Structured Autoencoders for Operator-theoretic decomposition and Model reduction

Karthik Duraisamy

Center for Data-Driven Computational Physics

UNIVERSITY of MICHIGAN
Thanks to...

Shaowu Pan
Chris Wentland
David Xu
Motivation

Decomposition & Reduced Order Modeling of Complex Multiscale Problems

Large scale simulations $O(10^6)$- $O(10^8)$ CPU hours / run
Complex physics: Flow, turbulence, combustion, heat transfer, etc
The Autoencoder

Structure: encoder (compression) + decoder (decompression)
- Encoder $\Phi(x; \theta_\Phi)$
- Decoder $\Psi(\cdot; \theta_\Psi)$
- POD: $\Phi \rightarrow U^T(\cdot), \Psi \rightarrow U(\cdot)$
- Trained as one single network, $\theta_\Phi$ and $\theta_\Psi$ are optimized jointly
- Automatically separated into encoder and decoder by cutting at the “bottleneck”

\[ \tilde{x} = \Psi(\cdot, \theta_\Psi) \circ \Phi(x, \theta_\Phi) \]
Embedding (in the right coordinates)

\[ \frac{d}{dt} \mathbf{x} = \mathbf{F}(\mathbf{x}(t)) \quad \rightarrow \quad \hat{\mathbf{x}}_{j+1} = \mathbf{W}_0 \mathbf{x}_j + \mathbf{W}_1 \mathbf{x}_{j-1} + \ldots + \mathbf{W}_L \mathbf{x}_{j-L} \]

Part 1

Operator-theoretic Learning & Decomposition
Koopman operator and linear embedding

Dynamical system $\dot{x} = f(x)$ ; $x \in \mathcal{M} \subset \mathbb{R}^N$

Dynamics of observables $^1 \mathcal{K}_t h = h \circ \phi_t,$

Koopman operator $\mathcal{K}_t : \mathcal{F} \mapsto \mathcal{F}, \mathcal{F} = L^2(\mathcal{M}, \mu)$

$h$ is any observable in $\mathcal{F}$, $\phi_t$ is the flow.

Significance

$\mathcal{K}_t$ is linear $\rightarrow$ global linearization, but inherently infinite-dimensional.

$^1$Koopman (1931), Mezić (2005)
Connections of Koopman to other operators

Dynamical system \( \dot{x} = f(x) \); \( x \in \mathcal{M} \subset \mathbb{R}^N \)

Liouville operator \( \mathcal{L} := f \cdot \nabla_x \)

Liouville PDE \( \frac{\partial u}{\partial t} = \mathcal{L}u \); \( u(\cdot, 0) = h \)

Generator \( \mathcal{K}_t h = h \circ \phi_t = e^{t\mathcal{L}} h \)

Liouville generates Koopman

Perron-Frobenius operator \( \rho(\cdot, t) = \mathcal{P}_t \circ \rho \)

Duality \( \langle h, \mathcal{P}_t \rho \rangle = \langle \mathcal{K}_t h, \rho \rangle \)

Perron-Frobenius is adjoint of Koopman
Spectral expansion of Koopman operators

\[ \mathcal{K}_t h(x) \triangleq h \circ \phi_t = \sum_{j=0}^{\infty} e^{\lambda_j t} \phi_j(x) s_j + K_t^r h(x) \]

Point spectral resolution of Koopman operator

\[ \mathcal{K}_t h(x) \triangleq h \circ \phi_t = \sum_{j=0}^{D} e^{\lambda_j t} \phi_j(x) s_j, \]

where

- \( \mathcal{F}_D = \text{span}\{\phi_1, \ldots, \phi_D\} \subset \mathcal{F} \): a finite Koopman invariant subspace that contains \( h \)
- \( s_j \): Koopman modes, determined by projecting \( h \) onto \( \mathcal{F}_D \)

Applications

Dynamical systems analysis, Optimal control, modal decomposition, model reduction, etc.
Koopman operators & “Deep” Learning

Several works since 2018

<table>
<thead>
<tr>
<th>Previous works</th>
<th>continuous /discrete</th>
<th>nonlinear reconstruction</th>
<th>continuous spectrum</th>
<th>uncertainty</th>
<th>stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yeung et al. [81]</td>
<td>discrete</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Li et al. [43]</td>
<td>discrete</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Takeishi et al. [73]</td>
<td>discrete</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Otto and Rowley [60]</td>
<td>discrete</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Lusch et al. [48]</td>
<td>discrete</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Morton et al. [58]</td>
<td>discrete</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>Erichson et al. [17]</td>
<td>discrete</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Our framework</td>
<td>continuous</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Extracting a Koopman-invariant subspace

Goal: Extracting the Koopman operator defined on $\mathcal{F} = L^2(\mathcal{M}, \mu)$.

Observation functionals $\phi : \mathcal{M} \mapsto \mathbb{R}$: $\|\phi\|_{\mathcal{F}} \triangleq \sqrt{\int_{\mathcal{M}} |\phi|^2 d\mu} < \infty$.

$\mathcal{J}[\Phi] = \max_{\psi \in \{\phi_1, \ldots, \phi_D\}} \min_{f \in \mathcal{F}_D} \|f - \mathcal{K}\psi\|_{\mathcal{F}}^2$

We are also interested in retrieving the state $x$.

Arbabi & Mezic 2019
Enforcing structure for Learning: “Physics information”

Searching the set of Koopman eigenfunctions $\Phi(x)$:

$$\Phi(x) = [\phi_1(x) \quad \phi_2(x) \quad \ldots \quad \phi_D(x)] \in \mathcal{F}_D \subset \mathcal{F},$$

such that

$$\frac{d\Phi}{dt} \triangleq \frac{dx}{dt} \cdot \nabla_x \Phi = f(x) \cdot \nabla_x \Phi = \Phi(x)K$$

Key observations

- if we knew the physics: $f(x)$, we can sample uniformly over the domain of interest without the need to generate trajectory
- if $\Phi$ is fixed, classical KDMD is equivalent to the Galerkin method with Lebesgue measure over the domain of interest
- Need $\Psi : \mathbb{R}^D \rightarrow \mathcal{M}$, such that $\Psi \circ \Phi = \mathcal{I}$, Therefore, we can recover the state $x$ from $\Phi$. 
Enforcing structure for Learning: Tractable optimization

\[ \mathcal{J}[\Phi] = \max_{\psi \in \{\phi_1, \ldots, \phi_D\}} \min_{f \in \mathcal{F}_D} \| f - \mathcal{K}\psi \|^2_{\mathcal{F}} \]

\[ \Phi^* = \arg\min_{\Phi \in \mathcal{F}_D, \exists \Psi: \mathbb{R}^D \rightarrow \mathcal{M}} \mathcal{J}[\Phi] \]

\[ \Psi \circ \Phi = \mathcal{I} \]

\[ W^*_\Phi = \arg\min_{W_\Phi, \exists \Psi \in \mathcal{C}(\mathbb{R}^D, \mathbb{R}^N)} \mathcal{J}[\Phi(\cdot; W_\Phi)] \]

\[ \Psi \circ \Phi(\cdot; W_\Phi) = \mathcal{I} \]

\[ \tilde{W}_\Phi, \tilde{W}_\Psi, \hat{K} = \arg\min_{W_\Phi, W_\Psi, K} \tilde{\mathcal{J}}[\Phi(\cdot; W_\Phi), K] + \mathcal{R}[\Phi(\cdot; W_\Phi), \Psi(\cdot; W_\Psi)] \]

\[ \tilde{\mathcal{J}}[\Phi, K] = \| \Phi K - \mathcal{K}\Phi \|^2_{\mathcal{F}_D} \]

\[ \mathcal{R}[\Phi, \Psi] = \| \Psi \circ \Phi - \mathcal{I} \|^2_{\mathcal{F}_N} \]
"Data-free", "Physics-informed"

\[
\begin{align*}
\hat{W}_\Phi, \hat{W}_\Psi, \hat{K} &= \text{argmin}_{W_\Phi, W_\Psi, K} \tilde{J}[^\Phi(\cdot; W_\Phi), K] + R[^\Phi(\cdot; W_\Phi), ^\Psi(\cdot; W_\Psi)] \\
\tilde{J}[^\Phi, K] &= \|^{\Phi}K - ^\mathcal{K}^{\Phi}\|^2_{\mathcal{F}_D} \\
R[^\Phi, ^\Psi] &= \|^{\Psi} \circ ^\Phi - ^\mathcal{I}\|^2_{\mathcal{F}_N}
\end{align*}
\]

Trajectory data, "Unknown physics"

\[
\begin{align*}
\hat{W}_\Phi, \hat{W}_\Psi, \hat{K} &= \text{argmin}_{W_\Phi, W_\Psi, K} \tilde{J}_r[^\Phi(\cdot; W_\Phi), K] + \tilde{P}[^\Phi(\cdot; W_\Phi), ^\Psi(\cdot; W_\Psi), K] \\
\tilde{J}_r[^\Phi, K] &= \|^{\Phi}e^{tK} - ^\mathcal{K}_t^{\Phi}\|^2_{G_D} \\
\tilde{P}[^\Phi, ^\Psi, K] &= \|^{\Psi} \circ ^\Phi e^{tK} - ^\mathcal{K}_t^\mathcal{I}\|^2_{G_N}
\end{align*}
\]
Enforcing structure for Learning: “DMD ResNet”

\[ \Phi_{svd}(z) = z\Lambda V_D, \quad \Psi_{svd}(\Phi) \triangleq \Phi V_D^\top \Lambda^{-1}, \]
\[ \Phi(z) = \Phi_{nn}(z)W_{enc,L} + b_{enc,L} + \Phi_{svd}(z)W_{enc,L}, \]
\[ \Psi(\Phi) = \Psi_{nn}(\Phi) + \Psi_{svd}(\Phi W_{dec,1}) , \]

(a) without skip connections  

(b) with skip connections

Figure 1: The loss surfaces of ResNet-56 with/without skip connections. The proposed filter normalization scheme is used to enable comparisons of sharpness/flatness between the two figures.
Enforcing structure for Learning : Stability

We propose: (where $\zeta_1, \ldots, \zeta_{D-1}, \sigma_1, \ldots, \sigma_D \in \mathbb{R}$.)

$$K_{\text{stable}} = \begin{bmatrix}
-\sigma_1^2 & \zeta_1 \\
-\zeta_1 & \ddots & \ddots \\
\ddots & \ddots & \ddots \\
-\zeta_{D-1} & -\sigma_D^2 & \zeta_{D-1}
\end{bmatrix}, \quad (5)$$

Theorem

$\forall D \in \mathbb{N}$, for any real square diagonalizable matrix $K \in \mathbb{R}^{D \times D}$ that only has non-positive real part of the eigenvalues $D \geq 2$, there exists a set of $\zeta_1, \ldots, \zeta_{D-1}, \sigma_1, \ldots, \sigma_D \in \mathbb{R}$ such that $K_{\text{stable}}$ is similar to $K$ over $\mathbb{R}$. Moreover, for any $\zeta_1, \ldots, \zeta_{D-1}, \sigma_1, \ldots, \sigma_D \in \mathbb{R}$, the real part of the eigenvalue of $K_{\text{stable}}$ is non-positive.

Unconditionally stable, and “expressive” → any diagonalizable matrix corresponding to a stable Koopman operator can be represented without loss of information.
Naïve “Autoencoders”

\[
\frac{1}{M} \sum_{i=1}^{M} \| \Psi(\Phi(x_i)) - x_i \|^2
\]
Bayesian Neural Networks & Variational Inference

- a faster alternative to MCMC for Bayesian inference
- traditionally requires tedious model-specific derivations and implementation
  - Automatic Differentiation Variational Inference (ADVI)³ leverages automatic differentiation (AD) to makes implementation of VI easier
- we build our framework based on Tensorflow + ADVI functionality in Edward⁴
Variational Inference

- minimize the KL divergence: \( \min_\xi \text{KL}(q(\Theta; \xi) || P(\Theta | D)) \)

- equivalently maximize the evidence lower bound (ELBO)

\[
\hat{\xi} = \arg \max_\xi \mathcal{L}(\xi) = \arg \max_\xi (\mathbb{E}_q[\log P(\Theta, D)] - \mathbb{E}_q[\log q(\Theta; \xi)])
\]

by re-parameterization-trick + stochastic gradient descent

\[
\Theta \sim \mathcal{N}(0, \Lambda)
\]

\[
D|\Theta \sim \prod_{i=1}^M \mathcal{N}\left( \begin{bmatrix} \Psi(\Phi(x_i)) - x_i \\ f(x_i) \cdot \nabla_x \Phi(x_i) - \Phi(x_i)K \end{bmatrix} ; \begin{bmatrix} 0 \\ 0 \end{bmatrix} , \begin{bmatrix} \sigma_{\text{rec}}^2 I & 0 \\ 0 & \sigma_{\text{lin}}^2 I \end{bmatrix} \right)
\]
Verification on Model dynamical systems

Duffing oscillator: Eigenfunctions (with uncertainty)

Prediction and sensitivity to data

- 100 data points.
- 1000 data points
- 10000 data points
Flow over cylinder: Prediction with uncertainties

• Gaussian white noise added
Flow over cylinder: Prediction with uncertainties

Physics-Informed Probabilistic Learning of Linear Embeddings of Non-linear Dynamics With Guaranteed Stability, Pan, S., and Duraisamy, K., SIADS, 2020
Multi-task learning framework to extract sparse Koopman-invariant subspaces

Steps:

- a-priori cross validation to choose an appropriate hyperparameter
- mode-by-mode error analysis
- choose a trade-off between reconstruction error and linear evolving error
- sparse reconstruction of system with multi-task learning

Turbulent Ship Airwake

- transient behavior is accurately reconstructed
- stable modes are successfully extracted from strongly nonlinear transient data
- left mode: due to side edge of superstructure. right mode: due to funnel

Summary

- Expressibility of deep neural nets $\rightarrow$ rich $\Omega_K$
- Nonlinear reconstruction $\rightarrow$ linear embedding
- Differential form $\rightarrow$ known governing eqns / no data and recurrent form $\rightarrow$ trajectory data
- Guaranteed stability
- SVD-DMD as a short-cut similar to ResNet
- Mean-field variational inference (MFVI) with hierarchical Bayesian model for uncertainty

Many opportunities to enforce structure in Autoencoders $\rightarrow$ flexible and powerful tools
Part 2

Learning Reduced Order Models of Parametric Spatio-temporal dynamics
Non-intrusive data-driven ROMs

\[ q^{n+1}_l = f(q^{n+1}_l, q^n_l, \ldots q^{n-l}_l, B(u^{n+1}), \mu) \]

Some recent works:

- B. Kramer, K. E. Willcox, AIAA Journal, 2019
- A. Mohan, D. Daniel, M. Chertkov, D. Livescu, arXiv, 2019
Basic Component: Convolutional Layer

- Convolutional layers preserve complex spatio-temporal “information”
- Convolutional operation on a local window $w$
  \[
  (x * w)_{ij} = \sum_{p=a}^{a} \sum_{q=b}^{b} x_{i-p,j-q} w_{p,q}
  \]
- Ideal for “localized” feature identification
- Rotation and translation invariant, if properly constructed

“Applied Deep Learning”
https://towardsdatascience.com/

http://cs231n.github.io/convolutional-networks/
Temporal Convolutional

- Performs dilated 1D convolutional operation in temporal/sequential direction
  \[ (x *_{d} w)_i = \sum_{p=0}^{k-1} x_{i-dp}w_p \]
- Exponential increase in reception field \( \Rightarrow \) an increasingly popular alternative to RNN/LSTM

Output/Conv-4: \( d=8 \)
reception field: 16

Conv-3: \( d=4 \)
reception field: 8

Conv-2: \( d=2 \)
reception field: 4

Conv-1: \( d=1 \)
reception field: 2

Input sequence

Source of image: github.com/philipperemy/keras-tcn
Training Multi-level convolutional AE networks

Level 1: CAE

Input: individual frames $q^i \in \mathbb{R}^n$ in sequence $Q = [q^1 \cdots q^n] \in \mathbb{R}^{n \times n}$

Encoder $\Phi^s : \mathbb{R}^n \rightarrow \mathbb{R}^{n_x}$

Code: $q^s_i \in \mathbb{R}^{n_x}$

Decoder $\Psi^s : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^n$

Output: $\tilde{q}^i$
Example CAE architecture
Prediction using Multilevel AE networks

**Level 1: CAE**
- **Input:** individual frames $q^i \in \mathbb{R}^n$ in sequence $Q = [q^1 \ldots q^{n_f}] \in \mathbb{R}^{n \times n_f}$
- **Encoder** $\Phi^x$: $\mathbb{R}^n \rightarrow \mathbb{R}^{n_z}$
- **Code:** $q_i^c \in \mathbb{R}^{n_z}$
- **Decoder** $\Psi^x$: $\mathbb{R}^{n_z} \rightarrow \mathbb{R}^n$
- **Output:** $\tilde{q}^i$

**Level 2: TCAE**
- **Input:** assembled sequence $Q_s = [q_1^s \ldots q_{n_f}^s] \in \mathbb{R}^{n_s \times n_f}$
- **Encoder** $\Phi^z$: $\mathbb{R}^{n_s \times n_f} \rightarrow \mathbb{R}^{n_z}$
- **Code:** $q_i \in \mathbb{R}^{n_z}$
- **Decoder** $\Psi^z$: $\mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_s \times n_f}$
- **Output:** $\tilde{Q}_s$

**Level 3: MLP**
- **Input:** parameter $\mu$
- **Regressor**
- **Output:** $q_\tau$ at $\mu$
Example TCAE architecture
Prediction using Multilevel AE networks

Level 1: CAE
- Input: individual frames $q^i \in \mathbb{R}^n$ in sequence $Q = [q^1 \ldots q^n]$ $\in \mathbb{R}^{n \times n_t}$
- Different variables treated as channels
- Encoder $\Phi^s: \mathbb{R}^n \rightarrow \mathbb{R}^{n_s}$
- Code: $q^s_i \in \mathbb{R}^{n_s}$
- Decoder $\Psi^s: \mathbb{R}^{n_s} \rightarrow \mathbb{R}^n$
- Output: $\tilde{q}^i$

Level 2: TCAE
- Input: assembled sequence $Q_s = [q^s_1 \ldots q^s_n]$ $\in \mathbb{R}^{n_s \times n_t}$
- Encoder $\Phi^t: \mathbb{R}^{n_s \times n_t} \rightarrow \mathbb{R}^{n_t}$
- Code: $q^t_i \in \mathbb{R}^{n_t}$
- Decoder $\Psi^t: \mathbb{R}^{n_t} \rightarrow \mathbb{R}^{n_s \times n_t}$
- Output: $\tilde{Q}_s$

Level 3: MLP
- Input: parameter $\mu$
- Regressor
- Output: $q_t$ at $\mu$
Time stepping

Level 2: many-to-one/many-to-many TCN

Input: assembled sequence
\( Q_{s}^{i,L} = [q_{s}^{i-L+1}, \ldots, q_{s}^{i}] \in \mathbb{R}^{n_{s} \times n_{T}} \) in a look-back window \( \tau \)

Output: \( q_{s} \) for one subsequent future step

Level 1: CAE

Output: frames for predicted future steps \( q_{s}^{i+1}, q_{s}^{i+2}, q_{s}^{i+3} \ldots \)

Forward propagation in training

Prediction
Example TCN architecture

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

*: Convolution direction

Many-to-one

Many-to-many
**Numerical Tests: Discontinuous compressible flow**

### Chart Description

The chart illustrates the comparison between the **Truth** (solid blue line), the **CAE reconstruction** (dashed orange line), and the **End of training** (dotted blue line) for the components Pressure, Density, and Velocity.

### Table of Results

<table>
<thead>
<tr>
<th>Component</th>
<th>CAE reconstruction</th>
<th>CAE + TCN (final step)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training Testing</td>
<td></td>
</tr>
<tr>
<td>Pressure</td>
<td>0.04% 0.1%</td>
<td>0.12%</td>
</tr>
<tr>
<td>Density</td>
<td>0.01% 0.04%</td>
<td>0.14%</td>
</tr>
<tr>
<td>Velocity</td>
<td>0.04% 0.08%</td>
<td>0.13%</td>
</tr>
</tbody>
</table>
Discontinuous compressible flow: Impact of data

(a) $n_t = 20$

(b) $n_t = 30$

(c) $n_t = 40$
Numerical Tests : 3D Ship Airwake

- Incompressible Navier-Stokes
- 576k DOF, 400 time snapshots
- Global parameter: sliding angle $\alpha$
- Training: $\alpha = 5^\circ : 5^\circ : 20^\circ$
- Prediction: $\alpha = 12.5$

$\alpha = 5^\circ$

$\alpha = 20^\circ$
Numerical Test: 3D Ship Airwake

<table>
<thead>
<tr>
<th>Component</th>
<th>CAE reconstruction</th>
<th>MLP + TCAE + CAE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training</td>
<td>Testing</td>
</tr>
<tr>
<td>U</td>
<td>0.12%</td>
<td>0.30%</td>
</tr>
<tr>
<td>V</td>
<td>0.09%</td>
<td>0.38%</td>
</tr>
<tr>
<td>W</td>
<td>0.08%</td>
<td>0.29%</td>
</tr>
</tbody>
</table>

Relative absolute error

Manuscript “Multi-level Convolutional Autoencoder Networks for Parametric Prediction of Spatio-temporal Dynamics,” Submitted CMAME
Summary

- Fully Data-driven framework
  - Multi-level neural network architecture
  - Convolutions in space & time
  - Non-linear manifolds
  - Fast training, faster prediction
  - Up to 6 orders of reduction in DoF
  - Total training time: 3.6 hours on one NVIDIA Tesla P100 GPU for 3D ship air wake
  - Prediction time: Seconds for a new parameter or hundreds of future steps

Caveats

- Require large amounts of data
- No indicator for choice of latent dimensions ⇒ use singular values to find an upper bound

Manuscript “Multi-level Convolutional Autoencoder Networks for Parametric Prediction of Spatio-temporal Dynamics” to be submitted to ArXiv in a week
Acknowledgments

- DARPA Physics of AI program (Technical Monitor: Dr. Ted Senator)
- Air Force Center of Excellence grant (Program Managers: Dr. Mitat Birkan and Dr. Fariba Fahroo)
- Office of Naval Research (Program manager: Dr. Brian Holm-Hansen)
- Computational infrastructure: NSF-MRI (Program manager: Dr. Stefan Robila)
Numerical Tests