

ADAM-SINDy: An efficient optimization framework for parameterized nonlinear dynamical system identification

Siva Viknesh ^{1,2} Younes Tatari ^{1,2} Chase Christenson ² and Amirhossein Arzani ^{1,2,*}

¹Department of Mechanical Engineering, *University of Utah*, Salt Lake City, Utah, USA

²Scientific Computing and Imaging Institute, *University of Utah*, Salt Lake City, Utah, USA



(Received 23 September 2025; accepted 24 December 2025; published 15 January 2026)

Identifying nonlinear dynamical systems characterized by nonlinear parameters presents significant challenges in deriving mathematical models that enhance understanding of physical phenomena. Traditional methods, such as Sparse Identification of Nonlinear Dynamics (SINDy) and symbolic regression, can extract governing equations from observational data; however, they also come with distinct advantages and disadvantages. This paper introduces a differentiable optimization methodology within the SINDy framework, termed ADAM-SINDy, which synthesizes the strengths of established approaches by employing the ADAM optimization algorithm. This integration facilitates the simultaneous optimization of nonlinear parameters and coefficients associated with nonlinear candidate functions, enabling efficient and precise parameter estimation without requiring prior knowledge of nonlinear characteristics such as trigonometric frequencies, exponential bandwidths, or polynomial exponents, thereby addressing a key limitation of the classical SINDy framework. Through an integrated global differentiable optimization, ADAM-SINDy dynamically computes all unknown variables in response to system-specific data, resulting in a more adaptive and efficient identification procedure that reduces the sensitivity to the library of candidate functions. The performance of the ADAM-SINDy methodology is demonstrated across a spectrum of dynamical systems, including benchmark coupled nonlinear ordinary differential equations such as oscillators, chaotic fluid flows, reaction kinetics, pharmacokinetics, as well as nonlinear partial differential equations (wildfire transport). The results demonstrate significant improvements in identifying parameterized dynamical systems and underscore the importance of concurrently optimizing all parameters, particularly those characterized by nonlinear parameters. These findings highlight the potential of ADAM-SINDy to extend the applicability of the SINDy framework in addressing more complex challenges in dynamical system identification.

DOI: [10.1103/dwkk-5g2h](https://doi.org/10.1103/dwkk-5g2h)

I. INTRODUCTION

System identification is a crucial aspect of accurately modeling the dynamics of physical, chemical, and biological systems. By providing a mathematical representation of these systems, system identification enables essential predictions, interpretation, optimization, and control strategies across diverse applications, from engineering to environmental science [1]. The process typically involves determining a system's mathematical model from observed data, using methods such as least squares, subspace identification, or frequency domain analysis. The process includes data collection, model structure selection, parameter estimation, validation, and refinement.

Machine learning significantly enhances system identification by offering advanced tools for modeling complex, nonlinear systems and uncovering hidden patterns in data.

Techniques such as supervised learning, reinforcement learning, neural networks, and Gaussian processes facilitate flexible, data-driven modeling. The integration of machine learning with traditional system identification has led to the development of methodologies for linear [2], nonlinear [3], hybrid [4], and parametric/nonparametric [5] dynamical systems. Although linear models are often simpler and computationally efficient, practical systems frequently contain nonlinear features that necessitate more sophisticated models. Addressing the challenge of defining appropriate models for nonlinear systems often requires expert insight. Despite significant progress in earlier research [6,7], nonlinear system identification remains a vibrant and promising research area. The increasing availability of data and advancements in computational methods have led to the development of automated techniques for nonlinear system identification. Recent advancements include methods such as genetic algorithms [8], symbolic regression [9], and deep learning-based algorithms [10–14], which have proven effective in deriving governing nonlinear equations from empirical data. These methods aim to generate meaningful models from data with minimal reliance on prior assumptions, making them particularly valuable when traditional model hypotheses are unavailable. Another important aspect of data-driven system

*Contact author: amir.arzani@sci.utah.edu

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

identification is the interpretability, often achieved by representing dynamics with only a few analytical terms. This characteristic establishes a meaningful connection between system identification studies and explainable artificial intelligence, which emphasizes transparency through either *post-hoc* explanations or inherently interpretable, by-design architectures [15–17].

Symbolic regression provides a unique and interpretable method for system identification, blending classical equation-based approaches with advanced optimization techniques to produce insightful models [18]. Currently, the use of genetic algorithms represents the state of the art in symbolic regression [19,20]. These algorithms effectively mimic natural selection by evolving a population of candidate models over successive generations. Through essential operations such as crossover, mutation, and selection, genetic algorithms efficiently search for equations that optimally capture the system dynamics. This comprehensive process not only facilitates the simultaneous discovery of both the structure and parameters of governing equations but also proves particularly effective in capturing intricate system behaviors. However, it is often susceptible to the bloating effect [21], where candidate solutions become overly complex, hindering interpretability and reducing the generalizability of the identified models. Several strategies have been developed to mitigate bloating and promote compact representations [22–25]. Despite its strengths, symbolic regression faces challenges in capturing chaotic systems due to difficulties in effective phase space sampling, often mitigated through embedding methods [26,27]. Advancements such as diffusion maps [28,29] and local linear embedding [30] address the curse of dimensionality by operating directly on the manifold of the dynamical system, improving the ability to capture complex behaviors. Moreover, innovations like linear basis initialization [31,32], fast function extraction [9], gradient-based optimization [33], invoking system symmetries/manifolds [34], and physics-constrained optimization [35,36] have significantly enhanced the performance of symbolic regression in predicting and identifying dynamical systems. A comprehensive discussion of these developments, along with current approaches for promoting more compact models, can be found in recent reviews [37,38].

As an alternative to symbolic regression, sparsity-promoting techniques that represent nonlinear equations as a sparse combination of basis functions from a comprehensive dictionary have been proposed, offering a computationally efficient alternative [39,40]. The Sparse Identification of Nonlinear Dynamics (SINDy) framework [41] is a widely recognized methodology for discovering interpretable and parsimonious mathematical models that balance accuracy and simplicity. It is based on the principle that many physical dynamical systems can be described by parsimonious models, where only a small set of terms is needed to represent the system's dynamics. By framing the identification problem as a sparse regression problem, SINDy facilitates the discovery of compact, interpretable models while avoiding the computational intensity associated with symbolic regression. This sparsity-driven approach mitigates the risk of overfitting and simplifies the model selection process. Unlike the broad unstructured search employed in symbolic regression, SINDy

operates within a constrained framework where candidate functions are drawn from a predefined library of candidate terms. This structured approach significantly reduces the computational burden. It allows for a more targeted and controlled exploration of the governing equations, ensuring that only the most relevant terms are retained in the final model. This methodology not only enhances computational efficiency but can also improve interpretability. Furthermore, SINDy has demonstrated its effectiveness across various domains, including fluid dynamics [42], plasma dynamics [43], turbulence closures [44], chaotic systems [45], and computational chemistry [46]. Additionally, SINDy has been extended to applications such as nonlinear model predictive control [47], rational functions [48,49], enforcing conservation laws and symmetries [42], interpretable operator learning [50], promoting stability [51], and generalizing for stochastic dynamics [52–54], including Bayesian perspectives [55].

Comparing symbolic regression and the SINDy methodology for dynamical system identification reveals key differences, particularly regarding the treatment of nonlinear parameters. Symbolic regression does not require prior knowledge of system-specific nonlinear parameters, such as Fourier frequencies or exponential growth/decay rates. Instead, it searches through a vast space of mathematical operators and nonlinear functions, such as polynomials, trigonometric functions, and logarithms, to construct an equation that best fits the given data. However, this exploratory nature of symbolic regression often entails considerable computational demands. Moreover, it may yield varying results for identical datasets and hyperparameters, posing challenges for reproducibility. In contrast, the SINDy framework *does* require prior knowledge of nonlinear parameters to guide the optimization process effectively. Moreover, efforts to make these parameters trainable during optimization, such as the Newton-gradient-based approach attempted in Ref. [56], have faced significant challenges due to optimization stiffness and the *nonconvexity* of the loss landscape, ultimately limiting their success. Without such prior nonlinear parameter information, SINDy may struggle to yield an accurate mathematical model, potentially leading to suboptimal performance. Nevertheless, when these parameter values are known or when a distribution of parameter ranges is considered that contains the correct parameter, SINDy offers a considerable computational advantage. It can derive the desired mathematical model in a fraction of the computational time required by symbolic regression. This efficiency is attributed to its formulation of the identification problem as a sparse regression task and a convex optimization problem, which streamlines the model selection process and reduces computational overhead.

While both symbolic regression and SINDy offer distinct advantages and limitations, their comparative strengths highlight the importance of continued advancements in nonlinear system identification. In this work, we introduce a differentiable SINDy optimization framework, referred to as ADAM-SINDy, that leverages the ADAM optimizer and is implemented within the PyTorch environment. This framework directly addresses a key methodological bottleneck in classical SINDy: the inability to optimize nonlinear parameters, such as frequencies and exponential growth/decay rates, embedded within nonlinear candidate functions, while

simultaneously selecting the most appropriate functions from a comprehensive library. By enabling joint optimization, ADAM-SINDy *simultaneously* identifies both the nonlinear parameters and the relevant candidate functions, thereby overcoming the constraint of predefined nonlinear parameter values that limits the classical SINDy formulation. Its effectiveness arises from exploiting the computational graph of gradient-based optimization, which supports efficient backpropagation and dynamic coupling among all trainable parameters. This substantially improves the robustness and precision of system identification, particularly in complex dynamical regimes where traditional SINDy struggles. Although ADAM-SINDy is not intended to surpass symbolic regression in all respects, its primary objective is to address the limitations of the classical SINDy framework, especially in the estimation of nonlinear parameters. By enabling simultaneous nonlinear parameter optimization and candidate-function selection, ADAM-SINDy provides a more efficient and scalable approach to identifying parameterized dynamical systems, thereby broadening the applicability of the SINDy methodology to complex settings. Our present work makes the following key contributions:

1. **Differentiable ADAM-SINDy:** The SINDy framework is augmented with differentiable ADAM optimization framework, termed ADAM-SINDy, which allows the simultaneous estimation of nonlinear parameters while identifying parsimonious and relevant candidate terms. This enhancement addresses a significant limitation of the classical SINDy method, which traditionally requires exact knowledge of the parameters in its library of candidate terms.

2. **Adaptive Hyperparameter Tuning:** The ADAM-SINDy framework integrates concurrent hyperparameter optimization (the sparsity knob) during the identification process, thereby mitigating the need for extensive manual tuning. In contrast, the classical SINDy approach, although capable of producing parsimonious models, is sensitive to the sparsity knob value, necessitating a delicate balance between accuracy and model complexity.

3. **Candidatewise Sparsity Knob:** ADAM-SINDy introduces a candidatewise sparsity knob, adopting a strategy inspired by the Iteratively Reweighted Least-Squares (IRLS) methodology [57–59]. This approach selectively penalizes incorrect terms while retaining the relevant ones throughout the optimization process, employing a concurrent, adaptive individual weighting strategy that is absent in the classical SINDy method. We also demonstrate that this strategy facilitates achieving machine-precision accuracy.

4. **Differential Equation-Based Loss Function:** The ADAM-SINDy optimization methodology allows for the incorporation of differential equation-based loss functions, akin to high-order Tikhonov regularization. This methodology provides an environment that can directly incorporate governing differential equations in certain types of problems into the optimization process, thereby accelerating loss decay and improving convergence rates.

The paper is structured as follows. Section II provides an overview of the classical SINDy methodology, outlining its mathematical foundation and then introducing the ADAM-SINDy methodology, emphasizing its advantages and the implementation of the gradient descent algorithm. This

section also details the master candidate library of nonlinear functions used in evaluating the classical SINDy and ADAM-SINDy frameworks. Section III examines benchmark problems across various applications, including ordinary and partial differential equations, focusing on problems that have nonlinear parameters, thereby challenging the classical SINDy method. Section IV discusses the advantages of the ADAM-SINDy method over the classical approach. Section V presents a conclusion. In the Appendixes, we provide additional details on the algorithm, along with theoretical and empirical justification for the specific optimization formulation used in ADAM-SINDy.

II. METHODS

A. Sparse Identification of Nonlinear Dynamics—SINDy

The SINDy algorithm is an efficient nonlinear system identification approach, grounded in the principle that the governing equations of a nonlinear system can be effectively represented by a sparse combination of appropriate predefined candidate basis functions [41]. The central objective of SINDy is to identify a minimal set of relevant basis functions from an extensive dictionary of candidate functions, leveraging sparsity-promoting techniques to achieve an optimal balance between model complexity and accuracy. Consider the problem of identifying nonlinear dynamical systems described by the differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)), \quad (1)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ denotes the state vector at time t , $\dot{\mathbf{x}}(t)$ represents its time derivative, and $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a nonlinear function governing the system's dynamics. The objective of the SINDy algorithm is to identify the function $\mathbf{f}(\mathbf{x}(t))$ from available data. The process begins by collecting a dataset consisting of measurements of m states at n discrete time steps. The data are organized as follows:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}(t_1)^\top \\ \mathbf{x}(t_2)^\top \\ \vdots \\ \mathbf{x}(t_n)^\top \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1(t_1) & \mathbf{x}_2(t_1) & \cdots & \mathbf{x}_m(t_1) \\ \mathbf{x}_1(t_2) & \mathbf{x}_2(t_2) & \cdots & \mathbf{x}_m(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_1(t_n) & \mathbf{x}_2(t_n) & \cdots & \mathbf{x}_m(t_n) \end{bmatrix}, \quad (2)$$

where each row represents a snapshot of the states at a given time. The matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$ contains the state vectors $\mathbf{x}(t_i)$, each corresponding to the system's state at time t_i . Similarly, the time derivatives of the state vectors, denoted as $\dot{\mathbf{X}} \in \mathbb{R}^{n \times m}$, are represented as follows:

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{x}}(t_1)^\top \\ \dot{\mathbf{x}}(t_2)^\top \\ \vdots \\ \dot{\mathbf{x}}(t_n)^\top \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{x}}_1(t_1) & \dot{\mathbf{x}}_2(t_1) & \cdots & \dot{\mathbf{x}}_m(t_1) \\ \dot{\mathbf{x}}_1(t_2) & \dot{\mathbf{x}}_2(t_2) & \cdots & \dot{\mathbf{x}}_m(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{\mathbf{x}}_1(t_n) & \dot{\mathbf{x}}_2(t_n) & \cdots & \dot{\mathbf{x}}_m(t_n) \end{bmatrix}. \quad (3)$$

This dataset is utilized as input to the system identification problem. A library of candidate functions, denoted $\Theta(\mathbf{X}, \Lambda) \in \mathbb{R}^{n \times p}$, is constructed, encompassing p number of functions, such as polynomials and trigonometric functions. A typical representative library is given below

$$\Theta(\mathbf{X}; \Lambda) = [1 \quad \mathbf{X}^A \quad \cdots \quad \sin(\mathcal{B}\mathbf{X}) \quad \cdots \quad \mathbf{X} \otimes \exp(\mathcal{C}\mathbf{X})], \quad (4)$$

where each column of $\Theta(\mathbf{X}; \Lambda)$ represents a candidate function derived from the state vector $\mathbf{X}$. The set $\Lambda = \{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$ contains the nonlinear parameters of the candidate library, with $\mathcal{A}$ specifying the highest polynomial orders, $\mathcal{B}$ the Fourier frequencies, and $\mathcal{C}$ the exponential exponents, which define the behavior of the candidate functions and are critical for accurately capturing the system's dynamics. Here, the symbol $\otimes$ indicates that columnwise Kronecker product of two candidate functions. Importantly, these nonlinear parameters must be either predetermined prior to the optimization process, which is often impractical in real-world engineering problems, or defined within a specified distribution range, which can result in an extensive candidate matrix and risk missing the precise nonlinear parameter. Accurate determination of these parameters is a crucial step for constructing the candidate functions and effectively identifying the most relevant functions for characterizing the system's dynamics.

The goal in SINDy is to select a subset of these candidate functions, which is achieved by formulating a sparse regression problem and aiming to determine a sparse corresponding coefficient vector $\Xi = [\xi_1, \xi_2, \dots, \xi_m] \in \mathbb{R}^{p \times m}$ indicating the active features from the dictionary $\Theta(\mathbf{X}; \Lambda)$, such that their weighted sum accurately describes $\dot{\mathbf{f}}(\mathbf{x}(t))$. Mathematically, this is expressed as

$$\dot{\mathbf{X}} = \Theta(\mathbf{X}; \Lambda) \Xi. \quad (5)$$

Sparsity-promoting regularization methods, such as Least Absolute Shrinkage and Selection Operator (LASSO) [60,61], are commonly used to enforce sparsity in Ξ , i.e., making most of its entries zero. However, LASSO does not always guarantee the sparsest solution, and alternative methods [62,63] may require prior knowledge of the expected number of nonzero elements in Ξ , which is often unknown. The LASSO augmented with a sequential thresholding approach [41], which iteratively solves least-squares problems and prunes coefficients below a given threshold, offers a more flexible alternative. The sparse regression problem can be formulated as

$$\min_{\Xi} [\|\dot{\mathbf{X}} - \Theta(\mathbf{X}; \Lambda) \Xi\|_2^2 + \lambda \|\Gamma \Xi\|_1], \quad (6)$$

where λ is a regularization parameter that enforces sparsity in the solution, balancing model complexity with accuracy. In the classical SINDy methodology [41], $\Gamma \in \mathbb{R}^{p \times p}$ is typically assumed to be an identity matrix. However, in more generalized settings where Γ is a nonidentity matrix, it serves to penalize individual candidate functions, either remaining fixed throughout the optimization process, as in the adaptive LASSO [64], or being iteratively updated during optimization, as demonstrated in IRLS method [57–59]. Furthermore, if Γ represents a differential equation, such as a gradient or Laplacian, the regularization approach resembles higher-order Tikhonov regularization [65]. Upon solving, the SINDy identified dynamical model is expressed as

$$\dot{\mathbf{x}}(t) = \Theta(\mathbf{x}(t)) \Xi. \quad (7)$$

The SINDy approach provides a parsimonious and interpretable framework for uncovering the underlying dynamics of a system by exploiting sparsity within a known function basis. This methodology is particularly advantageous when

the governing equations are not fully known but are presumed to be sparse within the space of candidate functions.

B. Differentiable ADAM optimization augmented SINDy: ADAM-SINDy

The ADAM-SINDy methodology, an extension of the traditional SINDy framework [41], introduces a differentiable optimization framework and substantial improvements by adopting ADAM optimization in place of the more commonly employed iterative least-squares algorithm for solving the LASSO problem. This methodological innovation significantly addresses key challenges in parameter estimation, while concurrently reducing the dependency on the predefined library of candidate functions, a key limitation that has often constrained the SINDy approaches. Unlike the classical SINDy framework, which heavily relies on the initial selection of candidate parameters, ADAM-SINDy *adaptively* identifies relevant terms from the candidate library (Ξ) while simultaneously optimizing nonlinear parameters (Λ). This adaptive process minimizes reliance on prior assumptions and enhances the capability to discover complex, nonlinearly defined parameters in a more robust manner. Nonlinear parameters, such as frequencies in trigonometric functions, bandwidths in exponential functions (e.g., kernels), and noninteger polynomial exponents, present challenges in the traditional SINDy approach, which requires prior knowledge of these parameters. ADAM-SINDy addresses this limitation by making these parameters optimizable within a global optimization framework. The sparse regression problem in Eq. (6) could be reformulated as a min-max or a joint-minimization optimization problem. The min-max optimization problem is written as

$$\min_{\Xi, \Lambda} \max_{\lambda, \Gamma} [\|\dot{\mathbf{X}} - \Theta(\mathbf{X}; \Lambda) \Xi\|_2^2 + \lambda \|\Gamma \Xi\|_1], \quad (8)$$

where the objective is minimized with respect to the coefficient matrix Ξ and nonlinear parameters Λ , and maximized with respect to sparsity-controlling variables λ and Γ . This min-max approach is similar to the adaptive weighting approach in physics-informed neural networks where the weights for different losses is adaptively updated [66]. Alternatively, the problem can also be expressed as a joint minimization

$$\min_{\Xi, \Lambda, \lambda, \Gamma} [\|\dot{\mathbf{X}} - \Theta(\mathbf{X}; \Lambda) \Xi\|_2^2 + \lambda \|\Gamma \Xi\|_1], \quad (9)$$

where all parameters, including coefficients and sparsity weights, are optimized simultaneously by minimizing the objective function. Mathematically, it may seem that joint minimization would initially drive Γ toward zero, which could remove the ℓ_1 penalty and eliminate sparsity. However, this mechanism allows Γ to approach zero only for the correct candidate functions while saturating at nonzero values for the incorrect ones. As a result, only the incorrect terms are penalized, and the correct terms are spared. In contrast, the min-max formulation is found to be unstable and prone to unbounded growth of Γ due to the convex nature of *both* the ℓ_2 and ℓ_1 norms. This leads to penalization of all terms, regardless of correctness, making the joint minimization the only stable and tractable optimization strategy for the present work.

Appendix B provides both theoretical analysis and empirical evidence explaining why joint minimization is preferred over the min-max approach.

In practice, either λ or Γ is selected to enforce sparsity. Among these, Γ is preferred due to its ability to treat each candidate function individually. Iteratively updating the weighting matrix Γ shares similar strategies with the IRLS technique, which is applied to both constrained [57] and unconstrained problems [59]. In the present work, we adopt a similar strategy by iteratively updating the Γ matrix, but with the distinction of updating the weights using gradients computed through backpropagation. Incorporating stochastic gradient descent and computational graph construction into the ADAM-SINDy methodology within the PyTorch environment offers several notable advantages over the classical SINDy approach:

1. *Simultaneous optimization:* ADAM-SINDy concurrently optimizes hyperparameters and identifies both nonlinear parameters and relevant candidate functions. Unlike SINDy, the proposed method does not require any prior knowledge about the nonlinear parameters in the candidate functions. ADAM-SINDy's integrated optimization framework reduces the need for extensive tuning processes, which are often required in traditional SINDy via trial-and-error methods or sequential optimization steps.

2. *Adaptive weighting:* ADAM-SINDy treats the regularization parameter, Γ , as an optimizable matrix (inspired by the IRLS methodology), where the candidate-specific weights are also optimized over epochs. As a result, the method facilitates an individualized treatment of the candidate functions, thereby enhancing the accuracy of parameter identification and improving the overall robustness of the system identification process. We also show that this approach facilitates achieving machine-precision accuracy.

3. *Higher-order regularization:* Utilizing a computational graph, ADAM-SINDy further enables the opportunity to efficiently apply higher-order Tikhonov regularization, by incorporating a differential equation within the ℓ_1 norm, efficiently computed using autodifferentiation functionality. This method is inspired by the physics-based loss functions used in Physics-Informed Neural Networks (PINN) [67], allowing for the imposition of complex constraints that ensure the identified model aligns closely with the governing physical laws. This augmented regularization framework guarantees that the identified dynamical system remains robust, interpretable, and consistent with the underlying physical principles. Here, we will briefly discuss this aspect and pursue the concept of physics-informed SINDy with ADAM-SINDy in future work.

A detailed description of the ADAM-SINDy algorithm is provided in Appendix A, accompanied by the discussion of both stochastic and full-batch optimization strategies. In Appendix B, we further provide theoretical/empirical justification for why ADAM-SINDy's joint-minimization approach, together with thresholding, works and facilitates machine-precision accuracy.

C. Problem statement

In this work, several benchmark dynamical systems problems are selected to illustrate the efficacy of the

ADAM-SINDy framework and to underscore its advantages relative to the classical SINDy approach. All SINDy methodologies are built upon a common master library that incorporates a diverse range of nonlinear functions, including polynomials, Fourier terms, exponential functions, and nonlinear terms combined with polynomial expressions. This library is formally defined as follows:

$$\Theta = [1 \quad \mathbf{X}^{\mathcal{A}} \quad \sin(\mathcal{B}\mathbf{X}) \quad \cos(\mathcal{C}\mathbf{X}) \quad \exp(\mathcal{D}\mathbf{X}) \quad \mathbf{X} \otimes \sin(\mathcal{E}\mathbf{X}) \quad \mathbf{X} \otimes \cos(\mathcal{F}\mathbf{X}) \quad \mathbf{X} \otimes \exp(\mathcal{G}\mathbf{X})], \quad (10)$$

where $\mathbf{X}$ denotes the input data and $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{F}, \mathcal{G}$ represent the adaptive nonlinear parameters (Λ) that are subject to optimization within the ADAM-SINDy framework. Here, $\mathcal{A}$ is the polynomial order, $\mathcal{B}$ and $\mathcal{C}$ represent Fourier frequencies, $\mathcal{D}$ corresponds to exponential growth/decay, $\mathcal{E}$ and $\mathcal{F}$ are the Fourier frequencies for Fourier terms combined with polynomial terms, and $\mathcal{G}$ represents the exponential growth/decay for the exponential terms combined with polynomial terms. Specifically, $\mathcal{A}$ is chosen to be 1, thereby encompassing polynomial terms up to the first order in the master library for all cases. However, in certain numerical experiments, $\mathcal{A}$ is rendered trainable when it corresponds to a noninteger rational exponent. It is noteworthy that the classical SINDy method relies on precise *a priori* knowledge of these parameters to successfully identify a sparse system. In practice, the classical SINDy method operates under the assumption of a convex loss landscape, where nonlinear parameters are generated by uniform sampling within a predefined range. Consequently, the classical SINDy approach is inherently constrained by its reliance on static parameters, which significantly increases the likelihood of incorrect system identification, particularly when essential functions are absent from the candidate library. As a result, the classical SINDy method frequently struggles to accurately capture the true dynamics of the system, especially in scenarios with nonlinear parameters, wherein the rigidity of predefined parameters—coupled with the absence of precise values—further impedes the effective identification of governing equations.

To investigate these limitations, two variants of the classical SINDy approach are explored: the SINDy-W method and the SINDy-W/o method. The SINDy-W method represents a fortunate scenario (unlikely in practical complex problems) where the precise value of the unknown nonlinear parameter is included within the range of candidate parameters Λ . While such a scenario may hold in controlled or well-defined environments, they are often inadequate in real-world engineering applications, where the underlying parameter values may be unknown or difficult to estimate. Conversely, the SINDy-W/o method assumes no prior knowledge of the parametric values and applies a broad range of distributed parameter values to account for a wider spectrum of possibilities. In our benchmark problems, the range of parameters is selected such that the unknown parameter falls within the range. The SINDy-W/o approach is the more realistic practical version of SINDy; however, it often struggles in cases when the predefined parameter ranges fail to contain the “exact value” required to capture the system's true behavior. This absence of prior knowledge frequently leads to less accurate, overly complex, and nonparsimonious models. Ultimately, these challenges

TABLE I. Hyperparameters utilized in the classical SINDy methods (SINDy-W and SINDy-W/o) and the ADAM-SINDy method across the benchmark problems considered in this study. The nonlinear parameters $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, $\mathcal{E}$, $\mathcal{F}$, $\mathcal{G}$ represent values employed in the library for the SINDy-W and SINDy-W/o approaches, demonstrating performance when the distributed range covers exact (SINDy-W) and approximate (SINDy-W/o) parameters of the system dynamics. Parameters in bold indicate exact values derived from the ground-truth data in the SINDy-W method. For the pharmacokinetics model, $\mathcal{A}$ refers to the exponents used for the state variable and the time-dependent function, enumerated accordingly. Hyperparameters specific to the wildfire PDE example are discussed in the relevant section.

Method	Parameter group	Harmonic oscillator	Van der Pol oscillator	ABC flow	Chemical reaction	Pharmacokinetics model
SINDy-W	λ	0.01	0.0075	0.1	0.025	0.01
	$\mathcal{A}$	1.0	1.0, 2.15	1.0	1.0	$\mathbf{X}^{1.0}, t^{(-0.4, -0.5, -0.55)}$
	$\mathcal{B}, \mathcal{C}, \mathcal{E}, \mathcal{F}$	0.25, 0.5, 0.75 , 1.0	π	$\pi/2, \pi/2.8, \pi/3$ $\pi/4, \pi/4.5, \pi/5$	π	π
	$\mathcal{D}, \mathcal{G}$	1.0	1.0	1.0	1.0, 1.015 , 1.025 , 2	1.0, 2.0
SINDy-W/o	λ	0.01	0.0075	0.1	0.025	0.01
	$\mathcal{A}$	1.0	1.0, 2.5	1.0	1.0	$\mathbf{X}^{1.0}, t^{(-0.4, -0.55)}$
	$\mathcal{B}, \mathcal{C}, \mathcal{E}, \mathcal{F}$	0.2, 0.5, 0.7, 1.0	π	$\pi/2, \pi/2.5, \pi/3.5$ $\pi/4, \pi/4.2, \pi/5$	π	π
	$\mathcal{D}, \mathcal{G}$	1.0	1.0	1.0	1.0, 2.0	1.0, 2.0
ADAM-SINDy	Epochs	50 000	32 000	60 000	40 000	40 000
	Initial learning rate	0.1	0.01	0.1	0.01	0.01
	Step decay epochs	4000	2500	4000	3000	4500

underscore the need for flexibility and adaptability in parameter selection to ensure accurate system identification, motivating the use of the ADAM-SINDy framework, which dynamically adjusts parameter values during optimization and overcomes the limitations of the classical methods' predefined parameter spaces. Here, we adopt the approach established in the classical SINDy method regarding the sparsity-promoting parameters λ and Γ [41]. Specifically, Γ is treated as an identity matrix, with the λ parameter being the sole variable used to control sparsity for the two variants considered: classical SINDy-W and SINDy-W/o methods. Notably, the numerical experiments conducted with the ADAM-SINDy framework indicate that the simultaneous optimization and utilization of both sparsity-promoting knobs is not beneficial.

The ADAM-SINDy methodology is implemented using the PyTorch library, considering both batchwise and full-batch optimization strategies, and employs the ADAM optimizer to facilitate the simultaneous optimization of all unknown variables involved in the sparse regression problem, including the parameters $\mathbf{A}$, the coefficients Ξ , and the sparsity knob Γ , with the parameter λ fixed at a value of 1 and not subject to optimization. In this implementation, the parameter $\mathbf{A} \in \mathbb{R}^{m \times \hat{p}}$ is defined as a rectangular matrix associated with $\hat{p}$ nonlinear candidate parameters corresponding to each of the m state variables. This structure facilitates vectorization, allowing the ADAM optimizer to be invoked only once for all nonlinear parameters rather than for each individual value. Consequently, while $\mathbf{A}$ parameters behave like scalar values algorithmically, it is implemented as a rectangular matrix in vectorized form for improved computational efficiency. Thus, the parameters $\mathbf{A}$ and Ξ are initialized as scalar matrices with all values of 1. Additionally, the diagonal matrix Γ , with trainable diagonal entries, is initialized using a Gaussian distribution with a standard deviation of 1. To enhance convergence during the optimization process, a step decay variant for the dynamic learning rate is employed, systematically reducing the learning rate with a decay rate of 0.5 across all numerical experiments conducted in this research. Furthermore, during

the sequential thresholding step, a tolerance value of 5×10^{-3} is utilized for ϵ , effectively eliminating small Ξ and $\mathbf{A}$ values. A comprehensive overview of the hyperparameters for both the classical SINDy and ADAM-SINDy methods is provided in Table I, detailing the parametric values chosen for the comparative analysis.

III. NUMERICAL EXPERIMENTS

This section details the benchmark problems used in this study, encompassing five systems of nonlinear ordinary differential equations (ODEs) and a sixth case involving a nonlinear partial differential equation (PDE). The final test case highlights the advantages of integrating trainable sparsifying hyperparameters, namely, λ and Γ , along with a sensitivity analysis. Each benchmark problem features a nonlinear parameter that poses challenges for the classical SINDy approach, and the SINDy-W method successfully identifies the governing dynamics for all cases as expected. These results underscore the challenges often encountered in practical SINDy applications (SINDy-W/o) when the correct nonlinear parameter is absent from the candidate library.

Given the two objective formulations considered in this work—joint-minimization and min-max optimization—a comparative analysis presented in Appendix B demonstrates that the min-max approach is unsuitable due to the absence of a saddle point. This deficiency hinders convergence and requires further enhancements, such as weight decay, making the approach ultimately less robust. However, theoretical and empirical evidence supports the joint-minimization formulation as a viable strategy for the identification of nonlinear dynamical systems. Thus, the joint-minimization approach has been adopted for the present work. In addition, the results from ADAM-SINDy using batchwise optimization are excluded, as this method failed to produce accurate outcomes for complex dynamical systems such as the pharmacokinetic model and wildfire PDE identification, where a batch size of approximately 70%–80% of

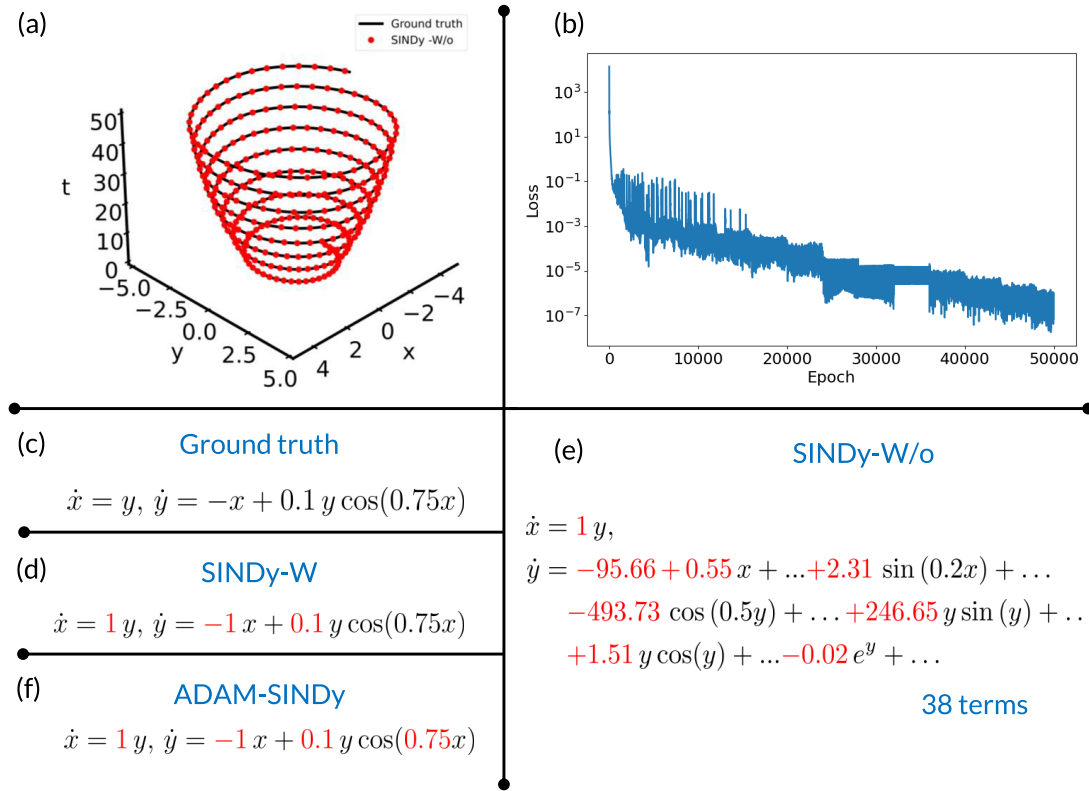


FIG. 1. Harmonic oscillator example. (a) Temporal trajectory comparison showing both the ground-truth and the SINDy-W/o method. (b) Loss plot illustrating the convergence behavior of the ADAM-SINDy method. A comparison of the governing equations is provided between the (c) ground truth and those identified by the (d) SINDy-W, (e) SINDy-W/o, and (f) ADAM-SINDy approaches. The red-highlighted numerical values indicate the optimized parameters obtained through these methodologies. For the SINDy-W/o method, a total of 38 terms were identified for $\dot{y}$, reflecting the complexity of the resulting dynamical system. Ground-truth data are generated up to $t = 50$ with the initial conditions $x_0 = -2$ and $y_0 = 0$, using a time step of 0.01.

the entire dataset was required, making it nearly equivalent to full-batch optimization. Consequently, only the full-batch ADAM-SINDy results are presented here, as this approach yielded reliable and accurate solutions for all test cases.

A. Harmonic oscillator—Fourier frequency

First, we investigate Fourier frequencies within the candidate matrix by considering the harmonic oscillator as a benchmark problem. This system models a particle experiencing a restoring force proportional to its displacement from equilibrium. The dynamics of the system are governed by the following set of ODEs:

$$\begin{aligned}\dot{x} &= ay, \\ \dot{y} &= -bx + cy \cos(dx),\end{aligned}\quad (11)$$

where $x(t)$ represents the displacement and $y(t)$ represents the velocity. The term bx corresponds to the linear restoring force as described by Hooke's law, while the cosine term introduces nonlinear effects through the frequency parameter d , which captures external factors such as damping or external driving forces. This example illustrates the effectiveness of the ADAM-SINDy framework in accurately computing the Fourier frequency d , as well as identifying other system coefficients, without the need for correct predefined parameter

distributions, a limitation inherent to the classical SINDy approach. In this case, the cosine frequency d is set to 0.75, a noninteger value, while the other parameters are chosen as $a = 1$, $b = 1$, and $c = 0.1$, to represent the ground-truth dynamics of the system, which is generated up to $t = 50$ with the initial condition of $x_0 = -2$ and $y_0 = 0$, for a time step of 0.01.

The results, depicted in Fig. 1, provide a comparative analysis of the governing equations corresponding to the ground-truth system and those identified by the SINDy-W, SINDy-W/o, and ADAM-SINDy methodologies. While the classical SINDy-W/o method yielded a trajectory that closely matches the ground-truth data, the identified system dynamics were unnecessarily complex and deviated from the true governing equations. In contrast, the ADAM-SINDy framework accurately computed the Fourier frequency and demonstrated a clear advantage by recovering the system's dynamics with greater accuracy, even in the presence of nonlinearity, as evidenced by quantitative analysis. SINDy-W also captures the true dynamical system, albeit by assuming precise knowledge of the frequency d in its library. It should be noted that in all figures, the trajectory for SINDy-W and ADAM-SINDy methods is not shown as these methods identify the exact governing equations; therefore, the trajectories are identical to the ground truth.

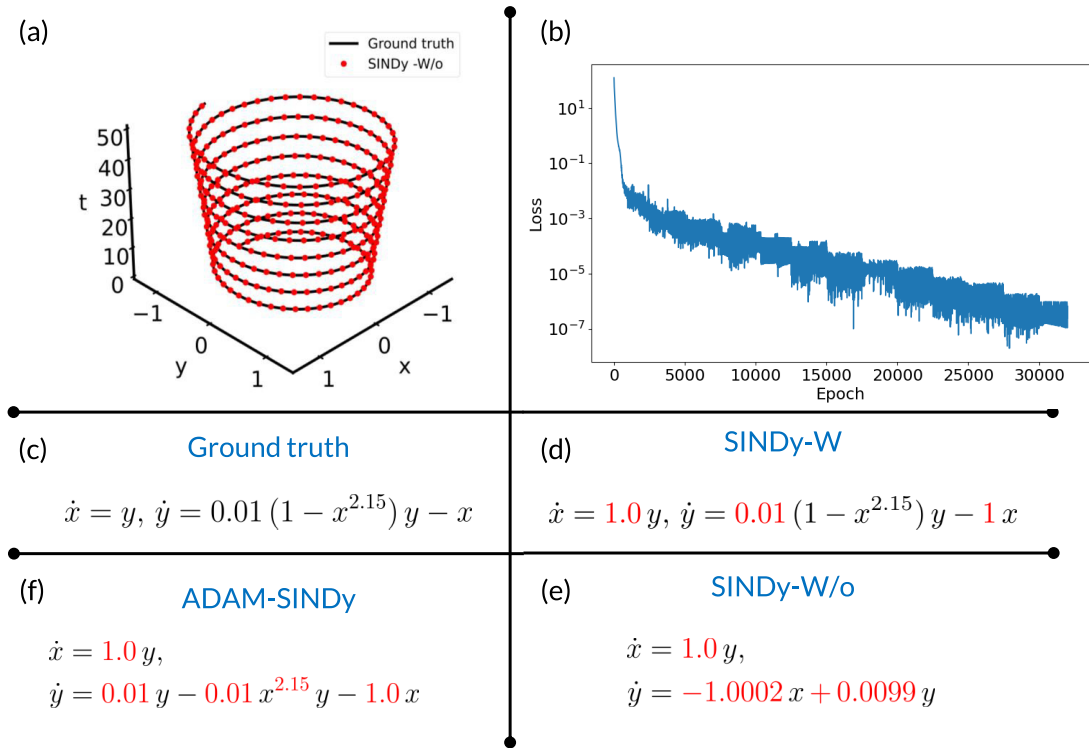


FIG. 2. Van der Pol oscillator example. (a) Temporal trajectory comparison showing both the ground truth and the SINDy-W/o method. (b) Loss plot illustrating the convergence behavior of the ADAM-SINDy method. A comparison of the governing equations is provided between the (c) ground truth and those identified by the (d) SINDy-W, (e) SINDy-W/o, and (f) ADAM-SINDy approaches. The red-highlighted numerical values indicate the optimized parameters obtained through these methodologies. Ground-truth data are generated up to $t = 200$ with the initial conditions $x_0 = 0$ and $y_0 = -1$, using a time step of 0.01.

B. Van der Pol oscillator—Fractional Polynomial Exponent

This example focuses on the computation of fractional polynomial exponents within the candidate matrix by employing the Van der Pol oscillator, a standard example of a nonlinear dynamical system characterized by self-sustained oscillations. The system is governed by the following set of first-order differential equations:

$$\begin{aligned}\dot{x} &= y, \\ \dot{y} &= \mu(1 - x^b)y - x,\end{aligned}\tag{12}$$

where the parameter μ introduces nonlinear damping, thereby influencing both the amplitude and frequency of the oscillations. For small values of μ , the system behaves similarly to a simple harmonic oscillator. However, as μ increases, the system's nonlinearity becomes more pronounced, leading to complex dynamics, including limit cycles with variable amplitude and frequency. The primary objective in this example is for the ADAM-SINDy framework to accurately identify the exponent b in the nonlinear term, particularly when b takes on rational values, as opposed to the conventional integer value of 2. Accurately identifying such rational exponents poses significant challenges for classical SINDy methods, which often struggle with precise computation in these cases (unless the precise exponent is known in SINDy-W). The ground truth for this system is defined by choosing $\mu = 0.01$ and $b = 2.15$, with the noninteger exponent contributing to the system's complex nonlinear behavior, which is generated up to $t = 200$ with the initial condition of $x_0 = 0$ and $y_0 = -1$,

for a time step of 0.01. In addition to the first-order polynomials in the candidate master library, the $\mathcal{A}$ parameter in Eq. (10) is made trainable in the ADAM-SINDy method to accommodate the chosen noninteger value, while a range of parameters (1, 2.15) and (1, 2.5) are selected for the classical SINDy-W and SINDy-W/o methodologies, respectively.

Figure 2 presents a detailed comparison of the governing equations for the ground-truth system and those identified by the SINDy-W, SINDy-W/o, and ADAM-SINDy methods. Although the SINDy-W/o method is able to align the trajectory with the ground-truth data, it identifies a significantly more complex and dense equation compared to the original system. In contrast, the ADAM-SINDy framework demonstrates its ability to accurately compute the noninteger exponent b and other coefficients of the governing equations by self-adapting the exponent. This capability is particularly noteworthy, as it enables the ADAM-SINDy approach to correctly identify relevant terms within the system's governing equations, even when the exponent is a rational number. Consequently, the ADAM-SINDy framework provides a more parsimonious representation of the system dynamics, highlighting its efficacy in dealing with noninteger exponent values that are difficult to guess *a priori* in the library of candidate terms.

A supplementary experiment was carried out to test the ability of ADAM-SINDy to recover dynamical systems containing *multiple* fractional-order terms simultaneously. Specifically, a model term of the form $x^{2.15}y^a$ was considered, where a was varied across noninteger values {1.05, 1.15, 1.25, 1.30}, perturbing the exponent of the y

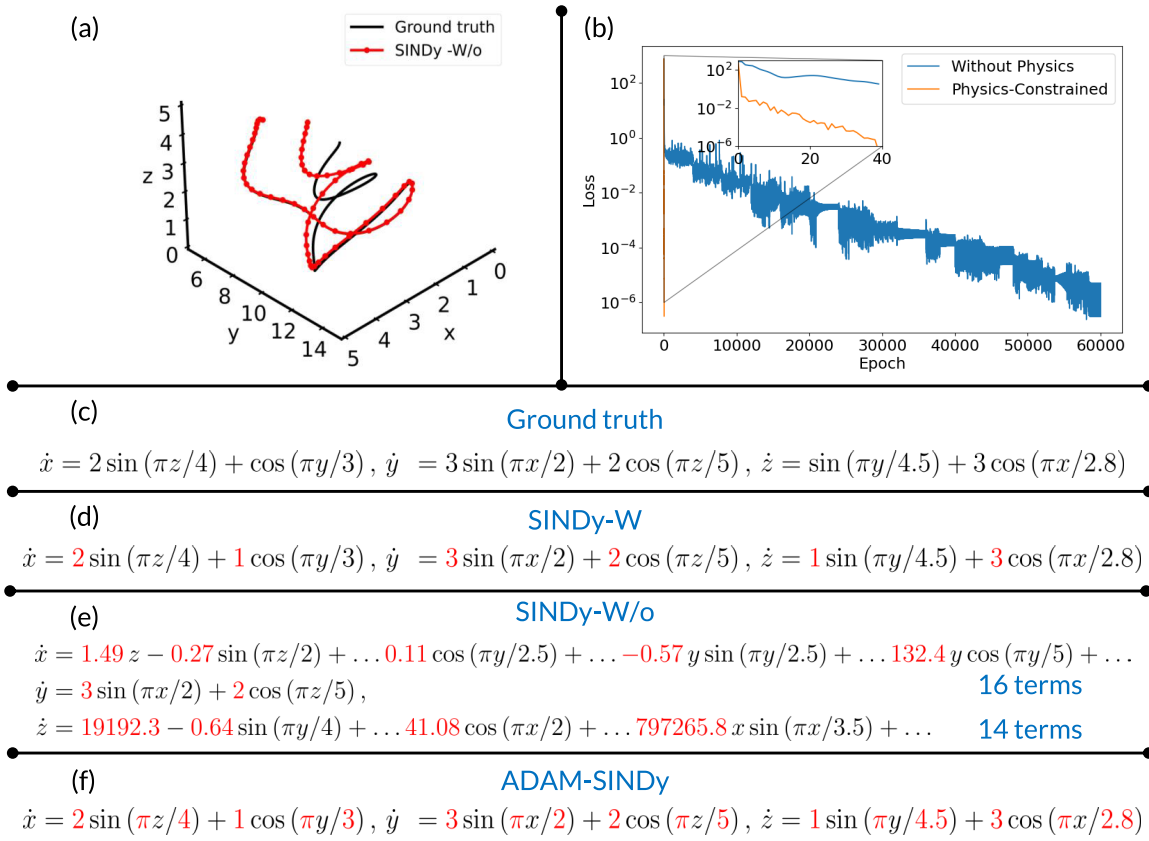


FIG. 3. ABC flow example. (a) Pathline trajectory comparison showing both the ground truth and the SINDy-W/o method. (b) Loss plot illustrating the convergence behavior of the ADAM-SINDy method with and without divergence-free condition. A comparison of the governing equations is provided between the (c) ground truth and those identified by the (d) SINDy-W, (e) SINDy-W/o, and (f) ADAM-SINDy approaches. Optimized parameter values, highlighted in red, indicate results obtained through these methodologies. For the SINDy-W/o method, a total of 16 and 14 terms were identified for $\dot{x}$ and $\dot{z}$, reflecting the complexity of the derived dynamical system. In the ADAM-SINDy method, all optimized Fourier frequency values are approximated to 16 decimal places, consistently aligning with the value of π . Ground-truth data are generated up to $t = 20$ with the initial conditions $x_0 = 0.5$, $y_0 = 0.2$, and $z_0 = 1.0$, using a time step of 0.01.

variable away from the usual integer assumption. The results, detailed in Appendix C 2, demonstrate that ADAM-SINDy accurately identifies the underlying fractional-order dynamics even when several fractional exponents are present, without requiring any modifications to the base framework.

C. ABC flow—multiple Fourier frequencies

In this section, we investigate the computation of multiple Fourier frequencies and amplitudes, focusing on the complex dynamics of an analytically defined chaotic fluid flow system. To facilitate this analysis, we examine the well-established benchmark problem known as the ABC (Arnold-Beltrami-Childress) flow, which is frequently employed to study intricate fluid dynamics and chaotic behavior in three-dimensional systems [68]. The velocity components of the ABC flow, which adhere to incompressibility, are formulated as follows:

$$\begin{aligned} \dot{x} &= A \sin(w_1 z) + C \cos(w_2 y), \\ \dot{y} &= B \sin(w_3 x) + A \cos(w_4 z), \\ \dot{z} &= C \sin(w_5 y) + B \cos(w_6 x), \end{aligned} \quad (13)$$

where x , y , and z represent the spatial coordinates, A , B , and C denote the amplitudes, and w_i correspond to the frequencies.

The ABC flow gives rise to dynamic behaviors that range from simple periodic motions to complex, chaotic patterns, contingent on the specific parameters selected. A key characteristic of the ABC flow is its ability to produce intricate three-dimensional trajectories, making it a suitable model for examining chaotic dynamics in an autonomous dynamical system. Traditionally, the ABC flow has been modeled using a single Fourier frequency for simplicity. However, this work extends the analysis to encompass six distinct frequencies $w \in (\frac{\pi}{2}, \frac{\pi}{2.8}, \frac{\pi}{3}, \frac{\pi}{4}, \frac{\pi}{4.5}, \frac{\pi}{5})$, with Fourier amplitudes of $A = 2$, $B = 3$, and $C = 1$, for which the ground truth generated up to $t = 20$ with the initial conditions $x_0 = 0.5$, $y_0 = 0.2$, and $z_0 = 1.0$, using a time step of 0.01. In this multifrequency context, both the SINDy-W and SINDy-W/o methods require an expanded candidate library, rendering the identification process more computationally intensive. In contrast, the ADAM-SINDy framework efficiently identifies all the Fourier frequencies and their corresponding amplitudes as illustrated in Fig. 3. As anticipated, the SINDy-W method successfully identifies all six Fourier frequencies and amplitudes. However, the SINDy-W/o method generates a complex and dense dynamical system equation that diverges significantly

from the ground-truth data, failing to produce accurate results despite including three correct frequencies among the six if the remaining three frequencies are incorrect. This underscores the importance of optimizing nonlinear parameters, positioning ADAM-SINDy as a robust and effective tool for identifying intricate dynamical systems such as the multifrequency ABC flow.

Additionally, a supplementary experiment was conducted to explore the potential of utilizing the computational graph inherent to stochastic gradient descent (PyTorch environment) to employ a differential divergence form in the ℓ_1 -norm regularization term. In fluid dynamics, ensuring a divergence-free velocity field, which aligns with the conservation of mass, is a critical constraint. To incorporate this physical principle into the optimization process, the sparse regression problem is redefined as follows:

$$\min_{\Xi, \Lambda} [\|\dot{\mathbf{X}} - \Theta(\mathbf{X}; \Lambda) \Xi\|_2^2 + \lambda \|\hat{\Gamma} \Xi\|_1], \quad (14)$$

where $\hat{\Gamma}$ represents the divergence equation, defined as $\partial \dot{x}_m / \partial x_m$ [where $\dot{x}_m = \Theta(\mathbf{X}; \Lambda_m) \Xi_m$], computed efficiently using PyTorch's autodifferentiation functionality. Notably, $\hat{\Gamma}$ is now a scalar value, distinct from the Γ matrix used previously. Consequently, the ℓ_1 -norm regularization term encapsulates both the divergence equation loss and sparsity ($\lambda > 0$), thereby guiding the total loss function. This approach selectively retains candidate functions that satisfy the divergence-free condition, thereby ensuring compliance with the continuity equation within the ADAM-SINDy optimization framework. By fixing λ to 1, this approach enables the identification of all six Fourier frequencies and their corresponding amplitudes within just 40 epochs, as shown in Fig. 3, marking a substantial reduction in computational effort compared to the standard ADAM-SINDy method, which typically requires around 60 000 epochs when Γ is a weighting matrix. This preliminary investigation suggests the potential for a physics-informed SINDy approach, which diverges from the primary focus of the current study and will be pursued in future work.

D. Chemical Reaction Kinetics—Exponential Exponent

Next, the computation of an exponential exponent within a dynamical system is examined through a chemical reaction kinetics problem that employs an exponential approximation of the Arrhenius rate law, which characterizes the nonlinear relationship between reaction rates and temperature, particularly in the context of coupled first-order reactions [69]. When the temperature increase due to an exothermic reaction is negligible compared to the ambient temperature, the reaction rate can be approximated as exponentially dependent on this incremental temperature change. By appropriately scaling the governing equations for mass and energy, the system is expressed as follows [69]:

$$\begin{aligned} \dot{\alpha} &= -k\alpha e^{g\theta} + \mu, \\ \dot{\theta} &= \alpha e^{h\theta} - \theta, \end{aligned} \quad (15)$$

where $\alpha(t)$ represents the concentration of an intermediate reactant, $\theta(t)$ denotes the temperature rise, k is the reaction rate constant, and μ corresponds to the reactant concentration.

The model assumes that the states θ and α evolve on a faster time scale compared to the slowly changing reactant consumption μ , invoking the pseudo-stationary-state hypothesis [70], thus μ interacting as a constant source term. This formulation offers insights into the temporal evolution of reactant concentrations and temperature, enabling the analysis of reaction kinetics and the impact of external perturbations on the reaction system.

Conventionally, the growth/decay rates g and h are assumed to be unity for the system under consideration [69,70]. However, this experiment adopts noninteger values for the parameters, specifically $g = 1.015$ and $h = 1.025$, while setting $k = 0.07$ and $\mu = 0.1$ to generate the ground-truth data up to $t = 200$ with initial conditions $\alpha_0 = 0$ and $\theta_0 = 5$ using a time step of 0.02. Figure 4 provides a comprehensive comparison of the classical SINDy-W, SINDy-W/o, and ADAM-SINDy methods in identifying the chemical reaction system. In this context, the SINDy-W/o method exhibits deviation from the ground-truth trajectory at α close to zero, failing to accurately replicate the dynamics. In contrast, the ADAM-SINDy method successfully identifies the noninteger exponent values and other relevant coefficients perfectly aligning with the ground-truth data, thereby demonstrating its efficacy in addressing systems characterized by exponential functions.

E. Pharmacokinetics Model—Time-dependent Reaction

This example investigates a dynamical system that incorporates a time-dependent function with a noninteger exponent in its governing equations, focusing on a pharmacokinetics model. Such models are commonly utilized to assess the rate of drug concentration across various compartments in the human body. Specifically, we consider a single-dose compartmental model [71] governed by the following system of equations:

$$\begin{aligned} k_g &= k_0 t^{-\eta}, \\ \dot{B} &= k_g G - k_b B, \\ \dot{G} &= -k_g G, \\ \dot{U} &= k_b B, \end{aligned} \quad (16)$$

where k_g denotes the time-dependent reaction rate [72], k_0 is the initial rate coefficient, and η represents the fractal kinetics exponent, constrained within the range $0 < \eta < 1$. The ground-truth data are generated using a drug absorption rate of $k_b = 0.15$, $k_0 = 0.72$, and a fractal kinetics exponent of $\eta = -0.5$, evolved up to $t = 10$ with the initial conditions $B_0 = 0$, $G_0 = 1$, and $U_0 = 0$ for a time step of 0.001.

In this example, the master library is expanded to include $\mathbf{X} \otimes t^\eta$, thereby accommodating the time-dependent function, with η treated as a trainable parameter within the ADAM-SINDy framework, while a range of parameters $(-0.4, -0.5, -0.55)$ and $(-0.4, -0.55)$ are defined for the SINDy-W and SINDy-W/o methodologies, respectively. Figure 5 presents a detailed comparison of the governing equations for the ground-truth system with those identified by the SINDy-W, SINDy-W/o, and ADAM-SINDy methodologies. Notably, the SINDy-W/o method closely aligns with the ground-truth trajectory, exhibiting no deviations. However,

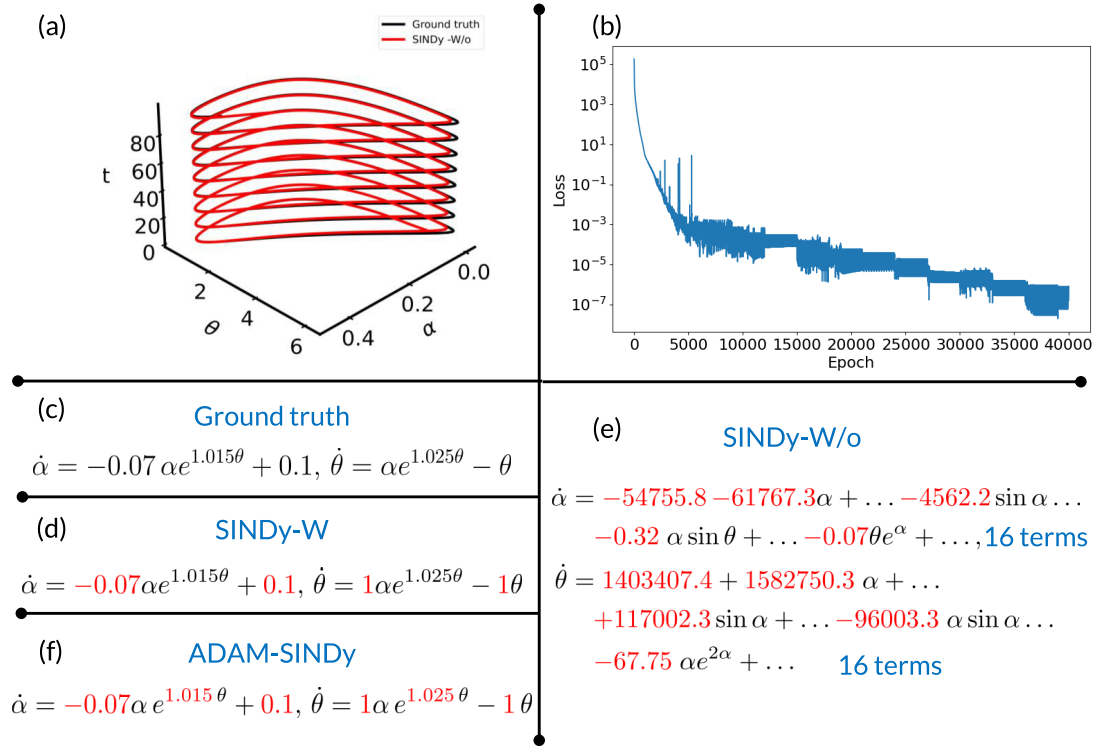


FIG. 4. Chemical reaction kinetics example. (a) Temporal trajectory comparison showing both the ground truth and the SINDy-W/o method. (b) Loss plot demonstrating the convergence behavior of the ADAM-SINDy method. A comparison of the governing equations is provided between the (c) ground truth and those identified by the (d) SINDy-W, (e) SINDy-W/o, and (f) ADAM-SINDy approaches. The optimized parameters, highlighted in red, represent the results obtained from these methodologies. For the SINDy-W/o method, a total of 16 terms were identified for both $\dot{\alpha}$ and $\dot{\theta}$, indicating the complexity of the resulting dynamical system. Ground-truth data are generated up to $t = 200$ with the initial conditions $\alpha_0 = 0$ and $\theta_0 = 5$, using a time step of 0.02.

the resulting equation from this method is significantly more complex and dense compared to the original system. In contrast, the ADAM-SINDy framework accurately computes the exponent η associated with the temporal function, along with other coefficients, mirroring the governing system precisely and thereby demonstrating superior performance compared to the classical SINDy methods.

F. Wildfire Transport—Partial Differential Equation

The classical SINDy method was initially developed to discover systems of ODEs but has been extended to encompass PDEs through a process known as PDE-FIND [73]. This extension is particularly pertinent for systems exhibiting dependencies in both spatial and temporal domains. In this section, we investigate wildfire dynamics modeled by PDEs, focusing on the identification of governing nonlinear parameters that elucidate the spatiotemporal evolution of temperature within the fire. These governing equations encapsulate critical processes, including heat transfer, combustion, and the influence of wind, thereby providing a comprehensive framework for understanding wildfire propagation across various fuel beds. The following nondimensional wildfire combustion model [74] delineates this behavior, incorporating terms for advection, linear diffusion, reaction, and natural cooling:

$$\frac{\partial T}{\partial t} + \vec{\nabla} \cdot \nabla T = \kappa \nabla^2 T + \beta e^{T/(1+\epsilon T)} - \alpha T, \quad (17)$$

where $T(\mathbf{x}, t)$ denotes the nondimensionalized absolute temperature, $\vec{\nabla}$ represents the normalized velocity vector of the fire front, κ is the normalized thermal diffusivity, β denotes the fuel mass fraction, ϵ is the inverse activation energy, and α represents the natural heat transfer coefficient. For this experiment, we consider the candidate matrix

$$\Theta = \begin{bmatrix} 1 & \mathbf{X} & \nabla \mathbf{X} & \nabla^2 \mathbf{X} & \nabla^3 \mathbf{X} & \nabla^4 \mathbf{X} & \exp\left(\frac{\mathbf{X}}{1+\mathcal{H}\mathbf{X}}\right) \end{bmatrix}, \quad (18)$$

which incorporates a diverse range of candidate functions. This matrix includes polynomials up to the first order, as well as an exponential term with $\mathcal{H}$ serving as an unknown optimizable parameter, consistent with Arrhenius' law—typically employed in modeling fuel combustion processes. Additionally, it encompasses higher-order partial spatial derivatives, extending to the fourth derivative.

The ground-truth data are generated by employing an upwind compact scheme for spatial derivatives and implicit-explicit Runge-Kutta time integration within the finite difference method, where the initial Heaviside firefront is situated at coordinates (8.0, 8.0) with a square side length of 4, computed up to $t = 0.08$ with a time step of 10^{-7} . The dimensionless parameters include a north-east facing freestream velocity magnitude of 10, a linear diffusion coefficient $\kappa = 1.1$, an inverse activation energy $\epsilon = 0.3$, a fuel $\beta = 1$ and a natural heat transfer coefficient $\alpha = 0.20$. In the context of the SINDy-W and SINDy-W/o methods, we consider the sets

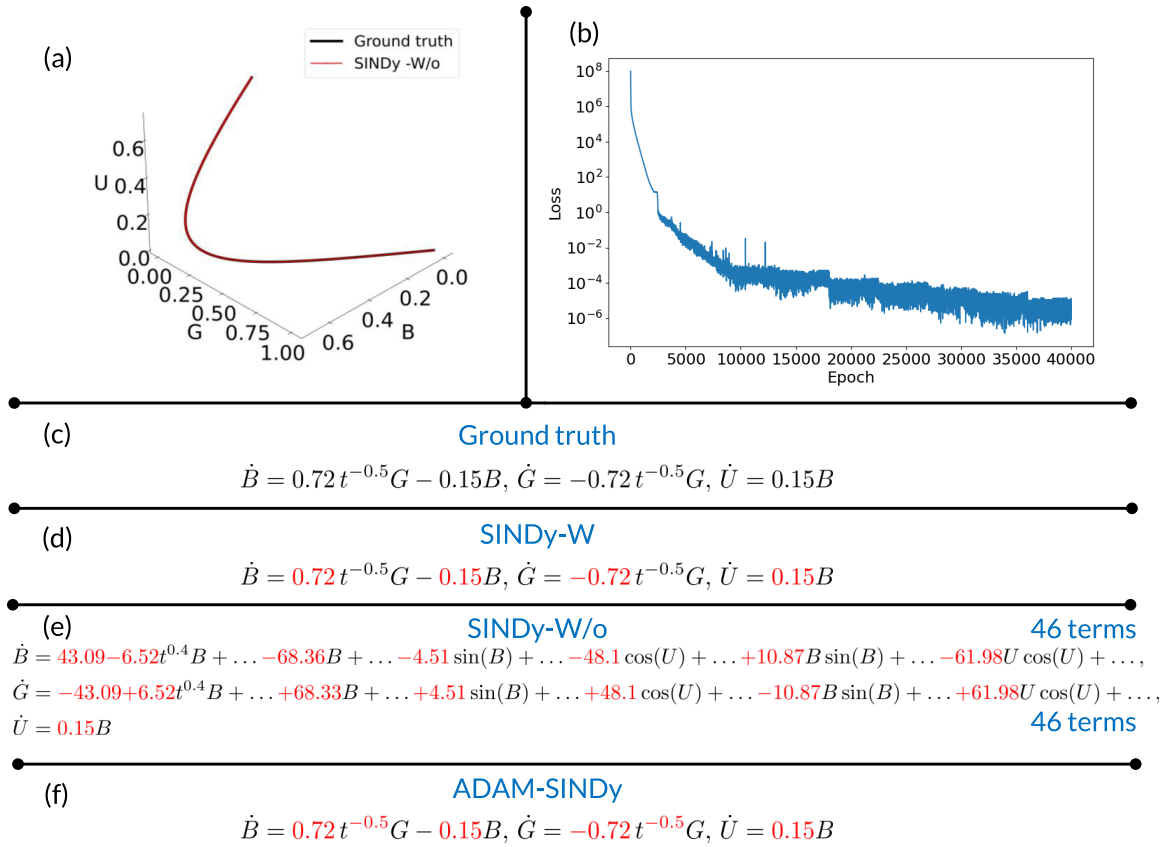


FIG. 5. Pharmacokinetics example. (a) Temporal trajectory comparison highlighting both the ground truth and the SINDy-W/o method. (b) Loss plot illustrating the convergence behavior of the ADAM-SINDy method. A comparison of the governing equations is provided between the (c) ground truth and those identified by the (d) SINDy-W, (e) SINDy-W/o, and (f) ADAM-SINDy approaches. The optimized parameters, highlighted in red, reflect the outcomes of these methodologies. For the SINDy-W/o method, a total of 46 terms were identified for both $\dot{B}$ and $\dot{G}$, indicating the complexity of the resulting dynamical system. Ground-truth data are generated up to $t = 10$ with the initial conditions $B_0 = 0$, $G_0 = 1$, and $U_0 = 0$, using a time step of 0.001.

$\mathcal{H} \in \{0.2, 0.3, 0.4\}$ and $\{0.2, 0.4\}$ for the identification of the wildfire equation, with the sparsity parameter set to $\lambda = 0.01$. Figure 6 illustrates a detailed comparison of the governing equations for the ground-truth system with those identified by the SINDy-W, SINDy-W/o, and ADAM-SINDy methodologies. As anticipated, the SINDy-W/o method identifies an overly complex and dense wildfire model in comparison to the ground-truth system. Interestingly, the ADAM-SINDy framework accurately identifies the exact wildfire combustion model, perfectly predicting the inverse activation energy associated with the reaction term.

G. Hyperparameter Robustness—Sparsity Knobs

In this section, we investigate the optimization efficacy of the ADAM-SINDy framework in system identification, focusing on its sensitivity to the sparsity-promoting hyperparameters, Γ and λ . The performance of ADAM-SINDy is compared with the ideal case, the SINDy-W method. First, the SINDy-W method is employed to systematically vary the λ parameter across one order of magnitude, ranging from smaller to larger values, in order to determine the bounds beyond which the model diverges significantly from the ground truth. Specifically, we identify the minimum λ ,

below which the SINDy-W method generates an overly dense system of equations, and the maximum λ , beyond which the model fails to capture meaningful dynamics. The computed bound values for the SINDy-W method are presented in Table II.

In the context of the ADAM-SINDy method, two scenarios are considered: (1) Γ is fixed to the value of 1, while λ is trainable, and (2) Γ is trainable, with its initialization varied by varying the standard deviation (Γ_σ) of a Gaussian distribution while keeping its mean (Γ_μ) at zero, and λ is fixed at a value of 1. Experiments are conducted to compute the bounds for both scenarios across all the numerical examples considered in the present work, and the corresponding bounds are tabulated in Table II. Generally, the ADAM-SINDy framework provides much wider bounds for the sparsity knobs, often at least one order magnitude higher, especially in cases with higher initialization values, indicating that ADAM-SINDy is more robust than the classical SINDy method. Comparing the trainable λ and trainable Γ cases in ADAM-SINDy, it is observed that in each example, one of the two approaches provides a wider bound for the sparsity knob, indicating no clear advantage. However, in the former scenario (trainable λ), ADAM-SINDy identifies the correct candidate terms, but the coefficients and nonlinear parameters are not always machine

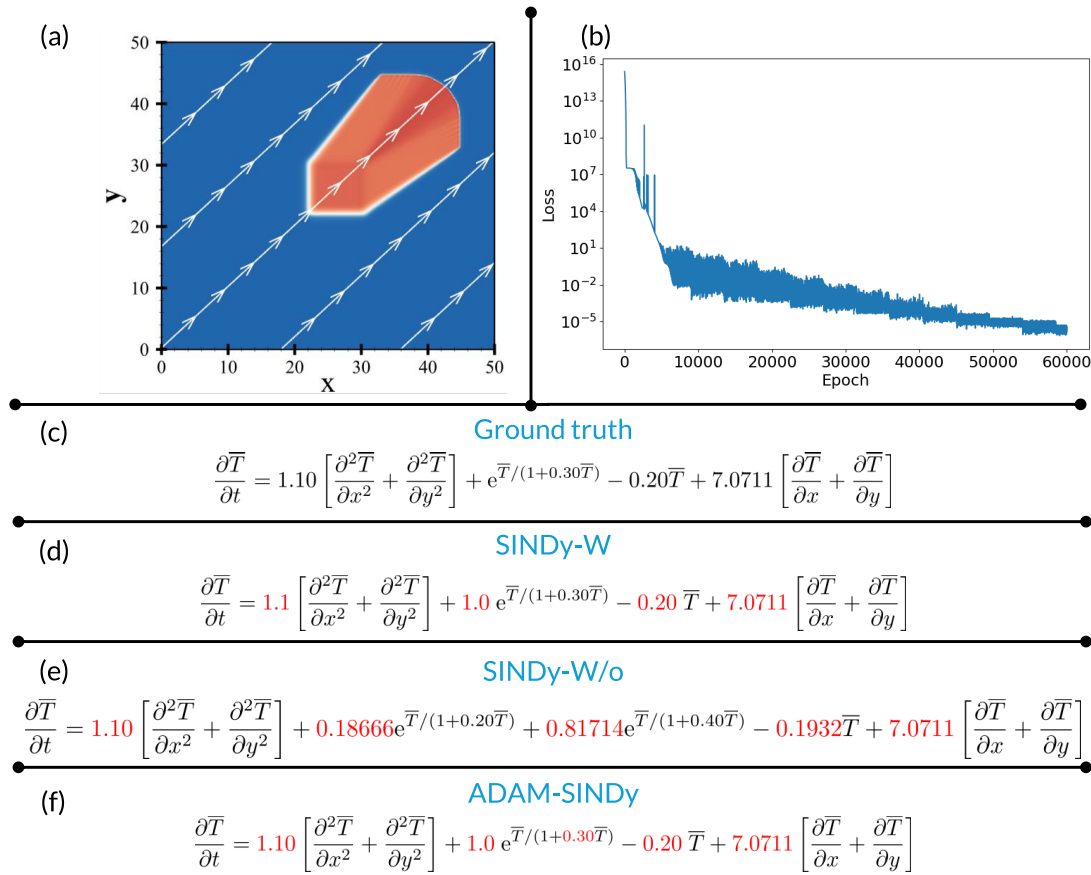


FIG. 6. Wildfire dynamics example. (a) Ground-truth temperature data superimposed with north-east facing wind streamlines at $t = 0.05$. (b) Loss plot illustrating the convergence behavior of the ADAM-SINDy method. A comparison of the governing equations is provided between the (c) ground truth and those identified by the (d) SINDy-W, (e) SINDy-W/o, and (f) ADAM-SINDy approaches. Optimized parameters, highlighted in red, indicate the results obtained through these methodologies. The initial Heaviside firefront is situated at coordinates (8.0, 8.0) with a square side length of 4, computed up to $t = 0.08$ with a time step of 10^{-7} .

precision. In the latter case (trainable Γ), the ADAM-SINDy method identifies the exact correct terms and nonlinear parameter values with machine precision.

Our experiments demonstrate that the latter optimization strategy, which imposes candidatewise sparsity, is recommended for nonlinear system identification. A notable example is the chemical kinetics problem, illustrated in Fig. 7, which represents the former scenario of trainable λ . It is

observed that the SINDy-W method fails to produce the ground-truth equation, whereas ADAM-SINDy remains robust, producing the correct terms with slight deviations in the parameter values when initialized with a higher λ value of 0.1. Furthermore, results for the latter scenario with trainable Γ are not shown here, as the identified terms and parameter values are observed to match the ground truth exactly, as demonstrated in Fig. 4.

TABLE II. Bounds on the sparsity-promoting hyperparameters for the SINDy-W and ADAM-SINDy methodologies, illustrating the conditions under which each method accurately identifies the dynamical system. In the ADAM-SINDy method, λ is trainable while Γ is fixed at 1 in the first column. It is noted that no λ values were found to work for the wildfire PDE example. In the second column, Γ is trainable while λ is fixed at 1 and Γ is initialized using a Gaussian distribution where its mean Γ_μ is fixed at zero and its standard deviation Γ_σ is varied as shown.

Experiment	SINDy-W	ADAM-SINDy	
		λ	Γ
Harmonic oscillator	$10^{-3} < \lambda < 10^{-1}$	$10^{-3} < \lambda < 10$	$10^{-8} < \Gamma_\sigma < 10$
Van der Pol oscillator	$10^{-3} < \lambda < 10^{-2}$	$10^{-5} < \lambda < 10$	$10^{-5} < \Gamma_\sigma < 10$
ABC flow	10^{-1}	10^{-1}	$10^{-2} < \Gamma_\sigma < 10$
Chemical reaction	$10^{-3} < \lambda < 10^{-1}$	$10^{-12} < \lambda < 10$	$10^{-9} < \Gamma_\sigma < 10$
Pharmacokinetics	$10^{-2} < \lambda < 10^{-1}$	$10^{-3} < \lambda < 10$	$1 < \Gamma_\sigma < 10$
Wildfire PDE	$10^{-6} < \lambda < 1$	—	$10^{-6} < \Gamma_\sigma < 1$

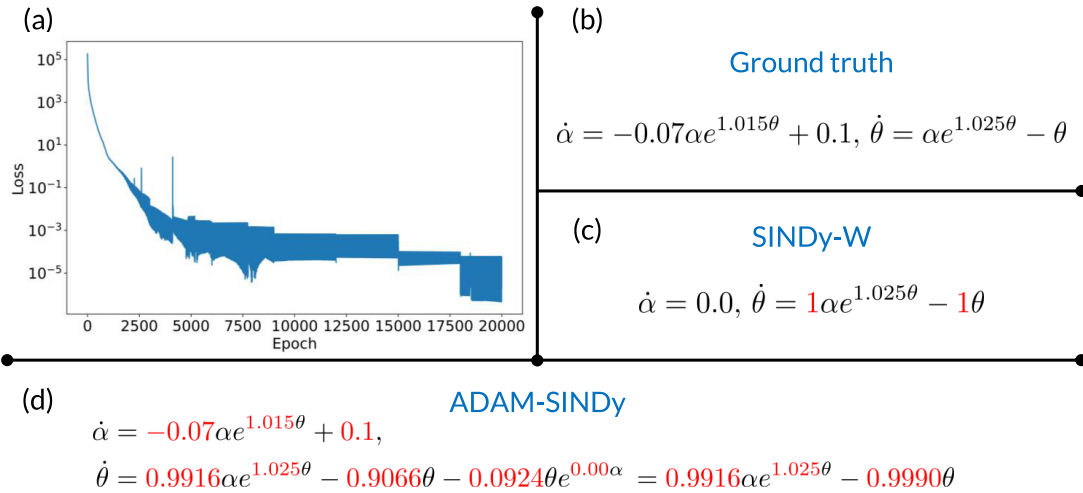


FIG. 7. The influence of the sparsity knob (λ) on chemical reaction kinetics results. (a) Loss plot showing the convergence of the ADAM-SINDy method for $\lambda = 0.1$. A comparison of the governing equations is provided between the (b) ground truth and those identified by the (c) SINDy-W and (d) ADAM-SINDy methods, for $\lambda = 0.1$. The red-highlighted numerical values indicate the optimized parameters obtained.

IV. DISCUSSION

Identification of nonlinear dynamical systems is a prominent research domain that plays a pivotal role in elucidating the underlying physics of various problems. The literature highlights extensive investigations that employ the SINDy and symbolic regression frameworks, both of which can identify governing equations from observational data. However, it is essential to recognize that these well-established methods possess distinct advantages and disadvantages, particularly concerning computational optimization. In this context, the newly developed ADAM-SINDy differentiable optimization framework emerges at the intersection of these established approaches, leveraging their computational optimization benefits. By leveraging both the differentiable optimization and the ADAM algorithm, ADAM-SINDy facilitates the simultaneous optimization of parameters and coefficients associated with candidate nonlinear functions. This innovative approach enables more efficient and accurate parameter estimation without the necessity for prior knowledge of nonlinear characteristics, such as frequencies in trigonometric functions or bandwidths in exponential functions. A key factor in ADAM-SINDy's success lies in its ability to exploit the computational graph inherent in the gradient descent algorithm implementation in the PyTorch environment. This structure facilitates efficient backpropagation of gradients, enabling dynamic interaction between candidate terms and nonlinear parameters during the optimization process. By allowing the model to adaptively refine both the nonlinear parameters and coefficients, ADAM-SINDy enhances the robustness and precision of system identification, particularly in complex dynamical systems where the classical SINDy method struggles with sparse, nonconvex landscapes. Notably, the strategy behind the simultaneous optimization of all unknown parameters and coefficients closely parallels the use of adaptive activation functions [75] in neural network research, where the parameters of nonlinear adaptive activation functions, along with the weights and biases of linear layers, are optimized concurrently.

Concurrently, ADAM-SINDy seeks to identify a parsimonious mathematical model while optimizing trainable sparsity-promoting hyperparameters, such as Γ , through two distinct objective formulations: joint minimization and min-max. While the min-max adaptive weighting strategies also appear in frameworks like PINNs [66] and transformers [76], we show that the joint-minimization strategy is more effective in practice, primarily because the convex nature of the ℓ_1 sparsity term over both Γ and Ξ precludes the existence of a saddle point required for the min-max formulation to converge reliably. As demonstrated in the sensitivity analysis (Sec. III G), both trainable hyperparameters Γ and λ can be used for ADAM-SINDy optimization; however, prioritizing Γ is recommended, as it consistently recovers the correct terms with machine precision, whereas λ may yield correct structures with slight numerical deviations. ADAM-SINDy supports both batchwise and full-batch optimization. For simpler dynamical systems, small randomly sampled batches often capture sufficient temporal information to accurately estimate governing parameters. In contrast, complex systems may require larger batches to encompass the necessary information, making full-batch optimization preferable. Accordingly, full-batch training is adopted in this work, leading to reduced uncertainty in the converged parameters within the ADAM-SINDy framework.

Particular emphasis is placed on the essential steps required for the joint-minimization approach to succeed, given the counterintuitive nature of this optimization problem. First, Γ must be incorporated within the ℓ_1 loss as a transformed value. In the present work, that transformation is the absolute value function, although alternatives such as an exponential functions are possible. Taking the absolute value of the Γ matrix prior to computing the loss forces $\Gamma \rightarrow 0$ rather than $-\infty$. This absolute value function also introduces stochasticity into the trajectories of Γ , as zero crossings occur, subsequently influencing Ξ and Λ , despite the use of full-batch optimization in the examples. Second, sequential thresholding is required to select a sparse system within ADAM-SINDy. Without this step, all Γ values would tend toward zero, eventually

reducing the problem to a least-squares fit after sufficient epochs. Thresholding allows for saturation of selected Γ values, thereby establishing a fixed minimum level of sparsity that can only increase with each epoch. The specific threshold value itself is an additional hyperparameter that may vary depending on the specific problem; however, the same threshold value was used across all examples in the present work, indicating robust behavior. The theoretical foundations of the joint-minimization approach are further elaborated in Appendix B.

Ultimately, the integrated simultaneous adaptive optimization of the nonlinear parameters, coefficient matrix, and hyperparameters based on training data positions ADAM-SINDy as a powerful and effective tool for modeling a diverse range of dynamical behaviors across various applications. The proposed ADAM-SINDy methodology is thoroughly investigated across a wide spectrum of dynamical systems, including benchmark coupled ODEs such as oscillators, fluid flows, reaction kinetics, and wildfire dynamics PDE. These comprehensive investigations demonstrated its efficacy in accurately computing nonlinear features such as multiple Fourier frequencies, noninteger exponents in polynomials, and exponential exponents. Furthermore, the investigation also explored the inherent limitations of the classical SINDy method, illustrating its consistent failure to identify the exact model when the precise parameters are absent from the candidate matrix. This comprehensive investigation highlights the considerable potential of the ADAM-SINDy optimization framework, thereby extending its applicability to the modeling of intricate and diverse dynamical systems across various contexts with limited prior knowledge.

Similarly, Boddupalli *et al.* [69] utilized symbolic regression via neural networks to identify dynamical systems, in particular, the numerical example of chemical reaction kinetics, the same system considered in this work (Sec. III D). Without explicit stacking within the neural network, the resulting model becomes overly complex or “bloated.” However, when exponential terms are explicitly combined with polynomials, the identified model closely approximates the ground-truth equation, although slight deviations in the coefficient values remain. In contrast, ADAM-SINDy accurately identifies the true dynamical system, capturing the dynamics without requiring additional treatments to ensure regression convergence. Besides, a methodological comparison between ADAM-SINDy and symbolic regression using the PySR framework, with a focus on model compactness and computational efficiency across all the considered numerical examples, is provided in Appendix C 1, demonstrating that PySR fails to discover a true representation for the Van der Pol oscillator involving fractional-order polynomials and instead yields a bloated expression, whereas ADAM-SINDy performs robustly. Generally, symbolic regression is ideal when little or no prior knowledge of the governing physics is available, enabling flexible discovery of candidate model structures, while ADAM-SINDy is preferred when partial knowledge exists, allowing the library to be tailored to known structures in the data, such as polynomials or periodic functions. In real-world scenarios, observational data are inevitably contaminated by noise from various sources, motivating an evaluation of ADAM-SINDy’s efficacy under such conditions. To assess

its robustness, we introduced Gaussian white noise of zero mean and nonzero standard deviation into a representative example, the Van der Pol oscillator. For a mild noise level of 10%, ADAM-SINDy recovers the underlying dynamical system, whereas for a higher noise level of 20% it produces a bloated, complex model. To address this, we applied total variational denoising (TVD) to preprocess the noisy data, after which ADAM-SINDy recovers a reliable dynamical system. More detailed discussions are provided in Appendix C 3.

A notable limitation of full-batch ADAM-SINDy is its substantial computational cost, arising from the use of the ADAM optimizer for the coefficient matrix. Although this approach offers advantages over classical SINDy, the improvement incurs considerable overhead. Computational efficiency may be enhanced by employing an iterative least-squares method for the coefficient matrix while retaining ADAM for nonlinear parameters and sparsity control, resulting in a hybrid *differentiable* least-squares-ADAM framework. Further investigation is required to develop robust methodologies that effectively integrate these approaches while preserving their respective strengths. Embedding physical constraints directly into the learning process has also proven highly beneficial. In the ABC flow example, enforcing a divergence-free condition via the computational graph led to a dramatic reduction in training time. This demonstrates that leveraging underlying physical laws can both improve model fidelity and significantly reduce computational cost, offering a promising direction for future work. Given the primary focus on “concurrent nonlinear parameter computations” to address a key limitation of classical SINDy, ADAM-SINDy does not fully overcome the challenge of constructing a master library containing all possible candidate terms for the problems considered. While a master library is included in the present setup, additional candidate functions were necessary for some examples. This limitation is intrinsic to this class of problems. Classical SINDy acknowledged this constraint (see the Supplemental Material in Ref. [41]) through an illustrative example involving sine series and polynomial summations. This example demonstrated that nonlinear functions tend to approximate higher-order polynomials as sparsity is enforced. Similarly, symbolic regression starts with simple polynomial functions and progressively introduces higher-order and nonlinear terms during the evolution stage. Developing strategies to accurately identify candidate functions, particularly in the presence of complex nonlinearities and high-order polynomials, remains an open problem.

Finally, the ADAM-SINDy methodology establishes a differentiable, joint optimization framework that could be broadly applicable to both sparse and symbolic regression problems, particularly in scenarios where optimizing internal parameters of nonlinear functions is a computational bottleneck. By leveraging the computational graph, ADAM-SINDy enables simultaneous optimization of all parameters, overcoming limitations that extends beyond SINDy to symbolic regression and genetic-fragment-based equation discovery methods. Furthermore, this framework can facilitate physics discovery in latent spaces, where reduced-order representations of high-dimensional systems are extracted via autoencoders. Integrated with the Latent Space Dynamics Identification framework [77], ADAM-SINDy can allow for

the extraction of governing equations with even more compact nonlinear latent representations, supporting interpretable and predictive modeling of complex physical phenomena.

V. CONCLUSION

In summary, the proposed ADAM-SINDy methodology successfully addresses the limitations of the classical SINDy framework by enabling simultaneous identification of both system nonlinear parameters and coefficients through ADAM optimization. By eliminating the need for predefined parameter distributions and extensive manual hyperparameter tuning, ADAM-SINDy enhances the efficiency, adaptability, and robustness of system identification, particularly in the presence of nonlinear parameters. The demonstrated improvements across diverse dynamical systems highlight the

potential of this approach to significantly broaden the applicability of the SINDy framework in complex modeling challenges.

ACKNOWLEDGMENTS

This research was supported by the National Science Foundation under Awards No. 2247173 and No. 2330212. The authors express their gratitude to Dr. Jacob Hochhalter, Dr. Sarang Joshi, and Dr. Mike Kirby for their valuable discussions about this work.

The authors declare no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this article are openly available [78].

APPENDIX A: ADAM-SINDy OPTIMIZATION ALGORITHM

ALGORITHM 1. ADAM-SINDy algorithm.

Input:

Observational data $\mathbf{X}$ and $\dot{\mathbf{X}}$
Tolerance ϵ and maximum iterations $N_{\max}$
Learning rate η_{Ξ} , η_{Λ} , η_{λ} , and η_{Γ}

Initialization:

Initialize $\Xi \leftarrow \Xi_0$, $\Lambda \leftarrow \Lambda_0$, $\lambda \leftarrow \lambda_0$, $\Gamma \leftarrow \Gamma_0$
Set iteration counter $n \leftarrow 0$

Optimization loop:

while $n < N_{\max}$ **do**

For each batch $b = 1, 2, \dots$

 Sample a batch of data $\mathbf{X}_b$ and $\dot{\mathbf{X}}_b$

 Compute $\Theta(\mathbf{X}_b; \Lambda)$

Compute Loss Function:

$$\mathcal{L}_b \leftarrow \|\dot{\mathbf{X}}_b - \Theta(\mathbf{X}_b; \Lambda)\Xi\|_2^2 + \lambda \|\Gamma|\Xi|\|_1$$

Update Parameters using Backpropagation:

$$\Xi_{n+1} \leftarrow \Xi_n - \eta_{\Xi} \nabla_{\Xi} \mathcal{L}_b(\Theta_b, \Xi, \Lambda, \lambda, \Gamma)$$

$$\Lambda_{n+1} \leftarrow \Lambda_n - \eta_{\Lambda} \nabla_{\Lambda} \mathcal{L}_b(\Theta_b, \Xi, \Lambda, \lambda, \Gamma)$$

Update regularization parameters:

where $s \leftarrow +1$ for ascent, $s \leftarrow -1$ for descent

if uniform sparsity **then**

$$\lambda_{n+1} \leftarrow \lambda_n + s \cdot \eta_{\lambda} \nabla_{\lambda} \mathcal{L}_b(\Theta_b, \Xi, \Lambda, \lambda, \Gamma)$$

else

$$\Gamma_{n+1} \leftarrow \Gamma_n + s \cdot \eta_{\Gamma} \nabla_{\Gamma} \mathcal{L}_b(\Theta_b, \Xi, \Lambda, \lambda, \Gamma)$$

Thresholding: For each element i , set $\Xi_{n+1}^i, \Lambda_{n+1}^i \leftarrow 0$ if $|\Xi_{n+1}^i|, |\Lambda_{n+1}^i| \leq \epsilon$

 Increment iteration counter: $n \leftarrow n + 1$

end while

Output:

Optimized parameters Ξ^* and Λ^*

The ADAM-SINDy framework, detailed in Algorithm 1, introduces a fully adaptive global optimization process for all parameters typically preselected in the classical SINDy framework. It begins with the initialization of key parameters— Ξ , Λ , λ , and Γ —which are iteratively optimized based on the provided input data. To enhance computational efficiency and robustness, the input data are divided into batches of size B . When the batch size is set equal to the size of the input data, the algorithm operates under a full-batch optimization procedure. During each epoch, for each batch (or the full dataset in the case of full-batch optimization), the loss function $\mathcal{L}_b$ is computed to measure the discrepancy between the observed system dynamics, $\dot{\mathbf{X}}_b$, and the model's predicted dynamics, represented by $\Theta(\mathbf{X}_b; \Lambda) \Xi$. This batchwise loss computation incorporates a sparsity-inducing term, regulated by either λ or Γ knob, ensuring that only the most relevant terms are selected.

The parameters Ξ , Λ , λ , and Γ are then updated using stochastic gradient descent on a per-batch basis, progressively minimizing the loss function over the entire dataset. For full-batch optimization, updates are applied after evaluating the loss on the complete dataset. Importantly, the updates for the regularization parameters λ and Γ differ from those of Ξ and Λ . Depending on the optimization formulation, the regularization parameters are updated via either gradient ascent or gradient descent, governed by a sign variable s . Specifically, in the min-max optimization formulation of Eq. (8), λ and Γ are updated through gradient ascent ($s = +1$), whereas in the joint-minimization formulation of Eq. (9), the updates proceed via gradient descent ($s = -1$). This flexible update methodology enables adaptive control of sparsity throughout training. After each batch update, a sequential thresholding step sets entries of both Ξ and Λ to zero if their magnitudes fall below a specified tolerance ϵ . This iterative procedure continues for each batch until the maximum number of iterations $N_{\max}$ is reached, ensuring efficient convergence to a sparse and accurate model representation.

Stochastic versus Full-Batch Optimization: The ADAM-SINDy framework provides the flexibility to implement both a batching strategy through stochastic gradient descent and a full-batch optimization approach. In the stochastic case, optimization is performed on smaller, randomly sampled subsets of the overall dataset. This method optimizes computational resource utilization by partitioning the data into manageable chunks, thereby reducing memory requirements and accelerating the optimization process. The batch size must be sufficiently large to capture the dynamics of interest while remaining small enough to facilitate efficient computation. In contrast, full-batch optimization processes the entire dataset in a single iteration. This approach leverages all available information, potentially yielding more accurate and stable updates to the optimization parameters. Notably, ADAM optimization employs momentum, which can help avoid local minima even in the absence of stochasticity introduced by mini-batches.

While both methodologies share the same optimization strategy, a primary difference lies in the number of temporal data points considered in each optimization step. In the stochastic batching case, the batch size is typically smaller than the total number of available temporal data points. These data points are randomly sampled and shuffled, with the final

batch potentially containing fewer points than the specified batch size. During implementation, temporal data are organized into rows based on the selected batch size, which dictates the number of rows in the ground-truth data used in each optimization step. Then the candidate function matrix is constructed, reflecting the selected batch size through the number of rows it contains. Importantly, the dimensionality of the unknown nonlinear parameters and the coefficient matrix remains invariant, as the underlying dynamics of the system are independent of data partitioning based on time steps. Consequently, employing a batching strategy in the sparse regression problem enhances computational efficiency, reduces memory requirements, and facilitates effective manipulation of the matrix size.

APPENDIX B: JOINT-MINIMIZATION VERSUS MIN-MAX OPTIMIZATION

In this Appendix, we compare two optimization formulations considered within ADAM-SINDy: the *min-max* formulation and the *joint-minimization* formulation. Both approaches aim to balance prediction accuracy and model sparsity but differ fundamentally in how sparsity is enforced. First, the min-max objective function is given by

$$\min_{\Xi, \Lambda} \max_{\Gamma} [\|\dot{\mathbf{X}} - \Theta(\mathbf{X}; \Lambda) \Xi\|_2^2 + \lambda \|\Gamma\| \|\Xi\|_1],$$

where the sparsity term is adversarially weighted by the variable Γ . In contrast, the joint-minimization formulation eliminates the adversarial structure by optimizing all variables cooperatively:

$$\min_{\Xi, \Lambda, \Gamma} [\|\dot{\mathbf{X}} - \Theta(\mathbf{X}; \Lambda) \Xi\|_2^2 + \lambda \|\Gamma\| \|\Xi\|_1],$$

treating Γ as a passive regularization weight rather than an adversarial one. In both cases, the objective function consists of a data fidelity term (the ℓ_2 norm) and a sparsity-promoting term (the ℓ_1 norm). However, the role of Γ as either an adversary or a cooperative weight introduces fundamentally different optimization dynamics.

1. Theoretical analysis of the optimization frameworks

To analyze the optimization dynamics, we consider the gradients of the objective function with respect to Ξ and Γ :

$$\begin{aligned} \nabla_{\Xi} L &= -2\Theta^T(\dot{\mathbf{X}} - \Theta\Xi) + \lambda |\Gamma| \cdot \text{sgn}(\Xi), \\ \nabla_{\Gamma} L &= \lambda |\Xi| \cdot \text{sgn}(\Gamma). \end{aligned}$$

where $\text{sgn}(z) = \{+1 \text{ if } z > 0, 0 \text{ if } z = 0, -1 \text{ if } z < 0\}$. These expressions highlight the fundamental instability in the optimization dynamics when Γ is treated adversarially (e.g., min-max). The gradient $\nabla_{\Gamma} L$ drives Γ to grow unbounded in directions that increase the sparsity penalty $\|\Gamma\| \|\Xi\|_1$, provided that Ξ contains nonzero values. As Γ increases, the regularization term increasingly dominates $\nabla_{\Xi} L$, subsequently driving $\Xi \rightarrow 0$. This feedback loop leads to pathological dynamics in which $|\Gamma| \rightarrow \infty$ and $\Xi \rightarrow 0$, effectively suppressing model expressivity and hindering recovery of the correct terms. Importantly, this instability arises from the fact that the sparsity term is convex in both Ξ and Γ . As a result, the overall objective lacks the

convex-concave structure required for the existence of a well-defined saddle point. Specifically, a min-max problem admits a stable saddle point only when the objective is convex in the minimization variables and concave in the maximization variables. In our setting, the ℓ_1 norm term is “convex” in both arguments Ξ and Γ , rendering the adversarial maximization over Γ ill-posed. This contrasts with adversarial formulations successfully employed in other domains—such as self-adaptive PINNs [66]—where adaptive weighting is applied to loss terms that are ideally driven to zero (e.g., initial or boundary condition errors). In sparse regression, however, the ℓ_1 penalty should not be driven to zero through increasing Γ unless Ξ is identically zero, making adversarial maximization of the penalty inherently self-defeating. Without a natural saturation point, Γ continues to grow, leading to complete sparsity and failure to recover meaningful terms.

We do acknowledge that there are methods for artificially introducing a saddle point into the min-max loss landscape that can benefit the optimization. For instance, hyperparameters such as weight decay can prevent excessive growth of Γ leading to a maximum sparsity enforcement. Alternative optimization algorithms, as opposed to ADAM, such as optimistic gradient descent [79] or optimistic mirror descent [80], introduce negative gradient components that can also yield saddle points for optimization. However, in the sparse regression setting, these additional parameters proved to be sensitive and problem-specific through testing, limiting the robustness of the min-max formulation. Thus, even though the min-max optimization with these additional enhancements could solve these problems, we discourage its utilization.

In contrast, the joint-minimization formulation avoids adversarial dynamics altogether, where the gradients $\nabla_{\Gamma}L$ and $\nabla_{\Xi}L$ act cooperatively to minimize the same objective. As a result, the gradient dynamics differ fundamentally. The parameter Γ no longer attempts to maximize the sparsity penalty but instead acts as a cooperative scaling factor on the ℓ_1 regularization term. The minimization, in combination with the absolute value function on Γ , will always drive Γ toward 0, differing from the unbounded growth in the min-max formulation. This movement toward zero is what allows for machine-precision accuracy in nonlinear parameters as the impact of the ℓ_1 penalty is reduced and as $\Gamma = 0$, we reach a least-squares problem. We can also verify that Γ will stabilize at nonzero values for incorrect candidate functions from the above $\nabla_{\Gamma}L$ equation and the thresholding step, preventing complete removal of the ℓ_1 penalty. When an incorrect candidate function is zeroed out due to sequential thresholding, the gradient becomes exactly zero, preventing further updates to $\nabla_{\Gamma}L$. Therefore, Γ saturates at a nonzero value for the incorrect terms. However, the algorithm allows the Γ values associated with correct candidate terms to continue moving toward 0, further increasing the relative impact of the ℓ_2 loss, while preventing the discovered model sparsity from decreasing. It is additionally noteworthy that the trajectory of $|\Gamma|$ does not decrease monotonically, contrary to intuition from the analytical gradient $\nabla_{\Gamma}L$. Instead, it exhibits oscillations over the training epochs as the true sign of Γ flips from positive to negative, since achieving an exact zero value of Γ is practically impossible without explicit clipping. This oscillatory behavior

also further justifies the incorporation of a scheduler (dynamic learning rate) to decrease the learning rate at specific epochs, with the resulting oscillations further scaled by minimum step size of $\eta_{\Gamma} \cdot \epsilon$. For Γ to consistently move closer to zero and thus increase the ℓ_2 impact on remaining model candidates, one of these parameters (ϵ or η_{Γ}) must decrease. Reducing ϵ risks allowing incorrect terms to re-enter the model; therefore, it is preferable to reduce the learning rate η_{Γ} instead.

Although this joint-minimization approach may appear mathematically ambiguous, it is consistent with the design of an adaptive weighting strategy—penalizing only incorrect terms while sparing correct ones. The sparsity parameters converge to zero for the correct terms (facilitating machine precision), whereas they saturate at a nonzero value for the incorrect terms. While this approach avoids the ill-posedness associated with adversarial optimization, it may reduce the efficacy of sparsity enforcement, particularly when the regularization parameter λ is small or the data fidelity term dominates the objective. In such cases, the learned weights Γ may fail to sufficiently penalize small-magnitude components in Ξ , resulting in denser models. Nevertheless, the joint formulation offers a more stable and tractable optimization landscape, making it better suited for robust and interpretable sparse regression in practice.

2. Empirical analysis of the optimization frameworks

To substantiate the theoretical discussion, we investigate the sparsity inducing Γ trajectory under both optimization formulations across all numerical examples considered in this work. Although the min-max formulation lacks a well-defined saddle point, we employ the *optimistic* gradient ascent/descent method—an algorithmic strategy developed to improve stability for adversarial training [81]. This method introduces a negative momentum term based on past gradients into the current updates, effectively mitigating oscillations and reducing the risk of divergence. Specifically, optimistic ADAM [79] is used for the maximization step, while vanilla ADAM is applied for the minimization step of the objective function.

A comparative analysis of the two objective formulations with respect to sparsity enforcement is presented in Fig. 8, where the evolution trajectories of $|\Gamma|$ are plotted for two randomly selected candidate functions: one corresponding to a correct term and the other to an incorrect term. Across all experiments, the sparsity value of Γ associated with the incorrect term remains consistently higher than that of the correct term. This behavior aligns with the design of the adaptive weighting strategy, which aims to penalize incorrect candidates more strongly while applying minimal or no penalty to correct ones. Furthermore, despite identical initializations of Γ , the min-max formulation imposes stronger sparsity constraints than the joint-minimization method, resulting in generally larger Γ values. Notably, this led to no candidate functions being selected under the min-max formulation, whereas the joint-minimization approach successfully identified correct terms as discussed in Sec. III. We further observe that replacing optimistic ADAM with vanilla ADAM for the maximization step intensifies sparsity enforcement; however, these results are omitted for brevity.

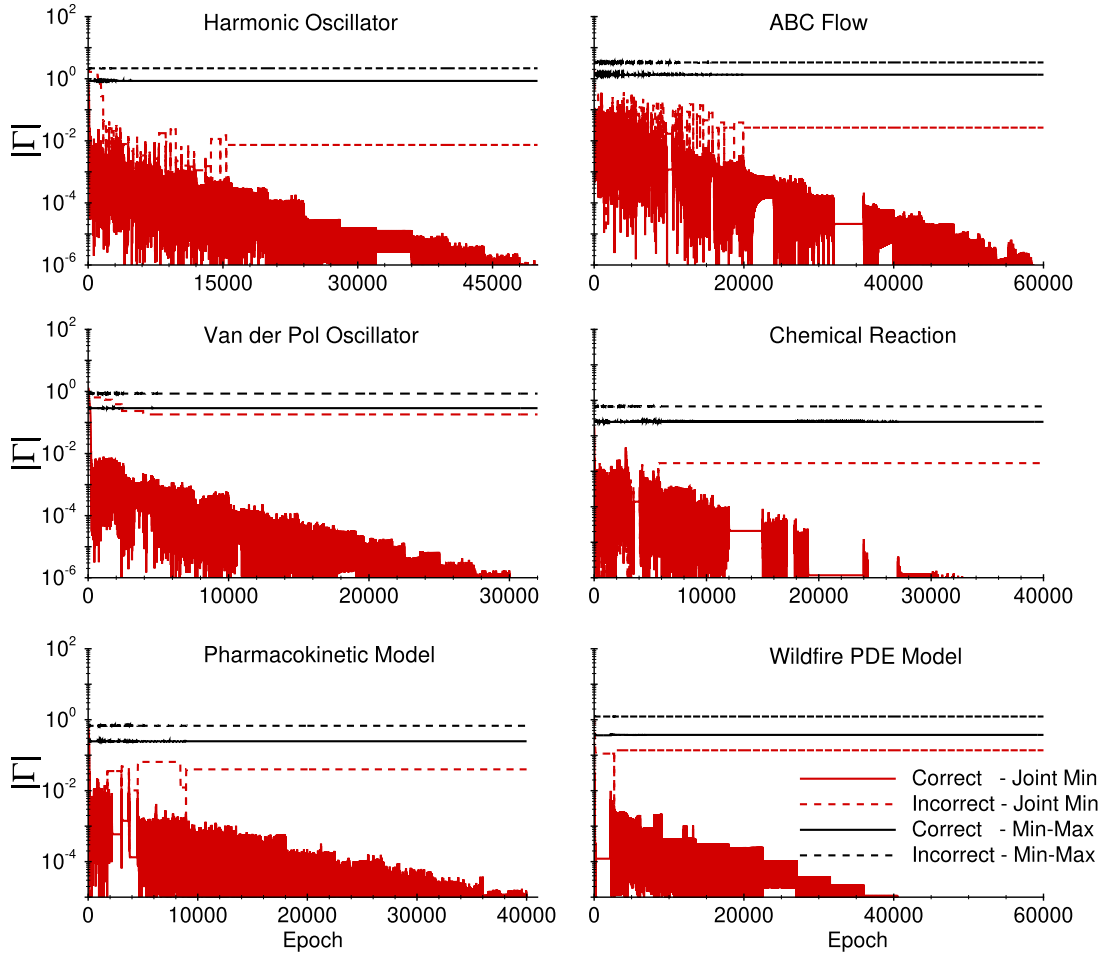


FIG. 8. Evolution of $|\Gamma|$ over training epochs for all numerical examples considered in this work. Each plot shows the $|\Gamma|$ trajectory for both randomly selected correct and incorrect terms, under both objective functions: joint-minimization and min-max formulations.

While the min-max formulation may appear advantageous in theory, its lack of convex-concave structure results in unstable adversarial dynamics that hinder practical sparse regression. The joint-minimization framework, by avoiding these adversarial effects, provides a more stable and reliable optimization landscape that still enforces sparsity effectively. Thus, it serves as the preferred strategy within the ADAM-SINDy framework throughout this work.

APPENDIX C: SUPPLEMENTARY EXPERIMENTS

In this Appendix, we demonstrate the performance of ADAM-SINDy through three supplementary experiments: (1) a methodological comparison with symbolic regression, (2) an evaluation of robustness under varying levels of Gaussian noise, and (3) evaluation in systems involving multiple *noninteger* polynomial exponents.

1. Comparison with symbolic regression

In this Appendix, we compare the performance of ADAM-SINDy with symbolic regression using the PySR framework [82] on the ODE examples considered here. Overall, symbolic regression accurately recovers the governing equations for all parameterized dynamical system examples except for the Van

der Pol oscillator, which involves a noninteger exponent in its nonlinear damping term, whereas ADAM-SINDy maintains robust performance across all numerical examples. The ground-truth model for the Van der Pol oscillator is

$$\dot{x} = y, \quad \dot{y} = 0.01(1 - x^{2.15})y - x.$$

This noninteger exponent presents a challenge for symbolic regression, which typically favors integer-power polynomials, whereas ADAM-SINDy remains robust under such conditions. For this system, PySR produces either oversimplified or highly complex expressions. A simplified but inaccurate model with a mean squared error (MSE) of 6×10^{-4} is

$$\dot{x} = y, \quad \dot{y} = -1.0002x,$$

whereas a more accurate model (MSE of 7×10^{-8}) is considerably more complex:

$$\begin{aligned} \dot{y} = & x[y(x^2(x^2(0.00103 - 7.12 \times 10^{-5}x^2) \\ & - 0.00779) - 0.00294) - 1.0] + 0.01x. \end{aligned}$$

These results highlight a fundamental trade-off in symbolic regression, as achieving minimal numerical loss can lead to bloated, less interpretable models, especially for systems involving noninteger (fractional) polynomial exponents. Such limitations indicate a potential direction for future research.

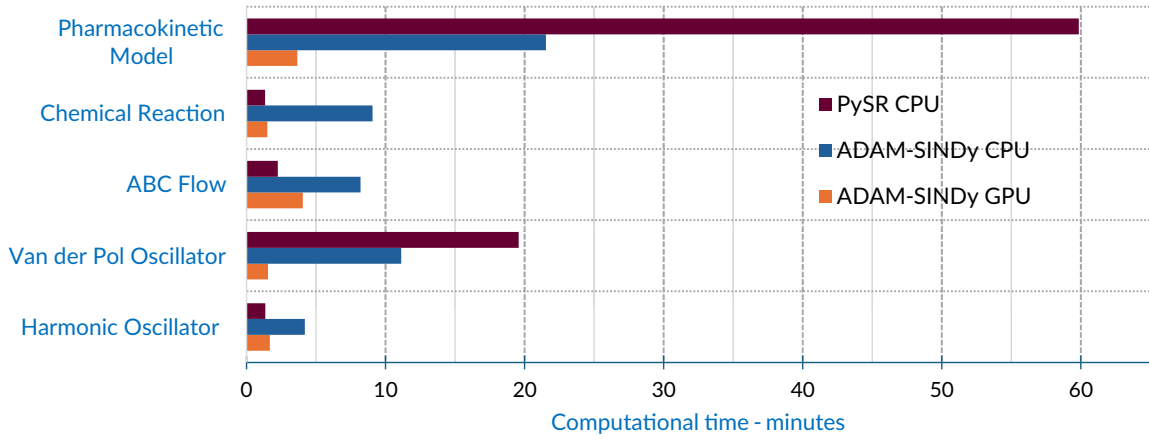


FIG. 9. Comparison of wall-clock computational time for identifying the governing dynamical systems using PySR (CPU) and ADAM-SINDy (CPU and GPU).

For the computational efficiency experiments on PySR and ADAM-SINDy, an MSE (l_2 norm) threshold of 10^{-6} was used as the stopping criterion, ensuring that both optimizations terminated even before the predefined fixed number of epochs, and enabling a one-to-one fair assessment of optimization performance. Although ADAM-SINDy supports GPU-based acceleration, all experiments were also conducted on the same CPU hardware (Intel i7-12700F, 32 GB DDR4, serial computation) to eliminate hardware-induced variability and ensure an unbiased comparison.

Figure 9 presents the wall-clock time required by each method to identify the governing dynamical systems. The results indicate that PySR achieves superior computational efficiency on CPU for certain dynamical systems, namely chemical reaction kinetics, the harmonic oscillator, and the ABC flow. However, PySR requires significantly more CPU time for more complex systems, including the pharmacokinetic model and the Van der Pol oscillator, and fails to recover the correct governing model in the latter case. When run on a GPU (NVIDIA RTX 3060 Ti), ADAM-SINDy demonstrates substantial improvements in computational efficiency, performing comparably to PySR in scenarios where PySR previously had an advantage. Nevertheless, PySR exhibits diminished performance for the Van der Pol oscillator, highlighting a scenario in which its efficiency advantage does not translate into accurate model discovery.

2. Multiple Fractional-order Polynomial Exponents

In this experiment, we extend the analysis to dynamical systems containing multiple noninteger (fractional) polynomial exponents, in contrast to the single fractional exponent case considered in the standard Van der Pol oscillator. This setting provides a rigorous test of ADAM-SINDy's ability to discover governing equations that lie beyond conventional integer-order polynomial libraries. Specifically, we consider the following ground-truth model:

$$\begin{aligned}\dot{x} &= y, \\ \dot{y} &= 0.01y - x^{2.15}y^{1.25} - x,\end{aligned}$$

which incorporates two distinct fractional exponents in the nonlinear term. Applying PySR to this system yields either oversimplified models that fail to capture the true dynamics or highly complex models that are difficult to interpret. For example, a simplified but less accurate model, with MSE of 7×10^{-4} , is given by

$$\dot{x} = y, \quad \dot{y} = -1.0003x,$$

whereas a more accurate model, with MSE of 9×10^{-7} , takes the following complex form:

$$\begin{aligned}\dot{y} &= x - 0.9999 + y[x(-0.0005y^2 - 0.0052 + 0.0003x^3) \\ &\quad - 0.0043] + 0.0099.\end{aligned}$$

In contrast, ADAM-SINDy accurately recovers the underlying dynamical system, even in the presence of multiple fractional exponents, by leveraging the master library defined in Eq. (10), with $\mathcal{A}$ treated as trainable—the same setup used for the van der Pol oscillator in Sec. III B. To further evaluate its performance, we considered additional fractional powers in the y term, specifically 1.05, 1.15, and 1.30. ADAM-SINDy consistently identifies both the correct fractional-order structure and the relevant candidate functions. These results underscore a key advantage of ADAM-SINDy over symbolic regression methods such as PySR, particularly when modeling systems with fractional-order dynamics.

3. Robustness to Gaussian noise

In this Appendix, we evaluate the robustness of ADAM-SINDy framework in comparison with the SINDy-W/o method under varying levels of additive white Gaussian noise. The assessment considers both the recovery of the true state trajectories and the identification of the underlying dynamical system. In the idealized SINDy-W setting, least-squares regression would exhibit greater resilience to noise because it assumes prior knowledge of the true candidate functions and nonlinear parameters; thus, a one-to-one comparison with the SINDy-W method here is inappropriate. The noise experiments are performed by contaminating the Van der Pol oscillator example with zero-mean Gaussian noise of varying standard deviation.

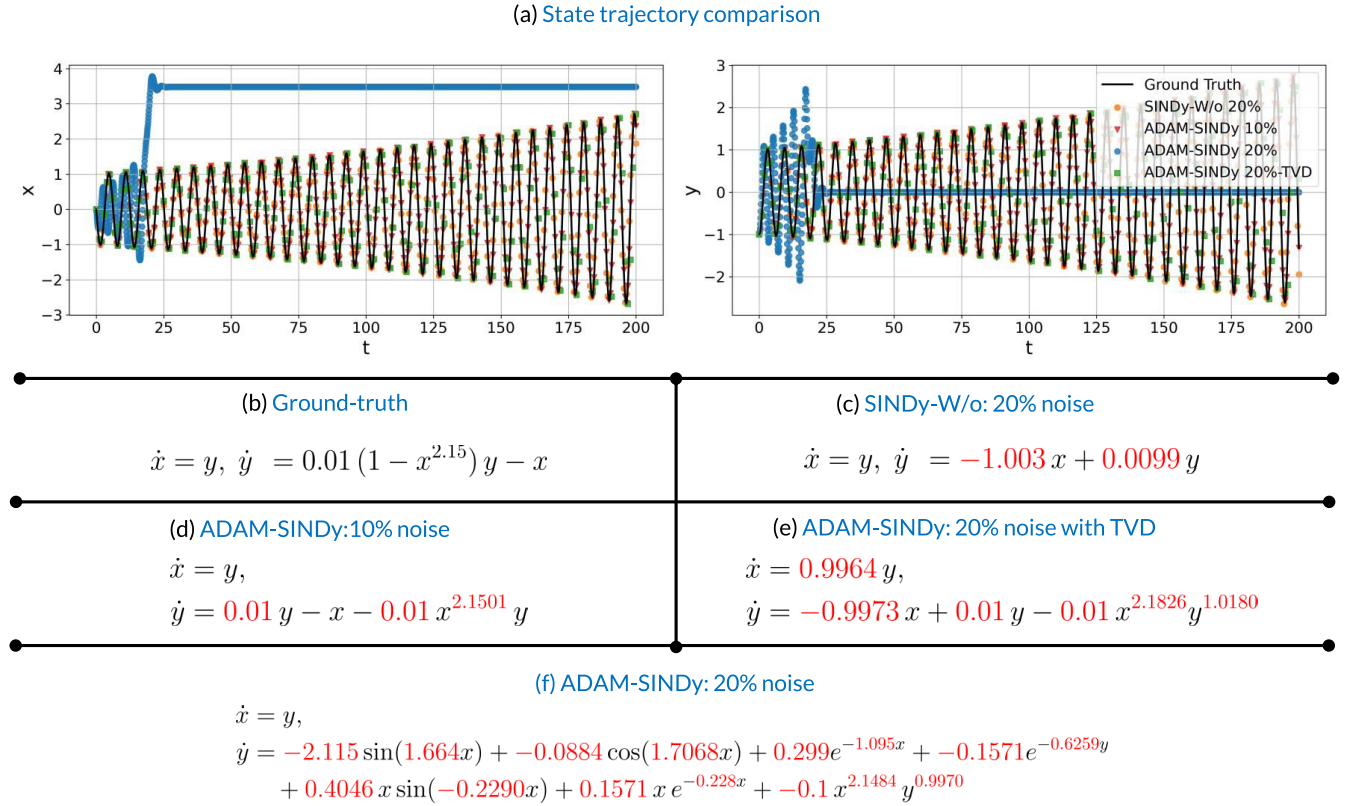


FIG. 10. Noise-level experiments on ADAM-SINDy and SINDy-w/o methods using the Van der Pol system. White Gaussian noise with zero mean and standard deviations of 10% and 20% was added to the data. The ADAM-SINDy with TVD corresponds to the case in which the noisy data are preprocessed using TVD before optimization. (a) Comparison of the ground-truth state trajectory with the two methodologies under different noise levels. Dynamical system results: (b) Ground truth, (c) SINDy-w/o with 20% noise, (d) ADAM-SINDy with 10% noise, (e) ADAM-SINDy with 20% noise preprocessed with TVD, and (f) ADAM-SINDy with 20% noise without preprocessing. The red-highlighted numerical values indicate the optimized parameters obtained.

Figure 10 presents a comparison of the state variable trajectories of the ground-truth Van der Pol system with those identified by the ADAM-SINDy and SINDy-w/o methods under different noise levels, illustrating the impact of noise and the effectiveness of preprocessing with Total Variation Denoising (TVD). Although the SINDy-w/o method cannot recover the exact ground-truth equations, it demonstrates remarkable robustness to noise up to 60%, producing the same equation as shown in Fig. 10(c) and approximating the ground-truth trajectory as closely as possible. For comparison, only the 20% noise case is presented. ADAM-SINDy accurately identifies the ground-truth system at a 10% noise level, with only minor deviations in polynomial exponents at the fourth decimal place, as shown in Fig. 10(d). At a

higher noise level of 20%, however, the method produces a more complex system [Fig. 10(f)], reducing interpretability. All ADAM-SINDy experiments here were conducted without modifying hyperparameters and tuning them may help mitigate this increased complexity, which is left for future work. To address high-noise scenarios, the 20% noisy data were preprocessed using TVD prior to applying ADAM-SINDy. This approach produces a dynamical system that faithfully approximates the ground-truth equation, *correctly* identifies the relevant candidate functions [Fig. 10(e)], and introduces expected discrepancies in both polynomial coefficients and exponents. These results demonstrate that TVD preprocessing enhances both the robustness and interpretability of ADAM-SINDy under challenging noise conditions.

- [1] L. Ljung, Perspectives on system identification, *Annu. Rev. Control* **34**, 1 (2010).
- [2] P. O. Van and B. M. De, *Subspace Identification for Linear Systems: Theory—Implementation—Applications* (Springer Science & Business Media, New York, USA, 2012).
- [3] O. Nelles, *Nonlinear Dynamic System Identification* (Springer, Berlin, Heidelberg, 2020).
- [4] F. Lauer and G. Bloch, *Hybrid System Identification: Theory and Algorithms for Learning Switching Models* (Springer, Switzerland AG, 2018), Vol. 478.
- [5] P. F. Pai and A. N. Palazotto, HHT-based nonlinear signal processing method for parametric and non-parametric identification of dynamical systems, *Int. J. Mech. Sci.* **50**, 1619 (2008).
- [6] S. N. Kumpati and P. Kannan, Identification and control of dynamical systems using neural networks, *IEEE Trans. Neural Networks* **1**, 4 (1990).
- [7] J. A. K. Suykens, J. P. L. Vandewalle, and B. L. M. De, *Artificial Neural Networks for Modeling and Control of Non-linear Systems* (Springer Science & Business Media, New York, USA, 2012).

- [8] A. C. Gondhalekar, E. P. Petrov, and M. Imregun, Parameters identification for nonlinear dynamic systems via genetic algorithm optimization, *J. Comput. Nonlinear Dyn.* **4**, 041002 (2009).
- [9] M. Quade, M. Abel, K. Shafi, R. K. Niven, and B. R. Noack, Prediction of dynamical systems by symbolic regression, *Phys. Rev. E* **94**, 012214 (2016).
- [10] G.-J. Both, S. Choudhury, P. Sens, and R. Kusters, DeepMoD: Deep learning for model discovery in noisy data, *J. Comput. Phys.* **428**, 109985 (2021).
- [11] A. Okunev, E. Burnaev, and J. Wang, Ordinary differential equation identification based on sparse Bayesian learning and collaborative neurodynamic optimization, in *2021 6th International Conference on Communication, Image and Signal Processing (CCISP)* (IEEE, Chengdu, China, 2021), pp. 75–79.
- [12] A. Okunev and E. Burnaev, Nonlinear system identification via sparse Bayesian regression based on collaborative neurodynamic optimization, *J. Inverse Ill-Posed Probl.* **32**, 1161 (2024).
- [13] R. Stephany and C. Earls, PDE-READ: Human-readable partial differential equation discovery using deep learning, *Neural Networks* **154**, 360 (2022).
- [14] P. Rajendra and V. Brahmajirao, Modeling of dynamical systems through deep learning, *Biophys. Rev.* **12**, 1311 (2020).
- [15] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K. R. Müller, Explaining deep neural networks and beyond: A review of methods and applications, *Proc. IEEE* **109**, 247 (2021).
- [16] A. Thampi, *Interpretable AI: Building Explainable Machine Learning Systems* (Simon and Schuster, Shelter Island, NY, 2022).
- [17] X. Zhong, B. Gallagher, S. Liu, B. Kailkhura, A. Hiszpanski, and T. Y. J. Han, Explainable machine learning in materials science, *npj Comput. Mater.* **8**, 204 (2022).
- [18] J. R. Koza, Genetic programming as a means for programming computers by natural selection, *Stat. Comput.* **4**, 87 (1994).
- [19] J. Bongard and H. Lipson, Automated reverse engineering of nonlinear dynamical systems, *Proc. Natl. Acad. Sci. USA* **104**, 9943 (2007).
- [20] M. Schmidt and H. Lipson, Symbolic regression of implicit equations, in *Genetic Programming Theory and Practice VII* (Springer, New York, USA, 2009), pp. 73–85.
- [21] G. Dick, Bloat and generalisation in symbolic regression, in *Asia-Pacific Conference on Simulated Evolution and Learning*, edited by G. Dick, W. N. Browne, P. Whigham, M. Zhang, L. T. Bui, H. Ishibuchi, Y. Jin, X. Li, Y. Shi, P. Singh, K. C. Tan, and K. Tang (Springer, Switzerland, 2014), pp. 491–502.
- [22] M. Castelli, L. Trujillo, L. Vanneschi, S. Silva, E. Z-Flores, and P. Legrand, Geometric semantic genetic programming with local search, in *Proceedings of the 2015 annual Conference on Genetic and Evolutionary Computation*, edited by S. Silva (Association for Computing Machinery, New York, United States, Madrid, Spain, 2015), pp. 999–1006.
- [23] Q. U. Nguyen and T. H. Chu, Semantic approximation for reducing code bloat in genetic programming, *Swarm Evol. Comput.* **58**, 100729 (2020).
- [24] L. F. dal Piccol Sotto and V. V. de Melo, Studying bloat control and maintenance of effective code in linear genetic programming for symbolic regression, *Neurocomputing* **180**, 79 (2016).
- [25] S. Razavi and E. R. Gamazon, Neural-network-directed genetic programmer for discovery of governing equations, [arXiv:2203.08808](https://arxiv.org/abs/2203.08808).
- [26] T. Sauer, J. A. Yorke, and M. Casdagli, Embedology, *J. Stat. Phys.* **65**, 579 (1991).
- [27] F. Takens, Detecting strange attractors in turbulence, in *Dynamical Systems and Turbulence, Warwick 1980: Proceedings of a Symposium Held at the University of Warwick 1979/80*, edited by D. R. and L.-S. Young (Springer, Berlin, Heidelberg, 2006), pp. 366–381.
- [28] J. B. Tenenbaum, V. D. Silva, and J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science* **290**, 2319 (2000).
- [29] A. Singer, R. Erban, I. G. Kevrekidis, and R. R. Coifman, Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps, *Proc. Natl. Acad. Sci. USA* **106**, 16090 (2009).
- [30] S. T. Roweis and L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science* **290**, 2323 (2000).
- [31] T. McConaghy, FFX: Fast, scalable, deterministic symbolic regression technology, in *Genetic Programming Theory and Practice IX*, edited by R. Riolo, E. Vladislavleva, and J. H. Moore (Springer, New York, NY, USA, 2011), pp. 235–260.
- [32] B. Burlacu, G. Kronberger, and M. Kommenda, Operon C++ an efficient genetic programming framework for symbolic regression, in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, edited by Carlos Artemio Coello Coello (Association for Computing Machinery, New York, United States, Cancún Mexico, 2020), pp. 1562–1570.
- [33] G. Kronberger, L. Kammerer, and M. Kommenda, Identification of dynamical systems using symbolic regression, in *Computer Aided Systems Theory—EUROCAST 2019: 17th International Conference, Las Palmas de Gran Canaria, Spain, February 17–22, 2019, Revised Selected Papers, Part I 17*, edited by R. Moreno-Díaz, A. Quesada-Arencibia, and F. Pichler (Springer, Las Palmas de Gran Canaria, Spain, 2020), pp. 370–377.
- [34] M. Schmidt and H. Lipson, Distilling free-form natural laws from experimental data, *Science* **324**, 81 (2009).
- [35] H. Oh, R. Amici, G. Bomarito, S. Zhe, R. M. Kirby, and J. Hochhalter, Inherently interpretable machine learning solutions to differential equations, *Eng. Comput.* **40**, 2349 (2024).
- [36] S.-M. Udrescu and M. Tegmark, AI Feynman: A physics-inspired method for symbolic regression, *Sci. Adv.* **6**, eaay2631 (2020).
- [37] J. Dong and J. Zhong, Recent advances in symbolic regression, *ACM Comput. Surv.* **57**, 1 (2025).
- [38] N. Makke and S. Chawla, Interpretable scientific discovery with symbolic regression: A review, *Artif. Intell. Rev.* **57**, 2 (2024).
- [39] W. X. Wang, R. Yang, Y. Lai, V. Kovanis, and C. Grebogi, Predicting catastrophes in nonlinear dynamical systems by compressive sensing, *Phys. Rev. Lett.* **106**, 154101 (2011).
- [40] H. Schaeffer, R. Caflisch, C. D. Hauck, and S. Osher, Sparse dynamics for partial differential equations, *Proc. Natl. Acad. Sci. USA* **110**, 6634 (2013).
- [41] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proc. Natl. Acad. Sci. USA* **113**, 3932 (2016).
- [42] J. Loiseau and S. L. Brunton, Constrained sparse Galerkin regression, *J. Fluid Mech.* **838**, 42 (2018).
- [43] M. Dam, M. Brøns, J. Juul Rasmussen, V. Naulin, and J. S. Hesthaven, Sparse identification of a predator-prey system from

- simulation data of a convection model, *Phys. Plasmas* **24**, 022310 (2017).
- [44] S. Beetham and J. Capece de Almeida, Formulating turbulence closures using sparse regression with embedded form invariance, *Phys. Rev. Fluids* **5**, 084611 (2020).
- [45] A. A. Kaptanoglu, L. Zhang, Z. G. Nicolaou, U. Fasel, and S. L. Brunton, Benchmarking sparse system identification with low-dimensional chaos, *Nonlinear Dyn.* **111**, 13143 (2023).
- [46] L. Boninsegni, F. Nüske, and C. Clementi, Sparse learning of stochastic dynamical equations, *J. Chem. Phys.* **148**, 241723 (2018).
- [47] E. Kaiser, J. N. Kutz, and S. L. Brunton, Sparse identification of nonlinear dynamics for model predictive control in the low-data limit, *Proc. R. Soc. A* **474**, 20180335 (2018).
- [48] K. Kaheman, J. N. Kutz, and S. L. Brunton, Sindy-PI: A robust algorithm for parallel implicit sparse identification of nonlinear dynamics, *Proc. R. Soc. A* **476**, 20200279 (2020).
- [49] P. Goyal and P. Benner, Discovery of nonlinear dynamical systems using a Runge–Kutta inspired dictionary-based sparse regression approach, *Proc. R. Soc. A* **478**, 20210883 (2022).
- [50] A. Arzani, L. Yuan, P. Newell, and B. Wang, Interpreting and generalizing deep learning in physics-based problems with functional linear models, *Eng. Comput.* **41**, 135 (2025).
- [51] A. A. Kaptanoglu, J. L. Callahan, A. Aravkin, C. J. Hansen, and S. L. Brunton, Promoting global stability in data-driven models of quadratic nonlinear dynamics, *Phys. Rev. Fluids* **6**, 094401 (2021).
- [52] J. L. Callahan, J. C. Loiseau, G. Rigas, and S. L. Brunton, Nonlinear stochastic modelling with Langevin regression, *Proc. R. Soc. A* **477**, 20210092 (2021).
- [53] A. Nabeel, A. Karichannavar, S. Palathingal, J. Jhawar, D. B. Brückner, D. R. M., and V. Guttal, Discovering stochastic dynamical equations from ecological time series data, *Am. Nat.* **205**, E100 (2025).
- [54] A. Gerardos and P. Ronceray, Principled model selection for stochastic dynamics, *Phys. Rev. Lett.* **135**, 167401 (2025).
- [55] S. Zhang and G. Lin, Robust data-driven discovery of governing physical laws with error bars, *Proc. R. Soc. A* **474**, 20180305 (2018).
- [56] K. Champion, P. Zheng, A. Y. Aravkin, S. L. Brunton, and J. N. Kutz, A unified sparse optimization framework to learn parsimonious physics-informed models from data, *IEEE Access* **8**, 169259 (2020).
- [57] E. J. Candes, M. B. Wakin, and S. P. Boyd, Enhancing sparsity by reweighted ℓ_1 minimization, *J. Fourier Anal. Appl.* **14**, 877 (2008).
- [58] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, Iteratively reweighted least squares minimization for sparse recovery, *Commun. Pure Appl. Math.* **63**, 1 (2010).
- [59] M.-J. Lai, Y. Xu, and W. Yin, Improved iteratively reweighted least squares for unconstrained smoothed ℓ_q minimization, *SIAM J. Numer. Anal.* **51**, 927 (2013).
- [60] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer, New York, USA, 2009), Vol. 2.
- [61] R. Tibshirani, Regression shrinkage and selection via the lasso, *J. R. Stat. Soc. B* **58**, 267 (1996).
- [62] A. Beck and Y. C. Eldar, Sparsity constrained nonlinear optimization: Optimality conditions and algorithms, *SIAM J. Optim.* **23**, 1480 (2013).
- [63] Z. Yang, Z. Wang, H. Liu, Y. C. Eldar, and T. Zhang, Sparse nonlinear regression: Parameter estimation under nonconvexity, in *International Conference on Machine Learning*, edited by M. F. Balcan and K. Q. Weinberger (PMLR, New York, USA, 2016), pp. 2472–2481.
- [64] H. Zou, The adaptive lasso and its oracle properties, *J. Am. Stat. Assoc.* **101**, 1418 (2006).
- [65] R. C. Aster, B. Borchers, and C. H. Thurber, *Parameter Estimation and Inverse Problems* (Elsevier, 2018).
- [66] L. D. McClenney and U. M. Braga-Neto, Self-adaptive physics-informed neural networks, *J. Comput. Phys.* **474**, 111722 (2023).
- [67] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* **378**, 686 (2019).
- [68] D. Oettinger and G. Haller, An autonomous dynamical system captures all lcss in three-dimensional unsteady flows, *Chaos* **26**, 103111 (2016).
- [69] N. Boddupalli, T. Matchen, and J. Moehlis, Symbolic regression via neural networks, *Chaos* **33**, 083150 (2023).
- [70] P. Gray and S. K. Scott, *Chemical Oscillations and Instabilities: Non-Linear Chemical Kinetics* (Oxford University Press, 1990).
- [71] N. A. Daryakenari, M. De Florio, K. Shukla, and G. E. Karniadakis, AI-Aristotle: A physics-informed framework for systems biology gray-box identification, *PLoS Comput. Biol.* **20**, e1011916 (2024).
- [72] P. Macheras and A. Iliadis, *Modeling in Biopharmaceutics, Pharmacokinetics and Pharmacodynamics: Homogeneous and Heterogeneous Approaches* (Springer, Switzerland, 2016), Vol. 30.
- [73] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, Data-driven discovery of partial differential equations, *Sci. Adv.* **3**, e1602614 (2017).
- [74] S. Viknesh, A. Tohidi, F. Afghah, R. Stoll, and A. Arzani, Role of flow topology in wind-driven wildfire propagation, *Phys. Fluids* **37**, 076608 (2025).
- [75] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, Adaptive activation functions accelerate convergence in deep and physics-informed neural networks, *J. Comput. Phys.* **404**, 109136 (2020).
- [76] G. M. Correia, V. Niculae, and A. F. Martins, Adaptively sparse transformers, *arXiv:1909.00015*.
- [77] W. D. Fries, X. He, and Y. Choi, LaSDI: Parametric latent space dynamics identification, *Comput. Methods Appl. Mech. Eng.* **399**, 115436 (2022).
- [78] <https://github.com/siva-viknesh/ADAM-SINDy>.
- [79] C. Daskalakis, A. Ilyas, V. Syrgkanis, and H. Zeng, Training GANs with optimism, *International Conference on Learning Representation*, *arXiv:1711.00141*.
- [80] P. Mertikopoulos, B. Lecouat, H. Zenati, C.-S. Foo, V. Chandrasekhar, and G. Piliouras, *Optimistic Mirror Descent in Saddle-point Problems: Going the Extra(-gradient) Mile* (International Conference on Learning Representations, 2019).
- [81] C. Daskalakis and I. Panageas, The limit points of (optimistic) gradient descent in min-max optimization, *Adv. Neural Inf. Process. Syst.* **31** (2018).
- [82] M. Cranmer, Interpretable machine learning for science with PySR and SymbolicRegression.jl, *arXiv:2305.01582*.