad. Sci.* **104**, 9943–9948 (2007).
2. Schmidt, M. D. & Lipson, H. Distilling free-form natural laws from experimental data. *Science* **324**, 81–5 (2009).
3. Schaeffer, H., Caflisch, R., Hauck, C. D. & Osher, S. Sparse dynamics for partial differential equations. *Proc. Natl Acad. Sci.* **110**, 6634–6639 (2013).
4. Daniels, B. C. & Nemenman, I. Automated adaptive inference of phenomenological dynamical models. *Nat. Commun.* **6**, 8133 (2015).
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**, 3932–3937 (2016).
6. Rudy, S. H., Brunton, S. L., Proctor, J. L. & Kutz, J. N. Data-driven discovery of partial differential equations. *Sci. Adv.* **3**, e1602614 (2017).
7. Schaeffer, H. Learning partial differential equations via data discovery and sparse optimization. *Proc. Roy. Soc. A: Math. Phys. Eng. Sci.* **473**, 20160446 (2017).
8. Lusch, B., Kutz, J. N. & Brunton, S. L. Deep learning for universal linear embeddings of nonlinear dynamics. *Nat. Commun.* **9**, 1–10 (2018).
9. Wang, Z., Huan, X. & Garikipati, K. Variational system identification of the partial differential equations governing the physics of pattern-formation: Inference under varying fidelity and noise. *Comput. Methods Appl. Mech. Eng.* **356**, 44–74 (2019).
10. Champion, K., Lusch, B., Kutz, J. N. & Brunton, S. L. Data-driven discovery of coordinates and governing equations. *Proc. Natl Acad. Sci.* **116**, 22445–22451 (2019).
11. Pfister, N., Bauer, S. & Peters, J. Learning stable and predictive structures in kinetic systems. *Proc. Natl Acad. Sci.* **116**, 25405–25411 (2019).
12. Yuan, Y. et al. Data driven discovery of cyber physical systems. *Nat. Commun.* **10**, 1–9 (2019).
13. Huang, Z. et al. Data-driven automated discovery of variational laws hidden in physical systems. *J. Mech. Phys. Solids* **1371**, 03871 (2020).
14. Loiseau, J.-C. & Brunton, S. L. Constrained sparse galerkin regression. *J. Fluid Mech.* **838**, 42–67 (2018).
15. Loiseau, J.-C., Noack, B. R. & Brunton, S. L. Sparse reduced-order modelling: sensor-based dynamics to full-state estimation. *J. Fluid Mech.* **844**, 459–490 (2018).
16. Lai, Z. & Nagarajaiah, S. Sparse structural system identification method for nonlinear dynamic systems with hysteresis/inelastic behavior. *Mech. Syst. Signal Processing* **117**, 813–842 (2019).
17. Li, S. et al. Discovering time-varying aerodynamics of a prototype bridge by sparse identification of nonlinear dynamical systems. *Phys. Rev. E* **100**, 022220 (2019).
18. Mangan, N. M., Brunton, S. L., Proctor, J. L. & Kutz, J. N. Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Trans. Mol. Biol. Multi-Scale Commun.* **2**, 52–63 (2016).
19. Hoffmann, M., Fröhner, C. & Noé, F. Reactive SINDy: Discovering governing reactions from concentration data. *J. Chem. Phys.* **150**, 025101 (2019).
20. Bhadriraju, B., Narasingam, A. & Kwon, J. S. Machine learning-based adaptive model identification of systems: Application to a chemical process. *Chemical Engineering Research and Design* **152**, 372–383 (2019).
21. Cichos, F., Gustavsson, K., Mehlig, B. & Volpe, G. Machine learning for active matter. *Nat. Mach. Intell.* **2**, 94–103 (2020).
22. Kaiser, E., Kutz, J. N. & Brunton, S. L. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proc. Roy. Soc. A: Math. Phys. Eng. Sci.* **474**, 20180335 (2018).
23. Champion, K. P., Brunton, S. L. & Kutz, J. N. Discovery of nonlinear multiscale systems: Sampling strategies and embeddings. *SIAM J. Appl. Dyn. Syst.* **18**, 312–333 (2019).
24. Dam, M., Brøns, M., Rasmussen, J. J., Naulin, V. & Hesthaven, J. S. Sparse identification of a predator-prey system from simulation data of a convection model. *Phys. Plasmas* **24**, 022310 (2017).
25. Boninsegna, L., Nuske, F. & Clementi, C. Sparse learning of stochastic dynamical equations. *J. Chem. Phys.* **148**, 241723 (2018).
26. Kaheman, K., Kutz, J. N. & Brunton, S. L. SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proc. Roy. Soc. A* **476**, 20200279 (2020).
27. Schaeffer, H., Tran, G. & Ward, R. Extracting sparse high-dimensional dynamics from limited data. *SIAM J. Appl. Math.* **78**, 3279–3295 (2018).
28. Zhang, L. & Schaeffer, H. On the convergence of the SINDy algorithm. *Multiscale Modeling Simul.* **17**, 948–972 (2019).
29. Rudy, S., Alla, A., Brunton, S. L. & Kutz, J. N. Data-driven identification of parametric partial differential equations. *SIAM J. Appl. Dyn. Syst.* **18**, 643–660 (2019).
30. Zhang, S. & Lin, G. Robust data-driven discovery of governing physical laws with error bars. *Proc. Roy. Soc. A: Math. Phys. Eng. Sci.* **474**, 20180305 (2018).
31. Vaddireddy, H., Rasheed, A., Staples, A. E. & San, O. Feature engineering and symbolic regression methods for detecting hidden physics from sparse sensor observation data. *Phys. Fluids* **32**, 015113 (2020).
32. Zhang, J. & Ma, W. Data-driven discovery of governing equations for fluid dynamics based on molecular simulation. *J. Fluid Mech.* **892**, A5 (2020).
33. Lagergren, J. H., Nardini, J. T., Michael Lavigne, G., Rutter, E. M. & Flores, K. B. Learning partial differential equations for biological transport models from noisy spatio-temporal data. *Proc. Roy. Soc. A* **476**, 20190800 (2020).
34. Gurevich, D. R., Reinbold, Patrick, A. K. & Grigoriev, R. O. Robust and optimal sparse regression for nonlinear PDE models. *Chaos: .Interdisciplinary J. Nonlinear Sci.* **29**, 103113 (2019).
35. Baydin, A. G., Pearlmutter, B. A., Radul, A. A. & Siskind, J. M. Automatic differentiation in machine learning: a survey. *J. Mach. Learning Res.* **18**, 5595–5637 (2017).
36. Sirignano, J. & Spiliopoulos, K. DGM: A deep learning algorithm for solving partial differential equations. *J. Comput. Phys.* **375**, 1339–1364 (2018).

37. 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).
38. Yang, Y. & Perdikaris, P. Adversarial uncertainty quantification in physics-informed neural networks. *Journal of Computational Physics* **394**, 136–152 (2019).
39. Sun, L., Gao, H., Pan, S. & Wang, J.-X. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Comput. Methods Appl. Mech. Eng.* **361**, 112732 (2020).
40. Raissi, M., Yazdani, A. & Karniadakis, G. E. Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. *Science* **367**, 1026–1030 (2020).
41. Raissi, M., Wang, Z., Triantafyllou, M. S. & Karniadakis, G. E. Deep learning of vortex-induced vibrations. *J. Fluid Mech.* **861**, 119–137 (2019).
42. Kissas, G. et al. Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4d flow mri data using physics-informed neural networks. *Comput. Methods Appl. Mech. Eng.* **358**, 112623 (2020).
43. Raissi, M. Deep hidden physics models: deep learning of nonlinear partial differential equations. *J. Mach. Learning Res.* **19**, 932–955 (2018).
44. Berg, J. & Nyström, K. Data-driven discovery of PDEs in complex datasets. *J. Comput. Phys.* **384**, 239–252 (2019).
45. Both, G.-J., Choudhury, S., Sens, P. & Kusters, R. Deepmod: deep learning for model discovery in noisy data. *J. Comput. Phys.* **428**, 109985 (2020).
46. Jin, W. et al. Reproducibility of scratch assays is affected by the initial degree of confluence: experiments, modelling and model selection. *J. Theor. Biol.* **390**, 136–145 (2016).
47. Maini, P. K., McElwain, D. L. S. & Leavesley, D. I. Traveling wave model to interpret a wound-healing cell migration assay for human peritoneal mesothelial cells. *Tissue Eng.* **10**, 475–482 (2004).
48. Xingjian, S. et al. In *Advances in Neural Information Processing Systems* 802–810 (2015).
49. Lu, L., Jin, P., Pang, G., Zhang, Z. & Karniadakis, G. E. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nat. Mach. Intell.* **3**, 218–229 (2021).
50. Li, Z. et al. Neural operator: Graph kernel network for partial differential equations. Preprint at https://arxiv.org/abs/2003.03485 (2020).
51. Chen, Y., Lu, L., Karniadakis, G. E. & Dal Negro, L. Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Opt. Express* **28**, 11618–11633 (2020).
52. Wang, S., Yu, X. & Perdikaris, P. When and why PINNs fail to train: a neural tangent kernel perspective. Preprint at https://arxiv.org/abs/2007.14527 (2020).
53. Lample, G. & Charton, F. Deep learning for symbolic mathematics. In *International Conference on Learning Representations* (2019).
54. Sahoo, S., Lampert, C. & Martius, G. S. Learning equations for extrapolation and control. In *Proc. 35th International Conference on Machine Learning* Vol. 80 (2018).
55. Boyd, S., Parikh, N., Chu, E., Peleato, B. & Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® Mach. learning* **3**, 1–122 (2011).
56. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR)* (2015).
57. Byrd, R., Lu, P., Nocedal, J. & Zhu, C. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.* **16**, 1190–1208 (1995).
58. Bezdek, J. C. & Hathaway, R. J. Convergence of alternating optimization. *Neural Parallel Sci. Comput.* **11**, 351–368 (2003).
59. Wen, W., Wu, C., Wang, Y., Chen, Y. & Li, H. Learning structured sparsity in deep neural networks. In *Proc. 30th International Conference on Neural Information Processing Systems* 2082–2090 (2016).
60. Liu, N. et al. Autocompress: an automatic DNN structured pruning framework for ultra-high compression rates. In *Proc. AAAI Conference on Artificial Intelligence* 34, 4876–4883 (2020).

Acknowledgements

The work is supported in part by the Beijing Outstanding Young Scientist Program (No. BJJWZYJH012019100020098) as well as the Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China. We also wish to acknowledge the support in part by the Engineering for Civil Infrastructure program at National Science Foundation under grant CMMI-2013067, the research award from MathWorks, and the Tier 1 Seed Grant Program at Northeastern University.

Author contributions

Y.L. and H.S. contributed to the ideation and design of the research; Z.C. and H.S. performed the research; Z.C., Y.L., and H.S. wrote the paper.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-021-26434-1.

Correspondence and requests for materials should be addressed to Yang Liu or Hao Sun.

Peer review information *Nature Communications* thanks the anonymous reviewer(s) for their contribution to the peer review of this work

Reprints and permission information is available at http://www.nature.com/reprints

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