SAND2021-12287C

Sandia National Laboratories

Exceptional service in the national interest

Parametric space—time model reduction with deep bases

Eric Parish, Sandia National Laboratories

Yukiko Shimizu, Sandia National Laboratories

Kookjin Lee, Arizona State University

Mechanistic Machine Learning and Digital Twins for Computational Science, Engineering, and Technology

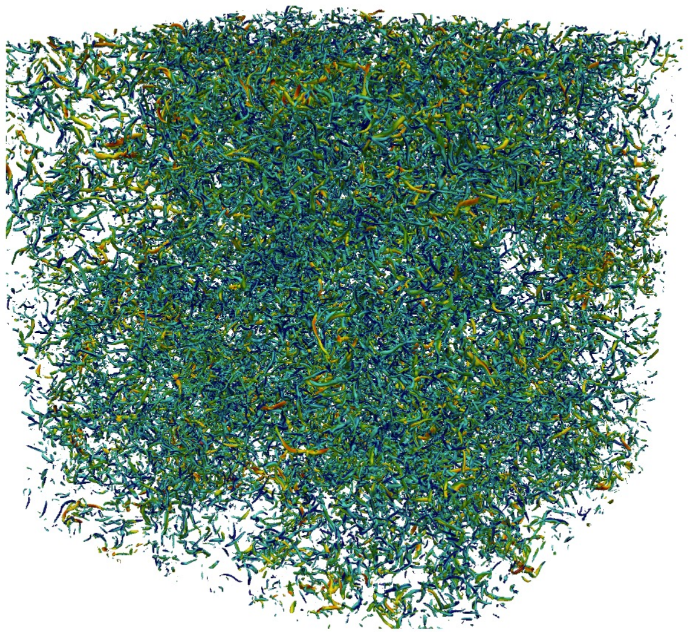
September 26-29, 2021, San Diego, CA.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

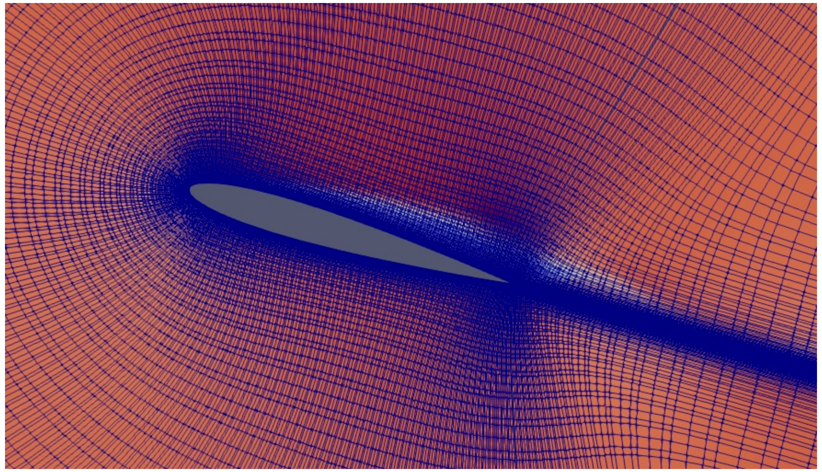
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Motivation

- Many-query applications are ubiquitous in science and engineering
  - Uncertainty quantification
  - Design and optimization
- Repeated executions of high-fidelity simulators can be **computationally expensive**
- Analysts often rely on *approximate* models that provide low-cost approximate solutions
  - Polynomial surrogates
  - Neural networks
  - **Projection-based reduced-order models**
- **This work combines ideas from deep learning and ROMs to make approximate models for convection dominated PDEs**



Homogeneous isotropic turbulence



Computational mesh of NACA 0012 airfoil

Projection-based reduced-order models

- We consider partial differential equations described by

$$\frac{\partial \mathsf{u}}{\partial t}(\boldsymbol{x}, t, \boldsymbol{\mu}) - \mathcal{F}(\mathsf{u}(\cdot, t, \boldsymbol{\mu}), t, \boldsymbol{\mu}) = 0$$

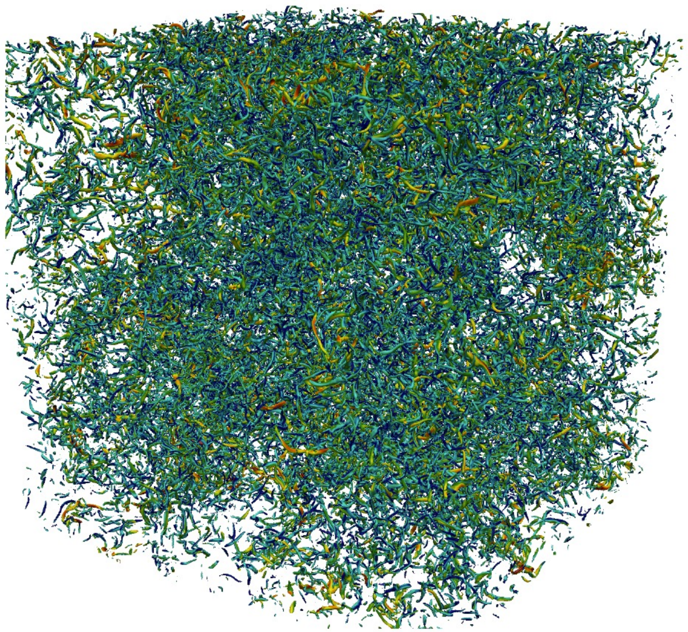
  - PDE state: $\mathsf{u} : \Omega \times [0, T] \times \mathcal{D} \rightarrow \mathbb{R}^{N_v}$
  - Parameters: $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^{N_\mu}$

- We assume a semi-discrete counterpart of the form

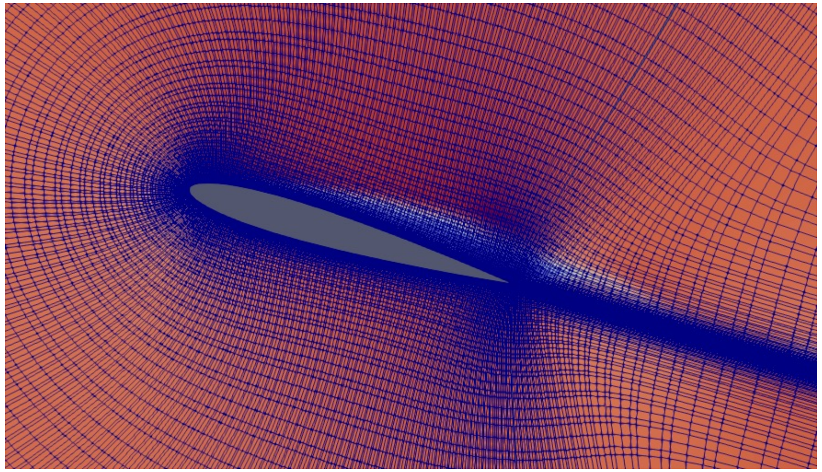
$$\frac{d\boldsymbol{u}}{dt}(t, \boldsymbol{\mu}) = \boldsymbol{f}(\boldsymbol{u}(t, \boldsymbol{\mu}), t, \boldsymbol{\mu})$$

  - Semi-discrete state: $\boldsymbol{u}(t, \boldsymbol{\mu}) \in \mathbb{R}^N$

- Solving these systems is computationally expensive
- Motivates reduced-order models



Homogeneous isotropic turbulence



Computational mesh of NACA 0012 airfoil

Data-driven projection-based reduced-order models

- Operate in an offline-online paradigm

1. Run a set of training simulations to generate snapshot matrix

$$\mathbf{S} \in \mathbb{R}^{N \times N_{\mathtt{train}}}$$

2. Perform dimension reduction on snapshot matrix to find bases

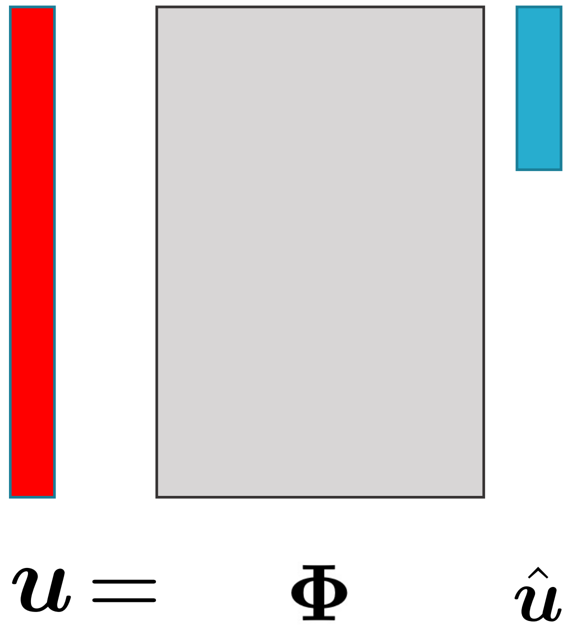
$$\boldsymbol{\Phi} \in \mathbb{R}^{N \times K}$$

3. Restrict state to live within subspace spanned by bases

$$\boldsymbol{u}(t, \boldsymbol{\mu}) \approx \tilde{\boldsymbol{u}}(t, \boldsymbol{\mu}) = \sum_{i=1}^{K} \boldsymbol{\phi}_i \hat{\boldsymbol{u}}_i(t, \boldsymbol{\mu})$$

4. Define ROM, e.g., via Galerkin projection

5. Results in a $K \ll N$ dimensional system



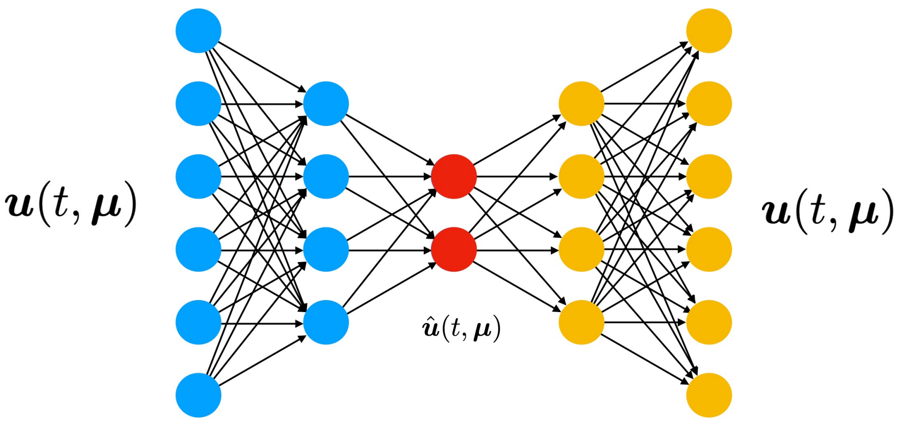
Outstanding challenges and manifold ROMs

- ROMs have been very effective for elliptic and parabolic systems, but…

- Difficult to obtain low-dimensional structure for non-smooth parametric dependencies and convection dominated systems
  - This is the so-called **Kolmogorov n-width limitation**

- Popular alternative: use a nonlinear manifold instead of a linear subspace[1,2]
  - Define a manifold based on, e.g., an autoencoder

$$\boldsymbol{u}(t, \boldsymbol{\mu}) \approx \boldsymbol{g}(\hat{\boldsymbol{u}}(t, \boldsymbol{\mu}))$$

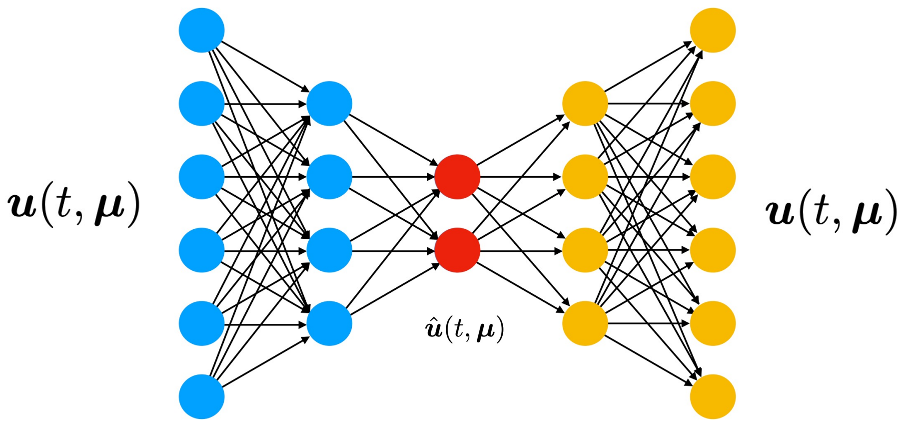


- **Accurate ROMs can be obtained with a low-dimensional manifold**

[1] Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders, K. Lee and C. Carlberg, JCP 2019
[2] A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder, Kim, Y Choi, D Widemann, T Zohdi, arXiv preprint arXiv:2009.11990

Difficulties with manifold ROMs

- Number of parameters in autoencoder scales with the dimension of the full-order model
  - Training manifold autoencoders becomes unfeasible for high-dimensional problems



- Hyper-reduction is non-trivial
  - Requires special modifications to the network architecture
  - For existing methods, training costs still scales with the dimension of the FOM[1]

- Manifold ROMs have not been shown to perform better in extrapolation than standard POD ROMs
  - Perform significantly worse in our experience

- **Motivates an alternative approach**

[1] A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder, Kim, Y Choi, D Widemann, T Zohdi, arXiv preprint arXiv:2009.11990

The Kolmogorov n-width and a matter of perspective

- Consider Burgers' equation as a demonstrative example

$$\frac{\partial \mathrm{u}}{\partial t}(x,t,\boldsymbol{\mu}) + \frac{1}{2}\frac{\partial \mathrm{u}^2}{\partial x}(x,t,\boldsymbol{\mu}) = \frac{1}{50}\exp(\boldsymbol{\mu}_2 x).$$
$$u(x,0,\boldsymbol{\mu}) = 1,\ u(1,0,\boldsymbol{\mu}) = \boldsymbol{\mu}_1,$$

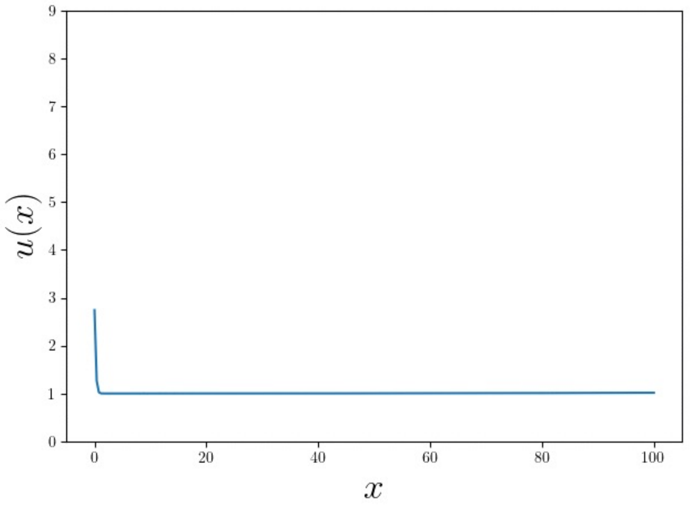
- The Kolmogorov n-width is typically observed from a "spatial" perspective

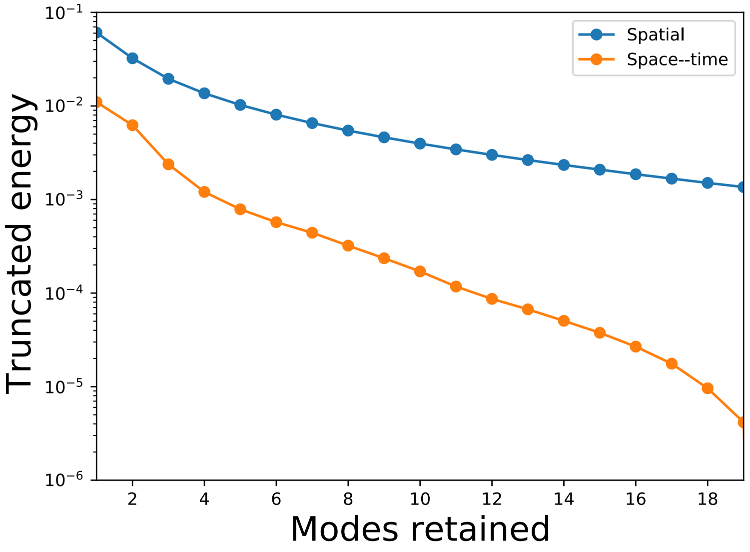
$$\tilde{\mathrm{u}}(\boldsymbol{x},t,\boldsymbol{\mu}) = \sum_{i=1}^{K} \boldsymbol{\phi}_i(\boldsymbol{x})\hat{\boldsymbol{u}}_i(t,\boldsymbol{\mu})$$

- From the space—time perspective we obtain lower-dimensional structure

$$\tilde{\mathrm{u}}(\boldsymbol{x},t,\boldsymbol{\mu}) = \sum_{i=1}^{K} \boldsymbol{\phi}_i(\boldsymbol{x},t)\hat{\boldsymbol{u}}_i(\boldsymbol{\mu})$$

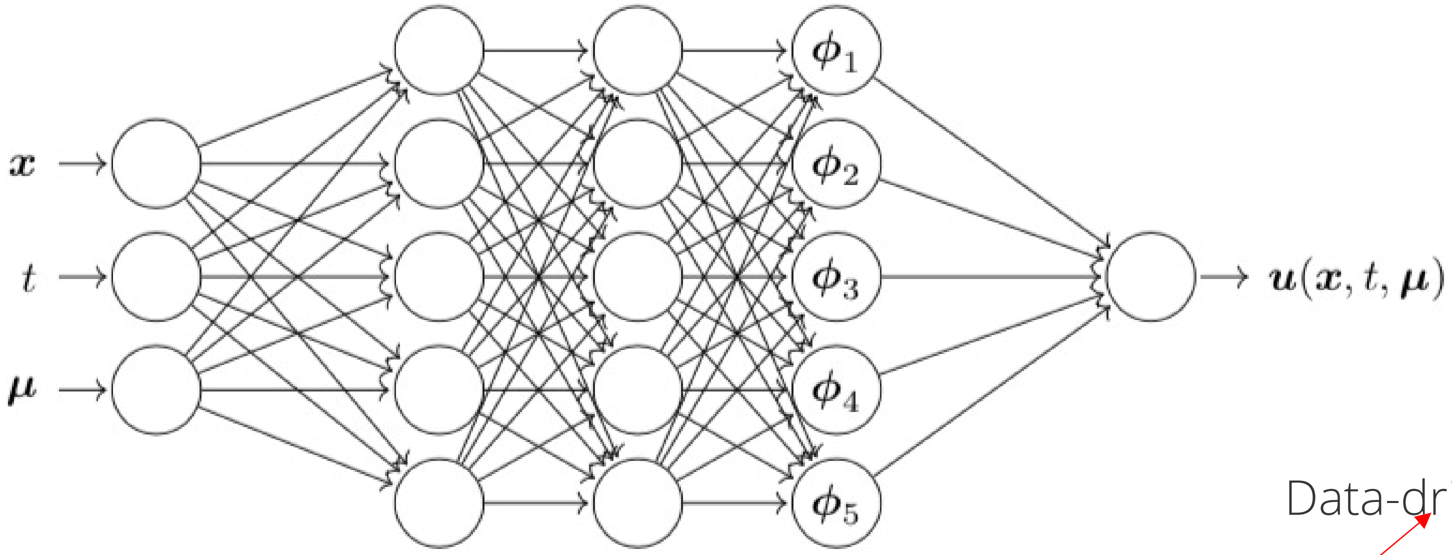
- **The Kolmogorov n-width depends on frame of reference**





Deep bases

- We propose to construct **linear** subspaces that depend on **space, time, and parameters** via deep neural networks
- We employ an architecture similar to that used in Physics-informed Neural Networks (PINNs)[1]



Data-driven basis functions

- Final layer comprises a linear subspace[2]

$$\boldsymbol{u}(\boldsymbol{x}, t, \boldsymbol{\mu}) = \sum_{i=1}^{K} \phi_i(\boldsymbol{x}, t, \boldsymbol{\mu})\hat{\boldsymbol{u}}$$

- **Subspace depends on space, time, and parameters**

M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, JCP, 2019

E. Cyr, M. Gulian, R. Patel, M. Perego, N. Trask, Robust training and initialization of deep neural networks: An adaptive basis viewpoint, PMLR, 2020

Model reduction with deep bases

- We propose an offline—online process for leveraging deep bases
- Offline stage:
  - Simulate system of interest for training instances
  - Train deep bases, e.g., by minimizing the MSE:

$$\boldsymbol{\theta}^*, \hat{\boldsymbol{u}}^* = \arg\min \sum_{i=1}^{N_{\text{train}}} \| \boldsymbol{u}_i^{\text{train}} - \boldsymbol{\Phi}(\boldsymbol{\mu}_i^{\text{train}}, t_i^{\text{train}}; \boldsymbol{\theta}) \hat{\boldsymbol{u}} \|_2$$

  - Project high-fidelity computational model onto low-dimensional subspace spanned by the deep bases

$$\left(\boldsymbol{\phi}_j(\boldsymbol{\mu}, t), \frac{d}{dt} \sum_{i=1}^{K} \boldsymbol{\phi}_i(\boldsymbol{\mu}, t) \hat{\boldsymbol{u}}_i \right) = \left(\boldsymbol{\phi}_j(\boldsymbol{\mu}, t), \boldsymbol{f}\Big(\sum_{i=1}^{K} \boldsymbol{\phi}_i(\boldsymbol{\mu}, t) \hat{\boldsymbol{u}}_i, t, \boldsymbol{\mu}\Big) \right)$$

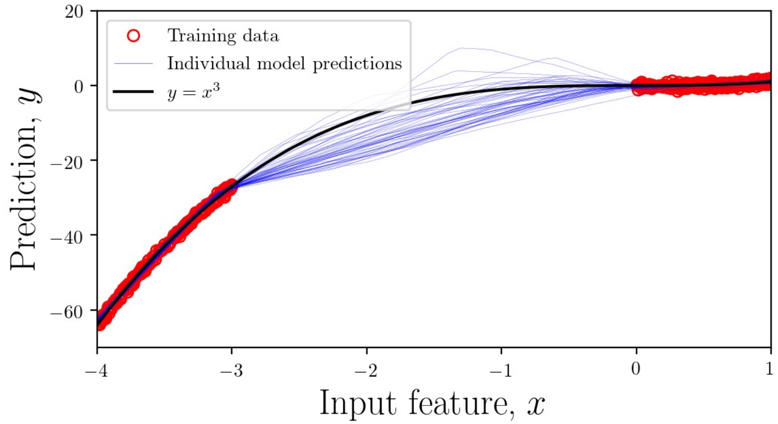
- Online stage:
  - Execute solve of the low-dimensional reduced-order model to obtain approximate solutions

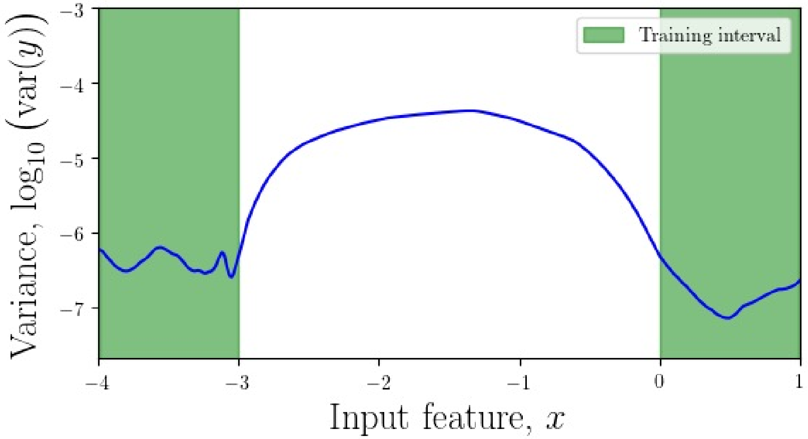
Empirical UQ with deep ensembles

- We need to quantify the accuracy in our ROM
  - Many techniques exist for "in-distribution" predictions
  - Difficult for out-of-distribution (extrapolation)!!!

- "Deep ensembles" is an empirical approach for UQ[1]
  - Identical networks with different initializations result in different predictions for out-of-distribution (extrapolation) data
  - Relies on stochastic training of neural network

- We investigate using ensembles of deep subspaces for empirical UQ





Example of deep ensembles for learning y=x^3

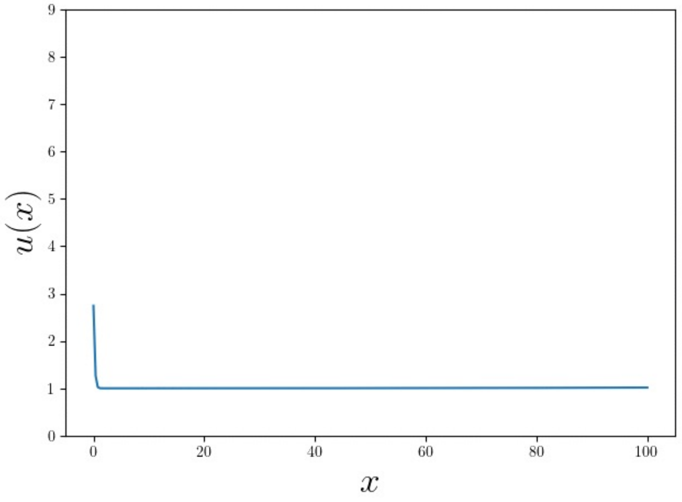
[1] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty using deep ensembles, NEURIPS, 2017.

Numerical example: Burgers Equation

- We consider model reduction of the parameterized Burgers' equation

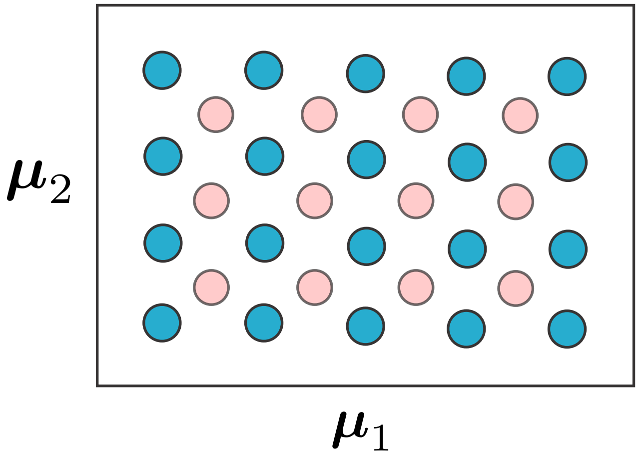
$$\frac{\partial \mathrm{u}}{\partial t}(x,t,\boldsymbol{\mu}) + \frac{1}{2}\frac{\partial \mathrm{u}^2}{\partial x}(x,t,\boldsymbol{\mu}) = \frac{1}{50}\exp(\boldsymbol{\mu}_2 x).$$

$$u(x,0,\boldsymbol{\mu}) = 1,\ u(1,0,\boldsymbol{\mu}) = \boldsymbol{\mu}_1,$$

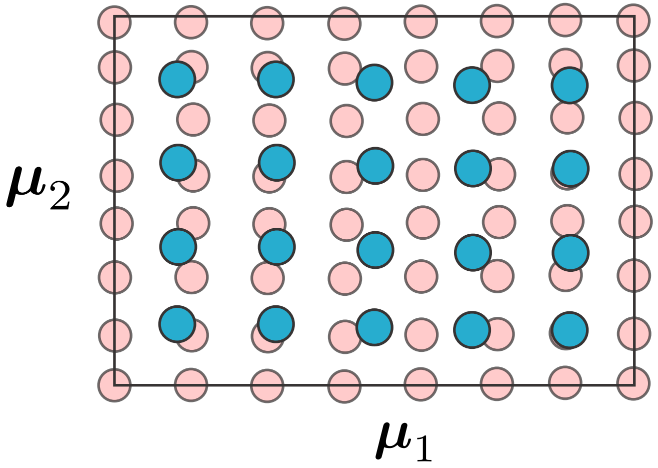


- We consider two training-testing setups

Training point
Testing point



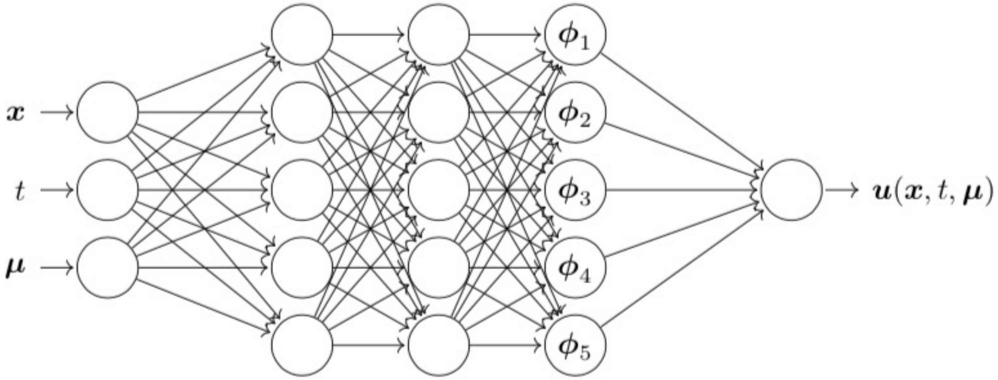
In distribution with no future state-prediction



Out-of-distribution with future state-prediction

Deep basis ROMs details

- Network architecture:
  - Fully connected layers
  - Width doubles at each layer
  - Examine various depths and widths



- Training details:
  - Trained in PyTorch
  - We train on 10% of the snapshot data (randomly sampled)
  - Training performed on GPUs
  - Each network is trained 3 times to quantify stochastic training

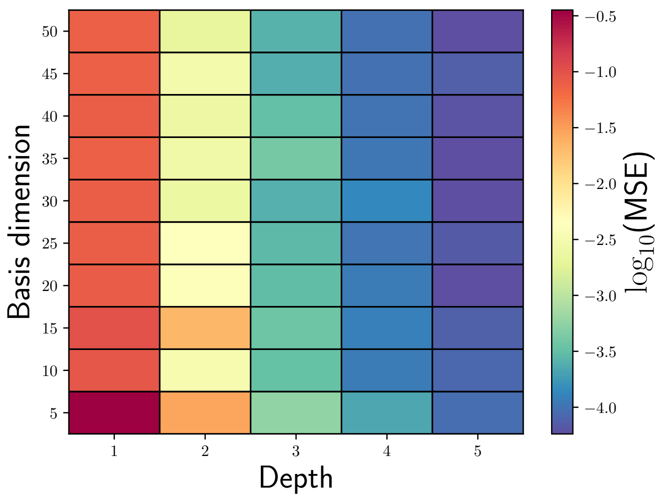
- ROM details:
  - We define our ROM based on ℓ^1 and ℓ^2 norm residual minimization **over all of time, space, and parameters**
  - Residual minimization problem is solved with scipy.optimize.least_squares

Results: ML predictions and a priori projection errors

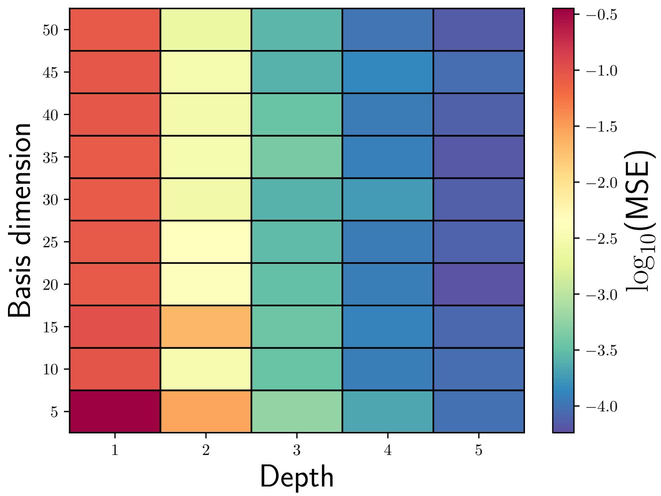
- Examine subspace capacity and generalizability
  - Compare *a priori* projection errors to ML prediction

- ML prediction is similar to optimal projection for I.D. data

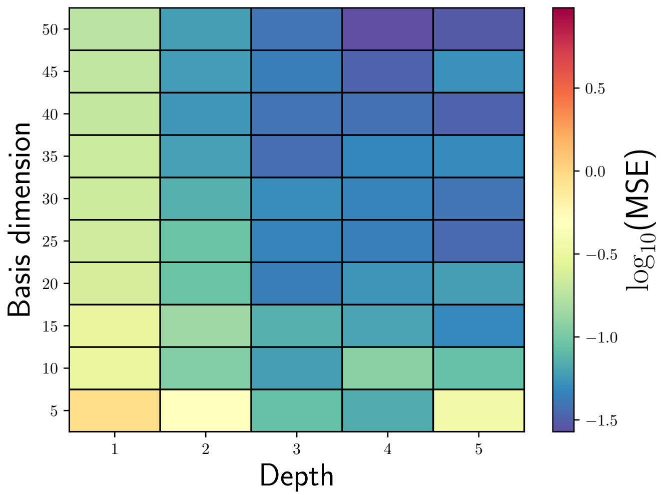
- **Optimal projection is much better for O.O.D. data**



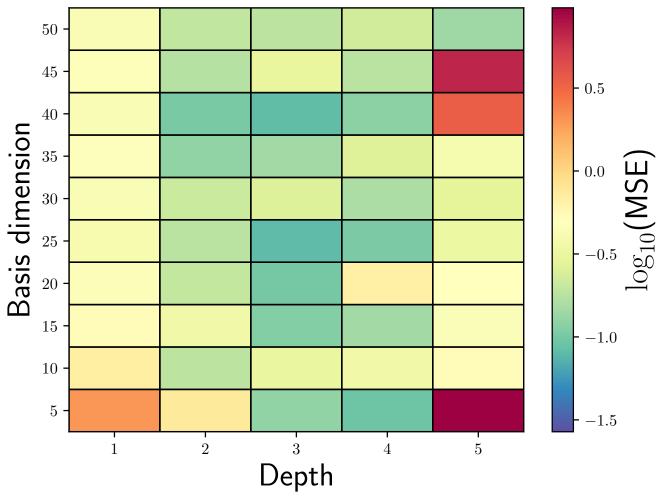
Optimal projection (in distribution)



ML prediction (in distribution)



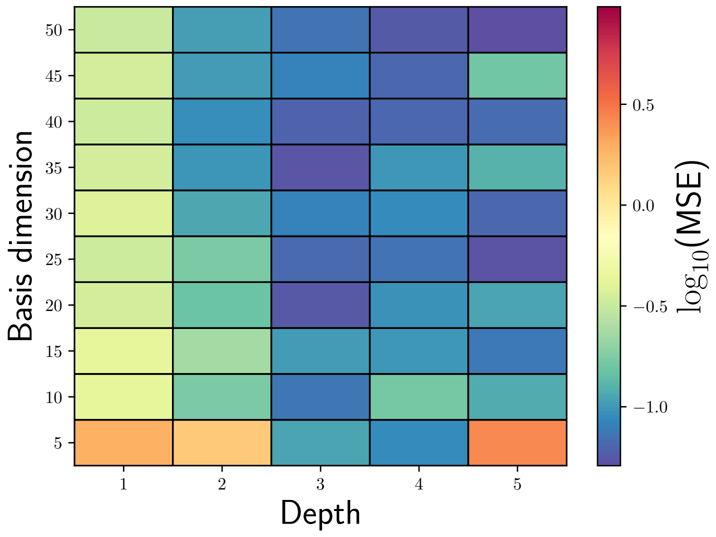
Optimal projection (out of distribution)



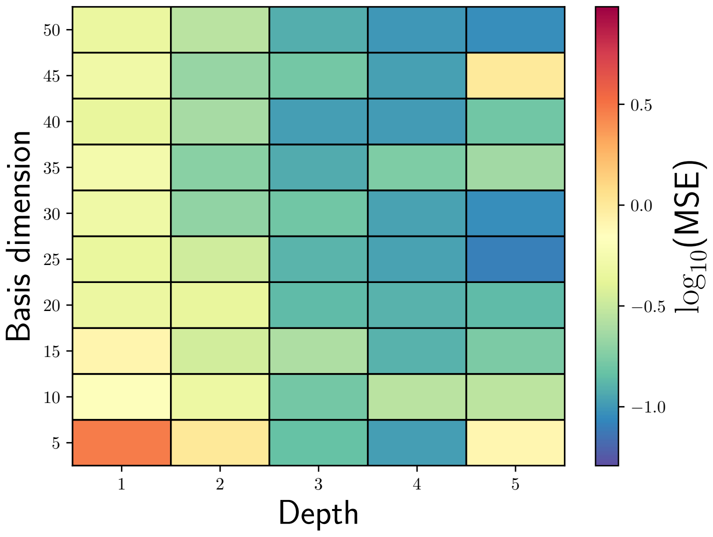
ML prediction (out of distribution)

Results: ROM and ML predictions for O.O.D dataset

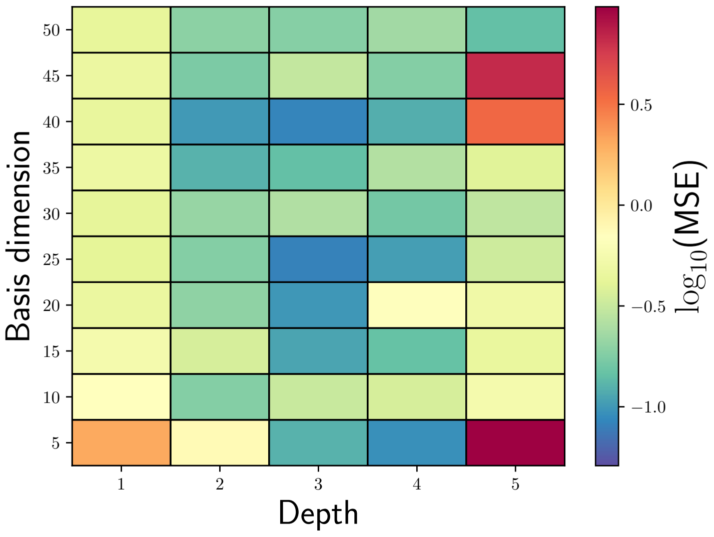
- Compiled errors across all parameter instances



Deep bases (L1 res-min)



Deep bases (L2 res-min)

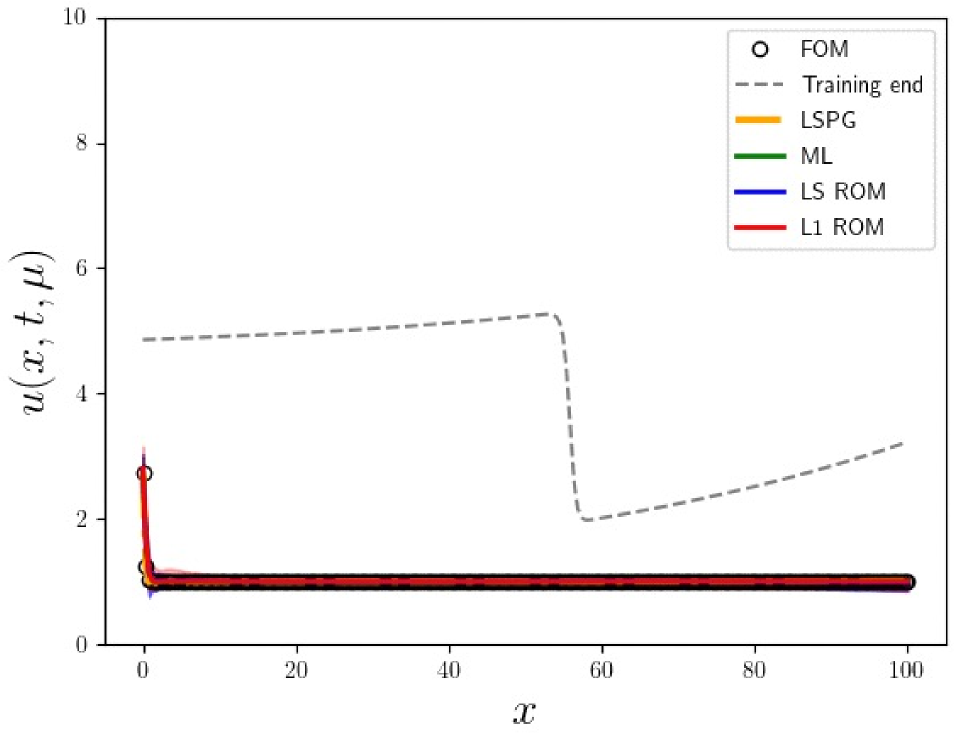


ML

	MSE (avg.)	MSE (median)	MSE (best)	ℓ^∞ (avg.)	ℓ^∞ (average)	ℓ^∞ (best)
Deep Bases-ResMin-ℓ^2	0.351	0.179	0.0411	4.666	4.231	2.620
Deep Bases-ResMin-ℓ^1	0.262	0.100	0.0290	4.298	3.893	2.297
ML	0.713	0.207	0.0445	6.514	4.191	2.870
LSPG ($K = 87$)	1.211	1.211	1.211	7.377	7.377	7.377

Results: ROM and ML predictions for O.O.D dataset

- Examine ML and ROM predictions at $\boldsymbol{\mu}_1 = 1,\ \boldsymbol{\mu}_2 = 3$



- ML and Deep bases ROMs outperform LSPG
- **Deep bases out performs ML in future state prediction**
- **Ensemble provides empirical UQ indicator**

Numerical example: Shallow water equations with Coriolis forcing

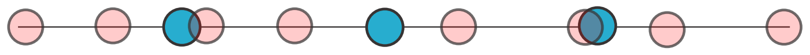
- Consider the shallow water equations parameterized by Coriolis forcing

$$\frac{\partial h}{\partial t} + \frac{\partial hu}{\partial x} + \frac{\partial hv}{\partial y} = 0$$

$$\frac{\partial hu}{\partial t} + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}\rho g h^2\right) + \frac{\partial}{\partial y}(hv) = -\mu v$$

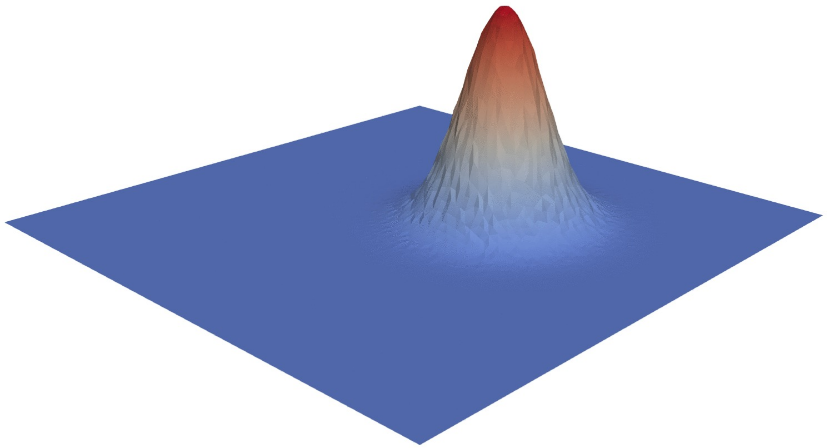
$$\frac{\partial hv}{\partial t} + \frac{\partial}{\partial x}(hu^2) + \frac{\partial}{\partial y}\left(hv + \frac{1}{2}\rho g h^2\right) = \mu u$$

- Testing training setup



Training point

Testing point

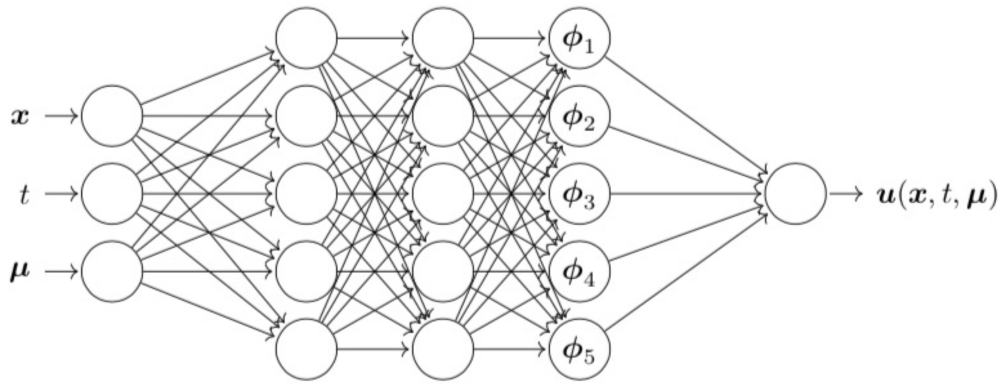


Surface plot of water height

x-velocity

Deep basis ROMs details

- Network architecture:
  - Fully connected layers
  - Width doubles at each layer
  - Set depth =6, final layer dimension = 10

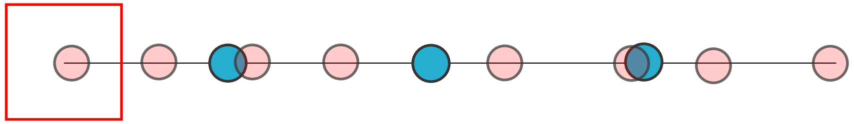


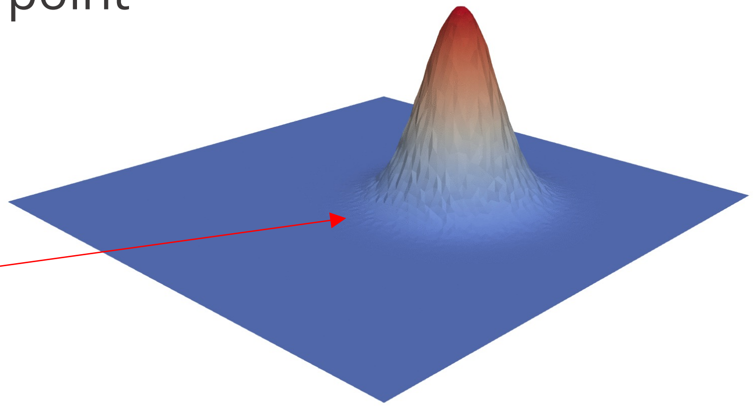
- Training details:
  - Trained in PyTorch
  - We train on 10% of the snapshot data (randomly sampled)
  - Training performed on GPUs
  - Each network is trained 8 times to quantify stochastic training

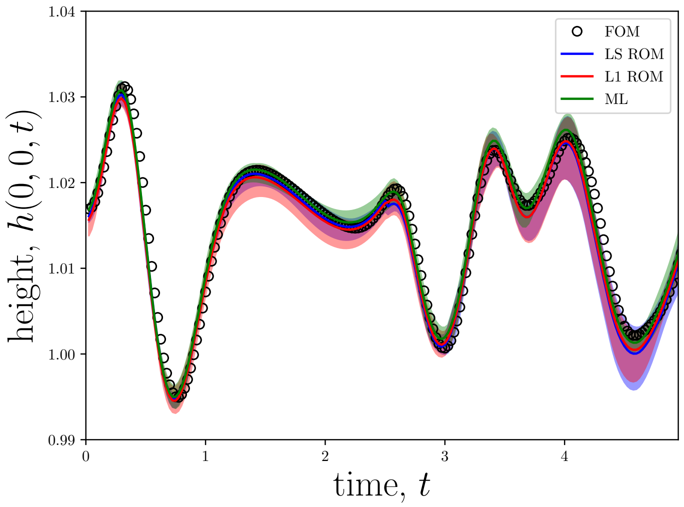
- ROM details:
  - We define our ROM based on ℓ^1 and ℓ^2 norm residual minimization over a time window of 2.5
  - Residual minimization problem is solved with scipy.optimize.least_squares

Results

- Examine prediction for water height in the middle of the domain as a function of time for predictive point







- Ensemble variance grows in time
- Global error metrics:

	MSE (avg.)	MSE (median)	MSE (best)
Deep Bases-ResMin-ℓ^2	0.0264	0.0263	0.0251
Deep Bases-ResMin-ℓ^1	0.0217	0.0252	0.0110
ML	0.0256	0.0256	0.0250

- L1 residual minimization yields the lowest MSE

Conclusions

- Projection-based reduced-order models are promising tools to generate accurate approximate solutions
- Difficult to identify low-dimensional subspaces for convection-dominated problems and problems exhibiting non-smooth parametric dependence
- The Kolmogorov "n-width" depends on the frame of reference
- We are investigating using "deep bases" emerging from fully connected MLPs
  - Stochastic training provides a tool for empirical UQ
- Numerical results on the Burgers' equation and shallow water equations demonstrate the potential of the approach
  - Deep bases ROM with l1 residual minimization outperforms purely data-driven approach in terms of MSE
  - Deep bases ROM with both l2 and l1 residual minimization results in lower residuals

Thank you!

This work was supported by Sandia ASC V&V P/T 103723/05.30.02. This presentation describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

[1] K. Lee and C. Carlberg , Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders, JCP 2019

[2] Kim, Y Choi, D Widemann, T Zohdi, A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder, arXiv preprint arXiv:2009.11990

[3] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, JCP, 2019

[4] E. Cyr, M. Gulian, R. Patel, M. Perego, N. Trask, Robust training and initialization of deep neural networks: An adaptive basis viewpoint, PMLR, 2020

[5] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty using deep ensembles, NEURIPS, 2017.