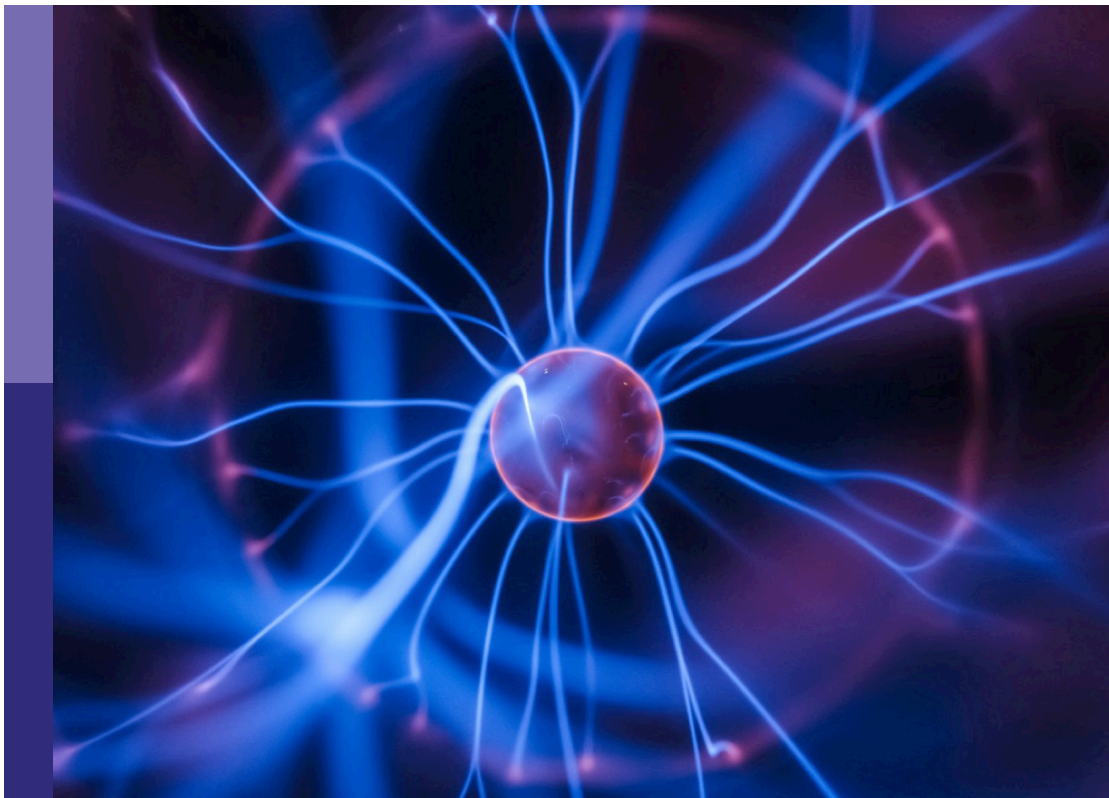
frontiers | Research Topics

Data-driven modeling and optimization in fluid dynamics: From physics-based to machine learning approaches

Edited by
Michel Bergmann, Laurent Cordier and Traian Iliescu

Published in
Frontiers in Physics
Frontiers in Applied Mathematics and Statistics



ISSN 1664-8714
ISBN 978-2-83251-070-4
DOI 10.3389/978-2-83251-070-4

About Frontiers

Frontiers is more than just an open access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers journal series

The Frontiers journal series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the *Frontiers journal series* operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews. Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the *Frontiers journals series*: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area.

Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers editorial office: frontiersin.org/about/contact

Data-driven modeling and optimization in fluid dynamics: From physics-based to machine learning approaches

Topic editors

Michel Bergmann — Inria Bordeaux - Sud-Ouest Research Centre, France
Laurent Cordier — UPR3346 Institut P' Recherche et Ingénierie en Matériaux, Mécanique et Energétique (Pprime), France
Traian Iliescu — Virginia Tech, United States

Citation

Bergmann, M., Cordier, L., Iliescu, T., eds. (2023). *Data-driven modeling and optimization in fluid dynamics: From physics-based to machine learning approaches*. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-83251-070-4

Table of contents

frontiers | Frontiers in Physics

TYPE Editorial
PUBLISHED 02 December 2022
DOI 10.3389/fphy.2022.1072691

OPEN ACCESS

EDITED AND REVIEWED BY
José S. Andrade Jr,
Federal University of Ceara, Brazil

*CORRESPONDENCE
Traian Iliescu,
iliescu@vt.edu

SPECIALTY SECTION
This article was submitted to Statistical and Computational Physics, a section of the journal Frontiers in Physics

RECEIVED 17 October 2022
ACCEPTED 22 November 2022
PUBLISHED 02 December 2022

CITATION
Bergmann M, Cordier L and Iliescu T (2022), Editorial: Data-driven modeling and optimization in fluid dynamics: From physics-based to machine learning approaches.
Front. Phys. 10:1072691.
doi: 10.3389/fphy.2022.1072691

Editorial: Data-driven modeling and optimization in fluid dynamics: From physics-based to machine learning approaches

Michel Bergmann[1], Laurent Cordier[2] and Traian Iliescu[3]*

[1]Inria Bordeaux - Sud-Ouest Research Centre, Talence, France, [2]UPR3346 Institut P' Recherche et Ingénierie en Matériaux, Mécanique et Energétique (Pprime), Poitiers, France, [3]Virginia Tech, Blacksburg, VA, United States

KEYWORDS

data-driven modeling, control, machine learning, computational fluid dynamics - CFD, reduced order model (ROM)

Editorial on the Research Topic
Data-driven modeling and optimization in fluid dynamics: From physics-based to machine learning approaches

Data-driven modeling has made a dramatic impact in computational science and engineering and, in particular, in computational fluid dynamics (CFD). One of the earliest uses of data in CFD is the proper orthogonal decomposition (POD), which was introduced by Lumley and his collaborators more than half a century ago. POD is based on a simple yet powerful idea: In the classical Galerkin framework (used in standard numerical methods, e.g., finite element or spectral methods), replace the general purpose basis functions with data-driven basis functions. This very simple idea has made a profound impact in CFD, reducing the computational cost of standard numerical methods by orders of magnitude and enabling challenging numerical simulations in shape optimization, flow control, and uncertainty quantification. Since Lumley's pioneering work, the field of data-driven modeling has witnessed a tremendous development. Probably the most exciting research area in this field is the use of machine learning. Over the last decade, the focus in data-driven modeling has shifted from physics-based strategies to machine learning approaches, in which instead of merely changing different components of classical methods (e.g., changing the basis in POD), one completely overhauls the entire framework (e.g., instead of using a Galerkin framework, one leverages machine learning algorithms to determine all the model operators).

At this point, one natural question is which strategy should be used in CFD? Should one use physics-based or machine learning models? We believe that, as is often the case when discussing numerical methods, the truth is somewhere in the middle. That is, we believe that data-driven models that combine the physical and mathematical insight with machine learning strategies can revolutionize CFD and break new barriers in shape optimization, flow control, and uncertainty quantification. This Research Topic, which

consists of 10 articles written by leaders in the field, surveys recent developments in data-driven modeling in CFD, covering a spectrum of modeling strategies, from physics-based to machine learning modeling.

Kaneko and Fischer put forth an augmented-basis method (ABM) to stabilize reduced order models (ROMs) of turbulent incompressible flows. The new strategy augments the classical POD basis functions with divergence-free projections of a subset of the nonlinear interaction terms that constitute a significant fraction of the time-derivative of the solution. The numerical investigation shows that the ABM outperforms the standard ROM and the Leray regularized ROM. Huang et al. propose a component-based domain-decomposition framework for the modeling of large-scale systems that cannot be directly accessed using the high-fidelity simulations (e.g., rocket engines or wind farms). The new framework decomposes the full system into different components, each of which can flexibly adopt different modeling strategies (e.g., reduced order modeling or full order modeling), balancing physical complexity with accuracy requirements. The authors investigate the new framework in the numerical simulation of complex flows involving combustion dynamics. Tsai and Fischer propose a time-averaged error indicator for regularized ROMs of two-dimensional unsteady natural convection in a high-aspect ratio slot parameterized with the Prandtl number, Rayleigh number, and slot angle with respect to the gravity. The authors show that the Leray-regularized ROMs provide a robust strategy for this class of flows. Chan et al. show that, for variable density flows with under-resolved features, there are differences in robustness between entropy stable schemes which incorporate the entropy projection and those which do not. These differences in robustness are observed to depend on the density contrast and persist across a range of polynomial degrees, mesh resolutions, and types of discretization. Chacón Rebollo et al. propose a new low-rank tensorized decomposition (LRTD) to approximate the solution of parametric non-symmetric elliptic problems. Furthermore, they prove that the truncated LRTD expansion strongly converges to the parametric solution. Finally, the numerical investigation for convection-diffusion problems supports the theoretical developments and illustrates the computational efficiency of the new algorithm.

Schmekel et al. use data from a direct numerical simulation (DNS) of a turbulent channel flow to train a convolutional neural network (CNN) and predict the number and volume of the coherent structures in the channel over time. The numerical investigation shows that the proposed CNN accurately predicts the temporal evolution of the coherent structures and displays very good agreement with the reference data. Jacobsen and Duraisamy utilize variational autoencoders (VAEs) for nonlinear dimension reduction to disentangle the low-dimensional latent variables and identify independent physical parameters that generated the data. A disentangled decomposition is interpretable and can be transferred to, e.g., design optimization and probabilistic reduced order modeling. To characterize the training process of the VAEs and to study disentanglement, the authors use a porous media flow modeled by the two-dimensional steady-state Darcy equations. Popov and Sandu propose a significant improvement of the multifidelity ensemble Kalman filter (MFEnKF), which combines a full order physical model and a hierarchy of reduced order surrogate models to increase the computational efficiency of data assimilation. In this new strategy, the linear framework is generalized to leverage nonlinear projection and interpolation operators implemented using autoencoders. The new approach, named NL-MFEnKF, enables the use of a much more general class of surrogate models than MFEnKF. Heaney et al. combine nonintrusive reduced order modeling (NIROM) and domain decomposition to enable ROMs to make predictions for unseen scenarios. The authors successfully test the new strategy in the numerical simulation of chaotic time-dependent flow of air past buildings. Heiland et al. propose the use of CNNs and POD to construct very low-dimensional linear parameter varying (LPV) approximations to the incompressible Navier-Stokes equations (NSE). These LPV approximations could be leveraged in challenging NSE control applications. The authors illustrate their theoretical developments in the numerical simulation of a two-dimensional flow around a cylinder.

The 10 articles in this Research Topic survey recent developments in data-driven modeling in CFD, with a particular emphasis on turbulent flows. This is an exciting research area, with many open problems and grand challenges waiting to be addressed.

Author contributions

TI, MB, and LR contributed to this editorial.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

frontiers | Frontiers in Physics

ORIGINAL RESEARCH
published: 13 April 2022
doi: 10.3389/fphy.2022.888832

Predicting Coherent Turbulent Structures via Deep Learning

D. Schmekel[1], *F. Alcántara-Ávila*[1], *S. Hoyas*[2] *and R. Vinuesa*[1]*

[1]*FLOW, Engineering Mechanics, KTH Royal Institute of Technology, Stockholm, Sweden,* [2]*Instituto de Matemática Pura y Aplicada, Universitat Politècnica de València, València, Spain*

Turbulent flow is widespread in many applications, such as airplane wings or turbine blades. Such flow is highly chaotic and impossible to predict far into the future. Some regions exhibit a coherent physical behavior in turbulent flow, satisfying specific properties; these regions are denoted as coherent structures. This work considers structures connected with the Reynolds stresses, which are essential quantities for modeling and understanding turbulent flows. Deep-learning techniques have recently had promising results for modeling turbulence, and here we investigate their capabilities for modeling coherent structures. We use data from a direct numerical simulation (DNS) of a turbulent channel flow to train a convolutional neural network (CNN) and predict the number and volume of the coherent structures in the channel over time. Overall, the performance of the CNN model is very good, with a satisfactory agreement between the predicted geometrical properties of the structures and those of the reference DNS data.

Keywords: turbulence, coherent turbulent structures, machine learning, convolutional neural networks, deep learning

OPEN ACCESS

Edited by:
Traian Iliescu,
Virginia Tech, United States

Reviewed by:
Omer San,
Oklahoma State University, United States
Diana Bistrian,
Politehnica University of Timişoara, Romania

***Correspondence:**
R. Vinuesa
rvinuesa@mech.kth.se

Specialty section:
This article was submitted to Statistical and Computational Physics, a section of the journal Frontiers in Physics

Received: *03 March 2022*
Accepted: *23 March 2022*
Published: *13 April 2022*

Citation:
Schmekel D, Alcántara-Ávila F, Hoyas S and Vinuesa R (2022) Predicting Coherent Turbulent Structures via Deep Learning. Front. Phys. 10:888832. doi: 10.3389/fphy.2022.888832

INTRODUCTION

Fluid flow is vital for a large variety of applications such as aircraft, heat pumps, lubrication, etc. [1]. Typically, for many applications, the flow is in a turbulent regime [1]. Such flow is characterized by being chaotic and highly non-linear, with large mixing amounts. Consequently, turbulent flow is a challenge to modellers [2]. It has been estimated that turbulence is responsible for up to 5% of the total CO_2 generated by humanity every year [3]. Even small gains in understanding turbulence can be very impactful. Fluid flow, including turbulent flow, is described by the Navier–Stokes equations, which are generally impossible to solve analytically. They can be solved numerically, but this has traditionally been prohibitively computationally expensive—only elementary geometries have been simulated [4,5]. In recent years, it has become possible to perform high-fidelity simulations of complex geometries [6–9].

One of the earlier studies on the structure of turbulence was carried out by Kline et al. [10]. Kline et al. also investigated the statistical properties of turbulence and found that most of the turbulence production takes place near the walls (at least at low Reynolds numbers). They observed specific regions in the flow, called coherent turbulent structures, which we will denote as structures. One essential type of coherent structure is strongly related to Reynolds stresses [11]. Typically, these Reynolds-stress structures may occupy around 4% of the volume but can be responsible for around 30% of the Reynolds stresses. The structures are also important for the transfer of several properties such as mass, heat, and momentum [12]. Many models created for studying turbulence are built upon these structures [13]. Traditionally, the focus of structures has been on *hairpins*, U-shaped structures formed near walls going to the outer region[14]. Hairpins were the basic building block in several models [15–17], which formed *hairpin clusters* [14]. Objections to these models have arisen

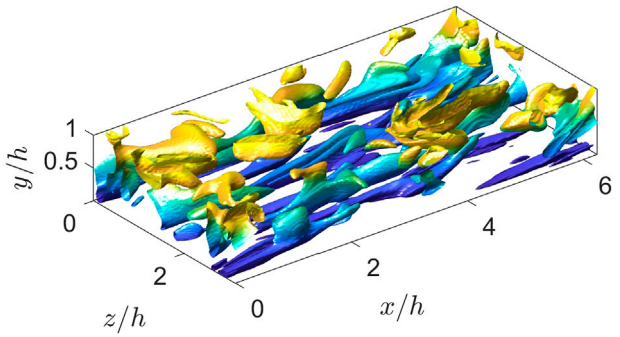


FIGURE 1 | Reyhnolds-stress structures at the bottom half of the channel. The structures are coloured by wall-normal distance. The flow moves from the left to the right of the figure.

since they have had problems at higher Reynolds numbers [18]. Instead, momentum-transfer models have been created, focusing on strong Reynolds-stress and momentum-transfer events. Some data supports these types of models for modeling momentum transfer in the logarithmic layer [13].

In this study we will use deep neural networks (DNNs), which are black-box methods [19,20] and are universal function approximators. They can approximate any sufficiently smooth function arbitrarily well. In the DNN framework, it is assumed that the phenomena under study can be described by some predetermined parameterizable function $f(x; \Theta)$, where Θ are the parameters. The values of Θ that best approximate the data are obtained by means of algorithms such as stochastic gradient descent and the back-propagation [21]. DNNs have been used successfully for modelling the temporal dynamics of turbulence [22,23], for non-intrusive sensing [24,25], for identifying patterns in complex flows [26] and for modelling the Reynolds stresses [27]. Two overviews of the current applications of DNNs in fluid mechanics can be found in [2,28]. Here we investigate the possibilities to predict the temporal evolution of coherent turbulent structures with machine-learning techniques. To this end, we create a DNN-based mode and assess the quality of its predictions, in terms of the number of structures, the total volume of the structures, and the volume of the largest structure. The goal is to develop a model capable of estimating plausible future scenarios of the flow, focusing on the characteristics of the turbulent structures. We also expect this model to exhibit appropriate generalization properties [29].

The article is structured as follows: in §2 we discuss the data collection and the network design; in §3 we present our results; and finally conclusions and discussions are presented in §4.

METHODS

Numerical Setup

We study wall-bounded turbulent structures in a turbulent channel flow, consisting of two infinitely large planes parallel to the x (streamwise) and z (spanwise) directions. The distance between the planes is $2h$. **Figure 1** shows an illustration of problem. A pressure gradient in the streamwise direction drives the flow, which has a friction Reynolds number $Re_\tau = 125$. The friction Reynolds number, defined as $Re_\tau = u_\tau h/\nu$, is the main control parameter in wall bounded turbulence. Here $u_\tau = \sqrt{\tau_w/\rho}$ is the friction velocity, ν is the kinematic viscosity, ρ is the density, and τ_w is the friction at the wall.

This simulation has been performed in a computational box of sizes $L_x = 2\pi h$, $L_y = 2h$ and $L_z = \pi h$. This box is large enough to accurately describe the statistics of the flow [30,31]. The streamwise, wall-normal, and spanwise velocity components are U, V and W or, using index notation, U_i. Statistically-averaged quantities in time, x and z are denoted by an overbar, $\bar{U}$, whereas fluctuating quantities are denoted by lowercase letters: $U = \bar{U} + u$. Primes are reserved for intensities: $u' = \overline{uu}^{1/2}$. The domain is periodic in x and z. The walls are at rest, and a pressure gradient drives the flow at the prescribed Reynolds number. This turbulent flow can be described by means of the mass balance and momentum equations:

$$\partial_j U_j = 0, \tag{1}$$

$$\partial_t U_i + U_j \partial_j U_i = -\partial_i P + \frac{1}{\mathrm{Re}_\tau} \partial_{jj} U_i, \tag{2}$$

where repeated subscripts indicate sumation over 1, 2, 3 and the pressure term includes the density. These equations have been solved using the LISO code [4], similar to the one described by Lluesma-Rodríguez et al. [32]. This code has successfully been employed to run some of the largest simulations of wall-bounded turbulent flows [4,33–37]. Briefly, the code uses the same strategy as that described by Kim et al. [38], but using a seven-point compact-finite-difference scheme in the y direction with fourth-order consistency and extended spectral-like resolution [39]. The temporal discretization is a third-order semi-implicit Runge–Kutta scheme [40]. The wall-normal grid spacing is adjusted to keep the resolution to $\Delta y = 1.5\eta$, i.e., approximately constant in terms of the local isotropic Kolmogorov scale $\eta = (\nu^3/\epsilon)^{1/4}$. Note that ϵ is the isotropic dissipation of turbulent kinetic energy. In wall units, Δy^+ varies from 0.3 at the wall, up to $\Delta y^+ \simeq 12$ at the centerline.

As a consequence of the self-sustaining mechanism, coherent structures in the form of counter-rotating rolls are triggered by pairs of ejections and sweeps extending beyond the buffer layer in a well-organised process called bursting. The ejections carry low streamwise velocity upwards from the wall ($u < 0$, $v > 0$), while the sweeps carry high streamwise velocity downwards to the wall ($u > 0$, $v < 0$). Based on a Reynolds stress quadrant classification, ejections and sweeps are Q2 and Q4 events, respectively. Lozano-Duran et al. [13] and Jiménez [18] reported the relation between counter-rotating rolls, streamwise streaks and Q2-Q4 pairs in turbulent Poiseuille flow by observing averaged flow fields conditioned to the presence of a wall-attached Q2-Q4 pair. A wall-attached event is an intense Reynolds stress structure (i.e. uv-structure) that approaches a wall below $y^+ < 20$. The reasoning for this definition is explained later. For a time-resolved view of the bursting process in turbulent Poiseuille channel at $Re_\tau \approx 4{,}200$, the interested reader is referred to [30]. Gandía Barberá

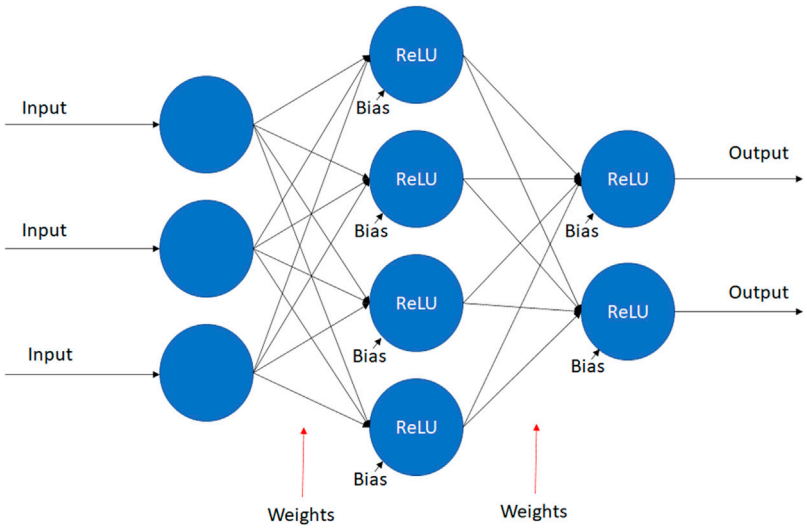


FIGURE 2 | Schematic representation showing how the output is calculated in a two-layer artificial neural network where rectified linear unit (ReLU) is the activation function.

et al. [41] performed this process again for Couette flows in presence of stratification.

In order to study the underlying physics of the flow, the coherent structures responsible for the transport of momentum are analysed. Jiménez [18] discussed that the intensity of a given parameter is considered as an indicator of coherence, among other characteristics. However, the selection of a threshold is only feasible if the parameter is intermittent enough to separate between high- and low-intensity regions. After analysing the intermittency of different parameters, it is found that quadratic parameters, specially the Reynolds stress, are more appropriate to describe intense coherent structures.

We are interested in using a DNN to predict how these structures evolve. Running the code we obtain a three-dimensional (3D) instantaneous flow fields (snapshots) sequence. Since the flow in the channel is statistically symmetric, we will only use the lower half of the channel for faster calculations. The final snapshots have $96 \times 76 \times 96$ grid points, in x, y and z respectively.

In order to identify the points that are part of structures in the velocity field we use the technique described in Lozano-Durán and Jiménez [30]. Essentially, a point p is said to be part of a structure if the following holds:

$$|u(x,y,z)v(x,y,z)| > Hu'(y)v'(y), \tag{3}$$

where H is the percolation index with a value of 1.75 [30,41]. We obtain binary 3D fields where a point in the field takes the value of 1 if and only if the point is part of a structure. A total of 1,000 fields were used for training and testing the DNN models, which are discussed next.

Deep-Learning Models

DNNs are parameterizable functions. These networks consist of artificial neurons, which are components originally inspired by brain neurons. A neuron is a function of the form $f(w^t x_i + b)$, where w, b are parameters, named weight and bias, respectively. Note that f is the activation function, an almost everywhere differentiable function, and x_i is the input vector. We can create an artificial neural network by using multiple neurons and connecting them in different ways, typically in layers. For example, a typical setup is to have a vector of neurons. Its output is used as input to the neuron in the next layer.

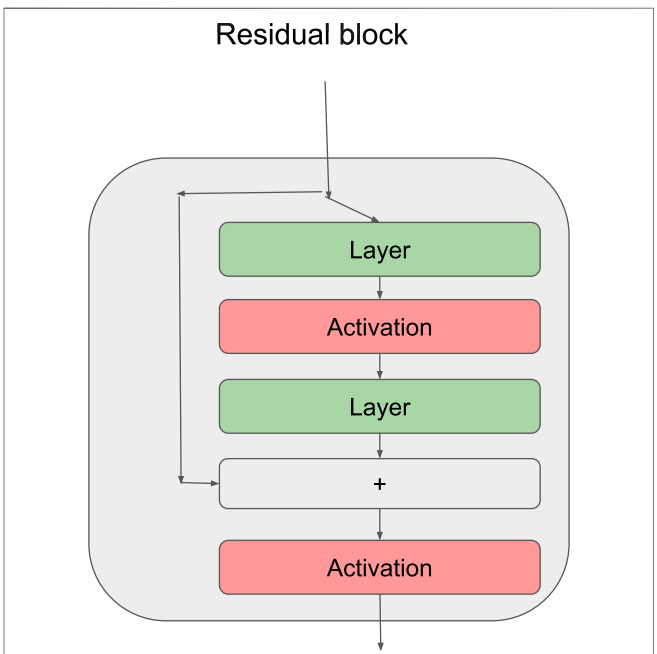


FIGURE 3 | Representation of a residual block, where two layers are skipped.

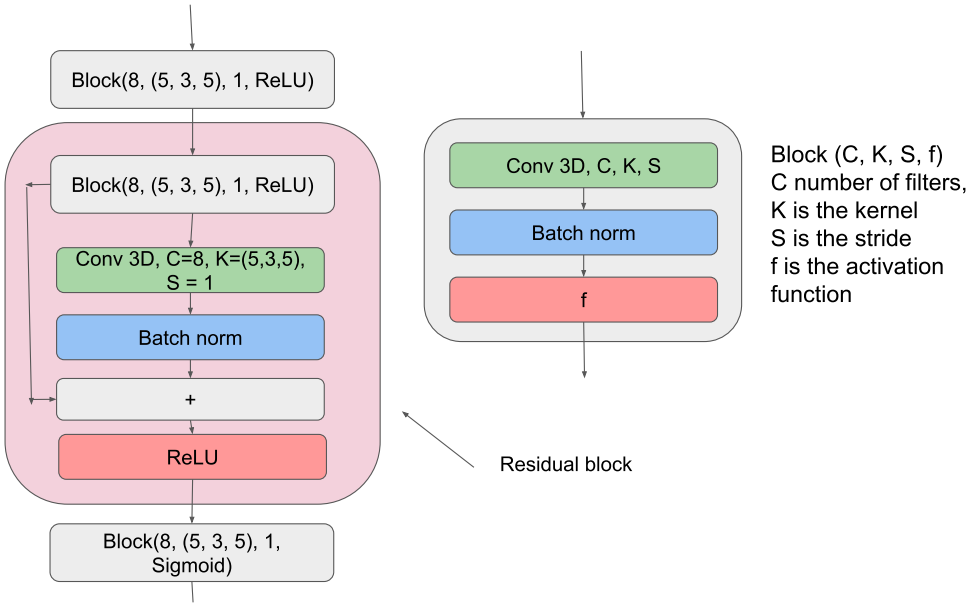


FIGURE 4 | Schematic representation of the CNN architecture employed in this study.

$$f\left(w_{21}\left(f\left(w_{11}x_{i1}+b_{11}\right)+w_{22}f\left(w_{1,2}x_{i2}+b_{12}\right)+b_2\right).\right. \quad (4)$$

This is an example of two layers, where the first layer is fed into the second one. **Figure 2** shows an illustration of a simple neural network. Since we analyze 3D fields, we use a convolutional neural network (CNN) [42]. This network is a type of DNN specifically designed to work with images. He et al. [43] further demonstrated that it is possible to improve the performance of CNNs by using skip connections. A skip-connection is a shortcut, allowing the input to skip layers, as shown in **Figure 3** and the following equation:

$$f\left(w_2^t\left(x_i+f\left(w_1^t x_i+b_1\right)+b_2\right).\right. \quad (5)$$

Recurrent neural networks (RNN) [44] are DNNs designed for modeling time series. They use their own previous output h_{i-1} in combination with the input x_i to calculate the next output:

$$\mathrm{RNN}\left(x_i, h_{i-1}\right)=h_i. \quad (6)$$

Ideally the network learns to encode useful information in the output allowing the network to "remember" the past and predict better. We will be investigating the potential of using a long-short-term-memory (LSTM) netowrk [15], since they have exhibited very good performance [15]. One notable drawback with LSTMs is the fact that they are not designed for image analysis. Their memory requirements scales quadratically with input size, thus requiring to downsample the input. Therefore, we will investigate two networks, one including an LSTM and one without it, as discussed below. There are several possible choices for the activation function. In this work we use the rectified linear unit (ReLU) everywhere but the last layer, which has the form:

$$\mathrm{ReLU}\left(x_i\right)=\max\left(x_i, 0\right). \quad (7)$$

This activation function has been shown empirically to exhibit excellent performance in computer-vision problems [45]. We use the sigmoid activation function for the last layer to ensure that the output is in the range [0,1]. We will also use batch normalization [46], in particular the batch norm, which has been empirically proven to decrease training time and improve performance [47]. We use the first 800 fields as a training set and the remaining as a validation set. Our training and validation data is split into sequences of 16 fields each. The network accepts a sequence, and for each image in the sequence, predicts the following field in the time-series. All the hyper-parameters are tuned empirically, and **Figure 4** shows the final architecture.

We train our networks by minimizing the binary cross-entropy (BCE) between the predicted and the reference fields. To minimize training and inference discrepancy we will use the algorithm developed by Bengio et al. [48] during training. Thus, for a given sample of real fields, $x_{i1}, x_{i2}, x_{i3}, \ldots, x_{im}$, the network will use the following algorithm:

Algorithm 1 :

$x'_{i2} \leftarrow \mathrm{DNN}(x_{i1})$
$k \leftarrow 2$
for k < m **do**
  u ← Uniform random(0,1)
  if u < p **then**
    $x'_{i(k+1)} \leftarrow \mathrm{DNN}(x_{ik})$
  else
    $x'_{i(k+1)} \leftarrow \mathrm{DNN}(x'_{ik})$
  end if

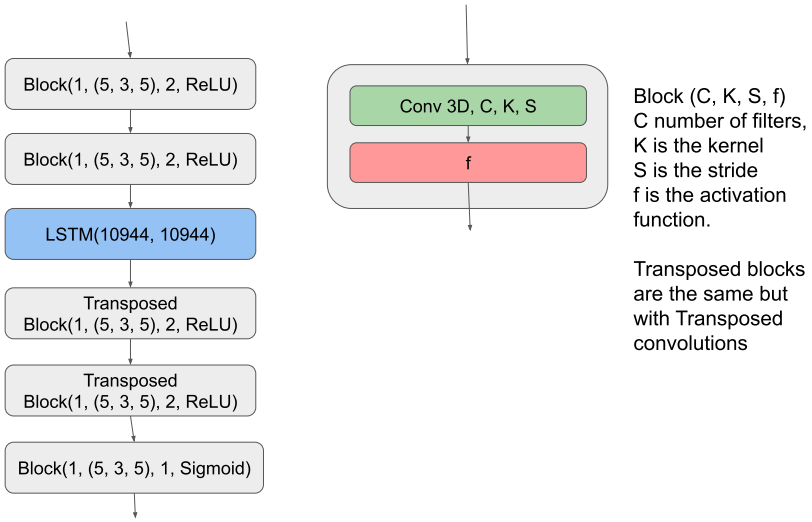


FIGURE 5 | Schematic representation of the CNN-LSTM architecture employed in this study.

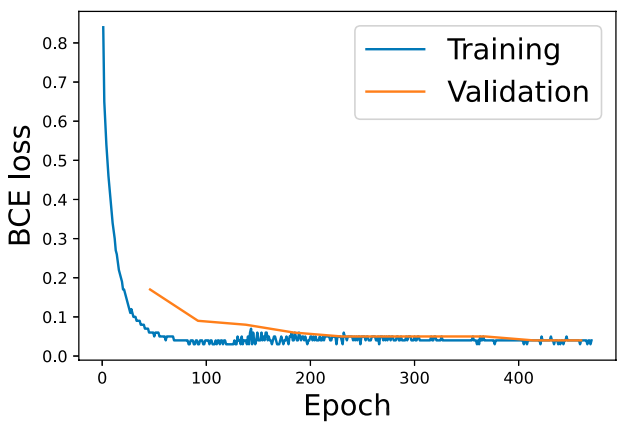


FIGURE 6 | Training and validation binary-cross-entropy (BCE) losses for the CNN architecture as a function of the training epoch.

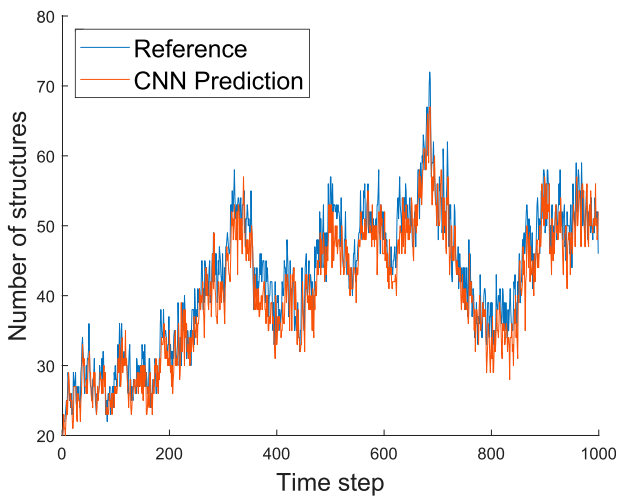


FIGURE 7 | Predicted and reference number of structures as a function of the time step for the inner region, *i.e.* $y \leq 0.2$. We observe that the number of structures is not constant over time.

We anneal p at the speed of the inverse-sigmoid function parameter $k = 30$ during the training. At the start, the network will mostly make predictions based on actual data, while at the end, it will use its predictions. Several metrics are used for the evaluation of the network. We assessed the loss of the network during training to confirm that the network converges as expected. We are also interested in studying metrics such as the number of predicted structures in the field. Since the network is not outputting binary images but fields where every value is in the range [0, 1], we will apply rounding to the output. In this work, we use the algorithm described by Aguilar-Fuertes et al. [14] to identify structures and the volume of the minimum enclosing boxes.

RESULTS

This study shows that the CNN-LSTM configuration, shown in **Figure 5**, exhibits poorer results. It only managed to learn the zero mapping, i.e. CNN-LSTM(x) = 0 $\forall x$. We hypothesize that this is caused by the field becoming too granular when downsampling so significantly. Thus, we will focus on the CNN architecture. Let us start by discussing the training process of the CNN configuration. In **Figure 6** we show the training and validation losses, which decrease as expected. We observe that our validation loss starts above the training loss at around 50 steps but converges to a very similar value at around 200 steps. This significant loss difference is due to us testing in inference mode. The figure shows that the training loss becomes noisier at around 150 steps. This result is expected because as we predict farther into the future, we use more predicted samples rather than the ground truth, thus leading to the accumulation of

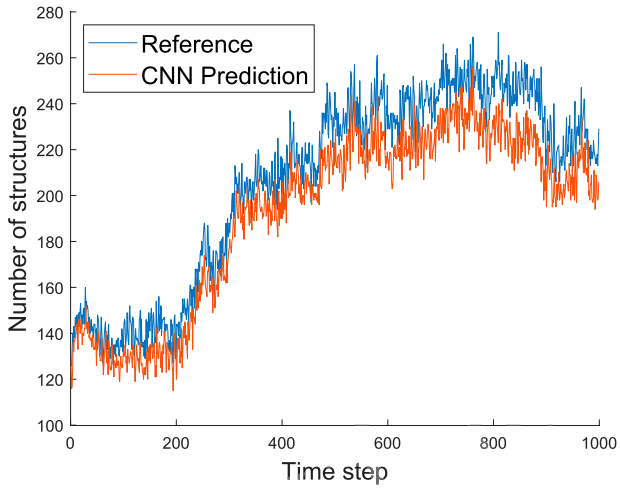


FIGURE 8 | Predicted and reference number of structures as a function of the time step for the wake region, *i.e.* $y > 0.2$. We observe that the number of structures is not constant over time.

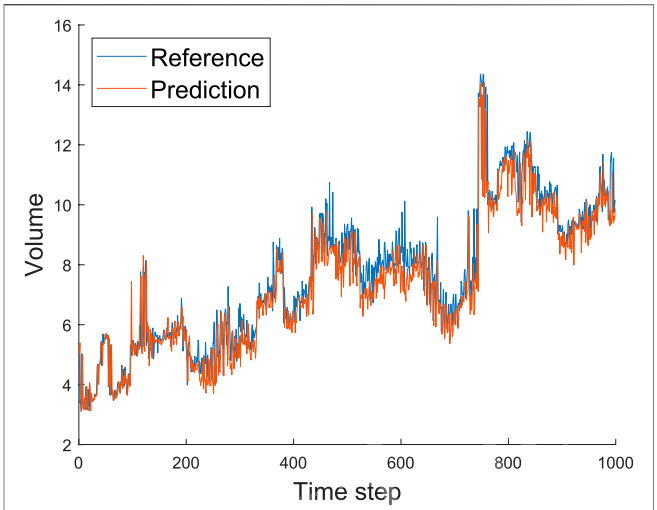


FIGURE 10 | Predicted and reference volume (scaled with h^3) of all the structures in the wake region ($y > 0.2$) as a function of the time step. Note that the network rarely overestimates this volume.

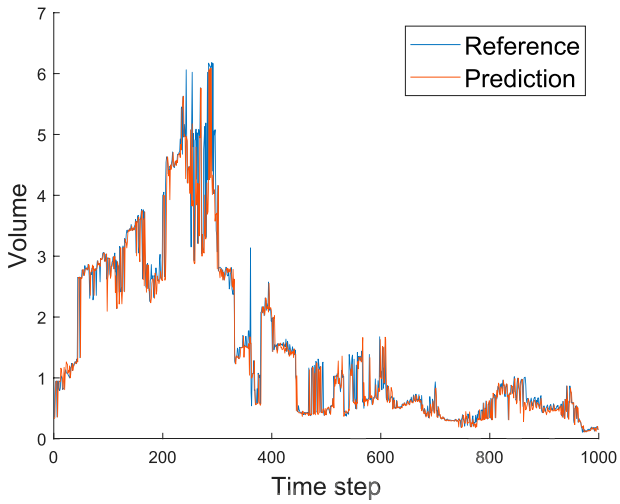


FIGURE 9 | Predicted and reference volume (scaled with h^3) of all the structures in the inner region ($y \leq 0.2$) as a function of the time step. Note that the network rarely overestimates this volume.

errors. Interestingly, the training and validation losses reach the same value of 0.04 towards the end of the prediction horizon. Note that, although this could be indicative of under-fitting, using more complex models (*i.e.* deeper networks with more channels) did not produce any improvements in the results. We note that this is a highly chaotic problem, where instantaneous predictions are highly challenging, although the dynamic behavior of the flow can be predicted with excellent accuracy [23].

Next, we will assess the number (and the volume) of the coherent turbulent structures identified in the reference simulation and the predicted fields. **Figures 7**, **8** show the number of identified structures in the inner ($y \leq 0.2$) and wake ($y > 0.2$) regions, respectively. The CNN architecture can accurately predict the evolution of the number of structures in time, with a small underestimation in the inner region and a slightly larger underestimation in the wake region. A plausible explanation for this result is that the network is conservative in its predictions. Consider the following scenario, where a point has a 10% chance of being part of a structure. The best possible guess would be a field of zeroes for a whole field, although it is doubtful that every point is zero. Similarly, the best prediction the network can make is likely zero for some points. In fact, these are the points near the edges of the structures that are the most challenging to predict. Thus we would expect the difference between predicted and real fields to grow proportionally to the number of structures. The data supports this explanation since the error is noticeably smaller when the number of structures is ≈ 150 compared to ≈ 250.

After predicting the number of structures in the turbulent fields, we analyze the volume of those objects. We show the evolution of the total volume of the structures in the inner and wake regions in **Figures 9**, **10**, respectively. It can be observed that the employed CNN architecture exhibits excellent accuracy in the volume predictions. In the inner region, the only significant discrepancy we observe is at around step 400, while in the wake region, a discrepancy is observed around step 600. These deviations can be explained by the process to calculate the volume of the structures, which relies on the volume of the bounding box [13]. Note that a wrongly predicted zero value (*i.e.*, no structure in that grid point) may have a significant effect if it disconnects a large structure. In this case, we will consider the volumes of two smaller boxes instead of the much larger volume of the complete bounding box. Interestingly, we do not see any network instance predicting a much larger volume than that of the real data. We expect the network to be slightly conservative for the same reasons outlined above, leading to underestimating the predicted volumes. In practice, the network only has to accurately predict the largest structures to obtain a correct prediction of the total volume. Furthermore, most of the time, these largest structures are not

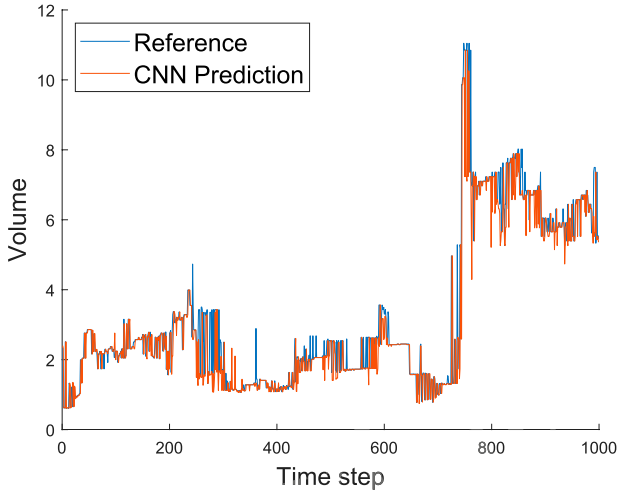


FIGURE 11 | Predicted and reference volume (scaled with h^3) of the largest structure in the domain as a function of the time step. Note that the sharp increase at around time step 750 matches the behavior in the wake region documented in **Figure 10**.

particularly sensitive to individual points. Thus, predicting the total volume is not a very challenging task.

Finally, in **Figure 11** we show the predicted and reference volumes of the largest structure in the domain as a function of the time step. Firstly, this figure shows that the largest structure is often responsible for over 50% of the total volume of all the structures in the domain. Interestingly, the CNN architecture exhibits very accurate results also when predicting the volume of the largest scales. Around time step 400, it can be observed that the volume difference between the predicted and real data is about one. The total volume difference supports our hypothesis that the (limited) discrepancies are associated with the calculation of the bounding-box volume. Furthermore, the sharp increase in maximum volume observed at around time step 750 is due to the merger of two different structures. All these results indicate that the CNN architecture can very accurately predict the geometrical properties of the structures, including the total number of objects and their volumes.

DISCUSSION AND CONCLUSION

In this work, we have designed a DNN capable of predicting the temporal evolution of the coherent structures in a turbulent channel flow. The employed CNN exhibits excellent agreement with the reference data, and some observed deviations are due to the method to calculate volumes based on bounding boxes. This also leads to scenarios where larger structures are responsible for a disproportionally large part of the total volume than their actual volume. Adding a single point to an edge of the structure is equivalent to adding a plane using this volume metric. Despite the mentioned caveats, this metric has been used to facilitate comparisons with other studies focused on coherent structures in turbulent channels. We also observe that the network predictions are conservative, with a general underprediction of the number of structures and their volume. This is associated with the rounding of the predictions: most points have a higher probability of being zero than one, and then the network will likely predict zero. This is not necessarily an issue, but future work will be focused on investigating the focal binary loss [50], to obtain a more even distribution. Note that our network shows signs of underfitting since the training and validation losses have approximately the same value. This was also the case in more complex networks investigated in this work. Overall, the performance of the CNN model is outstanding, with a satisfactory agreement between the predicted geometrical properties of the structures and those of the reference DNS data. In particular, throughout the whole time interval under study, our model leads to less than 2% error in the volume predictions and less than 0.5% in the predictions of number of structures.

When it comes to deep-learning models, including temporal information, we note the potential for further improving the predictions. This is because these models enable exploiting the spatial features in the data (as the CNN does) and the temporal correlations among snapshots, where multiple fields can be used as an input. In this work, we have also investigated adding a long-short-term-memory (LSTM) network [49] to handle the temporal information, although the significantly increased memory requirements of the new architecture limited its accuracy. Future work will aim at assessing more complicated architectures involving better downsampling, as in the U-net confgiration [50], or more efficient temporal networks such as temporal CNNs [51] or transformers [52].

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

DS performed the deep-learning analysis and wrote the paper; FA-A performed the simulations and edited the paper; SH performed the simulations and edited the paper; RV ideated the project, supervised and edited the paper.

FUNDING

RV acknowledges the financial support by the Göran Gustafsson foundation. SH was funded by Contract Nos. RTI2018-102256-B-I00 of Ministerio de Ciencia, innovación y Universidades/FEDER. Part of the analysis was carried out using computational resources provided by the Swedish National Infrastructure for Computing (SNIC).

ACKNOWLEDGMENTS

The authors thank Dr. H. Azizpour for his contributions on the deep-learning part of this work.

REFERENCES

1. Kundu PK, Cohen IM, Dowling DR. *Fluid Mechanics*. Cambridge, MA, USA: Academic Press (2015).
2. Vinuesa R, Brunton SL. The Potential of Machine Learning to Enhance Computational Fluid Dynamics. *Preprint arXiv:2110.02085* (2021).
3. Jiménez J. Near-wall Turbulence. *Phys Fluids* (2013) 25:101302.
4. Hoyas S, Jiménez J. Scaling of the Velocity Fluctuations in Turbulent Channels up to Reτ=2003. *Phys Fluids* (2006) 18:011702. doi:10.1063/1.2162185
5. Hoyas S, Oberlack M, Alcántara-Ávila F, Kraheberger SV, Laux J. Wall Turbulence at High Friction reynolds Numbers. *Phys Rev Fluids* (2022) 7: 014602. doi:10.1103/PhysRevFluids.7.014602
6. Noorani A, Vinuesa R, Brandt L, Schlatter P. Aspect Ratio Effect on Particle Transport in Turbulent Duct Flows. *Phys Fluids* (2016) 28:115103. doi:10.1063/1.4966026
7. Vinuesa R, Negi PS, Atzori M, Hanifi A, Henningson DS, Schlatter P. Turbulent Boundary Layers Around wing Sections up to Rec=1,000,000. *Int J Heat Fluid Flow* (2018) 72:86–99. doi:10.1016/j.ijheatfluidflow.2018.04.017
8. Abreu LI, Cavalieri AVG, Schlatter P, Vinuesa R, Henningson DS. Spectral Proper Orthogonal Decomposition and Resolvent Analysis of Near-wall Coherent Structures in Turbulent Pipe Flows. *J Fluid Mech* (2020) 900:A11. doi:10.1017/jfm.2020.445
9. Vinuesa R. High-fidelity Simulations in Complex Geometries: Towards Better Flow Understanding and Development of Turbulence Models. *Results Eng* (2021) 11:100254. doi:10.1016/j.rineng.2021.100254
10. Kline SJ, Reynolds WC, Schraub FA, Runstadler PW. The Structure of Turbulent Boundary Layers. *J Fluid Mech* (1967) 30:741–73. doi:10.1017/s0022112067001740
11. Ganapathisubramani B, Longmire EK, Marusic I. Characteristics of Vortex Packets in Turbulent Boundary Layers. *J Fluid Mech* (2003) 478:35–46. doi:10.1017/s0022112002003270
12. Gustavsson H. *Introduction to Turbulence*. Luleå: Division of Fluidmechanics, Luleå University of Technology (2006).
13. Lozano-Durán A, Flores O, Jiménez J. The Three-Dimensional Structure of Momentum Transfer in Turbulent Channels. *J Fluid Mech* (2012) 694:100–30. doi:10.1017/jfm.2011.524
14. Aguilar-Fuertes JJ, Noguero-Rodríguez F, Jaen Ruiz JC, García-RAffi LM, Hoyas S. Tracking Turbulent Coherent Structures by Means of Neural Networks. *Energies* (2021) 14:984. doi:10.3390/en14040984
15. Hochreiter S, Schmidhuber J. LSTM Can Solve Hard Long Time Lag Problems. In: MC Mozer, M Jordan, T Petsche, editors. *Advances in Neural Information Processing Systems*, Vol. 9. Cambridge, MA, USA: MIT Press (1996).
16. Adrian RJ, Meinhart CD, Tomkins CD. Vortex Organization in the Outer Region of the Turbulent Boundary Layer. *J Fluid Mech* (2000) 422:1–54. doi:10.1017/s0022112000001580
17. del Álamo JC, Jiménez J, Zandonade P, Moser RD. Self-similar Vortex Clusters in the Turbulent Logarithmic Region. *J Fluid Mech* (2006) 561:329–58.
18. Jiménez J. Coherent Structures in wall-bounded Turbulence. *J Fluid Mech* (2018) 842:P1. doi:10.1017/jfm.2018.144
19. Rudin C. Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. *Nat Mach Intell* (2019) 1: 206–15. doi:10.1038/s42256-019-0048-x
20. Vinuesa R, Sirmacek B. Interpretable Deep-Learning Models to Help Achieve the Sustainable Development Goals. *Nat Mach Intell* (2021) 3:926. doi:10.1038/s42256-021-00414-y
21. Bottou L. Stochastic Gradient Descent Tricks. In: *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer (2012). p. 421436–6. doi:10.1007/978-3-642-35289-8_25
22. Srinivasan PA, Guastoni L, Azizpour H, Schlatter P, Vinuesa R. Predictions of Turbulent Shear Flows Using Deep Neural Networks. *Phys Rev Fluids* (2019) 4: 054603. doi:10.1103/physrevfluids.4.054603
23. Eivazi H, Guastoni L, Schlatter P, Azizpour H, Vinuesa R. Recurrent Neural Networks and Koopman-Based Frameworks for Temporal Predictions in Turbulence. *Int J Heat Fluid Flow* (2020) 90:108816.
24. Guastoni L, Güemes A, Ianiro A, Discetti S, Schlatter P, Azizpour H, et al. Convolutional-network Models to Predict wall-bounded Turbulence from wall Quantities. *J Fluid Mech* (2021) 928:A27. doi:10.1017/jfm.2021.812
25. Güemes A, Discetti S, Ianiro A, Sirmacek B, Azizpour H, Vinuesa R. From Coarse wall Measurements to Turbulent Velocity fields through Deep Learning. *Phys Fluids* (2021) 33:075121.
26. Eivazi H, Clainche SL, Hoyas S, Vinuesa R. Towards Extraction of Orthogonal and Parsimonious Non-linear Modes from Turbulent Flows. *Preprint arXiv: 2109.01514* (2021).
27. Jiang C, Vinuesa R, Chen R, Mi J, Laima S, Li H. An Interpretable Framework of Data-Driven Turbulence Modeling Using Deep Neural Networks. *Phys Fluids* (2021) 33:055133. doi:10.1063/5.0048909
28. Brunton SL, Noack BR, Koumoutsakos P. Machine Learning for Fluid Mechanics. *Annu Rev Fluid Mech* (2020) 52:477–508. doi:10.1146/annurev-fluid-010719-060214
29. Schmekel D. *Predicting Coherent Turbulent Structures with*. Master's thesis. Stockholm, Sweden: KTH, Royal institute of technology (2022).
30. Lozano-Durán A, Jiménez J. Effect of the Computational Domain on Direct Simulations of Turbulent Channels up to Reτ = 4200. *Phys Fluids* (2014) 26: 011702. doi:10.1063/1.4862918
31. Lluesma-Rodríguez F, Hoyas S, Peréz-Quiles M. Influence of the Computational Domain on DNS of Turbulent Heat Transfer up to Re_τ = 2000 for Pr = 0.71. *Int J Heat Mass Transfer* (2018) 122:983–92.
32. Lluesma-Rodríguez F, Álcantara-Ávila F, Pérez-Quiles MJ, Hoyas S. A Code for Simulating Heat Transfer in Turbulent Channel Flow. *Mathematics* (2021) 9:756.
33. Avsarkisov V, Hoyas S, Oberlack M, García-Galache JP. Turbulent Plane Couette Flow at Moderately High reynolds Number. *J Fluid Mech* (2014) 751: R1. doi:10.1017/jfm.2014.323
34. Avsarkisov V, Oberlack M, Hoyas S. New Scaling Laws for Turbulent Poiseuille Flow with wall Transpiration. *J Fluid Mech* (2014) 746:99–122. doi:10.1017/jfm.2014.98
35. Kraheberger S, Hoyas S, Oberlack M. DNS of a Turbulent Couette Flow at Constant wall Transpiration up to. *J Fluid Mech* (2018) 835:421–43. doi:10.1017/jfm.2017.757
36. Alcántara-Ávila F, Hoyas S, Jezabel Pérez-Quiles M. Direct Numerical Simulation of thermal Channel Flow for and. *J Fluid Mech* (2021) 916:A29. doi:10.1017/jfm.2021.231
37. Oberlack M, Hoyas S, Kraheberger SV, Alcántara-Ávila F, Laux J. Turbulence Statistics of Arbitrary Moments of wall-bounded Shear Flows: A Symmetry Approach. *Phys Rev Lett* (2022) 128:024502. doi:10.1103/PhysRevLett.128.024502
38. Kim J, Moin P, Moser R. Turbulence Statistics in Fully Developed Channel Flow at Low Reynolds Number. *J Fluid Mech* (1987) 177:133–66. doi:10.1017/s0022112087000892
39. Lele SK. Compact Finite Difference Schemes with Spectral-like Resolution. *J Comput Phys* (1992) 103:16–42. doi:10.1016/0021-9991(92)90324-r
40. Spalart PR, Moser RD, Rogers MM. Spectral Methods for the Navier-Stokes Equations with One Infinite and Two Periodic Directions. *J Comput Phys* (1991) 96:297–324. doi:10.1016/0021-9991(91)90238-g
41. Gandía-Barberá S, Alcántara-Ávila F, Hoyas S, Avsarkisov V. Stratification Effect on Extreme-Scale Rolls in Plane Couette Flows. *Phys Rev Fluids* (2021) 6. doi:10.1103/PhysRevFluids.6.034605
42. Fukushima K, Miyake S. Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition. In: *Competition and Cooperation in Neural Nets*. Berlin, Germany: Springer (1982). p. 267–85. doi:10.1007/978-3-642-46466-9_18
43. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2016). p. 770–8. doi:10.1109/cvpr.2016.90
44. Medsker LR, Jain L. Recurrent Neural Networks. *Des Appl* (2001) 5:64–7.
45. Nwankpa C, Ijomah W, Gachagan A, Marshall S. Activation Functions: Comparison of Trends in Practice and Research for Deep Learning. *arXiv preprint arXiv:1811.03378* (2018).
46. Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: International

conference on machine learning. New York City, NY, USA: PMLR (2015). p. 448–56.
47. Santurkar S, Tsipras D, Ilyas A, Madry A. How Does Batch Normalization Help Optimization? In: Proceedings of the 32nd international conference on neural information processing systems (2018). p. 2488–98.
48. Bengio S, Vinyals O, Jaitly N, Shazeer N. Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. *arXiv preprint arXiv:1506.03099* (2015).
49. Lin T-Y, Goyal P, Girshick R, He K, Dollar P. Focal Loss for Dense Object Detection. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017). doi:10.1109/iccv.2017.324
50. Ronneberger O, Fischer P, Brox T. U-net: Convolutional Networks for Biomedical Image Segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Berlin, Germany: Springer (2015). p. 234–41. doi:10.1007/978-3-319-24574-4_28
51. Liu M, Zeng A, Lai Q, Xu Q. Time Series Is a Special Sequence: Forecasting with Sample Convolution and Interaction. *CoRR abs/2106.09305* (2021).
52. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention Is All You Need. *Adv Neural Inf Process Syst* (2017) 30.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers | Frontiers in Applied Mathematics and Statistics

ORIGINAL RESEARCH
published: 29 April 2022
doi: 10.3389/fams.2022.879140

Convolutional Neural Networks for Very Low-Dimensional LPV Approximations of Incompressible Navier-Stokes Equations

Jan Heiland[1,2*†], *Peter Benner*[1,2†] *and Rezvan Bahmani*[3]

[1] *Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany,* [2] *Faculty of Mathematics, Otto von Guericke University Magdeburg, Magdeburg, Germany,* [3] *School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran, Iran*

The control of general nonlinear systems is a challenging task in particular for large-scale models as they occur in the semi-discretization of partial differential equations (PDEs) of, say, fluid flow. In order to employ powerful methods from linear numerical algebra and linear control theory, one may embed the nonlinear system in the class of linear parameter varying (LPV) systems. In this work, we show how convolutional neural networks can be used to design LPV approximations of incompressible Navier-Stokes equations. In view of a possibly low-dimensional approximation of the parametrization, we discuss the use of deep neural networks (DNNs) in a semi-discrete PDE context and compare their performance to an approach based on proper orthogonal decomposition (POD). For a streamlined training of DNNs directed to the PDEs in a *Finite Element* (FEM) framework, we also discuss algorithmical details of implementing the proper norms in general loss functions.

Keywords: model reduction and model simplification, Navier-Stokes equation, data driven learning, linear parameter varying (LPV), convolutional neural network

AMS subject classifications: 65M22, 76D05.

Edited by:
Traian Iliescu,
Virginia Tech, United States

Reviewed by:
Omar Abu Arqub,
Al-Balqa Applied University, Jordan
Maria Strazzullo,
Politecnico di Torino, Italy

***Correspondence:**
Jan Heiland
heiland@mpi-magdeburg.mpg.de

***ORCID:**
Jan Heiland
orcid.org/0000-0003-0228-8522
Peter Benner
orcid.org/0000-0003-3362-4103

Specialty section:
This article was submitted to Statistical and Computational Physics, a section of the journal Frontiers in Applied Mathematics and Statistics

Received: *18 February 2022*
Accepted: *22 March 2022*
Published: *29 April 2022*

Citation:
Heiland J, Benner P and Bahmani R (2022) Convolutional Neural Networks for Very Low-Dimensional LPV Approximations of Incompressible Navier-Stokes Equations. Front. Appl. Math. Stat. 8:879140. doi: 10.3389/fams.2022.879140

NOVELTY STATEMENT

- Conceptual: Due to the quadratic nature of the Navier-Stokes equations, any encoder-decoder with a linear decoding part provides an affine LPV approximation of the state-space equations. We propose the use of convolutional neural networks (CNNs).
- Algorithmical: An efficient realization of the correct FEM norms within the training of a neural network. As a result, we provide basic routines that combine the *Finite Element* package *FEniCS* and the *Machine Learning* toolbox *PyTorch*.
- Numerically: A very low-dimensional, that is 3-dimensional, performant LPV approximation of a flow around a cylinder in the vortex shedding regime.

1. INTRODUCTION

The computer-aided controller design for a nonlinear control system

$$\dot{v} = f(v) + Bu$$

with an input u and an input operator B typically resorts to system insights (like in backstepping [1], feedback linearization [2, Ch. 5.3], or sliding mode control [3]), or the repeated computation

for subobtimal control laws like in model predictive control (MPC) [4]. The holistic but general approach via the Hamilton-Jacobi-Bellmann (HJB) equations is only feasible for very moderate system sizes or calls for model order reduction; see e.g., Breiten et al. [5] for a relevant discussion and an application in fluid flow control.

For general large-scale systems, MPC schemes seem to be a good choice since modern hardware and optimization algorithms can well mitigate the computational complexity while the continuous update of the prediction realizes a feedback loop as it is needed to react on inevitable perturbations in simulations and measurements. Nonetheless, stability guarantees for MPC schemes are difficult to establish a priori and the solving of nonlinear optimization problems at runtime limits their performance in particular for large-scale systems.

In view of these two limiting factors, alternatives are presented by methods that base on *extended linearizations* or *state-dependent coefficients* (SDC) (see e.g., Banks et al. [6]) schemes that are particular realizations of the representation of a nonlinear model as a *linear parameter varying* (LPV) system.

In an exact SDC representation, the flow f of the model is factorized as

$$f(v) = N(v)\, v$$

with a suitable $A\colon \mathbb{R}^n \to \mathbb{R}^{n,n}$ which exists under mild conditions. The SDC is a special case of an LPV representation

$$f(v) = \tilde{N}(\rho(v))\, v$$

with $\rho\colon \mathbb{R}^n \to \mathbb{R}^r$ and $N\colon \mathbb{R}^r \to \mathbb{R}^{n,n}$ suitably chosen and, possibly, $r < n$.

While these representations are exact reformulations of the model, a low-dimensional ($r \ll n$) and affine-linear parameter dependency might only exist as an approximation

$$f(v) \approx [N_0 + \sum_{i=1}^{r} \rho_i(v) N_i]\, v.$$

If an approximation can be afforded, many numerical approaches for the derivation of low-dimensional LPV representations apply. In fact, any model order reduction scheme that encodes the state in a reduced coordinate $\rho = \mu(v) \in \mathbb{R}^k$ and lifts it back to $\tilde{v} = \lambda(\rho)$ can turn an SDC representation into a low-dimensional LPV approximation via

$$f(v) = N(v)\, v \approx N(\tilde{v})\, v = N(\lambda(\rho))\, v =: \tilde{N}(\rho)\, v.$$

Even more, if the state-dependent coefficient matrix N is affine-linear in its argument and if the lifting λ is affine-linear, then the resulting LPV approximation is affine-linear. We will make use of this observation when discussing the Navier-Stokes equations and when designing the low-dimensional encodings.

An immediate advantage in view of large-scale systems is that for these pointwise linear problems, linear methods for controller design apply. Generally, an a-priori proof that a controller will stabilize the system is by no means easier in an LPV context. Nonetheless, conditions that can be checked or monitored numerically have been developed; see e.g., Benner and Heiland [7] for a result on SDC systems or [8] for a result for (affine) LPV systems.

This paper investigates the use of *convolutional neural networks* in combination with bases obtained from a POD to design such approximative LPV systems with affine parameter dependency:

$$\dot{v} = [A_0 + \sum_{i=1}^{r} \rho_i(v) A_i]\, v + Bu. \tag{1}$$

We focus on Navier-Stokes equations but the methodology applies to any system with states that are distributed in a spatial domain like spatially discretized approximations to PDE models.

The motivation for this study is the potential use of low-dimensional LPV representations in controller design. For example, for affine-linearly parametrizable coefficients as in Equation (1), one can derive series expansions (see e.g., Beeler et al. [9]) of the solution to the associated parameter-dependent Riccati equations and exploit them for efficient controller design; cp. [10]. Furthermore, if the image of ρ for the given system can be confined to a polygon, then one can provide a globally stabilizing controller (see e.g., Apkarian et al. [11]) through the *scheduling* of a set of linear controllers. Both approaches, however, hinge on a small dimension of $\rho(v)$ since the series expansion has to be considered in all parameter directions and since the scheduling requires the solution of a coupled system of r linear matrix inequalities of the size of the system dimension.

In view of these considerations, this work provides a particular solution to the following general problem:

Problem 1. Given a nonlinear system $\dot{v} = f(v) + Bu$,

(a) how to encode a current state $v(t) \in \mathbb{R}^n$ in a low dimensional parameter $\rho(t) \in \mathbb{R}^r$ and
(b) how to provide embeddings $\rho \to \tilde{N}(\rho) = N_0 + \sum_{i=1}^{r} \rho_i N_i \in \mathbb{R}^{n,n}$ so that

$$f(v) \approx \tilde{N}(\rho(v))\, v.$$

Existing general solutions for this task are known to result in larger dimensions of the parametrization ρ; see Koelewijn and Tóth [12] for relevant references and a *neural network* based approach toward a reduced order of ρ.

In any case, the existing strategies were designed for ODE models of moderate size rather than the treatment of high-dimensional nonlinear models that are associated with PDEs.

Therefore, we propose the use of model reduction techniques to derive LPV approximations with low parameter dimensions independently of the system size. Similar efforts can be spotted in earlier works (see e.g., Hashemi and Werner [13] where the Burgers' equation was considered) though with a different strategy: the model reduction techniques were used for reducing the overall system so that the natural SDC representation could be interpreted as a low-dimensional LPV approximation.

In what we propose, however, the system dimensions are not touched in order to ensure *accuracy* and *feature-completeness*,

but only parts of the nonlinear functions are replaced by de– and encoded variables to provide the low-dimensional LPV representation. Certainly, if controllers are to be designed, a state-space reduction might be necessary but can then be directed to the purpose of the controller model rather than the actual state equations.

In a regime that is dominated by convection, the encoding of a state of a Navier-Stokes equation in a very-low dimensional coordinate system cannot be simply done by a linear projection. This has been observed in numerical studies of flow problems and specifically analyzed for equations with wave like patterns [14, 15].

Successful low-dimensional parametrizations for convective phenomena have been established using a nonlinear preprocessing like the detection and explicit treatment of wave patterns; see e.g., Reiss et al. [16] for a method of adaptive shifting of POD modes along with wave fronts and [17] for a recent update that resorts to neural networks. A more generic approach was used in Sarna and Benner [18], where a superposition of the phase space of hyperbolic and parabolic parts was introduced and successfully exploited for efficient reduction of parabolic parts. A purely neural network based appraoch that explicitly addresses wave patterns has been discussed in Deo and Jaiman [19].

Recently, the use of neural networks for finding low-dimensional coordinates has been proposed, e.g., as an alternative to established POD techniques [20–23] or as an enhancement to them [24].

Considering fluid flow or Burgers' equations, it has been observed that neural networks can significantly outperform POD approaches at very-low dimensions in terms of approximation quality; see e.g., Lee and Carlberg [20, Figure 3] or Kim et al. [21, Figure 2]. However, the effort for setting up the surrogate model (cp. [24, Table 3] or [20, Section 8]) as well as the evaluation at runtime can be inferior to a plain POD approach; compare, e.g., the reported speed-ups in Kim et al. [21, Table 1].

We note that we can easily tolerate these performance limitations, as the major motivation of our work is to establish a model approximation of a particular structure.

In summary of the preceding considerations, we state that the presented investigations are motivated by and directed to support the following working hypotheses:

Working Hypothesis 1.

1. Neural networks can efficiently encode the state of a PDE and thus provide very low dimensional parametrizations.
2. For the synthesis of a controller model for a nonlinear PDE, the use of high-dimensional data and demanding computations is appropriate.

2. PRELIMINARIES

We briefly introduce the PDE model of interest, the concept of convolutional neural networks, and state relevant observations.

2.1. Navier-Stokes Equations

The incompressible Navier-Stokes equations

$$\frac{\partial}{\partial t}v + (v \cdot \nabla)v - \frac{1}{\mathsf{Re}}\Delta v + \nabla p = f \tag{2a}$$

$$\nabla \cdot v = 0 \tag{2b}$$

is a set of partial differential equations that is widely used to model incompressible fluid flows in a domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, on a time interval $[0, T] \subset \mathbb{R}$ in terms of the evolution of the velocity field $v\colon [0, T] \times \Omega \to \mathbb{R}^d$ and the pressure field $p\colon [0, T] \times \Omega \to \mathbb{R}$. Here, Re is the so-called *Reynolds* number that parametrizes the flow setup and f contains forces that act on the flow like gravity or, in a flow control setup, external inputs.

As we will detail below (in Section 4.1) after a spatial discretization and by means of divergence-free coordinates, the flow model reads

$$\dot{v} + N(v)v + A_0 v = f \tag{3}$$

and is readily expressed as a so-called *state-dependent coefficient* system

$$\dot{v} + N^0(v)v = f, \tag{4}$$

with

$$N^0(v) = A_0 + N(v). \tag{5}$$

Remark 1. The decomposition $N^0(v) = A_0 + N(v)$ is by no means unique. In particular, a similarly natural factorization $N^1(v)\, v := A_0 v + N(v)v$ exists and any combination

$$N^s(v) = sN^1(v) + (1 - s)N^0(v)$$

for a scalar s can be considered, too. Such a blending of the coefficients can be used to improve the model performance as we did in our numerical examples; cp. Remark 5.

In what follows, we will consider LPV systems that generalize *state-dependent coefficient* (SDC) systems by encoding the state in a parameter variable ρ. We will refer to $\rho(v)$ as the code of v and also distinguish an associated *encoder*

$$\mu\colon \mathbb{R}^n \to \mathbb{R}^r, \quad \text{with} \quad \mu(v) = \rho(v).$$

It will be convenient to refer to a *decoder* as

$$\mu^{-1}\colon \mathbb{R}^r \to \mathbb{R}^n, \quad \text{with} \quad \mu^{-1}(\rho) = \tilde{v},$$

by the vague requirement that $\tilde{v} = \mu^{-1}(\rho(v)) \approx v$ for all v of interest although an inverse to μ may not exist and although the inference of μ and μ^{-1} may be unrelated in practice.

Given an encoder μ and a decoder μ^{-1}, an LPV approximation to the state-dependent coefficient (Equation 5) is readily given as

$$N^0(v) \approx A_0 + \tilde{N}(\rho) := A_0 + N(\mu^{-1}(\rho)).$$

Remark 2. A particular property of quadratic systems and, thus, of the Navier-Stokes equations is that the natural choices of the state-dependent coefficient $N(v)$ are linear, i.e., $N(\lambda_1 v_1 + \lambda_2 v_2) = \lambda_1 N(v_1) + \lambda_2 N(v_2)$. Accordingly, if the decoder μ^{-1} is affine-linear, i.e.,

$$\mu^{-1}(\rho) = \tilde{v}(\rho) = \tilde{v}_0 + \sum_{i=1}^{r} \rho_i \tilde{v}_i,$$

for a shift $\tilde{v}_0$ and a some vectors $\{\tilde{v}_1, \ldots, \tilde{v}_r\}$, then the induced LPV representation is affine-linear as

$$A_0 + N(\mu^{-1}(\rho)) = A_0 + N(\tilde{v}_0 + \sum_{i=1}^{r} \rho_i \tilde{v}_i) = A_0 + N(\tilde{v}_0) + \sum_{i=1}^{r} \rho_i N(\tilde{v}_i) =: A_0 + \tilde{N}_0 + \sum_{i=1}^{r} \rho_i \tilde{N}_i.$$

Remark 3. If the decoder μ^{-1} or the SDC relation $v \to A(v)$ is nonlinear, then an additional approximation step is needed for an affine-linear LPV representation; see Koelewijn and Tóth [12].

2.2. Convolutional Neural Networks

Generally, a neural network of N_L *layers* can be expressed as a recursively defined map

$$x^{(\ell)} = \sigma(W^{(\ell)} x^{(\ell-1)} + b^{(\ell)}), \quad \ell = 1, \ldots, N_L$$

that maps the input variable $x^{(0)}$ onto the output variable $x^{(N_L)}$. It is defined in terms of the *layer widths* n_ℓ, the *weights* meaning the coefficients of the matrix $W^{(\ell)} \in \mathbb{R}^{n_\ell, n_{\ell-1}}$ and the *bias term* $b^{(\ell)} \in \mathbb{R}^{n_\ell}$, and the *activation function* $\sigma : \mathbb{R} \to \mathbb{R}$ that is applied componentwise to the vectors $x^{(\ell)} \in \mathbb{R}^{n_\ell}$.

The term of training a neural network refers to determining the *weights* by an optimization toward an optimality criterion (the *loss function*) evaluated at given sample points.

In convolutional neural networks, the linear map $W^{(\ell)}$ in each a layer is defined by a number of convolution kernels that convolve the current state. Typically, a linear contraction follows that merges neighboring states. The advantages of convolutional layers for PDE data is manifold.

- In each layer, the learnable parameters are given by the parameters of the convolution kernels so that the amount of parameters is independent of the possibly large state dimensions
- The convolution acts upon neighboring states which can respect and, notably, detect coherent spatial structures as they may be inherent in states of PDEs.
- The contraction operation reduces the number of variables in each channel in every layer so that a CNN can be designed to provide low-dimensional encodings.

An immediate obstacle that stands against the use of CNNs for PDEs is the need of tensorized grids, whereas a simulation of complex phenomena typically requires a locally refined and unstructured grid. We overcome this issue by simply interpolating the state values from the FEM grid to a tensorized grid.

For an introduction to the techniques of CNNs, we refer to O'Shea and Nash [25]. An application for spatially distributed data as in our case is well explained in Lee and Carlberg [20].

3. IMPLEMENTATION SETUPS

The provision of a low-dimensional affine-linear LPV approximation

$$N(v)\, v \approx [N_0 + \sum_{i=1}^{r} \rho_i(v) N_i]\, v$$

for the Navier-Stokes equations amounts to learning or computing

- an encoder $\mu : v \mapsto \rho$ and
- an embedding or lifting $\lambda : \rho \mapsto [N_0 + \sum_{i=1}^{r} \rho_i(v) N_k]$.

Note that λ can be defined without a decoder μ^{-1}. On the other hand, for the Navier-Stokes case, if an affine-linear decoder is given, then $\lambda : \rho \mapsto N(\mu^{-1}(\rho))$ readily provides an affine-linear parametrization; cp. Remark 2.

3.1. POD Parametrization

As a benchmark and for later use as a basis for the decoding, we consider the LPV representation that is induced by a POD reduction. Here, one uses a projection basis

$$\tilde{V}_p = \begin{bmatrix} \tilde{v}_1 & \tilde{v}_2 & \ldots & \tilde{v}_r \end{bmatrix} \tag{6}$$

that consists of the r leading singular vectors of a matrix of snapshots like

$$V = \begin{bmatrix} v_1 & v_2 & \ldots & v_k \end{bmatrix}. \tag{7}$$

The *POD* reduction itself bases on the property that the projection $V_p V_p^T$ minimizes the average projection error over the given data set (Equation 7), meaning that

$$\frac{1}{k} \sum_{j=1}^{k} \| v_j - \tilde{V}_p \tilde{V}_p^T v_j \|_M$$

is minimal over all r-dimensional linear projections of the data set, where the subscript M stands for a weight in the norm induced, e.g., by the symmetric positive (mass) matrix of an underlying an FEM scheme; cp. [26].

Accordingly, with

$$\tilde{v} \approx \tilde{V}_r \tilde{V}_r^T v =: \tilde{V}_r \rho,$$

the POD basis V_r defines a r-dimensional encoding via $\mu : v \mapsto \tilde{V}_r^T v$, a decoding via $\mu^{-1} : \rho \mapsto V_r \rho$, and an embedding λ for the

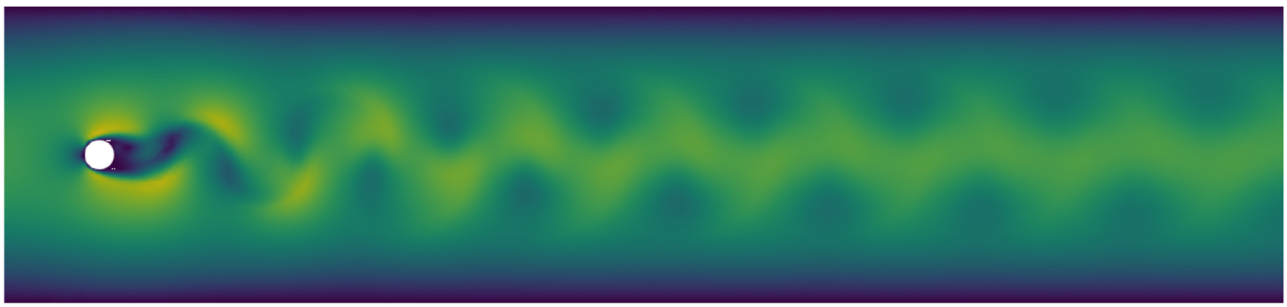


FIGURE 1 | Snapshot of the domain and the magnitude of the fully developed velocity field at Re = 40.

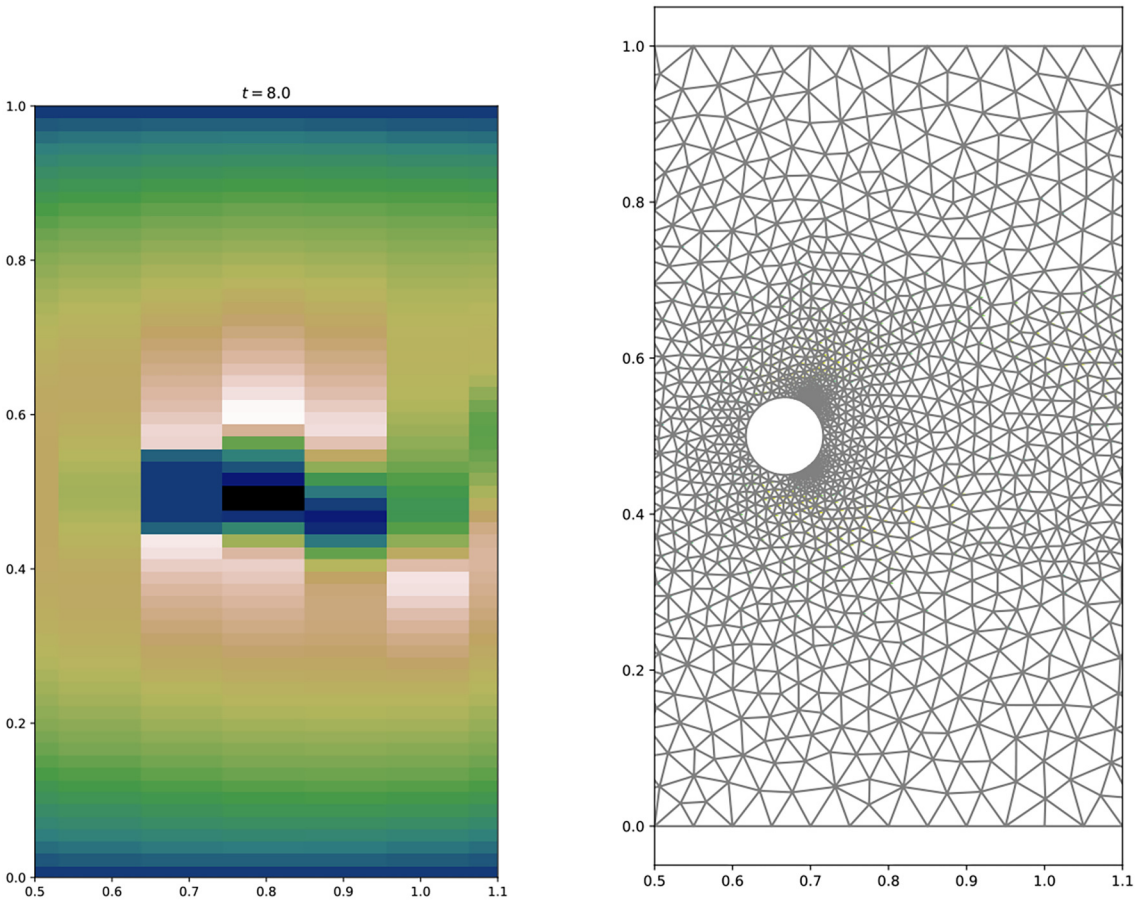


FIGURE 2 | Close up view of the computational domain with the FEM mesh (right) and the data representation on the tensorized mesh (left).

LPV approximation via

$$N(\upsilon)\, v \;\approx\; N(\tilde{V}_r \tilde{V}_p^{\mathsf{T}} v)\, v = N(\tilde{V}_r \rho)\, v = [\sum_{i=1}^{r} \rho_i N(\tilde{\upsilon}_i)]$$

$$v =: [\sum_{i=1}^{r} \rho_i \tilde{N}_i]\, v.$$

Remark 4. As it is common practice, for a better approximation quality and for consistency reasons, the data for the POD and, thus, also the POD bases, are shifted by a vector vs. which will be chosen to be the initial value in the simulations. Accordingly, the correct reconstruction reads $\tilde{\upsilon}(t) = \tilde{V} W \rho(t) + \upsilon_s$, which simply adds a constant and a few linear terms to the approximation or the corresponding loss functions.

3.2. Encoding of POD Coordinates

In this setup, we investigate whether a CNN can replace the POD encoding of the state

$$\upsilon \to (CNN) \to \rho \in \mathbb{R}^r$$

and also provide an enhanced decoding to POD coordinates via a full linear map W

$$\rho \to (W) \to \tilde{\rho} \in \mathbb{R}^{\tilde{r}}, \quad \tilde{r} > r,$$

and the embedding via the p-dimensional POD basis $\tilde{\upsilon} := \tilde{V}_{\tilde{r}}\tilde{\rho}$.

In this approach, the CNN and the matrix $W \in \mathbb{R}^{\tilde{r},r}$ is learned as neural network

$$\upsilon \to (CNN) \to \rho \to (W) \to \tilde{\rho} \in \mathbb{R}^{\tilde{r}}$$

with the loss function

$$l(v, \rho) := \|v - \tilde{V}_{\tilde{r}} W \tilde{\rho}\|_M.$$

If $r < \tilde{r}$, and the resulting embedding outperforms the standard $\tilde{r}$-dimensional POD reduction, then this approach provides a countable improvement in terms of dimensionality.

Moreover, the loss functions can be changed in order to direct the learning to best approximate the resulting convective behavior as described in the following subsection.

3.3. Convection-Informed Encoding of POD Coordinates

With the same approach but with

$$l(v, \rho) := \|N(\upsilon)\upsilon - N(\tilde{V}_{\tilde{r}} W \rho)\upsilon\|_{M^{-1}}.$$

as the loss function, the training of the decoder can be directed toward the actual goal—the low-dimensional parametrization of the convection part. Note the M^{-1} norm, that is the discrete version of the norm of $N(\upsilon)$ as a functional on the state space $L^2(\Omega)$.

4. IMPLEMENTATION ISSUES

In this section, we discuss implementation issues as the arise in the numerical treatment of incompressible Navier-Stokes equations by finite elements and the inclusion of FEM-norms in learning algorithms.

4.1. Semi-discretization, Divergence-Free Coordinates, and Boundary Conditions

A spatial discretization (see e.g., Behr et al. [27]) of the incompressible Navier-Stokes equations (2) leads to a system of type

$$\begin{aligned} M\dot{v} + N(\upsilon)\upsilon + A_0 v - J^{\mathsf{T}} p &= f, \\ Jv &= g, \end{aligned}$$

where M is a positive definite (mass) matrix, where $N(\upsilon)$ is the matrix that realizes the convection for a state $\upsilon(t)$, where A_0 encodes the diffusion part, where J^{T} and J stand for the discrete gradient and divergence operator, and where the vectors f and g accommodate possible inhomogenities and boundary conditions.

TABLE 1 | Table of parameters for the CAE-model.

Parameter	Description	Value in simulation
`cs`	Code size	`{3, 5, 8}`
`k`	Dimension of the POD basis for the decoding	`15`
`#layers`	Number of convolutional layers	`4`
`#channels`	Number of channels in each layer (including the input layer)	`(2)-4-8-10-12`
`kernel size`	The size of the convolution kernels in each layer	`5 x 5`
`stride`	The stride in both spatial directions[a]	`2`
`activation`	The the nonlinear activation function	`torch.ELU`

[a] *The factor by which the data is condensed after each convolution.*

In order to eliminate the inhomogeneity and possibly nonzero boundary conditions, one may shift the state by some vector vs. that fulfills the boundary conditions and the algebraic constraint, i.e., $J\upsilon_s = g$, so that the shifted system for $\upsilon_d(t) = \upsilon(t) - \upsilon_s$ reads

$$\begin{aligned} M\dot{\upsilon}_d + N(\upsilon_d)\upsilon_d + \bar{A}_0\upsilon_d - J^{\mathsf{T}} p &= \bar{f} \\ J\upsilon_d &= 0, \end{aligned}$$

where

$$\begin{aligned} \bar{A}_0\upsilon_d &:= A_0\upsilon_d + N(\upsilon_s)\upsilon_d + N(\upsilon_d)\upsilon_s \\ \text{and} \quad \bar{f} &:= f - A_0\upsilon_s - N(\upsilon_s)\upsilon_s. \end{aligned}$$

Finally, with the reasonable assumption that $J^T M^{-1} J$ is invertible, we find that with the projector $\Pi = I - M^{-1}J^{\mathsf{T}}(JM^{-1}J^{\mathsf{T}})J$ it holds that $\upsilon_d = \Pi\upsilon_d$ and that the solution υ_d is completely defined through the projected system (see e.g., Heiland [28, Thm. 8.6])

$$M\upsilon_d + \Pi^{\mathsf{T}} N(\upsilon_d)\upsilon_d + \Pi^{\mathsf{T}}\bar{A}_0\upsilon_d = \Pi^{\mathsf{T}}\bar{f}.$$

The practical implications are as follows: in a simulation, one needs to consider all data shifted by a constant vector vs. that fulfills the boundary conditions so that the snapshots υ_i can be assumed to comply with zero Dirichlet conditions. Then a reduced parametrization will target the shifted space with zero boundary conditions and can be lifted to the physical space by adding vs. again.

Generally, the projection Π needs not be computed explicitly as it will be implicitly realized during the time integration; [29]. However, if only the velocity is of interest, the model could be trained to best approximate $\Pi^{\mathsf{T}} N(\upsilon)\upsilon$ which resides on a submanifold of dimension (n_v − rank Π). If however, the pressure is of interest too, the LPV approximation should be trained toward a good representation of

$$N(\upsilon)\upsilon = \Pi^{\mathsf{T}} N(\upsilon)\upsilon + (I - \Pi^{\mathsf{T}})N(\upsilon)\upsilon$$

as the part $(I - \Pi^{\mathsf{T}})$ defines how the convection enters the pressure approximation.

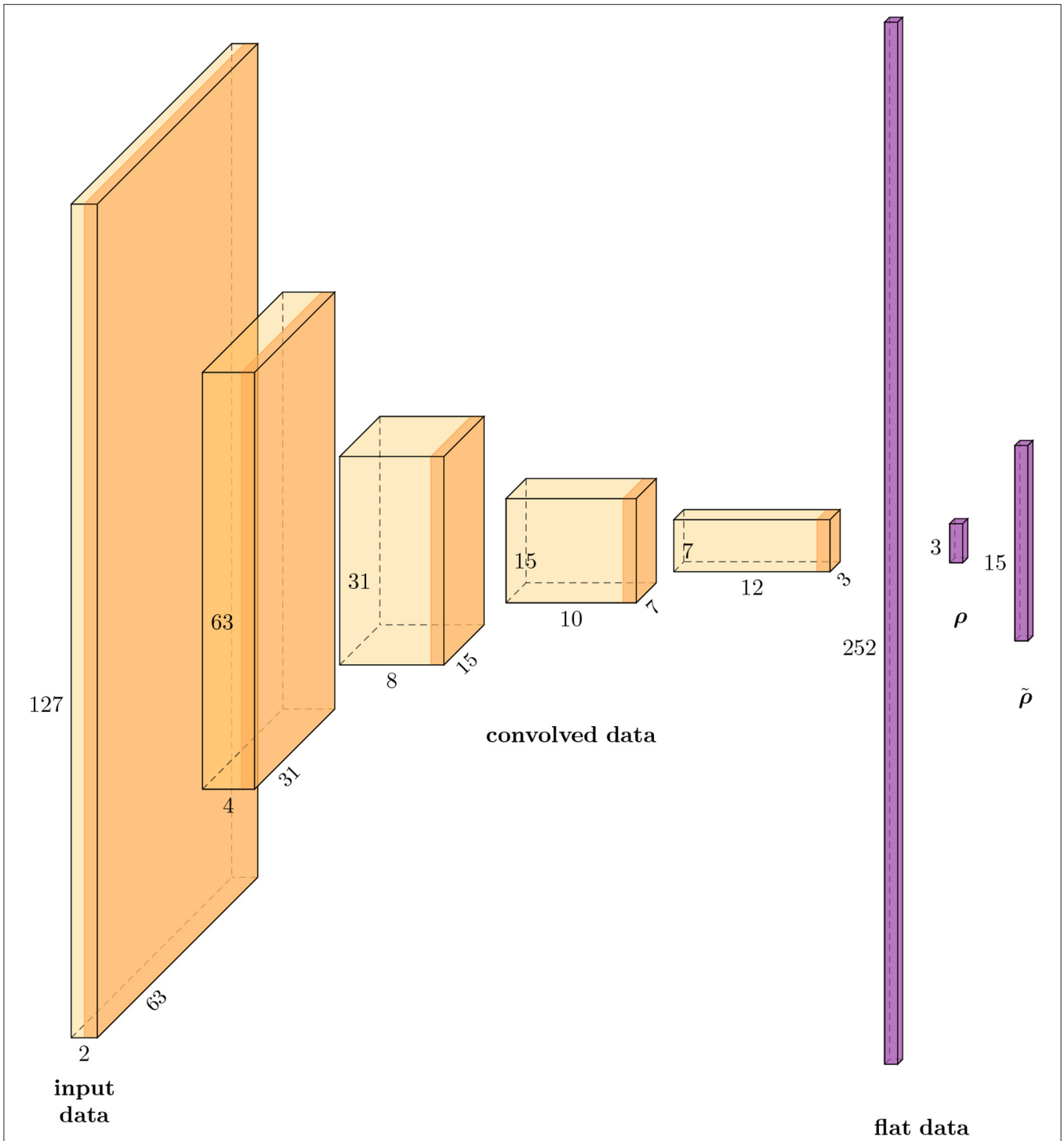


FIGURE 3 | Schematic illustration of the data sizes in each layer of the CAE-3 network that was used in the numerical example: Four convolutional layers, one reshape of the data to the vector (flat data), a full linear layer that reduces the data to $\boldsymbol{\rho}$ and a linear layer without activation for the decoding to $\tilde{\boldsymbol{\rho}}$.

4.2. Interpolation to Tensor Grids

As mentioned above, in order to employ standard CNNs, the data on an FEM mesh has to be interpolated to a tensorized mesh. For that and for a generic 2D flow setup we proceed as follows.

Let $\Omega \subset \mathbb{R}^2$ be the computational domain and let $(\xi_1, \xi_2) \in \Omega$ denote the spatial coordinates. Let

$$\mathcal{V} := \{\phi_1, \phi_2, \ldots, \phi_{n_v}\} \in L^2(\Omega; \mathbb{R}^2)$$

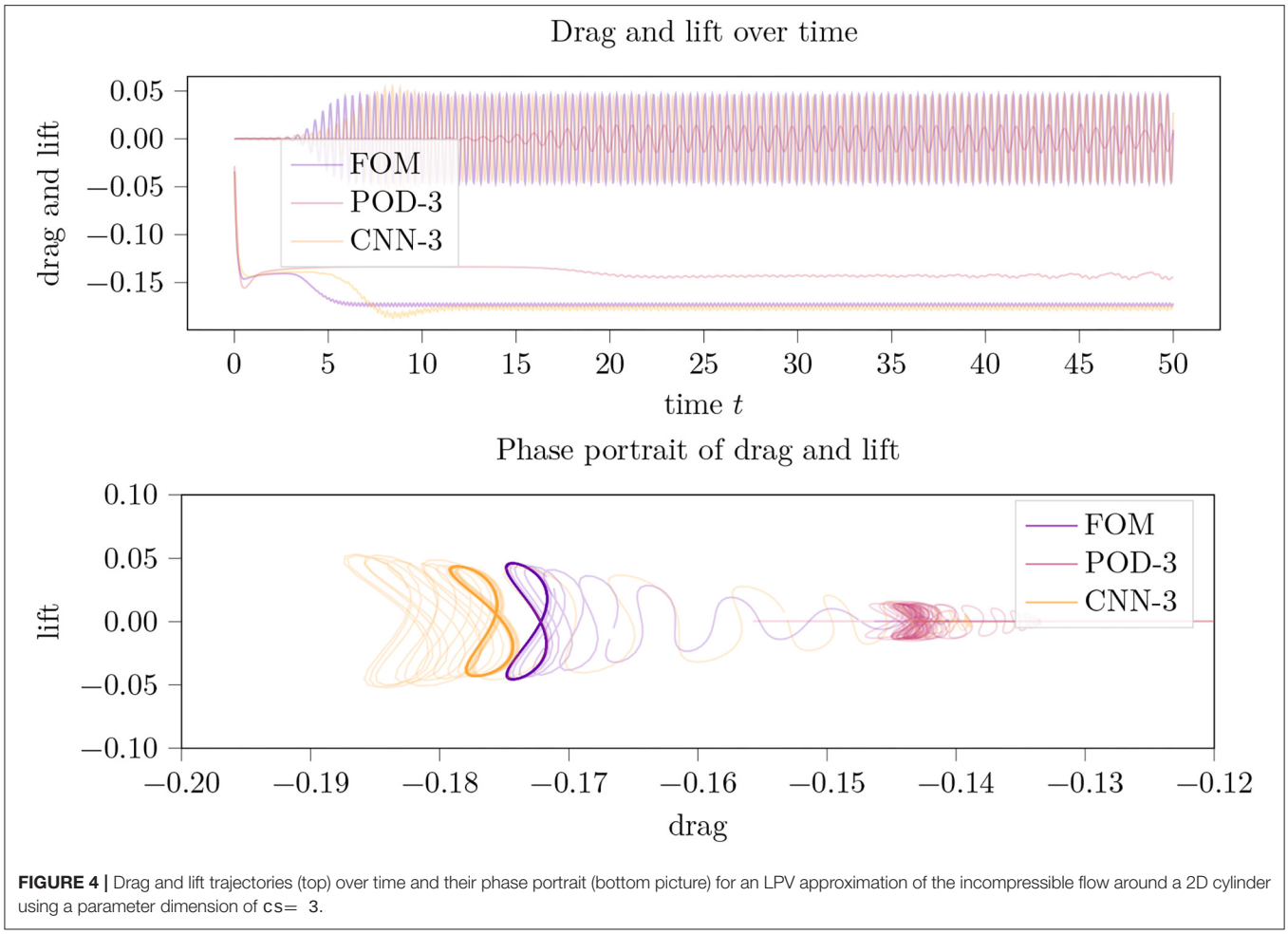


FIGURE 4 | Drag and lift trajectories (top) over time and their phase portrait (bottom picture) for an LPV approximation of the incompressible flow around a 2D cylinder using a parameter dimension of cs= 3.

be the ansatz space of the finite element discretization. Then every solution snapshot v_i has the function representation via

$$v_i(\xi_1, \xi_2) = \sum_{j=1}^{n_v} [v_i]_j \phi_j(\xi_1, \xi_2) \subset \mathbb{R}^2$$

where $[v_i]_j$ is the j-th component of the vector of coefficients v_i. Accordingly, it can be interpolated onto a tensorization

$$\mathbb{T} = \{(x_j, y_k) : j = 1, \ldots, n_x,\ k = 1, \ldots, n_y\}$$

of two 1D grids

$$\{x_1 < x_2 < \cdots < x_{n_y}\} \quad \text{and} \quad \{y_1 < y_2 < \cdots < y_{n_y}\}$$

into a $2 \times n_x \times n_y$ tensor $\mathbf{v}_i$ as

$$[\mathbf{v}_i]_{\ell jk} = \begin{cases} [v_i(x_j, y_k)]_\ell, & \text{if } (x_j, y_k) \in \Omega \\ 0, & \text{elsewhere} \end{cases}.$$

We denote this interpolation operator with the operator $\mathbf{P}\colon \mathbb{R}^{n_v} \to \mathbb{R}^{2 \times n_x \times n_y}$. The application of $\mathbf{P}$ to the data points v_i is the first operation in the processing of the v_i's in a CNN.

4.3. Realization of the FEM Norms in the Loss Functions

For the implementation of the learning based on FEM data, we need to include M or M^{-1} in the loss functions without breaking the automated computation of sensitivities[1]. For the realization of the M norm, we can resort to sparse factorizations $M = FF^T$ and use the equivalence of $\|v\|_M = \|F^T v\|_2$. In this way, the M-norm can realized in standard ML packages that have the 2-norm implemented as the *mean square error* loss function and that also support sparse matrix multiplication.

For the realization of M^{-1}, where no sparse factorization can be provided, we compute a M^{-1} optimal (cp. [26, Lem. 2.5]) snapshot basis $L \in \mathbb{R}^{n_v, k_c}$ for $N(v_i)v_i$, $i = 1, \ldots, k$ of dimension k_c. With that we can best approximate

$$\|N(v_i)v_i\|_{M^{-1}} \approx \|LL^T F^{-1} N(v_i)v_i\|_2 = \|L^T F^{-1} N(v_i)v_i\|_2$$

[1] A particularly powerful feature of DNN architectures that comes with the explicit formulation of DNNs in terms of fundamental functions is that the gradients of the current realization with respect to the parameters can be computed by algorithmic differentiations in a straight-forward way. All packages for neural networks make use of this functionality during the training of the network. Once, new functional dependencies are introduced, e.g., in the loss function, one has to take care that this so called *back propagation* of gradients is maintained.

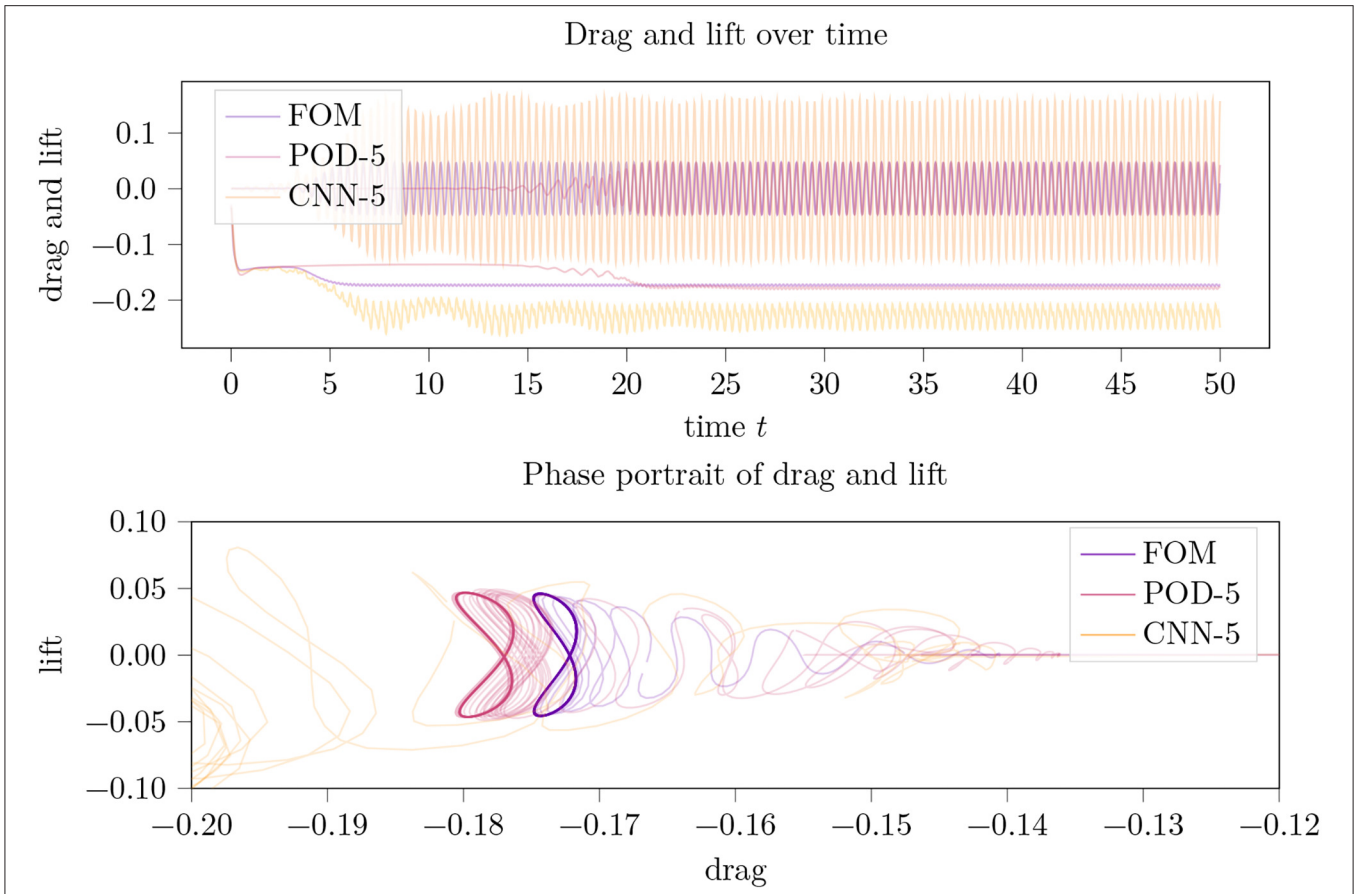


FIGURE 5 | Drag and lift trajectories (top) over time and their phase portrait (bottom picture) for an LPV approximation of the incompressible flow around a 2D cylinder using a parameter dimension of `cs= 5`. In the phase portrait plot, parts of CAE-5 were truncated.

where F is a factor of $M = FF^{\mathsf{T}}$ and where we have used that L is orthogonal so that it does not affect the 2-norm. In this way, the M^{-1} norm of $N(\upsilon_i)\upsilon_i$ can be well approximated with the standard mean-squared error and a premultiplication by the dense matrix $L^T F^{-1} \in \mathbb{R}^{k_c, n_v}$.

5. NUMERICAL EXAMPLE

We consider the well-known benchmark example of a 2D flow around a cylinder in a channel. In nondimensionalized coordinates the channel covers the rectangle $[0, 5] \times [0, 1]$ with the cylinder of radius $\mathsf{R} = 0.05$ centered at $(\frac{2}{3}, 0.5)$. The regime is parametrized by the Reynolds number Re that is computed using the velocity of the inflow parabola averaged over the inflow boundary and the radius and set to $\mathsf{Re} = 40$ in the presented simulations. As the starting value for $t = 0$, we impose the associated steady-state *Stokes* solution. With this initialization the flow immediately starts the transition into the characteristic periodic *vertex-shedding* regime which seems to be well developed at around $t = 8$. A snapshot of the domain and the developed flow at $t = 8$ is presented in **Figure 1**.

We will use data from the initial phase from $t = 0$ till $t = 8$ to generate low-dimensional LPV models. The performance of the reduced-order models will be directed toward how well the periodic regime is captured on a time frame till $t = 50$.

For the spatial discretization, we use $P_2 - P_1$ *Taylor-Hood* finite elements on a nonuniform grid that results in `42764` degrees of freedom in the velocity approximation. For the time integration we use the implicit one-step *Crank-Nicolson* scheme for the linear part and the explicit 2-step *Adams-Bashforth* scheme for the nonlinear part which combines into a 2nd order approximation. The finite element discretization was realized in the FEM toolbox `FEniCS` [30], the time integration and the connection to `PyTorch` (which was used for the setup, training and evalution of the (C)NNs) was handled via the *SciPy* interface `dolfin-navier-scipy` [31].

The solution is monitored via the induced forces onto the cylinder periphery that we compute by testing the (numerically computed) residual of the FEM solution $(\upsilon(t), p(t))$ against (a numerical realization of) the function ϕ that takes on the value $(1, 1)$ at the cylinder boundary and $(0, 0)$ elsewhere:

$$\mathcal{F}(t) = \int_\Omega [(\upsilon(t) \cdot \nabla)\upsilon(t) - \frac{1}{\mathsf{Re}}\Delta\upsilon(t) + \nabla p(t) - f(t)]\phi \, \mathrm{d}\xi, \quad (8)$$

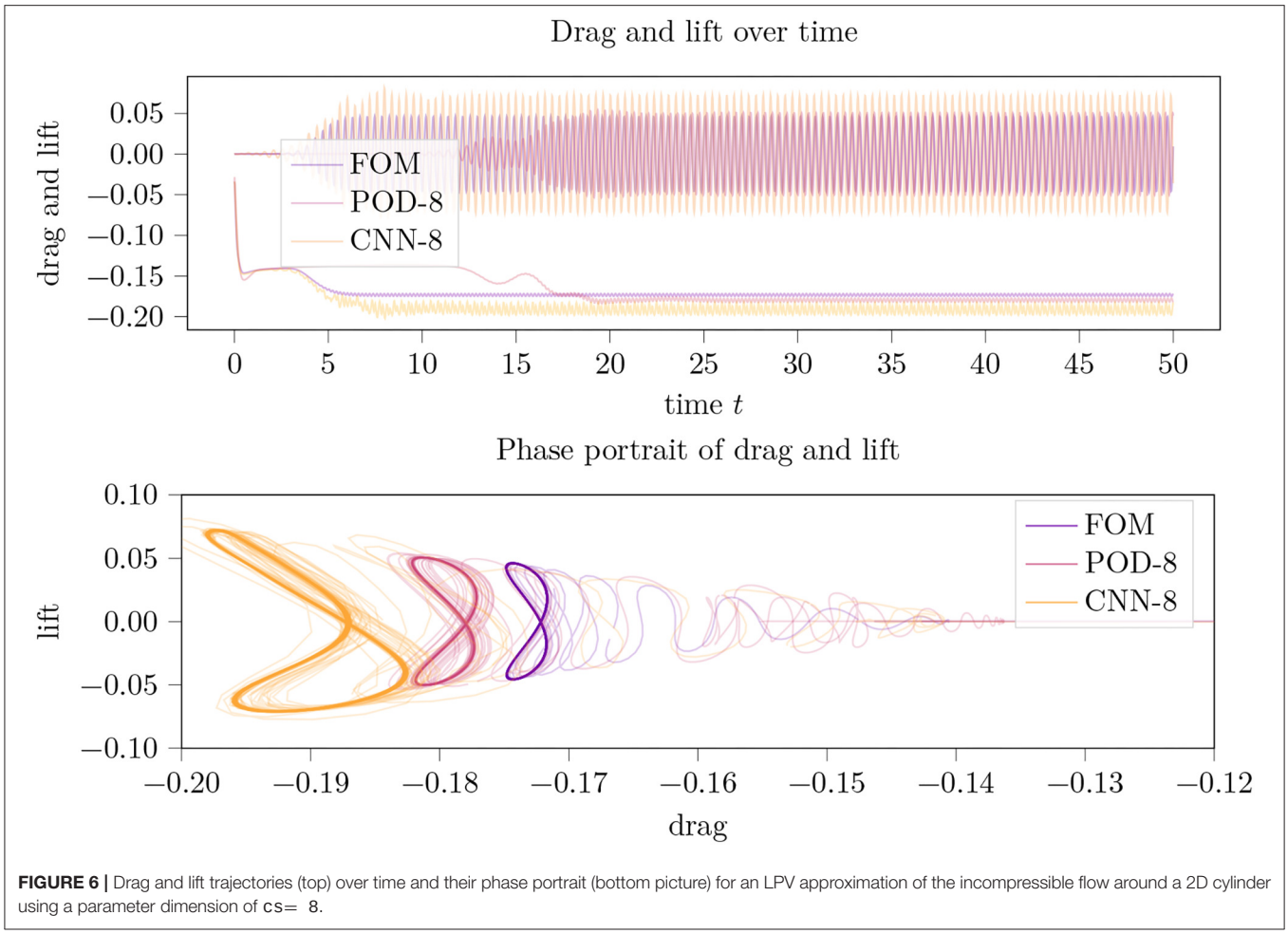


FIGURE 6 | Drag and lift trajectories (top) over time and their phase portrait (bottom picture) for an LPV approximation of the incompressible flow around a 2D cylinder using a parameter dimension of cs= 8.

where Ω is the computational domain; see Babuăka and Miller [32]. Since the flow passes the channel in ξ_1 direction, the first component of $\mathcal{F}(t)$ represents the current drag and the second component represents the lift force.

Once the semi-discrete model is defined, the overall procedure of setting up and evaluating a low-dimensional LPV surrogate model can be summarized in four major steps:

1. **Data Aquisition and Preparation.** This step generates the data used for computing the POD basis and for the CAE model training. For that, a simulation on the base of the original model is performed and solution snapshots at dedicated time instances are stored. In view of being used for the training of the CAE model, among others, the data is interpolated to a tensor grid.
2. **Training of the Encoder and Decoder.** By means of the snapshot data, a POD basis is computed. Also, the interpolated data are used to optimize the parameters of the CAE encoder and decoder.
3. **Setup of the LPV Approximation.** The CAE and the POD encoder and decoders that approximate a current velocity $\upsilon(t)$ by $\tilde{\upsilon}(t) = \tilde{W}\rho(t)$ for some basis $\tilde{W}$ and the code $\rho(t)$ is used to approximate the actual nonlinear $N(\upsilon(t))\,\upsilon(t)$ term in the model by a low-dimensional LPV approximation $\tilde{N}(\rho(t))\,\upsilon(t)$.
4. **Simulation with the LPV Model.** Finally, simulations of the original model with the nonlinearity replaced by the LPV approximation are performed and evaluated.

All these steps are explained in detail in the following subchapters.

5.1. Data Acquisition and Preparation

The data $[V]$ for the training of the CNN and the computation of POD bases $\tilde{V}$ and $\tilde{W}$ of the states υ_i and of the convection field $N(\upsilon_i)\upsilon_i$ is collected from the simulation on the time interval $[0, t_e]$ at ndp equally spaced data points.

For the use for training of the CNN, the states data υ_i is interpolated to the tensor grid by means of the interpolation operator $\mathbf{P}\colon \mathbb{R}^{n_v} \to \mathbb{R}^{2\times n_x \times n_y}$ to give the data set

$$[\mathbf{V}] = [\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_{\mathsf{ndp}}] = [\mathbf{P}\upsilon_1, \mathbf{P}\upsilon_2, \ldots \mathbf{P}\upsilon_{\mathsf{ndp}}] \tag{9}$$

Additionally, we recorded the maximal and minimal values of the data in $[\mathbf{V}]$ and linearly scaled all interpolations to the range of

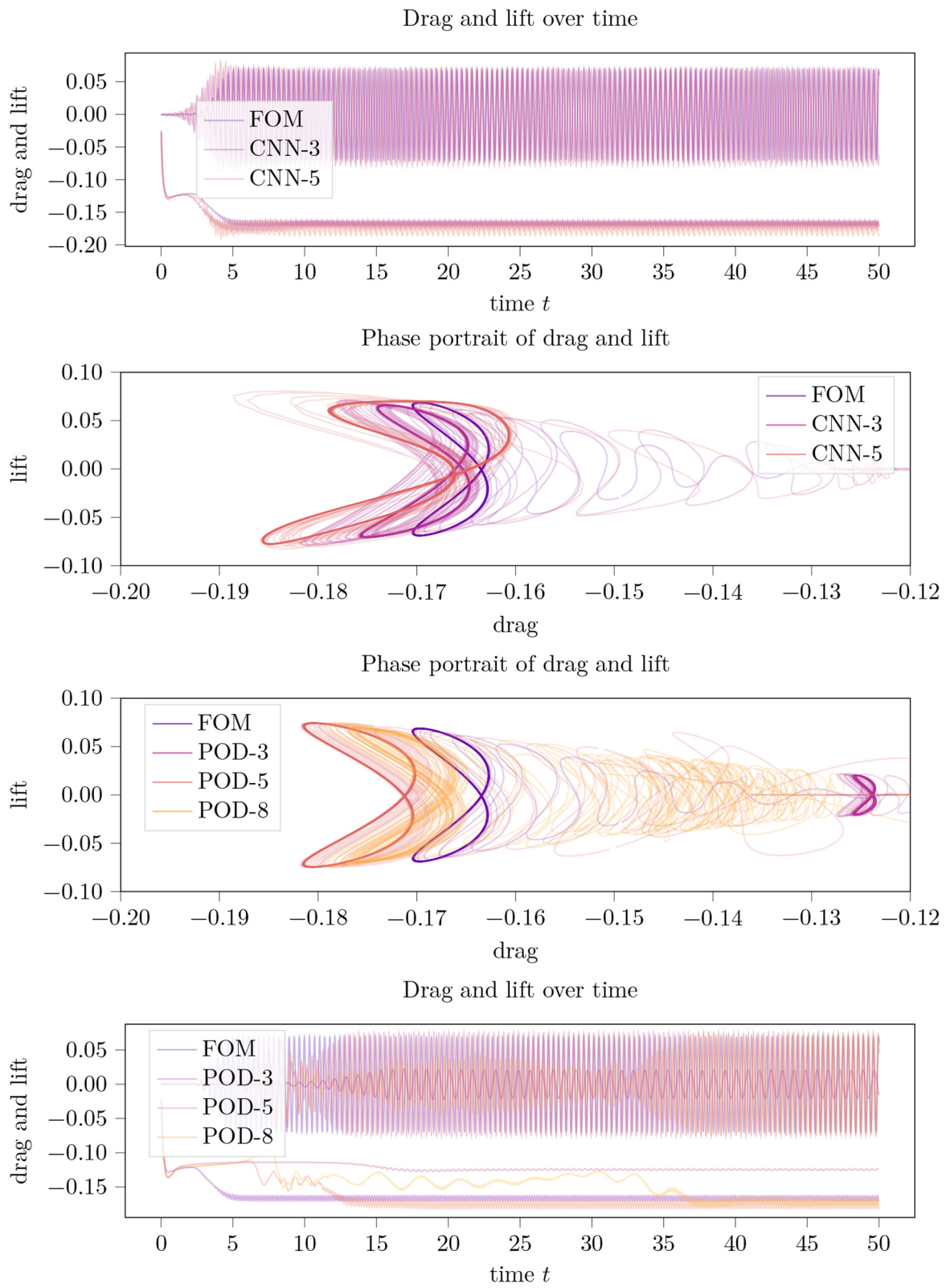


FIGURE 7 | Summarized results for the case of Re = 60. Drag and lift trajectories over time and their phase portraits for an LPV approximation of the incompressible flow around a 2D cylinder using a variable parameter dimension `cs` in the POD and CAE approximation and in comparison to the full order model (FOM).

$[-1, 1]$. The scaled and partially doubled data, we will denote by $[\mathbf{V}+]$.

Note that the interpolation to the tensor grid is not lossless in particular in the considered case where the dimension of the tensorized data is `2 x 63 x 127 = 16,002` is much smaller than the data on the full FEM grid, where the tensor grid is not adapted to the problem whatsoever, and where geometrical features of the domain are not represented in the tensorized data; cp. **Figure 2**.

At first, as preliminary tests showed, the trained neural network performed poorly on the first data points, e.g., the initial phase of the simulation. To shift the focus of the training toward the initial phase, we doubled a defined amount of the data points as follows: for a given percentage `p`, a subgrid of the snapshot time instances was computed that contained `p` percent of the data points exponentially distributed of the time range of the data. Hereby, the time differences between the data points were smallest at the beginning and grew exponentially toward the end of the time range. The selected data was appended to the data set. In this way, no additional information was added but due to the unequal distribution of the doubled data points, the iterative training of the CNN will focus more on the initial phase. As done in the presented numerical study, this doubling of data can be repeated with varying percentages.

5.2. Convolutional Neural Network Setup and Training

We define the CNN for the encoding via a number `Ncl` of convolutional layers followed by a fully connected linear layer with activation that maps the output of the repeated convolutions onto a vector of size `cs`—the *code size*. This encoding part will be denoted by CAE and we will write

$$\rho(t) = \mathsf{CAE}(\mathbf{P}\upsilon(t))$$

to express that a velocity state $\upsilon(t)$ has been encoded to $\rho(t)$ of dimension `cs` via the neural network. Note the inclusion of the interpolation $\mathbf{P}$ to the tensor grid.

For the decoding, we use a truly linear layer of input dimension `cs` and output dimension `k`, so that with a POD basis $\tilde{V}$ of dimension `k` the reconstruction reads

$$\tilde{\upsilon}(t) = \tilde{V}W\rho(t) + \upsilon_s, \tag{10}$$

where W is the `k` $\times$ `cs`-matrix that realizes the linear layer that maps $\rho \mapsto \tilde{\rho}$ and where vs. is the shift vector that was used to center the POD data; cp. Remark 4.

The parameters of the architecture of the CAE-model used in the presented numerical results are given in **Table 1**. See also **Figure 3** for an illustration.

The parameters of CAE as well as the coefficients of W are then trained to fit the data of $[\mathbf{V}+]$ with respect to the loss function

$$\left\|N(\upsilon_i)\upsilon_i - N\big(\tilde{V}W\mathsf{CAE}(\mathbf{P}\upsilon_i)\big)\upsilon_i)\right\|^2_{M^{-1}} + \left\|\upsilon_i - VW\mathsf{CAE}(\mathbf{P}\upsilon_i)\right\|^2_M. \tag{11}$$

For that, the data $[\mathbf{V}+]$ is randomly split into *batches* of size `Bs`, the mean value of Equation (11) over a batch is computed, and the parameters of CAE and W updated according to a stochastic gradient method. This procedure is repeated over the same data in a number of *epochs*.

5.3. Numerical Realization of the LPV Approximation

With the decoder matrix W, the POD basis $\tilde{V}$, and the CAE model for variable code sizes `cs` at hand, we approximate the actual nonlinearity $N(\upsilon)\upsilon$ by the linear-affine LPV approximation

$$N(\upsilon)\upsilon \approx \frac{1}{2}\big[N(\tilde{V}W\rho)\upsilon + N(\upsilon)\tilde{V}W\rho\big] \tag{12}$$

of dimension `cs` where $\rho = \mathsf{CAE}(\upsilon)$.

For comparison, we considered the plain POD LPV approximation

$$N(\upsilon)\upsilon \approx \frac{1}{2}\big[N(\tilde{V}_{\mathtt{cs}}\rho)\upsilon + N(\upsilon)\tilde{V}_{\mathtt{cs}}\rho\big] \tag{13}$$

with the POD basis $\tilde{V}_{\mathtt{cs}}$ of dimension `cs` and the POD coordinates $\rho = \tilde{V}_{\mathtt{cs}}^{\mathsf{T}}\nu$.

Remark 5. The blending

$$sN(\cdot)\tilde{\nu}(\rho) + (1-s)N(\tilde{\nu}(\rho))(\cdot)$$

of the two natural LPV representations was found to be beneficial since $N(\cdot)\rho$ tended to damp out the fluctuations whereas $N(\rho)(\cdot)$ triggered the unsteady behavior very well but led to blowups regardless whether the CAE or the POD approximation was considered. This observation can be explained by known stabilizing effect of linearizations of the first type (cp., e.g., the convergence analysis of iterative linearization schemes in Karakashian [33]) whereas in a linearization like $N(\cdot)\rho$ the ρ undergoes a differentiation which can explain the tendency for a blow-up.

Certainly, the value of s can be another parameter in the optimization of the approximation. For simplicity, we simply fixed it to $s = \frac{1}{2}$.

5.4. Numerical Simulation

For a variable code size `cs` that eventually defines the dimension of the affine-linear LPV approximation, we checked the performance of the CAE-`cs`-model and compared it to a POD approximation of the same dimension.

Since the cylinder wake is a chaotic system, in the sense that, e.g., the transition to the periodic regime is severely influenced by perturbations, a direct comparison of trajectories is uninformative. Therefore, we plotted the resulting curves of drag and lift for the full order simulation FOM and the approximations on top of each other to get a qualitative expression of the approximation. An informative comparison, however, can be derived from the analysis how well the

TABLE 2 | Table of parameters for the CAE-model optimization.

Description	Value in simulation
Number of data points	2000
Percentage of duplicated datapoints[a]	15&10
Optimization algorithm	torch.optim.Adam
Learning rate[b]	0.0075
Size of *batches* for the training[c]	25
Number of *epochs*[d]	25

[a] *We duplicated some data points to have a better focus on the initial phase of the simulation. Here,* 15&10 *means that we added 15% percent of the data and another 10% of the initial data on top of it; cp. Section 5.1.*
[b] *The optimization algorithm has many other parameters that can be altered. We used the default values except for the learning rate.*
[c] *How many data points are evaluated until the optimization algorithm updates the parameters.*
[d] *How often the optimization iterates over the full data set.*

approximations capture the limit cycle of the periodic regime. For that, we plot the phase portrait of drag vs. lift.

For the smallest investigated code size cs=3 (cp. **Figure 4**) we found that the CAE-3 approximation departed from the FOM-simulation in the initial phase but captured the limit cycle well with but a small distortion of the symmetry and an overestimation of the drag by about 2%. The POD-3-simulation, i.e., the LPV approximation by a POD basis of dimension 3, did not reach a clear limit cycle within the comparatively long time horizon and did not well reproduce the nominal values of drag and lift either.

For an increased code size cs=5, the POD approximation qualitatively and quantitatively (cp. **Figure 5**) improved to approximately the same level as CAE-3. The CAE approximation for this code size did not perform well at all.

For cs=8, the POD approximation improved only marginally whereas the CAE-model approximation reached a limit cycle again though with a huge distortion of the symmetry and a significant overestimation of drag and lift (cp. **Figure 6**).

Since, theoretically, the CAE-8 and CAE-5 model contain the CAE-3 model, their failure in the approximation basically means a failure of the optimization of the model parameters during the training. The manifold ways of adapting the parameters of the network architecture as well as those of the optimization procedure offers many ways of improving.

Even more, the interpolation to the uniform tensor grid (cp. **Figure 2**) means a significant loss of information so that slight improvements here, e.g., through a local refinement that preserves the tensor structure, will likely improve the approach.

Nonetheless, the good performance of CAE-3 fully supports our initial working hypotheses that a convolutional neural network can provide a very low-dimensional encoding targeted to an efficient affine LPV approximation of the incompressible Navier-Stokes equations. In this case, it took a parameter space of dimension cs=3 to well approximate the nonlinear incompressible Navier-Stokes equations of dimension 42764 just in the velocity part.

Finally, for a rough orientation about the computational costs, we provide the computational times as they can be read off the log files (i.e., only a single *wall clock measurement*). All experiments have been conducted on a computing cluster but without GPU support and restricted to two computing kernels. The training of the individual CAE models took about 30 min, the simulation of the full order model over the full time frame from $t = 0$ to $t = 50$ took about 80 min. The same simulation but with the POD reduced LPV model took about 130 min, whereas the CAE model took 420 min. The computational overhead of the POD model is mainly due to the blending that requires the evalution of the nonlinearity twice as often. This also holds true for the CAE model but accounts only for a part of the overtime.

Certainly, these timings can be improved significantly. However, the focus of the presented work was on reducing the model in terms of its structure by replacing the nonlinear term by a low-dimensional LPV formulation.

To evaluate the robustness of the presented method, we conduct the same experiments at Re = 60, i.e., in a regime that is even more convection-dominated and that is expected to be more difficult to approximat by a linear projection method. The results are displayed in **Figure 7** and are well inline with those reported for Re = 40 before. The CAE-3-model outperformed all POD-configurations, once a satisfactory setup of hyperparameters was found. In fact, for the training of the CAE-model for Re = 60 we added another batch p=20 percent of data focussed on the initial phase (cp. **Table 2**) while for the simulation we set $s = \frac{1}{3}$ (cp. Remark 5). Both updates to the Re = 40 case are natural as in this regime, the initial phase is shorter and the simulation is less stable. Interestingly, the POD-models gave a solid performance at small cs, but deteriorated for larger sizes of the POD basis. An explanation for this behavior might lie in the sensitivity of the problem and in numerical errors in the POD vector computation. We also note that adding stability to the system by an even smaller s was of no help as it damped the periodic behavior (with a result similar to the POD-3 approximation in **Figure 7**).

6. CONCLUSION AND OUTLOOK

In the presented work, we have provided a proof of concept on how CNNs in combination with POD can be used to generate very low-dimensional LPV approximations to nonlinear systems. For the considered Navier-Stokes equations and, generally, for any quadratic system, the LPV approximation is affine-linear if only the decoding from the coded variable ρ to the state reconstruction $\tilde{\nu}$ is a linear map.

The myriad of parameters that can be tuned in the design of DNNs and their training have not been investigated in depth (once a satisfying working setup has been found). Accordingly, there is a huge potential for improvements since the well working CAE-3 example is certainly no global optimum and the larger code sizes could, theoretically, be tuned for better approximations. A systematic investigation of the parameters is left to future research efforts.

Another future research direction is the direct identification of the parametrization matrices N_i, $i = 0, \ldots, r$, for an affine-linear

LPV-representation

$$N(\upsilon) \approx \tilde{N}(\rho(\upsilon)) = \tilde{N}_0 + \sum_{i=1}^{r} \rho_i(\upsilon)\tilde{N}_i$$

without resorting to POD coordinates. It was mentioned in Koelewijn and Tóth [12], that a neural network without the nonlinear activation that approximates $\rho(\upsilon) \rightarrow N(\rho(\upsilon))$ is such an affine-linear map with coefficients $\tilde{N}_i$ defined through the weighting matrices of the neural net. Accordingly, the same architectures and optimization algorithms can be used to design the parametrization from scratch. However, in the large-scale setting, it is not feasible to learn (and even just to store) these coefficients. A general approach to that would be sparsity enforcing methods in the learning of the weights. A more specific approach could consider transposed convolutional layers that reverses the convolutions and contractions but without the nonlinear activations. Certainly, the concatenated operations of the transposed convolutions and the reversal of the interpolation from the FEM to the tensor grid can be represented as one sparse operator. This is subject to further investigations.

DATA AVAILABILITY STATEMENT

The code used to compute the numerical results is available from https://doi.org/10.5281/zenodo.6401953.

AUTHOR CONTRIBUTIONS

JH conducted the basic research and the numerical experiments and wrote the manuscript. PB contributed relevant references and proofread and enhanced the manuscript. RB contributed to the numerical experiments and generated plots. All authors contributed to the article and approved the submitted version.

REFERENCES

1. Kokotovic PV. The joy of feedback: nonlinear and adaptive. *IEEE Control Syst Mag*. (1992) 12:7–17. doi: 10.1109/37.165507
2. Sontag ED. *Mathematical Control Theory. 2nd Edn. Texts in Applied Mathematics*. New York, NY: Springer-Verlag (1998).
3. Dodds SJ. Sliding mode control and its relatives. In: *Feedback Control. Linear, Nonlinear and Robust Techniques and Design with Industrial Applications*. London: Springer London (2015). p. 705–92.
4. Grüne L, Pannek J. *Nonlinear model predictive control. Theory and algorithms*. Cham: Springer (2017).
5. Breiten T, Kunisch K, Pfeiffer L. Feedback stabilization of the two-dimensional Navier-Stokes equations by value function approximation. *Appl Math Optim*. (2019) 80:599–641. doi: 10.1007/s00245-019-09586-x
6. Banks HT, Lewis BM, Tran HT. Nonlinear feedback controllers and compensators: a state-dependent Riccati equation approach. *Comput Optim Appl*. (2007) 37:177–218. doi: 10.1007/s10589-007-9015-2
7. Benner P, Heiland J. Exponential stability and stabilization of extended linearizations via continuous updates of riccati-based feedback. *Internat J Robust Nonlinear Control*. (2018) 28:1218–32. doi: 10.1002/rnc.3949
8. Apkarian P, Noll D. Controller design via nonsmooth multidirectional search. *SIAM J Control Optim*. (2006) 44:1923–49. doi: 10.1137/S0363012904441684
9. Beeler SC, Tran HT, Banks HT. Feedback control methodologies for nonlinear systems. *J Optim Theory Appl*. (2000) 107:1–33. doi: 10.1023/A:1004607114958
10. Alla A, Kalise D, Simoncini V. State-dependent Riccati equation feedback stabilization for nonlinear PDEs. *arXiv 2106. 07163*. (2021) doi: 10.48550/arXiv.2106.07163
11. Apkarian P, Gahinet P, Becker G. Self-scheduled H_∞ control of linear parameter-varying systems: a design example. *Autom*. (1995) 31:1251–61. doi: 10.1016/0005-1098(95)00038-X
12. Koelewijn PJW, Tóth R. Scheduling dimension reduction of LPV models-a deep neural network approach. In: *2020 American Control Conference, ACC 2020. Denver, CO, USA, July 1-3, 2020*. Denver, CO: IEEE (2020). p. 1111–7.
13. Hashemi SM, Werner H. LPV modelling and control of burgers' equation. *IFAC Proc Volumes*. (2011) 44:5430–5. doi: 10.3182/20110828-6-IT-1002.03318
14. Dahmen W, Plesken C, Welper G. Double greedy algorithms: reduced basis methods for transport dominated problems. *ESAIM Math Model Numer Anal*. (2014) 48:623–63. doi: 10.1051/m2an/2013103
15. Ohlberger M, Rave S. Reduced basis methods: success, limitations and future challenges. In: *Proceedings of the Conference Algoritmy*. Vysoké Tatry, Slovakia (2016). p. 1–12.
16. Reiss J, Schulze P, Sesterhenn J, Mehrmann V. The shifted proper orthogonal decomposition: a mode decomposition for multiple transport phenomena. *SIAM J Sci Comput*. (2018) 40:A1322–44. doi: 10.1137/17M1140571
17. Papapicco D, Demo N, Girfoglio M, Stabile G, Rozza G. The neural network shifted-proper orthogonal decomposition: a machine learning approach for non-linear reduction of hyperbolic equations. *arXiv[Preprint].arXiv:2108.06558*. (2021). doi: 10.1016/j.cma.2022.114687
18. Sarna N, Benner P. Data-Driven model order reduction for problems with parameter-dependent jump-discontinuities. *arXiv[Preprint].arXiv:2105.00547*. (2021). doi: 10.1016/j.cma.2021.114168
19. Deo IK, Jaiman R. Learning wave propagation with attention-based convolutional recurrent autoencoder net. *arXiv[Preprint].arXiv:2201.06628*. (2022). doi: 10.48550/arXiv.2201.06628
20. Lee K, Carlberg KT. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *J Comput Phys*. (2020) 404:108973. doi: 10.1016/j.jcp.2019.108973
21. Kim Y, Choi Y, Widemann D, Zohdi T. Efficient nonlinear manifold reduced order model. *arXiv[Preprint].arXiv:2011.07727*. (2020). doi: 10.48550/arXiv.2011.07727
22. Mojgani R, Balajewicz M. Physics-aware registration based auto-encoder for convection dominated PDEs. *arXiv[Preprint].arXiv.2006.15655*. (2020). doi: 10.48550/arXiv.2006.15655
23. Nikolopoulos S, Kalogeris I, Papadopoulos V. Non-intrusive surrogate modeling for parametrized time-dependent PDEs using convolutional autoencoders. *arXiv[Preprint].arXiv.2101.05555*. (2021). doi: 10.1016/j.engappai.2021.104652
24. Fresca S, Manzoni A. POD-DL-ROM: enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition. *arXiv[Preprint].arXiv:2103.0160*. (2021). 2103. 0160. doi: 10.1016/j.cma.2021.114181
25. O'Shea K, Nash R. An introduction to convolutional neural networks. *arXiv[Preprint].arXiv.1511.08458*. (2015). doi: 10.48550/arXiv.1511.08458
26. Baumann M, Benner P, Heiland J. Space-Time Galerkin POD with application in optimal control of semi-linear parabolic partial differential equations. *SIAM J Sci Comput*. (2018) 40:A1611–41. doi: 10.1137/17M1135281
27. Behr M, Benner P, Heiland J. Example setups of navier-stokes equations with control and observation: spatial discretization and representation via linear-quadratic matrix coefficients. *arXiv[Preprint].arXiv:1707.08711*. (2017). doi: 10.48550/arXiv.1707.08711
28. Heiland J. *Decoupling and Optimization of Differential-Algebraic Equations with Application in Flow Control* (Dissertation). TU Berlin (2014).

29. Altmann R, Heiland J. Continuous, semi-discrete, and fully discretized navier-stokes equations. In: Campbell S. Ilchmann A, Mehrmann V, Reis T, editors. *Applications of Differential-Algebraic Equations: Examples and Benchmarks. Differential-Algebraic Equations Forum*. Springer (2019). p. 277–312.
30. Logg A, Mardal KA, Wells G, editors. *Automated Solution of Differential Equations by the Finite Element Method. The Fenics Book. Vol. 84*. Berlin: Springer (2012).
31. Heiland J. *dolfin_navier_scipy: a python Scipy FEniCS interface*. Zenodo. (2019). Available online at: ttps://github.com/highlando/dolphin_navier_scipy. Github/Zenodo.
32. Babuăka I, Miller A. The post-processing approach in the finite element method-part 1: calculation of displacements, stresses and other higher derivatives of the displacements. *Int J Numer Meth Eng*. (1984) 20:1085–109. doi: 10.1002/nme.1620200610
33. Karakashian OA. On a Galerkin-Lagrange multiplier method for the stationary Navier-Stokes equations. *SIAM J Numer Anal*. (1982) 19:909–23. doi: 10.1137/0719066

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers | Frontiers in Physics

ORIGINAL RESEARCH
published: 10 May 2022
doi: 10.3389/fphy.2022.869681

Low-Rank Approximations for Parametric Non-Symmetric Elliptic Problems

Tomás Chacón Rebollo[1]*, *Macarena Gómez Mármol*[2] *and Isabel Sánchez Muñoz*[3]

[1]Departamento EDAN & IMUS, Universidad de Sevilla, Sevilla, Spain, [2]Departamento EDAN, Universidad de Sevilla, Sevilla, Spain, [3]Departamento Matemática Aplicada I, Universidad de Sevilla, Sevilla, Spain

In this study, we obtained low-rank approximations for the solution of parametric non-symmetric elliptic partial differential equations. We proved the existence of optimal approximation subspaces that minimize the error between the solution and an approximation on this subspace, with respect to the mean parametric quadratic norm associated with any preset norm in the space of solutions. Using a low-rank tensorized decomposition, we built an expansion of approximating solutions with summands on finite-dimensional optimal subspaces and proved the strong convergence of the truncated expansion. For rank-one approximations, similar to the PGD expansion, we proved the linear convergence of the power iteration method to compute the modes of the series for data small enough. We presented some numerical results in good agreement with this theoretical analysis.

Keywords: low-rank tensor approximations, non-symmetric problems, PGD mode computation, alternate least squares, power iteration algorithm

OPEN ACCESS

Edited by:
Traian Iliescu,
Virginia Tech, United States

Reviewed by:
Yusif Gasimov,
Azerbaijan University, Azerbaijan
Francesco Ballarin,
Catholic University of the Sacred Heart, Brescia, Italy

***Correspondence:**
Tomás Chacón Rebollo
chacon@us.es

Specialty section:
This article was submitted to Statistical and Computational Physics, a section of the journal Frontiers in Physics

Received: *04 February 2022*
Accepted: *28 March 2022*
Published: *10 May 2022*

Citation:
Chacón Rebollo T, Gómez Mármol M and Sánchez Muñoz I (2022) Low-Rank Approximations for Parametric Non-Symmetric Elliptic Problems. Front. Phys. 10:869681. doi: 10.3389/fphy.2022.869681

1 INTRODUCTION

This study deals with the low-rank tensorized decomposition (LRTD) of parametric non-symmetric linear elliptic problems. The basic objective in model reduction is to approximate large-scale parametric systems by low-dimension systems, which are able to accurately reproduce the behavior of the original systems. This allows tackling in affordable computing times, control, optimization, and uncertainty quantification problems related to the systems modeled, among other problems requiring multiple computations of the system response.

The PGD method, introduced by Ladévèze in the framework of the LATIN method (LArge Time INcrement method [22]) and extended to multidimensional problems by Ammar et al [2], has experienced an impressive development with extensive applications in engineering problems. This method is an *a priori* model reduction technique that provides a separate approximation of the solution of parametric PDEs. A compilation of advances in the PGD method may be found in [11].

Among the literature studying the convergence and numerical properties of the PGD, we can highlight [1], where the convergence of the PGD for linear systems of finite dimension is proved. In [8], the convergence of the PGD algorithm applied to the Laplace problem is proven, in a tensorized framework. The study [9] proves the convergence of the PGD for an optimization problem, where the functional framework is strongly convex and has a Lipschitz gradient in bounded sets. In [17], the authors prove the convergence of an algorithm similar to a PGD for the resolution of an optimization problem for a convex functional defined on a reflective Banach space. In [16], the authors prove the convergence of the PGD for multidimensional elliptic PDEs. The convergence is achieved because of the generalization of Eckart and Young's theorem.

The present study is motivated by [5], where the authors present and analyze a generalization of the previous study [16] when operator A depends on a parameter. The least-squares LRTD is introduced to solve parametric symmetric elliptic equations. The modes of the expansion are characterized in terms of optimal subspaces of a finite dimension that minimize the residual in the mean quadratic norm generated by the parametric elliptic operator. As a by-product, this study proves the strong convergence in the natural mean quadratic norm of the PGD expansion.

A review of low-rank approximation methods (including PGD) may be found in the studies [18, 19]. In particular, minimal residual formulations with a freely chosen norm for Petrov–Galerkin approximations are presented. In addition, the study [10] gives an overview of numerical methods based on the greedy iterative approach for non-symmetric linear problems.

In [3, 4], a numerical analysis of the computation of modes for the PGD to parametric symmetric elliptic problems is reported. The nonlinear coupled system satisfied by the PGD modes is solved by the power iteration (PI) algorithm, with normalization. This method is proved to be linearly convergent, and several numerical tests in good agreement with the theoretical expectations are presented.

Actually, for symmetric problems, the PI algorithm to solve the PGD modes turns out to be the adaptation of the alternating least-squares (ALS) method thoroughly used to compute tensorized low-order approximations of high-order tensors. The ALS method was used in the late 20th century within the principal components analysis (see [6, 7, 20, 21]) and extended in [27] to the LRTD approximation of high-order tensors. Its convergence properties were subsequently analyzed by several authors; local and global convergence proofs within several approximation frameworks are given in [14, 25, 26]. Several generalizations were reported; as mentioned, without intending to be exhaustive, in the studies [12–15, 23].

The convergence proofs within the studies [14, 25, 26] cannot be applied to our context as we are dealing with least-squares with respect to the parametric norm intrinsic to the elliptic operator, even for symmetric problems. Comparatively, the difference is similar to that between proving the convergence of POD or PGD approximations to elliptic PDEs. The POD spaces are optimal with respect to a user-given fixed norm, while the PGD spaces are optimal with respect to the intrinsic norm associated to the elliptic operator (see [5]). This use of the intrinsic parametric norm does not make it necessary the previous knowledge of the tensor object to be approximated (in our study, the parametric solution of the targeted PDE), as is needed in standard ALS algorithms.

In the present study, we propose an LRTD to approximate the solution of parametric non-symmetric elliptic problems based upon symmetrization of the problem (**Section 2**). Each mode of the series is characterized in terms of optimal subspaces of finite dimension that minimize the error between the parametric solution and its approximation on this subspace, but now with respect to a preset mean quadratic norm as the mean quadratic norm associated to the operator in the non-symmetric case is not well-defined (**Section 3**). We prove that the truncated LRTD expansion strongly converges to the parametric solution in the mean quadratic norm (**Section 4**).

The minimization problems to compute the rank-one optimal modes are solved by a PI algorithm (**Section 5**). We prove that this method is locally linearly convergent and identifies an optimal symmetrization that provides the best convergence rates of the PI algorithm, with respect to any preset mean quadratic norm (**Section 6**).

We finally report some numerical tests for 1D convection–diffusion problems that confirm the theoretical results on the convergence of the LRTD expansion and the convergence of the PI algorithm. Moreover, the computing times required by the optimal symmetrization are compared advantageously to those required by the PGD expansion (**Section 7**).

2 PARAMETRIC NON-SYMMETRIC ELLIPTIC PROBLEMS

Let us consider the mathematical formulation for parametric elliptic problems introduced in [5] that we shall extend to non-symmetric problems.

Let $(\Gamma, \mathcal{B}, \mu)$ be a measure space, where we assume that the measure μ is σ-finite. Let H be a separable Hilbert space endowed with the scalar product $(\cdot, \cdot)$ and associated norm $\|\cdot\|$, denote by H' the dual space of H and by $\langle \cdot, \cdot \rangle$ the duality pairing between H' and H. We will consider the Lebesgue space $L^2_\mu(\Gamma)$ and the Bochner space $L^2_\mu(\Gamma; H)$ and its dual space $L^2_\mu(\Gamma; H')$, denoting $\langle \cdot, \cdot \rangle$ as the duality between $L^2_\mu(\Gamma; H)$ and $L^2_\mu(\Gamma; H')$. We are interested in solving the parametric family of variational problems:

$$\begin{cases} \text{Find } u\colon \Gamma \to H \text{ such that} \\ a(u(\gamma), v; \gamma) = \langle f(\gamma), v \rangle, \quad \forall\, v \in H,\ \mu - \text{a.e. } \gamma \in \Gamma, \end{cases} \tag{1}$$

where $a(\cdot, \cdot; \gamma)\colon H \times H \to \mathbb{R}$ is a parameter-dependent, possibly non-symmetric, bilinear form and $f(\gamma) \in H'$ is a parameter-dependent continuous linear form.

It is assumed that $a(\cdot, \cdot; \gamma)$ is uniformly continuous and uniformly coercive on H μ-a. e. $\gamma \in \Gamma$ and there exist positive constants α and β independent of γ such that,

$$a(w, v; \gamma) \le \beta \|w\| \|v\|, \quad \forall\, w, v \in H,\ \mu - \text{a.e. } \gamma \in \Gamma, \tag{2}$$

$$\alpha \|w\|^2 \le a(w, w; \gamma), \quad \forall\, w \in H,\ \mu - \text{a.e. } \gamma \in \Gamma. \tag{3}$$

By the Lax–Milgram theorem, problem **(1)** admits a unique solution μ-a. e. $\gamma \in \Gamma$. To treat the measurability of u with respect to γ, let us consider the problem:

Let $\mathbf{f}$ be a function that belongs to $L^2_\mu(\Gamma; H')$ such that $\mathbf{f}(\gamma) = f(\gamma)$ μ-a. e. $\gamma \in \Gamma$,

$$\begin{cases} \text{Find } \mathbf{u} \in L^2_\mu(\Gamma; H) \text{ such that} \\ \bar{a}(\mathbf{u}, \mathbf{v}) = \langle \mathbf{f}, \mathbf{v} \rangle, \quad \forall\, \mathbf{v} \in L^2_\mu(\Gamma; H), \end{cases} \tag{4}$$

where $\bar{a}\colon L^2_\mu(\Gamma; H) \times L^2_\mu(\Gamma; H) \to \mathbb{R}$ is defined by

$$\bar{a}(\mathbf{w},\mathbf{v}) = \int_\Gamma a(\mathbf{w}(\gamma),\mathbf{v}(\gamma);\gamma)\,d\mu(\gamma). \tag{5}$$

By the Lax–Milgram theorem, owing to **(2)** and **(3)**, problem **(4)** admits a unique solution. Problems **(1)** and **(4)** are equivalent in the sense that (see [5])

$$\mathbf{u}(\gamma) = u(\gamma), \quad \mu - \text{a.e. } \gamma \in \Gamma.$$

We shall consider a symmetrized reformulation of the formulation **(4)**. Let us consider a family of inner products in H, $\{(\cdot,\cdot)_\gamma\}_{\gamma\in\Gamma}$ which generate the norms $\|\cdot\|_\gamma$ uniformly equivalent to the $\|\cdot\|$ norm and there exist $\alpha_H > 0$ and $\beta_H > 0$ such that,

$$\alpha_H \|v\| \le \|v\|_\gamma \le \beta_H \|v\| \quad \text{for all } \gamma \in \Gamma, \tag{6}$$

considering the associated scalar product in $L^2_\mu(\Gamma;H)$ and $\int_\Gamma(\cdot,\cdot)_\gamma\,d\mu(\gamma)$. As $\bar{a}$ is continuous and coercive on $L^2_\mu(\Gamma;H)$, there exists a unique isomorphism in $L^2_\mu(\Gamma;H)$ that we denote $\mathcal{A}$, such that

$$\int_\Gamma \big((\mathcal{A}\mathbf{w})(\gamma),\mathbf{v}(\gamma)\big)_\gamma\,d\mu(\gamma) = \bar{a}(\mathbf{w},\mathbf{v}), \quad \forall\,\mathbf{w},\,\mathbf{v} \in L^2_\mu(\Gamma;H). \tag{7}$$

Let us define a new bilinear form $\bar{b}$: $L^2_\mu(\Gamma;H)\times L^2_\mu(\Gamma;H)\to\mathbb{R}$ by

$$\bar{b}(\mathbf{w},\mathbf{v}) \coloneqq \bar{a}(\mathbf{w},\mathcal{A}\mathbf{v}). \tag{8}$$

It is to be noted that $\bar{b}$ is symmetric, as it can be written as

$$\bar{b}(\mathbf{w},\mathbf{v}) = \int_\Gamma \big((\mathcal{A}\mathbf{w})(\gamma),(\mathcal{A}\mathbf{v})(\gamma)\big)_\gamma\,d\mu(\gamma). \tag{9}$$

Thus, as a consequence of **(2)** and **(3)**, the form $\bar{b}$ defines a scalar product in $L^2_\mu(\Gamma;H)$ and generates a norm equivalent to the usual norm is this space.

Now, problem **(4)** is equivalent to

$$\begin{cases} \text{Find } \mathbf{u} \in L^2_\mu(\Gamma;H) \text{ such that} \\ \bar{b}(\mathbf{u},\mathbf{v}) = \langle \mathbf{f},\mathcal{A}\mathbf{v}\rangle, \quad \forall\,\mathbf{v} \in L^2_\mu(\Gamma;H). \end{cases} \tag{10}$$

We shall use this formulation to build optimal approximations of problem **(1)** on subspaces of finite-dimension, when the form $\bar{a}(\cdot,\cdot)$ is not symmetric. For a given integer $k \ge 1$, we denote by S_k the Grassmannian variety of H formed by its subspaces of dimension smaller than or equal to k and consider the problem:

$$\min_{Z\in S_k} \bar{b}(\mathbf{u}-\mathbf{u}_Z,\mathbf{u}-\mathbf{u}_Z), \tag{11}$$

where $\mathbf{u}$ is the solution of problem **(4)** and $\mathbf{u}_Z$ is its approximation given by the Galerkin approximation of problem **(10)** in $L^2_\mu(\Gamma;Z)$:

$$\begin{cases} \text{Find } \mathbf{u}_Z \in L^2_\mu(\Gamma;Z) \text{ such that} \\ \bar{b}(\mathbf{u}_Z,\mathbf{v}) = \langle \mathbf{f},\mathcal{A}\mathbf{v}\rangle, \quad \forall\,\mathbf{v} \in L^2_\mu(\Gamma;Z). \end{cases} \tag{12}$$

Then, for any $k \in \mathbb{N}$, an optimal subspace of dimension smaller than or equal to k is the best subspace of the family S_k that minimizes the error between the solution $\mathbf{u}$ and its Galerkin approximation $\mathbf{u}_Z$ on this subspace, with respect to the mean quadratic norm generated by $\bar{b}$. Theorem 4.1 of [5] proves the following result:

Theorem 2.1 For any $k \ge 1$, problem **(11)** admits at least one solution.

3 TARGETED-NORM OPTIMAL SUBSPACES

It is assumed that we give a family of inner products $\{(\cdot,\cdot)_{H,\gamma}\}_{\gamma\in\Gamma}$ on H, which generate norms $\|\cdot\|_{H,\gamma}$ uniformly equivalent to the reference norm in H. Eventually, we may set $(\cdot,\cdot)_{H,\gamma} = (\cdot,\cdot)$. Our purpose is to determine the inner products $(\cdot,\cdot)_\gamma$ (introduced in **Section 2** and which we will call $(\cdot,\cdot)_{\gamma,\star}$) in such a way that the corresponding bilinear form $\bar{b}$ defined by **(8)** actually is

$$\bar{b}(\mathbf{w},\mathbf{v}) = \int_\Gamma (\mathbf{w}(\gamma),\mathbf{v}(\gamma))_{H,\gamma}\,d\mu(\gamma).$$

In this way, the optimal subspaces are the solution of the problem

$$\min_{Z\in S_k} \int_\Gamma \|\mathbf{u}(\gamma)-\mathbf{u}_Z(\gamma)\|^2_{H,\gamma}\,d\mu(\gamma). \tag{13}$$

Let us consider the operators $A_{\gamma,\star}$: $H\mapsto H$ and the bilinear forms $(\cdot,\cdot)_{\gamma,\star}$ on $H\times H$ defined by

$$a\big(w,A_{\gamma,\star}v;\gamma\big) = (w,v)_{H,\gamma}, \quad \forall\,w,\,v \in H, \tag{14}$$

$$(w,v)_{\gamma,\star} = \big(A^{-1}_{\gamma,\star}w, A^{-1}_{\gamma,\star}v\big)_{H,\gamma}, \quad \forall\,w,\,v \in H. \tag{15}$$

It is to be noted that $A_{\gamma,\star}$ is an isomorphism on H and consequently $(\cdot,\cdot)_{\gamma,\star}$ is an inner product on H. Due to **(2)** and **(3)**, the norms generated by these inner products are uniformly equivalent to the reference norm in H. Moreover, by **(14)** and **(15)**,

$$\big(A_{\gamma,\star}w,v\big)_{\gamma,\star} = \big(w,A^{-1}_{\gamma,\star}v\big)_{H,\gamma} = a(w,v;\gamma). \tag{16}$$

Let us consider now the inner product in $L^2_\mu(\Gamma;H)$ given by $\int_\Gamma(\cdot,\cdot)_{\gamma,\star}$ and the isomorphism $\mathcal{A}_\star$: $L^2_\mu(\Gamma;H)\mapsto L^2_\mu(\Gamma;H)$ defined by

$$\int_\Gamma \big((\mathcal{A}_\star\mathbf{w})(\gamma),\mathbf{v}(\gamma)\big)_{\gamma,\star}\,d\mu(\gamma) = \bar{a}(\mathbf{w},\mathbf{v}), \quad \forall\,\mathbf{w},\,\mathbf{v} \in L^2_\mu(\Gamma;H). \tag{17}$$

Then, it holds.

Lemma 1 Let $A_\gamma : H \to H$ be the continuous linear operators defined by

$$\big(A_\gamma w,v\big)_\gamma = a(w,v;\gamma), \quad \forall\,w,\,v \in H,\ \mu - \text{a.e. } \gamma \in \Gamma. \tag{18}$$

Then, it holds

$$(\mathcal{A}\mathbf{w})(\gamma) = A_\gamma\mathbf{w}(\gamma), \quad \forall\,\mathbf{w} \in H,\ \mu - \text{a.e. } \gamma \in \Gamma. \tag{19}$$

This result follows from a standard argument using the separability of space H that we omit for brevity. Then, by Lemma 1, we have

$$(\mathcal{A}_\star\mathbf{w})(\gamma) = A_{\gamma,\star}\mathbf{w}(\gamma), \quad \forall\,\mathbf{w} \in L^2_\mu(\Gamma;H),\ \mu - \text{a.e. } \gamma \in \Gamma. \tag{20}$$

Let us denote by $\bar{b}_\star$ the bilinear form on $L^2_\mu(\Gamma;H)$ given by

$$\bar{b}_\star(\mathbf{w},\mathbf{v}) = \bar{a}(\mathbf{w},\mathcal{A}_\star\mathbf{v}), \quad \forall\, \mathbf{w},\ \mathbf{v} \in L^2_\mu(\Gamma;H). \tag{21}$$

Then, by **(20)** and **(14)**,

$$\begin{aligned}\bar{b}_\star(\mathbf{w},\mathbf{v}) &= \int_\Gamma a\big(\mathbf{w}(\gamma), A_{\gamma,\star}\mathbf{v}(\gamma);\gamma\big)\, d\mu(\gamma)\\ &= \int_\Gamma (\mathbf{w}(\gamma),\mathbf{v}(\gamma))_{H,\gamma}\, d\mu(\gamma).\end{aligned} \tag{22}$$

As a consequence, the optimal subspaces obtained by the least-squares problem **(11)** when $\bar{b} = \bar{b}_\star$ satisfy **(13)**.

Remark 1 *When the forms $a(\cdot,\cdot;\gamma)$ are symmetric, if we choose*

$$(w,v)_{H,\gamma} = a(w,v;\gamma), \quad \forall\, w,\ v \in H,\ \mu-\text{a.e. } \gamma \in \Gamma, \tag{23}$$

then $A_{\gamma,\star}$ defined by **(14)** *is the identity operator in H. From* **(21)**, *it follows that $\bar{b}_\star = \bar{a}$. We, thus, recover the intrinsic norm in the symmetric case to determine the optimal subspaces.*

4 A DEFLATION ALGORITHM TO APPROXIMATE THE SOLUTION

Following the PGD procedure, we approximate the solution of problem **(1)** by a tensorized expansion with summands of rank $\leq k$, obtained by deflation. For all $N \geq 1$, we approximate

$$\mathbf{u} \simeq \mathbf{u}_N := \sum_{i=1}^{N} \mathbf{s}_i, \text{ with } \mathbf{s}_i \in L^2_\mu(\Gamma;H), \tag{24}$$

computed by the following algorithm:

Initialization: *let* be $\mathbf{u}_0 = 0$.

Iteration: *assuming* $\mathbf{u}_{i-1}$, known for any $i \geq 1$, set $\mathbf{e}_{i-1} := \mathbf{u} - \mathbf{u}_{i-1}$. Then,

$$\mathbf{u}_i = \mathbf{u}_{i-1} + \mathbf{s}_i, \text{ where } \mathbf{s}_i = (\mathbf{e}_{i-1})_W \tag{25}$$

with $(\mathbf{e}_{i-1})_W$ the approximation of $\mathbf{e}_{i-1}$ given by the problem **(12)** on an optimal approximation subspace W solution of the problem **(11)**,

$$W := \operatorname*{argmin}_{Z \in S_k} \bar{b}\big(\mathbf{e}_{i-1} - (\mathbf{e}_{i-1})_Z, \mathbf{e}_{i-1} - (\mathbf{e}_{i-1})_Z\big). \tag{26}$$

It is to be noted that this algorithm does not need to know the solution $\mathbf{u}$ of problem **(4)** since $\mathbf{e}_{i-1}$ is defined in terms of the current residual $\mathbf{f}_i = \mathbf{f} - \mathcal{A}\mathbf{u}_{i-1}$ by

$$\begin{cases} \mathbf{e}_{i-1} \in L^2_\mu(\Gamma;H) \ \text{ such that} \\ \bar{a}(\mathbf{e}_{i-1},\mathbf{v}) = \langle \mathbf{f},\mathbf{v}\rangle - \bar{a}(\mathbf{u}_{i-1},\mathbf{v}) := \langle \mathbf{f}_i,\mathbf{v}\rangle \quad \forall \mathbf{v} \in L^2_\mu(\Gamma;H) \end{cases}$$

The convergence of the $\mathbf{u}_N$ to $\mathbf{u}$ is stated in Theorem 5.3 of [5] as the form $\bar{b}$ is an inner product in $L^2_\mu(\Gamma;H)$, with the generated norm equivalent to the standard one.

Theorem 4.1 The truncated series $\mathbf{u}_N$ determined by the deflation algorithm **(24)**–**(26)** satisfying

$$\lim_{N\to\infty} \bar{b}(\mathbf{u}-\mathbf{u}_N, \mathbf{u}-\mathbf{u}_N) = 0.$$

Consequently, $\mathbf{u}_N$ strongly converges in $L^2_\mu(\Gamma;H)$ to the solution $\mathbf{u}$ of problem **(4)**.

5 RANK-ONE APPROXIMATIONS

An interesting case from the application point of view arises when we consider rank-one approximations. Indeed, when $k = 1$, the solution of **(12)** $\mathbf{u}_Z$ can be obtained as

$$\mathbf{u}_Z(\gamma) = \varphi(\gamma)\, w, \quad \mu-\text{a.e. } \gamma \in \Gamma, \text{ for some } \varphi \in L^2_\mu(\Gamma),\ w \in H.$$

Then, the problem **(11)** can be written as (see [5], Sect. 6)

$$\min_{v\in H,\ \psi \in L^2_\mu(\Gamma)} J(v,\psi), \tag{27}$$

where

$$J(v,\psi) = \bar{b}(\mathbf{u} - \psi\otimes v, \mathbf{u} - \psi \otimes v). \tag{28}$$

Any solution of problem **(27)** has to verify the following conditions:

Proposition 1 If $(w,\varphi) \in H \times L^2_\mu(\Gamma)$ is a solution of problem **(27)**, then it is also a solution of the following coupled nonlinear problem:

$$\bar{a}\big(\varphi\otimes w, \varphi\otimes A_\gamma v\big) = \langle \mathbf{f}, \varphi\otimes A_\gamma v\rangle \quad \forall\, v \in H, \tag{29}$$

$$\bar{a}\big(\varphi\otimes w, \psi\otimes A_\gamma w\big) = \langle \mathbf{f}, \psi\otimes A_\gamma w\rangle \quad \forall\, \psi \in L^2_\mu(\Gamma). \tag{30}$$

We omit the proof of this result for brevity; let us just mention that conditions **(29)** and **(30)** are the first-order optimality conditions of the problem **(27)** that take place as the functional $J\colon H \times L^2_\mu(\Gamma) \to \mathbb{R}$ and is Gateaux-differentiable. It is to be noted that the PGD method corresponds to replacing A_γ by the identity operator in **(29)**–**(30)**. From Theorem 2.1 and Proposition 1, there exists at least a solution to problem **(27)** and then to problem **(29)**–**(30)**. However, as functional J is not convex, there is no warranty that it admits a unique minimum. Then, a solution to problem **(29)**–**(30)** could not be a solution to the problem **(27)**.

Relations **(29)**–**(30)** suggest an alternative way to compute the modes in the PGD expansion to solve **(1)**. Indeed, we propose a LRTD expansion for $\mathbf{u}$ given by

$$\mathbf{u} \simeq \mathbf{u}_N := \sum_{i=1}^{N} \varphi_i \otimes w_i, \tag{31}$$

where the modes $(w_i,\varphi_i) \in H \times L^2_\mu(\Gamma)$ are recursively computed as a solution of the problem $(\mathbf{f}_i := \mathbf{f} - \mathcal{A}\mathbf{u}_{i-1})$:

$$\begin{cases} \bar{a}\big(\varphi_i\otimes w_i, \varphi_i\otimes A_\gamma v\big) = \langle \mathbf{f}_i, \varphi_i\otimes A_\gamma v\rangle & \forall\, v \in H, \\ \bar{a}\big(\varphi_i\otimes w_i, \psi\otimes A_\gamma w_i\big) = \langle \mathbf{f}_i, \psi\otimes A_\gamma w_i\rangle & \forall\, \psi \in L^2_\mu(\Gamma). \end{cases} \tag{32}$$

If this problem is solved in such a way that its solution is also a solution of

$$\begin{aligned} &\min_{v\in H,\ \psi\in L^2_\mu(\Gamma)} J_i(v,\psi),\\ &\text{with } J_i(v,\psi) = \bar{b}\big(\mathbf{u} - \mathbf{u}_{i-1} - \psi\otimes v, \mathbf{u} - \mathbf{u}_{i-1} - \psi\otimes v\big), \end{aligned} \tag{33}$$

then expansion **(31)** will be optimal in the sense of the expansion **(24)**, where each mode of the series is computed in an optimal finite-dimensional subspace that minimizes the error.

6 COMPUTATION OF LOW-RANK TENSORIZED DECOMPOSITION MODES

In this section, we analyze the solution of the nonlinear problem **(32)** by the power iteration (PI) method. As the operator A_γ appears in the test functions, a specific treatment is needed, in particular, to compute targeted-norm subspaces. We also introduce a simplified PI algorithm that does not need to compute A_γ. Here, we report both.

We focus on solving the model problem **(29)**–**(30)**, which we assume to admit a nonzero solution. For simplicity, the notation $\mathbf{f}$ will stand either for the r. h. s. of the problem **(4)** or for any residual $\mathbf{f}_i$.

It is to be observed that **(30)** is equivalent to

$$\int_\Gamma \varphi(\gamma)\, a\big(w,\, A_\gamma w;\, \gamma\big)\, \psi(\gamma)\, d\mu(\gamma) \;= \int_\Gamma \langle \mathbf{f}(\gamma),\, A_\gamma w\rangle\, \psi(\gamma)\, d\mu(\gamma), \quad \forall\, \psi \in L^2_\mu(\Gamma),$$

and then

$$\varphi(\gamma) = \varphi(w,\gamma) \coloneqq \frac{\langle \mathbf{f}(\gamma),\, A_\gamma w\rangle}{a\big(w,\, A_\gamma w;\, \gamma\big)} \quad \mu - \text{a.e. } \gamma \in \Gamma. \tag{34}$$

Thus, problem **(29)**–**(30)** consists in

$$\begin{cases} \text{Find } w \in H \text{ such that} \\ \displaystyle\int_\Gamma \varphi(w,\gamma)^2\, a\big(w,\, A_\gamma v;\, \gamma\big)\, d\mu(\gamma) \;= \\ \displaystyle\int_\Gamma \varphi(w,\gamma)\, \langle \mathbf{f}(\gamma),\, A_\gamma v\rangle\, d\mu(\gamma), \quad \forall\, v \in H. \end{cases} \tag{35}$$

with $\varphi(w,\cdot) \in L^2_\mu(\Gamma)$ defined by **(34)**. For simplicity, we shall denote by $\varphi(w)$ the function $\varphi(w,\cdot)$.

Let us define the operator $T : H \to H$ that transforms $w \in H$ into $T(w) \in H$ solution to the problem

$$\int_\Gamma \varphi(w,\gamma)^2 a\big(T(w), A_\gamma v;\, \gamma\big)\, d\mu(\gamma) = \int_\Gamma \varphi(w,\gamma)\langle \mathbf{f}(\gamma), A_\gamma v\rangle\, d\mu(\gamma), \; \forall\, v \in H. \tag{36}$$

T is well-defined from **(2)** and **(3)**, and a solution of **(35)** is a fixed point of this operator. Moreover, as A_γ is linear,

$$\varphi(\lambda\, w) = \lambda^{-1}\, \varphi(w) \quad \text{and then} \quad T(\lambda\, w) = \lambda\, T(w), \quad \forall \lambda \in \mathbb{R}.$$

Thus, if (w, φ) is a solution to **(29)**–**(30)**, then $(\lambda\, w, \lambda^{-1}\varphi)$ is also a solution to this problem. So, we propose to find a solution to problem **(35)** with unit norm. For that, we apply the following PI algorithm with normalization:

Initialization:Given any nonzero $w^0 \in H$ such that $\varphi^0 = \varphi(w^0)$ is not zero in $L^2_\mu(\Gamma)$.

Iteration: Knowing $w^n \in H$, the following is computed :

$$\begin{aligned} &\text{i)} \quad \tilde{w}^{n+1} = T(w^n) \in H \\ &\text{ii)} \quad w^{n+1} = \frac{\tilde{w}^{n+1}}{\|\tilde{w}^{n+1}\|} \in H \\ &\text{iii)} \quad \varphi^{n+1} = \varphi\big(w^{n+1}\big) \in L^2_\mu(\Gamma). \end{aligned} \tag{37}$$

The next result states that this iterative procedure is well-defined.

Lemma 2 It is assumed that for some nonzero $w \in H$, it holds that $\varphi(w) \in L^2_\mu(\Gamma)$ is not zero. Then $\tilde{w} = T(w)$ is not zero in H *and* $\varphi(\tilde{w})$ is not zero in $L^2_\mu(\Gamma)$.

Proof First, by reduction to the absurd, it is assumed that $T(w) = 0$. From **(36)**, we have

$$\int_\Gamma \varphi(w,\gamma)\, \langle \mathbf{f}(\gamma),\, A_\gamma v\rangle\, d\mu(\gamma) = 0, \quad \forall\, v \in H.$$

In particular, for $v = w$ and taking into account **(34)**, we deduce that

$$\int_\Gamma \frac{|\langle \mathbf{f}(\gamma),\, A_\gamma w\rangle|^2}{a\big(w, A_\gamma w; \gamma\big)}\, d\mu(\gamma) = 0.$$

Then,

$$\langle \mathbf{f}(\gamma),\, A_\gamma w\rangle = 0, \quad \mu - \text{a.e. } \gamma \in \Gamma,$$

and thus, $\varphi(w) = 0$ is in contradiction with the initial hypothesis. This proves that $T(w)$ is not zero. Second, arguing again by reduction to the absurd, it is assumed that $\varphi(\tilde{w}) = 0$. From **(34)**, we have

$$\langle \mathbf{f}(\gamma),\, A_\gamma \tilde{w}\rangle = 0, \quad \mu - \text{a.e. } \gamma \in \Gamma.$$

Then, setting $v = \tilde{w}$ in **(36)** and using **(19)**, we obtain

$$\bar{b}(\varphi(w)\otimes\tilde{w}, \varphi(w)\otimes\tilde{w}) = \int_\Gamma a\big(\varphi(w,\gamma)\,\tilde{w}, \varphi(w,\gamma)\, A_\gamma \tilde{w}\big)\, d\mu(\gamma) = 0.$$

As $\bar{b}$ is a scalar product in $L^2_\mu(\Gamma; H)$, this implies that

$$\|\varphi(w)\otimes\tilde{w}\|_{L^2_\mu(\Gamma;H)} = \|\varphi(w)\|_{L^2_\mu(\Gamma)}\, \|\tilde{w}\| = 0.$$

We have already proven that $\tilde{w} \neq 0$. So, $\varphi(w)$ has to be equal to zero, in contradiction with the initial hypothesis. Thus, our assumption is false and $\varphi(\tilde{w})$ is not zero.

This result proves that if w^0 and φ^0 are not zero, then the algorithm **(37)** is well-defined.

6.1 Computation of Power Iteration Algorithm for Targeted-Norm Optimal Subspaces

From a practical point of view, in general, the algorithm **(37)** is computationally expensive. Indeed, in practice, H is a space of large finite-dimension and the integral in Γ is approximated by some quadrature rule with nodes $\{\gamma_i\}_{i=1}^M$. The method requires the computation of $A_{\gamma_i} v$ for all the elements v on a basis of H and all the γ_i.

It is to be noted that when targeted subspaces are searched for, in the way considered in **Section 3**, the expression of algorithm **(37)** simplifies. Indeed, as $a\,(w, A_{\gamma,\star} v;\, \gamma) = (w,v)_{H,\gamma}$, then

$$\varphi(w,\gamma) = \frac{\langle \mathbf{f}(\gamma),\, A_{\gamma,\star} w\rangle}{\|w\|^2_{H,\gamma}} \quad \mu - \text{a.e. } \gamma \in \Gamma. \tag{38}$$

In addition, the problem **(36)** that defines the operator T simplifies to problem:

$$\int_\Gamma \varphi(w,\gamma)^2\,(T(w),v)_{H,\gamma}\,d\mu(\gamma) = \int_\Gamma \varphi(w,\gamma)\langle \mathbf{f}(\gamma), A_{\gamma,\star}v\rangle\,d\mu(\gamma), \quad \forall\, v\in H. \tag{39}$$

We shall refer to the method **(38)**–**(39)** as the TN (targeted-norm) method.

6.2 A Simplified Power Iteration Algorithm

An approximate, but less expensive, method is derived from the observation that A_γ is an isomorphism in H. If we approximate the first equation in **(32)** by

$$\bar a\big(\varphi_i\otimes w_i, \varphi_i\otimes A_{\gamma_0}v\big) = \langle \mathbf{f}_i, \varphi_i\otimes A_{\gamma_0}v\rangle \quad \forall\, v\in H,$$

for some $\gamma_0\in\Gamma$, then this equation is equivalent to

$$\bar a\big(\varphi_i\otimes w_i, \varphi_i\otimes v\big) = \langle \mathbf{f}_i, \varphi_i\otimes v\rangle \quad \forall\, v\in H. \tag{40}$$

We then consider the following adaptation of the PI method **(37)** to compute an approximation of the solution of the optimality conditions **(32)**:

Iteration: Known $w^n\in H$, the following is computed:

$$\begin{aligned} &\text{i)} \quad \tilde w^{n+1} = \hat T(w^n)\in H\\ &\textit{ii)} \quad w^{n+1} = \frac{\tilde w^{n+1}}{\|\tilde w^{n+1}\|}\in H\\ &\text{iii)} \quad \varphi^{n+1} = \varphi\big(w^{n+1}\big)\in L^2_\mu(\Gamma). \end{aligned} \tag{41}$$

where $\hat T(w)$ is computed by

$$\int_\Gamma \varphi(w,\gamma)^2\,a\big(\hat T(w),v;\,\gamma\big)\,d\mu(\gamma) = \int_\Gamma \varphi(w,\gamma)\,\langle \mathbf{f}(\gamma),\,v\rangle\,d\mu(\gamma), \quad \forall\, v\in H,, \tag{42}$$

where $\varphi(w,\,\gamma)$ is defined by **(34)**. We shall refer to method **(41)**–**(42)** as the STN (simplified targeted-norm) method. The difference between the STN method and the standard PGD one is only the definition of the function φ that in this case is given by

$$\varphi(\gamma) \coloneqq \frac{\langle \mathbf{f}(\gamma),\,w\rangle}{a(w,\,w;\,\gamma)} \quad \mu-\text{a.e. } \gamma\in\Gamma.$$

6.3 Convergence of the Power Iteration Algorithms

In this section, we analyze the convergence of the PI algorithms **(37)** and **(41)** (Theorem 6.1). We prove that the method with optimal convergence rate corresponds to the operator $A_{\gamma,\star}$ introduced in **Section 3**, choosing all the inner products $\|\cdot\|_{H,\gamma}$ equal to the reference inner product $\|\cdot\|$.

As in [5], we shall assume that the iterates φ^n remain in the exterior of some ball of positive radius, say $\varepsilon>0$, of $L^2_\mu(\Gamma)$, that is,

$$\|\varphi^n\|_{L^2_\mu(\Gamma)}\ge\varepsilon, \quad n=0,\,1,\,2,\,\cdots. \tag{43}$$

This is a working hypothesis that makes sense whenever the mode that we intend to compute is not zero, considering that the w^n is normalized.

From the definition of A_γ and **(6)**, it holds

$$\tilde\alpha\,\|v\|\le\|A_\gamma v\|_\gamma\le\tilde\beta\,\|v\| \quad \text{for all } \gamma\in\Gamma, \quad \text{with } \tilde\alpha=\frac{\alpha}{\beta_H},\quad \tilde\beta=\frac{\beta}{\alpha_H}, \tag{44}$$

where α and β are given by **(2)** and **(3)**, respectively, and α_H and β_H are defined in **(6)**.

Let us define the function:

$$\delta(r,s) = 2\,\lambda\,k^2\left(1+\frac{2-r}{1-r}\,k^2\right)\left(1+\frac{2-r}{1-r}\,\lambda\,k^2\,s\right)s^2,$$

$$\text{where } k=\frac{\tilde\beta}{\tilde\alpha}, \quad \text{and} \quad \lambda=\begin{cases} k^2 & \text{for method (37)}\\ \dfrac{\beta}{\alpha} & \text{for method (41)}\end{cases}$$

It holds.

Theorem 6.1 It is assumed that **(43)** holds, and

$$\Delta=\delta(r,\bar s)<1, \quad \textit{for some } r\in(0,1) \textit{ with } \bar s=\frac{\|\mathbf{u}\|_{L^2_\mu(\Gamma;H)}}{\varepsilon}. \tag{45}$$

Then, there exists a unique solution with norm 1 w of problem **(35)**; the sequence $\{w^n\}_{n\ge1}$ computed by either method **(37)** or method **(41)** is contained in the ball $B_H(w,r)$ *if* $w^0\in B_H(w,r)$, and

$$\|w-w^{n+1}\|\le\Delta\,\|w-w^n\|, \quad \forall\, n\ge0. \tag{46}$$

As a consequence, the sequence $\{w^n\}_{n\ge1}$ *that is defined by either method* ***(37)*** *or method* ***(41)*** *is strongly convergent to w with linear rate and the following error estimate holds:*

$$\|w-w^n\|\le\Delta^n\,\|w-w^0\|, \quad \forall\, n\ge1, \quad \textit{whenever } w^0\in B_H(w,r). \tag{47}$$

Proof. Let us consider at first the method **(37)**. Let $x\in B_H(w,r)$ such that $\|\varphi(x)\|_{L^2_\mu(\Gamma)}\ge\varepsilon$. Denote $\tilde x=T(x)$ which by the definition of operator T in **(36)** is the solution to the problem

$$\int_\Gamma \varphi(x,\gamma)^2\,a\big(\tilde x,\,A_\gamma v;\,\gamma\big)\,d\mu(\gamma) = \int_\Gamma \varphi(x,\gamma)\,\langle \mathbf{f}(\gamma),\,A_\gamma v\rangle\,d\mu(\gamma), \quad \forall\, v\in H. \tag{48}$$

We aim to estimate $\|w-\frac{\tilde x}{\|\tilde x\|}\|$. To do that, from problems **(35)** and **(48)**, we obtain

$$\begin{aligned} &\int_\Gamma \varphi(w,\gamma)^2\,a\big(w-\tilde x,\,A_\gamma v;\,\gamma\big)\,d\mu(\gamma)\\ &= -\int_\Gamma\Big(\varphi(w,\gamma)^2-\varphi(x,\gamma)^2\Big)\,a\big(\tilde x,\,A_\gamma v;\,\gamma\big)\,d\mu(\gamma)\\ &\quad+\int_\Gamma\big(\varphi(w,\gamma)-\varphi(x,\gamma)\big)\quad\langle \mathbf{f}(\gamma),\,A_\gamma v\rangle\,d\mu(\gamma), \quad \forall\, v\in H. \end{aligned} \tag{49}$$

It holds

$$a\big(y,\,A_\gamma z;\,\gamma\big)=\big(A_\gamma y,\,A_\gamma z\big)_\gamma\le\tilde\beta^2\,\|y\|\,\|z\|, \quad \forall\, y,\,z\in H, \tag{50}$$

using **(44)**. Thus,

$$\langle \mathbf{f}(\gamma),\,A_\gamma v\rangle = a\big(u(\gamma),A_\gamma v;\gamma\big)\le\tilde\beta^2\,\|u(\gamma)\|\,\|v\|, \quad \forall\, v\in H. \tag{51}$$

Moreover,

$$a\big(y,\ A_\gamma y;\ \gamma\big) = \big(A_\gamma y,\ A_\gamma y\big)_\gamma \geq \tilde{\alpha}^2\ \|y\|^2 \quad \forall\, y \in H. \tag{52}$$

Setting $v = w - \tilde{x}$ in **(62)** and using **(50)**-**(52)**, we have

$$\tilde{\alpha}^2\ \|\varphi(w)\|^2_{L^2_\mu(\Gamma)}\ \|w-\tilde{x}\|^2 \leq \tilde{\beta}^2\ \Big(\|\varphi^2(w)-\varphi^2(x)\|_{L^1_\mu(\Gamma)}\ \|\tilde{x}\| + \|\varphi(w)-\varphi(x)\|_{L^2_\mu(\Gamma)}\ \|\mathbf{u}\|_{L^2_\mu(\Gamma;H)}\Big)\ \|w-\tilde{x}\|. \tag{53}$$

To bound the second term in the r. h s. of **(53)**, from **(34)**, it holds

$$\varphi(w,\gamma)-\varphi(x,\gamma) = \frac{a\big(u(\gamma),\ A_\gamma(w-x);\gamma\big)}{a\big(w,A_\gamma w;\gamma\big)} + \frac{a\big(x,A_\gamma(x-w);\gamma\big)+a\big(x-w,A_\gamma w;\gamma\big)}{a\big(w,A_\gamma w;\gamma\big)\,a\big(x,A_\gamma x;\gamma\big)}\quad a\big(u(\gamma),\ A_\gamma x;\gamma\big).$$

Then, using **(50)** and **(52)**,

$$\begin{aligned}|\varphi(w,\gamma)-\varphi(x,\gamma)| &\leq k^2\left(1+k^2\left(\frac{1}{\|x\|}+1\right)\right)\|u(\gamma)\|\ \|w-x\| \\ &\leq k^2\left(1+\frac{2-r}{1-r}\,k^2\right)\|u(\gamma)\|\ \|w-x\|,\end{aligned} \tag{54}$$

where $k = \frac{\tilde{\beta}}{\tilde{\alpha}}$. In the last estimate, we have used that as $x \in B_H(w,r)$, $\|w\| - r \leq \|x\|$ and then $\frac{1}{\|x\|} \leq \frac{1}{1-r}$. Therefore,

$$\|\varphi(w)-\varphi(x)\|_{L^2_\mu(\Gamma)} \leq \phi_1(r)\ \|u\|_{L^2_\mu(\Gamma;H)}\ \|w-x\|, \tag{55}$$

where

$$\phi_1(r) = k^2\left(1+\frac{2-r}{1-r}\,k^2\right).$$

It is to be noted that from **(34)**,

$$|\varphi(z,\gamma)| = \left|\frac{a\big(u(\gamma),A_\gamma z;\gamma\big)}{a\big(z,A_\gamma z;\gamma\big)}\right| \leq k^2\ \frac{\|u(\gamma)\|}{\|z\|}, \quad \forall z \in H. \tag{56}$$

Then, from **(54)** and **(56)**

$$\begin{aligned}|\varphi(w,\gamma)^2-\varphi(x,\gamma)^2| &\leq |\varphi(w,\gamma)-\varphi(x,\gamma)|\,|\varphi(w,\gamma)+\varphi(x,\gamma)| \\ &\leq k^2\,\phi_1(r)\left(1+\frac{1}{\|x\|}\right)\|u(\gamma)\|^2\ \|w-x\|.\end{aligned}$$

Hence,

$$\|\varphi(w)^2-\varphi(x)^2\|_{L^1_\mu(\Gamma)} \leq \phi_2(r)\ \|\mathbf{u}\|^2_{L^2_\mu(\Gamma;H)}\ \|w-x\|, \tag{57}$$

where

$$\phi_2(r) = k^2\,\phi_1(r)\,\frac{2-r}{1-r}.$$

Combining **(53)** with **(55)** and **(57)**, we deduce

$$\|w-\tilde{x}\| \leq \left(\frac{\|\mathbf{u}\|_{L^2_\mu(\Gamma;H)}}{\|\varphi(w)\|_{L^2_\mu(\Gamma)}}\right)^2 k^2\ \big(\phi_1(r)+\phi_2(r)\,\|\tilde{x}\|\big)\ \|w-x\|. \tag{58}$$

Setting $v = \tilde{x}$ in **(48)** and using **(52)** and **(51)**, we obtain

$$\tilde{\alpha}^2\ \|\varphi(x)\|^2_{L^2_\mu(\Gamma)}\ \|\tilde{x}\|^2 \leq \tilde{\beta}^2\ \|\varphi(x)\|_{L^2_\mu(\Gamma)}\ \|\mathbf{u}\|_{L^2_\mu(\Gamma;H)}\ \|\tilde{x}\|.$$

Thus,

$$\|\tilde{x}\| \leq k^2\ \frac{\|\mathbf{u}\|_{L^2_\mu(\Gamma;H)}}{\|\varphi(x)\|_{L^2_\mu(\Gamma)}} \leq k^2\ \frac{\|\mathbf{u}\|_{L^2_\mu(\Gamma;H)}}{\varepsilon} = k^2\,\bar{s}. \tag{59}$$

It holds $\|\frac{w}{\|w\|} - \frac{\tilde{x}}{\|\tilde{x}\|}\| \leq 2\,\|w-\tilde{x}\|$. Then, using **(58)** and **(59)**, we deduce

$$\left\|\frac{w}{\|w\|}-\frac{\tilde{x}}{\|\tilde{x}\|}\right\| \leq 2\,k^2\,\bar{s}^2\ \big(\phi_1(r)+\phi_2(r)\,k^2\,\bar{s}\big)\ \|w-x\|.$$

That is,

$$\left\|w-\frac{\tilde{x}}{\|\tilde{x}\|}\right\| \leq \Delta\ \|w-x\|, \quad \text{with}\ \Delta\ \text{given by}\ \ (\mathbf{45}). \tag{60}$$

Estimate **(46)** follows from this last inequality for $x = w^n$, assuming that $w^n \in B_H(w,r)$. Assuming $w^0 \in B_H(w,r)$ this recursively proves that all the w^n are in $B_H(w,r)$. Furthermore, suppose that there exists another solution to **(35)** with norm one in the ball $B_H(w,r)$, $w^\star$. In this case, estimating **(60)** for $x = w^\star$ implies

$$\|w-w^\star\| \leq \Delta\ \|w-w^\star\|$$

because $\tilde{x} = T(w^\star) = w^\star$. Then $w = w^\star$, and there is uniqueness of solution with norm one in the ball $B_H(w,r)$. Finally, **(47)** follows from **(46)** by recurrence.

Let us now consider method **(41)**. In this case $\tilde{x} = \hat{T}(x)$, by **(42)**, is the solution of the problem

$$\int_\Gamma \varphi(x,\gamma)^2\ a(\tilde{x},\ v;\ \gamma)\,d\mu(\gamma) = \int_\Gamma \varphi(x,\gamma)\ \langle \mathbf{f}(\gamma),\ v\rangle\ d\mu(\gamma), \quad \forall\, v \in H. \tag{61}$$

To estimate $\|w - \frac{\tilde{x}}{\|\tilde{x}\|}\|$, from problems **(35)** and **(61)**, we obtain

$$\begin{aligned}\int_\Gamma \varphi(w,\gamma)^2\ a(w-\tilde{x},\ v;\ \gamma)\,d\mu(\gamma) &= -\int_\Gamma \big(\varphi(w,\gamma)^2-\varphi(x,\gamma)^2\big)\ a(\tilde{x},\ v;\ \gamma)\,d\mu(\gamma) \\ &+ \int_\Gamma \big(\varphi(w,\gamma)-\varphi(x,\gamma)\big)\ \langle \mathbf{f}(\gamma),\ v\rangle\ d\mu(\gamma), \quad \forall\, v \in H.\end{aligned} \tag{62}$$

As $\langle \mathbf{f}(\gamma),\ v\rangle = a\ (\mathbf{u}(\gamma),\ v;\ \gamma) \leq \beta\ \|\mathbf{u}(\gamma)\|\ \|v\|$, $\forall\ v \in H$, setting $v = w - \tilde{x}$, we have

TABLE 1 | Convergence rates of the PI algorithm for the first TN modes.

n	Mode i = 1		Mode i = 2		Mode i = 3	
	$\|\mathbf{w}_1^n - \mathbf{w}_1^n - \mathbf{1}\|$	$\mathbf{r}_1^n$	$\|\mathbf{w}_2^z\mathbf{n} - \mathbf{w}_2^{n-1}\|$	$\mathbf{r}_2^n$	$\|\mathbf{w}_3^n - \mathbf{w}_3^{n-1}\|$	$\mathbf{r}_3^n$
1	7,4844E-01	—	4,6117E-01	—	3,8004E-02	—
2	9,6123E-02	7,78	8,3125E-02	5,54	5,9211E-02	0,641
3	4,1109E-03	23,38	8,3769E-03	9,92	6,8704E-03	8,61
4	1,7378E-04	23,65	7,9166E-04	10,58	7,5651E-04	9,08
5	7,3741E-06	23,56	7,4417E-05	10,63	8,2797E-04	9,13
6	3,1308E-07	23,55	6,9923E-06	10,64	9,0557E-06	9,14
7	—	—	6,5699E-07	10,64	9,9037E-07	9,14

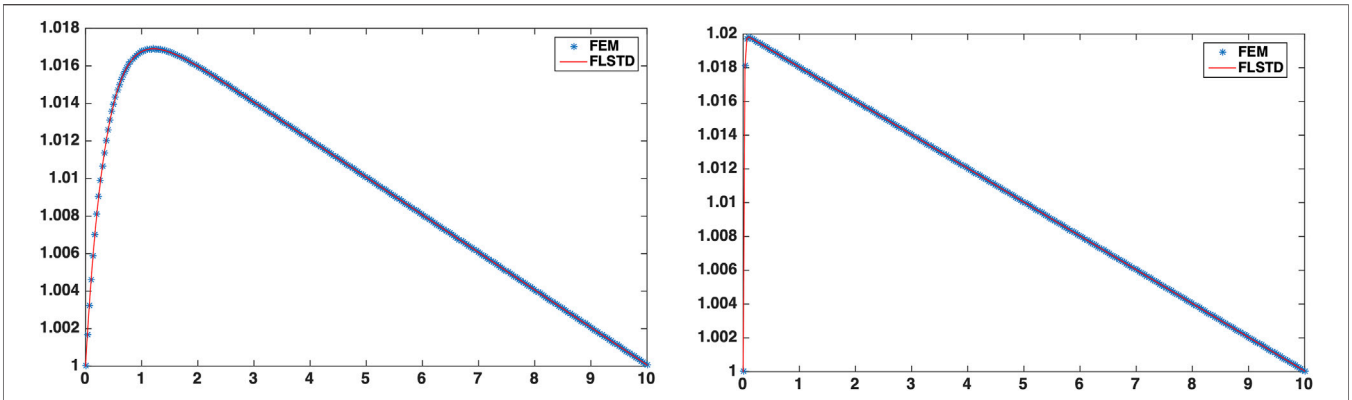


FIGURE 1 | Solutions of problem **(66)** by the LRTD (called FLSTD within the figure) expansion computed with the TN method. Left $\gamma = 2.7375$ Right $\gamma = 49.7624$.

TABLE 2 | Convergence rates of the PI algorithm for the first STN modes.

n	Mode i = 1		Mode i = 2		Mode i = 3	
	$\|\mathbf{w}_1^n - \mathbf{w}_1^{n-1}\|$	$\mathbf{r}_1^n$	$\|\mathbf{w}_2^n - \mathbf{w}_2^{n-1}\|$	$\mathbf{r}_2^n$	$\|\mathbf{w}_3^n - \mathbf{w}_3^{n-1}\|$	$\mathbf{r}_3^n$
1	8,0749E-1	—	5,0427E-1	—	3,9947E-1	—
2	1,4643E-1	5,51	1,7004E-1	2,96	1,1811E-1	3,38
3	9.8900E-3	14,08	4,0532E-2	4,19	8,9310E-2	1,32
4	6,0098E-4	16,45	8,8019E-3	4,60	4,1810E-2	2,13
5	3,6271E-5	16,56	1,8704E-3	4,70	1,8745E-2	2,23
6	2,1882E-6	16,57	3,9656E-4	4,72	8,2109E-3	2,28
7	1,3201E-7	16,57	8,3595E-5	4,73	3,5573E-3	2,30
8	7,9636E-9	16,57	1,7660E-5	4,73	1,5337E-3	2,31
9	—	—	—	—	6,5980E-4	2,32

$$\alpha \|\varphi(w)\|^2_{L^2_\mu(\Gamma)} \|w - \tilde{x}\|^2 \le \int_\Gamma \varphi(w,\gamma)^2 a(w-\tilde{x},\, w-\tilde{x};\, \gamma)\, d\mu(\gamma) \le$$
$$\beta \left(\|\varphi^2(w) - \varphi^2(x)\|_{L^1_\mu(\Gamma)} \|\tilde{x}\| + \|\varphi(w) - \varphi(x)\|_{L^2_\mu(\Gamma)} \|\mathbf{u}\|_{L^2_\mu(\Gamma;H)} \right) \|w - \tilde{x}\|. \tag{63}$$

The functions $\varphi(w)$ and $\varphi(x)$ have the same expressions for methods **(37)** and **(41)**.

Moreover, setting $v = \tilde{x}$ in **(61)**

$$\begin{aligned} \alpha \|\varphi(w)\|^2_{L^2_\mu(\Gamma)} \|\tilde{x}\|^2 &\le \int_\Gamma \varphi(w,\gamma)^2 a(\tilde{x},\, \tilde{x};\, \gamma)\, d\mu(\gamma) \\ &= \int_\Gamma \varphi(w,\gamma) \langle f(\gamma), \tilde{x} \rangle\, d\mu(\gamma) \\ &= \int_\Gamma a(\mathbf{u}(\gamma),\, \varphi(w,\gamma)(\tilde{x});\, \gamma)\, d\mu(\gamma) \\ &\le \beta \|\mathbf{u}\|_{L^2_\mu(\Gamma;H)} \|\varphi(w)\|_{L^2_\mu(\Gamma)} \|\tilde{x}\|. \end{aligned} \tag{64}$$

Hence, $\|\tilde{x}\| \le \frac{\beta}{\alpha}\, \bar{s}$. Then, similarly to **(58)**, we obtain

$$\begin{aligned} \|w - \tilde{x}\| &\le \frac{\beta}{\alpha}\, \bar{s}^2 \left(\phi_1(r) + \phi_2(r) \|\tilde{x}\| \right) \|w - x\| \\ &\le \frac{\beta}{\alpha}\, \bar{s}^2 \left(\phi_1(r) + \phi_2(r) \frac{\beta}{\alpha}\, \bar{s} \right) \|w - x\|. \end{aligned} \tag{65}$$

As $\|w - \frac{\tilde{x}}{\|\tilde{x}\|}\| \le 2\|w - \tilde{x}\|$, the conclusion follows as for method **(37)**.

TABLE 3 | Numerical behavior of the PGD, STN, and TN methods for Test 2.

Methods	Errors $\mathbf{L}^2_\mu(\Gamma, \mathbf{L}^2(\Omega))$	Errors $\mathbf{L}^2_\mu(\Gamma, \mathbf{H}^1_0(\Omega))$
PGD	$\|u - u_{94}\| = 9.69e-7$	$\|u - u_{94}\| = 3.30e-7$
STN	$\|u - u_{24}\| = 8.21e-7$	$\|u - u_{24}\| = 3.83e-7$
TN	$\|u - u_{12}\| = 6.31e-7$	$\|u - u_{12}\| = 3.52e-7$

Remark 2 *The optimal convergence rate Δ corresponds to $k = 1$ and $\lambda = 1$, that is, $\tilde{\alpha} = \tilde{\beta}$ and $\alpha = \beta$. As α and β are predetermined, the optimal convergence rate can only be obtained with method* **(37)**. *When the inner products $(\cdot,\cdot)_\gamma = (\cdot,\cdot)_{\gamma,\star}$ and thus the operator $A_\gamma = A_{\gamma,\star}$, introduced in Section* 3 are used *to construct the optimal targeted subspaces, it satisfies, by* **(15)**,

$$\|A_{\gamma,\star} v\|^2_{\gamma,\star} = \|v\|^2_{H,\gamma}, \quad \forall\, v \in H.$$

Then, from **(44)**, *choosing all the inner products $\|\cdot\|_{H,\gamma}$ equal to the reference inner product $\|\cdot\|$, we obtain $\tilde{\alpha} = \tilde{\beta} = 1$. Therefore, the convergence rate is optimal for method* **(37)** *with this choice. It can*

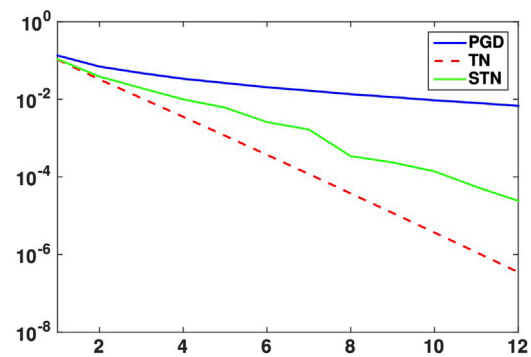


FIGURE 2 | Convergence history of the PGD, TN, and STN series for Test 2.

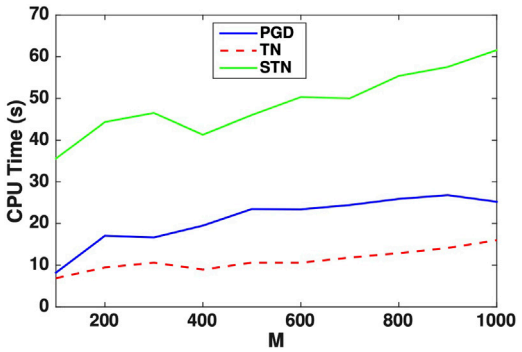


FIGURE 3 | Comparison of CPU times to compute the TN, STN, and PGD expansions.

be interpreted as preconditioning of the problem to solve, similar to classical preconditioning to accelerate convergence in solving linear systems (For example, see [24]).

Remark 3 *If we intend to compute a mode of order* $i \geq 2$, *Theorem* 6.1 *and Remark* 2 *also hold, replacing* $\mathbf{f}$ *by the residual* $\mathbf{f}_i = \mathbf{f} - \mathcal{A}\mathbf{u}_{i-1}$ *and* $\mathbf{u}$ *by the error* $\mathbf{u} - \mathbf{u}_{i-1}$, *where* $\mathbf{u}_{i-1}$ *is defined by* **(31)**.

7 NUMERICAL TESTS

In this section, we discuss the numerical results obtained with the methods TN **(38)**–**(39)** and STN **(41)**–**(42)** to solve some non-symmetric second-order PDEs. Our purpose, on the one hand, is to confirm the theoretical results stated in Theorem 4.1 and Theorem 6.1, and on the other hand, to compare the practical performances of these methods with the standard PGD.

We consider a parametric 1D advection–diffusion problem with fixed constant advection velocity β,

$$\begin{cases} \gamma\beta u' - u'' = \gamma f & \text{in} \quad \Omega = (0,10) \\ u(0) = 1, \ u'(10) = 0 \end{cases} \tag{66}$$

We assume that $\beta = 1$, and then γ is the Péclet number. The source term is $f = 1/500$. We have set $\Gamma = [2.5, 50]$; then, there is a large asymmetry of the advection–diffusion operator.

Once the nonhomogenous boundary condition at $x = 0$ is lifted, problem **(66)** is formulated under the general framework (1) when the space H and the bilinear form $a(\cdot,\cdot)$ are given by $H = \{v \in H^1(\Omega) \,|\, v(0) = 0\}$, and

$$a(u, v; \gamma) = \gamma(\beta u', v) + (u', v'). \tag{67}$$

We endow space H with the $H_0^1(\Omega)$ norm (that we still denote $\|\cdot\|$), which is equivalent to the $H^1(\Omega)$ norm on H.

In practice, we replace the continuous problem **(1)** by an approximated one on a finite element space H_h formed by piecewise affine elements. In addition, the integrals on Γ are approximated by a quadrature formula constructed on a subdivision of Γ into M subintervals,

$$\int_\Gamma \psi(\gamma)\, d\mu(\gamma) \simeq I_M(\psi) = \sum_{i=1}^{M} \omega_i\, \psi(\gamma_i).$$

This is equivalent to approximating the Lebesgue measure μ by a discrete measure μ_Δ located at the nodes of the discrete set $\Gamma_\Delta = \{\gamma_i,\ i = 1, \ldots, M\}$ with weights ω_i, $i = 1, \ldots, M$. Consequently, all the theoretical results obtained in the previous sections apply, by replacing the $L^2_\mu(\Gamma, H)$ space by $L^2_{\mu_\Delta}(\Gamma_\Delta, H_h)$. It is to be noted that

$$\|\mathbf{v}\|^2_{L^2_{\mu_\Delta}(\Gamma_\Delta, H_h)} = I_M(\|\mathbf{v}\|^2). \tag{68}$$

In our computations, we have used the midpoint quadrature formula with $M = 100$ equally spaced subintervals of Γ to construct I_M and constructed H_h with 300 subintervals of Ω of the same length.

Test 1:

This first experiment is intended to check the theoretical results on the convergence rate of the PI algorithm, stated in Theorem 6.1, for the TN and STN methods: we consider optimal targeted subspaces, in the sense of the standard $H_0^1(\Omega)$ norm. That is, using

$$(w, v)_{H,\gamma} = (w, v)_{H_0^1(\Omega)} = (w', v'), \quad \forall\, w,\ v \in H^1(\Omega)$$

to define the mappings $A_{\gamma,\star}$ by **(14)** and the form $\bar{b}^\star$ by **(22)**.

For each mode w_i of the LRTD expansion (31), we have estimated the numerical convergence rate of the PI algorithm by

$$r_i^{n+1} = \frac{\|w_i^n - w_i^{n-1}\|}{\|w_i^{n+1} - w_i^n\|}. \tag{69}$$

Tables 1 and **2** show the norm of the difference between two consecutive approximations and the ratios r_i^n. We display the results for the first three modes.

We observe that the PI method converges with a nearly constant rate for each mode, in agreement with Theorem 6.1. The convergence rate is larger for the TN method, also as expected from this theorem. It is also noted that the convergence rates are smaller for higher-order modes.

In **Figure 1**, we present the comparison between the solution obtained by finite elements for $\gamma = 2.7375$ and $\gamma = 49.7625$ and the truncated series sum for the TN, the results for the STN are similar.

Test 2:

In this test, we compare the convergence rates of PGD, TN, and STN methods to obtain the LRTD expansion **(31)** for the problem **(66)**.

Figure 2 displays the errors of the truncated series with respect to the number of modes, in norm $L^2_\mu(\Gamma, H_0^1(\Omega))$. A spectral convergence may be observed for the three expansions. We observe that the convergence of the TN expansion, in terms of the number of modes needed to achieve an error level, is much faster than the convergence of the STN expansion, while this one is faster than the PGD one. This is clarified if we consider the number of modes required to achieve an error smaller than a given level. We

display these numbers for an error level of 10^{-6} in **Table 3**, where much more modes are needed by the PGD expansion. The TN and STN methods, thus, appear to be well-adapted to fit the asymmetry of the operator.

Finally, we compare the CPU times required by the three methods. By construction, it is clear that to compute every single iteration of the PI method, the TN method is much more expensive since it involves the calculation of the $A_{y,*}$ operator for each finite element base function. However, due to the fast convergence of the associated LRTD expansion, it is less expensive than PGD to compute the expansion. **Figure 3** displays the CPU times for the TN, STN, and PGD methods as a function of the number of subintervals M considered in the partition of Γ. The STN method is more expensive than the PGD method; this arises due to the small convergence rate of the PI algorithm with the STN method. However, the TN method is less expensive than the PGD one, requiring approximately half the CPU time.

8 CONCLUSION

In this study, we have proposed a new low-rank tensorized decomposition (LRTD) to approximate the solution of parametric non-symmetric elliptic problems, based on symmetrization of the problem.

Each mode of the series is characterized as a solution to a calculus of variation problem that yields an optimal finite-dimensional subspace, in the sense that it minimizes the error between the parametric solution and its approximation on this subspace, with respect to a preset mean quadratic norm. We have proven that the truncated expansion given by the deflation algorithm strongly converges to the parametric solution in the mean quadratic norm.

The minimization problems to compute the rank-one optimal modes are solved by the power iteration algorithm. We have proven that this method is locally linearly convergent when the initial data are close enough to an optimal mode. We also have identified an optimal symmetrization that provides the best convergence rates of the PI algorithm, with respect to the preset mean quadratic norm.

Furthermore, we have presented some numerical tests for 1D convection–diffusion problems that confirm the theoretical results on the convergence of the LRTD expansion and the convergence of the PI algorithm. Moreover, the computing times required by the optimal symmetrization compare advantageously to those required by the PGD expansion.

In this study, we have focused on rank-one tensorized decompositions. In our forthcoming research, we intend to extend the analysis to ranks $k \geq 2$. This requires solving minimization problems on a Grassmann variety to compute the LRTD modes. We will also work on the solution of higher-dimensional non-symmetric elliptic problems by the method introduced in order to reduce the computation times as these increase with the dimension of the approximation spaces.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

TC and IS performed the theoretical derivations while MG did the computations.

FUNDING

The research of TC and IS has been partially funded by PROGRAMA OPERATIVO FEDER ANDALUCIA 2014-2020 grant US-1254587, which in addition has covered the cost of this publication. The research of MG is partially funded by the Spanish Research Agency - EU FEDER Fund grant RTI2018-093521-B-C31.

REFERENCES

1. Ammar A, Chinesta F, Falcó A. On the Convergence of a Greedy Rank-One Update Algorithm for a Class of Linear Systems. *Arch Computat Methods Eng* (2010) 17:473–86. Number 4. doi:10.1007/s11831-010-9048-z
2. Ammar A, Mokdad B, Chinesta F, Keunings R. A New Family of Solvers for Some Classes of Multidimensional Partial Differential Equations Encountered in Kinetic Theory Modeling of Complex Fluids. *J Non-Newtonian Fluid Mech* (2006) 139:153–76. doi:10.1016/j.jnnfm.2006.07.007
3. Azaïez M, Chacón Rebollo T, Gómez Mármol M. On the Computation of Proper Generalized Decomposition Modes of Parametric Elliptic Problems. *SeMA* (2020) 77:59–72. doi:10.1007/s40324-019-00198-7
4. Azaïez M, Belgacem FB, Rebollo TC. Error Bounds for POD Expansions of Parameterized Transient Temperatures. *Comp Methods Appl Mech Eng* (2016) 305:501–11. doi:10.1016/j.cma.2016.02.016
5. Azaïez M, Belgacem FB, Casado-Díaz J, Rebollo TC, Murat F. A New Algorithm of Proper Generalized Decomposition for Parametric Symmetric Elliptic Problems. *SIAM J Math Anal* (2018) 50(5):5426–45. doi:10.1137/17m1137164
6. ten Berge JMF, de Leeuw J, Kroonenberg PM. Some Additional Results on Principal Components Analysis of Three-Mode Data by Means of Alternating Least Squares Algorithms. *Psychometrika* (1987) 52:183–91. doi:10.1007/bf02294233
7. Bulut H, Akkilic AN, Khalid BJ. Soliton Solutions of Hirota Equation and Hirota-Maccari System by the (m+1/G')-expansion Method. *Adv Math Models Appl* (2021) 6(1):22–30.
8. Le Bris C, Lelièvre T, Maday Y. Results and Questions on a Nonlinear Approximation Approach for Solving High-Dimensional Partial Differential Equations. *Constr Approx* (2009) 30(3):621–51. doi:10.1007/s00365-009-9071-1
9. Cancès E, Lelièvre T, Ehrlacher V. Convergence of a Greedy Algorithm for High-Dimensional Convex Nonlinear Problems. *Math Models Methods Appl Sci* (2011) 21(12):2433–67. doi:10.1142/s0218202511005799

10. Cancès E, Lelièvre T, Ehrlacher V. Greedy Algorithms for High-Dimensional Non-symmetric Linear Problems. *Esaim: Proc* (2013) 41:95–131. doi:10.1051/proc/201341005
11. Chinesta F, Ammar A, Cueto E. Recent Advances and New Challenges in the Use of the Proper Generalized Decomposition for Solving Multidimensional Models. *Arch Computat Methods Eng* (2010) 17(4): 327–50. doi:10.1007/s11831-010-9049-y
12. Espig M, Hackbusch W, Rohwedder T, Schneider R. Variational Calculus with Sums of Elementary Tensors of Fixed Rank. *Numer Math* (2012) 122:469–88. doi:10.1007/s00211-012-0464-x
13. Espig M, Hackbusch W. A Regularized Newton Method for the Efficient Approximation of Tensors Represented in the Canonical Tensor Format. *Numer Math* (2012) 122:489–525. doi:10.1007/s00211-012-0465-9
14. Espig M, Hackbusch W, Khachatryan A. On the Convergence of Alternating Least Squares Optimisation in Tensor Format Representations. arXiv: 1506.00062v1 [math.NA] (2015).
15. Espig M, Hackbusch W, Litvinenko A, Matthies HG, Zander E. Iterative Algorithms for the post-processing of High-Dimensional Data. *J Comput Phys* (2020) 410:109–396. doi:10.1016/j.jcp.2020.109396
16. Falcó A, Nouy A. A Proper Generalized Decomposition for the Solution of Elliptic Problems in Abstract Form by Using a Functional Eckart-Young Approach. *J Math Anal Appl* (2011) 376:469–80. doi:10.1016/j.jmaa.2010.12.003
17. Falcó A, Nouy A. Proper Generalized Decomposition for Nonlinear Convex Problems in Tensor Banach Spaces. *Numer Math* (2012) 121:503–30. doi:10.1007/s00211-011-0437-5
18. Nouy A. Low-rank Tensor Methods for Model Order Reduction. In: *Handbook of Uncertainty Quantification (Roger GhanemDavid HigdonHouman Owhadi*. Eds. Philadelphia, PA: Springer (2017). doi:10.1007/978-3-319-12385-1_21
19. Nouy A. Low-rank Methods for High-Dimensional Approximation and Model Order Reduction. In: P Benner, A Cohen, M Ohlberger, K Willcox, editors. *Model Reduction and Approximations*. Philadelphia, PA: SIAM (2017).
20. Kiers HAL. An Alternating Least Squares Algorithm for PARAFAC2 and Three-Way DEDICOM. *Comput Stat Data Anal* (1993) 16:103–18. doi:10.1016/0167-9473(93)90247-q
21. Kroonenberg PM, de Leeuw J. Principal Component Analysis of Three-Mode Data of using Alternating Least Squares Algorithms. *Psychometrika* (1980) 45: 69–97. doi:10.1007/bf02293599
22. Ladévèze P. *Nonlinear Computational Structural Mechanics. New Approaches and Non-incremental Methods of Calculation*. Berlin: Springer (1999).
23. Mohlenkamp MJ. Musings on Multilinear Fitting. *Linear algebra and its applications* (2013) 438(2): 834–52. doi:10.1016/j.laa.2011.04.019
24. Rasheed SM, Nachaoui A, Hama MF, Jabbar AK. Regularized and Preconditioned Conjugate Gradient Like-Methods Methods for Polynomial Approximation of an Inverse Cauchy Problem. *Adv Math Models Appl* (2021) 6(2):89–105.
25. Uschmajew A. Local Convergence of the Alternating Least Squares Algorithm for Canonical Tensor Approximation. *SIAM J Matrix Anal Appl* (2012) 33(2): 639–52. doi:10.1137/110843587
26. Wang L, Chu MT. On the Global Convergence of the Alternating Least Squares Method for Rank-One Approximation to Generic Tensors. *SIAM J Matrix Anal Appl* (2012) 35(3):1058–72.
27. Zhang T, Golub GH. Rank-one Approximation to High Order Tensors. *SIAM J Matrix Anal Appl* (2001) 23(2):534–50. doi:10.1137/s0895479899352045

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers | Frontiers in Applied Mathematics and Statistics

ORIGINAL RESEARCH
published: 02 June 2022
doi: 10.3389/fams.2022.904687

Multifidelity Ensemble Kalman Filtering Using Surrogate Models Defined by Theory-Guided Autoencoders

Andrey A. Popov and Adrian Sandu*

Computational Science Laboratory, Department of Computer Science, Blacksburg, VA, United States

Data assimilation is a Bayesian inference process that obtains an enhanced understanding of a physical system of interest by fusing information from an inexact physics-based model, and from noisy sparse observations of reality. The multifidelity ensemble Kalman filter (MFEnKF) recently developed by the authors combines a full-order physical model and a hierarchy of reduced order surrogate models in order to increase the computational efficiency of data assimilation. The standard MFEnKF uses linear couplings between models, and is statistically optimal in case of Gaussian probability densities. This work extends the MFEnKF into to make use of a broader class of surrogate model such as those based on machine learning methods such as autoencoders non-linear couplings in between the model hierarchies. We identify the right-invertibility property for autoencoders as being a key predictor of success in the forecasting power of autoencoder-based reduced order models. We propose a methodology that allows us to construct reduced order surrogate models that are more accurate than the ones obtained *via* conventional linear methods. Numerical experiments with the canonical Lorenz'96 model illustrate that nonlinear surrogates perform better than linear projection-based ones in the context of multifidelity ensemble Kalman filtering. We additionality show a large-scale proof-of-concept result with the quasi-geostrophic equations, showing the competitiveness of the method with a traditional reduced order model-based MFEnKF.

Keywords: Bayesian inference, control variates, data assimilation, multifidelity ensemble Kalman filter, reduced order modeling, machine learning, surrogate models frontiers

OPEN ACCESS

Edited by:
Michel Bergmann,
Inria Bordeaux—Sud-Ouest Research Centre, France

Reviewed by:
Zheqi Shen,
Hohai University, China
Feng Bao,
Florida State University, United States

****Correspondence:***
Andrey A. Popov
apopov@vt.edu

Specialty section:
This article was submitted to Statistical and Computational Physics, a section of the journal Frontiers in Applied Mathematics and Statistics

Received: *25 March 2022*
Accepted: *29 April 2022*
Published: *02 June 2022*

Citation:
Popov AA and Sandu A (2022) Multifidelity Ensemble Kalman Filtering Using Surrogate Models Defined by Theory-Guided Autoencoders. Front. Appl. Math. Stat. 8:904687. doi: 10.3389/fams.2022.904687

1. INTRODUCTION

Data assimilation [1, 2] is a Bayesian inference process that fuses information obtained from an inexact physics-based model, and from noisy sparse observations of reality, in order to enhance our understanding of a physical process of interest. The reliance on physics-based models distinguishes data assimilation from traditional machine learning methodologies, which aim to learn the quantities of interest through purely data-based approaches. From the perspective of machine learning, data assimilation is a learning problem where the quantity of interest is constrained by prior physical assumptions, as captured by the model, and nudged toward the optimum solution by small amounts of data from imperfect observations. Therefore, data assimilation can be considered a form of physics-constrained machine learning [3, 4]. This work improves data assimilation methodologies by combining a mathematically rigorous data assimilation

approach and a data rigorous machine learning algorithm through powerful techniques in multilevel inference [5, 6].

The ensemble Kalman filter [7–9] (EnKF) is a family of computational methods that tackle the data assimilation problem using Gaussian assumptions, and a Monte Carlo approach where the underlying probability densities are represented by ensemble of model state realizations. The ensemble size, i.e., the number of physics-based model runs, is typically the main factor that limits the efficiency of EnKF. For increasing the quality of the results when ensembles are small, heuristics correction methods such as covariance shrinkage [10–12] and localization [13–15] have been developed. As some form of heuristic correction is required for operation implementations of the ensemble Kalman filter, reducing the need for such heuristic corrections in operational implementations is an important and active area of research.

The dominant cost in operational implementations of EnKF is the large number of expensive high fidelity physics-based model runs, which we refer to as "full order models" (FOMs). A natural approach to increase efficiency is to endow the data assimilation algorithms with the ability to use inexpensive, but less accurate, model runs [16, 17], which we refer to as "reduced order models" (ROMs). ROMs are constructed to capture the most important aspects of the dynamics of the FOM, at a fraction of the computational cost; typically they use a much smaller number of variables than the corresponding FOM. The idea of leveraging model hierarchies in numerical algorithms for uncertainty quantification [18] and inference [19–22] is fast gaining traction in both the data assimilation and machine learning communities. Here we focus on two particular types of ROMs: a proper orthogonal decomposition (POD) based ROM, corresponding to a linear projection of the FOM dynamics onto a small linear subspace [23], and a ROM based on autoencoders [24], corresponding to a non-linear projection of the dynamics onto a small dimensional manifold.

The multifidelity ensemble Kalman filter (MFEnKF) developed by the authors [25, 26] combines the ensemble Kalman filter with the idea of surrogate modeling. The MFEnKF optimally combines the information obtained from both the full-order and reduced order surrogate model runs with information begotten from the observations. By posing the data assimilation problem in terms of a mathematically rigorous variance reduction technique—the linear control variate framework—MFEnKF is able to provide robust guarantees about the accuracy of the inference results.

While numerical weather prediction is the dominant driver of innovation in data assimilation literature [27], other applications can benefit from our multifidelity approach such as mechanical engineering [28–30] and air quality modeling [31–33].

The novel elements of this work are: (i) identifying a useful property of autoencoders, namely right invertibility, that aides in the construction of reduced order models, and (ii) deriving a theory for an extension to the MFEnKF through non-linear interpolation and projection techniques; we call the resulting approach nonlinear MFEnKF (NL-MFEnKF). The right-invertibility property ensures the consistency of the reduced state representation through successive applications of the projection and interpolation operators. Our proposed NL-MFEnKF technique shows an advantage on certain regimes of a difficult-to-reduce problem, the Lorenz'96 equations, and shows promise on a large-scale fluids problem, the quasi-geostrophic equations.

This paper is organized as follows. Section 2 discusses the data assimilation problem, provides background on control variates, the EnKF, and the MFEnKF, as well as ROMs and autoencoders. Section 3 introduces NL-MFEnKF, the non-linear extension to the MFEnKF. Section 4 presents the Lorenz'96 and quasi-geostrophic models and their corresponding POD-ROMs. Section 5 introduces the physics-informed autoencoder and practical methods of how to train them and pick optimal hyperparameters. Section 6 provides the results of numerical experiments. Concluding remarks are made in Section 7.

2. BACKGROUND

Sequential data assimilation propagates imperfect knowledge about some physical quantity of interest through an imperfect model of a time-evolving physical system, typically with chaotic dynamics [34]. Without an additional influx of information about reality, our knowledge about the systems rapidly degrades, in the sense of representing the real system less and less accurately. Data assimilation uses noisy external information to enhance our knowledge about the system at hand.

Formally, consider a physical system of interest whose true state at time t_i is X_i^t. The time evolution of the physical system is approximated by the dynamical model

$$X_i = \mathcal{M}_{t_{i-1},t_i}(X_{i-1}) + \Xi_i, \tag{1}$$

where X_i is a random variable whose distribution describes our knowledge of the state of a physical process at time index i, and Ξ_i is a random variable describing the modeling error. In this paper we assume a perfect model ($\Xi_i \equiv 0$), as the discussion of model error in multifidelity methods is significantly outside the scope of this paper.

Additional independent information about the system is obtained through imperfect physical measurements of the observable aspects Y_i of the truth X_i^t, i.e., through noisy observations

$$Y_i = \mathcal{H}(X_i^t) + \eta_i, \quad \eta_i \sim \mathcal{N}(0, \Sigma_{\eta_i,\eta_i}), \tag{2}$$

where the "observation operator" $\mathcal{H}$ maps the model space onto the observation space (i.e., selects the observable aspects of the state).

Our aim is to combine the two sources of information in a Bayesian framework:

$$\pi(X_i \mid Y_i) \propto \pi(Y_i \mid X_i)\,\pi(X_i), \tag{3}$$

where the density $\pi(X_i)$ represents all our prior knowledge, $\pi(Y_i \mid X_i)$ represents the likelihood of the observations given said knowledge, and $\pi(X_i \mid Y_i)$ represents our posterior knowledge.

In the remainder of the paper we use the following notation. Let W and V be random variables. The exact mean of W is

denoted by μ_W, and the exact covariance between W and V by $\Sigma_{W,V}$. E_W denotes an ensemble of samples of W, and $\widetilde{\mu}_W$ and $\widetilde{\Sigma}_{W,V}$ are the empirical ensemble mean of W and empirical ensemble covariance of W and V, respectively.

2.1. Linear Control Variates

Bayesian inference requires that all available information is used in order to obtain correct results [35]. Variance reduction techniques [36] are methods that provide estimates of some quantity of interest with lower variability. From a Bayesian perspective they represent a reintroduction of information that was previously ignored. The linear control variate (LCV) approach [37] is a variance reduction method that aims to incorporate the new information in an optimal linear sense.

LCV works with two vector spaces, a principal space $\mathbb{X}$ and a control space $\mathbb{U}$, and several random variables, as follows. The principal variate X is a $\mathbb{X}$-valued random variable, and the control variate $\widehat{U}$ is a highly correlated $\mathbb{U}$-valued random variable. The ancillary variate U is a $\mathbb{U}$-valued random variable, which is uncorrelated with the preceding two variables, but shares the same mean as $\widehat{U}$, meaning $\mu_U = \mu_{\widehat{U}}$. The linear control variate framework builds a new $\mathbb{X}$-valued random variable Z, called the total variate:

$$Z = X - S(\widehat{U} - U), \tag{4}$$

where the linear "gain" operator S is used to minimize the variance in Z by utilizing the information about the distributions of the constituent variates X, $\widehat{U}$ and U.

In this work $\mathbb{X}$ and $\mathbb{U}$ are finite dimensional vector spaces. The dimension of $\mathbb{X}$ is taken to be n. The dimension of $\mathbb{U}$ we denote by r when it is the reduced order model state space. When $\mathbb{U}$ is the observation space, its dimension is denoted by m.

The following lemma is a slight generalization of [[37], Appendix].

Lemma 1 (Optimal gain)**.** *The optimal gain S that minimizes the trace of the covariance of Z (4) is:*

$$S = \Sigma_{X,\widehat{U}}(\Sigma_{\widehat{U},\widehat{U}} + \Sigma_{U,U})^{-1}. \tag{5}$$

Proof: Observe that,

$$\begin{aligned} \frac{\mathrm{d}\,\mathrm{tr}(\Sigma_{Z,Z})}{\mathrm{d}S} &= 2S(\Sigma_{\widehat{U},\widehat{U}} + \Sigma_{U,U}) - 2\Sigma_{X,\widehat{U}}, \\ \frac{\mathrm{d}^2\,\mathrm{tr}(\Sigma_{Z,Z})}{\mathrm{d}S^2} &= 2(\Sigma_{\widehat{U},\widehat{U}} + \Sigma_{U,U}) \otimes I > 0, \end{aligned} \tag{6}$$

meaning that the problem of finding the optimal gain is convex, and the minimum is unique and is defined by setting the first order optimality condition to zero,

$$2S(\Sigma_{\widehat{U},\widehat{U}} + \Sigma_{U,U}) - 2\Sigma_{X,\widehat{U}} = 0, \tag{7}$$

to which the solution is given by (5) as required.

We first discuss the case where the principal and control variates are related by linear projection and interpolation operators,

$$\widehat{U} = \Theta\, X, \qquad X \approx \widetilde{X} = \Phi\, \widehat{U}, \tag{8}$$

where Θ is the projection operator, Φ is the interpolation operator, and $\widetilde{X}$ is the reconstruction of X.

We reproduce below the useful result [25, Theorem 3.1].

Theorem 1. *Under the assumptions that $\widehat{U}$ and U have equal covariances, and that the principal variate residual is uncorrelated with the control variate, the optimal gain of (4) is half the interpolation operator:*

$$\Sigma_{\widehat{U},\widehat{U}} = \Sigma_{U,U} \quad \text{and} \quad \Sigma_{(X-\Phi\widehat{U}),\widehat{U}} = 0 \quad \Rightarrow \quad S = \frac{1}{2}\,\Phi. \tag{9}$$

Under the assumptions of Theorem 1 the control variate structure (4) is:

$$Z = X - \frac{1}{2}\,\Phi\,(\widehat{U} - U). \tag{10}$$

Remark 1. *Note that Theorem 1 does not require that any random variables are Gaussian. The above linear operator S remains optimal even for non-Gaussian random variables.*

2.2. Linear Control Variates With Non-linear Transformations

While working with linear transformations is elegant, most practical applications require reducing the variance of a non-linear transformation of a random variable. We now generalize the control variate framework to address this case.

Following [36], assume that our transformed principal variate is of the form $h(X)$, where h is some arbitrary smooth non-linear operator:

$$h : \mathbb{X} \to h(\mathbb{X}), \tag{11}$$

We also assume that the transformed control variate and the transformed ancillary variate are of the form $g(\widehat{U})$ and $g(U)$, respectively, where g is also some arbitrary smooth non-linear operator:

$$g : \mathbb{U} \to g(\mathbb{U}), \tag{12}$$

We define the total variate Z^h in the space $\mathbb{H}$ such that $h(\mathbb{X}) \subset \mathbb{H}$ by:

$$Z^h = h(X) - S\left(g(\widehat{U}) - g(U)\right), \tag{13}$$

with the optimal linear gain given by Lemma 1:

$$S = \Sigma_{h(X),g(\widehat{U})}\left(\Sigma_{g(\widehat{U}),g(\widehat{U})} + \Sigma_{g(U),g(U)}\right)^{-1}. \tag{14}$$

Theorem 2. *If $\widehat{U}$ and U independently and identically distributed, and share the same mean and covariance, the control variate structure (13) holds, and the optimal linear gain is:*

$$S = \frac{1}{2}\,\Sigma_{h(X),g(\widehat{U})}\,\Sigma^{-1}_{g(\widehat{U}),g(\widehat{U})}. \tag{15}$$

Proof: As $\widehat{U}$ and U are independently and identically distributed, they share the same mean and covariance under non-linear transformation,

$$\mu_{g(\widehat{U})} = \mu_{g(U)}, \quad \text{and} \quad \Sigma_{g(\widehat{U}),g(\widehat{U})} = \Sigma_{g(U),g(U)}. \tag{16}$$

In this case $g(\widehat{U}) - g(U)$ is unbiased (mean zero), and the control variate framework (14) can be applied. Thus, the optimal gain is given by (15) as required.

We now consider the case where the transformations of the principal variate and control variates represent approximately the same information,

$$h(X) \approx g(\widehat{U}), \tag{17}$$

and exist in the same space $\mathbb{H}$.

We now provide a slight generalization of Theorem 1 under non-linear transformation assumptions.

Theorem 3. *If the assumption of Theorem 2 hold, and the transformed principal variate residual is uncorrelated with the transformed control variate,*

$$\Sigma_{(h(X)-g(\widehat{U})),g(\widehat{U})} \overset{\text{assumed}}{=} 0, \tag{18}$$

then the optimal gain is

$$S = \frac{1}{2} I, \tag{19}$$

where I is the identity operator.

Proof: By simple manipulation of (15), we obtain:

$$\begin{aligned} S &= \frac{1}{2}\, \Sigma_{h(X),g(\widehat{U})}\, \Sigma^{-1}_{g(\widehat{U}),g(\widehat{U})}, \\ &= \frac{1}{2}\, \Sigma_{(h(X)-g(\widehat{U})+g(\widehat{U})),g(\widehat{U})}\, \Sigma^{-1}_{g(\widehat{U}),g(\widehat{U})}, \\ &= \frac{1}{2}\, \Sigma_{(h(X)-g(\widehat{U})),g(\widehat{U})}\, \Sigma^{-1}_{g(\widehat{U}),g(\widehat{U})} + \frac{1}{2}\, \Sigma_{g(\widehat{U}),g(\widehat{U})}\, \Sigma^{-1}_{g(\widehat{U}),g(\widehat{U})}, \\ &= \frac{1}{2} I. \end{aligned}$$

2.3. The Ensemble Kalman Filter

The EnKF is a statistical approximation to the optimal control variate structure (4), where the underlying probability density functions are represented by empirical measures using ensembles, i.e., a finite number of realizations of the random variables. The linear control variate framework allows to combine multiple ensembles into one that better represents the desired quantity of interest.

Let $\mathsf{E}_{X_i^b} \in \mathbb{R}^{n\times N_X}$ be an ensemble of N_X realizations of the n-dimensional principal variate, which represents our prior uncertainty in the model state at time index i from (1). Likewise, let $\mathsf{E}_{\mathcal{H}_i(X_i^b)} = \mathcal{H}_i(\mathsf{E}_{X_i^b}) \in \mathbb{R}^{m\times N_X}$ be an ensemble of N_X realizations of the m-dimensional control observation state variate, which represents the same model realizations cast into observation space. Let $\mathsf{E}_{Y_i} \in \mathbb{R}^{m\times N_X}$ be an ensemble of N_X "perturbed observations," which is a statistical correction required in the ensemble Kalman filter [9].

Remark 2 (EnKF Perturbed Observations)**.** *Each ensemble member of the perturbed observations is sampled from a Gaussian distribution with mean the measured value, and the known observation covariance from (2):*

$$[\mathsf{E}_{Y_i}]_{:,e} \sim \mathcal{N}(\mu_{Y_i}, \Sigma_{\eta_i,\eta_i}). \tag{20}$$

The prior ensemble at time step i is obtained by propagating the posterior ensemble at time $i-1$ through the model equations,

$$\mathsf{E}_{X_i^b} = \mathcal{M}_{t_{i-1},t_i}\left(\mathsf{E}_{X_{i-1}^a}\right), \tag{21}$$

where the slight abuse of notation indicates an independent model propagation of each ensemble member. Application of the Kalman filter formula constructs an ensemble $\mathsf{E}_{X_i^a}$ describing the posterior uncertainty:

$$\mathsf{E}_{X_i^a} = \mathsf{E}_{X_i^b} - \widetilde{K}_i\left(\mathsf{E}_{\mathcal{H}_i(X_i^b)} - \mathsf{E}_{Y_i}\right), \tag{22}$$

where the statistical Kalman gain is an ensemble-based approximation to the exact gain in Lemma 1:

$$\widetilde{K}_i = \widetilde{\Sigma}_{X_i^b,\mathcal{H}_i(X_i^b)}\left(\widetilde{\Sigma}_{\mathcal{H}_i(X_i^b),\mathcal{H}_i(X_i^b)} + \Sigma_{\eta_i,\eta_i}\right)^{-1}. \tag{23}$$

Remark 3 (Inflation)**.** *Inflation is a probabilistic correction necessary to account for the Kalman gain being correlated to the ensemble [38]. In inflation the ensemble anomalies (deviations from the statistical mean) are multiplied by a constant $\alpha > 1$, thereby increasing the covariance of the distribution described by the ensemble:*

$$\mathsf{E}_{X_{i+1}^b} \leftarrow \widetilde{\mu}_{X_{i+1}^b} + \alpha\left(\mathsf{E}_{X_{i+1}^b} - \widetilde{\mu}_{X_{i+1}^b}\right). \tag{24}$$

2.4. The Multifidelity Ensemble Kalman Filter

In this section, we present the standard Multifidelity Ensemble Kalman Filter (MFEnKF) with linear assumptions on the model, projection, and observation operators, and Gaussian assumptions on all probability distributions.

The MFEnKF [25] merges the information from a hierarchy of models and the corresponding observations into a coherent representation of the uncertain model state. To propagate this representation forward in time during the forecast phase, it is necessary that the models are decoupled, but implicitly preserve some underlying structure of the error information. We make use of the linear control variate structure to combine this information in an optimal manner.

Without loss of generality we discuss here a bi-fidelity approach, where one full-order model is coupled to a lower-fidelity reduced-order model. A telescopic extension to multiple fidelities is provided at the end of the section. Instead of having access to one model $\mathcal{M}$, assume that we have access to a hierarchy of models. In the bi-fidelity case, the principal space model (FOM) is denoted by $\mathcal{M}^X$ and the control space model (ROM) is denoted by $\mathcal{M}^U$.

We now consider the total variate

$$Z_i^{\mathrm{b}} = X_i^{\mathrm{b}} - \frac{1}{2}\,\Phi\,(\widehat{U}_i^{\mathrm{b}} - U_i^{\mathrm{b}}), \tag{25}$$

that describes the prior total information from a model that evolves in principal space (X_i^{b}) and a model that evolves in ancillary space ($\widehat{U}_i^{\mathrm{b}}$ and U_i^{b}).

Assume that our prior total variate is represented by the three ensembles $\mathsf{E}_{X_i^{\mathrm{b}}} \in \mathbb{R}^{n\times N_X}$ consisting of N_X realizations of the n-dimensional principal model state variate, $\mathsf{E}_{\widehat{U}_i^{\mathrm{b}}} \in \mathbb{R}^{r\times N_X}$ consisting of N_X realizations of the r-dimensional control model state variate, and $\mathsf{E}_{U_i^{\mathrm{b}}} \in \mathbb{R}^{r\times N_U}$ consisting of N_U realizations of the r-dimensional ancillary model state variate. Each of these ensembles has a corresponding ensemble of m-dimensional control observation space realizations.

MFEnKF performs sequential data assimilation using the above constituent ensembles, without having to explicitly calculate the ensemble of the total variates. The MFEnKF forecast step propagates the three ensembles form the previous step:

$$\begin{aligned}\mathsf{E}_{X_i^{\mathrm{b}}} &= \mathcal{M}^X_{t_{i-1},t_i}(\mathsf{E}_{X_{i-1}^{\mathrm{a}}}),\\ \mathsf{E}_{\widehat{U}_i^{\mathrm{b}}} &= \mathcal{M}^U_{t_{i-1},t_i}(\mathsf{E}_{\widehat{U}_{i-1}^{\mathrm{a}}}),\\ \mathsf{E}_{U_i^{\mathrm{b}}} &= \mathcal{M}^U_{t_{i-1},t_i}(\mathsf{E}_{U_{i-1}^{\mathrm{a}}}).\end{aligned} \tag{26}$$

Two observation operators $\mathcal{H}_i^X$ and $\mathcal{H}_i^U$ cast the principal model and control model spaces, respectively, into the control observation space. In this paper we assume that he principal model space observation operator is the canonical observation operator (2):

$$\mathcal{H}_i^X(X_i) := \mathcal{H}_i(X_i), \tag{27}$$

and that the control model space observation operator is the canonical observation operator (2) applied to the linear interpolated reconstruction (42) of a variable in control model space:

$$\mathcal{H}_i^U(U_i) := \mathcal{H}_i(\Phi U_i). \tag{28}$$

Additionally, we define an (approximate) observation operator for the total model variate :

$$\mathcal{H}_i^Z(Z_i) := \mathcal{H}_i^X(X_i) - \tfrac{1}{2}\left(\mathcal{H}_i^U(\widehat{U}_i) - \mathcal{H}_i^U(U_i)\right), \tag{29}$$

which, under the linearity assumptions on $\mathcal{H}_i^X$ of Theorem 3 and the underlying Gaussian assumptions on $\widehat{U}_i$ and U_i of Theorem 2, begets that $\mathcal{H}_i^Z = \mathcal{H}_i^X$. Even without the linearity assumption the definition (29) is operationally useful.

The MFEnKF analysis updates each constituent ensemble as follows:

$$\begin{aligned}\mathsf{E}_{X_i^{\mathrm{a}}} &= \mathsf{E}_{X_i^{\mathrm{b}}} - \widetilde{\mathbf{K}}_i\left(\mathsf{E}_{\mathcal{H}_i^X(X_i^{\mathrm{b}})} - \mathsf{E}_{Y_i^X}\right),\\ \mathsf{E}_{\widehat{U}_i^{\mathrm{a}}} &= \mathsf{E}_{\widehat{U}_i^{\mathrm{b}}} - \Theta\widetilde{\mathbf{K}}_i\left((\mathsf{E}_{\mathcal{H}_i^U(\widehat{U}_i^{\mathrm{b}})} - \mathsf{E}_{Y_i^X}\right),\\ \mathsf{E}_{U_i^{\mathrm{a}}} &= \mathsf{E}_{U_i^{\mathrm{b}}} - \Theta\widetilde{\mathbf{K}}_i\left((\mathsf{E}_{\mathcal{H}_i^U(U_i^{\mathrm{b}})} - \mathsf{E}_{Y_i^U}\right),\end{aligned} \tag{30}$$

with the heuristic correction to the means

$$\widetilde{\mu}_{X_i^{\mathrm{a}}} \leftarrow \widetilde{\mu}_{Z_i^a}, \quad \widetilde{\mu}_{\widehat{U}_i^{\mathrm{a}}} \leftarrow \Theta\widetilde{\mu}_{Z_i^a}, \quad \widetilde{\mu}_{U_i^{\mathrm{a}}} \leftarrow \Theta\widetilde{\mu}_{Z_i^a}, \tag{31}$$

applied in order to fulfill the unbiasedness requirement of the control variate structure:

$$\widetilde{\mu}_{Z_i^{\mathrm{a}}} = \widetilde{\mu}_{Z_i^{\mathrm{b}}} - \widetilde{\mathbf{K}}_i\left(\widetilde{\mu}_{\mathcal{H}_i^X(Z_i^{\mathrm{b}})} - \mu_{Y_i}\right). \tag{32}$$

The Kalman gain and the covariances are defined by the semi–linearization:

$$\widetilde{\mathbf{K}}_i = \widetilde{\Sigma}_{Z_i^{\mathrm{b}},\mathcal{H}_i^Z(Z_i^{\mathrm{b}})}\left(\widetilde{\Sigma}_{\mathcal{H}_i^Z(Z_i^{\mathrm{b}}),\mathcal{H}_i^Z(Z_i^{\mathrm{b}})} + \Sigma_{\eta_i,\eta_i}\right)^{-1} \tag{33}$$

$$\begin{aligned}\widetilde{\Sigma}_{Z_i^{\mathrm{b}},\mathcal{H}_i^Z(Z_i^{\mathrm{b}})} = \widetilde{\Sigma}_{X_i^{\mathrm{b}},\mathcal{H}_i^X(X_i^{\mathrm{b}})} &+ \tfrac{1}{4}\widetilde{\Sigma}_{\Phi\widehat{U}_i^{\mathrm{b}},\mathcal{H}_i^U(\widehat{U}_i^{\mathrm{b}})} + \tfrac{1}{4}\widetilde{\Sigma}_{\Phi U_i^{\mathrm{b}},\mathcal{H}_i^U(U_i^{\mathrm{b}})}\\ &- \tfrac{1}{2}\widetilde{\Sigma}_{X_i^{\mathrm{b}},\mathcal{H}_i^U(\widehat{U}_i^{\mathrm{b}})} - \tfrac{1}{2}\widetilde{\Sigma}_{\Phi\widehat{U}_i^{\mathrm{b}},\mathcal{H}_i^X(X_i^{\mathrm{b}})},\end{aligned} \tag{34}$$

$$\begin{aligned}\widetilde{\Sigma}_{\mathcal{H}_i^Z(Z_i^{\mathrm{b}}),\mathcal{H}_i^Z(Z_i^{\mathrm{b}})} = \widetilde{\Sigma}_{\mathcal{H}_i^X(X_i^{\mathrm{b}}),\mathcal{H}_i(X_i^{\mathrm{b}})} &+ \tfrac{1}{4}\widetilde{\Sigma}_{\mathcal{H}_i^U(\widehat{U}_i^{\mathrm{b}}),\mathcal{H}_i^U(\widehat{U}_i^{\mathrm{b}})} + \tfrac{1}{4}\widetilde{\Sigma}_{\mathcal{H}_i^U(U_i^{\mathrm{b}}),\mathcal{H}_i^U(U_i^{\mathrm{b}})}\\ &- \tfrac{1}{2}\widetilde{\Sigma}_{\mathcal{H}_i^X(X_i^{\mathrm{b}}),\mathcal{H}_i^U(\widehat{U}_i^{\mathrm{b}})} - \tfrac{1}{2}\widetilde{\Sigma}_{\mathcal{H}_i^U(\widehat{U}_i^{\mathrm{b}}),\mathcal{H}_i^X(X_i^{\mathrm{b}})}.\end{aligned} \tag{35}$$

In order to ensure that the control variate $\widehat{U}$ remains highly correlated to the principal variate X, at the end of each analysis step we replace the analysis control variate ensemble with the corresponding projection of the principal variate ensemble:

$$\mathsf{E}_{\widehat{U}_i^{\mathrm{a}}} \leftarrow \Theta\,\mathsf{E}_{X_i^{\mathrm{a}}}. \tag{36}$$

Some important properties of MFEnKF are:

- MFEnKF makes use of surrogate models to reduce the uncertainty in the full state.
- MFEnKF does not explicitly construct the total variates, and instead performs the assimilation on the constituent ensembles.
- Under Gaussian and linear assumptions, the sample mean of the MFEnKF is an unbiased estimate of the truth.

Remark 4 (MFEnKF Perturbed observations)**.** *There is no unique way to perform perturbed observations (remark 2) in the MFEnKF. We will present one way in this paper. As Theorem 2 requires both the control and ancillary variates to share the same covariance, we utilize here the 'control space uncertainty consistency' approach. The perturbed observations ensembles in (30) is defined by:*

$$[\mathsf{E}_{Y_i^X}]_{:,e} \sim \mathcal{N}(\mu_{Y_i}, \Sigma_{\eta_i,\eta_i}), \tag{37}$$

$$[\mathsf{E}_{Y_i^U}]_{:,e} \sim \mathcal{N}(\mu_{Y_i}, s\Sigma_{\eta_i,\eta_i}), \tag{38}$$

where the scaling factor is $s = 1$. See [25, Section 4.2] for a more detailed discussion about perturbed observations.

Remark 5 (MFEnKF Inflation)**.** *Similarly to the EnKF (see Remark 3), the MFEnKF also requires inflation in order to account for the statistical Kalman gain being correlated to its constituent ensembles. For a bi-fidelity MFEnKF, two inflation factors are required: α_X which acts on the anomalies of the principal and control variates (as they must remain highly correlated) and α_U which acts on the ensemble anomalies of the ancillary variate:*

$$\begin{aligned}\mathsf{E}_{X_{i+1}^b} &\leftarrow \widetilde{\mu}_{X_{i+1}^b} + \alpha_X\left(\mathsf{E}_{X_{i+1}^b} - \widetilde{\mu}_{X_{i+1}^b}\right),\\ \mathsf{E}_{\widehat{U}_{i+1}^b} &\leftarrow \widetilde{\mu}_{\widehat{U}_{i+1}^b} + \alpha_X\left(\mathsf{E}_{\widehat{U}_{i+1}^b} - \widetilde{\mu}_{\widehat{U}_{i+1}^b}\right),\\ \mathsf{E}_{U_{i+1}^b} &\leftarrow \widetilde{\mu}_{U_{i+1}^b} + \alpha_U\left(\mathsf{E}_{U_{i+1}^b} - \widetilde{\mu}_{U_{i+1}^b}\right).\end{aligned} \tag{39}$$

Remark 6 (Deterministic EnKF flavors)**.** *Many deterministic flavors of the EnKF [2] are extendable to the MFEnKF. The DEnKF [39] in particular is trivially extendanble to the non-linear multifidelity approach identified in this work. It has been the authors' experience, however that the perturbed observation flavor of the EnKF is more robust in the multifidelity setting. The authors' suspect that this is the case precisely because of its stochastic nature, leading it to better account for model error-based inaccuracies in the surrogate models. Accounting for this type of model error is outside the scope of this work.*

Remark 7 (Cost of the MFEnKF)**.** *It is known from [25] that, given a full order model with cost C_X with N_X ensemble members and a reduced order model with cost C_U and N_U ensemble members, then the MFEnKF is more effective than a normal EnKF with N ensemble members whenever,*

$$C_U \leq \frac{C_X(N - N_X)}{N_X + N_U}, \tag{40}$$

meaning that the optimal cost of the reduced order model is highly dependent on the desired full order ensemble size.

2.5. Autoencoders

We now generalize from the linear interpolation and projection assumed previously (8), and consider a class of non-linear projection and interpolation operators.

An autoencoder [24] is an artificial neural network consisting of two *smooth* components, an encoder θ and a decoder ϕ, such that given a variable X in the principal space, the variable

$$\widehat{U} = \theta(X), \tag{41}$$

resides in the control space of the encoder. Conversely the reconstruction,

$$X \approx \widetilde{X} = \phi(\widehat{U}), \tag{42}$$

is an approximation to X in the principal space, and which in some optimal sense approximately recovers the information embedded in X. While the relative dimension n of the principal space is relatively high, the arbitrary structure of an artificial neural networks allows the autoencoder to learn the optimal r-dimensional (small) representation of the data.

2.6. Non-linear Projection-Based Reduced Order Models

The important information of many dynamical systems can be expressed with significantly fewer dimensions than the discretization dimension n [40]. For many infinite dimensional equations it is possible to construct a finite-dimensional inertial manifold that represents the dynamics of the system (including the global attractor). The Hausdorff dimension of the global attractor of some dynamical system is a useful lower bound for the minimal representation of the dynamics, though a representation of just the attractor is likely not sufficient to fully capture all the "useful" aspects of the data. For data-based reduced order models an important aspect is the intrinsic dimension [41] of the data. The authors are not aware of any formal statements relating the dimension of an inertial manifold and the intrinsic dimension of some finite discretization of the dynamics. We assume that reduced dimension r is sufficient to represent either the dynamics or the data, or both, and allows to build a "useful" surrogate model.

We will now discuss the construction of reduced order models for problems posed as ordinary differential equations. The following derivations are similar to those found in [42], but assume vector spaces and no re-centering.

Just like in the control variate framework in Section 2.1, the full order model resides in the principal space $\mathbb{X} \subset \mathbb{R}^n$ and the reduced order model is defined in the space $\mathbb{U} \subset \mathbb{R}^r$, which is related to $\mathbb{X}$ through the smooth non-linear projection (41).

Given an initial value problem in $\mathbb{X}$:

$$\frac{\mathrm{d}X}{\mathrm{d}t} = f(X), \quad X(t_0) = X_0, \quad t \in [t_0, t_f], \tag{43}$$

and the projection operator (41), the induced reduced order model initial value problem in $\mathbb{U}$ is defined by simple differentiation of $U = \theta(X)$, by dynamics in the space $\mathbb{U}$,

$$\frac{\mathrm{d}U}{\mathrm{d}t} = \theta'(X)f(X), \quad X(t_0) = X_0, \quad t \in [t_0, t_f]. \tag{44}$$

As is common, the full order trajectory is not available during integration, as there is no bijection from $\mathbb{X}$ to $\mathbb{U}$, thus an approximation using the interpolation operator (42) that fully resides in $\mathbb{U}$ is used instead:

$$\frac{\mathrm{d}U}{\mathrm{d}t} = \theta'(\phi(U))f(\phi(U)), \quad U(t_0) = \theta(X_0), \quad t \in [t_0, t_f]. \tag{45}$$

Note that this is not the only way to obtain a reduced order model by using arbitrary projection and interpolation operators. It is however the simplest extension of the POD-based ROM framework.

Remark 8 (Linear ROM)**.** *Common methods for finding projection and interpolation operators make a linear assumption (methods such as POD), thus, in the linear case (8) the reduced order model (45) takes the form*

$$\frac{dU}{dt} = \Theta f(\Phi U), \quad U(t_0) = \Theta X_0, \quad t \in [t_0, t_f]. \tag{46}$$

3. NON-LINEAR PROJECTION-BASED MFENKF

We extend MFEnKF to work with non-linear projection and interpolation operators. The new algorithm is named NL-MFEnKF. Since existing theoretical extensions of the linear control variate framework to the non-linear case [43] are not completely satisfactory, violating the assumption of an unbiased estimate of the total variate, we resort to several heuristic assumptions to construct this algorithm. Heuristic approaches that work well in practice are widely used in data assimilation literature [2].

The main idea is to replace the optimal control variate structure for linear projection and interpolation operators (10) with one that works with their non-linear counterparts (13):

$$Z_i^b = X_i^b - \frac{1}{2}\left(\phi(\widehat{U}_i^b) - \phi(U_i^b)\right). \tag{47}$$

We assume that $\widehat{U}$ and U are independently and identically distributed, such that they obey the assumptions made in Theorem 2 and in Theorem 3 for the optimal gain.

Similar to MFEnKF (28), the control model space observation operator is the application of the canonical observation operator (2) to the reconstruction

$$\mathcal{H}_i^U(U_i) := \mathcal{H}_i(\phi(U_i)), \tag{48}$$

with the other observation operators Equations (27, 29) defined as in the MFEnKF.

Remark 9. *It is of independent interest to explore control model space observation operators that are not of the form (48). For example, if the interpolation operator ϕ is created through an autoencoder, the control model space observation operator $\mathcal{H}^U$ could similarly be a different decoder of the same latent space.*

The MFEnKF equations (30) are replaced by their non-linear counterparts in a manner similar to what is done with non-linear observation operators,

$$\begin{aligned}
\mathsf{E}_{X_i^a} &= \mathsf{E}_{X_i^b} - \widetilde{\mathbf{K}}_i\left(\mathsf{E}_{\mathcal{H}_i^X(X_i^b)} - \mathsf{E}_{Y_i^X}\right),\\
\mathsf{E}_{\widehat{U}_i^a} &= \mathsf{E}_{\widehat{U}_i^b} - \widetilde{\mathbf{K}}_i^\theta\left(\mathsf{E}_{\mathcal{H}_i^U(\widehat{U}_i^b)} - \mathsf{E}_{Y_i^X}\right),\\
\mathsf{E}_{U_i^a} &= \mathsf{E}_{U_i^b} - \widetilde{\mathbf{K}}_i^\theta\left(\mathsf{E}_{\mathcal{H}_i^U(U_i^b)} - \mathsf{E}_{Y_i^U}\right),
\end{aligned} \tag{49}$$

where, as opposed to (30), there are now two Kalman gains, defined by:

$$\widetilde{\boldsymbol{K}}_i = \widetilde{\Sigma}_{Z_i^b,\mathcal{H}_i^Z(Z_i^b)}\left(\widetilde{\Sigma}_{\mathcal{H}_i^Z(Z_i^b),\mathcal{H}_i^Z(Z_i^b)} + \Sigma_{\eta_i,\eta_i}\right)^{-1}, \tag{50}$$

$$\widetilde{\boldsymbol{K}}_i^\theta = \widetilde{\Sigma}_{\theta(Z_i^b),\mathcal{H}_i^Z(Z_i^b)}\left(\widetilde{\Sigma}_{\mathcal{H}_i^Z(Z_i^b),\mathcal{H}_i^Z(Z_i^b)} + \Sigma_{\eta_i,\eta_i}\right)^{-1}. \tag{51}$$

Here we take a heuristic approach and use semi-linear approximations of the covariances, similar to (34) and (35). The perturbed observations are defined like in MFEnKF (Remark 4).

Figure 1 provides a visual diagram of both the forecast and analysis steps of the NL-MFEnKF algorithm.

Remark 10 (Localization for the NL-MFEnKF)**.** *In operational data assimilation workflows, localization [2] is an important heuristic for the viability of the family of ensemble Kalman filter algorithms. While it is trivial to apply many B-localization techniques to the full-space Kalman gain (50), it is not readily apparent how one may attempt to do so for the reduced-space Kalman gain (51). Convolutional autoencoders [24] might provide an avenue for such a method, as they attempt to preserve some of the underlying spatial structure of the full space in the reduced space. An alternative is the use of R-localization, though, in the authors' view, there is a non-trivial amount of work to be done in order to formulate such a method.*

3.1. NL-MFEnKF Heuristic Corrections

For linear operators the projection of the mean is the mean of the projection. This is however not true for general non-linear operators. Thus, in order to correct the means like in the MFEnKF (31), additional assumptions have to be made.

The empirical mean of the total analysis variate (47) [similar to (32)] is

$$\widetilde{\mu}_{Z_i^a} = \widetilde{\mu}_{X_i^a} - \frac{1}{2}\left(\widetilde{\mu}_{\phi(\widehat{U}_i^a)} - \widetilde{\mu}_{\phi(U_i^a)}\right). \tag{52}$$

We use it to find the optimal mean adjustments in reduced space. Specifically, we set the mean of the analysis principal variate to be the mean of the analysis total variate (52),

$$\widetilde{\mu}_{X_i^a} \leftarrow \widetilde{\mu}_{Z_i^a}, \tag{53}$$

enforce the recorrelation of the principal and control variates (36) *via*

$$\mathsf{E}_{\widehat{U}_i^a} \leftarrow \theta(\mathsf{E}_{X_i^a}), \tag{54}$$

and define the control variate mean adjustment as a consequence of the above as,

$$\widetilde{\mu}_{\widehat{U}_i^a} \leftarrow \widetilde{\mu}_{\theta(X_i^a)}. \tag{55}$$

Unlike the linear control variate framework of the MFEnKF (25), the non-linear framework of the NL-MFEnKF (47) does not induce a unique way to impose unbiasedness on the control-space variates. There are multiple possible non-linear formulations to the MFEnKF, and multiple possible heuristic corrections of the mean the ancillary variate. Here we discuss three approaches based on:

1. control space unbiased mean adjustment,
2. principal space unbiased mean adjustment, and
3. Kalman approximate mean adjustment,

each stemming from a different assumption on the relationship between the ancillary variate and the other variates.

3.1.1. Control Space Unbiased Mean Adjustment

The assumption that the control variate $\widehat{U}_i^a$ and the ancillary variate U_i^a are unbiased in the control space implies that they share the same mean. The mean adjustment of $\widehat{U}^a$ in (55) directly defines the mean adjustment of the ancillary variate:

$$\widetilde{\mu}_{U_i^a} \leftarrow \widetilde{\mu}_{\widehat{U}_i^a}. \tag{56}$$

The authors will choose this method of correction in the numerical experiments for both its properties and ease of implementation.

3.1.2. Principal Space Unbiased Mean Adjustment

If instead we assume that the control variate $\phi(\widehat{U}_i)$ and the ancillary variate $\phi(U_i)$ are unbiased in the principal space (meaning that they have the same mean), then the mean of the total variate Z_i (47) equals the mean of the principal variate X_i, a desirable property.

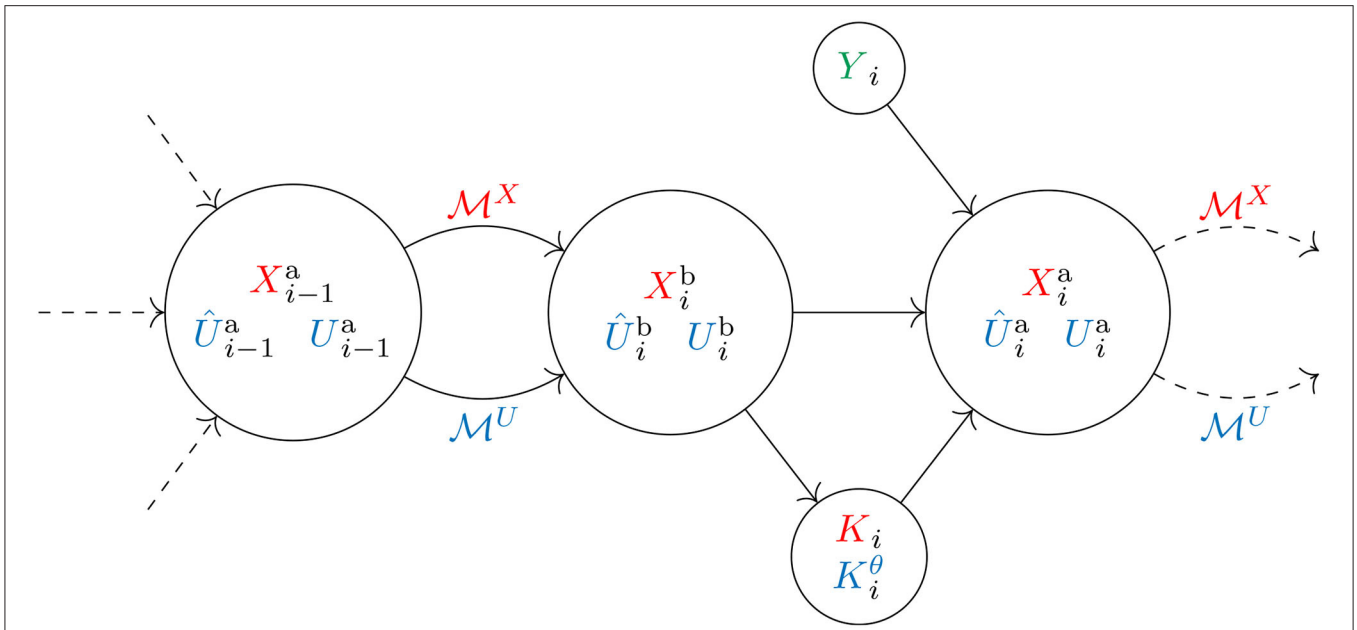


FIGURE 1 | A visual diagram of the NL-MFEnKF algorithm. The random variables are represented by an ensemble of realizations, and the directional acyclic graph provides an explicit (infinite) causal graph.

Finding a mean adjustment for U_i^a in the control space such that the unbiasedness is satisfied in the principal space is a non-trivial problem. Explicitly, we seek a vector ν such that:

$$\widetilde{\mu}_{\phi(\widehat{U}_i^a)} = \widetilde{\mu}_{\phi(U_i^a - \widetilde{\mu}_{U_i^a} + \nu)}, \tag{57}$$

resulting in the correction,

$$\mathsf{E}_{U_i^a} \leftarrow \mathsf{E}_{U_i^a} - \widetilde{\mu}_{U_i^a} + \nu. \tag{58}$$

The solution to (57) requires the solution of an expensive nonlinear equation. Note that (57) is equivalent to (56) under the assumptions of Theorem 2, in the limit of large ensemble size.

3.1.3. Kalman Approximate Mean Adjustment

Instead of assuming that the control and ancillary variates are unbiased, we can consider directly the mean of the control-space total variate:

$$\widetilde{\mu}_{\theta(Z_i^a)} = \widetilde{\mu}_{\theta(X_i^a)} - \frac{1}{2}\left(\widetilde{\mu}_{\widehat{U}_i^a} - \widetilde{\mu}_{U_i^a}\right), \tag{59}$$

defined with the mean values in NL-MFEnKF formulas (49). The following adjustment to the mean of the ancillary variate:

$$\widetilde{\mu}_{U_i^a} \leftarrow \widetilde{\mu}_{\theta(Z_i^a)}, \tag{60}$$

is not unbiased with respect to the control variate in any space, but provides a heuristic approximation of the total variate mean in control space, and does not affect the principal variate mean.

3.2. Telescopic Extension

As in [[25], Section 4.5] one can telescopically extend the bi-fidelity NL-MFEnKF algorithm to a hierarchy of $\mathcal{L}+1$ models of different fidelities. Assume that the nonlinear operator ϕ_ℓ interpolates from the space of fidelity ℓ to the space of fidelity $\ell-1$, where ϕ_1 interpolates to the principal space. A telescopic extension of (47) is

$$Z = X - \sum_{\ell=1}^{\mathcal{L}} 2^{-\ell}\left(\overline{\phi}(\widehat{U}_\ell) - \overline{\phi}(U_\ell)\right), \tag{61}$$

where the projection operator at each fidelity is defined as,

$$\overline{\phi}_\ell = \phi_1 \circ \cdots \circ \phi_\ell, \tag{62}$$

projecting from the space of fidelity ℓ to the principal space. The telescopic extension of the NL-MFEnKF is not analyzed further in this work.

4. DYNAMICAL MODELS AND THE CORRESPONDING POD-ROMS

For numerical experiments we use two dynamical systems: the Lorenz'96 model [44] and the Quasi-Geostrophic equations (QGE) [45–48].

For each of these models we construct two surrogates that approximate their dynamics. The first type of surrogate is a principal orthogonal decomposition-based quadratic reduced order model (POD-ROM), which is the classical approach to building the ROM. The second surrogate is an autoencoder neural network-based reduced order model (NN-ROM).

We will use the Lorenz'96 equations to test the methodology and derive useful intuition about the hyperparameters. For the POD-ROM and NN-ROMs for the Lorenz'96 equations we construct reduced order models (ROMs) for reduced dimension sizes of $r = 7, 14, 21, 28$, and 35.

We will use the Quasi-geostrophic equations to illustrate that our methodology can be applied in an operational setting. For both the POD-ROM and NN-ROMs we will build ROMs of a single reduced dimension size, $r = 25$.

The Lorenz'96, QGE, and corresponding the POD-ROM models are implemented in the ODE-test-problems suite [49, 50].

4.1. Lorenz'96

The Lorenz'96 model [44] can be conjured from the PDE [1, 51],

$$\frac{\mathrm{d}y}{\mathrm{d}t} = -yy_x - y + F, \tag{63}$$

where the forcing parameter is set to $F = 8$. In the semi-discrete version $y \in \mathbb{R}^n$, and the nonlinear term is approximated by a numerically unstable finite difference approximation,

$$\left[yy_x\right]_k = \left(\widehat{I}\,y\right)_k \cdot \left(\widehat{D}\,y\right)_k = \left([y]_{k-1}\right) \cdot \left([y]_{k-2} - [y]_{k+1}\right), \tag{64}$$

where $\widehat{I}$ is a (linear) shift operator, and the linear operator $\widehat{D}$ is a first order approximation to the first spatial derivative. The canonical $n = 40$ variable discretization with cyclic boundary conditions is used. The classical fourth order Runge-Kutta method is used to discretize the time dimension.

For the given discrete formulation of the Lorenz'96 system, 14 represents the number of non-negative Lyapunov exponents, 28 represents the rounded-up Kaplan-Yorke dimension of 27.1, and 35 represents an approximation of the intrinsic dimension of the system [calculated by the method provided by [52]]. To the authors' knowledge, the inertial manifold of the system, if it exists, is not known. The relatively high ratio between the intrinsic dimension of the system and the spatial dimension of the system makes constructing a reduced order model particularly challenging.

4.1.1. Data for Constructing Reduced-Order Models

The data to construct the reduced order models is taken to be 10,000 state snapshots from a representative model run. The snapshots are spaced 36 time units apart, equivalent to 6 months in model time. The first 5,000 snapshots provide the training data, and the next 5,000 are taken as testing data in order to test the extrapolation power of the surrogate models.

4.1.2. Proper Orthogonal Decomposition ROM for Lorenz'96

Using the method of snapshots [53], we construct optimal linear operators, $\Phi^T = \Theta \in \mathbb{R}^{r\times n}$, such that the projection captures the dominant orthogonal modes of the system dynamics. The reduced order model approximation with linear projection and interpolation operators (46) is quadratic [similar to [54, 55]]

$$\frac{\mathrm{d}u}{\mathrm{d}t} = a + Bu + u^T \mathcal{C} u, \tag{65}$$

where the corresponding vector a, matrix B, and 3-tensor $\mathcal{C}$ are defined by:

$$a = F\Theta 1_n, \tag{66a}$$

$$\mathrm{B} = -\Theta\Phi, \tag{66b}$$

$$[\mathcal{C}]_{jk\ell} = -\left(\widehat{I}\Phi_{:,j}\right)^T \left(\widehat{\mathrm{D}}\Phi_{:,k}\right) \Phi_{:,\ell}. \tag{66c}$$

4.2. Quasi-Geostrophic Equations

We will utilize the quasi-geostrophic equations (QGE) [45–48] as a proof-of-concept to showcase the proposed methodology in a more realistic setting. We follow the formulation used in [25, 55, 56],

$$\begin{gathered} \omega_t + J(\psi,\omega) - \mathrm{Ro}^{-1}\psi_x = \mathrm{Re}^{-1}\Delta\omega + \mathrm{Ro}^{-1}F, \\ J(\psi,\omega) \equiv \psi_y\omega_x - \psi_x\omega_y, \quad \omega = -\Delta\psi, \end{gathered} \tag{67}$$

where ω represents the vorticity, ψ is the corresponding streamfunction, Ro is the Rossby number, Re is the Reynolds number, and J represents the quadratic Jacobian term.

4.2.1. Data for Constructing Reduced-Order Models

For the QGE, we collect 10,000 state snapshot points spaced 30 days apart, equivalent to about 0.327 time units in our discretization. As we wish to simulate a realistic online scenario, all data will be used for surrogate model training. The validation of the surrogates will be done through their practical use in the MFEnKF and NL-MFEnKF assimilation frameworks.

4.2.2. Proper Orthogonal Decomposition ROM for QGE

By again utilizing the method of snapshots on the vorticity, we obtain the optimal linear operators $\Phi_\omega \in \mathbb{R}^{n\times r}$ $\Theta_\omega \in \mathbb{R}^{r\times n}$ (orthogonal in some inner product space) that capture the dominant linear dynamics in the vorticity space. The linear operators corresponding to the streamfunction are then obtained by solving the Poisson equation

$$\Theta_\omega = -\Delta\Theta_\psi, \tag{68}$$

with Φ_ψ being defined in a similar fashion, from which a quadratic ROM (65) is constructed as in [25]. In [25], it was shown that a reduced dimension of $r = 25$ is considered medium accuracy for the QGE, therefore this is the choice that we will use in numerical experiments.

5. THEORY-GUIDED AUTOENCODER-BASED ROMS

We now discuss building the neural network-based reduced order model (NN-ROM). Given the principal space variable X, consider an encoder θ that computes its projection U onto the control space (41), and a decoder ϕ that computes the reconstruction $\widetilde{X}$ (42).

Canonical autoencoders simply aim to minimize the reconstruction error:

$$X \approx \widetilde{X}, \tag{69}$$

which attempts to capture the dominant modes of the intrinsic manifold of the data. We identify a property of other reduced order modeling techniques which aims to preserve the physical consistency of the dynamics.

Recall the approximate dynamics in the reduced space (45) which provides an approximation to the reduced dynamics (44):

$$\frac{\mathrm{d}U}{\mathrm{d}t} = \theta'(X) f(X) \approx \theta'(\widetilde{X}) f(\widetilde{X}) = \theta'(\phi(U)) f(\phi(U)). \tag{70}$$

We derive a condition that creates a between the two tendencies in the right hand side of (70).

Theorem 4. *Assume the encoding of the reconstruction is the encoding of the full representation,*

$$\theta(\widetilde{X}) = \theta(X), \tag{71}$$

which we call the right-inverse property. Then the approximation (70) is bounded by,

$$\left\|\theta'(\widetilde{X})\right\| \left\|\phi'(U)\,\theta'(X) f(X) - f(\phi(U))\right\|. \tag{72}$$

Proof: We have $U = \theta(X)$, $\widetilde{X} = \phi(U) = \phi(\theta(X))$. By the right-inverse-property (71),

$$U = \theta(\phi(\theta(X))).$$

Differentiating with respect to time,

$$\frac{\mathrm{d}U}{\mathrm{d}t} = \theta'(\widetilde{X})\,\phi'(U)\,\theta'(X) f(X),$$

and approximating with (45) similar to in (70),

$$\theta'(\widetilde{X})\,\phi'(U)\,\theta'(X) f(X) \approx \theta'(\widetilde{X}) f(\phi(U)), \tag{73}$$

then the term $\theta'(\widetilde{X})$ now appears on both sides of the equation, and the error can be expressed as,

$$\left\|\theta'(\widetilde{X})\left[\phi'(U)\,\theta'(X) f(X) - f(\phi(U))\right]\right\| \le \left\|\theta'(\widetilde{X})\right\| \left\|\phi'(U)\,\theta'(X) f(X) - f(\phi(U))\right\|, \tag{74}$$

as required.

Remark 11. *The condition (72) is exact is difficult to enforce, as it would require the evaluation of the function f many times, which may be an intractable endeavor for large models. It is of independent interest to attempt and enforce this condition, or provide error bounds for certain flavors of models.*

For POD (Section 4.1.2), the right-inverse property (71) is automatically preserved by construction and the linearity of the methods, as

$$\Theta\Phi = I_r, \tag{75}$$

by the orthogonality of Θ and Φ. Therefore,

$$\Theta\Phi\Theta X = \Theta X, \quad \forall X \in \mathbb{R}^n. \tag{76}$$

For non-linear operators, the authors have not explicitly seen this property preserved, however, as the MFEnKF requires sequential applications of projections and interpolations, the authors believe that for the use-case outlined in this paper, the property is especially important.

It is of interest that the right invertibility property is implied by the mere fact that we are looking at preserving non-linear dynamics with the auto-encoder, but is otherwise agnostic to the type of physical system that we are attempting to reduce.

Remark 12. *Note that unlike the POD-ROM whose linear structure induces a purely r-dimensional initial value problem, the NN-ROM (45) still involves n-dimensional function evaluations. In a practical method it would be necessary to reduce the internal dimension of the ROM, however that is significantly outside the scope of this paper.*

5.1. Theory-Guided Autoencoder-Based ROM for Lorenz'96

We seek to construct a neural network $\mathcal{M}^{\mathrm{NN}}$ that is a surrogate ROM for the FOM $\mathcal{M}^X$. We impose that the induced dynamics (45) makes accurate predictions, by not only capturing the intrinsic manifold of the data, but also attempting to capture the inertial manifold of the system. Explicitly, we wish to ensure that the surrogate approximation error in full space,

$$\mathcal{M}^X(X) \approx \phi(\mathcal{M}^{\mathrm{NN}}(\theta(X))) \tag{77}$$

is minimized. We explicitly test the error in full space and not the reduced space, as the full space error is more relevant to the practitioner and for practical application of our methodology.

In this sense (45) would represent an approximation of the dynamics along a submanifold of the inertial manifold. In practice we compute (77) over a short trajectory in the full space started from a certain initial value, and a short trajectory in the latent space started form the projected initial value.

We will however not explicitly enforce (77) in the cost function, as that may be intractable for larger systems. We will instead only enforce the right-inverse property (71) by posing it as a weak constraint of the system.

Combining the canonical autoencoder reconstruction error term (69), and the right-inverse property (71), we arrive at the following loss function for each snapshot:

$$\ell_j(X_j) = \frac{1}{n}\left\|X_j - \phi(\theta(X_j))\right\|_2^2 + \frac{\lambda}{r}\left\|\theta(X_j) - \theta(\phi(\theta(X_j)))\right\|_2^2, \tag{78}$$

where the hyper-parameter λ represents the inverse relative weight of the mismatch of the right inverse property.

The full loss function, combining the cost functions for all T training snapshots:

$$L(X) = \sum_{j=1}^{T} \ell_j(X_j), \tag{79}$$

can be minimized through typical stochastic optimization methods.

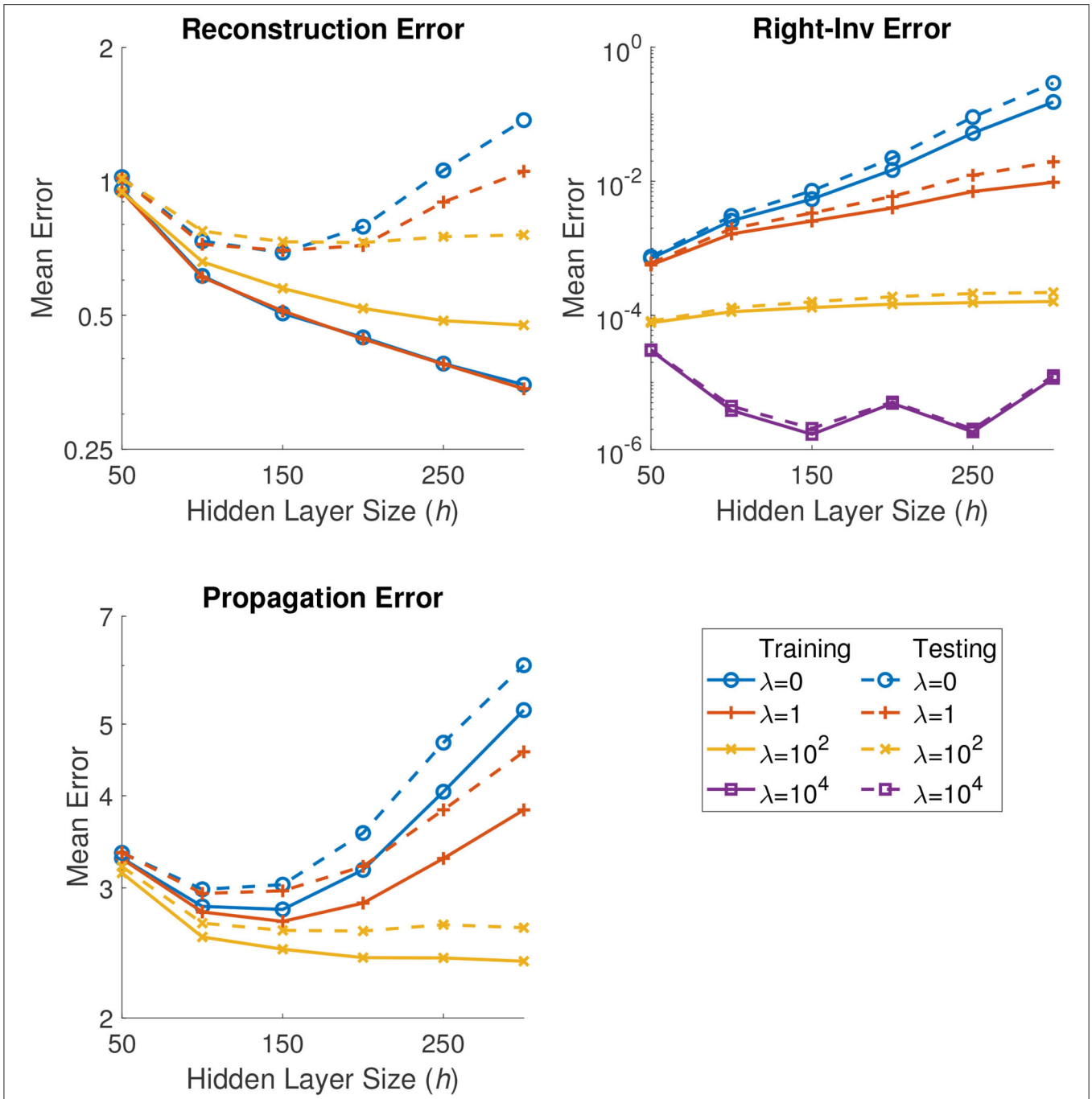


FIGURE 2 | Results for an autoencoder-based ROM of dimension $r = 28$ approximating the Lorenz'96 FOM. The figure illustrates the mean error for the three error types that we observe: reconstruction error (80), the right invertibility error (81), and the propagation error (82), on both the training and testing data sets.

For testing we will look at two errors from the cost function, the reconstruction error,

$$\frac{1}{T}\sum_{j=1}^{T}\frac{1}{n}\left\|X_j-\phi(\theta(X_j))\right\|_2^2, \tag{80}$$

which corresponds to the error in (69), and the right-invertibility error,

$$\frac{1}{T}\sum_{j=1}^{T}\frac{1}{r}\left\|\theta(X_j)-\theta(\phi(\theta(X_j)))\right\|_2^2 \tag{81}$$

corresponds to the error in (71).

Aside from the two errors above, for testing, we additionally observe the propagation error,

$$\frac{1}{T}\sum_{j=1}^{T}\frac{1}{n}\left\|\mathcal{M}^{X}_{t_J,(t_J+K\Delta t)}(X_J)-\phi(\mathcal{M}^{\mathrm{NN}}_{t_J,(t_J+K\Delta t)}(\theta(X_J)))\right\|_2^2, \quad J=\{1,2,\ldots,T\}, \tag{82}$$

which attempts to quantify the mismatch in (77) computed along K steps.

Similar to the POD model, we construct r-dimensional NN-based surrogate ROMs. To this end, we use one hidden layer networks with the element-wise tanh activation function for the encoder (41) and decoder (42):

$$\theta(X) = W_2^\theta \tanh(W_1^\theta X + b_1^\theta) + b_2^\theta, \quad W_1^\theta \in \mathbb{R}^{h\times n}, W_2^\theta \in \mathbb{R}^{r\times h}, b_1^\theta \in \mathbb{R}^h, b_2^\theta \in \mathbb{R}^r, \tag{83}$$

$$\phi(U) = W_2^\phi \tanh(W_1^\phi U + b_1^\phi) + b_2^\phi, \quad W_1^\phi \in \mathbb{R}^{h\times r}, W_2^\phi \in \mathbb{R}^{n\times h}, b_1^\phi \in \mathbb{R}^h, b_2^\phi \in \mathbb{R}^n, \tag{84}$$

where h is the hidden layer dimension, equal for both the encoder and decoder.

The corresponding linearization of the encoder is:

$$\theta'(X) = W_2^\theta \operatorname{diag}\left[1-\tanh(W_1^\theta X + b_1^\theta)^{\circ 2}\right] W_1^\theta, \tag{85}$$

where $(\cdot)^{\circ 2}$ represents element-wise exponentiation, is required for the reduced order dynamics (45).

There are two hyper-parameters of interest, the hidden layer dimension, h, and the right-inverse weak-constraint parameter λ.

We consider the following hidden layer dimensions, $h = 50, 100, 150, 200, 250$, and 300. Additionally we consider the following values for right-inverse weak-constraint weights $\lambda = 0, 1, 10^2, 10^4$. We fix the propagation error parameter to $K = 4$, which corresponds to 24 h in model time, and observe all three errors (80), (81), and (82) on both the training and testing data sets.

We employ the ADAM [57] optimization procedure to train the NN and to produce the various ROMs. Gradients of the loss function are computed through automatic differentiation.

Figure 2 shows results from a representative set of models corresponding to different choices of the λ hyperparameter and the ROM dimension $r = 28$. As can be seen, the value of $\lambda = 10^4$ is too strict of a parameter, thus the cost function ignores all errors other than the right-invertibility constraint, while all other values of λ are produce viable models. The inclusion of the right-invertibility constraint not only improves the propagation error, but also makes the produced models less dependent on the hidden layer dimension h on the test data reconstruction.

We consider the "best" model to be the one that which minimizes the propagation error (82) over the two parameters for each ROM dimension size. **Table 1** shows the optimal parameter choices corresponding to each ROM dimension. The "best" models are chosen for the numerical experiments. Aside from the case $r = 7$, the optimal right-inverse constraint parameter is $\lambda = 10^2$.

TABLE 1 | A table of the optimal autoencoder parameters for the Lorenz '96 NN-ROM for different ROM dimensions r.

r	h	λ
7	50	0
14	50	10^2
21	100	10^2
28	200	10^2
35	200	10^2

Here h is the hidden layer dimension, and λ the right-invertibility constraint weight parameter.

5.2. Theory-Guided Autoencoder-Based ROM for QGE

For a more realistic test case, we construct an autoencoder-based ROM for the QGE. The hyperparameters are chosen based on the information obtained using the Lorenz'96 model, rather than through exhaustive (and computationally-intensive) testing.

As in (83), we construct the encoder and decoder using

$$\theta(X) = W_2^\theta \sigma(W_1^\theta X + b_1^\theta) + b_2^\theta, \quad W_1^\theta \in \mathbb{R}^{h\times n}, W_2^\theta \in \mathbb{R}^{r\times h}, b_1^\theta \in \mathbb{R}^h, b_2^\theta \in \mathbb{R}^r, \tag{86}$$

$$\phi(U) = C\left(W_2^\phi \sigma(W_1^\phi U + b_1^\phi) + b_2^\phi\right), \quad W_1^\phi \in \mathbb{R}^{h\times r}, W_2^\phi \in \mathbb{R}^{n\times h}, b_1^\phi \in \mathbb{R}^h, b_2^\phi \in \mathbb{R}^n, \tag{87}$$

where σ is an approximation to the Gaussian error linear unit [58],

$$\sigma(z) = \frac{z}{1+e^{-1.702z}}, \tag{88}$$

with all operations computed element-wise, and h is the hidden dimension size. The extra constant term C in (87) is a 2D-convolution corresponding to the stencil,

$$\frac{1}{16}\begin{bmatrix} & 3 & \\ 3 & 4 & 3 \\ & 3 & \end{bmatrix} \tag{89}$$

that aims to ensure that the resulting reconstruction does not have sharp discontinuities.

The choice of the activation function in (88) corresponds to a more realistic choice in state-of-the-art neural networks, and helps with choosing a smaller hidden dimension size.

Similar to Section 4.2.2, we make the choice that the reduced dimension size is $r = 25$. Informed by the Lorenz'96 NN-ROM, we take the hidden layer dimension $h = 125$, a medium value in between $h = 100$ and $h = 150$. We again use the hyperparameter value $\lambda = 10^2$.

6. NUMERICAL EXPERIMENTS

The numerical experiments with the Lorenz'96 model compare the following four methodologies:

1. Standard EnKF with the Lorenz'96 full order model;

2. MFEnKF with the POD surrogate model, an approach named MFEnKF(POD);
3. NL-MFEnKF with the autoencoder surrogate model, named NL-MFEnKF(NN); and
4. MFEnKF with the autoencoder surrogate model, named MFEnKF(NN).

Since MFEnKF does not support non-linear projections and interpolations, in MFEnKF(NN) the ensembles are interpolated into the principal space, and assimilated under the assumption that $\Theta = \Phi = I$.

For sequential data assimilation experiments we observe all 40 variables of the Lorenz'96 system, with an observation error covariance of $\Sigma_{\eta_i,\eta_i} = I$. Observations are performed every 0.05 time units corresponding to 6 h in model time. We run 20 independent realizations (independent ensemble initializations) for 1, 100 time steps, but discard the first 100 steps for spinup.

The numerical experiments with the Quasi-geostrophic equations focus only on sequential data assimilation. We compare the following methodologies:

1. Standard EnKF with the QGE full order model;
2. MFEnKF with the POD surrogate model, an approach named MFEnKF(POD); and
3. NL-MFEnKF with the autoencoder surrogate model, named NL-MFEnKF(NN).

We observe 150 equally spaced variables directly, with an observation error covariance of $\Sigma_{\eta_i,\eta_i} = I$. Observations are performed every 0.0109 time units corresponding to 1 day in model time. We run 5 independent realizations (independent ensemble initializations) for 350 time steps, but discard the first 50 steps for spinup.

In order to measure the accuracy of some quantity of interest with respect to the truth, we utilize the spatio-temporal root mean square error (RMSE):

$$\mathrm{RMSE}(X, X^{\mathrm{t}}) = \sqrt{\frac{1}{Nn}\sum_{i=1}^{N}\sum_{k=1}^{n}\left([X_i]_k - \left[X_i^t\right]_k\right)^2}, \qquad (90)$$

throughout the rest of this section. Note here that the number of steps in a given experiment N is not necessarily the number of snapshot data point T.

6.1. Accuracy of ROM Models for Lorenz'96

Our first experiment is concerned with the preservation of energy by different ROMs, and seeks to compare the accuracy of NN-ROM against that of POD-ROM. For the Lorenz'96 model, we use the following equation [59] to model the spatio-temporal kinetic energy,

$$\mathrm{KE} = \sum_{i=1}^{T}\sum_{k=1}^{n}\left([y_i]_k\right)^2, \qquad (91)$$

where T is the number of temporal snapshots of either the training or testing data. **Table 2** shows the relative kinetic energies of the POD-ROM and the NN-ROM reconstructed solutions (42) (the energies of the reconstructed ROM solutions are divided by the kinetic energy of the full order solution) for both the training and testing data.

TABLE 2 | Relative kinetic energies preserved by the reconstructions of the POD-ROM and the NN-ROM solutions of the Lorenz'96 system on both the training and testing data.

r	POD-ROM		NN-ROM	
	Training	Testing	Training	Testing
7	0.52552	0.52351	0.67115	0.67358
14	0.70200	0.69696	0.78783	0.78590
21	0.82222	0.81983	0.91187	0.91292
28	0.90161	0.90051	0.96760	0.96913
35	0.96251	0.96142	0.99095	0.98845

Various reduced-order model dimensions r are considered.

The results lead to several observations. First, the NN-ROM always preserves more energy than the POD-ROM. We have achieved our goal to build an NN-ROM that is more accurate than the POD-ROM. Second, the NN-ROMs with dimensions $r = 21$ and $r = 28$ preserve as much energy as the POD-ROMs with dimension $r = 28$ and $r = 35$, respectively. Intuitively this tells us that they should be just as accurate. Third, all the models preserve almost as much energy on the training as on the testing data, meaning that the models are representative over all possible trajectories.

6.2. Impact of ROM Dimension for Lorenz'96

The second set of experiments seeks to learn how the ROM dimension affects the analysis accuracy for the various multifidelity data assimilation algorithms.

We take the principal ensemble size to be $N_X = 32$, and the surrogate ensemble sizes equal to $N_U = r - 3$, in order to always work in the undersampled regime. All multifidelity algorithms (Sections 3, 2.4) are run with inflation factors $\alpha_X = 1.05$ and $\alpha_U = 1.01$. The traditional EnKF using the full order model is run with an ensemble size of $N = N_X$ and an inflation factor $\alpha = 1.06$ to ensure stability. The inflation factors were chosen by careful hand-tuning to give a fair shot to all algorithms and models.

The results are shown in **Figure 3**. For the "interesting" dimensions $r = 28$, and $r = 35$, the NL-MFEnKF(NN) performs significantly better than the MFEnKF(POD). For a severely underrepresented ROM dimension of $r = 7$, $r = 14$, and $r = 21$ the MFEnKF(POD) outperforms the NL-MFEnKF(NN). The authors believe that this is due to the fact that a non-linear ROM size of less that $r = 28$ dimensions (the rounded-up Kaplan-Yorke dimension) is not sufficient to represent the full order dynamics without looking at additional constraints.

Of note is that, excluding the case of $r = 35$, the MFEnKF(NN) based on the standard MFEnKF method in the principal space is the least accurate among all algorithms, indicating that the non-linear method presented in this paper is truly needed for models involving non-linear model reduction.

We note that for $r = 35$, the suspected intrinsic dimension of the data, the NL-MFEnKF(NN) outperforms the EnKF, both in terms of RMSE and variability within runs. This is additionally

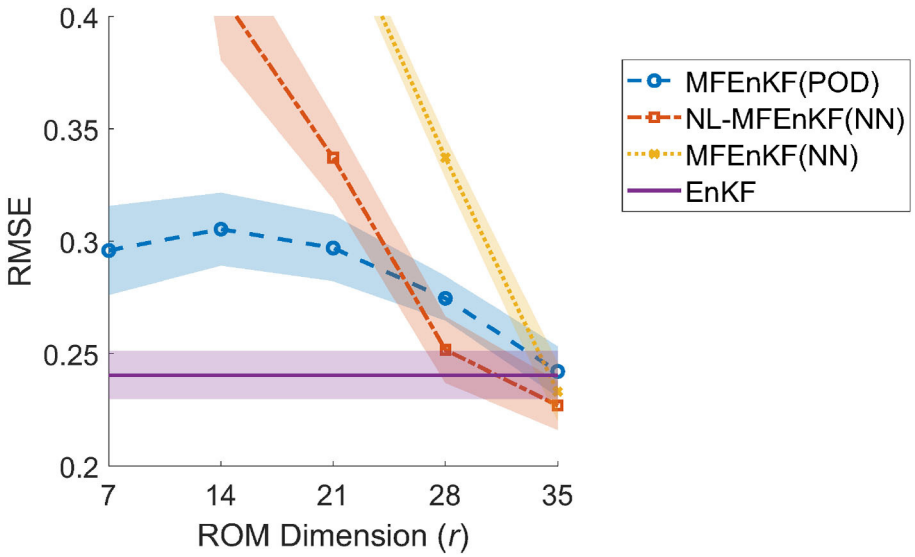


FIGURE 3 | Lorenz'96 analysis RMSE vs. ROM dimension (r) for three multifidelity data assimilation algorithms and the classical EnKF. Ensemble sizes are $N_X = 32$ and $N_U = r - 3$. Error bars show two standard deviations. The inflation factor for the surrogate ROMs is fixed at $\alpha_U = 1.01$, the inflation of $\alpha = 1.07$ is used for the EnKF, and $\alpha_X = 1.05$ is used for all other algorithms.

strengthened by the results of the MFEnKF(NN) assimilated in the principal space, as it implies that there is little-to-no loss of information in the projection into the control space.

We believe that these are very promising results, as they imply that simply capturing the Kaplan-Yorke dimension and properly accounting for the non-linearity of the system could potentially bring in surrogates defined by non-linear operators to data assimilation research.

6.3. Ensemble Size and Inflation for Lorenz'96

Our second to last set of experiments focuses on the particular ROM dimension $r = 28$, as we believe that it is representative of an operationally viable dimension reduction, covering the dimensionality of the global attractor, and experimentally it is the sweet spot where the NL-MFEnKF(NN) beats all others except EnKF.

For each of the four algorithms we vary the principal ensemble size $N_X = N$, and the principal inflation factor $\alpha_X = \alpha$. As before, we set the control ensemble size to $N_U = r - 3 = 25$ and the control-space inflation factor to $\alpha_U = 1.01$.

Figure 4 shows the spatio-temporal RMSE for various choices of ensemble sizes and inflation factors. The results show compelling evidence that NL-MFEnKF(NN) is competitive when compared to MFEnKF(POD); the two methods have similar stability properties for a wide range of principal ensemble size N_X and principal inflation α_X, but NL-MFEnKF(NN) yields smaller analysis errors for almost all scenarios for which the two methods are stable.

For a few points with low values of principal inflation α_X, the NL-MFEnKF(NN) is not as stable as the MFEnKF(POD). This could be due to either an instability in the NN-ROM itself, in the NL-MFEnKF itself, or in the projection and interpolation operators.

An interesting observation is that the MFEnKF(NN), which is assimilated naively in the principal space, becomes less stable for larger ensemble sizes N_X. One possible explanation for this is that the ensemble mean estimates become more accurate, thus the bias between the ancillary and control variates is amplified in (4), and more error is introduced from the surrogate model. This is in contrast to most other ensemble based methods, including all others in this paper, whose error is lowered by increasing ensemble size.

6.4. Ensemble Size and Inflation for QGE

Our last set of experiments focuses on the quasi-geostrophic equations. We use the POD-ROM developed in Section 4.2.2, and the NN-ROM discussed in Section 5.2.

As before, for each of the three algorithms we vary the principal ensemble size $N_X = N$ and principal inflation $\alpha_X = \alpha$. In order to better visualize the results, we fix the control ensemble size to $N_U = 12$, and the control inflation factor to $\alpha_U = 1.05$.

Figure 5 shows the spatio-temporal RMSE for various choices of ensemble sizes and inflation factors. The results provide evidence for the validity of the NL-MFEnKF approach for large-scale data assimilation problems.

For similar values of inflation and ensemble size, the MFEnKF with a POD surrogate is comparable to the NL-MFEnKF with an autoencoder-based surrogate. Both multilevel filters significantly outperform the standard EnKF. The authors believe that these results show convincingly that the NL-MFEnKF formulation is valid for surrogates based on non-linear projection and interpolation.

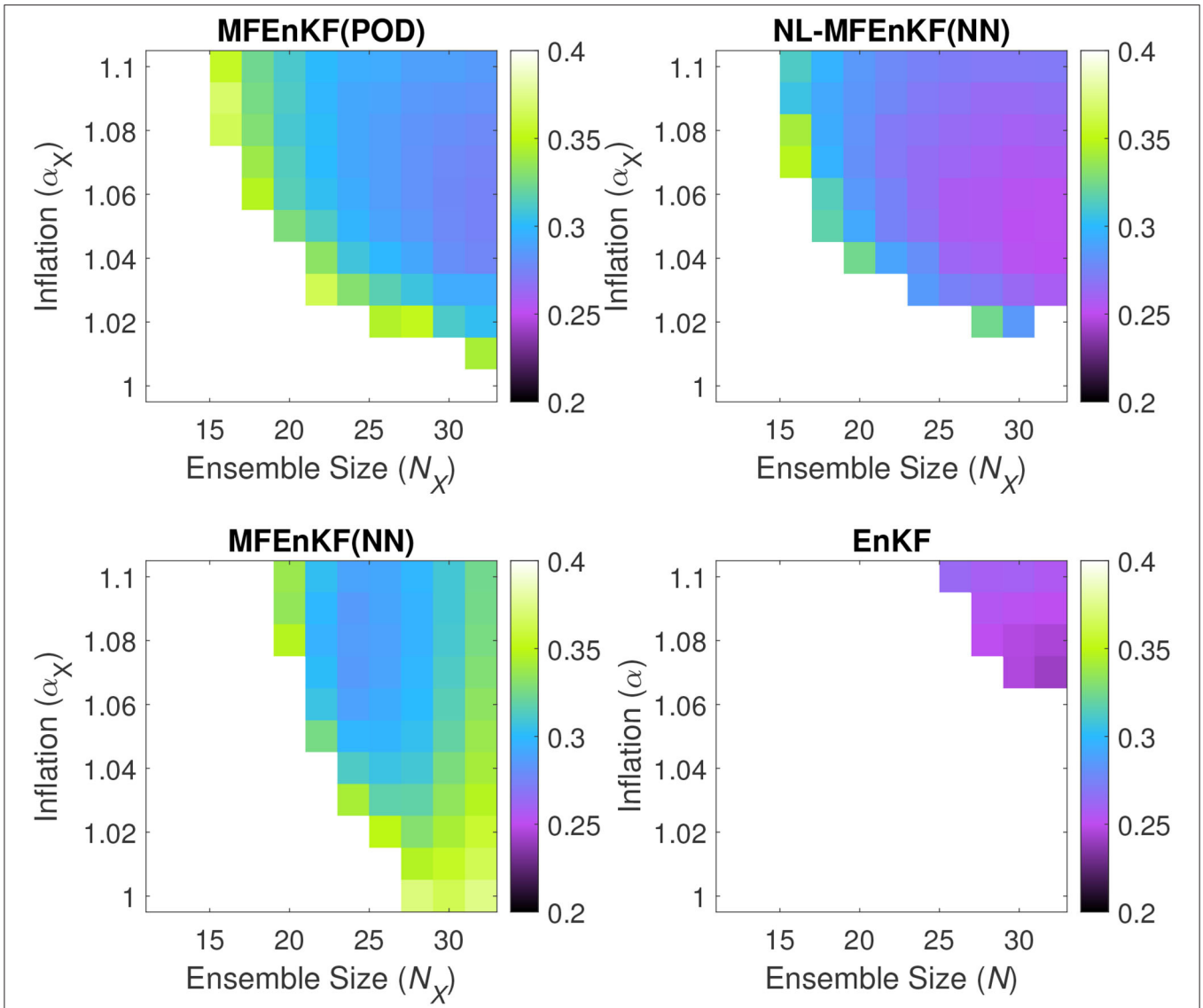


FIGURE 4 | Lorenz'96 analysis RMSE for four data assimilation algorithms and various values of the principal ensemble size N_X and inflation factor α_X. The surrogate ROM size is fixed to $r = 28$ with ensemble size of $N_U = 25$ and inflation of $\alpha_U = 1.01$. The **(top left)** represents the MFEnKF with a POD-ROM surrogate, the **(top right)** represents the NL-MFEnKF with a NN-ROM, the **(bottom left)** represents the MFEnKF with a NN-ROM surrogate assimilated in the principal space, and the **(bottom right)** represents the classical EnKF.

7. CONCLUSIONS

The multifidelity ensemble Kalman filter (MFEnKF) uses a linear control variate framework to increase the computational efficiency of data assimilation; the state of the FOM is the principal variate, and a hierarchy of linear projection ROMs provide the control variates. In this work, the linear control variate framework is generalized to incorporate control variates built using non-linear projection and interpolation operators implemented using autoencoders. The approach, named NL-MFEnKF, enables the use of a much more general class of surrogate models than MFEnKF.

We identify the right-invertibility property of autoencoders as an important feature to support the construction of non-linear reduced order models. This property has previously not been preserved by autoencoders. We propose a methodology for building ROMs based on autoencoders that weakly preserves this property, and show that enforcing this property enhances the prediction accuracy over the standard approach.

We use these elements to construct NL-MFEnKF that extends the multifidelity ensemble Kalman filter framework to work with nonlinear surrogate models. The results obtained in this paper indicate that reduced order models based on non-linear projections that fully capture the intrinsic dimension of the data provide excellent surrogates for use in multifidelity sequential

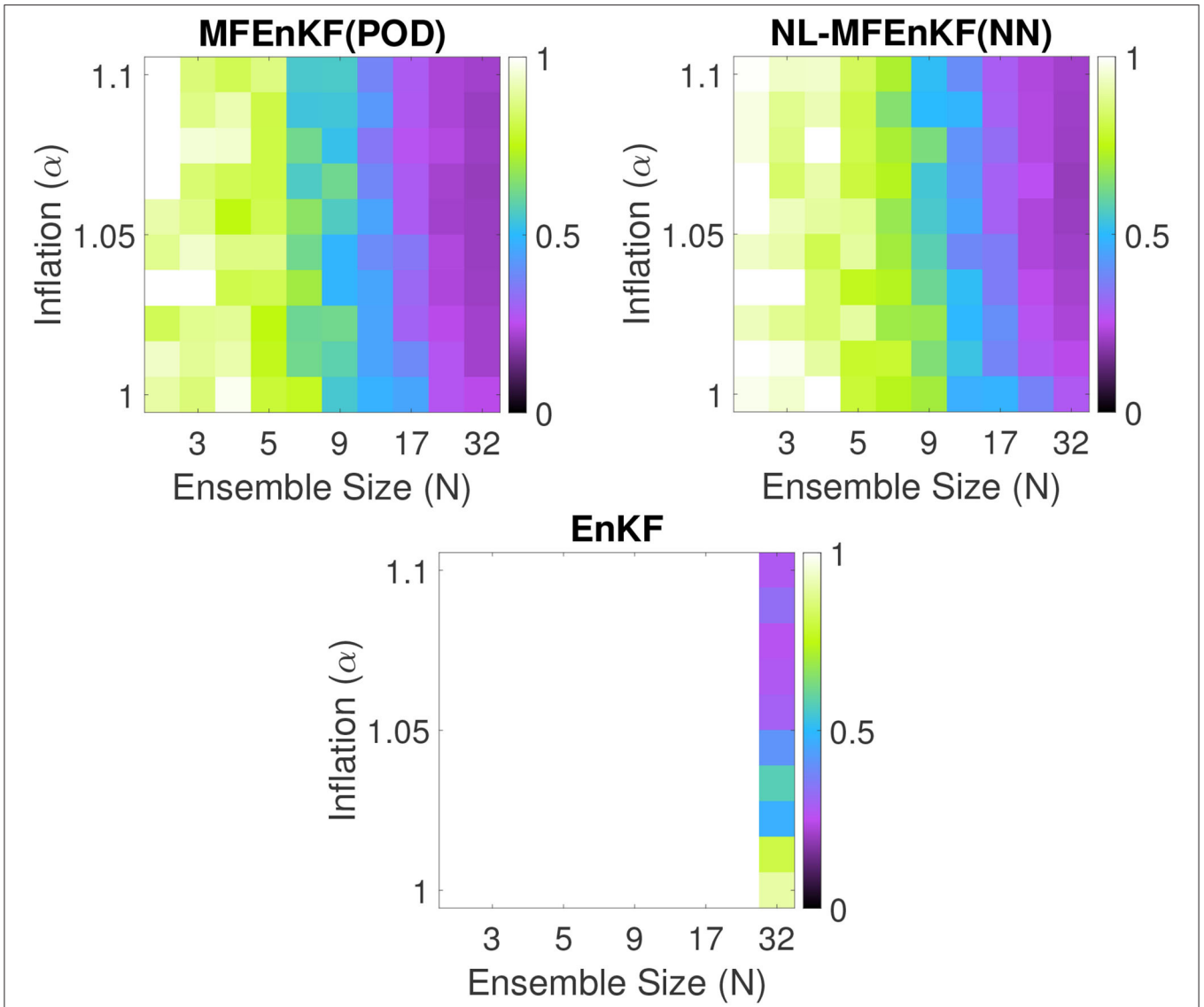


FIGURE 5 | QGE analysis RMSE for four data assimilation algorithms and various values of the principal ensemble size N_X and inflation factor α_X. The surrogate ROM size is fixed to $r = 25$ with ensemble size of $N_U = 12$ and inflation of $\alpha_U = 1.05$. The **(top left)** represents the MFEnKF with a POD-ROM surrogate, the **(top right)** represents the NL-MFEnKF with a NN-ROM, and the **(bottom)** represents the classical EnKF.

data assimilation. Moreover, nonlinear generalizations of the control variate framework result in small approximation errors, and thus the assimilation can be carried out efficiently in the space of a nonlinear reduced model.

Our Numerical experiments with both small scale (Lorenz '96) and medium scale (QGE) models show that the non-linear multifidelity approach has clear advantages over the linear multifilidety approach when the reduced order models are defined by non-linear couplings, and over the standard EnKF for similar high-fidelity ensemble sizes.

From the point of view of machine learning, the major limitations are the constructions of projection and interpolation operators, that do not account for the spatial features of the models, and the model propagation, which does not attempt to utilize state-of-the-art methods such as recurrent neural network models.

From the point of view of data assimilation, there are three limiting factors for the applicability of our method to operational workflows. The first is the use of the perturbed observations ensemble Kalman filter, the second is the absence of localization in our framework, and the third is the absence of model error both for the full-order and surrogate models.

One potential avenue of future research would be into adaptive inflation techniques for multifidelity data assimilation algorithms similar in vein to [60].

Future work addressing all the problems and research avenues above would lead to the successful application of the NL-MFENKF to operational problems such as numerical weather prediction.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

AP performed the numerical experiments and wrote the first draft. All authors contributed to the concept and design of the study. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

This work was supported by DOE through award ASCR DE-SC0021313, by NSF through award CDS&E–MSS 1953113.

ACKNOWLEDGMENTS

The authors would like to thank the rest of the members from the Computational Science Laboratory at Virginia Tech, and Traian Iliescu from the Mathematics Department at Virginia Tech.

REFERENCES

1. Reich S, Cotter C. *Probabilistic Forecasting and Bayesian Data Assimilation*. Cambridge: Cambridge University Press (2015). doi: 10.1017/CBO9781107706804
2. Asch M, Bocquet M, Nodet M. Data assimilation: methods, algorithms, and applications. *SIAM*. (2016) 29:2318–31. doi: 10.1137/1.9781611974546
3. Karpatne A, Atluri G, Faghmous JH, Steinbach M, Banerjee A, Ganguly A, et al. Theory-guided data science: a new paradigm for scientific discovery from data. *IEEE Trans Knowl Data Eng*. (2017) 29:2318–31. doi: 10.1109/TKDE.2017.2720168
4. Willard J, Jia X, Xu S, Steinbach M, Kumar V. Integrating physics-based modeling with machine learning: a survey. *arXiv preprint arXiv:200304919*. (2020). doi: 10.48550/arXiv.2003.04919
5. Giles MB. Multilevel Monte Carlo path simulation. *Oper Res*. (2008) 56:607–17. doi: 10.1287/opre.1070.0496
6. Giles MB. Multilevel Monte Carlo methods. *Acta Numer*. (2015) 24:259–328. doi: 10.1017/S096249291500001X
7. Evensen G. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *J Geophys Res*. (1994) 99:10143–62. doi: 10.1029/94JC00572
8. Evensen G. *Data Assimilation: the Ensemble Kalman Filter*. Heidelberg: Springer Science & Business Media (2009).
9. Burgers G, van Leeuwen PJ, Evensen G. Analysis scheme in the ensemble Kalman Filter. *Month Weath Rev*. (1998) 126:1719–24. doi: 10.1175/1520-0493(1998)126<1719:ASITEK>2.0.CO;2
10. Nino-Ruiz ED, Sandu A. Efficient parallel implementation of DDDAS inference using an ensemble Kalman filter with shrinkage covariance matrix estimation. *Clust Comput*. (2019) 22:2211–21. doi: 10.1007/s10586-017-1407-1
11. Nino-Ruiz ED, Sandu A. Ensemble Kalman filter implementations based on shrinkage covariance matrix estimation. *Ocean Dyn*. (2015) 65:1423–39. doi: 10.1007/s10236-015-0888-9
12. Nino-Ruiz ED, Sandu A. An ensemble Kalman filter implementation based on modified Cholesky decomposition for inverse covariance matrix estimation. *SIAM J Sci Comput*. (2018) 40:A867–86. doi: 10.1137/16M1097031
13. Petrie R. *Localization in the Ensemble Kalman Filter*. MSc Atmosphere, Ocean and Climate University of Reading (2008).
14. Popov AA, Sandu A. A Bayesian approach to multivariate adaptive localization in ensemble-based data assimilation with time-dependent extensions. *Nonlin Process Geophys*. (2019) 26:109–22. doi: 10.5194/npg-26-109-2019
15. Moosavi ASZ, Attia A, Sandu A. Tuning covariance localization using machine learning. In: *Machine Learning and Data Assimilation for Dynamical Systems track, International Conference on Computational Science ICCS 2019*. Vol. 11539 of Lecture Notes in Computer Science. Faro (2019). p. 199–212. doi: 10.1007/978-3-030-22747-0_16
16. Cao Y, Zhu J, Navon IM, Luo Z. A reduced-order approach to four-dimensional variational data assimilation using proper orthogonal decomposition. *Int J Numer Meth Fluids*. (2007) 53:1571–83. doi: 10.1002/fld.1365
17. Farrell BF, Ioannou PJ. State estimation using a reduced-order Kalman filter. *J Atmos Sci*. (2001) 58:3666–80. doi: 10.1175/1520-0469(2001)058<3666:SEUARO>2.0.CO;2
18. Peherstorfer B, Willcox K, Gunzburger M. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Rev*. (2018) 60:550–91. doi: 10.1137/16M1082469
19. Hoel H, Law KJH, Tempone R. Multilevel ensemble Kalman filtering. *SIAM J Numer Anal*. (2016) 54. doi: 10.1137/15M100955X
20. Chernov A, Hoel H, Law KJH, Nobile F, Tempone R. Multilevel ensemble Kalman filtering for spatio-temporal processes. *Numer Math*. (2021) 147:71–125. doi: 10.1007/s00211-020-01159-3
21. Chada NK, Jasra A, Yu F. Multilevel ensemble Kalman-Bucy filters. *arXiv preprint arXiv:201104342*. (2020). doi: 10.48550/arXiv.2011.04342
22. Hoel H, Shaimerdenova G, Tempone R. Multilevel ensemble Kalman filtering based on a sample average of independent EnKF estimators. *Found Data Sci*. (2019) 2:101–121. doi: 10.3934/fods.2020017
23. Brunton SL, Kutz JN. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge: Cambridge University Press (2019). doi: 10.1017/9781108380690
24. Aggarwal CC. *Neural Networks and Deep Learning*. Cham: Springer (2018). doi: 10.1007/978-3-319-94463-0
25. Popov AA, Mou C, Sandu A, Iliescu T. A multifidelity ensemble Kalman Filter with reduced order control variates. *SIAM J Sci Comput*. (2021) 43:A1134–62. doi: 10.1137/20M1349965
26. Popov AA, Sandu A. Multifidelity data assimilation for physical systems. In: Park SK, Xu L, editors. *Data Assimilation for Atmospheric, Oceanic and Hydrologic Applications*. Vol. IV. Springer (2021). doi: 10.1007/978-3-030-77722-7_2
27. Kalnay E. *Atmospheric Modeling, Data Assimilation, and Predictability*. Cambridge: Cambridge University Press (2003). doi: 10.1017/CBO9780511802270
28. Blanchard E, Sandu A, Sandu C. Parameter estimation for mechanical systems via an explicit representation of uncertainty. *Eng Comput Int J Comput Aided Eng Softw*. (2009) 26:541–69. doi: 10.1108/02644400910970185
29. Blanchard E, Sandu A, Sandu C. Polynomial chaos based parameter estimation methods for vehicle systems. *J Multi-Body Dyn*. (2010) 224:59–81. doi: 10.1243/14644193JMBD204
30. Blanchard E, Sandu A, Sandu C. A polynomial chaos-based Kalman filter approach for parameter estimation of mechanical systems. *J Dyn Syst Measure Control*. (2010) 132:18. doi: 10.1115/1.4002481
31. Constantinescu EM, Sandu A, Chai T, Carmichael GR. Ensemble-based chemical data assimilation. II: Covariance localization. *Q J R Meteorol Soc*. (2007) 133:1245–56. doi: 10.1002/qj.77

32. Constantinescu EM, Sandu A, Chai T, Carmichael GR. Ensemble-based chemical data assimilation. I: General approach. *Q J R Meteorol Soc*. (2007) 133:1229–43. doi: 10.1002/qj.76
33. Constantinescu EM, Sandu A, Chai T, Carmichael GR. Assessment of ensemble-based chemical data assimilation in an idealized setting. *Atmos Environ*. (2007) 41:18–36. doi: 10.1016/j.atmosenv.2006.08.006
34. Strogatz SH. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Boca Raton, FL: CRC Press (2018). doi: 10.1201/9780429399640
35. Jaynes ET. *Probability Theory: The Logic of Science*. Cambridge: Cambridge University Press (2003). doi: 10.1017/CBO9780511790423
36. Owen AB. *Monte Carlo theory, methods and examples* (2013). Available online at: https://artowen.su.domains/mc/
37. Rubinstein RY, Marcus R. Efficiency of multivariate control variates in Monte Carlo simulation. *Oper Res*. (1985) 33:661–77. doi: 10.1287/opre.33.3.661
38. Popov AA, Sandu A. An explicit probabilistic derivation of inflation in a scalar ensemble Kalman Filter for finite step, finite ensemble convergence. *arXiv:2003.13162*. (2020). doi: 10.48550/arXiv.2003.1316
39. Sakov P, Oke PR. A deterministic formulation of the ensemble Kalman filter: an alternative to ensemble square root filters. *Tellus A*. (2008) 60:361–71. doi: 10.1111/j.1600-0870.2007.00299.x
40. Sell GR, You Y. *Dynamics of Evolutionary Equations*. Vol. 143. New York, NY: Springer Science & Business Media (2013).
41. Lee JA, Verleysen M. *Nonlinear Dimensionality Reduction*. New York, NY: Springer Science & Business Media (2007). doi: 10.1007/978-0-387-39351-3
42. Lee K, Carlberg KT. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *J Comput Phys*. (2020) 404:108973. doi: 10.1016/j.jcp.2019.108973
43. Nelson BL. On control variate estimators. *Comput Oper Res*. (1987) 14:219–25. doi: 10.1016/0305-0548(87)90024-4
44. Lorenz EN. Predictability: a problem partly solved. In: *Proc. Seminar on Predictability*. Vol. 1. Reading (1996).
45. Foster EL, Iliescu T, Wang Z. A finite element discretization of the streamfunction formulation of the stationary quasi-geostrophic equations of the ocean. *Comput Methods Appl Mech Engrg*. (2013) 261:105–17. doi: 10.1016/j.cma.2013.04.008
46. Ferguson J. *A Numerical Solution for the Barotropic Vorticity Equation Forced by an Equatorially Trapped Wave*. Victoria, BC: University of Victoria (2008).
47. Majda AJ, Wang X. *Nonlinear Dynamics and Statistical Theories for Basic Geophysical Flows*. Cambridge: Cambridge University Press (2006). doi: 10.1017/CBO9780511616778
48. Greatbatch RJ, Nadiga BT. Four-gyre circulation in a barotropic model with double-gyre wind forcing. *J Phys Oceanogr*. (2000) 30:1461–71. doi: 10.1175/1520-0485(2000)030<1461:FGCIAB>2.0.CO;2
49. Roberts S, Popov AA, Sandu ASA. ODE Test Problems: a MATLAB suite of initial value problems. *arXiv [Preprint]*. (2019). arXiv: 1901.04098. Available online at: https://arxiv.org/pdf/1901.04098.pdf
50. Computational Science Laboratory. *ODE Test Problems*. (2021). Available online at: https://github.com/ComputationalScienceLaboratory/ODE-Test-Problems
51. van Kekem DL. *Dynamics of the Lorenz-96 Model: Bifurcations, Symmetries and Waves*. Groningen: University of Groningen (2018). doi: 10.1142/S0218127419500081
52. Bahadur N, Paffenroth R. Dimension estimation using autoencoders. *arXiv preprint arXiv:190910702*. (2019).
53. Sirovich L. Turbulence and the dynamics of coherent structures. I. Coherent structures. *Q Appl Math*. (1987) 45:561–71. doi: 10.1090/qam/910462
54. Mou C, Wang Z, Wells DR, Xie X, Iliescu T. Reduced order models for the quasi-geostrophic equations: a brief survey. *Fluids*. (2021) 6:16. doi: 10.3390/fluids6010016
55. San O, Iliescu T. A stabilized proper orthogonal decomposition reduced-order model for large scale quasigeostrophic ocean circulation. *Adv Comput Math*. (2015) 41:1289–319. doi: 10.1007/s10444-015-9417-0
56. Mou C, Liu H, Wells DR, Iliescu T. Data-driven correction reduced order models for the quasi-geostrophic equations: a numerical investigation. *Int J Comput Fluid Dyn*. (2020) 1–13.
57. Kingma DP, Ba J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:14126980*. (2014). doi: 10.48550/arXiv.1412.6980
58. Hendrycks D, Gimpel K. Gaussian Error Linear Units (GELUs). *arXiv:1606.08415*. (2020). doi: 10.48550/arXiv.1606.08415
59. Karimi A, Paul MR. Extensive chaos in the Lorenz-96 model. *Chaos*. (2010) 20:043105. doi: 10.1063/1.3496397
60. Anderson JL. Spatially and temporally varying adaptive covariance inflation for ensemble filters. *Tellus Ser A*. (2009) 61:72–83. doi: 10.1111/j.1600-0870.2008.00361.x

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers | Frontiers in Physics

ORIGINAL RESEARCH
published: 30 June 2022
doi: 10.3389/fphy.2022.890910

Disentangling Generative Factors of Physical Fields Using Variational Autoencoders

*Christian Jacobsen and Karthik Duraisamy**

Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, United States

The ability to extract generative parameters from high-dimensional fields of data in an unsupervised manner is a highly desirable yet unrealized goal in computational physics. This work explores the use of variational autoencoders for non-linear dimension reduction with the specific aim of *disentangling* the low-dimensional latent variables to identify independent physical parameters that generated the data. A disentangled decomposition is interpretable, and can be transferred to a variety of tasks including generative modeling, design optimization, and probabilistic reduced order modelling. A major emphasis of this work is to characterize disentanglement using VAEs while minimally modifying the classic VAE loss function (i.e., the Evidence Lower Bound) to maintain high reconstruction accuracy. The loss landscape is characterized by over-regularized local minima which surround desirable solutions. We illustrate comparisons between disentangled and entangled representations by juxtaposing learned latent distributions and the true generative factors in a model porous flow problem. Hierarchical priors are shown to facilitate the learning of disentangled representations. The regularization loss is unaffected by latent rotation when training with rotationally-invariant priors, and thus learning non-rotationally-invariant priors aids in capturing the properties of generative factors, improving disentanglement. Finally, it is shown that semi-supervised learning - accomplished by labeling a small number of samples (O (1%))–results in accurate disentangled latent representations that can be consistently learned.

Keywords: generative modeling, unsupervised learning, variational autoencoders, scientific machine learning, disentangling

OPEN ACCESS

Edited by:
Traian Iliescu,
Virginia Tech, United States

Reviewed by:
Claire Heaney,
Imperial College London, United Kingdom
Andrey Popov,
Virginia Tech, United States

****Correspondence:***
Karthik Duraisamy
kdur@umich.edu

Specialty section:
This article was submitted to Statistical and Computational Physics, a section of the journal Frontiers in Physics

Received: *07 March 2022*
Accepted: *19 May 2022*
Published: *30 June 2022*

Citation:
Jacobsen C and Duraisamy K (2022) Disentangling Generative Factors of Physical Fields Using Variational Autoencoders. Front. Phys. 10:890910. doi: 10.3389/fphy.2022.890910

1 INTRODUCTION

Unsupervised representation learning is a popular area of research because of the need for low-dimensional representations in unlabeled data. Low-dimensional *latent* representations of high-dimensional data have many applications ranging from facial image generation [1] and music generation [2] to autonomous controls [3] among many others. Generative adversarial networks (GANs) [4], variational autoencoders (VAEs) [5] and their variants [6–9], among other methods, aim to approximate an underlying distribution $p(y)$ of high-dimensional data through a two-step process. Compressed representations z are sampled from a low-dimensional–yet unknown–distribution $p(z)$. In the case of VAEs, which is the focus of this work, an encoding distribution $p(z|y)$ and a decoding distribution are learned simultaneously by maximizing a bound on the likelihood of the data (i.e., the evidence lower bound (ELBO) [5]). Thus, a mapping from the high-dimensional space to a low-dimensional space and the corresponding inverse mapping is

learned simultaneously, allowing approximations of both $p(y)$ *and* $p(z)$. Learning the lower-dimensional representation, or *latent space*, can facilitate computationally-efficient data generation and extract only the information necessary to reconstruct the data [10]. Modifications to the ELBO objective have been suggested in the literature, primarily with improved disentanglement in mind. The β-VAE [6] was introduced to improve disentanglement by adjusting the weight of regularization loss. FactorVAE [8] introduces a total correlation term (TC) to encourage learning a factorized latent representation. InfoVAE [9] augments the ELBO with a term to promote maximization of the mutual information between the data and the learned representation. Many other developments based on the VAE objective have been introduced in the literature. VAEs have been implemented in many applications including inverse problems [11], extracting physical parameters from spatio-temporal data [12], and constructing probabilistic reduced order models [13, 14], among others.

To illustrate the idea of disentanglement and its implications, consider a dataset consisting of images of teapots [15]. Each image is generated from 3 parameters indicating the color of the teapot (RGB) and 2 parameters corresponding to the angle the teapot is viewed from. Thus, even though the RGB image may be very high dimensional, the intrinsic dimensionality is just 5. Representation learning can be used to extract a low-dimensional latent model containing useful and meaningful representations of the high-dimensional images. Learned latent representations need not be *disentangled* to be useful in some sense, but disentanglement enhances interpretability of the representation. Disentanglement references a structure of the latent distribution in which changes in each parameter in the learned representation correspond directly to changes in a single yet different generative parameter. Humans tend to naturally and easily identify independent factors of variation, and thus a disentangled representation often corresponds to one which would be naturally identified by a human. A representation which is more naturally explained by a human observer is therefore one characterized by greater interpretability. In an unsupervised setting, one cannot guarantee that a disentangled representation can be learned.

The requirement for disentanglement depends on the task at hand, but a disentangled representation may be used in many tasks containing different objectives. Indeed, [10] state that "the most robust approach to feature learning is to disentangle as many factors as possible, discarding as little information about the data as is practical." In the teapot example, changes in one of the learned latent dimensions may correspond to changes in the color red and one of the viewing angles, which would indicate an entangled representation. Another example, more relevant to our work, is that of fluid flow over an airfoil. Learning a disentangled representation of the flow conditions along with the shape parameters using VAEs can allow rapid prediction of the flow field with enhanced interpretability of the latent representation, facilitating efficient computation of the task at hand. The disentangled representation can be transferred to a variety of tasks easily such as design optimization, developing reduced order models in the latent space or parameter inference from flow fields. It is the ability of disentangled representations to transfer across tasks with ease and interpretability which makes them so useful. In many practical physics problems, full knowledge regarding the underlying generative parameters of high-dimensional data may not exist, thus making it challenging to ascertain the quality of disentanglement.

Disentanglement using VAEs was first addressed in the literature [6]; [8] by modifying the strength of regularization in the ELBO loss, with the penalty of sub-optimal compression and reconstruction. FactorVAEs [6] encourage a factorized representation, which can be useful for disentanglement in the case of independent generative parameters, but undesirable when parameters are correlated. [16] suggest that the ability of the VAE to learn disentangled representations is not inherent to the framework itself, but an "accidental" byproduct of the typically assumed factorized form of the encoder. The prior distribution is of particular importance as the standard normal prior often assumed allows for rotation of the latent space with no effect on the ELBO loss. Disentangled representations are still often learned due to a factorized form of the encoding distribution with sufficiently large weight on regularization. Additional interpretations and insight into the disentanglement ability of VAEs are found in [17].

Our work on unsupervised representation learning is motivated from a computational-physics perspective. We focus on the application of VAEs for use with data generated by partial differential equations (PDEs). The central questions we seek to answer in this work are: 1) can we reliably disentangle parameters from data obtained from PDEs governing physical problems using VAEs, and 2) what are the characteristics of disentangled representations? Learning disentangled representations can be useful in many capacities: developing probabilistic reduced order models, design optimization, parameter extraction, and data interpolation, among others. Many of the applications of such representations, and the ability to transfer between them, rely heavily on the disentanglement of the latent space. Differences in disentangled and entangled representations are identified, and conclusions are drawn regarding the inconsistencies in learning such representations. Our goals are not to compare the available methods to promote disentanglement, as in [18], but rather to illustrate the use of VAEs without modifying the ELBO and to understand the phenomenon of disentanglement itself in this capacity. The use of hierarchical priors is shown to greatly improve the prospect of learning a disentangled representation in some cases without altering the standard VAE loss through the learning of non-rotationally-invariant priors. Along the way, we provide intuition on the objective of VAEs through connections to rate-distortion theory, illustrate some of the challenges of implementing and training VAEs, and provide potential methods to overcome some of these issues such as "vanishing KL" [19].

The outline of this paper is as follows: In **Section 2**, we introduce the VAE, connect it to rate-distortion (RD) theory, discuss disentanglement, and derive a bound on the classic VAE loss (the ELBO) using hierarchical priors (HP). In **Section 3**, we introduce a sample application of Darcy flow as the main

illustrative example of this work. In **Section 4**, we present challenges in training VAEs and include possible solutions, and investigate the ELBO loss landscape. We illustrate disentanglement of parameters on the Darcy flow problem, and provide insight into the phenomenon of disentanglement in **Section 5**. The use of a small amount of labeled data (semi-supervised learning) is considered in **Section 6**. In **Section 7**, conclusions and insights are drawn on the results of our work, and future directions are discussed.

The numerical experiments in this paper can be recreated using our code provided in https://github.com/christian-jacobsen/Disentangling-Physical-Fields.

2 VARIATIONAL AUTOENCODER FORMULATION

In many applications of representation learning, it is generally desirable that the latent representation be maximally compressed. In other words, the low dimensional representation contains only the information required to reconstruct the original data, discarding irrelevant information. The VAE framework used extensively in this work is a method of data compression with many ties to information theory [20]. In applications with little to no knowledge regarding the nature of obtained data, the latent factors extracted using VAEs can act as a set of features describing the generative parameters underlying the data. A direct correlation between the generative parameters and the compressed representation, or a disentangled representation, is sought such that the representation can be applied to a multitude of downstream tasks. Some example tasks include performing predictions on new generative parameters, interpreting the data in the case of unknown generative parameters, and computationally efficient design optimization.

Data snapshots obtained from some physical system or a model of that system is represented here by random variable $Y\colon \Omega \to \mathcal{Y}$ where Ω is a sample space and $\mathcal{Y}$ is a measurable space ($\mathcal{Y} = \mathbb{R}^m$ will be assumed for the remainder of this work). Each realization of Y is generated from a function of Θ such that $\Theta\colon \Omega \to \mathbb{R}^s$ is a random variable representing generative parameters with distribution $p(\theta)$. With no prior knowledge of Θ, the random variable $Z\colon \Omega \to \mathbb{R}^n$ represents the latent parameters to be inferred from the data. A probabilistic relationship between Θ and Y is sought in an unsupervised manner using only samples from $p(y)$.

The VAE framework infers a latent-variable model by replacing the posterior $p(z|y)$ with a parameterized approximating posterior $q_\phi(z|y)$ [5], known as the encoding distribution. A parameterized decoding distribution $p_\psi(y|z)$ is also constructed to predict data samples given samples from the latent space. Only the encoding distribution and the decoding distribution are learned in the VAE framework, but the *aggregated posterior* $q_\phi(z)$ (to the best of our knowledge, first referred to in this way by [21]), is of particular importance in disentanglement. It is defined as the marginal latent distribution induced by the encoder

$$q_\phi(z) \triangleq \int_{\mathcal{Y}} p(y) q_\phi(z|y) dy, \tag{1}$$

where the true data distribution is denoted by $p(y)$. The induced data distribution is the marginal output distribution induced by the decoder

$$p_\psi(y) \triangleq \int_{\mathbb{R}^n} p(z) p_\psi(y|z) dz. \tag{2}$$

It is noted that the true data distribution is typically unknown; only samples of data $\{y^{(i)}\}_{i=1}^N$ are available. The empirical data distribution is thus denoted $\hat{p}(y)$, and any expectation with respect to the empirical distribution is simply computed as an empirical average $\mathbb{E}_{\hat{p}(y)}[f(y)] \triangleq \frac{1}{N}\sum_{i=1}^N f(y^{(i)})$.

Learning the latent model is accomplished by simultaneously learning the encoding and decoding distributions through maximizing the evidence lower bound (ELBO), which is a lower bound on the log-likelihood [22]. To derive the ELBO loss, we begin by expanding the relative entropy between the data distribution and the *induced* data distribution

$$D_{KL}\left[p(y)||p_\psi(y)\right] = \mathbb{E}_{Y\sim p(y)}\left[\log p(y)\right] - \mathbb{E}_{Y\sim p(y)}\left[\log p_\psi(y)\right]$$

where the first term on the right hand side is the negative differential entropy $-H(Y)$. Noting that relative entropy D_{KL}—also often called the Kullback–Leibler divergence, which is a measure of the distance between two probability distributions—is always greater than or equal to zero and introducing Bayes' rule as

$$p_\psi(y) = \frac{p_\psi(y|z) p(z) p_\phi(z|y)}{p(z|y) p_\phi(z|y)},$$

we arrive at the following inequality

$$H(Y) + \mathbb{E}_{Y\sim p(y)}\left[D_{KL}\left[p_\phi(z|y)||p(z|y)\right]\right] \le \mathbb{E}_{p(y)}\left[\mathbb{E}_{q_\phi(z|y)}\left[\log p_\psi(y|z)\right]\right] - \mathbb{E}_{p(y)} D_{KL}\left[q_\phi(z|y)||p(z)\right].$$

Thus,

$$\mathbb{E}_{p(y)}\left[\log(p(y))\right] \ge \mathbb{E}_{p(y)}\left[\mathbb{E}_{q_\phi(z|y)}\left[\log p_\psi(y|z)\right]\right] - \mathbb{E}_{p(y)} D_{KL}\left[q_\phi(z|y)||p(z)\right], \tag{3}$$

where $p(z)$ is a prior distribution. The prior is specified by the user in the classic VAE framework. The right-hand side in **Eq. 3** is the well-known ELBO. Maximizing this lower bound on the log-likelihood of the data is done by minimizing the negative ELBO. The optimization is performed by learning the encoder and decoder parameterized as neural networks. The negative ELBO is defined as

$$-ELBO = \mathbb{E}_{p(y)} D_{KL}\left[q_\phi(z|y)||p(z)\right] + \mathbb{E}_{p(y)}\left[\mathbb{E}_{q_\phi(z|y)}\left[-\log p_\psi(y|z)\right]\right], \tag{4}$$

and we assume $\mathcal{L}_{VAE} \approx -ELBO$, where the difference results in the expectation being evaluated over the empirical data distribution in $\mathcal{L}_{VAE}$. The VAE loss function is defined as

$$\mathcal{L}_{VAE} = \mathbb{E}_{\hat{p}(y)} D_{KL}\left[q_\phi(z|y)||p(z)\right] + \mathbb{E}_{\hat{p}(y)}\left[\mathbb{E}_{q_\phi(z|y)}\left[-\log p_\psi(y|z)\right]\right], \tag{5}$$

where the first term on the right-hand side is the regularization loss $\mathcal{L}_{REG}$ and drives the encoding distribution closer (in the sense of minimizing KL divergence) to the prior distribution. The second term on the right-hand side is the reconstruction error $\mathcal{L}_{REC}$ and encourages accurate reconstruction of the data.

Selecting the prior distribution as well as the parametric form of the encoding and decoding distribution can allow closed form solutions to compute $\mathcal{L}_{VAE}$. The prior distribution is often conveniently chosen as a standard normal distribution $p(z) = \mathcal{N}(z; 0, I_{n\times n})$. The encoding and decoding distributions are also often chosen as factorized normal distributions $q_\phi(z|y) = \mathcal{N}(z; \mu_\phi(y), \mathrm{diag}(\sigma_\phi(y)))$ and $p_\psi(y|z) = \mathcal{N}(y; \mu_\psi(z), \mathrm{diag}(\sigma_\psi(y)))$, where the mean and log-variance of each distribution are functions parameterized by neural networks. Selecting the parameterized form of these distributions facilitates the reparameterization trick [5], allowing backpropagation through sampling operations during training. This selection of the prior, encoding, and decoding distributions allows a closed form solution to compute $\mathcal{L}_{VAE}$.

2.1 Disentanglement

Disentanglement is realized when variations in a single latent dimension correspond to variations in a single generative parameter. This allows the latent space to be interpretable by the user and improves transferability of representations between tasks. Disentanglement may not be required for some tasks which may not require knowledge on each parameter individually or perhaps only a subset of the generative parameters. Nevertheless, a disentangled representation can be leveraged across many tasks. [10] note that a disentangled representation captures each of the relevant features of the data, but downstream applications may only require a subset of these factors. We therefore hypothesize that disentangled representations lead to a more comprehensive range of downstream applications over non disentangled representations.

Many metrics of disentanglement exist in the literature [18], few of which take into account the generative parameter data. Often knowledge on the generative parameters is lacking, and these metrics can be used to evaluate disentanglement in that case (although there is no consensus on which metric is appropriate). In controlled experiments, however, knowledge on generative parameters is available, and correlation between the latent space and the generative parameter space can be directly determined. To evaluate disentanglement in a computationally efficient manner, we propose a disentanglement score

$$S_D = \frac{1}{n}\sum_i \frac{\max_j \left|\mathrm{cov}\left(z_i, \theta_j\right)\right|}{\sum_j \left|\mathrm{cov}\left(z_i, \theta_j\right)\right|}, \tag{6}$$

where z_i indicates the ith component of the latent vector $\forall i \in \{1, \ldots, n\}$ and θ_j indicates the jth component of the generative parameter vector $\forall j \in \{1, \ldots, s\}$. Noting that

$$\frac{\max_j \left|\mathrm{cov}\left(z_i, \theta_j\right)\right|}{\sum_j \left|\mathrm{cov}\left(z_i, \theta_j\right)\right|} \in [1/s, 1],$$

it is clear that $S_D \in [1/s, 1]$. It is noted that this score is not used *during* the training process. This score is created from the intuition that each latent parameter should be correlated to only a single generative parameter. One might note some issues with this disentanglement score. For instance, if multiple latent dimensions are correlated to the same generative parameter dimension, the score will be inaccurate. Similarly, if the latent dimension is greater than the generative parameter dimension, some latent dimensions may contain no information about the data and be uncorrelated to all dimensions, inaccurately reducing the score. For the cases presented here (we will use the score only when $n = s$), **Eq. 6** suffices as a reasonable measure of disentanglement. This score is used as an efficient means of scoring disentanglement when efficiency is important, but we propose another score based on comparisons between disentangled and entangled representations.

We observed empirically that disentanglement is highly correlated to a match in shape between the generative parameter distribution $p(\theta)$ and the aggregated posterior $q_\phi(z)$ (**Section 5**). A match in the scaled-and-translated shapes results in good disentanglement but an aggregated posterior which does not match the shape of the generative parameter distribution or contains incorrect correlations ("rotated") relative to the generative parameter distribution does not. Using this knowledge, another disentanglement metric is postulated to compare these shapes by leveraging the KL Divergence (**Eq. 7**) where $\circ$ denotes the Hadamard product. The disentanglement score is given by

$$S_{KL} = \min_{a,b} D_{KL}\left[p(\theta)||q_\phi(a \circ (z - b))\right]. \tag{7}$$

This metric compares the shapes of the two distributions by finding the minimum KL divergence between the generative parameter distribution and a scaled and translated version of the aggregated posterior. When $q_\phi(a \circ (z - b))$ is close to $p(\theta)$ for some vectors $a, b \in \mathbb{R}^n$, disentanglement is observed.

It is noted in [16] that rotation of the latent space certainly has a large effect on disentanglement, which is precisely what we observe (**Section 5**). Additionally, the ELBO loss is unaffected by rotations of the latent space when using rotationally-invariant priors such as standard normal (**Appendix A**).

2.1.1 β-VAE

The β-VAE objective gives greater weighting to the regularization loss,

$$\mathcal{L}_{\beta-VAE} = \beta\mathbb{E}_{\hat{p}(y)} D_{KL}\left[q_\phi(z|y)||p(z)\right] + \mathbb{E}_{\hat{p}(y)}\left[\mathbb{E}_{q_\phi(z|y)}\left[-\log p_\psi(y|z)\right]\right].$$

This encourages greater regularization, often leading to improved disentanglement over the standard VAE loss [6]. It is worth noting that when $\beta = 1$, with a perfect encoder and decoder, the VAE loss reduces to the Bayes rule [23]; [24]. More details on the β-VAE are provided in **Section 2.2**.

2.2 Connections to Rate-Distortion Theory

Rate-distortion theory [25, 26, 27] aids in a deeper understanding in the trade off and balance between the regularization and reconstruction losses. The general rate distortion problem is formulated before making these connections. Consider two random variables: data $Y\colon \Omega \to \mathbb{R}^m$ and a compressed representation of the data $Z\colon \Omega \to \mathbb{R}^n$. An encoder $p(z|y)$ is sought such that the compressed representation contains a minimal amount of information about the data subject to a bounded error in reconstructing the data. A model $\tilde{y}(z)$ is used to reconstruct Y from samples of Z, and a distortion metric $d(y, \tilde{y})$ is used as a measure of error in the reconstruction of Y with respect to the original data.

A rate-distortion problem thus takes the general form

$$R(D) = \min_{p(z|y)} I(Y;Z) \quad \text{s.t. } \mathbb{E}_{Y,Z}\left[d(y, \tilde{y}(z))\right] \leq D, \tag{8}$$

where $D \in \mathbb{R}$ is an upper bound on the distortion. Solutions to **Eq. 8** consist of an encoder $p(z|y)$ which extracts as little information as possible from Y while maintaining a bounded distortion on the reconstruction of Y from Z through the model $\tilde{y}(z)$. Mutual information is minimized to obtain a maximally compressed representation of the data. Learning unnecessary information leads to "memorization" of some aspects of the data rather than extracting only the information relevant to the task at hand.This optimization problem formulated as the rate-distortion Lagrangian is

$$\min \mathcal{J}(\beta) = \min_{p(z|y)} I(Y;Z) + \beta \mathbb{E}_{Y,Z} d y, \tilde{y}(z) - D. \tag{9}$$

Given an encoder and decoder, solutions to the rate-distortion problem lie on a convex curve referred to as the rate-distortion curve [20]. Points above this curve correspond to *realizable* yet sub-optimal solutions. Points below the RD curve correspond to solutions which are *not* realizable; no possible compression exists with distortion below the RD curve. As the RD curve is convex, optimal solutions found by varying β lie along the curve. Increasing β increases the tolerable distortion, decreasing the mutual information between the compressed representation and data, providing a more compressed representation. Conversely, decreasing β requires a more accurate reconstruction of the data, increasing the mutual information between compressed representation and data.

The β-VAE loss is tied to a rate-distortion problem. Rearranging the VAE regularization loss ($\mathcal{L}_{REG}$), we obtain

$$\mathcal{L}_{REG} = \mathbb{E}_{\hat{p}(y)} D_{KL}\left[q_\phi(z|y)||p(z)\right] = I_\phi(Y;Z)$$

which is equal to the mutual information between Y and Z according to the data and encoding distributions. Minimizing the β-VAE loss gives the optimization problem

$$\min_{\phi,\psi} \mathcal{L}_{\beta-VAE} = \min_{\phi,\psi} I_\phi(Y;Z) + \beta \mathbb{E}_{\hat{p}(y)p_\psi(z|y)}\left[-\log p_\phi(y|z)\right].$$

This optimization problem is similar to a rate-distortion problem with $d(y, \tilde{y}) = -\log p_\phi(y|z)$ and the mutual information $I_\phi(Y; Z)$ just an approximation to the true mutual information $I(Y; Z)$. Depending on β, solutions can be found at any location along the RD curve with each containing differing properties. RD curve for VAEs is simply an analogy: $\mathcal{L}_{REG}$ is considered the rate R and $\mathcal{L}_{REC}$ is considered the distortion D.

With increased β, the β-VAE minimizes the mutual information between the data and the latent parameters, limiting reconstruction accuracy. In [16], disentanglement is illustrated to be caused inadvertently through the assumed factored form of the encoding distribution even though rotations of the latent space have no effect on the ELBO. However, their proof relies on training in the "polarized" regime characterized by loss of information or "posterior collapse" [28]. Training in this regime often requires increasing the weight of the regularization loss, necessarily decreasing reconstruction performance in the process. In our work, we illustrate disentanglement through training VAEs with the ELBO loss ($\beta = 1$), keeping reconstruction accuracy high. [16] presents good insights into disentanglement.

2.3 Hierarchical Priors

Often the prior (in the case of classic VAEs, specified by the user) and generative parameter distributions (data dependent) may not be highly correlated. Hierarchical priors [7] (HP) can be implemented within the VAE network such that the prior is learned as a function of additional random variables, potentially leading to more expressive priors and aggregated posteriors. Hierarchical random variables ξ_i are introduced such that "sub-priors" can be assumed on each ξ_i (typically standard normal). In the case of a single hierarchical random variable

$$\begin{aligned} p(z) &= \int_\Xi p(z|\xi)p(\xi)d\xi = \int_\Xi \frac{p(\xi|z)}{p(\xi|z)} p(z|\xi)p(\xi)d\xi \\ &= \mathbb{E}_{\Xi\sim p(\xi|z)}\left[\frac{p(z|\xi)p(\xi)}{p(\xi|z)}\right]. \end{aligned}$$

The conditional distributions $p(\xi|z)$ and $p(z|\xi)$ are the *prior encoder* and *prior decoder*, respectively. These distributions can be approximated by parameterizing them with neural networks. The parameterized distributions are noted as $q_\gamma(\xi|z)$ and $p_\pi(z|\xi)$ where γ are the trainable parameters of the approximating prior encoder and π are the trainable parameters of the prior decoder. Thus, the VAE prior can be approximated through the prior encoding and decoding distributions

$$p(z) \approx \mathbb{E}_{\Xi\sim q_\gamma(\xi|z)}\left[\frac{p_\pi(z|\xi)p(\xi)}{q_\gamma(\xi|z)}\right]. \tag{10}$$

Rearranging the VAE regularization loss

$$\begin{aligned} \mathcal{L}_{REG} &= \int_{Y,Z} \hat{p}(y) q_\phi(z|y) \log \frac{q_\phi(z|y)}{p(z)} dydz \\ &= \mathbb{E}_{Y,Z\sim\hat{p}(y)q_\phi(z|y)}\left[\log q_\phi(z|y)\right] \\ &\quad - \int_{Y,Z} \hat{p}(y) q_\phi(z|y) \log p(z) dydz, \end{aligned} \tag{11}$$

and substituting the approximating hierarchical prior **Eq. 10** into **Eq. 11**, the final term on the right-hand side becomes

$$-\int_{Y,Z}\hat{p}(y)q_\phi(z|y)\log p(z)dydz = -\int_{Y,Z}\hat{p}(y)q_\phi(z|y)\log \left[\mathbb{E}_{\Xi\sim q_\gamma(\xi|z)}\left[\frac{p_\pi(z|\xi)p(\xi)}{q_\gamma(\xi|z)}\right]\right]dydz.$$

The logarithm function is strictly concave; therefore, by Jensen's inequality the right-hand side is upper bounded by

$$-\int_{Y,Z}\hat{p}(y)q_\phi(z|y)\log\left[\mathbb{E}_{\Xi\sim q_\gamma(\xi|z)}\left[\frac{p_\pi(z|\xi)p(\xi)}{q_\gamma(\xi|z)}\right]\right]dydz \leq -\int_{Y,Z}\hat{p}(y)q_\phi(z|y)\mathbb{E}_{\Xi\sim q_\gamma(\xi|z)}\left[\log\frac{p_\pi(z|\xi)p(\xi)}{q_\gamma(\xi|z)}\right]dydz.$$

This bound is rearranged to the form

$$\mathbb{E}_{Y,Z\sim\hat{p}(y)q_\phi(z|y)}D_{KL}\left[q_\gamma(\xi|z)||p(\xi)\right] - \mathbb{E}_{Y,Z\sim\hat{p}(y)q_\phi(z|y)}\left[\mathbb{E}_{q_\gamma(\xi|z)}\left[\log p_\pi(z|\xi)\right]\right]. \tag{12}$$

Equation 12 takes the same form as the overall VAE loss, but applied to the prior network itself. Thus, the hierarchical prior can be thought of as a system of sub-VAEs within the main VAE. In summary, the VAE loss is upper bounded by

$$\mathcal{L}_{VAE} \leq \mathbb{E}_{Y,Z\sim\hat{p}(y)q_\phi(z|y)}\left[\log q_\phi(z|y)\right] + \mathbb{E}_{Y,Z\sim\hat{p}(y)q_\phi(z|y)}\left[D_{KL}\left[q_\gamma(\xi|z)||p(\xi)\right]\right] \tag{13}$$

$$-\mathbb{E}_{Y,Z\sim\hat{p}(y)q_\phi(z|y)}\left[\mathbb{E}_{q_\gamma(\xi|z)}\left[\log p_\pi(z|\xi)\right]\right] \tag{14}$$

$$-\mathbb{E}_{Y,Z\sim\hat{p}(y)q_\phi(z|y)}\left[\log p_\psi(y|z)\right]\Big]. \tag{15}$$

Implementing hierarchical priors can aid in learning non-rotationally-invariant priors, frequently inducing a learned disentangled representation, as shown below.

3 APPLICATION TO DARCY FLOW

To characterize the training process of the VAEs and to study disentanglement, we employ an application of flow through porous media. A two-dimensional steady-state Darcy flow problem in c spatial dimensions (our experiments employ $c = 2$) is governed by [29].

$$\begin{aligned} u(x) &= -K(x)\nabla p(x), \quad x\in\mathcal{X} \\ \nabla\cdot u(x) &= f(x), \quad x\in\mathcal{X} \\ u(x)\cdot\hat{n}(x) &= 0, \quad x\in\partial\mathcal{X} \\ \int_{\mathcal{X}} p(x)dx &= 0. \end{aligned} \tag{16}$$

Darcy's law is an empirical law describing flow through porous media in which the permeability field is a function of the spatial coordinate $K(x)$: $\mathbb{R}^c \to \mathbb{R}$. The pressure $p(x)$: $\mathbb{R}^c \to \mathbb{R}$ and velocity $u(x)$: $\mathbb{R}^c \to \mathbb{R}^c$ are found given the source term $f(x)$: $\mathbb{R}^c \to \mathbb{R}$, permeability, and boundary conditions. The integral constraint is given to ensure a unique solution.

A no-flux boundary condition is specified, and the source term models an injection well in one corner of the domain and a production well in the other

$$f(x) = \begin{cases} r, & |x_i - \frac{1}{2}w| \leq \frac{1}{2}w, \quad i = 1,2 \\ -r, & |x_i - 1 + \frac{1}{2}w| \leq \frac{1}{2}w, \quad i = 1,2 \\ 0, & \text{otherwise} \end{cases}, \tag{17}$$

where $w = \frac{1}{8}$ and $r = 10$. The computational domain considered is the unit square $\mathcal{X} = [0,1]^2$.

3.1 Karhunen-Loeve Expansion Dataset

The dataset investigated uses a log-permeability field modeled by a Gaussian random field with covariance function k

$$K(x) = \exp(G(x)), \quad G(\cdot) \sim \mathcal{N}(\bar{\mu}, k(\cdot,\cdot)). \tag{18}$$

Generating the data first requires sampling from the permeability field (**Eq. 18**). We take the covariance function as

$$k(x,x') = \exp\left(-||x-x'||_2/l\right) \tag{19}$$

in our experiments, as in [29]. After sampling the permeability field, solving **Eq. 16** for the pressure and velocity fields produces data samples. We discretize the spatial domain on a 65×65 grid and use a second-order finite difference scheme to solve the system.

The intrinsic dimensionality of the data will be the total number of nodes in system (4,225 for our system) [29]. For dimensionality reduction, the intrinsic dimensionality s of the data is specified by leveraging the Karhunen-Loeve Expansion (KLE), retaining only the first s terms in

$$G(x) = \bar{\mu} + \sum_{i=1}^{s}\sqrt{\lambda_i}\theta_i\phi_i(x), \tag{20}$$

where λ_i and $\phi_i(x)$ are eigenvalues and eigenfunctions of the covariance function (**Eq. 19**) sorted by decreasing λ_i, and each θ_i are sampled according to some distribution $p(\theta)$, denoted the *generative parameter distribution*.

Each dataset contains some intrinsic dimensionality s, and we denote each dataset using the permeability field (**Eq. 18**) as KLEs. For example, a dataset with $s = 100$ is referred to as KLE100. Samples from datasets of various intrinsic dimension are illustrated in **Figure 1**. Variations on the KLE2 dataset are employed for our explorations in this work. The differences explored are related to varying the generative parameter distribution $p(\theta)$ in each set.

Each snapshot $y^{(i)}$ from a single dataset $\{y^{(i)}\}_{i=1}^N$ contains the pressure $p(x)$ and velocity fields $u(x)$ at each node in the computational domain. These are used as a 3-channel input to the VAE; the permeability field and KLE expansion coefficients (generative parameters) are saved and used only for evaluation purposes.

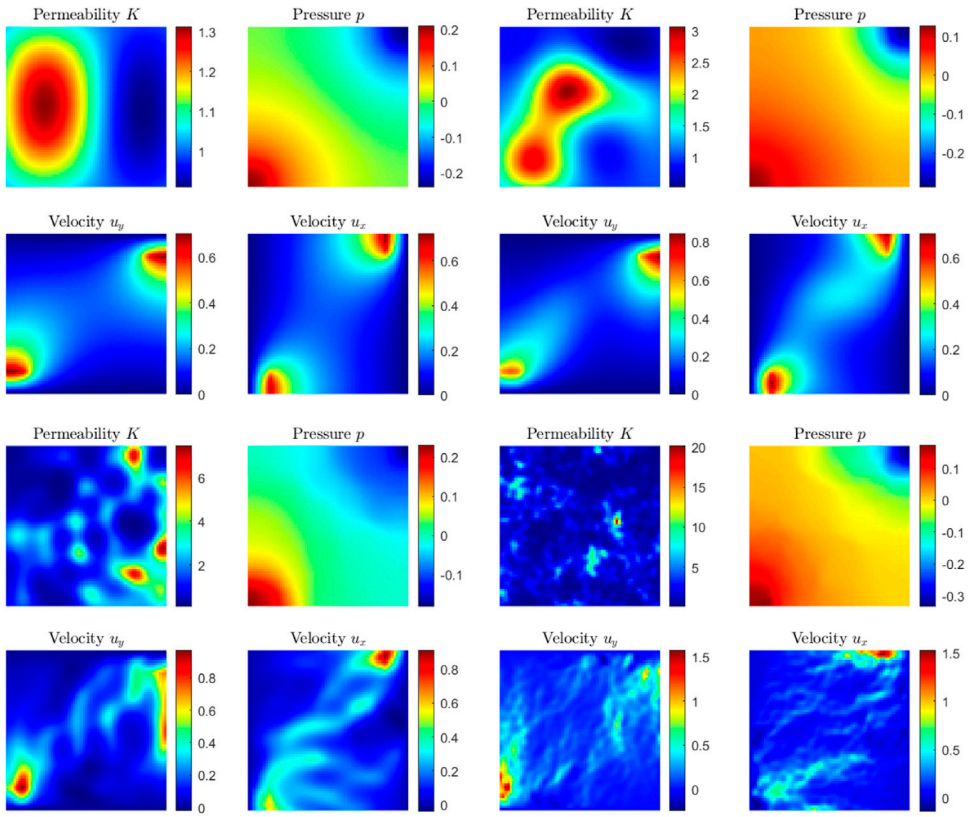


FIGURE 1 | Samples from datasets (*top left*) KLE2 (*top right*) KLE10 (*bottom left*) KLE100 (*bottom right*) KLE1000.

4 TRAINING SETUP AND LOSS LANDSCAPE

The process of training a VAE involves a number of challenges. For example, convergence of the optimizer to local minima can greatly hinder reconstruction accuracy and failure to converge altogether remains a possibility. A recurrent issue with VAE training in our experiments is that of over-regularization. Over-regularized solutions are characterized by disproportionately small regularization loss ($\mathcal{L}_{REG} \ll 1$). More information on this issue is detailed in **Section 4.2**.

To mitigate some of the issues inherent to training VAEs, we employ a training method tailored to avoid over-regularization. All experiments are performed using the Adam optimizer in Pytorch. We use $\mathcal{L}_{\beta-VAE}$ to train the models. The β value is varied with epochs, but at the end of training the model is converged with $\beta = 1$. The model is trained initially with $\beta_0 \ll 1$, typically around $\beta_0 = 10^{-7}$, for some number of epochs r_0 (depending on learning rates) until reconstruction accuracy is well below that of an over-regularized solution (**Section 3** illustrates this necessity). When β_0 is too small, the regularization loss can become too large, preventing convergence altogether. Training is continued by implementing a β scheduler [7] to slowly increase the weight of the regularization loss. The learning rate is then decreased to $lr_1 = c\,(lr_0)$ after some number of epochs r_1 to enhance reconstruction accuracy. This training method—in particular the heavily weighted reconstruction phase and the β scheduler—result in much more stable training which avoids the local minima characterized by over-regularization and improves convergence consistency. Similar methods have been employed to avoid this issue. In particular, [30] refers to this issue as "KL vanishing" and uses a cyclical β schedule to avoid the issue. However, this can take far more training epochs and cycle iterations to converge than the method employed here.

4.1 Architecture

The primary architecture for the VAE is adapted from [29] and a more detailed description including architecture optimization is given in the included **Supplementary Material**. This architecture consists of a series of encoding blocks to form the encoder, and a series of decoding blocks to form the decoder. Each encoding/decoding block consists of a dense block followed by and encoding/decoding layer. Contrary to the name, dense blocks do not contain any dense layers, but rather a series of skip connections and convolutional layers. Encoding and decoding layers consist of convolutions. The architecture is called DenseVAE and is used for all VAEs trained in this work. The latent and output distributions are assumed to be Gaussian. We use the dense block based architecture to parameterize the encoder mean and log-variance separately, as well as the decoder mean. The decoding distribution log-variance is learned but constant as introducing a learned output log-

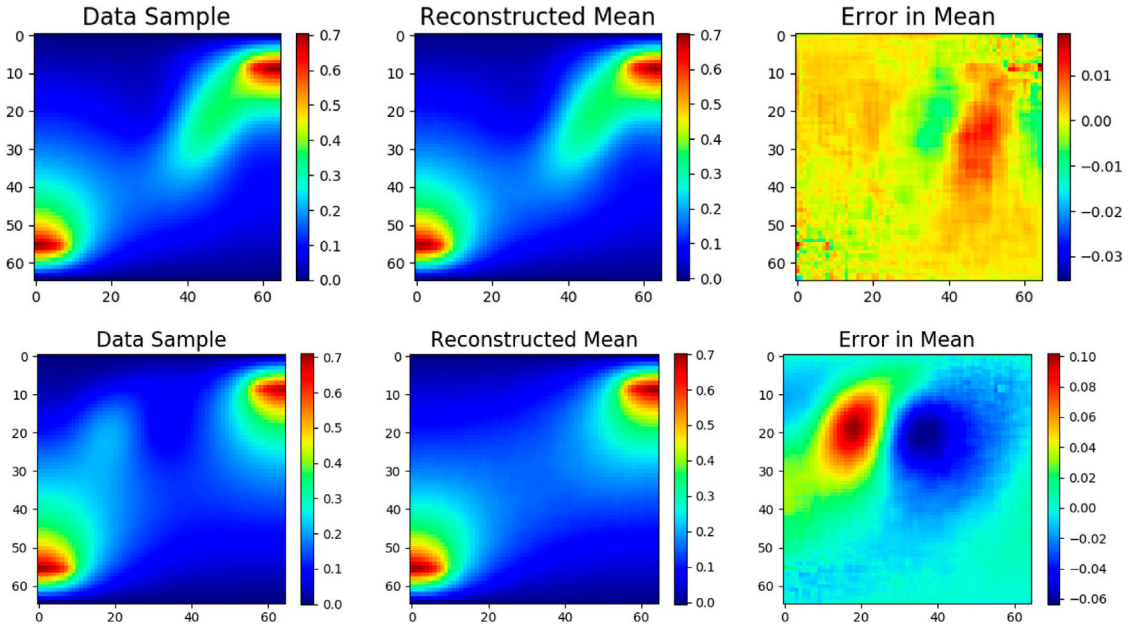


FIGURE 2 | (*upper*) Good reconstruction. (*lower*) Over-regularization.

variance did not aid in reconstruction or improving disentanglement properties in our experiments but increased training time.

4.2 Over-Regularization

Over-regularization has been identified as a challenge in the training of VAEs [30]. This phenomenon is characterized by the latent space containing no information about the data; i.e., the regularization loss becomes zero. The output of the decoder becomes identical accross all inputs. Thus, the output of the decoder is a constant distribution which does not depend on the latent representation. The constant distribution it learns becomes a normal distribution with mean and variance of the data. With zero regularization loss, the learned decoding distribution becomes $p_\psi(y|z) = \mathcal{N}(y; \hat{\mu}_y, \mathrm{diag}(\hat{\sigma}_y^2))\ \forall\ z$ where $\hat{\mu}_y = \frac{1}{N}\sum_{i=1}^{N} y^{(i)}$ and $\hat{\sigma}_y^2 = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - \hat{\mu}_y)^2$. This is proven to minimize $\mathcal{L}_{VAE}$ in Theorem 1. The solution is not shown to be unique, but our experiments indicate that this is the over-regularized solution found during training. As Theorem 1 illustrates validity for any encoder $q(z|y)$, this is the most robust solution for the VAE to converge to when over-regularization occurs. The decoder learns to predict as accurately as possible given nearly zero mutual information between the latent and data random variables. As the encoder and decoder are trained simultaneously, predicting a constant output regardless of z prevents the necessity of the decoder to adjust as the encoder changes. An empirical comparison between good reconstruction and over-regularization is shown in **Figure 2**.

Theorem 1 requires that the output variance is constant. Parameterizing the output variance with an additional network may aid in avoiding over-regularization.

THEOREM 1. : Given data $\{y^{(i)}\}_{i=1}^{N}$ and the VAE framework defined in **Section 2**, and assuming a decoding distribution of the form $p(y|z) = \mathcal{N}(y; \mu(z), \mathrm{diag}(\sigma^2))$, if $\mathcal{L}_{REG} = 0$, then $\arg\min_{\mu(z),\sigma^2} \mathcal{L}_{VAE} = \{\hat{\mu}_y, \hat{\sigma}_y^2\}$, where $\hat{\mu}_y = \frac{1}{N}\sum_{i=1}^{N} y^{(i)}$ and $\hat{\sigma}_y^2 = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - \hat{\mu}_y)^2$.

Proof: For any $q(z|y)$ s.t. $\mathcal{L}_{REG} = 0$:

$$\mathcal{L}_{VAE} = \mathcal{L}_{REC} = \mathbb{E}_{\hat{p}(y)q(z|y)}\left[-\log\left(p(y|z)\right)\right]$$
$$= \mathbb{E}_{q(z|y)}\left[\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{m}\frac{1}{2}\log(2\pi) + \log\left(\sigma_j\right) + \frac{1}{2\sigma_j^2}\left(y_j^{(i)} - \mu_j(z)\right)^2\right].$$

To minimize$\mathcal{L}_{VAE}$, take derivatives $\frac{\partial \mathcal{L}_{VAE}}{\partial \mu_j(z)}$ and $\frac{\partial \mathcal{L}_{VAE}}{\partial \sigma_j}$ (assuming derivative and expectation can be interchanged), where $j \in \{1, \ldots, m\}$:

$$\frac{\partial \mathcal{L}_{VAE}}{\partial \mu_j(z)} = \mathbb{E}_{q(z|y)}\left[-\frac{1}{N}\sum_{i=1}^{N}\frac{1}{\sigma_j^2}\left(y_j^{(i)} - \mu_j(z)\right)\right] = 0.$$

Thus, $\mathbb{E}_{q(z|y)}\left[\sum_{i=1}^{N} y_j^{(i)} - \mu_j(z)\right] = 0$ and

$$\mathbb{E}_{q(z|y)}\left[\mu_j(z)\right] = \frac{1}{N}\sum_{i=1}^{N} y_j^{(i)}. \tag{21}$$

Eq. 21 holds $\forall z, j$ if

$$\mu_j(z) = \hat{\mu}_j = \frac{1}{N}\sum_{i=1}^{N} y_j^{(i)}. \tag{22}$$

Taking the derivative w.r.t. variance, we have $\frac{\partial \mathcal{L}_{VAE}}{\partial \sigma_j} = \mathbb{E}_{q(z|y)}\left[\frac{1}{N}\sum_{i=1}^{N}\frac{1}{\sigma_j} - \frac{1}{\sigma_j^3}(y^{(i)} - \mu_j(z))^2\right] = 0$, and rearranging, we have

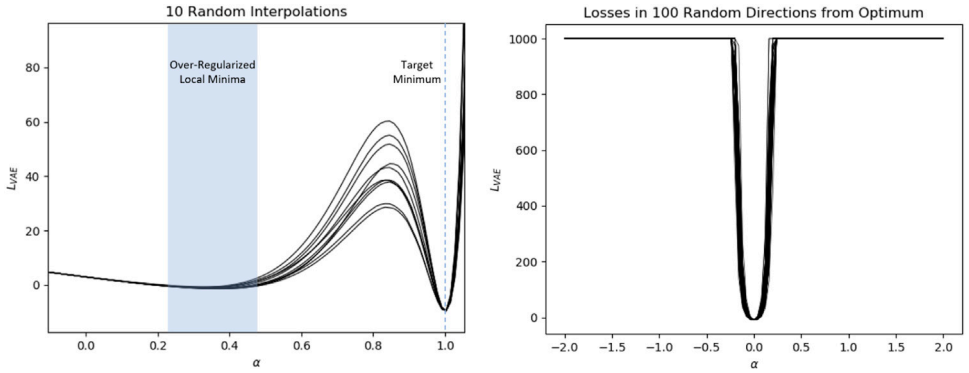


FIGURE 3 | (*left*) Loss along interpolated lines between 10 random weight initializations and a desirable converged solution. (*right*) Loss along 100 (of 1,000) random lines emanating from a desirable solution of the DenseVAE architecture. The parameter α indicates the distance along each random direction in parameter space and does not necessarily correspond to the same parameter α in the left figure. Note that the loss is limited to 1,000 for illustration purposes.

$$\sigma_j^2 = \frac{1}{N}\sum_{i=1}^{N}\left(y_j^{(i)} - \mu_j(z)\right)^2 \quad \forall \; z, j. \tag{23}$$

Substituting **Eq. 21** into **Eq. 23** results in:

$$\hat{\sigma}_j^2 = \frac{1}{N}\sum_{i=1}^{N}\left(y_j^{(i)} - \hat{\mu}_j\right)^2 \; \forall \; z, \; j. \tag{24}$$

With **Eqs 21**, **24** valid for all z and j, we can combine them into vector form and note that **Eq. 25** minimizes $\mathcal{L}_{VAE}$ as required.

$$\hat{\mu}_y = \begin{bmatrix} \hat{\mu}_1 \\ \vdots \\ \hat{\mu}_m \end{bmatrix}, \quad \hat{\sigma}_y^2 = \begin{bmatrix} \hat{\sigma}_1^2 \\ \vdots \\ \hat{\sigma}_m^2 \end{bmatrix}. \tag{25}$$

There exists a region in the trainable parameter loss landscape characterized by over-regularized local minimum solutions which partially surrounds the "desirable" solutions characterized by better reconstruction accuracy and latent properties. This local minima region is often avoided by employing the training method discussed previously, but random initialization of network parameters and changes in hyperparameters between training can render it difficult to avoid convergence to this region.

We illustrate the problem of over-regularization by training VAEs using the architecture described in **Section 4.1** on the KLE2 Darcy flow dataset with $p(\theta)$ being standard normal.

A VAE is trained with 512 training samples (each sample is $65 \times 65 \times 3$), converging to a desirable solution with low reconstruction error and nearly perfect disentanglement. The parameters of this trained network are denoted P_T. After the VAE is trained and a "desirable" solution obtained, 10 additional VAEs with identical setup to the desirable solution are initialized randomly using the Xavier uniform weight initialization on all layers. Each of the 10 initializations contain parameters P_i. A line in the parameter space is constructed between the converged "desirable" solution and the initialized solutions as a function of α:

$$P(\alpha) = (1-\alpha)P_i + \alpha P_T. \tag{26}$$

Losses are recorded along each of the 10 interpolated lines and plotted in **Figure 3**. Between the random initializations and "desirable" converged solutions there exists a region of local minima in the loss landscape, and these local minima are characterized by over-regularization. Losses illustrated are computed as an expectation over all training data and a Monte Carlo estimate of the reconstruction loss with 10 latent samples to limit errors due to randomness.

We include an illustration of the avoidance of the avoidance of these over-regularized local minima using our training method in the **Supplementary Material**.

Of interest is that this over-regularized local minima region does not fully surround the "desirable" region. Instead of interpolating in parameter space between random initializations and a converged solution, lines emanating away from the converged solution along 1,000 random directions in parameter space are created and the loss plotted along each. **Figure 3** illustrates that indeed no local minima are found around the converged solution. We note that there are around 800,000 training parameters in this case, so 1,000 random directions may not completely encapsulate the loss landscape around this solution.

The Xavier uniform weight initialization scheme, and most other initialization schemes, limit the norm of the parameters in parameter space to near the origin. The local minima region exists only between the converged solution region and points in parameter space near the origin. In this case, there may be alternative initialization schemes which can greatly aid in the convergence of VAEs. This has been observed in [31] where the initialization scheme proposed greatly accelerates the speed of convergence *and* accuracy of reconstruction.

Over-regularized local minima follow a similar path during training as desirable solutions. A region of attraction exists in the loss landscape, and falling too close to this region will result in an over-regularized solution, illustrated in **Figure 4**. One VAE which obtains a desirable solution shares a similar initial path with an over-regularized solution. Plotted are the VAE losses computed

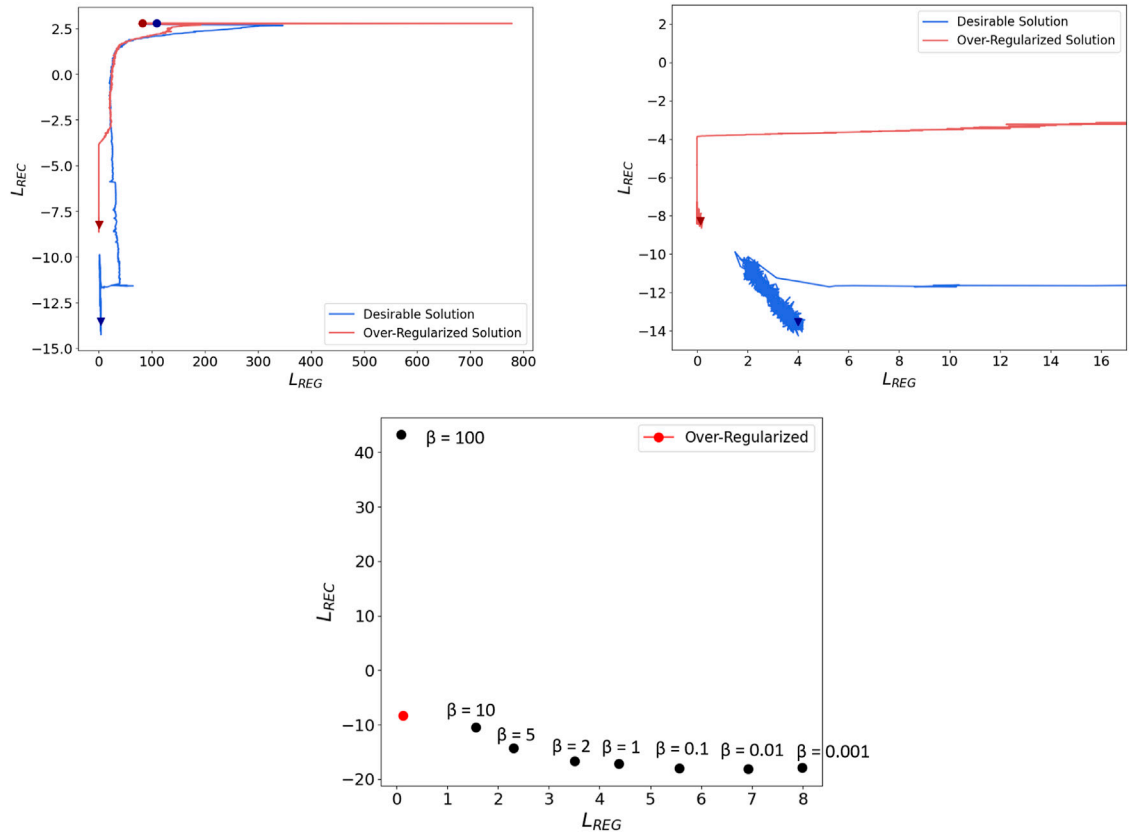


FIGURE 4 | RD plane illustrating training convergence of both desirable and over-regularized solutions to the RD curve ($\beta = 1$). (*right*) Scale adjusted. (*lower*) RD plane with points corresponding (from left to right) to $\beta = [100, 10, 5, 2, 1, 0.1, 0.01, 0.001]$. Many values of β between 5 and 100 fall into the over-regularized solution.

during training, not the training losses. The over-regularized solution breaks from the desired path too early, indicating a necessity for a longer reconstruction-heavy phase.

Training many VAEs with various β values facilitates a visualization of over-regularization in the RD plane. Each point in **Figure 4** shows the loss values of converged VAEs trained with different values of β. The over-regularized region of attraction prevents convergence to desirable solutions for many values of β. Interpolating in parameter space between each of these points (corresponding to a VAE with its own converged parameters) using the base VAE loss ($\beta = 1$), no other points on the RD curve are local minima of the VAE loss. In **Figure 4**, we observe that during training, the desirable solution reaches the RD curve but continues toward the final solution.

4.3 Properties of Desirable Solutions

Avoiding over-regularization aids in convergence to solutions characterized by low reconstruction error. Among solutions with similar final loss values, inconsistencies remain in latent properties. Two identical VAEs initialized separately often converge to similar loss values, but one may exhibit disentanglement while the other does not. This phenomenon is also explored in [18] and [16]. Two VAEs are trained with identical architectures, hyperparameters, and training method; they differ only in the random initialization of network parameters P. We denote the optimal network parameters found from one initialization as P_1 and optimal network parameters found from a separate initialization P_2. The losses for each converged solution are quite similar ($L_{VAE_1} \approx -9.50$, $L_{VAE_2} \approx -9.42$); however, disentanglement properties of each are dramatically different. We interpolate between these two solutions in parameter space (**Eq. 26**) and record losses and disentanglement scores along the line (**Figure 5**). The first network contains a nearly perfectly disentangled latent representation while the second network does not produce a disentangled representation. It is evident that multiple local minima exist in parameter space which converge to similar values in the loss landscape, but contain very different latent correlations. Local minima exist throughout the loss landscape, and with each initialization, a different local minimum may be found. Many such differing solutions are found throughout our experiments. This phenomenon is partially due to invariance of the ELBO to rotations of the latent space when using rotationally invariant priors. Disentanglement is heavily dependent on a factorized representation of the latent representation. With rotations not affecting the training loss, learning a disentangled representation seems to be somewhat random in this case.

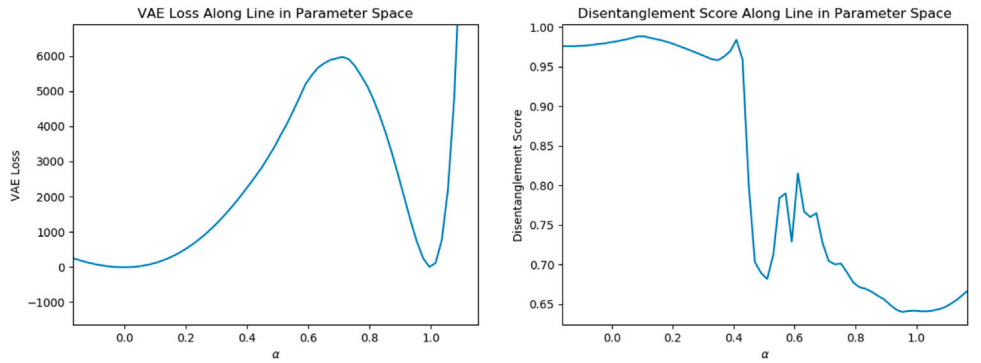


FIGURE 5 | (*left*) Loss variation along a line in parameter space between two converged solutions containing identical hyperparameters and training method but different network parameter initializations. (*right*) Disentanglement score along the same line.

This phenomenon exhibits the difficulties in disentangling generative parameters in an unsupervised manner; without prior knowledge of the factors of variation, conclusions cannot be drawn regarding disentanglement by observing loss values alone. In controlled experiments, knowledge of the underlying factors of variation is available, but when only data is available, full knowledge of such factors is often not. It is encouraging that the VAE does have the power to disentangle generative parameters in an unsupervised setting, but the nature of disentanglement must first be understood better to create identifying criterion.

5 CHARACTERIZING DISENTANGLEMENT

In this section, we explore the relationship between disentanglement, the aggregated posterior ($q_\phi(z)$), and the generative parameter distribution ($p(\theta)$) by incrementally increasing the complexity of $p(\theta)$. Disentanglement is first illustrated to be achievable but difficult using the classic VAE assumptions and loss due to a lack of enforcement of the rotation of the latent space caused by rotationally-invariant priors. Hierarchical priors are shown to aid greatly in disentangling the latent space by learning non-rotationally-invariant priors which enforce a particular rotation of the latent space through the regularization loss.

5.1 Standard Normal Generative Distributions

The intrinsic dimensionality of the data is set to $p = 2$ with a generative parameter distribution $p(\theta) = \mathcal{N}(\theta; 0, I_{2\times 2})$, the standard normal distribution. Limiting p to 2 aids greatly in the visualization of the latent space and understanding of the ideas investigated. The standard latent prior is identical to the generative parameter in this case, creating a relatively simple problem for the VAE.

Using the architecture described in **Section 4.1**, the relationship between regularization, reconstruction, and disentanglement and the number of training samples is illustrated in the included **Supplementary Material**. A similar study is performed in [18] with a greater sample size. Reconstruction losses continue to fall with the number of training data, indicating improved reconstruction of the data with increased number of samples; however, the regularization loss increases slightly with the number of training data. With too few samples, reconstruction performance is very poor and over-regularization (near zero regularization loss) seems unavoidable. Clear and consistent correlations exist among the loss values and number of training data, but disentanglement properties vary greatly among converged VAEs (**Section 4.3**). The compressed representations range from nearly perfect disentanglement to nearly completely entangled.

Although disentanglement properties are inconsistent between experiments, desirable properties of disentanglement are often observed. Training is performed using the maximum amount of available data (512 snapshots), and analysis included for 512 testing samples on the KLE2 dataset (regardless of $p(\theta)$). Regularization loss is large during the reconstruction phase in which $\beta_0 = 10^{-7}$, and the y-axis is truncated for clarity. A comparison between a test data sample and the reconstructed mean using the trained VAE is depicted in **Figure 6**, showing little error between the mean $\mu_\psi(z)$ of the decoding distribution and the input data sample. With small reconstruction error, a disentangled latent representation is learned. **Figure 6** also illustrates the aggregated posterior matching the prior distribution in shape. This is unsurprising with a generative parameter and prior distribution match and an expressive network architecture. Finally, **Figure 7** shows the correlation between the generative parameters of the training and testing data against the latent distribution as a qualitative measure of disentanglement. Each latent dimension is tightly correlated to a single but different generative parameter. **Figure 7** also illustrates the uncertainty in the latent parameters, effectively $q_\phi(z|\theta)$. The latent representation is fully disentangled; each latent parameter contains only information about a single generative factor.

5.2 Non Standard Gaussian Generative Distributions

The generative parameter distribution and the prior are identical (independent standard normal) in the previous example. Most

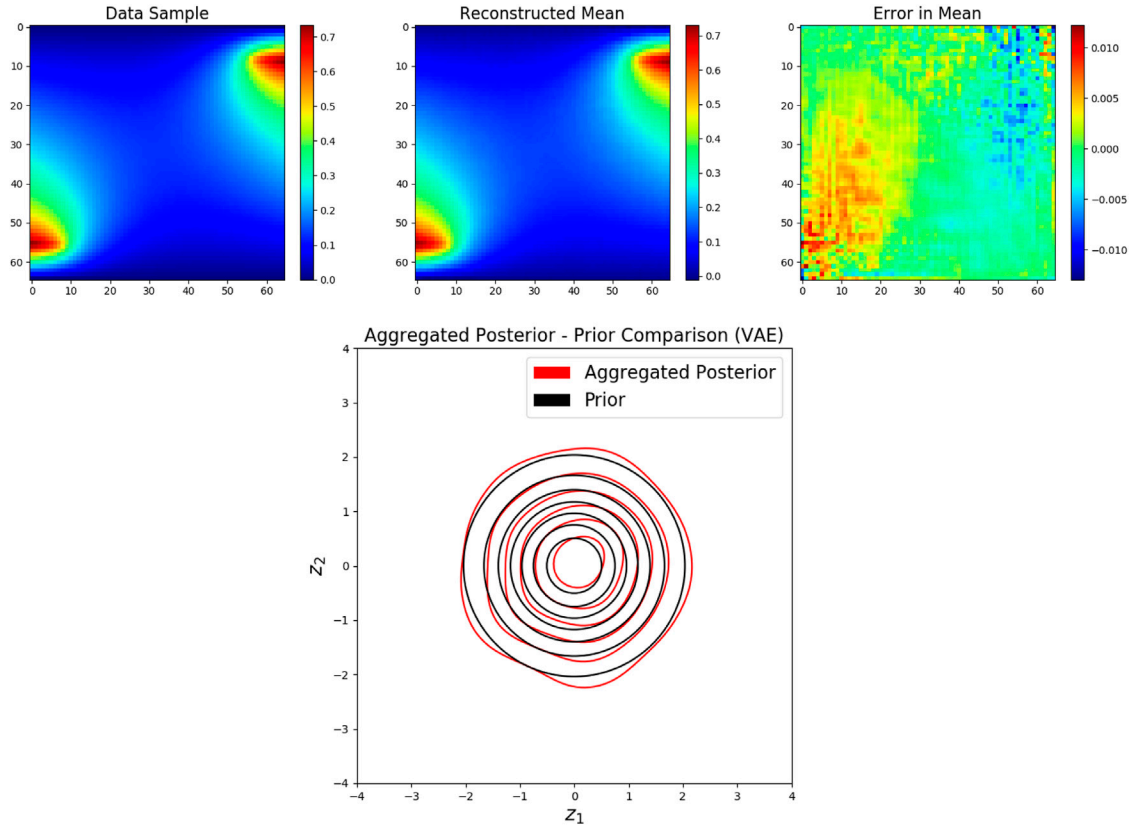


FIGURE 6 | (*left*) Data sample from unseen testing dataset. (*center*) Reconstructed data sample from trained VAE. (*right*) Error in the reconstruction mean. (*lower*) Comparison of aggregated posterior ($p_\phi(z)$) and prior ($p(z)$) distributions.

often, however, knowledge of the generative parameters is not possessed. The specified prior in this case is unlikely to match the generative parameter distribution. The next example illustrates the application of a VAE in which the generative parameter distribution and prior do not match. Another KLE2 dataset is generated with a non standard Gaussian generative parameter distribution. The generative parameter distribution is Gaussian, but scaled and translated relative to the previous example $p(\theta) = \mathcal{N}(\theta; [1; 1], [0.5, 0; 0, 0.5])$.

Training a standard VAE on this dataset results in high reconstruction accuracy, but undesirable disentanglement after many trials. With the use of an additional hierarchical prior network, good disentanglement can be achieved even with a mismatch in the prior and generative parameter distributions. The sub-prior (see **Section 2.3**) is the standard normal distribution, but the hierarchical network learns a non-standard normal prior. Still, the learned prior and generative parameter distributions do not match. **Figures 8**, **9** illustrate comparisons in results obtain from the VAE with and without the hierarchical prior network. When using hierarchical priors, the learned prior and aggregated posterior match reasonably well but do not match the generative parameter distribution. However, this does not matter as long as the latent representation is not *rotated* relative to the generative parameter distribution, as illustrated in the next example. Low reconstruction error and disentanglement are observed using hierarchical priors, but disentanglement was never observed using the standard VAE after many experiments. This may be because β is not large enough to enforce a regularization loss large enough to produced an aggregated posterior aligned with the axes of the generative parameter distribution. Therefore, the rotation of the learned latent representation will be random and disentanglement is unlikely to be observed, even in two dimensions. The hierarchical network consistently enforces a factorized aggregated posterior, which is essential for disentanglement when generative parameters are independent. One potential cause of this is the learning of non-rotationally-invariant priors, such as a factorized Gaussian with independent scaling in each dimension. The ELBO loss in this case *is* affected by rotations of the latent space, aligning the latent representations to the axes of the generative parameters.

A latent rotation can be introduced such that the reconstruction loss is unaffected, but regularization loss changes with rotation. Introducing a rotation matrix A with angle of rotation ω to rotate the latent distribution, the encoding distribution becomes $q_\phi(z|y) = \mathcal{N}(z; A\mu_\phi(y), A\text{diag}(\sigma_\phi(y))A^T)$. Reversing this rotation when computing the decoding distribution (i.e., $p_\psi(y|z) = \mathcal{N}(y; \mu_\phi(A^T z), \text{diag}(\sigma_\psi(A^T z)))$) preserves the reconstruction loss. However, the regularization loss can be

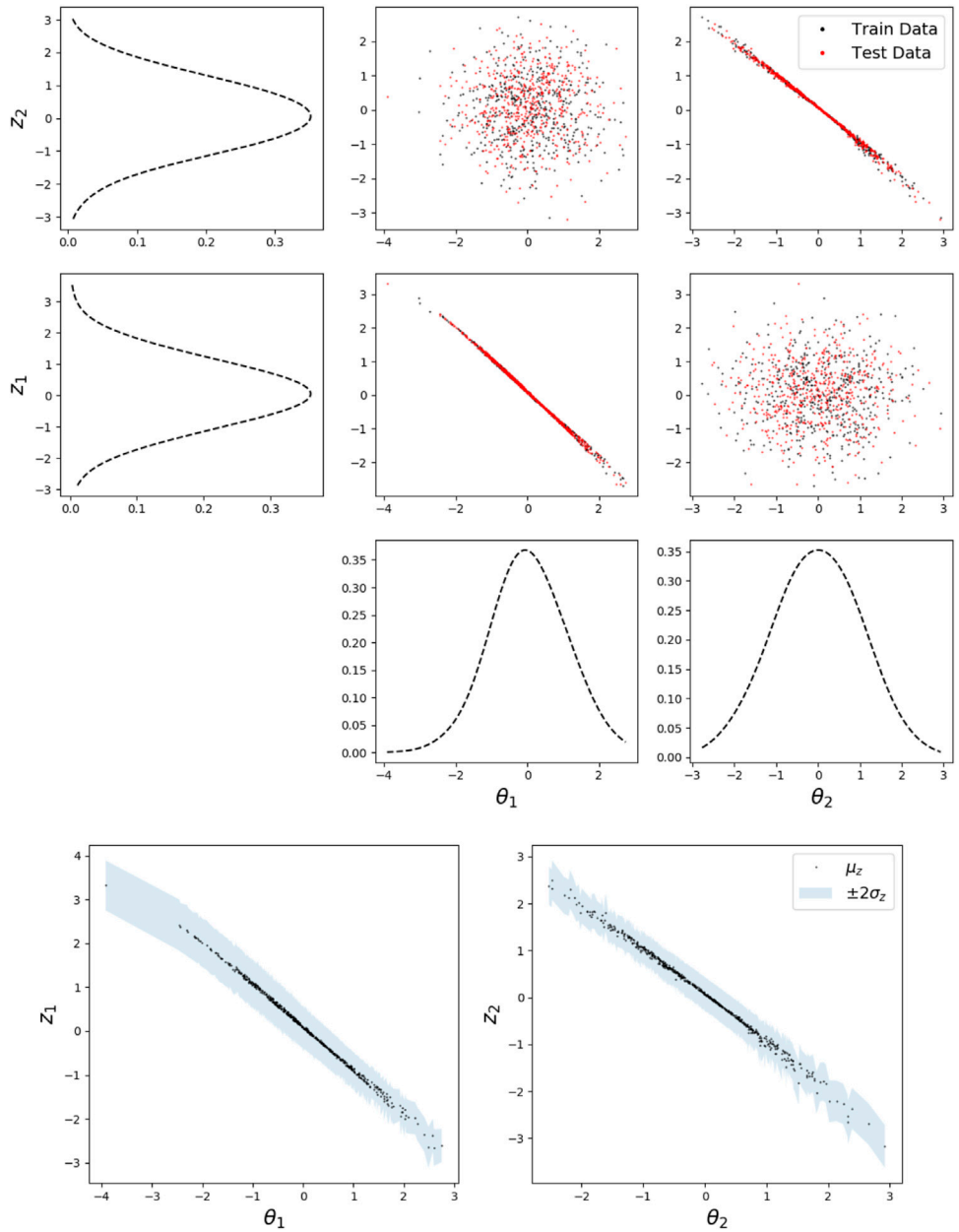


FIGURE 7 | (*upper*) Correlations between dimensions of generative parameters and mean of latent parameters. Also shown are the empirical marginal distributions of each parameter. (*lower*) Correlations between generative parameters and latent parameters with uncertainty for test data only.

plotted as a function of the rotation angle (included in the **Supplementary Material**). When a rotationally-invariant prior is used to train the VAE, regularization loss is unaffected by latent rotation. However, when the prior is non-rotationally-invariant, the regularization loss is affected by latent rotation. Thus, rotation of the latent space is enforced by the prior during training.

Although the hierarchical prior adds some trainable parameters to the overall architecture, the increase is only 0.048%. This is negligible, and it is assumed that this is not the root cause of improved disentanglement. Rather, it is the ability of the additional hierarchical network to consistently express a factorized aggregated posterior and learn non-rotationally-invariant priors which improves disentanglement. More insights are offered in the next example and **Section 7**.

5.3 Multimodal Generative Distributions

In this setup, disentanglement not only depends on a factorized $q_\phi(z)$, but the correlations in $p(\theta)$ must be preserved as well, i.e., rotations matter. The previous example illustrates a case in which the standard VAE fails in disentanglement but succeeds with the addition of hierarchical priors due to improved enforcement of a factorized $q_\phi(z)$ through learning non-

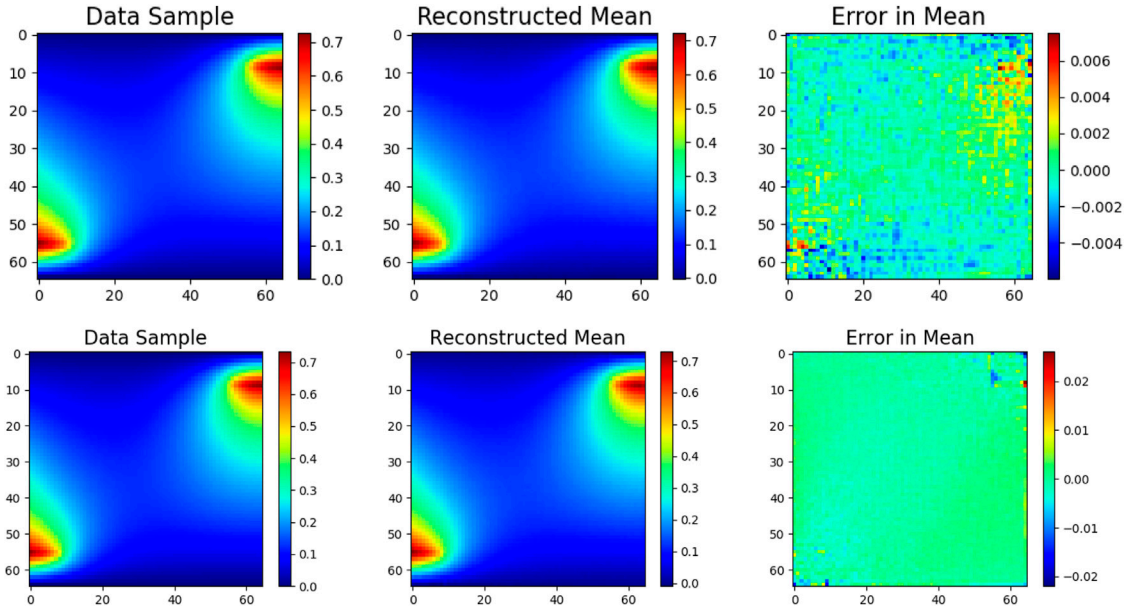


FIGURE 8 | (*top*) Reconstruction accuracy of a test sample on trained VAE without hierarchical network, (*bottom*) with hierarchical network.

rotationally-invariant priors. The generative parameter distribution is radially symmetric, thus visualization of rotations in $q_\phi(z)$ relative to $p(\theta)$ is difficult. To illustrate the benefits of using hierarchical priors for disentanglement, the final example uses data generated from a more complex generative parameter distribution with four lines of symmetry for better visualization. The generative parameter distribution is multimodal (a Gaussian mixture) and is more difficult to capture than a Gaussian distribution, but allows for better rotational visualization:

$$\begin{aligned} p(\theta) = &\frac{1}{4}\mathcal{N}(\theta; [-1;-1], [0.25, 0; 0, 0.25]) \\ &+ \frac{1}{4}\mathcal{N}(\theta; [1;1], [0.25, 0; 0, 0.25]) \\ &+ \frac{1}{4}\mathcal{N}(\theta; [-1;1], [0.25, 0; 0, 0.25]) \\ &+ \frac{1}{4}\mathcal{N}(\theta; [1;-1], [0.25, 0; 0, 0.25]). \end{aligned}$$

Training VAEs without hierarchical priors results in over-regularization more often than with the implementation of HP. Out of 50 trials, 10% trained without HP were unable to avoid over-regularization while all trials with HP successfully avoided over-regularization. More epochs are required in the reconstruction only phase (with and without HP) to avoid over-regularization than in previous examples. Disentanglement was never observed without the use of hierarchical priors. This again is due to rotation of the latent space relative to the generative parameter distribution due to rotationally invariant priors. To illustrate this concept, **Figure 10** illustrates the effects of rotation of the latent space on disentanglement. Clearly, rotation dramatically impacts disentanglement, and the standard normal prior does not enforce any particular rotation of the latent space.

Implementing hierarchical priors, consistent observation of not only better reconstruction (avoiding over-regularization) but also reasonable disentanglement of the latent space in roughly half of all trained VAEs (out of 50) exemplifies the improved ability of hierarchical priors to produce a disentangled latent representation. Reconstruction of test samples is more accurate when implementing the hierarchical prior network, as illustrated in **Figure 11**. We hypothesize that disentanglement is observed in roughly half of our experiments due to local minima in the regularization loss corresponding to 45° rotations of the latent space, illustrated in the **Supplementary Material**. The learned priors using HP are often non-rotationally-invariant and aligned with the axes. However, the posterior is often rotated 45-degrees relative to this distribution, creating a non-factorized and therefore non-disentangled representation.

Comparing $p(\theta)$, $p(z)$, and $q_\phi(z)$ with and without HP (**Figure 12**), stark differences are noticeable. Without the HP network, the aggregated posterior often captures the multimodality of the generative parameter distribution, but it is rotated relative to $p(\theta)$, creating a non-factorized $q_\phi(z)$. Training the VAE with hierarchical priors, the learned prior becomes non-rotationally invariant. The rotation of the aggregated posterior is therefore controlled by the orientation of the prior through the regularization loss, but mimics the shape of the generative parameter distribution. It is clear that the prior plays a significant role in terms of disentanglement: it controls the rotational orientation of the aggregated posterior.

A qualitative measure of disentanglement is compared in **Figure 12**. Without HP, the latent parameters are entangled; they are each weakly correlated to both of the generative parameters. Adding HP to the VAE results in disentanglement

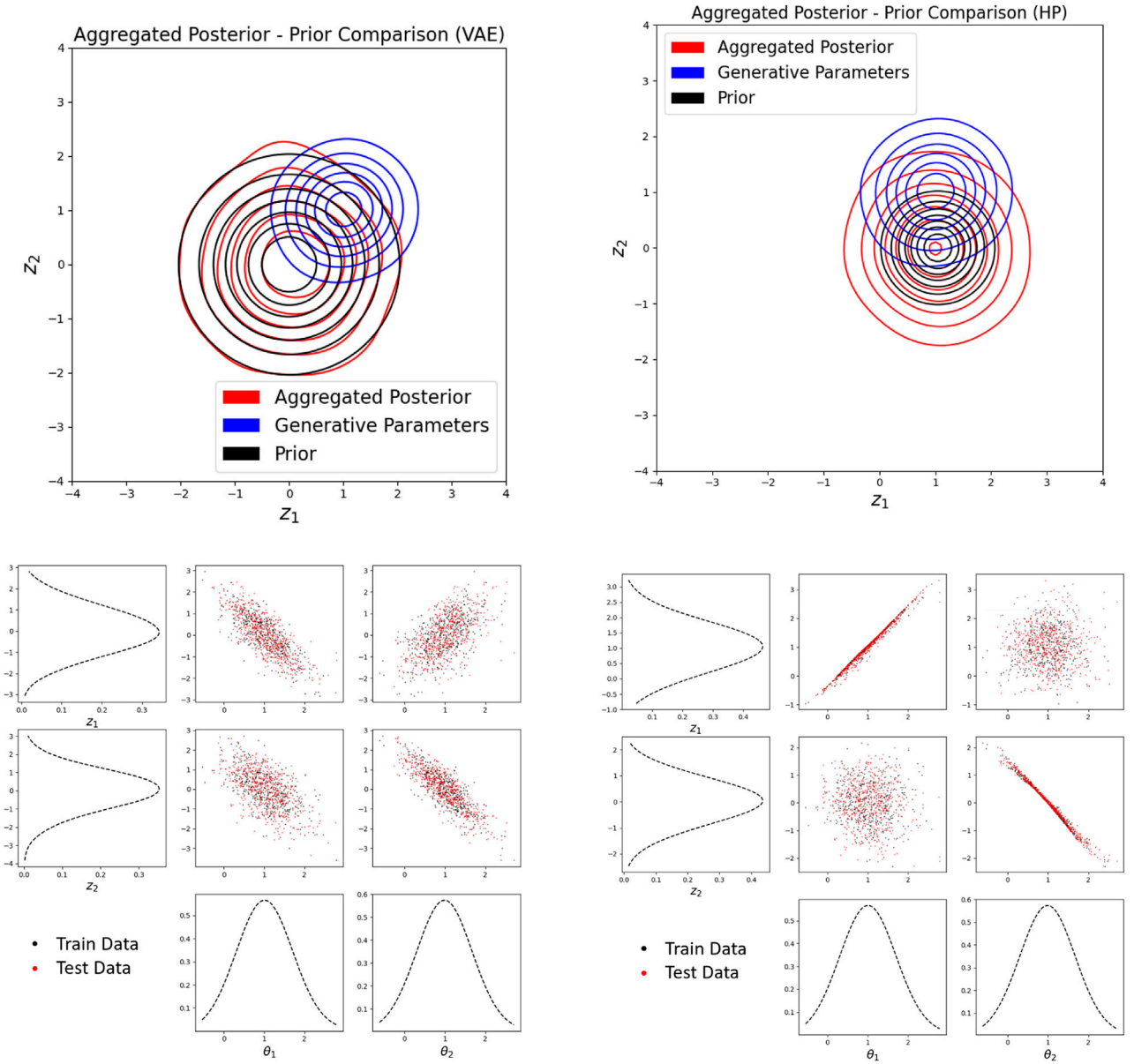


FIGURE 9 | (*upper left*) Aggregated poster, prior, and generative parameter distribution comparison on VAE without hierarchical network, (*upper right*) with hierarchical network. (*lower left*) Qualitative disentanglement in VAE trained without hierarchical network, (*lower right*) with hierarchical network.

in nearly half of our trials. When disentanglement does occur, each latent factor contains information on mostly a single but different generative factor. Through the course of our experiments, a relationship between disentanglement and the degree to which the aggregated posterior matches the generative parameter distribution is recognized. When disentanglement does not occur with the use of HP, the aggregated posterior is rotated relative to $p(\theta)$, or non-factorized (it has always been observed at around a 45-degree rotation). Only when $q_\phi(z)$ can be translated and scaled to better match $p(\theta)$, maintaining the correlations, does disentanglement occur. Thus, a quantitative measure of disentanglement (**Eq. 7**) is created from this idea. The KL divergence is estimated through sampling using the k-nearest neighbors (k-NN) approach (version $\epsilon 1$) found in [19]. The optimization is performed using the gradient-free Nelder-Mead optimization algorithm [32].

In low-dimensional problems, humans are adept at determining disentanglement from qualitative measurements of disentanglement such as **Figure 12**. It is, however, more difficult to obtain quantitative measurements of these properties. **Figure 13** shows the relationship between **Eq. 7** and a qualitative measurement of disentanglement. Lower values of S_{KL} indicate better disentanglement. This measure of disentanglement and

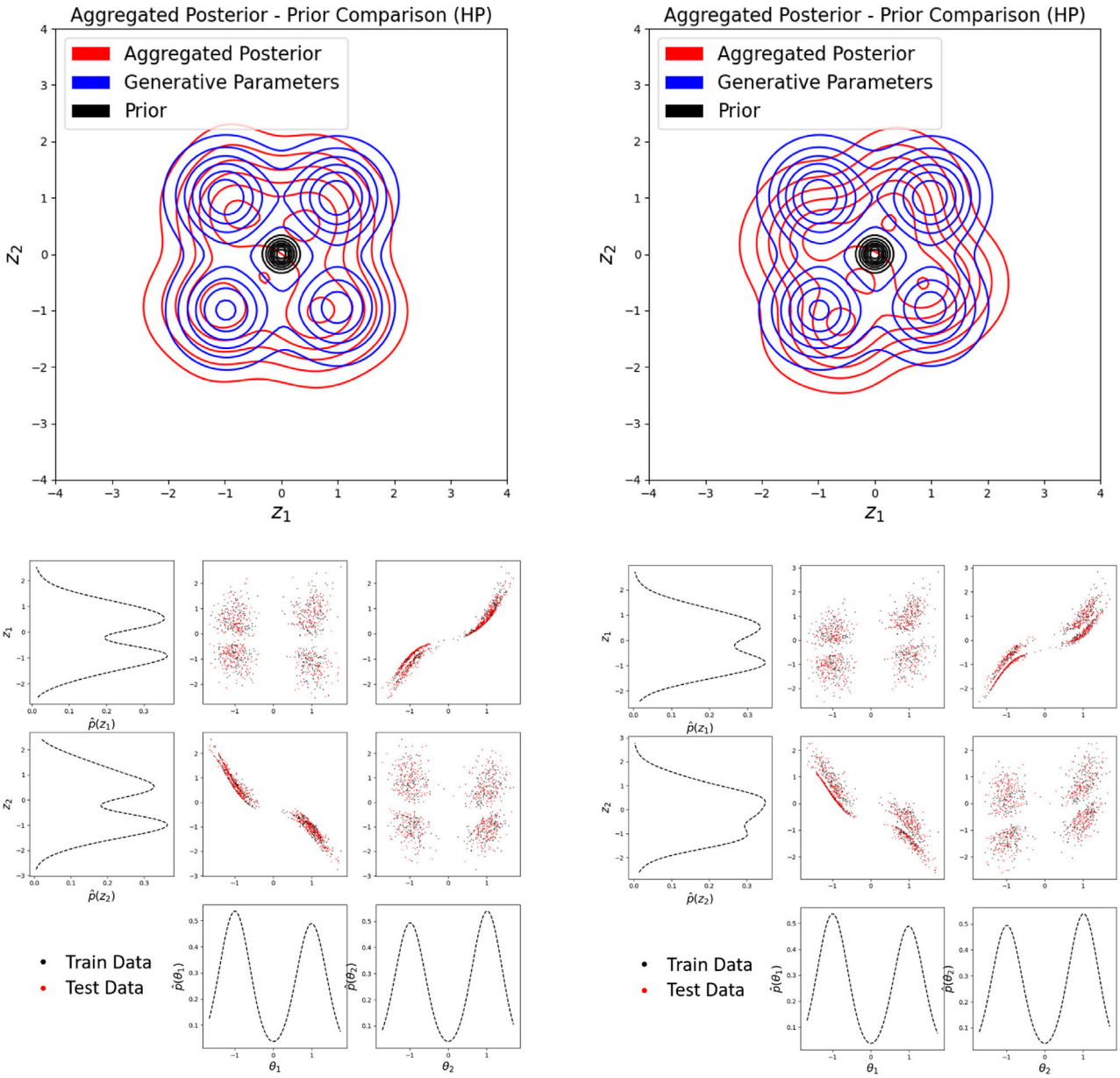


FIGURE 10 | (*top*) aggregated posterior comparison showing rotation of the latent space, (*bottom*) worse disentanglement when latent space is rotated.

the intuition behind it is discussed further in the conclusions section.

6 SEMI-SUPERVISED TRAINING

Difficulties with consistently disentangling generative parameters have been illustrated up to this point with an unsupervised VAE framework. In some cases, however, generative parameters may be known for some number of samples, suggesting the possibility of a semi-supervised approach. These labeled samples can be leveraged to further improve the consistency of learning a disentangled representation. Consider data consisting of two partitions: labeled data $\{y^{(i)}, \theta^{(i)}\}_{i=1}^{l}$ and unlabeled data $\{y^{(i)}\}_{i=l+1}^{u+l}$. A one-to-one mapping between the generative parameters θ and the learned latent representation z is sought when disentanglement is desired. Thus, enforcing the latent representation to match the generative parameters for labeled data in a semi-supervised approach should aid in achieving our desired objective more consistently.

We begin the intuition behind a semi-supervised loss function by illustrating its connection to the standard ELBO VAE loss. One method of deriving the ELBO loss is to first expand the

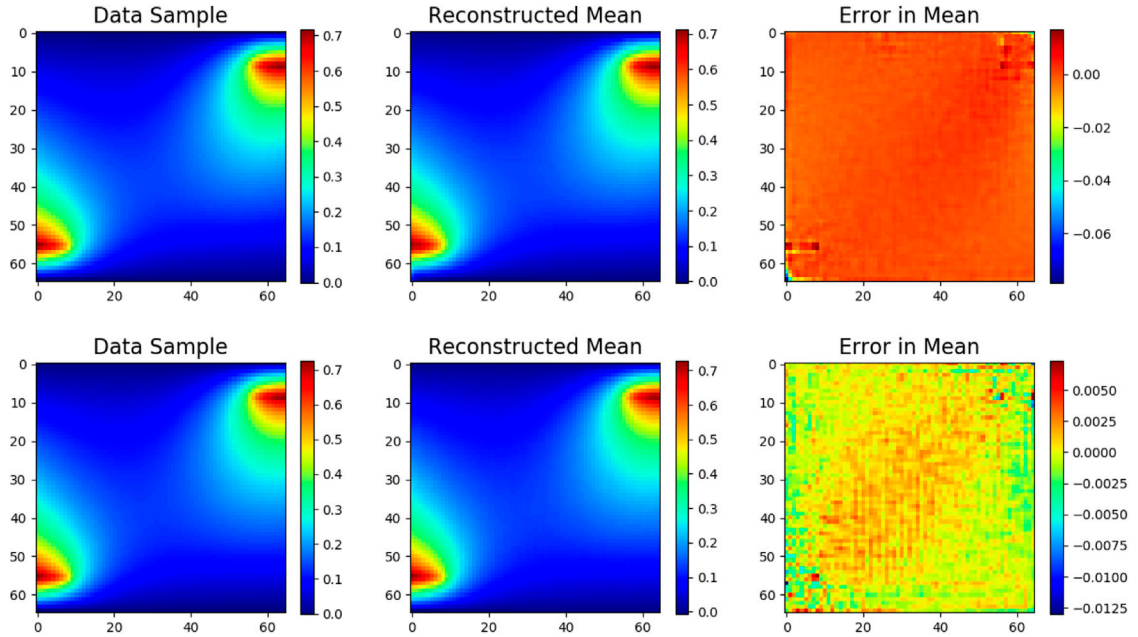


FIGURE 11 | (*top*) Reconstruction accuracy of a test sample using VAE trained on multimodal generative parameter distribution without hierarchical network, (*bottom*) with hierarchical network.

relative entropy between the data distribution and the induced data distribution to obtain

$$\begin{aligned} D_{KL}\left[p(y)||p_\psi(y)\right] = -H(Y) + \mathbb{E}_{p(y)}\left[D_{KL}\left[q_\phi(z|y)||p(z)\right]\right] \\ -\mathbb{E}_{p(y)}\left[D_{KL}\left[q_\phi(z|y)||p(z|y)\right]\right] \\ -\mathbb{E}_{p(y)q_\phi(z|y)}\left[\log p_\psi(y|z)\right] \end{aligned}$$

where $-H(Y)$ is constant and the "true" encoder $p(z|y)$ is unknown. Therefore, the term

$$\mathbb{E}_{p(y)}\left[D_{KL}\left[q_\phi(z|y)||p(z|y)\right]\right]$$

is usually ignored and we arrive at the ELBO, which upper bounds the left hand side. However, a relationship between z and y is known for labeled samples. This relationship can be used to assign $p\left(z^{(i)}|y^{(i)}\right)$ on the labeled partition. For unlabeled data, the standard ELBO loss is still used for training and the semi-supervised loss to be minimized becomes

$$\begin{aligned} \mathcal{L}_{VAE\text{-}SS}(\phi,\psi) = \mathbb{E}_{p(y)}\left[D_{KL}\left[q_\phi(z|y)||p(z)\right]\right] \\ -\mathbb{E}_{p_l(y)}\left[D_{KL}\left[q_\phi(z|y)||p(z|y)\right]\right] \\ -\mathbb{E}_{p(y)q_\phi(z|y)}\left[\log p_\psi(y|z)\right] \end{aligned} \tag{27}$$

where $p_l(y)$ is the distribution of inputs with corresponding labels, $p(y)$ is the distribution of all inputs (labeled and unlabeled), and $\mathbb{E}_{p_l(y)}[D_{KL}[q_\phi(z|y)||p(z|y)]]$ is denoted $\mathcal{L}_{SS}$.

Note that the term $\mathbb{E}_{p(y)}[D_{KL}[q_\phi(z|y)||p(z)] - D_{KL}[q_\phi(z|y)||p(z|y)]]$ is minimized by $q_\phi(z|y) = p(z|y)$ at $I(Z;Y)$, the mutual information between the generative parameters and high dimensional data. In unsupervised VAEs, the regularization term $\mathbb{E}_{p(y)}[D_{KL}[q_\phi(z|y)||p(y)]]$ is minimized when $q_\phi(z|y) = p(y)$. As observed in previous sections, disentanglement is observed when the aggregated posterior is "close" to the generative parameter distribution. With the semi-supervised loss being minimized when they are equivalent, the learned latent representations should be more easily and consistently disentangled.

However, empirically it is found that this loss is very sensitive to changes in network parameters and unreasonably small learning rates are required for stability. Additionally, there is no obvious way to determine the variance of $p\left(z^{(i)}|y^{(i)}\right)$ for each sample, only the mean is easily identifiable. We therefore propose to train with $\mathcal{L}_{SS} = \mathbb{E}_{p_l(y)}[-\log q_\phi(z|y)]$ instead such that the loss function becomes

$$\begin{aligned} \mathcal{L}_{VAE\text{-}SS}(\phi,\psi) = \mathbb{E}_{p(y)}\left[D_{KL}\left[q_\phi(z|y)||p(z)\right]\right] \\ -\mathbb{E}_{p_l(y)}\left[\log q_\phi(z|y)\right] \\ -\mathbb{E}_{p(y)q_\phi(z|y)}\left[\log p_\psi(y|z)\right]. \end{aligned} \tag{28}$$

Training with this loss achieves the desired outcome of consistently learning disentangled representations while being simple and efficient to implement.

Incorporating some labeled samples into training the VAE, a disentangled latent representation can be consistently learned. **Figure 14** illustrates the relationship between increasing the number of labeled samples and the disentanglement score of the learned latent representation. In each case, there are 512 unlabeled samples. Each trial varies in the number of labeled samples, and VAEs trained with the same number of labeled samples are trained with a different set of labeled samples. Ten VAEs are trained at each point, and the range illustrated

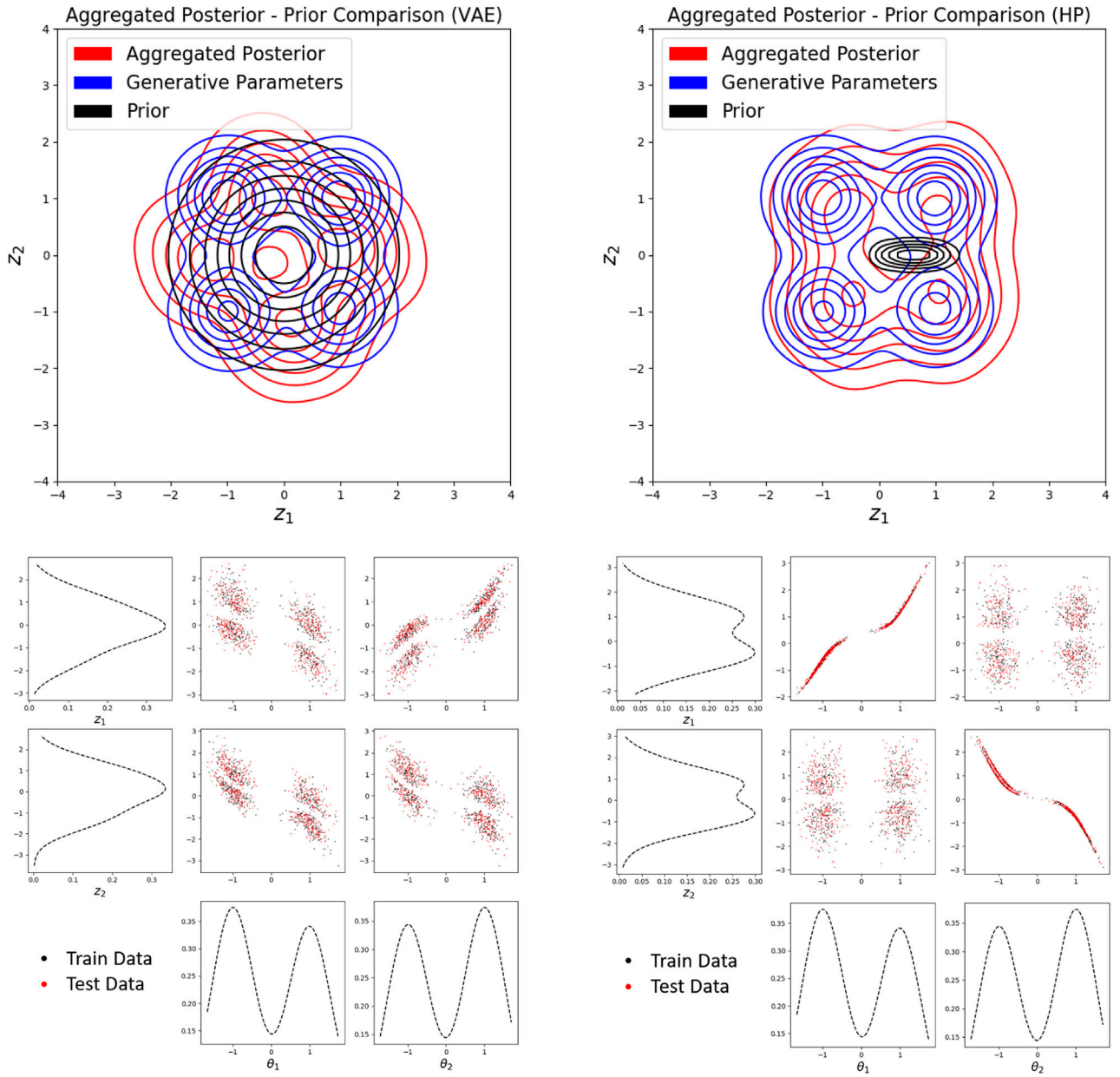


FIGURE 12 | (*upper left*) Aggregated posterior, prior, and generative parameter distribution comparison using VAE trained on multimodal generative parameter distribution without hierarchical network, (*upper right*) with hierarchical network. (*lower left*) Qualitative disentanglement using VAE trained on multimodal generative parameter distribution without hierarchical network, (*lower right*) with hierarchical network.

represents the maximum and minimum disentanglement score across the 10 trials.

The training losses do not seem to be effected by the number of labeled samples, only the disentanglement score is effected. With a low number of labeled samples, the semi-supervised VAE trains very similarly to the unsupervised VAE. That is, disentanglement is observed rather randomly, and the learned latent representation varies dramatically between trials. Labeling around 1% of the samples begins to result in consistently good disentanglement. Labeling between 3% and 8% results in learning disentangled latent representations which are nearly identical between trials. It follows from these results that disentangled representations can be consistently learned when training with **Eq. 28** when using a sufficient number of labeled samples (assuming a sufficiently expressive architecture).

Using a semi-supervised method also improves the ability of the VAE to predict data in regions of lower density. In **Figure 15**, we observe that the aggregated posterior matches the generative parameter distribution much better than the unsupervised case with just over 1% of the samples labeled. Additionally, regions of

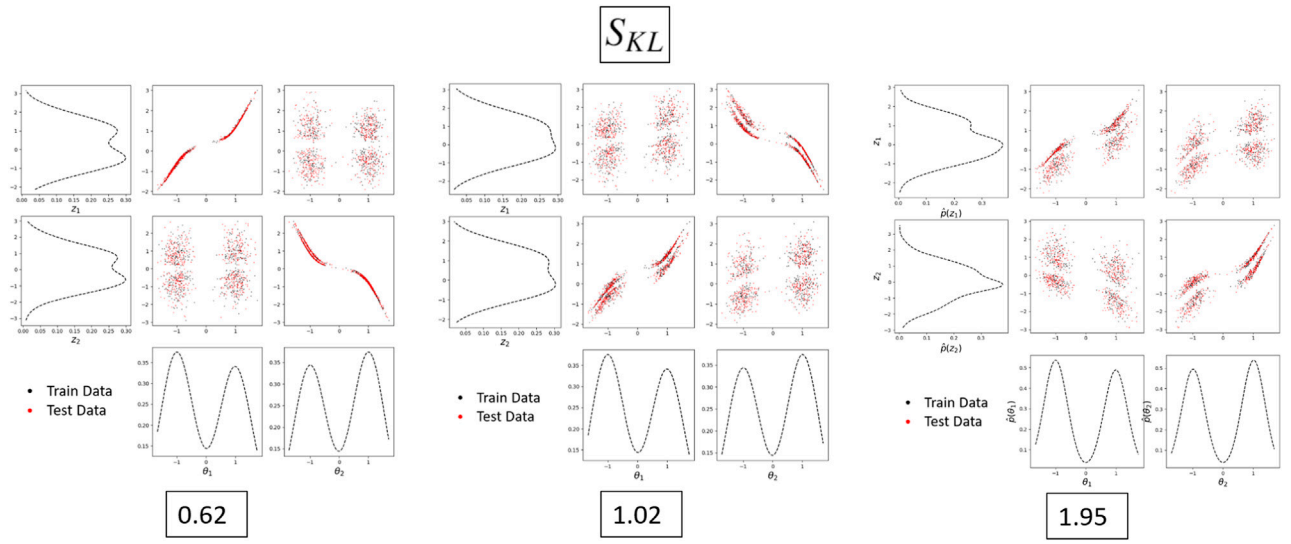


FIGURE 13 | A quantitative measure of disentanglement compared to a qualitative measure. As S_{KL} increases, the latent space becomes more entangled.

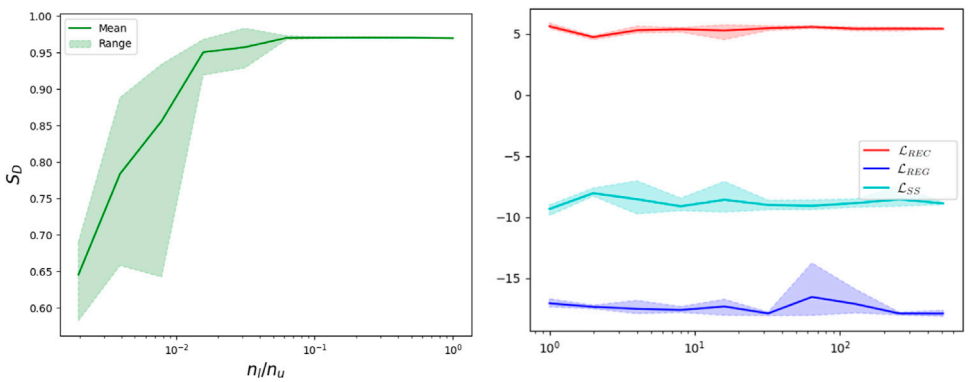


FIGURE 14 | (*left*) Disentanglement score mean increases with ratio of labeled to unlabeled samples when training with a semi-supervised loss. Disentanglement also becomes more consistently observed. (*right*) Training losses are unaffected by the number of labeled samples.

low density in the generative parameter distribution are better represented in the semi-supervised case over the unsupervised case; in other words, multimodality is better preserved (compare to **Figure 12**).

7 CONCLUDING REMARKS AND PERSPECTIVES

Learning representations such that each latent dimension corresponds to a single physical generative factor of variation is useful in many applications, particularly when learned in an unsupervised manner. Learning such *disentangled* representations using VAEs is dependent on many factors including network architecture, assumed form of distributions, prior selection, hyperparameters, and random seeding. The goal of our work is to develop 1) a consistent unsupervised framework to learn disentangled representations of data obtained through physical experiments or PDE simulations, and 2) to comprehensively characterize the underlying training process, and to recommend strategies to avoid sub-optimal representations.

Accurate reconstruction is desirable from a variety of perspectives, including being necessary for consistent disentanglement. Given two samples near one another in data space, and an accurate decoder, those two samples will be encourage to be near one another in latent space. This is a result of the sampling operation when computing the reconstruction loss. The reconstruction loss is minimized if samples near one another in latent space correspond to samples near one another in data space. Thus, finding an architecture suitable for accurate prediction from latent

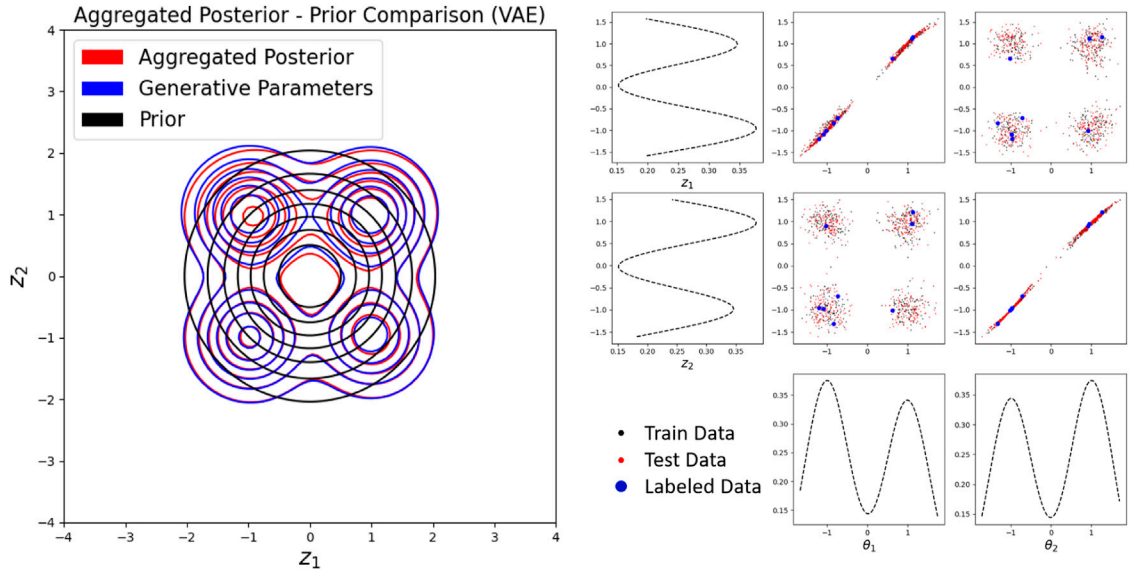


FIGURE 15 | (*left*) Aggregated posterior matches the generative parameter distribution with semi-supervised training. (*right*) Multi-modality is well preserved.

representations is of great importance to learn disentangled representations. In our experiments, different architectures were implemented before arriving at and refining the dense architecture (**Section 4.1**), which was found to accurately reconstruct the data from latent codes. Even with a suitable architecture, however, significant obstacles need to be overcome to arrive at a consistent framework for achieving disentangled representations. Over-regularization can often be difficult to avoid, especially when the variation among data samples is minor. This again emphasizes the necessity to accurately reconstruct the data first before attempting to learn meaningful representations. We have illustrated methods of avoiding over-regularization when training VAEs, but rotationally invariant priors can still create additional difficulties in the ability to disentangle parameters. We illustrated in **Section 5** that the standard normal prior typically assumed (which is rotationally-invariant) does not enforce any particular rotation of the latent space, often leading to entangled representations. Rotation of the latent space matters greatly, and without rotational enforcement on the encoder, disentanglement is rarely, or rather randomly, achieved when training with the ELBO loss. We have also shown that the implementation of hierarchical priors allows one to learn non-rotationally-invariant priors such that the regularization loss enforces a rotational constraint on the encoding distribution. However, the regularization loss can contain local minima as the latent space rotates, enforcing a non-factorized and thus incorrectly rotated aggregated posterior. This indicates the need for better prior selection, especially in higher latent dimensions when rotations create more complex effects.

Matching the aggregated posterior to the generative parameter distribution can also be enforce by including labeled samples during training. Including some number of labeled samples in the dataset and training with a semi-supervised loss, the aggregated posterior consistently matches the shape and orientation of the generative parameter distribution, effectively learning a disentangled representation. The multimodality of the data distribution is also better represented when using labeled data, indicating that the VAE can better predict data in regions of low density over the unsupervised version.

In reference to **Section 5**, the total correlation (TC) $D_{KL}[q_\phi(z)\|\prod_{i=1}^{n} q_\phi(z_i)]$ appears to be a useful and simpler measurement of disentanglement. When the generative parameters are completely independent (i.e., $p(\theta) = \prod_{i=1}^{p} p(\theta_i)$) and disentanglement occurs when a factorized $q_\phi(z)$ is learned (aligning the latent space axes with the generative parameter axes). This is the objective of the FactorVAE framework [8], which can successfully encourage a factorized $q_\phi(z)$ through the introduction of TC into the loss function, modifying to ELBO. However, considering a case in which the generative parameters are correlated, a factorized $q_\phi(z)$ is not necessarily desirable. It is in anticipation of a more correlated $p(\theta)$ that we use **Eq. 7** as a measure of disentanglement. Additionally, in our work we do not modify the standard VAE objective to produce more accurate reconstruction of the data.

Complete disentanglement has not been observed when generative parameters are correlated in our experiments, but after many trials the same conclusions have been drawn as the uncorrelated case: for disentanglement to occur, the aggregated posterior must contain the same "shape" as the generative parameter distribution - this includes correlations up to permutations of the axes. The **Supplementary Material** further illustrates these ideas. Future work will include disentangling correlated generative parameters, which may be facilitated through learning correlated priors using HP.

In addition to disentangling correlated generative parameters, our broader aim is to extend our work to more complex problems to create a general framework for consistent unsupervised or semi-supervised representation learning. Through our observations here regarding non-rotationally invariant priors along with insights gained from [16], we hypothesize that such a framework will be largely focused on both prior selection and the structural form of the encoding and decoding distributions. Additionally, in a completely unsupervised setting, one must find an encoder and decoder which disentangle the generative parameters, but the dimension of the generative parameters may be unknown. The dimension of the latent space is always user-specified; if the dimension of the latent space is too small or too large, how does this effect the learned representation? Can one successfully and consistently disentangle generative parameters in higher dimensions? These are some of the open questions to be addressed in the future.

The issue of over-regularization often greatly hinders our ability to train VAEs (**Section 4.2**). Different initialization strategies may be investigated to increase training performance and avoid the issue altogether. It has been shown that principled selection of activation functions, architecture, and initialization can greatly improve not only the efficiency of training, but also facilitate greater performance in terms of reconstruction [31].

The greater scope of this work is to develop an unsupervised and interpretable representation learning framework to generate probabilistic reduced order models for physical problems and use learned representations for efficient design optimization.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found in the article/ **Supplementary Material**.

AUTHOR CONTRIBUTIONS

CJ developed the methodology for generative modeling, wrote the code, and performed experiments. KD developed the problem statement, and templated ideas on simple problems. CJ was the primary author of the paper while KD provided the direction.

FUNDING

The authors acknowledge support from the Air Force under the grant FA9550-17-1-0195 (Program Managers: Dr. Mitat Birkan, and Dr. Fariba Fahroo).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphy.2022.890910/full#supplementary-material

REFERENCES

1. Li D, Zhang M, Chen W, Feng G. Facial Attribute Editing by Latent Space Adversarial Variational Autoencoders. In: International Conference on Pattern Recognition (2018). p. 1337–42. doi:10.1109/icpr.2018.8545633
2. Wang T, Liu J, Jin C, Li J, Ma S. An Intelligent Music Generation Based on Variational Autoencoder. In: International Conference on Culture-oriented Science Technology (2020). p. 394–8. doi:10.1109/iccst50977.2020.00082
3. Amini A, Schwarting W, Rosman G, Araki B, Karaman S, Rus D. Variational Autoencoder for End-To-End Control of Autonomous Driving with Novelty Detection and Training De-biasing. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (2018). p. 568–75. doi:10.1109/iros.2018.8594386
4. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative Adversarial Networks. *Adv Neural Inf Process Syst* (2014) 3.
5. Kingma D, Welling M. Auto-Encoding Variational Bayes. In: International Conference on Learning Representations (2013).
6. Higgins I, Matthey L, Pal A, Burgess CP, Glorot X, Botvinick M, et al. Eta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In: International Conference on Learning Representations (2017).
7. Klushyn A, Chen N, Kurle R, Cseke B, Smagt PVD. Learning Hierarchical Priors in VAEs. . In: Conference on Neural Information Processing Systems (2019).
8. Kim H, Mnih A. Disentangling by Factorising. In: International Conference on Machine Learning (2018).
9. Zhao S, Song J, Ermon S. InfoVAE: Information Maximizing Variational Autoencoders. *arXiv* **1706.02262** (2017).
10. Bengio Y, Courville A, Vincent P. Representation Learning: A Review and New Perspectives. *IEEE Trans Pattern Anal Mach Intell* (2013) 35:1798–828. doi:10.1109/tpami.2013.50
11. Tait DJ, Damoulas T. Variational Autoencoding of PDE Inverse Problems. *arXiv* **2006**.15641 (2020).
12. Lu PY, Kim S, Soljačić M. Extracting Interpretable Physical Parameters from Spatiotemporal Systems Using Unsupervised Learning. *Phys Rev* (2020) 10(3): 031056. doi:10.1103/physrevx.10.031056
13. Lopez R, Atzberger P. Variational Autoencoders for Learning Nonlinear Dynamics of Physical Systems. *arXiv* **2012.03448** (2020).
14. Donà J, Franceschi JY. Sylvain Lamprier, Patrick Gallinari. PDE-Driven Spatiotemporal Disentanglement. In: International Conference on Learning Representations (2021).
15. Eastwood C, Williams CKI. A Framework for the Quantitative Evaluation of Disentangled Representations. In: International Conference on Learning Representations (2018).
16. Rolinek M, Zietlow D, Martius G. Variational Autoencoders Pursue PCA Directions (By Accident). *arXiv* **1812.06775** (2019).
17. Burgess CP, Higgins I, Pal A, Matthey L, Watters N, Desjardins G, et al. Understanding Disentangling in β-VAE (2018). *arXiv* **1804.03599** (2018).
18. Locatello F, Bauer S, Lucic M, Gelly S, Schölkopf B, Bachem O. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. *arXiv* **1811**.12359 (2019).
19. Wang Q, Kulkarni SR, Verdu S. Divergence Estimation for Multidimensional Densities via k-Nearest-Neighbor Distances. *IEEE Trans Inform Theor* (2009) 55:2392–405. doi:10.1109/tit.2009.2016060
20. Alemi A, Poole B, Fischer I, Dillon J, Saurous RA, Murphy K. Fixing a Broken ELBO. Editor Dy J, Krause A. Proceedings of the 35th International Conference on Machine Learning, July 10-15, 2008. PMLR, 80, 159–168 (2018). Available at: http://proceedings.mlr.press/v80/alemi18a/alemi18a.pdf
21. Makhzani A, Shlens J, Jaitly N, Goodfellow I. Adversarial Autoencoders. . In: International Conference on Learning Representations (2016).
22. Odaibo SG. Tutorial: Deriving the Standard Variational Autoencoder (VAE) Loss Function. *arXiv* **1907.08956** (2019).

23. Yu R. A Tutorial on VAEs: From Bayes' Rule to Lossless Compression. *arXiv* **2006**.10273 (2020).
24. Duraisamy K. Variational Encoders and Autoencoders : Information-Theoretic Inference and Closed-form Solutions. *arXiv* **2101**.11428 (2021).
25. Berger T. *Rate Distortion Theory. A Mathematical Basis for Data Compression*. Englewood Cliffs: Prentice-Hall (1971).
26. Cover TM, Thomas JA. *Elements of Information Theory Wiley Series in Telecommunications and Signal Processing*. United States: Wiley-Interscience (2001).
27. Gibson J. *Information Theory and Rate Distortion Theory for Communications and Compression*. Springer Cham. doi:10.1007/978-3-031-01680-6 (2013).
28. Lucas J, Tucker G, Grosse RB, Norouzi M. Understanding Posterior Collapse in Generative Latent Variable Models. In: International Conference on Learning Representations (2019).
29. Zhu Y, Zabaras N. Bayesian Deep Convolutional Encoder-Decoder Networks for Surrogate Modeling and Uncertainty Quantification. *J Comput Phys* (2018) 366:415–47. doi:10.1016/j.jcp.2018.04.018
30. Fu H, Li C, Liu X, Gao J, Çelikyilmaz A, Carin L. Cyclical Annealing Schedule: A Simple Approach to Mitigating KL Vanishing. In: North American chapter of the Association for Computational Linguistics (2019).
31. Sitzmann V, Martel JN, Bergman AW, Lindell DB, Wetzstein G. Implicit Neural Representations with Periodic Activation Functions. *arXiv* **2006.09661** (2020).
32. Nelder JA, Mead R. A Simplex Method for Function Minimization. *Computer J* (1965) 7:308–13. doi:10.1093/comjnl/7.4.308

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

APPENDIX A: ROTATIONALLY-INVARIANT DISTRIBUTIONS

A matrix $\mathbf{R} \in \mathbb{R}^{n\times n}$ is a rotation matrix if for all $z \in \mathbb{R}^n$, $\|\mathbf{R}z\|_2 = \|z\|_2$.

A probability distribution $p(z)$ is said to be *rotationally-invariant* if $p(z) = p(\mathrm{R}z)$ for all $z \in \mathbb{R}^n$ and for all rotation matrices $\mathbf{R} \in \mathbb{R}^{n\times n}$.

The ELBO loss is unaffected by rotations of the latent space when training with a rotationally-invariant prior. This is shown in detail in [16].

frontiers | Frontiers in Physics

ORIGINAL RESEARCH
published: 01 July 2022
doi: 10.3389/fphy.2022.898028

On the Entropy Projection and the Robustness of High Order Entropy Stable Discontinuous Galerkin Schemes for Under-Resolved Flows

Jesse Chan[1]*, *Hendrik Ranocha*[2], *Andrés M. Rueda-Ramírez*[3], *Gregor Gassner*[3,4] and *Tim Warburton*[5]

[1]Department of Computational and Applied Mathematics, Rice University, Houston, TX, United States, [2]Department of Mathematics, University of Hamburg, Hamburg, Germany, [3]Department of Mathematics and Computer Science, University of Cologne, Cologne, Germany, [4]Center for Data and Simulation Science, University of Cologne, Cologne, Germany, [5]Department of Mathematics, Virginia Tech, Blacksburg, VA, United States

High order entropy stable schemes provide improved robustness for computational simulations of fluid flows. However, additional stabilization and positivity preserving limiting can still be required for variable-density flows with under-resolved features. We demonstrate numerically that entropy stable Discontinuous Galerkin (DG) methods which incorporate an "entropy projection" are less likely to require additional limiting to retain positivity for certain types of flows. We conclude by investigating potential explanations for this observed improvement in robustness.

Keywords: computational fluid dynamics, high order, discontinuous Galerkin (DG), summation-by-parts (SBP), entropy stability, robustness

OPEN ACCESS

Edited by:
Michel Bergmann, Inria Bordeaux—Sud-Ouest Research Centre, France

Reviewed by:
Bulent Karasozen, Middle East Technical University, Turkey
Mahboub Baccouch, University of Nebraska Omaha, United States

***Correspondence:**
Jesse Chan
jesse.chan@rice.edu

Specialty section:
This article was submitted to Statistical and Computational Physics, a section of the journal Frontiers in Physics

Received: *16 March 2022*
Accepted: *06 April 2022*
Published: *01 July 2022*

Citation:
Chan J, Ranocha H, Rueda-Ramírez AM, Gassner G and Warburton T (2022) On the Entropy Projection and the Robustness of High Order Entropy Stable Discontinuous Galerkin Schemes for Under-Resolved Flows. Front. Phys. 10:898028. doi: 10.3389/fphy.2022.898028

1 INTRODUCTION

Discontinuous Galerkin (DG) schemes have received interest within computational fluid dynamics (CFD) due to their high order accuracy and ability to handle unstructured curved meshes. In particular, there has been interest in DG methods for simulations of under-resolved flows [1–5]. Among such schemes, "entropy stable" DG methods based on a "flux differencing" formulation have received interest due to their robustness with respect to shocks and turbulence [6–9].

Entropy conservative and entropy stable flux differencing schemes were originally formulated for finite difference methods in [10, 11]. They were extended to tensor product grids using discontinuous spectral collocation schemes (also known as discontinuous Galerkin spectral element methods, or DGSEM) [12, 13]. Entropy stable collocation schemes were extended to simplicial meshes in [14, 15] using multi-dimensional summation-by-parts (SBP) operators [16]. Non-collocation entropy stable schemes have also been developed. These schemes began with staggered grid schemes on tensor product grids in [17], which were later extended to simplicial elements in [18]. "Modal" entropy stable DG formulations [19–21] have been utilized to construct a variety of new entropy stable schemes, including Gauss DG methods [22, 23] and reduced order models [24]. We note that under appropriate choices of quadrature, these "modal" formulations reduce to collocation-type entropy stable schemes. Entropy stable schemes have since been extended to an even wider array of discretizations, such as line DG methods, discontinuous Galerkin difference methods, and C^0 continuous discretizations [25–27].

The main difference between non-collocation and collocation-type entropy stable schemes is the use of transformations between conservative variables and entropy variables together with projection

or prolongation operators to facilitate a discrete proof of entropy stability. This is referred to as the "entropy projection" in [19, 25] and as the interpolation or prolongation of entropy variables in [17, 27]. This approach is also equivalent to the mixed formulation of [28]. We will refer to this transformation as the "entropy projection" for the remainder of the paper.

The motivation for introducing the entropy projection has been to enable the use of more accurate quadrature rules or novel basis functions. This has been at the cost of additional complexity and issues related to the sensitivity of the entropy variables for near-vacuum states [19, 27]. To the best of the authors' knowledge, no inherent advantages in using the entropy projection have been observed in the literature. This paper focuses on the following observation: high order entropy stable schemes based on the entropy projection appear to be more robust than entropy stable collocation schemes for two and three dimensional simulations of under-resolved variable-density fluid flows with small-scale features.

The structure of the paper is as follows: **Section 2** reviews mathematical formulations of entropy stable schemes which involve the entropy projection. **Section 3** documents the observed difference in robustness for a variety of problems in two and three dimensions, and provides analysis and numerical experiments which support that the primary difference between unstable and stable schemes is the entropy projection. **Section 4** conjectures potential explanations for why the entropy projection might improve robustness. We conclude with **Section 5**, which explores potential applications towards under-resolved flow simulations.

2 FORMULATION OF HIGH ORDER ENTROPY STABLE DISCONTINUOUS GALERKIN SCHEMES

In this section, we provide a brief description of high order entropy stable schemes in 1D. More detailed derivations, multi-dimensional formulations, and extensions to curved grids can be found in [14, 15, 19, 21, 22, 24].

The notation in this paper is motivated by notation in [15, 29]. Unless otherwise specified, vector and matrix quantities are denoted using lower and upper case bold font, respectively. Spatially discrete quantities are denoted using a bold sans serif font. Finally, the output of continuous functions evaluated over discrete vectors is interpreted as a discrete vector.

For example, if $\mathbf{x}$ denotes a vector of point locations, i.e., $(\mathbf{x})_i = x_i$, then $u(\mathbf{x})$ is interpreted as the vector

$$(u(\mathbf{x}))_i = u(\mathbf{x}_i).$$

Similarly, if $\mathbf{u} = u(\mathbf{x})$, then $f(\mathbf{u})$ corresponds to the vector

$$\big(f(\mathbf{u})\big)_i = f(u(\mathbf{x}_i)).$$

Vector-valued functions are treated similarly. For example, given a vector-valued function $\boldsymbol{f}\colon \mathbb{R}^n \to \mathbb{R}^n$ and a vector of coordinates $\mathbf{x}$, we adopt the convention that $(\boldsymbol{f}(\mathbf{x}))_i = \boldsymbol{f}(\boldsymbol{x}_i)$.

2.1 Conservation Laws With Entropy

In this section, we review the construction of entropy conservative and entropy stable schemes for a one-dimensional system of nonlinear conservation laws

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x} = \boldsymbol{s}(\boldsymbol{u}),$$

where $\boldsymbol{s}(\boldsymbol{u})$ is a source term. We assume the domain is exactly represented by a uniform mesh consisting of non-overlapping intervals D^k, and that the solution $u(x)$ is approximated by degree N polynomials over each element. We also introduce entropy conservative numerical fluxes $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ [30], which are bivariate functions of "left" and "right" states $\boldsymbol{u}_L$, $\boldsymbol{u}_R$. In addition to being symmetric and consistent, entropy conservative numerical fluxes satisfy an "entropy conservation" property

$$(\boldsymbol{v}_L - \boldsymbol{v}_R)^T \boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R) = \psi(\boldsymbol{u}_L) - \psi(\boldsymbol{u}_R). \tag{1}$$

here, $\boldsymbol{v}_L$, $\boldsymbol{v}_R$ are entropy variables evaluated at the left and right states, and $\psi(\boldsymbol{u})$ denotes the "entropy potential". Examples of expressions for entropy variables and entropy potentials can be found in [14].

2.2 Collocation Formulations

Degree N entropy stable collocation schemes are typically built from Legendre-Gauss-Lobatto (LGL) quadrature rules with $(N+1)$ points. Let $\mathbf{x}, \mathbf{w}$ denote vectors of quadrature points and weights on the reference interval $[-1, 1]$. Let $\ell_i(x)$ denote Lagrange polynomials at LGL nodes, and let $\mathbf{u}$ denote the vector of solution nodal values $u(x_i)$. Define the matrices

$$\mathbf{M} = \operatorname{diag}(\mathbf{w}), \qquad \mathbf{Q}_{ij} = \int_{-1}^{1} \frac{\partial \ell_j}{\partial x} \ell_i \mathrm{d}x,$$

$$\mathbf{B} = \begin{bmatrix} -1 & & \\ & & \\ & & 1 \end{bmatrix}, \qquad \mathbf{V}_f = \begin{bmatrix} 1 & \dots & 0 \\ 0 & \dots & 1 \end{bmatrix}.$$

here $\mathbf{V}_f$ is a face interpolation or extraction matrix which maps from volume nodes to face nodes. Flux derivatives are discretized using a "flux differencing" approach involving summation-by-parts (SBP) operators and entropy conservative fluxes [30]. An entropy stable collocation formulation can now be defined on an element D^k as follows:

$$h\mathbf{M}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} + \Big(\big(\mathbf{Q} - \mathbf{Q}^T\big) \circ \mathbf{F}\Big)1 + \mathbf{V}_f^T \mathbf{B} \mathbf{f}^* = \mathbf{s}(\mathbf{u}), \qquad \mathbf{F}_{ij} = \boldsymbol{f}_S\big(\mathbf{u}_i, \mathbf{u}_j\big), \tag{2}$$

where h is the size of the element D^k and $\circ$ denotes the matrix Hadamard product [10–12].[1] Here, $\mathbf{f}^*$ is a vector which contains numerical fluxes at the left and right endpoints of the interval

[1]Since the entries of $\mathbf{F}$ are vector-valued, the Hadamard product $(\mathbf{Q} - \mathbf{Q}^T) \circ \mathbf{F}$ should be understood as each scalar entry of $(\mathbf{Q} - \mathbf{Q}^T)$ multiplying each vector-valued entry of $\mathbf{F}$.

$$\mathbf{f}^* = \begin{bmatrix} f^*(\mathbf{u}_1^+, \mathbf{u}_1) \\ f^*(\mathbf{u}_{N+1}, \mathbf{u}_{N+1}^+) \end{bmatrix},$$

where $\mathbf{u}_1^+, \mathbf{u}_{N+1}^+$ denote exterior nodal values on neighboring elements. If f^* is an entropy conservative flux, then the resulting numerical method is semi-discretely entropy conservative. If f^* is an entropy stable flux (for example, Lax-Friedrichs flux, HLLC, and certain matrix penalizations [14, 31]) then the resulting scheme also dissipates entropy.

2.3 "Modal" Formulations

Degree N entropy stable "modal" DG schemes generalize collocation schemes to arbitrary choices of quadrature. In one dimension, this allow for the use of higher accuracy volume quadratures. In higher dimensions, modal formulations also enable more general choices of surface quadrature. These schemes introduce an additional "entropy projection" step to facilitate the semi-discrete proof of entropy stability or conservation.

We now assume the solution is represented using some arbitrary basis over each element, such that $u(x) = \sum_j \mathbf{u}_j \phi_j(x)$. Let $\boldsymbol{x}$, $\boldsymbol{w}$ now denote a general quadrature rule with positive quadrature weights. We define quadrature-based interpolation matrices $\mathbf{V}_q, \mathbf{V}_f$, the mass matrix $\mathbf{M}$, and the modal differentiation matrix $\hat{\mathbf{Q}}$

$$\left(\mathbf{V}_q\right)_{ij} = \phi_j(\mathbf{x}_i), \qquad \left(\mathbf{V}_f\right)_{1j} = \phi_j(-1), \qquad \left(\mathrm{V}_f\right)_{2j} = \phi_j(1),$$

$$\mathbf{M} = \mathbf{V}_q^T \mathrm{diag}(\mathbf{w}) \mathbf{V}_q, \qquad \hat{\mathbf{Q}}_{ij} = \int_{-1}^{1} \frac{\partial \phi_j}{\partial x} \phi_i \mathrm{d}x.$$

We introduce the quadrature-based projection matrix $\mathbf{P}_q = \mathbf{M}^{-1} \mathbf{V}_q^T \mathrm{diag}(\mathbf{w})$. Using $\mathbf{P}_q$ and $\hat{\mathbf{Q}}$, we can construct quadrature-based differentiation and extrapolation matrices $\mathbf{Q}, \mathbf{E}$

$$\mathbf{Q} = \mathbf{P}_q^T \hat{\mathbf{Q}} \mathbf{P}_q, \qquad \mathbf{E} = \mathbf{V}_f \mathbf{P}_q.$$

To accommodate general quadrature rules which may not include boundary points, we introduce hybridized SBP operators $\mathbf{Q}_h$ on the reference interval $[-1, 1]$

$$\mathbf{Q}_h = \frac{1}{2} \begin{bmatrix} \mathbf{Q} - (\mathbf{Q})^T & \mathbf{E}^T \mathbf{B} \\ -\mathbf{B}\mathbf{E} & \mathbf{B} \end{bmatrix}.$$

The use of such operators simplifies the implementation for general quadrature rules and nodal sets which do not include boundary nodes [19, 32]. Next, we define $\mathbf{V}_h$ as the interpolation matrix to both volume and surface quadrature points

$$\mathbf{V}_h = \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix}.$$

We also introduce the L^2 projection of the entropy variables and the "entropy projected" conservative variables $\tilde{\mathbf{u}}$

$$\mathbf{v} = \mathbf{P}_q v\left(\mathbf{V}_q \mathbf{u}\right), \qquad \tilde{\mathbf{u}} = u(\mathbf{V}_h \mathbf{v}),$$

which are defined by evaluating the mapping from conservative to entropy variables $u(v)$ using the projected entropy variables. Here, $v(u)$ denotes the mapping from conservative to entropy variables. Note that the projected entropy variables $\mathbf{v}$ is a vector corresponding to modal coefficients, while $\tilde{\mathbf{u}}$ corresponds to point values at volume and face quadrature points.

An entropy stable modal DG discretization over a single element D^k is then

$$h\mathbf{M} \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} + \mathbf{V}_h^T \left(\left(\mathbf{Q}_h - \mathbf{Q}_h^T \right) \circ \mathbf{F} \right) 1 + \mathbf{V}_f^T \mathbf{B} \mathbf{f}^* = \mathbf{s}(\mathbf{u}), \tag{3}$$

$$\mathbf{F}_{ij} = f_S\left(\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j\right), \qquad \mathbf{f}^* = \begin{bmatrix} f^*(\tilde{\mathbf{u}}_1^+, \tilde{\mathbf{u}}_1) \\ f^*(\tilde{\mathbf{u}}_{N+1}, \tilde{\mathbf{u}}_{N+1}^+) \end{bmatrix}.$$

Note that the right hand side formulation is evaluated not using the conservative variables $\mathbf{u}$, but the "entropy projected" conservative variables $\tilde{\mathbf{u}}$.

While we have presented entropy stable DG schemes using a general "modal" DG framework, the formulation reduces to existing methods under appropriate choices of quadrature and basis. For example, specifying LGL quadrature on a tensor product element recovers entropy stable spectral collocation schemes [22]. SBP discretizations without an underlying basis on simplices [14–16] can also be recovered for appropriate quadrature rules by redefining the interpolation and projection matrices $\mathbf{V}_q, \mathbf{P}_q$ [33].

3 NUMERICAL COMPARISONS OF COLLOCATION AND ENTROPY PROJECTION SCHEMES

In this section, we will demonstrate numerically that a significant difference in robustness is observed between collocation and entropy projection-based discretizations of the Euler and ideal MHD equations. For the Euler equations, we study the Kelvin-Helmholtz, Rayleigh-Taylor, and Richtmeyer-Meshkov instabilities, and for the MHD equations we study a magnetized Kelvin-Helmholtz instability. All of these examples exhibit small-scale turbulent-like features. Moreover, we observe a difference in robustness between entropy stable collocation and entropy projection-based methods independently of the polynomial degrees, mesh resolutions, and type of mesh (e.g., quadrilateral or triangular). We focus on the following entropy stable DG methods:

- On quadrilateral meshes:
  (1) DGSEM: collocation scheme based on the tensor product of one-dimensional (N + 1) point LGL quadrature,
  (2) Gauss DG: a "collocation" scheme based on the tensor product of one-dimensional (N + 1) point Gauss quadrature. The entropy projection is used to evaluate interface fluxes [22],
- On triangular meshes:
  (1) SBP: a collocation scheme based on multi-dimensional summation-by-parts finite difference operators [14, 16],
  (2) Modal: a modal formulation utilizing quadrature rules which exactly integrate entries of the volume and face mass matrices [19].

Remark 1. It is known that the Kelvin-Helmholtz, Rayleigh-Taylor, and Richtmeyer-Meshkov instabilities are notoriously sensitive to initial conditions and discretization parameters, and that numerical schemes may not converge to a unique solution [34, 35]. Instead, this paper focuses on these problems as stress tests of robustness.

Unless specified otherwise, all numerical experiments utilize a Lax-Friedrichs interface flux with Davis wavespeed estimate [36]. We also experimented with HLL and HLLC surface fluxes, but did not notice a significant difference. We also note that instead of discontinuous initial conditions, we utilize smoothed approximations for each problem considered here.

All experiments are also performed on uniform meshes. For triangular meshes, this mesh is constructed by bisecting each element of a uniform quadrilateral mesh along the diagonal. Unless specified otherwise, all results are produced using the Julia [37] simulation framework Trixi.jl [38, 39]. For most experiments, we utilize an optimized adaptive 4th order 9-stage Runge-Kutta method [40] implemented in OrdinaryDiffEq.jl [41]. The absolute and relative tolerances are set to 10^{-7} unless specified otherwise. Scripts generating main results are included in a companion repository for reproducibility [42].

We note that the robustness, efficiency, and high order accuracy of both entropy stable DGSEM and entropy stable Gauss DG schemes have been verified in previous works [7–9, 22, 23], and will not be addressed in detail in this paper. However, the difference in robustness between the two methods has not been previously observed in the literature, and will be the focus of this work.

3.1 Euler Equations of Gas Dynamics

We consider first the two and three-dimensional problems for the Euler equations of gas dynamics. The conservative variables for the three-dimensional Euler equations are density, momentum, and total energy, $\boldsymbol{u} = (\rho, \rho\mathbf{v}, E)$, where the vector $\mathbf{v} = (u, v, w)$ contains the velocities in x, y and z, respectively. The flux reads

$$\boldsymbol{f}(\boldsymbol{u}) = \begin{pmatrix} \rho\mathbf{v} \\ \rho\mathbf{v}\mathbf{v}^T + \mathbf{I}p \\ \mathbf{v}\left(\frac{1}{2}\rho\|\mathbf{v}\|^2 + \frac{\gamma p}{\gamma - 1}\right) \end{pmatrix},$$

where $\mathbf{I}$ is the 3×3 identity matrix, γ is the heat capacity ratio, and $p = (\gamma - 1)(E - \rho\|\mathbf{v}\|^2/2)$ is the gas pressure. For two-dimensional problems, we neglect the third component of the velocity, w, and $\mathbf{I}$ becomes the 2×2 identity matrix.

All the following experiments use the entropy conservative and kinetic energy preserving flux of Ranocha [43, 44]; however, similar results were observed when experimenting with the entropy conservative flux of Chandrashekar [45].

3.1.1 Two Dimensional Kelvin-Helmholtz Instability

We perform additional experiments analyzing the robustness of entropy stable DGSEM and Gauss DG for the Kelvin-Helmholtz instability. The domain is $[-1, 1]^2$ with initial condition from [46]:

$$\begin{aligned} \rho &= \frac{1}{2} + \frac{3}{4}B, & p &= 1, \\ u &= \frac{1}{2}(B - 1), & v &= \frac{1}{10}\sin(2\pi x), \end{aligned} \tag{4}$$

where $B(x, y)$ is a smoothed approximation to a discontinuous step function

$$B(x, y) = \tanh(15y + 7.5) - \tanh(15y - 7.5). \tag{5}$$

Each solver is run until final time $T_{\text{final}} = 15$. As can be observed in **Figure 1**, the solution differs significantly between the $N = 3$ and $N = 7$ simulations. This is likely a consequence of the well-known sensitivity of the Kelvin-Helmholtz instability to small perturbations and numerical resolutions [34, 35]. End times for each simulation can be found in **Table 1**.

3.1.2 Two Dimensional Rayleigh-Taylor Instability

The two-dimensional Rayleigh-Taylor instability generates small-scale flow features through buoyancy or gravity effects [47, 48]. The setup involves a heavy and light fluid suspended above one another separated by a curved interface, and buoyancy or gravity results in displacement of the lighter fluid into the heavier one. This displacement causes velocity shear and the formation of additional Kelvin-Helmholtz instabilities along the interface. The domain is $[0, 1/4] \times [0, 1]$.

Let $d_{a,b}(x) = a + \frac{1}{2}(1 + \tanh(sx))(b - a)$ denote a smoothed approximation (with slope s) to a discontinuous function with values a for $x < 0$ and b for $x > 0$. The initial condition is given by

$$\begin{aligned} \rho &= d_{2,1}\left(y - \frac{1}{2}\right), & p &= \begin{cases} 2y + 1 & y < 1/2 \\ y + 3/2 & y \geq 1/2 \end{cases}, \\ u &= 0 & v &= -\frac{c}{40}\cos(8k\pi x)\sin(\pi y)^6, \end{aligned}$$

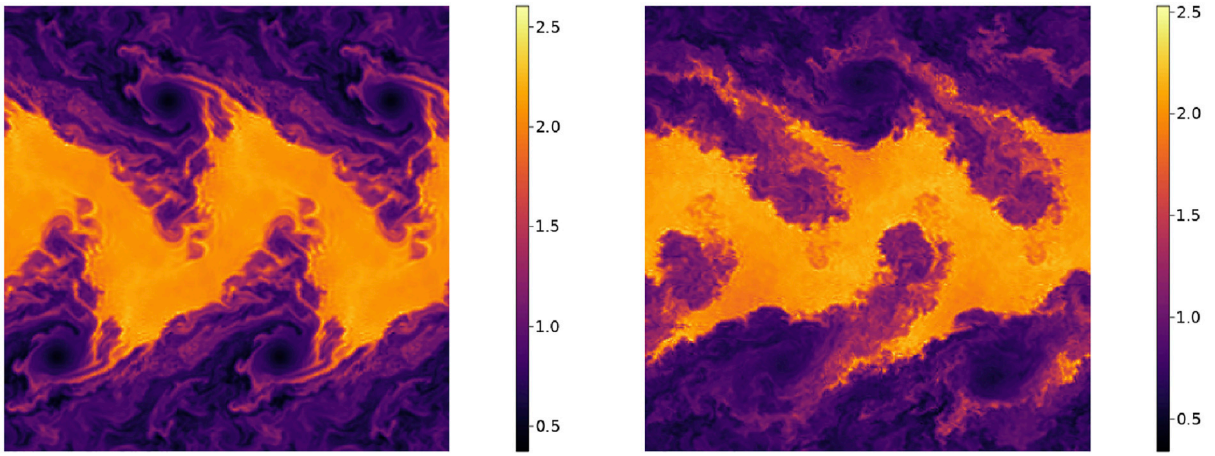
where $c = \sqrt{\gamma p/\rho}$ is the speed of sound. Here, we borrow from [49] and multiply the y-velocity perturbation by $\sin(\pi y)^6$ so that u, v satisfy wall boundary conditions. We also add gravity source terms to the y-momentum and energy equations:

$$\boldsymbol{s}(\boldsymbol{x}, t) = \begin{bmatrix} 0 & 0 & g\rho & g\rho v \end{bmatrix},$$

where $s = 15$. Note that the sign of gravity is such that the light fluid flows up into the heavy fluid. Reflective wall boundary conditions are imposed at all boundaries using mirror states, which results in an entropy stable scheme under the Lax-Friedrichs flux [14, 50]. **Figure 2** shows snapshots of the density for a degree $N = 3$ entropy stable Gauss DG scheme on a mesh of 32×128 elements at various times. End times for each simulation can be found in **Table 2**.

3.1.3 Two Dimensional Richtmeyer-Meshkov Instability

The Richtmeyer-Meshkov instability generates small-scale flow features by passing a shock over a stratified fluid [47, 51]. The domain for this setup is $[0, 40/3] \times [0, 40]$, and the initial density and pressure are given by



Density for entropy stable Gauss collocation at time $T = 10$ (degree $N = 3$ and a 64×64 mesh).

Density for entropy stable Gauss collocation at time $T = 10$ (degree $N = 7$ and a 32×32 mesh).

FIGURE 1 | Snapshots of density for the Kelvin-Helmholtz instability using an entropy stable Gauss DG scheme on uniform quadrilateral meshes.

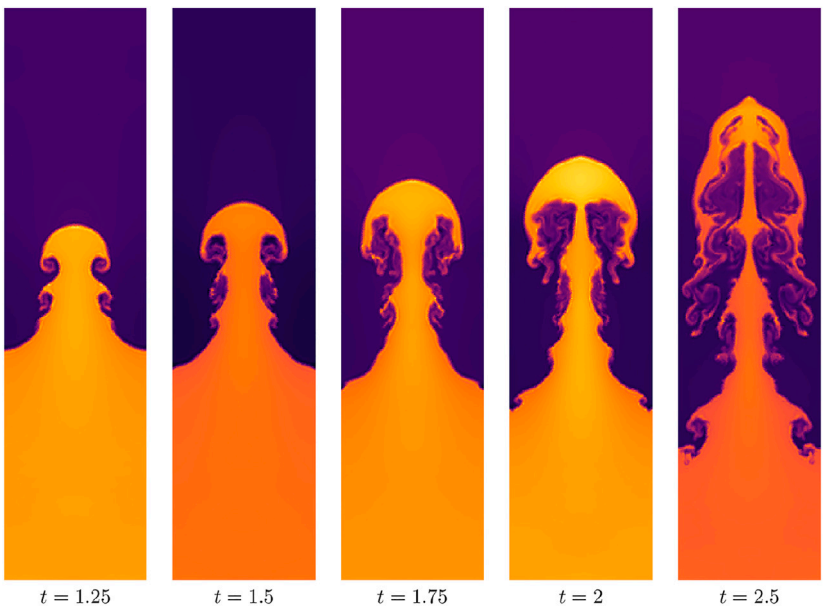


FIGURE 2 | Density for a Rayleigh-Taylor instability for a degree $N = 3$ entropy stable Gauss DG scheme on a mesh of 32×128 elements.

$$\rho = d_{1,\frac{1}{4}}\left(y - \left(18 + 2\cos\left(\frac{6\pi x}{L}\right)\right)\right) + d_{3.22,0}\left(|y-4| - 2\right),$$
$$p = d_{4.9,1}\left(|y-4| - 2\right),$$

where we again set the slope $s = 15$. The initial velocities are both set to zero, i.e., $u, v = 0$. We approximate the discontinuous initial condition using smoothed Heaviside functions with a slope of $s = 2$ due to the size of the domain. Reflective wall boundary conditions are imposed everywhere. **Figure 3** shows pseudocolor plots of the density using a degree $N = 3$ entropy stable Gauss DG on a uniform mesh of 32×96 quadrilateral elements. End times for each simulation can be found in **Table 3**.

3.1.4 Three-Dimensional Kelvin-Helmholtz Instability

For completeness, we also verify that a difference in robustness is observed for instability-type problems in three dimensions. Due to the high computational cost of entropy stable DG methods on tetrahedral meshes, we restrict ourselves to hexahedral meshes for the following experiments. We adapt the Kelvin-Helmholtz instability to three dimensions using the following initial condition:

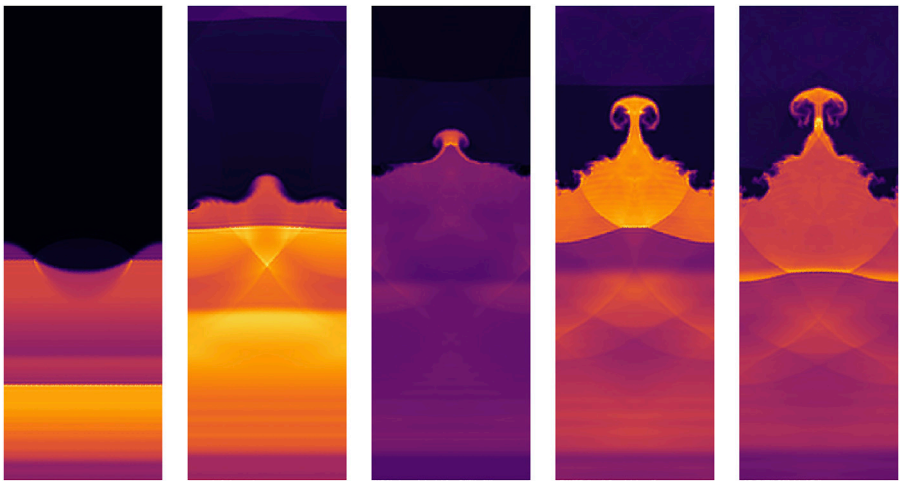


FIGURE 3 | Density for the Richtmeyer-Meshkov instability using a degree $N = 3$ entropy stable Gauss DG with 32×96 elements. The domain is $[0, 40/3] \times [0, 40]$.

TABLE 1 | End time for simulations of the 3D Kelvin-Helmholtz instability on hexahedral meshes. "Collocation" refers to a nodal DGSEM discretization, while "entropy projection" refers to a method based on Gauss nodes.

Degree	1	2	3	4	5	6	7
Solver							
3D KHI, $N_{\text{cells}} = 16$							
Collocation	10	2.73	2.111	1.978	2.059	1.797	1.893
Entropy projection	10	10	10	10	10	10	10
3D KHI, $N_{\text{cells}} = 32$							
Collocation	4.049	2.451	2.061	1.721	2.071	1.973	1.952
Entropy projection	10	10	10	10	10	10	10

Times colored blue correspond to simulations which did not crash and ran to completion, while times colored red denote simulations which did crash.

$$\rho = \frac{1}{2} + \frac{3}{4}B \qquad p = 1$$
$$u = \frac{1}{2}(B-1) \qquad v = \frac{1}{10}\sin(2\pi x)\sin(2\pi z) \qquad w = \frac{1}{10}\sin(2\pi x)\sin(2\pi z),$$

where B is defined as in **Eq. 5**. **Table 3** shows the results, which are similar to previous results for the two-dimensional test problems. We note for this example, both the relative and absolute adaptive time-step tolerances were set to 10^{-8} instead of 10^{-7}. This was necessary to avoid crashes for the entropy projection method at degrees $N = 6$ and $N = 7$ on the finer $N_{\text{cells}} = 32$ mesh.

3.2 Ideal GLM-MHD Equations

Next, we consider the ideal GLM-MHD equations. These equations use generalized Lagrange multiplier (GLM) technique to evolve towards a solution that bounds the magnetic field divergence. When the magnetic field divergence is non-zero, the GLM-MHD system requires the use of non-conservative terms to achieve entropy stability and to ensure Galilean invariance in the divergence cleaning technique.

The non-conservative GLM-MHD system without source terms reads

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \boldsymbol{f}(\boldsymbol{u}) + \boldsymbol{\Upsilon} = \boldsymbol{0}, \tag{6}$$

where the state variables are density, momentum, total energy, magnetic field, and the so-called divergence-correcting field, $\boldsymbol{u} = (\rho, \rho\mathbf{v}, E, \mathbf{B}, \psi)$, and the vectors $\mathbf{v} = (u, v, w)$ and $\mathbf{B} = (B_1, B_2, B_3)$ contain the velocities and magnetic field in x, y and z, respectively. The flux reads

$$\boldsymbol{f}(\boldsymbol{u}) = \begin{pmatrix} \rho\mathbf{v} \\ \rho\mathbf{v}\mathbf{v}^T + \mathbf{I}\left(p + \frac{1}{2}\|\mathbf{B}\|^2\right) - \mathbf{B}\mathbf{B}^T \\ \mathbf{v}\left(\frac{1}{2}\rho\|\mathbf{v}\|^2 + \frac{\gamma p}{\gamma - 1} + \|\mathbf{B}\|^2\right) + \mathbf{B}\left(c_h\psi - (\mathbf{v}\cdot\mathbf{B})\right) \\ \mathbf{v}\mathbf{B}^T - \mathbf{B}\mathbf{v}^T + \mathbf{I}c_h\psi \\ c_h\mathbf{B} \end{pmatrix},$$

where $\mathbf{I}$ is again the 3×3 identity matrix, γ is the heat capacity ratio, c_h is the hyperbolic divergence-cleaning speed, and $p = (\gamma - 1)(E - (\rho\|\mathbf{v}\|^2 - \|\mathbf{B}\|^2 - \psi^2)/2)$ is the gas pressure. Finally, the non-conservative term reads

$$\boldsymbol{\Upsilon} = (\nabla \cdot \mathbf{B})(0, \mathbf{B}, \mathbf{v}\cdot\mathbf{B}, \mathbf{v}, 0) + \left(0, \mathbf{0}, \psi(\mathbf{v}\cdot\nabla\psi), \mathbf{0}, \mathbf{v}\cdot\nabla\psi\right). \tag{7}$$

To construct a two-dimensional version of the GLM-MHD system, we replace $\mathbf{I}$ by a rectangular 3×2 identity matrix and neglect the flux in z. However, we keep the third component of the velocity and magnetic field because plasma systems admit three-dimensional electromagnetic interactions in two-dimensional problems. For details about the GLM-MHD system, we refer the reader to [52].

The non-conservative GLM-MHD system (**Eq. 6**) can be discretized using the collocation (**Eq. 2**) and modal (**Eq. 3**) formulations by replacing the volume term $\mathbf{F}$ and the surface term $\mathbf{f}^*$ [53]. In the collocation formulation the new terms read

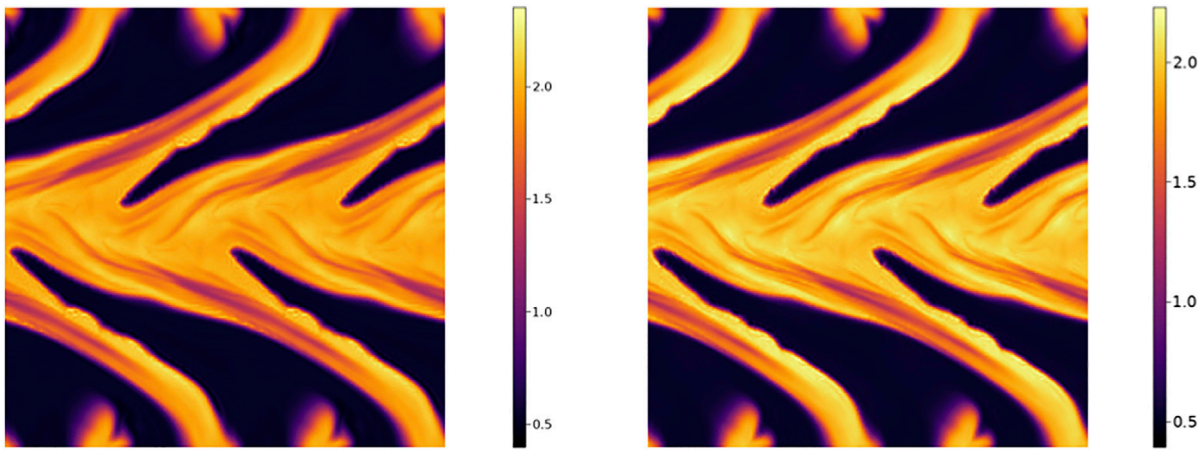


FIGURE 4 | Snapshots of density for the magnetized Kelvin-Helmholtz instability using an entropy stable Gauss DG scheme on uniform quadrilateral meshes.

$$\mathbf{F}_{ij} = \boldsymbol{f}_S\left(\mathbf{u}_i, \mathbf{u}_j\right) + \boldsymbol{\Phi}^{\diamond}\left(\mathbf{u}_i, \mathbf{u}_j\right),$$

$$\mathbf{f}^* = \begin{bmatrix} \boldsymbol{f}^*\left(\mathbf{u}_1^+, \mathbf{u}_1\right) + \boldsymbol{\Phi}^{\diamond}\left(\mathbf{u}_1^+, \mathbf{u}_1\right) \\ \boldsymbol{f}^*\left(\mathbf{u}_{N+1}, \mathbf{u}_{N+1}^+\right) + \boldsymbol{\Phi}^{\diamond}\left(\mathbf{u}_{N+1}, \mathbf{u}_{N+1}^+\right) \end{bmatrix}. \tag{8}$$

and in the modal formulation they read

$$\mathbf{F}_{ij} = \boldsymbol{f}_S\left(\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j\right) + \boldsymbol{\Phi}^{\diamond}\left(\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j\right),$$

$$\mathbf{f}^* = \begin{bmatrix} \boldsymbol{f}^*\left(\tilde{\mathbf{u}}_1^+, \tilde{\mathbf{u}}_1\right) + \boldsymbol{\Phi}^{\diamond}\left(\tilde{\mathbf{u}}_1^+, \tilde{\mathbf{u}}_1\right) \\ \boldsymbol{f}^*\left(\tilde{\mathbf{u}}_{N+1}, \tilde{\mathbf{u}}_{N+1}^+\right) + \boldsymbol{\Phi}^{\diamond}\left(\tilde{\mathbf{u}}_{N+1}, \tilde{\mathbf{u}}_{N+1}^+\right) \end{bmatrix}. \tag{9}$$

In addition to the symmetric two-point flux $\boldsymbol{f}_S$, we use a non-symmetric two-point term $\boldsymbol{\Phi}^{\diamond}$ to account for the non-conservative term in the equation. The following experiment uses the non-conservative term presented by Rueda-Ramírez et al. [53] and the entropy conservative flux of Hindenlang and Gassner [54], which is a natural extension of the entropy conservative, kinetic energy preserving, and pressure equilibrium preserving Euler flux of Ranocha [43, 44] to the GLM-MHD system.

3.2.1 Two Dimensional Magnetized Kelvin-Helmholtz Instability

To test the robustness of entropy projection schemes for the GLM-MHD system, we propose a modification of the Euler two-dimensional Kelvin-Helmholtz instability of **Section 3.1.1**. The domain is $[-1, 1]^2$ with the initial condition:

$$\begin{aligned} &\rho = \frac{1}{2} + \frac{3}{4}B, && p = 1, && \psi = 0, \\ &u = \frac{1}{2}(B - 1), && v = \frac{1}{10}\sin(2\pi x), && \omega = 0, \\ &B_1 = 0, && B_2 = 0.125, && B_3 = 0, \end{aligned} \tag{10}$$

where $B(x, y)$ is as defined in **Eq. 5**. Each solver is run until final time $T_{\text{final}} = 15$.

TABLE 2 | End time for simulations of the Kelvin-Helmholtz instability on quadrilateral and triangular meshes. On quadrilateral meshes, "collocation" refers to a nodal DGSEM discretization, while "entropy projection" refers to a method based on Gauss nodes. On triangular meshes, "collocation" refers to nodal SBP discretization, while "entropy projection" refers to a modal entropy stable DG method.

Degree	1	2	3	4	5	6	7
Solver							
KHI, quadrilateral mesh, $N_{\text{cells}} = 16$							
Collocation	15	4.807	3.769	4.433	3.737	3.369	3.642
Entropy projection	15	15	15	15	15	15	15
KHI, quadrilateral mesh, $N_{\text{cells}} = 32$							
Collocation	15	4.116	3.652	4.266	3.54	3.663	3.556
Entropy projection	15	15	15	15	15	15	15
KHI, triangular mesh, $N_{\text{cells}} = 16$							
Collocation	15	3.984	3.441	2.993	2.943	3.128	
Entropy projection	15	15	15	15	15	15	
KHI, triangular mesh, $N_{\text{cells}} = 32$							
Collocation	3.919	3.452	3.191	2.958	3.063	3.269	
Entropy projection	15	15	15	15	15	15	

Times colored blue correspond to simulations which did not crash and ran to completion, while times colored red denote simulations which did crash.

For this example, we set c_h as the maximum wave speed in the domain for the initial condition (**Eq. 10**) and keep it constant throughout the simulation. This standard way of selecting c_h has been shown to control the divergence error efficiently without affecting the time-step size [52, 55]. We observed that smaller values of c_h affect the robustness of the schemes for this problem, and higher values of c_h increase the stiffness of the problem which can also lead to a crash if the tolerance for the adaptive time-stepping method is set too loosely.

Figure 4 shows pseudocolor plots of the density at $T = 10$ for the magnetized Kelvin-Helmholtz instability problem obtained with the entropy stable Gauss DG using polynomial degrees $N = 3$

TABLE 3 | End time for simulations of the Rayleigh-Taylor instability on quadrilateral and triangular meshes. On quadrilateral meshes, "collocation" refers to a nodal DGSEM discretization, while "entropy projection" refers to a method based on Gauss nodes. On triangular meshes, "collocation" refers to nodal SBP discretization, while "entropy projection" refers to a modal entropy stable DG method.

Degree	1	2	3	4	5	6	7
Solver							
RTI, quadrilateral mesh, N_{cells} = 16							
Collocation	3.674	3.44	3.332	3.257	3.106	3.034	3.044
Entropy projection	15	15	15	15	15	15	15
RTI, quadrilateral mesh, N_{cells} = 32							
Collocation	3.996	3.144	3.44	3.155	3.031	2.972	2.976
Entropy projection	15	15	15	15	15	15	15
RTI, triangular mesh, N_{cells} = 16							
Collocation	4.297	2.87	3.238	3.229	2.927	2.881	
Entropy projection	15	15	15	15	15	15	
RTI, triangular mesh, N_{cells} = 32							
Collocation	3.6	2.896	3.197	3.227	3.032	2.778	
Entropy projection	15	15	15	15	15	15	

Times colored blue correspond to simulations which did not crash and ran to completion, while times colored red denote simulations which did crash.

TABLE 4 | End time for simulations of the Richtmeyer-Meshkov instability on quadrilateral and triangular meshes. On quadrilateral meshes, "collocation" refers to a nodal DGSEM discretization, while "entropy projection" refers to a method based on Gauss nodes. On triangular meshes, "collocation" refers to nodal SBP discretization, while "entropy projection" refers to a modal entropy stable DG method.

Degree	1	2	3	4	5	6	7
Solver							
RMI, quadrilateral mesh, N_{cells} = 16							
Collocation	30	30	27.96	24.94	8.851	8.853	8.85
Entropy projection	30	30	30	30	30	30	30
RMI, quadrilateral mesh, N_{cells} = 32							
Collocation	30	25.52	23.34	8.759	7.808	7.014	7.01
Entropy projection	30	30	30	30	30	30	30
RMI, triangular mesh, N_{cells} = 16							
Collocation	30	22.8	21.52	15.13	8.841	7.239	
Entropy projection	30	30	30	30	30	30	
RMI, triangular mesh, N_{cells} = 32							
Collocation	30	23.84	23.63	8.752	7.582	3.946	
Entropy projection	30	30	30	30	30	30	

Times colored blue correspond to simulations which did not crash and ran to completion, while times colored red denote simulations which did crash.

and $N = 7$ on uniform meshes of 64×64 and 32×32 quadrilateral elements, respectively. A comparison with **Figure 1** shows that the addition of a vertical magnetic field extends the flow features in the y direction and suppresses many of the vortical structures at $T = 10$. MHD turbulence eventually develops in the domain after $T = 10$, which leads to later crash times for this example. End times for each simulation can be found in **Table 3**.

3.3 Overview of Results

Tables 1,2,4,5 show what time the solver ran until for each solver on both quadrilateral and triangular meshes. We observe the pattern that, for degree $N > 1$, entropy stable methods which utilize the entropy projection appear be more robust than collocation-type schemes. Moreover, this pattern appears to hold independently of the polynomial degree and mesh size.

3.4 Dependence of Robustness on Atwood number

While the numerical results in the previous section indicate a difference between different entropy stable schemes, they do not provide insight into why and when this difference in robustness manifests. The goal of this section is to establish a relationship between robustness, the Atwood number (a measure of the density contrast), and the use of the "entropy projection" in an entropy stable scheme. We restrict our focus to the Kelvin-Helmholtz instability for this section.

The results presented so far are somewhat unexpected, as the robustness of high order entropy stable DG schemes has been documented for a variety of flows where shocks and turbulent features are present [7–9, 13]. In this section, we conjecture that the documented differences in robustness are due to the presence of both small-scale under-resolved features and significant variations in the density. For example, entropy stable DGSEM methods are known to be very robust for the Taylor-Green vortex, where the density is near-constant throughout the duration of the simulation.

We examine the connection between density contrast and robustness by parametrizing the initial condition by the Atwood number. Given a stratified fluid with two densities ρ_1, ρ_2, the Atwood number is defined as

$$A = \frac{\rho_2 - \rho_1}{\rho_1 + \rho_2} \in [0, 1),$$

where it is assumed that $\rho_2 \geq \rho_1$. For a constant-density flow, $A = 0$, while $A \to 1$ indicates a flow with very large density contrasts. We investigate the behavior of different entropy stable methods for a version of the Kelvin-Helmholtz instability parametrized by the Atwood number A:

$$\begin{aligned} \rho_1 &= 1 & \rho_2 &= \rho_1 \frac{1+A}{1-A} \\ \rho &= \rho_1 + B(\rho_2 - \rho_1) & p &= 1 \\ u &= B - \frac{1}{2} & v &= \frac{1}{10}\sin(2\pi x) \end{aligned}$$

Figure 5 shows the crash times for the Kelvin-Helmholtz instability using various entropy stable solvers at polynomial degrees 3 and 7. For quadrilateral meshes, we utilize entropy stable DGSEM solvers and entropy stable Gauss DG solvers. For triangular meshes, we utilize entropy stable multi-dimensional SBP solvers and entropy stable modal DG solvers. The DGSEM and SBP solvers are collocation-type schemes, while Gauss and modal DG solvers introduce the entropy projection.

For degree 3 quadrilateral solvers, we utilize a 32×32 mesh, while for degree 7 quadrilateral solvers, we utilize a 16×16 mesh. The mesh resolution is halved for polynomial degree 7 simulations so that the total number of degrees of freedom is kept constant. For triangular solvers, we again use 32×32 and 16×16 uniform meshes, but we compare polynomial degrees 3 and 6, as SBP quadrature rules are available only up to degree 6 in

TABLE 5 | End time for simulations of the magnetized Kelvin-Helmholtz instability on quadrilateral and triangular meshes. On quadrilateral meshes, "collocation" refers to a nodal DGSEM discretization, while "entropy projection" refers to a method based on Gauss nodes. On triangular meshes, "collocation" refers to nodal SBP discretization, while "entropy projection" refers to a modal entropy stable DG method.

Degree	1	2	3	4	5	6	7
Solver							
MHD KHI, quadrilateral mesh, $N_{\text{cells}} = 16$							
Collocation	15	15	11.503	10.988	10.315	10.230	10.270
Entropy projection	15	15	15	15	15	15	15
MHD KHI, quadrilateral mesh, $N_{\text{cells}} = 32$							
Collocation	15	11.639	11.048	11.111	11.483	10.169	10.919
Entropy projection	15	15	15	15	15	15	15
MHD KHI, triangular mesh, $N_{\text{cells}} = 16$							
Collocation	12.846	13.797	10.626	10.212	10.990	9.973	
Entropy projection	15	15	15	15	15	15	
MHD KHI, triangular mesh, $N_{\text{cells}} = 32$							
Collocation	14.875	11.121	9.748	10.081	10.307	10.219	
Entropy projection	15	15	15	15	15	15	

Times colored blue correspond to simulations which did not crash and ran to completion, while times colored red denote simulations which did crash.

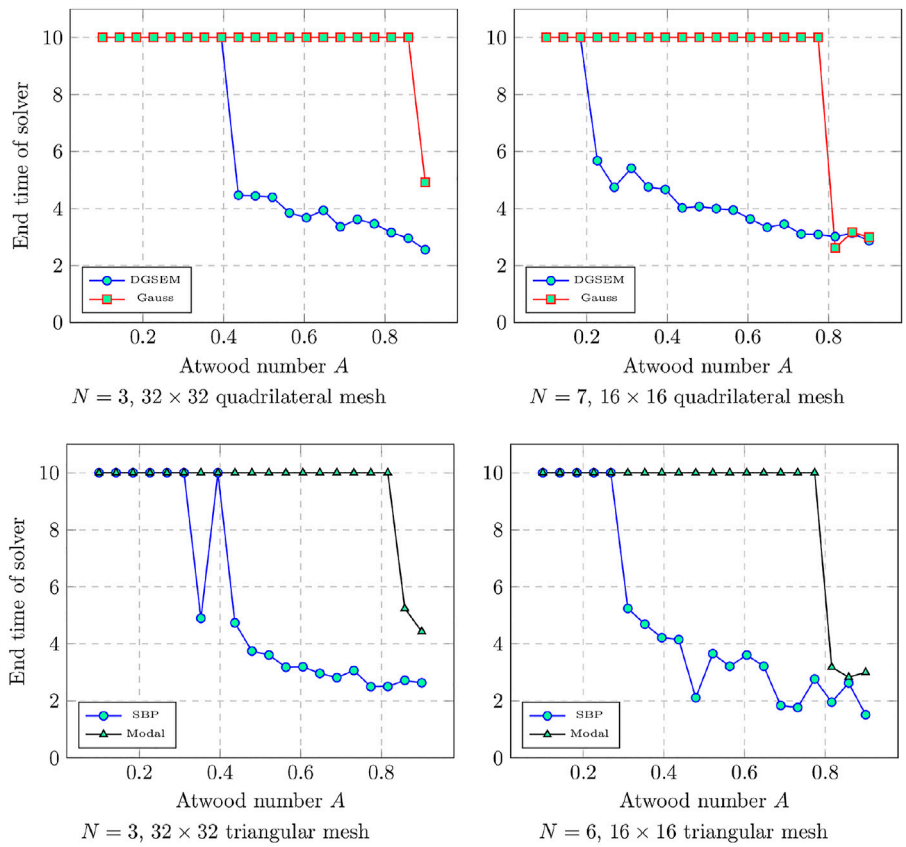


FIGURE 5 | Final times a solver an until as a function of Atwood number for the Kelvin-Helmholtz instability for DGSEM and various entropy stable solvers. End times less than final time $T_{\text{final}} = 10$ indicate a crash.

Trixi.jl. We run up to time $T_{\text{final}} = 10$ for $A \in [0.1, 0.9]$ and report the times each simulation ran until. For degree $N = 3$, we observe that schemes which involve the entropy projection runs until the final time $T_{\text{final}} = 10$. Collocation-type schemes run to completion for low Atwood numbers, but crash earlier and earlier as the Atwood number increases. At degree $N = 7$, we observe that while both collocation solvers and entropy projection solvers crash at higher Atwood numbers, entropy projection solvers begin

crashing at higher Atwood numbers. For example, on quadrilateral meshes, DGSEM crashes around Atwood number 0.3, while Gauss solvers crash around Atwood number 0.7. We note that crash times for entropy projection schemes also tend to depend on the adaptive time-stepping tolerance. For example, for $N = 3$ and a 32^2 mesh, Gauss collocation runs stably to $T_{\text{final}} = 10$ if the absolute and relative tolerances are reduced to 10^{-9}. The same is not true of entropy stable collocation-type schemes.

To provide another point of comparison, we ran simulations using an entropy stable DGSEM solver with sub-cell finite volume shock capturing [56] with Zhang-Shu positivity-preserving limiting for the density and pressure [57, 58], which we refer to as DGSEM-SC-PP for shock capturing and positivity preservation.[2] The entropy stable sub-cell finite volume-based shock capturing scheme utilizes a blending coefficient parameter $\alpha \leq \alpha_{\max}$ [56]. For these experiments, we set $\alpha_{\max} = 0.005$, which implies that the low order finite volume solution constitutes at most 0.5% of the final blended solution. Despite the fact that this shock capturing is very weak, the resulting solver greatly improves robustness and enables long simulation times: for $N = 3$ and a 32×32 mesh, DGSEM-SC-PP runs stably to time $T_{\text{final}} = 10$ for Atwood numbers up to 0.99. However, we have also observed that the minimum value of $\alpha_{\max}$ necessary to avoid solver failure depends on the mesh resolution. For example, for $N = 3$ and a 64×64 mesh, we observe that DGSEM-SC-PP with $\alpha_{\max} = 0.005$ crashes around $t = 6.4871$.

Remark 2. We note that DGSEM with $\alpha_{\max} = 0.005$ shock capturing but no positivity preservation is not robust for the Kelvin-Helmholtz instability. For the initial condition (**Eq. 4**), $N = 3$, and a 64×64 mesh, DGSEM with shock capturing crashes around time $t = 4.8891$. For $N = 7$ and a 32×32 mesh, DGSEM with shock capturing crashes around time $t = 5.0569$. In contrast, DGSEM with only positivity preservation results in the simulation stalling due to a very small time-step.

4 THE ROLE OF THE ENTROPY PROJECTION

4.1 Is Robustness Due Only to the Entropy Projection?

While the numerical results up to this point indicate that there is a significant difference in robustness for different entropy stable schemes, it is not yet clear that the increased robustness is due to the entropy projection. For example, the numerical experiments in **Section 3** compare entropy stable Gauss DG schemes to DGSEM on tensor product meshes and entropy stable "modal" DG methods to SBP schemes on triangular meshes. In both cases, a collocation scheme is compared to a scheme with higher accuracy numerical integration. Thus, it is not immediately clear whether the difference in robustness is due to the entropy projection or other factors such as the quadrature accuracy. We investigate whether the quadrature accuracy has a significant effect on stability by testing two additional variants of entropy stable DGSEM schemes on quadrilateral meshes. These schemes are purposefully constructed to be "bad" methods (in the sense that they introduce additional work without improving the expected accuracy), and are intended only to introduce the entropy projection. Both have quadrature accuracy similar to or lower than entropy stable DGSEM methods.

The first scheme utilizes LGL points for volume quadrature, but utilizes $(N + 1)$ point Clenshaw-Curtis quadrature at the faces. This scheme can be directly derived from a modal formulation and (despite the lower polynomial exactness of Clenshaw-Curtis quadrature) can be shown to be entropy stable on affine quadrilateral meshes using the analysis in [21]. In order to retain entropy stability, the solution must be evaluated using the entropy projection at face nodes. We argue that the use of Clenshaw-Curtis quadrature does not result in a significant increase in quadrature accuracy over LGL quadrature: while Clenshaw-Curtis quadrature has been shown to be similar to Gauss quadrature for integration of analytic functions [61], for lower numbers of points we observe that the accuracy is comparable to LGL quadrature. Moreover, it was argued in [62] that increasing quadrature accuracy only for surface integrals or only for volume integrals does not provide sufficient anti-aliasing. We refer to this method as "DGSEM with face-based entropy projection" in **Figure 6**.

Remark 3. We note that one can also build an entropy stable scheme from a combination of LGL volume points and Gauss face points. While this method possesses much of the simplicity and advantageous features of entropy stable DGSEM methods while also displaying improved robustness, this method results in a suboptimal rate of convergence by one degree [21].

The second scheme we test is similar to the staggered scheme of [17]. However, while the original scheme of Parsani et al. combines degree N Gauss points with degree $(N + 1)$ LGL points, we combine degree N Gauss points with degree N LGL points. This is a "useless" staggering in that it does not increase the accuracy of integration compared with DGSEM, and is intended only to introduce the entropy projection into the formulation.[3] We refer to this method as "DGSEM with volume-based entropy projection" in **Figure 6**.

Figure 6 shows snapshots of the density for the Kelvin-Helmholtz instability for a degree $N = 3$ mesh of 64×64 elements for each method. While the plots for the Gauss DG and DGSEM with face-based entropy projection have qualitative similarities, we observe that DGSEM with volume-based entropy

[2]For DGSEM-SC-PP, we utilize a 4-stage 3rd order adaptive strong stability preserving (SSP) Runge-Kutta time-stepping method [59, 60] with stepsize controller and efficient implementation of [40], which is necessary to ensure fully discrete positivity.

[3]This scheme can also be derived by beginning with an entropy stable DGSEM scheme and replacing the diagonal LGL mass matrix with the fully integrated dense mass matrix computed using Gauss quadrature. The resulting scheme can be made entropy stable by evaluating the spatial formulation using the entropy projection. More specifically, the appropriate entropy projection for this setting interpolates the entropy variables at Gauss nodes, then interpolates to LGL nodes.

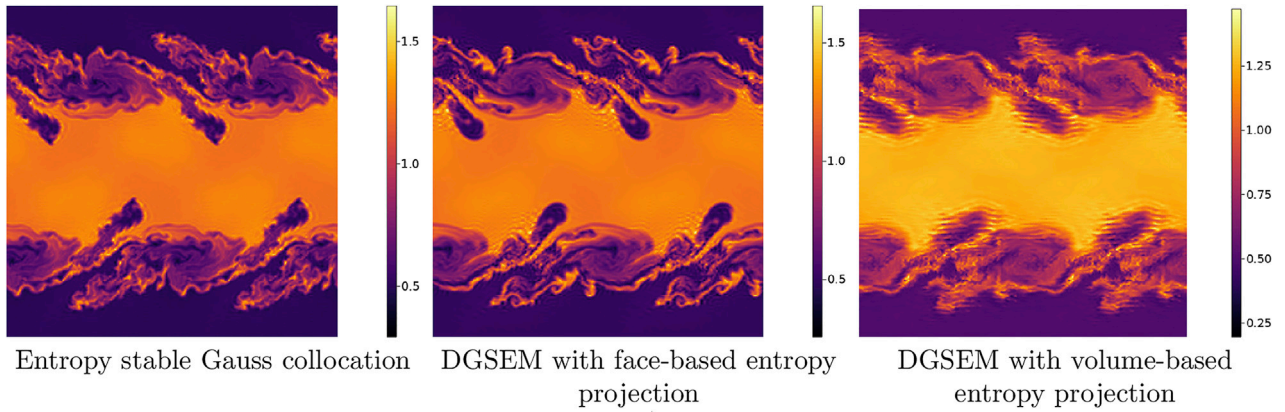


FIGURE 6 | Degree $N = 3$ and 64×64 grid Kelvin-Helmholtz simulations at $T = 5$. All methods run until $T = 25$, while DGSEM crashes at $T \approx 3.5$.

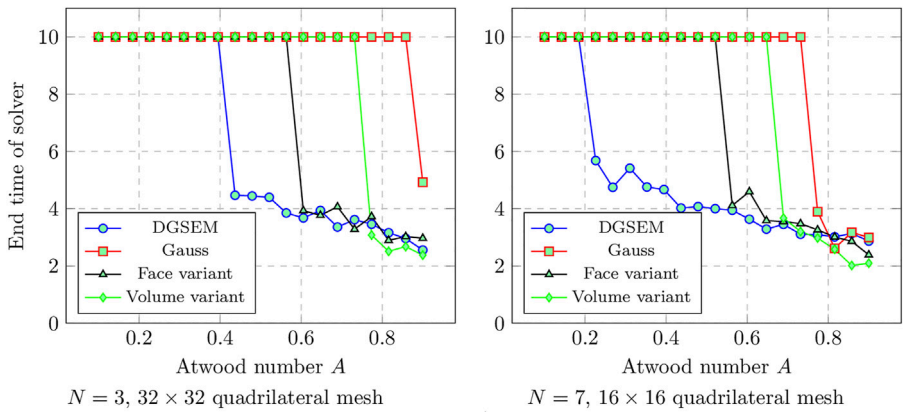


FIGURE 7 | Final times a solver an until as a function of Atwood number for the Kelvin-Helmholtz instability for DGSEM and different variants of entropy stable solvers based on the entropy projection. End times less than final time $T_{\text{final}} = 10$ indicate a crash.

projection results in a noisier solution. This may be due to inconsistency in terms of accuracy between the two quadrature rules used (e.g., $(N + 1)$ point LGL and Gauss quadratures). However, all three entropy projection schemes remain stable, and we have verified that they are able to run until $T = 25$ without crashing.

We also compute crash times for each method for the Kelvin-Helmholtz instability with Atwood numbers $A \in [0.1, 0.9]$. These crash times are also compared to crash times of an entropy stable DGSEM method. These computations are performed on both a degree $N = 3$ mesh of 32×32 elements, as well as a degree $N = 7$ mesh of 16×16 elements. **Figure 7** plots the crash times for each method. We observe that all schemes which involve the entropy projection run stably for a wider range of Atwood numbers than entropy stable DGSEM, and that this effect becomes even more pronounced for degree $N = 7$. However, for both the $N = 3$ and $N = 7$ experiments, the entropy stable Gauss schemes are stable for the widest ranges of Atwood numbers.

These results indicate that incorporating the entropy projection does have a significant effect on the robustness of an entropy stable method, but that the entropy projection is not the only factor which impacts robustness. However, a detailed analysis of factors such as quadrature accuracy is out of the scope of this current work.

4.2 Why Is There a Difference in Robustness for Different Entropy Stable Methods?

While the results from previous sections suggest that the entropy projection plays a role in the robustness of an entropy stable scheme, it is not clear *why* it plays a role. While we do not have a thorough theoretical understanding of the entropy projection, initial experiments indicate that entropy projection schemes behave most differently from collocation schemes when the solution is either under-resolved or have near-zero density or pressure.

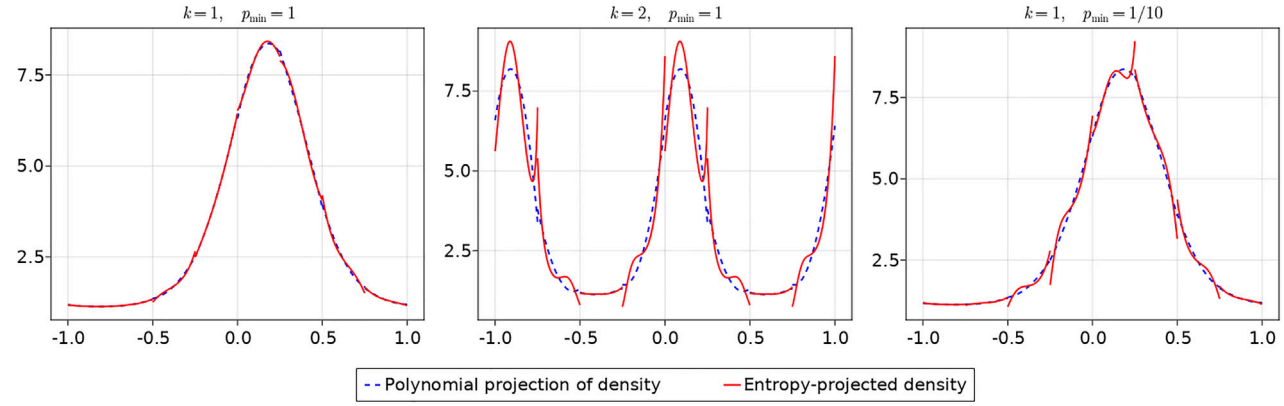


FIGURE 8 | Illustration of the effect of larger k (under-resolution) and smaller $p_{\min}$ (near-vacuum state) on the entropy projection. A degree $N = 2$ approximation and mesh of 8 elements were used.

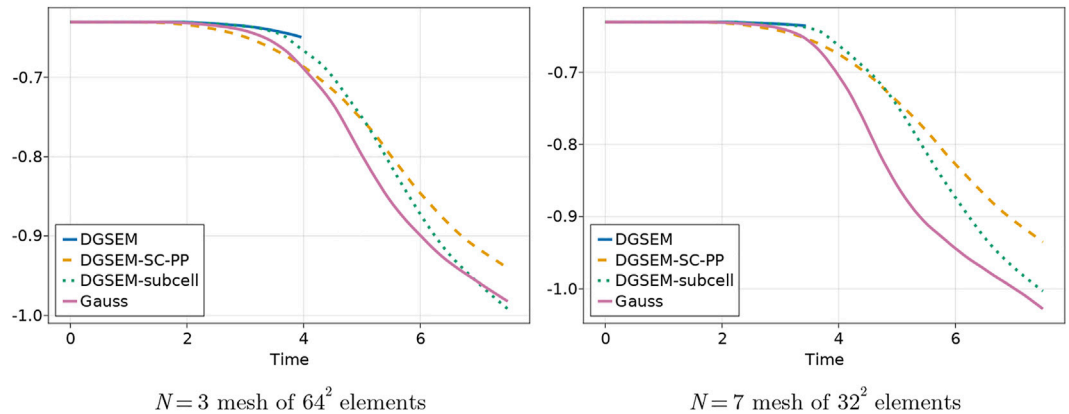


FIGURE 9 | Evolution of entropy over time for the Kelvin-Helmholtz instability.

We illustrate the aforementioned behavior of the entropy projection using the one-dimensional compressible Euler equations. The conservative variables for the Euler equations are density, momentum, and total energy $(\rho, \rho u, E)$. Let $s(\boldsymbol{u}) = \log(p/\rho^\gamma)$ denote the specific entropy. The entropy variables for the convex entropy $S(\boldsymbol{u}) = -\rho s(\boldsymbol{u})/(\gamma - 1)$ are given by

$$\boldsymbol{v}(\boldsymbol{u}) = \left(\frac{\gamma - s}{\gamma - 1} - \frac{\rho u^2}{2p}, \frac{\rho u}{p}, -\frac{\rho}{p}\right).$$

Recall that the main steps of the entropy projection are as follows:

(1) Evaluate the entropy variables using degree N polynomial approximations of the conservative variables
(2) Compute the quadrature-based L^2 projection of the entropy variables to degree N polynomials
(3) Re-evaluate the conservative variables in terms of the projected entropy variables.

These re-evaluated conservative variables are then used to compute contributions from an entropy stable DG formulation.

It was demonstrated numerically in [19] that the entropy projection is high order accurate for sufficiently regular solutions. However, the behavior of the entropy projection was not explored for under-resolved or near-vacuum solution states. We illustrate this behavior using the following solution state:

$$\rho = 1 + e^{2\sin(1+k\pi x)}, \qquad u = \frac{1}{10}\cos(1 + k\pi x), \qquad p = p_{\min} + \frac{1}{2}\left(1 - \cos\left(k\pi x - \frac{1}{4}\right)\right), \tag{11}$$

where $p_{\min} > 0$ is the minimum pressure, and k is a parameter which controls the frequency of oscillation. As k increases, the solution states in **Eq. 11** become more and more difficult to resolve, and as $p_{\min} \to 0$, the solution approaches vacuum and the entropy approaches non-convexity.

Figure 8 illustrates the effect of increasing k and decreasing $p_{\min}$ on the entropy projected conservative variables for a degree $N = 2$ approximation on a coarse mesh of eight elements. As k

TABLE 6 | End time for entropy conservative simulations of the Taylor-Green vortex on hexahedral meshes.

Degree	1	2	3	4	5	6	7
Solver							
$N_{\text{cells}} = 2^3$							
DGSEM	20	20	20	20	16.4	7.704	7.482
Gauss	20	20	20	20	20	20	20
CGSEM	20	20	20	20	20	20	20
$N_{\text{cells}} = 4^3$							
DGSEM	20	20	20	20	10.31	5.792	5.46
Gauss	20	20	20	20	20	20	20
CGSEM	20	20	20	20	20	20	20
$N_{\text{cells}} = 8^3$							
DGSEM	20	20	20	20	6.035	5.29	5.02
Gauss	20	20	20	20	20	20	20
CGSEM	20	20	20	20	20	20	17.5

Times colored blue correspond to simulations which did not crash and ran to completion, while times colored red denote simulations which did crash.

increases and the solution becomes under-resolved, the entropy projection develops large jumps at the interface. Similarly, as $p_{\min}$ decreases from 1 to 1/10, the entropy projection develops large jumps at the interface. We note that for both increased k and decreased $p_{\min}$, spikes do not appear in the interior of the element.

This indicates that the error in the entropy projection is influenced by both the numerical resolution and how close the entropy is to becoming non-convex. We denote the continuous entropy projection by $\tilde{\boldsymbol{u}} = \boldsymbol{u}(\Pi_N \boldsymbol{v}(\boldsymbol{u}_h))$. Then, by the mean value theorem, we can bound the difference between the conservative and entropy-projected variables

$$\|\boldsymbol{u}_h - \tilde{\boldsymbol{u}}\|_{L^\infty} = \|\boldsymbol{u}_h - \boldsymbol{u}(\Pi_N \boldsymbol{v}(\boldsymbol{u}_h))\|_{L^\infty} \leq \left\|\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{v}}\right\|_{L^\infty} \|\boldsymbol{v}(\boldsymbol{u}_h) - \Pi_N \boldsymbol{v}(\boldsymbol{u}_h)\|_{L^\infty},$$

where $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{v}}$ is evaluated at some intermediate state between $\boldsymbol{u}_h$ and $\tilde{\boldsymbol{u}}$. The latter term in the bound $\|\boldsymbol{v}(\boldsymbol{u}_h) - \Pi_N \boldsymbol{v}(\boldsymbol{u}_h)\|$ is small when the entropy variables are well-resolved, which we expect to be true when the solution is well-resolved and the mapping between conservative and entropy variables is well-conditioned. Conversely, high frequency components of the solution are often amplified when $\boldsymbol{v}(\boldsymbol{u})$ is highly nonlinear or the solution is under-resolved (this is the motivation behind filtering for stabilization [63–65]), and we expect $\|\boldsymbol{v}(\boldsymbol{u}_h) - \Pi_N \boldsymbol{v}(\boldsymbol{u}_h)\|$ to be large for such settings. The former term $\|\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{v}}\|$ is large when the mapping between conservative and entropy variables is nearly singular, which occurs when the entropy is nearly non-convex (for example, near-vacuum states).

4.2.1 What Role Does Entropy Dissipation Play?

The previous section illustrates that entropy projection schemes are likely to differ from collocation schemes most when the solution is under-resolved or has near-zero density or pressure. Moreover, since the entropy projected variables in **Figure 8** display spikes at the interfaces, it seems possible that the entropy projection would change the manner in which entropy dissipative interface dissipation terms are triggered. To test this hypothesis, we compute the evolution of entropy over time for the Kelvin-Helmholtz instability using both entropy stable Gauss DG and DGSEM-SC-PP, which is an entropy stable DGSEM with a shock capturing technique that consists in blending a sub-cell finite volume scheme with the DGSEM in an element-wise manner [56] and Zhang-Shu's positivity preserving limiter [57, 58]. The blending of the finite volume scheme is capped at 0.5% in order to avoid unnecessary numerical dissipation. We also compare entropy evolution for a scheme that blends a sub-cell finite volume scheme with the DGSEM in a subcell-wise manner [66], which we refer to as DGSEM-subcell. The blending factors are chosen for each node (or subcell) to enforce lower bounds on density and pressure based on the low order solution, $\rho \geq 0.1\ \rho^{\text{FV}}$, $p \geq 0.1\ p^{\text{FV}}$. For this choice of lower bound, we observe high order accuracy for a two-dimensional sinusoidal entropy wave [67]. While this scheme is not provably entropy stable, it was demonstrated numerically in [66] that the use of subcell blending factors requires significantly lower levels of limiting compared with an element-wise limiting factor.

Figure 9 shows the evolution of the integrated entropy over the entire domain (which we have shifted to be positive) for the Kelvin-Helmholtz instability. Since periodic boundary conditions are used, the integrated entropy for the semi-discrete formulation can be proven to decrease over time. We observe that all four methods display similar entropy dissipation behavior until time $t \approx 1.2$, after which DGSEM shows less entropy dissipation than either Gauss or DGSEM-SC-PP. However, while DGSEM-SC-PP initially dissipates more entropy than Gauss DG, the entropy dissipation for Gauss DG increases and overtakes that of DGSEM-SC-PP around time $t \approx 4$. Since entropy dissipation in both Gauss DG and DGSEM-SC-PP schemes is triggered by under-resolved flows (either through a modal indicator or through jump penalization terms) and since the Kelvin-Helmholtz instability generates increasingly small scales at larger times, this suggests that entropy dissipation for Gauss DG may be activated more strongly but at smaller scales than DGSEM-SC-PP. In contrast, Gauss DG dissipates more global entropy than DGSEM-subcell, though DGSEM-subcell eventually catches up to Gauss DG for $N = 3$.

Our initial hypothesis was that the entropy projection in Gauss DG schemes results in larger interface jumps, which would trigger more entropy dissipation through jump penalization terms. However, this does not appear to be consistent with numerical results for entropy conservative schemes. To test these schemes, we focus on the three-dimensional Taylor-Green vortex. We note that the observed loss of robustness stands in stark contrast to the observed robustness of high order entropy stable and split-form DGSEM for the Taylor-Green vortex [8, 13, 22]. This can be explained by the fact that the density remains near-constant over time for the Taylor-Green vortex; for a Kelvin-Helmholtz initial condition with a constant density, DGSEM runs stably up to final time $T = 25$ for each of the previous numerical settings. Thus, while the Taylor-Green vortex generates small-scale flow features, it is a more benign test case when evaluating the robustness of high order entropy stable DG schemes.

However, when using a purely entropy conservative scheme (which can be constructed by utilizing entropy conservative

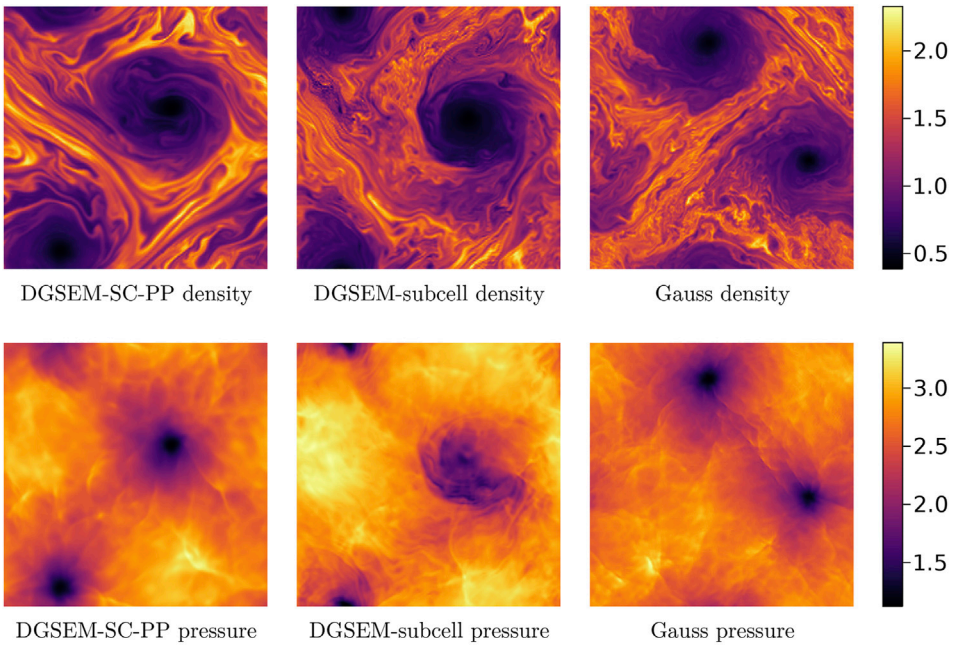


FIGURE 10 | Density and pressure for the Kelvin-Helmholtz instability at $T_{\text{final}} = 25$ on a $N = 3$ mesh of 64^2 elements.

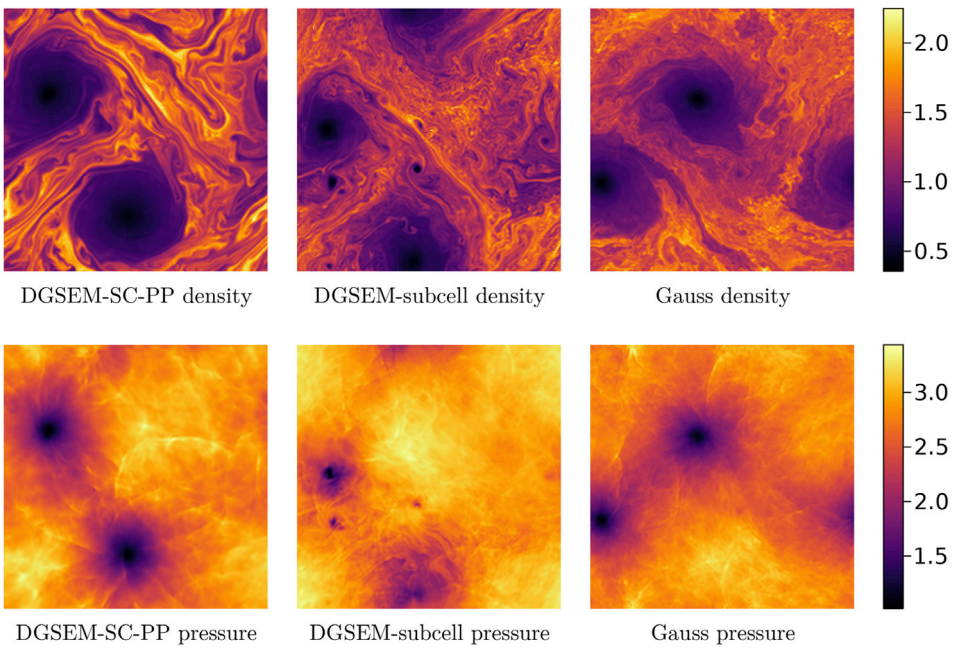


FIGURE 11 | Density and pressure for the Kelvin-Helmholtz instability at $T_{\text{final}} = 25$ on a $N = 7$ mesh of 32^2 elements.

interface fluxes), DGSEM methods can display non-robust behavior for the Taylor-Green vortex. We run the Taylor-Green vortex to final time $T_{\text{final}} = 20$ using a variety of entropy conservative schemes: DGSEM, Gauss DG, as well as an entropy stable C^0 continuous Galerkin spectral element method (CGSEM) and a periodic finite difference method. We note that, because an entropy conservative scheme can be constructed given any summation-by-parts or skew-symmetric operator [12, 14, 26], we are able to implement an entropy conservative C^0 continuous spectral element method and periodic finite difference method by constructing global difference

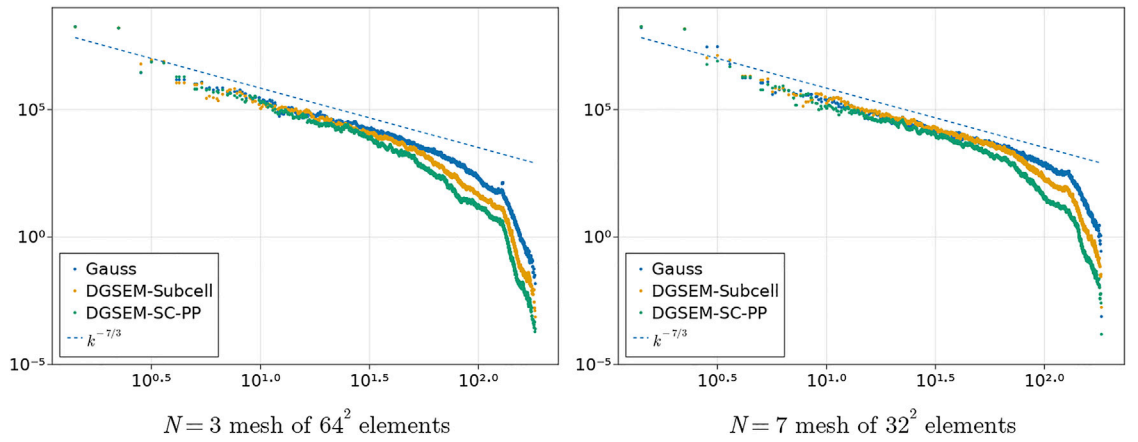


FIGURE 12 | Weighted power spectra for DGSEM-Subcell, entropy stable DGSEM-SC-PP, and entropy stable Gauss DG schemes.

operators from the tensor product of one-dimensional operators. These one-dimensional operators are provided by the Julia library SummationByPartsOperators.jl [68].

Table 6 shows the end simulation time for each solver. We observe that again, despite the absence of any entropy dissipation, the Gauss DG solver is more robust than the DGSEM solver. The continuous spectral element solver CGSEM is also significantly more robust than the DGSEM solver, though it does lose robustness at higher orders and finer grid resolutions. We also ran periodic finite difference operators for grids with 4, 6, 8, 10, 12 nodes in each dimension with orders of accuracy 2, 4, 6, 8, 10. We observe that the periodic finite difference operator is as robust as the Gauss DG solver: for every grid resolution and order specified, the finite difference solver ran up to the final time $T_{\text{final}} = 20$.

These experiments indicate that robustness for schemes involving the entropy projection is not solely due to the entropy dissipative terms. These experiments also show that robustness is improved for CGSEM and periodic finite difference solvers, neither of which contains interface terms. Since these results are on relatively coarse resolutions and utilize an entropy conservative scheme (when most practical schemes are entropy stable), further numerical experiments are necessary to carefully analyze the effect that different discretizations have on robustness.

5 APPLICATIONS TOWARD UNDER-RESOLVED SIMULATIONS

We conclude the paper with a discussion on a comparison between three schemes which include dissipative terms (entropy stable Gauss DG, entropy stable DGSEM-SC-PP, and DGSEM-subcell) for an under-resolved simulation. We run the Kelvin-Helmholtz instability using the initial condition (**Eq. 4**), but modify the y-velocity perturbation to break symmetry of the resulting flow

$$v = \frac{1}{10}\sin(2\pi x)\left(1 + \frac{1}{100}\sin(\pi x)\sin(\pi y)\right).$$

We run the simulation up to final time $T_{\text{final}} = 25$. We use both a degree $N = 3$ mesh of 64×64 elements and a degree $N = 7$ mesh of 32×32 elements, each of which contains the same number of degrees of freedom. Due to the sensitivity of the Kelvin-Helmholtz instability problem and the long time window of the simulation, the results for each scheme are qualitatively very different.

Figures 10, **11** show snapshots of density and pressure for the entropy stable DGSEM-SC-PP and Gauss DG schemes. We observe that in both cases, the flow scales present in the DGSEM-SC-PP scheme are noticeably larger than those observed in the Gauss scheme. This is notable because the DGSEM-SC-PP scheme applies a very small amount of shock capturing: dissipation is added by blending the high order scheme with a low order finite volume scheme, and the amount of the blended low order solution is capped at 0.5%. However, even a small amount of dissipation produces a noticeable change on small-scale features in the resulting flow. We also observe the presence of shocklets or compression waves in the pressure, which mirror observations made in [69].[4]

For $N = 3$, the scales observed in DGSEM-subcell scheme are noticeably smaller than those of DGSEM-SC-PP but similar to those of the Gauss DG scheme. For $N = 7$, the scales observed in the DGSEM-subcell scheme are again smaller than those of DGSEM-SC-PP, but appear to be slightly larger than those of the Gauss DG scheme. To avoid qualitative speculation, we compare these flows by computing the angle-averaged power spectra of the velocity weighted by $\sqrt{\rho}$ at final time $T_{\text{final}} = 25$ [70, 71]. We follow [3, 7] and generate a grid of uniformly spaced points by evaluating the degree N polynomial solution at $(N + 1)$ equally spaced points along each dimension in the interior of each element of a uniform Cartesian mesh. The power spectra can then be computed from a fast Fourier transform of the resulting data. **Figure 12** shows the power spectra, which appear

[4]We note that these "shocklets" are not strictly shock waves, as the flow is not supersonic.

consistent with a $k^{-7/3}$ rate of decay from two-dimensional turbulence theory [71]. Moreover, we observe that the entropy stable Gauss DG scheme retains more energetic information than both DGSEM-SC-PP and DGSEM-subcell, though a spurious spike in the energy for Gauss DG schemes is observed near the higher wavenumbers for $N = 3$.

6 CONCLUSION

This paper shows that for variable density flows which generate small-scale features, there are differences in robustness between entropy stable schemes which incorporate the entropy projection and those which do not. These differences in robustness are observed to depend on the Atwood number (measuring the density contrast) and persist across a range of polynomial degrees, mesh resolutions, and types of discretization. However, the mechanisms behind improved robustness for entropy projection schemes are currently unknown.

We note that any conclusions drawn concerning the robustness of DGSEM and Gauss DG should be restricted to the instability-type problems studied here. These results do not imply that Gauss is uniformly more robust than DGSEM. Moreover, Gauss schemes are more computationally expensive than DGSEM schemes and result in smaller maximum stable timesteps [22, 72–74], so the appropriate scheme will depend on the use case.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: https://github.com/trixi-framework/paper-2022-robustness-entropy-projection.

AUTHOR CONTRIBUTIONS

All authors contributed to the conception and design of the paper. JC, HR, and AR-R contributed numerical experiments. JC drafted the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

HR was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2044-390685587, Mathematics Münster: Dynamics-Geometry-Structure. This work has received funding from the European Research Council through the ERC Starting Grant "An Exascale aware and Uncrashable Space-Time-Adaptive Discontinuous Spectral Element Solver for Non-Linear Conservation Laws" (Extreme), ERC grant agreement no. 714487 (GG and AR-R). TW was supported in part by the Exascale Computing Project, a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering, and early testbed platforms, in support of the nation's exascale computing imperative. TW was also supported in part by the JC Faculty Chair in Science at Virginia Tech. JC gratefully acknowledges support from the National Science Foundation under award DMS-CAREER-1943186. GG and ARR acknowledge funding through the Klaus-Tschira Stiftung via the project "HiFiLab."

ACKNOWLEDGMENTS

This work used the Extreme Science and Engineering Discovery Environment (XSEDE) Expanse at the San Diego Supercomputer Center through allocation TG-MTH200014 [75]. This work was performed on the Cologne High Efficiency Operating Platform for Sciences (CHEOPS) at the Regionales Rechenzentrum Köln (RRZK) and on the group cluster ODIN. We thank RRZK for the hosting and maintenance of the clusters. The authors also thank Fabian Föll for providing the initial condition for the Richtmeyer-Meshkov instability.

REFERENCES

1. Gassner GJ, Beck AD. On the Accuracy of High-Order Discretizations for Underresolved Turbulence Simulations. *Theor Comput Fluid Dyn* (2013) 27: 221–37. doi:10.1007/s00162-011-0253-7
2. Beck AD, Bolemann T, Flad D, Frank H, Gassner GJ, Hindenlang F, et al. High-order Discontinuous Galerkin Spectral Element Methods for Transitional and Turbulent Flow Simulations. *Int J Numer Methods Fluids* (2014) 76:522–48. doi:10.1002/fld.3943
3. Moura RC, Mengaldo G, Peiró J, Sherwin SJ. On the Eddy-Resolving Capability of High-Order Discontinuous Galerkin Approaches to Implicit LES/under-resolved DNS of Euler Turbulence. *J Comput Phys* (2017) 330: 615–23. doi:10.1016/j.jcp.2016.10.056
4. Fernandez P, Nguyen NC, Peraire J. *On the Ability of Discontinuous Galerkin Methods to Simulate Under-resolved Turbulent Flows* (2018). *arXiv preprint arXiv:1810.09435*.
5. Lv Y, Ma PC, Ihme M. On Underresolved Simulations of Compressible Turbulence Using an Entropy-Bounded DG Method: Solution Stabilization, Scheme Optimization, and Benchmark against a Finite-Volume Solver. *Comput Fluids* (2018) 161:89–106. doi:10.1016/j.compfluid.2017.11.016
6. Flad D, Gassner G. On the Use of Kinetic Energy Preserving DG-Schemes for Large Eddy Simulation. *J Comput Phys* (2017) 350:782–95. doi:10.1016/j.jcp.2017.09.004
7. Winters AR, Moura RC, Mengaldo G, Gassner GJ, Walch S, Peiro J, et al. A Comparative Study on Polynomial Dealiasing and Split Form Discontinuous Galerkin Schemes for Under-resolved Turbulence Computations. *J Comput Phys* (2018) 372:1–21. doi:10.1016/j.jcp.2018.06.016
8. Rojas D, Boukharfane R, Dalcin L, Fernández DCDR, Ranocha H, Keyes DE, et al. On the Robustness and Performance of Entropy Stable Discontinuous Collocation Methods. *J Comput Phys* (2021) 426:109891. doi:10.1016/j.jcp.2020.109891
9. Parsani M, Boukharfane R, Nolasco IR, Fernández DCDR, Zampini S, Hadri B, et al. High-order Accurate Entropy-Stable Discontinuous Collocated Galerkin Methods with the Summation-By-Parts Property for Compressible CFD Frameworks: Scalable SSDC

Algorithms and Flow Solver. *J Comput Phys* (2021) 424:109844. doi:10.1016/j.jcp.2020.109844
10. Fjordholm US, Mishra S, Tadmor E. Arbitrarily High-Order Accurate Entropy Stable Essentially Nonoscillatory Schemes for Systems of Conservation Laws. *SIAM J Numer Anal* (2012) 50:544–73. doi:10.1137/110836961
11. Fisher TC, Carpenter MH. High-order Entropy Stable Finite Difference Schemes for Nonlinear Conservation Laws: Finite Domains. *J Comput Phys* (2013) 252:518–57. doi:10.1016/j.jcp.2013.06.014
12. Carpenter MH, Fisher TC, Nielsen EJ, Frankel SH. Entropy Stable Spectral Collocation Schemes for the Navier–Stokes Equations: Discontinuous Interfaces. *SIAM J Scientific Comput* (2014) 36:B835–B867. doi:10.1137/130932193
13. Gassner GJ, Winters AR, Kopriva DA. Split Form Nodal Discontinuous Galerkin Schemes with Summation-By-Parts Property for the Compressible Euler Equations. *J Comput Phys* (2016) 327:39–66. doi:10.1016/j.jcp.2016.09.013
14. Chen T, Shu CW. Entropy Stable High Order Discontinuous Galerkin Methods with Suitable Quadrature Rules for Hyperbolic Conservation Laws. *J Comput Phys* (2017) 345:427–61. doi:10.1016/j.jcp.2017.05.025
15. Crean J, Hicken JE, Fernández DCDR, Zingg DW, Carpenter MH. Entropy-stable Summation-By-Parts Discretization of the Euler Equations on General Curved Elements. *J Comput Phys* (2018) 356:410–38. doi:10.1016/j.jcp.2017.12.015
16. Hicken JE, Del Rey Fernández DC, Zingg DW. Multidimensional Summation-By-Parts Operators: General Theory and Application to Simplex Elements. *SIAM J Scientific Comput* (2016) 38:A1935–A1958. doi:10.1137/15m1038360
17. Parsani M, Carpenter MH, Fisher TC, Nielsen EJ. Entropy Stable Staggered Grid Discontinuous Spectral Collocation Methods of Any Order for the Compressible Navier–Stokes Equations. *SIAM J Scientific Comput* (2016) 38:A3129–A3162. doi:10.1137/15m1043510
18. Fernández DCDR, Crean J, Carpenter MH, Hicken JE. Staggered-grid Entropy-Stable Multidimensional Summation-By-Parts Discretizations on Curvilinear Coordinates. *J Comput Phys* (2019) 392:161–86. doi:10.1016/j.jcp.2019.04.029
19. Chan J. On Discretely Entropy Conservative and Entropy Stable Discontinuous Galerkin Methods. *J Comput Phys* (2018) 362:346–74. doi:10.1016/j.jcp.2018.02.033
20. Chan J, Wilcox LC. Discretely Entropy Stable Weight-Adjusted Discontinuous Galerkin Methods on Curvilinear Meshes. *J Comput Phys* (2019) 378:366–93. doi:10.1016/j.jcp.2018.11.010
21. Chan J. Skew-Symmetric Entropy Stable Modal Discontinuous Galerkin Formulations. *J Scientific Comput* (2019) 81:459–85. doi:10.1007/s10915-019-01026-w
22. Chan J, Del Rey Fernández DC, Carpenter MH. Efficient Entropy Stable Gauss Collocation Methods. *SIAM J Scientific Comput* (2019) 41:A2938–A2966. doi:10.1137/18m1209234
23. Chan J, Bencomo MJ, Del Rey Fernández DC. Mortar-based Entropy-Stable Discontinuous Galerkin Methods on Non-conforming Quadrilateral and Hexahedral Meshes. *J Scientific Comput* (2021) 89:1–33. doi:10.1007/s10915-021-01652-3
24. Chan J. Entropy Stable Reduced Order Modeling of Nonlinear Conservation Laws. *J Comput Phys* (2020) 423:109789. doi:10.1016/j.jcp.2020.109789
25. Pazner W, Persson PO. Analysis and Entropy Stability of the Line-Based Discontinuous Galerkin Method. *J Scientific Comput* (2019) 80:376–402. doi:10.1007/s10915-019-00942-1
26. Hicken JE. Entropy-stable, High-Order Summation-By-Parts Discretizations without Interface Penalties. *J Scientific Comput* (2020) 82:50. doi:10.1007/s10915-020-01154-8
27. Yan G, Kaur S, Banks JW, Hicken JE. *Entropy-stable Discontinuous Galerkin Difference Methods for Hyperbolic Conservation Laws* (2021). *arXiv preprint arXiv:2103.03826*.
28. Gkanis I, Makridakis C. A New Class of Entropy Stable Schemes for Hyperbolic Systems: Finite Element Methods. *Mathematics Comput* (2021) 90:1663–99. doi:10.1090/mcom/3617
29. Fernández DCDR, Carpenter MH, Dalcin L, Zampini S, Parsani M. Entropy Stable H/p-Nonconforming Discretization with the Summation-By-Parts Property for the Compressible Euler and Navier–Stokes Equations. *SN Partial Differential Equations Appl* (2020) 1:1–54. doi:10.1007/s42985-020-00009-z
30. Tadmor E. The Numerical Viscosity of Entropy Stable Schemes for Systems of Conservation Laws. I. *Mathematics Comput* (1987) 49:91–103. doi:10.2307/200825110.1090/s0025-5718-1987-0890255-3
31. Winters AR, Derigs D, Gassner GJ, Walch S. A Uniquely Defined Entropy Stable Matrix Dissipation Operator for High Mach Number Ideal MHD and Compressible Euler Simulations. *J Comput Phys* (2017) 332:274–89. doi:10.1016/j.jcp.2016.12.006
32. Chen T, Shu CW. Review of Entropy Stable Discontinuous Galerkin Methods for Systems of Conservation Laws on Unstructured Simplex Meshes. *CSIAM Trans Appl Mathematics* (2020) 1:1–52.
33. Wu X, Kubatko EJ, Chan J. High-order Entropy Stable Discontinuous Galerkin Methods for the Shallow Water Equations: Curved Triangular Meshes and GPU Acceleration. *Comput Mathematics Appl* (2021) 82:179–99. doi:10.1016/j.camwa.2020.11.006
34. Fjordholm US, Käppeli R, Mishra S, Tadmor E. Construction of Approximate Entropy Measure-Valued Solutions for Hyperbolic Systems of Conservation Laws. *Foundations Comput Mathematics* (2017) 17:763–827. doi:10.1007/s10208-015-9299-z
35. Schroeder PW, John V, Lederer PL, Lehrenfeld C, Lube G, Schöberl J. On Reference Solutions and the Sensitivity of the 2D Kelvin–Helmholtz Instability Problem. *Comput Mathematics Appl* (2019) 77:1010–28. doi:10.1016/j.camwa.2018.10.030
36. Davis S. Simplified Second-Order Godunov-type Methods. *SIAM J Scientific Stat Comput* (1988) 9:445–73. doi:10.1137/0909030
37. Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: A Fresh Approach to Numerical Computing. *SIAM Rev* (2017) 59:65–98. doi:10.1137/141000671
38. Schlottke-Lakemper M, Winters AR, Ranocha H, Gassner GJ. A Purely Hyperbolic Discontinuous Galerkin Approach for Self-Gravitating Gas Dynamics. *J Comput Phys* (2021) 442:110467. doi:10.1016/j.jcp.2021.110467
39. Ranocha H, Schlottke-Lakemper M, Winters AR, Faulhaber E, Chan J, Gassner GJ. Adaptive Numerical Simulations with Trixi. Jl: A Case Study of Julia for Scientific Computing. *Proc JuliaCon Conferences* (2022) 1:77.
40. Ranocha H, Dalcin L, Parsani M, Ketcheson DI. Optimized Runge-Kutta Methods with Automatic Step Size Control for Compressible Computational Fluid Dynamics. *Commun Appl Mathematics Comput* (2021) 2021. doi:10.1007/s42967-021-00159-w
41. Rackauckas C, Nie Q. DifferentialEquations.jl – A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia. *J Open Res Softw* (2017) 5:15. doi:10.5334/jors.151
42. [Dataset] Chan J, Ranocha H, Rueda-Ramírez AM, Gassner GJ, Warburton T. *Reproducibility Repository for on the Entropy Projection and the Robustness of High Order Entropy Stable Discontinuous Galerkin Schemes for Under-resolved Flows* (2022). Available at: https://github.com/trixi-framework/paper-2022-robustness-entropy-projection.
43. Ranocha H. Entropy Conserving and Kinetic Energy Preserving Numerical Methods for the Euler Equations Using Summation-By-Parts Operators. *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM* (2020) 2018:525–35. doi:10.1007/978-3-030-39647-3_42
44. Ranocha H, Gassner GJ. Preventing Pressure Oscillations Does Not Fix Local Linear Stability Issues of Entropy-Based Split-form High-Order Schemes. *Commun Appl Mathematics Comput* (2021) 1–24. doi:10.1007/s42967-021-00148-z
45. Chandrashekar P. Kinetic Energy Preserving and Entropy Stable Finite Volume Schemes for Compressible Euler and Navier-Stokes Equations. *Commun Comput Phys* (2013) 14:1252–86. doi:10.4208/cicp.170712.010313a
46. Rueda-Ramírez AM, Gassner GJ. *A Subcell Finite Volume Positivity-Preserving Limiter for DGSEM Discretizations of the Euler Equations* (2021). *arXiv preprint arXiv:2102.06017*. doi:10.23967/wccm-eccomas.2020.038
47. Richtmyer RD. Taylor Instability in Shock Acceleration of Compressible Fluids. *Tech rep., Los Alamos Scientific Lab N Mex* (1954) 13:297–319. doi:10.1002/cpa.3160130207
48. Youngs DL. Numerical Simulation of Turbulent Mixing by Rayleigh-Taylor Instability. *Physica D: Nonlinear Phenomena* (1984) 12:32–44. doi:10.1016/0167-2789(84)90512-8
49. Remacle JF, Flaherty JE, Shephard MS. An Adaptive Discontinuous Galerkin Technique with an Orthogonal Basis Applied to Compressible Flow Problems. *SIAM Rev* (2003) 45:53–72. doi:10.1137/s00361445023830

50. Hindenlang FJ, Gassner GJ, Kopriva DA. Stability of Wall Boundary Condition Procedures for Discontinuous Galerkin Spectral Element Approximations of the Compressible Euler Equations. In: *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2018*. Cham: Springer International Publishing (2020). p. 3–19. doi:10.1007/978-3-030-39647-3_1
51. Meshkov E. Instability of the Interface of Two Gases Accelerated by a Shock Wave. *Fluid Dyn* (1969) 4:101–4. doi:10.1007/BF01015969
52. Derigs D, Winters AR, Gassner GJ, Walch S, Bohm M. Ideal GLM-MHD: About the Entropy Consistent Nine-Wave Magnetic Field Divergence Diminishing Ideal Magnetohydrodynamics Equations. *J Comput Phys* (2018) 364:420–67. doi:10.1016/j.jcp.2018.03.002
53. Rueda-Ramírez AM, Hindenlang F, Chan J, Gassner G. *Entropy-Stable Gauss Collocation Methods for Ideal Magneto-Hydrodynamics* (2022). *arXiv preprint arXiv:2203.06062*.
54. Hindenlang F, Gassner G. *A New Entropy Conservative Two-point Flux for Ideal MHD Equations Derived from First Principles*. Madrid, Spain: Talk presented at HONOM (2019).
55. Mignone A, Tzeferacos P, Bodo G. High-order Conservative Finite Difference GLM–MHD Schemes for Cell-Centered MHD. *J Comput Phys* (2010) 229: 5896–920. doi:10.1016/j.jcp.2010.04.013
56. Hennemann S, Rueda-Ramírez AM, Hindenlang FJ, Gassner GJ. A Provably Entropy Stable Subcell Shock Capturing Approach for High Order Split Form DG for the Compressible Euler Equations. *J Comput Phys* (2021) 426:109935. doi:10.1016/j.jcp.2020.109935
57. Zhang X, Shu CW. On Positivity-Preserving High Order Discontinuous Galerkin Schemes for Compressible Euler Equations on Rectangular Meshes. *J Comput Phys* (2010) 229:8918–34. doi:10.1016/j.jcp.2010.08.016
58. Zhang X, Xia Y, Shu CW. Maximum-principle-satisfying and Positivity-Preserving High Order Discontinuous Galerkin Schemes for Conservation Laws on Triangular Meshes. *J Scientific Comput* (2012) 50:29–62. doi:10.1007/s10915-011-9472-8
59. Kraaijevanger JFBM. Contractivity of Runge-Kutta Methods. *BIT Numer Mathematics* (1991) 31:482–528. doi:10.1007/bf01933264
60. Conde S, Fekete I, Shadid JN. *Embedded Error Estimation and Adaptive Step-Size Control for Optimal Explicit strong Stability Preserving Runge–Kutta Methods* (2018). *arXiv preprint arXiv:1806.08693*.
61. Trefethen LN. Is Gauss Quadrature Better Than Clenshaw–Curtis? *SIAM Rev* (2008) 50:67–87. doi:10.1137/060659831
62. Kopriva DA. Stability of Overintegration Methods for Nodal Discontinuous Galerkin Spectral Element Methods. *J Scientific Comput* (2018) 76:426–42. doi:10.1007/s10915-017-0626-1
63. Orszag SA. On the Elimination of Aliasing in Finite-Difference Schemes by Filtering High-Wavenumber Components. *J Atmos Sci* (1971) 28:1074. doi:10.1175/1520-0469(1971)028<1074:oteoai>2.0.co;2
64. Hesthaven JS, Warburton T. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Berlin, Germany: Springer (2007).
65. Bardos C, Tadmor E. Stability and Spectral Convergence of Fourier Method for Nonlinear Problems: on the Shortcomings of the 2/3 De-aliasing Method. *Numerische Mathematik* (2015) 129:749–82. doi:10.1007/s00211-014-0652-y
66. Rueda-Ramírez AM, Pazner W, Gassner GJ. *Subcell Limiting Strategies for Discontinuous Galerkin Spectral Element Methods* (2022). *arXiv preprint arXiv: 2202.00576*.
67. Hindenlang FJ, Gassner GJ. On the Order Reduction of Entropy Stable DGSEM for the Compressible Euler Equations. In: *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2018*. Cham): Springer (2020). p. 21–44. doi:10.1007/978-3-030-39647-3_2
68. RanochaSummationByPartsOperators Hjl. A Julia Library of Provably Stable Discretization Techniques with Mimetic Properties. *J Open Source Softw* (2021) 6:3454. doi:10.21105/joss.03454
69. Terakado D, Hattori Y. Density Distribution in Two-Dimensional Weakly Compressible Turbulence. *Phys Fluids* (2014) 26:085105. doi:10.1063/1.4892460
70. San O, Kara K. Evaluation of Riemann Flux Solvers for WENO Reconstruction Schemes: Kelvin–Helmholtz Instability. *Comput Fluids* (2015) 117:24–41. doi:10.1016/j.compfluid.2015.04.026
71. San O, Maulik R. Stratified Kelvin–Helmholtz Turbulence of Compressible Shear Flows. *Nonlinear Process Geophys* (2018) 25:457–76. doi:10.5194/npg-25-457-2018
72. Gassner G, Kopriva DA. A Comparison of the Dispersion and Dissipation Errors of Gauss and Gauss–Lobatto Discontinuous Galerkin Spectral Element Methods. *SIAM J Scientific Comput* (2011) 33:2560–79. doi:10.1137/100807211
73. Chan J, Wang Z, Modave A, Remacle JF, Warburton T. GPU-accelerated Discontinuous Galerkin Methods on Hybrid Meshes. *J Comput Phys* (2016) 318:142–68. doi:10.1016/j.jcp.2016.04.003
74. Ranocha H, Schlottke-Lakemper M, Chan J, Rueda-Ramírez AM, Winters AR, Hindenlang F, et al. *Efficient Implementation of Modern Entropy Stable and Kinetic Energy Preserving Discontinuous Galerkin Methods for Conservation Laws* (2021). *arXiv preprint arXiv:2112.10517*.
75. Towns J, Cockerill T, Dahan M, Foster I, Gaither K, Grimshaw A, et al. XSEDE: Accelerating Scientific Discovery. *Comput Sci Eng* (2014) 16:62–74. doi:10.1109/mcse.2014.80

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers | Frontiers in Physics

ORIGINAL RESEARCH
published: 04 July 2022
doi: 10.3389/fphy.2022.910381

Extending the Capabilities of Data-Driven Reduced-Order Models to Make Predictions for Unseen Scenarios: Applied to Flow Around Buildings

Claire E. Heaney[1,2]*, *Xiangqi Liu*[1†], *Hanna Go*[1†], *Zef Wolffs*[1], *Pablo Salinas*[1], *Ionel M. Navon*[3] *and Christopher C. Pain*[1,2,4]

[1]Department of Earth Science and Engineering, Applied Modelling and Computation Group, Imperial College London, London, United Kingdom, [2]Centre for AI-Physics Modelling, Imperial-X, Imperial College London, London, United Kingdom, [3]Department of Scientific Computing, Florida State University, Tallahassee, FL, United States, [4]Data Assimilation Laboratory, Data Science Institute, Imperial College London, London, United Kingdom

OPEN ACCESS

Edited by:
Traian Iliescu,
Virginia Tech, United States

Reviewed by:
Kai Fukami,
University of California, United States
Ping-Hsuan Tsai,
University of Illinois at Urbana-Champaign, United States

***Correspondence:**
Claire E. Heaney
c.heaney@imperial.ac.uk

[†]*These authors have contributed equally to this work*

Specialty section:
This article was submitted to Statistical and Computational Physics, a section of the journal Frontiers in Physics

Received: *01 April 2022*
Accepted: *30 May 2022*
Published: *04 July 2022*

Citation:
Heaney CE, Liu X, Go H, Wolffs Z, Salinas P, Navon IM and Pain CC (2022) Extending the Capabilities of Data-Driven Reduced-Order Models to Make Predictions for Unseen Scenarios: Applied to Flow Around Buildings. Front. Phys. 10:910381. doi: 10.3389/fphy.2022.910381

We present a data-driven or non-intrusive reduced-order model (NIROM) which is capable of making predictions for a significantly larger domain than the one used to generate the snapshots or training data. This development relies on the combination of a novel way of sampling the training data (which frees the NIROM from its dependency on the original problem domain) and a domain decomposition approach (which partitions unseen geometries in a manner consistent with the sub-sampling approach). The method extends current capabilities of reduced-order models to generalise, i.e., to make predictions for unseen scenarios. The method is applied to a 2D test case which simulates the chaotic time-dependent flow of air past buildings at a moderate Reynolds number using a computational fluid dynamics (CFD) code. The procedure for 3D problems is similar, however, a 2D test case is considered sufficient here, as a proof-of-concept. The reduced-order model consists of a sampling technique to obtain the snapshots; a convolutional autoencoder for dimensionality reduction; an adversarial network for prediction; all set within a domain decomposition framework. The autoencoder is chosen for dimensionality reduction as it has been demonstrated in the literature that these networks can compress information more efficiently than traditional (linear) approaches based on singular value decomposition. In order to keep the predictions realistic, properties of adversarial networks are exploited. To demonstrate its ability to generalise, once trained, the method is applied to a larger domain which has a different arrangement of buildings. Statistical properties of the flows from the reduced-order model are compared with those from the CFD model in order to establish how realistic the predictions are.

Keywords: data-driven reduced-order modelling, non-intrusive reduced-order modelling, adversarial neural network, autoencoder, urban flows, machine learning, generalisation, inference

1 INTRODUCTION

Computational fluid dynamics codes can solve many complex problems thanks to advances in computing power and numerical methods. However, in order to obtain high-fidelity or high-resolution solutions, days or weeks of computational time may be required. Reduced-order modelling [1] is a popular technique, introduced to reduce the computational cost of producing high-resolution solutions albeit at the expense of generating these models in the first place, which can be substantial. Projection-based reduced-order models [2] have been widely used in computational science and consist of a dimensionality reduction stage (which identifies a suitable low-dimensional subspace) and a projection stage [in which the discretised high-fidelity model (HFM) is projected onto the low-dimensional subspace]. The reduced-order model (ROM) is then used to make predictions at a fraction of the cost of the HFM. Also known by the broader term of data-driven ROMs, non-intrusive reduced-order models (NIROMs) were then proposed, which replace the projection of the discretised HFM by interpolating between snapshots. Although classical interpolation methods can be used, machine learning (ML) techniques have become a popular choice for this task over the last 10 years. As well as being important for the learning the evolution of the solution, ML methods have also had an impact on dimensionality reduction, with many journal papers in the last 5 years reporting the use of autoencoders to identify the low-dimensional subspace for the ROM, see Heaney et al. [3]. One issue for neural networks is their ability to generalise, that is, to perform well for unseen data, and this is therefore also an issue for ML-based NIROMs [4]. For example, considering flow past buildings (the test case used here), if the shape, location or orientation of the buildings varies, and several configurations had been used to generate time-dependent snapshots, current methods used naïvely would struggle to interpolate successfully between different configurations of buildings in order to model unseen layouts. In this paper we supplement a NIROM method with a sub-sampling technique and a domain decomposition framework, both of which increase the ability of the ROM to generalise and solve problems based on unseen scenarios. We demonstrate that the method can make predictions for different configurations of buildings as well as for different-sized domains.

1.1 Related Work

The sub-sampling approach employed here was partially explored in Heaney et al. [3], in which, for dimensionality reduction, grids were randomly located within a pipe and solution fields were interpolated onto these grids, thereby generating data to train autoencoders. When generating data for the network to be used for prediction or inference, no randomly located subdomains were created and the solution fields were interpolated onto a small number of regularly-spaced subdomains. Being multiphase flow in a long, thin pipe, the solutions were dominated by advection in one direction, so a simpler approach could be used for that application. The method described here is general and can be applied to 2D and 3D flows, or indeed, 2D and 3D problems in computational physics in general.

For identifying a low-dimensional space in which to represent the snapshots, methods based on singular value decomposition (SVD) have been widely used. Proper Orthogonal Decomposition (POD) is one such SVD-based method and has been applied successfully to many fields such as reactor physics [5], urban flows [6] and fluid-structure interaction [7]. However, since 2018 there has been an explosion of interest in using autoencoders for dimensionality reduction, see references 26, 28–44, 48–52 in [3] and others in [8]. Due to the nonlinear activation functions, these networks find a nonlinear map between the high- and low-dimensional spaces, whereas with SVD-based methods, the mapping is linear. As a result, in some cases, autoencoders can find a more compact or a more accurate description of the reduced space. We choose a convolutional autoencoder as these networks have performed well in a number of studies for advection-dominated flows [9,10].

For learning the evolution of the snapshots in the low-dimensional space, classical methods were used initially

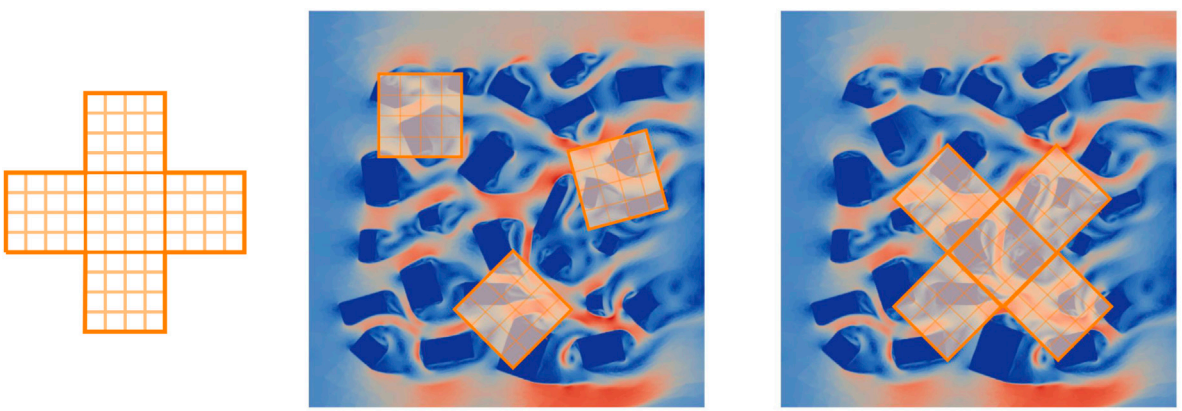


FIGURE 1 | Left: the star-shaped grid; centre: random placement of central subdomain of the grid (for obtaining data to train CAEs); right: random placement of the star-shaped grid (for obtaining data to train the predictive adversarial network).

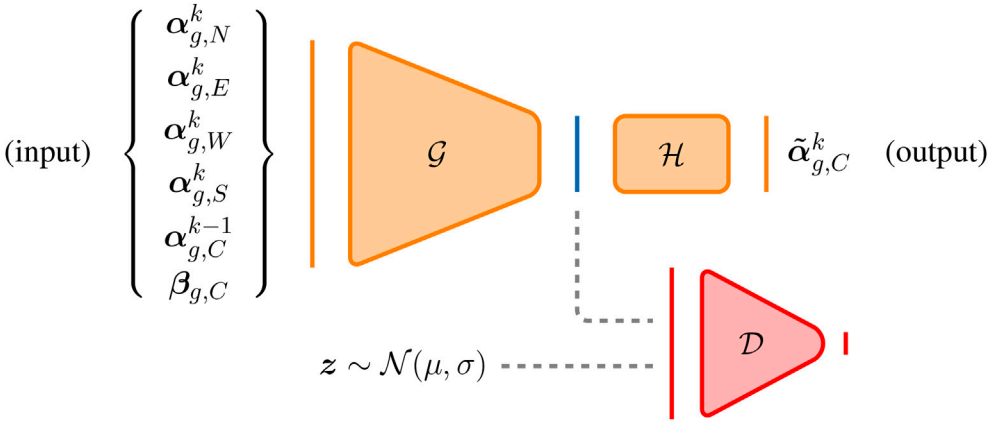


FIGURE 2 | The predictive adversarial network. The generator is represented by $\mathcal{G}$; the blue line represents the adversarial layer (and is the output of $\mathcal{G}$); the network $\mathcal{H}$ maps the values in the adversarial layer to the output; the input to the discriminator $\mathcal{D}$ is either a (genuine) sample from the prior distribution (here $\mathcal{N}(\mu, \sigma)$) or a (fake) sample from the output of the generator.

[11–13] which have been largely supplanted by neural networks, for example, Multi-layer Perceptron (MLP) [14], Long-Short Term Memory (LSTM) networks [15,16] and Gaussian Process Regression [17]. However, these networks can suffer from inaccuracies when predicting in time which can lead to the model diverging if the range of values of the reduced variables exceeds that seen during training [18–22]. To address this, we use an adversarial network. As the name suggests, adversarial networks use an adversarial training strategy which originates from generative adversarial networks (GANs) [23]. This type of neural network attempts to learn a distribution to which the training data belongs. Related networks are the adversarial autoencoder (AAE) [24] and Variational Autoencoders (VAEs). All three types of network set out to obtain better generalisation than other networks by attempting to obtain a smooth latent space with no gaps. Results in Makhzani et al. [24] show that the AAE performs better than the VAE on the MNIST digits. Imposing a prior distribution upon the variables of the latent space ensures that any set of latent variables, when passed through the decoder, should have a realistic output [24]. Currently, there exists only a small number of papers that use GANs, AAEs, VAEs, or combinations of these networks, for producing surrogate predictions of CFD modelling. Cheng et al. [25] combine a VAE and GAN to model the collapse of a water dam and Silva et al. [26] use a GAN to predict the spread of a virus within a small, idealised town originally modelled by an epidemiological model. Following Heaney et al. [3], we modify an adversarial autoencoder to make predictions in time. An alternative approach can be found in the work of Sanchez-Gonzalez et al. [27], who use graph-based networks and message passing to learn the system dynamics. Their networks can generalise well, being able to make predictions for different configurations (of ramps or barriers), although within the same domain.

Reduced-order modelling has long been combined with domain decomposition techniques. For example, for projection-based ROMs, Baiges et al. [28] restricts every POD basis function to one subdomain

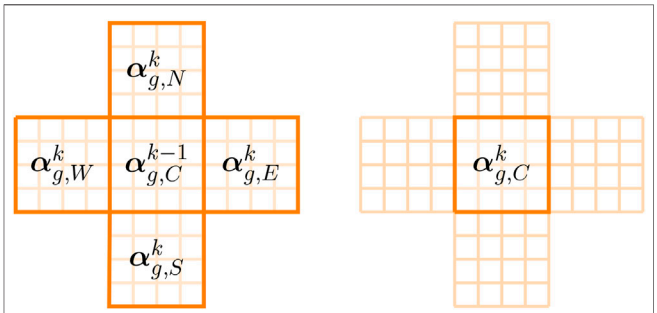


FIGURE 3 | Left: reduced variables associated with the five subdomains (*N*, *E*, *W*, *S*, and *C*) of the star-shaped grid at future (*k*) and current (*k* – 1) time levels, which, along with the reduced variables of the buildings ($\boldsymbol{\beta}_{g,C}$), form the input to the neural network; right: the reduced variables of the central grid at the future time level which form the output of the neural network.

of the partitioned domain. A similar method was used for non-intrusive ROMs [29], and was later adapted to partition the domain by reducing, as much as possible, the variation of the Reynolds stresses at the boundary between subdomains [30]. In this paper, the domain decomposition is associated with the prediction or online stage, when the domain of interest is decomposed into subdomains (that are the same size as those used in the sub-sampling procedure). The sub-sampling and domain decomposition approach we use bears some resemblance to the method reported in Yang and Grooms [31], which decomposes a domain into patches in order to facilitate the training of a neural network. However, our motivation for using domain decomposition is to make predictions for unseen scenarios and for domains that are significantly larger than (or in some way different from) those used in the training process.

Other approaches have been taken to build ROMs, such as dynamic mode decomposition (DMD) [32] and sparse identification of nonlinear dynamics (SINDy) [33]. DMD identifies both spatial and temporal modes and is often used as a diagnosis tool [34], however, examples do exist of DMD having been used to make predictions [35–37]. As with other SVD-based

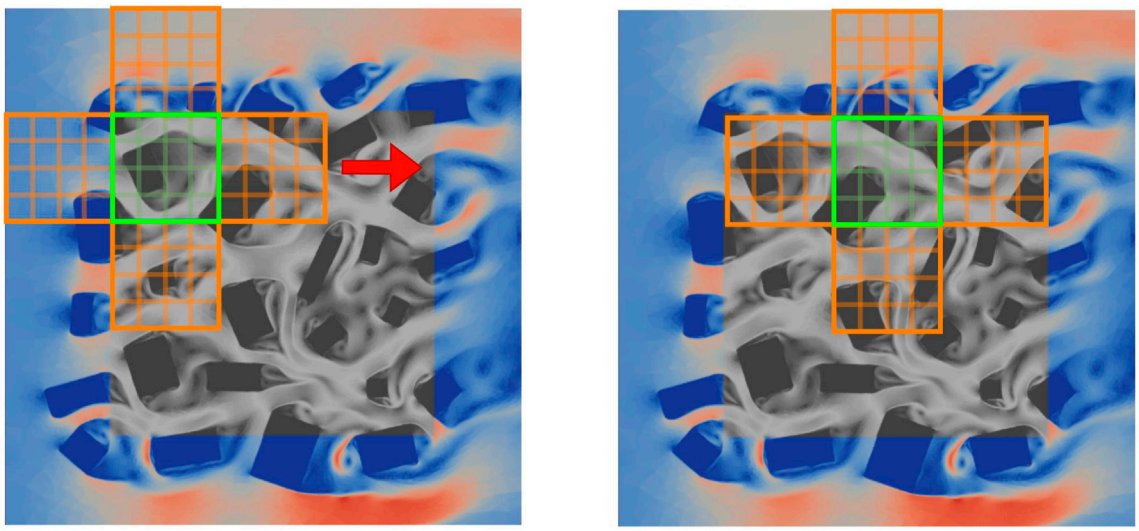


FIGURE 4 | Greyscale regions indicate the area where we seek a solution. The coloured regions show the magnitude of the velocity of the imposed boundary conditions. Left: Using the latest solutions of the four neighbouring subdomains (indicated in orange), and the previous solution in the central subdomain (shown in green) and the buildings fields in the central subdomain, a prediction is made for the central subdomain. Right: move to the next set of five subdomains and predict for the next central subdomain.

methods, DMD can struggle to capture symmetries and invariants in the flow fields [4], which is one reason why we opt for a combination of autoencoder (for dimensionality reduction) and adversarial network (for prediction). SINDy aims to find a sparse representation of a dynamical system relying on the assumption that, for many physical systems, only a small number of terms dominate the dynamical behaviour and has been applied to a number of fluid dynamics problems, including flow past a cylinder [8,38]. It can be difficult to compress accurately to a small number of variables, and SINDy was not used here, because we did not want to be restricted to using a small number of reduced variables.

Although much investigation has been carried out into parametrised NIROMs, challenges do remain. Hasegawa et al. [39] create a ROM consisting of an autoencoder and a LSTM which can predict the flow past a bluff body. The profile of the bluff body is controlled by 8 coefficients and a truncated trigonometric basis. Hesthaven and Ubbiali [14] address both geometrical and physical parametrisations with a NIROM based on POD and MLP. They solve a lid-driven cavity problem for a parallelogram-shaped domain which is defined by three parameters: two edge lengths (both in the range [1,2]) and one angle [in the range $(30^{\circ}, 150^{\circ})$]. In both cases, the ROMs were able to predict well for unseen scenarios. However, we wish to extend significantly the range of unseen scenarios for which ROM is capable of making predictions. For instance, our test case of flow past buildings consists of about 150 differently-sized buildings. To apply methods similar to those developed by Hasegawa et al. [39] or Hesthaven and Ubbiali [14] would be impractical in this case, due to the number of buildings. Furthermore, not only do we wish to be able to solve for unseen building configurations involving many buildings, we also want to be able to solve for larger domains than used when generating the training data.

In this paper we combine a sub-sampling technique with a domain decomposition method in order to make predictions for unseen scenrios. The sub-sampling technique essentially frees the ROM from its dependency on the domain of the original problem and enables it to make predictions for arbitrary domains. This method focuses on capturing high-resolution detail around many different buildings, rather than capturing the flow around one particular configuration of buildings. The domain decomposition method is used to partition an unseen domain into subdomains that relate to the snapshots obtained in the sub-sampling process. A convolutional autoencoder is used to compress the data and a predictive adversarial network is trained to predict the reduced variables representing the air flow around a group of buildings in a subdomain. In the unseen domain, the subdomains are regularly arranged, and predictions for the solutions in each subdoamin are generated. An iteration-by-subdomain approach [40] is used to achieve convergence of the global solution. The contribution of this work is twofold: a method is proposed that constructs a ROM using one configuration (of buildings) which is able to predict for an unseen configuration; and the unseen configuration can be associated with a larger domain than that of the original configuration. This article presents results for flow past buildings, and makes predictions on a domain that has over twice the area of the original configuration and a different arrangement of buildings.

In the remainder of this article, **Section 2** outlines the methodology, **Section 3** presents the results, and **Section 4** draws conclusions and outlines future work.

2 METHODOLOGY

The generation of NIROMs or data-driven reduced-order models typically consists of three stages: (1) solving the

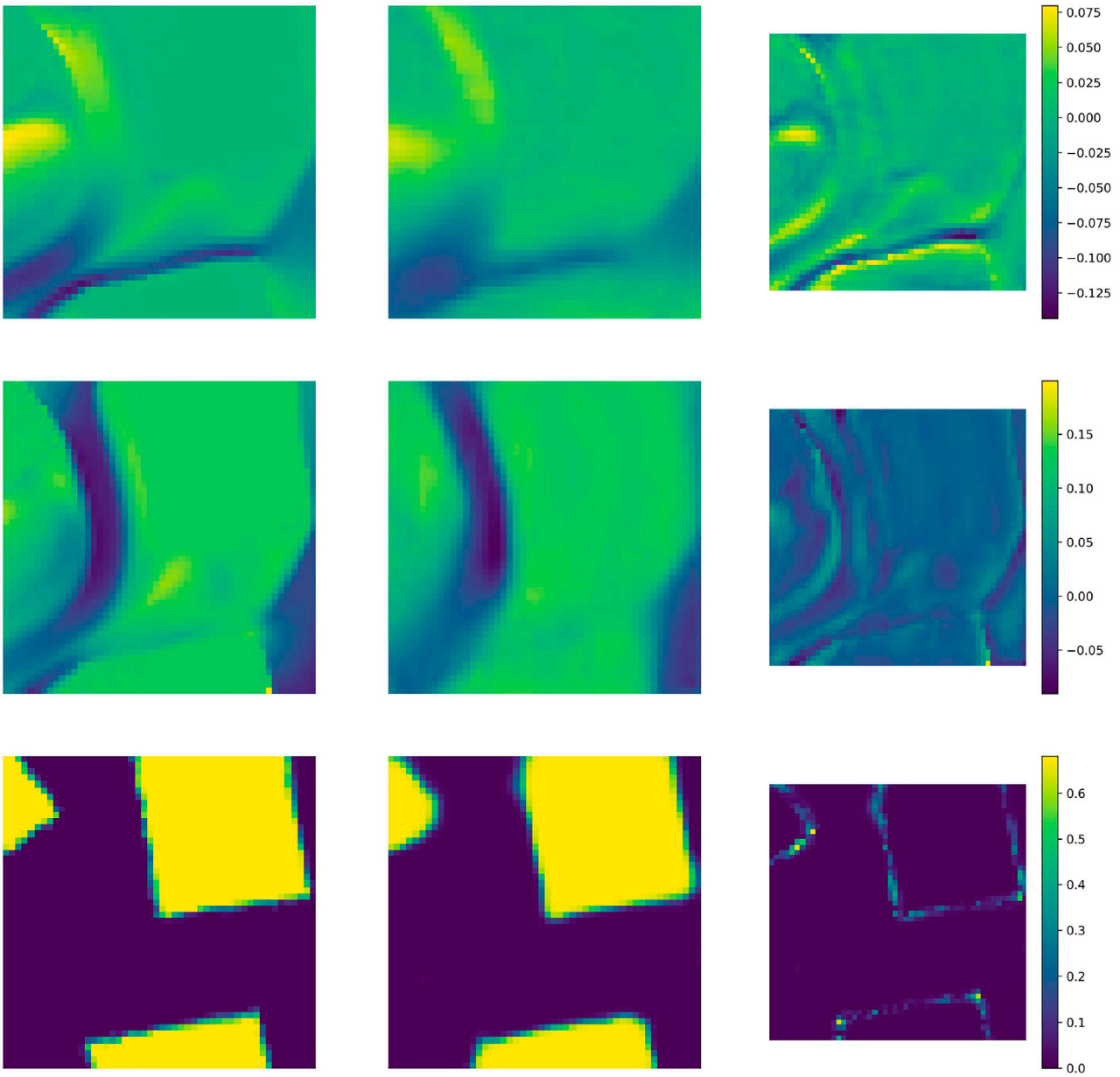


FIGURE 5 | Plots to demonstrate the quality of the autoencoders. The (unseen) solutions associated with one subdomain are compressed and reconstructed. Left column: results from the HFM; central column: reconstruction from the CAE; right column: pointwise difference. Top row: x – component of velocity; middle row: y – component of velocity; bottom row: buildings field.

HFM to produce the snapshots; (2) applying dimensionality reduction to the snapshots to obtain a low-dimensional space (**Subsection 2.2**); and (3) learning how the system evolves in low-dimensional space (**Subsection 2.3**). The method outlined in this paper has these stages, however, also makes use of a sub-sampling technique (**Subsection 2.1**) and domain decomposition in order to enable the reduced-order model to make predictions for unseen scenarios (**Subsection 2.4**). This frees the NIROM from its dependency on the original problem domain and paves the way for the model to make predictions for unseen scenarios including different building geometries and locations, and different sizes of domain. We use a convolutional autoencoder (CAE) for dimensionality reduction and a predictive adversarial network for prediction or inference as it is known in machine learning terminology.

Throughout this section we refer to the test case used here, air flow around buildings modelled in two dimensions (2D) using adapted, unstructured meshes [41]. There are two solution fields of importance for the reduced-order model: the velocity field and

TABLE 1 | Hyperparameter values used in the neural networks. Associated with the optimiser, β_1 and β_2 are the exponential decay rate for the first moment estimates and the exponential decay rate for the exponentially weighted infinity norm respectively.

	Velocity CAE	Buildings CAE	PAN
number of epochs	9,000	5,000	5,000
Optimiser	Adam	Adam	Adam
learning rate	5×10^{-4}	5×10^{-4}	5×10^{-4}
β_1	0.9	0.9	0.98
β_2	0.999	0.999	0.999
activation functions			
main network	elu	elu	elu
final layer	sigmoid	sigmoid	sigmoid
Discriminator	n/a	n/a	relu
batch size	32	32	128
latent variables	50	30	n/a

TABLE 2 | Left: architecture of the CAE for the velocity field. (The architecture of the CAE for the buildings field is similar, but with an input of (50,50,1) and a central dense layer of 30 neurons.) Right: architecture of the PAN.

CAE		PAN	
Input	(50,50,2)	input	280
Conv	(50,50,32)	Dense	256
MaxPool	(25,25,32)	Dropout	256
Conv	(25,25,16)	BatchNorm	256
MaxPool	(13,13,16)	Reshape	(2,2,64)
Conv	(13,13,8)	Conv (Adversarial)	(2,2,16)
MaxPool	(7,7,8)	UpSample	(4,4,16)
flatten	392	Conv	(4,4,32)
Dense	50	UpSample	(8,8,32)
Dense	392	Conv	(8,8,64)
reshape	(7,7,8)	UpSample	(16,16,64)
Conv	(7,7,8)	BatchNorm	(16,16,64)
UpSample	(14,14,8)	Conv	(16,16,128)
Crop	(13,13,8)	UpSample	(32,32,128)
Conv	(13,13,16)	BatchNorm	(32,32,128)
Upsample	(26,26,16)	Flatten	131072
Crop	(25,25,16)	Dense	50
Conv	(25,25,32)	Discriminator	
UpSample	(50,50,32)	Dense	64
Conv	(50,50,2)	Dense	100
		Dense	500
		Dense	1

a field which indicates where there is a building, referred to as the 'buildings field'.

2.1 Sub-sampling to Obtain Snapshots

To obtain the snapshots used for training the neural networks, a star-shaped structured grid (see **Figure 1A**), is randomly located and orientated within the domain (see **Figure 1C**), although some care is taken so that the grid is not too near the boundary of the domain. Each grid consists of five subdomains, four of which are neighbours of a central subdomain. For the CAE, the velocity and buildings fields are interpolated onto the grid at one randomly selected time level, although only data from the central subdomain is used (see **Figure 1B**). This is repeated for a total number of N^g grids, resulting in N^g snapshots. Both sets of snapshots are separated into training, validation and test datasets. For the predictive network that we refer to as PAN (Predictive Adversarial Network), the velocity field is interpolated onto the grid at two successive randomly selected time levels. The buildings field is also interpolated onto the grid at one of these time levels (the buildings field is constant through time). This is repeated for a total number of N^g grids. So, instead of using the entire solution fields as snapshots, only the part of the solution field that has been interpolated onto the grid is used as a snapshot. To capture the behaviour of the flows, many grids are used to generate many snapshots from which the neural networks can learn about the flow

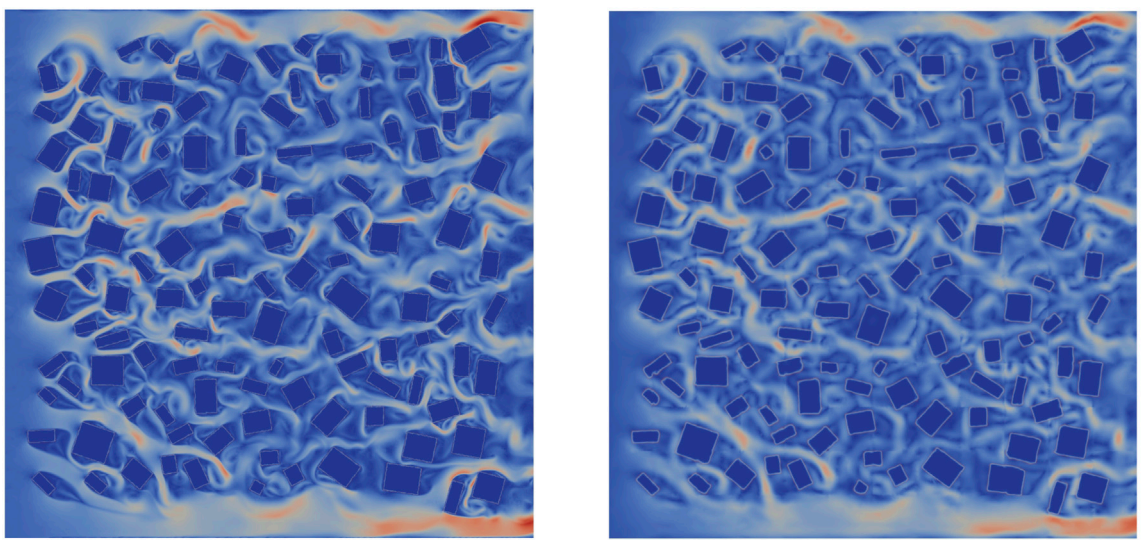


FIGURE 6 | The velocity magnitude at the 350th time level from the HFM (left) and the predictive adversarial network (right).

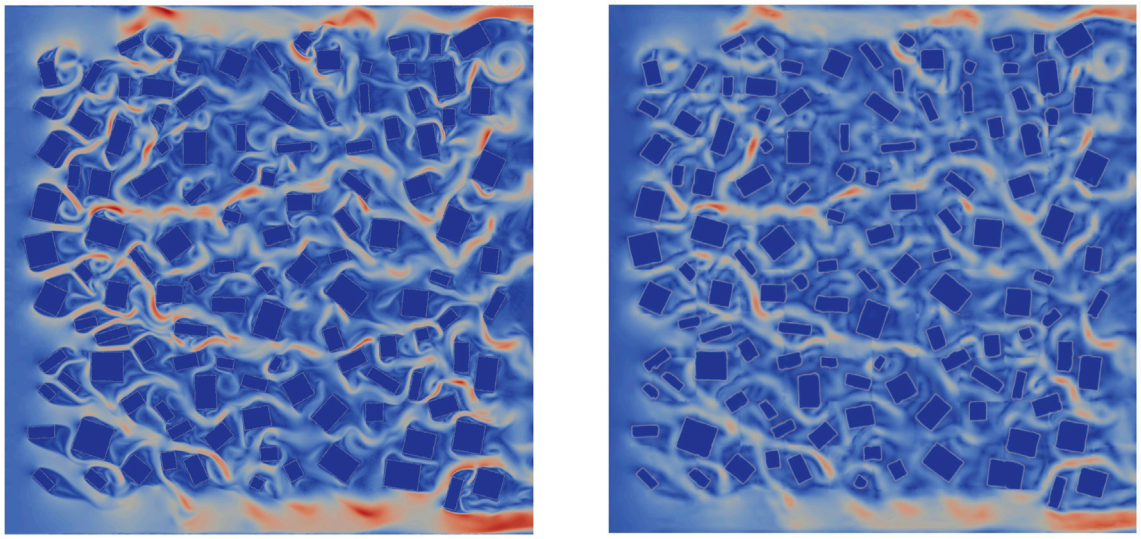


FIGURE 7 | The velocity magnitude at the 400th time level from the HFM (left) and the predictive adversarial network (right).

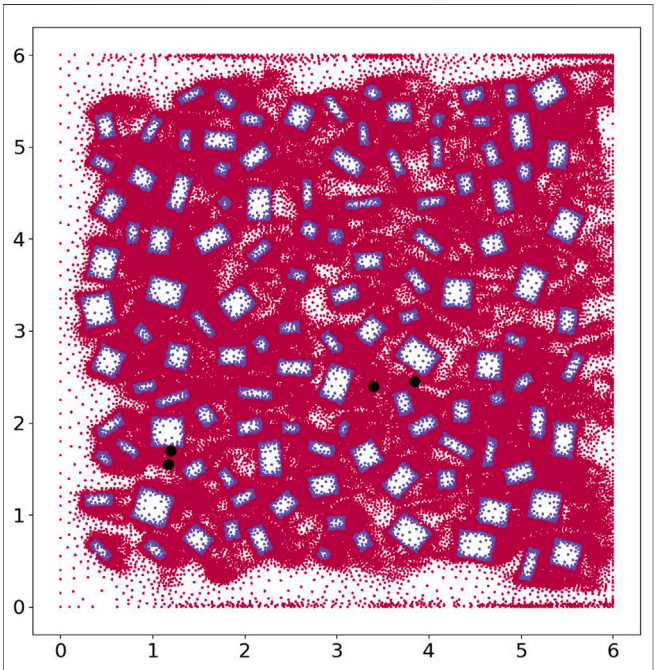


FIGURE 8 | Here we plot all the nodes within the 6 by 6 domain. The nodes within the building appear as blue and those outside of the building as red. We also show four points within the domain, in black, where we plot the histograms, or probability density functions, of the x – and y – components of velocity.

characteristics without being tied to a particular arrangement of buildings. This use of sub-sampling allows the neural networks to learn about fine-scale features such as eddies in the vicinity of buildings independently, to some extent, of the entire domain.

2.2 Dimensionality Reduction

Having been shown to compress data well for advection-dominated flows, we choose the convolutional autoencoder to reduce the dimension of the problem and find the low-dimensional subspace in which the HFM will be approximated. The CAE has been widely applied to reduced-order models in recent years and more details about this type of network along with schematic diagrams can be found in Gonzalez and Balajewicz [42], Xu and Duraisamy [43], Wu et al. [44], Nikolopoulos et al. [45]. In a nutshell, the CAE is a type of feed-forward neural network with convolutional layers that attempts to learn the identity map [46]. When used for compression, the CAE has a central 'bottleneck' layer which has fewer neurons than the input and output layers. The values of the neurons in this central layer are known as latent variables or reduced variables. The outer layers of the network consist of convolutional layers (which detect patterns or features in the flow fields) and pooling layers (which reduce the dimensions of the data), and at the centre of the network are fully connected layers. The autoencoder can be split into an encoder, which maps the input to the latent variables (compressing the data) and a decoder which maps the latent variables to the output (reconstructing the data). If f_u^{enc} and f_u^{dec} represent the encoder and decoder of the velocity field respectively, then the output of the autoencoder can be written as follows

$$\boldsymbol{u}_{g,C}^{\text{recon}} = f_u^{\text{dec}}\left(f_u^{\text{enc}}\left(\boldsymbol{u}_{g,C}\right)\right), \tag{1}$$

where $\boldsymbol{u}_{g,C}$ represents the velocity field that has been interpolated onto the central subdomain of structured grid g and is the input to the autoencoder, and $\boldsymbol{u}_{g,C}^{\text{recon}}$ represents the reconstruction and is the output of the autoencoder. Once trained, the reduced variables associated with a particular subdomain can be written as

$$\boldsymbol{\alpha}_{g,s}^{k} = f_u^{\text{enc}}\left(\boldsymbol{u}_{g,s}^{k}\right) \quad \text{where } s \in \{N, E, W, S, C\}. \tag{2}$$

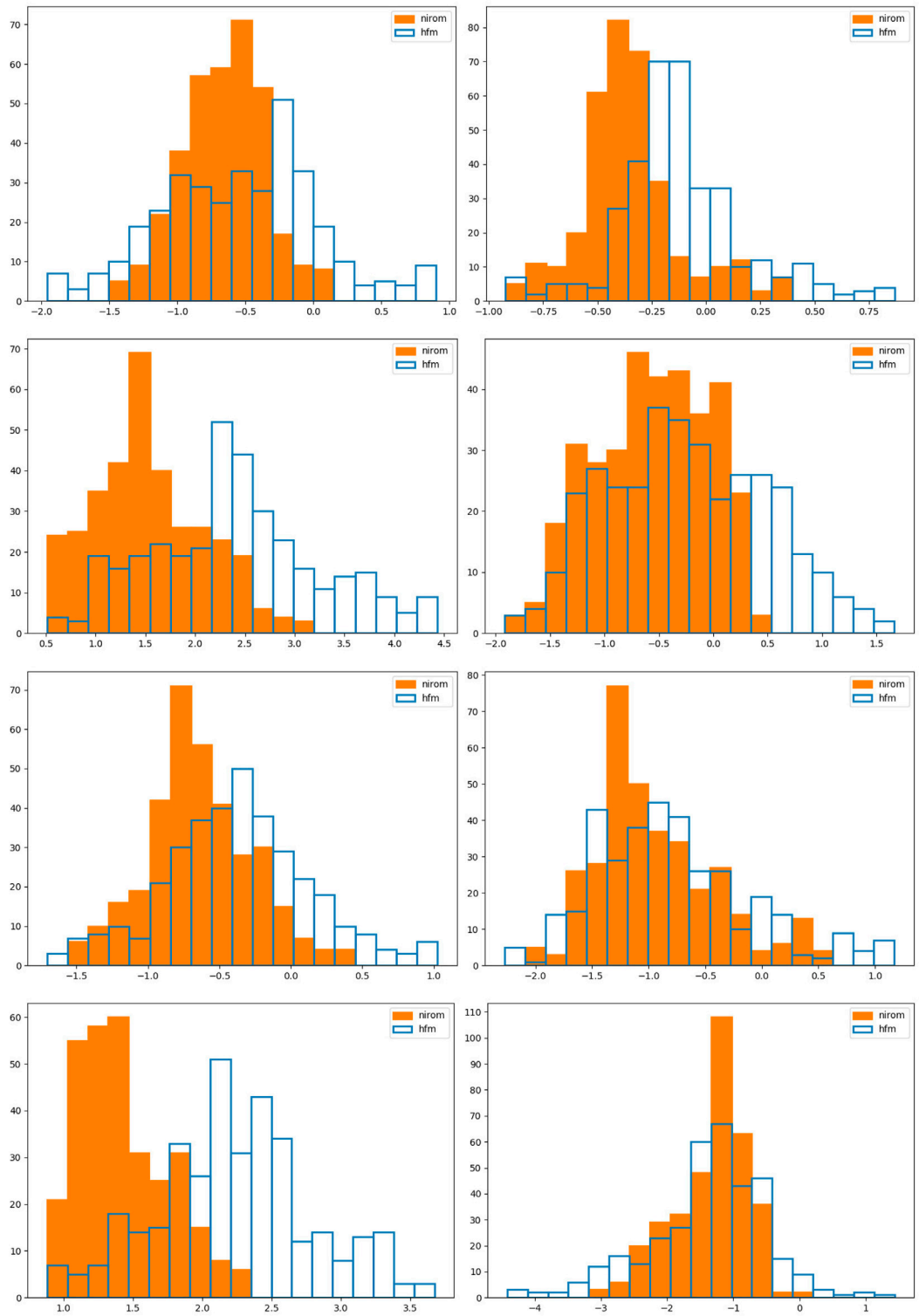


FIGURE 9 | Here we show histograms for the 6 by 6 test case at points 1, 2, 3 and 4 (see **Table 3**) in rows from top to bottom respectively. The first column of graphs show the x – component of velocity and the second the y – component. We compare in these graphs the histograms (or probability density functions) of the velocity components of the HFM in blue, and NIROM in orange.

A second CAE was trained to compress the buildings field, using data from N^g snapshots. The reduced variables associated with the buildings field can be written as

$$\boldsymbol{\beta}_{g,s} = f_b^{\text{enc}}\left(\boldsymbol{b}_{g,s}\right) \quad \text{where } s \in \{N, E, W, S, C\}. \tag{3}$$

where $\boldsymbol{b}_{g,C}$ is the buildings field of subdomain s of grid g. **Figure 1B** shows three such subdomains, randomly located and orientated within the domain, and superposed on the velocity field of the test case at a particular time. Although we could have used data from all the subdomains to train the autoencoders, we found this not to be necessary and only used data from the central subdomains. We do, however, obtain the reduced variables for any subdomain (*N*, *E*, *W*, *S* or *C*) with the trained encoder.

2.3 Prediction in Time

In this work, we follow [3] by modifying the adversarial autoencoder so that it can predict in time, and refer to it as a "Predictive Adversarial Network" (PAN). The adversarial autoencoder [24] uses an adversarial strategy to force the autoencoder's latent space to follow a prior distribution (P_{prior}) whilst the output aims to replicate the input as closely as possible. Thus, in addition to the encoder-decoder networks of a standard autoencoder, the adversarial autoencoder has a discriminator (the adversarial network) which is connected to the central layer of the encoder-decoder. The discriminator is trained to distinguish between samples from the prior distribution (true samples) and samples from latent space (fake samples). The modifications made to the adversarial autoencoder include: the inputs and outputs no longer have the same dimension (as is required for autoencoders that learn the identity map); the width of the layers does not fall below the width of the output layer (preventing additional compression to that already performed in the dimensionality reduction stage); and the loss function no longer minimises the difference between its input and output, rather the output and the desired output. A schematic diagram of the PAN can be seen in **Figure 2**, where $\mathcal{G}$ represents the generator, $\mathcal{H}$ maps from the adversarial layer (in blue) to the output of the network, and connected to the adversarial layer is the discriminator $\mathcal{D}$. The prior distribution chosen here is the normal distribution (or Gaussian distribution) with a mean of zero and a variance of one. This choice of distribution for the latent variables ($\boldsymbol{z}$) does not affect the distribution that the output of the network can have. The loss function for the predictive adversarial network is given by

$$\min_{\mathcal{G},\mathcal{H}} \mathbb{E}\left(\|\boldsymbol{\alpha}_{g,C}^k - \tilde{\boldsymbol{\alpha}}_{g,C}^k\|^2\right) + \min_{\mathcal{G}} \max_{\mathcal{D}} \left(\mathbb{E}_{\boldsymbol{z}\sim P_{\text{prior}}}\left[\log \mathcal{D}(\boldsymbol{z})\right] + \mathbb{E}_{\boldsymbol{\alpha}\sim P_{\text{data}}}\left[\log\left(1 - \mathcal{D}(\mathcal{G}(\boldsymbol{\alpha}))\right)\right]\right), \tag{4}$$

where $\tilde{\boldsymbol{\alpha}}_{g,C}^k$ are the reduced variables predicted by the network $(\mathcal{H} \circ \mathcal{G})$ for the central subdomain of grid g at time level k, $\boldsymbol{z} \sim P_{\text{prior}}$ is a sample from the desired distribution and $\boldsymbol{\alpha} \sim P_{\text{data}}$ is a sample of the reduced variables that have passed through the generator. The first term represents the error in the prediction of the reduced variables, and the second and third terms are the regularisation terms arising from the adversarial training which attempt to bring the posterior distribution of a hidden layer close to the prior distribution. During training, there are therefore three separate steps per mini-batch. First, the weights of $\mathcal{G}$ and $\mathcal{H}$ are updated as a result of minimising the error in the output of network; second, the weights of the discriminator network are updated so it can better tell apart the genuine samples from the generated samples; finally, weights of the generator are updated so it can better deceive the discriminator network.

For the prediction network, data from all five subdomains is used for training and inference, as shown in **Figure 3**. The star-shaped grid is used, as, when predicting in time, it is beneficial to have information from neighbouring regions. The input to the network consists of the reduced variables associated with the four neighbouring subdomains at the future time level (t^k) (see **Figure 3**(left)); the reduced variables associated with the central grid at the current time (t^{k-1}) (see **Figure 3**(left)); and the reduced variables associated with the central subdomain that describe the buildings. The output of the network is the reduced variables associated with the central subdomain at the future time level (see **Figure 3**(right)). If the predictive adversarial network is represented by f, this can be written as

$$\boldsymbol{\alpha}_{g,C}^k = f\left(\boldsymbol{\alpha}_{g,N}^k, \boldsymbol{\alpha}_{g,E}^k, \boldsymbol{\alpha}_{g,W}^k, \boldsymbol{\alpha}_{g,S}^k, \boldsymbol{\alpha}_{g,C}^{k-1}, \boldsymbol{\beta}_{g,C}\right). \tag{5}$$

2.4 Prediction for Unseen Scenarios

In order to increase the generalisation properties of the reduced-order model, in addition to using a sub-sampling technique to obtain the snapshots (as described in **Section 2.1**, we pose each new scenario within a domain decomposition framework.

2.4.1 Combining Subdomains to Model an Unseen Scenario

Having trained neural networks to be able to predict flows within a subdomain given the flows in neighbouring subdomains and the layout of the buildings, a new (and therefore unseen) domain can be constructed from a non-overlapping union of these subdomains. Initial conditions for both the velocity and buildings fields are required, which are

TABLE 3 | The coordinates of the points at which the probability distributions are generated for both the 6 by 6 case and the 9 by 9 case.

	Point	*x*-coordinate	*y*-coordinate
6 by 6 test case	1	1.2	1.7
	2	1.175	1.55
	3	3.4	2.4
	4	3.85	2.45
9 by 9 test case	1	2.45	5.45
	2	2.6	2.6
	3	5.4	5.4
	4	6.15	3.15

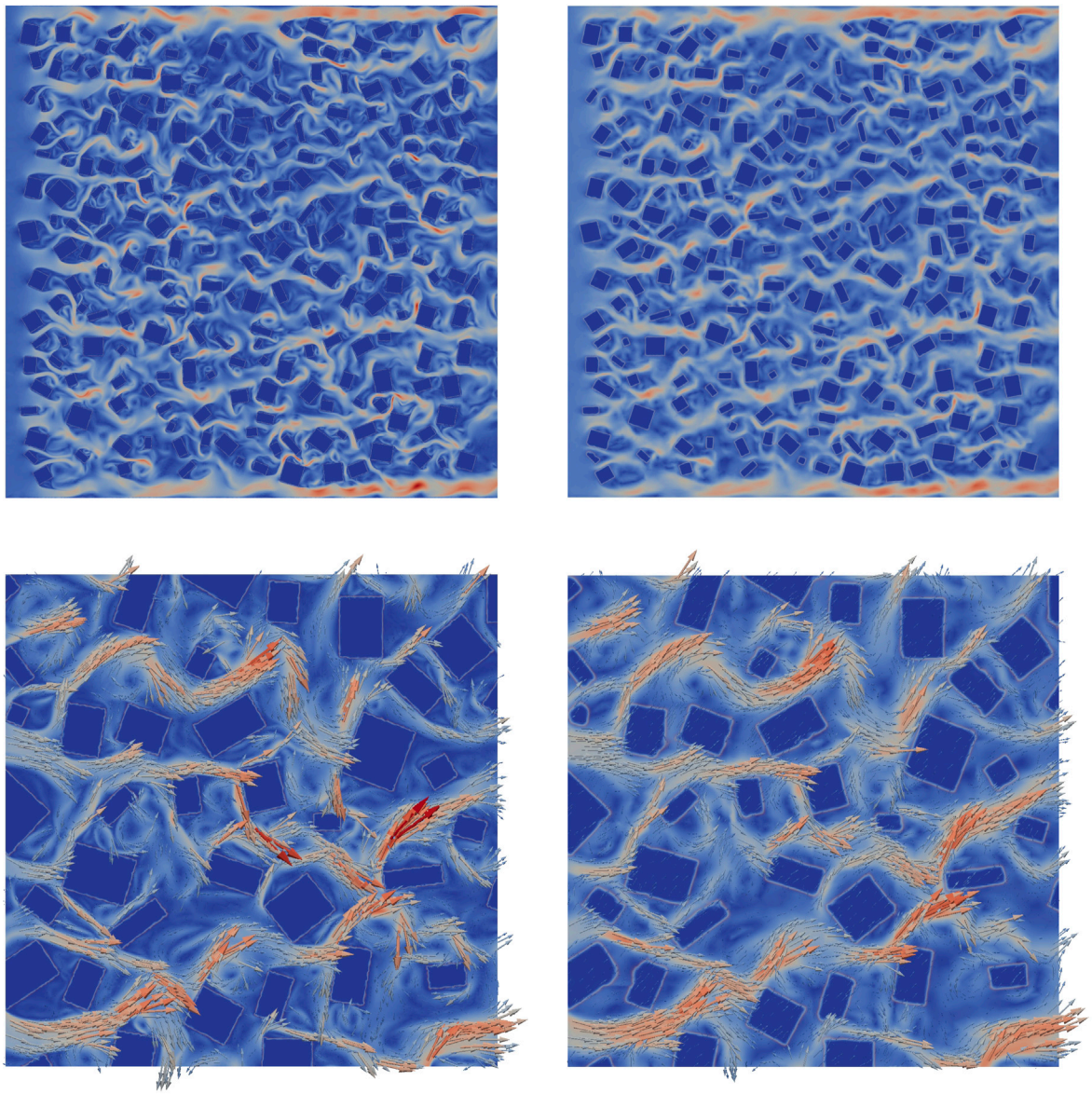


FIGURE 10 | Velocity magnitude across the 9 by 9 domain at time level 250. Top left: HFM, and top right: prediction by the ROM. Bottom left and right: velocity vectors over a 3 by 3 region [1,4]×[4,7] of the 9 by 9 domain, for HFM and NIROM respectively.

then encoded by the convolutional autoencoders. This provides a starting point from which to evolve the reduced variables in time. At a given time level, a prediction is made for the subdomains one-by-one (see **Figure 4**), with the variables being updated as and when the solutions are available in the manner of Gauss-Seidel iteration. An iteration-by-subdomain approach is used until convergence of the global solution is reached at that time level, and the process continues to find the solution at the next time level. In this manner, the solution of the NIROM is marched forward in time from an initial condition. **Figure 4** is a schematic diagram that shows how domain decomposition can be used to form an array of subdomains, and how the PAN is used with iteration-by-subdomain to solve for the global solution. This approach for prediction of flows for an unseen arrangement of buildings is summarised in Algorithm 1.

Algorithm 1. An algorithm for finding the solution for the reduced variables in a subdomain and sweeping over all the subdomains to obtain a converged solution over the whole domain. The algorithm marches forward in time from the initial condition to time level N^{time}.

!! set initial conditions for each subdomain i
$\boldsymbol{\alpha}_i^0 \; \forall i$
!! define two sets containing the internal subdomains
define $\mathcal{F}$, containing the internal subdomains ordered for a forward sweep
define $\mathcal{B}$, containing the internal subdomains ordered for a backwards sweep
for time level $k = 1, 2, \ldots, N^{\text{time}}$ **do**
  !! set boundary conditions
  $\boldsymbol{\alpha}_b^k \; \forall$ subdomains b on the boundary
  !! estimate the solution at the future time level k for all internal subdomains $\mathcal{I}$
  for subdomain $i \in \mathcal{I}$ **do**
    $\boldsymbol{\alpha}_i^k = \boldsymbol{\alpha}_i^{k-1}$
  end for
  !! sweep over subdomains
  for sweep iteration $j = 1, 2, \ldots, N^{\text{sweep}}$ **do**
    for subdomain $i \in \mathcal{F}$ **do**
      !! calculate the latent variables of subdomain i at time level k
      $\boldsymbol{\alpha}_{i,C}^k = f(\boldsymbol{\alpha}_{i,N}^k, \boldsymbol{\alpha}_{i,E}^k, \boldsymbol{\alpha}_{i,W}^k, \boldsymbol{\alpha}_{i,S}^k, \boldsymbol{\alpha}_{i,C}^{k-1}, \boldsymbol{\beta}_{i,C})$.
    end for
    for subdomain $i \in \mathcal{B}$ **do**
      !! calculate the latent variables of subdomain i at time level k
      $\boldsymbol{\alpha}_{i,C}^k = f(\boldsymbol{\alpha}_{i,N}^k, \boldsymbol{\alpha}_{i,E}^k, \boldsymbol{\alpha}_{i,W}^k, \boldsymbol{\alpha}_{i,S}^k, \boldsymbol{\alpha}_{i,C}^{k-1}, \boldsymbol{\beta}_{i,C})$.
    end for
  end for
end for

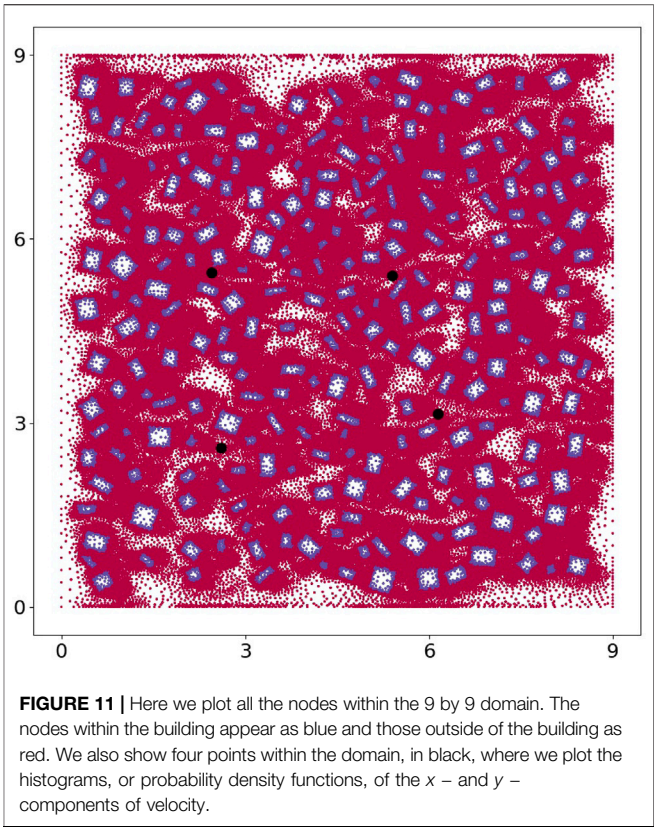


FIGURE 11 | Here we plot all the nodes within the 9 by 9 domain. The nodes within the building appear as blue and those outside of the building as red. We also show four points within the domain, in black, where we plot the histograms, or probability density functions, of the x – and y – components of velocity.

3 RESULTS

The methods described previously are now tested on flow past buildings modelled in 2D. Assuming an incompressible viscous fluid, the conservation of mass and the Navier-Stokes equations can be written as

$$\nabla \cdot \boldsymbol{v} = 0, \tag{6}$$

$$\rho\left(\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{v}\right) = -\nabla p + \nabla \cdot \left(\mu\left(\nabla \boldsymbol{v} + \nabla^T \boldsymbol{v}\right)\right) - \sigma \boldsymbol{v}, \tag{7}$$

where t represents time, $\boldsymbol{v}$ represents velocity, p represents pressure, μ is the dynamic viscosity, ρ is the density and σ is an absorption term that is zero outside the buildings and 10^6 inside the buildings. The boundary conditions that we use in conjunction with **Equations 6** and **7** are defined as follows. At the inlet we specify the normal velocity component which we set to unity. The tangential components at the inlet boundary on the left of the domains (see, for example, **Figure 6**) are set to zero. We also set a zero normal velocity boundary condition to the top and bottom boundaries of this domain along with a zero shear stress condition. At the outlet we set the normal and shear stress components to zero which effectively sets the pressure to near zero at the outlet. **Equations 6** and **7** together with the boundary conditions are discretised using a finite element representation for velocity and a control volume representation of pressure [47] combined in a P1DG-P1CV element [48,49]. An unstructured mesh is used which adapts through time, and an adaptive time step is also used. For more details of how the governing equations are discretised and solved, see Obeysekara et al. [48].

For ease of setting up this test case, we represent the areas occupied by buildings as a sink in the velocity field (through an absorption coefficient which acts on the velocity field, which can be seen in the term involving $\sigma \boldsymbol{v}$ in **Eq. 7**). By using adaptive meshes (adapting to σ and the velocity field), we obtain a sharp boundary between the buildings and the outside air flow, although this would be sharper if the building had been modelled explicitly. In any case, we believe that the CFD results are a good enough representation of flow past buildings to be used in this proof-of-concept paper.

The numerical solutions were found for two domains, one measuring 6 by 6 and the other measuring 9 by 9. These domains were populated with randomly located and orientated buildings. The lengths of both edges of each building were chosen randomly from the interval [0.1, 0.4] and a minimum gap of 0.075 was enforced between the buildings. A gap between the domain boundaries and the buildings was maintained. In practice the number of buildings for the 6 by 6 and 9 by 9 case is about 150 and 340 respectively.

A Reynolds number of 300 was used in both the simulations, and was based on the unity inlet velocity and minimum building edge length. The actual time step size was controlled by the Courant number, chosen to be 0.5, and the solutions were saved every 0.008 time units, giving the NIROM a time step size of 0.008. For a regular array of 17 square cylinders, Shams-ul Islam et al. [50] observed chaotic flows for Reynolds numbers greater than 125. In our case, we believe that Re = 300 is more than sufficient for the flow to be chaotic and therefore to present an interesting modelling challenge.

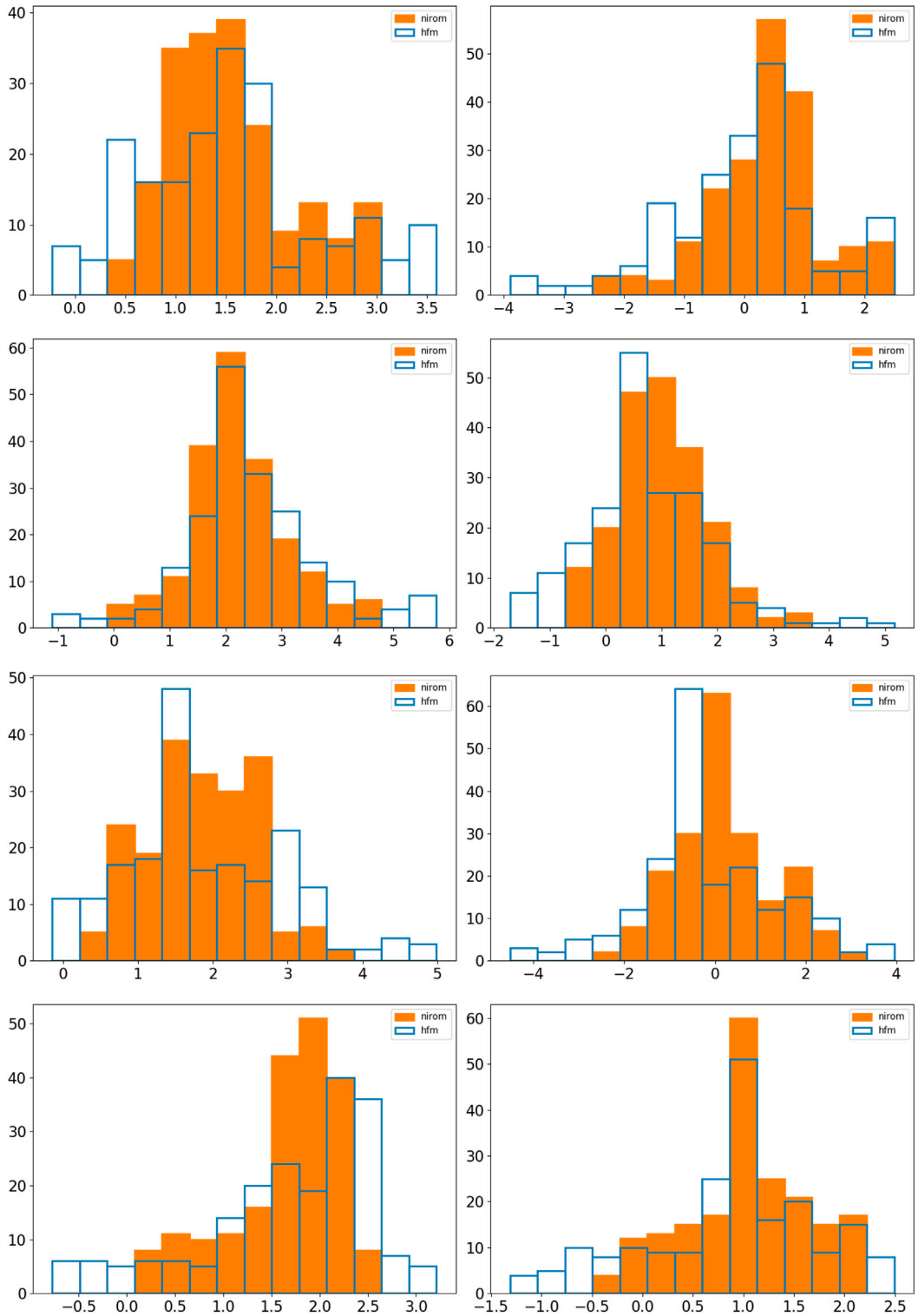


FIGURE 12 | Here we show histograms for the 9 by 9 test case at points 1, 2, 3 and 4 (see **Table 3**) in rows from top to bottom respectively. The first column of graphs show the x – component of velocity and the second the y – component. We compare in these graphs the histograms (or probability density functions) of the velocity components of the HFM (not seen in training), blue, and the prediction from NIROM in orange.

3.1 Dimensionality Reduction

The solutions from the 6 by 6 test case were interpolated onto central subdomains of size 0.5 by 0.5 for every time level. 95,000 snapshots were created in total for the velocity CAE (by selecting 95,000 randomly located and orientated grids, at a randomly selected time levels): 76,000 snapshots were used for training data and the remaining 19,000 snapshots were used as validation data. For the buildings field, a similar procedure was used to generate 95,000 snapshots. However, as the buildings do not change in time, there was no random sampling in time. After optimisation of both networks simultaneously, the chosen hyperparameter values are shown in **Table 1** and the architectures can be found in **Table 2**. **Figure 5** shows the velocity and buildings fields from the HFM (unseen example) and the reconstruction from the autoencoders for one subdomain. The velocity components are very well reconstructed and the buildings field is captured well. The pointwise error of the latter is confined to an extremely small region around the edge of the buildings.

3.2 Prediction for the 6 by 6 Test Case

The solutions from the 6 by 6 test case were interpolated onto the star-shaped grids (see **Figure 1**), within which, each subdomain is of size 0.5 by 0.5 (one 12th of the domain size and thus 144 subdomains fit into the 6 by 6 domain). To form an input-output pair for training, two successive time levels were chosen at random (from time levels 50 to 349) and the velocity fields associated with both time levels were interpolated onto the grid in order to have all the variables required by the PAN, see **Eq. 5**. The buildings field was interpolated onto the central subdomain of the grid. 75,000 snapshots were created in total for the PAN: 50,000 input-output pairs were used for training data and the remaining 25,000 input-output pairs were used as validation data. Hyperparameter optimisation was performed, revealing the optimal values for the PAN which are shown in **Table 1**, and the architectures in **Table 2**.

Once trained, the method is tested by predicting in time. An initial condition is used, based on the HFM results at time level 50, and the method described in Algorithm 1 and the accompanying text is used to march forward from time level 50 up to time level 400. **Figures 6** and **7** show the prediction of the adversarial network for two particular time levels beyond the training dataset but for the same buildings configuration as the training dataset. The ROM captures the velocity magnitudes well. It has managed to capture the areas where there are high velocities, in comparison to the HFM, although its resolution is reduced. Impressively, it is also able to capture many of the eddy structures that result from the interaction of the fluid with the buildings. Again we assume the truth is the HFM simulation when comparing the two images. This NIROM simulation would be expected, eventually, to deviate from the HFM as it is a chaotic flow and small velocity deviations will build up, potentially changing the flow structures significantly.

In **Figure 8** we plot all the nodes within the domain, with the nodes inside buildings appearing as blue and those outside the buildings as red. Thus we can see the position of the buildings and the density of the mesh at this instance in time, which corresponds to the results shown in **Figure 6**. We also show four points within the domain, in black, where we will plot the histograms, or probability density functions, of the $x-$ and $y-$ components of velocity, taken over time level 50 to time level 400. These histograms are shown in **Figure 9** and the coordinates of the points are given in **Table 3**. We see a qualitative agreement in terms of the statistics of the fluctuations and the range of velocities between the HFM and the NIROM. The narrower the histograms, the smaller the magnitude of the fluctuations in the velocity components. Thus, generally speaking the NIROM tends to fluctuate less than the HFM, probably because it has a little less resolution than the HFM. It also (again because of reduced resolution) has less frequently occurring large values of the velocity. However, given the complexity of the flows, the NIROM does remarkably well, even though there are some histograms that do not compare quite so well, such as the $x-$ component of the velocity at point 4.

3.3 Prediction for the Unseen 9 by 9 Test Case

Now an unseen configuration of buildings is used and the domain is increased from 6 by 6 to 9 by 9. The HFM is solved in order to have boundary conditions for the ROM. In the future, alternative methods to generate boundary conditions will be explored, including methods based on using the training data from the HFM [3], but also methods based on generative networks, which will ensure that the ROM is independent of the HFM in this regard. The initial condition for the NIROM is taken from time level 50 of the HFM. The domain is now split into 324 subdomains (of size 0.5 by 0.5). The predictive adversarial network is used to generate a solution for each internal subdomain (i.e., each subdomain that does not share an edge with the boundary). All internal subdomains are swept through until the global solution converges. Time-marching is applied to solve from the initial condition at time level 50 to time level 250, as outlined in Algorithm 1. Within each time step, the number of iterations needed for convergence is approximately 20, about 4 more than for the previous 6 by 6 problem. Convergence is assumed when the difference between latent variables associated with compressed velocity (outputs of the PAN in each of the 324 subdomains) is less than $e-4$ given that the magnitude of the latent variables is $\mathcal{O}(1)$ as $\boldsymbol{z} \sim \mathcal{N}(0,1)$. Predictions from the NIROM of the velocity magnitude at time level 250 can be seen in **Figure 10**. The regions of high speed (shown in red) are picked up by the ROM and promising agreement is obtained between the HFM and ROM. The two lower plots in **Figure 10** show the velocity magnitude and velocity vectors for the HFM (left) and the NIROM (right) over $[1,4]\times[4,7]$. The NIROM captures the flow path and some of the larger eddies, but does miss some of the smaller ones. The magnitude of the NIROM's velocities is generally slightly less than those of the HFM. The detail in the velocity vectors suggest chaotic flow. One would not expect the CFD and the NIROM to produce exactly the same results, because of the chaotic nature of these flows. Finally, in **Figure 11** we plot all the nodes within the 9 by 9 domain, with the nodes inside buildings appearing as blue and those outside the buildings as red. We also show four points within the domain, in black, where we will plot the histograms, or probability density functions, of the $x-$ and $y-$ components of velocity taken over time level 50 to time level 250. These histograms are shown in **Figure 12** and the coordinates of the points are given in **Table 3**.

Again, we see a qualitative agreement in terms of the statistics of the fluctuations and the range of velocities between the HFM and the NIROM. As for the 6 by 6 domain, generally speaking the NIROM tends to fluctuate less than the HFM, probably because it has a little less resolution than the HFM. However, given the complexity of the flows and the fact that this is an unseen domain with an unseen configuration of buildings, the NIROM does extremely well.

4 CONCLUSIONS AND FURTHER WORK

Here we have presented a data-driven or Machine Learning (ML) based non-intrusive reduced-order model (NIROM) which is capable of making predictions for a significantly larger domain than the one used to generate the snapshots or training data. This is a unique development and one which we hope paves the way to develop ML-based NIROMs that can make good predictions for unseen scenarios. Ultimately these methods could complement Computational Fluid Dynamics (CFD) codes when solving flow fields in urban environments as well as other CFD applications. This development relies on the combination of a novel way of sampling the training data [which can free the reduced-order model from the restriction of the domain of the high-fidelity model (HFM)] and a domain decomposition approach (which decomposes unseen geometries in a manner consistent with the sub-sampling approach).

The main conclusions are that: (1) one can predict (with the NIROM) the chaotic transient flows within the 2D problems, although sometimes the resolution is reduced in comparison to the CFD simulations; (2) the adversarial layer of the prediction algorithm is important in order to form stable solutions that remain within the distribution of the training data; (3) a convolutional autoencoder is able to compress the velocity and buildings fields to a high degree of accuracy; and (4) the approach was applied to make predictions for a domain of over twice the area and over twice the number of buildings as in the HFM used to generate the training data.

Future work will involve: (1) extending the problem domains to 3D and using more realistic building profiles; (2) generating boundary conditions with a generative network rather than using the CFD code, resulting in a method fully independent of the high-fidelity model; (3) using the residuals of the differential equations within the training procedure (Physics-Informed methods, for example, see [51]) and forcing the equation residuals to zero within the prediction step by using a method similar to the Residual DEIM approach [52].

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

CH, CP, PS, and IN contributed to conception and design of the study. XL, HG, ZW, CH, and CP worked on the software. CH, XL, HG, and CP worked on the methodology. CH, XL, HG, and CP worked on the first draft of the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

The authors would like to acknowledge the following EPSRC grants: RELIANT, Risk EvaLuatIon fAst iNtelligent Tool for COVID19 (EP/V036777/1); MAGIC, Managing Air for Green Inner Cities (EP/N010221/1); MUFFINS, MUltiphase Flow-induced Fluid-flexible structure InteractioN in Subsea applications (EP/P033180/1); the PREMIERE programme grant (EP/T000414/1); and INHALE, Health assessment across biological length scales (EP/T003189/1).

ACKNOWLEDGMENTS

We are grateful to Imperial College for use of their UKRI Open Access Block Grant, which has funded publication of this work. We would also like to thank the reviewers for their comments and suggestions, which have improved the paper.

REFERENCES

1. W Schilders, H van der Vorst, J Rommes, editors. *Model Order Reduction: Theory, Research Aspects and Applications*. Springer (2008). vol. 13 of The European Consortium for Mathematics in Industry.
2. Benner P, Gugercin S, Willcox K. A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems. *SIAM Rev* (2015) 57:483–531. doi:10.1137/130932715
3. Heaney CE, Wolffs Z, Tómasson JA, Kahouadji L, Salinas P, Nicolle A, et al. An AI-Based Non-intrusive Reduced-Order Model for Extended Domains Applied to Multiphase Flow in Pipes. *Phys Fluids* (2022) 34:055111. doi:10.1063/5.0088070
4. Brunton SL, Noack BR, Koumoutsakos P. Machine Learning for Fluid Mechanics. *Annu Rev Fluid Mech* (2020) 52:477–508. doi:10.1146/annurev-fluid-010719-060214
5. Buchan AG, Pain CC, Fang F, Navon IM. A POD Reduced-Order Model for Eigenvalue Problems with Application to Reactor Physics. *Int J Numer Meth Engng* (2013) 95:1011–32. doi:10.1002/nme.4533
6. Fang F, Zhang T, Pavlidis D, Pain CC, Buchan AG, Navon IM. Reduced Order Modelling of an Unstructured Mesh Air Pollution Model and Application in 2D/3D Urban Street Canyons. *Atmos Environ* (2014) 96:96–106. doi:10.1016/j.atmosenv.2014.07.021
7. Ballarin F, Rozza G. POD-galerkin Monolithic Reduced Order Models for Parametrized Fluid-Structure Interaction Problems. *Int J Numer Meth Fluids* (2016) 82:1010–34. doi:10.1002/fld.4252
8. Fukami K, Murata T, Zhang K, Fukagata K. Sparse Identification of Nonlinear Dynamics with Low-Dimensionalized Flow Representations. *J Fluid Mech* (2021) 926:A10. doi:10.1017/jfm.2021.697
9. Kadeethum T, Ballarin F, Choi Y, O'Malley D, Yoon H, Bouklas N. Non-intrusive Reduced Order Modeling of Natural Convection in Porous media Using Convolutional Autoencoders: Comparison with Linear Subspace Techniques. *Adv Water Resour* (2022) 160:104098. doi:10.1016/j.advwatres.2021.104098
10. Maulik R, Lusch B, Balaprakash P. Reduced-order Modeling of Advection-Dominated Systems with Recurrent Neural Networks and Convolutional Autoencoders. *Phys Fluids* (2021) 33:037106. doi:10.1063/5.0039986

11. Audouze C, De Vuyst F, Nair PB. Nonintrusive Reduced-Order Modeling of Parametrized Time-dependent Partial Differential Equations. *Numer Methods Partial Differential Eq* (2013) 29:1587–628. doi:10.1002/num.21768
12. Bui-Thanh T, Damodaran M, Willcox K. Proper Orthogonal Decomposition Extensions for Parametric Applications in Compressible Aerodynamics. In: *21st AIAA Applied Aerodynamics Conference* (2003). Florida: AIAA. doi:10.2514/6.2003-4213
13. Guénot M, Lepot I, Sainvitu C, Goblet J, Filomeno Coelho R. Adaptive Sampling Strategies for Non-intrusive POD-based Surrogates. *Eng Computations* (2013) 30:521–47. doi:10.1108/02644401311329352
14. Hesthaven JS, Ubbiali S. Non-intrusive Reduced Order Modeling of Nonlinear Problems Using Neural Networks. *J Comput Phys* (2018) 363:55–78. doi:10.1016/j.jcp.2018.02.037
15. Wang Z, Xiao D, Fang F, Govindan R, Pain CC, Guo Y-K. Model Identification of Reduced Order Fluid Dynamics Systems Using Deep Learning. *Int J Numer Methods Fluids* (2017) 86:255–68. doi:10.1002/fld.4416
16. Wiewel S, Becher M, Thuerey N. Latent Space Physics: Towards Learning the Temporal Evolution of Fluid Flow. *Comput Graphics Forum* (2019) 38:71–82. doi:10.1111/cgf.13620
17. Maulik R, Botsas T, Ramachandra N, Mason LR, Pan I. Latent-space Time Evolution of Non-intrusive Reduced-Order Models Using Gaussian Process Emulation. *Physica D: Nonlinear Phenomena* (2021) 416:132797. doi:10.1016/j.physd.2020.132797
18. Ahmed SE, San O, Rasheed A, Iliescu T. Nonlinear Proper Orthogonal Decomposition for Convection-Dominated Flows. *Phys Fluids* (2021) 33:121702. doi:10.1063/5.0074310
19. Fresca S, Manzoni A. POD-DL-ROM: Enhancing Deep Learning-Based Reduced Order Models for Nonlinear Parametrized PDEs by Proper Orthogonal Decomposition. *Comput Methods Appl Mech Eng* (2022) 388:114181. doi:10.1016/j.cma.2021.114181
20. Maulik R, Lusch B, Balaprakash P. Non-autoregressive Time-Series Methods for Stable Parametric Reduced-Order Models. *Phys Fluids* (2020) 32:087115. doi:10.1063/5.0019884
21. Quilodrán-Casas C, Arcucci R, Mottet L, Guo Y-K, Pain CC. Adversarial Autoencoders and Adversarial LSTM for Improved Forecasts of Urban Air Pollution Simulations (2021). *arXiv*. doi:10.48550/arXiv.2104.06297
22. Quilodrán-Casas C, Arcucci R, Pain CC, Guo Y-K. Adversarially Trained LSTMs on Reduced Order Models of Urban Air Pollution Simulations (2021). *arXiv*. doi:10.48550/arXiv.2101.01568
23. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative Adversarial Networks (2014). *arXiv*. doi:10.48550/arXiv.1406.2661
24. Makhzani A, Shlens J, Jaitly N, Goodfellow I, Frey B. Adversarial Autoencoders (2015). *arXiv*. doi:10.48550/arXiv.1511.05644
25. Cheng M, Fang F, Pain CC, Navon IM. An Advanced Hybrid Deep Adversarial Autoencoder for Parameterized Nonlinear Fluid Flow Modelling. *Comput Methods Appl Mech Eng* (2020) 372:113375. doi:10.1016/j.cma.2020.113375
26. Silva VLS, Heaney CE, Pain CC. Data Assimilation Predictive GAN (DA-PredGAN): Applied to Determine the Spread of COVID-19 (2021). *arXiv*. doi:10.48550/arXiv.2105.07729
27. Sanchez-Gonzalez A, Godwin J, Pfaff T, Ying R, Leskovec J, Battaglia PW. Learning to Simulate Complex Physics with Graph Networks. In: A Singh, editors. *Proceedings of the 37th International Conference on Machine Learning (PMLR)* (2020). jmlr.org
28. Baiges J, Codina R, Idelsohn S. A Domain Decomposition Strategy for Reduced Order Models. Application to the Incompressible Navier-Stokes Equations. *Comput Methods Appl Mech Eng* (2013) 267:23–42. doi:10.1016/j.cma.2013.08.001
29. Xiao D, Fang F, Heaney CE, Navon IM, Pain CC. A Domain Decomposition Method for the Non-intrusive Reduced Order Modelling of Fluid Flow. *Comput Methods Appl Mech Eng* (2019) 354:307–30. doi:10.1016/j.cma.2019.05.039
30. Xiao D, Heaney CE, Fang F, Mottet L, Hu R, Bistrian DA, et al. A Domain Decomposition Non-intrusive Reduced Order Model for Turbulent Flows. *Comput Fluids* (2019) 182:15–27. doi:10.1016/j.compfluid.2019.02.012
31. Yang LM, Grooms I. Machine Learning Techniques to Construct Patched Analog Ensembles for Data Assimilation. *J Comput Phys* (2021) 443:110532. doi:10.1016/j.jcp.2021.110532
32. Schmid PJ. Dynamic Mode Decomposition of Numerical and Experimental Data. *J Fluid Mech* (2010) 656:5–28. doi:10.1017/S0022112010001217
33. Brunton SL, Proctor JL, Kutz JN. Discovering Governing Equations from Data by Sparse Identification of Nonlinear Dynamical Systems. *Proc Natl Acad Sci U.S.A* (2016) 113:3932–7. doi:10.1073/pnas.1517384113
34. Brunton SL, Kutz JN. Methods for Data-Driven Multiscale Model Discovery for Materials. *J Phys Mater* (2019) 2:044002. doi:10.1088/2515-7639/ab291e
35. Bistrian DA, Navon IM. An Improved Algorithm for the Shallow Water Equations Model Reduction: Dynamic Mode Decomposition vs POD. *Int J Numer Meth Fluids* (2015) 78:552–80. doi:10.1002/fld.4029
36. Carlberg KT, Jameson A, Kochenderfer MJ, Morton J, Peng L, Witherden FD. Recovering Missing CFD Data for High-Order Discretizations Using Deep Neural Networks and Dynamics Learning. *J Comput Phys* (2019) 395:105–24. doi:10.1016/j.jcp.2019.05.041
37. Eivazi H, Veisi H, Naderi MH, Esfahanian V. Deep Neural Networks for Nonlinear Model Order Reduction of Unsteady Flows. *Phys Fluids* (2020) 32:105104. doi:10.1063/5.0020526
38. Vlachas PR, Arampatzis G, Uhler C, Koumoutsakos P. Multiscale Simulations of Complex Systems by Learning Their Effective Dynamics. *Nat Machine Intelligence* (2022) 4:359–66. doi:10.1038/s42256-022-00464-w
39. Hasegawa K, Fukami K, Murata T, Fukagata K. Machine-learning-based Reduced-Order Modeling for Unsteady Flows Around bluff Bodies of Various Shapes. *Theor Comput Fluid Dyn* (2020) 34:367–83. doi:10.1007/s00162-020-00528-w
40. Gastaldi L. A Domain Decomposition Method Associated with the Streamline Diffusion FEM for Linear Hyperbolic Systems. *Appl Numer Maths* (1992) 10:357–80. doi:10.1016/0168-9274(92)90057-K
41. Pain C, Umpleby A, de Oliveira C, Goddard A. Tetrahedral Mesh Optimisation and Adaptivity for Steady-State and Transient Finite Element Calculations. *Comput Methods Appl Mech Eng* (2001) 190:3771–96. doi:10.1016/S0045-7825(00)00294-2
42. Gonzalez FJ, Balajewicz M. Deep Convolutional Recurrent Autoencoders for Learning Low-Dimensional Feature Dynamics of Fluid Systems (2018). *arXiv*. doi:10.48550/arXiv.1808.01346
43. Xu J, Duraisamy K. Multi-level Convolutional Autoencoder Networks for Parametric Prediction of Spatio-Temporal Dynamics. *Comput Methods Appl Mech Eng* (2020) 372:113379. doi:10.1016/j.cma.2020.113379
44. Wu P, Gong S, Pan K, Qiu F, Feng W, Pain CC. Reduced Order Model Using Convolutional Auto-Encoder with Self-Attention. *Phys Fluids* (2021) 33:077107. doi:10.1063/5.0051155
45. Nikolopoulos S, Kalogeris I, Papadopoulos V. Non-intrusive Surrogate Modeling for Parametrized Time-dependent PDEs Using Convolutional Autoencoders (2021). *arXiv*. doi:10.48550/arXiv.2101.05555
46. Makkie M, Huang H, Zhao Y, Vasilakos AV, Liu T. Fast and Scalable Distributed Deep Convolutional Autoencoder for fMRI Big Data Analytics. *Neurocomputing* (2019) 325:20–30. doi:10.1016/j.neucom.2018.09.066
47. Salinas P, Pavlidis D, Xie Z, Jacquemyn C, Melnikova Y, Jackson MD, et al. Improving the Robustness of the Control Volume Finite Element Method with Application to Multiphase Porous media Flow. *Int J Numer Methods Fluids* (2017) 85:235–46. doi:10.1002/fld.4381
48. Obeysekara A, Salinas P, Heaney CE, Kahouadji L, Via-Estrem L, Xiang J, et al. Prediction of Multiphase Flows with Sharp Interfaces Using Anisotropic Mesh Optimisation. *Adv Eng Softw* (2021) 160:103044. doi:10.1016/j.advengsoft.2021.103044
49. Via-Estrem L, Salinas P, Xie Z, Xiang J, Latham J-P, Douglas S, et al. Robust Control Volume Finite Element Methods for Numerical Wave Tanks Using Extreme Adaptive Anisotropic Meshes. *Int J Numer Methods Fluids* (2020) 92:1707–22. doi:10.1002/fld.4845

50. Shams-ul Islam S, Nazeer G, Ying ZC. Numerical Investigation of Flow Past 17-cylinder Array of Square Cylinders. *AIP Adv* (2018) 8:065004. doi:10.1063/1.5022360
51. Chen W, Wang Q, Hesthaven JS, Zhang C. Physics-informed Machine Learning for Reduced-Order Modeling of Nonlinear Problems. *J Comput Phys* (2021) 446:110666. doi:10.1016/j.jcp.2021.110666
52. Xiao D, Fang F, Buchan A, Pain C, Navon I, Du J, et al. Non-linear Model Reduction for the Navier-Stokes Equations Using Residual DEIM Method. *J Comput Phys* (2014) 263:1–18. doi:10.1016/j.jcp.2014.01.011

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers | Frontiers in Physics

ORIGINAL RESEARCH
published: 24 August 2022
doi: 10.3389/fphy.2022.900064

Component-Based Reduced Order Modeling of Large-Scale Complex Systems

Cheng Huang[1]*, *Karthik Duraisamy*[2] *and Charles Merkle*[3]

[1]Department of Aerospace Engineering, University of Kansas, Lawrence, KS, United States, [2]Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, United States, [3]School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN, United States

Large-scale engineering systems, such as propulsive engines, ship structures, and wind farms, feature complex, multi-scale interactions between multiple physical phenomena. Characterizing the operation and performance of such systems requires detailed computational models. Even with advances in modern computational capabilities, however, high-fidelity (e.g., large eddy) simulations of such a system remain out of reach. In this work, we develop a reduced-order modeling framework to enable accurate predictions of large-scale systems. We target engineering systems which are difficult to simulate at a high-enough level of fidelity, but are decomposable into different components. These components can be modeled using a combination of strategies, such as reduced-order models (ROM) or reduced-fidelity full-order models (RF-FOM). Component-based training strategies are developed to construct ROMs for each individual component. These ROMs are then integrated to represent the full system. Notably, this approach only requires high-fidelity simulations of a much smaller computational domain. System-level responses are mimicked *via* external boundary forcing during training. Model reduction is accomplished using model-form preserving least-squares projections with variable transformation (MP-LSVT) (Huang et al., Journal of Computational Physics, 2022, 448: 110742). Predictive capabilities are greatly enhanced by developing adaptive bases which are locally linear in time. The trained ROMs are then coupled and integrated into the framework to model the full large-scale system. We apply the methodology to extremely complex flow physics involving combustion dynamics. With the use of the adaptive basis, the framework is demonstrated to accurately predict local pressure oscillations, time-averaged and RMS fields of target state variables, even with geometric changes.

Keywords: reduced order modeling, domain decomposition, model reduction, turbulent reacting flows, adaptive basis

OPEN ACCESS

Edited by:
Traian Iliescu,
Virginia Tech, United States

Reviewed by:
Tommaso Taddei,
Institut National de Recherche en Informatique et en Automatique (INRIA), France
Luca Dede',
Politecnico di Milano, Italy

***Correspondence:**
Cheng Huang
chenghuang@ku.edu

Specialty section:
This article was submitted to Statistical and Computational Physics, a section of the journal Frontiers in Physics

Received: *19 March 2022*
Accepted: *16 May 2022*
Published: *24 August 2022*

Citation:
Huang C, Duraisamy K and Merkle C (2022) Component-Based Reduced Order Modeling of Large-Scale Complex Systems. Front. Phys. 10:900064. doi: 10.3389/fphy.2022.900064

1 INTRODUCTION

Rapid advancements in computing technologies are enabling high-fidelity simulations of complex, multi-scale physics (e.g., turbulence [1] and combustion [2,3]) observed in real engineering systems. These simulations provide insight into the underlying physics, which cannot be quantitatively accessed through experiments. This insight is useful in improving the performance of engineering

systems and reducing failures. However, the high computational costs of high-fidelity simulations prohibit their integration into design and analysis of full-scale systems, which require repeated simulations to explore parameter spaces. One popular approach to address such challenges is through model order reduction (MOR), a common approach being projection-based reduced-order models (ROM) [4–6], which have demonstrated success in many applications such as flow control [7–9], aeroelasticity [10,11], hypersonics [12], and combustion [13,14]. Typically, the construction of ROMs involves three stages: 1) an offline training stage that performs high-fidelity simulations of the target systems for multiple parameters; 2) offline construction of reduced basis and projections on low-dimensional manifolds; and 3) online execution of ROMs by projecting the governing equations on the low-dimensional manifold. Despite the many successful examples of MOR, their direct applications in many practical large-scale engineering systems remain infeasible because the systems are so complex that high-fidelity simulations are completely inaccessible. Using an example from rocket combustion, a coarse-mesh ("low"-fidelity) large eddy simulation (LES) of a small-scale rocket engine [15] requires $O(10^7)$ CPU-hours, which even makes a single fine-mesh (high-fidelity) LES of this type of problems inaccessible, (estimated to require $> O(10^9)$ CPU-hours), let alone the high-fidelity LES of a large-scale rocket engine [16], a computation that would require > 10 times the resources of the small-scale problem.

To address this specific challenge of the lack of full-order model (FOM) data for large-scale systems, researchers have formulated domain-decomposition methods [17,18], or component-based methods [19] to develop a network of ROMs to model the target system. In addition to ROM applications, such ideas have been commonly used for computational fluid dynamics [20,21], port-hamiltonian system [22–24] etc. For consistency, we refer to this family of approaches as component-based reduced-order modeling (CBROM) methods in the current paper. CBROM methods leverage the fact that many large-scale engineering systems can be decomposed into components of identical features and the offline training of the ROMs can be performed based on each individual component for multiple parameters, which significantly reduces the cost of the offline training. The trained component-based ROMs can then be used for the identical components and coupled together to model different configurations of the large-scale systems.

To date, the majority of the success of CBROM methods has been in problems governed by linear PDEs. Willcox et al. [25] demonstrated the feasibility of constructing low-order models of blade row unsteady aerodynamics in a compressor. Maday and Ronquist [26] formulated a reduced basis element method and applied it to a thermal fin problem. Iapichino et al. [27] proposed a reduced basis hybrid method to solve the steady Stokes problem with applications to cardiovascular networks. Adopting the static-condensation reduced-basis-element method [19,28,29], Kapteyn et al. [30] demonstrated the development of a digital twin for a 12-ft wingspan unmanned aerial vehicle *via* a library of component-based ROMs. More recently, McBane and Choi [31] leveraged the static-condensation reduced-basis-element method and demonstrated a 1000× speedup with relative error $< 1\%$ for lattice-type structure design using component-wise reduced-order modeling.

In addition, some applications of component-based ROMs on nonlinear PDEs can also be found in the literature. One group of studies incorporate the FOM to model a subset of components in the target system while applying ROMs to the other components. Lucia et al. [32] demonstrated a combination of ROMs and FOM by domain-decomposition for modeling two-dimensional high-speed flows with moving shock waves, by applying the FOM for the shock-containing domain and ROMs for the other domains. Buffoni et al. [33] demonstrated similar ideas by partitioning the computational domain into two subdomains (one modeled using FOM while the other by ROM), and presented different approaches to couple ROM with FOM. Baiges et al. [34] demonstrated the improvement in predicting flow configurations that are not present in the training snapshots by integrating the FOM into the component-based modeling framework. Ahmed et al. [35] presented a hybrid analysis and modeling approach combining a physics-based FOM and a data-driven non-intrusive ROM towards predictive digital twin technologies. Another group of investigations aim at incorporating only ROMs rather than hybrid FOMs/ROMs in the component-based modeling framework. Hoang et al. [36] proposed the domain-decomposition least-squares Petrov-Galerkin (DD-LSPG) model-reduction method for parameterized systems of nonlinear algebraic equations. Xiao et al. [37–39] developed a domain-decomposition non-intrusive reduced-order model (DDNIROM) for turbulent flows. The current authors demonstrated the integration of component-based ROMs with a FOM in a quasi-1D Euler problem [40,41].

In the present work, we develop a component-based modeling framework that can flexibly adopt either reduced-order models (ROM) or full-order models (FOM) for different components of the target system based on the corresponding requirements of modeling accuracy and efficiency with the goal of enabling:

(1) Accurate simulations of large-scale systems, which cannot be directly accessed using high-fidelity simulations;
(2) Parametric studies of the large-scale system targeting many-query applications.

It is notable that in the current work, we choose rocket engine as the target system. This application involves compressible, reacting, chaotic flows and thus introduces complex challenges for reduced order modeling. We establish a component-based training strategy for ROM development, which only requires the high-fidelity simulations of the individual components, rather than the entire system. The trained ROMs are then coupled together (either with each other or with FOM) *via* a direct flux matching method for information transfer between components. The ROM formulation leverages model reduction techniques using model-form preserving least-squares projections with variable transformation (MP-LSVT) with physical realizability enforced on both temperature and species mass fractions [14] to achieve both global and local stabilization. Furthermore, the MP-LSVT ROM is incorporated with basis adaptation to achieve significant enhancement in modeling accuracy. Since our interests are focused on engineering systems involving combustion, we use extremely challenging turbulent reacting flow examples, relevant to rocket applications, to motivate and evaluate our framework. But it should be highlighted that our

component-based modeling framework is applicable to many other, unrelated disciplines, featuring systems that can be decomposed into different components.

The remainder of the paper is organized as follows. **Section 2** presents the full-order model (FOM) and time discretization. **Section 3** reviews the procedure for model reduction *via* MP-LSVT formulation. **Section 4** discusses basis-adaptation algorithms for ROM enhancement. **Section 5** presents the domain-decomposition framework, including both the component-based ROM training strategy and integration method in full system. **Section 6** presents numerical results based on single- and multi-injector model rocket combustor configurations with detailed assessment on the accuracy of the framework. In **Section 7**, we provide concluding remarks and perspectives.

2 FULL-ORDER MODEL

We define the physical domain Ω with boundary $\partial\Omega$, and then represent the governing equations of the full-order model (FOM) for Ω as a generic dynamical system

$$\begin{aligned} \frac{\mathrm{d}\mathbf{q}(\mathbf{q}_p)}{\mathrm{dt}} &= \mathbf{f}(\mathbf{q}_p, \mathrm{t}) \quad \text{in} \quad \Omega, \\ \text{with} \quad \mathbf{u}(\mathbf{q}_p) &= \mathbf{u}_{BC} \quad \text{on} \quad \partial\Omega, \\ \text{and} \quad \mathbf{q}_p(\mathrm{t}=0) &= \mathbf{q}_p{}^0, \end{aligned} \tag{1}$$

where $\mathrm{t} \in [0, T]$ is the solution time, which spans the time interval from 0 to T, $\mathbf{q}_p$: $[0,T] \to \mathbb{R}^N$ is the vector of solution (or state) variables, $\mathbf{u}_{BC}$: $[0,T] \to \mathbb{R}^{N_b}$ is the vector of states to be enforced at the boundary $\partial\Omega$ (i.e., boundary conditions), $\mathbf{q}_p{}^0 \in \mathbb{R}^N$ is the vector of states to be specified as the initial conditions at $t = 0$, $\mathbf{q}$: $\mathbb{R}^N \to \mathbb{R}^N$, $\mathbf{f}$: $\mathbb{R}^N \times [0,T] \to \mathbb{R}^N$, and $\mathbf{u}$: $\mathbb{R}^N \to \mathbb{R}^{N_b}$ are (typically highly non-linear) functions of $\mathbf{q}_p$. N is the total number of degrees of freedom in the system (e.g., for finite volume/element method, $N = N_{elem} \times N_{var}$, where N_{elem} is the total number of elements and N_{var} is the number of state variables in each element). N_b is the total number of degrees of freedom associated with the boundary $\partial\Omega$ and $N_b = N_{elem,BC} \times N_{var}$, where $N_{elem,BC}$ is the number of elements adjacent to the boundary $\partial\Omega$. For a FOM based on conservation laws, the function, $\mathbf{q}$, represents the conservative state. The function, $\mathbf{f}$, represents surface fluxes, source terms, and body forces arising from the spatial discretization of the governing equations. $\mathbf{u}$ represents the boundary condition state, and $\mathbf{u}_{BC}$ denotes the values of the state to be satisfied at the boundary.

Different time-discretization methods can be introduced to solve **Eq. 1** (e.g., linear multi-step, or Runge–Kutta methods [42]). For all the numerical examples presented in the current paper, we use linear multi-step methods for both FOM and ROM calculations and refer the reader to [14] for details. An l-step version of linear multi-step methods can be expressed as

$$\begin{aligned} \mathbf{r}(\mathbf{q}_p{}^n) \triangleq \mathbf{q}(\mathbf{q}_p{}^n) + \sum_{j=1}^{l} \alpha_j \mathbf{q}(\mathbf{q}_p{}^{n-j}) - \Delta\mathrm{t}\beta_0 \mathbf{f}(\mathbf{q}_p{}^n, \mathrm{t}^n) - \Delta\mathrm{t} \sum_{j=1}^{l} \beta_j \mathbf{f}(\mathbf{q}_p{}^{n-j}, \mathrm{t}^{n-j}) = 0 \quad (n \geq l), \\ \text{with} \quad \mathbf{u}(\mathbf{q}_p{}^n) = \mathbf{u}_{BC}{}^n \quad \text{on} \quad \partial\Omega. \end{aligned} \tag{2}$$

where $\Delta\mathrm{t} \in \mathbb{R}^+$ is the physical time step for the numerical solution, and the coefficients α_j, $\beta_j \in \mathbb{R}$ are determined based on l. If $\beta_0 = 0$, the method is explicit; otherwise, the method is implicit. $\mathbf{r}$: $\mathbb{R}^N \to \mathbb{R}^N$ is defined as the FOM equation residual. The state variables, $\mathbf{q}_p{}^n$, are solved for at each time step so that $\mathbf{r}(\mathbf{q}_p{}^n) = \mathbf{0}$.

3 MODEL-FORM PRESERVING MODEL REDUCTION FOR TRANSFORMED SOLUTION VARIABLES

For problems involving multiscale phenomena with strong convection and non-linear effects, it is well-recognized that ROM robustness can be a major issue. To address this challenge, we pursue the model-form preserving least-squares with variable transformation (MP-LSVT) formulation to construct the reduced-order model (ROM). This methodology is described below—we refer the reader to ref [14] for further details.

3.1 Construction of Proper Orthogonal Decomposition Bases for Solution Variables

The state $\mathbf{q}_p$ in **Eq. 1** can be expressed in a trial space $\mathcal{V}_p \triangleq \mathrm{Range}(\mathbf{V}_p)$, where $\mathbf{V}_p \in \mathbb{R}^{N \times n_p}$ is the trial basis matrix. Define $\mathbf{q}'_p(\mathrm{t}) \triangleq \mathbf{q}_p(\mathrm{t}) - \mathbf{q}_{p,\mathrm{ref}}$, where $\mathbf{q}_{p,\mathrm{ref}}$ is a reference state. Possible reference states include the initial FOM solution, $\mathbf{q}_{p,\mathrm{ref}} = \mathbf{q}_p(\mathrm{t} = \mathrm{t}_0)$, or the time-averaged FOM solution, $\mathbf{q}_{p,\mathrm{ref}} = \frac{1}{\mathrm{t}_1 - \mathrm{t}_0} \int_{\mathrm{t}_0}^{\mathrm{t}_1} \mathbf{q}_p(\mathrm{t})\, \mathrm{dt}$. We then seek a representation $\tilde{\mathbf{q}}_p$: $[0,T] \to \mathcal{V}_p$ such that

$$\mathbf{H}(\tilde{\mathbf{q}}_p - \mathbf{q}_{p,\mathrm{ref}}) = \mathbf{V}_p \mathbf{q}_r \tag{3}$$

where $\mathbf{q}_r$: $[0,T] \to \mathbb{R}^{n_p}$ is the reduced state with n_p representing the number of trial basis modes. In this work, $\mathbf{V}_p$ is computed *via* the proper orthogonal decomposition (POD) [4] from the singular value decomposition (SVD), which is a solution to

$$\min_{\mathbf{V}_p \in \mathbb{R}^{N \times n_p}} \|\mathbf{Q} - \mathbf{V}_p \mathbf{V}_p{}^T \mathbf{Q}\|_F \quad s.t. \quad (\mathbf{V}_p)^T \mathbf{V}_p = \mathbf{I}, \tag{4}$$

where $\mathbf{Q}$ is a data matrix in which each column is a snapshot of the solution $\mathbf{q}'_p$ at different time instances. A scaling matrix $\mathbf{H} \in \mathbb{R}^{N \times N}$ must be applied to $\mathbf{q}'_p$ such that the variables corresponding to different physical quantities in the data matrix $\mathbf{Q}$ have similar orders of magnitude. Otherwise, $\mathbf{Q}$ may be biased by physical quantities of higher magnitudes (e.g., total energy). In this work, we normalize all quantities by their L^2-norm, as proposed by Lumley and Poje [4].

$$\mathbf{H} = diag(\mathbf{H}_1, \ldots, \mathbf{H}_i, \ldots, \mathbf{H}_{N_{elem}}), \tag{5}$$

where $\mathbf{H}_i = diag(\phi^{-1}_{1,norm}, \ldots, \phi^{-1}_{N_{var},norm})$. Here, $\phi_{v,norm}$ represents the vth state variable and

$$\phi_{v,norm} = \frac{1}{\mathrm{t}_1 - \mathrm{t}_0} \int_{\mathrm{t}_0}^{\mathrm{t}_1} \frac{1}{\Omega} \int_{\Omega} \phi'^2_v(\mathrm{x}, \mathrm{t})\, \mathrm{dx}\, \mathrm{dt}. \tag{6}$$

3.2 Least-Squares With Variable Transformation

Leveraging the least-squares Petrov-Galerkin (LSPG) projection formulation proposed by Carlberg et al. [43], we develop a model-form preserving least-squares formulation for the FOM in **Eq. 1**. Our objective is to minimize the fully-discrete FOM equation residual **r**, defined in **Eq. 2**, on the physical domain Ω with respect to the reduced state, $\mathbf{q}_r$

$$\begin{aligned} \mathbf{q}_r^n &\triangleq \arg\min_{\mathbf{q}_r \in \mathbb{R}^{n_p}} \left\| \mathbf{Pr}\left(\tilde{\mathbf{q}}_p\right) \right\|_2^2, \\ \text{with } & \mathbf{u}\left(\tilde{\mathbf{q}}_p{}^n\right) = \mathbf{u}_{BC}{}^n \text{ on } \partial\Omega \\ \text{and } & \tilde{\mathbf{q}}_p{}^0 = \mathbf{q}_{p,\text{ref}} + \mathbf{H}^{-1}\mathbf{V}_p\left(\mathbf{V}_p\right)^T \mathbf{q}_p{}^0 \end{aligned} \tag{7}$$

where the approximate solution variables, $\tilde{\mathbf{q}}_p = \mathbf{q}_{p,\text{ref}} + \mathbf{H}^{-1}\mathbf{V}_p\mathbf{q}_r$. The equation residual, **r**, is scaled by **P** using the L^2-norm, similar to the scaling matrix **H** in **Eq. 5**

$$\mathbf{P} = diag\left(\mathbf{P}_1, \ldots, \mathbf{P}_i, \ldots, \mathbf{P}_{N_{elem}}\right), \tag{8}$$

where $\mathbf{P}_i = diag\left(\varphi_{1,norm}^{-1}, \ldots, \varphi_{N_{var},norm}^{-1}\right)$. Here, $\varphi_{v,norm}$ represents the vth evaluated quantity of $\mathbf{q}(\mathbf{q}_p)$

$$\varphi_{v,norm} = \frac{1}{t_1 - t_0}\int_{t_0}^{t_1}\frac{1}{\Omega}\int_{\Omega}\varphi_v'^2(x,t)\,dx\,dt, \tag{9}$$

such that each equation in **r** has similar contributions to the minimization problem in **Eq. 7**. It is worth pointing out that the treatment of boundary conditions in the MP-LSVT ROM is fully consistent with the FOM in **Eq. 2**, which guarantees that the boundary conditions are satisfied in the ROM and serves as an important building block in FOM/ROM coupling in the component-based domain-decomposition framework in **Section 5**.

Following **Eq. 7**, a reduced non-linear system of dimension n_p can then be obtained and viewed as the result of a Petrov-Galerkin projection

$$\begin{aligned} \left(\mathbf{W}_p{}^n\right)^T \mathbf{Pr}\left(\tilde{\mathbf{q}}_p{}^n\right) &= \mathbf{0}, \\ \text{with } \mathbf{u}\left(\tilde{\mathbf{q}}_p{}^n\right) &= \mathbf{u}_{BC}{}^n \text{ on } \partial\Omega \\ \text{and } \tilde{\mathbf{q}}_p{}^0 &= \mathbf{q}_{p,\text{ref}} + \mathbf{H}^{-1}\mathbf{V}_p\left(\mathbf{V}_p\right)^T \mathbf{q}_p{}^0 \end{aligned} \tag{10}$$

where $\mathbf{W}_p$ is the test basis

$$\mathbf{W}_p{}^n = \frac{\partial \mathbf{Pr}\left(\tilde{\mathbf{q}}_p{}^n\right)}{\partial \mathbf{q}_r^n} = \mathbf{P}\left(\tilde{\mathbf{\Gamma}}^n - \Delta t\beta_0\tilde{\mathbf{J}}^n\tilde{\mathbf{\Gamma}}^n\right)\mathbf{H}^{-1}\mathbf{V}_p, \tag{11}$$

with $\tilde{\mathbf{J}}^n = [\partial\mathbf{f}/\partial\mathbf{q}]^n_{\mathbf{q}_p=\tilde{\mathbf{q}}_p}$ and $\tilde{\mathbf{\Gamma}}^n = [\partial\mathbf{q}/\partial\mathbf{q}_p]^n_{\mathbf{q}_p=\tilde{\mathbf{q}}_p}$.

4 REDUCED-ORDER MODELS ENHANCEMENT *VIA* BASIS ADAPTATION

While the MP-LSVT method improves the robustness and accuracy of the ROM, predictive capabilities (e.g., future-state prediction) are still restricted by the use of linear static basis, which has been shown to be inadequate for predictions in problems with slow Kolmogorov N-width decay [14,44]. Several remedies have been proposed to address this challenge through, for example, localized linear bases [45,46], nonlinear bases [47,48], and online basis adaptation [49–51] etc. In the current work, we focus on online basis-adaption methods, which aim to update the trial basis $\mathbf{V}_p$ during the online ROM calculation (**Eq. 10**) such that

$$\mathbf{V}_p^n \triangleq \arg\min_{\mathbf{V}_p^n \in \mathbb{R}^{N\times n_p}} \left\| \mathbf{Pr}\left(\tilde{\mathbf{q}}_p{}^n\right) \right\|_2^2, \tag{12}$$

where **r** is the fully-discrete FOM equation residual defined in **Eqs. 2**, $\tilde{\mathbf{q}}_p{}^n = \mathbf{q}_{p,\text{ref}} + \mathbf{H}^{-1}\mathbf{V}_p^n\mathbf{q}_r^n$, and $\tilde{\mathbf{q}}_p{}^{n-j} = \mathbf{q}_{p,\text{ref}} + \mathbf{H}^{-1}\mathbf{V}_p^{n-j}\mathbf{q}_r^{n-j}$ while $\mathbf{q}_r^n$ and $\mathbf{q}_r^{n-j}$ are solutions to **Eq. 10**. This minimization problem can be solved exactly *via* the update

$$\mathbf{V}_p^n = \mathbf{V}_p^{n-1} + \delta\mathbf{V}_p, \tag{13}$$

where the basis at time-step $n-1$ is adapted to n, through an increment, $\delta\mathbf{V}_p \in \mathbb{R}^{N\times n_p}$, given by

$$\delta\mathbf{V}_p = \frac{\left(\hat{\mathbf{q}}_p{}^n - \tilde{\mathbf{q}}_p{}^n\right)\left(\mathbf{q}_r^n\right)^T}{\left\|\mathbf{q}_r^n\right\|_2^2}, \tag{14}$$

where $\hat{\mathbf{q}}_p{}^n \in \mathbb{R}^N$ represents the full-state information, which can be evaluated based on the FOM equation residual as follows

$$\begin{aligned} &\mathbf{q}\left(\hat{\mathbf{q}}_p{}^n\right) + \sum_{j=1}^{l}\alpha_j\mathbf{q}\left(\tilde{\mathbf{q}}_p{}^{n-j}\right) - \Delta t\beta_0\mathbf{f}\left(\tilde{\mathbf{q}}_p{}^n, t^n\right) - \Delta t\sum_{j=1}^{l}\beta_j\mathbf{f}\left(\tilde{\mathbf{q}}_p{}^{n-j}, t^{n-j}\right) = 0, \\ \text{or } &\mathbf{q}\left(\hat{\mathbf{q}}_p{}^n\right) + \sum_{j=1}^{l}\alpha_j\mathbf{q}\left(\tilde{\mathbf{q}}_p{}^{n-j}\right) - \Delta t\beta_0\mathbf{f}\left(\hat{\mathbf{q}}_p{}^n, t^n\right) - \Delta t\sum_{j=1}^{l}\beta_j\mathbf{f}\left(\tilde{\mathbf{q}}_p{}^{n-j}.t^{n-j}\right) = 0. \end{aligned} \tag{15}$$

here, we adopt an alternate formulation compared to [50] by updating the basis based on the full-state information evaluated at the current time step, n, $\hat{\mathbf{q}}_p{}^n$, instead of collecting at multiple time steps, which is similar to the work done by Zimmermann et al. [51]. We refer to this formulation as the one-step adaptive-basis approach. The rate of basis adaptation is empirically determined for the target applications and in the current work, we choose to adapt the basis at each time step.

To achieve gains in computational efficiency for the projection-based ROMs introduced in **Section 3**, hyper-reduction is required to obtain an approximation of the non-linear function (e.g., **f** in **Eq. 1**) based on a small number of sampled elements—for example, it can be achieved by the discrete empirical interpolation method (DEIM) [52], or its least-squares regression analogue, gappy POD Everson and Sirovich [53]. In addition, the full-state information evaluation (**Eq. 15**) in basis adaptation can be computationally expensive and also requires hyper-reduction for efficiency gain so that the evaluation is only needed at a small number of sampled elements. This can be achieved by incorporating the recently developed adaptive sampling techniques [50], which update the selection of sampled elements based on the basis adaptation. However, the current work mainly focuses on the development and demonstration of the component-based ROM framework. Therefore for conciseness, we are not including ROM hyper-reduction in the current paper since in principle, its presence or absence will not impact the validity of the

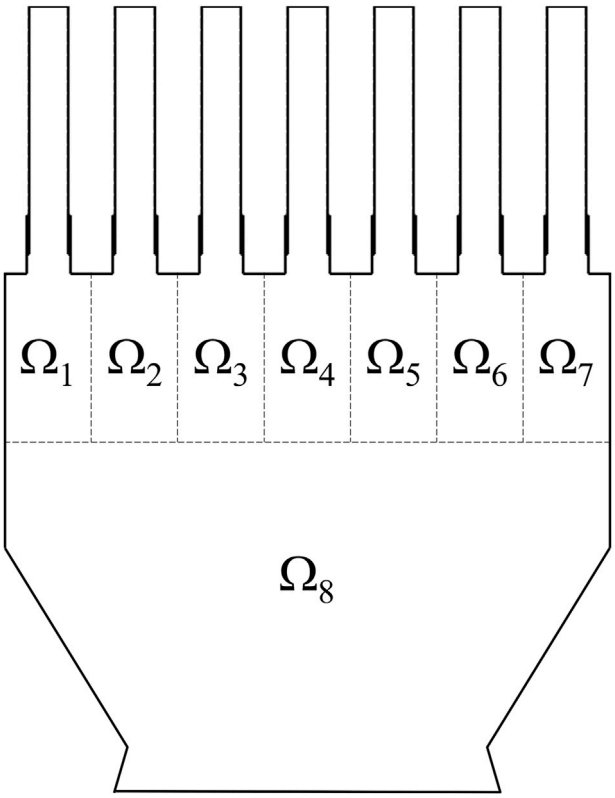


FIGURE 1 | An example of a multi-injector model rocket combustor with seven identical injector elements and the downstream combustor and nozzle [54].

framework. We will incorporate the hyper-reduced ROMs in the framework in future work.

5 COMPONENT-BASED DOMAIN-DECOMPOSITION FRAMEWORK

In this section, we introduce the component-based domain-decomposition framework for modeling large-scale engineering systems. Because our research has been primarily motivated by applications to propulsion systems for aerospace applications, we use a multi-injector model rocket combustor to assist in the description of the framework formulation. **Figure 1** presents a representative geometry composed of seven injector elements through each of which fuel and oxidizer in separate channel feed a downstream combustion chamber. The physical domain has been separated into eight components with seven for the injector elements (Ω_k, etc.) and one for the downstream combustor and nozzle (Ω_8). A set of similar configurations are also included in our numerical examples in **Section 6.2**. Even though the configuration in **Figure 1** represents a complicated engineering system, most of the components share identical geometric features, an attribute that is common in many engineering systems (e.g., compressors, gas turbine engines, and wind farms, etc.). The interior components Ω_k, where $k = 2, \ldots, 6$, are identical to each other geometrically, the two outer components Ω_1 and Ω_7 mirror each other in geometrical configuration (i.e., symmetric about the center axis), while Ω_8 does not resemble any other components. Therefore, the representation of the multi-injector rocket combustor in **Figure 1** can be simplified as the combination of three representative components: I ($\bar{\Omega}_{\mathrm{I}}$)—an interior injector element (e.g., Ω_3); II ($\bar{\Omega}_{\mathrm{II}}$)—an outer injector element (e.g., Ω_7); and III—the downstream combustor and nozzle (Ω_8), which enables the applications of the component-based domain-decomposition framework without accessing the expensive high-fidelity FOM of the full system. Furthermore, we denote the representative components that can be repetitively used in the full system (e.g., $\bar{\Omega}_{\mathrm{I}}$ and $\bar{\Omega}_{\mathrm{II}}$) as reference components. Based on *a priori* knowledge of the dynamics in the system [54,55], different modeling strategies can be adopted for different components (or subdomains) Ω_k, each of which is bounded by the physical boundaries, $\partial\Omega_k$ (e.g., inlets, outlets, walls surrounding Ω_k, represented by the solid lines in **Figure 1**), and interface boundaries (dashed lines in **Figure 1**) shared between components (e.g., k and m), $\partial\Omega_{km} \triangleq \Omega_k \cup \Omega_m$.

First, we introduce a general description of the domain-decomposition formulation

$$\begin{aligned} &\mathbf{B}_k\mathbf{r}\left(\bar{\mathbf{q}}_{p,k}^n\right) = \mathbf{0}, \\ &\text{with } \mathbf{u}_k\left(\bar{\mathbf{q}}_{p,k}^n\right) = \mathbf{u}_{BC,k}^n \text{ on } \partial\Omega_k, \\ &\text{and } \mathbf{v}_{km}\left(\bar{\mathbf{q}}_{p,k}^n\right) = \mathbf{v}_{km}\left(\bar{\mathbf{q}}_{p,m}^n\right) \text{ on } \partial\Omega_{km}, \end{aligned} \tag{16}$$

where k denotes the numbering of the sub-components in the formulation, $\mathbf{B}_k \in \mathbb{R}^{n_{B,k}\times N_k}$ denotes a matrix that enables the component to adopt either FOM or ROM for the corresponding kth subdomain (Ω_k), and N_k is the total number of degrees of freedom in Ω_k and $N_k = N_{elem,k} \times N_{var}$, where $N_{elem,k}$ is the total number of elements in Ω_k. For FOM, $n_{B,k} = N_k$, and $\mathbf{B}_k = \mathbf{I}$, similar to **Eq. 2**, and for ROM, $n_{B,k} = n_p$, and $\mathbf{B}_k = (\mathbf{W}_{p,k}{}^n)^T\mathbf{P}_k$, consistent with **Eq. 10**. $\mathbf{u}_k$: $\mathbb{R}^{N_k} \to \mathbb{R}^{N_{b,k}}$ represents the physical boundary condition state to be satisfied as $\mathbf{u}_{BC,k}$ on $\partial\Omega_k$, where $N_{b,k}$ is the total number of degrees of freedom associated with the boundary $\partial\Omega_k$, and $N_{b,k} = N_{elem,BC,k} \times N_{var}$, with $N_{elem,BC,k}$ as the number of elements adjacent to the boundary $\partial\Omega_k$. In addition, a (non-linear) function, $\mathbf{v}_{km}$: $\mathbb{R}^{N_k} \to \mathbb{R}^{N_{Interf,k}}$, is introduced to match the interface condition between subdomains Ω_k and Ω_m, where $N_{Interf,k}$ is the total number of degrees of freedom associated with the boundary $\partial\Omega_k$, and $N_{Interf,k} = N_{elem,Interf,k} \times N_{var}$, with $N_{elem,Interf,k}$ as the number of elements adjacent to the interface boundary $\partial\Omega_{km}$. Additional details are provided in **Section 5.2**.

Given the complexity and scale of the physics, small-scale local components with identical features (e.g., the interior and outer injector elements, Ω_3 and Ω_7, in **Figure 1**, the nozzle element in a gas turbine, the rotor blade in a compressor, or the wind turbine in a wind farm) often require high-fidelity modeling to achieve satisfying accuracy in many-query engineering applications. Therefore, ROMs can be an ideal candidate from the viewpoint of satisfying efficiency and accuracy requirements. On the other hand, large-scale system-level components (e.g., the downstream combustor and nozzle, Ω_8, in **Figure 1**) are

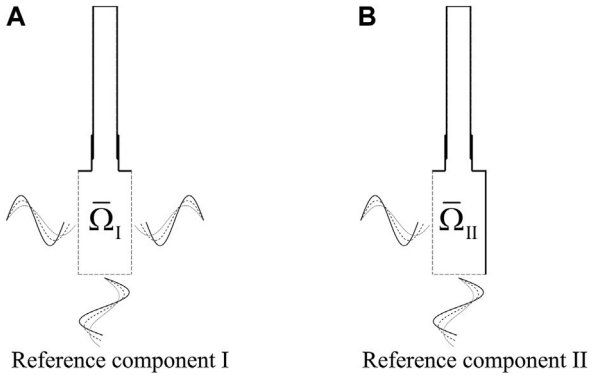


FIGURE 2 | Schematic of the component-based ROM training method based on **(A)** reference components I and **(B)** II.

usually governed by physics that is less demanding in numerical resolution. This makes the reduced-fidelity full-order model (RF-FOM) a good candidate for modeling of the large-scale system-level components—for example, coarse-mesh LES, nonlinear Euler model, unsteady Reynolds Averaged Navier Stokes models.

5.1 Component-Based Reduced-Order Model Training

To enable modeling of the full system (e.g., **Figure 1**), the FOM of which is not directly accessible, we develop a component-based ROM training method in the current section, that requires high-fidelity FOM simulations for only the reference components identified in the system. The method aims to generate a rich training dataset that contains representative dynamics of the components when integrated in the full system of various configurations (e.g., different numbers of injector elements), thus enabling the generation of predictive component-based ROMs, which is analogous to the localized ROM strategy developed for finite element method with representative work by Henning and Peterseim [56]; Eftang and Patera [28]; and Smetana and Patera [29]. To achieve this, we introduce unsteady perturbations at interface boundaries in the FOM simulations of the reference components, following **Eq. 1**, to fabricate the effects of system-level responses, as demonstrated in **Figure 2**. Most importantly, by enriching the functions used for the boundary perturbations, the dynamics of different full-system configurations can be embedded within the ROMs of the individual components. For example, the effects of system-level acoustics can be accounted for by imposing different pressure perturbations at the boundary

$$p(t) = p_{ref}\left[1 + \sum_{i=1}^{n_f} A_i \sin\left(2\pi f_i t\right)\right]. \tag{17}$$

where f_i is the frequency included in the boundary perturbations with A_i denoting the associated amplitude, reflecting the anticipated full-system responses, and n_f is the total number of frequencies, an indicator of the richness of the excited dynamics. Alternatively, velocity oscillations can be enforced at the boundaries following similar function in **Eq. 17** to mimic the effects of large-scale flow dynamics in the full system.

It is important to ensure that the imposed boundary conditions (dashed lines in **Figure 2**) do not imprint the dimensions of the individual component upon the combustion dynamics. All internal domains are subject to acoustic resonance at scales determined by their geometry, but the dimensions in the individual component are not representative of those of the full system and so must not appear in the training dataset. Accordingly, it is critical that the boundary conditions on the reference component be chosen such that its geometry does not impact the FOM solutions of the dynamics upon which the ROM is based. An effective way to accomplish this is by applying non-reflective boundary conditions through pertinent Riemann invariants. We adopt the formulation that proved to be effective for multi-dimensional reacting flow simulations [57]. It should be mentioned that the ROM training method in **Figure 2** requires a level of prior knowledge of the essential physics in the full-system (e.g., acoustics) to ensure that pertinent physics are included in the ROM training. For example, in rocket combustor design such as **Figure 1**, system-level acoustic frequencies can be estimated *a priori* based on the full-system configuration. Correspondingly, a multi-frequency perturbation can be imposed at the boundary conditions for ROM training, the frequency band of which covers the target full-system acoustic frequencies and therefore excites the essential dynamics anticipated within the component when integrated in the full-system. A similar idea has been demonstrated in simple 1D problems by the current authors Huang et al. [40]; Xu et al. [41]. To account for complex dynamics at the component interfaces, instead of directly imposing the perturbations at the interfaces, auxiliary domains can be introduced for the ROM training, which is discussed and demonstrated in **Section 6**.

5.2 Integration in Full System Simulations

Once the component-based models are constructed following the strategies in **Section 5.1**, the effective integration of these models in the domain-decomposition framework is another determining factor for the success of the framework. In this section, we use the example in **Figure 1** to illustrate the integration of the component-based models. Following the method in **Section 5.1**, ROMs are trained on the reference components, the trained ROMs can be used repetitively to model identical components in the full system to enable geometric variations (**Figure 3A** with $\hat{(\cdot)}$ indicating that the left element mirrors the right one) based on the premise that the injector elements in the full system share identical geometries as the reference components used for ROM training. As discussed above, the RF-FOM can be adopted to efficiently model the geometrically flexible components (e.g., the downstream combustor and nozzle) that vary with the full system configurations, given the less demanding requirements on modeling accuracy. Alternatively, ROMs could also be developed for the geometrically flexible components although ROM developments for geometric variations remains an open area of research.

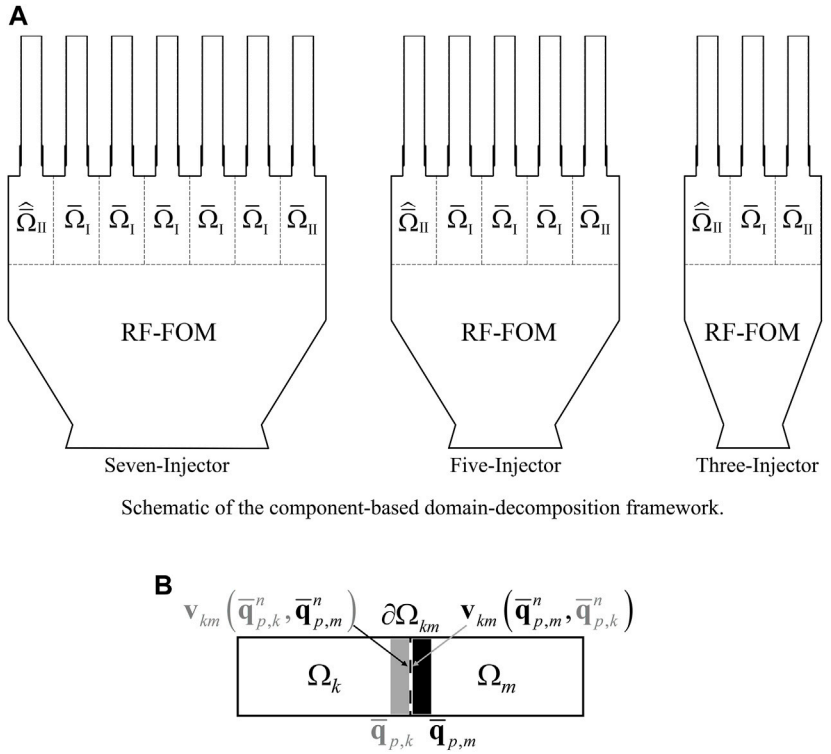


FIGURE 3 | Illustration of the component-based domain-decomposition framework. **(A)** Schematic of the component-based domain-decomposition framework. **(B)** Interfacing method for coupling different components.

To achieve accurate predictions of the system-level response governed by the component interactions, it is important to ensure that the essential information is transferred between components. We adopt a direct flux matching method *via* ghost cell assignment to couple the components at the interface with no overlap

$$\mathbf{v}_{km}\left(\bar{\mathbf{q}}^n_{p,k}, \bar{\mathbf{q}}^n_{p,m}\right) = \mathbf{v}_{km}\left(\bar{\mathbf{q}}^n_{p,m}, \bar{\mathbf{q}}^n_{p,k}\right) \quad \text{on} \quad \partial\Omega_{km}, \tag{18}$$

a schematic of which is shown in **Figure 3B**, describing the coupling between domain Ω_k and Ω_m through the interface $\partial\Omega_{km}$. The two adjacent mesh cells of $\partial\Omega_{km}$ are indicated by the shaded areas. When performing calculations based on **Eq. 16** at time step n at a cell with solution variables $\bar{\mathbf{q}}_{p,k}$ near $\partial\Omega_{km}$ in Ω_k, solution variables at the adjacent (or ghost) cell, denoted as $\bar{\mathbf{q}}_{p,m}$, are assigned by the corresponding neighboring domain Ω_m. The combination of $\bar{\mathbf{q}}_{p,k}$ and $\bar{\mathbf{q}}_{p,m}$ is then used to calculate the interface condition state, denoted as $\mathbf{v}_{km}(\bar{\mathbf{q}}_{p,k}{}^n, \bar{\mathbf{q}}_{p,m}{}^n)$, and vice versa, thus guaranteeing the interface condition state is matched as posed in **Eq. 16**—i.e., $\mathbf{v}_{km}(\bar{\mathbf{q}}_{p,k}{}^n) = \mathbf{v}_{km}(\bar{\mathbf{q}}_{p,m}{}^n)$. Specifically, we set the interface condition state function $\mathbf{v}_{km}$ to be the numerical fluxes (both inviscid and viscous) to better suit the finite volume scheme of the numerical solver used for the current work [58]. We remark that the interface method remains the same regardless of whether the domain is represented by FOM or ROM.

The major benefit of the direct-flux-matching interfacing method is that it inherently accounts for changes in flow characteristics at the interface and therefore important phenomena such as reverse flows are naturally supported. More importantly it makes the training of the component-based ROMs relatively independent of their coupling with other components in the framework, which allows more flexibility in the ROM training strategy. For example, auxiliary domains (e.g., adjacent injector elements) can be introduced in the component-based ROM training stage in **Figure 2** to better emulate interactions between injector elements in the training dataset. These aspects are demonstrated in the numerical results in **Section 6**.

6 NUMERICAL RESULTS

To assess the capabilities of component-based domain-decomposition modeling framework in predicting multi-scale multi-physics problems (e.g., reacting flows), two model rocket combustors are considered. The first configuration is a two-dimensional representation of a generic laboratory-scale single injector configuration [59]. This case is used to a) assess the component-based ROM training strategy and the interfacing method between components in the framework, and b) to explore the feasibility of using the framework to predict dynamics on different full-system geometries. The second configuration is a two-dimensional representation of a

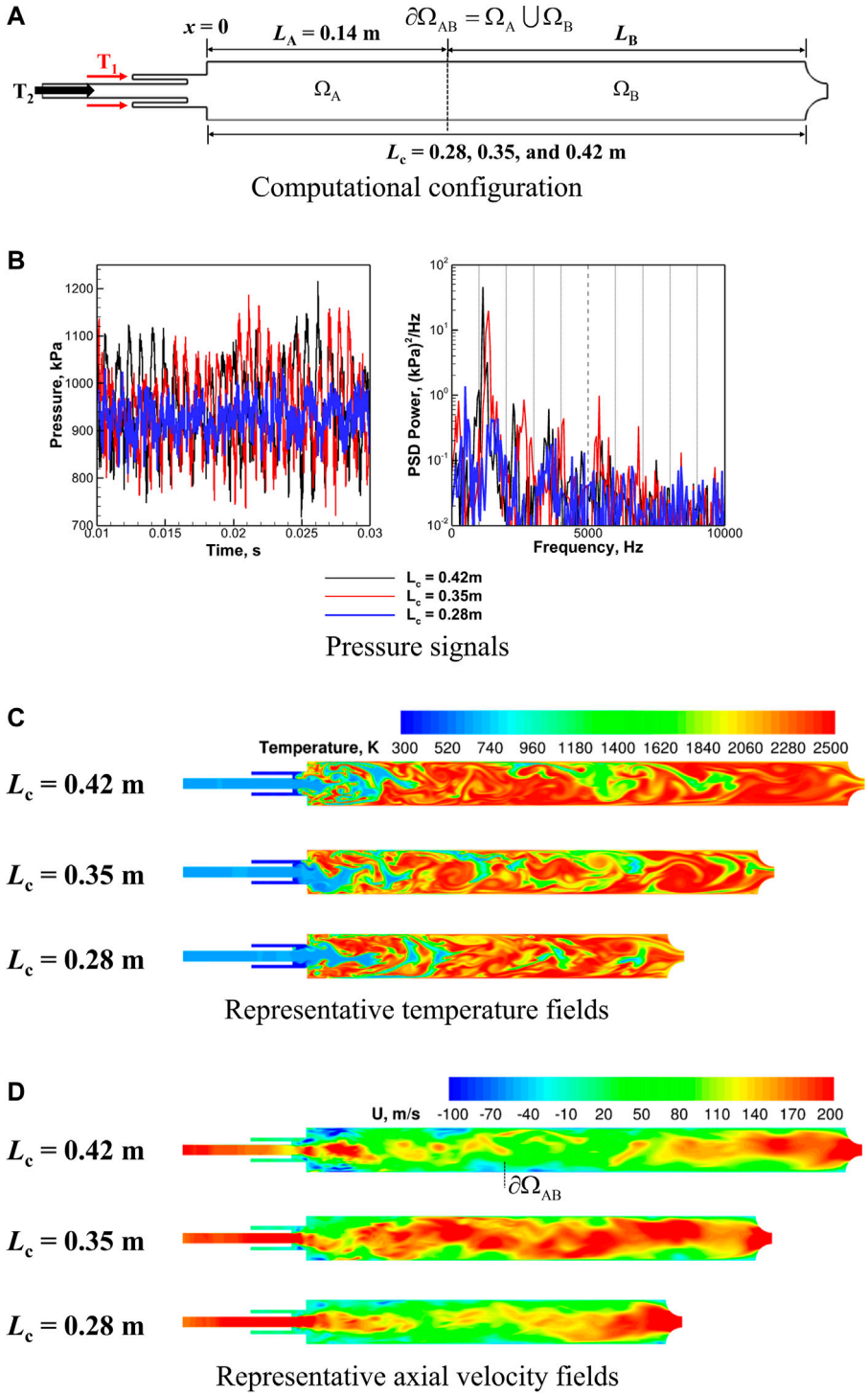


FIGURE 4 | Overview of 2D-planar single-injector model rocket combustor including **(A)** computational configuration and FOM-FOM results with **(B)** pressure signals measured at $x = 0$ in **(A,C)** representative snapshots of the temperature fields, and **(D)** representative snapshots of the axial velocity fields.

multi-injector model rocket combustor [60], and is used to evaluate the capabilities of the framework to model different geometric configurations of the full system as demonstrated in **Figure 3**.

The computational infrastructure used for the full- and reduced-order models solves conservation equations for mass, momentum, energy and species transport [58,61] in a fully coupled manner, which has been used to model a variety of

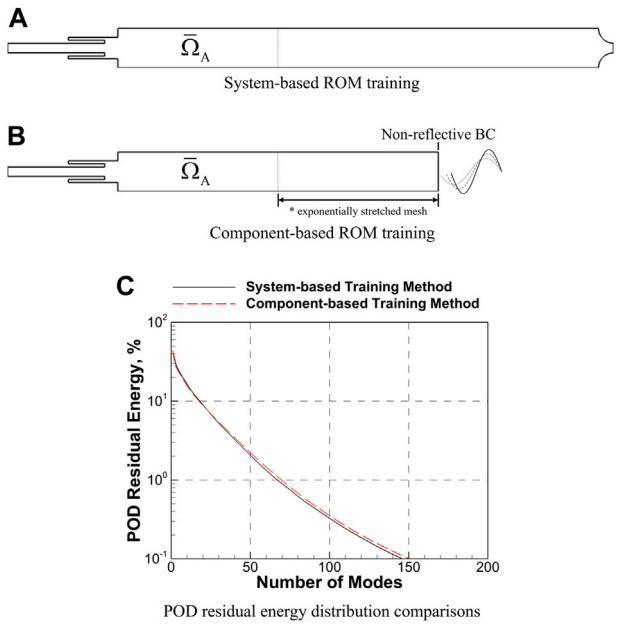


FIGURE 5 | ROM training strategies for the single-injector combustor, including **(A)** system-based and **(B)** component-based methods and **(C)** POD residual energy distribution comparisons the two training methods.

complex, practical reacting flow problems. More details of the FOM equations can be found in Appendix B of [14]. The FOM employs a cell-centered second-order accurate finite volume method for spatial discretization and uses the direct flux matching method as described in **Section 5.2** for parallel computation. The Roe scheme [62] is used to evaluate the inviscid fluxes and a Green-Gauss gradient reconstruction procedure [63] is used to compute the face gradients and viscous fluxes. A gradient limiter by Barth and Jespersen [64] is used to preserve monotonicity for flow fields with strong gradients. A ghost cell formulation is used for treatment of boundary conditions. Time integration for all FOM simulations uses the implicit second-order accurate backwards differentiation formula with dual time-stepping.

6.1 Single-Injector Model Rocket Combustor

First, we explore and demonstrate the component-based domain-decomposition framework on a 2D-planar representation of a generic laboratory-scale rocket combustor designed to study combustion dynamics [59]. The configuration is shown in **Figure 4A** and consists of a shear coaxial injector with an outer passage, T_1, that introduces fuel near the downstream end of the coaxial inner passage, T_2, which in turn feeds oxidizer to the combustion chamber with a choked nozzle downstream, resulting in combustion-driven acoustics to be sustained. Operating conditions in this single-injector combustor are maintained with an adiabatic flame temperature of approximately 2,700 K and a mean chamber pressure of 0.95 MPa. The T_1 stream contains gaseous methane (100% CH_4) at 300 K. The T_2 stream is 42% gaseous O_2 by mass and 58% gaseous H_2O by mass at 660 K. The T_1 and T_2 streams are maintained at constant mass flow rates, 0.46 kg/s and 5.40 kg/s, respectively. Combustion is represented by the flamelet progress variable (FPV) model [65] with GRI-1.2 [66] chemical kinetics, which consists of 32 species and 177 chemical reactions. The chemical species are treated as thermally perfect gases. Note that although 32 chemical species are modeled, the FPV model only solves transport equations for three scalar quantities: the mean mixture fraction (Z_{mean}), the mixture fraction variance (Z''^2), and the reaction progress variable (C_{mean}) [65]. Individual chemical species mass fractions are looked up from pre-computed flamelet manifolds [67].

As shown in **Figure 4A**, the single-injector configuration consists of two components, the upstream injector element (Ω_A) and the downstream combustor and nozzle (Ω_B). Three different lengths of the combustor (L_c) are investigated for this configuration by maintaining the upstream component (Ω_A) while varying the length of the downstream component (Ω_B), i.e., L_B. This change in length leads to different dynamic behaviors as shown in **Figures 4B–D** when both Ω_A and Ω_B adopt FOM as their modeling strategy, denoted as FOM-FOM, the solutions from which are taken as the *truth* to evaluate the component-based ROM framework. It can be readily seen from **Figure 4B** that by just varying the combustor length (L_c), different pressure oscillations can be sustained. The longer combustor lengths tend to drive higher pressure oscillation amplitudes (> 30% peak-to-peak) while the shorter length maintains lower amplitude, < 15%, (**Figure 4B** left). This features different frequencies as shown in the power spectral densities (PSD) in **Figure 4B** right. In addition, the combustion dynamics changes with L_c as seen in **Figures 4C,D**, which allows reverse flow at the component interface $\partial\Omega_{AB}$ under high-amplitude pressure oscillations. Thus, this single-injector combustor is an appropriate testbed for the component-based domain-decomposition framework proposed in **Section 5**, which adopts a ROM for the upstream component (Ω_A), and a RF-FOM for the downstream component (Ω_B).

6.1.1 Injector Element Reduced-Order Models Training

As mentioned in **Section 5.1**, the ROM training strategy is crucial to the success of the component-based framework. For the investigations on the single-injector configuration (**Figure 4**), two types of strategies are considered to train ROM for the upstream component (Ω_A): 1) system-based (only for framework verification), and 2) component-based methods as shown in **Figure 5**.

- The system-based approach (**Figure 5A**) simulates the complete geometry of interest (e.g., L_c = 0.42 m in **Figure 4**), trains the ROM based on the extracted snapshot solutions corresponding to the upstream injector-element component ($\bar{\Omega}_A$), and only serves to verify the feasibility of the component-based ROM framework.
- The component-based approach (**Figure 5B**), as the primary focus of the current work, simulates only the

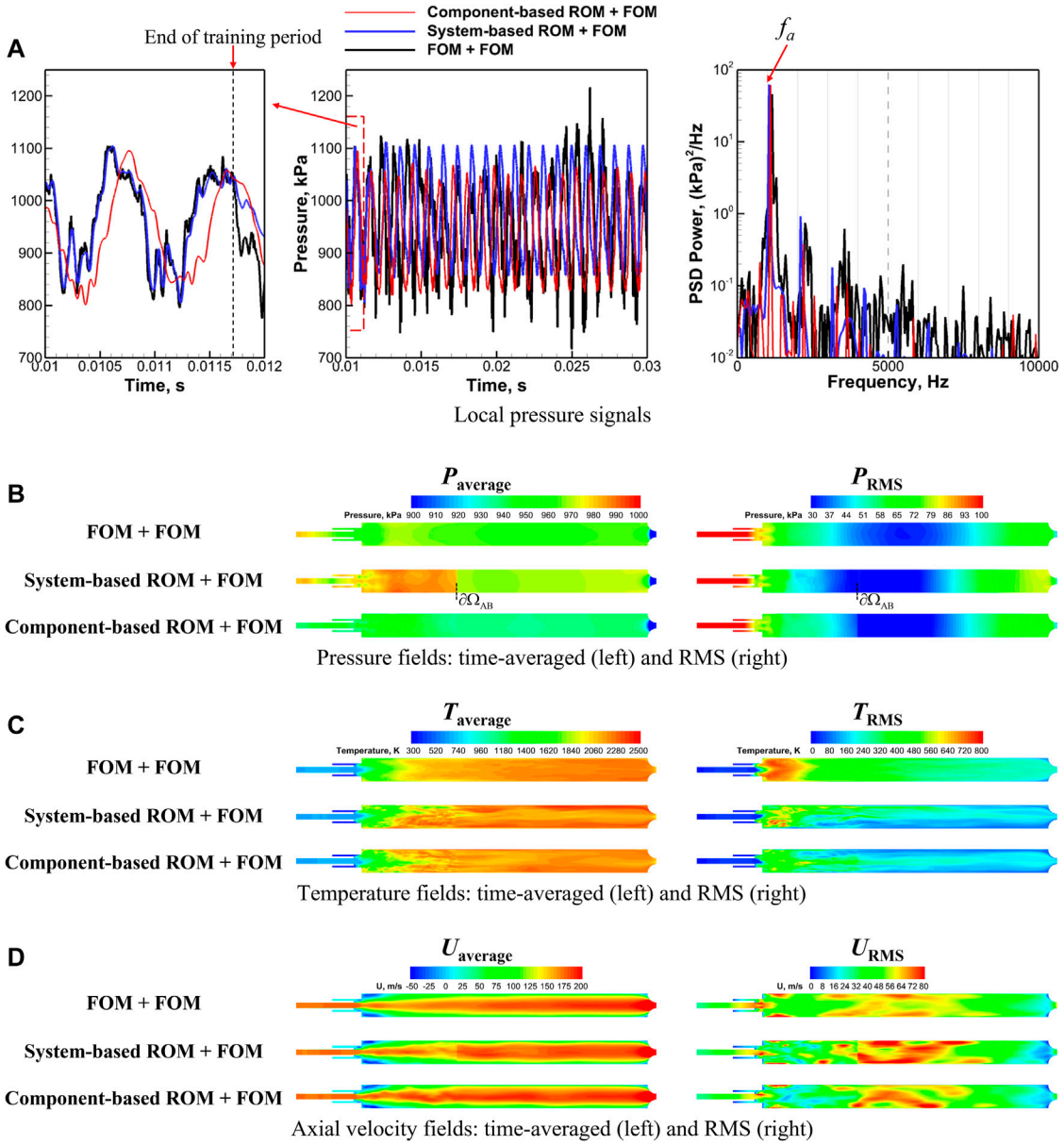


FIGURE 6 | Comparisons of **(A)** pressure signals measured at $x = 0$ in **(B)** pressure, **(C)** temperature, and **(D)** axial velocity time-averaged (left) and root-mean-square (right) fields with different models (component-based ROM vs. system-based ROM vs. FOM) used for the upstream injector element (Ω_A) for the single-injector combustor.

injector-element for ROM training and produces essential dynamics by forcing the downstream boundary conditions. Further, component-based training is necessary for systems whose full-scale characteristics are beyond the current available computing capabilities. Instead of imposing the forcing directly at the downstream end of the injector element (i.e., the dashed line in **Figure 5B**), an auxiliary domain with exponentially stretched mesh in the axial direction is added downstream for the component-based ROM training approach. The addition of the auxiliary domain is necessary to represent large-scale motions (e.g., vortex shedding) that are undamped in the downstream domain whereas small scale motions (e.g., chemical reaction) that would interfere with long-wavelength forcing are damped before the downstream boundary is reached. It inherently incorporates complex dynamics (e.g., reverse flow) at the component interface $\partial\Omega_{AB}$ (e.g., similar to what has been observed in **Figures 4B,C** in the training snapshots for ROM, which cannot be accounted if boundary conditions are directly applied. In addition, it reduces the influence of the training geometry (i.e., injector element + auxiliary domain) on the dynamics

in the training dataset for ROM, especially on the longitudinal acoustics.

In either of the above approaches, the trained ROM is coupled with different downstream components to model the full system. The component-based approach provides flexibility in generating dynamics and substantial savings in computational cost for ROM training since it eliminates the need for computation of a complete configuration in **Figure 4**.

We consider L_c = 0.42 m in **Figure 4** as the target full system, which exhibits a self-excited high-amplitude pressure oscillations at 1150 Hz as shown in **Figure 4A**. Both system-based and component-based methods are used to generate training snapshots for ROM development. To generate the essential dynamics during component-based training (**Figure 5B**), the Riemann invariant corresponding to backward characteristics (q_{u-c}) is perturbed at the downstream boundary using the following forcing function:

$$q_{u-c} = q_{u-c,ref}\left[1 + A\sin\left(2\pi f t\right)\right], \tag{19}$$

where $A = 0.1, f = 1150$Hz to generate similar pressure oscillations observed in the full system, and $q_{u-c,ref}$ represents the reference value of the Riemann variable of backward characteristics that maintains the nominal pressure. The training snapshots containing two acoustic cycles of information (i.e., $T_p = 2/f$) are used to generate the POD trial basis as described in **Section 3.1**, the characteristics of which are investigated to understand how well the POD trial basis represents the training dataset. The representation is evaluated using the POD residual energy:

$$\text{POD Residual Energy}\left(n_p\right),\ \% = \left(1 - \frac{\sum_{i=1}^{n_p}\tilde{\sigma}_i^2}{\sum_{i=1}^{n_{p,total}}\tilde{\sigma}_i^2}\right) \times 100, \tag{20}$$

where $\tilde{\sigma}_i$ is the ith singular value of the SVD used to compute the trial basis $\mathbf{V}_p$. Again, n_p is the number of vectors retained in the POD trial basis, and $n_{p,total}$ (= 1740) is the total number of snapshots in the dataset. The residual energy as a function of n_p, as shown in **Figure 5C** for both system-based and component-based training methods, reveals the information excluded by the POD representation for a given number of modes. Overall, the two ROM training methods show very similar POD residual energy decay. The results show that, to recover approximately 99% of the total energy, 70 and 68 modes are needed for system-based and component-based methods, respectively, while approximately 150 modes are required to reach 99.9% for both methods. This slow energy decay is indicative of the significant complexity of the system dynamics. Many fundamental projection-based ROM methodologies are tested on relatively simple problems requiring only ~ 10 trial basis modes to achieve 99.9% POD energy [68–70]. ROMs for more practical engineering systems, however, generally require ~ 100 trial basis modes [43,71,72].

6.1.2 Performance

Next, we couple the trained injector-element ROMs from **Section 6.1.1** with the downstream combustor and nozzle (Ω_B in **Figure 4**) *via* the interfacing method described in **Section 5.2** to model the full configuration of L_c = 0.42 m. Two acoustic cycles (1.74 ms) of snapshots (1,740 in total) are used to train the ROMs, which are constructed with the number of POD modes capturing 99% of the total energy. To consistently evaluate the modeling capabilities of the resulting framework based on the FOM-FOM results (i.e., the true solutions) in **Figure 4**, we adopt a FOM, instead of RF-FOM, for the downstream component (Ω_B). The coupled ROM-FOM framework is then used to predict 20 ms of dynamics and compared against the FOM-FOM results.

First, we evaluate the performance based on the local pressure signals measured at $x = 0$ in **Figure 4**, which has been often used as an important quantity of interest (QoI) to assess the accuracy of modeling tools in predicting combustion instability [58]. The predicted pressure signals, both time traces (left and middle) and PSDs (right), are compared in **Figure 6A** with different models used for the used for the upstream injector element (Ω_A) for the single-injector combustor. Furthermore, the peak-to-peak pressure oscillation amplitude (p'_{ptp} based on the root-mean-square (RMS) in **Eq. 21**) and the dominant acoustic frequency (f_a) are calculated based on the pressure time trace and PSD, respectively and compared against the FOM-FOM results for quantitative assessment, as summarized in **Table 1**. Overall, the ROM-FOM framework (either with system-based or component-based ROM) is able to predict the pressure amplitude and frequency with reasonable accuracy ($< 10\%$). As shown in **Figure 4**, within the training period, the system-based ROM-FOM replicates the FOM-FOM results closely, as expected. On the other hand, the component-based ROM-FOM also represents the pressure oscillations reasonably well even though the essential downstream component is excluded in ROM training. Moreover, the comparisons in **Figure 6A** confirm that the component-based ROM training strategy is feasible by emulating feedback responses from the downstream component with boundary forcing to train ROM, as illustrated in **Figure 5**. More importantly, both approaches enable long-time predictions (e.g., 1.74 ms training vs. 20 ms prediction), which is not commonly reported in the literature for ROM applications relevant to compressible fluid flow problems.

Second, we assess the predictive capabilities of the ROM-FOM framework based on two other QoIs, time-averaged and root-mean-square (RMS) fields of the state variables, which serve as crucial determining factors in many engineering applications

$$\mathbf{\Phi}_{\text{average}} = \frac{1}{n_t}\sum_{n=1}^{n_t}\mathbf{\Phi}^n, \quad \mathbf{\Phi}_{\text{RMS}} = \sqrt{\frac{1}{n_t}\sum_{n=1}^{n_t}\left(\mathbf{\Phi}^n - \mathbf{\Phi}_{\text{average}}\right)^2}, \tag{21}$$

where n_t is the total number of snapshots included to calculate the QoI, and $\mathbf{\Phi}^n$ represents the state variable of interest, e.g., pressure (P), temperature (T), and axial (or streamwise) velocity (U), at time step n. In addition, the errors of the ROM-FOM framework in predicting $\mathbf{\Phi}_{\text{average}}$ and $\mathbf{\Phi}_{\text{RMS}}$ are further quantified as follows

$$\epsilon_{\mathbf{\Phi}} = \frac{\|\mathbf{\Phi} - \mathbf{\Phi}_{\text{ref}}\|_2}{\|\mathbf{\Phi}_{\text{ref}}\|_2}, \tag{22}$$

where $\mathbf{\Phi}$ represents the QoIs (i.e., either the time-averaged or RMS field) for the error measurement, and $\mathbf{\Phi}_{\text{ref}}$ represents the QoIs calculated from the FOM-FOM framework. The errors, calculated based on **Eq. 22**, are summarized in **Table 2**. Though the ROM-FOM framework is able to provide reasonably accurate

TABLE 1 | Comparisons of the dominant acoustic frequency (f_a) and the peak-to-peak pressure amplitudes (p'_{ptp}) with different models (component-based ROM vs. system-based ROM vs. FOM) used for the upstream injector element for the single-injector combustor, corresponding to the results in **Figure 6A**.

Model for Ω_A	f_a, Hz	Error in f_a, %	p_{ptp}', kPa	Error in p_{ptp}', %
FOM	1,150	—	125.08	—
System-based ROM	1,050	8.70	116.25	7.06
Component-based ROM	1,100	4.35	112.93	9.71

TABLE 2 | Comparisons of the errors in predicting time-averaged and RMS fields of pressure (P), temperature(T), and axial velocity (U) corresponding to **Figures 6B-D** with different models (component-based ROM vs. system-based ROM) used for the upstream injector element for the single-injector combustor.

QoI	Model for Ω_A	ϵ_P	ϵ_T	ϵ_U
Time-averaged Fields	System-based ROM	1.28×10^{-2}	5.93×10^{-2}	6.06×10^{-2}
	Component-based ROM	2.63×10^{-2}	5.58×10^{-2}	7.72×10^{-2}
RMS Fields	System-based ROM	1.6×10^{-1}	4.03×10^{-1}	2.32×10^{-1}
	Component-based ROM	9.80×10^{-2}	4.63×10^{-1}	2.50×10^{-1}

predictions of the time-averaged fields ($< 3\%$ errors for P, and $< 8\%$ for T and U) with both system-based, and component-based ROM used for Ω_A, significant errors ($> 9\%$ for P, $> 20\%$ for U, and $> 40\%$ for T) are observed in predicting the RMS fields, which are directly related to the unsteady dynamics. Specifically, while the predictions of pressure RMS fields are acceptably accurate, the ROM-FOM framework exhibits difficulties in representing the RMS fields of state variables featuring strong advection (i.e., T and U), which can be largely attributed to the chaotic nature of the dynamics as seen in **Figures 4B,C**. The pressure field exhibits organized dynamics due to strong self-excited oscillations in the full system, which allows the ROM-FOM to provide reasonable predictions as the trial basis generated during the training stage is able to efficiently represent such organized dynamics. However, in turbulent reacting flows (characterized by transport of strong temperature gradients), chaotic and non-stationary features present a major challenge. The basis is unable to represent the unsteady features of T and U in the upstream component Ω_A, therefore producing significant errors when coupled with the FOM for the downstream component Ω_B in the resulting framework. This is not a flaw in the ROM formulation or the domain-decomposition framework, but rather a limitation of using a linear and static basis set to construct the ROM for the upstream component in the framework, which has also been discussed by the current authors in [14].

Such challenges and the limitation of using a linear and static basis set are further revealed by comparing the time-averaged and RMS fields between the FOM-FOM and ROM-FOM framework in **Figures 6B–D**, which shows significanly under-predicted magnitudes of the RMS fields by the ROM-FOM framework. More importantly, distinguishable mismatches between solutions in Ω_A and Ω_B are observed at the component interface $\partial\Omega_{AB}$ in ROM-FOM results, featuring abrupt changes in numerical values in the regions adjacent to the interface (e.g., P_{average}, U_{average}, U_{RMS} of system-based ROM-FOM and P_{RMS} of component-based ROM-FOM), which is absent in FOM-FOM results. The mismatches can be mainly attributed to the inconsistent orders of modeling accuracy between ROM in Ω_A, restricted by the POD basis, and FOM in Ω_B, restricted by the mesh resolution. The use of a static linear basis for ROM in Ω_A limits its predictive capabilities. It is pointed out that the interface mismatches are not unique to ROM/FOM coupling, but more general issues for finite-element [73] and finite-volume [74] methods, especially with non-matching grids (i.e., inconsistent orders of modeling accuracy)—for example, the coupling of low- and high-order CFD solvers (FOMs) may require—for example—an overset-mesh approach [21,75]. Though such methods can be effective in coupling of low- and high-order FOMs, it is not clear whether such methods can be applied directly to ROM and FOM coupling since ROM evolves on a reduced dimensional trajectory determined by the basis, while the FOM (either with low- or high-order numerical methods) solves the dynamical system on the full state space trajectory.

6.1.3 Performance Enhancement *via* Adaptive-Basis Reduced-Order Models

To address the above challenges, we seek to improve the ROM modeling accuracy *via* the one-step adaptive-basis approach introduced in **Section 4**. During the offline stage, 10 snapshots from the component-based ROM training demonstrated in **Figure 5B** are used to generate the initial POD basis $\mathbf{V}_p^0$ 5 POD modes, containing $> 99.9\%$ of the total energy, are chosen to develop the ROM for the upstream injector element (Ω_A). Then the POD basis is adapted at each time step based on the algorithm in **Eq. 13**, which is then used to construct an updated ROM. It is noted that even though significant reduction in the offline training cost is enabled by the adaptive-basis approach, the additional costs required to evaluate the full-state information in **Eq. 15**, can lead to an increase of the online-stage computational cost. On-line cost savings can be accomplished using hyper-reduction [52,53,76,77], which is not considered for the current work and will be included for future investigations. In the current work, we denote the ROM enhanced with basis adaptation as adaptive-basis ROM while in

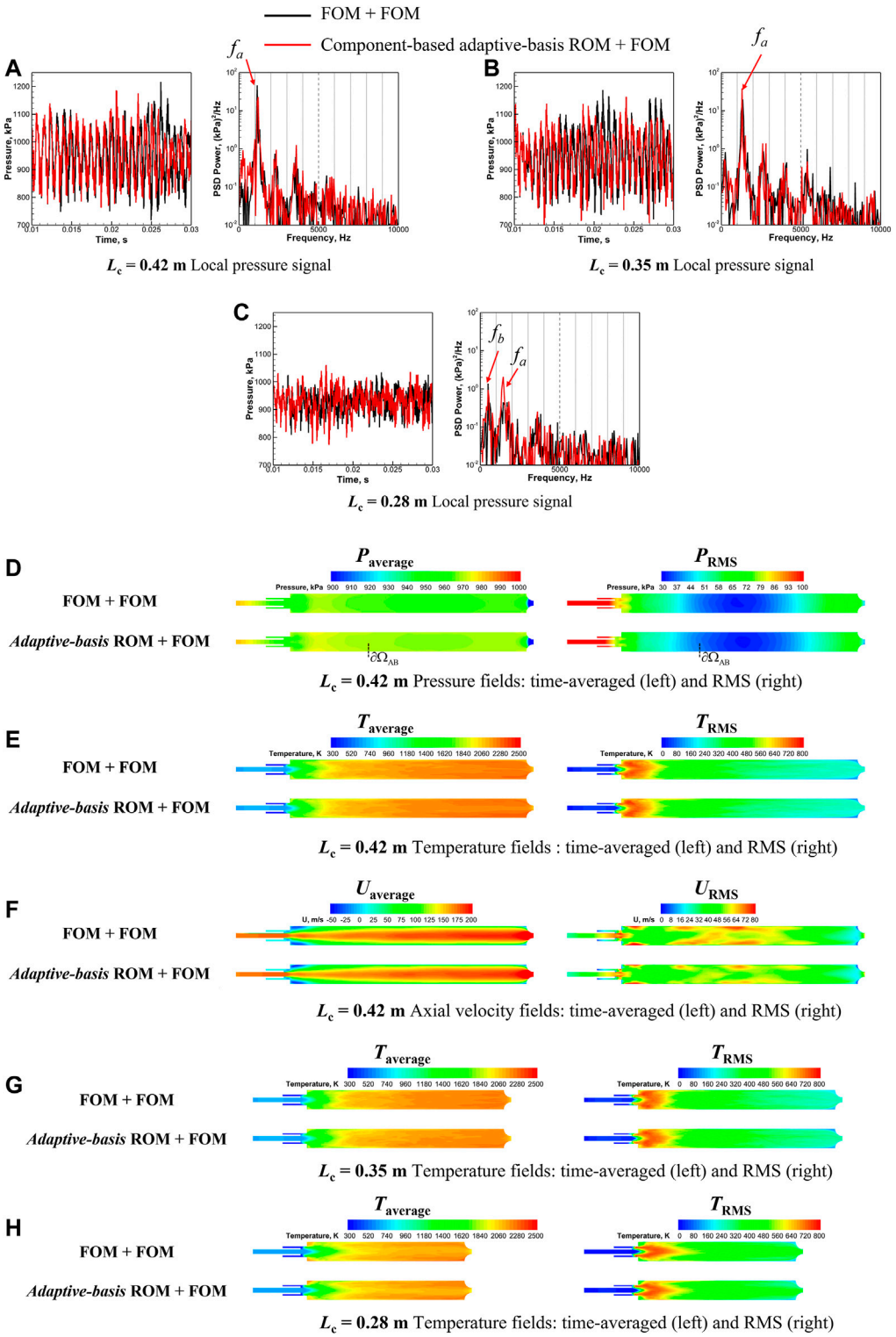


FIGURE 7 | Comparisons of **(A–C)** pressure signals measured at $x = 0$ in **Figure 4**, time-averaged (left) and root-mean-square (right) **(D)** pressure, **(E)** temperature, and **(F)** axial velocity fields for L_c = 0.42 m, and time-averaged (left) and root-mean-square (right) temperature fields for L_c = **(G)** 0.35 and **(H)** 0.28 m with component-based adaptive-basis ROM and FOM used for the upstream injector element (Ω_A) for the single-injector combustor.

TABLE 3 | Comparisons of the dominant acoustic frequency (f) and the peak-to-peak pressure amplitudes (p'_{ptp}) with different models (Adaptive-basis ROM vs. FOM) used for the upstream injector element for the single-injector combustor with different combustor lengths (L_c), corresponding to the results in **Figures 7A-C**.

L_c, m	Model for Ω_A	f, Hz	Error in f, %	p'_{ptp}, kPa	Error in p'_{ptp}, %
0.42	FOM	f_a: 1,150	—	125.08	—
	Adaptive-basis ROM	f_a: 1,200	4.35	123	1.67
0.35	FOM	f_a: 1,350	—	128.28	—
	Adaptive-basis ROM	f_a: 1,300	3.70	114.95	10.39
0.35	FOM	f_b: 500, f_a: 1,500	—	67.87	—
	Adaptive-basis ROM	f_b: 450, f_a: 1,400	f_b: 4.35, f_a: 6.70	59.49	12.34

TABLE 4 | Comparisons of the errors in predicting time-averaged and RMS fields of pressure (P), temperature(T), and axial velocity (U) using the framework corresponding to **Figures 7D-H** with adaptive-basis ROMs for the single-injector combustor with different combustor lengths (L_c).

QoI	L_c, m	ϵ_P	ϵ_T	ϵ_U
Time-averaged Fields	0.42	4.23×10^{-3}	1.95×10^{-2}	2.10×10^{-2}
	0.35	6.70×10^{-3}	1.94×10^{-2}	2.17×10^{-2}
	0.28	2.20×10^{-3}	2.50×10^{-2}	1.68×10^{-2}
RMS Fields	0.42	7.87×10^{-2}	9.70×10^{-2}	6.44×10^{-2}
	0.35	6.50×10^{-2}	$1.084.38 \times 10^{-2}$	8.49×10^{-2}
	0.28	1.08×10^{-1}	1.07×10^{-1}	8.30×10^{-2}

contrast, we denote the ROM of **Section 6.1.1** as the static-basis ROM. The adaptive-basis ROM is coupled with different downstream components (Ω_B) to model the full configuration with three different combustion lengths (L_c = 0.42, 0.35, and 0.28 m as shown in **Figure 4**), denoted as adaptive-basis framework.

Following similar evaluation procedures as in **Section 6.1.2**, local pressure signals measured at $x = 0$ are first compared with the FOM-FOM results in **Figures 7A–C** for different combustor lengths. The peak-to-peak pressure oscillation amplitude (p'_{ptp}) and the dominant acoustic frequency (f_a and f_b) are calculated based on the pressure time trace and PSD, respectively and compared against the FOM-FOM results for quantitative assessment, as summarized in **Table 3**. It can be readily seen that, compared to the static-basis results in **Figure 6**, the predictions of the pressure signals for L_c = 0.42 m are improved with adaptive-basis, especially in predicting p'_{ptp} ($< 2\%$ with adaptive-basis ROM versus $> 7\%$ with static-basis ROM). In addition, incorporating basis adaptation in the ROM enables more accurate predictions of the high-frequency pressure responses comparing the pressure PSD in **Figure 7A** (right) to **Figure 6A** (right). More importantly, the framework can also be extended to model other full-system configurations by coupling the upstream adaptive-basis ROM with different downstream components as illustrated in **Figure 4**. As exhibited in **Figures 7B,C**, the adaptive-basis framework is able to predict the pressure characteristics changes due to variations in full-system geometric configurations reasonably well, which shows $< 7\%$ errors in $f <$ 12% errors in p'_{ptp}, as summarized in **Table 3**. Though the characteristics of the pressure field appear to be similar between the single-injector combustors with L_c = 0.42 and 0.35 m, the pressure signals with L_c = 0.28 m exhibit significantly different behaviors, featuring approximately 50% reduction in p'_{ptp} and an additional low-frequency acoustic mode (f_b in **Figure 7C**) appearing in the PSD analysis, in addition to the dominant acoustic mode (f_a). Such distinguishable changes in QoIs (e.g., pressure characteristics) are well-captured using the adaptive-basis framework, which can provide important guidelines for engineering system design (e.g. to design a rocket combustor with reduced pressure oscillations).

Next, we extend the evaluations of adaptive-basis framework to the predictions of time-averaged and RMS fields defined in **Eq. 21**, the errors of which are calculated using **Eq. 22** and summarized in **Table 4**. Significant improvement (e.g., approximately $O(10)$ error reduction) in predicting the time-averaged and RMS fields of P, U, and T can be readily seen comparing **Table 4** (rows corresponding to L_c = 0.42 m) and **Table 2**. The time-averaged and RMS fields predicted using adaptive-basis framework are investigated further in **Figures 7D–F**, which shows excellent agreement with the FOM-FOM results with the magnitudes of the RMS fields predicted correctly and no distinguishable interface mismatches observed in **Figures 6B–D**.

Moreover, the adaptive-basis framework is demonstrated to be capable of predicting the time-averaged and RMS fields for different combustor lengths reasonably accurately as reflected in **Table 4**. Specifically, the time-averaged and RMS fields of temperature (T) are selected to further demonstrate the modeling capabilities of the adaptive-basis framework as shown in **Figures 7G,H** because the temperature dynamics is characterized by chaotic non-stationary and advection-dominated features as shown in **Figure 4**, which prove to be most challenging to represent with static-basis framework as shown in **Figure 6**. Overall, the adaptive-basis framework is able to represent the changes in the time-averaged and RMS temperature fields with variations in the combustor lengths.

6.2 Multi-Injector Model Rocket Combustor

Next, we proceed to demonstrate the component-based domain-decomposition framework on the multi-injector model rocket combustor configuration shown in **Figure 8A**, based on a laboratory rocket model engine [60], originally designed to study combustion instability of transverse acoustics. In **Figure 8A**, we take the five-injector configuration as an example for illustration and also consider the configurations

with three and seven injectors to demonstrate the capabilities of the framework in the following sections. As seen in **Figure 8A**, this configuration consists of five shear coaxial injectors, each of which is similar to the single-injector geometry in **Figure 4**, and is featured with an outer passage, T_1, that introduces fuel near the downstream end of the coaxial inner passage, T_2, that feeds oxidizer to the combustion chamber. The operating conditions in all the multi-injector combustor configurations are maintained with an adiabatic flame temperature of approximately 2,700 K and a mean chamber pressure of 1.3 MPa. The T_1 stream contains gaseous methane (100% CH_4) at 300 K. The T_2 stream is 42% gaseous O_2 by mass and 58% gaseous H_2O by mass at 660 K. Both the T_1 and T_2 streams are fed with constant mass flow rates, 0.67 kg/s and 19.75 kg/s, respectively. A non-reflective boundary condition is imposed at the downstream end of the computational domain with the goal of suppressing longitudinal acoustics in the streamwise direction, which promotes the generation of transverse acoustic waves in the spanwise direction. Similar to the single-injector configuration in **Section 6.1**, combustion is represented by the flamelet progress variable (FPV) model with GRI-1.2 chemical kinetics.

As shown in **Figure 8A**, the five-injector model rocket combustor can be represented by two reference components, the interior injector element (e.g., Ω_3) and the wall injector element (e.g., Ω_5). Therefore, the multi-injector configuration can be modeled using two ROMs trained based on the reference components, denoted as an all-ROM framework, as illustrated in **Section 5**. With the ROM/FOM coupled framework demonstrated using the single-injector configuration, we use the multi-injector configuration to evaluate and demonstrate the ROM/ROM coupled framework.

When the FOM is adopted to model all five components, denoted as all-FOM framework, the resulting representative snapshots of the flow fields are shown in **Figure 8B**, which exhibits two major features that have not been observed in the single-injector case in **Section 6.1**: 1) stronger interactions between components, featured with large-scale vortex shedding approaching the downstream end of the domain; and 2) more complex dynamics at the component interfaces (i.e., $\partial\Omega_{12}$, $\partial\Omega_{23}$, $\partial\Omega_{34}$, and $\partial\Omega_{45}$), featured with both positive and negative spanwise velocity.

6.2.1 Injector-Element Reduced-Order Models Training and Framework Integration

In this section, we discuss the ROM training strategies based on the two different components (the interior and wall injector elements) identified above and how to integrate the two trained ROMs into the framework to model the multi-injector model rocket combustor. We focus on the component-based methods for ROM training as illustrated in **Figures 9A,B** corresponding to the interior and wall injector element respectively. To incorporate the strong interactions between injector elements observed in **Figure 8B**, two adjacent injector elements are included for interior injector-element ROM training (**Figure 9A**) while one additional adjacent injector element is added for wall injector-element ROM training (**Figure 9B**), which cannot be easily accounted by imposing boundary conditions directly as conceptualized in **Figure 2** considering the complexity of the dynamics at the component interfaces. In addition, similar to the single-injector configuration, auxiliary domains with exponentially stretched mesh elements in the spanwise direction are added next to the additional injector elements in ROM training (two for interior injector-element and one for wall injector-element). This incorporates the complex dynamics at the component interfaces, especially for the abrupt changes in the directions of the flow characteristics (e.g., existence of both positive and negative spanwise velocity) observed in **Figure 8B**. Non-reflective boundary conditions are imposed at the downstream end for both the interior and wall injector element ROM training to be consistent with the target multi-injector configuration (**Figure 8A**). Forcing is imposed *via* non-reflective boundary conditions at the side boundaries with backward characteristics q_{u-c} perturbed using the same function in **Eq. 19** to generate the essential

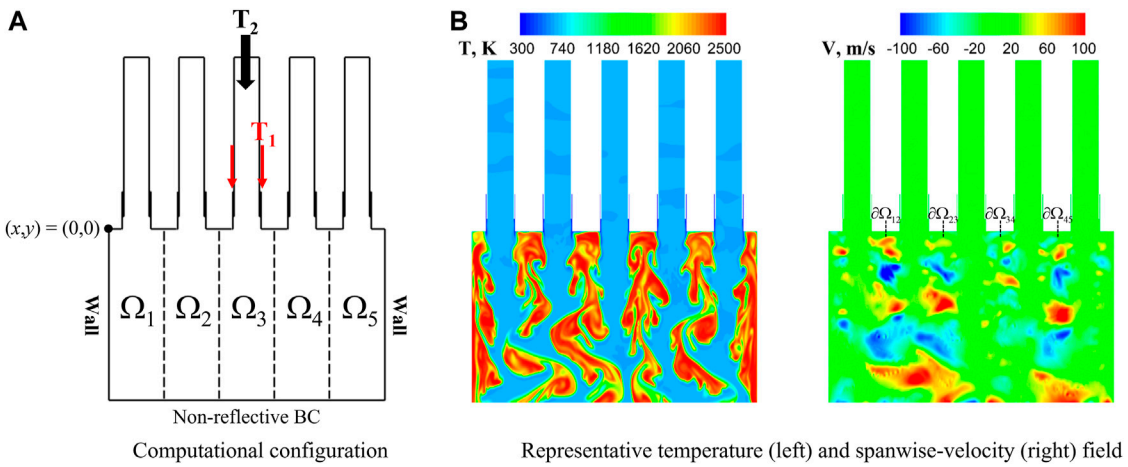


FIGURE 8 | (A) Computational configuration and **(B)** representative temperature and spanwise-velocity field of 2D-planar multi-injector model rocket combustor.

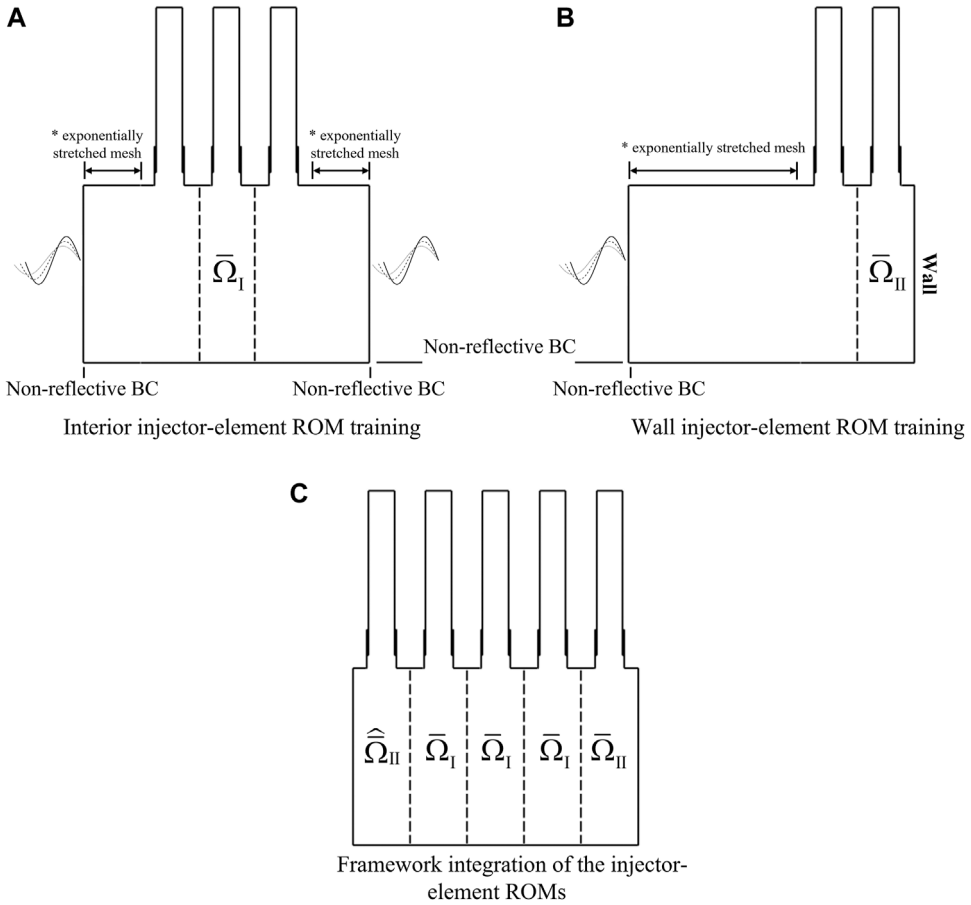


FIGURE 9 | Component-based ROM training strategies for **(A)** the interior injector element and **(B)** the wall injector element and **(C)** the integration of the injector-element ROMs for the 2D-planar multi-injector model rocket combustor.

dynamics anticipated in the full system. Here, we choose to impose the forcing with $A = 0$ to mimic a broad-and response, which presumably contains rich responses in the frequency domain. The solution snapshots are extracted corresponding to the regions bounded by dashed lines in **Figures 8A,B**, which are then used to construct the interior injector-element ROM and the wall injector-element ROM, respectively. The resulting ROMs are coupled through the direct flux matching method at the component interfaces adopted to model all the interior injector elements (Ω_2, Ω_3, and Ω_4 in **Figure 8A**) and $\bar{\Omega}_{II}$ for all the wall injector elements (Ω_1 and Ω_5 in **Figure 8A**), resulting in the all-ROM framework as shown in **Figure 9C**.

Since the two wall injector elements are geometrically identical (i.e., reflective symmetry) to each other, $\bar{\Omega}_{II}$ is mirrored about the center axis to generate a reflective counterpart when adopted to model Ω_1, reflected as $\hat{\bar{\Omega}}_{II}$. Comparisons of the training domains in **Figures 9A,B** with the full five-element configuration in **Figure 9C**, suggest that the training costs in **Figures 9A,B** are similar to that of the full five-element configuration. The advantage of this approach is that the training costs is fixed for any number of elements (e.g., three, seven, nine or more) and for larger systems, substantial cost savings can be realized. The current work serves as a first step toward modeling practical rocket engines which typically consist of hundreds of injector elements.

6.2.2 Performance

Based on the investigations using the single-injector configuration in **Section 6.1**, we apply the adaptive-basis method, introduced in **Section 4**, to develop the two component-based ROMs, $\bar{\Omega}_I$ and $\bar{\Omega}_{II}$. Similar to the single-injector case, 10 snapshots are collected from the offline ROM training stage for each component in **Figures 9A,B** to generate the initial two sets of trial basis $\mathbf{V}_p^0$. 5 POD modes covering $> 99.9\%$ of the total energy are selected to construct adaptive-basis ROMs *via* MP-LSVT formulation, respectively. The trial basis $\mathbf{V}_p^k$ is adapted at each time step k based on the formulation in **Eq. 13** and following the schematics in **Figure 9C**, the adaptive-basis ROMs are then coupled to model 3 multi-injector configurations with three, five, and seven injector elements.

Next, we proceed to evaluate the performance of the all-adaptive-basis-ROM framework based on the results from the

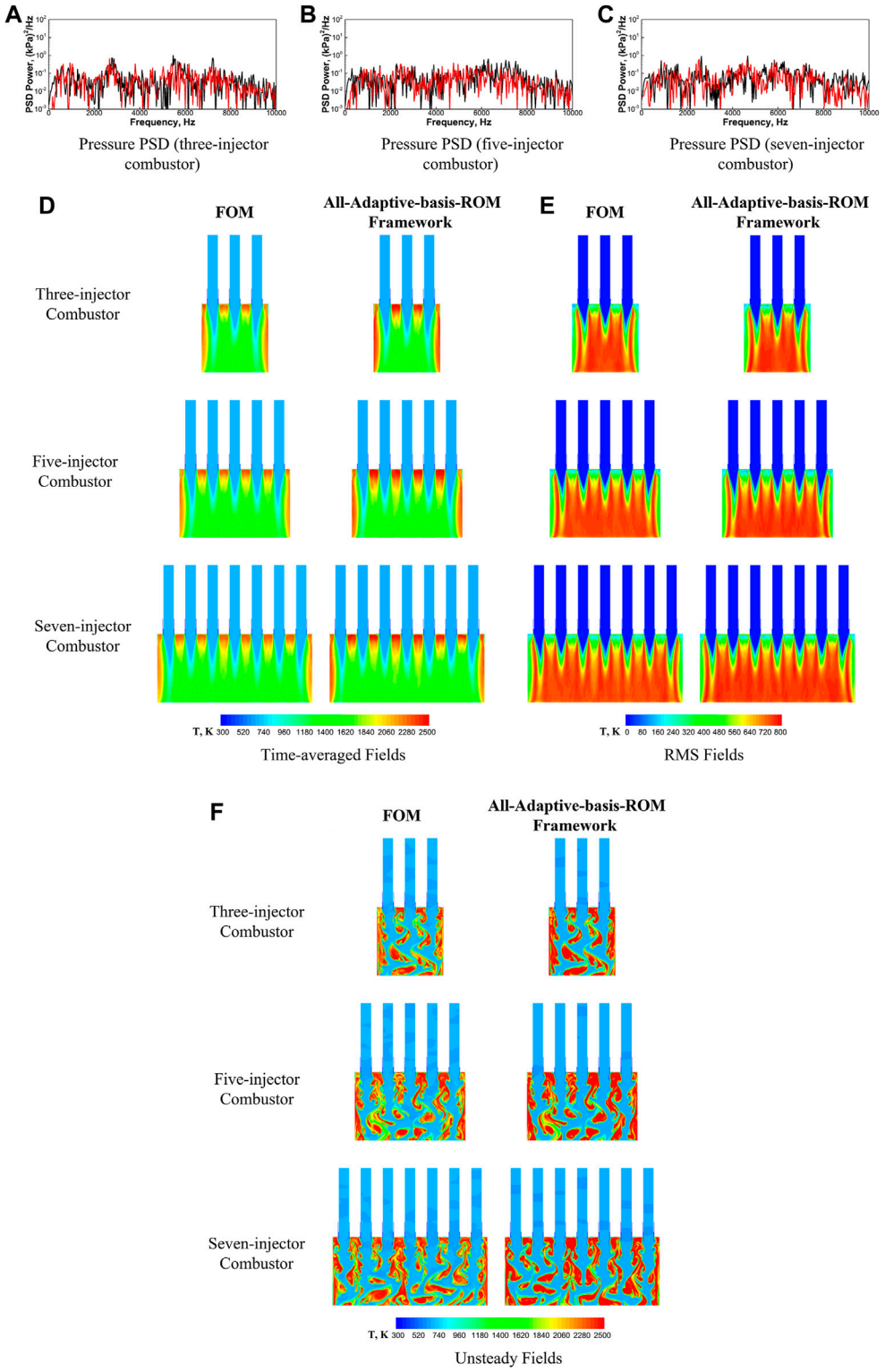


FIGURE 10 | Comparisons of pressure PSDs **(A–C)** measured at $(x, y) = (0, 0)$ in **Figure 8** with different models (all-adaptive-basis-ROM (Red) framework vs. FOM (Black), **(D)** time-averaged, **(E)** RMS, and **(F)** unsteady temperature fields with different models (all-adaptive-basis-ROM framework vs. FOM) used in the framework for the three-injector (top), five-injector (middle), and seven-injector (bottom) configurations for the 2D-planar multi-injector model rocket combustor.

TABLE 5 | Comparisons of the errors in predicting time-averaged and RMS fields of pressure (*P*), temperature(*T*), and axial velocity (*U*) corresponding to **Figures 10D-E** using the all-adaptive-basis-ROM framework for the multi-injector combustor.

QoI	Number of Injectors	ϵ_P	ϵ_T	ϵ_U
Time-averaged Fields	3	4.02×10^{-3}	3.57×10^{-2}	3.50×10^{-2}
	5	4.07×10^{-3}	3.66×10^{-2}	3.69×10^{-2}
	7	5.37×10^{-3}	3.25×10^{-2}	3.89×10^{-2}
RMS Fields	3	1.42×10^{-1}	5.36×10^{-2}	8.16×10^{-2}
	5	1.81×10^{-1}	6.13×10^{-2}	9.48×10^{-2}
	7	1.45×10^{-1}	5.63×10^{-2}	9.32×10^{-2}

FOM, which is taken as the ground truth solution. **Figures 10A–C** compares the local PSD predicted using the FOM and with the all-adaptive-basis-ROM framework, which are measured at the corner of the left wall, i.e., $(x, y) = (0, 0)$ in **Figure 8A** for all 3 multi-injector configurations. Different from the single-injector cases in **Section 6.1**, the true pressure signals from the all-FOM results in the multi-injector configurations do not exhibit distinguishable coherent oscillating patterns as in **Figure 4**. This aspect can be expected to be more challenging to be predicted by the ROM.

The predictions using the all-adaptive-basis-ROM framework show good agreement with the all-FOM framework results, especially in capturing the changes in PSD distributions due to the configuration variations with the number of injector elements increased from three to seven. For example, the wide-band frequency peak between 2,500 and 3,000 Hz in the three-injector combustor and the peak near 2,500 Hz for the seven-injector configuration are both well predicted by the all-adaptive-basis-ROM framework. More importantly, the broad-band PSD distributions (i.e., no identifiable frequency peaks) in the five-injector configuration are also accurately captured.

Having successfully demonstrated the ability of the adaptive ROM to capture changes in pressure oscillations arising from configuration variations, we next look at their ability to predict time-averaged and RMS fields of target state variables (*P*, *U*, and *T*) is assessed. The accuracy of the framework is evaluated based on the errors defined in **Eq. 22**, summarized in **Table 5**. It can be readily seen that the all-adaptive-basis-ROM framework is able to accurately predict the time-averaged fields of selected state variables with errors below 4% while even for the generally more challenging RMS fields, the prediction errors are shown to be below 18%, given the complexity and chaotic features of the dynamics present in the multi-injector problems, indicated by the broad-band frequency distributions in **Figures 10A–C**. Specifically, the time-averaged, RMS, and representative unsteady fields of temperature (*T*) are selected to demonstrate the predictive capabilities of the all-adaptive-basis-ROM framework as shown in **Figures 10D,E** for the three different multi-injector configurations, which exhibits good agreement between all-adaptive-basis-ROM and all-FOM framework results. But it still need to be pointed out that the all-adaptive-basis-ROM framework predicts elongated high-temperature zones between injector elements compared to the all-FOM framework results, indicating that the current methodology may require further improvement.

7 CONCLUSION

A component-based domain-decomposition framework is established for the modeling of large-scale systems that cannot be directly accessed using the high-fidelity simulations (e.g., a rocket engine, a wind farm, and a compressor). This approach decomposes the full system into different components, each of which can flexibly adopt different modeling strategies (e.g., ROM or FOM), balancing physical complexity with accuracy requirements. Under the premise that most of the components share identical features and can be represented by a few reference components, a component-based reduced-order model (ROM) training strategy is proposed and demonstrated, which requires only the high-fidelity simulations of the individual components. System-level feedback and responses in the training dataset is emulated by imposing boundary forcing. This leads to a significant saving in computational cost during ROM training. The model-form preserving least-squares with variable transformation (MP-LSVT) ROM formulation is pursued with enhancement through basis adaptation to construct the component-based ROMs. The trained ROMs can be adopted to model components with identical geometric features and coupled with either a reduced-fidelity full-order model (RF-FOM) or ROMs *via* a direct flux matching method to enable both accurate and efficient simulations of large-scale systems with different geometric configurations.

Detailed evaluations of the framework were first performed based on a planar single-injector model rocket configuration with varying combustor lengths, each of which exhibit different dynamic behaviors. The framework separates the single-injector configuration into two components, the upstream injector element and the downstream combustor + nozzle, the former of which adopts MP-LSVT ROM for modeling, while the latter adopts a FOM. Two methods, (system-based and component-based) are used to train the injector-element ROM. It was demonstrated that the upstream-component ROMs from both training methods, when coupled with the downstream-component FOM, can produce reasonably accurate predictions of the pressure oscillations while the component-based method requires much less computational cost. However, the ROM/FOM framework encounters difficulties in representing the time-averaged and root-mean-square (RMS) fields of the target state variables while distinguishable solution mismatches are observed at the component interface. To address this limitation, basis adaptation is incorporated in the MP-LSVT formulation to

enhance ROM capabilities, which significantly improves the predictive accuracy of the framework and more importantly, is capable of representing changes in dynamic behaviors due to the variations in combustor length.

The framework was then extended to a 2D-planar multi-injector model rocket configuration with different number of injector elements, which can be represented by two reference injector-element components. High-fidelity training simulations are then conducted on the two reference components to develop the component-based ROMs *via* MP-LSVT formulation with basis adaptation. The framework is demonstrated to be capable of predicting all the quantities-of-interest (QoIs) accurately, including local pressure oscillations, time-averaged and RMS fields of target state variables for the multi-injector configuration with different number of injector elements.

Though preliminary and demonstrated for only 2D problems, the component-based domain-decomposition framework with adaptive-basis ROMs is directly applicable to 3D problems and mostly importantly serves as a stepping stone towards modeling practical large-scale engineering systems (e.g., a RD-170 rocket engine [16]). Before this framework can be adopted by engineers in many-query applications (such as design and uncertainty quantification) of the full system, two major aspects need to be considered: 1) efficiency—hyper-reduction has to be considered to enable more efficient ROM calculations as mentioned in **Section 4**; and 2), scalability—the ROMs must be amenable for execution on memory-restricted computers such as desktop workstations or embedded systems—i.e., they need to be load-balanced and scalable in terms of computational resources available.

Discussions are provided by the current authors [14] on constructing scalable, load-balanced, and hyper-reduced static-basis ROMs while all these aspects remain to be further investigated for adaptive-basis ROM development. To address the remaining gaps, good avenue for future work can be on incorporating adaptive sparse sampling methods (e.g., [50]) in the adaptive-basis ROM to achieve computational efficiency enhancement, while exploring dynamic methods to achieve scalability when the sampling elements are getting adapted. In addition, the component-based framework is designed to be generally compatible with different types of ROM methods and hence instead of the intrusive ROM used in the current work, non-intrusive ROM methods [44,78] may also be considered for the future work.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors upon request.

AUTHOR CONTRIBUTIONS

CH, KD, and CM all contributed to formulation of the component-based reduced-order modeling framework and the component-based ROM training strategies. CH established the test problems, perform the numerical investigations, and wrote the first draft of the manuscript. KD formulated the adaptive-basis method provided in the manuscript and contributed to the method for ROM integration in the framework. CM formulated the component-based ROM framework and conceptualized its application for rocket combustion problems. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

The authors acknowledge support from the US Air Force under the Center of Excellence grant FA9550-17-1-0195, titled Multi-Fidelity Modeling of Rocket Combustor Dynamics (Program Managers: Dr. Mitat Birkan, Dr. Fariba Fahroo, and Dr. Ramakanth Munipalli).

REFERENCES

1. Wang ZJ, Li Y, Jia F, Laskowski GM, Kopriva J, Paliath U, et al. Towards Industrial Large Eddy Simulation Using the Fr/cpr Method. *Comput Fluids* (2017) 156:579–89. doi:10.1016/j.compfluid.2017.04.026
2. Aditya K, Gruber A, Xu C, Lu T, Krisman A, Bothien MR, et al. Direct Numerical Simulation of Flame Stabilization Assisted by Autoignition in a Reheat Gas Turbine Combustor. *Proc Combustion Inst* (2019) 37:2635–42. doi:10.1016/j.proci.2018.06.084
3. Oefelein JC. Advances in Modeling Supercritical Fluid Behavior and Combustion in High-Pressure Propulsion Systems. *AIAA Scitech Forum* (2019):0634. doi:10.2514/6.2019-0634
4. Lumley JL, Poje A. Low-dimensional Models for Flows with Density Fluctuations. *Phys Fluids* (1997) 9:2023–31. doi:10.1063/1.869321
5. Graham W, Peraire J, Tang K. Optimal Control of Vortex Shedding Using Low Order Models Part I: Open-Loop Model Development. *Int J Numer Methods* (1997) 44:945–72. doi:10.1002/(SICI)1097-0207(19990310)44:7<973::AID-NME538>3.0
6. Lucia DJ, Beran PS. Projection Methods for Reduced Order Models of Compressible Flows. *J Comput Phys* (2003) 188:252–80. doi:10.1016/s0021-9991(03)00166-9
7. Barbagallo A, Sipp D, Schmid PJ. Closed-loop Control of an Open Cavity Flow Using Reduced-Order Models. *J Fluid Mech* (2009) 641:1–50. doi:10.1017/s0022112009991418
8. Barbagallo A, Sipp D, Schmid PJ. Input–output Measures for Model Reduction and Closed-Loop Control: Application to Global Modes. *J Fluid Mech* (2011) 685:23–53. doi:10.1017/jfm.2011.271
9. Barbagallo A, Dergham G, Sipp D, Schmid PJ, Robinet JC. Closed-loop Control of Unsteadiness over a Rounded Backward-Facing Step. *J Fluid Mech* (2012) 703:326–62. doi:10.1017/jfm.2012.223
10. Lucia DJ, Beran PS, Silva WA. Reduced-order Modeling: New Approaches for Computational Physics. *Prog Aerospace Sci* (2004) 40:51–117. doi:10.1016/j.paerosci.2003.12.001
11. Lieu T, Farhat C. Adaptation of Aeroelastic Reduced-Order Models and Application to an F-16 Configuration. *AIAA J* (2007) 45:1244–57. doi:10.2514/1.24512
12. Blonigan PJ, Carlberg K, Rizzi F, Howard M, Fike JA. Model Reduction for Hypersonic Aerodynamics via Conservative LSPG Projection and Hyper-Reduction. *AIAA J.* (2021) 59 (4):1296–1312.
13. Huang C, Duraisamy K, Merkle CL. Investigations and Improvement of Robustness of Reduced-Order Models of Reacting Flow. *AIAA J* (2019) 57:5377–89. doi:10.2514/1.j058392

14. Huang C, Wentland CR, Duraisamy K, Merkle C. Model Reduction for Multi-Scale Transport Problems Using Model-form Preserving Least-Squares Projections with Variable Transformation. *J Comput Phys* (2022) 448: 110742. doi:10.1016/j.jcp.2021.110742
15. Urbano A, Selle L, Staffelbach G, Cuenot B, Schmitt T, Ducruix S, et al. Exploration of Combustion Instability Triggering Using Large Eddy Simulation of a Multiple Injector Liquid Rocket Engine. *Combustion and Flame* (2016) 169:129–40. doi:10.1016/j.combustflame.2016.03.020
16. Fedorov V, Chvanov V, Chelkis F, Ivanov N, Lozinskay I, Buryak A. *The Chamber Cooling System of Rd-170 Engine Family: Design, Parameters, and Hardware Investigation Data*. Sacramento, CA, USA: AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit (2006). doi:10.2514/6.2006-4363
17. Maday Y, Ronquist EM. A Reduced-Basis Element Method. *J Scientific Comput* (2002) 17:447–59. doi:10.1023/a:1015197908587
18. Iapichino L, Quarteroni A, Rozza G. Reduced Basis Method and Domain Decomposition for Elliptic Problems in Networks and Complex Parametrized Geometries. *Comput Math Appl* (2016) 71:408–30. doi:10.1016/j.camwa.2015.12.001
19. Phuong Huynh DB, Knezevic DJ, Patera AT. A Static Condensation Reduced Basis Element Method : Approximation Anda Posteriorierror Estimation. *ESAIM: Math Model Numer Anal* (2012) 47:213–51. doi:10.1051/m2an/2012022
20. Gropp WD, Keyes DE. Domain Decomposition Methods in Computational Fluid Dynamics. *Int J Numer Methods Fluids* (1992) 14:147–65. doi:10.1002/fld.1650140203
21. Sitaraman J, Floros M, Wissink A, Potsdam M. Parallel Domain Connectivity Algorithm for Unsteady Flow Computations Using Overlapping and Adaptive Grids. *J Comput Phys* (2010) 229:4703–23. doi:10.1016/j.jcp.2010.03.008
22. Chaturantabut S, Beattie C, Gugercin S. Structure-preserving Model Reduction for Nonlinear Port-Hamiltonian Systems. *SIAM J Sci Comput* (2016) 38: B837–65. doi:10.1137/15m1055085
23. Gugercin S, Polyuga RV, Beattie C, van der Schaft A. Structure-preserving Tangential Interpolation for Model Reduction of Port-Hamiltonian Systems. *Automatica* (2012) 48:1963–74. doi:10.1016/j.automatica.2012.05.052
24. Califano F, Rashad R, Schuller FP, Stramigioli S. Energetic Decomposition of Distributed Systems with Moving Material Domains: The Port-Hamiltonian Model of Fluid-Structure Interaction. *J Geometry Phys* (2022) 175:104477. doi:10.1016/j.geomphys.2022.104477
25. Willcox K, Peraire J, Paduano JD. Application of Model Order Reduction to Compressor Aeroelastic Models. *J Eng Gas Turbine Power* (2002) 124:332–9. doi:10.1115/1.1416152
26. Maday Y, Ronquist EM. The Reduced Basis Element Method: Application to a thermal Fin Problem. *SIAM J Sci Comput* (2004) 26:240–58. doi:10.1137/s1064827502419932
27. Iapichino L, Quarteroni A, Rozza G. A Reduced Basis Hybrid Method for the Coupling of Parametrized Domains Represented by Fluidic Networks. *Comput Methods Appl Mech Eng* (2012) 221-222:63–82. doi:10.1016/j.cma.2012.02.005
28. Eftang JL, Patera AT. A Port-Reduced Static Condensation Reduced Basis Element Method for Large Component-Synthesized Structures: Approximation and A Posteriori Error Estimation." in *Advanced Modeling and Simulation in Engineering Sciences* (2013).
29. Smetana K, Patera AT. Optimal Local Approximation Spaces for Component-Based Static Condensation Procedures. *SIAM J Sci Comput* (2016) 38: A3318–56. doi:10.1137/15m1009603
30. Kapteyn MG, Knezevic DJ, Huynh DBP, Tran M, Willcox KE. Data-driven Physics-Based Digital Twins via a Library of Component-Based Reduced-Order Models. *Int J Numer Methods Eng* (2020) 123:2986–3003. doi:10.1002/nme.6423
31. McBane S, Choi Y. Component-wise Reduced Order Model Lattice-type Structure Design. *Comput Methods Appl Mech Eng* (2021) 381:113813. doi:10.1016/j.cma.2021.113813
32. Lucia DJ, King PI, Beran PS. *Reduced Order Modeling of a Two-Dimensional Flow with Moving Shocks*. Computers & Fluids (2003) 32(7):917-38.
33. Buffoni M, Telib H, Iollo A. Iterative Methods for Model Reduction by Domain Decomposition. *Comput Fluids* (2009) 38:1160–7. doi:10.1016/j.compfluid.2008.11.008
34. Baiges J, Codina R, Idelsohn S. A Domain Decomposition Strategy for Reduced Order Models. Application to the Incompressible Navier–Stokes Equations. *Comput Methods Appl Mech Eng* (2013) 267:23–42. doi:10.1016/j.cma.2013.08.001
35. Ahmed SE, San O, Kara K, Younis R, Rasheed A. Multifidelity Computing for Coupling Full and Reduced Order Models. *PLoS One* (2021) 16:e0246092. doi:10.1371/journal.pone.0246092
36. Hoang C, Choi Y, Carlberg K. Domain-decomposition Least-Squares Petrov–Galerkin (Dd-lspg) Nonlinear Model Reduction. *Comput Methods Appl Mech Eng* (2021) 384:113997. doi:10.1016/j.cma.2021.113997
37. Xiao D, Fang F, Heaney CE, Navon IM, Pain CC. A Domain Decomposition Method for the Non-intrusive Reduced Order Modelling of Fluid Flow. *Comput Methods Appl Mech Eng* (2019) 354:307–30. doi:10.1016/j.cma.2019.05.039
38. Xiao D, Heaney CE, Fang F, Mottet L, Hu R, Bistrian DA, et al. A Domain Decomposition Non-intrusive Reduced Order Model for Turbulent Flows. *Comput Fluids* (2019) 182:15–27. doi:10.1016/j.compfluid.2019.02.012
39. Xiao C, Leeuwenburgh O, Lin HX, Heemink A. Efficient Estimation of Space Varying Parameters in Numerical Models Using Non-intrusive Subdomain Reduced Order Modeling. *J Comput Phys* (2021) 424:109867. doi:10.1016/j.jcp.2020.109867
40. Huang C, Anderson WE, Merkle CL, Sankaran V. Multifidelity Framework for Modeling Combustion Dynamics. *AIAA J* (2019) 57:2055–68. doi:10.2514/1.J057061
41. Xu J, Huang C, Duraisamy K. Reduced-order Modeling Framework for Combustor Instabilities Using Truncated Domain Training. *AIAA J* (2020) 58:618–32. doi:10.2514/1.J057959
42. Butcher J. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons (2016). p. 333–87. doi:10.1002/9781119121534.ch4
43. Carlberg K, Barone M, Antil H. Galerkin V. Least-Squares Petrov-Galerkin Projection in Nonlinear Model Reduction. *J Comput Phys* (2017) 330:693–734. doi:10.1016/j.jcp.2016.10.033
44. McQuarrie SA, Huang C, Willcox KE. Data-driven Reduced-Order Models via Regularised Operator Inference for a Single-Injector Combustion Process. *J R Soc New Zealand* (2021) 51:194–211. doi:10.1080/03036758.2020.1863237
45. Amsallem D, Zahr MJ, Washabaugh K. Fast Local Reduced Basis Updates for the Efficient Reduction of Nonlinear Systems with Hyper-Reduction. *Adv Comput Math* (2015) 41:1187–230. doi:10.1007/s10444-015-9409-0
46. Geelen R, Willcox K. Localized Non-intrusive Reduced-Order Modeling in the Operator Inference Framework. *Phil. Trans. R. Soc. A.* (2021): 3802021020620210206. doi:10.1098/rsta.2021.0206
47. Lee K, Carlberg KT. Model Reduction of Dynamical Systems on Nonlinear Manifolds Using Deep Convolutional Autoencoders. *J Comput Phys* (2020) 404:108973. doi:10.1016/j.jcp.2019.108973
48. Kim Y, Choi Y, Widemann D, Zohdi T. A Fast and Accurate Physics-Informed Neural Network Reduced Order Model with Shallow Masked Autoencoder. *J Comput Phys* (2022) 451:110841. doi:10.1016/j.jcp.2021.110841
49. Peherstorfer B, Willcox K. Online Adaptive Model Reduction for Nonlinear Systems via Low-Rank Updates. *SIAM J Sci Comput* (2015) 37:A2123–50. doi:10.1137/140989169
50. Peherstorfer B. Model Reduction for Transport-Dominated Problems via Online Adaptive Bases and Adaptive Sampling. *SIAM J Sci Comput* (2020) 42:A2803–36. doi:10.1137/19M1257275
51. Zimmermann R, Peherstorfer B, Willcox K. Geometric Subspace Updates with Applications to Online Adaptive Nonlinear Model Reduction. *SIAM J Matrix Anal Appl* (2018) 39:234–61. doi:10.1137/17m1123286
52. Chaturantabut S, Sorensen DC. Nonlinear Model Reduction via Discrete Empirical Interpolation. *SIAM J Sci Comput* (2010) 32:2737–64. doi:10.1137/090766498
53. Everson R, Sirovich L. Karhunen–Loève Procedure for Gappy Data. *J Opt Soc Am A* (1995) 12:1657. doi:10.1364/JOSAA.12.001657
54. Shipley K. *Multi-injector Modeling of Transverse Combustion Instability Experiments* (2014).*Thesis*
55. Harvazinski ME, Gejji R, Talley DG, Orth MR, Anderson WE, Pourpoint TL, et al. "Modeling of Transverse Combustion Instability," in *AIAA Scitech*. San Diego, CA, USA (2019), 1732
56. Henning P, Peterseim D. Oversampling for the Multiscale Finite Element Method. *Multiscale Model Simul* (2013) 11:1149–75. doi:10.1137/120900332
57. Comer AL, Huang C, Sardeshmukh S, Rankin BA, Harvazinski ME, Sankaran V, et al. Sensitivity Analysis of bluff-body Stabilized Premixed Flame Large-Eddy Simulations. *J Propulsion Power* (2021) 37:211–22. doi:10.2514/1.B37801

58. Harvazinski ME, Huang C, Sankaran V, Feldman TW, Anderson WE, Merkle CL, et al. Coupling between Hydrodynamics, Acoustics, and Heat Release in a Selfexcited Unstable Combustor. *Phys Fluids* (2015) 27:045102. doi:10.1063/1.4916673
59. Yu Y, Sisco JC, Rosen S, Madhav A, Anderson WE. Spontaneous Longitudinal Combustion Instability in a Continuously-Variable Resonance Combustor. *J Propulsion Power* (2012) 28:876–87. doi:10.2514/1.b34308
60. Morgan CJ, Shipley KJ, Anderson WE. Comparative Evaluation between experiment and Simulation for a Transverse Instability. *J Propulsion Power* (2015) 31:1696–706. doi:10.2514/1.B35759
61. Huang C, Gejji R, Anderson W, Yoon C, Sankaran V. Combustion Dynamics in a Single-Element Lean Direct Injection Gas Turbine Combustor. *Combustion Sci Technol* (2019) 192:2371–98. doi:10.1080/00102202.2019.1646732
62. Roe P. Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. *J Comput Phys* (1981) 43:357–72. doi:10.1016/0021-9991(81)90128-5
63. Mitchell C. *Improved Reconstruction Schemes for the Navier–Stokes Equations on Unstructured Meshes*. Reno, NV, USA: 32nd Aerospace Sciences Meeting and Exhibit (1994). doi:10.2514/6.1994-642
64. Barth T, Jespersen D. *The Design and Application of Upwind Schemes on Unstructured Meshes*. Reno, NV, USA: 27th Aerospace Sciences Meeting (1989). doi:10.2514/6.1989-366
65. Pierce CD, Moin P. Progress-variable Approach for Large-Eddy Simulation of Non-premixed Turbulent Combustion. *J Fluid Mech* (2004) 504:73–97. doi:10.1017/s0022112004008213
66. Frenklach M, Wang H, Goldenberg M, Smith G, Golden D, Bowman C, et al. *GRI-Mech—an Optimized Detailed Chemical Reaction Mechanism for Methane Combustion*. Gas Research Institute topical report no. GRI-95/0058 (1995).
67. Coclite A, Pascazio G, De Palma P, Cutrone L, Ihme M. An SMLD Joint PDF Model for Turbulent Non-premixed Combustion Using the Flamelet Progress-Variable Approach. *Flow Turbul Combust* (2015) 95:97–119. doi:10.1007/s10494-015-9609-1
68. Lee K, Carlberg KT. Model Reduction of Dynamical Systems on Nonlinear Manifolds Using Deep Convolutional Autoencoders. *J Comput Phys* (2020) 404:108973. doi:10.1016/j.jcp.2019.108973
69. Barone M, Kalashnikova I, Segalman D, Heidi K. Stable Galerkin Reduced Order Models for Linearized Compressible Flow. *J Comput Phys* (2009) 228:1932–46. doi:10.1016/j.jcp.2008.11.015
70. San O, Maulik R. Neural Network Closures for Nonlinear Model Order Reduction. *Adv Comput Math* (2018) 44:1717–50. doi:10.1007/s10444-018-9590-z
71. Stabile G, Ballarin F, Zuccarino G, Rozza G. A Reduced Order Variational Multiscale Approach for Turbulent Flows. *Adv Comput Math* (2019) 45:2349–68. doi:10.1007/s10444-019-09712-x
72. Grimberg S, Farhat C, Youkilis N. On the Stability of Projection-Based Model Order Reduction for Convection-Dominated Laminar and Turbulent Flows. *J Comput Phys* (2020) 419:109681. doi:10.1016/j.jcp.2020.109681
73. Farhat C, Macedo A, Lesoinne M, Roux FX, Magoules F, Bourdonnaie AL, et al. Two-level Domain Decomposition Methods with lagrange Multipliers for the Fast Iterative Solution of Acoustic Scattering Problems. *Comput Methods Appl Mech Eng* (2000) 184:213–39. doi:10.1016/s0045-7825(99)00229-7
74. Saas L, Faille I, Nataf F, Willien F. Finite Volume Methods for Domain Decomposition on Nonmatching Grids with Arbitrary Interface Conditions. *SIAM J Numer Anal* (2005) 43:860–90. doi:10.1137/s0036142903434059
75. Brazell MJ, Sitaraman J, Mavriplis DJ. An Overset Mesh Approach for 3d Mixed Element High-Order Discretizations. *J Comput Phys* (2016) 322:33–51. doi:10.1016/j.jcp.2016.06.031
76. Drmač Z, Gugercin S. A New Selection Operator for the Discrete Empirical Interpolation Method—Improved A Priori Error Bound and Extensions. *SIAM J Sci Comput* (2016) 38:A631–48. doi:10.1137/15m1019271
77. Peherstorfer B, Drmač Z, Gugercin S. Stability of Discrete Empirical Interpolation and Gappy Proper Orthogonal Decomposition with Randomized and Deterministic Sampling Points. *SIAM J Sci Comput* (2020) 42:A2837–64. doi:10.1137/19M1307391
78. Swischuk R, Kramer B, Huang C, Willcox K. Learning Physics-Based Reduced-Order Models for a Single-Injector Combustion Process. *AIAA J* (2020) 58:2658–72. doi:10.2514/1.j058943

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 07 September 2022
DOI 10.3389/fphy.2022.903169

OPEN ACCESS

EDITED BY
Traian Iliescu,
Virginia Tech, United States

REVIEWED BY
Marin I. Marin,
Transilvania University of Brașov, Romania
Changhong Mou,
University of Wisconsin-Madison, United States
Zhu Wang,
University of South Carolina, United States

*CORRESPONDENCE
Ping-Hsuan Tsai,
pht2@illinois.edu
Paul Fischer,
fischerp@illinois.edu

SPECIALTY SECTION
This article was submitted to Statistical and Computational Physics, a section of the journal Frontiers in Physics

RECEIVED 24 March 2022
ACCEPTED 07 July 2022
PUBLISHED 07 September 2022

CITATION
Tsai P-H and Fischer P (2022), Parametric model-order-reduction development for unsteady convection. *Front. Phys.* 10:903169. doi: 10.3389/fphy.2022.903169

Parametric model-order-reduction development for unsteady convection

Ping-Hsuan Tsai[1]* and Paul Fischer[1,2]*

[1]Department of Computer Science, University of Illinois, Urbana-Champaign, Champaign, IL, United States, [2]Department of Mechanical Science and Engineering, University of Illinois, Urbana-Champaign, Champaign, IL, United States

A time-averaged error indicator with POD-*h*Greedy is developed to drive parametric model order reduction (pMOR) for 2D unsteady natural convection in a high-aspect ratio slot parameterized with the Prandtl number, Rayleigh number, and slot angle with respect to the gravity. The error indicator is extended to accommodate the energy equation and Leray regularization. Despite being two-dimensional and laminar, the target flow regime presents several challenges: 1) there is a bifurcation in the angle parameter space; 2) the solution can be multivalued, even at steady state; and 3) the solution exhibits spatio-temporal chaos at several points in the parameter space. The authors explore several reduced-order models (ROMs) and demonstrate that Leray-regularized Galerkin ROMs provide a robust solution approach for this class of flows. They further demonstrate that error-indicated pMOR can efficiently predict several QOIs, such as mean flow, mean Nusselt number and mean turbulent kinetic energy, even in the presence of a bifurcation. Finally, they show that spatio-temporal chaos can lead to lack of reproducibility in both the full-order model and the reduced-order model and that the variance in the full-order model provides a lower bound on the pMOR error in these cases.

KEYWORDS

reduced basis method, model-order reduction (MOR), spatio-temporal chaos, a posteriori error estimate, proper orthogonal decomposition (POD), stabilization, leray regularization, incompressible Navier-Stokes equations

1 Introduction

Fluid-thermal analysis via direct numerical (DNS), large-eddy (LES), and even unsteady Reynolds-averaged Navier-Stokes (uRANS) have became tractable in geometries of ever increasing complexity due to advances in high-performance computing and modern algorithms over the past several decades. Despite these advances, when it comes to *routine analysis and design of thermal hydraulic systems*, which requires running hundreds of cases, the cost remains prohibitive. To overcome this

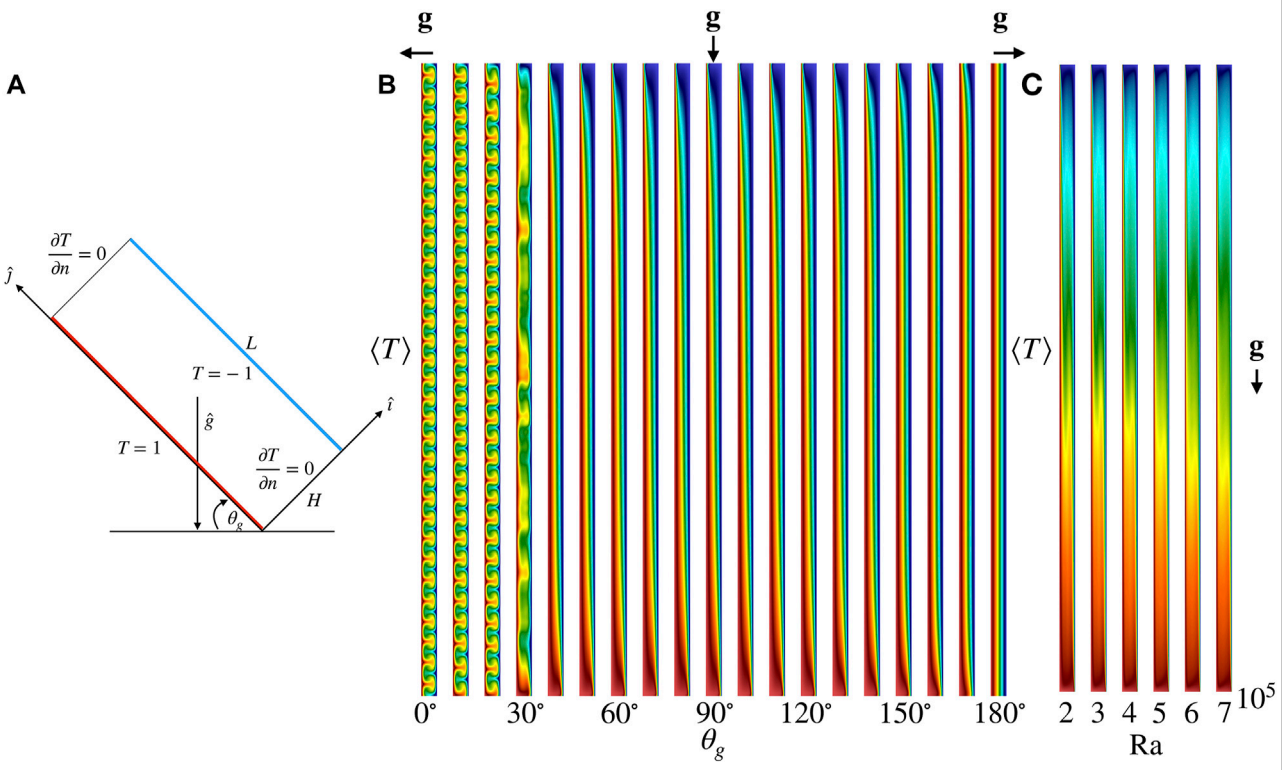


FIGURE 1
(A) Problem configuration. **(B)** FOM mean (or steady) temperature for 19 uniformly-spaced $\theta_g \in [0^\circ, 180^\circ]$ at Ra = 10^4 and Pr = 7.2. **(C)** FOM mean temperature for 6 uniformly-spaced Ra $\in [2 \times 10^5, 7 \times 10^5]$ at $\theta_g = 90^\circ$ and Pr = 7.2.

issue, a rapid turn-around tool for engineering query is required; *parametric model order reduction* (pMOR) is one of the promising approaches.

The main idea of pMOR is to reduce the computational burden by employing reduced-order models (ROMs) built on data from full-order models (FOMs) such as DNS, LES, or RANS-based simulations. A fundamental requirement in this case is to determine how well these approaches can reproduce the flow dynamics with same input parameters as the originating FOM, which is known as the *reproduction problem*. For turbulent flows, FOM can require $\mathcal{N} = 10^7 - 10^{11}$ degrees-of-freedom, while ROMs could potentially represent the flow dynamics which govern the behavior of quantities of interest (QOIs) with only $N \approx 10^1$–10^2 basis functions. Most often, the basis is obtained from a proper orthogonal decomposition (POD) of the FOM flow snapshots (i.e., the velocity and temperature fields). The basis is then used in conjunction with the governing partial differential equation to form a low-dimensional dynamical system that, ideally, captures the principal features of the underlying flows [1–8].

As noted in [9], a successful pMOR for *unsteady* flows must be able to address *1)* the *reproduction problem* and *2)* the *parametric problem*. In the reproduction problem, the ROM and FOM are evaluated at the same $p_j^\star \in \mathcal{P}_{\text{anchor}}$ and the ability of the ROM to recover the QOIs at $p_j^\star$ is examined. In the parametric problem, ROMs are built from a small number of FOMs that are generated over a set of *anchor* points in the parameter space, $\mathcal{P}_{\text{anchor}} = \{p_1^\star, \ldots, p_m^\star\}$, and the ability of ROMs to predict the QOIs at $p \notin \mathcal{P}_{\text{anchor}}$ is examined. An *error indicator* to assess the ROM's fidelity at any given $p \in \mathcal{P}_{\text{train}}$ is usually required to efficiently construct $\mathcal{P}_{\text{anchor}}$. *Error-indicated pMOR* has the potential to be an important engineering analysis tool [10, 11].

In this work we explore the pMOR process for a surprisingly challenging 2D natural convection problem that serves as a surrogate for more difficult 3D buoyancy-driven flows encountered in a variety of mechanical and nuclear engineering applications. The geometry is the tilted slot configuration of Figure 1 and the governing equations are the unsteady Boussinesq equations. The problem is characterized by four parameters, the Rayleigh number, Ra, the Prandtl number, Pr, the slot angle with respect to gravity, θ_g, and the aspect-ratio, Γ. However, for small aspect ratios, $\Gamma \leq 8$, we find the flow is rather simple. Hence in this work, we focus on the more challenging case of Γ = 40 with (Pr, Ra, θ_g) as the parameter space.

Figure 1A illustrates the problem configuration. The aspect ratio is defined as $\Gamma = L/H$, where H is the width of the slot and L is the height of the slot. We take $H = 1$ and $L = 40$ throughout the study. With $\theta_g = 0^\circ$ the flow corresponds to standard Rayleigh-Bénard convection, 90° corresponds to vertical slot convection, and $\theta_g = 180^\circ$ leads to a pure conduction solution with the cold side on the bottom of the horizontal slot. The Rayleigh and Prandtl numbers are

$$\mathrm{Ra} = \frac{\rho\beta\Delta T H^3 g}{\nu\alpha}, \quad \mathrm{Pr} = \frac{\nu}{\alpha}, \tag{1}$$

where ρ is the fluid density, β is the thermal expansion coefficient, $\Delta T = 2$ is the wall-to-wall temperature difference, H is the gap width, g is the gravitational acceleration, ν is the kinematic viscosity, and α is the thermal diffusivity. Figure 1B illustrates representative mean (or steady) temperature fields for 19 uniformly-spaced $\theta_g \in [0^\circ, 180^\circ]$ at $\mathrm{Ra} = 10^4$ and $\mathrm{Pr} = 7.2$. Figure 1C shows the mean temperature field for 6 uniformly-spaced $\mathrm{Ra} \in [2 \times 10^5, 7 \times 10^5]$ at $\theta_g = 90^\circ$ and $\mathrm{Pr} = 7.2$. The configuration has many interesting applications. For example, it represents energy-efficient double-glazed windows, in which the sealed air gap between the two panes acts as an added layer of insulation. Finding the optimum angle θ_g that enhances the heat transfer is an important question. Convection in a tilted fluid layer is also of meteorological and oceanographic interest. More information on the impact of θ_g and Γ on heat transport and flow organization for this configuration can be found in [12].

There has been significant work on pMOR development for the *steady* Boussinesq equations, including rigorous error estimation [10, 11, 13–15]. For pMOR, the steady problem is easier than the unsteady problem for several reasons: 1) rigorous error estimates are usually achievable, 2) there is often a well defined attractor, and 3) no temporal instability needs to be considered. Once the problem becomes unsteady many open research issues remain. To our knowledge, there are few pMOR works addressing the unsteady parameterized Boussinesq equations. In [16], the authors develop rigorous a posteriori error bounds applied to a 2D Rayleigh-Bénard problem parameterized with Gr and θ_g. However, due to exponential instability in time, the rigor is not for very high Gr and large final times. In [17], the authors overcame the high Gr issue by considering a space-time formulation which enabled effective long-time certification of a reduced basis approximation of noncoercive PDEs. However, the approach is limited due to large offline computational effort since only one snapshot is generated from one FOM solve due to the formulation. For example, to cover the parameter space, 125 FOMs are solved during the offline in their case.

Fick *et al.* [9] developed a POD-*h*Greedy pMOR to study challenging incompressible flow using a time-averaged error indicator. The authors showed that the error indicator is highly-correlated with the error in mean flow prediction and can be efficiently computed through an offline/online strategy. We view the methodology as having high potential for routine analysis and design of turbulent flows that are characteristic of thermal hydraulic systems. Hence, we explore that approach here by extending the time-averaged error indicator to accommodate the energy equation. In previous work [18] on ROM stabilization and turbulent thermal transport problems, we investigated the performance of pMOR with constrained stabilization [9], and Leray regularization [19]. Here, we extend the error indicator to support Leray regularization. For each approach, we assess the performance through the mean flow and QOIs including, mean Nusselt number (Nu), standard deviation in Nu, mean temperature fluctuation and mean turbulent kinetic energy (TKE)[1].

Even though our 2D model problem generates only laminar flows, pMOR is quite challenging in this application for several reasons: 1) there is bifurcation in θ_g parameter space; 2) the solution can be multivalued, even at steady state; and 3) the solution exhibits spatio-temporal chaos at several points in the parameter space. As our initial efforts happened to be focused in one of the spatio-temporal chaos regimes, we decided to map out a larger space to identify where pMOR could succeed, where it would have difficulty, and where it *a priori* could not succeed. Table 1 reflects a broad range of flow regimes identified inside the high-aspect ratio slot from hundreds of FOM simulations conducted at multiple Ra, θ_g with $\mathrm{Pr} = 0.07, 0.71, 7.2$. We categorize the flow into six types: 1) motionless, 2) steady, 3) periodic, 4) quasi-periodic, 5) chaotic and 6) spatio-temporal chaotic. We identify the flow regimes by examining the (mean) solution field and the energy and Nu histories. Such analysis can readily distinguish the motionless, steady and periodic flow cases. Even though the energy and Nu analysis seem to be a reliable way to distinguish the quasi-periodic and chaotic flow, it is only a heuristic—a more rigorous analysis is through computing the power spectrum of Nu or energy [20]. Tools such as Lyapunov exponent and fractal dimension are probably the most widely used diagnostic for chaotic systems [21, 22]. The first five types of low have consistent mean flow in differing time windows, each averaged over 500 convective time units (CTUs). We define a flow to be spatio-temporal chaotic [23] if its mean solution is not consistent in at least three different time windows. This type of flow has strong irregularities in both space and time and has been observed in Rayleigh Bénard convection and in other complex dynamical systems [24]. To characterize spatio-temporal chaos, one

1 Technically, since these flows are not turbulent the TKE should be referred to as *velocity variance*. Because the is more widely used and the mathematical formulation is the same in either case, we prefer to use the more widely recognized appellation, TKE.

TABLE 1 Distribution of six flow types with Ra and θ_g at Pr = 0.07, 0.71, 7.2.

flow \Pr	Pr = 0.07	Pr = 0.71	Pr = 7.2
motionless	$Ra < 1.1 \times 10^3$, $\theta_g = 0^\circ$	$Ra < 8.75 \times 10^2$, $\theta_g = 0^\circ$	$Ra < 9 \times 10^2$, $\theta_g = 0^\circ$
steady	$Ra = 10^3$, $\theta_g = 90^\circ$ $Ra = 10^4$, $\theta_g \in [0^\circ, 40^\circ] \cup [170^\circ, 180^\circ]$	$Ra = 10^3$, $\theta_g \in [0^\circ, 180^\circ]$ $Ra = 10^4$, $\theta_g \in [0^\circ, 30^\circ] \cup [70^\circ, 80^\circ] \cup [120^\circ, 180^\circ]$ $Ra = 1.5 \times 10^4$, $\theta_g = 90^\circ$	$Ra = 10^3$, $\theta_g \in [0^\circ, 180^\circ]$ $Ra = 10^4$, $\theta_g \in [0^\circ, 180^\circ] \backslash \{30^\circ\}$ $Ra \in [2 \times 10^4, 10^5]$, $\theta_g = 90^\circ$ $Ra = 8 \times 10^4$, $\theta_g \in [80^\circ, 180^\circ]$ $Ra = 3 \times 10^5$, $\theta_g \in [130^\circ, 180^\circ]$
periodic	$Ra \in [5 \times 10^3, 7 \times 10^3]$, $\theta_g = 90^\circ$	$Ra = 10^4, 1.75 \times 10^4$, $\theta_g = 90^\circ$	N\A
quasi-periodic	$Ra = 10^4$, $\theta_g \in [50^\circ, 110^\circ]$ $Ra = 8 \times 10^3, 9 \times 10^3, 2 \times 10^4$, $\theta_g = 90^\circ$	$Ra \in [1.8 \times 10^4, 2 \times 10^4]$, $\theta_g = 90^\circ$	N\A
chaotic	$Ra = 10^4$, $\theta_g \in [120^\circ, 160^\circ]$ $Ra \in [6 \times 10^4, 1.5 \times 10^5]$, $\theta_g = 90^\circ$	N\A	$Ra = 8 \times 10^4$, $\theta_g \in [40^\circ, 70^\circ]$ $Ra = 3 \times 10^5$, $\theta_g \in [50^\circ, 120^\circ]$ $Ra \in [2 \times 10^5, 8 \times 10^5]$, $\theta_g = 90^\circ$
spatio-temporal chaotic	$Ra \in [3 \times 10^4, 5 \times 10^4]$, $\theta_g = 90^\circ$	$Ra = 10^4$, $\theta_g \in [40^\circ, 60^\circ] \cup [100^\circ, 110^\circ]$ $Ra = 1.25 \times 10^4$, $\theta_g = 90^\circ$ $Ra \in [2.05 \times 10^4, 3 \times 10^5]$, $\theta_g = 90^\circ$	$Ra = 10^4$, $\theta_g = 30^\circ$ $Ra = 8 \times 10^4$, $\theta_g \in [0^\circ, 30^\circ]$ $Ra = 3 \times 10^5$, $\theta_g \in [0^\circ, 40^\circ]$ $Ra \in [2 \times 10^5, 8 \times 10^5]$, $\theta_g = 0^\circ, 10^\circ$

could also consider Lyapunov exponents at each grid point. A detail analysis of spatio-temporal chaos is beyond the scope of this paper. A comprehensive review on this topic can be found in [25].

In the present work, we start with solution reproduction problem, which represents the first step towards the development of a ROM for the parametric problem. We study the reproduction capability of the FOM, ROM and ROM with stabilization for the six types of flow reported in Table 1. We report only the cases of chaotic and spatio-temporal chaotic flow. Notice that it has been reported that ROM can often capture the first five types of flow accurately, in some cases, with the need of stabilization methods. However, for the spatio-temporal chaotic flow, to our knowledge, it has not been studied. We believe this is the first work investigate ROM's reproducibility of spatio-temporal chaotic flow.

The pMOR development is broken into several parametric problems. We start with a problem at $Ra = 10^3$ and Pr = 7.2 in which the solution is steady for all θ_g but nonetheless exhibits a bifurcation at $\theta_g = 20^\circ$[2]. We find either *h*- or *p*-Greedy with the error indicator based on the dual norm of the residual is able to drive pMOR successfully[3]. We next consider two sets of parametric problems: 1) problem parameterized with θ_g at higher *Ra*. 2) Problem parameterized with Ra. In the first set, similar to the steady case, a bifurcation is observed and the solution space is a blend of steady and unsteady solutions. In the second set, no bifurcation is observed and the solutions are all unsteady.

By proceeding in this manner, we are able to isolate several difficulties and eventually come up with an important observation for the pMOR: Accurate prediction ($< 10\%$) with pMOR is achievable if the solution in the parametric space is either only chaotic or the spatio-temporal chaos is not significant, regardless a bifurcation exists or not. Once the spatio-temporal chaos becomes significant, the performance of the pMOR deteriorates and the maximum errors of the mean flow and QOIs are dominated by the flow chaos.

The paper is organized as follows. In Section 2, we introduce the model problem and governing equations. In Section 2.1, we introduce the Galerkin formulation for the FOM. The ROM, as well as Leray regularization is introduced in 2.2. In Section 3, we consider the solution reproduction problem and assess the numerical performance. The parametric problem is discussed starting from Section 4. We first introduce POD-*h*Greedy algorithms in Section 4.1 with some remarks on applying POD-*p*Greedy to this model problem at the end of the section. We then introduce the time-averaged error indicator with thermal extension in Section 4.2. A straight-forward integration with Leray regularization is also shown in the same section. In Sections 4.3 and 4.4, we present the pMOR results with θ_g variation and Ra variation. In Section 5, we discuss the spatio-temporal chaos and multiple states issues found in this model problem. Finally, we conclude the paper in Section 6.

2 The ROM coefficients in the steady problems are typically found through a Newton minimization over the POD approximation space [10, 11].

3 The pMOR greedy strategy uses the maximal indicated error among the parametric training set to select the next anchor point. *p*-Greedy combines basis functions from FOMs at different anchor points to form an enriched approximation space; *h*-greedy builds an independent ROM for each anchor point Ngoc Cuong et al. [10].

2 A parametrized natural convection problem

We start with the Boussinesq equations for buoyancy-driven flow [26],

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = \nu\nabla^2\mathbf{u} + T\mathbf{g}(\theta_g), \qquad \nabla\cdot\mathbf{u} = 0, \quad (2)$$

$$\frac{\partial T}{\partial t} + (\mathbf{u}\cdot\nabla)T = \alpha\nabla^2 T, \quad (3)$$

where $\mathbf{u}$ is the velocity, p is the pressure, T is the temperature and $\mathbf{g}(\theta_g)$ is the unit vector represents the direction of the buoyancy force and it is defined by $\mathbf{g}(\theta_g) = \cos(\theta_g)\hat{\imath} + \sin(\theta_g)\hat{\jmath}$, with θ_g the angle of the slot with respect to the gravity. The velocity boundary conditions are no-slip ($\mathbf{u} = 0$). The temperature boundary conditions are no-flux (insulated) on the top and bottom, heated ($T = 1$) on the left wall, and cooled ($T = -1$) on the right wall. The initial conditions are $\mathbf{u} = 0$ and $T = 0$.

In our non-dimensional setting, we set $\nu = (\mathrm{Pr}/\mathrm{Ra})^{1/2}$ and $\alpha = (\mathrm{Pr}\,\mathrm{Ra})^{-1/2}$. The Rayleigh number, $\mathrm{Ra} = \rho\beta g H^3\Delta T/(\nu\alpha)$, represents the ratio of buoyancy force to thermal and momentum diffusive force The Prandtl number $\mathrm{Pr} = \nu/\alpha$, reflects the relative importance of momentum diffusivity compared to thermal diffusivity. With this nondimensionalization the characteristic velocity is $U_c = \sqrt{\beta g H \Delta T}$, the characteristic length is the slot width H, and the reference time is $t_r = H/U_c$. The temperature is made dimensionless by subtracting the temperature on the right wall and scaling with $\Delta T = 2$. We note that U_c is sometimes referred to as the "free-fall" velocity, indicating that one might expect $\|\mathbf{u}\| \approx 1$, with only a weak dependence on Ra. While that expectation is realized for $\theta_g = 0$, we in fact see much larger velocities ($\|\mathbf{u}\| \approx 40$) for $\theta_g = 90^\circ$ because the domain height $L = 40H$ in that case.

For unsteady problems, the QOIs are the mean flow, mean Nu, standard deviation in Nu, mean TKE and mean temperature fluctuation. The symbol $\langle\cdot\rangle$ is used to indicate a time-averaged quantity. The mean velocity and temperature field are defined as:

$$\langle\mathbf{u}\rangle = \frac{1}{J-J_0}\sum_{j=J_0+1}^{J}\mathbf{u}(t^j), \qquad \langle T\rangle = \frac{1}{J-J_0}\sum_{j=J_0+1}^{J}T(t^j), \quad (4)$$

with $t^j = j\Delta t$ and Δt being the time step. The selection of J_0 is based on when the solution reaches it statistically steady state. The mean quantities are then averaged over 500 CTUs, with the time scale defined above. The instantaneous Nusselt number is defined as

$$\mathrm{Nu}(t) = \frac{q''_w}{k(\Delta T)/H}, \quad (5)$$

with $q''_w = -\int_{\partial\Omega_h} k\nabla T\cdot\hat{\mathbf{n}}\,dS$ being the integrated heat flux on the heated wall, $\partial\Omega_h$. The mean Nu and the its standard deviation are then defined as

$$\langle\mathrm{Nu}\rangle := \frac{1}{J-J_0}\sum_{j=J_0+1}^{J}\mathrm{Nu}(t^j),$$
$$\mathrm{Std}(\mathrm{Nu}) := \sqrt{\frac{1}{J-J_0}\sum_{j=J_0+1}^{J}(\mathrm{Nu}(t^j)-\langle\mathrm{Nu}\rangle)}. \quad (6)$$

The mean TKE and mean temperature fluctuation are defined as

$$\langle\mathrm{TKE}\rangle = \frac{1}{2(J-J_0)}\sum_{j=J_0+1}^{J}\|\mathbf{u}(t^j)-\langle\mathbf{u}\rangle\|^2_{L^2},$$
$$\langle T_{\mathrm{fluc}}\rangle = \frac{1}{J-J_0}\sum_{j=J_0+1}^{J}\|T(t^j)-\langle T\rangle\|^2_{L^2}. \quad (7)$$

For steady problems, the QOIs are simply the steady solutions to Eqs. 2 and 3 and the corresponding Nu using Eq. 5.

2.1 Galerkin formulation for the full-order model

The FOM is constructed through the spectral element method (SEM) and the P_q–P_{q-2} velocity-pressure coupling [27], where the velocity is represented as a tensor-product Lagrange polynomial of degree q in the reference element $\hat{\Omega} := [-1,1]^2$ while the pressure is of degree $q-2$. The solution in $\Omega = \bigcup_e\Omega^e$ consists of local representations of $\mathbf{u}$, p, and T that are mapped from $\hat{\Omega}$ to Ω^e for each element, $e = 1,\ldots,E$. In the current FOMs for the slot problem we use $E = 516$ elements (an array of 6×86 in the $H\times L$ directions), of order $q = 9$, for a total of $\mathcal{N}\approx 42000$ grid points. The FOM simulations are performed using the open-source code Nek5000 [28].

For any $\mathbf{u}(\mathbf{x}, t)$, we have a corresponding vector of basis coefficients $\underline{\mathbf{u}} = [\mathbf{u}_1\ldots\mathbf{u}_{\mathcal{N}}]^T$ such that

$$\mathbf{u}(\mathbf{x},t) = \sum_{j=1}^{\mathcal{N}}\mathbf{u}_j(t)\phi_j(\mathbf{x}) \in \mathbf{X}_0^{\mathcal{N}} \subset \{\mathcal{H}\}_0^1, \quad (8)$$

with $\phi_j(\mathbf{x})$ the underlying spectral element basis functions spanning the FOM approximation space, $\mathbf{X}_0^{\mathcal{N}}$. Because the SEM is nodal-based, each $\mathbf{u}_j(t)$ represents the two velocity components at grid point $\mathbf{x}_j$ in the spectral element mesh at time t. Similarly, the temperature is given by

$$T(\mathbf{x},t) = \sum_{j=1}^{\tilde{\mathcal{N}}}T_j(t)\phi_j(\mathbf{x}) \in X_0^{\mathcal{N}} \subset \mathcal{H}_0^1. \quad (9)$$

Here, $\mathcal{H}^1$ is the set of square-integrable functions on Ω whose gradient is also square-integrable and $X^{\mathcal{N}}\subset\mathcal{H}^1$ is the finite dimensional SEM approximation space spanned by $\{\phi_j(\mathbf{x})\}$. $\mathcal{H}_0^1$ is the set of functions in $\mathcal{H}^1$ that vanish wherever Dirichlet conditions associated with Eq. 3 are applied on the domain boundary $\partial\Omega$ and $\mathcal{H}_b^1$ is the set of functions in $\mathcal{H}^1$ that satisfy the prescribed Dirichlet conditions for temperature. Bold-face

indicates that the space is spanned by vector-valued functions having d components ($d = 2$ or 3) and, in the case of $\mathbf{X}_0^{\mathcal{N}} \subset \{\mathcal{H}\}_0^1$, that the functions vanish where Dirichlet conditions are applied for Eq. 2. The pressure p is in $Y^{\mathcal{N}} \subset \mathcal{L}^2(\Omega)$, which is the space of piecewise continuous functions on Ω such that $\int_\Omega p^2\, d\mathbf{x} < \infty$. For convenience, we denote $\bar{\mathbf{Z}}^{\mathcal{N}} = (\mathbf{X}^{\mathcal{N}}, Y^{\mathcal{N}}, X^{\mathcal{N}})$ as the collection of the relevant finite-dimensional spaces and will add a subscript 0 or b where required to explicitly indicate the imposed boundary conditions.

Both the FOM and ROM are cast within the same Galerkin framework. To begin, we introduce several inner products for elements in the FOM space, $\bar{\mathbf{Z}}^{\mathcal{N}}$. For any pair of scalar fields $(p, q) \in \mathcal{L}^2(\Omega)$ and d-dimensional vector fields, $\mathbf{v}(\mathbf{x}) = [v_1(\mathbf{x}) \ldots v_d(\mathbf{x})]$, $\mathbf{u}(\mathbf{x}) = [u_1(\mathbf{x}) \ldots u_d(\mathbf{x})]$ whose components are also in $\mathcal{L}^2$, let

$$(q, p) := \int_\Omega q\, p\, d\mathbf{x}, \qquad (\mathbf{v}, \mathbf{u}) := \int_\Omega (v_1 u_1 + \cdots + v_d u_d)\, d\mathbf{x}. \tag{10}$$

Further, for $S, T \in X^{\mathcal{N}}$ and $\mathbf{v}, \mathbf{u} \in \mathbf{X}^{\mathcal{N}}$, let

$$a(S, T) := (\nabla S, \nabla T), \qquad a(\mathbf{v}, \mathbf{u}) := (\nabla \mathbf{v}, \nabla \mathbf{u}), \tag{11}$$

$$c(S, \mathbf{u}, T) := (S, \mathbf{u} \cdot \nabla T), \qquad c(\mathbf{v}, \mathbf{u}, \mathbf{w}) := (\mathbf{v}, \mathbf{u} \cdot \nabla \mathbf{w}). \tag{12}$$

For the FOM, we consider the (semi-discrete) weak form of Eqs. 2 and 3 [29], *Find* $(\bar{\mathbf{u}}, p, \bar{T}) \in \bar{\mathbf{Z}}_b^{\mathcal{N}}$ *such that, for all* $(\mathbf{v}, q, S) \in \bar{\mathbf{Z}}_0^{\mathcal{N}}$,

$$\left(\mathbf{v}, \frac{\partial \bar{\mathbf{u}}}{\partial t}\right) + \nu\, a(\mathbf{v}, \bar{\mathbf{u}}) - (\nabla \cdot \mathbf{v}, p) = -c(\mathbf{v}, \bar{\mathbf{u}}, \bar{\mathbf{u}}) + \left(\mathbf{v}, \mathbf{g}(\theta_g)\bar{T}\right), \tag{13}$$

$$-(q, \nabla \cdot \bar{\mathbf{u}}) = 0, \tag{14}$$

$$\left(S, \frac{\partial \bar{T}}{\partial t}\right) + \alpha\, a(S, \bar{T}) = -c(S, \bar{\mathbf{u}}, \bar{T}). \tag{15}$$

Here, we have introduced $\bar{\mathbf{u}} = \mathbf{u} + \mathbf{u}_0(\mathbf{x})$ and $\bar{T} = T + T_0(\mathbf{x})$ as functions that have been augmented by (potentially trivial) lifting functions, $\mathbf{u}_0$ and T_0, which are functions of space only. If these functions satisfy the (time-independent) boundary conditions, then one can account for inhomogeneous boundary conditions by moving them to the right-hand side. In the case of the ROM, the lifting functions can also provide an initial approximation to the solution. In the sequel, our principal unknowns will be $\mathbf{u}$ and T.

Following [30], we consider a semi-implicit scheme BDFk/EXTk to discretize Eqs. 13–15 in time; kth-order backward differencing (BDFk) is used for the time-derivative term, kth-order extrapolation (EXTk) is used for the advection and buoyancy terms and implicit treatment on the dissipation terms. As discussed in [30], $k = 3$ is used to ensure the imaginary eigenvalues associated with skew-symmetric advection operator are within the stability region of the BDFk/EXTk time-stepper. Denoting the solution at time $t^n = \Delta t \cdot n$ as $(\bar{\mathbf{u}}^n, p^n, \bar{T}^n)$, the full discretization of the FOM reads *Find* $(\mathbf{u}^n, p^n, T^n) \in \bar{\mathbf{Z}}_0^{\mathcal{N}}$ *such that, for all* $(\mathbf{v}, q, S) \in \bar{\mathbf{Z}}_0^{\mathcal{N}}$,

$$\frac{\beta_0}{\Delta t}(\mathbf{v}, \mathbf{u}^n) + \nu\, a(\mathbf{v}, \mathbf{u}^n) - (\nabla \cdot \mathbf{v}, p^n) = (\mathbf{v}, \mathbf{f}^n), \tag{16}$$

$$-(q, \nabla \cdot \mathbf{u}^n) = (q, \nabla \cdot \mathbf{u}_0), \tag{17}$$

$$\frac{\beta_0}{\Delta t}(S, T^n) + \alpha\, a(S, T^n) = (S, Q^n). \tag{18}$$

Equations 16–18 represent a linear unsteady Stokes plus unsteady heat equation to be solved at each time-step t^n. The inhomogeneous terms comprise the BDF, advection, buoyancy and lifting terms

$$(\mathbf{v}, \mathbf{f}^n) := -\sum_{s=1}^{k}\left[\frac{\beta_s}{\Delta t}(\mathbf{v}, \mathbf{u}^{n-s}) + \alpha_s\left(c(\mathbf{v}, \bar{\mathbf{u}}^{n-s}, \bar{\mathbf{u}}^{n-s}) - \left(\mathbf{v}, \mathbf{g}(\theta_g)\bar{T}^{n-s}\right)\right)\right] - \nu\, a(\mathbf{v}, \mathbf{u}_0), \tag{19}$$

$$(S, Q^n) := -\sum_{s=1}^{k}\left[\frac{\beta_s}{\Delta t}(S, T^{n-s}) + \alpha_s\, c(S, \bar{\mathbf{u}}^{n-s}, \bar{T}^{n-s})\right] - \alpha\, a(S, T_0). \tag{20}$$

Here, the β_ss and α_ss are the respective sth-order BDF and extrapolation coefficients for the BDFs/EXTs time-stepper [30]. Note that the right-hand side of Eq. 17 will be zero if $\mathbf{u}_0$ is divergence free or at least satisfies the weak divergence-free condition Eq. 14.

Under the assumption that $\nabla \cdot \mathbf{u}_0 = 0$, the compact matrix form [27, 31, 32] for Eqs. 16–20 is

$$\begin{bmatrix} \mathbf{H} & -\mathbf{D}^T \\ -\mathbf{D} & 0 \end{bmatrix}\begin{pmatrix} \underline{\mathbf{u}}^n \\ \underline{p}^n \end{pmatrix} = \begin{pmatrix} \underline{\hat{\mathbf{f}}}(\bar{\mathbf{u}}^n, \bar{T}^n; \theta_g) \\ 0 \end{pmatrix}, \tag{21}$$

$$H_\alpha \underline{T}^n = \underline{\hat{Q}}(\bar{\mathbf{u}}^n, \bar{T}^n). \tag{22}$$

Here, $\underline{\mathbf{u}}^n$, $\underline{p}^n$, and $\underline{T}^n$ are the vectors of spectral element basis coefficients. The corresponding block matrices are

$$\mathbf{H} = \begin{bmatrix} H_\nu & \\ & H_\nu \end{bmatrix}, \qquad \mathbf{D} = [D_1 \;\; D_2], \tag{23}$$

with $H_\nu = \frac{\beta_0}{\Delta t}M + \nu A$ and $H_\alpha = \frac{\beta_0}{\Delta t}M + \alpha A$, with matrices M and A defined below. The velocity data vectors are $\underline{\hat{\mathbf{f}}}(\bar{\mathbf{u}}^n, \bar{T}^n; \theta_g) = [\underline{\hat{f}}_1\; \underline{\hat{f}}_2]^T$, with

$$\underline{\hat{f}}_m := -\sum_{s=1}^{k}\left[\frac{\beta_s}{\Delta t}M\underline{u}_m^{n-s} + \alpha_s\left(C(\underline{\bar{\mathbf{u}}}^{n-s})\underline{\bar{u}}_m^{n-s} - g_m M\underline{\bar{T}}^{n-s}\right)\right] - \nu A\underline{u}_{m,0}, \qquad m = 1, 2. \tag{24}$$

where $g_1 = \cos(\theta_g)$ and $g_2 = \sin(\theta_g)$ represent the parametric forcing. The thermal load in Eq. 22 is

$$\underline{\hat{Q}}(\bar{\mathbf{u}}^n, \bar{T}^n) := -\sum_{s=1}^{k}\left[\frac{\beta_s}{\Delta t}M\underline{\bar{T}}^{n-s} + \alpha_s C(\underline{\bar{\mathbf{u}}}^{n-s})\underline{\bar{T}}^{n-s}\right] - \alpha A\underline{T}_0. \tag{25}$$

Entries of the respective stiffness, mass, convection, and gradient matrices are

$$A_{ij} = \int_\Omega \nabla\phi_i \nabla\phi_j \, d\mathbf{x}, \tag{26}$$

$$M_{ij} = \int_\Omega \phi_i \phi_j \, d\mathbf{x}, \tag{27}$$

$$C_{ij}(\mathbf{w}) = \int_\Omega \phi_i \cdot (\mathbf{w}\cdot\nabla)\phi_j \, d\mathbf{x}, \tag{28}$$

$$D_{m,ij} = \int_\Omega \psi_i \frac{\partial \phi_j}{\partial x_m} \, d\mathbf{x}, \quad m = 1, 2. \tag{29}$$

Note that $\{\phi_i(\mathbf{x})\}$ forms the spectral element velocity/temperature basis while $\{\psi_i(\mathbf{x})\}$ constitutes the pressure basis.

2.2 Galerkin formulation for the reduced-order model

Within the Galerkin framework of the preceding section it is relatively straightforward to develop a ROM. One defines a set of functions $\boldsymbol{\zeta}_j(\mathbf{x}) \in \mathbf{X}^N \subset \mathbf{X}^{\mathcal{N}}$, $\theta_j(\mathbf{x}) \in X^N \subset X^{\mathcal{N}}$ such that the coarse-space (ROM) solution, is expressed as

$$\bar{\mathbf{u}}_c(\mathbf{x}) = \sum_{j=0}^{N} \boldsymbol{\zeta}_j(\mathbf{x}) u_{c,j}, \qquad \bar{T}_c(\mathbf{x}) = \sum_{j=0}^{N} \theta_j(\mathbf{x}) T_{c,j}. \tag{30}$$

For the ROM, we insert the expansions Eq. 30 into Eqs. 13–15 and require equality for all $(\mathbf{v}, S)$ in $\mathbf{Z}_0^N$. In order to set the boundary conditions, we have augmented the trial (approximation) spaces $\mathbf{X}^N$ and X^N with the lifting function $\boldsymbol{\zeta}_0 := \mathbf{u}_0$ and $\theta_0 := T_0$. The corresponding test spaces, $\mathbf{X}_0^N := \{\boldsymbol{\zeta}_j\}_{j=1}^N$ and $X_0^N := \{\theta_j\}_{j=1}^N$, satisfying homogeneous boundary conditions, as is standard for Galerkin formulation. The coarse space $\mathbf{Z}_0^N := (\mathbf{X}_0^N, X_0^N)$ is typically based on a linear combination of full spectral element solutions of Eqs. 21 and 22, such as snapshots at certain time-points, t^n, or solutions at various parametric values. Under these conditions and with a carefully chosen $\mathbf{u}_0$, $\mathbf{X}^N$ is a set of velocity-space functions that are (weakly) divergence-free and the pressure terms drop out of Eqs. 13 and 14. We also note that $\boldsymbol{\zeta}_j$ and θ_j are *modal*, not nodal, basis functions. In this work, we consider proper orthogonal decomposition (POD) to construct the reduced-basis. The N-dimensional POD-space is the space that minimizes the averaged distance between the snapshot set and the N-dimensional subspace of the snapshots set in the H^1 semi-norm. Further details of the POD basis selection are provided in [18] and references therein.

The matrix form for the ROM is readily derived by constructing a pair of rectangular basis matrices, $\mathbf{B}$ and B, having entries

$$\mathbf{B}_{ij} = \boldsymbol{\zeta}_j(\mathbf{x}_i), \qquad B_{ij} = \theta_j(\mathbf{x}_i), \tag{31}$$

where the $\mathbf{x}_i$s are the spectral element nodal points. The coarse-system matrices are $H_{c,\nu} = \mathbf{B}^T \mathbf{H} \mathbf{B}$ and $H_{c,\alpha} = B^T H_\alpha B$ and the governing system for the ROM becomes

$$H_{c,\nu}\, \underline{\hat{u}}_c^{\,n} = \mathbf{B}^T \hat{\mathbf{f}}\left(\bar{\mathbf{u}}_c^n, \bar{T}_c^n; \theta_g\right), \qquad H_{c,\alpha}\, \underline{\hat{T}}_c^{\,n} = B^T \underline{\hat{Q}}\left(\bar{\mathbf{u}}_c^n, \bar{T}_c^n\right). \tag{32}$$

We refer Eq. 32 as Galerkin ROM (G-ROM) throughout the paper. The ROM coefficient vectors, $\underline{\hat{u}}_c^{\,n} = [u_{c,1}^n \ldots u_{c,N}^n]^T$, $\underline{\hat{T}}_c^{\,n} = [T_{c,1}^n \ldots T_{c,N}^n]^T$, are determined by solving the $2\ N \times N$ linear systems. Note that the coefficients for the lifting functions are prescribed: $u_{c,0} = T_{c,0} = 1$. The initial coefficients for the ROM are obtained by projecting the initial condition onto the coarse space $\mathbf{Z}_0^N$ with the H^1 semi-norm,

$$\underline{\hat{u}}_c^{\,0} = \mathbf{B}^T \mathbf{A} \underline{\mathbf{u}}^0, \qquad \underline{\hat{T}}_c^{\,0} = B^T A \underline{T}^0, \tag{33}$$

which follows from the fact that the columns of $\mathbf{B}$ and B are, respectively, $\mathbf{A}$- and A-orthonormal, where $\mathbf{A}$ = block-diagonal(A). To recover the spectral element representation, we simply prolong the N-length vectors $\underline{\hat{u}}_c^{\,n}$ and $\underline{\hat{T}}_c^{\,n}$ with the set of basis functions and add it with the lifting functions $\mathbf{u}_0$ and T_0

$$\underline{\bar{\mathbf{u}}}_c^{\,n} = \mathbf{B}\underline{\hat{u}}_c^{\,n} + \underline{\mathbf{u}}_0, \qquad \underline{\bar{T}}_c^{\,n} = B\underline{\hat{T}}_c^{\,n} + \underline{T}_0. \tag{34}$$

The functional representations, $\bar{\mathbf{u}}_c^n(\mathbf{x})$ and $\bar{T}_c^n(\mathbf{x})$, are then obtained from Eqs. 8 and 9.

Next we consider the Galerkin ROM with Leray regularization using a spatial filter, following ideas presented in [19]. The approach simply requires regularizing the *advecting* field in the Navier-Stokes equations and energy equation through a low-pass filter function F (i.e., $\bar{\mathbf{u}}_{\text{filtered}} = F(\bar{\mathbf{u}})$). As noted in [33, 34], a small amount of regularization is sufficient to make gains in proving existence and uniqueness of the solution to the Navier-Stokes equations. Thus, Leray regularization is of interest both from a numerical (and physical) stabilization viewpoint and from a theoretical perspective.

The formulation of G-ROM with Leray regularization is shown in Eq. 35 and the only difference comparing to G-ROM Eq. 32 are the velocity data and thermal load.

$$H_{c,\nu}\, \underline{\hat{u}}_c^{\,n} = \mathbf{B}^T \hat{\mathbf{f}}_{\text{filtered}}\left(\bar{\mathbf{u}}_c^n, \bar{T}_c^n; \theta_g\right), \qquad H_{c,\alpha}\, \underline{\hat{T}}_c^{\,n} = B^T \underline{\hat{Q}}_{\text{filtered}}\left(\bar{\mathbf{u}}_c^n, \bar{T}_c^n\right), \tag{35}$$

where $\hat{\mathbf{f}}_{\text{filtered}}(\bar{\mathbf{u}}^n, \bar{T}^n; \theta_g) = [\underline{\hat{f}}_1\ \underline{\hat{f}}_2]_{\text{filtered}}^T$. We use the subscript filtered to denote the advecting field in the velocity data and thermal load are being filtered,

$$\underline{\hat{f}}_{\text{filtered},m} := -\sum_{s=1}^{k}\left[\frac{\beta_s}{\Delta t} M \underline{u}_m^{n-s} + \alpha_s \left(C\left(\underline{\bar{\mathbf{u}}}_{\text{filtered}}^{n-s}\right)\underline{\bar{u}}_m^{n-s} - g_m M \underline{\bar{T}}^{n-s}\right)\right] - \nu A \underline{u}_{m,0}, \quad m = 1, 2. \tag{36}$$

and

$$\underline{\widehat{Q}}_{\text{filtered}}(\bar{\mathbf{u}}^n, \bar{T}^n): = -\sum_{s=1}^{k}\left[\frac{\beta_s}{\Delta t} M \underline{\bar{T}}^{n-s} + \alpha_s C\left(\underline{\bar{\mathbf{u}}}^{n-s}_{\text{filtered}}\right)\underline{\bar{T}}^{n-s}\right] - \alpha A \underline{T}_0. \tag{37}$$

In this work, we will be focusing on a PDE- (or differential-) based filter, which is characterized by a filter width, δ [35]. Following [19, 36], such filters are developed in a POD context as follows: *Find* $\bar{\mathbf{u}}_{\text{c,filter}} \in \mathbf{X}^N$ *such that*

$$\left((I - \delta^2\nabla^2)\bar{\mathbf{u}}^n_{\text{c,filter}}, \boldsymbol{\zeta}_j\right) = \left(\bar{\mathbf{u}}_{\text{c}}, \boldsymbol{\zeta}_j\right), \quad \forall\ j = 1, \ldots, N. \tag{38}$$

Besides the differential filter, one could also consider a more economic spatial filter, namely, a POD-projection (Proj) filter as discussed in [19]. In this case, one simply truncates the higher POD mode contributions when constructing $\bar{\mathbf{u}}^n_{\text{c,filtered}}$, just as one would do in a Fourier reconstruction [18].

Besides G-ROM and LDF-ROM, we also consider the constrained-evolution stabilization introduced in [9]. The idea behind this approach is to use information from the snapshot set to establish *a priori* limits on the ROM coefficients $\underline{\widehat{u}}_{\text{c}}$ by replacing Eq. 32 with a constrained minimization problem. At each time-step, the coefficients satisfy

$$\underline{\widehat{u}}^n_{\text{c}} = \arg\min_{\underline{u}_{\text{c}} \in \mathcal{R}^N} \frac{1}{2}\|H_{\text{c},y}\underline{\widehat{u}}_{\text{c}} - \mathbf{B}^T \underline{\widehat{\mathbf{f}}}\left(\bar{\mathbf{u}}^n_{\text{c}}, \bar{T}^n_{\text{c}}; \theta_g\right)\|^2_{H^{-1}}, \quad \text{s.t. } m_j \le u^n_j \le M_j. \tag{39}$$

where the constraints m_j and M_j on the basis coefficients $u^n_{\text{c},j}$, $j = 1, \ldots, N$ are derived from the observation snapshot set. A constrained minimization problem for the thermal ROM coefficients, $\underline{\widehat{T}}_{\text{c}}$, is derived similarly. We denote Eq. 39 as C-ROM. Further implementation details can be found in [9, 18].

3 The solution reproduction problem

In this section, we consider the solution reproduction problems for Ra = 2×10^5, 7×10^5 at Pr = 7.2 and $\theta_g = 90°$, where the solutions are chaotic. We assess the performance of G-ROM, C-ROM and LDF-ROM introduced in Section 2.1 through the accuracy of the mean field and the QOIs. The mean field is computed by averaging the POD coefficients and reconstructing with the rectangular basis matrices $\mathbf{B}$ and B. The QOIs are the mean Nu and Std(Nu), which are estimated through Eq. 6, and the mean TKE and mean temperature fluctuation, estimated through Eq. 7. The quantities are averaged over 500 CTUs.

Although of limited practical interest, the solution reproduction problem is an important step towards the development of a MOR procedure for the parametric problem. The reproduction results for spatio-temporal chaotic flow are presented and discussed in Section 5.1.

3.1 Numerical results

Results for Ra = 2×10^5 are shown in Figures 2A–D. The performances of G-ROM, C-ROM and LDF-ROM are indicated by blue, orange, green solid line. In LDF-ROM, the radius of the differential filter is $\delta = 0.015625$. Figure 2A shows the behavior of the relative $\mathcal{H}^1$ error in the mean flow prediction versus N. We observe less than 1% error in C-ROM with small N. The error in G-ROM and LDF-ROM decreases as N increases and eventually reaches 1% with $N = 100$. Similar trends for the mean Nu, Std(Nu) and mean temperature fluctuation are observed in Figures 2B–D. The FOM data is denoted as black solid line. For mean Nu prediction, we observe around 0.1% error in C-ROM for almost all N. The error in G-ROM and LDF-ROM decreases as N increases and LDF-ROM has error around 0.01% with $N = 100$. For Std(Nu) and mean temperature fluctuation prediction, we observe convergence in both QOIs with G-ROM and LDF-ROM. For C-ROM, the predictions are in good agreement with the FOM data for all values of N.Behaviors of the same quantities for Ra = 7×10^5 are shown in Figures 2E–H. In LDF-ROM, the radius of the differential filter is $\delta = 0.03125$. We still observe convergence in the mean flow and QOIs but it is much slower because the flow is more chaotic. The error in the mean flow is around 10% in G-ROM and the prediction in QOIs are over-estimated and less accurate than for the other methods. With LDF-ROM, the predictions are slightly better. C-ROM is still the most effective and has 5% error in mean flow prediction. Although the prediction in mean Nu is only as good as the LDF-ROM, the Std(Nu) and mean temperature fluctuation are bounded from above for all values of N and converge to the correct value as N increases.

Note that the differential filter radius δ selected for the two Ra yields the best accuracy in mean flow among the five differential filter radius $\delta = 0.25, 0.125, 0.0625, 0.03125, 0.015625$. Besides, we find $\delta = 0.015625$ yield the best results at smaller Ra and as Ra increases, results with $\delta = 0.03125$ becomes better and are comparable with $\delta = 0.015625$ at Ra = 7×10^5. The tendency is reasonable since the flow is more chaotic as Ra increases, therefore one should expect a larger δ to stabilize the flow.

From the results, we observe the mean flow and QOIs converge with N for all ROMs. With higher Ra, convergence in those quantities is much slower and a larger N is required to reach to the same accuracy as in the lower Ra case. Note that, because of the $O(N^3)$ online costs, requiring a large value of N for convergence might require off-line resources for timely simulation, which would greatly diminish the intrinsic advantage of the ROM/pMOR framework. This potentiality highlights the importance of stabilization methods. Indeed, we find that C-ROM is able to predict the mean flow and QOIs with a better accuracy with smaller N. On the other hand, although LDF-ROM is not as effective as C-ROM, and only slightly better

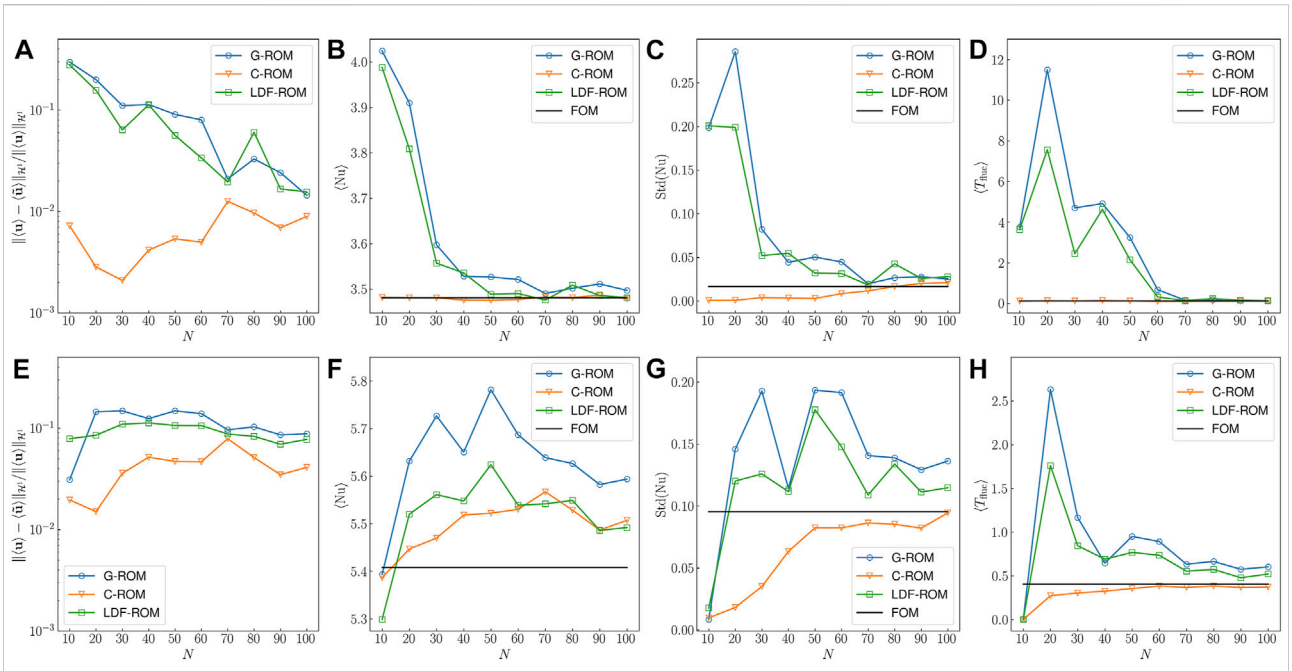


FIGURE 2
Performance comparison between G-ROM, C-ROM and LDF-ROM for the solution reproduction problem at $\theta_g = 90^\circ$ and Pr = 7.2. **(A–D)** Behavior of the relative $\mathcal{H}^1$ error in mean flow prediction, predicted mean Nu, Std(Nu), and mean temperature fluctuation as a function of number of modes, N at Ra = 2×10^5 ($\delta = 0.015625$ in LDF-ROM). **(E–H)** Performance for the same quantities at Ra = 7×10^5 ($\delta = 0.03125$ in LDF-ROM).

than G-ROM, it will play a key role in the pMOR presented in Section 4, especially in the parametric problem with θ_g variation.

4 The parametric problem

We turn now to study the performance of pMOR for the slot problem with G-ROM, C-ROM, and LDF-ROM. Two sets of parameterization are considered. With Pr = 7.2 fixed, we seek to estimate the solution and QOIs of Eqs. 13 and 15 for: 1) $\theta_g \in \mathcal{P} = [0^\circ, 180^\circ]$ at multiple Ra, and 2) Ra $\in \mathcal{P} = [2 \times 10^5, 7 \times 10^5]$ with $\theta_g = 90^\circ$. Throughout, we take the lifting functions to be the zero velocity field and the heat conduction solution.

For efficient selection of the pMOR anchor points we consider the POD-*h*Greedy algorithm proposed in [9] which combines POD in time with Greedy in parameter. The term Greedy refers to the optimization strategy of basing anchor point selection on the training point that exhibits the largest value in the error estimate. Error-indicated selection of the anchor points reduces the number of FOM solves and is thus critical for the feasibility of pMOR. Here, the error indicator corresponds to the dual norm of the residual associated with the time-averaged momentum and energy equations.

The section is organized as follows. In Section 4.1 we present the POD-*h*Greedy algorithm. In Section 4.2, we extend the time-averaged error indicator introduced in [9] to accommodate the energy equation and Leray regularization. Finally, in Sections 4.3 and 4.4, we present the numerical results.

4.1 Proper orthogonal decomposition-*h*Greedy algorithm

In this section, we present the POD-*h*Greedy algorithm for the construction of the reduced spaces $\{\mathbf{X}_{0,\ell}^N, X_{0,\ell}^N\}_{\ell=1}^L$, and the partition $\{\mathcal{I}_\ell\}_{\ell=1}^L$ of $\mathcal{P}$, based on the results of L full-order simulations associated with the parameters $p_1^\star, \ldots, p_L^\star$. The algorithm is similar to the one in [9] but with extensions for thermal fields.

To begin, we introduce the discretized parameter space $\mathcal{P}_{\text{train}} = \{p_i\}_{i=1}^{n_{\text{train}}}$, $p_1 \le \ldots \le p_{n_{\text{train}}}$, the integers L which fix the maximum number of offline solves, the integer $n_{\text{cand}} < L$, which is the number of ROM evaluations performed online for a given value of the parameters, and an error indicator Δ. The error indicator takes as input sequences $\{\bar{\mathbf{u}}_c^n\}_{n=0}^J$ and $\{\bar{T}_c^n\}_{n=0}^J$ and the value of the parameter, $p^\star$, and returns an estimate of the error in the prediction of the mean flow. We formally present the indicator in Section 4.2.

Algorithm 1 presents the computational procedure for both offline and online stage. The offline procedure starts with an anchor point that could either be selected randomly from the training space $\mathcal{P}_{\text{train}}$ or be user specified. At each iteration ℓ, a

FOM simulation at the anchor point $p_\ell^\star$ is conducted and returns a set of snapshots. The snapshot set is then processed through POD and returns the first N orthonormalized POD modes. The value of N is determined by reproduction problem at the anchor point. The ROM and the error indicator, Δ_ℓ, are then built with the reduced spaces $\mathbf{X}_{0,\ell}^N$ and $X_{0,\ell}^N$. The coefficients and the error estimates are then computed for each $p \in \mathcal{P}_{\text{train}}$ and the next anchor point is identified as the parameter that has the maximum value in the current (including previous) error estimate. The procedure starts again with the new selected anchor point. If the error indicator is sufficiently small over all points in $\mathcal{P}_{\text{train}}$ or the procedure reaches the maximum number of FOM solves L, the offline stage terminates.

Given the ROM/anchor point data $(\mathbf{X}_{0,\ell}^N, X_{0,\ell}^N, p_\ell^\star)$ for $\ell = 1, \ldots, L$, and error indicator, Δ, the hGreedy online stage starts with finding the n_{cand} candidate anchor points nearest to the test parameter p. The ROMs associated to the candidate anchor points are then used to compute the coefficients and error estimate at p. The coefficients are then returned based on the ROM that has smallest error estimate. The POD-hGreedy approach is analogous to h-refinement in the finite element method in that the POD bases are not shared between anchor points. Convergence is therefore expected to be linear in the distance from the nearest anchor point.

Offline stage: $[\{\mathbf{X}_{0,\ell}^N, X_{0,\ell}^N\}_{\ell=1}^L] = \text{Offline}(\mathcal{P}_{\text{train}}, L, \Delta)$.
Inputs: $\mathcal{P}_{\text{train}} = \{p_i\}_{i=1}^{n_{\text{train}}}$ = discretized parameter space, L = maximum number of offline solves, Δ = error indicator defined in (44).
Output: $\{(\mathbf{X}_{0,\ell}^N, X_{0,\ell}^N, p_\ell^\star)\}_{\ell=1}^L$ = reduced space/anchor point pairs.

$p_1^\star$ = smallest p in $\mathcal{P}_{\text{train}}$.
for $\ell = 1, \ldots, L$ **do**
  $[\{(\mathbf{u}^k(p_\ell^\star), T^k(p_\ell^\star))\}_{k=1}^K] = \text{DNS-solver}(p_\ell^\star)$.
  $[\{(\boldsymbol{\zeta}_{n,\ell}, \theta_{n,\ell})\}_{n=1}^N] = \text{POD}(\{(\mathbf{u}^k(p_\ell^\star), T^k(p_\ell^\star))\}_{k=1}^K, N)$.
  Define $\mathbf{X}_{0,\ell}^N = \text{span}\{\boldsymbol{\zeta}_{n,\ell}\}_{n=1}^N$, $X_{0,\ell}^N = \text{span}\{\theta_{n,\ell}\}_{n=1}^N$, build the ROM operators and error indicator Δ_ℓ.
  for $i = 1, \ldots, n_{\text{train}}$ **do**
    $[\{(\mathbf{u}_c^j(p_i), T_c^j(p_i))\}_{j=0}^J] = \text{ROM-solver}(p_i, \mathbf{X}_{0,\ell}^N, X_{0,\ell}^N)$.
    Compute the weak dual norm Δ_ℓ at p_i.
  end for
  $p_{\ell+1}^\star = \arg\max_{p \in \mathcal{P}_{\text{train}}} \min\{\Delta_{\ell'}(p)\}_{\ell'=1}^\ell$.
end for
Online stage: $[\{\mathbf{u}_c^j, T_c^j\}_j] = \text{Online}(\{(\mathbf{X}_{0,\ell}^N, X_{0,\ell}^N, p_\ell^\star)\}_{\ell=1}^L, \Delta, n_{\text{cand}}, p)$.
Inputs: $\{(\mathbf{X}_{0,\ell}^N, X_{0,\ell}^N, p_\ell^\star)\}_{\ell=1}^L$ = reduced space/anchor point pairs, n_{cand} = online ROM evaluations, p = input parameter.
Output: $\{\mathbf{u}_c^j, T_c^j\}_j$ = solution estimate.

Find the n_{cand} candidate anchors nearest to $p : p_1^\star, \ldots, p_{n_{\text{cand}}}^\star$.
for $i = 1, \ldots, n_{\text{cand}}$ **do**
  $[\{u_{c,(i)}^j, T_{c,(i)}^j\}_{j=0}^J] = \text{ROM-solver}(p_i^\star, \mathbf{X}_{0,(i)}^N, X_{0,(i)}^N)$.
  Compute the error estimate $\Delta_{(i)}(p)$.
end for
Return $\{(u_c^j, T_c^j) = (u_{c,(i^*)}^j, T_{c,(i^*)}^j)\}_j$, where i^* is the minimizer of $\{\Delta_{(i)}(p)\}_i$.

Algorithm 1. POD-hGreedy algorithm for the construction of $\{\mathbf{X}_{0,\ell}^N, X_{0,\ell}^N, \mathcal{I}_\ell\}_l$.

Another strategy is the POD-pGreedy algorithm, following the definitions of [37], as first proposed in [38] and analyzed in [39]. The algorithm combines data from different parameters to generate a single reduced basis set that covers the entire parameter space $\mathcal{P}$. The procedure is similar to Algorithm 1 but with few differences:

1) The reduced bases are shared between anchor points. POD is still used to construct the new basis but the collected snapshot set is projected onto the orthogonal complement of the existing basis.
2) In the online/training stage, only one ROM is used instead of a set of ROMs and there is no need to check for the nearest anchor points.
3) The anchor point is selected based on the single error estimate Δ in current iteration, rather than the individual estimates for each ROM.

Although it has a better convergence rate than POD-hGreedy, POD-pGreedy can easily fail for unsteady problems. Combining modes at different anchor points, especially ones whose solution exhibits different physics, can easily lead to instability and deteriorate the performance, as noted in [9]. Moreover, stabilizations that work for POD-hGreedy can fail in the POD-pGreedy approach. For example, in C-ROM, it is not clear how to construct the constraints for the combined basis. A naive approach is to apply POD to all the snapshots at anchor points. However, this approach is inefficient and can be limited by the computer storage requirements during the offline phase. Leray regularization with the projection filter (i.e., trivially truncated basis set for the advector) is also limited since the combined basis is no longer ordered in a Fourier-like, energy-decaying, sequence. To address this, one could apply POD to all the snapshots that have been collected but this approach is again limited by the storage and therefore not practical. An alternative is to consider DF filter, denoted as LDF-ROM here. Once the radius δ is specified, it will filter right amount of energy in each basis.

4.2 A time-averaged error indicator

In this section, we extend the time-averaged error indicator proposed in [9] to accommodate the energy equation and Leray regularization. The error indicator is based on the dual norm of the discrete time-averaged residual. Given the ROM solution sequence $\{\bar{\mathbf{u}}_c^n\}_{n=J_0+1}^J$ and $\{\bar{T}_c^n\}_{n=J_0+1}^J$ and the parameters of interest $\underline{p} = (\nu, \theta_g, \alpha)$, the discrete time-averaged residual for velocity and temperature are defined as: [4]

$$\langle R_u \rangle\left(\{\bar{\mathbf{u}}_c^n\}_{n=J_0}^J, \mathbf{v}; \nu, \theta_g\right) := \frac{1}{J - J_0} \sum_{n=J_0+1}^J r_u\left(\bar{\mathbf{u}}_c^n, \mathbf{v}; \nu, \theta_g\right), \quad \forall\, \mathbf{v} \in \mathbf{V}_{\text{div}}, \tag{40}$$

$$\langle R_T \rangle\left(\{\bar{T}_c^n\}_{n=J_0}^J, S; \alpha\right) := \frac{1}{J - J_0} \sum_{n=J_0+1}^J r_T\left(\bar{T}_c^n, S; \alpha\right), \quad \forall\, S \in X_0^{\mathcal{N}}, \tag{41}$$

4 In [16], the velocity and temperature residuals are coupled because the velocity-temperature solutions are obtained through a coupled Newton's method. Here, we do not couple the residuals because we solve Eqs. 21 and 22 separately.

where $r_u(\bar{\mathbf{u}}_c^n, \mathbf{v}; \nu, \theta_g) \in \mathbf{V}_{\text{div}}'$ (dual space of $\mathbf{V}_{\text{div}}$) and $r_T(\bar{T}_c^n, S; \alpha) \in X_0^{\mathcal{N}\prime}$ (dual space of $X_0^{\mathcal{N}}$) are the residual associated with Eq. 32 at time t^n and defined as

$$r_u\left(\bar{\mathbf{u}}_c^n, \mathbf{v}; \nu, \theta_g\right) = \sum_{s=1}^{3} \alpha_s \left[\left(\mathbf{v}, \mathbf{g}\left(\theta_g\right)\bar{T}_c^{n-s}\right) - c\left(\mathbf{v}, \bar{\mathbf{u}}_c^{n-s}, \bar{\mathbf{u}}_c^{n-s}\right)\right] - \sum_{s=0}^{3} \frac{\beta_s}{\Delta t}\left(\mathbf{v}, \mathbf{u}_c^{n-s}\right) - \nu a\left(\mathbf{v}, \bar{\mathbf{u}}_c^n\right), \tag{42}$$

$$r_T\left(\bar{T}_c^n, S; \alpha\right) = -\sum_{s=1}^{3} \alpha_s c\left(S, \bar{\mathbf{u}}_c^{n-s}, \bar{T}_c^{n-s}\right) - \sum_{s=0}^{3} \frac{\beta_s}{\Delta t}\left(S, T_c^{n-s}\right) - \alpha\left(S, \bar{T}_c^n\right). \tag{43}$$

Note for simplicity, we assume only BDF3/EXT3 is used for time discretization in Eqs. 42 and 43. Besides, the residual is defined over $\{\mathcal{V}\}_{\text{div}} := \{\mathbf{v} | \mathbf{v} \in \mathbf{X}_0^{\mathcal{N}}, \nabla \cdot \mathbf{v} = 0\}$ and $X_0^{\mathcal{N}}$, rather than $\mathbf{X}_0^N \subset \mathcal{H}_0^1$ and X_0^N, because we measure our reduced-basis error relative to the FOM.

We define the *time-averaged error indicator*, $\Delta: \otimes_{n=J_0}^{J} \mathbf{X}^N \times X^N \times \mathcal{P} \to \mathbb{R}_+$, as follows:

$$\Delta\left(\{\bar{\mathbf{u}}_c^n\}_{n=J_0}^J, \{\bar{T}_c^n\}_{n=J_0}^J; \underline{p}\right) := \sqrt{\|\langle R_u\rangle\left(\{\bar{\mathbf{u}}_c^n\}_{n=J_0}^J, \cdot; \nu, \theta_g\right)\|_{\mathbf{V}_{\text{div}}'}^2 + \|\langle R_T\rangle\left(\{\bar{T}_c^n\}_{n=J_0}^J, \cdot; \alpha\right)\|_{X_0^{\mathcal{N}\prime}}^2}. \tag{44}$$

The residuals Eqs. 42 and 43 can be further expressed in the matrix-vector form since the spaces $\{\mathcal{V}\}_{\text{div}}$ and $X_0^{\mathcal{N}}$ are finite dimensional,

$$r_u\left(\bar{\mathbf{u}}_c^n, \mathbf{v}; \nu, \theta_g\right) = \underline{\mathbf{v}}^T \underline{\mathbf{r}}_u^n = \underline{\mathbf{v}}^T \widehat{\mathbf{f}}\left(\bar{\mathbf{u}}_c^n, \bar{T}_c^n; \theta_g\right) - \underline{\mathbf{v}}^T \mathbf{H} \mathbf{B} \widehat{\underline{u}}_c^n n, \quad \forall \underline{\mathbf{v}} \in \mathbb{R}^{2\mathcal{N}}, \tag{45}$$

$$r_T\left(\bar{T}_c^n, S; \alpha\right) = \underline{S}^T \underline{r}_T^n = \underline{S}^T \widehat{\underline{Q}}\left(\bar{\mathbf{u}}_c^n, \bar{T}_c^n\right) - \underline{S}^T H_a B \widehat{\underline{T}}_c^n, \quad \forall \underline{S} \in \mathbb{R}^{\mathcal{N}}. \tag{46}$$

The matrix-vector version of the discrete time-averaged residual Eqs. 40 and 41 is then expressed as

$$\langle R_u\rangle\left(\{\bar{\mathbf{u}}_c^n\}_{n=J_0}^J, \mathbf{v}; \nu, \theta_g\right) = \underline{\mathbf{v}}^T \underline{\mathbf{R}}_u = \underline{\mathbf{v}}^T \left(\frac{1}{J - J_0} \sum_{n=J_0+1}^{J} \widehat{\mathbf{f}}\left(\bar{\mathbf{u}}_c^n, \bar{T}_c^n; \theta_g\right) - \mathbf{H} \mathbf{B} \widehat{\underline{u}}_c^n\right), \tag{47}$$

$$\langle R_T\rangle\left(\{\bar{T}_c^n\}_{n=J_0}^J, S; \alpha\right) = \underline{S}^T \underline{R}_T = \underline{S}^T \left(\frac{1}{J - J_0} \sum_{n=J_0+1}^{J} \widehat{\underline{Q}}\left(\bar{\mathbf{u}}_c^n, \bar{T}_c^n\right) - H_\alpha B \widehat{\underline{T}}_c^n\right), \tag{48}$$

$\forall \underline{\mathbf{v}} \in \mathbb{R}^{2\mathcal{N}}$ and $\forall \underline{S} \in \mathbb{R}^{\mathcal{N}}$. The norm of the residual is closely related to the error and it is tempting to use $\|\underline{R}_u\|_2$ and $\|\underline{R}_T\|_2$ to estimate the error. However, this is not correct since $\langle R_u\rangle(\{\bar{\mathbf{u}}_c^n\}_{n=J_0}^J, \cdot; \nu, \theta_g): \mathbf{V}_{\text{div}} \to \mathbb{R}$ and $\langle R_T\rangle(\{\bar{T}_c^n\}_{n=J_0}^J, \cdot; \alpha): X_0^{\mathcal{N}} \to \mathbb{R}$ are bounded *linear* functionals whose size is appropriately measured through the dual norm:

$$\|\langle R_u\rangle\left(\{\bar{\mathbf{u}}_c^n\}_{n=J_0}^J, \cdot; \nu, \theta_g\right)\|_{\{\mathcal{V}\}_{\text{div}}'} = \sup_{\mathbf{v} \in \mathbf{V}_{\text{div}}} \frac{\langle R_u\rangle\left(\{\bar{\mathbf{u}}_c^n\}_{n=J_0}^J, \mathbf{v}; \nu, \theta_g\right)}{\|\mathbf{v}\|_{\mathbf{V}_{\text{div}}}}, \tag{49}$$

$$\|\langle R_T\rangle\left(\{\bar{T}_c^n\}_{n=J_0}^J, \cdot; \alpha\right)\|_{X_0^{\mathcal{N}}} = \sup_{S \in X_0^{\mathcal{N}}} \frac{\langle R_T\rangle\left(\{\bar{T}_c^n\}_{n=J_0}^J, S; \alpha\right)}{\|S\|_{X_0^{\mathcal{N}}}}. \tag{50}$$

Thanks to the Riesz representation theorem, there exist a unique $\langle \hat{\mathbf{R}}_u \rangle \in \mathbf{V}_{\text{div}}$ and $\langle \hat{R}_T \rangle \in X_0^{\mathcal{N}}$ such that

$$\left(\langle \hat{\mathbf{R}}_u \rangle, \mathbf{v}\right)_{\mathbf{V}_{\text{div}}} = \langle R_u\rangle\left(\{\bar{\mathbf{u}}_c^n\}_{n=J_0}^J, \mathbf{v}; \nu, \theta_g\right), \quad \forall\, \mathbf{v} \in \mathbf{V}_{\text{div}}, \tag{51}$$

$$\left(\langle \hat{R}_T \rangle, S\right)_{X_0^{\mathcal{N}}} = \langle R_T\rangle\left(\{\bar{T}_c^n\}_{n=J_0}^J, S; \alpha\right), \quad \forall\, S \in X_0^{\mathcal{N}}. \tag{52}$$

It thus follows that

$$\|\langle R_u\rangle\left(\{\bar{\mathbf{u}}_c^n\}_{n=J_0}^J, \cdot; \nu, \theta_g\right)\|_{\mathbf{V}_{\text{div}}'} = \|\langle \hat{\mathbf{R}}_u \rangle\|_{\mathbf{V}_{\text{div}}}, \tag{53}$$

$$\|\langle R_T\rangle\left(\{\bar{T}_c^n\}_{n=J_0}^J, \cdot; \alpha\right)\|_{X_0^{\mathcal{N}\prime}} = \|\langle \hat{R}_T \rangle\|_{X_0^{\mathcal{N}}}. \tag{54}$$

Equations 51 and 52 allows one to compute the Riesz representers $\langle \hat{\mathbf{R}}_u \rangle$ and $\langle \hat{R}_T \rangle$ and Eqs. 53 and 54 allows one to evaluate the dual norm of the residual through Riesz representation without computing the supremum.

In practice, determination of the Riesz representers, $\langle \hat{\mathbf{R}}_u \rangle$ and $\langle \hat{R}_T \rangle$, is relatively straightforward because the coarse (i.e., ROM) and truth (FOM) representations live in finite-dimensional spaces, meaning that there is a direct linear-algebra problem to be solved for the Riesz representers. Expanding Eqs. 51 and 52, we have the corresponding linear algebra statement,

$$\begin{bmatrix} \mathbf{A} & -\mathbf{D}^T \\ -\mathbf{D} & 0 \end{bmatrix} \begin{pmatrix} \langle \underline{\hat{\mathbf{R}}}_u \rangle \\ \underline{p} \end{pmatrix} = \begin{pmatrix} \underline{\mathbf{R}}_u \\ 0 \end{pmatrix}, \tag{55}$$

$$A \langle \underline{\hat{R}}_T \rangle = \underline{R}_T. \tag{56}$$

Here, $\mathbf{A}$ corresponds to $\mathbf{H}$ introduced in Eq. 23 with $\beta_0 = 0$ and $\nu = 1$. We remark that the essential difference between the velocity and temperature representers is that the former satisfies the divergence-free constraint by virtue of the 2×2 block system in Eq. 55. Evaluation of the error indicator Δ entails solving Eqs. 55 and 56, computing the corresponding $\mathcal{H}^1$ norms of the outputs, and ultimately using these results in Eq. 44.

While use of the direct approach requires access to the FOM machinery in order to generate an error indicator, we note that such access is readily available during the pMOR training/construction phase. The advantage of this approach is that the number of Stokes/Poisson solves scales as the number of points in the training space, which is typically less than N^2. The other is through the offline/online computational decomposition which takes the advantage of the affine decomposition and expands the residual. By expanding the residuals $\langle R_u \rangle$ and $\langle R_T \rangle$, $2(N+1)^2 + 6(N+1)$ linear functionals are derived, where $2(N+1)^2$ is due to the convection term in the Navier-Stokes and energy equations. Applying the Riesz representation theorem to each linear functional, we end up solving $2(N+1)^2 + 6(N+1)$ Riesz representers, where $(N+1)^2 + 4(N+1)$ of them are solved through Stokes problems and $(N+1)^2 + 2(N+1)$ of them are solved through Poisson problems. Note that

the Riesz representers must be stored in order to accomplish the decomposition and each is a vector of size $\mathcal{N}$ since it lives in the FOM space. For example, if $N = 60$, one would have to store at least 7,200 vectors of size $\mathcal{N}$ which can be prohibitive, even for large multicore workstations. Even if there is no storage limitation, the offline cost is quite high when N is large as it scales quadratically. Once it is done, the online cost is $\mathcal{O}(N^2J + N^4)$, where $\mathcal{O}(N^2J)$ is to solve Eq. 32 and $\mathcal{O}(N^4)$ is required to compute the error estimate. Further details of the decomposition are provided in [9].

4.2.1 Time-averaged error indicator with Leray regularization

The integration of the time-averaged error indicator Δ with Leray regularization is rather straightforward. Recall the difference between G-ROM Eq. 32 and LDF-ROM Eq. 35 is simply the advecting field being filtered. Hence, the residuals $r_u(\bar{\mathbf{u}}_c^n, \mathbf{v}; \nu, \theta_g)$ and $r_T(\bar{T}_c^n, S; \alpha)$ for all $n = J_0 + 1, \ldots, J$ are simply modified with the filtered advecting field,

$$r_u\left(\bar{\mathbf{u}}_c^n, \mathbf{v}; \nu, \theta_g\right) = \sum_{s=1}^{3} \alpha_s \left[\left(\mathbf{v}, \mathbf{g}\left(\theta_g\right)\bar{T}_c^{n-s}\right) - c\left(\mathbf{v}, \bar{\mathbf{u}}_{c,\text{filtered}}^{n-s}, \bar{\mathbf{u}}_c^{n-s}\right)\right] - \sum_{s=0}^{3} \frac{\beta_s}{\Delta t}\left(\mathbf{v}, \mathbf{u}_c^{n-s}\right) - \nu a\left(\mathbf{v}, \bar{\mathbf{u}}_c^n\right), \tag{57}$$

$$r_T\left(\bar{T}_c^n, S; \alpha\right) = -\sum_{s=1}^{3} \alpha_s c\left(S, \bar{\mathbf{u}}_{c,\text{filtered}}^{n-s}, \bar{T}_c^{n-s}\right) - \sum_{s=0}^{3} \frac{\beta_s}{\Delta t}\left(S, T_c^{n-s}\right) - \alpha a\left(S, \bar{T}_c^n\right), \tag{58}$$

for all $\mathbf{v} \in \mathbf{V}_{\text{div}}$ and $S \in X_0^{\mathcal{N}}$. The corresponding time-averaged error indicator Δ is then defined based on the modified residuals Eqs. 57 and 58.

4.3 Parametric model order reduction results: θ_g variation

In this section, we consider the parametric problem parameterized with θ_g at Pr = 7.2. The problem has three characteristics: bifurcation, spatio-temporal chaos over a certain range of θ_g, and a solution manifold that is a blend of steady and unsteady solutions. To identify the major pMOR challenges for this case, three values of Ra are considered:

1) Ra = 1×10^4 where the FOM is steady except at $\theta_g = 30^\circ$.
2) Ra = 8×10^4 where the FOM is unsteady for $\theta_g \in [0^\circ, 70^\circ]$ and steady for $\theta_g \in [80^\circ, 180^\circ]$.
3) Ra = 3×10^5 where the FOM is unsteady for $\theta_g \in [0^\circ, 120^\circ]$ and steady for $\theta_g \in [130^\circ, 180^\circ]$.

In each case, the ROM is constructed through Algorithm 1. In order to assess performance, we generate FOM data for $\theta_g = 0^\circ$, 10°, …, 180° ($n_{\text{train}} = 19$ datapoints). The FOM solution is obtained by solving Eqs. 2 and 3. For parameters where the problem is steady, the solution and the Nu are collected after the solution difference between ten time steps is less than 10^{-6}. For unsteady problems, the mean flow, mean Nu, Std(Nu), mean TKE and mean temperature fluctuation are averaged over 500 CTUs after the solution has reached a statistically steady state.

Although not shown here, we remark that at Ra = 1×10^3, the problem is steady with a bifurcation at $\theta_g = 20^\circ$. In this case, either h- or p-Greedy with residual dual-norm base error indicator accurately estimates the solution and QOIs over the parameter space $\theta_g \in [0^\circ, 180^\circ]$.

4.3.1 Ra = 1×10^4

To examine the feasibility of the pMOR (Algorithm 1) in the unsteady case, we begin with Ra = 1×10^4, in which only one unsteady solution is introduced at $\theta_g = 30^\circ$.

Figure 3E shows the steady (or mean) velocity magnitude for 19 uniformly-space training points $\theta_g = 0^\circ$, 10°, …, 180°. The corresponding temperature distributions are in Figure 1B. At $\theta_g = 180^\circ$, we observe no flow and the temperature is simply the conduction solution. As θ_g decreases, we observe slot convection and then about $\theta_g = 40^\circ$ there is a bifurcation to the wavy flow and rolls in the velocity. Moreover, we observe spatial-temporal chaos at $\theta_g = 30^\circ$. Figures 3A–D show the results of the application of Algorithm 1 for the construction of the G-ROM, C-ROM and LDF-ROM for the pMOR. The algorithm starts with $\theta_{g,1}^\star = 0^\circ$ and is performed with $L = 8$ iterations.

Figure 3A demonstrates the selection process of anchor points (denoted by red circles) for the G-ROM case. We briefly walk through the process: At the first iteration, the error estimate $\Delta_1(\theta_g)$ for $\theta_g \in \mathcal{P}_{\text{train}}$ is computed (blue dashed line). With the largest error estimate, $\theta_g = 90^\circ$ is then identified as the second anchor point. The third anchor point is then selected from $\theta_g \in \mathcal{P}_{\text{train}}$ which maximizes the error estimate $\Delta_{1,2}(\theta_g)$: = $\min\{\Delta_1(\theta_g), \Delta_2(\theta_g)\}$ over $\mathcal{P}_{\text{train}}$. (We reiterate that minimizing over the individual error estimates is a property of the h-Greedy process—there is not a single unifying error estimate as is the case for p-Greedy.) The process continues until the error estimate reaches the desired tolerance or the number of offline solves reaches its maximum. The black solid line denotes the minimum of all error estimates computed up to current iteration. In this case, it represents $\min\{\Delta_1(\theta_g), \ldots, \Delta_8(\theta_g)\}$. Note that the error estimate at $\theta_g = 180^\circ$ in each model $\Delta_\ell(\theta_g = 180^\circ)$ is small due to the choice of lift function.

From Figure 3A, we observe the error estimate is small at anchor points where the problem is steady. On the other hand, although the error estimate $\Delta_3(\theta_g)$ (greed line) is small at $\theta_{g,3}^* = 30^\circ$ compared to other points in $\mathcal{P}_{\text{train}}$, it is not as small as the estimate at other anchor points. Because of the unsteadiness, it can't reach the same magnitude as in the steady cases.

Following this procedure for the other cases, we present models results for the G-ROM, C-ROM and LDF-ROM cases, denoted respectively blue, orange, green solid lines in Figures 3B–D. The behavior of the relative $\mathcal{H}^1$ error in the predicted

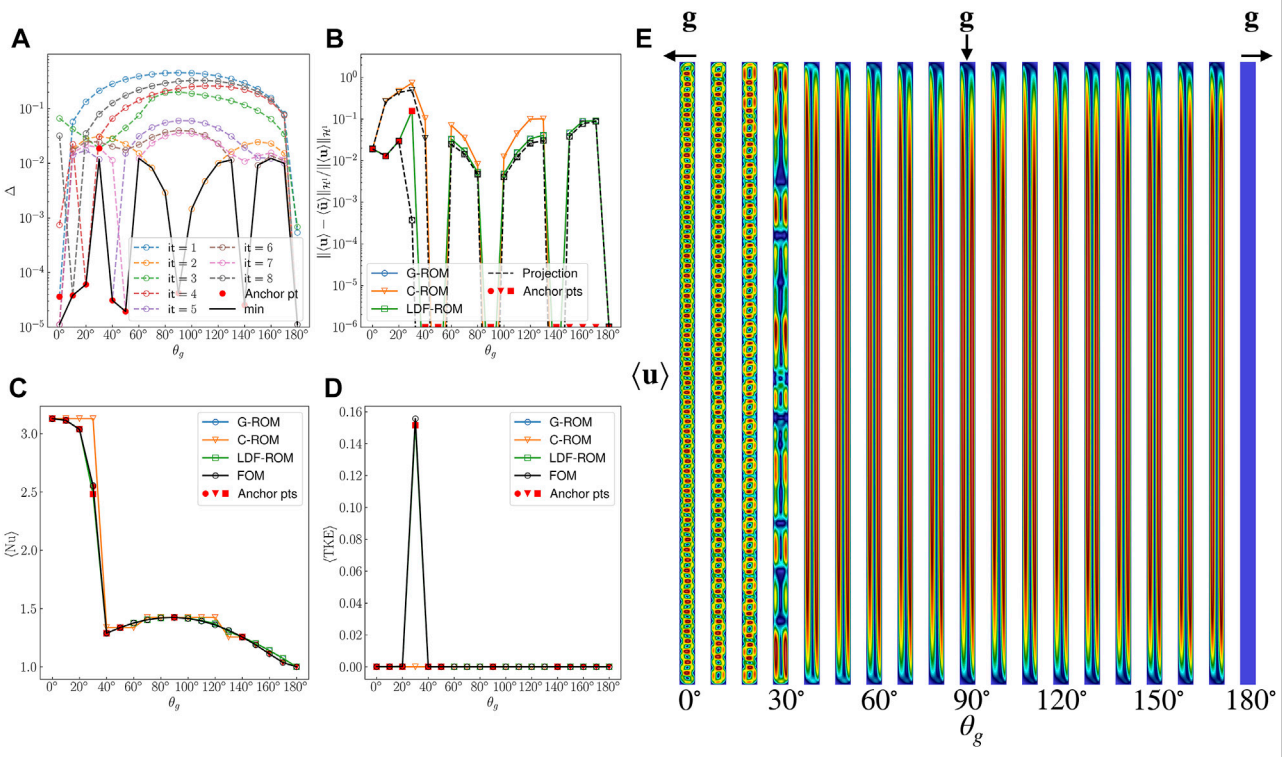


FIGURE 3
POD-*h*Greedy performance comparison between G-ROM, C-ROM and LDF ROM for the parametric problem parameterized with θ_g at Ra = 1×10^4 and Pr = 7.2. **(A)** Behavior of the error indicator Δ defined in Eq. 44 for eight iterations with G-ROM. **(B–D)** Behavior of the relative $\mathcal{H}^1$ error in predicted solution, mean Nu and mean TKE with θ_g based on eight anchor points. ($N = 1$ for all $\theta_g \in \mathcal{P}_{\text{anchor}}$ except $N = 70$ at $\theta^*_{g,3} = 30^\circ$.). **(E)** Mean (or steady) velocity magnitude for 19 uniformly-spaced $\theta_g \in [0^\circ, 180^\circ]$. (The corresponding temperature solutions are in Figure 1B).

solution is shown in Figure 3B. The corresponding Galerkin projection is denoted as the black dashed line. We found that G-ROM and LDF-ROM have a similar performance: the error in the solution is nearly identical to the Galerkin projection for cases where the problem is steady, including those that are not in the $\mathcal{P}_{\text{anchor}}$. The maximum error is at $\theta_g = 30^\circ$ where the problem is unsteady. Both methods have around 15% error in the mean flow. Note that at $\theta_g = 30^\circ$, the number of modes N is carefully selected since the ROM diverges after certain N due to the spatio-temporal chaos (15% error with $N = 70$, 17% error with $N = 80$ and 23% error with $N = 90$). On the other hand, the mean solution prediction made by C-ROM has 73% in maximum error and in order for C-ROM to reach same accuracy as in G-ROM and LDF-ROM, two more iterations are required. Already, with this modest Ra = 1×10^4, we find C-ROM is less efficient than G-ROM and LDF-ROM. The pMOR behavior for mean Nu and mean TKE are shown along with the FOM results in Figures 3C,D. Again, G-ROM and LDF-ROM are able to make accurate predictions while C-ROM has maximum 22% error in mean Nu and in particular is unable to capture the peak in mean TKE at $\theta_g = 30^\circ$.

Before closing this section, we highlight some observations with respect to the solution manifolds.

1) The solutions at $\theta_g \in [0^\circ, 30^\circ]$ are Rayleigh-Bénard with differing numbers of rolls, analogous to orthogonal sine and cosine functions at different wave numbers. Therefore, there is little hope in reproducing the solution except at selected anchor points. QOI's such as mean Nu, however, are less sensitive to precise mean flow fields and are therefore more tractable.
2) At $\theta_g = 170^\circ$ and $\theta_g = 180^\circ$ the thermal metrics are not too different despite the $\mathcal{O}(1)$ difference in velocity solutions.

The first issue is resolved by the error indicator picking $\theta_g \in [0^\circ, 30^\circ]$ as anchor points. The second issue can be a source of error as Ra increases. With $\theta_g = 160^\circ$ as an anchor point and the solution at $\theta_g = 180^\circ$ as the lift function, the error at $\theta_g = 170^\circ$ is 9% for Ra = 1×10^4, 16% for Ra = 8×10^4, and 19% for Ra = 3×10^5.

4.3.2 Ra = 8×10^4

Figure 4 shows pMOR results analogous to Figure 3 for the case Ra = 8×10^4. Here, we consider only the G-ROM and

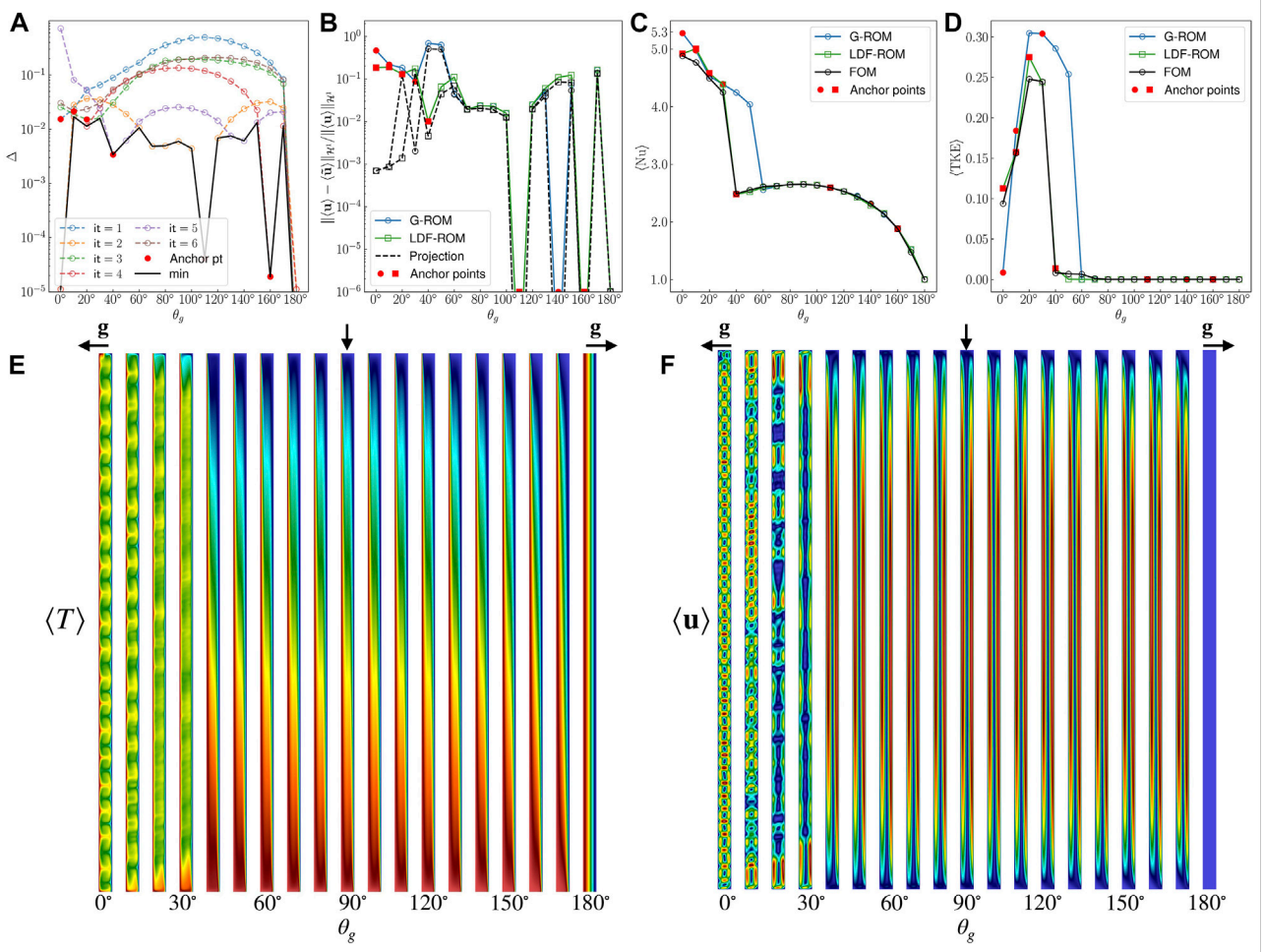


FIGURE 4
POD-*h*Greedy performance comparison between G-ROM and LDF ROM for the parametric problem parameterized with θ_g at Ra = 8×10^4 and Pr = 7.2. **(A)** Behavior of the error indicator Δ defined in Eq. 44 for six iterations with LDF-ROM. **(B–D)** Behavior of the relative $\mathcal{H}^1$ error in predicted solution, mean Nu, and mean TKE with θ_g based on six anchor points. (N = 100 at θ_g = 0°, 20°, N = 80 at θ_g = 10°, 30°, 40°, N = 1 at θ_g = 110°, 140°, 160°). **(E,F)** Mean (or steady) temperature and velocity solution for 19 uniformly spaced $\theta_g \in [0°, 180°]$.

LDF-ROM. The algorithm starts with $\theta^*_{g,1} = 0°$ and terminates at $L = 6$ iterations. Figures 4E,F show the FOM mean (or steady) temperature and velocity solution at the training points. (In an actual pMOR, these FOMs would of course not be computed *a priori*.) With this Ra, the bifurcation occurs at $\theta_g = 40°$. Moreover, we observe spatio-temporal chaos for $\theta_g \in [0°, 30°]$ with the lower values being more chaotic.

The anchor-point selection process with LDF-ROM is demonstrated in Figure 4A. Starting with $\theta^*_{g,1} = 0°$ the peak error in first iteration is at 110°, which is chosen to be $\theta^*_{g,2}$, and so on. Again, we find the error indicator is small at anchor points where the problem is steady and that it is larger where it is unsteady ($\theta_g \in [0°, 60°]$). Nonetheless, the error indicator is still able to identify where solution changes rapidly and select most of the anchor points in the region [0°, 40°].

The behaviors of the relative $\mathcal{H}^1$ error in the predicted solution with θ_g using G-ROM and LDF-ROM are shown in Figure 4B. For $\theta_g \in [80°, 180°]$, where the solution is steady, we find the estimation is almost identical to the Galerkin projection in both models. On the other hand, for $\theta_g \in [0°, 70°]$, where the solution is unsteady, we find the error at anchor points $\theta_g = 0°$, $\theta_g = 10°$ is large due to the spatio-temporal chaos. The maximum error is around 19% at $\theta_g = 10°$ with LDF-ROM while 69% at $\theta_g = 40°$ with G-ROM. Although the maximum error in G-ROM can be reduced by further iterations of the algorithm, the error will eventually be dominated by the high reproduction error arising from spatio-temporal chaos at $\theta_g = 0°$ and 10°.

The behavior for mean Nu and mean TKE are shown in Figures 4C,D for G-ROM and LDF-ROM. Despite large errors in the mean flow prediction at $\theta_g = 0°$, 10°, the LDF-ROM is able to predict mean Nu with a maximum error around 5% whereas

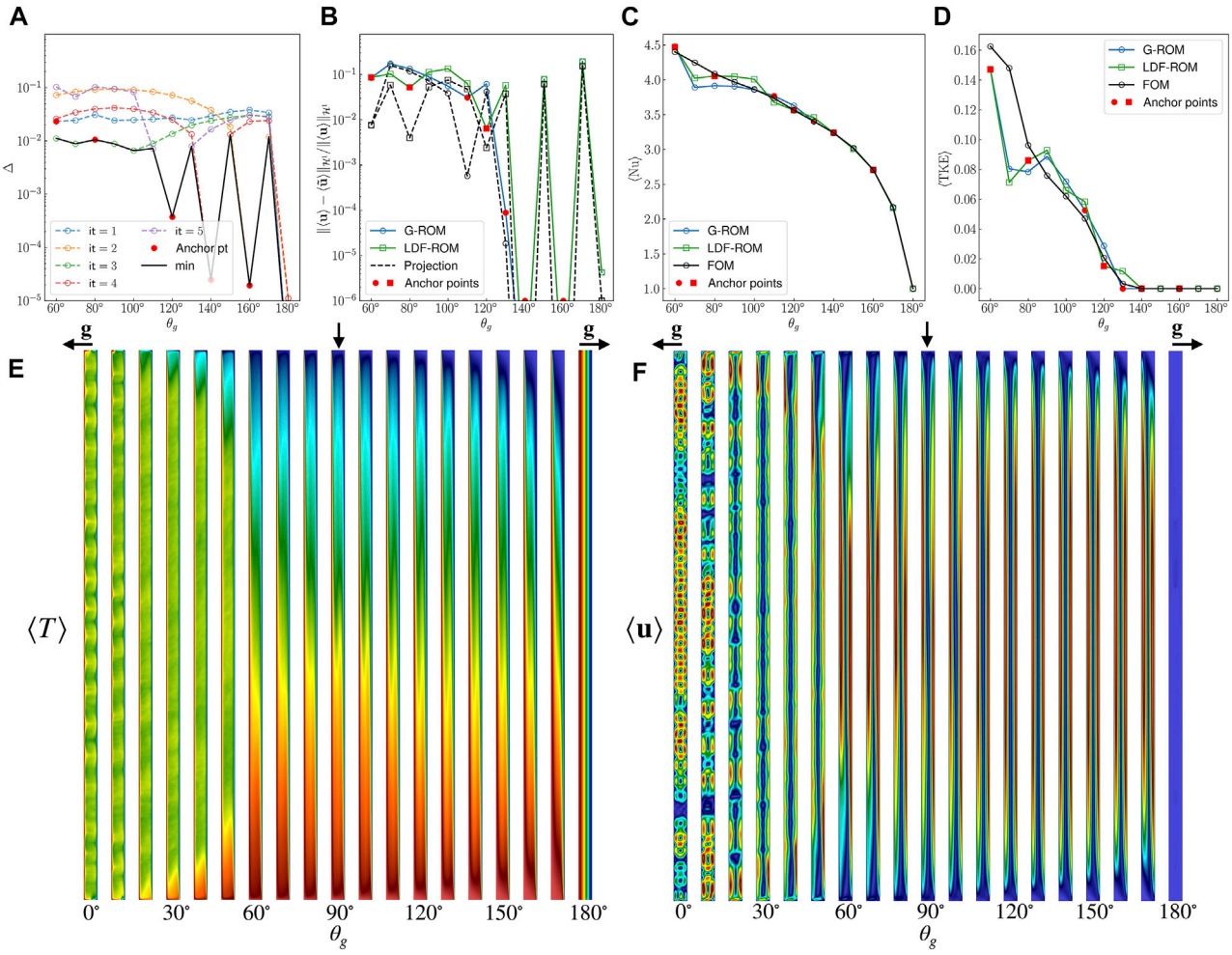


FIGURE 5
POD-*h*Greedy performance comparison between G-ROM and LDF ROM for the parametric problem parameterized with $\theta_g \in [60^\circ, 180^\circ]$ at Ra = 3×10^5 and Pr = 7.2. **(A)** Behavior of the error indicator Δ defined in Eq. 44 for five iterations with LDF-ROM. **(B–D)** Behavior of the relative $\mathcal{H}^1$ error in predicted solution, mean Nu and mean TKE with θ_g based on five anchor points. ($N = 80$ at $\theta_g = 60^\circ, 80^\circ, 110^\circ$, $N = 50$ at $\theta_g = 120^\circ$ and $N = 1$ at $\theta_g = 130^\circ, 140^\circ, 160^\circ$). **(E,F)** Mean (or steady) temperature and velocity solution for 19 uniformly spaced $\theta_g \in [0^\circ, 180^\circ]$.

G-ROM has maximum error around 28%. In addition, LDF-ROM is able to more accurately predict mean TKE than G-ROM.

4.3.3 Ra = 3×10^5

For (Pr,Ra) = ($7.2, 3 \times 10^5$) the flow is quite chaotic (similar to what is found for Pr = 0.71 at lower Ra). Figures 5E,F show the mean (or steady) temperature and velocity solution at $\theta_g = 0^\circ, 10^\circ, \ldots, 180^\circ$ (19 datapoints). This time, the bifurcation occurs at $\theta_g = 60^\circ$. We use this elevated Rayleigh-number case to explore the behavior of the *h*-Greedy pMOR convergence by considering application of the algorithm to two different training sets, $\mathcal{P}_1 = [60^\circ, 70^\circ, \ldots, 180^\circ]$ and $\mathcal{P}_2 = [0^\circ, 10^\circ, \ldots, 180^\circ]$. The set $\mathcal{P}_1$ excludes the spatio-temporal chaotic regime while $\mathcal{P}_2$ spans the full range of flow phenomena.

The anchor point selection process for $\mathcal{P}_1$ with LDF-ROM is demonstrated in Figure 5A, starting with $\theta^*_{g,1} = 60^\circ$ and proceeding for $L = 5$ iterations. Again, we observe that the error estimate at the anchor points, $\theta^*_{g,1} = 60^\circ$, $\theta^*_{g,3} = 80^\circ$ and $\theta^*_{g,5} = 120^\circ$ are larger than other anchor points because of unsteadiness. The behavior of the relative $\mathcal{H}^1$ error in predicted solution is shown in Figure 5B. For $\theta_g \in [130^\circ, 180^\circ]$, where the solution is steady, the ROM estimates at the anchor points are almost identical to the Galerkin projection. For $\theta_g \in [60^\circ, 120^\circ]$, where the solution is unsteady, the errors at the anchor points ($\theta_g = 60^\circ, 80^\circ, 110^\circ, 120^\circ$) are less than 10%. However, because of the irregular solution manifold, there is a 20% maximum error at $\theta_g = 170^\circ$, despite the ROM being based on the nearby $\theta_g = 160^\circ$ anchor point.

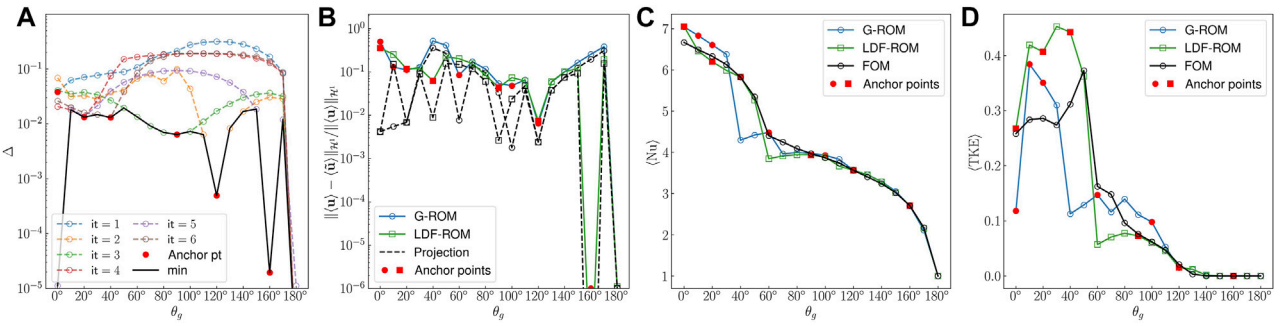


FIGURE 6
POD-*h*Greedy performance comparison between G-ROM and LDF ROM for the parametric problem parameterized with $\theta_g \in [60^\circ, 180^\circ]$ at Ra = 3×10^5 and Pr = 7.2. **(A)** Behavior of the error indicator Δ defined in Eq. 44 for five iterations with LDF-ROM. **(B–D)** Behavior of the relative $\mathcal{H}^1$ error in predicted solution, mean Nu and mean TKE with θ_g based on five anchor points. ($N = 80$ at $\theta_g = 10^\circ, 20^\circ, 40^\circ, 60^\circ, 90^\circ, 100^\circ$, $N = 70$ at $\theta_g = 0^\circ$, $N = 50$ at $\theta_g = 120^\circ$ and $N = 1$ at $\theta_g = 160^\circ$.)

The behavior of the mean Nu and mean TKE are shown in Figures 5C,D. The maximum error in the predicted mean Nu is around 5% with LDF-ROM and 8% with G-ROM. The mean TKE estimation is also reasonable but is underestimated at $\theta_g = 70^\circ$.

Next we examine the same problem configuration but with the full parameter space $\mathcal{P}_2$. The problem now includes spatio-temporal chaos for $\theta_g \in [0^\circ, 40^\circ]$ with the lower values being more chaotic. Figure 6 show the results of the application of Algorithm 1 for the construction of the LDF-ROM for the parametric problem with $\mathcal{P}_2$. The algorithm starts with $\theta^\star_{g,1} = 0°$ and is performed with $L = 6$ iterations. The anchor point selection process is demonstrated in Figure 6A. We observe the same issue as in the previous cases, where unsteadiness leads to larger error estimates than with the steady regimes.

The relative $\mathcal{H}^1$ error in predicted solution is shown in Figure 6B. Again we find the estimation is almost identical to the Galerkin projection for $\theta_g \in [130^\circ, 180^\circ]$ where the solution is steady. For $\theta_g \in [0^\circ, 120^\circ]$ where the solution is unsteady, the errors at anchor points $\theta_g = 20^\circ, 40^\circ, 90^\circ, 120^\circ$ are less than 10% but 35% at $\theta_g = 0^\circ$, which corresponds to "simple" Rayleigh-Bénard convection. Note that 35% is the error after carefully chosen N and spatial radius δ in Leray filtering. The corresponding mean Nu and mean TKE behavior are shown in Figures 6C,D. The maximum error in the predicted mean Nu is around 12%. For mean TKE, the estimation for $\theta_g \in [60^\circ, 180^\circ]$ is acceptable, while it is overestimated for $\theta_g \in [0^\circ, 50^\circ]$.

We are also aware that in some applications, the Std(Nu) could be considered as QOI. However, comparing to the mean Nu and mean TKE, we find Std(Nu) is in general a more challenging QOI. Figure 7 shows the predicted Std(Nu) in the three Ra cases. We observe accurate prediction in Ra = 1×10^4 case. However, unlike the mean TKE, the Std(Nu) soon becomes intractable with Ra = 8×10^4 even with Leray regularization and is even worse in Ra = 3×10^5.

4.4 Parametric model order reduction results: Ra variation

In this section, we consider the slot problem at $\theta_g = 90^\circ$ and Pr = 7.2 with the parametric space defined by $\mathrm{Ra} \in \mathcal{P} = [2 \times 10^5, 7 \times 10^5]$. Unlike the problem with θ_g variation, all solutions are unsteady and there is no parametric bifurcation. In order to assess performance, we generate FOM data, including mean flow, mean Nu, Std(Nu), mean TKE and mean temperature fluctuation, for Ra = 2×10^5, $2.5 \times 10^5, \ldots, 7 \times 10^5$ ($n_{\text{train}} = 11$ datapoints). The quantities are averaged over 500 CTUs once the solution reaches the statistically steady state.

Figure 8 shows the results of the application of Algorithm 1 for the pMOR using G-ROM, C-ROM and LDF-ROM. The solid line denotes the performance of the reduced model which minimizes the error indicator, and thus is selected by the Greedy procedure (cf. Algorithm 1, $n_{\text{cand}} = 2$). Anchor points are denoted as red circle while FOM data is denoted as black solid line. The algorithm starts with $\mathrm{Ra}^*_1 = 2 \times 10^5$ and is performed with $L = 5$ iterations. The number of POD basis N with anchor points are listed in the figure caption.

Figure 8A shows the behavior of the relative $\mathcal{H}^1$ error in mean flow prediction with Ra. First, we observe the errors at the anchor points are less than 10% with C-ROM and LDF-ROM while G-ROM has 10% error at $\mathrm{Ra}^*_2 = 7 \times 10^5$. The maximum error is roughly 11% in G-ROM, 10% in LDF-ROM and 8% in C-ROM. Comparing with the Galerkin projection error (denoted by the black dashed line), the pMOR accuracy is seen to be quite satisfactory throughout $\mathcal{P}$.

Figure 8B shows the behavior of the predicted mean Nu with Ra. At the anchor points, we observe good agreement between ROMs and FOM and that stabilization does improve its accuracy. The maximum relative error is roughly 8% in G-ROM, 6.5% in C-ROM and 5% in LDF-ROM. Figures 8C,D show the behavior

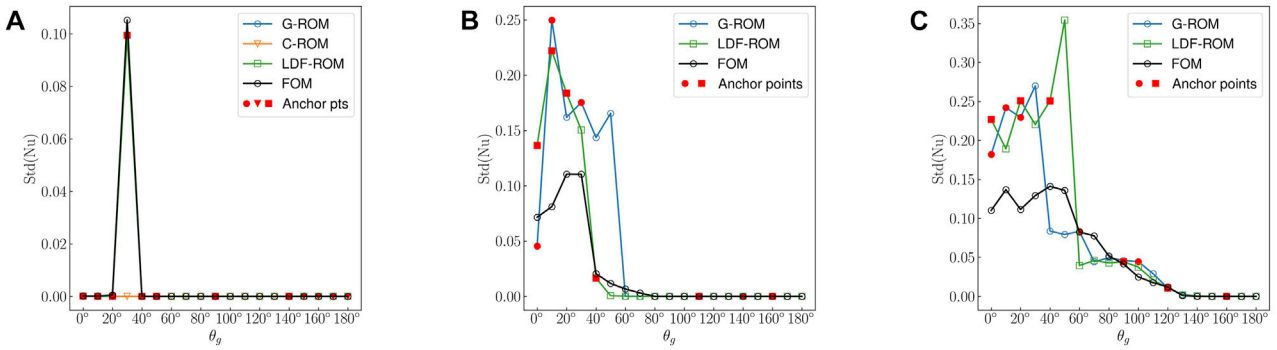


FIGURE 7
Behavior of the predicted Std(Nu) with θ_g. **(A)**: G-ROM, C-ROM and LDF-ROM estimation at Ra = 1×10^4 with eight anchor points. **(B)**: G-ROM and LDF-ROM estimation at Ra = 8×10^4 with six anchor points. **(C)**: G-ROM and LDF-ROM estimation at Ra = 3×10^5 with six anchor points.

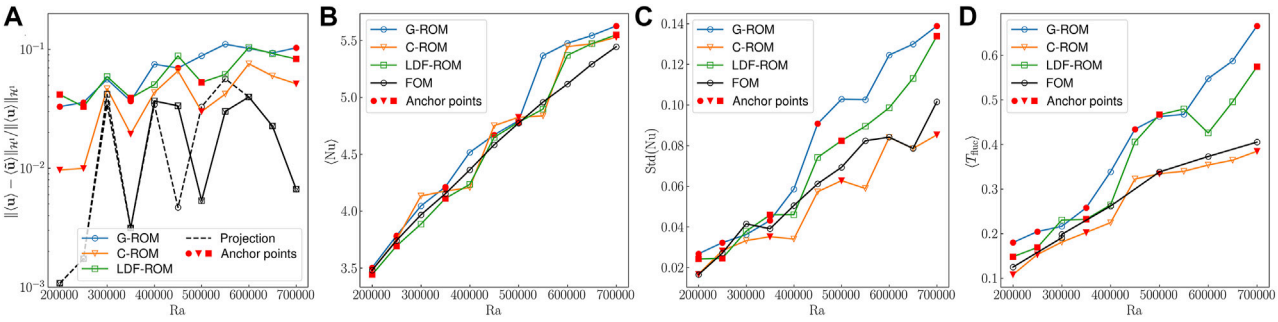


FIGURE 8
POD-*h*Greedy performance comparison between G-ROM, C-ROM and LDF ROM for the parametric problem parameterized with Ra at $\theta_g = 90^\circ$ and Pr = 7.2. **(A–D)** Behavior of the relative $\mathcal{H}^1$ error in predicted mean flow, predicted mean Nu, Std(Nu) and mean temperature fluctuation based on five anchor points. ($N = 80$, $\mathrm{Ra}_1^* = 2 \times 10^5$, $\mathrm{Ra}_2^* = 7 \times 10^5$, $\mathrm{Ra}_3^* = 2.5 \times 10^5$, $\mathrm{Ra}_4^* = 3.5 \times 10^5$, $\mathrm{Ra}_5^* = 5 \times 10^5$ for C-ROM and LDF-ROM and $\mathrm{Ra}_5^* = 4.5 \times 10^5$ for G-ROM).

of the predicted Std(Nu) and mean temperature fluctuation. In both QOIs, we find C-ROM outperforms the other two models. At Ra = 7×10^5, LDF-ROM is only slightly better than G-ROM.

This parametric space is in general more tractable than those involving variation in θ_g. This outcome might be anticipated by observing the mean temperature fields shown in Figure 1C, which suggests that the solution manifold with respect to Ra is quite smooth. This is also reflected in the QOIs, for example, the mean Nu, Std(Nu) and mean temperature fluctuation behave almost linearly as Ra increases.

5 Discussion

In this section, we investigate some of the flow behaviors exhibited by the FOM to better understand how they influence the relative performance of the ROMs. We note that we cannot, in general, expect a ROM to be able to predict FOM behavior if the flow itself is not predictable. Thus, variability in the FOM provides an anticipated lower bound on ROM performance for the reproduction problem.

5.1 Spatio-temporal chaos

As pointed out in the introduction, we classify a flow to be spatio-temporal chaotic by examining its consistency in mean flow over various time windows. (We use this simple metric here for convenience—we have also examined the flow fields and the time traces of multiple QOIs.) Here, we explore how lack of consistency influences four QOIs, mean Nu, Std(Nu), mean temperature fluctuation, and mean TKE, at three successive time windows, W1, W2 and W3. These quantities are used to indicate the variability in the FOM. As with the

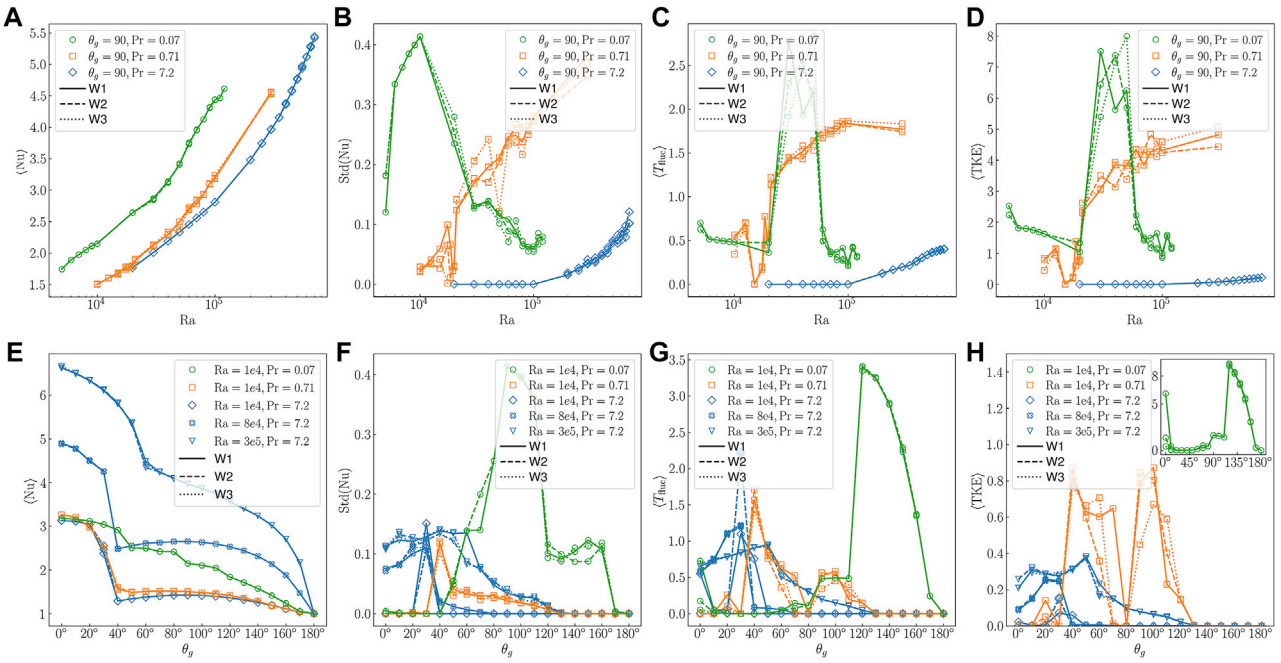


FIGURE 9
Parametric variability in the FOM: green–Pr = 0.07, orange–Pr = 0.71, and blue–Pr = 7.2. Each plot reveals absence/presence of chaotic effects by presenting statistics taken over three time windows, W1, W2, and W3. **(A–D)**: *Ra*-dependence of mean Nu, Std(Nu), mean temperature fluctuation, and mean TKE computed over time windows W1, W2 and W3. **(E–H)**: θ_g-dependence at fixed Ra.

preceding cases, we consider averaging times of 500 CTUs for each of the three windows. The starting time for W1 differs with given parameters as some cases take a longer time to reach a statistically steady state. For example, for Pr = 0.71 at Ra = 1.8×10^4 and $\theta_g = 90^\circ$, the flow is chaotic until 6,000 CTUs and then becomes periodic.

Figures 9A–D show the behavior of the four QOIs with Ra at three Pr for $\theta_g = 90^\circ$. Pr = 0.07 is denoted as green line, Pr = 0.71 is denoted as orange line, while Pr = 7.2 is denoted as blue line. Window W1 is denoted by a solid line, W2 by a dashed line, and W3 by a dotted line.

From Figures 9A–D we can see the following:

1) For Pr = 0.07 the QOIs are fairly consistent except for Ra $\in [3 \times 10^4, 5 \times 10^4]$.
2) For Pr = 0.71, we find large variability in Std(Nu), mean temperature fluctuation and mean TKE for Ra $> 2 \times 10^4$.
3) Pr = 7.2 exhibits the least variability.

For Ra where we find that the QOI variability is high, we have also examined the mean flow at multiple time windows and found that those are also inconsistent. In all cases, the mean Nu is quite repeatable.

Figures 9E–H show the behavior of the same QOIs as a function of θ_g. For Pr = 7.2, we consider three different values of Ra. We find for most of the θ_g, the variability is small except for small θ_g where we also report spatio-temporal chaotic flow. For Pr = 0.07, we find the QOIs has small variability with Ra = 1×10^4. However, for Pr = 0.71, we find large variability, especially in the mean TKE. Not only it has spatio-temporal chaotic flow (for example $\theta_g = 100^\circ$), but also the solution manifold is not smooth. By varying $\theta_g = 80^\circ$ to $\theta_g = 100^\circ$, the solution changes from steady to periodic then spatio-temporal chaotic.

From Figures 9E–H we observe the following:

1) For Pr = 0.07, Std(Nu) exhibits up to 50% variability (e.g., at $\theta_g = 70^\circ$) while $\langle T_{\text{fluc}} \rangle$ and ⟨TKE⟩ have orders-of-magnitude relative variability at $\theta_g = 0^\circ$.
2) For Pr = 0.71, $\langle T_{\text{fluc}} \rangle$ and ⟨TKE⟩ exhibit significant variability for $\theta_g \in [60^\circ, 110^\circ]$.
3) For Pr = 7.2, the most notable variation is at $\theta_g = 30^\circ$ for Std(Nu), $\langle T_{\text{fluc}} \rangle$, and ⟨TKE⟩ at Ra = 10^4. Remarkably, the higher Rayleigh number cases do not exhibit as much variance.

As in Figures 9A–D, the mean Nu is seen to be a repeatable QOI. It is worth noting the real challenge and sensitivity of this class of problems is illustrated in Figure 9H. Here, we observe for the (Pr, Ra) = (0.71, 10^4) case that the flow alternates from steady

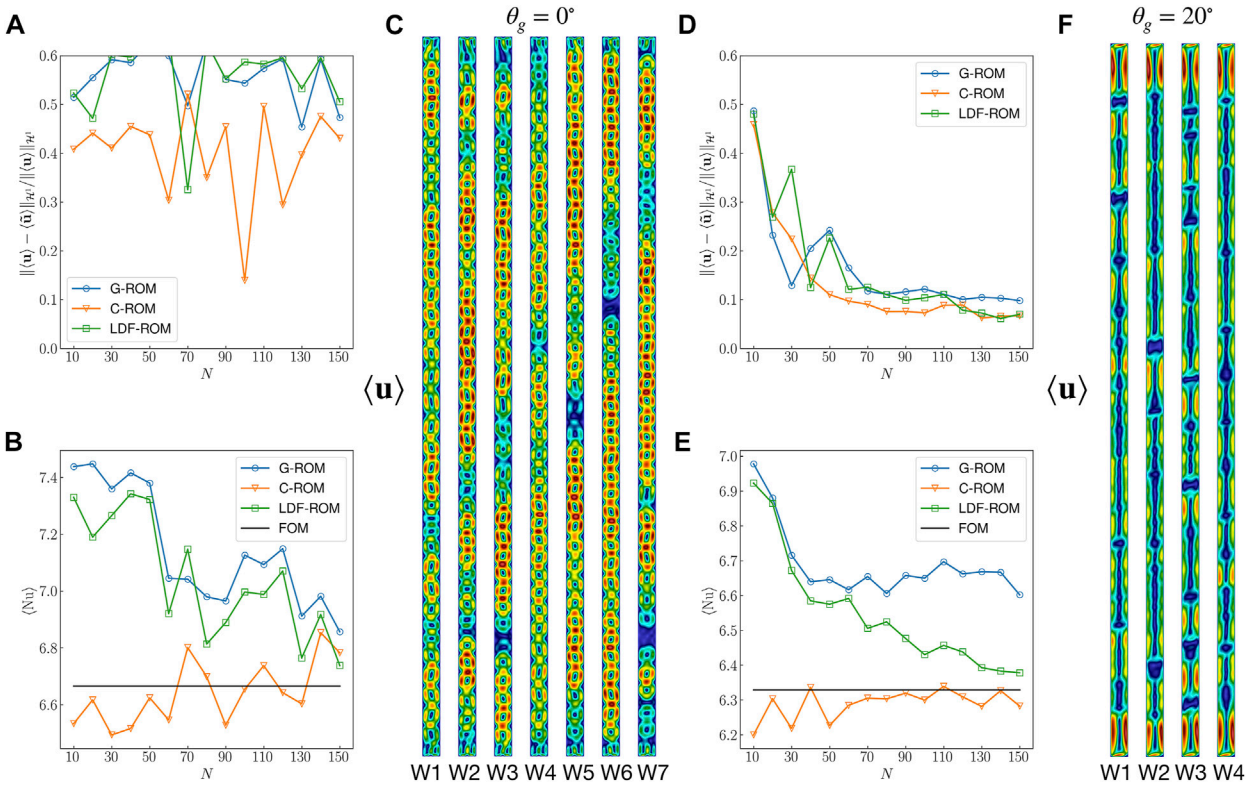


FIGURE 10
ROM performance comparison between problem having significant ($\theta_g = 0^\circ$) and less significant ($\theta_g = 20^\circ$) spatio-temporal chaotic flow at Pr = 7.2 and Ra = 3×10^5. **(A–C)** Behavior of the relative $\mathcal{H}^1$ error and mean Nu as a functions of N and magnitude of the mean velocity at multiple time windows W1, . . . , W7 at $\theta_g = 0^\circ$. **(D–F)** performance for the same quantities at $\theta_g = 20^\circ$.

to unsteady at multiple points along the one-dimensional θ_g parameter space as indicated by several distinct zeros in the TKE.In Figure 10, we further explore the influence of spatio-temporal chaos by examining the mean-flow distributions and ROM performance for (Pr, Ra) = (7.2, 3.5×10^4) at $\theta_g = 0^\circ$ and $\theta_g = 20^\circ$. Figures 10A–C show the behavior of the $\mathcal{H}^1$ error and mean Nu predictions as a function of N, along with the mean-velocity magnitude distributions over seven time windows for $\theta_g = 0^\circ$. Figures 10D–F show the same quantities but with $\theta_g = 20^\circ$ and only three time windows. In Figure 10C, we observe that for $\theta_g = 0^\circ$ the number of rolls in the mean velocity field changes across different time windows. Similar changes are observed, to a lesser extent, at $\theta_g = 20^\circ$. Hence, both solutions are categorized as spatio-temporal chaotic flow, but the $\theta_g = 0^\circ$ case is more significant. Comparing the mean flow error and mean Nu, we observe that the ROM convergence for the reproduction problem is slower (or nonexistent) at $\theta_g = 0^\circ$, while the convergence behavior is more favorable at $\theta_g = 20^\circ$.

We have also computed the relative error between FOM mean flows across seven time windows for the two values of θ_g. The maximum relative $\mathcal{H}^1$ error is 34% for $\theta_g = 0^\circ$ and 10% for $\theta_g = 20^\circ$. These FOM discrepancies can be considered as a bound on the predictive capabilities of the ROM. Indeed, the values of 34 and 10% are consistent with the lower bounds realized in Figures 10A,D.

In Figure 11, we examine the influence of Prandtl number by comparing results for Pr = 0.71 and 7.2 at (Ra, θ_g)= (3×10^5, 90°). Figures 11A–D show the convergence behavior for the $\mathcal{H}^1$ error and mean Nu as well as mean temperature and x-velocity fields at three time windows for Pr = 0.71, while Figures 11E–H show the same quantities for Pr = 7.2. From the mean fields, we observe that the number of rolls and its position changes with time window at Pr = 0.71, while minimal variance is observed at Pr = 7.2. Hence the solution at Pr = 0.71 is considered to be spatio-temporal chaotic while only chaotic at Pr = 7.2. Comparing the behavior of the relative $\mathcal{H}^1$ error in the mean flow and mean Nu, we observe convergence issues in the ROM at Pr = 0.71, while the same metrics converge at Pr = 7.2. We further compute the relative variance between two FOM mean flows across seven time windows for the two considered θ_g. The maximum relative $\mathcal{H}^1$ error is 14% for

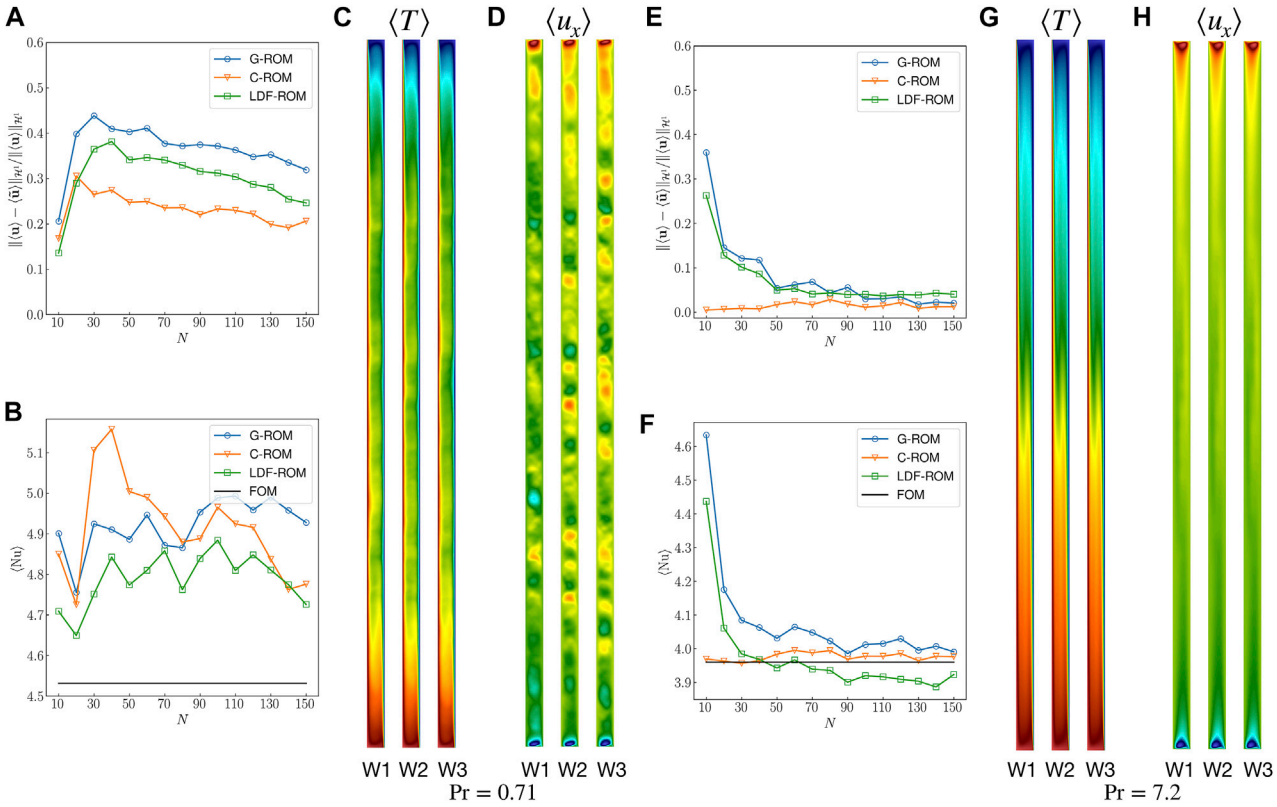


FIGURE 11
ROM performance comparison between Pr = 0.71 and Pr = 7.2 at Ra = 3 × 10^5 and $\theta_g = 90^\circ$. **(A–D)** Behavior of the relative $\mathcal{H}^1$ error and mean Nu as a functions of N, temperature and mean x-velocity at three time windows, W1, W2 and W3 at Pr = 0.71. **(E–H)**: Performance for the same quantities at Pr = 7.2.

Pr = 0.71 and only 1% for Pr = 7.2. This again explains why the approximation errors are different between the two cases. Considering these variance levels as a lower bound for the ROM, we can view 20% error in C-ROM as acceptable. On the other hand, the approximation error for Pr = 7.2 is able to reach below 10%.

In summary, the results of this section show that convergence issues and variations in the QOIs in the ROM can have high a correlation with the flow being spatio-temporal chaotic. From Table 1 and Figure 9, we see that Pr = 0.71 has a more complicated solution manifold found with the other two Prandtl numbers and also exhibits spatio-temporal chaos at a relatively small Rayleigh number, Ra = 10^4.

5.2 Multiple steady-state solutions

In this section, we report the existence of multiple steady-state solutions observed for the case (θ_g, Ra)=(0°, 10^4). The variations are characterized by different numbers of recirculation rolls, which are induced by using different initial conditions. Figure 12 shows steady temperature solutions generated by starting with steady solutions, χ_n from other values of $\theta_g = n^\circ$, save for the χ_{90} case, which corresponds to a single snapshot of the unsteady flow/temperature field at $\theta_g = 90^\circ$. Multiple steady states are also observed for this Prandtl number at $\theta_g = 10^\circ$ and 20° and have been reported by other authors as well [12, 14, 40].

For solution reproduction, the multiplicity of the solutions is not an issue as long as the ROM uses the same initial condition as the FOM. However, for parametric problem, these multiple states could easily lead to an incorrect (or at least, unexpected or unverifiable) conclusion. For example, if the ROM anchor at $\theta_g = 10^\circ$ is used to approximate the solution at $\theta_g = 0^\circ$. With the initial condition at $\theta_g = 10^\circ$, the approximate solution will be the third temperature solution shown in Figure 12. However, if one collects the FOM data at θ_g with zero initial condition, one will consider the first temperature solution as the truth solution. As we could consider those roll solutions as sine and

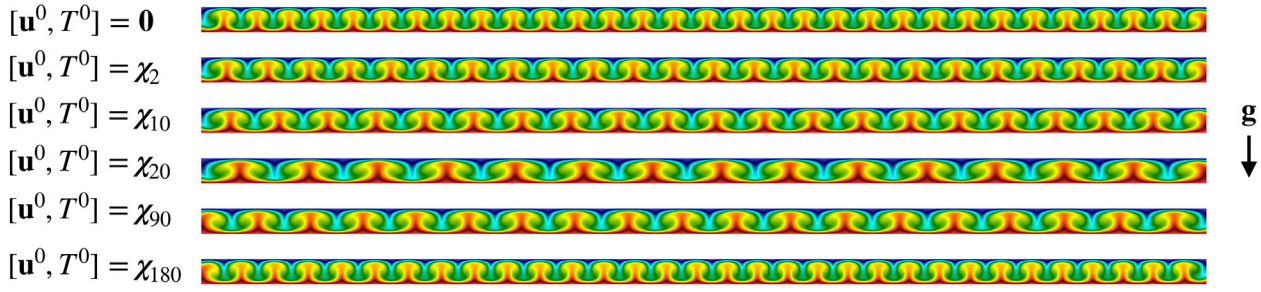


FIGURE 12
Different steady temperature solutions at $\theta_g = 0^\circ$, Ra = 1×10^4 and Pr = 0.71. The solutions were computed from different initial conditions, χ_n, corresponding to FOM solutions at $\theta_g = n^\circ$. For n = 2, 10, 20, and 180 the solution is steady, whereas χ_{90} is simply a snapshot from the $\theta_g = 90^\circ$ case.

cosine functions, the first and third solution are nearly orthogonal; their difference is $\mathcal{O}(1)$ and one would thus conclude that the pMOR had failed when it in fact had generated a valid solution.

5.3 Discussion summary

We have noted in Table 1 the broad range of flow regimes encountered for the tilted slot problem and in this section have illuminated a correlation between the flow states and predictive power of the MOR/pMOR framework. The cases with spatio-temporal chaos are generally the most challenging for model-order reduction and the pMOR errors are found to be (approximately) bounded from below by the variance observed in successive FOM simulations performed at the same parametric point. The development of the pMOR thus needs to be performed with care.

Two parameterizations were considered: 1) θ_g-variation, where a bifurcation exists and solution space is a blend of unsteady and steady solutions, and 2) Ra-variation, where no bifurcation exists and one finds only unsteady solutions. In the θ_g-variation problem, accurate prediction in mean flow and other QOIs by the pMOR was demonstrated in the Ra = 1×10^4 and Ra = 8×10^4 cases. In high Ra cases, acceptable prediction of Nu is achieved with LDF-ROM but a small mean-flow error is not realizable because of spatio-temporal chaos.

The results also indicate that the LDF-ROM is a better candidate for parametric problems with bifurcation than C-ROM. This observation is new, yet consistent with the results of [9], where the authors show that C-ROM is effective for parametric problems that do not have a bifurcation. For the parametric problem parameterized with Ra, without spatio-temporal chaos, we find that pMOR with any of the three methods, G-ROM, C-ROM, or LDF-ROM, is able to predict the mean flow quite well. In this case, C-ROM is the most accurate in mean flow prediction and other targeted QOIs. This result is not surprising given that the solution manifold does not have a bifurcation. Lastly, we remark that Std(Nu) is generally the most challenging QOI of those explored here.

From the results, we are able to make an important observation. For parametric problems where pMOR is successful (e.g., errors < 10%), the solution is either only chaotic (e.g., Ra variation with $\theta_g = 90^\circ$) or the solution does not have significant spatio-temporal chaos (e.g., θ_g variation with Ra = 1×10^4, 8×10^4). Once the spatio-temporal chaos becomes significant, the predictive power of pMOR deteriorates and the maximum errors are dominated by variance in the truth solution.

Although not shown here, we have also applied POD-*p*Greedy to this problem. In the parametric problem parameterized with θ_g, it works only in the steady case Ra = 1×10^3. Once the unsteady solution emerges, for example at Ra = 1×10^4, combining modes associated with different values of θ_g leads to an unstable ROM even with the Leray regularization. Although no rigorous proof is given, we hypothesize that the issue is due to the bifurcation in solution behavior. This point was also suggested in [9], which empirically showed that combining modes associated with qualitatively different behaviors might lead to poor prediction. By contrast, when the current problem is parameterized with Ra we find that POD-*p*Greedy is more efficient than the *h*-refinement approach.

6 Conclusion

In this paper an error-indicated pMOR is applied to a 2D unsteady natural convection in a tilted high-aspect ratio slot.

We first considered the solution reproduction problem (non-predictive case) to demonstrate the convergence of the ROMs and the effectiveness of the stabilization methods. We next addressed the parametric problem (predictive case) to validate the error indicator and, more broadly, the stabilized POD-*h*Greedy procedures. Principal contributions include, 1) extension of the error indicator proposed in [9] to buoyancy-driven flows; 2) demonstration that Leray-regularized Galerkin ROMs provide a robust solution approach for this class of flows; 3) identification of spatio-temporal chaos as a source of irreproducibility in both the FOM and the ROM and that the variance in the FOM provides a lower bound on the pMOR error in these cases; 4) observation that accurate prediction ($<10\%$) with pMOR is achievable if the solution in the parametric space is either only chaotic or the spatio-temporal chaos is not significant, regardless of whether a bifurcation exists or not. Once the spatio-temporal chaos becomes significant, the performance of the pMOR deteriorates and the maximum errors of the mean flow and QOIs are dominated by the flow chaos.

We also highlight a number of challenges that are particularly relevant for buoyancy-driven flows and which should be taken into consideration in the design of pMOR strategies for 3D buoyancy driven turbulent flow. First, one needs to be aware of potential convergence issues for the mean flow and other QOI predictions when the FOM exhibits large-scale spatio-temporal chaos. Second, it is difficult to combine modes associated with different flow regimes, especially for the *p*Greedy case. Third, even relatively simple (e.g., steady) flows can exhibit multiple states at a given parameter. And fourth, there are large offline costs both in terms of computational time and required storage for error indicator and $O(N^4)$ costs for online-only error indicators[5].

We outline potential next steps in pMOR development for this class of problems.

1) *Extension to higher dimensional parameter space.* In this work, we considered only one-dimensional parameter space since the pMOR behavior needed to be carefully diagnosed; however higher dimensional parameter spaces are more interesting for engineering applications.
2) *hp-Greedy with a bifurcation detection technique.* Although we find LDF-ROM is more efficient than C-ROM for parametric problems that have a bifurcation, the *h*-refinement strategy considered in this paper might require an infeasible number of offline simulations as the dimension of the parameter space increases. To tackle complex parametrizations, more advanced sampling strategies that combine *h*- and *p*-refinement [37], potentially with bifurcation detection, could be beneficial [42].

5 The $O(N^4)$ costs, which arise from the rank-3 advection tensor, might be mitigated by an $O(N^2)$ approximation to the advection operator, such as suggested in [41].

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding authors.

Author contributions

P-HT derived the formulation and error indicator, performed the numerical simulations, collected the data, analyzed and interpreted the results. PF supervised the project, provided critical feedback and helped on interpreted the results. Both authors prepared the manuscript.

Funding

This research is supported by the DOE Office of Nuclear Energy under the Nuclear Energy University Program (Proj. No. DE_NE0008780). Simulations were performed at the DOE Office of Science User Facility ALCF (Argonne Leadership Computing Facility).

Acknowledgments

We thank Anthony T. Patera (MIT) for his valuable comments, many contributions, insights and guidelines.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

1. Grepl MA, Patera AT. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: Math Model Numer Anal* (2005) 39:157–81. doi:10.1051/m2an:2005006

2. Rozza G, Huynh DBP, Patera AT. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch Comput Methods Eng* (2008) 15:229–75. doi:10.1007/s11831-008-9019-9

3. Merzari E, Ninokata H, Mahmood A, Rohde M. Proper orthogonal decomposition of the flow in geometries containing a narrow gap. *Theor Comput Fluid Dyn* (2009) 23:333–51. doi:10.1007/s00162-009-0152-3

4. Merzari E, Pointer WD, Fischer P. A POD-based solver for the advection-diffusion equation. *Fluids Eng Division Summer Meet* (2011) 44403:1139–47.

5. Quarteroni A, Rozza G, Manzoni A. Certified reduced basis approximation for parametrized partial differential equations and applications. *J Math Ind* (2011) 1:3. doi:10.1186/2190-5983-1-3

6. Wang Z, Akhtar I, Borggaard J, Iliescu T. Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison. *Computer Methods Appl Mech Eng* (2012) 237:10–26. doi:10.1016/j.cma.2012.04.015

7. Kaneko K, Fischer P. Augmented reduced order models for turbulence. *Front Phys* (Forthcoming 2022).

8. Quarteroni A, Manzoni A, Negri F. *Reduced basis methods for partial differential equations: An introduction*. Switzerland: Springer (2015).

9. Fick L, Maday Y, Patera AT, Taddei T. A stabilized POD model for turbulent flows over a range of Reynolds numbers: Optimal parameter sampling and constrained projection. *J Comput Phys* (2018) 371:214–43. doi:10.1016/j.jcp.2018.05.027

10. Ngoc Cuong N, Veroy K, Patera AT. Certified real-time solution of parametrized partial differential equations. In: *Handbook of materials modeling*. Dordrecht, Netherlands: Springer (2005). p. 1529–64.

11. Veroy K, Patera AT. Certified real-time solution of the parametrized steady incompressible Navier–Stokes equations: Rigorous reduced-basis a posteriori error bounds. *Int J Numer Methods Fluids* (2005) 47:773–88. doi:10.1002/fld.867

12. Wang Q, Wan ZH, Yan R, Sun DJ. Multiple states and heat transfer in two-dimensional tilted convection with large aspect ratios. *Phys Rev Fluids* (2018) 3:113503. doi:10.1103/physrevfluids.3.113503

13. Deparis S. Reduced basis error bound computation of parameter-dependent Navier–Stokes equations by the natural norm approach. *SIAM J Numer Anal* (2008) 46:2039–67. doi:10.1137/060674181

14. Deparis S, Rozza G. Reduced basis method for multi-parameter-dependent steady Navier–Stokes equations: Applications to natural convection in a cavity. *J Comput Phys* (2009) 228:4359–78. doi:10.1016/j.jcp.2009.03.008

15. Ballarin F, Rebollo TC, Ávila ED, Mármol MG, Rozza G. Certified Reduced Basis VMS-Smagorinsky model for natural convection flow in a cavity with variable height. *Comput Mathematics Appl* (2020) 80:973–89. doi:10.1016/j.camwa.2020.05.013

16. Knezevic DJ, Nguyen NC, Patera AT. Reduced basis approximation and a posteriori error estimation for the parametrized unsteady Boussinesq equations. *Math Models Methods Appl Sci* (2011) 21:1415–42. doi:10.1142/s0218202511005441

17. Yano M. A space-time Petrov–Galerkin certified reduced basis method: Application to the Boussinesq equations. *SIAM J Sci Comput* (2014) 36:A232–66. doi:10.1137/120903300

18. Kaneko K, Tsai PH, Fischer P. Towards model order reduction for fluid-thermal analysis. *Nucl Eng Des* (2020) 370:110866. doi:10.1016/j.nucengdes.2020.110866

19. Wells D, Wang Z, Xie X, Iliescu T. An evolve-then-filter regularized reduced order model for convection-dominated flows. *Int J Numer Methods Fluids* (2017) 84:598–615. doi:10.1002/fld.4363

20. Karimi A, Paul MR. Quantifying spatiotemporal chaos in Rayleigh-Bénard convection. *Phys Rev E* (2012) 85:046201. doi:10.1103/physreve.85.046201

21. Wolf A, Swift JB, Swinney HL, Vastano JA. Determining Lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena* (1985) 16:285–317. doi:10.1016/0167-2789(85)90011-9

22. Goldhirsch I, Sulem PL, Orszag SA. Stability and Lyapunov stability of dynamical systems: A differential approach and a numerical method. *Physica D: Nonlinear Phenomena* (1987) 27:311–37. doi:10.1016/0167-2789(87)90034-0

23. Cross M, Hohenberg P. Spatiotemporal chaos. *Science* (1994) 263:1569–70. doi:10.1126/science.263.5153.1569

24. Egolf DA, Melnikov IV, Pesch W, Ecke RE. Mechanisms of extensive spatiotemporal chaos in Rayleigh–Bénard convection. *Nature* (2000) 404:733–6. doi:10.1038/35008013

25. Cross MC, Hohenberg PC. Pattern formation outside of equilibrium. *Rev Mod Phys* (1993) 65:851–1112. doi:10.1103/revmodphys.65.851

26. Busse F. Non-linear properties of thermal convection. *Rep Prog Phys* (1978) 41:1929–67. doi:10.1088/0034-4885/41/12/003

27. Maday Y, Patera AT, Ronquist EM. A well-posed optimal spectral element approximation for the Stokes problem. *Tech Rep* (1987) 1987.

28. Fischer PF, Lottes JW, Kerkemeier SG. *Nek5000 web page* (2008).

29. Quarteroni A, Valli A. Numerical approximation of partial differential equations. In: *Springer series in computational mathematics*. Berlin: Springer (1994).

30. Fischer P, Schmitt M, Tomboulides A. Recent developments in spectral element simulations of moving-domain problems. In: *Recent progress and modern challenges in applied mathematics, modeling and computational science*. Berlin: Springer (2017). p. 213–44.

31. Rønquist EM, Patera AT. A Legendre spectral element method for the Stefan problem. *Int J Numer Methods Eng* (1987) 24:2273–99. doi:10.1002/nme.1620241204

32. Maday Y, Patera AT, Rønquist EM. *The PN × PN−2 method for the approximation of the Stokes problem*. Paris: Laboratoire d'Analyse Numérique (1992).

33. Guermond JL, Oden JT, Prudhomme S. Mathematical perspectives on large eddy simulation models for turbulent flows. *J Math Fluid Mech* (2004) 6:194–248. doi:10.1007/s00021-003-0091-5

34. Guermond JL, Pasquetti R, Popov B. Entropy viscosity method for nonlinear conservation laws. *J Comput Phys* (2011) 230:4248–67. doi:10.1016/j.jcp.2010.11.043

35. Mullen J. *Development of a parallel spectral element based large eddy simulation model for the flow of incompressible fluids in complex geometries*. Ph.D. thesis. Providence, Rhode Island: Brown University (1999).

36. Sabetghadam F, Jafarpour A. α regularization of the POD-Galerkin dynamical systems of the Kuramoto–Sivashinsky equation. *Appl Mathematics Comput* (2012) 218:6012–26. doi:10.1016/j.amc.2011.11.083

37. Eftang JL, Knezevic DJ, Patera AT. An hp certified reduced basis method for parametrized parabolic partial differential equations. *Math Computer Model Dynamical Syst* (2011) 17:395–422. doi:10.1080/13873954.2011.547670

38. Haasdonk B, Ohlberger M. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: Math Model Numer Anal* (2008) 42:277–302. doi:10.1051/m2an:2008001

39. Haasdonk B. Convergence rates of the POD–Greedy method. *ESAIM: Math Model Numer Anal* (2013) 47:859–73. doi:10.1051/m2an/2012045

40. Gelfgat AY, Bar-Yoseph P, Yarin A. Stability of multiple steady states of convection in laterally heated cavities. *J Fluid Mech* (1999) 388:315–34. doi:10.1017/s0022112099004796

41. Barrault M, Maday Y, Nguyen NC, Patera AT. An empirical interpolation method: Application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique* (2004) 339:667–72. doi:10.1016/j.crma.2004.08.006

42. Pichi F. *Reduced order models for parametric bifurcation problems in nonlinear PDEs*. Ph.D. thesis. Trieste, Italy: Scuola Internazionale Superiore di Studi Avanzati (2020).

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 29 September 2022
DOI 10.3389/fphy.2022.905392

OPEN ACCESS

EDITED BY
Traian Iliescu,
Virginia Tech, United States

REVIEWED BY
Imran Akhtar,
National University of Sciences and Technology (NUST), Pakistan
Birgul Koc,
Sevilla University, Spain

*CORRESPONDENCE
Kento Kaneko,
kaneko2@illinois.edu

SPECIALTY SECTION
This article was submitted to Statistical and Computational Physics, a section of the journal Frontiers in Physics

RECEIVED 27 March 2022
ACCEPTED 05 August 2022
PUBLISHED 29 September 2022

CITATION
Kaneko K and Fischer P (2022), Augmented reduced order models for turbulence.
Front. Phys. 10:905392.
doi: 10.3389/fphy.2022.905392

Augmented reduced order models for turbulence

Kento Kaneko[1]* and Paul Fischer[1,2,3]

[1]Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, Champaign, IL, United States, [2]Department of Computer Science, University of Illinois at Urbana-Champaign, Champaign, IL, United States, [3]Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL, United States

The authors introduce an augmented-basis method (ABM) to stabilize reduced-order models (ROMs) of turbulent incompressible flows. The method begins with standard basis functions derived from proper orthogonal decomposition (POD) of snapshot sets taken from a full-order model. These are then augmented with divergence-free projections of a subset of the nonlinear interaction terms that constitute a significant fraction of the time-derivative of the solution. The augmenting bases, which are rich in localized high wavenumber content, are better able to dissipate turbulent kinetic energy than the standard POD bases. Several examples illustrate that the ABM significantly out-performs L^2-, H^1- and Leray-stabilized POD ROM approaches. The ABM yields accuracy that is comparable to constraint-based stabilization approaches yet is suitable for parametric model-order reduction in which one uses the ROM to evaluate quantities of interests at parameter values that differ from those used to generate the full-order model snapshots. Several numerical experiments point to the importance of localized high wavenumber content in the generation of stable, accurate, and efficient ROMs for turbulent flows.

KEYWORDS

POD, ROM, pMOR, stabilization, turbulence

1 Introduction

Parametric model-order reduction (pMOR) is a promising approach to leveraging high-performance computing (HPC) for design and analysis in fluid-thermal engineering applications. The governing equations in this context are the time-dependent incompressible Navier-Stokes equations (NSE) and the thermal transport equation.

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\nabla p + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{f}, \qquad \nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

$$\partial_t T + \boldsymbol{u} \cdot \nabla T = \alpha \nabla^2 T. \tag{2}$$

Where ν and α parameterize the PDEs and the forcing function $\boldsymbol{f}$ can be the Boussinesq approximation term, for example.[1] The equations are assumed to hold in a suitable domain Ω with appropriate initial and boundary conditions. The Galerkin statement is.

$$\textit{Find } (\partial_t \boldsymbol{u}, p, \partial_t T) \in Y \coloneqq [\mathbf{H_0^1} \otimes L^2 \otimes H_0^1] \textit{ s.t. } \forall (\boldsymbol{v}, q, S) \in Y.$$

$$(\boldsymbol{v}, \partial_t \boldsymbol{u}) + (\boldsymbol{v}, \boldsymbol{u} \cdot \nabla \boldsymbol{u}) = (\nabla \cdot \boldsymbol{v}, p) - \nu (\nabla \boldsymbol{v}, \nabla \boldsymbol{u}) + (\boldsymbol{v}, \boldsymbol{f}),$$
$$(q, \nabla \cdot \boldsymbol{u}) = 0, \tag{3}$$
$$(S, \partial_t T) + (S, \boldsymbol{u} \cdot \nabla T) = -\alpha (\nabla S, \nabla T). \tag{4}$$

Here, L^2 is the space of square-integrable functions on Ω; H^1 is the space of functions in L^2 whose gradient is also in L^2; and H_0^1 is the space of functions in H^1 that vanish on subsets of the boundary, $\partial\Omega_D \subset \partial\Omega$, where homogeneous Dirichlet conditions are imposed. $\mathbf{H_0^1}$ is the vector counterpart to H_0^1.

To obtain a fully-accurate quantity of interest (QOI) such as friction factor, Nusselt number, or Strouhal number, one formally needs to obtain a full-order model (FOM) solution to the governing equations at discrete points in the parameter space of interest (e.g., spanned by a range of ν and α, of interest). Typically, the FOM constitutes a high-fidelity spectral- or finite-element solution to the governing equations, which can be expensive to solve, particularly for high Reynolds number cases that are typical of engineering applications. pMOR seeks to develop a sequence of *reduced-order models* (ROMs) that capture the behavior of the FOM and allow for parameter variation. For unsteady flows, the pMOR problem can be broken down into two subproblems: *reproduction*, wherein the ROM captures essential time-transient behavior of the FOM using the same parameter (anchor) point for each, and *parametric variation*, wherein the ROM is run at a *different* parametric point in order to predict the system behavior away from the anchor points at which the FOM simulation was conducted.

In this work, we focus primarily on the reproduction problem for challenging unsteady flows. We do, however, also consider pMOR, which we illustrate with an example from [1]. The thermal-fluids problem is the axisymmetric Rayleigh-Bénard configuration depicted in Figure 1A, which was studied by Tuckerman and Barkley [2,3]. The problem is parameterized by $\epsilon = \frac{\mathrm{Ra}-\mathrm{Ra_c}}{\mathrm{Ra_c}}$, where $\mathrm{Ra_c} = 1734$ is the critical Rayleigh number. The 2D axisymmetric domain has an aspect ratio of $\Gamma = 5$, shown in Figure 1A. For $\epsilon > 1.3843$, traveling waves move towards the centerline axis with a period that depends on ϵ. We perform FOM calculations at two anchor points, $\epsilon = 1.6$ and $\epsilon = 2.6$, from which we collect snapshots (full flow/temperature fields). We apply proper orthogonal decomposition (POD) to the snapshot sets from each of the FOMs and use 20 POD modes from each to form a reduced-order subspace Z^N comprising $N = 40$ basis functions. These modes are used in the weak- (Galerkin-) formulation of the governing equations, where the solution is restricted to $Z^N \subset Y$. The low-dimensional ROM is able to capture short- and long-time behavior as shown in the Nusselt number reproduction traces in Figure 1B. Moreover, as shown in Figure 1C, the pMOR is able to accurately predict the period of the traveling wave solutions both inside and outside the ϵ range spanned by the anchor points. Note that as $\epsilon \to 1.3843$, the period goes to infinity and FOM simulations near this limit become intractable. The ROM, however, is able to predict this critical value of ϵ to within a few percent.

While pMOR is a promising approach for engineering analysis and design, it is well known that even the reproduction problem is challenging for the classical POD-Galerkin approach at high Reynolds numbers after the flow transitions to turbulence. One common issue with this class of problems is that the ROM solution approaches an unphysical attractor. This behavior is attributed to a lack of dissipation, given that the truncated POD space lacks high-wavenumber modes that are capable of dissipating energy. One can induce additional dissipation by including more modes but the cost is high. The convective tensor reduction requires storage of N^3 entries for the advection operator with a corresponding work of $2N^3$ operations per timestep. While $N = 100$, with a cost of a two million operations per step and a million words in memory, may be tolerable, $N = 400$ with a cost of 128 million operations and 64 million words quickly makes pMOR less viable for running on a workstation, which is typically the target for this type of analysis tool.

Existing techniques for addressing the computational cost include the discrete empirical interpolation method (DEIM) [4], which effectively interpolates the convective term, and tensor decomposition, which aims to approximate the convective tensor by a low-rank tensor. We will show in our concluding examples that these methods will not, on their own, address the unphysical ROM dynamics. Stabilization of the ROM is critical and is the primary topic of this work. Several stabilization strategies are described in Section 2. The major contribution of this work is the development of a novel augmented-basis method (ABM), in which we add important modes to the standard POD bases. In many cases, the ABM increases both the stability and accuracy of the ROMs at a cost equivalent to standard POD approaches having the same total number of modes.

2 Background

The POD-Galerkin technique in fluid flow emerged from work to identify dominant flow features by [5] Model-reduction using POD modes as basis functions was introduced afterwards,

1 These equations are effectively in nondimensional form, which for forced conditions implies that $\nu = \mathrm{Re}^{-1}$, the inverse Reynolds number, and $\alpha = Pe^{-1}$, the inverse Peclet number. For buoyancy-driven flows these parameters typically scale with Rayleigh number (*Ra*) and Prandtl number (Pr), with the precise definition dependent on the chosen scaling.

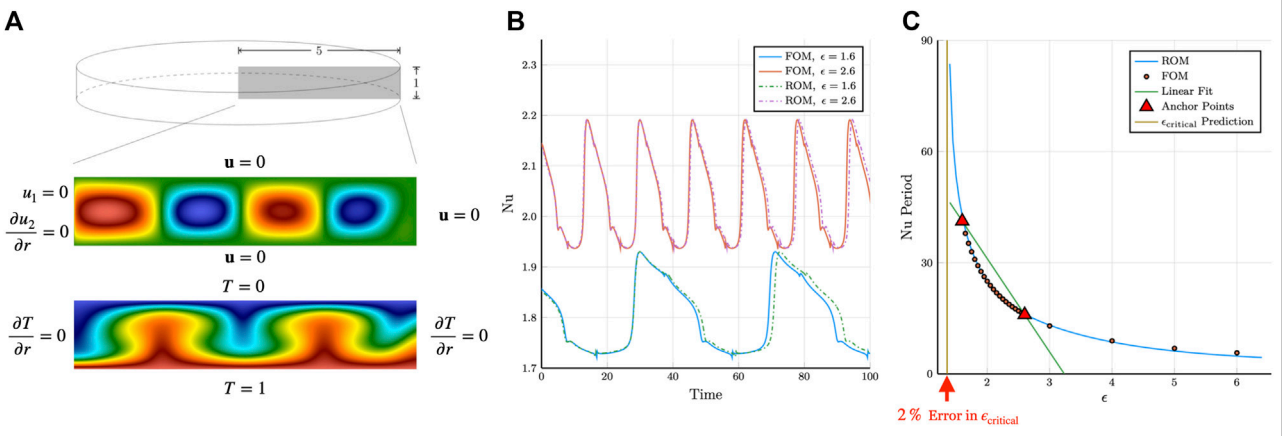


FIGURE 1
ROM application to axisymmetric Rayleigh–Bénard.

with a comprehensive analysis appearing in a later monograph by [5].

More complex PDEs with non-affine parameter dependencies were addressed by [6] using decomposition of the nonlinearity based on the empirical interpolation method (EIM). In this approach, successive interpolation modes are chosen to eliminate the error between the targeted term in FOM and the ROM at "magic points" designated as points in Ω where the error of the current interpolant (i.e., the approximant of the next mode) is maximal. This method was further extended by [4] with a POD decomposition of the nonlinear term and the choice of points restricted to discrete points produced by the spatial discretization of the PDE, called discrete empirical interpolation method (DEIM). While these methods enable application of pMOR to nonlinear problems the issue of insufficient dissipation and feasible stabilization for the NSE persists.

Due to its approach of treating nonlinear terms, DEIM has the potential to address the high-cost issue of including more modes. DEIM replaces the third-order convective tensor with a collocation-like decomposition at the discrete magic points, which yields a reduction of computational complexity from $O(N^3)$ to $O(N^2)$. Accounting for the constants, evaluation of advection using DEIM with $N = 200$ modes would be equivalent to using the full tensor with $N = 65$. For the same cost, DEIM thus permits the use of a richer approximation space.

To certify that the error in the ROMs that are produced is smaller than the acceptable tolerance, error indicators have been developed based on the residual of the ROM solutions in the full-order model (FOM) space. Error indicators for coercive elliptic PDEs are described by [7]. An a posteriori error indicator for time-dependent NSE is described by [8] which is described as the dual-norm of the residual of the time-averaged momentum equation. This error indicator is evaluated by accumulation of residual contributions from each term in the momentum equation at each time-step. This metric provides an error estimate for time-dependent ROM solutions, while not a strict bound on the error, allows an efficient selection of anchor point selection for pMOR. We do not consider these further here, but they are an important component for efficient pMOR and are discussed in a companion paper [9].

For addressing the issue of stability, several modifications to the original POD Galerkin approach have been proposed [10]. proposed a modification of the POD mode generation in which the H^1 inner-product is used to produce the Gramian, rather than L^2 inner-product, to emphasize the importance of gradients in the FOM snapshots [11]. introduced Leray regularization in the context of ROM in which the *advecting* field is smoothed (conveniently, by truncation of the modes in the case of POD-ROM). This regularization enhances the stability property of the dynamical behavior; however, the optimal choice of regularization (e.g., number of modes to truncate or shape of transfer function) is not known *a priori*.

An alternative stabilization approach, introduced by [8], is to replace the discrete ODE system by a constrained minimization problem at each timestep. During the evolution of the system, the basis coefficients are bound by the minimum and maximum coefficient values observed in the snapshot projection onto the truncated POD space. (If the constraints are inactive, one recovers the standard Galerkin-based trajectory.) With this approach, the ROM tends to stay close to the dynamics of the FOM. A challenge, however, is that this approach requires ad hoc modification of the bounds for parametric values where the FOM snapshots are unavailable. Applications of several of these stabilization techniques may be found in [1].

Methods that address the stability issue by constructing basis functions that satisfy the energy-balance that closely match the POD basis is introduced by [12]. In this work, existing work on stabilizing linear time-invariant (LTI) systems by [13], which seeks optimal combinations of snapshots to produce dynamically stable ROMs, is combined with work by [14], in which the kinetic energy behavior is stabilized by introducing an empirical turbulence closure term in the ROM. In this combined method by Balajewicz, in a preprocessing step, an *a priori* nonlinear constrained optimization problem is solved that minimizes the difference between the energy captured by the new modes and the energy captured by the POD modes for a given N subject to constraints: the columns of the transformation matrix are orthonormal and the empirical kinetic energy-balance is satisfied. The result is a set of transformed POD modes that are augmented directly with dissipation modes (also taken as linear combinations of the snapshots). The author demonstrated that this approach offers significant improvement over the standard POD-Galerkin approach for a 2D lid-driven cavity problem and a 2D mixing layer. Also discovered was the fact that by ensuring the kinetic energy-balance is dissipative to an arbitrary degree, the ROM solution becomes stabilized. Thus, by encapsulating the method by this ROM training stage, the amount of appropriate dissipation to be prescribed in the constrained optimization step can be found to produce a stable ROM with mean TKE behavior close to that of the FOM.

Another basis augmentation approach, introduced by [15,16], uses a combination of L^2 POD modes and H^1 POD modes that are subsets of the originating snapshot set. The idea is to have a small number of L^2 POD modes capture the dominant energy-carrying features of the flow while the H^1 POD modes (containing small-scale features) provide the necessary dissipation that is not realized by the L^2 POD modes alone. The authors successfully applied this augmented basis to 3D turbulent flow in injectors.

In the next section, we introduce a novel augmented basis method, that is designed to address accuracy and stability of ROMs for turbulent flow. Rather than drawing upon the snapshot set, the augmenting vectors are derived from the nonlinear interaction terms that directly influence the time-derivative of the NSE. This augmented basis set does not require extensive training (i.e., ROM-parameter optimization) and can be used within a standard Galerkin-ROM setting.

3 Augmented basis method

To motivate the ABM, we consider the Leray-projected form of the NSE, in which the velocity field evolution is described as:

$$\partial_t \boldsymbol{u} = \mathbb{P}\left[-\boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nu\nabla^2\boldsymbol{u}\right]. \tag{5}$$

Here, the pressure has been formally eliminated and its effects are represented by an abstract operator, $\mathbb{P}$, sometimes called the Leray projector, which will project the operand onto a space of divergence-free fields. While the Leray projector is a projection using the H^1 inner-product, we will use the L^2 inner-product for our definition. For the discretized system, particularly with the $P_N P_{N-2}$ spectral element discretization, this operator is well-defined [17].

For the spectral element method (SEM), we look to find the solution in a finite-dimensional space, $X^{\mathcal{N}}$, comprising piecewise Nth-order tensor-product polynomial bases mapped from a reference unit cube to each of E spectral elements, for a total of $\mathcal{N} \approx EN^3$ degrees-of-freedom per field (in 3D). Finally, in the POD-Galerkin approach, we restrict our attention to solutions $Z^N \subset X^{\mathcal{N}}$, where the basis is generally formed from a proper orthogonal decomposition of a sequence of SEM solution snapshots, using the method introduced by [18].

The method of snapshots forms a basis from a linear combination of FOM solution fields (each involving $O(\mathcal{N})$ spectral element basis coefficients). One forms the Gramian matrix, whose first N eigenvectors (ranked by eigenvalues from largest to smallest) are used to determine the linear combination of the snapshots that forms the N-dimensional basis for the ROM approximation space, Z^N. Because the snapshots are (weakly) divergence-free, so are all elements of Z^N, which means that pressure drops out of the ordinary differential equation that governs the ROM. In this work, the velocity POD modes are denoted as $\boldsymbol{\zeta}_i$, and the thermal POD modes are denoted as θ_i. For both of these collections of modes, the $i = 0$ modes correspond to a lifting function that satisfies the boundary conditions and is always associated with a coefficient value of $u_0 = 1$ and $T_0 = 1$. The choice of the lifting function may be a solution to the Stokes problem, the Poisson equation, or the time-averaged solution. For the examples in Section 4, the lifting function is based on time-averaged FOM solutions. For the POD-ROM, the hierarchy of the spaces of interest is $Z^N \subset X^{\mathcal{N}} \subset Y$. For this work, we consider a FOM discretization that is well-resolved such that the projection error from Y to $X^{\mathcal{N}}$ is minimal. We next show how the Z^N space derived by the classical POD-Galerkin method can be augmented such that the time-evolution of the solution in the extended space better approximates the time-evolution of the solution in $X^{\mathcal{N}}$.

Assuming that the solution to Eq. 5 exists near t^*, we can describe the local temporal behavior through a Taylor-series expansion involving a linear combination of all time-derivatives.

$$\boldsymbol{u}(\boldsymbol{x}, t^* + \epsilon) = \boldsymbol{u}(\boldsymbol{x}, t^*) + \epsilon\,\partial_t \boldsymbol{u}(\boldsymbol{x}, t^*) + \ldots \tag{6}$$

$$= \underbrace{\boldsymbol{u}(\boldsymbol{x}, t^*)}_{\text{Snapshot}} + \epsilon\,\mathbb{P}\{-\boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nu\nabla^2\boldsymbol{u}\} + \ldots \tag{7}$$

Therefore, in addition to capturing the dominant modes of the snapshots, we propose to augment the POD basis set Z^N with modes that can accurately represent the order ϵ terms on the right-hand side of (7) in order to construct $\boldsymbol{u}$ (x, $t^* + \epsilon$). The consequence of not representing the $O(\epsilon)$ term is deviation in the

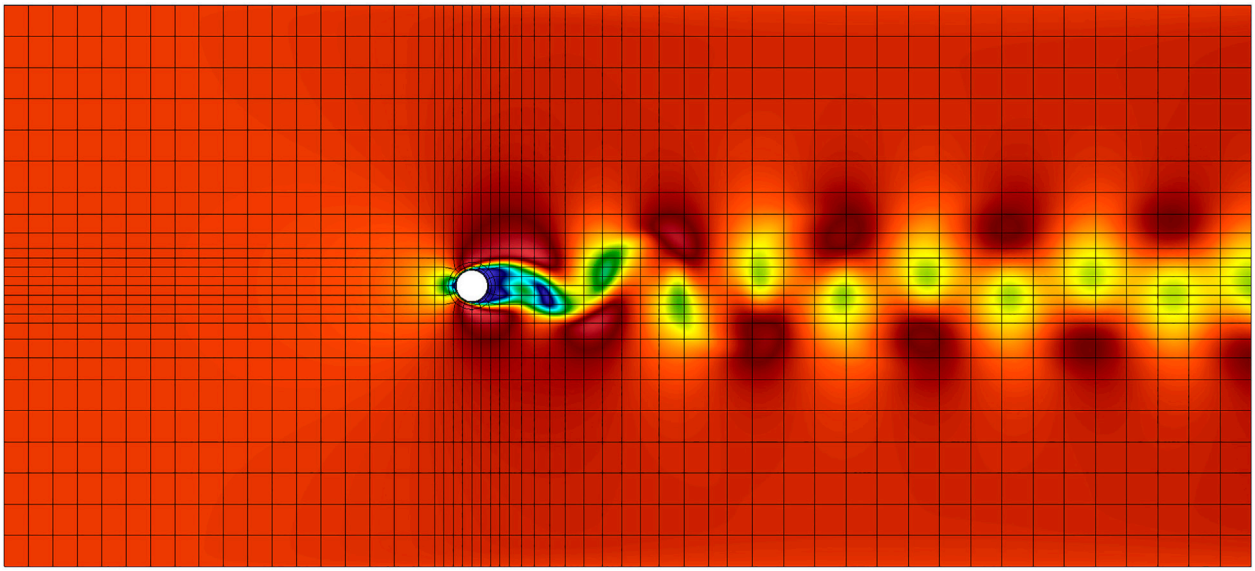


FIGURE 2
Velocity magnitude plot of a flow past a cylinder snapshot (Re = 100)

trajectory of the physical solution and the projected (Galerkin) solution.

Consider a solution $\boldsymbol{u}$ that lives in Z^N, meaning $\boldsymbol{u} = \sum_{i=0}^{N} u_i\, \boldsymbol{\zeta}_i$. Using (5), we can describe the time-derivative of the solution as

$$\begin{aligned}\partial_t \boldsymbol{u} &= \mathbb{P}\left[-\boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nu\nabla^2\boldsymbol{u}\right] \\ &= -\sum_{i,j=0}^{N} u_i u_j\, \mathbb{P}\left[\boldsymbol{\zeta}_i\cdot\nabla\boldsymbol{\zeta}_j\right] + \sum_{i=0}^{N} u_i\, \nu\mathbb{P}\left[\nabla^2\boldsymbol{\zeta}_i\right] \end{aligned} \tag{8}$$

Thus, we can accurately describe the time-derivative with $(N + 1)^2$ terms for the nonlinear term and $(N + 1)$ terms for the viscous operator.

We consider an example with Fourier basis to highlight this issue. When we have a band-limited solution state with the highest wavenumber k, the convection term would produce a solution at the next timestep of highest wavenumber $2k$, which does not live in the original space. Thus, the wavenumber $2k$ behavior is never observed in the evolution of the projected Galerkin system. With an augmentation of the basis with the high-wavenumber modes, we will face the same issue through lack of $4k$ mode representation. This issue is of course recursive. We are helped, however, by the fact that the higher wavenumber modes have higher rates of dissipation. Continuation of this process will therefore eventually yield only marginal returns in improved solution fidelity. We shall see, however, that addition of just a few modes can have a significant impact on the overall ROM performance.

Because of nonlinear advection, the solution will evolve outside the N-dimensional span of Z^N. We note that as the basis includes more fine-scale components, the convective contribution becomes small relative to the diffusive contribution; thus, the solution becomes closed as the minimal grid-size approaches 0, as is the case in FOM solvers (i.e., the exact solution is band-limited). In the POD-ROM, however, the basis is typically far from completing the relevant approximation space and the addition of the modes $\mathbb{P}[\boldsymbol{\zeta}_i\cdot\nabla\boldsymbol{\zeta}_j]$ and $\mathbb{P}[\nabla^2\boldsymbol{\zeta}_i]$ can provide an important first-order correction to Z^N.

For advection dominated problems, we can focus on the nonlinear contributions,

$$\partial_t \boldsymbol{u} \approx \mathbb{P}[\boldsymbol{u}\cdot\nabla\boldsymbol{u}] = \sum_{j,k=0}^{N} u_j u_k \mathbb{P}\left[\boldsymbol{\zeta}_k\cdot\nabla\boldsymbol{\zeta}_j\right] \tag{9}$$

Whenever we evolve the solution in the space Y, where the current solution lies in the truncated POD space Z^N, we see that the time derivative be reasonably represented with an additional $(N + 1)^2$ basis functions of the form $\boldsymbol{\phi}_{l=j+k(N+1)} = \mathbb{P}[\boldsymbol{\zeta}_k\cdot\nabla\boldsymbol{\zeta}_j]$. Obviously, this process is not closed, since more basis functions are required in the next timestep. Worse still, even starting with $\boldsymbol{u} \in Z^N$, the required number of additional basis functions will be $O(N^2)$ if we include all terms in (9), which comes with an $O(N^6)$ computational cost that is untenable, even for a small number of POD modes, N. We therefore seek to augment the original POD basis with subsets of these evolution basis that are most relevant to the dynamics.

The first subset captures the interaction between the lifting function, $\boldsymbol{\zeta}_0$, and all other modes. This choice ensures that both

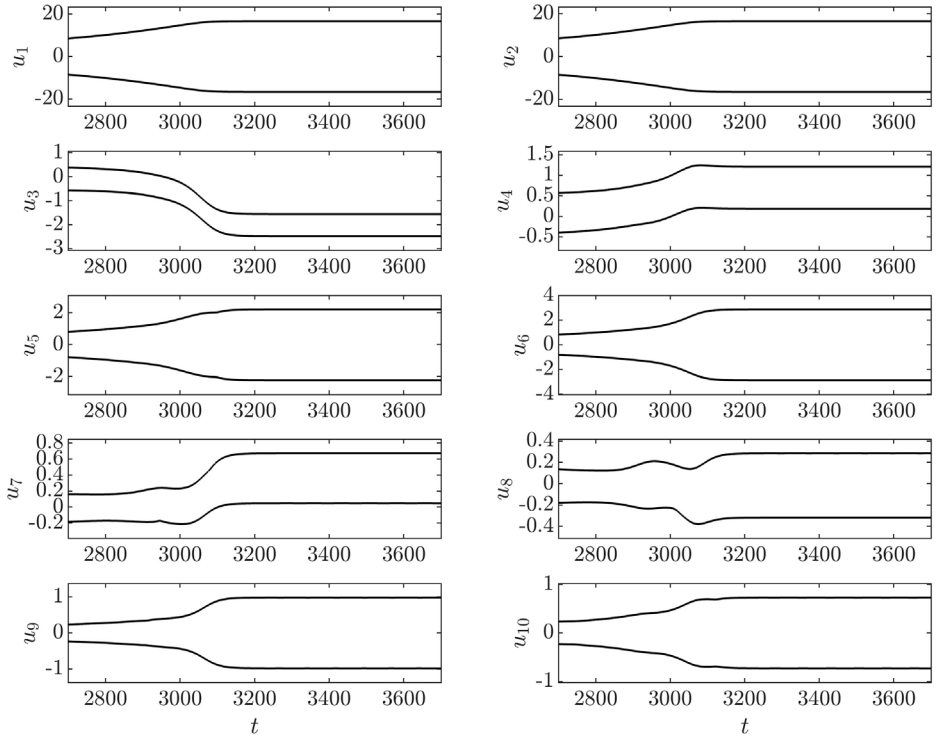


FIGURE 3
POD-ROM coefficient envelopes for flow past a cylinder (N = 10, $t \in$ [2,700, 3,700], Re = 100)

the Taylor dispersion induced by the lifting function and the transport of the mean momentum by the POD modes are accurately captured. This choice is also rationalized by the fact that the lifting function is ever-present in the solution so its convective interaction is important in accurate reproduction of the time-evolution by the ROM. Thus, we add the modes $\mathbb{P}[\boldsymbol{\zeta}_0 \cdot \nabla \boldsymbol{\zeta}_j + \boldsymbol{\zeta}_j \cdot \nabla \boldsymbol{\zeta}_0]$. Note that the two interactive terms can be combined because it is linear in each POD basis, $\boldsymbol{\zeta}_j$, meaning we only add $N + 1$ modes, which is still an $O(N)$ augmentation.

Next, we extract the diagonal entries, $\mathbb{P}[\boldsymbol{\zeta}_j \cdot \nabla \boldsymbol{\zeta}_j]$. This choice is justified by the fact that for each mode, $\boldsymbol{\zeta}_j$, the mode that is the most correlated with it is itself (i.e., when other modes might have a phase-shift, or different temporal frequencies associated with it, the auto-correlation dominates other interactions). So we consider addition of these N modes with the total additional modes being $2N + 1$.

For a thermal system with an advection-diffusion equation to describe its state, we can follow the same procedure as above for the lifting function interaction in the form of $\boldsymbol{\zeta}_0 \cdot \nabla \theta_j$; however, the auto-interaction modes are not obvious. For this work, we will choose $\boldsymbol{\zeta}_j \cdot \nabla \theta_j$, but there is no one-to-one correspondence between the dominant thermal modes and dominant velocity modes. One may come up with a more coherent substitute, but this choice remains an open question.

We note that ABM modes are not orthogonal in general, but are made orthogonal prior to running the ROM via eigendecomposition of the ROM mass matrix for numerical stability purposes. Disregarding round-off errors, the basis representation for a specific solution and test space do not affect the time-evolution of the solution for the standard POD-Galerkin ROM.

In summary, the ABM starts with N standard POD modes in Z^N and adds $2N + 1$ modes corresponding to advection by the lifting function, $\mathbb{P}[\boldsymbol{\zeta}_0 \cdot \nabla \boldsymbol{\zeta}_j + \boldsymbol{\zeta}_j \cdot \nabla \boldsymbol{\zeta}_0]$ and auto-advection, $\mathbb{P}[\boldsymbol{\zeta}_j \cdot \nabla \boldsymbol{\zeta}_j + \boldsymbol{\zeta}_j \cdot \nabla \boldsymbol{\zeta}_j]$, resulting in a total of $\hat{N} = 3N + 1$ basis functions, which are used in a standard Galerkin formulation. We will use $\hat{N}$ for the comparison against other (classic or stabilized) methods so that we have a fair cost comparison. The standard POD Galerkin ROM and ABM differ only in the choice of the underlying basis set.

4 Applications

We have demonstrated the effectiveness of the proposed augmentation method on several examples, including flow in a 2D Lid-Driven cavity (Re = 30, 000), 2D flow past baffles (Re = 800), 3D lid-driven cavity flow (Re = 10,000), flow over a hemisphere (Re = 2, 000), and turbulent pipe flow with forced

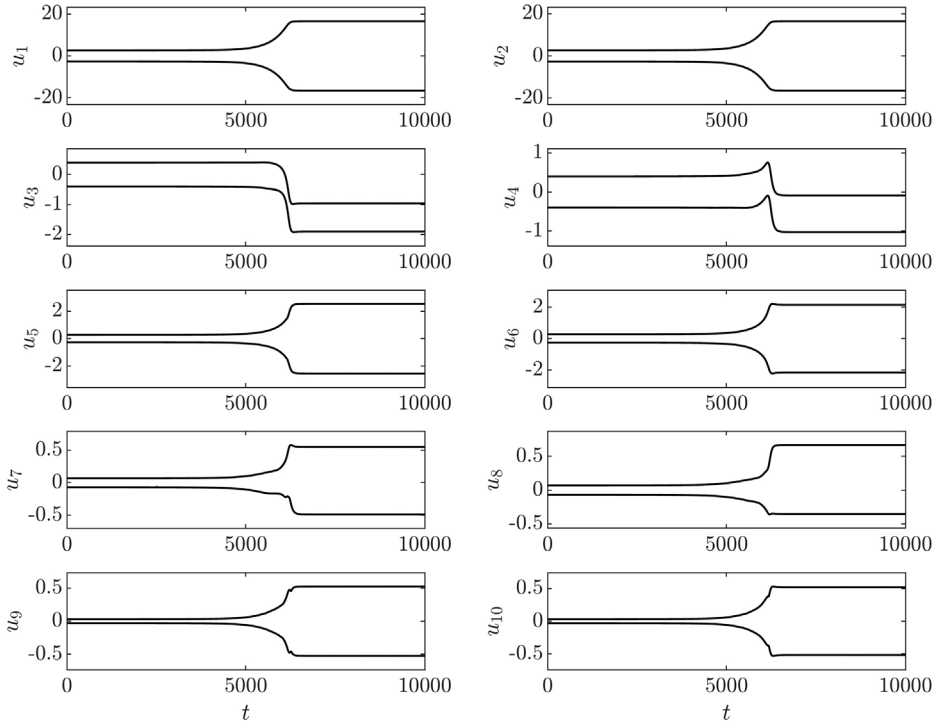


FIGURE 4
POD-ROM coefficient envelopes for flow past a cylinder (N = 20, $t \in [0, 10^4]$, Re = 100)

convection for Re = 4,000 −10,000. For brevity, we here consider only the latter two. We will also use the 2D flow past a cylinder problem to demonstrate long-time stable ROM solutions by use of ABM. In the following, we denote the FOM solution with $(\tilde{\boldsymbol{u}}, \tilde{T})$ and the various ROM solutions with $(\boldsymbol{u}, T)$. Time averages are defined as $\langle \cdot \rangle = \frac{1}{\tau}\int_{t_0+\tau}^{\tau} \cdot dt$ with integration times, τ, prescribed on a case-by-base basis. There are plots that compare different ROM strategies: Standard POD Galerkin (L^2-Glk), energy-based POD basis (H_0^1-Glk), Leray-filtered (L^2-Lry), ABM with lifting function interaction (L^2-Aug0), ABM with auto-interaction (L^2-AugD), and ABM with both interactions (L^2-AugC).

4.1 2D flow over a circular cylinder

Before we compare the stabilization properties of ABM to other methods, we first investigate the long-time stability properties of ABM on the flow past a cylinder problem. Although this is a canonical problem that is used to demonstrate the model-order reduction capabilities of the POD-Galerkin approach, there are commonly observed instability issues for a large domain. This phenomenon is documented in [19–22]. To establish that ABM addresses this long-time stability issue observed in ROMs of low Reynolds number cylinder flows, we first take 200 snapshots of a flow past a cylinder problem over 200 CTUs. The domain and boundary conditions are of that specified in [20]. Figure 2 shows a velocity magnitude plot of a snapshot used to produce the ROMs.

Figure 3 show reproduction of the 10 mode results in [20] with a difference in the ordering and signs of the POD modes stemming from the difference in snapshot count, snapshot timing, and possibly integration time for the mean flow which is used as the lifting function. For this problem, even if we increase the number of POD modes to N = 20, the growth of the instability is delayed, but is still present in the long-time solution as shown in Figure 4. Application of ABM to 10 originating POD modes resulting in a $\hat{N}$ = 21 ROM produced a long-time result (over 20,000 CTUs) that is free from the type of instability observed in the N = 20 POD-ROM. The envelopes of the coefficient trajectories of this ABM-ROM are shown in Figure 5 on top of the POD-ROM result.

With this example, we have demonstrated that ABM successfully address long-time stability issue observed in the low-dimensional models constructed by the POD-Galerkin methodology for the cylinder problem. In the next examples, we will show successful application of ABM to 3D turbulence problems.

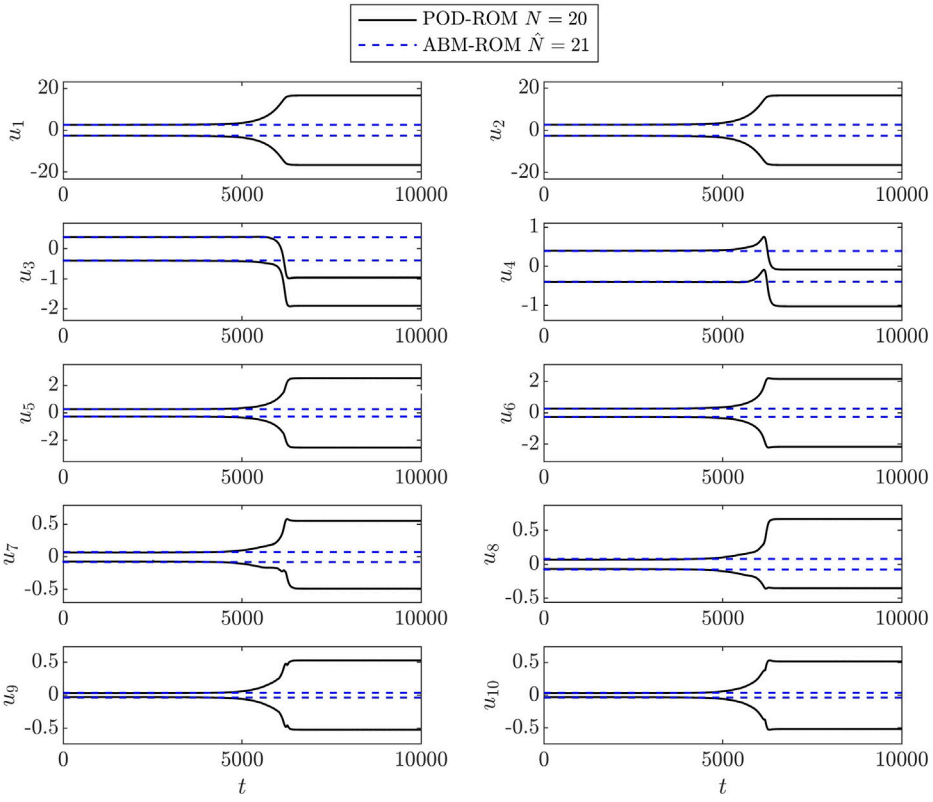


FIGURE 5
ABM-ROM coefficient envelopes for flow past a cylinder ($t \in [0, 2 \times 10^4]$, Re = 100)

4.2 3D flow over a hemisphere

Figure 6 shows a snapshot of flow past a wall-mounted hemisphere of height $D/2$ at $\mathrm{Re}_D = DU/\nu = 2000$. A Blasius profile with boundary-layer thickness $\delta_{99} = 0.6D$ is prescribed at the inlet, which $3.2D$ units upstream of the hemisphere center. Periodic boundary conditions are prescribed at $\pm\, 3.2D$ units in the spanwise direction and a stress-free condition is applied on the top surface, $3.2D$ units above the wall. Under these conditions, the flow exhibits periodic shedding of hairpin vortices, evidenced by the velocity distribution and λ_2 contours [23] in the hemisphere wake. The FOM, based on a spectral element mesh with $\mathcal{N} \approx 2$ million gridpoints was run for 100 convective time units (1 CTU = D/U) and 1,000 snapshots we collected to form the ROM POD bases.

The mean-velocity error as a function of $\hat{N}$ is shown in Figure 7 (left) for the five different ROMS. POD Galerkin with L^2 (L^2–Glk) and H^1_0 (H^1_0–Glk) Gramians, Leray-regularized Galerkin (L^2–Lry), Constrained-Galerkin (L^2–Cst), and ABM with combined lifting- and diagonal-interactions (L^2–AugC). The unstable L^2 and H^1_0 Galerkin results have several drop-outs for conditions that did not converge for this relatively high-Reynolds number application. Given enough basis functions, however, all cases converge, with the L^2–Cst being the best performer for $\hat{N} < 120$. Both L^2–Cst and L^2–AugC yield mean-field errors $< .01$ for the majority of the cases, with L^2–Cst generally being the best performer. Similar conclusions hold for the turbulence kinetic energy (TKE), the measure of kinetic energy contained in the fluctuations about the mean velocity field, shown in the right panel of 7. We reiterate that, while the constrained optimization solver performs well in the reproduction problem, it is not readily extended to pMOR because the parametric variation of the constraint limits is not known *a priori*.

4.3 Forced convection in turbulent pipe flow

The next example is that of forced convection in turbulent pipe flow with Reynolds number Re = 4,000, 5300, and 10,000 (based on pipe diameter), and Prandtl number Pr = 1. All the cases use the same spectral element distribution with differing polynomial orders. The mesh consists of 12.5 million grid points

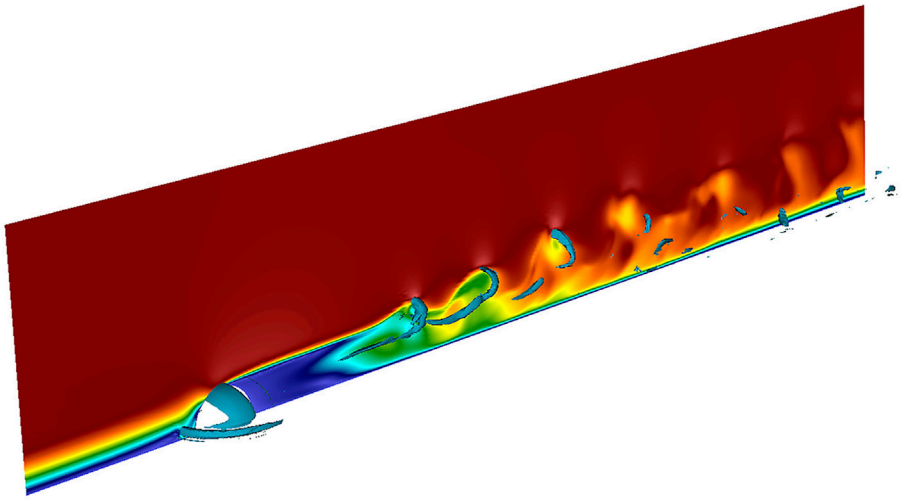


FIGURE 6
FOM velocity magnitude snapshot of flow over hemisphere (Re = 2,000) with overlaid λ_2 contour.

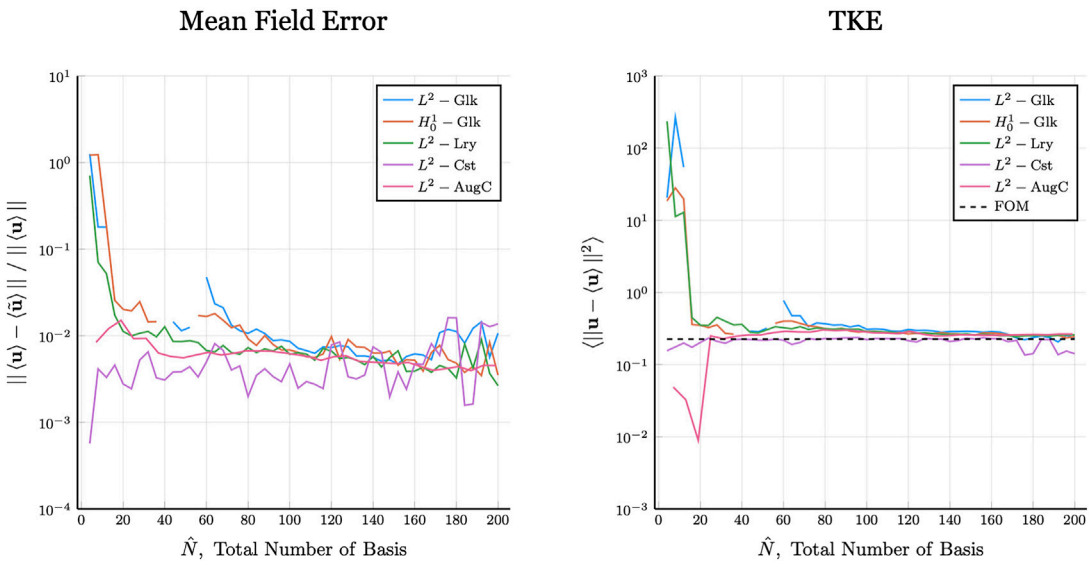


FIGURE 7
Flow over hemisphere (Re = 2,000): Mean and TKE error comparison, curve broken by blowup solution.

for Re = 4,000 and 5300, and 24.5 million points for Re = 10,000. The periodic domain length is $L = 4D$, which is generally inadequate for a full DNS of turbulence but deemed sufficient for the numerical tests in this study. For Re = 4,000, 5300, and 10,000, the respective FOM Nusselt numbers are Nu = 16.38, 21.42, and 36.14, which is in good agreement with the Dittus-Boelter relationship, $\mathrm{Nu} = 0.023\,\mathrm{Re}^{4/5}\,\mathrm{Pr}^{2/5}$. For all cases, the FOM is run until the solution is relaxed to a statistically steady state prior to gathering statistics or snapshot data. For each case, 1,000 snapshots are collected over 50 CTUs to form the Gramian, from which the POD basis is generated. Figure 8 shows typical snapshots of velocity magnitude and temperature that reveal the variation in range of scales for the different cases.

The governing equations for the FOM are the incompressible Navier–Stokes equations and the thermal advection-diffusion equation. Because of the constant-flow rate and periodic restriction

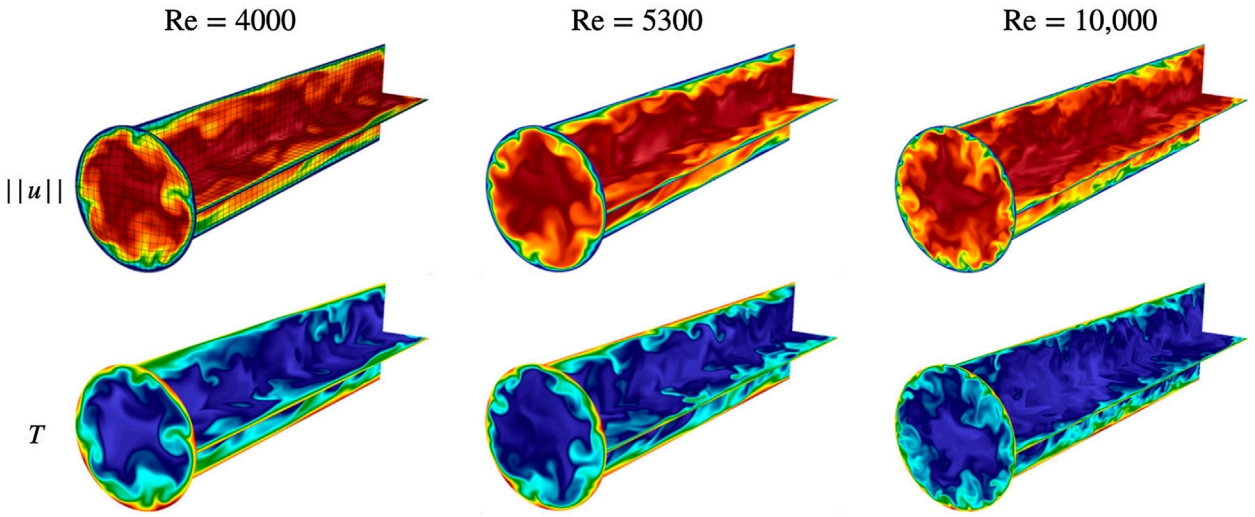


FIGURE 8
FOM velocity (top) and temperature (bottom) snapshots of 3D pipe flow at Re = 4,000, 5300, and 10,000.

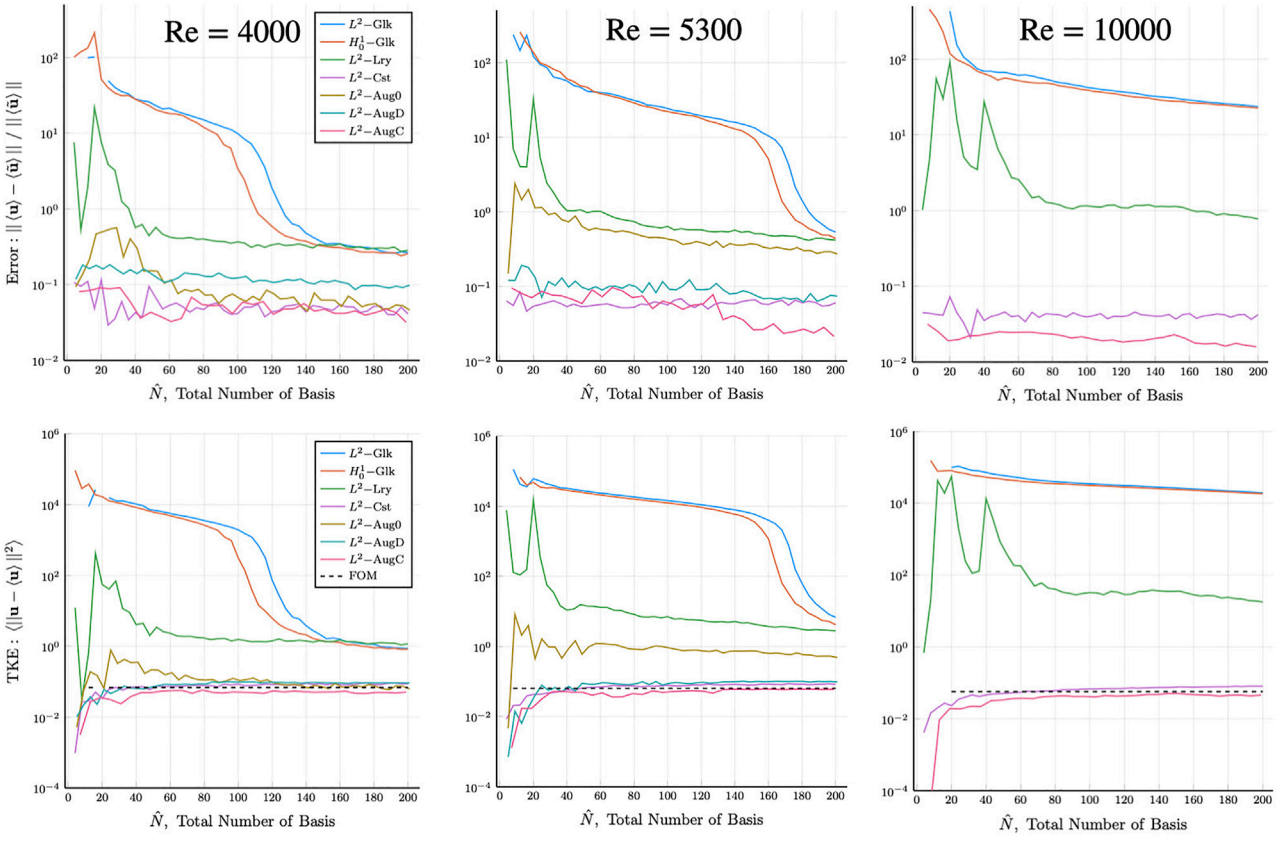


FIGURE 9
Pipe flow mean velocity error (top) and TKE results (bottom), for Re = 4,000, 5300, and 10,000 (FOM: τ = 50, ROM: τ = 500), curve broken by blowup solution.

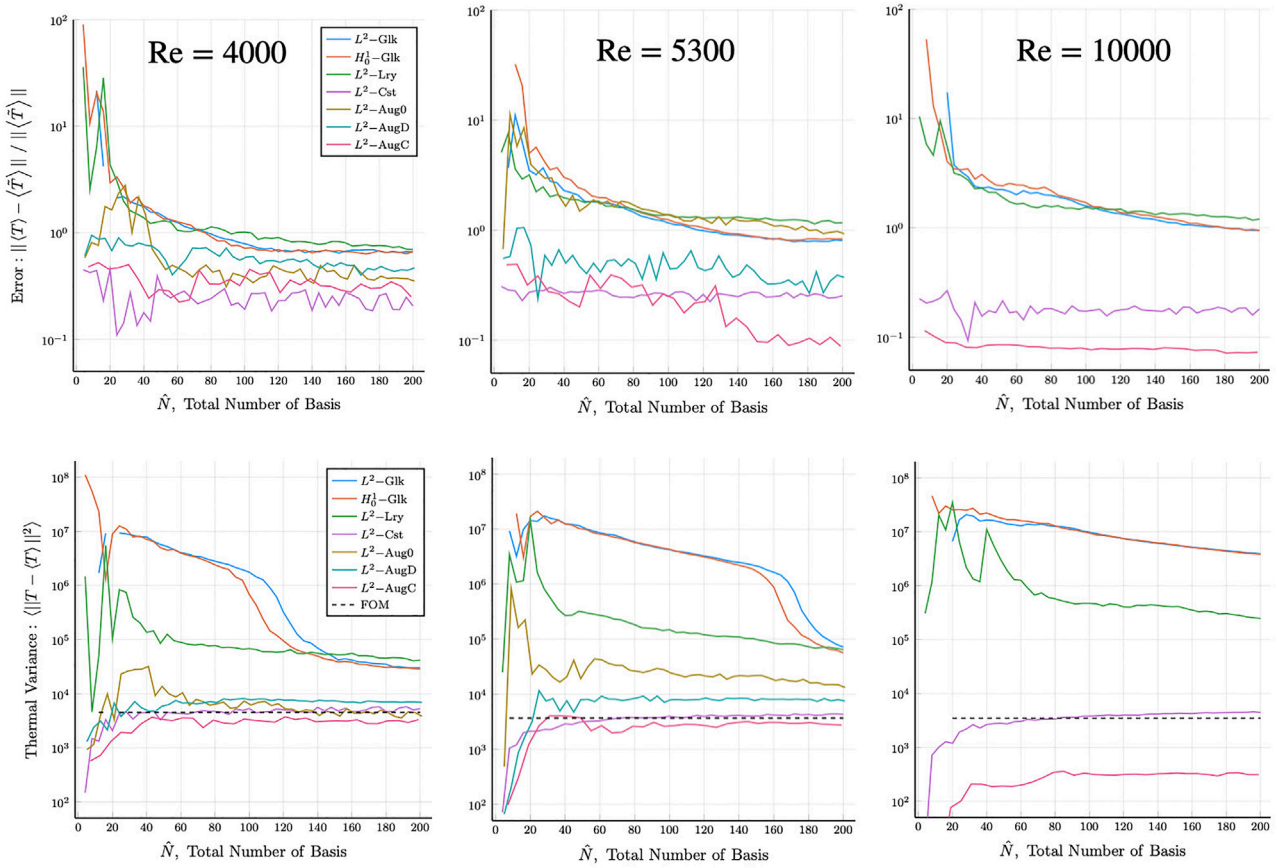


FIGURE 10
Pipe flow mean temperature error (top) and thermal variance results (bottom), for Re = 4,000, 5300, and 10,000 (FOM: τ = 50, ROM: τ = 500), curve broken by blowup solution.

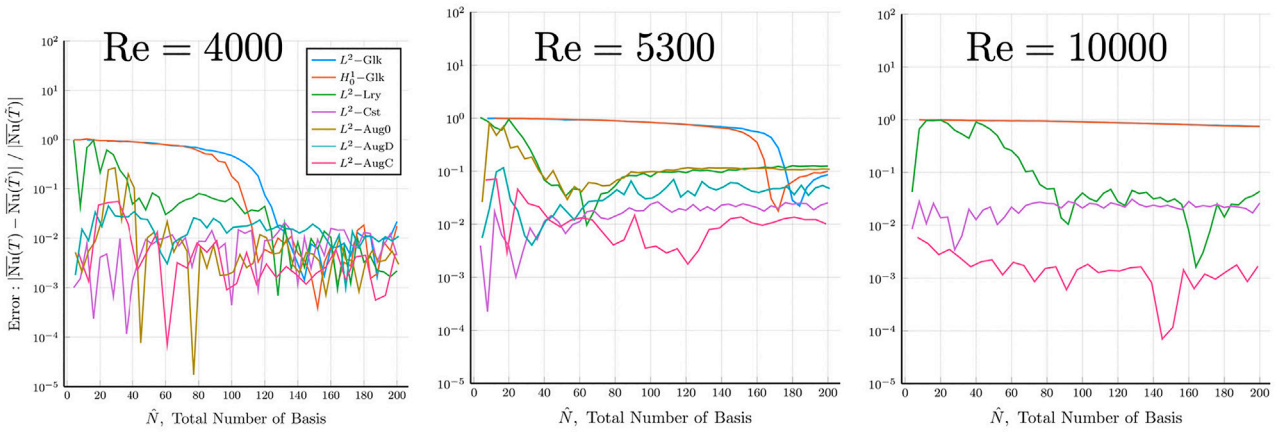


FIGURE 11
Pipe flow error in Nusselt number for Re = 4,000, 5300, and 10,000.

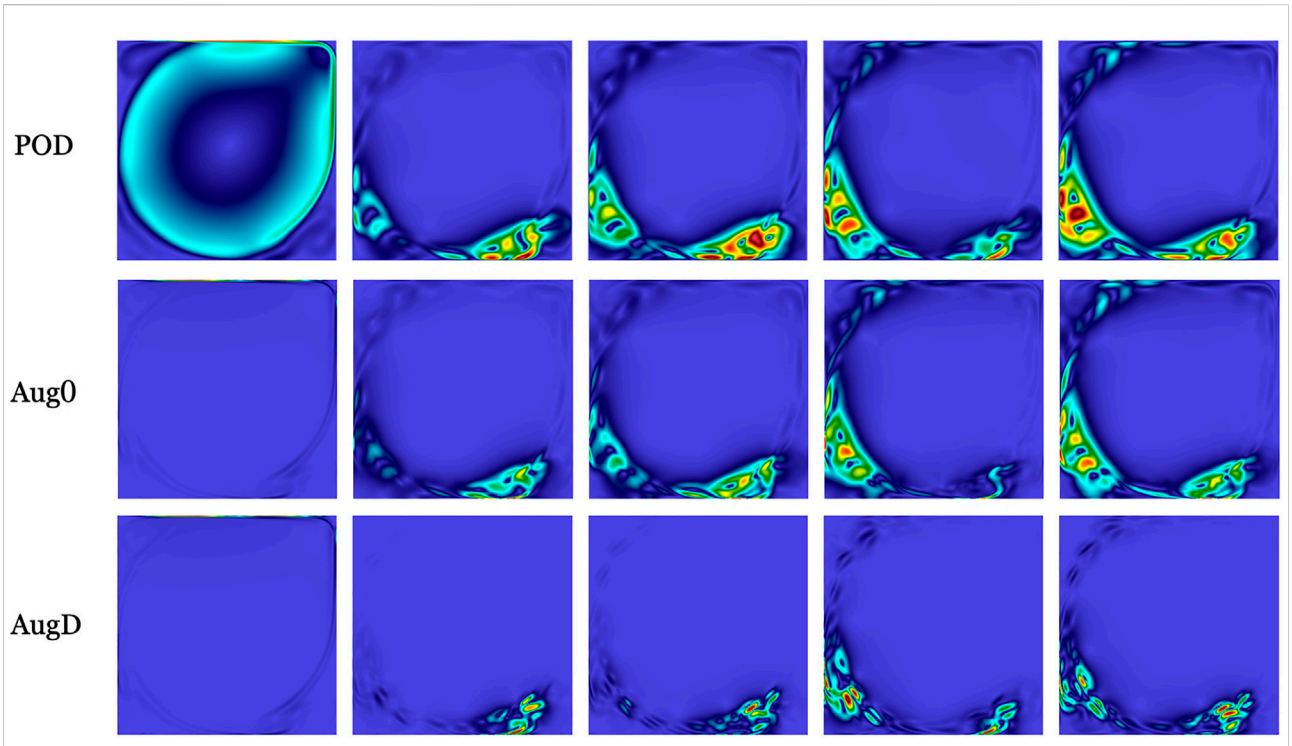


FIGURE 12
Magnitude of POD and ABM modes for a regularized lid-driven cavity problem.

on the solutions, we provide a brief discussion of modifications to the standard equations for the FOM and their effect on the ROM formulation. We start with the Navier–Stokes equations:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} = -\nabla p + \nu\nabla^2\boldsymbol{u} + \boldsymbol{f}(\boldsymbol{u}),\ \nabla\cdot\boldsymbol{u} = 0 \tag{10}$$

Here, $\boldsymbol{f}(\boldsymbol{u})$ is a uniform forcing vector field function in the streamwise direction, $\hat{\boldsymbol{z}}$, that enforces a time-constant flow-rate. In the time-discrete problem, the forcing term effectively adds an impulse-response streamwise velocity field with boundary layer thickness proportional to $\sqrt{\nu\Delta t}$. This impulse response is scaled appropriately at each time step to ensure that the mean velocity at each timestep conforms to the prescribed flow-rate. In the case of the ROM, the lifting function has the prescribed flow-rate and the remaining POD basis functions have zero flow-rate, meaning that the test-space Z^N only contains members with zero flow-rate. In the weak-form, the ROM forcing term therefore becomes

$$(\boldsymbol{v},\boldsymbol{f}) = \int_\Omega \boldsymbol{v}\cdot\boldsymbol{f}\,dV = f_z\left(\int_\Omega v_z\,dV\right) = 0. \tag{11}$$

Thus, the forcing term in the ROM formulation is zero.

The boundary conditions for the thermal problem are prescribed unit thermal flux on the walls. Therefore, we add a constant-slope ramp function γz such that the lifted temperature T, can be periodic in the domain. The equation becomes

$$\frac{\partial T}{\partial t} + \boldsymbol{u}\cdot\nabla\left(T + \gamma z\right) = \alpha\nabla^2 T \tag{12}$$

To ensure that thermal energy is conserved in the domain we set $\gamma = \frac{P}{Q} = 4$, where P is the circumference of the pipe and Q is the volumetric flow-rate.

The results for the ROMs are presented in Figures 9–11, which show the error and variance for the velocity and temperature as well as the Nusselt number behavior as a function of the total number of modes. The mean Nusselt number definition is based on the time-averaged streamwise velocity and temperature,

$$\overline{\mathrm{Nu}} = \frac{1}{\alpha\left(\overline{T_s} - \overline{T_b}\right)},\quad \overline{T_s} = \int_{\partial\Omega}\langle T\rangle dS,\quad \overline{T_b} = \frac{\int_\Omega\langle T\rangle\langle u_z\rangle dV}{\int_\Omega\langle u_z\rangle dV}. \tag{13}$$

The legends are ordered in the following manner: L^2 basis, H^1_0 basis, L^2 basis with Leray regularization, L^2 basis with constrained optimization, L^2 basis augmented with 0th-mode interaction, L^2 basis augmented with auto-correlation (diagonal), and L^2 basis augmented with combined 0th-mode and auto-correlation modes.

A common observation for Figures 9–11 is nominal convergence for the Re = 4,000 case for the L^2, H^1_0, and Leray

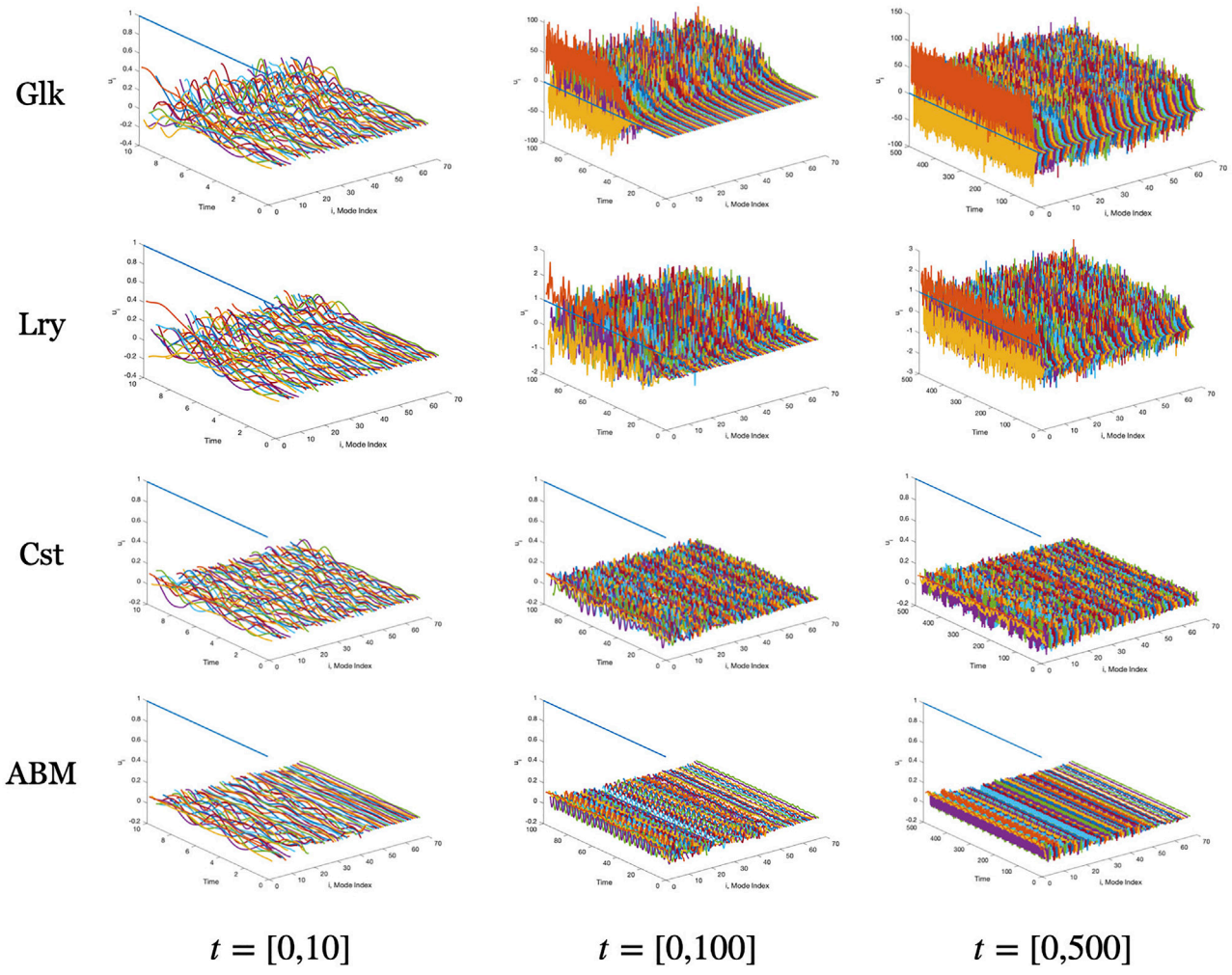


FIGURE 13
Temporal behavior of basis coefficients for standard POD (Glk), Leray-filtered (Lry), Constrained (Cst), and ABM-ROM solutions of turbulent pipe flow at Re = 5300 over (convective) time intervals $t \in [0, 10]$ [0.100], and [0.500].

regularization methods, albeit to relatively large asymptotic values. As the Reynolds number increases, more modes are required for the L^2 and H_0^1 formulations to converge, with the required number of modes apparently exceeding $\hat{N} = 200$ for Re = 10,000. Clearly, Leray outperforms standard L^2 and H_0^1, but is inferior to L^2–Cst and L^2–AugC, with the latter two having mean velocity error of just a few percent at Re = 10,000 (Figure 9, top right). The thermal behavior is similar, save that the mean-field error (Figure 10, top) is above 10% with the exception of L^2–AugC for Re = 10,000. Remarkably, this same case exhibits too little thermal variance, as seen in the lower right frame of Figure 10 (We explore this anomalous behavior in the next section.) On the other hand, the error in Nu for L^2–AugC at Re = 10,000 is uniformly less than 1% (Figure 11).

We close this section with a remark about DEIM as a possible alternative to ABM. Although DEIM allows for larger number of modes in the ROM for a given cost, its accuracy will not surpass that of the underlying ROM formulation on which it is based. So, for a classic L^2– or H^1–based formulation, DEIM will not yield an acceptable reconstruction result even at $N = 200$, whereas the constrained and ABM formulations realize convergence at much lower values of $\hat{N}$ and much lower costs.

5 Discussion

The ABM has been remarkably successful in advancing our ability to apply ROMs to high-Reynolds number flows. Several observations point to the stabilization properties of the ABM, rather than its approximation quality, as the principal driver for its success. Inspection of the modes for several cases indicate that the augmenting modes in the ABM have high wavenumber

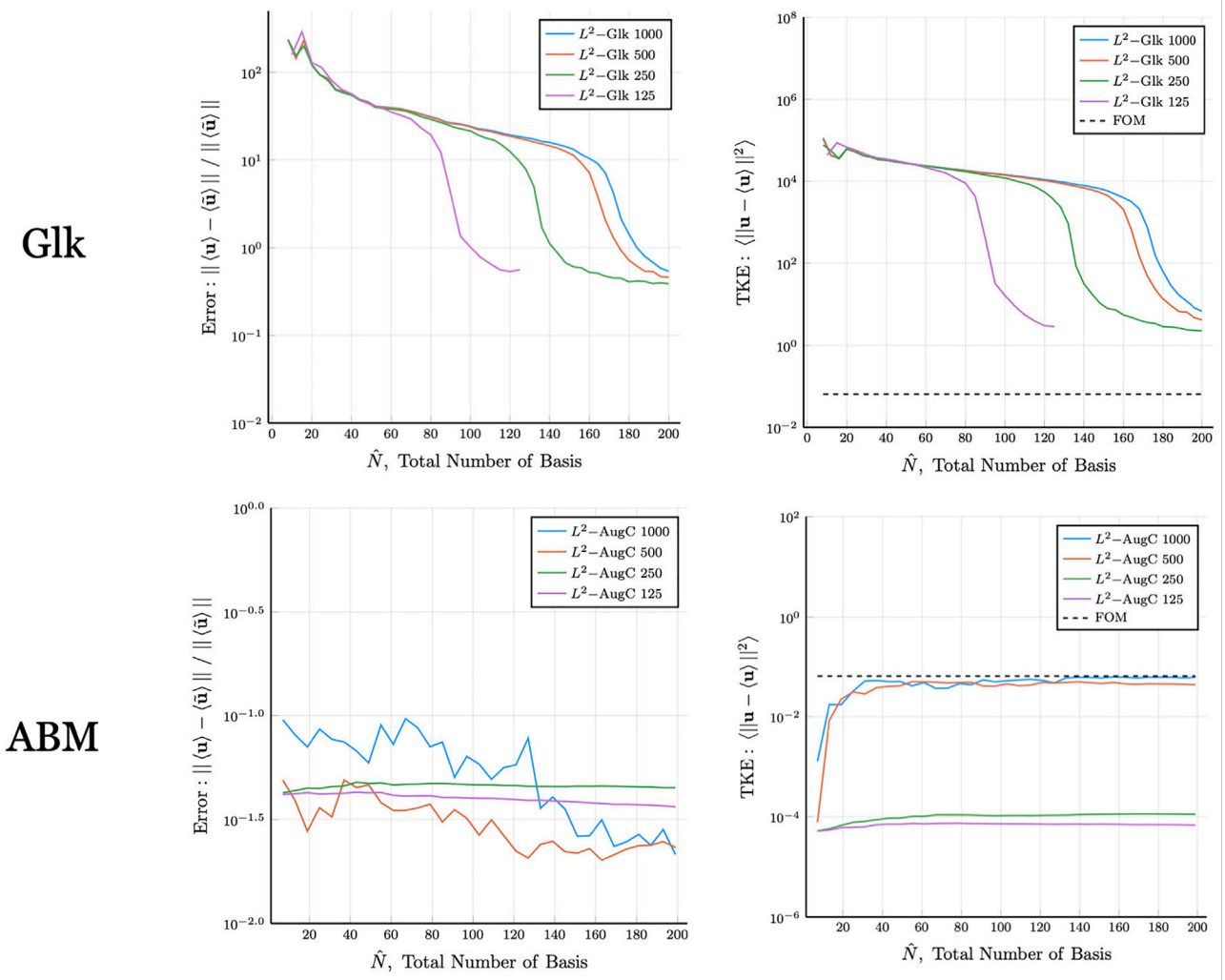


FIGURE 14
Convergence for mean velocity (left) and TKE (right) for turbulent pipe flow at Re = 5300 as a function of the number of modes for the POD-ROM (top) and ABM-ROM (bottom). The dimension of the snapshot space, K, is indicated in the legend.

content that is localized in Ω to regions of active flow dynamics. An example is illustrated in Figure 12, which shows the first 14 L^2–AugC modes for the case of a lid-driven cavity at Re = 30, 000. For $j = 0, \ldots, 4$, the first five POD modes, $\boldsymbol{\zeta}_j \in Z^N$, are in the top row; the first five 0-modes, $\mathbb{P}\{\boldsymbol{u}_0 \cdot \nabla \boldsymbol{u}_j + \boldsymbol{u}_j \cdot \nabla \boldsymbol{u}_0\}$, are in the center row; and the first five diagonal-modes, $\mathbb{P}\{\boldsymbol{u}_j \cdot \nabla \boldsymbol{u}_j\}$, are in the lower row (The 0–0 mode is of course not used twice when forming the augmented basis.) We see that the auto-interaction modes in particular feature high wavenumber content in regions of Ω where the POD modes have significant amplitude. Although it is not shown here, the augmented bases develop high wavenumber content at a much faster rate (i.e., lower mode number) than their high mode-number POD counterparts, which explains why it takes so long for the standard POD Galerkin method to stabilize in the Re = 4,000, 5300, and 10,000 pipe flow cases of the preceding section. In this sense, the augmenting modes are more wavelet-like than Fourier-like and therefore quite efficient in providing a localized dissipation mechanism for quadratic interactions. Using these bases thus makes some sacrifice on approximation properties (because we use fewer POD modes, which are optimal in generating low-rank approximations to the snapshot space in the same spirit as low-rank SVD-based matrix decompositions) in favor of better stabilization. Despite this trade-off, the ABM generally yields a much better overall approximation of the dynamics than even its stabilized POD counterparts, as is evident in the turbulent pipe flow case. While not shown in this work, ABM-ROM constructed from pipe-flow at Re = 5300 snapshots were stable even with parametric variation (in Reynolds number), but accuracy in the Nusselt number prediction decreased as the Reynolds number

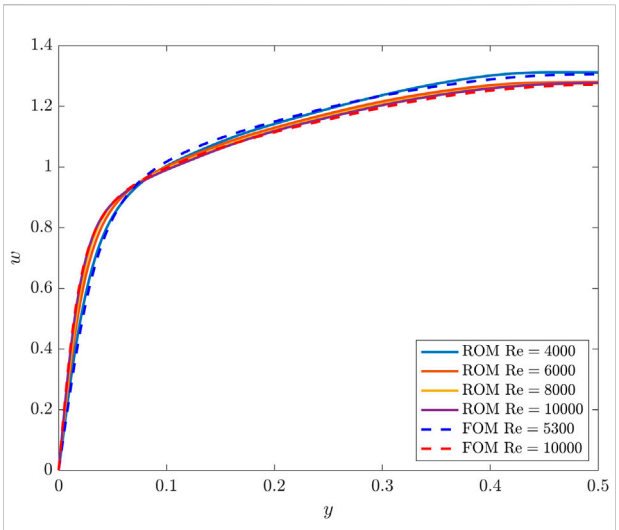


FIGURE 15
Time- and space-averaged axial velocity profiles for ABM-ROM pMOR.

moved away from 5300. How to address this phenomenon will be studied in subsequent works in the future.

This stabilization hypothesis is supported by the graphs of Figure 13, which shows the amplitudes of the basis coefficients for POD and ABM Galerkin ROM solutions to pipe flow at Re = 5300 as a function of time and mode number. The coefficient evolutions are shown over three time windows [0, 10] [0, 100], and [0, 500], which reveal the growth and saturation of the amplitudes. We see that for the ABM (in the lowest row), the coefficients are all smaller than unity, save for the ζ_0 coefficient, which is unity (All modes have unit 2-norm, so the coefficients represent the true amplitude of each scaled mode.) The ABM results also show that most of the energy is in the POD bases, corresponding roughly to the lower third of the mode indices. By contrast, the coefficients for the standard POD Galerkin modes (top row) quickly saturate to amplitudes in excess of 100, and all modes are excited. As is well known from under-resolved Navier-Stokes simulations, there is an energy pile up—manifest as high amplitude modal coefficients—when the representation lacks high wavenumber bases capable of dissipating energy. The Leray-regularized coefficients (second row) exhibit a behavior similar to the standard Galerkin approach, save that the coefficients are much more controlled, which peak amplitudes much closer to unity. The constrained approach (third row) also exhibits chaotic coefficient behavior but at much more controlled amplitudes than either the standard or Leray cases. Remarkably, the evolution on the $t = [0, 100]$ window indicates that the ABM coefficient behavior is nearly time periodic.

Another study to investigate the role of dissipation is illustrated in Figure 14. We focus initially on the upper left graph, which shows the mean-flow error for the standard POD-ROM case as a function of the number of modes $\hat{N} = N$. The modes are drawn from a set of POD bases functions based on K snapshots, where $K = 125, 250, 500$, or 1,000. Whenever $N = K$ it is clear that Z^N is equivalent to the snapshot space, which implies that the modes contain all the high frequency content present in the snapshots of a turbulent flow solution. We see that these cases have a lower error than cases where the number of modes is a relatively small fraction of the number of snapshots. The same trends are indicated in the TKE plots for the POD-ROM in the upper right graph. By contrast, the ABM-ROM needs very few total modes to yield a better estimate of the mean flow (lower left) and the best TKE predictions are obtained when the snapshot set is large (e.g., $K \geq 500$ in the lower right graph). If we have too few modes in the snapshot space, along with the nonlinear augmentation modes, the ABM-ROM appears to be overly dissipative. Therefore, we suspect that using more snapshots to produce a more accurate POD series may ensure an accurate ROM reproduction for high Reynolds number pipe flow cases.

The effectiveness of the ABM approach in a pMOR context is demonstrated by considering the pipe flow problem with two sets of snapshots at Re = 5300 and Re = 10,000. Combining 1,000 snapshots from each anchor point, we obtain 30 POD modes and 61 ABM modes using the average of the mean velocity solutions at the anchor points as the lifting function. Running this ROM for Re $\in$ {4,000, 5000, . . . , 10,000} resulted in a parameteric behavior that is consistent with the physical flow: as Reynolds number is increased, the boundary layer thickness decreases according to the law of the wall and the turbulence fluctuation adjacent to the wall region increases. Figure 15 depict the mean axial flow profile. This profile was produced by an additional spatial averaging in the axial and azimuthal directions. More detailed analysis will be conducted in the future, but this preliminary example pMOR application provides evidence that ABM-ROM is a promising approach for pMOR of turbulent flows.

6 Conclusion

We introduced a novel stabilization method, ABM, for ROM-based simulations of incompressible turbulent flows that augments the standard POD basis with approximate temporal derivatives. For a space of POD basis functions, $Z^N = \{\zeta_i\}$, $i = 0, \ldots, N$, we include and additional $2N + 1$ functions that are the Leray (divergence-free) projections of the nonlinear interactions with the lifting mode, $\{\zeta_0 \cdot\nabla\zeta_i + \zeta_i \cdot\nabla\zeta_0\}$, and nonlinear auto-interactions, $\{\zeta_i \cdot\nabla\zeta_i\}$. With these basis functions, the ROM proceeds in the standard Galerkin fashion and is seen to dramatically outperform standard L^2-and H^1_0-POD Galerkin ROM approaches as well as Leray-stabilized methods introduced by [1,11]. The ABM performs comparably to the constraint-based stabilization approach of [8], but the latter is restricted to the ROM reproduction problem (i.e., running at the

same parameter points as the originating FOM) because, in a pMOR setting, the correct basis-coefficient limits are not known at training points other than the anchor points.

We showed that the auxiliary modes of the ABM have high wavenumber content that is localized to regions in Ω where flow gradients are large and thus provide efficient dissipation mechanisms that are lacking in standard POD bases. We further demonstrated that, for standard POD methods, having a more complete POD space (i.e., incorporating $N \approx K$ modes from a relatively small snapshot space of rank K) yields lower errors than having $N' > N$ POD modes from a larger snapshot space of rank $K' > N'$. The reasoning is the same—the more complete space includes high wavenumber content in the ROM basis set that provides dissipation and hence stability. Analysis of the ROM coefficient time-traces for turbulent pipe flow at Re = 5300, illustrated that the amplitudes of all the modes for non-stabilized POD-ROM are orders of magnitude larger than their stabilized counterparts. While Leray-based stabilization mitigates this behavior, it still yields coefficient amplitudes that are roughly a factor of ten greater than observed in either the constrained or ABM-based formulations.

The ABM was also shown to be effective for predicting thermal QOIs such as Nusselt numbers. It was, however, a bit overly dissipative at Re = 10, 000. The study of the interplay between N and K indicates that this dissipation can be controlled with these two parameters and one might therefore use these parameters to gain insight to the root cause of the over-dissipation. Future work will include application of the ABM to higher Reynolds number flows and to more complex domains.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Materials, further inquiries can be directed to the corresponding author.

Author contributions

Research was performed by KK under advisement and direction of PF. The two authors both contributed to the production of the manuscript.

Funding

This work from NEUP with number DE NE0008780 for Turbulent Heat-Transfer ROM research was used. This research is supported by the DOE Office of Nuclear Energy under the Nuclear Energy University Program (Proj. No. DE_NE0008780). The research used resources at the Oak Ridge Leadership Computing Facility at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract DE-AC05-00OR22725 and at the Argonne Leadership Computing Facility, under Contract DE-AC02-06CH11357.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

1. Kaneko K, Tsai P-H, Fischer P. Towards model order reduction for fluid-thermal analysis. *Nucl Eng Des* (2020) 370:110866. doi:10.1016/j.nucengdes.2020.110866

2. Tuckerman L, Barkley D. Global bifurcation to traveling waves in axisymmetric convection. *Phys Rev Lett* (1988) 61:408–11. doi:10.1103/physrevlett.61.408

3. Barkley D, Tuckerman L. *Traveling waves in axisymmetric convection: The role of sidewall conductivity* (1989).

4. Chaturantabut S, Sorensen DC. Nonlinear model reduction via discrete empirical interpolation. *SIAM J Scientific Comput* (2010) 32:2737–64. 10.1137/090766498.(Accessed March 26, 2022)

5. Holmes P, Lumley JL, Berkooz G. *Turbulence, coherent structure, dynamical systems and symmetry*. Cambridge University Press (1996). doi:10.1017/CBO9780511622700

6. Barrault M, Maday Y, Nguyen N, Patera A. An 'empirical interpolation' method: Application to efficient reduced-basis discretization of partial differential equations. *C R Math* (2004) 339:667–72. doi:10.1016/j.crma.2004.08.006

7. Veroy K, Rovas DV, Patera A. A posteriori Error estimation for reduced-basis approximation of parametrized elliptic coercive partial differential equations: "Convex inverse" bound conditioners. *convex inverse" bound conditioners* (2002) 8:1007–28. doi:10.1051/cocv:2002041

8. Fick LH, Maday Y, Patera AT, Taddei T. A stabilized POD model for turbulent flows over a range of Reynolds numbers: Optimal parameter sampling and constrained projection. *J Comput Phys* (2018) 371:214–43. doi:10.1016/j.jcp.2018.05.027

9. Tsai P-H, Fischer P. Parametric model-order-reduction development for unsteady convection. *Front Phys* (2022).

10. Iollo A, Lanteri S, Desideri JA. Stability properties of POD-galerkin approximations for the compressible Navier-Stokes equations. *Theor Comput Fluid Dyn* (2000) 13:377–96. doi:10.1007/s001620050119

11. Wells D, Wang Z, Xie X, Iliescu T. An evolve-then-filter regularized reduced order model for convection-dominated flows. *Int J Numer Methods Fluids* (2017) 84:598–615. doi:10.1002/fld.4363

12. Balajewicz M. *A new approach to model order reduction of the Navier-Stokes equations*. Durham, NC: Duke University (2012). Ph.D. thesis.

13. Amsallem D, Farhat C. Stabilization of projection-based reduced-order models. *Int J Numer Methods Eng* (2012) 91:358–77. doi:10.1002/nme.4274

14. Cazemier W, Verstappen RWCP, Veldman AEP. Proper orthogonal decomposition and low-dimensional models for driven cavity flows. *Phys Fluids* (1998) 10:1685–99. doi:10.1063/1.869686

15. Akkari N, Mercier R, Lartigue G, Moureau V (2017). Stable POD-Galerkin Reduced Order Models for unsteady turbulent incompressible flows. In 55th AIAA Aerospace Sciences Meeting (Grapevine, United States)

16. Akkari N, Casenave F, Moureau V. Time stable reduced order modeling by an enhanced reduced order basis of the turbulent and incompressible 3d Navier–Stokes equations. *Math Comput Appl* (2019) 24:45. doi:10.3390/mca24020045

17. Fischer P. An overlapping schwarz method for spectral element solution of the incompressible Navier–Stokes equations. *J Comput Phys* (1997) 133:84–101. doi:10.1006/jcph.1997.5651

18. Sirovich L. Turbulence and the dynamics of coherent structures. I. Coherent structures. *Q Appl Math* (1987) 45:561–71. doi:10.1090/qam/910462

19. Noack BR, Afanasiev K, Morzyński M, Tadmor G, Thiele F. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *J Fluid Mech* (2003) 497:S0022112003006694–363. doi:10.1017/s0022112003006694

20. Sirisup S, Karniadakis GE. A spectral viscosity method for correcting the long-term behavior of pod models. *J Comput Phys* (2004) 194:92–116. doi:10.1016/j.jcp.2003.08.021

21. Akhtar I, Nayfeh AH, Ribbens CJ. On the stability and extension of reduced-order galerkin models in incompressible flows. *Theor Comput Fluid Dyn* (2009) 23:213–37. doi:10.1007/s00162-009-0112-y

22. Hay A, Borggaard J, Akhtar I, Pelletier D. Reduced-order models for parameter dependent geometries based on shape sensitivity analysis. *J Comput Phys* (2010) 229:1327–52. doi:10.1016/j.jcp.2009.10.033

23. Jeong J, Hussain F. On the identification of a vortex. *J Fluid Mech* (1995) 285:69–94. doi:10.1017/s0022112095000462

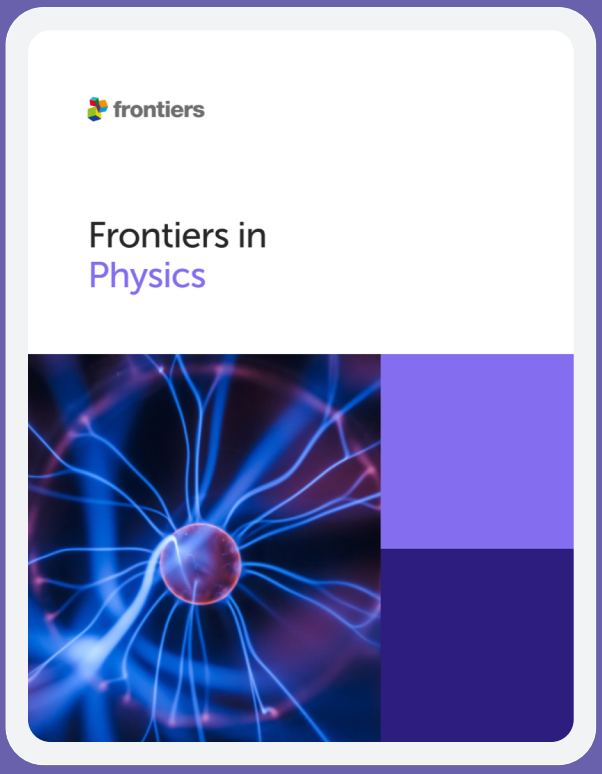
Frontiers in Physics

Investigates complex questions in physics to understand the nature of the physical world

Addresses the biggest questions in physics, from macro to micro, and from theoretical to experimental and applied physics.

Discover the latest Research Topics

See more →



Frontiers

Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

Contact us

+41 (0)21 510 17 00
frontiersin.org/about/contact