DATA-DRIVEN DISCOVERY OF GOVERNING PHYSICAL LAWS AND COORDINATES FOR ENGINEERING AND PHYSICS

- **Model discovery**
- **Manifolds & Embeddings**
- **Measurement & Sensors**

J. Nathan Kutz

Department of Applied Mathematics
University of Washington
Email: kutz@uw.edu
Web: faculty.washington.edu/kutz

faculty.washingon.edu/kutz

See link to “open source lectures”

Part 0

Applied Optimization
&
Machine Learning

Part 0

Applied Optimization

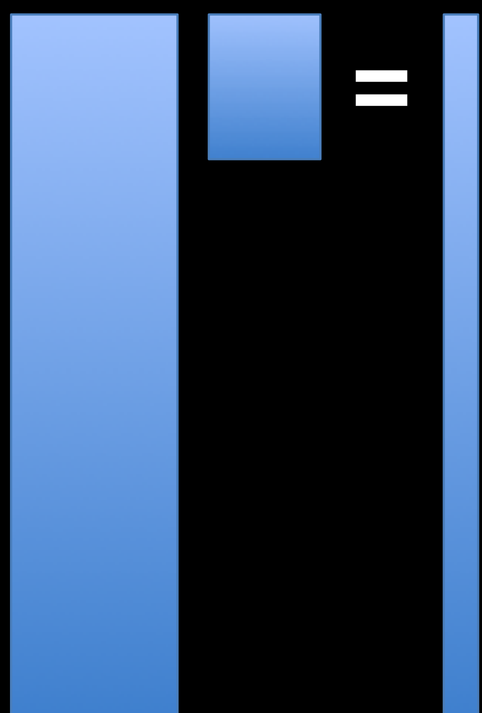
is

Machine Learning

$$Ax=b$$

Data Science Today

Under



Over

- \
- pinv
- Lasso
- Ridge
- Elastic net
- Robust fit

$$Ax=b$$

subject to

$$\min g(x)$$

$$f(A,x)=b$$

subject to

$$\min g(x)$$

Mathematical Framework

Dynamics

State-space

Parameters

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}, t, \Theta, \Omega)$$

Dynamics

Stochastic effects

Measurement

$$\mathbf{y}(t_k) = h(t_k, \mathbf{x}(t_k), \Xi)$$

Measurement model

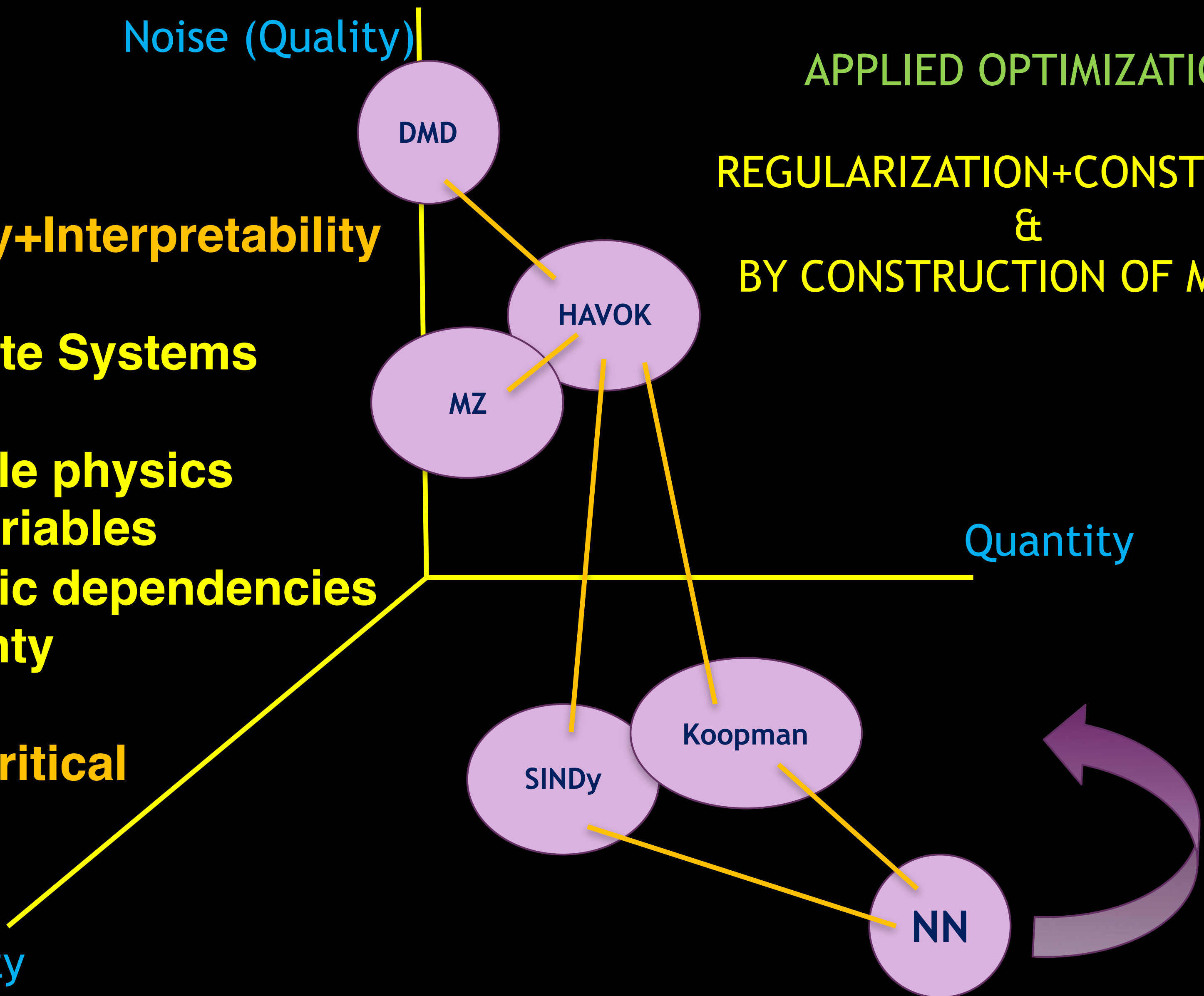
Measurement noise

A Diversity of Strategies

Generalizability+Interpretability

- **Coordinate Systems**
- **Noise**
- **Multi-scale physics**
- **Latent variables**
- **Parametric dependencies**
- **Uncertainty**

Parsimony is critical

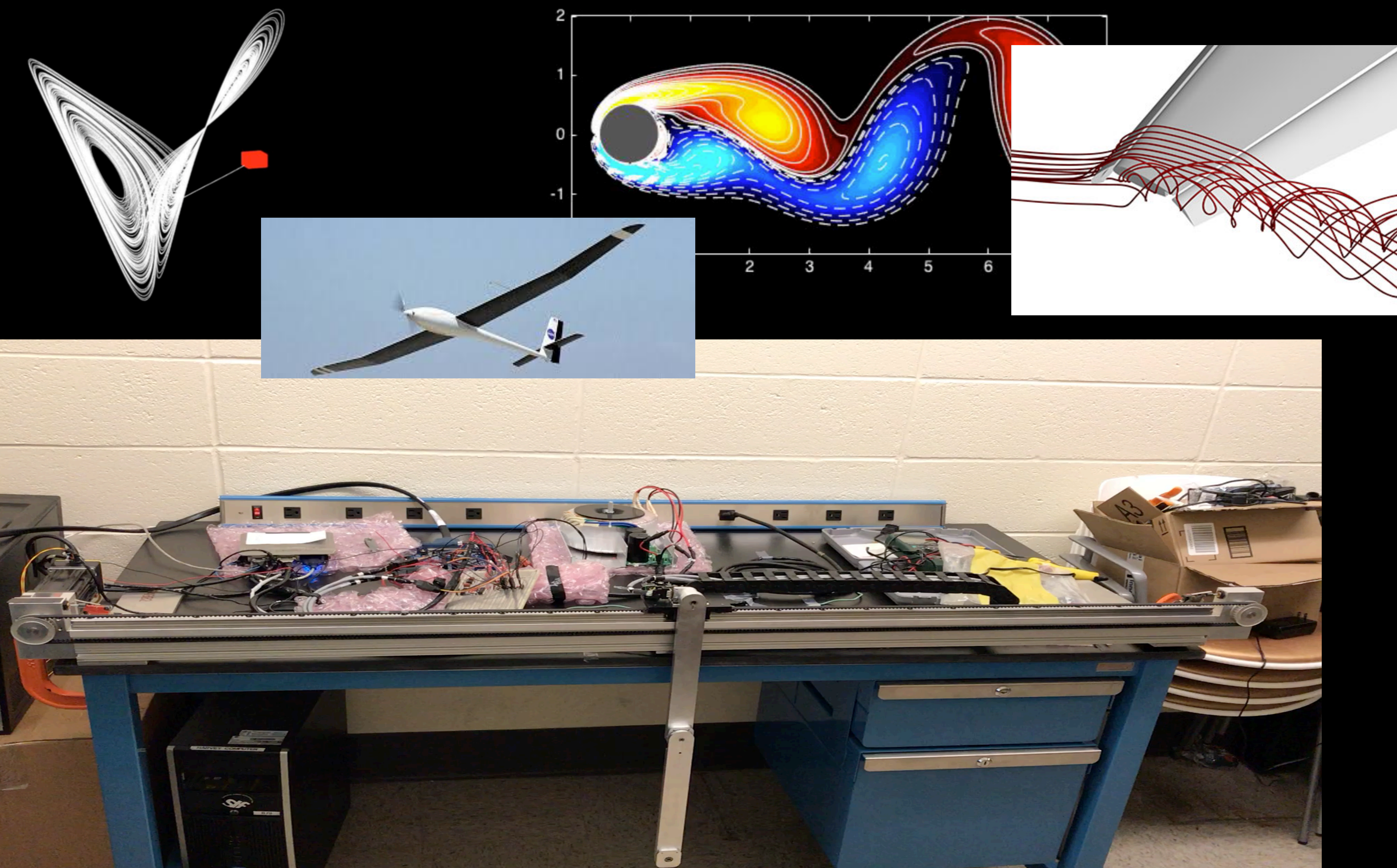


APPLIED OPTIMIZATION

REGULARIZATION+CONSTRAINTS
&
BY CONSTRUCTION OF MODEL

Targeted use of NN

Diverse Models



Part 1

Model Discovery

Steven Brunton
Mechanical Engineering
University of Washington

Joshua Proctor
Institute for Disease Modeling

Governing Dynamical Systems

Generic nonlinear , time-dependent, parametric system

$$\frac{d\mathbf{x}}{dt} = N(\mathbf{x}, t; \mu)$$

Measurements (assimilation)

$$G(\mathbf{x}, t_k) = 0$$

What Could the Right Side Be?

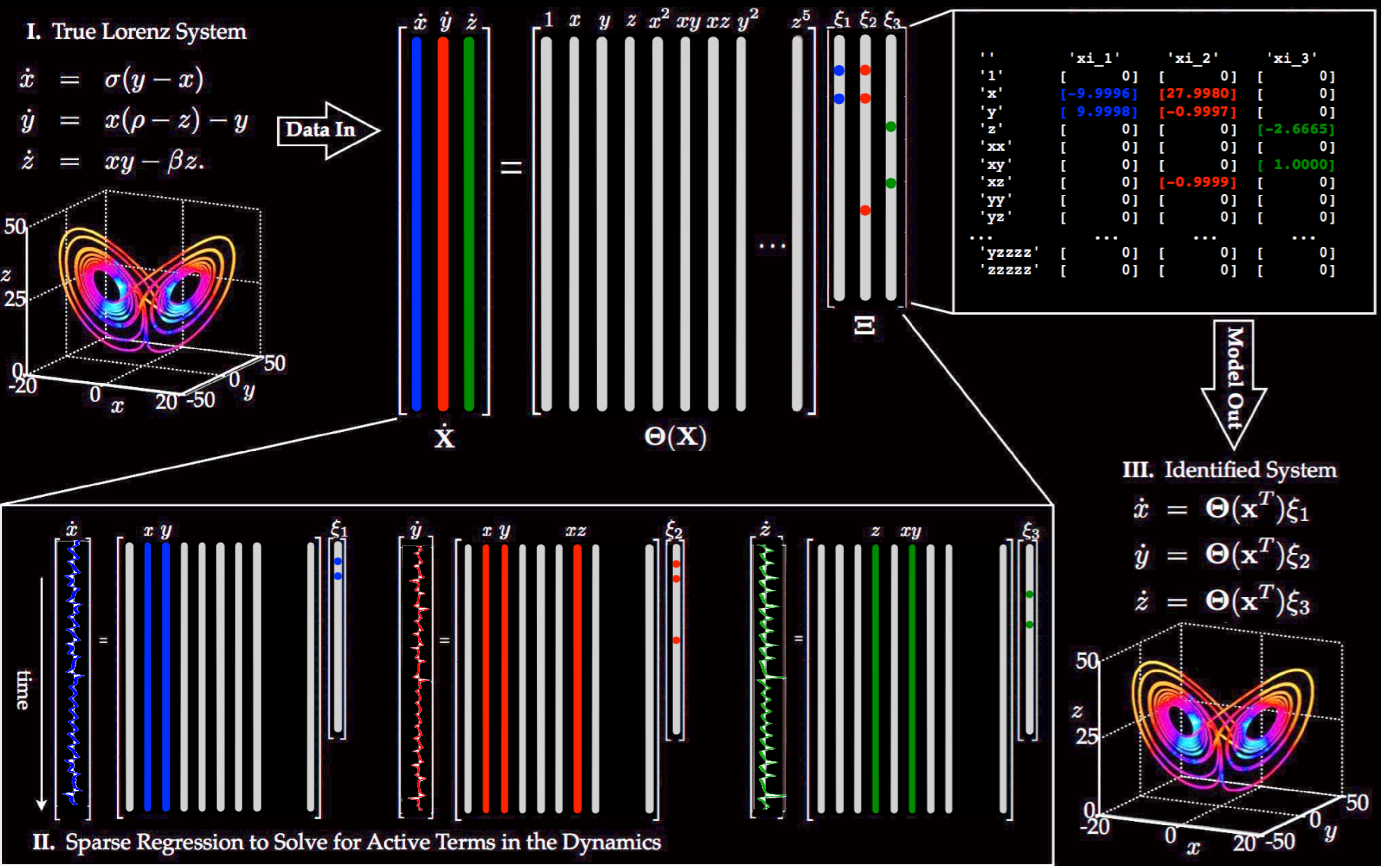
Limited by your imagination

$$\Theta(\mathbf{X}) = \begin{bmatrix} | & | & | & | & & | & | & | & | & \\ 1 & \mathbf{X} & \mathbf{X}^{P_2} & \mathbf{X}^{P_3} & \cdots & \sin(\mathbf{X}) & \cos(\mathbf{X}) & \sin(2\mathbf{X}) & \cos(2\mathbf{X}) & \cdots \\ | & | & | & | & & | & | & | & | & \end{bmatrix}$$

2nd degree polynomials

$$\mathbf{X}^{P_2} = \begin{bmatrix} x_1^2(t_1) & x_1(t_1)x_2(t_1) & \cdots & x_2^2(t_1) & x_2(t_1)x_3(t_1) & \cdots & x_n^2(t_1) \\ x_1^2(t_2) & x_1(t_2)x_2(t_2) & \cdots & x_2^2(t_2) & x_2(t_2)x_3(t_2) & \cdots & x_n^2(t_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_1^2(t_m) & x_1(t_m)x_2(t_m) & \cdots & x_2^2(t_m) & x_2(t_m)x_3(t_m) & \cdots & x_n^2(t_m) \end{bmatrix}$$

Sparse Identification of Nonlinear Dynamics (SINDy)



Nonlinear Systems ID

1. Collect Data

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^T(t_1) \\ \mathbf{x}^T(t_2) \\ \vdots \\ \mathbf{x}^T(t_m) \end{bmatrix} = \begin{array}{c} \xrightarrow{\text{state}} \\ \begin{bmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \cdots & x_n(t_m) \end{bmatrix} \end{array} \Big\downarrow \text{time}$$

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{x}}^T(t_1) \\ \dot{\mathbf{x}}^T(t_2) \\ \vdots \\ \dot{\mathbf{x}}^T(t_m) \end{bmatrix} = \begin{bmatrix} \dot{x}_1(t_1) & \dot{x}_2(t_1) & \cdots & \dot{x}_n(t_1) \\ \dot{x}_1(t_2) & \dot{x}_2(t_2) & \cdots & \dot{x}_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{x}_1(t_m) & \dot{x}_2(t_m) & \cdots & \dot{x}_n(t_m) \end{bmatrix}.$$

2. Build Library of Candidate Nonlinearities

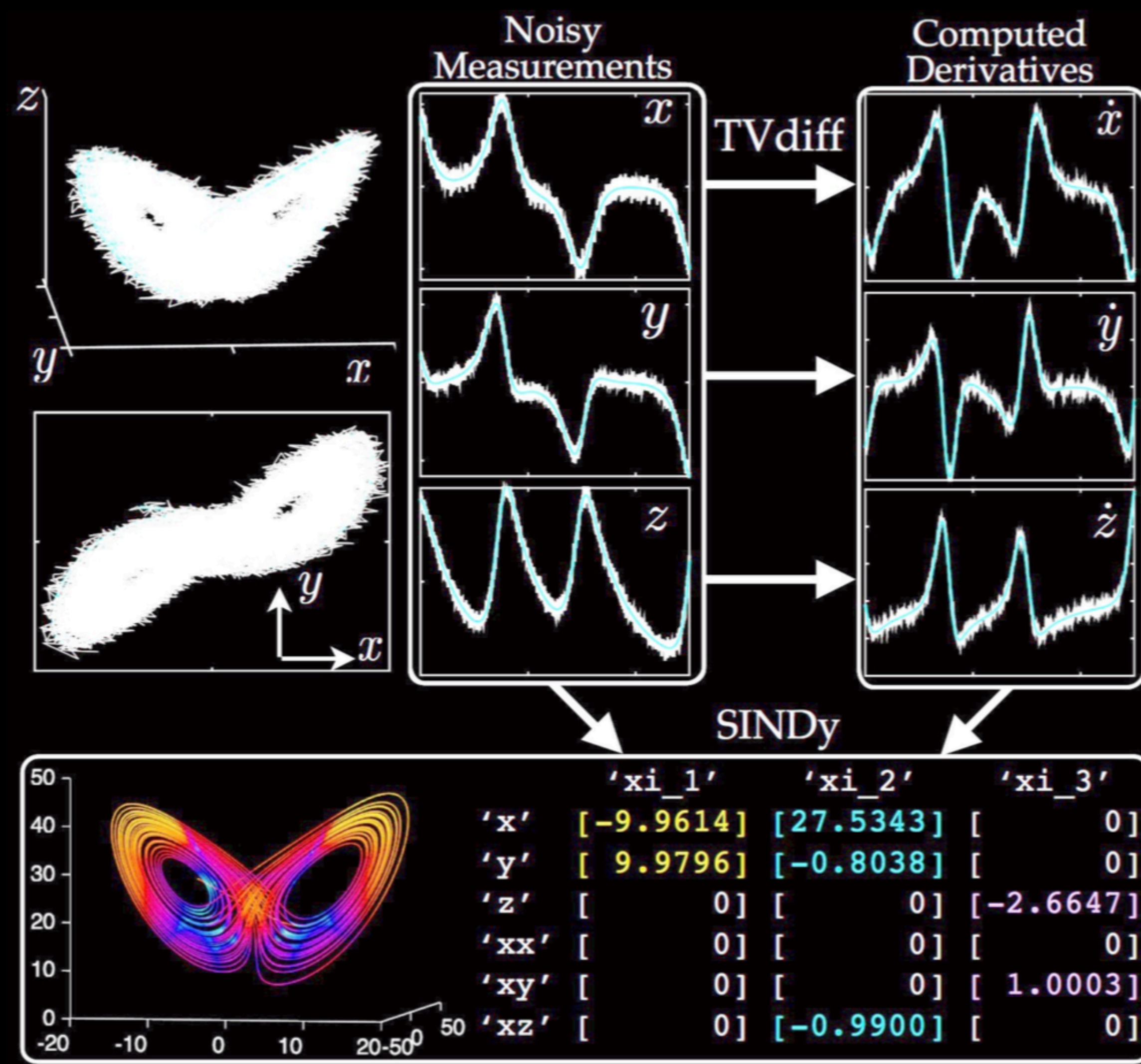
$$\Theta(\mathbf{X}) = \begin{bmatrix} | & | & | & | & & | & | & \\ \mathbf{1} & \mathbf{X} & \mathbf{X}^{P_2} & \mathbf{X}^{P_3} & \cdots & \sin(\mathbf{X}) & \cos(\mathbf{X}) & \cdots \\ | & | & | & | & & | & | & \end{bmatrix}.$$

3. Sparse Regression to Find Active Terms

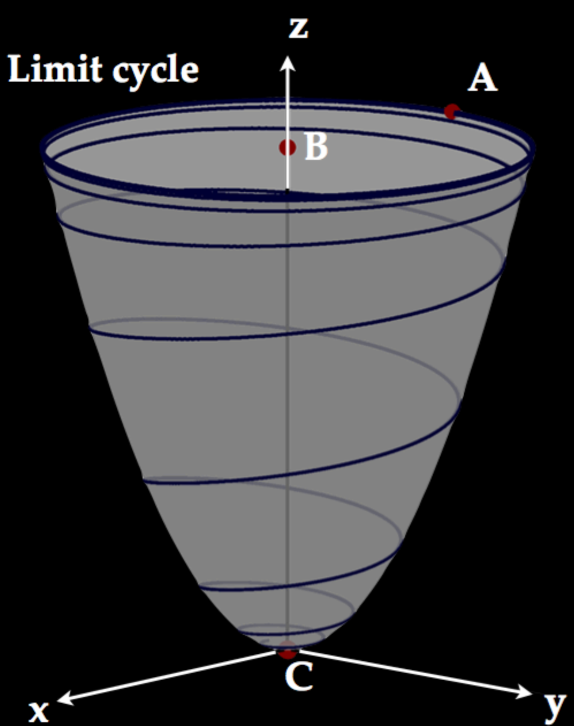
$$\dot{\mathbf{X}} = \Theta(\mathbf{X})\Xi.$$

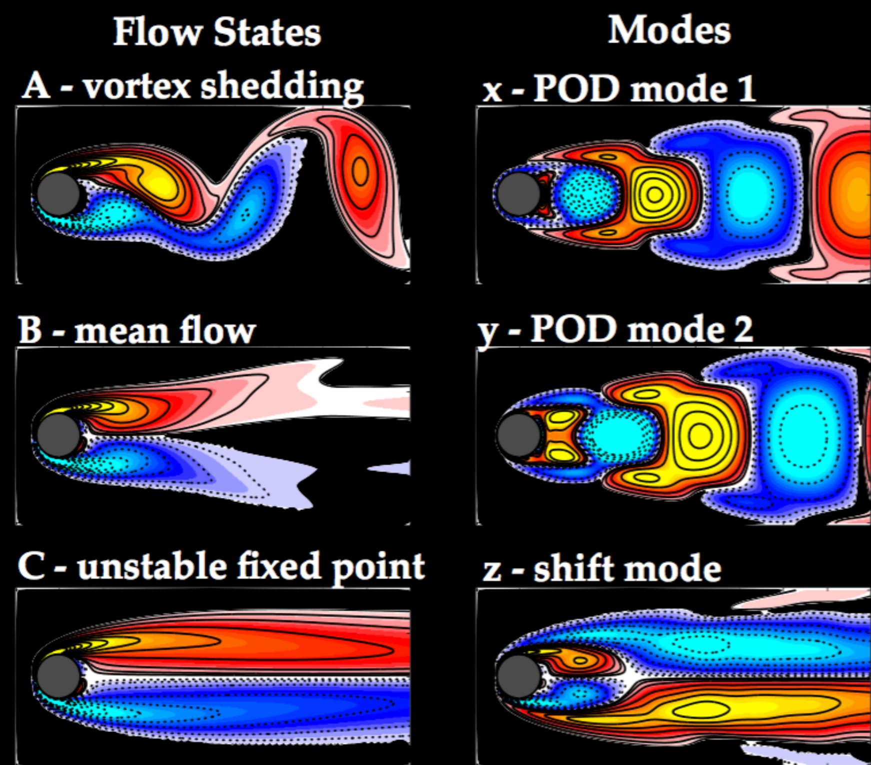
4. Nonlinear Model

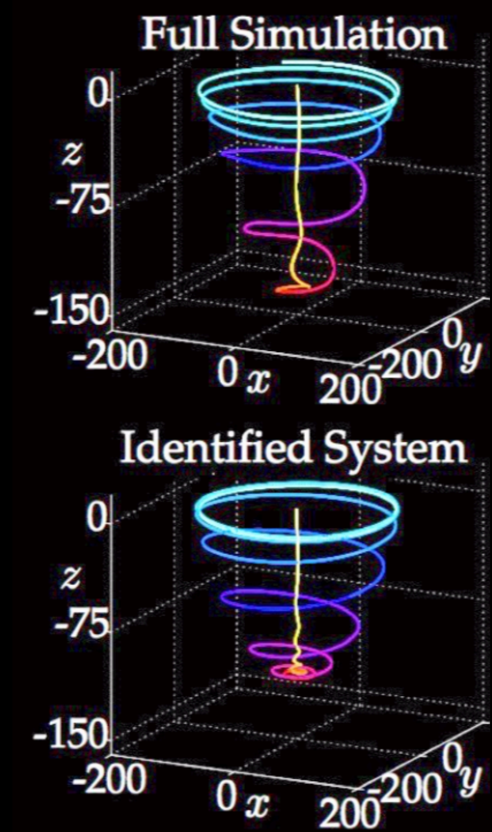
$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = \Xi^T(\Theta(\mathbf{x}^T))^T$$



Identifying Slow Manifolds







30 years of progress

$$
\begin{aligned}
\dot{x} &= \mu x - \omega y + Axz \\
\dot{y} &= \omega x + \mu y + Ayz \\
\dot{z} &= -\lambda(z - x^2 - y^2).
\end{aligned}
$$

1. Hopf bifurcations as path to turbulence
Ruelle & Takens, *Communications in Mathematical Physics*, 1971

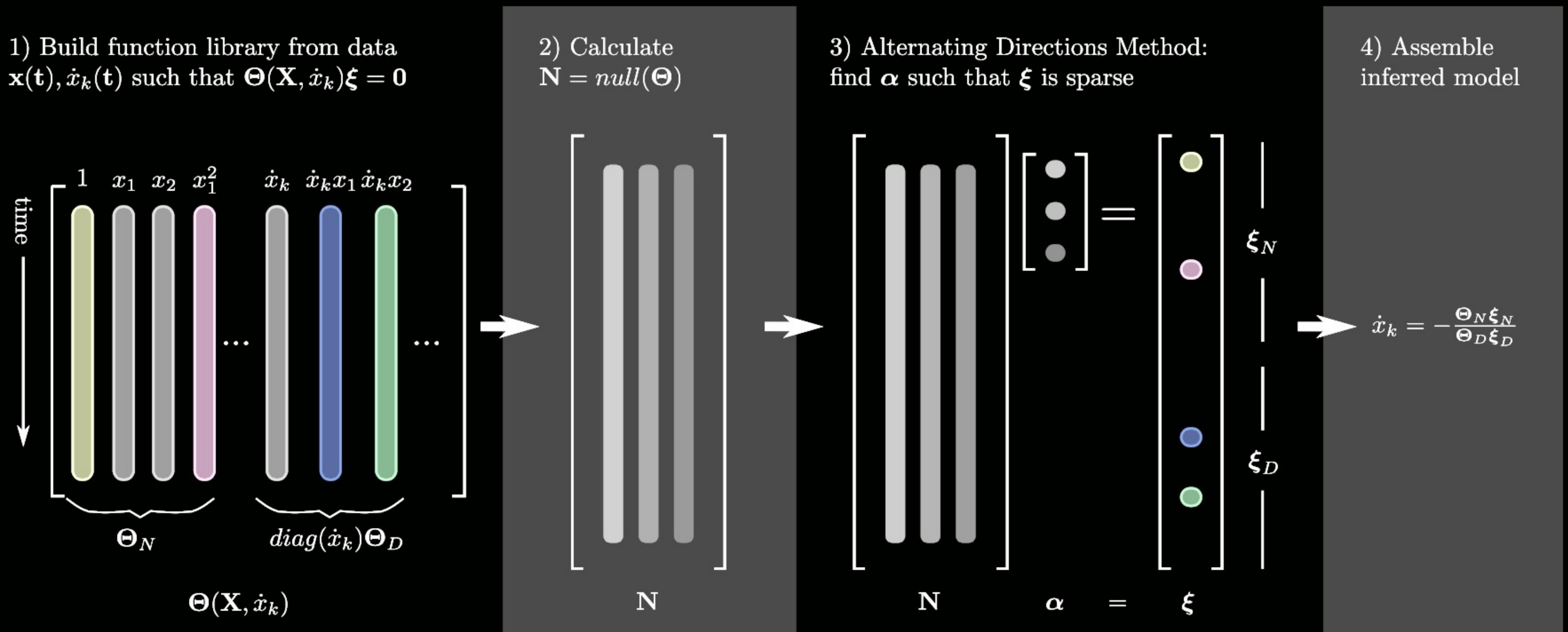
2. Vortex shedding and Hopf bifurcation
Jackson, *Journal of Fluid Mechanics, 1987.*

3. Mean-field model with slow manifold
Noack, Afanasiev, Morzynski, Tadmor, & Thiele,
Journal of Fluid Mechanics, 2003.

Modifications: Implicity-SINDy

Niall Mangan

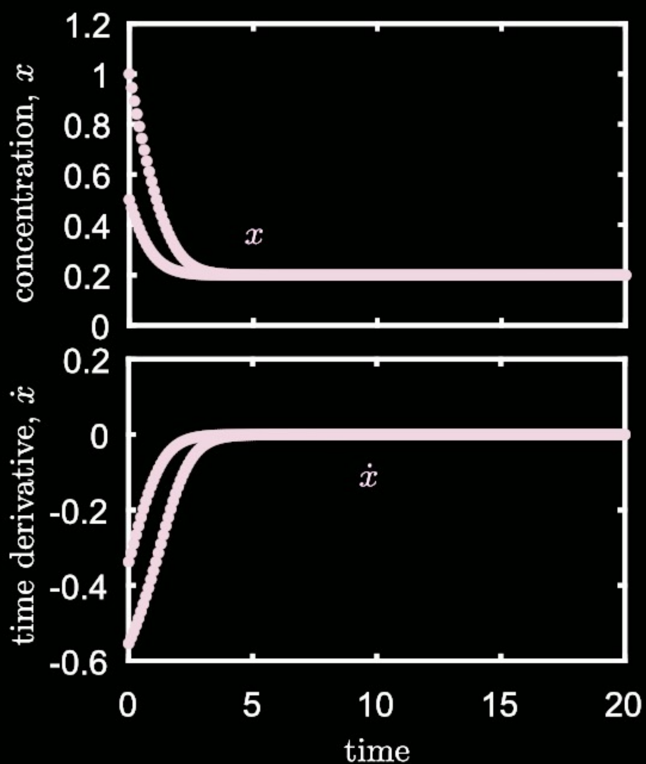
$$\dot{x}_k = \frac{f_N(\mathbf{x})}{f_D(\mathbf{x})}$$



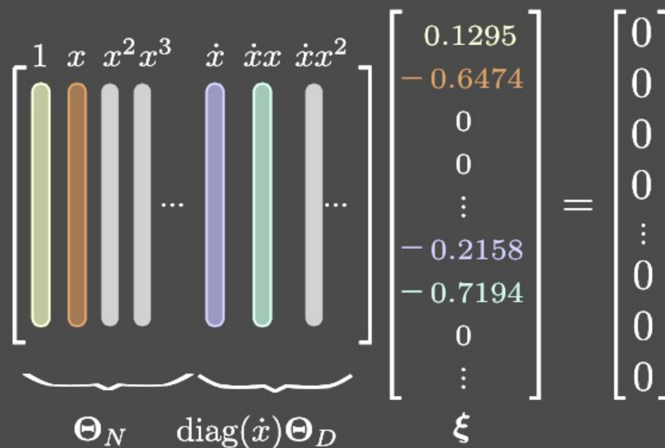
Michaelis-Menten: enzymatic reaction

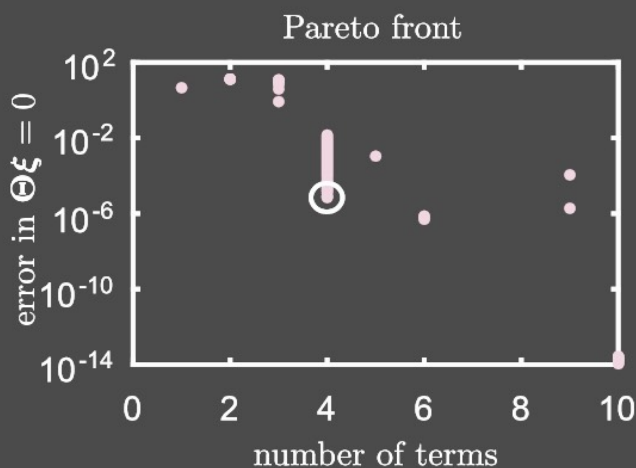
1) Generate test data from system:

$$\dot{x} = 0.6 - \frac{1.5x}{0.3 + x}$$



2) Build functional library. Sparsely select terms and find λ where error drops on Pareto front:



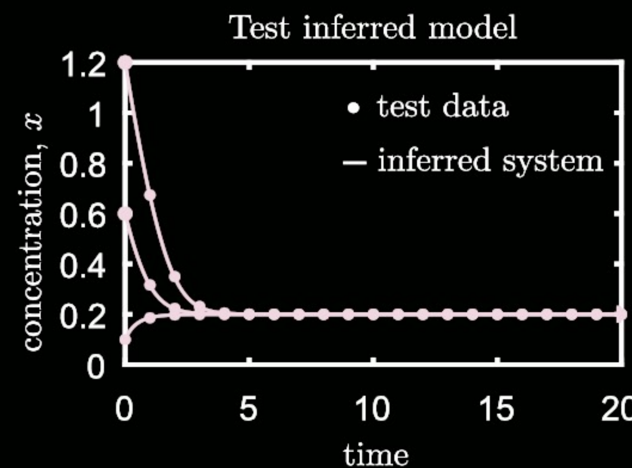


3) Construct inferred model and compare with data from new initial conditions:

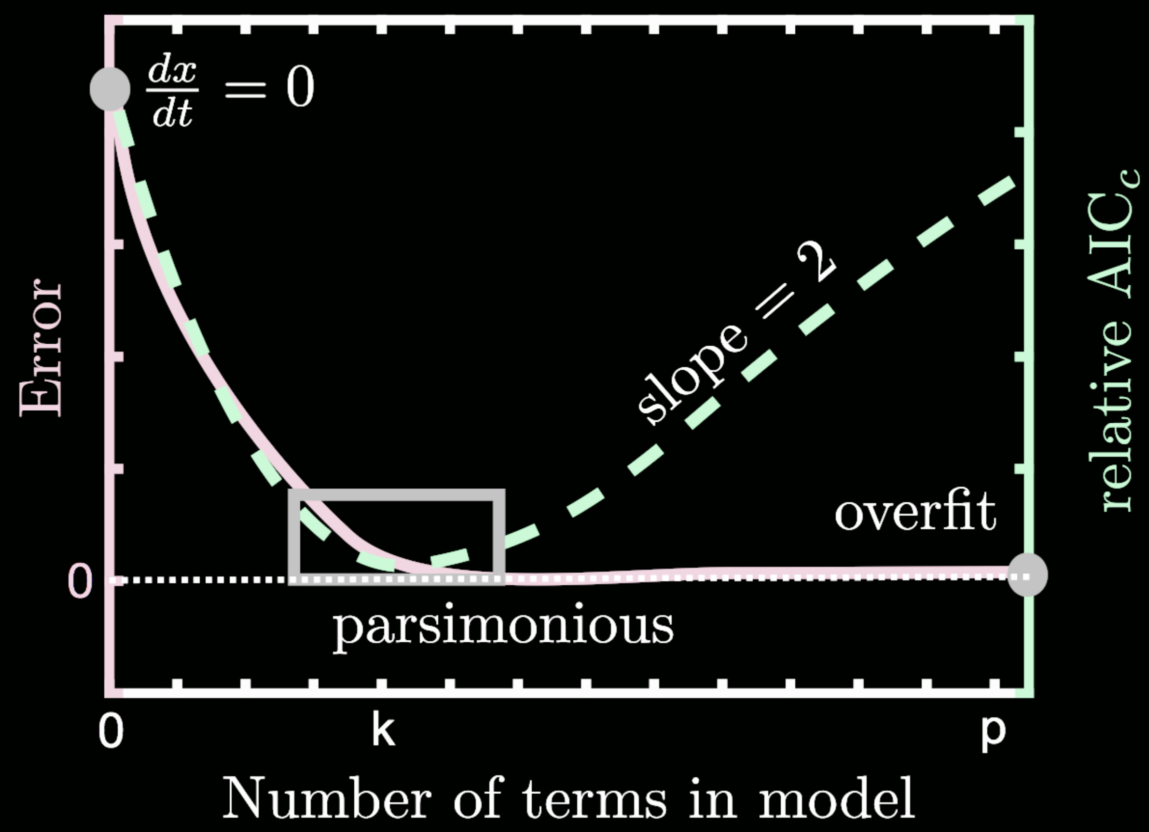
$$\dot{x} = \frac{0.1295 - 0.6474x}{0.2158 + 0.7194x} = \frac{0.6 - 3x}{1 + 3.33x}$$

$$= \frac{0.6(1 + 3.33x)}{1 + 3.33x} - \frac{1.999x + 3x}{1 + 3.33x}$$

$$= 0.6 - \frac{1.5x}{0.3 + x}$$



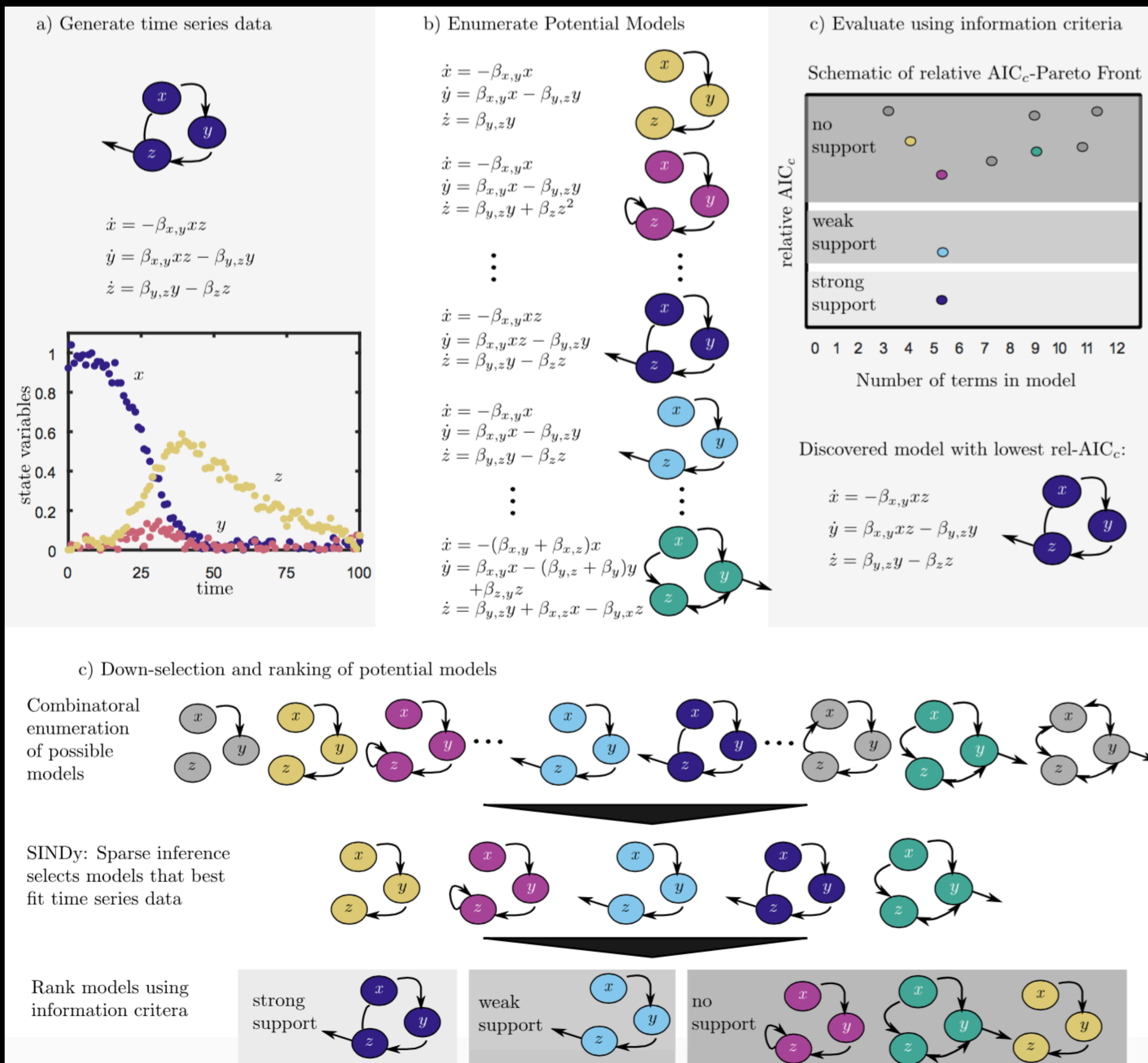
Parsimony



$$AIC_j = 2k - 2\ln(L(\mathbf{x}, \hat{\mu}))$$

$$L(\mathbf{x}, \mu) = P(\mathbf{x}|\mu) \text{ is the likelihood function}$$

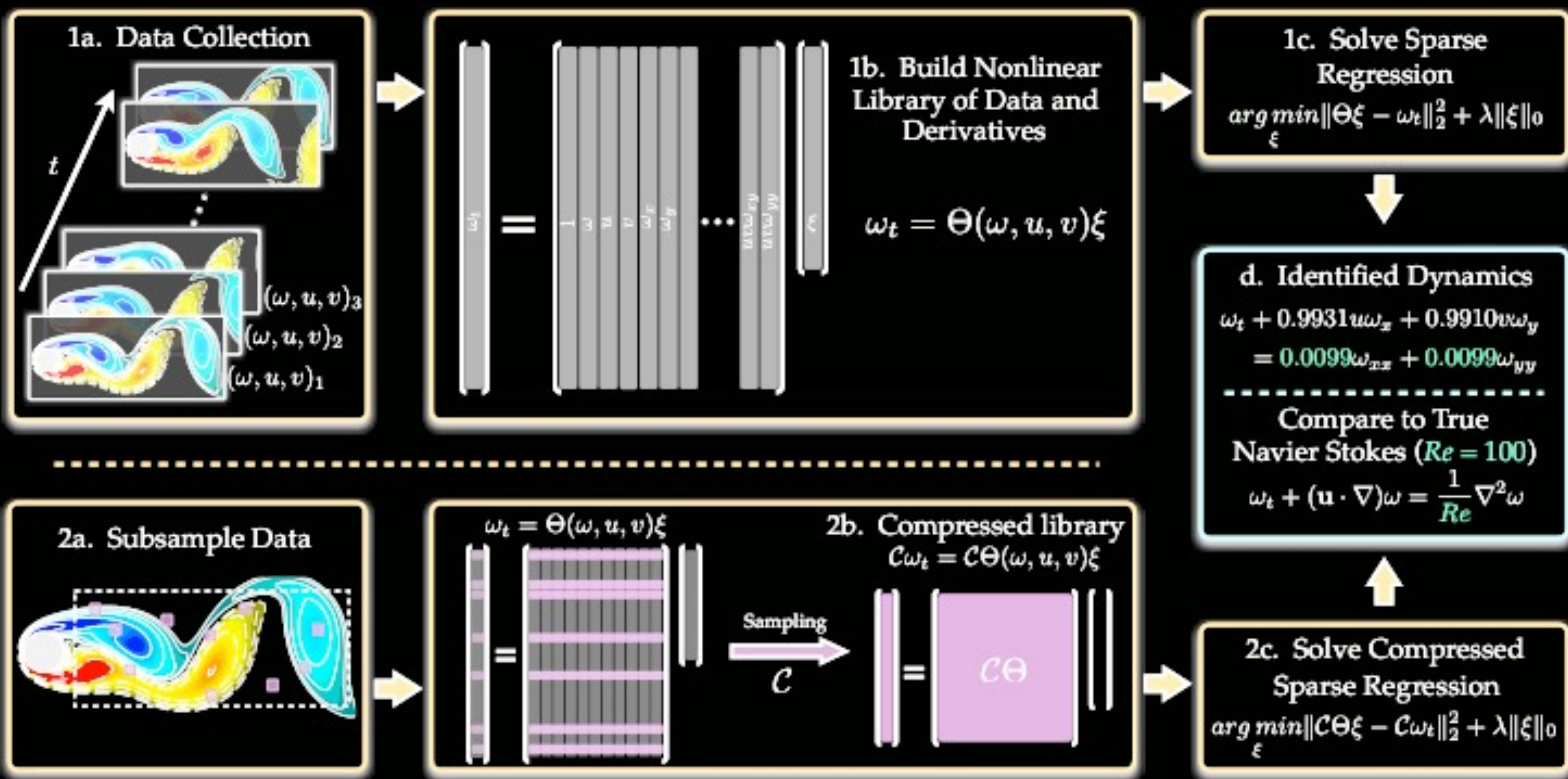
Model Selection and Information Theory



Model Selection

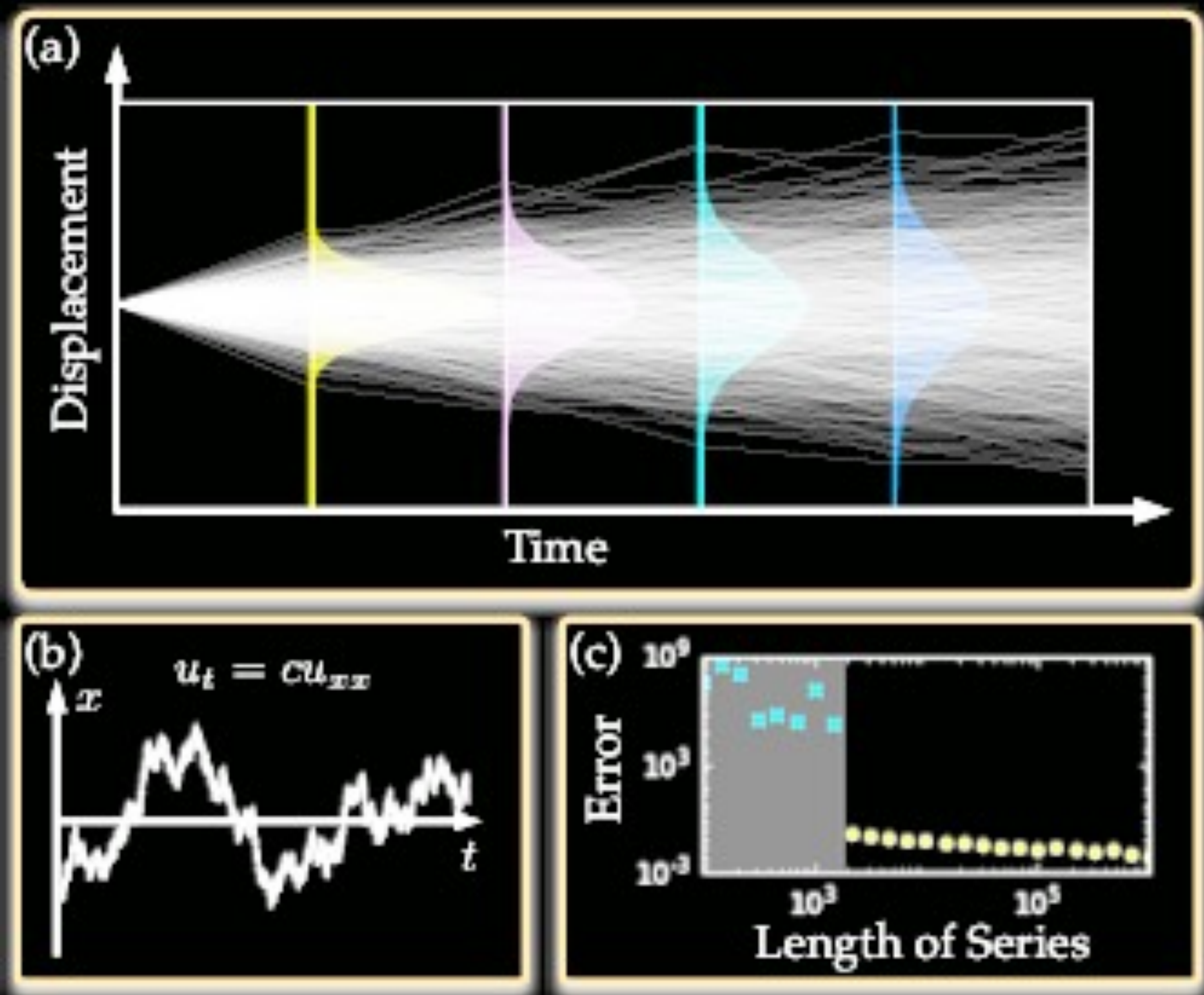
- 1950s KL divergence
- Early 70s AIC (Akaike)
- 78 BIC (G. Schwarz)
- BIC/AIC limited # of models

Discovering PDEs

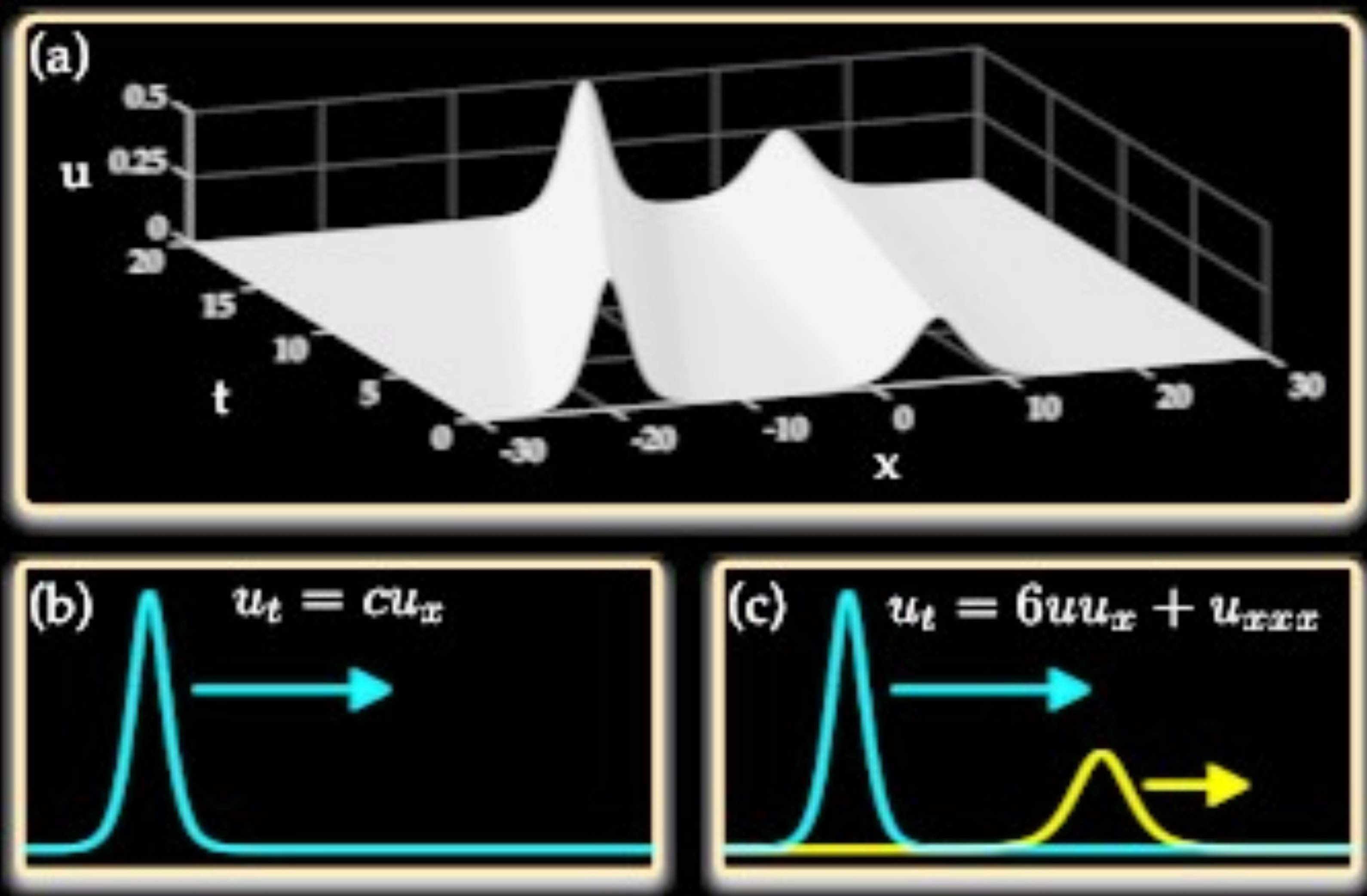


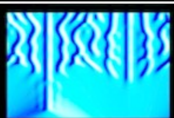
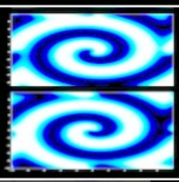
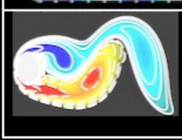
Sam Rudy

Lagrangian Measurements

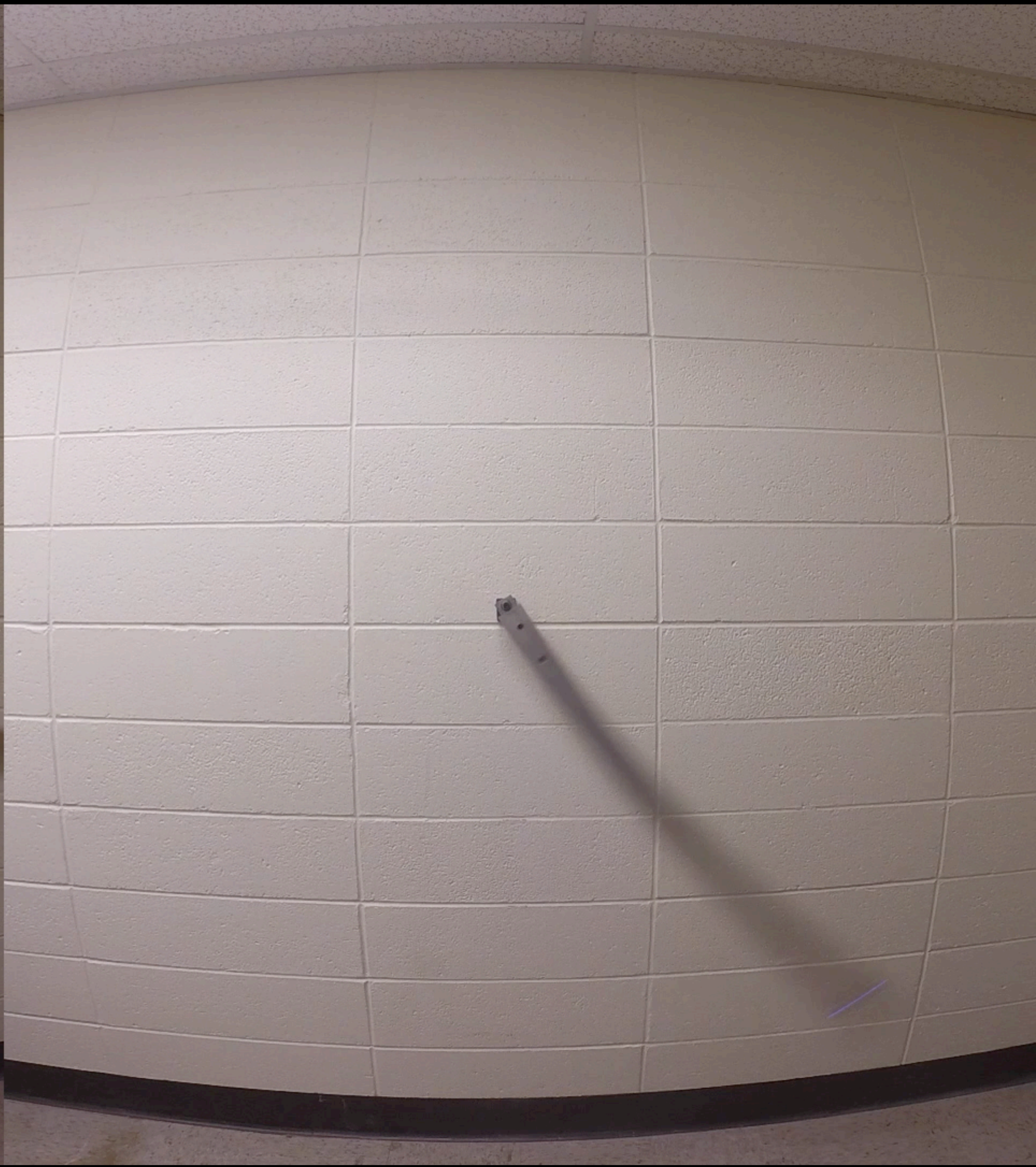


Disambiguation



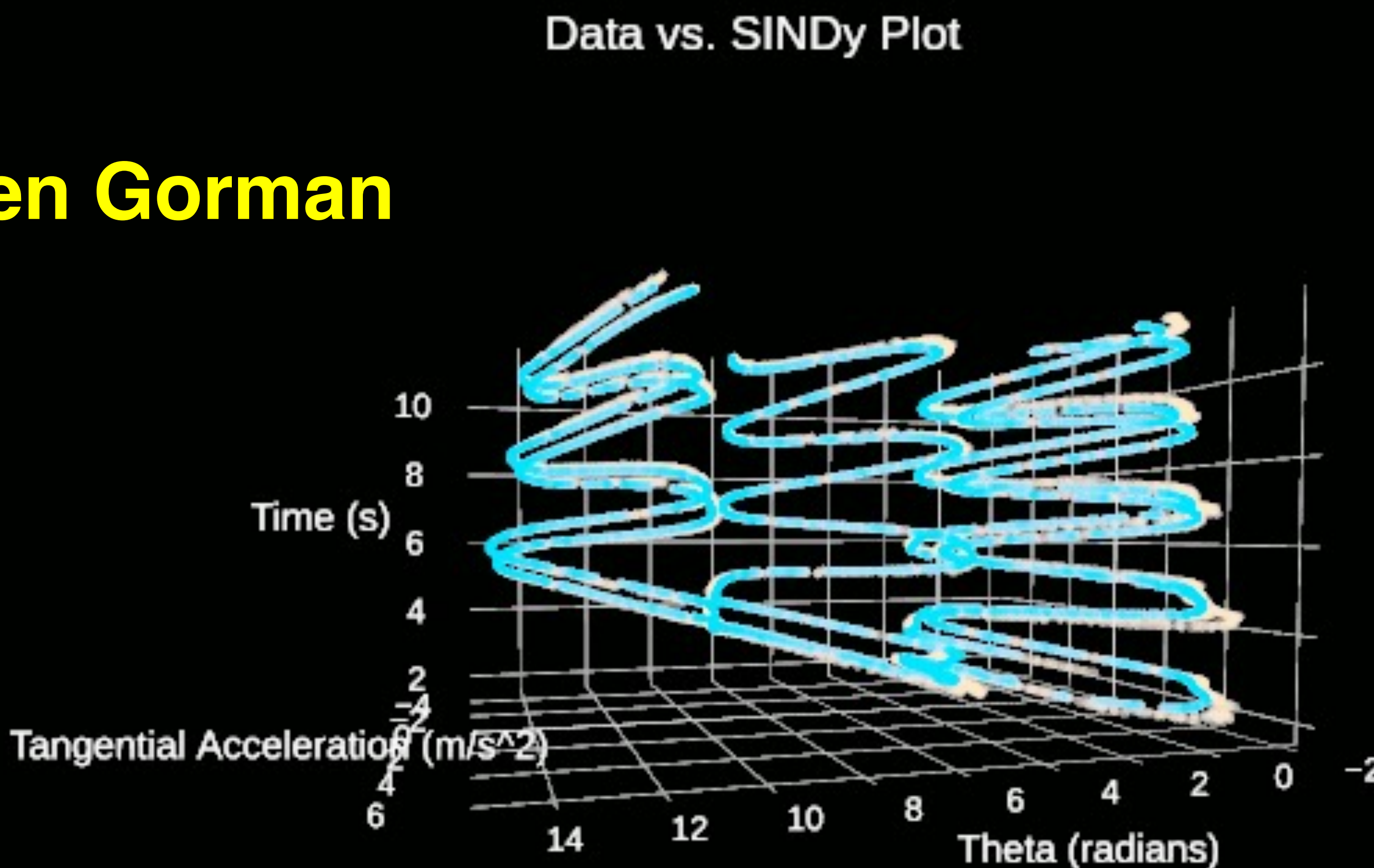
PDE	Form	Error (no noise, noise)	Discretization
 KdV	$u_t + 6uu_x + u_{xxx} = 0$	$1\% \pm 0.2\%, 7\% \pm 5\%$	$x \in [-30, 30], n = 512, \ t \in [0, 20], m = 201$
 Burgers	$u_t + uu_x - \epsilon u_{xx} = 0$	$0.15\% \pm 0.06\%, 0.8\% \pm 0.6\%$	$x \in [-8, 8], n = 256, \ t \in [0, 10], m = 101$
 Schrodinger	$iu_t + \frac{1}{2}u_{xx} - \frac{x^2}{2}u = 0$	$0.25\% \pm 0.01\%, 10\% \pm 7\%$	$x \in [-7.5, 7.5], n = 512, \ t \in [0, 10], m = 401$
 NLS	$iu_t + \frac{1}{2}u_{xx} + \lvert u \rvert^2 u = 0$	$0.05\% \pm 0.01\%, 3\% \pm 1\%$	$x \in [-5, 5], n = 512, \ t \in [0, \pi], m = 501$
 KS	$u_t + uu_x + u_{xx} + u_{xxxx} = 0$	$1.3\% \pm 1.3\%, 70\% \pm 27\%$	$x \in [0, 100], n = 1024, \ t \in [0, 100], m = 251$
 R-D	$u_t = 0.1\nabla^2 u + \lambda(A)u - \omega(A)v$ $v_t = 0.1\nabla^2 v + \omega(A)u + \lambda(A)v$ $A = u^2 + v^2, \omega = -\beta A^2, \lambda = 1 - A^2$	$0.02\% \pm 0.01\%, \ 3.8\% \pm 2.4\%$	$x, y \in [-10, 10], n = 256, \ t \in [0, 10], m = 201$ subsample $3 \cdot 10^5$
 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 $3 \cdot 10^5$

Experiments

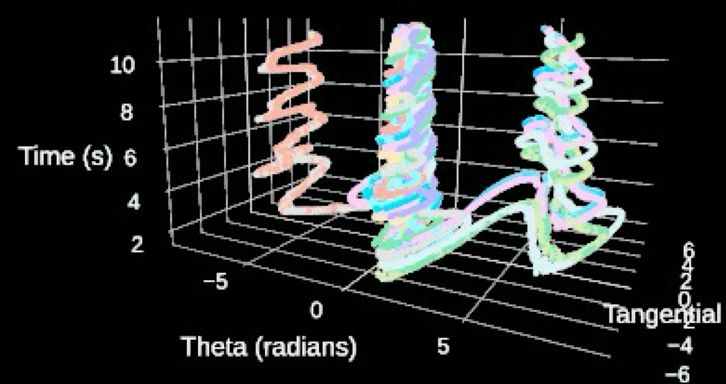


Arduino Magic

Taren Gorman




```
(77854, 2) (77854, 2)
With  -1 jobs, fit and predict STRidge took 5.747981 seconds.
dx_0 / dt =  1.0*x_1
dx_1 / dt =  -0.1460697460858498*x_1+-3.9120253716489075*sin(x_0)
```



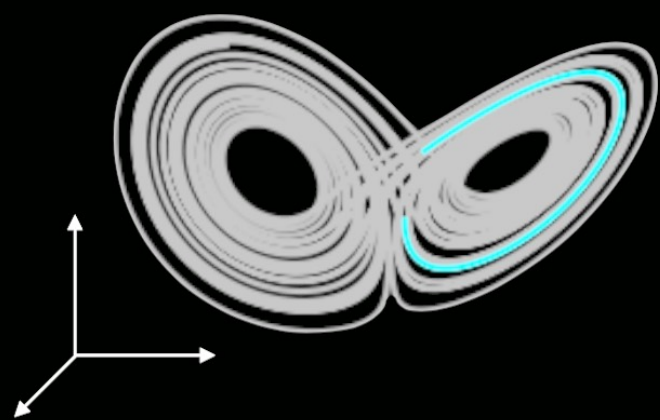
KEY CHALLENGES

- Limited measurements & data
- Noise
- Multi-scale physics
- Latent variables
- Parametric dependencies
- Stochastic systems

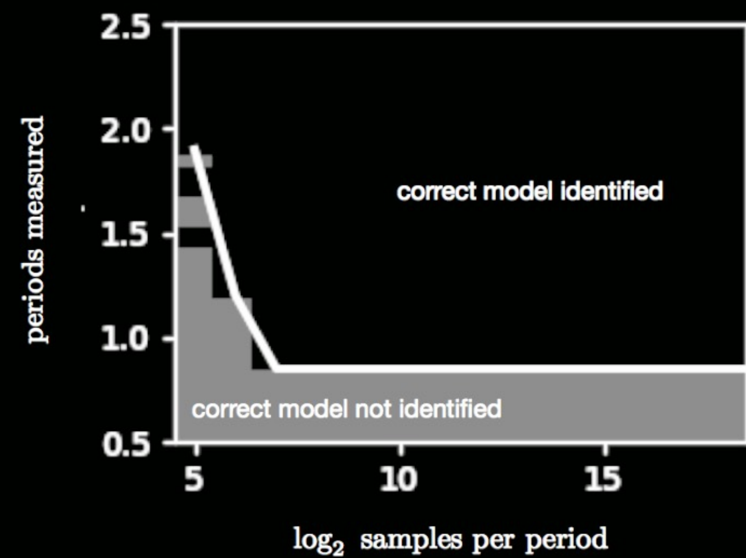
Multiscale Systems

Limits of Model Discovery

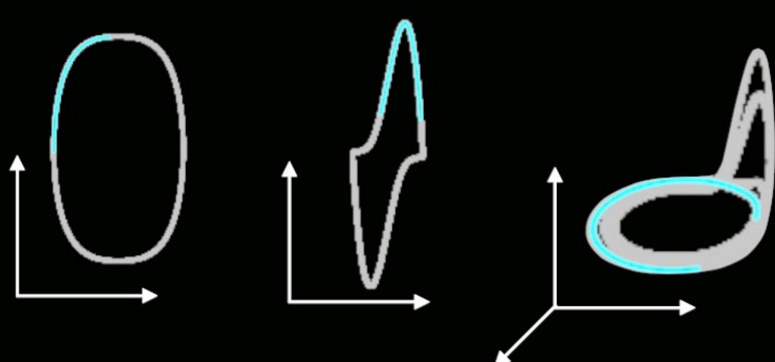
Lorenz system



SINDy sampling requirements



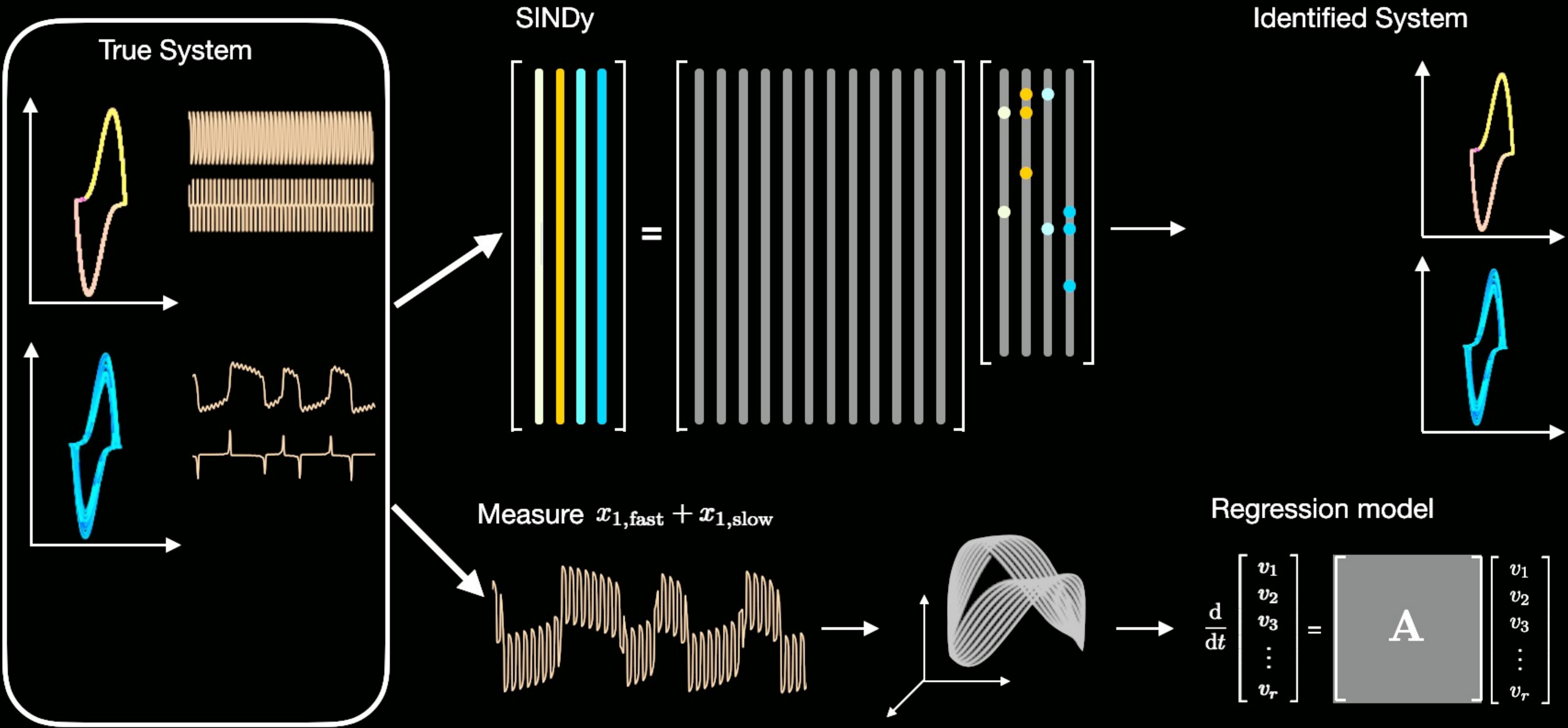
Duffing
Van der Pol
Rossler



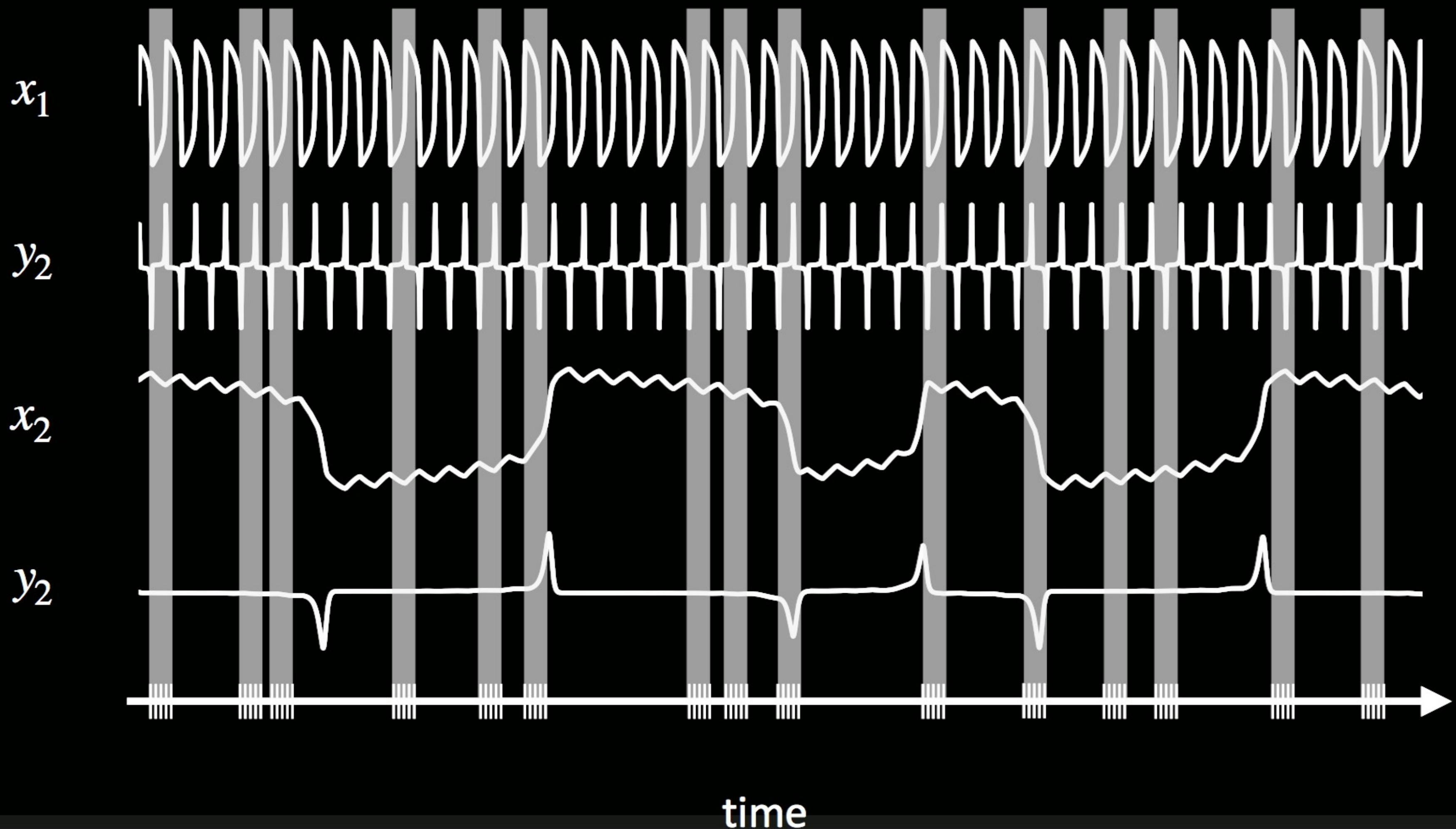
sampling requirements

Kathleen Champion

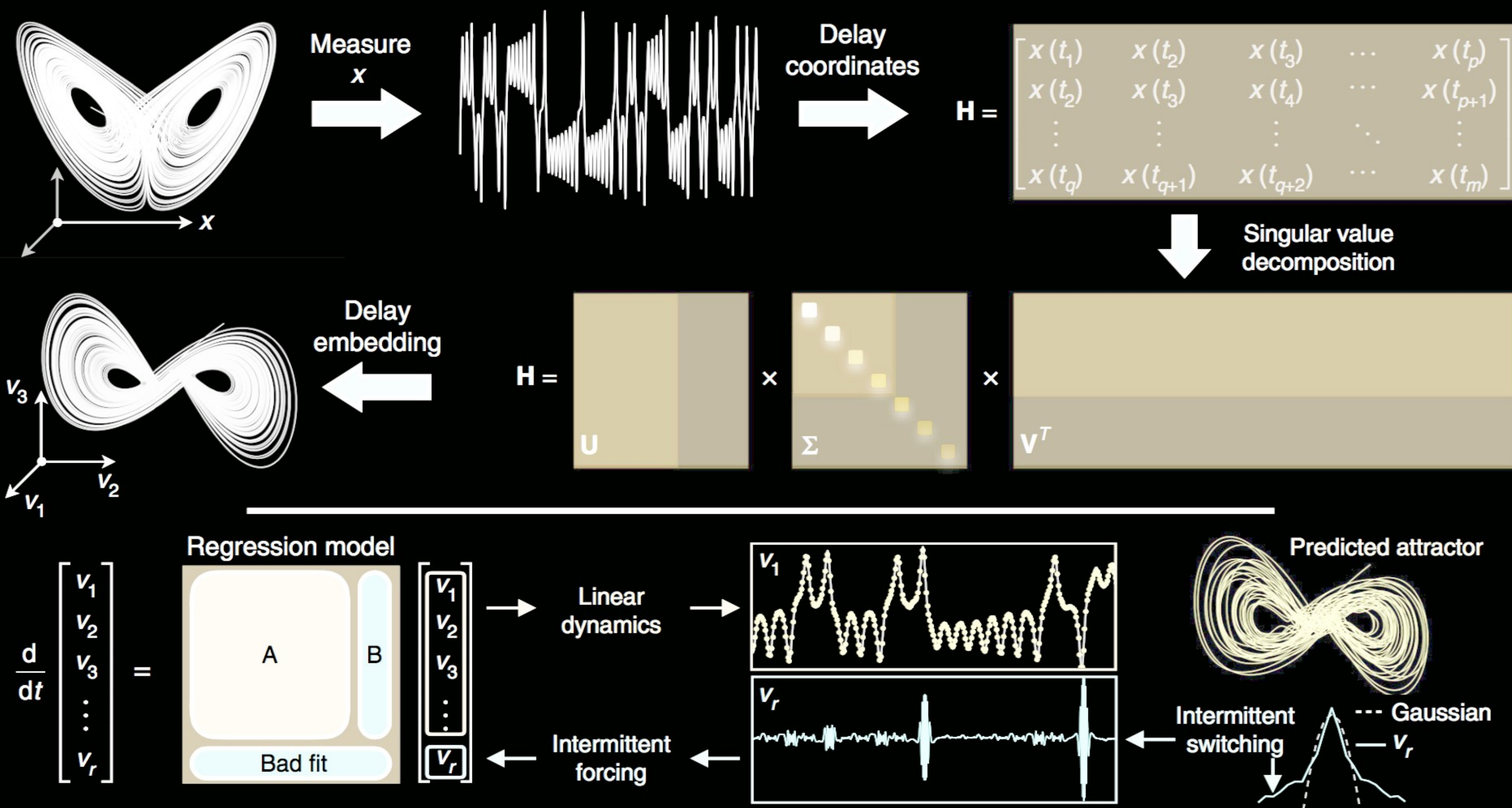
Multiscale Physics Discovery



Burst sampling



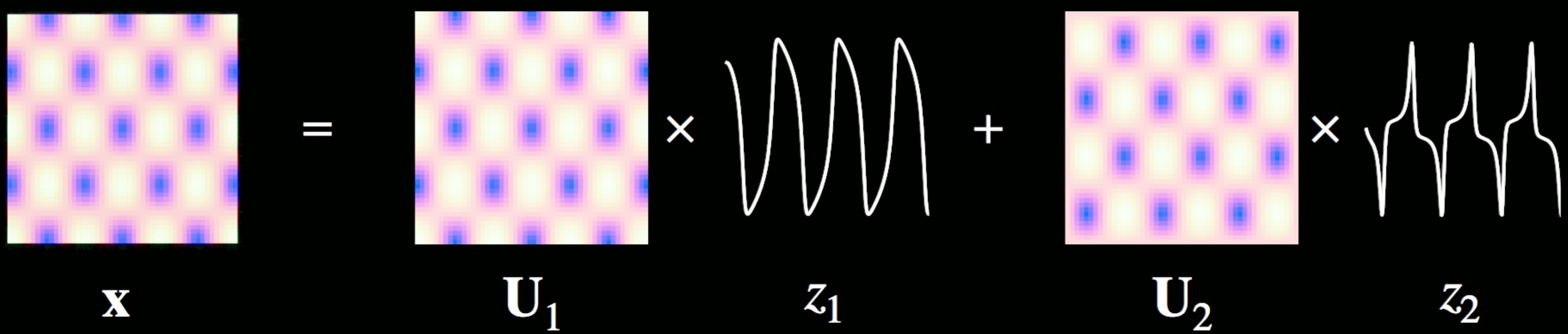
Latent Variables


Measure
x
Delay
coordinates
H =
x (t1) x (t2) x (t3) ⋯ x (tp)
x (t2) x (t3) x (t4) ⋯ x (tp+1)
x (tq) x (tq+1) x (tq+2) ⋯ x (tm)
Singular value
decomposition
H =
U
Σ
V^T
Delay
embedding
v3
v2
v1
Regression model
d/dt
v1
v2
v3
vr
=
A
B
Bad fit
Linear
dynamics
Intermittent
forcing
v1
vr
Predicted attractor
Intermittent
switching
Gaussian
vr

Latent variables

$$\mathbf{x} = \mathbf{U}_1 z_1 + \mathbf{U}_2 z_2$$

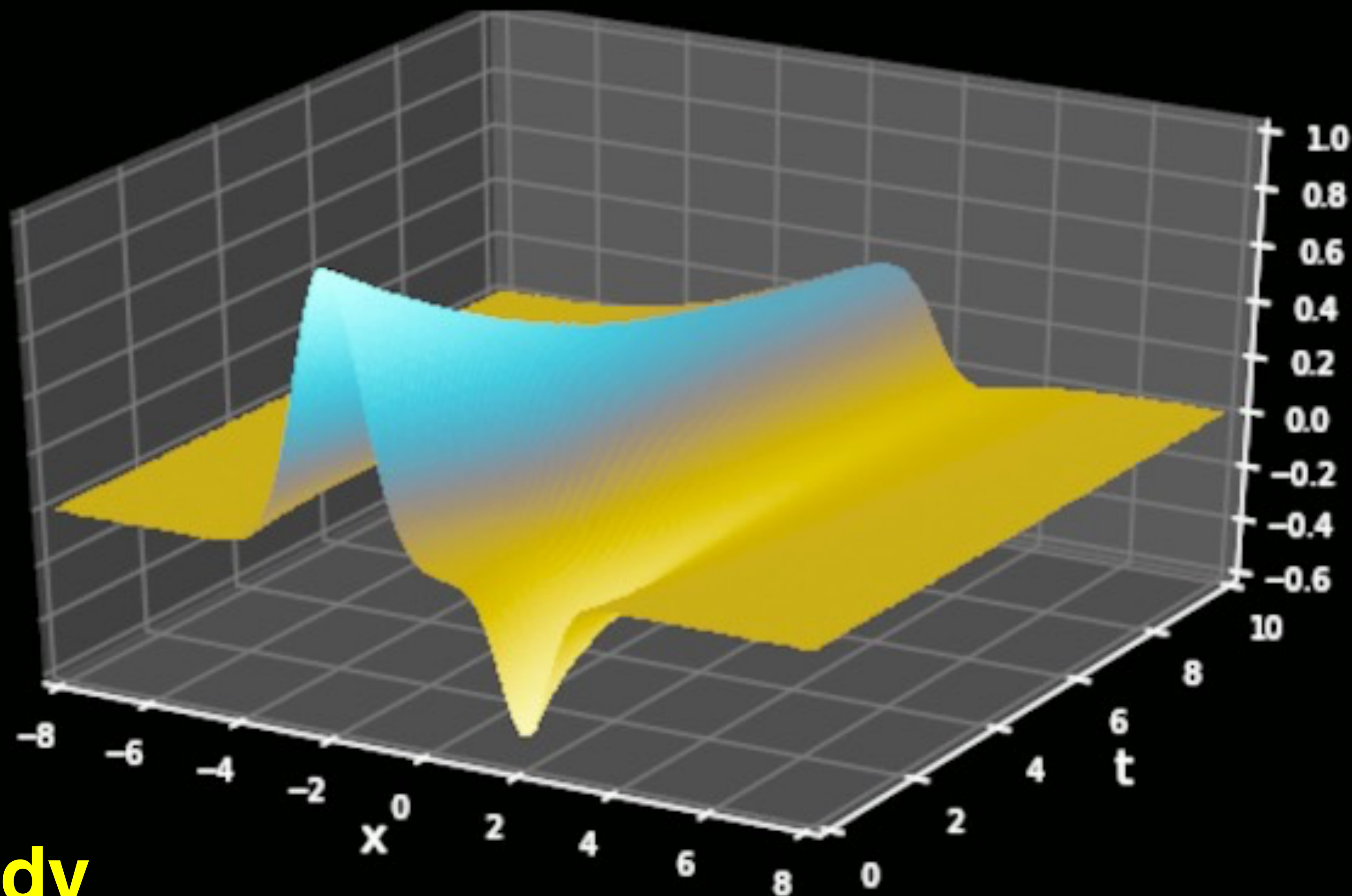
$$\dot{z}_1 = z_2$$
$$\dot{z}_2 = \mu(1 - z_1^2)z_2 - z_1$$

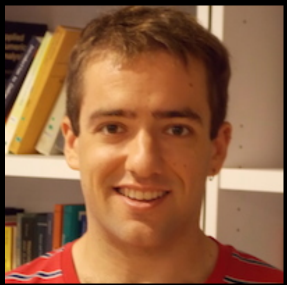


Parametric Systems

Parametric Burgers

$$u_t + \left(1 + \frac{1}{4}\sin(t)\right) uu_x - Du_{xx} = 0$$



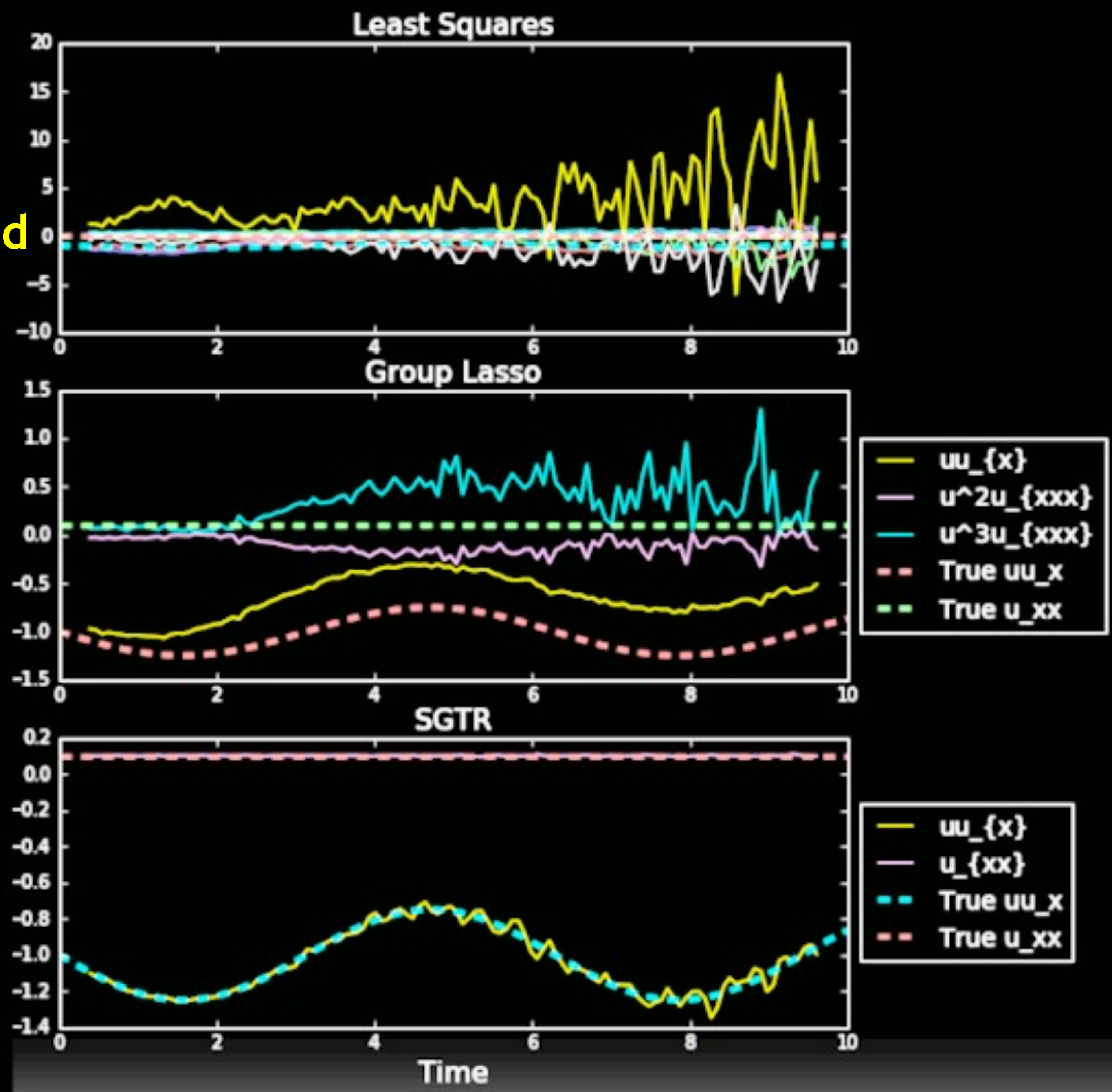


Sam Rudy

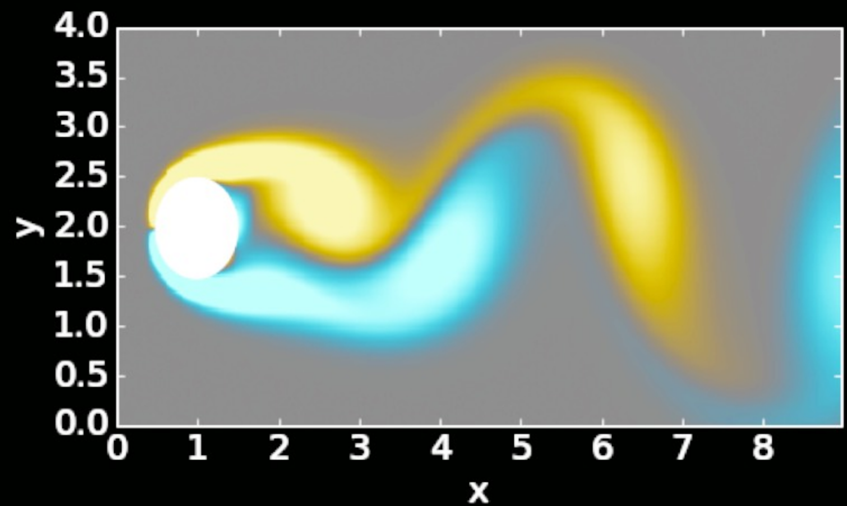
Parametric Discovery

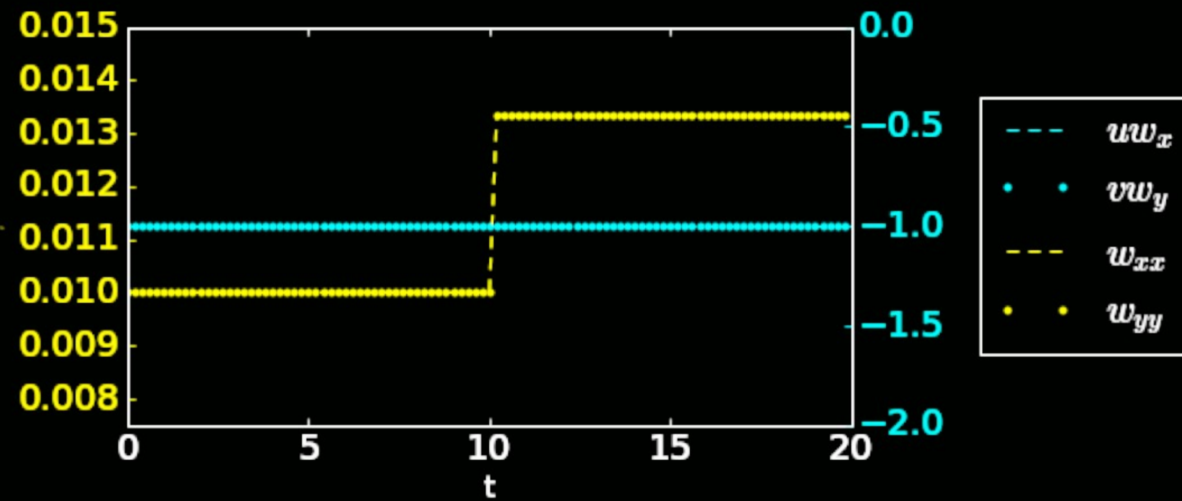
Group LASSO vs
Sequential Group Threshold
Regression (SGTR)

Our innovation: SGTR
(works amazingly well!)

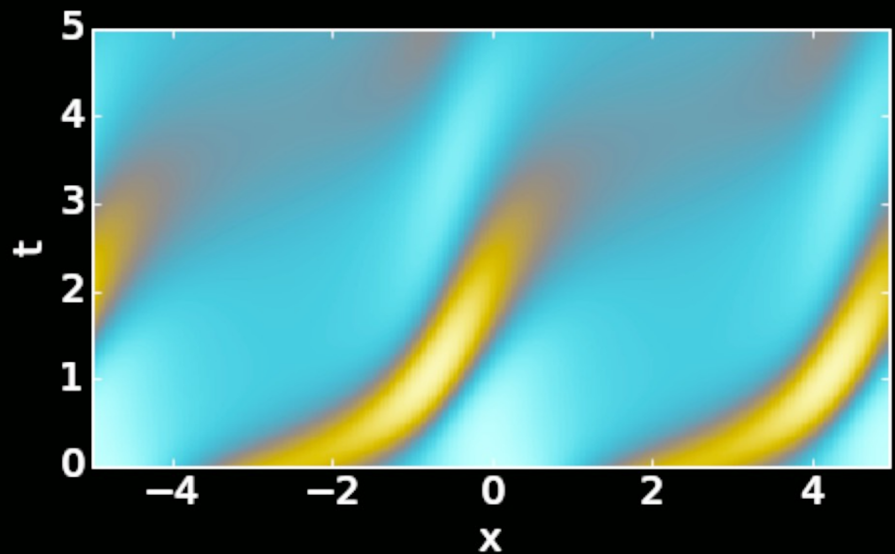


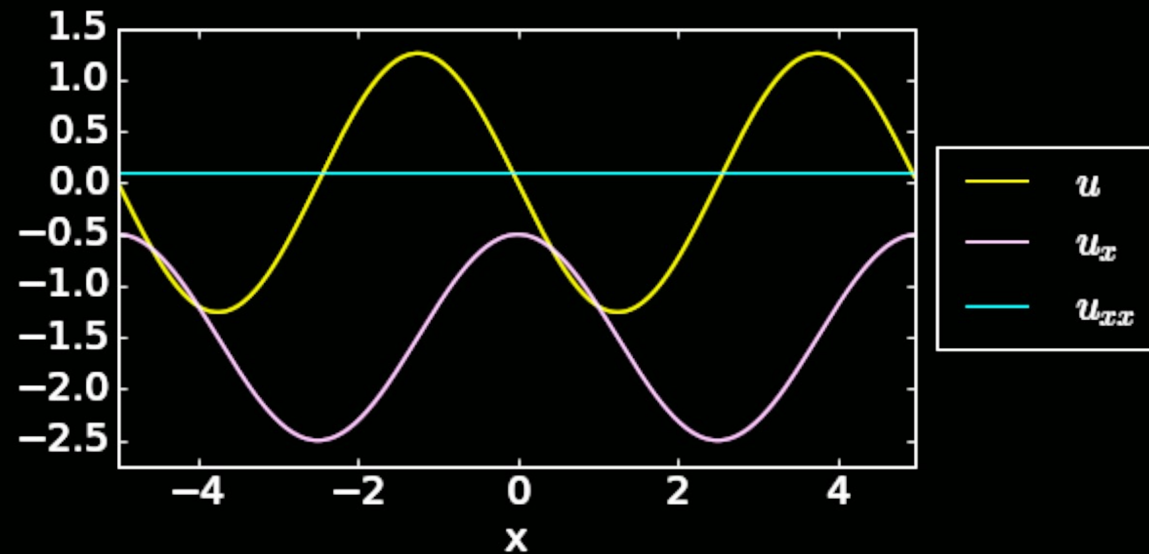
Parametric Dependence



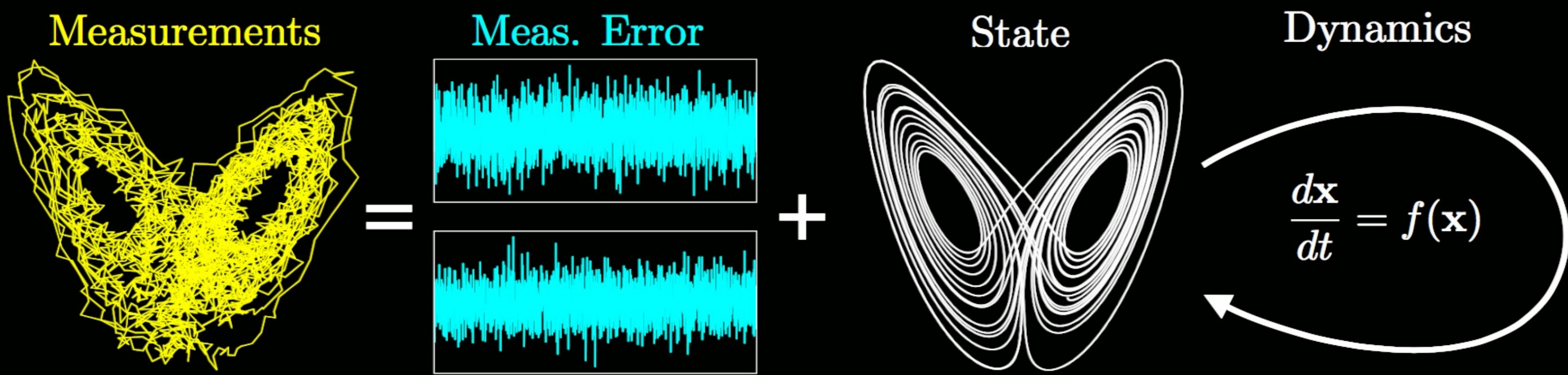


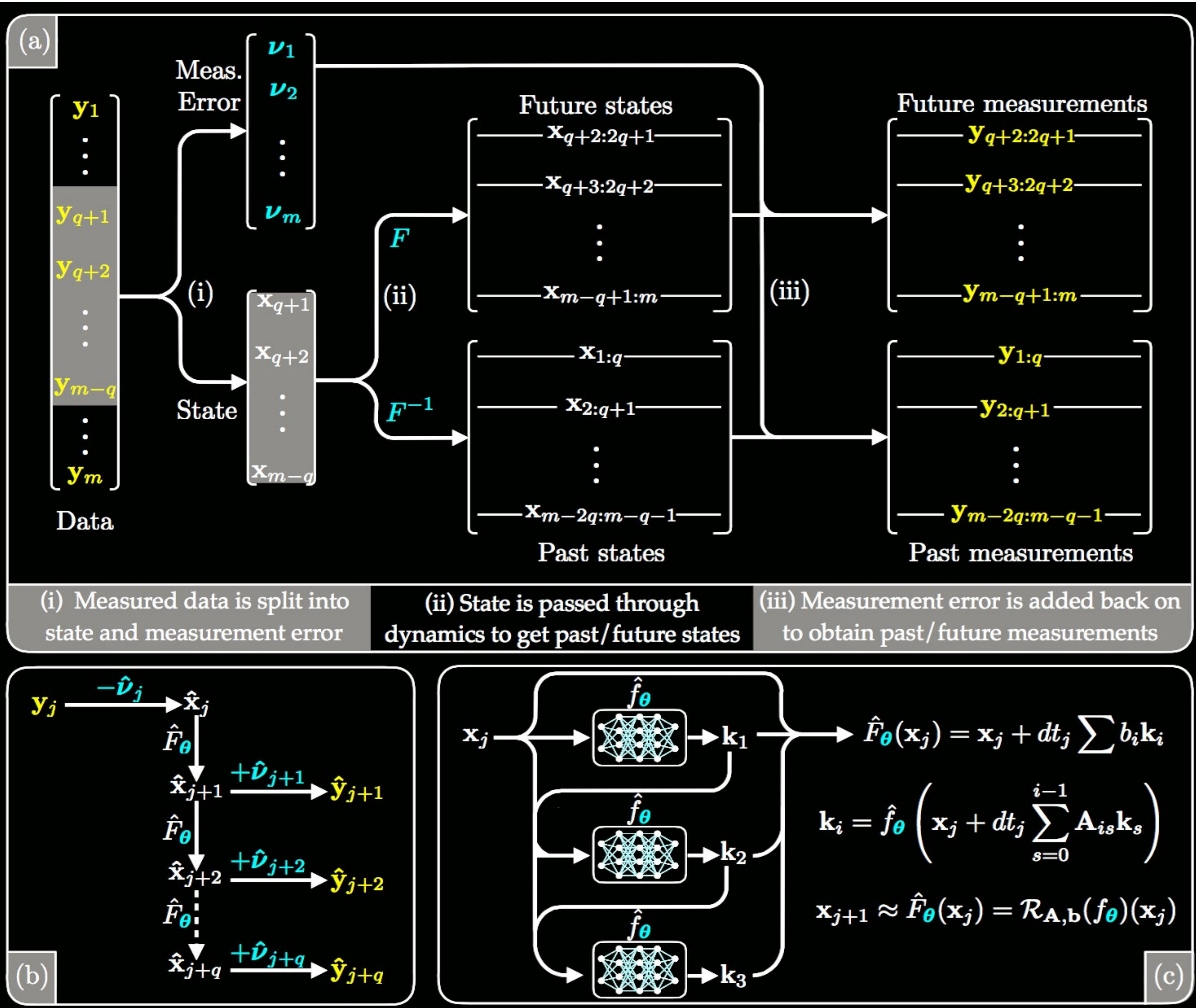
$$u_t = (c(x)u)_x + \epsilon u_{xx}$$



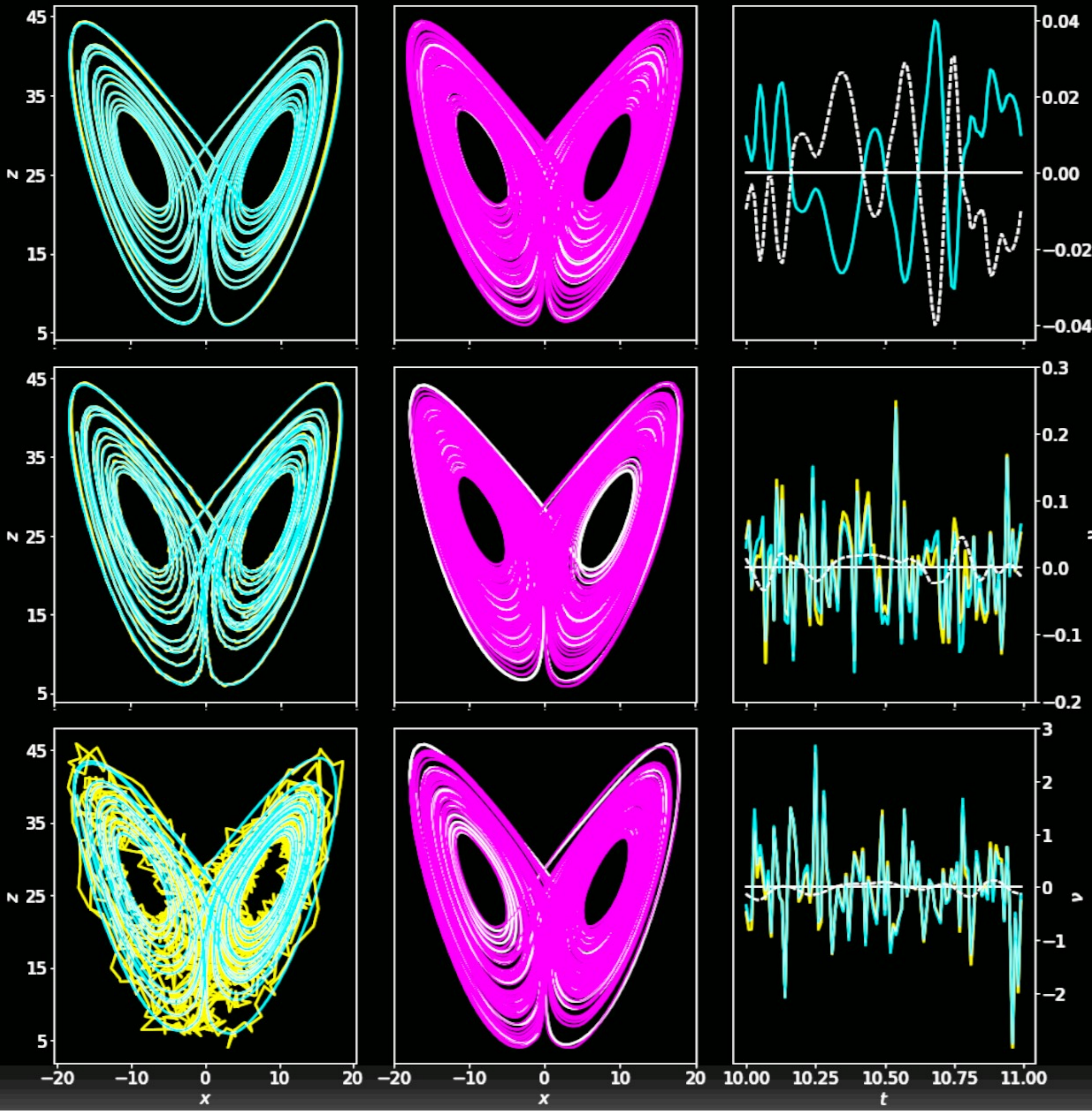


Noise


Measurements
=
Meas. Error
+
State
Dynamics
$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x})$$



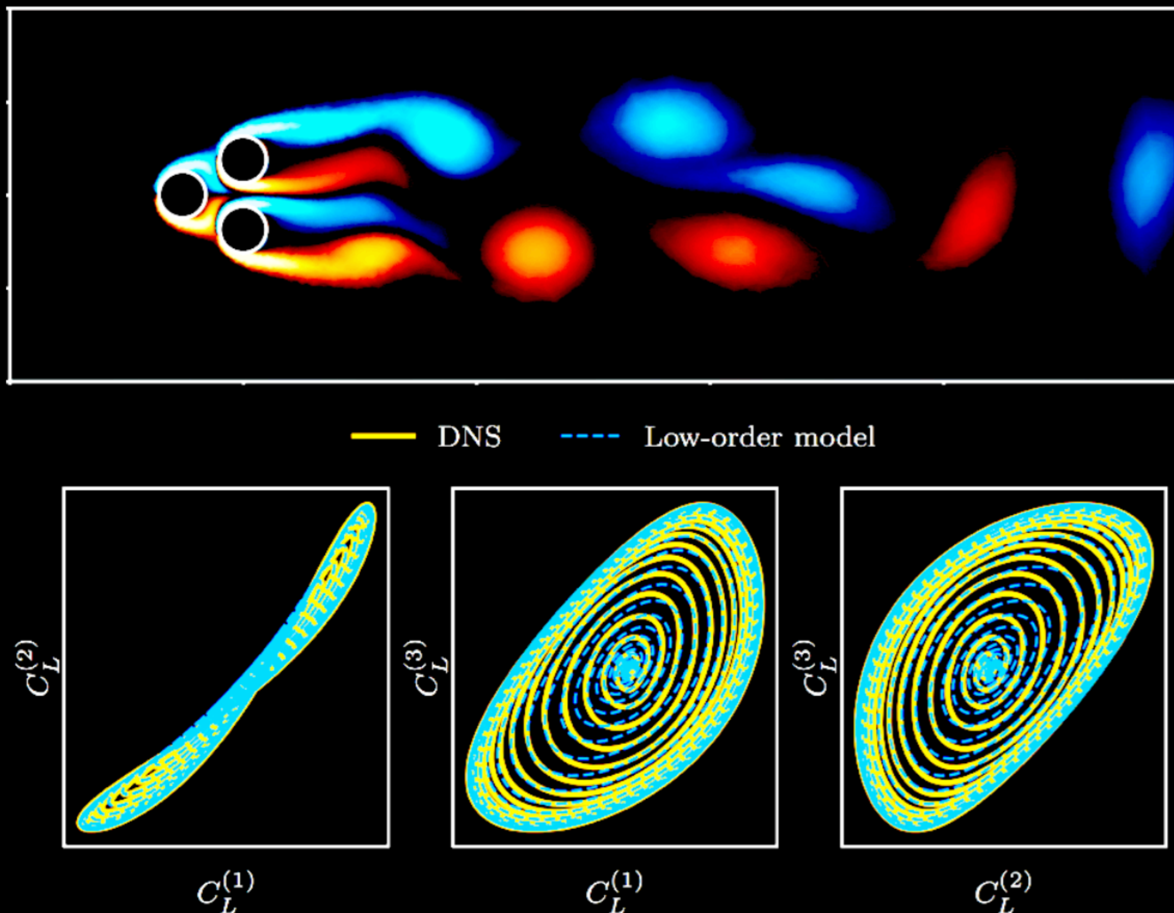
(a)
Meas. Error
ν_1 ν_2 ν_m
Data
$\mathbf{y}_1$ $\mathbf{y}_{q+1}$ $\mathbf{y}_{q+2}$ $\mathbf{y}_{m-q}$ $\mathbf{y}_m$
State
$\mathbf{x}_{q+1}$ $\mathbf{x}_{q+2}$ $\mathbf{x}_{m-q}$
(i)
(ii)
(iii)
F
F^{-1}
Future states
$\mathbf{x}_{q+2:2q+1}$ $\mathbf{x}_{q+3:2q+2}$ $\mathbf{x}_{m-q+1:m}$
Past states
$\mathbf{x}_{1:q}$ $\mathbf{x}_{2:q+1}$ $\mathbf{x}_{m-2q:m-q-1}$
Future measurements
$\mathbf{y}_{q+2:2q+1}$ $\mathbf{y}_{q+3:2q+2}$ $\mathbf{y}_{m-q+1:m}$
Past measurements
$\mathbf{y}_{1:q}$ $\mathbf{y}_{2:q+1}$ $\mathbf{y}_{m-2q:m-q-1}$
(i) Measured data is split into state and measurement error
(ii) State is passed through dynamics to get past/future states
(iii) Measurement error is added back on to obtain past/future measurements
(b)
$\mathbf{y}_j \xrightarrow{-\hat{\nu}_j} \hat{\mathbf{x}}_j$
$\hat{F}_\theta$
$\hat{\mathbf{x}}_{j+1} \xrightarrow{+\hat{\nu}_{j+1}} \hat{\mathbf{y}}_{j+1}$
$\hat{\mathbf{x}}_{j+2} \xrightarrow{+\hat{\nu}_{j+2}} \hat{\mathbf{y}}_{j+2}$
$\hat{\mathbf{x}}_{j+q} \xrightarrow{+\hat{\nu}_{j+q}} \hat{\mathbf{y}}_{j+q}$
(c)
$\mathbf{x}_j$
$\hat{f}_\theta$
$\mathbf{k}_1$ $\mathbf{k}_2$ $\mathbf{k}_3$
$\hat{F}_\theta(\mathbf{x}_j) = \mathbf{x}_j + dt_j \sum b_i \mathbf{k}_i$
$\mathbf{k}_i = \hat{f}_\theta \left(\mathbf{x}_j + dt_j \sum_{s=0}^{i-1} \mathbf{A}_{is} \mathbf{k}_s \right)$
$\mathbf{x}_{j+1} \approx \hat{F}_\theta(\mathbf{x}_j) = \mathcal{R}_{\mathbf{A},\mathbf{b}}(f_\theta)(\mathbf{x}_j)$



45
35
z 25
15
5
0.04
0.02
0.00
−0.02
−0.04
0.3
0.2
0.1
0.0
−0.1
−0.2
v
3
2
1
0
−1
−2
−20
−10
0
10
20
x
10.00
10.25
10.50
10.75
11.00
t

Control

Adding Energy Constraints



$$\min_{\xi, z} \|\boldsymbol{\Theta}(\boldsymbol{X})\boldsymbol{\Xi} - \dot{\boldsymbol{X}}\|_2^2 + z^T(\boldsymbol{C}\boldsymbol{\xi} - \boldsymbol{d})$$

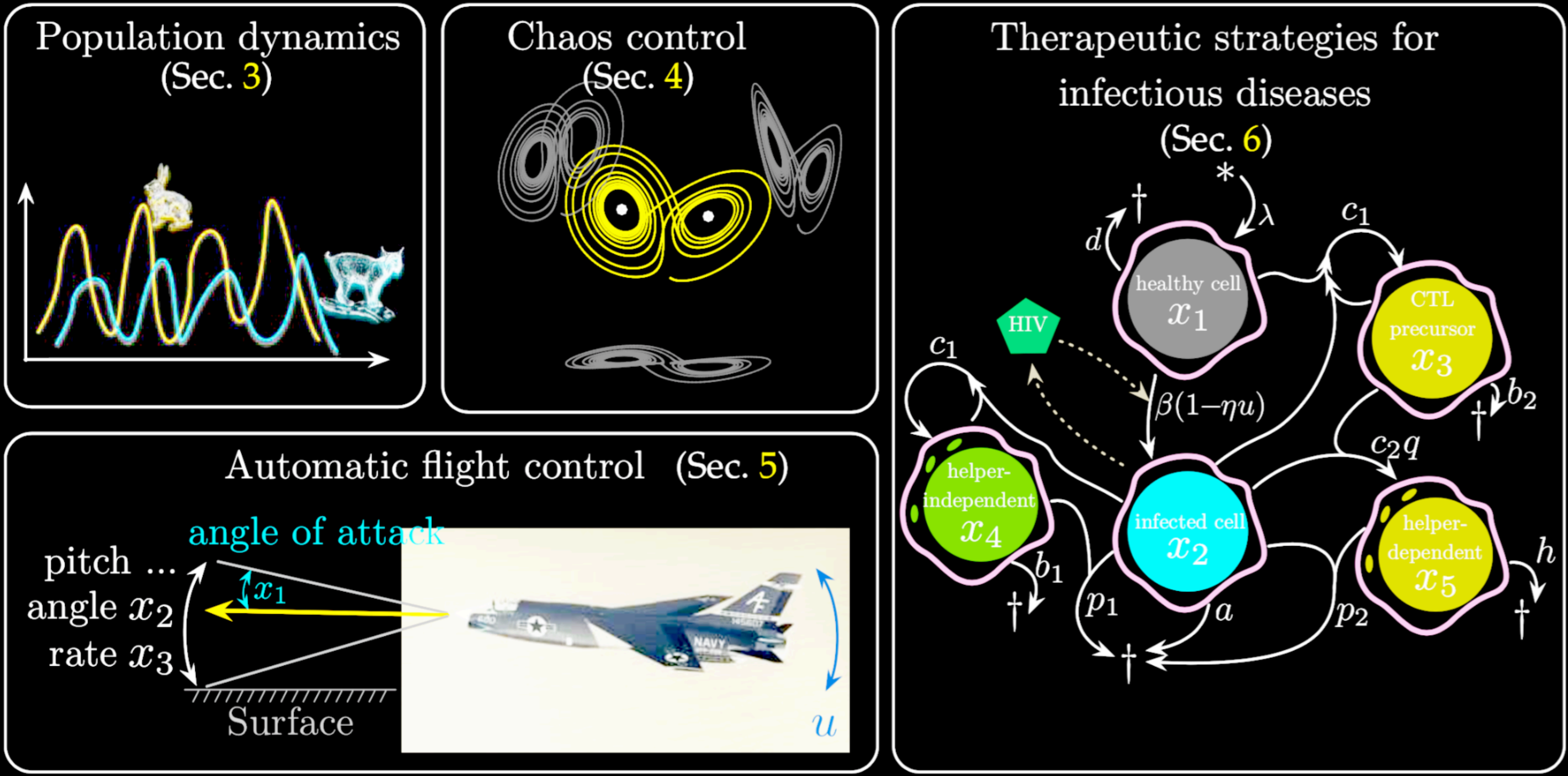
Spring-Mass Damper with Nonlinear Damping!

$$\ddot{x} - \underbrace{\left(0.2 - 0.24x^2 - 0.15\dot{x}^2\right)}_{k(x,\dot{x})}\dot{x} + 1.26x = 0$$

Loiseau & SLB, *JFM* 838, 2018

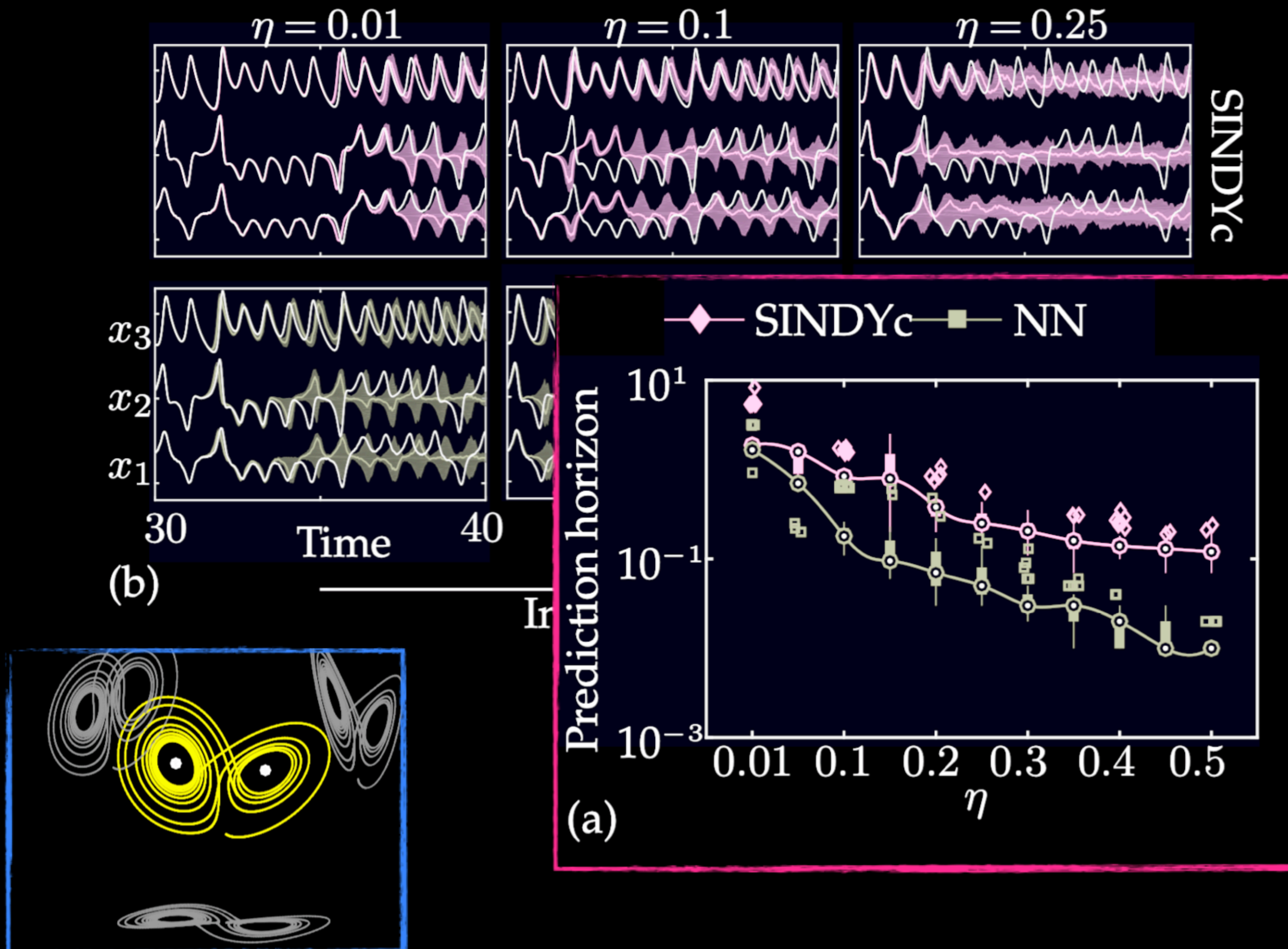
Loiseau, Noack, SLB, *JFM* 844, 2018

SINDY + MPC



Population dynamics
(Sec. 3)
Chaos control
(Sec. 4)
Therapeutic strategies for
infectious diseases
(Sec. 6)
healthy cell
x_1
CTL
precursor
x_3
HIV
helper-
independent
x_4
infected cell
x_2
helper-
dependent
x_5
d
λ
c_1
c_1
$\beta(1-\eta u)$
b_2
$c_2 q$
b_1
p_1
a
p_2
h
Automatic flight control (Sec. 5)
angle of attack
pitch ...
angle x_2
rate x_3
x_1
Surface
u

SINDY + MPC



$\eta = 0.01$
$\eta = 0.1$
$\eta = 0.25$
SINDYc
x_3
x_2
x_1
30
Time
40
(b)
In
SINDYc
NN
Prediction horizon
10^1
10^{-1}
10^{-3}
0.01 0.1 0.2 0.3 0.4 0.5
η
(a)

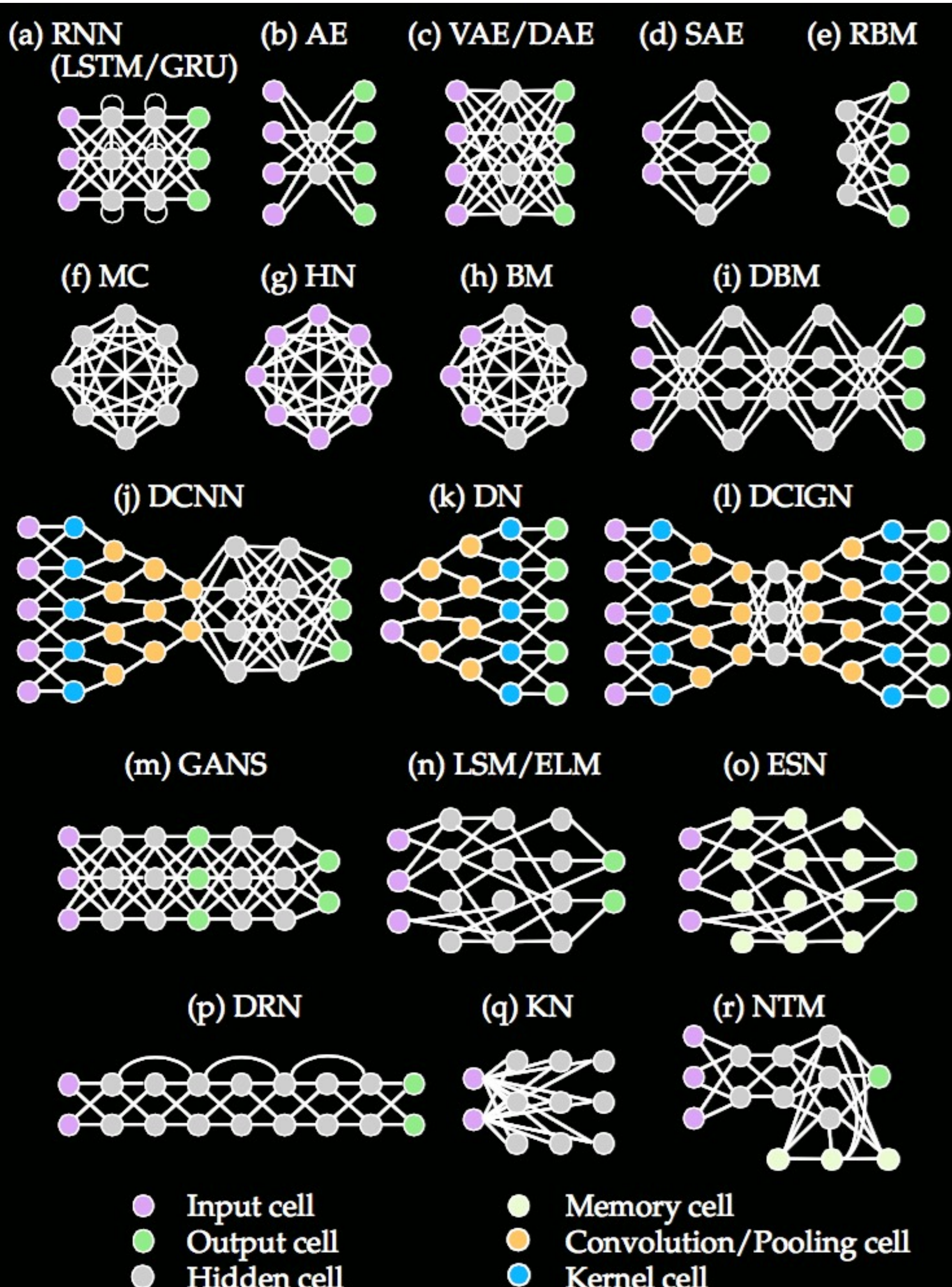
SINDy Innovations

Schaeffer -- corrupt data, PDEs, integral formulation, convergence
Dongbin Shu & co-workers (2018) – Sampling strategies
Guang Lin & co-workers (2018) -- Uncertainty Metrics

Zheng, Askham, Brunton, Kutz & Aravkin (2018) – SR3 sparse relaxed regularized regression (for SINDy, LASSO, CS, TV, Matrix Completion …)

Neural Nets

NN Zoo

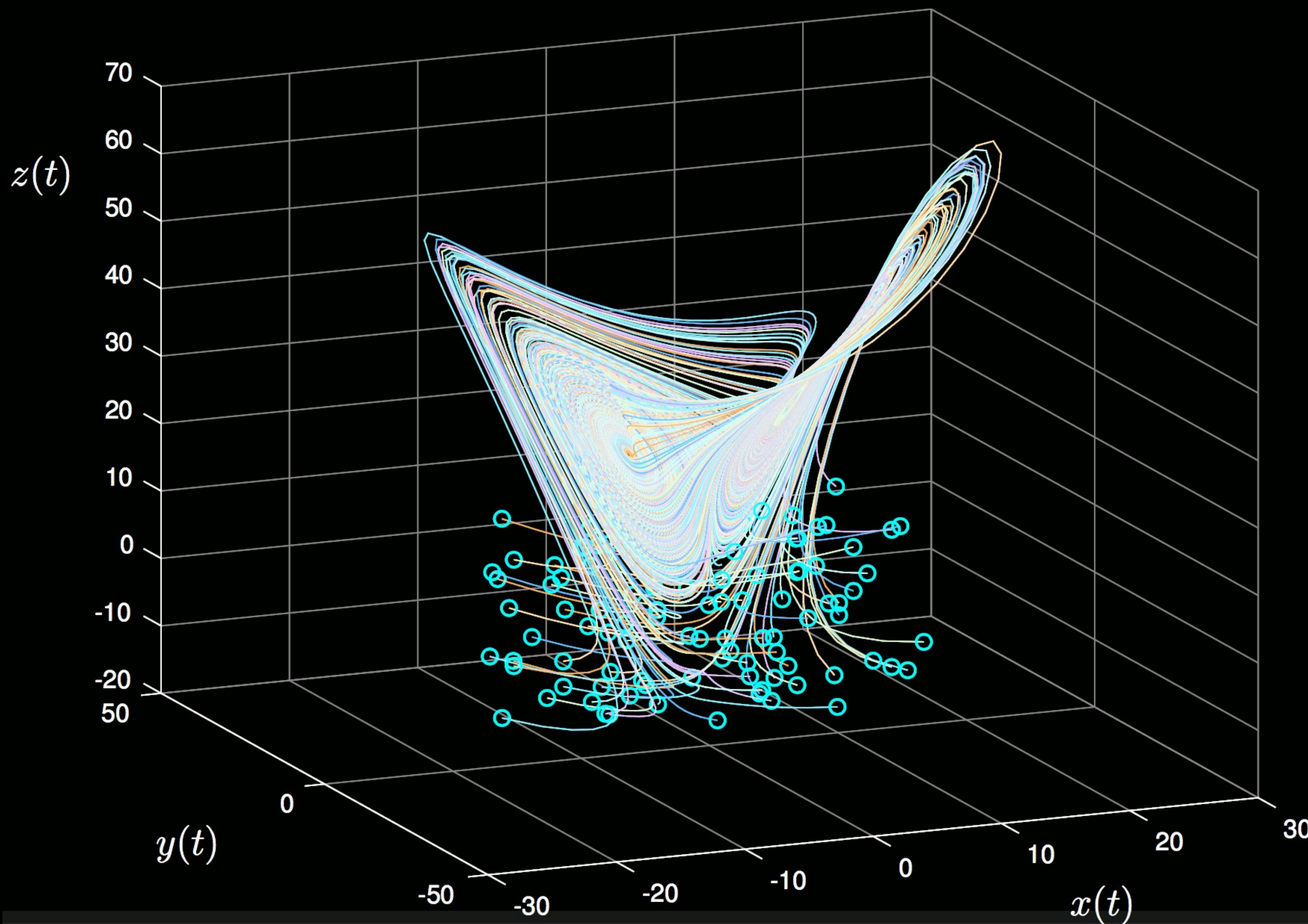


(a) RNN (LSTM/GRU)
(b) AE
(c) VAE/DAE
(d) SAE
(e) RBM
(f) MC
(g) HN
(h) BM
(i) DBM
(j) DCNN
(k) DN
(l) DCIGN
(m) GANS
(n) LSM/ELM
(o) ESN
(p) DRN
(q) KN
(r) NTM
Input cell
Output cell
Hidden cell
Memory cell
Convolution/Pooling cell
Kernel cell

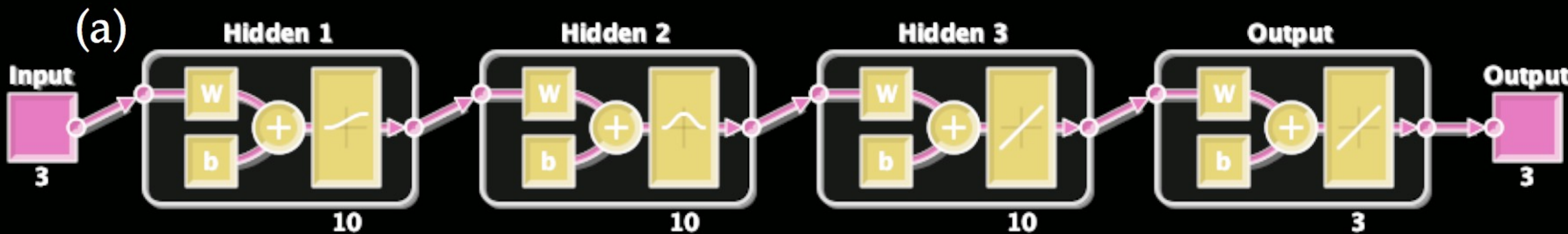
Neural Nets and Dynamics

```
% Simulate Lorenz system
dt=0.01; T=8; t=0:dt:T;
b=8/3; sig=10; r=28;

Lorenz = @(t,x)([ sig * (x(2) - x(1))         ; ...
                  r * x(1)-x(1) * x(3) - x(2) ; ...
                  x(1) * x(2) - b*x(3)           ]);
ode_options = odeset('RelTol',1e-10, 'AbsTol',1e-11);

input=[]; output=[];
for j=1:100  % training trajectories
    x0=30*(rand(3,1)-0.5);
    [t,y] = ode45(Lorenz,t,x0);
    input=[input; y(1:end-1,:)];
    output=[output; y(2:end,:)];
    plot3(y(:,1),y(:,2),y(:,3)), hold on
    plot3(x0(1),x0(2),x0(3),'ro')
end
```

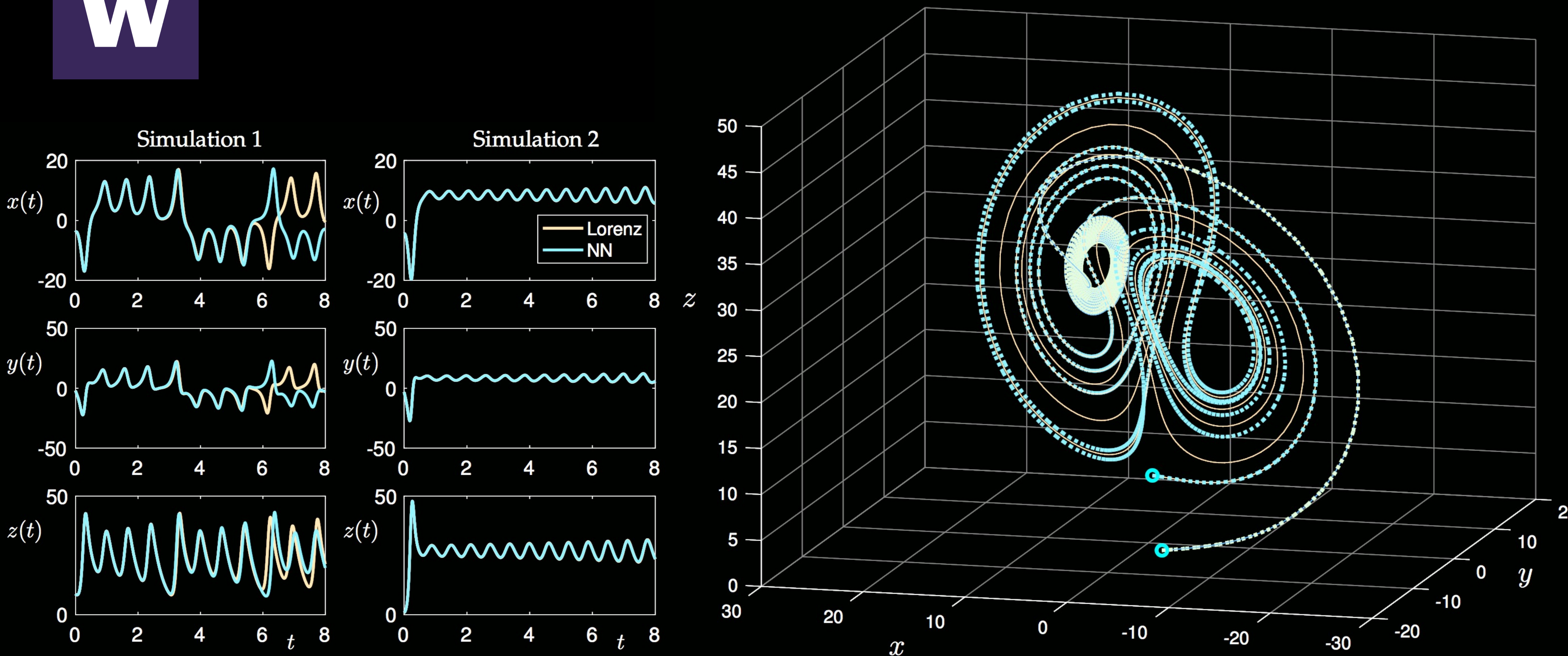
Trajectories for Training



MATLAB NN




```
net = feedforwardnet([10 10 10]);
net.layers{1}.transferFcn = 'logsig';
net.layers{2}.transferFcn = 'radbas';
net.layers{3}.transferFcn = 'purelin';
net = train(net,input.',output.');
```




```
ynn(1,:)=x0;
for jj=2:length(t)
    y0=net(x0);
    ynn(jj,:)=y0.'; x0=y0;
end
plot3(ynn(:,1),ynn(:,2),ynn(:,3),':','Linewidth',[2])
```

Part 2

Manifolds and Embeddings

Observables & Coordinates

Bernard Koopman 1931

Definition: Koopman Operator (Koopman 1931): *For a dynamical system*

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

where $\mathbf{x} \in \mathbb{R}^n$ *is in a state space* $\mathbf{x} \in \mathcal{M}$. *The Koopman operator* $\mathcal{K}$ *acts on a set of scalar observable variables* g_j *which comprise the vector* $\mathbf{g} : \mathcal{M} \to \mathbb{C}$ *so that*

$$\mathcal{K}\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{N}(\mathbf{x})).$$

Dynamic Mode Decomposition

Definition: Dynamic Mode Decomposition (Tu et al. 2014): *Suppose we have a dynamical system (1.17) and two sets of data*

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_M \\ | & | & & | \end{bmatrix}$$

$$\mathbf{X}' = \begin{bmatrix} | & | & & | \\ \mathbf{x}'_1 & \mathbf{x}'_2 & \cdots & \mathbf{x}'_M \\ | & | & & | \end{bmatrix}$$

with $\mathbf{x}_k$ *an initial condition to (1.17) and* $\mathbf{x}'_k$ *it corresponding output after some prescribed evolution time* τ *with there being* m *initial conditions considered. The DMD modes are eigenvectors of*

$$\mathrm{A}_{\mathbf{X}} = \mathbf{X}'\mathbf{X}^{\dagger}$$

where † *denotes the Moore-Penrose pseudoinverse.*

Travis Askham, ***optimized DMD***

Approximate Dynamical Systems

Linear dynamics
(equation-free)

$$\frac{d\tilde{\mathbf{x}}}{dt} = \mathbf{A}\tilde{\mathbf{x}}$$

Eigenfunction
expansion

$$\tilde{\mathbf{x}}(t) = \sum_{k=1}^{K} b_k \psi_k \exp(\omega_k t)$$

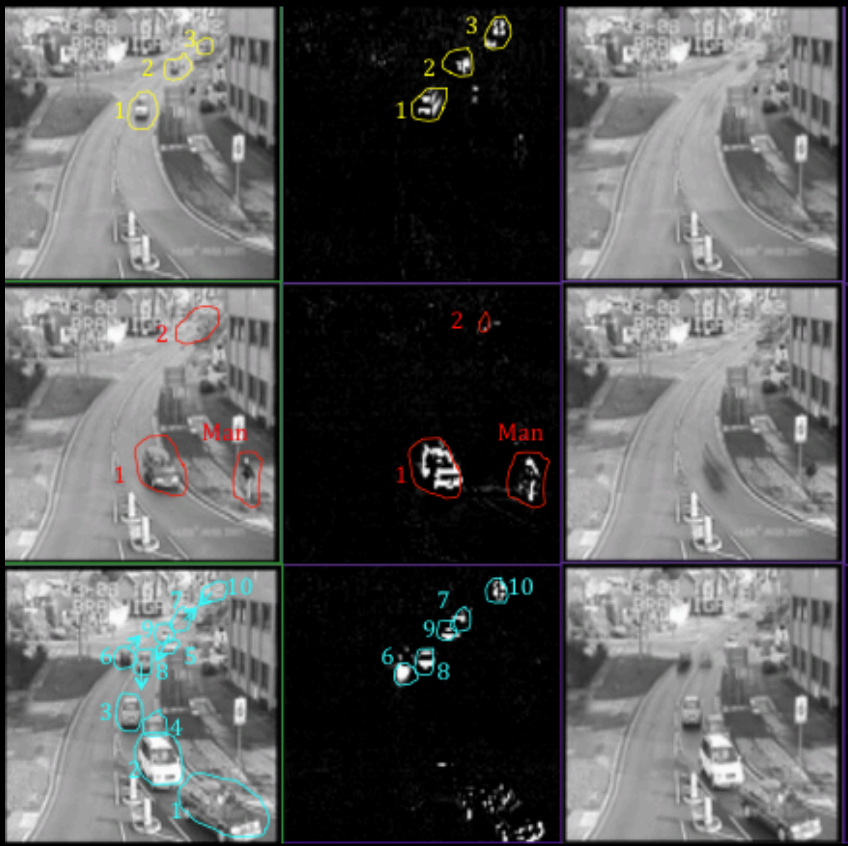
Least-square fit

$$\|\mathbf{x}(t) - \tilde{\mathbf{x}}(t)\| \ll 1$$

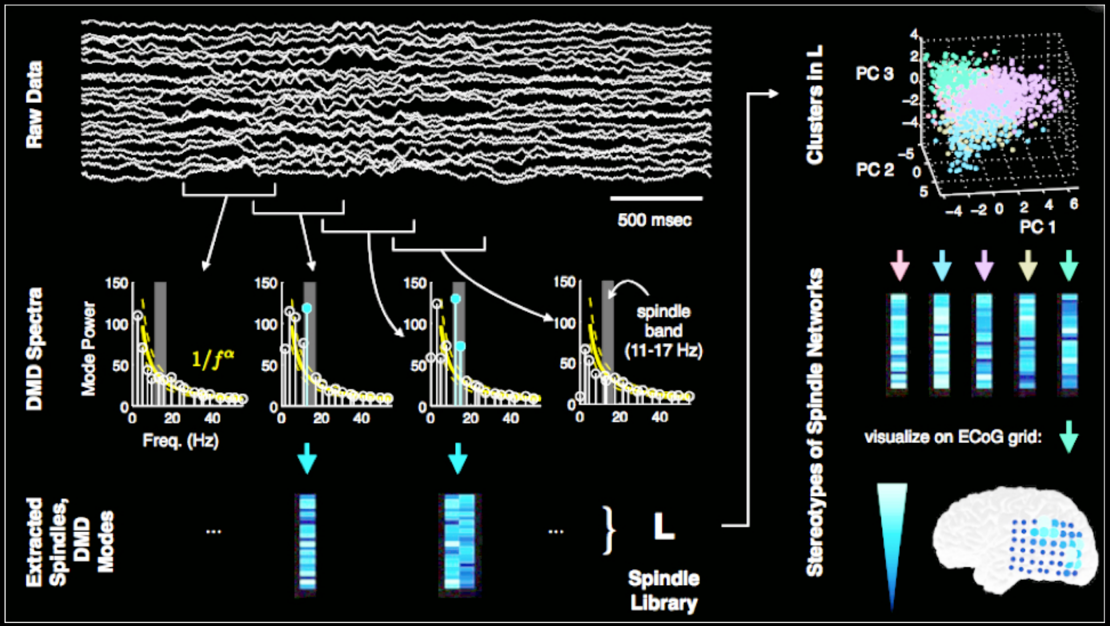
Some Applications

Dynamic Mode Decomposition for Financial Trading Strategies

Jordan Mann* and J. Nathan Kutz‡†



ECOG recordings



Erichson, Brunton & Kutz (2017)
Brunton, Johnson, Ojemann & Kutz (2017)

DMD with Control

Input

$$\mathbf{x}_{k+1} \approx \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k$$

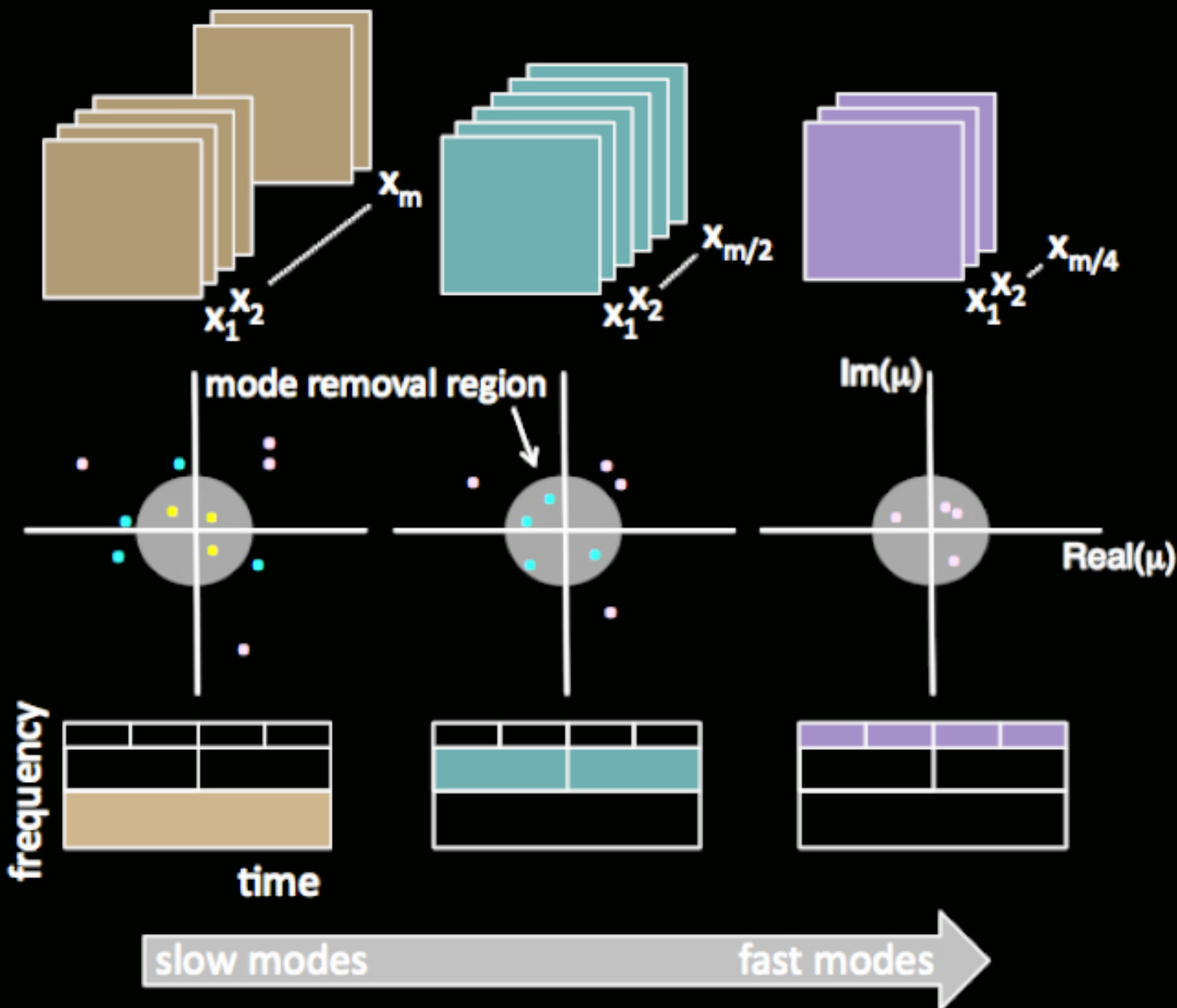
Input Snapshots

$$\boldsymbol{\Upsilon} = \begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_{m-1} \\ | & | & & | \end{bmatrix}$$

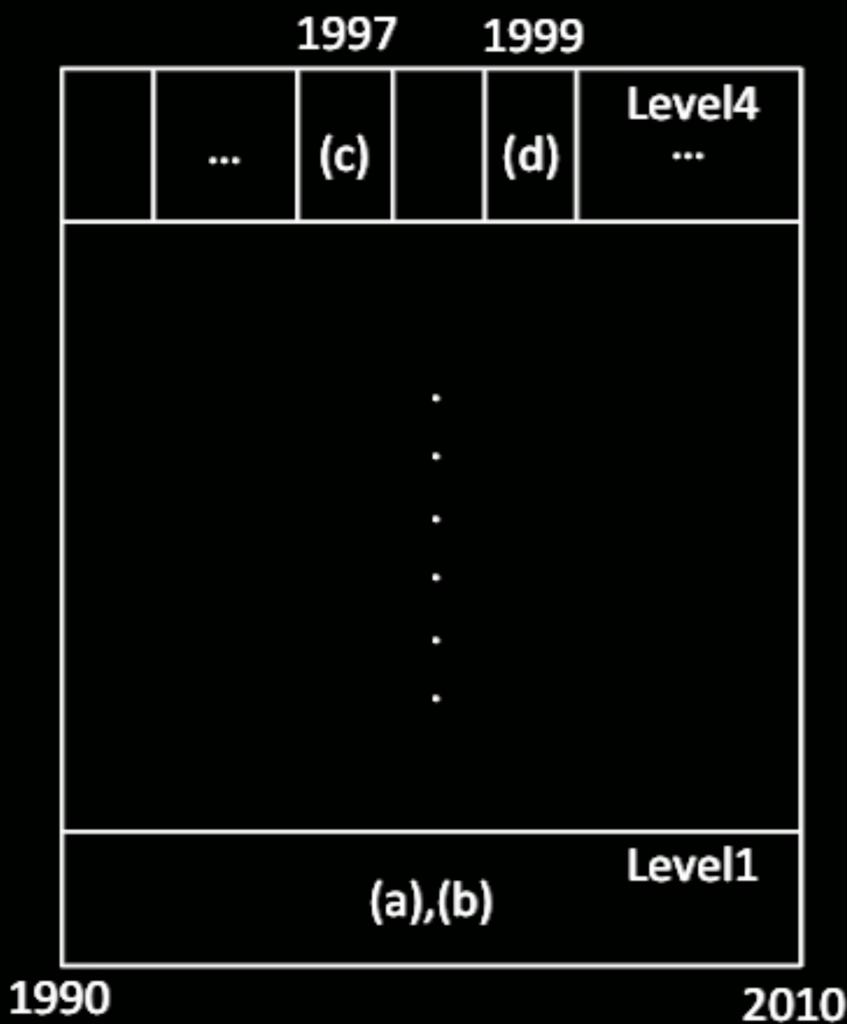
DMD generalization

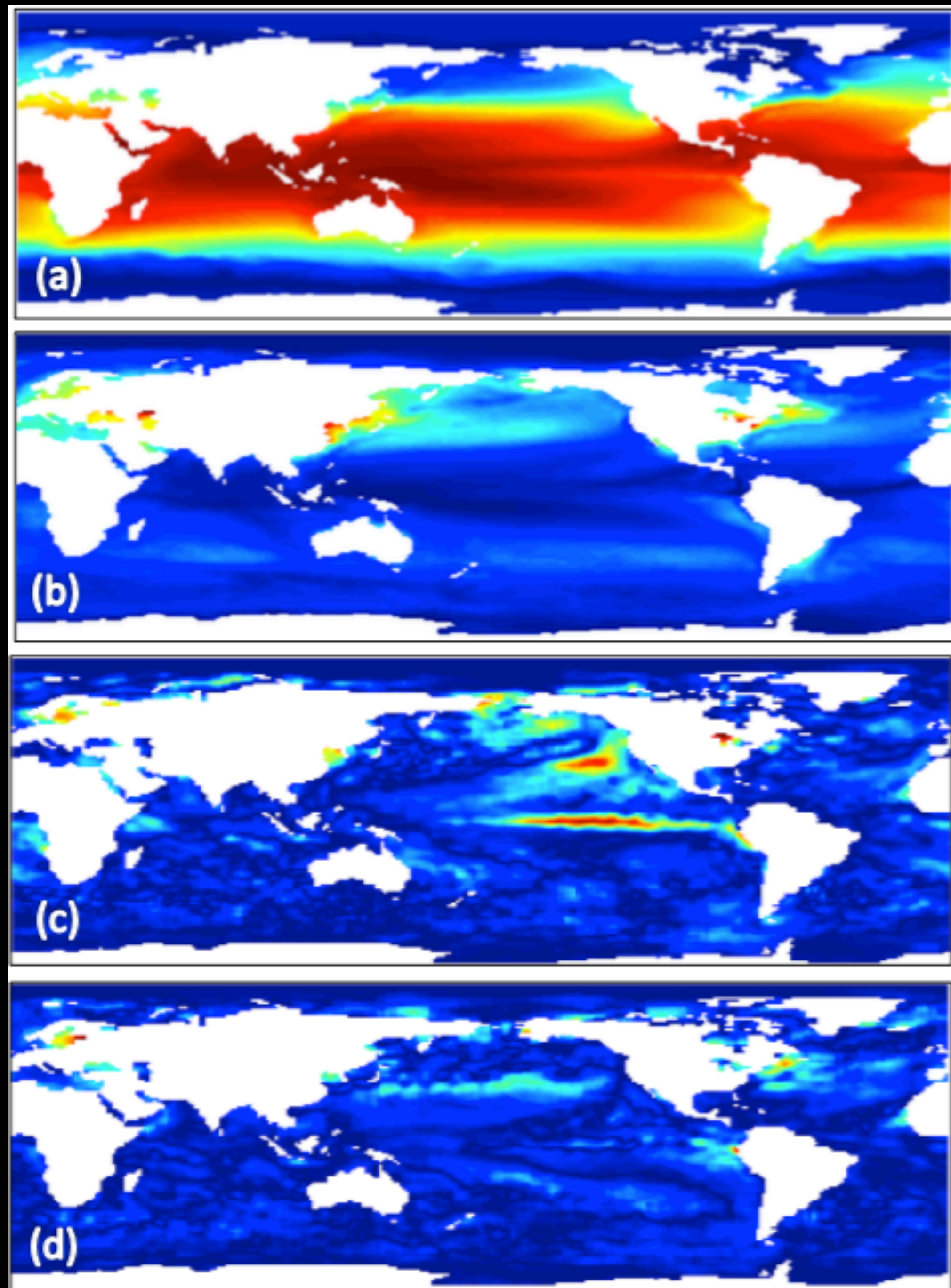
$$\mathbf{X}' \approx \mathbf{A}\mathbf{X} + \mathbf{B}\boldsymbol{\Upsilon}$$

Multi-Resolution DMD

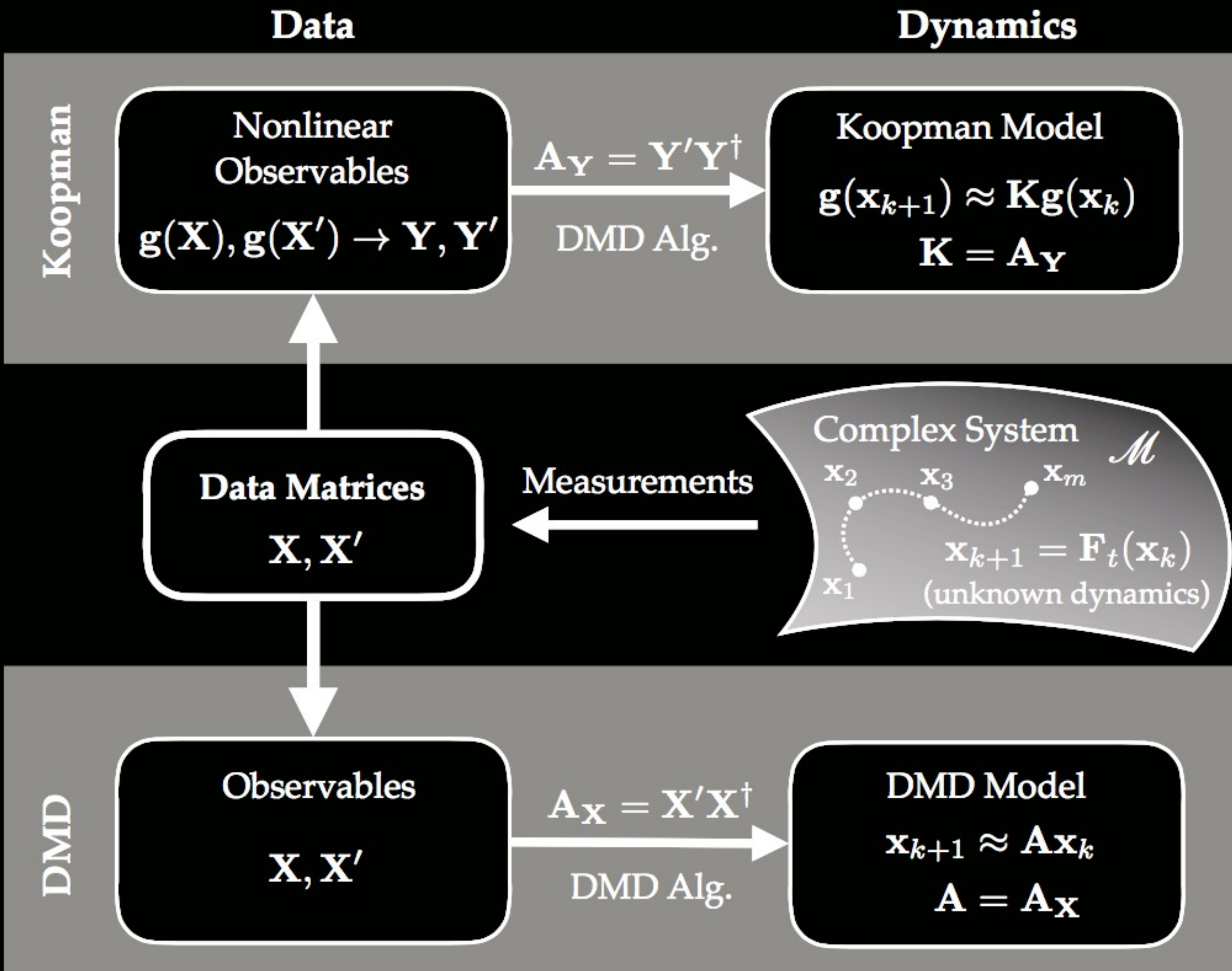


SST data & El Nino (1990s-2010+)



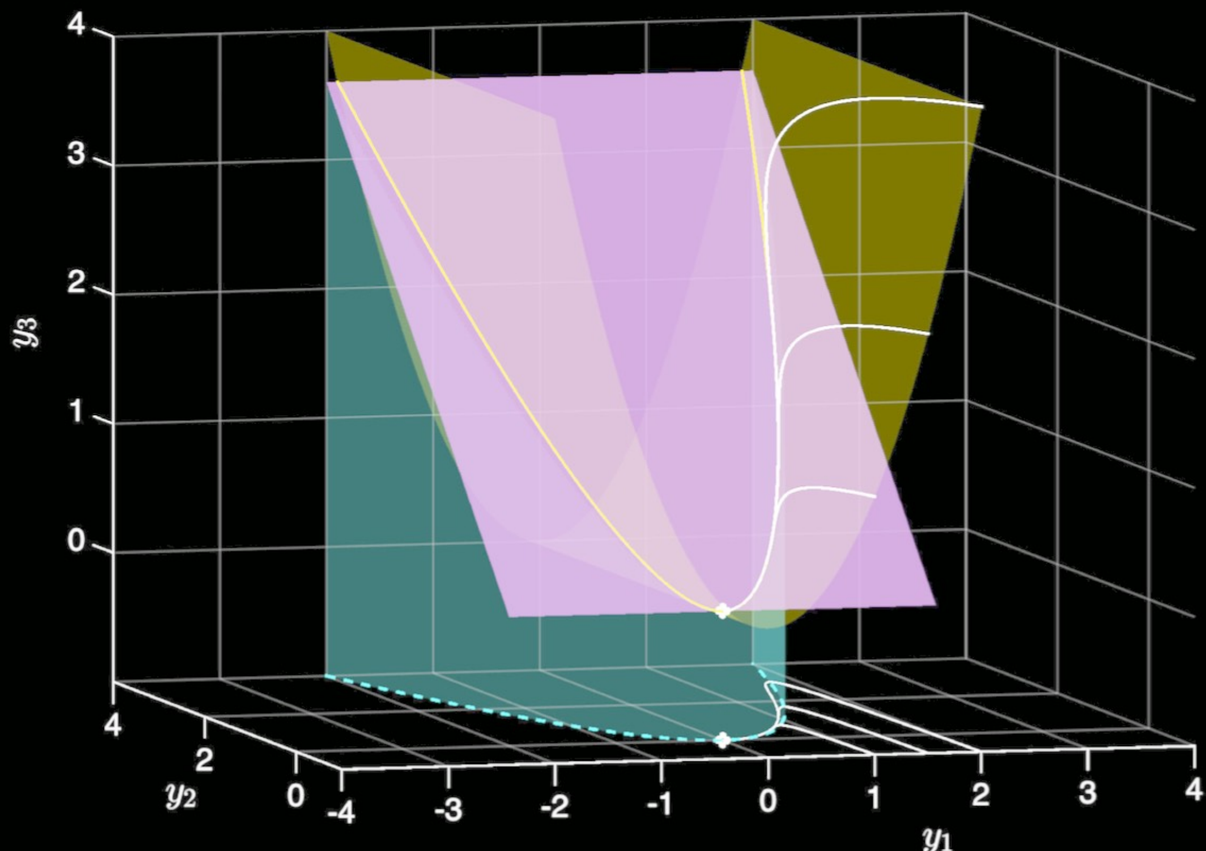


Koopman vs DMD: All about Observables!



Koopman Invariant Subspaces

$$\left.\begin{aligned}\dot{x}_1 &= \mu x_1 \\ \dot{x}_2 &= \lambda(x_2 - x_1^2)\end{aligned}\right\} \implies \frac{d}{dt}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \mu & 0 & 0 \\ 0 & \lambda & -\lambda \\ 0 & 0 & 2\mu \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad \text{for} \quad \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 \end{bmatrix}$$



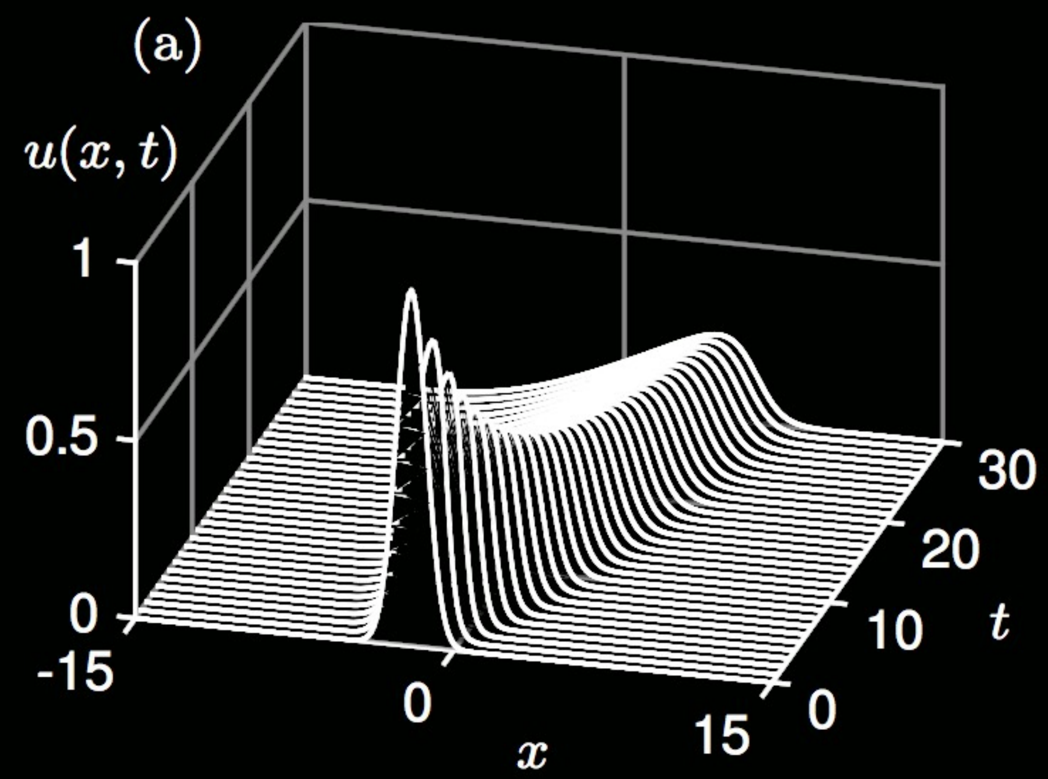
Burgers' Equation

$$u_t + uu_x - \epsilon u_{xx} = 0 \qquad \epsilon > 0,\ x \in [-\infty, \infty]$$

Cole-Hopf

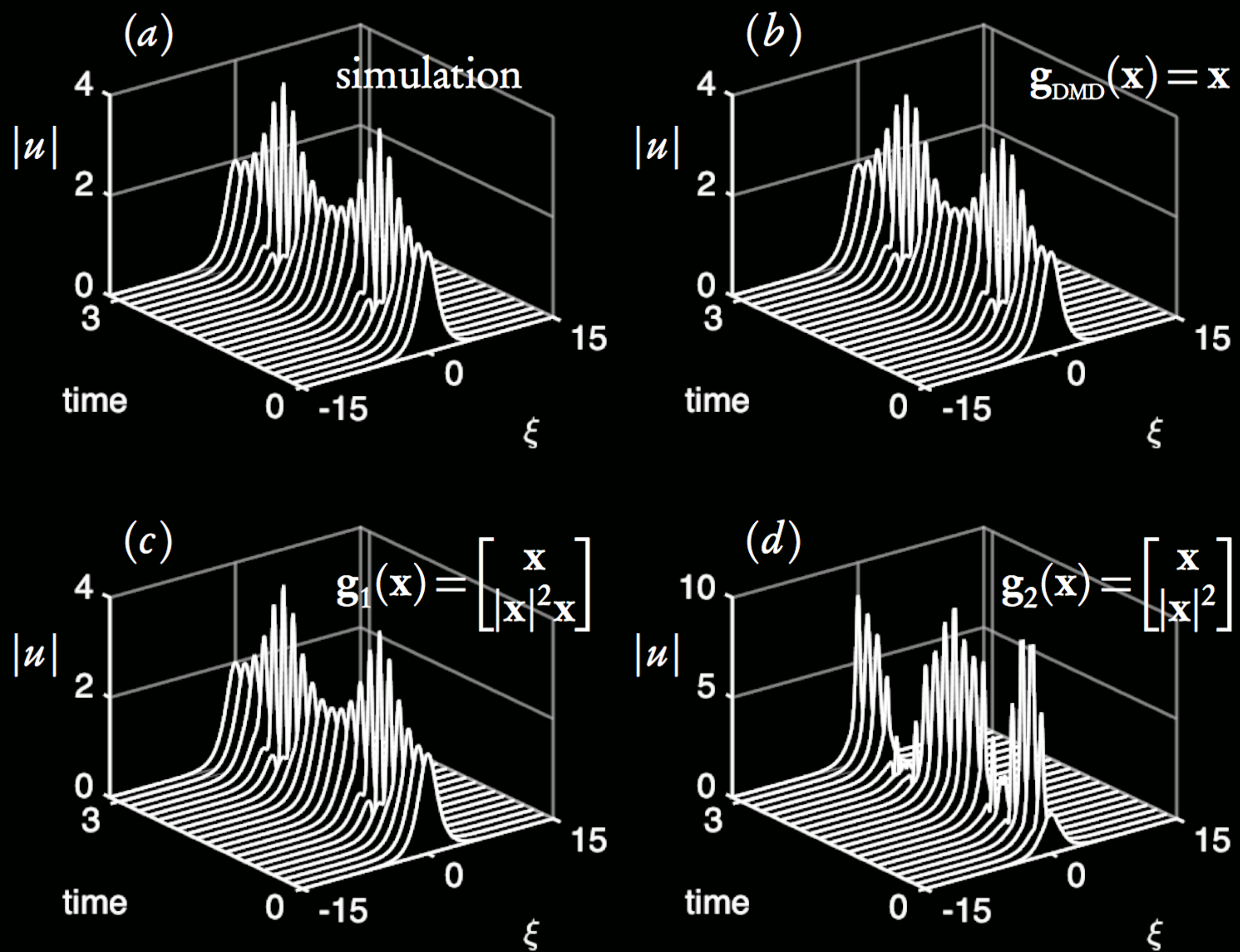
$$u = -2\epsilon v_x / v$$

$$v_t = \epsilon v_{xx}$$

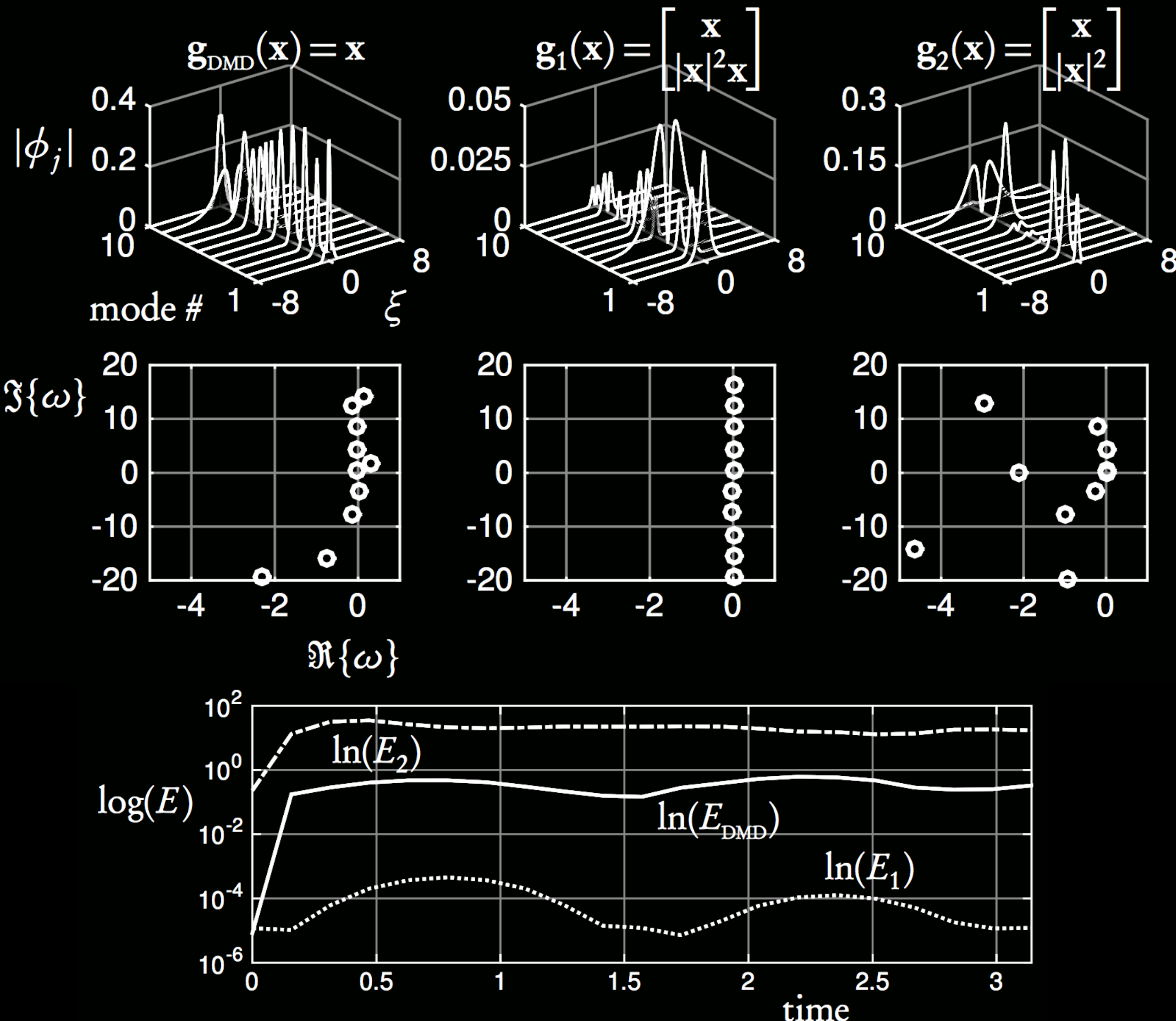


Nonlinear Schrodinger Equation

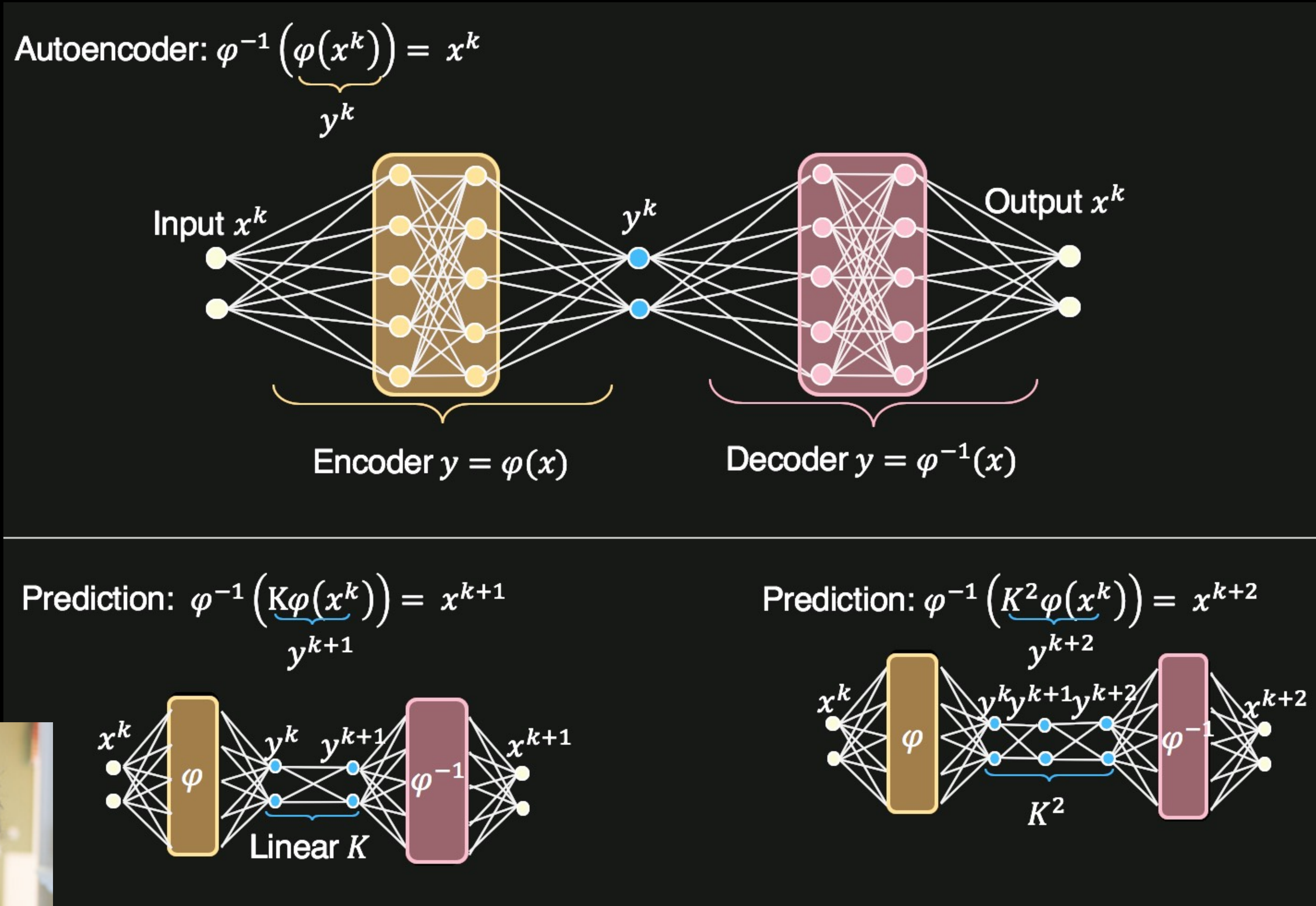
$$i\frac{\partial u}{\partial t}+\frac{1}{2}\frac{\partial^2 u}{\partial \xi^2}+|u|^2 u=0$$



Error and DMD Modes

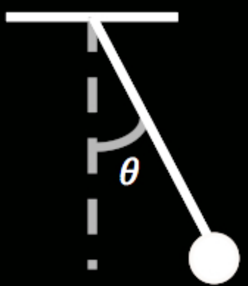


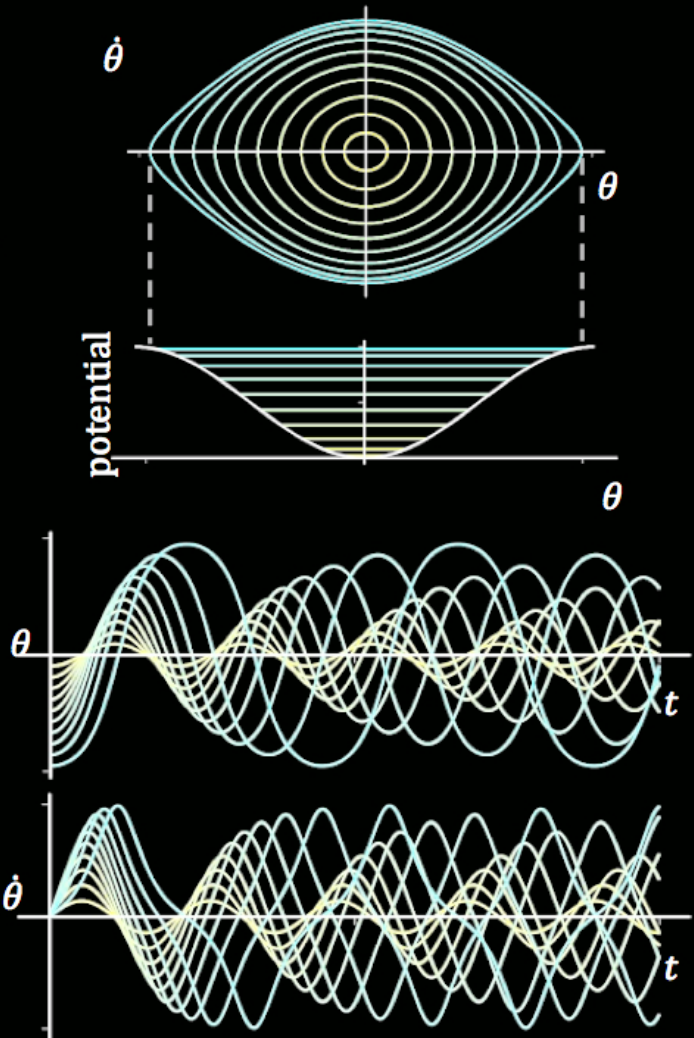
NNs for Koopman Embedding

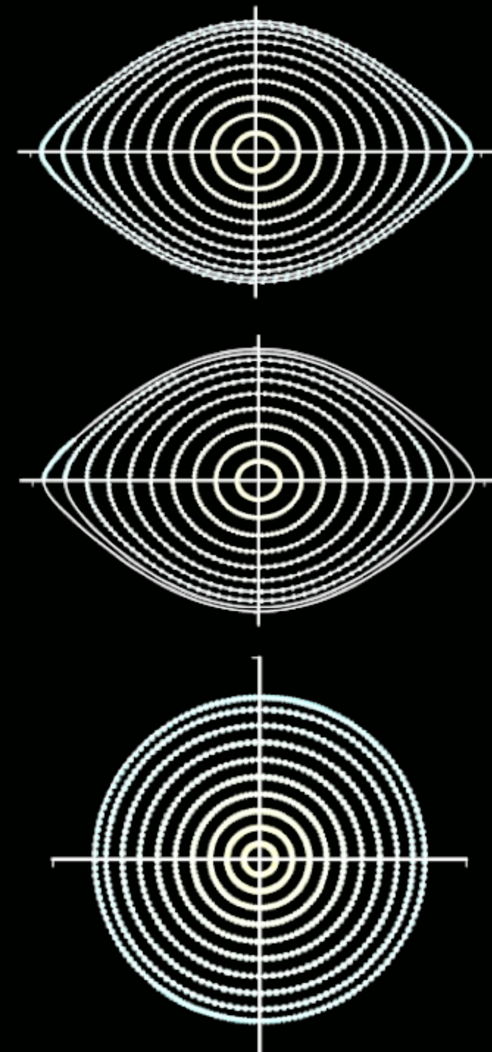


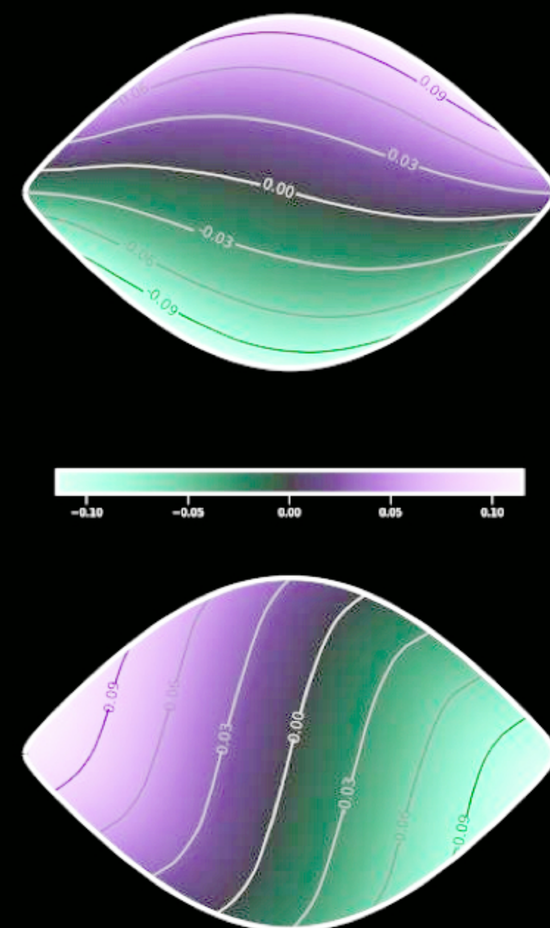
Bethany Lusch

The Pendulum

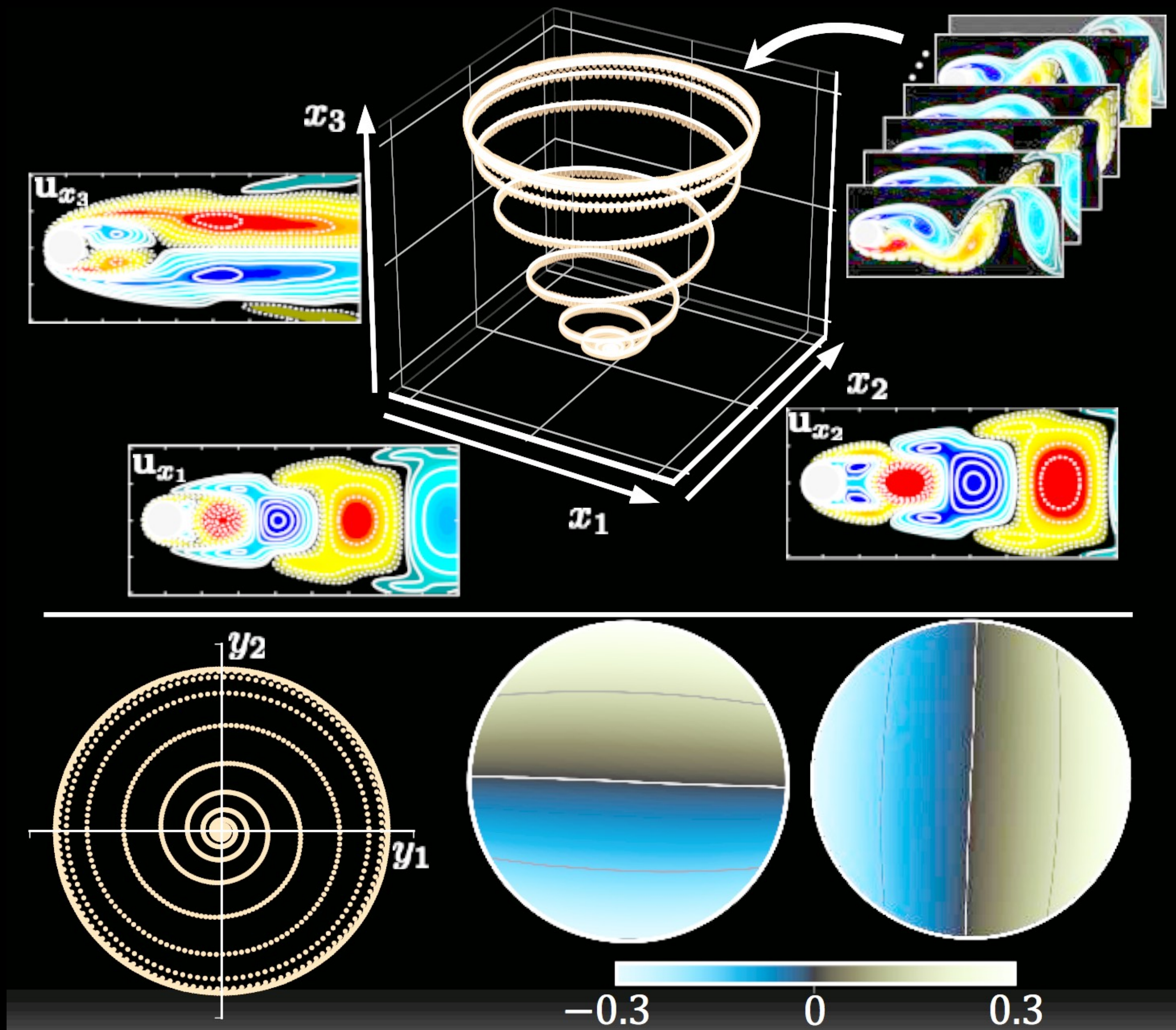




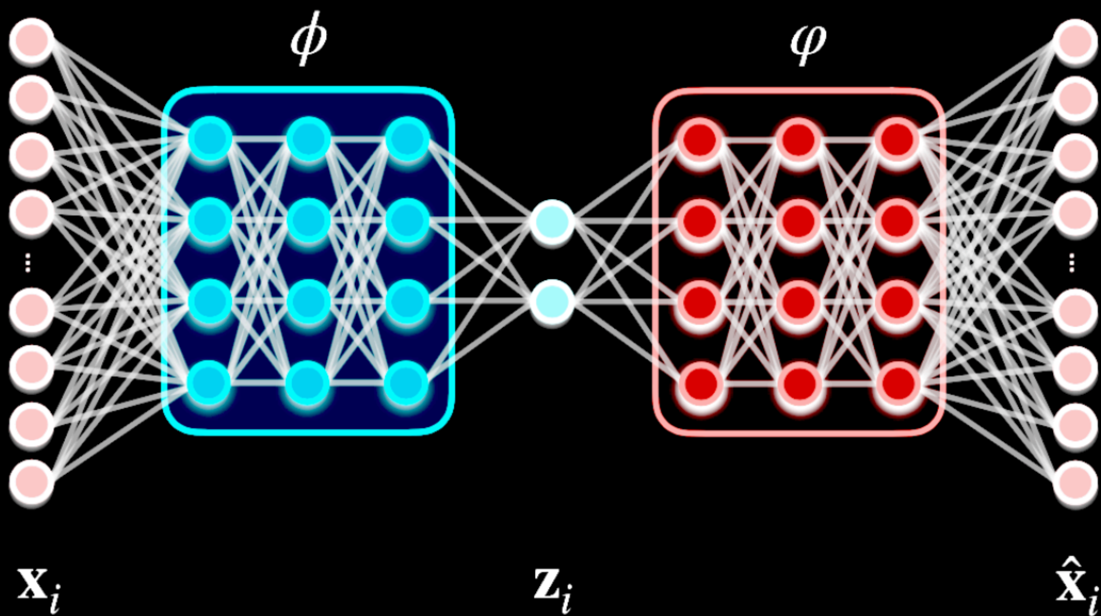


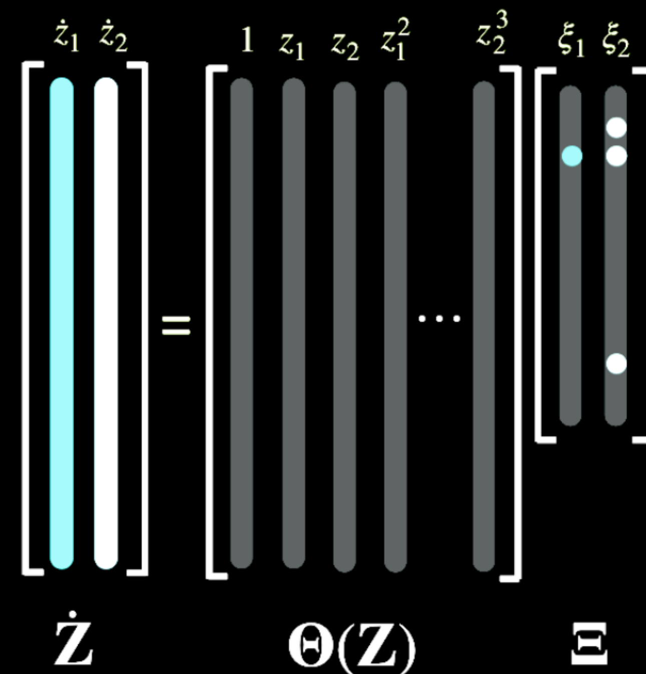


Flow Around a Cylinder



Autoencoder + SINDy





$\dot{\mathbf{z}}_i = \nabla_{\mathbf{x}}\phi(\mathbf{x}_i)\dot{\mathbf{x}}_i$ $\quad$ $\Theta(\mathbf{z}_i^T) = \Theta(\phi(\mathbf{x}_i)^T)$

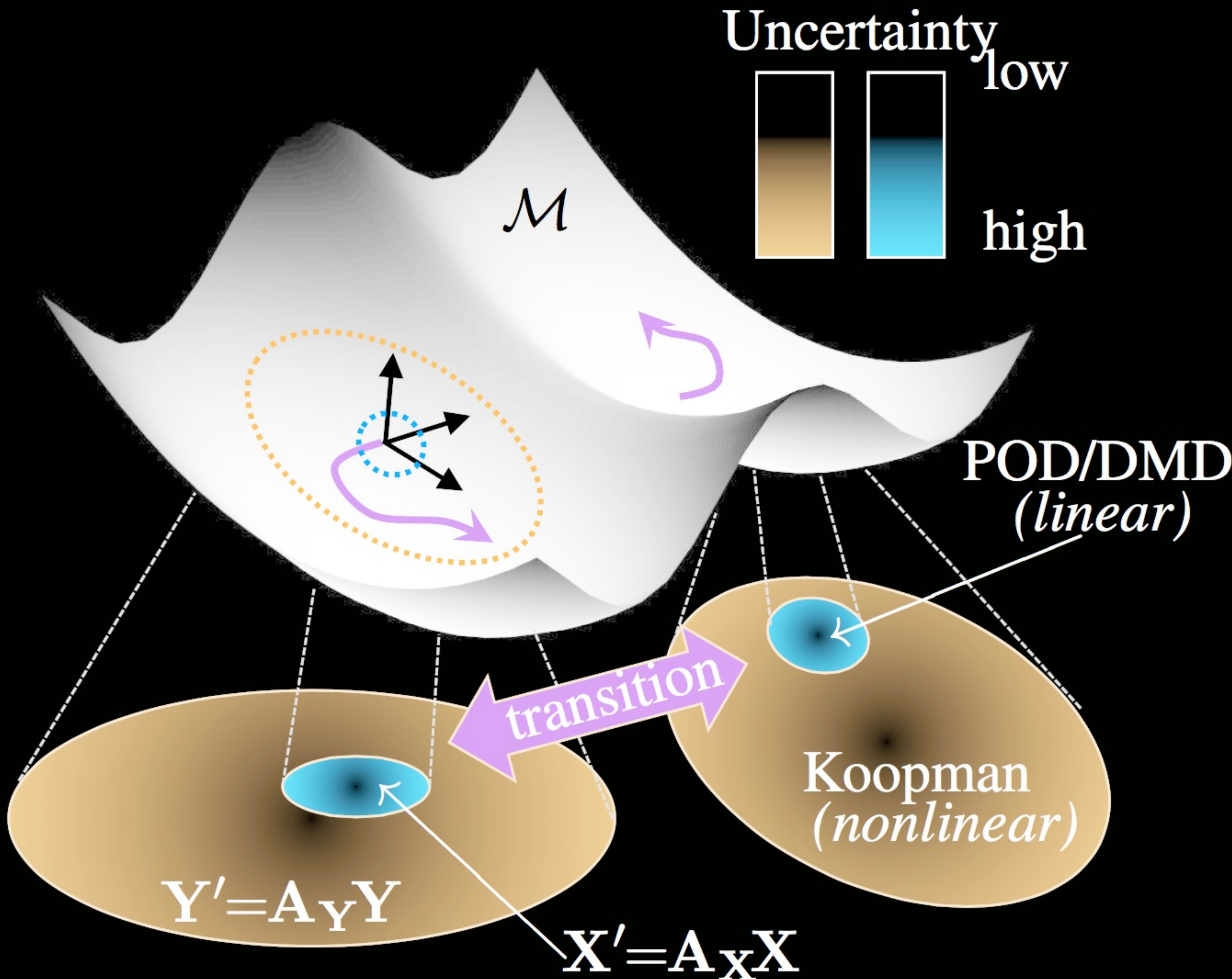
loss: $$\lambda_1 \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \varphi(\phi(\mathbf{x}_i))\|_2^2 \; + \; \lambda_2 \frac{1}{N}\sum_{i=1}^{N} \|\nabla_{\mathbf{x}}\phi(\mathbf{x}_i)\dot{\mathbf{x}}_i - \Theta(\phi(\mathbf{x}_i)^T)\Xi\|_2^2$$

autoencoder component

SINDy component

Kathleen Champion

Koopman and UQ



Part 3

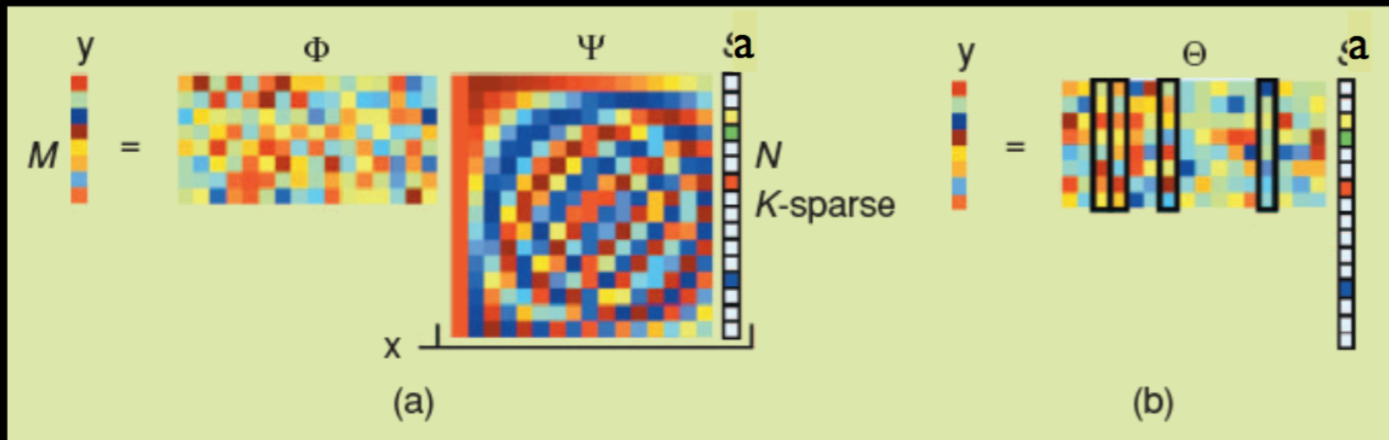
Measurement and Sensors

Randomized Linear Algebra

&

Matrix Sketching

Compressive Sensing: A Cartoon



from Baraniuk, 2007.

IN: low-dimensional measurements

random measurement matrix

OUT: sparse coefficients

$$y = \Phi x = \Phi\Psi a = \Theta a$$

transform basis

IMPORTANT: measurement matrix must be incoherent with respect to the transform basis

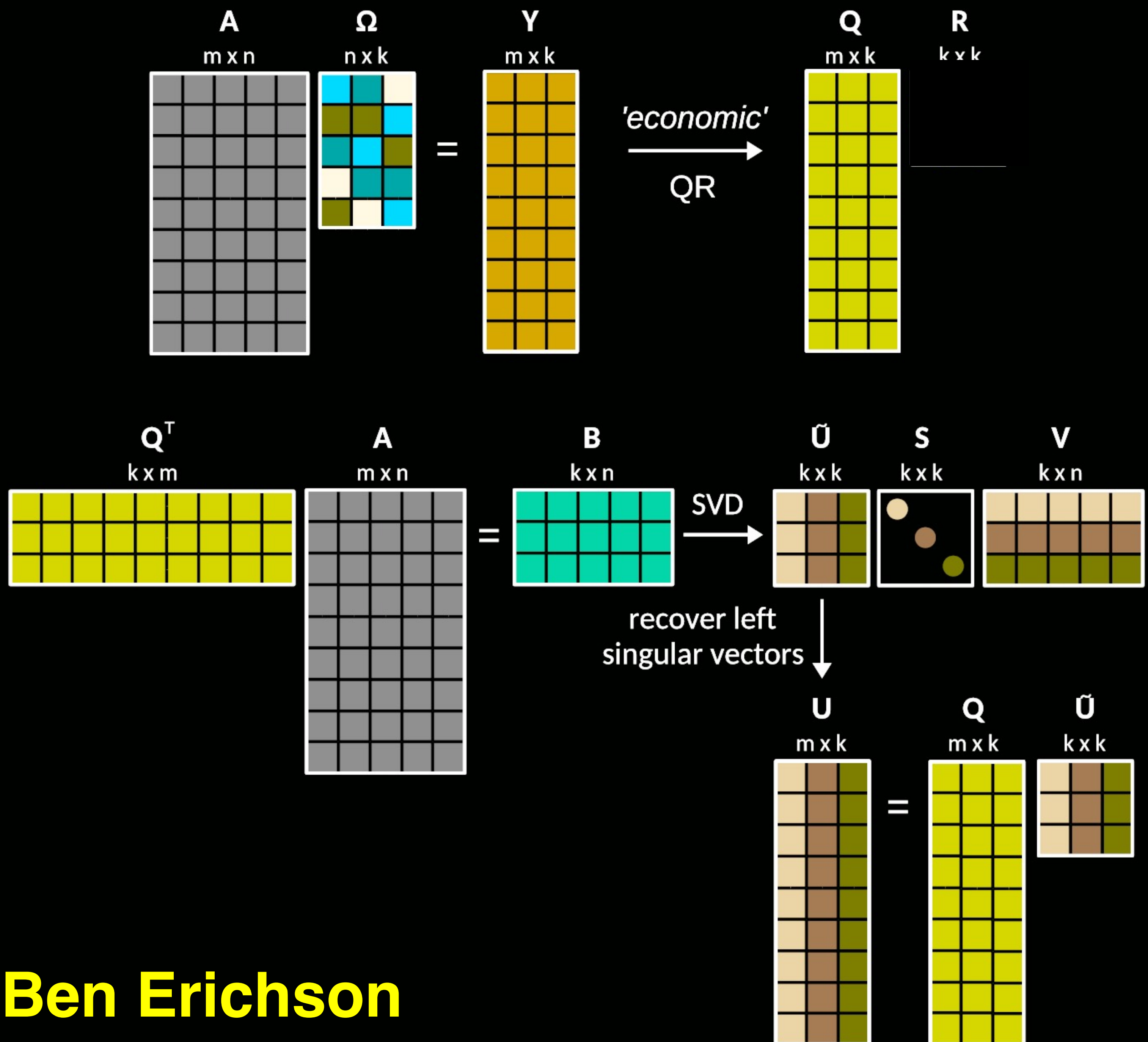
To reconstruct:

$$\text{minimize } ||a||_1,$$
$$\text{such that } y = \Theta a$$

Proofs by:

- Candès, Romberg & Tao, 2006.
- Donoho, 2006.

Randomized Linear Algebra



Ben Erichson

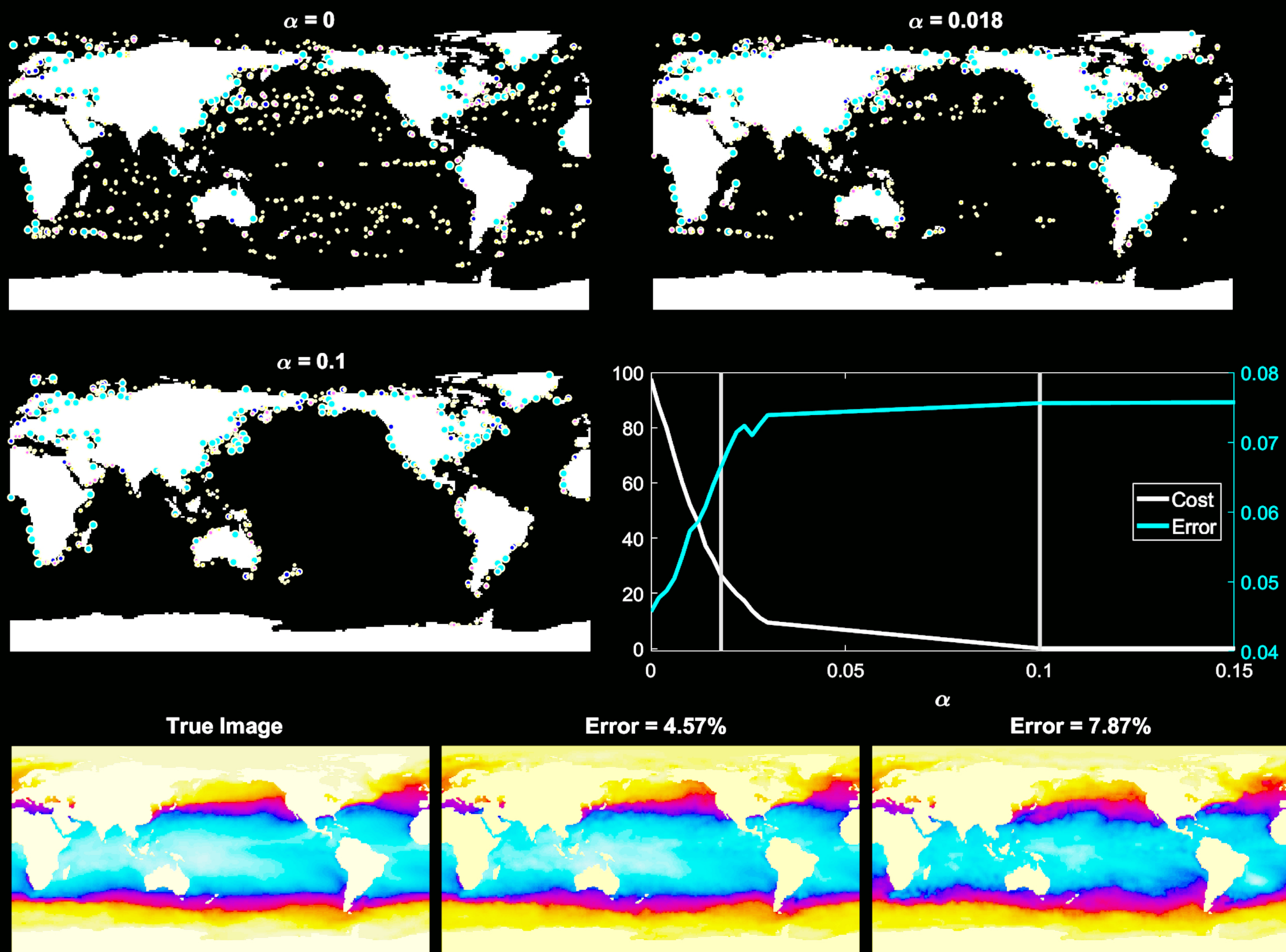
Gappy Methods

```
% QR sensor selection, p=K
[Q,R,pivot] = qr(Psi_r','vector');
pivot = pivot(1:K)
```

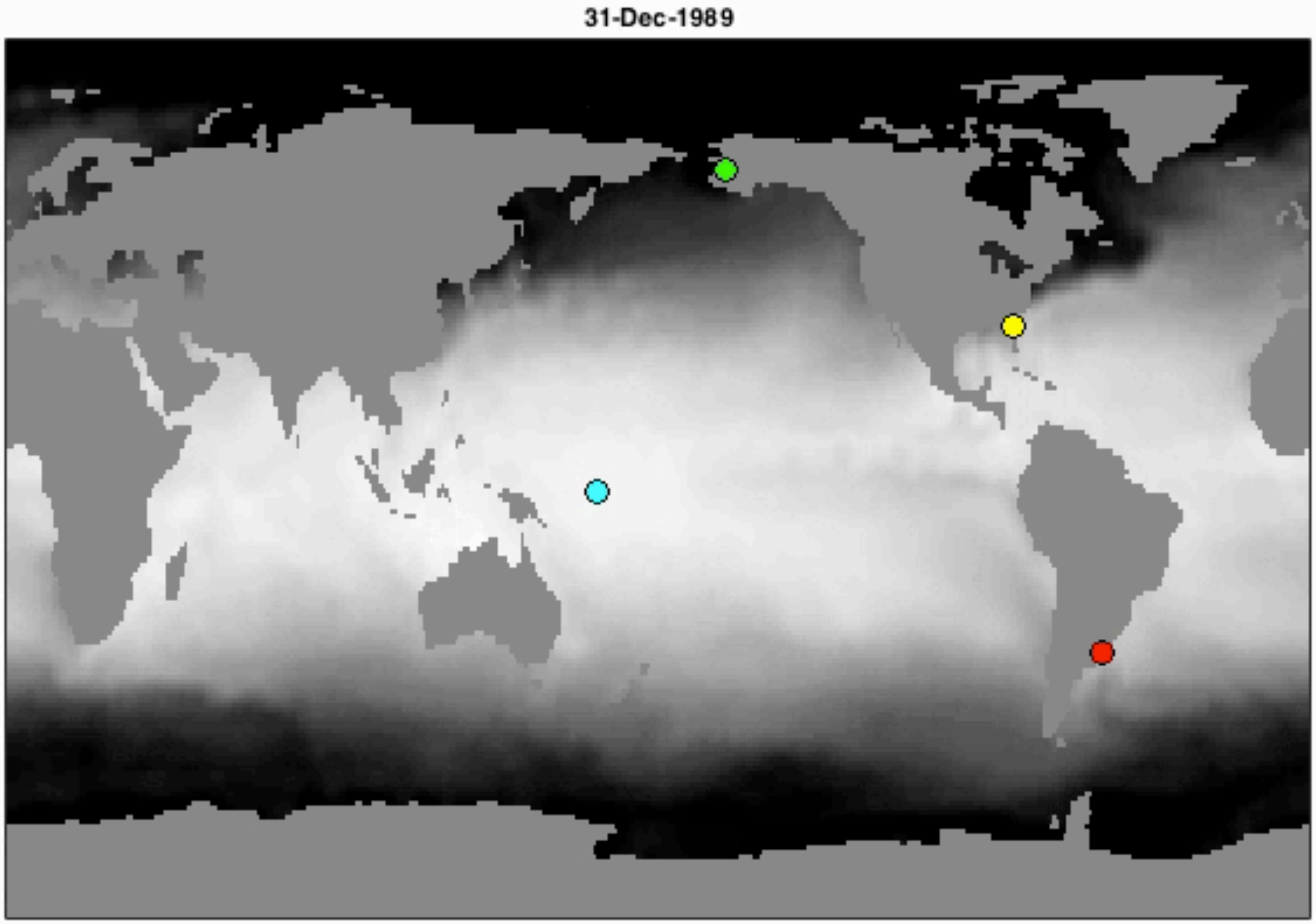
Everson & Sirovich (1995)
Willcox (2005)
Karniadakis co-workers (2009)
Maday, Patera et al & Sorenson et al (2010, 2012)
Gugerkin & Drmac (2015)
Manohar, Brunton, Kutz & Brunton (2017)

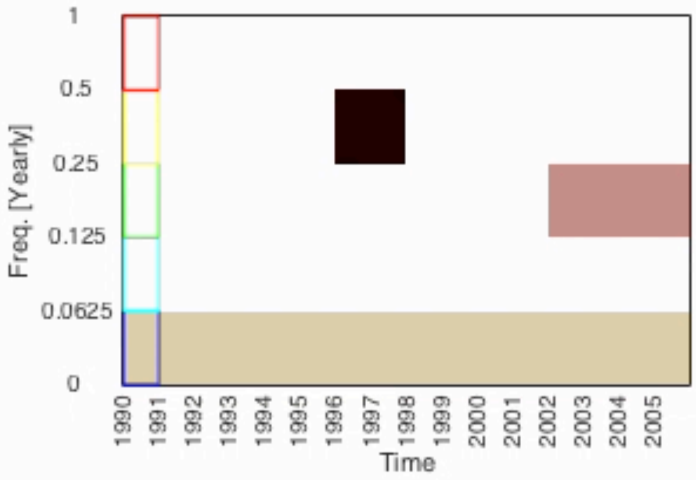
Krithika Manohar

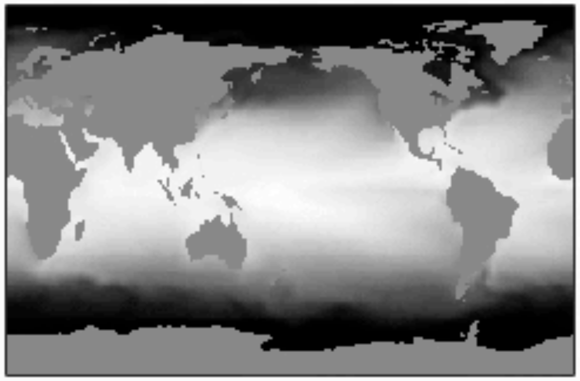
Cost with Sensors



Respect Multiscale Features



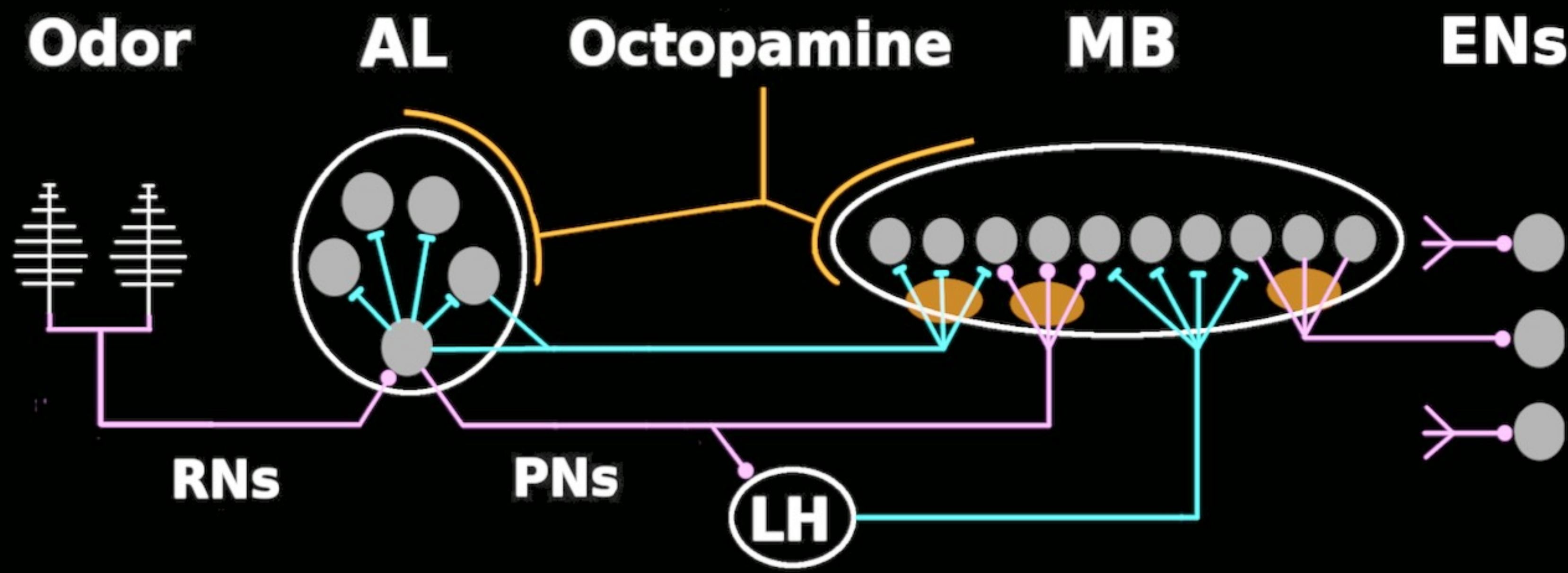




Fast Learning

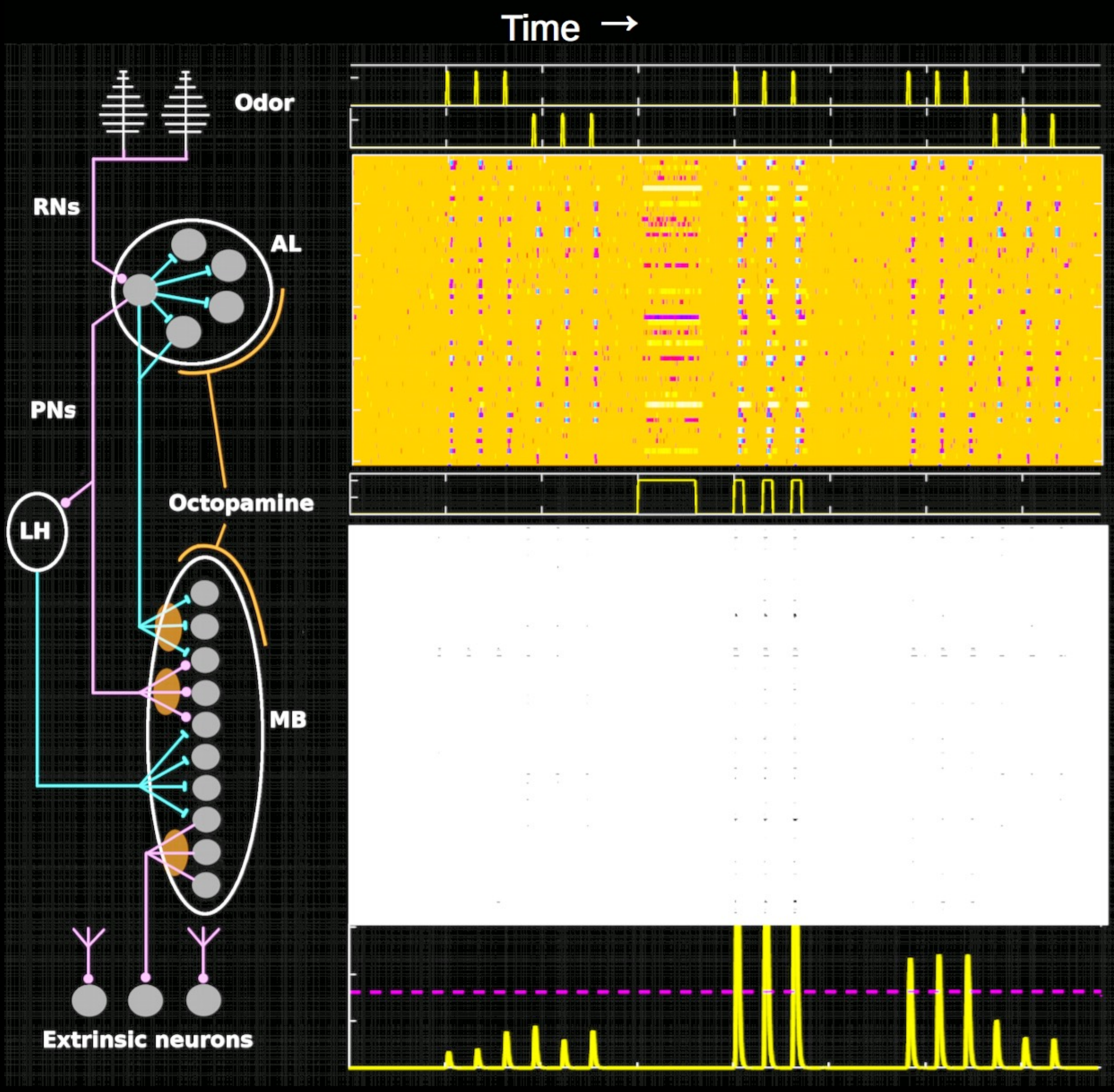
Charles Delahunt

Moth Olfactory System



Riffell et al. Science 2013
Campbell et al. J Neuro 2013
Olson et al. Neuron 2010
Turner et al. J NeuroPhysiol 2008 Hong, Wilson. Neuron 2015
Gupta, Stopfer. J NeuroSci 2012
Silbering et al. J NeuroSci 2003
Galizia. Eur J NeuroSci 2014
Caron et al. Nature 2013

Learning New Odors



Odor inputs

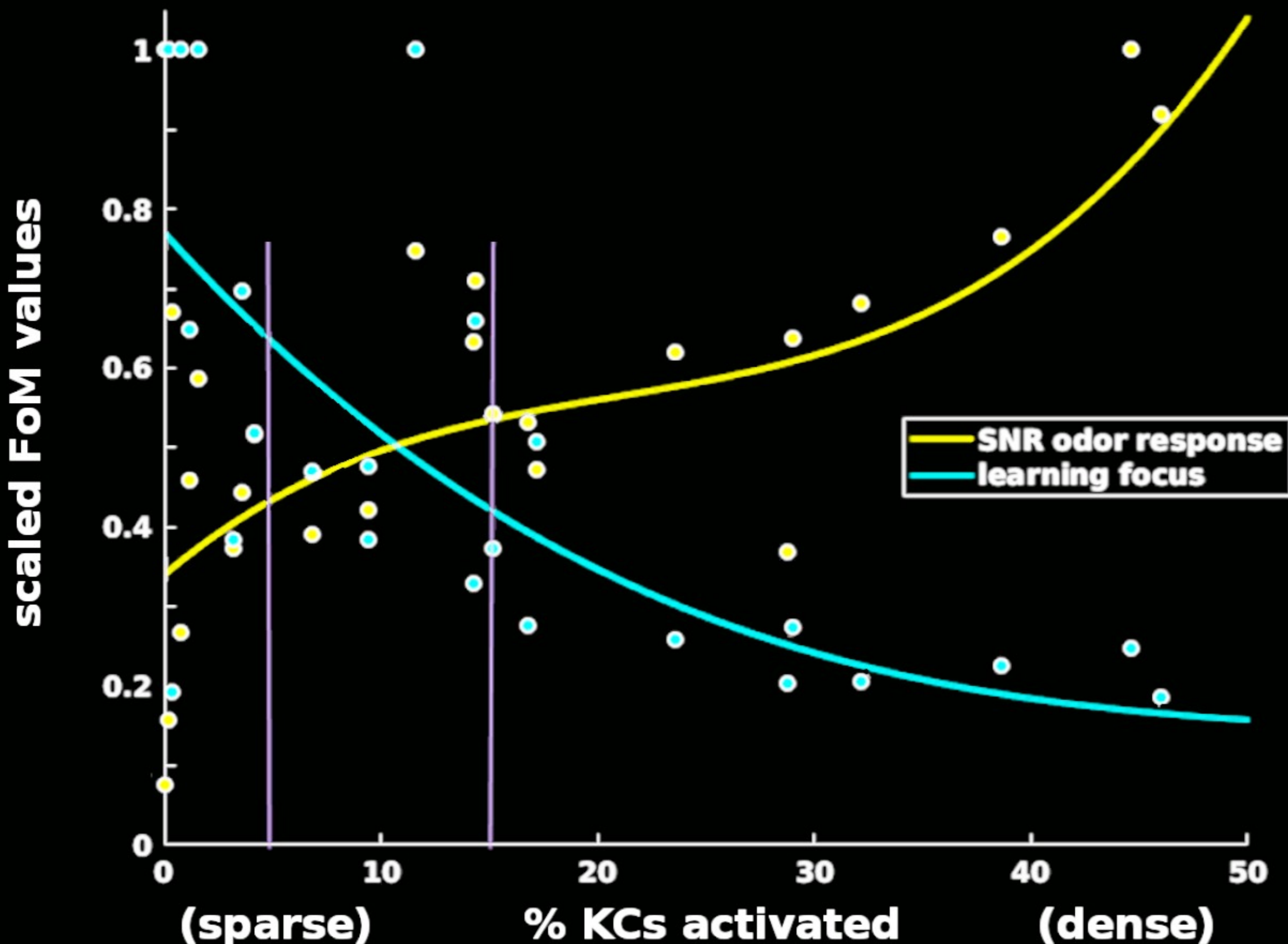
AL response

Octopamine

MB response

Readout neuron

Sparsity for Learning



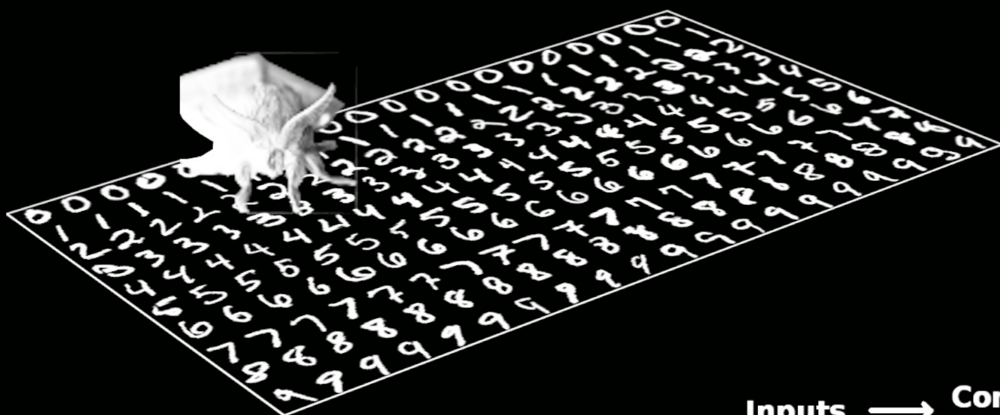
Signal to Noise = μ/σ of odor response.
high → reliable response.

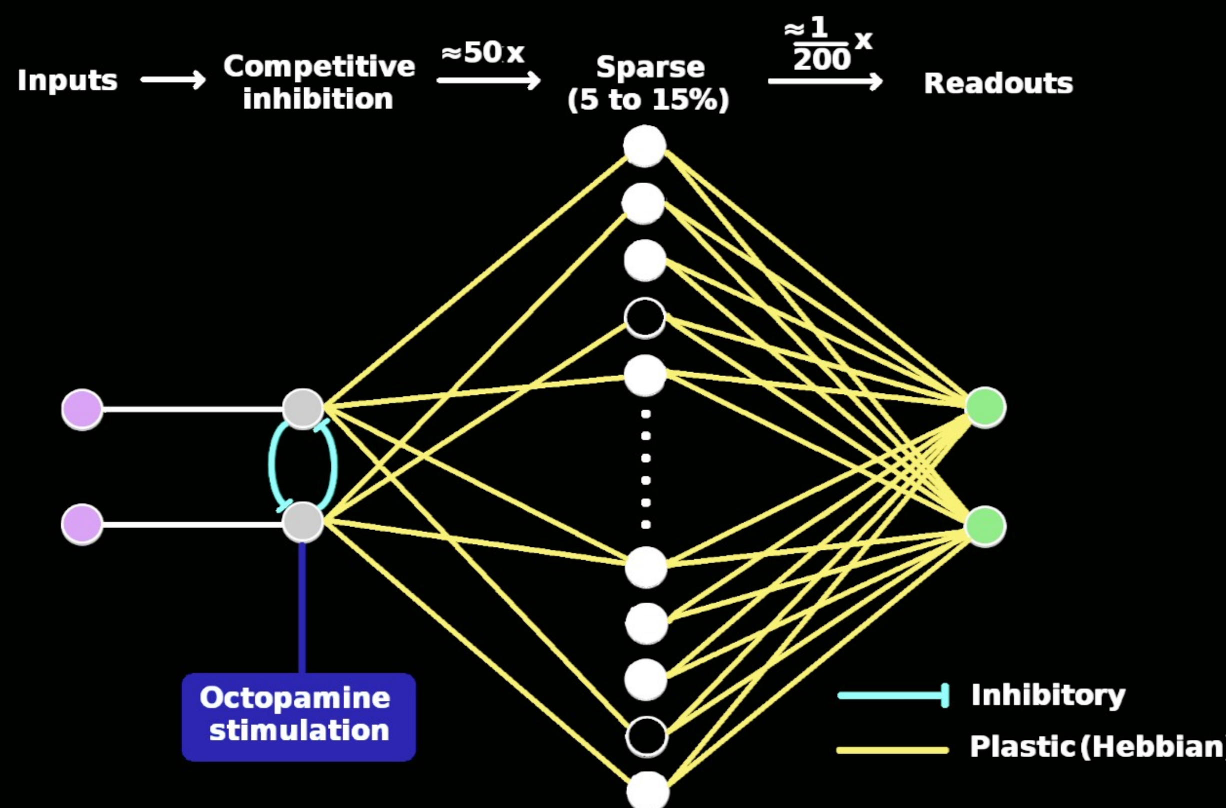
High → good.

Learning focus = ΔTrained / ΔControl.
high → focused learning.

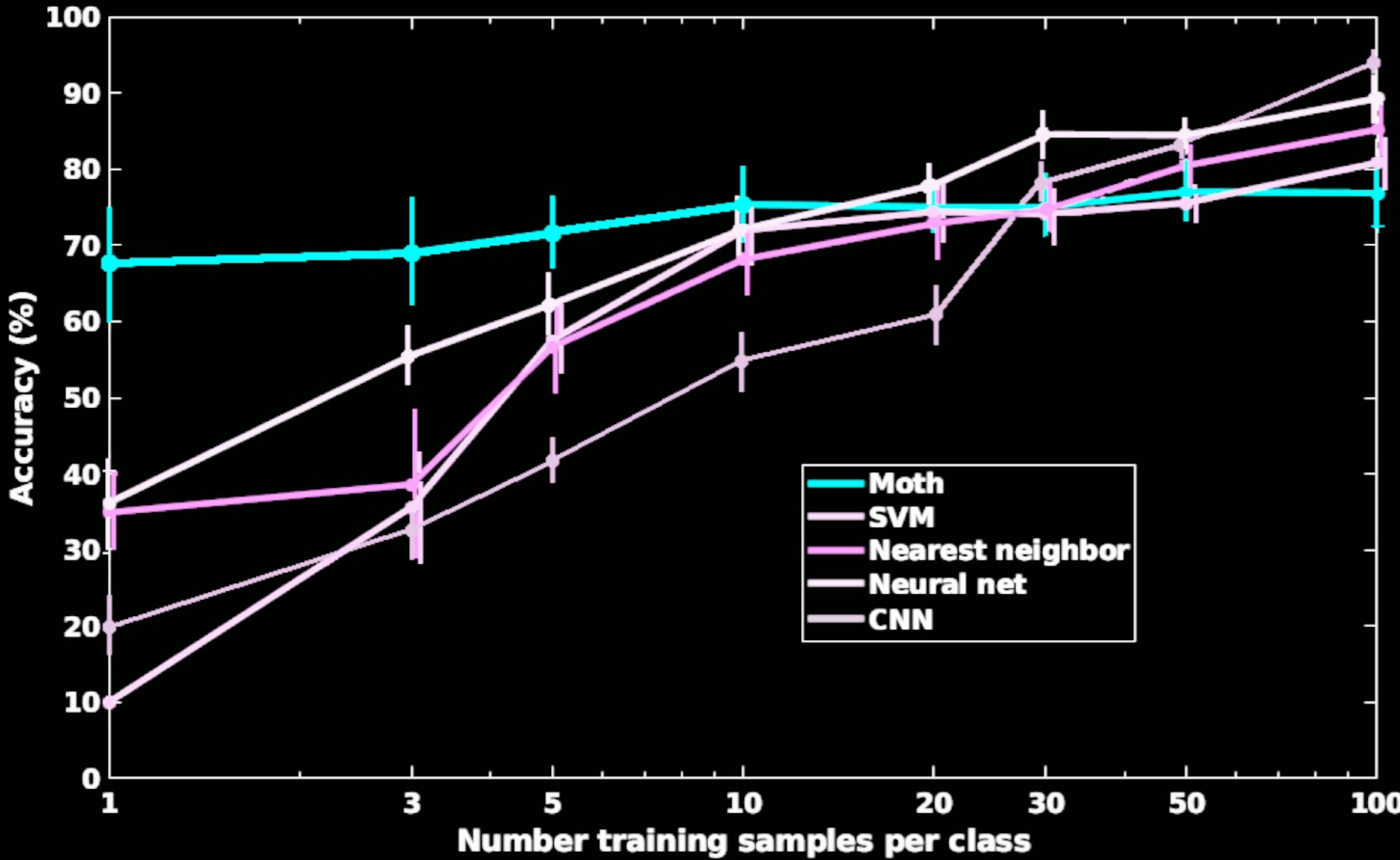
(Huerta, Nowotny; Peng, Chittka)

Rapid Learning in NNs





Comparisons



Part 4

Integration

A Diversity of Strategies

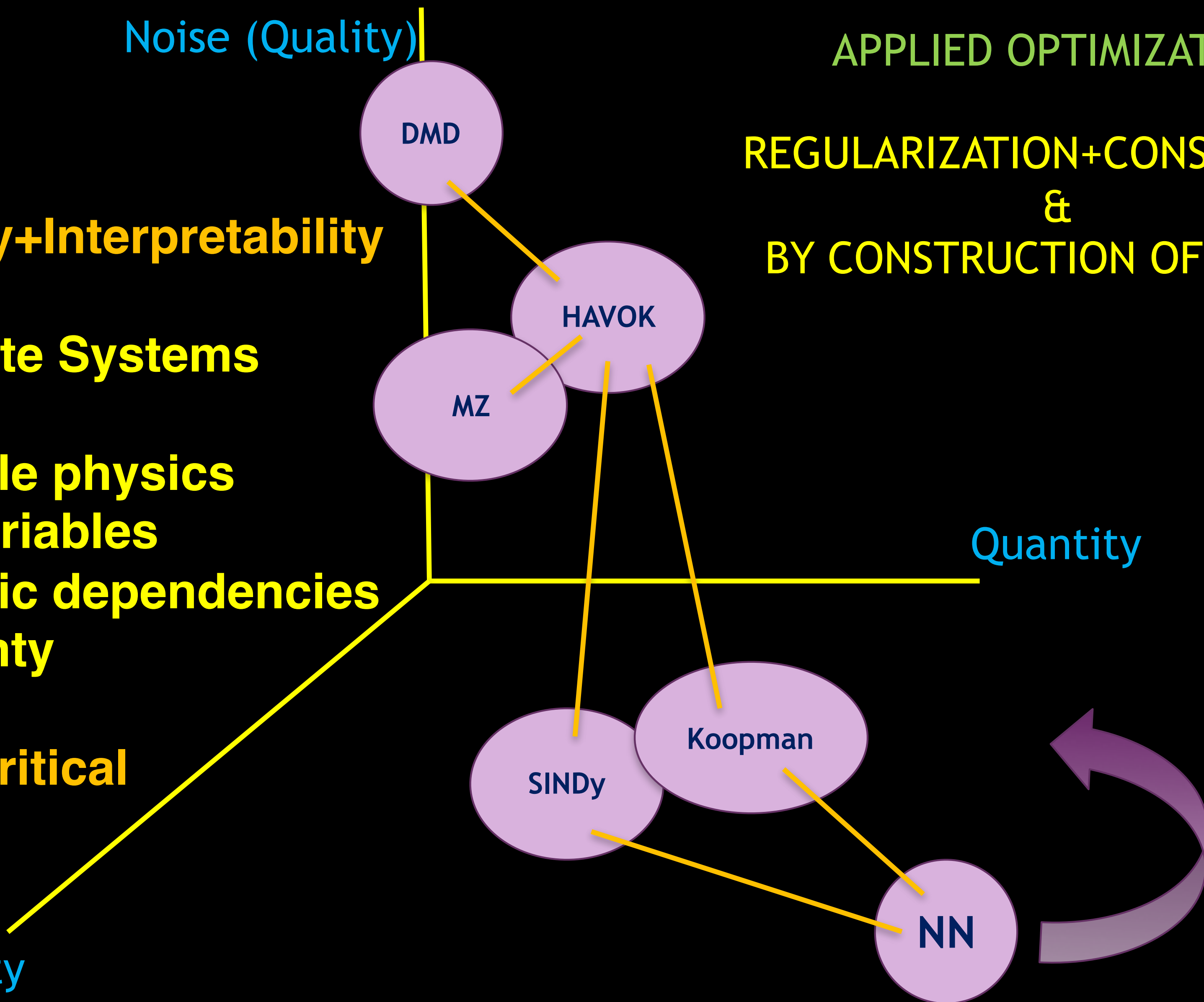
Generalizability+Interpretability

- **Coordinate Systems**
- **Noise**
- **Multi-scale physics**
- **Latent variables**
- **Parametric dependencies**
- **Uncertainty**

Parsimony is critical

APPLIED OPTIMIZATION

REGULARIZATION+CONSTRAINTS
&
BY CONSTRUCTION OF MODEL

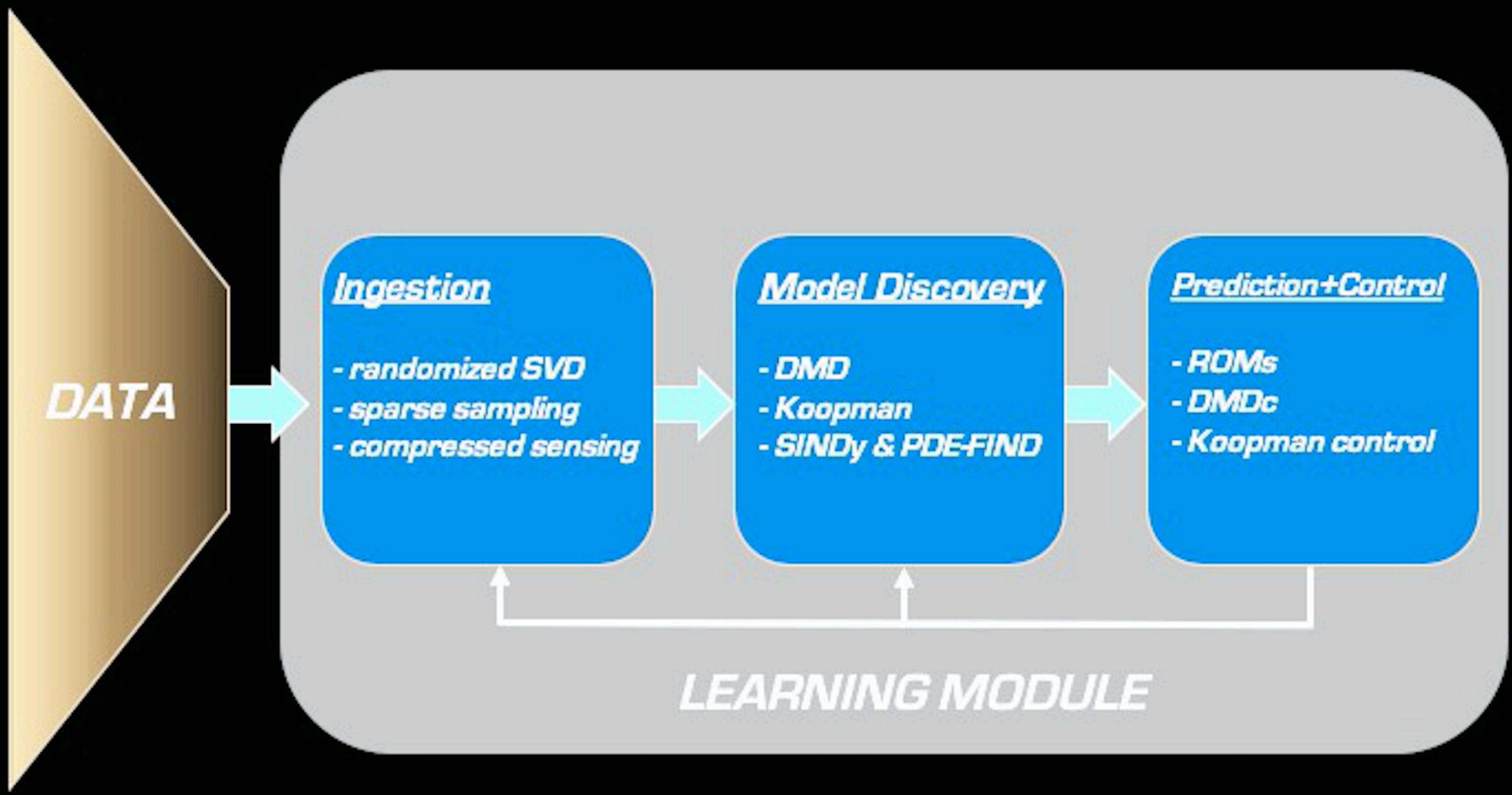


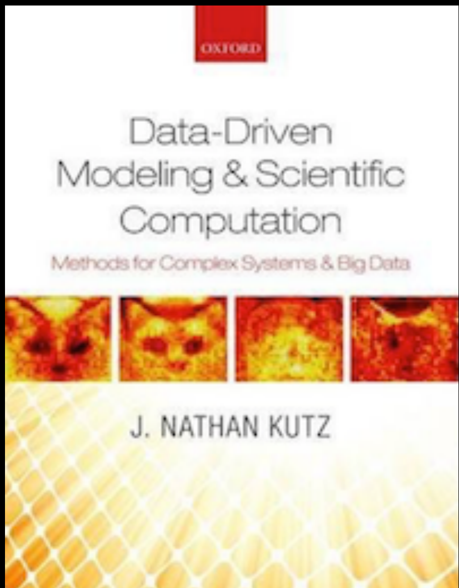
Targeted use of NN

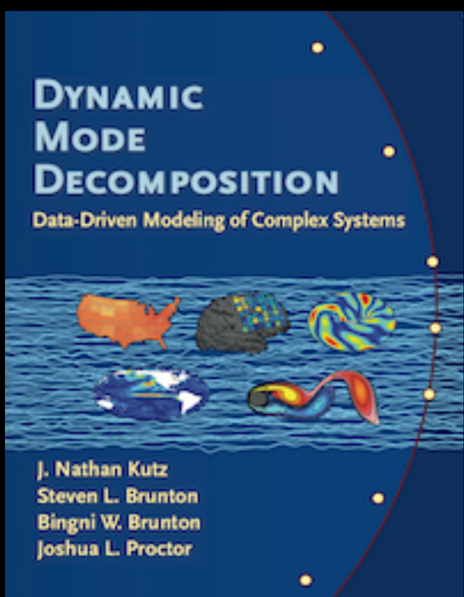
Conclusions

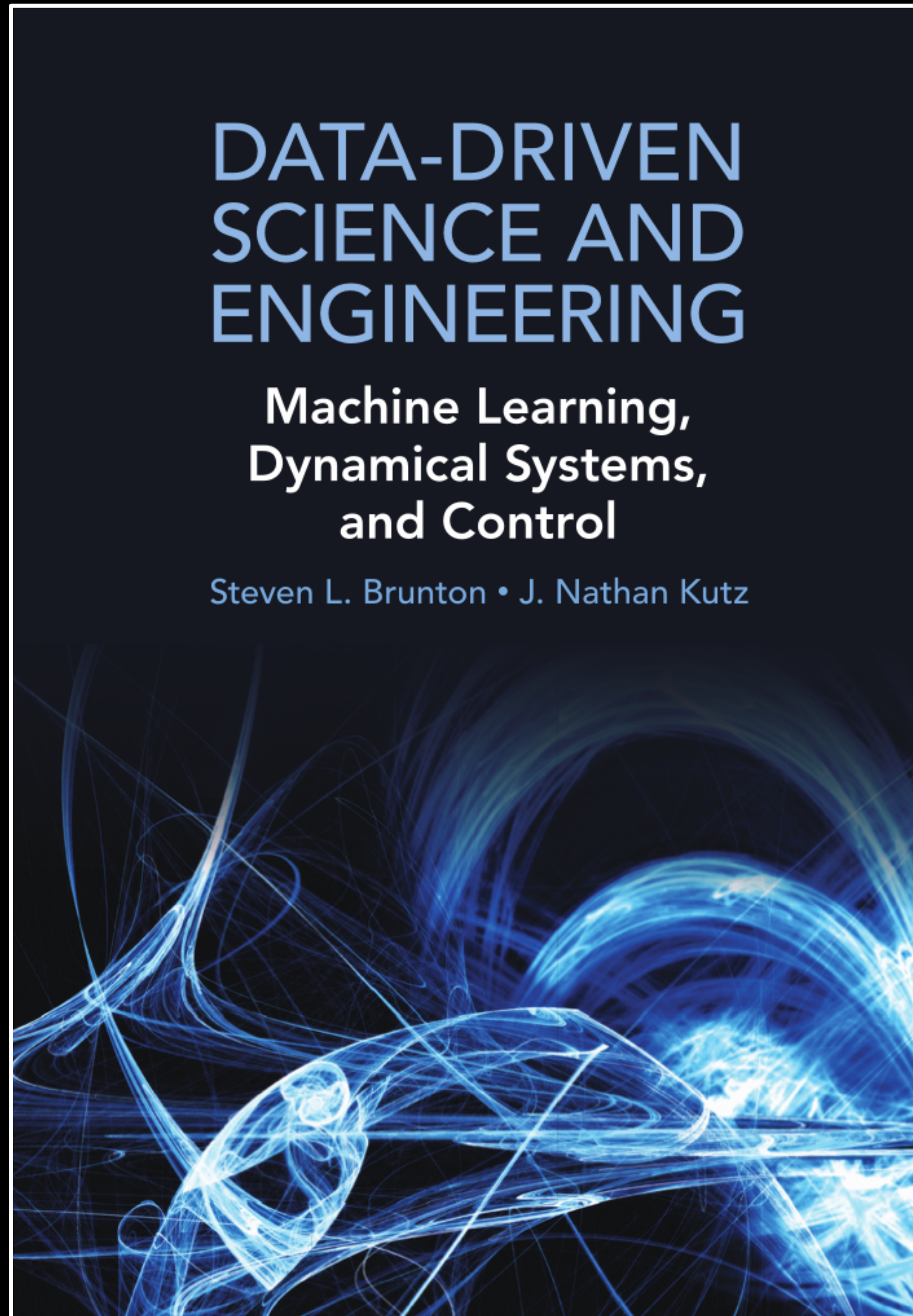
Model Selection & Sparse Regression Matter

- **Principled approach to determining dynamics**
- **(i) classification, (ii) reconstruction, (iii) future state prediction**
- **Sensors should be maximally informative**









YouTube Resources & Open Source Code

Course Structure

AMATH 563
Inferring Structure of Complex Systems

Model Selection + Background
- REGRESSION
- MODEL SELECTION
- BACKGROUND: SVD
- BACKGROUND: OPTIMIZATION
- BACKGROUND: CLASSIFICATION + CLUSTERING

Dynamics + Discovery
- MODEL DISCOVERY
- DATA ASSIMILATION
- DMD + KOOPMAN THEORY + EMBEDDINGS

Learning + Feature Extraction
- NEURAL NETWORKS + DEEP LEARNING
- RANDOMLIZED LINEAR ALGEBRA
- MULTI-RESOLUTION ANALYSIS + WAVELETS

Networks + Inference
- NETWORK CONCEPTS
- INFERENCE + CAUSALITY
- PROBABILISTIC GRAPHICAL MODELS

Inferring Structure of Complex Systems

About This Course

This webpage is designed as the primary source of lectures, notes, codes and data for AMATH 563: Inferring Structure Of Complex Systems.

Topics Covered

- Week 1: Model selection and regression
- Week 2: Model discovery
- Week 3: Data Assimilation

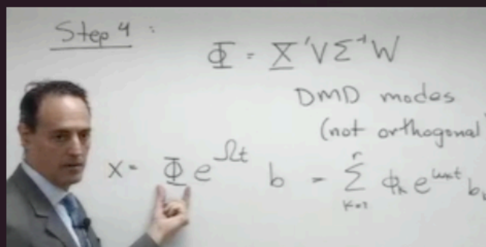
Homework + Data

Homework 1 | DATA

Homework 2 | PDECODES

Online Materials

Lecture 1



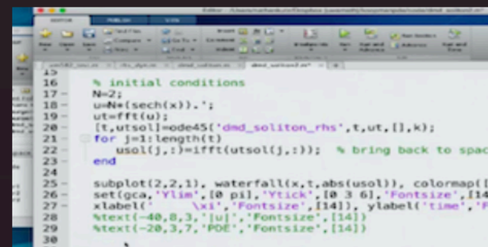
[Part 1] [Part 2]

Dynamic Mode Decomposition: This lecture provides an introduction to the Dynamic Mode Decomposition (DMD). The focus is on approximating a nonlinear dynamical system with a linear system.

MATLAB CODE

- dmd_intro.m

Lecture 2



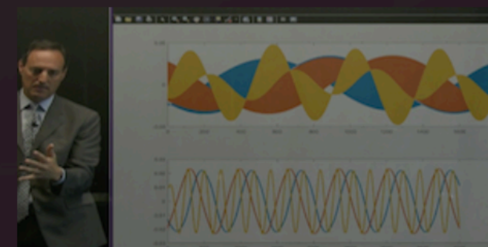
[view]

Koopman Theory: This lecture generalizes the DMD method to a function of the state-space, thus potentially providing a coordinate system that is intrinsically linear.

MATLAB CODE

- dmd_soliton.m
- dmd_soliton_rhs.m

Lecture 3



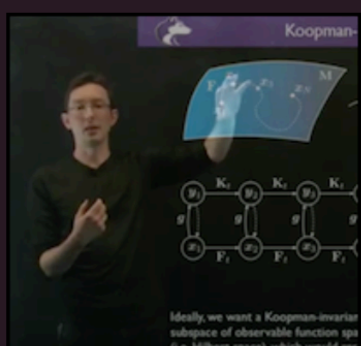
[view]

Time-Delay Embeddings: This lecture generalizes the Koopman/DMD method to a function of the state-space created by time-delay embedding of the dynamical trajectories.

MATLAB CODE

- time_delay.m
- rhs_dyn.m

KEY REFERENCES AND SUPPLEMENTARY VIDEOS



Koopman observable subspaces and finite linear representations of nonlinear dynamical systems for control

This video highlights the recent innovation of Koopman analysis for representing nonlinear systems and control.
[Part 1], [Part 2], [Part 3]

- S. L. Brunton, B. Brunton, J. L. Proctor and J. N. Kutz, **Koopman observable subspaces and finite linear representations of nonlinear dynamical systems for control** PLOS ONE (2016)

Math is Awesome