Adaptive Reduced Order Models for Chaotic Multi-Scale Problems

Cheng Huang and Ryan Camacho

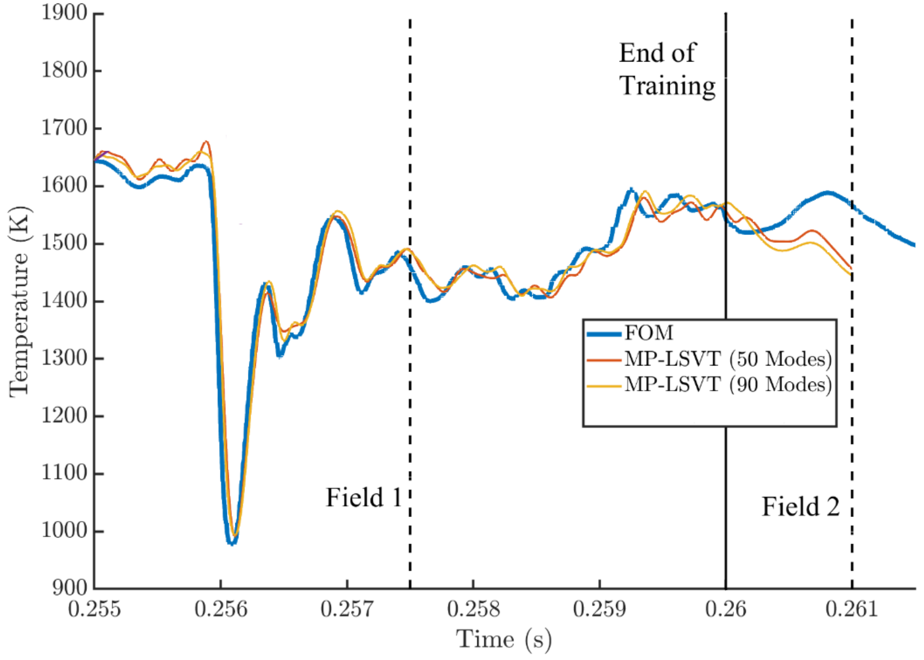
University of Kansas

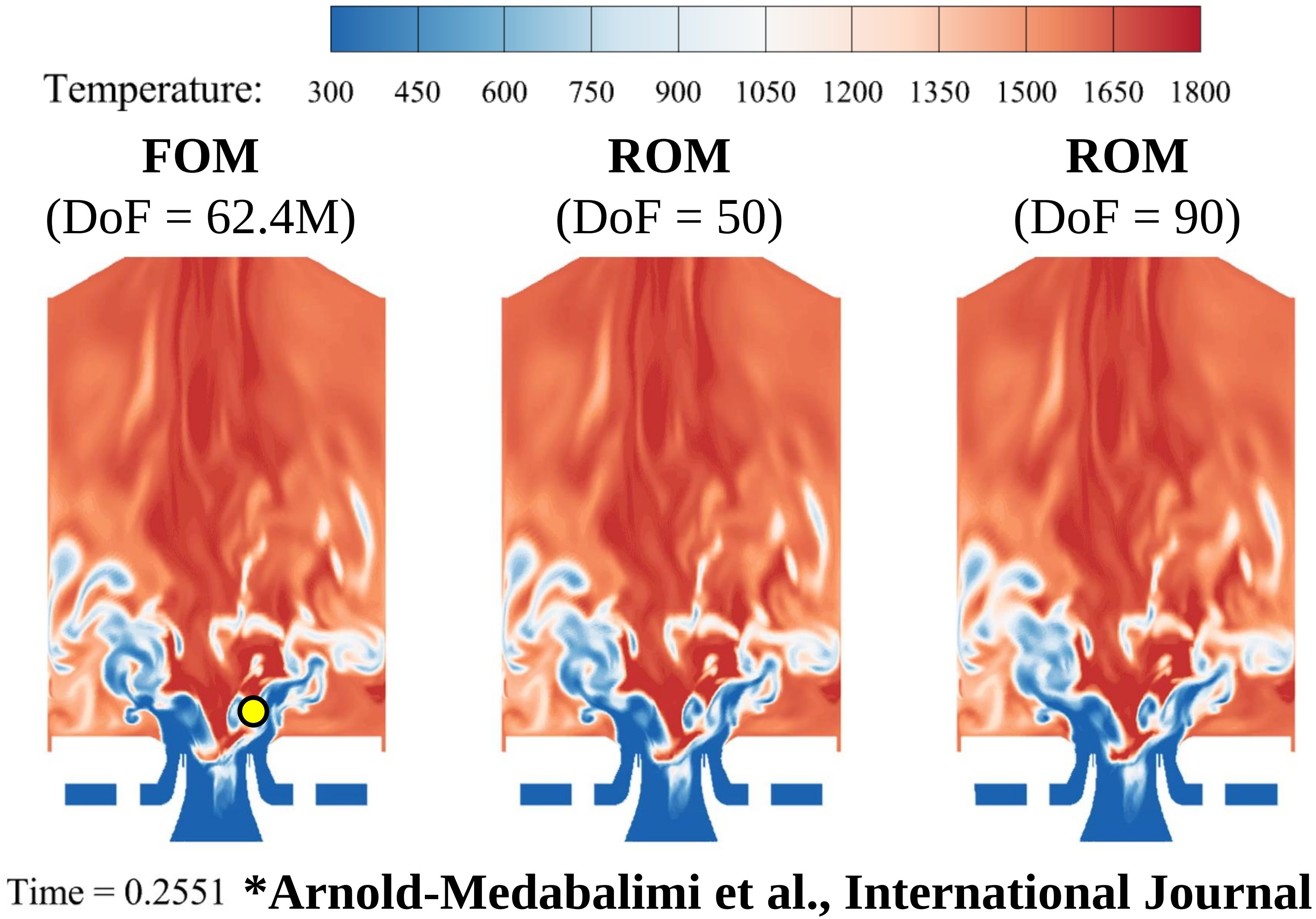
Karthik Duraisamy

University of Michigan

AF/COE Workshop
Thursday, August 31
Dayton, OH

ROM with linear subspace can *accurately* reproduce but cannot *accurately* predict chaotic multi-scale problems

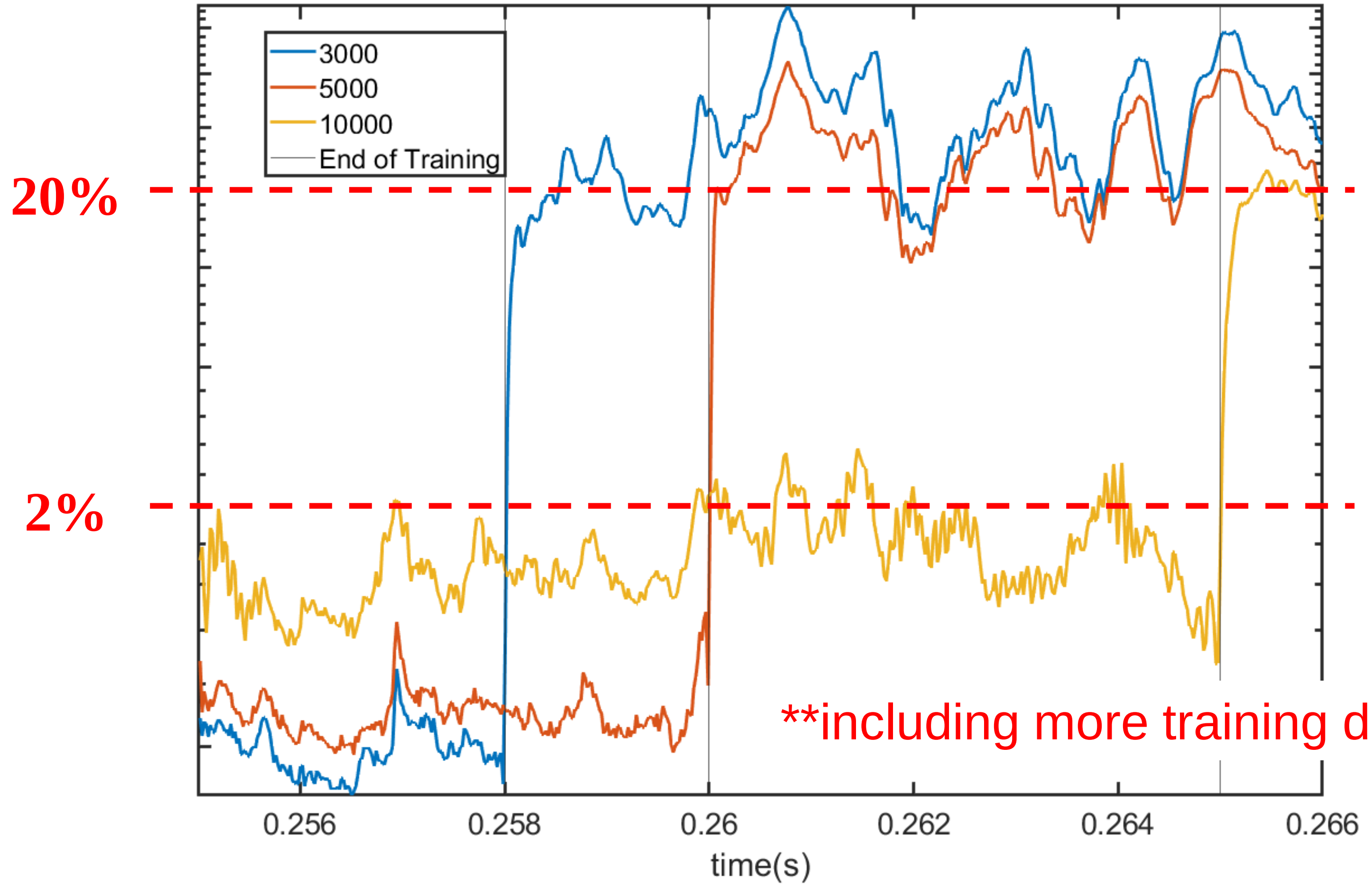




***Arnold-Medabalimi et al., International Journal of Spray and Combustion Dynamics, 2022**

ROM performance is limited by the Projected FOM

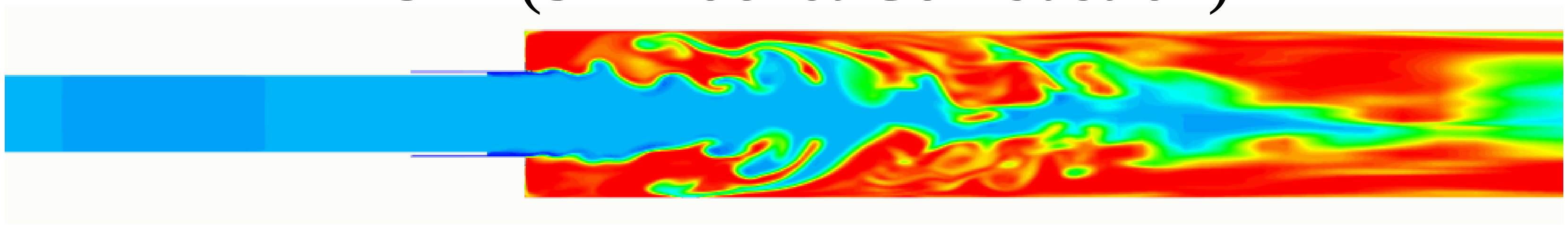
$$\text{Projection Error}(t) = \frac{\left\| \mathbf{q}_{FOM} - \tilde{\mathbf{q}}_{FOM} \right\|_2}{\left\| \mathbf{q}_{FOM} \right\|_2} \quad \text{where} \quad \boxed{\tilde{\mathbf{q}}_{FOM} = \mathbf{q}_{ref} + \mathbf{V}\mathbf{V}^T \mathbf{q}_{FOM}}$$

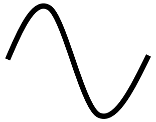


**including more training data does not seem to help

What is enough training data to predict QoI accurately?

FOM (3D Rocket Combustion)

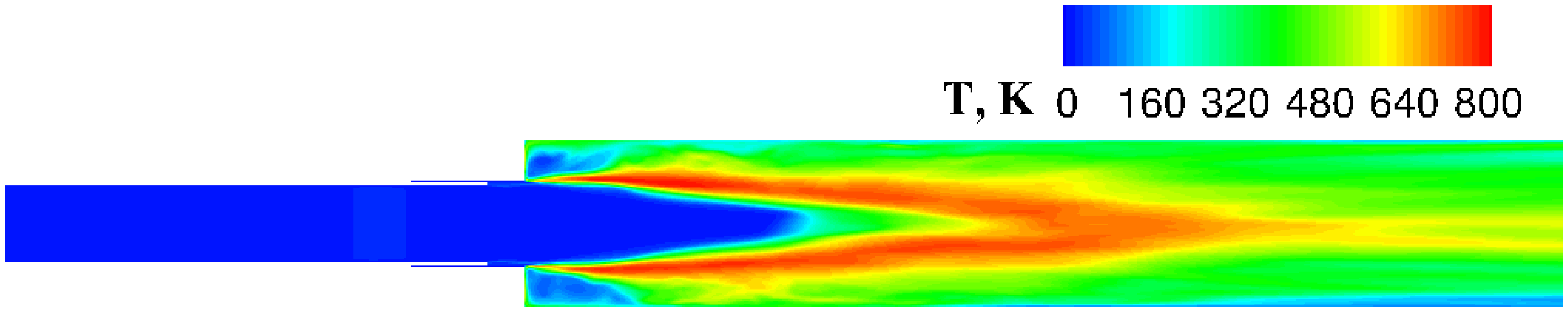




2000Hz @ 10%

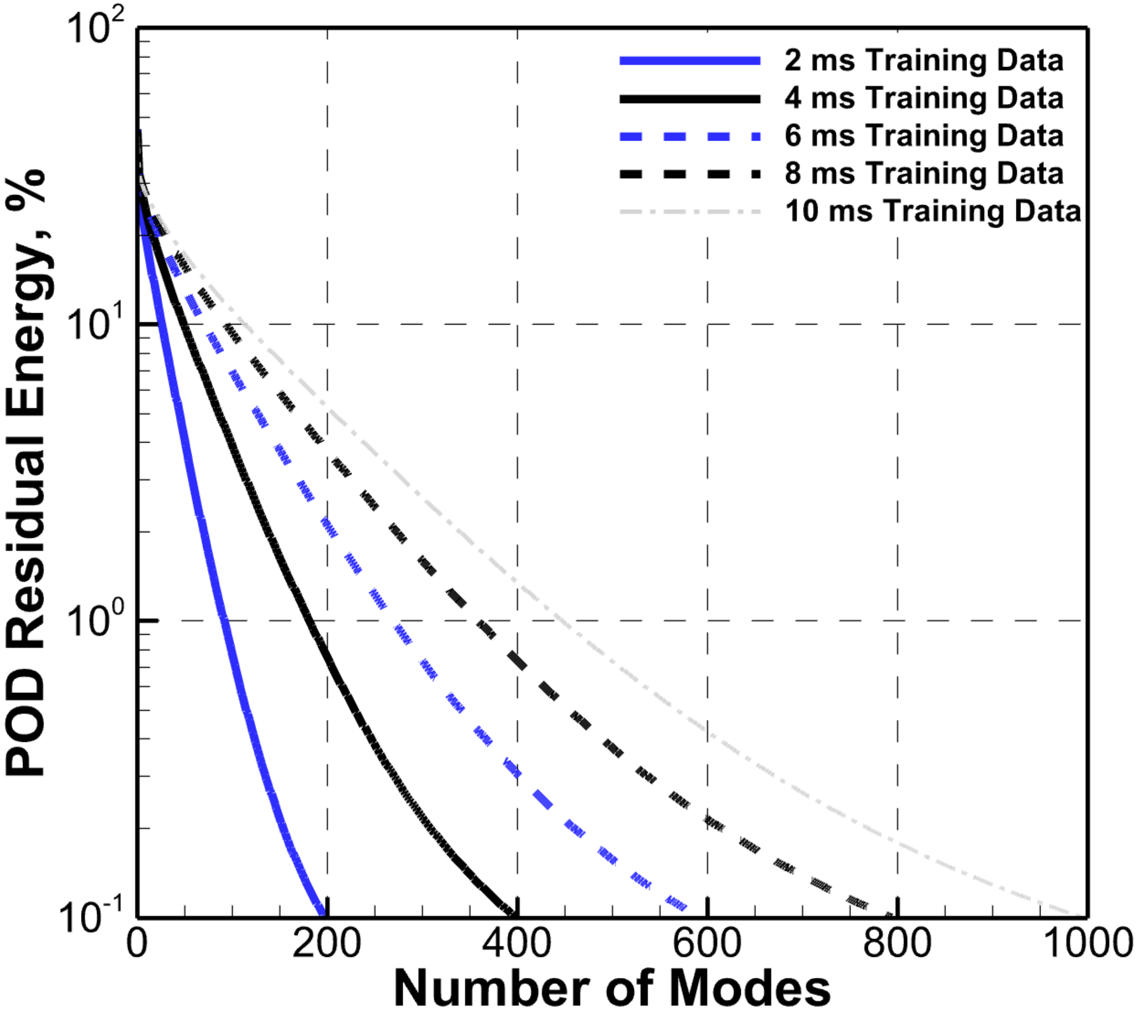
QoI – Temperature RMS

calculated based on **10ms** unsteady solutions



POD Energy Decay *vs* Training Data Amount

$$\text{Residual Energy,\%} = \left(1 - \sum_{n=1}^{K} \sigma_n^2 \Big/ \sum_{n=1}^{N_p} \sigma_n^2\right) \times 100\%$$



Slow decays of Kolmogorov N-width with linear subspace approximation

- Significant number of basis modes required to recover accurate representations of the target physics

Non-convergence of low-rank approx. versus training data amount

- Predictions of *unseen* physics requires significant amount of training data

a priori Analysis based on Projected FOM ($q_{ref} + VV^Tq_{FOM}$)

- challenging to use *linear static* basis for chaotic transport-dominated problems

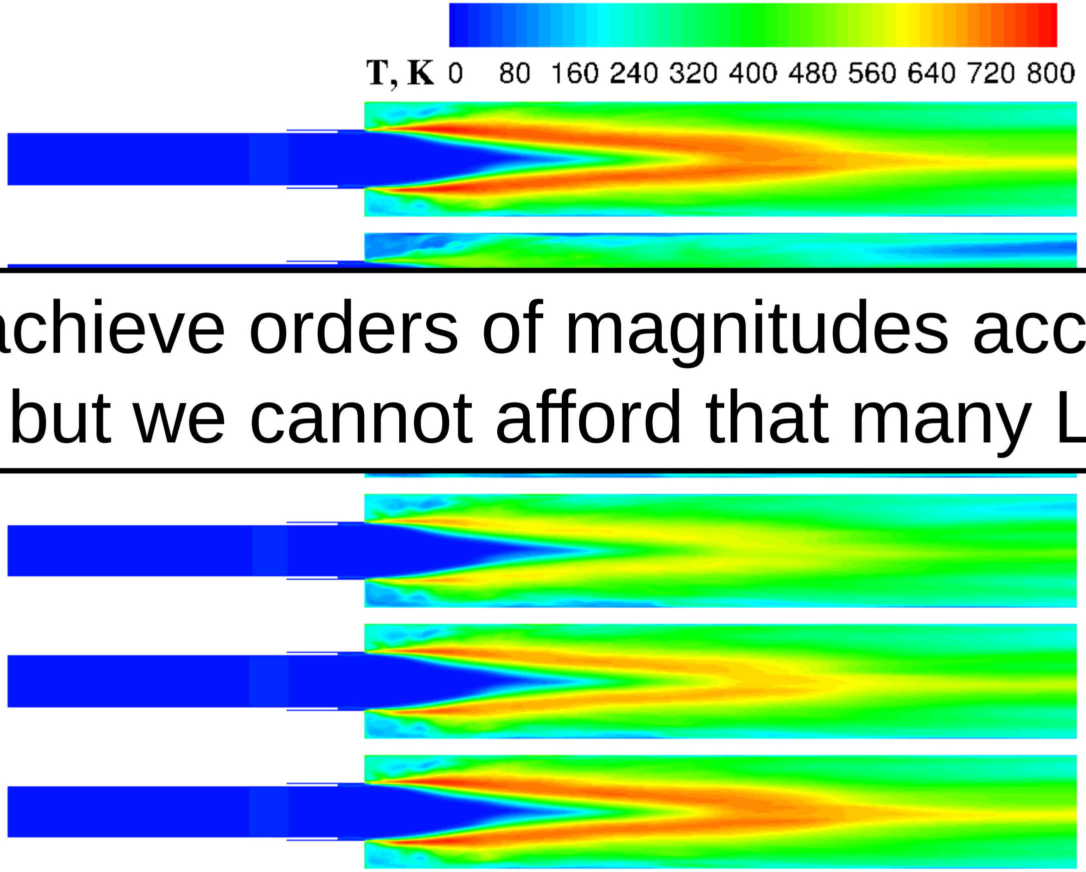
> **80%** training data needed to reach satisfying accuracy in QoI

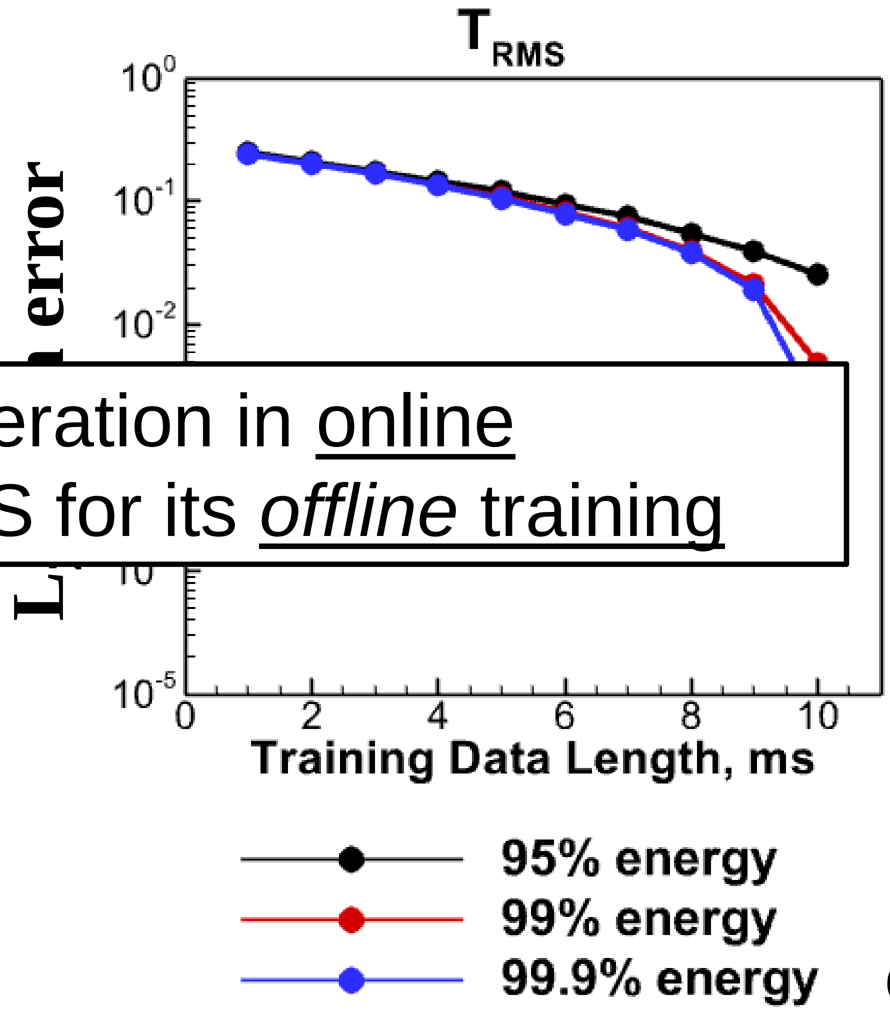
FOM

(6 ms training)

(8 ms training)

(10 ms training)





ROM can achieve orders of magnitudes acceleration in <u>online calculation</u> but we cannot afford that many LES for its <u>*offline* training</u>

Approaches to break the Kolmogorov barrier and enable predictive ROM

- **Local subspace**
  - Amsallem et al. (2012 & 2015), Geelen and Willcox (2022), …

- **Nonlinear manifold**
  - Lee and Carlberg (2019), Kim and Choi (2021), Barnett and Farhat (2022), Geelen and Willcox (2022), …

- **Adaptive MOR**
  - Peherstorfer (2015, 2020 & 2022), Ramezanian et al. (2021), Zucatti and Zahr (2023), …

Outline

- **Adaptive ROM Algorithm**
- **Test Case I**: 1D Freely Propagating Laminar Flame
- **Test Case II**: 2D Single-injector Rocket Combustor
- **Test Case III**: 2D Premixed RDE
- **Test Case IV**: 3D Single-injector Rocket Combustor
- **Test Case V**: 2D Single Injector with Variable Recess Length

Projection-based ROM with Adaptive Basis and Sampling

- Define fully discrete FOM equation residual, $\mathbf{r}$, *with* full states $\mathbf{q}_p$ approximated using basis changing with time ($\mathbf{V}^n$)

$$\frac{\mathrm{d}\mathbf{q}\left(\mathbf{q}_p\right)}{\mathrm{d}t} = \mathbf{f}\left(\mathbf{q}_p\right) \quad \rightarrow \quad \mathbf{r}\left(\tilde{\mathbf{q}}_p^n\right) \triangleq \frac{\mathbf{q}\left(\tilde{\mathbf{q}}_p^n\right) - \mathbf{q}\left(\tilde{\mathbf{q}}_p^{n-1}\right)}{\Delta t} - \mathbf{f}\left(\tilde{\mathbf{q}}_p^n\right) \text{ where } \tilde{\mathbf{q}}_p^n \triangleq \mathbf{q}_{p,ref} + \mathbf{V}^n \mathbf{q}_r^n$$

- Approximate $\mathbf{r}$ using hyper-reduction *with* sampling points changing with time ($\mathbf{S}_n$)

$$\bar{\mathbf{r}}\left(\tilde{\mathbf{q}}_p^n\right) \triangleq \mathbf{U}\left(\mathbf{S}_n^T \mathbf{U}\right)^+ \mathbf{S}_n^T \mathbf{r}\left(\tilde{\mathbf{q}}_p^n\right) \text{ with } \mathbf{U} = \mathbf{V}^n$$

- Minimize the hyper-reduced residual, $\bar{\mathbf{r}}$

$$\left\{\mathbf{q}_r^n, \mathbf{V}^n, \mathbf{S}_n\right\} = \underset{\mathbf{q}_r^n, \mathbf{V}^n, \mathbf{S}_n}{\arg\min} \left\| \mathbf{V}^n \left(\mathbf{S}_n^T \mathbf{V}^n\right)^+ \mathbf{S}_n^T \mathbf{r}\left(\mathbf{q}_{p,ref} + \mathbf{V}^n \mathbf{q}_r^n\right) \right\|_2^2$$

**intractable

We seek to solve three sequential minimization problem instead

$$\left\{\mathbf{q}_r^n, \mathbf{V}^n, \mathbf{S}_n\right\} = \underset{\mathbf{q}_r^n, \mathbf{V}^n, \mathbf{S}_n}{\arg\min} \left\| \mathbf{V}^n \left(\mathbf{S}_n^T \mathbf{V}^n\right)^+ \mathbf{S}_n^T \mathbf{r}\left(\mathbf{q}_{p,ref} + \mathbf{V}^n \mathbf{q}_r^n\right) \right\|_2^2$$

$$\Downarrow$$

$$\mathbf{q}_r^n = \underset{\mathbf{q}_r^n}{\arg\min} \left\| \mathbf{V}^{n-1} \left(\mathbf{S}_{n-1}^T \mathbf{V}^{n-1}\right)^+ \mathbf{S}_{n-1}^T \mathbf{r}\left(\mathbf{q}_{p,ref} + \mathbf{V}^{n-1} \mathbf{q}_r^n\right) \right\|_2^2$$

$$\downarrow$$

$$\mathbf{V}^n = \underset{\mathbf{V}^n}{\arg\min} \left\| \mathbf{V}^n \left[\left(\mathbf{V}^{n-1}\right)^+ \mathbf{F}_n\right] - \mathbf{F}_n \right\|_2^2$$

$$\downarrow$$

$$\mathbf{S}_n = \underset{\mathbf{S}_n}{\arg\min} \left\| \mathbf{F}_n - \mathbf{V}^n \left(\mathbf{S}_n \mathbf{V}^n\right)^+ \mathbf{S}_n \mathbf{F}_n \right\|_2^2$$

**full-model state is estimated and collected during online ROM stage

Estimate full-model state from FOM equation residual

$$\mathbf{F}_n = \begin{bmatrix} \bar{\mathbf{q}}_p^n & \cdots & \bar{\mathbf{q}}_p^{n+w} \end{bmatrix}$$

$$\mathbf{S}_{n-1}^T \mathbf{r}\left(\underbrace{\mathbf{S}_{n-1}\mathbf{S}_{n-1}^T \bar{\mathbf{q}}_p^n}\right) = \frac{\mathbf{S}_{n-1}^T \mathbf{q}\left(\mathbf{S}_{n-1}\mathbf{S}_{n-1}^T \bar{\mathbf{q}}_p^n\right) - \mathbf{S}_{n-1}^T \mathbf{q}\left(\mathbf{S}_{n-1}\mathbf{S}_{n-1}^T \tilde{\mathbf{q}}_p^{n-1}\right)}{\Delta t} - \mathbf{S}_{n-1}^T \mathbf{f}\left(\mathbf{S}_{n-1}\mathbf{S}_{n-1}^T \tilde{\mathbf{q}}_p^n\right) = 0$$

**full-state info at <u>sampled</u> points

and

**full-state info at <u>unsampled</u> points

$$\overbrace{\mathbf{S}_{n-1}^* \mathbf{S}_{n-1}^{*T} \bar{\mathbf{q}}_p^n} = \tilde{\mathbf{q}}_p^n \quad \text{where} \quad \tilde{\mathbf{q}}_p^n = \mathbf{q}_{p,ref} + \mathbf{V}^{n-1}\mathbf{q}_r^n \quad \text{and} \quad \mathbf{S}_{n-1}\mathbf{S}_{n-1}^T \cup \mathbf{S}_{n-1}^* \mathbf{S}_{n-1}^{*T} = \mathbf{I}$$

Basis Adaptation (multi-step)

*** Peherstorfer, SIAM J. Sci. Comput. 2020**

$$\mathbf{V}^n = \arg\min_{\mathbf{V}^n} \left\| \mathbf{V}^n \left[\left(\mathbf{V}^{n-1} \right)^+ \mathbf{F}_n \right] - \mathbf{F}_n \right\|_2^2$$

$$\downarrow$$

$$\{\boldsymbol{\alpha}_n, \boldsymbol{\beta}_n\} = \arg\min_{\boldsymbol{\alpha}_n, \boldsymbol{\beta}_n} \left\| \left(\mathbf{V}^{n-1} + \boldsymbol{\alpha}_n \boldsymbol{\beta}_n^T \right) \left[\left(\mathbf{V}^{n-1} \right)^+ \mathbf{F}_n \right] - \mathbf{F}_n \right\|_2^2$$

$$\downarrow$$

$\boldsymbol{\alpha}_n$ and $\boldsymbol{\beta}_n$ solved from svd of $\mathbf{R} = \mathbf{V}^{n-1} \left[\left(\mathbf{V}^{n-1} \right)^+ \mathbf{F}_n \right] - \mathbf{F}_n$

Basis Adaptation (one-step)

*** Huang and Duraisamy, JCP 2023**

$$\mathbf{V}^n = \underset{\mathbf{V}^n}{\arg\min} \left\| \mathbf{V}^n \left[\left(\mathbf{V}^{n-1} \right)^+ \bar{\mathbf{q}}_p^n \right] - \bar{\mathbf{q}}_p^n \right\|_2^2$$

$$\Downarrow$$

$$\delta\mathbf{V}^{n-1} = \underset{\delta\mathbf{V}^{n-1}}{\arg\min} \left\| \left(\mathbf{V}^{n-1} + \delta\mathbf{V}^{n-1} \right) \left[\left(\mathbf{V}^{n-1} \right)^+ \bar{\mathbf{q}}_p^n \right] - \bar{\mathbf{q}}_p^n \right\|_2^2$$

$$\downarrow$$

$$\delta\mathbf{V}^{n-1} = \frac{\left(\bar{\mathbf{q}}_p^n - \tilde{\mathbf{q}}_p^n \right)\left(\mathbf{q}_r^n \right)^T}{\left\| \mathbf{q}_r^n \right\|_2^2} \quad \text{where } \tilde{\mathbf{q}}_p^n = \mathbf{q}_{p,ref} + \mathbf{V}^{n-1}\mathbf{q}_r^n$$

Sampling Points Adaptation (every z_s time step)

$$\mathbf{S}_n = \underset{\mathbf{S}_n}{\arg\min} \left\| \mathbf{F}_n - \mathbf{V}^n \left(\mathbf{S}_n \mathbf{V}^n \right)^+ \mathbf{S}_n \mathbf{F}_n \right\|_2^2$$

$\Downarrow$

Estimate full - state info at all the points:

$$\mathbf{r}\left(\bar{\mathbf{q}}_p^n\right) = \frac{\mathbf{q}\left(\bar{\mathbf{q}}_p^n\right) - \mathbf{q}\left(\tilde{\mathbf{q}}_p^{n-1}\right)}{\Delta t} - \mathbf{f}\left(\bar{\mathbf{q}}_p^n\right) = 0$$

← **expensive step so the sampling points are adapted less frequently

$\downarrow$

Evaluate interpolation error:

$$\mathbf{e}_s^n = \bar{\mathbf{q}}_p^n - \mathbf{V}^n \left(\mathbf{S}_n \mathbf{V}^n \right)^+ \mathbf{S}_n \bar{\mathbf{q}}_p^n$$

$\downarrow$

Assign sampling points to highest values of $\mathbf{e}_s^n$

Adaptive ROM Algorithm

Solve FOM for a small time window (w_{init}) → obtain the initial basis $\mathbf{V}^0$ and sampling points $\mathbf{S}_0$

for $n = w_{\text{init}} + 1, \ldots, M$

Propagate the reduced states:

$$\mathbf{q}_r^n = \underset{\mathbf{q}_r^n}{\arg\min} \left\| \mathbf{V}^{n-1} \left(\mathbf{S}_{n-1}^T \mathbf{V}^{n-1} \right)^+ \mathbf{S}_{n-1}^T \mathbf{r} \left(\mathbf{q}_{p,ref} + \mathbf{V}^{n-1} \mathbf{q}_r^n \right) \right\|_2^2 \quad \Rightarrow \quad \boxed{\mathbf{q}_r^{n-1} \rightarrow \mathbf{q}_r^n}$$

if mod(n,z_s) == 0 *or* n == w_{init} +1 **then**

Estimate the full-model state: $\mathbf{S}_{n-1}^T \mathbf{r}\left(\mathbf{S}_{n-1} \mathbf{S}_{n-1}^T \bar{\mathbf{q}}_p^n \right) = 0$ and $\mathbf{S}_{n-1}^* \mathbf{S}_{n-1}^{*T} \bar{\mathbf{q}}_p^n = \tilde{\mathbf{q}}_p^n$

Update the basis: $\mathbf{V}^n = \mathbf{V}^{n-1} + \delta \mathbf{V}^{n-1}$ with $\delta \mathbf{V}^{n-1} = \left(\bar{\mathbf{q}}_p^n - \tilde{\mathbf{q}}_p^n \right) \left(\mathbf{q}_r^n \right)^T / \left\| \mathbf{q}_r^n \right\|_2^2$

Update the sampling points: $\mathbf{e}_s^n = \bar{\mathbf{q}}_p^n - \mathbf{V}^n \left(\mathbf{S}_n \mathbf{V}^n \right)^+ \mathbf{S}_n \bar{\mathbf{q}}_p^n \quad \Rightarrow \quad \boxed{\mathbf{S}_{n-1} \rightarrow \mathbf{S}_n}$

else

Estimate the full-model state: $\mathbf{S}_{n-1}^T \mathbf{r}\left(\mathbf{S}_{n-1} \mathbf{S}_{n-1}^T \bar{\mathbf{q}}_p^n \right) = 0$ and $\mathbf{S}_{n-1}^* \mathbf{S}_{n-1}^{*T} \bar{\mathbf{q}}_p^n = \tilde{\mathbf{q}}_p^n$

Update the basis: $\mathbf{V}^n = \mathbf{V}^{n-1} + \delta \mathbf{V}^{n-1}$ with $\delta \mathbf{V}^{n-1} = \left(\bar{\mathbf{q}}_p^n - \tilde{\mathbf{q}}_p^n \right) \left(\mathbf{q}_r^n \right)^T / \left\| \mathbf{q}_r^n \right\|_2^2$

end if

end for

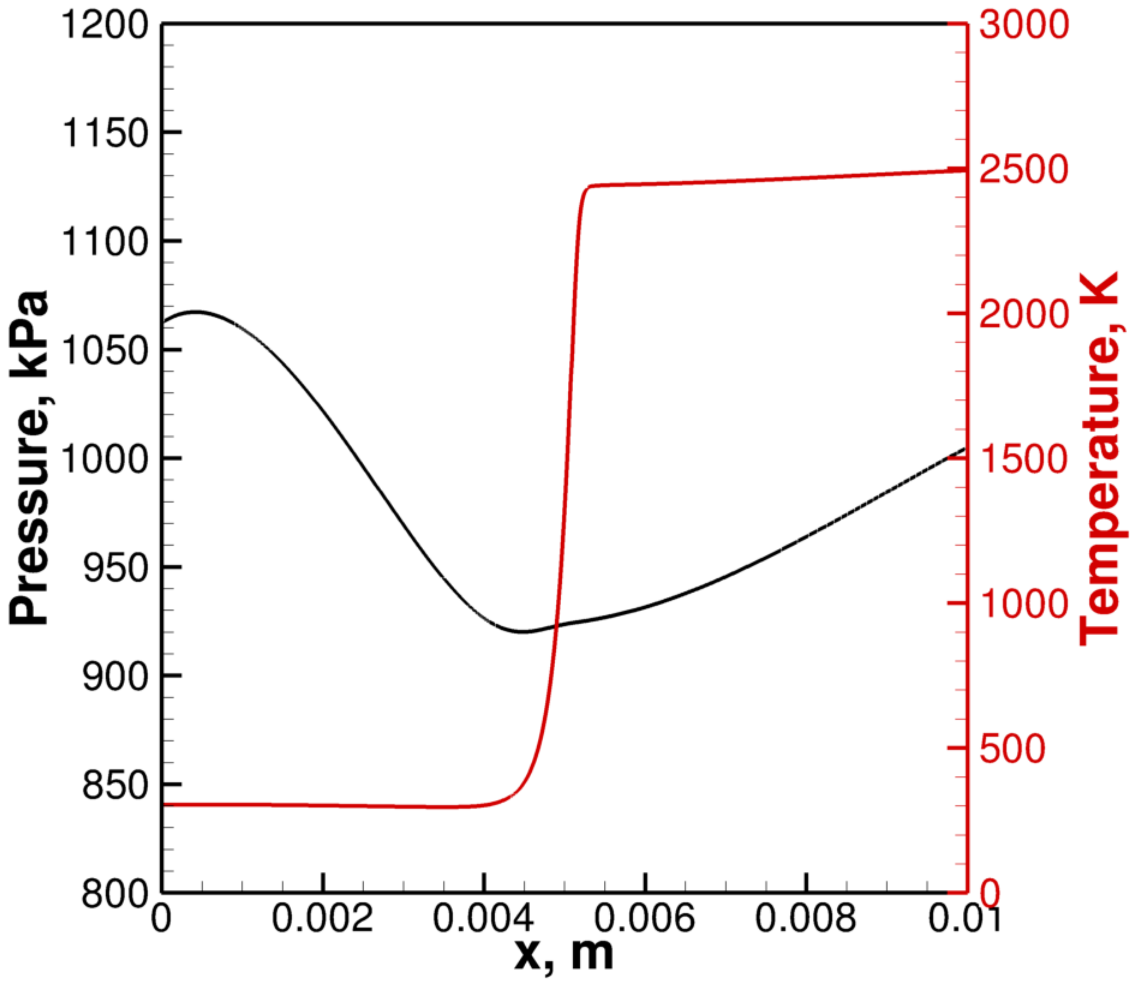
Outline

- **Adaptive ROM Algorithm**
- **Test Case I**: 1D Freely Propagating Laminar Flame
  - Incorporation of non-local coherence in adaptive ROM algorithm
  - Accuracy, efficiency, and *parametric* preditions
- **Test Case II**: 2D Single-injector Rocket Combustor
- **Test Case III**: 2D Premixed RDE
- **Test Case IV**: 3D Single-injector Rocket Combustor
- **Test Case V**: 2D Single Injector with Variable Recess Length

Test Case I: 1D Freely Propagating Laminar Flame

- **Inlet BC:** Non-reflective
- **Outlet BC:** Non-reflective with external acoustic perturbations
- **Single-step global reaction**:
  - Reactant → Product

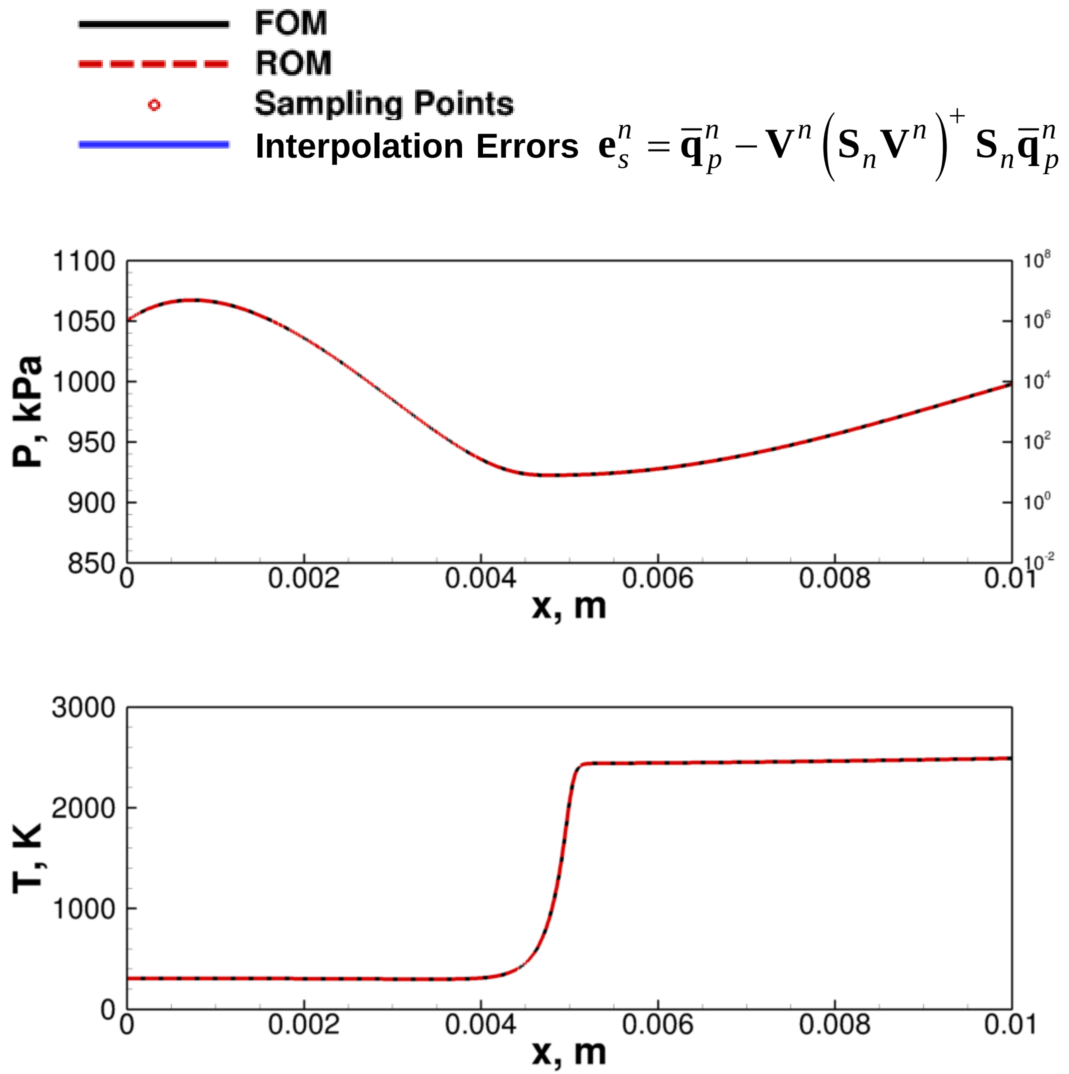
*** Test cases available in PERFORM**
https://romworkshop.engin.umich.edu/



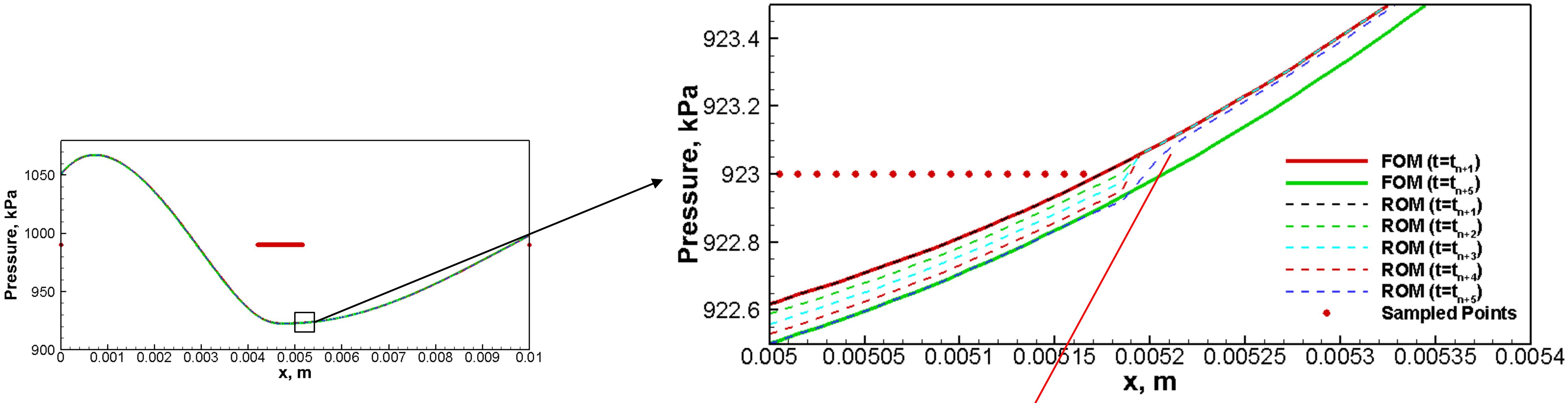
Adaptive ROM shows difficulties in predicting non-local dynamics

- *Offline* training: 10 snapshots
- Online testing: 4500 snapshots
- ROM dimension: 5
- Sampling points: 10% + z_s: 5

❖ Current adaptive ROM algorithm does not accommodate for non-local dynamic



Solutions between sampling points update



$$\mathbf{S}_{n-1}^T\mathbf{r}\left(\mathbf{S}_{n-1}\mathbf{S}_{n-1}^T\bar{\mathbf{q}}_p^n\right)=\frac{\mathbf{S}_{n-1}^T\mathbf{q}\left(\mathbf{S}_{n-1}\mathbf{S}_{n-1}^T\bar{\mathbf{q}}_p^n\right)-\mathbf{S}_{n-1}^T\mathbf{q}\left(\mathbf{S}_{n-1}\mathbf{S}_{n-1}^T\tilde{\mathbf{q}}_p^{n-1}\right)}{\Delta \mathrm{t}}-\mathbf{S}_{n-1}^T\mathbf{f}\left(\mathbf{S}_{n-1}\mathbf{S}_{n-1}^T\bar{\mathbf{q}}_p^n\right)=0$$

** **inaccurate** ROM solutions at the *unsampled* points in between sampling point update
→ **inaccurate** full-state info estimate when the points are sampled
→ **inaccurate** basis update → rapid accumulations of errors …

Estimate full-state information FOM equation residual
incorporating non-local full-model state

If not update sampling

at *sampled* points: $\mathbf{S}_{n-1}^T \mathbf{r}\left(\mathbf{S}_{n-1}\mathbf{S}_{n-1}^T \bar{\mathbf{q}}_p^n\right) = \dfrac{\mathbf{S}_{n-1}^T \mathbf{q}\left(\mathbf{S}_{n-1}\mathbf{S}_{n-1}^T \bar{\mathbf{q}}_p^n\right) - \mathbf{S}_{n-1}^T \mathbf{q}\left(\mathbf{S}_{n-1}\mathbf{S}_{n-1}^T \tilde{\mathbf{q}}_p^{n-1}\right)}{\Delta t} - \mathbf{S}_{n-1}^T \mathbf{f}\left(\mathbf{S}_{n-1}\mathbf{S}_{n-1}^T \bar{\mathbf{q}}_p^n\right) = 0$

at *unsampled* points: $\mathbf{S}_{n-1}^* \mathbf{S}_{n-1}^{*T} \bar{\mathbf{q}}_p^n = \tilde{\mathbf{q}}_p^n$ where $\tilde{\mathbf{q}}_p^n = \mathbf{q}_{p,ref} + \mathbf{V}^{n-1}\mathbf{q}_r^n$ and $\mathbf{S}_{n-1}\mathbf{S}_{n-1}^T \cup \mathbf{S}_{n-1}^* \mathbf{S}_{n-1}^{*T} = \mathbf{I}$

If update sampling

at *sampled* points: $\mathbf{S}_{n-1}^T \mathbf{r}\left(\mathbf{S}_{n-1}\mathbf{S}_{n-1}^T \bar{\mathbf{q}}_p^n\right) = \dfrac{\mathbf{S}_{n-1}^T \mathbf{q}\left(\mathbf{S}_{n-1}\mathbf{S}_{n-1}^T \bar{\mathbf{q}}_p^n\right) - \mathbf{S}_{n-1}^T \mathbf{q}\left(\mathbf{S}_{n-1}\mathbf{S}_{n-1}^T \tilde{\mathbf{q}}_p^{n-1}\right)}{\Delta t} - \mathbf{S}_{n-1}^T \mathbf{f}\left(\mathbf{S}_{n-1}\mathbf{S}_{n-1}^T \bar{\mathbf{q}}_p^n\right) = 0$

at *unsampled* points: $\mathbf{S}_{n-1}^{*T} \mathbf{r}^*\left(\mathbf{S}_{n-1}^*\mathbf{S}_{n-1}^{*T} \bar{\mathbf{q}}_p^n\right) = \dfrac{\mathbf{S}_{n-1}^{*T} \mathbf{q}\left(\mathbf{S}_{n-1}^*\mathbf{S}_{n-1}^{*T} \bar{\mathbf{q}}_p^n\right) - \mathbf{S}_{n-1}^{*T} \mathbf{q}\left(\mathbf{S}_{n-1}^*\mathbf{S}_{n-1}^{*T} \tilde{\mathbf{q}}_p^{n-z_s}\right)}{z_s \Delta t} - \mathbf{S}_{n-1}^{*T} \mathbf{f}\left(\mathbf{S}_{n-1}^*\mathbf{S}_{n-1}^{*T} \bar{\mathbf{q}}_p^n\right) = 0$

** no additional computational cost required

Adaptive ROM Algorithm Incorporating Non-Local Info

Solve FOM for a small time window (w_{init}) → obtain the initial basis $\mathbf{V}^0$ and sampling points $\mathbf{S}_0$

for $n = w_{\text{init}} + 1, \ldots, M$

Propagate the reduced states:

$$\mathbf{q}_r^n = \underset{\mathbf{q}_r^n}{\arg\min} \left\| \mathbf{V}^{n-1} \left(\mathbf{S}_{n-1}^T \mathbf{V}^{n-1} \right)^+ \mathbf{S}_{n-1}^T \mathbf{r} \left(\mathbf{q}_{p,ref} + \mathbf{V}^{n-1} \mathbf{q}_r^n \right) \right\|_2^2 \quad \Rightarrow \quad \boxed{\mathbf{q}_r^{n-1} \rightarrow \mathbf{q}_r^n}$$

if mod(n,z_s) == 0 *or* n == w_{init} +1 **then**

Estimate the full-model state: $\mathbf{S}_{n-1}^T \mathbf{r}\left(\mathbf{S}_{n-1} \mathbf{S}_{n-1}^T \bar{\mathbf{q}}_p^n \right) = 0$ and $\mathbf{S}_{n-1}^{*T} \mathbf{r}\left(\mathbf{S}_{n-1}^* \mathbf{S}_{n-1}^{*T} \bar{\mathbf{q}}_p^n \right) = 0$

Update the basis: $\mathbf{V}^n = \mathbf{V}^{n-1} + \delta \mathbf{V}^{n-1}$ with $\delta \mathbf{V}^{n-1} = \left(\bar{\mathbf{q}}_p^n - \tilde{\mathbf{q}}_p^n \right) \left(\mathbf{q}_r^n \right)^T / \left\| \mathbf{q}_r^n \right\|_2^2$

Update the sampling points: $\mathbf{e}_s^n = \bar{\mathbf{q}}_p^n - \mathbf{V}^n \left(\mathbf{S}_n \mathbf{V}^n \right)^+ \mathbf{S}_n \bar{\mathbf{q}}_p^n \quad \Rightarrow \quad \boxed{\mathbf{S}_{n-1} \rightarrow \mathbf{S}_n}$

else

Estimate the full-model state: $\mathbf{S}_{n-1}^T \mathbf{r}\left(\mathbf{S}_{n-1} \mathbf{S}_{n-1}^T \bar{\mathbf{q}}_p^n \right) = 0$ and $\mathbf{S}_{n-1}^* \mathbf{S}_{n-1}^{*T} \bar{\mathbf{q}}_p^n = \tilde{\mathbf{q}}_p^n$

Update the basis: $\mathbf{V}^n = \mathbf{V}^{n-1} + \delta \mathbf{V}^{n-1}$ with $\delta \mathbf{V}^{n-1} = \left(\bar{\mathbf{q}}_p^n - \tilde{\mathbf{q}}_p^n \right) \left(\mathbf{q}_r^n \right)^T / \left\| \mathbf{q}_r^n \right\|_2^2$

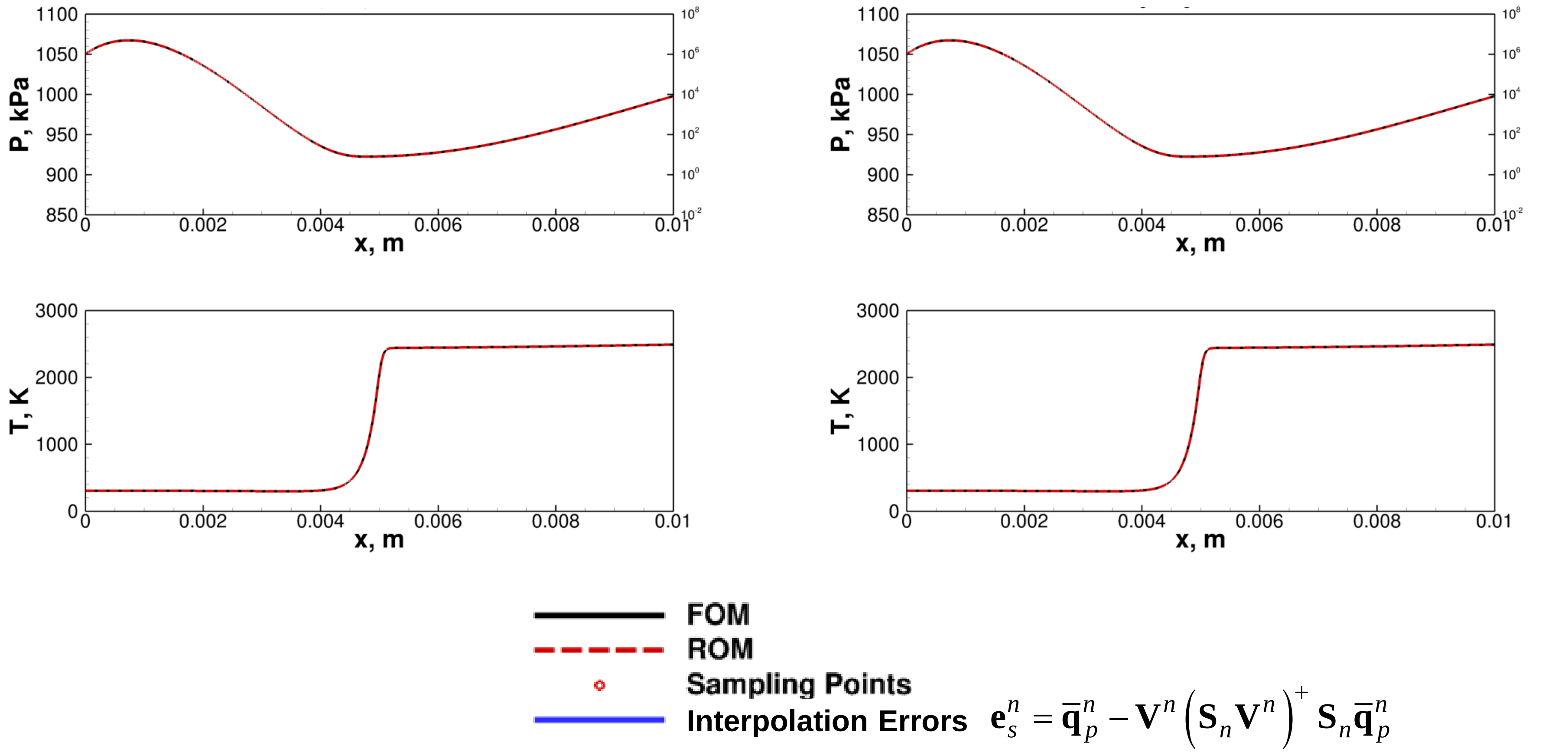
end if

end for

Incorporating non-local information is important

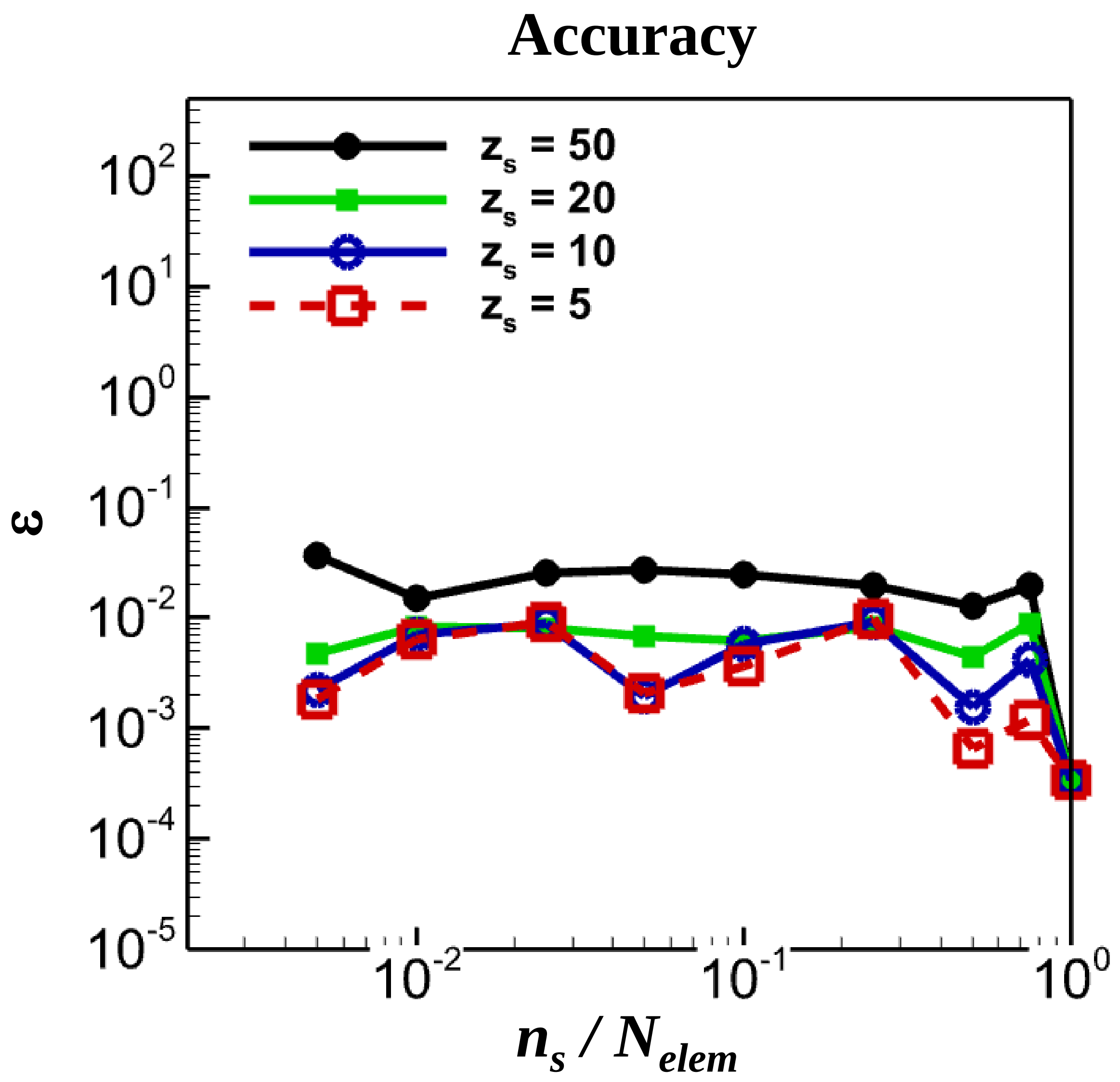
- *Offline* training: 10 snapshots
- Online testing: 4500 snapshots
- ROM dimension: 5 + Sampling points: 10% + z_s: 5

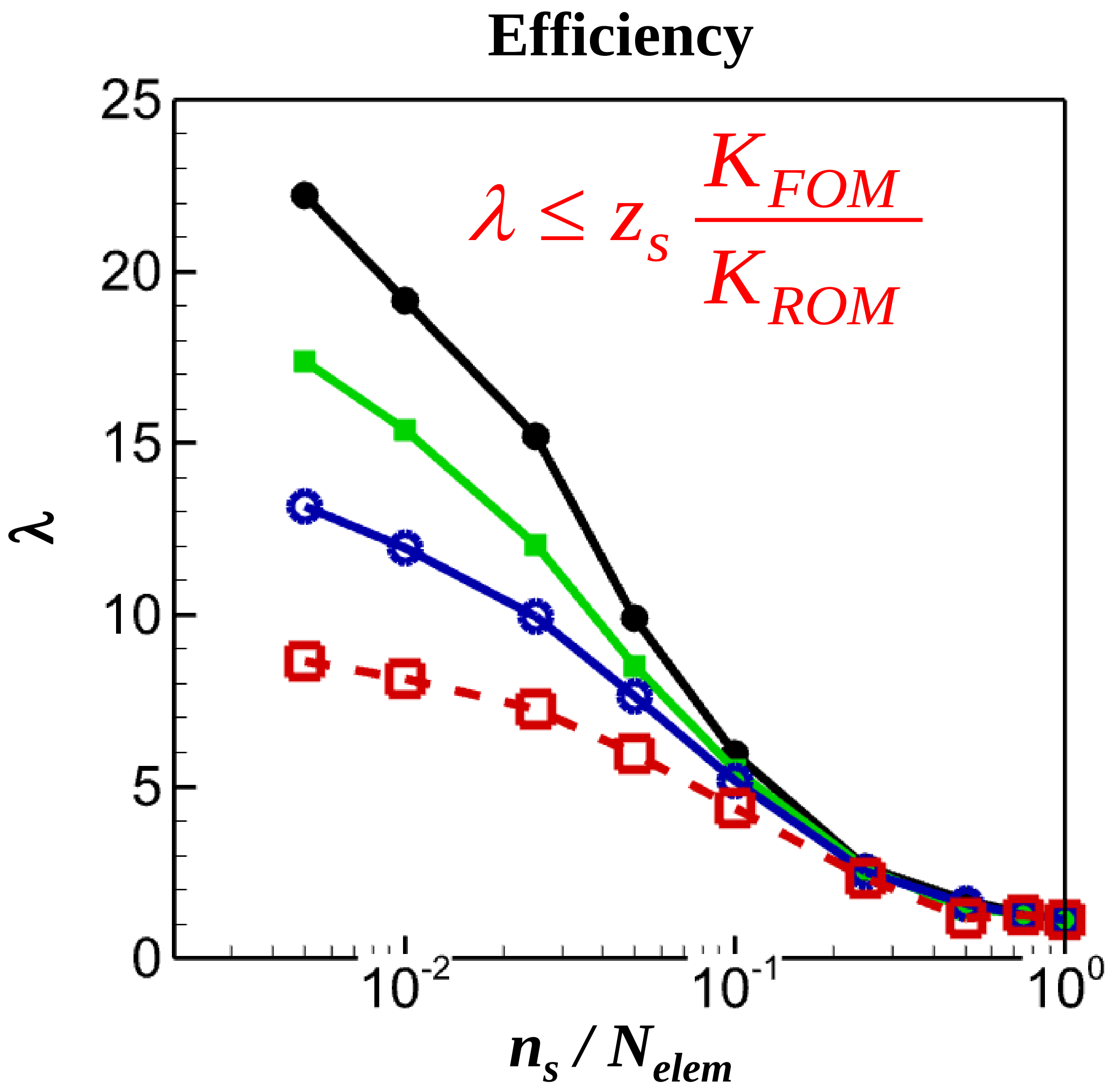
Without Non-local Info Incorporated **With Non-local Info Incorporated**



$$\mathbf{e}_s^n = \bar{\mathbf{q}}_p^n - \mathbf{V}^n \left(\mathbf{S}_n \mathbf{V}^n\right)^+ \mathbf{S}_n \bar{\mathbf{q}}_p^n$$

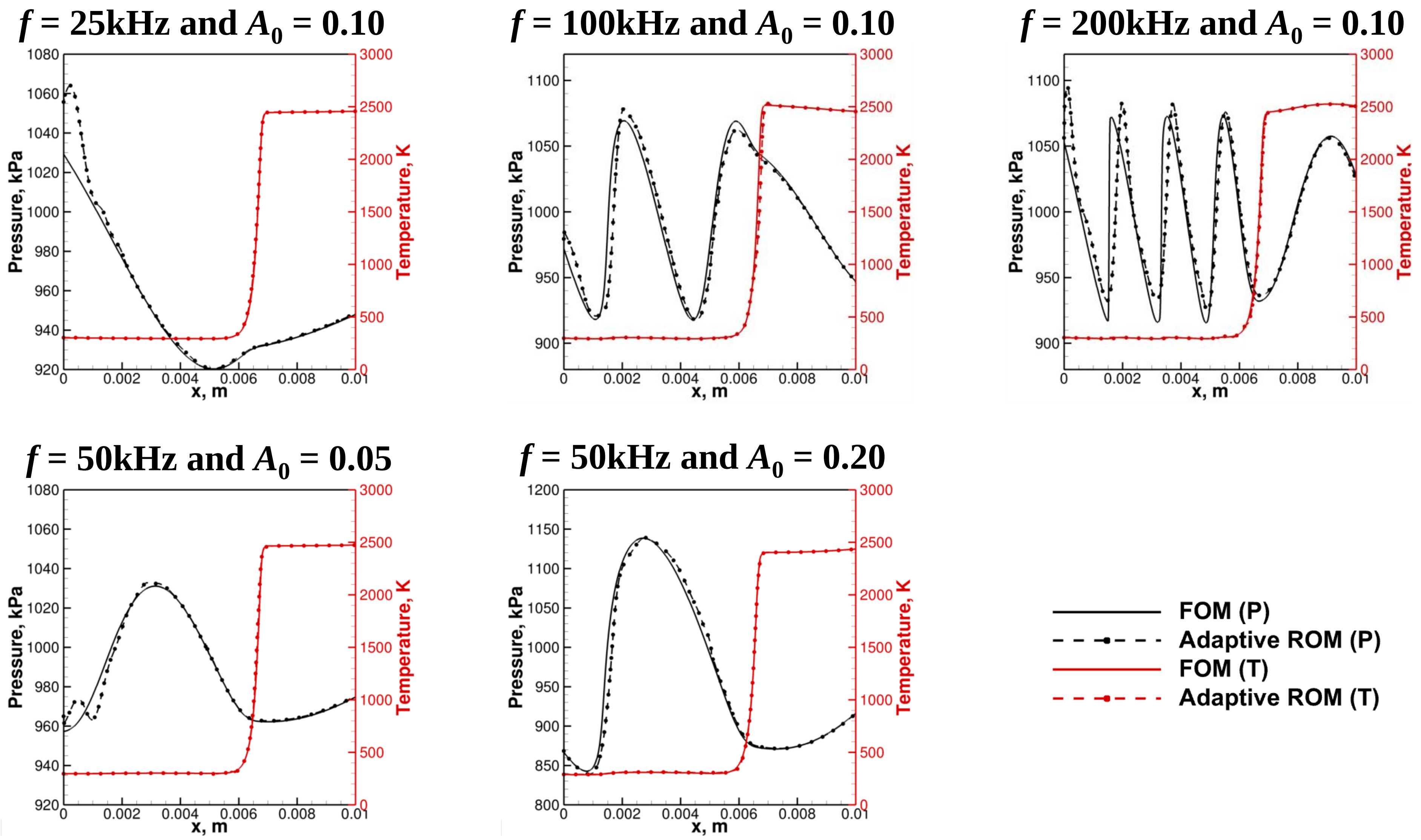
Performance of Adaptive ROM





Adaptive ROM inherently enables parametric predictions

** <u>*Offline* training</u>: 10 snapshots with ***f* = 50kHz and A_0 = 0.10**



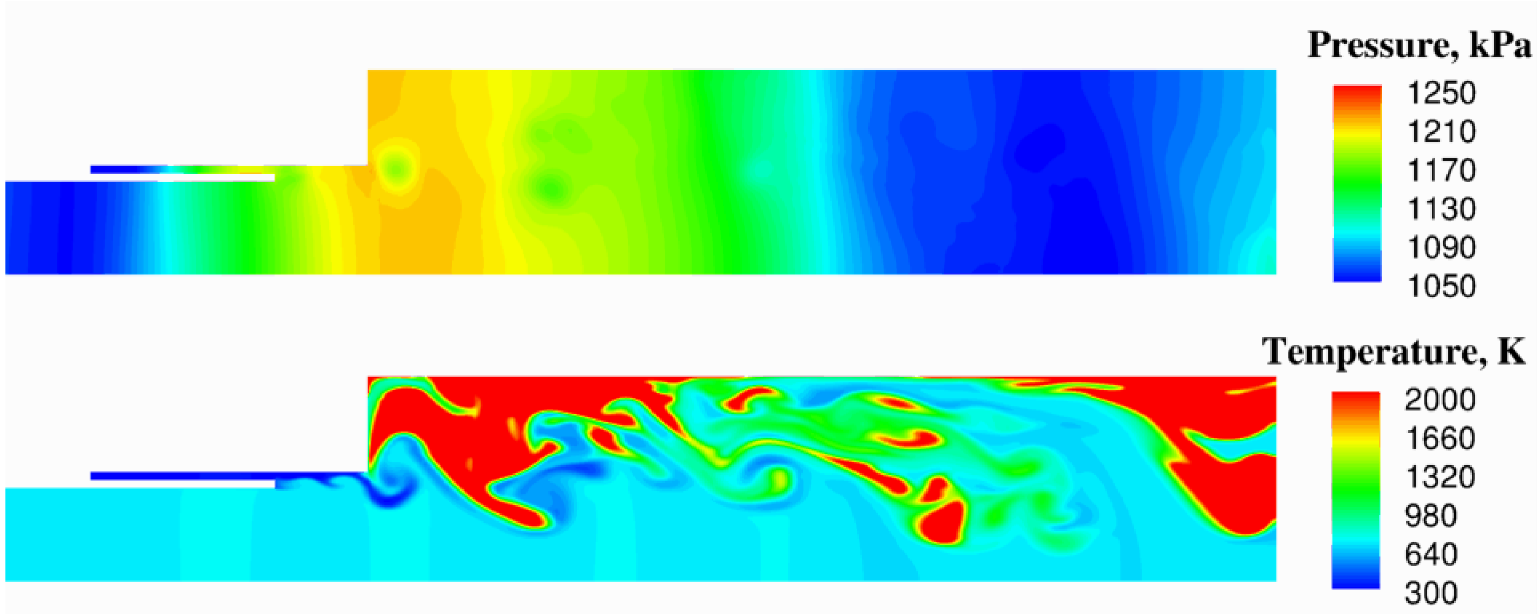
Outline

- **Adaptive ROM Algorithm**
- **Test Case I**: 1D Freely Propagating Laminar Flame
- **Test Case II**: 2D Single-injector Rocket Combustor
  - Long-term future-state predictions
  - Transience and parametric predictions
- **Test Case III**: 2D Premixed RDE
- **Test Case IV**: 3D Single-injector Rocket Combustor
- **Test Case V**: 2D Single Injector with Variable Recess Length

Test Case II: 2D Single-injector Rocket Combustor

- **Inlet BC:** T_{fuel} = 300K (100% CH_4)

  T_{ox} = 700K (42% O_2 + 58% H_2O)
- **Outlet BC:** Non-reflective with external acoustic perturbations
- Single-step global reaction: $CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O$



Data available at https://afcoe.engin.umich.edu/benchmark-data

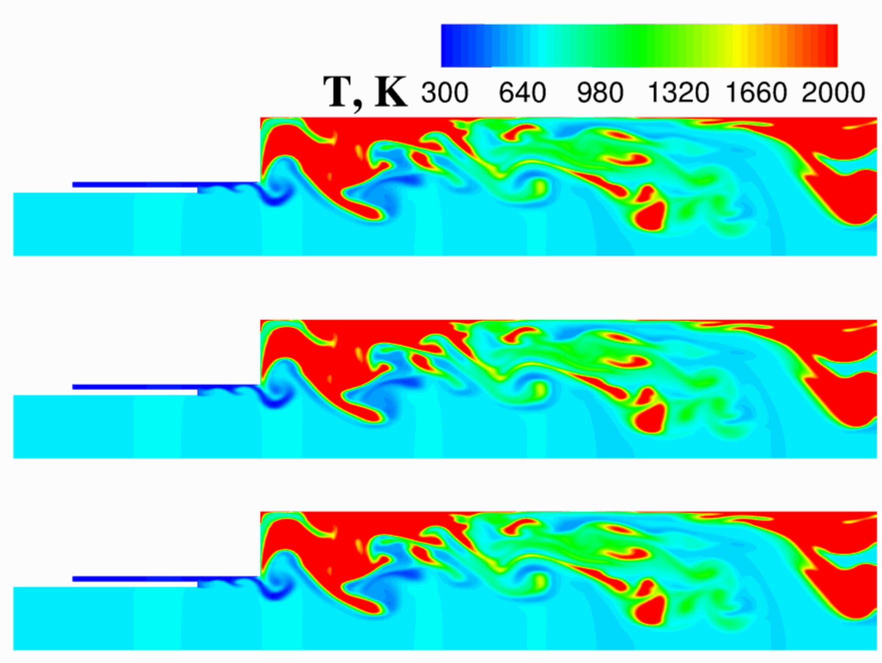
Adaptive ROM: 2D Single-injector Rocket Combustor

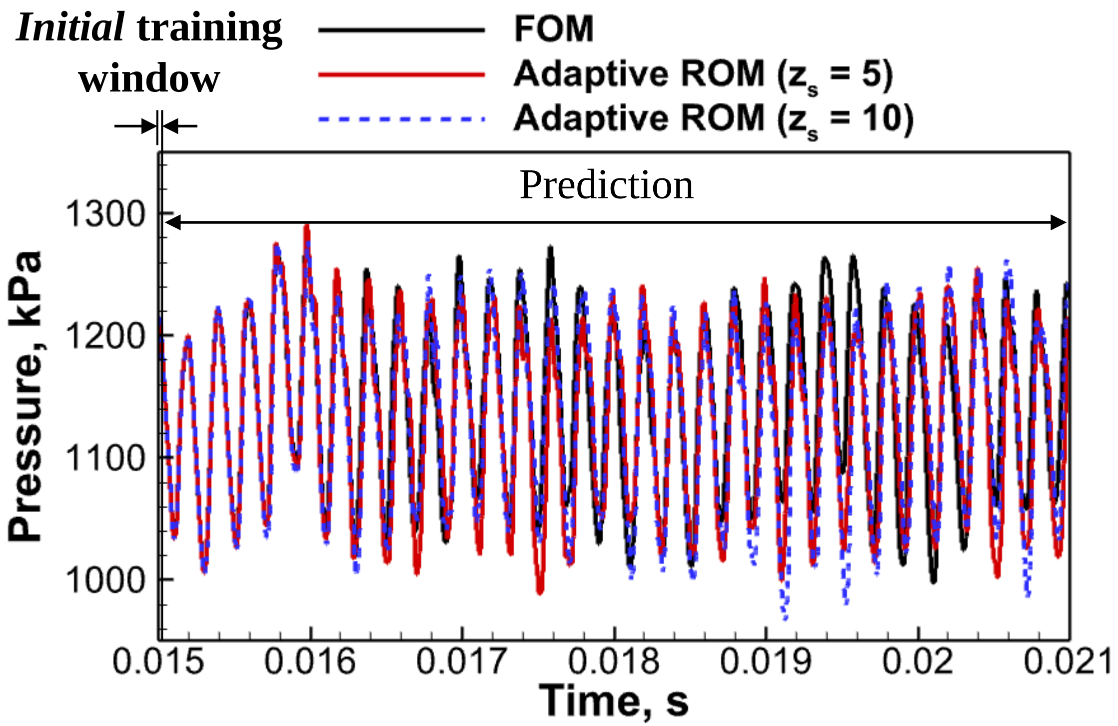
- **Initial training window**: 0.01 ms + **Dimension**: 5 + **Sampling points**: 1.0%
  - Prediction period: ~ 6 ms

FOM

Adaptive ROM
(z_s = 5)
~ O(9) acceleration

Adaptive ROM
(z_s = 10)
~ O(18) acceleration

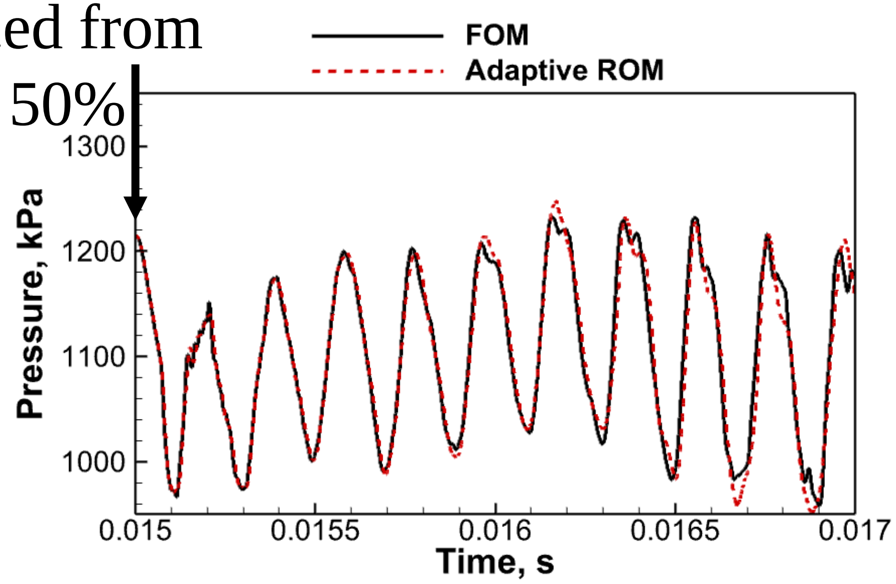




Adaptive ROM enables *transient* & *parametric* predictions

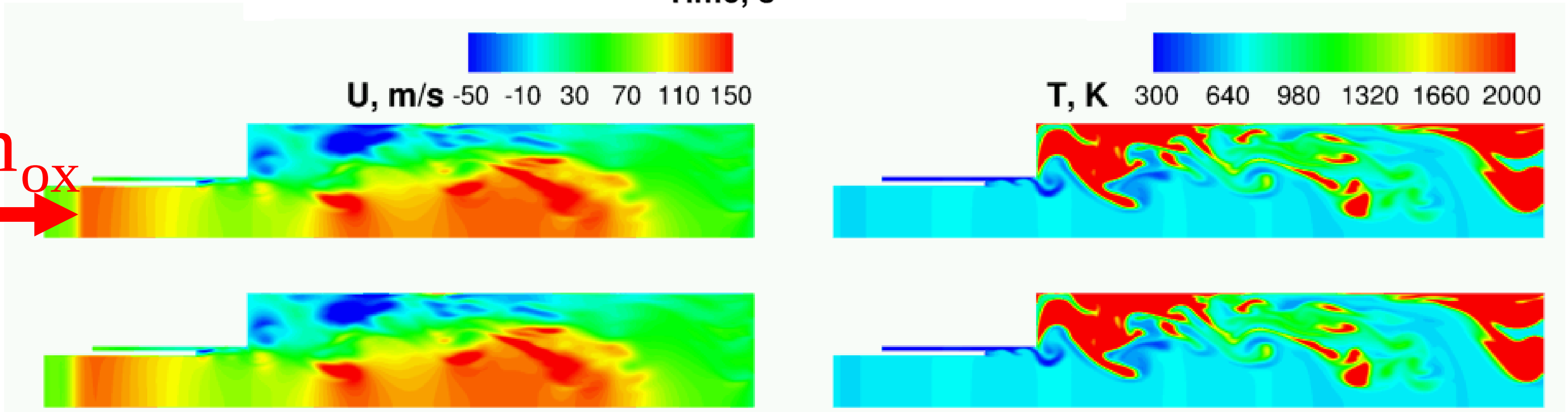
- **Training window**: 0.01 ms with 100% $\dot{m}_{ox}$ + **Dimension**: 5 + **Sampling points**: 1.0% + $\mathbf{z_s}$ = 10
  - Prediction period: ~ 2 ms with 50% $\dot{m}_{ox}$

* $\dot{m}_{ox}$ reduced from 100% to 50%



FOM

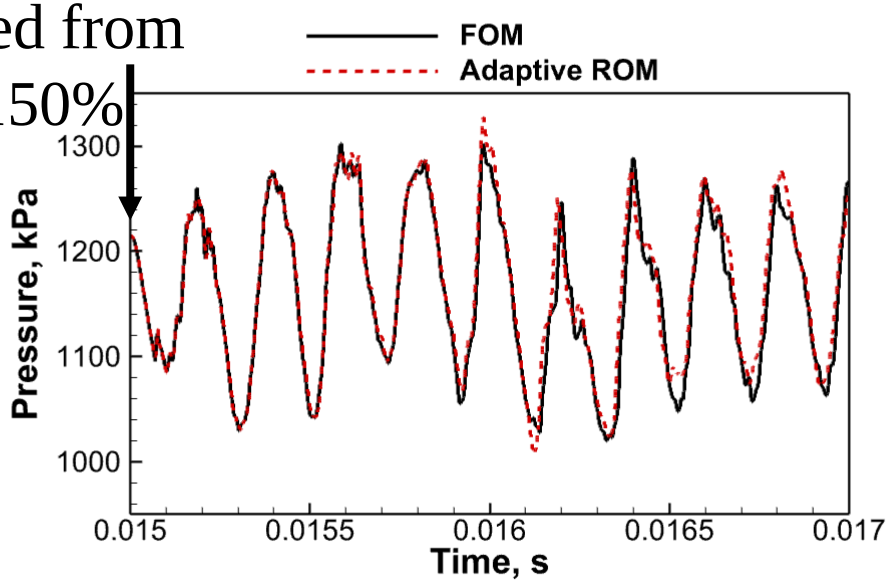
Adaptive ROM
(zs = 10)



Adaptive ROM enables *transient* & *parametric* predictions

- **Training window**: 0.01 ms with 100% $\dot{m}_{ox}$ + **Dimension**: 5 + **Sampling points**: 1.0% + $\mathbf{z}_s$ = 10
  - <u>Prediction period</u>: ~ 2 ms with 150% $\dot{m}_{ox}$

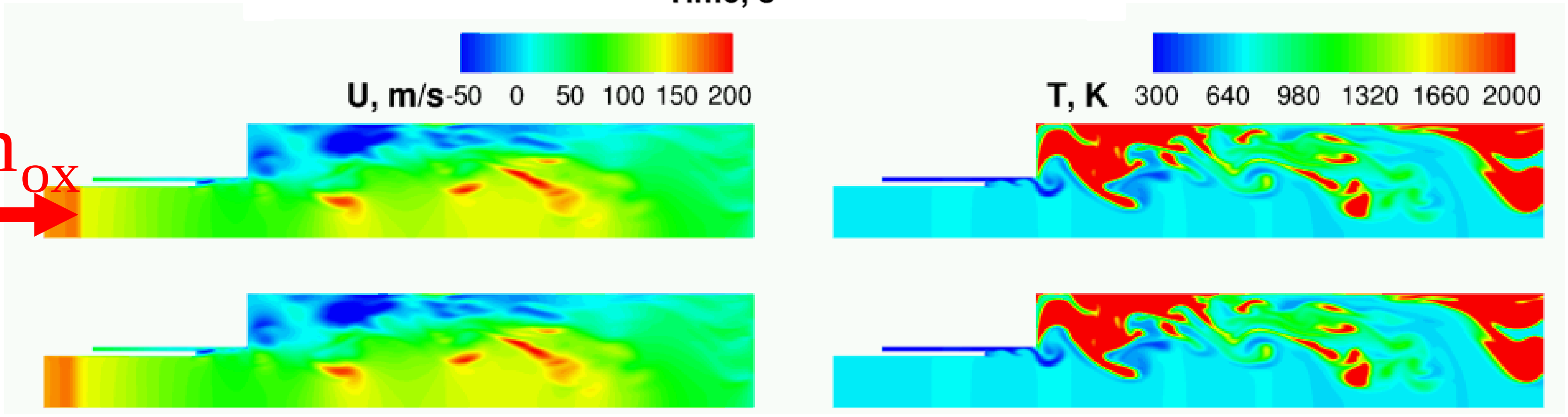
* $\dot{m}_{ox}$ reduced from 100% to 150%



$\dot{m}_{ox}$

FOM

Adaptive ROM
(zs = 10)

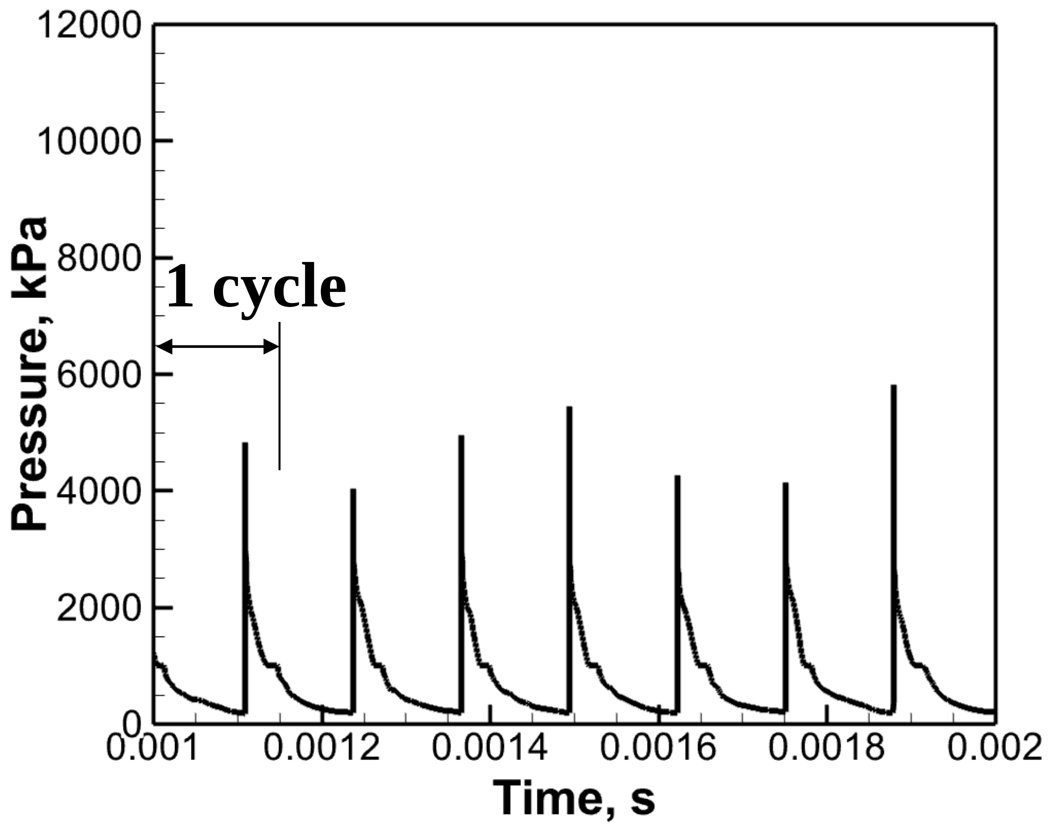


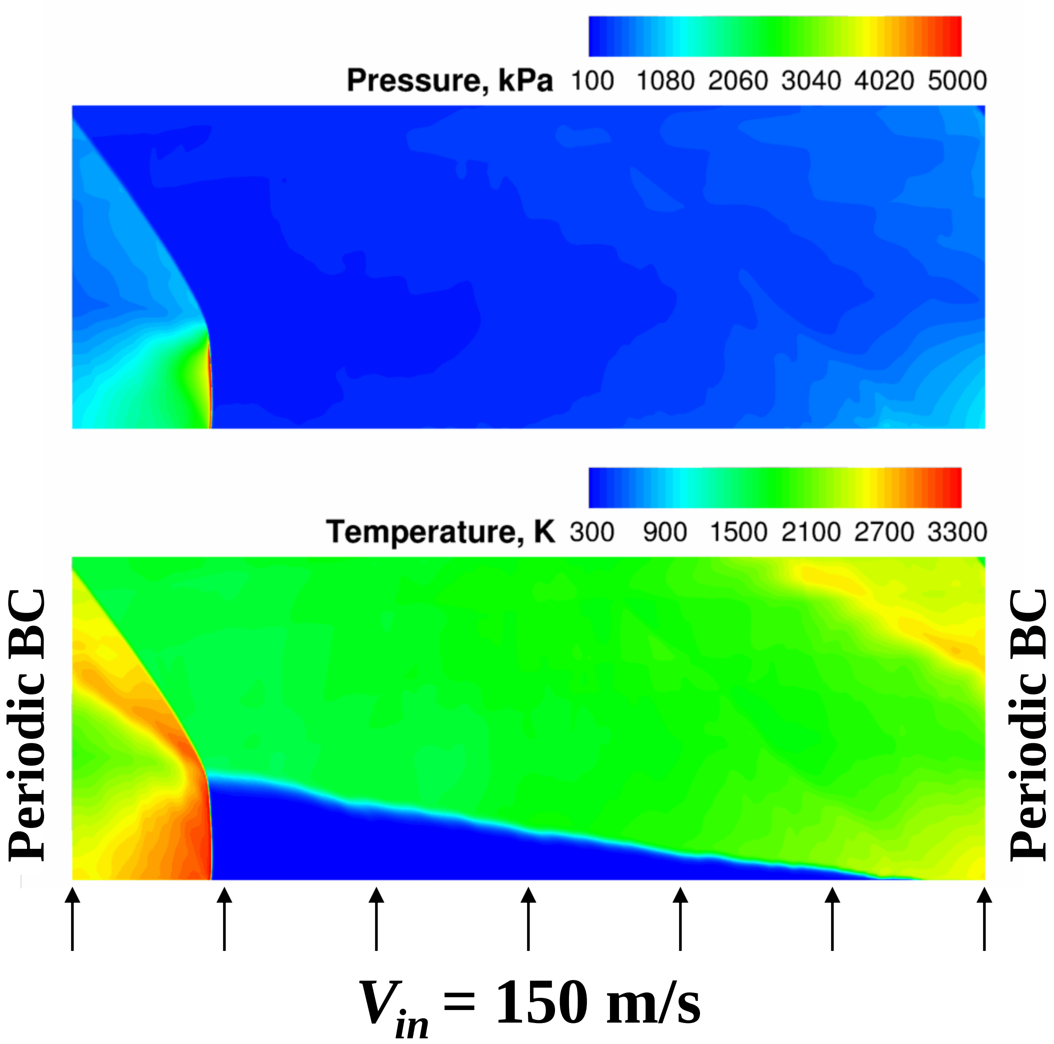
Outline

- **Adaptive ROM Algorithm**
- **Test Case I**: 1D Freely Propagating Laminar Flame
- **Test Case II**: 2D Single-injector Rocket Combustor
- **Test Case III**: 2D Premixed RDE
  - Parametric and initial transience predictions
- **Test Case IV**: 3D Single-injector Rocket Combustor
- **Test Case V**: 2D Single Injector with Variable Recess Length

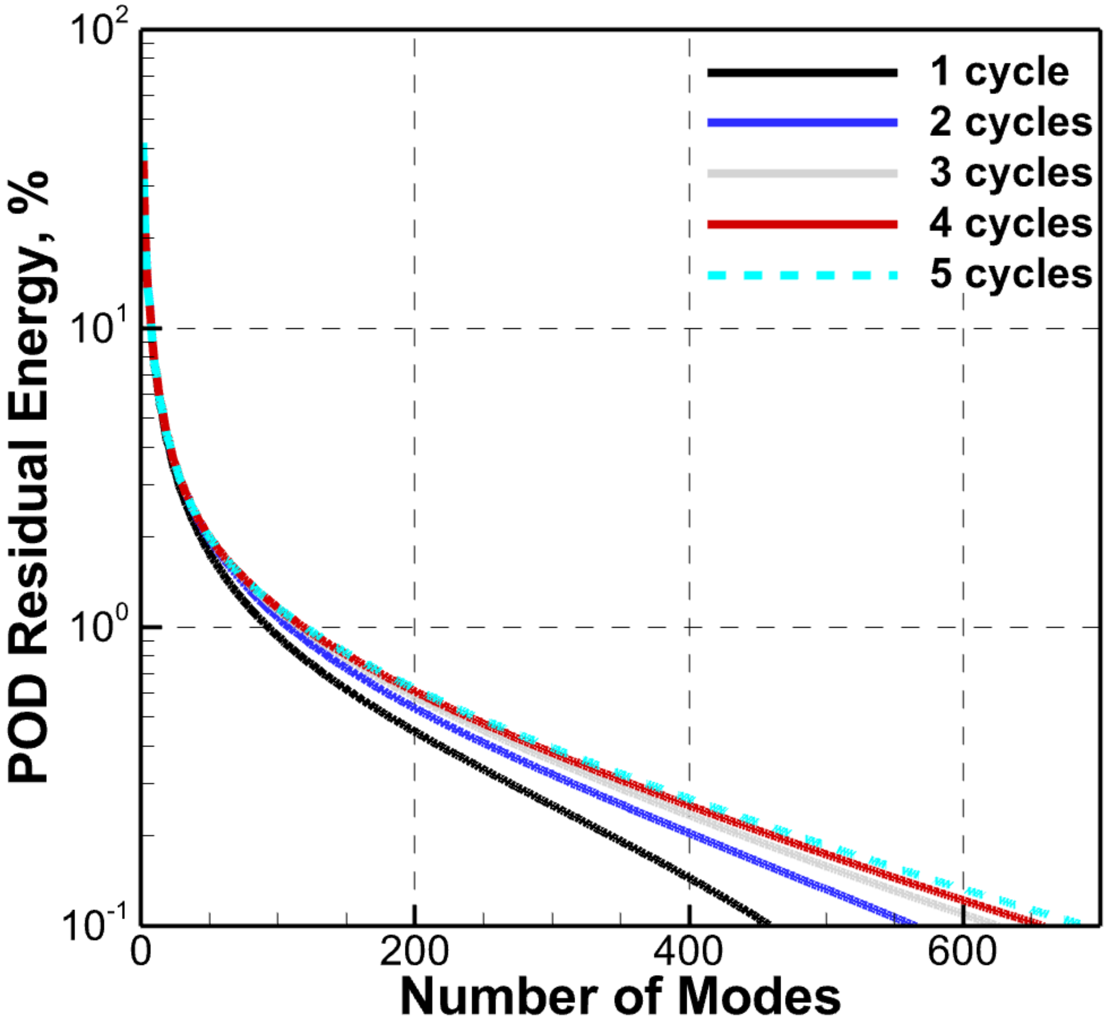
Test Case III: 2D Premixed RDE

- **Grid points**: 178,290
- **Time step**: 2 ns





POD Residual Energy *vs* Training Data Amount



Slow decays of Kolmogorov N-width with linear subspace approximation

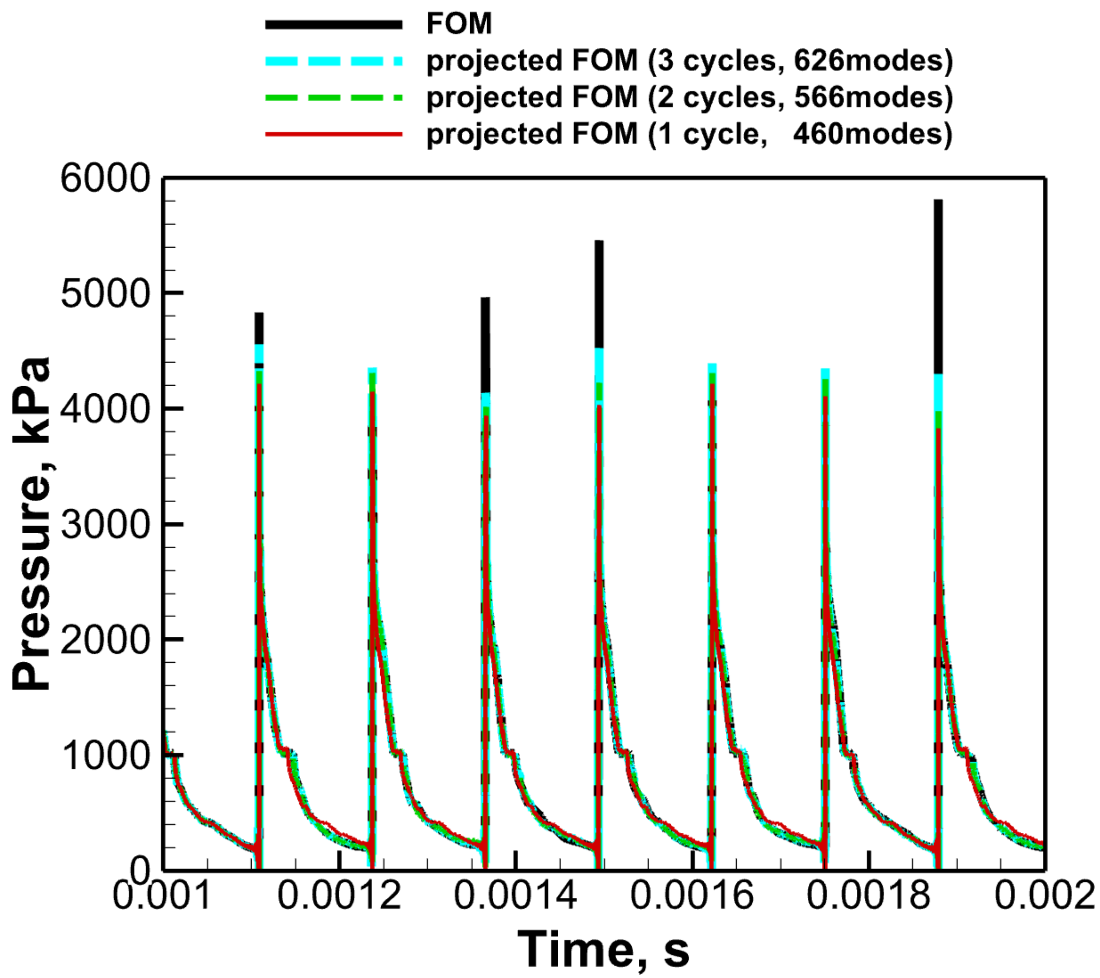
- Significant number of basis modes are required to recover accurate representations of the target physics

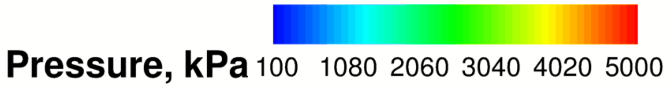
Convergence of low-rank approx. versus training data amount

- 1-cycle training data seems to be sufficient for accurate future-state predictions

FOM ($\mathbf{q}_p$) *vs* Projected FOM ($\mathbf{q}_{p,ref} + \mathbf{V}\mathbf{V}^T \mathbf{q}_p$)

- 1-cycle training data is sufficient for accurate future-state predictions





FOM

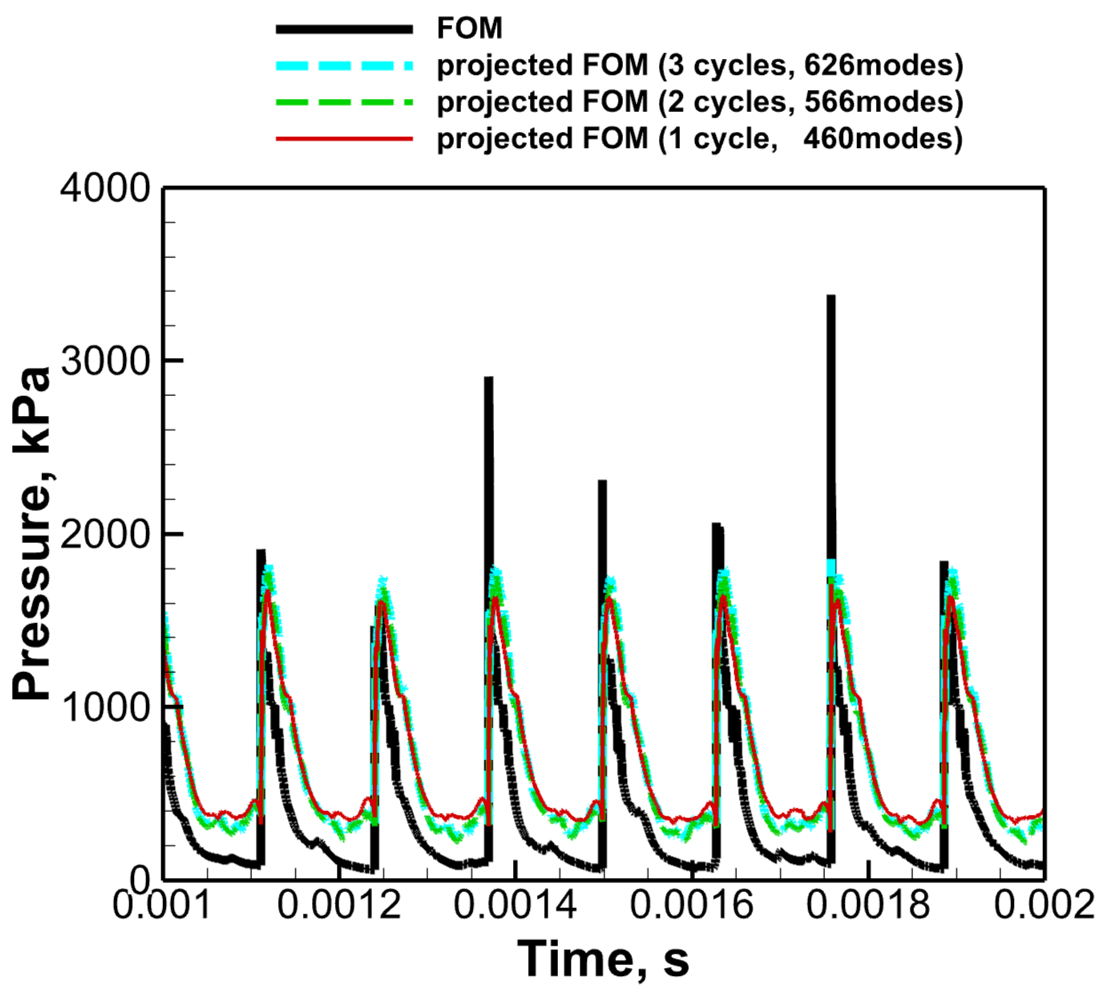
Projected FOM
(1 cycle, 460 modes)

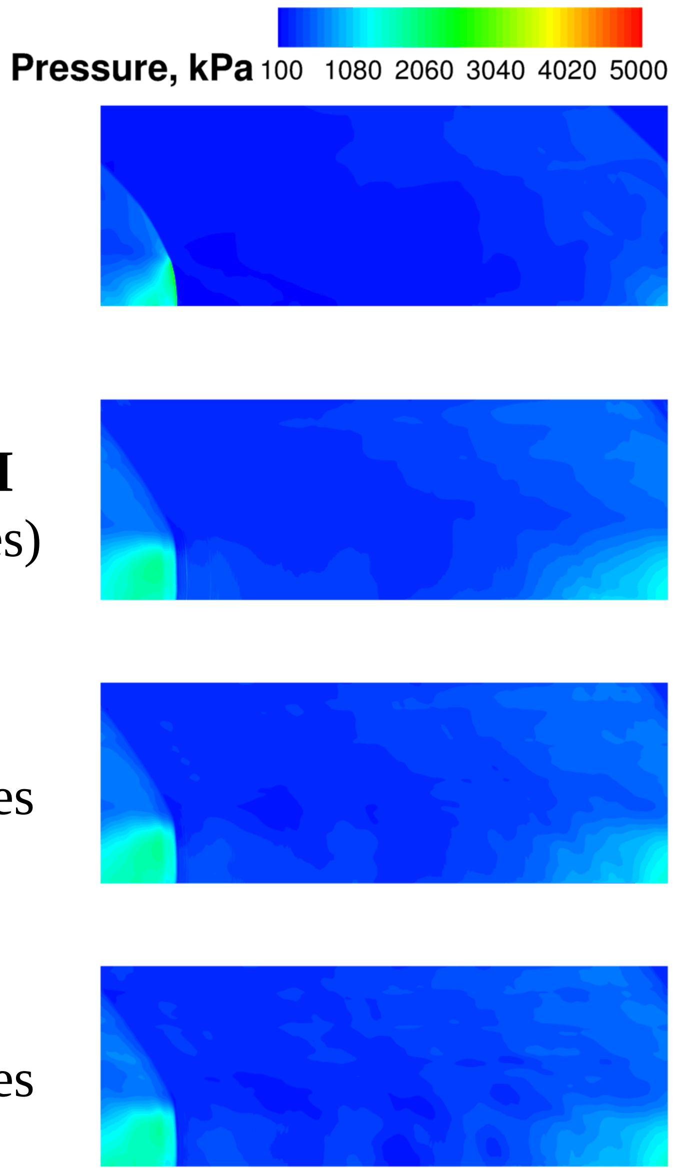
2 cycles, 566 modes

3 cycles, 626 modes

FOM ($\mathbf{q}_p$) *vs* Projected FOM ($\mathbf{q}_{p,ref} + \mathbf{V}\mathbf{V}^T \mathbf{q}_p$): Predicting V_{in} = 100m/s

- POD basis trained using snapshots from V_{in} = 150 m/s
- Static basis exhibits deficiency in parametric predictions





FOM

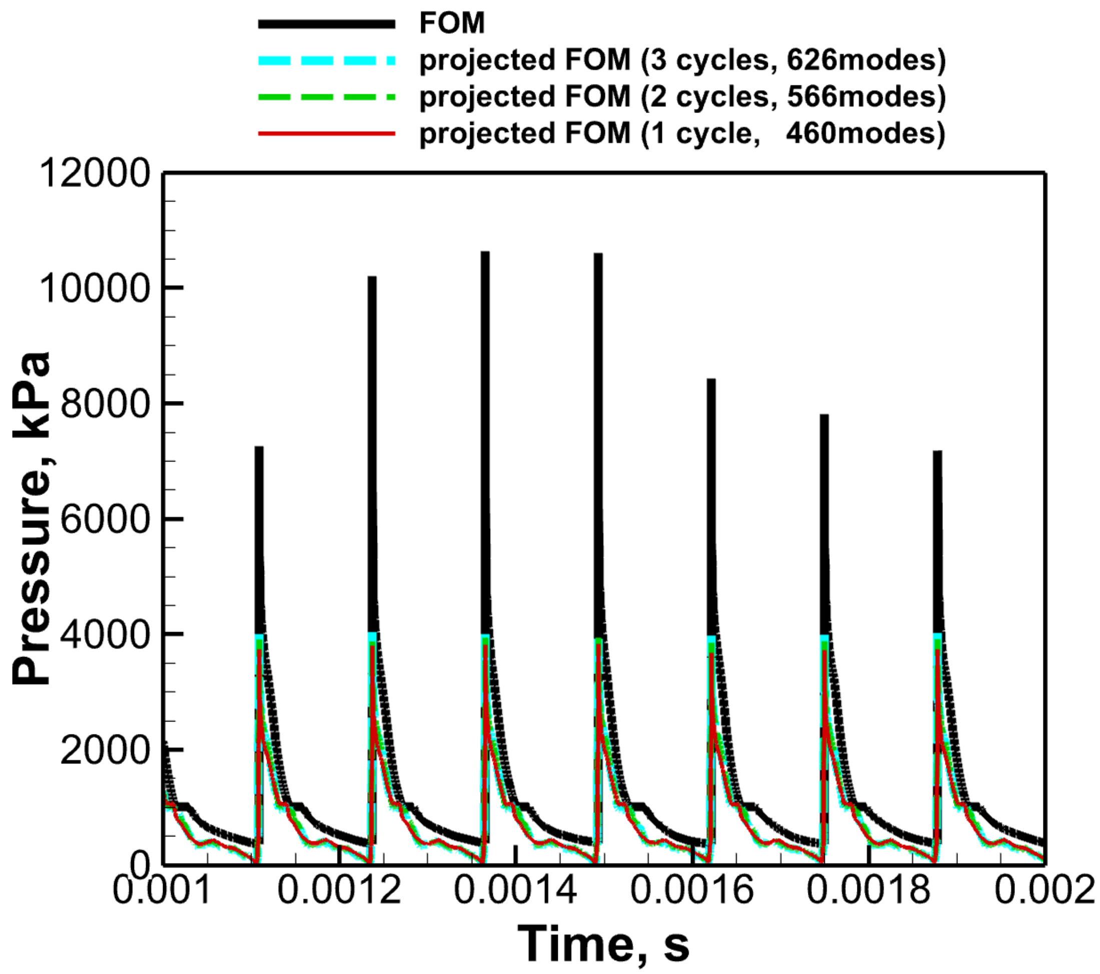
Projected FOM
(1 cycle, 460 modes)

2 cycles, 566 modes

3 cycles, 626 modes

FOM ($\mathbf{q}_p$) *vs* Projected FOM ($\mathbf{q}_{p,ref} + \mathbf{V}\mathbf{V}^T \mathbf{q}_p$): Predicting V_{in} = 200m/s

- POD basis trained using snapshots from V_{in} = 150 m/s
- Static basis exhibits deficiency in parametric predictions

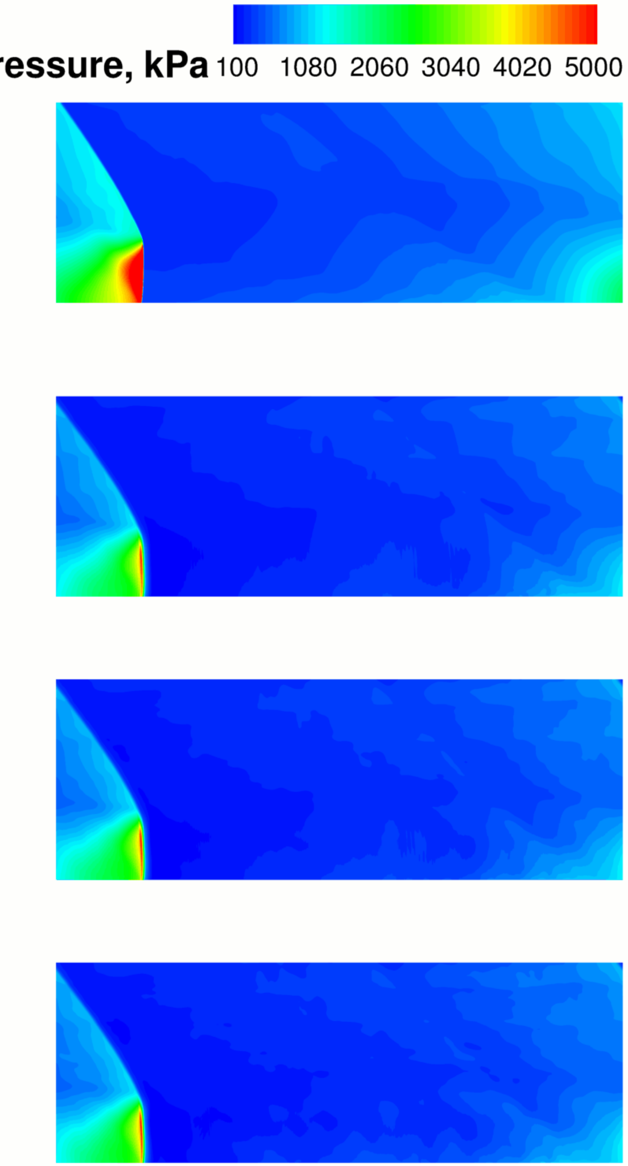


FOM

Projected FOM
(1 cycle, 460 modes)

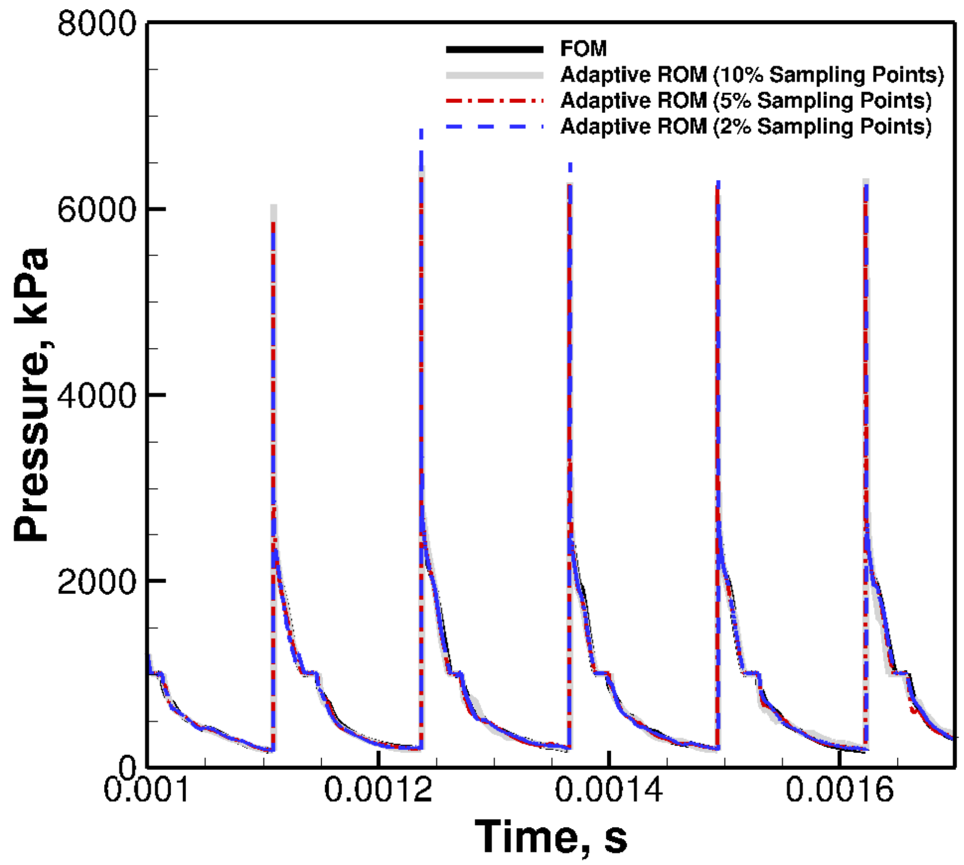
2 cycles, 566 modes

3 cycles, 626 modes



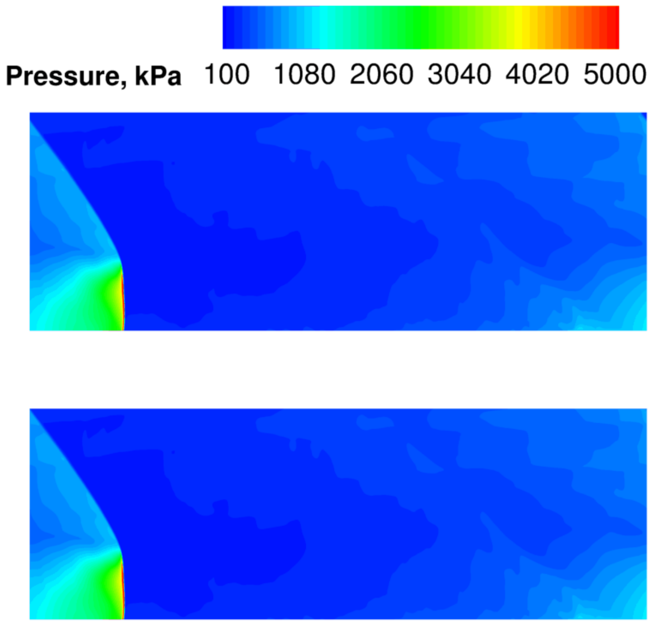
FOM vs Online Adaptive ROM (V_{in} = 150m/s)

- **Initial *offline* training window**: $w_{init} = 10$
- **Sampling points update frequency**: $z_s = 5$
- **Sampling points selection**: the magnitude of pressure gradients $\|\nabla \mathbf{p}\|_2$



FOM

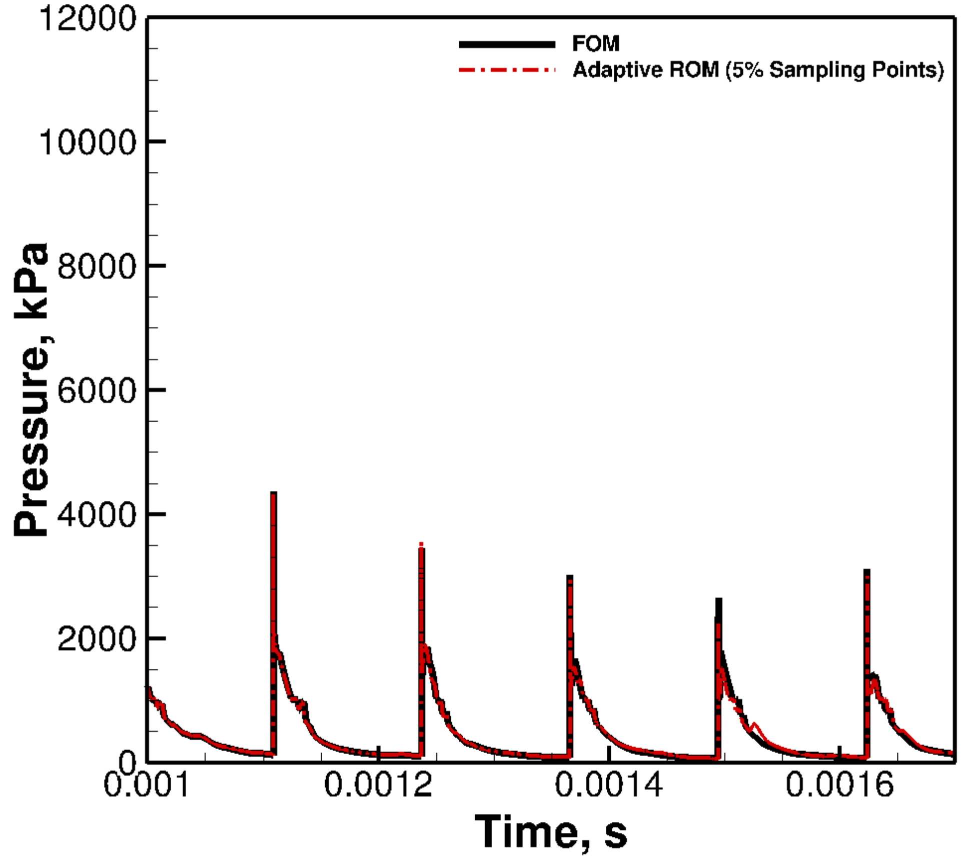
Adaptive ROM
(5% sampling, accurately predict the detonation wave)

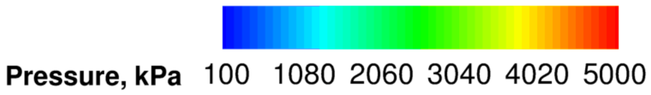


Sampling Points Distributions

FOM vs Online Adaptive ROM (V_{in} = 100m/s)

- **Initial *offline* training window**: $w_{init} = 10$ @ V_{in} = 150m/s
- **Sampling points update frequency**: $z_s = 5$
- **Sampling points selection**: the magnitude of pressure gradients $\|\nabla \mathbf{p}\|_2$





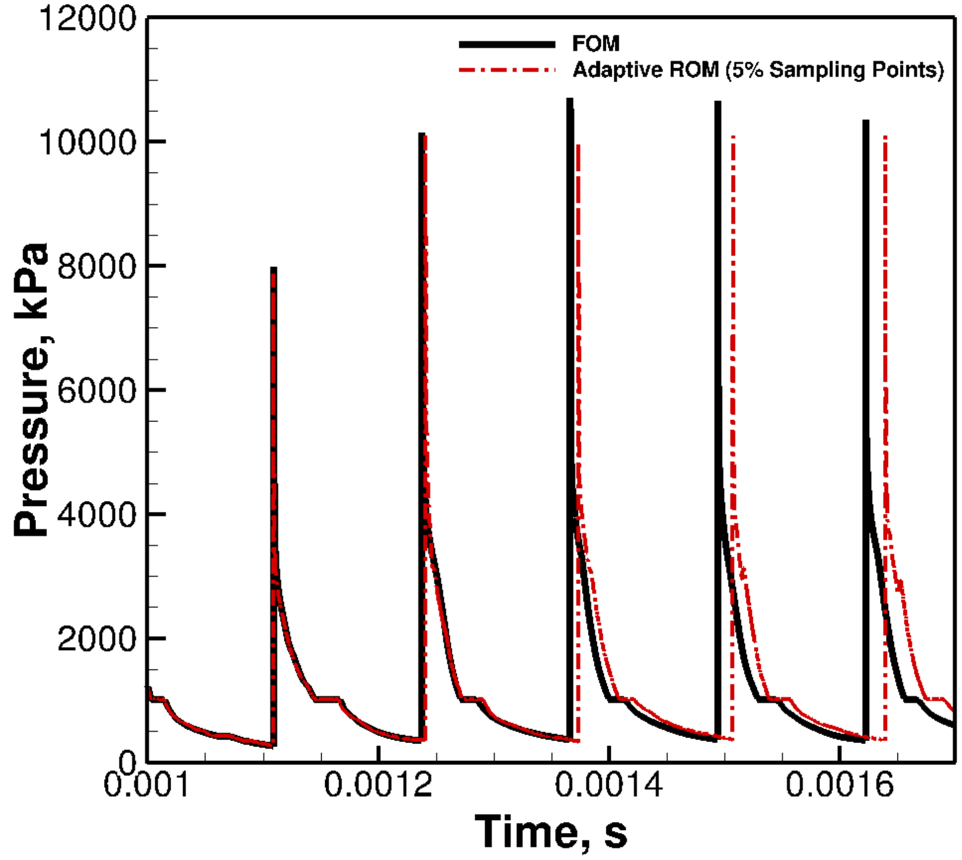
FOM

Adaptive ROM
(5% sampling, accurately predict the transient dynamics)

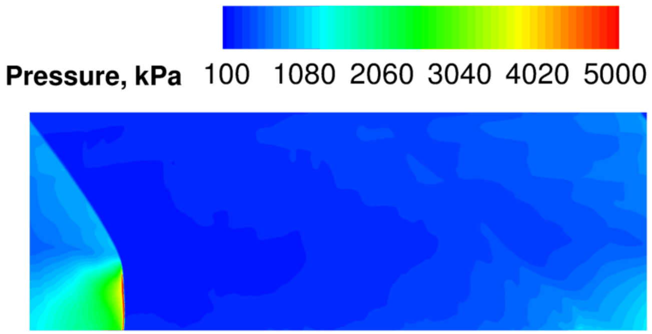
Sampling Points Distributions

FOM vs Online Adaptive ROM (V_{in} = <u>200m/s</u>)

- **Initial *offline* training window**: $w_{init} = 10$ @ V_{in} = 150m/s
- **Sampling points update frequency**: $z_s = 5$
- **Sampling points selection**: the magnitude of pressure gradients $\|\nabla \mathbf{p}\|_2$



FOM

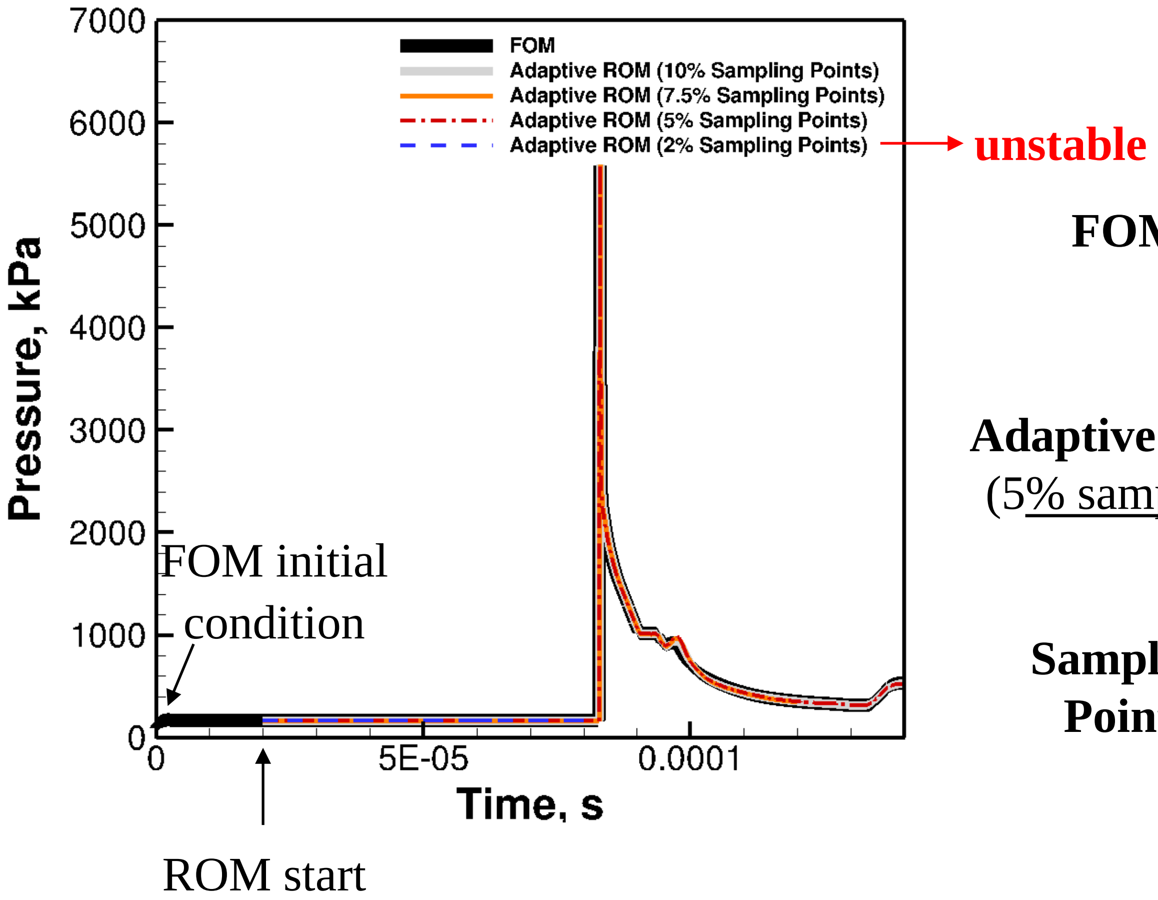


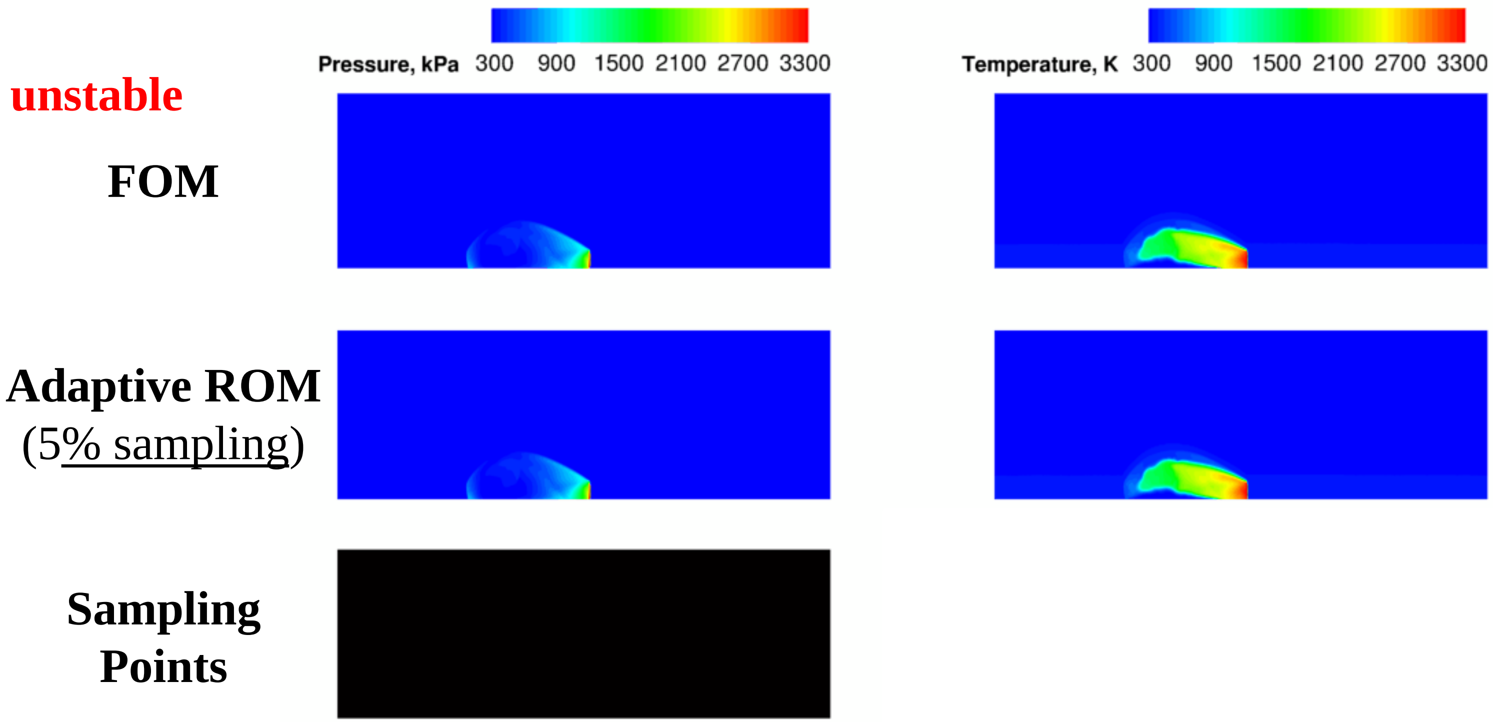
Adaptive ROM
(<u>5% sampling</u>,
predicted lagging
detonation wave)

Sampling Points Distributions

FOM vs Online Adaptive ROM (V_{in} = 150m/s initial transience)

- **Initial *offline* training window**: w_{init} = 10 from 0.02*ms*
- **Sampling points update frequency**: $z_s = 5$
- **Sampling points selection**: the magnitude of pressure gradients $\|\nabla \mathbf{p}\|_2$





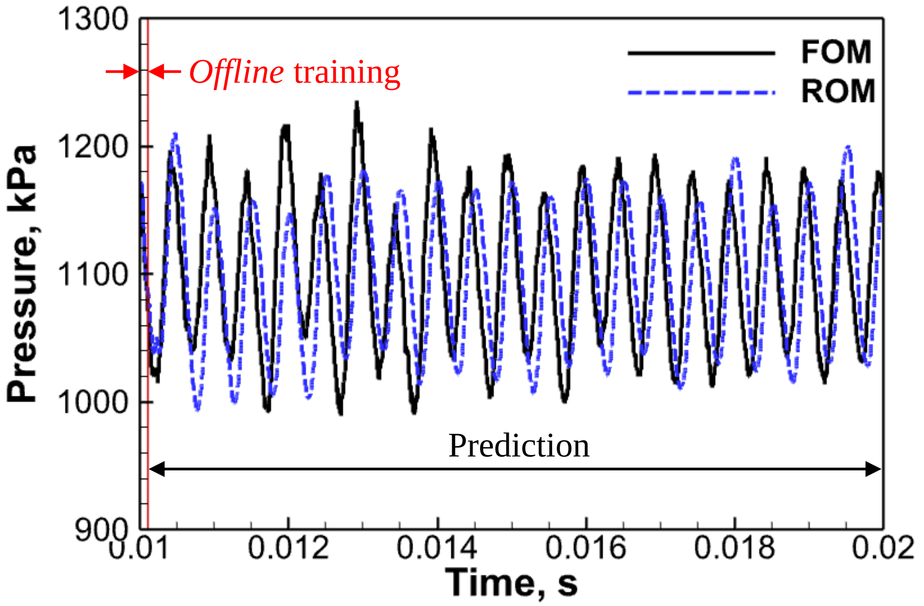
Outline

- **Adaptive ROM Algorithm**
- **Test Case I**: 1D Freely Propagating Laminar Flame
- **Test Case II**: 2D Single-injector Rocket Combustor
- **Test Case III**: 2D Premixed RDE
- **Test Case IV**: 3D Single-injector Rocket Combustor
- **Test Case V**: 2D Single Injector with Variable Recess Length

Test Case IV: 3D Single-injector Rocket Combustor

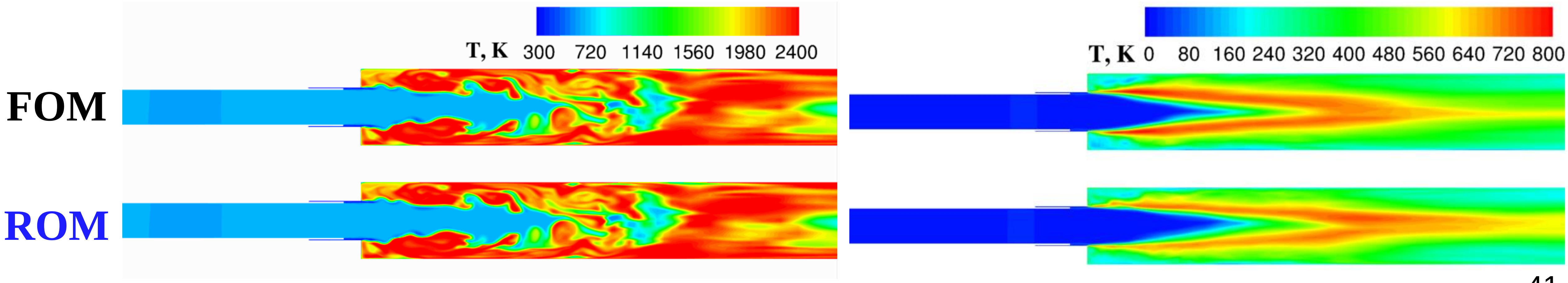
- 0.1*ms offline* training → 10*ms* prediction
- Dimension: 5
- Sampling points update time steps (z_s): 5
- Points sampled: 0.5%

***Local* Pressure Time Trace**



Unsteady Temperature Field

Temperature RMS



Outline

- **Adaptive ROM Algorithm**
- **Test Case I**: 1D Freely Propagating Laminar Flame
- **Test Case II**: 2D Single-injector Rocket Combustor
- **Test Case III**: 2D Premixed RDE
- **Test Case IV**: 3D Single-injector Rocket Combustor
- **Test Case V**: 2D Single Injector with Variable Recess Length

Test Case V: 2D Single Injector with Variable Recess Length

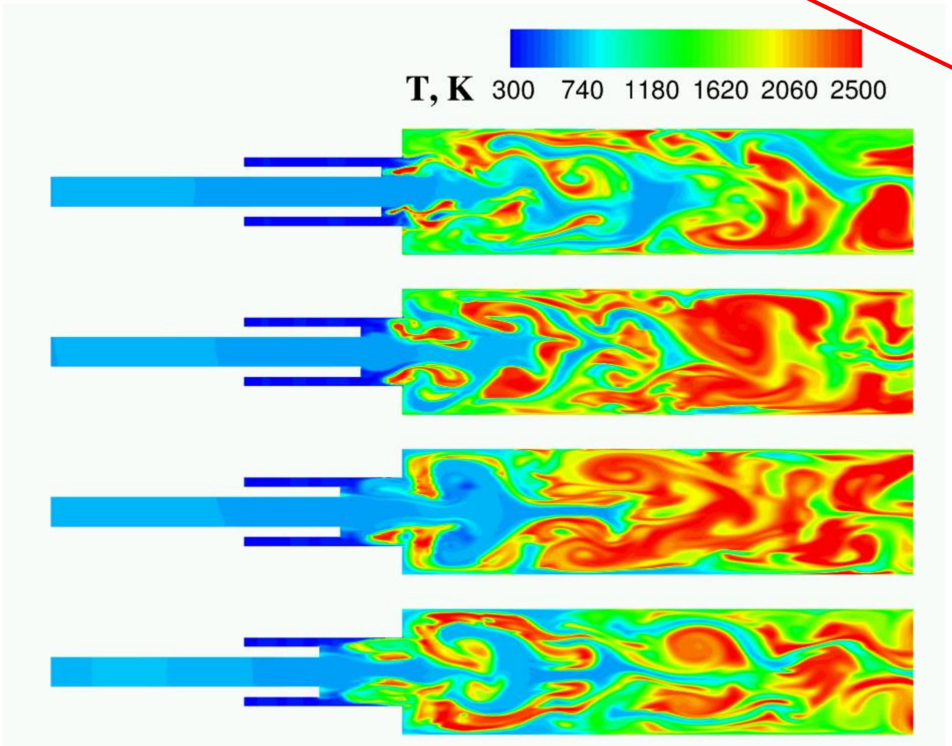
L_{recess} = 5.7, 11.3, 17.0, and 22.7 mm

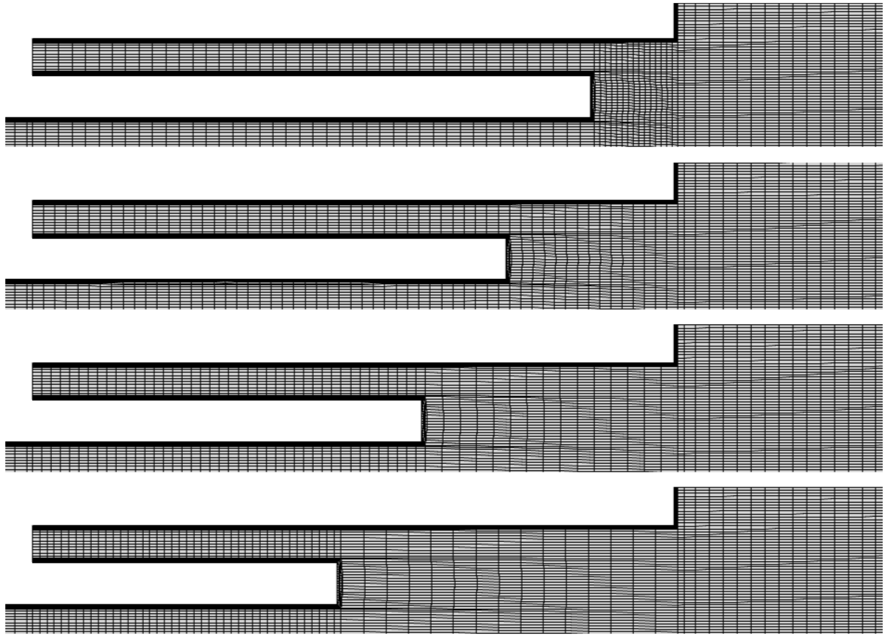
L_{recess} = 5.7 mm

L_{recess} = 11.3 mm

L_{recess} = 17.0 mm

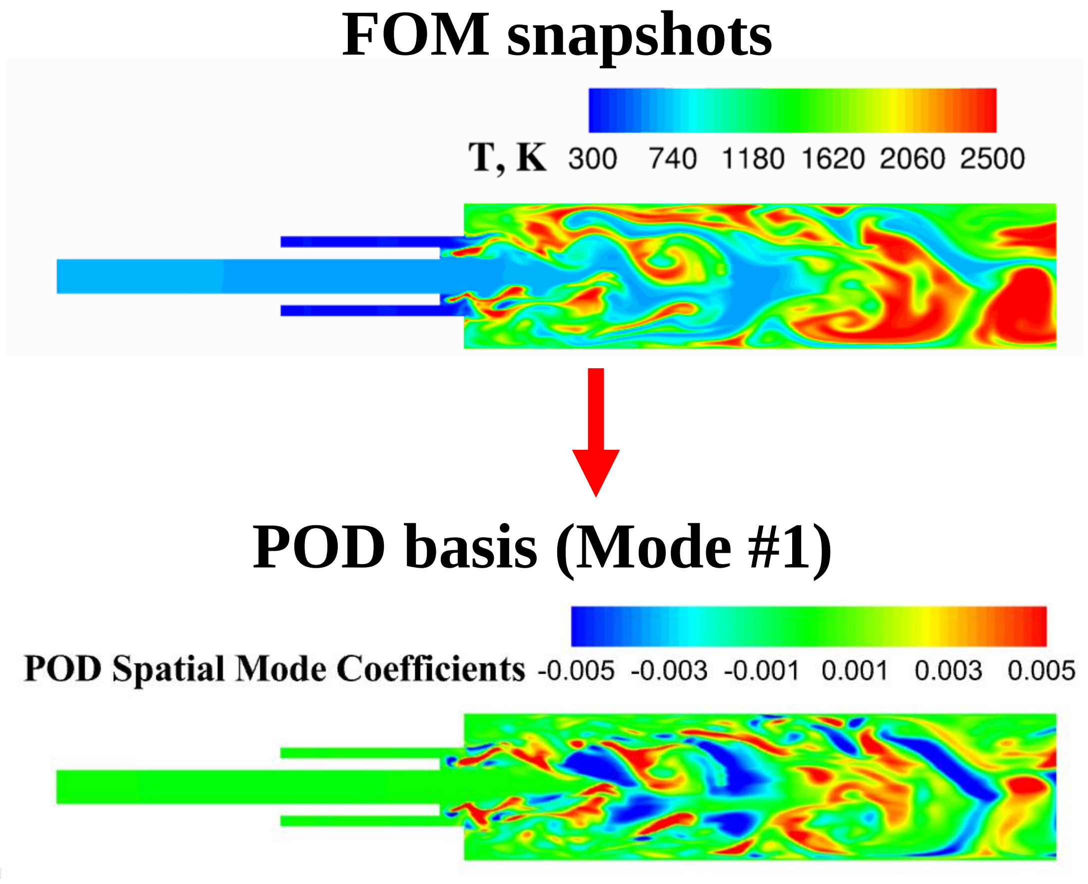
L_{recess} = 22.7 mm

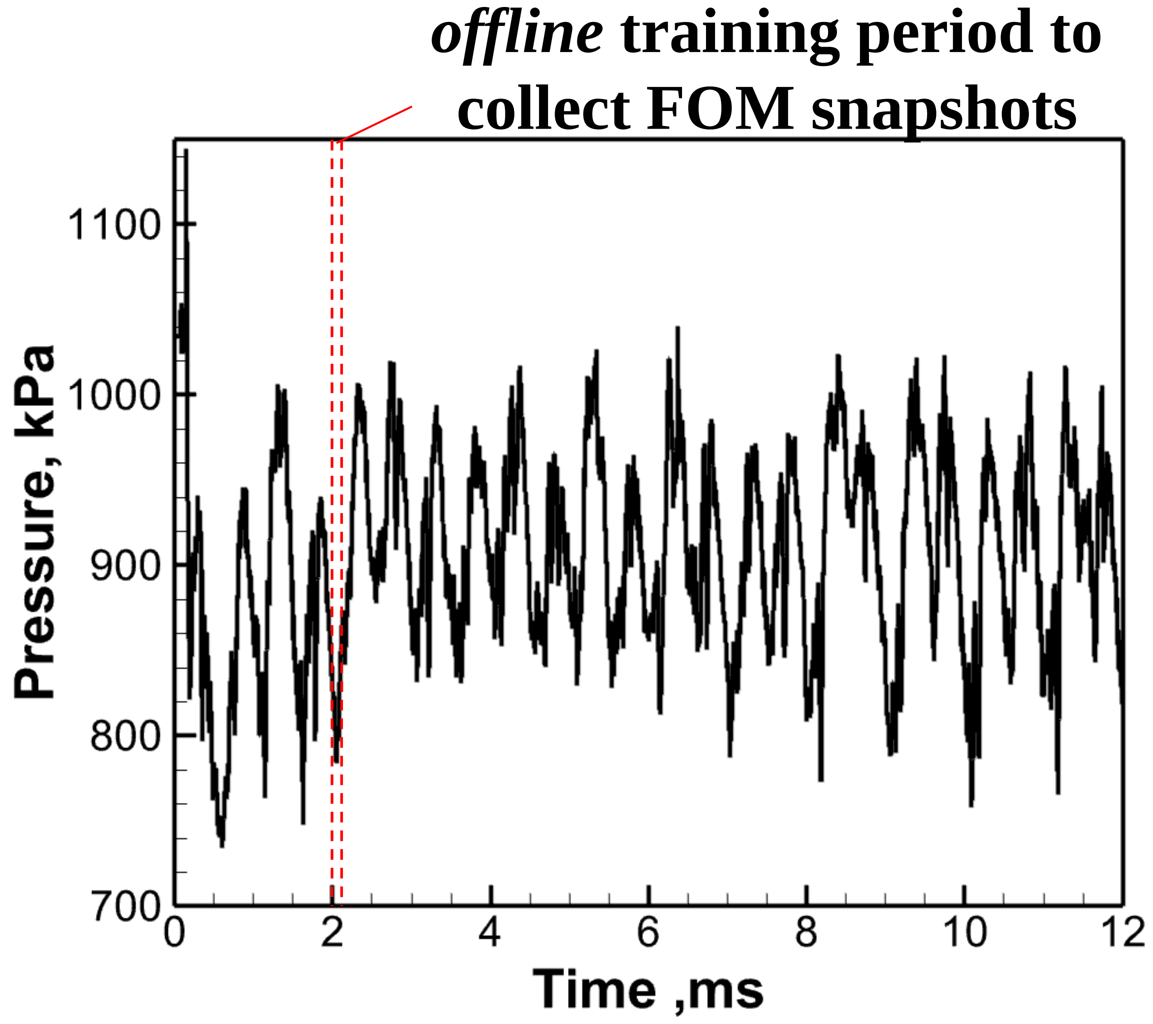




Step 1: offline Training (FOM simulation with L_{recess} = 5.7 mm)

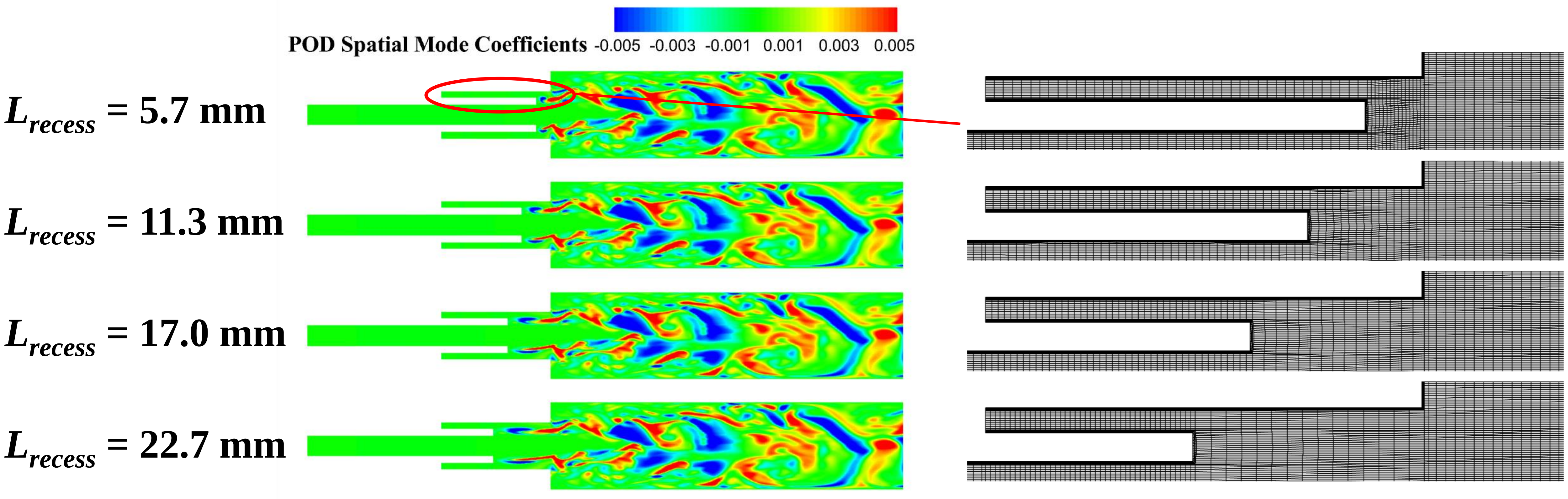
- *Offline* training cost: 2ms transience + 0.1 ms (1000 time steps) training period





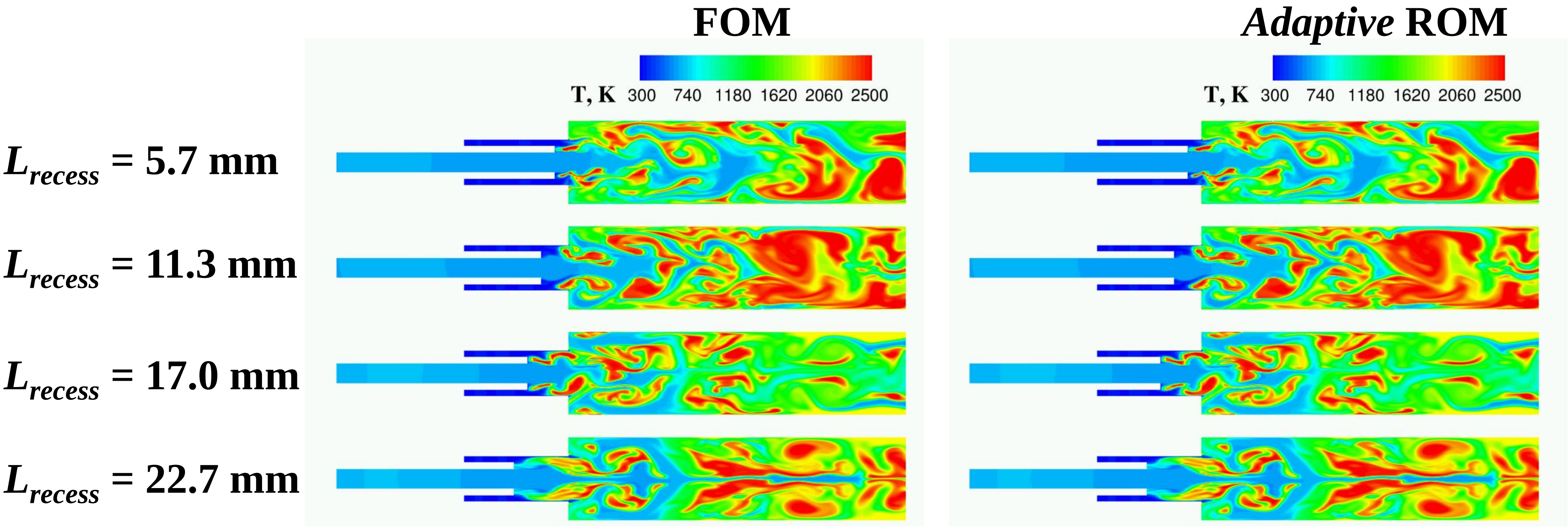
Step 2: map the POD basis from L_{recess} = 5.7 mm to other geometries

- The different geometries share similar mesh topology in the recess region
- The mapped POD basis is used to initialize the adaptive ROM



Step 3: Online prediction with adaptive ROM
– Unsteady Temperature

- The adaptive ROMs are used to predict the dynamics from 2 – 12ms
  - 6 modes + basis updated every time step + 1% sampling points + $z_s = 5$



Summary

An adaptive reduced-order model formulation is developed and implemented to *break* the Kolmogorov barrier and *enable* predictions of turbulent reacting flows

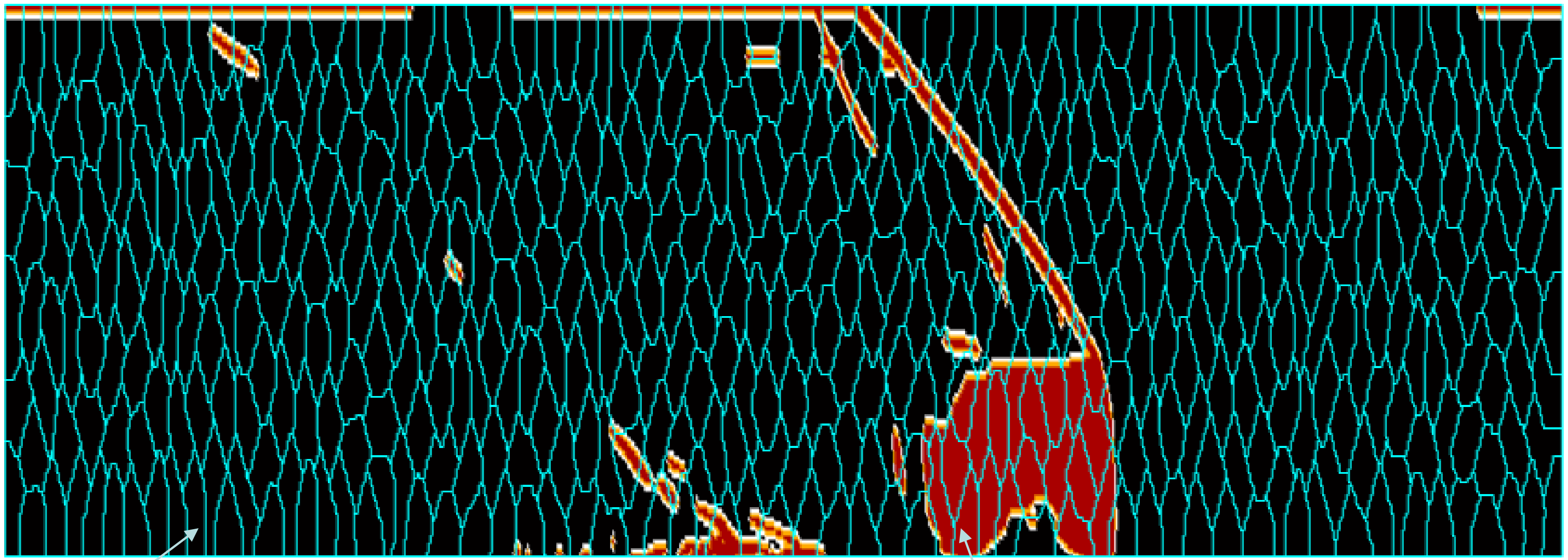
- Incorporating non-local information in ROM adaptation inherently enables predictions of dynamics exhibiting both local and non-local coherence
- The adaptive ROMs have been demonstrated using 1D, 2D, and 3D test problems to provide accurate predictions of future state, transience, and parametric behaviors

Huang and Duraisamy, JCP 2023

Backup

What's next? Improve scalability of adaptive ROM

Adaptive ROM (5% sampling)



**no points are sampled in this partition

**most of the points are sampled in this partition

Implementation of *Hyper-reduced* MP-LSVT

- Portable ROM module could be developed *but* the implementation of hyper-reduction is code dependent

CFD Code

$$S\mathbf{r}\left(\bar{\mathbf{q}}_r^{n+1}\right)$$

Hyper-reduction requires code modification (code dependent)
- Skipping cells/points
- Load balancing

$S\mathbf{r}\left(\bar{\mathbf{q}}_r^{n+1}\right)$

$S\left(\dfrac{\partial \mathbf{r}}{\partial \mathbf{q}_p}\right)^{n+1}$

ROM Module

$$\text{Solve } \left(\bar{\mathbf{W}}^{n+1}\right)^T \mathbf{U}\left(S\mathbf{U}\right)^+ S\mathbf{r}\left(\bar{\mathbf{q}}_r^{n+1}\right) = 0$$

$$\text{where } \bar{\mathbf{W}}^{n+1} = \mathbf{U}\left(S\mathbf{U}\right)^+ S\left(\frac{\partial \mathbf{r}}{\partial \mathbf{q}_p}\right)^{n+1} \mathbf{V}$$

$$\tilde{\mathbf{q}}_p^{n+1} = \mathbf{q}_{p,ref} + \mathbf{V}\bar{\mathbf{q}}_r^{n+1}$$

Current *Hyper-reduction* Implementation in GEMS

- Compatible with overset mesh solver

do i = 1, … , N_{cell}

    if (iblank(i) == 0) then

        cycle

    else if (iblank(i) == 1) then

        evaluate **r**

    else if (iblank(i) $\geq$ 2) then

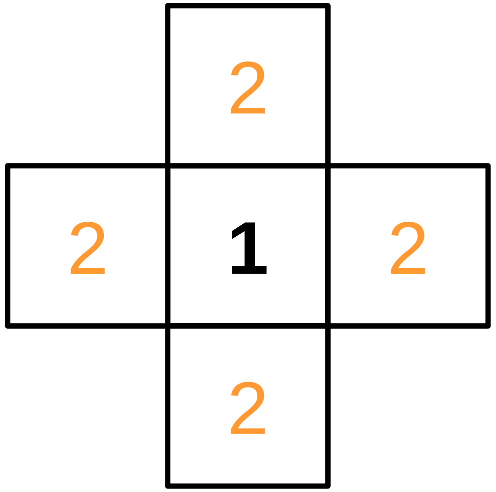
        interpolate state variables

    endif

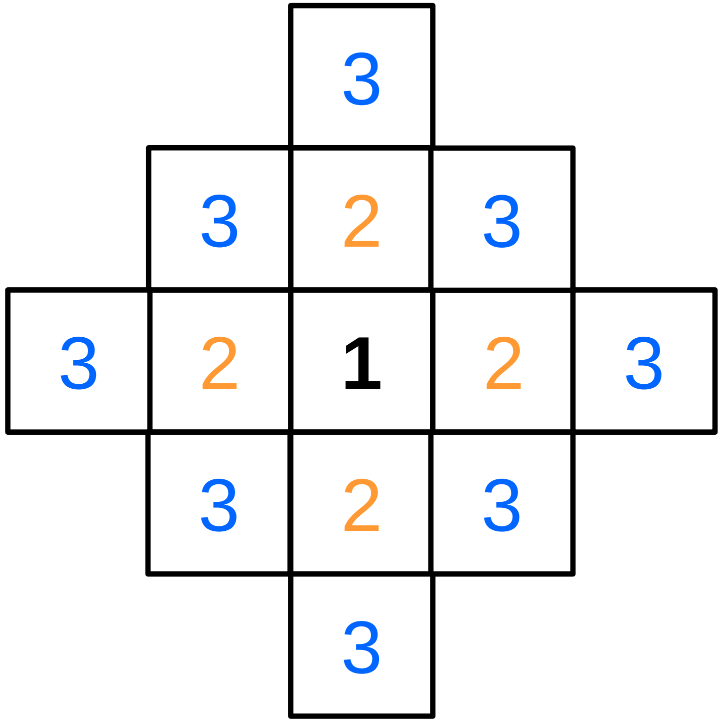
end do

- To provide flux for the sampling point, the neighboring stencil cells also need to be included and interpolated using DEIM

$$\bar{\mathbf{q}}_p^{n+1} \approx \mathbf{U}\left(S\mathbf{U}\right)^{+} S\tilde{\mathbf{q}}_p^{n+1}$$



1st order Flux Scheme



2nd order Flux Scheme

Implementation of *Adaptive Hyper-reduced* MP-LSVT

CFD Code

* Compatible with overset mesh solver

$$S_n \mathbf{r}\left(\hat{\mathbf{q}}_p^{n+1}\right) = 0$$

Evaluating full-state info requires code modification (code dependent)

$$S\mathbf{r}\left(\bar{\mathbf{q}}_r^{n+1}\right)$$

Hyper-reduction requires code modification (code dependent)

- Skipping cells/points
- *Dynamic* load balancing

$\mathbf{F}_S$

Basis & sampling adaptation module

$\mathbf{V}_n$ and S_n

ROM Module

$S_n \mathbf{r}\left(\bar{\mathbf{q}}_r^{n+1}\right)$

$S_n \left(\frac{\partial \mathbf{r}}{\partial \mathbf{q}_p}\right)^{n+1}$

Solve $\left(\bar{\mathbf{W}}^{n+1}\right)^T \mathbf{V}_n \left(S_n \mathbf{V}_n\right)^+ S_n \mathbf{r}\left(\bar{\mathbf{q}}_r^{n+1}\right) = 0$

where $\bar{\mathbf{W}}^{n+1} = \mathbf{V}_n \left(S_n \mathbf{V}_n\right)^+ S_n \left(\frac{\partial \mathbf{r}}{\partial \mathbf{q}_p}\right)^{n+1} \mathbf{V}_n$

$$\tilde{\mathbf{q}}_p^{n+1} = \mathbf{q}_{p,ref} + \mathbf{V}\bar{\mathbf{q}}_r^{n+1}$$