Generalizable Data-driven Model Augmentations Using Learning and Inference assisted by Feature-space Engineering

by

Vishal Srivastava

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in the University of Michigan
2022

Doctoral Committee:

Associate Professor Karthik Duraisamy, Chair
Professor Krzysztof Fidkowski
Associate Professor Eric Johnsen
Professor Kenneth Powell
Dr. Christopher L. Rumsey, NASA Langley

Vishal Srivastava

vsriv@umich.edu

ORCID iD: 0000-0002-6049-9929

Dedicated to my parents

Acknowledgements

I will begin by thanking my advisor, Professor Karthik Duraisamy, for guiding me throughout my time as a doctoral student at the University of Michigan, Ann Arbor. I owe him for his unwavering support, patience, understanding, and above all his trust in my abilities, especially during a period of several months when I was exploring different research problems and had not made any tangible progress. He gave me ample freedom to explore different research areas and as a result I gathered knowledge and skills across a wide variety of areas within computational science throughout my time as a PhD student. At the same time, he kept me on track and let me know of my weaknesses which helped me become a better researcher overall. In short, I could not have asked for a better PhD advisor.

Before moving on, I would be remiss if I do not express my deep gratitude towards Professor Tapan K. Sengupta, who introduced me to the world of numerical analysis and computational simulations when I was an undergraduate student. While I was keenly interested in research and had carried out research projects before taking the courses that he offered, I had not made up my mind about the research area I would like to pursue. His courses on ”Computational Fluid Dynamics” and ”Transition and Turbulence” made me fall in love with computational science. His passion for the subject, depth of knowledge and sheer brilliance made a deep impression on me. While he can be very strict and set in his ways (I am still a bit intimidated by him, if I am being honest), he continues to inspire me with the discipline and rigor with which he conducts research – qualities that I have yet to master. If I had to choose one individual who I attribute all my current and future achievements as a researcher to, it would be him.

I would like to thank the amazing people I met at the Computational Aerosciences Laboratory – shout-outs to Anand, Eric, Ayoub, Nicholas (Nick), Shaowu, Christopher (Chris), Rajarshi (Raj), Aniruddhe, Christian, Jasmin and Sahil for all the fun experiences and conversations we had (not many of which were related to research). I would especially like to thank Nick for being a great friend and being always there in my times of need. Among the several times he has helped me, I will never forget him staying with me in the emergency room until 3 AM when I broke my ankle and then giving me rides for follow-up appointments.

I was very lucky to have the opportunity to visit the National Institute of Aerospace (NIA) in 2019 and meet Dr. Christopher Rumsey, Dr. Gary Coleman, and Dr. Ponnampalam Balakumar. Though I was there only for three days, I had a glimpse into life as a research scientist. I was greatly motivated by their unquenchable eagerness to learn about upcoming techniques and the composure and meticulousness with which they approach problems. I would especially like to thank Dr. Rumsey for the discussions we had during and after my visit to NIA, a couple of which, directly or indirectly, matured into ideas in this thesis (e.g., hierarchical augmentations).

The work presented in this thesis was supported by funding from Defense Advanced Research Projects Agency (DARPA) under the EQUiPS (Enabling Quantification of Uncertainties in Physical Systems) program, Office of Naval Research (ONR), Toyota Motor Engineering and Manufacturing North America, and Advanced Research Projects Agency-Energy (ARPA-E) under the DIFFERENTIATE (Design Intelligence Fostering Formidable Energy Reduction and Enabling Novel Totally Impactful Advanced Technology Enhancements) program. I would also like to thank my dissertation committee – Professor Karthik Duraisamy, Professor Krzysztof Fidkowski, Professor Eric Johnsen, Professor Ken Powell and Dr. Christopher Rumsey – for their valuable feedback on this thesis to help make it better.

As far as living in Ann Arbor goes, I was very fortunate to have Aditya as my roommate for the first five years. I am horrible at cooking and had it not been for him, I would have ordered food in everyday (as I have done in the last few months of my PhD). While I enjoy working on various personal projects in solitude, we had a lot of fun and interesting conversations ranging from politics and history to anime and movies during the time that I used to be outside my room. He also persuaded me (quite strongly at times) to go on a few trips in and outside the city – experiences that I appreciate in retrospect and which I would never have had if it were not for him.

I would like to thank my parents for their constant encouragement, motivation and emotional support. They have always pushed me to strive for excellence and did their best to help me achieve the goals that I set for myself. As a matter of fact, they still encourage me to expand my horizons and step outside my comfort zone. I shall always remain indebted to them for the considerable sacrifices they made to ensure that I got the best of everything that I needed as a child.

Last but definitely not the least, I am very grateful to my girlfriend, Shruti, for a hundred different reasons but above all, for making even the most difficult days in these past few years feel like a breeze.

I am sure that I will continue to improve both as a researcher and as a person from the experiences I had and the lessons I learned as a doctoral student and I shall always cherish the time I spent at the University of Michigan.

Table of Contents

List of Tables

List of Figures

List of Appendices

Abstract

The recent emergence of data science as a scientific tool has led to the development of data-driven approaches to create new computational models, and to improve existing models. In many complex applications such as turbulent flow modeling, however, modeling accuracy can deteriorate drastically when these models are applied in physical conditions and geometries that are significantly different compared to those used in training. In other words, models are not adequately generalizable, which in turn, severely limits the scope of their applicability.

This work proposes a new class of approaches for generalizable physics-constrained data-driven modeling. In contrast to parameter calibration, corrections are made to the structure of the model by inferring augmentation functions using high-fidelity data. The augmentations are inferred as functions of local modeled quantities, referred to as features. In particular, two different algorithms are proposed and applied to practical problems.

The first formalism, named "Learning and Inference assisted by Feature-space Engineering (LIFE)", based on strongly-coupled integrated inference and machine learning, introduces methodological and algorithmic innovations to facilitate the inference of generalizable, robust and modular augmentations. The LIFE framework offers tools and guiding principles to help modelers design a feature-space that is conducive for inference of generalizable augmentations by ensuring that the features have an appropriate functional form such that the feature-space remains bounded and the physical conditions pertaining to the inadequacy under consideration span the bulk of feature-space. The set of techniques presented in this framework facilitate **localized learning**, which ensures that during a given inference iteration, augmentation behavior remains unchanged in regions within feature-space where no data is available. Localized learning makes the

augmentation more robust as it prevents spurious predictions and ensures that the augmented model falls back to its baseline behavior for unseen physical conditions. Lastly, the meticulous feature-space design and localized learning also pave way for **hierarchical augmentations**, i.e., augmentations introduced within an already augmented model to add capabilities that the original augmentation cannot provide. This allows a modeler to infer several levels of augmentation in decreasing order of generalizability (increasing order of specificity for a certain class of problems).

To demonstrate its capability, the LIFE framework is used to learn an augmentation introduced within a bare-bones intermittency transport equation (which itself was introduced into a variant of Wilcox's 1988 k-ω turbulence model) to predict bypass transition. After training on just **two** flat plate cases from the ERCOFTAC's T3 dataset, the inferred augmentation shows generalizably improved predictive accuracy across a range of different geometries (flat plates, turbine cascades and compressor cascades) and inflow conditions (significantly different Reynolds numbers, Mach numbers, freestream turbulence intensities, pressure gradients, etc.). This inference involves solving large inverse problems with tens/hundreds of thousands of degrees of freedom and leverages adjoint-based optimization. To improve the predictions for separation-induced bypass transition, a hierarchical augmentation is introduced along with the previously inferred augmentation.

While the LIFE approach performs well when the feature-space is carefully designed, the feature design process might not always be tractable. The problem is exacerbated further when the feature-space is high-dimensional, the augmentation function is highly nonlinear, and feature-space is sparsely populated with data. Embedding the augmentation function within a complex numerical solver might have considerable time and effort requirements, and hence can also present itself as a concern. In the light of these potential issues, a weakly-coupled Integrated Inference and Machine Learning (IIML) algorithm was developed, which offers the same benefits of learnability and consistency as the strongly-coupled IIML framework. A non-intrusive implementation was also developed wherein the augmentation function is created externally and only the spatial field of augmentation values needs to be passed to the numerical solver. Together, the

non-intrusive implementation and weakly-coupled IIML, facilitate a relatively effortless setup to solve the inference problem. Each inference iteration begins with updating the inadequacy fields for all the training cases independently (similar to an iteration performed during field inversion). This is followed by a machine learning step where the discrepancy between these newly updated fields and augmentation predictions is minimized. Finally to maintain consistency of the augmentation field with the converged results corresponding to the learned augmentation function, the model is solved again using the augmentation function for each training case and the resulting augmentation field is treated as the optimization iterate.

The viability of the IIML approach is demonstrated by inferring an augmentation for polymer electrolyte membrane fuel cells (PEMFCs) to improve the membrane water content predictions. The membrane water content is a critical control variable to ensure peak working efficiency as too dry or too wet conditions can lead to considerable deterioration in performance. The obtained augmentation showed some extent of generalizability in cases sharing the same geometry but different operating conditions. In addition to improved water content predictions, the current density predictions also improve significantly. Taken together, the LIFE and IIML approaches offer the modeler a choice between high generalizability/robustness, and minimal implementation effort, respectively.

Chapter 1

Overview

Analysis and optimization of complex physical systems constitute the central theme across most, if not all, engineering disciplines. Numerical models present an effective alternative for such applications, provided that the corresponding simulations offer sufficient predictive accuracy within reasonable computational costs. Traditional development of such models has relied on theoretical insight, empiricism and intuition.

While theoretical insight and intuition are aspects that remain unique to human intelligence, formal numerical techniques can be used to extract intricate empirical relationships within a model from available data. Experiments and high-fidelity simulations constitute the sources of information that can provide such data. Note here that obtaining data from either of these sources is both time- and resource-intensive, thus limiting its availability. This thesis presents guiding principles and techniques for use by expert modelers to infer complex parametrizations from limited data with the potential to generalize well to unseen configurations.

One of the most challenging problems, and of vital importance in engineering, is to model turbulence in fluid flows. Turbulence is a complex physical phenomenon that occurs across a multitude of scales separated by several orders of magnitude in both space and time. Accurate modeling of turbulence is critical for analysis, design and optimization of flow paths, as it can lead to enhanced mixing, which can further result in higher rates of momentum and heat transfer – quantities that can directly affect performance of the system in consideration. In order to alleviate computational costs of direct numerical simulations (DNS), several reduced-fidelity models have been developed in the past few decades. Two major categories of such models are large eddy simulations (LES) and Reynolds-Averaged Navier-Stokes (RANS) simulations. While these models attempt to mimic the original system behavior, they tend to suffer from significantly inaccurate predictions, especially for complex configurations.

Recent advances have enabled the development of data-driven frameworks to create new low-fidelity models or modify existing ones by learning intricate functional relationships from available high-fidelity data, to achieve better predictive accuracy. However, in general, these "learned" models tend to poorly generalize to system configurations considerably different from the ones used for training. In a number of cases, such models provide poor predictions even for those canonical cases that were used to calibrate the existing empirical models. This behavior can mainly be attributed to errors resulting from extrapolation, over-fitted models, and/or inadequacies in the functional structure used for learning.

Although it might be impossible to develop truly general low-fidelity models that provide reasonably accurate solutions for any configuration, efforts can be made towards extending the applicability of data-driven reduced-fidelity models by designing them to be as robust and generalizable as possible. While the methods discussed in this thesis can be applied to improve any reduced-fidelity model that can be written as a system of partial differential equations (PDEs), their development was mainly driven by the need to improve turbulence models for RANS simulations.

This chapter begins by describing the "closure" problem that appears when a reduced-fidelity model is constructed by coarse-graining the true governing equations (which in this context refers to reducing the degrees of freedom via phenomenological approaches) for any physical phenomena. Thereafter, LES and RANS equations are discussed in the context of coarse-graining the Navier-Stokes equations. This is followed by some major classes of model closures developed for RANS simulations in the past few decades. A brief commentary on the different data-driven modeling techniques developed over the past few years is then provided along with their advantages and limitations. Finally, the contributions and the structure of this thesis are laid out.

1.1 The closure problem

Consider a physical system described on a spatial domain Ω, the governing equations for which can be represented with appropriate boundary conditions as follows

$$\frac{\partial \boldsymbol{q}}{\partial t} + \mathscr{R}(\boldsymbol{q}) = 0 \quad \forall \quad \boldsymbol{x} \in \Omega \tag{1.1}$$

where $\boldsymbol{q}(\boldsymbol{x}, t)$ represents the field of state variables that is fully-resolved in space and time. To reduce the computational cost, one may choose to reduce the number of states that need to be solved for via a coarse-graining operation (e.g., spatial filtering, ensemble averaging, etc.). However, coarse-graining results in loss of information. Hence, there is a component of the true states of the system that remains unresolved. Thus, the true state variables can be decomposed into coarse-grained and fine-scale parts, represented by $\widetilde{\boldsymbol{q}}$ and $\widehat{\boldsymbol{q}}$, respectively. Performing the coarse-graining operation on the equations representing the high-fidelity system results in an unknown closure term $\mathscr{N}(\boldsymbol{q})$, arising from any non-linearities in the operator $\mathscr{R}$ and the resulting unaccounted contributions from $\widehat{\boldsymbol{q}}$. Note that the following representation is not an approximation.

$$\frac{\partial \widetilde{\boldsymbol{q}}}{\partial t} + \widetilde{\mathscr{R}(\boldsymbol{q})} = 0 \qquad \Rightarrow \qquad \frac{\partial \widetilde{\boldsymbol{q}}}{\partial t} + \mathscr{R}(\widetilde{\boldsymbol{q}}) + \mathscr{N}(\boldsymbol{q}) = 0 \tag{1.2}$$

A reduced-fidelity model is designed to make use of only the coarse-grained states, $\widetilde{\boldsymbol{q}}$, in order to reduce the computational costs. To achieve this, the function $\mathscr{N}$ needs to be approximated in the model by a function of only coarse-grained quantities, $\mathscr{N}_m$. This results in the set of coarse-grained equations for the reduced-fidelity model as follows.

$$\frac{\partial \widetilde{\boldsymbol{q}}_m}{\partial t} + \mathscr{R}(\widetilde{\boldsymbol{q}}_m) + \mathscr{N}_m(\widetilde{\boldsymbol{q}}_m) = 0 \quad \forall \quad \boldsymbol{x} \in \widetilde{\Omega} \tag{1.3}$$

Note here that $\widetilde{\Omega}$ represents the correspondingly coarse-grained version of the domain, and $\widetilde{\boldsymbol{q}}_m$ represents the solution to the reduced-fidelity model which may be different from the true coarse-grained solution $\widetilde{\boldsymbol{q}}$, owing to the approximated function $\mathscr{N}_m$. The closure problem refers to the problem of designing a function $\mathscr{N}_m$, such that the $\widetilde{\boldsymbol{q}}_m$ approximates $\widetilde{\boldsymbol{q}}$ as closely as possible. Secondary variables $\widetilde{\boldsymbol{s}}_m$ can often be introduced into the model which might help in approximating the closure term as $\mathscr{N}_m(\widetilde{\boldsymbol{q}}_m, \widetilde{\boldsymbol{s}}_m)$. The additional equations required to close the system can be given as follows.

$$\frac{\partial \widetilde{\boldsymbol{s}}_m}{\partial t} + \mathscr{G}_m(\widetilde{\boldsymbol{s}}_m, \widetilde{\boldsymbol{q}}_m) = 0 \quad \forall \quad \boldsymbol{x} \in \widetilde{\Omega} \tag{1.4}$$

The construction of operators $\mathscr{N}_m$ and $\mathscr{G}_m$ is a tedious and meticulous process which has evolved over decades through a combination of physical insight, mathematics and empiricism. In this work, the focus is restricted to steady-state reduced-fidelity models, which, using the terminology described above, can be represented as

$$\mathscr{R}(\widetilde{\boldsymbol{q}}_m) + \mathscr{N}_m(\widetilde{\boldsymbol{q}}_m, \widetilde{\boldsymbol{s}}_m) = 0 \quad ; \quad \mathscr{G}_m(\widetilde{\boldsymbol{s}}_m, \widetilde{\boldsymbol{q}}_m) = 0 \tag{1.5}$$

For ease of notation, we shall refer to this system of equations in a compact manner $\mathscr{R}_m(\widetilde{\boldsymbol{u}}_m; \boldsymbol{\xi}) = 0$, with the state variables and secondary variables combined into a single vector of model variables $\widetilde{\boldsymbol{u}}_m = \begin{bmatrix} \widetilde{\boldsymbol{q}}_m^T & \widetilde{\boldsymbol{s}}_m^T \end{bmatrix}^T$, and the inputs to the model (discretized domain, boundary conditions, etc.) embedded into the notation via $\boldsymbol{\xi}$.

1.1.1 Estimating model inadequacies

Any discrepancy between $\mathcal{N}(\boldsymbol{q})$ and $\mathcal{N}_m(\widetilde{\boldsymbol{u}}_m)$ is referred to as model inadequacy. Note that there are several ways of expressing the model inadequacy e.g., $\mathcal{N}(\boldsymbol{q}) - \mathcal{N}_m(\widetilde{\boldsymbol{u}}_m)$, $\mathcal{N}(\boldsymbol{q})/\mathcal{N}_m(\widetilde{\boldsymbol{u}}_m)$, or a more complex function. The model inadequacy as a whole might consist of several contributing inadequacies that only arise for a specific range of physical conditions. Except for a few simplified instances, an exact functional form for the model inadequacy is impossible to obtain in terms of the coarse-grained quantities. However, even an approximate version of this function (hereafter termed as an "augmentation" function), when introduced within the model, could help in improving its predictive accuracy significantly. There exist several techniques, including the ones in this thesis, that can be used to infer such model augmentations from high-fidelity data. However, these augmentations are not posed as direct functions of the model states. This is because prohibitively complex functional forms might be needed to infer and learn the corresponding functional relationship from data. Instead, they are posed as functions of some chosen local quantities called "features", $\boldsymbol{\eta}(\widetilde{\boldsymbol{u}}_m)$, which in turn are functions of model states, in the hope that the functional form would be relatively simpler. Hereafter, the augmentation functions are denoted as $\beta(\boldsymbol{\eta}(\widetilde{\boldsymbol{u}}_m))$.

1.2 Coarse-graining the Navier-Stokes equations

1.2.1 Large Eddy Simulations (LES)

Large eddy simulations aim to resolve flow structures (referred to as eddies) of orders of magnitude larger than or similar to the mesh spacing, while neglecting the sub-grid scale (SGS) ones. Functionally, the corresponding coarse-graining of the true solution is achieved by applying a spatial filter to the Navier-Stokes equations. The incompressible LES equations are, then, given as follows.

$$\begin{gathered}\frac{\partial \widetilde{u}_i}{\partial x_i} = 0 \\ \frac{\partial \widetilde{u}_i}{\partial t} + \widetilde{u}_j \frac{\partial \widetilde{u}_i}{\partial x_j} = -\frac{\partial \widetilde{p}}{\partial x_i} + \nu \frac{\partial^2 \widetilde{u}_i}{\partial x_j \partial x_j} + \frac{\partial \tau_{ij}^{SGS}}{\partial x_j}\end{gathered} \tag{1.6}$$

Here, $\widetilde{\cdot}$ denotes the filtering operation which can be represented by the following generic convolution operation where G is an appropriate convolution kernel chosen by the user.

$$\widetilde{u}_j(\boldsymbol{x}) = \iiint G(\boldsymbol{x} - \boldsymbol{x}')u_j(\boldsymbol{x}')d\boldsymbol{x}' \tag{1.7}$$

The term shown in blue represents the closure term in the model representing the contribution of subgrid scale stress, $\tau_{ij}^{SGS} = \widetilde{u_i u_j} - \widetilde{u}_i\widetilde{u}_j$. The first term in this expression for τ_{ij}^{SGS} is clearly a function of the fully resolved velocity field. Thus, a closure for τ_{ij}^{SGS}

needs to be designed. There exists a class of closure models which approximate the SGS stress tensor analogous to the viscous stress tensor, using a modeled quantity referred to as the eddy viscosity (ν_t) instead of the molecular viscosity (ν) as follows.

$$\tau_{ij}^{SGS} = 2\nu_t \widetilde{S}_{ij} \tag{1.8}$$

Here, $\widetilde{S}_{ij}$ is the mean strain rate tensor, defined as $\widetilde{S}_{ij} = \frac{1}{2}\left(\frac{\partial \widetilde{u}_i}{\partial x_j} + \frac{\partial \widetilde{u}_j}{\partial x_i}\right)$. Appropriately, these kinds of closures to the LES equations are termed as eddy viscosity models. The Smagorinsky [68] model is an example of an eddy viscosity model which approximates the eddy viscosity as $\nu_t = C_s \Delta^2 \sqrt{2\widetilde{S}_{ij}\widetilde{S}_{ij}}$, where C_s is a model constant and Δ is a measure of the local mesh spacing. The model needs to depend on the mesh spacing as the coarse-graining operation (filtering) depends on the mesh spacing as well. Even with this kind of coarse-graining, performing an accurate LES simulation still requires significant computational resources/time, thus rendering LES simulations infeasible for extensive use in design and optimization at present or in the near future.

1.2.2 Reynolds-Averaged Navier-Stokes (RANS) simulations

Reynolds-averaging decomposes any flow quantity ϕ as $\phi = \overline{\phi} + \phi'$, where $\overline{\phi}$ is the ensemble average of ϕ, and ϕ' denotes the fluctuations. An important property of ensemble averaging is that $\overline{\phi'} = 0$. Applying Reynolds-averaging to the Navier-Stokes equations, one obtains the RANS equations as follows.

$$\begin{gathered}\frac{\partial \overline{u}_i}{\partial x_i} = 0 \\ \frac{\partial \overline{u}_i}{\partial t} + \overline{u}_j \frac{\partial \overline{u}_i}{\partial x_j} = -\frac{\partial \overline{p}}{\partial x_i} + \nu \frac{\partial^2 \overline{u}_i}{\partial x_j^2} + \frac{\partial \tau_{ij}}{\partial x_j}\end{gathered} \tag{1.9}$$

The term shown in blue is the closure term, which accounts for contributions from the Reynolds stress tensor, $\tau_{ij} = -\overline{u_i' u_j'}$. Note here that while the system of equations for LES is consistent with Navier-Stokes equations, i.e., the equations transform into the Navier-Stokes equations at infinitesimally small mesh-spacing and time-step, the RANS equations do not exhibit such asymptotic consistency. Owing to their considerably lower computational costs, RANS simulations are the method of choice for preliminary design and optimization of flow configurations. Several classes of closure models exist for the Reynolds stress tensor, and these are given 1.3.

1.3 Closure Models for RANS Equations

1.3.1 Eddy-viscosity models

Eddy viscosity closures for RANS simulations rely on the hypothesis that owing to their diffusive nature, the Reynolds stresses can be assumed to behave similarly to the viscous stresses. This is known as Bousinessq's hypothesis, which can be mathematically written as follows.

$$-\overline{u_i'u_j'} = \nu_t \left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i} - \frac{2}{3}\frac{\partial \overline{u}_k}{\partial x_k}\delta_{ij} \right) - \frac{2}{3}k\delta_{ij} \tag{1.10}$$

Here, $k = \overline{u_m'u_m'}/2$ is referred to as the turbulence kinetic energy. While Bousinessq's hypothesis reduces the complexity of these models by reducing the number of unclosed quantities from the 6 Reynolds stress components to just the eddy viscosity, the underlying assumptions are problematic for two main reasons. Firstly, since the molecular timescale is typically smaller compared to the turbulence time scale, the assumption that the Reynolds stresses depend on the mean strain rate is, in general, invalid. Secondly, the hypothesis assumes that the Reynolds stresses are isotropic, which is incorrect. For instance, the blocking effects near the wall dampen the wall-normal Reynolds stress component much more than the other components. Despite these flaws, eddy-viscosity models can provide sufficiently accurate predictions for attached flows on simple geometries. This, along with their simplicity, makes them the closure models of choice for RANS simulations. Several sub-classes of eddy viscosity models exist to evaluate the eddy viscosity (ν_t) in different ways, three of which are described as follows.

Algebraic models

These are the simplest eddy-viscosity models which relate the eddy viscosity directly to flow quantities based on dimensional analysis. Prandtl [55] assumed the turbulence velocity scale as $\ell_m \left|\frac{\partial \overline{u}}{\partial y}\right|$, where ℓ_m is a mixing length that acts as the turbulence length scale, and y is the spatial coordinate along the wall-normal direction. Correspondingly, he approximated the eddy viscosity as shown in eqn. 1.11.

$$\nu_t = \ell_m^2 \left|\frac{\partial \overline{u}}{\partial y}\right| \tag{1.11}$$

However, the values for ℓ_m can vary with spatial location and also depend on the flow geometry and boundary conditions. Estimation of ℓ_m might even be intractable for complex geometries. Hence, Prandtl's model is incomplete in this sense. Other examples of algebraic models include the Cebeci-Smith model [69] and the Baldwin-Lomax model [6], where the eddy-viscosity is modeled differently in the inner and outer layers of the turbulent boundary layer and a damping function is used to ensure that τ_{ij} vanishes at the wall with the correct slope.

Algebraic models, by design, are not equipped to handle effects of flow history. To include these effects, additional PDEs can be solved for some modeled transport quantities along with the RANS equations. These modeled quantities can then be used along with mean flow quantities to calculate eddy-viscosity values. Such models are broadly categorized based on the number of transport variables that are introduced. A few widely used one- and two-equation models are briefly discussed as follows.

One-equation models

One equation models were introduced to include flow history effects in eddy viscosity models. A modeled scalar transport quantity, ϕ, is chosen for such models, the transport equations for which is specified in the following form.

$$\frac{\partial \phi}{\partial t} + \overline{u}_i \frac{\partial \phi}{\partial x_i} = \mathcal{P}_\phi - \mathcal{D}_\phi + \mathcal{T}_\phi \tag{1.12}$$

The terms $\mathcal{P}_\phi$, $\mathcal{D}_\phi$ and $\mathcal{T}_\phi$ represent the production, dissipation and transport of the quantity ϕ, respectively. While these terms are written separately for interpretability, it is, in fact, the balance of these terms that characterizes the behavior of the model. Thus, it is not unusual for each term to be inaccurate on its own but compensated by the other terms in whole.

In one of the first one-equation models, Prandtl [80] modeled the turbulence kinetic energy as a transport variable, the true governing equation for which is given in eqn. 1.13.

$$\frac{\partial k}{\partial t} + \overline{u}_i \frac{\partial k}{\partial x_i} = \tau_{ij} \frac{\partial \overline{u}_i}{\partial x_j} - \varepsilon + \frac{\partial}{\partial x_i} \left[\nu \frac{\partial k}{\partial x_i} + \frac{1}{2} \overline{u'_j u'_j u'_i} + \frac{1}{\rho} \overline{p' u'_j} \right] \tag{1.13}$$

On the right hand side of eqn. 1.13, the three terms in their respective order are the production, dissipation and transport of turbulence kinetic energy. The dissipation, ε, is given by the correlation $\nu \overline{\frac{\partial u'_i}{\partial x_j} \frac{\partial u'_i}{\partial x_j}}$. Within the transport term, the three terms refer to the molecular diffusion, turbulent transport and pressure diffusion, respectively. Modeling these terms in terms of resolved variables is necessary to obtain a model form similar to eqn. 1.12 with $\phi = k$. Using the gradient diffusion hypothesis, the turbulent transport and pressure diffusion can be modeled similarly to molecular diffusion, which results in the following expression for the modeled transport term ($\mathcal{T}_k$),

$$\mathcal{T}_k = \frac{\partial}{\partial x_i} \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_i} \right]$$

where σ_k is a model constant. τ_{ij} appearing in the production term is readily modeled using the Boussinesq hypothesis. Finally, based on dimensional analysis, Prandtl proposed the following approximation for the dissipation term $\mathcal{D}_k$,

$$\mathcal{D}_k = C_D k^{3/2} \ell_m$$

where ℓ_m is a turbulence length scale, and C_D and σ_k are model constants. The eddy viscosity is defined in this model based on dimensional analysis as $\nu_t = c k^{1/2} \ell_m$, where c is a constant value. Note here that in the presence of all these simplifications and approximations, the behavior of the modeled turbulence kinetic energy will be significantly different from the actual turbulence kinetic energy. Given the use of ℓ_m, this model is, again, incomplete. Emmons [19] independently proposed a similar model.

Nee and Kovasznay [48] presented a one-equation model which directly modeled the transport of eddy viscosity. This model still required a turbulence length scale within the transport equation which rendered it incomplete. Baldwin and Barth [5] used a modified turbulence Reynolds number, $\widetilde{R}_T$, to define the following transport quantity for their one-equation model,

$$\nu \widetilde{R}_T = \frac{\nu_t}{C_\mu D_1 D_2} \tag{1.14}$$

where C_μ is a model constant, and D_1 and D_2 are functions of wall distance. Empirical correlations and turbulent boundary layer analysis are used to define the production, dissipation and transport terms for the corresponding transport equation. Since the transport quantity is dimensionally consistent with and can be directly transformed to the eddy viscosity (ν_t), this model precludes the need for an additional velocity, length or time scale and is, hence, complete. A major limitation of this model is that the dissipation is proportional to gradients of ν_t and $\overline{R}_T$, which in effect prevents any streamwise decay of eddy viscosity in the freestream. Another drawback of this model is its destabilizing effect on numerical stability for certain free-shear flows. In his book, Wilcox [92] notes that the Baldwin-Barth model predicts a sharp discontinuity within mixing layers and jets, making the numerical implementation unstable for any grid resolution.

Perhaps the most notable and widely used one-equation model is the Spalart-Allmaras model [70]. Similar to the Baldwin-Barth model, the eddy viscosity is dimensionally consistent with and an explicit function of the Spalart-Allmaras transport variable, $\widetilde{\nu}$. Calibrated using simple metrics from fully developed mixing layer, far wake, and zero pressure gradient flow over a flat plate, coefficients within the production, dissipation and transport terms in this model are designed such that the log law is respected. Damping functions are used to modulate the dissipation close to the wall, and the model behavior below the log layer. As the dissipation term in the Spalart-Allmaras model is inversely proportional to the wall distance, this model, too, does not predict any streamwise decay in the freestream eddy viscosity. Since its calibration does not include jet-like flows, this model tends to significantly over-predict spreading rates for jets.

Other noteworthy one-equation models include the one proposed by Sekundov et al. [27] and the Wray-Agarwal turbulence model [29].

Two-equation models

Two-equation models use two additional modeled quantities (one of which is nearly always k), the transport equations for which can be written similar to eqn. 1.12. Kolmogorov [35] was among the first to use a second transport equation in addition to the one for k, to evaluate the specific dissipation, $\omega = \varepsilon/k$. Several researchers including Chou [10], Rotta [59], Zeierman and Wolfshtein [96] tried different scalar quantities for the second transport equation based on the turbulence length, dissipation and time scales and then related them, along with k, to the eddy viscosity (ν_t) via dimensional analysis. Similar to the true transport equation for the turbulence kinetic energy as shown in eqn. 1.13, the true transport equation for the dissipation of turbulence kinetic energy (ε) can also be derived. Hanjalic, in his thesis [30], modeled the production, dissipation and transport of ε in terms of resolved quantities (including the modeled k and ε) based on dimensional analysis, and Boussinesq and gradient-diffusion hypotheses. These approximations were used in the modeled ε transport equation used by Launder and Spalding in their $k - \varepsilon$ turbulence model [39], which is given in eqn. 1.15,

$$\frac{\partial \varepsilon}{\partial t} + \overline{u}_i \frac{\partial \varepsilon}{\partial x_i} = C_{\varepsilon 1} \frac{\varepsilon}{k} \left(\tau_{ij} \frac{\partial \overline{u}_i}{\partial x_j} \right) - C_{\varepsilon 2} \frac{\varepsilon^2}{k} + \frac{\partial}{\partial x_i} \left[\frac{\nu_t}{\sigma_\varepsilon} \frac{\partial \varepsilon}{\partial x_i} \right] \tag{1.15}$$

where $C_{\varepsilon 1}$, $C_{\varepsilon 2}$ and σ_ε are model constants. The eddy viscosity for such a case can be approximated via dimensional analysis as,

$$\nu_t = C_\mu k^2/\varepsilon \tag{1.16}$$

where, C_μ is a model constant. Many models have been proposed with variants of the same idea over the years. One of these models that is relevant to the current thesis is Wilcox's 1988 k-ω model. ω here refers to the specific dissipation of the turbulence kinetic energy k and can be analytically given as ε/k. Wilcox modeled his production term as $\tau_{ij}\dfrac{\partial u_i}{\partial x_j}$ where τ_{ij} refers to the turbulent shear stress which is calculated in the original version as $(\mu_t S^2 - k\nabla \cdot u)$ where S is the magnitude of the strain rate tensor. A slightly different variation of the model (which is used in this work) uses the vorticity magnitude Ω instead of the strain rate tensor S for the same evaluation.

1.3.2 Reynolds Stress closures

This class of Reynolds transport closures seeks to model the dissipation, pressure strain and turbulent transport terms in the Reynolds stress equations which can be obtaining by taking the first moment of the momentum equations as shown in Eqn. 1.17.

$$\frac{\partial u_i' u_j'}{\partial t} + u_j \frac{\partial u_i' u_j'}{\partial x_j} + \frac{\partial T_{kij}}{\partial x_k} = P_{ij} + R_{ij} - \varepsilon_{ij} \tag{1.17}$$

While the production term is known in the Reynolds stress model, the dissipation tensor ε_{ij}, the pressure-rate-of-strain tensor R_{ij} and the turbulent transport T_{kij} need to be modeled. In addition to transport equations for each of the components of the Reynolds stress, an additional transport equation is needed to provide closure. Usually this transport equation is written for the specific dissipation of TKE (ω). These models provide the capability to account for complex turbulence phenomena involving anisotropy, streamline curvature, rotating flows etc. Some notable model of this variety include those proposed by Launder, Reece and Rodi [7] and Wilcox [92].

1.4 A Brief Review of Data-driven Turbulence Modeling

As seen in the previous section, turbulence modeling has relied on first-principles-based and phenomenological approaches along with assumptions and empirical correlations to choose the model structure, and on data to obtain the model coefficients associated with the chosen structure. Hence, turbulence modeling has always been data-driven to some extent. However, epistemic uncertainties are prevalent within these models, given their approximate nature and limited use of data for calibration. These can be broadly classified into parametric and model-form uncertainties. Parametric uncertainties refer to the uncertainty in the numerical values of *parameters corresponding to a given model-form.* On the other hand, model-form uncertainties arise due to the inadequacies within the model structure. These structural inadequacies are usually a consequence of the approximations and assumptions made while building the model. Hence, even using the "best" model parameters for a given problem could still result in discrepancies between model predictions and data. The relatively obscure nature of model-form uncertainties makes them harder to estimate compared to their parametric counterparts.

The work by Kennedy and O'Hagan [33] proposed a Bayesian calibration framework to address model inadequacies which prepared the groundwork for several uncertainty quantification studies across different disciplines including turbulence modeling [26, 17, 8]. They used Gaussian process models to approximate the model outputs as a function of model states and model parameters, and the model inadequacies as a function of model states. Bayesian inference was used to infer λ as well as parameters of the Gaussian processes. However, as mentioned by Arendt et al [3], one of the major limitations of the framework by Kennedy and O'Hagan is that the resulting solutions are not necessarily identifiable, i.e., the accurate inference of the underlying true inadequacy field is not guaranteed.

The work of Oliver and Moser [50] was among the first few to address the quantification of model-form uncertainties in turbulence modeling, which they did by introducing discrepancies in the Reynolds Stress tensor and modeling them as spatially-dependent Gaussian random fields. Dow and Wang [13] also followed a similar approach to infer the spatial field of eddy viscosity needed to match the DNS velocity fields. The discrepancy between the eddy viscosity field predicted using the k-ω model and the one inferred using DNS data was represented as a Gaussian random field and propagated to estimate uncertainty bounds on the mean flow quantities.

Edeling et al [17] used a different approach to address model-form uncertainties. They used data from flat-plate boundary layers under various pressure gradients and performed a Bayesian calibration to obtain coefficients for different turbulence models. Predictions were then made using Bayesian Scenario Averaging over the individual predictions of this set of turbulence models. They also developed a "smart scenario sensor" that could automatically preferentally weight these predictions, which as they observed, not only resulted in a better mean prediction but also reduced the variance in predictions to the levels of measumerment error during experiments.

Early attempts related to introducing model corrections to alleviate structural inadequacies include the work by Parneix et al. [52] who obtained apriori estimates for the accuracy of second moment closures for computations on a backward facing step geometry. This was done by evaluating the values for one variable at a time while holding other variables constant at values predicted by DNS. The model equations were correspondingly modified to improve predictions on the same case. Raiesi et al. [57] tried a similar approach for one- and two-equation models. However the authors found that substituting the modeled turbulence quantities (such as k and ω) with the values of their high-fidelity counterparts did not improve the predictive accuracy of the model. In fact, for some cases such straightforward substitution led to deterioration in performance. This occurs because there is a considerable difference between the behavior of modeled turbulence variables and their real counterparts.

Tracey et al. [81], in 2013 proposed the idea of transforming inadequacy fields, i.e., functions of space, to functions of features (known local functions of model states). This idea was applied to learn the functional relationship between some chosen features – viz., eigenvalues of the anisotropy tensor, ratio of production-to-dissipation rate of turbulent kinetic energy, and a marker function to mask the free shear layer regions in the flow – and the discrepancies in the eigenvalues of the Reynolds anisotropy tensor $\left(a_{ij} = \overline{u_i' u_j'} - \frac{2}{3}k\delta_{ij}\right)$ between the flow predicted by the RANS solver and those obtained from DNS data. Xiao and coworkers [94] went further and modeled the discrepancies in turbulent kinetic energy and eigenvectors of the anisotropy tensor, in addition to the discrepancies in the eigenvalues while using a broader set of features.

In 2017, Ling and Templeton introduced Tensor Basis Neural Networks (TBNNs) [42] in order to learn a non-linear Reynolds Stress model from data by inferring the coefficients of the tensor basis expansion chosen to approximate these stress tensors. The features used in this work were five invariant quantities calculated using the mean strain rate and vorticity tensors. Since then, TBNNs have been used in different works to model scalar fluxes and turbulent heat fluxes.

A different class of methods based on symbolic regression techniques to obtain such functional relationships between features and inadequacies have also been introduced in several works. Weatheritt and Sandberg [89] used genetic programming methods to construct closures for the Reynolds anisotropy tensor in terms of invariant quantities derived from the velocity gradient tensor. Schmelzer et al. [62] used sparse regression techniques over a library of some chosen candidate functions (features) to obtain an algebraic Reynolds stress closure. An advantage of taking this route to infer data-driven models is the resulting simplicity and interpretability of the obtained model-forms.

In their original form, the aforementioned techniques for extracting model inadequacies do not explicitly enforce model consistency. In other words, the inadequacy inferred from these techniques is not necessarily consistent with the behavior of the model itself as the high-fidelity quantities which are directly used in the inference process may have a significantly different behavior compared to their modeled counterparts. For instance, the turbulence kinetic energy in a DNS can behave very differently when compared to the modeled turbulence kinetic energy in a RANS simulation. Hence, while an inconsistent inference of model inadequacy can work well for the cases used to perform the inference (training cases), this inherent incompatibility prevents such a model-form for inadequacy to be reliably used for other cases. Several model-consistent frameworks have also been proposed in order to get around this problem, all of which tend to ensure model-consistency via solving an inverse problem to find the optimal model-form for the inadequacy in consideration. A detailed discussion on model-consistency in this context can be found in a recent review [14].

Parish and Duraisamy [51] introduced the field inversion and machine learning (FIML) framework, which has since then been primarily used to improve predictive accuracy of RANS models for different kinds of flows [64, 65, 66]. However, the framework is applicable to virtually any problem involving the use of PDE-based mathematical models. Besides being model-consistent, the FIML framework can also make use of sparsely available high-fidelity field data. FIML infers the inadequacy term in two steps. Firstly, an inverse problem (field inversion) is solved to obtain the optimal spatial field of values corresponding to the inadequacy term. Then, the inadequacy term is hypothesized as a function of some locally-defined flow quantities (features) and the functional form for the same is fixed. A second inverse problem (machine learning) is then solved to obtain optimal parameters within this functional form. Although the inadequacy functions inferred using FIML are observed to work better compared to the baseline models on flows with geometries and boundary conditions similar to those in the training configurations, the accuracy deteriorated for flows with significantly different ones. Holland et al. [31, 32] further proposed an improvement in the framework where the two inference problems can be integrated into one. This integrated version of FIML, in general, considerably reduces the loss in information during the machine learning step and hence ensures better "learnability" during the process. As the framework proposed in this thesis heavily relies on it, FIML will be discussed in considerable detail in chapter 2. Franceschini et al. [20] used variational data assimilation, an idea similar to field inversion, to infer scalar and vector inadequacy corrections to the scalar- and momentum-transport equations in the Spalart Allmaras model, respectively using dense and sparse measurements from the velocity field. Similar to the idea of classic FIML, Volpiani et al. [85] built on this and added a machine learning step following data assimilation to obtain functional forms for a vector inadequacy term in the momentum-transport equation using DNS data for periodic hill geometries exhibiting separated flows. Strofer and Xiao [73] presented an end-to-end differentiable framework to formulate data-driven turbulence models. In their framework, a neural network is used to predict the coefficients in the integrity basis expansion of the anisotropy tensor. Since both the solver and the neural network are fully differentiable and since the neural network is embedded within the solver, the framework can make use of adjoint-driven techniques to directly evaluate the sensitivities w.r.t. the weights/biases of the neural network. Gradient-based optimization techniques are then used to minimize the discrepancy between the RANS predictions and high-fidelity data. Model-consistent inference and learning has also been used by Sirignano et al. [67] to introduce subgrid-scale closures in LES equations in the form of deep neural networks. The weights/biases of these deep neural networks are then optimized using sensitivities obtained using, what they refer to as, the stochastic adjoint method.

Other recent works that use symbolic identification to approximate inadequacy terms have also emphasized the importance of model-consistent (or as they call it, CFD-driven) approaches. Saidi et al [61] proposed a CFD-driven symbolic identification approach to obtain data-driven generalized eddy viscosity models for RANS simulations. In their work, they introduced tensor corrections in the Reynolds stress anisotropy tensor and the production term in the turbulence kinetic energy transport equation. They then expressed these tensor corrections in terms of a minimal integrity basis of ten tensors (calculated using strain rate tensor S_{ij} and vorticity tensor Ω_{ij}), the coefficients of which are functions of five invariants. Gene expression programming was subsequently used to obtain optimal functional forms for these coefficients which tend to minimize the discrepancies between the fields of flow quantities obtained from DNS (data) and corresponding RANS simulations (predictions). While noting the significant advantage of not necessitating the use of full numerical fields and second-order statistics unlike the CFD-free version, they reported comparable predictive improvements between the two approaches. Waschkowski et al [88] proposed a similar multi-objective CFD-driven approach where they use the EVE (EVolutionary algorithm for the development of Expressions) framework to infer different discrepancies simultaneously to obtain a nonlinear eddy viscosity model. An interesting observation in the context of data-driven turbulence modeling was made in that work, rightly pointing out that if different corrections in a model are trained independently, the resulting interplay between these corrections when used together would result in inaccurate predictions – an argument which they further use to make a case for simultaneous inference of different inadequacies as the only solution. However, as demonstrated in this thesis, being model-consistent also provides an option of hierarchical inference of different inadequacy terms. In other words, once an inadequacy term for the baseline model has been inferred from data, another inadequacy term can be inferred w.r.t. the new baseline model (containing the already inferred inadequacy term). Since the framework is model-consistent, the new inadequacy term will automatically be compatible with the one already present in the model. Thus, hierarchical inference of different inadequacies can work as an alternative to simultaneous inference.

1.5 Limitations of previous work

While Field Inversion and Machine Learning (FIML) provides a model-consistent framework to extract usable augmentations from high-fidelity data, it suffers from the following limitations:

- **Limited learnability from inferred results:** Even when the field inversion extracts augmentation fields, which when injected into the model can predict very accurately for the corresponding training case, the machine learning step might not be able to recreate these fields due to the following reasons:

    1. **Poor correlation between the features and the augmentation:** If the chosen features are poorly correlated with the augmentation, the augmentation might end up having very different values for feature values which are quite close to each other. If the chosen functional form is not expressive enough, the augmentation function will be significantly inaccurate. If the functional form is expressive enough, the augmentation function will be very noisy (as a consequence of high gradients) and the solver might anyway predict with a lower accuracy. The solver convergence will also suffer as the slightest changes in state values made via time-stepping might result in large changes to the augmentation which will translate into large residuals.

2. **Inconsistency across training cases:** Since the field inversion procedure is performed independently on each dataset and since significantly different augmentation fields can result in similar improvements in accuracy, even if the features correlate well with the augmentation for each training case, these correlations themselves may be inconsistent across cases. Thus, this will result in a cumulative deterioration in the correlation between the features and the augmentation and result in similar problems as mentioned in the previous point.

3. **Expressivity of the augmentation:** The functional form chosen for the augmentation might be inadequate to represent the functional relationship between the features and the augmentation. In such a case, the training error will never reduce beyond a certain value as the augmentation would be structurally incapable of achieving more accurate results, thus rendering a part of the inferred results unlearnable.

- **Limited generalizability:** Assuming that an augmentation function captures enough information from the field inversion results, its applicability is usually restricted to cases with geometries and boundary conditions similar to those in the training dataset for the following two reasons:

  1. **Minimal constraints on feature design:** The FIML framework specified that the features ought to be non-dimensional, local and Galilean-invariant. However, these do not constitute enough conditions to ensure generalizability. For instance, consider the following questions. How does one decide the number of features to be used? How does one decide the method for non-dimensionalizing the features? How does one decide the amount of data required to create a generalizable augmentation with desirable predictive capabilities? These concerns are not addressed in the FIML methodology in detail.

  2. **No control over augmentation behavior:** While the FIML methodology

is agnostic to the functional forms used to learn the augmentation, it does not discuss the consequences of using functional forms which are conventionally used in the machine learning community. While an augmentation that derives from such classes of functions can be learned with a decent degree of accuracy in the regions of feature-space where data is available (from field inversion), the augmentation is free to assume any value in the rest of the feature-space. This can lead to spurious behavior for cases which exhibit significantly different physics, in a part or the entirety of the spatial domain.

In addition to these limitations, the FIML framework (and other data-driven modeling techniques) addresses extrapolation as the ability of the model to predict accurately for a case outside the range of geometries and/or boundary conditions used during training. However, this description is technically inaccurate because the augmentation is defined to be a function of the chosen features and hence extrapolation would be defined as making predictions for features which are significantly far from the training datapoints in the feature-space. In this sense, it is impossible to guarantee accuracy under extrapolation without any prior knowledge about the behavior of the augmentation function in feature-space regions that are devoid of any training datapoints.

1.6 Contributions of the Present Work

1. **Learning and Inference assisted by Feature-space Engineering:**

   The main contribution of this thesis is a data-driven model augmentation framework termed as “Learning and Inference assisted by Feature-space Engineering (LIFE)” which can infer model-consistent, generalizable, robust and modular augmentations (see Chapters 3 and 4). Contributions within the LIFE framework are listed as follows:

   (a) **Improving generalizability via feature-space engineering**

   It is important to recognize that the apparent extrapolation of predictive accuracy that any data-driven modeling method in literature provides, is rarely an extrapolation in its true sense. While the physical conditions and/or the geometry of a test case might seem to lie outside the range of the ones that were used for training, the feature values obtained for such a configuration might very well lie within the range of feature values observed in the training set. Hence, although such a case might seem to provide evidence for improved predictive accuracy under extrapolation, it is in fact just a demonstration of good interpolation characteristics in the feature-space - something that modelers have used to their advantage. As mentioned before, it is mathematically impossible to guarantee predictive accuracy without any prior knowledge. However, while a data-driven model cannot be trusted to provide accurate predictions under extrapolation in the feature-space, the feature-space itself can be transformed to bring in as many configurations as possible within the range of interpolation of existing data. This work provides guiding principles to design efficient feature-spaces (combinations of features along with their chosen functional forms and non-dimensionalizations) using expert knowledge (physical understanding), empirical relationships and heuristics to minimize the chances of extrapolation.

(b) **Localized learning**

Even with a well-designed feature-space, the available data might be insufficient to populate all regions inside it. The best one can do for regions with very sparse or no datapoints, in absence of additional information, is to revert to the baseline model behavior. Hence, the learning technique should be flexible enough to learn the augmentation only in regions where data are present and leave the other regions unperturbed, something which is hard to do for traditional learning frameworks like neural networks and decision trees in their original form. This work provides details on implementation of different variants of localized learning along with the advantages and disadvantages associated with each of those.

(c) **Hierarchically modular augmentations**

While generalizability is a desired trait in a reduced-fidelity model, a specific application may require the model to be custom-tuned to predict with high accuracy for some preferred set of geometries and/or physical conditions even if it leads to reduced accuracy for other unrelated cases. Given the feature-space engineering guidelines and localized learning techniques proposed in this work, one can design augmentations for specific applications on top of any existing augmentations which are more generic. This can ensure consistency and robustness by forcing the model to revert to the behavior of the generic augmentation if a case ever arises that requires extrapolation in the feature space. This property can then be used to design several augmentations in increasing levels of specificity, while maintaining consistency with each other. This work demonstrates how hierarchically modular augmentations can be used in the context of transition models.

2. **Weakly-coupled Integrated Inference and Machine Learning (IIML)**

When dealing with a new solver, quickly embedding and setting up the integrated FIML framework can seem to be a daunting task, especially if the code-base is very large. A part of this work describes a novel weakly-coupled IIML strategy (see Chapter 5) that can be used to infer model augmentations while leveraging a non-intrusive and iterative solution strategy to solve the augmented model. Such a solution strategy only requires two nominal changes to the solver code: (1) Reading augmentation values predicted by an externally implemented augmentation function into an array, and (2) Replacing the inadequacy term within the augmented model with the appropriate entries from the array. Note that, for cases where the features cannot be designed to facilitate a one-to-one augmentation function, strongly-coupled IIML might struggle to preserve augmentation behavior in parts of the feature-space across inference iterations. This could result in poor predictive accuracy despite achieving excellent training accuracy. Weakly-coupled IIML, however, offers a more robust alternative in this regard. This technique is demonstrated via application to a model of a polymer electrolyte membrane fuel cell to improve membrane water content predictions.

3. **Robust Design Optimization via Interval-based estimates of Model-form Uncertainty**

   A minor contribution of this thesis (see Appendix A) is a simple yet effective strategy to use the FIML (or any other model augmentation) framework to develop several augmented models which can then be used to approximately quantify the model-form uncertainty using an interval-based estimate. This technique was used for aerodynamic shape optimization of an aircraft engine nozzle under non-linear constraints.

1.7 Organization

This thesis is organized as follows:

- Chapter 2 describes the existing Field Inversion and Machine Learning (FIML) frameworks in necessary detail.

- Chapter 3 describes the guiding principles and techniques presented under the LIFE framework illustrated using a 1D channel flow example.

- Chapter 4 introduces and describes a data-driven transition model inferred using the LIFE framework. Appropriate comparisons are provided to demonstrate both the advantages of LIFE over the existing FIML frameworks and the associated challenges that need to be addressed. A hierarchical model is also inferred using the LIFE framework to improve predictions for separation-induced transition for compressor cascade geometries.

- Chapter 5 introduces and describes a weakly-coupled IIML approach along with a non-intrusive iterative method to solve the augmented model which requires only minimal changes to the solver code. This is demonstrated by augmenting a polymer electrolyte membrane fuel cell (PEMFC) model via the weakly-coupled IIML framework.

- Chapter 6 summarizes the thesis and presents recommendations for future work.

Chapter 2

Background

2.1 Premise

In Section 1.1, it was argued that model-form inadequacy could be approximated (either in part or in full) by an augmentation function $\beta(\boldsymbol{\eta}(\widetilde{\boldsymbol{u}}_m))$ (where $\boldsymbol{\eta}$ are called features). The model-form inadequacy, as a whole, could consist of several smaller inadequacies. These contributing inadequacies might affect predictions within some physical regime and not the other. In most cases, there is little knowledge about the model-form inadequacy and it cannot be guaranteed that physical regimes corresponding to all contributing inadequacies are present in a finite dataset. Consequently, without sufficient prior knowledge, it cannot be guaranteed that the underlying model-form of the inadequacy can be fully inferred from finite data. Consider the following example. A turbulence model might be inaccurate while predicting skin friction under adverse pressure gradients and also while predicting spreading rate of a turbulent jet. The true functional form of the corresponding model-form inadequacy must contain appropriate terms that alleviate both these inaccuracies. However, if a training dataset contains data only from wall-bounded flows, then it is virtually impossible to improve predictive accuracy for a turbulent jet. Similarly, there could be other inadequacies unbeknownst to the user which might remain unaddressed even if the turbulent jet cases are included in the training dataset.

Assuming that we wish to address only the partial inadequacy that is manifested within the training dataset, the first step to quantify it is to choose how exactly it affects the model. To achieve this, an inadequacy term δ is introduced within the model formulation as $\mathscr{R}_m(\widetilde{\boldsymbol{u}}_m; \delta, \boldsymbol{\xi})$. Examples of how such inadequacy terms have been introduced into turbulence models in the available literature include multiplication with the production term of the Spalart-Allmaras model by Singh et al. in [66], addition to the eigenvalues of the Reynolds anisotropy tensor by Xiao et al. [81, 94], etc. Indeed, δ can also represent the vector of coefficients in the tensor basis expansion for the Reynolds anisotropy tensor as proposed by Ling et al. [41]. The following subsections deal with how numerical estimates for δ are obtained from a given configuration, how $\beta(\boldsymbol{\eta})$ can be learned using the inferred δ, and why model-consistent inference is required for predictive use.

2.1.1 Estimating model-form inadequacy

For most practical problems of interest, it is prohibitively difficult (if not impossible) to directly address the true functional form of the inadequacy term δ. It is relatively easier to estimate an optimal spatial inadequacy field $\delta(\boldsymbol{x})$ (consisting of numerical values for the inadequacy term for every discretized spatial location) by solving an inverse problem. However, the term "optimal" has a subjective context here. Depending on the applications that the augmented model is intended for, accurate prediction of some quantities may be more important compared to others. Hence, it is natural to orient this "optimality" such that these quantities of interest (QoI's) are predicted as accurately as possible. The QoI's can be local quantities (e.g., velocity, skin friction, etc.) or integral quantities (e.g., lift coefficient). Hereafter, $\boldsymbol{y}$ and $\boldsymbol{y}_m$ shall refer to the high-fidelity data and model predictions for the QoI's. This can be done by designing an appropriate cost function ($\mathcal{C}$) to quantify the "optimality" of a given $\delta(\boldsymbol{x})$. A simple example for the cost function can be defined as the L_2 norm of the difference between the predictions and true values of QoI's, i.e., $\mathcal{C}(\boldsymbol{y}, \boldsymbol{y}_m) = \|\boldsymbol{y}_m - \boldsymbol{y}\|$. Thus, the inverse problem is transformed into an optimization problem which aims to minimize the cost function. From what we have discussed until now, the following factors affect what part of the inadequacy is being estimated:

- what training configurations are chosen;
- how δ is introduced into the model; and

- how the cost function $\mathcal{C}$ is formulated.

2.1.2 Approximating inadequacies via augmentation functions

The inadequacy field, $\delta(\boldsymbol{x})$, while useful to quantify model-form inadequacy for a given case, cannot be used for predictive improvements on its own. However, if a functional relationship could be extracted between $\delta(\boldsymbol{x})$ and corresponding values of some carefully-chosen features, $\boldsymbol{\eta}(\widetilde{\boldsymbol{u}}_m, \boldsymbol{\zeta})$, this function could replace the $\delta(\boldsymbol{x})$ field in the numerical model for predictive use. Here, $\boldsymbol{\zeta}$ denotes local quantities independent of the state or secondary variables (e.g., distance from the closest wall) which are used to design features. Note that for the augmentation to be usable in a predictive setting with a wide range of applicability, features must strictly consist of local quantities. Note here that quantities such as two-point correlations, gradients, filters, etc. are treated as local quantities as well. The features must be invariant [93, 14] to appropriate transformations (e.g., rotation, as they must not depend on the orientation of the coordinate axes). Replacing the inadequacy term with the augmentation function, the model can be written as shown in Eqn. 2.1.

$$\mathscr{R}_m(\widetilde{\boldsymbol{u}}_m; \beta(\boldsymbol{\eta}(\widetilde{\boldsymbol{u}}_m, \boldsymbol{\zeta}); \boldsymbol{w}), \boldsymbol{\xi}) = 0 \tag{2.1}$$

It should be mentioned here that once the functional form of the augmentation (e.g., a neural network) is chosen, the goal is to infer the augmentation function parameters $\boldsymbol{w}$ that minimize the discrepancy between $\beta(\boldsymbol{\eta})$ and $\delta(\boldsymbol{x})$. Several techniques exist in the machine learning literature to solve this optimization problem.

2.1.3 Significance of Model-consistent Inference

If the full high-fidelity field of state variables is available, a naive way to solve for $\delta(\boldsymbol{x})$ could be to simply plug high-fidelity data into the model equations and estimate the numerical values of the inadequacy at all the spatial locations. However, it should be noted that the modeled states (especially the secondary variables $\widetilde{\boldsymbol{s}}_m$) could behave significantly differently compared to their physical counterparts. Hence, a functional relationship extracted between the features calculated using high-fidelity quantities and the

$\delta(\boldsymbol{x})$ obtained by substituting high-fidelity values into the model equations would not be consistent with the behavior of the model states $\widetilde{\boldsymbol{u}}_m$. As a consequence, the augmented model could predict results which are considerably different compared to those obtained using $\delta(\boldsymbol{x})$.

The solution is to use a model-consistent approach (similar to the one detailed in Section 2.1.1) which requires solving an inverse problem to obtain $\delta(\boldsymbol{x})$ such that the augmented model predicts $\boldsymbol{y}_m$ as close to $\boldsymbol{y}$ as possible. Since the model is used to make predictions, the features used to obtain the augmentation function parameters will depend on $\widetilde{\boldsymbol{u}}_m$ and hence, the augmentation will be model-consistent. Apart from providing model consistency, solving an inverse problem also allows for the use of integral quantities and sparsely available measurements as data sources in the inference process. This makes the framework conducive to use experimentally obtained data or sparsely available data from computational simulations as well. In his thesis, Singh [63] noted that even using a single scalar quantity (the lift coefficient) the augmentation function was able to correct the flow field in order to obtain a more accurate distribution of the pressure coefficient on the surface of an airfoil. This is a testament to the robustness of the FIML approach as far as the use of sparsely available data (in physical domain) is concerned. It must also be noted that such success also depends on the accuracy of the chosen baseline model.

The following section details the different versions of the Field Inversion and Machine Learning framework.

2.2 Field Inversion and Machine Learning

The Field Inversion and Machine Learning (FIML) framework, originally proposed by Duraisamy and co-workers [81, 15, 82, 51, 64, 65, 66], was formulated to reduce inadequacies in a numerical model by inferring optimal augmentation functions from higher-fidelity data such that the predictive accuracy is improved. FIML provides a model-consistent framework that leverages solution methods for inverse problems and hence can make use of sparsely available data in the computational domain or, in some cases, even a single scalar quantity like the coefficient of lift for an airfoil [63]. The next iteration of the FIML approach was introduced by Holland et al [31, 32] and is referred to here as "Integrated Inference and Machine Learning (IIML)". This version of FIML addresses the concerns regarding the consistency and learnability of the inadequacy fields obtained from different training cases. The two main variants of this framework are discussed as follows.

2.2.1 Classic FIML

FIML seeks to obtain the optimal augmentation function parameters $\boldsymbol{w}$ by dividing the problem into two steps: (1) a "Field Inversion" problem is solved separately for each dataset to infer optimal inadequacy fields $\delta(\boldsymbol{x})$ in the respective discretized domains; which is followed by (2) a "Machine Learning" step that uses supervised learning to extract $\boldsymbol{w}$ from the augmentation fields (and respective features) obtained using the field inversion process for all spatial coordinates across all training datasets.

Field Inversion

$$\begin{aligned} \delta^{*i}(\boldsymbol{x}) &= \arg\min_{\delta^i(\boldsymbol{x})} \left\{ \mathcal{C}^i(\boldsymbol{y}^i, \boldsymbol{y}^i_m(\widetilde{\boldsymbol{u}}^i_m)) + \lambda^i_\delta \mathcal{T}^i_\delta(\delta^i(\boldsymbol{x})) \right\} \\ \text{s.t.} \quad & \mathscr{R}_m(\widetilde{\boldsymbol{u}}^i_m; \delta^i(\boldsymbol{x}), \boldsymbol{\xi}^i) = 0 \quad \forall \quad i = 1, 2, \ldots, N \end{aligned} \tag{2.2}$$

Equation 2.2 represents the field inversion problem which is solved individually over each of the N training cases (cases for which available high-fidelity data is to be used to infer the augmentation function parameters) to obtain optimal inadequacy fields $\delta^{*i}(\boldsymbol{x})$ in the respective domains, where i is the case index. The inadequacy field is optimal in the sense that it minimizes an objective function, which consists of a cost function, $\mathcal{C}^i$, and a regularization term, $\mathcal{T}^i_\delta$, with λ^i_δ as the regularization constant. Hence, it should be noted here that the optimal inadequacy field (and consequently the augmentation function) depends on the objective function being minimized, i.e., a different objective function might result in a completely different field for the same inadequacy term. The cost function, as described previously, quantifies the discrepancy between the available data and the corresponding model predictions for observables belonging to a given case. The objective function can be regularized to help with the ill-posedness of the field inversion problem. Multiple regularization terms can be used which can serve as weak constraints to numerically limit or impose a physical requirement on the resulting inadequacy field. One of the simplest combinations of cost function and regularization is to use an L_2 norm to estimate both these quantities as shown below.

$$\mathcal{C}^i = \|\boldsymbol{y}^i - \boldsymbol{y}_m^i(\widetilde{\boldsymbol{u}}_m^i)\|_2^2 \qquad \mathcal{T}^i = \|\beta^i(\boldsymbol{x}) - \beta_0\|_2^2 \tag{2.3}$$

where β_0 is the baseline value of the model, i.e., $\beta_0 = 0$ if β was added to some term in the model and $\beta_0 = 1$ if it was multiplied to some term in the model. Here, it is also worth noting that it is preferred to multiply the augmentation to some term in the model as then the augmentation is a dimensionless quantity which could, arguably, make generalization easier. Note here that an additive inadequacy term which is non-dimensionalized with the same turbulent length and time scales as the source term, can also be viewed as a multiplicative inadequacy term. This kind of a regularization would ensure that the augmentation field being predicted remains close to the baseline value.

Given the very high-dimensional nature of the augmentation fields that need to be inferred, the field inversion problem is usually solved using a gradient-based optimization technique where the gradients can be computed via techniques like finite differences, complex-step differentiation, discrete adjoints, etc. The details for the discrete adjoint approach to sensitivity evaluation can be found in Appendix B. In case the reduction in the objective function relative to its baseline value is not sufficient, the procedure can be repeated by starting from the last obtained augmentation field with a reduced step size, if needed. This does not imply that the objective function will be minimized to a value close to zero for every inference problem. This is because the inadequacy term being considered might not be capable to correct the QoI's to exactly their high-fidelity values.

Machine Learning

Once the field inversion problem is solved for all available datasets, the supervised learning problem can be solved using the augmentation fields and correspondingly calculated feature fields from all datasets to optimize for the function parameters $\boldsymbol{w}$ as

$\boldsymbol{w} = \arg\min_{\boldsymbol{w'}} \mathcal{L}(\boldsymbol{\delta}^*, \beta(\boldsymbol{\eta}^*; \boldsymbol{w'}))$ where, $\boldsymbol{\delta}^*$ refers to the stacked vector containing optimal augmentation fields across all datasets, and $\boldsymbol{\eta}^*$ contains the respectively stacked feature values. This can be written as a vector/matrix created by stacking vectors/matrices from individual cases one below another

$$\boldsymbol{\delta}^* = \begin{bmatrix} \delta^{*1}(\boldsymbol{x}) \\ \delta^{*2}(\boldsymbol{x}) \\ \vdots \\ \delta^{*N}(\boldsymbol{x}) \end{bmatrix} \qquad \boldsymbol{\eta}^* = \begin{bmatrix} \boldsymbol{\eta}(\widetilde{\boldsymbol{u}}_m^{*1}(\boldsymbol{x}), \boldsymbol{\zeta}^1(\boldsymbol{x})) \\ \boldsymbol{\eta}(\widetilde{\boldsymbol{u}}_m^{*2}(\boldsymbol{x}), \boldsymbol{\zeta}^2(\boldsymbol{x})) \\ \vdots \\ \boldsymbol{\eta}(\widetilde{\boldsymbol{u}}_m^{*N}(\boldsymbol{x}), \boldsymbol{\zeta}^N(\boldsymbol{x})) \end{bmatrix}$$

$\mathcal{L}(.,.)$ refers to some appropriate loss function that measures the discrepancy between the optimal augmentation field data obtained from field inversion and predictions made by the augmentation function under training with respective features as inputs. A simple loss function is the squared L_2 norm of the difference between the two, given as follows

$$\mathcal{L}(\boldsymbol{\delta}^*, \beta(\boldsymbol{\eta}^*; \boldsymbol{w}')) = \|\boldsymbol{\delta}^* - \beta(\boldsymbol{\eta}^*; \boldsymbol{w}')\|_2^2$$

If the functional form for the augmentation is chosen to be a neural network, a mini-batch gradient descent technique is well-suited to learn the function parameters in most cases when used with Adam or L-BFGS optimizer. Note that, while the inferred augmentation from the field inversion step is fully consistent with the underlying model, the augmentation field provided by the field inversion process is not necessarily learnable as a function of the chosen features [32]. This can lead to a loss of information extracted in the field inversion step. As discussed previously, the augmentation field obtained from field inversion is not unique, it is possible that the field inversion results obtained from two different datasets correspond to different features-to-augmentation mappings. This inconsistency can also degrade the learnability of the model. The more recent variant of FIML, referred to as integrated inference and machine learning addresses these concerns.

2.2.2 Integrated Inference and Machine Learning

Integrated Inference and Machine Learning (IIML) frameworks are versions of the classic FIML framework where, throughout the inference process, there must exist some set of parameters $\boldsymbol{w}$ such that $\delta^i(\boldsymbol{x}_j) = \beta(\boldsymbol{\eta}(\widetilde{\boldsymbol{u}}^i_{m,j}, \boldsymbol{\zeta}^i_j); \boldsymbol{w})$ (where j is the index for a spatial location in a discretized computation domain). Compared to the field inversion problem, the IIML framework constrains the optimization, either strongly or weakly, to proceed within a learnable manifold in the space of inadequacy fields ($\mathbb{R}^{N^i_x}$) and ensures that the optimization iterates lie strictly in this manifold. N^i_x refers to the number of discrete spatial locations within the computational domain for the i^{th} training case. While currently available techniques are predominantly strongly-coupled in nature (FIML-Direct by Holland et al. [31, 32] for RANS, DPM by Sirignano et al. [67], End-to-end differentiable learning by Xiao and coworkers), there do exist weakly-coupled strategies like FIML-Embedded by Holland et al. [31] as well. It should be noted here that while DPM and end-to-end differentiable learning frameworks share similarities with FIML, they were developed independently and contain variations that allow their application to problems which the FIML framework, in its original form, was unequipped to handle.

The FIML-embedded technique detailed by Holland et al. in [31] essentially couples the machine learning step with solver iterations while solving the forward model (as a part of the field inversion strategy). The augmentation function is embedded within the solver and it is the augmentation predictions that enter into the model equations. Using the sensitivities obtained from the previous inference iteration, a field inversion step produces a target inadequacy field that needs to be predicted at solver convergence by the augmentation function. Following every solver iteration, the augmentation function is trained using the current feature values from all spatial locations to minimize the discrepancy between the augmentation predictions and the corresponding values in the target inadequacy field. However, training an augmentation between solver iterations is problematic as the feature-to-augmentation map should not be inferred while the residuals are significantly high as it can introduce spurious behavior within the augmentation function. Even if the training begins after a significant number of iterations to make sure that the solver has converged, there is no guarantee that the residuals will continue to be small later. Other drawbacks (as listed in their paper) include potential convergence issues due to inclusion of learning iterations within solver iterations and a lack of any straightforward implementations to simultaneously infer the augmentation from multiple training cases. Chapter 5 details a novel weakly-coupled IIML strategy that resolves these issues and offers a robust alternative for cases where augmentations resulting from strongly-coupled IIML do not predict well despite good training accuracy due to the augmentation behavior being overwritten in parts of the feature-space during the inference process.

A brief description of a strongly-coupled IIML technique (specifically FIML-Direct [31, 32], although DPM and end-to-end differentiable learning are very similar) is provided along with its benefits and limitations as follows.

Strongly-coupled IIML

The strongly-coupled version of IIML completely bypasses the inference of an inadequacy field. Instead, the augmentation function is embedded within the model and a single inference problem is solved to directly optimize the augmentation function parameters $\boldsymbol{w}$. Mathematically, the coupled inverse problem can be posed as the following optimization problem which needs to be solved simultaneously over all datasets.

$$\begin{aligned} \boldsymbol{w} = \arg\min_{\boldsymbol{w}'} \bigsqcup_{i=1}^{N} \left(\mathcal{C}^i(\boldsymbol{y}^i, \boldsymbol{y}^i_m(\widetilde{\boldsymbol{u}}^i_m)) + \lambda^i_\delta \mathcal{T}^i_\delta(\beta(\boldsymbol{\eta}(\widetilde{\boldsymbol{u}}^i_m, \boldsymbol{\zeta}^i); \boldsymbol{w}')) \right) \\ \text{s.t. } \mathscr{R}_m(\widetilde{\boldsymbol{u}}^i_m; \beta(\boldsymbol{\eta}(\widetilde{\boldsymbol{u}}^i_m, \boldsymbol{\zeta}^i); \boldsymbol{w}'), \boldsymbol{\xi}^i) = 0 \quad \forall \quad i = 1, 2, \ldots, N, \end{aligned} \tag{2.4}$$

where $\bigsqcup$ is a generic assembly operator to build a composite objective function using the individual objective functions from all N training cases calculated for each dataset. The assembly operator can be as simple as a sum, if all datasets are equally important for the inference, or it could be a weighted sum, if some datasets are to be assigned more importance than others, or it could be something even more complex as designed/needed by the user. Notice here that using several cases at once adds an implicit regularization to the problem. Like the field inversion problem, gradient based optimization techniques are usually employed to solve this optimization problem. For simple functions, the derivatives of the augmentation function w.r.t. $\boldsymbol{\eta}$ and $\boldsymbol{w}$ are usually calculated analytically. For complex functions, algorithmic differentiation can be used. In the machine learning community, algorithmic differentiation of neural networks is termed backpropagation.

If discrete adjoints are used to obtain sensitivities required to solve the optimization problem, there exists an efficient and flexible way to embed the augmentation function within the solver which is described as follows. The forward simulation requires only the augmentation values at all the discrete spatial locations in the computational domain. When using discrete adjoints, the operators for sensitivities w.r.t. $\widetilde{\boldsymbol{u}}_m^i$ and $\boldsymbol{w}$ can be written as follows

$$\frac{d}{d\boldsymbol{w}} = \sum_{i=1}^{N} \sum_{j=1}^{N_x^i} \frac{d\beta_j^i}{d\boldsymbol{w}} \frac{d}{d\beta_j^i} \tag{2.5}$$

$$\frac{d}{d\widetilde{u}_{m,jk}^i} = \frac{\partial}{\partial \widetilde{u}_{m,jk}^i} + \sum_{\ell=1}^{n_\eta} \frac{\partial \eta_{j\ell}^i}{\partial \widetilde{u}_{m,jk}^i} \frac{\partial \beta_j^i}{\partial \eta_{j\ell}^i} \frac{\partial}{\partial \beta_j^i} \tag{2.6}$$

where N_x^i refers to the total number of discretized spatial locations in the computational domain of the i^{th} training case and n_η refers to the number of features used to characterize the augmentation. $\widetilde{u}_{m,jk}^i$ and $\eta_{j\ell}^i$ refer to the k^{th} state and ℓ^{th} feature, respectively, at the j^{th} spatial location for the i^{th} training case. As can be seen from Eqn 2.5, sensitivities w.r.t. $\boldsymbol{w}$ can be assembled from sensitivities w.r.t. the spatial fields $\beta^i(\boldsymbol{x})$ (β_j^i simply corresponds to the j^{th} entry in this field). Also, note here that $\beta^i(\boldsymbol{x})$ is nothing but the inadequacy field predicted by the augmentation function for the i^{th} training case. Now to evaluate sensitivities w.r.t. β_j^i using discrete adjoints, one would need to use the operator $\dfrac{d}{d\widetilde{u}_{m,jk}^i}$ which only requires the derivatives $\dfrac{\partial \beta_j^i}{\partial \eta_{j\ell}^i}$ and does not depend on the functional form of the augmentation. Hence, one only needs to pass a linearized version of the augmentation function into the solver, which makes the solver conveniently agnostic of how the augmentation is implemented and eliminates the need to change the solver code every time a change is made to the functional form of the augmentation. However, it must be noted that despite the convenience provided by such a linearized version, the augmentation needs to be evaluated for each solver iteration and hence the augmentation function needs to be called within the residual calculation part of the code. This still requires significant changes to the solver code which could require considerable effort if the solver makes use of custom datatypes and external libraries.

Benefits over classic FIML

- It has been observed that classic FIML can suffer from a loss of information during the machine learning step owing to the inability of the chosen functional form to accurately approximate the augmentation inferred during the field inversion step as a function of the specified features. Integrated inference and training bypasses this problem as the weights are directly updated and consequently, the obtained augmentation field remains consistent with the functional form of the augmentation. Thinking in terms of the augmentation field, this constrains the optimization to minimize the objective function to find an augmentation field which is realizable w.r.t. its functional form.

- Since integrated inference and learning can infer the function parameters $\boldsymbol{w}$ while simultaneously assimilating data from multiple datasets, the augmentation fields for all datasets are constrained to be realizable w.r.t. the functional form of the augmentation as explained in the previous point. This has an added advantage that this procedure, unlike classic FIML, is by design prevented from learning augmentation fields from different datasets that behave differently in the feature space. In other words, consistency in the features-to-augmentation mapping across datasets is automatically enforced when using integrated inference and learning.

- Building on the previous point, this means that if we have even a handful of DNS(Direct Numerical Simulation) fields from which true augmentation values can be extracted (or are readily available), they can be used to enforce a near-physical relationship between the features and augmentation by simultaneously using a plethora of other sparse field data from experiments or higher fidelity simulations. **This is critical from the viewpoint of generalizable and physical augmentations.**

Limitations

- Unlike the classic FIML approach, the task of designing an appropriate feature space must precede the inference from data. This, at times, could be difficult for the modeler and demonstrates the preliminary need of an independent field inversion step which can lend crucial information about the quantities correlated to the augmentation field that may then be used to formulate features.

- In its original form, integrated inference and learning does not consider the significance of localized learning. For practical problems, the data being used might not populate the entire feature space, which means that if the optimization problem is not constrained to change the augmentation only in the vicinity of the feature space locations for which data is available, it might lead to spurious predictions in other regions, which might not only result in worse accuracy compared to the baseline model but also severely affect the stability of the numerical solver.

- When using complex learning algorithms such as neural networks or decision trees, the augmented model inherits non-linearities from the baseline model **and** the learning algorithm. This, combined with the previous point, can lead to a disorderly optimization behavior when solving the inference problem, in addition to the aforementioned potential deterioration in accuracy and/or numerical stability with every successive optimization iteration.

Chapter 3

Learning and Inference Assisted by Feature-space Engineering

To alleviate the limitations in integrated inference and machine learning and make the augmentation generalizable, a set of guiding principles is presented to choose augmentation points and design appropriate features. The notion of localized learning in the feature-space is also introduced to reduce spurious behavior in the augmented model. Since the framework requires significant efforts in feature design and deciding how learning takes place in different feature-space regions, we call this version of integrated inference and learning as "Learning and Inference assisted by Feature-space Engineering (LIFE)".

This chapter is structured as follows. Section 3.1 describes and lists the hurdles when trying to infer robust, efficient and generalizable augmentation functions. Section 3.2 then describes a 1D model used to predict fully developed turbulent channel flow. 3.3 and 3.4 propose guiding principles to take into consideration when introducing the augmentation and designing the features, respectively, and demonstrate these principles by augmenting the model presented in section 3.2. Section 3.5 deals with the notion of localized learning and provides three different ideas on how to make it work. Finally, section 3.6 presents a hierarchical augmentation framework that could be used to infer augmentations with varying levels of generalizability/specificity.

3.1 Identifying Challenges

3.1.1 Generalizability

While the FIML approaches described in chapter 2 perform well for problems with similar geometries and flow conditions, the resulting augmented models, in many instances, can perform worse than the baseline models on several geometries and/or boundary conditions which are significantly different from those used during training. The augmented model may perform worse than the baseline model on even those canonical configurations which were used to calibrate the baseline model. Evidence of this behavior can be seen in the work by Rumsey et al. [60] where inferring an augmentation for an adverse pressure gradient case resulted in a deterioration in predictive accuracy for a zero pressure gradient case. Such losses in existing predictive accuracy within a model are undesirable as the user can never be sure of whether a prediction made by the augmented model is more accurate/trustworthy compared to that made by the baseline model. To alleviate this problem, Rumsey et al. [60] proposed including data from the canonical configurations to ensure that the augmentation does not alter the model behavior for these configurations. While this is effective, doing so would be expensive and difficult for cases exhibiting physical conditions which belong to parts in the feature-space where the baseline model performs well and hence, where the augmentation function should leave the model unaltered. Although it is seemingly impossible to infer an augmentation from limited data that is applicable with high accuracy on any general problem, it is important that the behavior of the augmented model can be controlled in scenarios in which the augmentation has not been trained.

During the inference/calibration of any model using data, the task at hand, in essence, is to fit a function such that the available datapoints are predicted with little (if not zero) error. The underlying assumption is that this function will closely approximate the true functional dependence between the respective quantities if there is "enough" data. The term "enough", here, means that the available data densely populates any part of the feature-space which can be accessed while solving any arbitrary configuration. Note here, that there might be parts of a feature-space which can never be realized physically and hence, there can never be any possibility of training/prediction outside this “realizable” part of the feature-space.

The data-driven modeling community, at times, misuses the terms “interpolation” and “extrapolation” while describing input parameters and boundary conditions used during model validation being within or outside the range of those used during model training, respectively. Instead, the true test of good predictions under extrapolation would be when the model performs well for feature values encountered during model testing which are outside the range of feature values encountered during model training. Mathematically speaking, in general, predictions based on extrapolation in the feature-space cannot be trusted. The story does not end here, though. For complex problems like RANS modeling, more often than not, one would not have a feature-space densely populated with datapoints. In such cases, even interpolated predictions could be significantly inaccurate, especially when the true inadequacy has a highly non-linear functional form.

Given these concerns, a model augmentation should be called generalizable only when the augmented model produces consistently better results compared to the baseline model across test cases with significantly different geometries and boundary conditions. Note that, while we do not make it a requirement for the augmented model to predict accurate results when comparing with data, this notion of generalizability would still result in models which are objectively better than the existing ones.

Finally, the issue of feature design needs to be addressed. If the chosen features are not descriptive enough, physical conditions which require significantly different augmentation values might share nearly identical feature values. This would require a very nonlinear augmentation which in turn would require a large amount of data to resolve the augmentation behavior within the feature-space. If the behavior cannot be resolved the augmentation function might not be inferred accurately enough, which is undesirable. Thus, the features should be chosen carefully such that physical conditions requiring different augmentation values remain in distinctly separate feature-space regions. Achieving this is not a trivial task and requires good physical understanding, expert knowledge and intuition.

3.1.2 Efficiency

Since the augmentation needs to be evaluated at every solver iteration for every discrete spatial location within the computational domain, it is important that the calculation of the augmentation function is fast and resource-efficient in order to maintain the benefits of using a reduced-fidelity model. To do so, it is important that the functional form is as mathematically simple as possible, and is implemented in the most efficient manner possible as a computer program.

While the physical quantities used to formulate the features should be chosen based on a combination of empirical evidence, physical understanding, and expert knowledge, the functional forms of features that these quantities are transformed into could drastically impact the functional form of the augmentation itself. To illustrate this point, consider the following example. Let a single feature be denoted by η and the true augmentation function be denoted by $\beta(\eta) \equiv c_2\eta^2$. Now, if a neural network, using conventional activation functions, needs to approximate this relationship, one or more hidden layers containing several nodes will be required. However, if the functional form of the feature is changed from η to η^2 the same function becomes linear, i.e., the augmentation function becomes much simpler and the overall computational cost is significantly reduced. While this is a very trivial example, changing the functional forms of the features might very well mean the difference between hard-to-learn expensive augmentations and easily-inferred inexpensive ones for a complex problem. Practically, this could even mean that a poor functional form might result in augmentation functions that are virtually impossible to learn.

In addition to the functional form of the features, the manner in which the augmentation term is introduced can also affect the functional form of the augmentation function. For instance, assume that there exists a term which we need to augment in the model of the form $(1 + f(\widetilde{\boldsymbol{u}}_m))g(\widetilde{\boldsymbol{u}}_m)$. This can be augmented in several ways, some of which are mentioned as follows.

- $\beta(\boldsymbol{\eta})(1 + f(\widetilde{\boldsymbol{u}}_m))g(\widetilde{\boldsymbol{u}}_m)$
- $(\beta(\boldsymbol{\eta}) + f(\widetilde{\boldsymbol{u}}_m))g(\widetilde{\boldsymbol{u}}_m)$
- $(1 + \beta(\boldsymbol{\eta})f(\widetilde{\boldsymbol{u}}_m))g(\widetilde{\boldsymbol{u}}_m)$

Each of these will result in a different augmentation function, and one of these might be simpler in structure compared to others in terms of some already chosen features.

3.1.3 Robustness

It is undesirable for an augmentation implemented into a numerical solver to adversely impact solver stability and residual convergence characteristics. To prevent this, care must be taken while introducing the augmentation into the model and also when learning its functional form. When an augmentation is introduced into a solver with implicit time-stepping, it changes the Jacobian (or its approximation) of residuals w.r.t. states. The Jacobian is required to calculate the updated states in a time-stepping iteration. Thus, if possible, the augmentation should not drastically affect the stability characteristics associated with this Jacobian. In the interest of solver stability, it must be ensured that the augmentation remains continuous and that it does not exhibit large-amplitude small-scale fluctuations in the realizable part of the feature space. This is because such a behavior of the augmentation function would increase the numerical stiffness of the augmented model, which would result in poor residual convergence or could even yield diverging residuals. Additionally, for higher order numerical schemes, the augmentation can affect quadrature requirements as well.

3.2 A Sample Problem for Illustration - Fully-developed Turbulent Channel Flow

3.2.1 Problem Description

Fluid flow between parallel flat plates placed a finite distance apart is termed as planar channel flow. The plates may or may not be in relative motion parallel to each other. When such a flow has traveled a significant distance, the flow velocities become invariant along the flow direction and the flow is referred to as a fully-developed channel flow. If the relative velocity between the plates is aligned with the direction of the pressure gradient, this results in the flow becoming essentially 1D in nature, i.e., the only direction in which flow quantities (except pressure) vary is the direction normal to the plates. Hence, in an incompressible setting, after neglecting the derivatives in the flow (x-) direction, the mass and momentum conservation can be written as follows under the Boussinesq approximation for steady-state Reynolds-averaged Navier-Stokes equations.

$$\begin{aligned}
\frac{\partial v}{\partial y} &= 0 \Rightarrow v = 0 \\
v\frac{\partial u}{\partial y} &= -\frac{\partial p}{\partial x} + \frac{\partial}{\partial y}\left((\nu + \nu_t)\frac{\partial u}{\partial y}\right) = 0 \text{ (as } v = 0)
\end{aligned}$$

Hence, given a pressure gradient, the following PDE describes the x-velocity profile in the y-direction.

$$\frac{\partial}{\partial y}\left((\nu + \nu_t)\frac{\partial u}{\partial y}\right) = \frac{\partial p}{\partial x} \tag{3.1}$$

At low speeds, the flow can be assumed to be incompressible and ν (i.e., the kinematic molecular viscosity) can be approximated only as a function of temperature. Further, if the temperature of the fluid throughout the domain is the same as the temperature of the flat plates, ν becomes a constant. ν_t (i.e., kinematic eddy viscosity), on the other hand, varies along the y-direction. There exist several models in the literature which can approximate the distribution of ν_t along y, based on the velocity distribution.

The Spalart-Allmaras (SA) Turbulence Model - a short description

The Spalart-Allmaras model [70] is mathematically expressed (without transition modification) by the following PDE for steady flows, which can be used to evaluate an estimate of ν_t.

$$u_i \frac{\partial \widetilde{\nu}}{\partial x_i} = c_{b1} \widehat{S} \widetilde{\nu} + \frac{1}{\sigma} \left[\frac{\partial}{\partial x_i} \left((\nu + \widetilde{\nu}) \frac{\partial \widetilde{\nu}}{\partial x_i} \right) + c_{b2} \frac{\partial \widetilde{\nu}}{\partial x_i} \frac{\partial \widetilde{\nu}}{\partial x_i} \right] - c_{w1} f_w \frac{\widetilde{\nu}^2}{d^2} \tag{3.2}$$

Here, $\widetilde{\nu}$ is the Spalart-Allmaras working variable and can be used to calculate eddy viscosity as follows.

$$\nu_t = f_{v1} \widetilde{\nu} \tag{3.3}$$

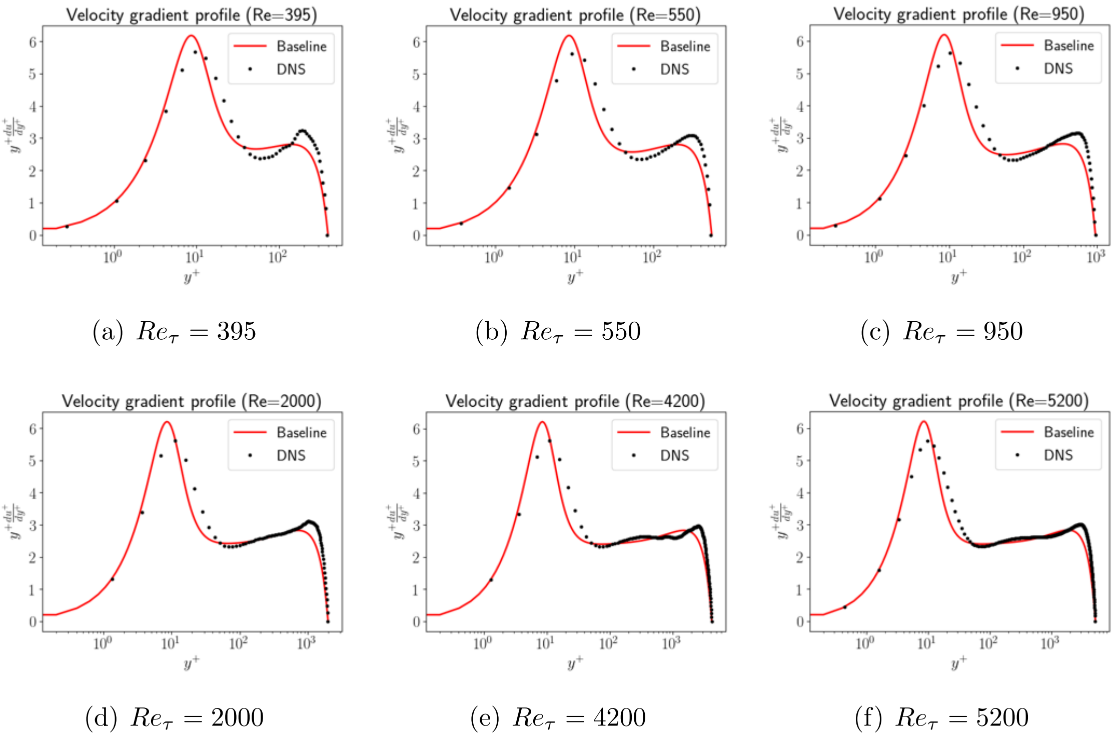
$$f_{v1} = \frac{\chi^3}{c_{v1}^3 + \chi^3} \tag{3.4}$$

$$\chi = \frac{\widetilde{\nu}}{\nu} \tag{3.5}$$

The full model description with the values and expressions for all parameters/functions can be found in [70]. The original calibration of the model by Spalart and Allmaras is briefly described as follows. The parameters c_{b1}, c_{b2} and σ were calibrated to predict the growth rate of a fully developed mixing layer and far wake with reasonable accuracy. Then, the parameter c_w was calculated using these parameters to ensure that the log-law is respected in a fully turbulent boundary layer. Following this a dampening function f_w was designed to predict the correct skin friction coefficient on a flat plate geometry under zero-pressure gradient at the location corresponding to $Re_x = 10^4$. Finally, the limiter f_{v1} was designed to ensure that the eddy viscosity diminishes to zero close to the wall. This knowledge of how the baseline model works will be relevant when discussing model augmentation in the following sections.

3.2.2 Discrepancy under consideration

Looking at the results from the baseline simulation of a 1D Poiseuille flow (Channel flow with plates stationary w.r.t. each other under a specified pressure gradient) in figures, 3.1 and 3.2, there exists a significant discrepancy between the results obtained using the SA model and those obtained via direct numerical simulation (DNS) in the buffer



(a) $Re_\tau = 395$

(b) $Re_\tau = 550$

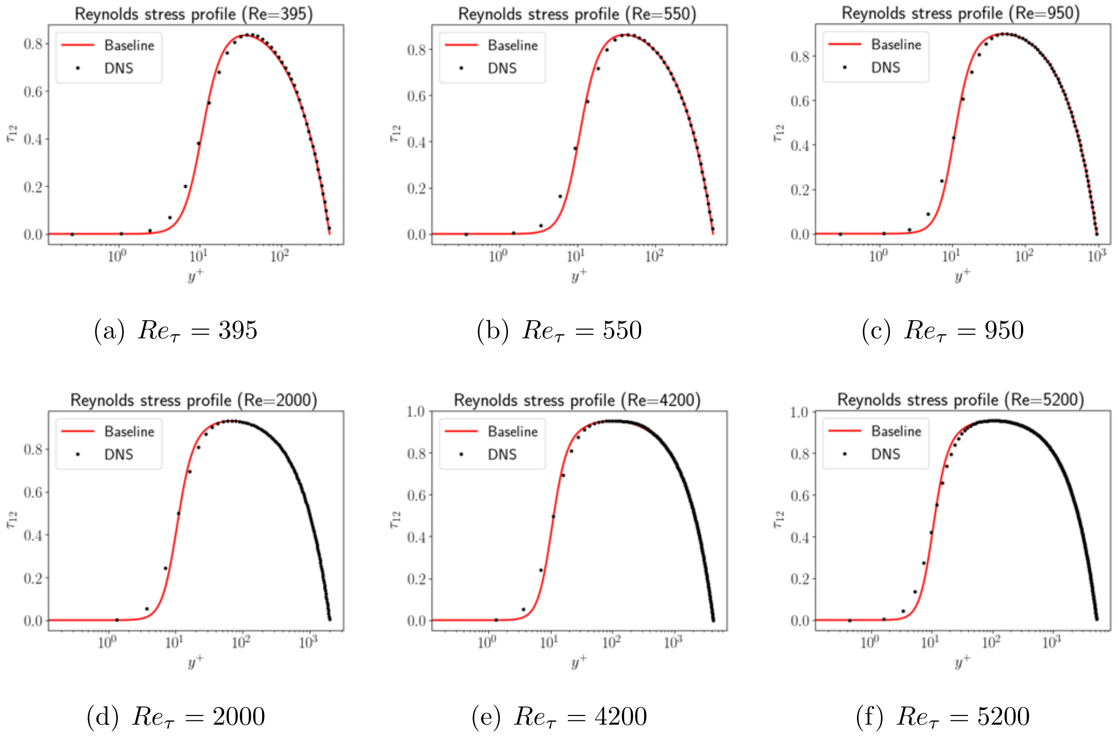
(c) $Re_\tau = 950$

(d) $Re_\tau = 2000$

(e) $Re_\tau = 4200$

(f) $Re_\tau = 5200$

Figure 3.1: Velocity gradient data vs. Spalart-Allmaras predictions for 1D Poiseuille flow



(a) $Re_\tau = 395$

(b) $Re_\tau = 550$

(c) $Re_\tau = 950$

(d) $Re_\tau = 2000$

(e) $Re_\tau = 4200$

(f) $Re_\tau = 5200$

Figure 3.2: Reynolds stress data vs. Spalart-Allmaras predictions for 1D Poiseuille flow

layer region (around the peaks in figure, 3.1). Although there exist slight discrepancies in the outer layer too, we shall target the model inadequacy responsible for only the discrepancy in the buffer layer region. In the following sections, we methodically lay a set of guiding principles which can help to introduce generalizable, efficient and robust model augmentations.

3.3 Guiding Principles for Introducing the Inadequacy Term

The following aspects could be taken into consideration while deciding how to introduce an augmentation in the baseline model:

3.3.1 Generalizability - reducing spurious behavior

The augmentation should be introduced such that it does not corrupt model behavior in regions where the inadequacy under consideration is not a concern. This is important for two reasons. First of all and most importantly, it restricts the augmentation from deteriorating the existing baseline model accuracy for regions in the feature space that have not been encountered during training. Secondly, it gives more freedom during feature design because the augmentation acts in isolated regions of the flow by design, and hence it does not have to depend on the features to figure this out. Obviously, this might not be possible for all cases, but modelers should always be on the lookout for such augmentation locations.

For the 1D channel flow example, it is undesirable to perturb the calibrations for free shear flows and hence directly augmenting the production or diffusion terms should be avoided. As the c_w coefficient ensures the log-law behavior, augmenting it would be unwise. Although there is debate over whether the log-law is indeed ubiquitous and sacred, that is not the discrepancy we are targeting here. Since the f_w function was designed to control the eddy viscosity in the outer layer, it would not serve as a good augmentation point either. However, augmenting the f_{v1} function could do the trick. By design, $f_{v1} = 1$ outside the inner layer, and hence, any augmentation here will result in changes only within the inner layer, thus isolating the region of interest. The main parameter that dictates how $\widetilde{\nu}$ gets dampened into ν_t by f_{v1} is χ, and hence χ was chosen to be augmented.

3.3.2 Efficiency

The augmentation should be introduced such that its functional form would be as simple as possible. While it is impossible, in most cases, to determine the complexity of the augmentation function beforehand, a good understanding of the model can help choose the manner in which the augmentation is introduced within a model such that the augmentation function becomes simpler. Note that, in the context of simplicity of the augmentation function, the manner in which the augmentation is introduced might affect the features that should be used and vice-versa.

For the 1D channel flow example, this is relatively simple. An additive, multiplicative or divisive augmentation can be chosen as $\chi + \beta$, $\beta\chi$ or χ/β, respectively. This is because of the following reasons. Firstly, the features for this augmentation would have to be functions of $\widetilde{\nu}$ and ν as there is no other physical/modeled quantity the augmentation needs to depend on as only the functional behavior of f_{v1} needs to be corrected. Also, observing the data and baseline predictions especially in Fig. 3.2, the discrepancy would not require χ to be changed beyond its baseline value by any extravagant amount and this change would be seemingly smooth w.r.t. χ, as will be demonstrated later by inference results. Note that these are assumptions made before inferring the augmentation. If the obtained augmentation violates any of these assumptions, it indicates that there might exist a more efficient way to augment the model. In this particular case, we choose a multiplicative augmentation as $\beta\chi$.

3.3.3 Robustness

As has been discussed before, robustness requires that the augmentation does not have large-amplitude small-scale fluctuations in the feature space and that the stability characteristics of the Jacobian are not deteriorated.

For the 1D channel flow case, the Jacobian is slightly modified only in the regions with small $\widetilde{\nu}$ and that too by a small amount as the augmentation β need not change χ by a huge amount. Since the behavior of the features, the augmented term and the discrepancy in observables is relatively smooth and nicely behaved, it is safe to assume that the augmentation function should be smooth and free of fluctuations as well. If this assumption is found false however, corrective measures need to be taken in order to maintain good convergence characteristics of the numerical solver.

3.4 Guiding principles for Feature Design

To determine the features that the augmentation function would depend on, one should use the following guiding principles.

3.4.1 Choice of features based on expert knowledge

The choice of features has a major role to play in how effective the augmentation function can be at addressing the model inadequacy in consideration. While automated feature selection techniques exist in literature, they heavily (if not prohibitively) depend on data to find the best features from amongst hundreds/thousands of possible candidate functions of local quantities for a complex problem like transition or turbulence. The author, hence, believes that human intuition and expert knowledge can lead the way in feature selection as it has proven effective for traditional modeling. This is one of the major reasons why the LIFE framework is aimed towards use by expert modelers. While choosing features, it is desired that there exists a causal relationship between the chosen features and the inadequacy targeted by the augmentation function. However, for steady-state models, the augmentation in turn influences the features by virtue of feedback, and hence quantities which do not share a causal relationship with the inadequacy and, rather are only correlated with it, can also serve as features.

For 1D channel flow, we assume that simply correcting the functional form of f_{v1} can achieve the desired result. Hence, the quantities chosen to design features are just $\widetilde{\nu}$ and ν. Since this is a steady-state model and since ν_t is a quantity derived from $\widetilde{\nu}$, ν_t can also be used as a flow quantity in feature design. In case the aforementioned assumption is false, more flow quantities would need to be brought in to design features.

3.4.2 Physics-based non-dimensionalization

Once the quantities to be used as features are chosen, they must be non-dimensionalized in order to be generalizably used for prediction. This is because similar physical phenomena can occur for significantly different magnitudes of dimensional model quantities and, since the LIFE framework aims at using as little data as possible to discover generalizable augmentation functions, proper non-dimensionalization of the features becomes imperative. While statistics from the training datasets can be used to non-dimensionalize features, the available data might not be sufficiently representative of the complete range of values that could be encountered during prediction on an unseen case. Thus, it is better to make use of physical/model quantities to non-dimensionalize features.

For 1D channel flow, taking inspiration from the existing non-dimensionalized term χ, we can use $\widetilde{\nu}/\nu$, $\widetilde{\nu}/\nu_t$ or ν_t/ν as a feature. All of these are dimensionless quantities, and given a specific augmentation function, are indicative of different locations in the inner layer.

We shall see more creative non-dimensionalization strategies later in this chapter.

3.4.3 Effectively-bounded Feature-space

While non-dimensionalization is critical, it should be noted that a non-dimensionalized feature can still be unbounded (e.g., ν_t/ν). Since the data used to learn the augmentation is limited, this can lead to extrapolated predictions by the augmentation function for features outside the range of values available from the training data. To circumvent this, the functional form of the non-dimensionalized feature must be chosen such that either the feature is mathematically bounded or it takes non-baseline values (baseline value is 0 for an additive augmentation and 1 for a multiplicative augmentation) only in a bounded part of the feature space. We shall call such a feature as "effectively" bounded, hereafter. Effective boundedness minimizes extrapolation errors and makes the augmentation more robust and generalizable.

For 1D channel flow, a mathematically bounded feature-space can be created. For instance, using the non-dimensionalized quantity ν_t/ν, a bounded feature can be designed as $\nu/(\nu_t + \nu)$. This change in functional form or limiting of the feature must be done mindfully in order to ensure that the augmentation function does not vary too much in a small part of the feature space as this might lead to a sub-optimally fitted augmentation in addition to robustness issues. For 1D channel flow, the aforementioned bounded functional form works because we are targeting regions characterized by values of ν_t not more than a few times that of ν.

3.4.4 Appropriate functional form for features

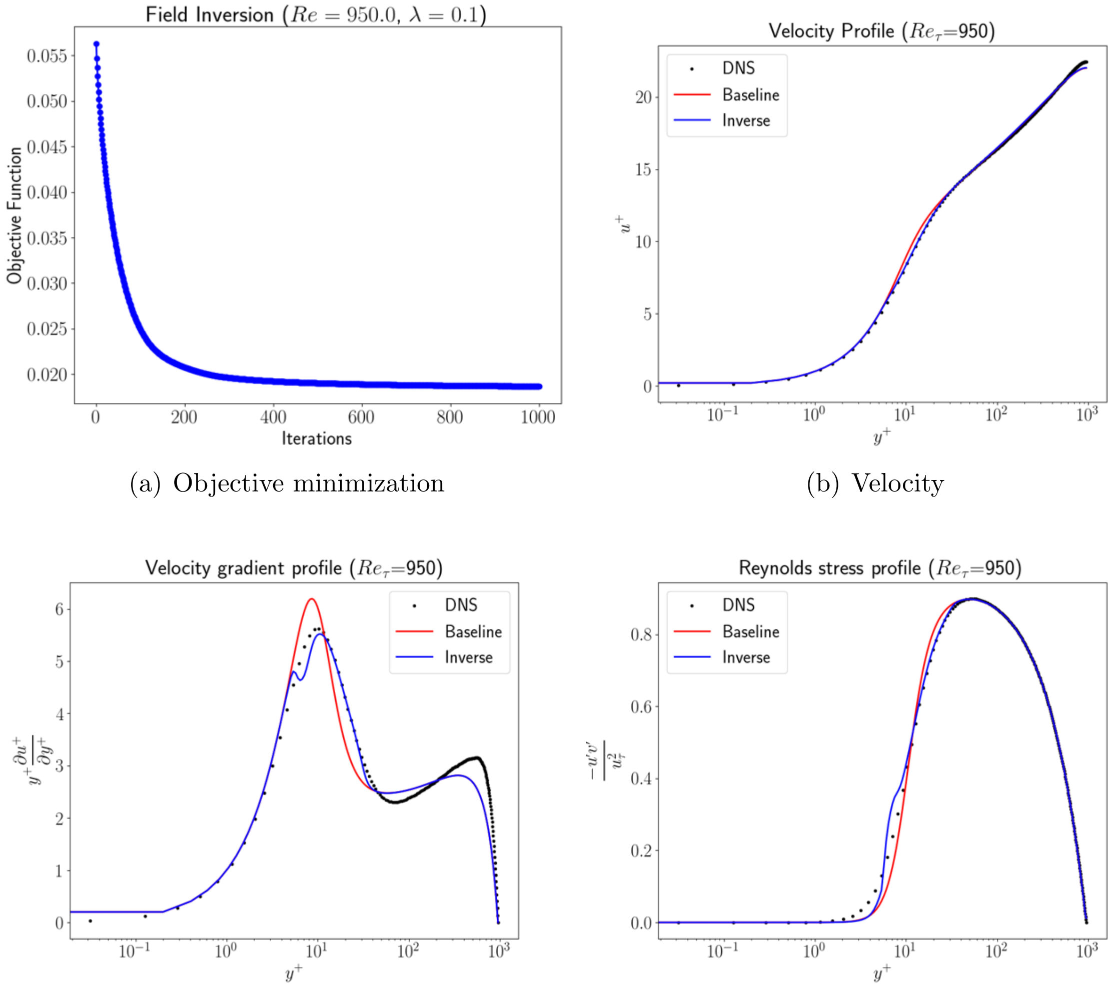
Several functional forms of a non-dimensionalized feature can offer effective boundedness. But, only a few of these forms might address the inadequacy in a major part of the feature space. For instance, the part of the feature space covered by the feature $\nu/(\nu_t+\nu)$ between the values of 0 and 0.3 reduces to being covered between the values 0 and 0.000729 for a different form of the same feature, viz., $(\nu/(\nu_t + \nu)^6)$. Note that, both features have the same mathematical range between 0 and 1. However, by the virtue of different functional forms, the regions denoting the same physical conditions span very differently in the feature space. For an augmentation which is predominantly affected within the said range, the feature $\nu/(\nu_t + \nu)$ offers a better-conditioned learning problem. Hence, the choice of which among different effectively bounded non-dimensional functional forms of a feature to use can play a significant role in setting up a well-conditioned learning problem.

For 1D channel flow, the feature $\nu/(\nu_t + \nu)$ was found adequate enough to improve the model predictions.

3.4.5 Parsimonious combination of features vs. one-to-one mapping

Finally, the augmentation should be a function of as small a number of features as possible to maintain simplicity, as a simpler augmentation could mean better generalizability when training on limited data. On the other hand, the features should be chosen such that there exists a one-to-one feature-to-augmentation mapping. While a perfect one-to-one mapping might be exceedingly difficult to achieve in the entire feature space, the property should be virtually preserved (i.e., if more than one *optimal* augmentation values correspond to a single location in the feature space, all of them should be sufficiently close to each other) in as large a region of the effectively bounded feature space as possible. More features imply a better chance of ensuring such a mapping, as more physical conditions can be uniquely attributed to a location in the feature space. Note that "mapping" here refers to the true relation between features and corresponding optimal augmentation values for all locations in the feature space. It does **not** refer to the augmentation function (which is one-to-one by definition and tries to approximate this mapping). While such a mapping is difficult to comprehend apriori, solving a field inversion problem can help in getting an approximate idea of how the mapping should look. If the mapping is not one-to-one, then during the inference and learning process, sensitivities from two different datapoints might try to pull the augmentation value at some location in the feature space in opposite directions and the so-obtained optimal augmentation function would predict a compromise between two significantly different values which are optimal w.r.t. each of the two datapoints. Physically, this translates to the feature space being inadequate to uniquely represent distinct physical phenomena corresponding to significantly different augmentation values. If the features are insufficient, the mapping will not be one-to-one and the training accuracy will suffer. On the other hand, if there is are more features than required, the available data would be sparser implying more interpolation error and the predictive accuracy on the test cases will suffer. Thus, a balance has to be attained by choosing just enough features to ensure a virtually one-to-one mapping in most of the effectively bounded feature space.

Since we have hypothesized that a single feature $\nu/(\nu_t+\nu)$ is sufficient for the 1D channel flow problem, the guidelines in this subsection do not apply here. But, we can still solve a field inversion problem and check the behavior of the augmentation function w.r.t. the feature. The following sum-squared discrepancy in velocity was used as the objective



(a) Objective minimization

(b) Velocity

(c) Velocity Gradient

(d) Reynolds stress

Figure 3.3: Field Inversion for 1D Poiseuille flow ($Re_\tau = 950$)

function to solve the field inversion problem.

$$\mathcal{J} = \|u_{\text{pred}} - u_{\text{DNS}}\|_2^2 + \lambda\|\delta - 1\|_2^2 \tag{3.6}$$

Steepest gradient descent was used to solve the optimization problem with a step size of $\dfrac{0.1}{\left\|\dfrac{\partial \mathcal{J}}{\partial \delta(\boldsymbol{x})}\right\|_\infty}$. The convergence of the objective function and the fields corresponding to the resulting inverse solution for the case $Re_\tau = 950$ are shown in Fig. 3.3. The feature-to-augmentation relationships extracted from such field inversion problems for all the available cases are shown in Fig. 3.4. From this figure, it can be observed that the feature-to-augmentation relationships for all Reynolds numbers seem to coincide,

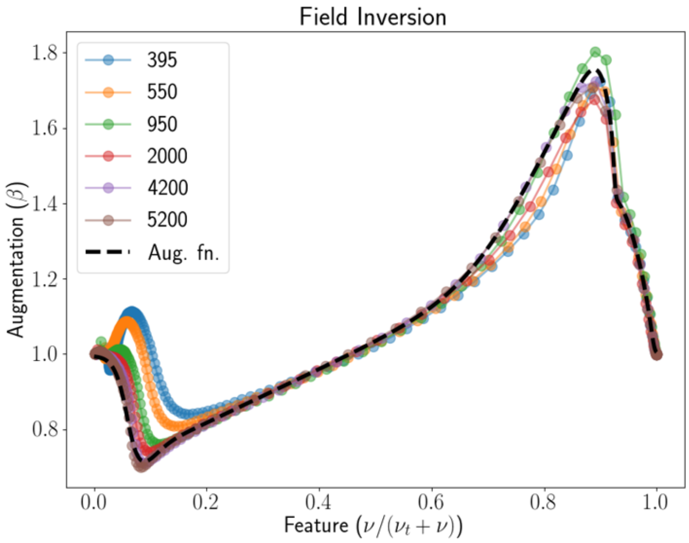


Figure 3.4: Feature-to-Augmentation relationships

especially for higher Reynolds numbers. For lower Reynolds numbers, there seems to be discrepancy for lower values of $\nu/(\nu_t + \nu)$ but that corresponds to regions of very high ν_t/ν and hence would affect regions either in the outer layer or the outer part of the buffer layer. This lack of a one-to-one relationship in a small part of the feature-space which would not affect the predictive accuracy by a large amount can be accepted. An augmentation function can then be hand-fitted to approximately *trace* these relationships as shown by the black dashed line. This hand-fitted augmentation function is given as follows.

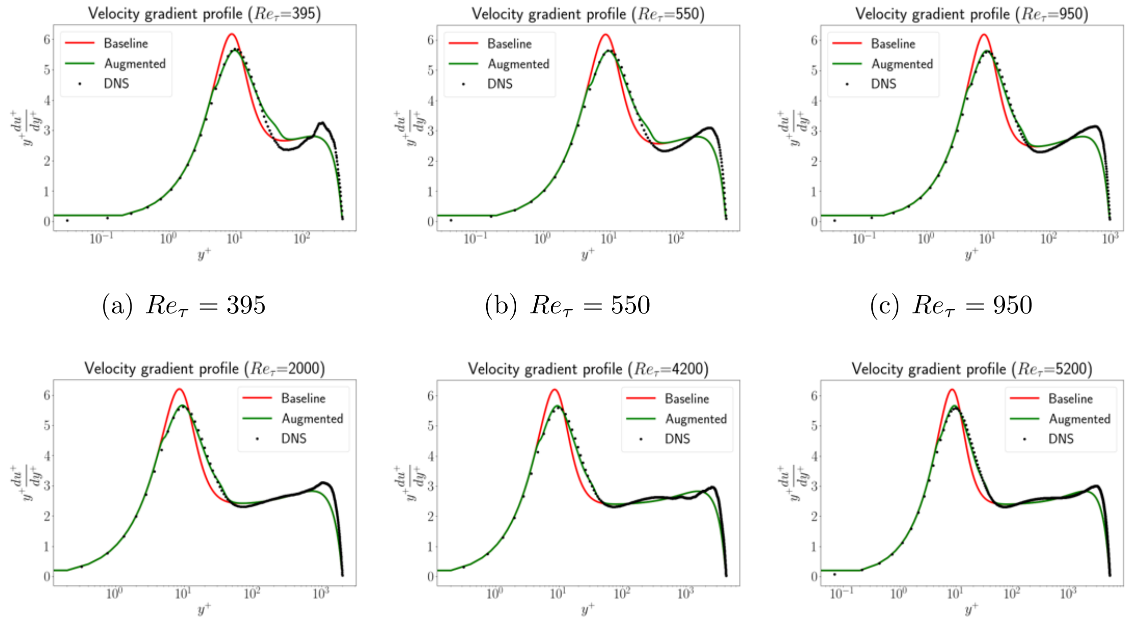
$$\beta(\eta) = \mathrm{sgm}(\phi_2(\eta), \phi_4(\eta), 555 - 600\eta)$$

$$\phi_4(\eta) = \mathrm{sgm}(\phi_3(\eta), 1.01, 298.8 - 300\eta)$$

$$\phi_3(\eta) = 1.42 - 60(\eta - 0.915)^2$$

$$\phi_2(\eta) = \mathrm{sgm}(\phi_1(\eta), 1, 65.1 - 70\eta)$$

$$\begin{aligned}\phi_1(\eta) = &\ \frac{1}{7.8}\log(\exp(7.8(3.1\eta - 0.96)) + \exp(7.8(0.71\eta + 0.675))) \\ - &\ 0.04\exp(-600(0.078 - \eta)^2) + \mathrm{sgm}(0, 0.32 - 0.7\eta, 100\eta - 6)\end{aligned}$$



(a) $Re_\tau = 395$ (b) $Re_\tau = 550$ (c) $Re_\tau = 950$

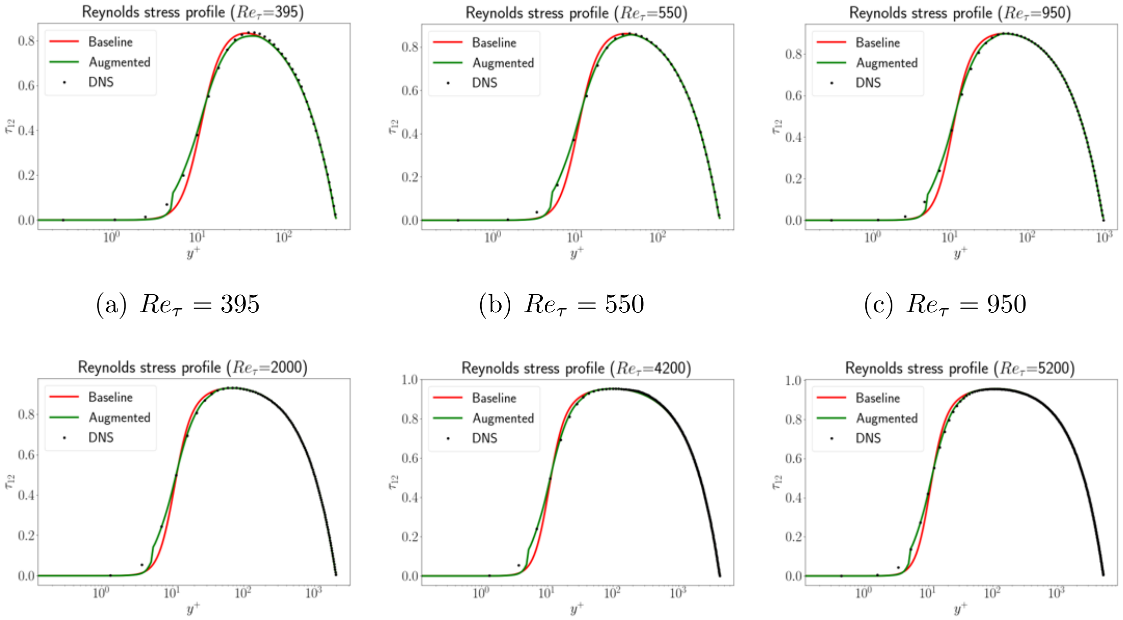
(d) $Re_\tau = 2000$ (e) $Re_\tau = 4200$ (f) $Re_\tau = 5200$

Figure 3.5: Velocity gradient data vs. augmented Spalart-Allmaras predictions for 1D Poiseuille flow

where the feature $\eta = \nu/(\nu_t + \nu)$ and the modified sigmoid function (sgm) is defined as

$$\mathrm{sgm}(x, y, z) = x + \frac{y - x}{1 + \exp(z)}$$

Using this augmentation function, the results for all cases of Poiseuille flow are shown in Fig. 3.5 and 3.6. Consistent and significant improvements across all cases of Poiseuille flow are seen using the hand-fitted augmentation function. A kink around $y^+ = 5$ can be observed for all predictions and is a result of the hand-fitted augmentation function not accurately matching the behavior of the inadequacy fields obtained via field inversion for regions with low values of $\nu/(\nu_t + \nu)$. When applied to a case of a Couette flow (channel flow with plates moving relative to each other and zero pressure gradient), similar improvements were noted as seen in Fig. 3.7. Thus, we have, with a very simple example, demonstrated that by adopting the above guiding principles and practices, generalizability of a model can be improved. However, it should be noted that designing augmentations and features is a complicated and intricate process for complex problems. Also, while field inversion provides with inadequacy fields which share a consistent re-



(a) $Re_\tau = 395$

(b) $Re_\tau = 550$

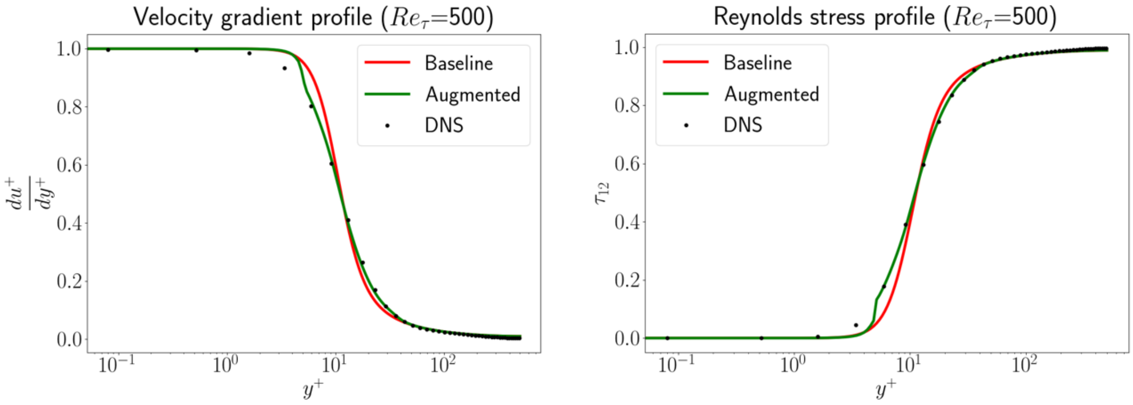
(c) $Re_\tau = 950$

(d) $Re_\tau = 2000$

(e) $Re_\tau = 4200$

(f) $Re_\tau = 5200$

Figure 3.6: Reynolds stress data vs. augmented Spalart-Allmaras predictions for 1D Poiseuille flow



(a) Velocity gradient

(b) Reynolds Stress

Figure 3.7: Data vs augmented Spalart-Allmaras predictions for a Couette flow

lationship across all training cases for this particular problem and provides predictions that are consistently more accurate using these inadequacy fields, it is fraught with limitations and inconsistencies as described in chapter 2. Thus, for more complex problems integrated inference and learning is recommended.

3.5 Localized Learning

Notice that in the 1D channel flow case shown above, the training data from different spatial locations in the discretized domain populate the entire feature-space (as seen in Fig. 3.4) for each training case, hence, characterizing the behavior of the augmentation function in the entire feature-space. Hereafter, we shall refer to such individual points in the feature-space, the coordinates for which are given by the feature values calculated at corresponding locations in the discretized physical domain of a training case, as a datapoint. For complex problems, the available data might not sufficiently populate the feature space to characterize the behavior of the augmentation function. A lack of datapoints in certain parts of the feature-space could lead to large errors during interpolation/extrapolation as has already been discussed in section 3.1, thus resulting in spurious predictions and deteriorating the performance of the augmented model even below that of the baseline model. Additionally, for steady-state models like RANS, the feature values obtained in one solver iteration influence the feature values obtained in the next iteration. The inference and learning process, however, works only with the feature values obtained at solver convergence. This means that the solver might be accessing non-populated parts of the feature-space when the residuals are not converged, and hence the augmentation values in these regions can affect the converged result. While there is no way to ascertain true augmentation behavior in the absence of data for a region in the feature-space, to make the augmented model always perform at least as well as the baseline model, these regions must be constrained to predict baseline values. This, in essence, implies that learning should take place only close to the available datapoints in the feature-space. This statement leads to two major concerns – (1) How can such a "*localized*" learning be ensured; and (2) How large of a vicinity should a datapoint affect in the feature-space? These questions are discussed in the following subsections.

3.5.1 How to perform localized learning?

While there may exist other ways of achieving this, we propose the following three methods to perform localized learning.

Neural Networks with artificial training datapoints

In their original form, conventional learning techniques relying on function architectures like neural networks, decision trees etc. are focused at learning a global representation of a single underlying function. Note here that this is not the case for kernel regression, but its implementation would make the computation of the augmentations slow and expensive, especially when the training dataset has a large number of datapoints. In order to perform localized learning with neural networks, significant numbers of artificial datapoints must be used to enforce the baseline behaviour in regions of the feature-space which are not populated by inferred datapoints. This would, firstly, require identification of regions in the feature-space outside the vicinity of any datapoints, and subsequently artificial datapoints would have to be sampled from such regions and used along with inferred data to update the function parameters (weights and biases). This is similar to what what done in [60], except that the artificial datapoints, here, are not sampled from any canonical case but are added for all regions that have few or no datapoints. A simple example of how artificial datapoints can be added for classic FIML and strongly-coupled IIML is illustrated in Fig. 3.8. In this particular example, a grid of "candidates" for artificial datapoints is constructed within the feature-space. Then, radial basis functions centered at the available datapoints are used to define the neighborhood that will be influenced by each datapoint. Note that this neighborhood can vary in the feature-space. All candidates for artificial datapoints that fall within this neighborhood are discarded and the remaining are accepted. The augmentation values for all the accepted artificial datapoints are, then, either set to the baseline value in the case of classic FIML, or calculated based on the previous iterate of the augmentation function parameters for

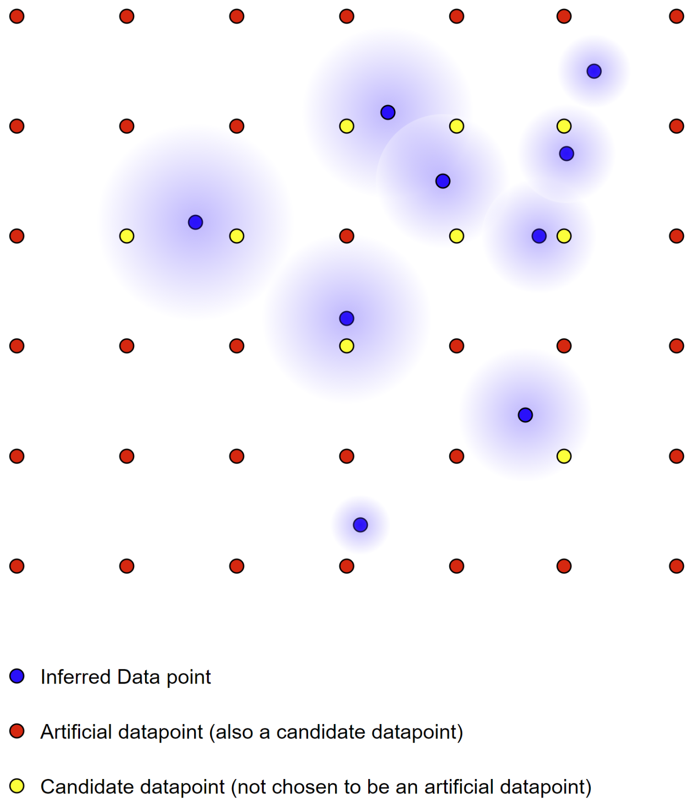

Inferred Data point
Artificial datapoint (also a candidate datapoint)
Candidate datapoint (not chosen to be an artificial datapoint)

Figure 3.8: Example of localized Learning by introducing artificial datapoints

strongly-coupled IIML. The parameters of the augmentation function are then updated using both, artificial datapoints and available data. Note that, for any generic point within the feature-space which is outside the neighborhood of any available datapoint, the updated augmentation may not exactly predict baseline values. However, the presence of artificial datapoints in the update process acts as a regularizer to ensure that the predictions are close to the baseline value. Localized learning via neural networks can be difficult for the following reasons. The neural network architecture must have enough complexity to represent local behavior of the augmentation function in several distinct regions of the feature-space. This could require a large number of hidden layers and/or nodes per hidden layer. Even if the architecture is complex enough to facilitate an intricate augmentation function, initialization of weights and biases within a neural network can affect learning and sometimes can prevent the neural network from representing the augmentation behavior accurately in all feature-space regions. More efficient methods to achieve localized learning with a neural network (including better initialization strategies) are, hence, a subject for further research.

Interpolation on a discretized feature-space

Conventional learning architectures can also be avoided altogether, and alternative architectures, which are more conducive for localized learning, can be introduced. One of the crudest of these architectures is discretizing the feature-space and parametrizing the augmentation function using augmentation values at the nodes/centroids of the resulting grid elements. Note that the local grid spacing, here, characterizes the vicinity of a datapoint in the feature-space. This offers a fast and efficient way of calculating the augmentation value at any point in the feature-space by the virtue of interpolation and can isolate learning for different regions in the feature-space. A downside of this approach is that it suffers from the curse of dimensionality and cannot handle a large number of features with sufficiently fine discretizations. It should be noted here that this method of localized learning does not employ neural networks or decision trees and is the method demonstrated in Chapter 4 of this work. Two implementations of this approach shall be discussed in greater detail in Chapter 4.

Neural networks on a discretized feature-space

As mentioned above, using a neural network in the entire feature-space to approximate a complex augmentation function might require a large number of parameters. One way of possibly reducing the number of parameters is by discretizing the feature-space and then learning individual neural networks for each grid element. This would reduce the number of parameters used by a single neural network and thus, make the calculation of augmentation values cheaper and faster. This might also facilitate easier localized learning using neural networks as they would be exposed to only a part of the feature-space where the augmentation mapping will be less intricate compared to that in the entire feature-space. In addition, the ability of neural networks to approximate complex functional forms (and hence offer complex interpolations within grid elements in the feature-space) implies that discretizations which are coarser than those required when using a discretized feature-space without neural networks can also perform well. This would lead to the curse of dimensionality becoming weaker in such a hybrid implementation.

3.5.2 How large of a vicinity should a datapoint affect in the feature-space?

The region in the feature-space in the immediate vicinity of a datapoint $\boldsymbol{\eta}_d$ that it modifies during localized learning will, hereafter, be referred to as the datapoint's **range of influence** which can be characterized by an **influence function**, $\mathcal{I}(\boldsymbol{\eta}; \boldsymbol{\eta}_d)$. Also, the influence function might not be isotropic in the feature-space. In other words, the range of influence for a point in the feature-space could be larger along one feature direction compared to another.

The range of influence is an important quantity when balancing training accuracy against generalizability. If the range of influence is too small, the training accuracy will be high but the augmentation would have poor generalization as most of the feature-space would still assume baseline values. On the other hand, if the range of influence is too large, a datapoint could affect far-off regions in the feature-space and hence the training accuracy might suffer. This trade-off is very similar to the over-fitting vs generalization trade-off encountered during training of a neural network.

For an augmentation formulated in a discretized feature-space such that augmentation values are predicted using an interpolation function for every feature-space grid cell, the range of influence can be directly controlled by controlling the grid spacing and there is no need for an explicit influence function. On the other hand, if the functional form of an augmentation is based on neural networks, the use of an influence function becomes important. The parameters of the influence function can be hypothesized as functions of the features themselves. While finding *the* optimal influence function might not be feasible, approximations can be made for the same, based on heuristics. For instance, a simple approximation for an influence function can be written as follows.

$$\mathcal{I}(\boldsymbol{\eta}; \boldsymbol{\eta}_d) = \exp\left(-(\boldsymbol{\eta} - \boldsymbol{\eta}_d)^T \boldsymbol{C}(\boldsymbol{\eta}_d)(\boldsymbol{\eta} - \boldsymbol{\eta}_d)\right) \tag{3.7}$$

Here, $\boldsymbol{C}(\boldsymbol{\eta}_d)$ is a covariance matrix which is a function of the feature values at the datapoint. These components can themselves be approximated as simple functions (linear, quadratic, etc.), and several inference problems can be solved by varying the parameters of these component functions to find the ones that provide the best training and validation accuracy w.r.t. available high-fidelity datasets. Note here that validation accuracy can be considered as a proxy to generalizability if the validation datasets are significantly different from training datasets in geometry and physical conditions. Clearly, there exists ample scope to explore different localized learning techniques along with strategies to optimally choose influence functions. **However, the main objective of this work w.r.t. localized learning is to introduce it in the LIFE framework and emphasize its importance in creating robust and generalizable augmentations, and the explorations of such sophisticated localized learning techniques and strategies is left as a subject for future work**.

3.6 Hierarchical Augmentations

There may be situations in real-world applications where a single augmentation cannot address all sources of model-form inadequacies within the model. There are two ways to deal with such issues: (1) Use a larger set of features to better characterize the augmentation; or (2) Use multiple augmentations with smaller numbers of features to address specific parts of the discrepancy. In the author's opinion, it is almost always better to take the second route, since a larger set of features could lead to a loss of generalizability as more features could over-specify physical conditions, and also because it provides the modeler with a chance to introduce multiple augmentation terms within the model. The functional form of each of these multiple augmentations could be much simpler compared to that of a single augmentation trying to resolve all sources of inadequacy. Now, these multiple augmentations could be inferred all at once, or one-by-one. While inferring all augmentation functions at once might be possible, it might take a large number of inference iterations to do so. Again, in the author's opinion, it is better to deal with augmentations one-by-one beginning with the one that is the most generic (i.e., will affect the widest range of problems) and making way to the most specific augmentation (i.e., one which affects only a particular class of problems). To do so, an augmentation is introduced in a manner that does not affect problems other than the class of problems it is intended for. Once inferred, the resulting augmented model can serve as the baseline model for the next level of augmentation and the process can be repeated. The augmentations built using this methodology are termed here as hierarchical augmentations. While the biggest challenge in designing hierarchical augmentations is ensuring that the effect of each subsequent augmentation is restricted to the problems it is intended for, it has two major practical benefits. Firstly, one can obtain several models in decreasing level of generalizability (i.e., increasing level of specificity to a class of problems). Hence, to adapt for a new class of problems, an existing level of the hierarchically augmented model with the appropriate level of generalizability can be picked off the shelf and tweaked as required. Secondly, it removes the necessity of re-using a dataset (which has been used for any preceding levels of augmentation) to constrain the learning process in order to ensure that model does not lose accuracy on such datasets, thus cutting down on training time. A hierarchical augmentation is presented in Section 4.3 to improve predictive accuracy for flows involving separation-induced transition.

Chapter 4

Application of LIFE - Bypass Transition Modeling

4.1 Introduction

Accurately predicting laminar-to-turbulent transition is crucial for reduced-fidelity fluid flow models as it can improve the preliminary design process of different flow surfaces such as aircraft wings, wind turbine blades, automobiles, gas turbine components (compressor and turbine blades) etc. Laminar-to-turbulent transition (hereafter referred to as transition) can be caused either by the amplification of small perturbations in the flow characterized by the so-called Tollmien-Schlichting waves, or via external disturbances such as freestream turbulence, surface roughness, flow separation, impinging wakes etc. where turbulent spots are introduced into the flow. The former is referred to as the natural transition and the latter is referred to as bypass transition. Predicting either modes of transition is a difficult task as they involve intricate interactions between flow quantities across a range of spatio-temporal scales. We shall focus our attention here to bypass transition modeling.

In fluid flows which are characterized by a freestream turbulence intensity of more than 1%, boundary layers can transition without the occurrence of the Tollmien-Schlichting waves [44]. Instead, the freestream turbulence triggers elongated disturbances (referred to as Klebanoff modes) which precede the onset of transition. These are formed due to the low-frequency disturbances from the freestream which penetrate the boundary layer. Given their 3-D nature, Klebanoff modes manifest themselves in the flow as forward and backward moving jets (or streaks) along the streamwise direction. As these streaks get lifted up towards the edge of the boundary layer, they interact with the high-frequency freestream disturbances, which leads to formation of turbulent spots that eventually lead to transition. Alternatively, transition can also be caused when a boundary layer undergoes separation and the shear layer encounters an inflection-point instability that triggers transition. In such a case, the consequent turbulent shear layer can reattach to the surface due to enhanced mixing resulting in what is referred to as a laminar-separation turbulent-reattachment bubble. It has been seen that the bubble length in the freestream direction is very sensitive to changes in the Reynolds number and angle of attack [45]. Note that for cases with high enough freestream turbulence, the flow may exhibit all three kinds of instabilities - inflection point, Tollmien-Schlichting waves, and Klebanoff modes - simultaneously, which interact with each other. Other pathways to bypass transition include surface roughness and wake impingement. Since it is impossible to capture minute details and interactions of such spatio-temporal structures using a steady Reynolds-Averaged Navier-Stokes (RANS) model for either of the aforementioned routes to bypass transition, empirical correlations are often used to predict transition onset and transition length for such simulations.

There are two major categories of approaches to model bypass transition in the context of RANS simulations - data correlation models and transport equation based models. Data correlation models make use of empirical correlations to predict the transition location where a switch can be made from laminar to turbulent computation to make predictions. Mayle [44] correlated the transition onset location for zero pressure gradient flow by a relatively simple expression as shown in Eqn. 4.1.

$$Re_{\theta,t} = 400\ Tu^{-0.625} \tag{4.1}$$

Abu-Ghannam and Shaw [1] presented such a correlation in 1980 to predict the momentum thickness at transition location while accounting for pressure gradients as shown in

Eqn. 4.2.

$$
\begin{aligned}
Re_{\theta,t} &= 163 + \exp\left[F(\lambda_\theta) - \frac{F(\lambda_\theta)}{6.91} Tu\right] \\
F(\lambda) &= \begin{cases} 6.91 + 12.75\lambda_\theta + 63.64\lambda_\theta^2 & \text{if } \lambda_\theta \geq 0 \\ 6.91 + 2.48\lambda_\theta - 12.27\lambda_\theta^2 & \text{otherwise} \end{cases}
\end{aligned}
\tag{4.2}
$$

Here θ refers to the momentum thickness, $\lambda_\theta = \dfrac{\theta^2}{\nu}\dfrac{\partial U_\infty}{\partial x}$ refers to the local pressure gradient parameter, and $Tu = \dfrac{100\sqrt{2k/3}}{U_\infty}$ refers to the freestream turbulence intensity. Suzen et al. [79] provided a similar correlation (Eqn. 4.3) which provides slightly better predictions for favorable pressure gradients.

$$
Re_{\theta,t} = \left(120 + 150\ Tu^{-2/3}\right) \coth\left(1.2 - 40000K\right) \tag{4.3}
$$

Here $K = \dfrac{\nu}{U_\infty^2}\dfrac{\partial U_\infty}{\partial x}$ refers to the acceleration parameter. Note that the switch from laminar to turbulent computation for data correlation models can be made gradual instead of a sudden jump across the transition location by introducing an intermittency term. Intermittency is a statistical quantity which is defined as the fraction of the time that the flow remains turbulent at a given location. Thus, intermittency is zero in fully laminar regions of the flow and unity in fully turbulent regions. Several algebraic models have been introduced to model the behavior of intermittency along the streamwise direction. One of the earliest and widely used among such models is the one proposed by Dhawan and Narasimha [12] which builds on the probability transition theory put forward by Emmons [18].

$$
\gamma = 1 - e^{-(x-x_t)^2/\ell_t^2} \qquad \forall\ x \geq x_t \tag{4.4}
$$

This model is given in Eqn. 4.4 where $\ell_t = \nu/(\sqrt{\widehat{n}\sigma}U_\infty)$ refers to a transition length. $\widehat{n}\sigma$ is related to the propagation rate of turbulent spots in a laminar boundary layer. A correlation for $\widehat{n}\sigma$ was provided by Steelant and Dick [72] in terms of Tu and K.

Transport equation based models, on the other hand, introduce additional transport equations into the RANS model corresponding to new scalar quantities that are used to diminish the eddy viscosity in the pre-transitional regions within the boundary layer either directly or indirectly. These models provides a more powerful approach to model the bypass transition phenomena as they account for the flow history effects and can also make use of existing empirical correlations in an approximate sense. Nearly all transport equation models use either laminar kinetic energy (energy contained within Klebanoff modes in a laminar boundary layer), k_L, or the intermittency (γ) as the transport scalar. Based on this choice, they are categorized into laminar fluctuation models and intermittency transport models, respectively.

Laminar fluctuation models, in general, are written as shown in Eqn. 4.5.

$$\frac{Dk_L}{Dt} = P_L - D_L + \boldsymbol{\nabla} \cdot T_L - R \tag{4.5}$$

Here, P_L, D_L, and T_L refer to production, destruction and transport of the laminar kinetic energy. The term R refers to the energy transfer from Klebanoff modes to turbulent fluctuations. R is also added as an extra source term in the transport equation for turbulent kinetic energy. Also note here that the term T_L is assumed to be purely viscous in nature. Notable laminar fluctuation models include the ones by Mayle and Schulz [45], Lardeau et al. [38], and Walters and Cokljat [86] among others.

Intermittency transport models, on the other hand, take effect by restricting the eddy viscosity to low values in the pretransitional regions of the boundary layer. This is done by multiplying either the eddy viscosity or the production term in the transport equation for turbulent kinetic energy with the intermittency variable. It should be noted that the intermittency transport models are a relatively new development and have proved more successful compared to their laminar fluctuation counterparts. The underlying transport equation stems from Eqn. 4.6 which can be derived from the correlations presented by Dhawan and Narasimha [12] for $x \geq x_t$ (where x_t refers to the transition location) under the approximation of small γ. Readers are directed to section 2.3.3.2 of [22] for the derivation in detail.

$$\frac{d\gamma}{dt} + \boldsymbol{u} \cdot \boldsymbol{\nabla}\gamma = 2(1-\gamma)\sqrt{\gamma}\frac{|\boldsymbol{u}|}{\ell_t} + \boldsymbol{\nabla} \cdot ((\nu + \nu_t)\,\boldsymbol{\nabla}\gamma) \tag{4.6}$$

Note here that while this transport equation does not contain x_t within its formulation, it has been derived using the algebraic correlation defined for $x \geq x_t$. Hence, the corresponding source term must remain inactive until the flow reaches the transition location or else the model shall predict fully turbulent flow at all locations. Most of the intermittency transport models achieve this by introducing an additional sink term which suppresses the intermittency in the pretransitional region. One of the earliest intermittency function models was proposed by Steelant and Dick [72] where a data correlation model was used to predict the transition onset location but its use was limited to only boundary layer computations. Suzen and Huang [78] extended this model to predict realistic γ distributions in the cross-stream direction. However, this extended model still depended on data correlation to predict the transition onset. Langtry and Menter [37] proposed one of the most widely used transition models in 2009 by modifying the SST k-ω model. In their method, data correlation was replaced by an auxiliary transport equation for the transition Reynolds number $Re_{\theta,t}$. This auxiliary transport equation uses a source term which activates outside the boundary layer to solve for $Re_{\theta,t}$ using correlations to local flow quantities and turns off within the boundary layer to let these values diffuse from the freestream. A local quantity, viz., the vorticity Reynolds number ($Re_\Omega = d^2\Omega/\nu$), was used as a surrogate to Re_θ (which is an integral quantity) when comparing with the critical Reynolds number ($Re_{\theta,c}$) to predict the onset of transition. Critical Reynolds number corresponds to the location where intermittency starts ramping up from zero and the transition occurs slightly downstream of this location. Note here that $Re_{\theta,c}$ is computed within the model as an empirical function of $Re_{\theta,t}$. It should be noted that the computation of $Re_{\theta,t}$, in their model, requires solving an implicit algebraic equation for the acceleration parameter λ_θ. The main criticism against this model is that it makes use of streamline direction and mean velocity (when estimating Tu and λ_θ to evaluate $Re_{\theta,t}$ in the freestream) which are not Galilean invariant and can be an issue for configurations with multiple moving surfaces. In 2012, Durbin [16] proposed a model which attempted to alleviate this problem by using only local Galilean invariant quantities. The model form he used is shown in Eqn. 4.7 and looks very similar to Eqn. 4.6.

$$\frac{d\gamma}{dt} + \boldsymbol{u} \cdot \boldsymbol{\nabla}\gamma = F_\gamma |\Omega| (\gamma_{\max} - \gamma)\sqrt{\gamma} + \boldsymbol{\nabla} \cdot \left(\left(\frac{\nu}{\sigma_l} + \frac{\nu_t}{\sigma_t} \right) \boldsymbol{\nabla}\gamma \right) \tag{4.7}$$

$\gamma_{\max}$ is set to a constant value of 1.1. This value is chosen to enable the intermittency to rapidly increase to unity. To ensure its value does not exceed beyond 1, the intermittency is explicitly reduced if it exceeds unity at all spatial locations in the computational domain after every solver iteration. The F_γ term is a limiter which serves two purposes. First, it activates when the ratio of Re_Ω and $(\nu_t\Omega)/(\nu\omega)$ exceeds beyond a empirically obtained constant value. Second, even when this criterion is reached, it deactivates if Re_Ω is too small. Note the absence of a sink term in the model. As discussed before, without a sink term the intermittency would reach unity in the entire domain by virtue of diffusion. To circumvent this issue, this model requires the intermittency to be explicitly set to zero for spatial locations where the eddy viscosity is significantly less than the molecular viscosity and/or Re_Ω is less than a predefined threshold after each solver iteration. An improved version of this model [23] was presented by Ge et al. in 2014. Major changes in this version included the introduction of a sink term and a modification to improve the predictive accuracy for separation-induced transition.

While one of these models can be chosen as a baseline model and augmented to improve predictive accuracy, this work presents a bare-bones intermittency-based transition model inspired from Durbin's model (2012) to demonstrate the capability of the LIFE framework. This chapter is broadly divided into two main parts. Section 4.2 details the inference of a model that can be used to predict transition due to freestream turbulence using the LIFE framework. Following that, section 4.3 describes the development of a hierarchical model that when used with the one described in section 4.2 can be used to predict separation-induced transition.

4.2 A Data-driven Bypass Transition Model

To build a data-driven model that can predict bypass transition triggered by turbulent fluctuations in the freestream, an augmented intermittency transport equation, similar in structure to the one in Durbin's 2012 model [16], was added to the Wilcox's 1988 k-ω model (with a vorticity based source term). A bounded feature-space was then carefully designed to enable identification of pre-transitional regions in the boundary layer where TKE production needs to be diminished. To enable localized learning, a simple interpolation-based functional form were chosen. Only two cases from the T3 dataset [58], viz., T3A and T3C1, were used to infer the augmentation function. The so-obtained augmentation was then tested on the T3B, T3C2, T3C3 and T3C4 cases from the T3 dataset; and the MUR116, MUR129, MUR224 and MUR241 cases from the VKI turbine cascade dataset. The augmentation shows good generalizability across all these different datasets which are characterized by different geometries, freestream turbulence intensities, Reynolds numbers, Mach Numbers, pressure gradients, etc.

4.2.1 Augmenting the 1988 k-ω turbulence model

Wilcox's k-ω model from 1988 [90] (with the production term modified to use vorticity magnitude instead of strain-rate magnitude) was chosen as the baseline turbulence model to be augmented for transition prediction. The PDEs for this two-equation model are given in equation 4.8.

$$\begin{aligned}
\rho\frac{\partial k}{\partial t} + \rho\boldsymbol{u}\cdot\boldsymbol{\nabla}k &= \mu_t\Omega^2 - \frac{2}{3}\rho k\boldsymbol{\nabla}\cdot\boldsymbol{u} - C_\mu\rho k\omega + \boldsymbol{\nabla}\cdot\left(\left(\mu+\frac{\mu_t}{\sigma_k}\right)\boldsymbol{\nabla}k\right) \\
\rho\frac{\partial \omega}{\partial t} + \rho\boldsymbol{u}\cdot\boldsymbol{\nabla}\omega &= C_{\omega 1}\frac{\omega}{k}\left(\mu_t\Omega^2 - \frac{2}{3}\rho k\boldsymbol{\nabla}\cdot\boldsymbol{u}\right) - C_{\omega 2}\rho\omega^2 + \boldsymbol{\nabla}\cdot\left(\left(\mu+\frac{\mu_t}{\sigma_\omega}\right)\boldsymbol{\nabla}\omega\right)
\end{aligned} \tag{4.8}$$

Here, the eddy viscosity (μ_t) is simply k/ω. Taking inspiration from Durbin's model from 2012 [16], an augmented PDE for intermittency can be written as in equation 4.9. Note that the term β_1 replaces the $\gamma_{\max}$ term in Durbin's model. While in Durbin's model, $\gamma_{\max}$ has a constant value, β_1 in the current model is a function and hence can make the source term act either like a production or a destruction term, as needed.

$$\rho\frac{\partial\gamma}{\partial t}+\rho\boldsymbol{u}\cdot\boldsymbol{\nabla}\gamma=\rho\left(\beta_1(\boldsymbol{\eta})-\gamma\right)\sqrt{\gamma}\Omega+\boldsymbol{\nabla}\cdot\left(\left(\frac{\mu}{\sigma_l}+\frac{\mu_t}{\sigma_k}\right)\boldsymbol{\nabla}\gamma\right) \tag{4.9}$$

The intermittency term is then multiplied to the production term in the k-equation, and hence, the entire augmented model can be written as in equation 4.10.

$$\begin{aligned}
&\rho\frac{\partial k}{\partial t}+\rho\boldsymbol{u}\cdot\boldsymbol{\nabla}k=\gamma\left(\mu_t\Omega^2-\frac{2}{3}\rho k\boldsymbol{\nabla}\cdot\boldsymbol{u}\right)-C_\mu\rho k\omega+\boldsymbol{\nabla}\cdot\left(\left(\mu+\frac{\mu_t}{\sigma_k}\right)\boldsymbol{\nabla}k\right)\\
&\rho\frac{\partial\omega}{\partial t}+\rho\boldsymbol{u}\cdot\boldsymbol{\nabla}\omega=C_{\omega1}\frac{\omega}{k}\left(\mu_t\Omega^2-\frac{2}{3}\rho k\boldsymbol{\nabla}\cdot\boldsymbol{u}\right)-C_{\omega2}\rho\omega^2+\boldsymbol{\nabla}\cdot\left(\left(\mu+\frac{\mu_t}{\sigma_\omega}\right)\boldsymbol{\nabla}\omega\right)\\
&\rho\frac{\partial\gamma}{\partial t}+\rho\boldsymbol{u}\cdot\boldsymbol{\nabla}\gamma=\rho\left(\beta_1(\boldsymbol{\eta})-\gamma\right)\sqrt{\gamma}\Omega+\boldsymbol{\nabla}\cdot\left(\left(\frac{\mu}{\sigma_l}+\frac{\mu_t}{\sigma_k}\right)\boldsymbol{\nabla}\gamma\right)
\end{aligned} \tag{4.10}$$

A few salient points regarding the construction of this augmented intermittency equation are as follows:

- The structure of the production term is identical to Durbin's model in order to make sure that the distribution of the intermittency is consistent with what Dhawan and Narsimha proposed in 1957.

- The diffusion term is again identical to Durbin's model with the reasoning that the diffusion characteristics of intermittency should remain, more or less, the same for both the models.

- The augmentation is introduced instead of the term γ_{max} in Durbin's model. In essence, this model-form of the intermittency equation would constrain the intermittency field to be as close to the augmentation field as possible. This also precludes the need for a sink term in the model as the augmentation function can act as both a source and a sink term.

- An intermittency equation was used instead of simply multiplying the production term in the k-equation with the augmentation so that the augmentation field can be smoothed into an intermittency field to aid solver stability and convergence.

- This particular way of augmenting the model results in a bounded augmentation (between 0 and 1) as the intermittency is a mathematically bounded quantity. A bounded augmentation would, in general, require a simpler functional form compared to an unbounded augmentation. Although, there might be certain cases characterized by separation-induced transition where the intermittency might slightly exceed 1, but would still remain below a conservative upper-bound of 2.
- All the information about which regions in the flow need to be laminar and which regions need to be turbulent, i.e., the information about the model inadequacy in Wilcox's k-ω model that we are trying to alleviate, comes from the augmentation itself.

4.2.2 Feature Design

The task of designing features for a complex application like predicting bypass transition requires considerable effort. This section outlines the rationale behind consideration of different feature candidates in an attempt to illustrate the nuances involved in this process.

In order to design a parsimonious feature-space which can distinguish between physical conditions relevant to the bypass transition phenomena at different locations in the flow, some basic understanding of the interplay between various model quantities is required.

At any stage in the simulation, if the production of k can be suppressed around the region in the buffer layer where ω is barely low enough to facilitate a net production of k in the first place, the value of k will drop in this region. This would result in the values of k dropping in neighboring regions which are farther from the wall too as a result of diffusion, thus lowering the value of μ_t in these regions. A drop in μ_t will lower the production term, thus making k drop even further. The negative feedback of k (by the virtue of the source term), cascade of lowering k values farther from the wall (by the virtue of diffusion term), and the lowered/nullified influence of high values of k downstream (by the virtue of the convective term) would result in the flow laminarizing at a given streamwise location.

Since the intermittency (γ) is being used to suppress the production of k (and hence locally laminarize the flow), very low values of γ would clearly be correlated to regions within a laminar boundary layer. Note that there do exist other laminar regions in the flow, viz., regions within the viscous sublayer of the turbulent parts of the boundary layer. Now that the task at hand is clear, appropriate features need to be designed to accomplish flow laminarization up to the transition location. This would require:

1. Feature(s) that can diagnose if the streamwise location that they correspond to is downstream or upstream of the transition location.

2. Feature(s) that can distinguish between points inside the viscous sublayer of a turbulent boundary layer and those within a laminar boundary layer.

3. Feature(s) that can distinguish between laminar and turbulent parts of the flow.

The following subsections will detail the different requirements and resulting feature choices that arose in a logical sequence.

Identifying whether transition has occurred

As described in section 4.1, several algebraic models (e.g., [1]) make use of comparisons between the local value of Re_θ and an estimate of $Re_{\theta,t}$ (obtained via empirical correlations) to ascertain whether a given streamwise location lies in the laminar or turbulent part of the boundary layer. Note here, that while transition momentum thickness Reynolds number, $Re_{\theta,t}$ (i.e., where the transition starts) is different from critical momentum thickness Reynolds number, $Re_{\theta,c}$ (i.e., where the intermittency starts increasing), in the interest of simplicity it was decided that only an $Re_{\theta,t}$ estimate shall be used in the process of feature design.

Since Re_θ is a non-local quantity (not to mention that it is very hard to compute it in the presence of strong pressure gradients), a local, more robust and preferably easy-to-compute quantity is required for feature design. Similarly, better correlations are required for $Re_{\theta,t}$ that depend on as much local information as possible.

A surrogate for Re_θ: The vorticity Reynolds number, Re_Ω, is a local flow quantity which is defined as follows in equation 4.11

$$Re_\Omega = \frac{\rho \Omega d_w^2}{\mu} \tag{4.11}$$

Here, Ω is the vorticity magnitude and d_w is the wall distance. This quantity was first used in the context of transition by van Driest and Blumer [83], where they proposed that the onset of transition can be modeled to occur when Re_Ω crosses a set threshold. Using the Blasius boundary layer solution given by Wilcox [91], Menter et al. [46] noted that the maximum value of Re_Ω in the wall-normal direction can be scaled to match Re_θ for any streamwise location in a Blasius boundary layer profile with good accuracy. Different researchers provide slightly different values for this scaling. In this work, the following scaling is used (as given by Durbin [16]).

$$Re_v = \frac{Re_\Omega}{2.188}, \qquad \max_{d_w} Re_v \approx Re_\theta \tag{4.12}$$

In addition, Langtry and Sjolander [36] observed that for different streamwise locations, the wall-normal location corresponding to maximum Re_Ω roughly matches the respective wall-normal location corresponding to the most rapid growth of laminar fluctuations (as observed from experimental data) very well. In Falkner-Skan boundary layers, $Re_v < Re_\theta$ for favorable gradients and $Re_v > Re_\theta$ for adverse pressure gradients. Since the flow transitions at lower values of Re_θ in regions of adverse pressure gradient and vice versa, this means that the same threshold for Re_v will cause the flow to transition sooner in regions of adverse pressure gradients and delay it in regions of favorable pressure gradients, which is desirable. This makes Re_v a very strong candidate for use as a surrogate to Re_θ for transition applications.

A surrogate for $Re_{\theta,t}$: The transition momentum thickness Reynolds number $Re_{\theta,t}$ depends on several factors, viz., freestream turbulence intensity, pressure gradient, Mach number etc. All of these quantities are determined by the inviscid part of the flow, i.e., the flow outside the boundary layer. This means that we need to estimate $Re_{\theta,t}$ outside the boundary layer and somehow transfer that value within the boundary layer. For instance, this has been done using an additional transport equation for $Re_{\theta,t}$ in the Langtry-Menter model. In this model, however, we choose a simpler route. Given that the freestream quantities will be only marginally modified (if at all) even with different intermittency distributions within the boundary layer, one can – for a given flow geometry and boundary conditions – assume the values of $Re_{\theta,t}$ in all the cells as flow parameters themselves. In such a case, a crude way of transferring information from the freestream into the boundary layer is by copying the freestream $Re_{\theta,t}$ estimates to all cells along the wall-normal direction. In this work, the index of the closest wall face is stored for all cells in the computational domain. Then, for every wall face, the freestream quantities are estimated as the average values of all cells that are located at a distance between $r - \delta r/2$ and $r + \delta r/2$ from the wall and have the wall face in consideration as their closest wall face. r and δr are user-specified constant values such that the boundary layer thickness at all locations is less than r and δr is sufficiently large to import values from at least one cell per wall face.

The task, now, is to find a simple approximation for $Re_{\theta,t}$. Praisner and Clark [54] obtained a simple correlation between the transition onset momentum thickness Reynolds number $Re_{\theta,t}$, turbulence intensity at the edge of the boundary layer (Tu_e), turbulence length scale at the edge of boundary layer (λ_e), and momentum thickness. This correlation is given as follows.

$$Re_{\theta,t} = A\left(Tu_e \frac{\lambda_e}{\theta}\right)^B \tag{4.13}$$

The values of the constants A and B were calibrated as 8.52 and -0.956, respectively, in order to match the transition onset data for several turbomachinery configurations spanning a range of Mach numbers, freestream turbulence intensities, pressure gradients and wall temperatures. In their paper, they further simplified the expression using $Tu_e =$

$100(u'_e/U_e)$ and assuming $B \approx -1$ as

$$100\left(\frac{u'_e}{\lambda_e}\right)\left(\frac{\rho_e \theta_t^2}{\mu_e}\right) = A_1 \tag{4.14}$$

In this form, they found $A_1 = 7.0 \pm 1.1$. In Wilcox's k-ω model u'_e/λ_e can be approximated using $C_\mu \omega_e$. Using this approximation along with the value $C_\mu = 0.09$, the relationship can be rewritten as follows.

$$\theta_t^2 = \frac{7\nu_e}{9\omega_e} \tag{4.15}$$

Here, it must be noted that in order for this approximation to hold, the freestream boundary condition for ω must be appropriately chosen such that the decay of turbulence intensity is consistent with observations from experiments or high-fidelity simulations. Finally we can write the surrogate for $Re_{\theta,t}$ as follows.

$$\overline{Re_{\theta,t}} = \sqrt{\frac{7U_e^2}{9\nu_e \omega_e}} \tag{4.16}$$

Note here that $\overline{Re_{\theta,t}}$ is only an approximation of and can be different from the true value of $Re_{\theta,t}$. Also note that while the estimate of U_e can be used with sufficient certainty in moderate pressure gradient problems, the quantity could be ambiguous for very high pressure gradients. The maximum value of Re_v in the wall-normal direction approximates Re_θ for zero pressure gradients and hence, $\max_{d_w} Re_v$ would also poorly estimate Re_θ for very high pressure gradients. On a different note, U_e does not follow Galilean invariance and hence might pose problems when predicting for configurations involving multiple moving surfaces.

Functional form for the resulting feature: The objective of this feature is to compare the surrogate values of Re_θ and $Re_{\theta,t}$ in order to predict whether transition has occurred. Although several approximations have been made while evaluating Re_v and $\overline{Re_{\theta,t}}$, they should still scale correctly. We are interested in regions where intermittency exhibits values significantly far from zero or unity and this behaviour would clearly correspond to regions where $\mathcal{O}(Re_v) \approx \mathcal{O}(\overline{Re_{\theta,t}})$. At other locations, if Re_v is small, the intermittency needs to be predicted close to zero, and if it is too high intermittency needs to be predicted close to unity. The region of interest would thus correspond to values of $Re_v/\overline{Re_{\theta,t}}$ around 1. If this ratio is too high, the intermittency should be 1. A conservative upper-bound of 3 is used for this feature. Note that the feature can only assume positive values and hence a lower-bound is not required. The final functional form of the first feature then becomes as shown in equation 4.17

$$\eta_1 = \min\left(\frac{Re_v}{\overline{Re_{\theta,t}}}, 3\right) \tag{4.17}$$

There, however, arise two new issues when using this feature. Firstly, the same value of Re_v can correspond to different locations within the laminar and turbulent boundary layer regions which might require different treatments. Secondly, the vorticity magnitude might not vanish away from the wall resulting in increasing values of Re_v as wall distance increases beyond the boundary layer. Hence, features are needed to: (1) classify laminar and turbulent regions and; (2) isolate laminar boundary layer regions from the regions outside the boundary layer as the augmentation needs to deviate from its baseline value only within the laminar regions of the boundary layer.

Identifying laminar and turbulent regions

Turbulent length scales can be used to identify regions inside the boundary layer which are laminar and turbulent. This can be done by using the ratio of wall distance (d_w) and turbulent length scale (estimated as $\sqrt{k}/\omega$ in the k-ω model) in order to compare them against each other. Note here that according to the sublayer analysis shown by Wilcox [90] we have the following.

$$\omega \to \frac{6\nu}{\beta d_w^2} \quad \text{and} \quad k \sim d_w^{3.23} \quad \text{for} \quad d_w \to 0 \tag{4.18}$$

This means that the estimated turbulent length scale, $\sqrt{k}/\omega$ will vary in proportion to $d_w^{3.615}$ close to a wall. So, behaviorally it would be better if the turbulent length scale was compared against $d_w^{3.615}$ but that would not result in a non-dimensional feature which would result in a severe loss of generalizability. The highly turbulent regions in a boundary layer (where intermittency would be close to unity) will correspond to $d_w/(\sqrt{k}/\omega) \ll 1$. Locations within the pre-transitional boundary layer, regions in viscous sublayer close to the wall and regions in freestream far enough from the wall will correspond to $d_w/(\sqrt{k}/\omega) \gg 1$. Note here that the third feature is designed to ensure that, among these three regions, the intermittency is predicted close to zero only for locations within the pre-transitional boundary layer. Finally, in order to make the feature bounded, the functional form is changed to as shown in equation 4.19.

$$\eta_2 = \frac{d_w}{d_w + \sqrt{k}/\omega} \tag{4.19}$$

Thus, $\eta_2 \rightarrow 0$ corresponds to $d_w/(\sqrt{k}/\omega) \ll 1$, and $\eta_2 \rightarrow 1$ corresponds to $d_w/(\sqrt{k}/\omega) \gg 1$.

Feature 3 - Distinguishing the pre-transitional boundary layer

Assuming the augmentation predicts low enough intermittency such that the eddy viscosity in the pre-transitional boundary layer is calculated as $\nu_t < \nu$, and also assuming that the freestream values of ν_t would always remain more than ν, regions in the pre-transitional boundary layer and those in the freestream can be distinguished using the quantity ν/ν_t. As far as the viscous sublayer is concerned, we have,

$$\lim_{d\rightarrow 0} \eta_2 = \lim_{d\rightarrow 0} \frac{d_w}{d_w + \sqrt{k}/\omega} \approx \lim_{d\rightarrow 0} \frac{1}{1 + Cd_w^{2.615}} = 1 \tag{4.20}$$

Thus, η_2 would start from 1 at the wall and decrease away from it, slowly in the viscous sublayer and then rapidly thereafter as the production of k picks up. On the other hand, $\nu/\nu_t \rightarrow \infty$ as $d_w \rightarrow 0$ and it will start decreasing rapidly as one moves away from the wall as well. Note that very close to the wall, it is very difficult to distinguish between the viscous sublayer of a turbulent boundary layer and a laminar boundary layer. Thus the intermittency will be predicted significantly lower than 1 very close to the wall. This is fine because the net production of k is nearly zero and a dampening the production makes, virtually, no difference in the prediction of the wall shear stress. As one moves farther from the wall, the decreasing values of η_2 and ν/ν_t cause the intermittency to increase and hence, at a distance where the net k production reaches significant values, γ catches up and becomes close to 1. Finally, to make ν/ν_t bounded, the functional form is changed to obtain the feature as shown in equation 4.21

$$\eta_3 = \frac{\nu}{\nu + \nu_t} \tag{4.21}$$

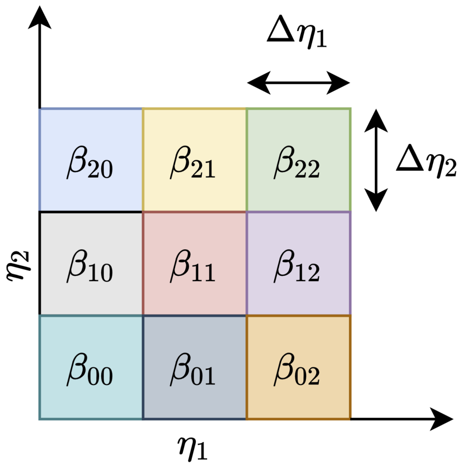
Unused feature candidates

While the features mentioned above were used as inputs to the augmentation function, several other candidates were also considered. Note here that the fact that attempts made using these candidates rendered inferior results compared to the aforementioned features does not mean that no variations of these candidates can be used as inputs to the augmentation function. They are listed here along with the rationale behind their consideration to help the reader in the development of similar features for fluid flow applications.

While deliberating the use of wall distance d_w in η_2, attempts were made to use the laminar length scale $\sqrt{\nu/\Omega}$, instead. However, for all variations that were tried out, it was observed that this caused the inference process to laminarize the entire boundary layer. Following this, bounded non-dimensional functions involving laminar and turbulent energy scales (viz., $\nu\Omega$ and k respectively) were also considered as candidates for η_2 and η_3. However, similar problems involving laminarization of the entire boundary layer were encountered when using these feature candidates as well. Also note that such a feature cannot differentiate between a laminar boundary layer and a laminar sublayer within a turbulent boundary layer by itself and hence must be used with a feature like η_2. Different functional forms involving either specific dissipation Reynolds number ($Re_\omega = d_w^2\omega/\nu$) or its wall-normal derivative scaled by d_w were also considered as both feature candidates and surrogates for $Re_{\theta,t}$. However, both of these attempts resulted in inferior results compared to the current feature set. This is mainly because the behavior of Re_ω differs significantly when compared to Re_Ω in the buffer layer and lower log layer regions and this is the region where the augmentation takes effect. Even so, Re_ω seems to be an interesting alternative to $Re_{\theta,t}$ and further investigations need to be made to fully explore its viability in data-driven transition modeling.

4.2.3 Functional form of the augmentation

The functional form of the augmentation chosen here is defined by linear interpolation on a uniform grid constructed in a bounded feature-space. This belongs to the second family of approaches (interpolation on a discretized feature-space) described in Section 3.5.1. Cell-centered values are used to estimate Green-Gauss gradients which are then used to perform linear interpolation within every grid cell. This is illustrated with a two-dimensional feature-space example in Fig. 4.1. The cell-centered augmentation values, hence, are the parameters of the augmentation function in this instance, and the size of cells (defined by grid spacing) decides the region of feature-space influenced by these augmentation function parameters. The grid spacing also controls the "accuracy vs generalizability" tradeoff described in Chapter 3. A smaller grid spacing would help infer a better-resolved augmentation function in the feature-space but would also increase the requirement of diverse data that can populate a significant region of the feature-space. Hence, when faced with limited data, grid spacing should be treated as a hyper-parameter



For any feature space location in cell 11

$$\frac{\partial\beta}{\partial\eta_1} = \frac{\beta_{12} - \beta_{10}}{2\Delta\eta_1}$$

$$\frac{\partial\beta}{\partial\eta_2} = \frac{\beta_{21} - \beta_{01}}{2\Delta\eta_2}$$

$$\beta = \beta_{11} + \frac{\partial\beta}{\partial\eta_1}(\eta_1 - 1.5\Delta\eta_1)\ldots$$

$$\ldots + \frac{\partial\beta}{\partial\eta_2}(\eta_2 - 1.5\Delta\eta_2)$$

Figure 4.1: Schematic of a Green-Gauss interpolation-based two-dimensional augmentation map

that can be optimized, if needed.

4.2.4 Results

Inferring the Augmentation

Only two flat plate cases from the T3 dataset by ERCOFTAC [58], viz., T3A and T3C1, were used to infer the augmentation function. The T3A case is characterized by a zero pressure gradient flow with the freestream turbulence intensity of 3.5% at the inlet. The T3C cases, on the other hand, are characterized by a monotonically increasing pressure gradient along the flow direction. This is achieved by contouring the top surface of the domain using a correlation given by Suluksna et al. [74]. Fig. 4.2 shows the comparison of "estimated" freestream velocity profile in the streamwise direction with the experimental data. Since it is difficult to ascertain a constant distance away from the wall where the freestream velocity can be recorded, surface pressure, $p_w(x)$ is used to obtain an estimate of the freestream velocity as $U_\infty(x) = \sqrt{2(p_{\text{in}} - p_w(x))/\rho + U_{\text{in}}^2}$, where p_{in} and U_{in} are inflow pressure and inflow velocity, respectively. The trends seem to agree except for a slight discrepancy in the favorable pressure gradient region which might be caused due to the highly curved top wall in this region causing an error in the estimated freestream velocity. Presented in Fig. 4.3 are the decay profiles of freestream turbulence intensity with flow direction for the T3A and T3C1 cases which match very well with experimental data. The corresponding meshes for the T3A/T3B case and T3C cases are shown in

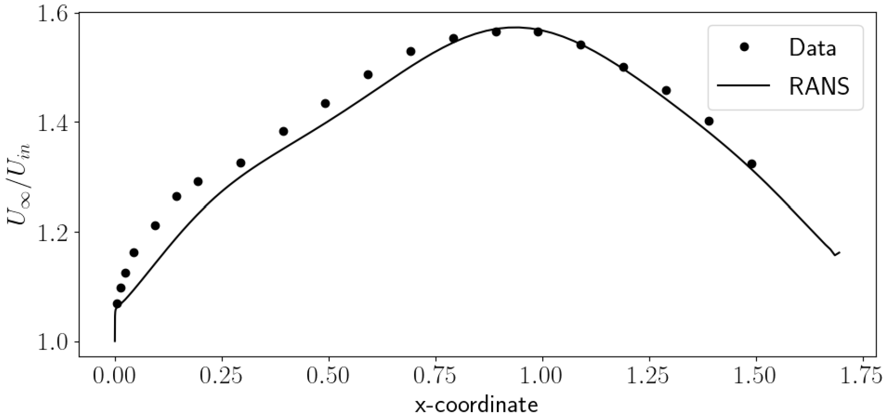
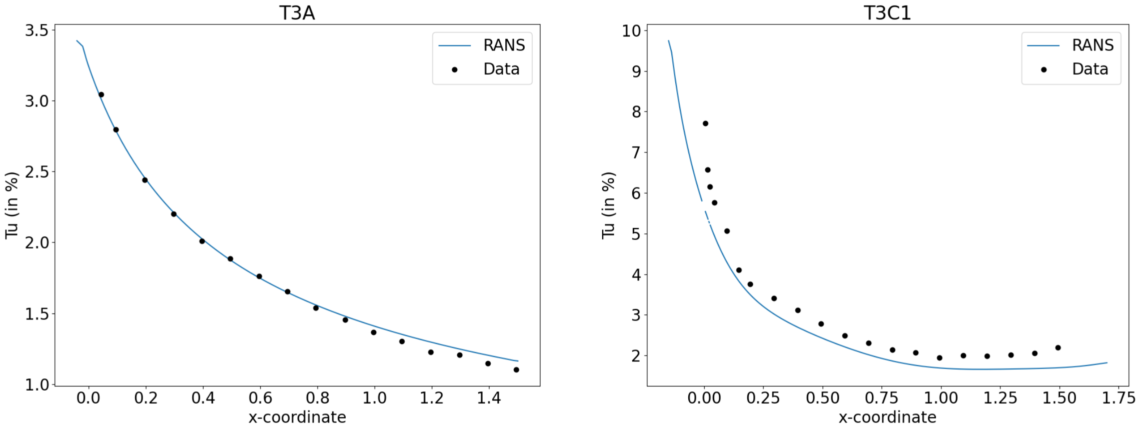


Figure 4.2: Comparison between predictions and data for U_∞/U_{in} profile

Figs. 4.4 and 4.5, respectively. This verifies that the boundary conditions for ω are



(a) T3A

(b) T3C1

Figure 4.3: Comparisons between predictions and data for decay of freestream turbulence

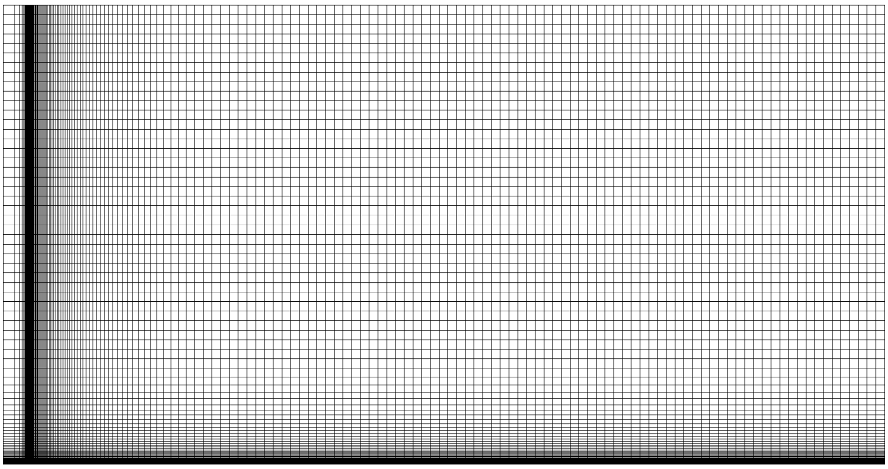


Figure 4.4: Mesh used for simulations of T3A and T3B flat plate cases

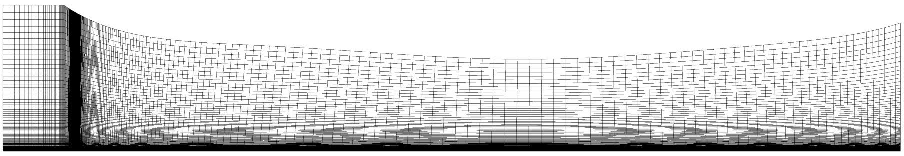


Figure 4.5: Mesh used for simulations of T3C cases

set correctly. Hence, the flow is subjected to favorable pressure gradients for the initial section of the plate followed by a region of adverse pressure gradients. For the T3C1 case, the transition occurs in the favorable pressure gradient region. The boundary conditions for both these flows is summarized in Table 4.1. The three-dimensional feature space

Cases	**T3A**	**T3C1**
Tu_{in}	0.035	0.1
$\nu_{t_{\text{in}}}/\nu$	14.0	50.0
$L(\text{in } m)$	1.5	1.65
$Re_{L,\text{in}}$	520000	660000
$\omega_{\text{in}}(\text{in } s^{-1})$	9.1	24.0

Table 4.1: Inflow conditions for the T3 cases used for training

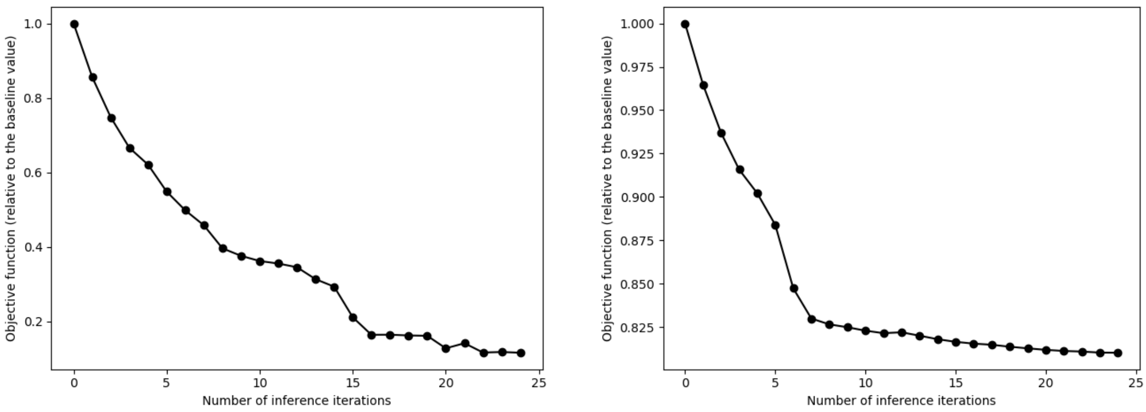
is discretized into a uniform grid with 30, 10, and 10 cells along features 1, 2 and 3, respectively. The combined cost function is defined to be the sum of mean squared discrepancies between skin friction values at all spatial locations where data is available for both the cases.

$$\mathcal{C} = \sum_{i_{\text{case}}}^{n_{\text{case}}} \|C_{f,i_{\text{case}}} - C_{f,i_{\text{case}}}^{\text{data}}\|_2^2$$

No regularization is used for this problem as using a standard Tikhonov regularization like $\|\beta_1 - 1\|_2^2$ would prevent β from reaching low enough values to cause laminarization. Also, it should be noted that there is ample implicit regularization present from three different sources:

1. The augmentation map is constrained to be a function of the chosen features
2. The augmentation map is required to provide best possible results for multiple training cases at the same time
3. The chosen functional form constrains the augmentation to manifest itself within a restricted family of functions

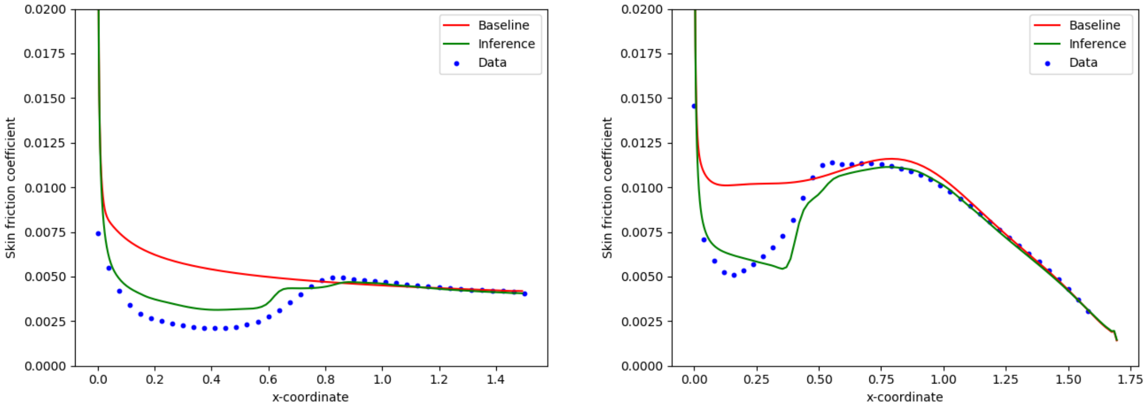
A steepest gradient descent algorithm is used to solve the optimization problem. The step length used for this gradient descent was $\dfrac{0.1}{\left\|\dfrac{\partial \mathcal{J}}{\partial \boldsymbol{w}}\right\|_\infty}$. The minimization of the objective function is shown in Fig. 4.6. While the objective function for either cases does not reach close to zero, the C_f plots for both the cases (see Fig. 4.7) clearly show that the augmented model predicts the transition locations well for both the cases. Note here that the baseline model simply uses $\beta_1 = 1$ everywhere in the computational domain and hence, in essence, predicts fully turbulent flow. Note that the C_f values for both the cases



(a) T3A

(b) T3C1

Figure 4.6: Objective minimization for simultaneous inference and learning from T3A and T3C1 cases

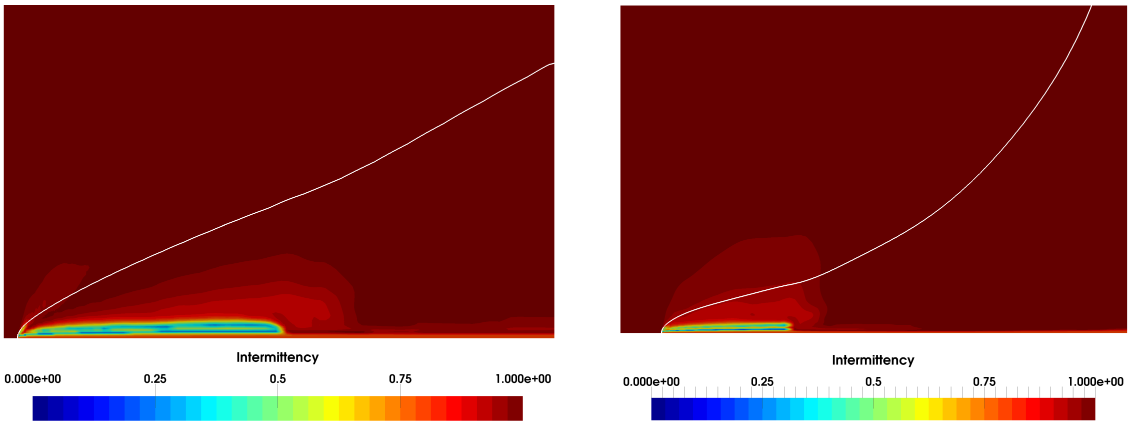


(a) T3A

(b) T3C1

Figure 4.7: Skin friction coefficient distributions for the inference cases

are slightly higher compared to the data in the respective pre-transitional regions. Also note that the transitional region for T3C1 is significantly more gradual compared to what the augmented model predicts. One of the main reasons for this behavior could be the constrained structure chosen for the augmentation function. This happens because the boundary layers are not fully laminarized in the pretransitional region of the boundary layer. This could be either due to the augmentation values not being predicted sufficiently low and/or insufficient thickness of the region within the boundary layer containing low augmentation values (see Fig. 4.8). The residual convergence plots for T3A corresponding to different inference iterations are shown in Fig. 4.9. The dashed lines correspond to the baseline model residuals, the solid lines correspond to the augmented model residuals,

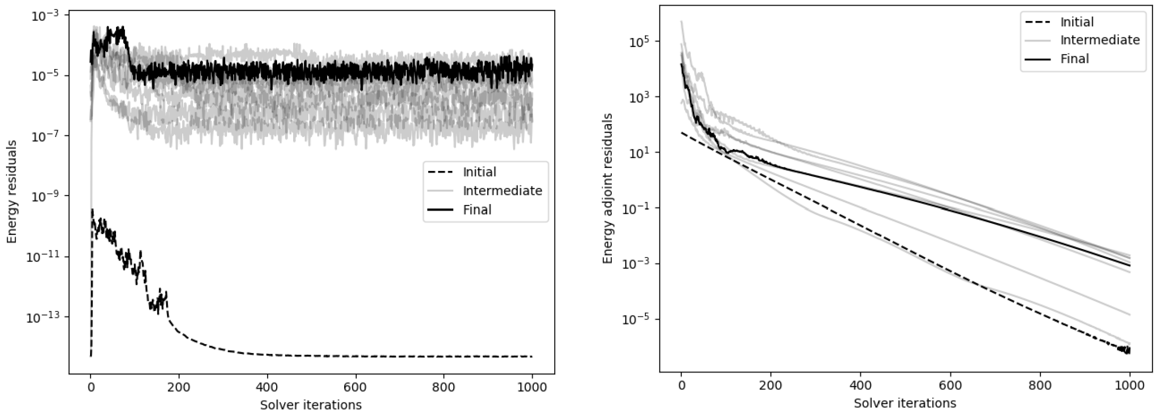


(a) T3A

(b) T3C1

Figure 4.8: Intermittency contours (γ) predicted by the augmented models for the cases used in inference. White line marks $U = 0.95U_\infty$. The scaling is 40x in the wall normal direction.

and the partially transparent curves correspond to the 23 inference iterations preceding the final iterate. Note that as the inference progresses the residual convergence grows



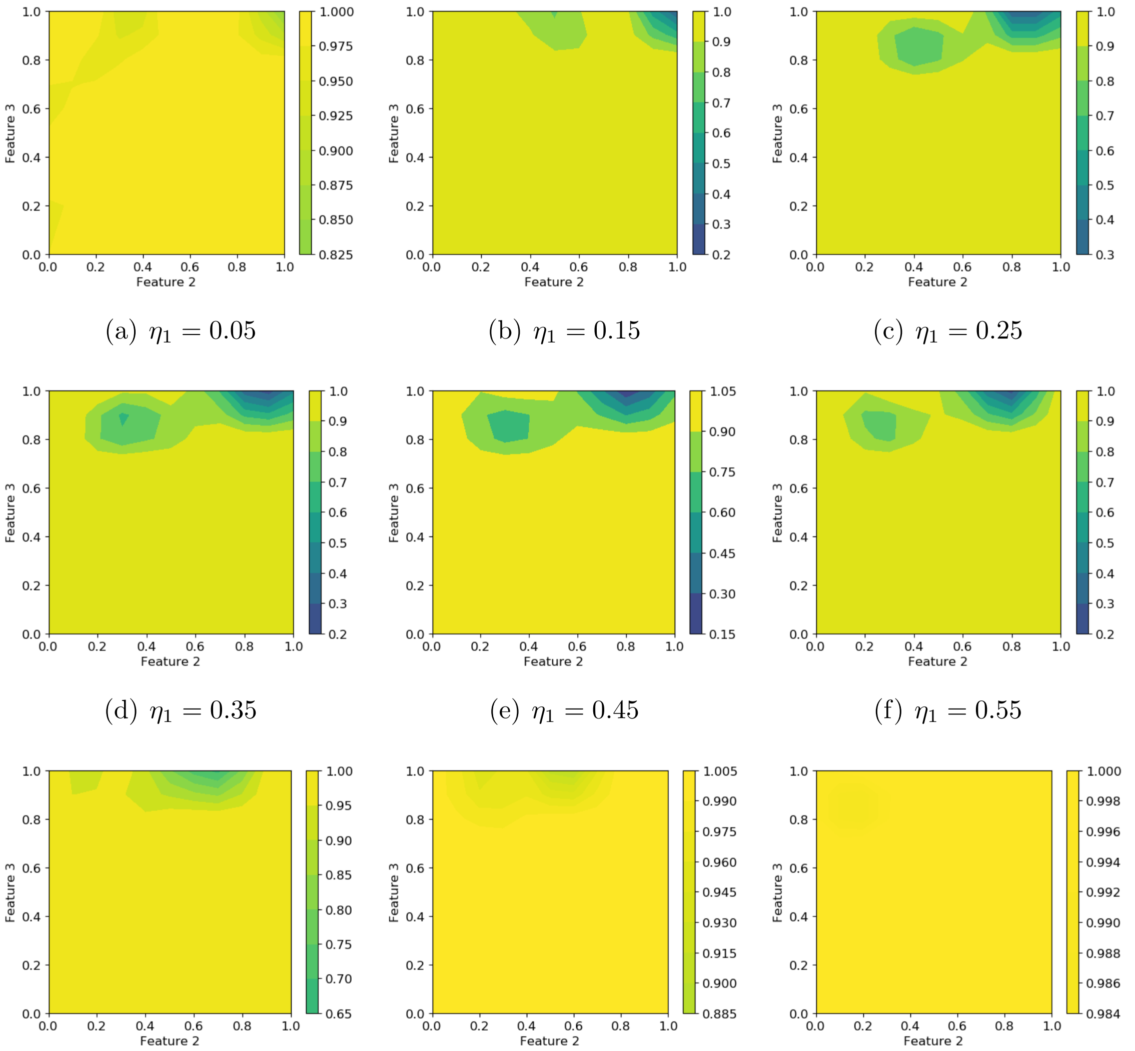
(a) Forward

(b) Adjoint

Figure 4.9: Forward and adjoint residual convergence across inference iterations for the T3A case

slightly worse while still remaining acceptable. The underlying reason for the residuals not converging to machine precision levels was later found to be the discontinuous nature of the augmentation at the cell interfaces in the feature-space grid.

The resulting augmentation map is shown in Fig. 4.10. It can be observed that the augmentation values are inferred to be small in regions of high η_2 and η_3 which correspond to laminar regions of the flow. Also, the augmentation reduces intermittency below a value of η_1 of about 0.75, above which the flow is always predicted turbulent.



(a) $\eta_1 = 0.05$

(b) $\eta_1 = 0.15$

(c) $\eta_1 = 0.25$

(d) $\eta_1 = 0.35$

(e) $\eta_1 = 0.45$

(f) $\eta_1 = 0.55$

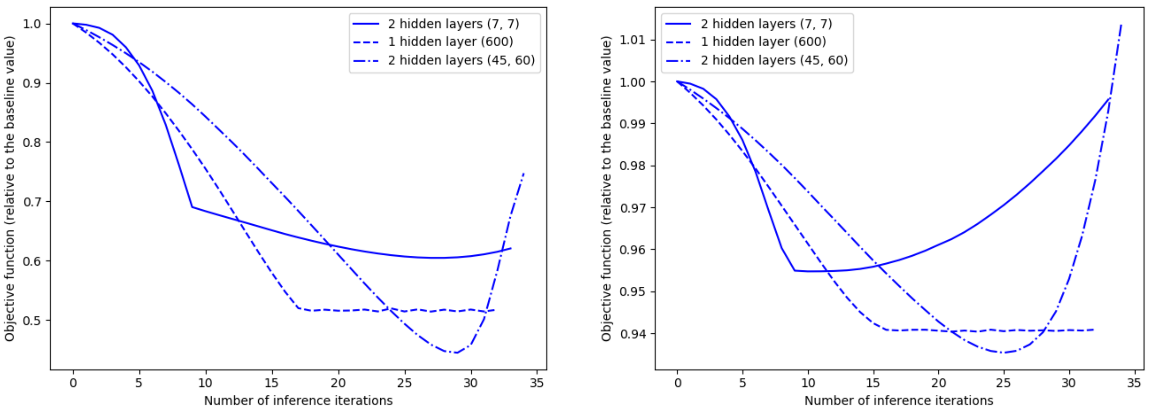
(g) $\eta_1 = 0.65$

(h) $\eta_1 = 0.75$

(i) $\eta_1 = 0.85$

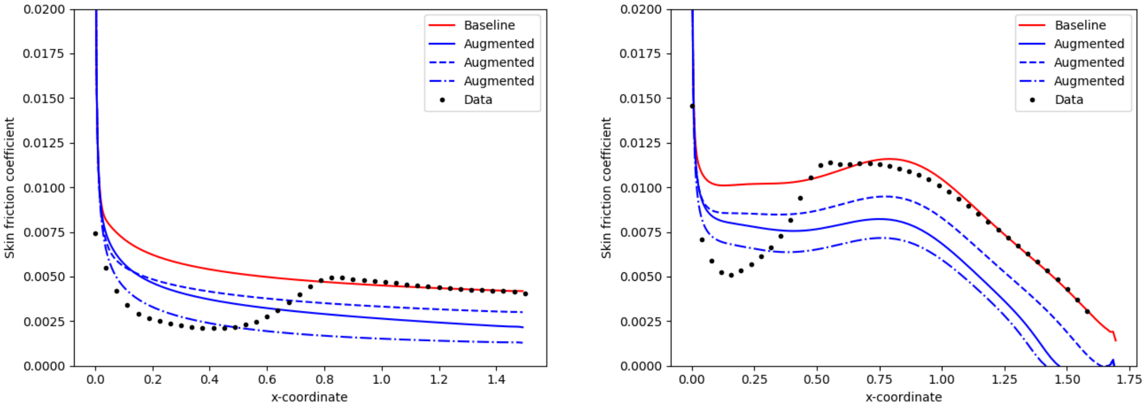
Figure 4.10: Augmentation contours of β_1 on feature space slices after training on T3A and T3C1 datasets

To illustrate the need of using a functional form capable of localized learning, Figs. 4.11 and 4.12 show the inference results from the functional forms being chosen as neural networks with three different architectures: with 2 hidden layers containing 7 nodes each, 1 hidden layer containing 600 nodes and 2 hidden layers containing 45 and 60 nodes, respectively. As shown in the plots, all three of these neural networks fail to infer the augmentation function from the T3A and T3C1 cases. The neural networks cause partial laminarization at all streamwise locations and, in some cases, this can lead to a continuous increase in objective function values with inference iterations as seen in Fig. 4.11. Note here that this is not a commentary on the ability of the neural network to



(a) T3A (b) T3C1

Figure 4.11: Objective minimization for simultaneous inference and learning from T3A and T3C1 cases using Neural Networks



(a) T3A (b) T3C1

Figure 4.12: Skin friction coefficient distributions for the inference cases with Neural Network based augmentation

represent the augmentation, rather its inability to prevent learned information in one part of the feature-space from being modified by inferred information in another part of the feature-space across training iterations.

Predictions on Unseen Cases

The augmentation function inferred from T3A and T3C1 as mentioned above was subsequently used to predict the skin friction distributions for other T3 cases, viz., T3B, T3C2, T3C3 and T3C5. Note that all of these case are subjected to different levels of freestream turbulence intensities at the inlet, pressure gradients, and Reynolds numbers.

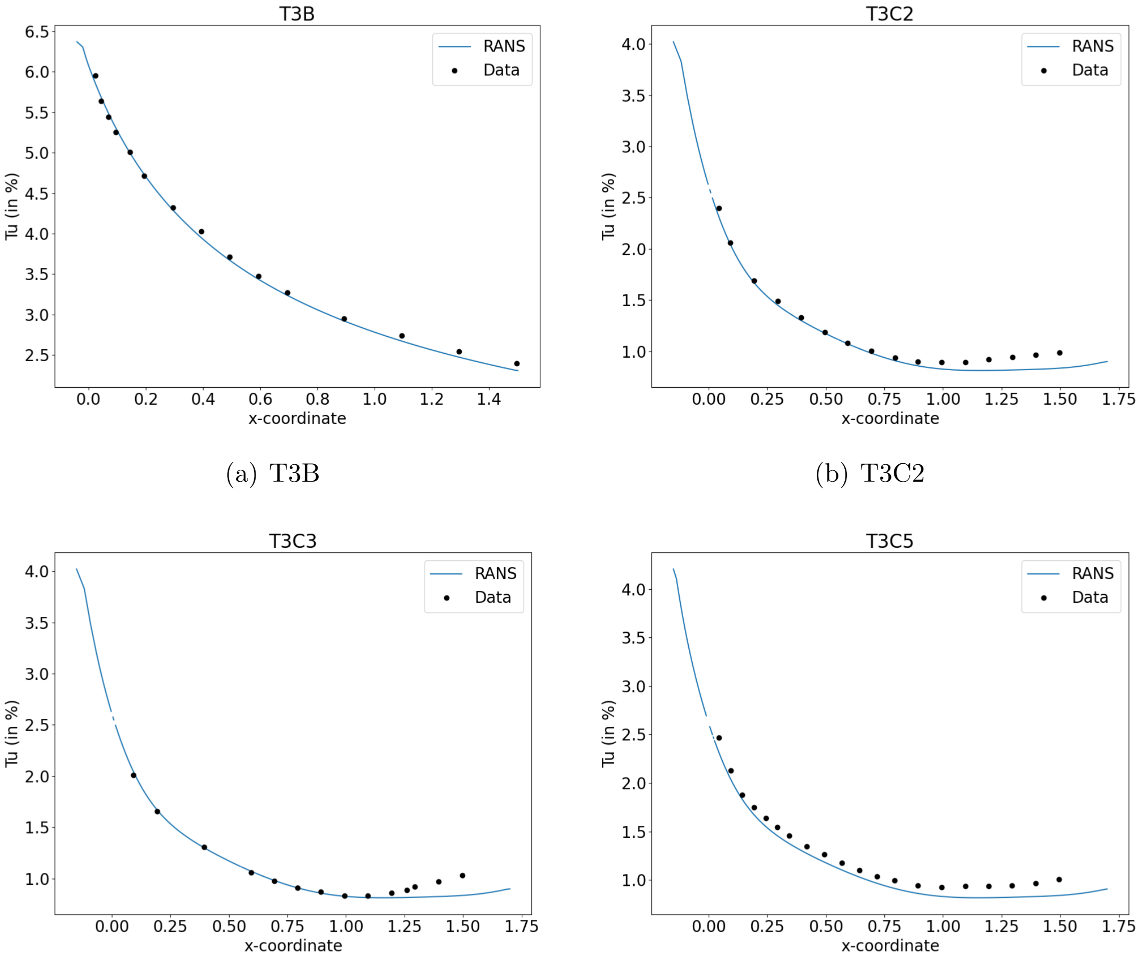
Table 4.2 shows the boundary conditions for all of the T3 cases used to test the inferred model.

Cases	**T3B**	**T3C2**	**T3C3**	**T3C5**
Tu_{in}	0.065	0.037	0.034	0.043
$\nu_{t_{\text{in}}}/\nu$	100.0	12.0	8.0	17.0
$L(\text{in } m)$	1.5	1.65	1.65	1.65
$Re_{L,\text{in}}$	940000	550000	418000	946000
$\omega_{\text{in}}(\text{in } s^{-1})$	7.943	11.4083	10.982	18.70738

Table 4.2: Inflow conditions for the T3 test cases

As seen in Fig. 4.13, the decay of freestream turbulence intensity is consistent with experimental data for all four cases, thus verifying the boundary conditions for ω. T3B is characterized by a zero pressure gradient flow. Transition occurs in a favorable pressure gradient region for T3C5 and in adverse pressure gradient regions for the T3C2 and T3C3 cases. The predictions for these cases are shown in Fig. 4.14. While the model successfully predicts the transition locations for T3B, T3C2, and T3C5 cases, it predicts transition locations which are significantly upstream for the T3C3 case. This is expected as the augmentation function was inferred using cases which exhibit transition within zero pressure gradient and favorable pressure gradient regions. In addition, this inability to predict transition for adverse pressure gradient cases might also signal to a deficiency in the surrogate chosen to predict $Re_{\theta,t}$.

To test the model on a geometry other than a flat plate, predictions for four single-stage high pressure turbine cascade cases from the VKI dataset are presented. The mesh used to perform the RANS simulations is shown in Fig. 4.15. The blade chord is 0.067 m in length and makes an angle of 55° with the flow direction. The inlet is located 0.055 m upstream of the leading edge while the outlet boundary is 0.242 m downstream of the leading edge. The mesh resolution next to the wall is on the order of $y^+ \approx 1$. The boundary conditions for the same are presented in Table 4.3. As can be seen from Fig. 4.16, the heat transfer coefficients plots show that transition locations are predicted



(a) T3B

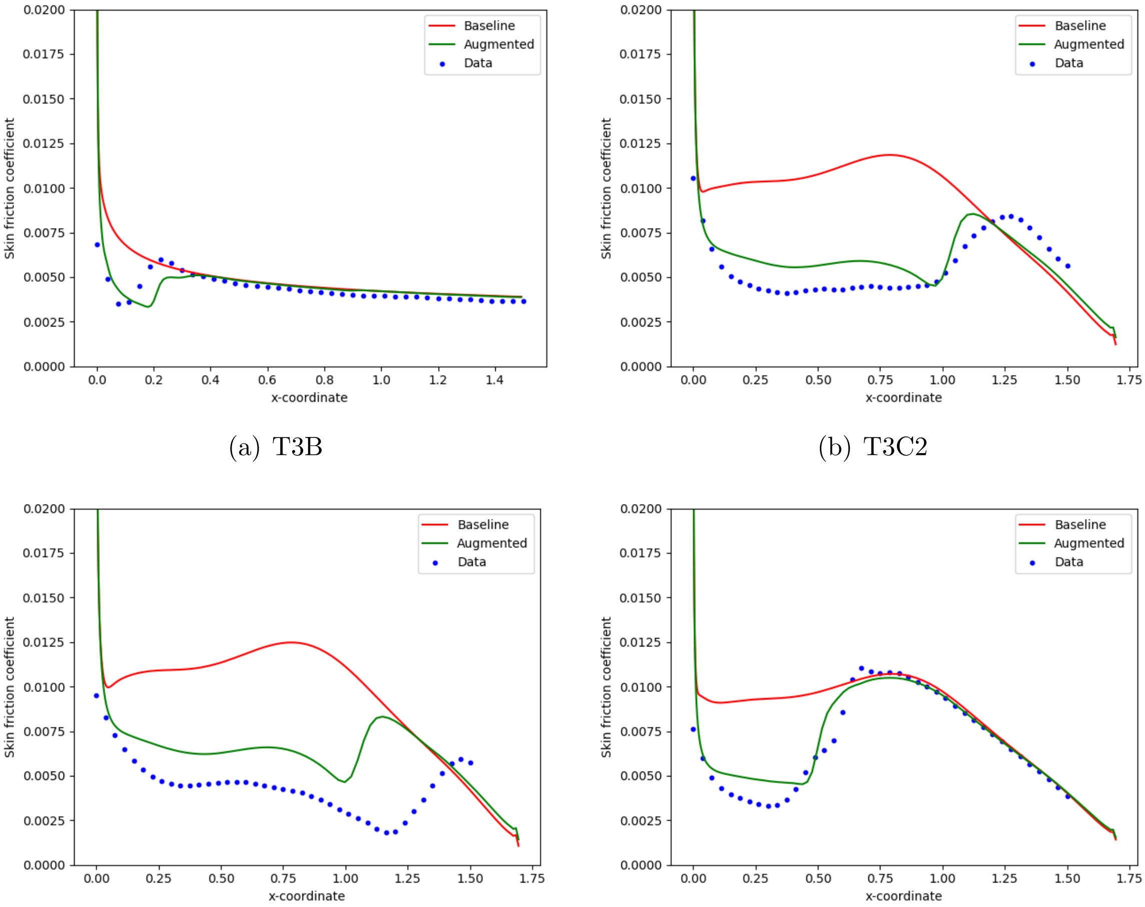
(b) T3C2

(c) T3C3

(d) T3C5

Figure 4.13: Comparisons between predictions and data for decay of freestream turbulence

well for the case MUR129 and MUR224. Looking closely at the data for the MUR241 case, while transition length is experimentally observed to be short for one of the surfaces (transition occurs between $x \approx 0.025$ and $x \approx 0.028$), transition seems to be quite gradual for the other surface. The corresponding predictions fail to predict this gradual transition. Note that while the predicted transition location is within this gradual transition range, it changes based on the wall distance value chosen for extraction of quantities at the edge of the boundary layer to estimate $\overline{Re_{\theta,t}}$. Lastly, observing the case MUR116, transition occurs significantly upstream of the actual transition locations on both the sides. To diagnose the issue with this case, this anomaly is juxtaposed with the sudden transition predicted for the MUR241 case. Looking at the contours of features 1 and 2, and the corresponding intermittency contours for both the cases near the predicted transition



(a) T3B

(b) T3C2

(c) T3C3

(d) T3C5

Figure 4.14: Skin friction coefficient profiles predicted using the augmentation inferred from the T3A and T3C1 cases

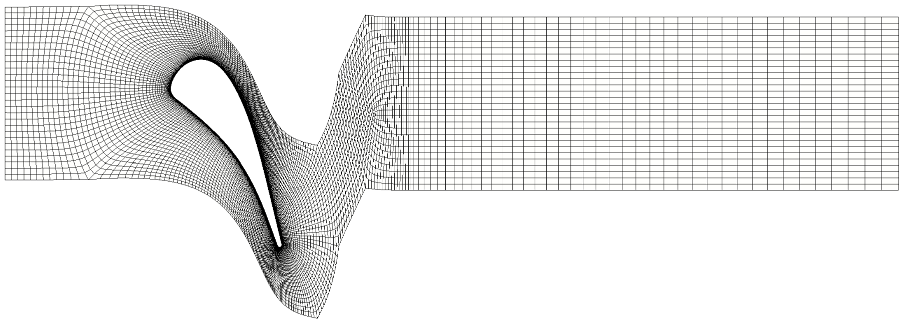
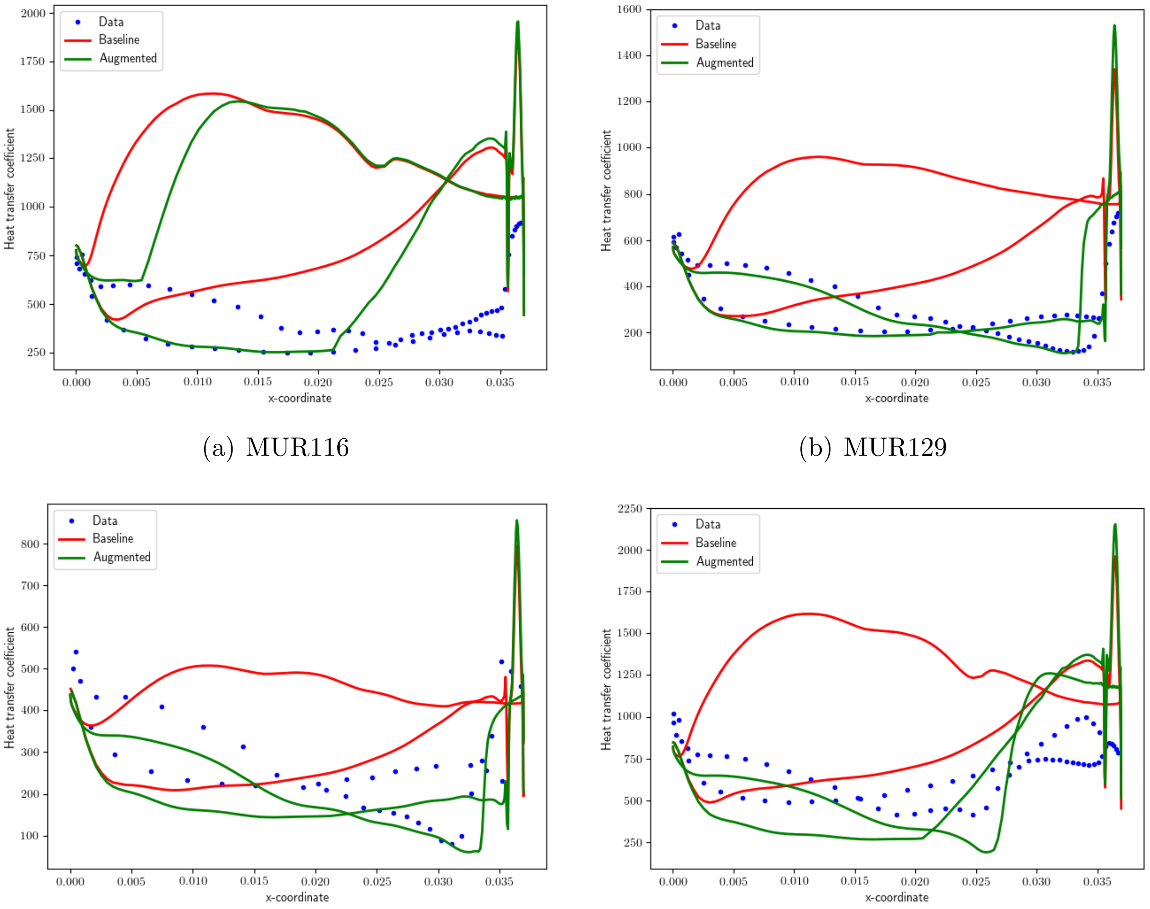


Figure 4.15: Mesh used to simulate VKI turbine cascade cases

locations as shown in Figs. 4.17 and 4.18, we can make the following observations. For the case MUR116, the intermittency quickly rises in a region where $0.1 \leq \eta_1 \geq 0.2$ and $\eta_3 \leq 0.8$. It seems that the available data was not sufficient for the augmentation to be inferred in this region of the feature-space. On the other hand, for the case MUR241, the sudden transition is in fact predicted within a region with relatively higher η_1 and η_3 in the

Cases	MUR116	MUR129	MUR224	MUR241
Tu_{in}	0.008	0.008	0.06	0.06
$\nu_{t_{\text{in}}}/\nu_{\text{in}}$	3	1.556	43.537	15.465
$p_{0,\text{in}}$(in bar)	3.269	1.849	0.909	3.257
$T_{0,\text{in}}$(in K)	418.9	409.2	402.6	416.4
p_{out}(in bar)	1.550	1.165	0.522	1.547
T_{wall}(in K)	300.0	300.0	300.0	300.0
ω_{in}(in s^{-1})	1.5×10^4	1.5×10^4	1.5×10^4	1.5×10^5

Table 4.3: Inflow, wall and outflow conditions for VKI test cases



(a) MUR116

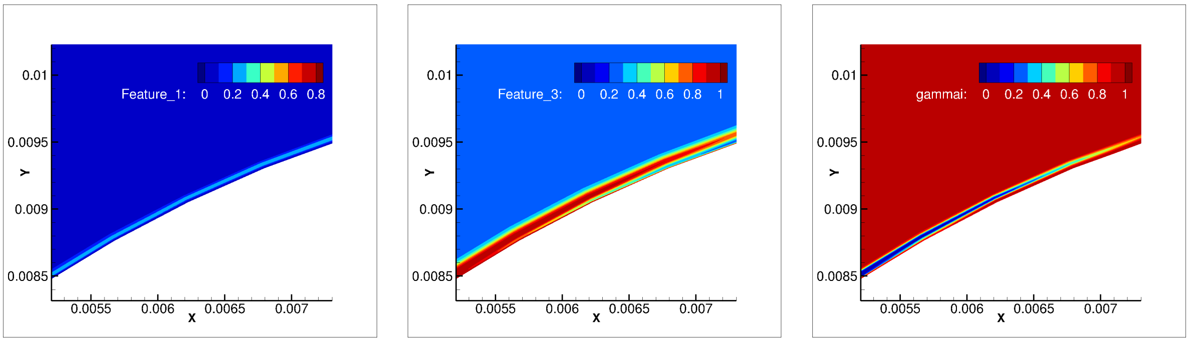
(b) MUR129

(c) MUR224

(d) MUR241

Figure 4.16: Heat transfer coefficient profiles predicted using the augmentation inferred from the T3A and T3C1 flat plate cases for the VKI turbine cascade cases

feature space where the augmentation was inferred indeed, however there could be several factors which might have contributed to prediction of a steep transition rather than a gradual one. These include imperfect features, insufficient resolution in the feature-space region in consideration and insufficient data to characterize the local behavior of the augmentation.

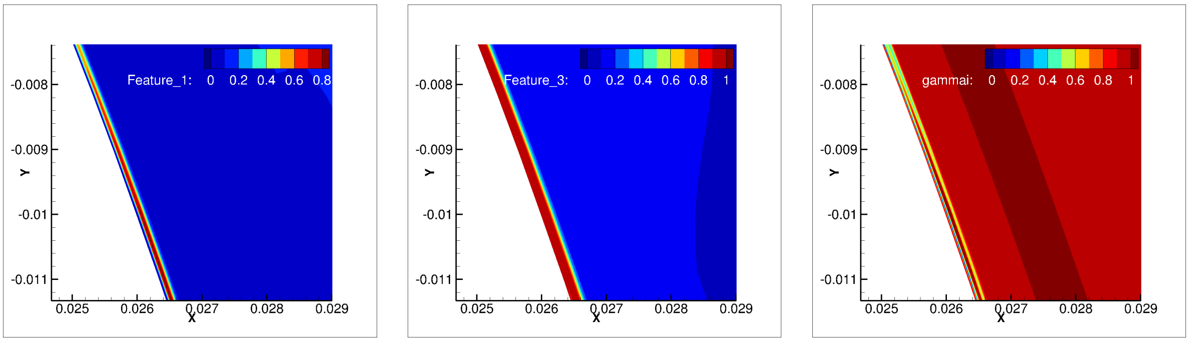


(a) Feature 1

(b) Feature 3

(c) Intermittency

Figure 4.17: Contours for MUR116 near the transition location



(a) Feature 1

(b) Feature 3

(c) Intermittency

Figure 4.18: Contours for MUR241 near the transition location

The effects of using a finer discretization in the feature-space, using a different user-specified distance to extract freestream quantities for use in calculation of η_1, or using only the T3A case for inference are illustrated in Appendix C. In summary, the supplementary results demonstrate that using a discretization in the feature-space that is too fine, while resulting in similar training accuracy as that obtained using the discretization shown here, could result in poorer generalizability. This observation also serves as motivation for future work to develop adaptive localized learning techniques to achieve a better balance between accuracy and generalizability. This is because a finer discretization with the same amount of available data results in a larger region of the feature-space remaining unaffected by the inference. Using predictions on VKI cases, it has been shown in Appendix C that small variations in the user-defined distance from where freestream quantities are extracted, in most cases, result in virtually similar transition locations. Moreover, using just the T3A case for inference could result in an inferior generalizability, as the use of multiple cases adds an implicit physical regularization to the inference problem and prevents the augmentation from overfitting to the needs of a single case.

4.3 Hierarchical Augmentation: Separation-induced Transition

The model developed in section 4.2 predicts the transition location well for most of the cases where transition occurs in the favorable and zero pressure gradient regions, however it predicts early transition within adverse pressure gradient regions of the flow as seen for the T3C3 flat plate case. To design a hierarchical augmentation, we consider transition prediction for compressor cascade geometries where flow separation is the predominant cause of bypass transition. LES data (provided by RTRC) are available for 6 single-stage configurations with the blade geometries belonging to the NACA65 family of airfoils. The meshes used for RANS simulations of all these cases are structured multiblock meshes. The full mesh for NACA65-010 is shown in Fig. 4.19. Fig. 4.20 shows the airfoil geometries used in all the six cases. The stagnation pressure, stagnation temperature, turbulent intensity and viscosity ratio for all the cases is shown in Table 4.4. The outlet back-pressures and flow angles are mentioned in Table 4.5. When applied to the compressor cascade cases, the augmented model obtained in Section 4.2 predicts transition near the location where flow separates, as can be observed from the comparisons between the predicted skin friction and the corresponding LES data. The

Quantities	$p_{0,\text{in}}$	$T_{0,\text{in}}$	Tu_{in}	$\nu_{t,\text{in}}/\nu_\infty$
Values	14.7705724 PSI	288.5672892 K	1 %	10

Table 4.4: Inflow conditions common for all compressor cascade cases

Cases	p_b (in PSI)	Flow angles (in degrees)
NACA65-010	14.695946	45
NACA65-410	14.715946	45
NACA65-1210	14.725946	45
NACA65-1810	14.725946	45
NACA65-1510	14.735946	60
NACA65-2110	14.735946	60

Table 4.5: Outlet back-pressure and flow angles for compressor cascade cases

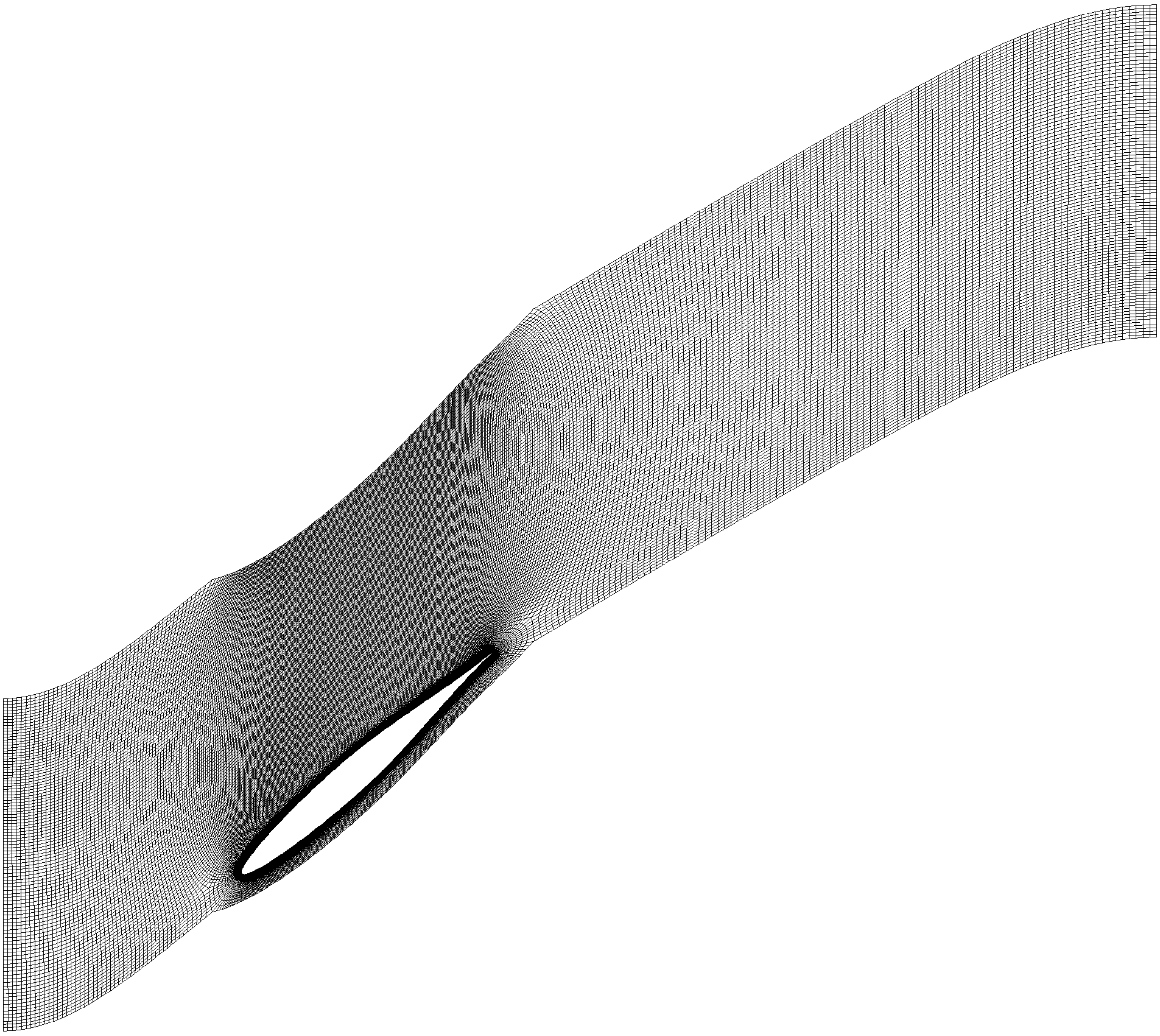


Figure 4.19: Mesh for the NACA65-010 single-stage compressor cascade case

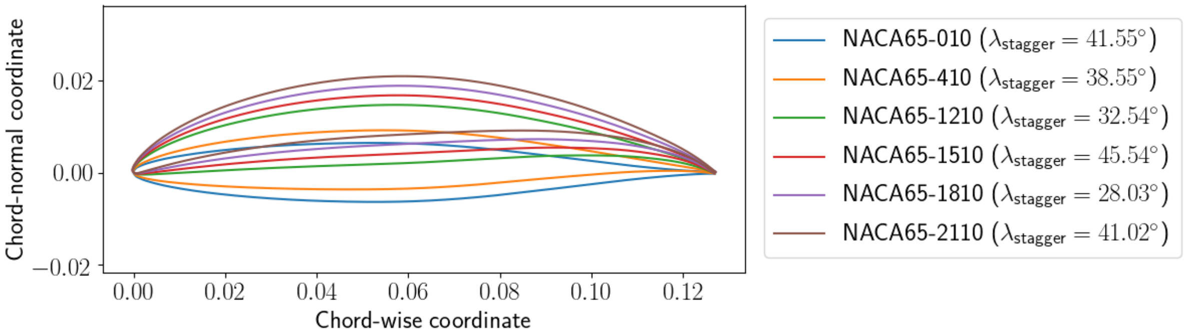


Figure 4.20: Blade Geometries used in the compressor cascade cases (along with respective blade stagger angles)

corresponding prediction on NACA65-010 is shown in Fig. 4.21. As discussed before, the baseline solution (shown in red) is fully turbulent at all locations.

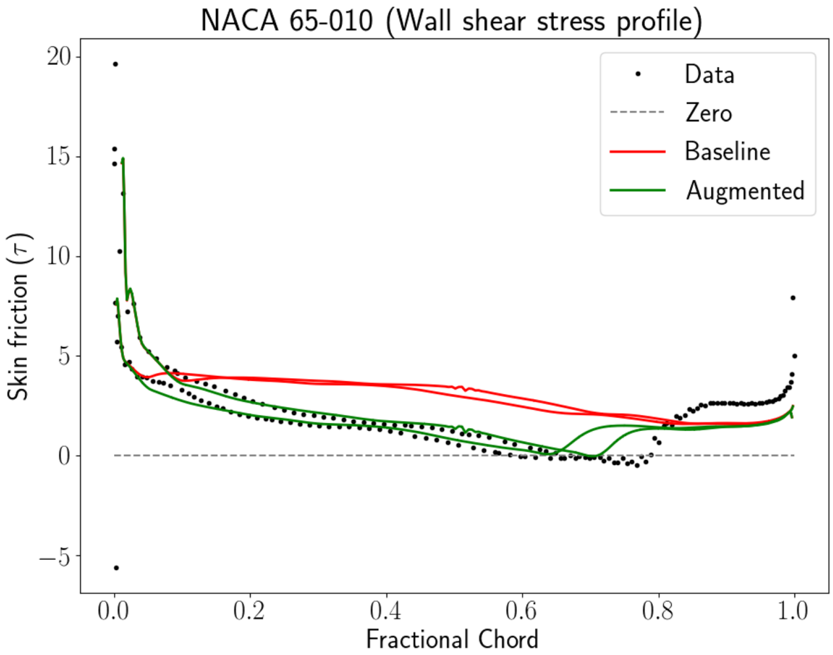


Figure 4.21: Using the augmented model from Section 4.2 to predict on the NACA65-010 compressor cascade geometry

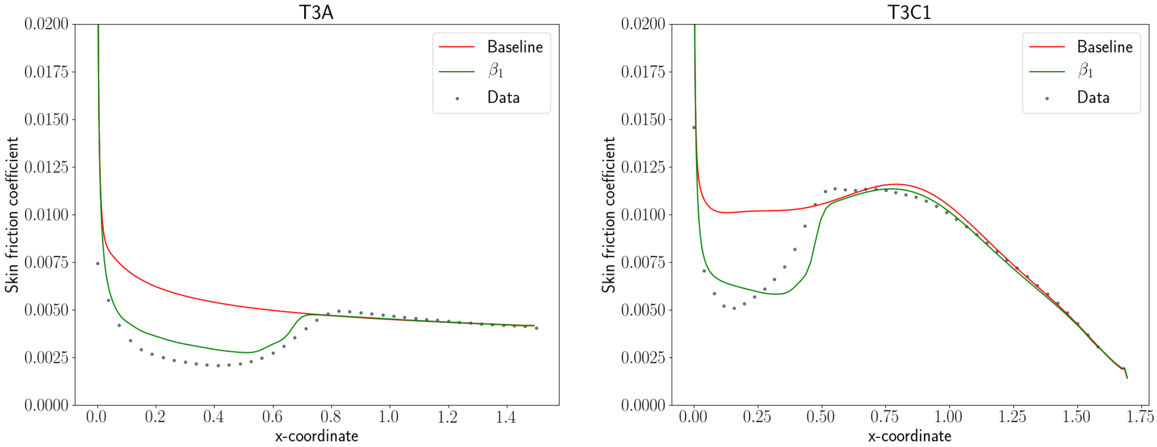
Attempts were made to simultaneously infer a single augmentation from the flat plate and compressor cascade cases which could predict well for both attached and separated flows. The first hurdle in the process was that for any inference strategy, the solver convergence would keep deteriorating with optimization iterations, which would eventually cause the adjoint solver to diverge and the optimization to stop. The cause for this behavior was found to be the discontinuous nature of the augmentation function. Since the functional form chosen for the augmentation in Section 4.2 makes use of Green-Gauss gradients within every cell in the feature-space, there could be jump discontinuities in the resulting augmentation along the cell boundaries. To avoid this, a C^0-continuous augmentation function was devised using multi-linear interpolation as shown in Section 4.3.1.

Once the issue with solver convergence was resolved, a single augmentation (with the same features as used before), when trained simultaneously using the T3 flat plate cases and compressor cascade cases, was unable to predict accurately for even the training cases. Since the physical mechanisms involved in bypass transition due to turbulent

fluctuations in the freestream are very different compared to those involved in separation-induced transition, it is not unexpected that a single augmentation would struggle to model both these phenomena. In order to differentiate between the two mechanisms, the introduction of an additional feature was attempted. However, the generalizability of the model was severely compromised in all such cases. This happened, probably, due to the overspecification of physical conditions which led to the augmentation differentiating between physical conditions requiring the same treatment from the augmentation.
In the end, the following hierarchical augmentation strategy was considered. The source term in the intermittency transport equation is modified to $(\beta_1\beta_2 - \gamma)\sqrt{\gamma}\Omega$ where β_2 is a new augmentation and is a function of the same features as β_1. Holding β_1 constant, the LIFE framework is used to infer optimal parameters for β_2 using LES data for the NACA65-010 case. A blending function σ is then designed. Ideally, σ should assume the value of unity for regions with flow separation (making β_2 fully active) and zero otherwise (making β_2 inactive). The source term in the intermittency transport equation is then modified yet again to incorporate σ as $(\beta_1\beta_2^{\sigma} - \gamma)\sqrt{\gamma}\Omega$.

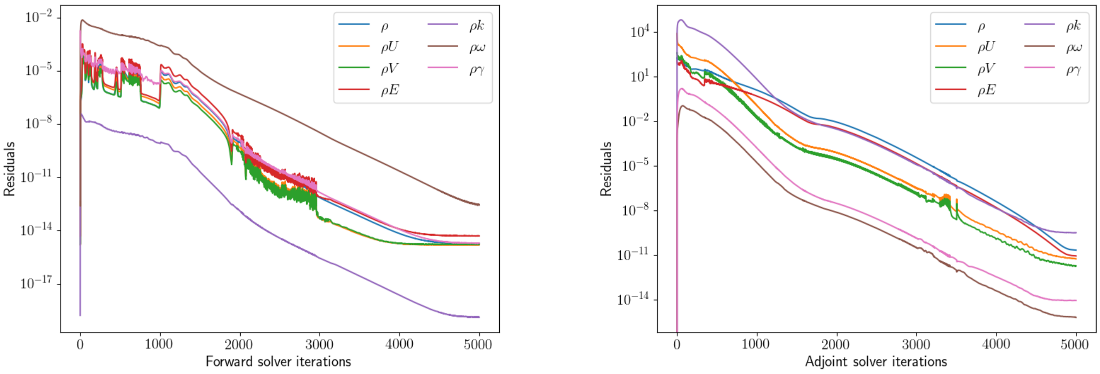
4.3.1 Improving Solver Convergence: A Continuous Functional Form

The discontinuous nature of the augmentation function was found to prevent the compressor cascade simulations from converging after a few inference iterations. To correct this, a slightly different interpolation-based functional form was chosen which could be continuous while supporting localized learning. To achieve this, the parameters of the augmentation were changed from cell-centered values to nodal values in the feature-space grid. The interpolation strategy was also changed from a linear interpolation based on Green-Gauss gradients to multi-linear interpolation. Note here that as the number of features increases, so does the size of the feature space and correspondingly the nodal values to be used for multi-linear interpolation within each grid cell grows exponentially. Since the augmentation calculation needs to take place for all spatial locations in the computational domain, it is important that the interpolation subroutine is written as optimally as possible to reduce computational time. Using an augmentation within the



(a) T3A (b) T3C1

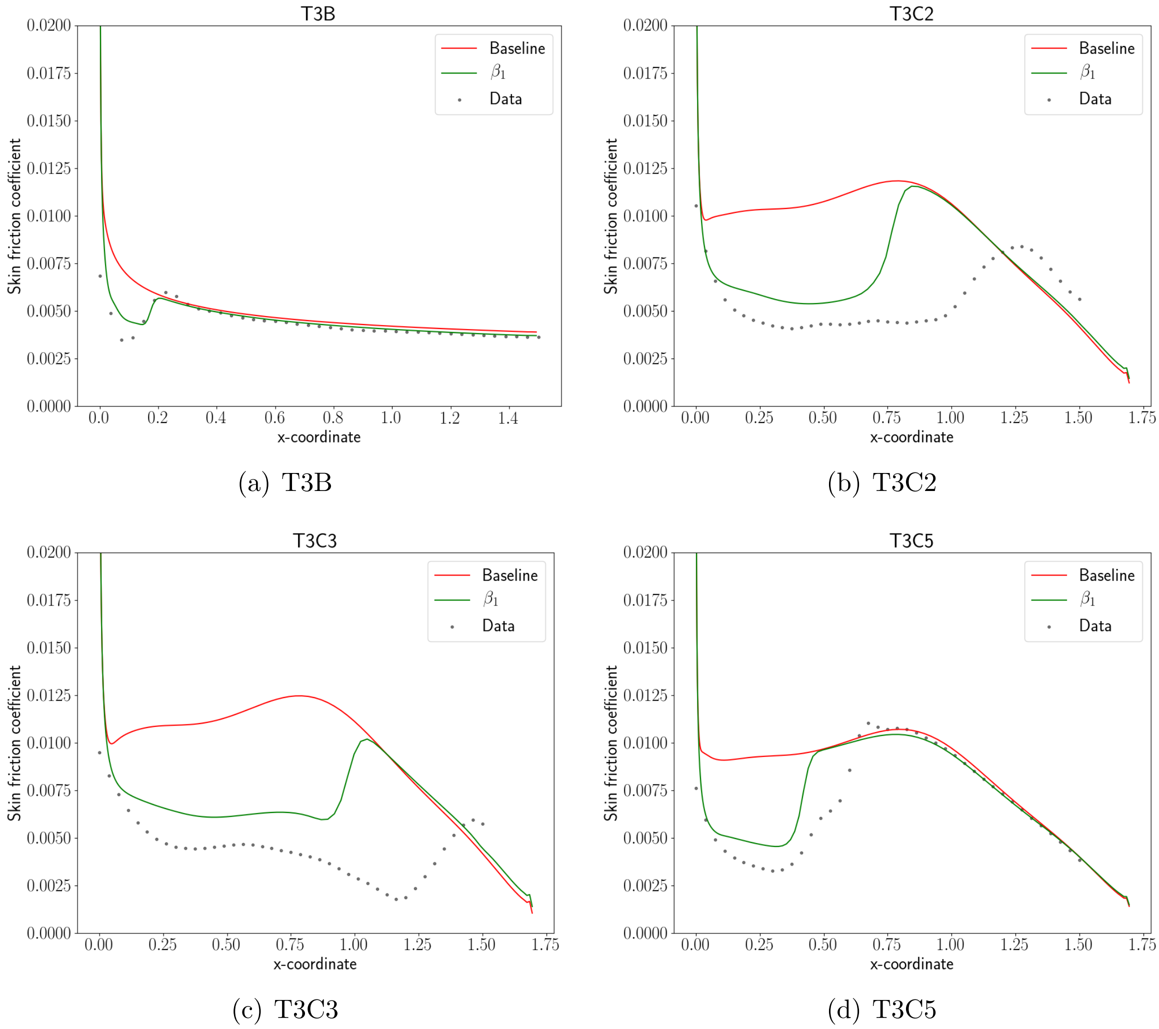
Figure 4.22: Skin friction coefficient distributions for the inference cases



(a) Forward (b) Adjoint

Figure 4.23: Residual minimization plots for the T3A case for the augmented model

3-dimensional feature-space designed in section 4.2.2, the results for inference from the T3A and T3C1 cases are shown in Fig. 4.22. The feature-space was discretized into 45, 15, and 15 cells (46 × 16 × 16 nodes) along the three feature directions and an step size of $\dfrac{0.1}{\left\|\dfrac{\partial \mathcal{J}}{\partial \boldsymbol{w}}\right\|_\infty}$ was used for the steepest gradient descent algorithm. The inferred results exhibit better laminarization in the pretransitional regions of the boundary layer compared to the discontinuous augmentation function inferred in section 4.2 (especially for the T3A case). The solver convergence plots for the augmented model are shown in Fig. 4.23 which show considerably better values of converged residuals when compared to that obtained using the Green-Gauss based interpolation method (Fig. 4.9). The corresponding predictions for the rest of the T3 cases cases using the augmented model are shown in Fig. 4.24. As can be observed from these plots, while the transition location is predicted



(a) T3B

(b) T3C2

(c) T3C3

(d) T3C5

Figure 4.24: Skin friction coefficient predictions for T3 cases using the inferred continuous augmentation

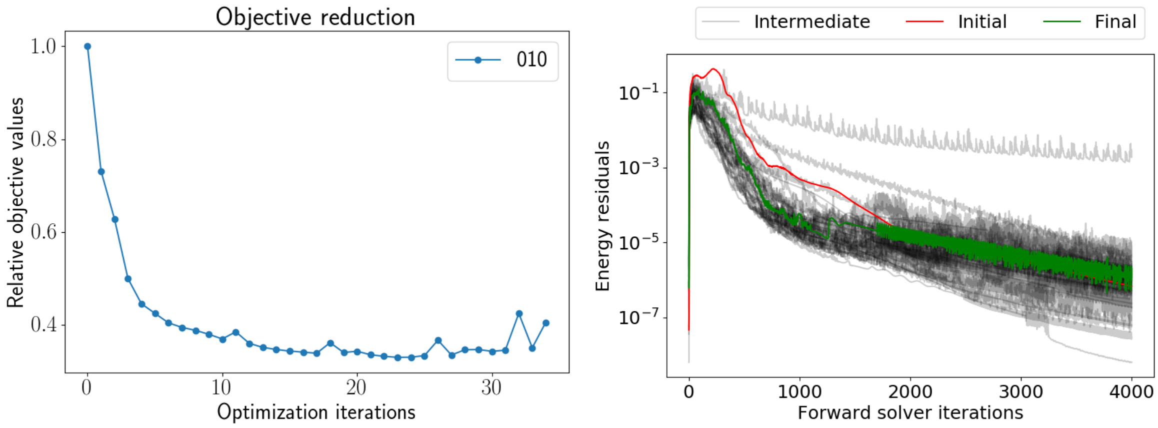
well for the T3B and T3C5 cases, transition is predicted significantly upstream compared to the experimentally observed location for cases T3C2 and T3C3. While this behavior can be attributed to the model not being exposed to transition occurring in regions of adverse pressure gradient, these predictions are even more inaccurate than the ones predicted with the previously shown discontinuous augmentation. When using a discontinuous functional form the intermittency can jump significantly across the cell interfaces in the feature-space. However, when using the aforementioned continuous functional form it has to do so gradually along the length of the cell and hence it is impossible to represent sharp gradients in the feature-space with a coarse discretization which causes the relatively high inaccuracies when compared to discontinuous augmentations. A potential solution for this problem is to create a finer discretization within the feature-space, but that would increase the amount of data required to infer a generalizable augmentation. Since the objective, here, is to demonstrate hierarchical augmentation, such investigations are topics for future work.

4.3.2 Inferring β_2 from the NACA 65-010 geometry

The sum squared discrepancy between the LES data and predictions for the wall shear stress was chosen as the cost function as shown in Eqn. 4.22.

$$\mathcal{C} = \|\tau_{\text{wall}}^{\text{pred}} - \tau_{\text{wall}}^{\text{data}}\|_2^2 \tag{4.22}$$

Again, no regularization is used for the same reasons as mentioned in section 4.2. The mesh used to perform the RANS simulations on this geometry is shown in Fig. 4.19. The step size for the steepest gradient descent was chosen to be $\dfrac{0.1}{\left\|\dfrac{\partial \mathcal{J}}{\partial \boldsymbol{w}}\right\|_\infty}$ and the feature-space for β_2 was discretized into 30, 10 and 10 cells along the η_1, η_2, and η_3 directions, respectively. The plots for the optimization history along with residual convergence are shown in Fig. 4.25. The residual convergence for the baseline model is shown in red and that for the most optimal iterate (iteration 27) is shown in green. Fig. 4.26 shows the comparison between the baseline model augmented only with β_1 and the hierarchically augmented model. The hierarchically augmented model predicts the transition location very accurately compared to the predictions when β_1 is used on its own. However, the wall shear stress is under-predicted in the fully turbulent region. Similar to the turbine cascade cases, this discrepancy is attributed to the turbulence model. Since the under-prediction of the wall shear stress was not the inadequacy in consideration, this discrepancy should be ignored while evaluating the capability of the hierarchical augmentation. It should be noted that since the augmentation function outputs are limited between the physical values of 0 and 1 and since the intermittency field closely follows the augmentation field, the intermittency cannot amplify the production term in the k-transport equation to compensate for this inadequacy.



(a) Objective Minimization

(b) Residual Convergence

Figure 4.25: Objective minimization and residual convergence while inferring β_2 from the NACA 65-010 case

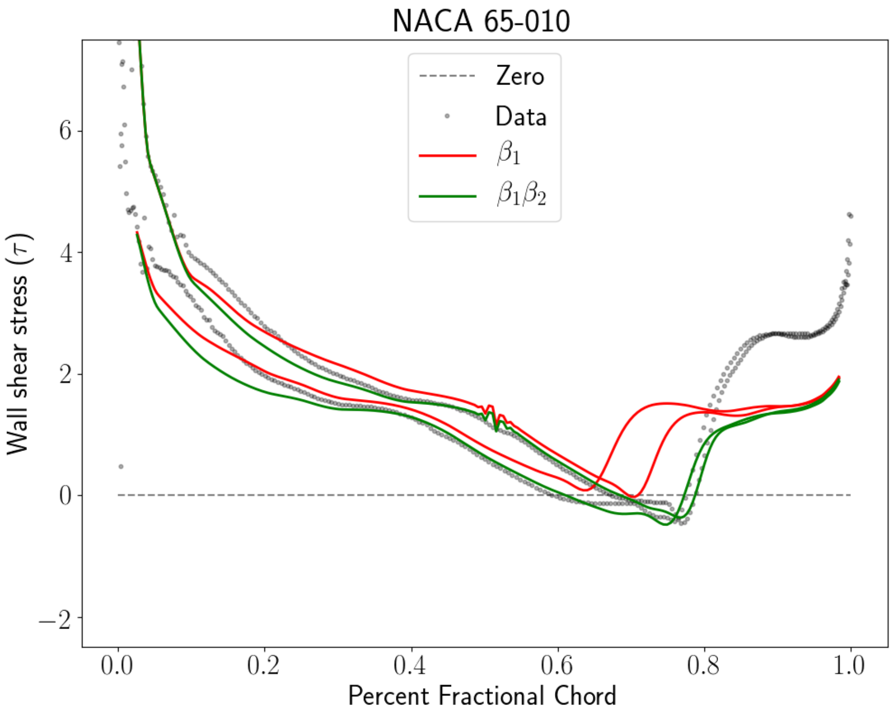


Figure 4.26: Objective minimization for inference and learning from NACA 65-010 case

4.3.3 Formulating a blending function

To create a blending function σ that varies between zero and unity, one or more sigmoid functions can be applied to an appropriate quantity f_σ. One of the first candidates chosen for f_σ was $d_w\Omega/(d_w\Omega + U)$, with the rationale that this quantity would always remain below 0.5 for boundary layer velocity profiles that do not exhibit an inflection point. Since inflection points are observed when the flow is subjected to high adverse pressure gradients and also when the flow undergoes separation, this candidate can potentially detect regions characterized by adverse pressure gradients and flow separation. However, since this quantity involves the velocity magnitude U, a different choice was explored to reduce the dependency of the model on quantities which do not follow Galilean invariance. Ge [22] used the function $\omega d_w(\boldsymbol{n_w} \cdot \boldsymbol{\nabla})|\boldsymbol{S}|/\sqrt{2}|\boldsymbol{S}|^2$ to differentiate the separated regions of the flow. Here $\boldsymbol{n_w}$ is the wall-normal direction corresponding to the nearest point on the wall. This can be calculated by evaluating the gradient of the wall distance (d_w) and normalizing the result (to ensure that the magnitude is unity). Note that for body-fitted grids with low cell skewness, this approximation is fairly accurate close to the walls and that is exactly where the blending function is needed. $|\boldsymbol{S}|$ denotes the magnitude of the strain rate tensor. Some changes made to this function in order to be effectively used as a candidate for f_σ are mentioned as follows:

- The vorticity magnitude $\Omega = |\boldsymbol{\Omega}|$ was used instead of $|\boldsymbol{S}|$

- The laminar length scale $\ell = \sqrt{\nu/\Omega}$ is used instead of d_w as it produced better results. Note here that ℓ calculated at the wall corresponds to a length equal to $\Delta y^+ = 1$. Since the vorticity reduces away from the wall for velocity profiles without inflection points, ℓ would correspondingly increase away from the wall. For attached flows under adverse pressure gradients ℓ decreases for some distance away from the wall and then starts increasing. For separated flows, this quantity would increase to a very high value as one moves away from the wall until a location with zero vorticity is reached. Beyond this location, ℓ would first decrease to a minimum value and then keep on increasing with wall distance.

- The functional form was bounded in order to restrict the variation of f_σ between -1 and 1. Note that bounding f_σ is not necessary, but doing so allows for its use as an additional feature in future work.

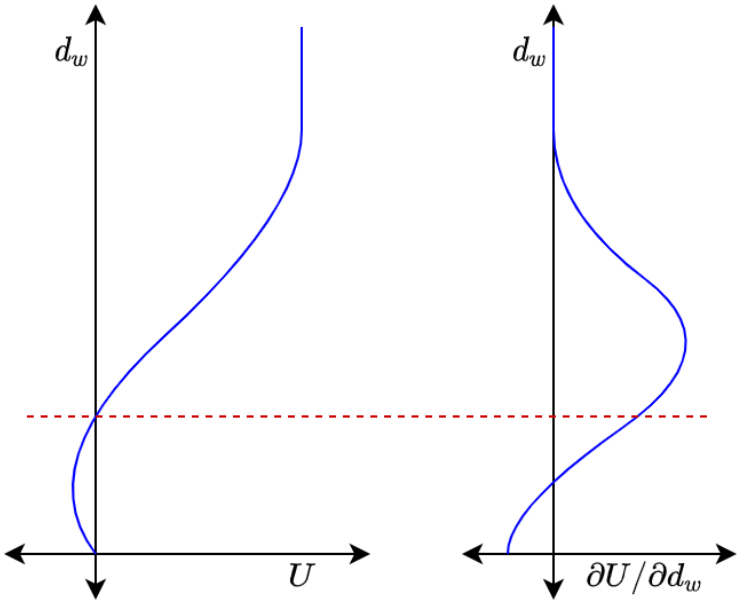


Figure 4.27: Schematic depicting velocity and its wall normal derivative (signed vorticity) in a separated flow

The final functional form of the chosen f_σ is shown in Eqn. 4.23.

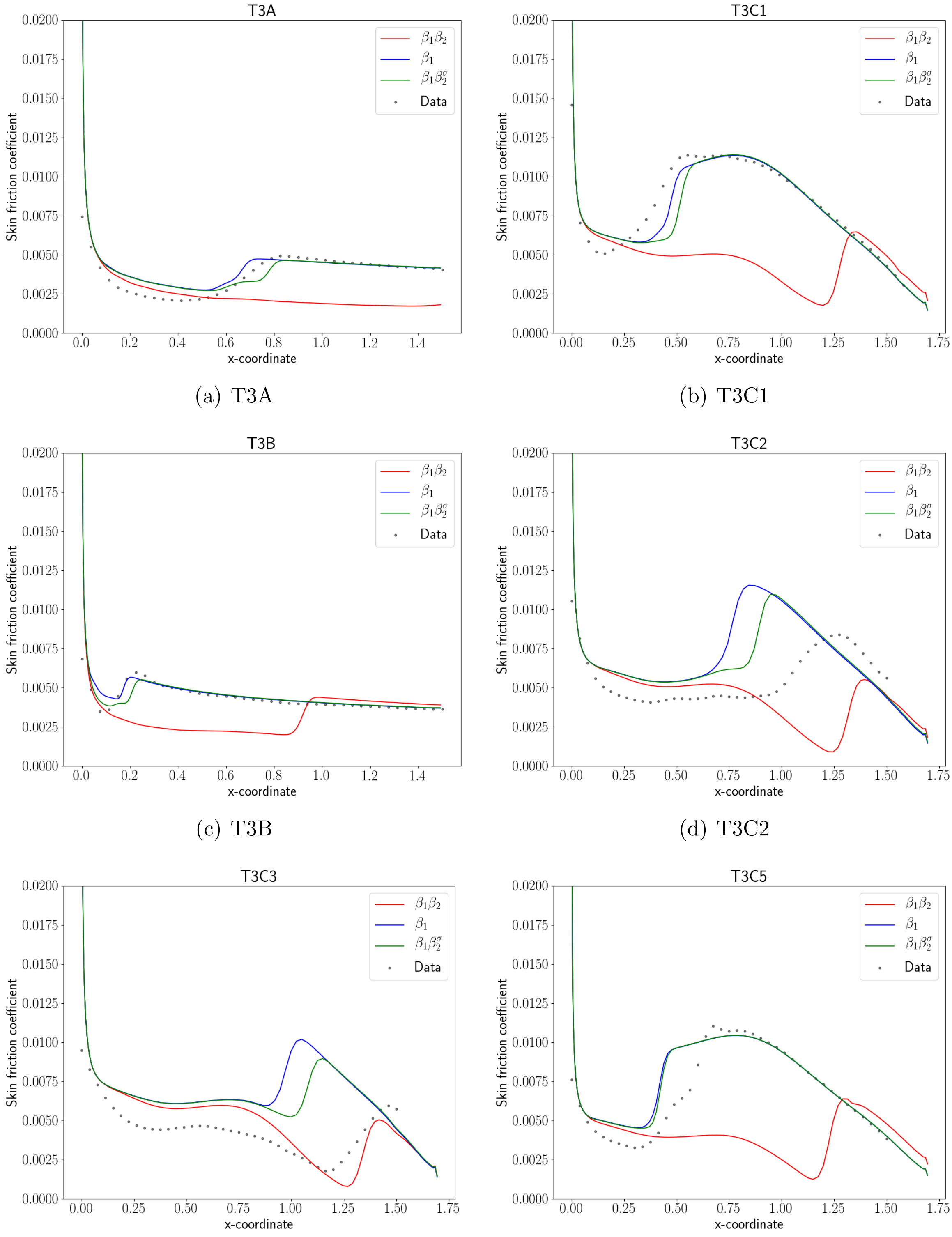
$$f_\sigma = \left(\frac{\sqrt{\nu}(\boldsymbol{n_w}\cdot\boldsymbol{\nabla})\Omega}{\sqrt{\nu}|(\boldsymbol{n_w}\cdot\boldsymbol{\nabla})\Omega| + \Omega^{1.5}}\right)\left(\frac{\omega}{\sqrt{2}\Omega + \omega}\right) \tag{4.23}$$

The following analysis explains why this function can help in differentiating separated flows from attached flows. For velocity profiles with no inflection, the wall normal derivative of the vorticity magnitude (and hence f_σ) is always non-positive. Within the laminar regions of the flow, the production term in the ω transport equation is nearly zero. Thus, ω decays rapidly (faster than even an exponential) with d_w. Next to the wall, ω is very high compared to Ω owing to the boundary condition for ω for smooth walls. Hence, $\omega/(\sqrt{2}\Omega + \omega)$ is very close to 1 next to the wall and decreases with wall distance to a point and then starts increasing again (as ω in the freestream is some non-zero value but Ω is close to zero). For attached flows under adverse pressure gradients, the vorticity increases up to the point of inflection resulting in comparatively lower values of $\omega/(\sqrt{2}\Omega+\omega)$ near the point of inflection. As can be seen in the schematic shown in Fig. 4.27, it is clear that there exists a region around the edge of the separation bubble where the vorticity magnitude increases with wall distance, i.e., where f_σ is positive. β_2 needs to be applied within this region. However, the threshold needs to be set to a slightly negative value to ensure that β_2 is enabled in a region thick enough to lower the intermittency values sufficiently in order to laminarize the boundary layer. Also note that, around the upper boundary of this positive region where $f_\sigma \approx 0$, i.e., where Ω reaches its maximum value, the values of Re_Ω (and hence η_1) might be high enough to make β_1 predict high intermittency values which makes it even more important to make sure that β_2 keeps the intermittency suppressed in this region as much as possible, thus necessitating a slightly negative threshold. A side effect of this negative threshold is that β_2 can start affecting very thin regions in attached flows. This effect is more pronounced for regions under adverse pressure gradients. Finally, for locations where f_σ is negative within the separation bubble, the value of Ω and d_w is sufficiently small that η_1 would not cause β_1 to ramp the intermittency up. When practically applied, some trial and error is required to obtain an optimal value for this negative threshold such that separation-induced transition is predicted well without compromising the accuracy of the attached flows by a significant amount. For this problem, a thresholding value of -0.05 is chosen, below which β_2 is inactive. The final functional form of σ is given in Eqn. 4.24.

$$\sigma = \frac{1}{1 + \exp(-(f_\sigma - 0.05)/0.003)} \tag{4.24}$$

4.3.4 Predictions using a blended hierarchical augmentation

Using the blending function σ designed in section 4.3.3, the predictions for cases in the T3 dataset and other cases in the RTRC dataset are shown in Figs. 4.28 and 4.29 respectively. As can be seen, the negative threshold implemented within the blending function slightly delays the transition location for cases where transition takes place within regions of favorable pressure gradients (T3A, T3B, T3C1, T3C5) with significant delays observed for attached flows where it takes place under adverse pressure gradients (T3C2 and T3C3) which improves the predictions in these cases. Looking closely at predictions for unseen compressor cascade cases (which exhibit separated flow) in Fig. 4.29, a few observations can be made.



(a) T3A

(b) T3C1

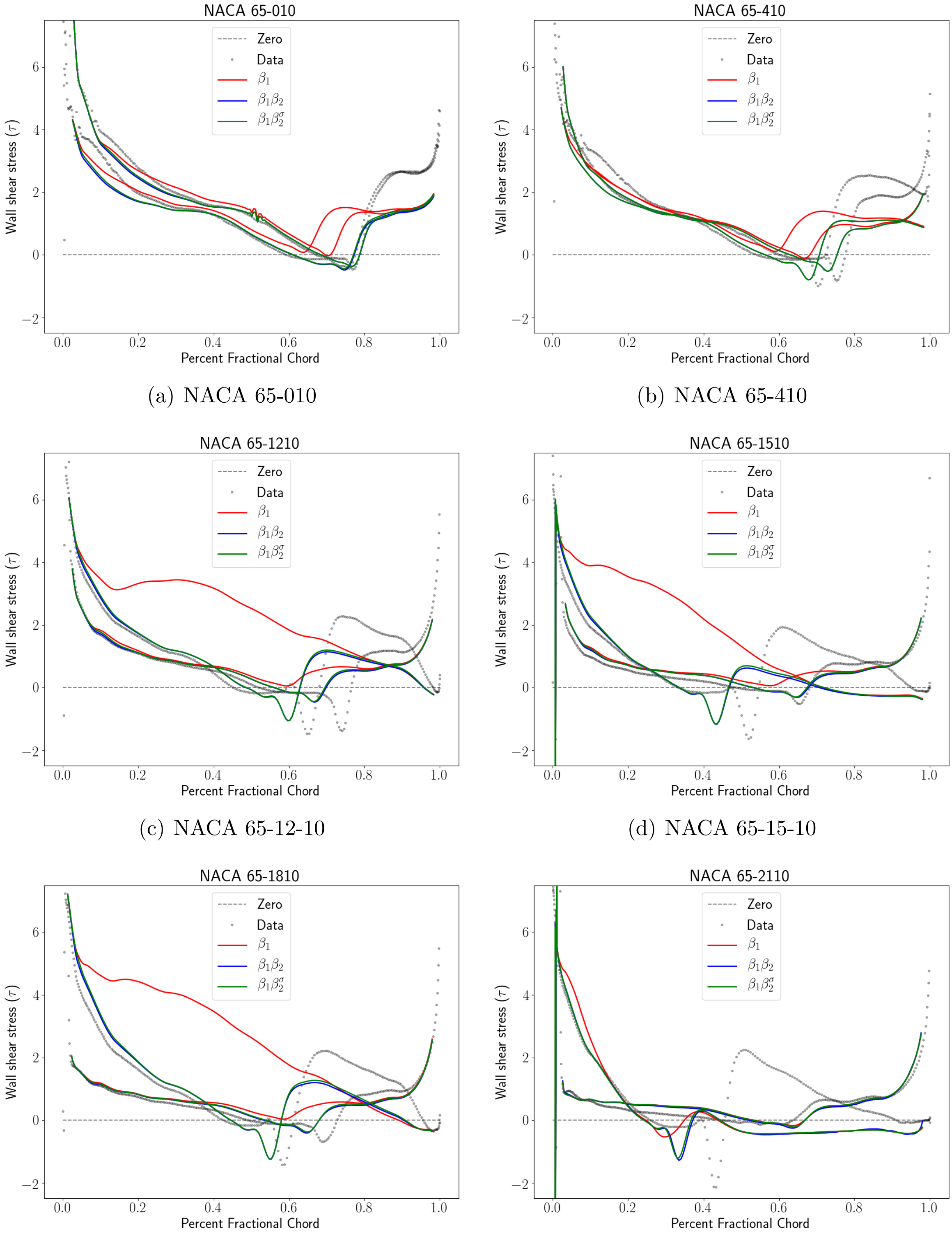
(c) T3B

(d) T3C2

(e) T3C3

(f) T3C5

Figure 4.28: Skin friction predictions for T3 flat cases using the hierarchical augmentation with and without the blending function



(a) NACA 65-010

(b) NACA 65-410

(c) NACA 65-12-10

(d) NACA 65-15-10

(e) NACA 65-18-10

(f) NACA 65-12-10

Figure 4.29: Wall shear stress predictions for RTRC cases using hierarchical augmentation with and without the blending function

- The transition location is predicted with large errors, when the model is augmented with β_1 alone.

- Using β_2 consistently improves predictive accuracy across all cases. However, the predicted transition location may still be slightly upstream when compared with data. The results may be improved by using better features and/or using a more optimal discretization in the feature space. These are subjects for future work.

- Finally, the blending function has a very small effect on the predictions made using β_2, which demonstrates that the blending function works well.

4.4 Remaining Challenges

While the LIFE framework did enable inference of generalizable transition models, a number of challenges remain to be addressed. Firstly, as discussed in the previous sections, while a discontinuous functional form with the same discretization allows the augmentation function to have sharper gradients in the feature-space and hence improves predictive accuracy on unseen cases, a continuous functional form enables superior solver convergence. While this thesis applies interpolation-based strategies for localized learning, other strategies like training neural networks with artificial datapoints also need to be tested. Such strategies might help in resolving the augmentation behavior in the feature-space while minimally affecting stability of the numerical solver. Secondly, the features used in this thesis need to be improved and modified such that the augmentation can discern different physical conditions more effectively. The use of freestream quantities is undesirable and makes for a rather ambiguous implementation and hence correlations involving local quantities must be developed for use in feature design. Finally, there is a need for uncertainty quantification strategies that can leverage the fact that some regions of the feature-space are not accessed at all during the inference process and that predictions involving such regions carry high epistemic uncertainties. Such uncertainty quantification methods, when combined with existing uncertainty quantification and propagation methods, can result in a powerful engineering design tool for practical aerodynamic applications.

Chapter 5

Weakly-Coupled Integrated Inference and Machine Learning

While the LIFE framework presented and demonstrated in the previous chapters helps in creating generalizable augmentations, designing a good feature-space can be a long and difficult process. However, several industrial applications do not require full generalizability and rather focus on only a limited range of physical configurations. In such cases, feature design is not as important as creating a usable augmentation that improves the predictive accuracy of an existing model for such limited range of applications. Another practical problem of interest is that the solver codes are intricate and might involve complex datatypes and libraries which could make the embedding of an augmentation within the model difficult and time-consuming. Such scenarios demand for an easy-to-use methodology which can bypass these hurdles and accelerate development of viable augmentations while still retaining desirable characteristics of integrated inference and machine learning. In this chapter, a novel weakly-coupled IIML methodology is presented which is subsequently demonstrated by augmenting a reduced-fidelity polymer electrolyte membrane fuel cell (PEMFC) model. A non-intrusive iterative method to solve augmented model equations is used to bypass the requirement of embedding the augmentation (or its linearized form) within the solver code. Note here that the approach used in this chapter does not used localized learning. Instead conventional neural networks are used as functional forms for the augmentation functions.

5.1 A Non-intrusive Iterative Method to Solve Augmented Model Equations

In order to perform strongly-coupled integrated inference and machine learning, the augmentation function has to be embedded into the solver for both evaluation of the objective function and its sensitivities to the parameters $\boldsymbol{w}$. However, embedding the augmentation involves significant changes to the solver code, thus requiring considerable effort. When testing several augmentation candidates and/or working with an intricate solver, being able to work with an augmentation function that does not need to be implemented within the numerical solver can save time, effort and resources while allowing increased flexibility, ease-of-use and portability.

Assuming that an augmentation function $\beta(\boldsymbol{\eta};\boldsymbol{w})$ is given, we need to solve the model as described in eqn. 5.1.

$$\mathscr{R}(\widetilde{\boldsymbol{u}}_m;\delta(\boldsymbol{x}),\boldsymbol{\xi})=0 \quad s.t. \quad \delta(\boldsymbol{x})=\beta(\boldsymbol{\eta}(\widetilde{\boldsymbol{u}}_m;\boldsymbol{\xi});\boldsymbol{w}) \tag{5.1}$$

To do this without embedding the augmentation function $\beta(\boldsymbol{\eta})$ into the solver, one can solve the augmented model in an iterative manner as shown in eqn. 5.2.

$$\mathscr{R}(\widetilde{\boldsymbol{u}}_{m,i+1};\delta_i(\boldsymbol{x}),\boldsymbol{\xi})=0 \quad s.t. \quad \delta_i(\boldsymbol{x})=\rho\delta_{i-1}(\boldsymbol{x})+(1-\rho)\beta(\boldsymbol{\eta}(\widetilde{\boldsymbol{u}}_{m,i};\xi);\boldsymbol{w}) \tag{5.2}$$

Here, ρ is a relaxation factor to avoid stability issues in the numerical solver. In the current work, ρ is chosen as 0.3 by trial and error to allow the iterative solver to converge for all training cases. $\delta^{(0)}(\boldsymbol{x})$ can assume a constant value of 0 or 1 throughout the domain depending on whether the augmentation term is additive or multiplicative, respectively. An augmentation residual can be defined as shown in eqn. 5.3.

$$R_{\text{aug}}=\|\delta_i(\boldsymbol{x})-\delta_{i-1}(\boldsymbol{x})\|_2 \tag{5.3}$$

A stopping criterion of $R_{\text{aug}}<10^{-3}$ was found to be enough for the simulations performed in this work to achieve reasonably converged field solutions. While convergence and stability are not guaranteed, an overwhelming number of the configurations tested in this work converged, while the remaining exhibited an oscillatory behavior in the augmentation residual. It is noteworthy here, that since the augmentation field changes in increasingly smaller amounts from one augmentation iteration to the next (given a well-chosen value of the relaxation factor ρ), the computational cost required for the solver to converge keeps decreasing as iterations progress. Hence, while the computational cost to solve the augmented model is significantly greater than that required to solve the baseline model, it does not exactly scale with the number of augmentation iterates for a well-chosen value of rho. Thus, carefully choosing the convergence criterion can be instrumental in significantly reducing the computational costs of solving a model with the aforementioned non-intrusive iterative solution method.

5.2 Weakly-coupled Integrated Inference and Machine Learning

This version of IIML constrains the inadequacy field to stay consistent with the functional form chosen for the augmentation by solving the field inversion and machine learning problems in a predictor-corrector fashion. Here, the weakly-coupled IIML framework is described in detail which can simultaneously infer from multiple data sources.

This is done by learning the augmentation each time the inadequacy field is updated, i.e., after every iteration of field inversion. Note that while the inadequacy fields are updated independently for all training cases, the machine learning step acts a synchronizing step for these individual optimization problems. Data from the inadequacy fields ($\delta^i(\boldsymbol{x})$) and corresponding feature fields ($\boldsymbol{\eta}^i(\boldsymbol{x})$) is collated from all training cases and a sufficient number of machine learning iterations (epochs) are performed to ensure that the feature-to-augmentation map learns any new information from the updated flow fields. After the machine learning step, a “field correction” is performed by solving the model again with the newly learned augmentation function ($\beta(\boldsymbol{\eta}(\widetilde{\boldsymbol{u}}_m^i, \boldsymbol{\zeta}^i); \boldsymbol{w})$). When the simulation converges, the predicted augmentation field ($\beta^i(\boldsymbol{x}) = \beta(\boldsymbol{\eta}(\widetilde{\boldsymbol{u}}_m^{(i,\beta)}, \boldsymbol{\zeta}^i); \boldsymbol{w})$) is used as the input to the field inversion for the next inference iteration. The superscript (i, β) denotes that the velocity field corresponds to the i^{th} training case and is obtained by solving the model with the augmentation function. Solving the model again with the updated augmentation function is crucial to ensure that the model predictions are consistent with the augmentation function throughout the inference process. Finally, the sensitivity $\dfrac{d\mathcal{J}}{d\beta(\boldsymbol{x})}$ is calculated and the inadequacy field is updated using a steepest gradient method, similar to a field inversion iteration. The step length α^i needs to be set manually. In this work, it was set to $\dfrac{0.05}{\left\|\dfrac{d\mathcal{J}^i}{d\beta^i(\boldsymbol{x})}\right\|_\infty}$. In summary, the following three consistencies are ensured when using the weakly-coupled IIML described above.

1. Formulating the objective as a function of model predictions ensures that the inadequacy field iterates $\beta^i(\boldsymbol{x})$ are model-consistent.

2. Machine learning ensures that inadequacy field iterates $\beta^i(\boldsymbol{x})$ are always consistent with the functional form of the augmentation function across all iterations.

3. Field correction ensures that the augmentation field iterates $\beta^i(\boldsymbol{x})$ correspond to the converged solution obtained by solving the augmented model characterized by the updated augmentation function parameters $\boldsymbol{w}$.

A flowchart describing this process is shown in Fig. 5.1.

It should be noted here that the optimization trajectory for weakly-coupled IIML could be significantly different from that for its strongly-coupled counterpart. The reason for this is explained as follows. For the i^{th} training case, the computational domain for which consists of N_x^i discrete spatial location, the discretized inadequacy field can be represented in an $\mathbb{R}^{N_x^i}$. Now, the set Δ^i of all inadequacy fields $\delta^i(\boldsymbol{x})$ for which there exist some set of parameters $\boldsymbol{w}$ such that $\delta^i(\boldsymbol{x}) = \beta(\boldsymbol{\eta}(\widetilde{\boldsymbol{u}}_m^i, \boldsymbol{\zeta}^i); w)$ and $\mathscr{R}_m(\widetilde{\boldsymbol{u}}_m^i; \delta^i(x), \boldsymbol{\xi}^i) = 0$, will form a nonlinear manifold in $\mathbb{R}^{N_x^i}$. Strongly-coupled IIML is, by structure, constrained to explore only this nonlinear manifold. The field inversion process (which only consists of gradient-descent-based inadequacy field updates), however, is free to find an optimal solution in the entire N-dimensional space. By introducing the machine learning and field correction steps between gradient-descent-based inadequacy field updates,

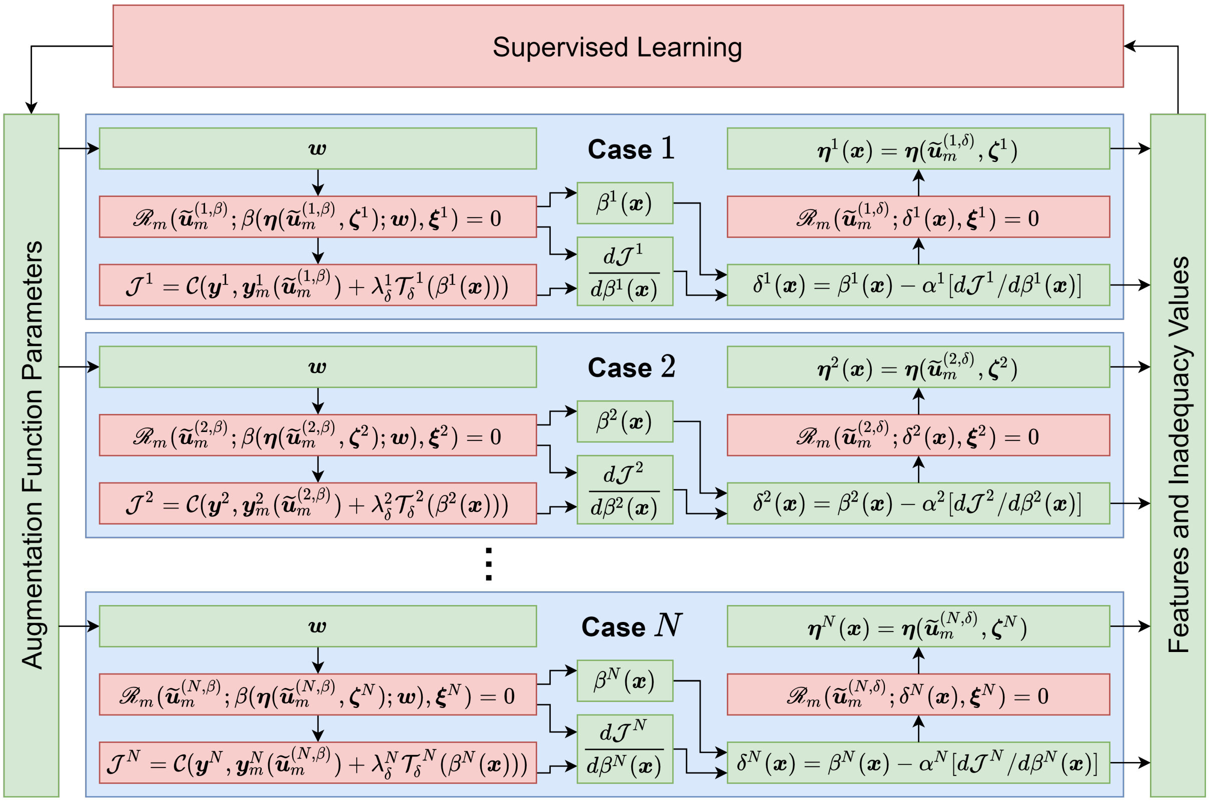

Supervised Learning
Augmentation Function Parameters
Features and Inadequacy Values
Case 1
$\boldsymbol{w}$
$\mathscr{R}_m(\widetilde{\boldsymbol{u}}_m^{(1,\beta)};\beta(\boldsymbol{\eta}(\widetilde{\boldsymbol{u}}_m^{(1,\beta)},\boldsymbol{\zeta}^1);\boldsymbol{w}),\boldsymbol{\xi}^1)=0$
$\mathcal{J}^1=\mathcal{C}(\boldsymbol{y}^1,\boldsymbol{y}_m^1(\widetilde{\boldsymbol{u}}_m^{(1,\beta)})+\lambda_\delta^1\mathcal{T}_\delta^1(\beta^1(\boldsymbol{x})))$
$\beta^1(\boldsymbol{x})$
$\frac{d\mathcal{J}^1}{d\beta^1(\boldsymbol{x})}$
$\boldsymbol{\eta}^1(\boldsymbol{x})=\boldsymbol{\eta}(\widetilde{\boldsymbol{u}}_m^{(1,\delta)},\boldsymbol{\zeta}^1)$
$\mathscr{R}_m(\widetilde{\boldsymbol{u}}_m^{(1,\delta)};\delta^1(\boldsymbol{x}),\boldsymbol{\xi}^1)=0$
$\delta^1(\boldsymbol{x})=\beta^1(\boldsymbol{x})-\alpha^1[d\mathcal{J}^1/d\beta^1(\boldsymbol{x})]$
Case 2
$\boldsymbol{w}$
$\mathscr{R}_m(\widetilde{\boldsymbol{u}}_m^{(2,\beta)};\beta(\boldsymbol{\eta}(\widetilde{\boldsymbol{u}}_m^{(2,\beta)},\boldsymbol{\zeta}^2);\boldsymbol{w}),\boldsymbol{\xi}^2)=0$
$\mathcal{J}^2=\mathcal{C}(\boldsymbol{y}^2,\boldsymbol{y}_m^2(\widetilde{\boldsymbol{u}}_m^{(2,\beta)})+\lambda_\delta^2\mathcal{T}_\delta^2(\beta^2(\boldsymbol{x})))$
$\beta^2(\boldsymbol{x})$
$\frac{d\mathcal{J}^2}{d\beta^2(\boldsymbol{x})}$
$\boldsymbol{\eta}^2(\boldsymbol{x})=\boldsymbol{\eta}(\widetilde{\boldsymbol{u}}_m^{(2,\delta)},\boldsymbol{\zeta}^2)$
$\mathscr{R}_m(\widetilde{\boldsymbol{u}}_m^{(2,\delta)};\delta^2(\boldsymbol{x}),\boldsymbol{\xi}^2)=0$
$\delta^2(\boldsymbol{x})=\beta^2(\boldsymbol{x})-\alpha^2[d\mathcal{J}^2/d\beta^2(\boldsymbol{x})]$
$\vdots$
Case N
$\boldsymbol{w}$
$\mathscr{R}_m(\widetilde{\boldsymbol{u}}_m^{(N,\beta)};\beta(\boldsymbol{\eta}(\widetilde{\boldsymbol{u}}_m^{(N,\beta)},\boldsymbol{\zeta}^N);\boldsymbol{w}),\boldsymbol{\xi}^N)=0$
$\mathcal{J}^N=\mathcal{C}(\boldsymbol{y}^N,\boldsymbol{y}_m^N(\widetilde{\boldsymbol{u}}_m^{(N,\beta)})+\lambda_\delta^N\mathcal{T}_\delta^N(\beta^N(\boldsymbol{x})))$
$\beta^N(\boldsymbol{x})$
$\frac{d\mathcal{J}^N}{d\beta^N(\boldsymbol{x})}$
$\boldsymbol{\eta}^N(\boldsymbol{x})=\boldsymbol{\eta}(\widetilde{\boldsymbol{u}}_m^{(N,\delta)},\boldsymbol{\zeta}^N)$
$\mathscr{R}_m(\widetilde{\boldsymbol{u}}_m^{(N,\delta)};\delta^N(\boldsymbol{x}),\boldsymbol{\xi}^N)=0$
$\delta^N(\boldsymbol{x})=\beta^N(\boldsymbol{x})-\alpha^N[d\mathcal{J}^N/d\beta^N(\boldsymbol{x})]$

Figure 5.1: Flowchart showing the weakly-coupled IIML procedure

the weakly-coupled IIML performs a nonlinear projection operation from a point in the N-dimensional space to a point within the learnable manifold. Hence, within each inference iteration, the inadequacy field can jump out of the learnable manifold after the gradient-descent-based update and is projected back into the manifold by the machine learning and field correction step. This difference in how the iterations progress for the strongly- and weakly-coupled IIML can result in different optimization trajectories within the manifold.

Also note here that the way in which the augmentation function is updated between the two methods is significantly different. While the augmentation function parameters are updated in a single step in the strongly-coupled approach, the weakly-coupled approach provides the flexibility to partially learn from the inadequacy field to drop the loss function $\mathcal{L}$ below some threshold and stop. This flexibility is instrumental for problems where the features are designed such that the true feature-to-augmentation map might not be one-to-one in several parts of the domain. If the strongly-coupled approach is used for such a setup, then there is a possibility that the augmentation behavior being learned during a given inference iteration might be overwritten in some part of the feature-space during a later iteration. This problem worsens if the forward solver restarts from its converged state from the previous learning iteration because it would not access the feature-space locations needed to reach that restart state and hence would not react to the corresponding augmentation behavior being overwritten. This might result in a better training accuracy but when used with a different set of initial conditions, even the training cases might lose the accuracy they gained, or even perform worse in some cases. While the weakly-coupled approach does not eliminate this problem, the partial learning capability helps in retaining the previously learned augmentation behavior for a larger number of inference iterations. This property was instrumental in augmenting the fuel cell model in this work as the problem of poor predictive performance despite good training results was observed when strongly-coupled IIML was used.

5.3 Polymer Electrolyte Membrane Fuel Cells (PEMFCs)

5.3.1 Introduction

The automotive industry is one of the leading producers of greenhouse gas emissions. To meet the challenges of climate change and reduce greenhouse gas emissions, there has been a steady push for development of alternative power-train systems with lower emissions. One such alternative is the Fuel cell (FC) [49], which is an electrochemical device that directly convert chemical energy into electricity with high efficiency. Despite major advancements, the cost and durability of PEMFC vehicles remain a challenge for their large scale adoption in the market. For better control and management of a fuel cell, it is necessary to have physics-based models on-board a vehicle that can run efficiently in real-time with sufficient predictive accuracy [11, 95]. This is due to the fact that direct measurements of important internal states of a fuel cell are very difficult and/or prohibitively expensive in real-time [56]. For instance, one such quantity that significantly affects the performance of a fuel cell is the water content inside the channels and it is difficult to reliably measure in a reasonably feasible manner, thus bringing in the utility of the aforementioned models. There are number of different approaches for modeling fuel cells going from simple 1D models to complex 3D models [4]. On the other hand, the present reduced order fuel cell models [24, 84], which meet the limited computational requirements of an embedded computer, do not achieve satisfactory performance (in terms of model accuracy) or are too difficult to calibrate due to a lack of available information on internal system states. The past few years have seen an emergence of a variety of data-driven techniques which can be used to improve the predictive accuracy of existing low-fidelity models by inferring model-form corrections from available high-fidelity data. Among such techniques are Field Inversion and Machine Learning, symbolic identification approaches, etc. as described in Chapter 1.

In the past few years, machine learning methods have been used to design data-driven surrogate models and control strategies. A brief literature review for the same can be given as follows. Zhu et al. [97] used artificial neural networks (ANN) with considerable success to create a surrogate model for a high temperature proton exchange membrane fuel cell which was further used to conduct a parameter study for the fuel cell geometry and operating conditions - quantities that also served as the inputs to the ANN. Li et al. [40] used data-driven classification strategies supported by carefully chosen feature extraction and data labeling techniques for the diagnosis of water related faults such as membrane drying and flooding. Sun et al. [77] used a hybrid methodology (using both model-based and data-driven) to construct optimal PID and ADRC control strategies for the fuel cell stack cooling. Napoli et al. [47] used classical neural networks along with stacking strategies to develop data-driven fuel cell models to predict the output voltage and cathode temperature of a fuel cell given the stack current and the flow rates for different gases. Ma et al. [43] used recurrent neural networks with G-LSTM (grid long short-term memory) neurons to train and predict the degradation to a fuel cell's performance due to impurities in the incoming hydrogen or changes in the operating conditions. Wang et al. [87] used support vector machines (SVM) to create a data-driven surrogate model from 3D simulation data which was then used to optimize the catalyst layer composition using a genetic algorithm. Using inlet pressures of hydrogen and oxygen, stack temperature and relative humidity as inputs, Han et al. [28] compared the voltage and current predictions obtained from data-driven surrogate models trained using neural networks and support vector machines. A common theme among the aforementioned works is that the data-driven models can predict scalar outputs like stack voltage and stack current but not field quantities within the fuel cell itself. Secondly, most of these models are purely data-driven and do not incorporate physical laws manifested in the traditional models.

In this work, we used the weakly-coupled IIML methodology with the non-intrusive iterative solution strategy to improve the accuracy of an existing linearized 1+1D fuel cell model, in order to match higher-fidelity water content data obtained from the predictions of a proprietary 2D model from Toyota while using only a handful of the available datasets.

5.3.2 Physical modeling of Fuel Cells

A fuel cell is an electrochemical energy conversion device that directly converts chemical energy to electrical energy. In polymer electrolyte membrane fuel cells (PEMFC), hydrogen gas is supplied as the fuel. Hydrogen travels through the gas diffusion layer (GDL) to the catalyst layer. At the anode catalyst layer, a hydrogen oxidation reaction produces protons and electrons.

$$H_2 \longrightarrow 2H^+ + 2e^-$$

Electrons flow through an external circuit to create an electric current, while protons cross the polymer electrolyte membrane. Finally, in the cathode catalyst layer, electrons and protons recombine together with oxygen/air (which is supplied to the cathode channel) to create water in an oxygen reduction reaction.

$$\frac{1}{2}O_2 + 2H^+ + 2e^- \longrightarrow H_2O$$

Modeling of fuel cells requires a description of dynamics in both the through-plane and along-channel dimensions. A schematic is presented in Fig. 5.2 to better illustrate the structure and working of a fuel cell. Due to the large discrepancy in length scales between

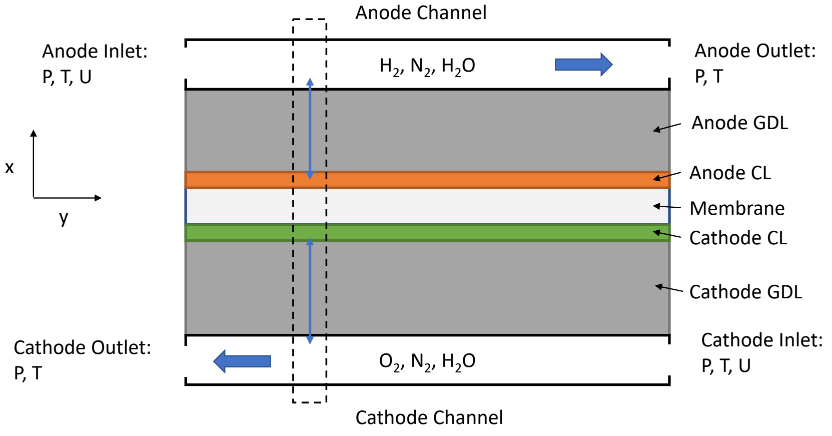


Figure 5.2: Schematic detailing variation of quantities within a fuel cell. The sketch highlights the concentration and temperature gradients across the membrane, catalyst layers, and gas diffusion layers in the through-plane direction.

these dimensions (the aspect ratio is around 10^{-3}, with a 100μm thick GDL and 10cm long channels), the model is usually decomposed into a through-plane model (along the x-direction) an an along-the-channel model (along the y-direction), with coupling between the two dimensions at the GDL-channel interface only (a '1+1D' model).

Full through-cell model

The full through-cell model is a transient model, based on the steady-state model presented by Vetter and Schumacher [84]. The modeling domains are channels, gas diffusion layers (GDLs), and catalyst layers (CLs) in the anode and cathode, with a polymer electrolyte membrane between them as shown by the dashed box in Fig. 5.2 for transport in the x-direction. The letter Ω is used to indicate a modeling domain. The subscripts *ch*, *gdl*, *cl*, and *mb* arer used to denote a channel, gas diffusion layer (GDL), catalyst layer (CL), or polymer electrolyte membrane (PEM) domains respectively, the and superscripts *ca* or *an* denote the cathode and anode sides, respectively. Microporous layers are ignored in this model (following [84]).

Conservation of current and Ohm's law result in the following elliptic system relating the electron potential ϕ_e and the proton potential ϕ_p, to the current densities i_p and i_e and the interfacial current density j.

$$\frac{\partial i_p}{\partial x} = aj \quad \text{where} \quad i_p = -\sigma_p(\lambda, T)\frac{\partial \phi_p}{\partial x} \tag{5.4}$$

$$\frac{\partial i_e}{\partial x} = -aj \quad \text{where} \quad i_e = -\sigma_e\frac{\partial \phi_p}{\partial x} \tag{5.5}$$

Here, a refers to the surface area density, and σ_p and σ_e refer to the electrical conductivity for the protons and electrons, respectively. The conservation of the ionomer water content, λ, is enforced using the water transport model introduced by Springer [71], which consists of a diffusion term and an electro-osmotic drag term, as shown in the following equation.

$$\frac{\varepsilon_i}{V_m}\frac{\partial \lambda}{\partial t} = -\frac{\partial N_\lambda}{\partial x} + S_{ad} + r_{\mathrm{H_2O}} \quad \text{where} \quad N_\lambda = -\frac{D_\lambda(\lambda, T)}{V_m}\frac{\partial \lambda}{\partial x} + \frac{n_d(\lambda)}{F}i_p \tag{5.6}$$

Here, ε_i represents the ionomer volume fraction (which is assumed constant in this model), V_m refers to the equivalent volume of dry membrane, D_λ refers to the diffusivity of the membrane, F is the Faraday's constant, and $r_{\mathrm{H_2O}}$ refers to the rate at which water is produced within the membrane as a consequence of the oxygen reduction reaction in the cathode catalyst layer. S_{ad} is the source term which controls the adsorption/desorption of water within the ionomer membrane. This term is given as follows.

$$S_{ad} = \frac{k_{ad}}{h_{cl}V_m}(\lambda_{eq} - \lambda) \tag{5.7}$$

Here, λ_{eq} refers to the equilibrium membrane water content and is usually given as a function of temperature and relative humidity. k_{ad} refers to the rate of adsorption (when $\lambda < \lambda_{eq}$) or desorption (when $\lambda > \lambda_{eq}$) and is usually a function of λ and temperature.

Gas transport is modeled using gas concentrations (denoted by c) instead of the typically used gas mole fractions. Fickian diffusion is used for the fluxes with an effective diffusivity factor to account for the reduced diffusivity in the porous medium. Additional source terms are used for phase changes from adsorption/desorption and evaporation/condensation.

$$\frac{\partial}{\partial t}(\varepsilon_g c_{\mathrm{H_2O}}) = -\frac{\partial N_{\mathrm{H_2O}}}{\partial x} - S_{ad} - S_{ec} \quad \text{where} \quad N_{\mathrm{H_2O}} = -D^{\mathrm{eff}}_{\mathrm{H_2O}}(s,T)\frac{\partial c_{\mathrm{H_2O}}}{\partial x} \tag{5.8}$$

The gas porosity, ε_g, is given in terms of the liquid water saturation s and the porosity ε_p. Similarly, we can obtain transport equations for hydrogen and oxygen gases, with their source terms arising from the chemical reactions.

$$\frac{\partial}{\partial t}(\varepsilon_g c_{\mathrm{H_2}}) = -\frac{\partial N_{\mathrm{H_2}}}{\partial x} + r_{\mathrm{H_2}} \quad \text{where} \quad N_{\mathrm{H_2}} = -D^{\mathrm{eff}}_{\mathrm{H_2}}(s,T)\frac{\partial c_{\mathrm{H_2}}}{\partial x} \tag{5.9}$$

$$\frac{\partial}{\partial t}(\varepsilon_g c_{\mathrm{O_2}}) = -\frac{\partial N_{\mathrm{O_2}}}{\partial x} + r_{\mathrm{O_2}} \quad \text{where} \quad N_{\mathrm{O_2}} = -D^{\mathrm{eff}}_{\mathrm{O_2}}(s,T)\frac{\partial c_{\mathrm{O_2}}}{\partial x} \tag{5.10}$$

The liquid water saturation, s, is governed by the following equation.

$$\frac{1}{V_w}\frac{\partial}{\partial t}(\varepsilon_\ell c_s) = -\frac{\partial N_s}{\partial x} + S_{ec} \quad \text{where} \quad N_s = -\frac{D^{\mathrm{eff}}_s(s,T)}{V_w}\frac{\partial c_s}{\partial x} \tag{5.11}$$

The liquid volume fraction, ε_ℓ, is given as $\varepsilon_\ell = s\varepsilon_p$ and the capillary liquid water diffusivity, D_s, is given as $D_s = \frac{\kappa}{\mu}\frac{\partial p_c}{s}$. It should be noted that this model is isothermal, so the channel temperature is assumed uniform in the through-cell direction.

The respective source term definitions are given as follows. The Butler-Volmer relation is used to model the exchange-current density j_{cl} induced by the half-reactions in the catalyst layers.

$$j_{cl} = i_0(c_k,T)\left(\exp\left(\frac{2\beta F}{RT}\eta\right) - \exp\left(-\frac{2(1-\beta)F}{RT}\eta\right)\right) \quad \text{where} \quad k \in \{O_2, H_2\} \tag{5.12}$$

Here, η is the overpotential given by

$$\eta = \phi_e - \phi_p - U(c_k,T) \quad \text{where} \quad k \in \{O_2, H_2\} \tag{5.13}$$

i_0 is the exchange-current density and U is the reversible potential difference, both of which are functions of temperature T and the appropriate concentration ($c_{\mathrm{H_2}}$ or $c_{\mathrm{O_2}}$). F

is the Faraday constant. The sign convention used here assumes that j_{cl} is positive at the anode (where the oxidation of hydrogen occurs). Since no reactions occur outside the catalyst layers, the interfacial current density can be written as follows.

$$j = \begin{cases} j_{cl}, & x \in \Omega_{cl} \\ 0, & \text{otherwise} \end{cases} \tag{5.14}$$

The rate of consumption of hydrogen and oxygen can be written in terms of the interfacial current density as follows.

$$r_{H_2} = -\frac{aj}{2F}, \qquad r_{O_2} = \frac{aj}{4F} \tag{5.15}$$

The evaporation/condensation source term can be given as follows in terms of the water vapor concentration and saturation concentration (which is a function of the saturation pressure p_{sat}, which in turn varies with temperature).

$$S_{ec} = \gamma_{ec}(c_{\text{H}_2\text{O}} - c_{\text{sat}}), \qquad c_{\text{sat}} = \frac{p_{\text{sat}}(T)}{RT} \tag{5.16}$$

The rate of evaporation and condensation is given as follows.

$$\gamma_{ec} = \begin{cases} \gamma_e(T)s_{\text{red}}, & c_{\text{H}_2\text{O}} < c_{\text{sat}} \\ \gamma_c(T)(1 - s_{\text{red}}) & c_{\text{H}_2\text{O}} > c_{\text{sat}} \end{cases} \tag{5.17}$$

Here, s_{red} is the reduced liquid water saturation and is given as $s_{\text{red}} = (s - s_{\text{im}})/(1 - s_{\text{im}})$ with s_{im} referring to the immobile saturation.

1-D channel model

The through-cell model is coupled to a 1-D channel model through its boundary conditions, and the channel model governs how these boundary conditions vary along the channel spatial variable y. A counter-flow channel configuration is considered in this model as shown in Fig. 5.2.

The anode and cathode channels have different lengths, but must be modeled on the same

1-D grid to capture the coupling through the membrane. Thus, the spatial dimensions in each channel are non-dimensionalized by the channel length L_{ch}, so that a common spatial variable $y \in [0,1]$ can be used for computations. The concentrations of water, hydrogen, oxygen and nitrogen, are governed by the conservation of mass and their transport is modeled using a convective-diffusive flux. Thus, for any gas $k \in \{\mathrm{H_2O}, \mathrm{O_2}, \mathrm{H_2}, \mathrm{N_2}\}$, we have

$$\frac{\partial c_{k,ch}}{\partial t} = -\frac{1}{L_ch}\frac{\partial N_{k,ch}}{\partial y} + \frac{w}{h_{ch}}S_{k,ch} \quad \text{where} \quad N_{k,ch} = -\frac{D_{k,ch}}{L_{ch}}\frac{\partial c_{k,ch}}{\partial y} + c_{k,ch}v_{ch} \tag{5.18}$$

The gas flow velocity in the channel, v_{ch}, is governed by the following equation

$$\frac{\partial v_{ch}}{\partial y} = \frac{RT_{ch}}{L_{ch}p_{ch}}\frac{w}{h_{ch}}\sum_k S_{k,ch} \tag{5.19}$$

The source term of a species into a channel is equal to the flux of that species from the GDL into the channel in consideration. Hence,

$$S^{an}_{k,ch} = -N_k|_{x=0} \quad \text{and} \quad S^{ca}_{k,ch} = N_k|_{x=h_{tot}} \tag{5.20}$$

To ensure the conservation of mass in the model, it is important to keep track of the liquid water in the channels. Any accumulated liquid water in the channel is convected away by the gas flow velocity with velocity v_{ch}.

$$\frac{\partial s_{ch}}{\partial t} = -\frac{1}{L_{ch}}\frac{\partial (s_{ch}v_{ch})}{\partial y} + \frac{w}{h_{ch}}S_{s,ch} \tag{5.21}$$

It is assumed that the temperature in both the channels is equal to the temperature in the cooling channel which is assumed to vary linearly in y. The cooling channel is oriented in the same direction as the anode channel with inlet at $y = 1$ and outlet at $y = 0$. Thus, we can write $T_{ch} = T_{in} + \Delta T(1 - y)$. Similarly, it is assumed that the pressure varies linearly in both the channels as well. Note that, pressure unlike temperature can be significantly different in the two channels. Thus, one may write,

$$p_{ch}^{an} = p_{in}^{an} + \Delta p^{an}(1-y) \quad \text{and} \quad p_{ch}^{ca} = p_{in}^{ca} + \Delta p^{ca} y \tag{5.22}$$

Lastly, the channel current density, i_{ch}, and the cathode channel potential, $\phi_{e,ch}^{ca}$, are related by Ohm's law in the channel.

$$i_{ch} = -\frac{\sigma_{ch}}{(L_{ch}^{ca})^2}\frac{\partial^2 \phi_{e,ch}^{ca}}{\partial y^2}$$

The need for a reduced order model

Solving a full order model, with appropriately discretized through-cell and channel length scales is exceedingly expensive for on-board real-time use in control systems of devices using PEMFCs. This computational cost can be compounded by any inadequacies in the model which might require further data-driven computations to improve predictions. Thus, it is imperative to use a reduced-order model for quick computations. While there might be additional inadequacies in such a potentially inexpensive model, these inadequacies may be compensated for using data-driven techniques for model augmentation, as is done in this work using integrated inference and learning. To this end a reduced-order through-cell model by Sulzer et al. [76] is used in addition to the aforementioned full channel model for this work. The two models are coupled in the sense that the through-cell model provides the boundary conditions for the channel model and the channel model provides information to evaluate the source term in the through-cell model.

5.3.3 Augmenting the Numerical Solver

The reduced-order through-cell model along with the full channel model construct a system of differential algebraic equations (DAEs) which are implemented in python using the PyBaMM library [75] and numerically solved within the CasADi framework [2] via the Sundials solver. After testing different ways to augment the model, the most promising approach seems to be modifying the algebraic model used to evaluate the equilibrium water content, λ_{eq} (used to calculate S_{ad} in Eqn. 5.8), by multiplying it with the augmentation function β, as it was observed that λ (membrane water content) remained sensitive to it across various physical conditions, viz., dry/humid, low/high current density, low/high temperatures etc. The augmented form of the source term S_{ad} (see Eqn. 5.7) is shown in Eqn. 5.23.

$$S_{ad}^{\text{aug}} = \frac{k_{ad}}{h_{cl} V_m} (\beta_{\text{aug}}(\boldsymbol{\eta}_{\text{aug}}; \boldsymbol{w}) \lambda_{eq} - \lambda) \tag{5.23}$$

Here, $\boldsymbol{\eta}_{\text{aug}}$ represents the features and $\boldsymbol{w}$ represents the parameters that characterize the augmentation function. The feature set used for this application contained the following quantities.

1. Mole fraction of water vapor in the anode channel

2. Temperature inside the cathode channel

3. Mole fraction of water vapor in the cathode channel

4. Water content in the anode catalyst layer

5. Water vapor concentration in the anode catalyst layer

6. Water content in the cathode catalyst layer

7. Water vapor concentration in the cathode catalyst layer

8. Membrane water content

In this work, the functional form for the augmentation was chosen to be a neural network with 2 hidden layers containing 7 nodes each. The sigmoid activation function was used in the hidden layers. The ReLU activation function was used in the output layer to ensure that the augmentation was non-negative. The Keras library [9] was used to create and train the network. The Adam optimizer [34] was used to train the model for a total of 500 epochs after every gradient-descent-based update of the augmentation field. The learning rate was set to be 10^{-3}.

5.4 Results

The available dataset contains 1224 cases, each uniquely characterized by different inflow conditions. The high-fidelity data used to infer the augmentation function is the resulting steady-state x-averaged membrane water content. The cost function for any case with index j was defined as

$$\mathcal{C}_j = \|\lambda_j - \lambda_{\text{data},j}\|_2^2 \tag{5.24}$$

where λ refers to the spatial field of the membrane water content along the channel direction y. Since no regularization is used, the cost function is identical to the individual objective function for a given case. The combined objective function for all the cases is calculated as the weighted sum of the individual cost functions of all training cases with all weights set to unity. Mathematically,

$$\mathcal{J} = \sum_i \alpha_i \mathcal{C}_i \tag{5.25}$$

where α_j represents the weights for the j^{th} case which in this particular instance are all set to 1.

The spatial domain used to solve the model is discretized along the channel into 20 spatial nodes. As mentioned before, the unsteady model is used to obtain the steady-state solution by running it for a sufficiently long amount of physical time which in this case was 1000 seconds. Due to the relatively low dimensionality of the spatial discretization, finite differences were found feasible to obtain the sensitivities of the cost function w.r.t. the augmentation field, β. The step-size used for finite differences was 10^{-4}. The model was trained on only 14 configurations out of 1224. The corresponding IDs for these training cases in the dataset are 40, 100, 125, 155, 190, 230, 400, 685, 740, 840, 865, 1000, 1090 and 1200.

A representative plot for the residual histories of the states being solved for a given augmentation field is shown in Fig. 5.3. As can be seen, the residuals approach zero within the chosen time interval that the model is solved for. It must be noted that no

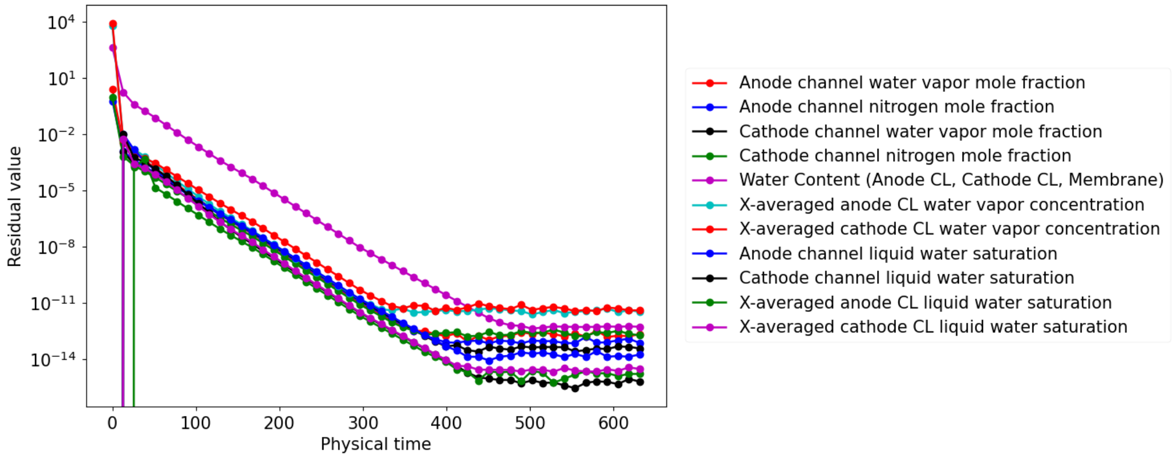


Figure 5.3: Representative plot for residual decay of all the state variables being solved for by the model

residual-based stopping criteria is built into the DAE solver used for this work.

A representative plot for the augmentation residual (R_{aug}) history resulting from the iterative solution of the augmented model is presented in Fig. 5.4. Such iterative solutions need to be performed during both training and prediction. Note here that a stopping condition of $R_{\text{aug}} < 10^{-3}$ was found to provide a sufficiently converged result. While there do exist a few cases where such a convergence cannot be achieved and the residuals keep oscillating, no cases exhibit divergent behavior. Even in the cases where the augmentation residuals keep oscillating, the residual magnitudes are very small (of the order of 10^{-2}).

5.4.1 Training

The minimization history of the combined objective function for all 14 cases is shown in Fig. 5.5. The optimization could not proceed beyond iteration 23 because any subsequent augmentation function iterates caused the solver to diverge. Predictive improvements in ionomer water content (λ) distributions w.r.t. available high-fidelity data for all training cases are plotted in Fig. 5.6. As can be seen in the figure, some cases show very good improvements while some improve only marginally. This behavior can be attributed to the combined objective function being less sensitive to the feature-space regions where the features corresponding to the marginally improved cases lie. While a more careful choice of the training cases and the corresponding weights to the individual objective functions

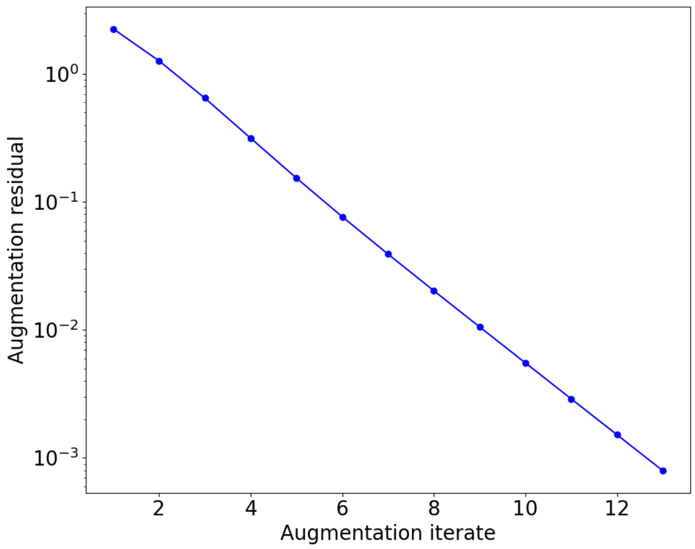


Figure 5.4: Representative plot for R_{aug} convergence across the iterative solution of the augmented model

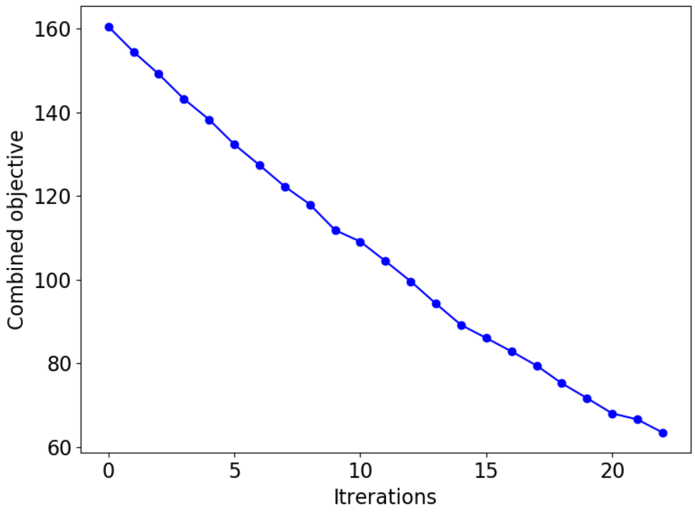
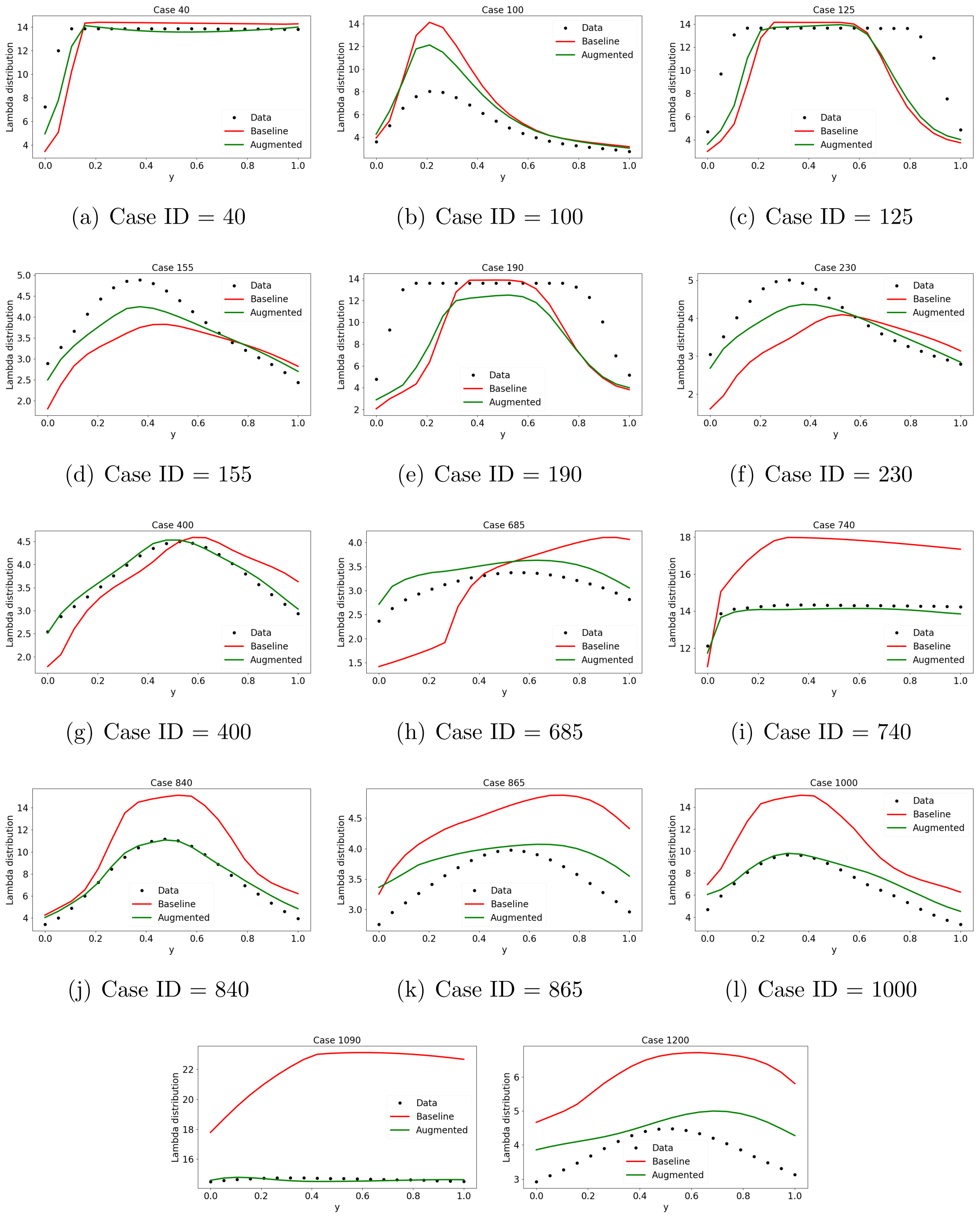


Figure 5.5: Optimization history for weakly-coupled inference and learning

within the combined objective function might help, the objective here is to demonstrate the viability of the IIML approach to obtain generalizable improvements to the model.

5.4.2 Testing

Once the training was completed, the resulting model was further tested over all available 1224 cases, the results for which are summarized in Fig. 5.7 using the following



(a) Case ID = 40

(b) Case ID = 100

(c) Case ID = 125

(d) Case ID = 155

(e) Case ID = 190

(f) Case ID = 230

(g) Case ID = 400

(h) Case ID = 685

(i) Case ID = 740

(j) Case ID = 840

(k) Case ID = 865

(l) Case ID = 1000

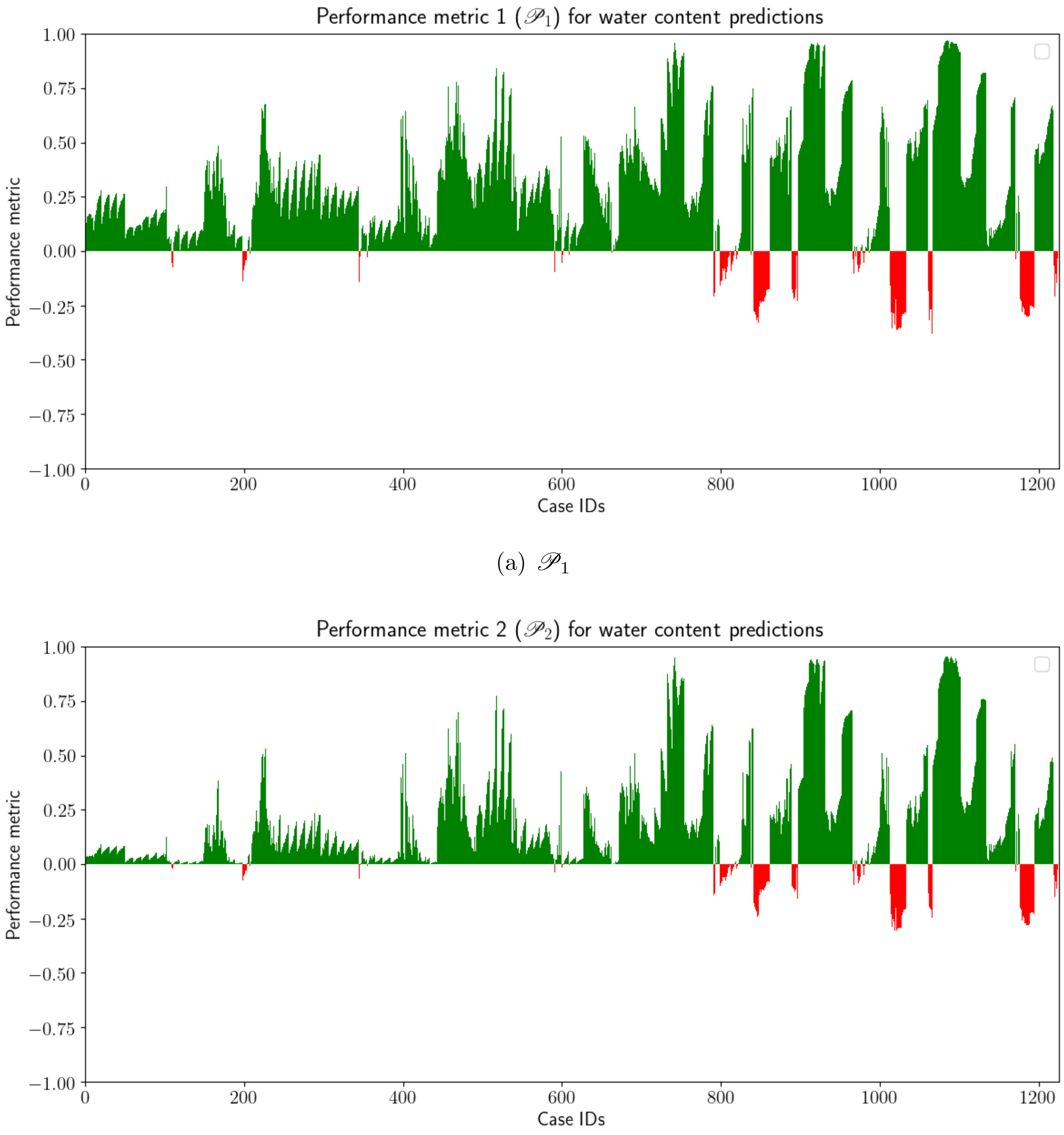
(m) Case ID = 1090

(n) Case ID = 1200

Figure 5.6: Ionomer water content predictions for the training cases

performance metrics, $\mathscr{P}_1$ and $\mathscr{P}_2$, which are defined for any quantity of interest q as

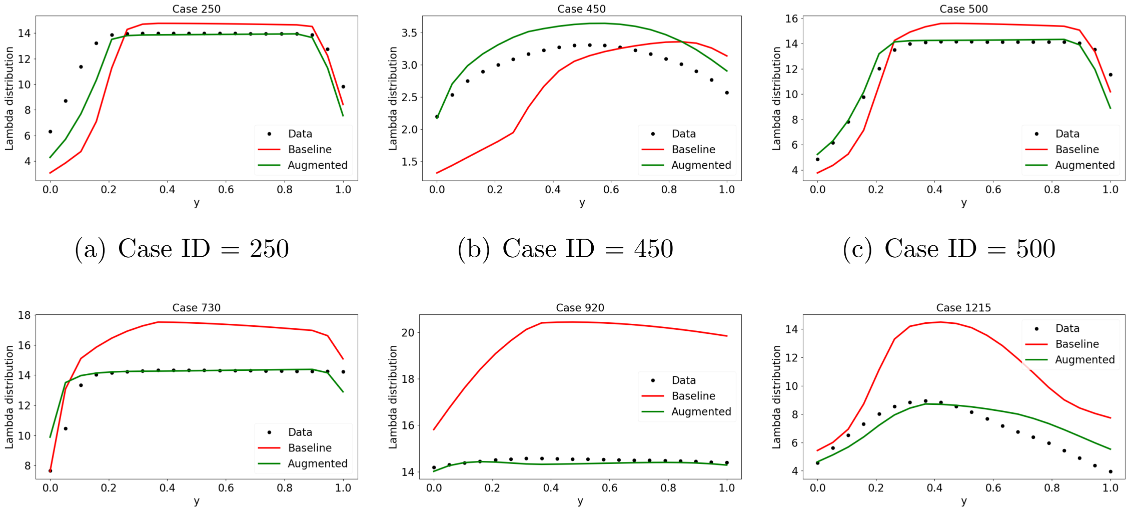
$$\mathscr{P}_1(q) = \frac{2\left\|q_{\text{baseline}} - q_{\text{data}}\right\|_2}{\left\|q_{\text{augmented}} - q_{\text{data}}\right\|_2 + \left\|q_{\text{baseline}} - q_{\text{data}}\right\|_2} - 1 \tag{5.26}$$



(a) $\mathscr{P}_1$

(b) $\mathscr{P}_2$

Figure 5.7: Performance metrics for ionomer water content predictions across all 1224 cases



(a) Case ID = 250

(b) Case ID = 450

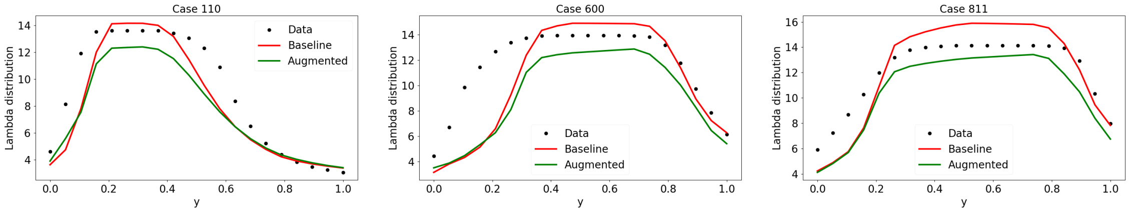
(c) Case ID = 500

(d) Case ID = 730

(e) Case ID = 920

(f) Case ID = 1215

Figure 5.8: Ionomer water content predictions for cases with high $\mathscr{P}_1$ values

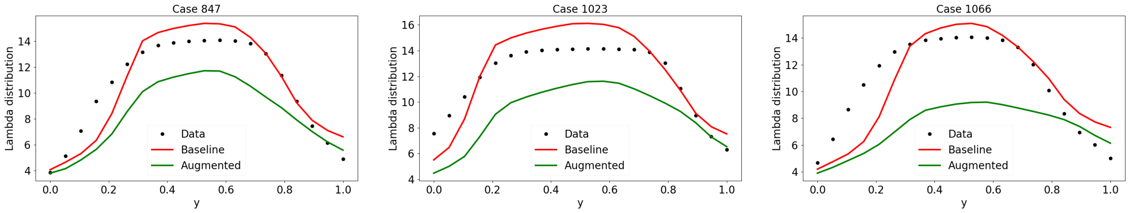


(a) Case ID = 110

(b) Case ID = 600

(c) Case ID = 811

Figure 5.9: Ionomer water content predictions for cases with $\mathscr{P}_1$ values closest to zero



(a) Case ID = 847

(b) Case ID = 1023

(c) Case ID = 1066

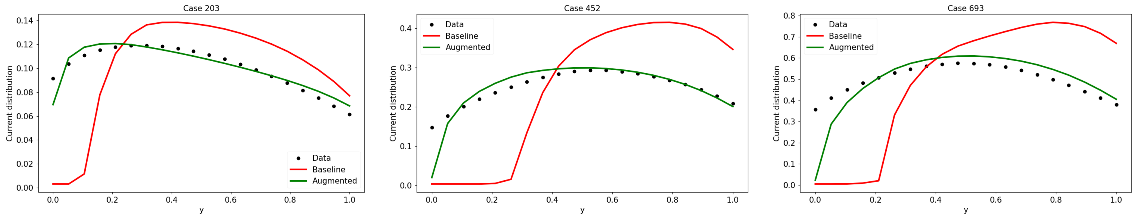
Figure 5.10: Ionomer water content predictions for cases with low $\mathscr{P}_1$

$$\mathscr{P}_2(q) = \frac{\mathscr{P}_1(q)\, \|q_{\text{augmented}} - q_{\text{baseline}}\|_2}{\|q_{\text{augmented}} - q_{\text{data}}\|_2 + \|q_{\text{baseline}} - q_{\text{data}}\|_2} \tag{5.27}$$

The performance metric $\mathscr{P}_1$, by design, are positive for cases where the augmented model gives a smaller L_2 error compared to the baseline model and vice versa. The performance metric $\mathscr{P}_2$ scales $\mathscr{P}_1$ with a relative difference between the predictions from the augmented and baseline models. Thus, for a given case, a high ratio $\mathscr{P}_1/\mathscr{P}_2$ means that the baseline and augmented profiles are very close and that the baseline profile was reasonably accurate in the first place. The cases where accuracy has improved are shown in green whereas the cases where it has deteriorated are shown in red. 1087 out of 1224 cases exhibited a lower L2 error compared to the baseline. Figs. 5.8, 5.9 and 5.10 show representative results associated with highly improved, marginally different and significantly deteriorated performance metrics. As can be seen in the results, the model seems to improve the predictions for a range of different physical conditions after training on just 14 representative cases. Also, it should be noticed that for some cases with only a marginal difference between L2 errors, the predictions are significantly different while predicting more accurately in one part of the physical space while falling short of even the baseline model in others. Given the complex interactions between various sub-models within the fuel-cell model itself and such a high-dimensional feature-space, a model would require a highly intricate functional form and a large amount of data to make accurate predictions for any given inflow conditions, if such predictions are possible at all. The objective here is to demonstrate the range of applicability of such models. A small number of training cases are used to eliminate the possibility of a significant number of testing cases being very similar to the training cases.

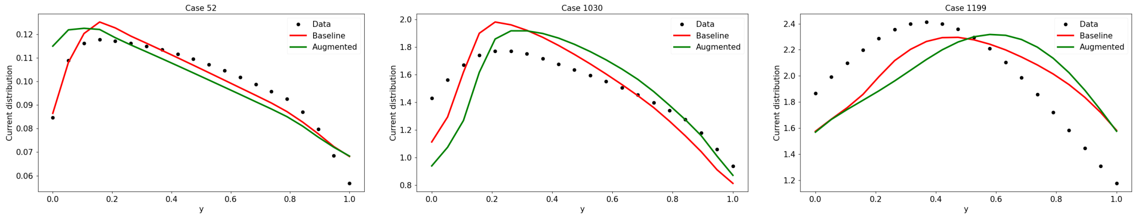
5.4.3 Changes in Current Density Predictions

To judge the quality of predictions for other physical quantities, individual comparisons for the current density distributions are presented in Figs. 5.11 and 5.12 for a few selected cases which show better and worse results compared to the corresponding high-fidelity data, respectively. The performance metrics w.r.t. the predictions for current density distributions are summarized in 5.13. Since the current density is not the intended



(a) Case ID = 203 (b) Case ID = 452 (c) Case ID = 693

Figure 5.11: Current density predictions for cases with high performance metrics



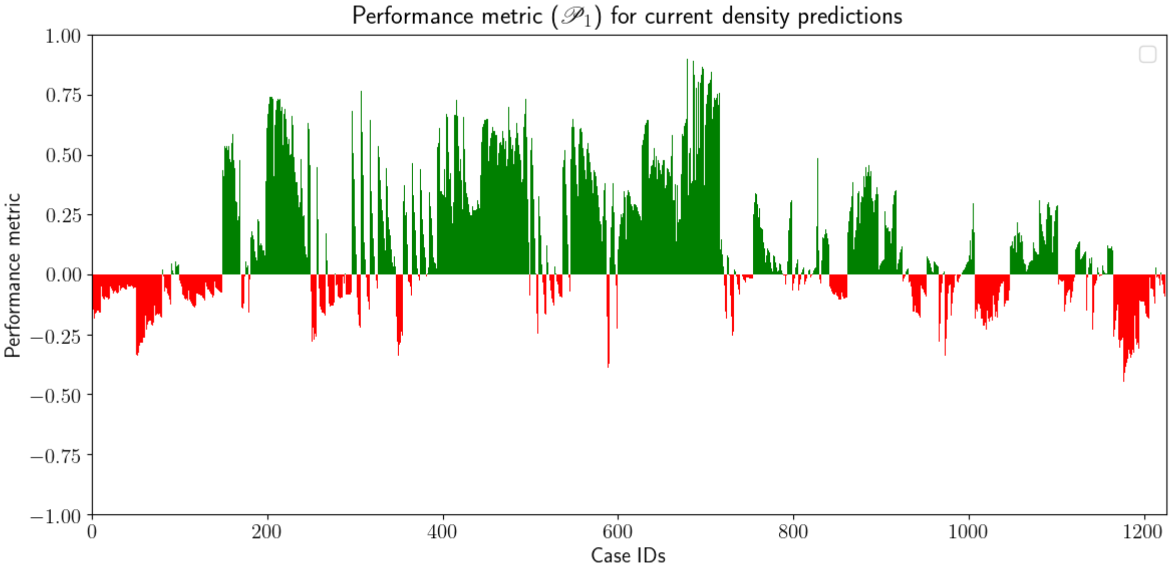
(a) Case ID = 52 (b) Case ID = 1030 (c) Case ID = 1199

Figure 5.12: Current density predictions for cases with low performance metrics

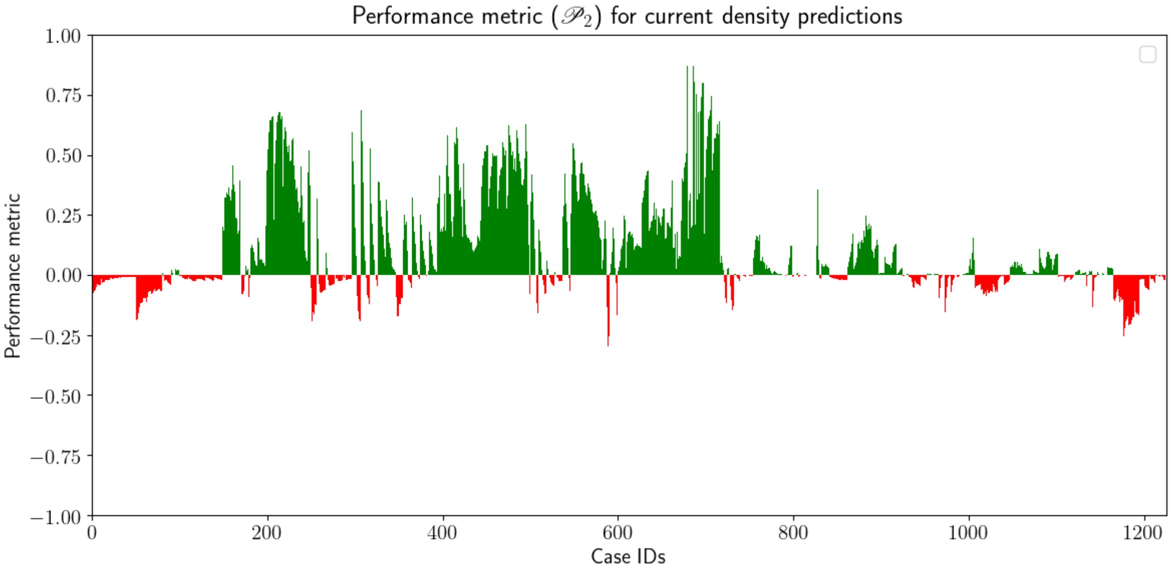
output of the augmented model, the presented results are not completely unexpected. However, it should be noted here that even for several cases with fairly low $\mathscr{P}_1$, $\mathscr{P}_2$ is significantly smaller in magnitude, i.e., the predictions from the augmented model are close to those from the baseline model. Thus, for most cases, the augmented model either stays close to the baseline model or improves it. For several cases with high performance metrics, we do see a significant correction in the current density predictions. Finally, note that for a few cases (e.g., case 1199), even though the prediction error for the ionomer water content decreases, even the qualitative trends for the water content are wrong, and correspondingly, the current density predictions also contain significant errors when compared to the high-fidelity data. Further work and analysis is needed to ascertain whether such cases require a different treatment during the inference process.

5.4.4 Effects of a smaller training dataset

To illustrate the impact of adding/removing training configurations, a second model was trained using only 7 training configurations (case IDs 40, 125, 190, 400, 740, 865 and 1090) instead of 14, and as can be seen from Fig. 5.14, the performance of the augmented model immediately deteriorates and it is able to achieve better-than-baseline performance for



(a) $\mathscr{P}_1$



(b) $\mathscr{P}_2$

Figure 5.13: Performance metrics for current density predictions across all 1224 cases

only 777 cases out of a total of 1224 that it was tested for. It can also be seen that

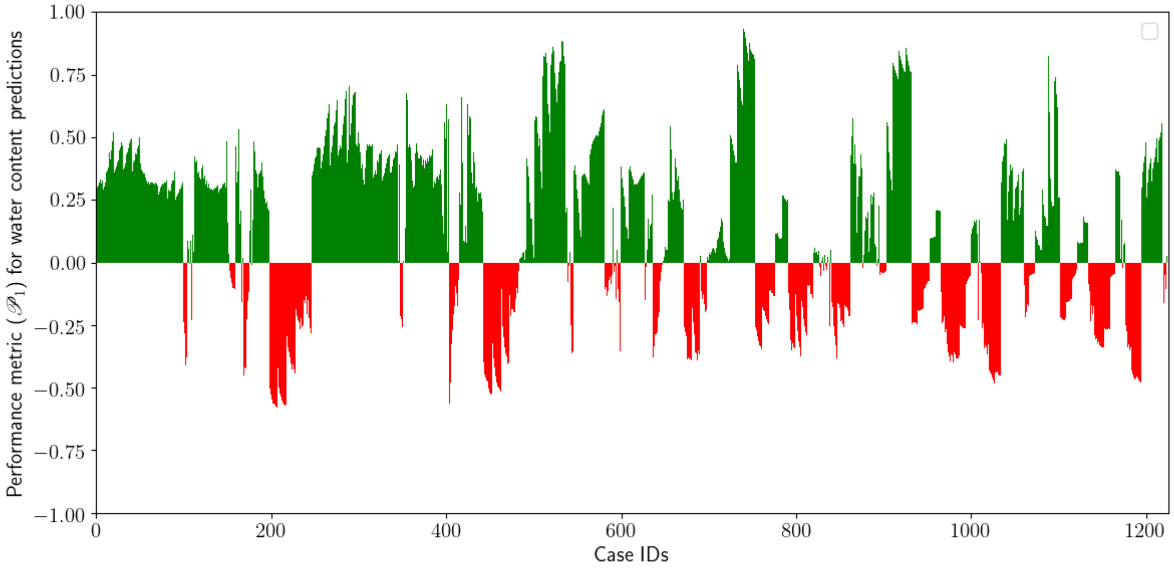


Figure 5.14: Performance metric for all cases when trained with only 7 (instead of 14) cases

the performance metric for many cases deteriorates drastically, while it improves for a handful of cases. These are either cases which were used during training or which share very similar inflow conditions with them. This behavior is caused by the inference and learning process overfitting to the augmentation behavior specific to the few training cases it has been provided with. While it improves predictions on the training cases, overfitting is an undesirable outcome as it results in poorer predictive accuracy for cases different than those in the training dataset and hence, hurts generalizability. Thus, an ideal training dataset contains as low number of configurations as possible while representing as wide a range of physical behaviors as possible.

Chapter 6

Summary

Machine learning techniques offer systematic approaches to extract model-form inadequacies as intricate functions of modeled quantities from available data. Several such data-driven frameworks have been introduced in the past few years and progress has been made in the direction of extracting model-consistent relationships from data obtained via experiments or high-fidelity simulations. In practice, however, improvement in predictive accuracy is typically limited to geometries and boundary conditions similar to those in the training dataset. Moreover, data-driven techniques can sometimes lead to a deterioration in predictive accuracy compared to the baseline model, even in canonical problems only mildly different from those in the training dataset.

This thesis presents new principles, techniques and infrastructure associated with learning and inference of data-driven model augmentations. Two complementary model-consistent frameworks were proposed: (1) Learning and Inference assisted by Feature-space Engineering (LIFE); and (2) Non-intrusive Weakly-coupled Integrated Inference and Learning (IIML); which offer a choice between high generalizability/robustness, and minimal implementation effort, respectively.

6.1 LIFE

The LIFE framework is based on strongly-coupled inference and machine Learning, and offers modelers tools and guiding principles to design the feature-space, and techniques to construct a functional form for the augmentation capable of localized learning. While a meticulous choice of features is critical to generalizability, localized learning offers robustness by minimizing unknown spurious behavior in the resulting augmentation. The following guiding principles were laid out in the framework to help design features that are conducive for a generalizable augmentation.

1. Physical quantities sharing a direct causal relationship with the inadequacy should be used to design features. This requirement can be relaxed for steady-state models, where correlated quantities may also be used.

2. Physics-based non-dimensionalization must be used wherever possible as it offers far better generalization of the inferred augmentation compared to its statistics-based counterpart.

3. Features must be designed, if possible, to be effectively bounded, as limited data can populate a significant fraction of a bounded feature-space.

4. Features must be judiciously used, as using too few features might result in the augmentation not being able to distinguish regions where different augmentation values are required and using too many could lead to a distinction between feature-space regions where the required augmentation values are very close, hence requiring more data to characterize the feature-space.

Localized learning is pivotal in improving the robustness of an augmentation as it ensures that the augmentation values remain unaffected in feature-space regions which are far from any available datapoint. Hence, if an unseen geometry or boundary condition corresponds to a region within the feature-space which remained unpopulated by datapoints in the overall inference process, the model reverts to its baseline behavior instead of predicting unreliable values. The range of influence of a datapoint is an important quantity that determines the vicinity in which it modifies the augmentation behavior in the feature-space. It needs to be either set by the user before the inference process, or optimized as a hyperparameter. For an augmentation with an interpolation-based functional form, the resolution of the grid constructed in the feature-space acts as the range of influence. If the range of influence is too small, the training accuracy improves while the generalizability deteriorates, and vice-versa. Thus, a balance has to be maintained for optimal results. Finally, by virtue of localized learning within a bounded and parsimonious feature-space, the LIFE framework also becomes modular. Augmented models can be treated as baseline models to introduce newer augmentations which are referred to as hierarchical augmentations. While isolating the effect of such hierarchical augmentations to physical conditions of interest can sometimes be a challenge, the framework allows users to build several levels of such augmentations in decreasing order of generalizability. Note that this modularity cannot be achieved unless the augmentation behavior can be characterized in all parts of the feature space which requires careful feature design and localized learning.

6.2 IIML

The two main steps that could potentially require significant time and effort when constructing data augmented models include embedding the augmentation within the numerical solver and designing a good feature-space. It must be noted here that it is not always possible to design a feature-space that is conducive to generalizable augmentations. The weakly-coupled IIML framework (when combined with the non-intrusive iterative method to solve augmented model equations presented in Section 5.1) is designed to enable users to setup the inference problem quickly and with minimal changes to the numerical solver. This framework provides users with the same advantages as its strongly-coupled counterpart, viz., constraining the inference problem on a learnable manifold and enforcing consistency among the functional relationships shared between the features and the augmentation across the training dataset. The methodology involves solving an inverse problem to obtain a spatial field of augmentation values such that each gradient-descent-based update of this field (performed independently on each training case) during the inference process is followed by a machine learning step that collects spatial fields of features and augmentations from all training cases and extracts the learnable information from it. Following the machine learning step, a forward run is carried out independently for each dataset to ensure consistency of the converged solution with the augmented model. An added advantage of the weakly-coupled learning approach is that it can facilitate partial learning (i.e., reducing the cost function to a set non-zero threshold or up to a set number of iterations) which can help the augmentation retain the information learned in the previous iterations. This makes the augmentation more resilient against possible modifications in regions of the feature-space where the augmentation behavior has already been learned. The ensuing robustness can be critical when learning augmentations in sub-optimal and/or high-dimensional feature-spaces. While a sub-optimal feature-space does significantly sacrifice on generalizability, the weakly-coupled framework is robust enough to extract usable augmentations in such feature-spaces that improve predictions on unseen cases similar to those present in the training dataset.

6.3 Application in Transition modeling

To demonstrate its capabilities, the LIFE framework was used to create a data-driven model for bypass transition. A bare-bones intermittency transport equation was introduced within a variant of Wilcox's 1988 k-ω model. An augmentation was introduced within this equation and a three-dimensional feature-space was carefully designed. An interpolation-based functional form was chosen for the augmentation to facilitate localized learning. When trained on only two flat plate cases, the model consistently improved predictions across a diverse range of unseen cases which included other flat plate, turbine cascade and compressor cascade cases characterized by different Reynolds numbers, Mach numbers, freestream turbulence intensities, etc. It should be noted here that discrepancies between the true transition locations and those predicted by the augmented model were seen in a few cases. These were mainly caused either due to the augmentation not being characterized in the corresponding feature-space regions or due to the transition being induced by flow separation. To extend the capability of the model to predict separation-induced transition, a hierarchical augmentation term was introduced alongside the already inferred one. Its parameters were then inferred using LES data for one of the compressor cascade cases while holding the original augmentation parameters constant. To ensure that it activates only when separation-induced transition occurs, an appropriate blending function was also designed. The hierarchical augmentation further improved predictions across all compressor cascade cases. While it offers generalizability, robustness and modularity, the LIFE framework comes with the following caveats.

- Designing good features can be an arduous and time-consuming task and requires knowledge, intuition and experience.
- Optimally resolving the feature-space is key to balance generalizability and accuracy during localized learning.
- Isolating the effects of hierarchical augmentations to regions of interest could be difficult.

Finally, the guiding principles outlined here are not to be considered exhaustive, and future work (specific to a domain or otherwise) might help in making the framework more robust and efficient.

6.4 Application in Fuel Cell Modeling

The weakly-coupled IIML framework was demonstrated by augmenting a Polymer Electrolyte Membrane Fuel Cell (PEMFC) model to improve ionomer water content predictions. The augmented model was solved using a non-intrusive iterative method which did not require the augmentation function to be implemented within the numerical solver. When trained on only a handful of representative cases to create a training dataset, the overall predictive accuracy improved across a range of input configurations on the same geometry. In addition to improvements in ionomer water content predictions, the respective current density distributions were also predicted more accurately. While there were cases where the predictive accuracy deteriorated, these cases were comparatively far fewer in number. It was also observed that while the predictive accuracy dropped when only a subset of the training dataset was used, it improved for certain training cases as the augmentation did not have to compromise as much during training to improve performance across all training cases. Thus, it was shown that in cases in which strongly-coupled IIML was incapable of inferring an augmentation that is consistent across inference iterations (and hence cannot be reliably used in a predictive setting), the weakly-coupled counterpart offers a comparatively better alternative that reduces the chances of such a problem occurring. Hence, if the application is restricted to configurations similar to those in the training dataset, the weakly-coupled IIML can be used in conjunction with non-embedded augmentations to quickly obtain "usable" augmentations without spending considerable resources towards embedding the augmentation, feature design, localized learning etc.

6.5 Future Work

The current version of the LIFE framework makes use of a rudimentary functional form capable of localized learning. Significant efforts need to be made to improve techniques that can offer adaptive resolution for localized learning within the feature-space without compromising computational efficiency. On the other hand, one needs to address and estimate contributions from the following sources of epistemic uncertainties to make meaningful predictions for use in design, analysis and optimization. The following are directions for future work:

1. **Chosen model-form:** The way an augmentation term is introduced within the model can lead it to address some part of the model inadequacy better than others.

2. **Imperfect feature selection:** If the chosen features are not optimal w.r.t. the augmentation term introduced within the model, significantly different augmentation values might be needed for points that are very close in the feature-space in order to improve the predictive accuracy.

3. **Lack of data in a feature-space region:** If no training datapoint is available for a region in the feature-space, then even though the corresponding predictions will have the baseline value by virtue of localized learning, they should be highly uncertain.

4. **Resolution in the feature-space:** Given the limited availability of data, the

augmentation function values need to be interpolated in the rest of the feature-space and hence such interpolation might also contribute to the uncertainty.

5. **Uncertainty quantification:** A rudimentary uncertainty quantification strategy would be to construct different augmentations by varying model-forms, features, training data and functional forms and then estimating an interval-based measure of uncertainty as demonstrated in Appendix B. However, more formal and sophisticated techniques are needed to improve such estimates.

Note that points 2-5 are closely linked and hence cannot be tackled independently. Finally, The modularity resulting from the use of localized learning makes LIFE an excellent candidate for use as a symbiotic architecture enabling both data-driven inference of generalizable models, and design-of-experiments, simultaneously – an avenue of practical interest worth exploring in the future.

Appendices

Appendix A

Design under Model-form Uncertainty - A Case Study

RANS models are currently the industry's workhorse for preliminary design and optimization of flow geometries, and will likely remain so for at least a couple of decades. However, the inadequacies in these models lead to significant errors when predicting quantities of interests for intricate geometries in multi-physics simulations. While model augmentation strategies can attempt to correct these predictions, it is nearly impossible to account for all sources of model inaccuracy and hence there will always exist some measure of model-form uncertainty within the predictions from these models. Estimation and minimization of these model-form uncertainties can result in better-informed simulation-based design strategies, and can even play an important role in multi-fidelity optimization frameworks. The work described in this appendix describes the robust design of an aircraft engine nozzle under model-form uncertainties within a class of ML-augmented Spalart-Allmaras models.

A.1 Creating a Data-driven Family of Models

An interval-based estimate of the uncertainty in the quantity of interest was used as a measure of the model-form uncertainty within the family of models used. The aforementioned family of models was constructed by inferring augmentations to the Spalart-Allmaras model using the Field Inversion and Machine Learning (FIML) framework. The augmentation term in consideration was introduced into the model by simply multiplying it with the production term within the model. Skin friction data from six different model geometries are used to infer the augmentation term and build six respective ML-augmented variants of the Spalart-Allmaras model. The "high-fidelity" skin friction data used here to infer the augmentation function was obtained using the Wilcox stress-ω model from 2008 [92]. The maximum aposteriori estimate for the numerical field of the augmentation values obtained via Bayesian inference for each case independently is used to generate feature values at all spatial locations in the respective computational domains. The features used for this application are listed as follows:

$$\begin{aligned}\eta_1 &= \frac{\rho|S|d_w^2}{\mu_L}\\ \eta_2 &= \frac{\mu_t|S|}{\max(\tau_w, 10^{-10}\mathrm{Nm}^{-2})} \qquad (\text{A.1})\\ \eta_3 &= d_w\end{aligned}$$

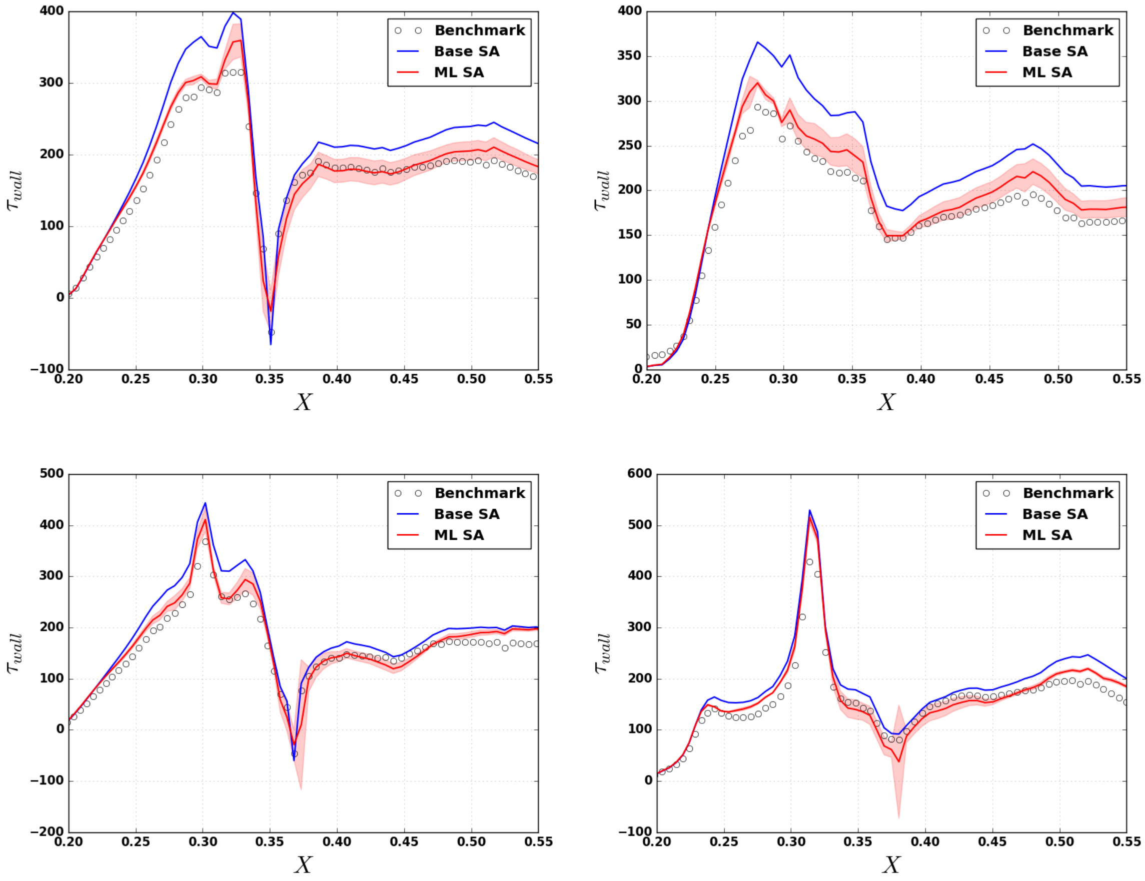


Figure A.1: Prediction of skin friction over the nozzle wall for four different nozzle geometries

Using the feature-augmentation value pairs at all spatial locations, separate ML models are trained for each case using the AdaBoost algorithm [21] (from the openly available SKLearn library [53]). Fig. A.1 shows predictions of some of these ML variants for their respective training cases. The bands seen around the red curves (predictions by the augmented models) correspond to the confidence intervals which are a part of ML predictions.

A.2 Setting up the Optimization Problem

The axisymmetric inner nozzle surface is parametrized by a 2-D cubic B-spline curve consisting of 15 control points. Repeated knots at the inlet, throat and outlet of the nozzle ensure desirable geometric characteristics at these locations. The first four control points were fixed to guarantee a smooth geometry near the inlet, leaving 11 control points (22 x-y coordinate pairs) free for optimization. The throat of the nozzle however is characterized by two points with very close x-values and the same y-values to ensure that the throat has the minimum cross-sectional area. Thus, 21 design variables (removing the redundant y-coordinate from the count) characterize the geometry. The following constraints were imposed on the control polygon to ensure that: (1) the variation in the coordinates is within 40% of the initial values; (2) the nozzle converges upstream of the throat and diverges downstream; (3) the throat always has the minimum cross section; and (4) the gradients in the geometry are not too steep.

$$
\begin{array}{ccccccc}
0.6x_i^{\text{initial}} & < & x_i & < & 1.4x_i^{\text{initial}} & & \text{for} \quad i \in [5,15] \\
0.6y_i^{\text{initial}} & < & y_i & < & 1.4y_i^{\text{initial}} & & \text{for} \quad i \in [5,15] \\
-1.0(x_i - x_{i-1}) & < & y_i - y_{i-1} & < & 0 & & \text{for} \quad i = 6 \\
-1.8(x_i - x_{i-1}) & < & y_i - y_{i-1} & < & 0 & & \text{for} \quad i = 7 \\
-0.20(x_i - x_{i-1}) & < & y_i - y_{i-1} & < & 0 & & \text{for} \quad i \in \{5,8\} \\
0.04(x_i - x_{i-1}) & < & y_i - y_{i-1} & < & 0.16(x_i - x_{i-1}) & & \text{for} \quad i = 10 \\
0.04(x_i - x_{i-1}) & < & y_i - y_{i-1} & < & 0.20(x_i - x_{i-1}) & & \text{for} \quad i = 11 \\
0.07(x_i - x_{i-1}) & < & y_i - y_{i-1} & < & 0.30(x_i - x_{i-1}) & & \text{for} \quad i = 12 \\
0 & < & y_i - y_{i-1} & < & 0.20(x_i - x_{i-1}) & & \text{for} \quad i = 13
\end{array}
$$

$$
\begin{array}{llllll}
0 & < & y_i - y_{i-1} & < & 0.10(x_i - x_{i-1}) & \text{for} \quad i = 14 \\
0 & < & y_i - y_{i-1} & < & 0.05(x_i - x_{i-1}) & \text{for} \quad i = 15 \\
0.01 & < & x_i - x_{i-1} & & & \text{for} \quad i \in [5, 15] \\
0.001 & < & y_i - y_8 & & & \text{for} \quad i \in [5, 15] - \{8\}
\end{array}
$$

The objective of the deterministic optimization problem is to maximize the thrust (T) predicted by the baseline Spalart-Allmaras model under the geometric constraints as shown above.

$$\underset{\boldsymbol{x}}{\text{maximize}} \quad T_{\text{SA}}(\boldsymbol{x}) \tag{A.2}$$

This deterministic optimization was performed without and with the non-linear mass (m) constraint.

$$m(\boldsymbol{x}) < 75\text{kg} \tag{A.3}$$

The objective of the robust design optimization problem is to minimize the maximum discrepancy between thrust predicted using the baseline and all the ML-augmented variants of the turbulence model.

$$
\begin{aligned}
\underset{\boldsymbol{x}}{\text{minimize}} \quad & \max_{\mathcal{M}^i}(T_{\mathcal{M}^i}(\boldsymbol{x})) - \min_{\mathcal{M}^i}(T_{\mathcal{M}^i}(\boldsymbol{x})) \\
& s.t. \quad T_{\mathcal{M}^i}(\boldsymbol{x}) > 21\text{kN} \quad \forall\, \mathcal{M}^i
\end{aligned} \tag{A.4}
$$

The subscript $\mathcal{M}^i$ refers to the i^{th} model in the chosen family of models. Note that one of these models is the baseline Spalart-Allmaras model itself. This robust design optimization was also performed without and with an additional non-linear mass constraint similar to the deterministic case (see eqn. A.3).

A.3 Computational Framework

The computational framework consisted of the DAKOTA optimization package calling SU^2 and Gmsh, to run flow simulations and return thrust values, and to remesh the updated geometry for each optimization iteration, respectively. Sensitivity w.r.t. design variables were obtained using the finite difference method, resulting in 22 (1 baseline + 21 perturbed design variables) simulations per optimization iteration for the deterministic case, which increased by a factor of 7 (1 baseline SA + 6 ML variants) for the robust design optimization case. Within DAKOTA, the NPSOL subpackage, which uses sequential quadratic programming, was used as the optimizer. A flowchart describing the entire framework is shown in Fig. A.2.

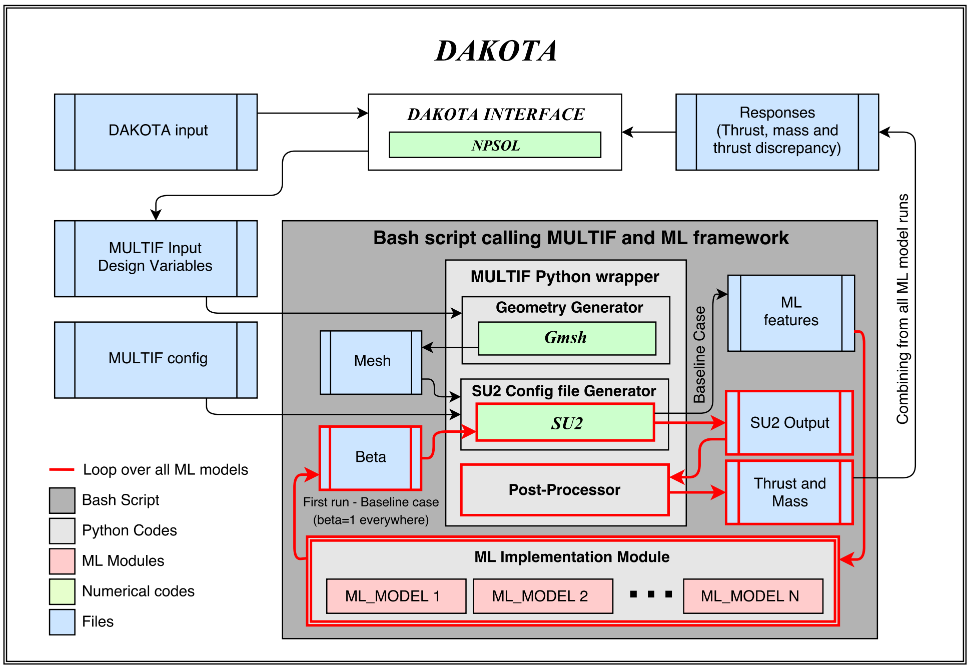


Figure A.2: Computational framework used to solve the optimization problems

A.4 Results and Conclusions

A.4.1 Optimization without mass constraints

The thrust values across optimization iterations, for both the deterministic optimization (which maximizes thrust) and robust design optimization (which minimizes discrepancy in thrust across all model variants), are shown in Fig. A.3. A similar plot for thrust discrepancy across iterations for the robust design optimization is shown in Fig. A.4. As can be seen, the interval-based robust design optimization optimizes to a slightly lower thrust compared to its deterministic counterpart. Note that the DUU-based optimization reduces the discrepancy from 60 N in the baseline geometry to around just 1 N in the corresponding optimized geometry. The optimized geometries for both the cases can also be seen in Fig. A.5 along with the baseline geometry. It can be observed that the robust

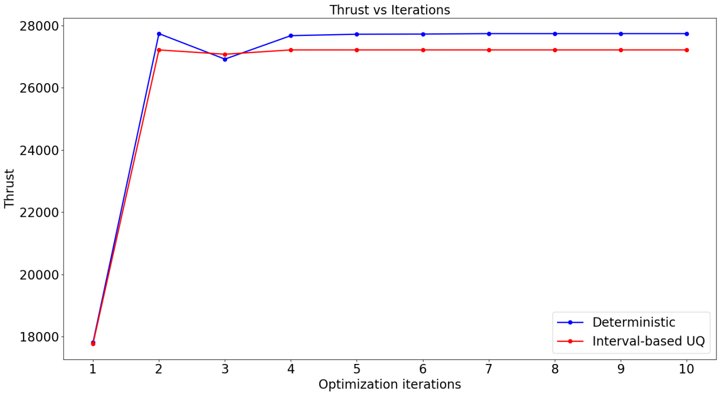


Figure A.3: Thrust values across iterations for both deterministic and DUU-based optimization without mass constraints

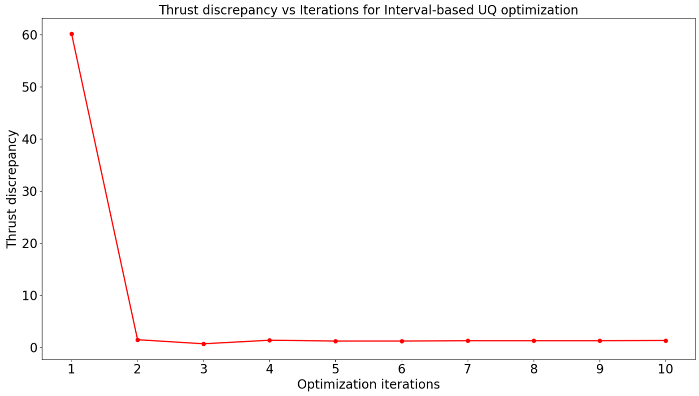


Figure A.4: Thrust discrepancy values across iterations for DUU-based optimization without mass constraints

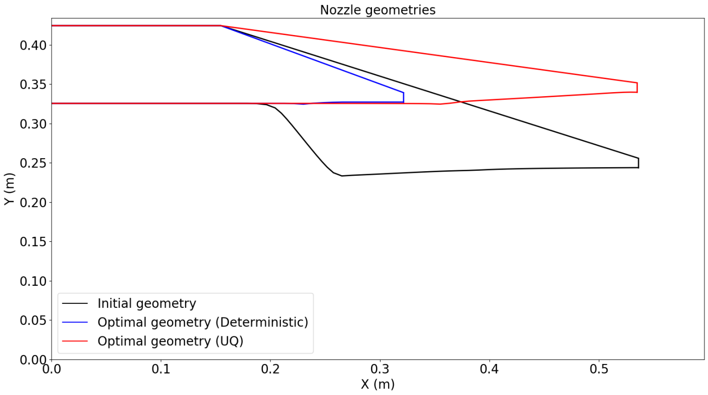
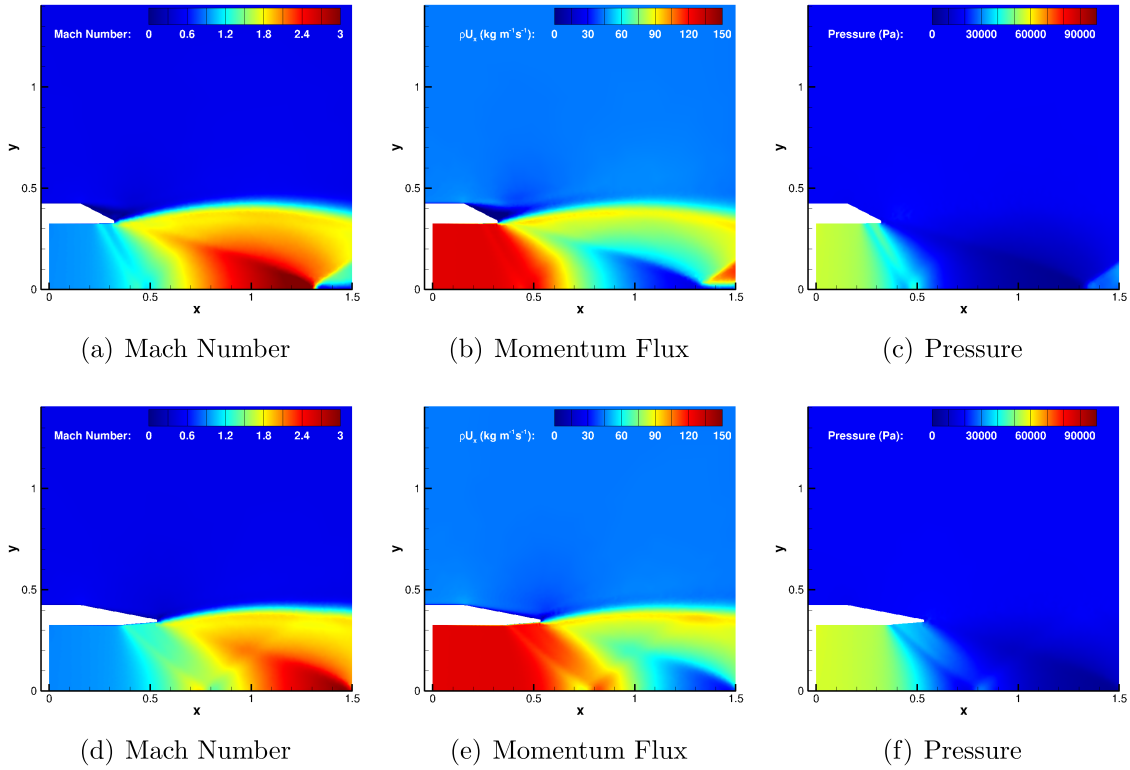


Figure A.5: Baseline and optimized geometries (when optimized without mass constraints)

design optimization results in a geometry which is considerably longer compared to the one optimized using its deterministic counterpart and hence has a considerably larger



(a) Mach Number (b) Momentum Flux (c) Pressure

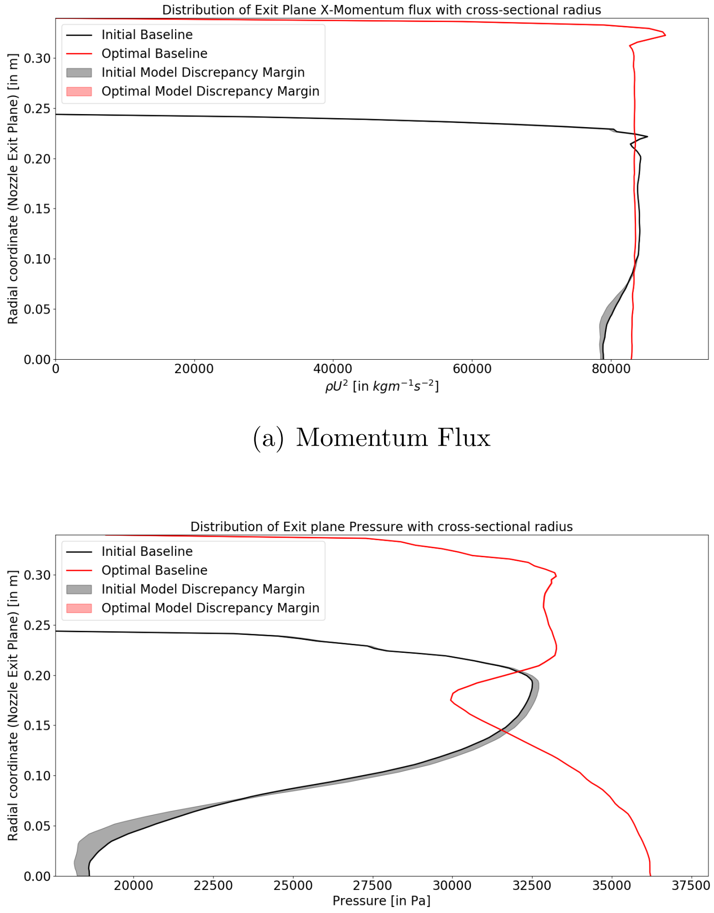
(d) Mach Number (e) Momentum Flux (f) Pressure

Figure A.6: Contours of Mach number, momentum flux and pressure for geometries obtained via deterministic optimization (top) and DUU (bottom) without mass constraints

volume and mass. The contours for Mach numbers, pressure and momentum flux for all the three geometries are shown in Fig. A.6. Plots comparing the momentum flux as well as pressure with respective interval-based uncertainty bounds along the radial coordinate at the nozzle exit plane for the baseline and DUU-optimized geometry are shown in Fig. A.7.

A.4.2 Optimization with mass constraints

Repeating both the deterministic and DUU-based optimizations under the added mass constraints resulted in the geometry shapes as shown in Fig. A.8. Note here that, with the additional mass constraint, the DUU-based optimization results in a volume (and hence mass) of the nozzle which comparable to the one obtained using deterministic optimization, unlike the previous case. This can be seen in the values of mass across optimization iterations for both the optimization types as shown in Fig. A.9, where the optimal geometries have weights of around 60 kg and 70 kg for the deterministic



(a) Momentum Flux

(b) Pressure

Figure A.7: Comparing baseline and optimized (DUU without mass constraints) properties at the nozzle exit plane with uncertainty bounds

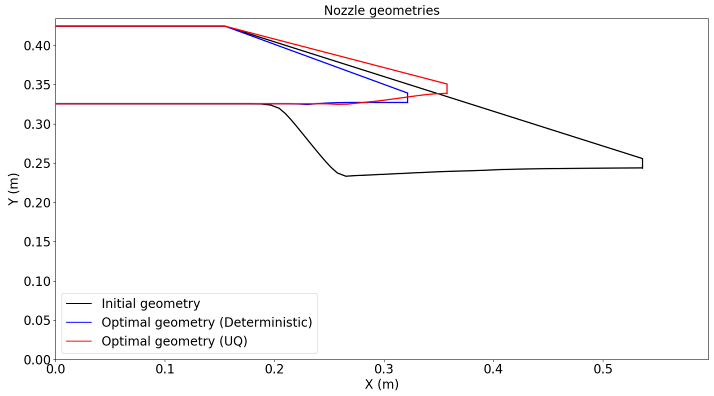


Figure A.8: Baseline and optimized geometries (when optimized with mass constraints)

optimization and DUU, respectively. The corresponding values of thrust and thrust discrepancies across optimization iterations are plotted in figs. A.10 and A.11. As can

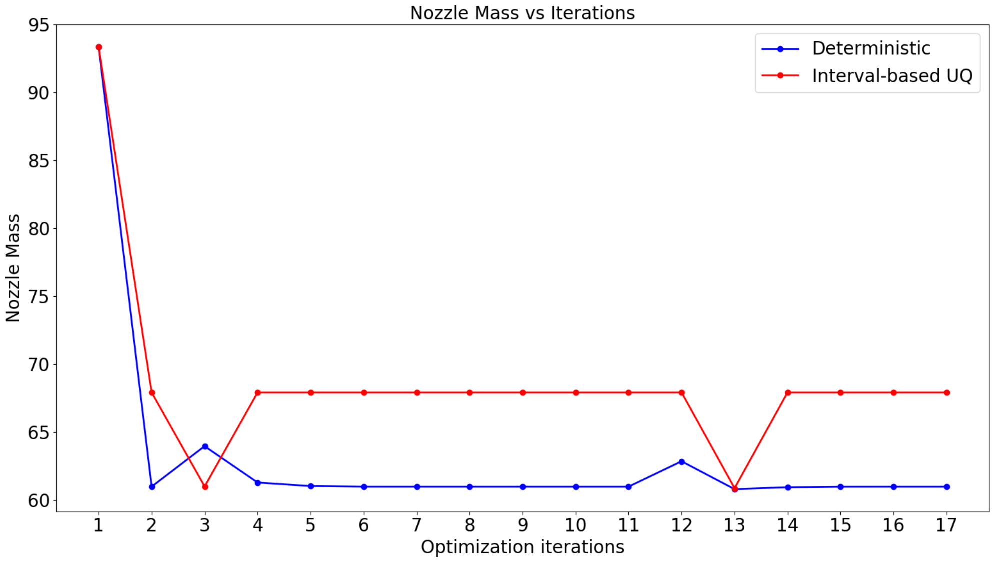


Figure A.9: Mass corresponding to iterates of nozzle geometry

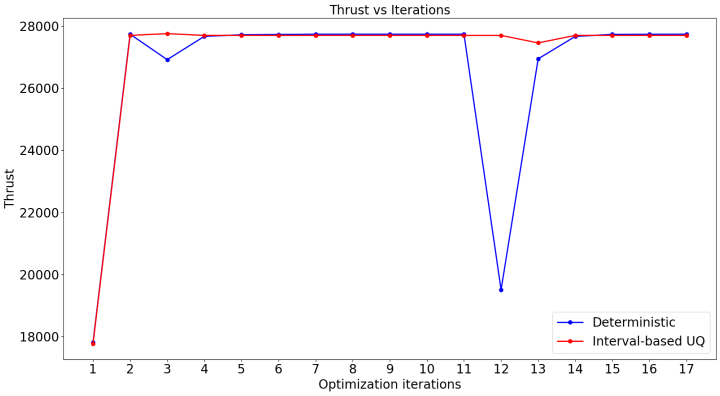


Figure A.10: Thrust values across iterations for both deterministic and DUU-based optimization with mass constraints

be seen here, the difference in thrust produced by the two optimal geometries is marginal compared to what it was without the use of mass constraints. Also, the discrepancy in the thrust values is reduced to around 1 N, similar to what was observed before. This indicates that there are multiple possible geometries with a minimal thrust discrepancy and the geometry which one obtains after the DUU-based optimization heavily depends on the constraints set during the formulation for the optimization problem. Finally, the contours for the geometries obtained using both the optimization techniques are shown in Fig. A.12 and the plots of momentum flux and pressure with their discrepancy bounds

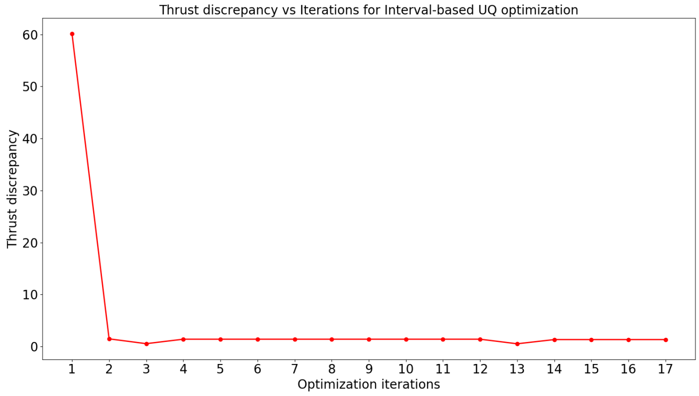
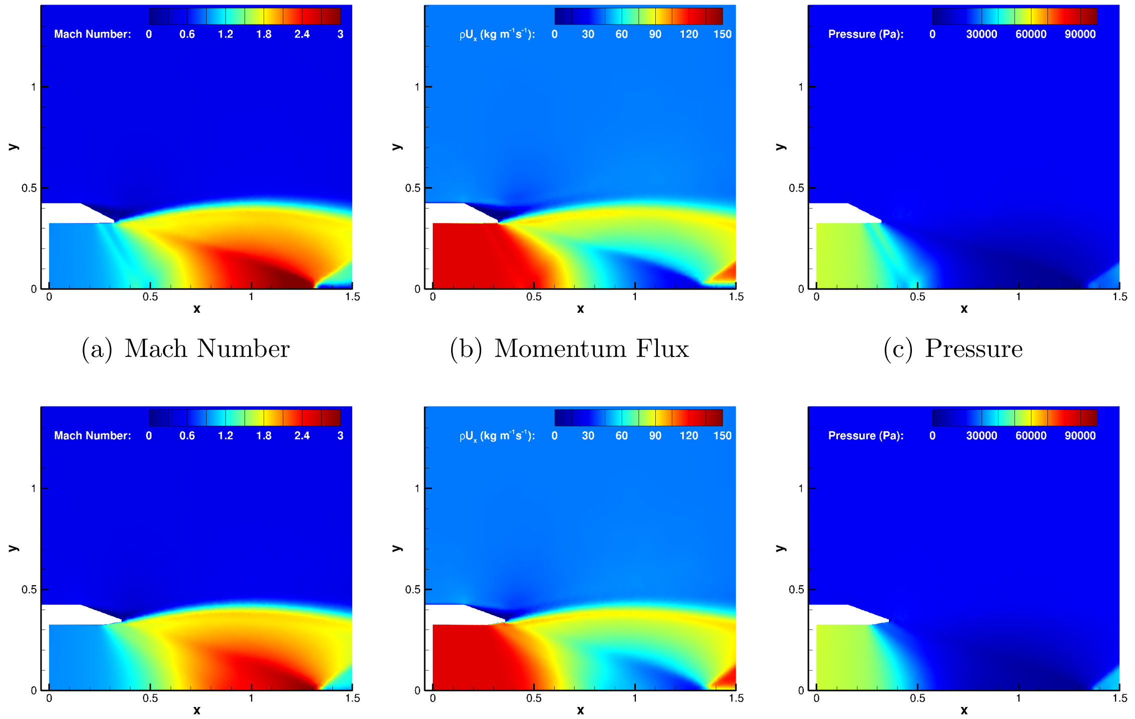


Figure A.11: Thrust discrepancy values across iterations for DUU-based optimization with mass constraints



(a) Mach Number

(b) Momentum Flux

(c) Pressure

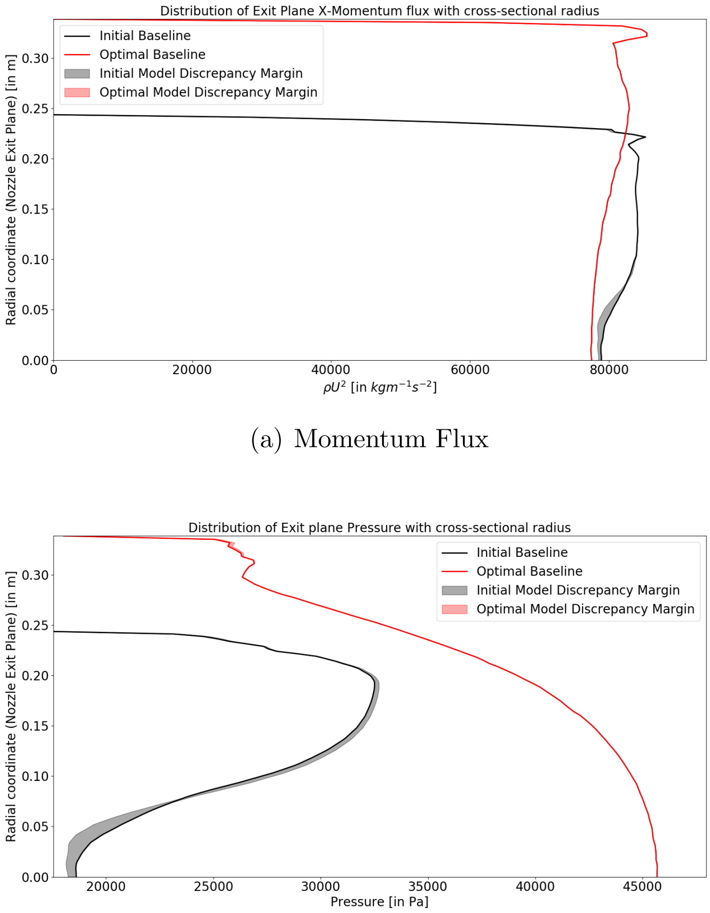
(d) Mach Number

(e) Momentum Flux

(f) Pressure

Figure A.12: Contours of Mach number, momentum flux and pressure for geometries obtained via deterministic optimization (top) and DUU (bottom) with mass constraints

along the radial coordinate at the nozzle exit plane for the baseline and DUU-optimized geometry are shown in Fig. A.13.



(a) Momentum Flux

(b) Pressure

Figure A.13: Comparing baseline and optimized (DUU with mass constraints) properties at the nozzle exit plane with uncertainty bounds

Appendix B

Sensitivity Evaluation via Method of Adjoints

B.1 A Simple Introduction

Given a system of PDE's represented as $\mathscr{R}(\boldsymbol{u}, \boldsymbol{\xi}) = 0$, consider an objective function $\mathcal{J}(\boldsymbol{u}, \boldsymbol{\xi})$ that needs to be minimized. Here, $\boldsymbol{u}$ refers to the states of the system and $\boldsymbol{\xi}$ refers to the inputs to the system. Since $\boldsymbol{u}$ is an implicit function of $\boldsymbol{\xi}$ (via the PDE system), the corresponding optimization problem is constrained and can be mathematically written as follows.

$$\underset{\boldsymbol{\xi}}{\text{minimize}}\ \mathcal{J}(\boldsymbol{u}, \boldsymbol{\xi}) \qquad s.t. \quad \mathscr{R}(\boldsymbol{u}, \boldsymbol{\xi}) = 0$$

The optimization techniques can be broadly classified into gradient-based and gradient-free. Gradient-free techniques are currently intractable for inputs which are high-dimensional which is the case when inferring model augmentations. Gradient-based techniques, however, as the name suggests, require the sensitivities $d\mathcal{J}/d\boldsymbol{\xi}$. A naive approach would be to evaluate finite difference approximations for each input variable. This would require as many function evaluations as there are inputs which could be tedious and time-consuming. However, the method of adjoints, which is explained as follows, offers a more efficient alternative. Since the states are implicit functions of the inputs, the sensitivity of $\mathscr{R}$ and $\mathcal{J}$ w.r.t. $\boldsymbol{\xi}$ can be written as follows.

$$\frac{d\mathscr{R}}{d\boldsymbol{\xi}} = \frac{\partial\mathscr{R}}{\partial\boldsymbol{\xi}} + \frac{\partial\mathscr{R}}{\partial\boldsymbol{u}}\frac{d\boldsymbol{u}}{d\boldsymbol{\xi}}$$

$$\frac{d\mathcal{J}}{d\boldsymbol{\xi}} = \frac{\partial\mathcal{J}}{\partial\boldsymbol{\xi}} + \frac{\partial\mathcal{J}}{\partial\boldsymbol{u}}\frac{d\boldsymbol{u}}{d\boldsymbol{\xi}}$$

Under the constraint that the residuals need to be zero, one can write

$$\frac{d\mathscr{R}}{d\boldsymbol{\xi}} = \frac{\partial\mathscr{R}}{\partial\boldsymbol{\xi}} + \frac{\partial\mathscr{R}}{\partial\boldsymbol{u}}\frac{d\boldsymbol{u}}{d\boldsymbol{\xi}} = 0$$

which implies,

$$\frac{d\boldsymbol{u}}{d\boldsymbol{\xi}} = -\left[\frac{\partial\mathscr{R}}{\partial\boldsymbol{u}}\right]^{-1}\frac{\partial\mathscr{R}}{\partial\boldsymbol{\xi}}$$

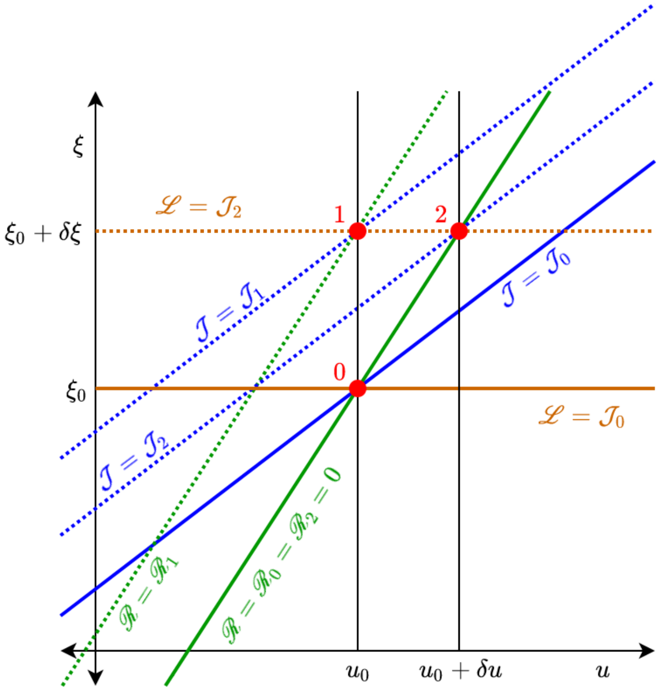
Substituting this value into the expression for $d\mathcal{J}/d\boldsymbol{\xi}$, we have the following expression

$$\frac{d\mathcal{J}}{d\boldsymbol{\xi}} = \frac{\partial\mathcal{J}}{\partial\boldsymbol{\xi}} - \frac{\partial\mathcal{J}}{\partial\boldsymbol{u}}\left[\frac{\partial\mathscr{R}}{\partial\boldsymbol{u}}\right]^{-1}\frac{\partial\mathscr{R}}{\partial\boldsymbol{\xi}}$$

The term $-\frac{\partial\mathcal{J}}{\partial\boldsymbol{u}}\left[\frac{\partial\mathscr{R}}{\partial\boldsymbol{u}}\right]^{-1}$ is referred to as ψ^T, i.e., the transpose of the adjoint vector. Note here that only a single matrix-vector linear system is needed to be solved to obtain ψ as $\mathcal{J}$ is a scalar quantity. On the other hand, if $\left[\frac{\partial\mathscr{R}}{\partial\boldsymbol{u}}\right]^{-1}\frac{\partial\mathscr{R}}{\partial\boldsymbol{\xi}}$ is solved instead, it would require solving as many matrix-vector systems as there are inputs. Clearly, solving for ψ is more efficient. The partial derivatives involved in the sensitivity calculation can be calculated using one of the following methods: finite differences, complex-step differentiation and forward- or reverse-mode algorithmic differentiation (a.k.a. automatic differentiation or AD). In the transition modeling work presented in this thesis, reverse-mode AD is used from a readily available computational package called ADOL-C [25].

B.2 The Adjoint Vector as a Lagrange Multiplier: A Geometric Interpretation

Given $\boldsymbol{\xi}_0$ and $\boldsymbol{u}_0$ such that $\mathscr{R}(\boldsymbol{u}_0, \boldsymbol{\xi}_0) = 0$, $\mathscr{R}$ and $\mathcal{J}$ can be approximated by their linearized versions in the immediate neighborhood of $(\boldsymbol{u}_0, \boldsymbol{\xi}_0)$ in the $\boldsymbol{u}$-$\boldsymbol{\xi}$ space. Fig. B.1 shows a representative set of contours for $\mathscr{R}$ and $\mathcal{J}$ for unidimensional $\boldsymbol{u}$ and $\boldsymbol{\xi}$ vectors in such a neighborhood, along with a naive evaluation of sensitivities under the linear approximation. The idea of using Lagrangian multipliers involves creating a function $\mathscr{L} = \mathcal{J} + \lambda\mathscr{R}$ that depends only on $\boldsymbol{\xi}$ in this neighborhood, i.e., it stays constant when $\boldsymbol{u}$ is changed. Also note that, by design, for any given $\boldsymbol{\xi}$, the value of $\mathscr{L}$ will be exactly


ξ
$\xi_0 + \delta\xi$
ξ_0
$\mathscr{L} = \mathscr{J}_2$
1
2
0
$\mathscr{J} = \mathscr{J}_1$
$\mathscr{J} = \mathscr{J}_0$
$\mathscr{J} = \mathscr{J}_2$
$\mathscr{L} = \mathscr{J}_0$
$\mathscr{R} = \mathscr{R}_1$
$\mathscr{R} = \mathscr{R}_0 = \mathscr{R}_2 = 0$
u_0
$u_0 + \delta u$
u

Starting at point 0:
$\mathscr{J}_0 = \mathscr{J}(u_0, \xi_0)$ and $\mathscr{R}_0 = \mathscr{R}(u_0, \xi_0) = 0$

Perturbing ξ to reach point 1:
$\mathscr{J}_1 = \mathscr{J}_0 + \dfrac{\partial \mathscr{J}}{\partial \xi}\delta\xi$ and $\mathscr{R}_1 = \dfrac{\partial \mathscr{R}}{\partial \xi}\delta\xi$

Finding δu to compensate for non-zero $\mathscr{R}_1$:
$$\mathscr{R}_2 = \mathscr{R}_1 + \frac{\partial \mathscr{R}}{\partial u}\delta u = \frac{\partial \mathscr{R}}{\partial \xi}\delta\xi + \frac{\partial \mathscr{R}}{\partial u}\delta u = 0$$
$$\Rightarrow \delta u = -\left[\frac{\partial \mathscr{R}}{\partial u}\right]^{-1} \frac{\partial \mathscr{R}}{\partial \xi}\delta\xi$$

Finding the objective function at point 2:
$$\mathscr{J}_2 = \mathscr{J}_1 + \frac{\partial \mathscr{J}}{\partial u}\delta u$$
$$\Rightarrow \mathscr{J}_2 = \mathscr{J}_0 + \left(\frac{\partial \mathscr{J}}{\partial \xi} - \frac{\partial \mathscr{J}}{\partial u}\left[\frac{\partial \mathscr{R}}{\partial u}\right]^{-1}\frac{\partial \mathscr{R}}{\partial \xi}\right)\delta\xi$$

Figure B.1: Schematic showing linearized contours of the $\mathscr{R}$, $\mathscr{J}$, and $\mathscr{L}$ functions in a small neighborhood in the u-ξ space along with a naive sensitivity evaluation method

the same as $\mathscr{J}$ when $\mathscr{R} = 0$. Hence, $\dfrac{\partial \mathscr{L}}{\partial \boldsymbol{\xi}}$ must be the same as $\dfrac{d\mathscr{J}}{d\boldsymbol{\xi}}$. The only task which remains now is to find the appropriate numerical values that constitute the Lagrangian multiplier λ. This is trivial as we only need to impose that $\dfrac{\partial \mathscr{L}}{\partial \boldsymbol{u}} = 0$ resulting in the expression as follows.

$$\frac{\partial \mathscr{J}}{\partial \boldsymbol{u}} + \lambda\frac{\partial \mathscr{R}}{\partial \boldsymbol{u}} = 0 \qquad \Rightarrow \qquad \lambda = -\frac{\partial \mathscr{J}}{\partial \boldsymbol{u}}\left[\frac{\partial \mathscr{R}}{\partial \boldsymbol{u}}\right]^{-1}$$

As can be seen, $\lambda = \psi^T$. Hence, geometrically speaking, the adjoint vector (or the Lagrange multiplier) simply consists of the scaling factors needed to ensure that the collective variation of $\mathscr{R}$ w.r.t. $\boldsymbol{u}$ cancels the variation of $\mathscr{J}$ with $\boldsymbol{u}$. In other words, the residuals when scaled with Lagrange multipliers and added to the objective function, rotate the contour lines of $\mathscr{J}$ about the points where they intersect $\mathscr{R} = 0$ resulting in contour lines for $\mathscr{L}$ which run parallel to the $\boldsymbol{u}$-axis (as shown in Fig. B.1).

Appendix C

Supplementary Results for the Transition Model

C.1 Results when training only with T3A

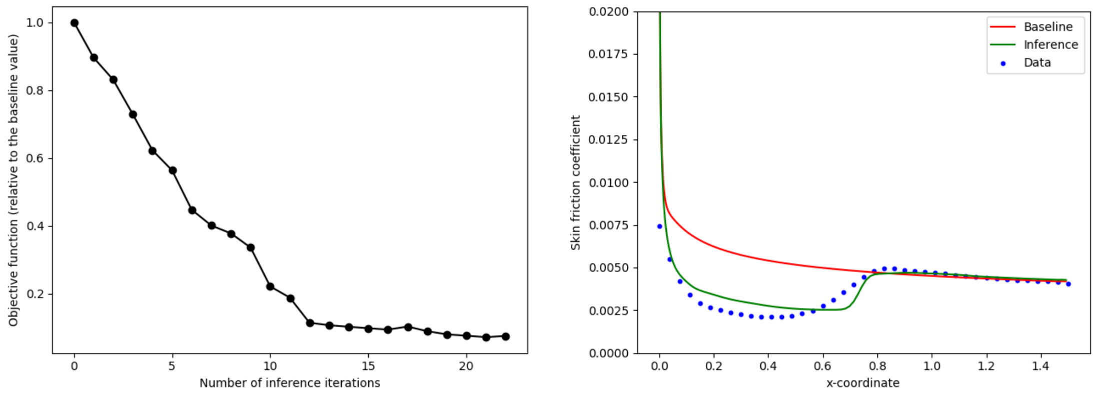
The main issue with training only with T3A is that the behavior in the augmentation is learnt only on the basis of data from a zero pressure gradient case. The more cases are added to the mix, the stronger is the consistency of an optimal augmentation for different problems.

C.1.1 Optimization convergence and feature space contours

Looking at Fig. C.1, although the convergence of the objective function appears similar to that observed when both T3A and T3C1 were simultaneously used to learn the augmentation, the skin friction predictions and augmentation contours in the feature space (Fig. C.2) have significant differences between the two cases. The effect of this difference between the two augmentations can be seen in the following sections. It is however, noteworthy how a single additional case with unseen physical behavior can make a considerable difference in the predictions. This suggests the ability of the framework to effectively extract information about the features-to-augmentation functional relationship.

C.1.2 Predictions on T3 cases

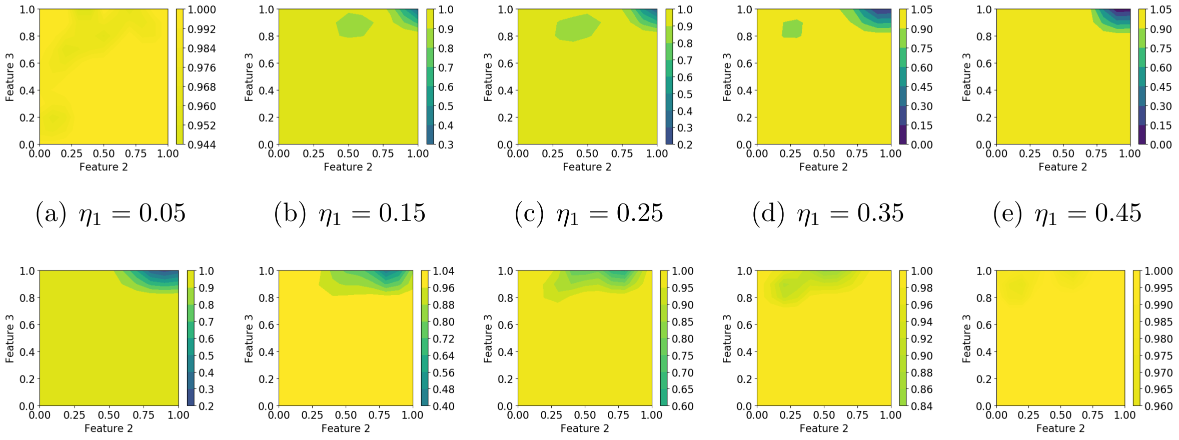
It can be seen in Fig. C.3 that the prediction trained on T3A alone is completely wrong for T3C1 and T3B, while it appears reasonable for the other cases. Another important



(a) Optimization convergence

(b) Skin friction prediction

Figure C.1: Augmentation training using data from T3A only



(a) $\eta_1 = 0.05$

(b) $\eta_1 = 0.15$

(c) $\eta_1 = 0.25$

(d) $\eta_1 = 0.35$

(e) $\eta_1 = 0.45$

(f) $\eta_1 = 0.55$

(g) $\eta_1 = 0.65$

(h) $\eta_1 = 0.75$

(i) $\eta_1 = 0.85$

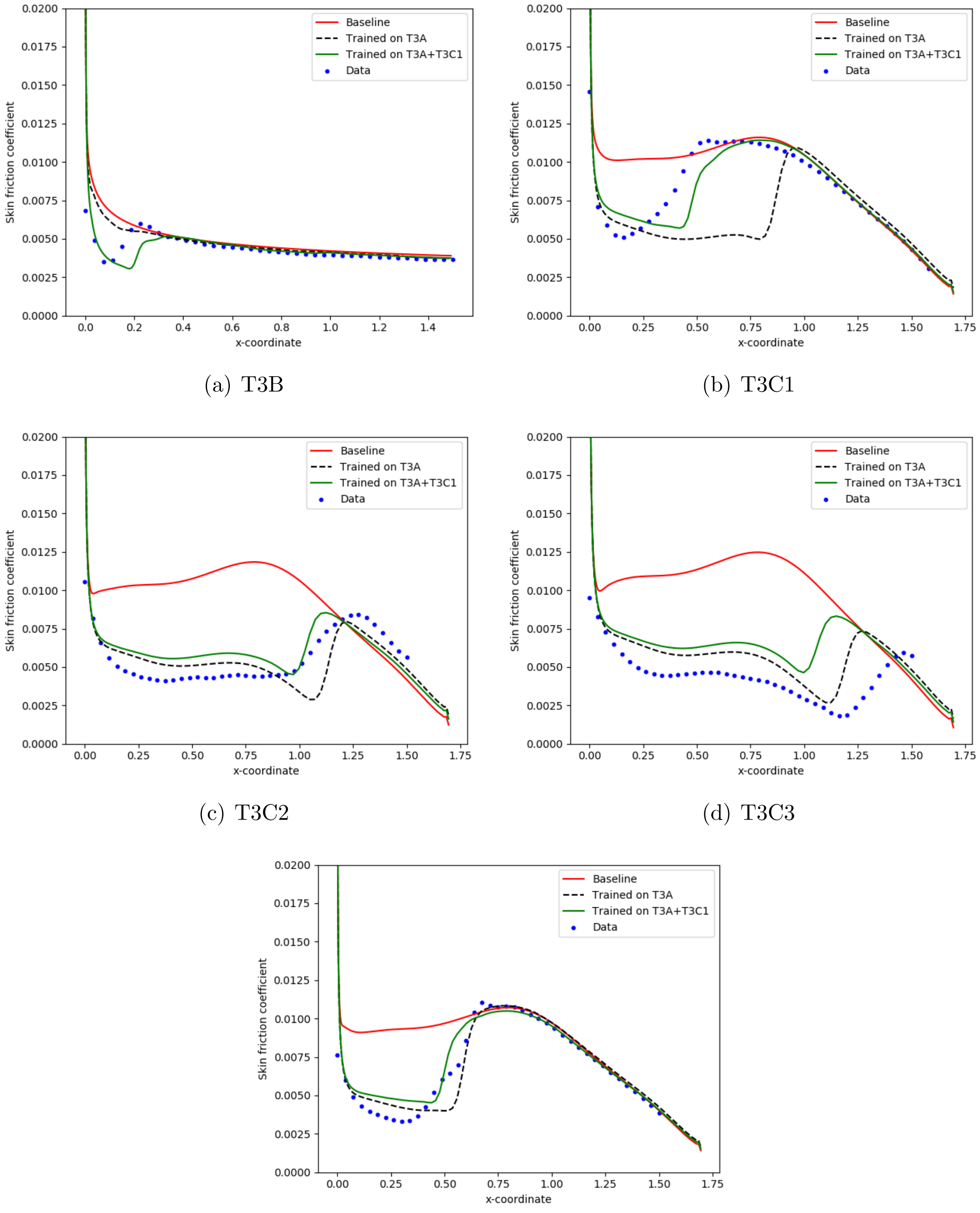
(j) $\eta_1 = 0.95$

Figure C.2: Feature maps (x-axis: η_2, y-axis: η_3, uniform color-bar range [0,1] across all plots)

observation is that the transition locations for T3C1, T3C2 and T3C3 are over-predicted. This is due to the fact that the augmentation remains at a lower value for a longer distance spuriously as the augmentation has no way of differentiating between how η_1 changes for different pressure gradients as the training is performed only for a zero pressure gradient case.

C.1.3 Prediction on VKI cases

As can be seen in Fig. C.4, training a model only on the T3A data results in significantly inaccurate transition location predictions on at least one side of the blade except MUR224 when compared to the results presented in the main text where both T3A and T3C1 cases



(a) T3B

(b) T3C1

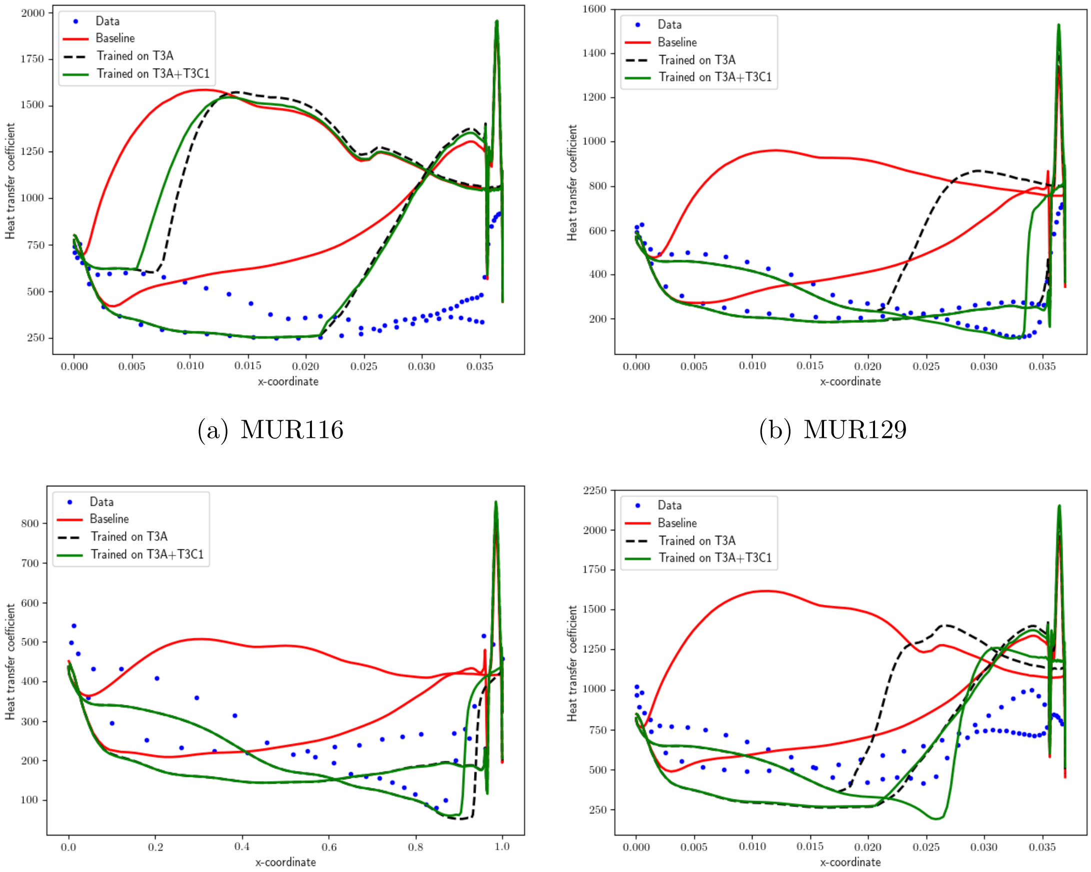
(c) T3C2

(d) T3C3

(e) T3C5

Figure C.3: Skin friction coefficients for T3C cases

were used for training. This is in accordance with the explanation provided in the last section. Since the transition model has little information on how the features behave in the presence of non-zero pressure gradients, additional data which can highlight such behavior (in the main text as the T3C1 case) is required to extract information about the behavior of the augmentation in feature space.



(a) MUR116

(b) MUR129

(c) MUR224

(d) MUR241

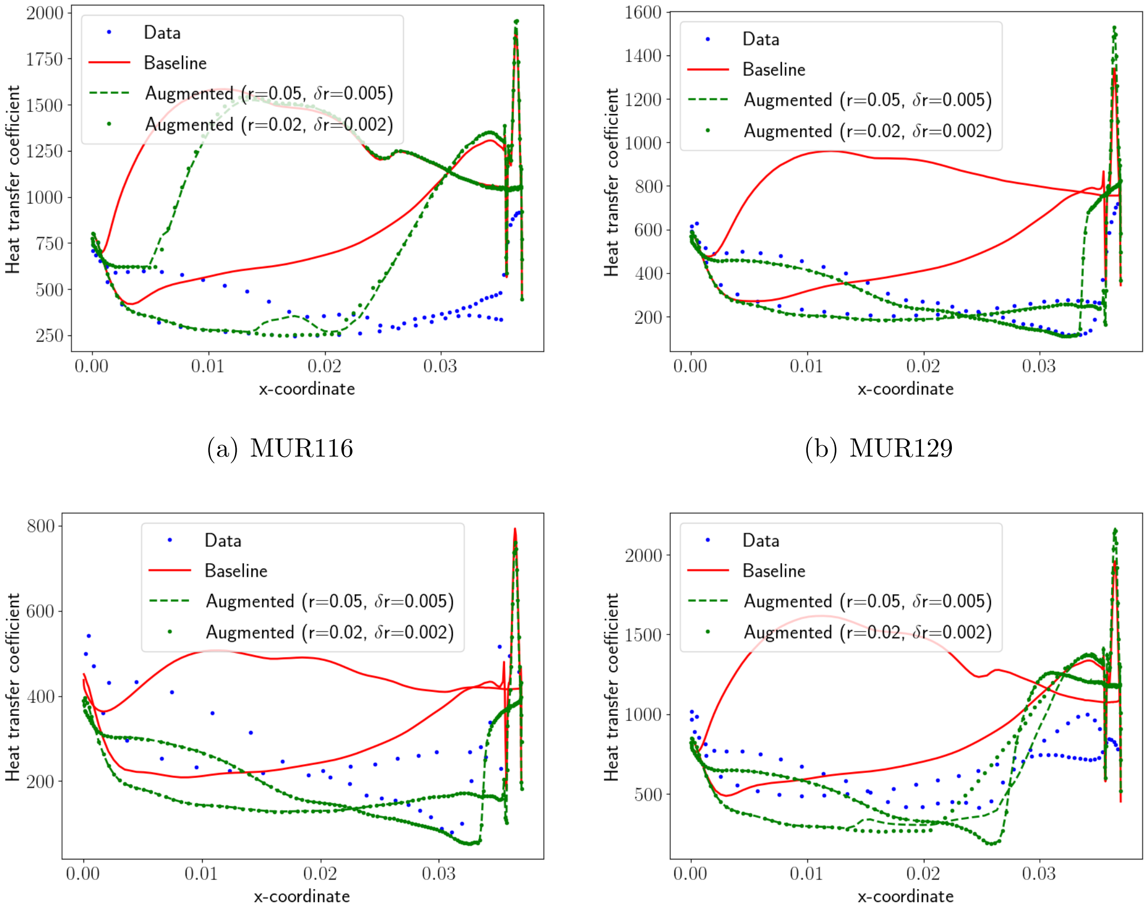
Figure C.4: Heat transfer coefficients for VKI cases

C.2 Effect of changing the user-specified freestream distance

As shown in Fig. C.5, we found that varying the user-specified wall distance used to extract freestream quantities to calculate η_1 usually has a small effect on the predictions for the turbine blades. A minor discrepancy is observed in MUR116 (characterized by a small bump in the heat transfer coefficient), whereas a major discrepancy, resulting in considerable different transition behavior, is seen for MUR241.

C.3 Effect of using a finer discretization in feature-space

For comparison purposes, a finer feature-space discretization was also used to obtain the augmentation function, the training results and augmentation contours on feature-space



(a) MUR116

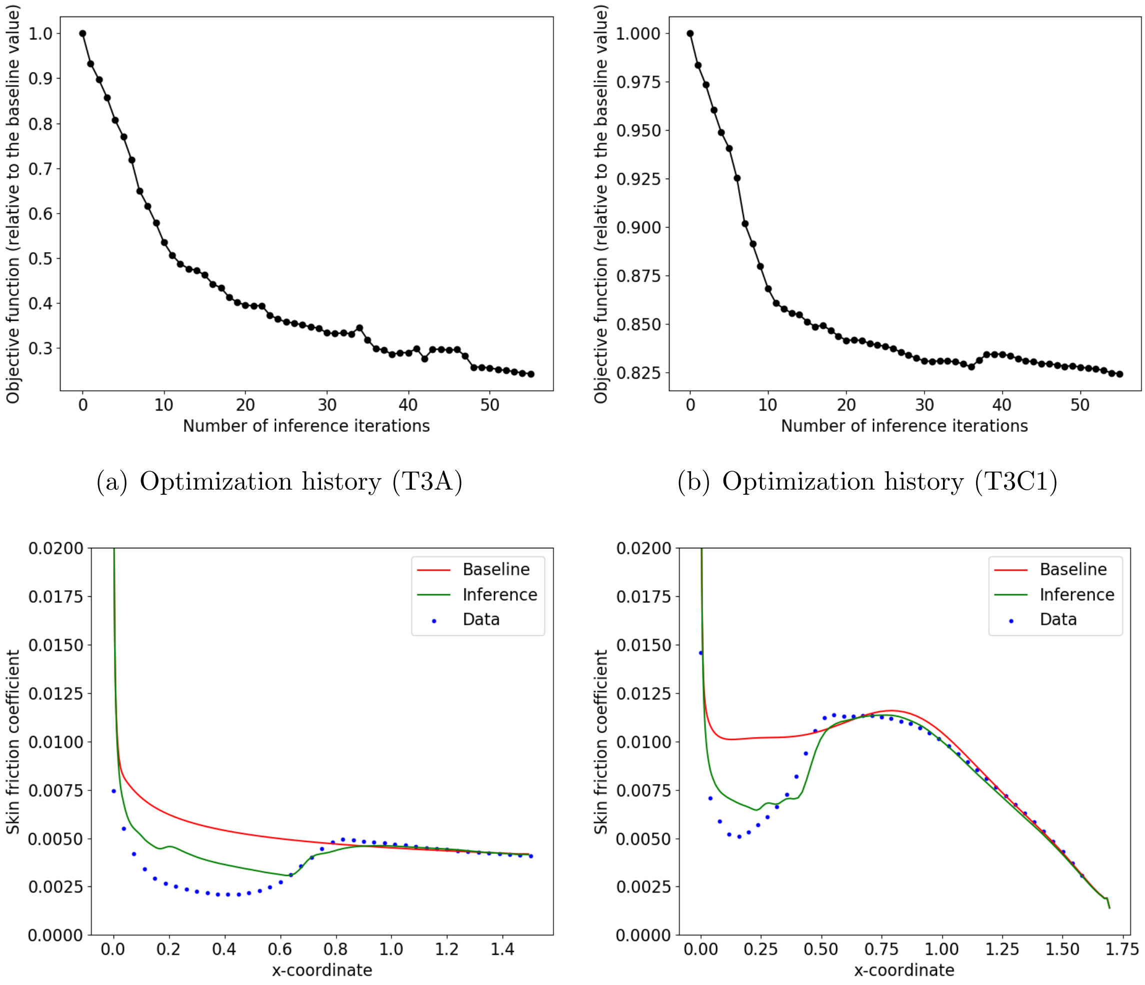
(b) MUR129

(c) MUR224

(d) MUR241

Figure C.5: Comparison between different preset distance intervals

slices for which are shown in Figs. C.6 and C.7. The feature space was divided into subdomains of size 1/30 along all three feature space directions (90, 30, and 30 cells along the first, second, and third features respectively). As can be noticed in Fig. C.7, the influence of the changes made by the data have been restricted to smaller regions owing to the smaller cell sizes. Figs. C.8 and C.9 show the results from testing the augmentation on the T3B, T3C2, T3C3, T3C5, MUR116, MUR129, MUR224, and MUR241 cases. As can be seen from the results, while the augmentation learned on the finer grid seems to predict the transition locations for T3 cases with nearly similar accuracy (with little laminarization in the T3B case and slightly premature transition in T3C5) as its counterpart trained on the coarser grid, almost all results from the VKI cases exhibit premature transition. This happens because the limited region of influence that the available data has in the feature space allow some feature space locations to exhibit baseline behavior (which did not happen when using the augmentation learned on a coarser grid as the region of influence covered a larger part of the feature space).



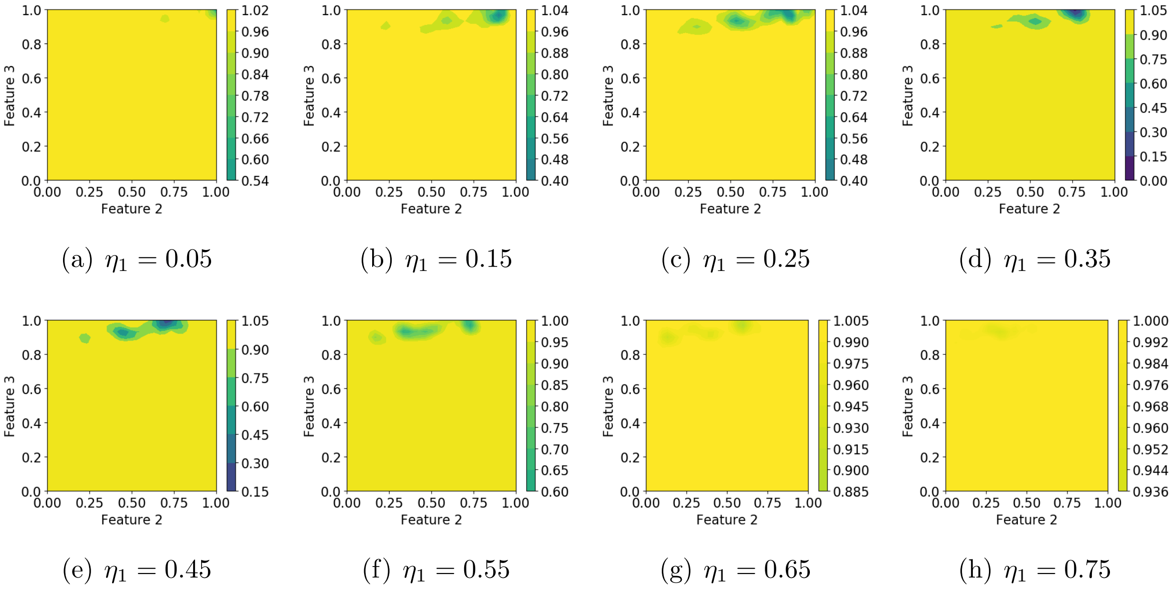
(a) Optimization history (T3A)

(b) Optimization history (T3C1)

(c) Skin friction profile (T3A)

(d) Skin friction profile (T3C1)

Figure C.6: Training results on a finer feature-space grid



(a) $\eta_1 = 0.05$

(b) $\eta_1 = 0.15$

(c) $\eta_1 = 0.25$

(d) $\eta_1 = 0.35$

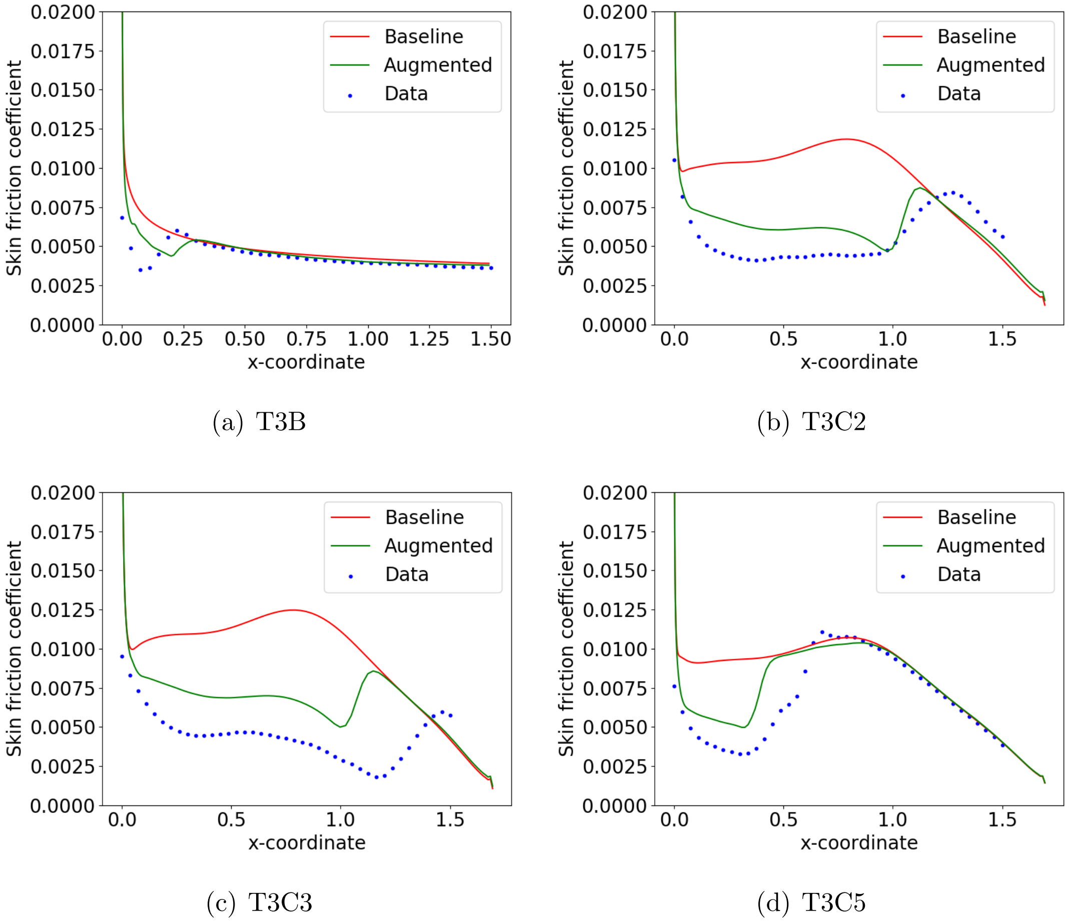
(e) $\eta_1 = 0.45$

(f) $\eta_1 = 0.55$

(g) $\eta_1 = 0.65$

(h) $\eta_1 = 0.75$

Figure C.7: Augmentation contours on feature-space slices when using a fine discretization



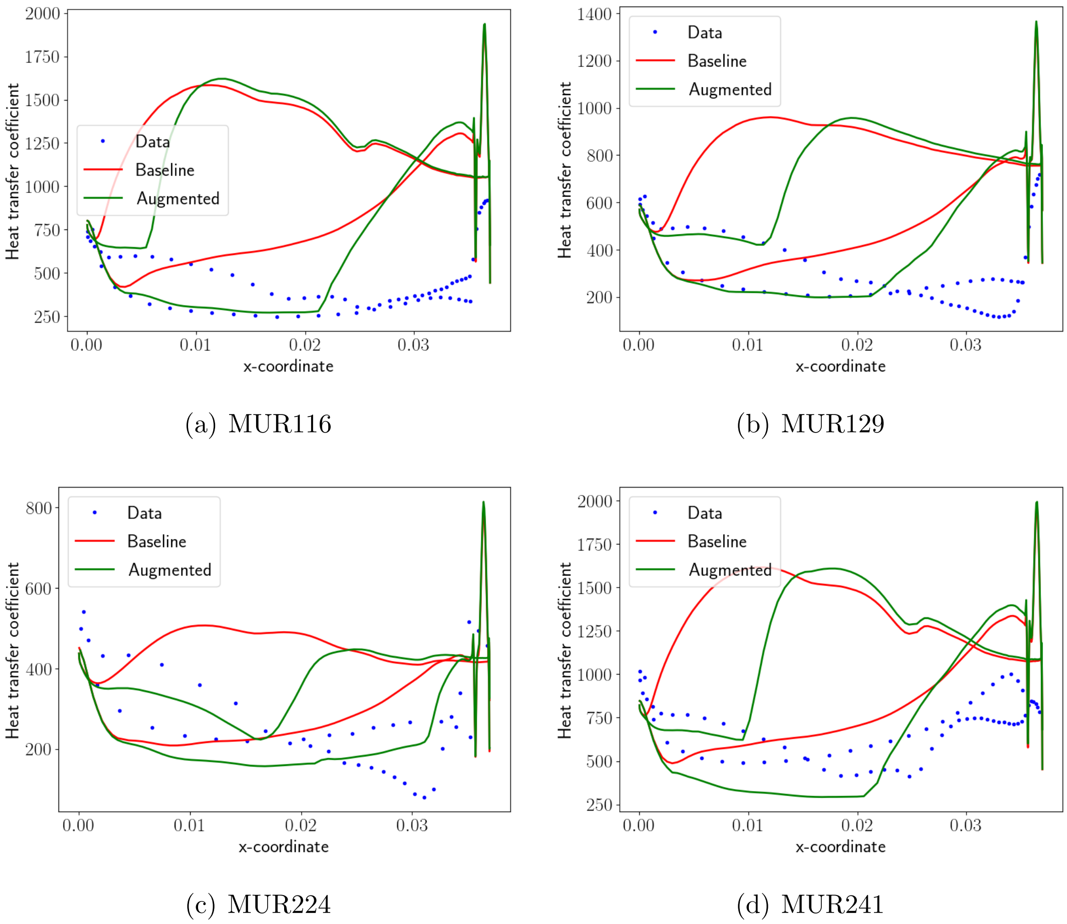
(a) T3B

(b) T3C2

(c) T3C3

(d) T3C5

Figure C.8: Skin friction coefficient profiles for the T3 test cases obtained from augmentation inferred on a finely-discretized feature-space



(a) MUR116

(b) MUR129

(c) MUR224

(d) MUR241

Figure C.9: Heat transfer coefficient profiles for the VKI test cases obtained from the augmentation inferred on a finely-discretized feature-space

Bibliography

[1] B. J. Abu-Ghannam and R. Shaw. Natural transition of boundary layers—the effects of turbulence, pressure gradient, and flow history. Journal of Mechanical Engineering Science, 22(5):213–228, 1980.

[2] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. Casadi: a software framework for nonlinear optimization and optimal control. Mathematical Programming Computation, 11(1):1–36, 2019.

[3] P. D. Arendt, D. W. Apley, and W. Chen. Quantification of Model Uncertainty: Calibration, Model Discrepancy, and Identifiability. Journal of Mechanical Design, 134(10), 09 2012. ISSN 1050-0472. doi: 10.1115/1.4007390. URL `https://doi.org/10.1115/1.4007390`. 100908.

[4] M. Arif, S. C. Cheung, and J. Andrews. Different approaches used for modeling and simulation of polymer electrolyte membrane fuel cells: a review. Energy & Fuels, 34 (10):11897–11915, 2020.

[5] B. Baldwin and T. Barth. A one-equation turbulence transport model for high reynolds number wall-bounded flows. In 29th Aerospace Sciences Meeting, page 610, 1991. doi: 10.2514/6.1991-610. URL `https://arc.aiaa.org/doi/abs/10.2514/6.1991-610`.

[6] B. Baldwin and H. Lomax. Thin-layer approximation and algebraic model for separated turbulent flows. In 16th Aerospace Sciences Meeting, page 257, 1978. doi: 10.2514/6.1978-257. URL `https://arc.aiaa.org/doi/abs/10.2514/6.1978-257`.

[7] R.-D. Cécora, B. Eisfeld, A. Probst, S. Crippa, and R. Radespiel. Differential reynolds stress modeling for aeronautics. In 50th AIAA Aerospace Sciences Meeting including the New Horizons forum and aerospace exposition, page 465, 2012.

[8] S. H. Cheung, T. A. Oliver, E. E. Prudencio, S. Prudhomme, and R. D. Moser. Bayesian uncertainty analysis with applications to turbulence modeling. Reliability Engineering & System Safety, 96(9):1137–1149, 2011.

[9] F. Chollet et al. Keras. `https://keras.io`, 2015.

[10] P. Y. CHOU. On velocity correlations and the solutions of the equations of turbulent fluctuation. Quarterly of Applied Mathematics, 3(1):38–54, 1945. ISSN 0033569X, 15524485. URL `http://www.jstor.org/stable/43633490`.

[11] W. Daud, R. Rosli, E. Majlan, S. Hamid, R. Mohamed, and T. Husaini. Pem fuel cell system control: A review. Renewable Energy, 113:620–638, 2017.

[12] S. Dhawan and R. Narasimha. Some properties of boundary layer flow during the transition from laminar to turbulent motion. Journal of Fluid Mechanics, 3(4): 418–436, 1958. doi: 10.1017/S0022112058000094.

[13] E. Dow and Q. Wang. Quantification of structural uncertainties in the k -w turbulence model. In 52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 2011. doi: 10.2514/6.2011-1762. URL `https://arc.aiaa.org/doi/abs/10.2514/6.2011-1762`.

[14] K. Duraisamy. Perspectives on machine learning-augmented reynolds-averaged and large eddy simulation models of turbulence. Physical Review Fluids, 2021.

[15] K. Duraisamy and P. Durbin. Transition modeling using data driven approaches. In Proc. CTR Summer Program, Stanford University, pages 427–434, 2014. URL `https://web.stanford.edu/group/ctr/Summer/SP14/09_Large-eddy_simulation/07_duraisamy.pdf`.

[16] P. Durbin. An intermittency model for bypass transition. International Journal of Heat and Fluid Flow, 36:1–6, 2012.

[17] W. Edeling, P. Cinnella, and R. P. Dwight. Predictive rans simulations via bayesian model-scenario averaging. Journal of Computational Physics, 275:65–91, 2014.

[18] H. W. EMMONS. The laminar-turbulent transition in a boundary layer-part i. Journal of the Aeronautical Sciences, 18(7):490–498, 1951. doi: 10.2514/8.2010. URL `https://doi.org/10.2514/8.2010`.

[19] H. W. Emmons. Shear flow turbulence. In Journal of Applied Mechanics-Transactions of the ASME, volume 21, pages 283–283. ASME-AMER SOC MECHANICAL ENG 345 E 47TH ST, NEW YORK, NY 10017, 1954.

[20] L. Franceschini, D. Sipp, and O. Marquet. Mean-flow data assimilation based on minimal correction of turbulence models: Application to turbulent high reynolds number backward-facing step. Phys. Rev. Fluids, 5:094603, Sep 2020. doi: 10.1103/PhysRevFluids.5.094603. URL `https://link.aps.org/doi/10.1103/PhysRevFluids.5.094603`.

[21] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55, 1997.

[22] X. Ge. A bypass transition model based on the intermittency function. PhD thesis, 2015.

[23] X. Ge, S. Arolla, and P. Durbin. A bypass transition model based on the intermittency function. Flow, turbulence and combustion, 93(1):37–61, 2014.

[24] A. Goshtasbi, B. L. Pence, J. Chen, M. A. DeBolt, C. Wang, J. R. Waldecker, S. Hirano, and T. Ersal. A mathematical model toward real-time monitoring of automotive pem fuel cells. Journal of The Electrochemical Society, 167(2):024518, 2020.

[25] A. Griewank, D. Juedes, and J. Utke. Algorithm 755: Adol-c: A package for the automatic differentiation of algorithms written in c/c++. ACM Transactions on Mathematical Software (TOMS), 22(2):131–167, 1996.

[26] S. Guillas, N. Glover, and L. Malki-Epshtein. Bayesian calibration of the constants of the k–ε turbulence model for a cfd model of street canyon flow. Computer methods in applied mechanics and engineering, 279:536–553, 2014.

[27] A. Gulyaev, V. Kozlov, and A. Sekundov. A universal one-equation model for turbulent viscosity. Fluid Dynamics, 28(4):485–494, 1993. doi: 10.1007/BF01342683. URL https://doi.org/10.1007/BF01342683.

[28] I.-S. Han and C.-B. Chung. Performance prediction and analysis of a pem fuel cell operating on pure oxygen using data-driven models: A comparison of artificial neural network and support vector machine. International Journal of Hydrogen Energy, 41 (24):10202–10211, 2016.

[29] X. Han, M. Rahman, and R. K. Agarwal. Development and application of wall-distance-free wray-agarwal turbulence model (wa2018). In 2018 AIAA Aerospace Sciences Meeting, page 0593, 2018. doi: 10.2514/6.2018-0593. URL https://arc.aiaa.org/doi/abs/10.2514/6.2018-0593.

[30] K. Hanjalic. Two-dimensional asymmetric turbulent flow in ducts. PhD thesis, University of London, 1970.

[31] J. R. Holland, J. D. Baeder, and K. Duraisamy. Field inversion and machine learning with embedded neural networks: Physics-consistent neural network training. AIAA Aviation 2019 Forum, 2019. doi: 10.2514/6.2019-3200. URL https://arc.aiaa.org/doi/abs/10.2514/6.2019-3200.

[32] J. R. Holland, J. D. Baeder, and K. Duraisamy. Towards integrated field inversion and machine learning with embedded neural networks for rans modeling. AIAA Scitech 2019 Forum, 2019. doi: 10.2514/6.2019-1884. URL https://arc.aiaa.org/doi/abs/10.2514/6.2019-1884.

[33] M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63(3):425–464, 2001.

[34] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[35] A. N. Kolmogorov. Equations of turbulent motion in an incompressible fluid. In Dokl. Akad. Nauk SSSR, volume 30, pages 299–303, 1941.

[36] R. Langtry and S. Sjolander. Prediction of transition for attached and separated shear layers in turbomachinery. In 38th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, page 3641, 2002.

[37] R. B. Langtry and F. R. Menter. Correlation-based transition modeling for unstructured parallelized computational fluid dynamics codes. AIAA journal, 47(12): 2894–2906, 2009.

[38] S. Lardeau, A. Fadai-Ghotbi, and M. Leschziner. Modelling bypass and separationinduced transition by reference to pre-transitional fluctuation energy. ERCOFTAC bulletin, 80:72–76, 2009.

[39] B. E. Launder and D. B. Spalding. The numerical computation of turbulent flows. In Numerical prediction of flow, heat transfer, turbulence and combustion, pages 96–116. Elsevier, 1983.

[40] Z. Li, R. Outbib, D. Hissel, and S. Giurgea. Data-driven diagnosis of pem fuel cell: A comparative study. Control Engineering Practice, 28:1–12, 2014.

[41] J. Ling, R. Jones, and J. Templeton. Machine learning strategies for systems with invariance properties. Journal of Computational Physics, 318:22 – 35, 2016. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2016.05.003. URL http://www.sciencedirect.com/science/article/pii/S0021999116301309.

[42] J. Ling, A. Kurzawski, and J. Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. Journal of Fluid Mechanics, 807:155–166, 2016.

[43] R. Ma, T. Yang, E. Breaz, Z. Li, P. Briois, and F. Gao. Data-driven proton exchange membrane fuel cell degradation predication through deep learning method. Applied energy, 231:102–115, 2018.

[44] R. E. Mayle. The 1991 IGTI Scholar Lecture: The Role of Laminar-Turbulent Transition in Gas Turbine Engines. Journal of Turbomachinery, 113(4):509–536, 10 1991. ISSN 0889-504X. doi: 10.1115/1.2929110. URL `https://doi.org/10.1115/1.2929110`.

[45] R. E. Mayle and A. Schulz. Heat Transfer Committee and Turbomachinery Committee Best Paper of 1996 Award: The Path to Predicting Bypass Transition. Journal of Turbomachinery, 119(3):405–411, 07 1997. ISSN 0889-504X. doi: 10.1115/1.2841138. URL `https://doi.org/10.1115/1.2841138`.

[46] F. Menter, T. Esch, and S. Kubacki. Transition modelling based on local variables. In Engineering Turbulence Modelling and Experiments 5, pages 555–564. Elsevier, 2002.

[47] G. Napoli, M. Ferraro, F. Sergi, G. Brunaccini, and V. Antonucci. Data driven models for a pem fuel cell stack performance prediction. International journal of hydrogen energy, 38(26):11628–11638, 2013.

[48] V. Nee and L. Kovasznay. The calculation of the incompressible turbulent boundary layer by a simple theory. Physics of Fluids, 12:473, 1968.

[49] A. Olabi, T. Wilberforce, and M. A. Abdelkareem. Fuel cell application in the automotive industry and future perspective. Energy, 214:118955, 2021.

[50] T. A. Oliver and R. D. Moser. Bayesian uncertainty quantification applied to RANS turbulence models. Journal of Physics: Conference Series, 318(4):042032,

dec 2011. doi: 10.1088/1742-6596/318/4/042032. URL `https://doi.org/10.1088/1742-6596/318/4/042032`.

[51] E. J. Parish and K. Duraisamy. A paradigm for data-driven predictive modeling using field inversion and machine learning. Journal of Computational Physics, 305: 758–774, 2016.

[52] S. Parneix, D. Laurence, and P. A. Durbin. A Procedure for Using DNS Databases. Journal of Fluids Engineering, 120(1):40–47, 03 1998. ISSN 0098-2202. doi: 10.1115/1.2819658. URL `https://doi.org/10.1115/1.2819658`.

[53] 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, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[54] T. Praisner and J. Clark. Predicting transition in turbomachinery: Part i—a review and new model development. In ASME Turbo Expo 2004: Power for Land, Sea, and Air, pages 161–174. American Society of Mechanical Engineers Digital Collection, 2004.

[55] L. Prandtl. 7. bericht über untersuchungen zur ausgebildeten turbulenz. ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik, 5(2):136–139, 1925. doi: https://doi.org/10.1002/zamm.19250050212. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/zamm.19250050212`.

[56] K. Priya, S. R. Choudhury, K. Sathish Kumar, and N. Rajasekar. Pem fuel cell modeling using genetic algorithm—a novel approach. In Advances in Smart Grid and Renewable Energy, pages 541–550. Springer, 2018.

[57] H. Raiesi, U. Piomelli, and A. Pollard. Evaluation of turbulence models using direct numerical and large-eddy simulation data. Journal of Fluids Engineering, 133(2), 2011.

[58] P. E. Roach and D. H. Brierley. The influence of a turbulent free-stream on zero pressure gradient transitional boundary layer development part 1: Test cases t3a and t3b. Numerical Simulation of Unsteady Flows and Transition to Turbulence, 3 (7):319–327, 2018.

[59] J. Rotta. Statistische theorie nichthomogener turbulenz. Zeitschrift für Physik, 129 (6):547–572, 1951.

[60] C. L. Rumsey, G. N. Coleman, and L. Wang. In search of data-driven improvements to rans models applied to separated flows. In AIAA SCITECH 2022 Forum, page 0937, 2022.

[61] I. B. H. Saïdi, M. Schmelzer, P. Cinnella, and F. Grasso. Cfd-driven symbolic identification of algebraic reynolds-stress models. Journal of Computational Physics, 457:111037, 2022.

[62] M. Schmelzer, R. P. Dwight, and P. Cinnella. Discovery of algebraic reynolds-stress models using sparse symbolic regression. Flow, Turbulence and Combustion, 104(2): 579–603, 2020.

[63] A. P. Singh. A framework to improve turbulence models using full-field inversion and machine learning. PhD thesis, 2018.

[64] A. P. Singh, K. Duraisamy, and Z. J. Zhang. Augmentation of turbulence models using field inversion and machine learning. 55th AIAA Aerospace Sciences Meeting, 2017. doi: 10.2514/6.2017-0993. URL https://arc.aiaa.org/doi/abs/10.2514/6.2017-0993.

[65] A. P. Singh, R. Matai, A. Mishra, K. Duraisamy, and P. A. Durbin. Data-driven augmentation of turbulence models for adverse pressure gradient flows. 23rd AIAA Computational Fluid Dynamics Conference, 2017. doi: 10.2514/6.2017-3626. URL https://arc.aiaa.org/doi/abs/10.2514/6.2017-3626.

[66] A. P. Singh, S. Medida, and K. Duraisamy. Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. AIAA Journal, 55(7):2215–2227, 2017. doi: 10.2514/1.J055595. URL `https://doi.org/10.2514/1.J055595`.

[67] J. Sirignano, J. F. MacArt, and J. B. Freund. Dpm: A deep learning pde augmentation method with application to large-eddy simulation. Journal of Computational Physics, 423:109811, 2020. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2020.109811. URL `https://www.sciencedirect.com/science/article/pii/S0021999120305854`.

[68] J. Smagorinsky. General circulation experiments with the primitive equations: I. the basic experiment. Monthly Weather Review, 91(3):99 – 164, 1963. doi: 10.1175/1520-0493(1963)091⟨0099:GCEWTP⟩2.3.CO;2. URL `https://journals.ametsoc.org/view/journals/mwre/91/3/1520-0493_1963_091_0099_gcewtp_2_3_co_2.xml`.

[69] A. M. Smith and T. Cebeci. Numerical solution of the turbulent-boundary-layer equations. Technical report, DOUGLAS AIRCRAFT CO LONG BEACH CA AIRCRAFT DIV, 1967.

[70] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. In 30th Aerospace Sciences Meeting and Exhibit, page 439, 1992. doi: 10.2514/6.1992-439. URL `https://arc.aiaa.org/doi/abs/10.2514/6.1992-439`.

[71] T. E. Springer, T. A. Zawodzinski, S. Gottesfeld, Z. T.A., and S. Gottesfeld. Polymer Electrolyte Fuel Cell Model. J. Electrochem. Soc., 138(8):2334–2341, 1991.

[72] J. Steelant and E. Dick. Modelling of bypass transition with conditioned navier–stokes equations coupled to an intermittency transport equation. International journal for numerical methods in fluids, 23(3):193–220, 1996.

[73] C. A. M. Ströfer and H. Xiao. End-to-end differentiable learning of turbulence models from indirect observations. Theoretical and Applied Mechanics Letters, 11 (4):100280, 2021.

[74] K. Suluksna, P. Dechaumphai, and E. Juntasaro. Correlations for modeling transitional boundary layers under influences of freestream turbulence and pressure gradient. International Journal of Heat and Fluid Flow, 30(1):66 – 75, 2009. ISSN 0142-727X. doi: https://doi.org/10.1016/j.ijheatfluidflow.2008.09.004. URL http://www.sciencedirect.com/science/article/pii/S0142727X0800146X.

[75] V. Sulzer, S. G. Marquis, R. Timms, M. Robinson, and S. J. Chapman. Python battery mathematical modelling (pybamm). Journal of Open Research Software, 9 (1), 2021.

[76] V. Sulzer, P. Mohtat, and J. B. Siegel. Reduced-order modeling of pem fuel cells using asymptotic analysis, Apr 2022. URL ecsarxiv.org/yntze.

[77] L. Sun, G. Li, Q. Hua, and Y. Jin. A hybrid paradigm combining model-based and data-driven methods for fuel cell stack cooling control. Renewable Energy, 147: 1642–1652, 2020.

[78] Y. B. Suzen and P. G. Huang. Modeling of Flow Transition Using an Intermittency Transport Equation . Journal of Fluids Engineering, 122(2):273–284, 02 2000. ISSN 0098-2202. doi: 10.1115/1.483255. URL https://doi.org/10.1115/1.483255.

[79] Y. B. Suzen, G. Xiong, and P. G. Huang. Predictions of transitional flows in low-pressure turbines using intermittency transport equation. AIAA Journal, 40(2): 254–266, 2002. doi: 10.2514/2.1667.

[80] W. Tollmien, H. Schlichting, H. Görtler, and F. Riegels. Über ein neues formelsystem für die ausgebildete turbulenz. In Ludwig Prandtl Gesammelte Abhandlungen, pages 874–887. Springer, 1961.

[81] B. Tracey, K. Duraisamy, and J. Alonso. Application of supervised learning to quantify uncertainties in turbulence and combustion modeling. In 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 2013. doi: 10.2514/6.2013-259. URL https://arc.aiaa.org/doi/abs/10.2514/6.2013-259.

[82] B. D. Tracey, K. Duraisamy, and J. J. Alonso. A machine learning strategy to assist turbulence model development. In 53rd AIAA Aerospace Sciences Meeting, page 1287, 2015. doi: 10.2514/6.2015-1287. URL https://arc.aiaa.org/doi/abs/10.2514/6.2015-1287.

[83] E. R. VAN DRIEST and C. B. BLUMER. Boundary layer transition- freestream turbulence and pressure gradient effects. AIAA Journal, 1(6):1303–1306, 1963. doi: 10.2514/3.1784. URL https://doi.org/10.2514/3.1784.

[84] R. Vetter and J. O. Schumacher. Free open reference implementation of a two-phase pem fuel cell model. Computer Physics Communications, 234:223–234, 2019.

[85] P. S. Volpiani, M. Meyer, L. Franceschini, J. Dandois, F. Renac, E. Martin, O. Marquet, and D. Sipp. Machine learning-augmented turbulence modeling for rans simulations of massively separated flows. Physical Review Fluids, 6(6):064607, 2021.

[86] D. K. Walters and D. Cokljat. A three-equation eddy-viscosity model for reynolds-averaged navier–stokes simulations of transitional flow. Journal of fluids engineering, 130(12), 2008.

[87] B. Wang, B. Xie, J. Xuan, and K. Jiao. Ai-based optimization of pem fuel cell catalyst layers for maximum power density via data-driven surrogate modeling. Energy Conversion and Management, 205:112460, 2020.

[88] F. Waschkowski, Y. Zhao, R. Sandberg, and J. Klewicki. Multi-objective cfd-driven development of coupled turbulence closure models. Journal of Computational Physics, 452:110922, 2022.

[89] J. Weatheritt and R. Sandberg. A novel evolutionary algorithm applied to algebraic modifications of the rans stress–strain relationship. Journal of Computational Physics, 325:22 – 37, 2016. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2016.08.015. URL http://www.sciencedirect.com/science/article/pii/S0021999116303643.

[90] D. C. Wilcox. Reassessment of the scale-determining equation for advanced turbulence models. AIAA journal, 26(11):1299–1310, 1988.

[91] D. C. Wilcox. The remarkable ability of turbulence model equations to describe transition. NUMERICAL AND PHYSICAL ASPECTS OF AERODYNAMIC FLOWS, 1992.

[92] D. C. Wilcox et al. Turbulence modeling for CFD, volume 2. DCW industries La Canada, CA, 1998.

[93] J.-L. Wu, H. Xiao, and E. Paterson. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. Physical Review Fluids, 3(7):074602, 2018.

[94] H. Xiao, J.-L. Wu, J.-X. Wang, R. Sun, and C. Roy. Quantifying and reducing model-form uncertainties in reynolds-averaged navier–stokes simulations: A data-driven, physics-informed bayesian approach. Journal of Computational Physics, 324:115–136, 2016. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2016.07.038. URL https://www.sciencedirect.com/science/article/pii/S0021999116303394.

[95] H. Yuan, H. Dai, X. Wei, and P. Ming. A novel model-based internal state observer of a fuel cell system for electric vehicles using improved kalman filter approach. Applied Energy, 268:115009, 2020.

[96] S. Zeierman and M. Wolfshtein. Turbulent time scale for turbulent-flow calculations. AIAA journal, 24(10):1606–1610, 1986.

[97] G. Zhu, W. Chen, S. Lu, and X. Chen. Parameter study of high-temperature proton exchange membrane fuel cell using data-driven models. International Journal of Hydrogen Energy, 44(54):28958–28967, 2019.