Coarse-grained models for prediction, uncertainty quantification, and extreme event statistics of turbulent flows in engineering and geophysical settings using physics-consistent data-driven closures

by

Alexis-Tzianni Charalampopoulos

Diploma of Naval Architecture & Marine Engineering, NTUA (2016)
S.M., Massachusetts Institute of Technology (2019)

Submitted to the Department of Mechanical Engineering
and MIT Institute for Data, Systems, and Society
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Mechanical Engineering and Statistics
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2023

Author . . . .
Department of Mechanical Engineering
and MIT Institute for Data, Systems, and Society
May 02, 2023

Certified by . . . .
Themistoklis P. Sapsis
Professor of Mechanical and Ocean Engineering
Thesis Supervisor

Accepted by . . . .
Nicolas Hadjiconstantinou
Chairman, Committee on Graduate Students

Coarse-grained models for prediction, uncertainty quantification, and extreme event statistics of turbulent flows in engineering and geophysical settings using physics-consistent data-driven closures

by

Alexis-Tzianni Charalampopoulos

Submitted to the Department of Mechanical Engineering
and MIT Institute for Data, Systems, and Society
on May 05, 2023, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Mechanical Engineering and Statistics

Abstract

Modeling and analysis of turbulent fluid flows remains one of the challenging areas of fluid mechanics where integration of the full equations is associated with extreme computational cost, while their simplification inevitably introduces important model errors. In this work we are aiming to answer three questions: How can we use the governing equations and available datasets to formulate physics-constrained data-driven closures that will provide accurate coarse-grained evolution equations? Can we formulate the corresponding physics-constrained closures for the quantification of uncertainty? Can high-order statistics and statistics of extreme events be computed from data-augmented coarse-scale or reduced-order models? These questions are motivated by real-world problems, such as multiphase fluid flows as well as climate modeling.

To address these questions we adopt a statistical formulation of the governing equations and employ machine-learning ideas to formulate, physics informed, non-local in space and time closure schemes for turbulent, possibly multi-phase, fluid flows found in engineering and geophysical settings. The generic form of the systems we study includes a linear operator, external forcing and a bi-linear, energy-preserving operator. We use the predictions of neural networks to integrate over a coarse grid the evolution of turbulent anisotropic fluid flows on which bubbles that act as passive inertial tracers are being transported.

To answer the first question we develop closures for the mean flow field that we complement with physical constraints, which follow from the energy-preserving character of the bi-linear operator. Next, we proceed with the second question by formulating the second-order moment equations for which we also derive data-driven closures that we complement with appropriate physical constraints. We utilize recurrent and convolutional layers to capture both temporal and spatially non-local effects. The addition of the physical constraints not only improves the performance of the resulted closures but also stabilizes the coarse-grained equations in cases that are otherwise unstable. The approach is tested both for closing the mean equation and the covariance evolution equation in a second-order statistical framework. The closure schemes for turbulent fluid flows are complemented with a method that aims to predict high-order moments of bubble cluster deformation from second-order statistics. This is achieved with the introduction of a hybrid quadrature method of moments appropriate for finite-dimensional dynamical systems. We demonstrate the resulted closures and assess the generalizability properties in different Reynolds and flow configurations.

To answer the third question we machine learn non-intrusive correction operators that use as input imperfect, i.e. coarse-scale, climate model outputs that typically have discrepancies due to low resolution. An important challenge in this case is the chaotic character of the underlying dynamics which makes the machine learning of the correction operator an intractable task. To overcome this obstacle we design a new approach based on nudging, a popular method for data-assimilation in geophysical modeling, to create consistent training datasets between the input and the output of the correction operator. We illustrate that the resulted non-intrusive correction operator is able to improve inaccuracies of the coarse-scale model attractor making it consistent with the target attractor. This allow us to obtain extreme event statistics of climate extreme events such as hurricanes and atmospheric rivers, using as input only inexpensive coarse-scale simulations. To this end, the introduced approach paves the way for the parsimonious study of climate-change scenarios and the effectiveness of possible measures or policies.

Thesis Supervisor: Themistoklis P. Sapsis
Title: Professor of Mechanical and Ocean Engineering

Acknowledgments

I express my sincere gratitude to my advisor, Prof. Themis Sapsis, for continuously supporting my work at MIT during the past six years. He taught an invaluable amount of knowledge and life lessons. I was given ample research freedom in tandem with the necessary guidance in times of difficulties. I cannot thank him enough for everything he did for me during this journey.

I would like to thank my doctoral committee members, Prof. Raffaele Ferrari and Prof. Youssef Marzouk for taking the time to participate in my research and provide valuable feedback and critiques along the way. Be it through their classes I took over the years or comments and input during discussions, their influence helped shape the trajectory of my work.

I am grateful to all my collaborators: Prof. Spencer Bryngelson, Prof. Tim Colonius, Dr. Ruby Leung, Dr. Shixuan Zhang and Dr. Bryce Harrop for the collaboration on the work presented in chapter 6 and chapter 7. Their knowledge and stimulating discussions greatly helped me during this journey.

I would also like to thank my fellow lab mates throughout the years: Dr. Saviz Mowlavi, Dr. Zong Yi Wan, Stephen Guth as well as all members of the Stochastic Analysis and Nonlinear Dynamics (SAND) lab: Bianca Champenois, Brady Hammond, Prof. Mohammad Farazmand, Dr. Antoine Blanchard, Dr. Maor Farid, Dr. Sam Rudy, Dr. Ethan Pickering, Dr. Debbie Eeltnik, Dr. Sualeh Khursid, Dr. Alireza Mojahed, Dr. Antonio Alvares and Dr. Ben Barthel, for stimulating discussions and recommendations upon the topics presented in this thesis. My warmest gratitude to Barbara Smith and Yvette Lai for helping me with expense reporting and conference-room booking. I also thank Leslie Regan, Una Sheehan and Saana McDaniels for their help with other administration-related matters and their always

prompt responses.

I would like to thank my family and friends for their support throughout the completion of my degree. Thank you for your unconditional support and love ever since the beginning.

Finally, I would also like to thank ONR for sponsoring the MURI program High-Fidelity Simulation Methodologies for Multi-Phase Flows, part of which is the research presented in this thesis. I would also like to thank DARPA for sponsoring this work via grant OHR grant HR00112290029 with the program AI-assisted Climate Tipping-point Modeling under the program manger Dr. Joshua Elliott.

Contents

List of Figures

List of Tables

Bibliographic notes

This thesis is based on a number of publications, which are listed below.

1. Charalampopoulos A., Zhang S., Bryce H., Leung R. & Sapsis T. (2023), Statistics of extreme events in climate models via coarse-scale simulations and machine learning correction operators based on nudged datasets, (submitted)

2. Charalampopoulos A. & Sapsis T. (2022), Uncertainty quantification of turbulent systems via physically consistent and data-informed reduced-order models, in Physics of Fluids 34 (7)

3. Charalampopoulos A., Bryngelson S., Colonius T. & Sapsis T. (2022), Hybrid quadrature moment method for accurate and stable representation of non-Gaussian processes and their dynamics, in Philosophical Transactions of the Royal Society A 380 (2229)

4. Charalampopoulos A. & Sapsis T. (2022), Machine learning turbulent closure models for multiphase flows, in Physical Review of Fluids 7 (2)

5. Bryngelson S., Charalampopoulos A., Sapsis T. & Colonius T. (2020), A Gaussian moment method and its augmentation via LSTM recurrent neural networks for the statistics of cavitating bubble populations, in International Journal of Multiphase Flows 137, 103262

Chapter 1

Introduction

The earliest version of a neural network (NN) dates back to 1958 and the groundbreaking work of psychologist Frank Rosenblatt [139]. In his work, Rosenblatt drew from the concept of Hebbian learning [61] and successfully created a neural network with image recognition abilities, Perceptron. A year later, Arthur Samuel coins the term machine learning (ML), motivated by a program of IBM that was trained to play checkers [4]. This family of computational models drew inspiration from psychology and brain science, which motivated the employed anthropomorphic naming conventions. It is hard to imagine that the pioneers of that era could have anticipated the drastic impact of their research in the upcoming decades.

In particular, the introduction of the back-propagating algorithm, [141, 143] allowed these models to fully utilize the contemporary advances in high-performance computing and storage hardware, thus leading ML and NN to surpass the then state-of-the-art results for a number of problems around computer vision and language processing. These successes, together with the easy applicability of these models to different problems, garnered increasing attention from computer science researchers, resulting in the continuing development of ever-more potent algorithms that keep on shattering the performance records on standardized tests.

In parallel, ML drew the attention of scientists working on complex dynamical systems, particularly, the prediction and analysis of their temporal evolution. This connection should arise naturally, as since the 1940's Kolmogorov coined turbulence as a possible problem for statistical learning theory. This approach provides an alternative to the established systematic study of dynamical systems via derivation of partial differential equations (PDEs) that describe the physics and underlying dynamics. Under an ML-based framework, data are no longer used only for the verification of theoretical models, but instead allow scientists to make complicated decisions directly using information extracted from data. While such an approach is still contentious among the scientific community, its ability to bypass possible internal system complexities for which a first principle derivation is cumbersome, eventually highlighted ML as a complementary alternative (e.g., [161, 92, 25]).

Yet, only after the success of ML in image recognition will this potential link be further explored. Eventually, in the first application of deep neural networks (DNNs) to turbulent flows, Milano & Koumoutsakos [114] showcased their ability to predict turbulence within boundary layers. With the introduction of mini-batch optimization algorithms [85, 167, 78], scientists were able to develop models that seamlessly handle the rapid accumulation of data from sources like experiments, measurements and high-fidelity simulations for incorporating ML in turbulent systems.

Despite the promising results of ML, it cannot be treated as a panacea for dynamical modelling. While a lot of ML tools are treated as plug-and-play in much of the image processing community, this is not the case for dynamical systems. This intuitive comment was theoretically solidified by David Wolpert who, with his *no free lunch theorem* [183], proved that without prior information for the target distribution, there is no single learning algorithm that is best suited for all possible data sets. In addition, by their very design, ML algorithms are constructed as black boxes that provide descriptions. Hence, a purely ML-based modeling, while possessing a a lot of vastly expressive tools, severely suffers in explaining observed phenomena, especially in the immense model spaces of turbulent systems. These two drawbacks naturally restrict the probability of any purely ML-driven model to extrapolate and generalize to other problems with even the slightest of differences.

Hence, a successful application of ML to dynamical systems requires a problem-dependent approach that leverages valuable knowledge about the system of interest. This knowledge ranges from using the underpinning physical laws that govern the system, appropriate physical constraints that should be respected by the system, to the employment of particular NN architectures, like the convolution [84] and recurrence [142], to correctly fit the properties of the system. These practices increase accuracy and convergence speed of the model, but above all, improve its generalizability. This family of ML approaches will be henceforth referred to as physics-constrained machine learning (PCML).

A review of PCML methods focused on turbulent closures is presented in chapter 2. Each of these PCML methods provide their unique values and can be effective under the right framework. At the same time, they suffer from particular weaknesses, from relying on near-perfect physical models, to the formulation of models that do not scale well, to long-time stability issues. This thesis hopes to provide some insights into PCML turbulent closure schemes and the prediction of the statistics of turbulent models in general. In the following chapters, we present directions on how data-driven uncertainty quantification can be achieved in realistic scenarios.

1.1 Contributions

This thesis explores machine learning strategies designed to incorporate physical knowlodge in nonlinear dynamical systems, with particular focus on turbulence. Throughout this work we will continue along with the blended, physics-constrained ML perspective that aims to extract information from both data observations and imperfect physical reduced-order models (ROMs). To cope with the intrinsic limitations of working with reduced-order spaces and coarse-scale resolutions, we employ ML models that implicitly augment the partial state representation with time history - an idea motivated by the well-celebrated embedding theorems of Whitney [182] and Takens [168]. We present our studies on important engineering application areas like anisotropic turbulent jets, climate statistics, as well as bubble dynamics.

1.1.1 Turbulent closure schemes for statistics of anisotropic flows

A main goal of this work is to formulate energy-preserving and spatio-temporally non-local turbulent closures which are *a priori* consistent with the conservation properties of quadratic nonlinearities. This property is of particular importance in fluid mechanics through the presence of the advection term. These constraints follow from the energy-conserving properties of the nonlinear advection operator in Navier-Stokes and have been utilized previously in the context of uncertainty quantification and stochastic closure models [147, 150, 102].

We first employ this method to predict the coarse-scale mean state of the system. Specifically, we utilize machine learning schemes which represent the effect of the small scales at each spatial location, using as input the large scale features of the flow in a spatial neighborhood of this location. Past values of the large scale features are also employed as inputs for the turbulent closures in a causal manner. These data-driven schemes are enforced to be consistent with physical constraints expressing the energy exchanges between resolved and unresolved scales. In contrast to previous efforts where the full system equation is used as a constraint, assuming perfect knowledge of the equation form and/or parameters (e.g. [137]), the formulated constraint in this work expresses a universal property of the advection terms, i.e. that they do no create or destroy kinetic energy of the flow.

In a similar fashion, we seek to expand the aforementioned approach to allow for prediction of second-order statistics of turbulent flows as well. Similar to the approach of traditional modified quasilinear Gaussian closures (MQG), a second-order statistical framework is employed for this study. This framework allows for a computationally cheap reduced-order model. Uncertainty is modelled under a parametric Bayesian point of view. In previous MQG approaches, the nonlinear energy transfers, that result in non-Gaussian effects, are modelled via a quasi-linear approach, where the closure is tuned to specific steady-state statistics. The current method replaces this assumption with spatio-temporally nonlocal neural networks. The scheme once again respects the energy transfers dictated by the nonlinear term. This is achieved by using steady-state statistics of flow realizations that are incorporated during the training process. These physical constraints allow for correct prediction of mode-to-mode energy transfers during simulations. The utilization of deep neural networks allow for a richer representation of nonlinear energy transfers and higher accuracy, not only for the mean but also for the second-order statistics of the flow. Past values of these features are also employed as inputs for the turbulent closures in a causal manner.

1.1.2 Non-intrusive prediction of climate statistics via transfer learning

Apart from data-informed closure terms we explore the ability of non-intrusive neural networks to correct the statistics of coarse-scale turbulent systems. This thesis presents a systematic framework for improving the predictions of statistical quantities for turbulent systems, with a focus on correcting climate simulations obtained by coarse-scale models. Specifically, failure to incorporate all relevant scales in climate simulations leads to discrepancies in the energy spectrum as well as higher order statistics. Accurate but computationally cheap coarse-scale models are essential in climate simulations, where high-fidelity simulations are prohibitively expensive to run for multiple scenarios or long simulation time. Especially for climate change studies, multiple long-time scenarios need to be simulated, thus making accurate coarse-scale reduced-order models necessary for such applications.

While high resolution simulations or reanalysis data are available, at least for short periods, they cannot be directly used as training datasets to machine learn a correction for the coarse-scale climate model outputs, since chaotic divergence, inherent in the climate dynamics, makes datasets from different resolutions incompatible. To overcome this fundamental limitation we employ coarse-resolution model (here we employ Energy Exascale Earth System Model, E3SM) simulations nudged towards high quality climate realizations, here in the form of ERA5 reanalysis data. The nudging term is sufficiently small to not “pollute” the coarse-scale dynamics over short time scales, but also sufficiently large to keep the coarse-scale simulations “close” to the ERA5 trajectory over larger time scales. The result is a “compatible" pair of the ERA5 trajectory (used as output training data) and the weakly nudged coarse-resolution E3SM output that is used as input training data to machine learn a correction operator. We emphasize that the nudging step is used only for the training phase. Once training is complete, we perform free-running coarse-scale E3SM simulations without nudging and use those as input to the machine-learned correction operator to obtain high-quality (corrected) outputs. The model is applied to atmospheric climate data with the purpose of predicting global and local statistics of various quantities of a time-period of a decade. Using ERA5 datasets that are not employed for training, we demonstrate that the produced datasets from the ML-corrected coarse E3SM model have statistical properties that closely resemble the reanalysis data.

1.1.3 Closure schemes for high-order moments of bubble clusters

The final contribution of this thesis revolves around predicting high-order moments of nonlinear dynamical systems via the values of lower-order moments. In particular, we focus on quadrature-based moment methods (QBMMs) which are used to invert these moments with a quadrature rule, approximating the required statistics for bubble clusters. QBMMs have been shown to accurately model sprays and soot with a relatively compact set of moments. However, significantly non-Gaussian processes such as bubble dynamics lead to numerical instabilities when extending their moment sets accordingly. To circumvent these stability issues, this work employs neural networks to enhance the predictive abilities of standard 2-by-2-node (4-node) CHyQMOM, which only requires access to first and second-order moments. This approach avoids both the numerical instabilities and high computational costs of evolving higher-order moments. The method follows the recent success of deep neural networks for improving multiphase flow models [97, 98, 29, 21]. We expand on a previous such effort that used neural networks to close strictly-Gaussian moment transport equations [21]. Here, we instead seek data-informed corrections to a CHyQMOM method [40, 127]. By doing this, one has control over the resulting quadrature nodes and weights. This makes correcting moment approximations straightforward and consolidates the two neural networks of [21] to one. This allows for computation of even out-of-training-set moments, in contrast to data-informed moment methods that use low-order moments to learn specific high-order ones [67, 69, 68]. Extension from [21] includes non-uniform and long-time pressure forcings, making the trained model appropriate for computational fluid dynamics (CFD) solvers.

1.2 Thesis organization

An overview of the thesis organization follows. In chapter 2 we provide an overview of existing ML methods for nonlinear dynamical systems. In chapter 3, descriptions and explanations for universal tools used by such methods are included. Some fundamental knowledge regarding turbulent systems is also provided. In chapter 4, we present an energy-preserving spatio-temporally non-local turbulent closure scheme. This scheme is a priori consistent with the conservation properties of the advection term in Navier-Stokes equations and is tested on anisotropic turbulent flows. Along the same line, chapter 5 expands upon the previous work and describes a data-informed closure scheme for second-order statistics of dynamical systems with an energy-preserivng quadratic nonlinearity. This model is applied on toy models and more realistic turbulent flows as well. In chapter 6, a non-intrusive transfer learning approach for learning the statistics of climate models is described. In chapter 7, a closure scheme for high-order moment prediction of bubble cluster deformation is presented. Conclusory remarks are provided in chapter 8, followed by a short discussion on possible future work.

Chapter 2

Background

Abstract

In this chapter, we present the relevant background for the methodologies described thereafter. Short expositions of relevant literature regarding the machine learning methods presented in this thesis are provided. First, literature regarding data-informed turbulent closure schemes is presented. Then, an exposition of uncertainty quantification schemes for turbulent systems follows. Finally, quadrature methods of moments techniques are presented.

2.1 Review of related methodologies

2.1.1 Data-informed turbulent closure schemes

Turbulent fluid flows in nature and engineering are characterized by a wide range of spatial and temporal scales with nonlinear interactions making their reduced order modeling a challenging task. Over the last decades several ideas have emerged that successfully model turbulent fluid flows, such as Large Eddy Simulations [116, 144]. However, these methods still require very high resolution in order to satisfactorily model the large scale dynamics, as well as features associated with those. This is an important computational obstacle especially for applications involving uncertainty quantification, optimization, and risk analysis where there is a need for a large number of accurate simulations.

The machine-learning advances of the 1960's sparked an interest in utilizing neural networks to develop reduced order models for turbulent flows. The machine-learning closures abandon the path of a closed-form expression for the closure terms into utilizing experimental or costly high-fidelity computations to train a neural network and predict the nonlinear energy transfers between resolved and unresolved scales. However, despite the association of turbulence and statistical learning since the 1940's, no concrete results exist from that period regarding turbulence. The lack of results will lead to the 1973 Lighthill report and its subsequent debate (which is free on youtube for everyone to watch). In this debate Lighthill attacks the ML community for making high promises it has failed to accomplish. Lighthill strongly believed that due to combinatorial increase in the complexity of the state space of physical problems (as we introduce more degrees of freedom) ML will never be able to be used outside of toy examples. This will lead to the so called AI winter of the 70's and early 80's.

With the re-introduction of ML via DNNs, interest from the community of dynamical systems arose again. For the first time since the Lighthill report, there was hope that the new ML models were expressive enough for the real-world high-dimensional application. The first application of deep neural networks to turbulent flows appears to be the work of Milano & Koumoutsakos [114]. They utilized nonlinear autoencoders to develop a nonlinear embedding for a turbulent flow near a wall. Their work showcased that they were able to learn the flow just by using information on the wall only. Their approach can be thought of as a nonlinear version of the widely used in turbulence proper orthogonal decomposition (POD) [94],[95], [157]. Their use of neural networks for Principal Components Analysis (PCA) [8] improved the predictions possible by traditional POD.

Since then there has been a plethora of efforts focusing on machine-learning closures using different data-driven schemes, such as artificial neural networks in fluid flows [41, 156, 109, 186] and multiphase flows [97, 98], random forest regressions [184], spatially nonlocal schemes such as convolutional neural networks [194], stochastic data-driven representations using generative adversarial networks [163], reinforcement learning [120] and with applications ranging from engineering turbulence to geophysics and beyond (see [20] for a recent review).

One of the great advantages of machine-learning closures is their capability to seamlessly model non-locality in time. In this work, a non-local but causal modeling of the closure terms, implies that for the prediction of the closure terms at a particular time, only present and past information is used. Indeed, there is no a priori reason to expect that the closure terms of a complex system will behave in a Markovian manner, i.e. depend only on the current reduced-order state of the system. On the contrary, Takens embedding theorem [168] states that if we observe only a limited number of the state variables of a system, in principle, we can still obtain the attractor of the full system by using delay embedding of the observed state variables. Therefore, it is essential to incorporate memory effects when we model closure terms for turbulent fluid flows. This approach has found success in a number of physical applications involving bubble motion and multiphase flows [180, 21], as well as the reduced-order modeling of chaotic dynamical systems [178, 180, 179].

On the other hand, machine-learning schemes allows us to parametrize the closure terms using a large number of input variables opening the possibility for non-local models in space (see [194] for an application to the advection of a passive scalar).

Spatially, non-local models have been advocated for turbulent closures and there is a plethora of related ideas ranging from scale-dependent closures [133], non-local Reynolds Stress models [56], and fractional-operators closures [146]. Several ideas related to functional neural networks or operator neural networks have shown great promise in this direction [32, 33], and have recently been of great interest in the context of turbulent closure models [193, 3].

Beyond local closures deep neural networks have also been used successfully in combination with the underlying governing equations for reconstructing complex fluid flows and identifying flow parameters. Specifically, physics-informed neural networks [135] identify the optimal solution (either the flow itself or parameters associated with it) by minimizing an objective function that contains the Navier-Stokes equations, as well as scattered data in space and time. Inclusion of the governing equations significantly improves the behavior of the data-driven scheme, while the representation of the solution in terms of a neural networks circumvents the need for a grid or spatial discretization scheme. The method has shown great promise for reconstructing fluid flows given spatio-temporal measurements [137], as well as recovering macroscopic quantities such as lift or drag for vortex-induced vibration problems [136]. Previous efforts along this line include the embedding of symmetries such as Galilean invariance to the neural net predictions for an anisotropic Reynolds stress tensor [90, 89].

2.1.2 Uncertainty Quantification of turbulent systems

The defining property of turbulent dynamical systems is the existence of multiple and persistent sources of instability. While from a physical perspective such systems are usually viewed as deterministic in nature, uncertainty manifests into their study via an abundance of mechanisms. Fundamental model assumptions like model structure, constitutive laws, geometry, as well as initial and boundary conditions, may be only approximations of the truth, and as a result carry uncertainty into the implementation. Furthermore, spatio-temporal discretization errors push the model predictions away from the underlying mathematical model, allowing only for bounds between the numerical and analytical solutions. Additionally, other input parameters of the model may exhibit randomness. This effect can be either an intrinsic property of that quantity (i.e. aleatoric uncertainty) or stem from an inability to accurately measure its otherwise deterministic value, due to computational and experimental limitations. Hence, all these effects need to be accounted for in the study of unstable complex systems.

Despite the many sources of randomness, uncertainty quantification (UQ) attempts can be separated into the following categories: (i) propagation of stochasticity from input parameters, via the model, to the output values, (ii) parametrization of input-parameter uncertainty via backwards-propagation of the model output uncertainty. Both kinds of UQ studies have been applied to all fields where nonlinear multiscale systems are observed. UQ is used for performance prediction of integrated circuits with thousands of sub-micrometer parts [73]. It is accounted for in assessing the structural integrity of large structures with variability in their material properties [45, 46] and in imperfect neuroscopic models in material science [74, 31]. It is used in combustion to describe complex kinetic mechanisms [129]. Furthermore, in the last 50 years, UQ tools have been widely adopted by researchers interested in turbulent flows. They have been used to model permeability of porous media in multiphase flows [44, 43]. UQ is used to account for randomness in microfluidic applications [79] and thermal problems [63, 185]. It is used in engineering applications with shear turbulence [170]. In addition, it has become a major research interest in climate studies [176, 145, 101] and climate change in particular [103].

The most straightforward UQ approach is the well-known Monte Carlo (MC) method [113, 58, 57]. Techniques inspired by the MC method, perform numerous deterministic simulations of the system for sampled conditions, and proceed to perform a-posteriori statistical analysis on the numerical results [47]. Yet, repeated simulations of turbulent models are still prohibitively expensive both to generate and store [13]. Hence, MC techniques find limited applications usually only on low-dimensional systems. As a result, the need for efficient uncertainty propagation schemes arises for more complex and computationally expensive systems.

To reduce the computational cost of UQ, many approaches project the initial system on a low-dimensional subspace of pre-selected modes [104]. The first such approaches derived reduced-order models based on an energy-based POD [158, 159, 160, 65]. A similar concept is used in deriving reduced-order models via balanced POD based on linear theory [82, 99]. Orthogonal decompositions are also used to derive dynamically orthogonal field equations [148, 147]. Finite-series representations of randomness are also used in truncated polynomial chaos (PC) expansions [119, 66, 79, 59]. While all these models have found success in weakly chaotic regimes, they suffer from the fact that in turbulent systems, non-energetic modes can intermittently have significant impact on energetic modes [35, 17, 150, 149]. As a result, despite the low computational cost of truncated-series methods, such approaches are antithetical to the nature of turbulence. Non-modal UQ approaches, include the utilization of the fluctuation-dissipation theorem (FDT) [51, 1, 52, 2, 53, 103, 77] and the modified quasilinear Gaussian closure (MQG) scheme [147], which utilizes a second-order moment framework and models the impact of non-Gaussian statistics via incorporating statistical steady-state information appropriately.

It is therefore clear that for reduced-order models to accurately predict that statistical properties of the reference turbulent systems, a physically consistent modeling of the intermittent energy transfers between energetic modes needs to be employed. Hence, tools that allow for the approximation of complicated operators are of interest. Apart from recent conventional basis expansions [190, 72, 151] and parametrizations [104, 105], deep neural networks have recently seen success in reduced-order modelling of turbulent systems both in a deterministic and in a statistical framework [41, 156, 109, 108, 186, 97, 98, 29, 28]. Despite their lack of closed-form expressions for the closure terms, neural networks have reliably approximated many intricate operators in nonlinear dynamics. In addition, neural networks can seamlessly incorporate spatio-temporal non-locality in their predictions, a property that suits many reduced-order models. Utilizing temporal delays in a closure scheme is theoretically justified via Takens' embedding theorem [168], which states that under the constraint of observing a limited number of the state variables of a system, in principle, we can still obtain the attractor of the full system by using delay embedding of the observed state variables (i.e. a non-Markovian approach). Such methods have seen success in reduced-order modelling of highly non-linear dynamical systems [178, 180, 179, 21, 28, 29].

2.1.3 Quadrature Methods of Moments

The dynamics of dispersions of small particles or bubbles in a fluid are important to many engineering and medical applications. In medicine, ultrasounds, generated via small cavitating bubbles, are employed during cataract removal [76], to stop internal bleeding [175, 174], and in other procedures like tumor necrosis [7]. Focused shockwaves can cavitate bubbles that ablate kidney stones during lithotripsy treatment [34, 132]. Their interaction with biological tissue or manufactured soft materials also attracts the medical [19, 124, 121, 38, 162] and material science communities [42, 10, 171]. Bubble cavitation is also responsible for damage and noise in hydraulic pipe systems [181, 165], hydro turbines [39, 81, 96], and propellers [153, 71]. At the same time, soots are critical to combustion [75, 9, 128, 118] and aerosols are used in many industrial processes [155, 83, 70]. In nature, cavitation is used as part of the hunting strategies of some marine animals, including humpback whales [87, 86, 22], mantis shrimps [126], and snapping shrimps [11, 80].

While the dynamics of these particles can be simulated directly for a specific (sampled) dispersion by tracking each particle, distribution statistics are typically sought in applications. In flows with large spatial gradients, a large ensemble of such simulations (Monte Carlo) is required to gather these statistics [192, 140]. The poor scaling of Monte Carlo (MC) makes such simulations expensive, and particle tracking also interferes with efficient parallelization. By instead phase-averaging the equations of motion [188], a two-way coupled set of Eulerian equations that are more suitable to parallelization and GPU processing is obtained. However, the averaged equations involve solving the generalized population balance equation (PBE) [138]. The PBE evolves the dispersed phase number density function (NDF) as a function of its dynamic variables [23]. For example, the relevant variables for bubbles dynamics are the bubble radii and their radial velocities. However, further treatment is still required. The PBE is a partial differential equation in the dynamic variables, separate from the spatial and temporal variables of the flow equations, making this approach intractable for large simulations.

Quadrature-based moment methods (QBMMs) are a low-cost approach to approximately solving a PBE. Introduced in [111], QBMMs have seen rapid improvement [107]. In brief, they prescribe a finite moment set and invert it to an optimal set of quadrature nodes and weights in the dynamic system phase space. The success of QBMMs has led to the creation of open-source libraries for them [23, 125]. In the case of multiple dynamic variables, conditional QBMMs like conditional quadrature method of moments (CQMOM) [187] and conditional hyperbolic-MOM (CHyQ-MOM) [40, 127] are preferred. These methods can efficiently solve many problems but suffer from a combinatorial explosion of their computational cost when higher accuracy is needed. This problem stems from the need to evolve all moments up to a higher order to increase accuracy. Worse still, these methods can exhibit numerical instabilities when third- or higher-order moments are evolved [107].

Chapter 3

Theoretical Foundation

Abstract

This chapter presents an overview of the theoretical foundation behind the main methodologies of this thesis. First, a framework for turbulent closure schemes is described for systems with a energy-preserving quadratic nonlinearity. This class of systems includes standard toy models like the Lorentz-96 system as well as more complex models like the Navier-Stokes equations and quasi-geostrophic flows. An expansion for the first and second order statistics of such systems is derived and the closure terms that need to be determined are highlighted and discussed. Then, a discussion on supervised learning for neural networks is discussed. Afterwards, an overview on constructing and training (in a supervised manner) neural network models for sequence to sequence mapping is presented. Fully-connected neural networks, recurrent neural networks and convolutional neural networks are described and explained.

3.1 Second-order expansion of systems with quadratic nonlinearity

In the following two chapters, this thesis combines a second-order statistical formulation with neural networks that are trained under appropriate physical constraints. This framework produces accurate uncertainty quantification predictions for nonlinear dynamical problems. Let $\boldsymbol{u}$ be a field describing the state of the system. The evolution equation of $\boldsymbol{u}$ has the following general form

$$\frac{d\boldsymbol{u}}{dt} = A\boldsymbol{u} + \boldsymbol{B}\left(\boldsymbol{u}, \boldsymbol{u}\right) + \boldsymbol{F}(t) + \dot{W}_k(t;\omega)\boldsymbol{\sigma}_k(t), \tag{3.1}$$

where A is a linear operator, $\boldsymbol{F}$ denotes a deterministic external forcing and $\dot{W}_k\boldsymbol{\sigma}_k$ corresponds to a stochastic forcing with white noise characteristics. The operator $\boldsymbol{B}$ is assumed to be quadratic and energy-preserving, i.e.

$$\boldsymbol{B}(\boldsymbol{u}, \boldsymbol{u}) \cdot \boldsymbol{u} = 0. \tag{3.2}$$

This restrictive definition of $\boldsymbol{B}$ is valid for many important problems in fluid mechanics, retaining the physical relevance of the formulation. For example, $\boldsymbol{B}$ can be viewed as the advection term of turbulent flows, a class of problems that has historically attracted the attention of reduced-order modelling literature.

Using well-known linear algebra results [164], the linear operator A can be decomposed as

$$A = \frac{1}{2}(A - A^T) + \frac{1}{2}(A + A^T) = L + D, \tag{3.3}$$

where L is a skew-symmetric linear operator and D is a symmetric linear operator. Throughout this work D will also be assumed to be negative definite, which implies that it corresponds to a linear dissipative process. The quantity of interest $\boldsymbol{u}$ is analyzed using the finite-dimensional expansion

$$\boldsymbol{u} = \overline{\boldsymbol{u}} + \boldsymbol{u}' = \overline{\boldsymbol{u}} + \sum_{i=1}^{N} Z_i(t;\omega)\boldsymbol{v}_i, \tag{3.4}$$

where $\boldsymbol{v}_i$ form a prescribed orthonormal basis, while Z_i are zero-mean, time-dependent random functions. The symbol ω denotes the random argument and the mean field $\overline{\boldsymbol{u}}$ can be interpreted as an ensemble average, $\mathbb{E}\left[\boldsymbol{u}\right] = \overline{\boldsymbol{u}}$, with respect to ω. Using the above representation, the original dynamical equation can be re-written as

$$\begin{aligned} \frac{d\overline{\boldsymbol{u}}}{dt} + \frac{d\boldsymbol{u}'}{dt} = [L+D]\overline{\boldsymbol{u}} + [L+D]\boldsymbol{u}' + \boldsymbol{B}(\overline{\boldsymbol{u}},\overline{\boldsymbol{u}}) + \boldsymbol{B}(\boldsymbol{u}',\overline{\boldsymbol{u}}) \\ +\boldsymbol{B}(\overline{\boldsymbol{u}},\boldsymbol{u}') + \boldsymbol{B}(\boldsymbol{u}',\boldsymbol{u}') + \boldsymbol{F} + \dot{W}_k(t;\omega)\boldsymbol{\sigma}_k(t). \end{aligned} \tag{3.5}$$

By taking the expectation of the above equation, we derive the dynamical equation for the average state $\overline{\boldsymbol{u}}$, i.e.

$$\frac{d\overline{\boldsymbol{u}}}{dt} = [L+D]\overline{\boldsymbol{u}} + \boldsymbol{B}(\overline{\boldsymbol{u}},\overline{\boldsymbol{u}}) + \sum_{i=1}^{N}\sum_{j=1}^{N} \overline{Z_i Z_j}\boldsymbol{B}(\boldsymbol{v}_i,\boldsymbol{v}_j) + \boldsymbol{F}. \tag{3.6}$$

For a second-order statistical framework, an evolution equation for the covariance matrix $R_{ij} = \overline{Z_i Z_j}$ is also required. To this end, first the equation for the perturbations $\boldsymbol{u}' = \boldsymbol{u} - \overline{\boldsymbol{u}}$ is derived:

$$\begin{aligned} \frac{d\boldsymbol{u}'}{dt} =& [L+D]\boldsymbol{u}' + \boldsymbol{B}(\overline{\boldsymbol{u}},\boldsymbol{u}') + \boldsymbol{B}(\boldsymbol{u}',\overline{\boldsymbol{u}}) \\ &+ \sum_{i=1}^{N}\sum_{j=1}^{N}[Z_i Z_j - R_{ij}]\boldsymbol{B}(\boldsymbol{v}_i,\boldsymbol{v}_j) + \dot{W}_k\boldsymbol{\sigma}_k. \end{aligned} \tag{3.7}$$

The projection of the evolution equation for $\boldsymbol{u}'$ onto a basis function $\boldsymbol{v}_n$, yields

$$\begin{aligned}\frac{dZ_n}{dt} &= + \sum_{i=1}^{N} Z_i[A\boldsymbol{v}_i + \boldsymbol{B}(\overline{\boldsymbol{u}}, \boldsymbol{v}_i) + \boldsymbol{B}(\boldsymbol{v}_i, \overline{\boldsymbol{u}})] \cdot \boldsymbol{v}_n \\ &+ \sum_{i=1}^{N} \sum_{j=1}^{N} [Z_i Z_j - R_{ij}] \boldsymbol{B}(\boldsymbol{v}_i, \boldsymbol{v}_j) \cdot \boldsymbol{v}_n + \dot{W}_k \boldsymbol{\sigma}_k \cdot \boldsymbol{v}_n.\end{aligned} \tag{3.8}$$

By multiplying the above equation with Z_m and taking the ensemble average, we have

$$\begin{aligned}\overline{\frac{dZ_n}{dt} Z_m} &= + \sum_{i=1}^{N} R_{im}[A\boldsymbol{v}_i + \boldsymbol{B}(\overline{\boldsymbol{u}}, \boldsymbol{v}_i) + \boldsymbol{B}(\boldsymbol{v}_i, \overline{\boldsymbol{u}})] \cdot \boldsymbol{v}_n \\ &+ \sum_{i=1}^{N} \sum_{j=1}^{N} \overline{Z_i Z_j Z_m} \boldsymbol{B}(\boldsymbol{v}_i, \boldsymbol{v}_j) \cdot \boldsymbol{v}_n + Z_m \dot{W}_k \boldsymbol{\sigma}_k \cdot \boldsymbol{v}_n,\end{aligned} \tag{3.9}$$

since $\overline{R_{ij} Z_m} = R_{ij} \overline{Z_m} = 0$. Hence, the evolution of the elements of the covariance matrix is dictated by the equation

$$\begin{aligned}\frac{d}{dt} R_{mn} &= \sum_{i=1}^{N} R_{im}[A\boldsymbol{v}_i + \boldsymbol{B}(\overline{\boldsymbol{u}}, \boldsymbol{v}_i) + \boldsymbol{B}(\boldsymbol{v}_i, \overline{\boldsymbol{u}})] \cdot \boldsymbol{v}_n \\ &+ \sum_{i=1}^{N} R_{in}[A\boldsymbol{v}_i + \boldsymbol{B}(\overline{\boldsymbol{u}}, \boldsymbol{v}_i) + \boldsymbol{B}(\boldsymbol{v}_i, \overline{\boldsymbol{u}})] \cdot \boldsymbol{v}_m \\ &+ \sum_{i=1}^{N} \sum_{j=1}^{N} \overline{Z_i Z_j Z_m} \boldsymbol{B}(\boldsymbol{v}_i, \boldsymbol{v}_j) \cdot \boldsymbol{v}_n \\ &+ \sum_{i=1}^{N} \sum_{j=1}^{N} \overline{Z_i Z_j Z_n} \boldsymbol{B}(\boldsymbol{v}_i, \boldsymbol{v}_j) \cdot \boldsymbol{v}_m \\ &+ (\boldsymbol{\sigma}_k \cdot \boldsymbol{v}_m)(\boldsymbol{\sigma}_k \cdot \boldsymbol{v}_n).\end{aligned} \tag{3.10}$$

The evolution equation for the covariance of the system can then be written as

$$\frac{d\mathbf{R}}{dt} = \mathcal{L}\mathbf{R} + \mathbf{R}\mathcal{L}^* + \mathcal{Q}_\sigma + \mathcal{Q}, \tag{3.11}$$

where $\mathcal{L}$ is a linear operator expressing dissipation and energy transfers between the mean field and the stochastic modes

$$\mathcal{L}_{ij} = \left[(L + D)\,\boldsymbol{v}_j + \boldsymbol{B}(\overline{\boldsymbol{u}}, \boldsymbol{v}_j) + \boldsymbol{B}(\boldsymbol{v}_j, \overline{\boldsymbol{u}})\right] \cdot \boldsymbol{v}_i, \tag{3.12}$$

with $\mathcal{L}^*$ being its transpose. The operator $(\mathcal{Q}_\sigma)_{ij} = (\boldsymbol{\sigma}_k \cdot \boldsymbol{v}_j)\,(\boldsymbol{\sigma}_k \cdot \boldsymbol{v}_i)$ models effects due to the stochastic external forcing and $\mathcal{Q}$ corresponds to third-order effects that express energy fluxes between different stochastic modes

$$\begin{aligned} \mathcal{Q}_{mn} = &\sum_{i=1}^{N}\sum_{j=1}^{N} \overline{Z_i Z_j Z_m} \boldsymbol{B}(\boldsymbol{v}_i, \boldsymbol{v}_j) \cdot \boldsymbol{v}_n \\ &+ \sum_{i=1}^{N}\sum_{j=1}^{N} \overline{Z_i Z_j Z_n} \boldsymbol{B}(\boldsymbol{v}_i, \boldsymbol{v}_j) \cdot \boldsymbol{v}_m. \end{aligned} \tag{3.13}$$

In more detail, $\mathcal{L}$ includes the effects of the linear operators on each mode, as well as energy transfers between the mean and each mode via the energy-preserving nonlinear operator. All these effects can be studied and understood under the framework of second-order statistics. It is the mode-to-mode nonlinear energy transfers that require knowledge of higher moments for their estimation. In order to close the system for the covariance and the mean, a model for the third-order terms $\mathcal{Q}$ appearing in Eq. (3.10) is required.

3.2 Neural networks for time sequences

3.2.1 Supervised learning

Supervised machine learning is a subcategory of ML, revolving around training a predictive model based on a limited number of labeled observations. The goal of the process is accurate prediction of the target system's output. Ideally, within the functional space of possible models, supervised learning seeks to find a function $h(\cdot)$ that best approximates the response of the system. The quality of approximation is measured by the risk functional:

$$M(h) = \int \mathcal{L}\left(h(x), y\right) P(x, y) \mathrm{d}x \mathrm{d}y, \tag{3.14}$$

where the loss $\mathcal{L}$ measures the discrepancy between the output produced by h and the true system y for a given input x. Probability distribution $P(x, y)$ corresponds to the sampled data. However, in general, $P(x, y)$ is not known, and thus task is reduced to minimizing risk over the finite number of training data points. Hence, the task becomes

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(h(x^{(i)}; \theta), y^{(i)}\right), \tag{3.15}$$

where θ denotes a finite set of parameters that describe the neural network. $\mathcal{D}_{\text{train}} = \left\{x^{(i)}, y^{(i)}\right\}_{i=1}^{N}$, denotes the training dataset. Defining the appropriate model space Θ is crucial for the success of a supervised ML problem, especially for generalizing the model outside the training set. This thesis will explore appropriate constraints and modelling choices for supervised learning in turbulent flows and nonlinear dynamical systems.

3.2.2 Fully-connected neural networks

Fully-connected (feedforward) neural networks consist of a number of neurons, each representing a scalar variable of the system under investigation. The term feedforward denotes the fact that neurons in one layer are only dependent on ones in the previous layer. These NNs comprise the most basic class of NN models. The functional dependence between the vector input $\boldsymbol{x}$ and scalar output z of a single neuron has the generic form

$$z = a\left(\boldsymbol{w}^T \boldsymbol{x} + b\right), \tag{3.16}$$

Input vector $\boldsymbol{x}$ consists of the neuron predictions of the previous layer. Parameters $\boldsymbol{w}$ and $\boldsymbol{b}$ are respectively the weight matrix and bias vector of the neuron. Function $a\left(\cdot\right)$ is known as the activation function and is generally nonlinear. For a given input and a fixed network structure, we can easily calculate the model output and measure the quality of prediction on the given data by computing a predefined cost function:

$$\mathcal{J}(\theta) = \sum_n \boldsymbol{\mathcal{L}}\left(\boldsymbol{z}^{(n)}(\boldsymbol{x}^{(n)}, \theta), \boldsymbol{y}^{(n)}\right), \tag{3.17}$$

where $\boldsymbol{y}$ denotes the ground truth corresponding to a prediction $\boldsymbol{z}$, and θ represent the set of all model parameters including weights and biases. The cost sums over all cases (indexed by n), where a loss function $\boldsymbol{\mathcal{L}}$ is used to measure the difference for a single prediction-truth pair. The overall cost is minimized with respect to parameters θ, which defines the model that can be utilized to make predictions for unseen input.

By varying $\boldsymbol{w}$ and $\boldsymbol{b}$ one obtains a somewhat limited range of functions mapping $\boldsymbol{x}$ to $\boldsymbol{z}$. Hence, rich representations are achieved by stacking several fully-connected layers on top of each other. The expressive power of the resulting model significantly multiplies as the number of layers and neurons increase, making it capable of accurately approximating a broad range of functions. Networks with more than one layer (called hidden) between the first (input) and the last (output) are referred to as deep neural networks. A schematic of such networks is shown in fig. 3-1. Conveniently, the gradient of the cost function in this form can be calculated via a well-known algorithm known as backpropagation [142] which calculates the error and gradient first for the last layer and moves layer-by-layer back to the first.

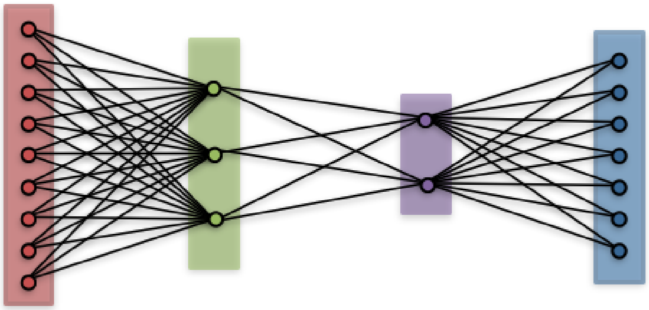


FIG. 3-1. A four-layer fully-connected neural netowrk mapping a 9-dimensional input to a 7-dimensional output.

3.2.3 Recurrent neural networks

Recurrent neural networks (RNNs) are a special class of NNs specifically designed for sequence to sequence mapping. Such systems may exhibit strong non-Markovian characteristics and thus NNs that preserve information from previous time-steps are needed. RNNs incorporate information from previous states of the system via a vector called hidden state. This variable acts as an internal memory for the model providing non-Markovian properties to the trained model. In a generic framework, the model is parametrized by weight matrices $\{\boldsymbol{W}_{hx}, \boldsymbol{W}_{hh}, \boldsymbol{W}_{oh}\}$ and bias vectors $\{\boldsymbol{b}_h, \boldsymbol{b}_o\}$. All tunable parameters, via stochastic gradient descent, are jointly denoted by θ. Given an input sequence $\{\boldsymbol{x}_t\}_{t=1}^{T}$, the RNN computes the sequence

$$\begin{aligned} \boldsymbol{h}_t &= \sigma_h\left(\boldsymbol{W}_{hx}\boldsymbol{x}_t + \boldsymbol{W}_{hh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_h\right) \\ \boldsymbol{z}_t &= \sigma_z\left(\boldsymbol{W}_{zh}\boldsymbol{h}_t + \boldsymbol{b}_z\right), \end{aligned} \tag{3.18}$$

where $\{\boldsymbol{h}_t\}_{t=1}^{T}$ denotes the hidden state and a sequence of outputs $\{\boldsymbol{z}_t\}_{t=1}^{T}$ the output sequence. Activation functions $\sigma_h(\cdot)$ and $\sigma_z(\cdot)$ are general nonlinear functions. A schematic of an RNN is shown in fig. 3-2.

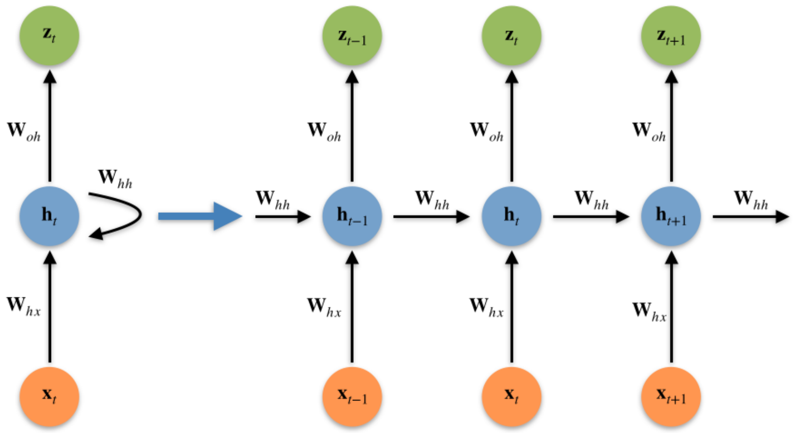


FIG. 3-2. Computational graph for recurrent neural network, $\boldsymbol{x}_t, \boldsymbol{h}_t$ and $\boldsymbol{z}_t$ are the input, hidden and output states at time t.

Once again, in a supervised learning setting, model parameters θ are obtained by defining a finite training set of size N and minimizing the cost function

$$\mathcal{J}(\theta) = \sum_{n=1}^{N}\sum_{t=1}^{T}\mathcal{L}\left(\boldsymbol{z}_t^{(n)}\left(\theta, \boldsymbol{X}_{:t}^{(n)}\right), \boldsymbol{y}_t^{(n)}\right), \tag{3.19}$$

where the number of time steps T denotes the number of time delays the model incorporates during training. Memory is always truncated to a finite length to allow for a stable gradient-based optimization scheme.

3.2.4 Long short-term memory

As mentioned in the previous subsection, RNNs experience difficulties to converge to a satisfying solution during training when long-term dependencies are incorporated. This is a result of the loss function containing implicit dependencies from the output of all previous time steps. As a result, the nested form of these dependecies turn into a large product of terms when calculating the derivatives of the loss function during stochastic gradient descent. Hence, if most of these terms are larger than 1, the gradient would diverge and the model fail to converge. On the other hand, if most terms are smaller than 1, the gradient would vanish and the model would get stuck in sub-optimal position of the state space [12].

The long short-term memory (LSTM) networks [64], seeks to address this issue through introducing specialized units known as memory cells. This type of cell is a self-recurrent neuron, whose prediction only undergoes linear transformations. The prediction of the model interacts with the input and memory state via the so-called input, forget and output gates. In particular, the states of the model are governed by the following evolution equations

$$\begin{aligned}
\boldsymbol{i}_t &= \sigma\left(\boldsymbol{W}_{ih}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{ix}\boldsymbol{x}_t + \boldsymbol{b}_i\right) \\
\boldsymbol{f}_t &= \sigma\left(\boldsymbol{W}_{fh}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{fx}\boldsymbol{x}_t + \boldsymbol{b}_f\right) \\
\boldsymbol{o}_t &= \sigma\left(\boldsymbol{W}_{oh}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{ox}\boldsymbol{x}_t + \boldsymbol{b}_o\right) \\
\tilde{\boldsymbol{C}}_t &= \tanh\left(\boldsymbol{W}_{ch}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{cx}\boldsymbol{x}_t + \boldsymbol{b}_c\right) \\
\boldsymbol{C}_t &= \boldsymbol{i}_t \circ \tilde{\boldsymbol{C}}_t + \boldsymbol{f}_t \circ \boldsymbol{C}_{t-1} \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \circ \tanh\left(\boldsymbol{C}_t\right),
\end{aligned} \tag{3.20}$$

where $\sigma(\cdot)$ is the sigmoid function and $(\cdot)$ denotes element-wise multiplication. Vectors $\boldsymbol{i}_t$, $\boldsymbol{f}_t$ and $\boldsymbol{o}_t$ are the input, forget and output gates respectively. $\boldsymbol{C}_t$ represents the cell memory (formally called cell state) calculated as a linear combination of a new candidate cell state $\tilde{\boldsymbol{C}}_t$ and the previous memory $\boldsymbol{C}_{t-1}$, weighted by the input and forget gate values. $\boldsymbol{h}_t$ is the output of the LSTM and can be regarded as a nonlinear copy of the cell state that helps to decide how to process the input in the next step.

All states of the system have the same dimension. This particular setup allows the model to drop dependencies from past time steps that are not important and thus the vanishing gradient problem is resolved. In total, 8 weight matrices and 4 bias vectors make up the parameter space of a single LSTM unit. Despite having more parameters and a much more intricate forward pass, LSTM can be easily implemented using software packages equipped with automatic differentiation.

3.2.5 Convolutional neural networks

All the architectures described so far comprise of a number of fully-connected NNs. Such an approach implies that all possible states of the system can influence all other states as well at any time. However, this property usually does not hold true in most dynamical systems. In fact, generally a localized spatial or temporal region is only important for predictions at a particular location. The convolutional neural networks (CNNs) are another class of neural network models capable of incorporating this property. They are characterized by the convolution operator (1-D)

$$\boldsymbol{z}_t = \left(\boldsymbol{F} *_d \boldsymbol{x}\right)(t) = \sum_{k=0}^{K} \boldsymbol{F}_k \boldsymbol{x}_{t-k}. \tag{3.21}$$

where $\boldsymbol{F}$ is a kernel matrix, to be determined during training. The convolution operation can be seen as a sparse variance of the fully connected mode. In addition, this approach introduces an idea of isotropicity in the model, as each output is computed from the same model parameters with a simple positional shift. To incorporate anisotropic effects, multiple convolutions are usually incorporated.

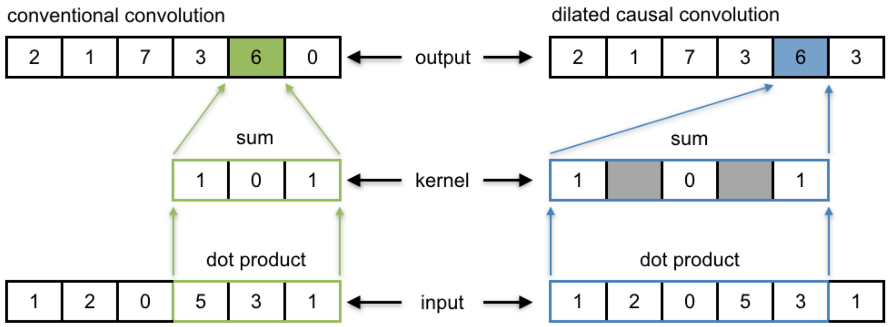


FIG. 3-3. 1D convolution operation for sequences: (left) normal convolution and (right) dilated causal convolution.

While such convolutions can readily be used for temporal sequences, they must be mended if causality is an issue. To that end causal and dilated convolutions [123, 6] were introduced. The general form of this convolution operation is

$$\boldsymbol{z}_t = (\boldsymbol{F} *_d \boldsymbol{x})(t) = \sum_{k=0}^{K} \boldsymbol{F}_k \boldsymbol{x}_{t-d\cdot k}. \tag{3.22}$$

In a dynamical system context, it corresponds to the restriction that output states are strictly determined by historical events, hence being causal. Dilated convolution refers to the fact that two adjacent $\boldsymbol{F}_k$'s are applied to input which are d steps apart, where d is a hyperparameter representing the dilation rate. Using a large dilation rate allows the output to depend on a wider range of input. A schematic of causal convolutions is shown in fig. 3-3.

Chapter 4

Energy preserving closure-schemes for first-order statistics

Abstract

In this chapter we present an online turbulence closure scheme for the first-order statistics of turbulent flows. While we use the problem formulation presented in chapter 3, we restrict ourselves to the study of the mean equation only. Second-orders statistics will be studied in chapter 5. We first formulate the objective function used in the training phase. This step also includes the physical constraint and its derivation using Gauss theorem. We subsequently consider a forced two-dimensional jet flow. We first take into account the invariance of the flow in one direction to derive one-dimensional machine-learned closures using DNS information. As a next step, we apply the method on the computation of two-dimensional turbulent closures that do not rely on this special symmetry. We compare the obtained coarse-scale model with DNS and assess its generalizability properties for different Reynolds numbers, as well as different jet profiles which have not been used in the training phase. We thoroughly examine the role of the physical constraint on the stability properties and accuracy of the coarse-scale equations. In addition, we assess our closure scheme on capturing the evolution of concentration for inertial tracers, such as bubbles and aerosols. The results presented in this chapter were published here [30].

4.1 Formulation of energy-preserving closure schemes

Our aim is to derive Eulerian, data-driven closure schemes for turbulent fluid flows, as well as for inertial tracers advected by those. These closure schemes will not only rely on DNS training data, but also on the physical constraint that follows from the energy conservation principles that the nonlinear advection terms satisfy [147, 102]. The effectiveness of the closure schemes is assessed by how well the coarse-scale equations can reproduce the mean flow characteristics for problems that reach a statistical equilibrium. Higher order closures may be utilized to improve predictions for higher order statistics such as the flow spectrum. However, in this work we will focus on closures that aim to model the mean flow characteristics.

First, we introduce a spatial-averaging operator that will define the coarse scale version of the quantities of interest and their evolution equations. Specifically, we decompose any field of interest f as

$$f = \overline{f} + f', \tag{4.1}$$

similarly to eq. (3.4), where $\overline{f}$ corresponds to the large-scale component of the quantity and f' corresponds to the small-scale component. As a result we always have $\overline{f'} = 0$.

4.1.1 Averaged Navier-Stokes equations

We consider the incompressible Navier-Stokes equations in dimensionless form:

$$\frac{D\mathbf{u}}{Dt} = -\nabla p + \frac{1}{\text{Re}}\Delta \mathbf{u} + \nu \nabla^{-4}\mathbf{u} + \mathbf{F}, \tag{4.2}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{4.3}$$

where $\mathbf{u}$ is the velocity field of the fluid, p its pressure, Re is the Reynolds number of the flow, $\frac{D}{Dt}$ is the material derivative operator and $\mathbf{F}$ denotes an external forcing term. Parameter ν is a hypoviscosity coefficient aiming to remove energy from large scales and maintain the flow in a turbulent regime. Using the decomposition eq. (4.1) into the fluid flow eqs. eq. (4.2) and applying the averaging operator we obtain:

$$\partial_t \overline{\mathbf{u}} = -\overline{\mathbf{u}} \cdot \nabla \overline{\mathbf{u}} - \overline{\mathbf{u}' \cdot \nabla \mathbf{u}'} - \nabla \overline{p} + \frac{1}{\mathrm{Re}} \Delta \overline{\mathbf{u}} + \nu \nabla^{-4} \overline{\mathbf{u}} + \overline{\mathbf{F}}, \tag{4.4}$$

$$\nabla \cdot \overline{\mathbf{u}} = 0. \tag{4.5}$$

Clearly, the averaged evolution equations do not comprise a closed system anymore due to the nonlinearity of the advection term. As a result, the term $\overline{\mathbf{u}' \cdot \nabla \mathbf{u}'}$ is not defined by the evolution equations and needs to be parametrized.

4.1.2 Averaged advection equation for inertial tracers

One can follow a similar process for the advection equation governing the motion of small inertial tracers. In particular for small inertia particles their Lagrangian velocity, $\mathbf{v}$, is a small perturbation of the underlying fluid velocity field [54, 55]:

$$\mathbf{v} = \mathbf{u} + \epsilon \left(\frac{3R}{2} - 1 \right) \frac{D\mathbf{u}}{Dt} + O(\epsilon^2), \quad \epsilon = \frac{St}{R} \ll 1, \quad R = \frac{2\rho_f}{\rho_f + 2\rho_p}, \tag{4.6}$$

where $\epsilon = \frac{St}{R} \ll 1$ is a parameter representing the importance of inertial effects, St is the particle or bubble Stokes number, and $R = \frac{2\varrho_f}{\varrho_f + 2\varrho_p}$ is a density ratio with ϱ_p and ϱ_f being the density of the particle or bubble and the flow respectively. Eq. (4.6), arises from geometric singular perturbation theory in the limit of small particle inertia [54]. For this asymptotic limit the presented manifold is always attracting trajectories exponentially fast [55]. However, errors due to the finite order truncation of the asymptotic expansion will result in inaccuracies for time scales larger than $O(\epsilon^{-2})$.

By introducing ρ as the concentration of tracers at a particular point, we can write the following transport equation

$$\partial_t \rho + \nabla \cdot (\mathbf{v}\rho) = \nu_2 \Delta^4 \rho. \tag{4.7}$$

The right-hand-side of the transport equation represents a hyperviscosity term. Introducing the decomposition of eq. (4.1) in the evolution eqs. (4.6) and (4.7), we obtain

$$\overline{\mathbf{v}} = \overline{\mathbf{u}} + \epsilon \left(\frac{3R}{2} - 1 \right) \left(\partial_t \overline{\mathbf{u}} + \overline{\mathbf{u}} \cdot \nabla \overline{\mathbf{u}} + \overline{\mathbf{u}' \cdot \nabla \mathbf{u}'} \right), \tag{4.8}$$

$$\partial_t \overline{\rho} + \nabla \cdot (\overline{\mathbf{v}} \overline{\rho}) + \overline{\nabla \cdot (\mathbf{v}' \rho')} = \nu_2 \Delta^4 \overline{\rho}. \tag{4.9}$$

Once again, the closure term $\overline{\nabla \cdot (\mathbf{v}' \rho')}$ appears, which requires parametrization. Note that the evolution equations of the carrier fluid (eq. (4.4) and (4.5) and the transported inertial particles (eq. (4.8) and (4.9)) are both in dimensionless form.

4.1.3 Data-driven parametrization of the closure terms

While the full Navier-Stokes equations and the associated advection equations are Markovian and spatially-local, i.e. the evolution of the flow or concentration field in a specific location and time instant depends only on the current time instant and the current neighborhood, this is not necessarily the case for the averaged version of these equations. In particular, for the averaged equations we typically do not have access to the full-state information, required to fully describe the evolution of the system. In this case the missing information is the small-scale dynamics.

To overcome this limitation we recall Takens embedding theorem [168], which states that if we observe only a limited number of the state variables of a system, in principle, we can still obtain the attractor of the full system by using delay embeddings of the observed state variables. In other words, under appropriate conditions there is a map between the delays of the observed state variables and the full state system. Although the theorem itself is several decades old, we can now rely on recently developed data-driven schemes that can implement such mapping as part of their training process, enhancing the accuracy of predictions (see e.g. [178, 180]). To this end, we parametrize the closure terms with *non-local in time* (but still causal) representations, based on Long-Short-Term Memory (LSTM) recurrent neural networks (RNN) and Temporal Convolutional Networks (TCN) [64]. The specific RNN implementation was picked based on its tested ability to incorporate long-term memory effects of hundreds of time-delays, while simpler RNN models suffer from vanishing or exploding gradients [12].

Beyond non-locality in time we also choose to employ non-locality in space. That is, given a point in space $\mathbf{x}$, we use information from points that lie in a small neighborhood of $\mathbf{x}$. Clearly, incorporating information from the entirety of the domain is not only computationally infeasible but also redundant and can lead to stability issues. For this reason we use convolutions in space to make sure that we incorporate information only from a region around each point and not from the entire domain. The parameterization is based on a stacked LSTM architecture [50], which utilizes LSTM layers with the detail that all input and recurrent transformations are convolutional.

As a result, the closure terms are modeled in the following form:

$$\begin{aligned} \overline{\mathbf{u'} \cdot \nabla \mathbf{u'}}(\mathbf{x}, t) &= \mathbb{D}_{\mathbf{u}} \left[\theta_1; \overline{\xi}[\alpha(\mathbf{x}), \chi(t)]\right] , \\ \overline{\nabla \cdot (\mathbf{v'} \rho')}(\mathbf{x}, t) &= \mathbb{D}_{\rho} \left[\theta_2; \overline{\zeta}[\alpha(\mathbf{x}), \chi(t)]\right] , \end{aligned} \tag{4.10}$$

where $\overline{\xi}$ and $\overline{\zeta}$ are (averaged) flow features to be selected, $\alpha(\mathbf{x})$ denotes a pre-selected neighborhood of points around $\mathbf{x}$ over which the averaged state is considered, i.e. $\alpha(\mathbf{x}) = \{\mathbf{x}, \mathbf{x} + \mathbf{x}_1, \mathbf{x} + \mathbf{x}_2, ..., \mathbf{x} + \mathbf{x}_N\}$, and $\chi(t)$ denotes the history of the averaged state backwards from time t, i.e. $\chi(t) = \{t, t - \tau_1, ..., t - \tau_2, ..., t - \tau_N\}$. The vectors θ_1 and θ_2 denote the hyperparameters of the neural networks and their optimization is performed empirically. The spatial neighborhood, $\alpha(\mathbf{x})$, is selected such that if we further increase it, the training error does not significantly reduce any more. Note however, that the number of points in space that have to be considered in $\alpha(\mathbf{x})$ is dependent on the discretization of the domain, i.e., if we increase the resolution of our model the number of neighborhood points in $\alpha(\mathbf{x})$ should increase respectively so that the input information used as input in the closure always corresponds to the same spatial neighborhood. In the numerical study that follows, the the effect of spatial discretization is investigated. We use a similar approach for the temporal history, $\chi(t)$.

Temporal integration

We point out that our numerical goal is inline prediction. This means that the neural nets described by eq. (4.10) must be coupled with the evolution eqs. (4.4)-(4.9). For a simple forward Euler scheme for temporal integration, this would imply that by knowing the values of $\overline{\mathbf{u}}, \overline{\rho}$ at time t we can predict the closure terms at time t using eq. (4.10) and use their values to integrate eqs. (4.4)-(4.9) by one time-step δt so that we compute $\overline{u}_1, \overline{\rho}$ at time $t + \delta t$. However, if we want to use a higher-order integration scheme like a 4th-order explicit Runge-Kutta, we need to evaluate the closure terms at time $t + \delta t/2$ as well. Since, we do not have access to the required time-history for such a prediction, we instead integrate in time not by δt but by $2\delta t$ and thus get a time-integration error of the for $O[(2\delta t)^4]$.

4.1.4 Physical constraints

An important feature of our data-driven closure schemes is the requirement to satisfy certain physical principles. Specifically, we utilize the energy flux constraint that the advection term does not alter the total kinetic energy of the model [147, 150]. This constraint, follows from Gauss identity. Specifically, for any scalar fielsd, ψ, β, and divergence-free field, Φ, we have from Gauss identity:

$$\int_\Omega \frac{\partial \psi}{\partial x_j} \beta \Phi_j d\mathbf{x} = -\int_\Omega \frac{\partial \beta}{\partial x_j} \psi \Phi_j d\mathbf{x} + \int_{\partial\Omega} \psi \beta \Phi_j n_j d\mathbf{x}, \tag{4.11}$$

where n_j is the unit vector on the boundary, $\partial\Omega$. Applying the above for $\psi = \beta = u_k$ and $\Phi_j = u_j$, we obtain the general three-dimensional constraint:

$$\int_\Omega \mathbf{u} \cdot (\mathbf{u} \cdot \nabla)\mathbf{u} \mathrm{d}\mathbf{x} = \int_{\partial\Omega} \mathcal{E}\ \mathbf{u} \cdot \mathbf{n}\ d\mathbf{x}, \quad \mathcal{E} = \frac{1}{2}\mathbf{u} \cdot \mathbf{u} \tag{4.12}$$

where Ω is the domain in which the fluid flow is defined. The above constraint essentially expresses the fact that the nonlinear advection terms do not change the total kinetic energy of the system. In what follows we will consider the case of periodic boundary conditions, where the right hand side in (4.12) vanishes. However, the same ideas are applicable for arbitrary boundary conditions. We apply the decomposition (4.1) and the spatial averaging operator to this equation and obtain:

$$\begin{aligned}\int_\Omega \overline{\mathbf{u}} \cdot (\overline{\mathbf{u}} \cdot \nabla)\overline{\mathbf{u}} \mathrm{d}\mathbf{x} + \int_\Omega \overline{\mathbf{u}} \cdot \overline{(\mathbf{u}' \cdot \nabla)\mathbf{u}'} \mathrm{d}\mathbf{x} + \int_\Omega \overline{\mathbf{u}' \cdot (\overline{\mathbf{u}} \cdot \nabla)\mathbf{u}'} \mathrm{d}\mathbf{x} \\ + \int_\Omega \overline{\mathbf{u}' \cdot (\mathbf{u}' \cdot \nabla)\overline{\mathbf{u}}} \mathrm{d}\mathbf{x} + \int_\Omega \overline{\mathbf{u}' \cdot (\mathbf{u}' \cdot \nabla)\mathbf{u}'} \mathrm{d}\mathbf{x} = 0,\end{aligned} \tag{4.13}$$

From the last equation we have the *physical constraint* that the closure term $\mathbb{D}_{\mathbf{u}}$ must satisfy

$$\begin{aligned}\int_\Omega \overline{\mathbf{u}} \cdot \mathbb{D}_{\mathbf{u}} \left[\theta_1; \overline{\xi}[\alpha(\mathbf{x}), \chi(t)]\right] d\mathbf{x} = A[\mathbf{u}] \triangleq - \int_\Omega \overline{\mathbf{u}} \cdot (\overline{\mathbf{u}} \cdot \nabla)\overline{\mathbf{u}} \mathrm{d}\mathbf{x} - \int_\Omega \overline{\mathbf{u}' \cdot (\overline{\mathbf{u}} \cdot \nabla)\mathbf{u}'} \mathrm{d}\mathbf{x} \\ - \int_\Omega \overline{\mathbf{u}' \cdot (\mathbf{u}' \cdot \nabla)\overline{\mathbf{u}}} \mathrm{d}\mathbf{x} - \int_\Omega \overline{\mathbf{u}' \cdot (\mathbf{u}' \cdot \nabla)\mathbf{u}'} \mathrm{d}\mathbf{x},\end{aligned} \tag{4.14}$$

where $A[\mathbf{u}]$ is a function that depends on the training data and the discretization. Such a constraint can be added to the training process in a straightforward manner through the objective function. We emphasize that one could formulate a physical constraint based e.g. on the Navier-Stokes equations directly. However, this assumes exact knowledge of the flow-specifics. This is not the case here since the above constraint expresses a universal property, i.e. that advection terms do not create or destroy energy.

4.1.5 Objective function for training

In terms of the training process itself, we normalize the input and output data as usually suggested (see e.g. [152]). The loss function for this problem is chosen to be the single-step prediction mean square error superimposed with the physical constraint. This can be formulated as

$$L(\theta_1) = \int_{\Omega\times T} \left|\left|\mathbb{D}_{\mathbf{u}}\left[\theta_1; \overline{\xi}\right] - \overline{(\mathbf{u}' \cdot \nabla)\mathbf{u}'}\right|\right|^2 d\mathbf{x}dt + \lambda \int_T \left| \int_\Omega \overline{\mathbf{u}} \cdot \mathbb{D}_{\mathbf{u}}\left[\theta_1; \overline{\xi}\right] d\mathbf{x} - A[\mathbf{u}] \right| dt, \tag{4.15}$$

where λ is a hyperparameter which is chosen so the two terms of the loss function are of the same order of magnitude. More specifically, $\lambda = \lambda^* = 10^1$. It also reported that if $\lambda = \lambda^*/10$ is chosen, then the results are almost identical to the case where the constraint is not used. Furthermore, if $\lambda = 10\lambda^*$ then the generated closure is unstable both for unimodal and bimodal jets. On the other hand, for the advection equation we have the objective function:

$$L(\theta_2) = \int_{\Omega\times T} \left|\left|\mathbb{D}_\rho\left[\theta_2; \overline{\zeta}[\alpha(\mathbf{x}), \chi(t)]\right] - \nabla \cdot \left(\overline{\mathbf{v}'\rho'}\right)\right|\right|^2 d\mathbf{x}dt. \tag{4.16}$$

Note that a similar constraint with the one in eq. (4.15) can be formulated for the mass conservation property of the tracers. However, this approach is not pursued here. An important question is which flow features are important as input for each of the two models. We examine this issue in detail in the following sections.

4.1.6 Imitation learning

While the single-step prediction-error is used for training, the aim this work aims to use these models for multi-step prediction. Any such predictor introduces errors and these compounding errors change the input distribution for future prediction steps, breaking the train-test independent and identically distributed (i.i.d) assumption that is common in supervised learning. Under these circumstances, the error can be shown to grow exponentially [177]. This effect was observed in the current setup as well, with the averaged equations becoming unstable. To alleviate this problem a version of the imitation learning presented in [177], the Data as Demonstrator (DAD) method, is used. It is also noted that since in the current setup, the evolution equation of the carrier fluid is independent of the evolution of the transported particles, the process can be showcased for only the carrier fluid closure. This process is shown in an algorithmic manner in Algorithm 1.

Expanding on what Algorithm 1 presents, the algorithm first computes the reference flow features $\overline{\mu}^{\text{Ref}}$ and reference closure terms values $\mathbb{D}_{\mathbf{u}}^{\text{Ref}}$ from DNS. Then, the neural network for the carrier fluid closure-term $\mathbb{D}_{\mathbf{u}}^{\text{Ref}}$ is trained. For the next step, define a fixed number of time-steps M_T and an error tolerance δ_1. The goal of this stage is, for all time-instances that are included in the training set as the fluid flow is evolved for M_T time-steps forward in time, the L^2-error at the final time-step is smaller than δ_1. This condition has to be satisfied for all flows that are included in the training set. If this condition is satisfied, the closure is assumed to be stable and thus the training process is rendered complete. If not, for each initial training point, mark as M_s the time-step forward in time at which the condition was first violated. The imitation algorithm requires that an artificial data point $(\overline{\mu}^*|_{t=t+M_s}, \mathbb{D}_{\mathbf{u}}^*(t+M_s))$ is created, which corresponds to the value of the closure term that is required, so that the flow features return to their appropriate reference value $\mu^{\text{DNS}}(t+M_s+1)$ at the M_s+1 time-step forward. These data-points are not physical solutions of some DNS solver. They artificially introduce stability to the closure scheme so as to allow the model to return close to the training data when it deviates. This extra amount of data are introduced to the previous set of training data. and steps 1 and 2 of Algorithm 1 are repeated.

Algorithm 1: Training of closure scheme

```
Input: Reference closure terms $\{\mathbb{D}_i^{\text{DNS}}(t_0), ..., \mathbb{D}_i^{\text{DNS}}(t_p)\}$ $i = \mathbf{u}, \rho$, computed
       from DNS
Data: Neural network (NN) architecture, averaging-operator,
      discretization, $\delta t$ and input flow features $\overline{\mu}$
Result: Predicted closure terms $\{\mathbb{D}_i^{\text{ML}}(t_0), ..., \mathbb{D}_i^{\text{ML}}(t_p)\}$ $i = \mathbf{u}, \rho$.
Set $\overline{\mu}^{\text{Ref}} = \overline{\mu}^{\text{DNS}}$ and $\mathbb{D}_{\mathbf{u}}^{\text{Ref}} = \mathbb{D}_{\mathbf{u}}^{\text{DNS}}$;
Train $\mathbb{D}_{\mathbf{u}}^{\text{ML}}$ using $\mathbb{D}_{\mathbf{u}}^{\text{Ref}}$;
for $i = 1, ..., 20$ do
    for $t = 0, \delta t, ..., T\delta t$ do
        $s \to t$;
        while $\left\|\mathbb{D}_{\boldsymbol{u}}^{ML}(s) - \mathbb{D}_{\boldsymbol{u}}^{DNS}(s)\right\| > \delta_1$ do
            Predict $\overline{\mathbf{u}}^{\text{ML}}(s + \delta t)$;
            Predict $\mathbb{D}_{\mathbf{u}}^{\text{ML}}(s + \delta t)$;
            $s \to s + \delta t$;
        end
        Compute $\mathbb{D}_{\mathbf{u}}^{*}(s)$ so that $\overline{\mathbf{u}}^{\text{ML}}(s + \delta t) = \overline{\mathbf{u}}^{\text{DNS}}(s + \delta t)$;
        Set $(\overline{\mu}^{\text{Ref}}, \mathbb{D}_{\mathbf{u}}^{\text{Ref}}) = (\overline{\mu}^{\text{DNS}}, \mathbb{D}_{\mathbf{u}}^{\text{ML}}) \cup (\overline{\mu}^{*}|_{s}, \mathbb{D}_{\mathbf{u}}^{*}(s))$;
    end
    Train $\mathbb{D}_{\mathbf{u}}^{\text{ML}}$ using $\mathbb{D}_{\mathbf{u}}^{\text{Ref}}$;
end
```

This process is repeated until the closure is stable and displays good predictive accuracy in all training cases. A more rigorous display of this condition is seen in Algorithm 1. For the present setup, this was achieved after 20 iterations. The same process is followed for the closure scheme of the transported particles, $\mathbb{D}_\rho = \overline{\nabla \cdot (\mathbf{v}'\rho')}$.

Note that noise, in the form of some artificial colored or white noise, is not added to the training data. However, the use of the DAD algorithm increases the training size by generated new training data that have noise, as a result of the error in the predictions of the neural network as it is propagated as the flow evolves.

4.2 Fluid flow setup

For the validation and assessment of the formulated closures we consider a two-dimensional turbulent jet where bubbles are also advected as passive inertial tracers. Specifically, the velocity field governing the bubbles is different from that of the underlying fluid flow (due to inertia effects), but the bubbles do not affect in any way the underlying fluid flow.

We setup a turbulent jet that fluctuates around a steady-state jet solution, $\mathbf{u}_{jet}$. In its dimensionless form this system of equations can be written as

$$\frac{D\mathbf{u}}{Dt} = -\nabla p + \frac{1}{\mathrm{Re}}\Delta\mathbf{u} + \nu\nabla^{-4}(\mathbf{u} - \mathbf{u}_{jet}) + \mathbf{F}, \tag{4.17}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{4.18}$$

where $\mathbf{u} = (u_1, u_2)$ and $Re = O(10^3)$. The domain is assumed rectangular, doubly periodic, i.e. $\mathbf{x} = (x, y) \in S^2 = [0, 2\pi] \times [0, 2\pi]$. For initial conditions, since we desire anisotropy in our flow, we use Gaussian jet structures of the general form

$$u_{1,jet} = \sum_i A_i \exp\Big[- c_i(y - \beta_i)^2 \Big], \quad u_{2,jet} = 0, \tag{4.19}$$

where A_i, c_i, β_i are parameters. The role of the external forcing term, $\mathbf{F}$, is twofold: i) it contains a large-scale component to maintain the jet structure, by balancing the dissipation term, and ii) it has a small-scale and small-amplitude component to perturb the flow and trigger instabilities so that the we enter a turbulent regime. To achieve turbulence we choose a forcing term that acts only on a specific waveband with $6 \leq \|\mathbf{k}\| \leq 7$. Exciting a flow with a forcing localized only in a narrow wavenumber interval is common practice in the turbulence literature [16, 14, 110, 15].

Therefore, we adopt a form $\mathbf{F} = -\frac{1}{R}\Delta \boldsymbol{u}_{jet} + f$, with f being

$$f(\mathbf{x}, t) = \sum_i A_i(t) \cos(\mathbf{k} \cdot \mathbf{x} + \omega_i), \tag{4.20}$$

where $6 \leq |\mathbf{k}| \leq 7$, $A_i(t)$ are random vectors that follow a Gaussian white noise distribution (each one independent from the other) and ω_i are phases sampled from a uniform distribution over $[0, 2\pi]$. The standard deviation for these amplitudes is set to 0.03. This ensures that the energy and enstrophy inputs are localized in Fourier space and only a limited range of scales around the forcing is affected by the details of the forcing statistics. Furthermore, such a forcing ensures that the system reaches a jet-like statistical steady state after a transient phase. Due to the small-scale forcing being essentially homogeneous in space we can deduce that the statistical steady state profile is only dependent on y (since our large-scale forcing and initial conditions depend only on y). We solve this flow using a spectral method and 256^2 modes.

For the bubbles we use the perturbed advection field (eq. (4.6)) and the corresponding advection equation (4.7). For the simulations presented we use the inertial parameters, $\epsilon = 0.05$ and $R = 2$, which correspond to small bubbles. A typical snapshot of the described flow can be seen in Figure 4-1. During training, unimodal jets of different Reynolds number are used with parameter values $A_1 = 1, c_1 = 2, \beta_1 = \pi$. Furthermore, for testing purposes, bimodal jets are considered with the Reynolds number varying and parameters $A_1 = 1, \beta_1 = 0.8\pi, c_1 = 3$ and $A_2 = 1, \beta_2 = 1.2\pi, c_2 = 3$.

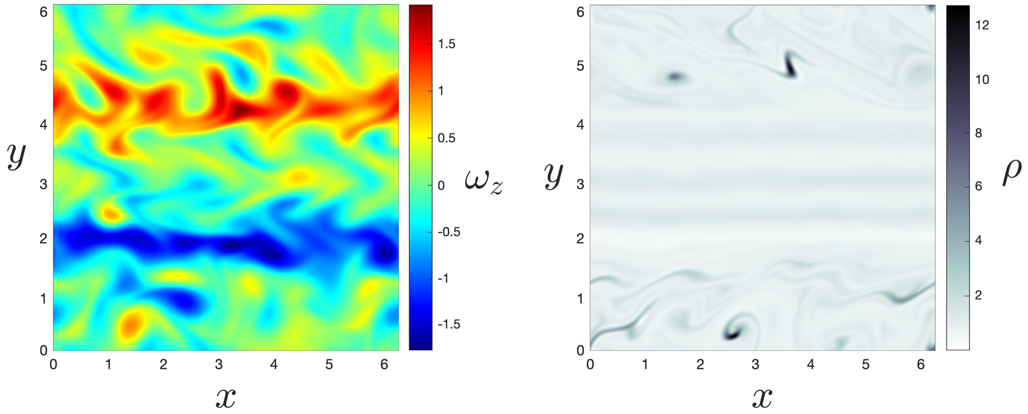


FIG. 4-1. Snapshot of vorticity (left) and bubble density field (right) for a bimodal turbulent jet for $Re = 1000$, and bubble parameters $\epsilon = 0.05$ and $R = 2$.

4.3 Training of the closures

In this case our averaging operator was chosen to be local in both dimensions, x and y:

$$\bar{f}(x, y) = \frac{1}{2\pi} \iint_{S^2} w_{l_x}(y' - y) w_{l_y}(y' - y) f(x, y') \mathrm{d}x \mathrm{d}y', \tag{4.21}$$

where $l_x = l_y = \frac{2\pi}{12}$ are the averaging windows in the x and y directions respectively. As objective functions we used equations (4.15) and (4.16). The spatial and temporal neighborhood used in the closures are chosen as:

$$\alpha(\mathbf{y}) = \{\mathbf{y} + m_1 \delta\mathbf{x} + m_2 \delta\mathbf{y}\}, \;\; \delta x = 2\pi/48, \delta y = 2\pi/48 \;\; \text{and} \;\; m_1, m_2 = -2, -1, ..., 2,$$

$$\chi(t) = \{t - m\delta\tau\}, \delta\tau = 1/100,\ m = 1, 2, ..., 12.$$

Similarly with the one-dimensional closures, these numbers are based on the fact that further increase did not result significant difference in the training and validation errors. We employ the same neural network architectures that we used in the previous section.

Feature selection

For the closure term $\mathbb{D}_{\mathbf{u}}$ (corresponding to the fluid flow) we try as possible flow features the quantities $\overline{\boldsymbol{u}}, \partial_t\overline{\boldsymbol{u}}, \mathbb{D}_{\mathbf{u}}$. Results are shown for the case where the constraint is adopted (cTCN and cLSTM) in Table 3 in terms of training and validation errors. In this case, we observe that the single most important feature is the history of the Reynolds stresses. Furthermore, we see that the optimal combination consists of all the examined features.

Table 4.1. Feature selection for closure of Navier-Stokes.

ξ feature selection					
		cTCN		cLSTM	
Features	Dimension	Train-MSE	Val-MSE	Train-MSE	Val-MSE
$\overline{\boldsymbol{u}} = (\overline{u_1}, \overline{u_2})$	2	0.235	0.521	0.258	0.572
$\partial_t\overline{\boldsymbol{u}} = (\partial_t\overline{u_1}, \partial_t\overline{u_2})$	2	0.098	0.480	0.112	0.388
$\mathbb{D}_{\mathbf{u}}$	2	0.081	0.094	0.092	0.114
$\overline{\boldsymbol{u}}, \partial_t\overline{\boldsymbol{u}}$	4	0.069	0.485	0.100	0.522
$\overline{\boldsymbol{u}}, \mathbb{D}_{\mathbf{u}}$	4	0.048	0.082	0.067	0.094
$\partial_t\overline{\boldsymbol{u}}, \mathbb{D}_{\mathbf{u}}$	4	0.027	0.048	0.034	0.063
$\overline{\boldsymbol{u}}, \partial_t\overline{\boldsymbol{u}}, \mathbb{D}_{\mathbf{u}}$	6	0.020	0.039	0.027	0.055

For the closure of the transport equation we carry out the same process in Table 4. We observe that the single most important features seems to be $\overline{\rho}$, similarly with the one-dimensional case. For the results that follow we choose the combination $\overline{\mathbf{v}}, \overline{\rho}, \partial_t \overline{\mathbf{v}}, \partial_t \overline{\rho}, \mathbb{D}_\rho$, which results in the minimum validation and testing errors.

Table 4.2. Feature selection for closure of bubble transport equation.

ζ feature selection					
		cTCN		cLSTM	
Features	Dimension	Train-MSE	Val-MSE	Train-MSE	Val-MSE
$\overline{\rho}$	1	0.199	0.398	0.228	0.451
$\overline{\boldsymbol{v}} = (\overline{v_1}, \overline{v_2})$	2	0.320	0.591	0.318	0.515
$\overline{\boldsymbol{v}}, \overline{\rho}$	3	0.141	0.386	0.162	0.404
$\overline{\boldsymbol{v}}, \overline{\rho}, \partial_t \overline{\boldsymbol{v}}$	5	0.085	0.176	0.087	0.192
$\overline{\boldsymbol{v}}, \overline{\rho}, \partial_t \overline{\rho}, \mathbb{D}_\rho$	5	0.051	0.091	0.061	0.125
$\overline{\boldsymbol{v}}, \partial_t \overline{\rho}, \partial_t \overline{\boldsymbol{v}}, \mathbb{D}_\rho$	6	0.030	0.049	0.038	0.068
$\overline{\boldsymbol{v}}, \overline{\rho}, \partial_t \overline{\rho}, \partial_t \overline{\boldsymbol{v}}, \mathbb{D}_\rho$	7	0.016	0.032	0.031	0.047

4.4 Validation and generalizability

Here we aim to showcase the application of our method on two-dimensional coarse-scale closures. We consider two cases: (i) we train on *unimodal* jets for flows with Reynolds number $\mathrm{Re} \in \{650, 750, 850\}$ and test on *unimodal* jets in the range $\mathrm{Re} \in [500, 1000]$ and not included in the training set; and (ii) we once again train on *unimodal* jets (same Reynolds as before) and test on *bimodal* jets in the same Reynolds range as in case (i). For the coarse-scale model we employ a resolution of 48×48, complemented with the ML-closure terms. We compare the energy spectra at the statistical steady state of the flows between the coarse-scale predictions and the two-dimensional reference solutions, i.e. for $t \in [200, 600]$.

4.4.1 Testing on a unimodal jet

In Fig. 4-2 we present the space-time-averaged mean-square error between the $x - y$ locally averaged DNS flow field [using Eq. 4.21], $\bar{u}^*$, and the coarse scale model, $\bar{u}$:

$$||\bar{u}^* - \bar{u}||_2^2 = \frac{1}{(2\pi)^2 T} \int_0^{2\pi} \int_0^{2\pi} \int_{t_0}^{t_0+T} (\bar{u}^*(x,y,t) - \bar{u}(x,y,t))^2 \mathrm{d}x\mathrm{d}y\mathrm{d}t. \qquad (4.22)$$

The results are in full consistency with the one-dimensional closures, i.e. cTCN has

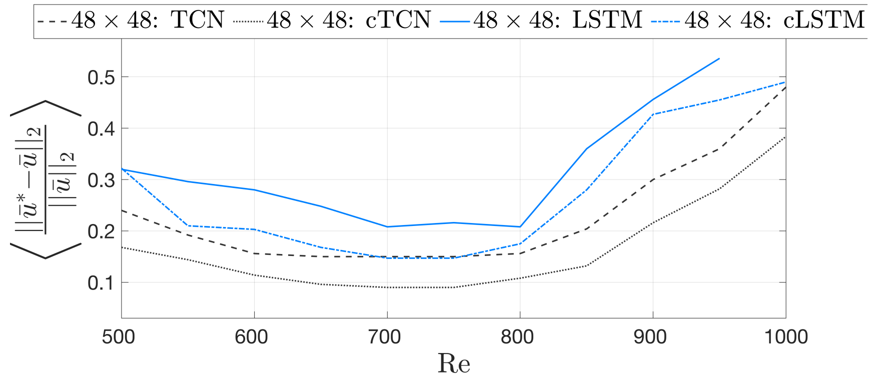


FIG. 4-2. Normalized mean-square error Eq. 4.22 of each two-dimensional closure model using TCN, LSTM and their constrained versions for unimodal jets. Training includes data for unimodal jets with Re $\in \{650, 750, 850\}$.

the best performance. We also present a detailed comparison for Re = 800 between the coarse model and the DNS in terms of the energy spectrum and mean profile the flow (Fig. 4-3). The energy spectrum is computed by obtaining the spatial Fourier transform at each time instant and then considering the variance of each Fourier coefficient over time. We plot the energy spectrum in terms of the absolute wavenumber values. For both the flow field and bubble field the coarse-model is able to accurately capture the mean profiles, and the large scale features of the spectrum.

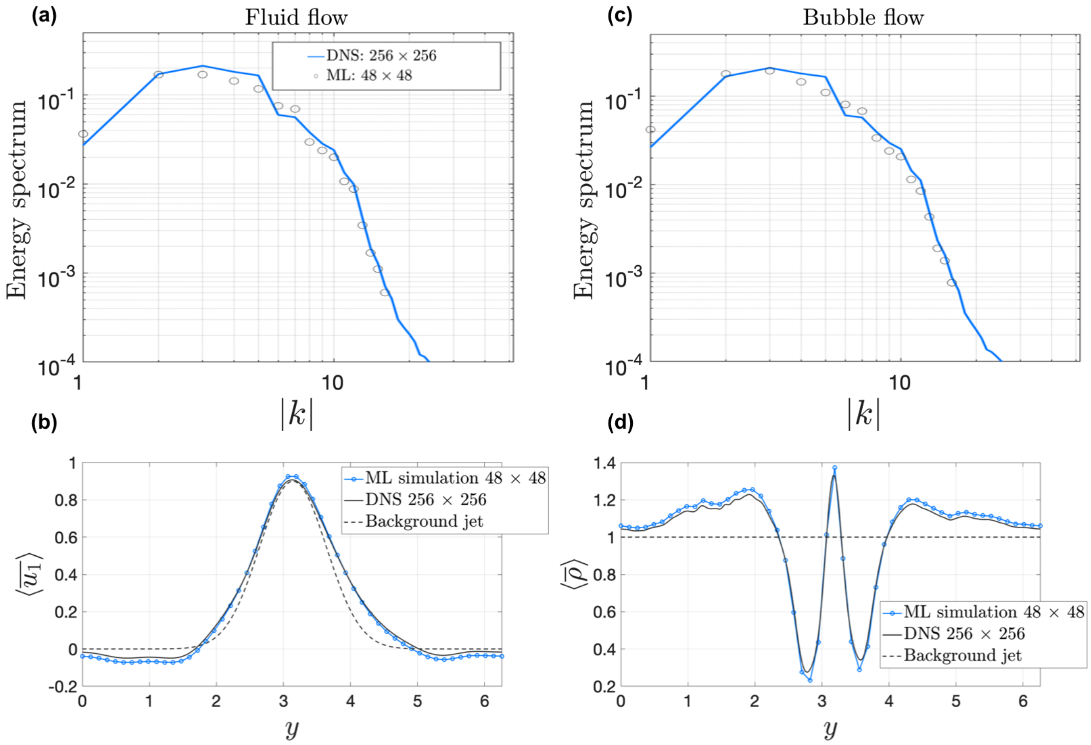


FIG. 4-3. Comparison of energy spectra and mean profiles for the flow velocity field [(a), (c)] and the bubble velocity field [(b), (d)] for unimodal jets. Blue lines correspond to DNS simulations on a 256 × 256 grid and the black circles correspond to the coarse model using two-dimensional closures (cTCN) on a 48 × 48 grid for Re = 800.

4.4.2 Testing generalizability on bimodal-jets

We proceed to test the generalizability of the two-dimensional closure schemes on bimodal jets. The setup is identical with the one adopted for one-dimensional closures. In Fig. 4-4 we compare the normalized mean-square error between the locally averaged DNS solution and the one obtained form the coarse model. Consistent with previous results, the cTCN has the best performance. It is interesting to note that the performance is even better in the Reynolds regime outside the training data, i.e. for Re > 850.

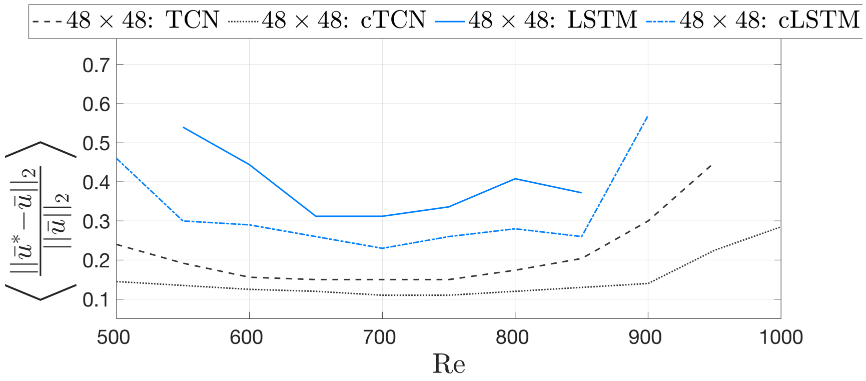


FIG. 4-4. Normalized mean-square error Eq. 4.22 for two-dimensional coarse models applied on bimodal jet flows. Training used data from unimodal flows with Re = $\{650, 750, 850\}$.

The energy spectrum of the fluid velocity and the bubble velocity field, as well as the corresponding mean profiles are compared with DNSs for Re = 800 in Fig. 4-5. In this case, we note that while there is good agreement between the mean profiles, there is some discrepancy between the approximate and exact spectra. To understand better the source of this discrepancy we plot the energy spectrum of the flow in the k_x, k_y space (Fig. 4-6). As can be seen the coarse model overestimates the spread of the energy of the fluctuations only in the $x-$direction, which is consistent with the fact that the mean $y-$profile of the flow is accurately modeled. This is not surprising given that the developed closures in this paper are designed to capture well the mean flow characteristics and not necessarily the energy spectrum. A closure approach based on second-order statistical equations (see e.g. Ref. [150]) is beyond the scope of this paper and will be considered elsewhere.

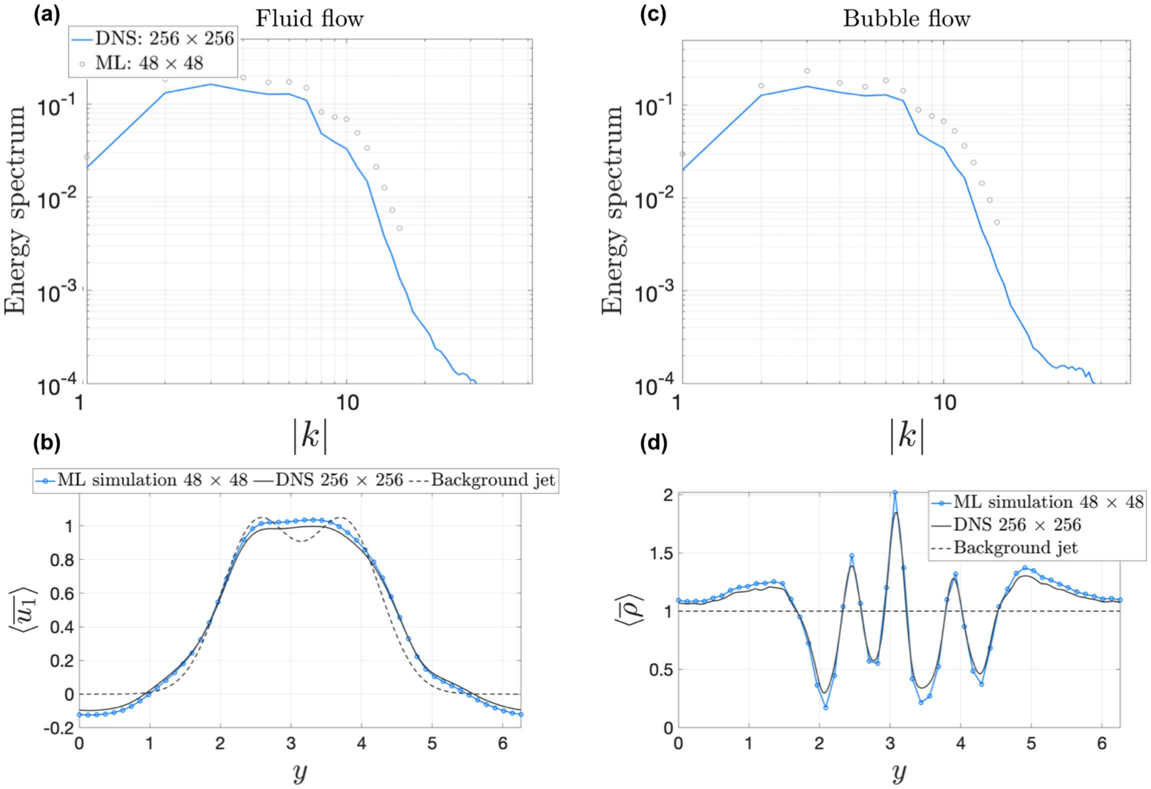


FIG. 4-5. Comparison of energy spectra and mean profiles for the fluid velocity field [(a), (c)] and the bubble velocity field [(b), (d)] for the case of a bimodal jet with Re = 800. Blue lines correspond to DNSs on a 256 × 256 grid and the black circles correspond to a coarse-model with two-dimensional cTCN closures on a 48 × 48 grid.

The overall improvement in the two-dimensional predictions due to the adoption of the physical constraint is summarized in Table 4.3, where we show the improvement of the mean-square error for the mean flow, averaged over different Reynolds numbers. We note that for the TCN architecture the improvement is more pronounced, close to 30%, and it is also robust for the case of a bimodal jet. A quick comparison with the error-decrease values in Table A.3 shows the L^2-error improvement is slightly better in the 1D (zonally-averaged) coarse-scale model than in the 2D coarse-scale model. This result could be attributed to the fact that a 1D model will be less turbulent compared to a 2D coarse-scale simulation, due to zonally-averaging the flow and thus essentially neglecting perturbations along the x-axis, making the simulations more stable.

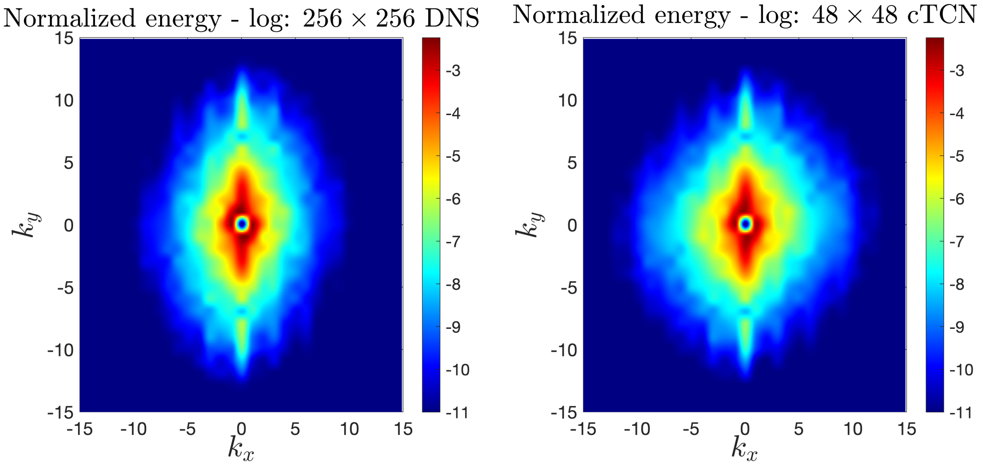


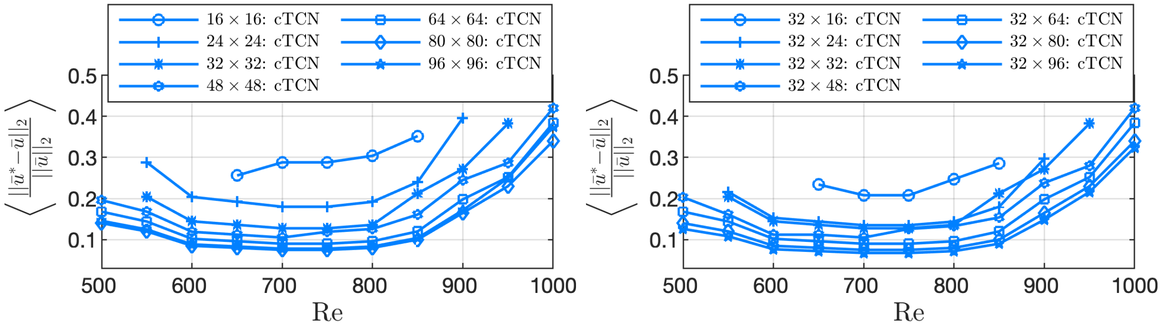
FIG. 4-6. Energy spectrum of the fluid flow for the bimodal jet for Re = 800. Comparison between the DNS (left) and the coarse-model based on 2D cTCN closures (right).

Table 4.3. Error decrease (Reynolds-averaged) due to the physical constraint for two-dimensional closure schemes.

Architecture	Jet-type	Error decrease
TCN-2D	Unimodal	30%
LSTM-2D	Unimodal	20%
TCN-2D	Bimodal	33%
LSTM-2D	Bimodal	31%

4.4.3 Dependence on the coarse-model grid-size

Finally, we showcase a numerical investigation for the relationship between the chosen grid-size for the coarse-scale simulations and the mean-square error of the velocity of the fluid flow. For all results presented below, training data was chosen as previously (unimodal jets) and results are presented for bimodal jets. In Fig. 4-7(a), we vary the size of a $N_x \times N_y$ grid with $N_x = N_y \in \{16, 24, 32, 40, 48, 64, 80, 96\}$. We notice that there is significant improvement in our predictions as we refine the grid up to a grid-size of 64×64 where the error saturates.



(a)

(b)

FIG. 4-7. Root-mean-square error for different grid-sizes using two-dimensional cTCN closures on bimodal jet.

Since the variation of the mean profile of the flow is only along the y-direction, having a coarser resolution along the less significant x-direction should not hinder the predictions. To validate this property we maintain a constant discretization in the x-direction and vary the grid-size only in the $y-$ direction. Results are demonstrated in Fig. 4-7(b), showing clearly that by having a fine resolution only in the $y-$direction is sufficient to achieve comparable performance with the fine resolution case in both directions: $N_x = N_y = 96$.

4.4.4 Limitations

While the proposed method adequately predicts flows that are distinct from those of the training set, it is important to state that the computed closure is not expected to be as effective in every fluid flow. This is due to the fact that depending on the specifics of a fluid flow, such as dimensionality, boundary conditions, domain geometry, excitation terms, and the presence of additional dynamics such as Coriolis terms, the nonlinear interactions between scales are different. Therefore, aiming to machine learn universal closures that will work for every case is beyond the scope of this thesis. Instead, our approach is to employ data from flows that have some common features with the flow we are interested in modeling, and combine this data with a universal constraint, the energy conservation by the nonlinear terms, to increase the accuracy of the computed closures. The optimal choice of input features is also expected to vary depending on the specifics of the flow and therefore a numerical examination of different combinations should be performed to achieve the most effective closure.

Chapter 5

Energy preserving closure-schemes for second-order statistics

Abstract

This chapter, based on [27], presents a data-driven, energy-conserving closure method for the coarse-scale evolution of the mean and covariance of turbulent systems. Spatio-temporally non-local neural networks are employed for calculating the impact of non-Gaussian effects to the low-order statistics of dynamical systems with an energy-preserving quadratic nonlinearity. This property, which characterizes the advection term of turbulent flows, is encoded via an appropriate physical constraint in the training process of the data-informed closure. This condition is essential for the stability and accuracy of the simulations as it appropriately captures the energy transfers between unstable and stable modes of the system. The numerical scheme is implemented for a variety of turbulent systems, with prominent forward and inverse energy cascades. These problems include prototypical models such as an unstable triad-system and the Lorentz-96 system, as well as more complex models: the 2-layer quasi-geostrophic flows and incompressible, anisotropic jets where passive inertial tracers are being advected on. Training data are obtained through high-fidelity direct numerical simulations. In all cases, the hybrid scheme displays its ability to accurately capture the energy spectrum and high-order statistics of the systems under discussion. The generalizability properties of the trained closure models in all the test cases are explored, using out of sample realizations of the systems.

5.1 Problem formulation

The methodology of this chapter follows that presented in section 3.1. To synopsize the results here, we assume that the evolution equation of a quantity of interest $\boldsymbol{u}$ has the following general form

$$\frac{d\boldsymbol{u}}{dt} = A\boldsymbol{u} + \boldsymbol{B}\left(\boldsymbol{u}, \boldsymbol{u}\right) + \boldsymbol{F}(t) + \dot{W}_k(t;\omega)\boldsymbol{\sigma}_k(t), \tag{5.1}$$

where A is a linear operator, $\boldsymbol{F}$ denotes a deterministic external forcing and $\dot{W}_k\boldsymbol{\sigma}_k$ corresponds to a stochastic forcing with white noise characteristics. The operator $\boldsymbol{B}$ is assumed to be quadratic and energy-preserving, i.e.

$$\boldsymbol{B}(\boldsymbol{u}, \boldsymbol{u}) \cdot \boldsymbol{u} = 0. \tag{5.2}$$

This restrictive definition of $\boldsymbol{B}$ is valid for many important problems in fluid mechanics, retaining the physical relevance of the formulation. For example, $\boldsymbol{B}$ can be viewed as the advection term of turbulent flows, a class of problems that has historically attracted the attention of reduced-order modelling literature.

The quantity of interest $\boldsymbol{u}$ is analyzed using the finite-dimensional expansion

$$\boldsymbol{u} = \overline{\boldsymbol{u}} + \boldsymbol{u}' = \overline{\boldsymbol{u}} + \sum_{i=1}^{N} Z_i(t;\omega)\boldsymbol{v}_i, \tag{5.3}$$

where $\boldsymbol{v}_i$ form a prescribed orthonormal basis, while Z_i are zero-mean, time-dependent random functions. The symbol ω denotes the random argument and the mean field $\overline{\boldsymbol{u}}$ can be interpreted as an ensemble average, $\mathbb{E}\left[\boldsymbol{u}\right] = \overline{\boldsymbol{u}}$, with respect to ω. Using the above representation, a mean state evolution equation can be written as

$$\frac{d\overline{\boldsymbol{u}}}{dt} = [L + D]\overline{\boldsymbol{u}} + \boldsymbol{B}(\overline{\boldsymbol{u}}, \overline{\boldsymbol{u}}) + \sum_{i=1}^{N}\sum_{j=1}^{N} \overline{Z_i Z_j} \boldsymbol{B}(\boldsymbol{v}_i, \boldsymbol{v}_j) + \boldsymbol{F}. \tag{5.4}$$

For a second-order statistical framework, an evolution equation for the covariance matrix $R_{ij} = \overline{Z_i Z_j}$ can then be written as

$$\frac{d\mathbf{R}}{dt} = \mathcal{L}\mathbf{R} + \mathbf{R}\mathcal{L}^* + \mathcal{Q}_\sigma + \mathcal{Q}, \tag{5.5}$$

where $\mathcal{L}$ is a linear operator expressing dissipation and energy transfers between the mean field and the stochastic modes

$$\mathcal{L}_{ij} = [(L + D)\,\boldsymbol{v}_j + \boldsymbol{B}(\overline{\boldsymbol{u}}, \boldsymbol{v}_j) + \boldsymbol{B}(\boldsymbol{v}_j, \overline{\boldsymbol{u}})] \cdot \boldsymbol{v}_i, \tag{5.6}$$

with $\mathcal{L}^*$ being its transpose. The operator $(\mathcal{Q}_\sigma)_{ij} = (\boldsymbol{\sigma}_k \cdot \boldsymbol{v}_j)\,(\boldsymbol{\sigma}_k \cdot \boldsymbol{v}_i)$ models effects due to the stochastic external forcing and $\mathcal{Q}$ corresponds to third-order effects that express energy fluxes between different stochastic modes

$$\begin{aligned} \mathcal{Q}_{mn} = &\sum_{i=1}^{N}\sum_{j=1}^{N} \overline{Z_i Z_j Z_m} \boldsymbol{B}(\boldsymbol{v}_i, \boldsymbol{v}_j) \cdot \boldsymbol{v}_n \\ &+ \sum_{i=1}^{N}\sum_{j=1}^{N} \overline{Z_i Z_j Z_n} \boldsymbol{B}(\boldsymbol{v}_i, \boldsymbol{v}_j) \cdot \boldsymbol{v}_m. \end{aligned} \tag{5.7}$$

In more detail, $\mathcal{L}$ includes the effects of the linear operators on each mode, as well as energy transfers between the mean and each mode via the energy-preserving nonlinear operator. All these effects can be studied and understood under the framework of second-order statistics. It is the mode-to-mode nonlinear energy transfers that require knowledge of higher moments for their estimation. In order to close the system for the covariance and the mean, a model for the third-order terms $\mathcal{Q}$ is required. To this end, neural networks are utilized to parameterize them. The architecture and constraints used are presented in the following subsections.

5.1.1 Physical constraints related to nonlinear energy transfers

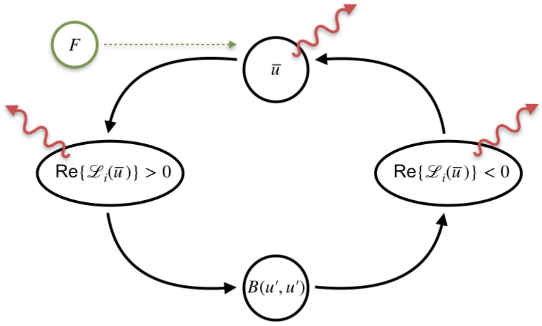
Before describing the details of the data-driven model for $\mathcal{Q}$, we note that an important feature of the presented closure scheme is the requirement to satisfy certain physical principles, which characterize turbulent systems. More specifically, modes that carry small energy or variance can still have important, dynamical effects on the large variance modes through nonlinear energy transfers. A schematic of this property can be seen in Fig. 5-1. The external force excites the mean, which then through the linear part of the covariance equation $\mathcal{L}$ transfers part of this energy to the unstable modes. Since, these modes are unstable, the linear operator $\mathcal{L}$ can only increase them in amplitude. Hence, the only way for the system to reach a statistical steady state is due to i) dissipation, and ii) the nonlinear terms $\mathcal{Q}$ transfer energy from the unstable modes to the stable ones. Therefore, it is critical to accurately represent the third-order closure terms. Using the fact that the quadratic term is energy-preserving by construction, eq. (5.2) holds for $\boldsymbol{u}'$ and thus, one gets $\boldsymbol{B}\left(\boldsymbol{u}', \boldsymbol{u}'\right) \cdot \boldsymbol{u}' = 0$. Taking the expectation of this relations and expanding it using eq. (5.3) one gets

$$\sum_{i=1}^{N} \sum_{m=1}^{N} \sum_{n=1}^{N} \overline{Z_m Z_n Z_i} \boldsymbol{B}\left(\boldsymbol{v}_m, \boldsymbol{v}_n\right) \cdot \boldsymbol{v}_i = 0, \tag{5.8}$$

or in terms of the closure term[150]

$$\mathrm{Tr}[\mathcal{Q}] = 0. \tag{5.9}$$

This constraint is enforced during training to improve the numerical stability of the predictions. It is emphasized that correctly capturing this constraint is necessary for the purpose of correctly modeling the energy exchanges between stable and unstable modes and thus capturing the correct statistical steady steady of the system under discussion.



F
$\overline{u}$
Re$\{\mathscr{L}_i(\overline{u})\} > 0$
Re$\{\mathscr{L}_i(\overline{u})\} < 0$
$B(u', u')$

FIG. 5-1. Energy flow in the systems under consideration.

5.1.2 Data-driven parametrization of the closure terms

While the dynamical systems under study are Markovian and spatially-local, i.e. the evolution of **u** in a specific location and time instant depends only on the current time instant and the current neighborhood, this is not the case for the reduced-order averaged version of these equations. In particular, for the evolution of the mean and covariance equations one typically does not have access to the full-state information required to fully describe the evolution of the system (in this case third-order moments).

As has been showcased by recently proposed data-driven schemes for dynamical systems[178, 180], Takens embedding theorem can be used to enhance the accuracy of predictions [168]. The theorem states that even if only a limited number of the state variables of a system are observed, in principle, one can still obtain the attractor of the full system by using delay embeddings of the observed state variables. To this end, the closure terms are parametrized with *non-local in time* (but still causal) representations, based on Temporal Convolutional Networks [123] (TCN) and Long Short-Term Memory networks[64] (LSTM). These ML architectures have been employed successfully in previous work [29] on computing turbulent closures just for the mean equation (5.4).

In terms of spatial information, the entire mean field and the covariance are utilized as input for the data-informed closure scheme. As a result, the closure terms are modeled in the following form:

$$\begin{aligned}\mathbb{D}_{n,m}\left[\Theta;\overline{\xi}[\chi(t)]\right] &= \sum_{i=1}^{N}\sum_{j=1}^{N}\overline{Z_iZ_jZ_m}\boldsymbol{B}(\boldsymbol{v}_i,\boldsymbol{v}_j)\cdot\boldsymbol{v}_n \\ &+\sum_{i=1}^{N}\sum_{j=1}^{N}\overline{Z_iZ_jZ_n}\boldsymbol{B}(\boldsymbol{v}_i,\boldsymbol{v}_j)\cdot\boldsymbol{v}_m,\end{aligned} \tag{5.10}$$

where $\overline{\xi} = \{\overline{\boldsymbol{u}}, \boldsymbol{R}\}$ are 2nd order statistics of the system, and $\chi(t)$ denotes the history of the statistics state backwards from time t, i.e. $\chi(t) = \{t, t-\tau_1, ..., t-\tau_2, ..., t-\tau_N\}$. The entire mean field $\overline{\boldsymbol{u}}$ and covariance $\boldsymbol{R}$ are used as input for the neural network. The vector Θ denotes the hyperparameters of the neural network while the temporal neighborhood, $\chi(t)$, is selected such that if further increased, the training error does not significantly reduce any more. The number of points in time that have to be considered depends on the temporal discretization of the domain.

5.1.3 Objective function for training

In terms of the training process itself, the input and output data are normalized as typically suggested [152]. The loss function for this problem is chosen to be the single-step prediction mean square error superimposed with the physical constraint. This can be formulated as

$$L(\Theta) = \frac{1}{T}\int_T \sum_n \sum_m \left|\left|\mathbb{D}_{n,m}\left[\Theta;\bar{\xi}\right] - \mathcal{Q}_{n,m}\right|\right|^2 dt + \lambda \int_T \mathrm{Tr}[\mathcal{Q}]dt, \tag{5.11}$$

where λ is a weight that measures the relative importance between the data and the physical constraint, typically chosen to be 1.

5.2 Applications

5.2.1 Triad system

The first application involves a three-dimensional dynamical system that consists of 3 Langevin equations coupled via quadratic and energy-preserving nonlinearities [106]. This triad system acts as a simple surrogate model for barotropic instability. It can be viewed as the result of projection of the fluid equations to three modes, one corresponding to the mean flow and the other two corresponding to wave perturbations. Under these assumptions, the system has the form

$$\begin{aligned}\begin{bmatrix} \mathrm{d}u_1 \\ \mathrm{d}u_2 \\ \mathrm{d}u_3 \end{bmatrix} &= \underbrace{\begin{bmatrix} -\gamma_1 & 0 & 0 \\ 0 & -\gamma_2 & 0 \\ 0 & 0 & -\gamma_3 \end{bmatrix}}_{D} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \mathrm{d}t \\ &+ \underbrace{\begin{bmatrix} 0 & \lambda_{12} & \lambda_{13} \\ -\lambda_{12} & 0 & \lambda_{23} \\ -\lambda_{13} & -\lambda_{23} & 0 \end{bmatrix}}_{L} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \mathrm{d}t + \underbrace{\begin{bmatrix} \beta_1 u_2 u_3 \\ \beta_2 u_1 u_3 \\ \beta_3 u_1 u_2 \end{bmatrix}}_{B(\mathbf{u},\mathbf{u})} \mathrm{d}t \\ &+ \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \mathrm{d}t + \begin{bmatrix} \sigma_1 \mathrm{d}W_1 \\ \sigma_2 \mathrm{d}W_2 \\ \sigma_3 \mathrm{d}W_3 \end{bmatrix}, \end{aligned} \tag{5.12}$$

where D and L are respectively the symmetric and skew-symmetric component of the linear operator, F_1, F_2, F_3 correspond to deterministic external excitation and

$\mathrm{d}W_1, \mathrm{d}W_2, \mathrm{d}W_3$ are independent white-noise processes. It is also assumed that

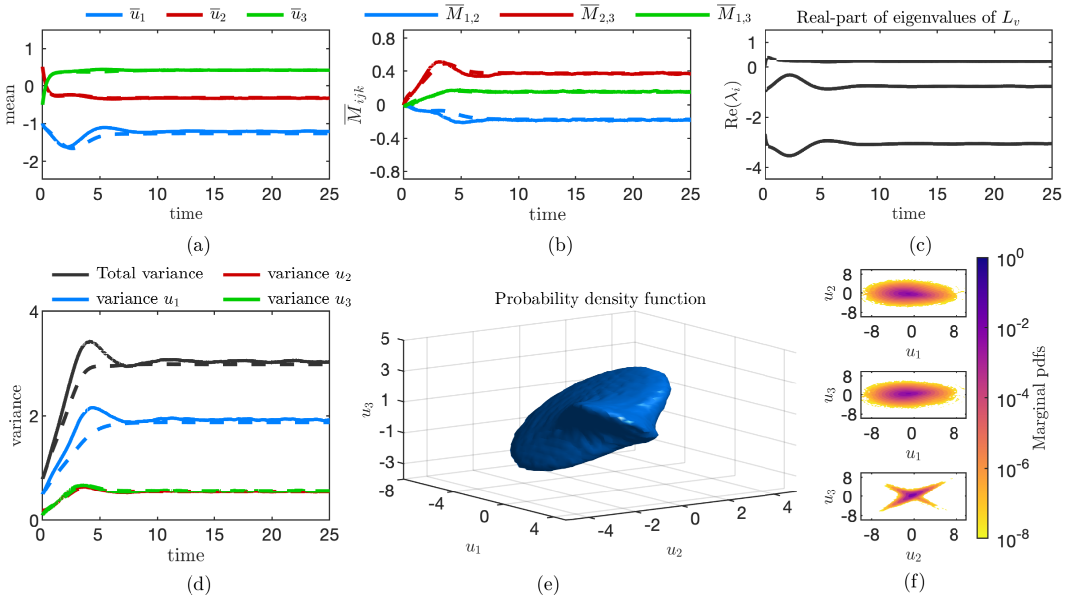
$$\beta_1 + \beta_2 + \beta_3 = 0, \tag{5.13}$$

to ensure that the quadratic term is energy-preserving. Interaction of this reduced-order model with other modes of the full system is modeled via the white-noise terms and linear dissipation, a standard practice in many stochastic climate models [88, 60, 36].

The goal of this section is to examine the triad system as a prototypical model for turbulent anisotropic flows. A property of such flows is that the mean exhibits persistent instabilities along certain directions in phase space. Under the assumption that u_1 corresponds to the mean-flow variable, this instability can be added to the triad system via a negative γ_1 value. Linear dissipation is set as $\gamma_1 = -0.4$ and $\gamma_2 = \gamma_3 = 2$, following a previously presented setup [149]. The skew-symmetric part is set as $\lambda_{12} = 0.03, \lambda_{13} = 0.06$ and $\lambda_{23} = -0.09$. The coefficients of the nonlinear term are set as $\beta_1 = 2$ and $\beta_2 = \beta_3 = -1$. For the first test-case constant deterministic external forcing is tested. To ensure that the perturbation variables u_2, u_3 are energetic (and thus remove energy from the mean), the deterministic forcing is set to $F_1 = 0, F_2 = -1$ and $F_3 = 1$. Finally, the white noise amplitudes are tuned so as to ensure the system achieves a statistical steady state. For this to happen the amplitudes are chosen as $\sigma_1 = 0.25$ and $\sigma_2 = \sigma_3 = 0.79$. Initial conditions are sampled from the distribution

$$\begin{bmatrix} u_1(0) \\ u_2(0) \\ u_3(0) \end{bmatrix} \sim \begin{bmatrix} \mathcal{N}(1, 0.5) \\ \mathcal{N}(0.5, 0.2) \\ \mathcal{N}(-0.5, 0.1) \end{bmatrix}. \tag{5.14}$$

For reference data, 10^5 samples of this system are computed (each time using a new sample of both the white noise forcing and the initial conditions). Reference statistics of this dynamical systems are obtained from this MC simulation. All simulations are carried out with time-step $\mathrm{d}t = 0.01$ from $t = 0$ till $t = 25$. The system is integrated in time using the forward Euler method. With respect to the reduced-order model, a single-layer LSTM neural network is used during training.



(a)

(b)

(c)

(d)

(e)

(f)

FIG. 5-2. Comparison between MC (solid line) and reduced-order (dashed-line) results for triad system with constant forcing. (a) Evolution of mean of the system; (b) Evolution of off-diagonal components of covariance matrix; (c) Evolution of real part of eigenvalues of L_v; (d) Evolution of total system variance and variance of each mode; (e) Contour of the steady-state system pdf for $u|_{f(u)=10^{-5}}$; (f) Marginal pdfs of the system at steady-state.

The statistical properties of the system as well as a comparison with the low-order statistical predictions of the proposed data-informed method are depicted in figure 5-2. Subplots (a) and (d) establish that u_1 is indeed the most energetic mode of the system and that the system achieves a statistical steady state. Furthermore, as seen from the marginal on the $u_2 - u_3$ plane on subplot (f), the system exhibits strong anisotropic behaviour in this plane. The reduced-order model is able to accurately predict the statistical equilibrium of the system. In addition, it adequately captures the intermittent dynamical evolution of the mean and variance of the system. The good agreement with the MC data is observed despite the fact that the pdf of the system is strongly non-Gaussian in the $u_3 - u_2$ space and the mean variable u_1 is by construction linearly unstable.

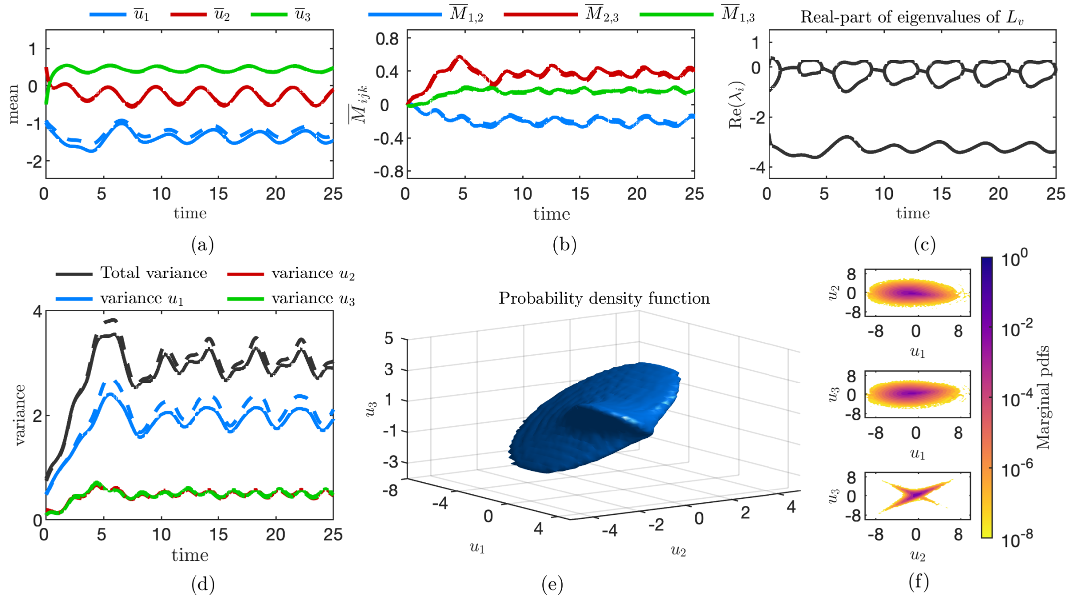


FIG. 5-3. Comparison between MC (solid line) and reduced-order (dashed-line) results for triad system with periodic forcing. (a) Evolution of mean of the system; (b) Evolution of off-diagonal components of covariance matrix; (c) Evolution of real part of eigenvalues of L_v; (d) Evolution of total system variance and variance of each mode; (e) Contour of pdf at $t = 25$ for $u|_{f(u)=10^{-5}}$; (f) Marginal pdfs of the system at $t = 25$.

The next test-case involves periodic forcing while the other parameters of the system remain the same as before. In more detail, the deterministic part of the forcing is now set to

$$F_1 = 0, \quad F_2 = -1 + 0.5 \sin \frac{\pi t}{2}, \quad F_3 = 1 + 0.2 \cos \frac{\pi t}{2}. \tag{5.15}$$

In addition, the parameters of the stochastic forcing are set to

$$\sigma_1 = 0.25 - \frac{1.3^2}{10} \sin^2 \frac{\pi t}{2}, \quad \sigma_2 = \sigma_3 = 0.79 - \frac{2}{5} 0.79 \cdot 1.3^2 \sin^2 \frac{\pi t}{2}. \tag{5.16}$$

MC results for the system as well as a comparison with the results of the data-informed closure scheme (trained with the previous set of forcing parameters) are shown in figure 5-3(a, d). Again, despite the strongly non-Gaussian nature of the problem (fig. 5-3 (b, e, f)) the closure scheme is able to accurately predict the first- and second-order statistics throughout the duration of the simulation.

5.2.2 L96 model

As a next step, a higher-dimensional problem, with a large number of positive Lyapunov exponents, is considered [93]. Specifically, the model under discussion is governed by the equation

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = u_{i-1}(u_{i+1} - u_{i-2}) - u_i + F, \tag{5.17}$$

for $N = 40$ and F deterministic external forcing. This system acts as a prototypical model to study baroclinic instability in midlatitude flows. The dynamics comprise of an energy-conserving quadratic nonlinearity, a linear dissipation term and external forcing. The forcing term is assumed to be spatially homogeneous, which as a result implies that the system is translationally invariant in space. This property implies that Fourier modes can be used for the basis expansion and that the covariance matrix can be assumed to be diagonal. One can observe that for different magnitudes of forcing, the number of unstable modes changes (from 0 up to 11). Hence, this is model is an excellent case to assess the presented method.

To illustrate numerically the UQ properties of the proposed closure scheme, the model is trained and tested on an aperiodic forcing generated by the Ornstein-Uhlenbeck process

$$dF = -\frac{1}{\tau_F}(F - \overline{F})dt + \sigma_F dW, \tag{5.18}$$

where $\overline{F}$ is the mean value around which the process oscillates and dW models white noise. Here we chose as modes v_n the Fourier modes. Also, we do not perform any covariance reduction, i.e. we model the full covariance matrix.

Training occurs with a single stochastic forcing realization. The reference statistics are derived by averaging over a distribution of initial conditions. To derive the required training data 10^4 direct numerical simulations are averaged. These statistics allow for the calculation of the time-evolution of $\mathbb{E}[u]$ and $\mathbb{E}[Z_n^2] = \mathbb{E}\{[(u - \mathbb{E}[u]) \cdot v_n]^2\}$. The neural network is then trained using the derived data and under the constraint of Eq. (5.9), which is equivalent to the constrain that the quadratic operator should be energy conserving. A single-layer LSTM neural network with 150-timesteps as time-history is used. Testing of the closure scheme is carried out for 4 out-of-sample random realizations of the forcing. Only 10 time-steps are used as initial input for the neural network. Results are presented in Figure 5-4. For all realizations, we have excellent agreement between the spectrum predicted from the data-driven closure and the MC simulation. This is the case despite the strongly transient nature of the excitation that pushes the system away from its statistical steady state.

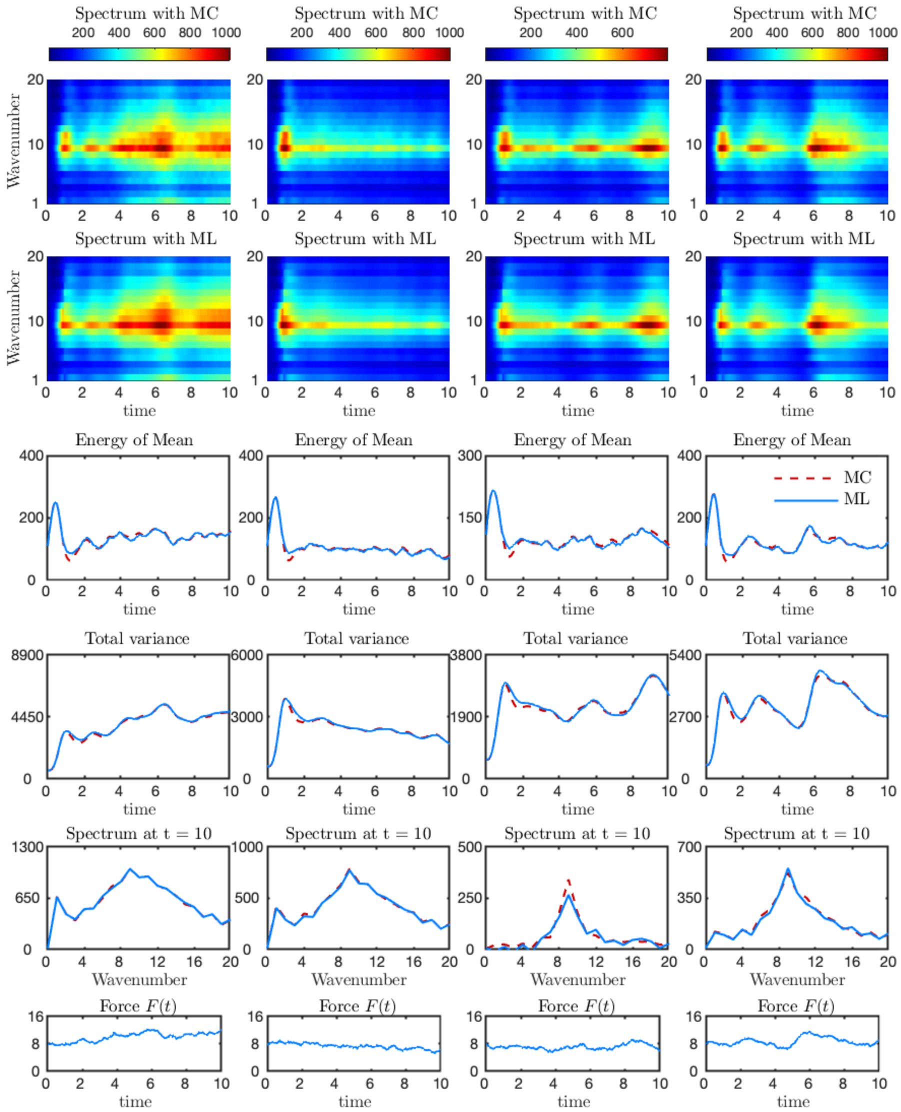


FIG. 5-4. Comparison of ML uncertainty quantification scheme with exact statistics produced by the Monte-Carlo method. Results are shown for different dynamical regimes of the aperiodic forcing parameter F generated as an Ornstein-Uhlenbeck process. The colorplots present the evolution of the exact and approximated spectrum. We also present the energy of the mean and the trace of the covariance over time. At the last row we show the steady state spectrum (exact and approximate).

5.2.3 Multiphase flow

The next test case is an anisotropic multiphase flow setup. It involves the advection of bubbles, which are assumed to be passive inertial tracers, over an incompressible fluid flow. The fluid flow is governed by the Navier-Stokes equations in dimensionless form:

$$\frac{D\boldsymbol{u}}{Dt} = -\boldsymbol{\nabla} p + \frac{1}{\mathrm{Re}}\Delta\boldsymbol{u} + \nu\Delta^{-2}\boldsymbol{u} + \boldsymbol{F}, \tag{5.19}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0, \tag{5.20}$$

where $\boldsymbol{u}$ is the velocity field of the fluid, p its pressure, Re is the Reynolds number of the flow, D/Dt is the material derivative operator and $\boldsymbol{F}$ denotes an external forcing term. Parameter ν is a hypoviscosity coefficient aiming to remove energy from large scales and maintain the flow in a turbulent regime.

One can follow a similar process for the advection equation governing the motion of small inertial tracers. In particular for small inertia particles their Lagrangian velocity, $\boldsymbol{v}$, is a small perturbation of the underlying fluid velocity field [54]:

$$\boldsymbol{v} = \boldsymbol{u} + \epsilon\left(\frac{3R}{2} - 1\right)\frac{D\boldsymbol{u}}{Dt} + O\left(\epsilon^2\right), \tag{5.21}$$

where

$$\epsilon = \frac{\mathrm{St}}{R} \ll 1, \quad R = \frac{2\rho_f}{\rho_f + 2\rho_p}. \tag{5.22}$$

The parameter ϵ represents the importance of inertial effects, while St is the bubble Stokes number, measuring the ratio between the characteristic timescales of the bubbles over that of the flow. Parameter R is a density ratio with ϱ_p and ϱ_f being the density of bubble and the fluid respectively. By introducing ρ as the concentration of tracers at a particular point, the following transport equation can be derived for the bubble flow:

$$\partial_t \rho + \boldsymbol{\nabla} \cdot (\boldsymbol{v}\rho) = \nu_2 \Delta^4 \rho. \tag{5.23}$$

The right-hand-side of the transport equation represents a hyperviscosity term. The hyperviscosity parameter is tuned so as to remove energy from scales close to the resolution limit of the numerical simulations.

Utilizing the series expansion presented here for the variables of the problem:

$$\boldsymbol{u} = \overline{\boldsymbol{u}} + \sum_k Z_{k,u}, \boldsymbol{v}_{k,u}, \quad \rho = \overline{\rho} + \sum_k Z_{k,\rho} v_{k,\rho}, \tag{5.24}$$

one can develop evolution equations for the mean and the variance for both the fluid flow and the bubble flow. For the particular problem, the data-informed closure-scheme is trained on data from unimodal jet flows (details of the jet configuration have been presented previously[29]). The Reynolds numbers used in training are Re $\in \{650, 750, 850\}$. The reference simulations are carried out on a 256×256 grid with doubly periodic boundary conditions on a $[0, 2\pi]^2$ rectangular domain. All flows are evolved until they reach a statistical equilibrium. Since MC simulations are prohibitively expensive for this problem, time-averaging is used for the derivation of the energy spectra in the statistical equilibrium.

The derived closure is first tested on *unimodal* jets in the range Re $\in [500, 1000]$. For the mean-field model (eq. (5.4)) a coarse resolution of 32×32 is employed. For the covariance of both the flow field and the density, Fourier modes are utilized. Specifically, modes with wavenumber $|k| \leq 48$ are considered in the covariance evolution (eq. (**??**)) that is complemented with the ML closures.

Figure 4-2 presents the space-time-averaged mean-square error between the $x - y$

locally averaged DNS flow field, $\overline{\boldsymbol{u}^*}$, and the coarse scale model, $\overline{\boldsymbol{u}}$:

$$||\overline{\boldsymbol{u}^*} - \overline{\boldsymbol{u}}||_2^2 = \frac{1}{(2\pi)^2 T} \int_0^{2\pi} \int_0^{2\pi} \int_{t_0}^{t_0+T} \left(\overline{\boldsymbol{u}^*} - \overline{\boldsymbol{u}}\right)^2 \mathrm{d}\boldsymbol{x}\mathrm{d}t. \tag{5.25}$$

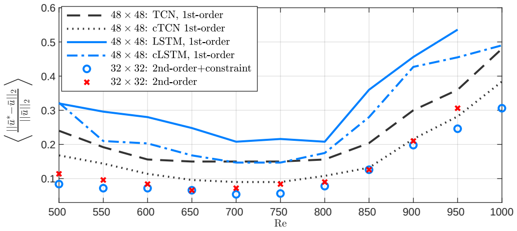


FIG. 5-5. Normalized mean-square error of each two-dimensional closure model using TCN, LSTM and their constrained versions with the energy-conservation (5.4) for unimodal jets. Training includes data for unimodal jets with Re ∈ {650, 750, 850}.

The numerical results of the presented scheme are compared with first-order closures where the ML correction is applied only on the mean equation (5.4) and the covariance is not modeled [29]. For the 1st-order closure both TCN and LSTM architectures were assessed. The constrained versions, cTCN and cLSTM, correspond to closures where, in addition to the L^2-error of the closure terms, the physical constraint related to the energy-preserving property of the advection term is included (eq. (4.15)). As expected the second-order closure scheme outperforms the corresponding first-order schemes, even using a coarser resolution. The inclusion of the constraint significantly improves the accuracy of the predicted statistics for all cases. Furthermore, to showcase the importance of using the constraint derived in eq. (5.9), a version of the 2nd-order closure scheme with $\lambda = 0$ is also presented. As it can be seen, numerical results are improved when including the constraint.

As an additional test-case, the closure scheme is tested on *bimodal* jets with Re ∈ [500, 1000] (details of the bimodal jet configuration have been presented previously[29]).

The closure is trained on the same data set as before, i.e. on *unimodal* jets with Reynolds number Re $\in \{650, 750, 850\}$. Figure 5-6 presents the space-time-averaged mean-square error between the $x - y$ locally averaged DNS flow field. In this case, including the constraint during training not only improves numerical accuracy of the results but allows us to avoid numerical instabilities for Reynolds numbers outside the training data-set.

In Figure 5-7 we compare the energy spectra for the fluid and bubble flows as obtained by DNS and the second order closure scheme. We also include a comparison with the first-order closure obtained in previous work [29]. For both the *unimodal* and *bimodal* jets we have accurate computation of the energy spectra of the fluid flow and the advected bubble dispersion in the statistical equilibrium. Furthermore, the second order model clearly outperforms the corresponding first-order scheme, indicating that incorporating additional physics through the second order equation is leading to better performance and improved stability, albeit the additional computational cost.

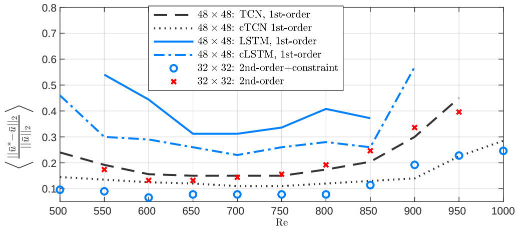
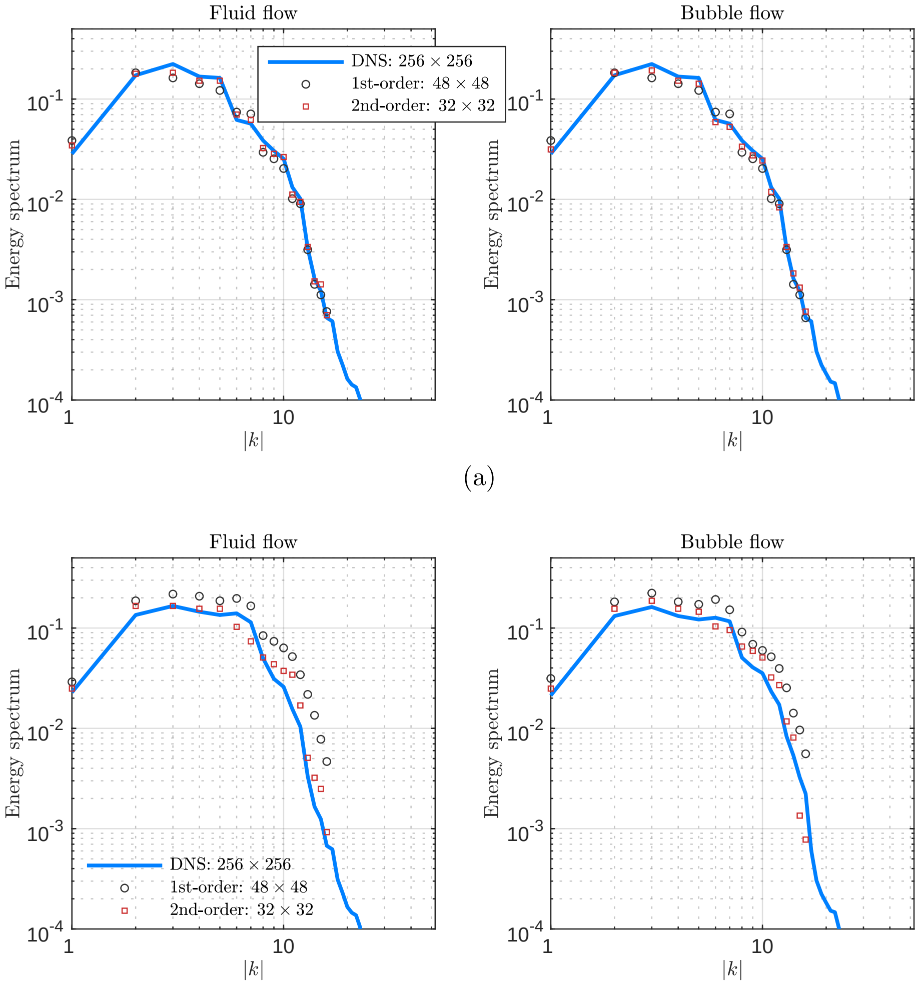


FIG. 5-6. Normalized mean-square error for different closure models and their constrained versions for bimodal jets. Training utilizes data for unimodal jets with Re $\in \{650, 750, 850\}$.



(a)

(b)

FIG. 5-7. Energy spectra for the fluid flow and bubble flow for (a) a unimodal jet with Re = 800 and (b) a bimodal jet with Re = 800.

5.2.4 Application to quasi-geostrophic (QG) flows

The last test-case involves a quasi-geostrophic (QG) model. The quasigeostrophic model stems from shallow-water equations for horizontally dominated (i.e. columnar) motion of homogeneous fluid in each (thin) layer:

$$\frac{Du}{Dt} + f \times u = -g\nabla\eta, \quad \text{and} \quad \frac{\partial h}{\partial t} + \nabla \cdot (uh) = 0, \tag{5.26}$$

where $h = \eta + H$ is the total depth of a single layer, u is the velocity field and f the Coriolis accelearation. The quasigeostrophic equation arises when one considers only nearly geostrophic motions (which we must consider independent of ageostrophic phenomena as well). Under these assumptions one has the simplifications

$$\psi = \frac{g\eta}{f_0}, \quad \text{and} \quad \frac{D}{Dt} = \frac{\partial}{\partial t} - \frac{\partial \psi}{\partial y}\frac{\partial}{\partial x} + \frac{\partial \psi}{\partial x}\frac{\partial}{\partial y}, \tag{5.27}$$

with ψ being the streamfunction describing the horizontal flow. Under these conditions one can derive the quasigeostrophic equation (for a single layer)

$$\frac{\partial q}{\partial t} + J(\psi, q) = 0, \tag{5.28}$$

where q is the potential vorticity defined as

$$q = \nabla^2\psi + f - \frac{f_0^2}{gH_0}\psi + f_0\frac{H_0 - H(x,y)}{H_0}. \tag{5.29}$$

As a result, potential vorticity determines the velocity field that transports it, and thus the dynamics of the flow are completely controlled by it. While quasi-geostrophic models can have an arbitrary number of layers, we focus on a 2-layer quasi-geostrophic (QG) model. model considered here[130] consists of an advection-diffusion equation for the potential vorticity q_i, in each of two immiscible layers with fractional layer thickness

$$\delta = H_1/H_0, \quad 1 - \delta = H_2/H_0, \tag{5.30}$$

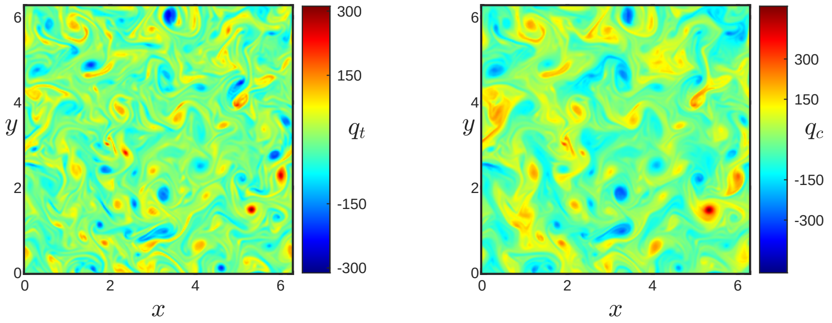
respectively (where $H_0 = H_1 + H_2$) and mean zonal velocities $U_1 > U_2$. The domain under discussion is a square doubly periodic domain with rigid-lid surface boundary conditions. The governing equations for the potential vorticity (PV) of each layer become

$$\begin{aligned} \partial_t q_1 = &- J(\psi_1, q_1) - \beta\partial_x\psi_1 \\ &- (U_1 - U_2)L_P^{-2}(1-\delta)\partial_x\psi_1 \\ &- (U_1 - U_2)(1-\delta)\partial_x q_1 + F_1, \end{aligned} \tag{5.31}$$

and

$$\begin{aligned} \partial_t q_2 = &- J(\psi_2, q_2) - \beta\partial_x\psi_2 \\ &+ (U_1 - U_2)L_P^{-2}\delta\partial_x\psi_2 \\ &+ (U_1 - U_2)\delta\partial_x q_2 + F_2 - r\nabla^2\psi_2, \end{aligned} \tag{5.32}$$

where the field q_1 corresponds to the upper-layer PV and q_2 to the bottom-layer PV, with ψ_i being the respective streamfunctions and $J(a,b) = \partial_x a\partial_y b - \partial_y a\partial_x b$ is the Jacobian operator. The β terms arise from the variation of the vertical projection of Coriolis frequency with latitude and the k_d^2 terms result from the imposed shear.



(a)

(b)

FIG. 5-8. Typical snapshots (vorticity fields) of the (a) barotropic and (b) baroclinic mode for baroclinic ocean turbulence at high latitudes.

Potential vorticity is defined via the following relations

$$
\begin{aligned}
q_1 &= \nabla^2\psi_1 - \frac{f_0^2}{g' H_1}(\psi_1 - \psi_2),\\
q_2 &= \nabla^2\psi_2 + \frac{f_0^2}{g' H_2}(\psi_1 - \psi_2) + f_0\frac{h_b}{H_2}.
\end{aligned}
\tag{5.33}
$$

The dynamics can also be described in terms of the barotropic and baroclinic modes and their corresponding streamfunctions

$$
\begin{aligned}
q_t &= \delta q_1 + (1-\delta) q_2 = \nabla^2\psi_t\\
\psi_t &= \delta\psi_1 + (1-\delta)\psi_2\\
q_c &= \sqrt{\delta(1-\delta)}(q_1 - q_2) = (\nabla^2 - k_d^2)\psi_c\\
\psi_c &= \sqrt{\delta(1-\delta)}(\psi_1 - \psi_2).
\end{aligned}
\tag{5.34}
$$

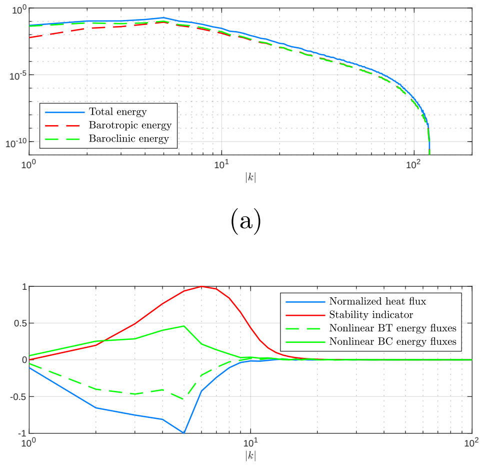
The model dynamics can then be rewritten in terms of the barotropic and baroclinic modes. For periodic boundary conditions with a flat bottom, this yields

$$
\begin{aligned}
\partial_t q_t =& - J(\psi_t, q_t) - J(\psi_c, q_c) \\
& - (1-\delta) r \nabla^2 (\psi_t - a^{-1}\psi_c) \\
& - U \partial_x \nabla^2 \psi_c - \beta \partial_x \psi_t - \nu \nabla^4 q_t \\
\partial_t q_c =& - J(\psi_t, q_c) - J(\psi_c, q_t) - \xi J(\psi_c, q_c) \\
& + \sqrt{\delta(1-\delta)} r \nabla^2 (\psi_t - a^{-1}\psi_c) - \beta \partial_x \psi_c \\
& - U \partial_x (\nabla^2 \psi_t + \lambda^2 \psi_t + \xi \nabla^2 \psi_c) - \nu \nabla^4 q_c,
\end{aligned}
\tag{5.35}
$$

where

$$
\lambda^2 = k_d^2, \quad \xi = \frac{1-2\delta}{\sqrt{\delta(1-\delta)}}, \tag{5.36}
$$

with ξ expressing the triple interaction coefficient and $U = \sqrt{\delta(1-\delta)}\,(U_1 - U_2)$ is the shear intensity. Three different regimes can be distinguished for this model, designated by values of the model parameters: (i) ‘low latitude’ or weakly supercritical ($\beta \approx k_d^2/2, r = 1$), (ii) ‘mid latitude’ or moderately supercritical ($\beta \approx k_d^2/4, r = 4$) and (iii) ‘high latitude’ or strongly supercritical ($\beta \approx 0, r = 16$). Hence, β decreases as latitude increases, while the bottom friction coefficient r is increased with in order to keep the inverse energy cascade from reaching the resolution limit. The drag coefficient varies as β changes to control the scale at which the inverse energy cascade is absorbed.



(a)

(b)

FIG. 5-9. (a): Barotropic, baroclinic and total energy with respect to the wavenumber $|k|$. (b): Wavenumber-averaged heat flux normalized over its maximum value; Stability indicator: $\max_{|\boldsymbol{k}|=k} \mathrm{Re}\lambda_i(k)$ normalized over its maximum magnitude, where $\lambda_i(k)$ are the vertical eigenvalues for each wavenumber; Wavenumber-averaged BT/BC nonlinear energy fluxes.

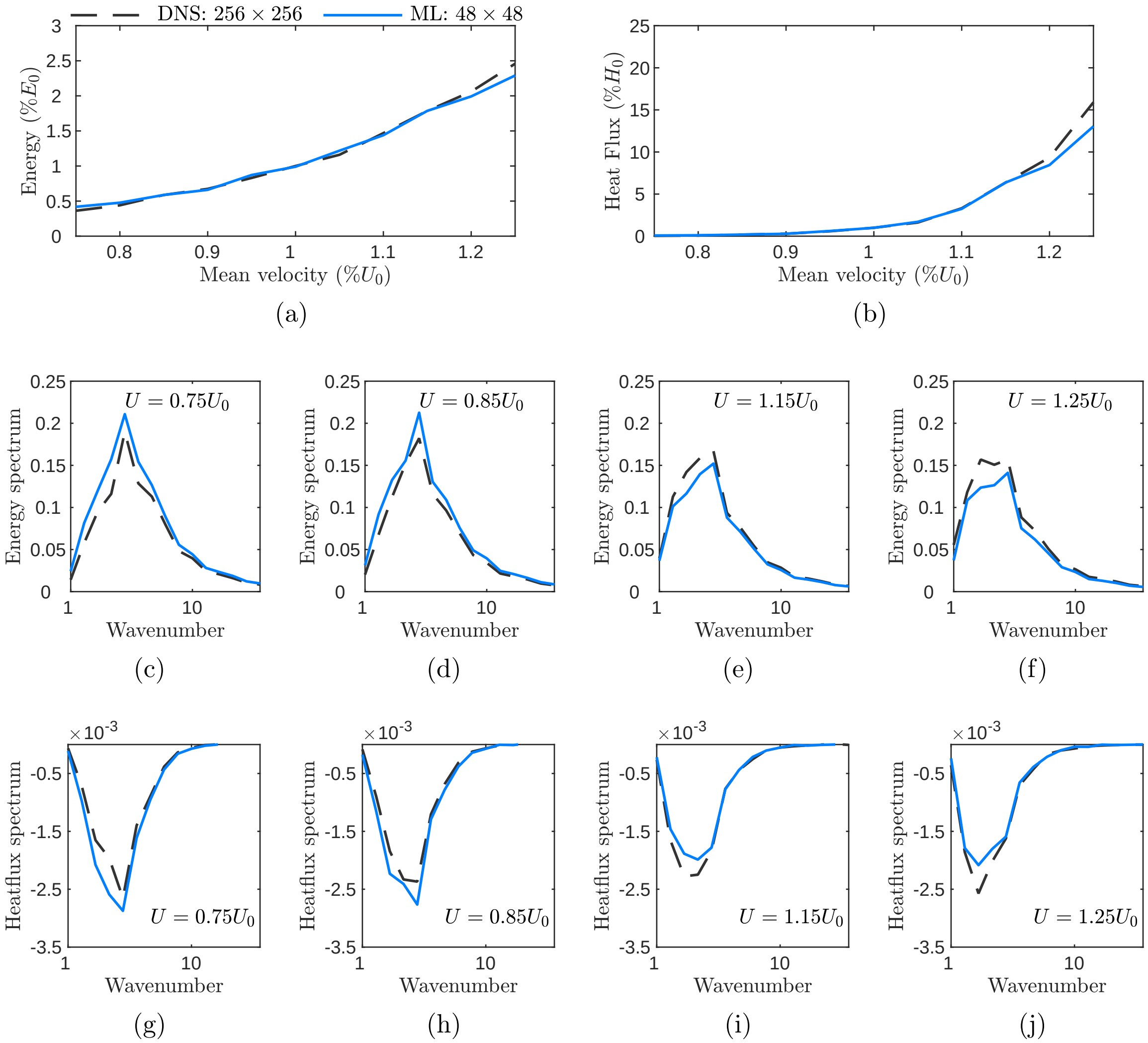
The results presented here are derived for parameters $\delta = 0.2, r = 9, \beta = 10$ and $\lambda = 10$; a set of parameters that corresponds to baroclinic ocean turbulence at high-latitudes. A typical snapshot of the q_t, q_c is shown in Fig. 5-8. Typical energy properties of such flows are shown in Fig. 5-9.

For the implementation of the presented method, the potential vorticity is expanded to

$$q_j = \sum_{i=1}^{N} Z_i(t) v_i. \tag{5.37}$$

Due to the periodicity of the domain, Fourier modes are used as a basis.

To test the performance of the closure scheme, we first train the model for $U = 1.00$ and to assess its performance we test it for different mean velocity, chosen within the interval $U \in [0.75, 1.25]$. For training, validation and testing the flow is solved



(a) (b)

(c) (d) (e) (f)

(g) (h) (i) (j)

FIG. 5-10. Comparison between DNS results (dashed line) and ML results (solid line) for (a-b) percentage comparison of the average energy and heat flux for different shear stresses and (c-j) energy and heat flux radially averaged spectrum for 4 different testing cases.

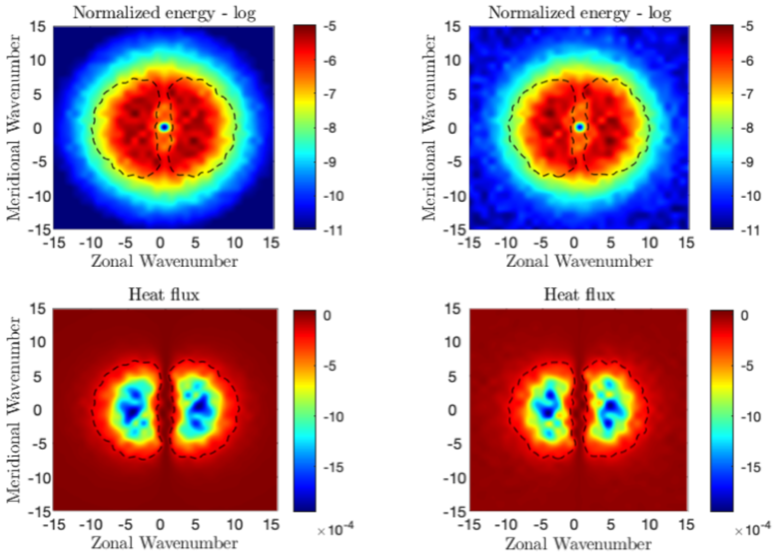


FIG. 5-11. Comparison between DNS simulations and ML model for $U = 0.95$. Results show total energy spectrum and heat flux. The black dashed line is the $-10\% \max_k |\langle H_f \rangle_{kl}|$ contour of the heat flux field.

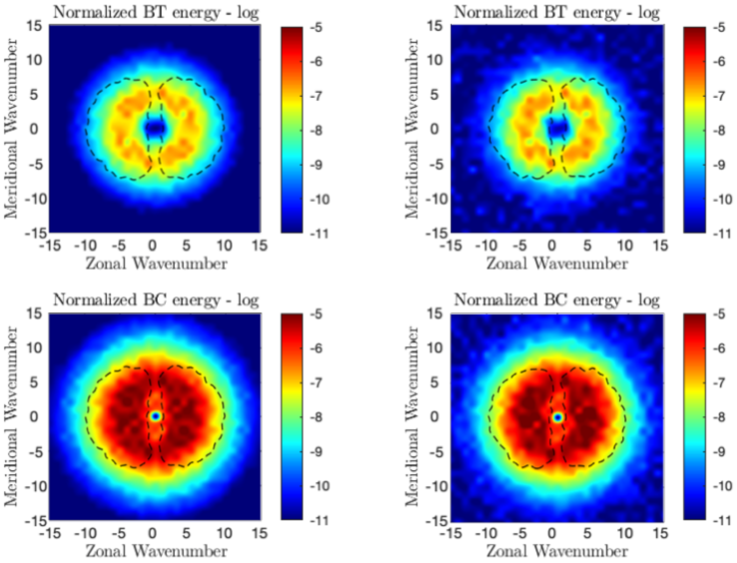


FIG. 5-12. Comparison of barotropic and baroclinic energy between spectral code and ML-closure scheme for $U = 0.95$.

assuming doubly periodic lateral boundary conditions and a 256×256 descritization. For the coarse-scale simulations a discretization 48×48 is utilized for the mean flow, while for the covariance we include all Fourier modes with wavenumbers $|\boldsymbol{k}| \leq 16$. To assess the performance of the closure scheme, the first metric used will be the energy of the system and its spectrum

$$\begin{aligned} E^{\text{total}} &= E^{\text{t}} + E^{\text{c}} \\ &= \sum_{\boldsymbol{k}} \left[|\boldsymbol{k}|^2 |\hat{\psi}_{k,l}|^2 + \left(|\boldsymbol{k}|^2 + \lambda^2 \right) |\hat{\tau}_{k,l}|^2 \right], \end{aligned} \tag{5.38}$$

where E^{t} and E^{c} are the energy carried by the barotropic and baroclinic modes respectively. Parameters k and l signify the zonal and meridional component of wavenumber, respectively. Furthermore, the heatflux and its spectrum will also be used as metrics

$$H_f = \frac{\lambda}{U^2} \sum_{|\boldsymbol{k}|} \frac{i\boldsymbol{k} \hat{q}_{\psi,kl} \hat{q}^*_{\tau,kl}}{|\boldsymbol{k}|^2 \left(|k|^2 + \lambda^2 \right)}. \tag{5.39}$$

In Figure 5-10 the total mean energy and heatflux are shown for different values of $U \in [0.75, 1.25]$; we present the results from the coarse resolution solver with the ML closures and compare these with the DNS. We also show the radially-averaged energy and heatflux spectra and note the favorable comparison between the DNS results and the data-informed closure-scheme. For a more detailed comparison, the energy and heatflux spectra for the case $U = 0.95$ are also computed. Results are presented in Fig. 5-11, where the total normalized energy spectrum, the heat flux and nonlinear energy fluxes are compared with DNS calculations. In Fig. 5-12 the energy components carried by the barotropic and baroclinic modes respectively are also compared with DNS results. In all cases, the coarse-scale simulation is able to

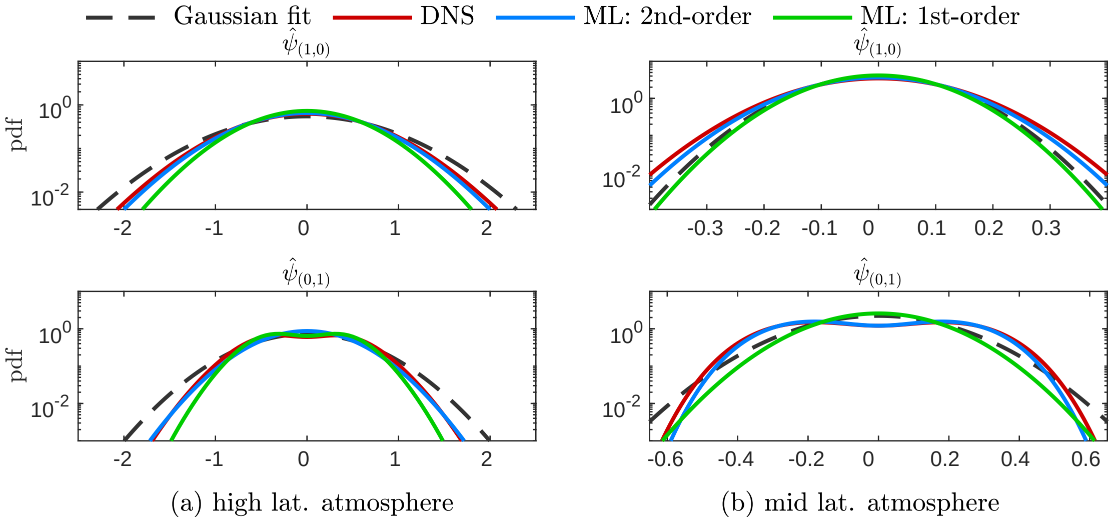


FIG. 5-13. Comparison of pdf of deviations between DNS (red line), 1st-order-closure (green line), 2nd-order-closure (blue line). The best fitted Gaussian pdf for the DNS results (black line) is also shown with dashed line. Results are shown for (a) high latitude. and (b) mid latitude atmospheric flows.

accurately capture the equilibrium statistics of the flow.

Furthermore, in Fig. 5-13 the marginal pdfs for the leading modes $\hat{\psi}_{(1,0)}, \hat{\psi}_{(0,1)}$ of the barotropic streamfunction $\psi = (\psi_1 + \psi_2)/2$ are presented. Training takes place for flows with parameters $(\beta, r) = (0.8, 0.2), (1.5, 0.1), (2.5, 0.1)$. We compare the closure schemes for two cases: the mid-latitude case that corresponds to $k_d = 4, \beta = 2, r = 0.1$, and the high-latitude case that corresponds to $k_d = 4, \beta = 1, r = 0.2$.

We also include a comparison with the first-order closure scheme [29]. The first-order closure results use the same resolution as the presented closure, with a TCN-based architecture and a constraint to the loss function that enforces the energy-preserving property of the quadratic nonlinearity of the system. We observe that the second-order closure captures very accurately the non-Gaussian structure of the pdf. This is not the case for the first-order closure which typically underestimates the variance but also misses the bimodal character of the mode $\hat{\psi}_{(0,1)}$. Similarly, in

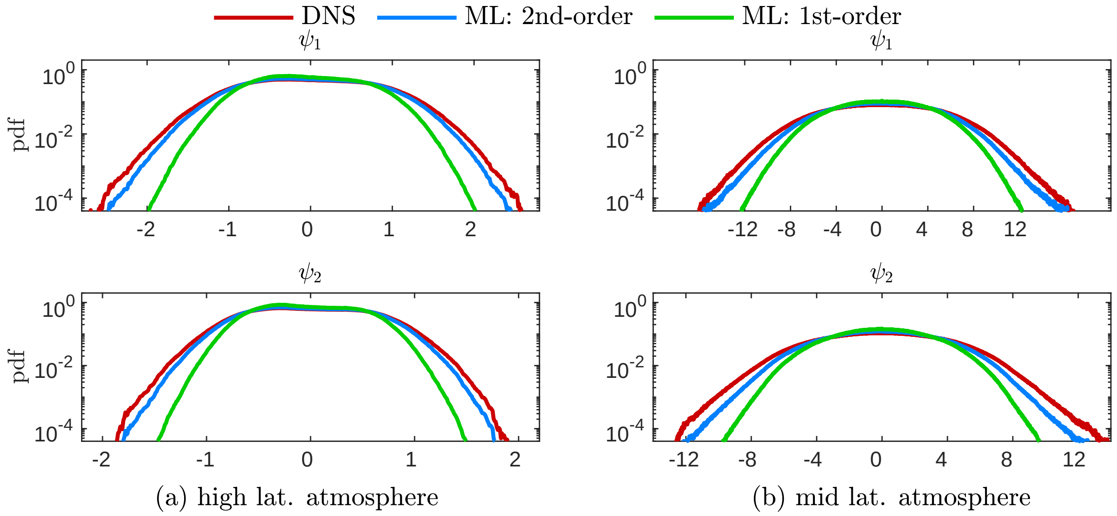


FIG. 5-14. Comparison of pdf of streamfunctions between DNS (red line), 1st-order-closure (green line), 2nd-order-closure (blue line). Results are shown for (a) high latitude. and (b) mid latitude atmospheric flows.

Fig. 5-14, the pdf of the top and bottom layer streamfunctions are shown. Again, the proposed 2nd-order closure outperforms the 1st-order closure scheme. Discrepancies in the tails of the pdf may be due to unresolved high wavenumbers, especially since the large-scale modes seem to be well approximated by the 2nd-order scheme. Finally, it is noted that training the second-order scheme with the same hyperparameters but setting $\lambda = 0$ yields a closure that becomes unstable as the flow evolves, highlighting the importance of the constraint in numerical stability.

Chapter 6

Nonintrusive statistical corrections of climate models

Abstract

In the previous chapters, online hybrid closure schemes were presented. This chapter discusses a non-intrusive systematic framework for improving the predictions of statistical quantities for turbulent systems. The focus here lies on correcting climate simulations obtained by coarse-scale models, in an offline manner. Specifically, failure to incorporate all relevant scales in climate simulations leads to discrepancies in the energy spectrum as well as higher order statistics. While high resolution simulations or reanalysis data are available, at least for short periods, they cannot be directly used as training datasets to machine learn a correction for the coarse-scale climate model outputs, since chaotic divergence, inherent in the climate dynamics, makes datasets from different resolutions incompatible. To overcome this fundamental limitation we employ coarse-resolution model simulations nudged towards high quality climate realizations. The nudging term is sufficiently small to not "pollute" the coarse-scale dynamics over short time scales, but also sufficiently large to keep the coarse-scale simulations "close" to the reference trajectory over larger time scales. We emphasize that the nudging step is used only for the training phase. Once training is complete, we perform free-running coarse-scale simulations without nudging and use those as input to the machine-learned correction operator to obtain high-quality (corrected) outputs. Results are shown for quasi-geostrophic flows and realistic climate data.

6.1 Method formulation

Accurate statistical climate predictions require high-fidelity simulations that come with large computational cost. As a result, improving upon the predictions of coarse-scale climate models has become a critical goal in order to develop credible climate scenarios. The developed method aims to augment the accuracy of statistical properties of coarse-scale free-running (i.e. without the influence of observations which are obviously not available for future projections) climate models, via corrections based on past data. While many approaches tackle this problem with an online (i.e. intrusive) correction term in the evolution equations of the model, they usually face severe stability issues. To circumvent this obstacle, a non-intrusive approach is developed. Hence, after a free-running coarsely resolved climate output has been generated, the hybrid approach corrects the model output in a post-processing manner. A reference high-fidelity dataset is selected for testing the effectiveness of the scheme.

Since the goal of the approach is to correct the long-time statistics of coarse-scale climate simulations in a post-processing manner, it is important to isolate the main discrepancies between the coarse-scale simulations and the reference data that are responsible for these differences. In general, discrepancies between two turbulent simulations, one high-fidelity (i.e. reference) and one free-running coarse, can be grouped into two categories: (i) discrepancies due to chaotic divergence; (ii) discrepancies due to deformation of the attractor due to coarse-scale resolution.

Chaotic divergence is an intrinsic property of turbulent systems. It can be observed even between two solutions of the same dynamical system, with ever slightly different initial conditions. It is a manifestation of the fact that by definition, at least one of the eigenvalues of the linear part of the system is positive. As a result, infinitesimal energy transferred to perturbations along these directions will result in finite magnitude perturbations. The system is allowed to equilibrate with the intervention of nonlinear terms that will transfer this energy from the unstable perturbations to stable ones. However, the two deviating trajectories will still remain on the same attractor and thus retain the same statistical properties. Therefore these chaos-induced discrepancies should not contribute to the correction scheme for long-time statistics.On the other hand, difference in long-time statistics implies a different statistical steady-state and thus different attractors. These intrinsic dynamical differences between the simulations produce energy discrepancies in various scales between the produced datasets. It is exactly these corrections we aim to learn and fix.

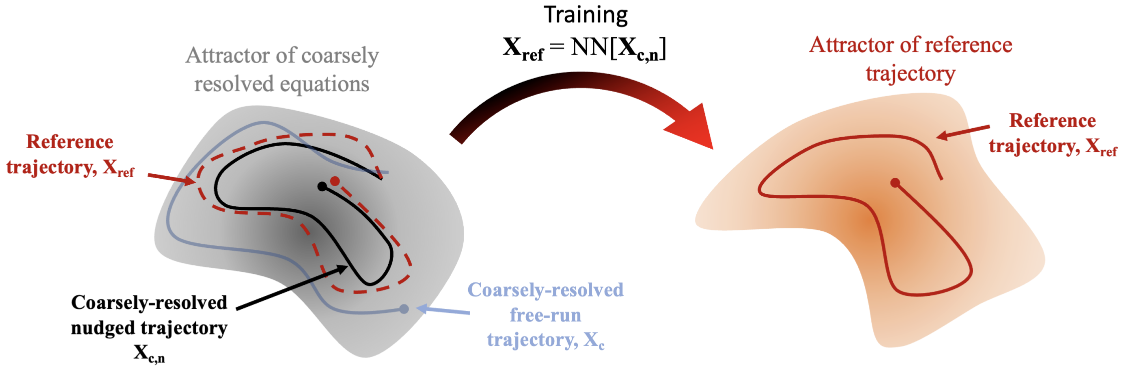

Training
$\mathbf{X}_{\mathbf{ref}} = \mathrm{NN}[\mathbf{X}_{\mathbf{c,n}}]$
Attractor of coarsely resolved equations
Reference trajectory, $\mathbf{X}_{\mathbf{ref}}$
Coarsely-resolved nudged trajectory $\mathbf{X}_{\mathbf{c,n}}$
Coarsely-resolved free-run trajectory, $\mathbf{X}_{\mathbf{c}}$
Attractor of reference trajectory
Reference trajectory, $\mathbf{X}_{\mathbf{ref}}$

FIG. 6-1. **Description of the method** that learns a map between the attractor of the coarsely-resolved equations and the attractor of the reference trajectory. Left: the red dashed curve represents a reference trajectory. The black curve is a coarsely-resolved nudged trajectory towards the reference trajectory. The blue curve is the free-run coarsely-resolved trajectory that is not used for training (shown for reference). Right: the target attractor and the target trajectory (red), same as the dashed curve shown at the left plot. For training we use the coarsely-resolved nudged trajectory as input and the reference trajectory as output to machine learn a map between them. After we obtain the map we use as input coarsely-resolved free-run simulations (blue) and obtain a trajectory that accurately captures the shape of the target attractor.

Given the two previous observations, it is clear that it is not possible to use a dataset of free-running climate simulations and a reference simulation and try to machine learn a map between the two. To eliminate the problematic component, i.e. chaos-induced divergence we design a new coarse-scale dataset (we call it nudged and denote it as $\mathbf{X}_{c,n}$). Ideally, one can produce a dataset that is preserving the coarse-scale behavior of the climate model but does not suffer from the chaos-induced divergence with respect to the reference solution. To this end, the concept of nudging, that has been used extensively in the context of data assimilation [166, 191], is employed. Specifically, we utilize the coarse-scale solver with an extra term, the nudging term, that is 'pulling' the coarse-scale solution close to the reference solution:

$$\frac{\partial \mathbf{X}_{c,n}}{\partial t} = \mathcal{D}\left(\mathbf{X}_{c,n}\right) + \mathcal{P}\left(\mathbf{X}_{c,n}\right) + \mathcal{N}\left(\mathbf{X}_{c,n}; \mathbf{X}^{\text{ref}}\right), \tag{6.1}$$

where $\mathcal{D}$ correspond to the dynamics, $\mathcal{P}$ corresponds to physics terms and the relaxation term $\mathcal{N}$ is called the nudging tendency and it corrects the coarse-scale solution based on the reference solution. The non-rigorous separation between dynamics and physics is done to exemplify that the two terms are resolved in a different manner in the E3SM solver. In this study, the nudging tendency $\mathcal{N}$ is given by the algebraic term

$$\mathcal{N}\left(\mathbf{X}_{c,n} - \mathbf{X}^{\text{ref}}\right) = -\frac{1}{\tau}\left(\mathbf{X}_{c,n} - \mathcal{H}\left[\mathbf{X}^{\text{ref}}\right]\right). \tag{6.2}$$

Parameter τ is a relaxation timescale that has a large value (so that $1/\tau$ is small compared with the other terms in the equation), and $\mathcal{H}$ is an operator that maps $\mathbf{X}^{\text{ref}}$ to the coarse resolution. A schematic of the proposed mapping learned during training can be seen in fig. 6-1. The resulted nudged trajectory (black curve on the right panel) is subjected to this very small perturbation, the nudging term that is keeping it close to the reference trajectory. Moreover, because the overall magnitude of the nudging term is very small the long-time statistics of the coarsely resolved before-nudge trajectory should be close to that of the free-running coarsely-resolved trajectory that starts from the before-nudge state (shown with blue color). The latter will naturally diverge from the reference solution if not continuously being nudged due to chaotic properties, even if it was initiated very close to it. Having the before-nudge trajectory we can now use it to machine learn a scheme that will map it to the reference trajectory. That is the basic approach of our framework. We emphasize that nudging is used ONLY for the generation of training data. Once the map has been trained we will feed it with free-run simulations (i.e. free-run coarse-scale climate simulations without nudging) to obtain outputs that have corrected long-time statistics, i.e. represent the target attractor accurately.

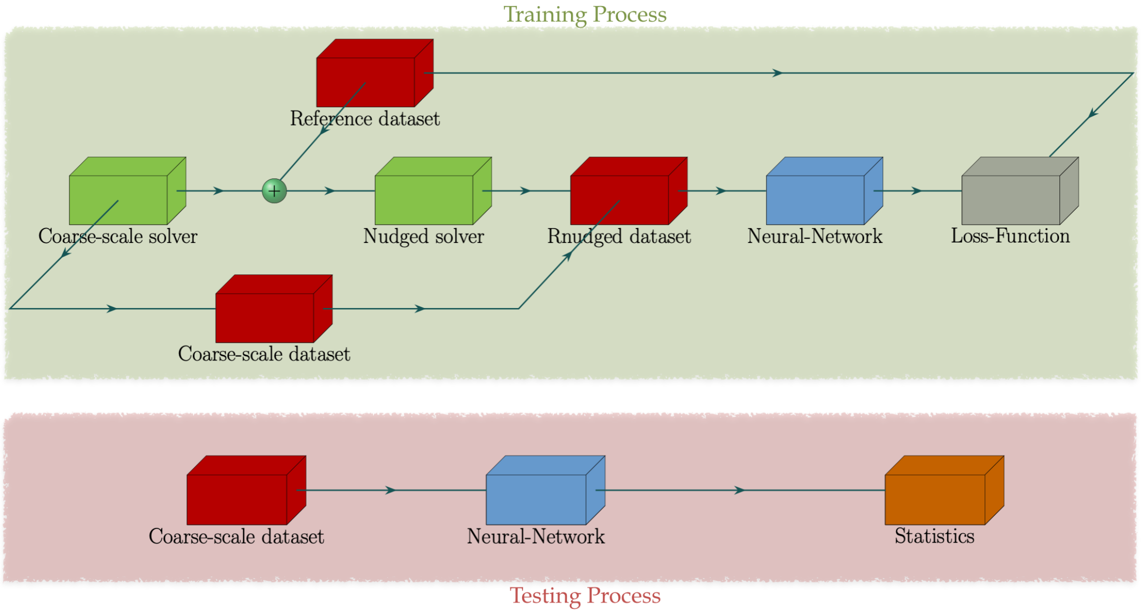


FIG. 6-2. Schematics of the training process (top) and testing process (bottom), for the non-intrusive hybrid method.

The resulting training and testing process are described in fig. 6-2. During training, the Nudged solver is used to produce training data. After a spectral correction of the data (described in the next section), the resulting R-nudged dataset is used as input for the neural network. The neural network then learns a mapping between the reference data and the input R-nudged dataset. The neural network used is described in detail in subsection 'Neural Network Architecture'. During testing, the coarse-scale solver is used to generate a free-running coarse-scale dataset. This dataset is used as input to the trained neural network which produces a corrected dataset with the desired statistics. *Hence, during testing, the model is not assessed in its ability to mimic the reference data snapshot-by-snapshot but by its ability to learn its underlying statistics.*

Revisiting the nudging procedure, parameter τ is chosen so the nudged solution $\mathbf{X}_c$ satisfies two properties: (a) it reduces the divergence of the nudged simulation from the reference solution $\mathcal{H}\left[\mathbf{X}^{\mathrm{ref}}\right]$, i.e. allowing for a generalizable mapping between the two datasets; (b) it resembles the statistical properties of the coarse-scale free-running simulation. The second property is important in the context of machine learning to ensure that the learned mapping during training will be applicable while testing using coarse data. This implies that the attractor of the Nudged simulation has the same shape as the attractor of the free-running coarse-scale dataset. However, no parameter τ can be found that explicitly satisfies this condition. This is due to the arbitrariness of the algebraic form of the nudging term. While an algebraic term is easily implemented it yields a constant dissipation rate across all wavenumbers which are in general not consistent with the dynamics of the system. This leads to suppression of extreme events and thus statistics with less heavy tails. A remedy for this issue is show in subsection 'Spectral Correction of Nudged Dataset', where the energy spectrum of nudged simulations is brought closer to that of the coarse-scale free-running.

6.2 Application to quasigeostrophic flows

The first model under investigation simulates two-layer quasigeostrophic flows. In a dimensionless form, its evolution equation is given by

$$\frac{\partial q_j}{\partial t} + \boldsymbol{v}_j \cdot \boldsymbol{\nabla} q_j + \left(\beta + k_d^2 U_j\right) \frac{\partial \psi_j}{\partial x} = -\delta_{2,j} r \Delta \psi_j - \nu \Delta^s q_j, \tag{6.3}$$

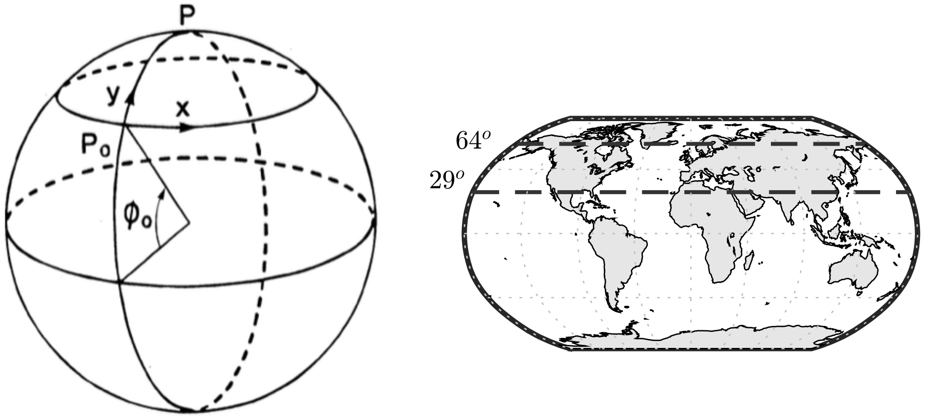
where $j = 1, 2$ corresponds to the upper and lower layer respectively and the flow is defined in the horizontal domain $(x, y) \in [0, 2\pi]^2$. Doubly periodic boundary conditions are assumed. A mean zonal flow of intensity $U_1 = U$ and $U_2 = -U$ is imposed on each layer respectively. The two layers are assumed to have the same width, k_d denotes the deformation radius, r the bottom-drag coefficient and β is the beta-plane approximation parameter. The potential vorticity (PV) q_j and corresponding streamfunction ψ_j are related via the inversion formulae

$$q_j = \Delta \psi_j + \frac{k_d^2}{2} \left(\psi_{3-j} - \psi_j\right), \quad j = 1, 2. \tag{6.4}$$

The quasigeostrophic model described by eq. (6.3) is used here to simulate mid latitude and high latitude atmospheric flows that are forced by an imposed shear current [134]. While in the previous chapter quasigeostrophic flows were used to study oceanic turbulence cases, here we focus on atmospheric flows since subsequently the model will be applied to predict ERA5 atmopheric reanalysis data. The principal parameter to be varied is the beta-plane approximation parameter β. To locate an interval of values for β that correspond to physically relevant simulations, the Coriolis acceleration need to be taken into account. Taking into account the dimensionless form of eq. (6.3), the Coriolis frequency at some latitude ϕ_0 is defined as

$$f = 2\frac{\Omega L_y}{U_{\text{scale}}} \sin\left(\phi_0 + L_y \frac{y}{R}\right), \tag{6.5}$$

where Ω is the frequency of the rotation of the earth and set to $\Omega = 7.2925 \cdot 10^{-5}\,[\text{rad/sec}]$. R is the radius of the earth and set to $R = 6378\,[\text{km}]$. Parameters L_y and U_{scale} are scales for the meridional extent and velocity field respectively.



(a)

(b)

FIG. 6-3. (a) Schematic of beta-plane approximation on a globe. (b) The meridional extent that the simulations with $\beta \in [1, 2]$ correspond to.

A schematic depicting the domain on a globe is seen at fig. 6-3, subfigure (a). Utilizing a Taylor expansion, the beta-plane approximation coefficient arises from

$$\begin{aligned} f &= 2\frac{\Omega L_y}{U_{\text{scale}}} \sin\phi_0 + 2\Omega \frac{L_y^2}{U_{\text{scale}}} \frac{\cos\phi_0}{R} y + O\left(L_y^2 \frac{y^2}{R^2}\right) \\ &\approx 2\frac{\Omega L_y}{U_{\text{scale}}} \sin\phi_0 + \underbrace{2\Omega \frac{L_y^2}{U_{\text{scale}}} \frac{\cos\phi_0}{R}}_{\beta} y \end{aligned} \quad . \tag{6.6}$$

For atmospheric flows, a standard assumption is $L_y \sim 330$ [km] and $U_{\text{scale}} \sim 100$ [m/sec]. Hence, a range of β values for midlatitude and high latitude cases, corresponds to $\beta \in [1, 2]$. Indeed, one can check that $\beta = 1$ corresponds to $\phi_0 \approx 29^{\text{o}}$ and $\beta = 2$ corresponds to $\phi_0 \approx 64^{\text{o}}$. The extent of this regime on a globe map is seen in fig. 6-3, subfigure (b) and relevant parameter values can be seen in table 6.1.

In this investigation, neural networks are trained given as input the predictions of a free running coarse-scale simulation $(\psi_1^{\text{coarse}}, \psi_2^{\text{coarse}})$ of eq. (6.3), with the target output being a modified time-series $\left(\text{e.g. } \psi_1^{\text{ML}}, \psi_2^{\text{ML}}\right)$ that will have the same statistics as a fine-scale reference simulation $\left(\psi_1^{\text{ref}}, \psi_2^{\text{ref}}\right)$. The de-coupling of the data-informed correction process and the initial simulation phase is justified by the fact that the goal is not to make phase corrections at each time-step but retrieve the correct statistics for the current flow parameters.

regime	β	k_d	U	r	ν	s
atmosphere, high lat.	1	4	0.2	0.1	1×10^{-13}	4
atmosphere, mid lat.	2	4	0.2	0.1	1×10^{-13}	4

Table 6.1. Parameter values for different atmosphere regimes.

As discussed in section 6.1, it is not feasible for a neural network to learn a generalizable mapping directly between $(\psi_1^{\text{coarse}}, \psi_2^{\text{coarse}})$ and $\left(\psi_1^{\text{ref}}, \psi_2^{\text{ref}}\right)$. To that end, to produce coarse-scale simulations for training, a relaxation term Q is added to the evolution eq. (6.3) to become

$$\frac{\partial q_j}{\partial t} + \boldsymbol{v}_j \cdot \boldsymbol{\nabla} q_j + \left(\beta + k_d^2 U_j\right) \frac{\partial \psi_j}{\partial x} = -\delta_{2,j} r \Delta \psi_j - \nu \Delta^s q_j + Q\left(q_j - q_j^{ref}\right), \qquad (6.7)$$

where $j = 1, 2$. The term Q is called nudging tendency and it corrects the coarse-scale solution based on the fine-scale reference solution. In this study, the nudging tendency Q is given by the algebraic term

$$Q\left(q_j - q_j^{\text{ref}}\right) = -\frac{1}{\tau}\left(q_j - \mathcal{H}\left[q_j^{\text{ref}}\right]\right). \tag{6.8}$$

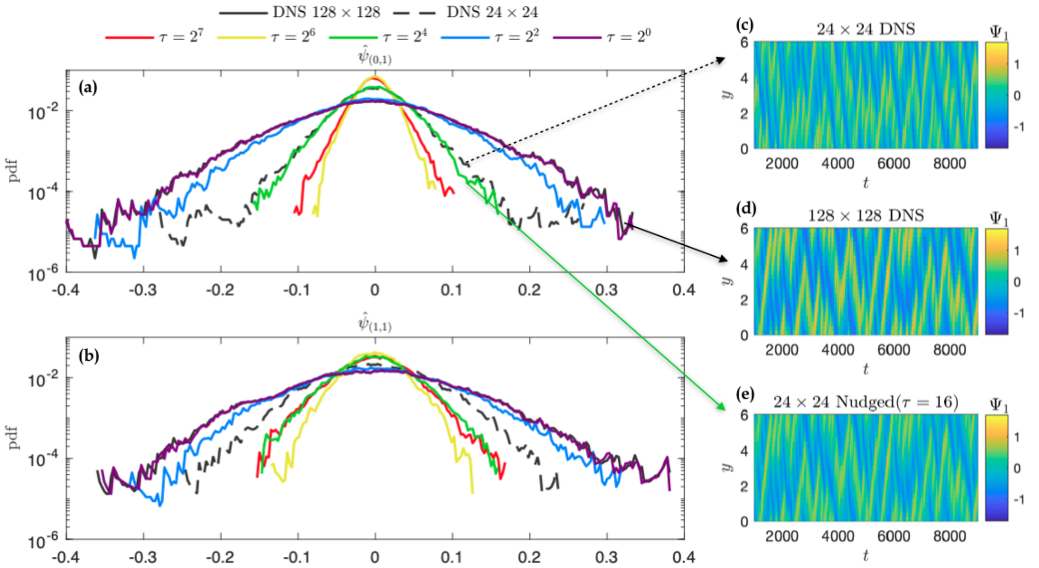
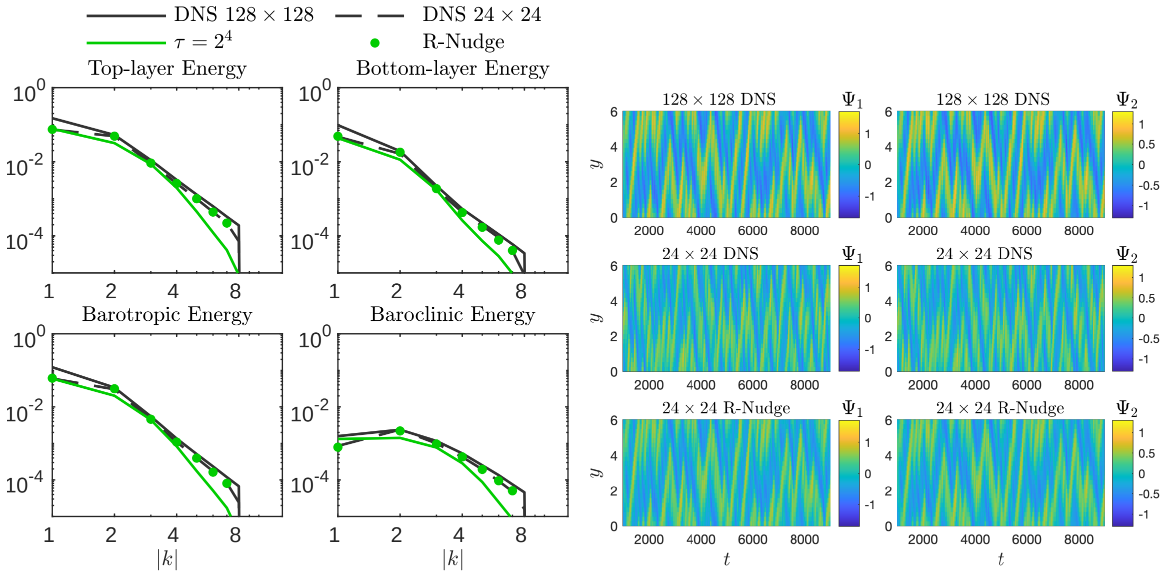


FIG. 6-4. Pdfs of wavenumber amplitudes (a) $\hat{\psi}_{(0,1)}$ and (b) $\hat{\psi}_{(1,1)}$ of the barotropic mode for various simulations. Time evolution of the zonally-averaged profile of the top layer streamfunction for (c) coarse-scale free running simulation (d) reference fine-scale simulation (e) nudged coarse-scale simulation for $\tau = 16$. Flow parameters correspond to the mid latitude case with $\beta = 2.0$ and $r = 0.1$.

Parameter τ is a relaxation timescale to be determined, and $\mathcal{H}$ is an operator that maps q_j^{ref} to the coarse resolution. In this particular test case, $\mathcal{H}$ simply removes the high-frequency wavenumbers from the reference solution. Parameter τ is chosen so as the nudged solution q_j^{nudged} satisfies two properties: (i) it stops the nudged simulation from diverging from the reference solution q_j^{ref}, i.e. allowing for the mapping between the two time histories (ii) it resembles the statistical properties of the coarse-scale free running simulation. The second property is important to ensure that the mapping learned during training, will be applicable during testing with un-nudged data $(\psi_1^{\text{coarse}}, \psi_2^{\text{coarse}})$. As can be seen in fig. 6-4, a value that approximately satisfies both properties is $\tau = 16$.

6.2.1 Reverse Spectral Nudging

As discussed in section 6.1 , using algebraically nudged data for training, yields predictions with biases in the tails of the statistics. A main reason for the discrepancies is due to different statistical behaviour of the nudged solution with respect to the free-running coarse data.



(a) (b)

FIG. 6-5. (a) Energy spectra of the Top layer, bottom layer, barotropic and baroclinic mode. (b) Predictions of zonally-averaged streamfunctions for reference simulation, coarse-scale free runing simulation and coarse-scale R-nudged simulation.

One can observe this from fig. 6-4, where the statistics of the nudged solution differ when computing the pdf of $\hat{\psi}_{(1,1)}$. This is a result of discrepancies in the energy spectrum of the nudged solution with respect to the coarse-scale solution, which can be seen in fig. 6-5 for the same flow parameters and $\tau = 2^4$. These energy spectra differences lead to different statistical behaviours of testing data $(\psi_1^{\text{coarse}}, \psi_2^{\text{coarse}})$ and training data $\left(\psi_1^{\text{nudge}}, \psi_2^{\text{nudge}}\right)$.

Discrepancies in the training and testing input distributions will lead to the neural network behaving differently in the two schemes [152]. To remedy the energy spectra differences between the testing data $(\psi_1^{\text{coarse}}, \psi_2^{\text{coarse}})$ and training data $\left(\psi_1^{\text{nudge}}, \psi_2^{\text{nudge}}\right)$, a new method is developed and employed. The process is called 'Reverse Spectral Nudging' with its purpose being to match the energy spectrum of the nudged solution to that of the coarse-scale solution to improve the training process. Hence, while traditional nudging schemes correct the coarse-scale solution with data from the reference solution, the proposed scheme further processes the nudged data by matching its energy spectrum to that of the corresponding free running coarse-scale flow. The corrected nudged data is termed as $\left(\psi_1^{\text{RS-nudge}}, \psi_2^{\text{RS-nudge}}\right)$ and defined as

$$\psi_i^{\text{RS-nudge}}(x, y, t) = \sum_{k,l} R_{k,l} \hat{\psi}_{k,l}^{\text{nudge}}(t) e^{i(kx+ly)}, , \tag{6.9}$$

where $\psi_{k,l}^{\text{nudge}}(t)$ are the spatial Fourier coefficients of $\left(\psi_1^{\text{nudge}}, \psi_2^{\text{nudge}}\right)$ and

$$R_{k,l} = \sqrt{\frac{\mathcal{E}_{k,l}^{\text{coarse}}}{\mathcal{E}_{k,l}^{\text{nudge}}}}, \quad \text{and} \quad \mathcal{E}_{k,l} = \frac{1}{T}\int_0^T \hat{E}_{k,l}(t)\mathrm{d}t = \frac{1}{T}\int_0^T |\hat{\psi}_{k,l}(t)|^2 \mathrm{d}t. \tag{6.10}$$

This scheme produces new data that have exactly the energy spectrum of the free running coarse simulation. Hence, training and testing data come from the same distributions. This property improves significantly the accuracy of the resulted ML scheme. In fig. 6-5, subfigure (a) demonstrates for a mid latitude flow that the energy spectra of the R-nudged solution coincide with the coarse-scale free running spectra. In addition, subfigure (b) shows that the R-nudged data still follow the reference data, allowing for a mapping between $\left(\psi_1^{\text{R-Nudge}}, \psi_2^{\text{R-Nudge}}\right)$ and $\left(\psi_1^{\text{ref}}, \psi_2^{\text{ref}}\right)$.

6.2.2 Loss Function

The first term to be used in the loss function is the L^2-error of the streamfunction predictions:

$$L = \sum_{j=1}^{2} \int_0^{L_x} \int_0^{L_y} \left(\psi_j^{\text{ML}} - \psi_j^{\text{ref}}\right)^2 \mathrm{d}x\mathrm{d}y. \tag{6.11}$$

To introduce some physical consistency in the predictions of the neural network, physical constraints are also incorporated. Let h_j^{eq} be the equilibrium depth of layer j and $h_j^{'}$ be a local variation from this depth. It then naturally follows from mass conservation of an incompressible flow that

$$\int_0^{L_x} \int_0^{L_y} h_j^{'} \mathrm{d}x\mathrm{d}y = 0, \quad j = 1, 2. \tag{6.12}$$

Using geostrophic balance, depth variations can be related to the streamfunction via the equations

$$\psi_1 = \frac{g}{h_0}\left(h_1^{'} + h_2^{'}\right), \quad \psi_2 = \frac{g}{h_0}\left(h_1^{'} + h_2^{'}\right) + \frac{g'}{f_0} h_2^{'}. \tag{6.13}$$

Combining eq. (6.13) with the mass conservation principle eq. (6.12), one gets a physical constraint with respect to the streamfunctions:

$$\int_0^{L_x}\int_0^{L_y}\psi_j \mathrm{d}x\mathrm{d}y = 0, \quad j = 1, 2. \tag{6.14}$$

Incorporating only eq. (6.11) and eq. (6.14) in a loss function however, implies that the predictions for the two layers can be trained independently, a property that is non-physical. The main physical constraint across layers is that while the flow can exhibit velocity jumps across layers, no pressure jumps can be observed. To incorporate this cross-layer constraint, we introduce to addition terms, (i) the inversion relation eq. (6.4) and (ii) the system dynamics eq. (6.3). Incorporating all these terms, the final loss function becomes

$$\begin{aligned}\mathcal{L} = &\sum_{i=1}^{2}\alpha_i \overbrace{\int_0^{L_x}\int_0^{L_y}\left(\psi_i^{\mathrm{ML}} - \psi_i^{\mathrm{ref}}\right)^2 \mathrm{d}x\mathrm{d}y}^{\text{Streamfunction } L^2\text{-error}} + \sum_{i=1}^{2}\beta_i \overbrace{\left|\int_0^{L_x}\int_0^{L_y}\psi_i^{\mathrm{ML}}\mathrm{d}x\mathrm{d}y\right|}^{\text{Continuity/ Mass Conservation}} \\ &+ \sum_{i=1}^{2}\gamma_i \overbrace{\int_0^{L_x}\int_0^{L_y}\left[\Delta\psi_i^{\mathrm{ML}} + \frac{k_d^2}{2}\left(\psi_{3-i}^{\mathrm{ML}} - \psi_i^{\mathrm{ML}}\right) - q_i^{\mathrm{ref}}\right]^2 \mathrm{d}x\mathrm{d}y}^{\text{Vorticity } L^2\text{-error}} \\ &+ \sum_{i=1}^{2}\delta_i \overbrace{\int_0^{L_x}\int_0^{L_y}\left(\frac{\partial q_i^{\mathrm{ML}}}{\partial t} - \frac{\partial q_i^{\mathrm{ref}}}{\partial t}\right)^2 \mathrm{d}x\mathrm{d}y}^{\text{System Dynamics}}\end{aligned} \tag{6.15}$$

where the time-derivative of q_i^{ML} is determined by the equation

$$\frac{\partial q_j^{\mathrm{ML}}}{\partial t} = -\overbrace{\left(\boldsymbol{v}_j^{\mathrm{ML}} + \boldsymbol{V}_j\right)\cdot\boldsymbol{\nabla} q_j^{\mathrm{ML}}}^{\text{Advection}} + \overbrace{\left(\beta + k_d^2 U_j\right)\frac{\partial \psi_j^{\mathrm{ML}}}{\partial x}}^{\text{Coriolis}} - \overbrace{\delta_{2,j} r \Delta\psi_j^{\mathrm{ML}}}^{\text{Bottom Drag}}. \tag{6.16}$$

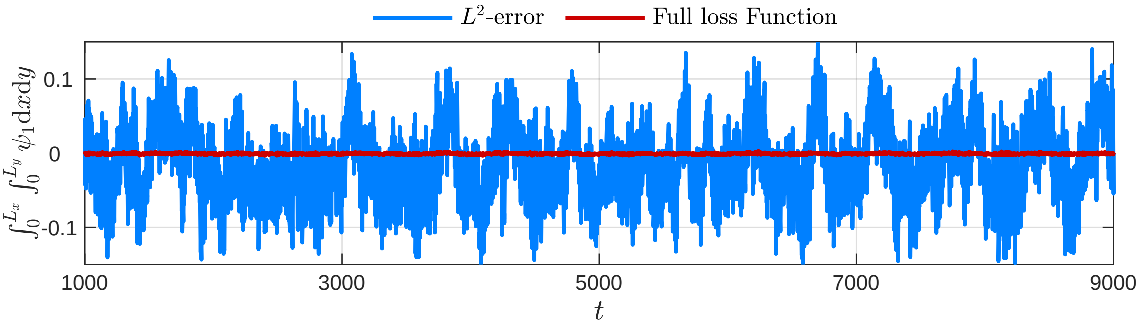


FIG. 6-6. Mass conservation error as calculated for the layer for a neural-network trained with loss function eq. (6.11) (blue line) and loss function eq. (6.15).

where $\boldsymbol{V}_j = (U_j, 0)$ corresponds to the steady background mean zonal flow. Parameters $\{\alpha_j, \beta_j, \gamma_j, \delta_j\}$ for $j = 1, 2$, are factors used to regularize the terms. A validation of enhanced performance when using the new loss function eq. (6.15) can be seen in fig. 6-6, where the predicted mass conservation error at the top layer is computed for a neural network trained with eq. (6.11) and with eq. (6.15).

6.2.3 Neural Network Architecture

This subsection discusses how recurrent neural networks (RNN) are used for the data-informed mappings previously described. In particular, long short-term memory (LSTM) [64] neural networks are employed. Of great interest is the ability of this model to generalize beyond the data seen during training. At first this is investigated in out-of-sample data from the training flow and later further tested on different flow setups. The architecture of the LSTM-based neural-network is shown in fig. 6-7. It consists of an input fully connected layer that compresses the streamfunctions of the two layers from a 24 × 24 × 2 dimension to a 60-valued vector. This layer has a tanh activation function. This vector is then passed as input to a long short-term memory (LSTM) neural network. The output of the neural network is then passed through an output fully connected neural network to produce the final data-informed corrected predictions. The output layer has a linear activation function.

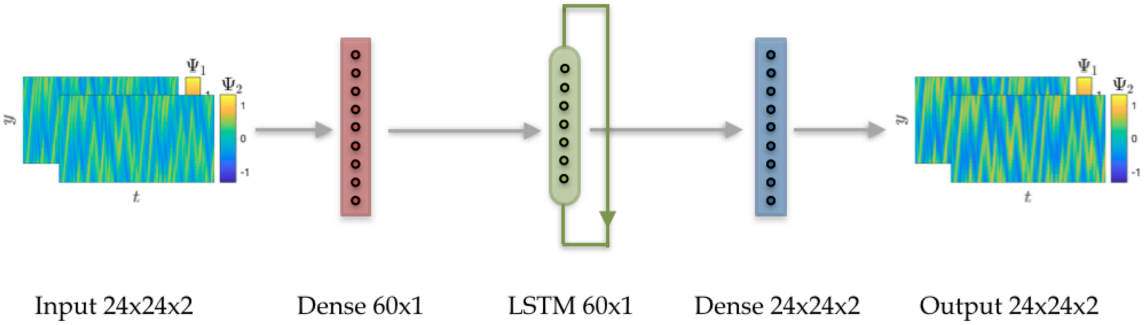


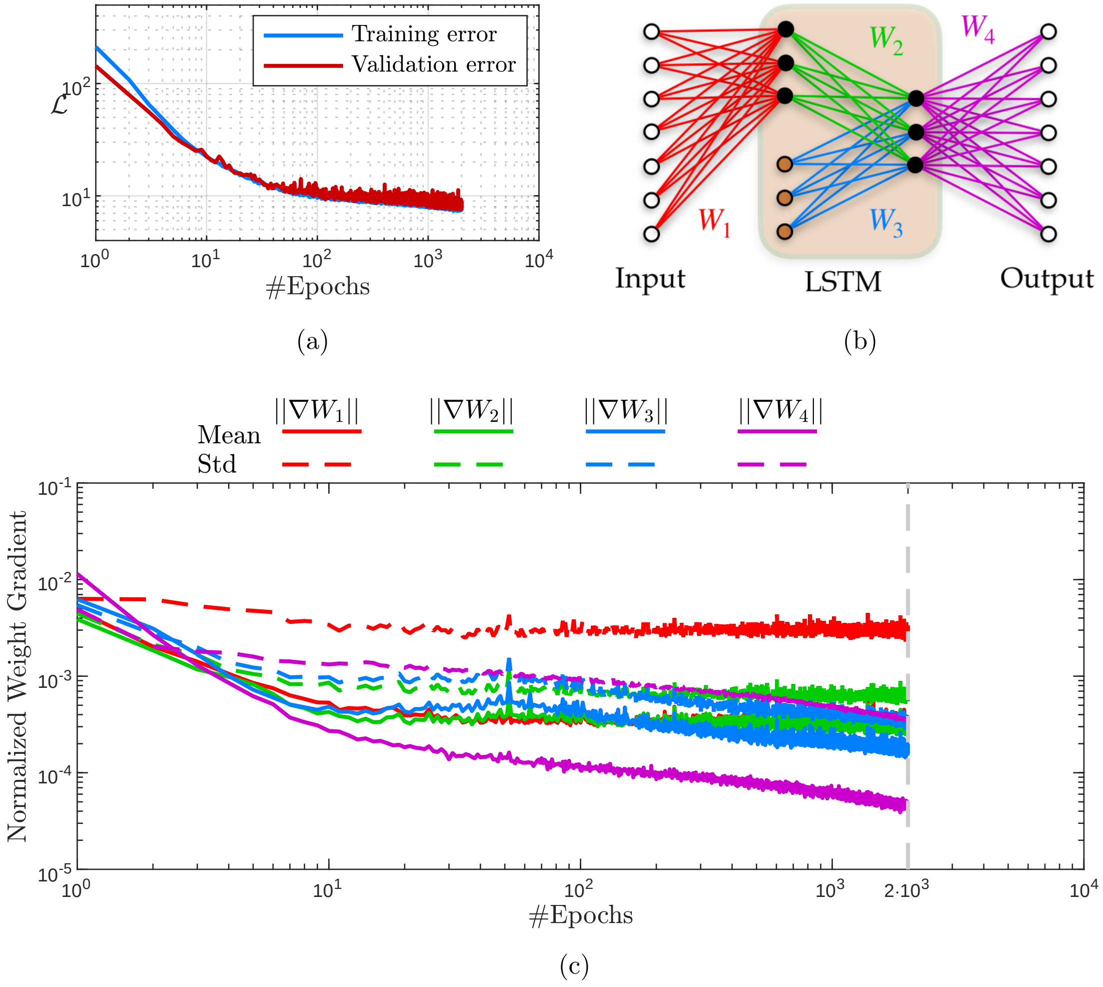
FIG. 6-7. Neural-network architecture using LSTM.

In addition to temporal nonlocality, the model is nonlocal in space. Due to the small size of the system, for a prediction at point $\boldsymbol{x}$ in space, information from the entire domain is used. In the case of E3SM dataset fully-connected layers will be replaced by convolutional layers.

6.2.4 Data assimilation

For initial results, the model is trained on the mid latitude flow parameters. The simulations are carried out until $T = 9000$. For training and testing the time-interval $[1000, 9000]$ is kept where the flow has achieved statistical equilibrium. The neural network is trained using the Adam method [78], which is a first-order gradient descent method. Input and reference data are normalized so that $\psi_1, \psi_2 \in [-1, 1]$. The neural network is trained for 2000 epochs. For training, the time-interval $[1000, 4600]$ is used and the interval $[4600, 5000]$ is used for validation. The data-size used during training here is conservative, to ensure that the model has enough data to learn from. R-nudged data are used as input data. Training takes place for a flow with $\beta = 2$ and

$r = 0.1$.



(a)

(b)

(c)

FIG. 6-8. (a) Training and validation error for the proposed neural network; (b) Schematic diagram of the separation of weights along the architecture of the neural network; (c) Normalized norm of mean and standard deviation of weight gradients for the weights as defined in (b).

In fig. 6-8, subfigure (a) we show the training and validation error at the end of the training process. The close resemblance of validation and training error showcase the generalizability of the scheme. To further show that the neural network has in fact "learned" a meaningfull representation, the gradients of the weights of the architecture are computed for each batch, for each epoch. As explained in [154], when the mean weight gradients are much larger than the corresponding standard deviations, the neural network is still applying an L^2 best fit of the training data. It is when the standard deviation becomes significantly larger than the mean that the neural network starts compressing the data, trying to "forget" features that are not usefull for its predictions. This process can be seen in subfigure (c) of fig. 6-8. Subfigure (b) contains an explanation of the grouping of weights along the architecture. In particular W_2 contains weights that relate the LSTM input to its output, while W_3 contains weights that relate the LSTM memory to its output.

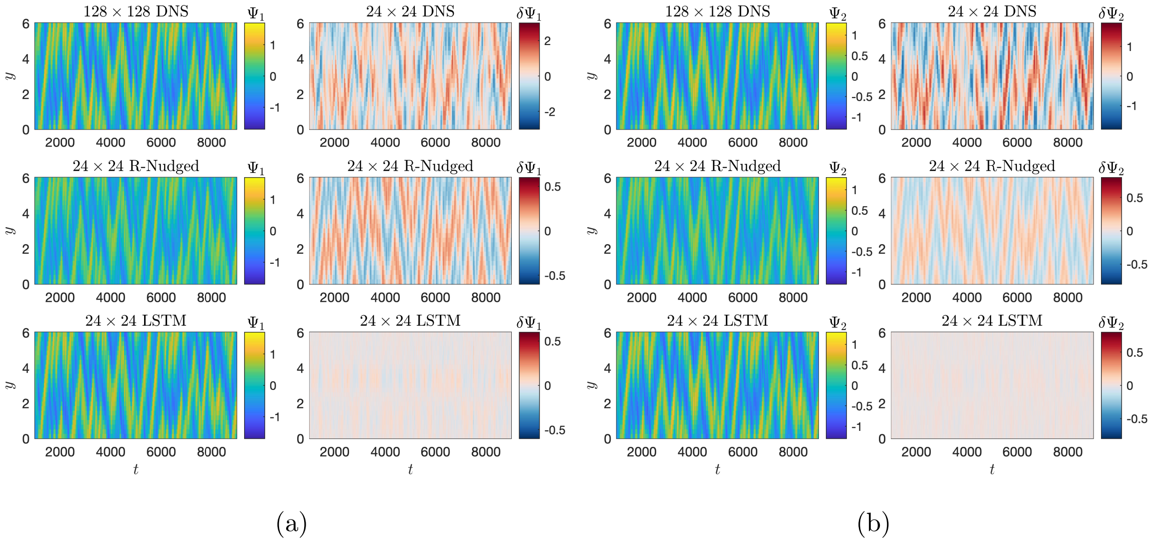


FIG. 6-9. Zonally-averaged time-series of (a) ψ_1; and (b) ψ_2. $\delta\Psi_1$ and $\delta\Psi_2$ correspond to biases of the predicted fields with respect to the reference 128 × 128 DNS data. Results are presented for $\beta = 2$ and $r = 0.1$.

As a next metric, the zonally-averaged bias of the streamfunctions with respect to the reference data is computed. Results are shown in fig. 6-9. The bias of the

LSTM predictions (while using R-Nudged data as input), is very low both in the training region $[1000, 4600]$ and the testing region $[4600, 9000]$. Therefore, we can conclude that the neural network generalizes well outside of its training regime.

Temporal Correlation Coefficient (TCC), $T = 100$								
Upper half					Lower half			
	$t_0 = 2000$		$t_0 = 4000$		$t_0 = 2000$		$t_0 = 4000$	
Model	ψ_1	ψ_2	ψ_1	ψ_2	ψ_1	ψ_2	ψ_1	ψ_2
24×24 DNS	0.09	0.09	0.45	0.52	0.01	0.04	0.40	0.49
Nudged 64	0.57	0.65	0.65	0.74	0.48	0.57	0.63	0.73
Nudged 16	0.87	0.92	0.81	0.87	0.83	0.87	0.84	0.89
Nudged 4	0.98	0.98	0.96	0.97	0.96	0.98	0.96	0.97
24×24 LSTM	0.96	0.97	0.99	0.99	0.97	0.96	0.98	0.98

Table 6.2. Temporal correlation coefficient for different simulations.

As an additional metric, the Temporal Correlation Coefficient (TCC) is computed for different cases. It is reminded, that the TCC, for a region $(x, y) \in [x_1, x_2] \times [y_1, y_2]$ and a time-interval $t \in [t_0, t_0 + T]$ is defined as:

$$\mathrm{TCC}\left(\psi_i; t_0\right) = \frac{1}{\Delta x \Delta y} \int_{x_1}^{x_2} \int_{y_1}^{y_2} \frac{\int_{t_0}^{t_0+T} \left(\psi_i^{\mathrm{ML}} - \overline{\psi_i^{\mathrm{ML}}}\right) \left(\psi_i^{\mathrm{ref}} - \overline{\psi_i^{\mathrm{ref}}}\right) \mathrm{d}t}{\sqrt{\int_{t_0}^{t_0+T} \left(\psi_i^{\mathrm{ML}} - \overline{\psi_i^{\mathrm{ML}}}\right)^2 \mathrm{d}t \int_{t_0}^{t_0+T} \left(\psi_i^{\mathrm{ref}} - \overline{\psi_i^{\mathrm{ref}}}\right)^2 \mathrm{d}t}} \mathrm{d}x \mathrm{d}y, \tag{6.17}$$

where $\Delta x = x_2 - x_1$, $\Delta y = y_2 - y_1$ and

$$\overline{\psi_i} = \frac{1}{\Delta x \Delta y T} \int_{x_1}^{x_2} \int_{y_1}^{y_2} \int_{t_0}^{t_0+T} \psi_i \mathrm{d}x \mathrm{d}y \mathrm{d}t. \tag{6.18}$$

The results are shown for the upper half and lower half of the domain, in table 6.2. Results are shown for both streamfunctions with time origins $t_0 = 2000, 4000$ and duration $T = 100$. In all cases, the LSTM trained with R-nudged data scores TCC values similar to a strongly nudged simulation with $\tau = 4$.

In addition, the Spatial Anomaly Correlation Coefficient (ACC) is studied. ACC for a region $(x, y) \in [x_1, x_2] \times [y_1, y_2]$ and a time-interval $t \in [t_0, t_0 + T]$ is defined as:

$$\mathrm{ACC}\left(\psi_i; t_0\right) = \frac{1}{T}\int_{t_0}^{t_0+T} \frac{\int_0^{L_x}\int_0^{L_y}\left(\psi_i^{\mathrm{ML}} - \overline{\psi_i^{\mathrm{ML}}}\right)\left(\psi_i^{\mathrm{ref}} - \overline{\psi_i^{\mathrm{ref}}}\right)\mathrm{d}x\mathrm{d}y}{\sqrt{\int_0^{L_x}\int_0^{L_y}\left(\psi_i^{\mathrm{ML}} - \overline{\psi_i^{\mathrm{ML}}}\right)^2\mathrm{d}x\mathrm{d}y\int_0^{L_x}\int_0^{L_y}\left(\psi_i^{\mathrm{ref}} - \overline{\psi_i^{\mathrm{ref}}}\right)^2\mathrm{d}x\mathrm{d}y}}\mathrm{d}t. \tag{6.19}$$

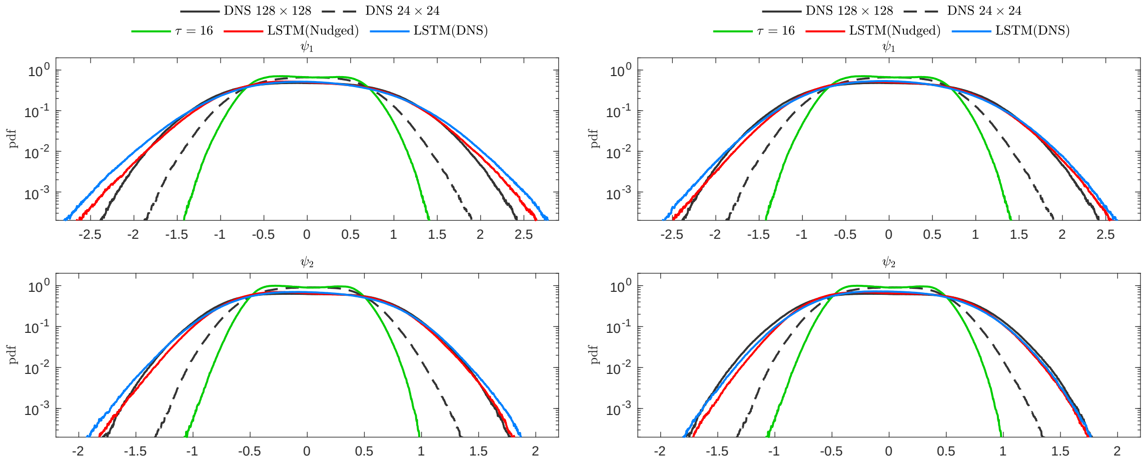
The results are shown for the upper half and lower half of the domain, in table 6.3. Results are shown for both streamfunctions with time origins $t_0 = 2000, 4000$ and duration $T = 100$. Similarly with TCC, the LSTM trained with R-nudged data scores ACC values similar to a strongly nudged simulation with $\tau = 4$.

Spatial Anomaly Correlation Coefficient (ACC), $T = 100$								
	Upper half				Lower half			
	$t_0 = 2000$		$t_0 = 4000$		$t_0 = 2000$		$t_0 = 4000$	
Model	ψ_1	ψ_2	ψ_1	ψ_2	ψ_1	ψ_2	ψ_1	ψ_2
24 × 24 DNS	0.06	0.09	0.49	0.61	-0.01	0.04	0.49	0.61
Nudged 64	0.54	0.65	0.70	0.81	0.48	0.59	0.69	0.80
Nudged 16	0.87	0.91	0.86	0.91	0.85	0.90	0.87	0.92
Nudged 4	0.97	0.98	0.97	0.98	0.97	0.98	0.97	0.98
24 × 24 LSTM	0.98	0.99	1.00	1.00	0.99	0.99	0.99	0.99

Table 6.3. Parameter values for different atmosphere regimes.

6.2.5 Statistical quantification

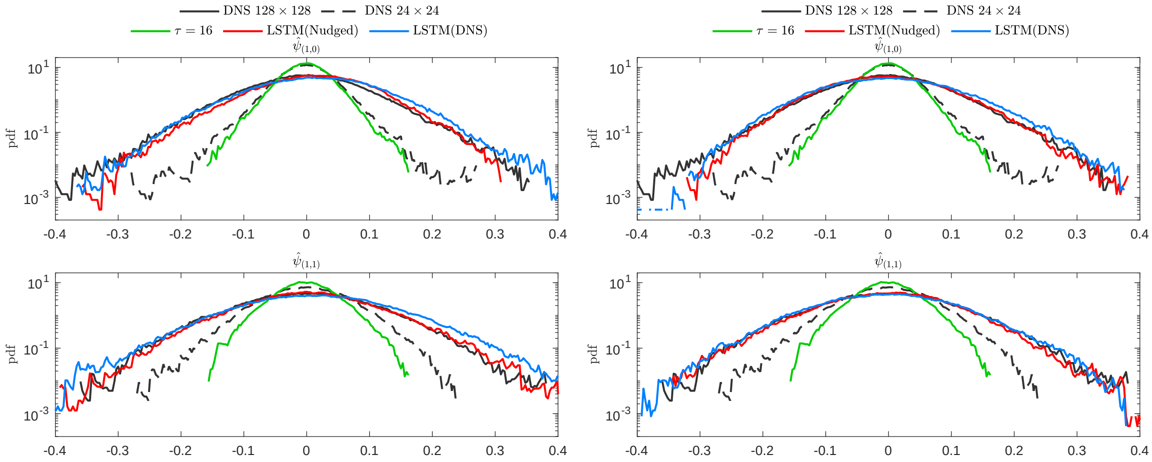
We now move to test the model on data not used during training. Specifically, the ability of the model to learn the statistics of the flows when free-running coarse-scale simulations are used as input is compared to when nudged data are used. The simulations are carried out until $T = 9000$. For training and testing, the time-interval $[1000, 9000]$ is kept where the flow has achieved statistical equilibrium. Input and reference data are normalized so that $\psi_1, \psi_2 \in [-1, 1]$. The neural network is trained for 2000 epochs. For training, the time-interval $[1000, 4600]$ is used and the interval $[4600, 5000]$ is used for validation. The data-size used during training here is conservative, to ensure that the model has enough data to learn from. R-nudged data are used as input data. Training takes place for a flow with $\beta = 2$ and $r = 0.1$.



(a)

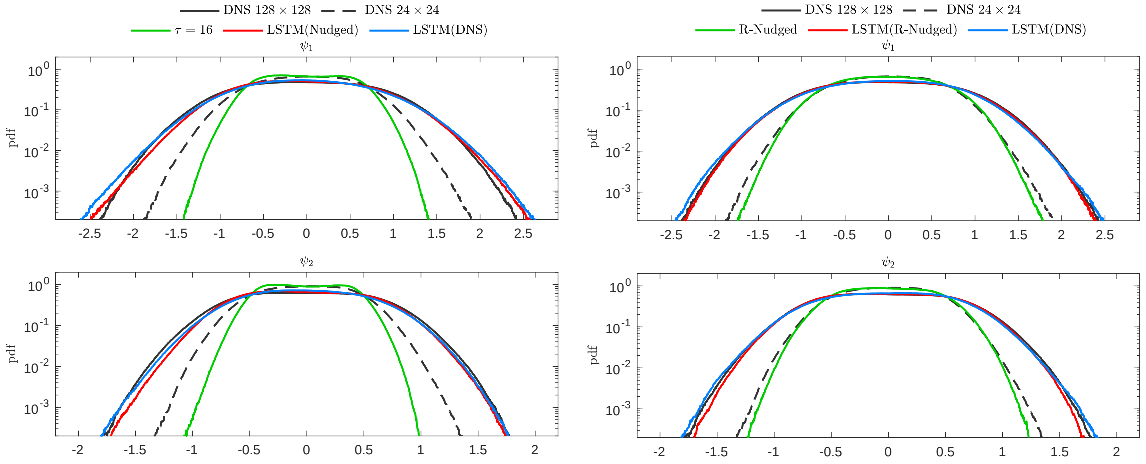
(b)

FIG. 6-10. Probability density function of ψ_1 and ψ_2 using as loss-function (a) the L^2-error eq. (6.11); and (b) the full loss function of eq. (6.15). Results are shown for $\beta = 2$ and $r = 0.1$.



(a) (b)

FIG. 6-11. Probability density function of the amplitude of wavenumbers $(1, 0)$, $(1, 1)$ of the barotropic streamfunction, using as loss-function (a) the L^2-error eq. (6.11); and (b) the full loss function of eq. (6.15). Results are shown for $\beta = 2$ and $r = 0.1$.



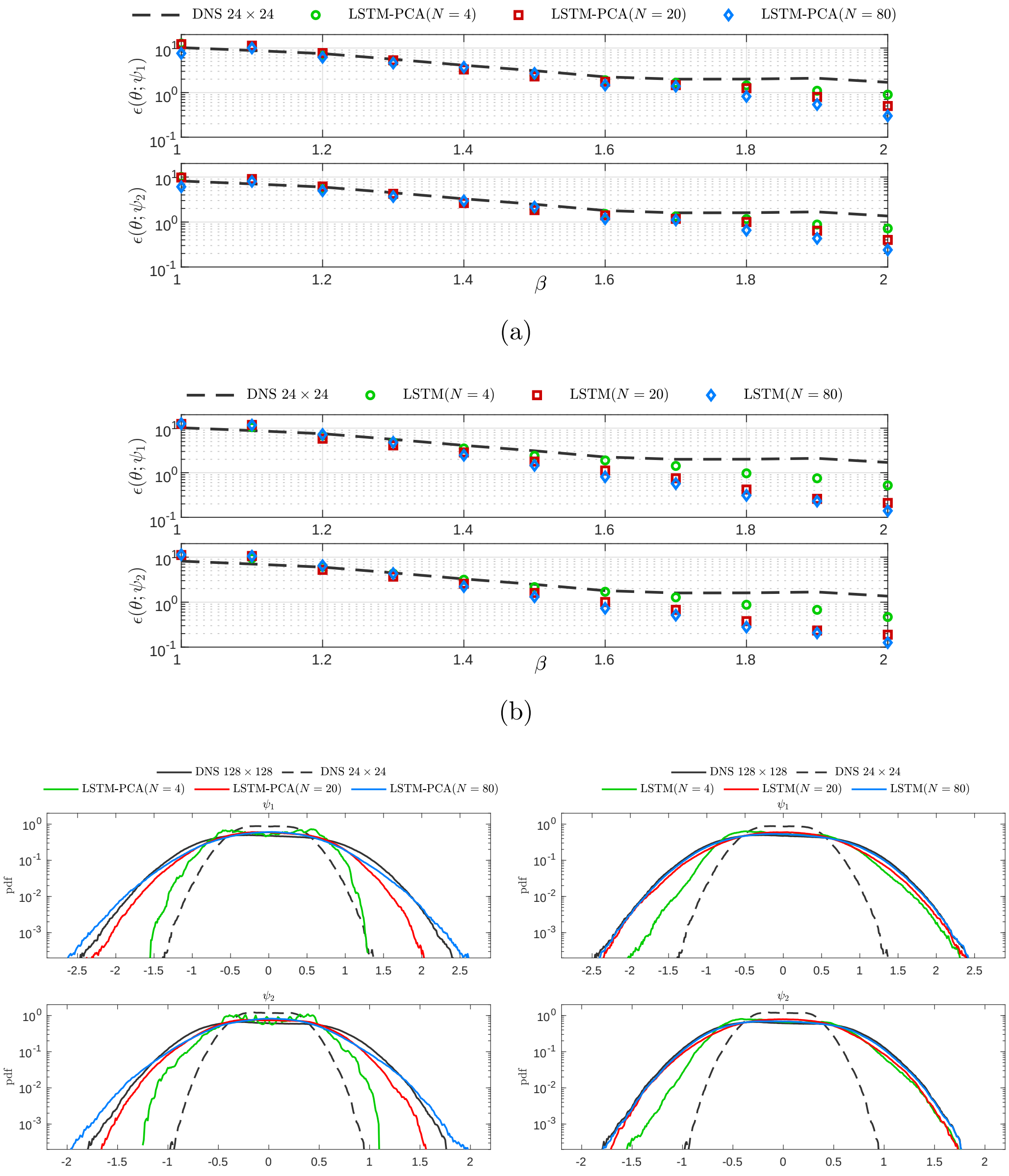
(a) (b)

FIG. 6-12. Probability density function of ψ_1 and ψ_2 using as training data (a) nudged-data; and (b) R-nudged data. Results are shown for $\beta = 2$ and $r = 0.1$.

The importance of using physical constraints can be further highlighted by looking at computed statistical quantities when using as loss function eq. (6.11) vs eq. (6.15). Results can be seen in fig. 6-10 and fig. 6-11. Both neural networks were trained with the same hyperparameters but with different loss-function. Training took place for 2000 epochs using nudged-data with $\tau = 16$ as input. The solid red line corresponds to predictions made with nudged data as input, while the blue line corresponds to predictions made with free-running coarse-scale simulations as input. In fig. 6-10 it is evident that using eq. (6.15) markedly improves the predictions of the tails of the pdf.

Further improvement in statistics is showcased in fig. 6-12. The predictions of the LSTM that was trained with R-nudged data, are significantly improved along the tails. Furthermore, the difference in the predicted pdf when using the training data and free-running data is smaller compared to the LSTM that was trained on nudged data. Furthermore, This is a result of the pdf of R-nudged data being very similar to that of the free running coarse data, allowing for a smoother transfer learning between data sets.

We now move to the numerical investigation of the hybrid approach for different β values. Training took place only at $\beta = 2$ and $r = 0.1$. For training, 10% of the training flow data were used, randomly sampled in time. Results are shown in fig. 6-13. Six different models are trained. First, hybrid models are trained on spatial data using 4, 20 and 80 hidden units. In the case of PCA modes, half of the modes are obtained from the top layer and the other half from the bottom layer. On the other side, for the physical data case, modes can incorporate cross-layer information. The logarithm pdf error for each model are shown and compared to free-running coarse scale simulations, at (a) for LSTM-PCA modesl and (b) for the LSTM architecture with two fully connected layers.



(a)

(b)

(c)

(d)

FIG. 6-13. (a) Logarithm pdf error of streamfunction for LSTM trained on (a) PCA (b) physical data, as a function β. Training took place at $\beta = 2.0$. Probability density function of ψ_1 and ψ_2 $\beta = 2.0$ for (c) PCA input data and (d) physical input data. Training took place at $\beta = 2$ and $r = 0.1$. N corresponds to the dimension of the latent space the LSTM works on in both cases.

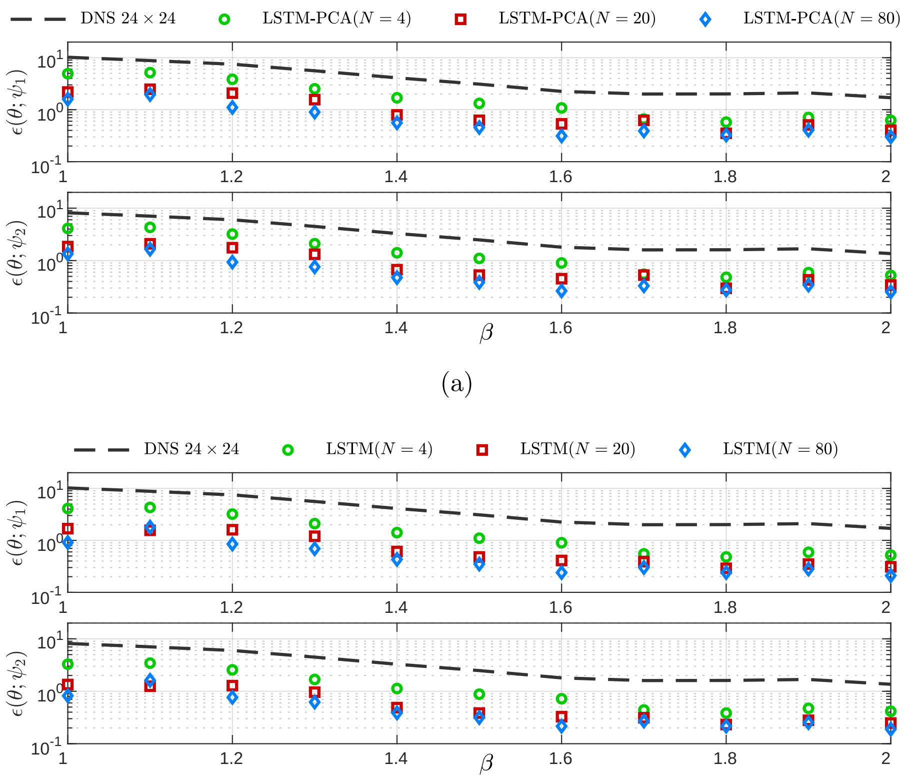
In both cases, increasing the dimension of the latent space improves the accuracy of the predictions. Furthermore, error is low only for β values close to the training point $\beta = 2.0$. Finally, LSTM models seem to outperform LSTM-PCA models. To test the accuracy of the predictions, a logarithmic pdf error is defined as

$$\epsilon\left(\theta; \psi_i\right) = \int \left|\max\left\{-5, \log_{10} p^{\text{ref}}_{\psi_i}(s;\theta)\right\}\right| - \left|\max\left\{-5, \log_{10} p^{\text{ML}}_{\psi_i}(s;\theta)\right\}\right| \mathrm{d}s \tag{6.20}$$

Note that this measure includes the difference of the logarithm of the predicted pdfs and thus is a very strict measure. For a closure look at the model predictions, subfigure (c) includes a pdf comparison between all LSTM-PCA models and reference and coarse data and (d) includes a comparison between all LSTM architectures, with fully connected layers for projection, and reference and coarse data, for $\beta = 2.0$.

For the second numerical investigation, both schemes are trained on flow data with $\beta = \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$. 10% of the time series for each flow parameter is used during training. Results regarding the logarithm pdf error for each β value are shown in fig. 6-14. For all cases, the LSTM-PCA underperforms with respect to the architecture with physical input.

A few justification can be given for this behaviour. First, the PCA modes were taken independently for each layer. However, the flow undergoes a baroclinic instability, and thus cross-layer information is important to correctly capture the statistics. As a result, layer-by-layer PCA projection will not be the most efficient way to compress data and thus suffer when contrasted with cross-layer projections of the same latent-space dimension. Furthermore, using different projection schemes for the input and output of the data, allows for a more efficient representation of the two datasets. The ability of the input and output fully-connected layers to represented different projections, may have added to the increased accuracy of that scheme.



(a)

(b)

FIG. 6-14. (a) Logarithm pdf error eq. (6.20) of streamfunction for LSTM trained on (a) PCA (b) physical data, as a function β. Training took place at $\beta = 1.0, 1.2, 1.4, 1.6, 1.8, 2.0$.

6.3 Application to climate data

A reference dataset is selected for testing the effectiveness of the scheme, the ERA5 dataset [62]. It is a reanalysis dataset of hourly global atmospheric data, on a 30[km] horizontal resolution, produced by the Copernicus Climate Change Service (C3S). For the generation of coarse-scale climate data, the atmospheric component of the Energy Exascale Earth System Model (E3SM) is used. In particular, version

2 of the E3SM Atmospheric Model (EAMv2) [37, 169, 49]. Appropriate boundary conditions over the Earth's surface are imposed, prescribing the sea-surface temperature (SST) [122]. Simulations are run on an unstructured grid of approximately $1^o(\sim 110[\mathrm{km}])$ resolution per sigma-level and 72 levels along the vertical direction. The vertical levels extend from the Earth surface up to altitude of about 64[km], corresponding to $\sim 0.1[\mathrm{hPa}]$. The evolution model equations for the coarse-scale model have the form

$$\frac{\partial \mathbf{X}_c}{\partial t} = \mathcal{D}\left(\mathbf{X}_c\right) + \mathcal{P}\left(\mathbf{X}_c\right), \tag{6.21}$$

where $\mathbf{X}_c = (U, V, T, Q)$ represent the set of coarsely-resolved system variables, $\mathcal{D}$ is the operator containing the dynamics of the system [189, 48] and $\mathcal{P}$ is the operator concerning the physics of the system [117, 91, 115]. Variables (U, V) correspond the zonal and meridional components of wind velocity, T is wind temperature and Q is specific humidity. The vertical velocity component W is derived from (U, V) in E3SM and thus is ignored as input. From here on now, the coarse-scale free-running dataset will be labeled as CLIM and will be denoted as $\mathbf{X}_c$.

Given the two previous observations, it is clear that it is not possible to use a dataset of CLIM and one of ERA5 and try to machine learn a map between the two, i.e. a map that takes as input a CLIM timeseries and produces as output an ERA5 timeseries. To eliminate the problematic component, i.e. chaos-induced divergence we design a new CLIM dataset (we call it nudged CLIM and denote it as $\mathbf{X}_{c,n}$) that is preserving the long-time statistics of the original CLIM dataset but does not suffer from the chaos-induced divergence with the ERA5. This is done by employing the concept of nudging that has been used extensively in the context of data assimilation. Specifically, we utilize the EAMv2 solver (generator of the CLIM dataset) with an extra term, the nudging term, that is ‘pulling’ the CLIM solution close to the ERA5 solution:

$$\frac{\partial \mathbf{X}_{c,n}}{\partial t} = \mathcal{D}\left(\mathbf{X}_{c,n}\right) + \mathcal{P}\left(\mathbf{X}_{c,n}\right) + \mathcal{N}, \tag{6.22}$$

where the relaxation term $\mathcal{N}$ is called nudging tendency and it corrects the coarse-scale solution based on the ERA5 reference solution. In this study, the nudging tendency $\mathcal{N}$ is given by the algebraic term

$$\mathcal{N}\left(\mathbf{X}_{c,n} - \mathbf{X}^{\mathrm{ref}}\right) = -\frac{1}{\tau}\left(\mathbf{X}_{c,n} - \mathcal{H}\left[\mathbf{X}^{\mathrm{ref}}\right]\right). \tag{6.23}$$

Parameter τ is a relaxation timescale that has a large value (so that $1/\tau$ is small compared with the other tems in the equation), and $\mathcal{H}$ is an operator that maps $\mathbf{X}^{\mathrm{ref}}$ to the coarse resolution.

6.3.1 Spectral Correction of Nudged Dataset

As described in the previous subsection, the nudged dataset $\mathbf{X}_{c,n}$ is used during training to eliminate discrepancies due to chaotic divergence between input data and ERA5 reference data. The corrected nudged data is termed as $\mathbf{X}_{c,rsn}$ and defined as

$$\mathbf{X}_{c,rsn}\left(\phi, \theta, t; k\right) = \sum_{m,n} R_{m,n}\{\hat{\mathbf{X}}_c(t)\}_{m,n} e^{i(m\phi + n\theta)},, \tag{6.24}$$

where $\{\hat{\mathbf{X}}_c(t)\}_{m,n}$ are the spatial Fourier coefficients of $\mathbf{X}_{c,n}$ and

$$R_{k,l} = \sqrt{\frac{\mathcal{E}_{k,l}^{\mathrm{coarse}}}{\mathcal{E}_{k,l}^{\mathrm{nudge}}}}, \tag{6.25}$$

and

$$\mathcal{E}_{k,l} = \frac{1}{T}\int_0^T \hat{E}_{k,l}(t)\mathrm{d}t = \frac{1}{T}\int_0^T |\{\hat{\mathbf{X}}_c(t)\}_{k,l}|^2\mathrm{d}t. \tag{6.26}$$

6.3.2 Neural Network Architecture

In the current implementation, training is done on a layer-by-layer basis. A schematic of the configuration for training on a particular layer is shown in fig. 6-15. The model receives as input the predictive variables $\mathbf{X} = \mathbf{X}(\phi, \theta, t; k)$, where ϕ is the longitudinal angle and θ the latitudinal one. Snapshots of the entire horizontal discretization of the layer are used. Afterwards, a custom "split" layer separates the input into non-overlapping subregions. These subregions are periodically padded via a custom padding process, tasked with respecting the spherical periodicity of the domain. Then, each subregion is independently passed through a series of convolutional layers. The purpose of this process is to extract anisotropic local features in each subregion.

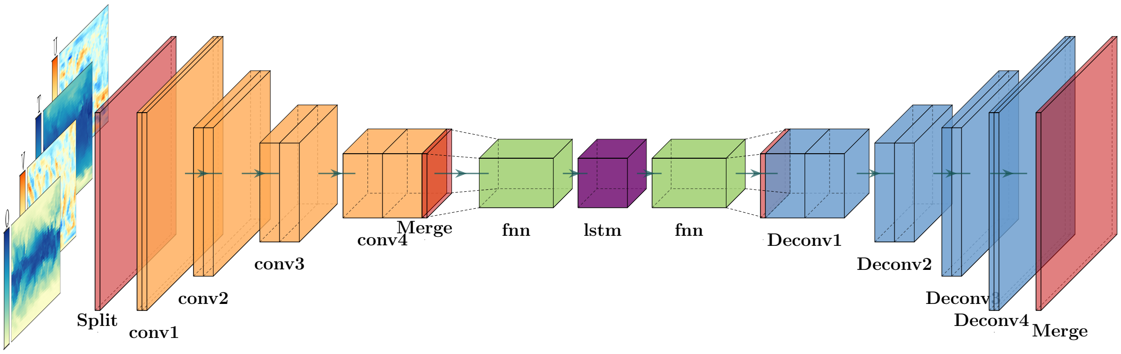


FIG. 6-15. Neural network architecture of the non-instrusive model for a training on a particular sigma-level.

Afterwards, the local information extracted from each subregion is concatenated in a single vector via a custom ‘merge’ layer. The global information is now passed through a linear fully-connected layer, that acts as a basis projection of the spatial data onto a reduced-order latent space. The latent space data are then corrected by a long short-term memory (LSTM) layer [64]. Subsequently they are projected back to physical space via another linear fully-connected layer. Afterwards, global information is split into the same subregions of the input, and distributed to a series of independent deconvolution layers that upscale the data to the original resolution. Finally, a custom ‘merge’ layer gathers the information from each subregion and produces the final corrected snapshot.

In addition to temporal nonlocality, the model is nonlocal in space. Note, that in terms of the LSTM layer, this information comes in the form of the latent space coefficients, which in general correspond to global modes that correspond to rows of the fully connected layer’s matrix. Under the assumption that both fully-connected layers have linear activation functions, the model can be mathematically depicted as a basis projection. Hence, the fully connected layers act as projection schemes to (a) compress input data to a latent space of low dimensionality, and (b) project the LSTM prediction to physical space.

The used loss function is a standard mean-square error (MSE) loss

$$\mathcal{L} = \alpha \sum_t \sum_\phi \sum_\theta \cos\left(2\pi \frac{\theta}{360}\right) \|\hat{\mathbf{X}} - \mathbf{X}^{\text{ref}}\|^2, \tag{6.27}$$

where α is a normalization coefficient. Each term in the sum is multiplied by the cosine of its latitude to approximate integration over a sphere. Without this term, the model would over-emphasize on learning the corrections at the poles.

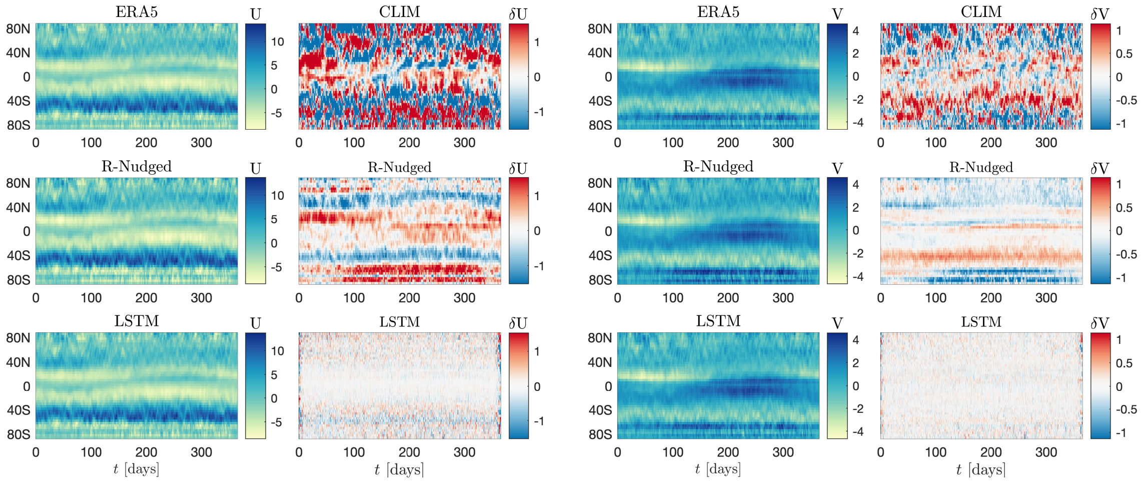
6.3.3 Data assimilation

The numerical exploration of the proposed method begins with some validation results. Specifically, nudged data not used for training are used as input, with the purpose of checking whether the produced output has good accuracy. Training takes places over 1000 epochs and eq. (6.27) is used as the loss function. In fig. 6-16, results regarding zonally averaged predictions of zonal and meridional velocities U and V respectively, are shown. The left column of subfigure (a) displays the zonally averaged predictions of zonal velocity U, for ERA5 reanalysis data, Nudged data and neural network predictions. The right column displays the biases compared to ERA5 predictions for CLIM, Nudged and the neural network predictions. Similarly, the left column of subfigure (b) displays the zonally averaged predictions for meridional velocity V, with the same biases displayed. Results are shown for sigma-level 71, i.e. the one closest to the surface of the earth. The results show that the neural network clearly is able to learn a correction for both velocity components.

Now, the model is tested on unseen free-running and coarse-scale climate models. This dataset (CLIM) is not nudged and thus does not include any ERA5 information. Since the free-running dataset diverges from ERA5 in terms of phases, specific extreme events cannot be studied, as they are absent from CLIM. Hence, long-time statistics are studied now.

In fig. 6-17, the predicted probability density functions (pdf) are shown for the four different predictive variables, (U, V, T, Q). Solid black lines correspond to ERA5 data and dashed black lines correspond to CLIM. Blue lines correspond to neural network predictions using CLIM data as input (i.e. testing data). Results are shown for sigma-level closest to the earth's surface. Data are averaged over the time-period 2007-2017.

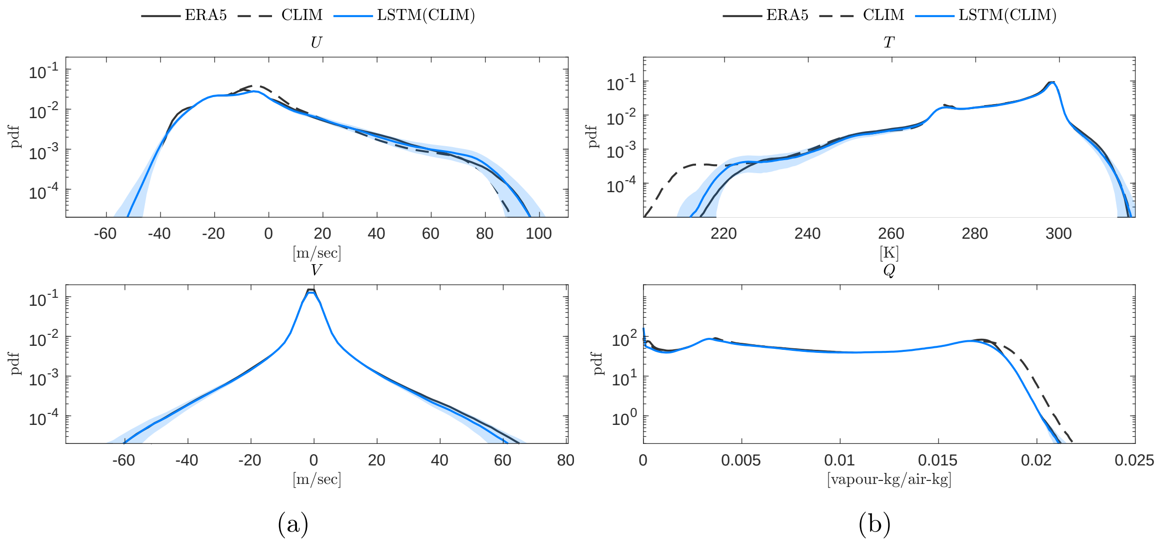
The predictions of the neural network that was trained with Nudged data, significantly improve the prediction of the tails. This can be seen both for the pdfs of the well-predicted quantities (U, V) by CLIM as well as for (T, Q), two quantities whose tails are not well predicted by CLIM. These results showcase the ability of the model to generalize beyond training data (Nudged data). This is a result of the pdf of the implemented spectral corrections to the Nudged data, making them display very similar statistics to that of the free running coarse data. This property allows for smoother transfer learning between data sets. A more thorough validation of the climate dataset is presented in appendix B.



(a)

(b)

FIG. 6-16. (a) Zonally averaged predictions for U (left column) and biases with respect to ERA5 predictions (right column). (b) Zonally averaged predictions for V (left column) and biases with respect to ERA5 predictions (right column).



(a) (b)

FIG. 6-17. Predicted pdfs for (a) horizontal velocity components U (top) and V (bottom); and (b) for temperature T (top) and specific humidity Q (bottom). Solid black lines correspond to ERA5 data, dashed black lines correspond to CLIM and blue lines correspond to neural network predictions using CLIM data as input (i.e. testing data). The blue envelope corresponds to a 95% confidence interval in the calculation of the LSTM-corrected pdf.

6.3.4 Statistical quantification

We now move to predict statistics for a derived integral quantity, in particular, mean integrated vapor transport (IVT) over the period 2007-2017. Since IVT is strongly anisotropic, extracting local features, and especially atmospheric rivers, is vital. Therefore the local convolutions derived from each subregion is important for its correction estimation. 25 subregions are used in this numerical test, using different convolutions on each one (i.e. extract local features).

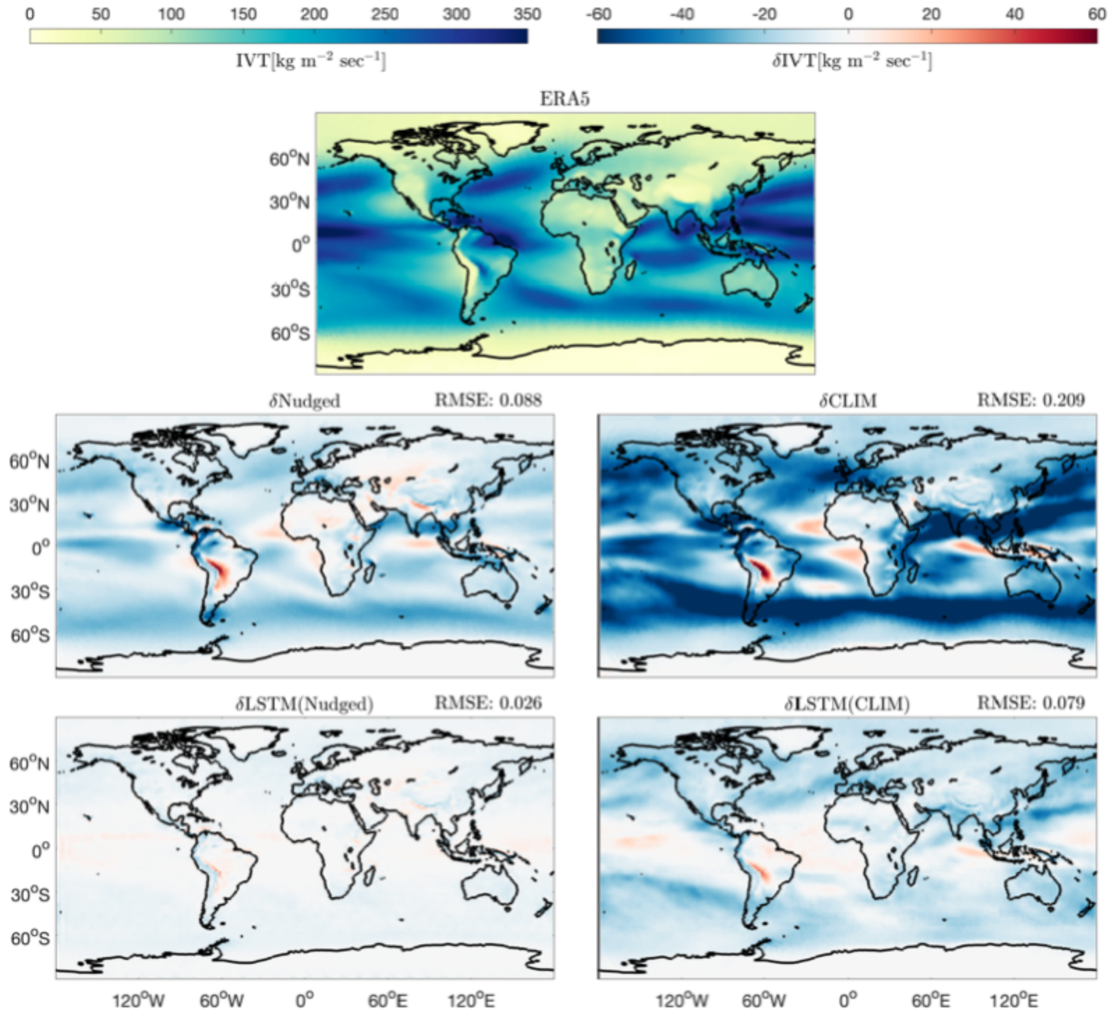


FIG. 6-18. IVT predictions averaged over the period 2007-2017. Mean IVT predictions are shown in the top row for ERA5 data. Biases from ERA5 predictions are shown for, Nudged datasets and CLIM free-running datasets, together with corrected results via our non-intrusive approach. A 1-subregion partition was used for these results.

In fig. 6-18, the first row corresponds to ERA5 predictions. The other rows correspond to biases with respect to ERA5. As expected, the neural network exhibits very small biases when Nudged data are used as input. This is expected since these data were used for training. However, the model significantly decreases biases even when CLIM data are used as input. In addition, it is able to decrease the root mean-squared error (RMSE) below that of the Nudged dataset, a simulation that exploits ERA5 information at every time-step.

The final numerical result involves the statistics of tropical cyclones for the time-period 2007-2017. For tracking tropical cyclones, the software package TempestExtreme is used [172, 173]. The following steps are used to track cyclones: (i) Find local minima of sea-level pressure (SLP). (ii) Eliminate smaller minima within 2 great-circle distance (gcd) degrees. (iii) Check that SLP raises by 200 [Pa] within 8 gcd. (iv) Check that temperature at 400 [mbar] drops by 0.4 [K] within 8 gcd. (v) Check that velocity is higher than 10 [m/sec] for 8 snapshots. (vi) Check that geopotential height is larger than 100 for at least 8 snapshots. (vii) Check that phenomenon lasts at least 54 hours, with a max gap of 24 hours.

During training, the neural network will track dissipated cyclonic structures in the Nudged simulations, that correspond to tropical cyclones in ERA5. It will then amplify them allowing them to be recognized as tropical cyclones. As a result, if the cyclonic structures present in CLIM are more dissipated than in Nudged, the learned mapping will not be transferable. To overcome this issue, we post process the SLP values of Nudged data. To that end the following conditional means are computed

$$c(\phi,\theta) = \frac{1}{\mathrm{TC}(\phi,\theta)} \sum_{t=1}^{N_t} \delta SLP(t,\phi,\theta), \tag{6.28}$$

where TC is the tropical cyclone density over the globe.

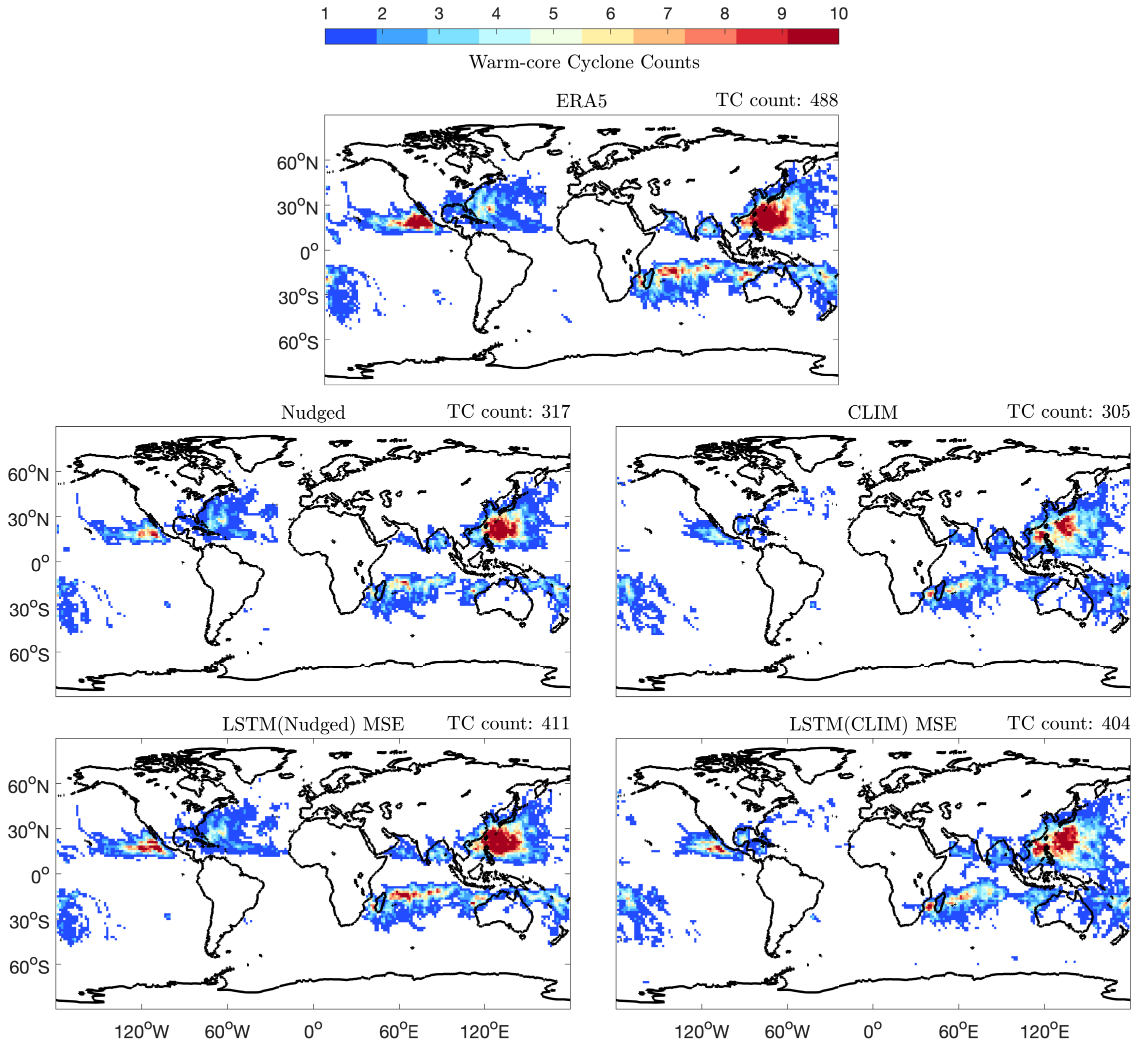


FIG. 6-19. Tropical cyclone counts over the period 2007-2017. Results are derived using ERA5 datasets, CLIM free-running datasets and CLIM datasets corrected via our non-intrusive LSTM approach. Cyclones are tracked via the TempestExtremes software.

Then, the following spatially dependent coefficient is computed

$$R(\phi,\theta) = \frac{c^{\mathrm{c}}(\phi,\theta)}{c^{\mathrm{c,n}}(\phi,\theta)}. \tag{6.29}$$

Finally, the SLP Nudged data are corrected as follows

$$SLP^{\mathrm{c,rnsn}} = \begin{cases} R(\phi,\theta)SLP^{\mathrm{c,n}} + P_0, & SLP < P_0 \\ SLP^{\mathrm{c,n}}, & SLP \geq P_0 \end{cases}. \tag{6.30}$$

Results are shown in fig. 6-19. In terms of total number of tropical cyclones predicted, the neural network corrections are much better than the ones predicted by the Nudged dataset and CLIM. Nudged data predict 317 tropical cyclones, while CLIM predicts 305. On the other hand, the neural network corrections predict 411 tropical cyclones when using Nudged data and 404 tropical cyclones when using CLIM data. These predictions are much closer to the 488 cyclones predicted from ERA5 data. By focusing on the predictions when using CLIM as input, we notice that barely any new cyclones are predicted in the Atlantic. Looking further into this issue it is our belief that this is a problem of CLIM, where it does not generate enough vorticity over the Atlantic for tropical cyclones to form.

Chapter 7

Hybrid Quadrature-based method of moments

Abstract

The focus of this chapter, based on [28], is predicting the dynamics of a bubble cluster as it deforms within a fluid. Solving the population balance equation (PBE) for the dynamics of a dispersed phase coupled to a continuous fluid is expensive. Still, one can reduce the cost by representing the evolving particle density function in terms of its moments. In particular, quadrature-based moment methods (QBMMs) invert these moments with a quadrature rule, approximating the required statistics. QBMMs have been shown to accurately model sprays and soot with a relatively compact set of moments. However, significantly non-Gaussian processes such as bubble dynamics lead to numerical instabilities when extending their moment sets accordingly. We solve this problem by training a recurrent neural network (RNN) that adjusts the QBMM quadrature to evaluate unclosed moments with higher accuracy. The proposed method is tested on a simple model of bubbles oscillating in response to a temporally fluctuating pressure field. The approach decreases model-form error by a factor of 10 when compared to traditional QBMMs. It is both numerically stable and computationally efficient since it does not expand the baseline moment set. Additional quadrature points are also assessed, optimally placed and weighted according to an additional RNN. These points further decrease the error at low cost since the moment set is again unchanged.

7.1 Problem formulation

7.1.1 Ensemble-averaged flow equations

This work focuses on the fluid-coupled dynamics of a dispersion of small, spherical bubbles transported in a compressible carrier fluid. The mixture phase-averaged evolution equations for the continuous fluid are

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}) &= 0, \\ \frac{\partial \rho \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}\boldsymbol{u} + p\boldsymbol{I}) &= 0, \\ \frac{\partial E}{\partial t} + \boldsymbol{\nabla} \cdot (E + p)\boldsymbol{u} &= 0, \end{aligned} \tag{7.1}$$

with $\rho, \boldsymbol{u}, p$, and E being the mixture density, velocity vector, pressure, and total energy, respectively. The system of equations is complemented by appropriate initial and boundary or radiation conditions specific to each individual problem. The void fraction of the bubbles is α and a dilute assumption $\alpha \ll 1$ is made. The bubbles are defined by their instantaneous bubble radii R, its time derivative $\dot{R}$. The bubbles are assumed monodisperse and so have the same equilibrium radius R_o.

The mixture pressure p is deduced from the ensemble phase-averaging method [188, 24] as

$$p = (1 - \alpha) p_l + \alpha \left(\frac{\overline{R^3 p_{bw}}}{\overline{R^3}} - \rho \frac{\overline{R^3 \dot{R}^2}}{\overline{R^3}} \right), \tag{7.2}$$

where p_{bw} and p_l are the bubble wall and liquid pressures, respectively [5]. Liquid pressure p_l follows from the stiffened-gas equation of state [112], though this model can be substituted for another if required. The usual coefficients for water are used [24].

The overbars in 7.2 denote raw moments μ of the bubble dispersion as

$$\overline{R^i \dot{R}^j} = \mu_{i,j} = \int_\Omega R^i \dot{R}^j f(R, \dot{R}; R_o) \mathrm{d}R \mathrm{d}\dot{R}, \tag{7.3}$$

where f is the number density function of the bubbles. This paper focused on a new, improved method for computing these moments, which will be introduced in section 7.2.

The void fraction transports as [24]:

$$\frac{\partial \alpha}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \alpha = 3\alpha \frac{\overline{R^2 \dot{R}}}{\overline{R^3}}. \tag{7.4}$$

The moments required to close the governing flow equations are thus

$$\boldsymbol{\mu}_{\text{Targ.}} = \left\{ \overline{R^3 \dot{R}^2}, \overline{R^3}, \overline{R^2 \dot{R}}, \overline{R^3 p_{bw}} \right\}. \tag{7.5}$$

7.1.2 Bubble model

To close the governing equations of the previous subsection, a model for the bubble dynamics, in terms of the dynamical variables R and $\dot{R}$, is required. We use a Rayleigh–Plesset equation for this:

$$R\ddot{R} + \frac{3}{2}\dot{R}^2 + \frac{4}{\mathrm{Re}} \frac{\dot{R}}{R} = \left(\frac{R_o}{R} \right)^{3\gamma} - \frac{1}{C_p}, \tag{7.6}$$

which is dimensionless via the reference bubble size R_o, liquid pressure p_0, and liquid density ρ_0. In (7.6), C_p is the ratio between the fluid and bubble pressures and Re is a Reynolds number

$$\mathrm{Re} = \sqrt{\frac{p_0}{\rho_0}} \frac{R_o}{\nu_0}, \tag{7.7}$$

where ν_0 is the liquid kinematic viscosity. For the cases considered here $\mathrm{Re} = 10^3$.

This model assumes the bubbles remain spherical and compress via a polytropic process with coefficient $\gamma = 1.4$. While this model can be generalized to include heat exchange and liquid compressibility, these effects are not critical to our study and thus omitted here. Based on this model, the bubble wall pressure p_{bw} simplifies the last moment of $\boldsymbol{\mu}_{\mathrm{Targ.}}$ as

$$\overline{R^3 p_{bw}} = \mu_{3(1-\gamma),0}. \tag{7.8}$$

We also define a dimensionless time $t^* = t\omega_0$, where ω_0 is the natural frequency of the bubbles. To simplify the notation, t will be used in place of t^* hereon.

7.1.3 Population balance formulation

A number density function f describes the statistics of the bubbles. The generalized population balance equation is

$$\frac{\partial f}{\partial t} + \frac{\partial}{\partial R}(f\dot{R}) + \frac{\partial}{\partial \dot{R}}(f\ddot{R}) = 0, \tag{7.9}$$

assuming the bubbles do not coalesce or break up, though these effects can be included via empirically modeled terms if desired. QBMMs solve (7.9) by representing f as a set of raw moments $\boldsymbol{\mu}$ [40, 23]. Through an appropriate inversion procedure, these methods can transform these moments into quadrature nodes and weights in phase space. This allows for the approximation of f via a weighted sum of Dirac delta functions. Hence, the following quadrature rule can approximate any raw moment

$$\mu_{i,j} = \sum_k w_k \xi_{1,k}^i \xi_{2,k}^j, \tag{7.10}$$

where $\xi_{i,j}$ is the j-th quadrature point locations for the i-th internal coordinate.

7.2 Hybrid quadrature method moment formulation

We now present the hybrid, data-informed method for predicting the moments of cavitating bubbles. Subsection 7.2.1 presents 4-node CHyQMOM predictions [23]. Subsection 7.2.2 details the hybrid neural-network model that improves the predictions.

7.2.1 CHyQMOM

For two-dynamic-variable cases, conditioned moment methods are computationally preferable to traditional QMOM [187]. We use CHyQMOM because it can close a second-order moment system with fewer carried moments than CQMOM [40]. For a 4-node quadrature rule, it uses the first- and second-order moments

$$\boldsymbol{\mu} = \{\mu_{1,0}, \mu_{0,1}, \mu_{2,0}, \mu_{1,1}, \mu_{0,2}\}, \tag{7.11}$$

in tandem with a Gaussian closure assumption for the third-order moments $\mu_{3,0}, \mu_{0,3}$. The particular chosen scheme displayed similar fidelity with more involved quadrature schemes [187] for the bubble dispersions discussed here. The CHyQMOM inversion process for obtaining the nodes $\boldsymbol{\xi}$ and weights $\boldsymbol{w}$ is presented in appendix C.1.

Taking the time-derivative of each of the (7.11) moments and applying (7.6) results in

$$
\begin{aligned}
\frac{\partial \mu_{1,0}}{\partial t} &= \mu_{0,1}, \\
\frac{\partial \mu_{0,1}}{\partial t} &= -\frac{3}{2}\mu_{-1,2} - \frac{4}{\mathrm{Re}}\mu_{-2,1} + \mu_{-4,0} - C_p\mu_{-1,0}, \\
\frac{\partial \mu_{2,0}}{\partial t} &= 2\mu_{1,1}, \\
\frac{\partial \mu_{1,1}}{\partial t} &= -\frac{1}{2}\mu_{0,2} - \frac{4}{\mathrm{Re}}\mu_{-1,1} + \mu_{-3,0} - C_p, \\
\frac{\partial \mu_{0,2}}{\partial t} &= -3\mu_{-1,3} - \frac{8}{\mathrm{Re}}\mu_{-2,2} + 2\mu_{-4,1} - 2C_p\mu_{-1,1},
\end{aligned}
\tag{7.12}
$$

which are called the moment transport equations. The quadrature rule (7.10) approximates unclosed moments in (7.12).

While this scheme is computationally cheap, it is challenging to extend to include additional quadrature points without potential numerical instabilities or need to decrease the time-step [107]. Thus, truncation errors can affect approximation of the right-hand-side of (7.12) and the extrapolation out of the low-order moment space to the moments of $\boldsymbol{\mu}_{\mathrm{Targ.}}$ of (7.5). We next present an augmented method that treats these issues without introducing numerical instability or high computational cost.

7.2.2 Data-informed corrections

We improve the CHyQMOM moment inversion procedure by adding a correction term to the 4-node quadrature rule and introducing additional quadrature nodes. The unaugmented CHyQMOM quadrature rule is denoted via $\{\boldsymbol{w}^{(\mathrm{QBMM})}, \boldsymbol{\xi}^{(\mathrm{QBMM})}\}$.

For these corrections, a long short-term memory (LSTM) RNN is employed. The LSTM incorporates non-Markovian memory effects into the reduced-order model. This approach is known to be capable of improving predictions of reduced-order models [178, 21].

The corrections $\{\boldsymbol{w}', \boldsymbol{\xi}'\}$ serve as input predictions for the first- and second-order moments as well as the pressure $\{\mu_{1,0}, \mu_{0,1}, \mu_{2,0}, \mu_{1,1}, \mu_{0,2}, C_p\}$. They are modelled as

$$\{\boldsymbol{w}'(t), \boldsymbol{\xi}'(t)\} = \mathbb{G}[\Theta; \boldsymbol{\mu}(\chi(t)), C_p(\chi(t)), \mathrm{Re}], \tag{7.13}$$

where $\mathbb{G}$ denotes the functional representation of the employed neural networks, the vector Θ denotes hyperparameters and optimized parameters of the neural network as obtained during training. More detail on the implementation is in section 7.3, subsection 7.3.1. The chosen values for the hyperparameters are included in appendix C.2. The history of the reduced-order model states is

$$\chi(t) = \{t, t - \tau_1, ..., t - \tau_N\}. \tag{7.14}$$

The hybrid quadrature rule follows as

$$\boldsymbol{w} = \boldsymbol{w}^{(\mathrm{QBMM})} + \boldsymbol{w}' \quad \text{and} \quad \boldsymbol{\xi} = \boldsymbol{\xi}^{(\mathrm{QBMM})} + \boldsymbol{\xi}'. \tag{7.15}$$

The neural network loss function is designed to ensure the target high-order moments $\boldsymbol{\mu}_T$ can be accurately computed and that the low-order moments $\boldsymbol{\mu}$ evolve accurately. Hence, the right-hand-side of (7.12) is included in the loss function as

$$\begin{aligned}\mathcal{L} = \sum_{0 \le i,j \le 2} \alpha_{i,j} \left(\frac{\partial \mu_{i,j}^{(\mathrm{ML})}}{\partial t} - \frac{\partial \mu_{i,j}^{(\mathrm{MC})}}{\partial t} \right)^2 + \sum_{(i,j) \in \mathcal{I}} \beta_{i,j} \left(\sum_k w_k \xi_{1,k}^i \xi_{2,k}^j - \mu_{i,j}^{(\mathrm{MC})} \right)^2 \\ + \lambda \sum_k \mathrm{Relu}(-w_k),\end{aligned} \tag{7.16}$$

where,

$$\mathcal{I} = \{(0,0),(1,0),(0,1),(2,0),(1,1),(0,2),(3,0),(2,1),(3,2),(3-3\gamma,0)\}, \quad (7.17)$$

and

$$\alpha_{i,j} = \left\| \frac{\partial \mu_{i,j}^{(\mathrm{MC})}}{\partial t} \right\|_{\infty}^{-1}, \quad \beta_{i,j} = \left\| \mu_{i,j}^{(\mathrm{MC})} \right\|_{\infty}^{-1}. \quad (7.18)$$

The variables $\left\{\mu_{i,j}^{(\mathrm{ML})}, \mu_{i,j}^{(\mathrm{MC})}\right\}$ correspond to moment $\mu_{i,j}$ as predicted by the proposed hybrid approach and by Monte Carlo simulations respectively. The first term in (7.16) minimizes, in the L^2 sense, the right-hand-sides of (7.12) (given $\boldsymbol{\mu}$). The second term in (7.16) minimizes prediction error for both $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_{\mathrm{Targ.}}$, while it also penalizes the network when the weights don't sum up to 1 (under the assumption that $\mu_{0,0} = 1$). The last term in (7.16) penalizes negative-valued weights.

The discretized moment transport equation (7.12) and the quadrature rule (7.10) compute the time-derivatives $\partial \mu_{i,j}^{(\mathrm{ML})}/\partial t$ required in (7.16) as

$$\begin{aligned}
\frac{\partial \mu_{1,0}^{(\mathrm{ML})}}{\partial t} &= \sum_k w_k \xi_{1,k} \\
\frac{\partial \mu_{0,1}^{(\mathrm{ML})}}{\partial t} &= -\frac{3}{2} \sum_k w_k \xi_{1,k}^{-1} \xi_{2,k}^{2} - \frac{4}{\mathrm{Re}} \sum_k w_k \xi_{1,k}^{-2} \xi_{2,k} + \sum_k w_k \xi_{1,k}^{-4} - C_p \sum_k w_k \xi_{1,k}^{-1}, \\
\frac{\partial \mu_{2,0}^{(\mathrm{ML})}}{\partial t} &= 2 \sum_k w_k \xi_{1,k} \xi_{2,k} \\
\frac{\partial \mu_{1,1}^{(\mathrm{ML})}}{\partial t} &= -\frac{1}{2} \sum_k w_k \xi_{2,k}^{2} - \frac{4}{\mathrm{Re}} \sum_k w_k \xi_{1,k}^{-1} \xi_{2,k} + \sum_k w_k \xi_{1,k}^{-3} - C_p,
\end{aligned} \quad (7.19)$$

$$
\begin{aligned}
\frac{\partial \mu_{0,2}^{(\mathrm{ML})}}{\partial t} = -3\sum_k w_k \xi_{1,k}^{-1}\xi_{2,k}^{3} - \frac{8}{\mathrm{Re}}\sum_k w_k \xi_{1,k}^{-2}\xi_{2,k}^{2} + 2\sum_k w_k \xi_{1,k}^{-4}\xi_{2,k} \\
-2C_p\sum_k w_k \xi_{1,k}^{-1}\xi_{2,k}.
\end{aligned} \tag{7.20}
$$

Once trained, the scheme

$$
\boldsymbol{\mu} \xrightarrow{(\mathrm{ML})} \{\boldsymbol{w}, \boldsymbol{\xi}\}
$$

results in a new quadrature rule that evaluates the right-hand-side of (7.12). The moment transport equations (7.12) then evolve via an adaptive 4th-order Runge–Kutta time-stepper. Algorithm 2 describes this procedure.

The baseline 2-by-2-node scheme of [40] imposes certain symmetry assumptions for the reconstructed NDF. In general, this may be non-physical for bubble populations. A more physically consistent approach may be [187]. However, the proposed hybrid scheme varies the positions of the baseline quadrature scheme. Therefore, it allows for non-symmetrical NDFs while, contrary to [187], only resolving moments up to order 2 as input for scheme. The method of [187] may also be considered when generalizing the hybrid approach to high-order schemes. This is because the realizability condition of [40] is meaningfully different when more than 2 nodes are used per direction, rendering it more restrictive. Note that the closure terms need to be evaluated at times t, $t+\delta t/2$, and $t+\delta t$. The neural network does not make predictions at $t+\delta/2$, so the equations are instead integrated in time by $2\delta t$ instead of δt.

Algorithm 2: Hybrid CHyQMOM

```
Input: $\boldsymbol{\mu} = \{\mu_{1,0}, \mu_{0,1}, \mu_{2,0}, \mu_{1,1}, \mu_{0,2}\}$; $C_p, \mathrm{Re}$.
Data: NN architecture, CHyQMOM method, 4th-order-accurate
      Runge–Kutta (RK4), Re, $C_p$, time interval $t \in [0, T]$, error-tolerance
      $\tau_{\mathrm{tol}}$, maximum time-step $\delta t_{\max}$.
Result: $\boldsymbol{\mu}(t_i)$ and $\boldsymbol{\mu}_{\mathrm{Targ.}}(t_i)$ for $i = 0, 1, \ldots, n$
Train $\mathbb{D}_{\boldsymbol{u}}^{(\mathrm{ML})}$ with $\mathbb{D}_{\boldsymbol{u}}^{(\mathrm{MC})}$;
$n \leftarrow 0$ ;
while $t \leq T$ do
    $s \leftarrow t$;
    $\left\{\boldsymbol{w}^{(\mathrm{QBMM})}, \boldsymbol{\xi}^{(\mathrm{QBMM})}\right\} = \mathrm{CHyQMOM}\Big[\boldsymbol{\mu}(s)\Big]$ // Moment inversion
    $\{\boldsymbol{w}', \boldsymbol{\xi}'\}(s) = \mathbb{G}\Big[\boldsymbol{\mu}(s); \left\{\boldsymbol{w}^{(\mathrm{QBMM})}, \boldsymbol{\xi}^{(\mathrm{QBMM})}\right\}(s), C_p(s)\Big]$ // ML correction
    $\{\boldsymbol{w}, \boldsymbol{\xi}\}(s) = \left\{\boldsymbol{w}^{(\mathrm{QBMM})} + \boldsymbol{w}', \boldsymbol{\xi}^{(\mathrm{QBMM})} + \boldsymbol{\xi}'\right\}(s)$ // Set quadrature rule
    $\{\boldsymbol{\mu}, \boldsymbol{\mu}_T, \partial\boldsymbol{\mu}/\partial t\}(s) = \mathrm{Quadrature}\Big[\{\boldsymbol{w}, \boldsymbol{\xi}\}(s)\Big]$ // Moment projection
    $\delta t \leftarrow \delta t_{\max}$;
    flag $\leftarrow 1$;
    while flag $> 1$ do
        $\delta t \leftarrow \delta t/2$;
        $\boldsymbol{\mu}^1(s + \delta t_{\max}) = \mathrm{RK4}\Big[\{\boldsymbol{\mu}, \partial\boldsymbol{\mu}/\partial t\}(s); \delta t\Big]$ // Evolve moments
        $\boldsymbol{\mu}^2(s + \delta t_{\max}) = \mathrm{RK4}\Big[\{\boldsymbol{\mu}, \partial\boldsymbol{\mu}/\partial t\}(s); \delta t/2\Big]$ // Evolve moments
        flag $\leftarrow \mathrm{Floor}\left[\max_{0 \leq l+m \leq 2} \|\mu^1_{l,m} - \mu^2_{l,m}\|_2/\tau_{\mathrm{tol}}\right]$
    end
    $t \leftarrow s + \delta t$;
    $n = n + 1$ ;
end
```

7.3 Application to evolution of bubble populations

7.3.1 Pressure signals

The capabilities of the data-enhanced CHyQMOM method to predict the statistics of bubble populations are explored. The pressure term C_p excites the bubbles causing oscillations. The representation of C_p used here should be general enough to include pressure profiles seen in actual fluid flows. In a generic framework, let C_p have a finite Fourier-series expansion

$$C_p(t) = 1 + \sum_{i=1}^{N} \alpha_i \sin\left[2\pi f_i t + \phi_i\right], \tag{7.21}$$

where t corresponds to nondimensional time, nondimenionalized by the natural oscillation frequency of the bubbles ω_0, f_i are the included dimensionless frequencies, and α_i and ϕ_i are the corresponding amplitude and phase. It is stressed that $C_p = 1$ corresponds to the equilibrium pressure of the bubbles (for which $R = 1$ and $\dot{R} = 0$).

Most cavitating flow applications do not contain pressure frequencies higher than the natural oscillation frequency of the bubbles [18] (with a notable exception of some high-intensity focused ultrasound treatments). We operate under this constraint hereon, though higher frequencies could be included if desired. On the other hand, very low frequencies are uninteresting because they cause the bubbles to evolve quasi-statically. Hence, without significant loss of generality, the dimensionless frequencies of C_p are in the interval $[1/10, 1/5]$. The phases of the waveforms that make up C_p are independently sampled from a uniform distribution $\mathcal{U}$ in $[0, 2\pi]$

$$\phi_i \sim \mathcal{U}([0, 2\pi]), \quad i = 1, 2, ..., 6. \tag{7.22}$$

Applications dictate the possible observed pressure amplitudes. For example, significantly low pressures are not relevant for most applications. To set an empirical threshold approximating this condition, the pressures must not cause the used Monte Carlo simulation configuration to become numerically unstable. The solver itself uses an adaptive 3rd-order Runge–Kutta scheme with minimum time-step 10^{-6} and relative error tolerance of 10^{-7}. Thus, we design a pressure distribution from which all samples are numerically stable and physically realistic. Algorithm 3 details this process.

```
Algorithm 3: Forcing Amplitude Sampling
$\alpha_i \sim \mathcal{U}([0,1]), \quad i = 1, 2, ..., 6;$
$\alpha \leftarrow \sum_{i=1}^{6} \alpha_i;$
$\alpha_i \leftarrow \max\left(5\alpha/3\right)\alpha_i, \quad i = 1, 2, \ldots, 6$
```

Previous experimental works can also be used to justify that the forcing constraint in algorithm 3 avoids abrupt cavitation. This is estimated by the cavitation number

$$\sigma = \frac{1 - p_V(T_0)/p_0}{\rho U_0^2/2p_0}, \tag{7.23}$$

where $p_V(T_0)$ is the vapor pressure of the liquid at reference temperature T_0 and U_0 is the reference velocity [18]. If the liquid cannot withstand negative pressures then vapor bubbles appear when the liquid pressure is p_V. Thus, nucleation occurs when

$$\sigma \geq -\min_t \left\{ \frac{C_p(t) - 1}{\rho U_0^2/2p_0} \right\}. \tag{7.24}$$

Without loss of generality, we can choose $\rho U_0^2/(2p_0) = 1$ to simplify the following calculations. For flows around axisymmetric headforms, with Reynolds number in the range of 4.5×10^5, if $\sigma \leq 0.40$, the formed nuclei grow explosively up to a certain bubble size [26]. This phenomenon renders numerical simulations for flows with $\sigma \leq 0.40$ considerably more expensive compared to $\sigma > 0.40$ cases in order to achieve the same numerical accuracy. For the pressure profiles presented here, the case $\sigma \leq 0.40$ is avoided when using algorithm 3.

Figure 7-1 shows example pressure profiles $C_p(t)$ that are used to test the fidelity of the hybrid moment inversion method. Herein, the end of this time window, $t \in [40, 50]$, is used to assess model fidelity. This enables the bubble dynamics to evolve from a specific initial state to one more representative of those found in actual flows.

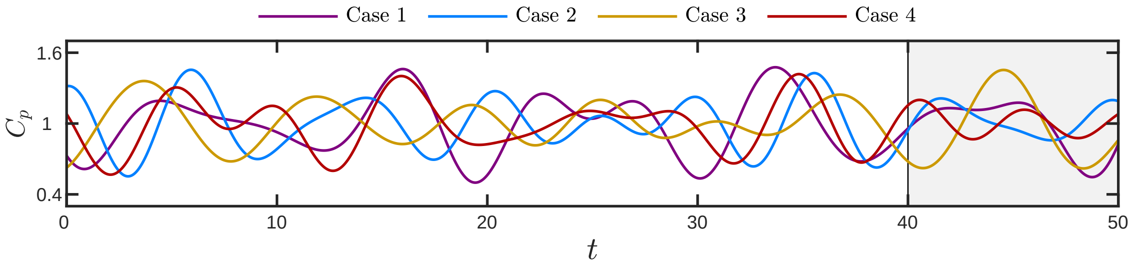


FIG. 7-1. The time-history of example realizations of C_p. Comparisons of the time-history of the evolved moments $\boldsymbol{\mu}$ and target moments $\boldsymbol{\mu}_{\text{Targ.}}$ between different numerical schemes are performed in the shaded time-interval $t \in [40, 50]$.

7.3.2 LSTM RNN training procedure

We simulate 1000 samples of individual bubbles for each realization of C_p. The bubbles are initialized via samples from normal distributions with variances $\sigma_R^2 = 0.05^2$ and $\sigma_{\dot{R}}^2 = 0.05^2$ for R and $\dot{R}$ respectively. The values of R and $\dot{R}$ are sampled independently from one another. The choice of initial distributions, given small variance, is not particularly important for the evolution of the statistics. Each case is evolved until $t = 50$, which in this dimensionless system corresponds to 50 natural periods of bubble oscillations. The individual bubble dynamics are then averaged to obtain the Monte Carlo reference statistics for each C_p realization. To showcase that 1000 samples provide MC data of adequate accuracy, Figure 7-2 shows the L^2-error for a particular set of moments as the number of samples N increases, compared to reference MC data with $N = 10^5$.

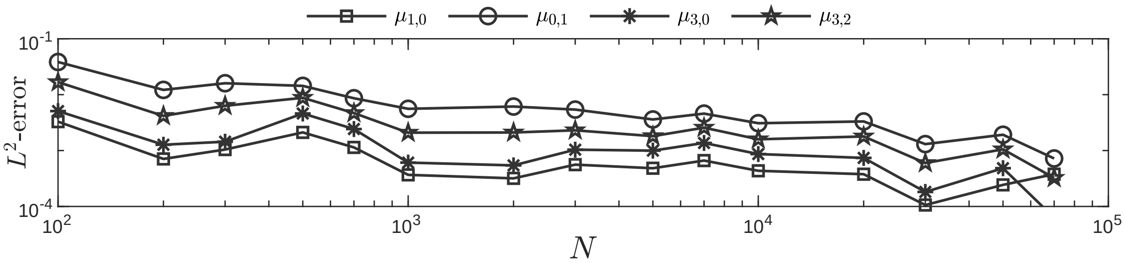
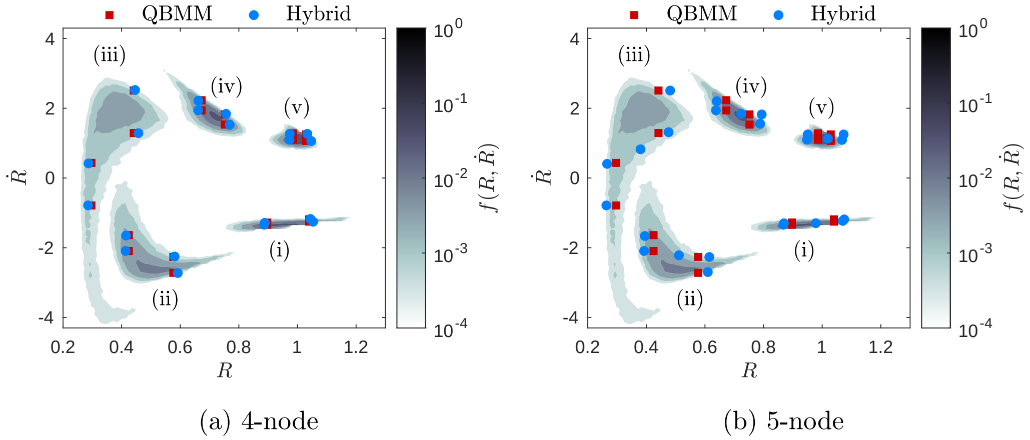


FIG. 7-2. L^2-error of MC data for variable N compared to reference MC data with $N = 10^5$.

For the numerical investigation, 200 samples of C_p from (7.21) are used. From these, 50 are randomly selected for training, with the remaining 150 cases used during testing. The number of samples used for training deviates from common practices (where about 80% of data are used) and instead is chosen so that it is large enough to avoid over-fitting but small enough to still allow for the sampling of qualitatively different pressure profiles during testing. The neural network includes 28 hidden units and is trained on the entire time history of each of the 50 selected samples of C_p. Figure 7-3 shows f and the $\boldsymbol{\xi}$ for one realization of C_p at different time instances. The same figure displays the CHyQMOM nodes as estimated by both the standard 4-node CHyQMOM scheme and the 4- and 5-node hybrid schemes.



(a) 4-node

(b) 5-node

FIG. 7-3. Temporal snapshots of f computed via Monte Carlo and the positions of the quadrature nodes for the 4-node CHyQMOM quadrature scheme (QBMM) and the 4- and 5-node hybrid CHyQMOM quadrature schemes (Hybrid). The labels (i)–(v) correspond to times $t = 43.9, 44.1, 44.2, 44.4$, and 44.6, respectively.

During training, the LSTM memory size is set to 256 time-instances, with each $\delta t = 0.01$ apart. The Adam method [78] trained each neural network for 500 epochs, minimizing the loss function (7.16). A table with the values of the hyperparameters of the neural networks is presented in appendix C.2. A 4th-order Runge–Kutta adaptive time stepper evolves the predictions of the hybrid scheme. The time integration scheme is adaptive, but the neural network predictions are uniform, so the neural network corrections are limited to the associated fixed time step $\delta t = 0.01$. To initialize the neural network during testing, the MC data for the time-interval $[0, 0.31]$ are used as input.

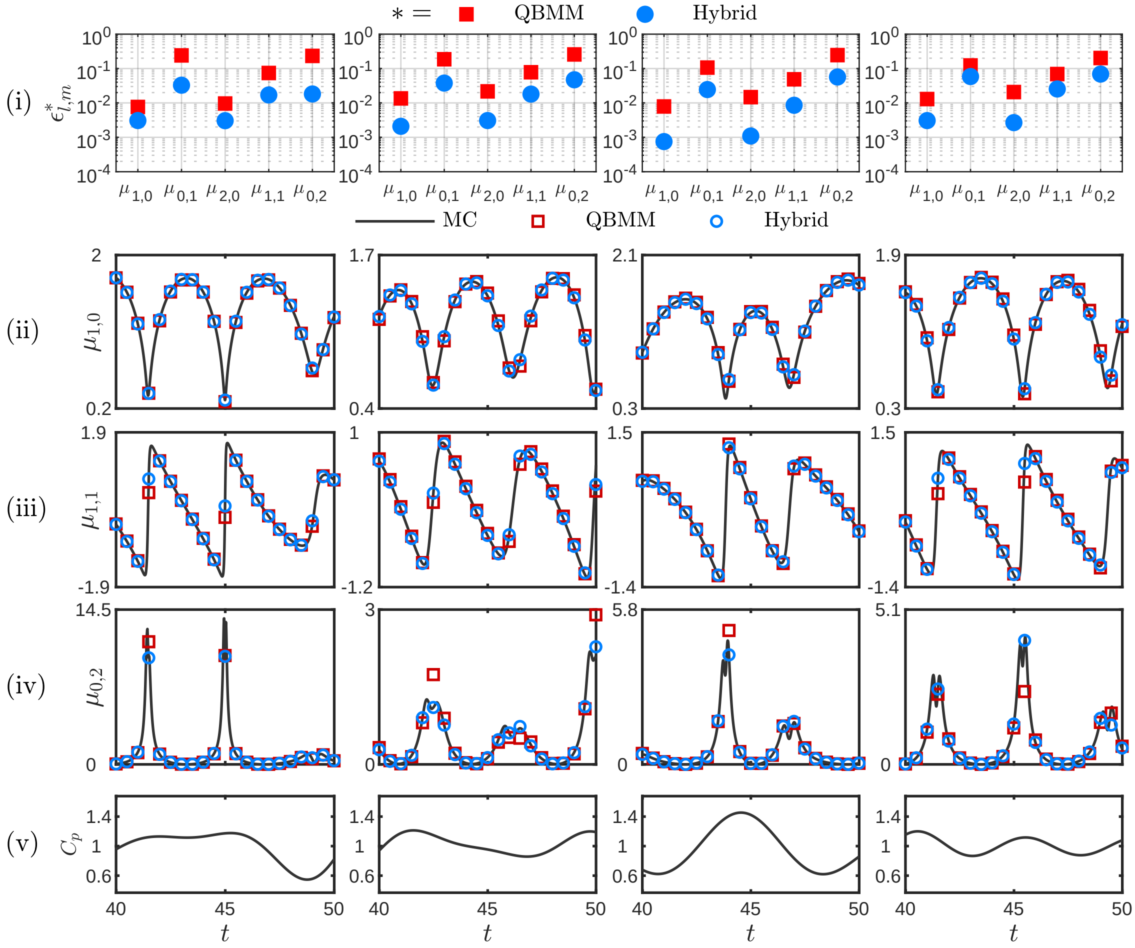
7.3.3 Low-order moment evolution and error quantification

The model-form relative error is computed via a discrete L^2 error

$$\epsilon_{l,m}^{(*)} = \sqrt{\frac{\sum_{i=1}^{N_t} \left[\mu_{l,m}^{(*)}(t_i) - \mu_{l,m}^{(\mathrm{MC})}(t_i)\right]^2}{\sum_{i=1}^{N_t} \left[\mu_{l,m}^{(\mathrm{MC})}(t_i)\right]^2}}, \tag{7.25}$$

where (MC) indicates Monte Carlo surrogate truth data and $* = \{\mathrm{ML}, \mathrm{QBMM}\}$. The t_i are $N_t = 5000$ uniformly spaced times in the interval $t \in [0, 50]$. Results regarding the low-order moments are presented in figure 7-4.

Figure 7-4 (i) shows $\epsilon_{l,m}^{*}$ for the first- and second-order moments $\boldsymbol{\mu}$ for the 4-node schemes. Rows (ii–iv) shows the evolution of select moments for $t \in [40, 50]$ and row (v) shows the corresponding C_p. All results correspond to 4 randomly selected testing samples (a)–(d) as labeled. We observe a smaller ϵ for the hybrid scheme than standard CHyQMOM for all 4 cases considered. The largest errors for both approaches appear for moment $\mu_{0,2}$ (row (iv)), which exhibits large and intermittent fluctuations when the bubbles collapse. The CHyQMOM method deviates most from the MC surrogate-truth data during intervals of high compression (small C_p), with hybrid CHyQMOM tracking the reference solution more accurately. Thus, the observed increase in accuracy varies significantly from case-to-case and moment-to-moment, from 10 times smaller errors to only 20% improvements. The evolution of the L^2 error for Case 2 is shown in figure C-1 of appendix C.3.

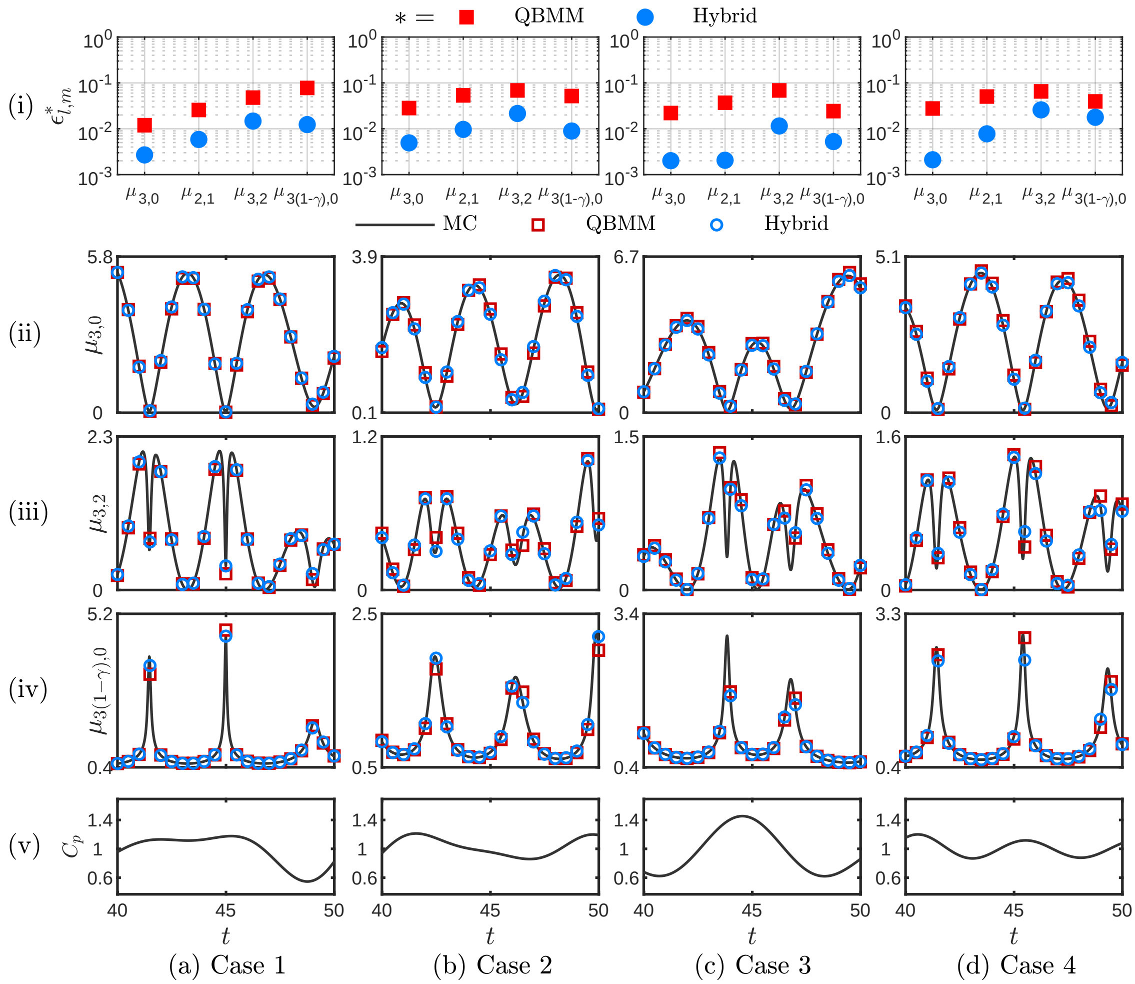


(a) Case 1 (b) Case 2 (c) Case 3 (d) Case 4

FIG. 7-4. Low-order moment evolution for 4-node CHyQMOM and hybrid CHyQMOM methods. Results compare against surrogate-truth Monte Carlo (MC) data.

7.3.4 High-order moment extrapolation

The next quantity of interest is the L^2-error in predicting the target higher-order moments $\boldsymbol{\mu}_{\text{Targ.}}$. Figure 7-5 presents these results for the same 4 testing samples presented in figure 7-4. For all moments (ii)–(iv), hybrid CHyQMOM significantly improves the predictions of $\boldsymbol{\mu}_{\text{Targ.}}$. This improvement is associated with the more accurate evolution of the low-order moments $\boldsymbol{\mu}$ and these targeted moments in the neural network training procedure.



(a) Case 1 (b) Case 2 (c) Case 3 (d) Case 4

FIG. 7-5. Comparison of predictions of target moments $\boldsymbol{\mu}_{\mathrm{Targ.}}$ for Monte Carlo data (black line), CHyQMOM predictions (red dashed line), and hybrid CHyQMOM predictions (blue line) for 4 quadrature nodes.

To better study the typical reduction in error for the 4-node hybrid CHyQMOM scheme, we compute the percent improvement of the L_2 error as

$$\mathcal{Q}_{l,m} = 100\frac{\epsilon_{l,m}^{(\mathrm{QBMM})} - \epsilon_{l,m}^{(\mathrm{ML})}}{\epsilon_{l,m}^{(\mathrm{QBMM})}}. \tag{7.26}$$

Figure 7-6 shows a histogram of $\mathcal{Q}$ calculated for example low-order moments $\boldsymbol{\mu}$ and target moments $\boldsymbol{\mu}_{\mathrm{Targ.}}$. For all testing samples, the 4-node hybrid scheme improves the accuracy of the standard CHyQMOM method.

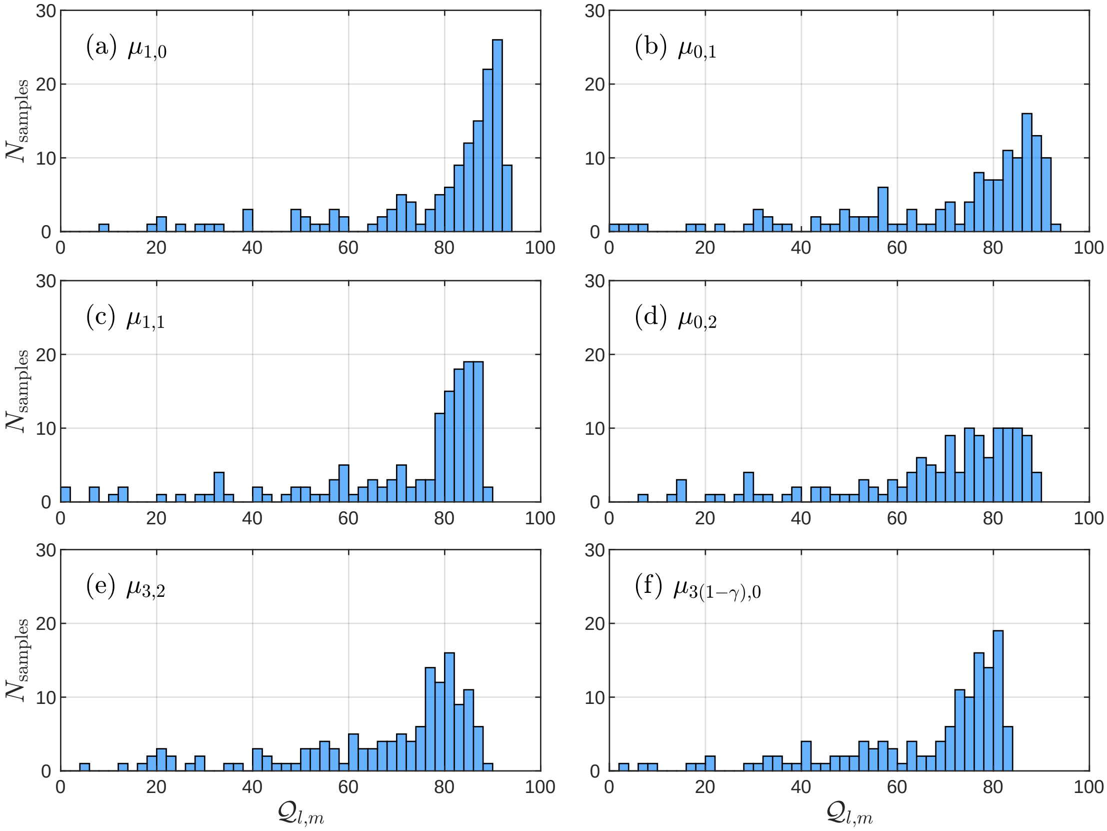


FIG. 7-6. Histogram of L^2-error improvement $\mathcal{Q}$ for hybrid CHyQMOM over traditional CHyQMOM for example (a–d) low-order moments and (e–f) target high-order moments. Cases are drawn from 150 realizations of C_p.

Further, for both low-order moments ($\boldsymbol{\mu}$) and target moments ($\boldsymbol{\mu}_{\mathrm{Targ.}}$), the error is reduced by more than 50% for more than 80% of the sampled C_p cases. Target moments $\boldsymbol{\mu}_{\text{Targ.}}$ exhibit L^2-error improvement ranging from 5% to 96%. The variation in error improvement is due to the amplitude range of the sampled C_p and how closely the time-series of C_p corresponds to one of the training samples. Thus, results can improve by including more training samples.

7.3.5 Additional quadrature nodes

Another potential route to method improvement is to increase the number of quadrature nodes. While the number of quadrature nodes can change, the evolved moments remain $\boldsymbol{\mu}$. The algorithm for this is included in appendix C.1.

To quantify the effect of this change, $\epsilon_{l,m}^{1/2}(*)$ is computed, which is the median $\epsilon_{l,m}^{(*)}$ error among the 150 test samples. We then define

$$C_{l,m} = \frac{\epsilon_{l,m}^{1/2}(\text{ML})}{\epsilon_{l,m}^{1/2}(\text{QBMM})} \tag{7.27}$$

to quantify the decrease in the L^2-error when using higher-node-count hybrid CHyQMOM compared to the standard 4-node CHyQMOM.

Figure 7-7 shows that the accuracy of the hybrid predictions is improved as the number of nodes N_ξ increases. However, the error improvement gains diminish once 7 nodes are reached. Further, including additional nodes to the quadrature rule increases the computational time needed to perform a single time-step evolution for the system. The computational cost of 4-node hybrid CHyQMOM per time-step is 8.9 times the cost of CHyQMOM. For 5, 6, and 7 nodes, the hybrid method costs per time-step are 11.5, 13.8, and 16.2 times that of 4-node CHyQMOM[1]. Hence,

[1] Simulations were performed using `PyQBMMlib` [23] on 1 core of a 2.3 GHz Intel Core i9 CPU.

diminishing improvements are observed as the number of nodes increases to more than 6, as the simulations require significantly more computations per time-step for comparable accuracy.

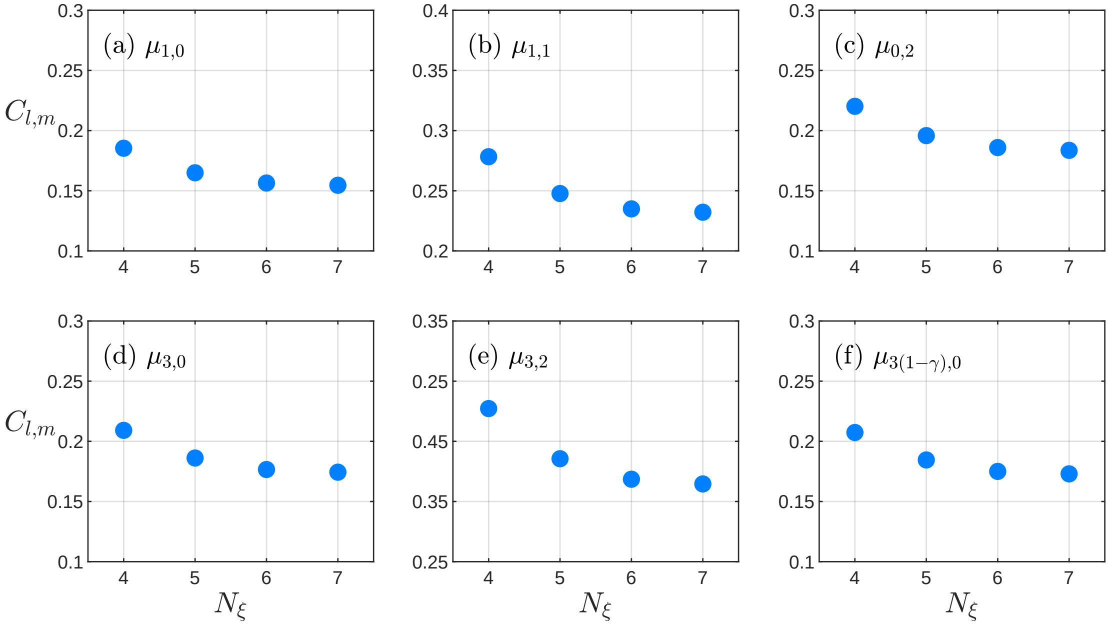


FIG. 7-7. Median error decrease while using hybrid CHyQMOM over 4-node CHyQMOM for different numbers of nodes for the hybrid CHyQMOM scheme.

Chapter 8

Conclusions and Future Work

8.1 Summary of results

This thesis demonstrated a multitude of machine learning strategies for predicting the statistics of turbulent and highly nonlinear dynamical systems under appropriate physics-constrained frameworks. All approaches revolve around employing statistical formulations of the problem and utilizing recurrent neural networks to accurately estimate the reduced-order dynamics from delay coordinates. These neural networks can replace closure terms of coarse-scale equations of turbulent flows, improve existing numerical schemes or entirely replace the dynamical system. The viability of the method was tested on a number of applications from climate modelling to bubble cluster dynamics.

In chapter 4, we have demonstrated the application of the energy conservation property of the advection terms on machine learning non-local closures for turbulent fluxes. We have adopted two neural network architectures, based on LSTM and TCN, to further include memory effects in our analysis. Clearly, the physical constraint is not restricted to these two frameworks and can be employed in other machine learning architectures, as well as other fluid systems (e.g. environmental flows). We demonstrated the computed closures in two-dimensional jets in an unstable regime and showed that closures obtained from unimodal jets can be used for different jet geometries. The adoption of the physical constraint significantly and consistently improved the accuracy of the mean-flow predictions obtained from the corresponding coarse-scale models independently of the adopted architecture or the flow setup. This improvement was on average 26% for one-dimensional closures and 29% for two-dimensional closures, being notably larger for flows that were not used for training. Moreover, the constraint improved the numerical stability of the coarse-scale models especially in Reynolds numbers, which were higher than the ones included in the training data sets. While the adopted examples are relatively simple, yet unstable, fluid flows, the presented energy constraints do not depend on the complexity of the flow.

In chapter 5, we have formulated and assessed a data-informed turbulence-closure scheme that respects the underlying conservation properties of nonlinear advection. The method employs a second-order framework for the uncertainty quantification of nonlinear and turbulent dynamical systems. We first applied our approach to prototypical problems of nonlinear dynamics, like the unstable triad system and the Lorentz-96 model. In both cases, the data-informed approach produced results in good agreement with reference MC simulations. Furthermore, the method was applied to more realistic turbulent flows, involving anisotropic multiphase flows and a 2-layer quasigeostrophic model. The obtained results demonstrated the improvement of applying the closure at the second-order level, as opposed to mean-flow closures, but also how the results are improved by encoding the energy conservation property of the nonlinear terms in the training process. In addition, we illustrated that the

ML closure of the covariance equation allows for accurate modeling of highly nontrivial non-Gaussian statistics that govern the response of low wavenumbers in the QG model.

In chapter 6, we have formulated and assessed a data-informed hybrid scheme for accurately computing the statistics of climate models. The method employs a nudged solver during training with appropriate spectral corrections to the produced dataset. During testing, the model is assessed on free-running coarse-scale climate models. The approach was applied to realistic atmospheric climate data. Free-running EAMv2 simulations were used as baseline while ERA5 reanalysis data were employed as reference truth. First, the ability of the model to predict global statistics of horizontal velocities (U, V), temperature T and specific humidity Q was assessed. In all cases, the data-informed approach produced results in good agreement with reference ERA5 predictions. Furthermore, the model was tasked with predicting mean IVT over the period 2007-2017, a highly anisotropic quantity. Results are again in good agreement with ERA5, with the corrected dataset having a smaller mean-square error than a nudged dataset. Finally, the model was used to predict statistics of tropical cyclones throughout the globe. The model's ability to correct the number of predicted cyclones over the pacific was demonstrated.

In chapter 7, a data-informed conditional hyperbolic quadrature method for statistical moments was presented. The method was applied to the statistics of a population of spherical bubbles oscillating in response to time-varying randomized forcing. The forcing is designed to resemble any possible function with a banded frequency spectrum from 1/5 to 1/10 the natural frequency of the bubbles. Results showed that the method reduces closure errors when compared against a standard 4-node CHyQMOM scheme. The hybrid method reduced errors more significantly for the extrapolated higher-order moments that close the phase-averaged flow equations. This improvement was achieved via recurrent neural networks that include time history during training and were trained using a fixed time-step. This result is significant because higher-order QBMM schemes are generally numerically unstable for this problem, so another approach is required to improve accuracy. Thus, while the presented hybrid scheme is about a factor of 10 more expensive than CHyQMOM, its numerical cost should be viable for many applications. Furthermore, the presented method can be effectively applied to any dynamical system with non-Gaussian statistics where high-order moments are of interest.

8.2 Future work

While the proposed online closure schemes of chapter 4 and chapter 5, adequately predict flows that are distinct from those of the training set, it is important to state that the computed closure is not expected to be as effective in every fluid flow. This is due to the fact that depending on the specifics of a fluid flow, such as dimensionality, boundary conditions, domain geometry, excitation terms, and the presence of additional dynamics such as Coriolis terms, the nonlinear interactions between scales are different. Therefore, aiming to machine learn universal closures that will work for every case is beyond the scope of this thesis. Instead, our approach is to employ data from flows that have some common features with the flow we are interested in modeling, and combine this data with a universal constraint, the energy conservation by the nonlinear terms, to increase the accuracy of the computed closures. The optimal choice of input features is also expected to vary depending on the specifics of the flow and therefore a numerical examination of different combinations should be performed to achieve the most effective closure.

In chapter 6, while the non-intrusive hybrid approach was shown to effectively capture statistics of extreme events, further future pathways can be highlighted. First, active learning can determine optimal datasets for the training of this system. This step is of particular importance in a setup where the datasets and models are very large and training is cumbersome. Furthermore, clustering techniques can be used to interpret the selection process by uncovering potentially important physical phenomena that should be included in training. Another path is a systematic study of uncertainty quantification in the predictions of the model itself. For that purpose, an ensemble of neural networks that are trained over the same dataset can be used. The variance between the predictions of these NNs can be vital in quantifying uncertainty and error propagation in scenarios where ground truth cannot exist. This technique can also be used during training to assess how much the proposed architecture tends to overestimate or underestimate its own uncertainty. Finally, the proposed model can be used for fututre climate scenarios as a computationally efficient model for adequately accurate extrapolation of climate statistics under different scenarios.

Appendix A

Energy preserving closure-schemes for first-order statistics in 1D flows

We study the effectiveness of the proposed closure scheme in chapter 4, when we take advantage of the translational invariance of the flow in the $x-$direction. This allows us to obtain a closed averaged equation for the $y-$profile of the jet. In the second case we do not rely on this symmetry and obtain closures directly for the two-dimensional flow. We will compare the adopted architectures for both the case of utilizing the energy constraint presented above and not.

A.0.1 One-dimensional closures for the jet profile

We take advantage of the translational invariance of the flow in the $x-$direction and select the spatial-averaging operator to be integration along the full x-direction and local spatial averaging along the y-direction:

$$\bar{f}(y) = \frac{1}{2\pi} \iint_{S^2} w_l(y' - y) f(x, y') \mathrm{d}x \mathrm{d}y' \tag{A.1}$$

where, $w_l(y'-y) = \frac{1}{2l}[H(y'-y-l)-H(y'-y+l)]$ is the piece-wise constant averaging window of length $2l$; here $l = \frac{2\pi}{20}$. Applying the averaging operator to the governing equations we obtain the equation for the $y-$profile of the jet. Note that based on our averaging operator for this case, we have $\bar{u}_2 = 0$ and $\bar{u}_1$ is only a function of y. To this end, we have:

$$\partial_t\overline{u}_1 = -\mathbb{D}_{u_1} + \frac{1}{\mathrm{Re}}\partial_y^2\overline{u}_1 + \nu\nabla^{-4}(\overline{u}_1 - u_{1,jet}) + \overline{F}_1, \tag{A.2}$$

$$\partial_t\overline{\rho} + \partial_y(\overline{v_2}\,\overline{\rho}) + \mathbb{D}_\rho = \nu_2\partial_y^4\overline{\rho}. \tag{A.3}$$

Therefore, the objective function used for training of the flow closure takes the form:

$$L(\theta_1) = \int_{\Omega\times T}\left|\left|\mathbb{D}_{u_1}\left[\theta_1;\overline{\xi}\right] - \overline{(\mathbf{u}'\cdot\nabla)\mathbf{u}'}\right|\right|^2 \mathrm{d}\mathbf{x}\mathrm{d}t + \lambda\int_T\left|\int_\Omega \overline{u}_1\cdot\mathbb{D}_{u_1}\left[\theta_1;\overline{\xi}\right]\mathrm{d}\mathbf{x} - A[\mathbf{u}]\right|\mathrm{d}t, \tag{A.4}$$

where $A[\mathbf{u}] = -\int_\Omega \overline{\mathbf{u}'\cdot(\mathbf{u}'\cdot\nabla)\mathbf{u}'}\mathrm{d}\mathbf{x}$. On the other hand, the objective function for the density field closure takes the form:

$$L(\theta_2) = \int_{\Omega\times T}\left|\left|\mathbb{D}_\rho\left[\theta_2;\overline{\zeta}[\alpha(\mathbf{x}),\chi(t)]\right] - \nabla\cdot\left(\overline{\mathbf{v}'\rho'}\right)\right|\right|^2 d\mathbf{x}dt. \tag{A.5}$$

The neighborhood $\alpha(y)$ is selected to have five nodes in total:

$$\alpha(y) = \{y + m\delta y\},\ \delta y = 2\pi/80,\quad m = -2,-1,...,2,$$

while the temporal horizon in the past is selected as

$$\chi(t) = \{t - m\delta\tau\}, \delta\tau = 10^{-2},\ m = 1, 2, ..., 12.$$

Both the spatial extent of the neighborhood and the memory are chosen as the threshold values above which any further increase does not result significant difference in the training and validation errors.

Neural network architecture

We assess two different architectures for our closure scheme. In the first case, we represent the flow closure with 3 convolutional-LSTM layers and the density closure with 4 convolutional-LSTM layers (16 time-delays). Further increase of the number of layers does not offer any significant improvement in the training and validation errors. The adopted architectures are presented in Figure A-1 (a-b). For the second machine learning architecture we use 4-layer temporal convolutions to model the memory terms of our closure for both the flow and the density fields. The architecture in this case is depicted in Figure A-2(a-b). An important difference between the two architectures that is worth emphasizing is associated with their computational cost. Specifically, in the LSTM architecture we have a memory term that is updated at each time-step and to this end, LSTM needs to only operate on the flow features at each time-step. On the other hand, TCN layers operate on the entire included time-history making them more computationally expensive.

Feature selection

The selection of the flow features that are used as inputs for the data-driven closures is done numerically by testing different combinations of basic flow features. We eventually choose the combination that minimizes the training and the validation error. It is important to emphasize that if we rely only on the training error we run the risk of over-fitting. We carry out this process individually for each of the closure terms for the Navier-Stokes equation and the transport equation. For the closure term $\mathbb{D}_{\mathbf{u}}$ we select as possible flow features the quantities: $\overline{u}_1, \partial_t \overline{u}_1, \partial_y (\overline{u_1' u_2'})$. All input flow features are separately normalized to have variance equal to 1. Table 1 summarizes performance for different combinations over 100 periods for the TCN and LSTM architectures with the physical constraint. It is observed that while $\{\partial_t \overline{u}_1, \partial_y (\overline{u_1' u_2'})\}$ drastically decreases training error, the inclusion of the mean flow profile $\overline{u}_1$ strongly improves validation error. Since the best validation error is achieved in this case, all considered flow features are employed.

For the implementation of the presented closure scheme the input features are not imposed to be Galilean invariant. Requiring the input features to be Galilean invariant can be justified when the closure scheme is intended to model drastically different flows (different boundary conditions, flow geometry or significantly different Reynolds number). Machine learning universal closures for turbulence is beyond the scope of this effort. This is because this task is associated with extreme obstacles, such as selection of training data which are representative of essentially every possible dynamical regime and geometry, and the assumption that there is a machine learning architecture that can generalize well over such a vast range of conditions. Instead, the goal here is to develop closures that can generalize well over a family of flows with common topology and dynamics. To this end, the need for the features to be Galilean is not necessary, since the frame of reference for all the produced jet flows is the same and the magnitude of the examined jets is very similar.

Table A.1. Feature selection for the one-dimensional closure of the Navier-Stokes.

ξ feature selection					
		cTCN		cLSTM	
Features	Dimension	Train-MSE	Val-MSE	Train-MSE	Val-MSE
$\overline{u}_1$	1	0.094	1.712	0.102	1.501
$\partial_t \overline{u}_1$	1	0.037	0.535	0.041	0.501
$\partial_y(\overline{u'_1 u'_2})$	1	0.028	0.144	0.033	0.139
$\overline{u}_1, \partial_t \overline{u}_1$	2	0.042	0.418	0.056	0.511
$\overline{u}_1, \partial_y(\overline{u'_1 u'_2})$	2	0.023	0.159	0.028	0.157
$\partial_t \overline{u}_1, \partial_y(\overline{u'_1 u'_2})$	2	0.021	0.092	0.026	0.085
$\overline{u}_1, \partial_t \overline{u}_1, \partial_y(\overline{u'_1 u'_2})$	3	0.021	0.029	0.025	0.032

Table A.2. Feature selection for closure of bubble transport equation.

ζ feature selection					
		cTCN		cLSTM	
Features	Dimension	Train-MSE	Val-MSE	Train-MSE	Val-MSE
$\overline{\rho}$	1	0.109	0.150	0.123	0.171
$\overline{\boldsymbol{v}}$	2	0.603	0.673	0.592	0.625
$\overline{\boldsymbol{v}}, \overline{\rho}$	3	0.081	0.090	0.094	0.101
$\overline{\boldsymbol{v}}, \overline{\rho}, \partial_t \overline{\boldsymbol{v}}, \partial_t \overline{\rho}$	6	0.058	0.060	0.061	0.064
$\overline{\boldsymbol{v}}, \overline{\rho}, \partial_t \overline{\boldsymbol{v}}, \partial_y(\overline{\rho' v'_2})$	6	0.028	0.039	0.042	0.088
$\overline{\boldsymbol{v}}, \overline{\rho}, \partial_t \overline{\rho}, \partial_y(\overline{\rho' v'_2})$	5	0.027	0.036	0.035	0.049
$\overline{\boldsymbol{v}}, \overline{\rho}.\partial_t \overline{\boldsymbol{v}}, \partial_t \overline{\rho}, \partial_y(\overline{\rho' v'_2})$	7	0.025	0.031	0.033	0.044

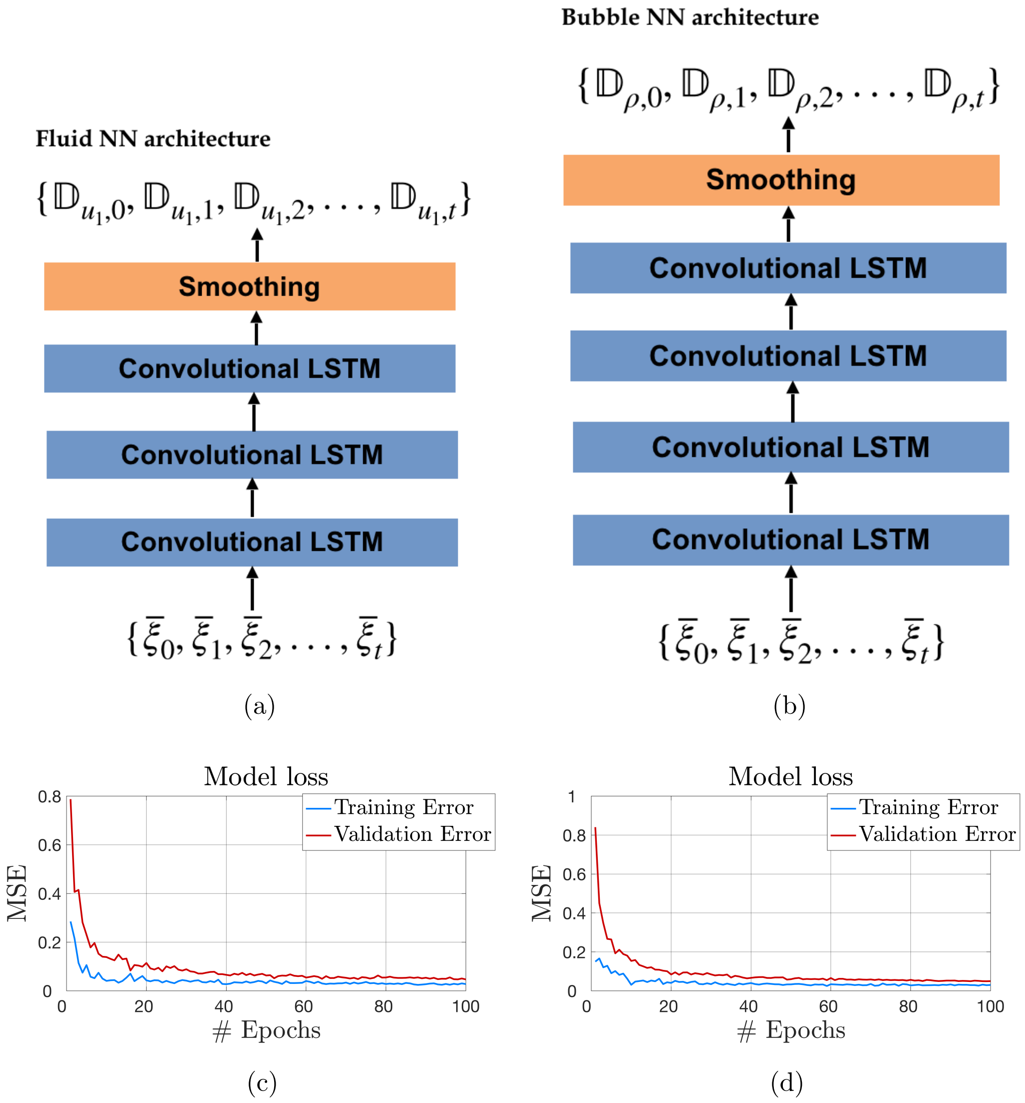
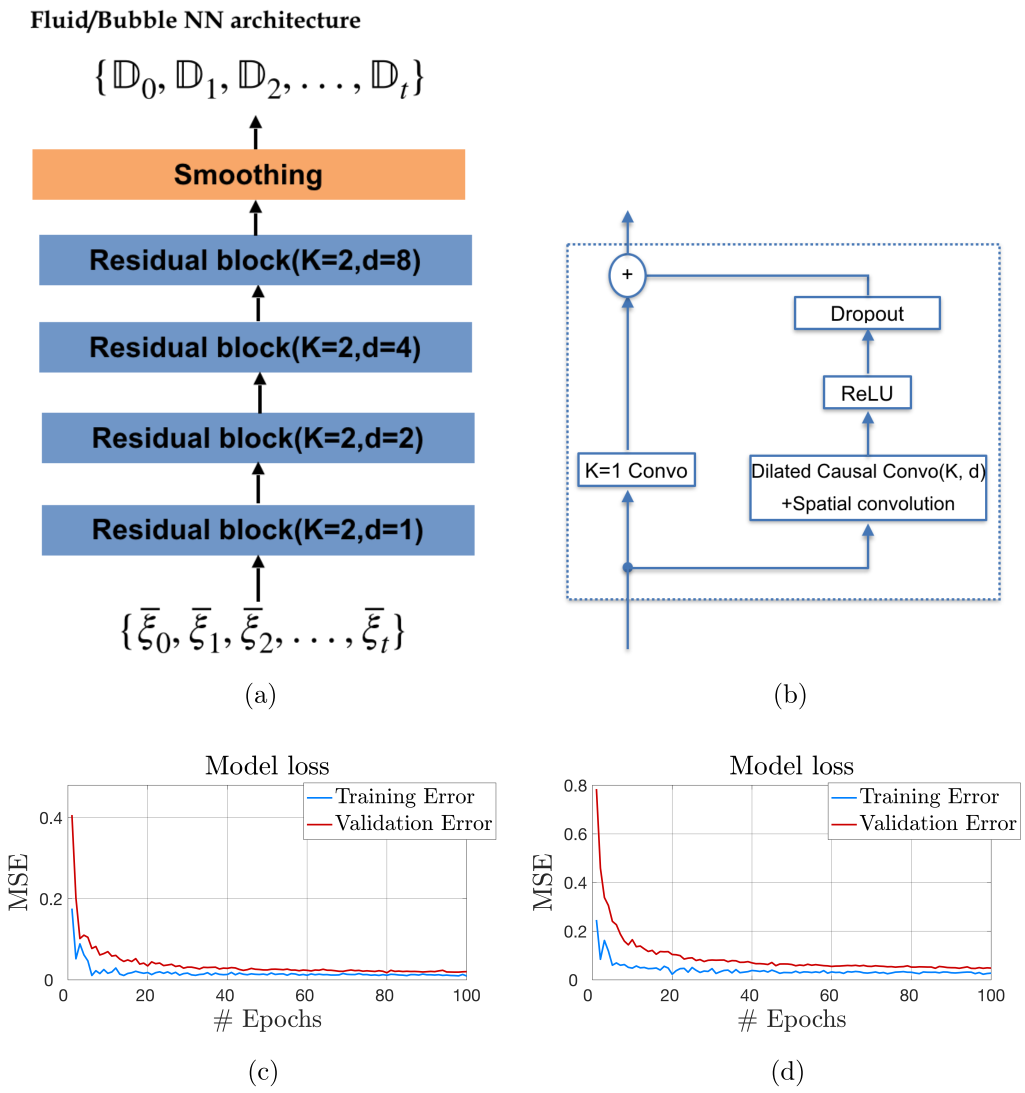


FIG. A-1. (a) Architecture of the LSTM neural network for parametrizing the term $\mathbb{D}_t = \partial_y(\overline{u_1' u_2'})$. (b) Architecture of the LSTM neural network for parametrizing the term $\mathbb{D}_t = \partial_y(\overline{v_2' \rho'})$. (c) Mean square training-error (solid line) and validation error (dashed line) for $\mathbb{D}_u$. (d) Mean square training-error (solid line) and validation error (dashed line) for $\mathbb{D}_\rho$.



(a) (b)

(c) (d)

FIG. A-2. (a) Architecture of the TCN neural network for parametrizing the term $\partial_y(\overline{u_1'u_2'})$ and $\partial_y(\overline{v_2'\rho'})$. (b) Inner architecture of a residual block. (d) Mean square training-error (solid line) and validation error (dashed line) for $\mathbb{D}_u$. (e) Mean square training-error (solid line) and validation error (dashed line) for $\mathbb{D}_\rho$.

For the closure of the transport equation we carry out the same process in Table 2, where we present the training and validation error over 100 periods. We observe that the single most important feature is $\bar{\rho}$. Based on the mean-square error (both training and validation) we choose the combination of $\overline{u}, \overline{\rho}, \partial_t \overline{u}, \partial_y(\overline{\rho' v'})$. In Figure A-1 we present the value of both validation and training error with respect to the number of epochs. These are quite similar, hinting towards generalizability of the predictions. It is worth mentioning that when spatial derivatives are added to the selected feature sets shown previously, training and validation error didn't improve. This could be a result of non-local information of the (averaged) fluid flow profile and density profile used as input, allowing the convolutions to combine these values in a finite-difference sense to derive spatial derivative information that is needed. However, in the case of trying to test this model to drastically different flow set-ups, the Galilean invariant partial spatial derivatives are probably appropriate to replace the non-Galilean averaged features, i.e. the fluid flow profile and density profile.

A.1 Validation and generalizability for one-dimensional closures

To showcase the generalizability properties of the obtained closures we train on unimodal jets and test on bimodal ones. We mention again that the averaged model is one-dimensional and we use 80 points in space to simulate it. We compare the results of the averaged model with the predictions of the two-dimensional reference solutions that we computed using a spectral method and 256^2 modes. Each training case contains data in the time-interval $T = [200, 600]$.

A.1.1 Validation on unimodal jets

As seen in Sec. 4.2, the unperturbed jet profile is chosen as,

$$u_{1,\text{jet}} = \exp\Big[-2(y-\pi)^2\Big], \quad u_{2,\text{jet}} = 0. \tag{A.6}$$

We train four different models on unimodal jets of Re $\in \{650, 750, 850\}$. We use LSTM and TCN architectures with and without enforcing the physical constraint of eq. (A.4). In Fig. A-3, we present the time- and y-averaged mean-square error between the x-averaged profile of the DNS, $\bar{u}^*$ and the coarse scale model, $\bar{u}$:

$$||\bar{u}^* - \bar{u}||_2^2 = \frac{1}{2\pi T}\int_0^{2\pi}\int_{t_0}^{t_0+T}(\bar{u}^*(y,t) - \bar{u}(y,t))^2 dydt. \tag{A.7}$$

We observe that the TCN models clearly outperform the LSTM based closures. Moreover, training using the objective function that includes the physical constraint (Eq. A.4) for the advective terms (cTCN and cLSTM) significantly improves the testing results for the two architectures by 23% and 25%, respectively (Table A.3). This improvement comes at no additional cost in terms of data, but only using the physical constraint associated with the advection terms, which does not depend on the knowledge of any physical quantity of the flow or any other system-specific information.

In Fig. A-4 we present additional results for the cTCN model (best performer). We showcase results both for the time-averaged profile for the fluid velocity and for the bubble distribution for Re = 1000. Comparisons are made between the time-averaged results that the one-dimensional closure scheme produces and the time- and $x-$ averaged results of the two-dimensional reference solution. Specifically, the

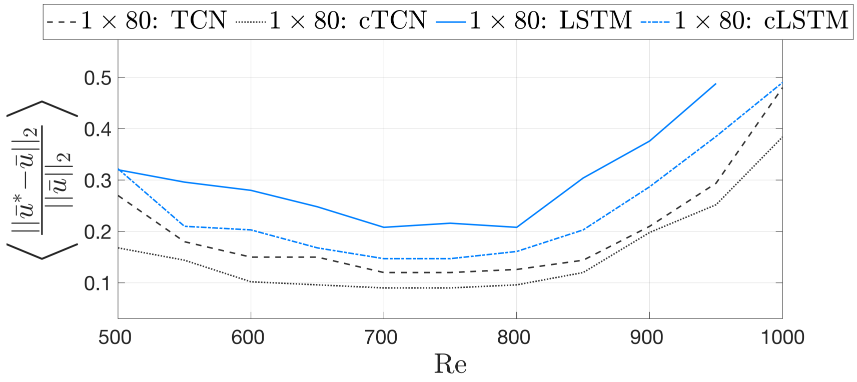


FIG. A-3. Normalized error (A.7) for one-dimensional closure models using TCN, LSTM and their constrained versions, cTCN and cLSTM for unimodal jets. Training data includes unimodal jets with Re $\in \{650, 750, 850\}$.

time-averaged jet-profile is computed as

$$\langle \overline{u}_1 \rangle (y) = \frac{1}{T} \int_{t_0}^{t_0+T} \overline{u}_1(y,t) dt, \tag{A.8}$$

where $\bar{u}_1$ is the $x-$averaged reference solution and the temporal averaging parameters are chosen as $t_0 = 200$ and $T = 400$; note that we omit the first transient part of the simulation.

The results appear to be in very good agreement showcasing that our closure scheme is able to predict the statistical steady state of the flow. We can also observe the Rayleigh instability that the initial jet profile (dashed line) undergoes due to the excitation by the external forcing. We note that the slight asymmetry that the velocity profile exhibits is due to a minor inhomogeneity of the forcing term along the y direction. We apply the same operation described above to compute $\langle \overline{\rho} \rangle$ from both the reference solution and the data-based closure scheme (Fig. A-4(b)). Again, we obtain very good agreement between the machine learning approach and the reference solution, which has a non-trivial form, as the bubbles seem to cluster

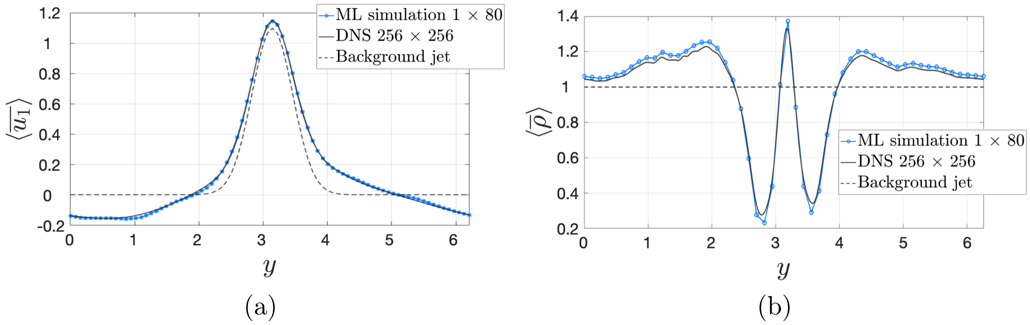


FIG. A-4. Time-averaged profile of $\overline{u}_1$ (left) and $\bar{\rho}$ (right) for the one-dimensional cTCN closure model (blue line) and the DNS (black line). The shape of the jet that is imposed by the large-scale forcing is depicted with dashed line. The simulation corresponds to Re = 1000 while training included Re $\in \{650, 750, 850\}$.

around the core of the jet and be repelled from the adjacent areas of the jet core.

A.1.2 Testing generalizability on bimodal jets

Next we test the generalizability of the closure schemes presented in the previous section on bimodal jets. Once again we state that we train on unimodal jets (as described previously) while we test our scheme on bimodal jets with the unperturbed jet-structure of the fluid flow chosen (as seen in Sec. 4.2) as

$$u_1 = \exp[-3(y-0.8\pi)^2] + \exp[-3(y-1.2\pi)^2], \quad \mathrm{Re} \in [500, 1000]. \tag{A.9}$$

In Fig. A-5 we present the normalized mean-square error Eq. A.7 between the reference solution and the one predicted by the closure model. As we can see, employing the physical constraint during training significantly improves the results. Specifically, we note that the errors of the cTCN and cLSTM remain at the same levels observed in the unimodal jet case (i.e. we have good generalizability properties in

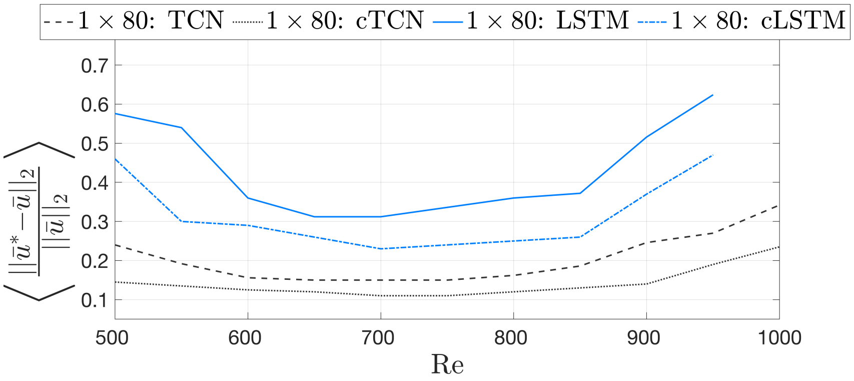
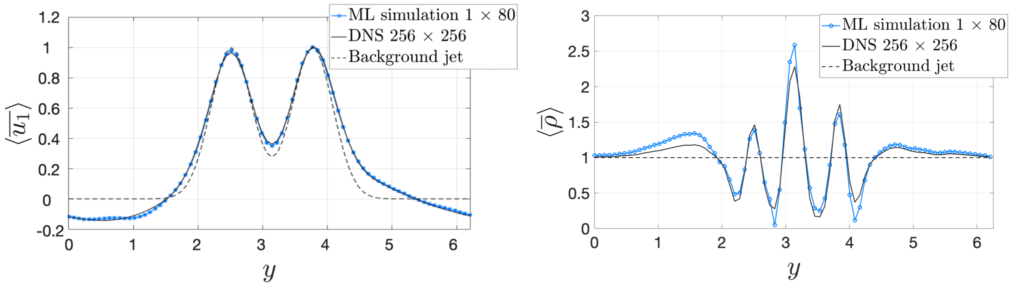


FIG. A-5. Normalized error Eq. A.7 for one-dimensional closure models error for one-dimensional closures applied on bimodal jets. Training data includes unimodal jets with $\text{Re} = \{650, 750, 850\}$.

different flows), while the unconstrained version has significantly worse performance compared with the unimodal jet case. In addition, we observe much better behavior of the constrained closures when we move to Reynolds numbers higher than the ones used for training. In fact, for sufficiently large Reynolds the schemes based on the unconstrained closures become unstable.

Results regarding the time-averaged jet profile of $\overline{u}_1$ and ρ are depicted in Fig. A-6 for the cTCN architecture, where we can observe excellent agreement of the predicted profile with DNSs. We also apply the closure scheme on the transport Eq. 4.7 to compute the distribution of bubbles. We present the comparison of the mean distribution of bubbles between the cTCN closure model and DNS in Fig. A-6(b). We note that the error for this case is slightly more pronounced compared with the one observed for the mean flow velocity in Fig. A-6(a). This can be attributed to two factors: (i) the predictions of the transport model rely on the predictions of the coarse-scale model for the velocity field (hence error accumulates); and (ii) the closure model for the bubbles relies only on data since the energy-preserving constraint is not relevant.



(a)

(b)

FIG. A-6. Time-averaged profile of $\overline{u}_1$ and $\bar{\rho}$ for the one-dimensional cTCN closure model (blue line) and the DNS (black line). The simulation corresponds to Re $=1000$ and a bi-model background jet, while training data for the closures correspond to unimodal jets with Re $\in \{650, 750, 850\}$. The shape of the jet that is imposed by the large-scale forcing is depicted with dashed line.

A summary of the error improvement in the one-dimensional predictions due to the adoption of the physical constraint is presented in Table A.3. We present the improvement of the mean-square error for the mean flow, averaged over different Reynolds numbers. For all cases this percentage ranges between 23% and 29% with the improvement being more pronounced for the bimodal setups, i.e. the setup that was not used for training.

Table A.3. Error decrease (Reynolds-averaged) due to the physical constraint for one-dimensional closure schemes.

Architecture	Jet-type	Error decrease
TCN-1D	Unimodal	23%
LSTM-1D	Unimodal	25%
TCN-1D	Bimodal	29%
LSTM-1D	Bimodal	27%

Appendix B

Nonintrusive statistical corrections of climate models

B.1 Further Data assimilation results for climate model

For the next numerical validation, the focal point shifts to extreme events. Again we plan to use as input nudged data of certain events. These data have not been used for training. The goal is to see how well the extreme event can be reproduced from the trained ML scheme if the input is just nudged data.

The first extreme event under discussion is a tropical cyclone named hurricane Sandy. Hurricane Sandy impacted the Caribbeans and the East Coast of the United States from 10/23 to 11/02 of 2012. For this case, training takes place for all of 2012, excluding the time-interval 10/22 to 11/03. The neural-network is again trained for 1000 epochs. In fig. B-1, the maximum wind-speed for time interval 10/22-11/03 is shown over the spatial region of $[40^o\text{W}, 100^o\text{W}] \times [0^o\text{N}, 50^o\text{N}]$. The first row shows ERA5 predictions. Second row shows the Nudged prediction together with the Nudged biases with respect to ERA5 data. The last column shows the same results for corrections applied to the Nudged data via the trained neural-network.

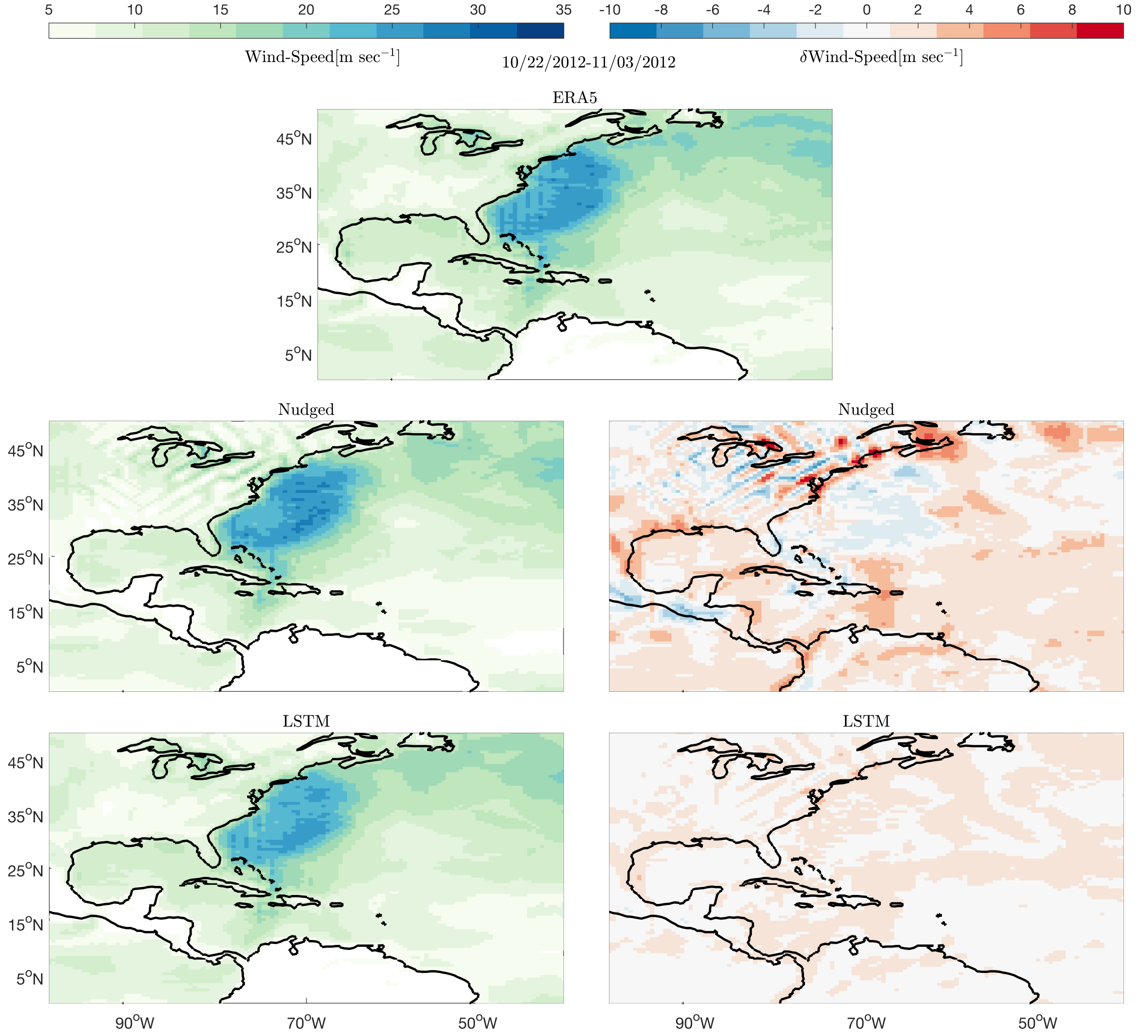


FIG. B-1. Maximum wind-speed predictions for different datasets over the time-period 11/22-11/03. Results are shown for ERA5, Nudged and neural-network corrections. Left column corresponds to predictions and right column to biases with respect to ERA5.

The model displays a clear ability to reduce biases over the time-interval it was not trained, showcasing the generalizability of the scheme. Furthermore, Nudged data display some form of aliasing over the continental US and Canada. These effects are significantly removed via the neural-network corrections.

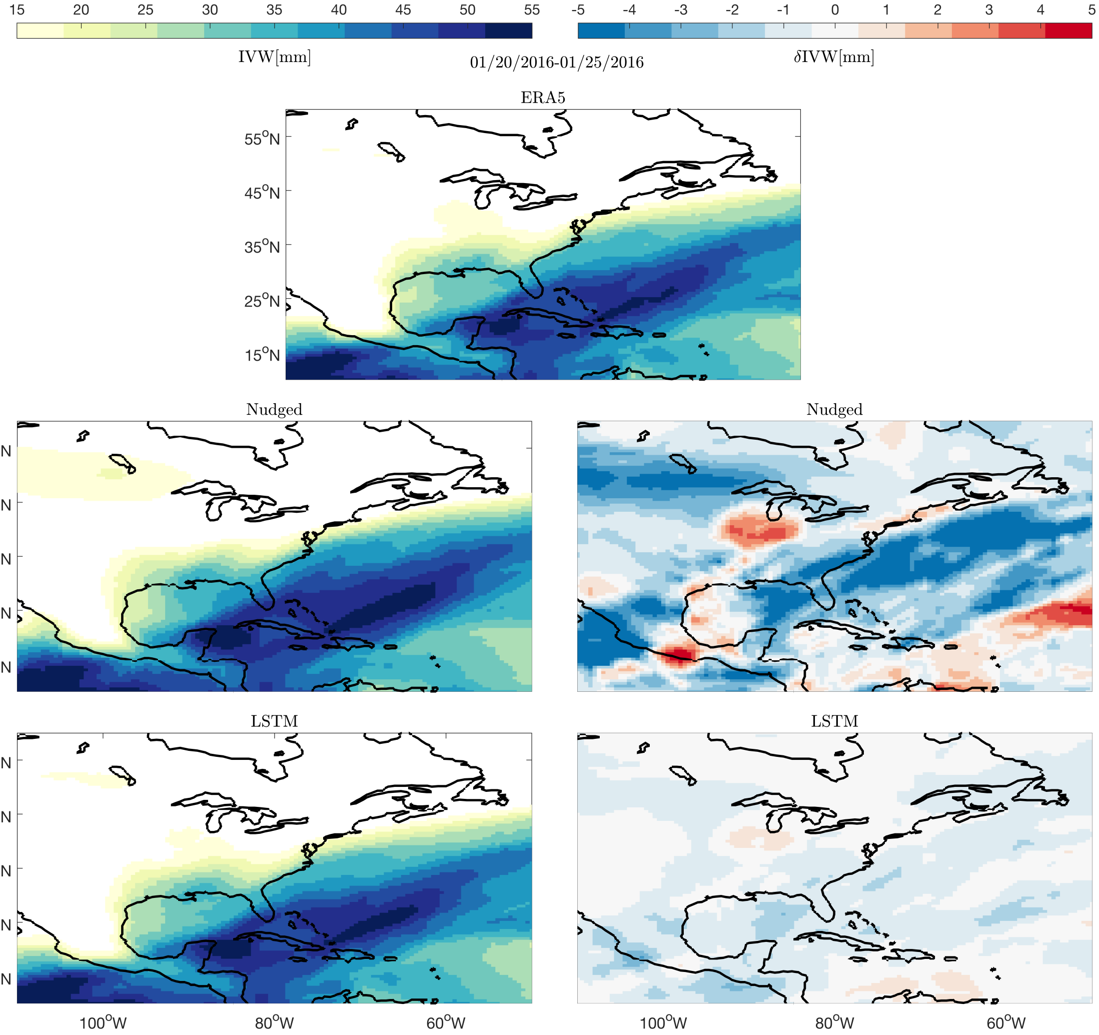


FIG. B-2. Maximum integrated water vapor predictions for different datasets over the time-period 01/20-01/26. Results are shown for ERA5, Nudged and neural-network corrections. Left column corresponds to predictions and right column to biases with respect to ERA5.

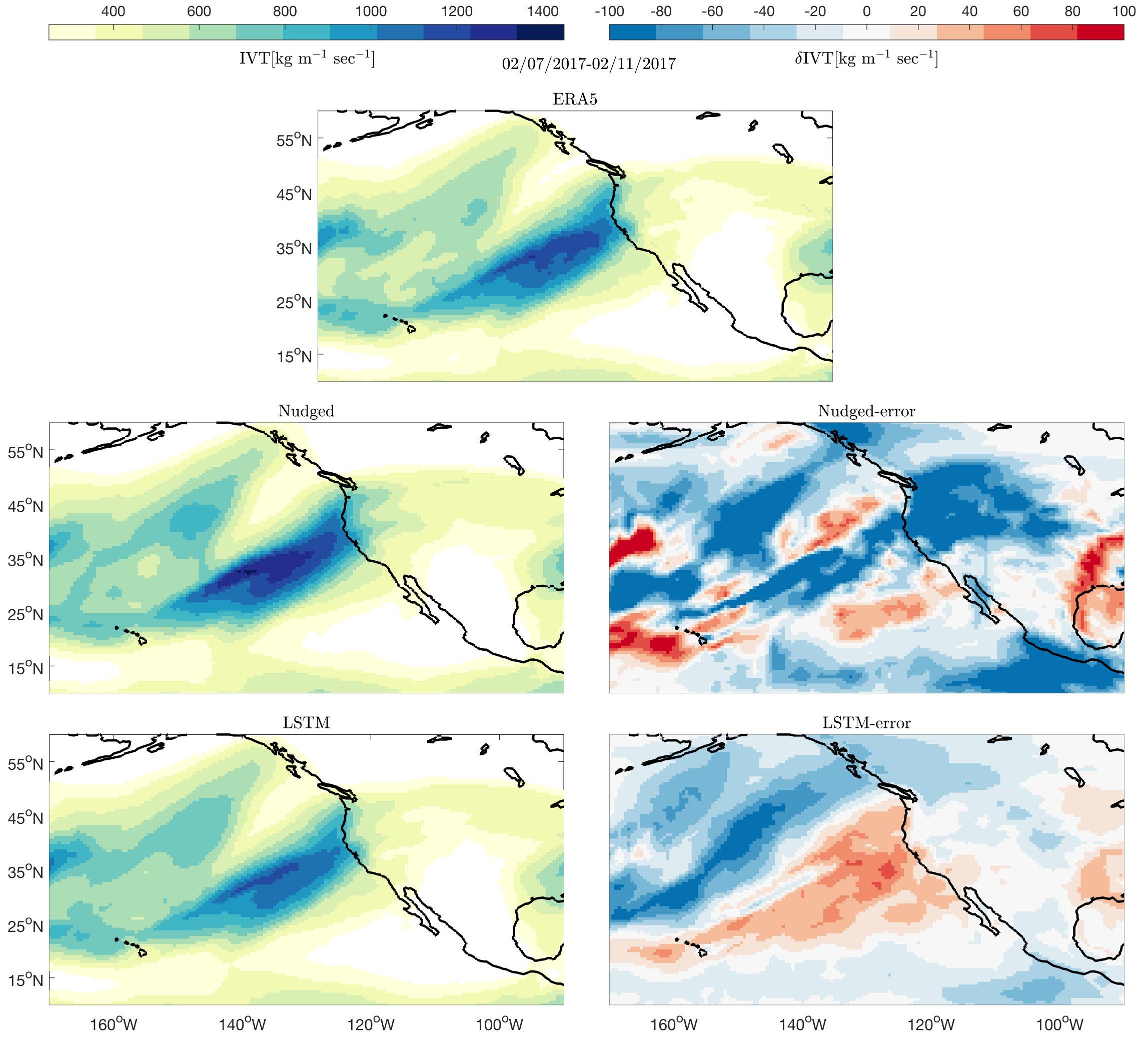


FIG. B-3. Maximum integrated vapor transport predictions for different datasets over the time-period 02/04-02/14. Results are shown for ERA5, Nudged and neural-network corrections. Left column corresponds to predictions and right column to biases with respect to ERA5.

Moving forward to other climate extreme events for numerical validation, we focus on a particular extratropical cyclone. The extreme event under discussion is an extratropical cyclone that caused a category 4 snowstorm over the East Coast from 01/20 to 01/26 of 2016. For this case, training takes place for all of 2016, excluding the time-interval 01/20 to 01/26. The neural-network is again trained for 1000 epochs. In fig. B-2, the maximum integrated water vapor for time interval 01/20-01/26 is shown over the spatial region of $[50^o\mathrm{W}, 110^o\mathrm{W}] \times [10^o\mathrm{N}, 60^o\mathrm{N}]$. The first row shows ERA5 predictions. Second row shows the Nudged prediction together with the Nudged biases with respect to ERA5 data. The last column shows the same results for corrections applied to the Nudged data via the trained neural-network. The generalizability of the learned mapping is showcased again here. It is also noted that IVW is not a quantity directly predicted by the hybrid model. It is an integral quantity that is derived from the predictive fields.

As a final study of a speficic extreme event, an atmospheric river is chosen. The study focuses on an atmospheric river that impacted the West Coast of the United States from 02/04 to 02/14 of 2017. For this case, training takes place for all of 2017, excluding the time-interval 02/04 to 02/14. The LSTM is again trained for 1000 epochs. In fig. B-3, the maximum integrated vapor transport for time interval 02/04-02/14 is shown over the spatial region of $[90^o\mathrm{W}, 170^o\mathrm{W}] \times [10^o\mathrm{N}, 60^o\mathrm{N}]$. The first row shows ERA5 predictions. Second row shows the Nudged prediction together with the Nudged biases with respect to ERA5 data. The last column shows the same results for corrections applied to the Nudged data via the trained neural-network. Integrated vapor transport is defined as

$$\mathrm{IVT} = \sqrt{\left(\frac{1}{g}\int_{P_{\mathrm{surf}}}^{P_0} U^2 Q \mathrm{d}P\right)^2 + \left(\frac{1}{g}\int_{P_{\mathrm{surf}}}^{P_0} V^2 Q \mathrm{d}P\right)^2}. \tag{B.1}$$

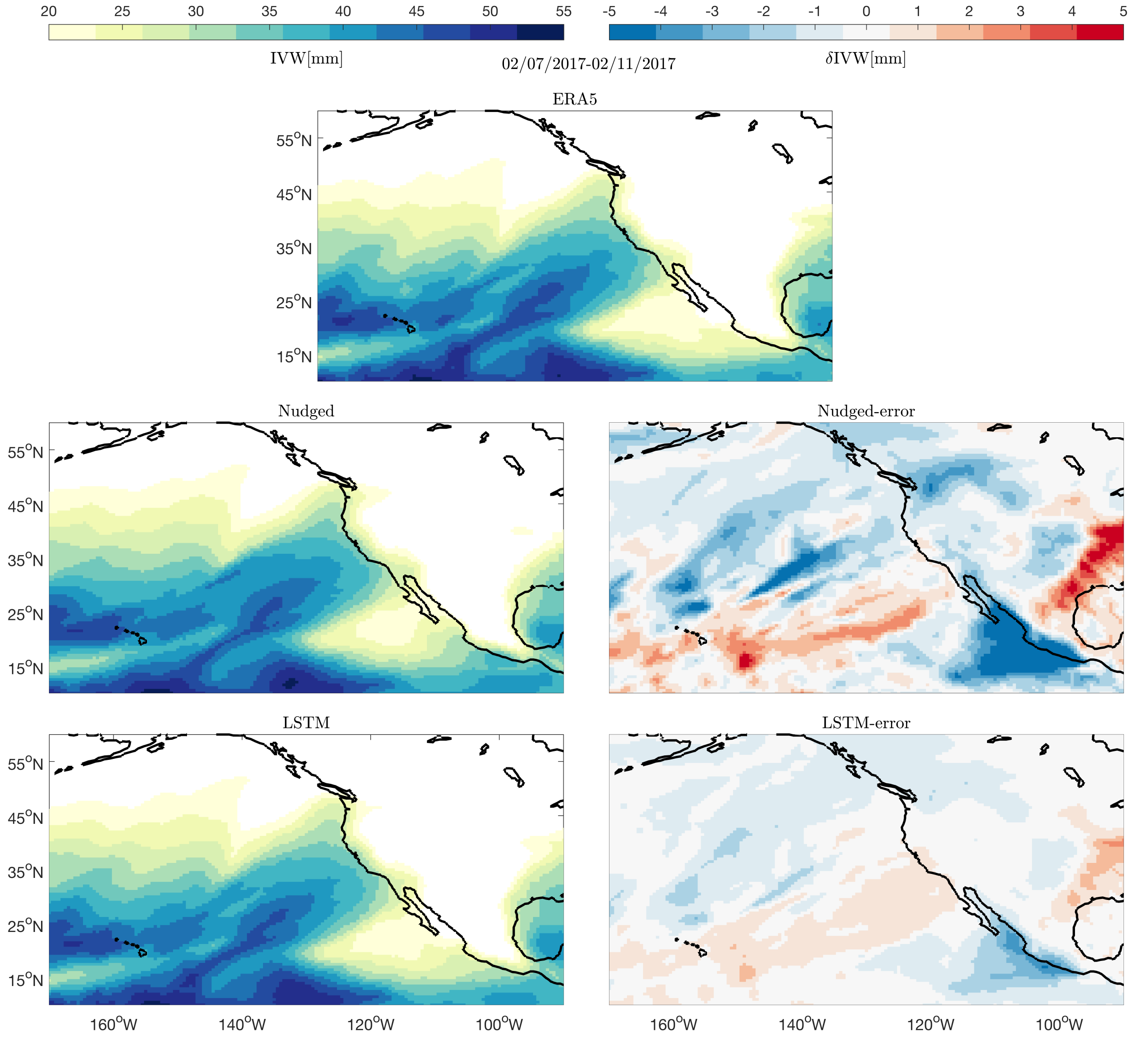
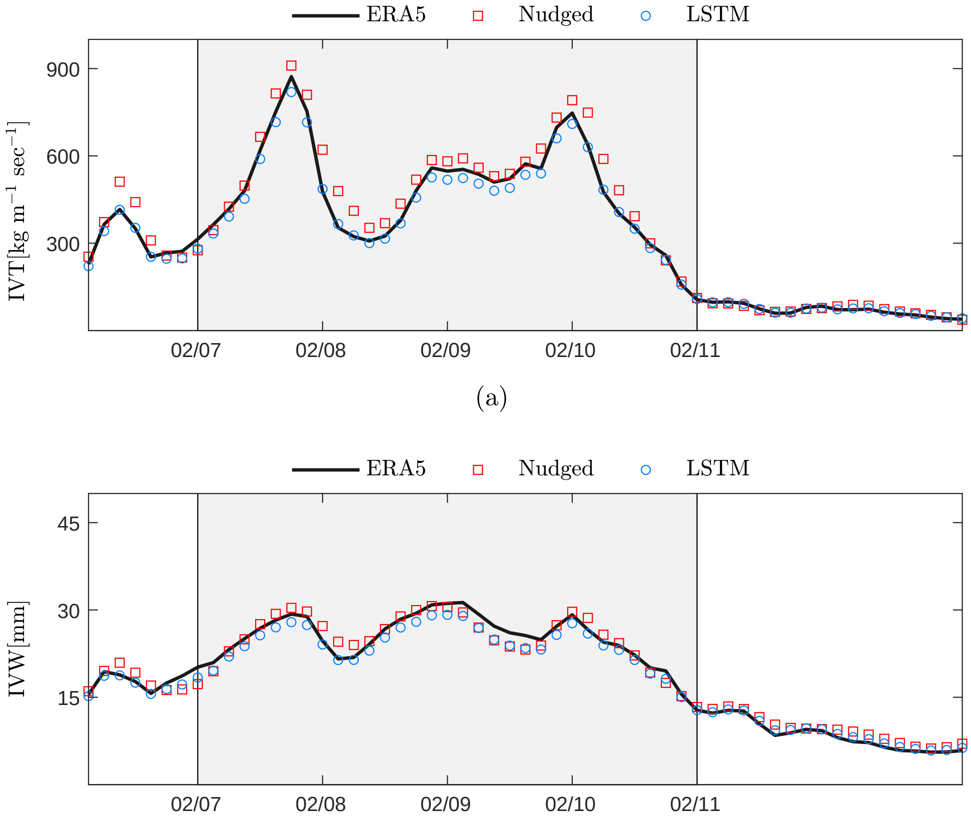


FIG. B-4. Maximum integrated water vapor predictions for different datasets over the time-period 02/04-02/14. Results are shown for ERA5, Nudged and neural-network corrections. Left column corresponds to predictions and right column to biases with respect to ERA5.

The model displays a clear ability to reduce biases over the time-interval it was not trained, showcasing the generalizability of the scheme.

In fig. B-4, the maximum integrated vapor transport for time interval 02/04-02/14 is shown over the spatial region of $[90^o\text{W}, 170^o\text{W}] \times [10^o\text{N}, 60^o\text{N}]$. The first row shows ERA5 predictions. Second row shows the Nudged prediction together with the Nudged biases with respect to ERA5 data. The last column shows the same results for corrections applied to the Nudged data via the trained neural-network. Integrated vapor transport is defined as

$$\mathrm{IVW} = \frac{1}{g}\int_{P_{\mathrm{surf}}}^{P_0} Q \mathrm{d}P. \tag{B.2}$$



(a)

(b)

FIG. B-5. Predictions of (a) IVT and (b) IVW, over the Orville Dam for the time period 02/04 to 02/14 of 2017.

The model displays a clear ability to reduce biases over the time-interval it was not trained, showcasing the generalizability of the scheme. Finally, the time-series for IVT and IVW over the Orville Dam, during the same time period is shown in fig. B-5. The Orville Dam in California suffered a spillway on 02/11/2017, damaging its concrete surface. This was caused by the excess rain during the occurence of the atmospheric river. The model effectively tracks the ERA5 predictions for the duration of the extreme event.

B.2 Active search

Throughout chapter 6, the presented non-intrusive models were trained by randomly sampling time-intervals from the 11-year available Nudged dataset. This proposed methodology showcased its ability to accurately capture the long-time statistics of extreme events. However, there is no theoretical or heuristic guarantee that the chosen training dataset will yield optimal or even adequate results. This is due to the uncertainty of (i) what properties constitute a training point "useful"; and (ii) are these "useful" points included in the randomly chosen training dataset. Without this information included during training, the model's performance will be hindered during testing. That is because in a supervised learning setup, only during training can gradient-descent schemes optimize the weights of the neural-networks. As a result, it is of outmost importance to ensure all relevant information is included during training. Furthermore, it is desirable to recognise redundant data and draw theoretical conclusions.

To improve the training process, we will use Bayesian sequential selection algorithms similar to Bayesian experimental design (BED). These algorithms are widely use to identify the critically informative training data while disregarding those that are unrelated or misleading. This approach dynamically couples the model and data based on its merits towards improving the aforementioned model, and evaluates the performance of the model on the full training set which guarantees that no key information is wasted. One scenario where this approach is utilized is to avoid the double-descent phenomenon, a phenomenon in which the accuracy of a ML model deteriorates before it improves as the model complexity, training set size or, in case of deep neural networks (DNNs). A recent study by Pickering and Sapsis [131] demonstrates the efficacy of this approach, referred to as information FOMO (Fear Of Missing Out) in reducing the training dataset size to its most informative data and consequently, rectifying the double descent phenomenon. The physical system studied in this article concerns a one-dimensional partial differential equation originally proposed by Majda, McLaughlin, and Tabak (MMT) [100] for the study of 1D wave turbulence.

The goal of this study is to identify optimal training datasets for predicting accurate statistics of extreme events for different climate free-runs. This is particularly important for this application as we have a vast set of data points and outputs, which if used inefficiently can use to large wastes of computational time. It is also expected that different data-points will be crucial for different target extreme events. As a result, a number of different targets are described in the next subsection, and detailed results will be given in the future reports.

For any form of value of data analysis, a method to estimate uncertainty in the predictions of the trained models is needed. To quantify uncertainty in the neural network predictions, an ensemble of neural networks, of size N, is employed. These neural networks are trained over the same dataset and have the same architecture and hyperparameters. The only difference between them is that the weights of each member of the ensemble are initialized with different values sampled from a random process. Using this ensemble, model uncertainty can be quantified via the variance

$$\sigma(\mathbf{X}) = \sqrt{\sum_{n=1}^{N} \frac{\left(Y_n(\mathbf{X}) - \overline{Y}(\mathbf{X})\right)^2}{N-1}}, \tag{B.3}$$

where $\mathbf{X}$ corresponds to the input data and Y is the quantity of interest. Y_n is the predictions of the nth neural network of the ensemble, and

$$\overline{Y}(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^{N} Y_n(\mathbf{X}). \tag{B.4}$$

This metric can be used to asses the statistical characteristics of spatio-temporally local generalization errors as well as uncertainty in non-local quantities, such as PCA modes and global pdf predictions.

Data value (or importance) will be determined by the score that the *acquisition* function assigns to each data. The acquisition function we will adopt is given by

$$a(\mathbf{X}) = \sigma^2(\mathbf{X}) \frac{p_x(\mathbf{X})}{p_y(y(\mathbf{X}))}, \tag{B.5}$$

where $\mathbf{X}$ is the input data and y is the output or quantity of interest. This output y or quantity of interest can be for example, the temperature over a specific region, the density count for cyclones. The acquisition function above consists of two terms, $\sigma^2(\mathbf{X})$ and $p_x(\mathbf{X})/p_y(y(\mathbf{X}))$. The former, attempts to assign larger importance to those regions of the input domain that haven't been populated with enough data. The latter, however, considers those regions of the dataset important where rare and extreme events occur, i.e., $p_y(y(\mathbf{X})) \ll 1$. All together, the acquisition function mentioned above, considers subdomains of the dataset valuable, if 1) they are not populated with data points, and 2) the model predicts they contain extreme and rare events.

Of course, the aforementioned acquisition function can and will be tailored in such a way that is most suitable to our needs, for instance, it can be modified such that it deems a subset of the data valuable if, for instance, it:

- improves the accuracy and predictive capacity of the model over local regions of the globe, e.g., a particular city, state, country, region, etc.;
- helps the model to accurately predict extreme local events such as hurricanes;
- results in accurate modeling of the spatial or temporal mean fields of variables of interest.

This approach can then be used to augment the training set with training samples with the highest acquisition function score. Repeating this process for multiple iterations can yield an optimal training set for the neural network. The iterative process for active sampling can be seen in fig. B-6.

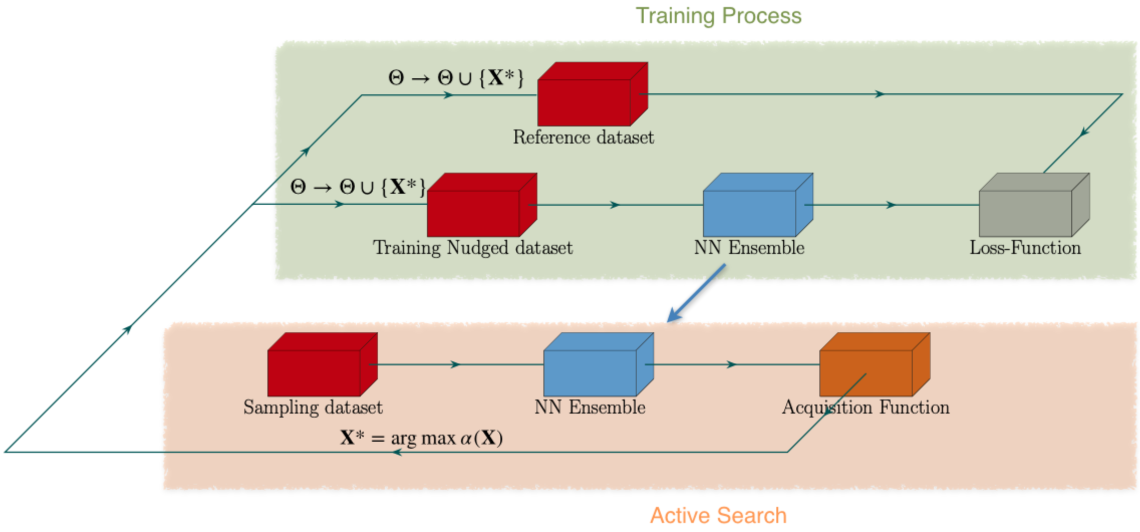


FIG. B-6. Schematics of the iterative active sampling process. Training process (top) and active search (bottom), for the non-intrusive hybrid method.

First, a small training set denoted by Θ is randomly selected. An ensemble of neural networks are trained over Θ. Then, the trained neural networks are used to predict the proposed acquisition function for a target quantity y. The acquisition function is computed over the entire sampling dataset. Once the largest values of the acquisition function are found, the datapoints are added to the training set Θ. The number of datapoints added to the training set a hyperparemeter of the scheme. The neural networks are then trained over the new training set, with re-initialized weights. The process is repeated until satisfactory accuracy has been achieved.

B.2.1 Forecast Targets

With respect to forecast targets, PCA modes for a variety of physical quantities are examined. The purpose of using PCA modes is to reduce the dimensionality of the target quantity y in the acquisition function. Below, preliminary results for the first PCA mode of temperature are shown. Specifically, the first PCA mode is computed over the horizontal region $[0^o\text{N}, 60^o\text{N}] \times [40^o\text{W}, 110^o\text{W}]$, for the sigma-level closest to the surface of the earth.

Implementing the scheme proposed in fig. B-6, two distinct choices can be made for "Sampling dataset", during the active search step. While training can only be carried out over the Nudged dataset, the acquisition function can be computed either over the Nudged dataset or the free-running CLIM dataset. As a result, two sampling approaches are tested here. Sampling Approach #1 involves using the Nudged dataset to compute the acquisition function. Then, the samples with the largest acquisition function value are directly added to the training set. On the other hand, Sampling Approach #2 involves the free-running CLIM dataset for the acquisition function. With this approach however, the samples with the largest acquisition function value cannot be directly incorporated in training. Instead, the selected samples are used to find the training samples withing the Nudged dataset that are closest to the selected free-running ones.

The two methods are compared to validate that using only the Nudged dataset during training can indeed help as learn vital information for the testing free-running datasets as well. Results are shown in fig. B-7 for an ensemble of $N = 5$.

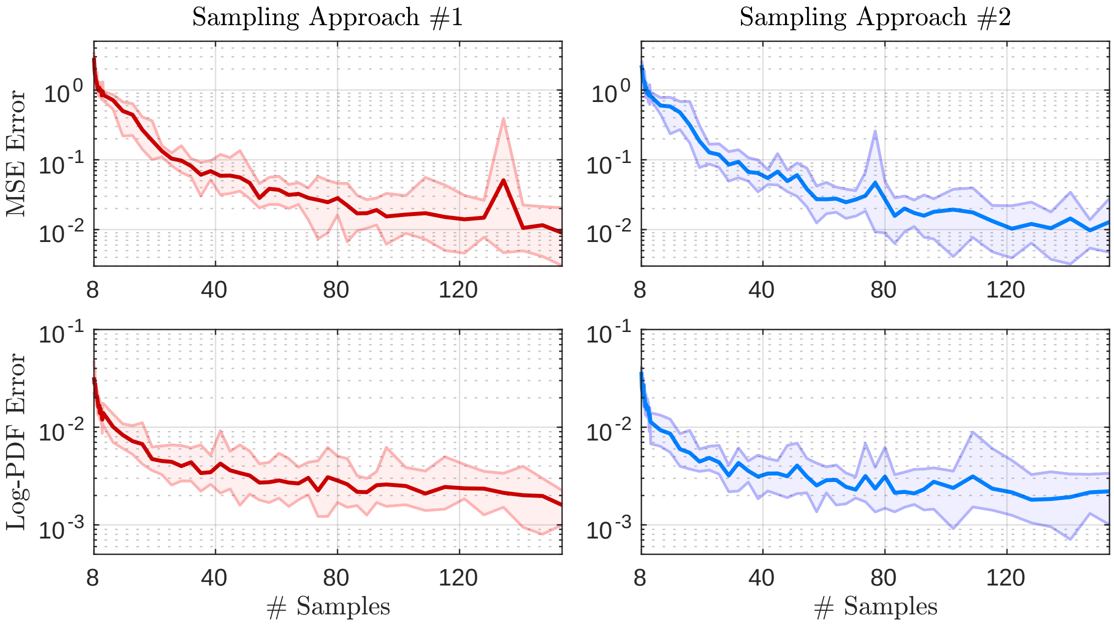


FIG. B-7. Top row: MSE Test error (i.e. out-of-sample data) as a function of training samples. Bottom row: Log-PDF Test error as a function of training samples. Results are shown for an ensemble of 5 neural-networks. Solid lines correspond to mean prediction over an ensemble of 5 realizations. Shaded areas engulf the predictions of all 5 realizations.

In fig. B-7, two metrics are shown. The top row shows convergence of the mean-square-error

$$\text{MS Error} = \frac{1}{|\mathcal{D}_{\text{sample}} \setminus \mathcal{D}_{\text{train}}|} \sum_{\mathbf{X} \in \mathcal{D}_{\text{sample}} \setminus \mathcal{D}_{\text{train}}} \left(\overline{Y}(\mathbf{X}) - Y^{\text{ref}}(\mathbf{X}) \right)^2 , \tag{B.6}$$

where Y^{ref} corresponds to the PCA predictions from the ERA5 dataset. Neural-netowrk predictions are derived by passing through all snapshots of the input data. Then, only the samples not included in the training process are used for the error calculation. This metric corresponds to the proposed metric I of section 3.1 from Milestone 7. The second metric (second row), shows the convergence of the log-pdf error

$$\text{Log-PDF Error} = \frac{1}{Y_{\text{max}} - Y_{\text{min}}} \int_{Y_{\text{min}}}^{Y_{\text{max}}} |\log_{10} p_{\mathcal{M}} \left(y; \mathcal{D}_{\text{train}} \right) - \log_{10} p_{\text{ref}}(y)| \mathrm{d}y. \tag{B.7}$$

This metric corresponds to the proposed metric III of section 3.1 from Milestone 7. The two methods exhibit the same rate of convergence for both metrics, showcasing that the two schemes are equivalent. Hence, we can assume that the optimal dataset selected over the Nudged dataset will contain information important for the free-running dataset during testing.

In fig. B-8, a comparison between Active Sampling and standard Monte Carlo sampling is shown. The same two metrics are used. With respect to the mean-square error, both methods appear to converge at the same rate. However, active sampling converges faster with the respect to the log-PDF error. These two results hint that the active sampling approach allows for faster sampling of extreme events in the behaviour of the dominant PCA mode. This behaviour can be evaluated on other quantities as well, to see its range of applicability. Of particular interest is the convergence properties for PCA modes of other physical quantities as well as indicators that are used as priors for extreme events.

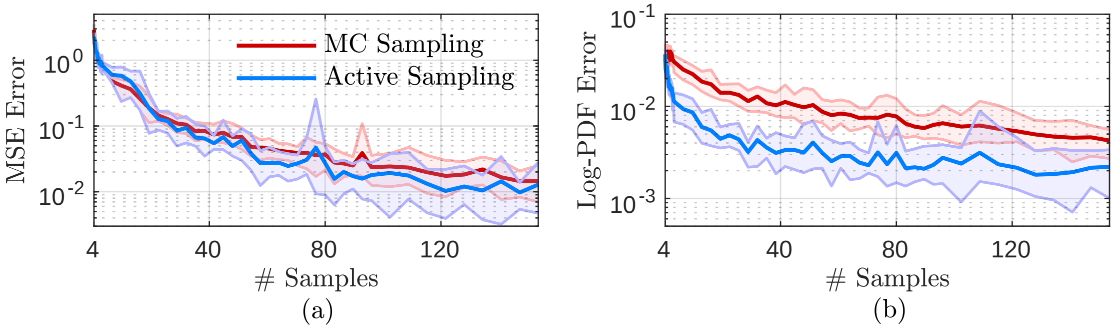


FIG. B-8. (a): MSE Test error (i.e. out-of-sample data) as a function of training samples. (b): Log-PDF Test error as a function of training samples. Results are shown for an ensemble of 5 neural-networks. Solid lines correspond to mean prediction over an ensemble of 10 realizations. Shaded areas engulf the predictions for all 5 realizations.

Appendix C

Hybrid Quadrature-based method of moments

C.1 CHyQMOM inversion algorithm

This is the inversion algorithm for the 4-node CHyQMOM scheme. Given the first and second-order moments $\{\mu_{1,0}, \mu_{0,1}, \mu_{2,0}, \mu_{1,1}, \mu_{0,2}\}$ it computes the nodes $(\xi_i, \dot{\xi}_i)$ and corresponding weights w_i for $i = 1, 2, 3, 4$, in phase-space. In this work we assume $\mu_{0,0} = 1$. To tail the algorithm to our hybrid scheme of arbitrary number of quadrature nodes, the algorithm adds some fictitious extra nodes to the scheme with zero-valued weights to match the desired number of nodes of the hybrid scheme.

Algorithm 4: CHyQMOM inversion algorithm

```
$w_i = 0.25, \quad 1 \leq i \leq 4.$;
$w_i = 0.00, \quad 4 < i \leq N$;
$\sigma_R = \sqrt{\mu_{2,0} - \mu_{1,0}^2}$;
$\alpha = \frac{\mu_{1,1} - \mu_{1,0}\mu_{0,1}}{\sigma_R}$;
$\sigma_{\dot{R}} = \sqrt{\mu_{0,2} - \alpha^2 - \mu_{0,1}^2}$;
$\xi_i = \mu_{1,0} + \sigma_R$;
$\xi_2 = \mu_{1,0} + \sigma_R$;
$\xi_3 = \mu_{1,0} - \sigma_R$;
$\xi_4 = \mu_{1,0} - \sigma_R$;
$\xi_i = \mu_{1,0}, \quad 4 < i \leq N$;
$\dot{\xi}_1 = \mu_{0,1} + \alpha + \sigma_{\dot{R}}$;
$\dot{\xi}_2 = \mu_{0,1} + \alpha - \sigma_{\dot{R}}$;
$\dot{\xi}_3 = \mu_{0,1} - \alpha + \sigma_{\dot{R}}$;
$\dot{\xi}_4 = \mu_{0,1} - \alpha - \sigma_{\dot{R}}$;
$\dot{\xi}_i = \mu_{0,1}, \quad 4 < i \leq N$;
```

C.2 Neural network hyperparameters

Hyperparameter	Value
Epochs	500
Learning rate	10^{-5}
Batch size	32
Activation function	tanh
Recurrent activation function	hard sigmoid
Dropout coefficient	0.10
Recurrent dropout coeffificent	0.10
LSTM is stateful	True
Kernel initializer	Zeros
Recurrent initializer	Zeros
Bias initializer	Zeros
hidden units	28

Table C.1. Hyperparameters used to train the neural networks.

C.3 Evolution of mean-square error

Evolution of L^2 error as time progresses, defined as

$$E_{l,m}^{(*)}(k) = \sqrt{\frac{\sum_{i=1}^{k} \left[\mu_{l,m}^{(*)}(t_i) - \mu_{l,m}^{(\mathrm{MC})}(t_i)\right]^2}{\sum_{i=1}^{k} \left[\mu_{l,m}^{(\mathrm{MC})}(t_i)\right]^2}}, \tag{C.1}$$

where (MC) indicates Monte Carlo surrogate truth data and $* = \{\mathrm{ML}, \mathrm{QBMM}\}$ and k denotes the last time-step till which the error is calculated. Results are presented for Case (2) of Figure 7-4 and Figure 7-5.

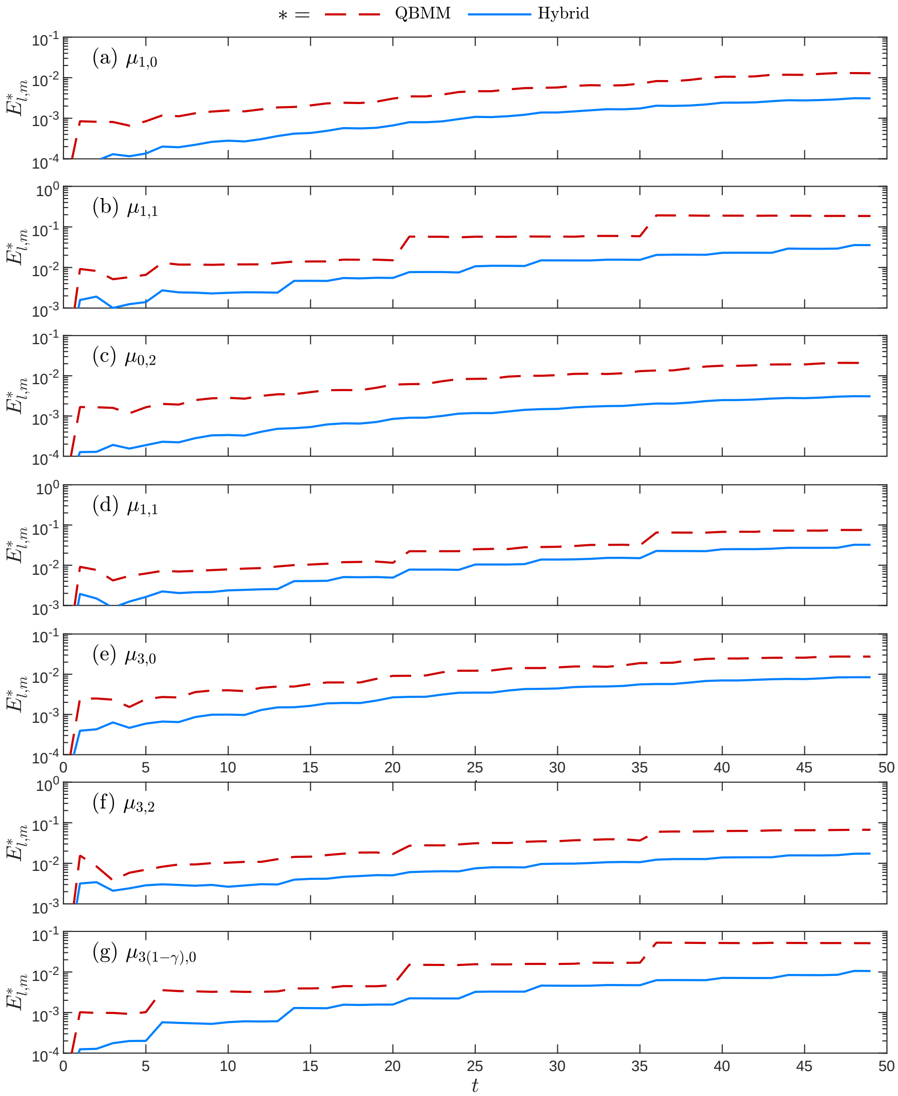


FIG. C-1. Evolution of L^2 error for different moments for a particular forcing realization.

Bibliography

[1] Rafail V Abramov and Andrew J Majda. Blended response algorithms for linear fluctuation-dissipation for complex nonlinear dynamical systems. *Nonlinearity*, 20(12):2793, 2007.

[2] Rafail V Abramov and Andrew J Majda. A new algorithm for low-frequency climate response. *Journal of the Atmospheric Sciences*, 66(2):286–309, 2009.

[3] Ali Akhavan-Safaei, Mehdi Samiee, and Mohsen Zayernouri. Data-driven fractional subgrid-scale modeling for scalar turbulence: A nonlocal les approach. *Journal of Computational Physics*, 446:110571, 2021.

[4] Samuel Al. Some studies in machine learning using the game of checkers. *IBM journal on Research and Development*, 3:210–229, 1959.

[5] Keita Ando, Tim Colonius, and Christopher E Brennen. Numerical simulation of shock propagation in a polydisperse bubbly liquid. *International Journal of Multiphase Flow*, 37(6):596–608, 2011.

[6] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271*, 2018.

[7] MR Bailey, VA Khokhlova, OA Sapozhnikov, SG Kargl, and LA Crum. Physical mechanisms of the therapeutic effect of ultrasound (a review). *Acoustical Physics*, 49(4):369–388, 2003.

[8] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

[9] Michael Balthasar and Markus Kraft. A stochastic approach to calculate the particle size distribution function of soot particles in laminar premixed flames. *Combustion and Flame*, 133(3):289–298, 2003.

[10] Carlos Barajas and Eric Johnsen. The effects of heat and mass diffusion on freely oscillating bubbles in a viscoelastic, tissue-like medium. *The Journal of the Acoustical Society of America*, 141(2):908–918, 2017.

[11] Raymond T Bauer. *Remarkable shrimps: adaptations and natural history of the carideans*, volume 7. University of Oklahoma Press, 2004.

[12] Yoshua Bengio, Patrice Simard, Paolo Frasconi, et al. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[13] Thomas Bengtsson, Peter Bickel, and Bo Li. Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. In *Probability and statistics: Essays in honor of David A. Freedman*, pages 316–334. Institute of Mathematical Statistics, 2008.

[14] Guido Boffetta and S Musacchio. Evidence for the double cascade scenario in two-dimensional turbulence. *Physical Review E*, 82(1):016307, 2010.

[15] Vadim Borue. Spectral exponents of enstrophy cascade in stationary two-dimensional homogeneous turbulence. *Physical review letters*, 71(24):3967, 1993.

[16] Annalisa Bracco and James C McWilliams. Reynolds-number dependency in homogeneous, stationary two-dimensional turbulence. *Journal of Fluid Mechanics*, 646:517–526, 2010.

[17] Michal Branicki and Andrew J Majda. Fundamental limitations of polynomial chaos for uncertainty quantification in systems with intermittent instabilities. *Communications in mathematical sciences*, 11(1):55–103, 2013.

[18] Christopher E Brennen. *Cavitation and bubble dynamics.* Cambridge University Press, 2014.

[19] Christopher Earls Brennen. Cavitation in medicine. *Interface focus*, 5(5):20150022, 2015.

[20] Steven L Brunton, Maziar S Hemati, and Kunihiko Taira. Special issue on machine learning and data-driven methods in fluid dynamics. *Theoretical and Computational Fluid Dynamics*, 34(4):333–337, 2020.

[21] Spencer H Bryngelson, Alexis Charalampopoulos, Themistoklis P Sapsis, and Tim Colonius. A Gaussian moment method and its augmentation via LSTM recurrent neural networks for the statistics of cavitating bubble populations. *International Journal of Multiphase Flow*, page 103262, 2020.

[22] Spencer H Bryngelson and Tim Colonius. Simulation of humpback whale bubble-net feeding models. *The Journal of the Acoustical Society of America*, 147(2):1126–1135, 2020.

[23] Spencer H Bryngelson, Tim Colonius, and Rodney O Fox. QBMMlib: A library of quadrature-based moment methods. *SoftwareX*, 12:100615, 2020.

[24] Spencer H Bryngelson, Kevin Schmidmayer, and Tim Colonius. A quantitative comparison of phase-averaged models for bubbly, cavitating flows. *International Journal of Multiphase Flow*, 115:137–143, 2019.

[25] Peter M Caldwell, Christopher S Bretherton, Mark D Zelinka, Stephen A Klein, Benjamin D Santer, and Benjamin M Sanderson. Statistical significance of climate sensitivity predictors obtained by data mining. *Geophysical Research Letters*, 41(5):1803–1808, 2014.

[26] Steven L Ceccio and Christopher E Brennen. Observations of the dynamics and acoustics of travelling bubble cavitation. *Journal of Fluid Mechanics*, 233:633–660, 1991.

[27] A Charalampopoulos and T Sapsis. Uncertainty quantification of turbulent systems via physically consistent and data-informed reduced-order models. *Physics of Fluids*, 34(7):075120, 2022.

[28] Alexis-Tzianni Charalampopoulos, Spencer H Bryngelson, Tim Colonius, and Themistoklis P Sapsis. Hybrid quadrature moment method for accurate and stable representation of non-gaussian processes and their dynamics. *arXiv preprint arXiv:2110.01374*, 2021.

[29] Alexis-Tzianni G Charalampopoulos and Themistoklis P Sapsis. Machine-learning energy-preserving nonlocal closures for turbulent fluid flows and inertial tracers. *arXiv preprint arXiv:2102.07639*, 2021.

[30] Alexis-Tzianni G Charalampopoulos and Themistoklis P Sapsis. Machine-learning energy-preserving nonlocal closures for turbulent fluid flows and inertial tracers. *Physical Review Fluids*, 7(2):024305, 2022.

[31] Abhijit Chatterjee and Dionisios G Vlachos. An overview of spatial microscopic and accelerated kinetic monte carlo methods. *Journal of computer-aided materials design*, 14(2):253–308, 2007.

[32] Tianping Chen and Hong Chen. Approximations of continuous functionals by neural networks with application to dynamic systems. *IEEE Transactions on Neural networks*, 4(6):910–918, 1993.

[33] Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, 1995.

[34] Andrew J Coleman, John E Saunders, Lawrence A Crum, and Mary Dyson. Acoustic cavitation generated by an extracorporeal shockwave lithotripter. *Ultrasound in medicine & biology*, 13(2):69–76, 1987.

[35] DT Crommelin and AJ Majda. Strategies for model reduction: Comparing different optimal bases. *Journal of the Atmospheric Sciences*, 61(17):2206–2217, 2004.

[36] Timothy DelSole and Brian F Farrell. The quasi-linear equilibration of a thermally maintained, stochastically excited jet in a quasigeostrophic model. *Journal of Atmospheric Sciences*, 53(13):1781–1797, 1996.

[37] John M Dennis, Jim Edwards, Katherine J Evans, Oksana Guba, Peter H Lauritzen, Arthur A Mirin, Amik St-Cyr, Mark A Taylor, and Patrick H Worley. Cam-se: A scalable spectral element dynamical core for the community atmosphere model. *The International Journal of High Performance Computing Applications*, 26(1):74–89, 2012.

[38] Benjamin Dollet, Philippe Marmottant, and Valeria Garbin. Bubble dynamics in soft and biological matter. *Annual Review of Fluid Mechanics*, 51:331–355, 2019.

[39] Xavier Escaler, Eduard Egusquiza, Mohamed Farhat, Francois Avellan, and Miguel Coussirat. Detection of cavitation in hydraulic turbines. *Mechanical systems and signal processing*, 20(4):983–1007, 2006.

[40] Rodney O Fox, Frédérique Laurent, and Aymeric Vié. Conditional hyperbolic quadrature method of moments for kinetic equations. *Journal of Computational Physics*, 365:269–293, 2018.

[41] Masataka Gamahara and Yuji Hattori. Searching for turbulence models by artificial neural network. *Physical Review Fluids*, 2(5):054604, 2017.

[42] R Gaudron, MT Warnez, and E Johnsen. Bubble dynamics in a viscoelastic medium with nonlinear elasticity. *Journal of Fluid Mechanics*, 766:54–75, 2015.

[43] Roger Ghanem. Probabilistic characterization of transport in heterogeneous media. *Computer Methods in Applied Mechanics and Engineering*, 158(3-4):199–220, 1998.

[44] Roger Ghanem and S Dham. Stochastic finite element analysis for multiphase flow in heterogeneous porous media. *Transport in porous media*, 32(3):239–262, 1998.

[45] Roger G Ghanem and Pol D Spanos. Spectral stochastic finite-element formulation for reliability analysis. *Journal of Engineering Mechanics*, 117(10):2351–2372, 1991.

[46] Roger G Ghanem and Pol D Spanos. *Stochastic finite elements: a spectral approach*. Courier Corporation, 2003.

[47] Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice*. CRC press, 1995.

[48] Jean-Christophe Golaz, Vincent E Larson, and William R Cotton. A pdf-based model for boundary layer clouds. part i: Method and model description. *Journal of the atmospheric sciences*, 59(24):3540–3551, 2002.

[49] Jean-Christophe Golaz, Luke P Van Roekel, Xue Zheng, Andrew F Roberts, Jonathan D Wolfe, Wuyin Lin, Andrew M Bradley, Qi Tang, Mathew E Maltrud, Ryan M Forsyth, et al. The doe e3sm model version 2: overview of the physical model and initial model evaluation. *Journal of Advances in Modeling Earth Systems*, 14(12), 2022.

[50] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[51] Andrey Gritsun and Grant Branstator. Climate response using a three-dimensional operator based on the fluctuation–dissipation theorem. *Journal of the atmospheric sciences*, 64(7):2558–2575, 2007.

[52] Andrey Gritsun, Grant Branstator, and Andrew Majda. Climate response of linear and quadratic functionals using the fluctuation–dissipation theorem. *Journal of the Atmospheric Sciences*, 65(9):2824–2841, 2008.

[53] Martin Hairer and Andrew J Majda. A simple framework to justify linear response theory. *Nonlinearity*, 23(4):909, 2010.

[54] George Haller and Themistoklis Sapsis. Where do inertial particles go in fluid flows? *Physica D: Nonlinear Phenomena*, 237(5):573–583, 2008.

[55] George Haller and Themistoklis Sapsis. Localized instability and attraction along invariant manifolds. *SIAM Journal on Applied Dynamical Systems*, 9(2):611–633, 2010.

[56] Peter Hamlington and Werner Dahm. Reynolds stress closure including nonlocal and nonequilibrium effects in turbulent flows. In *39th AIAA Fluid Dynamics Conference*, page 4162, 2009.

[57] John Hammersley. *Monte carlo methods.* Springer Science & Business Media, 2013.

[58] John M Hammersley. Monte carlo methods for solving multivariable problems. *Annals of the New York Academy of Sciences*, 86(3):844–874, 1960.

[59] Mohammad Amin Hariri-Ardebili and Bruno Sudret. Polynomial chaos expansion for uncertainty quantification of dam engineering problems. *Engineering Structures*, 203:109631, 2020.

[60] TK Hasselman. Modal coupling in lightly damped structures. *AIAA Journal*, 14(11):1627–1628, 1976.

[61] Donald Olding Hebb. *The Organization of Behavior: A Psychological Theory.* Wiley New York, 1949.

[62] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, et al. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.

[63] Tran Duong Hien and Michał Kleiber. Stochastic finite element modelling in linear transient heat transfer. *Computer Methods in Applied Mechanics and Engineering*, 144(1-2):111–124, 1997.

[64] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[65] Philip Holmes, John L Lumley, Gahl Berkooz, and Clarence W Rowley. *Turbulence, coherent structures, dynamical systems and symmetry.* Cambridge university press, 2012.

[66] Thomas Y Hou, Wuan Luo, Boris Rozovskii, and Hao-Min Zhou. Wiener chaos expansions and numerical solutions of randomly forced equations of fluid mechanics. *Journal of computational physics*, 216(2):687–706, 2006.

[67] Juntao Huang, Yingda Cheng, Andrew J Christlieb, and Luke F Roberts. Machine learning moment closure models for the radiative transfer equation i: directly learning a gradient based closure. *arXiv:2105.05690*, 2021.

[68] Juntao Huang, Yingda Cheng, Andrew J Christlieb, and Luke F Roberts. Machine learning moment closure models for the radiative transfer equation iii: enforcing hyperbolicity and physical characteristic speeds. *arXiv:2109.00700*, 2021.

[69] Juntao Huang, Yingda Cheng, Andrew J Christlieb, Luke F Roberts, and Wen-An Yong. Machine learning moment closure models for the radiative transfer equation ii: enforcing global hyperbolicity in gradient based closures. *arXiv:2105.14410*, 2021.

[70] Mubashir Hussain, Jitendra Kumar, and Evangelos Tsotsas. A new framework for population balance modeling of spray fluidized bed agglomeration. *Particuology*, 19:141–154, 2015.

[71] Bin Ji, Xianwu Luo, Xiaoxing Peng, Yulin Wu, and Hongyuan Xu. Numerical analysis of cavitation evolution and excited pressure fluctuation around a propeller in non-uniform wake. *International Journal of Multiphase Flow*, 43:13–21, 2012.

[72] Shixiao W Jiang and John Harlim. Modeling of missing dynamical systems: Deriving parametric models using a nonparametric framework. *Research in the Mathematical Sciences*, 7(3):1–25, 2020.

[73] Arun Kaintura, Tom Dhaene, and Domenico Spina. Review of polynomial chaos-based methods for uncertainty quantification in modern integrated circuits. *Electronics*, 7(3):30, 2018.

[74] Markos A Katsoulakis, Andrew J Majda, and Dionisios G Vlachos. Coarse-grained stochastic processes for microscopic lattice systems. *Proceedings of the National Academy of Sciences*, 100(3):782–787, 2003.

[75] Andrei Kazakov and Michael Frenklach. Dynamic modeling of soot particle coagulation and aggregation: Implementation with the method of moments and application to high-pressure laminar premixed flames. *Combustion and Flame*, 114(3-4):484–501, 1998.

[76] Charles D Kelman. Phaco-emulsification and aspiration: a new technique of cataract removal: a preliminary report. *American journal of ophthalmology*, 64(1):23–35, 1967.

[77] M Amin Khodkar and Pedram Hassanzadeh. Data-driven reduced modelling of turbulent rayleigh–bénard convection using dmd-enhanced fluctuation–dissipation theorem. *Journal of Fluid Mechanics*, 852, 2018.

[78] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[79] Omar M Knio, Habib N Najm, Roger G Ghanem, et al. A stochastic projection method for fluid flow: I. basic formulation. *Journal of computational Physics*, 173(2):481–511, 2001.

[80] Phoevos Koukouvinis, Christoph Bruecker, and Manolis Gavaises. Unveiling the physical mechanism behind pistol shrimp cavitation. *Scientific reports*, 7(1):1–12, 2017.

[81] Pardeep Kumar and RP Saini. Study of cavitation in hydro turbines—a review. *Renewable and Sustainable Energy Reviews*, 14(1):374–383, 2010.

[82] Sanjay Lall, Jerrold E Marsden, and Sonja Glavaški. A subspace approach to balanced truncation for model reduction of nonlinear control systems. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 12(6):519–535, 2002.

[83] Frédérique Laurent and Marc Massot. Multi-fluid modelling of laminar polydisperse spray flames: origin, assumptions and comparison of sectional and sampling methods. *Combustion theory and modelling*, 5(4):537, 2001.

[84] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[85] Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 2002.

[86] Tim Leighton, Dan Finfer, Ed Grover, and Paul White. An acoustical hypothesis for the spiral bubble nets of humpback whales and the implications for whale feeding. *Acoustics Bulletin*, 22(1):17–21, 2007.

[87] Timothy G Leighton, Simon D Richards, and Paul R White. Trapped within a wall of sound. *Acoustics bulletin*, 29:24–29, 2004.

[88] Cecil E Leith. Climate response and fluctuation dissipation. *Journal of Atmospheric Sciences*, 32(10):2022–2026, 1975.

[89] Julia Ling, Reese Jones, and Jeremy Templeton. Machine learning strategies for systems with invariance properties. *Journal of Computational Physics*, 318:22–35, 2016.

[90] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.

[91] X Liu, P-L Ma, H Wang, S Tilmes, B Singh, RC Easter, SJ Ghan, and PJ Rasch. Description and evaluation of a new four-mode version of the modal aerosol module (mam4) within version 5.3 of the community atmosphere model. *Geoscientific Model Development*, 9(2):505–522, 2016.

[92] Lennart Ljung. Black-box models from input-output measurements. In *IMTC 2001. Proceedings of the 18th IEEE instrumentation and measurement technology conference. Rediscovering measurement in the age of informatics (Cat. No. 01CH 37188)*, volume 1, pages 138–146. IEEE, 2001.

[93] Edward N Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1. Reading, 1996.

[94] JL Lumley. Similarity and the turbulent energy spectrum. *Physics of Fluids*, 10(4):855–858, 1967.

[95] John L Lumley. Coherent structures in turbulence. In *Transition and turbulence*, pages 215–242. Elsevier, 1981.

[96] Xian-wu Luo, Bin Ji, and Yoshinobu Tsujimoto. A review of cavitation in hydraulic machinery. *Journal of Hydrodynamics*, 28(3):335–358, 2016.

[97] Ming Ma, Jiacai Lu, and Gretar Tryggvason. Using statistical learning to close two-fluid multiphase flow equations for a simple bubbly system. *Physics of Fluids*, 27(9):092101, 2015.

[98] Ming Ma, Jiacai Lu, and Gretar Tryggvason. Using statistical learning to close two-fluid multiphase flow equations for bubbly flows in vertical channels. *International Journal of Multiphase Flow*, 85:336–347, 2016.

[99] Zhanhua Ma, Clarence W Rowley, and Gilead Tadmor. Snapshot-based balanced truncation for linear time-periodic systems. *IEEE Transactions on Automatic Control*, 55(2):469–473, 2010.

[100] AJ Majda, DW McLaughlin, and EG1431687 Tabak. A one-dimensional model for dispersive wave turbulence. *Journal of Nonlinear Science*, 7(1):9–44, 1997.

[101] Andrew Majda and Xiaoming Wang. *Nonlinear dynamics and statistical theories for basic geophysical flows.* Cambridge University Press, 2006.

[102] Andrew J Majda. Statistical energy conservation principle for inhomogeneous turbulent dynamical systems. *Proceedings of the National Academy of Sciences*, 112(29):8937–8941, 2015.

[103] Andrew J Majda and Boris Gershgorin. Quantifying uncertainty in climate change science through empirical information theory. *Proceedings of the National Academy of Sciences*, 107(34):14958–14963, 2010.

[104] Andrew J Majda and Di Qi. Strategies for reduced-order models for predicting the statistical responses and uncertainty quantification in complex turbulent dynamical systems. *SIAM Review*, 60(3):491–549, 2018.

[105] Andrew J Majda and Di Qi. Linear and nonlinear statistical response theories with prototype applications to sensitivity analysis and statistical control of complex turbulent dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(10):103131, 2019.

[106] Andrew J Majda, Ilya Timofeyev, and Eric Vanden Eijnden. Models for stochastic climate prediction. *Proceedings of the National Academy of Sciences*, 96(26):14687–14691, 1999.

[107] Daniele L Marchisio and Rodney O Fox. *Computational models for polydisperse particulate and multiphase systems.* Cambridge University Press, 2013.

[108] Romit Maulik, Arvind Mohan, Bethany Lusch, Sandeep Madireddy, Prasanna Balaprakash, and Daniel Livescu. Time-series learning of latent-space dynamics for reduced-order model closure. *Physica D: Nonlinear Phenomena*, 405:132368, 2020.

[109] Romit Maulik, Omer San, Adil Rasheed, and Prakash Vedula. Subgrid modelling for two-dimensional turbulence using neural networks. *Journal of Fluid Mechanics*, 858:122–144, 2019.

[110] Andrea Mazzino, Paolo Muratore-Ginanneschi, and Stefano Musacchio. Scaling properties of the two-dimensional randomly stirred navier-stokes equation. *Physical review letters*, 99(14):144502, 2007.

[111] Robert McGraw. Description of aerosol dynamics by the quadrature method of moments. *Aerosol Science and Technology*, 27(2):255–265, 1997.

[112] Ralph Menikoff and Bradley J Plohr. The Riemann problem for fluid flow of real materials. *Reviews of Modern Physics*, 61(1):75, 1989.

[113] Nicholas Metropolis and Stanislaw Ulam. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.

[114] Michele Milano and Petros Koumoutsakos. Neural network modeling for near wall turbulent flow. *Journal of Computational Physics*, 182(1):1–26, 2002.

[115] Eli J Mlawer, Steven J Taubman, Patrick D Brown, Michael J Iacono, and Shepard A Clough. Radiative transfer for inhomogeneous atmospheres: Rrtm, a validated correlated-k model for the longwave. *Journal of Geophysical Research: Atmospheres*, 102(D14):16663–16682, 1997.

[116] Parviz Moin, Kyle Squires, W Cabot, and Sangsan Lee. A dynamic subgrid-scale model for compressible turbulence and scalar transport. *Physics of Fluids A: Fluid Dynamics*, 3(11):2746–2757, 1991.

[117] Hugh Morrison and Andrew Gettelman. A new two-moment bulk stratiform cloud microphysics scheme in the community atmosphere model, version 3 (cam3). part i: Description and numerical tests. *Journal of Climate*, 21(15):3642–3659, 2008.

[118] Michael Edward Mueller, Guillaume Blanquart, and Heinz Pitsch. A joint volume-surface model of soot aggregation with the method of moments. *Proceedings of the Combustion Institute*, 32(1):785–792, 2009.

[119] Habib N Najm. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annual review of fluid mechanics*, 41:35–52, 2009.

[120] Guido Novati, Hugues Lascombes de Laroussilhe, and Petros Koumoutsakos. Automating turbulence modelling by multi-agent reinforcement learning. *Nature Machine Intelligence*, 3(1):87–96, 2021.

[121] Ryota Oguri and Keita Ando. Cavitation bubble nucleation induced by shock-bubble interaction in a gelatin gel. *Physics of Fluids*, 30(5):051904, 2018.

[122] KW Oleson, DM Lawrence, GB Bonan, B Drewniack, M Huang, CD Koven, S Levis, F Li, WJ Riley, ZM Subin, et al. Technical description of version 4.5 of the community land model (clm)(technical note no. ncar/tn-503+ str). boulder, co: National center for atmospheric research earth system laboratory, 2013.

[123] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[124] Shucheng Pan, Stefan Adami, Xiangyu Hu, and Nikolaus A Adams. Phenomenology of bubble-collapse-driven penetration of biomaterial-surrogate liquid-liquid interfaces. *Physical Review Fluids*, 3(11):114005, 2018.

[125] Alberto Passalacqua, Frédérique Laurent, Ehsan Madadi-Kandjani, Jeffrey C Heylmun, and Rodney O Fox. An open-source quadrature-based population balance solver for openfoam. *Chemical Engineering Science*, 176:306–318, 2018.

[126] Sheila N Patek, WL Korff, and Roy L Caldwell. Deadly strike mechanism of a mantis shrimp. *Nature*, 428(6985):819–820, 2004.

[127] Ravi G Patel, Olivier Desjardins, and Rodney O Fox. Three-dimensional conditional hyperbolic quadrature method of moments. *Journal of Computational Physics: X*, 1:100006, 2019.

[128] Julien Pedel, Jeremy N Thornock, Sean T Smith, and Philip J Smith. Large eddy simulation of polydisperse particles in turbulent coaxial jets using the direct quadrature method of moments. *International Journal of Multiphase Flow*, 63:23–38, 2014.

[129] Brian D Phenix, Joanna L Dinaro, Menner A Tatang, Jefferson W Tester, Jack B Howard, and Gregory J McRae. Incorporation of parametric uncertainty into complex kinetic mechanisms: application to hydrogen oxidation in supercritical water. *Combustion and Flame*, 112(1-2):132–146, 1998.

[130] Norman A Phillips. Energy transformations and meridional circulations associated with simple baroclinic waves in a two-level, quasi-geostrophic model. *Tellus*, 6(3):274–286, 1954.

[131] Ethan Pickering and Themistoklis P Sapsis. Information fomo: The unhealthy fear of missing out on information. a method for removing misleading data for healthier models. *arXiv preprint arXiv:2208.13080*, 2022.

[132] Yuriy A Pishchalnikov, Oleg A Sapozhnikov, Michael R Bailey, James C Williams Jr, Robin O Cleveland, Tim Colonius, Lawrence A Crum, Andrew P Evan, and James A McAteer. Cavitation bubble cluster activity in the breakage of kidney stones by lithotripter shockwaves. *Journal of Endourology*, 17(7):435–446, 2003.

[133] Fernando Porté-Agel, Charles Meneveau, and Marc B Parlange. A scale-dependent dynamic model for large-eddy simulation: application to a neutral atmospheric boundary layer. *Journal of Fluid Mechanics*, 415:261–284, 2000.

[134] Di Qi and Andrew J Majda. Predicting extreme events for passive scalar turbulence in two-layer baroclinic flows through reduced-order stochastic models. *Communications in Mathematical Sciences*, 16(1):17–51, 2018.

[135] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[136] Maziar Raissi, Zhicheng Wang, Michael S Triantafyllou, and George Em Karniadakis. Deep learning of vortex-induced vibrations. *Journal of Fluid Mechanics*, 861:119–137, 2019.

[137] Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030, 2020.

[138] Doraiswami Ramkrishna. *Population balances: Theory and applications to particulate systems in engineering.* Elsevier, 2000.

[139] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[140] Daniel E Rosner, Robert McGraw, and Pushkar Tandon. Multivariate population balances via moment and Monte Carlo simulation methods: an important sol reaction engineering bivariate example and "mixed" moments for the estimation of deposition, scavenging, and optical properties for populations of nonspherical suspended particles. *Industrial & Engineering Chemistry Research*, 42(12):2699–2711, 2003.

[141] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[142] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[143] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[144] Pierre Sagaut and Yu-Tai Lee. Large eddy simulation for incompressible flows: An introduction. scientific computation series. *Appl. Mech. Rev.*, 55(6):B115–B116, 2002.

[145] Rick Salmon. *Lectures on geophysical fluid dynamics.* Oxford University Press, 1998.

[146] Mehdi Samiee, Ali Akhavan-Safaei, and Mohsen Zayernouri. A fractional subgrid-scale model for turbulent flows: Theoretical formulation and a priori study. *Physics of Fluids*, 32(5):055102, 2020.

[147] Themistoklis P Sapsis. Attractor local dimensionality, nonlinear energy transfers and finite-time instabilities in unstable dynamical systems with applications to two-dimensional fluid flows. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 469(2153):20120550, 2013.

[148] Themistoklis P Sapsis and Pierre FJ Lermusiaux. Dynamically orthogonal field equations for continuous stochastic dynamical systems. *Physica D: Nonlinear Phenomena*, 238(23-24):2347–2360, 2009.

[149] Themistoklis P Sapsis and Andrew J Majda. Blending modified gaussian closure and non-gaussian reduced subspace methods for turbulent dynamical systems. *Journal of Nonlinear Science*, 23(6):1039–1071, 2013.

[150] Themistoklis P Sapsis and Andrew J Majda. Statistically accurate low-order models for uncertainty quantification in turbulent dynamical systems. *Proceedings of the National Academy of Sciences*, 110(34):13705–13710, 2013.

[151] Christoph Schwab and Jakob Zech. Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in uq. *Analysis and Applications*, 17(01):19–55, 2019.

[152] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[153] SD Sharma, K Mani, and VH Arakeri. Cavitation noise studies on marine propellers. *Journal of Sound and Vibration*, 138(2):255–283, 1990.

[154] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[155] Alaric Sibra, Joel Dupays, Angelo Murrone, Frédérique Laurent, and Marc Massot. Simulation of reactive polydisperse sprays strongly coupled to unsteady flows in solid rocket motors: Efficient strategy using eulerian multi-fluid methods. *Journal of Computational Physics*, 339:210–246, 2017.

[156] Anand Pratap Singh, Shivaji Medida, and Karthik Duraisamy. Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. *AIAA Journal*, pages 2215–2227, 2017.

[157] Lawrence Sirovich. Turbulence and the dynamics of coherent structures. i. coherent structures. *Quarterly of applied mathematics*, 45(3):561–571, 1987.

[158] Lawrence Sirovich. Turbulence and the dynamics of coherent structures. i. coherent structures. *Quarterly of applied mathematics*, 45(3):561–571, 1987.

[159] Lawrence Sirovich. Turbulence and the dynamics of coherent structures. iii. dynamics and scaling. *Quarterly of Applied mathematics*, 45(3):583–590, 1987.

[160] Lawrence Sirovich. Turbulence and the dynamics of coherent structures. iii. dynamics and scaling. *Quarterly of Applied mathematics*, 45(3):583–590, 1987.

[161] Jonas Sjöberg, Qinghua Zhang, Lennart Ljung, Albert Benveniste, Bernard Delyon, Pierre-Yves Glorennec, Håkan Hjalmarsson, and Anatoli Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12):1691–1724, 1995.

[162] Jean-Sebastien Spratt, Mauro Rodriguez, Kevin Schmidmayer, Spencer H Bryngelson, Jin Yang, Christian Franck, and Tim Colonius. Characterizing viscoelastic materials via ensemble-based data assimilation of bubble collapse observations. *Journal of the Mechanics and Physics of Solids*, 152:104455, 2021.

[163] Karen Stengel, Andrew Glaws, Dylan Hettinger, and Ryan N King. Adversarial super-resolution of climatological wind and solar data. *Proceedings of the National Academy of Sciences*, 117(29):16805–16815, 2020.

[164] Gilbert Strang. *Introduction to linear algebra*, volume 3. Wellesley-Cambridge Press Wellesley, MA, 1993.

[165] Victor L Streeter. Transient cavitating pipe flow. *Journal of Hydraulic Engineering*, 109(11):1407–1423, 1983.

[166] Jian Sun, Kai Zhang, Hui Wan, Po-Lun Ma, Qi Tang, and Shixuan Zhang. Impact of nudging strategy on the climate representativeness and hindcast skill of constrained eamv1 simulations. *Journal of Advances in Modeling Earth Systems*, 11(12):3911–3933, 2019.

[167] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013.

[168] Floris Takens. Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980*, pages 366–381. Springer, 1981.

[169] Mark A Taylor, A St Cyr, and Aimé Fournier. A non-oscillatory advection operator for the compatible spectral element method. In *International Conference on Computational Science*, pages 273–282. Springer, 2009.

[170] AAR Townsend. *The structure of turbulent shear flow.* Cambridge university press, 1980.

[171] CK Turangan, GJ Ball, AR Jamaluddin, and TG Leighton. Numerical studies of cavitation erosion on an elastic–plastic material caused by shock-induced bubble collapse. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2205):20170315, 2017.

[172] Paul A Ullrich and Colin M Zarzycki. Tempestextremes: A framework for scale-insensitive pointwise feature tracking on unstructured grids. *Geoscientific Model Development*, 10(3):1069–1090, 2017.

[173] Paul A Ullrich, Colin M Zarzycki, Elizabeth E McClenny, Marielle C Pinheiro, Alyssa M Stansfield, and Kevin A Reed. Tempestextremes v2. 1: a community framework for feature detection, tracking, and analysis in large datasets. *Geoscientific Model Development*, 14(8):5023–5048, 2021.

[174] Shahram Vaezy, Roy Martin, Pierre Mourad, and Lawrence Crum. Hemostasis using high intensity focused ultrasound. *European journal of ultrasound*, 9(1):79–87, 1999.

[175] Shahram Vaezy, Roy Martin, Udo Schmiedl, Michael Caps, Shari Taylor, Kirk Beach, Steve Carter, Peter Kaczkowski, George Keilman, Scott Helton, et al. Liver hemostasis using high-intensity focused ultrasound. *Ultrasound in medicine & biology*, 23(9):1413–1420, 1997.

[176] Geoffrey K Vallis. *Atmospheric and oceanic fluid dynamics.* Cambridge University Press, 2017.

[177] Arun Venkatraman, Martial Hebert, and J Andrew Bagnell. Improving multi-step prediction of learned time series models. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[178] Pantelis R Vlachas, Wonmin Byeon, Zhong Y Wan, Themistoklis P Sapsis, and Petros Koumoutsakos. Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2213):20170844, 2018.

[179] Pantelis R Vlachas, Jaideep Pathak, Brian R Hunt, Themistoklis P Sapsis, Michelle Girvan, Edward Ott, and Petros Koumoutsakos. Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks*, 126:191–217, 2020.

[180] Zhong Yi Wan, Pantelis Vlachas, Petros Koumoutsakos, and Themistoklis Sapsis. Data-assisted reduced-order modeling of extreme events in complex dynamical systems. *PloS one*, 13(5):e0197704, 2018.

[181] Michael E Weyler, VL Streeter, and PS Larsen. An investigation of the effect of cavitation bubbles on the momentum loss in transient pipe flow. *Journal of Basic Engineering*, 93(1):1–7, 1971.

[182] Hassler Whitney. Differentiable manifolds. *Annals of Mathematics*, pages 645–680, 1936.

[183] David H Wolpert. The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390, 1996.

[184] Jin-Long Wu, Heng Xiao, and Eric Paterson. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Physical Review Fluids*, 3(7):074602, 2018.

[185] Dongbin Xiu and George Em Karniadakis. A new stochastic approach to transient heat conduction modeling with uncertainty. *International Journal of Heat and Mass Transfer*, 46(24):4681–4693, 2003.

[186] XIA Yang, Suhaib Zafar, J-X Wang, and Heng Xiao. Predictive large-eddy-simulation wall modeling via physics-informed neural networks. *Physical Review Fluids*, 4(3):034602, 2019.

[187] Cansheng Yuan and Rodney O Fox. Conditional quadrature method of moments for kinetic equations. *Journal of Computational Physics*, 230(22):8216–8246, 2011.

[188] DZ Zhang and A Prosperetti. Ensemble phase-averaged equations for bubbly flows. *Physics of Fluids*, 6(9):2956–2970, 1994.

[189] Guang J Zhang and Norman A McFarlane. Sensitivity of climate simulations to the parameterization of cumulus convection in the canadian climate centre general circulation model. *Atmosphere-ocean*, 33(3):407–446, 1995.

[190] He Zhang, John Harlim, and Xiantao Li. Computing linear response statistics using orthogonal polynomial based estimators: An rkhs formulation. *arXiv preprint arXiv:1912.11110*, 2019.

[191] Shixuan Zhang, Kai Zhang, Hui Wan, and Jian Sun. Further improvement and evaluation of nudging in the e3sm atmosphere model version 1 (eamv1). *Geoscientific Model Development Discussions*, pages 1–37, 2022.

[192] H Zhao, A Maisels, T Matsoukas, and C Zheng. Analysis of four Monte Carlo methods for the solution of population balances in dispersed systems. *Powder Technology*, 173(1):38–50, 2007.

[193] Xu-Hui Zhou, Jiequn Han, and Heng Xiao. Frame-independent vector-cloud neural network for nonlocal constitutive modeling on arbitrary grids. *Computer Methods in Applied Mechanics and Engineering*, 388:114211, 2022.

[194] Jiawei Zhuang, Dmitrii Kochkov, Yohai Bar-Sinai, Michael P Brenner, and Stephan Hoyer. Learned discretizations for passive scalar advection in a two-dimensional turbulent flow. *Physical Review Fluids*, 6(6):064605, 2021.