

Automatically discovering ordinary differential equations from data with sparse regression

Kevin Egan ¹, Weizhen Li ¹ & Rui Carvalho ^{1,2}✉

Discovering nonlinear differential equations that describe system dynamics from empirical data is a fundamental challenge in contemporary science. While current methods can identify such equations, they often require extensive manual hyperparameter tuning, limiting their applicability. Here, we propose a methodology to identify dynamical laws by integrating denoising techniques to smooth the signal, sparse regression to identify the relevant parameters, and bootstrap confidence intervals to quantify the uncertainty of the estimates. We evaluate our method on well-known ordinary differential equations with an ensemble of random initial conditions, time series of increasing length, and varying signal-to-noise ratios. Our algorithm consistently identifies three-dimensional systems, given moderately-sized time series and high levels of signal quality relative to background noise. By accurately discovering dynamical systems automatically, our methodology has the potential to impact the understanding of complex systems, especially in fields where data are abundant, but developing mathematical models demands considerable effort.

¹Department of Engineering, Durham University, Lower Mountjoy, South Road, Durham DH1 3LE, UK. ²Institute for Data Science, Durham University, South Road, Durham DH1 3LE, UK. ✉email: rui.carvalho@dur.ac.uk

Since Newton discovered the second law of motion, scientists have sought to formulate mathematical models in the form of differential equations that accurately represent natural phenomena. In the past half-century, dynamical systems have been employed in various disciplines such as physics^{1,2}, chemistry³, biology^{4–9}, neuroscience^{10–12}, epidemiology^{13,14}, ecology^{15,16} and environmental sciences^{17,18}. Nonetheless, developing these models remains challenging and typically requires considerable effort from specialists in the relevant fields^{19,20}.

As early as the 1980s, scientists turned to statistical methods to reverse engineer governing equations for nonlinear systems from data²¹. This approach, often referred to as the inverse problem²² or system identification²³, aims to automatically discover mathematical models that accurately represent the inherent dynamics. Building on this foundation, symbolic regression has been instrumental in advancing our ability to develop more interpretable models of complex systems^{24,25}. Sparse regression has emerged as a practical method for this problem, eliminating the time-consuming task of determining equations manually. A remarkable breakthrough is the sparse identification of nonlinear dynamics (SINDy)²⁶, an approach that employs a sparsity-promoting framework to identify interpretable models from data by only selecting the most dominant candidate terms from a high-dimensional nonlinear-function space. This methodology has significantly advanced system identification, serving as a foundational influence for numerous subsequent sparse regression techniques^{27–29}. Over time, SINDy has evolved and expanded its framework, incorporating Bayesian sparse regression³⁰ and ensemble methods to estimate inclusion probabilities³¹, integrating neural networks^{32,33}, and deploying tools to better manage noisy data³⁴.

Among these advancements, a distinct variant, SINDy with AIC, aims to automate the model selection procedure³⁵. This approach uses a grid of sparsity-promoting threshold parameters in conjunction with the Akaike information criterion (AIC) to determine the model that most accurately characterizes the dynamics of a given system. However, it encounters several obstacles that limit its practicality. Key challenges include its dependence on prior knowledge of the governing equations for model validation and identification, as well as the requirement for high-quality measurements given its limited capacity to compute numerical derivatives from previously unseen data. Furthermore, the efficacy of SINDy with AIC has only been demonstrated on data sets generated using specific initial conditions, sufficient observations, and low levels of noise, indicating the need for more comprehensive and rigorous analyses to assess its performance in diverse settings.

While existing methods, such as SINDy, use the Savitzky-Golay filter to both reduce noise and compute numerical derivatives, they require users to manually select the filter parameters^{36–38}. Additionally, effective system identification often hinges on rigorous variable selection methods. To address these concerns, our contribution lies in developing an automated approach that employs a grid to fine-tune the Savitzky-Golay filter parameters and subsequently leverages bootstrapping to estimate confidence intervals and establish the governing terms of the system. As a result, our algorithm significantly improves the accuracy and efficiency of model discovery in low to medium-noise conditions while requiring only the assumptions of model sparsity and the presence of governing terms in the design matrix. We demonstrate the effectiveness of our approach by examining its success rate on synthetic data sets generated from known ordinary differential equations, exploring a range of initial conditions, time series of increasing length, and various noise intensities. Our algorithm automates the discovery of three-dimensional systems

more efficiently than SINDy with AIC, achieving higher identification accuracy with moderately sized data sets and high signal quality.

Results

Modeling systems of ODEs with linear regression. Ordinary differential equations (ODEs) are often used to model dynamical systems in the form of

$$\frac{d}{dt}x_j(t) = \dot{x}_j(t) = f_j(x(t)) \quad j = 1, \dots, m, \quad (1)$$

where $x = x(t) = (x_1(t) x_2(t) \dots x_m(t))^T \in \mathbb{R}^m$ is a state space vector, and $f(x(t)) : \mathbb{R}^m \rightarrow \mathbb{R}^m$ describes the system's evolution in time³⁹. We approximate the dynamics symbolically by

$$\dot{x}_j \approx \theta_F^T(x) \beta_j, \quad j = 1, \dots, m, \quad (2)$$

where $\beta_j \in \mathbb{R}^p$ is a sparse coefficient vector of system parameters and $\theta_F(x)$ is a feature vector containing p symbolic functions, each representing an ansatz that we can use to describe the dynamics.

To identify the system from data, we first construct a state matrix $\tilde{\mathbf{X}} \in \mathbb{R}^{n \times m}$ from measurements of $x(t)$ taken at times t_1, t_2, \dots, t_m , then apply the Savitzky-Golay filter⁴⁰ to smooth each column $\tilde{\mathbf{x}}_j = SG(\tilde{\mathbf{x}}_j)$ and calculate the derivative $\dot{\tilde{\mathbf{x}}}_j$. We next consolidate \mathbf{X} and $\dot{\mathbf{X}}$ and build the block design matrix $\Theta(\mathbf{X}) \in \mathbb{R}^{n \times p}$:

$$\Theta(\mathbf{X}) = \begin{pmatrix} | & | & | & & | & | \\ \mathbf{1} & \mathbf{X} & \mathbf{X}^{[2]} & \dots & \mathbf{X}^{[d]} & \Phi(\mathbf{X}) \\ | & | & | & & | & | \end{pmatrix}, \quad (3)$$

where $\mathbf{X}^{[i]}$ for $i = 1, \dots, d$ is a matrix whose column vectors denote all monomials of order i in $x(t)$, and $\Phi(\mathbf{X})$ can contain nonlinear functions such as trigonometric, logarithmic, or exponential²⁶.

Finally, we perform a linear regression with the above matrices:

$$\dot{\mathbf{X}} = \Theta(\mathbf{X})\mathbf{B} + \mathbf{E}, \quad (4)$$

where $\mathbf{B} \in \mathbb{R}^{p \times m}$ and $\mathbf{E} \in \mathbb{R}^{n \times m}$ denote the coefficient and residual matrices, respectively.

Automatic regression for governing equations (ARGOS). Our approach, ARGOS, aims to automatically identify interpretable models that describe the dynamics of a system by integrating machine learning with statistical inference. As illustrated in Fig. 1, our algorithm comprises several key phases to solve the system in Eq. (4). These include data smoothing and numerical approximation of derivatives, as well as the use of bootstrap sampling with sparse regression to develop confidence intervals for variable selection.

In the first phase, we employ the Savitzky-Golay filter, which fits a low-degree polynomial to a local window of data points, reducing noise in the state matrix and approximating the derivative numerically⁴⁰. The Savitzky-Golay filter enables us to mitigate the effects of noise while maintaining our goal of automating the entire identification process. To optimize the filter, we set polynomial order $o = 4$ and construct a grid of window lengths l ⁴¹. For each column of the noisy state matrix $\tilde{\mathbf{X}}$, we find the optimal l^* that minimizes the mean squared error between each noisy signal $\tilde{\mathbf{x}}_j$ and its smoothed counterpart \mathbf{x}_j (Algorithm 1 in Supplementary Note 1).

Following smoothing and differentiation, we construct the design matrix $\Theta(\mathbf{X})$ with monomials up to the d -th degree and extract the columns of $\dot{\mathbf{X}}$ and \mathbf{B} from Eq. (4). As the noise in the data increases, the mean squared error between \mathbf{x}_j and $\tilde{\mathbf{x}}_j$ increases

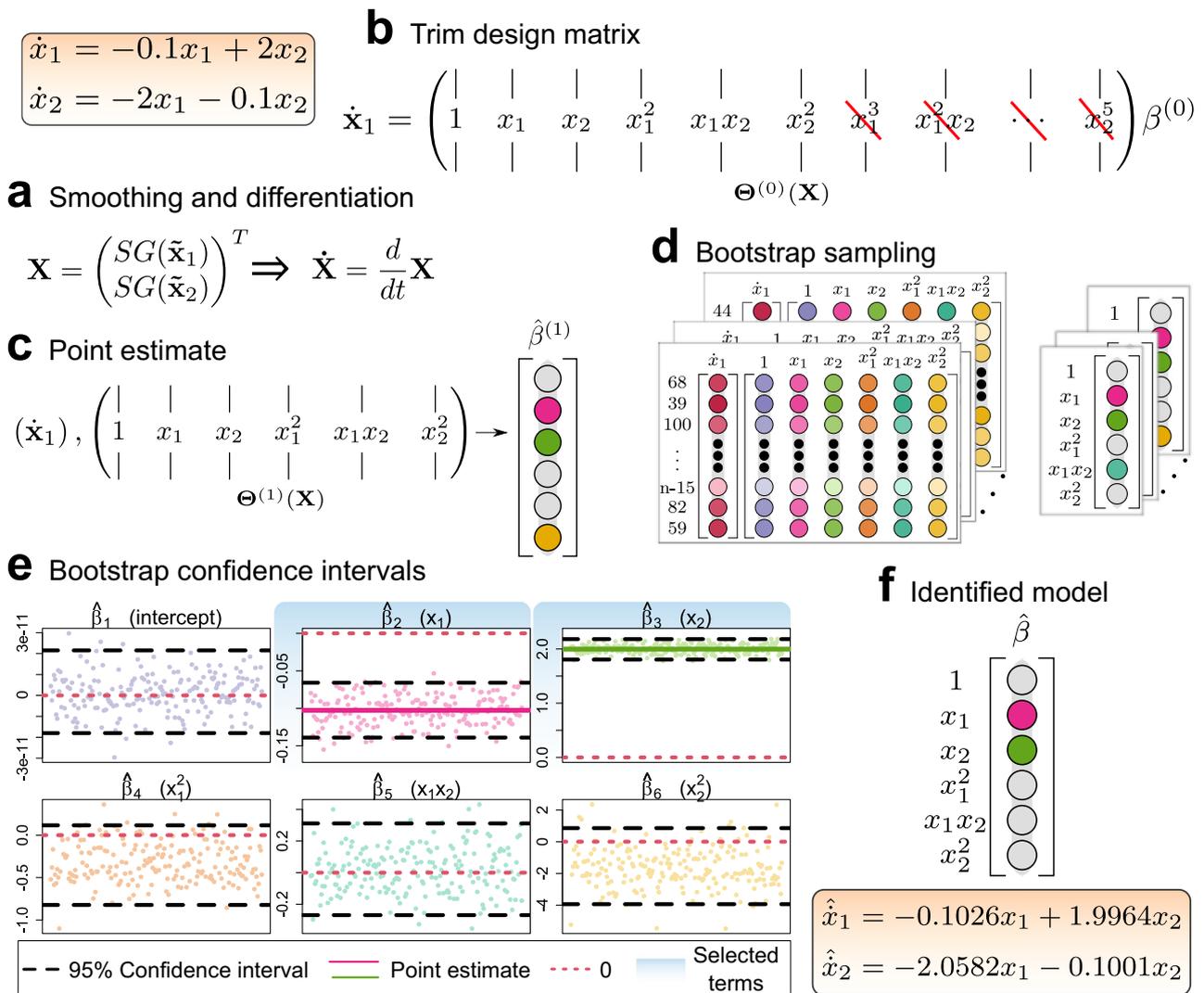


Fig. 1 Automatic regression for governing equations (ARGOS). This example illustrates the process of identifying the \dot{x}_j equation of a two-dimensional damped oscillator with linear dynamics. We first **(a)** smooth each noisy state vector in \mathbf{X} and calculate the derivative $\dot{\mathbf{x}}$, using the Savitzky-Golay filter. Next, we **(b)** construct the design matrix $\Theta^{(0)}(\mathbf{X})$, containing the observations $x(t)$ and their interaction terms up to monomial degree $d = 5$ —see Eq. (3). Following Eq. (5), we perform sparse regression using either the lasso or the adaptive lasso and determine the highest-order monomial degree with a nonzero coefficient in the estimate $\hat{\beta}^{(0)}$ (in this example, we detect terms up to $d = 2$). We then trim the design matrix to include only terms up to this order and **(c)** perform sparse regression again with the trimmed design matrix and the previously used algorithm (lasso or adaptive lasso), apply ordinary least squares (OLS) on the subset of selected variables, and determine the final $\hat{\beta}^{(1)}$ point estimates. Finally, we **(d)** employ bootstrap sampling to obtain 2000 sample estimates and **(e)** develop 95% bootstrap confidence intervals to **(f)** identify the $\hat{\beta}$ by selecting the coefficients whose intervals contain the point estimate but do not include zero.

exponentially with the signal-to-noise ratio (SNR), consequently distorting the columns of $\Theta(\mathbf{X})$ (Supplementary Fig. S7). Using $\Theta(\mathbf{X})$, we formulate a regression problem to identify the governing equations of each component of the system:

$$\dot{\mathbf{x}}_j = \Theta(\mathbf{X})\beta_j + \epsilon_j, \quad j = 1, \dots, m. \quad (5)$$

We then apply either the least absolute shrinkage and selection operator (lasso)⁴² or the adaptive lasso⁴³ during the model selection process (Algorithm 2 in Supplementary Note 1). Both algorithms add the ℓ_1 penalty to the ordinary least squares regression (OLS) estimate, shrinking coefficients to zero. This allows for the selection of the nonzero terms for parameter and model inference.

After identifying an initial sparse regression estimate of β_j in Eq. (5), we trim the design matrix to include only monomial terms up to the highest-order variable with a nonzero coefficient in the estimate. Using the updated design matrix, we reapply the sparse

regression algorithm and employ a grid of thresholds to develop a subset of models, each containing only coefficients whose absolute values exceed their respective thresholds (see the Algorithm implementation subsection in the Methods). Next, we perform OLS on the selected variables of each subset to calculate unbiased coefficients and determine the point estimates from the regression model with the minimum Bayesian information criterion (BIC)⁴⁴. As a final step, we bootstrap this sparse regression process with the trimmed design matrix to obtain 2000 sample estimates⁴⁵. We then construct 95% bootstrap confidence intervals using these sample estimates and identify a final model consisting of variables whose confidence intervals do not include zero and whose point estimates lie within their respective intervals.

Assessing ARGOS systematically. To evaluate the effectiveness of our approach, we expanded several well-known ODEs using

100 random initial conditions, emulating real-world settings where we cannot select these initial values. We then generated data sets with varying time series lengths n and SNRs (see the Building the data sets and tests subsection in the Methods) before introducing a success rate metric, defined as the proportion of instances where an algorithm identified the correct terms of the governing equations from a given dynamical system. This metric allowed us to quantitatively measure the performance of an algorithm across different dynamical systems, as well as different SNR and n values (see Supplementary Tables S1 and S2). Figure 2 highlights success rates exceeding 80%, demonstrating that our method consistently outperformed SINDy with AIC in identifying the underlying system of the data. We accurately represented linear systems with less than 800 data points and medium SNRs, underscoring the method's ability to handle straightforward dynamics (see Supplementary Note 2). Even with only moderately-sized data sets or medium SNRs, we successfully identified three out of five of the two-dimensional ODEs using the lasso with ARGOS, showcasing the effectiveness of integrating classic statistical learning algorithms within our framework^{46,47}. The adaptive lasso was able to identify the non-linear ODEs in three dimensions with higher accuracy than the other algorithms tested (see Supplementary Note 3)⁴⁸. These results suggest that the adaptive lasso is suitable for identifying non-linear ODEs in higher dimensional systems. In practice, we recommend employing the adaptive lasso for such systems, while the lasso can serve as a valuable initial exploration tool for most cases.

The systematic analysis, presented in Fig. 2, emphasized the efficacy of our approach as n and SNR increased. The importance of data quality and quantity is further supported by Fig. 3, which illustrates the frequency at which our approach identified each term in the design matrix across different values of n and SNR. The boxes in the figure delineate regions where each algorithm achieved model discovery above 80% for the Lorenz system, providing insights into the selected terms contributing to the success and failure of each method across different settings. When faced with limited observations and low signal quality, our approach identified overly sparse models that failed to represent the governing dynamics accurately, while SINDy with AIC selected erroneous terms, struggling to obtain a parsimonious representation of the underlying equations. Figure 3 also illustrates the decline in our method's performance for deterministic systems, as it identified several ancillary terms for the Lorenz dynamics when $\text{SNR} = \infty$. The decrease in identification accuracy stemmed from the identified model's violation of the homoscedasticity assumption in linear regression, which occurs when residuals exhibit non-constant variance. Figure 4 demonstrates that our method did not satisfy this assumption when identifying the \dot{x}_1 equation of the Lorenz system. Consequently, our approach selected additional terms to balance the variance among the model's residuals while sacrificing correct system discovery. As the noise in the system slightly increased, however, homoscedasticity in the residuals became more pronounced, enabling our approach to distinguish the equation's true underlying structure. In contrast, SINDy with AIC avoids traditional statistical concerns by comparing the predicted \hat{X} with the true X . However, our method embraces statistical inference, even with its inherent challenges. This adherence proves advantageous in real-world applications, particularly where data contains low levels of noise in the signal, enhancing our method's reliability in identifying accurate governing equations. While this approach may encounter issues in noiseless environments due to the assumption of homoscedasticity, it maintains a rigorous framework for model discovery. Our method's commitment to statistical inference, despite potential drawbacks, underscores its effectiveness in extracting meaningful insights from observational data, even when true system dynamics are elusive.

Our method outperformed SINDy with AIC in identifying a range of ODEs, especially three-dimensional systems. One potential explanation for the lesser performance of SINDy with AIC is that multicollinearity in the design matrix often causes OLS to produce unstable coefficients. Due to the sensitivity of the estimated coefficients, small changes in the data can lead to fluctuations in their magnitude, making it difficult for the sparsity-promoting parameter to determine the correct model. As a result, the initial phase of the hard-thresholding procedure of SINDy with AIC inadvertently removed the true dynamic terms of the underlying system. Therefore, this model selection approach will likely face persistent challenges when discovering higher-dimensional systems that contain additional multicollinearity in the design matrix.

Figure 5 shows the computational time, measured in seconds, required for our approach and SINDy with AIC to perform model discovery. While our method demanded greater computational effort for the two-dimensional linear system than SINDy with AIC, it demonstrated better efficiency in identifying the Lorenz dynamics as n increased. The decrease in efficiency of SINDy with AIC can be attributed to its model selection process, which involves enumerating all potential prediction models—a procedure that becomes progressively more expensive with data in higher dimensions³⁵. In contrast, our approach displayed a similar rate of increase in computational complexity as the time series expanded for both systems, suggesting that our method was less affected by the growing data dimensionality than SINDy with AIC. Thus, our method offers a more efficient alternative for identifying three-dimensional systems with increasing time series lengths.

Comparison with Ensemble-SINDy. Expanding on our analysis in the subsection Assessing ARGOS systematically, we directed our focus to a more recent alternative within the SINDy framework to emphasize the advantages of our method. In particular, we examined Ensemble-SINDy (ESINDy), a variant that employs bagging (bootstrap aggregation) and bragging (robust bagging) to obtain a distribution of estimates before thresholding based on specific inclusion probabilities³¹. For a direct comparison with our method, we integrated ESINDy with Algorithm 1 (see Supplementary Note 1) to smooth the data and approximate the derivative automatically. During the model discovery phase, we used ESINDy's default values for the thresholding hyperparameter $\lambda_{\text{SINDy}} = 0.2$ and inclusion probability tolerances $\text{tol}_{\text{stan}} = 0.6$ and $\text{tol}_{\text{lib}} = 0.4$ for its standard and library versions³¹.

In Fig. 6a, ESINDy with bagging improved identification performance as the length of the time series n increased. However, the performance of other ESINDy variants decreased with increasing n , signaling a complex relationship between hyperparameter fine-tuning and the length of the time series. Moreover, the inconsistent performance across different time series lengths suggests that achieving optimal results with ESINDy would necessitate frequent hyperparameter adjustments, a task that is often impractical and resource-intensive in real-world scenarios, where data characteristics can vary widely, and re-tuning may not always be feasible. Figure 6b further indicates that ESINDy with bagging and library ESINDy with bragging are advantageous, especially with high levels of SNR. Nevertheless, achieving this level of performance requires careful tuning of multiple hyperparameters, highlighted by fluctuating performance as the observations increase.

By contrast, our approach performs its hyperparameter tuning process automatically, enhancing adaptability for various systems. We employ cross-validation to determine the optimal

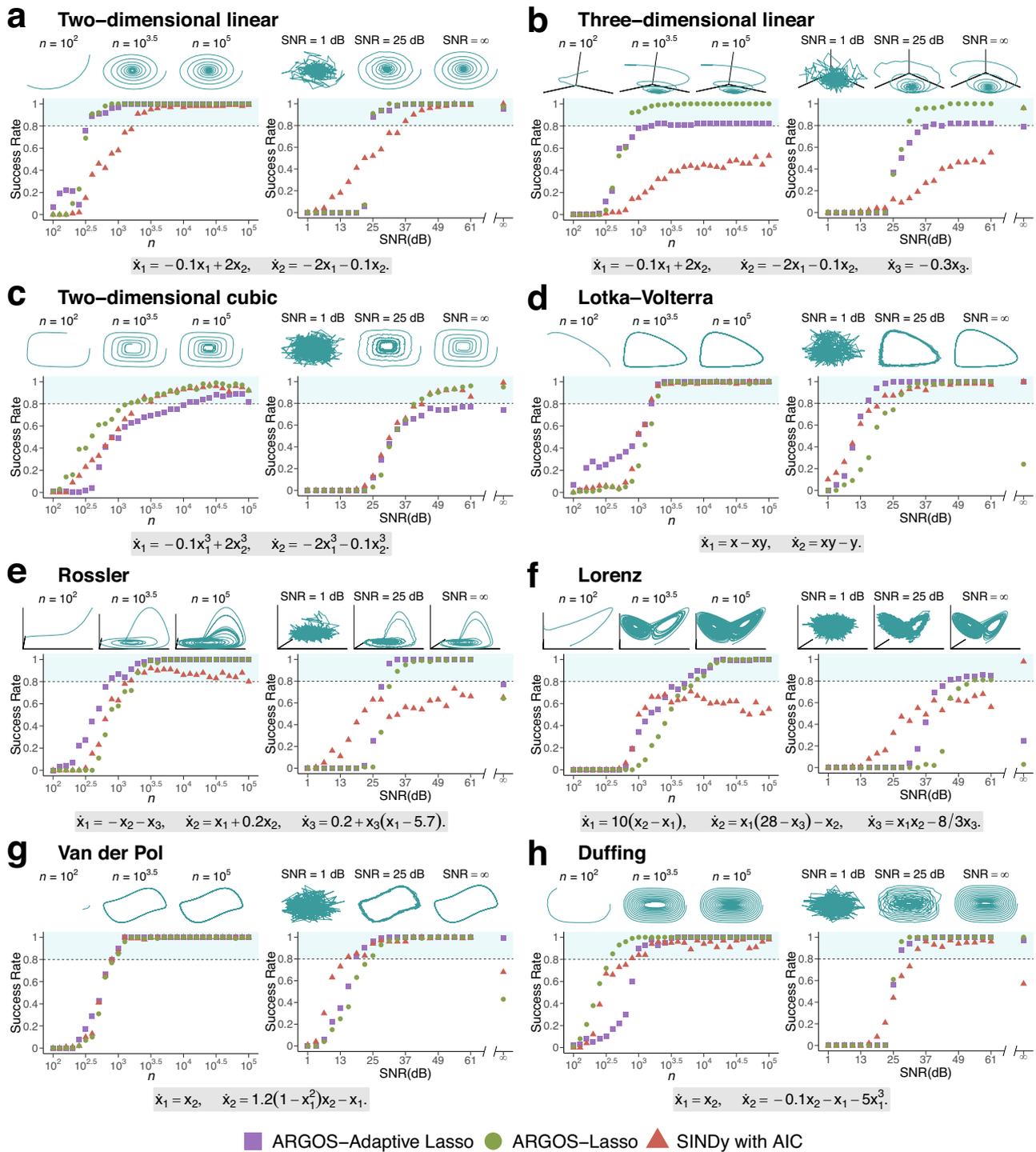


Fig. 2 Success rate of ARGOS versus SINDy with AIC for linear and nonlinear systems. We generate 100 random initial conditions and examine the success rate of ARGOS and SINDy with AIC in discovering the correct terms of the governing equations from each system at each value of n and signal-to-noise ratio (SNR). **a, b** Linear systems. First-order nonlinear systems in two (**c, d**) and three (**e, f**) dimensions. **g, h** Second-order nonlinear systems. We increase the time-series length n while holding SNR = 49dB (left panels) and fix $n = 5000$ when increasing the SNR (right panels). Shaded regions represent model discovery above 80%.

regularization parameter λ for the lasso and the adaptive lasso. Then, we use a grid of hard thresholding values η and select the one that minimizes the Bayesian Information Criterion (BIC) for the final model. Figures 7 and S6 demonstrate the variability in these hyperparameters, reinforcing the impracticality of manual tuning without specific domain-specific knowledge (see Supplementary Note 3). Finally, we perform bootstrap sampling to develop confidence intervals, allowing for inference on the

identified predictors rather than constraining estimates based on a user-specified threshold.

Discussion

We have demonstrated an automatic method, ARGOS, for extracting dynamical systems from scarce and noisy data while only assuming that the governing terms exist in the design matrix.

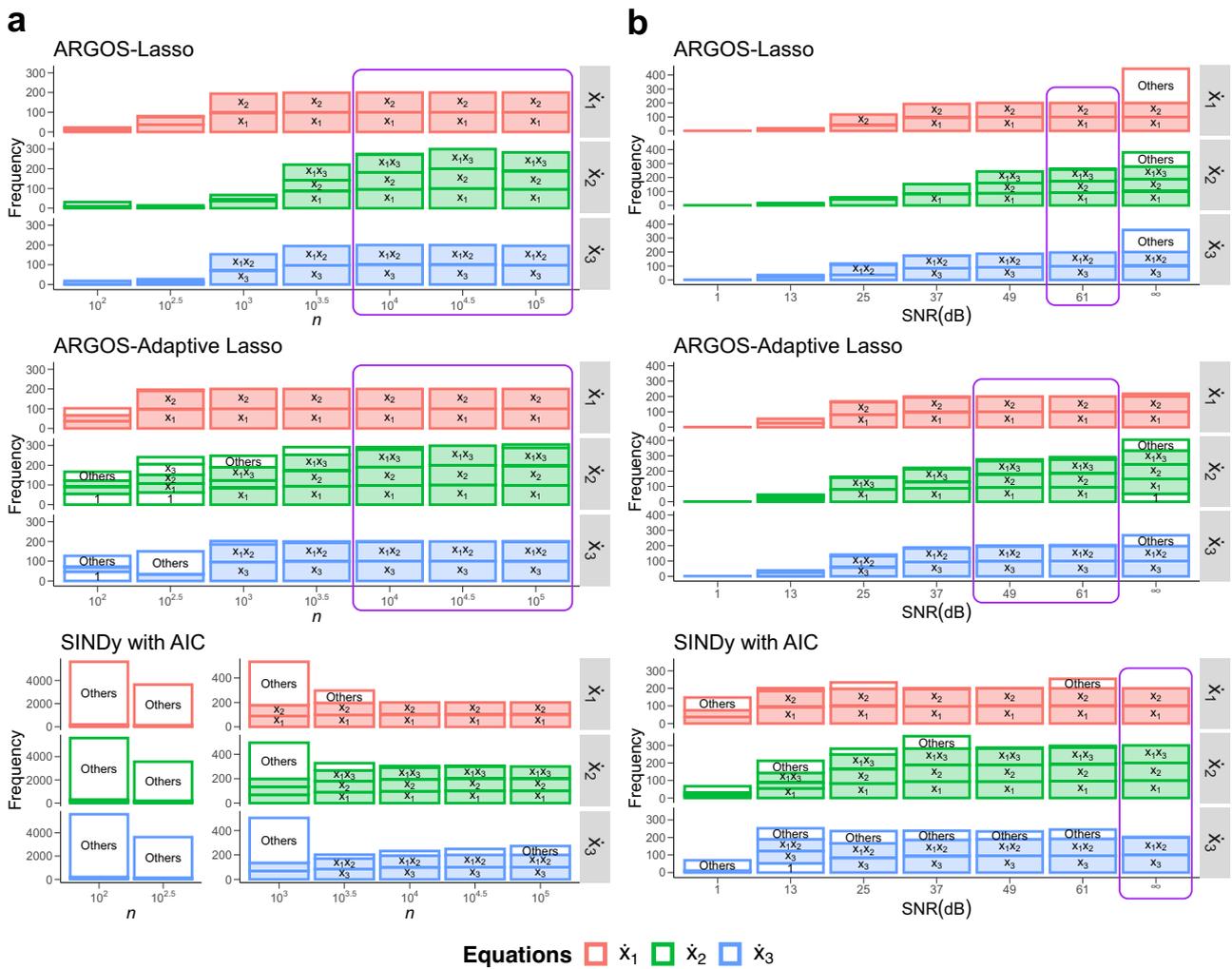


Fig. 3 Frequency of identified variables for the Lorenz system across algorithms. Colors correspond to each governing equation; filled boxes indicate correctly identified variables, while white boxes denote erroneous terms. Panels show the frequency of identified variables for data sets with (a) increasing time-series length n (signal-to-noise ratio (SNR) = 49 dB), and (b) SNR ($n = 5000$). Purple-bordered regions demarcate model discovery above 80%.

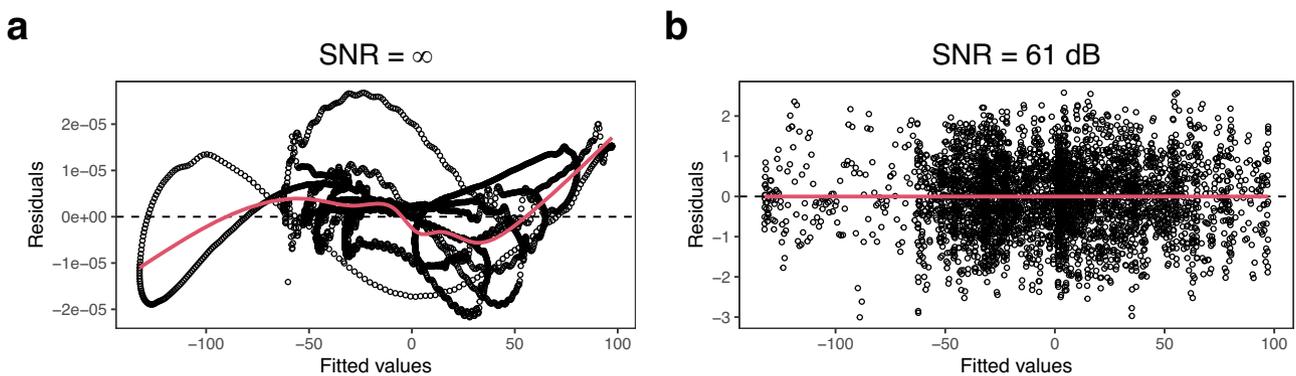


Fig. 4 Residuals vs. Fitted Diagnostics for the ARGOS-Lasso identified model of the Lorenz x_1 equation. Comparison of residuals for the prediction models identified using the lasso with ARGOS for the Lorenz system’s x_1 equation when data are (a) noiseless and (b) contaminated with signal-to-noise ratio (SNR) = 61 dB. Circular points depict the residuals, and the red curve, generated using the Locally Weighted Scatterplot Smoothing (LOWESS) method, illustrates the trend of residuals in relation to the fitted values.

Our approach combines the Savitzky-Golay filter for signal denoising and differentiation with sparse regression and bootstrap sampling for confidence interval estimation, effectively addressing the inverse problem of inferring underlying dynamics from observational data through reliable variable selection. By

examining diverse trajectories, we showcased the capabilities of our algorithm in automating the discovery of mathematical models from data, consistently outperforming the established SINDy with AIC, especially when identifying systems in three dimensions.

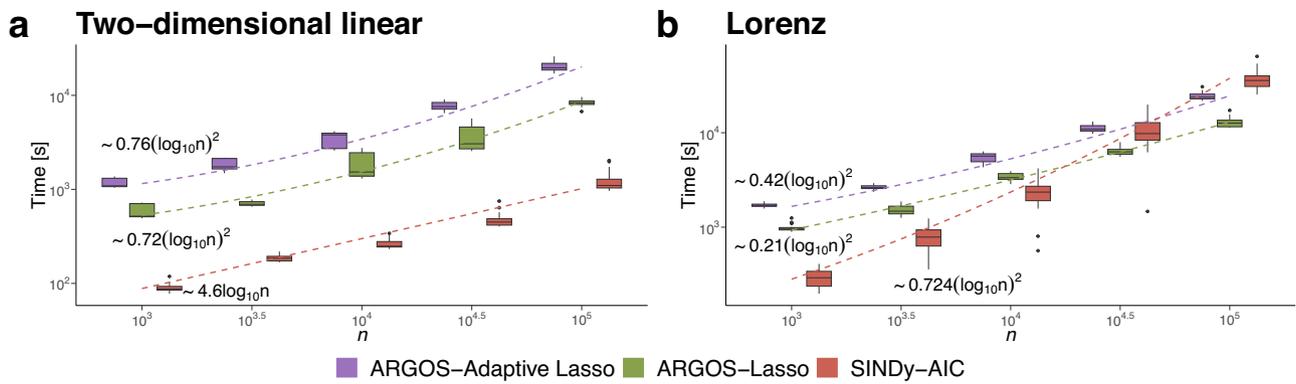


Fig. 5 Time-complexity (seconds) between ARGOS and SINDy with AIC. Boxplots depict the computational time required for model discovery over 30 instances for (a) a two-dimensional damped oscillator with linear dynamics and (b) the Lorenz system. The black bar within each box represents the median computational time. Whiskers extending from each box show 1.5 times the interquartile range. Data points beyond the end of the whiskers are outlying points. Equations accompanying the dashed lines indicate the fitted mean computational time for each algorithm at various values of time-series length n .

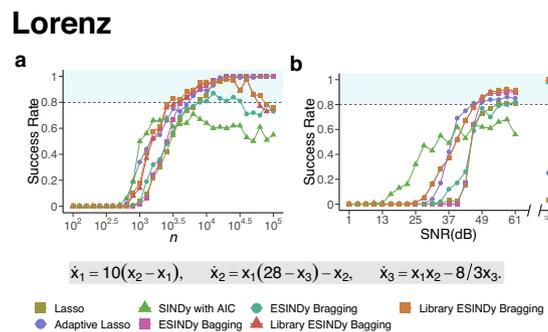


Fig. 6 Success rate of ARGOS versus alternative SINDy methods for the Lorenz system. Using 100 random initial conditions, we examine the success rate of ARGOS, SINDy with AIC, and Ensemble-SINDy in discovering the correct terms of the governing equations at each value of time-series length n and signal-to-noise ratio (SNR). We (a) increase n with a constant SNR = 49dB and (b) keep $n = 5000$ when modulating the SNR. Shaded regions denote model discovery rates exceeding 80%.

In our study, we diverge from the approach originally taken by SINDy with AIC³⁵, particularly in calculating the derivative and the breadth of testing scenarios. Unlike SINDy with AIC, which was originally examined using derivatives calculated directly from the true Lorenz system equations, our methodology employs the Savitzky–Golay filter for automated smoothing and numerical derivative approximation (detailed in Algorithm 1 in Supplementary Note 1). This distinction is non-trivial, as direct derivative calculation, while precise, often lacks feasibility in real-world applications where true system equations are not readily accessible. Conversely, while the Savitzky–Golay method introduces an element of approximation error, it significantly enhances the applicability of our approach in diverse and unpredictable real-world scenarios. Furthermore, the systematic analysis we adopted is more expansive, assessing each method’s effectiveness across a range of initial conditions, not just a specific set. This rigorous testing paradigm not only underscores our method’s robustness but also provides a more holistic view of its performance in varied practical contexts, which is an aspect that was not previously as thoroughly explored with SINDy with AIC.

While we have shown promising results with our approach, it is important to note several potential limitations. First, although our method effectively automates model discovery, it can only correctly represent the true governing equations if the active terms are present in the design matrix, a constraint inherent in regression-based identification procedures. Building on this point, we stress the importance of data quantity and quality as

identification accuracy improved with sufficient observations and moderate to high signal-to-noise ratios. We also found that our method performs better when data contains low levels of noise, as opposed to noiseless systems. The linear regression assumption of homoscedasticity is violated under noiseless conditions, and the method identifies spurious terms to develop a more constant variance among the residuals. However, this issue can be mitigated in the presence of a small amount of noise in the data, leading to a more constant variance in the residuals of the true model and enabling more accurate identification. Lastly, as the number of observations and data dimensionality increase, bootstrap sampling becomes computationally demanding, which can significantly prolong the model selection process and limit our algorithm’s applicability in real-time. Nonetheless, obtaining confidence intervals through bootstrap sampling serves as a reliable approach for our method, allowing us to eliminate superfluous terms and select the ones that best represent the underlying equations, ultimately leading to more accurate predictions of the system’s dynamics.

In this information-rich era, data-driven methods for uncovering governing equations are increasingly crucial in scientific research. By developing automated processes, researchers can develop concise models that accurately represent the underlying dynamics in their data, accelerating advancements across various disciplines in science. Our study endorses an inference-based approach that combines statistical learning and model assessment methods, emphasizing the importance of thorough model

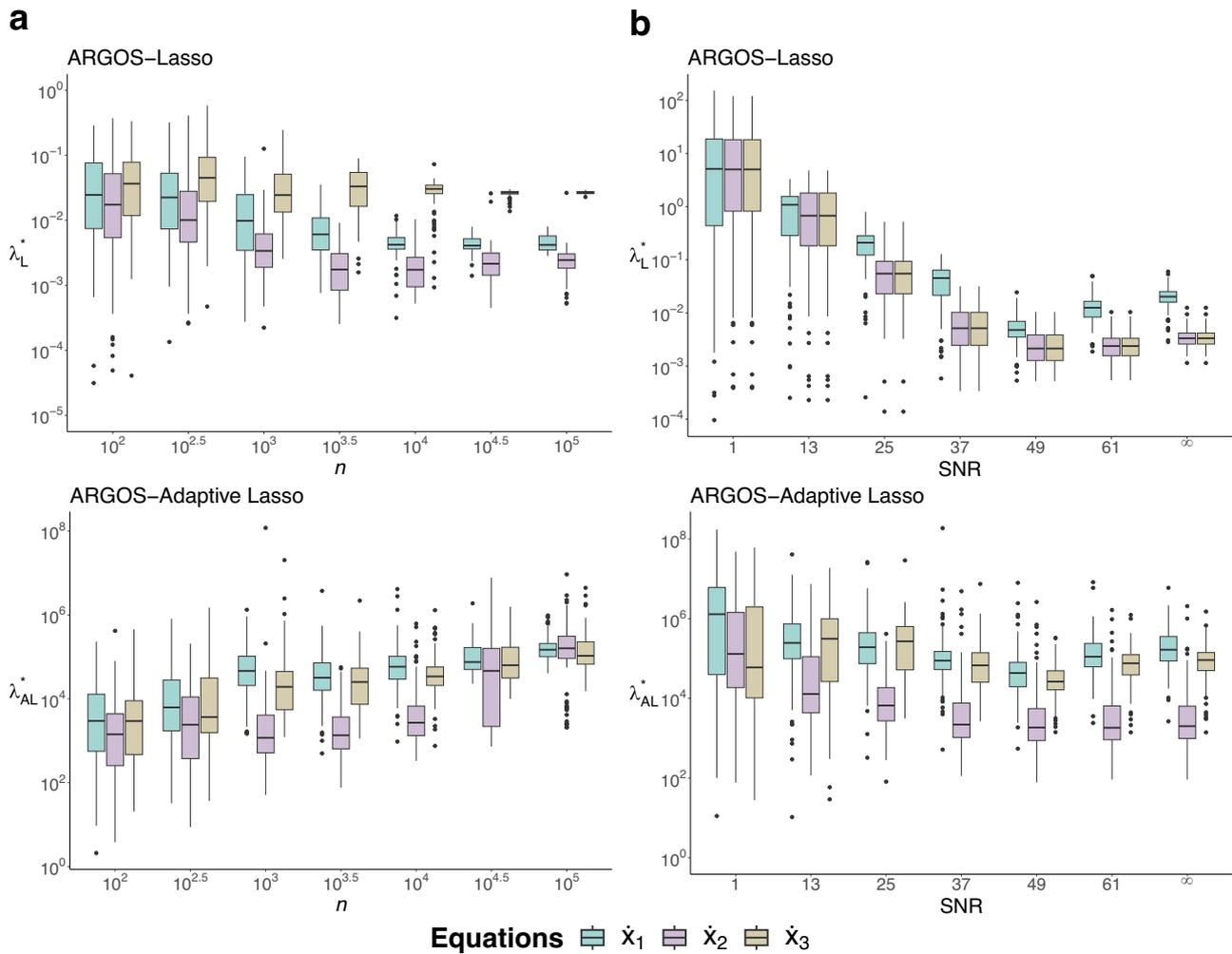


Fig. 7 Comparison of the optimal regularization parameter λ^* under various conditions for the Lorenz system. Boxplots depict λ^* distributions from 100 initial conditions. The black bar within each box represents the median value of λ^* . Whiskers extending from each box show 1.5 times the interquartile range. Data points beyond the end of the whiskers are outlying points. Colors represent each governing equation identified from data sets with increasing (a) time-series length n (signal-to-noise ratio (SNR) = 49 dB), and (b) SNR ($n = 5000$).

evaluation for building confidence in discovering governing equations from data. As data-driven techniques advance, we look forward to further developments in automatic system identification that will continue contributing to the search for the elusive laws governing many intricate systems.

Methods

The lasso and adaptive lasso for variable selection. For the j th column of $\tilde{\mathbf{X}}$ and \mathbf{B} in Eq. (4), we implement sparse regression by adding weighted ℓ_q penalties to the OLS regression estimate⁴⁹

$$\operatorname{argmin}_{\beta} \left\{ \sum_{i=1}^n \left(\hat{x}_i - \beta_0 - \sum_{k=1}^p \theta(\mathbf{X})_{i,k} \beta_k \right)^2 + \lambda \sum_{k=1}^p w_k |\beta_k|^q \right\}. \quad (6)$$

When all weights $w_k = 1$ for $k = 1, \dots, p$, Eq. (6) represents the lasso for $q = 1$ and ridge regression for $q = 2$ ^{42,50}. Furthermore, the adaptive lasso is derived from the lasso by incorporating pilot estimates $\tilde{\beta}$ and setting $w_k = 1/|\tilde{\beta}_k|^\nu$ ⁴³. The weighted penalty in the adaptive lasso can be interpreted as an approximation of the ℓ_p penalties with $p = 1 - \nu$ ⁵¹. Therefore, fixing $\nu = 1$ allows us to achieve a soft-threshold approximation to the ℓ_0 penalty, providing an alternative to the hard-thresholding in the

SINDy algorithm, which requires a choice of the cut-off hyperparameter²⁶.

As λ increases in Eq. (6), ridge regression, the lasso, and the adaptive lasso shrink the coefficients toward zero. However, of these three methods, the lasso and the adaptive lasso perform variable selection by reducing small coefficients to exactly zero⁴². We use `glmnet` to solve Eq. (6) by producing a default λ grid and applying 10-fold cross-validation to determine the optimal initial tuning parameter λ_0^* ⁵². We then refine the grid around λ_0^* with 100 points spanning $[\lambda_0^*/10, 1.1 \cdot \lambda_0^*]$ and impose this updated grid on `glmnet` to solve Eq. (6) again, identifying the optimal λ^* that best predicts $\tilde{\mathbf{x}}_j$.

The lasso is effective when only a few coordinates of the coefficients β are nonzero. However, like OLS, the lasso provides unstable estimates when predictors are collinear, whereas ridge regression produces more stable solutions when multicollinearity exists in the data⁴⁹. Therefore, we apply ridge regression to the data in the first stage of the adaptive lasso to obtain stable pilot estimates $\tilde{\beta}$ and reduce the effects of multicollinearity⁴³.

The second stage of the adaptive lasso then uses the $\tilde{\beta}$ pilot estimates to calculate the weights vector w , enabling variable selection by solving the problem in Eq. (6). Here, we calculate the weights vector w using pilot estimates $\tilde{\beta}$ corresponding to the optimal λ_{ridge}^* ridge regression model before identifying a

separate tuning parameter $\lambda_{\text{adaptive lasso}}^*$. In doing so, we make Eq. (6) less computationally expensive since we optimize twice on a single parameter rather than simultaneously optimizing over λ_{ridge}^* and $\lambda_{\text{adaptive lasso}}^*$ ⁵³.

The adaptive lasso often yields a sparser solution than the lasso since applying individual weights to each variable places a stronger penalty on smaller coefficients, reducing more of them to zero. Here, small $\tilde{\beta}$ coefficients from the first stage of the adaptive lasso lead to a larger penalty in the second. Larger penalty terms in the second stage of the adaptive lasso result in more coefficients being set to zero than the standard lasso method. Furthermore, a smaller penalty term enables the adaptive lasso to uncover the true coefficients and reduce bias in the solution⁵³.

The adaptive lasso, valuable for system identification, obtains the oracle property when the $\tilde{\beta}$ pilot estimates converge in probability to the true value of β at a rate of $1/\sqrt{n}$ (\sqrt{n} -consistency). As n increases, the algorithm will select the true nonzero variables and estimate their coefficients as if using maximum likelihood estimation⁴³.

Algorithm implementation. When applying ARGOS for model selection, we use threshold values strategically to establish a range conducive to successful model discovery. Specifically, we employ a logarithmically spaced grid defined as $\eta = 10^{-8}, 10^{-7}, \dots, 10^1$ to threshold the sparse regression coefficients. Subsequently, we perform OLS on each subset $\mathcal{K}_i = \{k : |\hat{\beta}_k| \geq \eta_i\}$, $i = 1, \dots, \text{card}(\eta)$ of selected variables, determining an unbiased estimate for β ⁴⁹. We then calculate the BIC for each η regression model and select the model with the minimum value, further promoting sparsity in the identification process^{44,54}.

The number of bootstrap sample estimates B must be large enough to develop confidence intervals for variable selection⁴⁵. Therefore, we collect $B = 2000$ bootstrap sample estimates and sort them by $\hat{\beta}_k^{\text{OLS}\{1\}} \leq \hat{\beta}_k^{\text{OLS}\{2\}} \leq \dots \leq \hat{\beta}_k^{\text{OLS}\{B\}}$. We then use the $100(1 - \alpha)\%$ confidence level, where $\alpha = 0.05$, to calculate the integer part of $B\alpha/2$ and develop estimates of the lower and upper bounds: $CI_{\text{lo}} = \lfloor B\alpha/2 \rfloor$ and $CI_{\text{up}} = B - CI_{\text{lo}} + 1$. Finally, we implement these calculations to develop confidence intervals $[\hat{\beta}_k^{\text{OLS}\{CI_{\text{lo}}\}}, \hat{\beta}_k^{\text{OLS}\{CI_{\text{up}}\}}]$ from our sample distribution⁵⁵.

To automatically identify the system using SINDy with AIC, we deploy Algorithm 1 (see Supplementary Note 1), which facilitates signal smoothing and derivative approximation with the Savitzky-Golay filter. This consistent use of Algorithm 1 ensures a standardized comparison between SINDy with AIC with our proposed algorithm.

Both our method and SINDy with AIC use information criteria in their methodologies, but their applications diverge. Specifically, SINDy with AIC employs AIC to ascertain the optimal model between the estimated system state-space, $\tilde{\mathbf{X}}$, and the true system state-space, \mathbf{X} , representing the ground truth³⁵. In contrast, our method leverages BIC to determine the prediction model corresponding to the optimal hard-thresholding parameter η^* , thereby promoting sparsity in the model selection process⁵⁴.

Building the data sets and tests. We conducted two sets of numerical experiments to assess the impact of data quality and quantity on the performance of the algorithms. Central to our approach is the use of a distribution of random initial conditions. By leveraging this strategy, we evaluated our method's efficiency with random data, reflecting its potential in real-world settings marked by inherent unpredictability and variability (see Supplementary Note 4).

To evaluate the algorithms' performance with limited data, we first kept the signal-to-noise ratio constant (SNR = 49) and

increased the number of observations n for each ODE system. We generated 100 random initial conditions and used temporal grids starting with $t_{\text{initial}} = 0$ and a varying t_{final} between 1 ($n = 10^2$) and 1000 ($n = 10^5$) with a time step $\Delta t = 0.01$. For the Lorenz equations, we used $\Delta t = 0.001$, resulting in t_{final} values ranging from 0.1 ($n = 10^2$) and 100 ($n = 10^5$)²⁶. Furthermore, we implement the systems' corresponding Δt as dt in Algorithm 1 for smoothing and differentiation (see Supplementary Note 1).

To examine the algorithms' performance under noisy conditions, we varied the SNR in the data by corrupting the state matrix with a zero-mean Gaussian noise matrix $\mathbf{Z} \sim \mathcal{N}(0, \sigma_z^2)$. In this setting, we determined the standard deviation σ_{z_j} of each column of \mathbf{Z} as

$$\sigma_{z_j} = \sigma_{x_j} \cdot 10^{-\frac{\text{SNR}}{20}}, \quad j = 1, \dots, m, \quad (7)$$

and develop the noise corrupted $\tilde{\mathbf{X}}$ as⁵⁶

$$\tilde{\mathbf{X}} = \mathbf{X} + \mathbf{Z}. \quad (8)$$

Keeping n constant, we again used 100 random initial conditions and generated $\tilde{\mathbf{X}}$ matrices increasing in noise levels such that SNR = 1, 4, ..., 61 dB with $\Delta\text{SNR} = 3$ dB, including a noiseless system (SNR = ∞).

For both the tests with varying n and SNR, we constructed the design matrix $\Theta^{(0)}(\mathbf{X})$ with monomial functions up to $d = 5$ of the smoothed columns of \mathbf{X} ²⁶. We then performed system identification with each data set and calculated the success rate of each algorithm as the probability of extracting the correct terms of the governing equations. Additionally, we analyzed the most frequently selected variables of each method.

Finally, we measured the computational time, in seconds, for running our method and SINDy with AIC by performing model discovery on 30 instances of the two-dimensional linear system and the Lorenz system for time series lengths $n = 10^2, 10^{2.5}, \dots, 10^5$, using one CPU core with a single thread.

Data availability

All data generated for this study can be generated with the code available at <http://github.com/kevinegan31/ARGOS>. We provide a snapshot of the data on GitHub.

Code availability

All code used in this study is available at <http://github.com/kevinegan31/ARGOS>.

Received: 9 May 2023; Accepted: 25 December 2023;

Published online: 09 January 2024

References

- Ruelle, D. & Takens, F. On the nature of turbulence. *Commun. Math. Phys.* **20**, 167–192 (1971).
- Watts, D. J. & Strogatz, S. H. Collective dynamics of 'small-world' networks. *Nature* **393**, 440–442 (1998).
- Petrov, V., Gáspár, V., Masere, J. & Showalter, K. Controlling chaos in the Belousov–Zhabotinsky reaction. *Nature* **361**, 240–243 (1993).
- Mackey, M. C. & Glass, L. Oscillation and chaos in physiological control systems. *Science* **197**, 287–289 (1977).
- Tyson, J. J., Chen, K. & Novak, B. Network dynamics and cell physiology. *Nat. Rev. Mol. Cell Biol.* **2**, 908–916 (2001).
- Steuer, R., Gross, T., Selbig, J. & Blasius, B. Structural kinetic modeling of metabolic networks. *Proc. Natl Acad. Sci.* **103**, 11868–11873 (2006).
- Karsenti, E. Self-organization in cell biology: A brief history. *Nat. Rev. Mol. Cell Biol.* **9**, 255–262 (2008).
- Kholodenko, B. N., Hancock, J. F. & Kolch, W. Signalling ballet in space and time. *Nat. Rev. Mol. Cell Biol.* **11**, 414–426 (2010).
- Altrock, P. M., Liu, L. L. & Michor, F. The mathematics of cancer: Integrating quantitative models. *Nat. Rev. Cancer* **15**, 730–745 (2015).
- Chialvo, D. R. Emergent complex neural dynamics. *Nat. Phys.* **6**, 744–750 (2010).

11. Todorov, E. Optimality principles in sensorimotor control. *Nat. Neurosci.* **7**, 907–915 (2004).
12. Breakspear, M. Dynamic models of large-scale brain activity. *Nat. Neurosci.* **20**, 340–352 (2017).
13. Sugihara, G. & May, R. M. Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. *Nature* **344**, 734–741 (1990).
14. Earn, D. J. D., Rohani, P., Bolker, B. M. & Grenfell, B. T. A simple model for complex dynamical transitions in epidemics. *Science* **287**, 667–670 (2000).
15. Sugihara, G. et al. Detecting causality in complex ecosystems. *Science* **338**, 496–500 (2012).
16. Wood, S. N. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature* **466**, 1102–1104 (2010).
17. Nicolis, C. & Nicolis, G. Reconstruction of the dynamics of the climatic system from time-series data. *Proc. Natl Acad. Sci.* **83**, 536–540 (1986).
18. Steffen, W. et al. Trajectories of the Earth system in the Anthropocene. *Proc. Natl Acad. Sci.* **115**, 8252–8259 (2018).
19. Waltz, D. & Buchanan, B. G. Automating Science. *Science* **324**, 43–44 (2009).
20. Schmidt, M. D. et al. Automated refinement and inference of analytical models for metabolic networks. *Phys. Biol.* **8**, 055011 (2011).
21. Crutchfield, J. P. & McNamara, B. Equations of motion from a data series. *Complex Syst.* **1**, 417–452 (1987).
22. Tarantola, A. Inverse Problem Theory and Methods for Model Parameter Estimation (Society for Industrial and Applied Mathematics, Philadelphia 2005).
23. Hong, X. et al. Model selection approaches for non-linear system identification: A review. *Int. J. Syst. Sci.* **39**, 925–946 (2008).
24. Schmidt, M. & Lipson, H. Distilling free-form natural laws from experimental data. *Science* **324**, 81–85 (2009).
25. Udrescu, S.-M. & Tegmark, M. AI Feynman: A physics-inspired method for symbolic regression. *Sci. Adv.* **6**, 2631 (2020).
26. Brunton, S. L., Proctor, J. L. & Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl Acad. Sci.* **113**, 3932–3937 (2016).
27. Zhang, S. & Lin, G. Robust data-driven discovery of governing physical laws with error bars. *Proc. Royal Soc. A: Math., Phys. Eng. Sci.* **474**, 20180305 (2018).
28. Cortiella, A., Park, K.-C. & Doostan, A. Sparse identification of nonlinear dynamical systems via reweighted ℓ_1 -regularized least squares. *Comp. Methods Appl. Mech. Eng.* **376**, 113620 (2021).
29. Schaeffer, H., Tran, G. & Ward, R. Extracting sparse high-dimensional dynamics from limited data. *SIAM J. Appl. Math.* **78**, 3279–3295 (2018).
30. Hirsh, S. M., Barajas-Solano, D. A. & Kutz, J. N. Sparsifying priors for Bayesian uncertainty quantification in model discovery. *Royal Soc. Open Sci.* **9**, 211823 (2022).
31. Fasel, U., Kutz, J. N., Brunton, B. W. & Brunton, S. L. Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proc. Royal Soc. A: Math., Phys. Eng. Sci.* **478**, 20210904 (2022).
32. Kaheman, K., Brunton, S. L. & Kutz, J. N. Automatic differentiation to simultaneously identify nonlinear dynamics and extract noise probability distributions from data. *Mach. Learning: Sci. Technol.* **3**, 015031 (2022).
33. Lusch, B., Kutz, J. N. & Brunton, S. L. Deep learning for universal linear embeddings of nonlinear dynamics. *Nat. Commun.* **9**, 4950 (2018).
34. Delahunt, C. B. & Kutz, J. N. A toolkit for data-driven discovery of governing equations in high-noise regimes. *IEEE Access* **10**, 31210–31234 (2022).
35. Mangan, N. M., Kutz, J. N., Brunton, S. L. & Proctor, J. L. Model selection for dynamical systems via sparse regression and information criteria. *Proc. Royal Soc. A: Math., Phys. Eng. Sci.* **473**, 20170009 (2017).
36. de Silva, B. M., Higdun, D. M., Brunton, S. L. & Kutz, J. N. Discovery of physics from data: Universal laws and discrepancies. *Front. Artificial Intel.* **3**, 25 (2020).
37. Cortiella, A., Park, K.-C. & Doostan, A. A priori denoising strategies for sparse identification of nonlinear dynamical systems: a comparative study. *J. Comput. Inf. Sci. Eng.* **23**, 011004 (2023).
38. Lejarza, F. & Baldea, M. Discovering governing equations via moving horizon learning: The case of reacting systems. *AIChE J.* **68**, e17567 (2022).
39. Guckenheimer, J. & Holmes, P. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields* **42**. (Springer Science & Business Media, New York 2013).
40. Savitzky, A. & Golay, M. J. E. Smoothing and differentiation of data by simplified least squares procedures. *Anal. Chem.* **36**, 1627–1639 (1964).
41. Press, W.H., Teukolsky, S.A., Vetterling, W.T. & Flannery, B.P. *Numerical Recipes: The Art of Scientific Computing*, 3rd edn. (Cambridge University Press, New York 2007).
42. Tibshirani, R. Regression shrinkage and selection via the Lasso. *J. Royal Stat. Soc. Series B (Methodological)* **58**, 267–288 (1996).
43. Zou, H. The Adaptive Lasso and its oracle properties. *J. Am. Stat. Assoc.* **101**, 1418–1429 (2006).
44. Schwarz, G. Estimating the dimension of a model. *Annals Stat.* **6**, 461–464 (1978).
45. Efron, B. & Tibshirani, R. An Introduction to the Bootstrap. *Monogr. Stat. Appl. Probability* **57**. (Chapman & Hall, New York 1993).
46. Lotka, A. J. Contribution to the theory of periodic reactions. *J. Phys. Chem.* **14**, 271–274 (1910).
47. Naozuka, G. T., Rocha, H. L., Silva, R. S. & Almeida, R. C. SINDy-SA framework: Enhancing nonlinear system identification with sensitivity analysis. *Nonlinear Dyn.* **110**, 2589–2609 (2022).
48. Tran, G. & Ward, R. Exact recovery of chaotic systems from highly corrupted data. *Multiscale Model. Simulation* **15**, 1108–1129 (2017).
49. Tibshirani, R., Friedman, J. H. & Hastie, T. The Elements of Statistical Learning (Springer, New York, 2009).
50. Hoerl, A. E. & Kennard, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **12**, 55–67 (1970).
51. Hastie, T., Tibshirani, R. & Wainwright, M. *Statistical Learning with Sparsity: The Lasso and Generalizations*. (CRC Press, New York, 2015).
52. Friedman, J., Hastie, T. & Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. *J. Stat. Soft.* **33**, 1–22 (2010).
53. Bühlmann, P. & van de Geer, S. *Statistics for High-Dimensional Data*, (Springer Series in Statistics) (Springer Berlin, Heidelberg, 2011).
54. Zou, H., Hastie, T. & Tibshirani, R. On the “degrees of freedom” of the lasso. *Annals Stat.* **35**, 2173–2192 (2007).
55. Zoubir, A. M. & Iskander, D. R. *Bootstrap Techniques for Signal Processing*. (Cambridge University Press, Cambridge, 2004).
56. Lyons, R.G. *Understanding Digital Signal Processing*, 3rd edn. (Pearson, Boston, MA 2011).

Acknowledgements

R.C. has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 872172 (TESTBED2 project: <https://www.testbed2.org>). K.E. & W.L. have received funding from the Department of Engineering at Durham University under the Durham Doctoral Studentship programme.

Author contributions

K.E. carried out the research, wrote the code for the tests, designed the schematic, and performed the computational experiments. K.E. and W.L. wrote the code for ARGOS, SINDy with AIC, and plots, analyzed the results, and generated the figures. W.L. drew the schematic. All authors discussed the results and wrote the paper. R.C. received funding to support this work and designed the research.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s42005-023-01516-2>.

Correspondence and requests for materials should be addressed to Rui Carvalho.

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

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



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024