Recurrent Autoencoder Networks for Koopman Spectral Analysis

Samuel Otto
Bill Eggert

Introduction and Motivation

- We want to explore data driven modeling of dynamical systems using machine learning

- Dynamical systems (e.g., turbulent flow) are traditionally handled with Direct Numerical Simulation, or State Space Reduction

- We propose another method: RNN’s with autoencoders

The Basics

- Suppose we have measured data $\{x_j \in \mathbb{R}^n\}_{j=1}^{m}$, but a model for the dynamics $x_{t+1} = f(x_t)$ is unavailable
  - We want to model how observables (functions $g: \mathbb{R}^n \to \mathbb{R}$) change in time
  - We would love to write $g(x_{t+1}) = \mathcal{U}g(x_t)$ where $\mathcal{U}$ is the "Koopman operator"
  - Approximate the evolution of observables on a finite-dimensional invariant subspace spanned by ϕ's

Linear Dynamics in Feature Space

- Let $z = \Phi(x) \triangleq \begin{bmatrix} \phi_1(x) \\ \vdots \\ \phi_q(x) \end{bmatrix} \in \mathbb{R}^q$ be the vector of nonlinear observable functions in feature space at state $x \in \mathbb{R}^n$
- Find an approximate linear update $\Phi\big(f(x)\big) \approx A\Phi(x)$ hence $z_{t+1} \approx Az_t$
- Dynamics are assumed to be linear in feature space $\mathbb{R}^q$

$$\begin{tikzcd}
x_t \arrow[rrr] \arrow[ddd] & & & x_{t+1} \arrow[ddd] \\
& \mathbb{R}^n \arrow[r, "\text{Nonlinear dynamics}\; f"] \arrow[d, "\Phi"'] & \mathbb{R}^n \arrow[d, "\Phi"] & \\
& \mathbb{R}^q \arrow[r, "\text{Matrix}\; A"] & \mathbb{R}^q & \\
z_t \arrow[rrr] & & & Az_t \approx z_{t+1}
\end{tikzcd}$$

Approximating Koopman

- $\{\phi_i \in L^2(\mathbb{R}^n)\}_{i=1}^{q}$ span a subspace of observable functions
- To each $\alpha \in \mathbb{R}^q$ define the corresponding observable function $g = \bar{\Phi}\alpha = \alpha_1\phi_1 + \cdots + \alpha_q\phi_q \in L^2(\mathbb{R}^n)$ i.e. $g(x) = \Phi(x)^T\alpha$
- We have $\mathcal{U}g(x) = g\big(f(x)\big) = \Phi\big(f(x)\big)^T\alpha \approx \Phi(x)^T A^T\alpha$ hence $\mathcal{U}g \approx \bar{\Phi}A^T\alpha$
- The matrix A^T can be used to approximate the Koopman operator restricted to $\mathcal{R}(\bar{\Phi}) = span\{\phi_1, \dots, \phi_q\} \subset L^2(\mathbb{R}^n)$ as shown in the following diagram

"$\bar{\Phi} = [\phi_1 \dots \phi_q]$"

$$
\begin{tikzcd}
g = \alpha_1\phi_1 + \cdots + \alpha_q\phi_q \arrow[rrr, mapsto] & & & g \circ f \approx \alpha_1{}'\phi_1 + \cdots + \alpha_q{}'\phi_q \\
& \mathcal{R}(\bar{\Phi}) \subset L^2(\mathbb{R}^n) \arrow[r, "\mathcal{U}"] & \mathcal{R}(\bar{\Phi}) \subset L^2(\mathbb{R}^n) & \\
& \mathbb{R}^q \arrow[u, "\bar{\Phi}"] \arrow[r, "A^T"] & \mathbb{R}^q \arrow[u, "\bar{\Phi}"'] & \\
\alpha \arrow[uuu, mapsto] \arrow[rrr, mapsto] & & & \alpha' = A^T\alpha \arrow[uuu, mapsto]
\end{tikzcd}
$$

The Koopman Operator

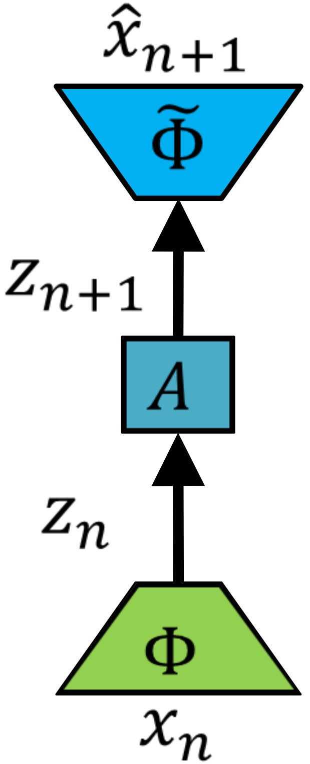
- To approximate $\mathcal{U}$ without leaving $span\{\phi_i\}_{i=1}^{q}$, we need to learn a Koopman invariant subspace.
- Previous methods require us to choose a basis $\{\phi_i\}_{i=1}^{q}$, but good choices are hard to find.
- Additionally these systems are huge (millions of state variables for turbulent flow)
- Proposed solution: RNN of autoencoders to learn Φ and A (therefore approximate $\mathcal{U}$)

Deep Learning Approach

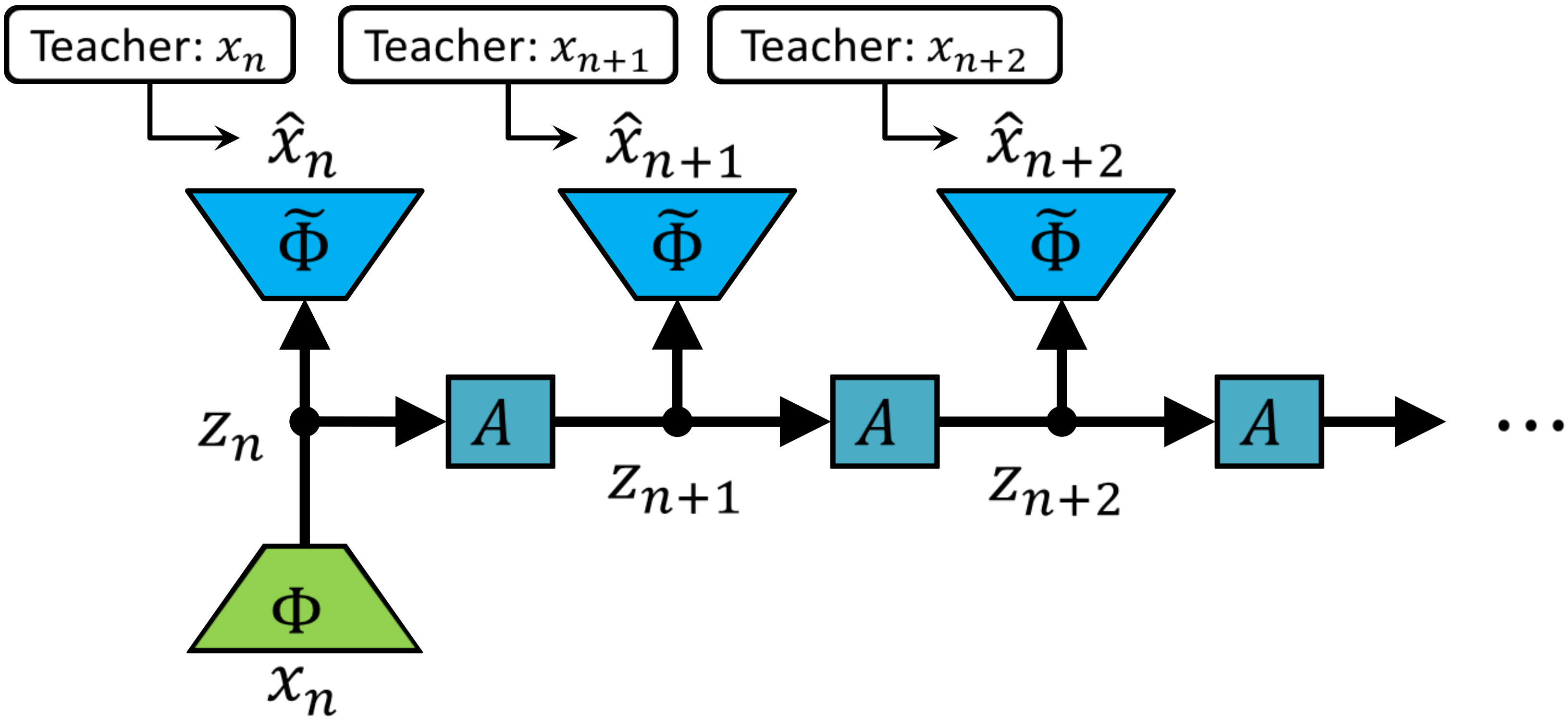
- Want: linear dynamics in an intrinsic feature space comprised of nonlinear observables
- Use autoencoder to learn an optimal set of nonlinear observables
  - Nonlinear encoder and decoder
- Constrain dynamics to be linear in the intrinsic feature space
- Autoencoder need not be contractive since having linear dynamics is a strong constraint.

Naïve Approach

- Introduce nonlinear encoder Φ and decoder $\widetilde{\Phi}$
- Introduce linear dynamics A in intrinsic feature space
- The diagram is under-constrained
  - Just a deep feed-forward network learning to update the state
  - There is nothing forcing the nonlinear mapping and the dynamics to be decoupled
- The maps Φ and $\widetilde{\Phi}$ must not be allowed to take part in the update



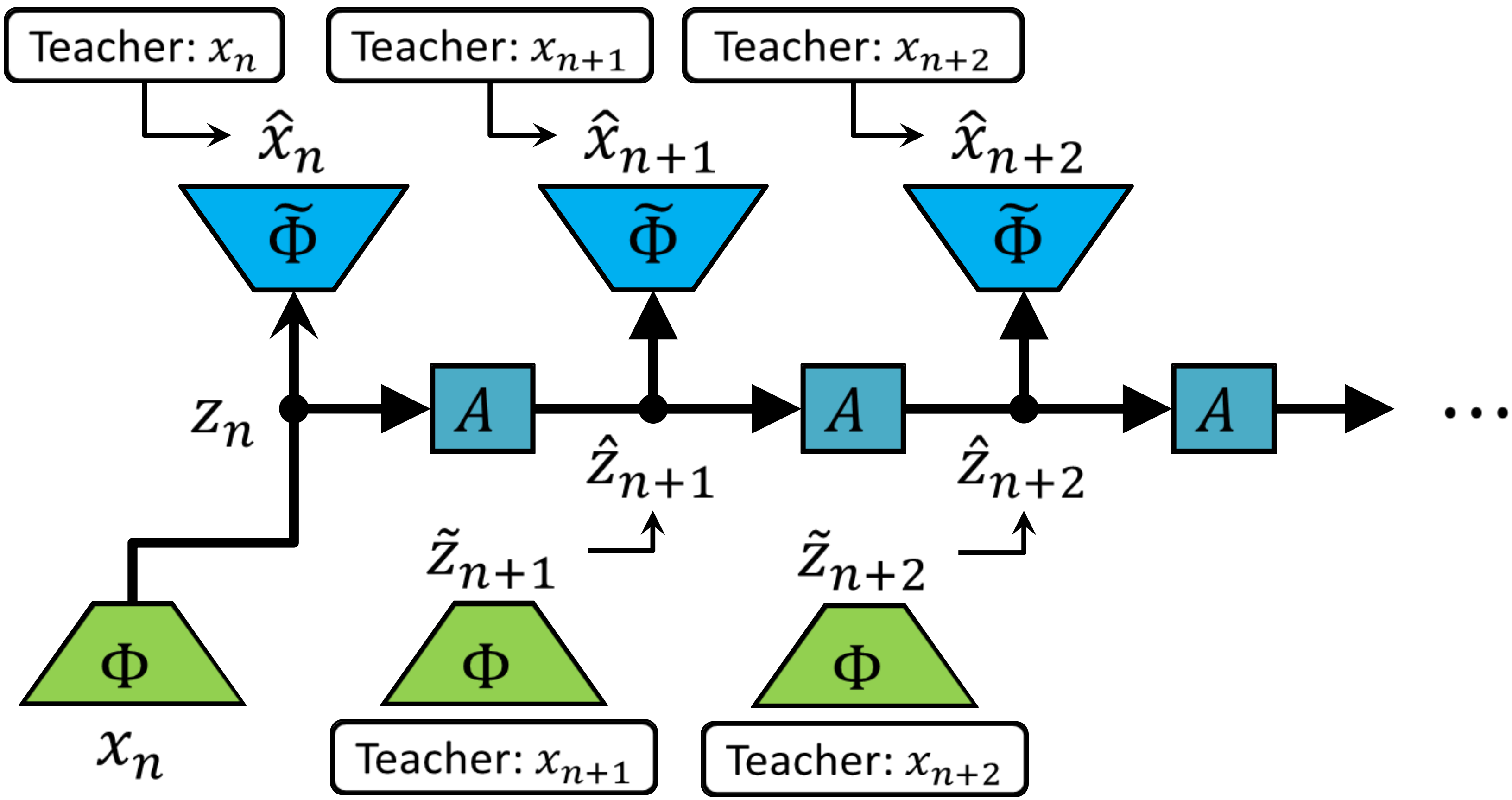
Recurrent Autoencoder Network (I)



- Linear dynamics: Matrix $A \in \mathbb{R}^{q \times q}$
- Nonlinear embedding: encoder Φ and decoder $\widetilde{\Phi}$
- Cost function: Decaying error norm with $\alpha \geq 1$

$$J(A, \Phi, \widetilde{\Phi}) = \mathbb{E}_{x_n \sim P_{data}} \left[\sum_{k=0}^{M} \frac{1}{\alpha^k} \| \hat{x}_{n+k} - x_{n+k} \|^2 \right]$$

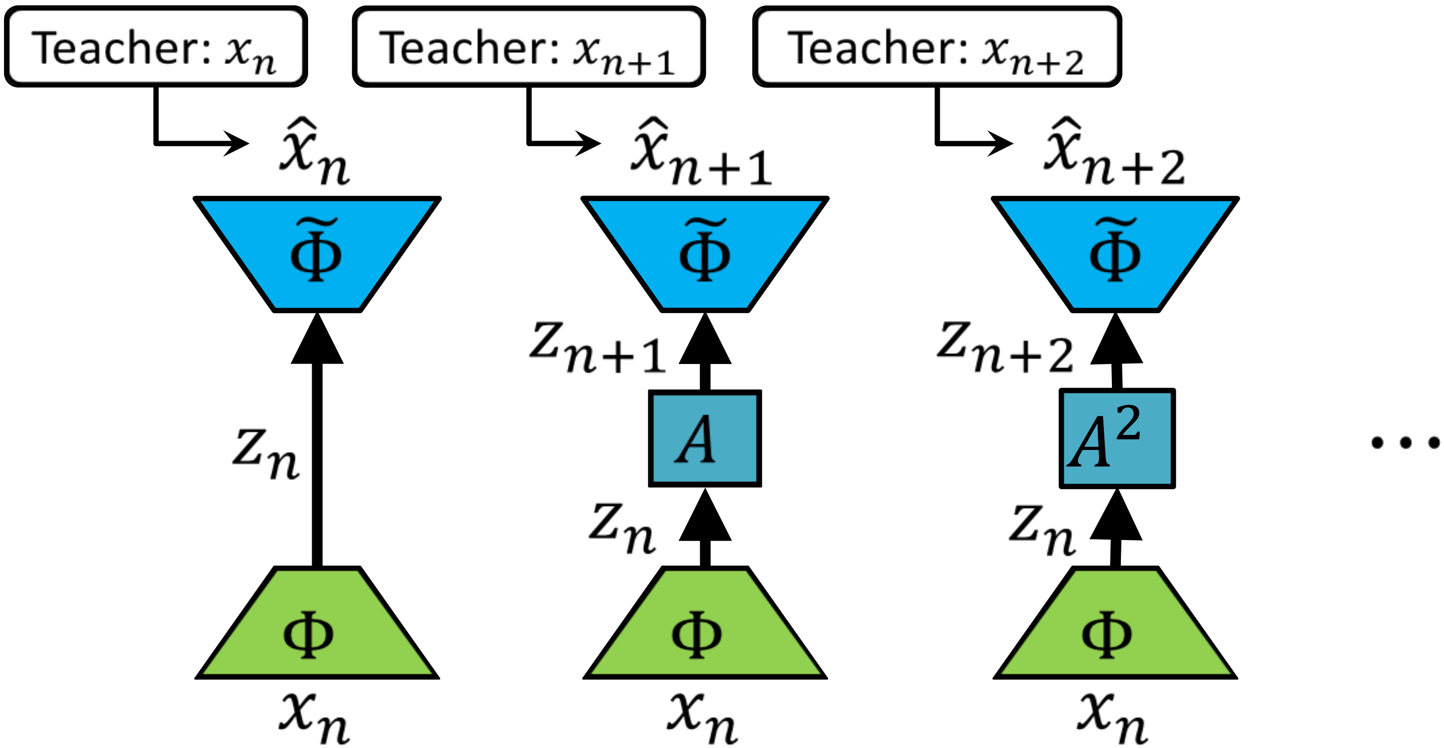
Recurrent Autoencoder Network (II)



- Introduce teacher values on the code z too
- Cost function: Decaying error norm with $\alpha \geq 1$

$$J(A, \Phi, \widetilde{\Phi}) = \mathbb{E}_{x_n \sim P_{data}} \left[\sum_{k=0}^{M} \frac{1}{\alpha^k} \|\hat{x}_{n+k} - x_{n+k}\|^2 + \beta \sum_{k=1}^{M} \frac{1}{\alpha^k} \|\hat{z}_{n+k} - \tilde{z}_{n+k}\|^2 \right]$$

Recurrent Autoencoder Network (III)



- Train on varying number of time steps using $A^0, A^1, \ldots, A^k$
- Cost function: Decaying error norm with $\alpha \geq 1$

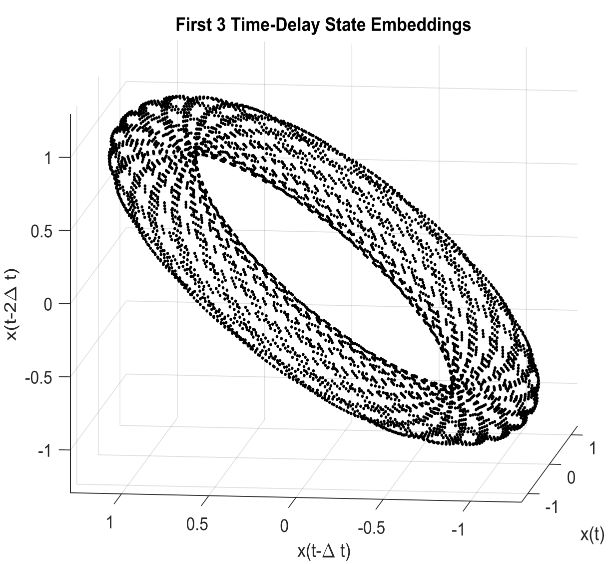
$$J(A, \Phi, \tilde{\Phi}) = \mathbb{E}_{x_n \sim P_{data}} \left[\sum_{k=0}^{M} \frac{1}{\alpha^k} \|\hat{x}_{n+k} - x_{n+k}\|^2 \right]$$

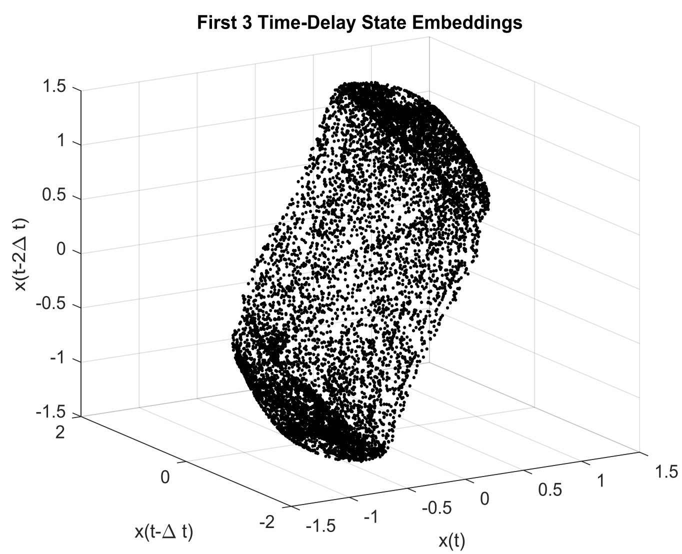
Training Methodology

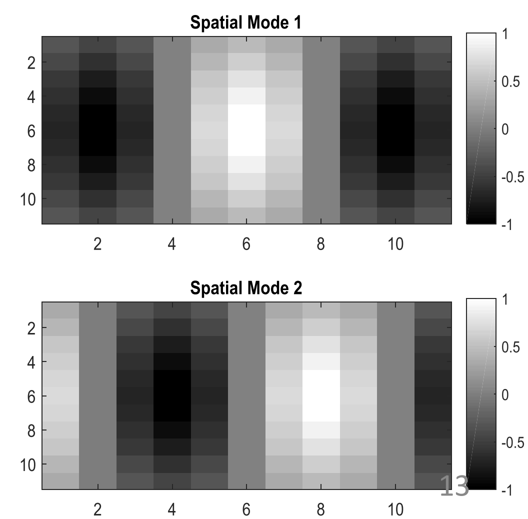
1. Directly train like RNN with linear state dynamics
   – Use Back Propagation (BP) and Stochastic Gradient Descent (SGD) to update both Φ and A
2. Alternating linear and nonlinear updates
   – Fix A and update Φ using BP and SGD
   – Fix Φ and update A using SVD

Proposed Test Cases

- Test on contrived linear and nonlinear dynamical systems
  - –Linear spring
  - –Nonlinear pendulum
  - –Forced Duffing equation (Chaotic)
- Consider different state embeddings
  - –Time-delay embedding $x_t = [s_t, s_{t-1}, \ldots, s_{t-k}]^T$
  - –Modal image embedding: Use states as coefficients on spatial modes







Proposed Test Cases

- Results using recurrent autoencoder network will be compared to
  - Dynamic Mode Decomposition (DMD)
  - Extended Dynamic Mode Decomposition (EDMD)
  - Kernel Dynamic Mode Decomposition (KDMD)
- Use modal decomposition in intrinsic space to extract dynamically important structures and their time evolution
  - Koopman spectral analysis ➔ Koopman modes, eigenvalues
  - Eigenvalues give frequency and damping of modes

Koopman Spectral Analysis

- Eigenvalue decomposition of dynamics in intrinsic space

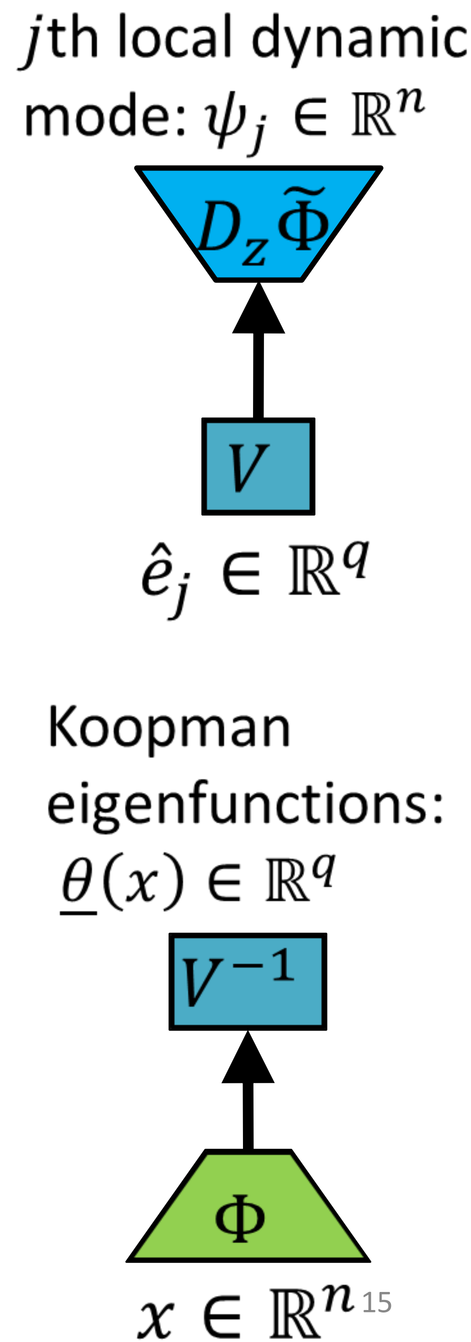
$$A = V\Lambda V^{-1} \qquad V = \left[v_1 \cdots v_q\right]$$

- The *Koopman eigenfunctions* are approximated

$$\underline{\theta}(x) = \begin{bmatrix} \theta_1(x) \\ \vdots \\ \theta_q(x) \end{bmatrix} = V^{-1}\Phi(x)$$

- The jth *Koopman mode* is v_j
- The jth *local dynamic mode* may be an interesting object for examination

$$\psi_j(x_0) = D_z\widetilde{\Phi}(z_0)v_j = D_z\widetilde{\Phi}(z_0)V\hat{e}_j$$



Koopman Spectral Analysis

- Expanding the learned observables in terms of Koopman eigenfunctions

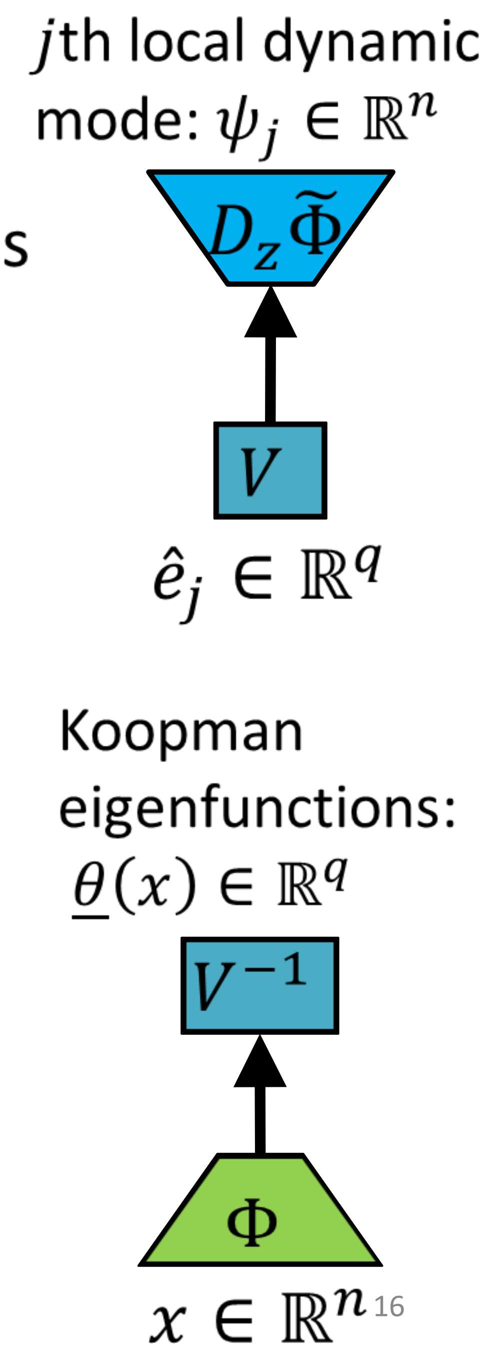
$$\Phi(x) = \sum_{j=1}^{q} \theta_j(x) v_j$$

- Applying the Koopman operator

$$\Phi(x_{n+1}) = \mathcal{U}\Phi(x_n) = \sum_{j=1}^{q} \lambda_j \theta_j(x_n) v_j$$

- We have the modal decomposition of the intrinsic space dynamics

$$\Phi(x_n) = \sum_{j=1}^{q} \lambda_j^n \theta_j(x_0) v_j \Longrightarrow \hat{x}_n \approx \widetilde{\Phi}\left[\sum_{j=1}^{q} \lambda_j^n \theta_j(x_0) v_j\right]$$



Koopman Spectral Analysis

- Local dynamic modes are tangent to the dynamics in empirical space

$$x_{n+1} \approx x_n + \sum_{j=1}^{q} (\lambda_j - 1)\theta_j(x_n)\psi_j$$

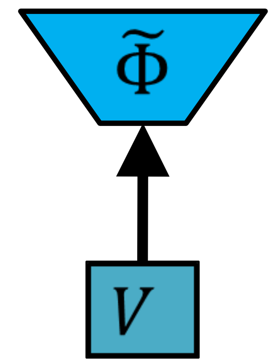
- Local dynamic modes approximate perturbation dynamics

$$\Psi_n = [\psi_1(x_n) \cdots \psi_q(x_n)]$$
$$x_n = \bar{x}_n + \Delta x_n$$

➔Modal decomposition in tangent space using 1st order approximation

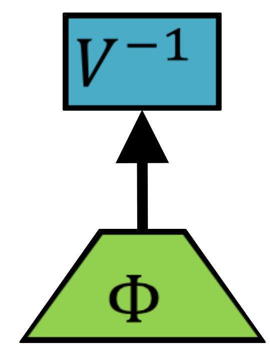
$$\Delta x_{n+1} = \Psi_n \Lambda \Psi_n{}^{+} \Delta x_n$$

$\Delta x = \Psi \Delta h$



$\Delta h \in \mathbb{R}^q$

$\Delta h \approx \Psi^{+} \Delta x$



$\Delta x \in \mathbb{R}^n$

Identification of Dynamically Important Structures

C. W. Rowley, I. Mezic, S. Bagheri, P. Schlatter, D. S. Henningson "Spectral Analysis of Nonlinear Flows", 2009, J. Fluid Mech

- Identification of Koopman modes for fluid jet in crossflow
- Linear observables were used
- Koopman modes extract dynamically important flow structures and their corresponding time evolution

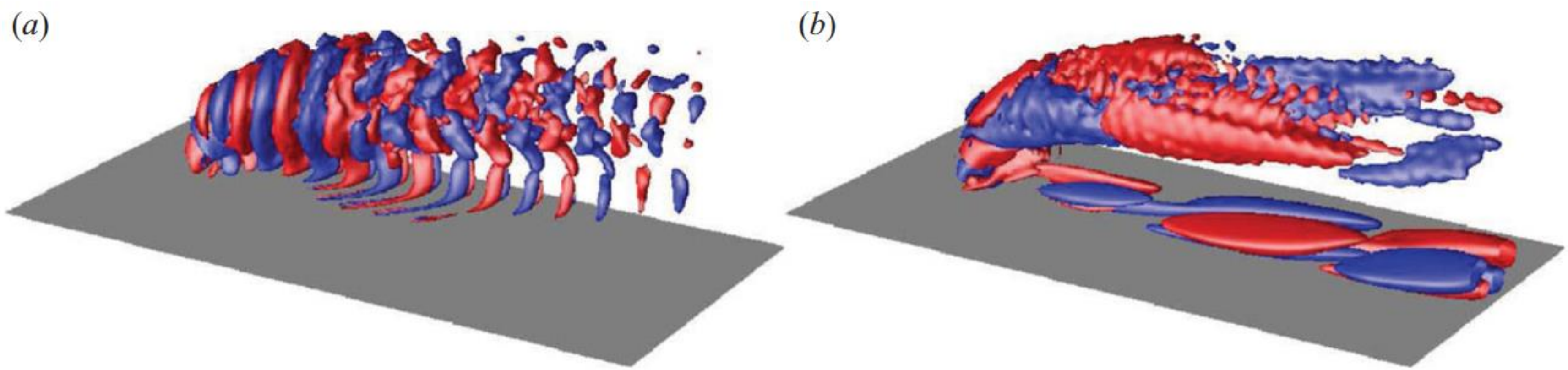


FIGURE 3. Positive (red) and negative (blue) contour levels of the streamwise velocity components of two Koopman modes. The wall is shown in grey. (*a*) Mode 2, with $\|\boldsymbol{v}_2\| = 400$ and $St_2 = 0.141$. (*b*) Mode 6, with $\|\boldsymbol{v}_6\| = 218$ and $St_6 = 0.0175$.