Data-Driven Discovery of Koopman Embeddings for Spatio-Temporal Systems

SIAM DS 2019

Advanced Data-Driven Techniques and Numerical Methods in Koopman Operator Theory

J. Nathan Kutz, Kathleen Champion, Bethany Lusch, Steven Brunton

Department of Applied Mathematics

University of Washington

Email: kutz@uw.edu

Web: faculty.washington.edu/kutz

arXiv (2019)

Bernard Koopman 1931

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

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

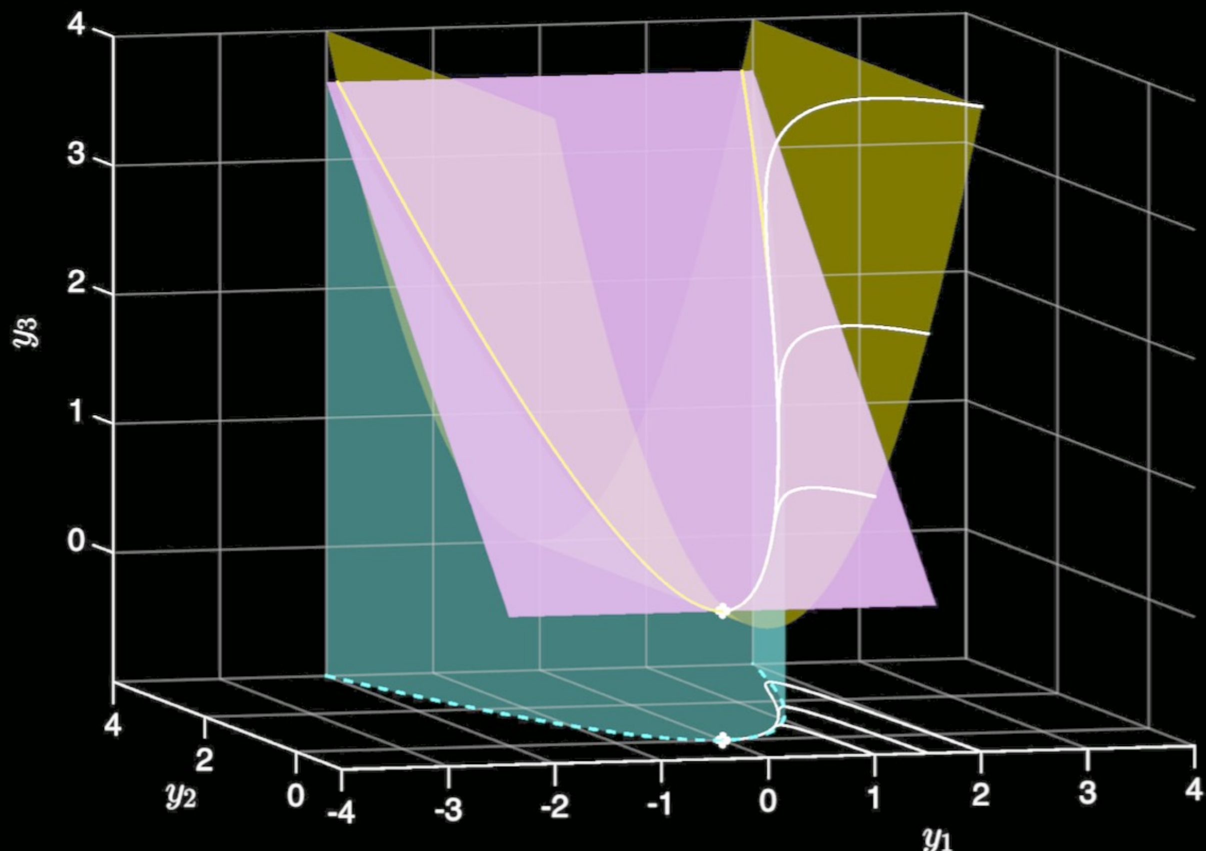
where $\mathbf{x} \in \mathbb{R}^n$ *is in a state space* $\mathbf{x} \in \mathcal{M}$. *The Koopman operator* $\mathcal{K}$ *acts on a set of scalar observable variables* g_j *which comprise the vector* $\mathbf{g} : \mathcal{M} \rightarrow \mathbb{C}$ *so that*

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

Mezic (2004), Schmid (2010), Rowley et al (2009)
Coifman, Kevrekidis, co-workers - Diffusion Maps
Williams et al - EDMD

Koopman Invariant Subspaces

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



Brunton, Proctor & Kutz, PLOS ONE (2018)

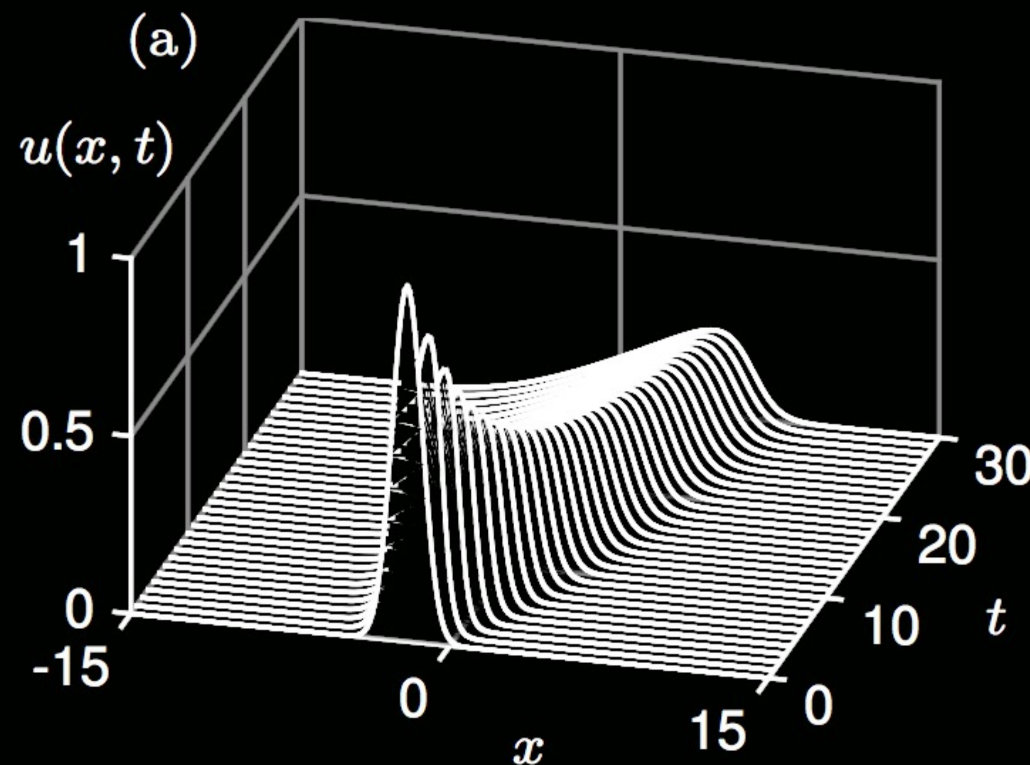
Burgers' Equation

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

Cole-Hopf

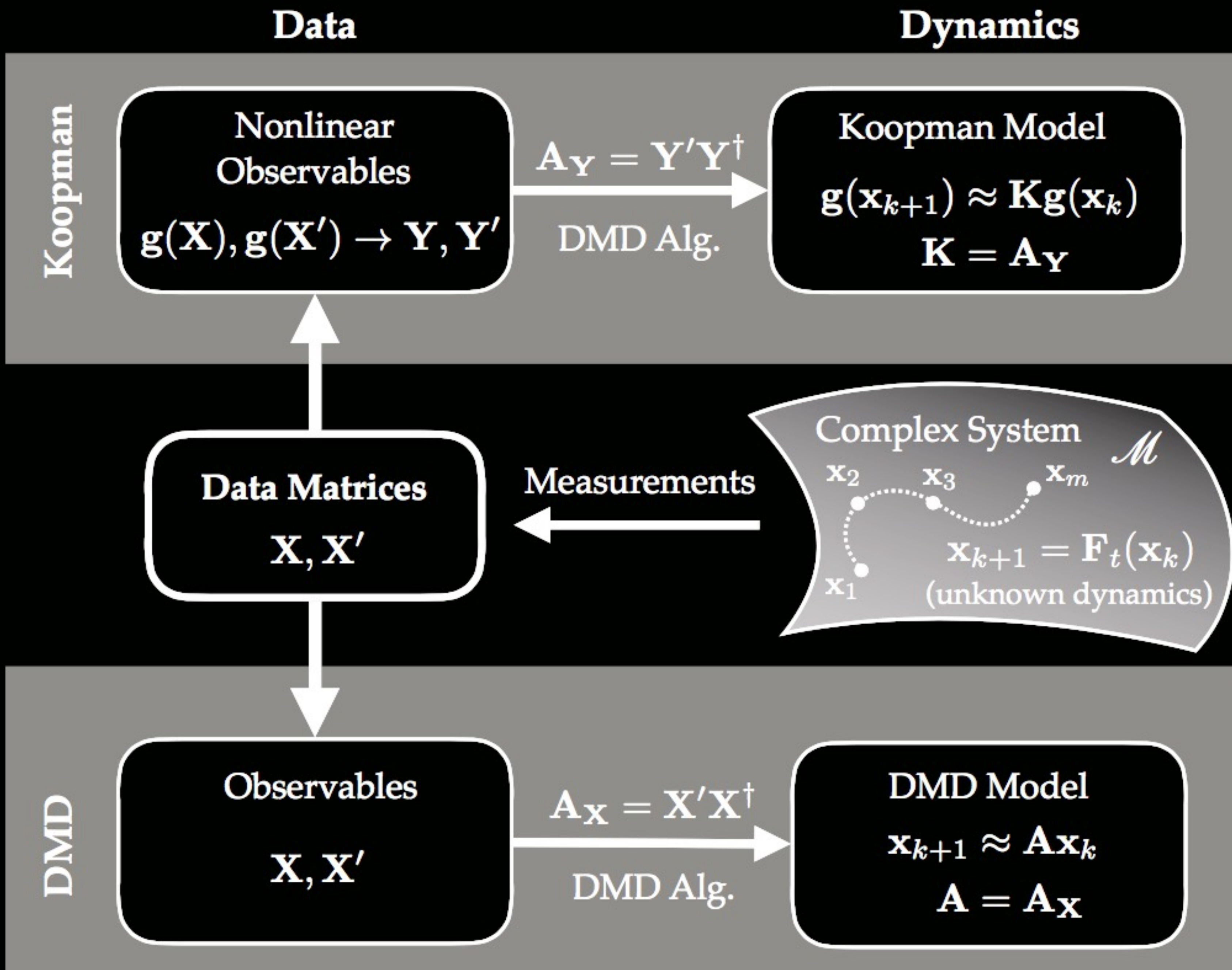
$$u = -2\epsilon v_x / v$$

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



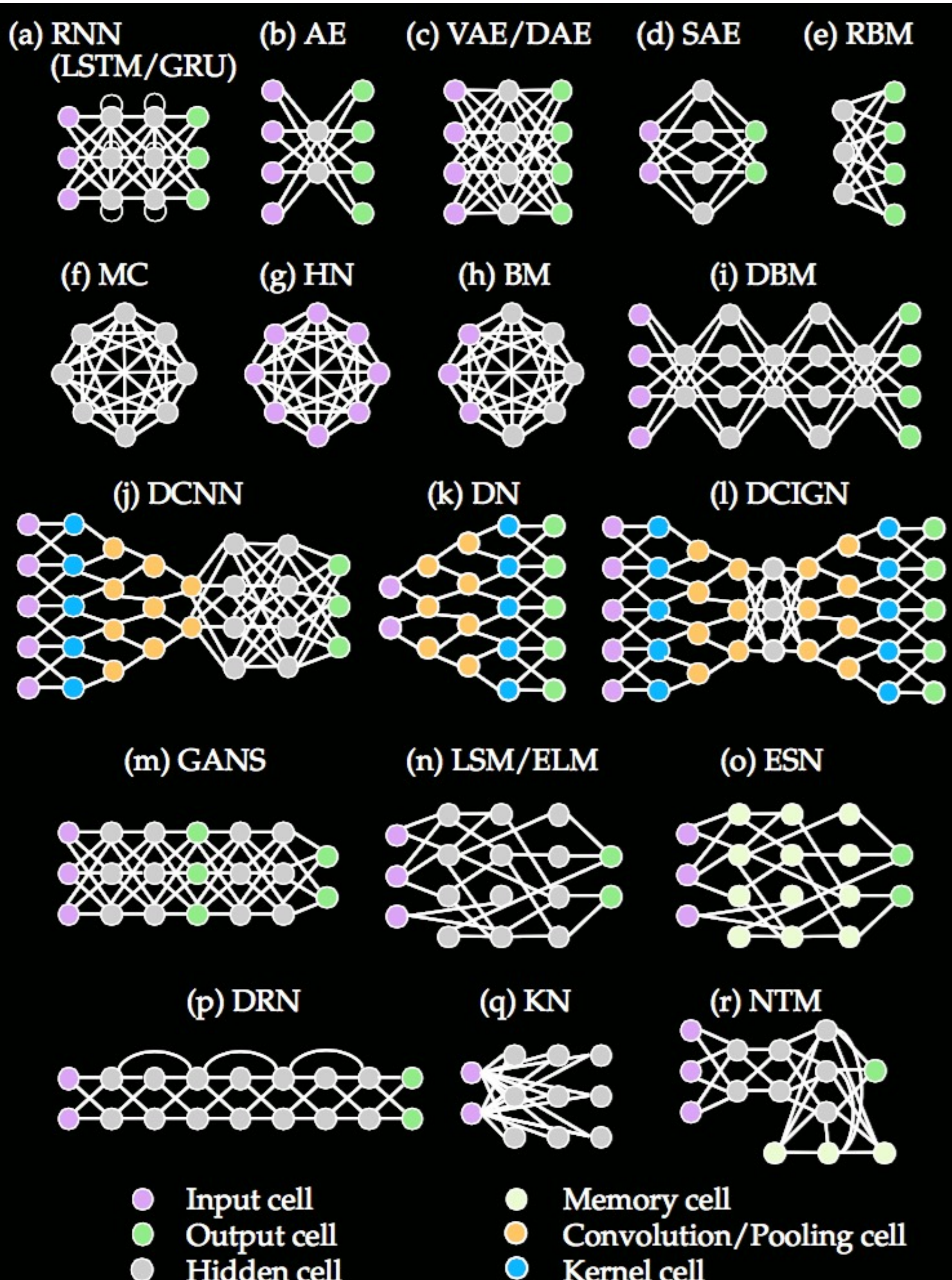
Kutz, Proctor & Brunton, Complexity (2018)

Koopman vs DMD: All about Observables!

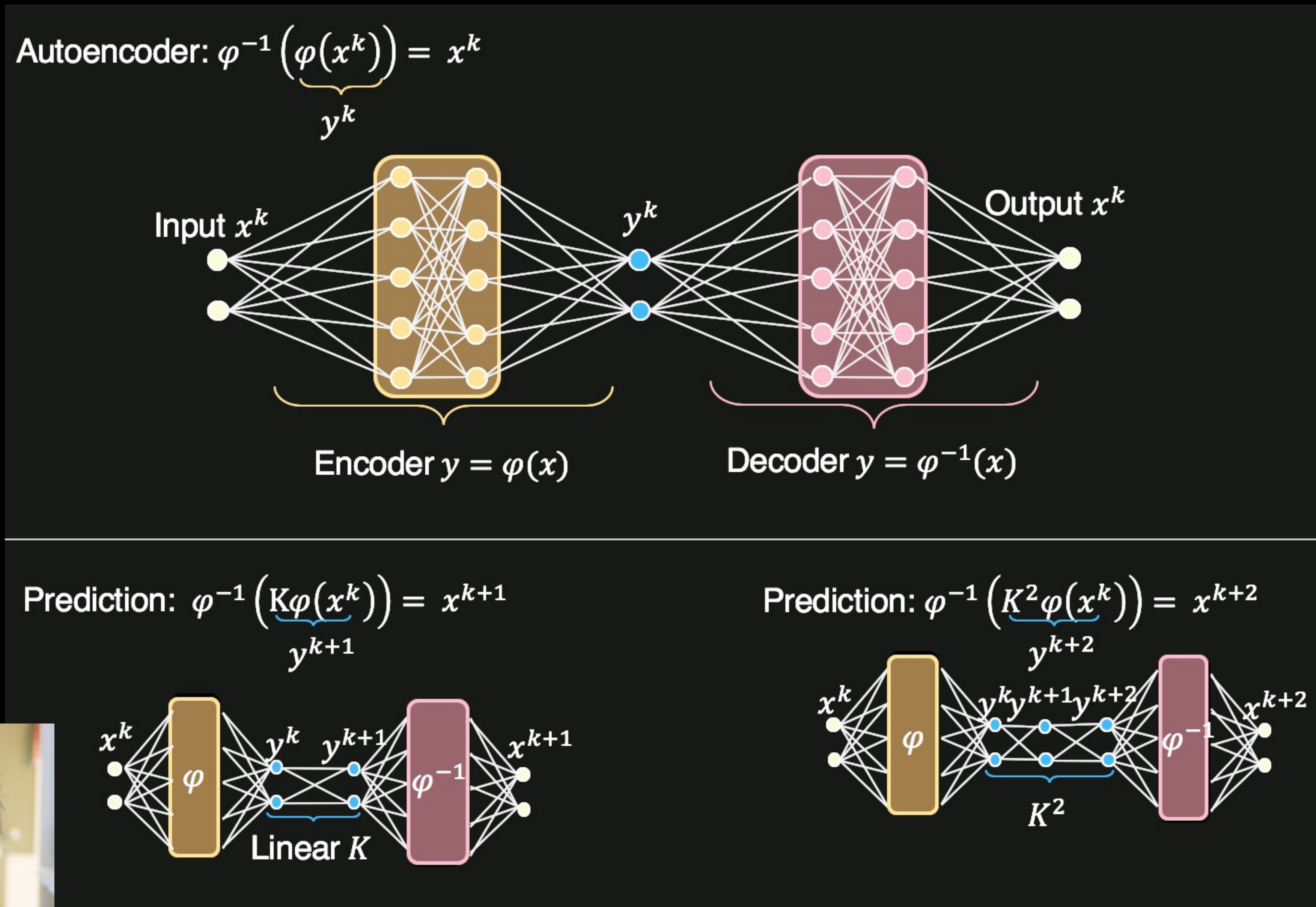


Neural Nets

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

NNs for Koopman Embedding



Bethany Lusch

Lusch et al. Nat. Comm (2018)

Failure!

(obviously)

Duffing Oscillator

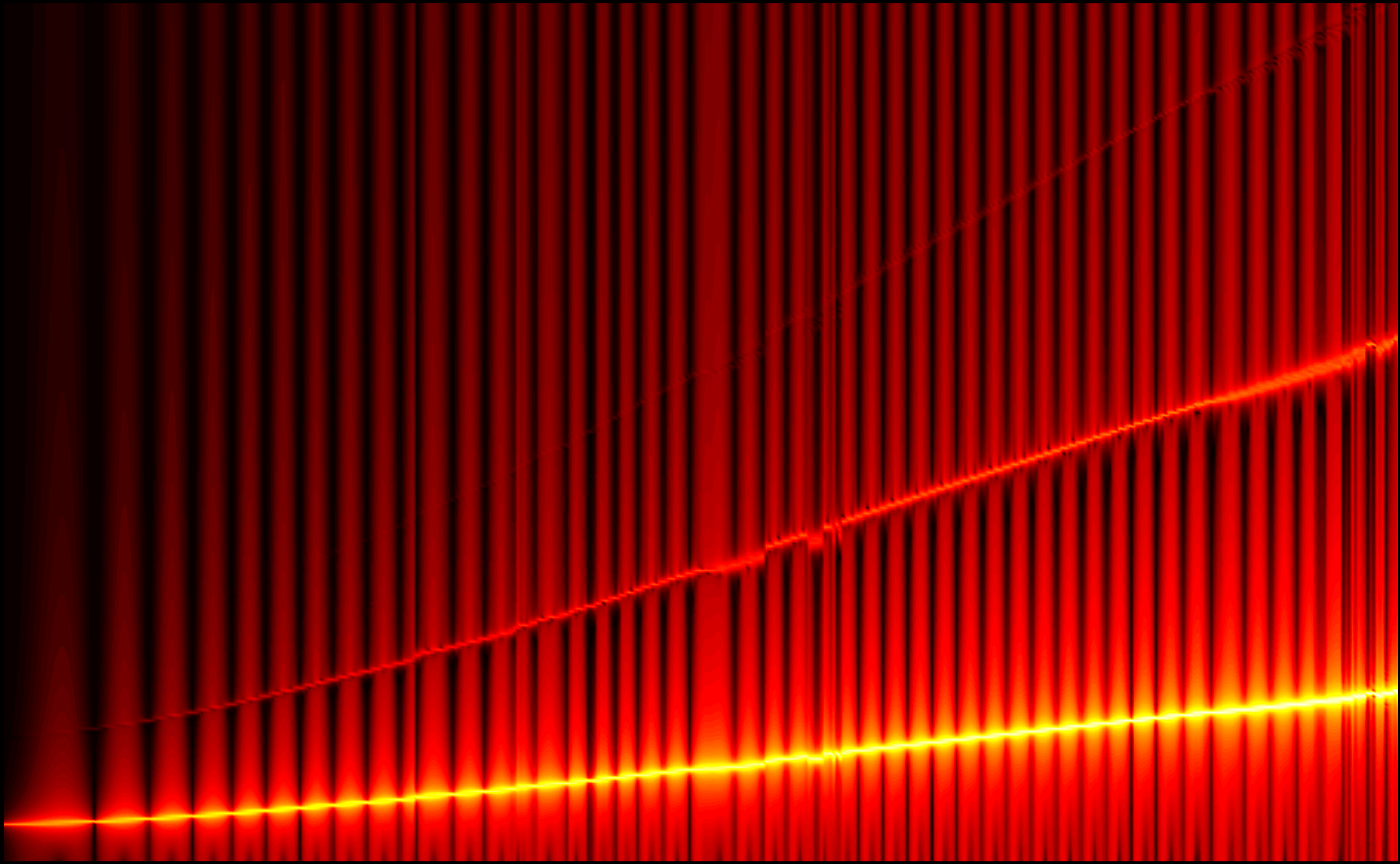
Poincare-Lindstedt Expansion: let $\tau = \omega t$ so that

$$y_{tt} + y + \epsilon y^3 = 0 \Rightarrow \omega^2 y_{\tau\tau} + y + \epsilon y^3 = 0$$

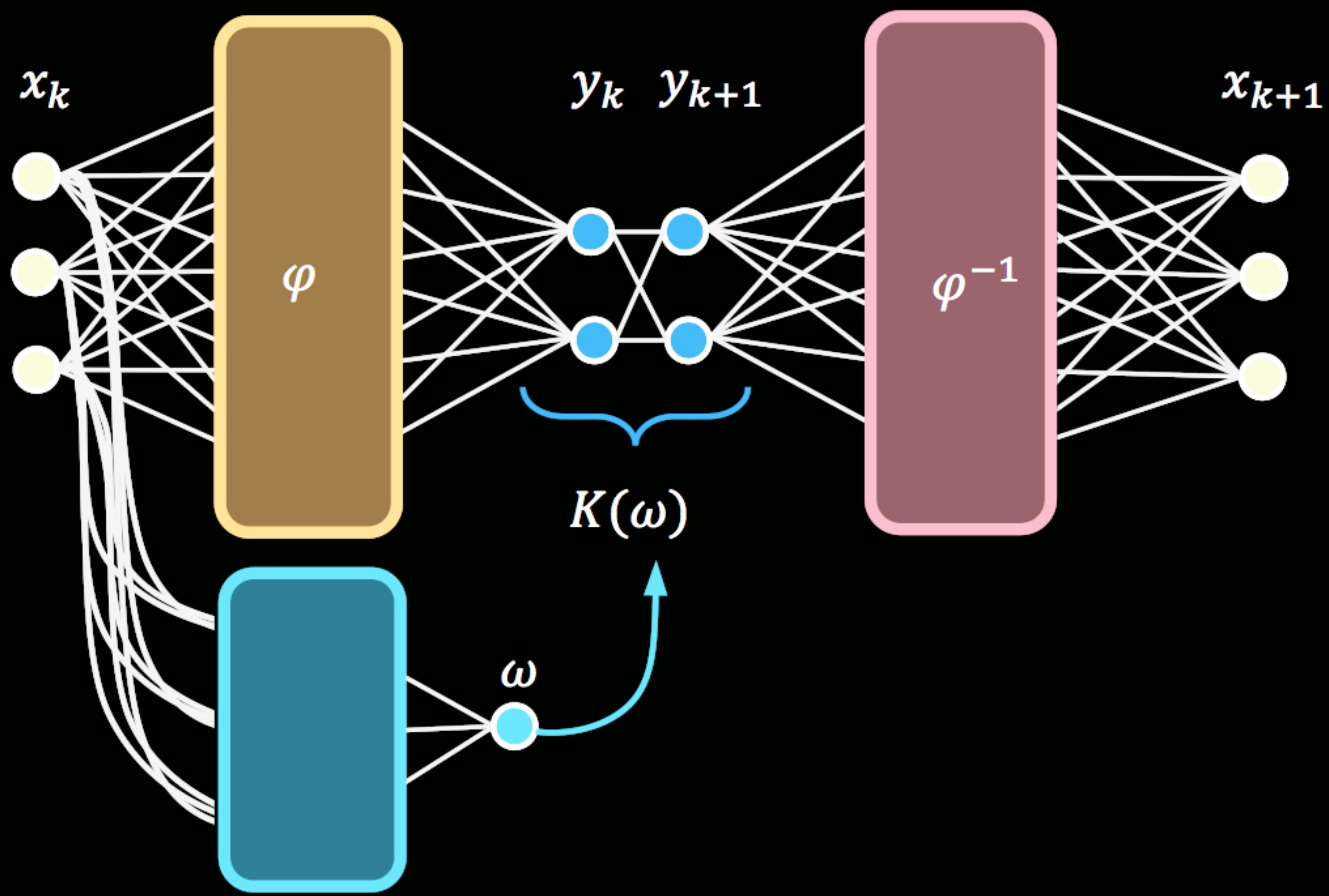
Nonlinearity: Shifts Frequencies + Generates Harmonics

$$y = A \sin[(1+\epsilon 3A^2/8)t] + \epsilon \left\{ \frac{3A^3}{32} \sin\left[\left(1+\epsilon \frac{3A^2}{8}\right)t\right] - \frac{A^3}{32} \sin\left[3\left(1+\epsilon \frac{3A^2}{8}\right)t\right] \right\}$$

Spectrogram

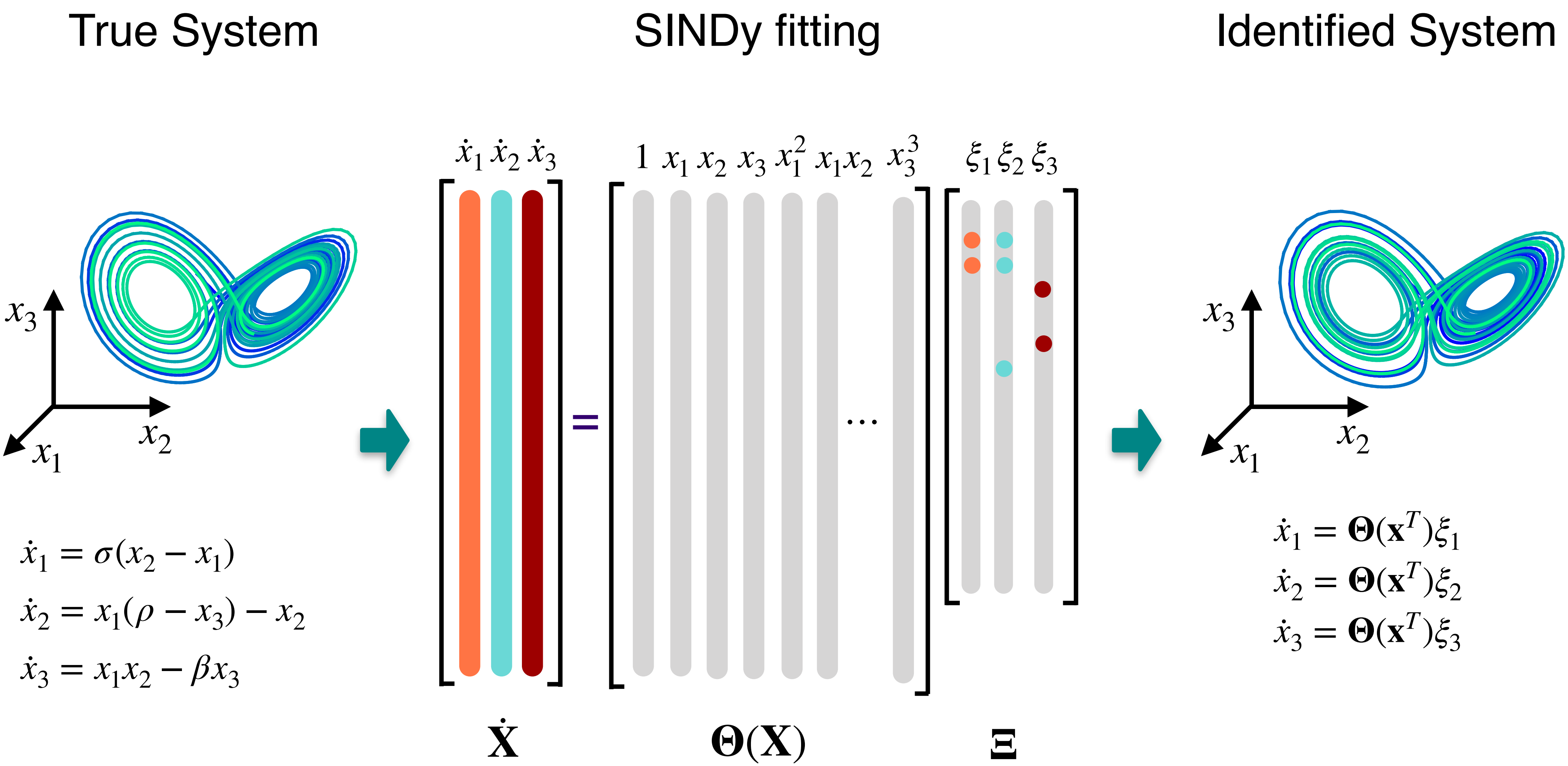


Handling the Continuous Spectra

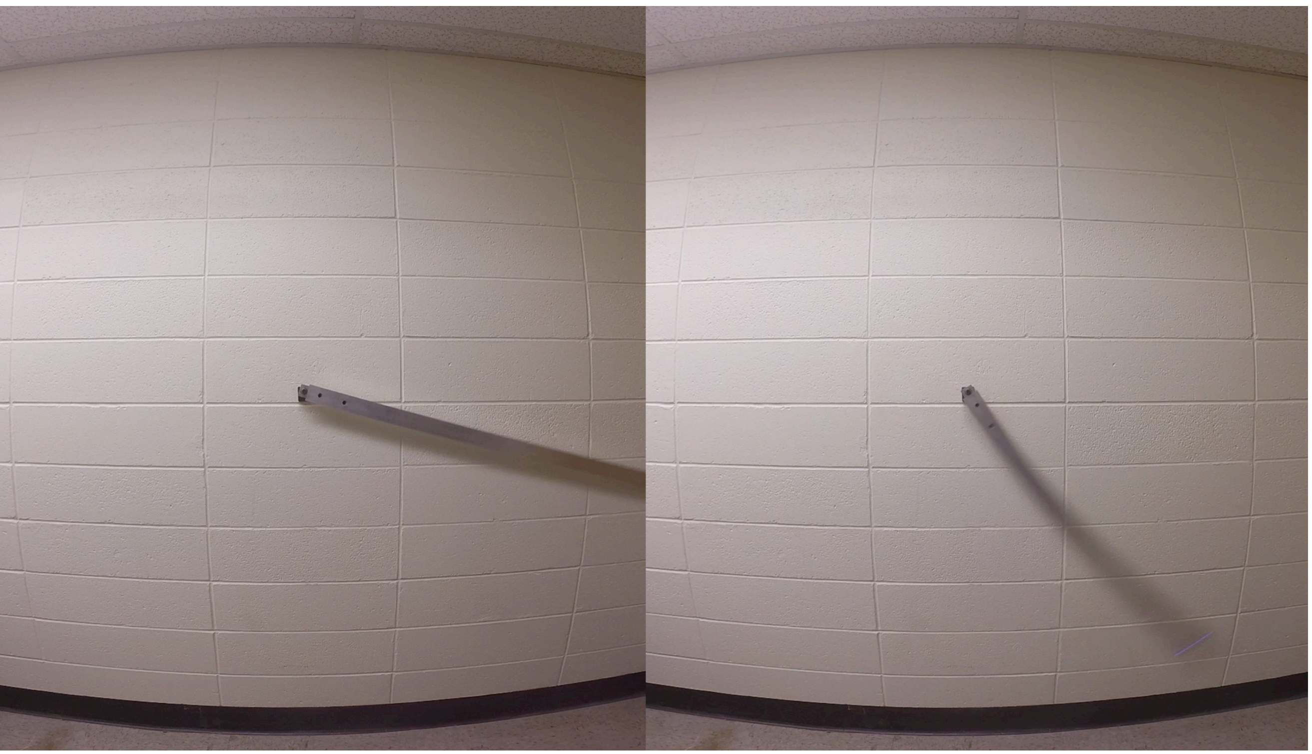


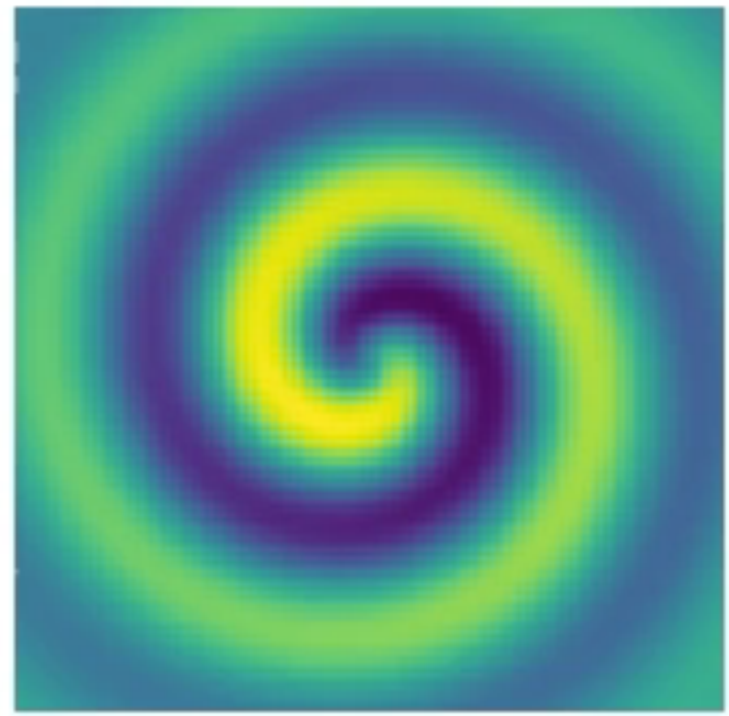
Relax Koopman

Sparse Identification of Nonlinear Dynamics (SINDy)

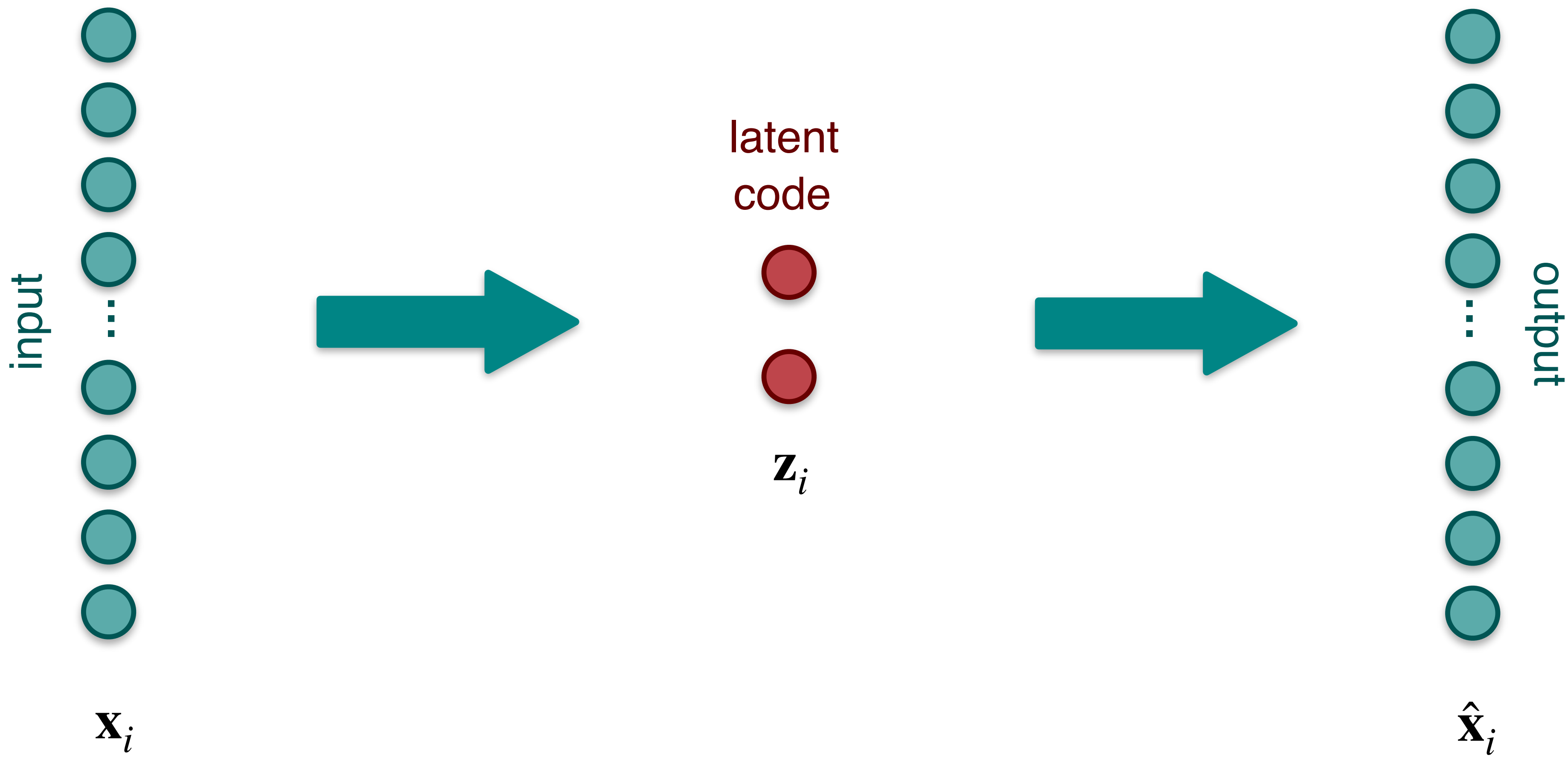


What if we don't know the right coordinates?



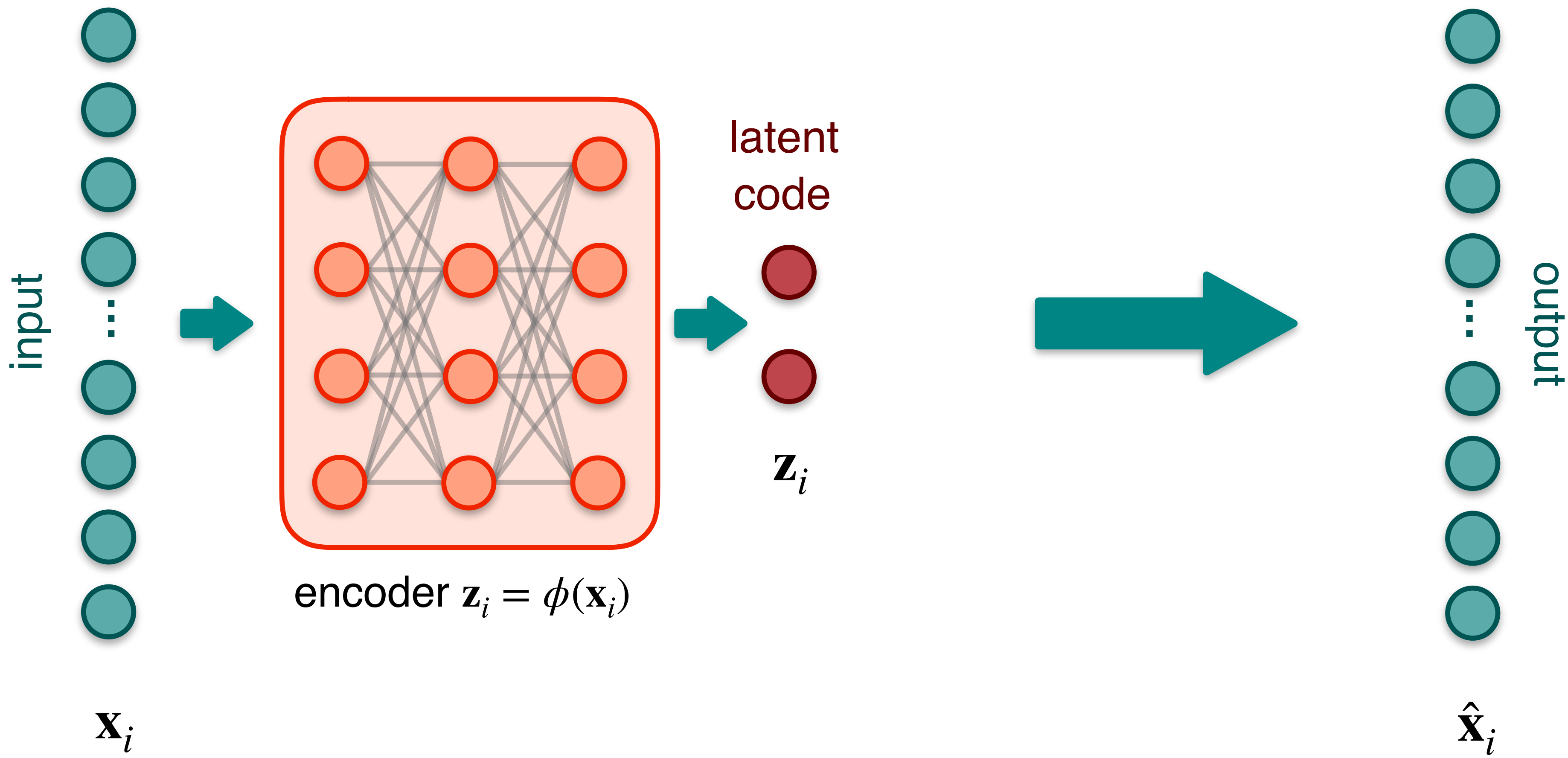


Autoencoder



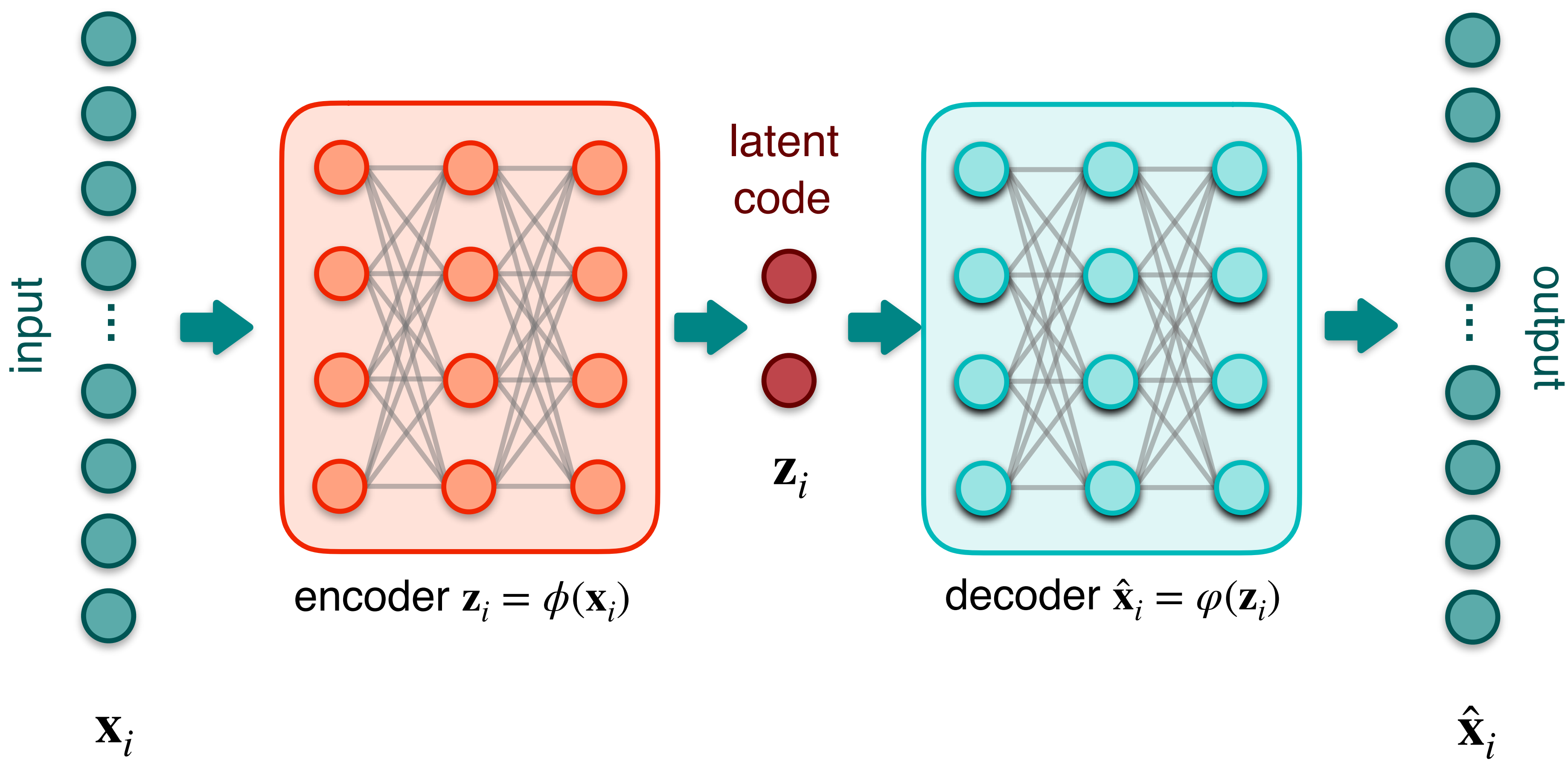
loss function: $$\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$$

Autoencoder



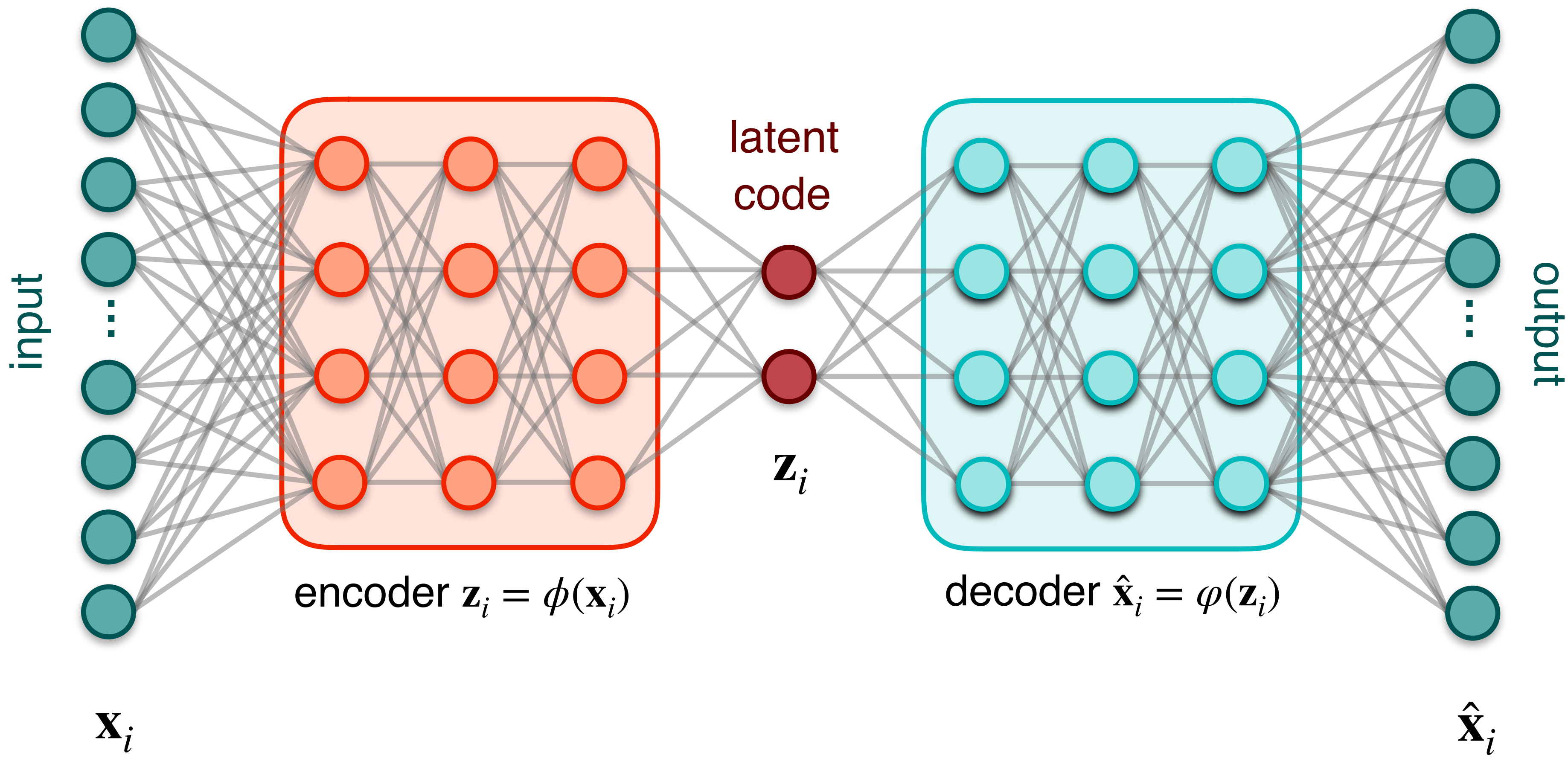
loss function: $\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$

Autoencoder



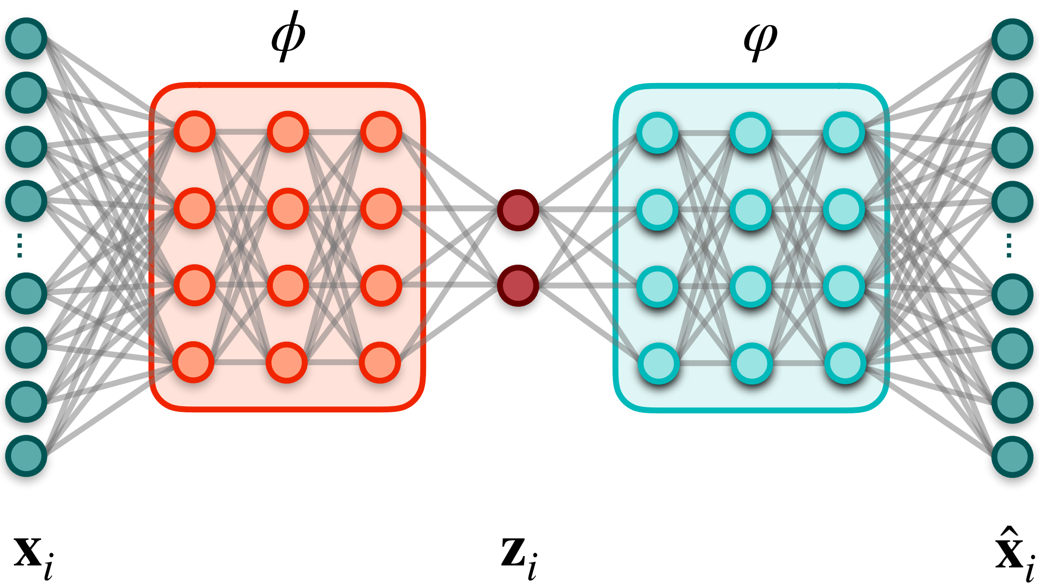
loss function: $\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$

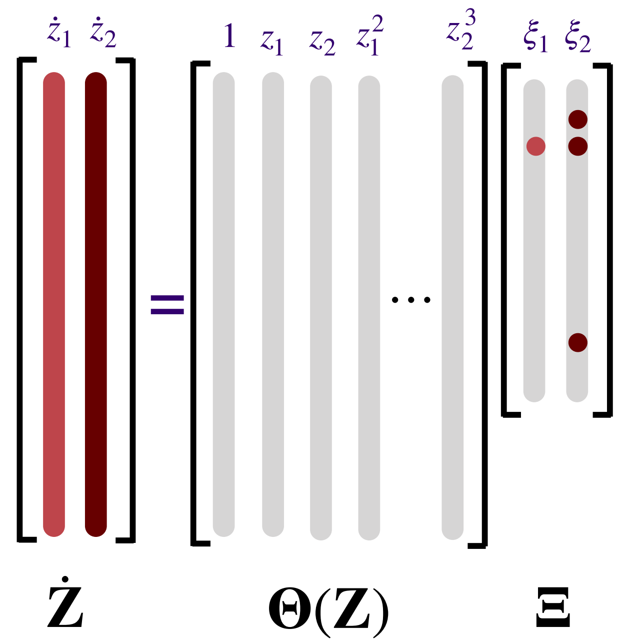
Autoencoder



loss function: $\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$

Autoencoder + SINDy

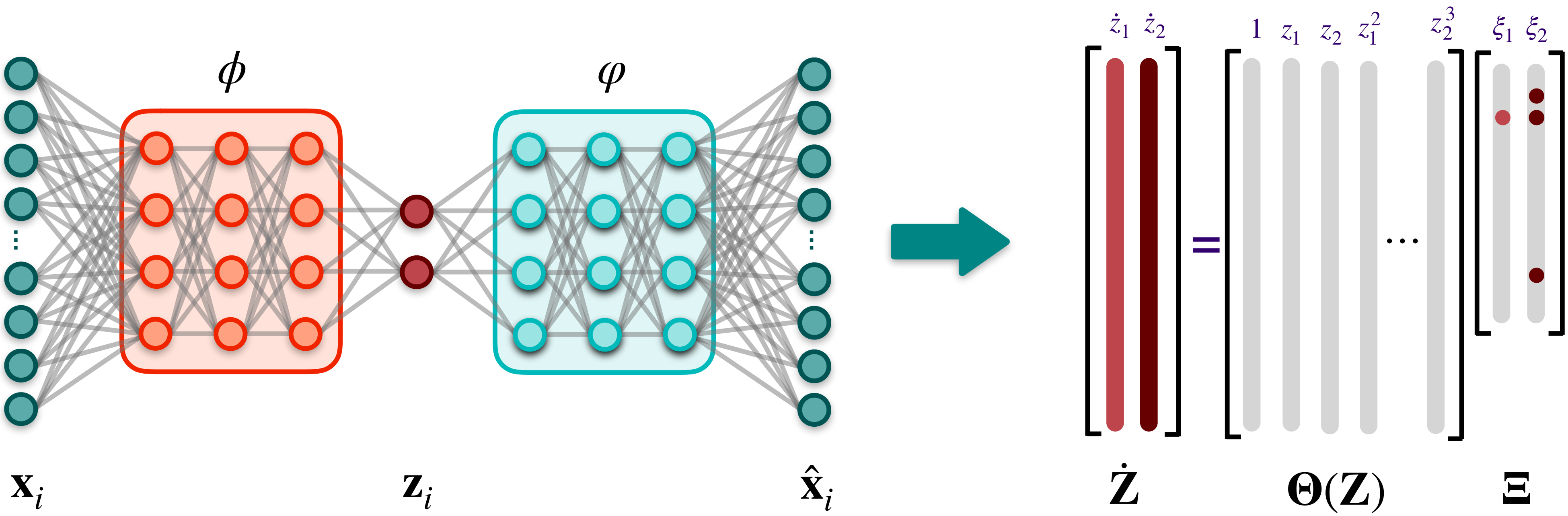




loss: $$\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$$

loss: $$\frac{1}{N}\sum_{i=1}^{N}\|\dot{\mathbf{z}}_i - \mathbf{\Theta}(\mathbf{z}_i^T)\mathbf{\Xi}\|_2^2$$

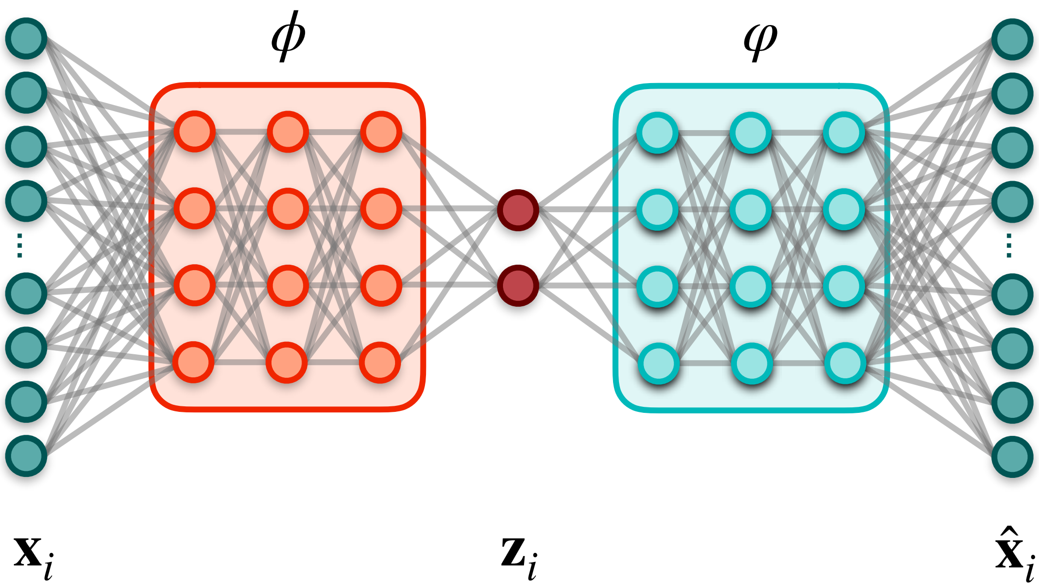
Autoencoder + SINDy

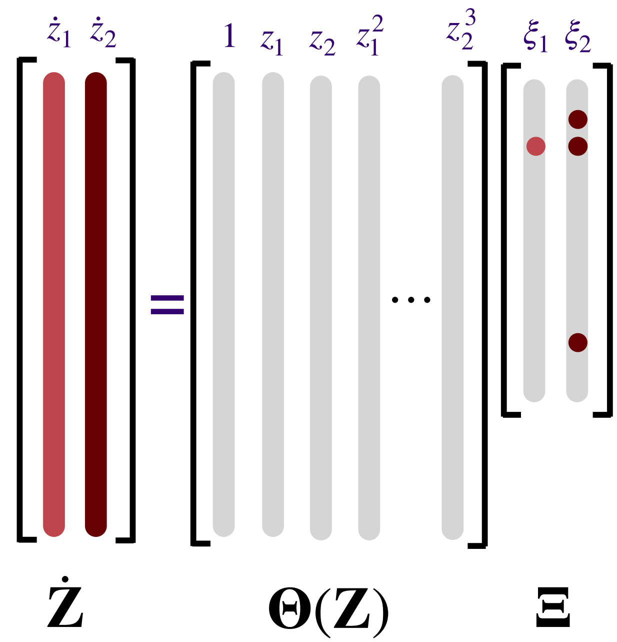


loss: $\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$

loss: $\frac{1}{N}\sum_{i=1}^{N} \|\dot{\mathbf{z}}_i - \boldsymbol{\Theta}(\mathbf{z}_i^T)\boldsymbol{\Xi}\|_2^2$

Autoencoder + SINDy

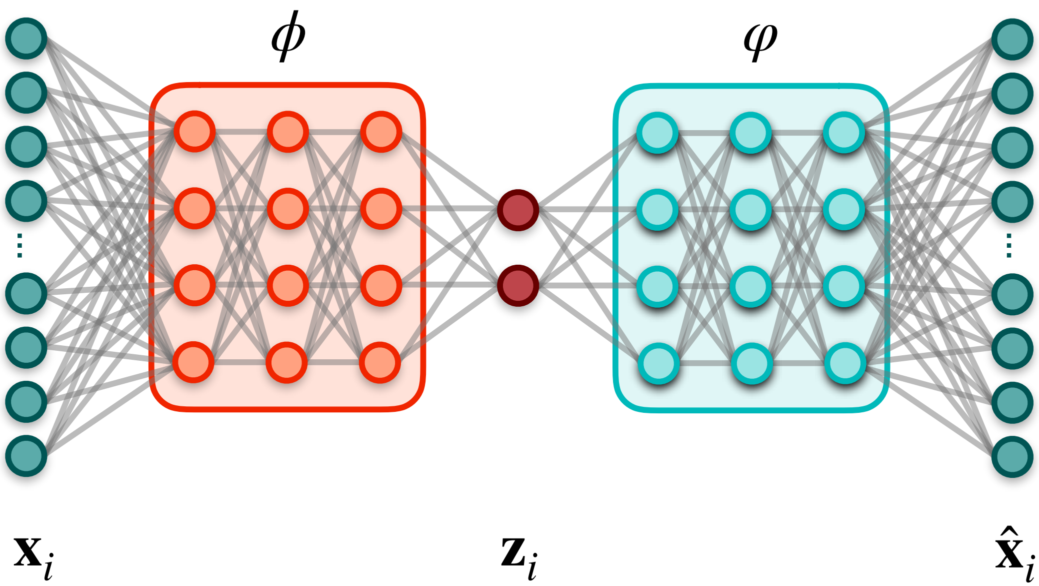


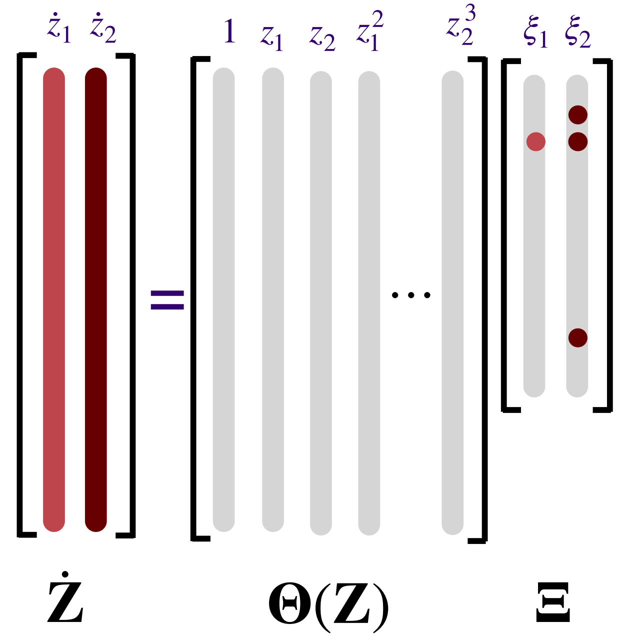


loss: $$\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$$

loss: $$\frac{1}{N}\sum_{i=1}^{N} \|\dot{\mathbf{z}}_i - \boldsymbol{\Theta}(\mathbf{z}_i^T)\boldsymbol{\Xi}\|_2^2$$

Autoencoder + SINDy



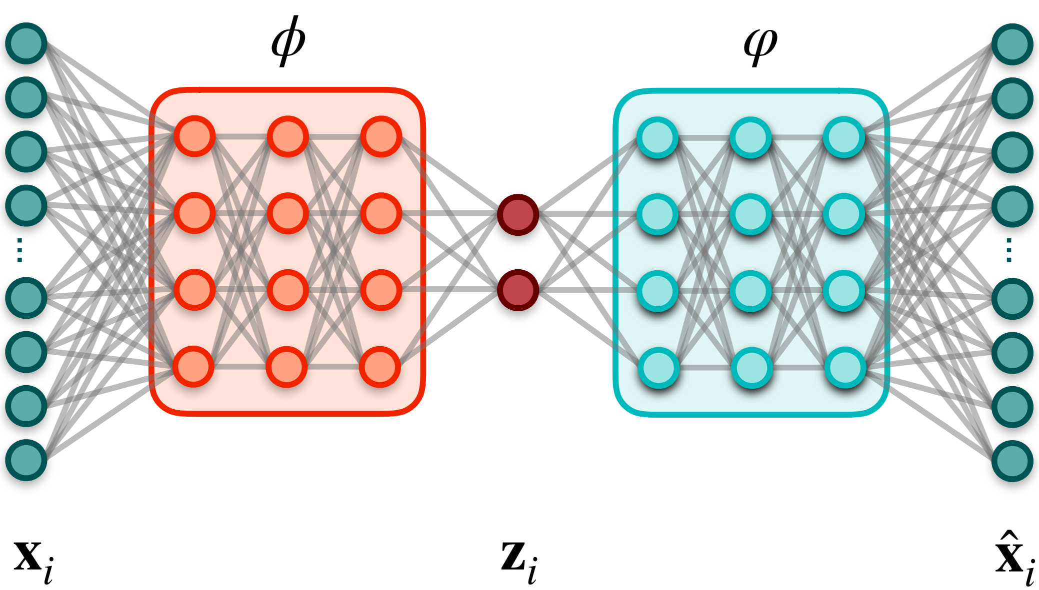


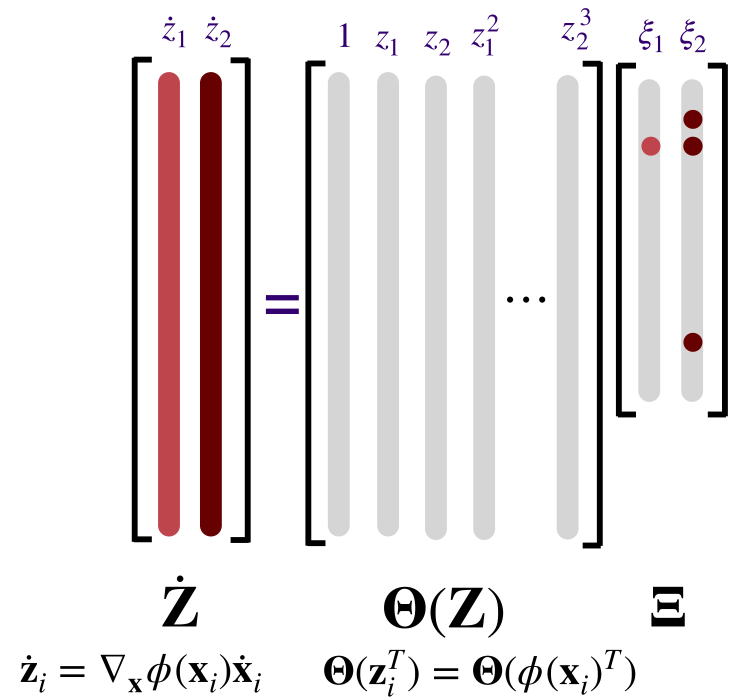
$\dot{\mathbf{z}}_i = \nabla_{\mathbf{x}} \phi(\mathbf{x}_i)\dot{\mathbf{x}}_i$

loss: $$\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$$

loss: $$\frac{1}{N}\sum_{i=1}^{N} \|\dot{\mathbf{z}}_i - \boldsymbol{\Theta}(\mathbf{z}_i^T)\boldsymbol{\Xi}\|_2^2$$

Autoencoder + SINDy

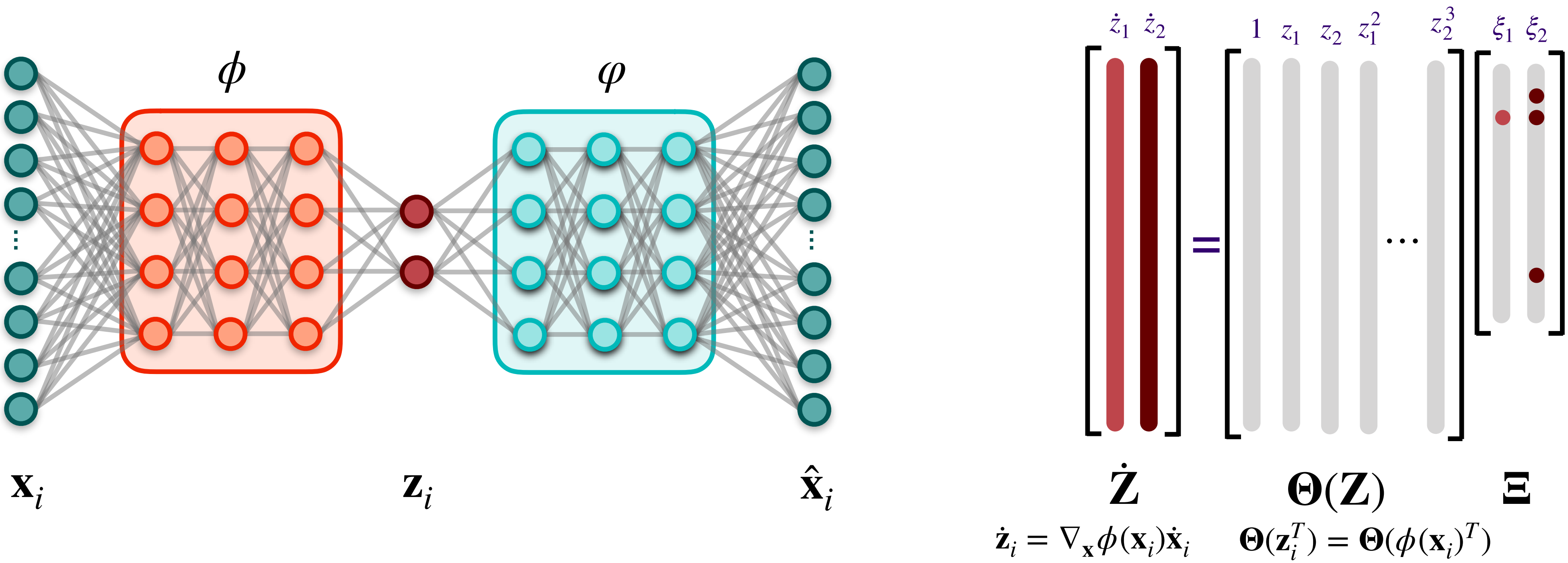




loss: $\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$

loss: $\frac{1}{N}\sum_{i=1}^{N}\|\dot{\mathbf{z}}_i - \boldsymbol{\Theta}(\mathbf{z}_i^T)\boldsymbol{\Xi}\|_2^2$

Autoencoder + SINDy

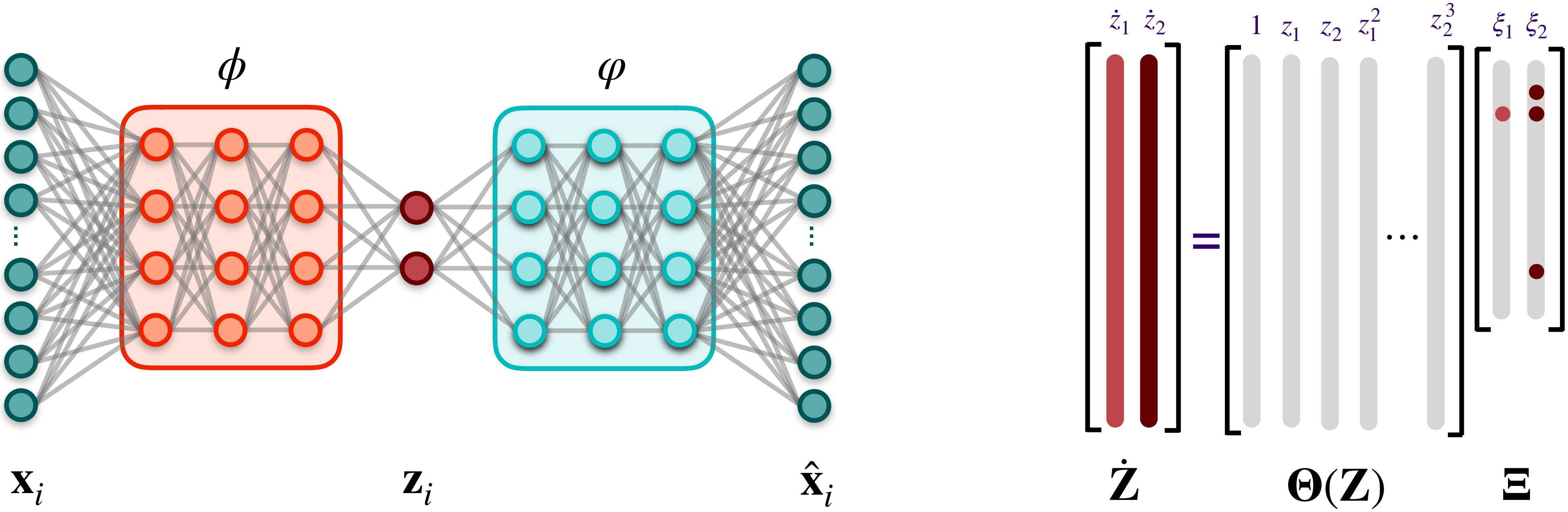


loss: $\lambda_1 \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \varphi(\phi(\mathbf{x}_i))\|_2^2 \; + \; \lambda_2 \frac{1}{N}\sum_{i=1}^{N} \|\nabla_{\mathbf{x}}\phi(\mathbf{x}_i)\dot{\mathbf{x}}_i - \boldsymbol{\Theta}(\phi(\mathbf{x}_i)^T)\boldsymbol{\Xi}\|_2^2$

autoencoder component

SINDy component

Autoencoder + SINDy

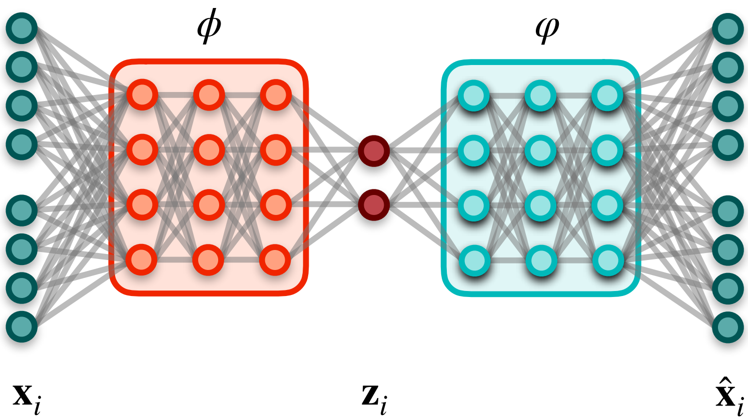


$\dot{\mathbf{z}}_i = \nabla_{\mathbf{x}}\phi(\mathbf{x}_i)\dot{\mathbf{x}}_i$ $\quad$ $\mathbf{\Theta}(\mathbf{z}_i^T) = \mathbf{\Theta}(\phi(\mathbf{x}_i)^T)$

loss: $\lambda_1 \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \varphi(\phi(\mathbf{x}_i))\|_2^2 \; + \; \lambda_2 \frac{1}{N}\sum_{i=1}^{N} \|\nabla_{\mathbf{x}}\phi(\mathbf{x}_i)\dot{\mathbf{x}}_i - \mathbf{\Theta}(\phi(\mathbf{x}_i)^T)\mathbf{\Xi}\|_2^2 + \lambda_3\|\mathbf{\Xi}\|_1$

autoencoder component $\quad$ **SINDy component**

Autoencoder + SINDy



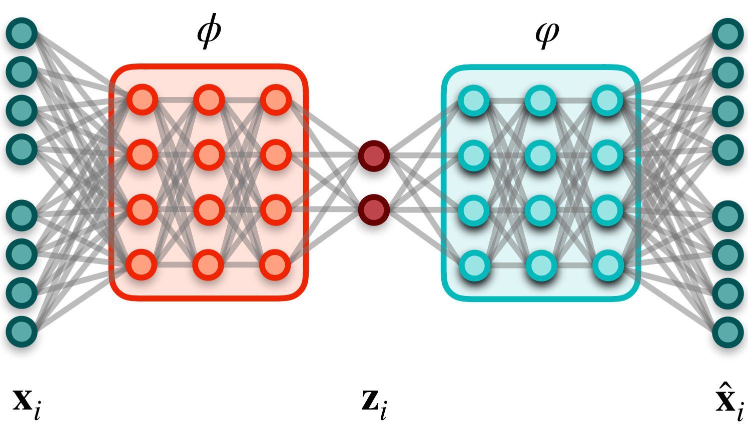
loss:

$$\lambda_1 \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 + \lambda_2 \frac{1}{N}\sum_{i=1}^{N} \|\dot{\mathbf{z}}_i - \boldsymbol{\Theta}(\mathbf{z}_i^T)\boldsymbol{\Xi}\|_2^2 + \lambda_3 \|\Xi\|_1$$

L_1 L_2 L_3

> ***Issue:*** training shrinks norm of **z** to minimize loss function

Autoencoder + SINDy



loss:

$$\lambda_1 \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 + \lambda_2 \frac{1}{N}\sum_{i=1}^{N} \|\dot{\mathbf{z}}_i - \boldsymbol{\Theta}(\mathbf{z}_i^T)\boldsymbol{\Xi}\|_2^2 + \lambda_3 \|\boldsymbol{\Xi}\|_1$$

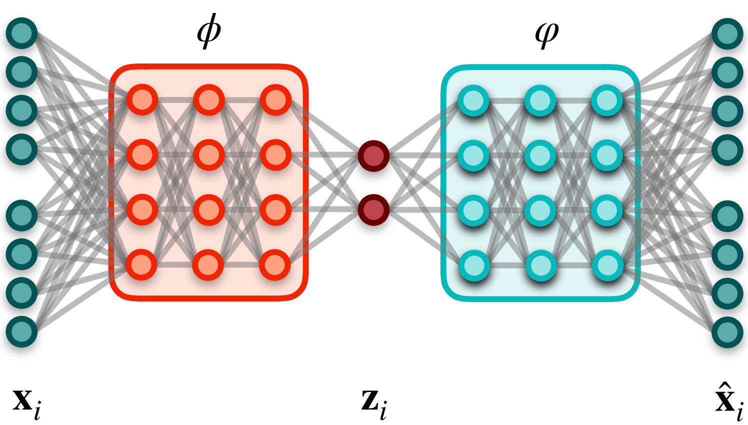
L_1 L_2 L_3

> ***Issue:*** training shrinks norm of **z** to minimize loss function
>
> ***Solution:*** use the following to enforce SINDy loss

new L_2 :

$$\frac{1}{N}\sum_{i=1}^{N} \|\dot{\mathbf{x}}_i - \nabla_{\mathbf{z}}\varphi(\mathbf{z}_i)\underbrace{\boldsymbol{\Theta}(\mathbf{z}_i^T)\boldsymbol{\Xi}}_{\dot{\mathbf{z}}_i}\|_2^2 = \frac{1}{N}\sum_{i=1}^{N} \|\dot{\mathbf{x}}_i - \nabla_{\mathbf{z}}\varphi(\phi(\mathbf{x}_i))\underbrace{\boldsymbol{\Theta}(\phi(\mathbf{x}_i)^T)\boldsymbol{\Xi}}_{\dot{\mathbf{z}}_i}\|_2^2$$

Autoencoder + SINDy



loss:

$$\lambda_1 \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 + \lambda_2 \frac{1}{N}\sum_{i=1}^{N} \|\dot{\mathbf{z}}_i - \boldsymbol{\Theta}(\mathbf{z}_i^T)\boldsymbol{\Xi}\|_2^2 + \lambda_3 \|\boldsymbol{\Xi}\|_1$$

L_1 L_2 L_3

> ***Issue:*** training shrinks norm of **z** to minimize loss function

> ***Solution:*** use the following to enforce SINDy loss

new L_2 :

$$\frac{1}{N}\sum_{i=1}^{N} \|\dot{\mathbf{x}}_i - \nabla_{\mathbf{z}}\varphi(\mathbf{z}_i)\underbrace{\boldsymbol{\Theta}(\mathbf{z}_i^T)\boldsymbol{\Xi}}_{\dot{\mathbf{z}}_i}\|_2^2 = \frac{1}{N}\sum_{i=1}^{N} \|\dot{\mathbf{x}}_i - \nabla_{\mathbf{z}}\varphi(\phi(\mathbf{x}_i))\underbrace{\boldsymbol{\Theta}(\phi(\mathbf{x}_i)^T)\boldsymbol{\Xi}}_{\dot{\mathbf{z}}_i}\|_2^2$$

> New loss function:

$$\lambda_1 \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 + \lambda_2 \frac{1}{N}\sum_{i=1}^{N} \|\dot{\mathbf{x}}_i - \nabla_{\mathbf{z}}\varphi(\mathbf{z}_i)\boldsymbol{\Theta}(\mathbf{z}_i^T)\boldsymbol{\Xi}\|_2^2 + \lambda_3 \|\boldsymbol{\Xi}\|_1$$

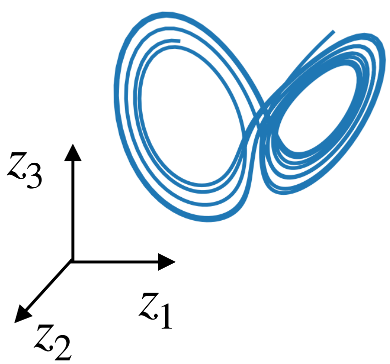
Achieving sparsity

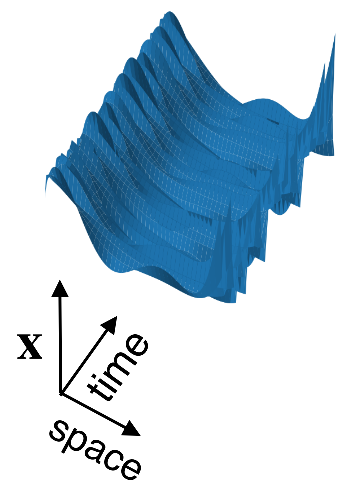
- With L1 penalty alone, get model that has many very small coefficients but is not truly sparse

$$\lambda_1 \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 + \lambda_2 \frac{1}{N} \sum_{i=1}^{N} \|\dot{\mathbf{x}}_i - \nabla_{\mathbf{z}} \varphi(\mathbf{z}_i) \boldsymbol{\Theta}(\mathbf{z}_i^T) \boldsymbol{\Xi}\|_2^2 + \lambda_3 \|\boldsymbol{\Xi}\|_1$$

- Instead combine L1 penalty with sequential thresholding

Example problem





$$\mathbf{x}(t) = \mathbf{A}\mathbf{z}(t) + \mathbf{B}\begin{pmatrix} z_1^3(t) \\ z_2^3(t) \\ z_3^3(t) \end{pmatrix}$$

$$\mathbf{x}(t) \in \mathbb{R}^{128}$$

$$\mathbf{A}, \mathbf{B} \in \mathbb{R}^{128 \times 3}$$

Example problem

Lorenz model

Equations

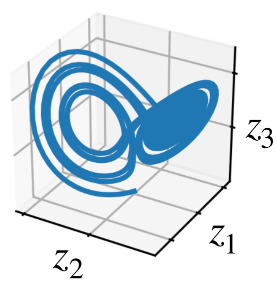
$$\dot{z}_1 = -10z_1 + 10z_2$$

$$\dot{z}_2 = 28z_1 - z_2 - z_1z_3$$

$$\dot{z}_3 = -2.7z_3 + z_1z_2$$

Coefficient Matrix Ξ

Dynamics

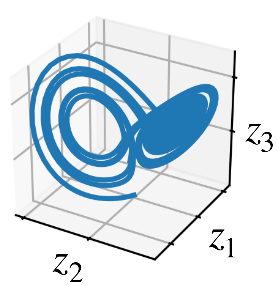


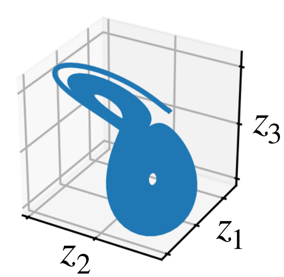
Example problem

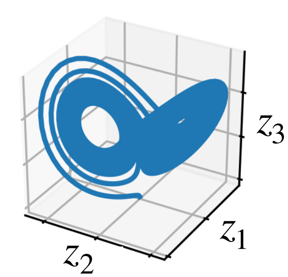
	Lorenz model	Discovered model	Discovered model (transformed)
Equations	$\dot{z}_1 = -10z_1 + 10z_2$	$\dot{z}_1 = -8.5z_2z_3$	$\dot{z}_1 = -10.2z_1 + 8.8z_2$
	$\dot{z}_2 = 28z_1 - z_2 - z_1z_3$	$\dot{z}_2 = 9.2 - 2.9z_2 + 1.1z_1z_3$	$\dot{z}_2 = 26.7z_1 - 8.5z_1z_3$
	$\dot{z}_3 = -2.7z_3 + z_1z_2$	$\dot{z}_3 = -8.8z_1 - 10.3z_3$	$\dot{z}_3 = -2.9z_3 + 1.1z_2$

Coefficient Matrix Ξ

Dynamics







Discovery Models & Coordinates Simultaneously