OVERCOMING SMALL DATASETS IN MACHINE LEARNING STUDIES OF MULTI-PHYSICS FLOWS IN PROPULSION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Wai Tong Chung

June 2024

This dissertation is online at: https://purl.stanford.edu/xw487vj6602

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Matthias Ihme, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Gianluca Iaccarino

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Hai Wang

Approved for the Stanford University Committee on Graduate Studies.

Stacey F. Bent, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format.

Abstract

While machine learning (ML) methods can offer numerous opportunities in modeling multi-physics flows, these approaches often rely on the availability of large datasets for generating reliable predictions. This can be challenging for propulsion applications, especially since data generated by industrial sensors, experiments, and numerical simulations of flow phenomena in propulsion systems can be challenging to collect.

In this dissertation, we directly address current gaps in data availability by developing a 2.2 TB ML dataset from 34 high-fidelity direct numerical simulations (DNS) of turbulent flows. We employ this data for benchmarking super-resolution of turbulent flows, and provide insights into the role of different deep learning designs and computational scale in a popular ML application within multi-physics flows.

To address issues in accessing data from relatively under-explored flow configurations (*e.g.*, real, hypersonic, and multiphase fluids) in propulsion systems, we investigate opportunities offered by linear regression and random forest models in modeling subgrid-scale (SGS) closure on small turbulent transcritical DNS dataset. Through *a priori* analysis, interpretable metrics from random forest models, along with weights within linear regressors, are shown to assist in discovering analytical expressions for modeling SGS stresses and a closure term that arises from a real-fluid equation-of-state.

To ameliorate spurious errors that can arise when integrating insufficiently trained ML models within multi-physics flow solvers, we develop a strategy involving an ML-based classifier that assigns three different combustion models of varying fidelity and cost within a shared simulation domain. Results from *a posteriori* simulations show that this data-assisted framework demonstrates promise as a tool for controlling the

fidelity-cost trade-off in numerical multi-physics flow simulations.

Finally, we investigate the benefits of combining domain knowledge with ML by integrating a deep learning model with a stochastic differential equation for predicting the spatio-temporal behavior of laser ignition kernels with sparse ensemble data of a model rocket combustor. Results show that this hybrid reduced-order model can predict dominant ignition modes observed from corresponding experimental measurements, and generate spatially resolved ignition probability, at lower costs than high-fidelity turbulent reacting flow simulations approaches.

Overall, the efforts within this dissertation contribute towards overcoming data limitations in ML-based modeling within science and engineering, specifically in the context of multi-physics flows found in propulsion systems.

Acknowledgments

I am grateful to Prof. Matthias Ihme for years of guidance and trust in these research efforts. His energy and excitement for new ideas, along with his expertise in predictive modeling, turbulent reacting flows, and propulsion, have been essential to a productive and positive PhD experience at Stanford. His support has been especially important, due to the overlap of my PhD with the COVID-19 pandemic.

I would also like to thank Prof. Gianluca Iaccarino, Prof. Hai Wang, Prof. Eric Darve, and Prof. Juan Alonso for being part of my defense committee. During my early years at Stanford, I had formative learning experiences at the lectures on linear algebra, combustion, and high-performance computing from Profs. Iaccarino, Wang, and Darve, respectively, which has been formative for performing this research. In addition, I have learned greatly through the perspective gained from interacting with their laboratories via the PSAAP-III program.

I am grateful to my collaborators Peter Ma, Aashwin Mishra, Nikolaos Perakis, Ki Sung Jung, Jacqueline H. Chen, David Wu, Jack Guo, Davy Brouzet, Prof. Mohsen Talei, Nguyen Ly, Giselle Fernández-Godino, Donald Lucas, Pushan Sharma, Bassem Akoush, Alex Tamkin, Prof. Bruno Savard, Prof. Alexei Poludnenko, Donatella Passiatore, Charlélie Laurent, Jen Zen Ho, Jordan Kildare, Prof. Michael Evans, Prof. Paul Medwell, Walter Reade, and Chris Cundy for fun collaborations in different research projects.

I would like to thank past and present members of the Fx Lab who have contributed to a wonderful work culture and environment where I can thrive. I am especially grateful to Danyal Mohaddes for acting as a mentor and friend within the lab. Outside the Fx Lab, I am grateful to the friends I've made at Stanford including

Nick Kateris, Elise Loppinet, Travis McGuire, Omkar Shende, and Carlos Gonzalez for their companionship and ability in tolerating my sense of humor. Outside of Stanford, I thank Prof. Salvador Navarro-Martinez and Prof. Alex Taylor at Imperial College London for starting my journey in research, scientific computing, turbulent reacting flows, and propulsion systems.

From my home in Malaysia, I am most thankful to my parents for their sacrifices in supporting me for longer than I can remember. Similarly, I am also grateful to have received care and support from my two older siblings, Wai Hoe and Wei Ling. I appreciate my partner Allison Cong, for choosing to be part of my life, and her family in San Jose who have made me feel at home during the holidays and weekend get-togethers while in California.

Lastly, I am thankful for the financial support from the U.S. Department of Energy, National Nuclear Security Administration, the Predictive Science Academic Alliance Program, the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, Stanford Institute for Human-Centered AI Graduate Fellowship, Air Force Office of Scientific Research, National Aeronautics and Space Administration, Stanford Engineering Fellowship, and German Research Foundation (Deutsche Forschungsgemeinschaft – DFG) in making my research possible.

Nomenclature

Acronyms

CaRT classification and regression tree

CNN convolutional neural network

DA data-assisted

DNS direct numerical simulation

EDSR enhanced deep residual super-resolution

ENO essentially non-oscillatory

EoS equation-of-state

FLOP floating point operation

FNO Fourier neural operator

FPV flamelet/progress variable

FRC finite-rate chemistry

HIT homogeneous isotropic turbulence

IM inert mixing

LES large-eddy simulation

MDI mean decrease in impurity

MIC maximal information coefficient

MINE maximal information-based non-parametric exploration

ML machine learning

MSE mean-squared error

NN neural network

NRMSE normalized root mean-squared error

OOD out-of-distribution

PDF probability density function

PEC Pareto-efficient combustion

PR Peng-Robinson

QoI quantity-of-interest

RANS Reynolds-averaged Navier-Stokes

RCAN residual channel attention network

ReLU rectified linear unit

RMS root mean-squared

RRDB residual-in-residual dense block

SDE stochastic differential equation

SGS subgrid-scale

SR super-resolution

SSIM structural similarity index measure

TKE turbulent kinetic energy

Greek Characters

χ	data input/feature
Δt	numerical timestep
Δ	numerical grid cell spacing
δ_f	flame thickness
δ_{ij}	identity matrix component
ϵ	error
ϵ_{gen}	generalization error
η_k	Kolmogorov length-scale
Γ	moving average of squared neural network gradients
γ	moving average of neural network gradients
Γ^*	bias-corrected moving average of squared neural network gradients
γ^*	bias-corrected moving average of neural network gradients
κ	wavenumber
κ_η	Kolmogorov wavenumber
Λ	tunable weighting factor
λ	thermal conductivity
μ	dynamic viscosity
ν^{sgs}	eddy viscosity
$\overline{\Delta}$	filter width
$\overline{\Delta}$	large-eddy simulation filter width

ϕ arbitrary quantity

Φ_k transported chemical quantity

ψ arbitrary quantity

ρ density

σ non-linear transformation function

σ_n non-linear/activation function at the n-th neural network layer

τ duration

τ_{ig} ignition duration

τ_{ij} viscous stress tensor component

τ_{ij}^{sgs} subgrid-scale stress tensor component

Θ machine learning model parameter/weight

θ labeling threshold

Υ data target/label

ε turbulent dissipation rate

ζ compressibility factor

f_n arbitrary function applied within the n-th layer of a neural network

$\cdot^{sgs}$ subgrid-scale quantity

$\dot{\omega}_k$ source term of chemical scalar k

Latin Characters

$\mathbf{E}_{\boldsymbol{\Upsilon}}$ expectation across labels

$\mathbf{E}_{\mathcal{D}}$ expectation across all possible training datasets

$\dot{m}$	mass flow rate
κ_I	large-eddy wavenumber
$\mathcal{C}$	model coefficient
$\mathcal{F}$	filter
$\mathcal{K}$	manifold model
$\mathcal{M}$	manifold reconstruction function
$\mathcal{V}_n$	the n-th node/vertex within a computational graph
a	Peng-Robinson intermolecular force coefficient
b	Peng-Robinson volume displacement coefficient
C	progress variable
d	diameter
D_k	molecular diffusivity of species k
dW	Wiener process
E	energy
e^i	specific internal energy
e^k	specific kinetic energy
e^t	specific total energy
f	arbitrary function
f_{lin}	linear regression hypothesis for modeling an arbitrary function
f_{ML}	machine learning hypothesis for modeling an arbitrary function
f_{NN}	neural network hypothesis for modeling an arbitrary function

f_{sym} symbolic regression hypothesis for modeling an arbitrary function

h_k^s partial sensible enthalpy of species k

h_n output from the n-th layer of a neural network

h_n output of the n-th neural network layer

J_{CE} cross-entropy loss

j_{kj} diffusion flux component of the k-th chemical scalar

J_{MSE} mean-squared error loss

J_{task} machine learning loss function for a single sample for a given task

J_{tot} machine learning loss function across a set of samples

L domain length

l_I integral length-scale

l_{char} characteristic length-scale

M_k molar mass of species k

N_x number of grid cells in the axial direction

N_c number of convolutional layer channels

N_{exp} number of experiments

N_{feat} number of features/inputs

N_p number of model parameters

N_{samp} Number of samples

N_{vox} number of voxels

N_w number of moving windows

p	pressure
P_{ig}	ignition probability
Q	quantity-of-interest
q_i	heat flux component
R	specific gas constant
r	radial coordinate
R_u	universal gas constant
R_{ij}	spin tensor component
S_L	laminar flame speed
S_{ij}	strain tensor component
T	temperature
t	temporal coordinate
t_I	eddy turnover time
t_{chem}	chemical time-scale
t_{conv}	mixing time-scale
T_{LES}	temperature resolved in large-eddy simulation
u'	fluctuating velocity
u_i	velocity component
V	volume
x	axial coordinate
x_i	spatial dimension component

y	longitudinal coordinate
Y_k	mass fraction of species k
Z	mixture fraction
z	transverse coordinate
Da	Damköhler number
Ka	Karlovitz number
Re	Reynolds number

Other Characters

$\overline{\cdot}$	filtered quantity
$\widetilde{\cdot}$	Favre-filtered quantity
$\langle \cdot \rangle$	volume-averaged quantity
$\cdot^{sgs}$	subgrid-scale quantity
$\hat{\cdot}$	machine learning prediction

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Predictive modeling of multi-physics flows found in propulsion systems can be computationally challenging [1, 2]. In these flows, multiscale turbulent mixing and chemical reactions can impose significant computational restrictions that arise from grid resolution requirements, stiff differential equations, and large dimensionality [3, 4]. In recent years, *machine learning* (ML) has offered cost-effective and promising modeling approaches that can help address these challenges [5, 6]. These techniques have been shown to be useful for model discovery [7], model optimization [8], computational acceleration [9], reduced-order modeling [10], data analysis [11], fault detection [12], and control [13]. By improving our ability to model and understand multi-physics flow phenomena, these data-driven techniques offer opportunities for improving the efficiency, performance, and robustness of next-generation propulsion systems [6].

Within ML, *deep learning* has emerged as the most popular family of techniques due to its (i) proliferation via publicly available software packages [14, 15] (which are compatible with parallel computing systems for training), (ii) flexibility in managing computational trade-offs [16], and (iii) ability to scale its predictive accuracy with increasing diversity and volume of data [17]. In mature ML fields such as *computer vision* or *natural language processing*, large training datasets associated with deep learning can be conveniently extracted from the World Wide Web [18, 19]. In contrast,

scientific and engineering datasets such as within propulsion and multi-physics flows require significantly more effort to develop. Databases of thermochemical properties and flow physics behavior can require laborious experimental measurements [20] and costly scientific computation [4]. While sensor data can be applied to train ML models for control and diagnostics applications within propulsion systems, this data is often proprietary and reserved for commercial interests [6].

This dissertation considers several strategies towards overcoming limitations introduced by current gaps in suitable datasets for employing ML techniques within propulsion and multi-physics. Firstly, we develop an affordable open-source framework for curating large datasets involving multi-physics flows, in order to directly address current gaps in data availability [21]. With this data, we demonstrate the effectiveness of deep learning models when learning a flow physics task, in the presence of big data. For problems where data cannot be easily accessed, we also develop strategies for improving the application of ML techniques in these conditions, by utilizing model interpretability [7], treating out-of-distribution errors [9], and combining data-driven and physics-based approaches [10] in different multi-physics flow problems.

1.2 Background

This dissertation considers *supervised learning* models [22] that are trained on flow physics data. Supervised learning models rely on datasets that have been preprocessed into inputs/*features* and target outputs/*labels*. Linear regression [23], classification and regression trees (CaRTs) [24], and neural networks (NNs) [25] are examples of supervised learning methods that are typically employed in *classification* and *regression* tasks. In flow physics and propulsion studies, classification applications can involve choosing and blending from a set of predefined models within a simulation domain [9, 26], as well as diagnosing faults and detecting anomalous events [27]. Regression applications include closure modeling [21], predicting spatio-temporal dynamics [28], inverse modeling [29], and discovering analytic models [7].

In these applications, NN-based approaches can offer the ability to tune various

computational trade-offs through the manipulation of neural architectures consisting of stacked NN layers [30]. When stacking beyond approximately ten layers [31], this NN can be referred to as a deep learning model, which has been shown to enables automatic processing of unstructured spatial data without significant data preprocessing [30]. Numerous deep learning architectures with different properties have been formed through this flexible framework [21]. For example, model architectures such as MeshGraphNet [32] and Fourier neural operators (FNO) [33] employ graph and spectral convolution layers to ensure that flow predictions are mesh invariant.

While deep learning architectures can outperform their shallow counterparts in terms of predictive accuracy, these methods can require significant amounts of training data to achieve good performance [16]. Thus, the availability of large multi-physics flow datasets is crucial for proliferating deep learning approaches that can tackle propulsion-related problems [21]. In relation to this, high-fidelity numerical simulations have been essential for providing detailed insights into multi-physics flow phenomena within propulsion, and can act as a reliable source of ML data. Direct numerical simulations (DNS) accurately describe flow physics, as long as the grid resolves the smallest length-scales associated with turbulent dissipation [34]. With up to $\mathcal{O}(10^9)$ voxels, $\mathcal{O}(10^6)$ core-hours of simulation time, and $\mathcal{O}(10^4)$ cores on parallel computing facilities [35–38], high-fidelity DNS of many real-world flows cannot be performed due to prohibitive costs. Thus, it is common to employ coarser grids with large-eddy simulations (LES) [39, 40], or by only evolving time-/ensemble-averaged quantities with Reynolds-averaged Navier-Stokes (RANS) simulations [41, 42] – both of which rely on closure models (for the consideration of under-resolved phenomena) that can be discovered from DNS data.

Many existing flow simulation datasets focus on LES and RANS simulations. McConkey et al. [41] released a dataset for improving turbulence models in incompressible non-reacting RANS. AirfRANS [42] provides both 2D incompressible and compressible non-reacting RANS data, specifically on airfoil configurations. For reacting flows, Huang [40] released a 2D LES dataset for developing reduced-order models. The largest flow physics dataset, the Johns Hopkins Turbulence Database [43], provides 3D DNS data from turbulent incompressible non-reacting flow simulations.

Since these datasets are either 2D, incompressible, or non-reacting, they are not suitable for propulsion applications involving compressible, reacting, and turbulent phenomena. This is one reason why ML studies involving these applications employ self-generated private datasets [6, 44] – introducing challenges to transparent and reproducible model evaluation.

DNS data provides opportunities for developing closure models for LES and RANS, which have typically relied on analytic models [45–52]. One method for evaluating closure models involves *a priori* analysis, where modeled subgrid-scale (SGS) terms are compared with exact unclosed terms extracted from filtered DNS. For example, Ihme et al. [53] performed *a priori* analysis of turbulent reacting DNS data to identify models that can accurately capture the unclosed chemical source term in the flamelet/progress variable (FPV) [54] transport equations. Selle et al. [55] performed *a priori* analysis on a three-dimensional DNS database of supercritical binary mixtures in turbulent mixing layers to demonstrate that the Smagorinsky model [45] performed poorly when predicting SGS stresses, while the gradient [46] and scale-similar [47] models performed well. In the same work, the consideration of previously neglected unclosed terms for pressure and heat flux were shown to be essential under supercritical conditions. Unnikrishnan et al. [56] performed *a priori* analysis on two-dimensional DNS of a transcritical reacting liquid-oxygen/gaseous-methane (LOX/GCH_4) mixture to demonstrate that the mixed SGS model [51] was three times more accurate than the sole use of the dynamic Smagorinsky [48].

ML methods offer an alternative approach for developing closure models in multi-physics flows [7]. *A priori* studies have been performed to demonstrate that NNs can provide accurate closure for turbulent combustion [57–59]. Henry de Frahan et al. [60] demonstrated that NN models can generate as accurate results as a CaRT approach with 25-fold improvement in computational costs when predicting the sub-filter probability density function (PDF). Ranade and Echekki [61] conducted an *a posteriori* study (where the closure model is integrated with a multi-physics flow solver) to show that NN models can be trained with experimental data to generate closure models for chemical scalars in RANS simulations of turbulent jet flames. In many of these efforts, the developed ML models suffer issues related to model

generalizability, especially when predicting with out-of-distribution (OOD) inputs.

Another popular ML application within multi-physics reacting flow solvers involves the reduction of computational costs that arise from complex combustion chemistry [6]. Prior to the popularity of ML techniques, numerous strategies have been employed for reducing the computational cost of detailed chemical mechanisms via tabulation approaches. Some of the most popular of these *tabulated chemistry* models are categorized under flamelet methods, which represent combustion chemistry through solutions of representative flame configurations, such as laminar counterflow diffusion flames, freely propagating premixed flames, or homogeneous reactor systems. Examples of flamelet methods include the Burke-Schumann solution [62], the flame-prolongation in intrinsic lower-dimensional manifold [63], the flamelet-generated manifold method [64], and the FPV method [53, 54]. These reduced manifold models are commonly employed to describe specific combustion regimes – a multitude of which can exist within practical combustors. However, expert knowledge and experimental data is often required to correctly assign the most appropriate combustion model. One solution to this issue is provided by dynamic adaptive chemistry methods [65–67] that save computational cost by reducing detailed chemical mechanisms, and transitioning between smaller sets of chemical models to represent combustion regimes of different chemical fidelity. A general mathematical framework was proposed by Wu et al. [68, 69] through the Pareto-efficient combustion (PEC) approach. In this approach, the compliance of a combustion submodel with the underlying flowfield representation is assessed through the construction of a so-called drift term, taking into consideration user-specific requirements about quantities-of-interest (QoIs) and computational cost [70].

ML methods can be employed towards developing accurate and affordable combustion models for these multi-physics flow solvers. For example, NNs has been successfully integrated within simulations of turbulent reacting flows as non-linear approximators for representing chemical reactions [71–73]. Sen and Menon [74], as well as Alqahtani and Echekki [75], also demonstrated that NNs can be used for replacing stiff ODE solvers in turbulent flame simulations, with good accuracy and CPU performance. Ihme et al. [59], Kempf et al. [76], and Owoyele et al. [77] used

optimal NN tabulation to replace conventional tabulation methods in manifold-based simulations. Chatzopoulos and Rigopoulos [78], and Franke et al. [79] demonstrated that training data extracted from 100 laminar flamelets was sufficient for training NNs for representing chemistry in simulations of more complex turbulent flame configurations. With this generic training set, NNs showed a small capacity for extrapolation, but it was noted that accurate predictions were challenging if the target predictions deviated too largely away from the training set.

ML methods can offer benefits in cost and accuracy when examining flow physics phenomena that require significant efforts and resources, such as with the statistical characterization of ignition [10]. During ignition, stochasticity arises from the interaction of the ignition kernel with the turbulent fuel/oxidizer mixture, and from variations in kernel deposition energy, which can affect the evolution of the ignition kernel [80].

Ensembles of experimental measurements are typically used to investigate stochastic ignition under a variety of conditions. For example, Ahmed et al. [81] performed ensemble tests on a CH_4/air counterflow flame, and found that convective and local strain effects were significant in influencing ignition probability in this fundamental configuration. Several studies [82–84] have focused on the effects of fuel chemistry and mixture composition in determining ignition probability in experimental configurations that represent gas turbine combustors. In another work, Cordier et al. [85] employed a premixed CH_4/air confined swirl combustor configuration to show the importance of turbulent flow properties in influencing kernel trajectory and ignition probability. Strelau et al. [86] investigated laser ignition in a non-premixed gaseous methane/oxygen (CH_4/O_2) model rocket combustor and found three distinct modes of successful ignition: (i) direct ignition, where the hot plasma is deposited in a reactive fuel/oxidizer mixture and rapidly transitions into a sustained flame, (ii) indirect ignition, where the hot plasma is deposited outside the reactive mixture and transitions to a flame after the asymmetric plasma kernel slowly interacts with the turbulent jet, and (iii) failed ignition.

High-fidelity simulations can also provide detailed insights about ignition phenomena. For instance, Lacaze et al. [87] performed LES of a model rocket combustor [88] to examine ignition processes as well as flame propagation, anchoring, and stabilization mechanisms. Wang et al. [89] examined DNS of a laser kernel in quiescent air to demonstrate the importance of initial kernel morphology on influencing plasma-ejection that leads to the aforementioned indirect ignition. Ensemble studies with high-fidelity simulations can incur large computational costs, especially when treating combustion with finite-rate chemistry (FRC) [90, 91]. Thus, many ensemble simulations have employed cost-efficient combustion models involving global [92] or tabulated [93] chemistry.

Physics-based and ML-based reduced-order modeling approaches present a cost-effective alternative for evaluating the statistical behavior of forced ignition phenomena to predict the presence of successful ignition. Physics-based models typically combine (i) probabilistic arguments with (ii) flowfields obtained from high-fidelity inert simulations. Neophytou et al. [94] estimated ignition probability with an ensemble-based model that predicts volume fraction of burned gases using spark location, initial kernel size, as well as time-averaged turbulent velocity and mixture fraction flowfields that were extracted from inert LES. Esclapez et al. [95] employed flowfields from inert LES to develop a stochastic differential equation-based (SDE) model that directly transports PDFs to generate an ignition map for a swirl combustor. While these approaches have demonstrated their predictive capabilities, they do not consider phenomena related to kernel morphology and plasma-ejection found in laser ignition systems. Using ML, Sforzo and Seitzman [96] developed a support vector machine-based approach for predicting ignition probability within partially-stirred reactor simulations. Popov et al. [97] showed that convolutional NNs (CNNs) outperformed other reduced-order approaches in ignition boundary prediction of a jet-in-crossflow configuration. The authors noted that this deep learning approach can extract spatial information effectively, but suffers from poor OOD predictions if developed without considering essential physics.

This summarizes related efforts in predictive modeling (via both physics/chemistry- and ML-based approaches) of multi-physics flow phenomena found within propulsion

systems. Throughout this section, we highlighted the importance of sufficient training data in enabling the development of ML-based approaches in a variety of multi-physics flow problems involving closure modeling, combustion modeling, and reduced-order modeling – especially when extrapolating ML models to OOD conditions. Next, we present the research objectives contained within this dissertation.

1.3 Objectives

To address issues that can arise from current gaps in multi-physics flow data, the work in this dissertation has the following objectives:

- To directly address gaps in multi-physics flows data through the curation of large datasets.
- To examine opportunities provided by various ML methods in multi-physics flow problems with limited data.
- To develop a strategy for mitigating OOD errors during the integration of ML methods with physical models.
- To identify benefits from combining physics-based knowledge with ML in treating data limitations.

1.4 Accomplishments

Here, we present the key accomplishments found within this dissertation in the context of the objectives discussed in Section 1.3. In summary, we:

- Curated and distributed the largest public dataset of turbulent compressible reacting flows, aimed towards training ML models [21].
- Performed a reproducible benchmark that provided insight towards the scaling behavior of deep learning models in a multi-physics flow problem [21].

- Demonstrated the utility of interpretable ML approaches involving sparse symbolic regression and CaRTs in discovering analytic expressions of closure models for turbulent transcritical flows [7].
- Integrated an ML-based classifier with well-understood combustion models within a multi-physics flow solver that enabled the user-specified control of the cost-accuracy trade-off when simulating a rocket combustor configuration [9].
- Developed a reduced-order modeling approach by combining a deep learning model with a physics-based differential equation for capturing laser-induced ignition within a rocket combustor configuration [10].

1.5 Dissertation Outline

The remaining parts of this dissertation are structured as follows:

- Chapter 2 provides the governing equations for the multi-physics high-fidelity flow simulations presented in this work. In addition, supervised learning methods, focusing on CaRTs [24] and NNs [25], will be outlined in this chapter as well.
- Chapter 3 describes a 2.2 TB dataset containing 744 full-domain samples from 34 high-fidelity DNS of turbulent flows. With this data, we benchmark a total of 49 variations of five deep learning approaches for turbulence modeling via 3D super-resolution (SR). In addition, we analyze scaling behavior of these models. We demonstrate that (i) deep learning predictive performance can scale with model size and cost, (ii) architecture choice matters significantly, especially for smaller models, and (iii) the benefits of physics-based losses can persist at moderate model sizes.
- Chapter 4 examines opportunities from two interpretable ML approaches, namely the random forest regressor and the sparse symbolic regression, in discovering analytic expressions for SGS closure for flows found in high-pressure propulsion

systems. To this end, inert and reacting DNS of transcritical LOX/GCH_4 flows are performed. Using this data, *a priori* analysis is performed on the Favre-filtered DNS data to compare the accuracy of random forest SGS-models with conventional physics-based SGS-models. SGS stresses calculated with the gradient model are shown to have good agreement with the exact terms extracted from filtered DNS. Results demonstrate that random forests can perform as effectively as algebraic models when modeling SGS stresses, when trained on a sufficiently representative database and with a suitable choice of the feature set. The employment of the random forest feature importance score is shown to enable the discovery of an analytic model for SGS stresses through sparse symbolic regression. This is approach is also employed demonstrated towards modeling the SGS temperature, a term that arises from filtering the non-linear real-fluid equation-of-state (EoS), via *a priori* analysis.

- Chapter 5 outlines the integration of random forest classifiers within a multi-physics flow solver for local and dynamic submodel assignment in turbulent combustion simulations. This method is demonstrated in simulations of a single-element gaseous-oxygen/gaseous-methane (GOX/GCH_4) rocket combustor; *a priori* as well as *a posteriori* assessments are conducted to (i) evaluate the accuracy and adjustability of the classifier for targeting different QoIs, and (ii) assess improvements, resulting from the data-assisted (DA) combustion model assignment, in predicting target QoIs during simulation runtime. Results from the *a priori* study show that random forests, trained with local flow properties as input variables and combustion model errors as training labels, assign three different combustion models – FRC, FPV, and inert mixing (IM) – with reasonable classification performance even when targeting multiple QoIs. Applications in *a posteriori* studies demonstrate improved predictions from DA simulations, in temperature and carbon dioxide (CO) mass fraction, when compared with monolithic FPV calculations.

- Within Chapter 6, we investigate the potential of combining flow physics knowledge with ML in predicting laser ignition of a gaseous CH_4/O_2 model rocket

combustor. To this end, we introduce a hybrid stochastic physics-embedded deep learning framework that combines sparse experimental data from Schlieren measurements with inert LES for predicting the spatio-temporal evolution of ignition kernel location and morphology. This model combines a SDE for modeling kernel dynamics and a deep learning model for representing kernel morphology, which is essential in laser ignition. Results demonstrate that this model can reasonably capture behavior associated with the three dominant ignition modes, namely direct, indirect, and failed ignition, along with statistics associated with kernel growth and position, at lower computational costs than high-fidelity reacting simulations. In addition, we demonstrate that this modeling framework can be employed for generating spatially resolved ignition probability maps by incorporating physics to represent kernel interaction with the turbulent jet. We note that limitations in accuracy can be observed when predicting with vastly OOD data. Nevertheless, these results demonstrate that this physics-embedded ML approach can statistically characterize forced ignition in a cost-effective manner, as long as sufficiently representative data is available.

- In Chapter 7, we summarize key conclusions from preceding chapters and discuss recommendations for future work.

Chapter 2

Theoretical and Computational Methods*

2.1 Governing Equations

2.1.1 Conservation Equations

Multi-physics flow phenomena in propulsion systems can be simulated by solving governing equations for transporting mass, momentum, energy, and scalar, respectively [98]:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_j}{\partial x_j} = 0\,, \tag{2.1a}$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j}\,, \tag{2.1b}$$

$$\frac{\partial \rho e^t}{\partial t} + \frac{\partial \rho e^t u_j}{\partial x_j} = -\frac{\partial p u_j}{\partial x_j} + \frac{\partial \tau_{ij} u_i}{\partial x_j} - \frac{\partial q_j}{\partial x_j}\,, \tag{2.1c}$$

$$\frac{\partial \rho \Phi_k}{\partial t} + \frac{\partial \rho \Phi_k u_j}{\partial x_j} = -\frac{\partial j_{kj}}{\partial x_j} + \dot{\omega}_k\,, \tag{2.1d}$$

*This chapter contains select figures and method descriptions from the ML review paper by Ihme et al. [6], with significant modifications made for this dissertation. M. Ihme, W.T. Chung, and A.A. Mishra contributed equally to reviewing ML fundamentals and applications.

with density ρ, velocity component u_i, pressure p, viscous stress tensor component τ_{ij}, specific total energy e^t, and heat flux component q_j. Φ_k represents transported chemical scalars, while j_k and $\dot{\omega}_k$ are the corresponding diffusion flux and source term, respectively.

When employing FRC, the chemical species mass fraction Y_k is transported, *i.e.*, $\Phi_k \equiv Y_k$ for species $k = [1, N_s]$ where N_s is the number of species. In this case, the diffusion flux j_{ki} for multi-component flow with the mixture-averaged diffusion model is defined as [99]:

$$j_{ki} = -\rho D_k \frac{\partial Y_k}{\partial x_i} + \rho Y_k \sum_{j=1}^{N_s} D_j \frac{\partial Y_j}{\partial x_i}\,, \tag{2.2}$$

where D_k is the molecular diffusivity of species k. Assuming negligible Dufour effects, heat flux q_i in a multi-component flow is defined as [98]:

$$q_i = -\lambda \frac{\partial T}{\partial x_i} + \sum_{k=1}^{N_s} j_{ki} h_k^s\,, \tag{2.3}$$

where λ is the thermal conductivity, T is the temperature, and h_k^s is the partial sensible enthalpy of species k.

For Newtonian flows, assuming negligible bulk viscosity, the stress tensor τ_{ij} is defined as:

$$\tau_{ij} = 2\mu \left(S_{ij} - \frac{1}{3} S_{kk} \delta_{ij} \right), \tag{2.4a}$$

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \tag{2.4b}$$

with dynamic viscosity μ.

The specific total energy e^t is defined as the sum of specific internal energy e^i and specific kinetic energy e^k:

$$e^t = e^i + e^k, \tag{2.5a}$$

$$e^k = \frac{1}{2} u_i u_i\,. \tag{2.5b}$$

In the compressible flow formulation used in this work, pressure p in Equation (2.1) is obtained from an EoS. For gas-like phases, this is typically represented with the ideal gas EoS:

$$p = \rho R T \,, \tag{2.6a}$$

$$R = \frac{R_u}{M_{mix}} \,, \tag{2.6b}$$

$$\frac{1}{M_{mix}} = \sum_{k=1}^{N_s} \frac{Y_k}{M_k} \,, \tag{2.6c}$$

with specific gas constant R, universal gas constant R_u, mean molar mass M_{mix} of a multi-component mixture, and molar mass M_k of species k.

For high pressure flows found in propulsion systems, the Peng-Robinson (PR) cubic EoS [100] can be employed to model real-fluid thermodynamics:

$$p = \frac{\rho R T}{1 - b\rho} - \frac{a\rho^2}{1 + 2b\rho - b^2\rho^2} \,, \tag{2.7}$$

where the coefficients a and b account for effects of intermolecular forces and volumetric displacement, and are dependent on temperature and composition [101]. Details regarding the evaluation of specific heat capacity, internal energy, and partial enthalpy from the PR EoS is described in Ma et al. [102].

2.1.2 Large-eddy Simulation

Due to large computational costs, solving Equation (2.1) directly via DNS is not tractable for simulating many real-world propulsion systems. LES offers a feasible approach that resolves only the large-scale flow structures, with the finest scales treated with closure models [46, 50–52]. Formally, the compressible LES equations

are explicitly obtained through Favre-filtering [103] an arbitrary quantity ϕ:

$$\widetilde{\phi} = \frac{\overline{\rho\phi}}{\overline{\rho}}\,, \tag{2.8a}$$

$$\overline{\phi(\mathbf{x})} = \int_V \phi(\mathbf{x})\mathcal{F}(\mathbf{x}-\mathbf{y})\,d\mathbf{y}\,, \tag{2.8b}$$

where $\widetilde{\cdot}$ denotes a Favre-filtered quantity, $\overline{\cdot}$ is a filtered quantity obtained through a volume V integral with a LES filter $\mathcal{F}$ across two spatial coordinate locations given by $\mathbf{x}$ and $\mathbf{y}$. After Favre-filtering, the governing equations become:

$$\frac{\partial\overline{\rho}}{\partial t} + \frac{\partial(\overline{\rho}\widetilde{u}_j)}{\partial x_j} = 0\,, \tag{2.9a}$$

$$\frac{\partial(\overline{\rho}\widetilde{u}_i)}{\partial t} + \frac{\partial(\overline{\rho}\widetilde{u}_i\widetilde{u}_j)}{\partial x_j} = -\frac{\partial\overline{p}}{\partial x_i} + \frac{\partial\overline{\tau}_{ij}}{\partial x_j} + \frac{\partial\tau_{ij}^{sgs}}{\partial x_j}\,, \tag{2.9b}$$

$$\frac{\partial(\overline{\rho}\widetilde{e}^t)}{\partial t} + \frac{\partial(\overline{\rho}\widetilde{e}^t\widetilde{u}_j)}{\partial x_j} = -\frac{\partial(\overline{p}\widetilde{u}_j)}{\partial x_j} + \frac{\partial(\overline{\tau}_{ij}\widetilde{u}_i)}{\partial x_j} + \frac{\partial(\tau_{ij}^{sgs}\widetilde{u}_i)}{\partial x_j} - \frac{\partial\overline{q}_j}{\partial x_j} - \frac{\partial q_j^{sgs}}{\partial x_j}\,, \tag{2.9c}$$

$$\frac{\partial(\overline{\rho}\widetilde{\Phi}_k)}{\partial t} + \frac{\partial(\overline{\rho}\widetilde{\Phi}_k\widetilde{u}_j)}{\partial x_j} = -\frac{\partial\overline{j}_{kj}}{\partial x_j} - \frac{\partial j_{kj}^{sgs}}{\partial x_j} + \overline{\dot{\omega}}_k\,, \tag{2.9d}$$

with superscript $\cdot^{sgs}$ denoting SGS quantities that are typically treated via closure modeling.

This dissertation largely focuses on the SGS stress tensor. For treatment of other SGS terms, we refer the reader to several texts [98, 103, 104] and review articles [105, 106]. The SGS stress tensor from the LES momentum equation (Equation (2.9b)) is given by:

$$\tau_{ij}^{sgs} = \overline{\rho}(\widetilde{u_i u_j} - \widetilde{u}_i\widetilde{u}_j)\,, \tag{2.10}$$

which can be treated by analytic models such as the Vreman [50] and gradient [46] models.

The Vreman SGS model [50], which is derived from the eddy-viscosity hypothesis, can be employed towards treating the unclosed SGS stresses τ_{ij}^{sgs} in Equation (2.9):

$$\tau_{ij}^{sgs,v} \approx -2\overline{\rho}\nu^{sgs}\widetilde{S}_{ij} + \frac{1}{3}\overline{\tau}_{kk}\delta_{ij}\,, \tag{2.11}$$

where the eddy viscosity ν^{sgs} is evaluated for a filter width $\overline{\Delta}$ as follows:

$$\nu^{sgs} = \mathcal{C}_v \sqrt{\frac{B}{g_{ij} g_{ij}}}\,, \tag{2.12a}$$

$$g_{ij} = \frac{\partial \widetilde{u}_i}{\partial x_j}\,, \tag{2.12b}$$

$$B = \beta_{11}\beta_{22} - \beta_{12}^2 + \beta_{11}\beta_{33} - \beta_{13}^2 + \beta_{22}\beta_{33} - \beta_{23}^2\,, \tag{2.12c}$$

$$\beta_{ij} = \overline{\Delta}^2 g_{ki} g_{kj}\,, \tag{2.12d}$$

where a Vreman coefficient $\mathcal{C}_v$ of 0.07 is typically used in isotropic turbulence [50].

The gradient model by Clark et al. [46] is extracted from the first term in the Taylor series expansion of the filtering operation, and is given by:

$$\tau_{ij}^{sgs,g} \approx \overline{\rho} \mathcal{C}_g \overline{\Delta}^2 \frac{\partial \widetilde{u}_i}{\partial x_k} \frac{\partial \widetilde{u}_j}{\partial x_k}\,, \tag{2.13}$$

where a coefficient $\mathcal{C}_g$ of 1/12 is typically used [46].

2.2 Machine Learning Methods

2.2.1 Supervised Learning

In supervised learning [22], data consists as a set of structured input-output pairs for a given learning task. The purpose of a supervised learning algorithm is to estimate *parameters* $\boldsymbol{\Theta}$ that can model an optimal *hypothesis* f_{ML} that approximates the distribution of label vectors $\boldsymbol{\Upsilon}$ during *inference* with feature vector $\boldsymbol{\chi}$ to generate predictions $\hat{\boldsymbol{\Upsilon}}$:

$$\hat{\boldsymbol{\Upsilon}} = f_{ML}(\boldsymbol{\chi}|\boldsymbol{\Theta})\,, \tag{2.14a}$$

$$f_{ML}(\boldsymbol{\chi}|\boldsymbol{\Theta}) \approx \boldsymbol{\Upsilon}\,, \tag{2.14b}$$

where the features may consist of raw data inputs in the so-called *end-to-end learning* [30] (typically done with deep learning), or pre-processed forms of the raw data

that have been subject to *feature extraction/engineering* [107].

Candidates for this optimal hypothesis are evaluated through an error measure, *i.e.*, a total objective *loss* function, J_{tot} [22]. During *training*, an iterative optimization scheme minimizes the error measure to estimate optimal parameters $\boldsymbol{\Theta}$. Across N_{samp} number of data samples, this loss function is expressed as:

$$J_{tot}[\boldsymbol{\Upsilon}, f_{ML}(\boldsymbol{\chi}|\boldsymbol{\Theta})] = \frac{1}{N_{samp}} \sum_{\text{all samples}} J_{task}[\boldsymbol{\Upsilon}, f_{ML}(\boldsymbol{\chi}|\boldsymbol{\Theta})] \,. \tag{2.15}$$

Similarly, the specific objective function also J_{task} depends on the given learning task. Within supervised learning, we note that the model parameters $\boldsymbol{\Theta}$ are the values that are determined solely during a training loop. For example, model parameters are equivalent to the model weights in NNs and linear regression. All other design choices that improve predictive accuracy (such as model choice, model architecture, and loss function choice) are known as *hyperparameters*. The selection of these hyperparameters are typically guided by a combination of empirical findings from existing literature, intuition from ML theory, and hyperparameter search (which involves a separate optimization procedure for determining optimal choices) [22].

In a classification task, an ML model f_{ML} learns to predict a class (indexed by $1 < k \leq N_{class}$ classes) from the features $\boldsymbol{\chi}$. Thus, prior to training, the prediction targets $\Upsilon_k \in \boldsymbol{\Upsilon}$ are labeled as:

$$\Upsilon_k = \begin{cases} 1 \,, & \text{if the sample belongs to the class } k, \\ 0 \,, & \text{otherwise.} \end{cases} \tag{2.16}$$

During training, the cross entropy loss:

$$J_{CE}(\boldsymbol{\phi}, \boldsymbol{\psi}) = -\sum_{k=1}^{N_{class}} \phi_k \log(\psi_k) \,, \tag{2.17}$$

which measures the difference between an estimated probability distribution $\boldsymbol{\psi}$ against a true distribution $\boldsymbol{\phi}$ via information theory [22], is minimized. When combining Equation (2.16) with Equation (2.17), the cross-entropy loss reduces to a negative

log-likelihood function:

$$J_{CE}[\boldsymbol{\Upsilon}, f_{ML}(\boldsymbol{\chi}|\boldsymbol{\Theta})] = -\log[f_{ML}(\boldsymbol{\chi}|\boldsymbol{\Theta})]\,, \tag{2.18}$$

where $0 < f_{ML}(\boldsymbol{\chi}|\boldsymbol{\Theta}) \leq 1$, which can be perceived as the likelihood function within statistics [22]. As such, Equation (2.18) can also be referred to as the negative log-likelihood function. By minimizing this function, the classification task can be viewed formally as a form of maximum likelihood estimation, *i.e.*, estimating the parameters of an assumed probability distribution given some observed data [22].

For regression problems, a common loss function is the mean-squared error (MSE):

$$J_{MSE}(\phi, \psi) = (\phi - \psi)^2\,, \tag{2.19}$$

which is simply the Euclidean distance between two arbitrary quantities ϕ and ψ. If the data possesses a Gaussian distribution, minimization of the MSE can also be shown to be equivalent to the minimization of the negative log-likelihood function [22].

In supervised learning, the performance of a model is often evaluated in terms of its ability to generalize to OOD data. This generalizability can be formally measured through a test generalization error ϵ_{gen}. By considering a supervised learning method evaluated via an MSE function, we can express this generalization error as the expectation $\mathbf{E}_{\mathcal{D}}$ of the squared error across all possible training datasets [22]:

$$\begin{aligned}\epsilon_{gen} =& \mathbf{E}_{\mathcal{D}}\left\{[\boldsymbol{\Upsilon} - f_{ML}(\boldsymbol{\chi}|\boldsymbol{\Theta})]^2\right\}\,,\\ =& \underbrace{\mathbf{E}_{\mathcal{D}}[f_{ML}(\boldsymbol{\chi}|\boldsymbol{\Theta})] - \mathbf{E}_{\boldsymbol{\Upsilon}|\boldsymbol{\chi}}(\boldsymbol{\Upsilon})}_{\text{Bias}} + \underbrace{\mathbf{E}_{\mathcal{D}}\left(\{\mathbf{E}_{\mathcal{D}}[f_{ML}(\boldsymbol{\chi}|\boldsymbol{\Theta})] - f_{ML}(\boldsymbol{\chi}|\boldsymbol{\Theta})\}^2\right)}_{\text{Variance}}\\ &+ \underbrace{\mathbf{E}_{\boldsymbol{\Upsilon}}[\boldsymbol{\Upsilon} - \mathbf{E}_{\boldsymbol{\Upsilon}|\boldsymbol{\chi}}(\boldsymbol{\Upsilon})]}_{\text{Noise}}\,,\end{aligned} \tag{2.20}$$

where $\mathbf{E}_{\boldsymbol{\Upsilon}}$ is the expectation across the labels and $\mathbf{E}_{\boldsymbol{\Upsilon}|\boldsymbol{\chi}}$ is the feature-conditioned expectation across labels.

As shown in Equation (2.20), the generalization error ϵ_{gen} which can be decomposed to model variance, bias, and noise terms. Model bias represents the model error

that is independent of data encountered, and can be viewed as the inherent error in the type of ML model chosen. An example of this involves the linear regression model, which is biased to assume a linear relationship between features and labels. This biased assumption results in poor capturing of any non-linear relationships within the data, and can result in *underfitting*. Model variance captures the sensitivity of the ML model to variations across different datasets. A model with high variance will capture patterns in a given training data that may not be found in other similar datasets. This issue is found in *expressive* ML models with large number of model parameters, such as decision trees or deep learning models. The noise term represents the irreducible error that arises from uncertainties in feature/label representation.

2.2.2 Linear Regression

As mentioned in Section 2.2.1, linear regression [23] is one of the simplest forms of supervised learning. A linear model f_{lin} is typically expressed as the weighted sum of input $\boldsymbol{\chi}$:

$$f_{lin}(\boldsymbol{\chi}|\boldsymbol{\Theta}) = \sum_{j=1}^{N_{feat}} \Theta_{jk}\chi_j \,, \tag{2.21}$$

with k number of outputs and N_{feat} is the number of features $\chi_j \in \boldsymbol{\chi}$. During supervised learning, the MSE (Equation (2.19)) loss can be minimized to estimate the optimal model parameters $\boldsymbol{\Theta}$.

2.2.3 Sparse Symbolic Regression

In sparse regression, the loss function across all samples is regularized with the l_1-norm of the model parameters/coefficient or *lasso* method [108]:

$$J_{tot}[\boldsymbol{\Upsilon}, f_{lin}(\boldsymbol{\chi}|\boldsymbol{\Theta})] = \frac{1}{N_{samp}} \sum_{\text{all samples}} J_{MSE}[\boldsymbol{\Upsilon}, f_{lin}(\boldsymbol{\chi}|\boldsymbol{\Theta})] + \Lambda||\boldsymbol{\Theta}||_1 \,, \tag{2.22}$$

where Λ is a regularization parameter for controlling the trade-off between the MSE loss and the l_1-norm that is determined via a hyperparameter search. During optimization, the l_1-norm of the model coefficients $||\boldsymbol{\Theta}||_1$ are also minimized. This encourages sparsity, *i.e.*, reduces the number of terms in the linear model, as zero-valued model parameters are favored during optimization.

This regularized regression approach can be useful for discovering symbolic/analytic models [109]. This is done by including candidate model terms as the regression input $\boldsymbol{\chi}$, and allowing an optimization method to discover the optimal linear combination of terms that result in a symbolic expression that matches a set of target values $\boldsymbol{\Upsilon}$. The inclusion of the sparse loss regularization term $\Lambda||\boldsymbol{\Theta}||_1$ results in zero-valued coefficients for candidate terms that are not relevant to the target symbolic expression. To extend the method beyond linear expressions, non-linearities can be introduced by replacing $\boldsymbol{\chi}$ with non-linear terms $\sigma(\boldsymbol{\chi})$ constructed from the original variables. In this dissertation, we construct a model with non-linear variables by evaluating d-order polynomial functions:

$$f_{sym}(\boldsymbol{\chi}|\boldsymbol{\Theta}) = f_{lin}(\sigma(\boldsymbol{\chi})|\boldsymbol{\Theta})\,, \tag{2.23a}$$

$$\sigma(\boldsymbol{\chi}) = \begin{bmatrix} 1 & \chi_1 & \chi_2 & \cdots & \chi_n & \chi_1^2 & \chi_1\chi_2 & \cdots & \chi_n^d \end{bmatrix}, \tag{2.23b}$$

where n number of features in the original feature vector $\boldsymbol{\chi}$. Equation (2.23) shows that the dimensionality of this approach scales to the order of polynomial functions $\mathcal{O}(n^d)$. Hence, the number of candidate variables must be reduced for this method to remain tractable.

2.2.4 Neural Networks

NNs consist of successive layers of weighted mathematical operations that are arranged in a network structure [22]. The outputs of a k-layered NN can be expressed

as the output of the final layer $\boldsymbol{h}_k$:

$$f_{NN}(\boldsymbol{\chi}|\boldsymbol{\Theta}) = \boldsymbol{h}_k\,, \tag{2.24a}$$

$$\boldsymbol{h}_n = f_n(\boldsymbol{h}_{n-1}|\boldsymbol{\Theta}_n)] \quad \text{where} \quad n = [1,\ldots,k]\,, \tag{2.24b}$$

$$\boldsymbol{h}_0 \equiv \boldsymbol{\chi}\,, \tag{2.24c}$$

where f_n is an arbitrary mathematical operation and $\boldsymbol{\Theta}_n$ represents the model weights – both at the n-th layer. Since the inputs $\boldsymbol{\chi}$ are propagated successively from the zero- to k-th NN layers during inference, Equation (2.24) is known as *forward propagation*.

Training a NN involves the estimation of the model parameters $\boldsymbol{\Theta}$ through minimization of a loss function J_{tot} via an iterative *gradient descent* optimization scheme. Model parameters in NNs can be updated via a vanilla batch gradient descent scheme [22] in can be expressed as:

$$\boldsymbol{\Theta}_n^{it+1} = \boldsymbol{\Theta}_n^{it} - \Lambda \frac{\partial J_{tot}}{\partial \boldsymbol{\Theta}_n^{it}}\,, \tag{2.25}$$

where Λ is a tunable learning rate, which scales the size of the parameter update, that is typically determined via a hyperparameter search. During gradient descent, large parameter updates result in faster training but can result in convergence issues. In contrast, smaller parameter updates are more computationally stable, but can be costly as more iterations are needed to converge to an optimal condition. Thus, a fixed learning rate in the vanilla gradient descent scheme can lead to poor convergence.

As such, recent ML models employ an adaptive gradient descent scheme. The Adaptive Moment (Adam) estimation scheme [110] is a popular optimization approach

that changes the size of parameter updates through moving averages:

$$\boldsymbol{\Theta}_n^{it+1} = \boldsymbol{\Theta}_n^{it} - \Lambda \frac{\boldsymbol{\gamma}_n^{*,it}}{\sqrt{\boldsymbol{\Gamma}_n^{*,it}} + \mathcal{C}_0}\,, \tag{2.26a}$$

$$\boldsymbol{\gamma}_n^{*,it} = \frac{\boldsymbol{\gamma}_n^{it}}{1-(\mathcal{C}_1)^{it}}\,, \tag{2.26b}$$

$$\boldsymbol{\Gamma}_n^{*,it} = \frac{\boldsymbol{\Gamma}_n^{it}}{1-(\mathcal{C}_2)^{it}}\,, \tag{2.26c}$$

$$\boldsymbol{\gamma}_n^{it} = \mathcal{C}_1 \boldsymbol{\gamma}_n^{it-1} + (1-\mathcal{C}_1)\frac{\partial J_{tot}}{\partial \boldsymbol{\Theta}_n^{it}}\,, \tag{2.26d}$$

$$\boldsymbol{\Gamma}_n^{it} = \mathcal{C}_2 \boldsymbol{\Gamma}_n^{it-1} + (1-\mathcal{C}_2)\left(\frac{\partial J_{tot}}{\partial \boldsymbol{\Theta}_n^{it}}\right)^2\,, \tag{2.26e}$$

where $\mathcal{C}_0$ is a small constant added for numerical stability, with coefficients $\mathcal{C}_1 = 0.9$ and $\mathcal{C}_2 = 0.999$ typically used to scale the contribution (or *momentum*) of moving averages of the NN gradients $\boldsymbol{\gamma}_n^{it}$ and the squared gradients $\boldsymbol{\Gamma}_n^{it}$ from previous iterations. $\hat{\boldsymbol{\gamma}}_n^{it}$ and $\hat{\boldsymbol{\Gamma}}_n^{it}$ are the moving averages that are scaled to correct for the bias from previous gradient iterations.

For a k-layered NN, the model weights $\boldsymbol{\Theta}_n^{it}$ of the n-th layer are updated by passing back gradients (*i.e.*, *backpropagation* [25]) from succeeding layers through the chain rule, in each iteration:

$$\frac{\partial J_{tot}}{\partial \boldsymbol{\Theta}_n^{it}} = \frac{\partial J_{tot}}{\partial \boldsymbol{h}_k}\frac{\partial \boldsymbol{h}_k}{\partial \boldsymbol{h}_{k-1}} \cdots \frac{\partial \boldsymbol{h}_{n+1}}{\partial \boldsymbol{h}_n}\frac{\partial \boldsymbol{h}_n}{\partial \boldsymbol{\Theta}_n^{it}}\,. \tag{2.27}$$

The initial weights can be initialized randomly through a uniform or normal distribution [22]. However, when many layers are used in deep learning models, *i.e.*, $n \gg 1$, NN gradients can *explode* (*i.e.*, gradients approach infinity) or *vanish* (*i.e.*, gradients approach zero) due to large number of multiplications performed during chain rule operations. For an NN with N_{param} number of weights, scaling the initial random weights to an appropriate range with a standard deviation of $\mathcal{O}(1/\sqrt{N_{param}})$ has been shown to ameliorate this issue. Glorot [111] and He [112] initialization are popular variations incorporating this scaling approach.

Note that Equation (2.25) expresses vanilla batch gradient descent across the

Algorithm 1 Pseudocode for Mini-batch Gradient Descent with Adam optimizer.

Require: Training data, learning rate, initial parameters, batch size,
while not converged **do**
  Randomly shuffle the training data
  for each mini-batch **do**
    for each training sample in mini-batch **do**
      Compute gradients with Equation (2.27).
    end for
    Compute mean of gradients in mini-batch
    Update parameters $\boldsymbol{\Theta}$ with Equation (2.26).
  end for
end while
Ensure: Updated parameters $\boldsymbol{\Theta}$

entire *batch* of available training data. Evaluating and storing the gradients for large number of samples can be computationally expensive. Thus, parameter updates can also be performed after gradient computations of a *mini-batch* [30] (with a user-defined size) can be partitioned from the training data, as shown in the training loop in Algorithm 1. The batch size can be treated as a hyperparameter, but is often chosen based on memory constraints of computing hardware. Here, *epochs* are typically used to measure the number of optimization iterations in mini-batch optimization. Specifically, one epoch corresponds to the number of iterations required to loop through all training data once. During optimization, it is common to train for multiple epochs (depending on the available number of data samples).

Fully Connected Layers

The outputs of the n-th layer h_n of this architecture [22] (shown in Figure 2.1) are computed via:

$$\boldsymbol{h}_n = \sigma_n[f_{lin}(\boldsymbol{h}_{n-1}|\boldsymbol{\Theta}_n)]\,, \tag{2.28}$$

where σ_n is an activation function at the n-th layer that is typically used to introduce non-linearities to the mathematical operations. The sigmoid activation function has traditionally been employed in early NN work [22]. For a single input χ, this can be

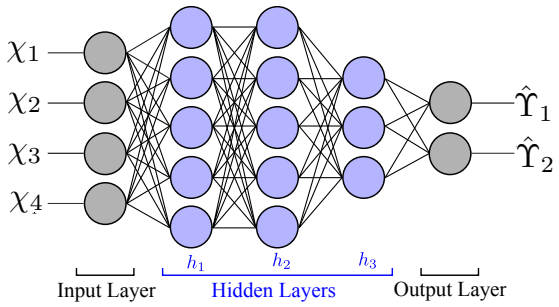


Figure 2.1: A fully connected NN with four features $\boldsymbol{\chi}$, 2 predictions $\hat{\boldsymbol{\Upsilon}}$, and three hidden layers $\boldsymbol{h}$.

expressed as:

$$\text{sigmoid}(\chi) = \frac{1}{1 + \exp(-\chi)}\,. \tag{2.29}$$

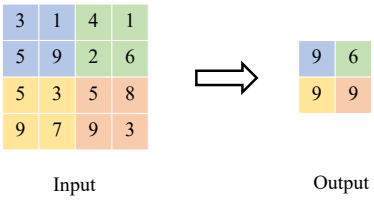
However, later studies have found that the presence of near-zero gradients of this function can lead to vanishing gradients in backpropagation, especially with large number of NN layers [30]. Thus, the rectified linear unit (ReLU) [112] function is preferred as the activation function in more recent work:

$$\text{ReLU}(\chi) = \max(0, \chi)\,. \tag{2.30}$$

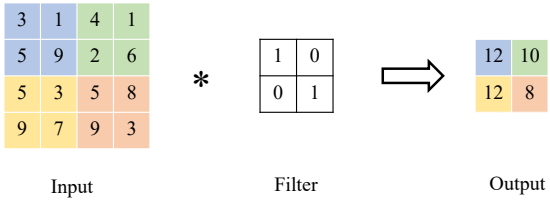
Convolutional layers

Convolutional NN [113] (CNN) architectures consist of convolutional layers made up of multiple *filters*. In computer vision, a filter is a function that operates on a local neighborhood of a pixel to generate an output [30]. These can be viewed as moving windows that move across at images, flowfields, or multidimensional tensors for extracting features such as texture, edges, and flowfield-related pattern.

The simplest filter replaces the corresponding pixel in the output by the maximum value of the pixel's neighborhood in the input. These replacements correspond to the pooling operations used in NNs (Figure 2.2a), where a max-pooling operation is employing using a 2×2 filter, with the filter moving across two rows and columns (strides). A more complicated filtering operation involving convolutions (Figure 2.2b)



(a) Max-pooling with a 2×2 filter and a stride of 2.



(b) Convolution with a 2×2 filter and a stride of 2.

Figure 2.2: Operations in a CNN.

occurs when the filtered result $h_{i,j} \in \boldsymbol{h}$, at row i and column j of a 2D matrix (representing an image or flowfield) is a weighted sum over a small neighborhood of pixels $\chi_{i+k,j+l} \in \boldsymbol{\chi}$:

$$h_{i,j} = \sum_{k=1}^{K} \sum_{l=1}^{L} \chi_{i+k,j+l} \, \Theta_{k,l} \, . \tag{2.31}$$

The entries of the weight filter or mask $\Theta_{k,l} \in \boldsymbol{\Theta}$ are referred to as the filter coefficients and K and L are the widths of the filter. Note that while this chapter has focused on 2D CNNs, these operations can be arbitrarily extended for larger number of dimensions.

Depending on the problem, filter sizes and strides can be chosen via hyperparameter tuning or by the following intuition: filters which are too large will result in large information losses, while filters which are too small will result in low sharing of information with neighboring pixels [30]. Using large strides has the same effect of downsampling the spatial data.

Deep Learning

A deep learning model is an NN model with more than approximately ten layers [31], where the specific design of each deep learning architecture can be further motivated

by empirical findings and intuition. One empirical finding is that the sole use of fully connected layers within an NN does not lead to significant benefits in prediction beyond five layers [114]. This is one reason why fully connected NNs (without other types of layers) are sometimes considered traditional ML, and not deep learning [30]. Another empirical finding is that deep learning models tend to improve in predictive performance with increasing number of model parameters [16]. However, since computational complexity within 2D convolutional and fully connected layers scales approximately quadratically with the dimensionality of layer inputs, deep and narrow architectures are preferred over shallow and broad architectures [30].

As previously mentioned, the reliance on chain rule (Equation (2.27)) for training can also result in issues related to vanishing and exploding gradients. To address this, several layer normalization schemes have been proposed [115, 116]. Another solution seen in popular deep learning models such as ResNet [114] and U-Net [117] is the use of residual/skip connections. This type of NN mechanism uses a residual addition operation to allow backpropagation-based training to skip a few layers, thus reducing the amount of chain rule operations that can lead to zero- or infinite-valued gradients. Formally, the output of a residual connection $\boldsymbol{h}_n^{skip}$ is the sum of the outputs of the n-th layer $\boldsymbol{h}_n$ with the outputs of the m-th preceding layer $\boldsymbol{h}_{n-m}$:

$$\boldsymbol{h}_n^{skip} = \boldsymbol{h}_n + \boldsymbol{h}_{n-m} \,. \tag{2.32}$$

2.2.5 Classification and Regression Trees

CaRTs are a family of supervised learning approaches that represents complex relationships by recursively partitioning a high-dimensional input space [24]. Here, we discuss the decision tree and the random forest algorithms.

Decision Trees

A decision tree [118] is a CaRT with a tree structure that consists of nodes representing decision points and branches representing outcomes. The n-th internal node $\mathcal{V}_n$ corresponds to a feature in the input space and splits the data based on a threshold

value Θ_j at the j-th split, leading to two or more child nodes. The final layer of nodes, *i.e.*, the leaf nodes, of the tree represent the final decision or prediction, which can be a class label in classification trees or a continuous value in regression trees. As shown in Algorithm 2, inference is performed on a decision tree through a recursive search [24].

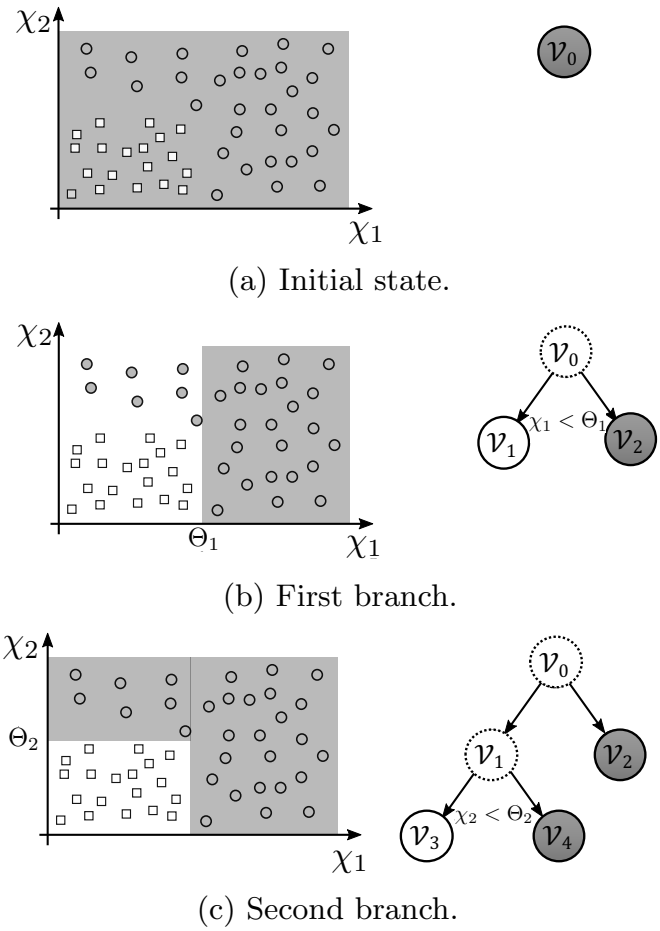
Algorithm 2 Pseudocode for Recursive Decision Tree Inference

```
procedure PREDICT(node, features χ)
    if node is a leaf then
        return node.label
    end if
    if node.feature(χ) ≤ node.threshold then
        return PREDICT(node.leftChild, χ)
    else
        return PREDICT(node.rightChild, χ)
    end if
end procedure
```

Figure 2.3 shows the construction of a decision tree, which involves selecting the optimal feature $\chi_{opt} \in \boldsymbol{\chi}$ and threshold value Θ_j to split the data at the j-th split. This selection is performed through the optimization of a loss function that measures the purity of resulting partitions. In this dissertation, the Gini impurity [24] is minimized for training classification trees. For the l-th feature at the j-th split, this is expressed as:

$$J_{Gini} = 1 - \sum_{k=1}^{N_{class}} \phi_k(\chi_l, \Theta_j)^2 \,, \tag{2.33}$$

where $0 < \phi_k(\chi_l, \Theta_j) \leq 1$ is the proportion of samples belonging to class k in a given node. Thus, by minimizing J_{Gini}, the proportion of samples belonging to the correct class (*i.e.*, purity) is being maximized. For regression problems, the optimal split threshold can be determined by minimizing the MSE loss (Equation (2.19)) [24]. As summarized in Algorithm 3, the splitting of the data $\mathcal{D}$ at the nodes is performed recursively until a stopping criterion is met [24]. Typical stopping criterion that are determined via hyperparameter search include minimum number of data samples in



(a) Initial state.

(b) First branch.

(c) Second branch.

Figure 2.3: Step-wise construction of a decision tree for a two-dimensional feature space with binary classification.

a node or the depth of the decision tree.

Random forests

In relation to Equation (2.20), decision trees have low model bias but high model variance [119]. As such, decision trees are prone to overfitting to the training data. Since model variance is associated with sensitivities to a specific training dataset, model variance can be reduced by forming a random forest [119], *i.e.*, an ensemble of decision trees that have been trained on different variations of the initial training data.

Algorithm 3 Pseudocode for Recursive Decision Tree Training

```
procedure GrowTree(data 𝒟, loss J_tot, depth)
    if StoppingCondition(𝒟, depth) then
        return CreateLeafNode(label(𝒟))
    end if
    optimal feature χ_opt, optimal threshold Θ_j ← FindBestSplit(𝒟, J_tot)
    𝒟_left, 𝒟_right ← SplitData(𝒟, χ_opt, Θ_j)
    leftChild ← GrowTree(𝒟_left, depth + 1)
    rightChild ← GrowTree(𝒟_right, depth + 1)
    return CreateTreeNode(χ_opt, Θ_j, leftChild, rightChild)
end procedure
```

This process of ensembling with permutations of the training data resampled from the initial training data is known as *bootstrap aggregating* or *bagging*. When growing each constituent decision tree in a random forest, the set of features used in determining the optimal threshold Θ_k (see Algorithm 3) is also subsampled randomly in each tree, which ensures further variations in between the trees. Here, the number of decision trees and feature subsample size are additional hyperparameters that are tuned by the user.

Inference with a random forest algorithm involves collecting independent predictions from each constituent decision tree, followed by aggregating the resulting ensemble. In classification, majority voting is a common approach to aggregating predictions, while in regression, the mean ensemble prediction is typically employed.

2.2.6 Machine Learning Interpretability

When employing ML models, interpretability can be useful for understanding model shortcomings, faults in code, and aid in scientific discovery [120]. Linear regression (as discussed in Section 2.2.2) is often regarded as highly interpretable [120]. Assuming that all features have been scaled to similar magnitudes, the magnitudes of the model weights provides a direct relationship of the influence any given feature has on the prediction $f_{lin}(\boldsymbol{\chi}|\boldsymbol{\Theta})$. Specifically, larger absolute values of a given model weight signify that changes in the predicted outcome are more sensitive to a corresponding feature. In NNs, this direct relationship between model parameters and data features

is obscured during the forward pass of features across multiple non-linear hidden layers (as described in Section 2.2.4). As such, NN and deep learning approaches have earned a reputation for being black-box models, which are capable of providing accurate predictions at the expense of being uninterpretable.

In contrast, CaRT approaches are known for providing a balanced trade-off between accuracy and interpretability. This is because the features and thresholds (discussed in Section 2.2.5) used to split data samples into the models' tree structure can be examined to understand model behavior. One such measure of interpretability from CaRT approaches is the feature importance score, which is often measured through a mean decrease in impurity (MDI) that is given by [121]:

$$\text{Feature Importance}(\chi_{opt}) = \frac{\sum_{\text{nodes using } \chi_{opt}} \text{MDI}}{\sum_{\text{all nodes}} \text{MDI}}\,, \tag{2.34}$$

where the MDI at a given CaRT node provides a numerical value for the effectiveness of a given feature $\chi_{opt} \in \boldsymbol{\chi}$ in splitting the data during training (see Algorithm 3).

The MDI used for regression in this dissertation is evaluated through the reduction in variance of labels at a given node $\text{Var}(\boldsymbol{\Upsilon}_{node})$ [121]. The variance reduction by a feature at a single node represents the improvement in homogeneity/purity of the target variables $\boldsymbol{\Upsilon}$ due to the split at a given node based on a particular feature, and is calculated as follows [24]:

$$\text{MDI} = \text{Var}(\boldsymbol{\Upsilon}_{parent}) - (\phi_{left}\text{Var}(\boldsymbol{\Upsilon}_{left}) + \phi_{right}\text{Var}(\boldsymbol{\Upsilon}_{right}))\,, \tag{2.35}$$

where ϕ_{left} and ϕ_{right} are the proportions of the data going to the left and right child nodes, respectively. Similarly, $\boldsymbol{\Upsilon}_{left}$ and $\boldsymbol{\Upsilon}_{right}$ are the training labels that are split into the left and right child nodes, respectively, while $\boldsymbol{\Upsilon}_{parent}$ is the training labels in the parent node (*i.e.*, prior to splitting). Thus, the feature importance score in a regression tree measures the contribution of each feature within $\boldsymbol{\chi}$ to reducing the overall variance of the target variable, reflecting how important the feature is for generating accurate predictions within the model.

2.2.7 Further Practical Considerations

The development of ML models is often influenced by trade-offs, as seen in Section 2.2.6 with the discussion on the interpretability-expressiveness trade-off across different ML approaches. The decomposition of the generalization error (Equation (2.20)) is often related to the bias-variance trade-off, *i.e.*, the trade-off associated with model underfitting and overfitting. This trade-off often guides the practical choices in the development of a supervised learning model, such as the treatment of the learning data, as well as model design and selection.

Practically, high bias is associated with underfitting, when the selected ML model is insufficiently expressive for learning a given task. For instance, shortcomings in linear regression in modeling non-linear quantities seen within closure modeling can be treated through the use of non-linear regressors, such as CaRTs and NNs. The expressiveness of an NN can be increased by increasing the number of model parameters within its architecture. Therefore, a straightforward approach towards addressing underfitting issues involves the selection of more expressive and scalable approaches, typically involving NNs or CaRT-based approaches [22].

However, a model that has too many parameters for a given task can overfit, *i.e.*, where the model cannot predict well on datasets that are substantially OOD from the data seen during training because of the spurious patterns learned from an insufficiently representative dataset. A brute-force approach for treating overfitting can involve increasing the scale of training data, which ameliorates the tendency of an ML model in learning spurious trends through an increased diversity of learning data [16]. A dataset can also be synthetically generated through data augmentation [122], which increases the diversity of training data through randomized data transformations (such as flowfield rotations, flipping, and cropping).

Another naïve approach involves reducing the number of model parameters by reducing the number of NN layers or decision tree depth. However, this is not an optimal approach if trends within the data are not homogeneous, *i.e.*, when different regimes within the data require different levels of expressiveness. The introduction of soft constraints and assumptions to the model could offer better opportunities for ameliorating overfitting under these mixed regime conditions [22]. For example, the

inclusion of the l_2-norm $\|\mathbf{\Theta}\|_2$ to the total loss J_{tot} is known to regularize the model through encouraging the presence of uniform model weight distributions in NNs. This added constraint reduces the over-reliance on any individual input in each NN layer, which reduces the tendency of the model to learn spurious trends, while maintaining expressiveness from a large number of model parameters. A similar principle is often employed in many scientific ML problem in physics-informed losses [123], where loss regularization terms associated with conservation equations can be used to constrain NN behavior.

In order to monitor the occurrence of overfitting and underfitting during learning, a common practice involves monitoring the objective function J_{tot} across representative samples of unseen data, *i.e.*, a test set [30]. Practically, this test dataset is often curated through randomly selecting the learning data in an approximately 80:20 ratio for the training and test data, respectively [30]. Here, model parameters $\mathbf{\Theta}$ are tuned via an iterative optimization scheme during training, while the test data kept separately for the sole use of evaluation. Since there are hyperparameters (such as model type, number of ensembles, number of NN layers, and regularization approach) that are not accounted for during training, it is also common practice to curate a validation dataset by further subsampling the test set via random selection for the sole purpose of hyperparameter tuning. This results in a typical 80:10:10 ratio for train, validation, and test sets, respectively [30].

Chapter 3

Large Multi-Physics Flow Dataset for Machine Learning*

3.1 Introduction

As discussed in Chapter 1, the predictive performance of deep learning methods scales with the volume of training data. However, the availability of datasets within multi-physics flow domains can be limited when compared to other ML domains. In this chapter, we directly address these gaps through the development of a large scale ML dataset for turbulent reacting and non-reacting flows.

To this end, we present the Bearable Large Accessible Scientific Training Network-of-datasets (BLASTNet), a cost-effective community-driven weakly centralized framework that utilizes a public repository for increasing access to scientific data. With this approach, we curate BLASTNet 2.0, a dataset with 744 full-domain samples from 34 DNS configurations, which will be the focus of this chapter. This dataset aims to address limitations in data availability for compressible turbulent non-reacting

*This chapter contains previously published work from Chung et al. [21], with minor modifications. W.T. Chung planned, implemented, performed, and analyzed experiments, developed the ML dataset, and implemented ML models. B. Akoush and P. Sharma assisted in the development of the ML dataset and implementation of ML models. A. Tamkin assisted with planning experiments. K.S. Jung, J.H. Chen, J. Guo, D. Brouzet, M. Talei, B. Savard, and A.Y. Poludnenko contributed to dataset development.

and reacting flows, which are found in propulsion [124, 125], automotive [126, 127], energy [39, 128], and environmental [129, 130] applications. BLASTNet data is potentially suited for ML problems in turbulent non-reacting and reacting flows, which can also involve inverse problems [131, 132], physics discovery [7, 133], dimensionality reduction, regime classification, and turbulence-chemistry closure modeling [134].

In this chapter, we also demonstrate the utility of BLASTNet data for 3D SR of turbulent flows [21]. To this end, we pre-process DNS data from BLASTNet 2.0 to form the Momentum128 3D SR dataset for benchmarking this task. While SR via deep learning has been subjected to numerous competition and benchmark studies that target problems within other ML domains (such as computer vision [135–139], 3D medical imaging [140], and 3D microscopy [141]), extensive and easily reproducible benchmarks on SR models for turbulent flows have not been performed prior to this work. Within multi-physics flows, studies on ML-based SR are nascent compared to these other ML domains, and have largely focused on demonstrating feasibility [44, 142–145], mostly by modifying existing image SR models with convolutional architectures. Due to computational memory constraints, many of these studies focus on 2D configurations [142, 143, 145], with 3D SR investigations only demonstrated recently [44, 144].

As summarized in Figure 3.1, we:

- Curate BLASTNet 2.0, a diverse public 3D compressible turbulent flow DNS dataset.
- Benchmark performance and cost of five 3D ML approaches [132, 146–149] for SR with this publicly accessible dataset.
- Show that SR model performance can scale with the logarithm of model size and cost.
- Demonstrate the persisting benefits of a popular physics-based gradient loss term [132] with increasing model size.

In Section 3.2, we provide information on the BLASTNet 2.0 and Momentum128 3D SR datasets. Our benchmark setup is described in Section 3.3, with results discussed

in Section 3.4, before the conclusions in Section 3.5.

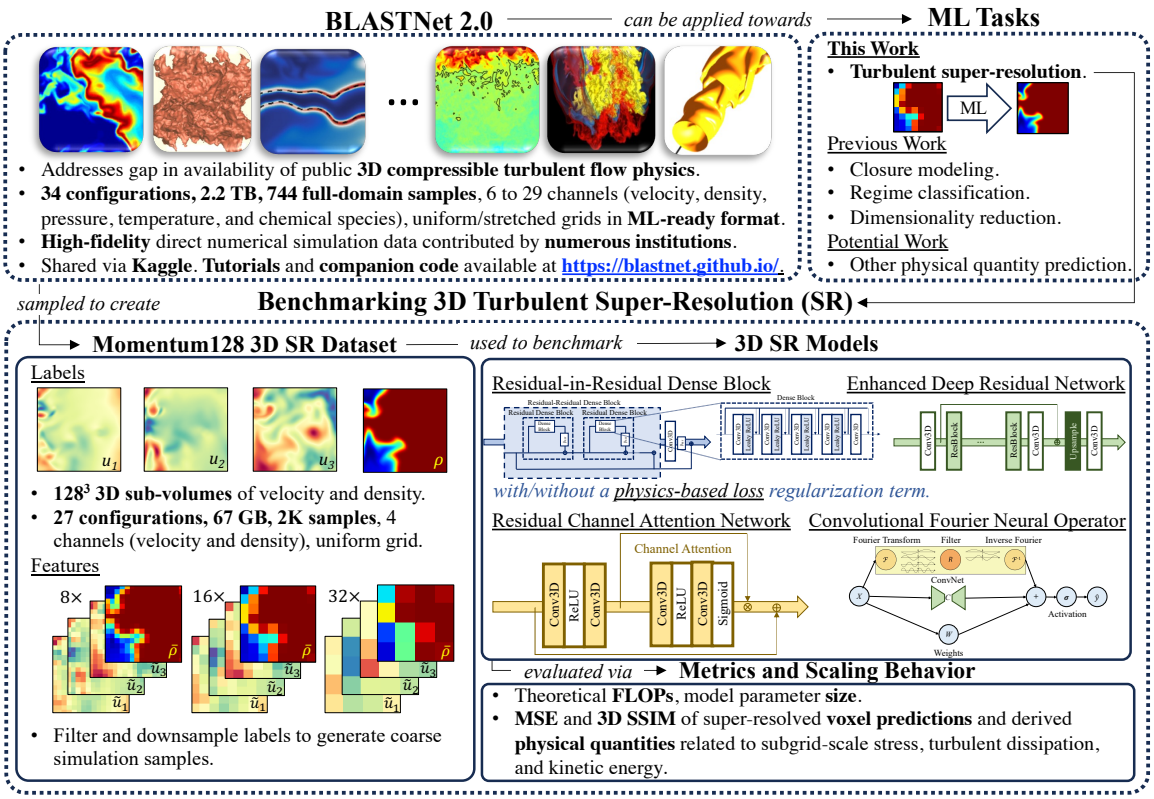


Figure 3.1: Summary of this chapter.

3.2 Datasets

3.2.1 BLASTNet 2.0

BLASTNet 2.0 consists of turbulent compressible flow DNS data, on Cartesian spatial grids, generated by solving governing equations for mass, momentum, energy, and chemical species, *i.e.*, Equation (2.1). The BLASTNet 2.0 dataset is developed with these properties in mind:

Fidelity All DNS data is collected from well-established numerical solvers [37, 102, 150–152] with spatial discretization schemes ranging from 2nd- to 8th-order accuracy,

while time-advancement accuracy range from 2nd- to 4th-order. Low-order schemes require finer discretization compared to high-order schemes, to achieve similar accuracy and numerical stability [153]. However, all simulations are spatially resolved to the order of the Kolmogorov length-scale, ranging from 3.9 to 41 μm depending on the configuration, with a corresponding temporal discretization that ensures numerical stability.

Size and Diversity BLASTNet 2.0 contains a total of 744 full-domain samples (2.2 TB) from a diverse collection of 34 simulation configurations: non-reacting decaying homogeneous isotropic turbulence (HIT) [7], reacting forced HIT [152], two parametric variations of reacting jet flows [37], six configurations of non-reacting transcritical channel flows [154], a reacting channel flow [36], a partially-premixed slot burner configuration [35], and 22 parametric variations (with different turbulent and chemical time-scales) of a freely-propagating flame configuration [155].

Community-involvement BLASTNet 2.0 consists of data contributions from six different institutions. As mentioned in Appendix A.1, our long-term vision and maintenance plan for this dataset involves seeking additional contributions from members of the broader flow community.

Cost-effective Storage, Distribution, and Browsing To circumvent Kaggle storage constraints, we partition the data into a network of < 100 GB subsets, with each subset containing a separate simulation configuration. This partitioned data can then be uploaded as separate datasets on Kaggle. To consolidate access to this data, all Kaggle download links are presented in `https://blastnet.github.io`, with the inclusion of a `bash` script for downloading all data through the Kaggle API. In addition, Kaggle notebooks are attached to each subset to enable convenient data browsing on Kaggle's cloud computing platform. This approach enables cost-effective distribution of scientific data that adheres to FAIR principles [156], as further detailed in Appendix A.1.

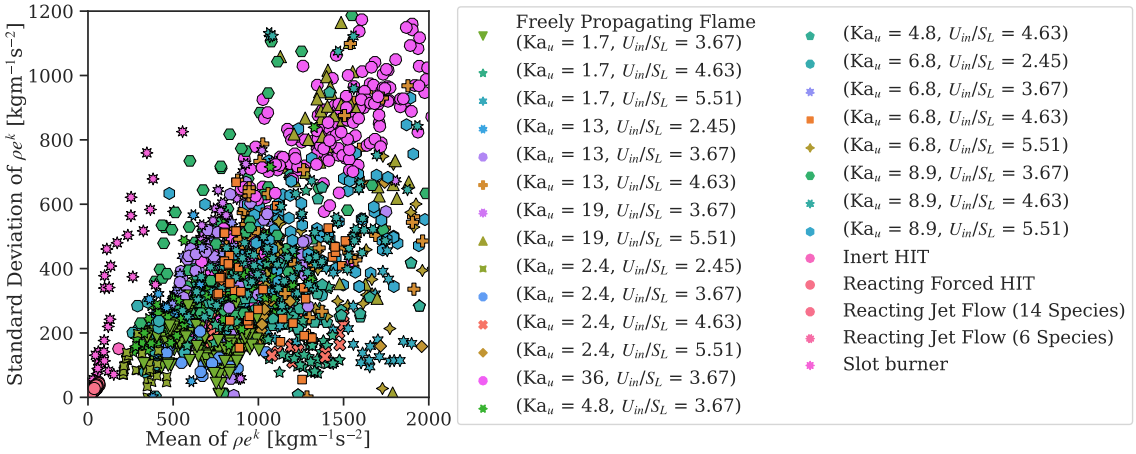


Figure 3.2: Statistics of the specific kinetic energy ρe^k of each 128^3 sub-volume in the Momentum128 3D SR dataset. Note that the freely propagating flame configuration [155] has 22 permutations of varying unburned Karlovitz numbers Ka_u and inlet velocity U_{in} (normalized by laminar flame speed S_L).

Consistent Format Data, generated from different numerical solvers, initially exists in a range of formats (`.vtk`, `.vtu`, `.tec`, and `.dat`) that are not readily formatted for training ML models. Thus, all flowfield data are processed into a consistent format – little-endian single-precision binaries that can be read via `np.fromfile`/`np.memmap`. The choice of this data format enables high I/O speed in loading arrays. We provide `.json` files that store additional information on configurations, chemical mechanisms and transport properties. See Appendix A.2 for more data format details.

Licensing and Ethics All data is generated by the present authors and licensed via CC BY-NC-SA 4.0. Other than the contributors' names and institutions, no personal-identifiable information is published in this data. No offensive content is published with this flow physics dataset.

3.2.2 Momentum128 3D SR Dataset

BLASTNet 2.0 is further processed for training due to constraints in (i) memory and (ii) grid properties. Currently, the single largest sample (92 GB) in BLASTNet 2.0 contains 1.3B voxels and 15 channels, which cannot fit into typical GPU memory. In

addition, the spatial grid is stretched depending on the resolution requirements of the flow domain. As shown in Figure 3.1, we circumvent these two issues by sampling 128^3 sub-volumes of density ρ and velocity u_i from the uniform-grid regions from all BLASTNet data. This results in 12,750 sub-volume samples (427 GB). We choose this sub-volume size to enable 32× SR (the resulting feature sub-volume is 4^3 which is larger than a kernel size of 3), while maintaining a low memory footprint. In order to develop a compressible turbulence benchmark dataset that can be easily downloaded, we select 2,000 sub-volumes to form a 67 GB dataset that can fit into a single Kaggle repository. To ensure that these 2,000 samples are representative of the different flows encountered in each configuration, we:

1. Extract mean, variance, skewness, and kurtosis (statistical moments for characterizing turbulence [34]) from the three velocity components of 12,750 sub-volumes.

2. Apply k-means clustering with the elbow method (using the statistical moments as features) to partition the sub-volumes in 18 clusters.

3. Select 2,000 samples while ensuring that the proportion of clusters are well-balanced.

The resulting sub-volumes form the labels of BLASTNet Momentum128 3D SR dataset. Figure 3.2 demonstrate the mean and standard deviation of the volume-specific kinetic energy ρe^k, which we use to characterize all channel variables (see Equation (2.5)). Each distinct marker represents a different simulation configuration. Since flows from the same configuration possess similar statistics, the different configurations from BLASTNet 2.0 can result in a dataset with a variety of flow conditions. We Favre-filter and downsample the labels by 8, 16, and 32× (LES is typically an order of magnitude coarser than DNS [34, 157]) to generate inputs for turbulent SR. To summarize our SR dataset, at a given voxel of sample i, the channels of each label correspond to $\Upsilon_i = [\rho,u_1,u_2,u_3]^\intercal$, while the feature channels consist of $\chi_i = [\overline{\rho},\widetilde{u}_1,\widetilde{u}_2,\widetilde{u}_3]^\intercal$. For the purpose of the present benchmark study, we further split the 2,000 sub-volumes as follows:

Train, Validation, and Baseline Test Sets 80:10:10 split via random selection with a uniform distribution. The training set contains 1,382 samples, and both validation and baseline test sets contain 173 samples each.

Parametric Variation Set A 144-sample subset for model evaluation from an unseen parametric variation configuration with approximately 15% higher mean velocities and velocity fluctuations than the train, validation, and baseline test sets.

Forced HIT Set A 128-sample subset for model evaluation from an unseen flow type (forced HIT) with 30-fold higher pressure and 34-fold lower velocity fluctuations.

3.3 Benchmark Configuration

3.3.1 ML Models and Methods

As shown in Figure 3.1, three well-studied 2D ResNet-based [114] SR models are modified from their original repositories for 3D SR: (i) Residual-in-Residual Dense Block (RRDB) [146], (ii) Enhanced Deep Residual SR (EDSR) [147], and (iii) Residual Channel Attention Networks (RCAN) [148]. As discussed in Section 2.2.4, convolution networks possess inductive biases that are suitable for problems involving spatial grids, such as in flow physics [158, 159]. We choose to study these models due to their differences in architecture paradigms. Specifically, RRDB employs residual layers within residual layers; EDSR features an expanded network width; RCAN utilizes long skip connections and channel attention mechanisms. In addition, we consider two additional scientific ML approaches: (i) a Conv-FNO model [149], modified for SR, and (ii) an RRDB model regularized with a physics-based loss (see Section 2.2.7). In the Conv-FNO model, outputs of an FNO layer and a convolutional layer were added to the outputs of each residual block in the EDSR. This modification enables us to examine combining FNO layers with convolution blocks that have been demonstrated to perform well in SR applications [149].

The physics-based loss used in this chapter can be expressed as:

$$J_{phys} = (1-\Lambda)J_{MSE} + \Lambda J_{grad}\,, \quad \text{where} \tag{3.1a}$$

$$J_{grad} = \frac{\Delta^2}{3N_{vox}N_c}\sum_{i=1}^{N_{vox}}\sum_{j=1}^{N_c}\sum_{k=1}^{3}\left[\left(\frac{\partial\Upsilon}{\partial x_k}\right)_{ij} - \left(\frac{\partial\hat{\Upsilon}}{\partial x_k}\right)_{ij}\right]^2, \tag{3.1b}$$

for a given target $\Upsilon \in \boldsymbol{\Upsilon}$. N_{vox} is the number of voxels and N_c the number of channel variables of the output $\hat{\Upsilon} \in \hat{\boldsymbol{\Upsilon}}$. Δ is the distance between each voxel. The gradient terms are evaluated via a second-order central differencing scheme that is optimized for GPU calculations. This is done on both super-resolved and ground truth fields before inputting the gradient terms into the MSE function. This gradient term enables the ML models to implicitly learn transport phenomena that arise in flow physics PDEs. For example, advection in the mass conservation equation can be expressed as:

$$\frac{\partial \rho u_j}{\partial x_j} = \rho\frac{\partial u_j}{\partial x_j} + u_j\frac{\partial \rho}{\partial x_j}\,, \tag{3.2}$$

which requires the gradients of all channel variables to be predicted correctly. These arguments can also be applicable to the advection of momentum, which is the transport term responsible for turbulent phenomena [34]. We employ the weighting factor $\Lambda = 0.99$, which was determined from a hyperparameter search on RRDB models with 2.7M number of parameters.

To investigate the scaling behavior of the model architectures, we vary the number of parameters by changing the network depth and width of RRDB, EDSR, and RCAN models. The hyperparameters within the Conv-FNO model was first determined by comparing two approaches with the same number of parameters (3.0M): (i) one with large number of Fourier modes, and (ii) one with deep and wide Conv-FNO blocks. Since the approach number of modes with deep and wide Conv-FNO blocks demonstrated better validation MSE, another hyperparameter search was performed to determine the optimal number of Fourier modes until the GPU memory was fully consumed at five Fourier modes. We scale the Conv-FNO in Section 3.4 by increasing

the FNO channel size, since this approach led to good scaling behavior, especially when compared to increasing the number of Fourier modes. All other hyperparameters are maintained from their original studies, with all models initialized via He et al. [112]. To summarize, the architecture settings for RRDB, EDSR, RCAN, and Conv-FNO are shown in Table 3.1. In this table, we list the number of residual blocks, growth factor in RRDB blocks, the channel width, kernel size, number of FNO modes, RCAN residual groups, and residual scaling factors – which are hyperparameters for the model architectures.

Table 3.1: Hyperparameters of RRDB, EDSR, and RCAN models investigated in this work.

RRDB Parameters	0.6M	0.9 M	1.4M	2.7M	4.9M	11.4M	17.8M	50.2M
Residual (RRDB) Blocks	1	1	1	1	2	5	8	23
First Channel Size	4	16	32	64	64	64	64	64
Kernel Size	3	3	3	3	3	3	3	3
RRDB Growth Factor	32	32	32	32	32	32	32	32
Residual Scaling	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
EDSR Parameters	0.5M	1.0M	1.4M	2.8M	5.1M	11.1M	17.8M	34.6M
Residual Blocks	32	32	32	32	32	32	32	32
Channel Size	14	20	24	34	46	68	86	120
Kernel Size	3	3	3	3	3	3	3	3
Residual Scaling	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
RCAN Parameters	0.5M	0.9M	1.5M	2.7M	5.1M	11.8M	16.4M	48.3 M
Residual Blocks	1	1	1	1	1	10	20	20
Channel Size	26	34	44	60	64	64	64	64
Residual Groups	1	1	1	1	1	2	3	10
Kernel Size	3	3	3	3	3	3	3	3
Residual Scaling	1	1	1	1	1	1	1	1
Conv-FNO Parameters	–	0.6M	1.8M	2.6M	5.2M	9.4M	20.6M	32.9M
FNO modes	–	2	2	2	2	2	2	2
FNO Channel Size	–	14	20	24	34	46	68	86
Conv-FNO Blocks	–	32	32	32	32	32	32	32
Convolutional Kernel Size	–	3	3	3	3	3	3	3
Residual Scaling	–	0.1	0.1	0.1	0.1	0.1	0.1	0.1

Similar to other turbulent SR studies [44, 143], all models are trained with MSE loss, unless otherwise stated. For evaluation, we select models with the best MSE

after training for 1,500 epochs with a batch size of 64 across 16 Nvidia V100 GPUs. With the Adam optimizer [110], learning rate is initialized at `1e-4` and halved every 300 epochs. Both the number of training iterations and learning scheduling are chosen to match other SR studies [146–148] and are found to be sufficient for the SR predictions, as will be shown in Section 3.4. All other hyperparameters are maintained from their original studies, with He initialization [112] used on all initial model weights. Data augmentation is performed via variants of random rotation and flip transformations – which we modified to ensure augmented data remains consistent with mass conservation. Specifically, this is necessary for maintaining the reflective and rotational invariance of the divergence of momentum, after transformation. The steps to ensure this are summarized in Figure 3.3, which demonstrates how flip and rotation operations can still result in continuity-consistent transformations on a notional 2D flowfield. These operations have been extended for 3D random flip and rotation, which we employed during training.

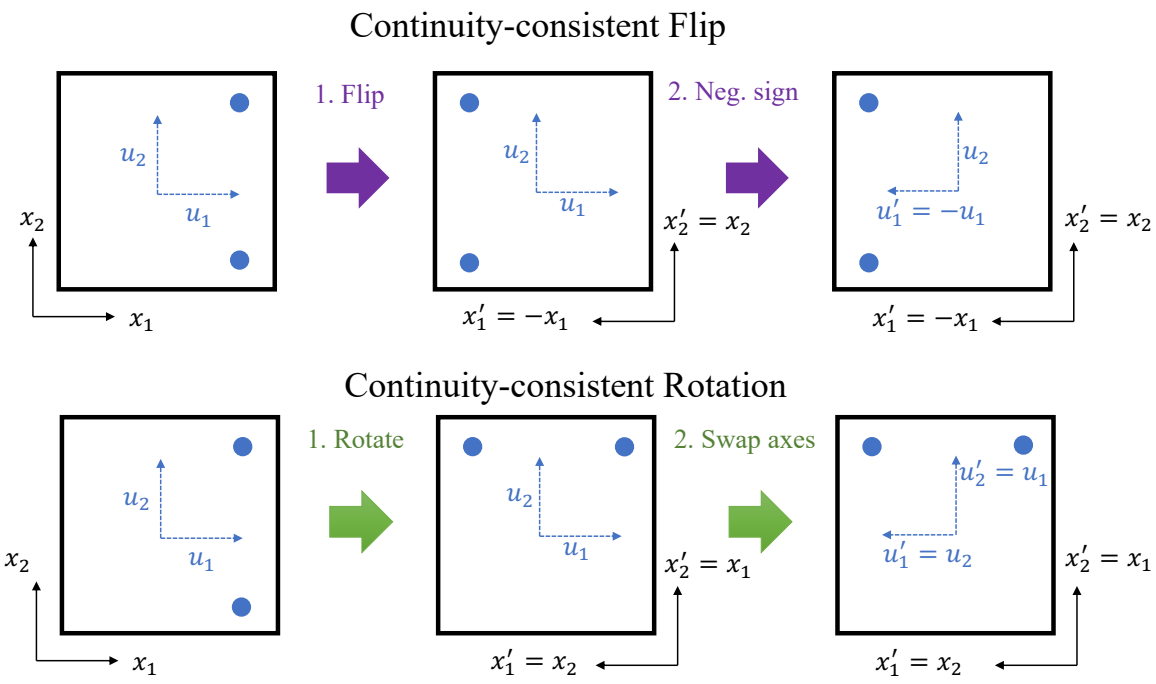


Figure 3.3: Continuity-consistent augmentation on a 2D image that preserves reflective and rotational invariance of the $\partial \rho u_j / \partial x_j$.

Training is performed with automatic mixed-precision from `Lightning 1.6.5` [160]. Prior to training, data is normalized with means and standard deviations of density and velocity extracted from the train set. During evaluation, this normalization resulted in poor accuracy for the Forced HIT set, due to the significantly different magnitudes of density and velocity. However, Section 3.4 will show that good performance can be achieved when normalization is performed with the mean and standard deviation of each distinct evaluation set. Thus, all evaluation sets are normalized with their own mean and standard deviation, prior to testing. All 49 model variations are trained with three different seeds, resulting in a total computational cost of approximately 15,000 GPU-hours.

3.3.2 Metrics

We compare the performance of each model by examining local and global quantities of each sample. For the local quantities, we employ Metric = {SSIM, NRMSE}:

$$\text{Metric}_{\rho,u_i} \equiv \frac{1}{4}\left[\text{Metric}(\rho,\hat{\rho}) + \sum_{i=1}^{3}\text{Metric}(u_i,\hat{u}_i)\right], \tag{3.3a}$$

$$\text{Metric}_{sgs} \equiv \frac{1}{3}\sum_{i=1}^{3}\text{Metric}\left(\frac{\partial\tau_{ij}^{sgs}}{\partial x_j},\frac{\partial\hat{\tau}_{ij}^{sgs}}{\partial x_j}\right), \tag{3.3b}$$

with $\hat{\cdot}$ denoting an arbitrary predicted quantity. Metric_{ρ,u_i} evaluates each channel of the predictions via macro-averaging. To measure the suitability of SR for turbulence modeling via *a priori* analysis, we measure Metric_{sgs}, which evaluates the predicted divergence of the SGS stress.

We evaluate our models with the 3D version of the structural similarity index measure (SSIM) [161]. SSIM is calculated by passing a sliding window of size 9×9×9 (similar to the original SSIM paper [161]) across two arbitrary quantities ϕ and ψ (at a given voxel and channel), and evaluating their statistical quantities. Specifically, SSIM is defined by:

$$\text{SSIM}(\phi,\psi) = \frac{1}{N_w}\sum_{i=1}^{N_w}\left(\frac{2\overline{\phi}\,\overline{\psi}+\mathcal{C}_1^2}{\overline{\phi}^2+\overline{\psi}^2+\mathcal{C}_1^2}\cdot\frac{2s_{\phi\psi}+\mathcal{C}_2^2}{s_\phi^2+s_\psi^2+\mathcal{C}_2^2}\right)_i, \tag{3.4}$$

with variance $s^2_{\{\phi,\psi\}}$, and covariance $s_{\phi\psi}$ for N_w number of sliding windows. In computer vision applications with RGB images, $\mathcal{C}_1 = 0.01$ and $\mathcal{C}_2 = 0.03$ are typically used to ensure numerical stability [161]. However, we found these values insufficient for numerical stability in this work. Hence, we employed $\mathcal{C}_1 = 0.1$ and $\mathcal{C}_2 = 0.3$. SSIM is a common image metric, but has also become a popular ML metric for evaluating flow simulations due to its employment of mean, variance, and covariance quantities – suited for evaluating the statistical nature of turbulence [34, 162, 163]. In addition, this metric is intuitive for both readers familiar and unfamiliar with turbulent flows – SSIM of 0 denotes dissimilar fields while SSIM of 1 denotes highly similar fields.

We also employ a conventional definition of normalized root MSE (NRMSE) for evaluating the ML models:

$$\mathrm{NRMSE}(\phi, \psi) = \frac{\sum_{i=1}^{N_{vox}} (\phi_i - \psi_i)^2}{\sum_{i=1}^{N_{vox}} \phi_i^2}\,, \tag{3.5}$$

for evaluating global physical properties predicted by the ML model by considering the $\mathrm{NRMSE}_{\{\langle e\rangle^k,\varepsilon\}}$ of turbulent dissipation rate ε (rate of conversion of turbulent kinetic energy to heat) and volume-averaged kinetic energy $\langle e\rangle^k$ (momentum component in energy conservation of a fixed control volume):

$$\langle e\rangle^k = \frac{1}{V}\int_V \rho e^k \, dV\,, \tag{3.6a}$$

$$\varepsilon = \frac{1}{V}\int_V \frac{\tau_{ij}}{\rho}\frac{\partial u_i'}{\partial x_j}\, dV\,, \tag{3.6b}$$

with sample volume V and velocity fluctuation u_i'.

In this chapter, theoretical floating point operations (FLOPs) for the ML models is estimated via `THOPs` (`https://github.com/Lyken17/pytorch-OpCounter`) which has been used in other studies [164, 165]. Einstein summation operations in FNO layers were evaluated through modifying `THOPS` with `np.einsum_path`, while Fourier and inverse Fourier transforms are estimated as $5N_{points}\log N_{points}$ [33].

Table 3.2: Comparison of SSIM of five models at three SR ratios, with tricubic interpolation. Mean and standard deviation from three seeds are reported here. **Bold** term represents best mean.

Models	Baseline Test Set		Param. Variation Set		Forced HIT Set		Size	Cost
	↑SSIM$_{\rho,u_i}$	↑SSIM$_{sgs}$	↑SSIM$_{\rho,u_i}$	↑SSIM$_{sgs}$	↑SSIM$_{\rho,u_i}$	↑SSIM$_{sgs}$	N_p	↓GFLOPs
Tricubic 8×	0.820	0.431	0.800	0.418	0.951	0.711	–	**23**
RRDB 8×	0.907±0.003	0.715±0.004	0.898±0.003	0.755±0.002	0.997±0.000	0.891±0.003	50.2M	1430
(+ Grad. Loss)	**0.936±0.003**	**0.802±0.003**	**0.929±0.001**	**0.825±0.001**	0.998±0.000	**0.944±0.005**		
EDSR 8×	0.928±0.004	0.748±0.012	0.916±0.005	0.775±0.010	0.999±0.000	0.937±0.005	34.6M	2122
RCAN 8×	0.928±0.000	0.753±0.002	0.916±0.001	0.778±0.001	**0.999±0.000**	0.941±0.003	16.4M	671
Conv-FNO 8×	0.846±0.016	0.566±0.019	0.845±0.011	0.614±0.015	0.993±0.001	0.845±0.008	33.0M	1276
Tricubic 16×	0.652	0.175	0.620	0.173	0.876	0.432	–	**23**
RRDB 16×	0.724±0.001	0.506±0.004	0.700±0.001	0.512±0.002	0.971±0.000	0.805±0.003	50.3M	1074
(+ Grad. Loss)	**0.739±0.008**	**0.554±0.001**	**0.719±0.004**	**0.556±0.002**	**0.973±0.000**	**0.816±0.001**		
EDSR 16×	0.716±0.005	0.477±0.018	0.693±0.005	0.481±0.019	0.969±0.001	0.783±0.008	37.8M	1944
RCAN 16×	0.672±0.039	0.408±0.066	0.665±0.024	0.415±0.058	0.961±0.009	0.737±0.050	17.3M	573
Conv-FNO 16×	0.629±0.020	0.343±0.027	0.640±0.013	0.355±0.022	0.951±0.006	0.690±0.022	34.6M	1068
Tricubic 32×	**0.508**	0.060	0.476	0.087	0.758	0.156	–	**23**
RRDB 32×	0.503±0.001	**0.194±0.005**	0.482±0.000	0.186±0.006	0.845±0.001	0.494±0.011	50.4M	1030
(+ Grad. Loss)	0.505±0.001	0.184±0.009	**0.483±0.001**	**0.188±0.002**	**0.850±0.000**	**0.516±0.012**		
EDSR 32×	0.502±0.004	0.173±0.006	0.481±0.002	0.187±0.004	0.845±0.001	0.463±0.005	40.9M	1921
RCAN 32×	0.473±0.006	0.168±0.007	0.469±0.002	0.185±0.005	0.837±0.003	0.448±0.012	18.2M	561
Conv-FNO 32×	0.476±0.004	0.155±0.012	0.470±0.001	0.178±0.003	0.842±0.002	0.435±0.013	36.2M	1023

3.4 Results

We summarize SSIMs of RRDB, EDSR, RCAN, and Conv-FNO in Table 3.2, along with model parameters N_p and inferencing cost (in FLOPs for a batch size of 1). The 8× SR models shown here possess the best SSIMs across different sizes for a given model approach (see Table 3.1). Models, with the same network depth and width, are then initialized and trained for 16 and 32× SR. For 8 and 16× SR, RRDB (with gradient loss) performs the best across most of the metrics and evaluation sets, with RCAN demonstrating the highest SSIM$_{\rho,u_i}$ at 8× SR. At 32× SR, all shown models exhibit lower SSIM$_{\rho,u_i}$ than tricubic interpolation in the baseline test set, indicating that SR is difficult to learn at high ratios. However, all models exhibit higher SSIM$_{sgs}$ than tricubic interpolation for all SR ratios. This indicates that SR models may still be useful for turbulence modeling at high SR ratios.

Figure 3.4, demonstrates that model predictions of specific kinetic energy ρe^k (a physical quantity that combines predictions of all four channels) from all models presented in Table 3.2 increasingly lose fine turbulent structures as SR ratio increase.

Nevertheless, when compared to tricubic interpolation, the SR models can still recover the magnitudes of the energy at these SR ratios. Further results from the NRMSE metrics evaluated on the 8× SR models are shown in Table 3.3, which also demonstrates that all ML models significantly outperform tricubic interpolation on baseline test and forced HIT sets. Here, gradient loss RRDB performs best in most of the metrics. However, EDSR outperforms with $\text{NRMSE}_{\langle e \rangle^k}$, as the gradient loss only offers minor improvements to $\langle e \rangle^k$.

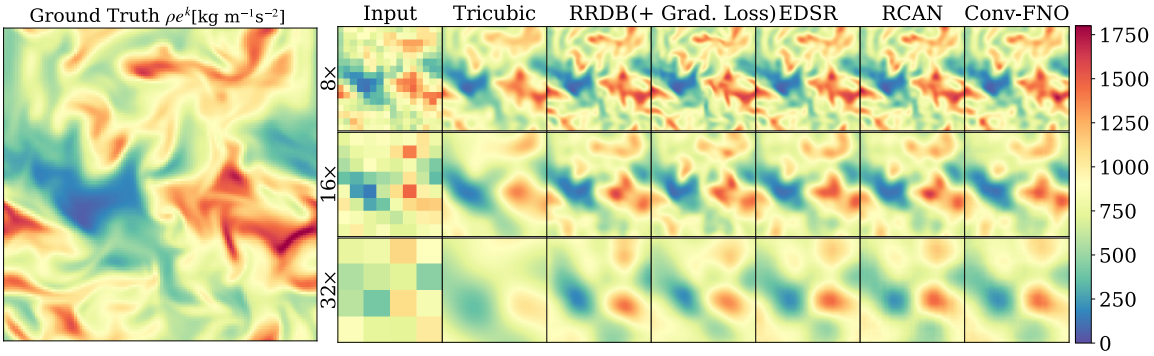


Figure 3.4: Specific kinetic energy ρe^k prediction of one sample from the parametric variation set with models from Table 3.2.

Table 3.3: Comparison of NRMSE for five models at 8× SR ratio, with tricubic interpolation. Mean and standard deviation from three seeds are reported here. **Bold** term represents best mean.

Models	Baseline Test Set				Forced HIT Set			
	↓NRMSE_{ρ,u_i} ($\times 10^{-2}$)	↓NRMSE_{sgs} ($\times 10^{-1}$)	↓$\text{NRMSE}_{\langle e \rangle^k}$ ($\times 10^{-4}$)	↓NRMSE_{ϵ} ($\times 10^{-1}$)	↓NRMSE_{ρ,u_i} ($\times 10^{-3}$)	↓NRMSE_{sgs} ($\times 10^{-2}$)	↓$\text{NRMSE}_{\langle e \rangle^k}$ ($\times 10^{-6}$)	↓NRMSE_{ϵ} ($\times 10^{-4}$)
Tricubic 8×	5.09	7.51	8.89	4.33	8.82	31.12	734.55	451.68
RRDB 8×	0.92±0.01	2.46±0.04	0.39±0.10	1.16±0.00	0.19±0.01	2.15±0.08	39.83±32.51	0.74±0.29
(+ Grad. Loss)	**0.60±0.00**	**1.41±0.01**	0.41±0.17	**0.54±0.01**	0.13±0.01	**1.23±0.17**	33.77±21.44	0.55±0.17
EDSR 8×	0.86±0.04	2.30±0.15	**0.29±0.06**	1.10±0.06	0.10±0.01	1.67±0.25	**0.60±0.24**	**0.21±0.03**
RCAN 8×	0.86±0.00	2.31±0.01	0.32±0.01	1.14±0.00	**0.09±0.00**	1.39±0.11	0.62±0.05	0.23±0.02
ConvFNO 8×	1.46±0.07	4.42±0.23	0.74±0.19	1.64±0.05	0.56±0.11	6.94±0.75	163.50±191.46	3.66±2.22

Scaling behavior of RRDB is shown in Figure 3.5, which compares ground truth and input values of ρe^k (shown in the first column) with 8× SR predictions from tricubic interpolation and variations of RRDB models. For the model predictions, the first row visualizes the specific kinetic energy $\hat{\rho}\hat{e}^k$, while the second row shows the error $|\epsilon_{\rho e^k}| = |\hat{\rho}\hat{e}^k - \rho e^k|$ normalized by ρe^k_{max}. Our discussion is focused on

the predictions in the cyan box. At $N_p = 0.6$M, RRDB is unable to reconstruct ρe^k accurately. RRDB's prediction is more accurate than tricubic interpolation at $N_p = 4.9$M, but spurious structures that originate from the coarse grid can be seen. For $N_p = 50.2$M, the model is sufficiently expressive for eliminating the spurious structures from the flow. The addition of the gradient loss term is shown to reduce prediction errors from RRDB 50.2M. This trend in improvement is also visible in the bottom row, which shows the mean divergence of SGS stresses (Equation (2.10)).

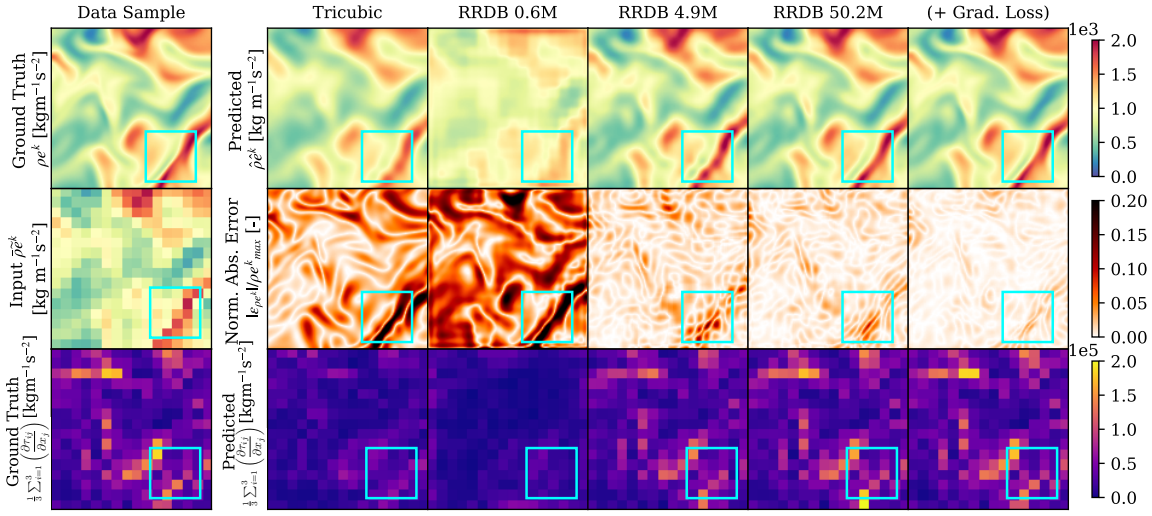


Figure 3.5: Predictions from various RRDB models, showing gradual improvement in the cyan box.

Scaling behavior of RRDB (with and without gradient loss), EDSR, RCAN, and Conv-FNO models are examined in Figure 3.6. SSIM_{sgs} scales differently compared to SSIM_{ρ,u_i}, indicating the importance of evaluating derived physical quantities from model predictions in flow physics applications. For both SSIMs across all evaluation sets, RCAN models demonstrate better performance than EDSR and vanilla RRDB models for $N_p < 17$M, but performance deteriorates after this model size. The gradient loss term improves RRDB predictions for all model parameters explored, resulting in SSIM_{sgs} that exceeds RCAN after $N_p = 1.4$M for the baseline test and Parametric Variation sets. Thus, this loss term is shown to benefit moderately sized models ($N_p = 50.2$M) and data (67 GB), which is in contrast to the notion that physics-based losses are mostly helpful for small models and datasets [158]. Conv-FNO is

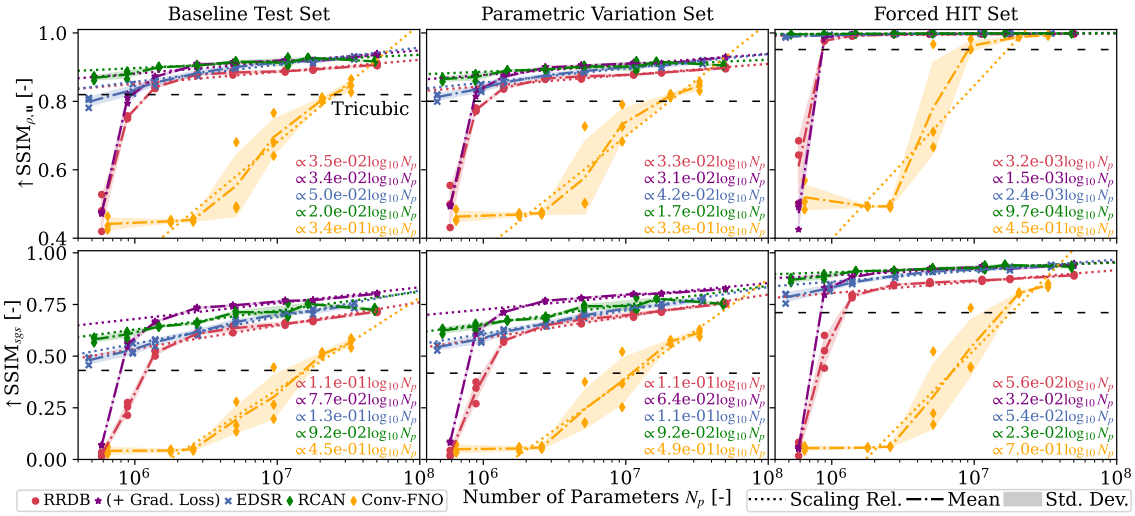


Figure 3.6: Scaling behavior of RRDB (with and without gradient-based loss), EDSR, RCAN and Conv-FNO. RRDB, EDSR and Conv-FNO models continue to scale at large model sizes.

seen to outperform the baseline tricubic prediction after approximately 20M parameters. FNO layers are memory-intensive due to high number of dimensions found in the spectral convolution weights (six in total: one for batches, two for channels, and three for Fourier modes). This memory-intensive nature has been acknowledged by FNO's original developers, with attempts to address this remaining an active research pursuit [166].

For all models, both SSIMs are found to scale with $\log_{10} N_p$. All ResNet-based models share similar slopes in the scaling relationship between SSIM_{sgs} and $\log_{10} N_p$ in the test and Parametric Variation set. However, these slopes can differ when evaluated on another flow configuration. This is seen with the idealized flows in the Forced HIT set, where higher SSIMs from all predictions and baseline are observed.

Figure 3.7 shows the relationship between SSIM_{sgs} and inference cost (in FLOPs) for the five model approaches. SSIM_{sgs} for EDSR, RCAN, and RRDB (with gradient loss) models scales with cost similarly, after approximately 100 GFLOPs. A steeper scaling relationship is observed for both Conv-FNO and vanilla RRDB. Vanilla RRDB models also do not demonstrate a strong linear relationship with $\log_{10}$ GFLOPs when

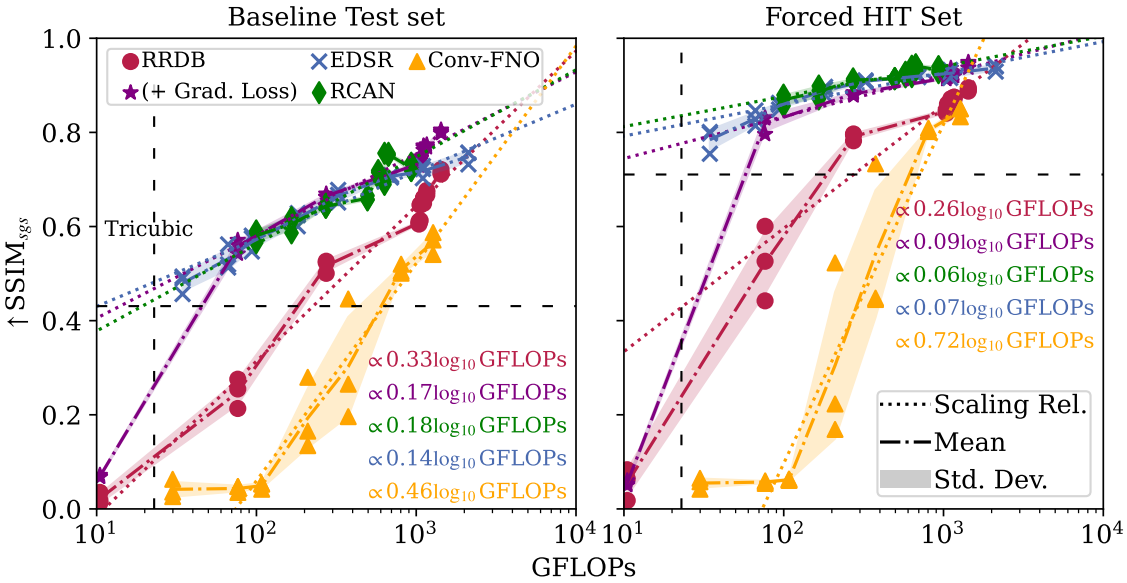


Figure 3.7: Scaling behavior with cost.

tested on the Forced HIT set.

3.5 Summary

In this work, we released BLASTNet 2.0, a public 3D compressible turbulent reacting and non-reacting flow dataset, to directly address gaps in data availability within multi-physics flow applications. From this data, we extracted the Momentum128 3D SR dataset, which we employed for benchmarking 3D SR models at 8, 16, and 32× SR. SR models are shown to score well in SSIM-based metrics and capture fine turbulent structures at 8× SR. For the higher SR ratios, these fine structures cannot be captured, but the SR models can still recover the magnitude of large flow structures. Through our scaling analysis, we demonstrated that benefits from a gradient-based physics-based loss persist with model scale up to approximately 50M model parameters – providing empirical evidence that disagrees with the postulated notion that physics-based methods are useful mostly in small model scenarios [158]. We observed that model performance scales with the logarithm of model parameters, and that

the scaling relationship between SSIM_{sgs} and inference cost are similar for RRDB (with gradient loss), EDSR, and RCAN. We also demonstrated that the choice of model architecture can matter significantly, especially when developing small models for real-time scientific computing applications, and that physics-based losses can improve some metrics of poorly performing architectures. With this chapter, we demonstrated that BLASTNet 2.0 can provide a rich resource for training and evaluating ML models for scientific and engineering turbulent flows.

Chapter 4

Subgrid-scale Closure with Interpretable Machine Learning*

4.1 Introduction

In the previous chapter, we show that deep learning models trained on large datasets can be highly accurate and flexible approaches for closure modeling via turbulent SR. However, data for complex multi-physics flows, such as in real-fluids found within propulsion systems, can be challenging to obtain, when compared to the previously curated gas-phase configurations. Specifically, transcritical flows can be found within high pressure rocket engines that operate under conditions that exceed the thermodynamic critical limits of both fuel and oxidizer, and can give rise to complex behaviors that pose challenges for numerical modeling and simulations [167], as highlighted in Section 1.2. In this chapter, we examine opportunities provided by alternative ML methods in developing closure models on a small transcritical flow dataset.

To this end, we perform DNS calculations of inert and reacting LOX/GCH_4 non-premixed transcritical mixtures in the presence of decaying turbulence, in order to evaluate algebraic and ML models for predicting unclosed SGS terms for high pressure

*This chapter contains previously published work from Chung et al. [7], with minor modifications. W.T. Chung planned, performed, and analyzed experiments, and performed simulations. A.A. Mishra assisted with planning experiments.

propulsion applications. Thus, the objectives of this chapter can be summarized as follows:

- To identify and quantify limitations of conventional algebraic SGS stresses in transcritical flows.
- To utilize alternatives to deep learning, namely the random forest regressor and the sparse symbolic regression, in constructing data-informed models for SGS stresses.
- To apply these ML methods towards modeling additional SGS terms that can arise from real-fluid effects.

The mathematical models for simulating the turbulent transcritical flows are presented in Section 4.2. Details regarding the DNS configuration are discussed in Section 4.3. Section 4.4 describes ML methods employed in the present work. Results from this *a priori* study are discussed in Section 4.5, before offering concluding remarks in Section 4.6.

4.2 Mathematical Models

4.2.1 Governing Equations

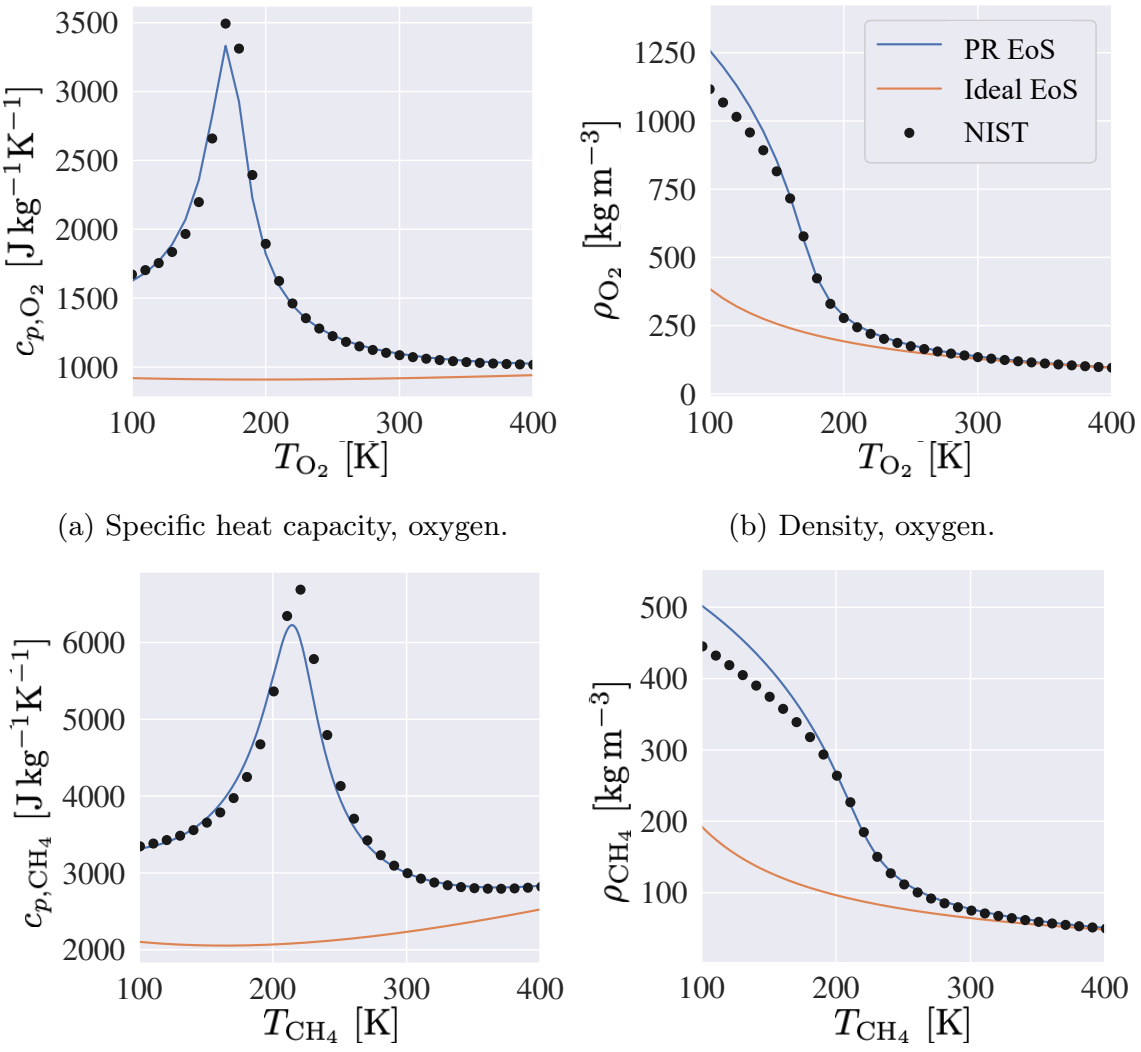
The governing equations that are solved in the present chapter to generate the DNS data are the conservation equations for mass, momentum, energy, and chemical species (see Equation (2.1)). For the flow-conditions considered in this study, a filter width of 16Δ is equivalent to the integral length-scale. Since LES should resolve the inertial subrange, we employ a maximum filter width of 8Δ to obtain filtered DNS. Through *a priori* analysis, SGS quantities can then be extracted from the filtered DNS and compared with approximations through algebraic and ML-based closure models.

Here, the real-fluid thermodynamic states are modeled via the PR EoS. Since O_2 and CH_4 mixtures are a miscible system, where the effects of phase separation are

not encountered due to the similarity of the critical states and molecular properties, this transcritical configuration can be represented by a cubic EoS [168]. Figure 4.1 compares PR and ideal EoS for CH_4 and O_2. At the initial conditions of 120 K and 300 K for O_2 and CH_4, specific heat capacity evaluated from the PR EoS is in excellent agreement with NIST data. However, it can be seen that the PR EoS overpredicts the oxidizer density but provides accurate results for the fuel. Since this chapter is primarily focusing on the development and assessment of a data-driven modeling framework for the constructing SGS closures, we believe that this discrepancy is acceptable for the present study.

In this chapter, the two-step five-species CH4-BFER mechanism [170] is employed, which was applied to investigate a supercritical gas turbine combustor at 20 MPa [171]. In DNS of trans- and supercritical combustion, reduced chemical mechanisms [172, 173] have been employed to circumvent large computational costs incurred by solving non-ideal state equations. Takahashi's high-pressure correction [174] is used to evaluate the binary diffusion coefficients. Since only two species are used in the inert simulations, the binary diffusion coefficients are exact. Thermal conductivity and dynamic viscosity are evaluated using Chung's method with high-pressure correction [175]. For multi-species mixtures in the reacting cases, Chung's pressure correction is known to produce oscillations [102, 176], especially for dynamic viscosity. Hence, transport properties of the mixture are evaluated through mole-fraction-averaging, after employing Chung's method on each individual species. A similar approach has been applied in prior studies [56, 177].

Simulations are performed by employing an unstructured compressible finite-volume solver [7, 102, 178]. A central scheme, which is 4th-order accurate on uniform meshes, is used along with a 2nd-order essentially non-oscillatory (ENO) scheme. The ENO scheme is activated only in regions of high local density variations, using a threshold-based sensor to describe sharp interfaces present in transcritical flows. Due to the density gradients present at trans- and supercritical conditions, an entropy-stable flux correction technique [102] is used to dampen non-linear instabilities in the numerical scheme. The double-flux method by Ma et al. [102] is used with a dynamic sensor to eliminate spurious pressure oscillations. A Strang-splitting scheme



(a) Specific heat capacity, oxygen.

(b) Density, oxygen.

(c) Specific heat capacity, methane.

(d) Density, methane.

Figure 4.1: Comparison of Peng-Robinson (PR) and ideal equations-of-states (EoS) for (a,b) O_2 and (c,d) CH_4 with NIST [169] data at $p = 10$ MPa.

is employed for time-advancement, combining a strong stability preserving 3rd-order Runge-Kutta (SSP-RK3) scheme for integrating the non-stiff operators with a semi-implicit scheme [179] for advancing the chemical source terms.

4.2.2 Additional Closure Terms from Real-fluid Effects

We investigate the effects of additional non-linearities from the real-fluid EoS, by employing an analysis similar to Huo and Yang [180] that they applied to model SGS density. Equation (2.7) can be reexpressed with a compressibility factor ζ:

$$p = \rho R T \zeta \tag{4.1}$$

By rearranging and Favre-filtering, we obtain:

$$\widetilde{T} = \widetilde{p \cdot (\rho R \zeta)^{-1}} \tag{4.2}$$

However in the present LES solver, the Favre-filtered temperature is obtained by inputting filtered quantities into the real-fluid EoS:

$$\widetilde{T} = \overline{p} \cdot [\overline{\rho}\overline{R}\bar{\zeta}(\overline{\rho}, \overline{p}, \widetilde{Y})]^{-1} + \left[\widetilde{p \cdot (\rho R \zeta)^{-1}} - \overline{p} \cdot (\overline{\rho}\overline{R}\bar{\zeta})^{-1}\right] \tag{4.3a}$$

$$\widetilde{T} = \widetilde{T}_{LES}(\overline{\rho}, \overline{p}, \widetilde{Y}) + T^{sgs} \tag{4.3b}$$

which gives rise to an SGS temperature T^{sgs}, *i.e.*, the second term on the right-hand-side.

T^{sgs} is typically neglected in ideal-gas configurations. This is often an acceptable assumption, as shown by the ideal EoS case in Figure 4.2. In the transcritical inert case, $|T^{sgs}|/\widetilde{T}$ of approximately 0.05 is observed, which is similar with observations from another study [55]. However, T^{sgs} becomes non-negligible for the transcritical reacting cases, where $|T^{sgs}|/\widetilde{T}$ exceeds values of 0.1 in the reacting regions, where multi-species compositions are present, and regions with high density gradient. Non-negligible SGS EoS terms are also reported by other studies [180, 181]. This added significance of T^{sgs} arises from applying the filtering operation on density and multi-species mass fractions, and then feeding the filtered quantities into a highly non-linear equation.

Amplified non-linearities in transcritical reacting flow present an additional source

of uncertainty in SGS modeling. To investigate this, we employ conventional algebraic and novel data-driven methods for predicting the SGS fluxes from the LES momentum equation (Equation (2.9b)). Two algebraic SGS models, namely the Vreman and the gradient model (as described in Section 2.1.2), as well as random forest regressors are evaluated. Additionally, we demonstrate the employment of random forest feature importance scores for assisting the discovery of algebraic SGS stress models by sparse symbolic regression. Since algebraic models for SGS temperature (Equation (4.3)) have not been developed, we then evaluate the ability of an interpretable ML algorithm in modeling T^{sgs}.

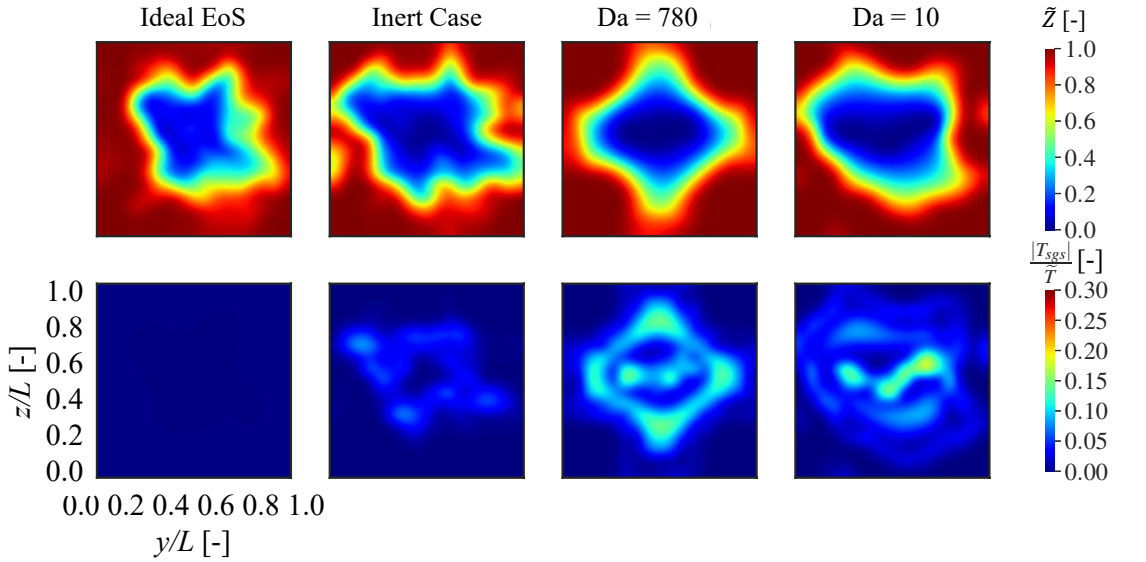


Figure 4.2: Comparisons of filtered mixture fraction $\widetilde{Z}$ and magnitude of normalized SGS temperature $|T^{sgs}|/\widetilde{T}$ between an ideal-gas case and three transcritical cases. A filter width of $\overline{\Delta} = 8\Delta$ is employed.

4.3 DNS Configuration

Inert and reacting DNS are performed on a three-dimensional cubic domain, with length L, a mixture of LOX/GCH$_4$ shown in Figure 4.3. In this setup, a spherical liquid O_2 core, with a radius $r = 0.25L$, is initialized in gaseous CH_4, where the

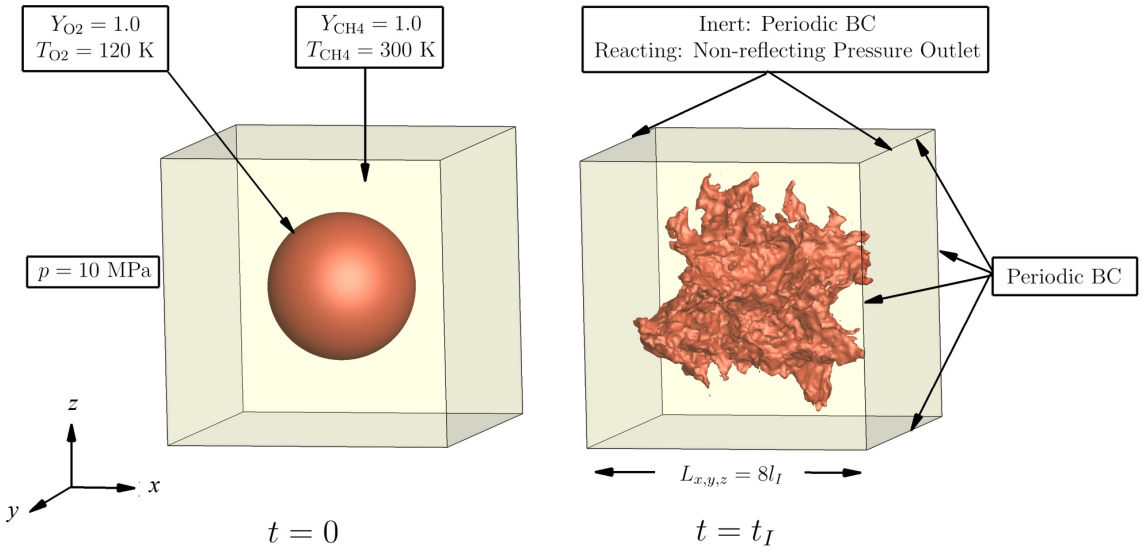


Figure 4.3: DNS investigated at initial time $t = 0$ and one eddy turnover time $t = t_I$. Isosurface shows stoichiometric mixture fraction $Z = 0.2$ for the inert case.

radial profile of the initial condition is chosen to match inert and reacting steady one-dimensional Cantera [182] counterflow diffusion flame calculations, solved in mixture-fraction space and incorporating the PR EoS, under the same fuel and oxidizer conditions. For the reacting cases, the initial temperature and composition profile corresponds to maximum strain rates (from one-dimensional flames) of 2×10^5 s^{-1} and 2×10^6 s^{-1} for cases Da = 780 and Da = 10, respectively. Fuel and oxidizer temperatures are set to $T_{CH4} = 300$ K and $T_{O2} = 120$ K, respectively, while the pressure is set at 10 MPa. The laminar flame speed S_L of a stoichiometric premixed flame of $S_L = 0.306$ ms^{-1} is evaluated through Cantera [182] at a pressure of 10 MPa and initial temperature of 210 K (the average of fuel and oxidizer temperature). Note that the critical temperature T_c and pressure P_c for oxidizer and fuel are $T_{c,O2} = 154.6$ K and $P_{c,O2} = 5.04$ MPa, and $T_{c,CH4} = 190.6$ K and $P_{c,CH4} = 4.60$ MPa, respectively.

These operating conditions are chosen to match practical LOX/GCH$_4$ combustors, and were investigated in previous studies [183, 184]. Periodic boundary conditions are used for all boundaries for the inert case. For the reacting cases, non-reflecting pressure outlets are used in both boundaries in the x-direction, while the remaining

boundaries are periodic.

The initial velocity profile was generated with a synthetic isotropic turbulence generator by Saad et al. [185] with zero mean velocity, based on the von Kármán-Pao energy spectrum:

$$E(\kappa) = \mathcal{C}\frac{u'^2}{\kappa_I}\frac{(\kappa/\kappa_I)^4}{[1+(\kappa/\kappa_I]^{17/6}}\exp\left[-2\left(\frac{\kappa}{\kappa_\eta}\right)^2\right], \tag{4.4a}$$

$$\mathcal{C} = 1.453, \tag{4.4b}$$

$$\kappa_I = 0.746834/l_I, \tag{4.4c}$$

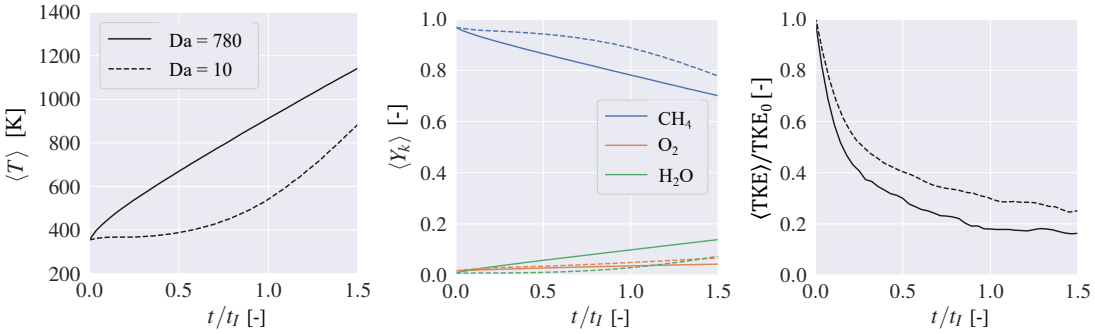
where u' is the fluctuating velocity, κ is the wavenumber, and κ_η the Kolmogorov wavenumber. The chosen scaling constant $\mathcal{C}$ and large-eddy wavenumber κ_I are typical for isotropic turbulence [186]. In all cases, the integral length-scale l_I and root mean-squared (RMS) velocity fluctuation u' have been chosen to produce a turbulent Reynolds number Re_t of 80, which has been computed with the averaged kinematic viscosity of O_2 and CH_4 at 120 K and 300 K, respectively.

In the reacting cases, two different Damköhler numbers, Da, of 780 and 10 are investigated, corresponding to flamelet and unsteady regimes [187], respectively. The Damköhler number is given by the ratio of physical time-scale t_{conv} and chemical time-scale t_{chem}:

$$\mathrm{Da} = \frac{t_{conv}}{t_{chem}}, \tag{4.5}$$

where $t_{chem} = 0.412$ μs is approximated from the extinction strain rate of a one-dimensional counterflow diffusion flame of a LOX/GCH_4 mixture under similar conditions, and physical time is evaluated from the eddy turnover time $t_{conv} = t_I$. Figure 4.4a shows that the mean temperature $\langle T \rangle$ is lower when Da = 10 than when Da = 780, due to the presence of local extinction. This is also reflected in Figure 4.4b where the consumption of CH_4 is slower in the case Da = 10 than the case Da = 780. This decrease in temperature and composition also results in a slower decay of the turbulence, as shown by the mean turbulent kinetic energy $\langle \mathrm{TKE} \rangle$, normalized by the initial TKE, shown in Figure 4.4c.

An additional inert simulation with ideal gas law is performed to demonstrate



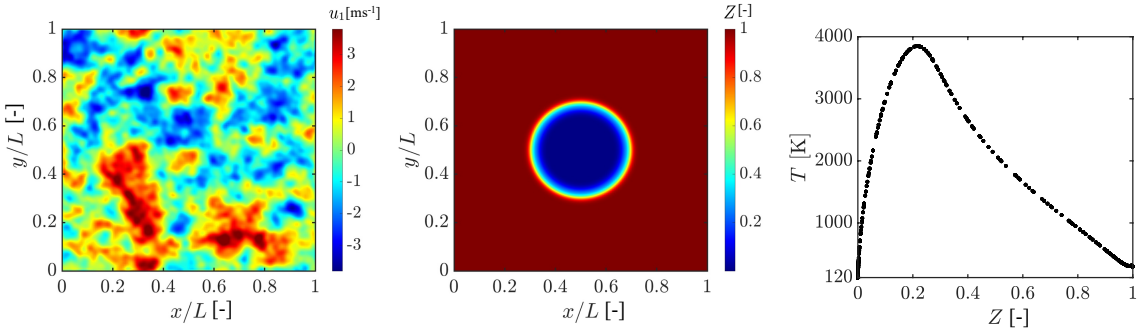
(a) Temperature. (b) Mass fraction. (c) Turbulent kinetic energy.

Figure 4.4: Temporal evolution of global temperature T, mass fraction Y_k, and normalized turbulent kinetic energy TKE for two reacting cases.

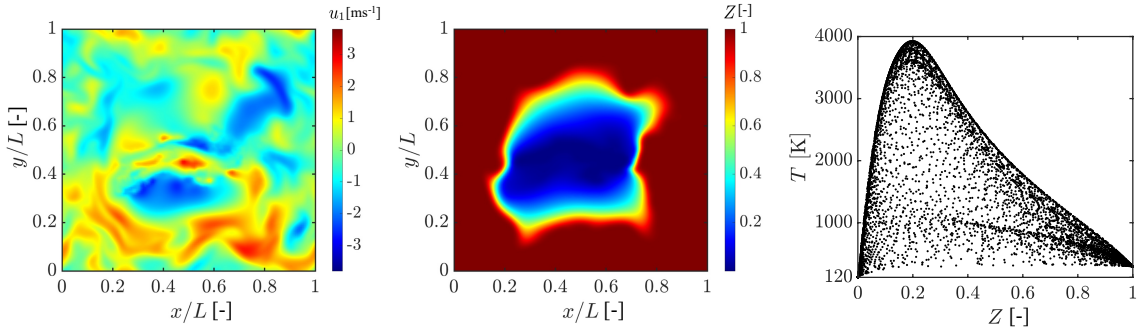
real-fluid effects on SGS terms that can arise from the non-linearities of the PR EoS. For this ideal configuration, atmospheric conditions $p = 101.325$ kPa at room temperature are employed, with T_{CH4} and T_{O2} at 300 K.

In this study, analysis is performed on all cases after $t = \mathrm{argmax}(t_I, t_{chem})$, which is typically done for DNS of combustion under decaying turbulence in order to ensure the flowfields are independent of initialization [188]. Instantaneous flowfields for axial velocity component u_1, mixture fraction Z, and mixture-fraction conditioned temperature T for the reacting cases at $t = 0$ and $t = t_I$ are shown in Figure 4.5.

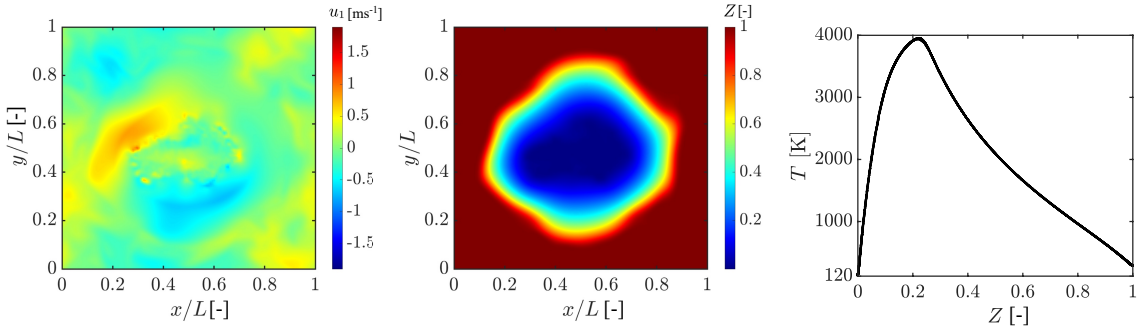
Table 4.1 summarizes the DNS cases examined in this study. The domain lengths in all direction were chosen to be eight times the size of the integral length-scale l_I to minimize effects of the boundary conditions. The cell size Δ is prescribed on the order of the Kolmogorov length-scale η_k, ensuring that all length-scales are resolved. In addition, a mesh refinement study was performed, where the energy spectra of velocity was found to converge between 128^3 and 256^3. Simulations for all three cases are advanced with an acoustic CFL number of unity, corresponding to timesteps of 2.5 and 0.5 ns for cases Da = 780 and Da = 10, respectively. The simulations were performed using 960 Intel Xeon (E5-2698 v3) processors, and 2.3 μs and 0.6 μs of physical time could be completed in about an hour wall clock time for cases Da = 780 and Da = 10, respectively.



(a) Case Da = 10 at initial time $t = 0$.



(b) Case Da = 10 at eddy turnover time $t = t_I$.



(c) Case Da = 780 at eddy turnover time $t = t_I$.

Figure 4.5: Axial velocity u_1, mixture fraction Z, and conditional temperature T for the reacting cases at transverse location $z = 0$.

4.4 ML Methods

In this chapter, we employ the random forest (as described in Section 2.2.5) as our regression algorithm for predicting SGS stresses and SGS temperature. Table 4.2 summarizes the input/features, outputs, and data for the random forests employed in this study. All random forests are trained with snapshots at one eddy turnover time

Table 4.1: Summary of DNS cases.

Case	$N_{x,y,z}$	$L_{x,y,z}$ [μm]	Re_t	l_I [μm]	η_k [μm]	Δ [μm]	t_I [μs]	u' [ms^{-1}]
Inert	128	500	80	62.5	2.32	3.91	286	0.22
Da = 780	128	500	80	62.5	2.32	3.91	286	0.22
Da = 10	128	60	80	7.50	0.278	0.469	4.12	1.80
Ideal EoS	128	500	80	62.5	2.32	3.91	3	20.67

$t = t_I$ and tested on the three cases at $t = 1.5t_I$, to avoid issues related to overfitting on the training set.

For SGS stresses, two different sets of feature, or inputs, are employed to train the random forests. One feature set corresponds to a domain-blind random forest RF_BLIND, consisting only of velocity, and the first and second spatial derivatives of velocity. The other set considers Galilean invariant basis functions constructed from strain $\widetilde{S}_{ij}$ and rotation $\widetilde{R}_{ij}$ tensors as features, shown to predict anisotropy well in a previous study [189]. These Galilean invariant features are used to train the random forest RF_INFORM. In order to investigate the generalizability of random forests in the absence of a vast representative dataset, we evaluate the predictive performance of three additional random forests RF_INERT, RF_DA780, and RF_DA10, which are trained solely from the inert, Da = 780, and Da = 10 cases, respectively.

In addition, we also examine the performance of random forest in predicting thermodynamic quantities. Since SGS temperature is significant for reacting transcritical cases, training and testing data for RF_TSGS are taken from the two transcritical reacting cases.

Table 4.2: Random forests employed in this study.

Random forest	RF_INFORM	RF_BLIND	RF_INERT	RF_DA780	RF_DA10	RF_TSGS
Training data ($t = t_I$)	Inert, Da = 780, Da = 10	Inert, Da = 780, Da = 10	Inert	Da = 780	Da = 10	Da = 780, Da = 10
Testing data ($t = 1.5t_I$)	Inert, Da = 780, Da = 10					Da = 780, Da = 10
Features (Input)	$\widetilde{S}_{ij}$, $\widetilde{S}_{ik}\widetilde{S}_{kj}$, $\widetilde{R}_{ik}\widetilde{R}_{kj}$, $\widetilde{S}_{ik}\widetilde{R}_{kj} - \widetilde{R}_{ik}\widetilde{S}_{kj}$	$\widetilde{u}_i$, $\frac{\partial \widetilde{u}_i}{\partial x_j}$, $\frac{\partial^2 \widetilde{u}_i}{\partial x_j x_k}$				T_{LES}, $\frac{\partial T_{LES}}{\partial x_j}$, $\frac{\partial^2 T_{LES}}{\partial x_j x_k}$
Output	τ_{ij}^{sgs}					T^{sgs}

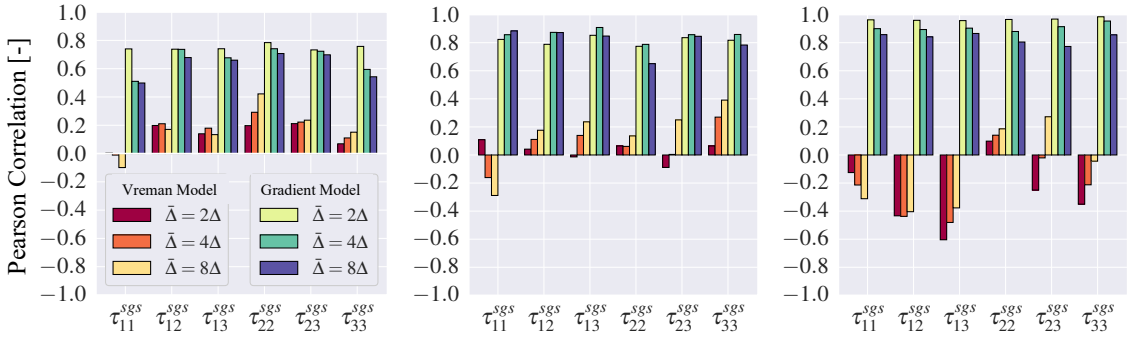
In the present investigation, the random forest regressor implementation from the Scikit-learn library [190] is used. Here, a random forest consisting of fifty decision

trees is employed. The hyperparameters of the random forest are determined using a random grid search approach with a 3-fold cross-validation set. Training is performed once *a priori*, and requires 88 s of walltime with 8 CPUs, when trained on data coarsened for three different filter sizes from a single timestep. Prediction time for a 64^3 dataset requires 2.4 s on a single CPU.

4.5 Results

4.5.1 Algebraic SGS Stress Models

A priori analysis is performed by comparing SGS stresses τ_{ij}^{sgs} computed from filtered DNS, with SGS stress modeled by the Vreman model (Equation (2.11)) and Clark's gradient model (Equation (2.13)). The performance of these SGS models is evaluated through the Pearson correlation coefficient, which measures the linear correlation between two variables. A Pearson correlation of 1 and -1 corresponds to perfectly positive and negative linear relationships, respectively, whereas a correlation of 0 indicates a negligible linear relationship.

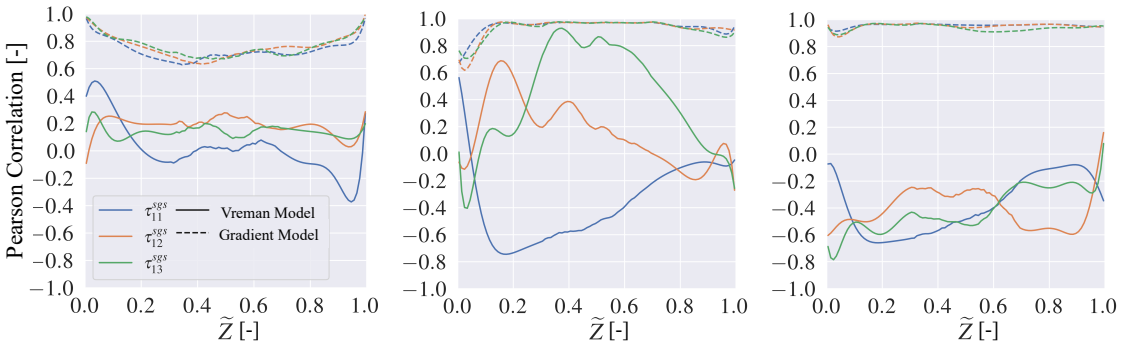


(a) Inert case. (b) Da = 780. (c) Da = 10.

Figure 4.6: Pearson correlation between exact and algebraically modeled SGS stresses for three different filter widths $\overline{\Delta}$.

Figure 4.6 presents the resulting Pearson correlation between exact and algebraically modeled SGS stresses for three different filter widths $\overline{\Delta}$ for all three DNS

cases specified in Table 4.1, at time $t = 1.5t_I$. For all three cases and filter sizes, negative correlations and weak positive correlations ranging from approximately -0.6 to 0.4 are observed for the Vreman model. Negative correlations suggest deviations from the eddy-viscosity hypothesis, which causes the Vreman model to be ineffective. In all three cases and three filter sizes, strong positive correlations, ranging from 0.5 to 0.95, suggest that the gradient model is highly suitable for modeling SGS stresses in transcritical inert and reacting flows.



(a) Inert case. (b) Da = 780. (c) Da = 10.

Figure 4.7: Conditional Pearson correlation with respect to mixture fraction $\widetilde{Z}$ between exact and algebraically modeled SGS stresses τ_{1i}^{sgs} for a single filter width $\overline{\Delta} = 2\Delta$.

The effectiveness of the Vreman and gradient models are further assessed by examining the conditional Pearson correlation for τ_{1i}^{sgs} with respect to the mixture fraction $\widetilde{Z}$ at filter size $\overline{\Delta} = 2\Delta$. The mixture fraction for the reacting cases have been evaluated using Bilger's definition. Figure 4.7a shows that weak correlations ranging from -0.4 to 0.5 are observed throughout the inert case. In both reacting cases in Figures 4.7b and 4.7c, the deviations from eddy-viscosity is much larger than the inert case, as denoted by the presence of highly negative correlations (-0.8) in the Vreman model. In the inert case, the gradient model has the highest correlation of approximately 1.0 in pure CH_4 and pure O_2, and the lowest correlation of 0.6 when $\widetilde{Z} = 0.5$. For the case Da = 780, the gradient model has the lowest correlation (0.7) close to the pure O_2 stream, with the correlation steadily increasing as the mixture approaches stoichiometry ($Z_{st} = 0.2$), after which the correlations remain high (0.85

to 1.0). For the case Da = 10, the correlations for the gradient model are high (0.8 to 1.0) throughout the entire mixture.

The accuracy of the gradient model in predicting the magnitude of SGS stresses is evaluated by examining the least squares fit between the exact and modeled SGS stresses. A slope greater than unity indicates underprediction of the modeled SGS stresses, while a slope less than unity indicates overprediction. Figure 4.8 shows that the slopes from the gradient model range from 1 to 4.5. The average of the slopes is 1.98, which suggests that the gradient model with a constant coefficient should employ $\mathcal{C}_g = 1/6$, instead of the typical $\mathcal{C}_g = 1/12$. However, since a wide range of coefficients are observed, a dynamic gradient model scheme is likely more suited in *a posteriori* simulations. This is confirmed by results from *a posteriori* evaluations of the dynamic gradient model from transcritical inert DNS [191].

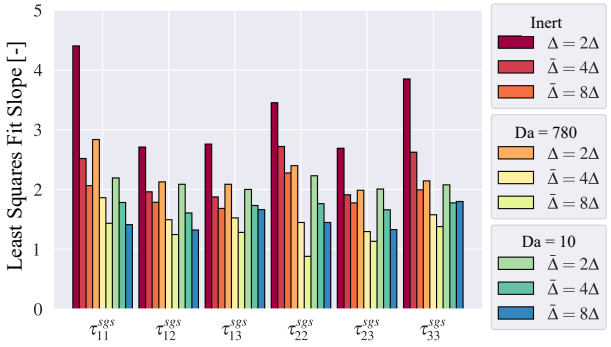
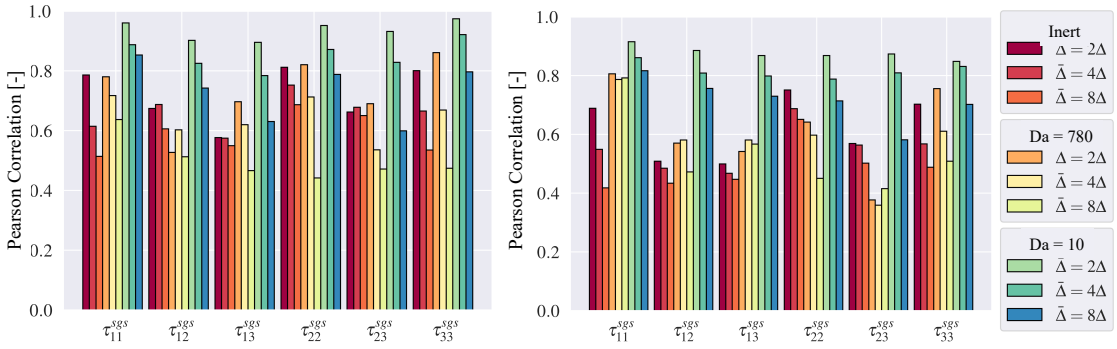


Figure 4.8: Slopes from least squares fits of exact and gradient modeled SGS stress for three different filter widths $\overline{\Delta}$.

4.5.2 Random Forest SGS Stress Models

The *a priori* analysis performed in Section 4.5.1 is repeated in this section for the SGS stresses modeled by random forest regressors. Figure 4.9 presents the Pearson correlation between exact SGS stresses and the SGS stresses modeled by the random forests RF_BLIND and RF_INFORM. Details regarding the input, output, and training of these two random forests are described in Table 4.2. Figure 4.9a shows that strong correlations (0.4 to 0.95) are observed when the random forest is trained

with an uninformed approach, which is similar to the gradient model and higher than the Vreman model in Figure 4.6. Figure 4.9b demonstrates that the employment of invariant basis functions as features decreases the range of correlations (0.35 to 0.9) by 0.05. This small decrease is likely caused by the additional constraints placed on the random forest when forming a hypothesis space.

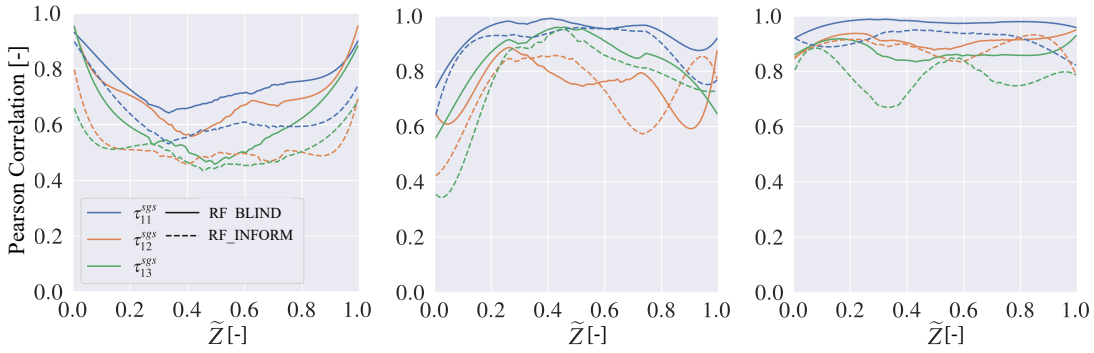


(a) RF_BLIND. (b) RF_INFORM.

Figure 4.9: Pearson correlation between exact and random forest modeled SGS stresses for three different filter widths $\overline{\Delta}$.

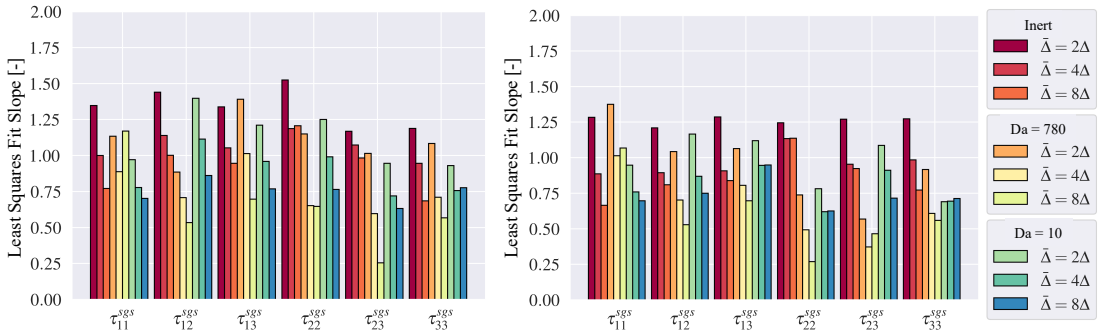
Figure 4.10 presents the Pearson correlation between exact and random forest SGS stresses τ_{1i}^{sgs} conditioned to mixture fraction $\widetilde{Z}$ at $\overline{\Delta} = 2\Delta$. In the inert case, shown in Figure 4.10a, the highest correlation from RF_BLIND of approximately 0.95 is observed in pure CH_4 and pure O_2, and lowest correlation of 0.5 when $Z = 0.5$. For the case Da = 780 in Figure 4.10b, RF_BLIND possesses the lowest correlation (0.7) close to the O_2 stream, with the correlation steadily increasing as the mixture approaches stoichiometric conditions ($\widetilde{Z}_{st} = 0.2$), after which the correlations remain high (0.85 to 1.0). For the case Da = 10, shown in Figure 4.10c, the correlations for the gradient model are high (0.8 to 1.0) throughout the entire mixture. The conditional Pearson correlation produced from RF_BLIND in all three cases are similar qualitatively and quantitatively to correlations from the gradient model in Figure 4.7. This suggests that RF_BLIND has approximated a function similar to the gradient model, even when trained solely on exact SGS stresses and without any prior knowledge of the gradient Model. The correlations from RF_INFORM share similar qualitative behaviors as the correlations from RF_BLIND, but with up to a

0.2 lower values.



(a) Inert case. (b) Da = 780. (c) Da = 10.

Figure 4.10: Conditional Pearson correlation as a function of mixture fraction $\widetilde{Z}$ between exact and random forest modeled SGS stresses τ_{1i}^{sgs} for a single filter width $\overline{\Delta} = 2\Delta$.



(a) RF_BLIND. (b) RF_INFORM.

Figure 4.11: Slopes from least squares fits of exact and random forest modeled SGS stress for three different filter widths $\overline{\Delta}$.

Figure 4.11 presents slopes from least squares fits between the exact and the random forest SGS stresses. Figure 4.11a shows that the slopes from RF_BLIND range from 0.25 to 1.6, with an average slope of 0.96, which demonstrates excellent agreement between modeled and exact magnitudes of SGS stresses. The employment of invariant features leads to lower slopes (0.25 to 1.35), with an average slope of 0.867,

as presented in Figure 4.11b. The use of the invariant feature set not only leads to lower correlations, but also to an overprediction in magnitudes of SGS stresses.

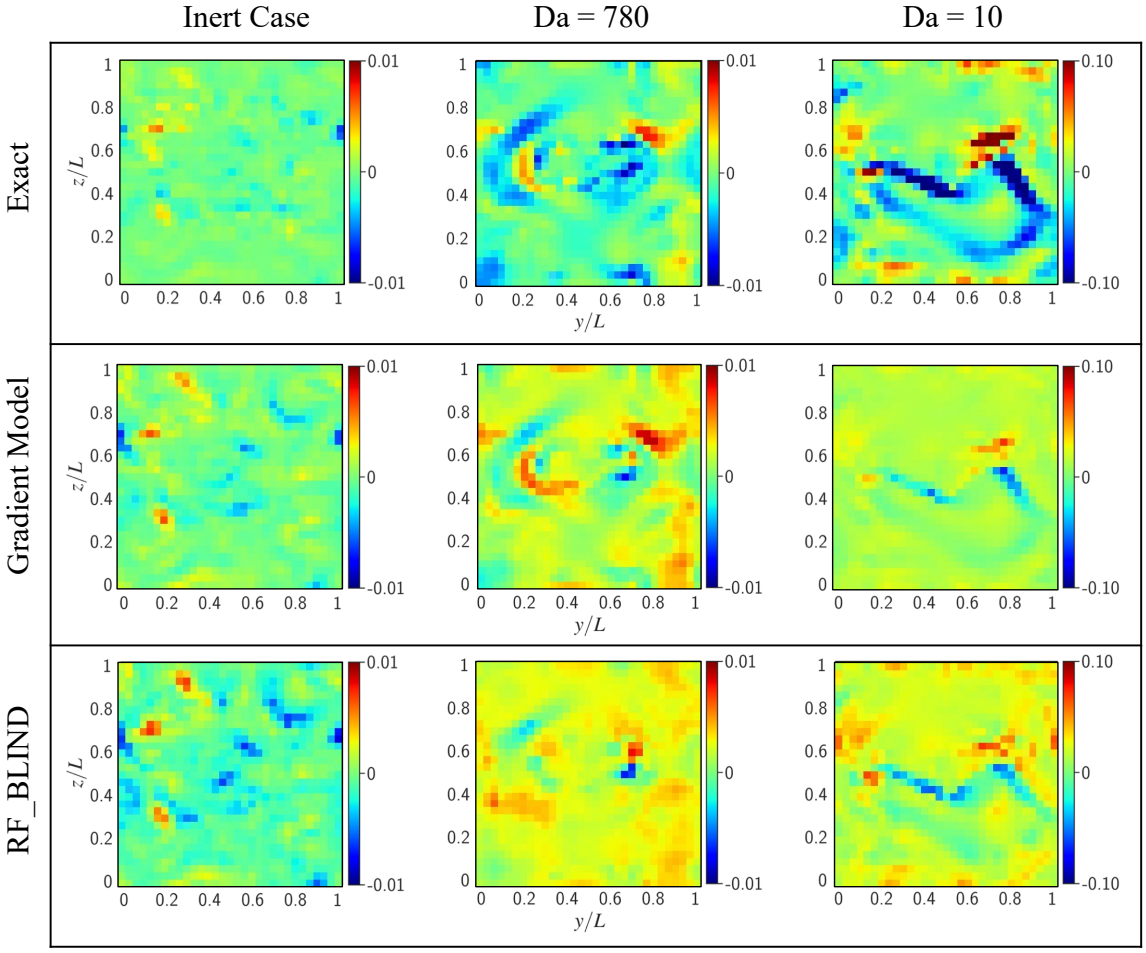


Figure 4.12: Comparison of exact and modeled SGS stress $\tau_{12}^{sgs}/\overline{\rho}$ [$\mathrm{m^2 s^{-2}}$] at filter width $\overline{\Delta} = 4\Delta$ at axial location $x = 0$.

Figure 4.12 compares instantaneous fields for the exact and modeled SGS stress $\tau_{12}^{sgs}/\overline{\rho}$ at filter width $\overline{\Delta} = 4\Delta$. In the inert case, both SGS stresses from the gradient model and RF_BLIND are in good agreement with the exact term. For Da = 780, the gradient model is in better agreement with the exact term than RF_BLIND. This is further supported by the difference in Pearson correlation for this particular case shown by the gradient model (0.9) and RF_BLIND (0.6) in Figures 4.6 and 4.9, respectively. For Da = 10, RF_BLIND predicts the magnitude of the SGS stress better

than the gradient model, which is also observed in the slopes shown by RF_BLIND (0.9) and the gradient model (1.5) shown in Figures 4.8 and 4.11.

Figure 4.13 presents Pearson correlation from examining the generalizability of random forests in the presence of limited data. As presented in Table 4.2, we employ three different random forest regressors, each trained on only one DNS case, and examine their performance when tested on the two remaining cases. Random forest RF_INERT demonstrates a similar range of correlations (0.5 to 0.85) to RF_ALL when tested on the inert case with a filter size $\overline{\Delta} = 2\Delta$. However, lower ranges are observed for RF_INERT when tested on the cases Da = 780 (0.4 to 0.75) and Da = 10 (0.5 to 0.9). RF_DA780 also possesses a similar correlation as RF_BLIND when tested on case Da = 780 (0.5 to 0.9), but worse correlations when tested on the inert case (0.4 to 0.8) and case Da = 10 (0.8 to 0.9). Lastly, RF_DA10 performs similarly to RF_BLIND when tested on Da = 10 (0.85 to 0.95) but performs worse when tested on the inert (0.5 to 0.8) and Da = 780 (0.55 to 0.8) cases. These three random forests perform as well as RF_BLIND on a test set that is represented well by the training set. However, the effectiveness of random forests decreases when modeling on out-of-sample distributions. Nevertheless, these out-of-sample predictions are more accurate than the Vreman model, thus demonstrating an appreciable degree of generalizability.

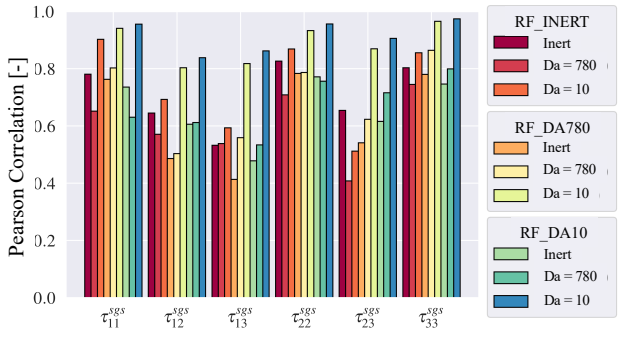


Figure 4.13: Pearson correlation between exact and random forest modeled SGS stresses, from three different random forest regressors, for a single filter width $\overline{\Delta} = 2\Delta$.

4.5.3 Data-driven Discovery of SGS Stress Model

In this section, we examine how the interpretability of random forests can be employed as a tool for model discovery.

Figure 4.14 presents feature importance scores extracted from RF_BLIND for τ_{1i}^{sgs}. For all three SGS stresses τ_{1i}^{sgs} shown, the highest scores are from $\partial\widetilde{u}_1/\partial x_k$ and $\partial\widetilde{u}_i/\partial x_k$ for three spatial dimensions. We employ this observation to formulate a sparse symbolic regression problem (see Section 2.2.3):

$$\frac{\tau_{ij}^{sgs}}{\overline{\rho}u'^2} = f_{sym}\left(\frac{\overline{\Delta}}{u'}\frac{\partial\widetilde{u}_i}{\partial x_k}, \frac{\overline{\Delta}}{u'}\frac{\partial\widetilde{u}_j}{\partial x_k}\right) \tag{4.6}$$

where the independent variables consist of 2nd-order polynomial functions of the non-dimensionalized selected features. Equation (4.6) is non-dimensionalized by density, filter width and initial RMS velocity to ensure dimensional consistency in the final model. This is essential for improving the dimensionality of this sparse symbolic regression problem. Since the dimensionality scales with n^d for n number of candidate variables, as discussed in Section 2.2.3, the employment of the feature importance score for reducing 30 candidate variables to six candidate variables results in a 25-fold reduction in dimensionality.

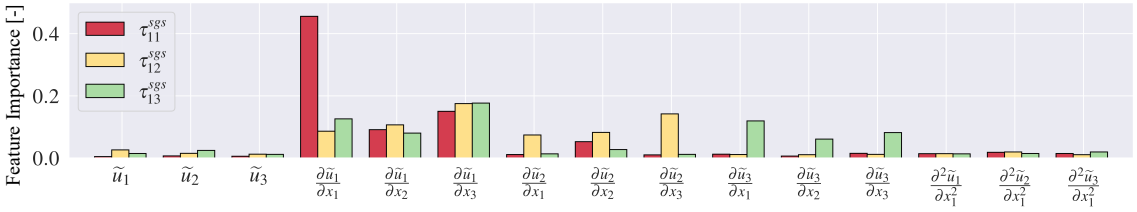


Figure 4.14: Fifteen feature importance scores from RF_BLIND. The other fifteen features, with importance scores less than 0.02, are not shown for brevity.

The following equations present the SGS model that resulted from applying sparse

symbolic regression:

$$\tau_{11}^{sgs} \simeq \overline{\rho}\overline{\Delta}^2 \left(0.116\frac{\partial\widetilde{u}_1}{\partial x_1}\frac{\partial\widetilde{u}_1}{\partial x_1} + 0.191\frac{\partial\widetilde{u}_1}{\partial x_2}\frac{\partial\widetilde{u}_1}{\partial x_2} + 0.207\frac{\partial\widetilde{u}_1}{\partial x_3}\frac{\partial\widetilde{u}_1}{\partial x_3}\right) \tag{4.7a}$$

$$\tau_{12}^{sgs} \simeq \overline{\rho}\overline{\Delta}^2 \left(0.113\frac{\partial\widetilde{u}_1}{\partial x_1}\frac{\partial\widetilde{u}_2}{\partial x_1} + 0.102\frac{\partial\widetilde{u}_1}{\partial x_2}\frac{\partial\widetilde{u}_2}{\partial x_2} + 0.134\frac{\partial\widetilde{u}_1}{\partial x_3}\frac{\partial\widetilde{u}_2}{\partial x_3}\right) \tag{4.7b}$$

$$\tau_{13}^{sgs} \simeq \overline{\rho}\overline{\Delta}^2 \left(0.119\frac{\partial\widetilde{u}_1}{\partial x_1}\frac{\partial\widetilde{u}_3}{\partial x_1} + 0.117\frac{\partial\widetilde{u}_1}{\partial x_2}\frac{\partial\widetilde{u}_3}{\partial x_2} + 0.109\frac{\partial\widetilde{u}_1}{\partial x_3}\frac{\partial\widetilde{u}_3}{\partial x_3}\right) \tag{4.7c}$$

$$\tau_{22}^{sgs} \simeq \overline{\rho}\overline{\Delta}^2 \left(0.215\frac{\partial\widetilde{u}_2}{\partial x_1}\frac{\partial\widetilde{u}_2}{\partial x_1} + 0.135\frac{\partial\widetilde{u}_2}{\partial x_2}\frac{\partial\widetilde{u}_2}{\partial x_2} + 0.164\frac{\partial\widetilde{u}_2}{\partial x_3}\frac{\partial\widetilde{u}_2}{\partial x_3}\right) \tag{4.7d}$$

$$\tau_{23}^{sgs} \simeq \overline{\rho}\overline{\Delta}^2 \left(0.123\frac{\partial\widetilde{u}_2}{\partial x_1}\frac{\partial\widetilde{u}_3}{\partial x_1} + 0.116\frac{\partial\widetilde{u}_2}{\partial x_2}\frac{\partial\widetilde{u}_3}{\partial x_2} + 0.134\frac{\partial\widetilde{u}_2}{\partial x_3}\frac{\partial\widetilde{u}_3}{\partial x_3}\right) \tag{4.7e}$$

$$\tau_{33}^{sgs} \simeq \overline{\rho}\overline{\Delta}^2 \left(0.251\frac{\partial\widetilde{u}_3}{\partial x_1}\frac{\partial\widetilde{u}_3}{\partial x_1} + 0.177\frac{\partial\widetilde{u}_3}{\partial x_2}\frac{\partial\widetilde{u}_3}{\partial x_2} + 0.124\frac{\partial\widetilde{u}_3}{\partial x_3}\frac{\partial\widetilde{u}_3}{\partial x_3}\right) \tag{4.7f}$$

The resulting model can be rewritten as:

$$\tau_{ij}^{sgs} \simeq \overline{\rho}\overline{\Delta}^2 \left(\mathcal{C}_1\frac{\partial\widetilde{u}_i}{\partial x_1}\frac{\partial\widetilde{u}_j}{\partial x_1} + \mathcal{C}_2\frac{\partial\widetilde{u}_i}{\partial x_2}\frac{\partial\widetilde{u}_j}{\partial x_2} + \mathcal{C}_3\frac{\partial\widetilde{u}_i}{\partial x_3}\frac{\partial\widetilde{u}_j}{\partial x_3}\right) \tag{4.8}$$

where the resulting model coefficients $\mathcal{C}_{\{1,2,3\}}$ range from 0.102 to 0.251. Equation (4.8) is similar in form to the gradient model (Equation (2.13)), but possesses three model coefficients instead of one. By observing that $\mathcal{C}_{\{1,2,3\}}$ are of the same order of magnitudes, and collapsing the three coefficients by evaluating the average model coefficients, we recover the gradient model:

$$\tau_{ij}^{sgs} \simeq \overline{\rho}\mathcal{C}_x\overline{\Delta}^2\frac{\partial\widetilde{u}_i}{\partial x_k}\frac{\partial\widetilde{u}_j}{\partial x_k} \tag{4.9}$$

where the model coefficient $\mathcal{C}_x = 0.147$ is similar in value to the suggested model coefficient of 0.167 from Section 4.5.1. This result demonstrates that the employment of sparse symbolic regression, in conjunction with random forest feature importance can be employed to discover an algebraic expression, similar to the effective gradient model, for modeling SGS stresses in transcritical flows.

Since the present method relies on the random forest feature importance score,

a statistical test must be employed to test for the effects of significant correlation amongst the features. If multiple features in the modeling basis are significantly correlated, they act as exchangeable surrogates for each other during the calculation of feature importance scores. This is similar to the phenomenon of multicollinearity in classical statistics [192]. Under such conditions, metrics such as the MDI are susceptible to correlation bias, and can generate erroneous importance scores [193, 194]. As a note, almost all algorithms for estimating feature importance, including Shapley additive explanations [195] exhibit such correlation bias. As an alternative, Principal Component Analysis may be utilized to engender orthogonal bases for new features that are independent. However, these derived features are often difficult to ascribe physical meanings to, obfuscating their utility toward interpretability.

We utilize the Spearman correlation as a statistical test for evaluating the correlation amongst the features in the modeling basis. While the Pearson correlation is a statistical tool used for evaluating linear relationships, the Spearman correlation evaluates the monotonicity of variables in both linear and non-linear functions, *i.e.*, whether the increasing or decreasing trend is being preserved. Spearman correlations of 1 and -1 correspond to a perfect monotonic relationship, while 0 corresponds to a negligible monotonic relationship. Figure 4.15 shows that Spearman correlations between different features from RF_BLIND are weak (between -0.4 and 0.4), which indicates that the feature importance scores are not spurious.

4.5.4 SGS Temperature Model

In this section, we extend the application of the present data-driven methods towards modeling SGS temperature. Figure 4.16 presents the Pearson correlation and slopes from least squares fits between exact and random forest-modeled SGS temperature. High correlations (0.7 to 0.9) and slopes ranging from 0.7 to 1.5 are observed for all three filter widths, indicating good performance from the random forest SGS temperature model.

Unlike the random forests for modeling SGS stresses in Section 4.5.2, the feature importance scores from RF_TSGS do not provide physical insight due to the issue

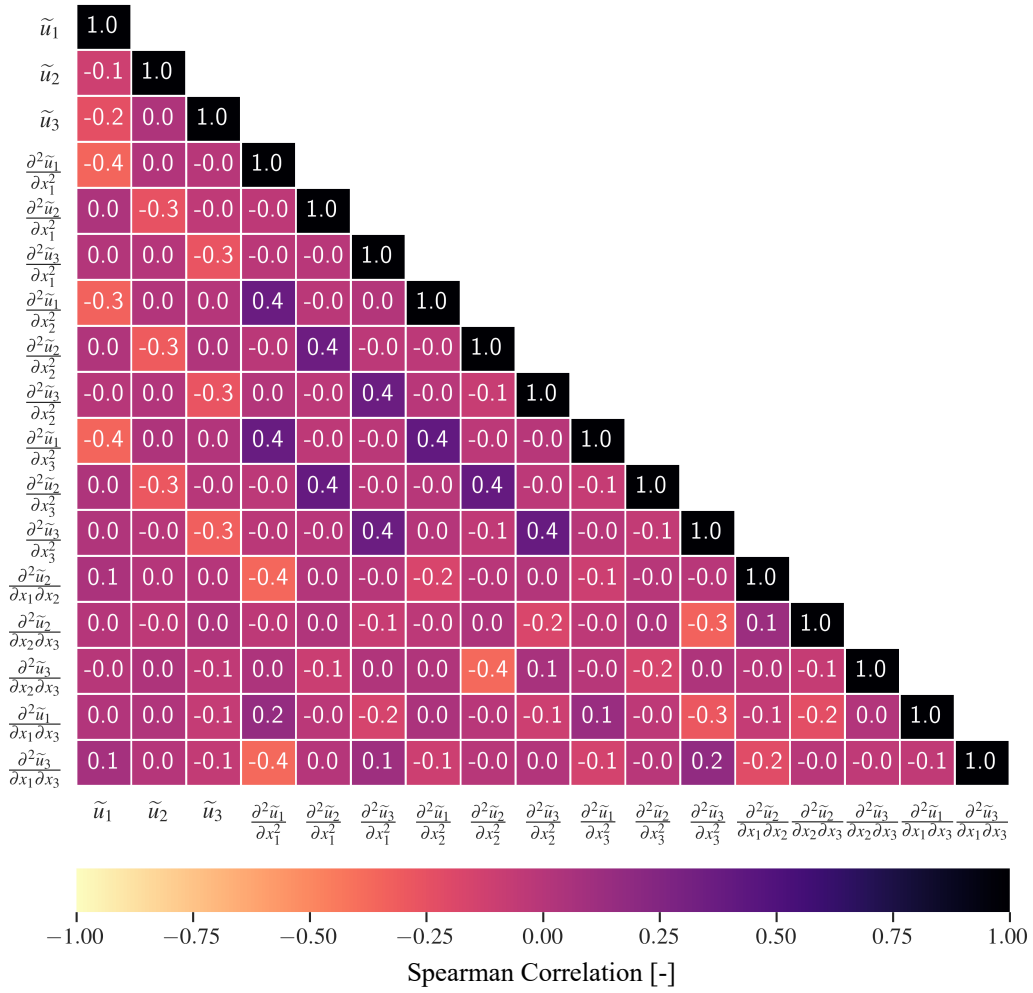


Figure 4.15: Spearman correlation matrix for selected features from RF_BLIND. Features with correlations less than 0.2 are not shown for brevity.

of multicollinearity, as T_{LES} and its gradients are used as features. In a reacting configuration, large temperature gradients are usually observed in a certain temperature range, and thus both these quantities can be significantly correlated. Nevertheless, a sparse symbolic regression problem can still be formulated without reducing the number of independent variables, as the feature set for T^{sgs} is three times smaller than the feature set for τ_{ij}^{sgs}. We repeat the sparse symbolic regression procedure

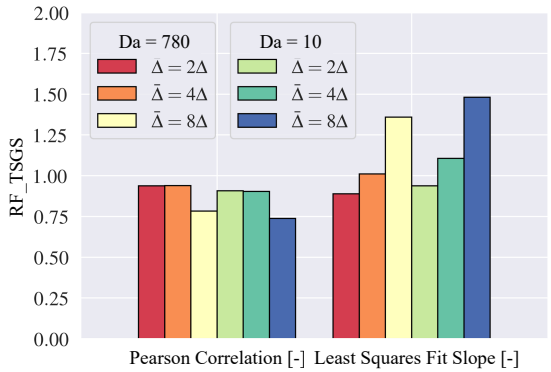


Figure 4.16: Pearson correlation and slopes from least squares fits between exact and random forest-modeled SGS temperature, for three different filter widths $\overline{\Delta}$.

from Section 4.5.3:

$$T^{sgs} = f_{sym}\left(\widetilde{T}_{LES}, l_{char}\frac{\partial \widetilde{T}_{LES}}{\partial x_k}, l_{char}^2\frac{\partial^2 \widetilde{T}_{LES}}{\partial x_k \partial x_k}\right), \tag{4.10}$$

where the independent variables consist of 2nd-order polynomial functions of the features from RF_TSGS. Note that the independent variables are ensured to be dimensionally consistent with T^{sgs} by multiplying the gradients with a characteristic length-scale l_{char}. This characteristic length-scale can be chosen either as the filter width $\overline{\Delta}$ or a flame thickness δ_f. In the present study, δ_f can be extracted from the DNS by dividing the difference between flame and inert temperature by the maximum temperature gradient.

The following equations present the SGS temperature model that resulted from applying sparse symbolic regression:

$$T^{sgs} = \frac{\delta_F^2}{\widetilde{T}_{LES}}\left[0.00082\left(\frac{\partial \widetilde{T}_{LES}}{\partial x_1}\right)^2 + 0.00109\left(\frac{\partial \widetilde{T}_{LES}}{\partial x_2}\right)^2 + 0.00109\left(\frac{\partial \widetilde{T}_{LES}}{\partial x_3}\right)^2\right] \tag{4.11}$$

where $l_{char} = \delta_f$ has been chosen since a better fit is obtained when performing the least squares fit between the exact and modeled SGS temperature. By taking the

average of the model coefficients, we obtain the algebraic expression:

$$T^{sgs} = \frac{\mathcal{C}_T \delta_F^2}{\widetilde{T}_{LES}} \left(\frac{\partial \widetilde{T}_{LES}}{\partial x_k} \right)^2 \tag{4.12}$$

where $\mathcal{C}_T = 0.001$.

Figure 4.17 presents the Pearson correlation and slopes from least squares fits between exact and SGS temperature from the discovered algebraic T^{sgs} model. High correlations of approximately 0.9 are observed for $\overline{\Delta} = 2$ and $\overline{\Delta} = 4$, while a reasonable correlation of approximately 0.5 is seen for $\overline{\Delta} = 8$. The lower correlation compared to RF_TSGS is likely caused by the presence of the l_1-norm in Equation (2.22), which encourages less significant terms to vanish from the discovered model. Least squares fit slopes ranging from 0.8 to 1.3 are observed for all three filter widths.

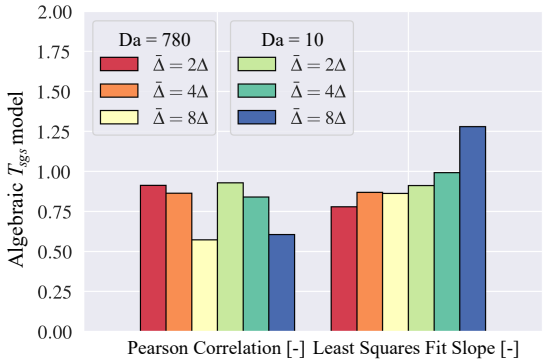


Figure 4.17: Pearson correlation and slopes from least squares fit between exact and algebraic-modeled SGS temperature.

4.6 Summary

This chapter explores random forest regressors and sparse symbolic regression approaches in modeling SGS closure in a small transcritical flow dataset. To this end, DNS of inert and reacting transcritical LOX/GCH_4 non-premixed mixtures under

decaying turbulence were performed. Pressure and temperature were chosen to correspond to conditions in rocket combustors to examine conditions for which commonly-employed SGS are less matured.

A priori analysis was conducted by comparing exact SGS stresses from Favre-filtered DNS data with algebraic and data-driven SGS models. This analysis showed that the SGS stresses evaluated by Vreman SGS model correlated poorly with the corresponding exact terms. In contrast, good correlations were observed from the gradient SGS model. Results demonstrated a wide range of magnitude errors in the gradient model, which suggests that a dynamic gradient model approach is suited in *a posteriori* simulations. Random forests demonstrated high correlations when trained on datasets which are representative of the test sets, with reasonable predictions for the magnitude of SGS stresses. However, correlations were shown to decrease significantly when tested out-of-sample.

Sparse symbolic regression was performed to discover an algebraic expression for SGS stresses from non-linear transformations of velocity and its derivatives. The interpretability of random forests was demonstrated to reduce the dimensionality of the sparse symbolic regression problem by 25 times, by employing the feature importance score for variable selection. The derived algebraic expression was shown to be similar to the gradient model.

Sparse symbolic regression was also performed to evaluate SGS temperature, a term which emerges from filtering the non-linear real-fluids EoS. The discovered algebraic expression demonstrated reasonable correlations and magnitudes when predicting SGS temperature. A random forest SGS temperature model was shown to perform better than the algebraic model.

Results demonstrated that random forests can perform as effectively or better as suitable algebraic models when modeling SGS stresses, if trained on a sufficiently representative database. However, in the absence of such a database, this good performance was not replicated. Nevertheless, the employment of random forests can provide insight into the discovery of SGS models via symbolic regression through the feature importance score, as long as features are not significantly correlated.

Chapter 5

Classification within a Reacting Flow Solver*

5.1 Introduction

In the previous chapter, we discussed results that highlighted potential OOD errors in ML-based predictions when trained on insufficiently representative data. These errors can introduce stability issues when integrating ML methods within numerical multi-physics flow solvers. The present chapter presents a strategy for ameliorating this issue by employing a classification algorithm that assigns well-tested domain knowledge-based combustion submodels (as discussed in Section 1.2) of varying fidelity and complexity within a shared simulation domain. Thus, the potential approximation errors made by the ML algorithm are limited by the predictive capability of the lowest performing submodel.

In the approach that is proposed in this work, local thermo-physical quantities in the flowfield are utilized as features for a random forest algorithm that spatially and dynamically assigns combustion submodels. Errors made by submodels, when predicting user-defined QoI, are used to construct the labels used for training random

*This chapter contains previously published work from Chung et al. [9], with minor modifications. W.T. Chung planned, performed, and analyzed experiments, and performed simulations. A.A. Mishra assisted with planning experiments. N. Perakis assisted in performing simulations.

forest models. Overall computational fidelity and cost of the simulation is determined by a user-defined submodel error threshold during training. This approach couples the assigned combustion submodels in the *a posteriori* simulations by employing the mass-conserving approach developed by Wu et al. [69].

To summarize, this chapter involves the following objectives:

- To integrate ML-based classification models for combustion submodel assignment within a reacting flow physics solver, and assess the resulting DA simulations.
- To evaluate the suitability, accuracy, and adjustability of random forests for submodel assignment.

To this end, we evaluate this ML-based approach on simulations of a GOX/GCH_4 single-element rocket combustor [196]. Methods for simulating this turbulent reacting flow configuration are discussed in Section 5.2. The experimental configuration, computational setup and baseline simulations using monolithic combustion models are discussed in Section 5.3. The data-driven framework is introduced in Section 5.4. Results from *a priori* and *a posteriori* assessments of the random forest models are presented and discussed in Section 5.5, before offering concluding remarks in Section 5.6.

5.2 Mathematical Models

5.2.1 Governing Equations

The governing equations that are solved in the present chapter for the LES are the Favre-filtered conservation equations for mass, momentum, energy, and chemical species (see Equation (2.9)). The combustion models that are employed in the present study are described in detail in Section 5.2.2.

Simulations are performed by employing an unstructured compressible finite-volume solver [69, 102, 178]. A central scheme, which is 4th-order accurate on uniform meshes, is used along with a 2nd-order ENO scheme. The ENO scheme

is activated only in regions of high local density variation using a threshold-based sensor. A Strang-splitting scheme is employed for time-advancement, combining a strong stability preserving 3rd-order Runge-Kutta (SSP-RK3) scheme for integrating the non-stiff operators with a semi-implicit Rosenbrock-Krylov scheme [179] for advancing the chemical source terms. The dynamic Smagorinsky model [48] is used as closure for the SGS stresses. Turbulence/chemistry interaction is accounted for using the dynamic thickened-flame model [52], employing a maximum thickening factor of 3, which is estimated through 1D flame calculations *a priori*. Outside the flame region, both turbulent Prandtl and Schmidt numbers are prescribed at constant values of 0.7.

5.2.2 Combustion Models

In this chapter, we perform LES calculations that employ three different combustion submodels, namely the FRC model, the FPV model [53, 54], and IM model. The FRC model is defined by solving the species transport equation, Equation (2.9d), through direct integration. This method does not rely on strong assumptions on flame structure and is suitable for representing complex flows as well as intermediate species and unsteady effects. Despite the high-fidelity offered by FRC, since the cost of evaluating the chemical source terms scale linearly with the number of species, the utilization of a large chemical mechanism can be prohibitively costly. FPV approach aims to alleviate the computational cost of combustion chemistry by representing the thermochemical state space using a low-dimensional manifold based on flamelets, a series of one-dimensional diffusion flames. FPV relies on the observation that laminar diffusion flames are weakly affected by the presence of turbulence, which allows the turbulent diffusion flame to be represented by flamelets. While FPV is computationally efficient, it assumes adiabaticity and cannot model effects of heat-flux across boundaries well. Lastly, IM models can only consider mixing without combustion chemistry.

The representation of transported chemical scalar Φ_k between FRC and the two tabulated chemistry models is dissimilar: FRC uses a chemical state-vector $\widetilde{Y}_k =$

$[\widetilde{Y}_1, \ldots, \widetilde{Y}_{N_S}]^T$, consisting of N_S number of chemical species, while the FPV and IM state-vector is represented in terms of a low-dimensional manifold $\widetilde{Y}_k = \mathcal{M}(\widetilde{\Phi}_k^{\mathcal{K}})$, where $\widetilde{\Phi}_k^{\mathcal{K}}$ is the state vector that is used to parameterize the manifolds in models $\mathcal{K} = \{\mathrm{FPV}, \mathrm{IM}\}$. With the flame being artificially thickened as discussed in Section 5.2.1, FPV is parameterized by the mixture fraction and progress variable $\widetilde{\Phi}_k^{\mathrm{FPV}} = [\widetilde{Z}, \widetilde{C}]^T$, which differs from the conventional practice of using a presumed-PDF closure [69]. The progress variable is defined as a linear combination of species mass fractions of combustion products (carbon dioxide, water, carbon monoxide, and hydrogen, respectively) [197]: $C = Y_{\mathrm{CO_2}} + Y_{\mathrm{H_2O}} + Y_{\mathrm{CO}} + Y_{\mathrm{H_2}}$. For an inert and adiabatic mixture, the thermochemical state is fully parameterized by a single scalar, $\widetilde{\Phi}_k^{\mathrm{IM}} = [\widetilde{Z}]$.

The present framework resolves the discrepancy in scalar representation when coupling different combustion models with the approach developed by Wu et al. [69]. In this approach, a transport equation for mixture fraction is solved holistically in all models. Reconstruction of the chemical state-vector needed for FRC involves interpolation from the chemistry tables that stores all species, whereas the reconstruction of the progress variable needed for tabulated chemistry involves the sum of all major combustion product species: CO_2, CO, H_2O, and H_2. To ensure consistency between the submodels, the aforementioned reconstruction is applied for the inactive combustion model at the submodel interface at every timestep. Since the conservation laws for mass, momentum, and energy are universal among all combustion submodels, these properties are conserved throughout the domain. In addition, the choice of the dynamically-thickened flame model for the FRC and both manifold-based models avoids potential complications, since this closure model has been successfully applied to previous non-premixed flame simulations employing FRC and tabulated chemistry models [69, 198, 199].

The GRI-3.0 model [200], involving $N_S = 33$ chemical species, is used to describe the reaction chemistry in all combustion models. FRC is incorporated into the LES solver using the Cantera library interface [182]. The molecular diffusion of chemical species is modeled with constant Lewis numbers, which are calculated at equilibrium condition of a stoichiometric CH_4 and O_2 mixture. The chemistry table employed in the FPV-model is constructed from the solution of steady-state counterflow diffusion

flames that are solved in composition space [201].

5.3 Configuration

5.3.1 Experimental Configuration

To evaluate the merit of the classification method, we perform simulations of a single-element GOX/GCH_4 rocket combustor [196]. The experimental configuration consists of a co-axial injector element where the oxidizer flows through a central jet with diameter $d_o = 4\,\text{mm}$ and the fuel is injected via an annulus with inner and outer diameters $d_{f,i} = 5\,\text{mm}$ and $d_{f,o} = 6\,\text{mm}$. The combustion chamber with a total length of 285 mm has a cylindrical shape with diameter $d_{ch} = 12\,\text{mm}$. A conical nozzle is attached at the end of the combustion chamber, having a contraction ratio of 2.5. This setup results in a Mach number of approximately 0.25 in the combustion chamber, which is similar to typical flight configurations. The combustor operates at a nominal operating pressure of 20 bar and a global oxidizer-to-fuel ratio of 2.6, with mass flow rates of oxidizer $\dot{m}_o$ and fuel $\dot{m}_f$ measured at 34.82 g/s and 13.39 g/s, respectively. The temperature of the oxidizer and the fuel supplied at the injector inlet are $T_o = 275\,\text{K}$ and $T_f = 269\,\text{K}$. Static wall pressure and wall heat flux are measured through thermocouples and pressure transducers, installed along the chamber wall.

5.3.2 Computational Setup

In this model-assignment problem, we consider an axisymmetrical domain that is representative of the single-element GOX/GCH_4 rocket combustor, as shown in Figure 5.1. The domain consists of a 3° combustor sector, with a truncation at 0.4 mm to remove the singularity at the centerline. Axisymmetric simulations of rocket combustors have been frequently employed to obtain insight in the turbulent combustion process [202, 203], while offering feasible computational costs. This was found to be crucial for the exploration of a wider range of parameters in the DA method, especially with the use of a detailed FRC-model consisting of 33 chemical species in the present study.

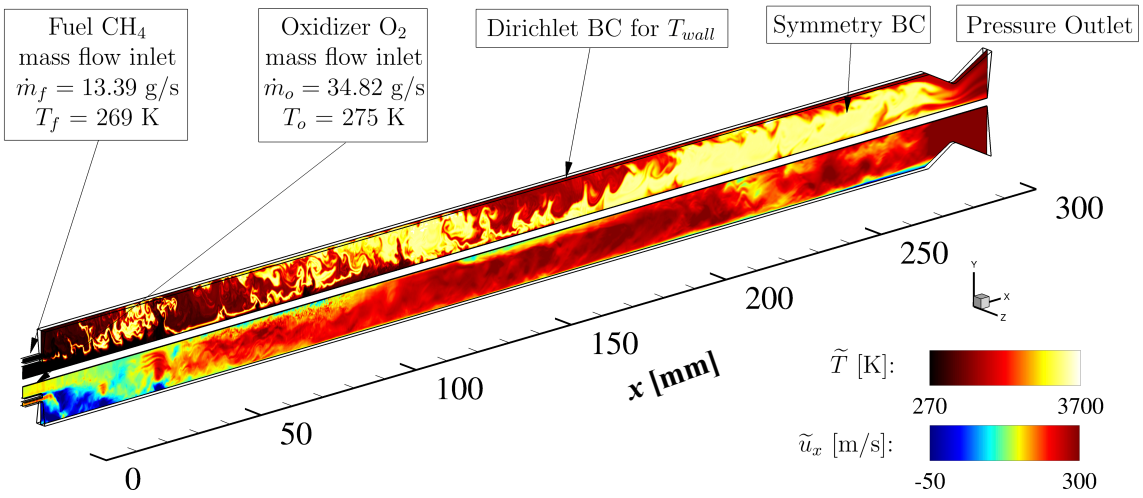
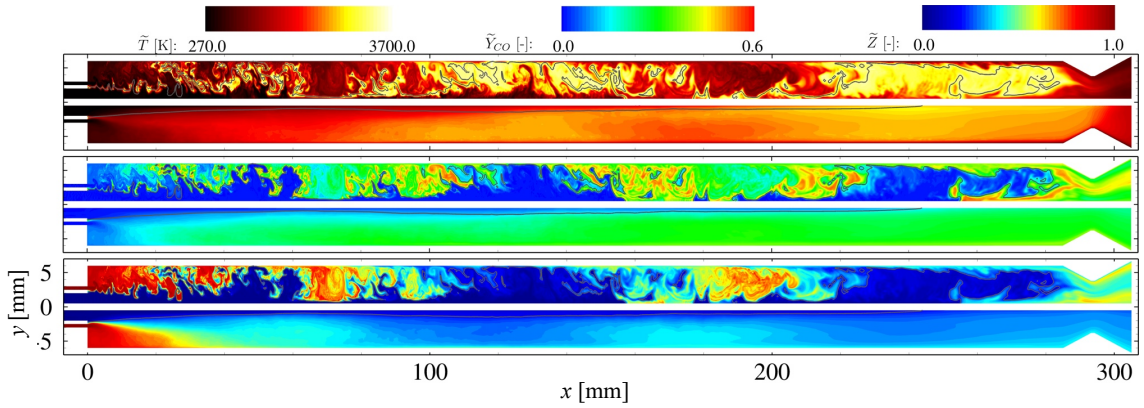


Figure 5.1: Computational domain presented in conjunction with instantaneous temperature (top) and axial velocity (bottom) fields from monolithic FRC simulations.

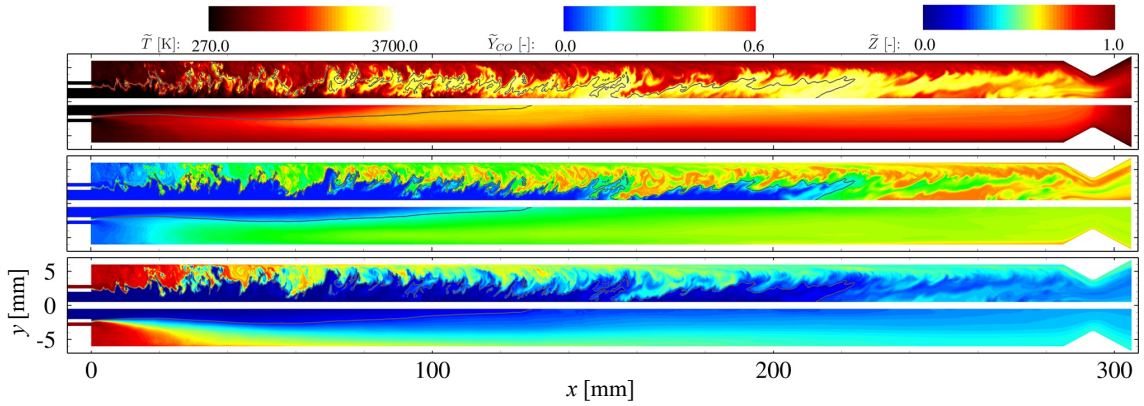
At the inlets, the fuel and oxidizer mass flow rates and temperature are prescribed following the experimental measurements [196]. At the chamber and nozzle walls, the temperature profile is defined as a Dirichlet boundary condition, which is obtained from the measurements by Perakis and Haidn [204]. The bottom and axisymmetric faces are prescribed with symmetry boundary conditions. All remaining boundaries are defined as adiabatic non-slip walls, except the exhaust (which is modeled as a pressure outlet). The computational domain is discretized by a block-structured mesh consisting of 2×10^5 cells. The wall-normal direction is resolved down to 30 μm, and a wall model [205] is employed for the viscous sublayer. Simulations are performed using 600 Intel Xeon (E5-2680v2) processors. The solution is advanced using a typical timestep of 25 ns, corresponding to a convective CFL number of 1.0.

5.3.3 Baseline Results from Monolithic Combustion LES

Simulations of the rocket combustor are first performed using monolithic FRC and monolithic FPV simulations. Flowfields are initialized with equilibrium products and temperature, thus allowing the monolithic FPV simulation to ignite. Instantaneous and time-averaged fields of temperature, CO mass fraction, and mixture fraction



(a) Monolithic FRC simulation.



(b) Monolithic FPV simulation.

Figure 5.2: Temperature, CO mass fraction, and mixture fraction fields (from top to bottom) for (a) monolithic FRC and (b) monolithic FPV simulations. Upper half: instantaneous fields, bottom half: time-averaged fields. The location of the stoichiometric mixture $\widetilde{Z}_{st} = 0.2$ is shown by black lines.

from monolithic FRC calculations and monolithic FPV simulations are shown in Figures 5.2a and 5.2b, respectively. Results from the FRC simulations are qualitatively similar to previous simulations [202, 206], where a non-uniform mixture fraction field, a long oxidizer core, and an agglomeration of cold rich gases to the chamber wall are observed. In contrast, some notable differences are observable from the FPV simulations, shown in Figure 5.2b. In particular, a thicker thermal boundary layer is seen for the FPV simulation. This difference is consistent with other LES studies [207]

which have shown that an adiabatic FPV model, as employed in the present study, mispredicts the wall-heat loss and exothermic CO-recombination in the boundary layer [206].

5.4 ML Methods

In the present data-driven framework, the procedure for incorporating a supervised learning algorithm (see Section 2.2.1) for combustion submodel assignment is as follows:

1. Generate data either from experimental measurements or numerical simulations. In this work, we use the instantaneous flowfield solutions from the FRC simulation of the GOX/GCH_4 rocket combustor as the learning dataset, discussed in Section 5.3.

2. Assign labels to the training data. Prior to training, each training data point is typically assigned a target response. In this work, we present a multi-class classification problem for optimal assignment of three combustion models with labels:

$$\Upsilon_k = \begin{cases} 1, & \text{if the sample belongs to the class } k \\ 0, & \text{otherwise} \end{cases}, \tag{5.1}$$

   where $k = 1$, 2, and 3 corresponds to IM, FPV, and FRC combustion models, respectively. Hence, we use the local combustion submodel error of two essential local QoIs, namely T and Y_{CO}, to programmatically assign labels. Details are presented in Section 5.4.1.

3. Construct the feature vector. In this work, we apply a feature selection method based on the Maximal Information Coefficient (MIC) [208], as discussed in Section 5.4.2, to construct a feature set consisting of local thermophysical quantities $\boldsymbol{\chi} = [\widetilde{Z}, \widetilde{C}, \overline{\rho}, \widetilde{T}, \mathrm{Pr}_\Delta, \|\nabla\widetilde{Z}\|_2]^\intercal$ that include the mixture fraction, progress variable, density, local Prandtl number, and Euclidean norm of the mixture fraction gradient for a given sample.

4. Train, validate, and test the classification algorithm. In this work, a random forest classifier (see Section 2.2.5) is used for combustion submodel assignment.

5.4.1 Label Assignment

We present a multi-class classification problem for optimal assignment of three combustion models $\mathcal{K} = \{\text{IM, FPV, FRC}\}$. In this problem, we consider the FRC model as the combustion model of highest fidelity but at the expense of highest computational cost. Hence, regions with local scalar predictions by IM and FPV models that match those of FRC can be considered optimally assigned. Therefore, we assign labels in the training set based on the normalized combustion submodel error $\epsilon^{\mathcal{K}}_{\boldsymbol{Q}}$ of QoI $\alpha \in \boldsymbol{Q}$ between FRC and the models of lower fidelity [68]:

$$\epsilon^{\mathcal{K}}_{\boldsymbol{Q}} = \sum_{\alpha \in \boldsymbol{Q}} \Theta_\alpha \frac{|\alpha^{\text{FRC}} - \alpha^{\mathcal{K}}|}{\|\alpha^{\text{FRC}}\|_\infty} \text{ with } \mathcal{K} \in \{\text{FPV}, \text{IM}\} \, , \tag{5.2}$$

where the error for considering N number of QoIs is a weighted linear combination of each individual submodel error. The weights for each QoI Θ_α are subject to the following constraints: $\sum_{\alpha \in \boldsymbol{Q}}^{N} \Theta_\alpha = 1$ and $\Theta_\alpha \geq 0$. In this study, the use of temperature and mass fractions of CO and OH as QoIs. In the combined use of both temperature and CO mass fraction, $\boldsymbol{Q} = \{\widetilde{T}, \widetilde{Y}_{\text{CO}}\}$, both QoIs are equally weighted: $\Theta_T = 0.5$ and $\Theta_{\text{CO}} = 0.5$. Similarly, for the combined use of three QoIs $\boldsymbol{Q} = \{\widetilde{T}, \widetilde{Y}_{\text{CO}}, \widetilde{Y}_{\text{OH}}\}$, all QoIs are equally weighted: $\Theta_T = 0.33$, $\Theta_{\text{CO}} = 0.33$, and $\Theta_{\text{OH}} = 0.33$. Temperature $\widetilde{T}$ is chosen as a proxy to describe the combustion efficiency and engine performance. The CO mass fraction $\widetilde{Y}_{\text{CO}}$ is chosen to challenge the deficiencies of tabulation methods in capturing intermediate species [69]. OH mass fraction $\widetilde{Y}_{\text{OH}}$ is selected since radical formation is essential in combustion phenomena.

FRC data is used to reconstruct FPV and IM QoIs $\alpha \in \boldsymbol{Q}$ by interpolating the generated flamelet tables using reconstructed values of mixture fraction and progress variable:

$$\alpha^{\mathcal{K}} \approx \mathcal{M}^{\mathcal{K}}_{\text{table}}(\widetilde{Z}_{\text{FRC}}, \widetilde{C}_{\text{FRC}}) \text{ where } \mathcal{K} \in \{\text{FPV}, \text{IM}\} \, . \tag{5.3}$$

The mixture fraction is computed using Bilger's definition [209], while the progress

variable is computed using the sum of major combustion products, as described in Section 5.2.2. We must note that since $\alpha^{\mathcal{K}}$ is reconstructed from FRC data, the resulting error metric $\epsilon_{\boldsymbol{Q}}^{\mathcal{K}}$ is an approximation of the true errors between FRC and tabulated chemistry. However, the use of this error metric is well-justified since Bilger's mixture fraction and the sum of major combustion products are robust quantities for bridging FRC and tabulated methods. Labels are assigned programmatically, as demonstrated in Algorithm 4. In this algorithm, a model of higher fidelity is assigned when the QoI submodel error $\epsilon_{\boldsymbol{Q}}^{\mathcal{K}}$ exceeds a user-defined threshold $\theta_{\boldsymbol{Q}}^{\mathcal{K}}$, with FRC chosen when all conditions for selecting FPV and IM are not met. While $\theta_{\boldsymbol{Q}}^{\mathrm{FPV}}$ and $\theta_{\boldsymbol{Q}}^{\mathrm{IM}}$ can be assigned distinct values, throughout this study we will explore cases that use the same threshold for both IM and FPV, *i.e.*, $\theta_{\boldsymbol{Q}}^{\mathrm{IM}} = \theta_{\boldsymbol{Q}}^{\mathrm{FPV}} = \theta_{\boldsymbol{Q}}$ for simplicity.

Algorithm 4 Assigning labels in the training set

```
if ε_Q^IM < θ_Q^IM then
    use inert mixing (IM)
else if ε_Q^FPV < θ_Q^FPV then
    use tabulated chemistry (FPV)
else
    use finite-rate chemistry (FRC)
end if
```

5.4.2 Feature Selection

Adding uninformative features to the learning dataset can reduce accuracy and computational efficiency of learning algorithms [107]. Carrying out appropriate feature selection beforehand can improve the interpretability of the predictions of the trained model. To this end, feature selection can be used for identifying the most descriptive and discriminative features from the raw dataset to use as inputs for our learning algorithms. In this work, we select features from local quantities and group parameters that can characterize the reacting flow, combustion state, and turbulence.

For feature selection, we rely on the Maximal Information-based Non-parametric Exploration (MINE) tools [208] that utilize mutual information between variable pairs to ascertain the strength of relationships between variables based on instantaneous

flowfield representations from a monolithic FRC simulation. MINE utilizes MIC to ensure (i) generality, where the association between the variables are not limited to a particular form such as linear associations, and (ii) equitability, where the effect of noise on different relationships is similar.

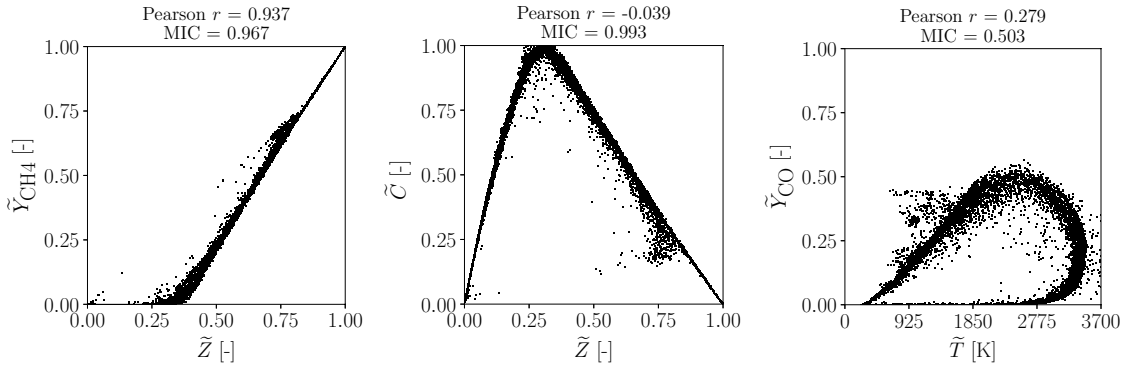
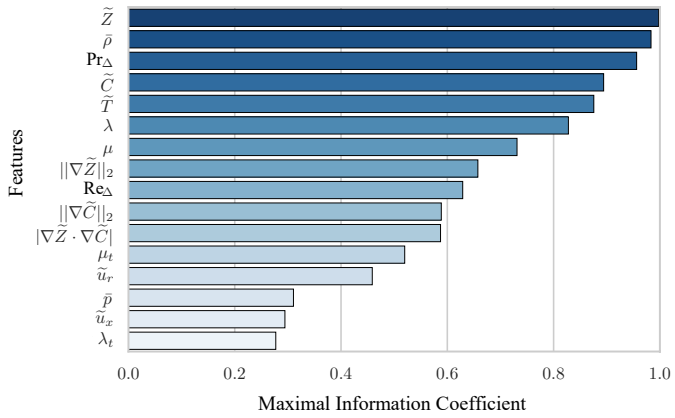


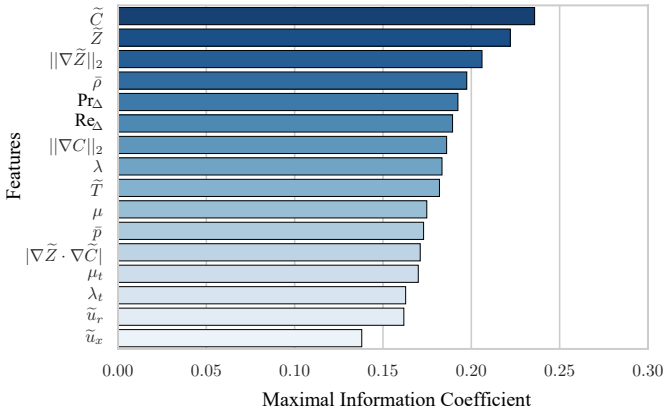
Figure 5.3: Comparison between Maximum Information Coefficient (MIC) and Pearson's Correlation Coefficient (Pearson r) for (a) near-linear scatter points, and (b,c) non-linear scatter points.

While Pearson's correlation has been utilized to ascertain the strength of relationships between variables in scientific applications, this does not account for any non-linear relationships. This is illustrated in Figure 5.3, where Pearson's coefficient, or Pearson r, is compared to MIC for different scatter points. As can be seen in Figure 5.3a, for linear relationships with noise, both coefficients are similar. However, in Figures 5.3b and 5.3c, non-linear associations between variables are ignored by Pearson's correlation coefficient while MIC is able to account for such complex relationships. Mutual-information-based measures that ensure generality and equitability, like MIC, can be used to compare different features, rank them and select subsets of the most descriptive and discriminative features. Additionally, such mutual information based feature selection is model agnostic and can be used across different ML models, as a pre-processing step. In this vein, MIC measure has been utilized for feature selection in prior works with success [210].

Figures 5.4a and 5.4b show MIC scores relating 16 potential features with IM



(a) MIC relating features with IM model error $\epsilon^{\mathrm{IM}}_{\{T,\mathrm{CO}\}}$.



(b) MIC relating features with FPV model error $\epsilon^{\mathrm{FPV}}_{\{T,\mathrm{CO}\}}$.

Figure 5.4: Maximal information coefficient score for features and model error.

model error $\epsilon^{\mathrm{IM}}_{\{T,\mathrm{CO}\}}$ and FPV model error $\epsilon^{\mathrm{FPV}}_{\{T,\mathrm{CO}\}}$, respectively. These 16 potential features consist of thermophysical quantities and dimensionless quantities that characterize each cell within the domain. Dimensionless quantities include the local Prandtl number, $\mathrm{Pr}_\Delta = \widetilde{\nu}/\widetilde{D}_T$, comparing the local ratio of viscosity and thermal diffusivity, and the local Reynolds number, $\mathrm{Re}_\Delta = \Delta|\widetilde{\boldsymbol{u}}|/\nu$, which is the ratio of inertial forces and viscous force within each cell and Δ denotes the characteristic length of each computational cell. It can be seen that the MIC scores for $\epsilon^{\mathrm{FPV}}_{\{T,\mathrm{CO}\}}$ are much lower than for $\epsilon^{\mathrm{IM}}_{\{T,\mathrm{CO}\}}$. This indicates that it is more challenging to form statistical relationships between features and FPV model errors than for IM model error. This

observation is consistent with the intuition that it is much easier to identify failure of the IM models than the shortfall of the FPV model.

In the following, the top five features from both MIC tests are used to construct the feature set consisting of mixture fraction, progress variable, density, local Prandtl number, and Euclidean norm of the mixture fraction gradient: $\boldsymbol{\chi} = [\widetilde{Z}, \widetilde{C}, \overline{\rho}, \widetilde{T}, \mathrm{Pr}_\Delta, \|\nabla \widetilde{Z}\|_2]^\intercal$. The inclusion of Pr_Δ in the feature set is unexpected, since Pr_Δ is approximately constant and has weak temperature dependence. However, given that Pr_Δ is slightly higher in fuel and oxidizer when compared to combustion products, small variations within flowfield prove useful for the random forests. We note that the data-driven framework in this study presently restricts the construction of feature and label sets to local quantities for simplicity.

5.4.3 Random Forest Classifier

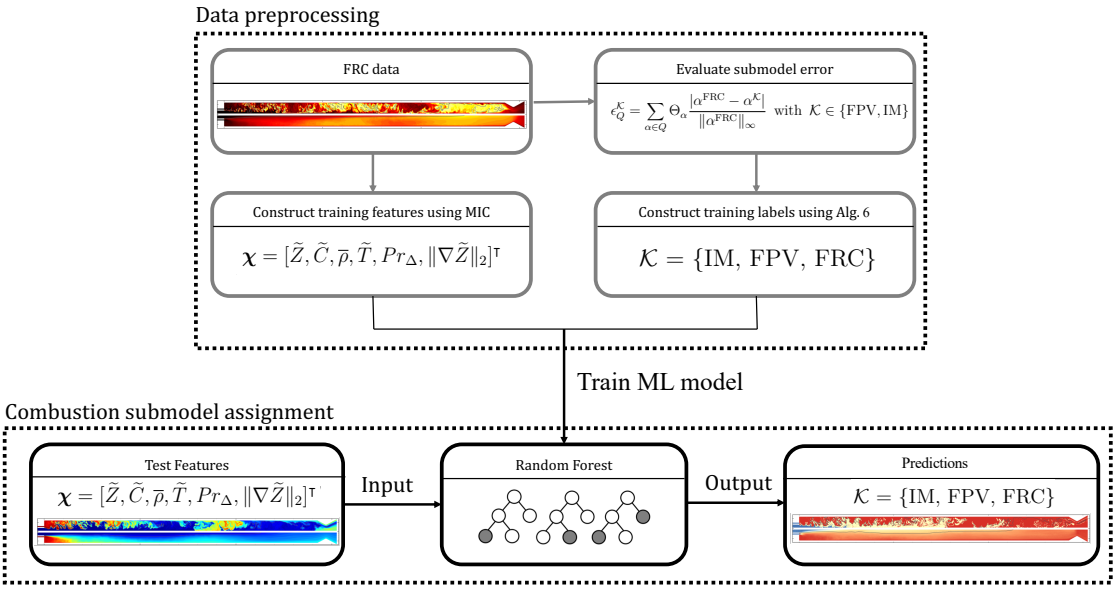


Figure 5.5: Application of random forest classifier for combustion submodel assignment of a single element GOX/GCH_4 rocket combustor.

In this study, we employ the random forest (see Section 2.2.5) as our classification algorithm. In the present investigation, the random forest classifier from the

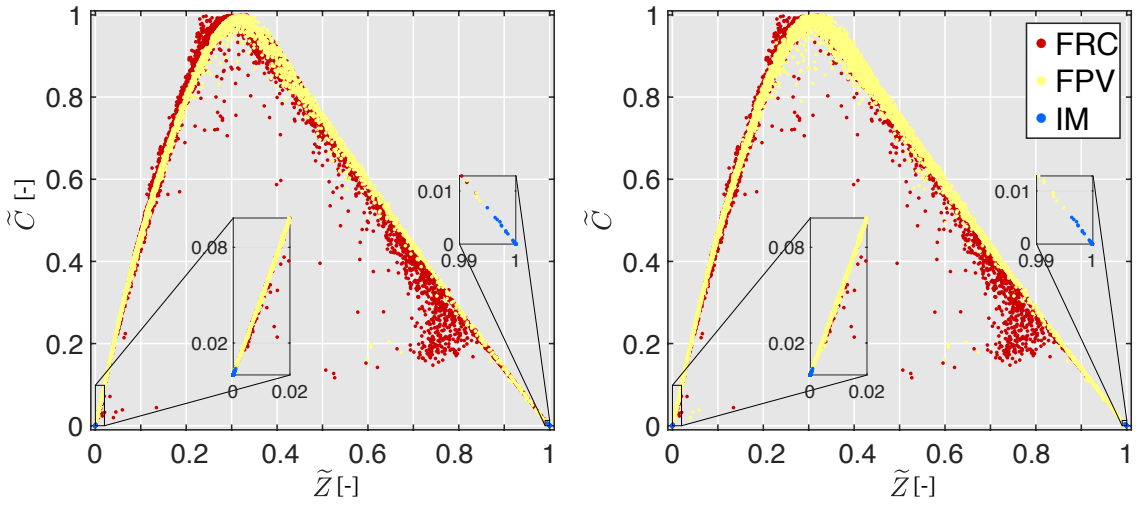
OpenCV library [211] is used. Classification cost scales with the number of trees, tree depth and the number of training points [24]. Hence, a random forest consisting of twenty decision trees, and maximum depth of ten nodes is employed. Additionally, 1×10^4 training points have been randomly sampled from a single LES snapshot consisting of 2×10^5 cells. A similar approach is used in other supervised learning problems [212]. We must note that the flow in the present configuration is statistically stationary, and thus training data from a single snapshot was found to be sufficient for representing the thermophysical behavior of the combustor. The number of trees, tree depth, and the number of training points are determined *a priori* by ensuring that the classification performance remains unchanged on a validation set. Training is performed once *a priori*, and requires 530 ms of walltime with 1 CPU. In *a posteriori* simulations, random forest evaluations for 2×10^5 cells at each timestep require 1 ms of walltime with 600 CPUs.

5.5 Results

This section assesses the random forest classifier as a method for combustion submodel assignment in DA simulations. *A priori* assessment is performed first to investigate the behavior of random forests when targeting different QoIs. This is followed by an *a posteriori* assessment to study improvements in target QoIs and other quantities that result from the use of random forests in transient DA simulations. Table 5.1 summarizes the eight cases, with different QoIs and combustion submodel error threshold values $\theta_{\boldsymbol{Q}}$, explored in both *a priori* and *a posteriori* assessment.

Table 5.1: Cases investigated in the present study.

Case	θ_T=0.05	θ_T=0.02	θ_{CO}=0.05	θ_{CO}=0.02	$\theta_{\{T,\mathrm{CO}\}}$=0.05	$\theta_{\{T,\mathrm{CO}\}}$=0.02	$\theta_{\{T,\mathrm{CO,OH}\}}$=0.05	$\theta_{\{T,\mathrm{CO,OH}\}}$=0.02
QoI, $\boldsymbol{Q}$	$\widetilde{T}$	$\widetilde{T}$	$\widetilde{Y}_{\mathrm{CO}}$	$\widetilde{Y}_{\mathrm{CO}}$	$\{\widetilde{T}, \widetilde{Y}_{\mathrm{CO}}\}$	$\{\widetilde{T}, \widetilde{Y}_{\mathrm{CO}}\}$	$\{\widetilde{T}, \widetilde{Y}_{\mathrm{CO}}, \widetilde{Y}_{\mathrm{OH}}\}$	$\{\widetilde{T}, \widetilde{Y}_{\mathrm{CO}}, \widetilde{Y}_{\mathrm{OH}}\}$
Model threshold, $\theta_{\boldsymbol{Q}}$	0.05	0.02	0.05	0.02	0.05	0.02	0.05	0.02
Assessment	*A priori*	*A priori*	*A priori*	*A priori*	*A priori*, *A posteriori*	*A priori*, *A posteriori*	*A priori*	*A priori*



(a) $\theta_{\{T,\mathrm{CO}\}} = 0.02$ (b) $\theta_{\{T,\mathrm{CO}\}} = 0.05$

Figure 5.6: Training data for two different combustion submodel error thresholds $\theta_{\{T,\mathrm{CO}\}}$.

5.5.1 *A priori* Assessment

A priori assessment involves using the random forest classifier to assign suitable combustion submodels in a test dataset that is created from a monolithic FRC simulation at an unseen timestep. Temperature and CO and OH mass fraction $\alpha \in \{\widetilde{T}, \widetilde{Y}_{\mathrm{CO}}, \widetilde{Y}_{\mathrm{OH}}\}$ in the test set is then used as QoI for reconstructing the true response, through the procedure described in Section 5.4.1, for comparison with random forest predictions. Figure 5.6 shows the use of this labeling approach on the training data in $\widetilde{Z}$-$\widetilde{C}$ composition space for $\theta_{\{T,\mathrm{CO}\}} = 0.02$ and $\theta_{\{T,\mathrm{CO}\}} = 0.05$, respectively. In both cases, IM is shown to be assigned at points where $\widetilde{C} \approx 0$, FPV is assigned mostly to conditions near the equilibrium composition. The submodel assignment reverts back to FRC in regions dominated by non-equilibrium effects and heat-losses that are not captured by the adiabatic steady-state flamelet formulation. Employing $\theta_{\{T,\mathrm{CO}\}} = 0.02$ is seen to be more stringent than employing $\theta_{\{T,\mathrm{CO}\}} = 0.05$, with a 0.18 greater fraction of scatter data on the stable branch assigned as FRC, especially for fuel-rich mixtures. It should be noted that while most out-of-flamelet regions would be assigned FRC, some regions with low reactivity and far from stoichiometry (*e.g.*, $\widetilde{Z} = 0.7$) generate

smaller errors which are then be assigned FPV.

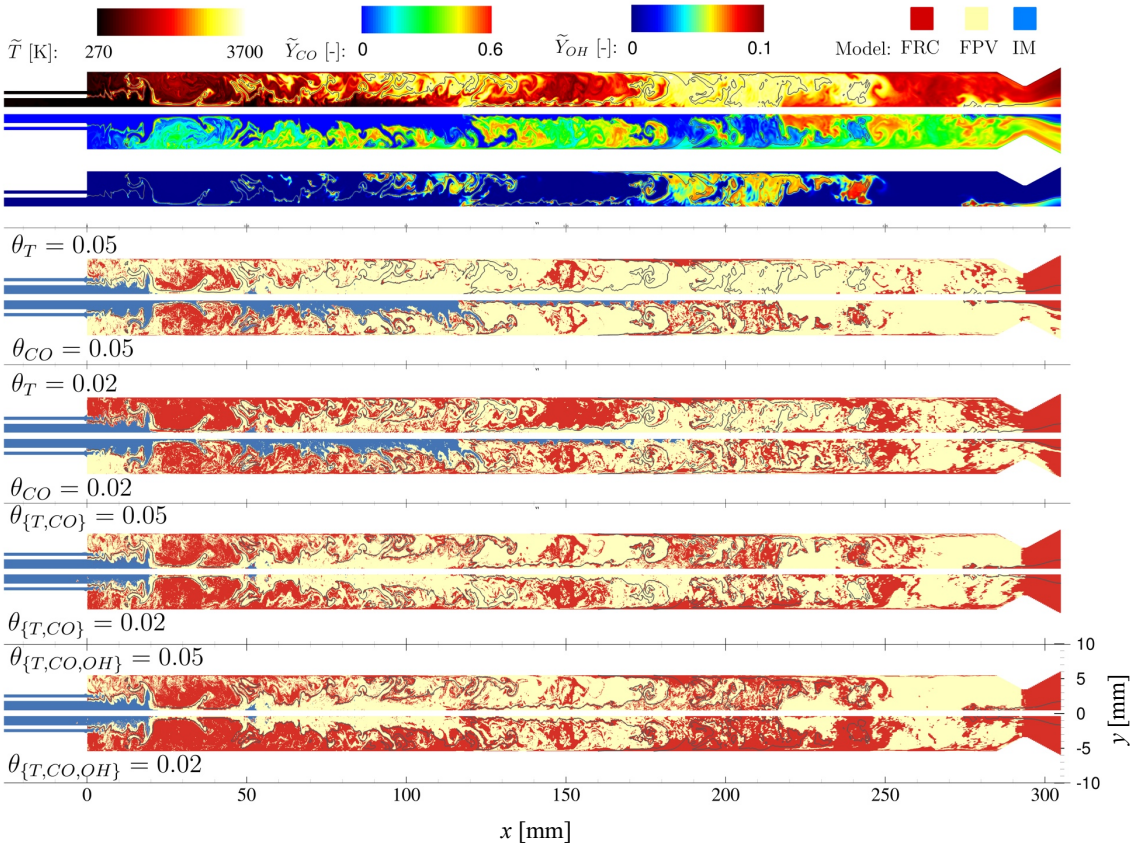


Figure 5.7: *A priori* analysis, comparing combustion model assignments. Instantaneous temperature, and mass fractions of CO and OH of the test set are also presented; stoichiometric isocontour with $\widetilde{Z}_{st} = 0.2$ is shown in black.

Figure 5.7 demonstrates the *a priori* combustion submodel assignment on an unseen FRC-simulation snapshot using the six different random forest cases summarized in Table 5.1. For all six cases, IM is assigned at the injector and the oxidizer core. In general, FRC is assigned to the near-wall and fuel-rich regions within the combustor, where intermediate reactions are not captured well by tabulated chemistry submodels. Using temperature as QoI and a model threshold of $\theta_T = 0.05$ results in an IM assignment of 5% of the domain, 28% FRC assignment, with the rest being described by the FPV model. Constraining the temperature model threshold $\theta_T = 0.02$ results in FRC assignment in 62% of the domain, with IM assignment remaining unchanged.

Using $\widetilde{Y}_{\mathrm{CO}}$ as QoI and a model threshold of $\theta_{\mathrm{CO}} = 0.05$ results in greater (18% of the domain) IM assignment, since the CO mass fraction in most of the oxidizer core is close to zero. FRC is assigned to 34% of the domain. Reducing the CO model threshold $\theta_{\mathrm{CO}} = 0.02$ results in 47% FRC assignment, with IM assignment unchanged. Finally, the combined use of both temperature and CO mass fraction as QoI, $\boldsymbol{Q} = \{\widetilde{T}, \widetilde{Y}_{\mathrm{CO}}\}$, results in submodel assignment with combined characteristics of employing each individual QoI. $\theta_{\{T,\mathrm{CO}\}} = 0.05$ results in 31% FRC assignment within the domain, while $\theta_{\{T,\mathrm{CO}\}} = 0.02$ results in 52% FRC assignment. Adding OH mass fraction to the QoI set $\boldsymbol{Q} = \{\widetilde{T}, \widetilde{Y}_{\mathrm{CO}}, \widetilde{Y}_{\mathrm{OH}}\}$ increases the FRC assignment to 37% and 70% for thresholds $\theta_{\{T,\mathrm{CO,OH}\}} = 0.05$ and $\theta_{\{T,\mathrm{CO,OH}\}} = 0.02$, respectively. Results demonstrate that reducing model threshold $\theta_{\boldsymbol{Q}}$ and increasing the number of QoIs increases submodel assignment of FRC. The submodel assignments for each case are summarized in Table 5.2.

Table 5.2: *A priori* analysis of classifier, summarizing submodel assignment and assignment accuracy.

Case	θ_T=0.05	θ_T=0.02	θ_{CO}=0.05	θ_{CO}=0.02	$\theta_{\{T,\mathrm{CO}\}}$=0.05	$\theta_{\{T,\mathrm{CO}\}}$=0.02	$\theta_{\{T,\mathrm{CO,OH}\}}$=0.05	$\theta_{\{T,\mathrm{CO,OH}\}}$=0.02
IM:FPV:FRC	5:67:28	5:33:62	18:48:34	18:35:47	6:63:31	6:42:52	6:57:37	6:24:70
True Classification	0.774	0.725	0.756	0.715	0.753	0.734	0.709	0.691

Table 5.2 also summarizes the true classification of random forests for the eight different cases. Here, true classification is defined as the percentage of classifier assignments that correctly match the true output responses evaluated directly from simulation data. The true classification fraction range from approximately 0.7 to 0.8, which is comparable to the use of random forests on another classification problem in a flow physics context [213]. Higher true classification can be achieved through the use of complex deep learning classifiers, which requires (i) more elaborate efforts than the random forests in hyperparameter tuning and (ii) much larger datasets for good performance, and should be subject to further study.

From Figure 5.7, we observe that model assignment in all six cases is not spatially smooth, and that model assignment appears speckled. This is because the smoothness of classification boundaries formed within the 6-dimensional feature space is not

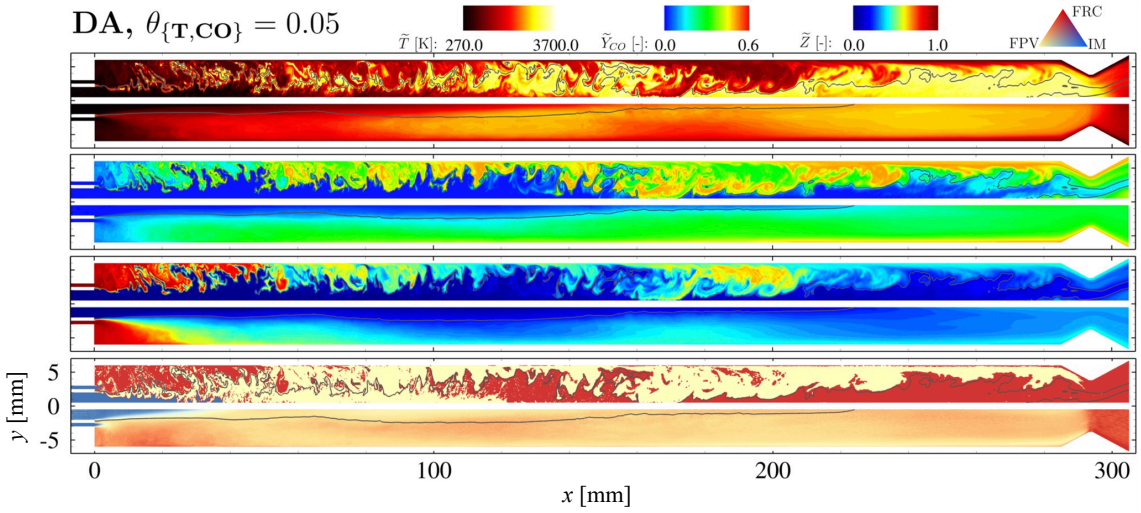
translated when transformed to physical space. This is a common issue in classification problems involving spatial data, such as in medical imaging or image processing. Two strategies can be employed to improve spatial smoothness in classification problems [68, 214]: (i) applying the classification techniques to a neighborhood of cells, or (ii) applying a spatial filter on the predicted labels and discretizing the filtered labels. In the *a posteriori* assessment in Section 5.5.2, we apply the latter strategy, since it is better suited with the current framework that uses local quantities as QoIs and features.

These results demonstrate that the present DA framework enables a fully adjustable level of simulation fidelity through the use of varying submodel error threshold values. Random forests are demonstrated to be a reasonably accurate and simple approach for the combustion submodel assignment problems.

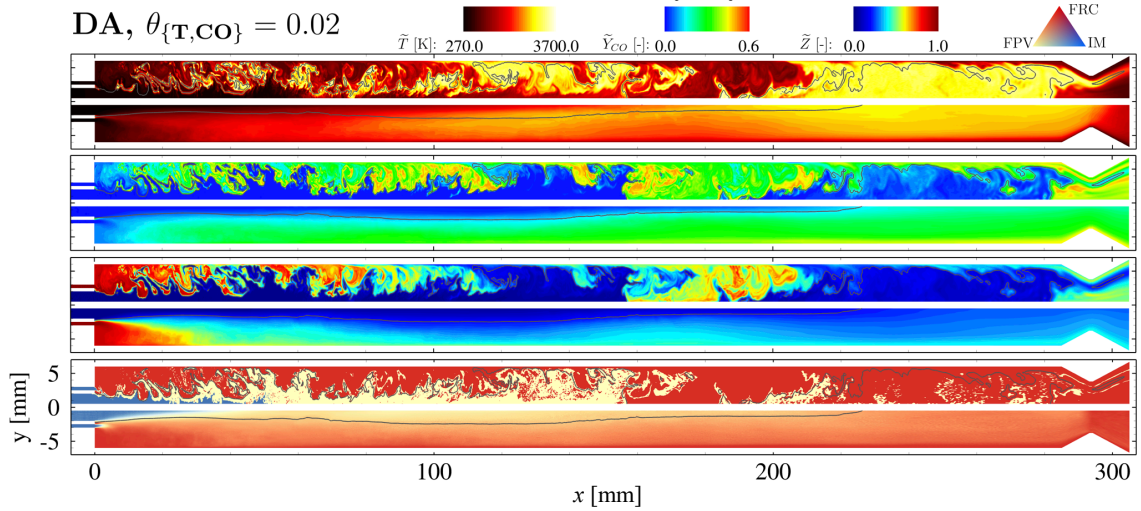
5.5.2 *A posteriori* Assessment: Data-assisted LES

DA simulations using two different model thresholds, $\theta_{\{T,\mathrm{CO}\}} = 0.05$ and $\theta_{\{T,\mathrm{CO}\}} = 0.02$ are performed by employing random forest classifiers in-flight during simulation runtime. The discussion from this section also includes comparisons with monolithic FRC and FPV simulations.

Figure 5.8a shows that employing model threshold $\theta_{\{T,\mathrm{CO}\}} = 0.05$ on the DA simulation results in temperature predictions that are in good agreement with the monolithic FRC simulation, shown in Figure 5.2a. However, time-averaged results show that a thin layer of CO develops at the chamber wall at 170 mm. Additionally, a thicker thermal boundary layer is also observed when compared to monolithic FRC simulations. Nonetheless, both species and thermal boundary layers are thinner than the monolithic FPV simulations that were presented in Figure 5.2b. Averaged FRC utilization with $\theta_{\{T,\mathrm{CO}\}} = 0.05$ is at 34% of the domain, with IM-utilization at 4%. In addition, a thin intermittent area close to the wall is also assigned FRC. This indicates that the random forest recognizes the importance of wall effects on CO and temperature but that the user-defined model error threshold $\theta_{\{T,\mathrm{CO}\}} = 0.05$ is too large.



(a) *A posteriori* DA LES with $\theta_{\{T,\mathrm{CO}\}} = 0.05$.

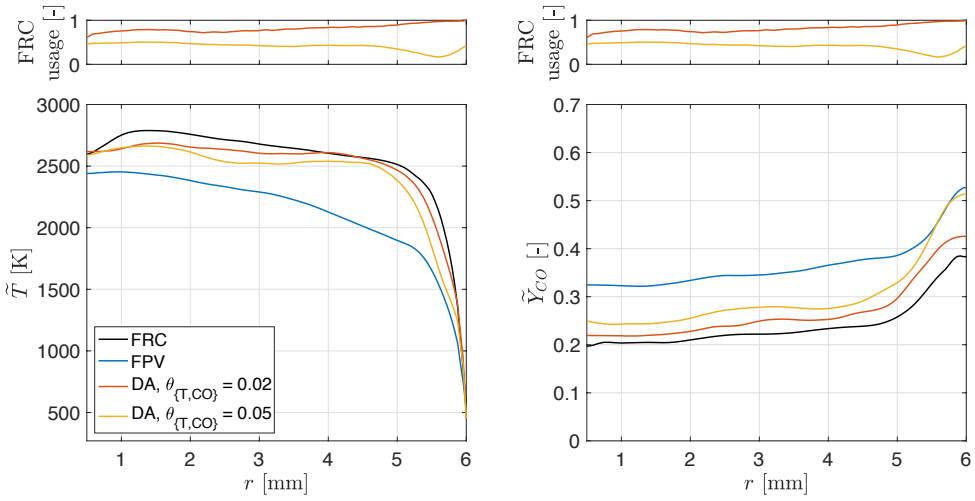


(b) *A posteriori* DA LES with $\theta_{\{T,\mathrm{CO}\}} = 0.02$.

Figure 5.8: Temperature, CO mass fraction, and mixture fraction fields (from top to bottom) from *a posteriori* DA LES for (a) $\theta_{\{T,\mathrm{CO}\}} = 0.05$ and (b) $\theta_{\{T,\mathrm{CO}\}} = 0.02$. Upper half: instantaneous fields, bottom half: time-averaged fields; stoichiometric isocontour with $\tilde{Z}_{st} = 0.2$ is shown in black.

Figure 5.8b shows that tightening the model threshold $\theta_{\{T,\mathrm{CO}\}} = 0.02$ results in temperature, CO, and mixture fraction fields that agree with the monolithic FRC simulation, shown in Figure 5.2a. Model assignment using this threshold results in 60% FRC utilization. Before $x = 150$ mm FRC is assigned to all fuel-rich and

near-wall regions. For $x > 150$ mm, FRC is assigned to most of the domain where incomplete combustion products and intermediate species are dominant.



(a) Temperature (b) CO mass fraction

Figure 5.9: Comparisons of time-averaged radial profiles of (a) temperature and (b) CO mass fraction between monolithic FRC, monolithic FPV, and data-assisted (DA) simulations at an axial distance $x = 250$ mm. Time-averaged utilization of FRC is included.

Figure 5.9 shows comparisons of radial profiles of time-averaged temperature and CO mass fraction at an axial distance of 250 mm. Effects of wall-heat loss on the monolithic FPV simulation is seen to reduce the overall temperature and thicken the thermal boundary layer, which in turn results in greater CO mass fraction. Using a model threshold of $\theta_{\{T,\mathrm{CO}\}} = 0.05$, DA-predictions for temperature and CO mass fraction profiles away from the wall are in good agreement with monolithic FRC simulations, and averaged FRC submodel utilization ranges between 16% and 38%. At $r = 5$ mm, the random forest is able to recognize when the absolute error between temperature diminishes and thus assigns less FRC accordingly, which results in greater temperature and CO mass fraction deviation from monolithic FRC simulations. After $r = 5.7$ mm, the random forest begins to recognize the importance of near-wall effects and assigns more FRC. However, this FRC utilization is still insufficient for recreating monolithic FRC simulations. Further constraining the DA-simulation threshold

to $\theta_{\{T,\mathrm{CO}\}} = 0.02$ improves the agreement with monolithic FRC-simulations. However, small errors can still be seen even with high FRC submodel utilization that ranges from 61% to 90%.

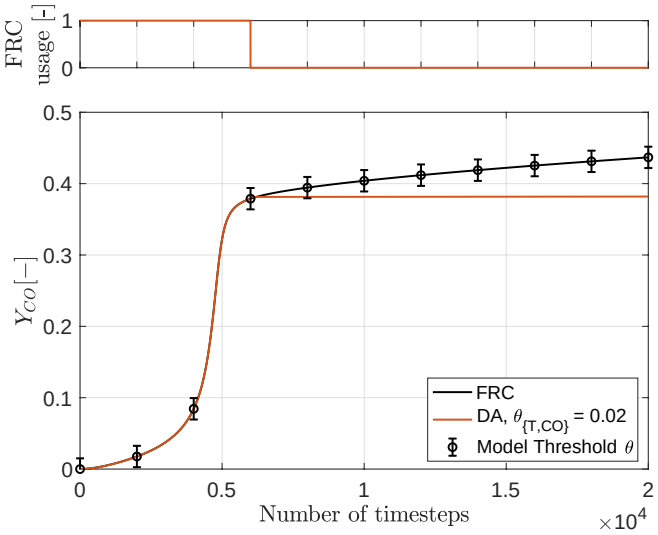
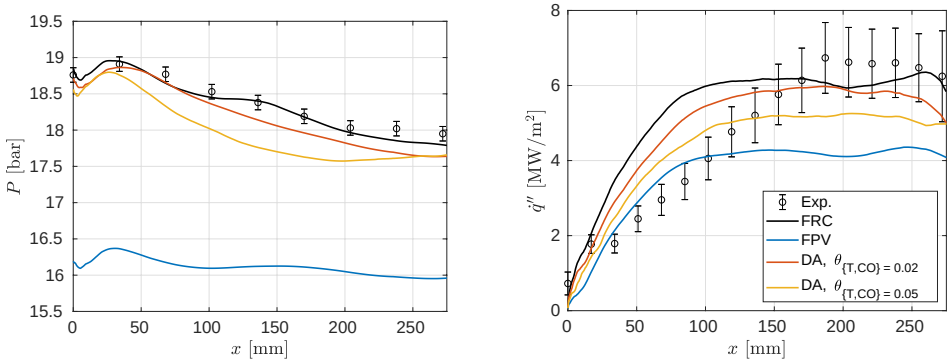


Figure 5.10: FRC and DA-assisted calculation of CO mass fraction as a function of timestep in a 0D homogeneous reactor.

Results from Figure 5.9 show that the present DA modeling approach can generate simulation results that are in agreement with monolithic FRC calculations. However, errors observed are greater than the local model error threshold $\theta_{\{T,\mathrm{CO}\}}$ used for training the random forests. This is caused by small changes in one state that can result in significant deviations in later states. This effect is illustrated by applying DA combustion modeling with local model error threshold $\theta_{\{T,\mathrm{CO}\}} = 0.02$ on CO mass fraction, using a rich CH_4/air mixture ($Z = 0.55$) in a constant pressure homogeneous reactor at 20 bar and initial temperature of 1800 K, as shown in Figure 5.10. In this setup, it is observed that while the random forest correctly assigns the correct model based on local model error at 5800 timesteps, the CO trajectory leads to a total error exceeding the local error threshold of 0.02 as the DA simulation no longer has knowledge of the monolithic FRC CO production beyond this timestep and cannot recover to the correct state. However, the benefit of the present approach is that, in the worst-case, errors made do not exceed errors made by the lowest fidelity

combustion model employed.

Generating numerical predictions that match experimental wall measurements are challenging for this rocket combustor case, since these quantities are dependent on overall flow and temperature fields in a highly nonlinear system. Studies [206, 215] comparing LES and RANS results have reported up to 8% deviation from wall pressure measurements. Wall heat flux predictions are more sensitive to simulation parameters, where deviations up to 75% have been reported in the same studies. While the aim of the present study is not to find simulation results that match the experimental results, LES calculations of wall pressure and wall heat flux are presented with measurements by Perakis and Haidn [204] in Figure 5.11 to quantify effects of applying the DA formulation on overall combustor behavior.



(a) Wall pressure (b) Wall heat flux

Figure 5.11: Comparison of simulation results for (a) wall pressure and (b) wall heat flux calculations with experimental measurements [204].

Figure 5.11a shows that wall pressure predictions between monolithic FRC agree well with experimental measurements. The DA simulation with $\theta_{\{T,\mathrm{CO}\}} = 0.02$ shows a small underprediction, but still possesses reasonable agreement with monolithic FRC. The DA simulation with $\theta_{\{T,\mathrm{CO}\}} = 0.05$ shows a greater underprediction. Wall pressure underprediction can be caused by reduced fuel conversion [216]. This is likely the case, since higher CO levels in both cases are observed in Figure 5.9. Additionally, the monolithic FPV simulation also demonstrates the lowest pressure and highest CO levels.

Figure 5.11b shows that wall heat flux predictions for FRC simulation are in good agreement with experimental data after $x = 120$ mm, but with a steeper heat flux rise. This steep heat flux rise is likely due to the misrepresentation of turbulent mixing in a thin axisymmetric domain, and is also seen in other axisymmetric studies [202, 203]. Tightening the model threshold $\theta_{\{T,\mathrm{CO}\}}$ results in better convergence with monoolithic FRC calculations. The DA simulation with $\theta_{\{T,\mathrm{CO}\}} = 0.02$ is in reasonable agreement with the FRC simulation, while the FPV simulation demonstrates the lowest heat flux due to low overall temperatures from low combustion efficiency.

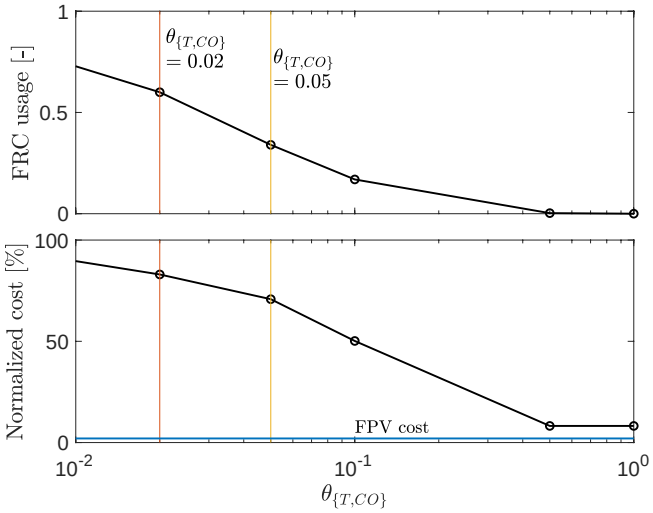


Figure 5.12: FRC utilization and normalized computational cost versus combustion submodel error threshold $\theta_{\{T,\mathrm{CO}\}}$.

Figure 5.12 shows FRC usage and corresponding computational cost (normalized by FRC cost) of the DA simulation as a function of combustion submodel error threshold $\theta_{\{T,\mathrm{CO}\}}$ when computed using 600 Intel Xeon (E5-2680v2) processors. Each timestep in the FPV simulation requires 50 ms of walltime to solve, while each timestep in the FRC requires a walltime of 2,300 ms. When $\theta_{\{T,\mathrm{CO}\}} = 0.50$, the classifier does not assign FRC in the entire domain, resulting in a normalized cost of 8%. This additional cost represents the overhead from the random forest evaluation and the coupling of the three combustion submodels in the same domain. Simulations performed in this study utilized 34% ($\theta_{\{T,\mathrm{CO}\}} = 0.05$) and 60% FRC ($\theta_{\{T,\mathrm{CO}\}} = 0.02$),

which resulted in 70% and 80% of FRC cost, respectively. These results demonstrate that classification algorithms can be utilized in high-fidelity simulations to reduce computational cost. Further reductions of the computational cost are achievable by combining the method proposed in this work with regression techniques [78, 79] to reduce the complexity of the FRC representation.

5.5.3 Generalization

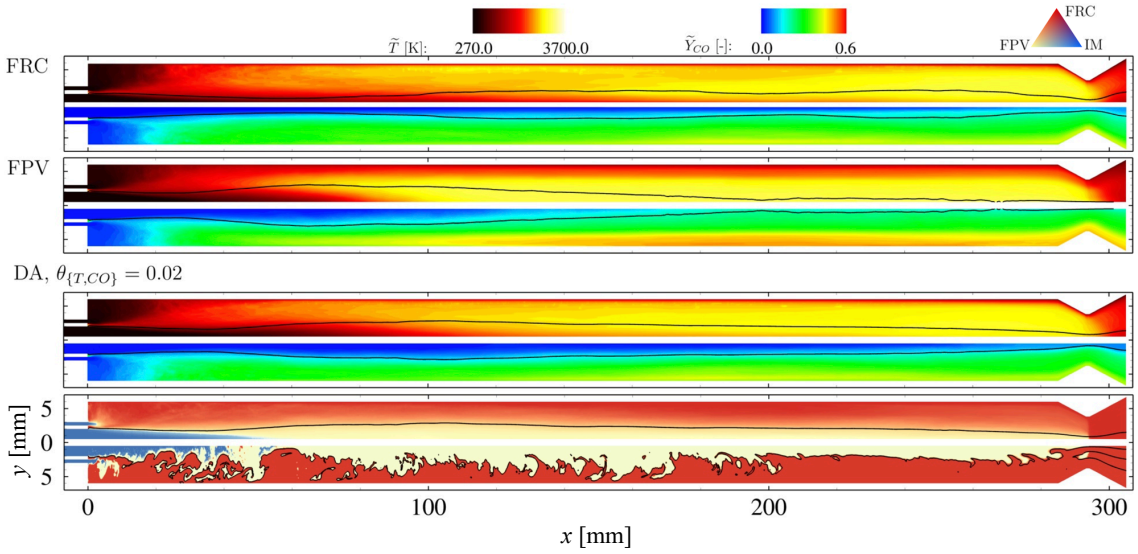


Figure 5.13: Comparison of time-averaged temperature and CO mass fraction fields for monolithic FRC, monolithic FPV, and *a posteriori* DA LES ($\theta_{\{T,\mathrm{CO}\}} = 0.02$) on a configuration with three times the inlet mass flow rate. Time-averaged and instantaneous model assignment for DA LES is shown at the bottom. Stoichiometric isocontour with $\widetilde{Z}_{st} = 0.2$ is shown in black.

In order to demonstrate the ability of random forests to generalize, additional LES are performed on a modified configuration with three times the inlet mass flow, while keeping all other parameters constant. Figure 5.13 compares time-averaged temperature and CO mass fraction fields for monolithic FRC, monolithic FPV, and *a posteriori* DA LES ($\theta_{\{T,\mathrm{CO}\}} = 0.02$) for this setup. All three LES cases in this modified configuration demonstrate a longer oxidizer core than the original configuration (Figure 5.2) due to higher flow velocity, indicating less complete combustion. When

compared to FRC, FPV overpredicts the thickness of the thermal boundary layer and CO formation. DA LES with model threshold ($\theta_{\{T,\mathrm{CO}\}} = 0.02$) predicts temperature and CO flowfields in good agreement with monolithic FRC calculations. Random forest assigns FPV to the lean side of the flame, while assigning FRC to the rich side. This is also seen in the DA case of the original configuration in Figure 5.8 from 0 to 150 mm, where major combustion products have not fully formed. Model assignment using this threshold results in 51% FRC and 6% IM utilization, resulting in 77% of the FRC cost.

Results from this modified configuration demonstrate that the present DA approach can be applied to different configurations as long as the training data can represent the underlying thermophysical behavior. We note that all simulations and training data from the present study employ the same mesh. Since the random forest classifies well in this modified configuration, this method should still be effective for different mesh resolutions as long as the flow can be represented by local points of the training data. The generalizability of this method improves with increasing availability of representative data.

5.6 Summary

This chapter introduced a DA modeling approach, employing random forest classifiers, as a method for dynamic and local combustion model assignment in reacting flow simulations. *A priori* assessment was conducted on the random forests, which were fed with six input features based on local thermofluid properties, to evaluate the behavior of the classifiers during submodel assignment when targeting different QoIs. Random forests were shown to assign three different candidate combustion models – FRC, FPV, and IM – based on predefined QoIs with fraction of true classification ranging from approximately 0.70 to 0.80.

Two cases of *a posteriori* simulations using random forest classifiers for combustion submodel assignment during simulation runtime, were performed. Time-averaged results of temperature and CO mass fraction demonstrated that the DA simulation produced species and temperature profiles in better agreement with monolithic FRC

than monolithic FPV calculations. The use of the random forest with submodel error threshold of $\theta_{\{T,\mathrm{CO}\}} = 0.02$ results in significant improvements from monolithic FPV simulations in all quantities at a 20% lower cost than monolithic FRC calculations. An additional DA LES ($\theta_{\{T,\mathrm{CO}\}} = 0.02$), performed on a modified configuration with three times the inlet mass flow rate, demonstrated that the present approach can be applied to different configurations as long as the training data can represent the relevant thermophysical behavior.

Results from this chapter demonstrate that integration between ML models and numerical solvers can benefit from domain knowledge in managing OOD errors. The resulting integrated framework shows promise as a tool for managing fidelity-cost trade-offs in high-fidelity simulations.

Chapter 6

Hybrid Physics-Machine Learning Model for Laser Ignition*

6.1 Introduction

The previous chapter involves the integration of ML methods with numerical methods for simulating a model rocket combustor configuration. Similarly, this chapter examines opportunities for integrating deep learning with an SDE for modeling laser ignition within rocket combustors. As discussed in Section 1.2, laser-ignited propulsion systems often require ensemble measurements or simulations for a robust understanding of the system behavior. Since these ensemble datasets can be costly to collect, this chapter focuses on identifying opportunities for modeling laser ignition behavior with sparse ensemble datasets.

To this end, we introduce a reduced-order physics-embedded SDE-ML framework for spatio-temporal modeling of laser ignition by integrating an SDE for modeling kernel dynamics with a deep learning model trained solely for representing ignition kernel morphology. We evaluate this approach for modeling ignition within the gaseous

*This chapter contains work accepted for publication from Chung et al. [10], with minor modifications. W.T. Chung planned, performed, and analyzed experiments, and developed modeling techniques. C. Laurént assisted with planning experiments and developing model techniques. D. Passiatore performed simulations.

CH_4/O_2 model rocket combustor [86] detailed in Section 6.2, while the methods employed within this work are introduced in Section 6.3. We discuss results from this work in Section 6.4, before providing concluding remarks in Section 6.5.

6.2 Configuration

In this work, we model ignition within the gaseous CH_4/O_2 model rocket combustor by Strelau et al. [86]. This optically accessible configuration was designed specifically for statistically characterizing laser ignition phenomena of gaseous mixtures within rockets. The experimental configuration consists of a shear co-axial injector where the oxidizer flows through a central axisymmetric jet with diameter $d_o = 3.57\,\text{mm}$, while fuel is injected through an annulus with inner and outer diameters $d_{f,i} = 5.33\,\text{mm}$ and $d_{f,o} = 6.35\,\text{mm}$. The cylindrical combustion chamber has a total length of 111 mm and a diameter of $d_{ch} = 50.8\,\text{mm}$. The combustor operates at a global oxidizer-to-fuel ratio of approximately 3, with mass flow rates of oxidizer $\dot{m}_o = 6.58\,\text{g/s}$ and fuel $\dot{m}_f = 2.11\,\text{g/s}$ that correspond to sonic and subsonic conditions, respectively. The temperature of the oxidizer and the fuel supplied at the injector inlet are $T_o = 242\,\text{K}$ and $T_f = 282\,\text{K}$, respectively. Prior to ignition, the reactants are injected to pressurize the chamber until reaching a quasi-steady-state with a nominal operating pressure of 1.4 bar. The laser is then deployed near this non-premixed mixture via an Innolas Spitlight Standard 600-10 laser in single-shot mode, which generated a 532 nm laser pulse with laser energy of $E_{laser} = 22 \pm 7\,\text{mJ}$.

In this work, we generate datasets for developing and testing the SDE-ML model. In particular, we process Schlieren measurements of $N_{exp} = 153$ laser ignition tests, across 21 laser deposition locations in the jet centerplane. The measurements are recorded at an acquisition rate of 500 kHz. Approximately five ignition tests were performed in laser locations where ignition probability $P_{ig} = 0$ and $P_{ig} = 1$, resulting in an estimated uncertainty of ± 0.2. For laser locations where ignition probability $0 < P_{ig} < 1$, approximately ten ignition tests were performed, resulting in an estimated uncertainty of ± 0.1.

Previous studies [94, 95] have noted that reduced-order models for forced ignition phenomena can be constructed with inputs from inert simulations to represent flows during the early stages of ignition – where dilational effects from heat release are not yet relevant. A non-reacting LES is calculated with a high-order finite-difference compressible flow solver [217]. For spatial discretization, a sixth-order skew-symmetric scheme is employed with a targeted essentially non-oscillatory scheme [218] for shock-capturing, while time advancement is performed via a third-order SSP-RK3 method [219]. LES closure is provided via the Smagorinsky model [45], with turbulent Prandtl and Schmidt numbers of 0.7. Multi-component species transport [220] is employed to represent molecular mixing within this configuration. This LES was advanced with a timestep size/of 16 ns, corresponding to an approximate acoustic CFL = 0.1, on 96 NVIDIA V100 GPUs with a wall-clock-time of 0.6 s per timestep.

The LES domain consists of a single-block curvilinear mesh with 221M ($960\times480\times480$) mesh points, which is stretched in all three directions. The finest mesh spacing is at the injector exit, where the mesh is locally uniform with cell size of 88 μm, was chosen by examining the sensitivity of LES predictions of shock train locations across different mesh resolutions. These shock train locations, which have previously been used to validate jet topology of rocket combustors [85], were determined by observing the presence of peaks in pixel intensity and density magnitudes in the centerline of time-averaged measurements and LES, respectively. Figure 6.1 compares instantaneous experimental Schlieren and simulated density fields, with reasonable agreement seen between the visible shock train locations in the predictions. We note that one of the key challenges presented by this complex high-speed multi-physics flow configuration is the development of a data-driven model that is sufficiently robust to the limited information provided by experimental measurements, as well as discrepancies between simulation and measured data.

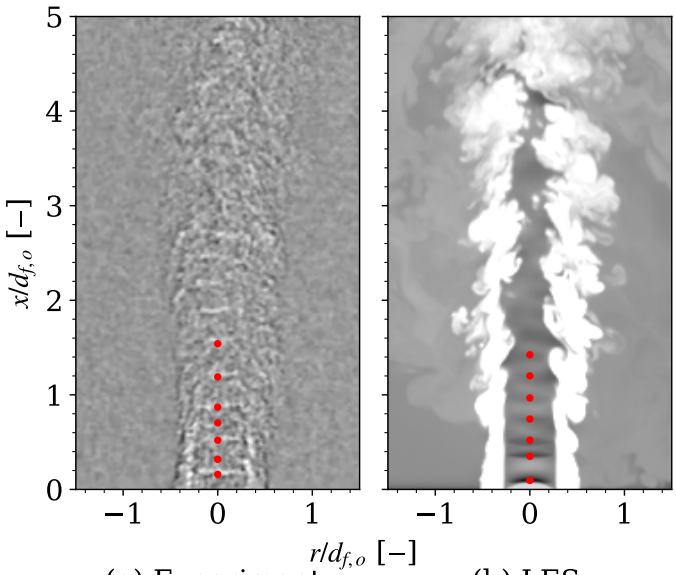


Figure 6.1: Instantaneous (a) experimental Schlieren measurements [86] with (b) density fields from the present LES.

6.3 Methods

6.3.1 SDE-ML Framework

In this section, we describe the 2D SDE-ML approach for utilizing (i) both 3D LES and 2D experimental data for modeling kernel morphology via the deep learning approach, and (ii) LES statistics for informing SDE-based kernel transport, as summarized by Figure 6.2. Ensemble calculations can be performed with this approach to evaluate statistical behavior of laser ignition in the present configuration.

In this work, we represent space- and time-dependent kernel morphology $\Upsilon^n(\mathbf{x}, t^n)$ as a binary flowfield of ignited and non-ignited segments. Thus, the spatio-temporal evolution of kernel morphology for timestep n is expressed as:

$$\Upsilon^n(\mathbf{x}, t^n) = \begin{cases} 1 & \text{ignited cell at } t^n\,, \\ 0 & \text{non-ignited cell at } t^n\,. \end{cases} \tag{6.1}$$

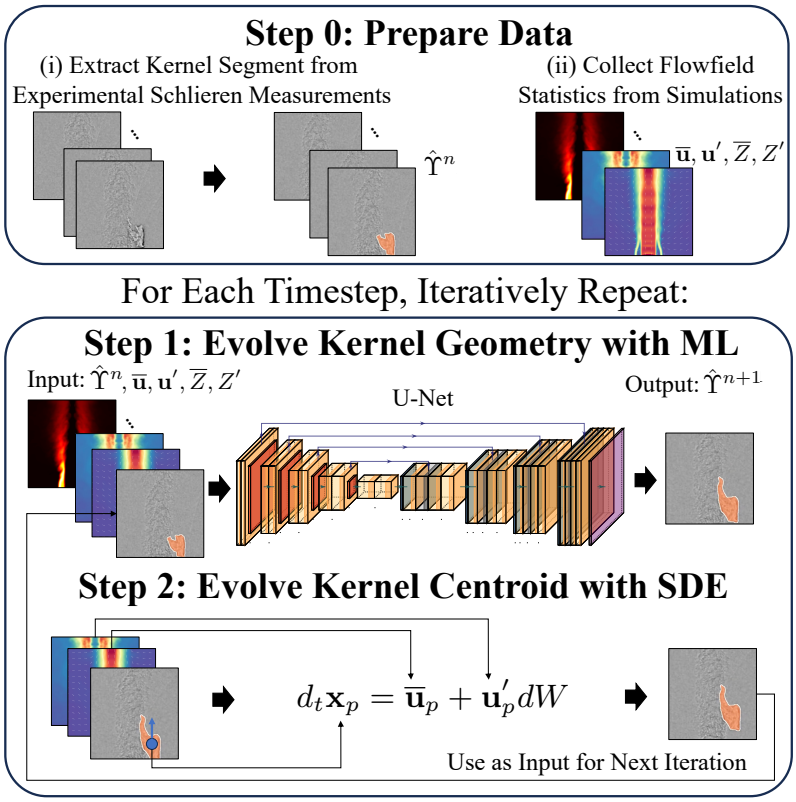


Figure 6.2: SDE-ML framework for modeling stochastic ignition.

To obtain this binary flowfield, we process Schlieren measurements of the present configuration. We note that the proposed SDE-ML method is agnostic to the source of kernel morphology data, and could be extended to employ data from high-fidelity reacting LES. However, ML models typically perform more accurately with increasing data diversity and volume [17]. Thus, we employ already available data from the 153 experimental tests to demonstrate the present framework. In future work, this data scarcity could be addressed by extending the present framework with multi-fidelity training [221] that combines small samples of high-fidelity reacting LES with large samples of more affordable and lower-fidelity simulation data.

The use of conventional image processing techniques, involving spatio-temporal filters and edge detection algorithms, can be difficult for this specific configuration due to (i) differences in time-scales encountered between direct and indirect ignition, and

(ii) measured structures from the jet obstructing ignition kernel structure. Thus, we employ an open-source video segmentation tool [222] for extracting a total of 19,765 frames of kernel morphology segments Υ^n (partially shown in orange in Figure 6.2). Each ignition test case is segmented starting from time after laser deposition $\tau = t - t_{laser} = 4\,\mu$s to avoid imaging artifacts that arise from the pressure wave that forms immediately after laser deposition. Each of the frames of the kernel segments is manually inspected for quality prior to ML training.

An ML model f_{ML} is employed for autoregressively generating next-timestep predictions $\hat{\Upsilon}^{n+1}$ of kernel morphology via segmentation, which classifies flowfield segments as ignited and non-ignited via:

$$\hat{\Upsilon}^{n+1}(\mathbf{x}, t^{n+1}) = f_{ML}(\hat{\Upsilon}^n, \overline{\mathbf{u}}, \mathbf{u}', \overline{Z}, Z')\,, \tag{6.2}$$

with an initial condition $\hat{\Upsilon}^0 \equiv \Upsilon^0$. The ML inputs consist of stacked channels of (i) mean $\bar{\cdot}$ and fluctuating $\cdot'$ components of velocity $\mathbf{u} = [\widetilde{u}_x, \widetilde{u}_r]^\intercal$ and mixture fraction Z extracted from temporal- and azimuthal-averaged flowfields over $320\,\mu$s (representative of the time taken after laser deposition to transition to a flame), as well as (ii) the kernel segments from the previous timestep. These seven inputs are stacked together with axial, transverse, and channel dimensions of $N_x \times N_r \times N_c = 250 \times 160 \times 7$ that can be fed into a CNN. To predict the evolution of kernel morphology, we employ a U-Net [117] model (shown in Figure 6.2) – a well-established deep learning architecture suited for segmentation tasks. This CNN uses a small moving window, known as a filter, that performs a mathematical operation (typically convolution and pooling operations [6]) on a spatial neighborhood of values (in the first CNN layer containing $[\hat{\Upsilon}^n, \overline{\mathbf{u}}, \mathbf{u}', \overline{Z}, Z']^\intercal$), thereby enabling higher prediction accuracy in spatial problems compared to other ML algorithms. A key aspect of this model design involves the use of skip connections that connect processed features from the later hidden layers with unprocessed features from the earlier layers – which helps in preserving fine-grained details during segmentation.

When generating ML predictions, CNN filters at the input layer supplement the kernel segment with surrounding flowfield information from the inert LES statistics.

Even though the LES flowfield remains the same across data samples, the flowfields vary from the perspective of the convolutional filter and provide local mixture information for the CNN window containing an ignited kernel segment. If these channels containing local mixture information were not present, the CNN would not have sufficient information to propagate or extinguish the ignition kernel, since there would be no inputs that inform the CNN on the proximity of the ignition kernel to the reactive/non-reactive mixtures. This choice of ML inputs is motivated by previous work involving rule-based and analytic reduced-order ignition models [94, 95]. The computational domain ($N_x \times N_r = 250 \times 160$) for this reduced-order model is discretized with a uniform mesh size of $180\,\mu$m in both axial and radial directions. This grid size was selected by considering limitations in computational memory during multi-GPU training of the ML model, and is similar in size with the inert LES grid described in Section 6.2.

To model stochastic variations of different ignition trajectories, we treat the ignition kernel centroid as a Lagrangian particle with position $\mathbf{x}_p$ that is advanced via the following SDE:

$$d_t \mathbf{x}_p = \overline{\mathbf{u}}_p + \mathbf{u}'_p dW\,, \tag{6.3a}$$

$$\mathbf{u}_p = \textstyle\int \hat{\Upsilon}\mathbf{u}\, d\hat{\Upsilon} / \int \hat{\Upsilon}\, d\hat{\Upsilon}\,, \tag{6.3b}$$

$$\mathbf{x}_p = \textstyle\int \hat{\Upsilon}\mathbf{x}\, d\hat{\Upsilon} / \int \hat{\Upsilon}\, d\hat{\Upsilon}\,, \tag{6.3c}$$

where a Gaussian distribution $\mathcal{N}(0, \sigma)$ introduces turbulent fluctuations via a Wiener process dW. The Gaussian standard deviation $\sigma = 0.125$ was determined with a hyperparameter search on a validation set discussed in Section 6.3.2. Both mean and fluctuating velocity components of the particle are approximated by averaging the velocity field that intersects with the area of kernel segments. Equation (6.3a) is advanced via a forward Euler scheme with a timestep of $\Delta t_{\text{SDE}} = \Delta t_{ML} = 8\,\mu s$, which enables the kernel centroid to be transported across the different cells in the SDE-ML grid, while ensuring that the number of Schlieren samples are sufficiently large ($\mathcal{O}(10^4)$) for training a deep learning model. During position updates, the ML channel with the kernel segment $\hat{\Upsilon}^n$ is updated with the new kernel $\hat{\Upsilon}^{n+1}$, while preserving

the other channels (since flowfield statistics are assumed to be time-independent), before repeating the ML and SDE steps for the next timestep iteration. We note that prior to SDE advancement, all ignited cells are spatially translated by $\Delta\mathbf{x}_{ML} = \mathbf{x}_p(\Upsilon^{n+1}) - \mathbf{x}_p(\Upsilon^n)$ so that kernel transport is governed solely by the SDE without advective effects from the ML model. A wall-clock-time of 0.3 s with one V100 GPU generates a single ignition trajectory with 128 timesteps that compounds to a model duration of 1 ms.

6.3.2 ML Setup

The present U-Net contains 82 layers, corresponding to approximately 31M trainable parameters, with the initial weights set via He initialization [112]. During training, the Adam optimizer minimizes the cell-wise cross-entropy loss with an initial learning rate of `1e-4` and batch size of 32. These hyperparameters were selected to match default U-Net settings [117]. Training this U-Net via distributed data parallelism and mixed precision on `PyTorch Lightning 1.6.5` [160] on four V100 GPUs requires approximately 1 hour wall-clock-time.

The processed dataset, with the input-output pair described by Equation (6.2), is split to consider 18 in-distribution and three OOD laser deposition locations. The three OOD locations are selected to each represent direct/indirect/failed ignition phenomena that can be used to evaluate the SDE-ML model's behavior when extrapolating beyond seen laser deposition locations. 80% of the in-distribution data is used for training, while the remaining 20% are split evenly for validating and unit testing the ML component of the present framework via cell accuracy (defined as the percentage of cells that are classified correctly). Without the SDE component, teacher forced (where the ML input consists only of the ground-truth samples Υ^n) next-frame predictions from this model provides a cell accuracy of 99.7% in the test set. Without the SDE component, we note that in the autoregressive (where the ML outputs $\hat{\Upsilon}^n$ are iteratively used as the next timestep's inputs, as described by Equation (6.2)) next-frame predictions, small errors from previous iterations accumulate (a well-known ML property [28]) – leading to approximately 90% test cell accuracy

after 30 ML timesteps (with an in-prediction duration of 240 μs).

6.4 Results

Here, we report results from ensemble kernel trajectory predictions made by the present SDE-ML modeling framework on the present configuration. In this work, each sample of initial kernel morphology condition Υ^0 is used to generate an ensemble with 100 stochastic variations, *i.e.*, approximately 100-fold larger than the experimental ensemble.

Figure 6.3 shows instantaneous predictions from the SDE-ML model (in red) that qualitatively captures kernel behavior seen in experimental measurements (in grayscale) for (a) direct, (b) indirect, and (c) failed ignition phenomena. At time after laser deposition $\tau = 4\,\mu$s, a small ignition kernel is observed at the three different laser deposition locations, with a pressure wave that is surrounding the kernel. In the direct ignition seen in Figure 6.3a, the kernel deposited in the central jet rapidly transitions into a sustained flame within $\tau = 28\,\mu$s and continues to propagate, which is also seen with the SDE-ML model. In Figure 6.3b, we see that the kernel deposited outside the jet does not ignite the fuel/oxidizer mixture until approximately $\tau = 140\,\mu$s. After this time, the hot plasma ejected from the asymmetric kernel interacts with the central jet, which causes a transition to a sustained flame with an oblong morphology. This indirect ignition phenomenon highlights the influence of kernel morphology in laser ignition, which is reasonably captured by the SDE-ML model. In Figure 6.3c, we observe that the SDE-ML can also accurately capture the absence of a sustained flame up until a long duration of $\tau = 500\,\mu$s.

We examine the statistical behavior of SDE-ML ensemble predictions by comparing the temporal evolution of mean kernel centroid positions against corresponding measurements in Figure 6.4. These positions are overlaid on top of mean velocity magnitudes and unit vectors. The SDE-ML model predicts trajectories that qualitatively agree with the measurements. Specifically, it can be seen that the kernels are transported (i) downstream by the jet, and (ii) radially due to entrainment. Near the central jet core ($r < 0.5d_{f,o}$), mean velocity trajectory predictions agree with the

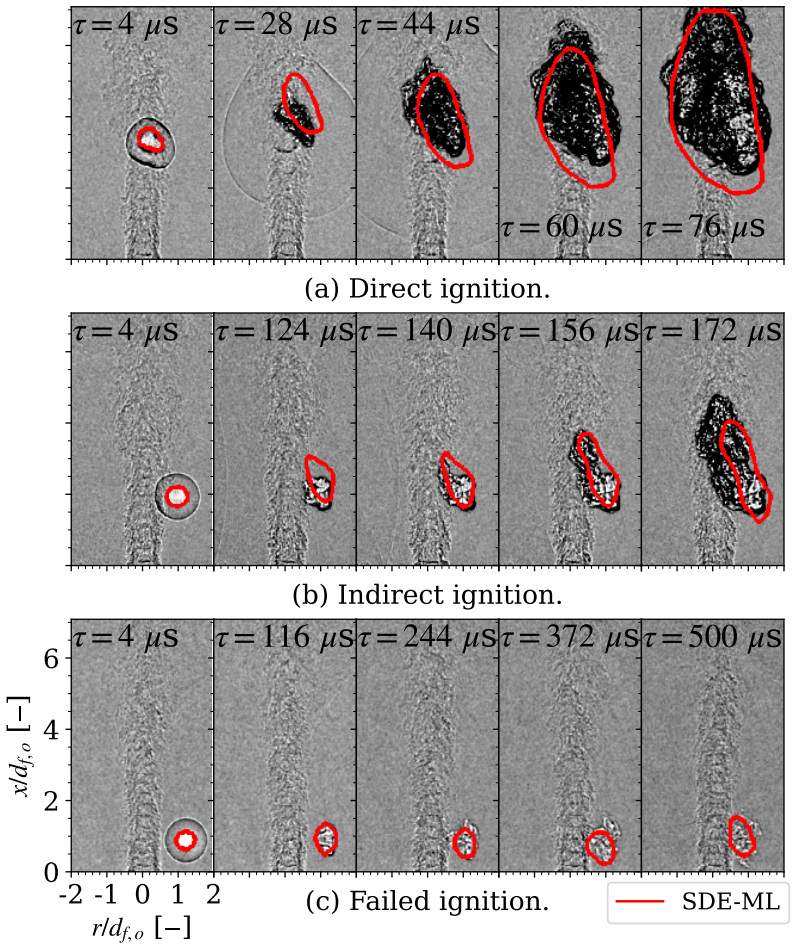


Figure 6.3: Comparisons of ignition kernel predictions from the SDE-ML model against experimental measurements of direct/indirect/failed ignition for time after later deposition τ.

experiment in time and position. However, we note that outside this range, quantitative discrepancies are observed in the SDE-ML predictions. On average, the ignition kernels predicted by the SDE-ML model in these locations are not transported as deeply (radial differences up to $0.4d_{f,o}$) into the jet as the averaged experimental measurements, indicating an under-prediction of kernel centroid radial velocity by the SDE-ML model. In addition, the kernels in these locations require up to 50% more time to reach the same axial location, when compared to experimental measurements. Sources of deviation in the kernel trajectory in this reduced-order model

result from the potential discrepancies between LES input and real-world velocity. LES improvements (including turbulence modeling, boundary condition treatment, and grid refinement) would result in flowfields that transport the modeled kernel in a more similar trajectory to the experiments. In addition, another source of discrepancy is the 2D treatment of this 3D configuration, which captures the trajectory of the kernel in a projected 2D plane. As such, any azimuthal velocity in the 3D configuration will cause an apparent motion of the kernel towards and away from the jet, which is not accounted for in the present modeling approach. Another potential limitation in the current 2D approach is the absence of laser deposition angles within the modeling framework. However, laser angle was not varied in the corresponding experimental configuration [86], so examining the full extent of 3D ignition is beyond the scope of the present work, and should be investigated in future work.

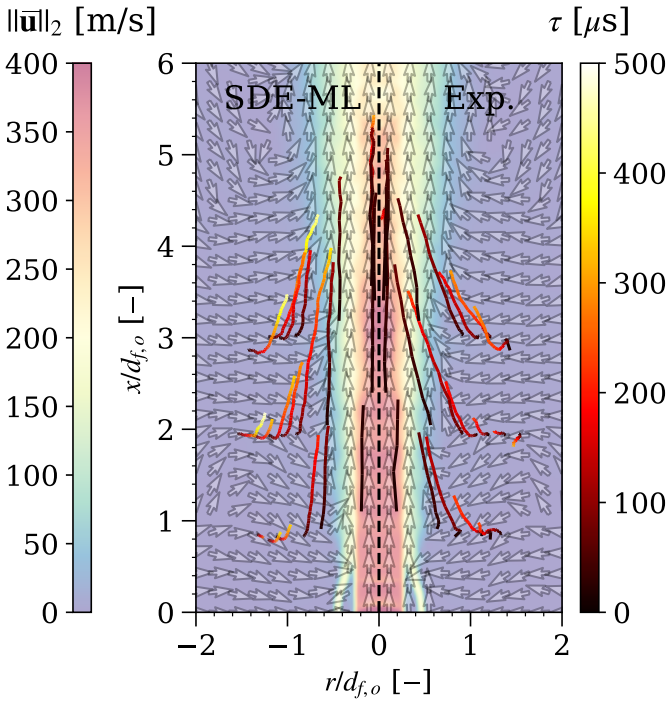


Figure 6.4: Mean kernel position trajectory from ensemble SDE-ML predictions against ensemble experimental measurements across time after laser deposition τ. LES mean velocity magnitude (with translucent unit vectors) is also shown.

Nevertheless, this velocity treatment is still sufficient for capturing the probability

of ignition time τ_{ig} from the ensemble experiments shown in Figure 6.5. In another reduced-order modeling study [94], successful ignition was determined by monitoring the ignition kernel growth rate. Here, ignition time is defined as the time taken for the kernel to exceed a growth rate of 0.2 mm^2/s in both SDE-ML predictions and experimental measurements. Estimated probabilities of the ignition time from ensemble predictions and measurements show good agreement in the distribution peak ($\sim$20%) at approximately $\tau_{ig} \approx 10\,\mu$s, which correspond to the large proportion of direct ignition cases within the present ensemble. After $\tau_{ig} \approx 10\,\mu$s, long-tailed ignition time distributions are seen for both predictions and measurements, which are a result of the delayed indirect ignition that is influenced by stochastic variations of kernel interactions with the turbulent fuel/oxidizer mixture. Ensemble SDE-ML predictions of τ_{ig} possess a standard deviation of 123 μs, which is within 18% difference with measured standard deviation of 88 μs.

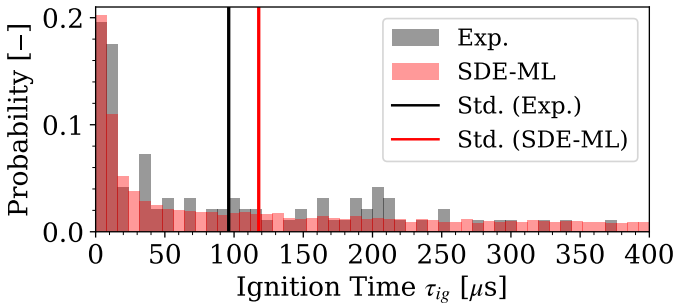
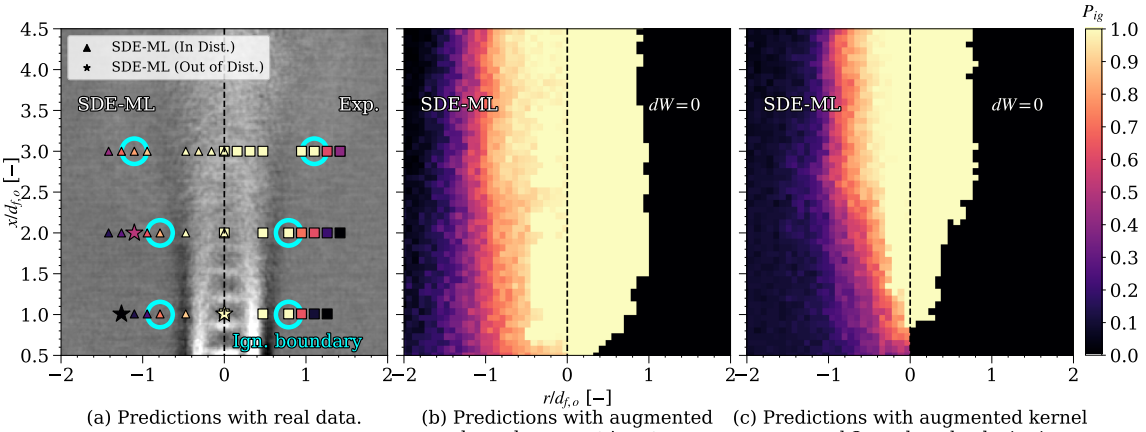


Figure 6.5: Comparison of normalized ignition time τ_{ig} distributions from SDE-ML predictions against measurements.

In this work, we map the probability of successful ignition with real data and two augmented datasets. Real data consists of the segmented Schlieren data (stacked with inert LES statistics), as described in Section 6.2. Since experimental ignition tests were only performed for 21 different laser deposition locations, an augmented dataset was generated with the purpose of estimating a spatially resolved ignition probability map. This augmented kernel dataset was created by spatially translating all ignited cells of a single frame of a kernel segment $\Upsilon^0(\tau = 4\,\mu\text{s}, x = 2d_{f,o}, r = 1.1d_{f,o})$ by $\Delta\mathbf{x}_{aug} = \mathbf{x}_{aug} - \mathbf{x}_p^{ML}(\Upsilon^0)$. This generates 1740 augmented initial kernel



(a) Predictions with real data. (b) Predictions with augmented kernel segment inputs. (c) Predictions with augmented kernel segments and 2× reduced velocity inputs.

Figure 6.6: SDE-ML predictions of ignition probability P_{ig} maps. Ensemble-averaged experimental Schlieren measurements (along with measured ignition boundaries in cyan) are shown in (a), while SDE-ML predictions without the stochastic component ($dW = 0$) are shown in (b) and (c).

morphologies Υ^0_{aug} that mimic kernels deposited with laser energy $E_{laser} = 26\,\text{mJ}$ at locations $3\,\text{mm} \leq x_{aug} < 30\,\text{mm}$ and $r_{aug} < 14\,\text{mm}$. These locations were chosen to avoid interactions between the kernels and the domain boundaries, since laser deposition does not typically occur near the combustor walls. A second augmented dataset is created to evaluate SDE-ML performance when tested with unseen flowfield conditions by reducing the LES velocity in the augmented kernel dataset by two-fold.

Figure 6.6a compares ignition probability maps predicted with real data against experimental measurements in the 21 laser deposition locations. This spatially sparse ignition probability map is shown on top of ensemble-averaged experimental Schlieren measurements. Cyan circles are used to highlight the ignition boundary measured from the experiments, which is defined here as the radial distance before ignition probability decreases below unity. At the ignition boundaries, the SDE-ML model predicts ignition probabilities of 0.72, 0.84, and 0.8 at axial locations $x = d_{f,o}$, $2d_{f,o}$, and $3d_{f,o}$, respectively. Of the three ignition probabilities, only $P_{ig}(x = d_{f,o})$ exceeds the measured uncertainty of ± 0.2. This under-prediction of ignition probability is likely caused by the under-prediction of mean radial velocity magnitude observed in Figure 6.4, which reduces the likelihood of the hot kernel interacting with the cold

reactants near the central jet.

The ignition probabilities at three OOD laser deposition locations that lead to direct ($x = d_{f,o}$, $r = 0$), indirect ($x = 2d_{f,o}$, $r = 1.1d_{f,o}$), and failed ($x = d_{f,o}$, $r = 1.3d_{f,o}$) ignition are also evaluated. At these locations, the SDE-ML model correctly predicts unity and zero probability for the direct and failed ignition cases, respectively. For the indirect ignition case, a predicted ignition probability of $P_{ig}^{pred} = 0.5$ is observed, which is within the uncertainty range of $P_{ig}^{exp} = 0.6 \pm 0.1$. Thus, the SDE-ML model can predict the kernel ignition probability reasonably well at slightly OOD deposition locations.

The spatially resolved ignition probability map predicted with the augmented kernel dataset is shown in Figure 6.6b. Here, we compare SDE-ML predictions with and without the stochastic component ($dW = 0$). Both approaches are seen to generate a spatially coherent ignition probability map, even though its training data does not extend beyond sparse laser deposition locations. However, we note that without the stochastic component, the ML model can only generate deterministic predictions, resulting in a binary probability map.

Figure 6.6c shows ignition probability maps predicted with the augmented kernel dataset with velocity inputs reduced by two-fold. Above $x > d_{f,o}$, a smaller region with unity ignition probability is observed with and without the stochastic component, which is a result of lower mean radial velocities. Below $x < d_{f,o}$, the model predicts only failed ignition when $dW = 0$ indicating that the predicted ignition kernel loses hot plasma significantly before being transported to a reactive fuel/oxidizer mixture. Whether this is physically plausible is uncertain within this work, given the absence of measurements under these velocity conditions. We note that all ML-based models cannot predict reliably with vastly OOD inputs [6]. Uncertainty quantification [223, 224] could partially address this issue by providing further interpretability on model confidence in the presence of unfamiliar inputs, and could potentially be employed for guiding additional experimental measurements. However, these methods are active scientific pursuits that are currently beyond the scope of this chapter.

6.5 Summary

This chapter introduced a reduced-order physics-embedded SDE-ML modeling framework that employs simulation data and sparse ensemble measurements for statistically characterizing ignition behavior within an experimental gaseous CH_4/O_2 model rocket combustor configuration.

This SDE-ML framework was shown to qualitatively capture the evolution of ignition kernels within rocket combustors. In particular, the SDE-ML model was able to capture the influence of asymmetric kernel morphology on stochastic indirect ignition. In addition, ensemble ignition time predictions showed reasonable statistical agreement with ensemble experimental measurements. Ensemble predictions of mean kernel trajectory and ignition probability demonstrated qualitative agreement with the ensemble measurements. Quantitative differences arose from limitations in the treatment of kernel velocity (potential discrepancies in the LES input and 2D treatment of this model). This resulted in errors in ignition probability at ignition boundary and kernel position of up to 0.28 and $0.4d_{f,o}$, respectively.

To demonstrate that this SDE-ML framework can develop reasonable predictions when tested on unseen situations, we showed that the model reasonably captures ignition probabilities associated with direct, indirect, and failed ignition modes at slightly OOD laser deposition locations. The versatility of the SDE-ML model was further evaluated with augmented datasets that mimic inputs for generating a spatially resolved ignition probability map under different velocity conditions. The SDE-ML model can generate a spatially coherent ignition probability map, with only a training dataset consisting of ignition kernels deposited in sparse laser deposition locations. However, we note that providing the model with vastly OOD inputs potentially results in spurious phenomena.

These results highlight the potential and limitations in combining physics-based modeling approaches with data-driven methods for leveraging sparse experimental measurements and cost-effective non-reacting simulation data in capturing different ignition modes within a complex multi-physics flow configuration.

Chapter 7

Conclusions and Future Work*

7.1 Key Findings

This dissertation contributes methods and knowledge towards overcoming data-related limitations within ML techniques that are employed towards scientific and engineering applications, in the context of propulsion and multi-physics flows.

This limitation is directly addressed through the development of an accessible framework for curating and distributing large datasets for multi-physics flows related to propulsion. This resulted in BLASTNet 2.0 – a 2.2 TB public dataset containing 744 full-domain samples from 34 high-fidelity DNS configurations of turbulent flows. We process this dataset for benchmarking the behavior of various deep learning approaches in turbulent SR problem. Through our scaling analysis, we provide empirical measurements of the relationship between predictive performance with ML model size and cost when trained with a large dataset. From the same analysis, we also demonstrate that NN architecture design can matter significantly, especially for smaller ML models, and the benefits of employing physics-based losses can persist at moderate model sizes. These findings provide useful insights in the design of deep learning models for multi-physics flow applications, while BLASTNet 2.0 provides a

*This chapter contains select descriptions and discussions from the artificial intelligence (AI) review paper by Ihme and Chung [225], with significant modifications made for this dissertation. M. Ihme and W.T. Chung contributed equally to reviewing AI progress and applications.

rich data resource for training and evaluating models for scientific and engineering turbulent flow problems.

In problems where training data is not sufficient, such as in complex real-fluid configurations, alternatives to deep learning can be considered. We examine the opportunities offered by symbolic regression and random forest models through *a priori* analysis of closure terms in inert and reacting DNS of turbulent transcritical configurations. We find that these methods can provide benefits related to model interpretability. Specifically, the feature importance score from the random forest, in conjunction with interpretable weights in sparse symbolic regression, can be used to inform the discovery of analytic expressions of SGS models.

One issue that can arise from insufficiently trained ML models is the presence of OOD errors in the resulting predictions. We introduce a modeling approach that ameliorates this issue by bounding potential OOD ML errors through dynamic combustion submodel assignment. Through this approach, predictive errors are bounded by the worst-performing domain knowledge-based model, which enables stable deployment of ML models within numerical simulations. Here, random forests are trained to assign three different candidate combustion models within the shared LES domain of a 2D model rocket configuration. *A posteriori* simulations employing this modeling approach demonstrated that this ML-integrated simulation approach can assist in managing fidelity-cost trade-offs in high-fidelity simulations of turbulent reacting flows.

We present another approach for integrating domain knowledge with ML for overcoming limitations in obtaining costly simulation and experimental data of stochastic multi-physics flow phenomena. Specifically, we introduce a physics-embedded SDE-ML reduced-order modeling framework that relies on sparse ensemble measurements and non-reacting simulation data for statistically characterizing ignition behavior within a rocket combustor configuration. Here, we employ an SDE for representing the turbulent transport of ignition kernels, while a deep learning model is used to predict the time evolution of the ignition kernel morphology. This SDE-ML is able to qualitatively capture different laser ignition modes, as well as the influence of asymmetric kernel morphology on ignition behavior. Ignition timing, kernel position, and

ignition probability predicted by the SDE-ML model are also shown to agree reasonably with quantitative experimental measurements. These results highlight potential benefits in cost- and data-effective predictive modeling of complex multi-physics flows, which can be obtained when combining ML with domain knowledge.

7.2 Recommendations for Future Research

This dissertation has demonstrated numerous approaches towards employing ML techniques in applications involving multi-physics flows and propulsion systems. However, several open challenges (involving limited datasets and OOD errors) and opportunities (involving large datasets) still remain, which could be addressed by:

Diversifying Data and Benchmarks for Multi-Physics Flow Problems

We presented BLASTNet 2.0, in Chapter 3, which provides access to terabytes of 3D turbulent reacting flow data. While this work presents a sustainable framework for curating large flow physics datasets, BLASTNet 2.0 only contains reacting flow data for a select number of H_2 and CH_4, and is not sufficient for representing a wide range of multi-physics flow conditions. Thus, future work should focus on increasing the diversity of flow configurations with the BLASTNet framework. In addition, BLASTNet 2.0 consists largely of decorrelated snapshots of statistically stationary flow configurations. In order to extend BLASTNet towards ML tasks that involve temporal dependency and dynamic behavior, future iterations of BLASTNet should consider correlated temporal snapshots.

The sensitivity of certain ML models to variations in datasets (as formalized in Equation (2.20)) also motivates the introduction of open benchmarks in order to provide transparent and consistent insights into developed ML approaches. In Chapter 3, we released an open benchmark with BLASTNet 2.0 data that provided insight into effective deep learning design choices specifically for turbulent closure modeling problem via ML-based SR. Given the growing number of ML applications related to scientific discovery and modeling within multi-physics flows (as highlighted in Chapter 1), the development of open benchmarks that cater to new learning tasks within

multi-physics flows could lead to improved ML models for any new applications.

Towards ML Foundation Models for Multi-physics Flows

The availability of large datasets and public benchmarks for multi-physics flows could lead to the development of *foundation models* [226] for solving a wide range of predictive modeling problems with promising accuracy and cost-effectiveness, as has been seen recently in the atmospheric sciences [227].

An ML *foundation model* is a general purpose ML model (typically with >1B trainable parameters) that has been (i) pre-trained offline via *self-supervised learning* [228], *i.e.*, supervised learning with minimal labeling of large and diverse datasets (typically terabyte-scale), which can then be (ii) *fine-tuned*, *i.e.*, further offline training with supervised transfer learning [229] involving smaller and more specific datasets for tailored downstream applications. When compared to pre-training approaches, fine-tuning techniques are reasonably matured, and can be completed with a few hours on a single GPU processor [230]. Thus, promising directions that can proliferate ML foundation models within a broader range of multi-physics flow domains would involve the improvement of pre-training techniques via (i) discovering effective self-supervised learning tasks and (ii) improving the cost-effectiveness of ML architectures – both for fully utilizing large flow physics datasets (such as BLASTNet in Chapter 3).

Firstly, self-supervised learning enables ML models to harness large datasets in a cost-effective manner by focusing training on information inherent within data structures (such as smoothness or continuity within fluid fields), instead of information provided by costly labeled datasets. Previous studies [231, 232] have shown that the predictive performance of fine-tuned models can benefit from the discovery of new domain-specific self-supervised learning tasks for upstream pre-training. Within multi-physics flows, self-supervised learning have begun to be explored, as seen with super-resolution in this dissertation. Outside this work, experimental measurement reconstruction [233, 234] and next-timestep prediction [28, 235] are promising self-supervised learning tasks. However, we note that these previous efforts have largely focused on demonstrating the utility of these self-supervised learning tasks without any form of transfer learning, and there remains open opportunities in investigating

the influence of these tasks on downstream fine-tuning performance.

While self-supervised learning techniques reduce manual efforts during data processing, pre-training large deep learning models can still incur large computational costs. Computational complexity of many popular deep learning operations (such as in fully-connected, 2D convolution, and self-attention layers) can scale approximately quadratically with input dimensions. Thus, the discovery of new mathematical operations that can efficiently represent non-linear patterns is essential for leveraging large datasets with foundation models. Promising developments in this direction include state-space layers [236] (which uses generalized 1D convolution operations to achieve linear complexity), and specialized matrix structures [237] (for achieving sub-quadratic complexity). In relation to this, the benchmark tools presented in Chapter 3 could be employed towards developing of future NN operations suited for multi-physics flow data.

Further Integration of ML with Existing Flow Physics Approaches

This dissertation has explored the use of physics-based regularization (Chapter 3), integration of ML within multi-physics flow solvers (Chapter 5), and hybrid physics-ML reduced order modeling (Chapter 6). These efforts highlight opportunities for further combining domain knowledge from multi-physics flows with ML. Future work could involve the exploration of novel physics- and chemistry-based loss function terms [158]. In addition, a wide range of analytical reduced-order modeling could be combined with ML for predicting complex multi-physics flow phenomena.

Further integration of the ML model with multi-physics flow solvers via two-way coupling could offer opportunities for online learning techniques that aid model performance in OOD conditions. In Chapter 5, ML training was performed offline, prior to simulation calculations. A pre-trained model was then coupled one-way with the multi-physics flow solver for *a posteriori* calculations, *i.e.*, the ML model could affect the outputs from the multi-physics flow solver, but remained unaffected by the multi-physics flow solver. Similar to work performed in more matured ML domains [238], future efforts in two-way coupling could involve the use of unseen flow conditions from a simulation for fine-tuning model parameters in-flight via online

transfer learning [229] or reinforcement learning [239]. These efforts have begun to show promise in multi-physics flow applications, including turbulence modeling [240] and flow control [241]. Specific potential extensions of these efforts towards propulsion applications would include introducing improvements to algorithm robustness [242] for deployment in safety-critical systems, as well as accuracy and cost [243], especially with challenging multiscale physical/chemical phenomena.

Further Investigations in Ameliorating and Understanding OOD Errors

A significant portion of this dissertation has been dedicated towards understanding effects from OOD inputs. For example, in Chapter 6, we demonstrated that the application of a deep learning regressor in OOD flow conditions could result in potentially spurious ignition predictions. In order to deploy ML models in safety critical systems encountered in propulsion domains, further work should be performed to understand and quantify the limitations of ML approaches in OOD conditions. This could involve the extension of ML-tailored uncertainty quantification techniques [244] that can provide further insight on model confidence in the presence of unfamiliar inputs. Improved understanding of deep learning models could also lead to better strategies in reducing these OOD errors in these black-box approaches. A promising direction in interpreting deep learning models involves the examination of their underlying mathematical operations, *i.e. mechanistic interpretation* [245], which has so far shown some degree of success in small models and simple architectures.

Exploration of New Approaches for Flow Physics Discovery

ML can offer opportunities for the systematic discovery of trends, models, and phenomena within multi-physics flows. In Chapter 4, we explored the employment of interpretable ML methods for the discovery of closure models. One key result from this chapter is the re-discovery of an SGS stress model (which was previously derived through Taylor series expansion) through insights gained from interpretable ML. The robustness of this proposed approach can be explored within a wider range of flow physics configurations and closure problems. In addition, the potential of

ML approaches for discovering trends within complex flows involving multiphase, detonating, and hypersonic phenomena can be explored. This can involve the application and discovery of methods beyond the supervised learning techniques shown in this dissertation. More matured scientific domains have demonstrated the use of reinforcement learning [239] for discovering new computational and mathematical algorithms [246, 247], and genetic algorithms [248] for discovering analytical expressions for physical phenomena [249], which can be extended to benefit computational modeling of multi-physics flows in propulsion.

Appendix A

BLASTNet Supplementary Documentation

A.1 Maintenance Plan and Long Term Preservation

The contributors to BLASTNet 2.0 are committed to maintaining and preserving this dataset. Maintenance of this dataset will largely involve tracking and fixing issues that might be discovered after release. To facilitate this, we host an issues webpage (`https://github.com/blastnet/blastnet.github.io/issues`) for user feedback. All data is shared via Kaggle, ensuring that the data will be preserved and available in the long-term. In addition, our maintenance plan involves adhering to the FAIR principles [156] for scientific data management, with the specific details as follows:

Findable All data are indexed and can be easily searched via both Kaggle and BLASTNet platforms. To ensure that the data is findable, a `http://schema.org` structured metadata is employed, as detailed in Appendices A.2 and A.3. All BLASTNet datasets share a global and persistent DOI at Zenodo: `https://doi.org/10.5281/zenodo.7242864`.

Accessible Both data and descriptive metadata are retrievable via the Kaggle command-line API. This protocol is free and available at `https://github.com/Kaggle/kaggle-api`, with authentication and authorization provided through a Kaggle account. We provide a `bash` script for users to download all data (shared in multiple repositories) at once with this API. Users can also download the data directly from Kaggle repositories.

Interoperable The data and descriptive metadata use accessible formats that can be read by standard `python numpy` and `json` packages. BLASTNet's `http://schema.org` structured metadata also references the structured metadata of each separate BLASTNet repository (providing information on specific contributors and Kaggle URLs). We have attempted to use accessible language when generating these metadata.

Reusable The descriptive metadata contains information on the flow configuration (initial conditions, chemistry, numerics, and source publication). In addition, all Kaggle repositories employ a CC BY-SA NC 4.0 license. The structured `http://schema.org` metadata provides rich information that passes the rich results test (`https://search.google.com/test/rich-results`). All data and descriptive metadata are presented in consistent little-endian single-precision binaries and `.json` files, guaranteeing acceptable standards for fast I/O, sufficient floating-point precision, and broad accessibility via widely-used `python` packages.

A.2 Additional BLASTNet 2.0 Details

BLASTNet 2.0 contains pre-processed DNS data shared via a network of Kaggle repositories, with links consolidated at the landing page `https://blastnet.github.io`.

A.2.1 Data Format and Directory Structure

Data, generated from different multi-physics flow solvers initially exists in a range of formats (`.vtk`, `.vtu`, `.tec`, and `.dat`) that are not readily formatted for training ML models. Thus, we pre-process all generated data into a consistent and convenient format consisting of physical and chemical data, descriptive metadata, and web metadata, along with instructions for reading the data. This information is summarized in Figure A.1.

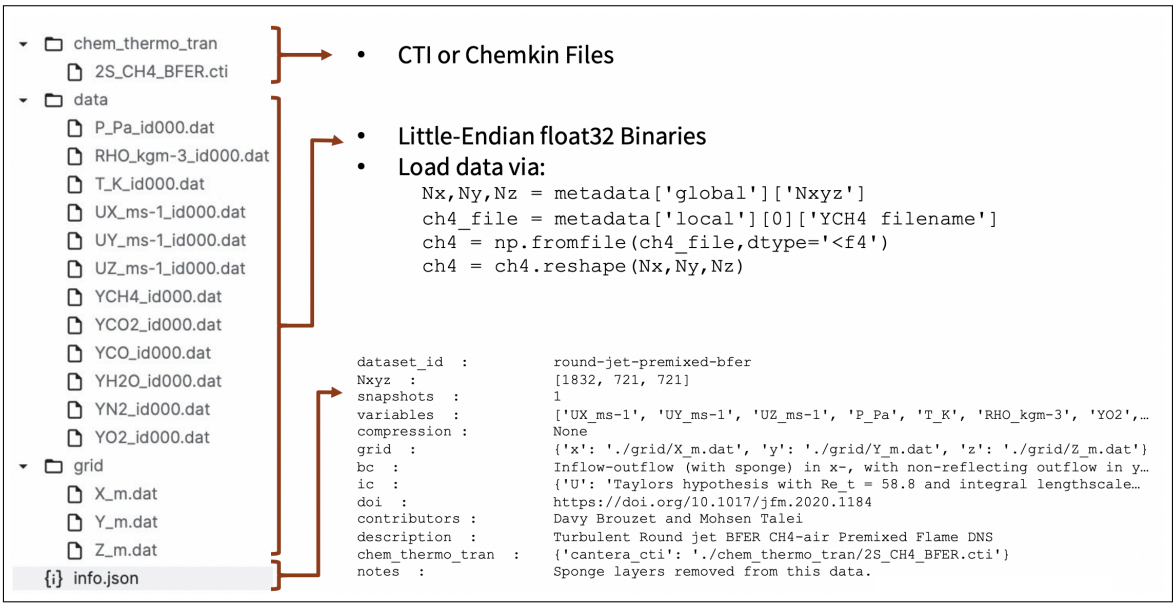


Figure A.1: Directory structure and reading instructions for an instance of a BLASTNet configuration.

Files on Flow Physics and Chemistry

All flowfield data are processed into a consistent format – little-endian single-precision binaries that can be read with `np.fromfile`/`np.memmap`, as shown in Figure A.1. The choice of this data format enables high I/O speed in loading arrays. As also shown in Figure A.1, we also provide `.json` files (see Appendix A.2.1) that store additional information on configurations, contributors, solvers, and corresponding source publications. Chemical mechanisms and transport properties are shared through

Cantera [182] `.cti/.xml/.yaml` or Chemkin [250] `fortran` files. Thus, BLASTNet data contains all information needed to reconstruct any derived auxiliary quantities (such as vorticity, viscosity, turbulence closure terms, along with heat and chemical transport coefficients) from the conservation equations.

Descriptive Metadata

The binary data described in Appendix A.2.1 contains information on physical and chemical data, without much context. Details involving global information such as configuration, boundary/initial conditions, solvers, related publications, and spatial grid information, as well as local temporal information (if any) are provided through an `info.json` file in each Kaggle repository. Listings 1 and 2 present the `python` code used to generate global and local information in one example of `info.json`.

Structured Web Metadata

A `http://schema.org` metadata has been added to `https://blastnet.github.io/datasets`, and tested with `https://search.google.com/test/rich-results`.

Reading Data

As shown in Figure A.1, BLASTNet data can be read by (i) loading the descriptive metadata with the `json` package on `python`, and (i) using `np.fromfile/np.memmap` to load and reshape the data.

A.3 Additional Momentum128 3D SR Dataset Details

A.3.1 Data Format and Directory Structure

The Momentum128 3D SR Dataset contains velocity and density sub-volumes (see Appendix A.3.1) extracted and processed from BLASTNet 2.0, along with descriptive

Listing 1 Python command for generating global metadata for a BLASTNet Kaggle repository.

```
metadata['global'] = {
        "dataset_id": "waitongchung/inert-ch4o2-hit-dns",
             "Nxyz": [129,129,129],
         "snapshots": 98,
         "variables": ["UX_ms-1","UY_ms-1","UZ_ms-1",
                       "P_Pa","T_K","RHO_kgm-3",
                       "YO2","YCH4"],
       "compression": "None",
             "grid": {"x": "./grid/X_m.dat",
                      "y": "./grid/Y_m.dat",
                      "z": "./grid/Z_m.dat"},
          "numerics": {"spatial": "4th order central-differencing
                                   with 2nd order ENO",
                      "temporal": "3rd-order SSP-RK3 (non-stiff)
                                   and semi-implicit ROWPLUS (stiff)",
                        "solver": "CharlesX"},
                "bc": "Periodic in x-, y-, and z-directions.",
                "ic": {"U": "HIT Von Karman Pao with Re_t = 80 and
                             integral length-scale of 62.5E-6m",
                   "T [K]": 300,
                  "P [Pa]": 101325,
                 "Mixture": "CH4-O2 inert branch from 1D
                             cantera counterflow calculations."},
               "doi": "https://doi.org/10.1016/j.combustflame.2021.111758",
      "contributors": "Wai Tong Chung and Matthias Ihme",
       "description": "Compressible Inert CH4-O2 Homogeneous
                       Isotropic Turbulence DNS",
  "chem_thermo_tran": {"description": "FRC and Mixture-Averaged Transport
                                       with constant lewis number",
                       "cantera_xml": "./chem_thermo_tran/bfer.xml"}
}
```

metadata (Appendix A.3.1), web metadata (Appendix A.3.1), and instructions for reading the data in Appendix A.3.1.

Files on Flow Physics

All 2000 sub-volumes (labels with $128 \times 128 \times 128$ number of voxels) of density and three velocity components $[\rho, u_i]$ are also presented in little-endian single-precision binary format, similarly to BLASTNet 2.0 (see Appendix A.2.1), which can be read with `np.fromfile` or `np.memmap`. This is shown in Figure A.2, which also shows the five data splits described in Section 3.2.2. In addition, Favre-filtered features for 8, 16, and 32× SR are also provided, along with pre-trained weights from all models reported in this study. The sub-volume files are named with `<Variable Name and`

Listing 2 Python command for generating local metadata for a BLASTNet Kaggle repository.

```
metadata['local'] = [
                {"id": 0,
            "time [s]": 6.88389e-06,
    "UX_ms-1 filename": "./data/UX_ms-1_id000.dat",
    "UY_ms-1 filename": "./data/UY_ms-1_id000.dat",
    "UZ_ms-1 filename": "./data/UZ_ms-1_id000.dat",
    "   P_Pa filename": "./data/P_Pa_id000.dat",
       "T_K filename": "./data/T_K_id000.dat",
  "RHO_kgm-3 filename": "./data/RHO_kgm-3_id000.dat",
       "YO2 filename": "./data/YO2_id000.dat",
      "YCH4 filename": "./data/YCH4_id000.dat"},
                {"id": 1, ...},
                    ...,
                {"id": 97, ...}
]
```

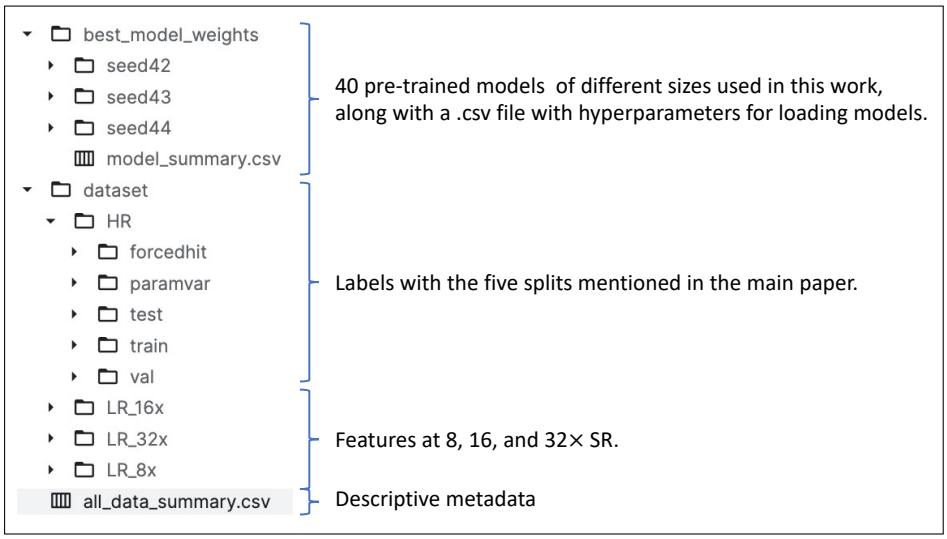


Figure A.2: Directory structure of the Momentum128 3D SR dataset.

`SI Unit>_id<hash value>.dat`, where the hash value provides a unique ID based on the spatial coordinates of the sub-volume location and the index of configuration.

Descriptive Metadata

In addition to the sub-volumes, we provide `.csv` files that provide information on hash ID, Kaggle ID, short configuration description, k-means cluster index, and spatial grid

size for the different dataset splits used in this work.

Structured Web Metadata

A `http://schema.org` metadata has been added to `https://blastnet.github.io/datasets`, and tested with `https://search.google.com/test/rich-results`.

Reading Data

Similar to BLASTNet 2.0, this data can be read by using `np.fromfile/np.memmap` to load and reshape the data.

Bibliography

[1] T. Poinsot, Prediction and control of combustion instabilities in real engines, Proc. Combust. Inst. 36 (2017) 1–28.

[2] M. Ihme, Requirements towards predictive simulations of turbulent combustion, AIAA Paper 2019-0996 (2019).

[3] H. J. Curran, Developing detailed chemical kinetic mechanisms for fuel combustion, Proc. Combust. Inst. 37 (2019) 57–81.

[4] P. Domingo, L. Vervisch, Recent developments in DNS of turbulent combustion, Proc. Combust. Inst. 39 (2023) 2055–2076.

[5] K. Duraisamy, G. Iaccarino, H. Xiao, Turbulence modeling in the age of data, Annu. Rev. Fluid Mech. 51 (2019) 357–377.

[6] M. Ihme, W. T. Chung, A. A. Mishra, Combustion machine learning: Principles, progress and prospects, Prog. Energy Combust. Sci. 91 (2022) 101010.

[7] W. T. Chung, A. A. Mishra, M. Ihme, Interpretable data-driven methods for subgrid-scale closure in LES for transcritical LOX/GCH_4 combustion, Combust. Flame 239 (2022) 111758.

[8] Y. Zhang, W. Dong, L. A. Vandewalle, R. Xu, G. P. Smith, H. Wang, Neural network approach to response surface development for reaction model optimization and uncertainty minimization, Combust. Flame 251 (2023) 112679.

[9] W. T. Chung, A. A. Mishra, N. Perakis, M. Ihme, Data-assisted combustion simulations with dynamic submodel assignment using random forests, Combust. Flame 227 (2021) 172–185.

[10] W. T. Chung, C. Laurent, D. Passiatore, M. Ihme, Ensemble predictions of laser ignition with a hybrid stochastic physics-embedded deep-learning framework, Proc. Combust. Inst. 40 (2025). Accepted.

[11] S. Nakaya, K. Omi, T. Okamoto, Y. Ikeda, C. Zhao, M. Tsue, H. Taguchi, Instability and mode transition analysis of a hydrogen-rich combustion in a model afterburner, Proc. Combust. Inst. 38 (2021) 5933–5942.

[12] N. Kuzhagaliyeva, A. Thabet, E. Singh, B. Ghanem, S. M. Sarathy, Using deep neural networks to diagnose engine pre-ignition, Proc. Combust. Inst. 38 (2021) 5915–5922.

[13] M. T. Henry de Frahan, N. T. Wimer, S. Yellapantula, R. W. Grout, Deep reinforcement learning for dynamic control of fuel injection timing in multi-pulse compression ignition engines, Int. J. Engine Res. 23 (2022) 1503–1521.

[14] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. `https://www.tensorflow.org`.

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., PyTorch: An imperative style, high-performance deep learning library, Adv. Neural Inform. Process. Syst. 32 (2019) 8024–8035.

[16] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de las Casas, L. A. Hendricks, J. Welbl, A. Clark, et al., An empirical analysis of compute-optimal large language model training, Adv. Neural Inform. Process. Syst. 35 (2022) 30016–30030.

[17] C. Sun, A. Shrivastava, S. Singh, A. Gupta, Revisiting unreasonable effectiveness of data in deep learning era, Proc. IEEE Int. Conf. Comput. Vis. (2017) 843–852.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, Proc. IEEE Conf. Comput. Vision Pattern Recognit. (2009) 248–255.

[19] G. Wenzek, M.-A. Lachaux, A. Conneau, V. Chaudhary, F. Guzmán, A. Joulin, É. Grave, CCNet: Extracting high quality monolingual datasets from web crawl data, Proc. Lang. Resour. Eval. 12 (2020) 4003–4012.

[20] K. Kohse-Höinghaus, Clean combustion: Chemistry and diagnostics for a systems approach in transportation and energy conversion, Prog. Energy Combust. Sci. 65 (2018) 1–5.

[21] W. T. Chung, B. Akoush, P. Sharma, A. Tamkin, K. S. Jung, J. H. Chen, J. Guo, D. Brouzet, M. Talei, B. Savard, A. Y. Poludnenko, M. Ihme, Turbulence in focus: Benchmarking scaling behavior of 3D volumetric super-resolution with BLASTNet 2.0 data, Adv. Neural Inf. Process. Syst. 36 (2023) 77430–77484.

[22] C. M. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag, Berlin, Germany, 2006.

[23] A. M. Legendre, Nouvelles Méthodes pour la Détermination des Orbites des Comètes, Courcier, Paris, France, 1805.

[24] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and Regression Trees, Chapman & Hall, New York, NY, 1984.

[25] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, Nature 323 (1986) 533–536.

[26] R. Maulik, O. San, J. D. Jacob, C. Crick, Sub-grid scale model classification and blending through deep learning, J. Fluid Mech. 870 (2019) 784–812.

[27] B. Li, Z. Yang, X. Zhang, G. He, B.-Q. Deng, L. Shen, Using machine learning to detect the turbulent region in flow past a circular cylinder, J. Fluid Mech. 905 (2020) A10.

[28] P. Sharma, W. T. Chung, B. Akoush, M. Ihme, A review of physics-informed machine learning in fluid mechanics, Energies 16 (2023) 2343.

[29] M. Raissi, Z. Wang, M. S. Triantafyllou, G. E. Karniadakis, Deep learning of vortex-induced vibrations, J. Fluid Mech. 861 (2019) 119–137.

[30] I. J. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, Cambridge, MA, 2016.

[31] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012).

[32] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, P. Battaglia, Learning mesh-based simulation with graph networks, Proc. Int. Conf. Learn. Represent. 9 (2021).

[33] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, Proc. Int. Conf. Learn. Repr. 9 (2021).

[34] S. B. Pope, Turbulent Flows, Cambridge University Press, Cambridge, United Kingdom, 2000.

[35] K. S. Jung, S. O. Kim, T. Lu, J. H. Chen, C. S. Yoo, On the flame stabilization of turbulent lifted hydrogen jet flames in heated coflows near the autoignition limit: A comparative DNS study, Combust. Flame 233 (2021) 111584.

[36] B. Jiang, D. Brouzet, M. Talei, R. L. Gordon, Q. Cazeres, B. Cuenot, Turbulent flame-wall interactions for flames diluted by hot combustion products, Combust. Flame 230 (2021) 111432.

[37] D. Brouzet, M. Talei, M. J. Brear, B. Cuenot, The impact of chemical modelling on turbulent premixed flame acoustics, J. Fluid Mech. 915 (2021) A3.

[38] Q. Wang, M. Ihme, Y.-F. Chen, J. Anderson, A TensorFlow simulation framework for scientific computing of fluid flows on tensor processing units, Comput. Phys. Commun. 274 (2022) 108292.

[39] L. Esclapez, P. C. Ma, E. Mayhew, R. Xu, S. Stouffer, T. Lee, H. Wang, M. Ihme, Fuel effects on lean blow-out in a realistic gas turbine combustor, Combust. Flame 181 (2017) 82–99.

[40] C. M. Huang, 2D benchmark reacting flow dataset for reduced order modeling exploration [Data set], 2020. University of Michigan-Deep Blue Data.

[41] R. McConkey, E. Yee, F.-S. Lien, A curated dataset for data-driven turbulence modelling, Sci. Data 8 (2021) 255.

[42] F. Bonnet, J. A. Mazari, P. Cinnella, P. Gallinari, AirfRANS: High fidelity computational fluid dynamics dataset for approximating Reynolds-Averaged Navier–Stokes solutions, Adv. Neural Inf. Process. Syst. 35 (2022) 23463–23478.

[43] Y. Li, E. Perlman, M. Wan, Y. Yang, C. Meneveau, R. Burns, S. Chen, A. Szalay, G. Eyink, A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence, J. Turbul. 9 (2008) N31.

[44] M. Bode, M. Gauding, Z. Lian, D. Denker, M. Davidovic, K. Kleinheinz, J. Jitsev, H. Pitsch, Using physics-informed enhanced super-resolution generative adversarial networks for subfilter modeling in turbulent reactive flows, Proc. Combust. Inst. 38 (2021) 2617–2625.

[45] J. Smagorinsky, General circulation experiments with the primitive equations, Mon. Weather Rev. 91 (1963) 99–164.

[46] R. A. Clark, J. H. Ferziger, W. C. Reynolds, Evaluation of subgrid-scale models using an accurately simulated turbulent flow, J. Fluid Mech. 91 (1979) 1–16.

[47] J. Bardina, J. Ferziger, W. C. Reynolds, Improved subgrid-scale models for large-eddy simulation, AIAA Paper 1980-1357 (1980).

[48] P. Moin, K. Squires, W. Cabot, S. Lee, A dynamic subgrid-scale model for compressible turbulence and scalar transport, Phys. Fluids A 3 (1991) 2746–2757.

[49] M. Germano, U. Piomelli, P. Moin, W. H. Cabot, A dynamic subgrid-scale eddy viscosity model, Phys. Fluids A 3 (1991) 1760–1765.

[50] A. W. Vreman, An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications, Phys. Fluids 16 (2004) 3670–3681.

[51] B. Vreman, B. Geurts, H. Kuerten, On the formulation of the dynamic mixed subgrid-scale model, Phys. Fluids 6 (1994) 4057–4059.

[52] O. Colin, F. Ducros, D. Veynante, T. Poinsot, A thickened flame model for large eddy simulations of turbulent premixed combustion, Phys. Fluids 12 (2000) 1843–1863.

[53] M. Ihme, C. M. Cha, H. Pitsch, Prediction of local extinction and re-ignition effects in non-premixed turbulent combustion using a flamelet/progress variable approach, Proc. Combust. Inst. 30 (2005) 793–800.

[54] C. D. Pierce, P. Moin, Progress-variable approach for large-eddy simulation of non-premixed turbulent combustion, J. Fluid Mech. 504 (2004) 73–97.

[55] L. C. Selle, N. A. Okong'o, J. Bellan, K. G. Harstad, Modelling of subgrid-scale phenomena in supercritical transitional mixing layers: An a priori study, J. Fluid Mech. 593 (2007) 57–91.

[56] U. Unnikrishnan, J. C. Oefelein, V. Yang, A priori analysis of subfilter scalar covariance fields in turbulent reacting LOX-CH_4 mixing layers, AIAA Paper 2019-1495 (2019).

[57] C. J. Lapeyre, A. Misdariis, N. Cazard, D. Veynante, T. Poinsot, Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates, Combust. Flame 203 (2019) 255–264.

[58] A. Seltz, P. Domingo, L. Vervisch, Z. M. Nikolaou, Direct mapping from LES resolved scales to filtered-flame generated manifolds using convolutional neural networks, Combust. Flame 210 (2019) 71–82.

[59] M. Ihme, C. Schmitt, H. Pitsch, Optimal artificial neural networks and tabulation methods for chemistry representation in LES of a bluff-body swirl-stabilized flame, Proc. Combust. Inst. 32 (2009) 1527–1535.

[60] M. T. Henry de Frahan, S. Yellapantula, R. King, M. S. Day, R. W. Grout, Deep learning for presumed probability density function models, Combust. Flame 208 (2019) 436–450.

[61] R. Ranade, T. Echekki, A framework for data-based turbulent combustion closure: A posteriori validation, Combust. Flame 210 (2019) 279–291.

[62] S. P. Burke, T. E. W. Schumann, Diffusion flames, Ind. Eng. Chem. 20 (1928) 998–1004.

[63] O. Gicquel, N. Darabiha, D. Thévenin, Laminar premixed hydrogen/air counterflow flame simulations using flame prolongation of ILDM with differential diffusion, Proc. Combust. Inst. 28 (2000) 1901–1908.

[64] J. A. van Oijen, L. P. H. de Goey, Modelling of premixed laminar flames using flamelet-generated manifolds, Combust. Sci. Tech. 161 (2000) 113–137.

[65] Y. Liang, S. B. Pope, P. Pepiot, A pre-partitioned adaptive chemistry methodology for the efficient implementation of combustion chemistry in particle PDF methods, Combust. Flame 162 (2015) 3236–3253.

[66] W. Xie, Z. Lu, Z. Ren, L. Hou, Dynamic adaptive chemistry via species timescale and Jacobian-aided rate analysis, Proc. Combust. Inst. 36 (2017) 645–653.

[67] S. Yang, R. Ranjan, V. Yang, S. Menon, W. Sun, Parallel on-the-fly adaptive kinetics in direct numerical simulation of turbulent premixed flame, Proc. Combust. Inst. 36 (2017) 2025–2032.

[68] H. Wu, Y. C. See, Q. Wang, M. Ihme, A Pareto-efficient combustion framework with submodel assignment for predicting complex flame configurations, Combust. Flame 162 (2015) 4208–4230.

[69] H. Wu, P. C. Ma, T. Jaravel, M. Ihme, Pareto-efficient combustion modeling for improved CO-emission prediction in LES of a piloted turbulent dimethyl ether jet flame, Proc. Combust. Inst. 37 (2019) 2267–2276.

[70] Q. Douasbin, M. Ihme, C. Arndt, Pareto-efficient combustion framework for predicting transient ignition dynamics in turbulent flames: Application to a pulsed jet-in-hot-coflow flame, Combust. Flame 223 (2021) 153–165.

[71] F. C. Christo, A. R. Masri, E. M. Nebot, S. B. Pope, An integrated PDF/neural network approach for simulating turbulent reacting systems, Proc. Combust. Inst. 26 (1996) 43–48.

[72] J. A. Blasco, N. Fueyo, C. Dopazo, J. Ballester, Modelling the temporal evolution of a reduced combustion chemical system with an artificial neural network, Combust. Flame 113 (1998) 38–52.

[73] J. A. Blasco, N. Fueyo, J. C. Larroya, C. Dopazo, J. Y. Chen, Single-step time-integrator of a methane-air chemical system using artificial neural networks, Comput. Chem. Eng. 23 (1999) 1127–1133.

[74] B. A. Sen, S. Menon, Linear eddy mixing based tabulation and artificial neural networks for large eddy simulations of turbulent flames, Combust. Flame 157 (2010) 62–74.

[75] S. Alqahtani, T. Echekki, A data-based hybrid model for complex fuel chemistry acceleration at high temperatures, Combust. Flame 223 (2021) 142–152.

[76] A. Kempf, F. Flemming, J. Janicka, Investigation of lengthscales, scalar dissipation, and flame orientation in a piloted diffusion flame by LES, Proc. Combust. Inst. 30 (2005) 557–565.

[77] O. Owoyele, P. Kundu, M. M. Ameen, T. Echekki, S. Som, Application of deep artificial neural networks to multi-dimensional flamelet libraries and spray flames, Int. J. Engine Res. 21 (2020) 151–168.

[78] A. K. Chatzopoulos, S. Rigopoulos, A chemistry tabulation approach via rate-controlled constrained equilibrium (RCCE) and artificial neural networks (ANNs), with application to turbulent non-premixed $CH_4/H_2/N_2$ flames, Proc. Combust. Inst. 34 (2013) 1465–1473.

[79] L. L. C. Franke, A. K. Chatzopoulos, S. Rigopoulos, Tabulation of combustion chemistry via artificial neural networks (ANNs): Methodology and application to LES-PDF simulation of Sydney flame L, Combust. Flame 185 (2017) 245–260.

[80] E. Mastorakos, Forced ignition of turbulent spray flames, Proc. Combust. Inst. 36 (2017) 2367–2383.

[81] S. F. Ahmed, R. Balachandran, E. Mastorakos, Measurements of ignition probability in turbulent non-premixed counterflow flames, Proc. Combust. Inst. 31 (2007) 1507–1513.

[82] E. Bach, J. Kariuki, J. R. Dawson, E. Mastorakos, H.-J. Bauer, Spark ignition of single bluff-body premixed flames and annular combustors, AIAA Paper 2013-1182 (2013).

[83] K. Prieur, D. Durox, J. Beaunier, T. Schuller, S. Candel, Ignition dynamics in an annular combustor for liquid spray and premixed gaseous injection, Proc. Combust. Inst. 36 (2017) 3717–3724.

[84] B. Sforzo, H. Dao, S. Wei, J. Seitzman, Liquid fuel composition effects on forced, nonpremixed ignition, J. Eng. Gas Turbine Power 139 (2017) 031509.

[85] M. Cordier, A. Vandel, G. Cabot, B. Renou, A. M. Boukhalfa, Laser-induced spark ignition of premixed confined swirled flames, Combust. Sci. Tech. 185 (2013) 379–407.

[86] R. Strelau, M. Frederick, W. C. Senior, R. Gejji, C. D. Slabaugh, Modes of laser spark ignition of a model rocket combustor, AIAA Paper 2023-2377 (2023).

[87] G. Lacaze, B. Cuenot, T. Poinsot, M. Oschwald, Large eddy simulation of laser ignition and compressible reacting flow in a rocket-like configuration, Combust. Flame 156 (2009) 1166–1180.

[88] O. Gurliat, V. Schmidt, O. Haidn, M. Oschwald, Ignition of cryogenic H_2/LOX sprays, Aerosp. Sci. Technol. 7 (2003) 517–531.

[89] J. M. Wang, D. A. Buchta, J. B. Freund, Hydrodynamic ejection caused by laser-induced optical breakdown, J. Fluid Mech. 888 (2020) A16.

[90] T. Jaravel, J. Labahn, B. Sforzo, J. Seitzman, M. Ihme, Numerical study of the ignition behavior of a post-discharge kernel in a turbulent stratified crossflow, Proc. Combust. Inst. 37 (2019) 5065–5072.

[91] K. Maeda, T. Teixeira, J. M. Wang, J. Hokanson, C. Melone, M. Di Renzo, S. Jones, J. Urzay, G. Iaccarino, An integrated heterogeneous computing framework for ensemble simulations of laser-induced ignition, AIAA Paper 2023-3597 (2023).

[92] L. Esclapez, E. Riber, B. Cuenot, Ignition probability of a partially premixed burner using LES, Proc. Combust. Inst. 35 (2015) 3133–3141.

[93] Y. Tang, M. Hassanaly, V. Raman, B. A. Sforzo, J. Seitzman, Probabilistic modeling of forced ignition of alternative jet fuels, Proc. Combust. Inst. 38 (2021) 2589–2596.

[94] A. Neophytou, E. S. Richardson, E. Mastorakos, Spark ignition of turbulent recirculating non-premixed gas and spray flames: A model for predicting ignition probability, Combust. Flame 159 (2012) 1503–1522.

[95] L. Esclapez, F. Collin-Bastiani, E. Riber, B. Cuenot, A statistical model to predict ignition probability, Combust. Flame 225 (2021) 180–195.

[96] B. Sforzo, J. Seitzman, Modeling ignition probability for stratified flows, J. Propuls. Power 33 (2017) 1294–1304.

[97] P. P. Popov, D. A. Buchta, M. J. Anderson, L. Massa, J. Capecelatro, D. J. Bodony, J. B. Freund, Machine learning-assisted early ignition prediction in a complex flow, Combust. Flame 206 (2019) 451–466.

[98] T. Poinsot, D. Veynante, Theoretical and Numerical Combustion, Thierry Poinsot, Toulouse, France, fourth edition, 2022.

[99] R. J. Kee, M. E. Coltrin, P. Glarborg, Chemically Reacting Flow: Theory and Practice, John Wiley & Sons, Hoboken, NJ, 2005.

[100] D.-Y. Peng, D. B. Robinson, A new two-constant equation of state, Ind. Eng. Chem. Fundam. 15 (1976) 59–64.

[101] B. E. Poling, J. M. Prausnitz, J. P. O'Connell, The Properties of Gases and Liquids, McGraw-Hill, New York, NY, 2001.

[102] P. C. Ma, Y. Lv, M. Ihme, An entropy-stable hybrid scheme for simulations of transcritical real-fluid flows, J. Comp. Phys. 340 (2017) 330–357.

[103] N. Peters, Turbulent Combustion, Cambridge Monographs on Mechanics, Cambridge University Press, Cambridge, United Kingdom, 2000.

[104] T. Echekki, E. Mastorakos, Turbulent Combustion Modeling: Advances, New Trends and Perspectives, Springer Science & Business Media, Berlin, Germany, 2010.

[105] D. Veynante, L. Vervisch, Turbulent combustion modeling, Prog. Energy Combust. Sci. 28 (2002) 193–266.

[106] H. Pitsch, Large-eddy simulation of turbulent combustion, Annu. Rev. Fluid Mech. 38 (2006) 453–482.

[107] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, H. Liu, Feature selection: A data perspective, ACM Comput. Surv. 50 (2017) 1–45.

[108] R. Tibshirani, Regression shrinkage and selection via the Lasso, J. R. Stat. Soc. Series B Stat. Methodol. 58 (1996) 267–288.

[109] K. Champion, B. Lusch, J. N. Kutz, S. L. Brunton, Data-driven discovery of coordinates and governing equations, Proc. Natl. Acad. Sci. U.S.A. 116 (2019) 22445–22451.

[110] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, Proc. Int. Conf. Learn. Repr. 3 (2015).

[111] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, Proc. Mach. Learn. Res. 9 (2010) 249–256.

[112] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification, Proc. IEEE Int. Conf. Comput. Vis. (2015) 1026–1034.

[113] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (1998) 2278–2324.

[114] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, Proc. IEEE Conf. Comput. Vision Pattern Recogn. (2016) 770–778.

[115] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, Proc. Mach. Learn. Res. 37 (2015) 448–456.

[116] J. L. Ba, J. R. Kiros, G. E. Hinton, Layer normalization, arXiv preprint 1607.06450 (2016).

[117] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv. 18 (2015) 234–241.

[118] R. Messenger, L. Mandell, A modal search technique for predictive nominal scale multivariate analysis, J. Am. Stat. Assoc. 67 (1972) 768–772.

[119] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32.

[120] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, B. Yu, Definitions, methods, and applications in interpretable machine learning, Proc. Natl. Acad. Sci. U.S.A. 116 (2019) 22071–22080.

[121] G. Louppe, L. Wehenkel, A. Sutera, P. Geurts, Understanding variable importances in forests of randomized trees, Adv. Neural Inform. Process. Syst. 26 (2013).

[122] C. Shorten, T. M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (2019) 1–48.

[123] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comp. Phys. 378 (2019) 686–707.

[124] M. Bitter, S. Scharnowski, R. Hain, C. J. Kähler, High-repetition-rate PIV investigations on a generic rocket model in sub-and supersonic flows, Exp. Fluids 50 (2011) 1019–1030.

[125] J.-P. Hickey, P. C. Ma, M. Ihme, S. S. Thakur, Large eddy simulation of shear coaxial rocket injector: Real fluid effects, AIAA Paper 2013-4071 (2013).

[126] W. T. Chung, P. C. Ma, M. Ihme, Examination of diesel spray combustion in supercritical ambient fluid using large-eddy simulations, Int. J. Engine Res. 21 (2020) 122–133.

[127] J. Guo, D. Brouzet, W. T. Chung, M. Ihme, Analysis of ducted fuel injection at high-pressure transcritical conditions using large-eddy simulations, Int. J. Engine Res. 25 (2024) 305–319.

[128] J. E. Temme, P. M. Allison, J. F. Driscoll, Combustion instability of a lean premixed prevaporized gas turbine combustor studied using phase-averaged PIV, Combust. Flame 161 (2014) 958–970.

[129] E. Mueller, W. Mell, A. Simeoni, Large eddy simulation of forest canopy flow for wildland fire modeling, Can. J. For. Res. 44 (2014) 1534–1544.

[130] F. Morandini, X. Silvani, J.-L. Dupuy, A. Susset, Fire spread across a sloping fuel bed: Flame dynamics and heat transfers, Combust. Flame 190 (2018) 158–170.

[131] M. Raissi, A. Yazdani, G. E. Karniadakis, Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations, Science 367 (2020) 1026–1030.

[132] J. Yu, L. Lu, X. Meng, G. E. Karniadakis, Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems, Comput. Methods Appl. Mech. Eng. 393 (2022) 114823.

[133] R. S. M. Freitas, A. Péquin, R. M. Galassi, A. Attili, A. Parente, Model identification in reactor-based combustion closures using sparse symbolic regression, Combust. Flame 255 (2023) 112925.

[134] W. T. Chung, K. S. Jung, J. H. Chen, M. Ihme, BLASTNet: A call for community-involved big data in combustion machine learning, Appl. Energy Combust. Sci. 12 (2022) 100087.

[135] E. Agustsson, R. Timofte, NTIRE 2017 challenge on single image super-resolution: Dataset and study, Proc. IEEE Conf. Comput. Vis. Pattern Recognit. W. (2017) 1122–1131.

[136] E. Ershov, A. Savchik, D. Shepelev, N. Banić, M. S. Brown, R. Timofte, K. Koščević, M. Freeman, V. Tesalin, D. Bocharov, et al., NTIRE 2022 challenge on night photography rendering, Proc. IEEE Conf. Comput. Vis. Pattern Recognit. W. (2022) 1287–1300.

[137] B. Arad, R. Timofte, R. Yahel, N. Morag, A. Bernat, Y. Cai, J. Lin, Z. Lin, H. Wang, Y. Zhang, et al., NTIRE 2022 spectral recovery challenge and data set, Proc. IEEE Conf. Comput. Vis. Pattern Recognit. W. (2022) 863–881.

[138] J. Cornebise, I. Orsolic, F. Kalaitzis, Open high-resolution satellite imagery: The WorldStrat dataset – With application to super-resolution, Adv. Neural Inform. Process. Syst. 35 (2022) 25979–25991.

[139] S. Nah, S. Baik, S. Hong, G. Moon, S. Son, R. Timofte, K. Mu Lee, NTIRE 2019 challenge on video deblurring and super-resolution: Dataset and study, Proc. IEEE Conf. Comput. Vis. Pattern Recognit. W. (2019) 1996–2005.

[140] H. Liu, J. Liu, J. Li, J.-S. Pan, X. Yu, DL-MRI: A unified framework of deep learning-based MRI super resolution, J. Healthc. Eng. 2021 (2021) 5594649.

[141] Y. Xue, Q. Yang, G. Hu, K. Guo, L. Tian, Deep-learning-augmented computational miniature mesoscope, Optica 9 (2022) 1009–1021.

[142] Z. Deng, C. He, Y. Liu, K. C. Kim, Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework, Phys. Fluids 31 (2019) 125111.

[143] K. Fukami, K. Fukagata, K. Taira, Super-resolution reconstruction of turbulent flows with machine learning, J. Fluid Mech. 870 (2019) 106–120.

[144] L. Yu, M. Z. Yousif, M. Zhang, S. Hoyas, R. Vinuesa, H.-C. Lim, Three-dimensional ESRGAN for super-resolution reconstruction of turbulent flows with tricubic interpolation-based transfer learning, Phys. Fluids 34 (2022) 125126.

[145] Z. Wang, X. Li, L. Liu, X. Wu, P. Hao, X. Zhang, F. He, Deep-learning-based super-resolution reconstruction of high-speed imaging in fluids, Phys. Fluids 34 (2022) 037107.

[146] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, C. Change Loy, ESRGAN: Enhanced super-resolution generative adversarial networks, Proc. Eur. Conf. Comput. Vis. W. (2018) 63–79.

[147] B. Lim, S. Son, H. Kim, S. Nah, K. Mu Lee, Enhanced deep residual networks for single image super-resolution, Proc. IEEE Conf. Comput. Vision Pattern Recogn. W. (2017) 136–144.

[148] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, Y. Fu, Image super-resolution using very deep residual channel attention networks, Proc. Eur. Conf. Comput. Vis. 15 (2018) 286–301.

[149] G. Wen, Z. Li, K. Azizzadenesheli, A. Anandkumar, S. M. Benson, U-FNO — An enhanced Fourier neural operator-based deep-learning model for multiphase flow, Adv. Water Resour. 163 (2022) 104180.

[150] J. H. Chen, A. Choudhary, B. de Supinski, M. DeVries, E. R. Hawkes, S. Klasky, W. K. Liao, K. L. Ma, J. Mellor-Crummey, N. Podhorszki, R. Sankaran, S. Shende, C. S. Yoo, Terascale direct numerical simulations of turbulent combustion using S3D, Comput. Sci. Discov. 2 (2009) 015001.

[151] O. Desjardins, G. Blanquart, G. Balarac, H. Pitsch, High order conservative finite difference scheme for variable density low Mach number turbulent flows, J. Comp. Phys. 227 (2008) 7125–7159.

[152] A. Y. Poludnenko, E. S. Oran, The interaction of high-speed turbulence with flames: Global properties and internal flame structure, Combust. Flame 157 (2010) 995–1011.

[153] P. Moin, K. Mahesh, Direct numerical simulation: A tool in turbulence research, Annu. Rev. Fluid Mech. 30 (1998) 539–578.

[154] J. Guo, X. I. A. Yang, M. Ihme, Structure of the thermal boundary layer in turbulent channel flows at transcritical conditions, J. Fluid Mech. 934 (2022) A45.

[155] B. Savard, E. R. Hawkes, K. Aditya, H. Wang, J. H. Chen, Regimes of premixed turbulent spontaneous ignition and deflagration under gas-turbine reheat combustion conditions, Combust. Flame 208 (2019) 402–419.

[156] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, et al., The FAIR Guiding Principles for scientific data management and stewardship, Sci. Data 3 (2016) 160018.

[157] X. I. A. Yang, K. P. Griffin, Grid-point and time-step requirements for direct numerical simulation and large-eddy simulation, Phys. Fluids 33 (2021) 015108.

[158] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nat. Rev. Phys. 3 (2021) 422–440.

[159] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, S. Hoyer, Machine learning-accelerated computational fluid dynamics, Proc. Natl. Acad. Sci. U.S.A. 118 (2021) e2101784118.

[160] W. Falcon, The PyTorch Lightning team, PyTorch Lightning, 2022. `https://www.pytorchlightning.ai`.

[161] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: From error visibility to structural similarity, IEEE Trans. Image Process. 13 (2004) 600–612.

[162] A. Glaws, R. King, M. Sprague, Deep learning for in situ data compression of large turbulent flow simulations, Phys. Rev. Fluids 5 (2020) 114602.

[163] T. Bao, S. Chen, T. T. Johnson, P. Givi, S. Sammak, X. Jia, Physics guided neural networks for spatio-temporal super-resolution of turbulent flows, Proc. Mach. Learn. Res. 180 (2022) 118–128.

[164] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, J. Pineau, Towards the systematic reporting of the energy and carbon footprints of machine learning, J. Mach. Learn. Res. 21 (2020) 1–43.

[165] Y. Nirkin, L. Wolf, T. Hassner, HyperSeg: Patch-wise hypernetwork for real-time semantic segmentation, Proc. IEEE Conf. Comput. Vision Pattern Recogn. (2021) 4061–4070.

[166] C. White, R. Tu, J. Kossaifi, G. Pekhimenko, K. Azizzadenesheli, A. Anandkumar, Speeding up Fourier neural operators via mixed precision, arXiv pre-print 2307.15034 (2023).

[167] J. Bellan, Future challenges in the modelling and simulations of high-pressure flows, Combust. Sci. Tech. 192 (2020) 1199–1218.

[168] J. Zips, C. Traxinger, M. Pfitzner, Thermodynamic analysis and large-eddy simulations of LOX-CH_4 and LOX-H_2 flames at high pressure, AIAA Paper 2018-4765 (2018).

[169] E. W. Lemmon, M. O. McLinden, D. G. Friend, Thermophysical Properties of Fluid Systems in NIST Chemistry WebBook, NIST Standard Reference Database Number 69, National Institute of Standards and Technology, Gaithersburg, MD, 2018.

[170] B. Franzelli, E. Riber, L. Y. M. Gicquel, T. Poinsot, Large eddy simulation of combustion instabilities in a lean partially premixed swirled flame, Combust. Flame 159 (2012) 621–637.

[171] S. T. Chong, Y. Tang, M. Hassanaly, V. Raman, Turbulent mixing and combustion of supercritical jets, AIAA Paper 2017-0141 (2017).

[172] J. Bellan, Direct numerical simulation of a high-pressure turbulent reacting temporal mixing layer, Combust. Flame 176 (2017) 245–262.

[173] W. K. Bushe, C. Devaud, J. Bellan, A priori evaluation of the double-conditioned conditional source-term estimation model for high-pressure heptane turbulent combustion using DNS data obtained with one-step chemistry, Combust. Flame 217 (2020) 131–151.

[174] S. Takahashi, Preparation of a generalized chart for the diffusion coefficients of gases at high pressures, J. Chem. Eng. Jpn. 7 (1975) 417–420.

[175] T. H. Chung, M. Ajlan, L. L. Lee, K. E. Starling, Generalized multiparameter correlation for nonpolar and polar fluid transport properties, Ind. Eng. Chem. Res. 27 (1988) 671–679.

[176] A. M. Ruiz, G. Lacaze, J. C. Oefelein, R. Mari, B. Cuenot, L. Selle, T. Poinsot, Numerical benchmark for high-Reynolds-number supercritical flows with large density gradients, AIAA J. 54 (2016) 1445–1460.

[177] J. C. Oefelein, Advances in modeling supercritical fluid behavior and combustion in high-pressure propulsion systems, AIAA Paper 2019-0634 (2019).

[178] Y. Khalighi, J. W. Nichols, F. Ham, S. K. Lele, P. Moin, Unstructured large eddy simulation for prediction of noise issued from turbulent jets in various configurations, AIAA Paper 2011-2886 (2011).

[179] H. Wu, P. C. Ma, M. Ihme, Efficient time-stepping techniques for simulating turbulent reactive flows with stiff chemistry, Comput. Phys. Commun. 243 (2019) 81–96.

[180] H. Huo, V. Yang, Subgrid-scale models for large-eddy simulation of supercritical combustion, AIAA Paper 2013-706 (2013).

[181] U. Unnikrishnan, X. Wang, S. Yang, V. Yang, Subgrid scale modeling of the equation of state for turbulent flows under supercritical conditions, AIAA Paper 2017-4855 (2017).

[182] D. G. Goodwin, H. K. Moffat, I. Schoegl, R. L. Speth, B. W. Weber, Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes, 2022. `https://www.cantera.org`.

[183] H. Huo, V. Yang, Supercritical LOX/methane combustion of a shear coaxial injector, AIAA Paper 2011-326 (2011).

[184] U. Unnikrishnan, J. C. Oefelein, V. Yang, Direct numerical simulation of a turbulent reacting liquid-oxygen/methane mixing layer at supercritical pressure, AIAA Paper 2018-4564 (2018).

[185] T. Saad, D. Cline, R. Stoll, J. C. Sutherland, Scalable tools for generating synthetic isotropic turbulence with arbitrary spectra, AIAA J. 55 (2017) 327–331.

[186] C. Bailly, D. Juves, A stochastic approach to compute subsonic-noise using linearized Euler's equations, AIAA Paper 1999-1872 (1999).

[187] F. Williams, Descriptions of nonpremixed turbulent combustion, AIAA Paper 2006-1505 (2006).

[188] M. Schoepplein, J. Weatheritt, R. Sandberg, M. Talei, M. Klein, Application of an evolutionary algorithm to LES modelling of turbulent transport in premixed flames, J. Comput. Phys. 374 (2018) 1166–1179.

[189] J. Ling, R. Jones, J. Templeton, Machine learning strategies for systems with invariance properties, J. Comp. Phys. 318 (2016) 22–35.

[190] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[191] E. S. Taskinoglu, J. Bellan, A posteriori study using a DNS database describing fluid disintegration and binary-species mixing under supercritical pressure: heptane and nitrogen, J. Fluid Mech. 645 (2010) 211–254.

[192] T. K. Kumar, Multicollinearity in regression analysis, Rev. Econ. Stat 57 (1975) 365–366.

[193] A. Altmann, L. Toloşi, O. Sander, T. Lengauer, Permutation importance: A corrected feature importance measure, Bioinform. 26 (2010) 1340–1347.

[194] L. Toloşi, T. Lengauer, Classification with correlated features: Unreliability of feature ranking and solutions, Bioinform. 27 (2011) 1986–1994.

[195] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Adv. Neural Inf. Process. Syst. 30 (2017) 4765–4774.

[196] S. Silvestri, M. P. Celano, C. Kirchberger, G. Schlieben, O. Haidn, O. Knab, Investigation on recess variation of a shear coax injector for a single element GOX-GCH_4 combustion chamber, Trans. JSASS Aerospace Tech. Jpn. 14 (2016) 101–108.

[197] M. Ihme, L. Shunn, J. Zhang, Regularization of reaction progress variable for application to flamelet-based combustion models, J. Comp. Phys. 231 (2012) 7715–7721.

[198] A. Felden, E. Riber, B. Cuenot, Impact of direct integration of analytically reduced chemistry in LES of a sooting swirled non-premixed combustor, Combust. Flame 191 (2018) 270–286.

[199] A. W. Vreman, B. A. Albrecht, J. A. van Oijen, L. P. H. de Goey, R. J. M. Bastiaans, Premixed and nonpremixed generated manifolds in large-eddy simulation of Sandia flame D and F, Combust. Flame 153 (2008) 394–416.

[200] G. P. Smith, D. M. Golden, M. Frenklach, N. W. Moriarty, B. Eiteneer, M. Goldenberg, C. T. Bowman, R. K. Hanson, S. Song, W. C. Gardiner Jr., V. V. Lissianski, Z. Qin, GRI-Mech 3.0, 2000. `http://www.me.berkeley.edu/gri-mech/`.

[201] H. Pitsch, FlameMaster v3.1: A C++ computer program for 0D combustion and 1D laminar flame calculations, 1998.

[202] J. Zips, H. Müller, M. Pfitzner, Non-adiabatic tabulation methods to predict wall-heat loads in rocket combustion, AIAA Paper 2017-1469 (2017).

[203] P. E. Lapenna, R. Amaduzzi, D. Durigon, G. Indelicato, F. Nasuti, F. Creta, Simulation of a single-element GCH_4/GOx rocket combustor using a non-adiabatic flamelet method, AIAA Paper 2018-4872 (2018).

[204] N. Perakis, O. J. Haidn, Inverse heat transfer method applied to capacitively cooled rocket thrust chambers, Int. J. Heat Mass Transf. (2019) 150–166.

[205] S. Kawai, J. Larsson, Dynamic non-equilibrium wall-modeling for large eddy simulation at high Reynolds numbers, Phys. Fluids 25 (2013) 015105.

[206] N. Perakis, O. J. Haidn, M. Ihme, Investigation of CO recombination in the boundary layer of CH_4/O_2 rocket engines, Proc. Combust. Inst. 38 (2020). In press.

[207] P. C. Ma, H. Wu, M. Ihme, J.-P. Hickey, Nonadiabatic flamelet formulation for predicting wall heat transfer in rocket engines, AIAA J. 56 (2018) 2336–2349.

[208] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, P. C. Sabeti, Detecting novel associations in large data sets, Science 334 (2011) 1518–1524.

[209] R. W. Bilger, Turbulent jet diffusion flames, Prog. Energy Combust. Sci. 1 (1976) 87–109.

[210] R. Ge, M. Zhou, Y. Luo, Q. Meng, G. Mai, D. Ma, G. Wang, F. Zhou, McTwo: A two-step feature selection algorithm based on maximal information coefficient, BMC Bioinform. 17 (2016) 142.

[211] G. Bradski, The OpenCV library, Dr. Dobb's J. 25 (2000) 120–123.

[212] J.-L. Wu, H. Xiao, E. Paterson, Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework, Phys. Rev. Fluids 3 (2018) 74602.

[213] J. Ling, J. Templeton, Evaluation of machine learning algorithms for prediction of regions of high Reynolds-averaged Navier-Stokes uncertainty, Phys. Fluids 27 (2015) 085103.

[214] K. Schindler, An overview and comparison of smooth labeling methods for land-cover classification, IEEE Trans. Geosci. Remote Sens. 50 (2012) 4534–4545.

[215] H. Müller, J. Zips, M. Pfitzner, D. Maestro, B. Cuenot, L. Selle, R. Ranjan, P. Tudisco, S. Menon, Numerical investigation of flow and combustion in a single-element GCH_4/GOX rocket combustor: A comparative LES study, AIAA Paper 2016-4997 (2016).

[216] C. Roth, O. Haidn, A. Chemnitz, T. Sattelmayer, Y. Daimon, G. Frank, H. Müller, J. Zips, M. Pfitzner, R. Keller, P. Gerlinger, D. Maestro, B. Cuenot, H. Riedmann, L. Selle, Numerical investigation of flow and combustion in a single-element GCH_4/GOX rocket combustor, AIAA Paper 2016-4995 (2016).

[217] M. Di Renzo, L. Fu, J. Urzay, HTR solver: An open-source exascale-oriented task-based multi-GPU high-order code for hypersonic aerothermodynamics, Comput. Phys. Commun. 255 (2020) 107262.

[218] L. Fu, X. Y. Hu, N. A. Adams, A family of high-order targeted ENO schemes for compressible-fluid simulations, J. Comp. Phys. 305 (2016) 333–359.

[219] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Rev. Soc. Ind. Appl. Math. 43 (2001) 89–112.

[220] A. Ern, V. Giovangigli, Multicomponent Transport Algorithms, Springer Berlin, Heidelberg, Germany, 1994.

[221] S. De, A. Doostan, Neural network training using l_1-regularization and bi-fidelity data, J. Comp. Phys. 458 (2022) 111010.

[222] J. Yang, M. Gao, Z. Li, S. Gao, F. Wang, F. Zheng, Track anything: Segment anything meets videos, arXiv preprint 2304.11968 (2023).

[223] H. Wang, D. A. Sheen, Combustion kinetic model uncertainty quantification, propagation and minimization, Prog. Energy Combust. Sci. 47 (2015) 1–31.

[224] H. R. Fairbanks, L. Jofre, G. Geraci, G. Iaccarino, A. Doostan, Bi-fidelity approximation for uncertainty quantification and sensitivity analysis of irradiated particle-laden turbulence, J. Comp. Phys. 402 (2020) 108996.

[225] M. Ihme, W. T. Chung, Artificial intelligence as a catalyst for combustion science and engineering, Proc. Combust. Inst. 40 (2025). Accepted.

[226] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al., On the opportunities and risks of foundation models, arXiv preprint 2108.07258 (2021).

[227] C. Bodnar, W. P. Bruinsma, A. Lucic, M. Stanley, J. Brandstetter, P. Garvan, M. Riechert, J. Weyn, H. Dong, A. Vaughan, et al., Aurora: A foundation model of the atmosphere, arXiv preprint 2405.13063 (2024).

[228] R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, et al., A cookbook of self-supervised learning, arXiv preprint 2304.12210 (2023).

[229] S. Thrun, Learning to Learn, Springer, New York, NY, 1998.

[230] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, QLoRA: Efficient finetuning of quantized LLMs, Adv. Neural Inf. Process. Syst. 36 (2023) 10088–10115.

[231] B. Wallace, B. Hariharan, Extending and analyzing self-supervised learning across domains, Proc. Eur. Conf. Comput. Vis. 16 (2020) 717–734.

[232] A. Tamkin, G. Banerjee, M. Owda, V. Liu, S. Rammoorthy, N. Goodman, DABS 2.0: Improved datasets and algorithms for universal self-supervision, Adv. Neural Inf. Process. Syst. 35 (2022) 38358–38372.

[233] M. Dai, B. Zhou, J. Zhang, R. Cheng, Q. Liu, R. Zhao, B. Wang, B. Gao, 3-D soot temperature and volume fraction reconstruction of afterburner flame via deep learning algorithms, Combust. Flame 252 (2023) 112743.

[234] T. Yoon, S. W. Kim, H. Byun, Y. Kim, C. D. Carter, H. Do, Deep learning-based denoising for fast time-resolved flame emission spectroscopy in high-pressure combustion environment, Combust. Flame 248 (2023) 112583.

[235] K. Stachenfeld, D. B. Fielding, D. Kochkov, et al., Learned simulators for turbulence, Proc. Int. Conf. Learn. Represent. 10 (2022).

[236] J. T. H. Smith, A. Warrington, S. Linderman, Simplified state space layers for sequence modeling, Proc. Int. Conf. Learn. Represent. 11 (2023).

[237] D. Fu, S. Arora, J. Grogan, I. Johnson, E. S. Eyuboglu, A. Thomas, B. Spector, M. Poli, A. Rudra, C. Ré, Monarch Mixer: A simple sub-quadratic GEMM-based architecture, Adv. Neural Inf. Process. Syst. 36 (2023) 77546–77603.

[238] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Adv. Neural Inf. Process. Syst. 33 (2020) 1877–1901.

[239] R. S. Sutton, A. G. Barto, Reinforcement Learning: An Introduction, MIT press, Cambridge, MA, 2018.

[240] G. Novati, H. L. de Laroussilhe, P. Koumoutsakos, Automating turbulence modelling by multi-agent reinforcement learning, Nat. Mach. Intell. 3 (2021) 87–96.

[241] S. Verma, G. Novati, P. Koumoutsakos, Efficient collective swimming by harnessing vortices through deep reinforcement learning, Proc. Natl. Acad. Sci. U.S.A. 115 (2018) 5849–5854.

[242] K. Alhazmi, S. M. Sarathy, Adaptive phase shift control of thermoacoustic combustion instabilities using model-free reinforcement learning, Combust. Flame 257 (2023) 113040.

[243] G. Novati, P. Koumoutsakos, Remember and forget for experience replay, Proc. Mach. Learn. Res. 97 (2019) 4851–4860.

[244] E. Saetta, R. Tognaccini, G. Iaccarino, Uncertainty quantification in autoencoders predictions: Applications in aerodynamics, J. Comp. Phys. 506 (2024) 112951.

[245] C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, S. Carter, Zoom in: An introduction to circuits, Distill 5 (2020) e00024–001.

[246] A. Fawzi, M. Balog, A. Huang, T. Hubert, B. Romera-Paredes, M. Barekatain, A. Novikov, F. J. R Ruiz, J. Schrittwieser, G. Swirszcz, et al., Discovering faster matrix multiplication algorithms with reinforcement learning, Nature 610 (2022) 47–53.

[247] D. J. Mankowitz, A. Michi, A. Zhernov, M. Gelmi, M. Selvi, C. Paduraru, E. Leurent, S. Iqbal, J.-B. Lespiau, A. Ahern, et al., Faster sorting algorithms discovered using deep reinforcement learning, Nature 618 (2023) 257–263.

[248] N. Makke, S. Chawla, Interpretable scientific discovery with symbolic regression: A review, Artif. Intell. Rev. 57 (2024) 2.

[249] M. Cranmer, A. Sanchez-Gonzalez, P. Battaglia, R. Xu, K. Cranmer, D. Spergel, S. Ho, Discovering symbolic models from deep learning with inductive biases, Adv. Neural Inf. Process. Syst. 33 (2020) 17429–17442.

[250] R. J. Kee, F. M. Rupley, J. A. Miller, Chemkin-II: A Fortran chemical kinetics package for the analysis of gas-phase chemical kinetics, Sandia National Laboratory Technical Report SAND-89-8009 (1989).