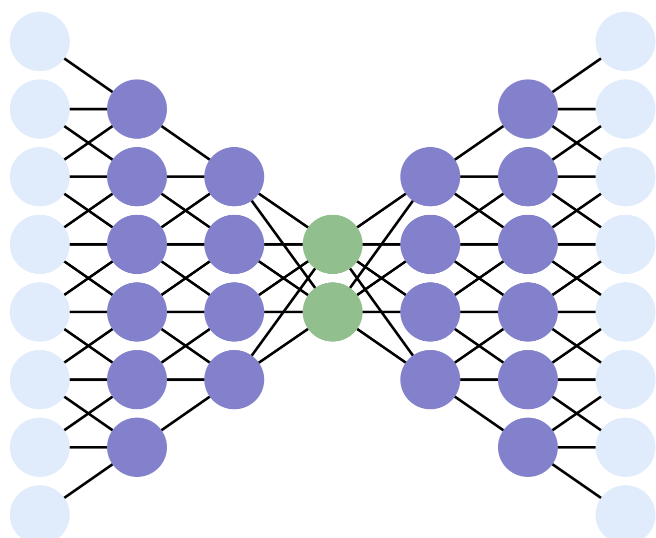
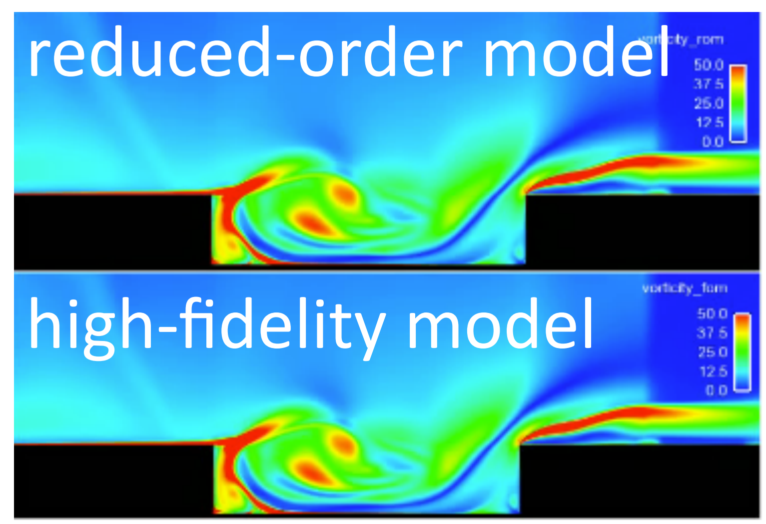
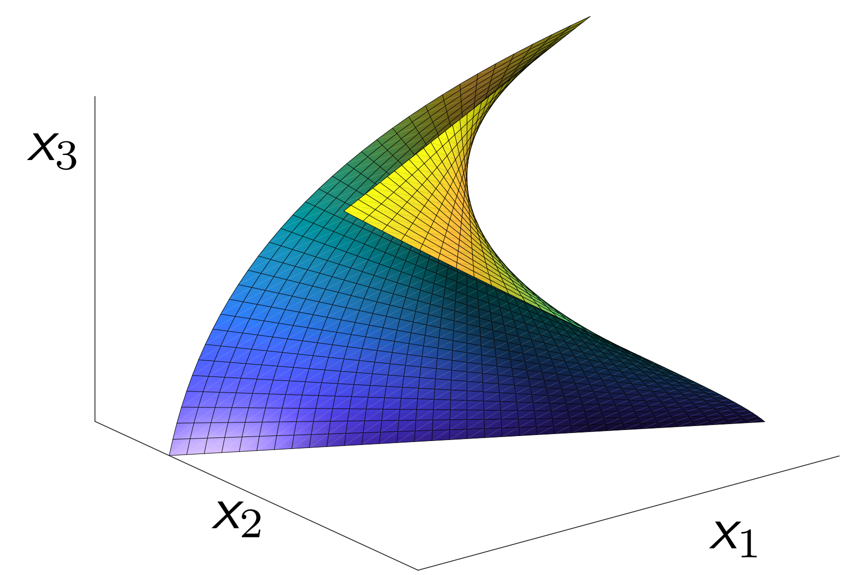
Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders







Kookjin Lee and Kevin Carlberg

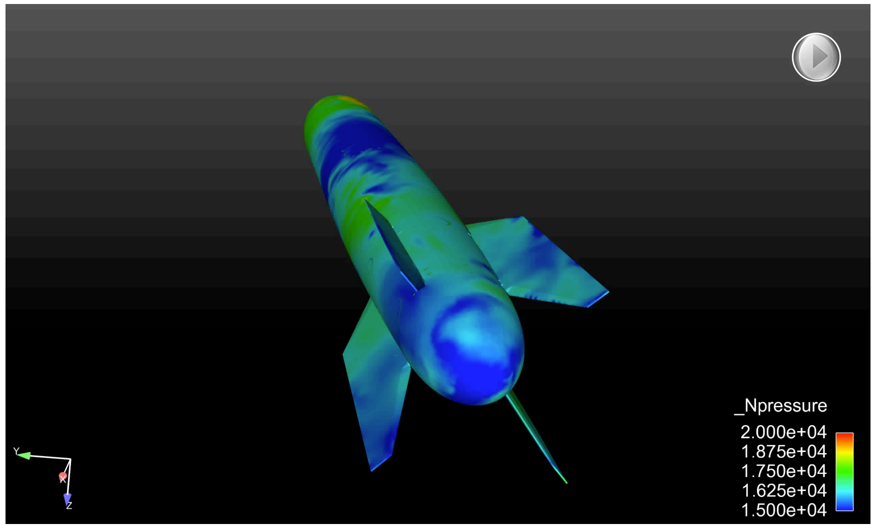
Sandia National Laboratories

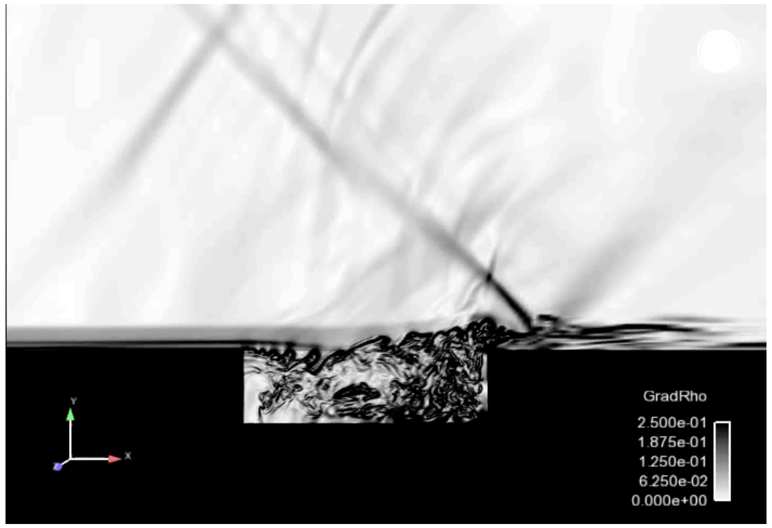
SIAM CSE

Spokane, Washington

High-fidelity simulation

+ Indispensable in science and engineering
- Extreme-scale models required for high fidelity





+ *Validated and predictive*: matches wind-tunnel experiments to within 5%
- *Extreme scale*: 100 million cells, 200,000 time steps
- *High simulation costs*: 6 weeks, 5000 cores

computational barrier

Time-critical applications

◉ design ◉ uncertainty quantification ◉ health monitoring ◉ control

Goal: break computational barrier

How does classical model reduction work?

$$\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) = \boldsymbol{\Phi}\, \hat{\mathbf{x}}(t)$$

FOM ODE

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t)$$

$\xrightarrow{\text{residual minimization}}$

Galerkin ODE

$$\frac{d\hat{\mathbf{x}}}{dt} = \boldsymbol{\Phi}^T \mathbf{f}(\boldsymbol{\Phi}\hat{\mathbf{x}}, t)$$

$\Updownarrow$

$$\mathbf{r}\left(\frac{d\mathbf{x}}{dt}, \mathbf{x}, t\right) = \mathbf{0}$$

$\Updownarrow$

$$\boldsymbol{\Phi}\frac{d\hat{\mathbf{x}}}{dt}(\boldsymbol{\Phi}\hat{\mathbf{x}}, t) = \underset{\mathbf{v} \in \text{range}(\boldsymbol{\Phi})}{\arg\min} \|\mathbf{r}(\mathbf{v}, \boldsymbol{\Phi}\hat{\mathbf{x}}, t)\|_2$$

time discretization $\downarrow$

time discretization $\downarrow$

FOM OΔE

$$\mathbf{r}^n(\mathbf{x}^n) = 0, \quad n = 1, \ldots, T$$

Galerkin OΔE

$$\boldsymbol{\Phi}^T \mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{x}}^n) = 0, \quad n = 1, \ldots, T$$

$\xleftarrow{\text{residual minimization}}$

LSPG OΔE

[C., Bou-Mosleh, Farhat, 2011]

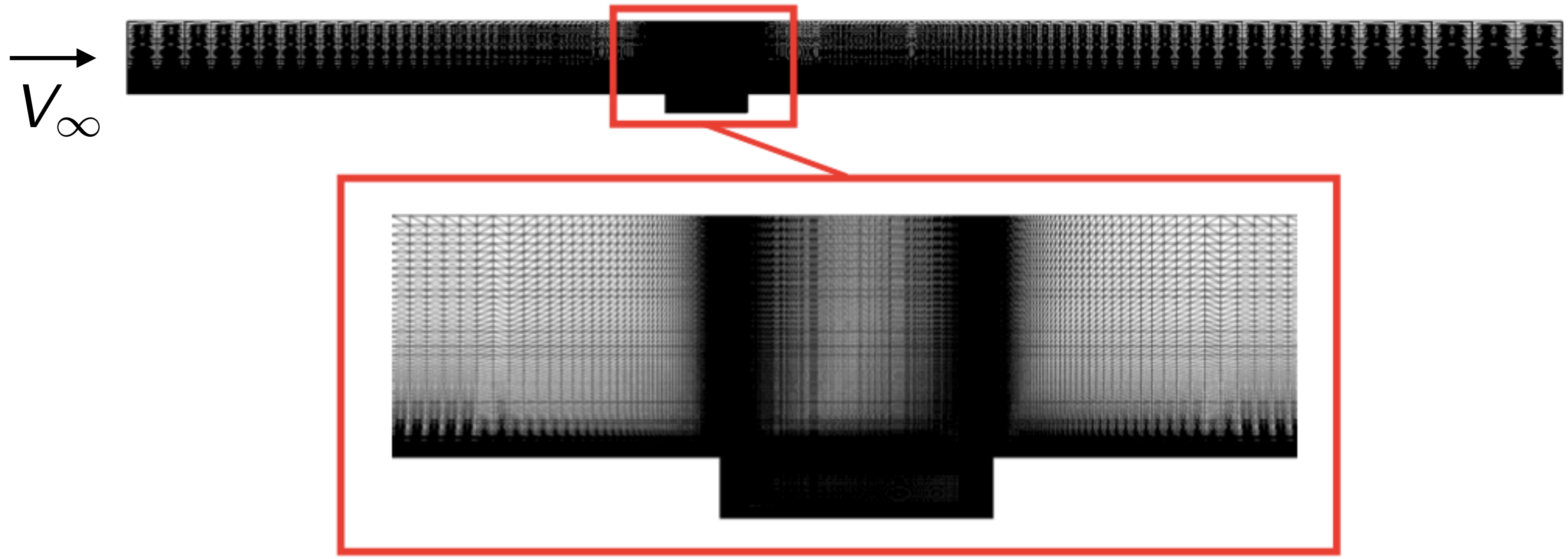
$$\boldsymbol{\Psi}^n(\hat{\mathbf{x}}^n)^T \mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{x}}^n) = \mathbf{0}, \quad n = 1, \ldots, T$$

$\Updownarrow$

$$\boldsymbol{\Phi}\hat{\mathbf{x}}^n = \underset{\mathbf{v} \in \text{range}(\boldsymbol{\Phi})}{\arg\min} \|\mathbf{r}^n(\mathbf{v})\|_2, \quad n = 1, \ldots, T$$

- FOM ODE residual: $\mathbf{r}(\mathbf{v}, \mathbf{x}, t) := \mathbf{v} - \mathbf{f}(\mathbf{x}, t)$
- FOM OΔE residual: $\mathbf{r}^n(\mathbf{w}) := \alpha_0 \mathbf{w} - \Delta t \beta_0 \mathbf{f}(\mathbf{w}, t^n) + \sum_{j=1}^{k} \alpha_j \mathbf{x}^{n-j}(\boldsymbol{\nu}) - \Delta t \sum_{j=1}^{k} \beta_j \mathbf{f}(\mathbf{x}^{n-j}, t^{n-j})$
- LSPG test basis: $\boldsymbol{\Psi}^n(\hat{\mathbf{w}}) := \left(\alpha_0 \mathbf{I} + \beta_0 \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\boldsymbol{\Phi}\hat{\mathbf{w}}, t^n)\right)\boldsymbol{\Phi}$
- Detailed comparative analysis: C, Barone, Antil, *J Comp Phys*, 2017.

Captive carry



- Unsteady Navier–Stokes
- Re = 6.3 x 10^6
- M_∞ = 0.6

Spatial discretization

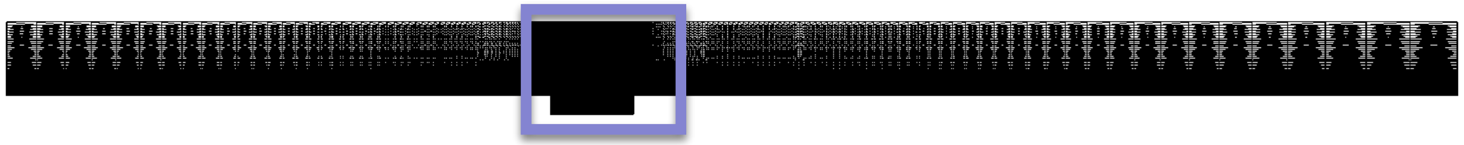
- 2nd-order finite volume
- DES turbulence model
- 1.2×10^6 degrees of freedom

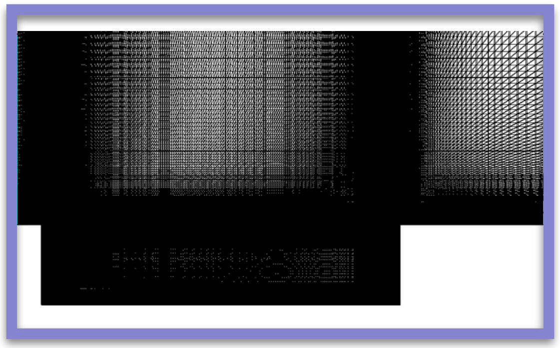
Temporal discretization

- 2nd-order BDF
- Verified time step $\Delta t = 1.5 \times 10^{-3}$
- 8.3×10^3 time instances

LSPG ROM with sample mesh [C., Barone, Antil, 2017]

$$\mathbf{\Phi}\hat{\mathbf{x}}^n = \underset{\mathbf{v}\in\text{range}(\mathbf{\Phi})}{\arg\min} \|\mathbf{r}^n(\mathbf{v})\|_2$$

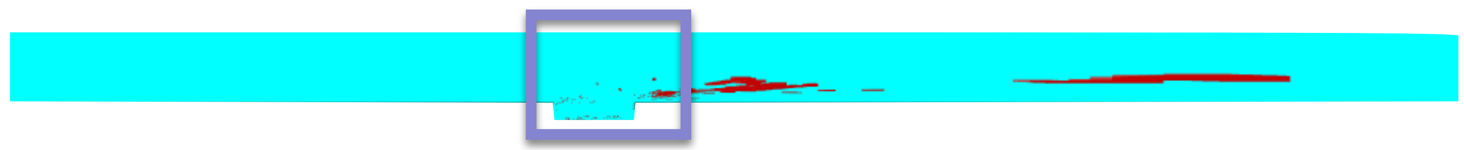




LSPG ROM with sample mesh [C., Barone, Antil, 2017]

$$\boldsymbol{\Phi}\hat{\mathbf{x}}^n = \underset{\mathbf{v}\in\mathrm{range}(\boldsymbol{\Phi})}{\arg\min} \|\mathbf{r}^n(\mathbf{v})\|_{\Theta}$$

sample mesh

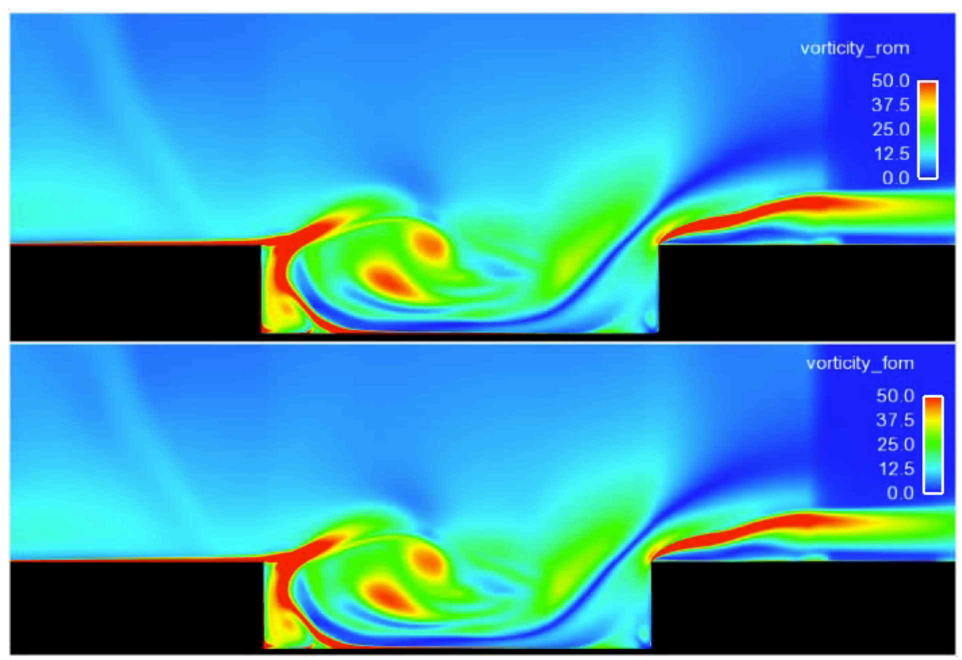


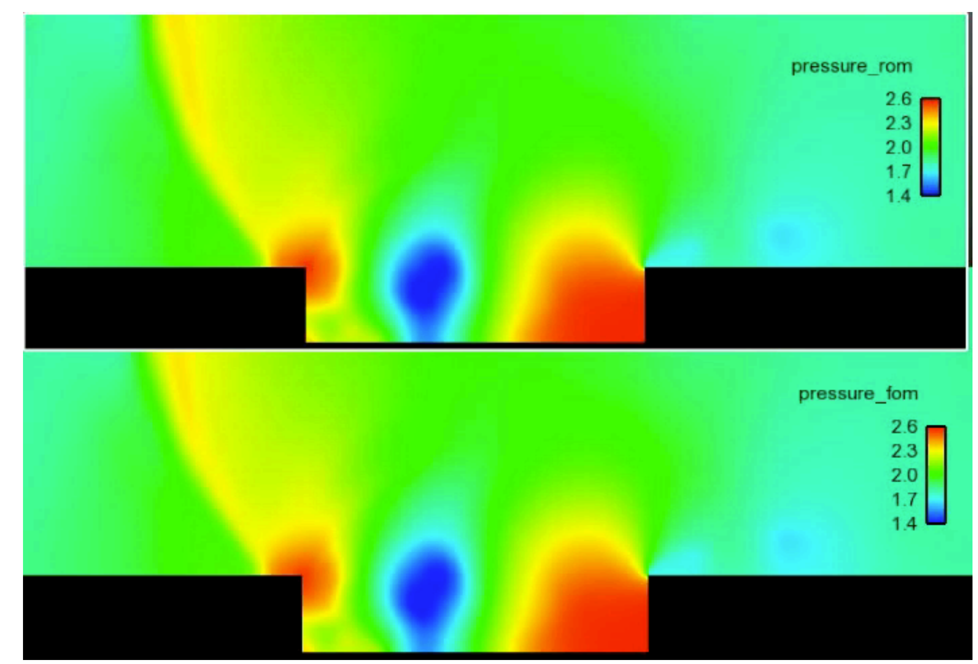
+ *HPC on a laptop*

vorticity field | *pressure field*

LSPG ROM
32 min, 2 cores

high-fidelity
5 hours, 48 cores





+ *229x savings in core–hours*
+ *< 1% error in time-averaged drag*

… so why doesn't everyone use ROMs?

Good performance is not guaranteed

$$\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) = \boldsymbol{\Phi}\,\hat{\mathbf{x}}(t)$$

1) ***Linear-subspace assumption is strong*** ← *This talk*

Reference: *Lee and C. "Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders," arXiv e-Print, 1812.08373 (2018).*

2) ***Accuracy limited by content of*** $\boldsymbol{\Phi}$ ← *Etter, 3:05pm today, this room*

Reference: *Etter and C. "Online adaptive basis refinement and compression for reduced-order models," arXiv e-Print, 1902.10659 (2019).*

Good performance is not guaranteed

$$\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) = \boldsymbol{\Phi}\,\hat{\mathbf{x}}(t)$$

1) ***Linear-subspace assumption is strong*** ← *This talk*

Reference: *Lee and C. "Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders," arXiv e-Print, 1812.08373 (2018).*

2) ***Accuracy limited by content of*** $\boldsymbol{\Phi}$ ← *Etter, 3:05pm today, this room*

Reference: *Etter and C. "Online adaptive basis refinement and compression for reduced-order models," arXiv e-Print, 1902.10659 (2019).*

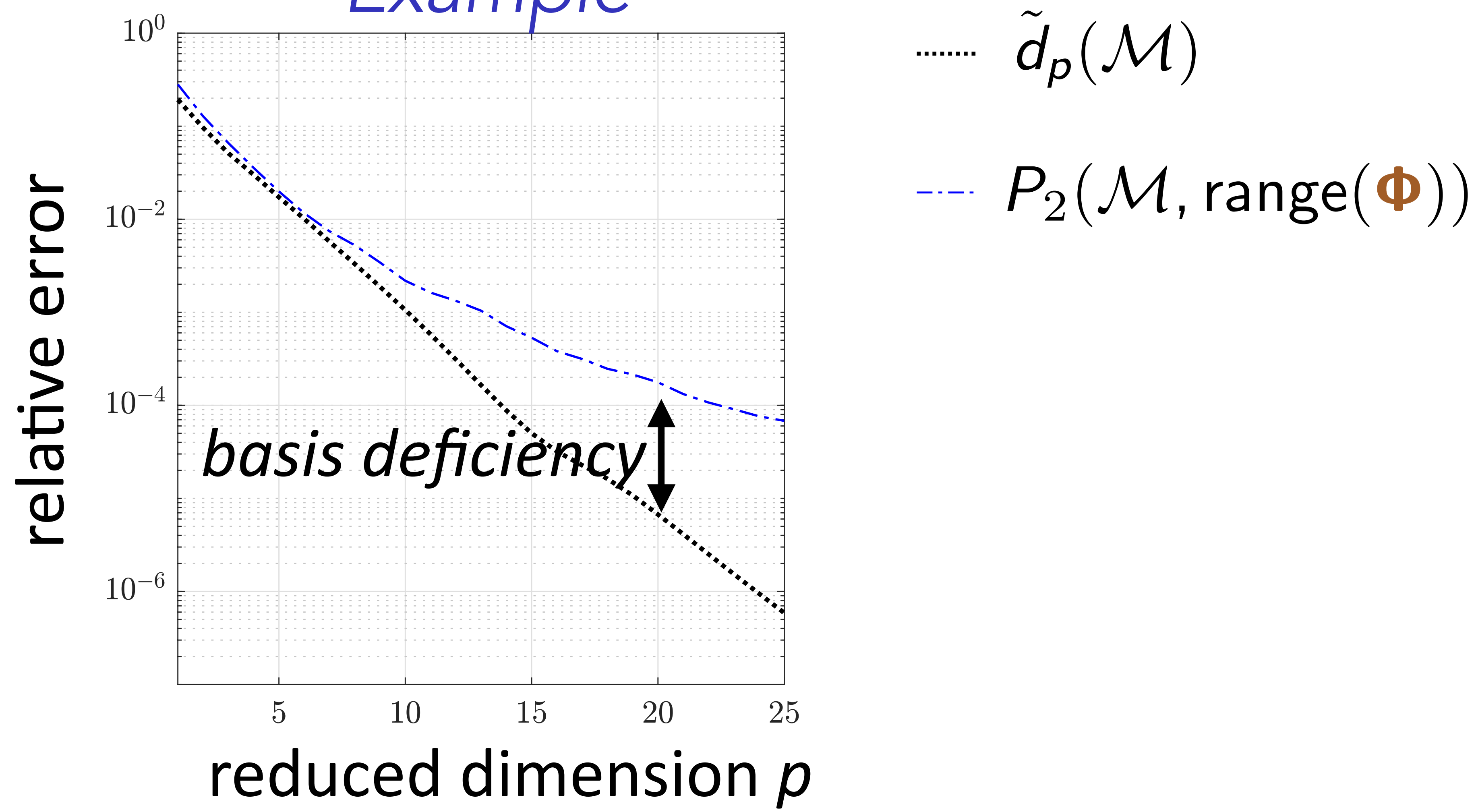
Kolmogorov-width limitation of linear subspaces

- $\mathcal{M} := \{\mathbf{x}(t, \boldsymbol{\mu}) \mid t \in [0, T_{\text{final}}],\ \boldsymbol{\mu} \in \mathcal{D}\}$: solution manifold
- $\mathcal{S}_p$: set of all p-dimensional linear subspaces
- $d_p(\mathcal{M}) := \inf_{\mathcal{S} \in \mathcal{S}_p} P_\infty(\mathcal{M}, \mathcal{S})$, $P_\infty(\mathcal{M}, \mathcal{S}) := \sup_{\mathbf{x} \in \mathcal{M}} \inf_{\mathbf{y} \in \mathcal{S}} \|\mathbf{x} - \mathbf{y}\|$

Kolmogorov-width limitation of linear subspaces

- $\mathcal{M} := \{\mathbf{x}(t, \boldsymbol{\mu}) \mid t \in [0, T_{\text{final}}],\ \boldsymbol{\mu} \in \mathcal{D}\}$: solution manifold
- $\mathcal{S}_p$: set of all p-dimensional linear subspaces
- $\tilde{d}_p(\mathcal{M}) := \inf_{\mathcal{S} \in \mathcal{S}_p} P_2(\mathcal{M}, \mathcal{S})$, $P_2(\mathcal{M}, \mathcal{S}) := \sqrt{\sum_{\mathbf{x} \in \mathcal{M}} \inf_{\mathbf{y} \in \mathcal{S}} \|\mathbf{x} - \mathbf{y}\|^2} / \sqrt{\sum_{\mathbf{x} \in \mathcal{M}} \|\mathbf{x}\|^2}$

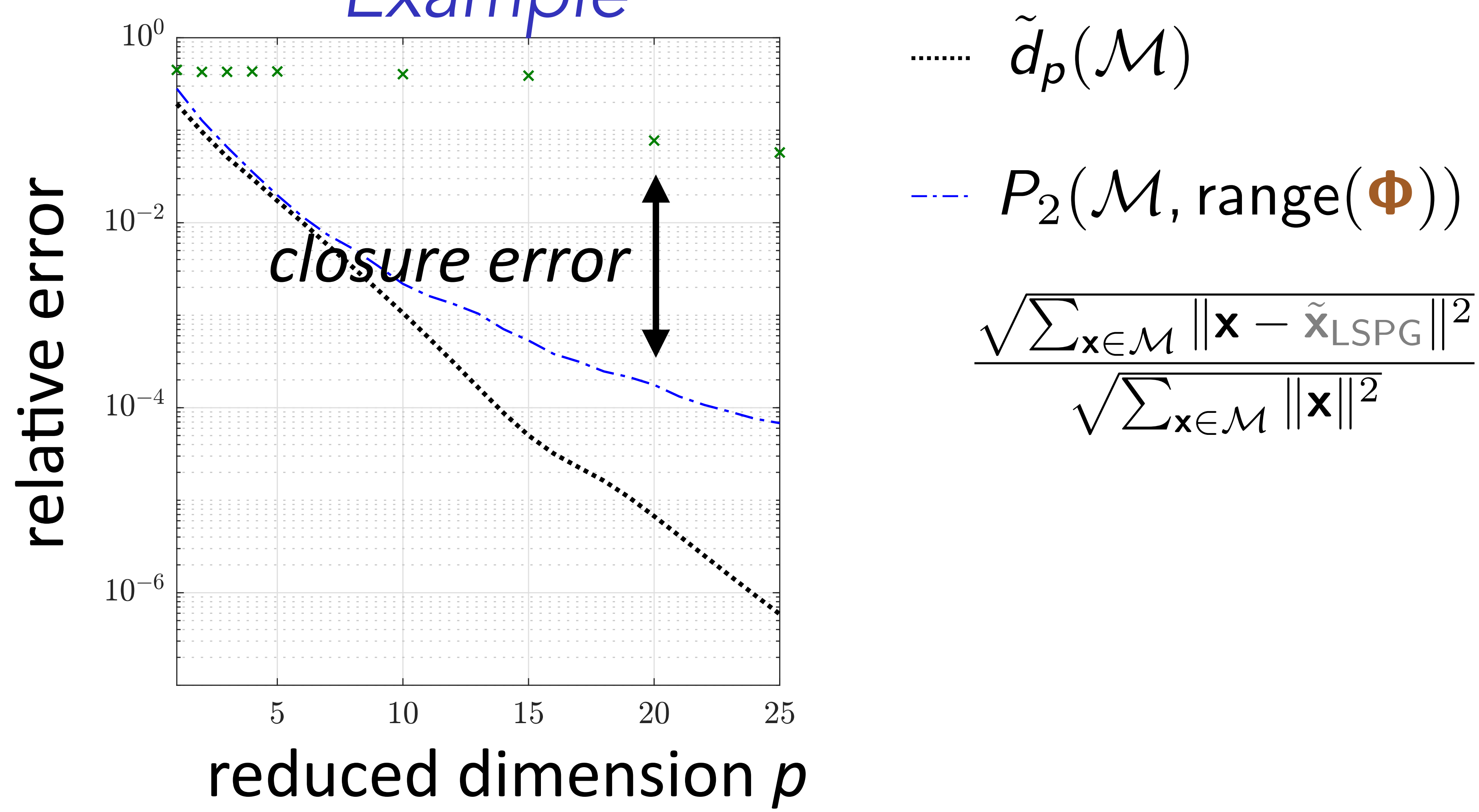
Example



Kolmogorov-width limitation of linear subspaces

- $\mathcal{M} := \{\mathbf{x}(t,\boldsymbol{\mu}) \mid t \in [0, T_{\text{final}}],\ \boldsymbol{\mu} \in \mathcal{D}\}$: solution manifold
- $\mathcal{S}_p$: set of all p-dimensional linear subspaces
- $\tilde{d}_p(\mathcal{M}) := \inf_{\mathcal{S}\in\mathcal{S}_p} P_2(\mathcal{M},\mathcal{S})$, $P_2(\mathcal{M},\mathcal{S}) := \sqrt{\sum_{\mathbf{x}\in\mathcal{M}} \inf_{\mathbf{y}\in\mathcal{S}} \|\mathbf{x}-\mathbf{y}\|^2} \Big/ \sqrt{\sum_{\mathbf{x}\in\mathcal{M}} \|\mathbf{x}\|^2}$

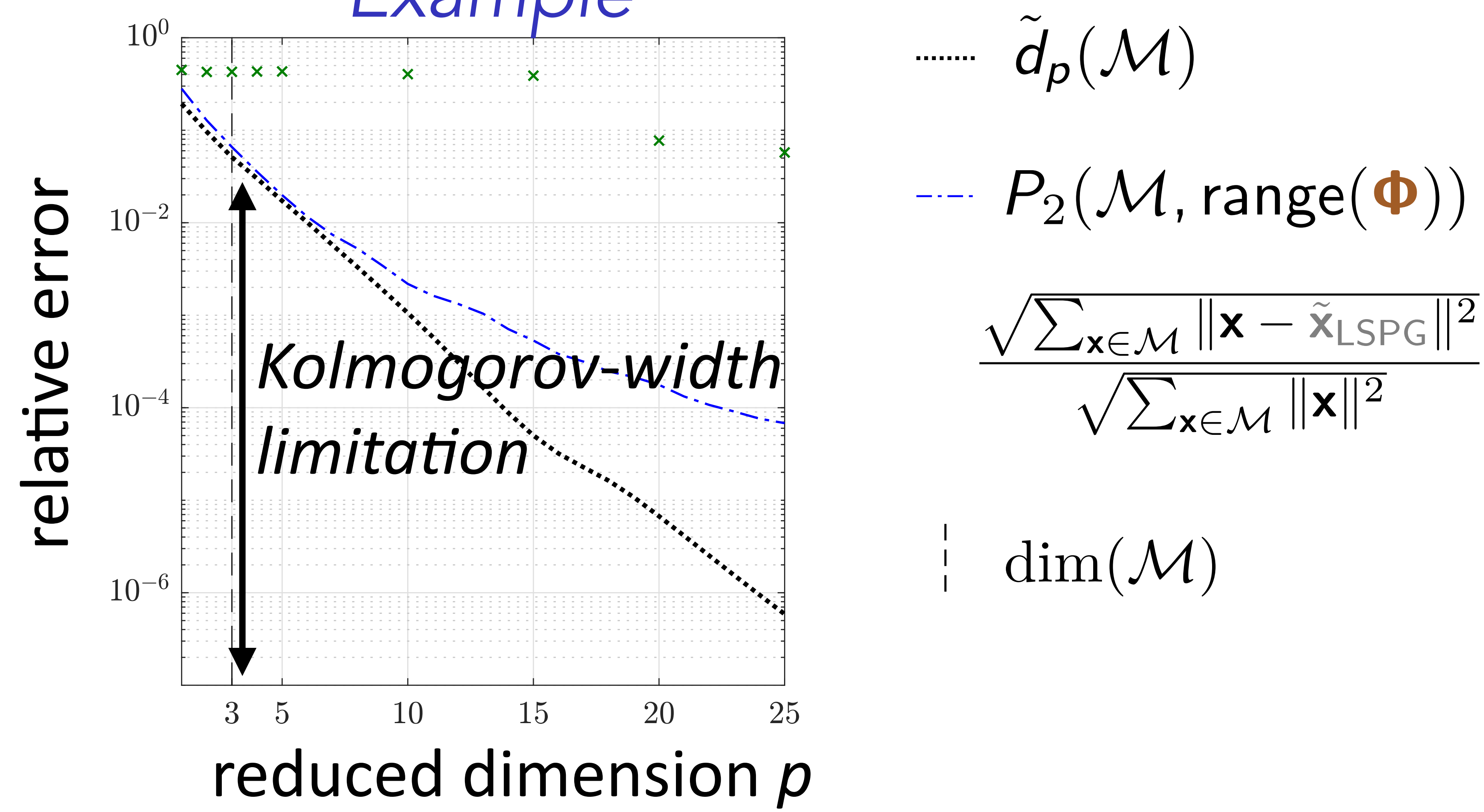
Example



Kolmogorov-width limitation of linear subspaces

- $\mathcal{M} := \{\mathbf{x}(t, \boldsymbol{\mu}) \mid t \in [0, T_{\text{final}}],\ \boldsymbol{\mu} \in \mathcal{D}\}$: solution manifold
- $\mathcal{S}_p$: set of all p-dimensional linear subspaces
- $\tilde{d}_p(\mathcal{M}) := \inf_{\mathcal{S} \in \mathcal{S}_p} P_2(\mathcal{M}, \mathcal{S})$, $P_2(\mathcal{M}, \mathcal{S}) := \sqrt{\sum_{\mathbf{x} \in \mathcal{M}} \inf_{\mathbf{y} \in \mathcal{S}} \|\mathbf{x} - \mathbf{y}\|^2} \Big/ \sqrt{\sum_{\mathbf{x} \in \mathcal{M}} \|\mathbf{x}\|^2}$

Example



- Kolmogorov-width limitation: significant error for $p = \dim(\mathcal{M})$

***Goal**: overcome limitation via projection onto a nonlinear manifold*

Overcoming Kolmogorov-width limitation

Transform/update the linear subspace

[Ohlberger and Rave, 2013; Iollo and Lombardi, 2014; Gerbeau and Lombardi, 2014; Peherstorfer and Willcox, 2015; Welper, 2017; Mojgani and Balajewicz, 2017; Reiss et al., 2018; Zimmermann et al., 2018; Peherstorfer, 2018; Rim and Mandli, 2018; Rim and Mandli, 2018; Nair and Balajewicz, 2019; Cagniart et al., 2019]

\+ Can work much better than a fixed basis

\- Some require problem-specific knowledge or characteristics

\- Do not consider manifolds of general nonlinear structure

Overcoming Kolmogorov-width limitation

Transform/update the linear subspace

[Ohlberger and Rave, 2013; Iollo and Lombardi, 2014; Gerbeau and Lombardi, 2014; Peherstorfer and Willcox, 2015; Welper, 2017; Mojgani and Balajewicz, 2017; Reiss et al., 2018; Zimmermann et al., 2018; Peherstorfer, 2018; Rim and Mandli, 2018; Rim and Mandli, 2018; Nair and Balajewicz, 2019; Cagniart et al., 2019]

\+ Can work much better than a fixed basis

\- Some require problem-specific knowledge or characteristics

\- Do not consider manifolds of general nonlinear structure

***A priori* construction of local linear subspaces**

[Dihlmann et al., 2011; Drohmann et al., 2011; Amsallem, Zahr, Farhat, 2012; Peherstorfer et el., 2014; Taddei et al., 2015]

\+ Tailored bases for Voronoi diagrams of time/spatial domain, state space

\- Do not consider manifolds of general nonlinear structure

Overcoming Kolmogorov-width limitation

Transform/update the linear subspace

[Ohlberger and Rave, 2013; Iollo and Lombardi, 2014; Gerbeau and Lombardi, 2014; Peherstorfer and Willcox, 2015; Welper, 2017; Mojgani and Balajewicz, 2017; Reiss et al., 2018; Zimmermann et al., 2018; Peherstorfer, 2018; Rim and Mandli, 2018; Rim and Mandli, 2018; Nair and Balajewicz, 2019; Cagniart et al., 2019]

+ Can work much better than a fixed basis
- Some require problem-specific knowledge or characteristics
- Do not consider manifolds of general nonlinear structure

A priori **construction of local linear subspaces**

[Dihlmann et al., 2011; Drohmann et al., 2011; Amsallem, Zahr, Farhat, 2012; Peherstorfer et el., 2014; Taddei et al., 2015]

+ Tailored bases for Voronoi diagrams of time/spatial domain, state space
- Do not consider manifolds of general nonlinear structure

Model reduction on nonlinear manifolds [Gu, 2011; Kashima, 2016; Hartman and Mestha, 2017]

- Kinematically inconsistent [Kashima, 2016; Hartman and Mestha, 2017]
- Limited to piecewise linear manifolds [Gu, 2011]
- Solutions lack optimality [Gu, 2011; Kashima, 2016; Hartman and Mestha, 2017]

Goals

Overcome shortcomings of existing methods

+ Enable manifolds with general nonlinear structure
+ Kinematically consistent
+ Satisfy optimality property

Manifold Galerkin and LSPG projection

Practical nonlinear-manifold construction

+ No problem-specific knowledge or characteristics required
+ Use same snapshot data as POD

Deep convolutional autoencoders

Goals

Overcome shortcomings of existing methods

+ Enable manifolds with general nonlinear structure
+ Kinematically consistent
+ Satisfy optimality property

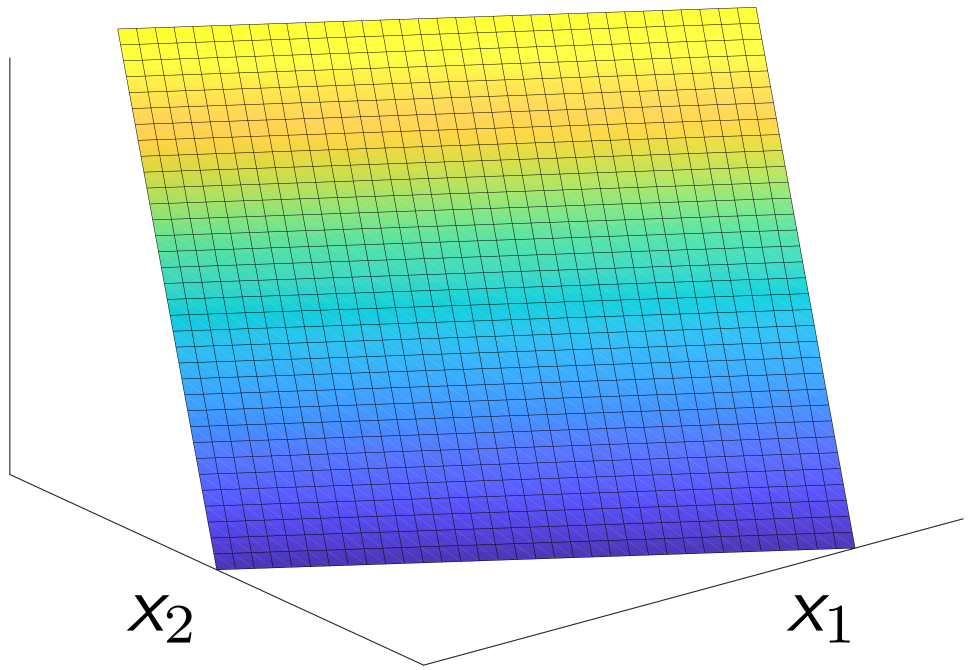
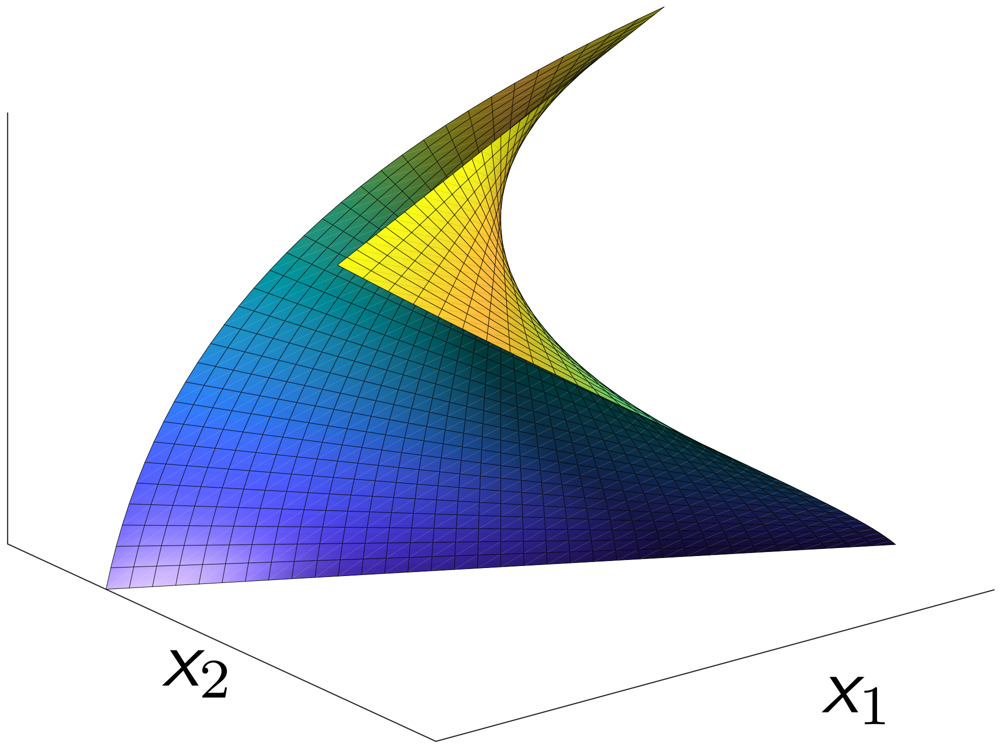
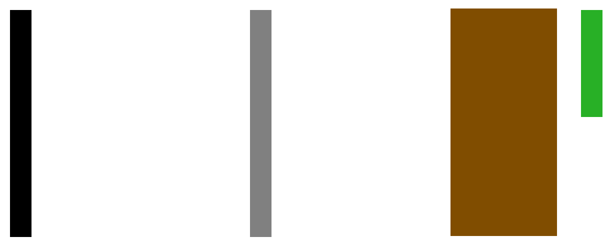
Manifold Galerkin and LSPG projection

Practical nonlinear-manifold construction

+ No problem-specific knowledge required
+ Use same snapshot data as POD

Deep convolutional autoencoders

Nonlinear trial manifold

	Linear trial subspace	***Nonlinear trial manifold***
	$\mathrm{range}(\boldsymbol{\Phi}) := \{\boldsymbol{\Phi}\hat{\mathbf{x}} \mid \hat{\mathbf{x}} \in \mathbb{R}^p\}$	$\mathcal{S} := \{\mathbf{g}(\hat{\mathbf{x}}) \mid \hat{\mathbf{x}} \in \mathbb{R}^p\}$
example *N=3* *p=2*		
state	$\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) = \boldsymbol{\Phi}\,\hat{\mathbf{x}}(t) \in \mathrm{range}(\boldsymbol{\Phi})$	$\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) = \mathbf{g}(\hat{\mathbf{x}}(t)) \in \mathcal{S}$
		
		+ manifold has general structure
velocity	$\frac{d\mathbf{x}}{dt} \approx \frac{d\tilde{\mathbf{x}}}{dt} = \boldsymbol{\Phi}\frac{d\hat{\mathbf{x}}}{dt} \in \mathrm{range}(\boldsymbol{\Phi})$	$\frac{d\mathbf{x}}{dt} \approx \frac{d\tilde{\mathbf{x}}}{dt} = \nabla\mathbf{g}(\hat{\mathbf{x}})\frac{d\hat{\mathbf{x}}}{dt} \in T_{\hat{\mathbf{x}}}\mathcal{S}$
		+ kinematically consistent

Manifold Galerkin and LSPG projection

	Linear-subspace ROM	***Nonlinear-manifold ROM***
Galerkin	$\frac{d\hat{\mathbf{x}}}{dt} = \underset{\hat{\mathbf{v}}\in\mathbb{R}^n}{\mathrm{argmin}} \Vert \mathbf{r}(\boldsymbol{\Phi}\hat{\mathbf{v}}, \boldsymbol{\Phi}\hat{\mathbf{x}}; t)\Vert_2$	$\frac{d\hat{\mathbf{x}}}{dt} = \underset{\hat{\mathbf{v}}\in\mathbb{R}^n}{\mathrm{argmin}} \Vert \mathbf{r}(\nabla\mathbf{g}(\hat{\mathbf{x}})\hat{\mathbf{v}}, \mathbf{g}(\hat{\mathbf{x}}); t)\Vert_2$
	$\Updownarrow$	$\Updownarrow$
	$\boldsymbol{\Phi}\frac{d\hat{\mathbf{x}}}{dt} = \underset{\hat{\mathbf{v}}\in\mathrm{range}(\boldsymbol{\Phi})}{\mathrm{argmin}} \Vert \hat{\mathbf{v}} - \mathbf{f}(\boldsymbol{\Phi}\hat{\mathbf{x}}; t)\Vert_2$	$\nabla\mathbf{g}(\hat{\mathbf{x}})\frac{d\hat{\mathbf{x}}}{dt} = \underset{\hat{\mathbf{v}}\in T_{\hat{\mathbf{x}}}\mathcal{S}}{\mathrm{argmin}} \Vert \hat{\mathbf{v}} - \mathbf{f}(\mathbf{g}(\hat{\mathbf{x}}); t)\Vert_2$
	$\Updownarrow$	$\Updownarrow$
	$\frac{d\hat{\mathbf{x}}}{dt} = \boldsymbol{\Phi}^T\mathbf{f}(\boldsymbol{\Phi}\hat{\mathbf{x}}; t)$	$\frac{d\hat{\mathbf{x}}}{dt} = \nabla\mathbf{g}(\hat{\mathbf{x}})^+\mathbf{f}(\mathbf{g}(\hat{\mathbf{x}}); t)$
LSPG	$\hat{\mathbf{x}}^n = \underset{\hat{\mathbf{v}}\in\mathbb{R}^p}{\mathrm{argmin}} \Vert \mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{v}})\Vert_2$	$\hat{\mathbf{x}}^n = \underset{\hat{\mathbf{v}}\in\mathbb{R}^p}{\mathrm{argmin}} \Vert \mathbf{r}^n(\mathbf{g}(\hat{\mathbf{v}}))\Vert_2$

\+ Satisfy residual-minimization properties

Error bound

Theorem

If the following conditions hold:

1. $\mathbf{f}(\cdot; t)$ is Lipschitz continuous with Lipschitz constant κ
2. Δt is small enough such that $0 < h := |\alpha_0| - |\beta_0|\kappa\Delta t$, then

$$\|\mathbf{x}^n - \mathbf{g}(\hat{\mathbf{x}}_{\mathrm{G}}^n)\|_2 \leq \frac{1}{h}\|\mathbf{r}_{\mathrm{G}}^n(\mathbf{g}(\hat{\mathbf{x}}_{\mathrm{G}}))\|_2 + \frac{1}{h}\sum_{\ell=1}^{k}|\gamma_\ell|\|\mathbf{x}^{n-\ell} - \mathbf{g}(\hat{\mathbf{x}}_{\mathrm{G}})\|_2$$

$$\|\mathbf{x}^n - \mathbf{g}(\hat{\mathbf{x}}_{\mathrm{LSPG}}^n)\|_2 \leq \frac{1}{h}\min_{\hat{\mathbf{v}}}\|\mathbf{r}_{\mathrm{LSPG}}^n(\mathbf{g}(\hat{\mathbf{v}}))\|_2 + \frac{1}{h}\sum_{\ell=1}^{k}|\gamma_\ell|\|\mathbf{x}^{n-\ell} - \mathbf{g}(\hat{\mathbf{x}}_{\mathrm{LSPG}})\|_2$$

+ Manifold LSPG sequentially minimizes the error bound

Equivalence

Proposition

Linear-subspace and nonlinear-manifold LSPG projection are equivalent if

- the trial manifold is affine, i.e., $\mathbf{g} : \hat{\mathbf{x}} \mapsto \mathbf{A}\hat{\mathbf{x}} + \mathbf{b}$

Linear-subspace and nonlinear-manifold Galerkin projection are equivalent if

- the trial manifold is affine, i.e., $\mathbf{g} : \hat{\mathbf{x}} \mapsto \mathbf{A}\hat{\mathbf{x}} + \mathbf{b}$, and
- the Jacobian matrix $\mathbf{A}$ is orthogonal.

Theorem

Manifold Galerkin and manifold LSPG are equivalent if

1. the nonlinear trial manifold $\mathcal{S}$ is twice continuously differentiable,
2. $\|\hat{\mathbf{x}}^{n-j} - \hat{\mathbf{x}}^{n}\| = O(\Delta t)$ for $n = 1, \ldots, T$ and $j = 1, \ldots, k$, and
3. the limit $\Delta t \to 0$ is taken.

Goals

Overcome shortcomings of existing methods

+ Enable manifolds with general nonlinear structure
+ Kinematically consistent
+ Satisfy optimality property

Manifold Galerkin and LSPG projection

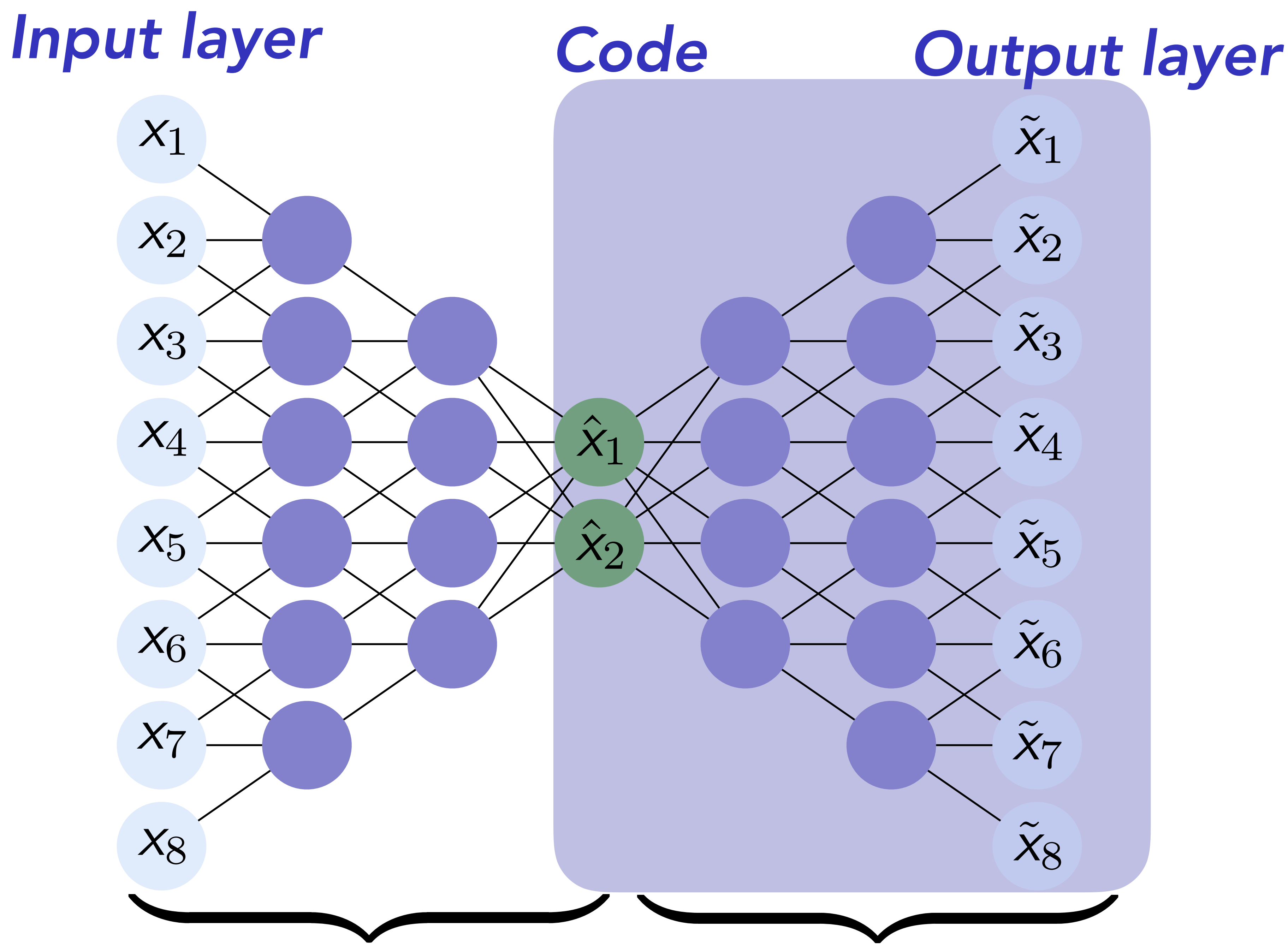
Practical nonlinear-manifold construction

+ No problem-specific knowledge required
+ Use same snapshot data as POD

Deep convolutional autoencoders

$$\mathcal{S} := \{\mathbf{g}(\hat{\mathbf{x}}) \,|\, \hat{\mathbf{x}} \in \mathbb{R}^p\}$$

Deep autoencoders



Encoder $\mathbf{h}_{\text{enc}}(\cdot;\boldsymbol{\theta}_{\text{enc}})$ ***Decoder*** $\mathbf{h}_{\text{dec}}(\cdot;\boldsymbol{\theta}_{\text{dec}})$

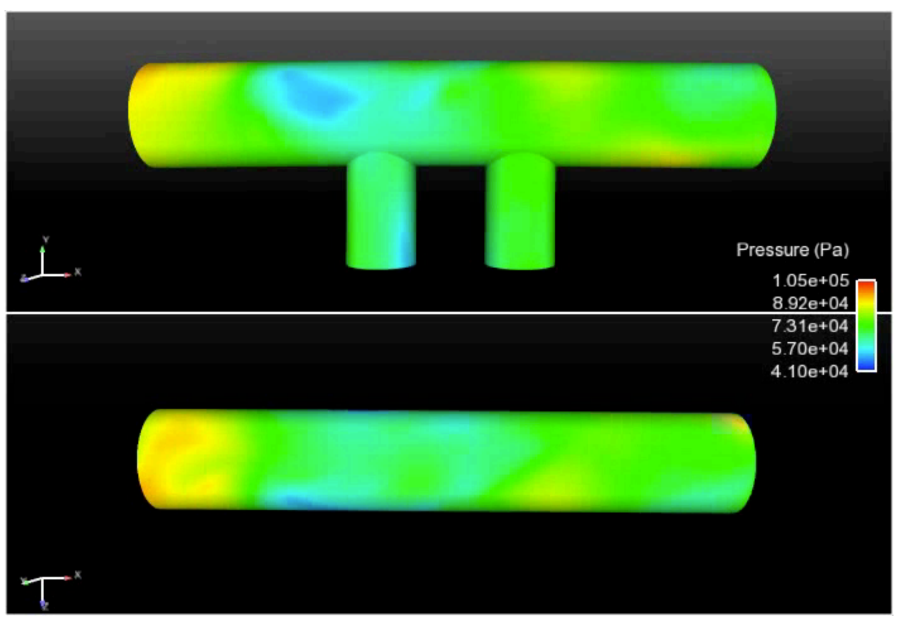
$$\tilde{\mathbf{x}} = \mathbf{h}_{\text{dec}}(\cdot;\boldsymbol{\theta}_{\text{dec}}) \circ \mathbf{h}_{\text{enc}}(\mathbf{x};\boldsymbol{\theta}_{\text{enc}})$$

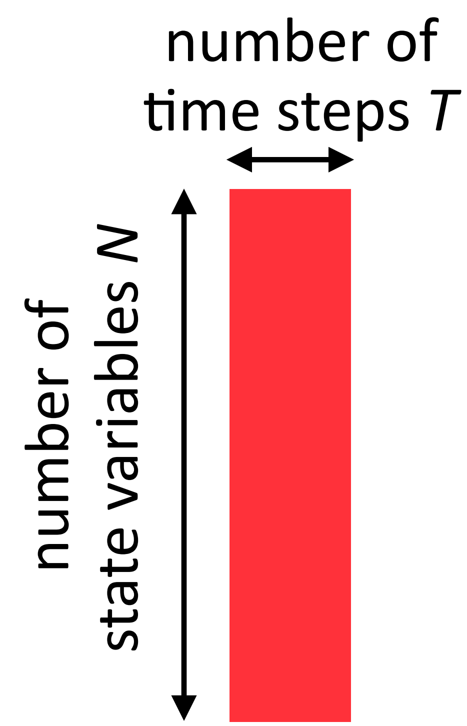
+ If $\tilde{\mathbf{x}} \approx \mathbf{x}$ for parameters $\boldsymbol{\theta}^{\star}_{\text{dec}}$, $\mathbf{g} = \mathbf{h}_{\text{dec}}(\cdot;\boldsymbol{\theta}^{\star}_{\text{dec}})$ produces an accurate manifold

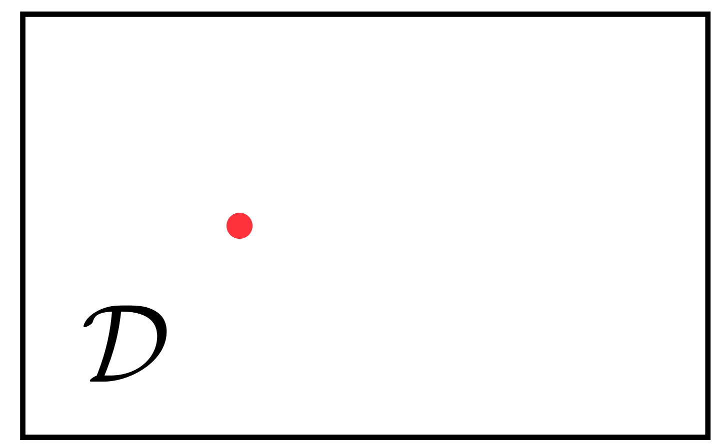
Training algorithm

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify structure in data



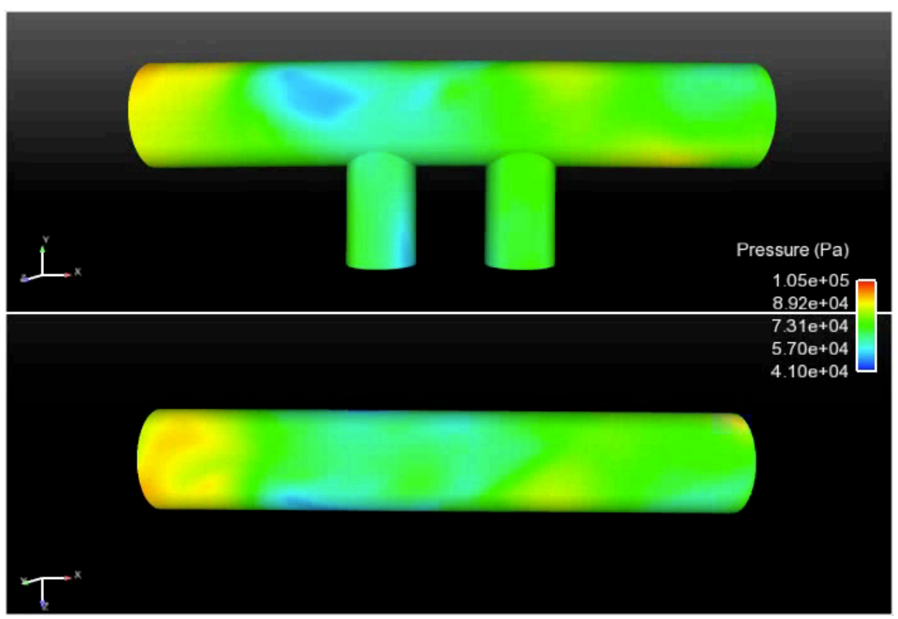


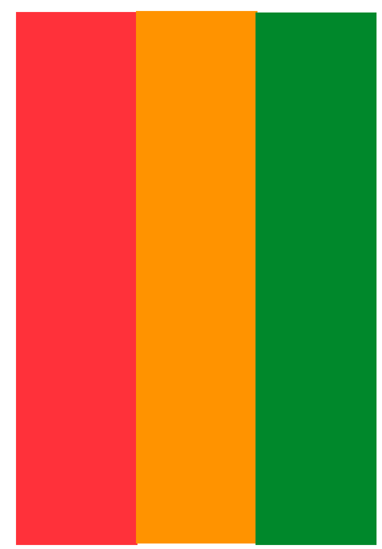


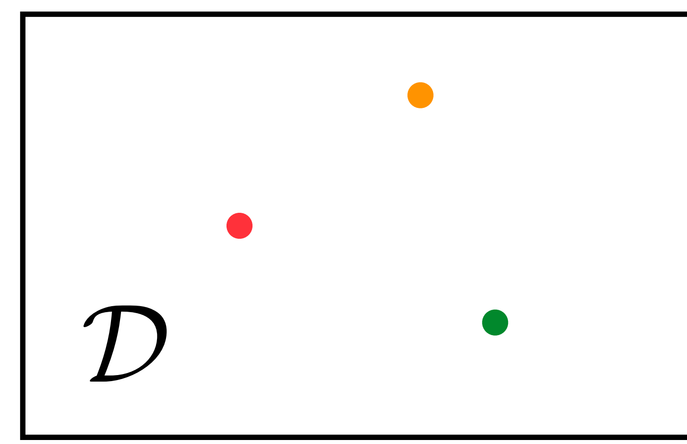
Training algorithm

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify structure in data



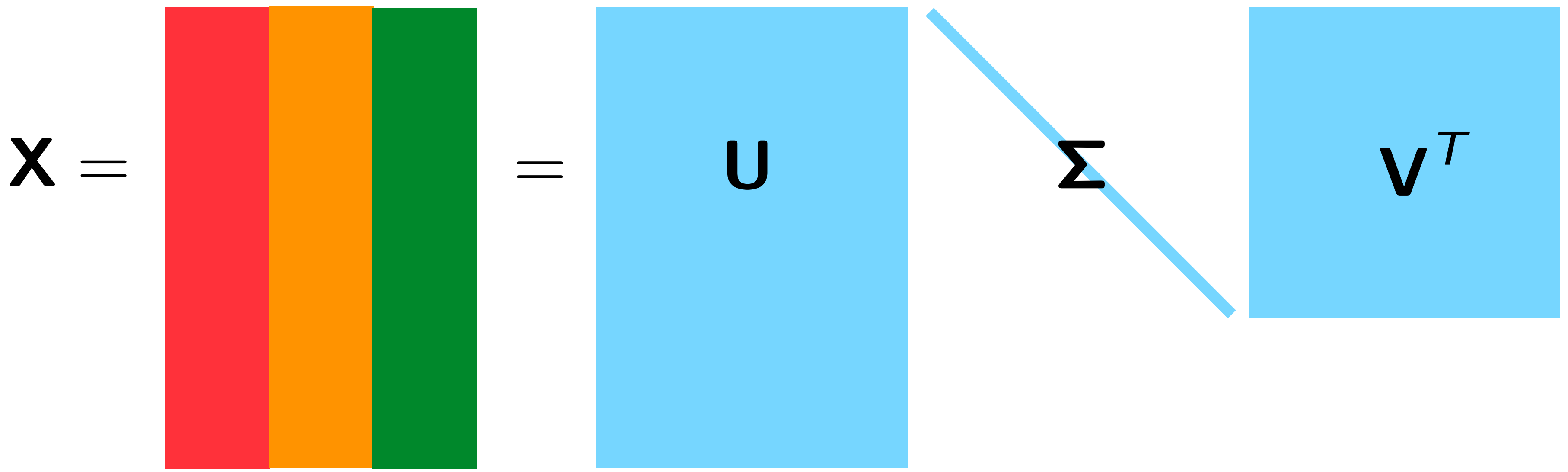
$\mathbf{X} =$ 



Training algorithm: POD

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

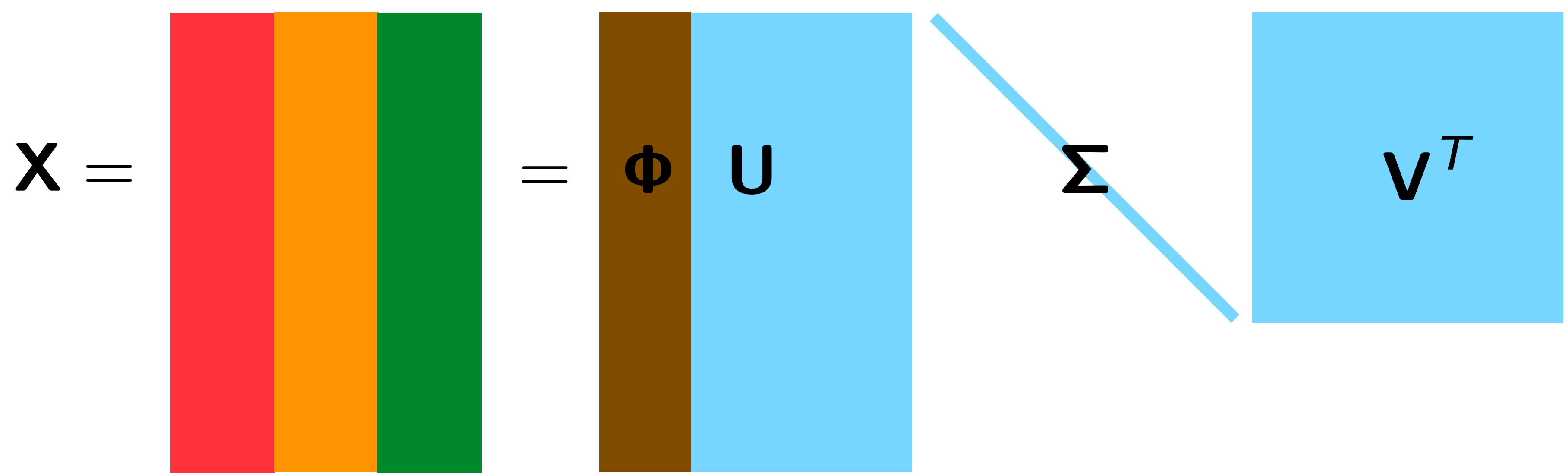
1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify structure in data



Training algorithm: POD

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify structure in data

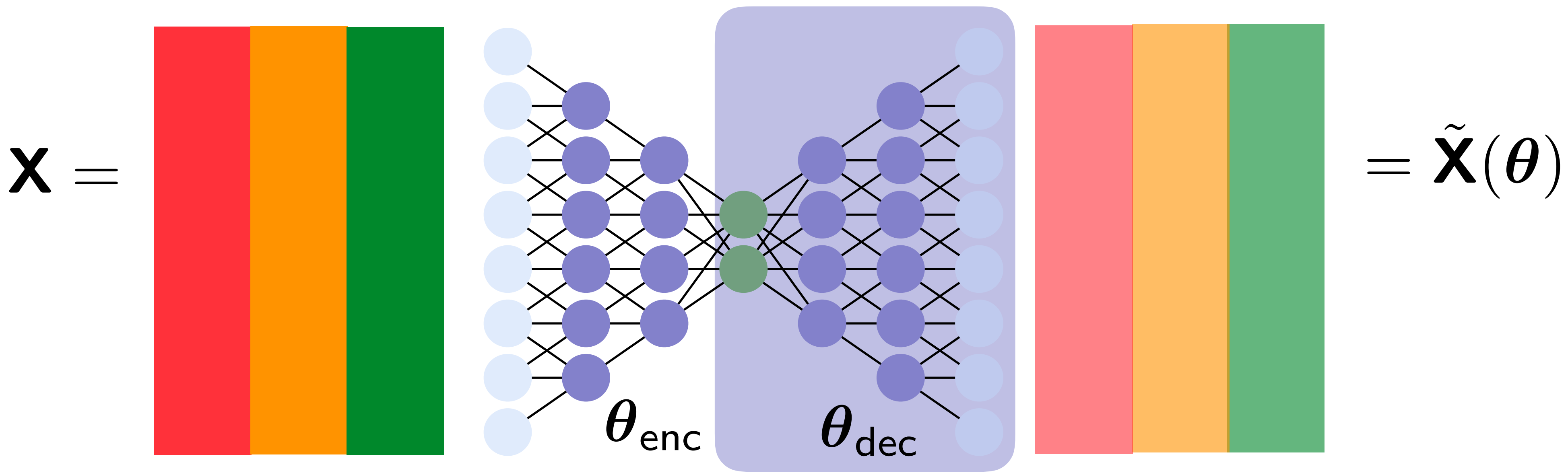


- $\boldsymbol{\Phi}$ Satisfies $\underset{\bar{\boldsymbol{\Phi}} \in \mathbb{R}^{N \times p},\, \bar{\boldsymbol{\Phi}}^T \bar{\boldsymbol{\Phi}} = \mathbf{I}}{\text{minimize}} \|\mathbf{X} - \bar{\boldsymbol{\Phi}} \bar{\boldsymbol{\Phi}}^T \mathbf{X}\|_F$

Training algorithm: autoencoder

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify structure in data



- Compute $\boldsymbol{\theta}^\star$ by approximately solving $\underset{\boldsymbol{\theta}}{\text{minimize}} \|\mathbf{X} - \tilde{\mathbf{X}}(\boldsymbol{\theta})\|_F$
- Define nonlinear trial manifold by setting $\mathbf{g} = \mathbf{h}_{\text{dec}}(\cdot; \boldsymbol{\theta}^\star_{\text{dec}})$

\+ Same snapshot data, no specialized problem knowledge

Numerical results

1D Burgers' equation

$$\frac{\partial w(x,t;\boldsymbol{\mu})}{\partial t} + \frac{\partial f(w(x,t;\boldsymbol{\mu}))}{\partial x} = 0.02e^{\alpha x}$$

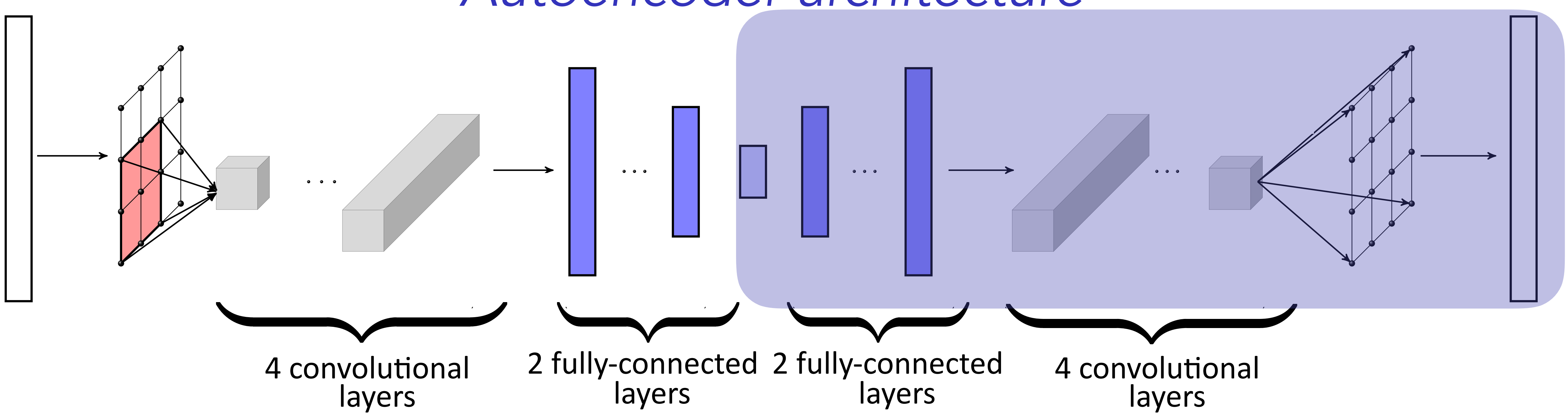
- $\boldsymbol{\mu}$: α, inlet boundary condition
- *Spatial discretization*: finite volume
- *Time integrator:* backward Euler

2D reacting flow

$$\frac{\partial \mathbf{w}(\vec{x},t;\boldsymbol{\mu})}{\partial t} = \nabla\cdot(\kappa\nabla\mathbf{w}(\vec{x},t;\boldsymbol{\mu})) - \mathbf{v}\cdot\nabla\mathbf{w}(\vec{x},t;\boldsymbol{\mu}) + \mathbf{q}(\mathbf{w}(\vec{x},t;\boldsymbol{\mu});\boldsymbol{\mu})$$

- $\boldsymbol{\mu}$: two terms in reaction
- *Spatial discretization*: finite difference
- *Time integrator*: BDF2

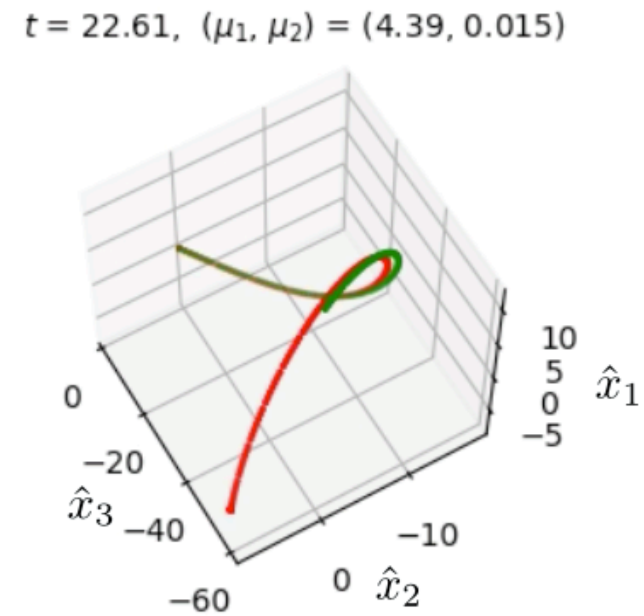
Autoencoder architecture



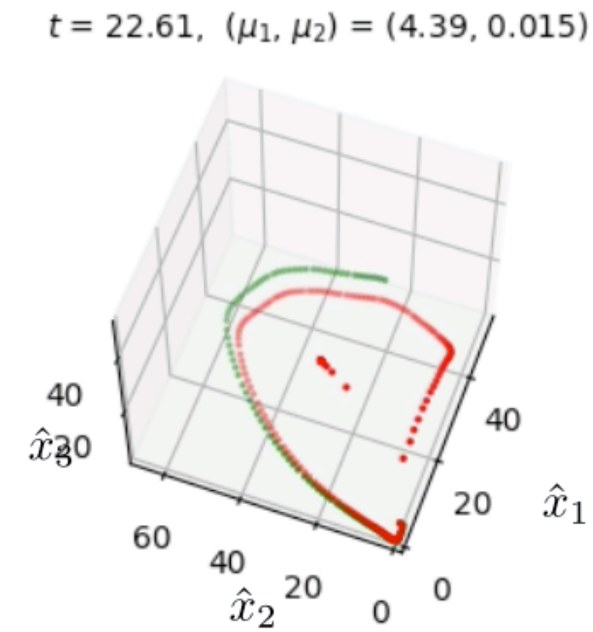
Manifold interpretation: Burgers' equation

FOM

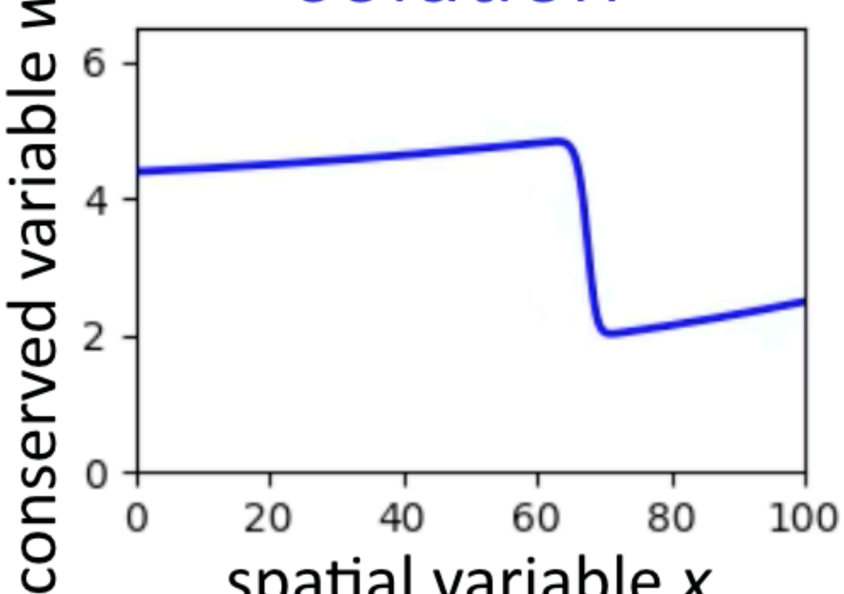
POD, p=3
projection



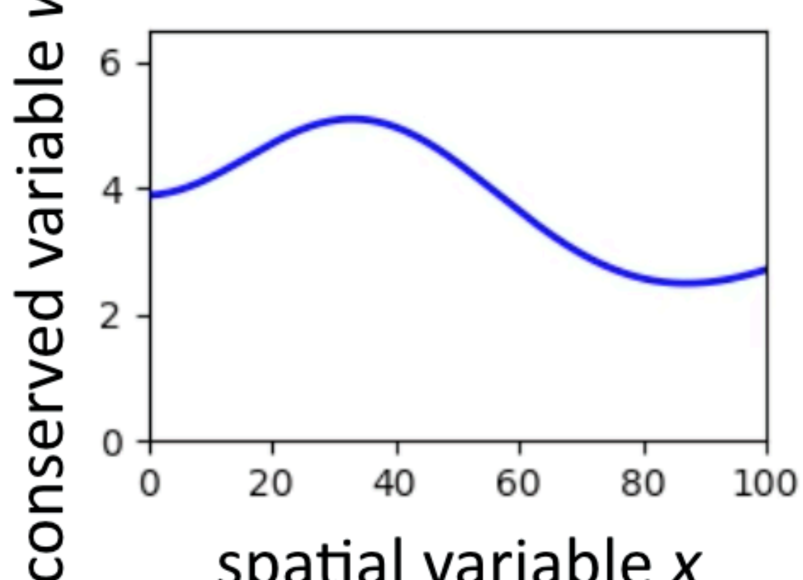
Autoencoder, p=3
projection



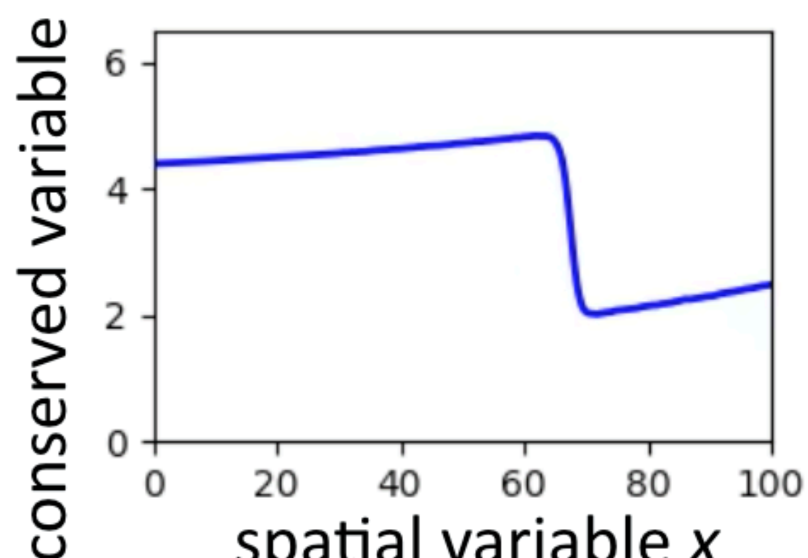
solution



reconstruction



reconstruction



+ *Projection error onto 3-dimensional manifold nearly perfect*

Manifold LSPG outperforms optimal linear subspace

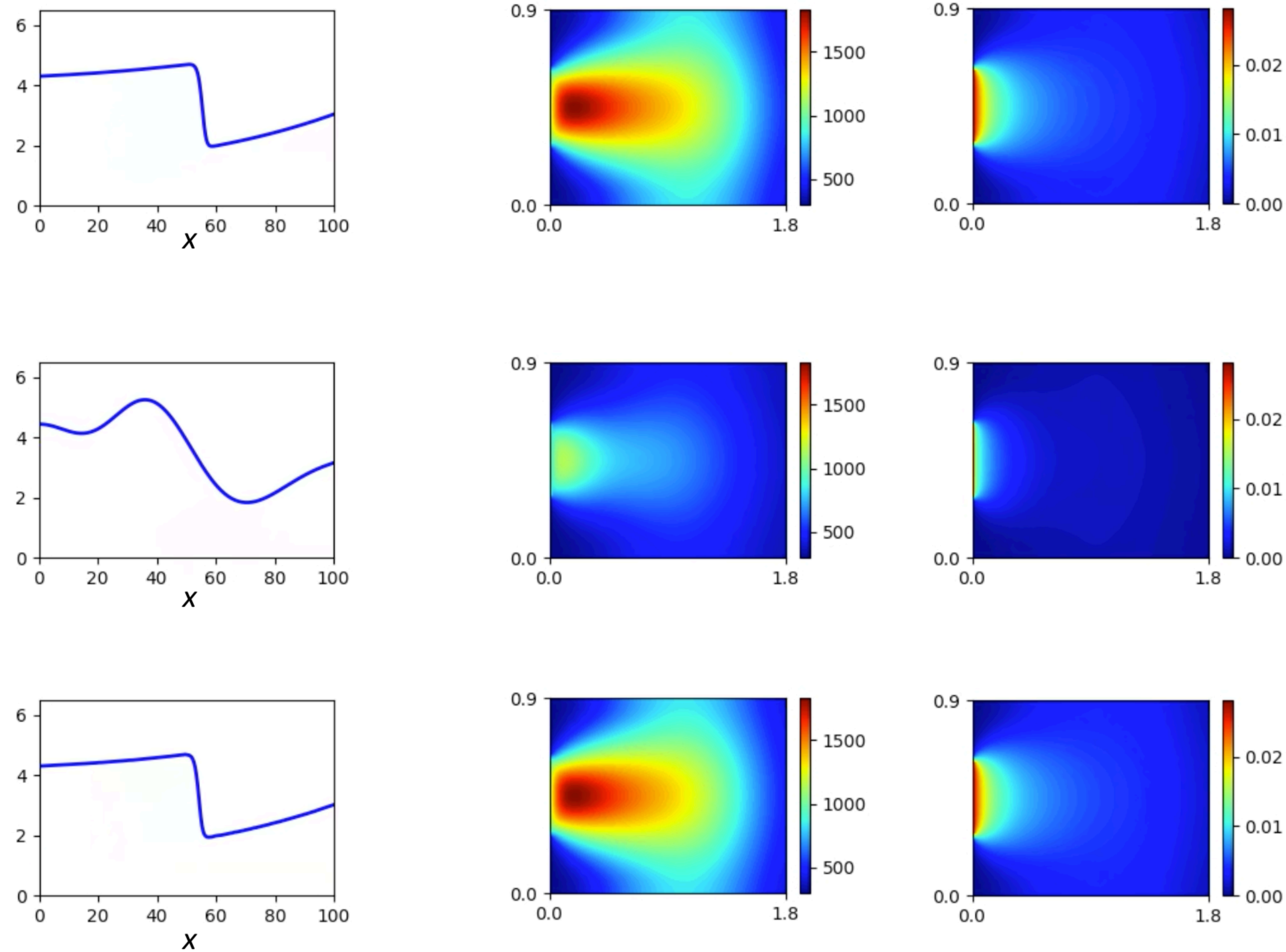
1D Burgers' equation — conserved variable

2D reacting flow — temperature, *H_2 fraction*

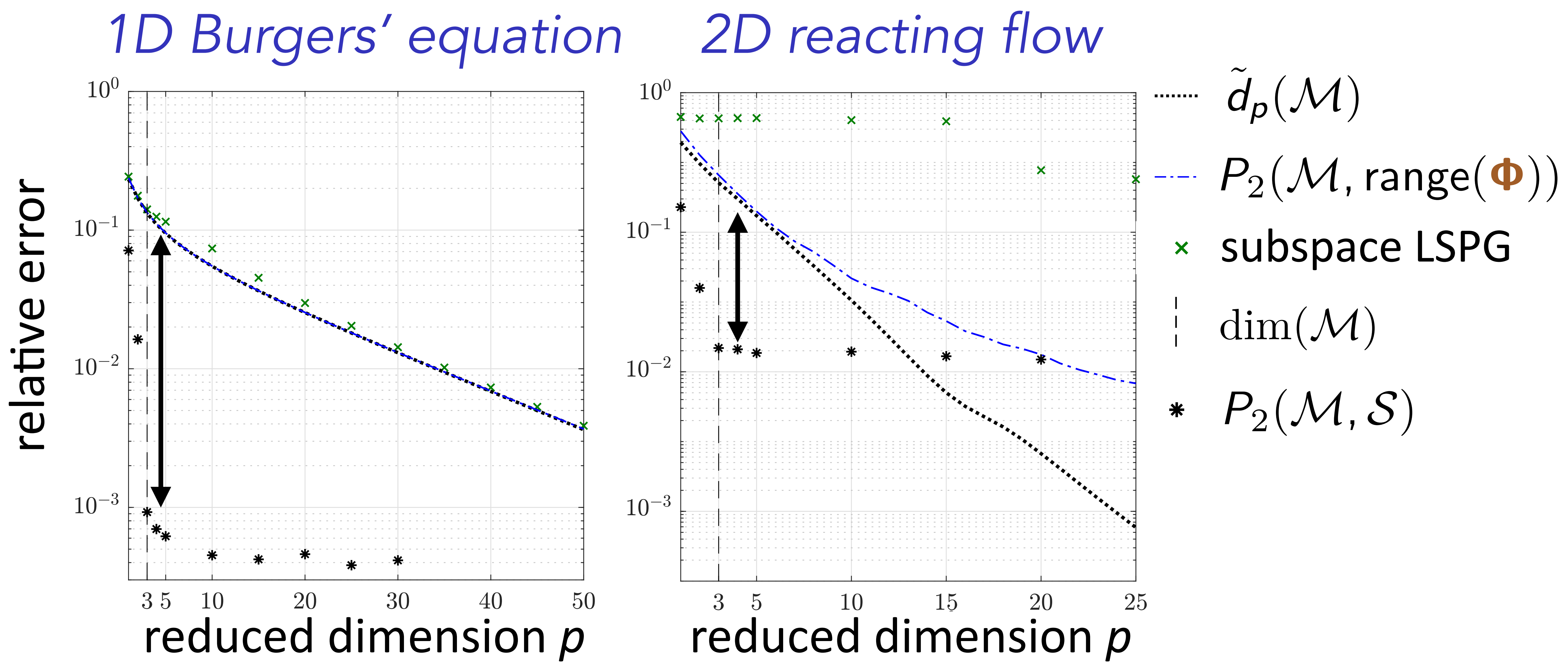
high-fidelity model

POD-LSPG p=5

Manifold LSPG p=5

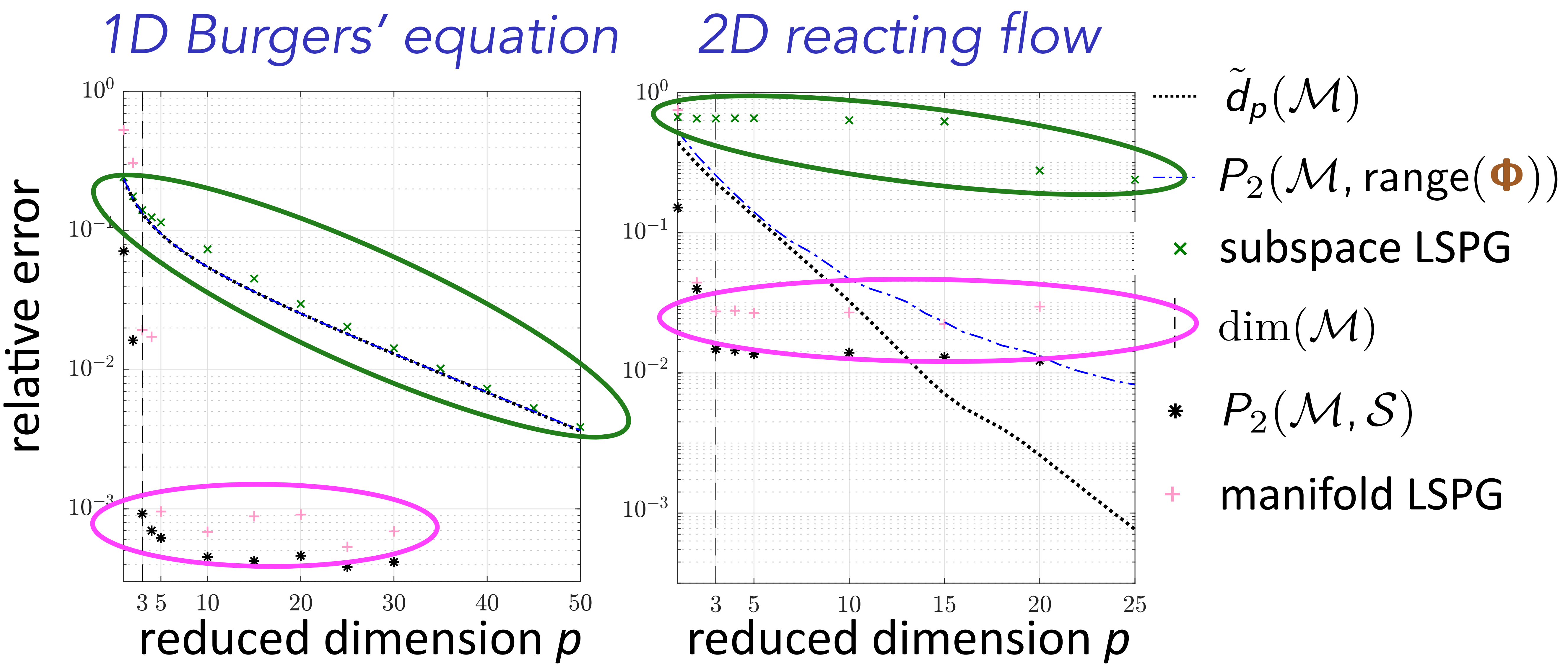


Method overcomes Kolmogorov-width limitation



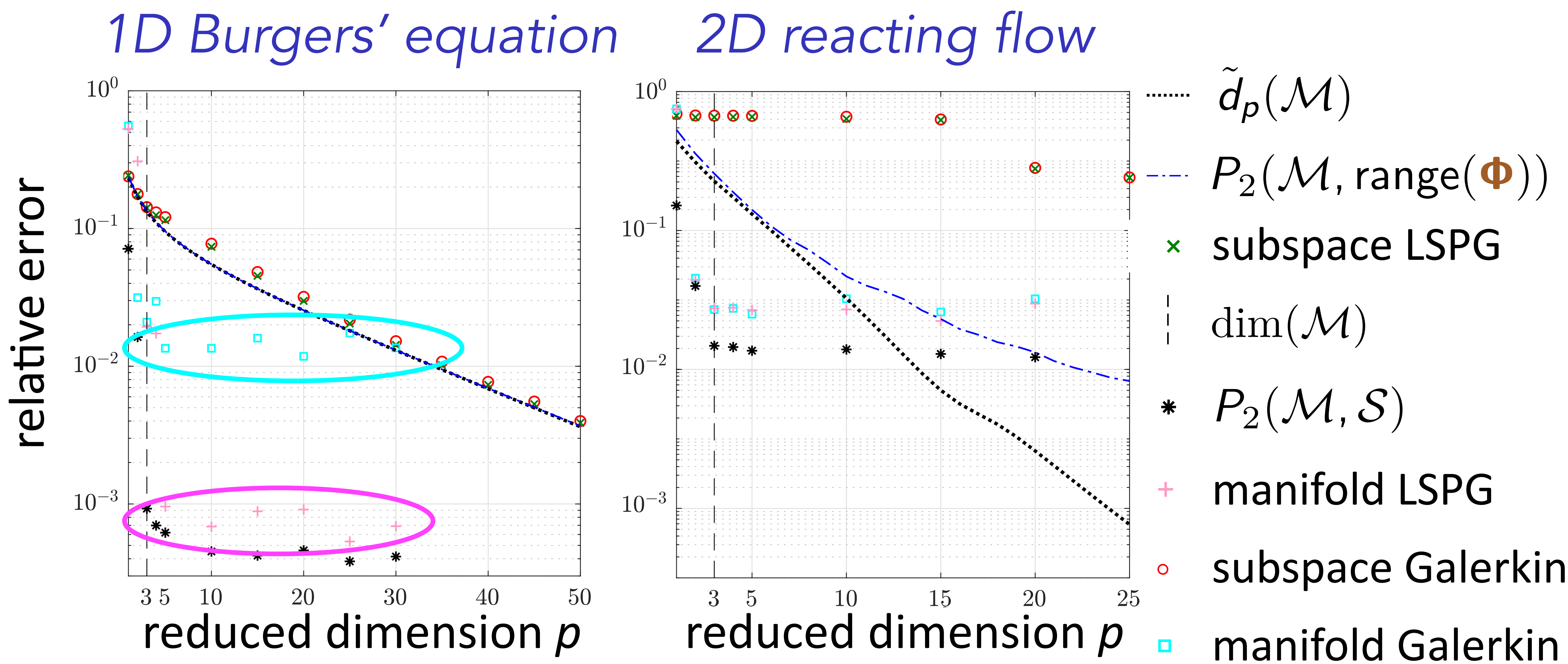
+ Autoencoder manifold significantly better than optimal linear subspace

Method overcomes Kolmogorov-width limitation



+ Autoencoder manifold significantly better than optimal linear subspace
+ Manifold LSPG orders-of-magnitude more accurate than subspace LSPG

Method overcomes Kolmogorov-width limitation



+ Autoencoder manifold significantly better than optimal linear subspace
+ Manifold LSPG orders-of-magnitude more accurate than subspace LSPG
+ Method shatters Kolmogorov-width limitation
+ Manifold LSPG outperforms manifold Galerkin on 1D Burgers' equation

Outlook

Manifold Galerkin

$$\frac{d\hat{\mathbf{x}}}{dt} = \underset{\hat{\mathbf{v}} \in \mathbb{R}^n}{\operatorname{argmin}} \|\mathbf{r}(\nabla \mathbf{g}(\hat{\mathbf{x}})\hat{\mathbf{v}}, \mathbf{g}(\hat{\mathbf{x}}); t)\|_2$$

Manifold LSPG

$$\hat{\mathbf{x}}^n = \underset{\hat{\mathbf{v}} \in \mathbb{R}^p}{\operatorname{argmin}} \|\mathbf{r}^n(\mathbf{g}(\hat{\mathbf{v}}))\|_2$$

Interpretation

- First work demonstrating *physics-constrained* time evolution of codes

Gradient computation

- Backpropagation used to compute decoder Jacobian $\nabla \mathbf{g}(\hat{\mathbf{x}})$
- Quasi-Newton solvers directly call TensorFlow

Forward-compatible extensions

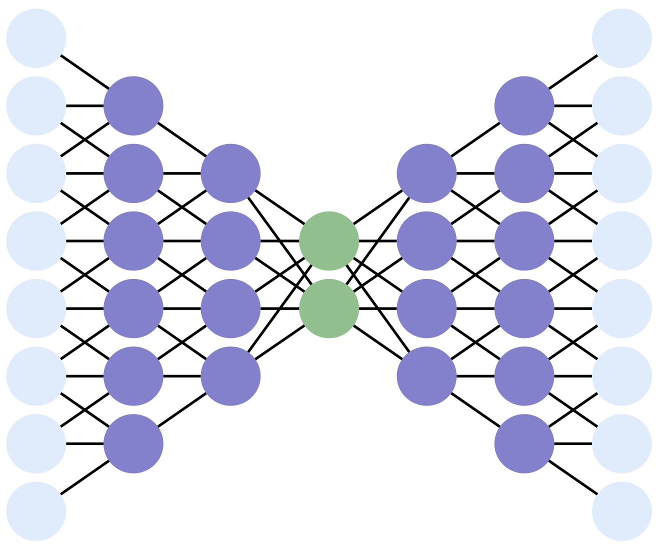
- *Hyper-reduction*: convolutional layers preserve sparsity
- *Structure preservation*: equality constraints enforcing conservation

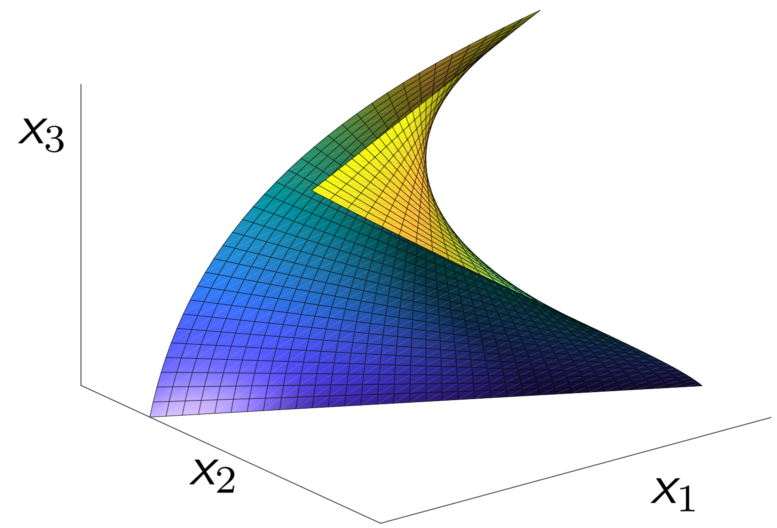
Future work

- Detailed study of architecture, amount of requisite training
- Integration in large-scale code

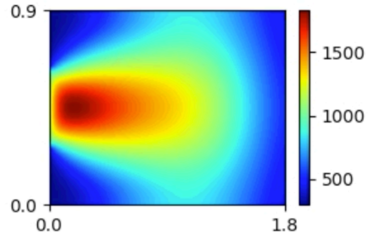
Questions?

Reference: *Lee and C. "Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders," arXiv e-Print, 1812.08373 (2018).*

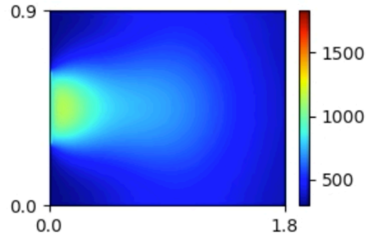




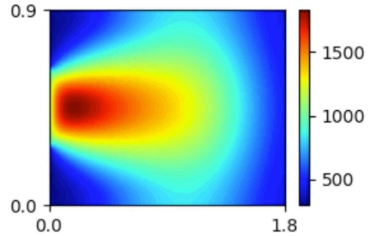
high-fidelity model



POD-LSPG p=5



Manifold LSPG p=5



Our group in Livermore, California has staff and postdoc openings (email me: ktcarlb@sandia.gov)

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. Lawrence Livermore National Laboratory is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under Contract DE-AC52-07NA27344.