SINDy – Sparse Identification of Nonlinear Dynamics

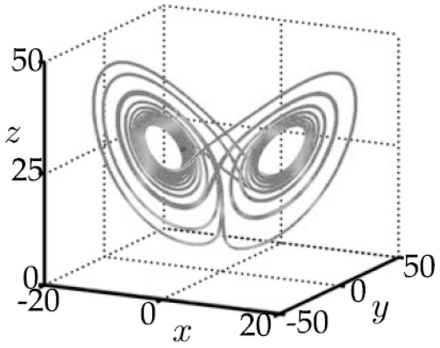
Imperial I-X Symbolic Model Discovery Workshop 2025

Urban Fasel

Imperial College London

SINDy

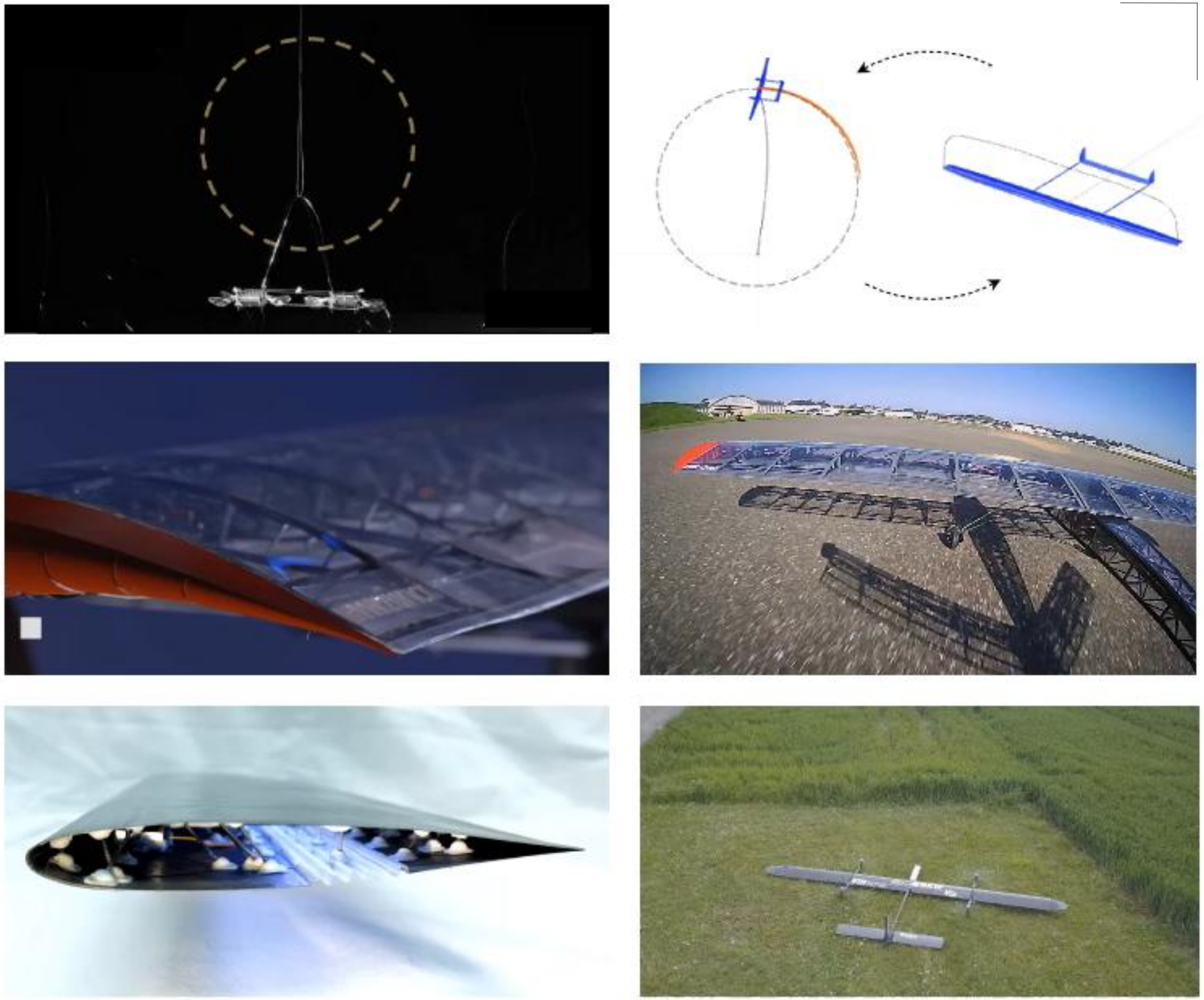
$$\underbrace{\textit{Data} \longrightarrow \textit{Dynamics (assumptions)} \longrightarrow \textit{Model structure \& coefficients}}$$



$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x})$$

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = x(\rho - z) - y$$
$$\dot{z} = xy - \beta z$$

My work



(Adaptive) flight systems

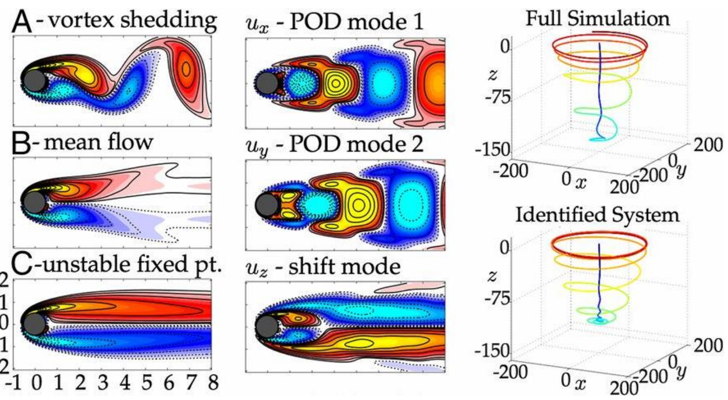
1. Flapping wing MAV
2. Renewable energy systems
3. Composite additive manufacturing morphing wing drones
4. Morphing wings
5. Airborne wind energy

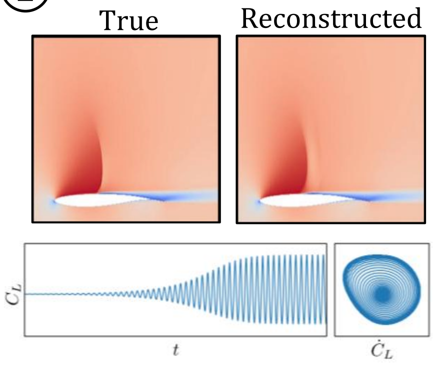
Methods

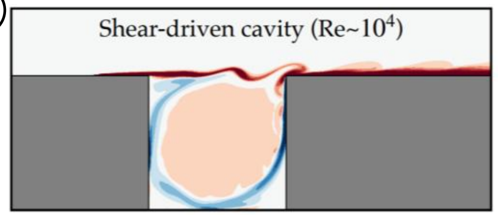
1. Co-design optimization
2. **Data driven modeling & control**
   - DMDc, SINDyC, SINDy-RL
   - E-SINDy, B-SINDy, SINDy-CP
   - Poincaré SINDy, Slow Manifolds

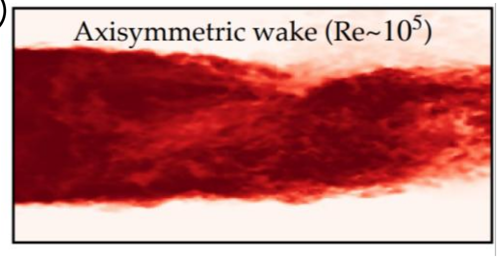
SINDy – applications

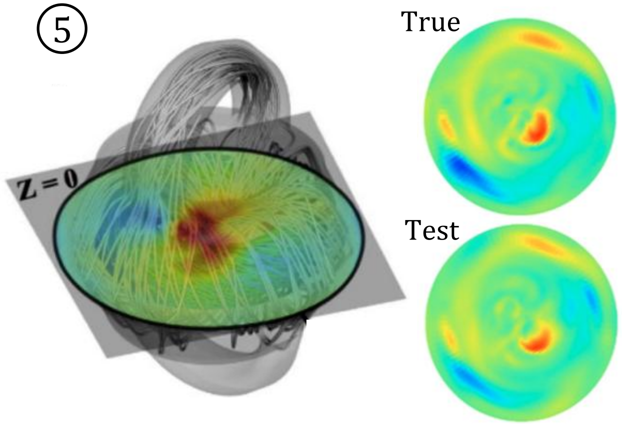
1. [Vortex shedding past a cylinder]()
   - *Time history of POD coefficients:*
     - $\dot{x} = \mu x - \omega y + Axz$
     - $\dot{y} = \omega x + \mu y + Ayz$
     - $\dot{z} = -\lambda(z - x^2 - y^2)$
2. [Shock wave dynamics 2D airfoil transonic buffet conditions]()
   - *Parametric c_L model for different α*
     - $c_L(r, \phi) = c_0 + c_1 r + c_2 r\cos(\varphi) + c_3 r\sin(\varphi) + c_4 r^2\cos(2\varphi) + c_5 r^2\sin(2\varphi)$
3. [Cavity flow]()
   - *Coefficients of 2 active DMD modes*
     - $\dot{\alpha}_1 = \lambda_1\alpha_1 - \mu_1\alpha_1|\alpha_1|^2$
     - $\dot{\alpha}_5 = \lambda_5\alpha_5 - \mu_5\alpha_5|\alpha_5|^2$
4. [Experimental measurements turbulent bluff body wake]()
   - *Statistical behavior of the CoP (learning drift and diffusion of SDE)*
     - $\dot{r} = \lambda r - \mu r^3 + \frac{\sigma^2}{2r} + (\sigma_0 + \sigma_1^2)w(t)$
5. [Plasma dynamics (magnetohydrodynamics): 3D spheromak sim]()
   - *Dominant POD coefficient dynamics*
     - $\dot{a}_1 = 0.091\mathrm{a}_2 + 0.009\mathrm{a}_5$
     - $\dot{a}_2 = -0.091\mathrm{a}_1 + 0.008\mathrm{a}_5 - 0.011\mathrm{a}_6$
     - …
6. [Experimental weakly turbulent fluid flow in a thin electrolyte layer]()
   - *Measured velocity field, identify PDE: form similar to N-S*
     - $\partial_t \mathbf{u} = c_1(\mathbf{u}\cdot\nabla)\mathbf{u} + c_2\nabla^2\mathbf{u} + c_3\mathbf{u} - \rho^{-1}\nabla p + \rho^{-1}\mathbf{f}$
7. [Turbulent 3D channel flow ($Re = 1000$) Johns Hopkins database]()
   - *Identify PDEs: N-S, continuity equation, boundary conditions*
     - $\partial_t \mathbf{u} = -(\mathbf{u}\cdot\nabla)\mathbf{u} - 0.995\nabla p + 4.93\cdot 10^{-5}\nabla^2\mathbf{u}$

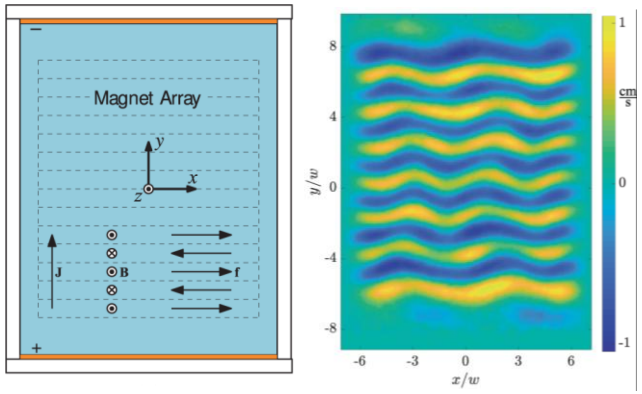
1


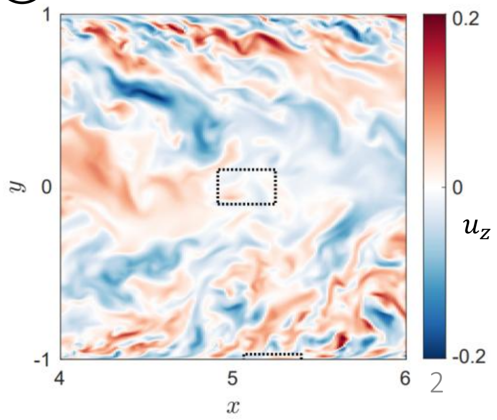
2


3


4


5


6


7


Workshop outline

Part 1: SINDy – Sparse Identification of Nonlinear Dynamics

- **Intro**: identifying ODEs
- Python **example**: *Lorenz system*
- **PySINDy**
- SINDy **challenges and limitations**

"Vanilla" SINDy

SINDy extensions

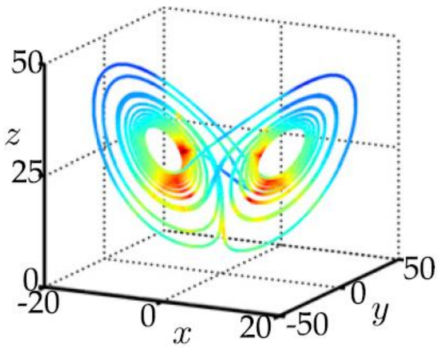
Part 2: Library (ODEs, PDEs, rational functions)

Part 3: Model selection

Part 4: Noise robustness: weak form & ensemble SINDy

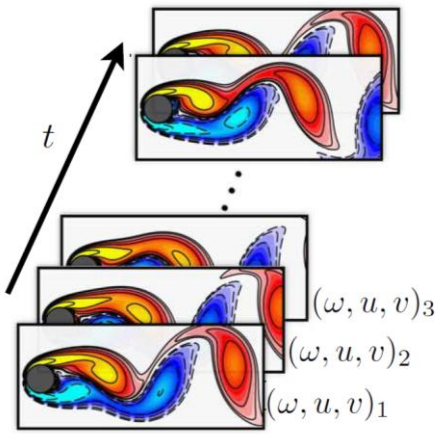
Identifying ODEs and PDEs from data – SINDy

Data → **Dynamics (assumptions)** → **Model structure & coefficients**



$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x})$$

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = x(\rho - z) - y$$
$$\dot{z} = xy - \beta z$$



$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{N}(\mathbf{u})$$

$$\omega_t + (\mathbf{u} \cdot \nabla)\omega = \frac{1}{Re}\nabla^2\omega$$

SL Brunton, JL Proctor, JN Kutz (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems.

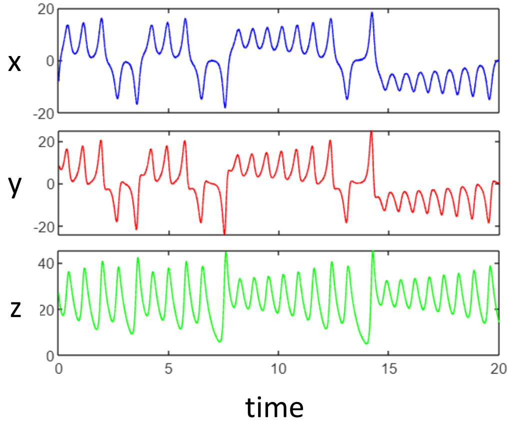
SINDy

1) True Lorenz System

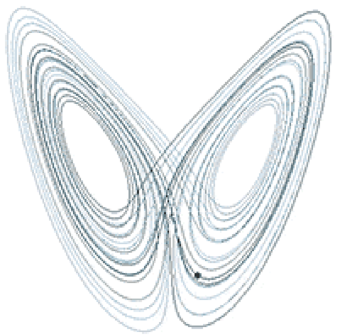
$$\begin{aligned}\dot{x} &= \sigma(y - x)\\ \dot{y} &= x(\rho - z) - y\\ \dot{z} &= xy - \beta z.\end{aligned}$$

Collect time series data *(ρ = 28, σ = 10, β = 8/3)*

Time series



Phase space (attractor)

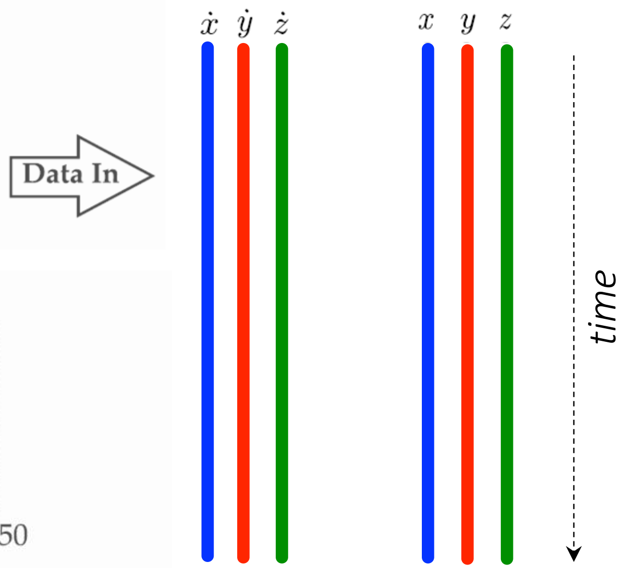


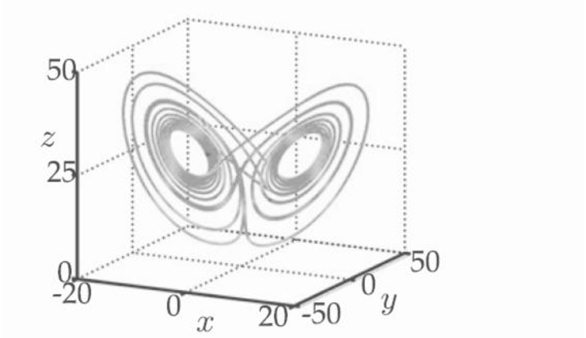
SL Brunton, JL Proctor, JN Kutz (2016) [Discovering governing equations from data by sparse identification of nonlinear dynamical systems].

SINDy

1) Lorenz System Data

$$\begin{aligned}\dot{x} &= \sigma(y-x)\\ \dot{y} &= x(\rho - z) - y\\ \dot{z} &= xy - \beta z.\end{aligned}$$





Approximating time derivatives

Finite difference (central, forward or backward) → *smooth data*

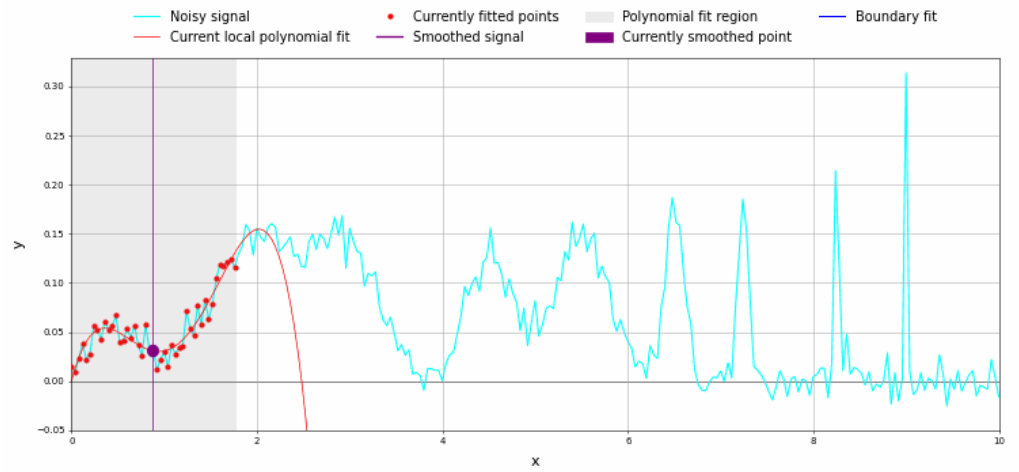
- Different orders of accuracy
- e.g. 1st derivative using central difference
  - 2nd order accuracy: $\dot{x}_t \approx \frac{1}{\Delta t}\left(\frac{1}{2}x_{t+1} - \frac{1}{2}x_{t-1}\right)$
  - 4th order accuracy: $\dot{x}_t \approx \frac{1}{\Delta t}\left(-\frac{1}{12}x_{t+2} + \frac{2}{3}x_{t+1} - \frac{2}{3}x_{t-1} + \frac{1}{12}x_{t-2}\right)$

Smoothed Finite Difference → *noisy data*

- Finite Difference on smoothed/filtered data
  - e.g. Savitzky Golay: least-squares fit of a polynomial to the data
  - applied to the data before differentiation

Polynomial (Savitzky Golay) derivative → *noisy data*

- Analytically compute the derivative of the fitted polynomial



SL Brunton, JL Proctor, JN Kutz (2016) Discovering governing equations from data by sparse identification

Sequential thresholded least squares algorithm

Sparse regression: penalised least squares

$$\rightarrow \ \hat{\xi}_k = \mathrm{argmin}_{\xi_k} \left\| \dot{\mathbf{X}}_k - \mathbf{\Theta}(\mathbf{X})\xi_k \right\|_2^2 + \lambda \|\xi_k\|_0$$

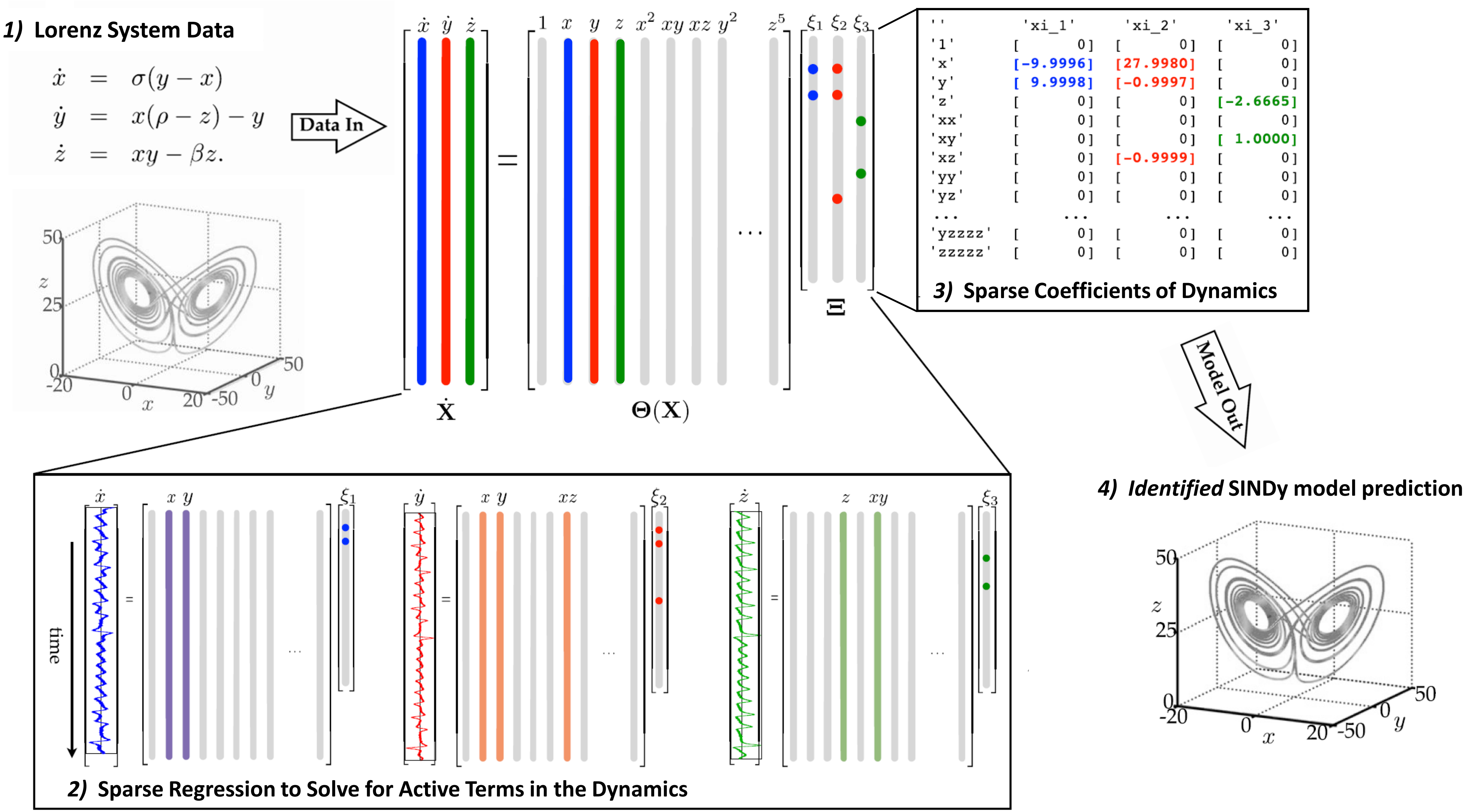
```
function Xi = sparsifyDynamics(Theta,dXdt,lambda,n)
% Compute Sparse regression: sequential least squares
Xi = Theta\dXdt;  % Initial guess: Least-squares

% Lambda is our sparsification knob.
for k=1:10
    smallinds = (abs(Xi)<lambda);  % Find small coefficients
    Xi(smallinds)=0;               % and threshold
    for ind = 1:n                  % n is state dimension
        biginds = ~smallinds(:,ind);
% Regress dynamics onto remaining terms to find sparse Xi
        Xi(biginds,ind) = Theta(:,biginds)\dXdt(:,ind);
    end
end
```

Sparse Regression to Solve for Active Terms in the Dynamics

SL Brunton, JL Proctor, JN Kutz (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems.

SINDy

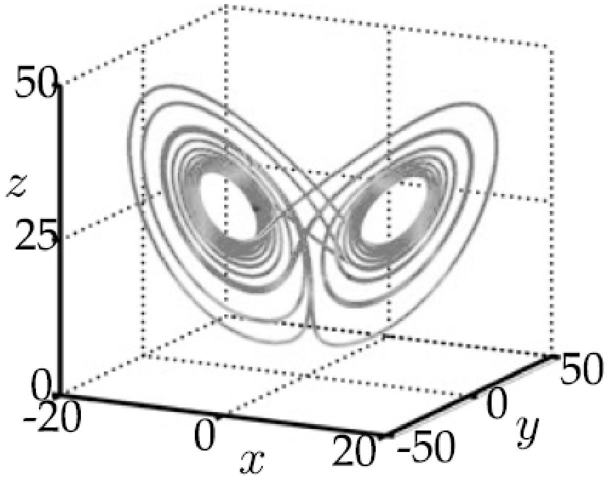


SL Brunton, JL Proctor, JN Kutz (2016) [Discovering governing equations from data by sparse identification of nonlinear dynamical systems]().

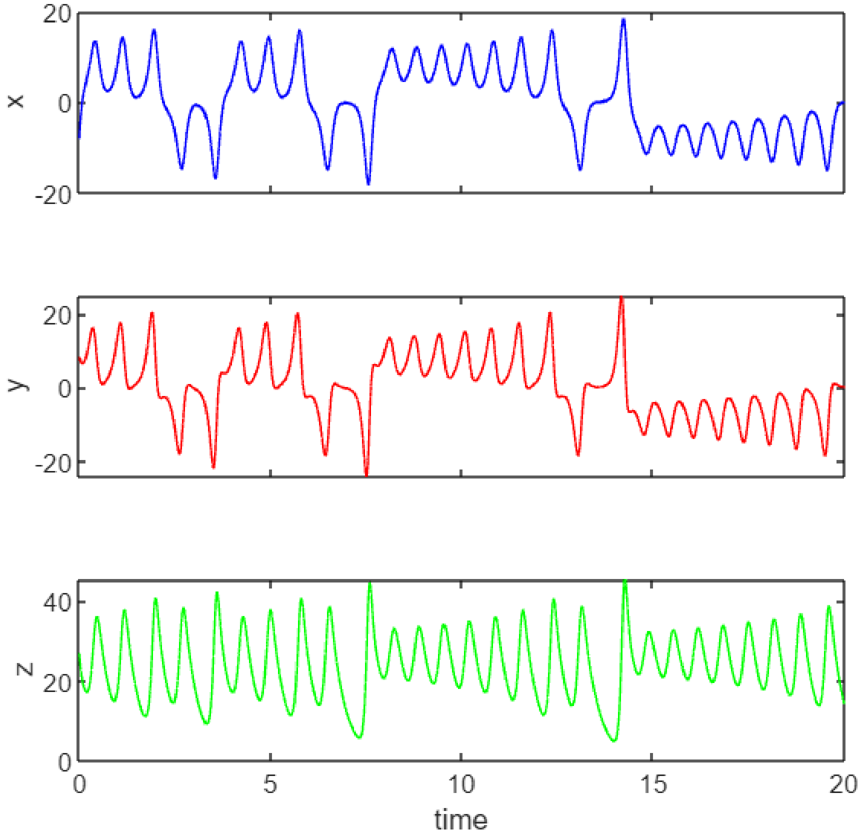
Tutorial 1 → https://github.com/urban-fasel/I-X_workshop_2025

Lorenz system

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= x(\rho - z) - y \\ \dot{z} &= xy - \beta z. \end{aligned}$$



Data: time series x, y, z



Alternative: MATLAB tutorial → https://github.com/urban-fasel/FiltonWorkshop2024

Challenges / limitations "vanilla" SINDy

Data	→	how much and what quality is needed?
Library	→	how to choose an effective library of candidate terms?
Optimization	→	what algorithm/regularization to use?
Model selection	→	how to select models / tune hyperparameters?
(**Coordinates**	→	do we measure the right variables?)

Tutorial 2 – challenges / limitations of SINDy

Workshop outline

Part 1: SINDy – Sparse Identification of Nonlinear Dynamics

- **Intro**: identifying ODEs
- Python **example**: *Lorenz system*
- **PySINDy**
- SINDy **challenges and limitations**

"Vanilla" SINDy

SINDy extensions

Part 2: Library (ODEs, PDEs, rational functions)

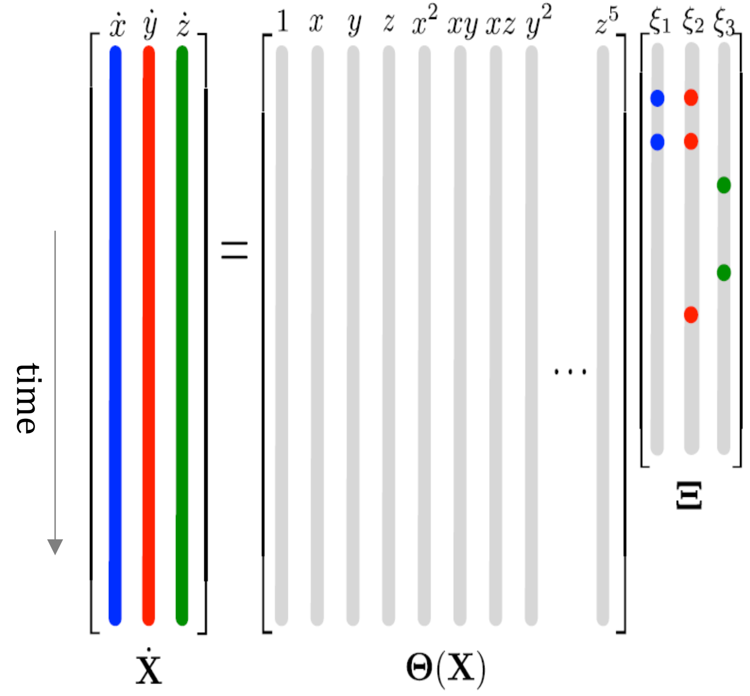
Part 3: Model selection

Part 4: Noise robustness: weak form & ensemble SINDy

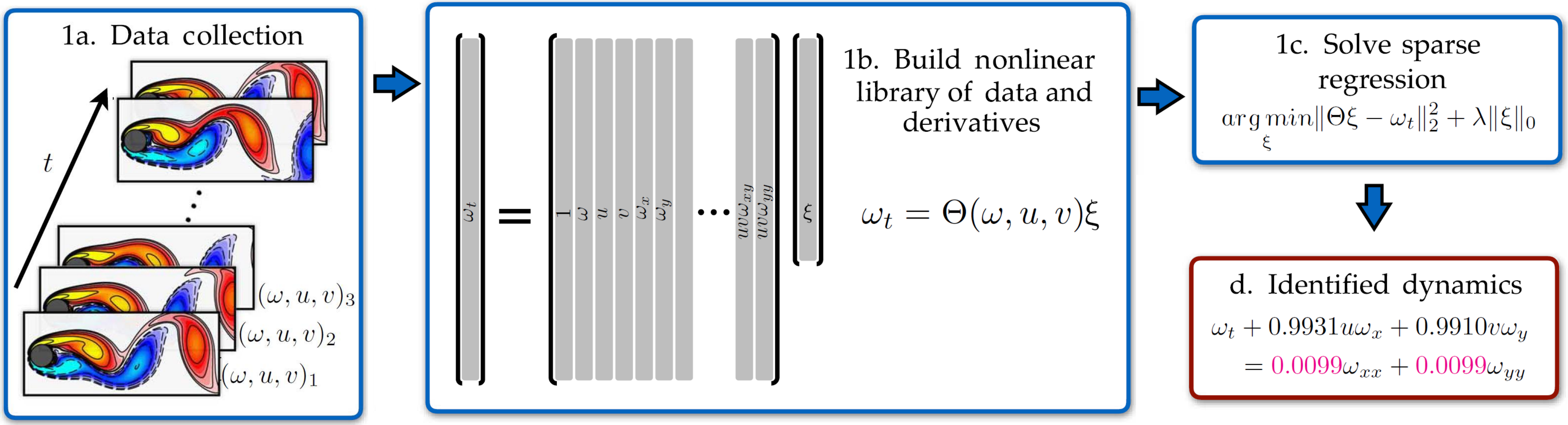
Library – general approach

General approach to selecting the library

- Start with linear terms → DMD model
  - Check accuracy
    - error reconstructing $\dot{\mathbf{X}}$
    - model prediction error
- Increase order
  - Add quadratic, then higher order polynomials
- Trigonometric functions
- <u>Generally</u>: try small, isolated libraries first

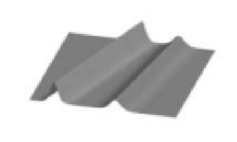
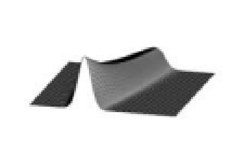
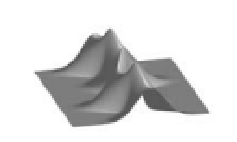
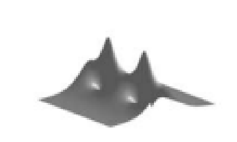
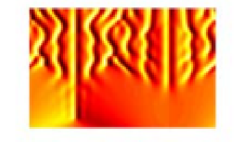
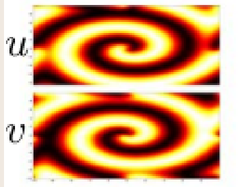


PDEs



SH Rudy, SL Brunton, JL Proctor, JN Kutz (2017*)* [Data-driven discovery of partial differential equations](#).

PDEs

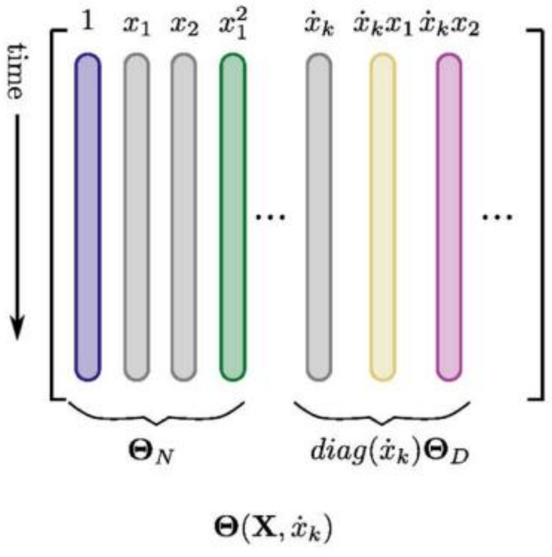
PDE	Form	Error (no noise, noise)	Discretization
 KdV	$u_t + 6uu_x + u_{xxx} = 0$	$1 \pm 0.2\%, 7 \pm 5\%$	$x \in [-30, 30], n = 512,\ t \in [0, 20], m = 201$
 Burgers	$u_t + uu_x - \epsilon u_{xx} = 0$	$0.15 \pm 0.06\%, 0.8 \pm 0.6\%$	$x \in [-8, 8], n = 256,\ t \in [0, 10], m = 101$
 Schrödinger	$iu_t + \frac{1}{2}u_{xx} - \frac{x^2}{2}u = 0$	$0.25 \pm 0.01\%, 10 \pm 7\%$	$x \in [-7.5, 7.5], n = 512,\ t \in [0, 10], m = 401$
 NLS	$iu_t + \frac{1}{2}u_{xx} + \|u\|^2 u = 0$	$0.05 \pm 0.01\%, 3 \pm 1\%$	$x \in [-5, 5], n = 512,\ t \in [0, \pi], m = 501$
 KS	$u_t + uu_x + u_{xx} + u_{xxxx} = 0$	$1.3 \pm 1.3\%, 52 \pm 1.4\%$	$x \in [0, 100], n = 1024,\ t \in [0, 100], m = 251$
 Reaction Diffusion	$u_t = 0.1\nabla^2 u + \lambda(A)u - \omega(A)v$ $v_t = 0.1\nabla^2 v + \omega(A)u + \lambda(A)v$ $A^2 = u^2 + v^2, \omega = -\beta A^2, \lambda = 1 - A^2$	$0.02 \pm 0.01\%,\ 3.8 \pm 2.4\%$	$x, y \in [-10, 10], n = 256,\ t \in [0, 10], m = 201$ subsample 1.14%
 Navier-Stokes	$\omega_t + (\mathbf{u} \cdot \nabla)\omega = \frac{1}{Re}\nabla^2 \omega$	$1 \pm 0.2\%\ ,\ 7 \pm 6\%$	$x \in [0, 9], n_x = 449,\ y \in [0, 4], n_y = 199,$ $t \in [0, 30], m = 151,$ subsample 2.22%

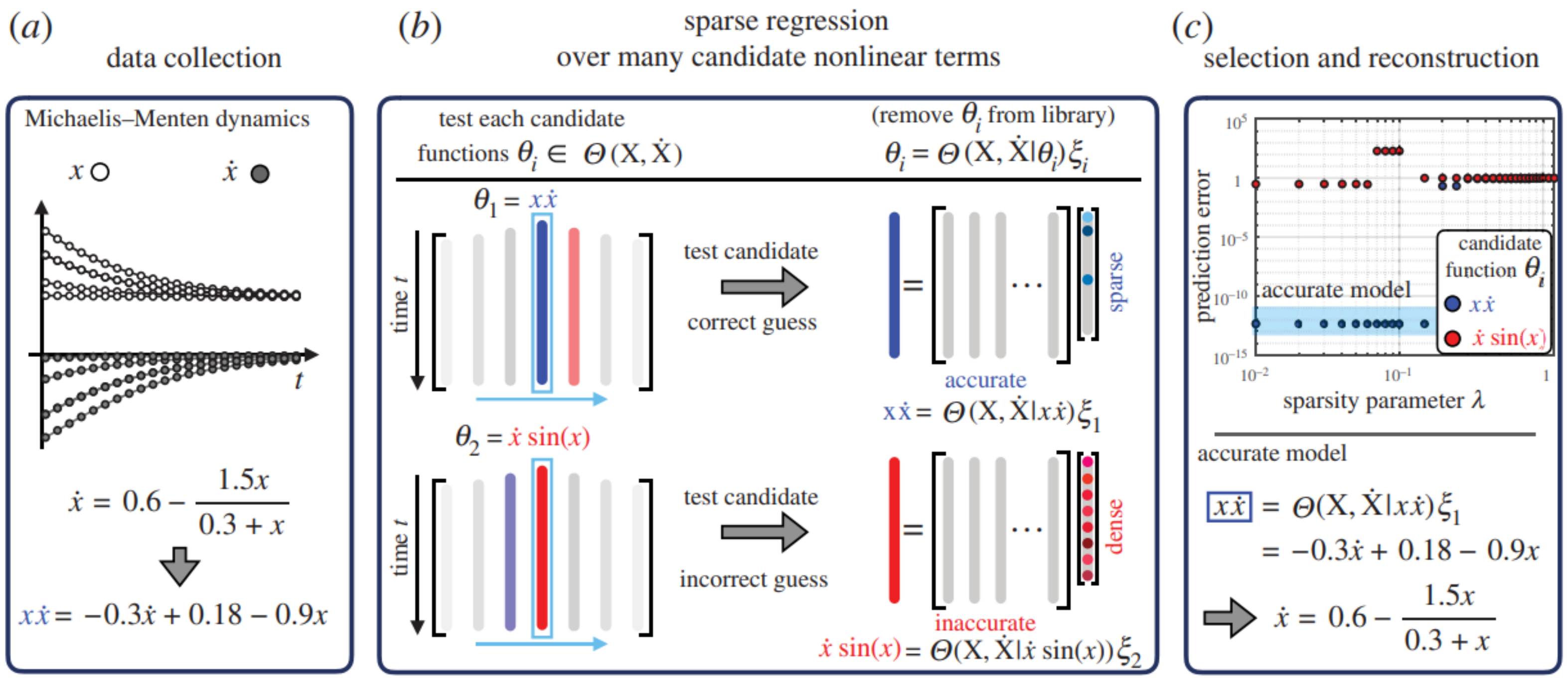
SH Rudy, SL Brunton, JL Proctor, JN Kutz (2017) [Data-driven discovery of partial differential equations](#).

Library – rational functions

Extending SINDy: handle larger classes of dynamical systems

- Rational function: $\dot{x}_k = \frac{f_N(\mathbf{x})}{f_D(\mathbf{x})} \quad \rightarrow \quad f_N(\mathbf{x}) - f_D(\mathbf{x})\dot{x}_k = 0 \quad \rightarrow \quad \boldsymbol{\Theta}(\mathbf{X}, \dot{x}_k)\boldsymbol{\xi}_k = \mathbf{0}$
  - difficult to describe as a linear combination of library features
  - e.g. biological systems: Michaelis-Menten dynamics: $\dot{x} = 0.6 - \frac{1.5x}{0.3+x}$



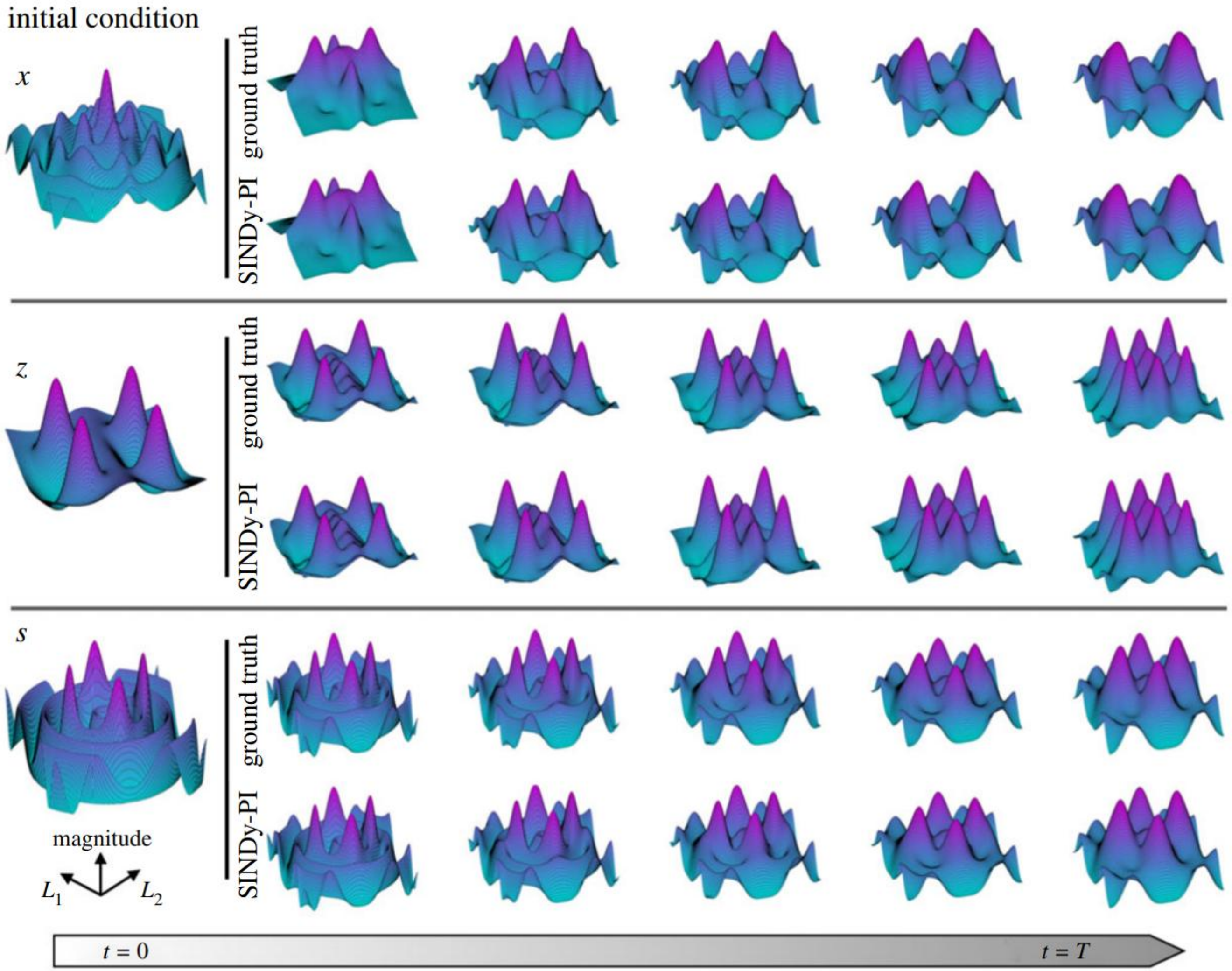


SINDy-PI algorithm

1. Test multiple possible left-hand sides (in parallel):
   - Move candidate terms to LHS
2. Calc model prediction error
3. Select best model:
   - Sparsity & accuracy

NM Mangan, SL Brunton, JL Proctor, JN Kutz (2016) Inferring Biological Networks by Sparse Identification of Nonlinear Dynamics
K Kaheman, JN Kutz, SL Brunton (2020) SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics

Library – rational functions



Belousov–Zhabotinsky reaction

- 4 coupled PDE with rational nonlinearities

$$\frac{\partial x}{\partial \tau} = \frac{1}{\varepsilon}\left(\frac{fz(q-x)}{q+x} + x - x^2 - \beta x + s\right) + \frac{D_x}{D_u}\Delta x,$$
$$\frac{\partial z}{\partial \tau} = x - z - \alpha z + \gamma u + \frac{D_z}{D_u}\Delta z,$$
$$\frac{\partial s}{\partial \tau} = \frac{1}{\varepsilon_2}(\beta x - s + \chi u) + \frac{D_s}{D_u}\Delta s,$$
$$\frac{\partial u}{\partial \tau} = \frac{1}{\varepsilon_3}\left[\alpha z - \left(\gamma + \frac{\chi}{2}\right)u\right] + \frac{D_u}{D_u}\Delta u,$$

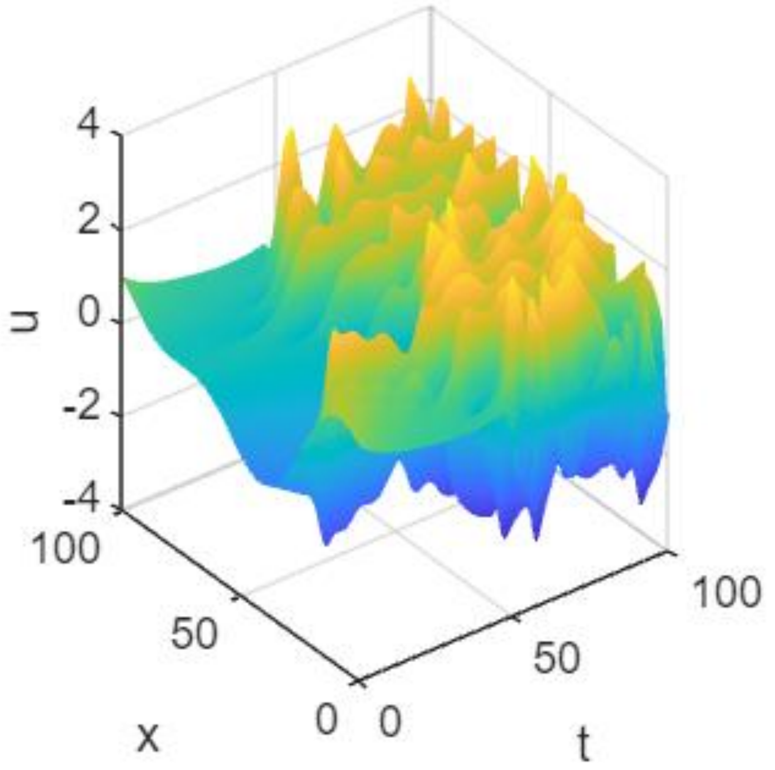
- SINDy-PI accurately identifies correct PDE
  - Not possible with standard SINDy
  - **MATLAB**: https://github.com/dynamicslab/SINDy-PI
  - **PySINDy**: interactive notebook

K Kaheman, JN Kutz, SL Brunton (2020) SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics

PDE tutorial – Kuramoto-Sivashinsky equation

- **Kuramoto-Sivashinsky equation**: $u_t = -uu_x - u_{xx} - u_{xxxx}$
  - **Describes** e.g. chaotic dynamics of laminar flame fronts (Sivashinsky 1977) or reaction-diffusion systems (Kuramoto and Tsuzuki 1076).
  - **Challenging** PDE to identify, because it involves higher order partial derivatives.

- **Tutorial 3**
  - **Library $\mathbf{\Theta}$**: 14 terms in the library
    - Polynomials: u, u^2
    - Partial derivatives: $u_x, u_{xx}, u_{xxx},\ u_{xxxx}$
    - Mixed terms: $u \cdot u_x,\ u \cdot u_{xx},\ u^2 \cdot u_x,\ \dots$
  - **Derivatives**: finite difference
  - **Optimizer**: sequentially thresholded ridge regression



Workshop outline

Part 1: SINDy – Sparse Identification of Nonlinear Dynamics

- **Intro**: identifying ODEs
- Python **example**: *Lorenz system*
- **PySINDy**
- SINDy **challenges and limitations**

"Vanilla" SINDy

SINDy extensions

Part 2: Library (ODEs, PDEs, rational functions)

Part 3: Model selection

Part 4: Noise robustness: weak form & ensemble SINDy

Model selection

Model selection is **not simply about reducing error**, but about producing a model that has high degree of **interpretability**, **generalization**, and **predictive capabilities**.

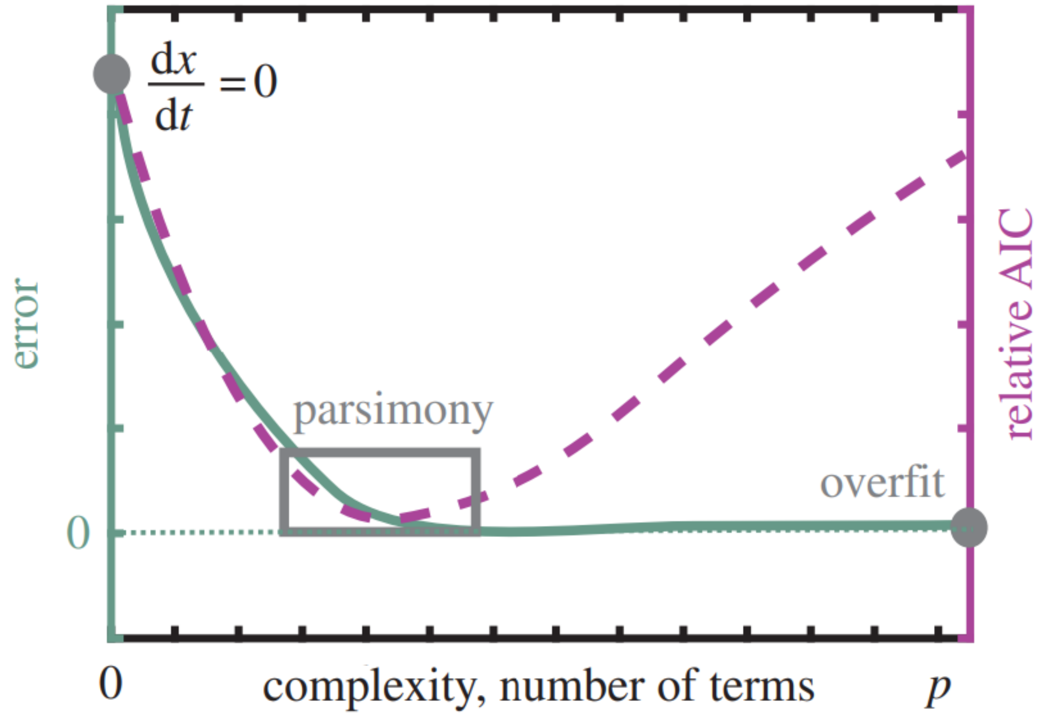
→ selecting **accurate** and **sparse** model

Model selection methods

- Cross validation
- Prediction error & sparsity
- Stability selection

Tutorials

- Compare methods on Lorenz system data

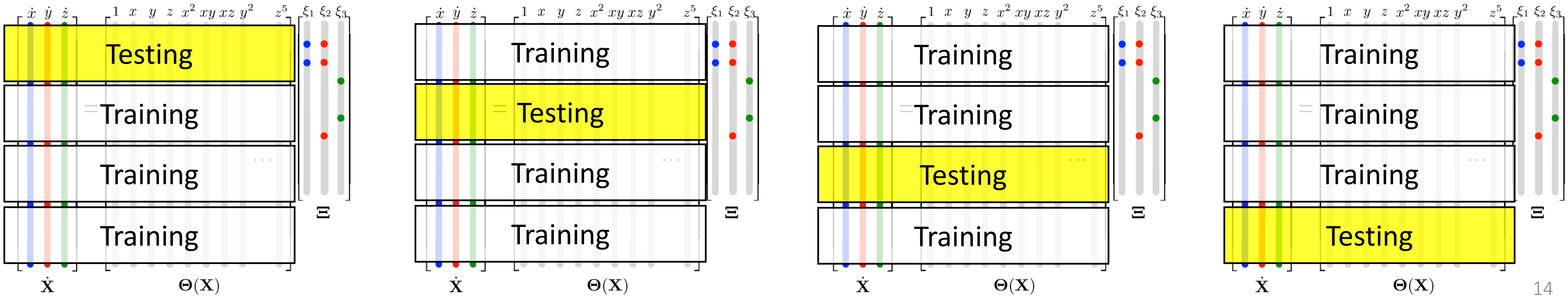


Model selection – k-fold cross-validation

Model selection: sweep through λ-path → repeat following 3 steps for increasing λ (model complexity)

1. Partition data in k (random) subsamples. *(e.g. 4-fold CV shown here)*
2. Build k SINDy models:
   - use $k-1$ subsamples for training
   - test the model using the withheld testing sample: $\epsilon_k = \left\|\dot{\mathbf{X}}_k - \boldsymbol{\Theta}(\mathbf{X}_k)\boldsymbol{\xi}_k\right\|_2^2$
3. Average the test scores ϵ_k → $\epsilon_\lambda = \frac{1}{k}\sum\epsilon_k$

Select model (choose λ) with lowest test score ϵ_λ



Model selection – Prediction error & sparsity

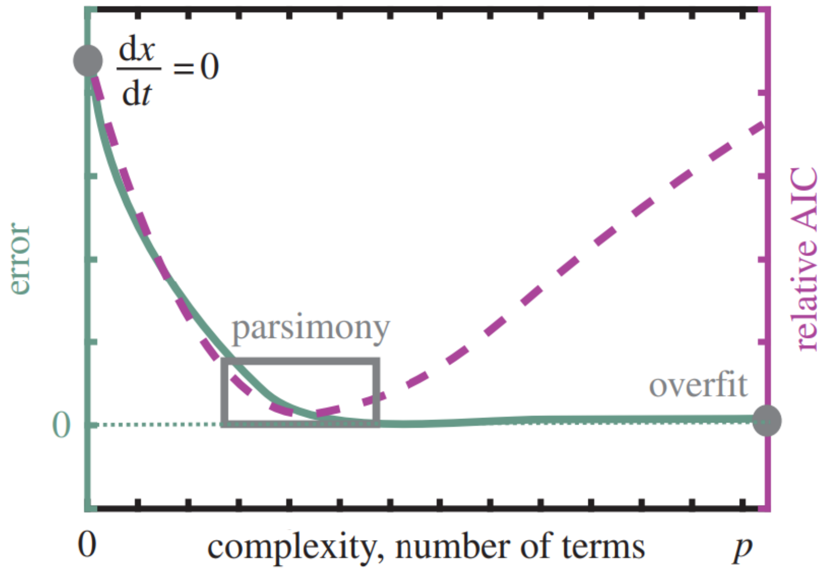
Sum of model prediction error plus number of active terms

- SINDy model prediction error $\rightarrow$ $E_{avg} = \sum_\tau |y_i - g(x_i;\mu)|$
  - y_i observed outcomes (time series data)
  - $g(x_i;\mu)$ SINDy model prediction (numerically integrated SINDy ODE)

- **Model score**: $\mathrm{AIC} = m \ln\left(\frac{1}{m}\left(\sum_{i=1}^{m} \mathrm{E_{avg}}\right)^2\right) + 2k$
  - m number of test time series (e.g. starting from different initial conditions)
  - k number of non-zero terms

model quality (prediction error)

model complexity (number of terms)



- We can compute AIC score for a range of λ $\rightarrow$ sweep through λ-path
  - Elbow (minimum) in AIC curve: *allows for intuitive model selection*

Idea introduced first in SINDy Akaike Information Criteria paper: NM Mangan, JN Kutz, SL Brunton, JL Proctor (2017) [Model selection for dynamical systems via sparse regression and information criteria]

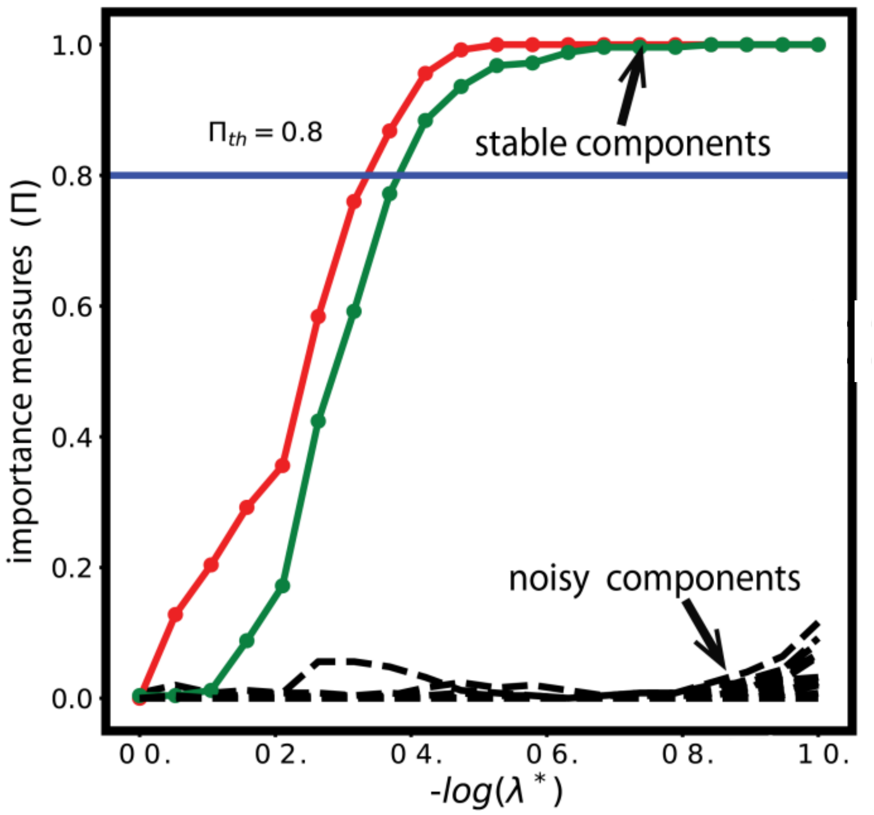
Model selection – stability selection

Calculate the stability (importance) of each coefficient over the regularization path λ

1. Generate B random subsamples (size $N/2$) of the library and derivative data without replacement

2. Compute B SINDy models over regularization path λ

3. Calculate the λ-dependent stability (or importance) Π_k^λ
   - **Stability**: probability of a coefficient to be non-zero
   - Find stable support: $\Pi_k^\lambda > \pi_{th} = 0.8$
     - e.g. coefficient is non-zero in 80% of the models



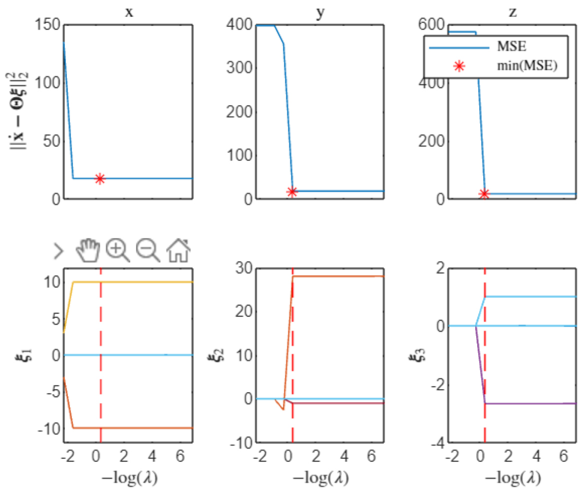
S Maddu, BL Cheeseman, IF Sbalzarini, CL Muller (2019) Stability selection enables robust learning of partial differential equations from limited noisy data.

Model selection – summary

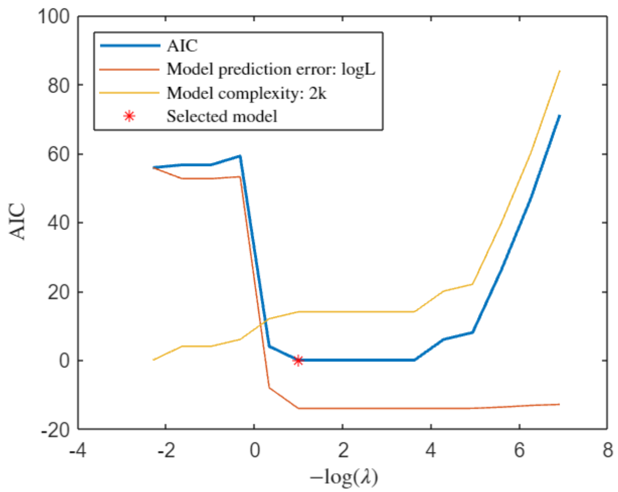
Model selection is **not simply about reducing error**, but about producing a model that has high degree of **interpretability**, **generalization**, and **predictive capabilities.**

→ selecting **accurate** and **sparse** model

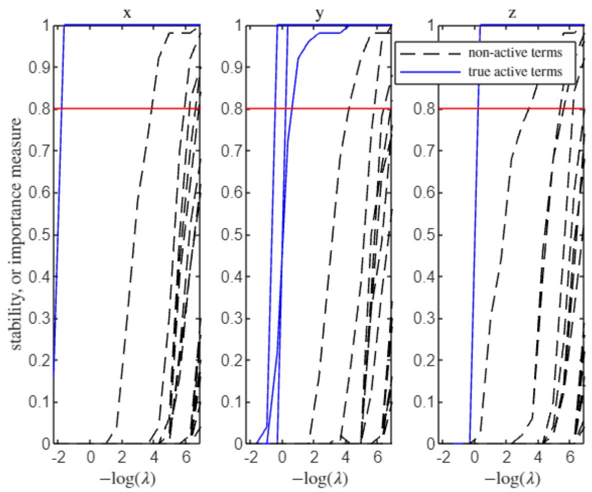
Cross validation



Prediction error & sparsity



Stability selection



Tutorials model selection

- **Workshop tutorial**
  - https://github.com/urban-fasel/I-X_workshop_2025
- **PySINDy**
  - Prediction error and sparsity: AIC is implemented in example 16
- **MATLAB live scripts**
  - https://github.com/urban-fasel/FiltonWorkshop2024

Workshop outline

Part 1: SINDy – Sparse Identification of Nonlinear Dynamics

- **Intro**: identifying ODEs
- Python **example**: *Lorenz system*
- **PySINDy**
- SINDy **challenges and limitations**

"Vanilla" SINDy

SINDy extensions

Part 2: Library (ODEs, PDEs, rational functions)

Part 3: Model selection

Part 4: Noise robustness: weak form & ensemble SINDy

Noise robustness: weak form & ensemble SINDy

Literature

- **Ensemble SINDy:**
  - U Fasel, JN Kutz, BW Brunton, SL Brunton (2022) Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control.
  - SM Hirsh, DA Barajas-Solano, JN Kutz (2022) Sparsifying priors for Bayesian uncertainty quantification in model discovery.
  - LM Gao, U Fasel, SL Brunton, JN Kutz (2023) Convergence of uncertainty estimates in Ensemble and Bayesian sparse model discovery.
  - U Fasel (2025) Sparse Identification of Nonlinear Dynamics with Conformal Prediction.

- **Bayesian SINDy:**
  - SM Hirsh, DA Barajas-Solano, JN Kutz (2022) Sparsifying priors for Bayesian uncertainty quantification in model discovery.
  - L Fung, U Fasel, M Juniper (2025) Rapid Bayesian identification of sparse nonlinear dynamics from scarce and noisy data.

- **Integral and weak form SINDy:**
  - H Schaeffer, SG McCalla (2017) Sparse model selection via integral terms.
  - PAK Reinbold, DR Gurevich, RO Grigoriev (2020) Using noisy or incomplete data to discover models of spatiotemporal dynamics.
  - D Messenger, DM Bortz (2021) Weak SINDy for partial differential equations.
  - AA Kaptanoglu, …, ZG Nicolaou, … (2022) **PySINDy**, SINDyCP

Ensemble SINDy

Combining predictions from multiple models

→ Improve accuracy of model prediction

→ Reduce variance and help avoid overfitting

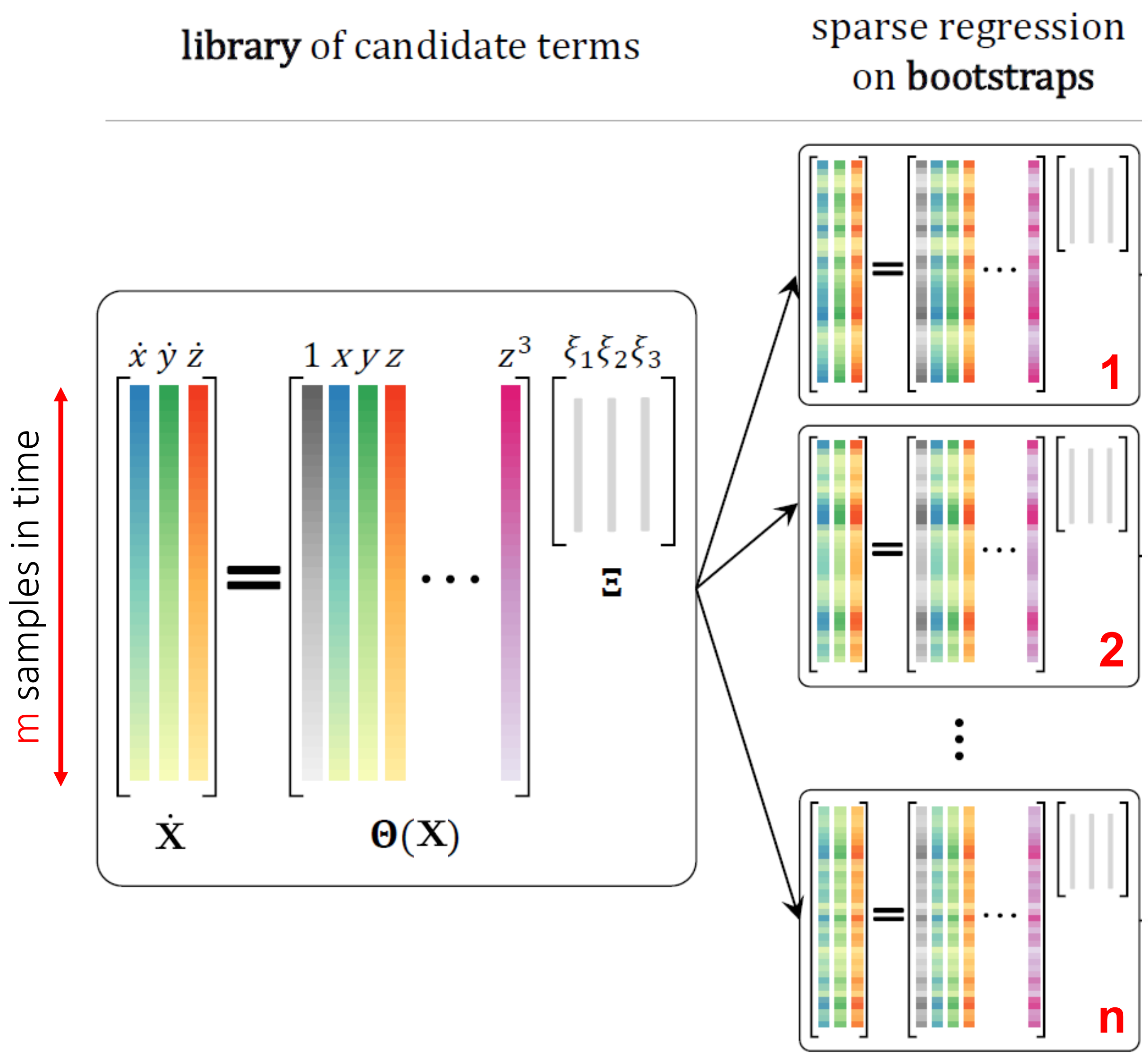
Popular ensemble learning method

→ Bootstrap aggregating or bagging

Connections to model selection

→ Cross-validation and stability selection use ensemble of models to select best model

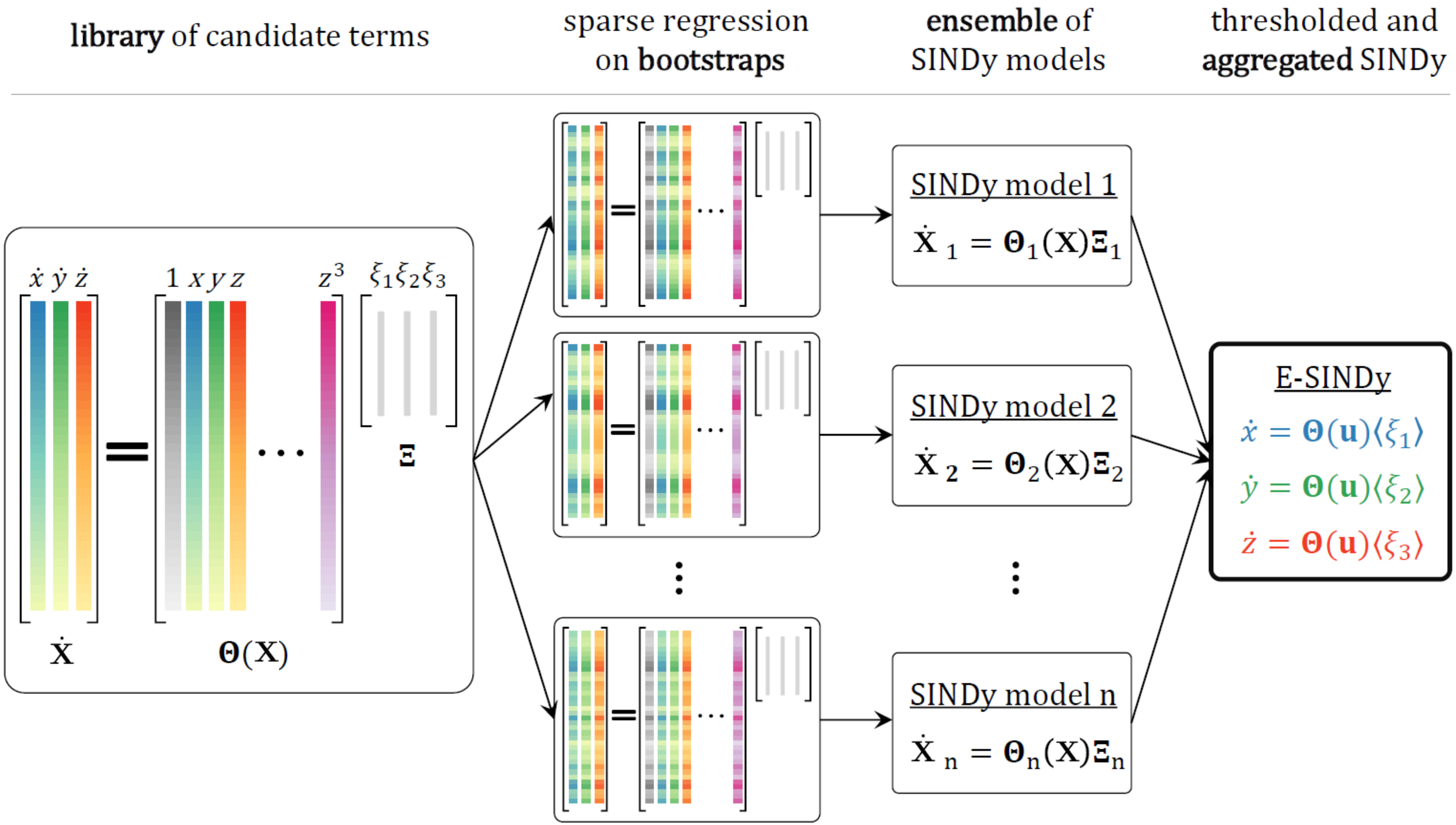
Ensemble SINDy



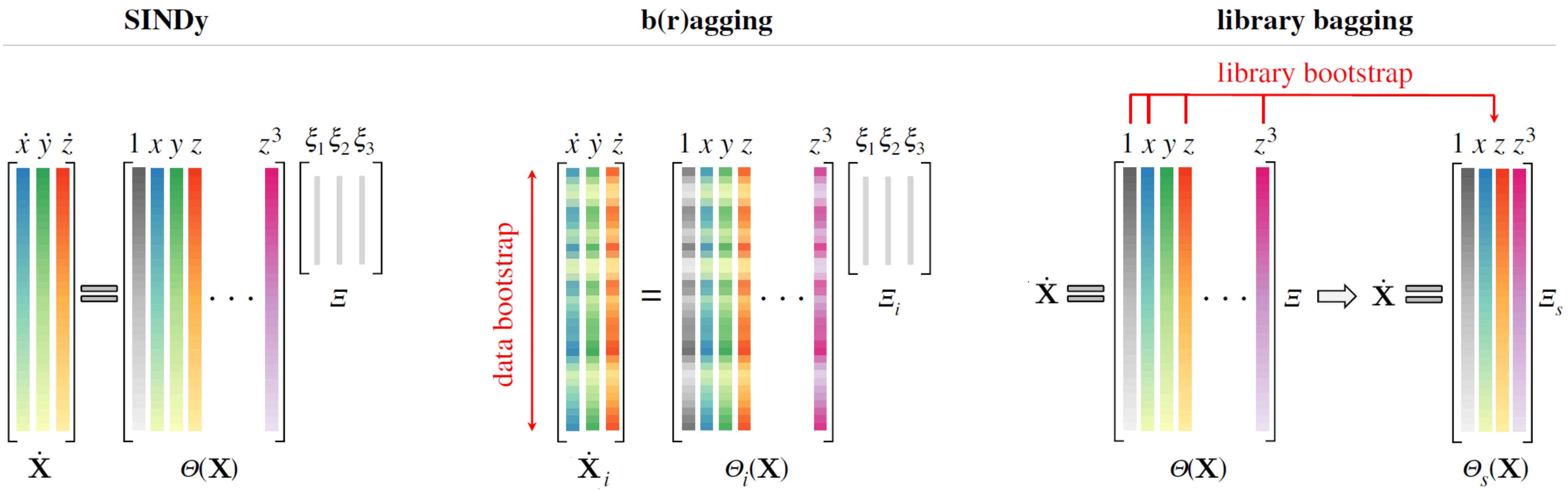
SINDy Bootstrap:

- randomly sample m new rows with replacement from the original m rows of the data matrices.
- generate n bootstraps and identify n SINDy models in the ensemble

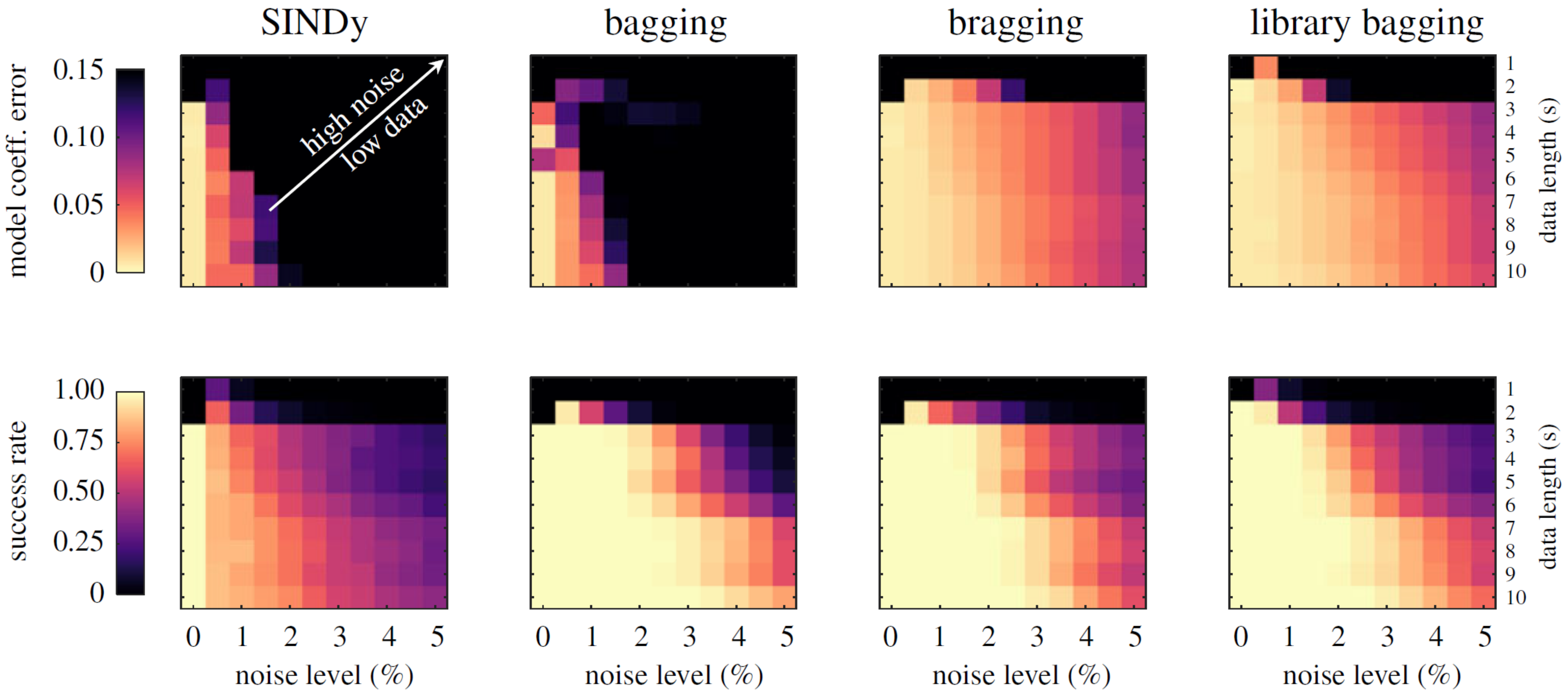
Ensemble SINDy



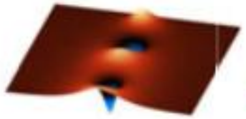
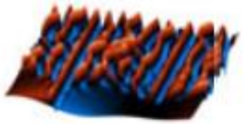
Ensemble SINDy



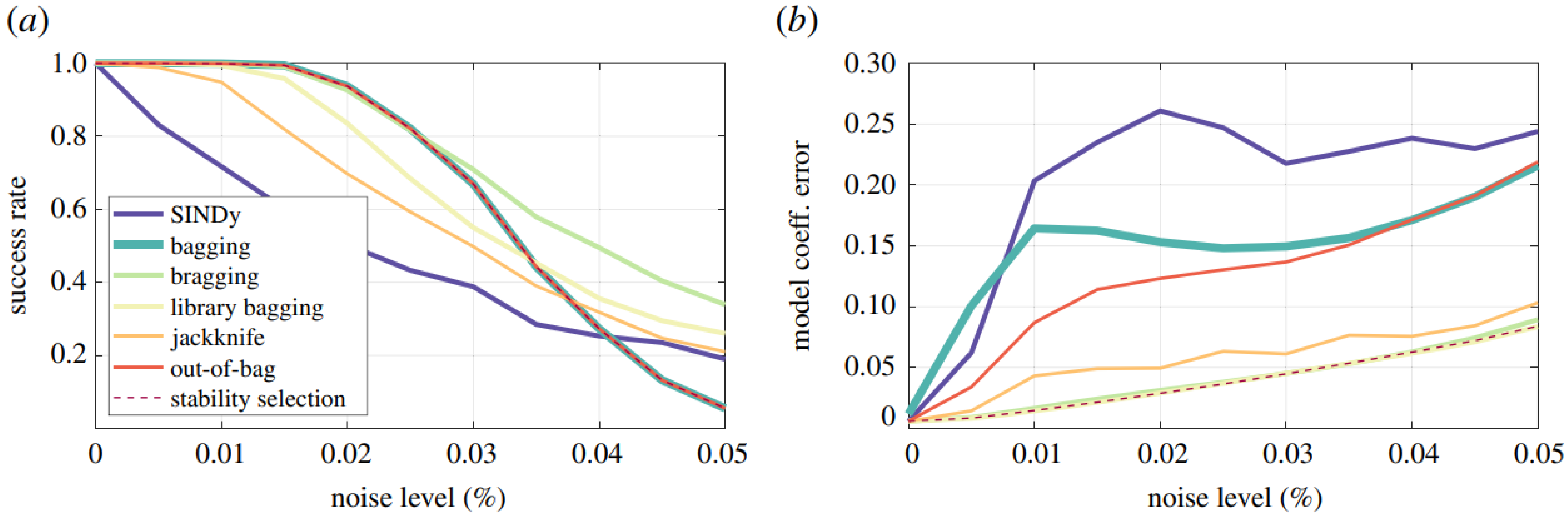
Ensemble SINDy



Ensemble SINDy

PDE		noise level		form	model error WSINDy | E-WSINDy	success rate WSINDy | E-WSINDy
	inviscid Burgers		100%	$u_t + 0.5uu_x = 0$	2.6% | **2.5%**	99% | **100%**
	Korteweg de Vries		100%	$u_t + 0.5uu_x + u_{xxx} = 0$	27.5% | **4.0%**	93.5% | **100%**
	nonlinear Schrödinger		50%	$iu_t + 0.5u_{xx} + \|u\|^2u = 0$	13.0% | **11.3%**	82.0% | **100%**
	Kuramoto–Sivashinsky		100%	$u_t + 0.5uu_x + u_{xx} + u_{xxxx} = 0$	29.5% | **24.7%**	87.5% | **99.5%**
	reaction–diffusion		20%	$u_t = 0.1\nabla^2 u + \lambda(A)u - \omega(A)v$ $v_t = 0.1\nabla^2 v + \omega(A)u + \lambda(A)v$ $A^2 = u^2 + v^2,\ \omega = -\beta A^2,\ \lambda = 1 - A^2$	77.7% | **7.1%**	0.0% | **99.5%**

Ensemble SINDy – different sampling strategies



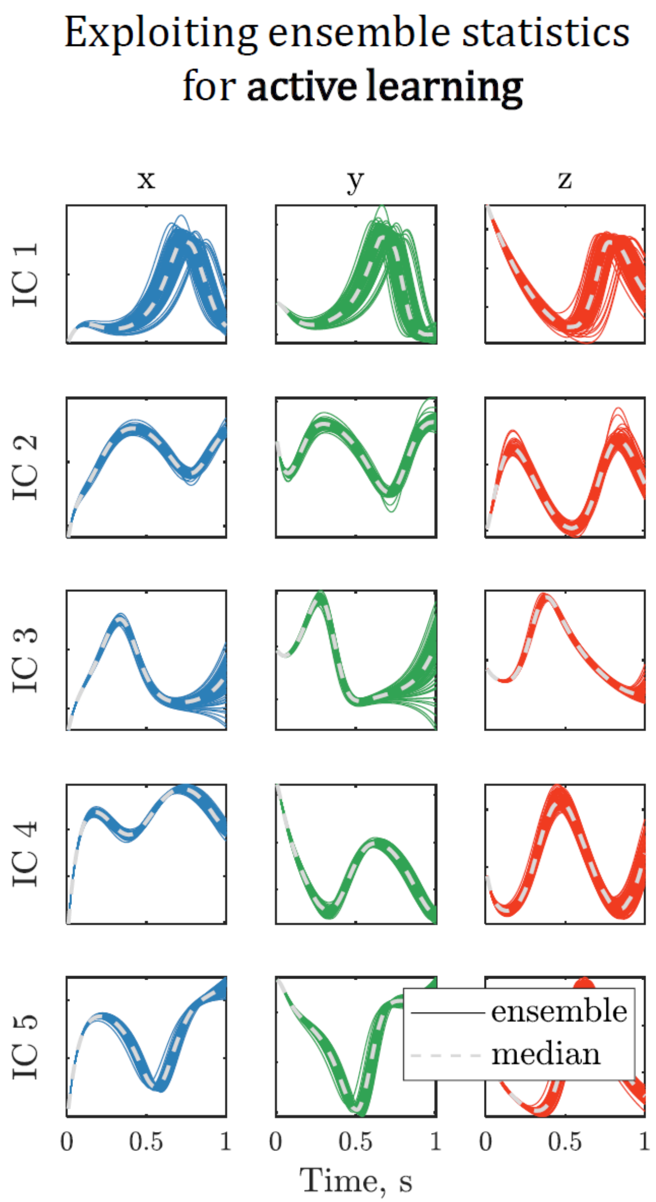
(a) Success rate: identifying correct model structure

(b) model coefficient error

Bragging (robust bagging) SINDy generally performs best

- Robust bagging: aggregate by taking median of identified models

Ensemble SINDy – active learning



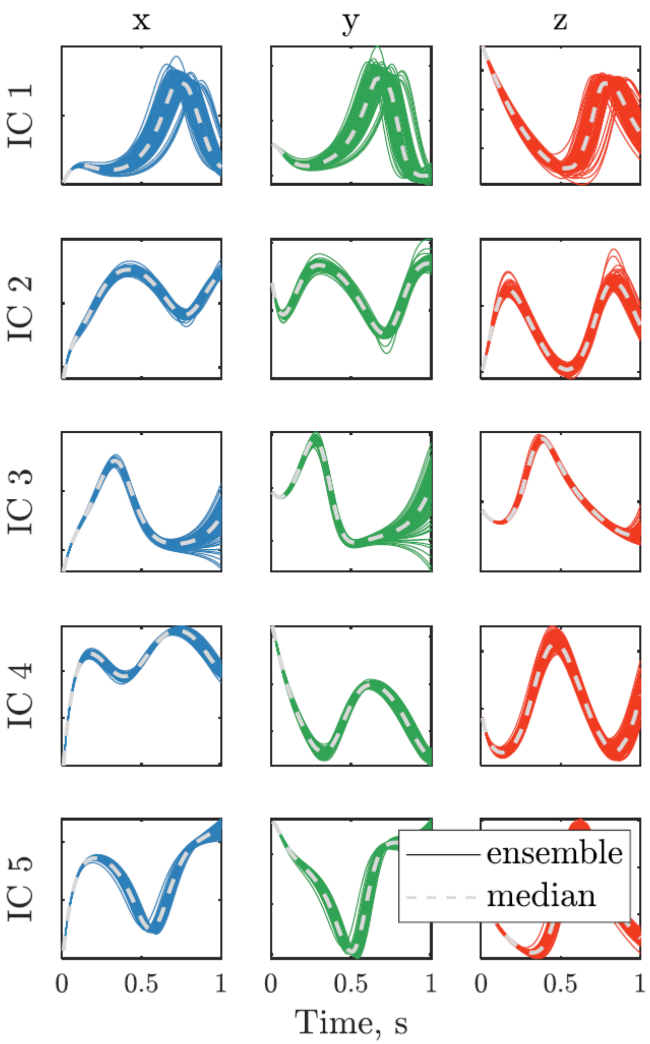
- Collecting data may be time consuming and expensive
- **Efficient exploration**: maximally inform model discovery

E-SINDy active learning:

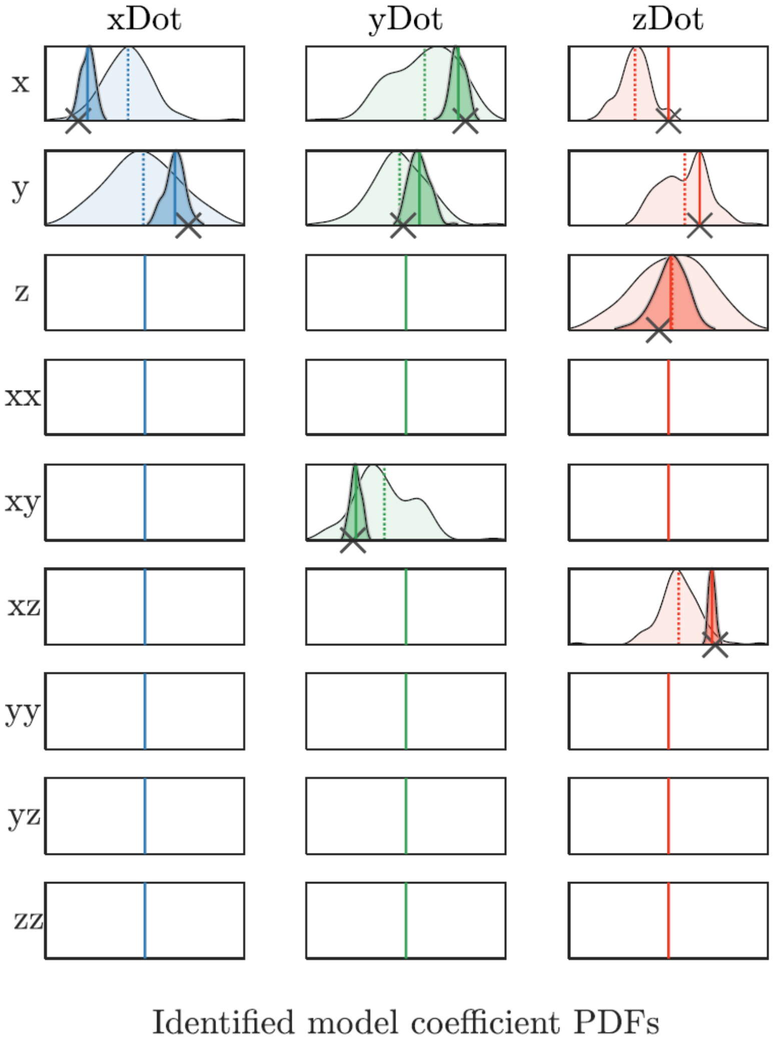
- Exploit ensemble statistics to **find and explore high uncertainty regions** in the feature space
- Iteratively identify E-SINDy models to **improve data efficiency** of model discovery

Ensemble SINDy – active learning

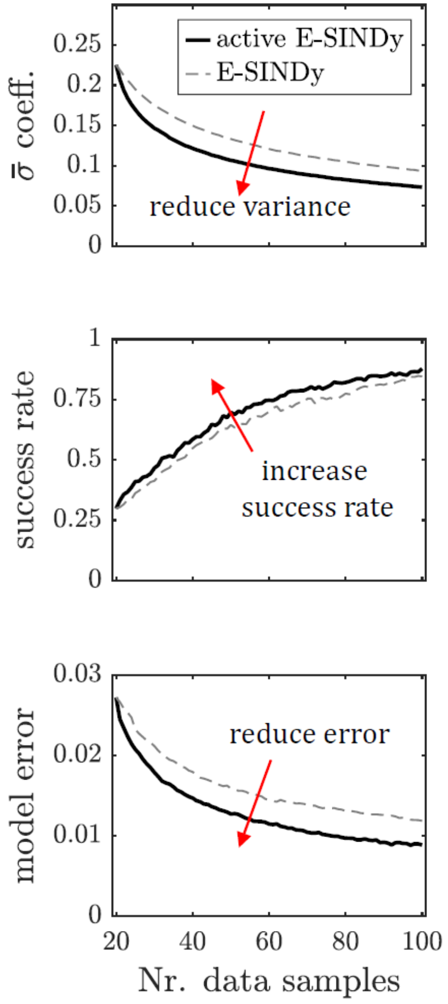
Exploiting ensemble statistics
for **active learning**



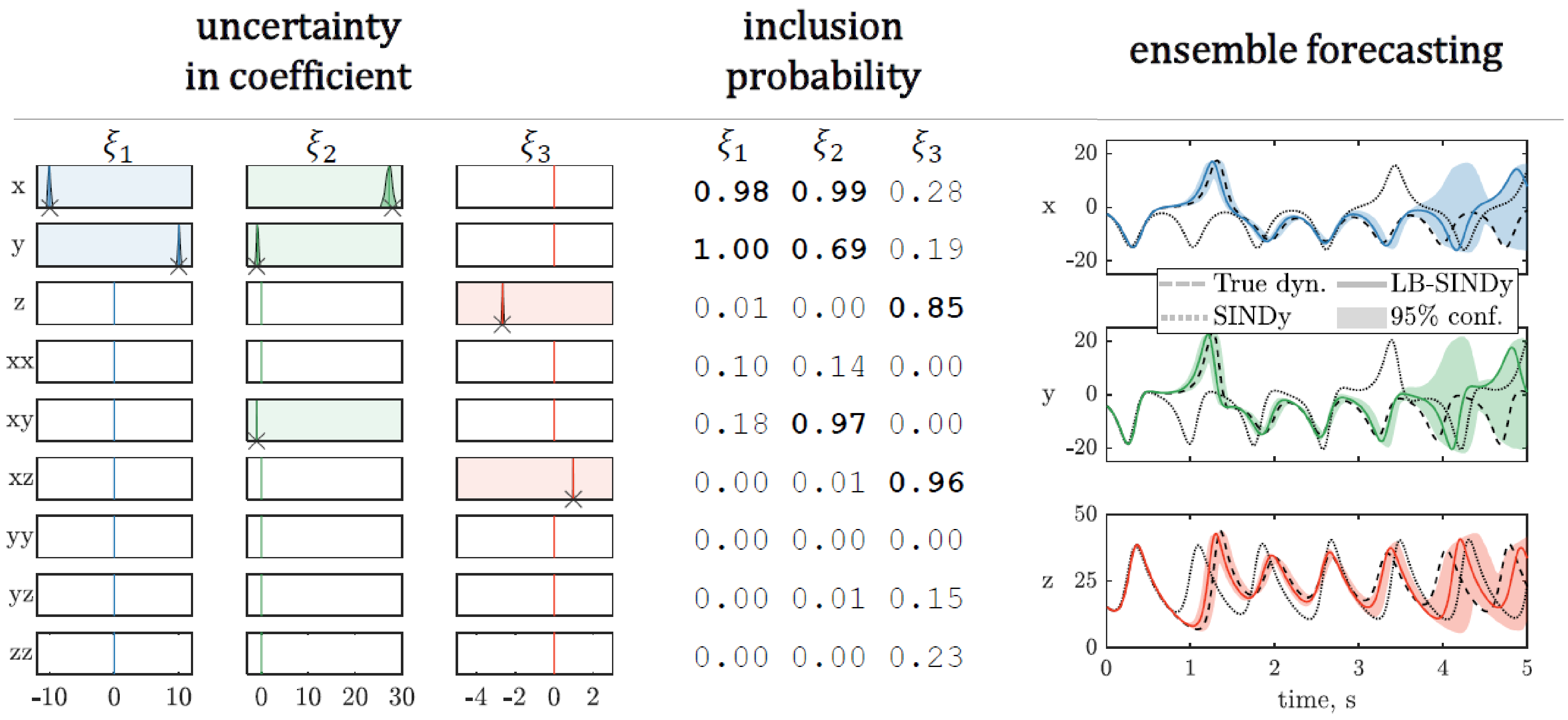
Reducing variance
of model coefficient



Improving **data-efficiency**
of model discovery



Ensemble SINDy – probabilistic models



- **Robust model discovery**: high noise, low data limit
- E-SINDy identifies **probabilistic models** at low computational cost

Ensemble SINDy tutorials

- **Workshop tutorial**
  - https://github.com/urban-fasel/I-X_workshop_2025
- **PySINDy**
  - https://github.com/dynamicslab/pysindy/blob/master/examples/13_ensembling.ipynb
  - ODE and PDE examples
- **MATLAB**
  - **E-SINDy vs SINDy**: Identify Lorenz system on noisy data
  - GitHub RSPA paper: https://github.com/urban-fasel/EnsembleSINDy
    - ODE and PDE examples
    - Active learning
    - MPC

Some E-SINDy follow up papers …

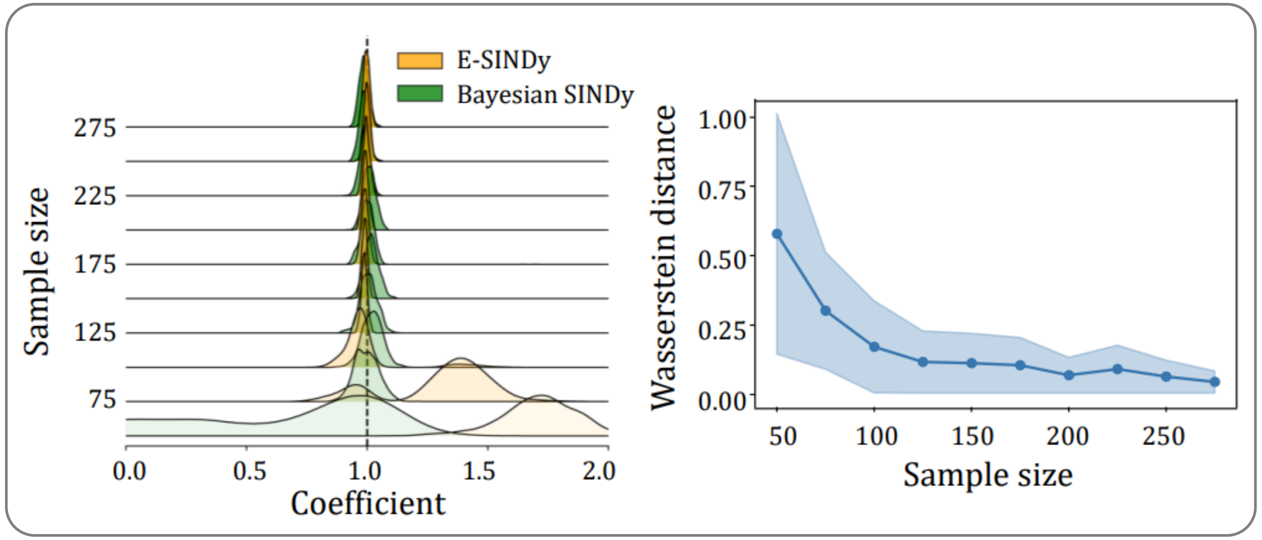
E-SINDy connection to UQ and Bayesian SINDy…

→ is asymptotically equivalent to Bayesian-SINDy

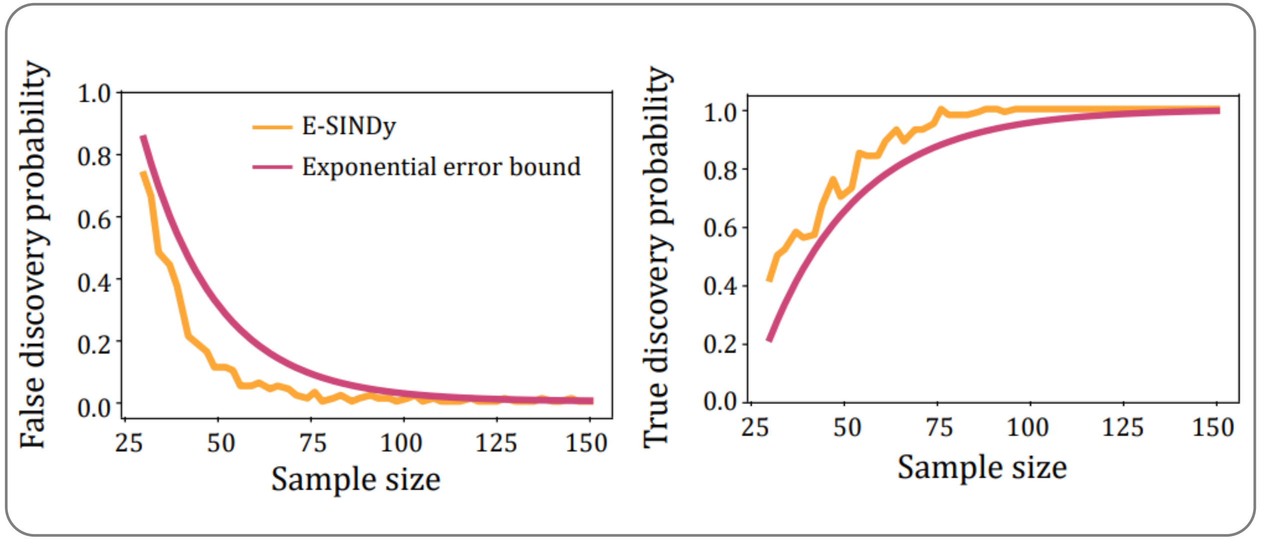
→ has lower computational cost → ≈3000 times less expensive

→ can perform valid uncertainty quantification with statistical guarantee

(a) Uncertainty quantification guarantees



(b) Exponential convergence theorem



Paper: LM Gao, U Fasel, SL Brunton, JN Kutz (2023) Convergence of uncertainty estimates in Ensemble and Bayesian sparse model discovery.

Some E-SINDy follow up papers ...

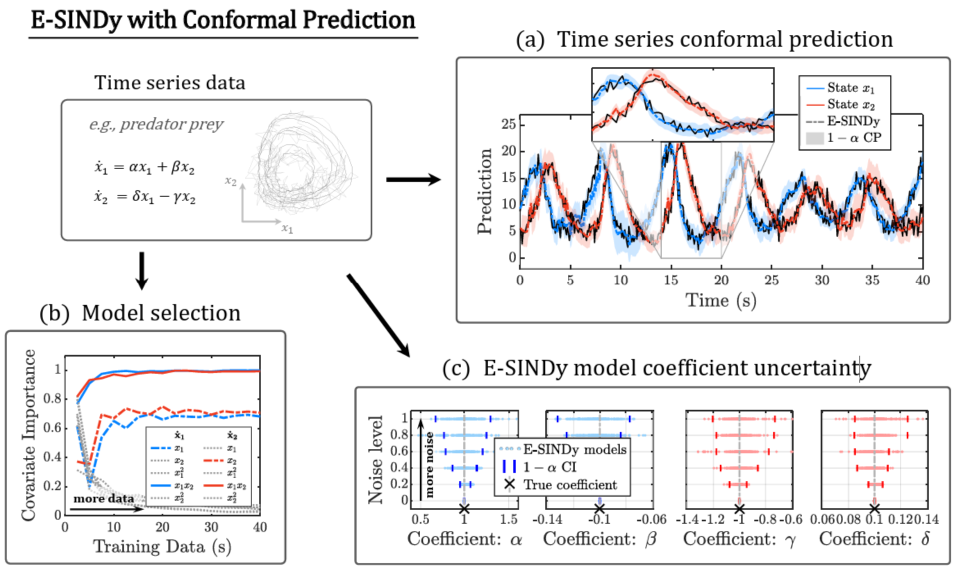
E-SINDy with conformal prediction

→ uncertainty quantification in time series prediction with **coverage guarantees**

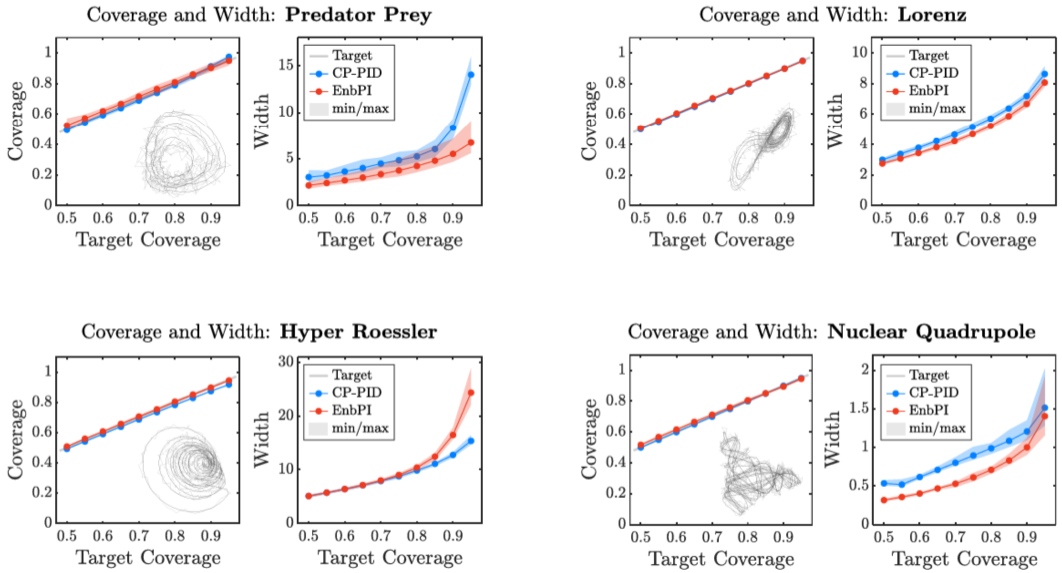
→ **model selection** using library feature (covariate) importance measure

→ quantifying uncertainty in the identified **model coefficients**

(a) Method overview



(b) Coverage and width over target coverage



Paper: U Fasel (2025) Sparse Identification of Nonlinear Dynamics with Conformal Prediction.

Weak SINDy

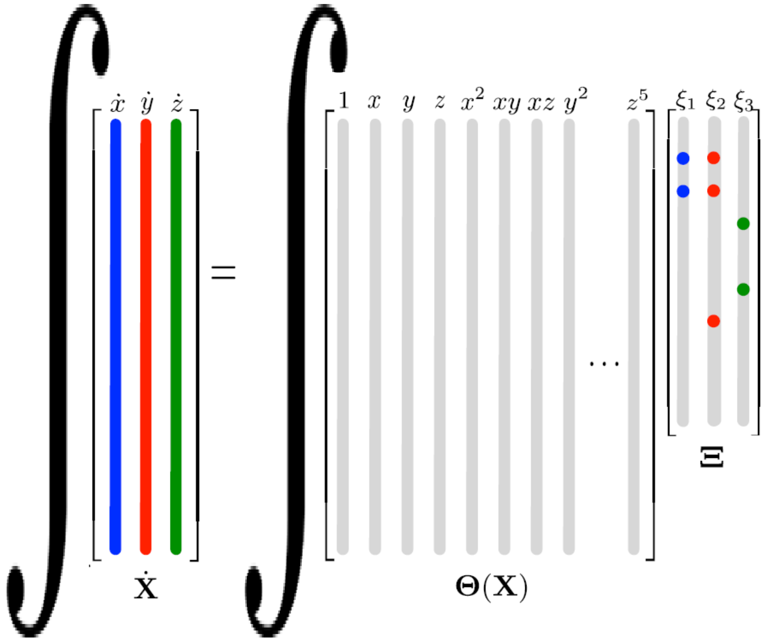
Integral formulation: H Schaeffer, SG McCalla (2017) [Sparse model selection via integral terms](#).

- Reformulate $\dot{x}_i = \Theta(\boldsymbol{x})\boldsymbol{\xi}_i = \sum f_j(\boldsymbol{x})\boldsymbol{\xi}_{i,j}$ to integral form

  $\rightarrow x_i(t) = x_i(0) + \sum d_j(\boldsymbol{x}, t)\xi_{i,j}$

  - $d_j(x,t) = \int_0^t f_j\big(\boldsymbol{x}(\tau)\big)d\tau$ integrated library function
  - $f_j(\boldsymbol{x})$ library function
  - $\xi_{i,j}$ model coefficient

- Solve ℓ_0-penalized least squares → same as standard SINDy



Weak form SINDy: generalizes integral form SINDy

- Integration against test function
- Moving derivatives off of the data onto test functions via integration by parts
- Different implementations
  - PAK Reinbold, DR Gurevich, RO Grigoriev (2020) Using noisy or incomplete data to discover models of spatiotemporal dynamics.
  - D Messenger, DM Bortz (2021) Weak SINDy for partial differential equations.
  - AA Kaptanoglu, …, ZG Nicolaou, … (2022) **PySINDy**, SINDyCP

Weak SINDy

Idea: Move derivatives off of the data onto a test functions via integration by parts

- PDE: $u_t = \mathbf{\Theta}(u)\boldsymbol{\xi} = \mathbf{\Theta}(u, u_x, u_{xx}, \dots, x)\boldsymbol{\xi}$
  - $u(x,t)$: state, $\mathbf{\Theta}$: library, $\boldsymbol{\xi}$: coefficients x: space, t: time
- Test function: $\phi(x,t)$ smooth & compactly supported: $\phi(x,t)$ *vanishes along domain bound*
  - e.g. $\phi(x,t) = (x^2-1)^p (t^2-1)^q$ on domain $\Omega = \{(\mathrm{x},\mathrm{t}) : |\mathrm{x}| \leq 1,\ |\mathrm{t}| \leq 1\}$

Integration by parts: remember product rule $\big(f(x)g(x)\big)' = f'(x)g(x) + f(x)g'(x)$

- $\int_a^b f(x)g'(x)dx = \int_a^b \big(f(x)g(x)\big)'dx - \int_a^b f'(x)g(x)dx = f(b)g(b) - f(a)g(a) - \int_a^b f'(x)g(x)dx$

Weak form: Multiply each term in PDE with test function and integrate over k different domains Ω_k

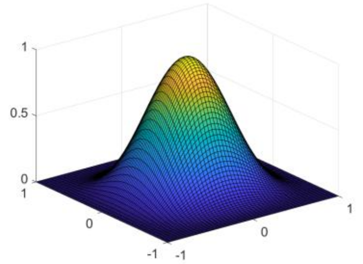
- Domain Ω_k, e.g. 2D space x and time t, $d\Omega = dx\,dt$, $\partial\Omega_k$: domain bounds
- LHS: $\int_{\Omega_k} \phi u_t \, d\Omega = \cancelto{0}{[\phi(x,t)u(x,t)]_{\partial\Omega_k}} - \int_{\Omega_k} \phi_t u \, du = -\int_{\Omega_k} \phi_t u \, du$
- RHS: term by term integration by parts: move derivatives $u_x, u_{xx}, \dots$ from data onto test function

Weak form parameter estimation (SINDy on next slide)

Kuramoto-Sivashinsky equation: $u_t = -uu_x - u_{xx} - u_{xxxx}$

1. **Integration by parts**: multiply each term with test function and move derivative onto test function

   - Term u_t: $q_0^k = \int_{\Omega_k} \phi u_t \, d\Omega = -\int_{\Omega_k} \phi_t u \, d\Omega$
   - Term uu_x: $q_1^k = \int_{\Omega_k} \phi u \, u_x \, d\Omega = -\int_{\Omega_k} (\phi_x u + \phi u_x) \, u \, d\Omega = -\int_{\Omega_k} \phi_x u^2 \, d\Omega - \int_{\Omega_k} \phi u \, u_x \, d\Omega \quad \rightarrow \quad \int_{\Omega_k} \phi u u_x \, d\Omega = -\frac{1}{2} \int_{\Omega_k} \phi_x u^2 \, d\Omega$
   - Term u_{xx}: $q_2^k = \int_{\Omega_k} \phi u_{xx} \, d\Omega = -\int_{\Omega_k} \phi_x u_x \, d\Omega = \int_{\Omega_k} \phi_{xx} u \, d\Omega$
   - Term u_{xxxx}: $q_3^k = \int_{\Omega_k} \phi u_{xxxx} \, d\Omega = \int_{\Omega_k} \phi_{xxxx} u \, d\Omega$

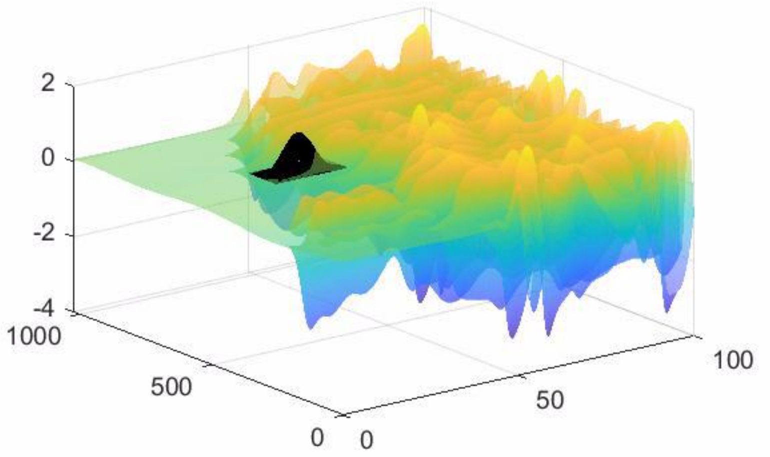


2. **Test function**: $\phi = \left(\underline{x}^2 - 1\right)^p \left(\underline{t}^2 - 1\right)^q$

   - $p = 4$, $q = 3$, $\underline{t} = (t - t_k)/H_t$, $\underline{x} = (x - x_k)/H_x$

3. **Integrate over domain**: $\Omega_k = \{(x,t) : |x - x_k| \leq H_x, |t - t_k| \leq H_t\}$

   - Integrate all q_i^k over N different domains Ω_k of size H_x and H_t



New system: $\boldsymbol{q}_0 = \sum_{n=1}^{3} \boldsymbol{q}_n \xi_n = \boldsymbol{Q}\boldsymbol{\xi}$ $\qquad \boldsymbol{q}_0 \in \mathbb{R}^{\mathrm{k}}$, $\boldsymbol{Q} \in \mathbb{R}^{\mathrm{k} \times 3}$, $\boldsymbol{\xi} \in \mathbb{R}^3$

$\rightarrow \quad \hat{\boldsymbol{\xi}} = \operatorname{argmin}_{\xi} \, \|\boldsymbol{q}_0 - \boldsymbol{Q}\boldsymbol{\xi}\|_2^2 \; + \; \lambda \|\xi\|_0$

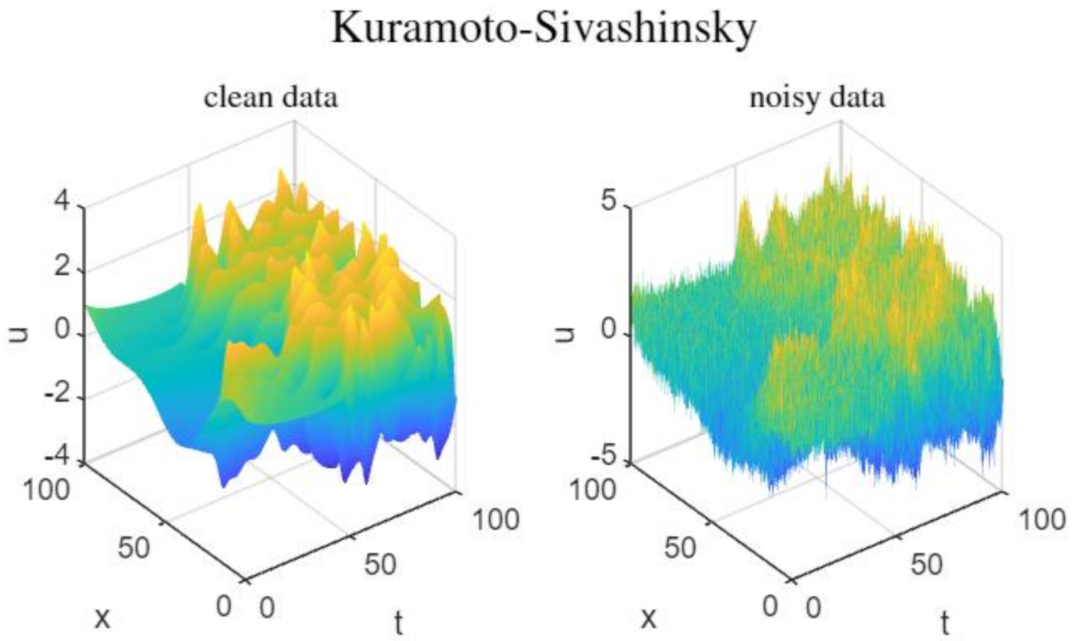
Weak form SINDy

Workshop tutorial

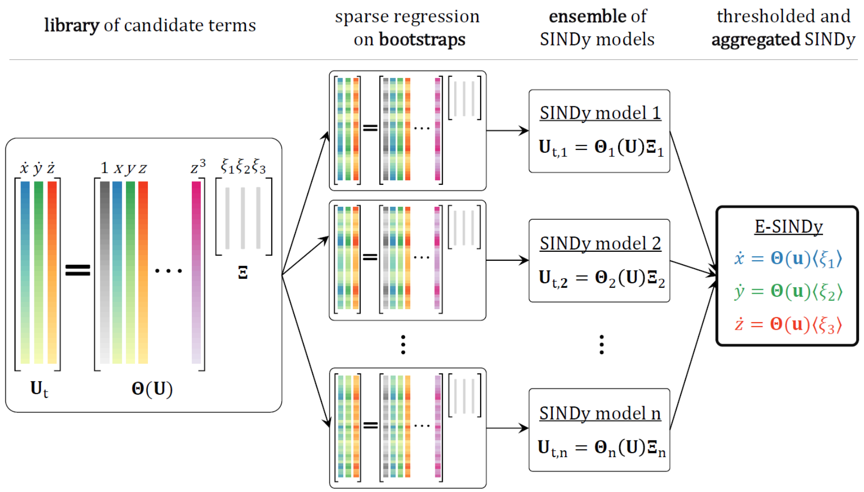
- https://github.com/urban-fasel/I-X_workshop_2025

MATLAB example: https://github.com/urban-fasel/FiltonWorkshop2024 → *Tutorial 7*

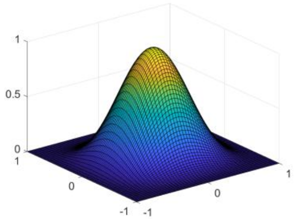
- **Kuramoto-Sivashinsky equation**: $u_t = -uu_x - u_{xx} - u_{xxxx}$

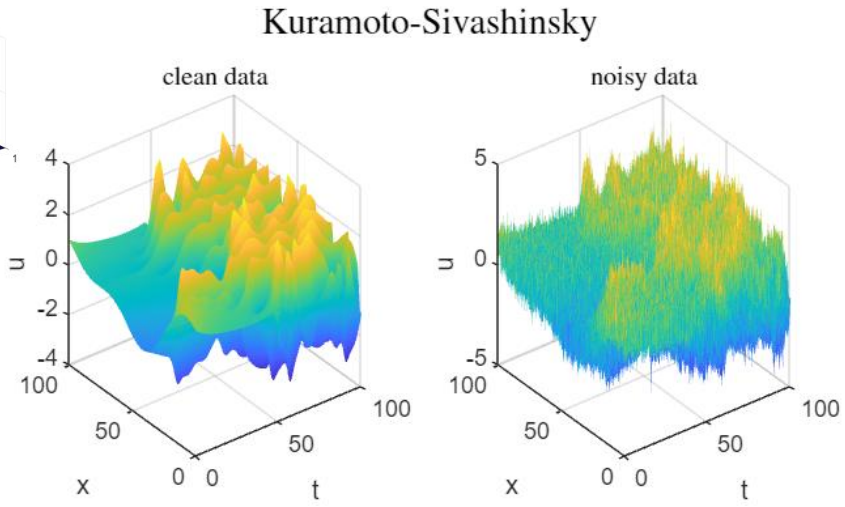


Part 4 summary



- **Ensemble SINDy**
  - Ensembling for noise robust model identification
  - Active learning
  - UQ-SINDy





- **Weak-SINDy**
  - Integral and weak form SINDy methods
  - Weak form parameter estimation
  - Weak SINDy KS example

- **Additional examples and tutorials**
  - E-SINDy: https://github.com/dynamicslab/pysindy/blob/master/examples/13_ensembling.ipynb
  - Weak SINDy: https://github.com/dynamicslab/pysindy/blob/master/examples/12_weakform_SINDy_examples.ipynb

Workshop outline

Part 1: SINDy – Sparse Identification of Nonlinear Dynamics

- **Intro**: identifying ODEs
- Python **example**: *Lorenz system*
- **PySINDy**
- SINDy **challenges and limitations**

"Vanilla" SINDy

SINDy extensions

Part 2: Library (ODEs, PDEs, rational functions)

Part 3: Model selection

Part 4: Noise robustness: weak form & ensemble SINDy