AIAA AVIATION Forum
June 27-July 1, 2022, Chicago, IL & Virtual
AIAA AVIATION 2022 Forum

10.2514/6.2022-3487

On filtering in non-intrusive data-driven reduced-order modeling

Ionuţ-Gabriel Farcaş [*]
The University of Texas at Austin, Austin, TX, 78712

Ramakanth Munipalli[†]
Air Force Research Laboratory, Edwards AFB, CA 93524

Karen E. Willcox[‡]
The University of Texas at Austin, Austin, TX, 78712

This paper presents a method to enhance data-driven reduced-order modeling with a preprocessing step in which the training data is filtered prior to training the reduced model. Filtering the data prior to training the reduced model has a number of potential benefits for data-driven modeling: it attenuates (or even eliminates) frequency content that would otherwise be difficult or impossible to capture with the reduced model, it smoothens discontinuities in the data that would be difficult to capture in a low-dimensional representation, and it reduces noise in the data. This leads to the reduced model learning process becoming numerically better conditioned, less sensitive to numerical errors in the training data, and less prone to overfitting when the amount of training data is limited. This paper first illustrates the effects of filtering in a one-dimensional periodic inviscid Burgers' equation with a solution characterized by a moving shock. A second example considers a large-scale rotating detonation engine simulation with more than 3.8 million degrees of freedom for which only a few hundred down-sampled snapshots are available from the high-fidelity simulation. A reduced-order model is derived from these snapshots using operator inference. The results show that spatial filtering of snapshots prior to inferring the reduced model reduces approximation accuracy over the training regime, but improves prediction accuracy over the testing regime. This result indicates the potential benefits of filtering to reduce overfitting, which is particularly important for complex physical systems where the amount of training data is limited.

I. Introduction

Recent advances in computational science and high-performance computing enable the simulation of complex real-world problems with ever-increasing accuracy; however, for many important engineering applications the computational cost of a single simulation of the full-order model (FOM) is so large that only a limited number of cases can be studied. This limitation makes outer-loop scenarios (i.e., scenarios that require ensembles of high-fidelity simulations, such as design optimization, uncertainty quantification, parameter inference and control) computationally prohibitive. In some applications, such as the numerical simulation of reactive flows in rocket engines [1], even performing a single simulation for a sufficiently long time interval to extract meaningful information can be computationally prohibitive. In this work we aim to address these computational cost challenges by advancing the methodologies of non-intrusive reduced-order models (ROMs) trained on data stemming from numerical simulations.

We target reduced-order modeling of multiscale, multiphysics problems, where the data have a complex structure, with a broadband or high frequency content. Large classes of problems (e.g., hyperbolic problems and phase-field models) exhibit discontinuities or sharp interfaces, which pose particular challenges for creating accurate and predictive data-driven ROMs. Building ROMs of such complex problems is made even more challenging because generally only small data training sets are available due to the tremendous computational cost of the high-fidelity simulations, even on large supercomputers. This sparsity of data makes data-driven ROMs prone to overfitting. Furthermore, data-driven ROMs are known to be sensitive to noise, which means that care must be taken when the simulation code employs discretization techniques that introduce noise into the numerical solutions.

[*] Postdoctoral fellow, Oden Institute for Computational Engineering and Sciences, ionut.farcas@austin.utexas.edu, AIAA Member.
[†] Senior Research Aerospace Engineer, Combustion Devices (RQRC), ramakanth.munipalli@us.af.mil, AIAA Senior Member.
[‡] Director, Oden Institute for Computational Engineering and Sciences, kwillcox@oden.utexas.edu, AIAA Fellow.

Our goal in this work is to enhance data-driven reduced-order modeling to overcome the aforementioned challenges and to thus enable the construction of accurate and predictive ROMs for complex problems. We propose enhancing the ROM learning task with a preprocessing step in which we filter the training data prior to learning the reduced model. Using filtering in numerical simulations is not a new idea and has been considered in several settings in past research. For example, in the context of reduced-order modeling of computational fluid dynamics (CFD) problems, [2] proposed a data-driven filtered ROM, as follows. In the first step, an intrusive Galerkin projection is used to filter the nonlinear PDE and construct a filtered ROM. In the second step, a data-driven model is used as a closure to the filtered ROM by using a quadratic ansatz that models the interaction between the resolved and unresolved modes. The coefficients of the closed filtered ROM are found by solving an optimization problem that minimizes the difference between the FOM data and the ansatz. In [3], a robust principal component analysis (RPCA) filter [4] was employed to eliminate experimental noise from PIV measurements prior to constructing ROMs via dynamic mode decomposition (DMD) [5, 6] or proper-orthogonal decomposition (POD) [7]. In the context of laser Doppler vibrometry experiments conducted to identify structural faults in frescoes, [8] employed spatial filtering to remove noise due to surface irregularities that affect the direction in which the incident laser beam is reflected prior to perform POD. In [9], in the context of a 3D turbulent flow behind a circular cylinder, a spatial low-pass filter was employed to precondition the snapshots prior to constructing POD modes with the goal of separating the structures in the flow based on size. Still in the context of turbulence and scale separation but in fusion plasmas, [10] proposed using a space-filter approach to derive coarse-grained equations in configuration space for a quasi-neutral hybrid-kinetic plasma model. In addition, in certain complex simulation scenarios such as the gyrokinetic Vlasov equation, discretization schemes such as finite differences can introduce non-physical effects. To this end, [11] proposed adding hyperdiffusion terms in the gyrokinetic Vlasov equation, acting as a numerical filter, to cancel or mitigate these effects.

Rather than focusing on specific applications or scenarios, we consider filtering here as a generic tool for making training data more amenable to data-driven reduced-order modeling. While filtering will eliminate information from the snapshot training data, it will not necessarily compromise the accuracy of the resulting learned ROM—in fact, we will show that in some cases it leads to improved performance, particularly in avoiding overfitting. To learn data-driven ROMs, we employ the Operator Inference (OpInf) approach proposed in [12] which we enhance with Gaussian filtering. Gaussian filtering or blurring acts as a low-pass filter, making it effective in removing discontinuities in the training data. We illustrate the effects of filtering in a one-dimensional periodic inviscid Burgers' equation with a solution characterized by a moving shock. We then consider a large-scale eddy simulation (LES) of a rotating detonation engine (RDE) [1] in which we have available only a a few hundred down-sampled snapshots. We show that our filtered OpInf approach reduces overfitting for this RDE problem, leading to improved ROM predictions beyond the training data. We note that to the best of our knowledge, this represents one of the first studies in which data-driven ROMs are constructed for realistic, large-scale LES RDE simulations.

The remainder of this paper is organized as follows. Section II introduces the problem setup and summarizes the OpInf approach that we use here to construct data-driven ROMs. Section III presents the contribution of this paper, which is to augment data-driven reduced-order modeling methods with snapshot filtering, and illustrates the effects of filtering using the one-dimensional periodic inviscid Burgers' equation. In Section IV, we demonstrate the usefulness of the proposed filtering approach in a challenging three-dimensional RDE problem with more than 3.8 million degrees of freedom. Section V concludes the paper.

II. Learning data-driven reduced-order models for large-scale systems via operator inference

This section first introduces the large-scale problem setup (Section II.A), then summarizes the operator inference approach used in this paper to derive non-intrusive data-driven ROMs (Section II.B).

A. Large-scale problem setup

We consider the large-scale system of nonlinear equations defined on the time domain $[t_\mathrm{i}, t_\mathrm{f}]$, with t_i the initial time and t_f the final time,

$$\frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t} = \mathbf{f}(\mathbf{q}, t), \quad \mathbf{q}(t_\mathrm{i}) = \mathbf{q}_\mathrm{init}, \tag{1}$$

where $\mathbf{q}(t) \in \mathbb{R}^n$ is the n-dimensional vector of state variables at time t, $\mathbf{q}_\mathrm{init}$ is a specified initial condition, and $\mathbf{f} : \mathbb{R}^n \times [t_\mathrm{i}, t_\mathrm{f}] \rightarrow \mathbb{R}^n$ is a nonlinear function the defines the time evolution of the system state. We write the governing equations (1) in the form of ordinary differential equations (ODEs), where for the target problems of interest, these

ODEs represent the large-scale system of discretized partial differential equations (PDEs) and the dimension n scales with the (large) dimension of the spatial discretization.

In this work, we target the derivation of predictive reduced-order models using a non-intrusive data-driven approach. We consider the setup where training data are available over the time interval $[t_{\mathrm{i}}, t_{\mathrm{tr}}]$ with $t_{\mathrm{tr}} < t_{\mathrm{f}}$ denoting the end of the training time horizon. We collect n_t snapshots over the training horizon by solving the high-fidelity model (1) and recording the high-dimensional state solution at n_t time instants. Note that in large-scale problems, $n \gg n_t$. The state solution at time t_k is referred to as the kth snapshot and is denoted $\mathbf{q}_k$. The snapshot matrix $\mathbf{Q}$ collects the snapshots and has snapshot $\mathbf{q}_k$ as its kth column:

$$\mathbf{Q} = \begin{bmatrix} | & | & & | \\ \mathbf{q}_1 & \mathbf{q}_2 & \dots & \mathbf{q}_{n_t} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n \times n_t}.$$

Given the snapshot matrix $\mathbf{Q}$, the goal is to learn non-intrusive data-driven ROMs. In the following, we will summarize the Operator Inference approach that will be used throughout this paper, noting that the presented filtering idea is readily applicable to other data-driven approaches such as dynamic-mode decomposition (DMD) [5, 6].

B. Learning data-driven reduced models via non-intrusive discrete operator inference

Operator inference (OpInf) learns a reduced model with polynomial structure from data, where the structure is specified by the underlying governing equations [12]. The first step is to compute the representation of the available snapshots in a low-dimensional subspace. This is achieved by first computing the thin singular value decomposition (SVD) of the snapshot matrix $\mathbf{Q}$:

$$\mathbf{Q} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top,$$

with $\mathbf{U} \in \mathbb{R}^{n \times n_t}$, $\mathbf{\Sigma} \in \mathbb{R}^{n_t \times n_t}$ and $\mathbf{V} \in \mathbb{R}^{n_t \times n_t}$. $\mathbf{\Sigma}$ is a diagonal matrix containing the singular values of $\mathbf{Q}$ in non-decreasing order $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{n_t}$, where σ_j denotes the jth singular value. The first $r \ll n$ columns of $\mathbf{U}$ (i.e., the left singular vectors corresponding to the r largest singular values) form the POD basis $\mathbf{U}_r \in \mathbb{R}^{n \times r}$. We then obtain the low-dimensional representation of the snapshots in the reduced-order subspace spanned by $\mathbf{U}_r$ by computing

$$\hat{\mathbf{Q}} = \mathbf{U}_r^\top \mathbf{Q} \in \mathbb{R}^{r \times n_t}.$$

First, we summarize the *time-continuous* or *semi-discrete* OpInf, in which the reduced operators of the reduced ODE system are learned by solving a linear least-squares problem. For example, for a ROM with quadratic form

$$\frac{\mathrm{d}\hat{\mathbf{q}}}{\mathrm{d}t} = \hat{\mathbf{A}}_{\mathrm{sd}}\hat{\mathbf{q}} + \hat{\mathbf{H}}_{\mathrm{sd}}\left(\hat{\mathbf{q}} \otimes \hat{\mathbf{q}}\right), \tag{2}$$

we must determine the reduced operators $\hat{\mathbf{A}}_{\mathrm{sd}} \in \mathbb{R}^{r \times r}$ and $\hat{\mathbf{H}}_{\mathrm{sd}} \in \mathbb{R}^{r \times r^2}$, where subscript *sd* stands for *semi-discrete*. Note that in the semi-discrete OpInf, we must also estimate the corresponding time derivatives of these projected snapshots, $\frac{d}{dt}\hat{\mathbf{Q}} \in \mathbb{R}^{r \times n_t}$. OpInf determines the reduced operators that best match the projected snapshot data in a minimum residual sense by solving the linear least squares problem

$$\underset{\hat{\mathbf{A}}_{\mathrm{sd}}, \hat{\mathbf{H}}_{\mathrm{sd}}}{\operatorname{argmin}} \left\| \hat{\mathbf{Q}}^\top \hat{\mathbf{A}}_{\mathrm{sd}}^\top + \left(\hat{\mathbf{Q}} \otimes \hat{\mathbf{Q}}\right)^\top \hat{\mathbf{H}}_{\mathrm{sd}}^\top - \frac{d}{dt}\hat{\mathbf{Q}}^\top \right\|_F^2 + \lambda_1 \left\|\hat{\mathbf{A}}_{\mathrm{sd}}\right\|_F^2 + \lambda_2 \left\|\hat{\mathbf{H}}_{\mathrm{sd}}\right\|_F^2, \tag{3}$$

where F denotes the Frobenius norm. Note that, following [13], Tikhonov regularization is introduced to (3) to avoid overfitting and to account for model misspecification and numerical noise in the estimated time derivatives, with $\lambda_1, \lambda_2 \in \mathbb{R}$ the scalar regularization hyper-parameters.

In certain scenarios, however, it can be difficult to accurately approximate the time derivatives of the projected snapshots, $\frac{d}{dt}\hat{\mathbf{Q}}$, in (3). This is true for applications in which the large-scale simulation code that generates the training data employs a small time step that prohibits saving every snapshot to disk due to the storage requirements. An inaccurate approximation of the time-derivative introduces noise in the OpInf least-squares learning problem, which in turn can lead to an inaccurate ROM. In such cases, we employ the *discrete* version of OpInf, which is used to learn the reduced operators of a difference ROM. For a quadratic difference model

$$\hat{\mathbf{q}}[k+1] = \hat{\mathbf{A}}_{\mathrm{d}}\, \hat{\mathbf{q}}[k] + \hat{\mathbf{H}}_{\mathrm{d}}\left(\hat{\mathbf{q}}[k] \otimes \hat{\mathbf{q}}[k]\right), \tag{4}$$

where $\hat{\mathbf{q}}[k]$ denotes the discrete reduced state at time t_k and subscript d stands for *discrete*, the reduced operators of the discrete OpInf model are learned by solving

$$\underset{\hat{\mathbf{A}}_\mathrm{d},\hat{\mathbf{H}}_\mathrm{d}}{\operatorname{argmin}} \left\| \hat{\mathbf{Q}}_1^\top \hat{\mathbf{A}}_\mathrm{d}^\top + \left(\hat{\mathbf{Q}}_1 \otimes \hat{\mathbf{Q}}_1\right)^\top \hat{\mathbf{H}}_\mathrm{d}^\top - \hat{\mathbf{Q}}_2^\top \right\|_F^2 + \lambda_1 \left\|\hat{\mathbf{A}}_\mathrm{d}\right\|_F^2 + \lambda_2 \left\|\hat{\mathbf{H}}_\mathrm{d}\right\|_F^2, \tag{5}$$

where

$$\hat{\mathbf{Q}}_1 = \begin{bmatrix} | & | & & | \\ \hat{\mathbf{q}}_1 & \hat{\mathbf{q}}_2 & \dots & \hat{\mathbf{q}}_{n_t-1} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{r \times n_t - 1} \quad \text{and} \quad \hat{\mathbf{Q}}_2 = \begin{bmatrix} | & | & & | \\ \hat{\mathbf{q}}_2 & \hat{\mathbf{q}}_3 & \dots & \hat{\mathbf{q}}_{n_t} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{r \times n_t - 1}. \tag{6}$$

Note that in the case of a linear system, the discrete OpInf formulation is equivalent to DMD [12].

III. Enhancing non-intrusive data-driven reduced-order modeling with filtering

This section describes the contribution of this paper, which is to introduce snapshot filtering to data-driven reduced-order modeling methods. We first give an overview and technical motivation for the approach (Section III.A) and then describe Gaussian filtering (Section III.B). We present one specific algorithm that instantiates the ideas, using the OpInf model reduction approach with a Gaussian filter (Section III.C). We conclude the section through an illustrative example that is challenging for POD-based model reduction strategies, the inviscid Burgers' equation (Section III.D).

A. Motivation for using filtering in data-driven reduced-order modeling

Consider the setting where the goal is to train a data-driven ROM from simulation data stemming from one of the following scenarios: (i) simulation data characterized by discontinuities or sharp interfaces (e.g, shocks, phase-field models); (ii) simulation data with a broadband or high frequency content; (iii) simulation data that contains noise due to, e.g., the underlying discretization scheme (for example, particle-in-cell methods). These examples all represent situations for which the construction of accurate and predictive data-driven ROMs is known to be challenging. This challenge is exacerbated when the amount of simulation data available is limited, which is often the case for realistic problems that go beyond the typical one-dimensional and two-dimensional academic problems.

Instead of training the ROMs directly using the given simulation data, we first filter the simulation data (in space) and subsequently use the filtered data to train the reduced model. This idea is depicted in Figure 1. The goal of filtering is to make simulation data more amenable to training data-driven ROMs. If, for example, the training snapshots have discontinuities, filtering will make the data smoother. While this filtering eliminates information from the snapshot training data, it will not necessarily compromise the accuracy of the resulting learned ROM—in fact, we will show that in some cases it leads to improved performance. This is because it is generally impossible for a ROM to capture the entire frequency spectrum of the training data. The reduction from n to r degrees of freedom occurs because only the dominant dynamics (typically the lower-frequency, large-scale structures) are retained in the ROM. Thus the filtering is a preprocessing step that attenuates (or eliminates) frequency content that would otherwise be difficult or impossible to capture with the reduced model. In doing so, the inference problem to learn the ROM becomes numerically better conditioned, less sensitive to numerical errors in the training data, and less prone to overfitting when the amount of training data is limited.

B. Gaussian filtering

Consider a continuous solution field $s(x, y, z)$, where $[x, y, z] \in \Omega$ are the spatial coordinates (here with $d = 3$ dimensions) over the problem domain Ω. The field s could be, for example, the instantaneous pressure or temperature distribution over the problem domain. The continuous form of the Gaussian filter in d spatial dimensions is given by

$$G(x_1, \dots, x_d; \tau_1, \dots, \tau_d) = \frac{1}{(\sqrt{2\pi})^d \prod_{i=1}^d \tau_i} \exp\left(-\sum_{i=1}^{d} x_i^2/2\tau_i^2\right),$$

where x_i denotes the ith spatial coordinate direction. For $d = 3$ spatial dimensions, we have $[x_1, x_2, x_3] = [x, y, z]$ and

$$G(x, y, z; \tau_1, \tau_2, \tau_3) = \frac{1}{(\sqrt{2\pi})^3 \tau_1 \tau_2 \tau_3} \exp\left(-x^2/2\tau_1^2 - y^2/2\tau_2^2 - z^2/2\tau_3^2\right).$$

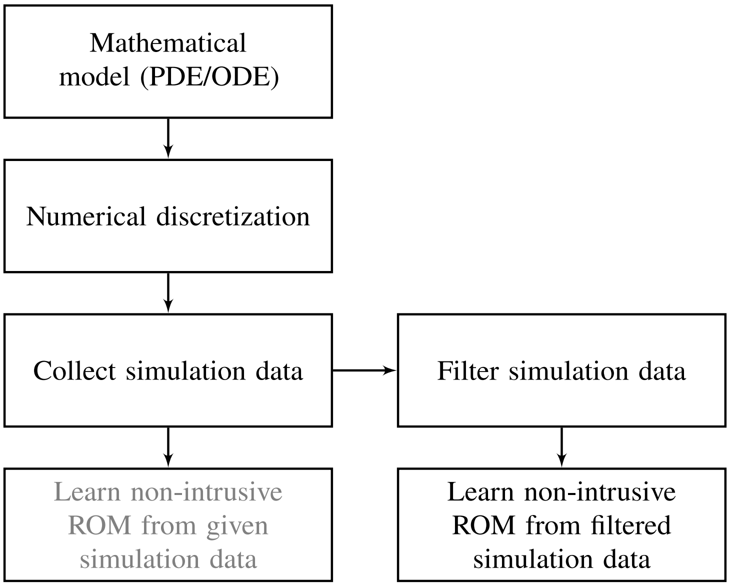
Mathematical model (PDE/ODE)
Numerical discretization
Collect simulation data
Filter simulation data
Learn non-intrusive ROM from given simulation data
Learn non-intrusive ROM from filtered simulation data

Fig. 1 On the left, we summarize the general steps employed in standard non-intrusive data-driven reduced-order modeling in which the reduced models are trained directly from simulation data. In contrast, in our proposed approach, instead of training the ROMs directly from the simulation data, we first filter the data and train the reduced model on the filtered data, as shown on the right of the figure.

Here, τ_i is the standard deviation, also referred to as the filter width, in the ith spatial direction. In general, one can select different filter widths for different directions. The Gaussian filter is defined over a finite filtering domain, with R_i the filter radius in the ith spatial dimension. This filter radius defines the size of the neighborhood over which the filtering is applied. In this work, we set $R_i = 4\tau_i$, which corresponds to employing 0.9999% of the full Gaussian profile. We denote the filtered version of s as $\tilde{s}$:

$$\tilde{s} = G * s, \tag{7}$$

where $*$ denotes convolution in space. In practice, Gaussian filtering is applied to the spatially discretized snapshots, implemented as a convolution sum over grid points falling within the filter radius.

The filter depends on hyper-parameters $(\tau_1, \tau_2, \ldots, \tau_d)$ whose values influence the amount of performed filtering: the larger the value of τ_i, the more filtering is conducted in direction x_i. Note that since the d-variate Gaussian density used to define the kernel is the product of d one-dimensional Gaussian densities, i.e.,

$$G(x_1, \ldots, x_d; \tau_1, \ldots, \tau_d) = \frac{1}{(\sqrt{2\pi})^d \prod_{i=1}^{d} \tau_i} \exp\left(-\sum_{i=1}^{d} x_i^2/2\tau_i^2\right) = \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi}\tau_i} \exp\left(-x_i^2/2\tau_i^2\right),$$

the d-dimensional filter can be implemented as a sequence of d one-dimensional convolution filters. There are several readily available Gaussian filtering implementations. Here, we employ the implementation provided by the function `scipy.ndimage.gaussian_filter`[*] from the `scipy.ndimage` module in `python`.

We filter each snapshot $\mathbf{q}_k \in \mathbb{R}^n$ in the given snapshot matrix $\mathbf{Q}$ separately. For each snapshot, we separate out the different physical quantities (temperature, pressure, velocity, etc.) and we filter each physical field separately. We then recombine the physical quantities to obtain a filtered version of the snapshot. Upon filtering all n_t snapshots, we obtain the filtered snapshot matrix which we denote by $\mathbf{F} \in \mathbb{R}^{n \times n_t}$.

Since Gaussian filtering smooths the underlying data, it is effective in removing discontinuities in the training data. Moreover, it is also effective in eliminating high-frequency content in data-sets stemming from complex physics simulations, such as reactive flows. We note nevertheless that other filters, such as total variation filtering [14, 15] or robust PCA filtering [4] can be employed as well.

[*] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.gaussian_filter.html`

C. Operator Inference with filtered training data

In this section, we present one specific algorithm that instantiates the idea of filtering using the OpInf model reduction approach summarized in Section II.B together with a Gaussian filter. In Section II.B, we have presented both the semi-discrete (3) and discrete (5) versions of OpInf. However, the only algorithmic difference between two OpInf learning problems consists of the projected data matrices used on the left- and right-hand sides of the corresponding least-square minimization problems: in the semi-discrete version, the left-hand side depends on $\hat{\mathbf{Q}}$ and the right-hand side is $\frac{d}{dt}\hat{\mathbf{Q}}$, whereas in the discrete variant we have $\hat{\mathbf{Q}}_1$ on the left-hand side and $\hat{\mathbf{Q}}_2$ on the right-hand side, with $\hat{\mathbf{Q}}_1$ and $\hat{\mathbf{Q}}_2$ defined in (6). Therefore, without loss of generality, we present here a generic OpInf algorithm with filtering in which the pair $(\hat{\mathbf{L}}, \hat{\mathbf{R}})$ is used to generally denote the left- and right-hand side projected matrices.

We summarize the steps in Algorithm 1. The inputs are the original snapshot matrix, $\mathbf{Q}$, the Gaussian filter widths $\tau_1, \tau_2, \ldots, \tau_d$ and the value of the reduced dimension, r. In the first step, we apply Gaussian filtering to the snapshot matrix $\mathbf{Q}$ and obtain the filtered snapshot matrix $\mathbf{F}$, noting that if there are multiple physical variables (e.g., velocity, pressure, etc.), we filter them separately. From hereon, all computations are in terms of $\mathbf{F}$. We determine the rank-r POD basis $\mathbf{U}_r$ from the singular value decomposition of the filtered matrix $\mathbf{F}$. Note that the rank-r POD basis truncation is a form of filtering as well, whereby the (typically high-frequency) modes corresponding to the singular values smaller than the rth singular value are filtered out. Once we have determined the POD basis, we project the filtered snapshot matrix on the subspace spanned by the column vectors of $\mathbf{U}_r$ to obtain $\hat{\mathbf{F}} = \mathbf{U}_r^\top \mathbf{F}$. In the next step, we determine, from $\hat{\mathbf{F}}$, the pair $(\hat{\mathbf{L}}, \hat{\mathbf{R}})$ used in the OpInf learning problem. Additionally, we determine the Tikhonov regularization parameters λ_1 and λ_2; an effective strategy was proposed in [13]. At this point, the OpInf learning problem (3) is fully specified, and we solve it to find the reduced linear and quadratic operators, denoted $\hat{\mathbf{A}}$ and $\hat{\mathbf{H}}$, which are returned upon termination.

Algorithm 1 Operator Inference with filtered training data.

```
Input: Snapshot data $\mathbf{Q} \in \mathbb{R}^{n \times n_t}$, Gaussian filtering hyper-parameters $\tau_1, \tau_2, \ldots, \tau_d$ and reduced dimension $r \in \mathbb{N}$
Result: Reduced operators $\hat{\mathbf{A}}, \hat{\mathbf{H}}$
Filter snapshot matrix $\mathbf{Q}$ with Gaussian filtering depending on $\tau_1, \tau_2, \ldots, \tau_d$ to obtain the filtered snapshot matrix $\mathbf{F}$
    ▷ in problems with multiple physical variables (e.g., pressure, velocity, etc.), we filter each variable separately
Compute rank-$r$ POD basis $\mathbf{U}_r$ from the SVD of the filtered snapshot matrix $\mathbf{F}$    ▷ POD truncation
Project snapshots onto the $r$-dimensional POD subspace $\hat{\mathbf{F}} \leftarrow \mathbf{U}_r^\top \mathbf{F}$
Determine the pair $(\hat{\mathbf{L}}, \hat{\mathbf{R}})$ used in the OpInf learning problem from the projected filtered snapshots $\hat{\mathbf{F}}$
Determine the Tikhonov regularization parameters $\lambda_1$ and $\lambda_2$ (see [13, 16])
Solve the corresponding regularized learning problem ((3) or (5)) to find reduced model operators $\hat{\mathbf{A}}, \hat{\mathbf{H}}$
return $\hat{\mathbf{A}}, \hat{\mathbf{H}}$
```

The filtering hyper-parameters $\tau_1, \tau_2, \ldots, \tau_d$ are user-specified input parameters in Algorithm 1. Nevertheless, their values can be tuned via, e.g., a grid search similar to the procedure used to determines the Tikhonov regularization parameters λ_1 and λ_2. This hyper-parameters search can added as an outer-loop around the regularization parameters tuning procedure. We note, however, that both these procedures can be computationally expensive if the given data set is large. Efficient tuning strategies of the filtering hyper-parameters will be addressed in our future research.

D. Illustrative example: periodic one-dimensional inviscid Burgers' equation

We illustrate the proposed idea in a model problem that is generally challenging for POD-based model reduction. We consider the inviscid Burgers' equation defined on the periodic 1D spatial domain $[0, 1)$ and time domain $[0, 1]$:

$$\frac{\partial q}{\partial t} + \frac{1}{2}\frac{\partial q^2}{\partial x} = 0, \quad q(0, x) = 1/2 + \sin(2\pi x). \tag{8}$$

The nonlinearity due to the convective term leads to the formation of a shock after a short time that travels from left to right. Due to the advective nature of the shock, the snapshot matrix will have slowly decaying singular values, which in turn means that many POD basis vectors must be retained to achieve acceptable ROMs. Furthermore, the shock discontinuity leads to Gibbs oscillations in the reconstructed ROM solutions. Using this example, we show that filtering reduces the accuracy of the discontinuity representation but leads to faster decaying singular values and attenuated Gibbs oscillations in the reconstructed solution.

To solve (8) numerically, we use finite volume solvers from the Clawpack framework [17, 18]. We use $n = 500$

points for spatial discretization. The semi-discrete version of (8) is quadratic, i.e.,

$$\frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t} = \mathbf{H}\left(\mathbf{q}\otimes\mathbf{q}\right), \quad \mathbf{q}_0 = \mathbf{q}(0).$$

We collect $n_t = 1,001$ snapshots every 10^{-3} seconds. We employ the semi-discrete (time-continuous) OpInf procedure (3) and use it to learn the reduced operator $\hat{\mathbf{H}}_{\mathrm{sd}} \in \mathbb{R}^{r\times r^2}$ of the structure-preserving quadratic ROM

$$\frac{\mathrm{d}\hat{\mathbf{q}}}{\mathrm{d}t} = \hat{\mathbf{H}}_{\mathrm{sd}}\left(\hat{\mathbf{q}}\otimes\hat{\mathbf{q}}\right), \quad \hat{\mathbf{q}}_0 = \mathbf{U}_r\mathbf{q}_0, \tag{9}$$

where $\mathbf{U}_r$ is the corresponding POD reduced basis of order r. We employ all available 1,001 snapshots for training and use the OpInf reduced model to reconstruct the training data. We consider three widths in Gaussian filtering: (i) $\tau = 0.5$, (ii) $\tau = 2.0$ and (iii) $\tau = 10.0$. We note that the Gaussian filter was implemented such that it preserves the periodic boundary condition. In all performed experiments, we have estimated the time derivatives of the projected snapshots $\frac{d}{dt}\hat{\mathbf{Q}} \in \mathbb{R}^{r\times 1,001}$ in the OpInf learning problem (3) via a fourth-order forward difference scheme.

First, we analyze the effect of filtering on the decay of the singular values of the filtered snapshot matrix. We visualize the singular values normalized by their sum in Figure 2, on the left. Observe the faster decay with more filtering: a small value of the filter width has little effect on the data and therefore on the singular values as well. In contrast, using $\tau = 2.0$ leads to a faster decay, whereas $\tau = 10.0$ leads to a very fast decay. As a consequence, more filtering requires fewer modes to retain the same amount of energy. For example, to retain 99.5% of the total energy, $r = 25$ modes are needed when no filtering is performed. To retain the same amount of energy when filtering using width $\tau = 0.5$, $r = 23$ modes are necessary; $r = 25$ modes retain 99.58% of the total energy. Filtering using $\tau = 2.0$ decreases the number of modes needed to retain 99.5% of the total energy to $r = 16$, whereas $r = 25$ modes retain 99.85% of the energy. Finally, when filtering using the largest width, $\tau = 10.0$, we need only $r = 8$ POD modes to retain 99.5% of the total energy, while $r = 25$ retain more than 99.9999%.

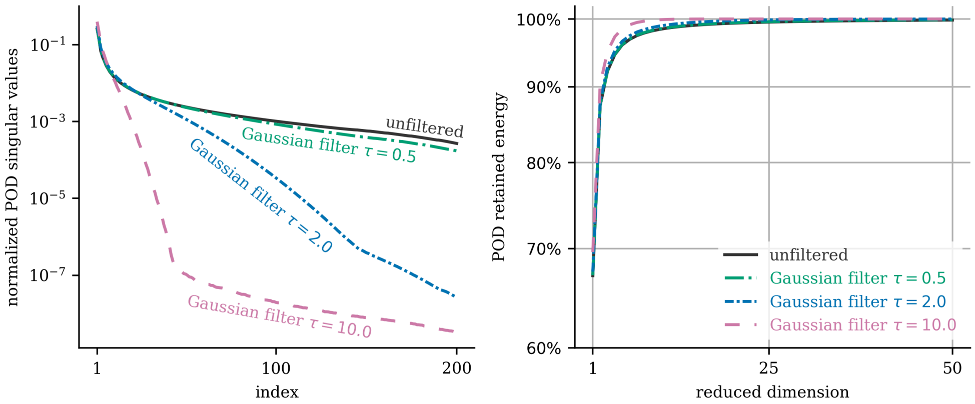

Fig. 2 Inviscid Burgers' equation (8): normalized singular values (left) and POD retained energy (right).

While it is not surprising that filtering leads to more rapid singular value decay, the key question is whether the approximation introduced to the training data has a negative effect on the accuracy inferred OpInf ROM. To assess this question, we train OpInf reduced models using both unfiltered and filtered data. In all experiments, the reduced models have reduced dimension $r = 25$. We visualize, on the top part in Figure 3, the original and the filtered state data at three selected time steps: $t = 0.050$ (before the shock is formed; depicted on the left), $t = 0.500$ (after the shock formed; depicted in the center) and $t = 0.800$ (when the shock is close to the right boundary; depicted on the right). Since filtering using $\tau = 0.5$ has an insignificant effect on the data, for an easier visualization we plot only the results corresponding to $\tau = 2.0$ and $\tau = 10.0$. The choice $\tau = 2.0$ leads to little spatial filtering, i.e., the shock is slightly smoothed but it remains sharp. In contrast, as expected, $\tau = 10.0$ smoothens the shock significantly. Note that intuitively, Gaussian filtering here has a similar effect to adding a viscosity term into the underlying model. In the bottom part of Figure 3, we depict the reconstructed solutions from the OpInf reduced model simulations. Without filtering, the reduced model solution exhibits severe Gibbs oscillations around the shock. Slightly smoothing the shock

Downloaded by Ionut Farcas on June 25, 2022 | http://arc.aiaa.org | DOI: 10.2514/6.2022-3487

using width $\tau = 2.0$ decreases the amplitudes of the oscillations and leads overall to a small increase in the accuracy of the solution. Filtering using $\tau = 10.0$ leads to an highly accurate OpInf reconstruction of the filtered data and a solution that does not exhibit Gibbs oscillations; however, this reconstruction is rather inaccurate with respect to the original unfiltered data. We therefore see that filtering trades off smoothing with accuracy. In some application settings, the smoother $\tau = 10.0$ reduced model may be preferable and better behaved, even though it has introduced inaccuracies in the representation of the shock.

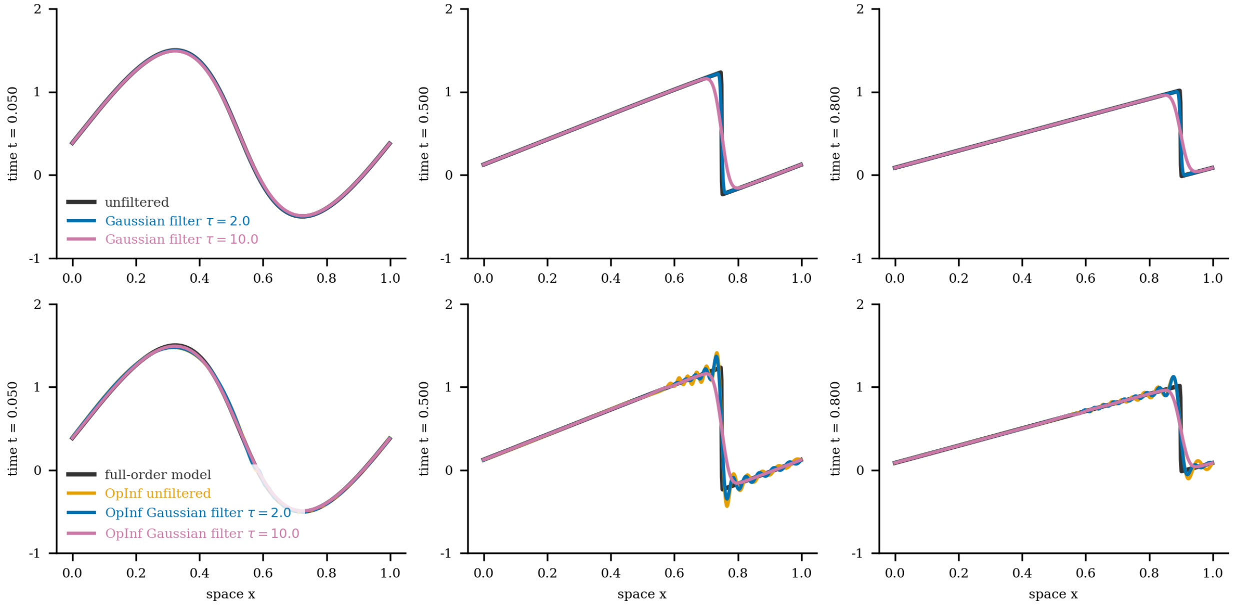

Fig. 3 Inviscid Burgers' equation (8)**. Top: original and filtered state data at three selected time steps:** $t = 0.050$ **(left),** $t = 0.500$ **(center) and** $t = 0.800$ **(right). Bottom: the corresponding OpInf reduced model simulations.**

Figure 4 plots the relative errors

$$\varepsilon(\mathbf{Q}_{\text{FOM}}, \mathbf{Q}_{\text{ROM}}) = \frac{\|\mathbf{Q}_{\text{FOM}} - \mathbf{Q}_{\text{ROM}}\|_F^2}{\|\mathbf{Q}_{\text{FOM}}\|_F^2} \tag{10}$$

for reduced dimensions $r \leq 25$, where $\mathbf{Q}_{\text{FOM}}$ denotes the original FOM data and $\mathbf{Q}_{\text{ROM}}$ denotes the reconstruction of the solution from the ROM simulations. We observe that filtering using $\tau = 2.0$ leads to a small increase in accuracy since it decreases the amplitude of Gibbs oscillations around the shock. In contrast, filtering using $\tau = 10.0$ yields a less accurate reduced model for which the relative error saturates around $r = 15$. This saturation is not surprising since $r \geq 15$ POD modes retain more than 99.99% of the total energy of the corresponding filtered snapshot matrix. The remaining relative error (at a value of approximately 0.02) is due to the difference between the filtered and unfiltered snapshot data.

IV. Filtered reduced-order modeling for a rotating detonation engine simulation

This section first describes the problem setup for a challenging three-dimensional rotating detonation engine (RDE) simulation (Section IV.A) and the associated setup for our data-driven reduced-order modeling (Section IV.B). We apply the proposed filtering approach to derive reduced models in Section IV.C.

A. Problem setup for high-fidelity large-eddy simulation

We consider reduced-order modeling for the RDE problem presented in [1]. The physics of the problem are modelled using the Navier-Stokes equations and a detailed chemistry model. Turbulence is modelled using a one-equation transported eddy viscosity. The chemistry employs a modified version of the Westbrook-Dryer mechanism, originally developed for methane-air combustion, but modified here for the purposes of methane-oxygen detonation. As described

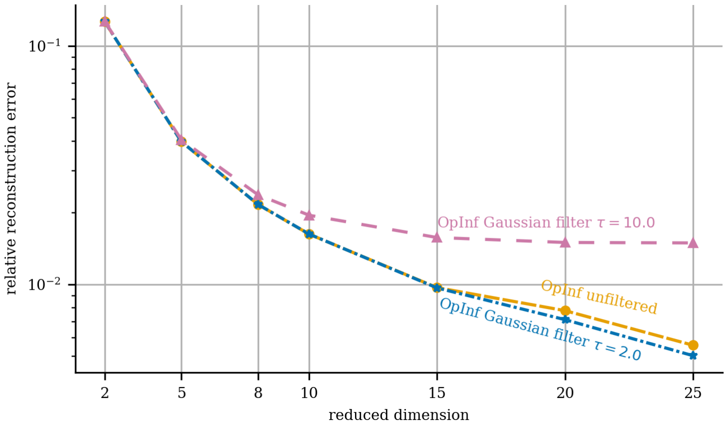

Fig. 4 Inviscid Burgers' equation (8)**: relative reconstruction error** (10) **of the training set using OpInf.**

in [1], the modification is implemented so as to retrieve the Chapman-Jouguet detonation speed in a one-dimensional combustion problem. This mechanism contains a total of six species and three unique elementary reactions.

The computational domain considers a 45-degree sector of the full RDE in which a single wave travels stably in one direction. The sector has nine injectors. Spatial discretization uses a second-order McCormack finite-volume scheme and a second-order central difference viscous term. The results are evolved in time using a second order McCormack scheme as well. Time steps were computed dynamically. During the steady cycle, the time step takes on values around 1.2ns. Parallelized over $1,672$ cores, 2.0 ms of real-time simulation (which includes a start-up transient, stabilization of the cycle, and several complete cycles of the quasi-steady state) required $100,000$ CPU hours on the Onyx high-performance computer, a part of the DoD Supercomputing Resource Center (DSRC)[1]. We depict the full RDE simulation domain on the left in Figure 5. Here, we build ROMs for the combustion chamber only, which has spatial dimensions $1 < x < 76$ mm, and y and z such that the radius $R = \sqrt{y^2 + z^2} < 38.1$ mm. The original simulation data, performed using a non-orthogonal grid, has been interpolated onto a structured grid. This results in a computational domain for the combustion chamber that has $n_x = 314,874$ spatial degrees of freedom, visualized in Figure 5 (right).

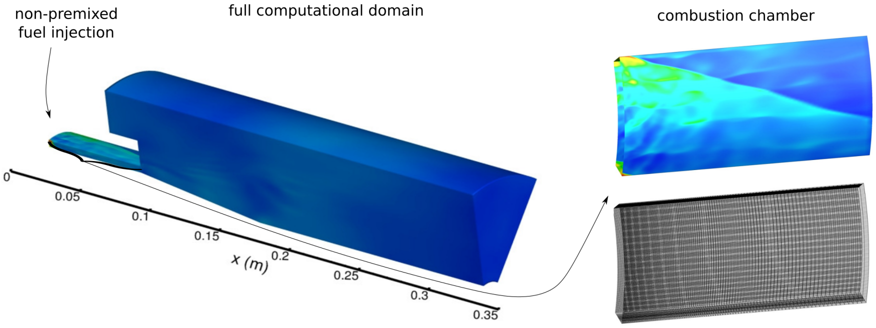

Fig. 5 Left: full simulation domain of the 45**-degree RDE sector that has been studied in [1]. In this work, we are constructing data-driven ROMs for the combustion chamber only, depicted on the top right, which has spatial dimensions** $1 < x < 76$ **mm and** y **and** z **such that the radius** $R = \sqrt{y^2 + z^2} < 38.1$ **mm. The computational domain used here, visualized on the bottom right, contains** $n_x = 314,874$ **spatial degrees of freedom.**

B. Filtered discrete operator inference

We have 441 snapshots available in the quasi-steady state regime, from time $t = 9.9338 \times 10^{-4}$ seconds to $t = 1.3756 \times 10^{-3}$ seconds. This is approximately 0.4 ms of real-time, which corresponds to roughly 22 passages of rotating detonation waves. We are only considering a 45-degree spatial sector in this calculation; periodicity implies that there are eight waves present in the complete annulus. The time interval of 0.4 ms represents nearly three revolutions of the entire wave system. The available snapshots have been down-sampled by a factor of about 720 from the original CFD simulations, resulting in an average step size between the down-sampled snapshots of 8.6691×10^{-7} seconds.

The large down-sampling factor renders numerical approximations of the time derivative of the projected data matrix appearing in the right-hand size in the time-continuous OpInf learning problem (3) inaccurate and noisy, which in turn results in an inaccurate OpInf reduced model. To this end, we construct discrete non-intrusive data-driven ROMs (4) and learn the reduced operators of the difference model via (5). Following the work in [13, 19], we build the ROMs using the specific volume flow variables, species mass fractions and temperature. That is, we transform the data snapshots to represent the following 12 state variables:

$$q = [1/\rho \;\; p \;\; v_x \;\; v_y \;\; v_z \;\; w_{\mathrm{CH_4}} \;\; w_{\mathrm{O_2}} \;\; w_{\mathrm{H_2O}} \;\; w_{\mathrm{CO_2}} \;\; w_{\mathrm{H_2}} \;\; w_{\mathrm{CO}} \;\; T]^\top, \tag{11}$$

where ρ is density, p is pressure, v_x, v_y, v_z are the three velocity components, $w_{[\cdot]}$ are the species mass fractions of the six chemical species CH_4, O_2, H_2O, CO_2, H_2 and CO, and T is temperature. As discussed in [13, 19], this choice of variables renders many terms in the governing equations quadratic, which makes it appropriate to use the OpInf ROM with quadratic form (5). The dimension of our data snapshots is therefore $n = 314,874 \times 12 = 3,778,488$. Table 1 shows the minimal and maximal values of the state variables in the snapshots used for learning the ROM.

Table 1 Minimal and maximal values of quantities in high-fidelity snapshots.

state variable	minimum value	maximum value
specific volume $\zeta = \frac{1}{\rho}\left(\frac{m^3}{kg}\right)$	0.6285	6.3212
pressure p (MPa)	0.2140	1.1406
velocity v_x (m/s)	−64.8222	1397.0931
velocity v_y (m/s)	−531.4510	529.4986
velocity v_z (m/s)	−985.0364	812.2877
CH_4 mass fraction $w_{\mathrm{CH_4}}$	0.0000	0.5376
O_2 mass fraction $w_{\mathrm{O_2}}$	1.6077×10^{-6}	0.6163
H_2O mass fraction $w_{\mathrm{H_2O}}$	0.0767	0.4399
CO_2 mass fraction $w_{\mathrm{CO_2}}$	0.0045	0.2842
H_2 mass fraction $w_{\mathrm{H_2}}$	0.0008	0.0551
CO mass fraction w_{CO}	0.0650	0.5731
temperature T (K)	762.4429	3410.2900

We use the first $n_t = 294$ snapshots, i.e., the first two thirds of the available dataset, to train an OpInf reduced model. The training horizon ends at $t = 1.2488 \times 10^{-3}$ seconds. Note that since the twelve physical state variables have significantly different scales, we center and scale them variable-by-variable prior to performing OpInf: each state variable is first centered around the mean field (over the 294 training timesteps) in that variable, and then scaled by the maximum absolute value of that variable so that the values for each state variable do not exceed $[-1, 1]$. The remaining 147 snapshots beyond the training horizon are used as testing data to assess the predictive capability of the reduced model. We note that the complexity of this problem as well as the small size of the available data set make this an excellent scenario for ascertaining the predictive capabilities of data-driven reduced models in real-world problems.

To study the effect of filtering on the predictive power of OpInf in the RDE problem, we use two filter widths in Gaussian filtering: one that provides little spatial filtering, i.e., $\tau = 1.0$, and another that provides more filtering, $\tau = 5.0$. We filter the 12 physical variables in (11) separately using the same filter width τ in all three spatial dimensions.

C. Results

Filtering effect on the decay of the centered and scaled POD singular values. Figure 6 plots the normalized singular values (with respect to their sum) and POD retained energy corresponding to the centered and scaled training data set with and without Gaussian filtering. To retain 95% of the total energy of the unfiltered snapshot data, $r = 136$ modes are needed. The smoothing effect of Gaussian filtering leads to a faster decay of the singular values which in turn requires fewer modes to retain the same of amount of energy. When using filtering width $\tau = 5.0$, only 85 POD modes are needed to retain 95% of the total energy (a reduction of 36%).

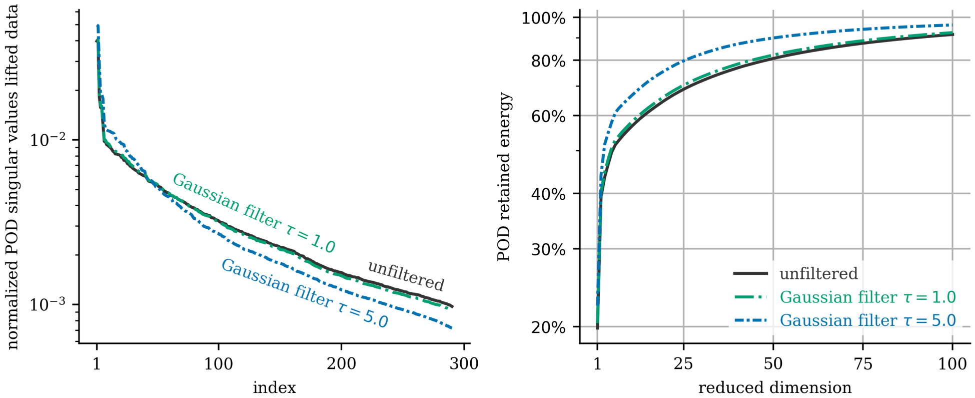

Fig. 6 RDE with 294 training snapshots: normalized singular values (left) and POD retained energy (right).

Filtering effect on the accuracy of the reduced model. We now assess the impact of filtering on the accuracy of the discrete OpInf reduced model. The small size of the training set (294 snapshots) limits the maximum reduced dimension of a quadratic OpInf reduced model to $r = 22$, because this sets the maximum number of operator coefficients that can be inferred via the OpInf regression problem. In the following we study the accuracy of the OpInf reduced models for reduced dimensions $r \leq 22$. Without filtering, 22 modes retain only 66.7% of the total energy, whereas filtering increases this amount up to 77.6% when $\tau = 5.0$.

Figure 7 plots the relative errors (10) in the ROM predictions of the training set for four state variables: pressure (upper left), temperature (upper right), fuel (CH_4) mass fraction (bottom left) and oxidizer (O_2) mass fraction (bottom right). Overall, with or without filtering, the OpInf reduced model simulated solutions accurately predict the training set, with errors smaller than 6%. Using the smaller filtering width $\tau = 1.0$ has almost no effect on the accuracy of the discrete OpInf model, except for reduced dimension $r = 19$ where we observe a small improvement. Using the larger filtering width $\tau = 5.0$ introduces more error into the ROM predictions of the training set, because of the smoothing approximations introduced into the snapshot set.

We next use the OpInf ROMs to simulate into the prediction regime. Figure 8 plots the resulting relative solution errors (10). All prediction errors plotted in Figure 8 are smaller than 8%, indicating that the OpInf ROMs with reduced dimensions $r \leq 22$ provide accurate predictions in the complex scenario under consideration. For pressure prediction we observe a clear improvement when using more filtering. For example, for reduced dimension $r = 22$, the error is 44% smaller than when using the unfiltered data to train the reduced model. Moreover, we see that while filtering does not necessarily improve the prediction power of the reduced model for the other three state variables, performing more filtering using $\tau = 5.0$ leads to a smoother error. In contrast, the errors corresponding to the other two cases show some deterioration for reduced dimensions $r \geq 19$. These results illustrate that one benefit of filtering is to prevent overfitting of the reduced model, which is particularly important with limited snapshot data as we have here. Although the filtered reduced model showed a small increase in error over the training regime, it provides a more predictive reduced model beyond the training horizon.

Pressure traces at two selected spatial probes. To further illustrate the enhanced prediction accuracy of the reduced model trained using data filtered with width $\tau = 5.0$, Figure 9 plots pressure traces at two selected spatial probes within the combustion chamber. The probes have coordinates $(x, y, z) = (0.03, 0.035, 0.000)$ for probe 1 and $(x, y, z) = (0.04, 0.035, 0.000)$ for probe 2. The reduced dimension of the OpInf reduced models is $r = 22$. The vertical dashed line at $t = 1.2488 \times 10^{-3}$ seconds separates the training from the prediction regimes. We depict the predicted pressure trace values using dashed lines. We see that both OpInf ROM solutions match well the frequency of the FOM pressure traces. However, as observed in the prediction error figure (Figure 8), filtering using width $\tau = 5.0$ leads to more accurate predictions, which here translates into a more accurate matching of amplitudes.

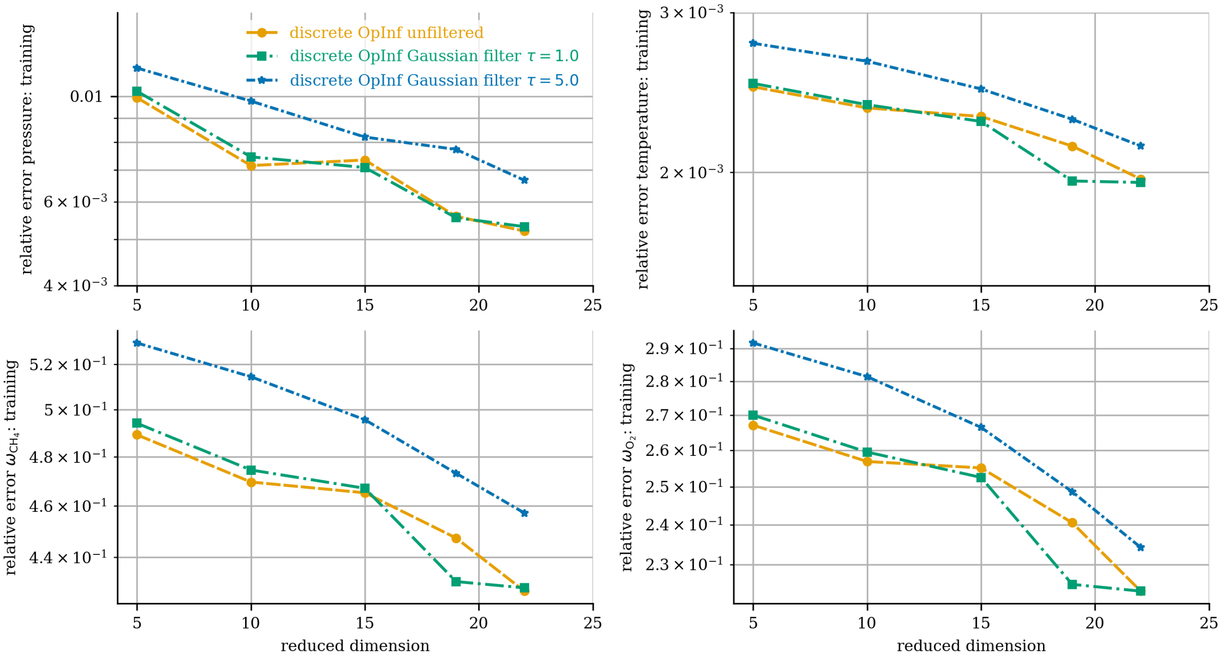

Fig. 7 Relative ROM reconstruction error (10) **of the training set** (294 **snapshots) for four state variables: pressure (top left), temperature (top right),** CH_4 **mass fraction (bottom left) and** O_2 **mass fraction (bottom right).**

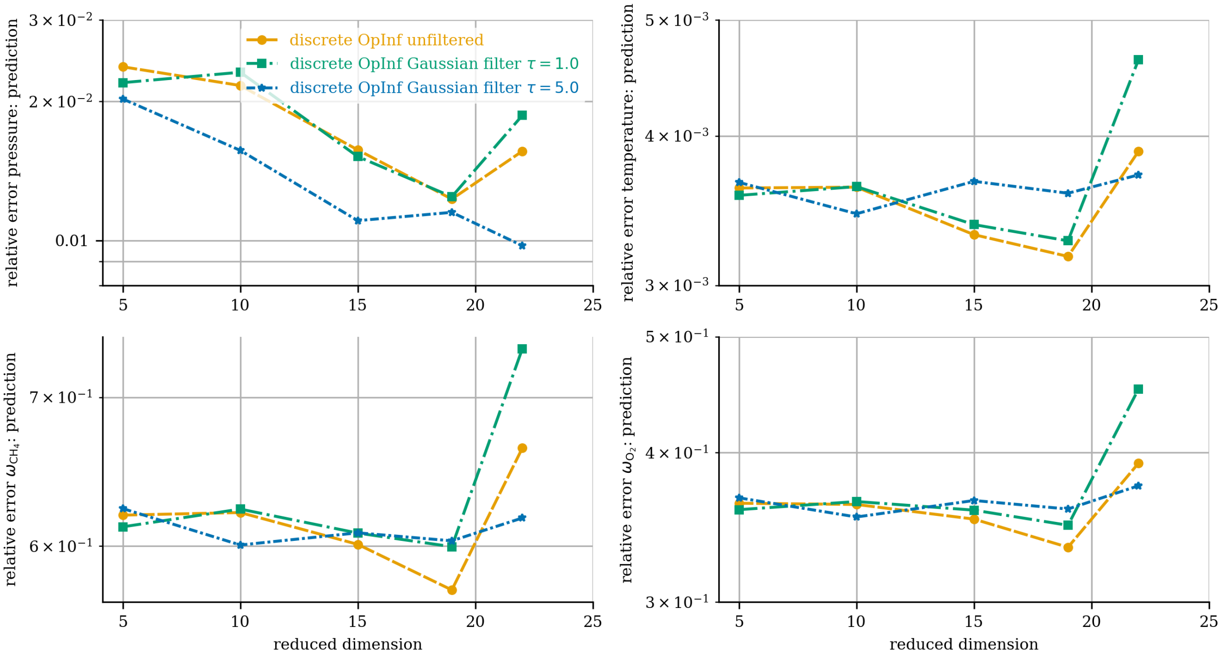

Fig. 8 Relative ROM solution errors (10) **in the prediction regime** (147 **snapshots) for four state variables: pressure (top left), temperature (top right),** CH_4 **mass fraction (bottom left) and** O_2 **mass fraction (bottom right).**

Pressure field at a selected cross section. We visualize the ROM pressure field solution at the cross section located at radius $R = 37$ mm, which is a cylindrical sector located about 1 mm within the outer radius of the combustion chamber.

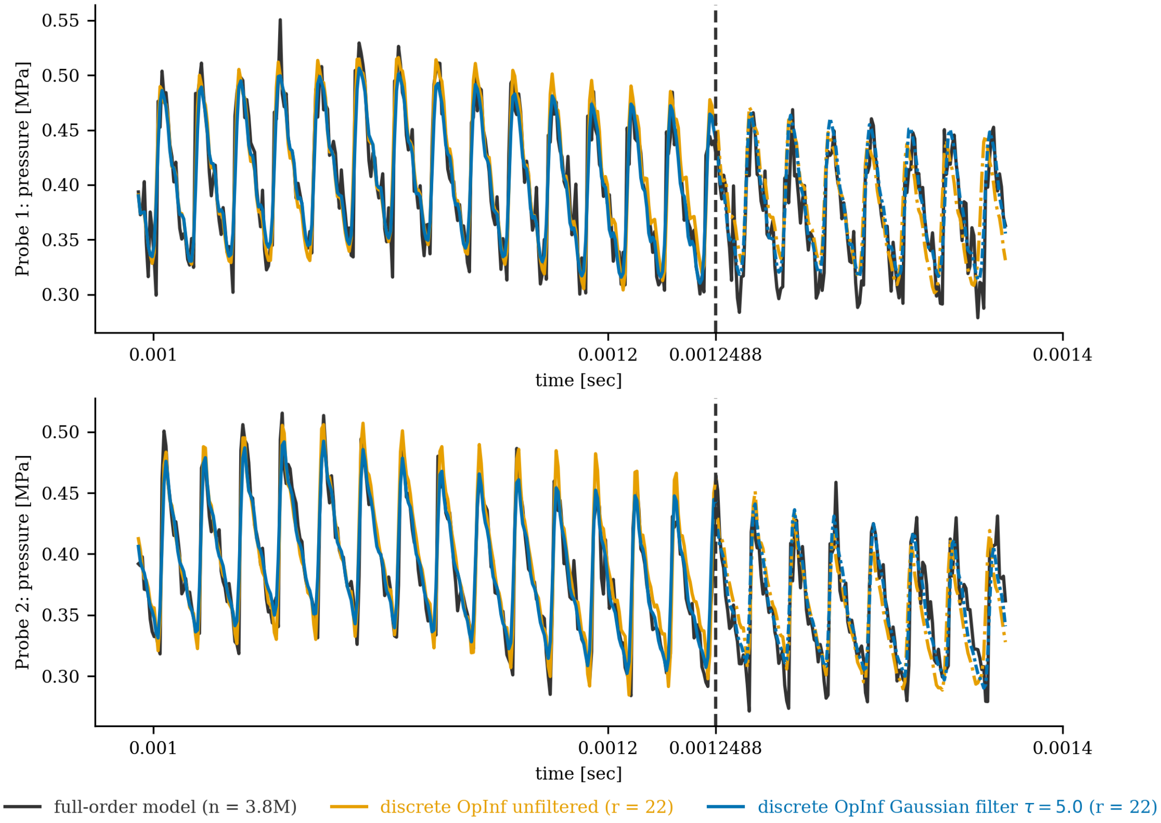

Fig. 9 Pressure traces at probes located at $(x, y, z) = (0.03, 0.035, 0.000)$ (probe 1) and $(x, y, z) = (0.04, 0.035, 0.000)$ (probe 2). The vertical dashed line at $t = 1.2488 \times 10^{-3}$ seconds separates the training from the prediction regimes.

Figure 10 compares the FOM and ROM solutions with reduced dimension $r = 22$, and plots the results at two time instances in the training regime: $t = 1.1150 \times 10^{-3}$ seconds (50th snapshot) and $t = 1.2026 \times 10^{-3}$ seconds (240th snapshot), and two instances in the prediction regime: $t = 1.2894 \times 10^{-3}$ seconds (355th snapshot) and $t = 1.3756 \times 10^{-3}$ seconds (410th snapshot). The four time instances correspond to the pressure wave at four different locations in its periodic cycle. The figure shows that the reduced model solutions capture the pressure field remarkably well, with the filtered ROM being more accurate than the unfiltered ROM in the prediction regime. Figure 11 shows the corresponding point-wise relative errors in the pressure fields for these four snapshots.

V. Conclusions

This paper proposes an approach to enhance data-driven reduced-order modeling with a preprocessing step in which the training data is filtered prior to training the reduced model. A key benefit of filtering is seen to be preventing overfitting of the ROM. This is particularly important when the amount of training data is limited, as is often the case when the data are produced by high-fidelity multiscale, multiphysics simulations. This is the situation for the problem considered here, where the high-fidelity training simulations of an RDE are limited to short time horizons due to their computational expense, and, further, due to the dataset size, not all snapshot data can be stored. For this RDE problem, the results in this paper show that even though the ROM has been trained using only 294 snapshots, the discrete operator inference model is accurate and predictive, with relative prediction errors in pressure, temperature, fuel and oxidizer fields shown to be less than 8%. Applying Gaussian filtering to the snapshot set resulted in a tradeoff between reducing

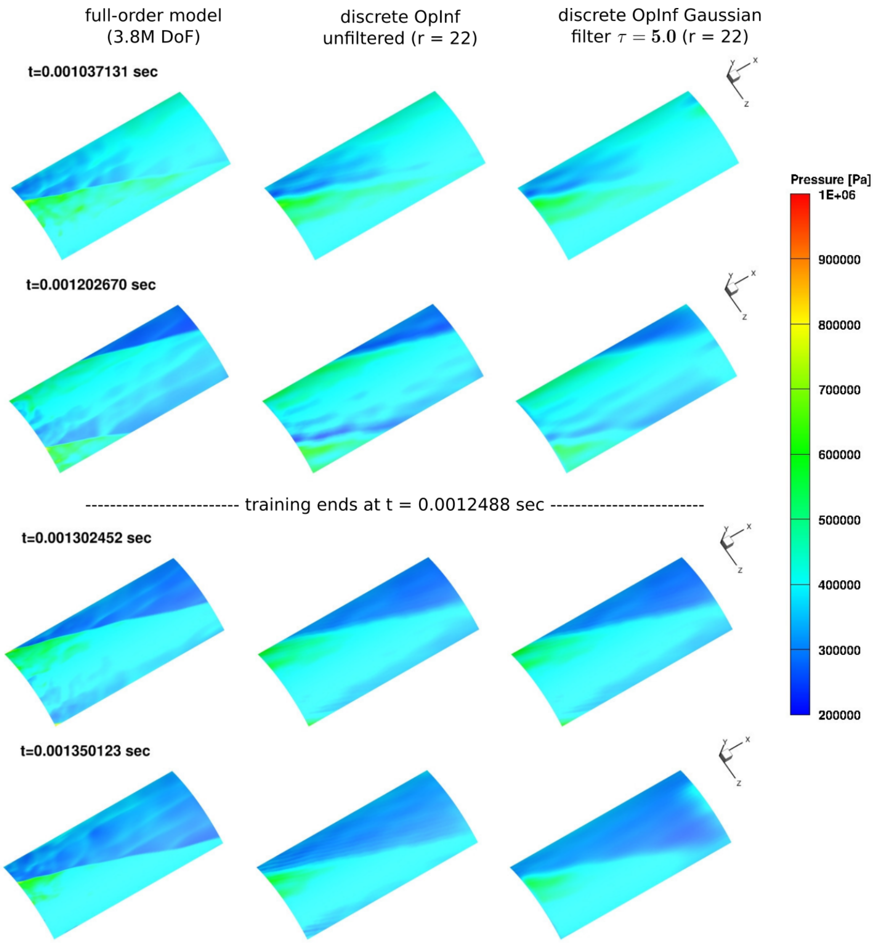

Fig. 10 ROM pressure field predictions at the cross section located at radius $R = 37$ mm at four time instances. The first two snapshots are in the training regime and the latter two are in the prediction regime.

the accuracy of the ROM over the training regime and improving the prediction power of the ROM in the testing regime. While further investigations are necessary to fully characterize the effects of training data filtering, these promising results indicate the potential benefits of the approach for scientific machine learning.

Acknowledgements

This work was supported in part by AFRL/Jacobs contract RAPT1-0000000548 and the Air Force Center of Excellence on Multifidelity Modeling of Rocket Combustor Dynamics under grant FA9550-17-1-0195.

References

[1] Lietz, C., Desai, Y., Munipalli, R., Schumaker, S. A., and Sankaran, V., *Flowfield analysis of a 3D simulation of a rotating detonation rocket engine*, 2019. https://doi.org/10.2514/6.2019-1009.

[2] Xie, X., Mohebujjaman, M., Rebholz, L. G., and Iliescu, T., "Data-Driven Filtered Reduced Order Modeling of Fluid Flows," *SIAM Journal on Scientific Computing*, Vol. 40, No. 3, 2018, pp. B834–B857. https://doi.org/10.1137/17M1145136.

[3] Scherl, I., Strom, B., Shang, J. K., Williams, O., Polagye, B. L., and Brunton, S. L., "Robust principal component analysis for modal decomposition of corrupt fluid flows," *Physical Review Fluids*, Vol. 5, 2020, p. 054401. https://doi.org/10.1103/PhysRevFluids.5.054401.

[4] Candès, E. J., Li, X., Ma, Y., and Wright, J., "Robust Principal Component Analysis?" *Journal of the ACM*, Vol. 58, No. 3, 2011. https://doi.org/10.1145/1970392.1970395.

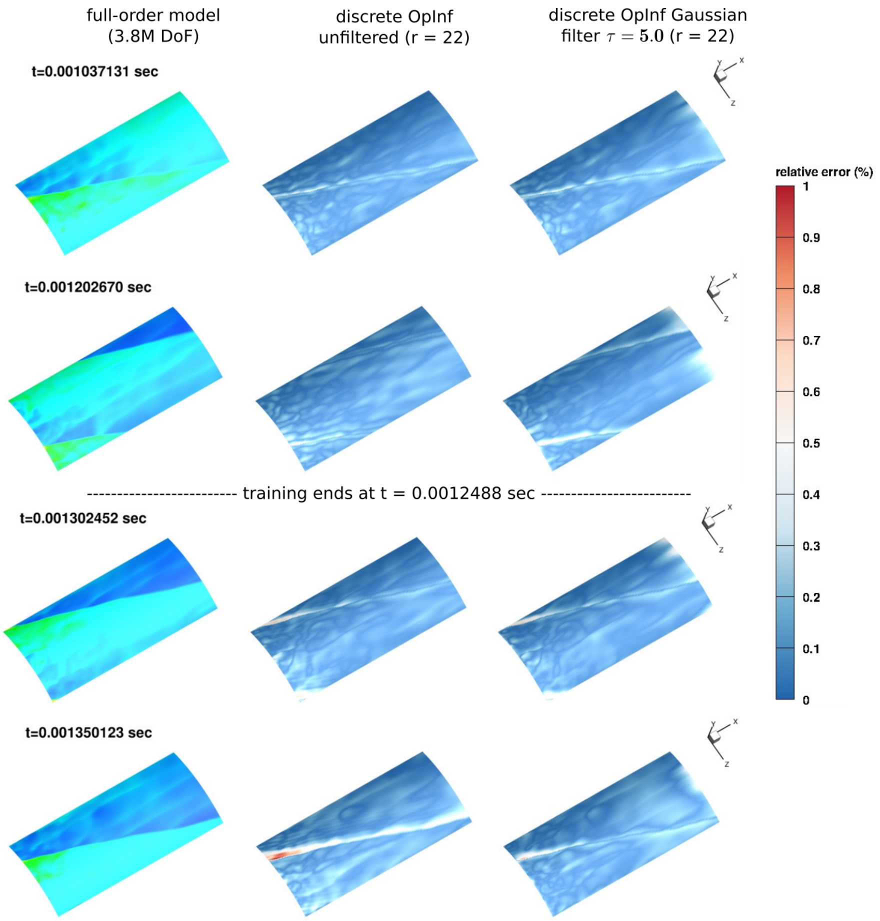

Fig. 11 Pressure fields at the cross section located at radius $R = 37$ mm. Left: FOM solution. Middle: Relative error for unfiltered OpInf ROM. Right: Relative error for filtered OpInf ROM. The first two snapshots are in the training regime and the latter two are in the prediction regime.

[5] Schmid, P. J., "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, Vol. 656, 2010, p. 5–28. https://doi.org/10.1017/S0022112010001217.

[6] Tu, J. H., Rowley, C. W., Luchtenburg, D. M., Brunton, S. L., and Kutz, J. N., "On dynamic mode decomposition: Theory and applications," *Journal of Computational Dynamics*, Vol. 1, No. 2, 2014, pp. 391–421.

[7] Berkooz, G., Holmes, P., and Lumley, J. L., "The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows," *Annual Review of Fluid Mechanics*, Vol. 25, No. 1, 1993, pp. 539–575. https://doi.org/10.1146/annurev.fl.25.010193.002543.

[8] Prazenica, R. J., Kurdila, A. J., and Vignola, J. F., "Spatial filtering and proper orthogonal decomposition of scanning laser Doppler vibrometry data for the nondestructive evaluation of frescoes," *Journal of Sound and Vibration*, Vol. 304, No. 3, 2007, pp. 735–751. https://doi.org/https://doi.org/10.1016/j.jsv.2007.03.027.

[9] Aradag, S., Siegel, S., Seidel, J., Cohen, K., and McLaughlin, T., "Filtered POD-based low-dimensional modeling of the 3D turbulent flow behind a circular cylinder," *International Journal for Numerical Methods in Fluids*, Vol. 66, No. 1, 2011, pp. 1–16. https://doi.org/https://doi.org/10.1002/fld.2238.

[10] Cerri, S. S., and Camporeale, E., "Space-filter techniques for quasi-neutral hybrid-kinetic models," *Physics of Plasmas*, Vol. 27, No. 8, 2020, p. 082102. https://doi.org/10.1063/5.0012924.

[11] Pueschel, M., Dannert, T., and Jenko, F., "On the role of numerical dissipation in gyrokinetic Vlasov simulations of plasma microturbulence," *Computer Physics Communications*, Vol. 181, No. 8, 2010, pp. 1428–1437. https://doi.org/https://doi.org/10.1016/j.cpc.2010.04.010.

[12] Peherstorfer, B., and Willcox, K., “Data-driven operator inference for nonintrusive projection-based model reduction,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 306, 2016, pp. 196–215. https://doi.org/https://doi.org/10.1016/j.cma.2016.03.025.

[13] McQuarrie, S. A., Huang, C., and Willcox, K. E., “Data-driven reduced-order models via regularised Operator Inference for a single-injector combustion process,” *Journal of the Royal Society of New Zealand*, Vol. 51, No. 2, 2021, pp. 194–211. https://doi.org/10.1080/03036758.2020.1863237.

[14] Goldstein, T., and Osher, S., “The Split Bregman Method for L1-Regularized Problems,” *SIAM Journal on Imaging Sciences*, Vol. 2, No. 2, 2009, pp. 323–343. https://doi.org/10.1137/080725891.

[15] Rudin, L. I., Osher, S., and Fatemi, E., “Nonlinear total variation based noise removal algorithms,” *Physica D: Nonlinear Phenomena*, Vol. 60, No. 1, 1992, pp. 259–268. https://doi.org/https://doi.org/10.1016/0167-2789(92)90242-F.

[16] Qian, E., Farcas, I.-G., and Willcox, K., “Reduced operator inference for nonlinear partial differential equations,” *SIAM Journal on Scientific Computing*, 2022.

[17] Clawpack Development Team, “Clawpack software,” , 2020. https://doi.org/https://doi.org/10.5281/zenodo.4025432, URL http://www.clawpack.org, version 5.7.1.

[18] Mandli, K. T., Ahmadia, A. J., Berger, M., Calhoun, D., George, D. L., Hadjimichael, Y., Ketcheson, D. I., Lemoine, G. I., and LeVeque, R. J., “Clawpack: building an open source ecosystem for solving hyperbolic PDEs,” *PeerJ Computer Science*, Vol. 2, 2016, p. e68. https://doi.org/10.7717/peerj-cs.68.

[19] Swischuk, R., Kramer, B., Huang, C., and Willcox, K., “Learning Physics-Based Reduced-Order Models for a Single-Injector Combustion Process,” *AIAA Journal*, Vol. 58, No. 6, 2020, pp. 2658–2672. https://doi.org/10.2514/1.J058943.