Component-based ROMs (ROM Networks)

Cheng Huang
University of Kansas

Karthik Duraisamy
University of Michigan

Charles Merkle
Purdue University

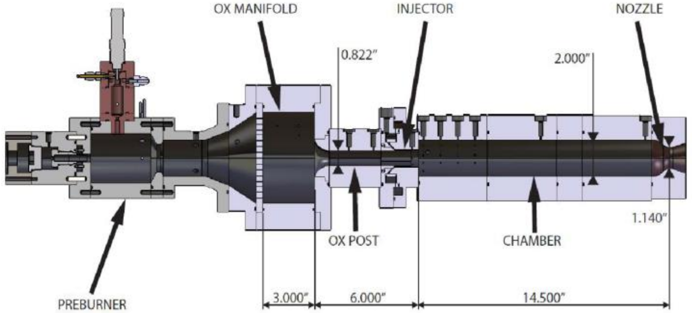
AF/COE Workshop
Thursday, August 31
Dayton, OH

High-fidelity simulations of rocket combustion are expensive

Goal

Given no Full-Order Model (FOM) of the full-scale engine, develop ROM framework to model a class of rocket engines to engage CFD in design

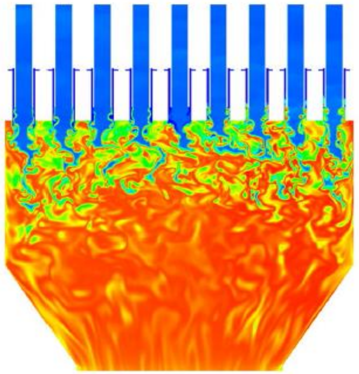
1 element
(1-2 weeks on 1000s of cores)



Purdue HAMSTER experiment
(Harvazinski et al., 2020 AIAA SciTech)

~2M CPU hours affordable!

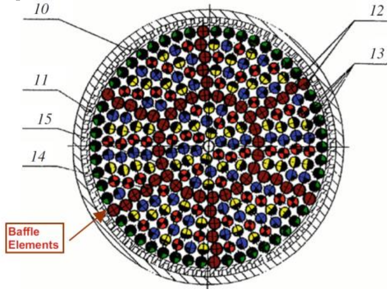
9 elements
(1-2 months on 1000s of cores)



Purdue 9-element transverse chamber
(Harvazinski et al., 2019 AIAA SciTech)

10M CPU hours

100s of elements
(> 10-20 months on > 10,000s of cores)



RD-170 element distribution
(Haeseler and Haidn, 2017)

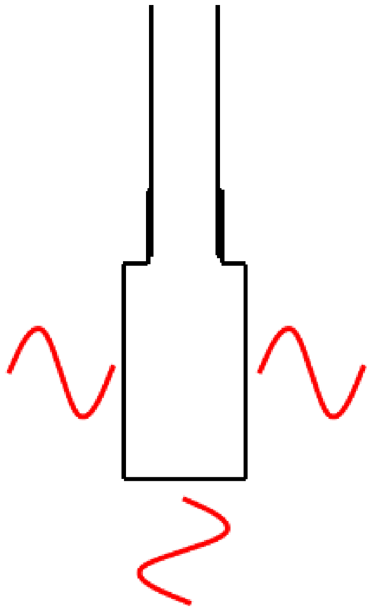
100M CPU hours (still under-resolved!)

Component-based ROM Framework

With <u>no FOM available</u>, develop ROM Framework for <u>a class of systems</u>

- Develop sufficiently rich FOM dataset to construct injector ROM with
  - One (or small number of) injector
  - Boundary perturbations to excite essential dynamics

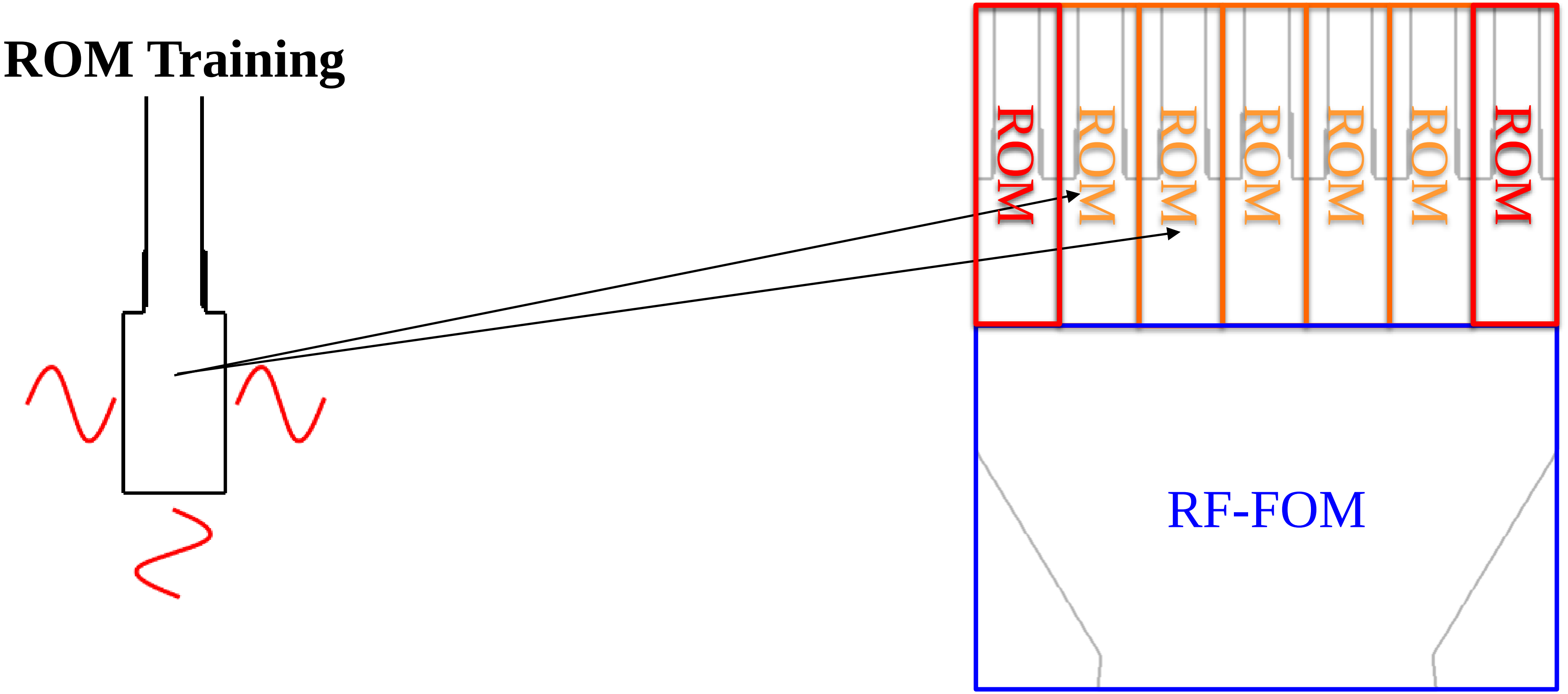
ROM Training



Component-based ROM Framework

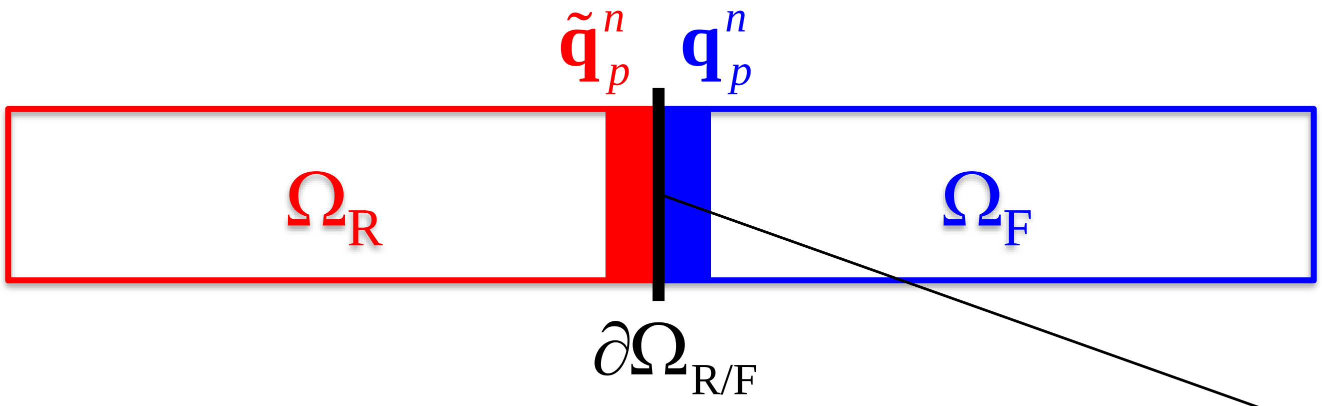
With no FOM available, develop ROM Framework for a class of systems

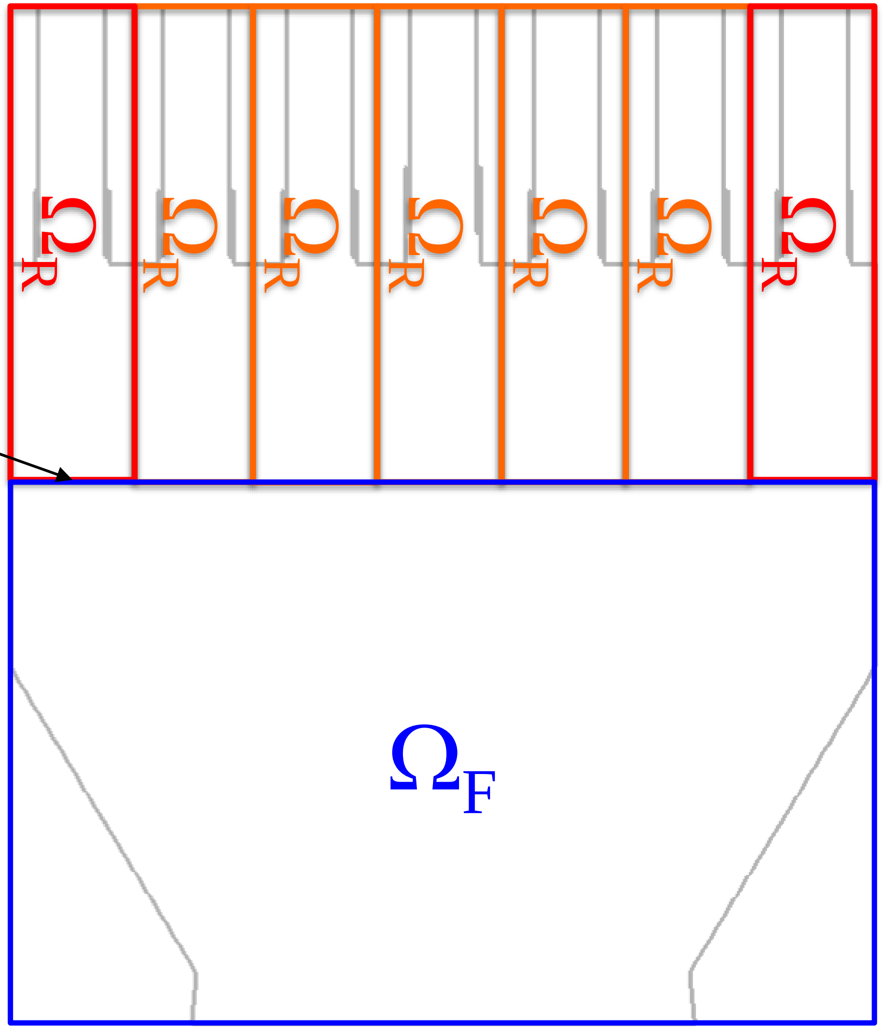
- Develop sufficiently rich FOM dataset to construct injector ROM
- Couple different components to enable the full system modeling
  - Flexible predictions (number of injectors, operating conditions, etc)



Domain Decomposition and Components Coupling

ROM / FOM Coupling





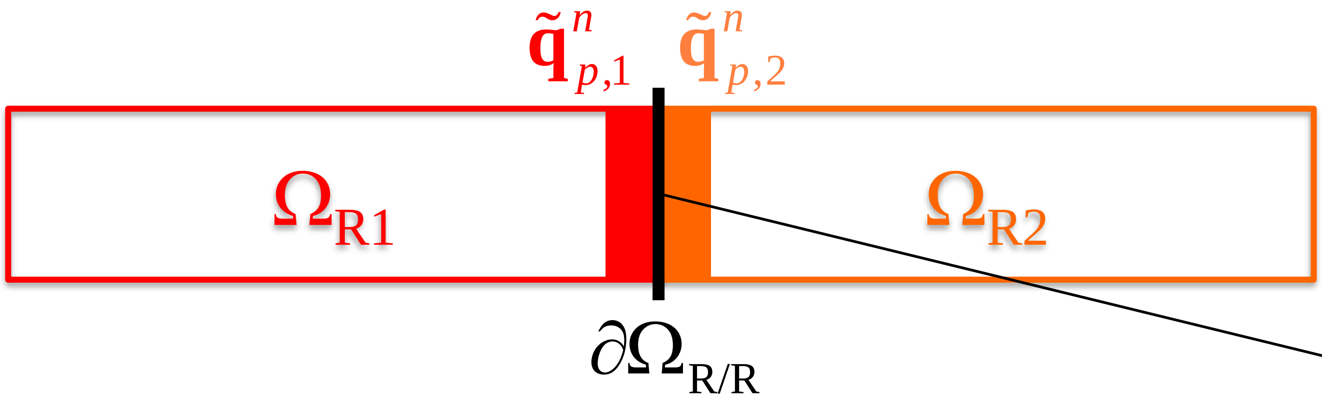
$$\begin{cases} \left(\mathbf{W}^n\right)^T \mathbf{r}\left(\mathbf{q}_r^n\right) = 0 & \text{on } \Omega_R \\ \mathbf{u}\left(\tilde{\mathbf{q}}_p^n, \mathbf{q}_p^n\right) = \mathbf{u}^n & \text{on } \partial\Omega_{R/F} \end{cases} \qquad \begin{cases} \mathbf{r}\left(\mathbf{q}_p^n\right) = 0 & \text{on } \Omega_F \\ \mathbf{u}\left(\mathbf{q}_p^n, \tilde{\mathbf{q}}_p^n\right) = \mathbf{u}^n & \text{on } \partial\Omega_{R/F} \end{cases}$$

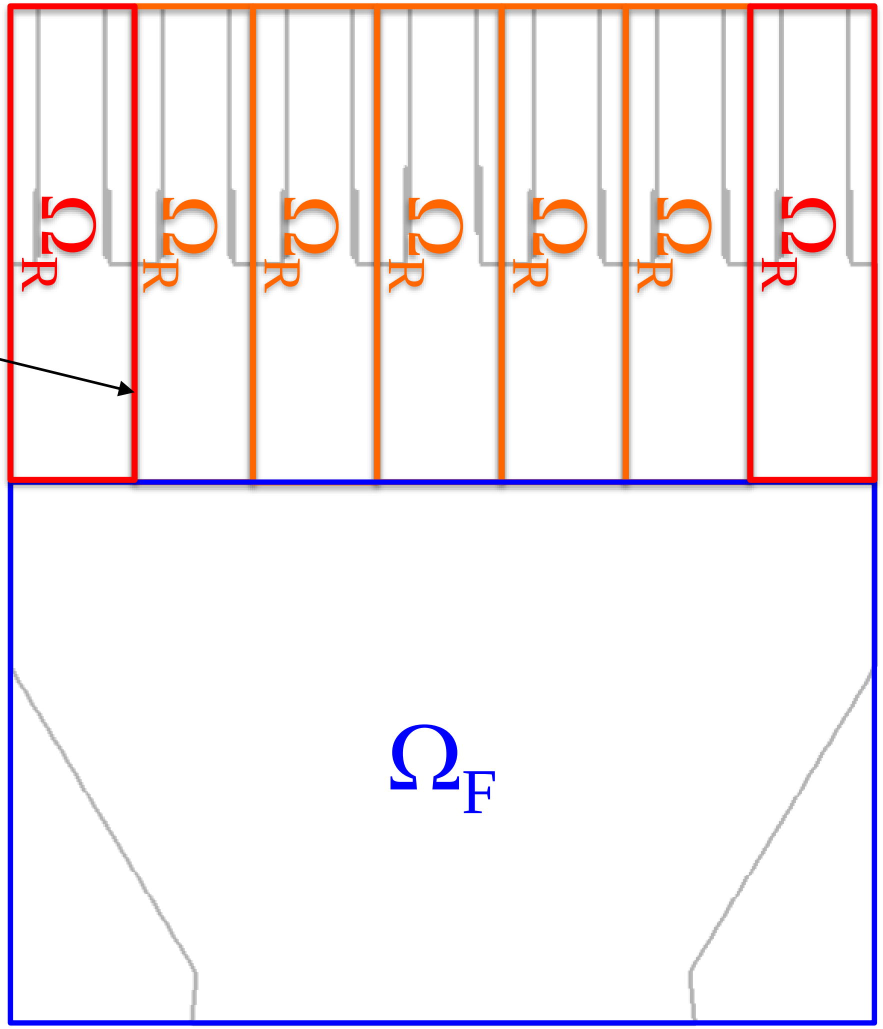
where $\tilde{\mathbf{q}}_p = \mathbf{q}_{p,ref} + \mathbf{V}\mathbf{q}_r$

$$\text{e.g. } \mathbf{u}\left(\tilde{\mathbf{q}}_p^n, \mathbf{q}_p^n\right) = \frac{1}{2}\left[\mathbf{u}_{left}\left(\tilde{\mathbf{q}}_p^n\right) + \mathbf{u}_{right}\left(\mathbf{q}_p^n\right)\right] - \frac{1}{2}\left|\frac{\partial \mathbf{u}}{\partial \mathbf{q}}\right|\left(\mathbf{q}_p^n - \tilde{\mathbf{q}}_p^n\right)$$

Domain Decomposition and Components Coupling

ROM / ROM Coupling





$$\begin{cases} \left(\mathbf{W}_1^n\right)^T \mathbf{r}\left(\mathbf{q}_{r,1}^n\right) = 0 & \text{on } \Omega_{\mathrm{R1}} \\ \mathbf{u}\left(\tilde{\mathbf{q}}_{p,1}^n, \tilde{\mathbf{q}}_{p,2}^n\right) = \mathbf{u}^n & \text{on } \partial\Omega_{\mathrm{R/R}} \end{cases} \qquad \begin{cases} \left(\mathbf{W}_2^n\right)^T \mathbf{r}\left(\mathbf{q}_{r,2}^n\right) = 0 & \text{on } \Omega_{\mathrm{R2}} \\ \mathbf{u}\left(\tilde{\mathbf{q}}_{p,2}^n, \tilde{\mathbf{q}}_{p,1}^n\right) = \mathbf{u}^n & \text{on } \partial\Omega_{\mathrm{R/R}} \end{cases}$$

where $\tilde{\mathbf{q}}_{p,1}^n = \mathbf{q}_{p,1,ref} + \mathbf{V}_1\mathbf{q}_{r,1}$ and $\tilde{\mathbf{q}}_{p,2} = \mathbf{q}_{p,2,ref} + \mathbf{V}_2\mathbf{q}_{r,2}$

e.g. $\mathbf{u}\left(\tilde{\mathbf{q}}_{p,1}^n, \tilde{\mathbf{q}}_{p,2}^n\right) = \frac{1}{2}\left[\mathbf{u}_{left}\left(\tilde{\mathbf{q}}_{p,1}^n\right) + \mathbf{u}_{right}\left(\tilde{\mathbf{q}}_{p,2}^n\right)\right] - \frac{1}{2}\left|\frac{\partial \mathbf{u}}{\partial \mathbf{q}}\right|\left(\tilde{\mathbf{q}}_{p,2}^n - \tilde{\mathbf{q}}_{p,1}^n\right)$

**** Information is exchanged between components in full states**

- Consistent formulation as FOM/FOM interfacing for parallel implementation
    - ➢ Easy implementation in existing (multi-domain/paralell) solvers/codes
- Complex interface dynamics (e.g., flow separation or reverse flow) inherently accounted <u>if ROM 'sees' such physics</u>
- Compatible with different types of ROMs (both non-intrusive and intrusive)
- Making ROM training and coupling relatively independent of each other
- Easily extendable to structured/unstructured mesh coupling

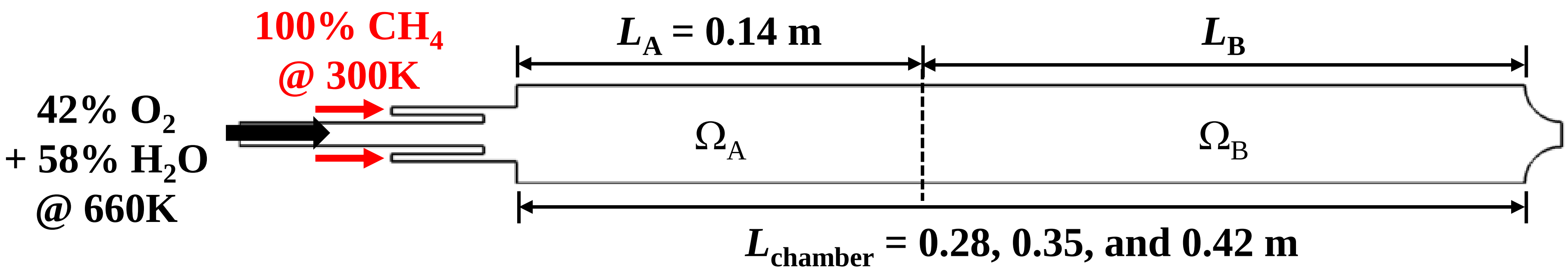
Outline

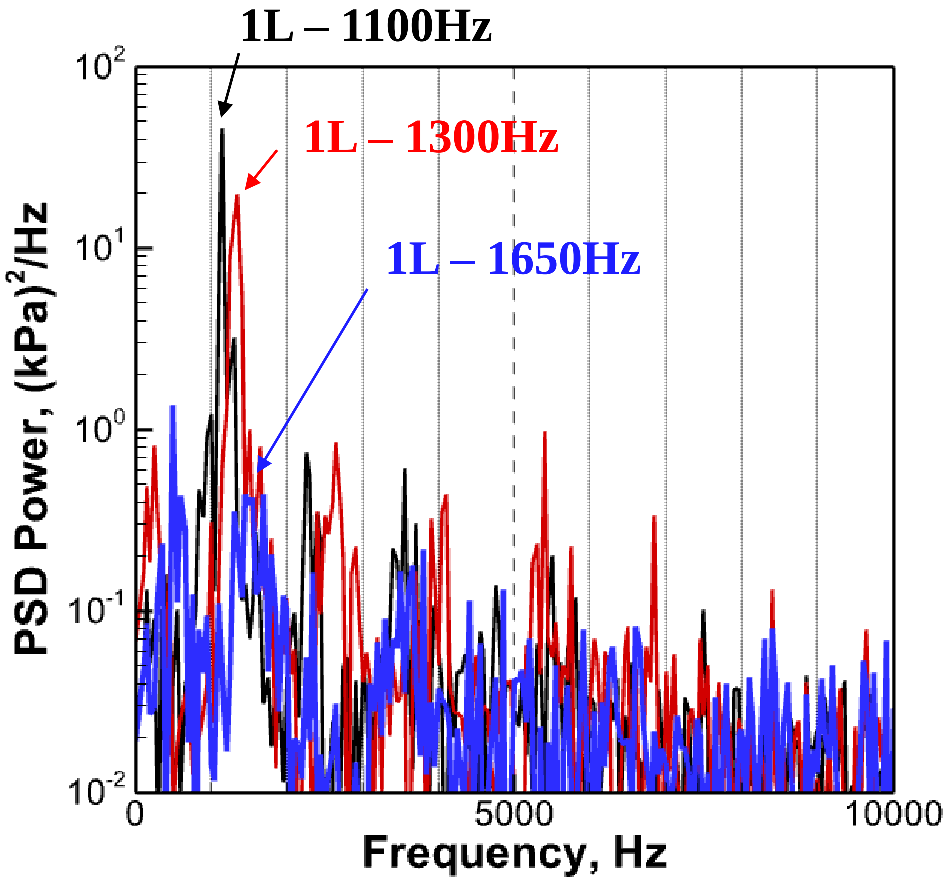
- **Test Case I**: 2D Single-Injector Rocket Combustors with Variable Geometries

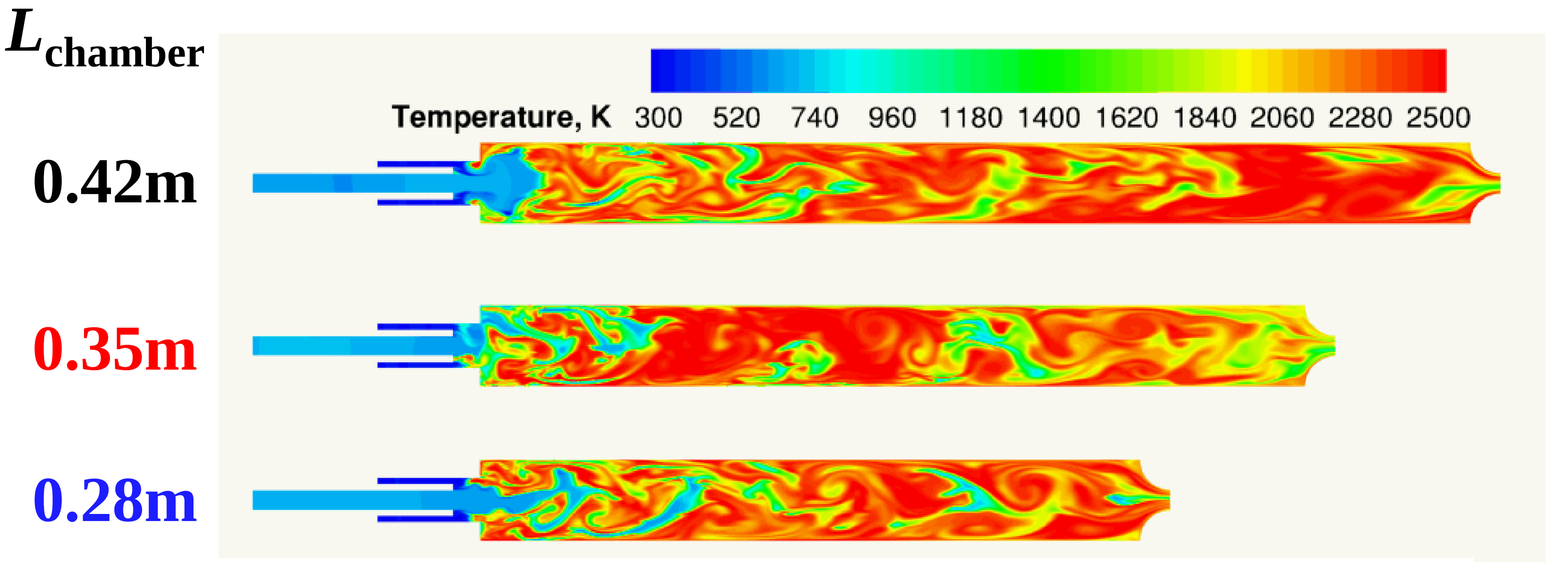
- **Test Case II**: 2D Single Injector with Variable Mass Flow Rates

- **Test Case III**: 2D Multi-Injector Rocket Combustors with Variable Geometries

Test Case I: 2D Single-Injector Rocket Combustors with Variable Geometries

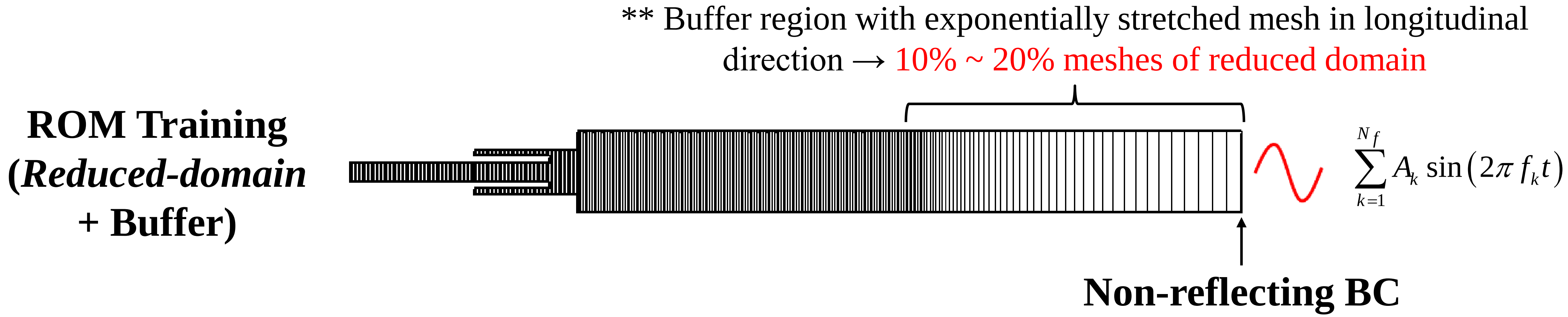






ROM Training: *Reduced-domain* + Buffer

- **A buffer region is added downstream to the reduced-domain**

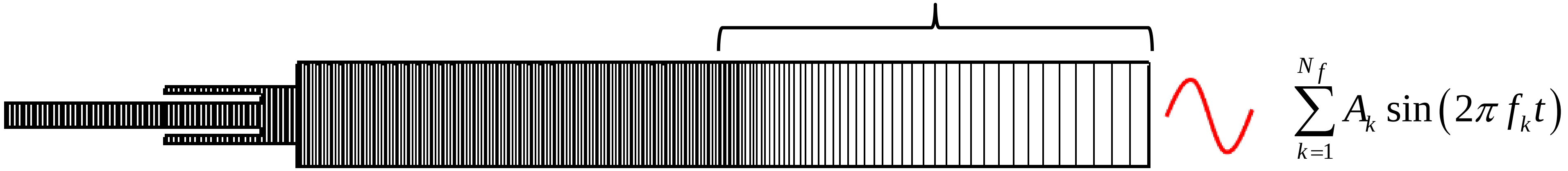


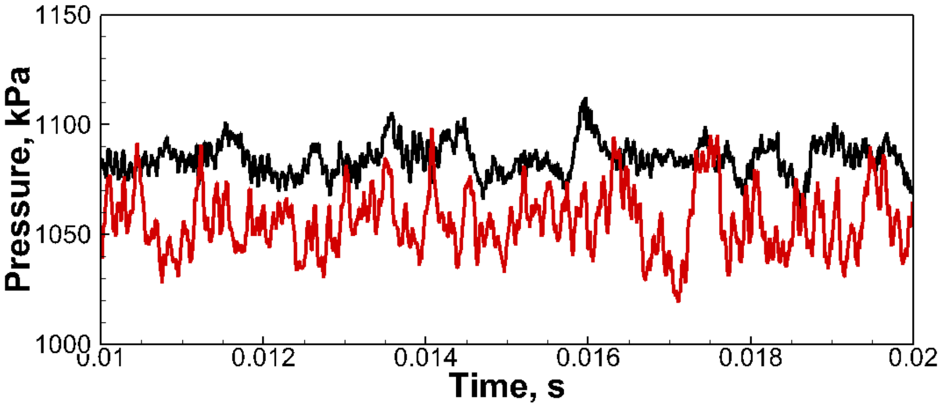
ROM Training: *Reduced-domain* + Buffer

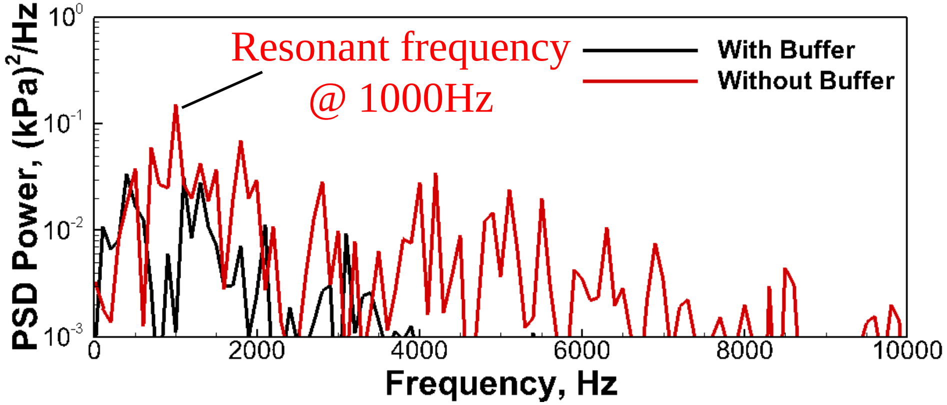
- **A buffer region is added downstream to the reduced-domain**
  - To attenuate acoustic wave reflections from the downstream boundary

** Buffer region with exponentially stretched mesh in longitudinal direction → 10% ~ 20% meshes of reduced domain

ROM Training (*Reduced-domain* + Buffer)





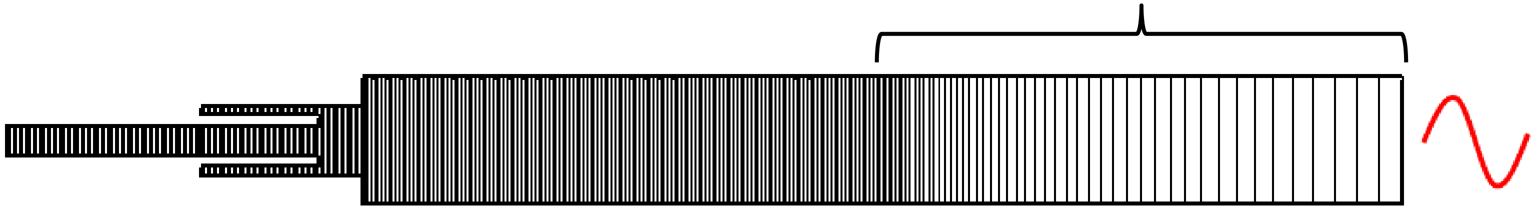


ROM Training: *Reduced-domain* + Buffer

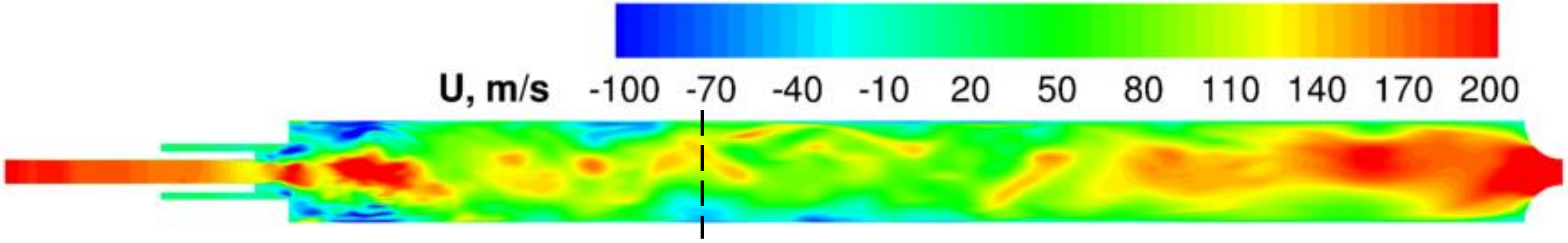
- **A buffer region is added downstream to the reduced-domain**
  - To attenuate acoustic wave reflections from the downstream boundary
  - To account for reverse flow dynamics at the interface

** Buffer region with exponentially stretched mesh in longitudinal direction → 10% ~ 20% meshes of reduced domain

ROM Training (*Reduced-domain* + Buffer)



$$\sum_{k=1}^{N_f} A_k \sin\left(2\pi f_k t\right)$$

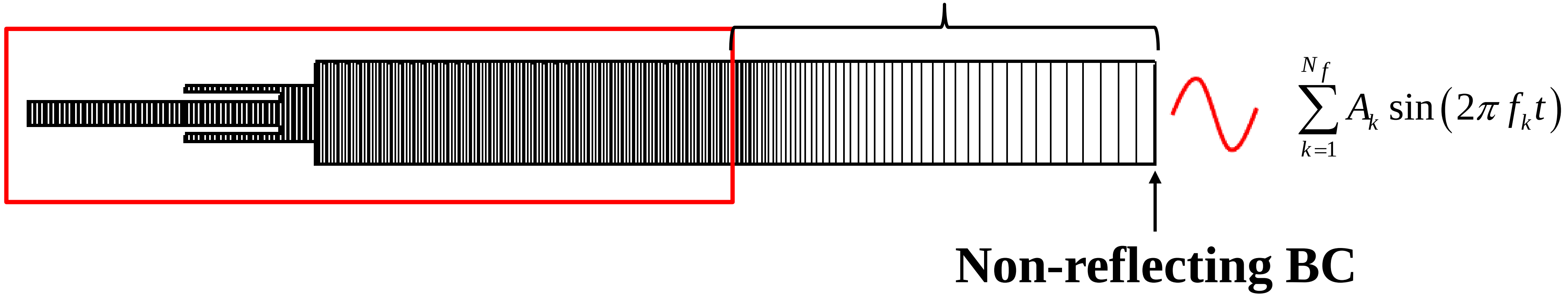


ROM Training: *Reduced-domain* + Buffer

- **A buffer region is added downstream to the reduced-domain**
  - To attenuate acoustic wave reflections from the downstream boundary
  - To account for reverse flow dynamics at the interface
  - Only upstream portion used to construct the ROM (<u>buffer region excluded</u>)

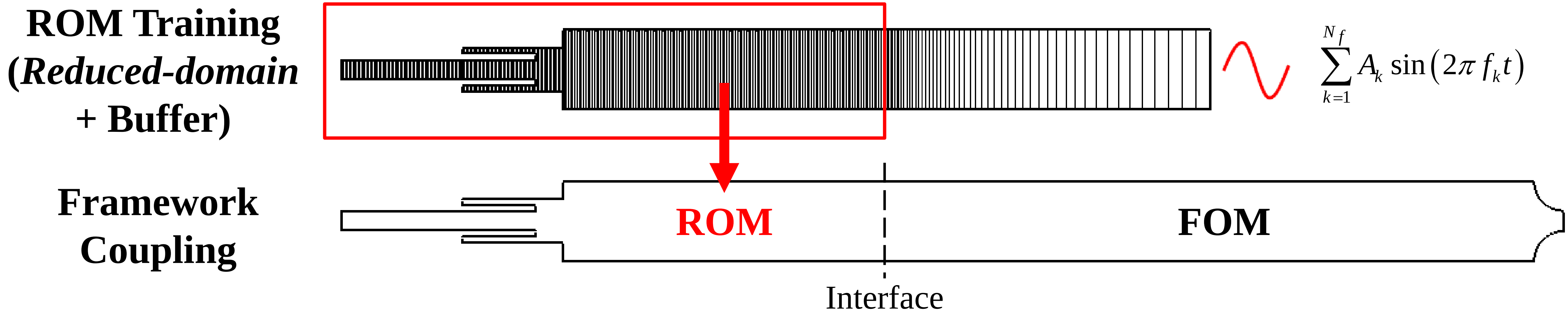
** Buffer region with exponentially stretched mesh in longitudinal direction → 10% ~ 20% meshes of reduced domain

ROM Training (*Reduced-domain* + Buffer)



Component-based ROM Framework

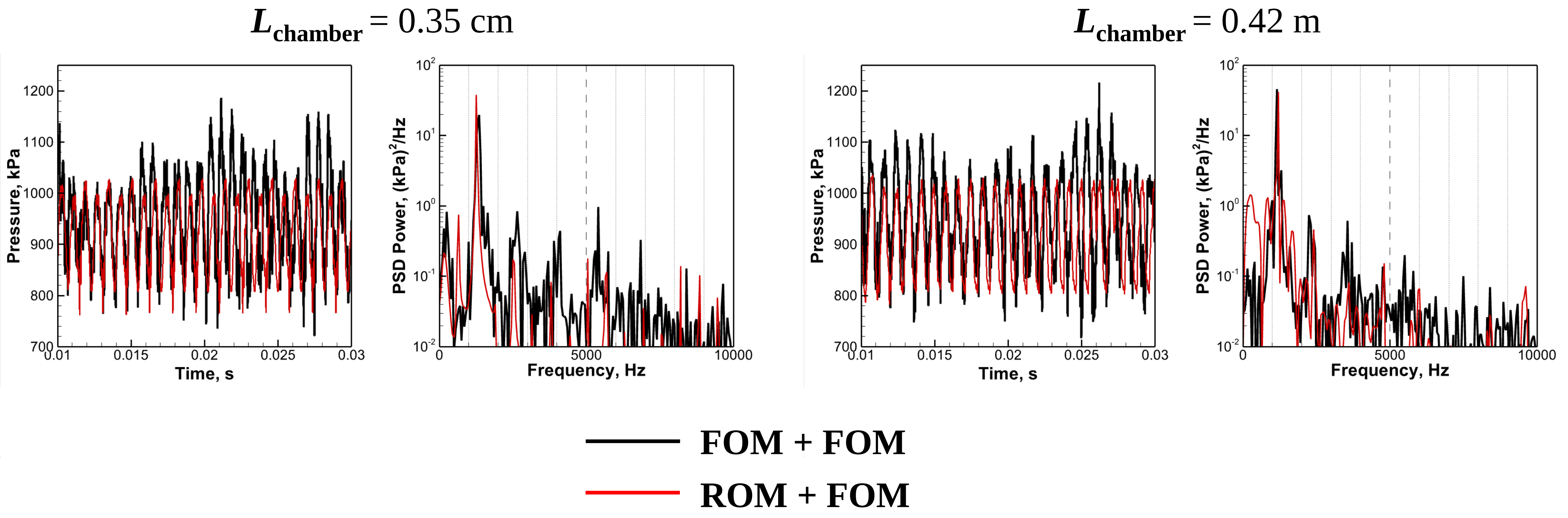
- **ROM training**: downstream perturbations to mimic full-domain acoustics
  - $N_f = 2$ with $f_1 = 1100$Hz (1L for $L_{chamber} = 0.42$ m) and $A_1 = 0.05$
    $f_2 = 1300$Hz (1L for $L_{chamber} = 0.35$ m) and $A_2 = 0.05$
- **ROM/FOM Coupling:** flux exchange at interface
  - FOM is used in the downstream to evaluate the framework performance



** no hyper-reduction in ROM

Comparisons of *Local* Pressure Signals – Unstable Configurations

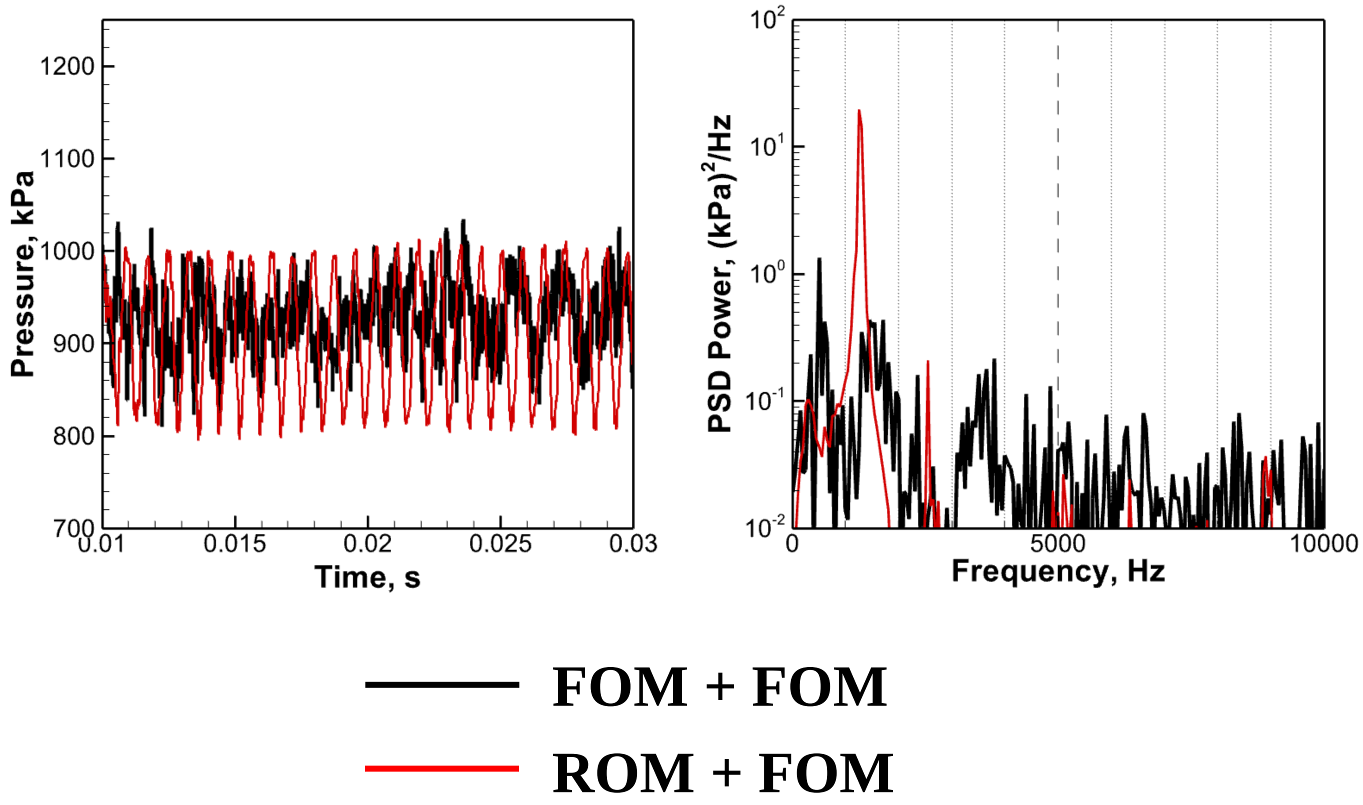
- ROM (trained using ~ **1.6** ms) + FOM to predict **20** ms full-domain dynamics
- ROM+FOM framework reasonably captures the instability characteristics of the longer chamber lengths



Comparisons of *Local* Pressure Signals – Stable Configuration

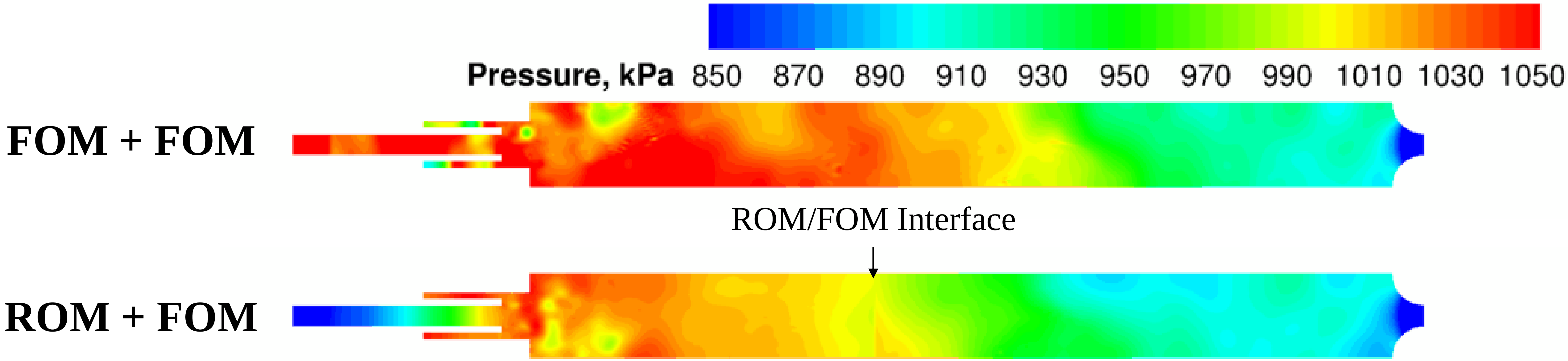
- ROM (trained using ~ **1.6** ms) + FOM to <u>predict</u> **20** ms full-domain dynamics
- ROM+FOM framework reasonably captures the oscillation amplitude but <u>fails to predict the dominant frequencies</u> of the shorter chamber length

$\boldsymbol{L}_{\textbf{chamber}} = 0.28$ m

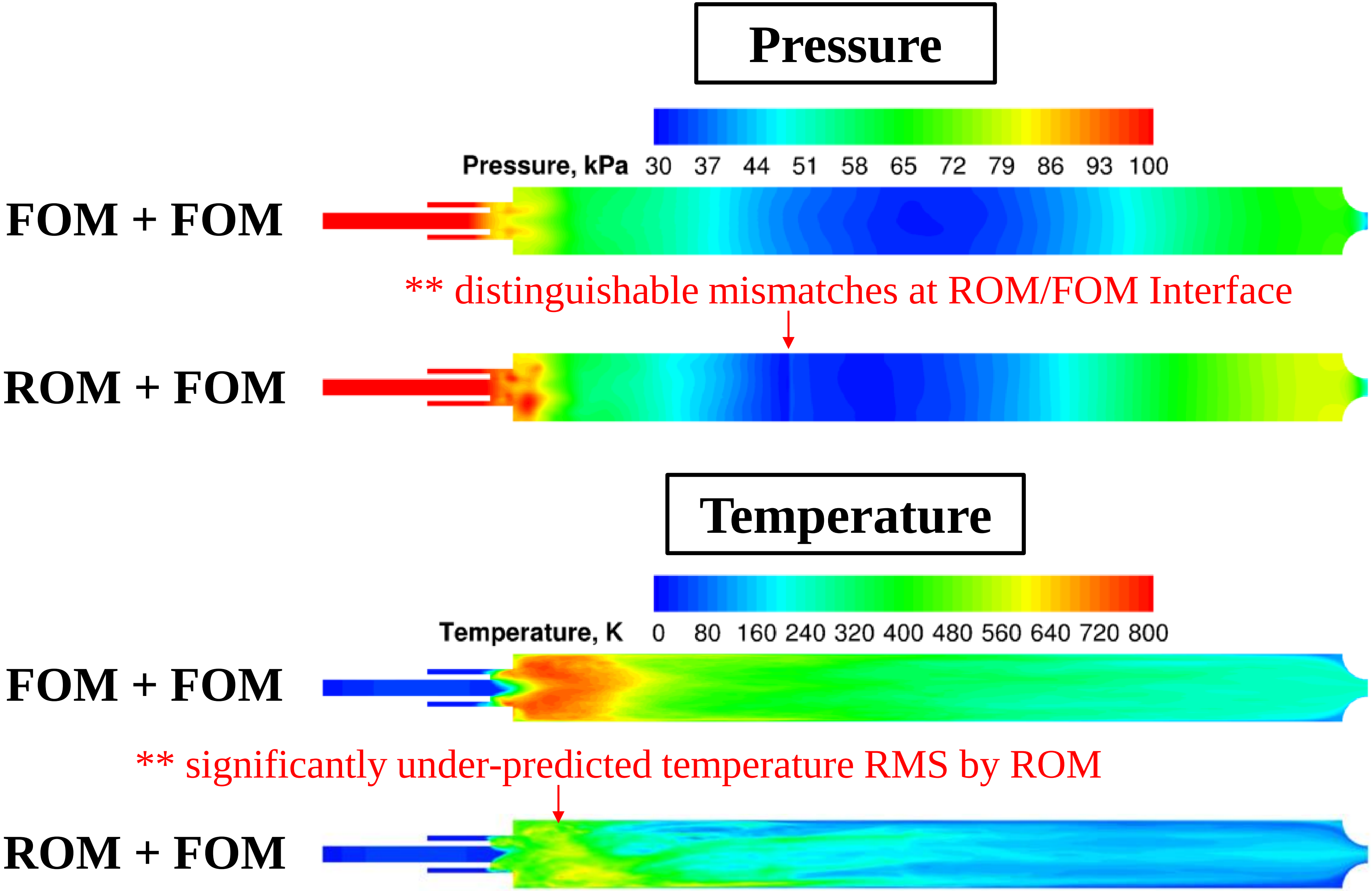


Comparisons of Unsteady Pressure Fields – $L_{chamber} = 0.42$ m

- **Distinguishable mismatches at ROM/FOM interface**
  - The upstream ROM is not able to predict the correct pressure dynamics with the information (e.g., upstream running characteristics) from the downstream FOM

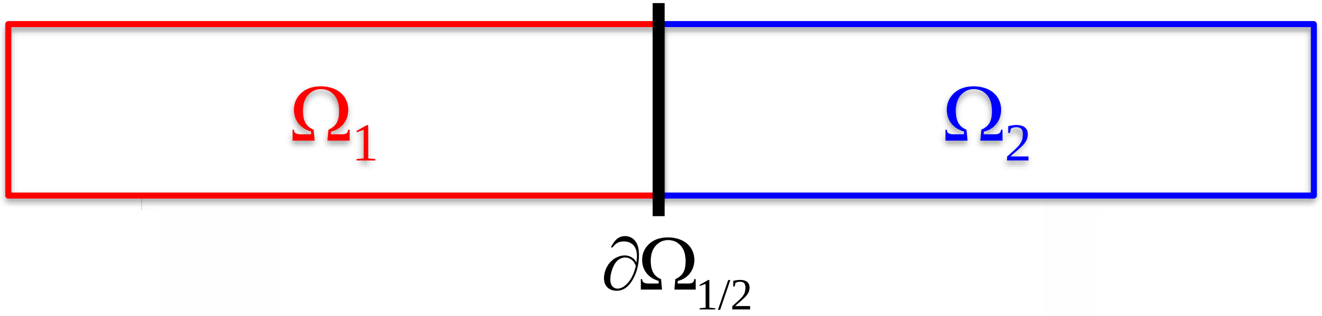


Comparisons of RMS Fields – $L_{chamber}$ = 0.42 m



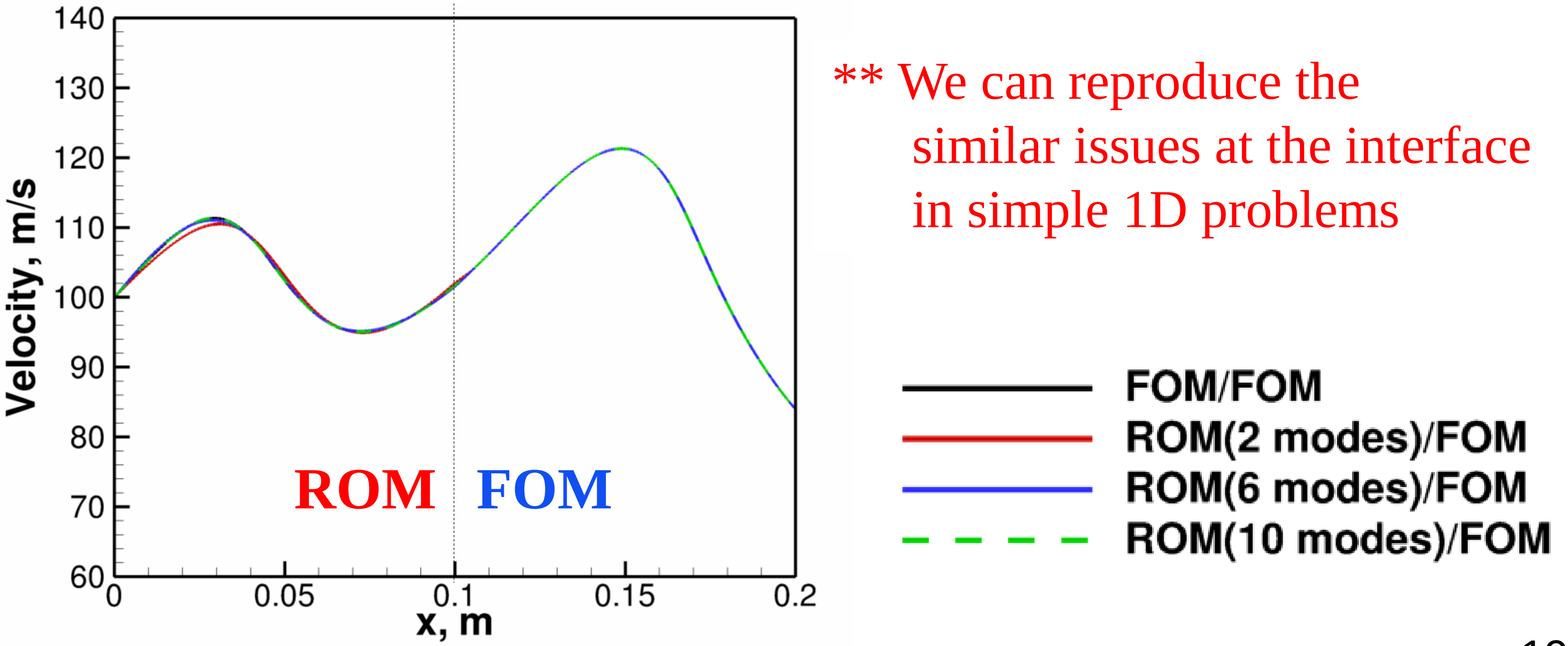
Domain Decomposition (ROM/FOM)

$$\mathbf{W}^T \mathbf{r}\left(\tilde{\mathbf{q}}_{p,1}, \mathbf{q}_{p,2,\partial\Omega_{1/2}}\right) = 0 \text{ on } \Omega_1 \quad \mathbf{r}\left(\mathbf{q}_{p,2}, \tilde{\mathbf{q}}_{p,1,\partial\Omega_{1/2}}\right) = 0 \text{ on } \Omega_2$$



Two types of remedies

- *Improve ROM predictive capabilities*
  - More POD modes + more training dataset
  - Adaptive ROM
- *Impose constraints at the interface* to match the conditions between two domains



** We can reproduce the similar issues at the interface in simple 1D problems

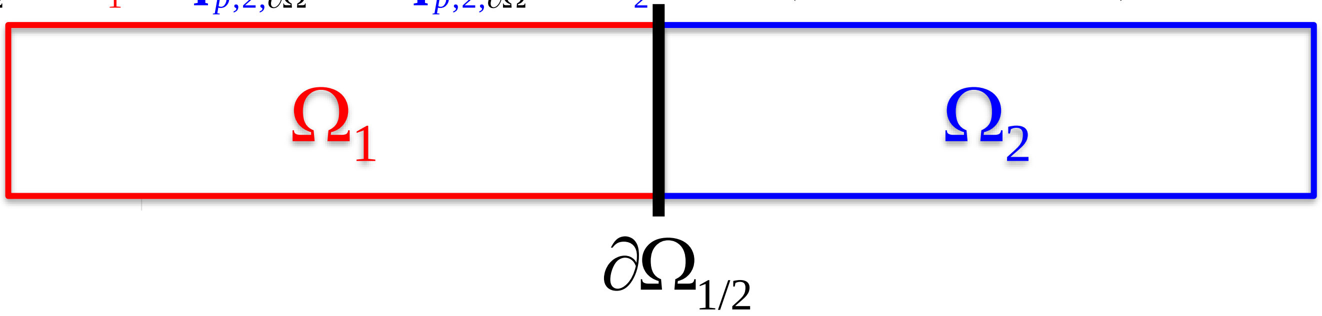
Domain Decomposition *with* Constraints (ROM/FOM)

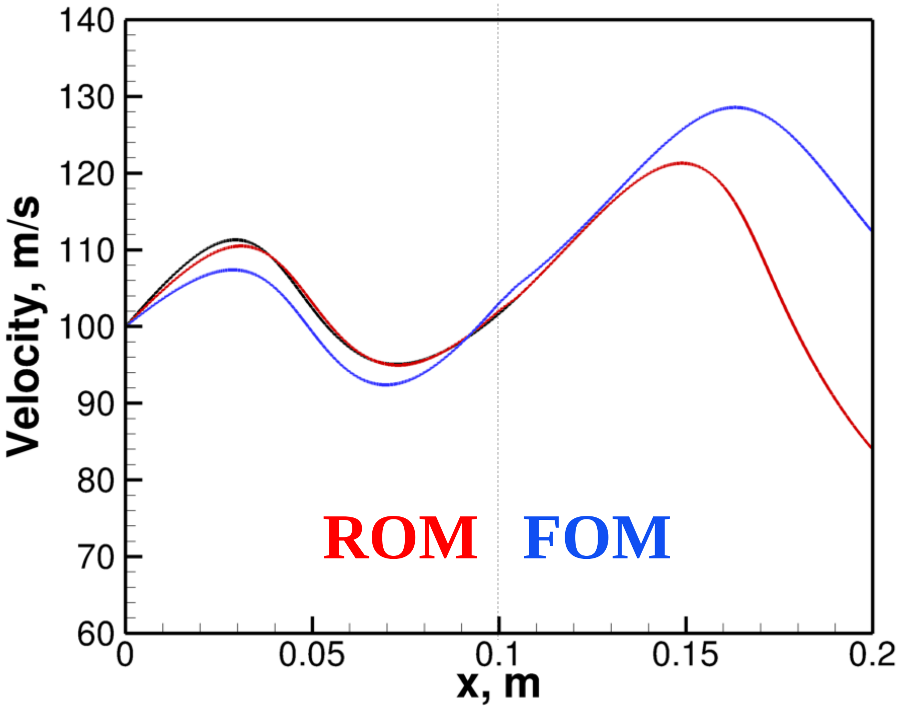
$$\arg\min_{\tilde{\mathbf{q}}_{p,1}} \left\| \mathbf{r}\left(\tilde{\mathbf{q}}_{p,1}, \mathbf{q}_{p,2,\partial\Omega_{1/2}}\right) \right\|_2^2 \text{ on } \Omega_1$$

$$s.t.\ \tilde{\mathbf{q}}_{p,1,\partial\Omega} + \nabla\tilde{\mathbf{q}}_{p,1,\partial\Omega} \cdot d\Omega_1 = \mathbf{q}_{p,2,\partial\Omega} + \nabla\mathbf{q}_{p,2,\partial\Omega} \cdot d\Omega_2$$

$$\mathbf{r}\left(\mathbf{q}_{p,2}, \tilde{\mathbf{q}}_{p,1,\partial\Omega_{1/2}}\right) = 0 \text{ on } \Omega_2$$

- ❖ Farhat et al, 2000
- ❖ Lucia et al., 2003
- ❖ Baiges et al, 2013
- ❖ Hoang et al., 2020
- ❖ Xiao et al, 2021





Imposing constraints in ROM eliminates the interface mismatch

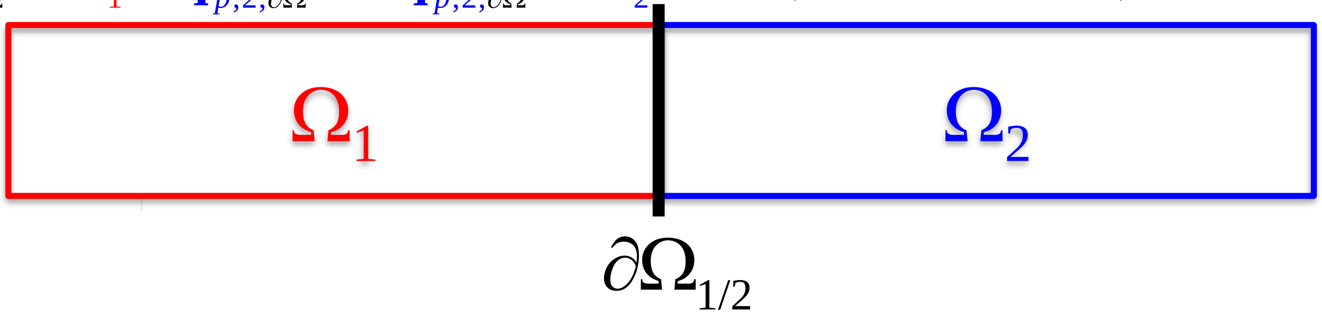
Domain Decomposition *with* Constraints (ROM/FOM)

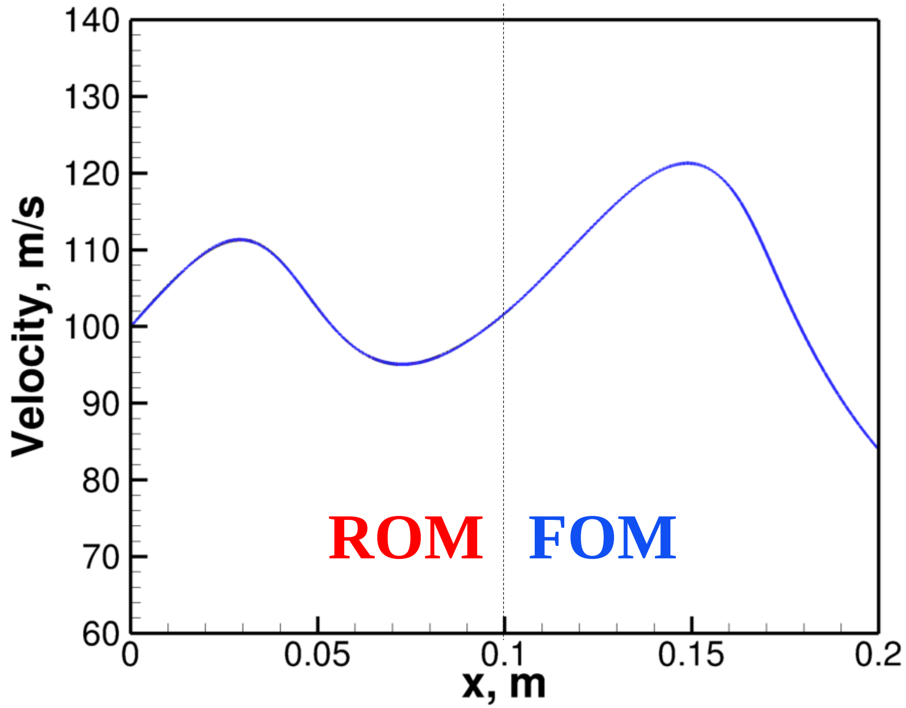
$$\arg\min_{\tilde{\mathbf{q}}_{p,1}} \left\| \mathbf{r}\left(\tilde{\mathbf{q}}_{p,1}, \mathbf{q}_{p,2,\partial\Omega_{1/2}}\right)\right\|_2^2 \text{ on } \Omega_1$$

$$s.t.\ \tilde{\mathbf{q}}_{p,1,\partial\Omega} + \nabla\tilde{\mathbf{q}}_{p,1,\partial\Omega} \cdot d\Omega_1 = \mathbf{q}_{p,2,\partial\Omega} + \nabla\mathbf{q}_{p,2,\partial\Omega} \cdot d\Omega_2$$

$$\mathbf{r}\left(\mathbf{q}_{p,2}, \tilde{\mathbf{q}}_{p,1,\partial\Omega_{1/2}}\right) = 0 \text{ on } \Omega_2$$

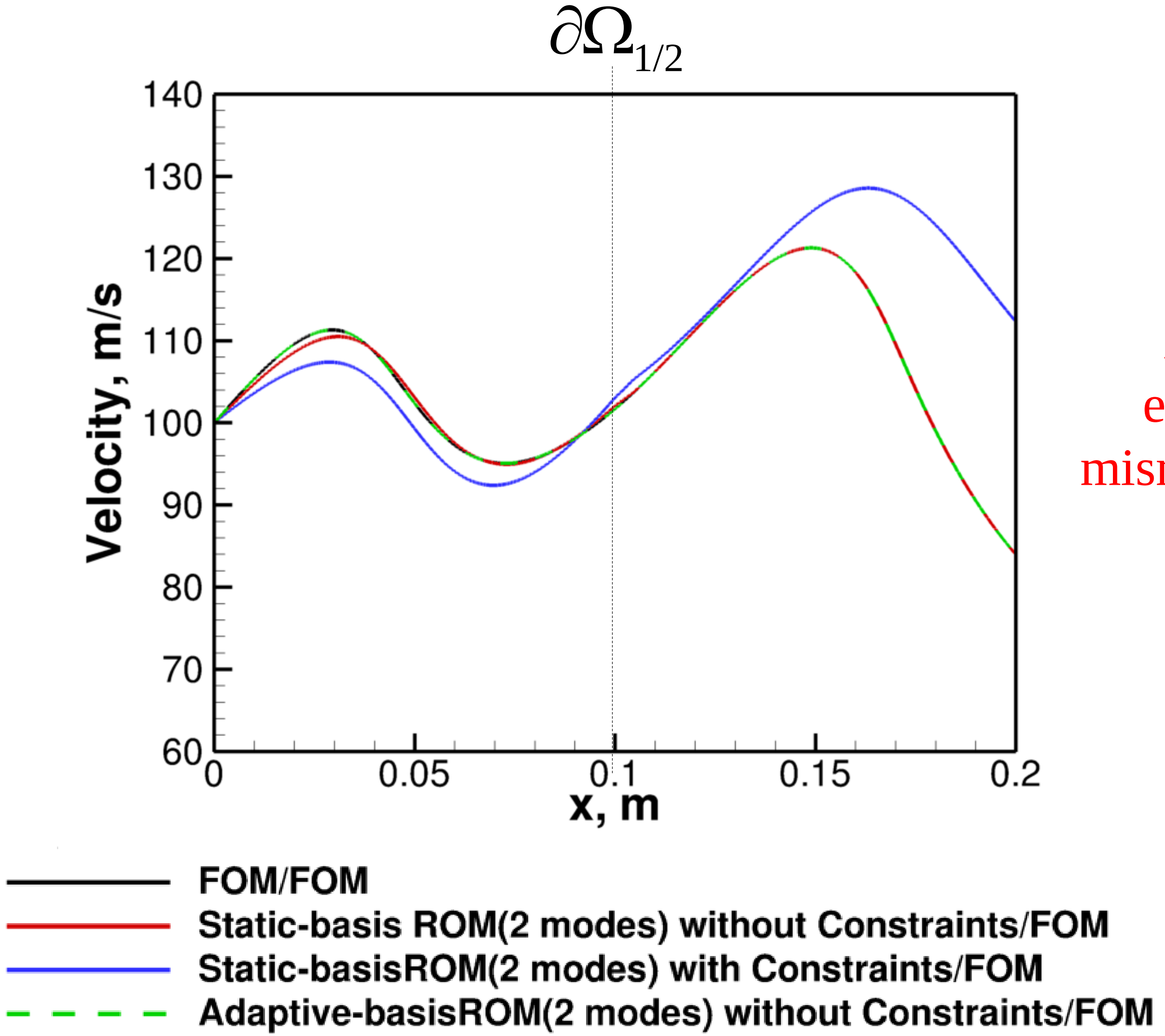
- Farhat et al, 2000
- Lucia et al., 2003
- Baiges et al, 2013
- Hoang et al., 2020
- Xiao et al, 2021





Imposing constraints in ROM eliminates the interface mismatch BUT compromises the ROM accuracy

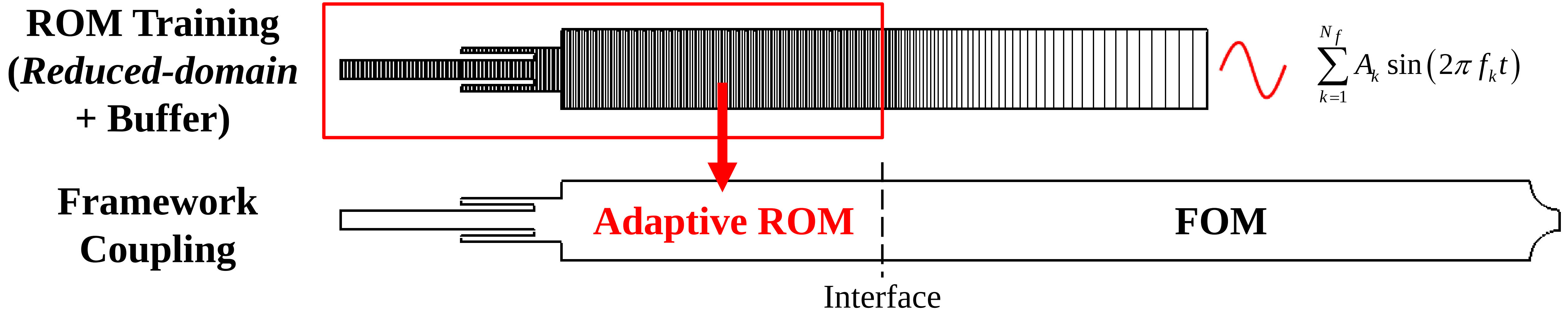
Domain Decomposition (Adaptive-basis ROM/FOM)



Adaptive-basis ROM eliminates the interface mismatch AND improves the ROM accuracy

Component-based ROM Framework

- **ROM training**: downstream perturbations to mimic full-domain acoustics
  - $N_f = 2$ with $f_1 = 1100\text{Hz}$ (1L for $L_{\text{chamber}} = 0.42$ m) and $A_1 = 0.05$
    $f_2 = 1300\text{Hz}$ (1L for $L_{\text{chamber}} = 0.35$ m) and $A_2 = 0.05$
- **ROM/FOM Coupling:** flux exchange at interface
  - FOM is used in the downstream to evaluate the framework performance

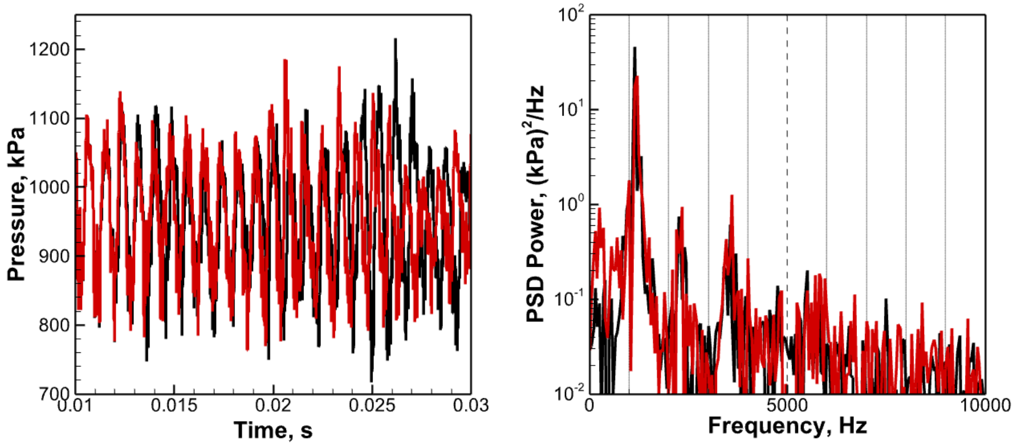


**** no hyper-reduction in adaptive ROM**

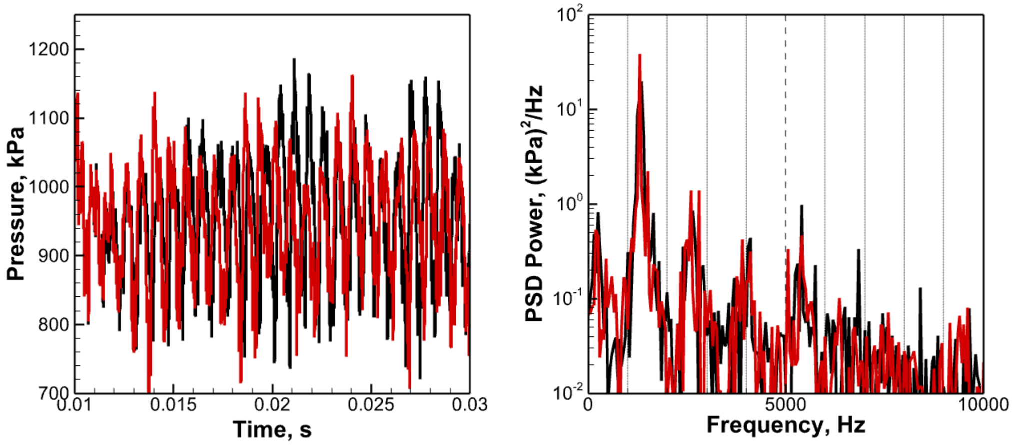
Comparisons of *Local* Pressure Signals

- ROM (trained using ~ **0.01** ms) + FOM to predict **20** ms full-domain dynamics
- Adaptive ROM+FOM framework accurately predict the instability behaviors for different chamber lengths

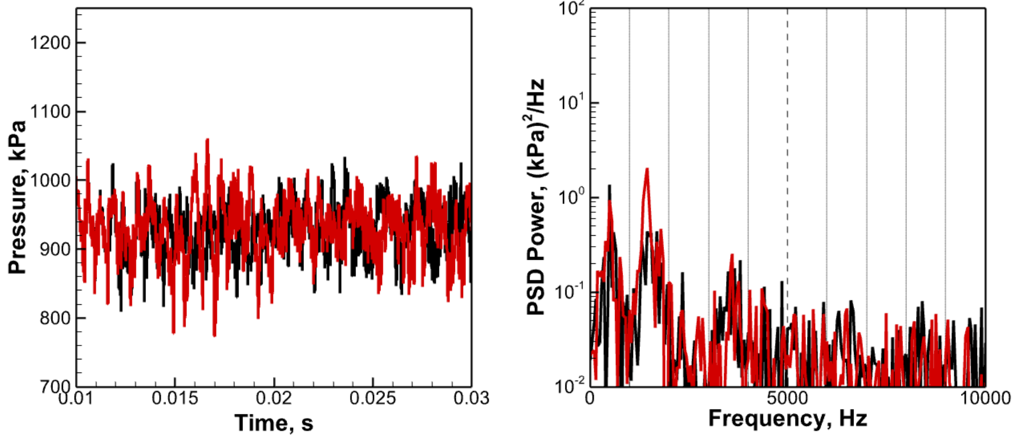
$L_{chamber}$ = **0.42 m**



$L_{chamber}$ = **0.35 m**



$L_{chamber}$ = **0.28 m**

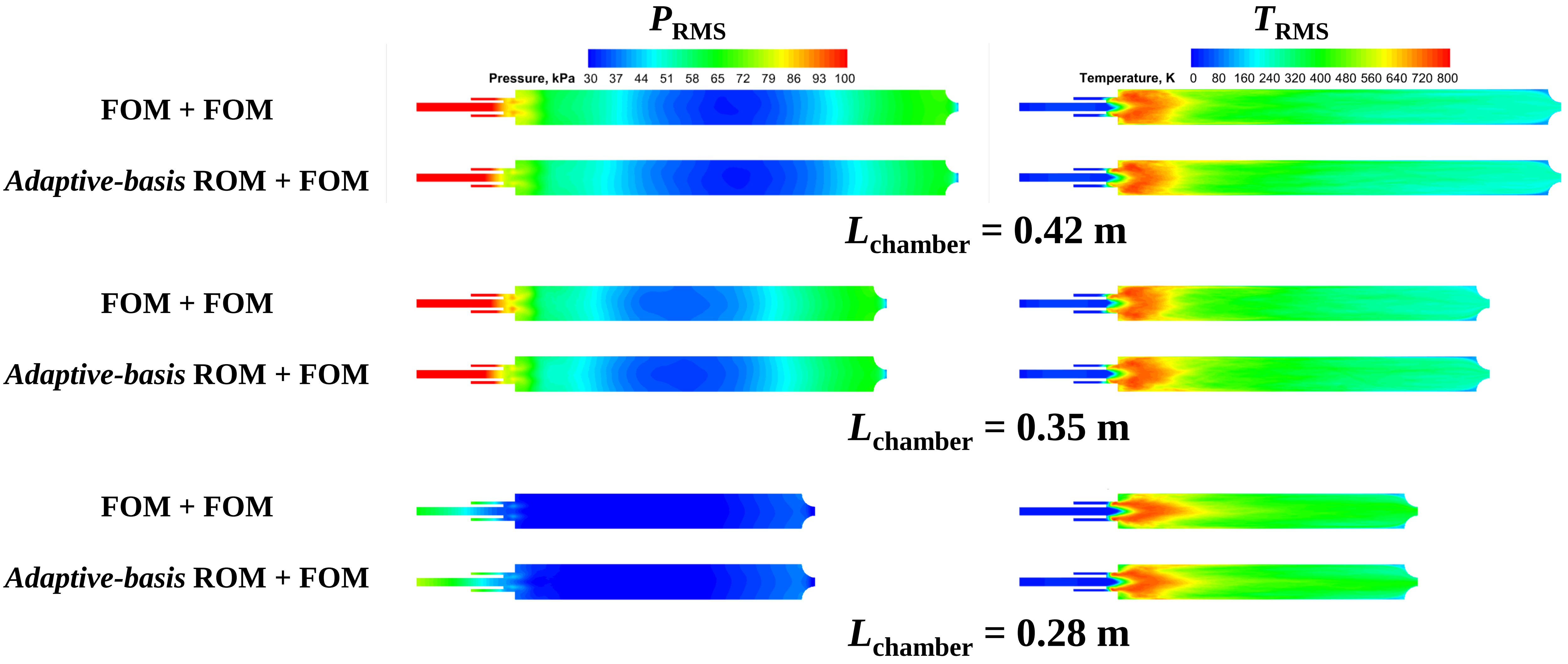


FOM + FOM

ROM + FOM

Comparisons of RMS Fields

- Adaptive ROM+FOM framework accurately predict the RMS for different chamber lengths



Outline

- **Test Case I**: 2D Single-Injector Rocket Combustors with Variable Geometries

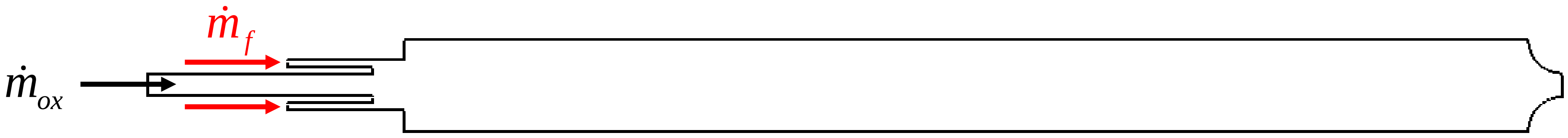
- **Test Case II**: 2D Single Injector with Variable Mass Flow Rates

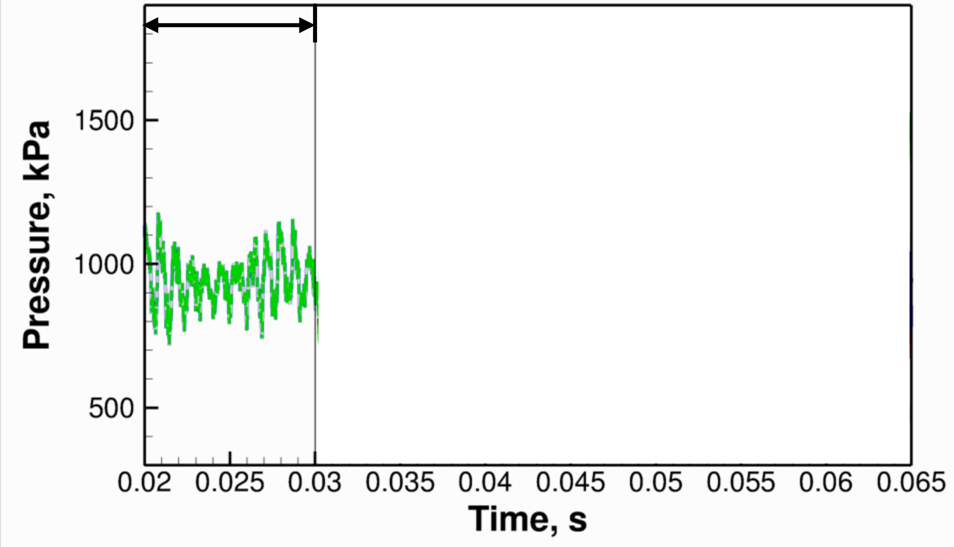
- **Test Case III**: 2D Multi-Injector Rocket Combustors with Variable Geometries

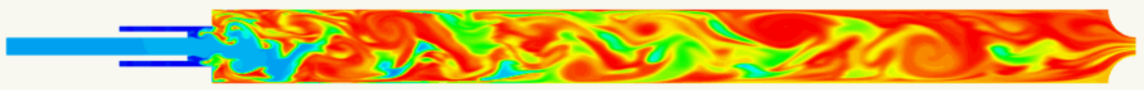
Test Case II: 2D Single Injector with Variable Mass Flow Rates

Single-injector simulated with reference mass flow rates (ϕ_{ref} = 0.8) from 0 – 30ms

- Mass flow rates of both *fuel* and *ox* are changed after 30ms leading to different limit cycles







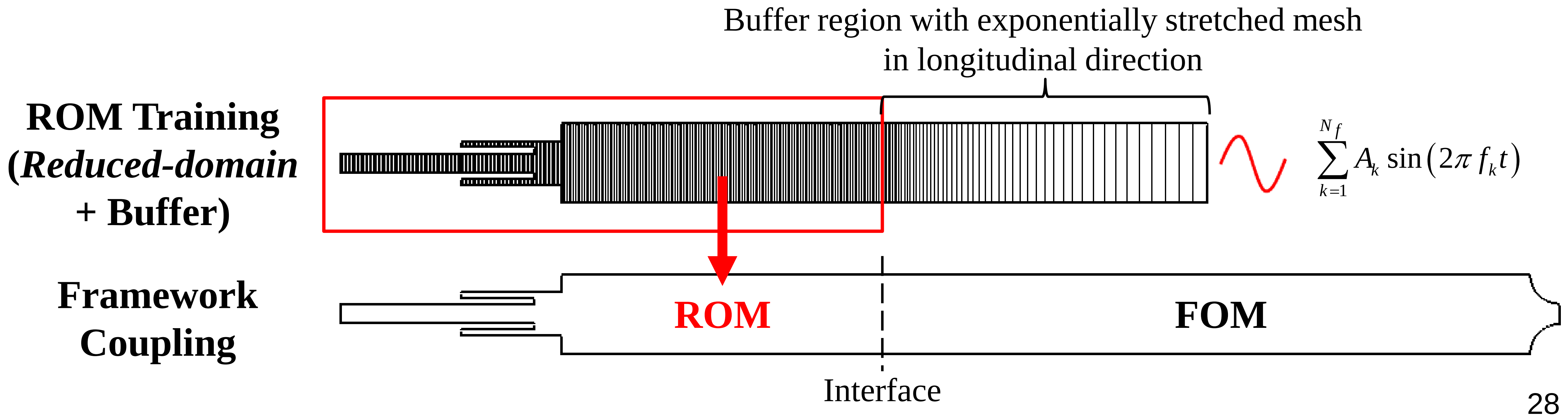
Component-based ROM Framework

- **ROM Training**: *Reduced-domain* + buffer

  → *Single* downstream boundary forcing

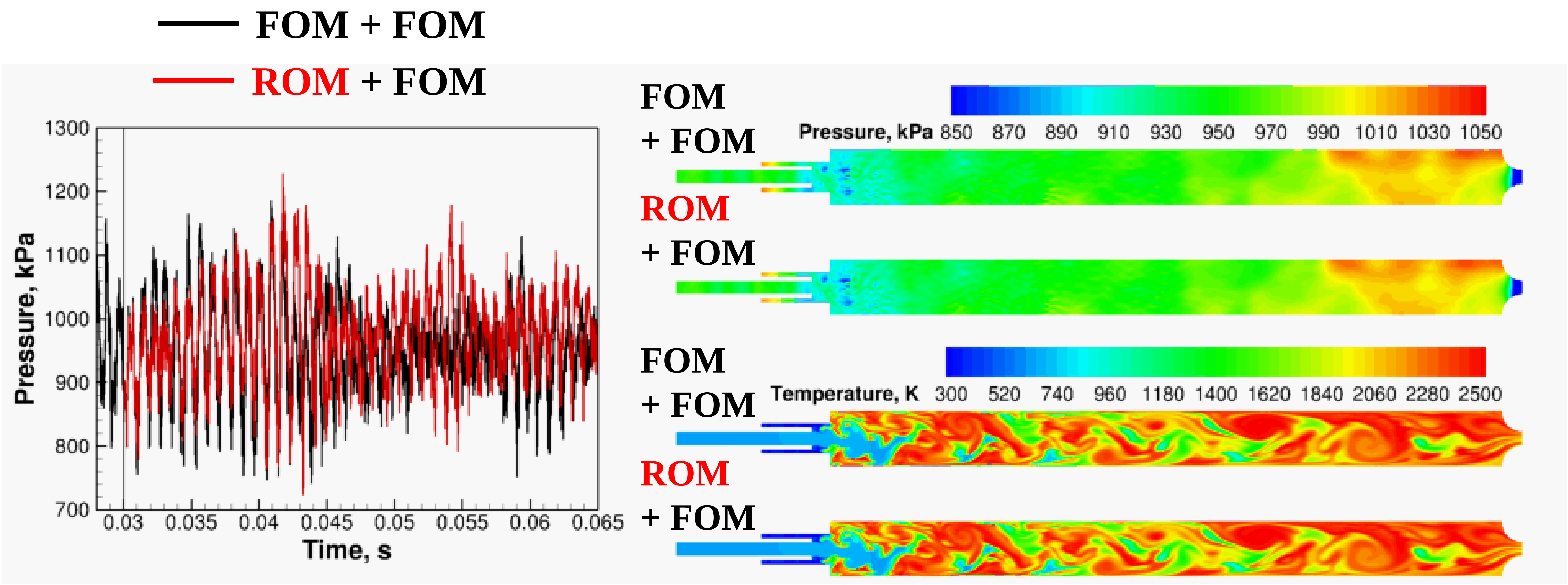
  → *Adaptive* ROM with **2% sampling points and z_s = 5**

- **ROM/FOM Coupling:** flux exchange at interface
  - FOM is used in the downstream to evaluate the framework performance



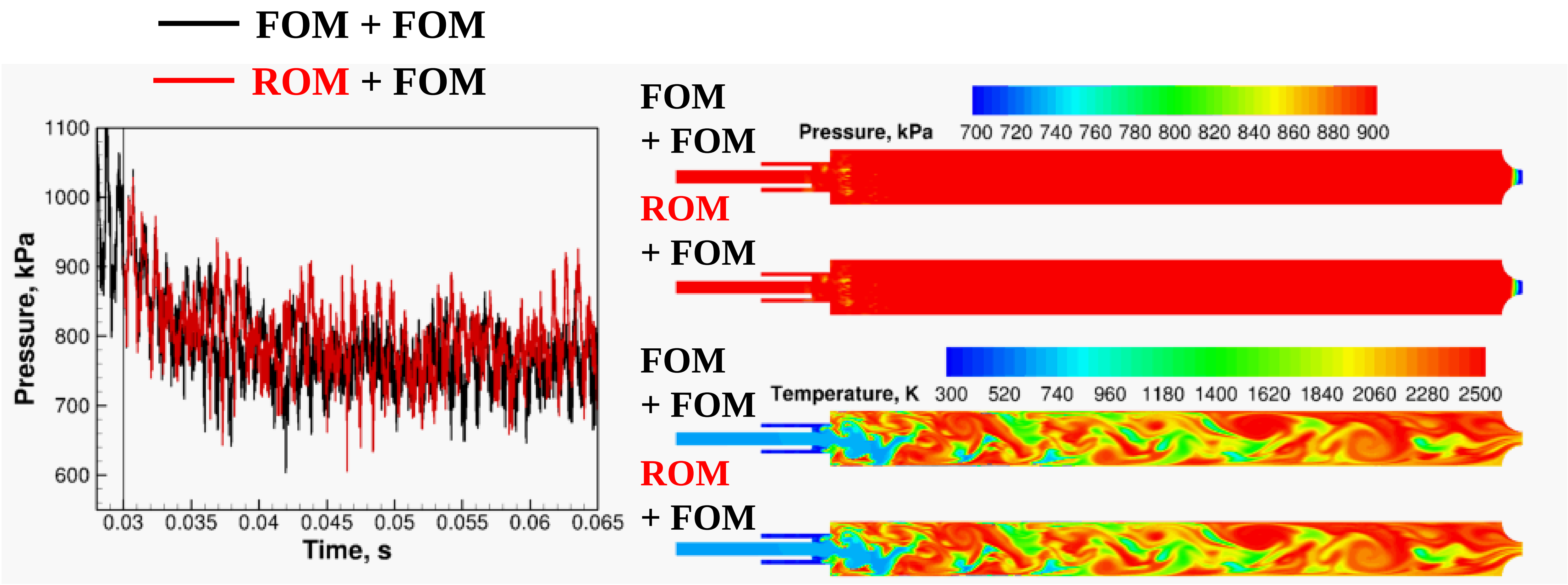
$$100\%\ \dot{m}_{f,ref} + 100\%\ \dot{m}_{ox,ref} \quad (\phi = 0.8)$$

- As the calculations proceed, phase differences between FOM+FOM and ROM+FOM show up in the local pressure signals



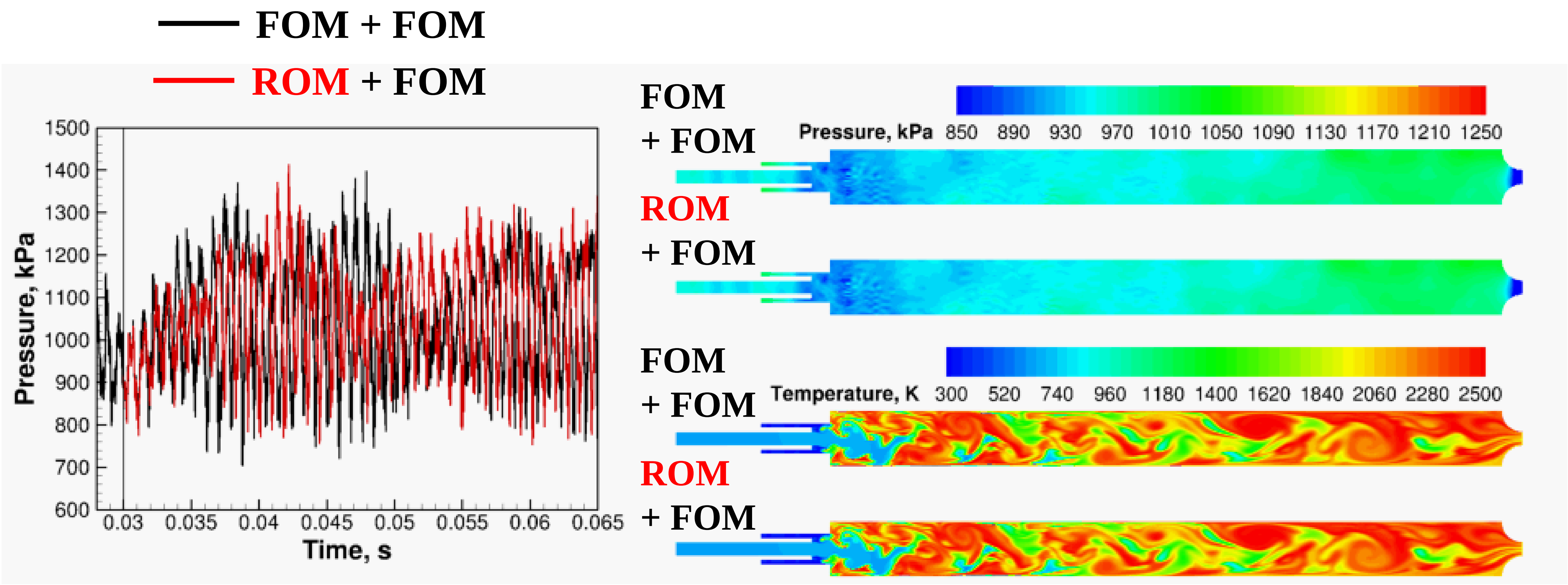
$$50\%\ \dot{m}_{f,ref} + 100\%\ \dot{m}_{ox,ref}\quad (\phi = 0.4)$$

- As the calculations proceed, phase differences between FOM+FOM and ROM+FOM show up in the local pressure signals



$$150\%\ \dot{m}_{f,ref} + 100\%\ \dot{m}_{ox,ref} \quad (\phi = 1.2)$$

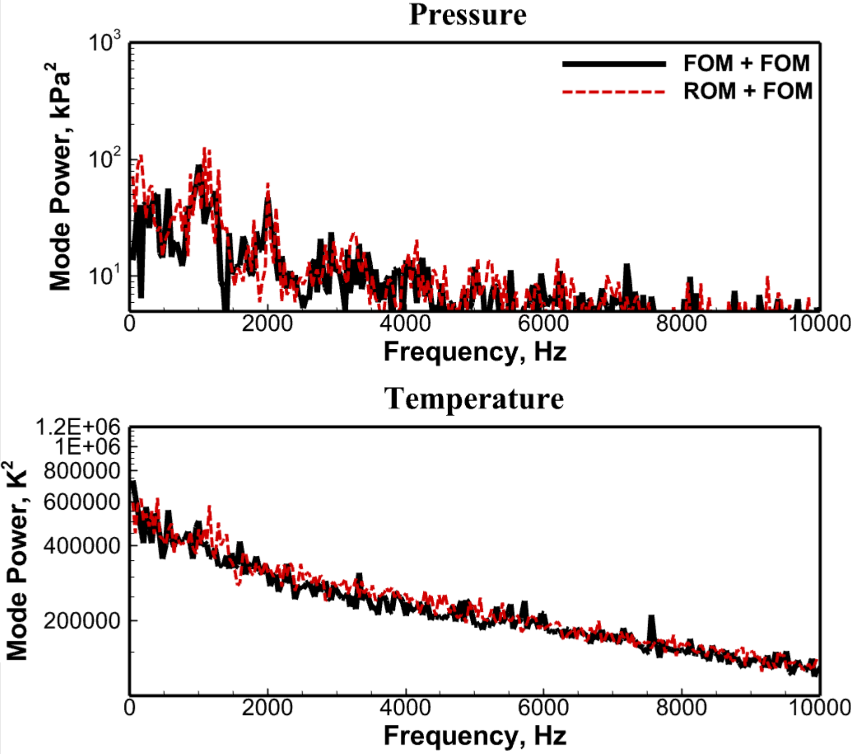
- As the calculations proceed, phase differences between FOM+FOM and ROM+FOM show up in the local pressure signals



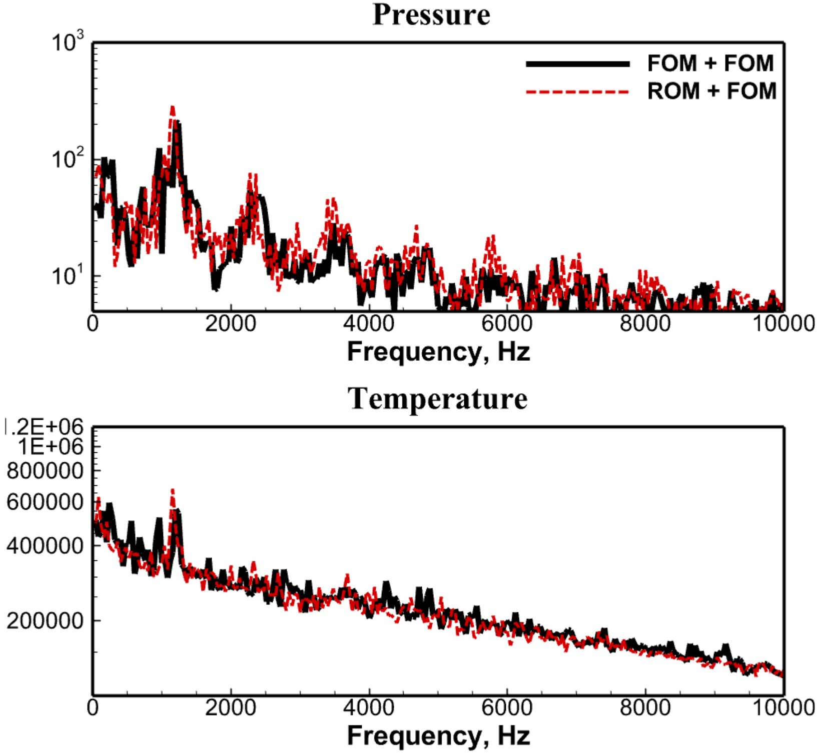
DMD Spectra Comparisons – Equivalence Ratio Effects

- DMD analysis based on snapshots from 40 to 65ms

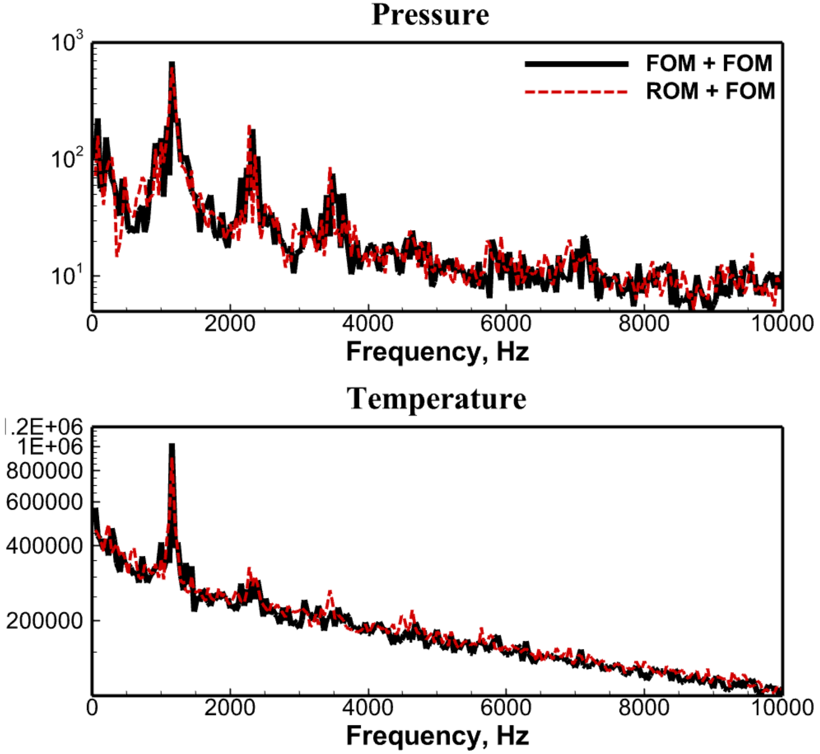
$50\%\ \dot{m}_{f,ref} + 100\%\ \dot{m}_{ox,ref}\quad (\phi = 0.4)$



$100\%\ \dot{m}_{f,ref} + 100\%\ \dot{m}_{ox,ref}\quad (\phi = 0.8)$

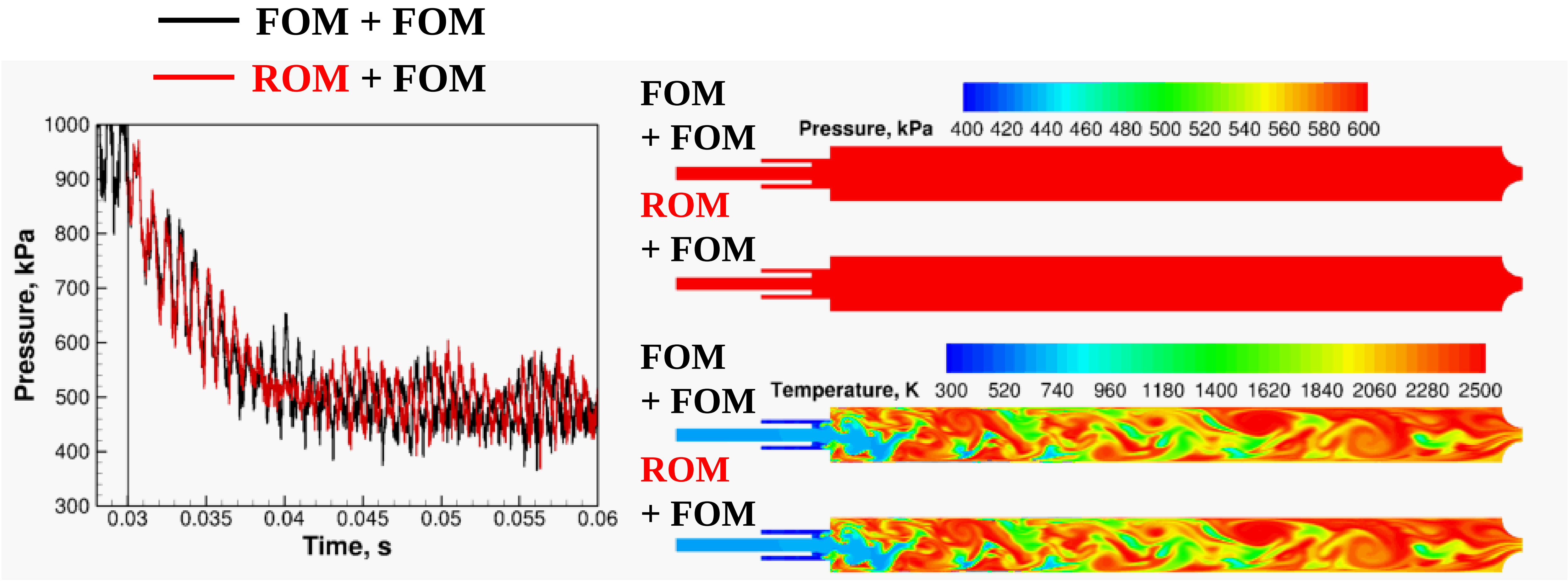


$150\%\ \dot{m}_{f,ref} + 100\%\ \dot{m}_{ox,ref}\quad (\phi = 1.2)$



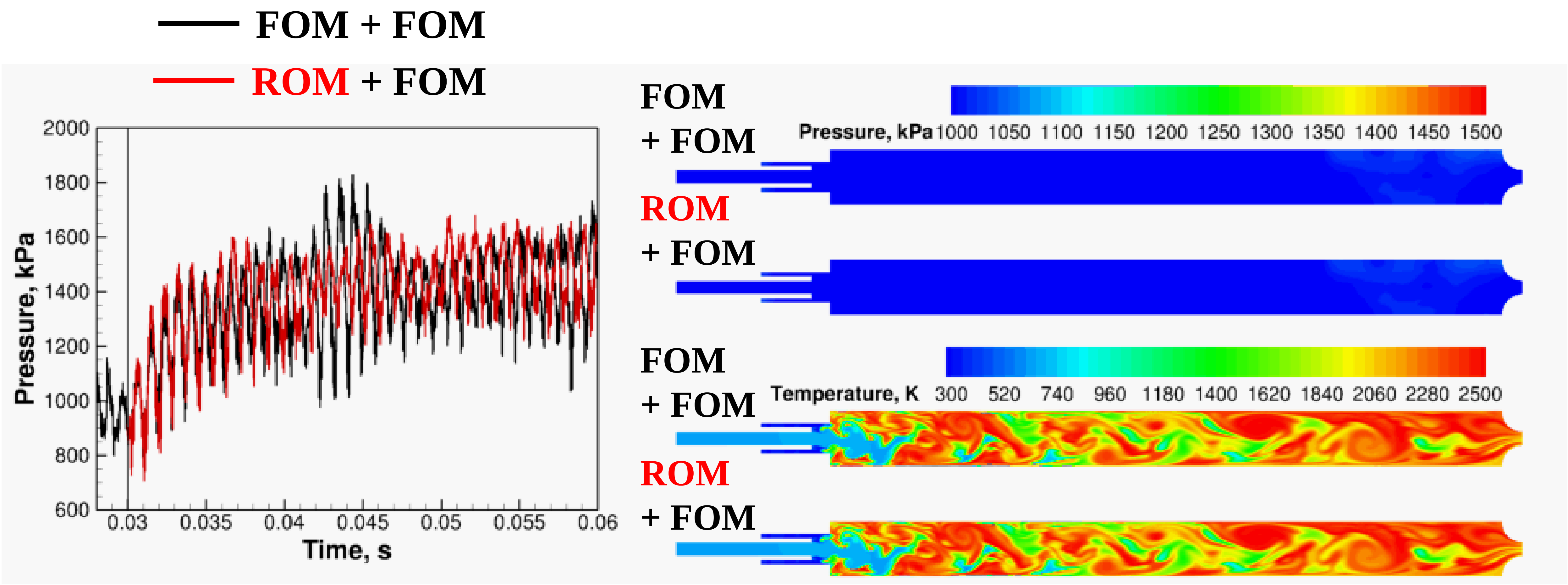
$$50\%\ \dot{m}_{f,ref} + 50\%\ \dot{m}_{ox,ref} \quad (\phi = 0.8)$$

- Phase shifts in local pressure signals are minor here but still present



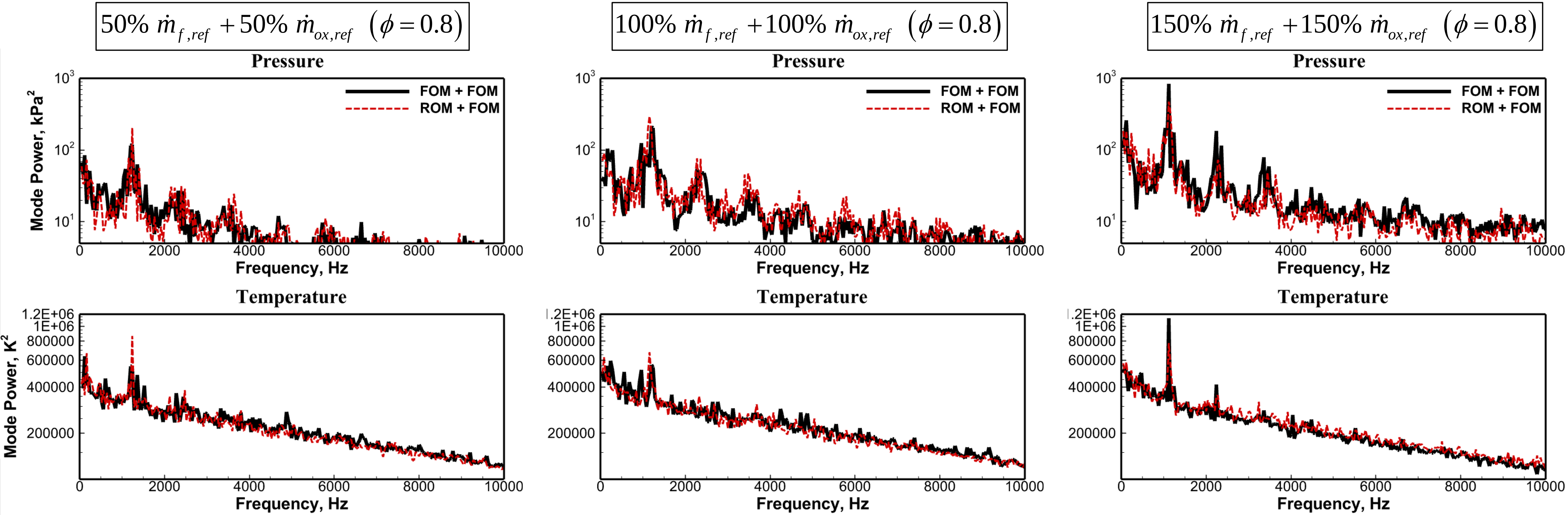
$$150\%\ \dot{m}_{f,ref} + 150\%\ \dot{m}_{ox,ref}\quad (\phi = 0.8)$$

- Phase shifts in local pressure signals are minor here but still present



DMD Spectra Comparisons – Mean Flow Effects

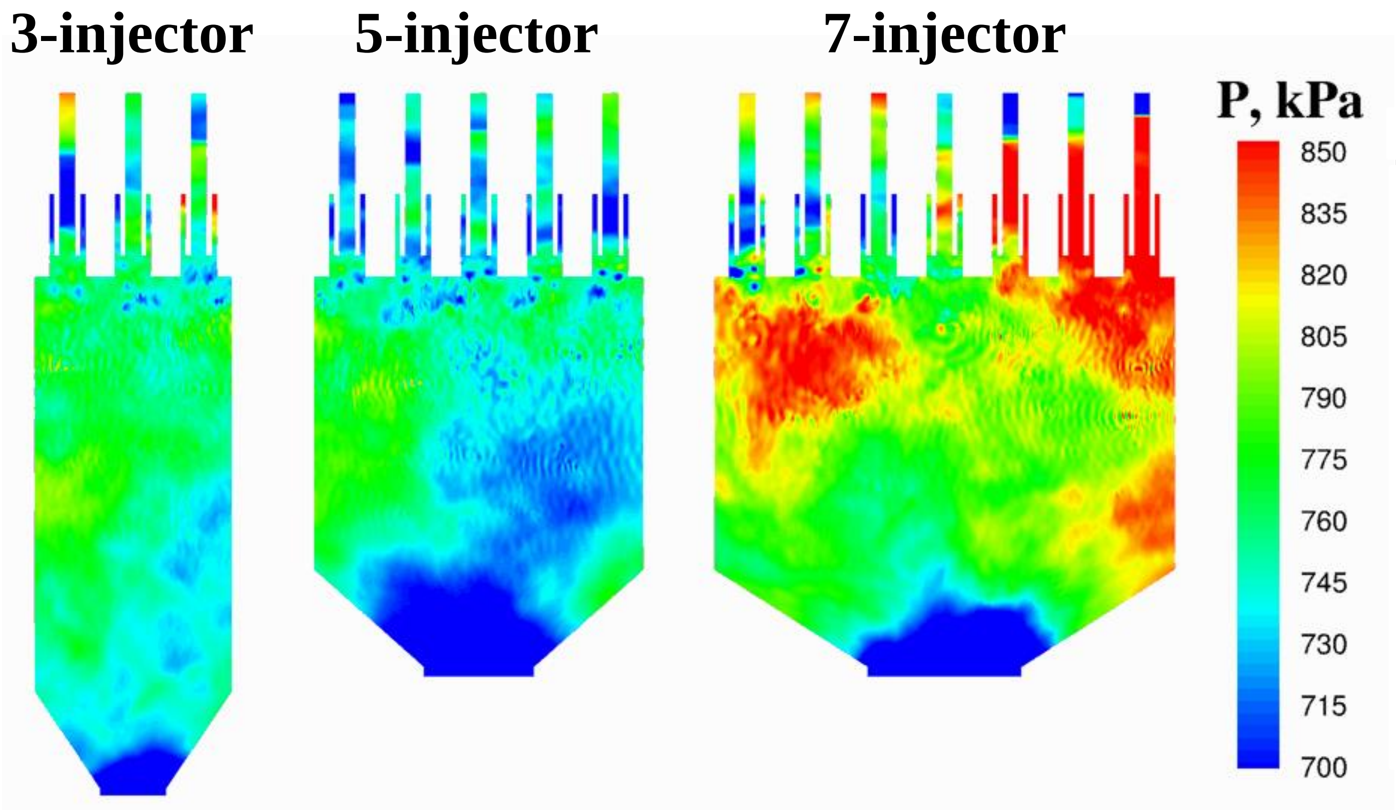
- DMD analysis based on snapshots from 40 to 65ms

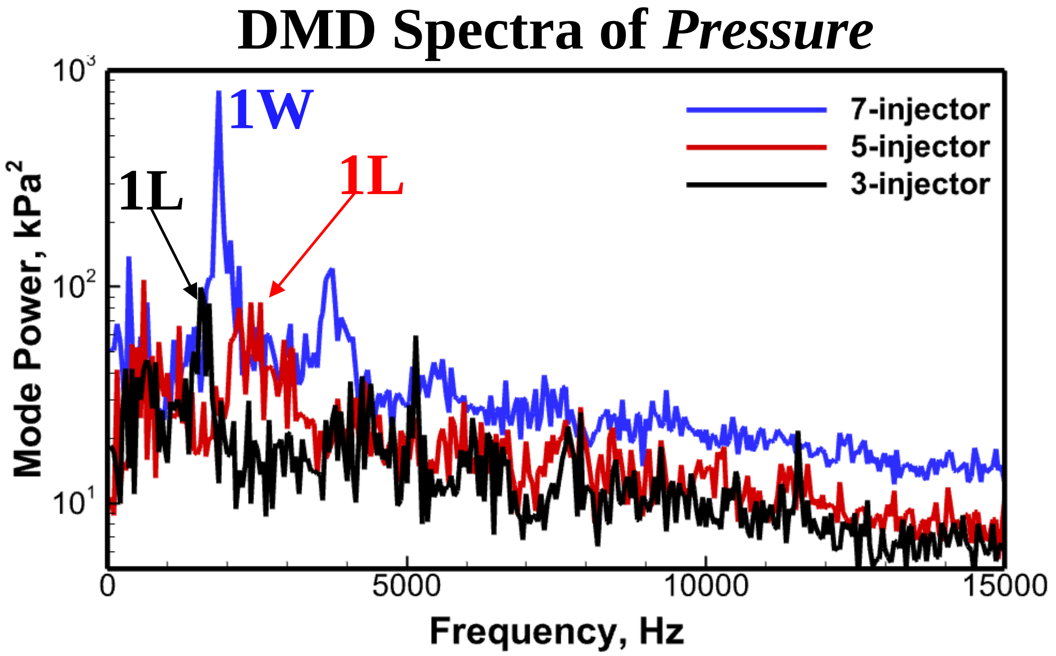


Outline

- **Test Case I**: 2D Single-Injector Rocket Combustors with Variable Geometries
- **Test Case II**: 2D Single Injector with Variable Mass Flow Rates
- **Test Case III**: 2D Multi-Injector Rocket Combustors with Variable Geometries

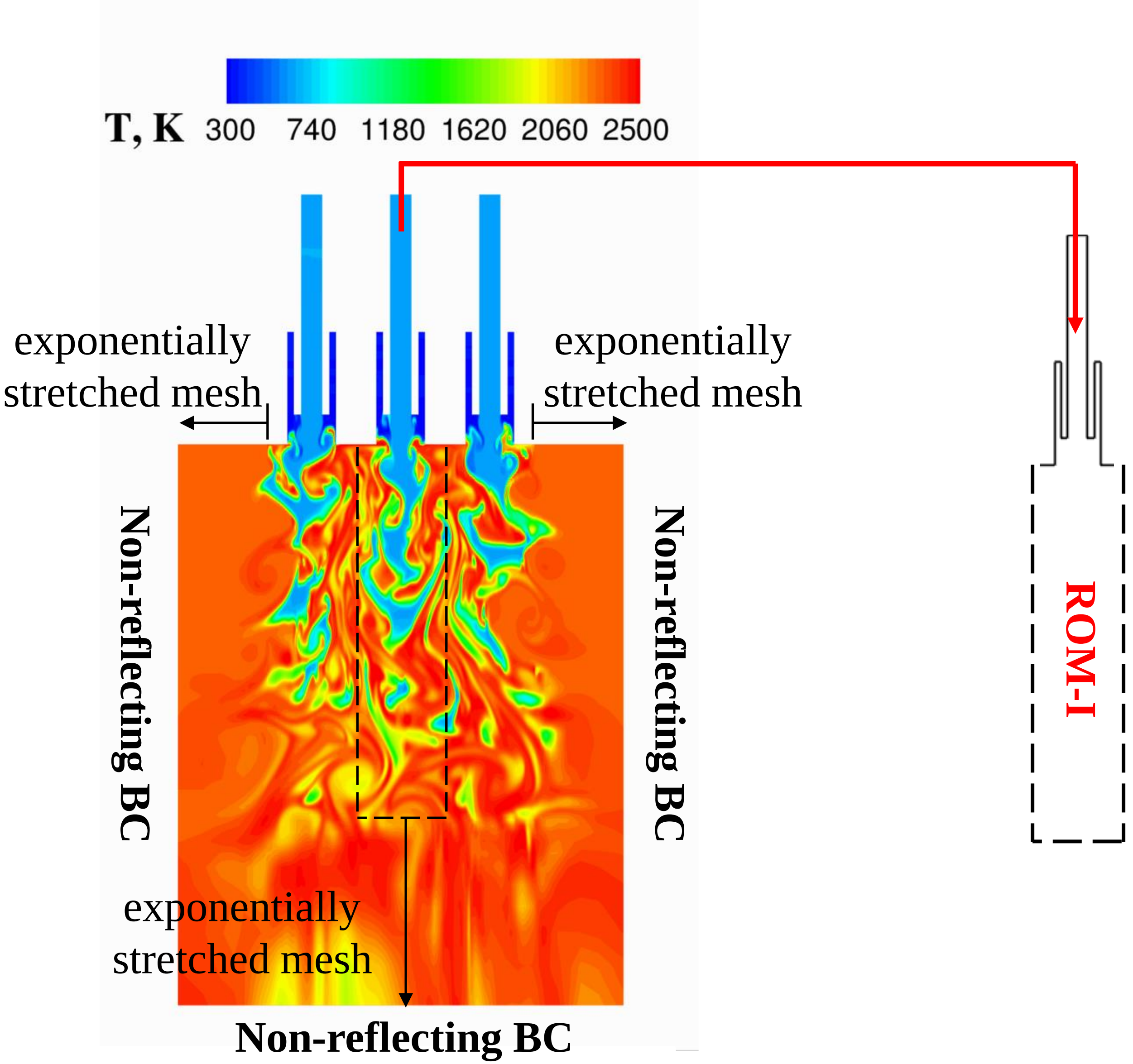
Test Case III: 2D Multi-Injector Rocket Combustors with Variable Geometries





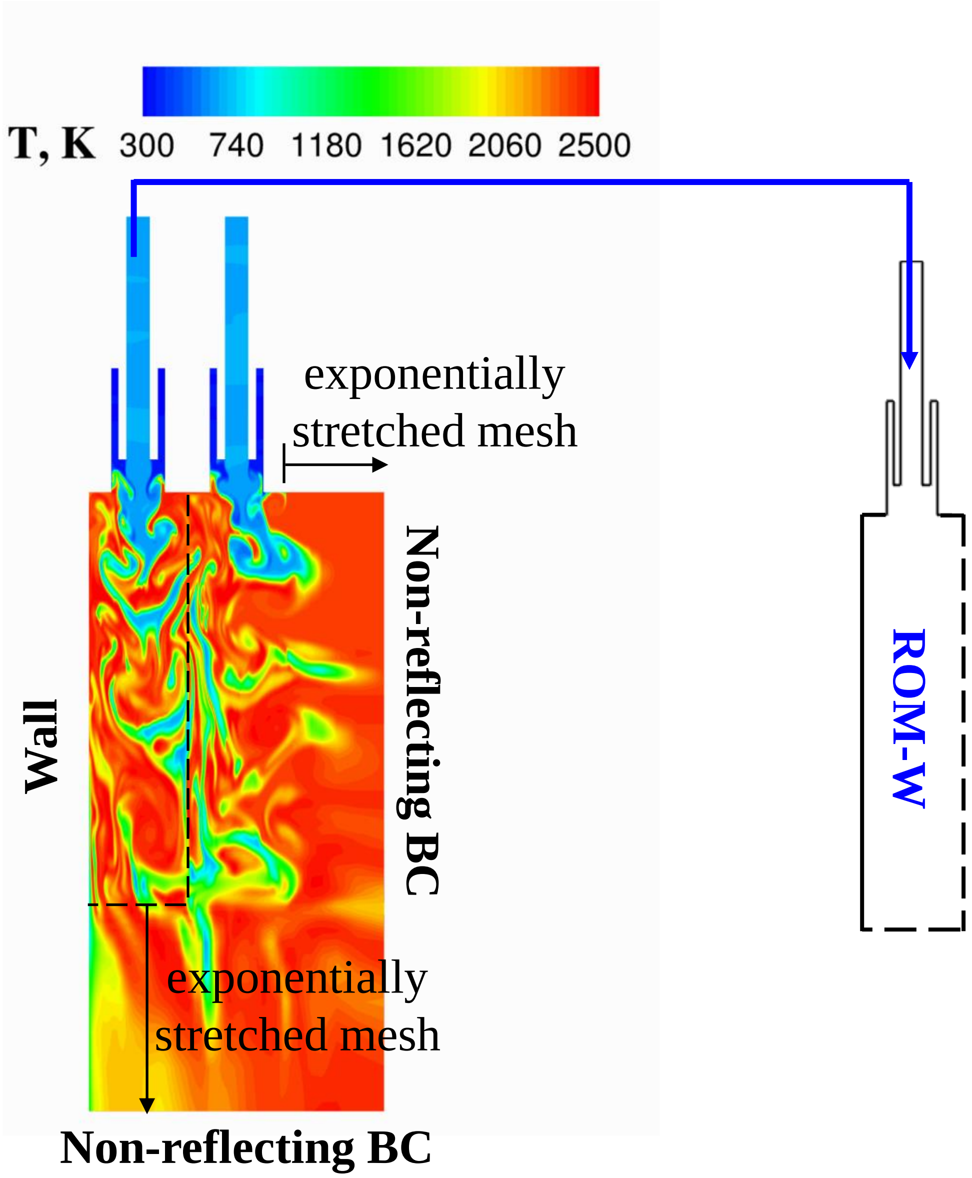
3-injector Training for ROM of Interior Injector (ROM-I)

- **Two auxiliary injectors included to:**
  - Account for interactions between injectors
  - Incorporate the complex interface dynamics
- **Extended buffer regions included to:**
  - Account for reverse flow at the interface
  - Mitigate the effects of numerical wave reflections from the boundaries
- **Run FOM for 100 time steps**

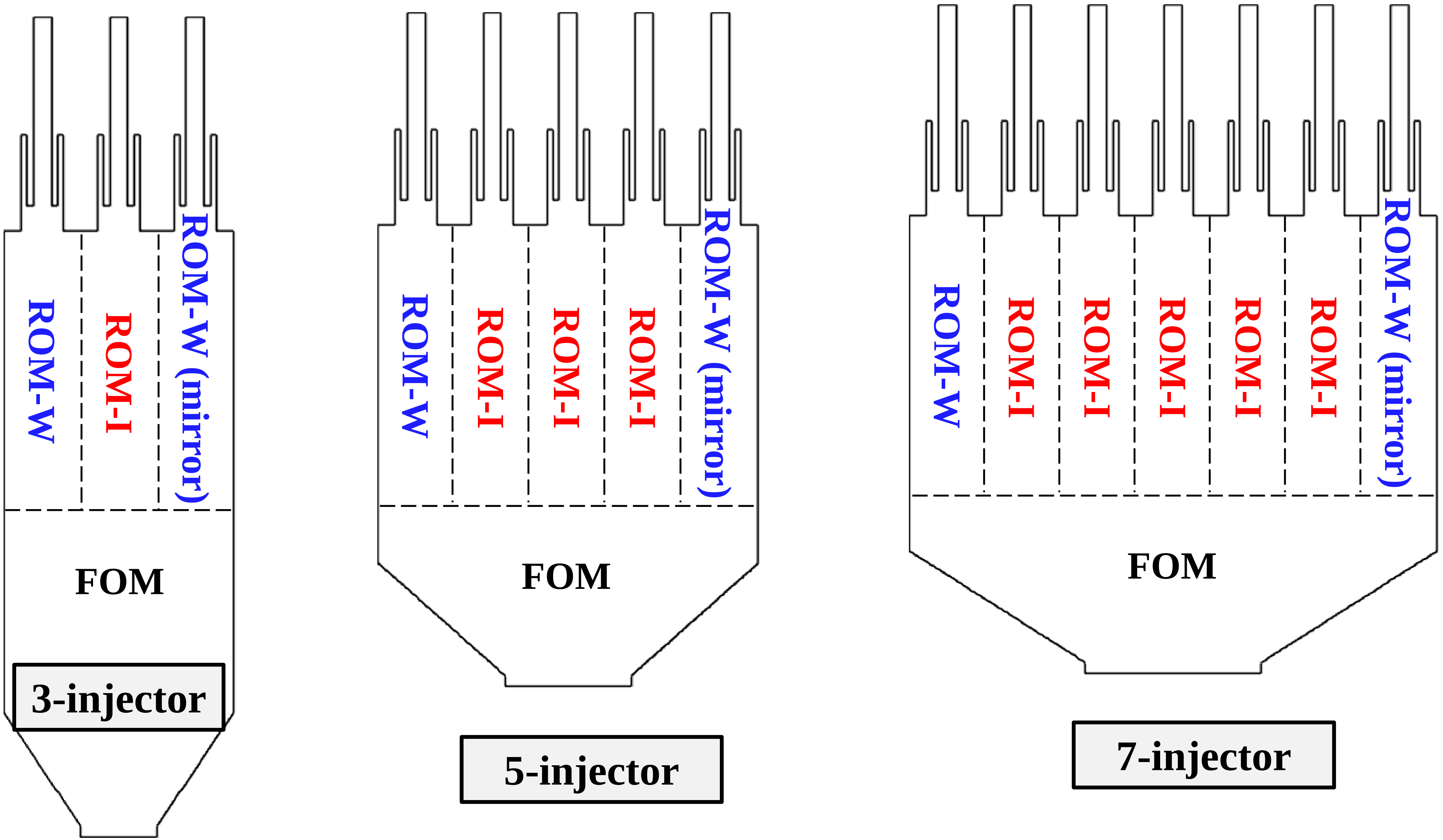


2-injector Training for ROM of Wall Injector (**ROM-W**)

- **<u>One</u> auxiliary injectors included to:**
  - Account for interactions between injectors
  - Incorporate the complex interface dynamics
- **Extended buffer regions included to:**
  - Account for reverse flow at the interface
  - Mitigate the effects of numerical wave reflections from the boundaries
- **Run FOM for 100 time steps**

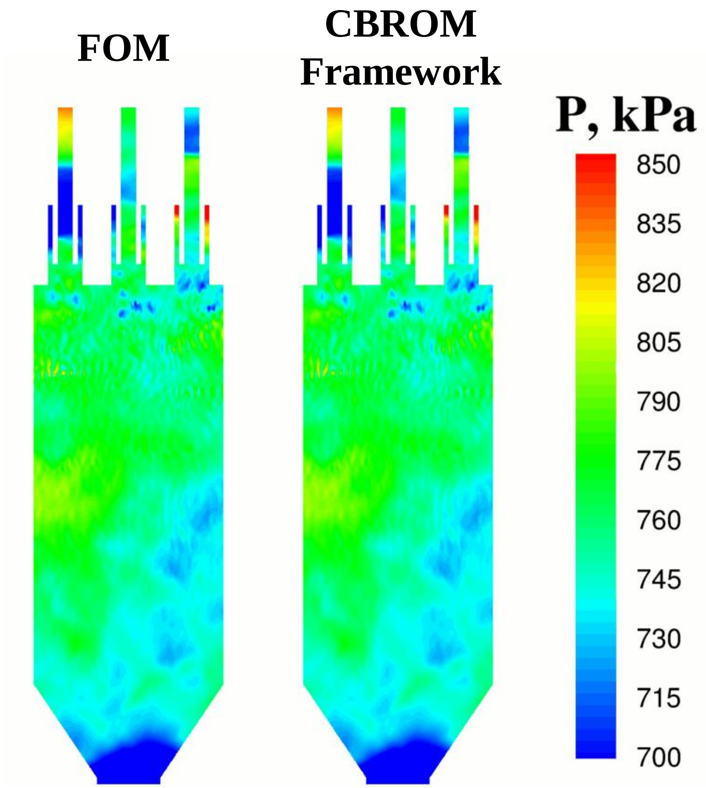


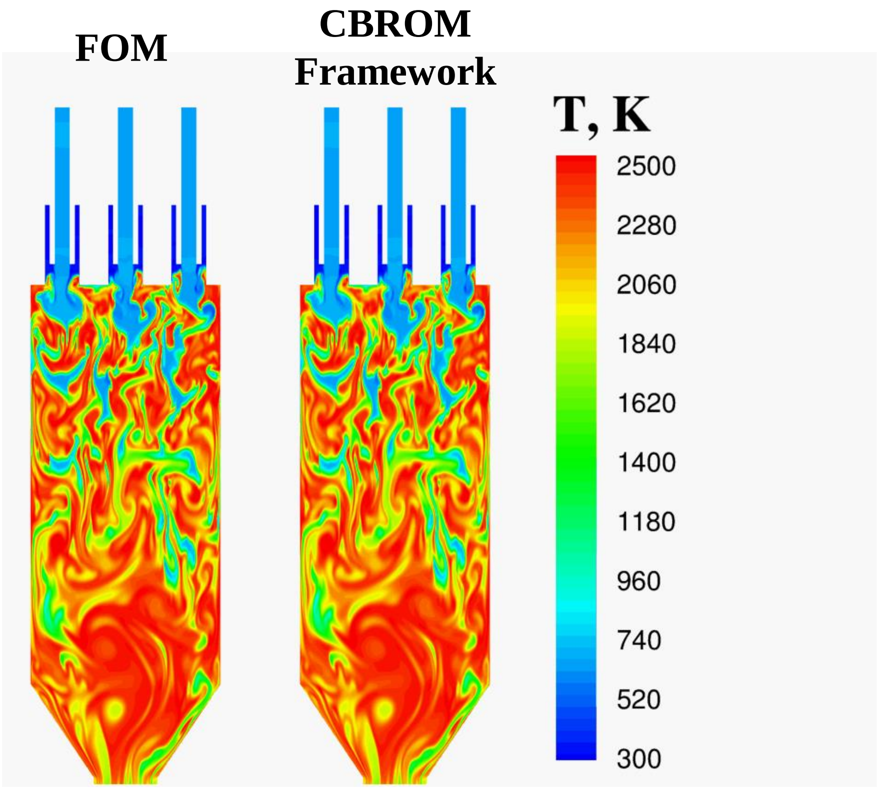
Component-based ROM Framework

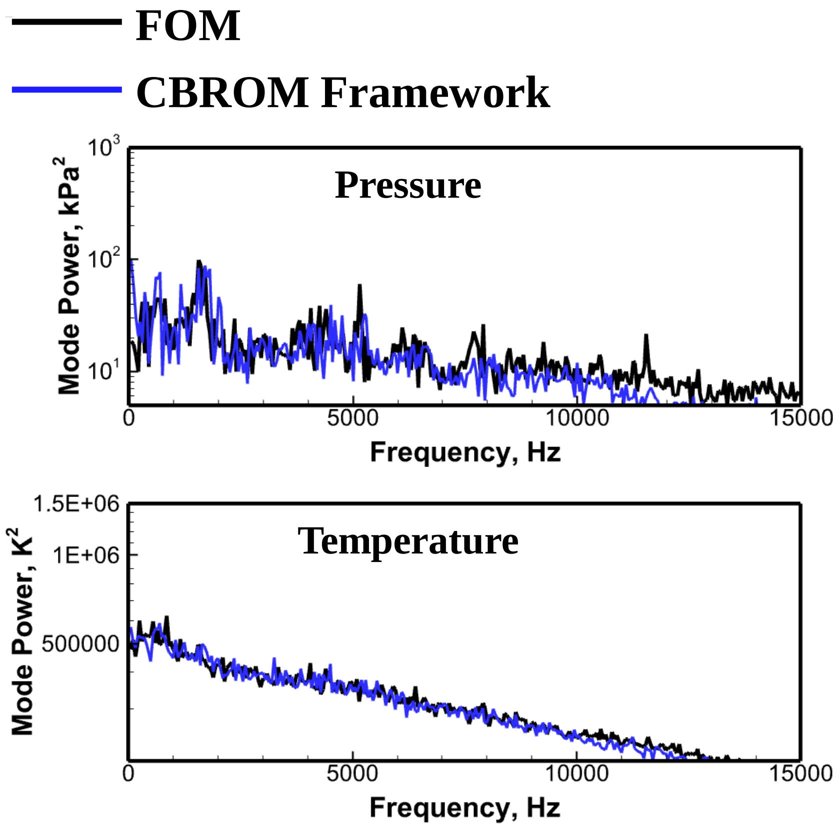


FOM vs CBROM Framework: 3-injector Configuration

Adaptive ROM (One ROM-I and Two ROM-W)

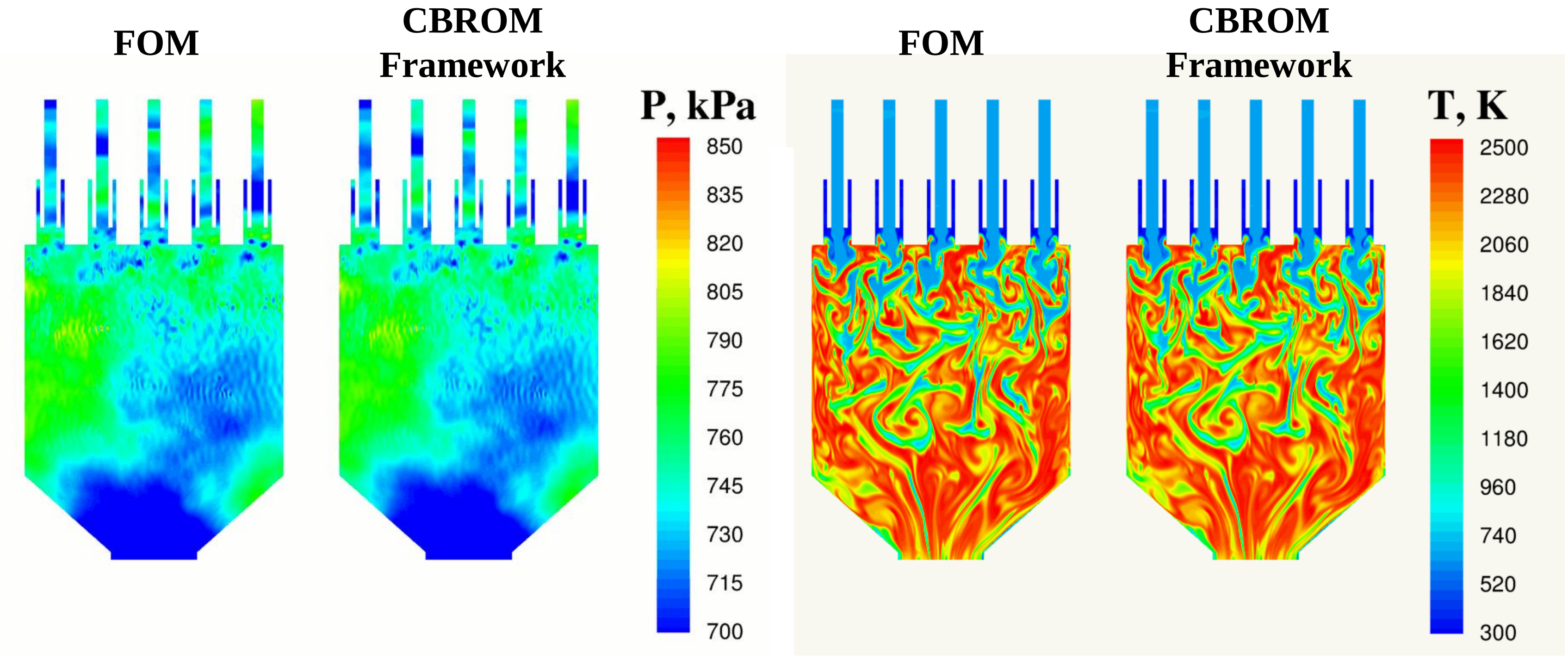


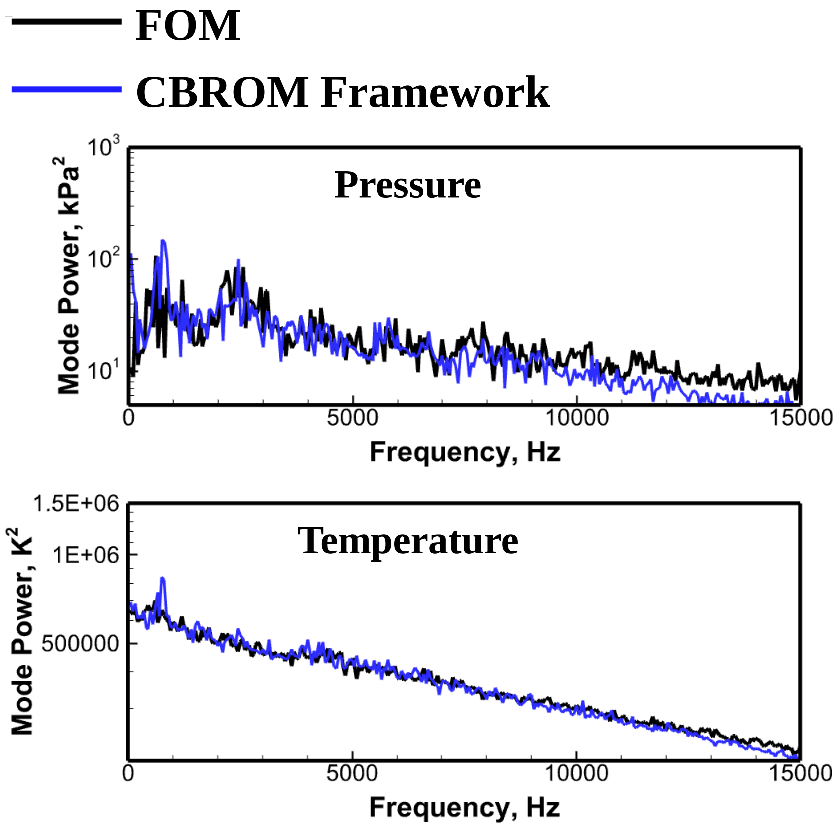




FOM vs ROM Framework: 5-injector Configuration

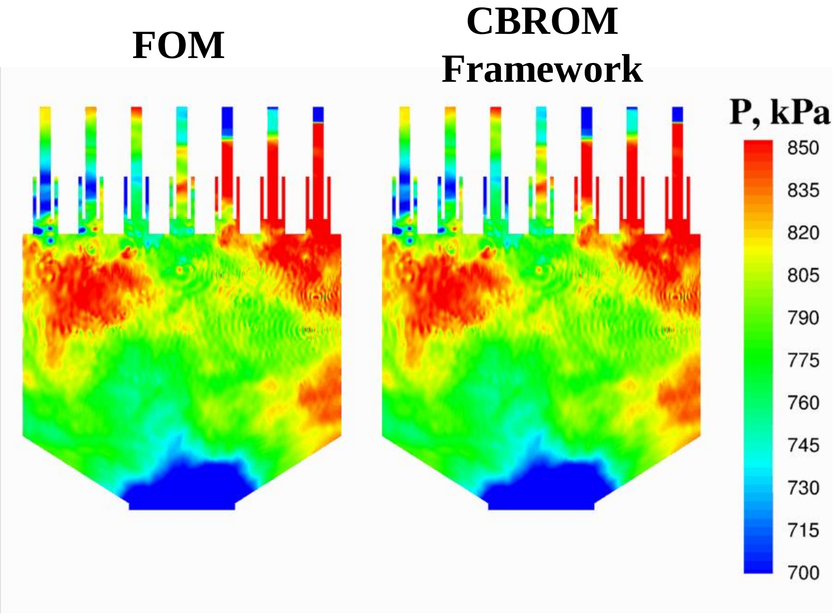
Adaptive ROM (Three ROM-I and Two ROM-W)

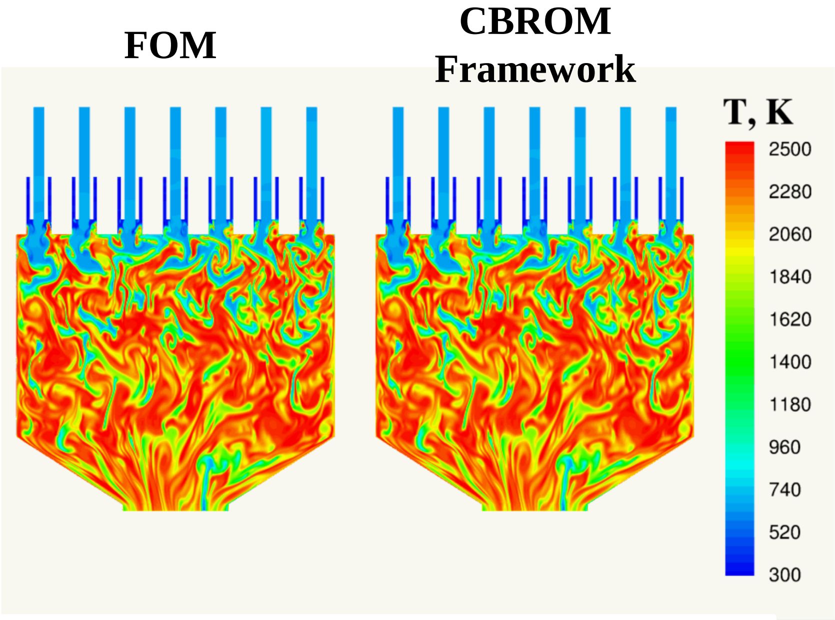


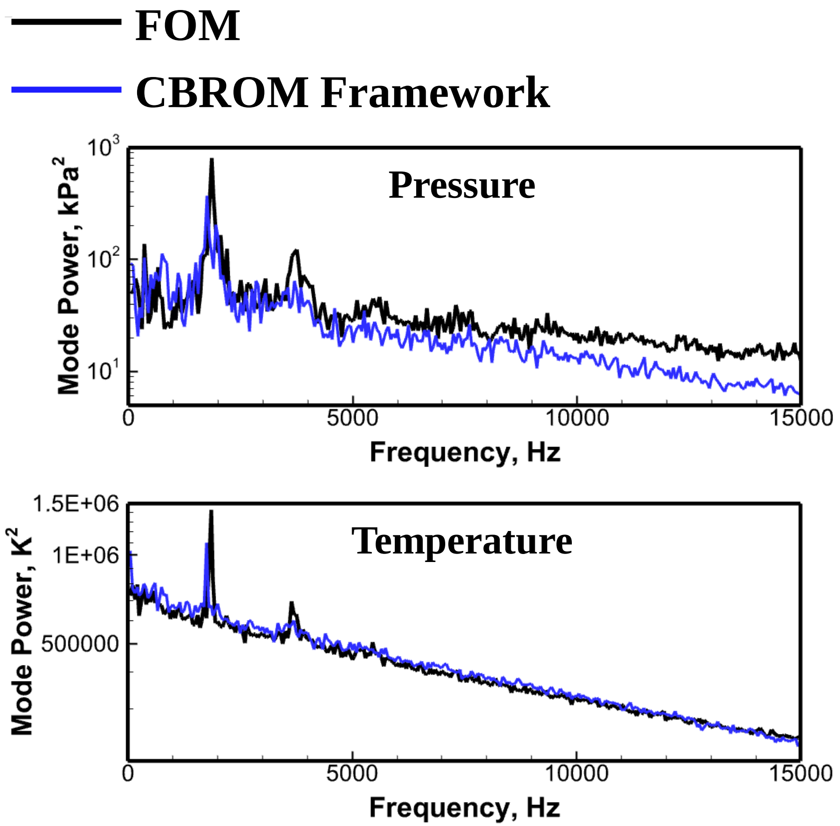


FOM vs ROM Framework: 7-injector Configuration

Adaptive ROM (Five ROM-I and Two ROM-W)







Summary

Component-based ROM framework demonstrated on modeling *self-excited* combustion dynamics in rocket combustors

- ROM training with reduced number of injectors + buffer
  - ➢ No need to perform FOM simulations on the full configuration
- Adaptivity enables predictive ROM in the framework
  - ➢ Accurate predictions of the dynamics for different combustor configurations and operating conditions

Huang, Duraisamy, and Merkle, Frontiers in Physics 2022

Huang, AIAA SciTech 2023