SINDy
A whirlwind tour of its ecosystem

Jean-Christophe Loiseau

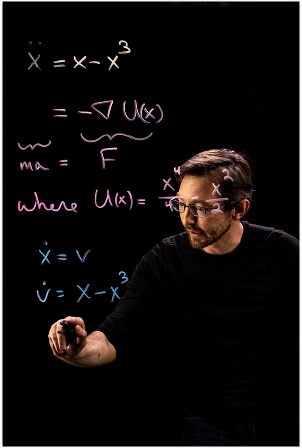
Arts & Métiers

jean-christophe.loiseau@ensam.eu

loiseaujc.github.io

github.com/loiseaujc

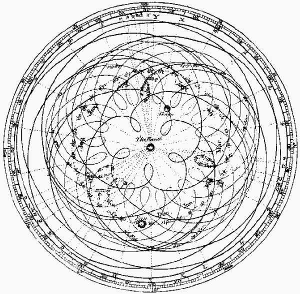
What is SINDy?



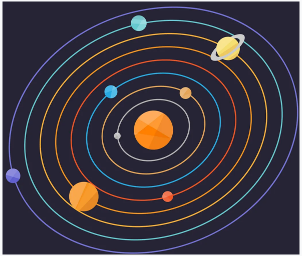
Steven Brunton (UW, USA)

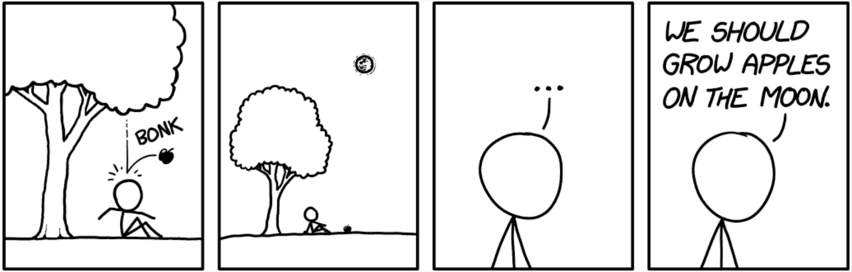
- Stands for *Sparse Identification of Nonlinear Dynamics*.
- First paper published in 2015 in PNAS.
- Framework for identifying equations from data leveraging *sparse regression* algorithms.
- Developed into a complete ecosystem since the seminal work of Steve.

Observation – Using suitable coordinates, many systems in the physical sciences are described by a set of **sparse** equations.



Observation – Using suitable coordinates, many systems in the physical sciences are described by a set of **sparse** equations.




BONK
...
WE SHOULD GROW APPLES ON THE MOON.

- Vanilla SINDy
- Constrained SINDy
- Weak SINDy
- Ensemble SINDy
- SINDy for control
- SINDy-MPC
- SINDy-PI
- MANDy
- Langevin Regression
- Bayesian SINDy
- CINDy
- . . .

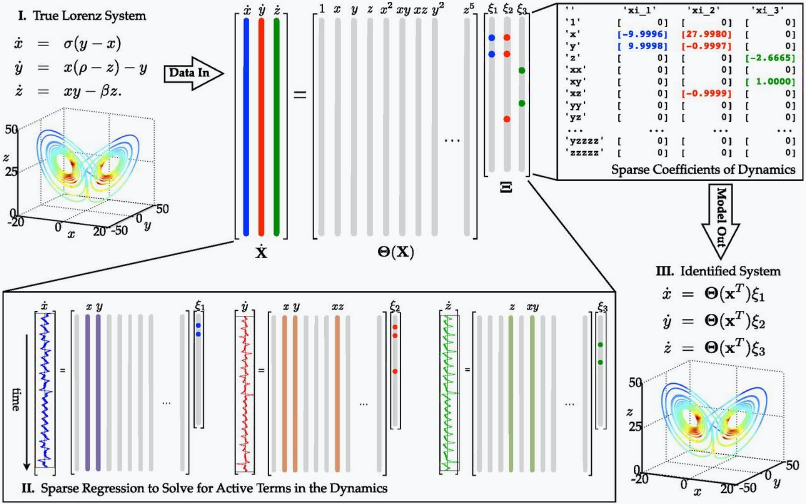
README License

PySINDy

CI failing docs failing pypi package 1.7.5 codecov 95% JOSS 10.21105/joss.02104 JOSS 10.21105/joss.03994 DOI 10.5281/zenodo.7808834

PySINDy is a sparse regression package with several implementations for the Sparse Identification of Nonlinear Dynamical systems (SINDy) method introduced in Brunton et al. (2016a), including the unified optimization approach of Champion et al. (2019), SINDy with control from Brunton et al. (2016b), Trapping SINDy from Kaptanoglu et al. (2021), SINDy-PI from Kaheman et al. (2020), PDE-FIND from Rudy et al. (2017), and so on. A comprehensive literature review is given in de Silva et al. (2020) and Kaptanoglu, de Silva et al. (2021).

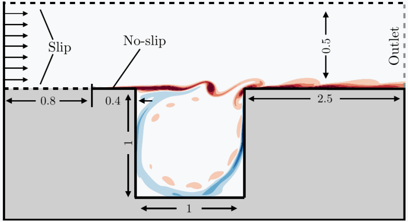
SINDy for Ordinary Diff. Eq.



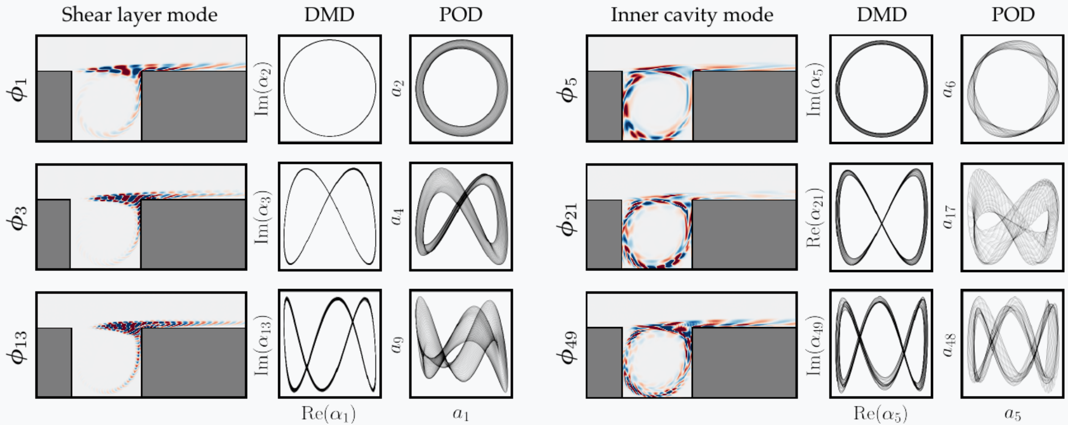
$$
\begin{aligned}
\underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \|\boldsymbol{\alpha}\|_0 \\
\text{subject to} \; & \int_0^T (\dot{\mathbf{x}} - f(\mathbf{x}))\, dt = \mathbf{0} \\
& f(\mathbf{x}) - \sum_{i=1}^{n} \vartheta_i(\mathbf{x}) \alpha_i = \mathbf{0}, \\
& h(\mathbf{x}, \boldsymbol{\alpha}) = \mathbf{0} \\
& g(\mathbf{x}, \boldsymbol{\alpha}) \preccurlyeq \mathbf{0}.
\end{aligned}
$$

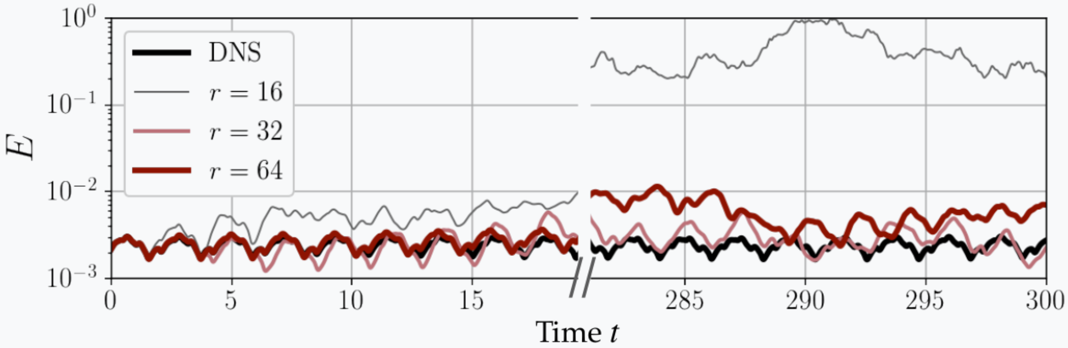
$$
\begin{aligned}
\underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \|\boldsymbol{\alpha}\|_1 \\
\text{subject to} \quad & \left\|\dot{\mathbf{X}} - \boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\alpha}\right\|_F^2 \leqslant \varepsilon \\
& h(\mathbf{x}, \boldsymbol{\alpha}) = \mathbf{0} \\
& g(\mathbf{x}, \boldsymbol{\alpha}) \preccurlyeq \mathbf{0}.
\end{aligned}
$$

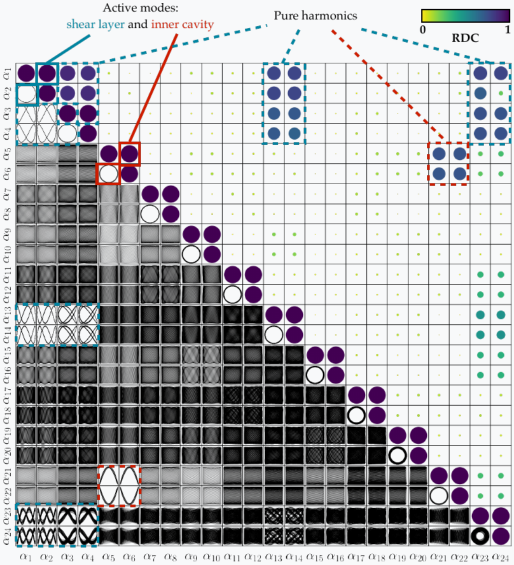
Identifying normal forms



- Classical example in flow control.
- At $\mathrm{Re} = 7500$, the dynamics are *quasiperiodic*.
- 64 POD modes required to capture 99% of the fluctuations.


Shear layer mode
DMD
POD
Inner cavity mode
DMD
POD
ϕ_1
$\mathrm{Im}(\alpha_2)$
a_2
ϕ_5
$\mathrm{Im}(\alpha_5)$
a_6
ϕ_3
$\mathrm{Im}(\alpha_3)$
a_4
ϕ_{21}
$\mathrm{Re}(\alpha_{21})$
a_{17}
ϕ_{13}
$\mathrm{Im}(\alpha_{13})$
a_9
ϕ_{49}
$\mathrm{Im}(\alpha_{49})$
a_{48}
$\mathrm{Re}(\alpha_1)$
a_1
$\mathrm{Re}(\alpha_5)$
a_5


DNS
$r = 16$
$r = 32$
$r = 64$
10^0
10^{-1}
10^{-2}
10^{-3}
E
0
5
10
15
285
290
295
300
Time t

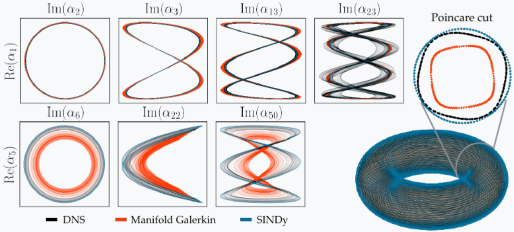


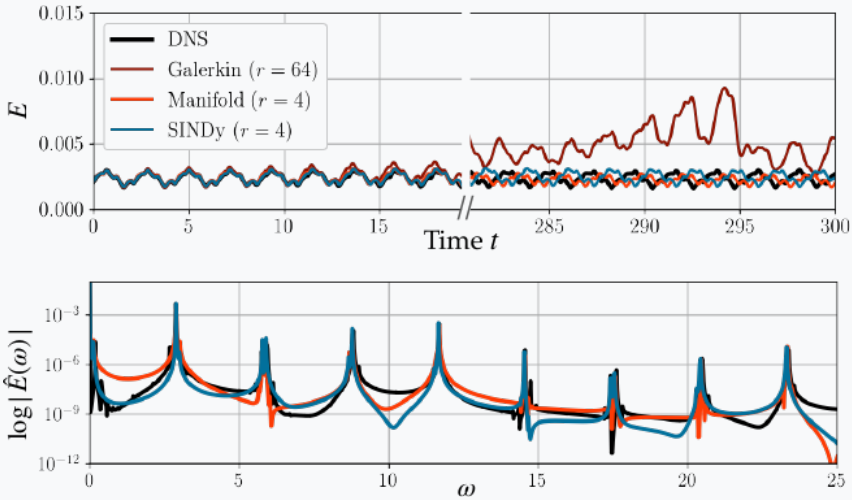
- Many modes are artifacts coming from representing a low-dimensional manifold in a large Euclidean space.
- Need to find a way to break the *Kolmogorov n-width*.

- Using only the four active degrees of freedom, SINDy identifies

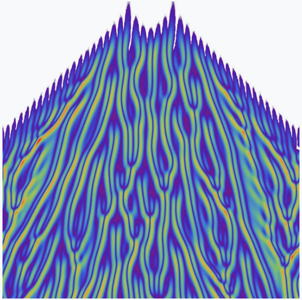
$$
\begin{aligned}
\dot{x} &= \lambda_1 x - \mu_1 |x|^2 x \\
\dot{y} &= \lambda_2 y - \mu_2 |y|^2 y.
\end{aligned}
$$

- ROM consistent with the known bifurcation diagram of the problem.




DNS
Galerkin ($r = 64$)
Manifold ($r = 4$)
SINDy ($r = 4$)
E
Time t
$\log|\hat{E}(\omega)|$
ω

SINDy for Partial Diff. Eq.



- Extending SINDy to PDE is straightforward.
- Massively over-determined constrained least-squares problem.

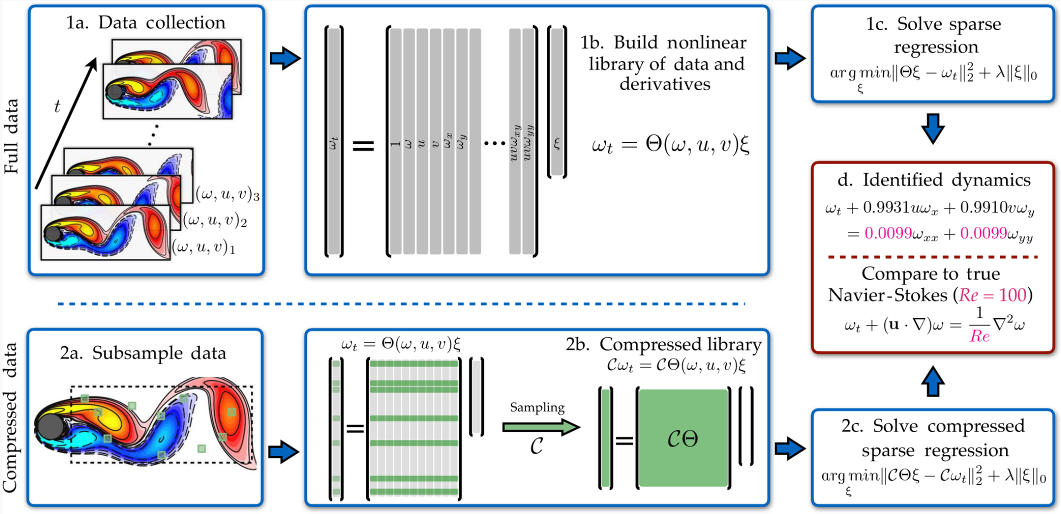
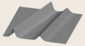
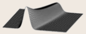
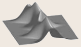
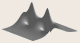
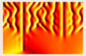
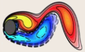

Full data
1a. Data collection
t
$(\omega, u, v)_3$
$(\omega, u, v)_2$
$(\omega, u, v)_1$
ω_t
1 ω u v ω_x ω_y $\cdots$ $uv\omega_{xy}$ $uv\omega_{yy}$
ξ
1b. Build nonlinear library of data and derivatives
$\omega_t = \Theta(\omega, u, v)\xi$
1c. Solve sparse regression
$\arg\min_{\xi} \|\Theta\xi - \omega_t\|_2^2 + \lambda\|\xi\|_0$
d. Identified dynamics
$\omega_t + 0.9931u\omega_x + 0.9910v\omega_y$
$= 0.0099\omega_{xx} + 0.0099\omega_{yy}$
Compare to true Navier-Stokes ($Re = 100$)
$\omega_t + (\mathbf{u} \cdot \nabla)\omega = \frac{1}{Re}\nabla^2\omega$
Compressed data
2a. Subsample data
$\omega_t = \Theta(\omega, u, v)\xi$
2b. Compressed library
$\mathcal{C}\omega_t = \mathcal{C}\Theta(\omega, u, v)\xi$
Sampling
$\mathcal{C}$
$\mathcal{C}\Theta$
2c. Solve compressed sparse regression
$\arg\min_{\xi} \|\mathcal{C}\Theta\xi - \mathcal{C}\omega_t\|_2^2 + \lambda\|\xi\|_0$

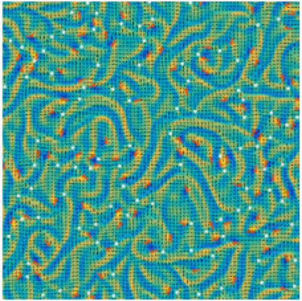
Table 1. Summary of regression results for a wide range of canonical models of mathematical physics. In each example, the correct model structure is identified using PDE-FIND. The spatial and temporal sampling of the numerical simulation data used for the regression is given along with the error produced in the parameters of the model for both no noise and 1% noise. In the reaction-diffusion system, 0.5% noise is used. For Navier-Stokes and reaction-diffusion, the percent of data used in subsampling is also given. NLS, nonlinear Schrödinger; KS, Kuramoto-Sivashinsky.

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	$u_t + uu_x + u_{xx} + u_{xxxx} = 0$	$1.3 \pm 1.3\%, 52 \pm 1.4\%$	$x \in [0, 100], n = 1024, \ t \in [0, 100], m = 251$
u v  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%

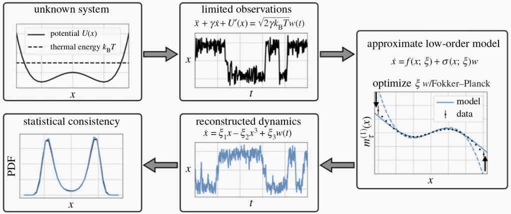
- Physical assumptions of *smoothness*, *locality* and *symmetry* can be used to design an admissible dictionary.

- Currently being used at CEA to infer a PDE from experimental PIV measurements of active turbulence.

- Recent extension of incorporate assumptions of *smoothness*, *locality* and *symmetry* to design *a priori* a physically admissible dictionary.



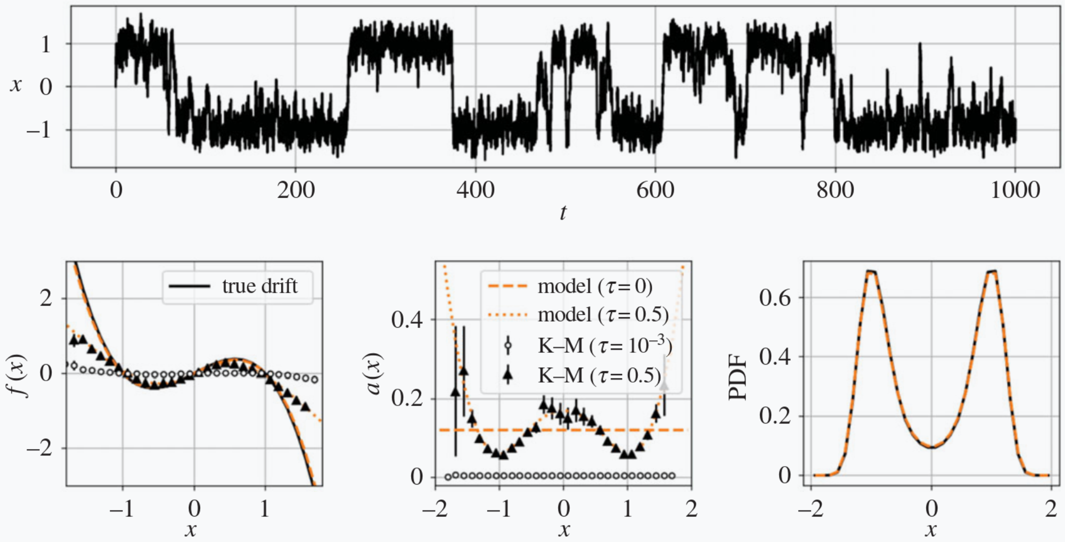
SINDy for Stochastic Diff. Eq.



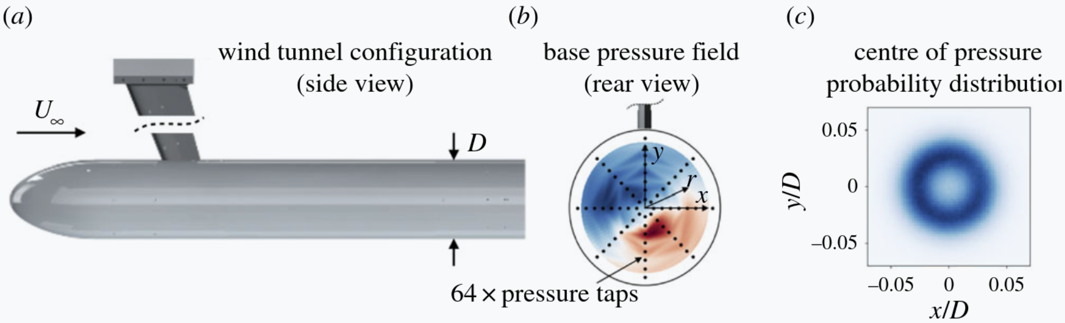
- Many systems in physics can be described by *Langevin equations*

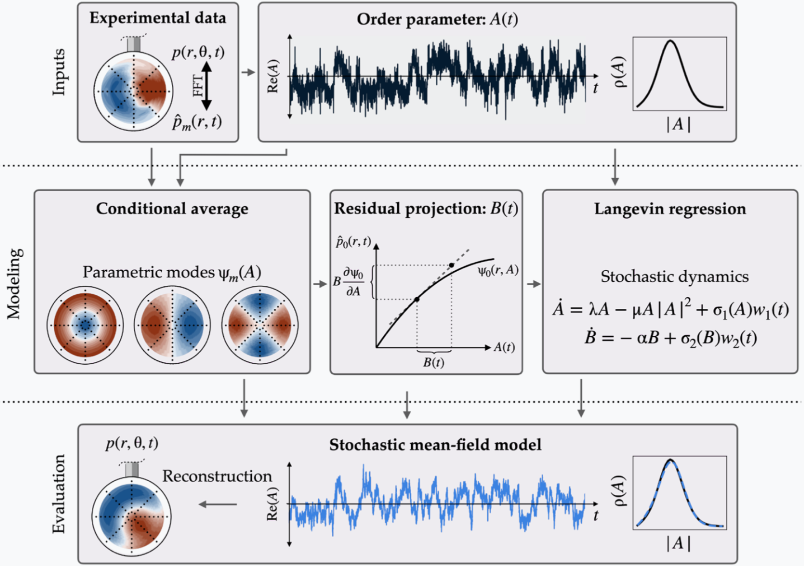
$$\dot{x} = f(x) + \sigma(x)\eta.$$

- Identifying the drift and diffusion terms are is slightly more involved than vanilla SINDy.



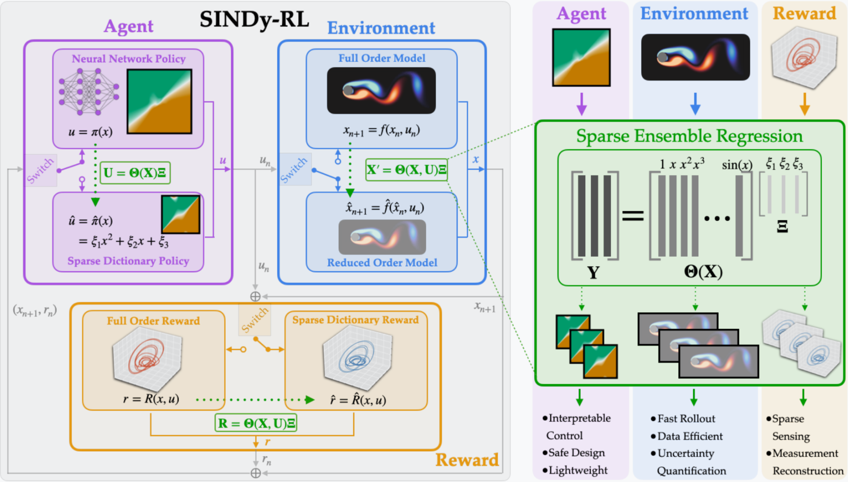
x
t
true drift
f(x)
a(x)
model (τ=0)
model (τ=0.5)
K–M (τ=10^-3)
K–M (τ=0.5)
PDF

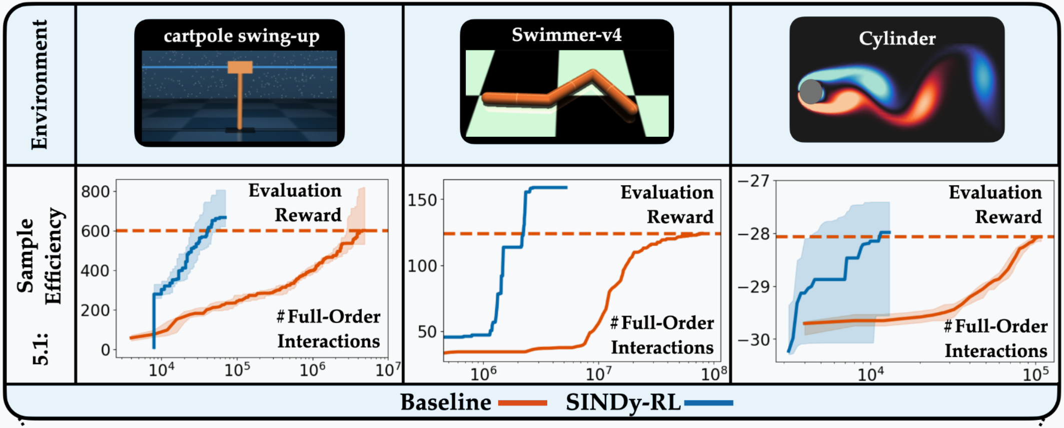

(a)
wind tunnel configuration
(side view)
U_∞
D
(b)
base pressure field
(rear view)
y
r
x
64 × pressure taps
(c)
centre of pressure
probability distributio
0.05
0
–0.05
y/D
–0.05
0
0.05
x/D



Inputs
Experimental data
$p(r,\theta,t)$
FFT
$\hat{p}_m(r,t)$
Order parameter: $A(t)$
Re(A)
t
ρ(A)
|A|
Modeling
Conditional average
Parametric modes $\psi_m(A)$
Residual projection: $B(t)$
$\hat{p}_0(r,t)$
$B\frac{\partial\psi_0}{\partial A}$
$\psi_0(r,A)$
A(t)
B(t)
Langevin regression
Stochastic dynamics
$\dot{A} = \lambda A - \mu A|A|^2 + \sigma_1(A)w_1(t)$
$\dot{B} = -\alpha B + \sigma_2(B)w_2(t)$
Evaluation
$p(r,\theta,t)$
Reconstruction
Stochastic mean-field model
Re(A)
t
ρ(A)
|A|

SINDy for Reinf. Learning




Environment
cartpole swing-up
Swimmer-v4
Cylinder
5.1: Sample Efficiency
Evaluation Reward
Full-Order Interactions
Baseline
SINDy-RL

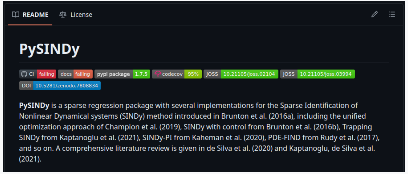
Conclusion

Conclusion

- Since 2015, **SINDy** has evolved into a mature ecosystem.
  - Ordinary diff. eq., partial diff. eq., control systems, etc.

- Despite its versatility, **SINDy** is not a silver bullet.
  - Requires quite a bit of domain expertise.

PySINDy



- Open-source Python package with a simple and `scikit-learn` compatible API.
- We're always on the look for new contributors!
  - More computationally efficient algorithms.
  - New/better variants of **SINDy**.

This presentation wouldn’t have been possible without many collaborators, including (but not limited to):

Steven Brunton, Bing Brunton, Nathan Kutz, Jared Callaham, Kathleen Champion, Brian da Silva, Alan Kaptanoglu, Kadierdan Kaheman, Urban Fasel, Sam Rudy, Zachary Nicolaou, Georgios Rigas, Nicholas Zolman and many others.

Thank you for your attention!

Any questions?

loiseaujc.github.io

github.com/dynamicslab/pysindy