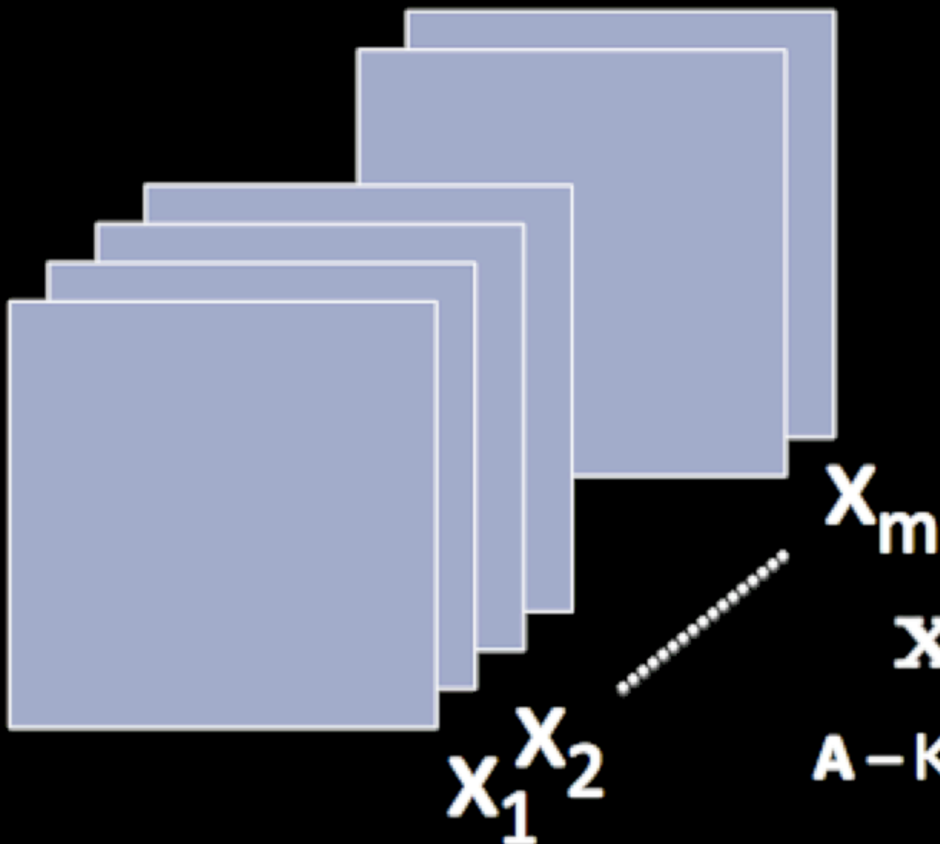
Computing Koopman Operators

J. Nathan Kutz
Department of Applied Mathematics
University of Washington
Seattle, WA 98195-3925
Email: kutz@uw.edu

May 29, 2019 – SIAM Dynamical Systems

Data Snapshots



$$\mathbf{x}_{j+1} = \mathbf{A}\mathbf{x}_j$$

A – Koopman operator [1]

determine Koopman through all frames

Schmid (2008/10), Rowley et al (2009)

Dynamic Mode Decomposition

Definition: Dynamic Mode Decomposition (Tu et al. 2014 [1]): *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*

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

where $\dagger$ *denotes the Moore-Penrose pseudoinverse.*

[1] Tu et al J. Comp. Dyn. (2014)

Approximate Dynamical Systems

Linear dynamics (equation-free)

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

Eigenfunction expansion

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

The Algorithm

svd | $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*$ $\qquad$ $\mathbf{Y} = \mathbf{A}\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*$

$$\mathbf{U}^*\mathbf{Y}\mathbf{V}\boldsymbol{\Sigma}^{-1} = \mathbf{U}^*\mathbf{A}\mathbf{U} \equiv \tilde{\mathbf{A}}$$

eig | $\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\boldsymbol{\Lambda}$

eigenvalues: growth/decay, oscillations

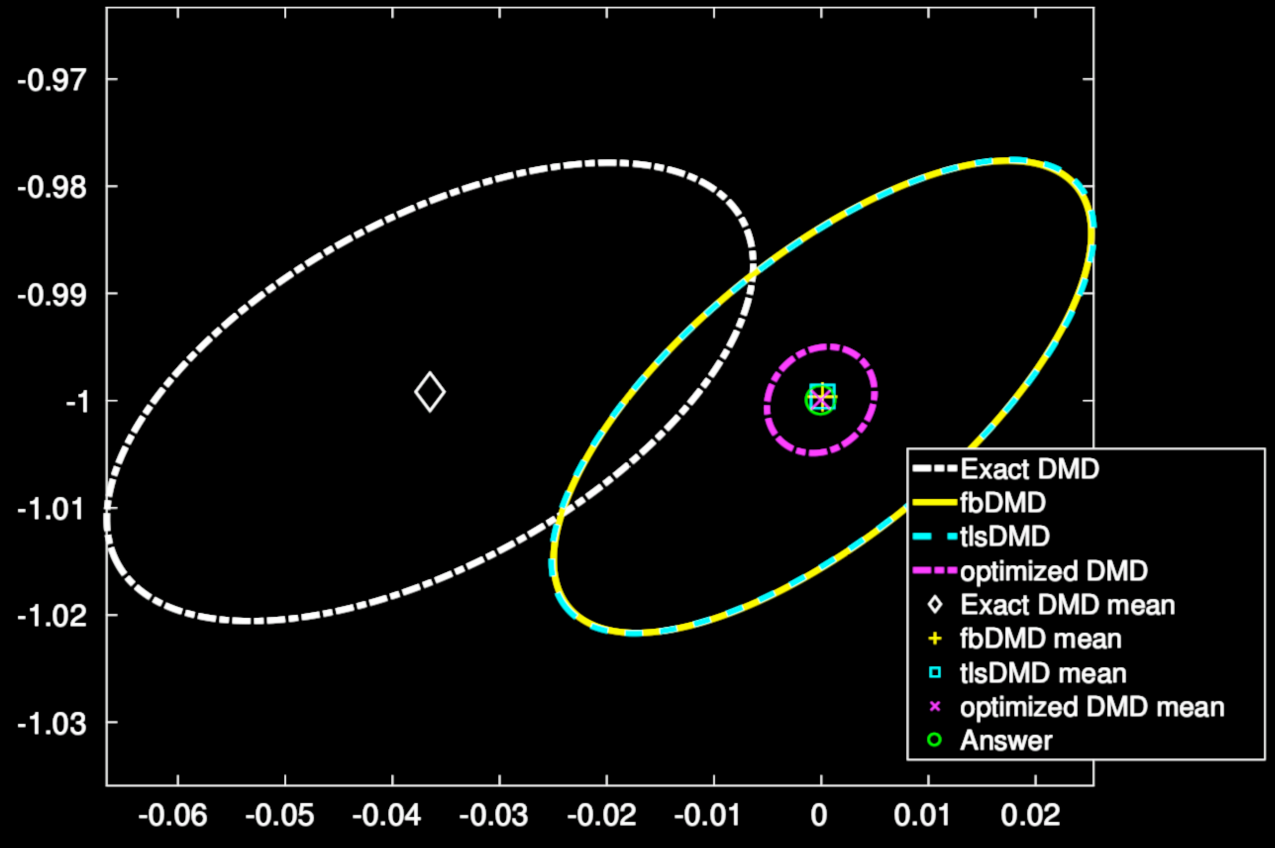
$$\boldsymbol{\Phi} = \mathbf{Y}\mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{W}$$

DMD modes: spatial correlations between measurements

Programming Break

DMD Algorithms

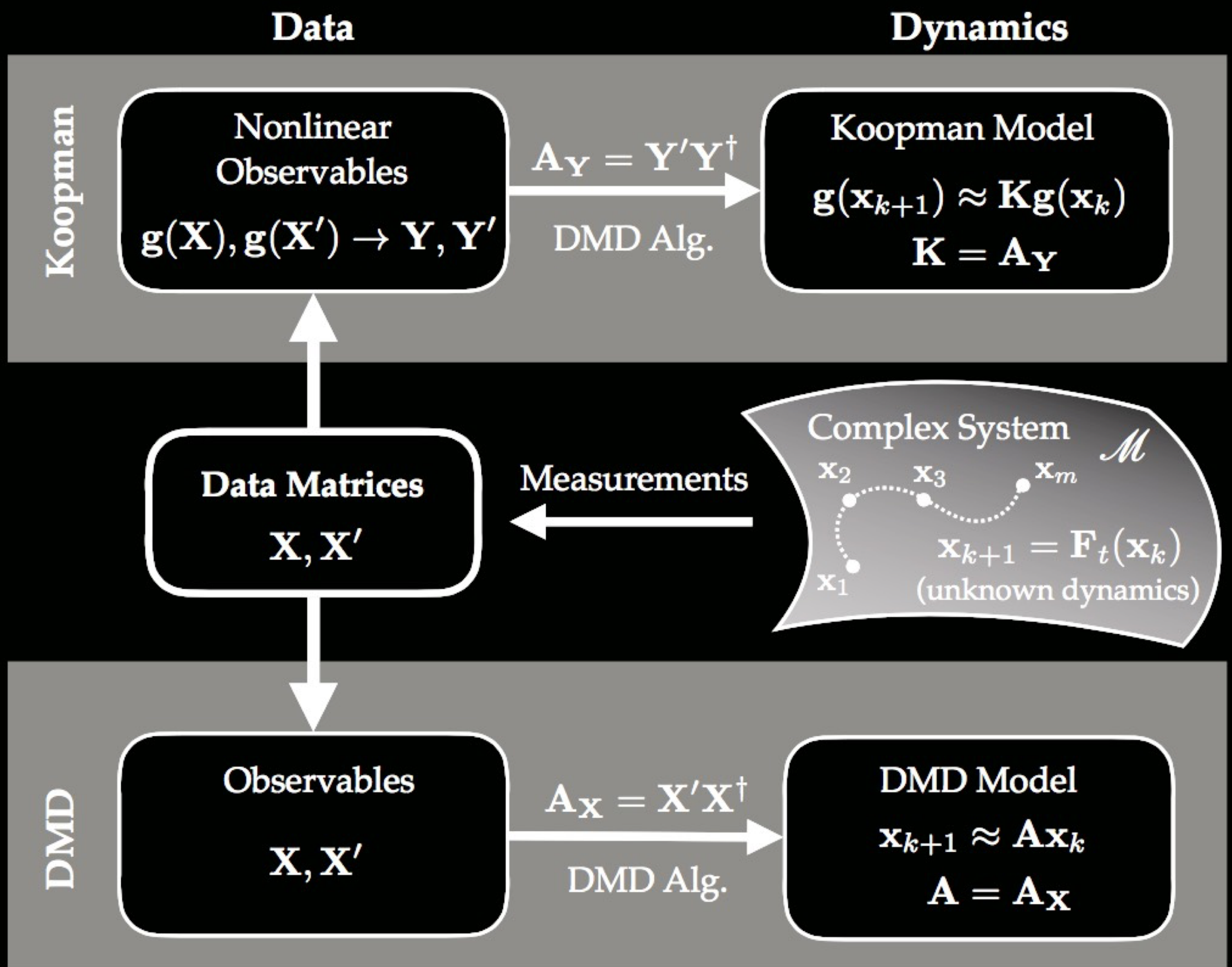
- Forward-Backward DMD (Dawson et al 2016)
- Total Least-Squares DMD (Hemati et al 2017)
- Optimized DMD (Askham & Kutz 2018)



DMD for Koopman

- Extented DMD (Williams et al 2015)
- Kernel DMD (Williams et al 2016)
- Judicious choice (Kutz et al 2018)

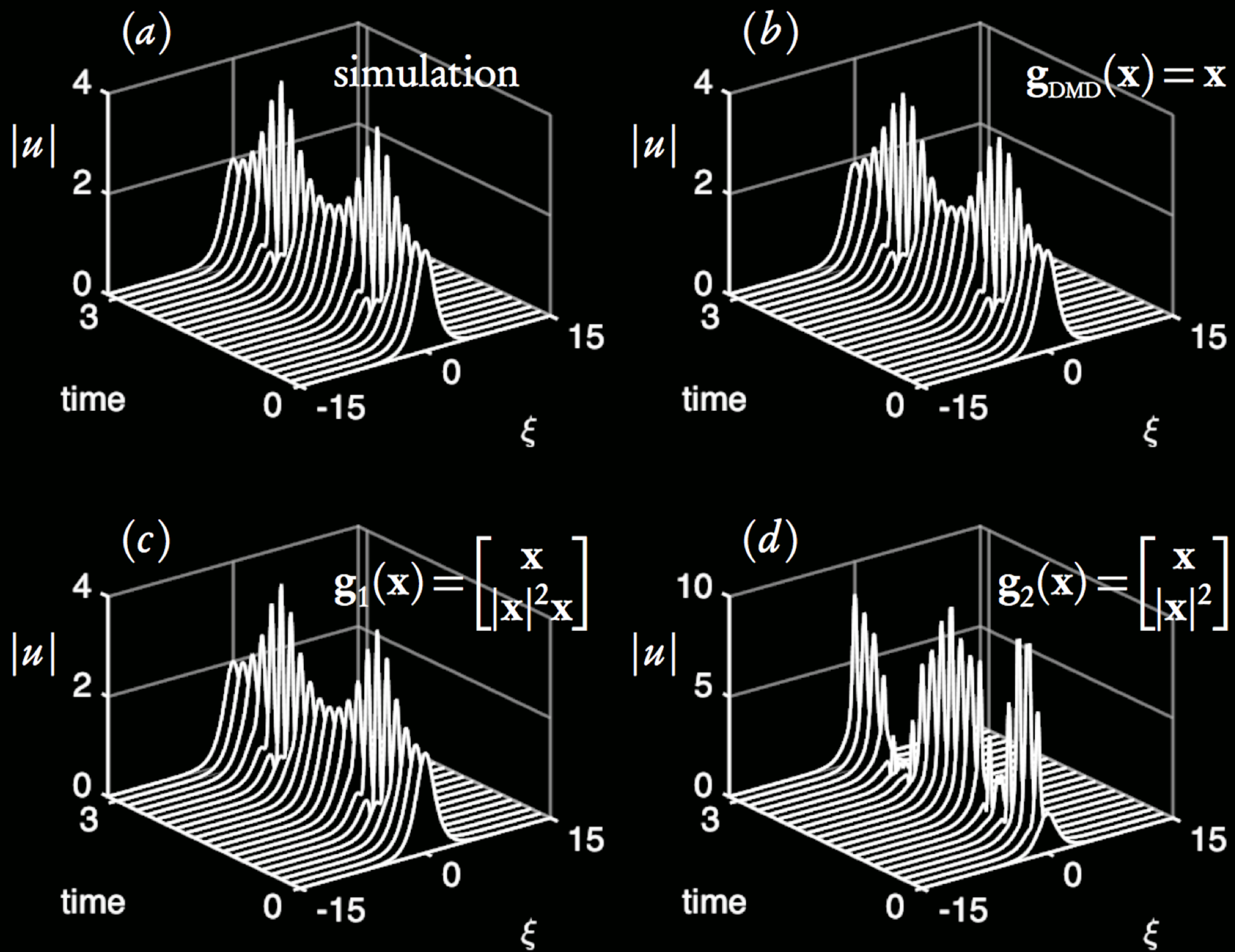
Koopman vs DMD: All about Observables!



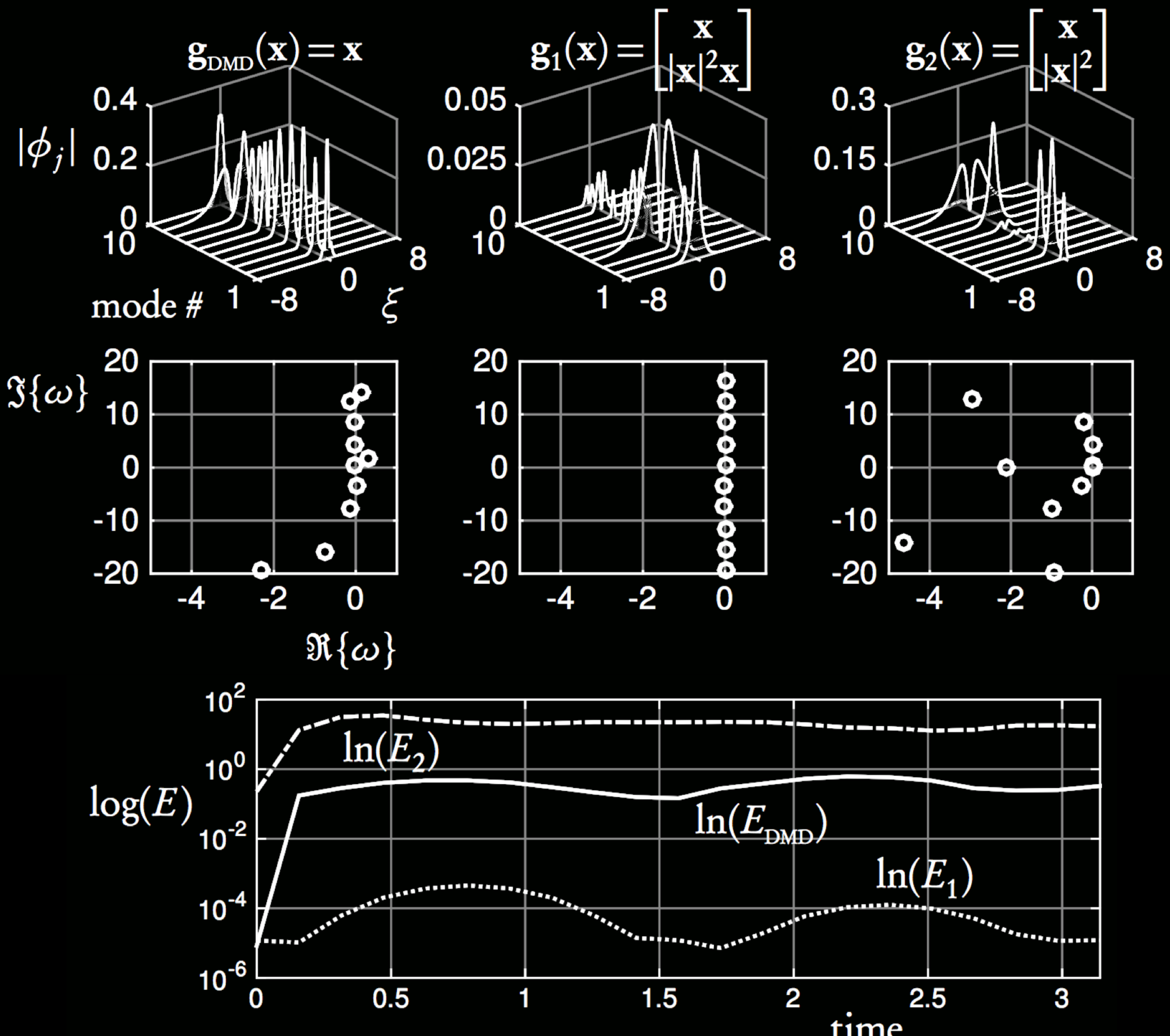
Programming Break

Nonlinear Schrodinger Equation

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

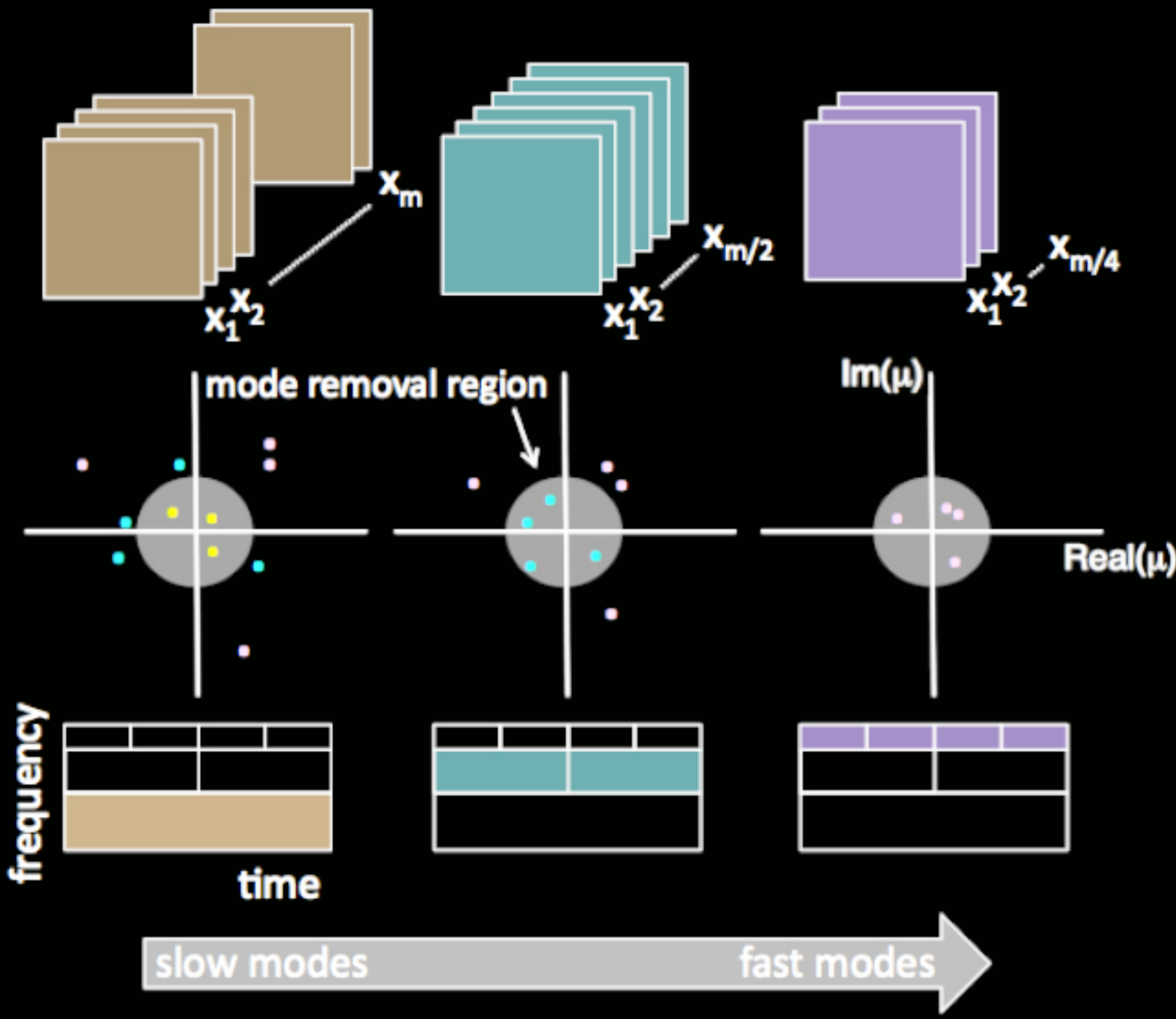


Error and DMD Modes

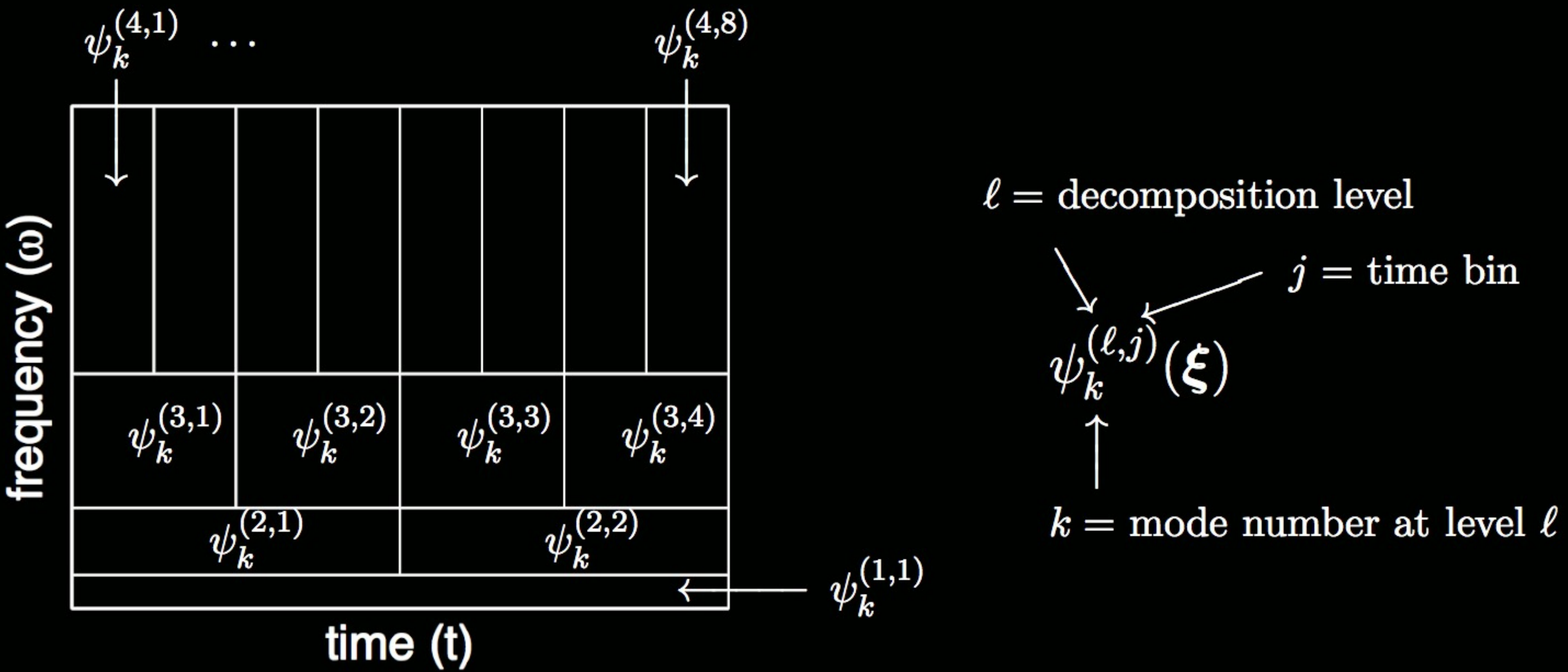


Multi-Resolution DMD

Multi-Resolution DMD



Multi-Resolution DMD



Multi-Resolution Separation

Slow vs Fast Separation

$$\mathbf{x}_{\mathrm{mrDMD}}(t) = \sum_{k=1}^{M} b_k(0)\psi_k^{(1)}(\mathbf{x})\exp(\omega_k t) = \underbrace{\sum_{k=1}^{m_1} b_k(0)\psi_k^{(1)}(\mathbf{x})\exp(\omega_k t)}_{\text{(slow modes)}} + \underbrace{\sum_{k=m_1+1}^{M} b_k(0)\psi_k^{(1)}(\mathbf{x})\exp(\omega_k t)}_{\text{(fast modes)}}$$

$$\mathbf{x}_{M/2} = \sum_{k=m_1+1}^{M} b_k(0)\psi_k^{(1)}(\mathbf{x})\exp(\omega_k t)$$

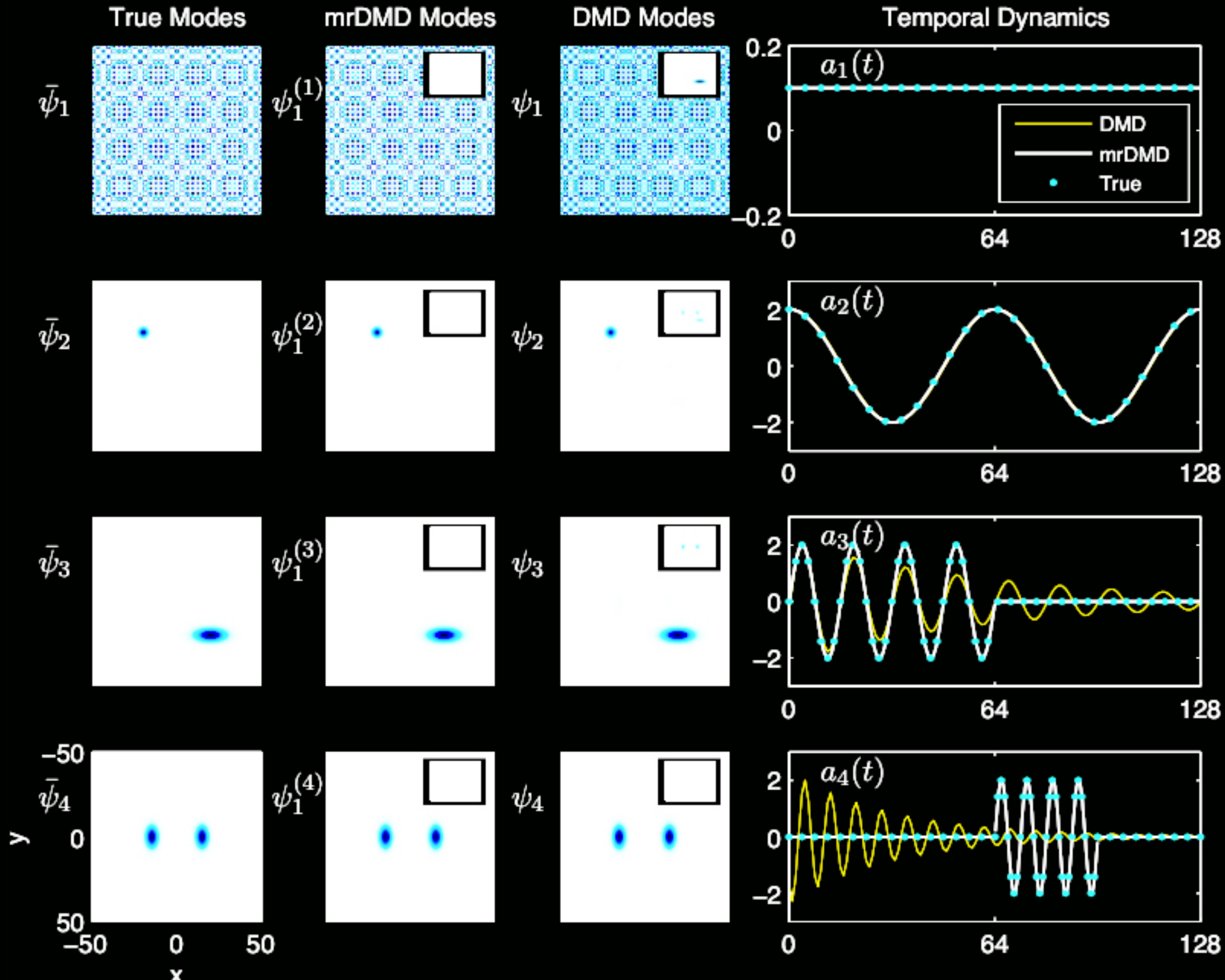
$$\mathbf{X}_{M/2} = \mathbf{X}_{M/2}^{(1)} + \mathbf{X}_{M/2}^{(2)}$$

General Multiscale Separation

$$\mathbf{x}_{\mathrm{mrDMD}}(t) = \sum_{k=1}^{m_1} b_k^{(1)}\psi_k^{(1)}(\mathbf{x})\exp(\omega_k^{(1)} t) + \sum_{k=1}^{m_2} b_k^{(2)}\psi_k^{(2)}(\mathbf{x})\exp(\omega_k^{(2)} t) + \sum_{k=1}^{m_3} b_k^{(3)}\psi_k^{(3)}(\mathbf{x})\exp(\omega_k^{(3)} t) + \cdots$$

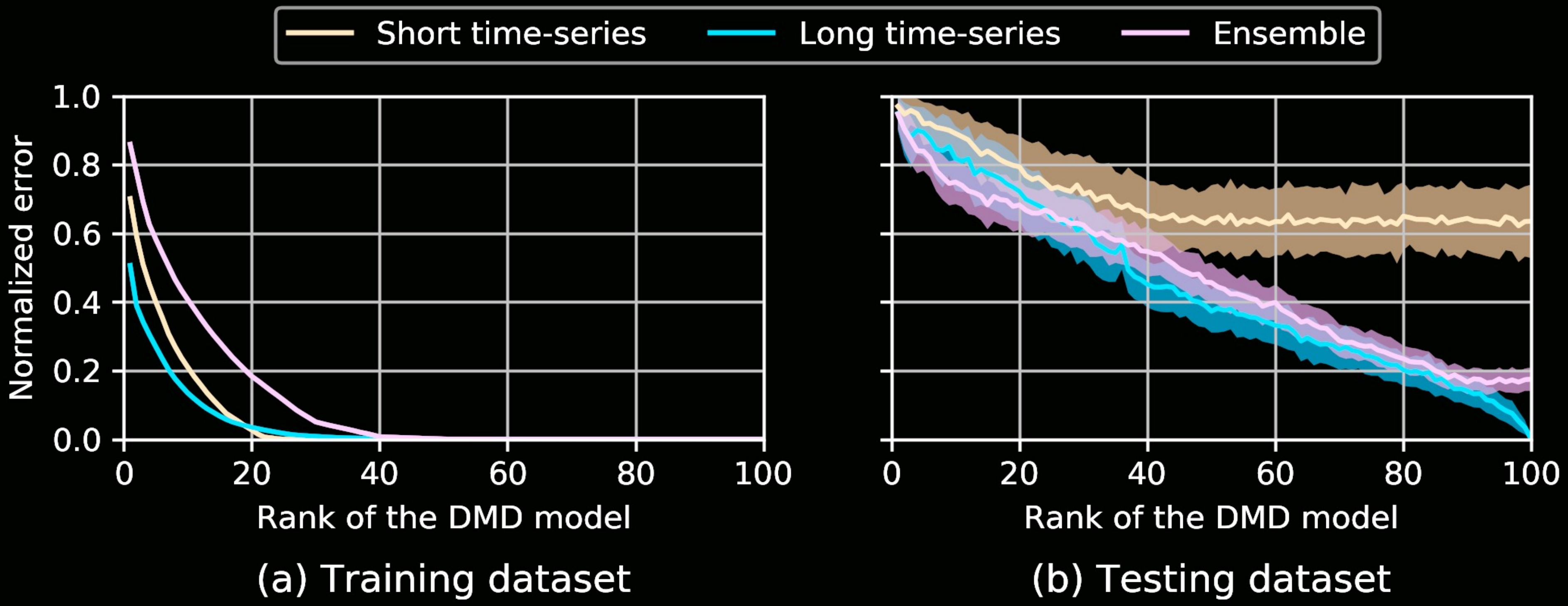
(or Brunton, Proctor, Kutz, PNAS 2016)

Dynamics Example



DMD Forecasting

pyDMD Package (example 8)



J.C. Loiseau

DMD with Control (DMDc)

Modified Regression

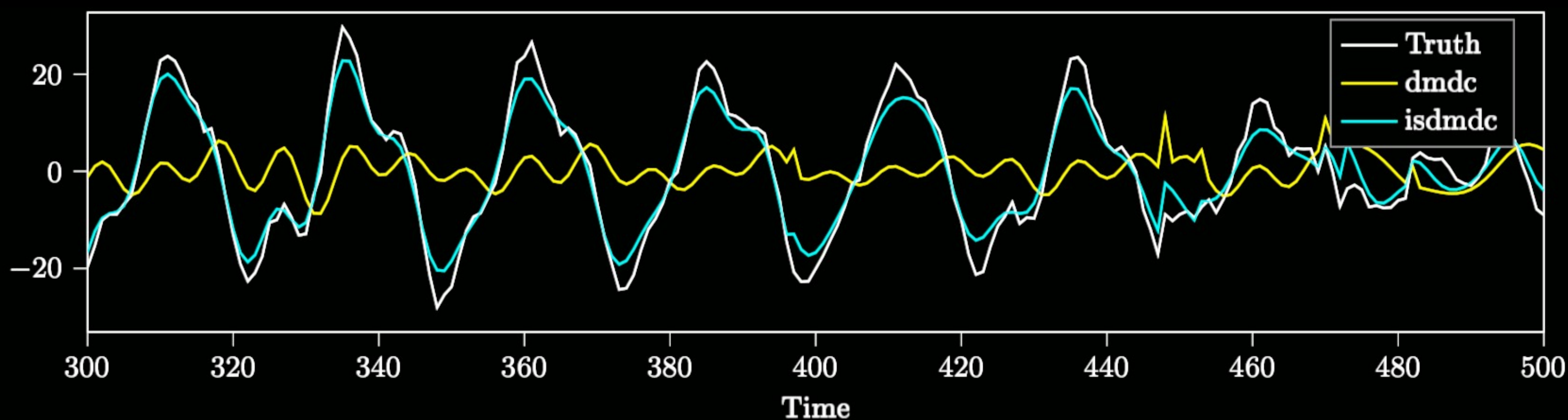
Input

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

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

Input Snapshots

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



Proctor, Brunton & Kutz, SIADS (2016)

Fiesler & Kutz, SIADS (2019)