SAND2024-13776R

SANDIA REPORT

SAND2024-xxxxxx
Printed September 2024

Sandia National Laboratories

Data-driven closure modeling for hypersonic turbulent flows

Eric Parish, Matthew Barone, David Ching, Cyrus Jordan, Nathan Miller, Gary Nicholson, Kevin Gitushi, Steven Beresh (Sandia National Laboratories)

Niloy Gupta and Karthik Duraisamy (University of Michigan, Ann Arbor)

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

Available to DOE and DOE contractors from

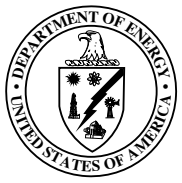
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods



ABSTRACT

The Reynolds-averaged Navier–Stokes (RANS) equations remain a workhorse technology for simulating compressible fluid flows of practical interest. Due to model-form errors, however, RANS models can yield erroneous predictions that preclude their use on mission-critical problems. This report summarizes work performed from FY22-FY24 focused on improving RANS models for hypersonic flows using data-driven modeling and scientific machine learning. In this work we:

1. Investigate the current capabilities of RANS models in Sandia's parallel aerodynamics and re-entry code (SPARC) for hypersonic flows with a focus on shock boundary layer interactions (SBLIs),
2. Assess several established corrections that exist in the literature aimed at improving predictions for SBLIs,
3. Develop improved models for the Reynolds stress tensor using tensor-basis neural networks,
4. Develop a neural-network-based variable turbulent Prandtl number model to reduce errors in wall heating in SBLIs.
5. Begin future investigations including employing the LIFE framework to improve wall heating predictions in SBLIs as well as the ensemble Kalman filter.

We find that current RANS models in SPARC are deficient for complex SBLI flows. In particular, no current model jointly predicts wall heat flux, wall shear stress, and wall pressure with reasonable accuracy. Existing corrections help, but do not alleviate this issue altogether. The development of improved models for the Reynolds stress tensor via tensor-basis neural networks results in more predictive RANS models across a suite of low-speed and high-speed cases. For hypersonic boundary layers, the inclusion of the wall-normal Reynolds stress via TBNNs has an appreciable impact on the wall-normal momentum balance and wall quantities. However, we find that improvements to the Reynolds stress tensor do not address the over-prediction in wall heat flux in SBLIs. We find that a neural-network-based variable turbulent Prandtl number model systematically and substantially improves wall heating predictions for a range of SBLI cases.

This page intentionally left blank.

ACKNOWLEDGEMENT

The authors acknowledge the SPARC development team for numerous insightful conversations that benefited this work. The authors additionally acknowledge all persons who have contributed to the NASA turbulence modeling resource, without which the present work would not be possible. The authors additionally acknowledge Pratikkumar Raje, with whom we had a number of beneficial conversations from which this work benefited.

This page intentionally left blank.

CONTENTS

This page intentionally left blank.

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

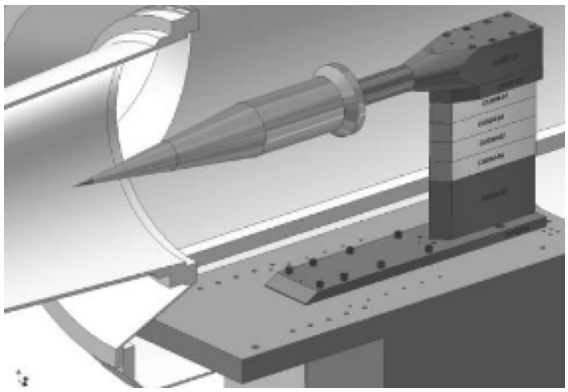
Simulating hypersonic turbulent flows is of growing interest to the aerospace community. Despite the growth in computing power over the past decades, scale-resolving modeling approaches such as Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS) remain impractical for most geometries in realistic operating environments. As a result, simulation approaches based on the Reynolds-averaged Navier–Stokes (RANS) equations remain the workhorse technology. RANS-based methods solve the averaged equations of motion, which greatly reduces the computational cost. The averaging process, however, results in unclosed terms that must be modeled. Developing accurate models is difficult, and as a result RANS models can yield erroneous predictions that are a dominant source of uncertainty in analyses of interest.

RANS models have several notable deficiencies when applied to hypersonic flows. These deficiencies include:

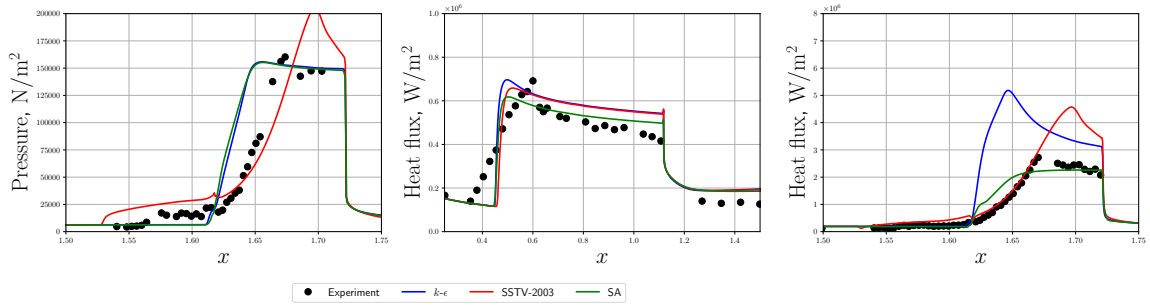
- Two-equations RANS models drastically over-predict wall heat flux around the point of reattachment for shock boundary layer interactions and compression ramps [100, 13]. This over-prediction appears to worsen as the Reynolds number increases.
- RANS models tend to over-predict skin friction and heat flux for high Reynolds number, cold-wall high-speed boundary layers [1, 15, 74].
- RANS models tend to *under-predict* skin friction and heat flux for low Reynolds number, cold-wall high-speed boundary layers [4].
- RANS models struggle to predict the size of the separation region in shock boundary layer interactions and compression ramps. In general, models underpredict the size of the separation region if stress limiters or other corrections are not employed [100].
- Standard models do not faithfully reproduce the law of the wall when there are significant variations in density [100]. This issue is more prominent for the $k-\epsilon$ model than $k-\omega$-based models and the Spalart–Allmaras model.

To make the above deficiencies concrete, we show results for RANS simulations of run 46 of the HIFiRE ground test studies [93]. This configuration comprises a Mach 7 flow over a cone-slice-ramp geometry. Figure 1-1 shows the geometry and predictions for wall quantities-of-interest (QoIs) for three baseline RANS models implemented in Sandia's parallel aerodynamics and re-entry code (SPARC): the $k-\epsilon$ model, the SSTV-2003 model, and the Spalart–Allmaras model[1]. We observe

[1]The below results are for model implementations that properly account for turbulent kinetic energy; see Chapter 2.



(a) HIFiRE ground test geometry. The figure is from Ref. [93].



(b) Wall QoIs from various RANS models.

Figure 1-1. Performance of RANS models on run 46 of the HIFiRE test geometry.

that all models fail to capture the size of the separation bubble and fail to accurately predict the rise in heat flux at the flare. The Spalart–Allmaras model is the only model to give reasonable heat flux predictions, but it does so partially due to mis-predicting flow separation. Both the SSTV-2003 and $k-\epsilon$ model dramatically over-predict wall heat flux. We additionally observe that all models over-predict wall heat flux leading up to the expansion, and the slope at which the wall heat flux is decreasing appears to under-predicted. This result is typical for flow these types of flows.

Many works have sought to improve RANS models for hypersonic flows. We provide a brief review here, and note that a more detailed review of the relevant literature will be provided in subsequent chapters. Catris and Aupoix [9] modified the diffusion terms in RANS models to account for density variations to recover the log-law in equilibrium boundary layers. Danis and Durbin [15] proposed a compressibility correction that takes Mach number and wall-cooling into account. The correction resulted in improved predictions for both wall profiles (skin friction, heat transfer) and wall-normal profiles (velocity, temperature). In a similar spirit, Barone et al. [4] proposed several compressibility corrections for the SA model including a transformation of the eddy viscosity, a new near-wall damping function, and a normal stress correction. These corrections showed improvements for

high-Reynolds number boundary layers. Rumsey [74] performs a detailed investigation of a variety of compressibility corrections for the $k-\omega$ model for hypersonic boundary layers, and concludes that there is a need for an improved understanding of turbulence models applied to hypersonic boundary layers.

Various corrections have been proposed for other deficiencies, such as performance in shock boundary layer interactions (SBLIs) and wall heating predictions. Coakley, Huang, and others [12, 13] proposed various compressibility corrections to improve model performance in SBLIs. These corrections include rapid compression corrections that help control the size of the separation region and length-scale limiters that reduce over-prediction in peak heat flux. Xiao et al. [103] and Ott et al. [59] have examined the development of variable turbulent Prandtl number models to improve heat transfer predictions. These models incur a higher computational cost due to the need to solve additional transport equations, and require further development for high-speed SBLIs [59]. Another approach has been proposed by Roy and Sinha [70, 69], who developed a variable turbulent Prandtl number model based on the linearized Rankine Hugoniot conditions applied to shock turbulence interaction. The model results in improved wall heating and has been demonstrated on a variety of hypersonic cases, but at the time of this writing, has not gained widespread use. In contrast to non-local or transport-equation-based models, several groups have pursued the development of algebraic turbulent heat flux and variable Prandtl number models [90, 6, 45, 99], several of which are targeted at hypersonic flows [90, 6]. These models are often based on approximations to the transport equations for the turbulent heat flux, but have yet to show systematic success on hypersonic SBLIs. We emphasize that many other works exist.

While the aforementioned approaches have made select improvements, state-of-the-art one and two-equation RANS models continue to struggle to provide reasonable predictions for wall shear stress, wall heat flux, and wall pressure in hypersonic SBLIs. Most models additionally continue to struggle to accurately predict wall QoIs for cold-wall, flat plate boundary layers. Scientific machine learning (SciML) approaches provide an alternative path to improving RANS models. Rather than relying on the traditional approach that combines phenomenological, scaling, and theoretical arguments, these techniques rely on high-fidelity data from simulation and/or experiment to learn a data-driven model. Numerous ML efforts have been undertaken to this end [17, 46, 95, 102, 62, 84, 80, 82, 31, 87, 97, 96, 98, 107]. We attempt to provide a brief (but necessarily non-comprehensive) review here. First, various researchers have leveraged direct numerical simulation (DNS) and/or experimental data to both calibrate and develop novel calibration procedures for RANS model parameters [58, 10, 18, 11, 67]. These techniques involve updating scalar parameters in turbulence closures such that the model output more faithfully matches experimental or high-fidelity simulation data. More ambitiously, Duraisamy et al. developed a field inversion and machine learning (FIML) framework that involves learning field corrections to turbulence models [62, 84, 80, 82, 31, 87]. The approach employs adjoint solvers and machine learning (ML) to optimize and learn a random corrective field applied, e.g., to turbulent production. FIML is a higher-capacity learning approach than parameter calibration as it more aggressively changes a turbulence model due to learning field corrections. FIML and its extensions have been applied successfully in various settings including

separated flows over airfoils [80] and transition [87].

Both parameter calibration and FIML are goal-oriented calibration methods in that they tune a term of interest (parameters in the former case and a full random field in the latter case) such that the calibrated RANS model associates with a reduced error on a specified quantity-of-interest (QoI) functional. Another body of work directly targets the intrinsic deficiencies of RANS models by modeling, e.g., the Reynolds stress directly. These approaches attempt to build better constitutive relations for the Reynolds stress via data-driven techniques with the idea that improved Reynolds stress predictions will propagate through to improved QoI predictions. Along the same time as the initial developments of FIML, Ling et al. pioneered *tensor-basis neural networks* (TBNNs) [46]. TBNNs attempt to learn the anisotropic component of the Reynolds stress tensor via a basis expansion of invariant tensors employed by Pope [65] in his generalized eddy viscosity hypothesis; neural networks are employed to learn the coefficients of this basis as a function of field invariants (e.g., trace of the strain rate squared). TBNNs have been shown to be capable of improved predictions of separated flows, secondary flows in ducts, and flows with curvature. Recent work has extended TBNNs to scalar flux modeling [51, 50]. Also around the same time as the development of both FIML and TBNNs, Xiao et al. developed a similar data-driven strategy, termed physics-informed machine learning (PIML), for improving Reynolds stress predictions [102]. Their approach differs slightly from Ling et al. in that they (1) model terms pertaining to an eigen-decomposition of the Reynolds stress tensor, (2) they model a discrepancy in the turbulent kinetic energy, and (3) they attempt to model only the *discrepancy* in the Reynolds stress (e.g., the difference between a standard RANS model prediction and the truth value), opposed to the entire Reynolds stress. This approach has additionally been shown to provide improved predictions on duct and separated flows. Recent work by Peters et al. has proposed a similar approach in where they model discrepancies to the Reynolds stress tensor in the mean strain-rate tensor eigenframe [63]. Yet another body of work has sought to develop interpretable ML models. For instance, Weatheritt, Sandberg, and colleagues have developed a framework based on evolutionary algorithms to learn explicit expressions for the anisotropy tensor (discrepancy) [97, 96, 98, 107].

The present work aims to improve the improve the predictive performance of turbulence models for hypersonic flows via data-driven approaches. We target two deficiencies in RANS models: the Reynolds stress tensor and the turbulent heat flux. In our approach, we first augment an existing model for the Reynolds stress with a data-driven discrepancy term constructed using tensor-basis neural networks. This formulation has the advantage that it builds upon an existing model form and thus corrects, rather than replaces, existing models. Further, the use of TBNNs makes the data-driven correction Galilean invariant and interpretable. Second, we develop a variable turbulent Prandtl number model using field inversion and machine learning to improve predictions for wall heating in SBLIs. The novel contributions of this work include:

- The development, assessment, and interpretation of a TBNN trained on a database including numerous hypersonic flow configurations,

- The development and assessment of a variable turbulent Prandtl number model that reduces wall heating errors in hypersonic SBLIs,
- The development of SciML-based approaches to improve the robustness of ML models via a calibrated deep ensemble approach and Lipschitz continuous neural networks.
- Preliminary investigations of the Ensemble Kalman filter and LIFE framework for training more consistent variable turbulent Prandtl number models.

The structure of this report will be as follows. In Chapter 2, we outline the RANS equations, several one and two-equation models that are currently available in SPARC, and several established corrections for hypersonic flows. Next, in Chapter 3 we perform a validation study of Sandia's implemented turbulence models on a suite of hypersonic flow cases. Chapter 4 outlines our data-driven approach for improving the Reynolds stress tensor and presents results across a suite of low-speed and high-speed test cases. Next, Chapter 5 outlines our approach for learning a variable turbulent Prandtl number model to improve wall heating predictions in hypersonic SBLIs, while Chapter 6 outlines future directions. Conclusions are provided in Chapter 7.

This page intentionally left blank.

2. GOVERNING EQUATIONS

We consider solutions to the Favre-averaged Navier–Stokes equations as given by

$$\frac{\partial \overline{\rho}}{\partial t} + \frac{\partial}{\partial x_j}\left(\overline{\rho}\tilde{u}_j\right) = 0,$$
$$\frac{\partial}{\partial t}\left(\overline{\rho}\tilde{u}_i\right) + \frac{\partial}{\partial x_j}\left(\tilde{u}_j\overline{\rho}\tilde{u}_i\right) = -\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial \sigma_{ij}}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_j},$$
$$\frac{\partial}{\partial t}\left(\overline{\rho}\tilde{E}\right) + \frac{\partial}{\partial x_j}\left(\overline{\rho}\tilde{u}_j\left(\tilde{E} + \frac{\overline{p}}{\overline{\rho}}\right)\right) =$$
$$\frac{\partial}{\partial x_j}\left(\sigma_{ij}\tilde{u}_i + \overline{\sigma_{ij}u_i''}\right) - \frac{\partial}{\partial x_j}\left(-\frac{c_p\tilde{\mu}}{\mathrm{Pr}}\frac{\partial \tilde{T}}{\partial x_j} + c_p\overline{\rho u_j''T''} - \tilde{u}_i\tau_{ij} + \frac{1}{2}\overline{\rho u_i''u_i''u_j''}\right),$$

where $\overline{\rho}$ is the averaged fluid density, $\tilde{u}_i$ is the Favre-averaged velocity, $\overline{p}$ is the Reynolds-averaged pressure, $\tau_{ij} \equiv -\overline{\rho u_i'' u_j''}$ is the Reynolds stress, $\sigma_{ij} = 2\tilde{\mu}\left(S_{ij} - \frac{1}{3}\frac{\partial \tilde{u}_k}{\partial x_k}\delta_{ij}\right)$ is the viscous stress tensor, $S_{ij} = \frac{1}{2}\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right)$ is the mean rate-of-strain tensor, $\tilde{T}$ is the temperature, $\tilde{\mu} \equiv \tilde{\mu}(\tilde{T})$ is the molecular viscosity modeled with Sutherland's law, $\tilde{E} = \tilde{e} + \frac{1}{2}\tilde{u}_i\tilde{u}_i + \tilde{k}$ is the Favre-averaged energy, $\tilde{e} = c_v\tilde{T}$ is the specific internal energy with c_v being the specific-heat coefficient at constant volume, and $\tilde{k}$ is the turbulent kinetic energy (TKE). A term denoted by v'' denotes a fluctuation about the Favre-average of v, while we use $\tilde{v}$ to denote the Favre-averaging process applied to v. Similarly, a term denoted by $\overline{v}$ denotes the Reynolds average of v. We assume a calorically perfect gas with an equation of state

$$\overline{p} = (\gamma - 1)\left[\overline{\rho}\tilde{E} - \frac{1}{2}\overline{\rho}\left(\tilde{u}_1^2 + \tilde{u}_2^2 + \tilde{u}_3^2\right) - \overline{\rho}\tilde{k}\right].$$

Closing the governing equations for mass, momentum, and energy requires modeling the Reynolds stress, turbulent heat flux, molecular diffusion, and turbulent transport. The standard assumptions for these terms are:

- Molecular diffusion and turbulent transport are modeled as

$$\overline{\sigma_{ij}u_i''} - \frac{1}{2}\overline{\rho u_i''u_i''u_j''} \approx \left(\tilde{\mu} + \frac{\mu_t}{\sigma_k}\right)\frac{\partial \tilde{k}}{\partial x_j},$$

  where μ_t is the turbulent viscosity and σ_k is a constant defined by a turbulence model.

- The Reynolds stress is modeled with a linear eddy viscosity model,

$$\tau_{ij} \approx 2\mu_t \left(S_{ij} - \frac{1}{3}\frac{\partial \tilde{u}_k}{\partial x_k}\delta_{ij} \right) - \frac{2}{3}\overline{\rho}\tilde{k}\delta_{ij}. \tag{2.0.1}$$

- The turbulent heat flux is modeled using a gradient-diffusion hypothesis

$$c_p \overline{\rho u_j'' T''} \approx -\frac{c_p \mu_t}{\mathrm{Pr}_t}\frac{\partial \tilde{T}}{\partial x_j}, \tag{2.0.2}$$

  where $\mathrm{Pr}_t = 0.9$ is the turbulent Prandtl number.

The above assumptions require the turbulent eddy viscosity, μ_t, and turbulent kinetic energy for closure.

2.1. RANS closure models

This section outlines the Spalart–Allmaras model, the SST-V 2003 model, and a $k-\epsilon$ model, which are the closure models currently implemented in Sandia's parallel aerodynamics and re-entry code (SPARC).

2.1.1. Spalart–Allmaras model

The Spalart–Allmaras (SA) model is a one-equation model that evolves a working variable for the turbulent eddy viscosity. SPARC supports the SA-neg variant. We refer the reader to the SA page on the NASA turbulence modeling resource [73] (NASA TMR) for the full definition of the governing equations under the "SA-neg" section.

2.1.2. SST-V 2003

The Menter Shear Stress transport (SST) family of models are $k-\omega$-based models that are designed to behave as a $k-\omega$ model in the inner region of the boundary layer and to switch to a $k-\epsilon$ model outside of the boundary layer. There are a variety of SST model variants, and SPARC supports the SSTV-2003 variant of the Menter SST family of models. The reader is referred directly to the SSTV-2003 section of the SST page on the NASA turbulence modeling resource (TMR) [73] for the definition of the governing equations. We highlight two distinguishing features of the SST-V 2003 variant which have a significant impact for hypersonic SBLIs. First, the "2003" family of SST models employs the strain invariant opposed to the vorticity invariant in the definition of the eddy viscosity. In an SBLI, the strain invariant will be larger than the vorticity invariant. This in turn decreases the magnitude of the eddy viscosity and enhances separation. Second, the "V"

family employs the vorticity invariant for computing turbulent production. This modeling choice additionally acts to decrease turbulence and enhance separation. We will observe that both of these modeling choices result in poor predictions for hypersonic SBLI flows.

At the time of this writing, SPARC makes several assumptions in the implementation of the SSTV-2003 model:

1. Turbulent kinetic energy is not included in the definition of the total energy, i.e., SPARC uses $\tilde{E} = \tilde{e} + \frac{1}{2}\tilde{u}_i\tilde{u}_i$ instead of $\tilde{E} = \tilde{e} + \frac{1}{2}\tilde{u}_i\tilde{u}_i + \tilde{k}$.
2. The turbulent kinetic energy is neglected in the definition of the Reynolds stress as appearing in the energy flux, i.e., the $\frac{2}{3}\overline{\rho}\tilde{k}\delta_{ij}$ term in Eq. (2.0.1) is neglected for the Reynolds stress tensor appearing in the viscous flux of the energy equation.

Due to poor predictions resulting from the SSTV-2003 variant, we will additionally consider the "standard" SST variant as outlined in the NASA TMR [73][1]. This model employs the vorticity invariant in the definition of eddy viscosity and computes the full production term.

2.1.3. $k-\epsilon$

Lastly, we consider the $k-\epsilon$ model currently supported by SPARC. The model is not a standard formulation, and as such we give its full definition here. The model is given by

$$\begin{aligned}
&\frac{\partial \overline{\rho}\tilde{k}}{\partial t} + \frac{\partial}{\partial x_i}\left(\overline{\rho}\tilde{u}_i\tilde{k}\right) = \frac{\partial}{\partial x_i}\left[\left(\tilde{\mu} + \frac{\mu_t}{\sigma_k}\right)\frac{\partial \tilde{k}}{\partial x_i}\right] + P_k - \rho\epsilon, \\
&\frac{\partial \overline{\rho}\tilde{\epsilon}}{\partial t} + \frac{\partial}{\partial x_i}\left(\overline{\rho}\tilde{u}_i\tilde{\epsilon}\right) = \frac{\partial}{\partial x_i}\left[\left(\mu + \frac{\mu_t}{\sigma_\epsilon}\right)\frac{\partial \tilde{\epsilon}}{\partial x_i}\right] + \frac{\tilde{\epsilon}}{\tilde{k}}\left(C_{\epsilon 1} f_1 P_k - C_{\epsilon 2} f_2 \overline{\rho}\tilde{\epsilon}\right) + S_\epsilon, \\
&\mu_t = f_\mu C_\mu \frac{k^2}{\epsilon}.
\end{aligned} \tag{2.1.1}$$

The low-Reynolds number functions and model constants are

$$f_\mu = \left(1 + 4\mathrm{Re}_t^{-3/4}\right)\tanh\left(\frac{\mathrm{Re}_k}{125}\right), f_1 = 1.0 - \exp\left[-\left(\frac{\mathrm{Re}_t}{40}\right)^2\right] + \frac{40\mathrm{Re}_k}{\mathrm{Re}_k^2 + 10000}, \; f_2 = 1 - \frac{2}{9}\exp\left[-(\mathrm{Re}_t)^2\right],$$

$$\mathrm{Re}_t = \frac{\overline{\rho}\tilde{k}^2}{\mu\tilde{\epsilon}}, \; \mathrm{Re}_k = \frac{\overline{\rho}\sqrt{\tilde{k}}y}{\mu}, S_\epsilon = \frac{14}{9}\frac{C_{\epsilon 2}\mu\tilde{\epsilon}}{\tilde{k}}\left[\sum_{i=1}^{3}\left(\frac{\partial\sqrt{\tilde{k}}}{\partial x_i}\right)^2\right],$$

$$C_\mu = 0.09, \sigma_k = 1.0, \sigma_\epsilon = 1.3, C_{\epsilon 1} = 1.43, C_{\epsilon 2} = 1.92.$$

We note that TKE is properly accounted for in all $k-\epsilon$ results presented in this report.

[1] This model was added in a research branch and is not supported by SPARC at the time of this writing

2.2. Established corrections for hypersonic flows

A variety of corrections have been proposed for high-speed flows. These include compressibility corrections to, e.g., account for dilatational effects, length-scale limiting to address over-predictions in wall heat flux in SBLIs, and stress limiters and rapid compression corrections to improve the prediction of the separation region in SBLIs. We outline several such corrections here.

2.2.1. Catris and Aupoix correction

Two-equation RANS models have been shown to poorly predict the logarithmic region of the boundary layer in variable density boundary layers. Analysis in Wilcox [100], for instance, shows that both the $k-\omega$ and $k-\epsilon$ models result in a velocity profile that follows the law of the wall, but with a "constant" that depends on the wall-density ratio. For large wall-density ratios, which happens in all but strongly cooled walls, both the $k-\epsilon$ and $k-\omega$ model will over-predict the "constant" in the law of the wall and will have a correspondingly lower wall shear stress. This issue is more pronounced for $k-\epsilon$-based models than $k-\omega$ models.

The Catris and Aupoix correction [9] (CA-correction) aims to address this deficiency by correcting the diffusion terms to recover log-law behavior of the boundary layer velocity profile. The correction is applicable to a variety of RANS models and can be implemented either by changing the working variables or adding additional source terms. For the $k-\epsilon$ model, the correction results in the governing equations

$$\begin{aligned}
&\frac{\partial \overline{\rho}\tilde{k}}{\partial t} + \frac{\partial}{\partial x_i}\left(\overline{\rho}\tilde{u}_i\tilde{k}\right) = \frac{\partial}{\partial x_i}\left[\frac{1}{\tilde{\rho}}\left(\tilde{\mu} + \frac{\mu_t}{\sigma_k}\right)\frac{\partial \tilde{\rho}\tilde{k}}{\partial x_i}\right] + P_k - \rho\epsilon, \\
&\frac{\partial \overline{\rho}\tilde{\epsilon}}{\partial t} + \frac{\partial}{\partial x_i}\left(\overline{\rho}\tilde{u}_i\tilde{\epsilon}\right) = \frac{\partial}{\partial x_i}\left[\left(\mu + \frac{\mu_t}{\sigma_\epsilon}\right)\frac{\partial \tilde{\epsilon}}{\partial x_i}\right] + \frac{\tilde{\epsilon}}{\tilde{k}}\left(C_{\epsilon 1} f_1 P_k - C_{\epsilon 2} f_2 \overline{\rho}\tilde{\epsilon}\right) + S_\epsilon + \rho\tilde{\epsilon}\left(\frac{\partial \tilde{u}_i}{\partial x_i}\right) + \\
&\qquad \frac{3}{2}\frac{\partial}{\partial x_i}\frac{1}{\tilde{\rho}}\left(\tilde{\epsilon}\left(\mu + \frac{\mu_t}{\sigma_\epsilon}\right)\frac{\partial \tilde{\rho}}{\partial x_i}\right) + \frac{1}{\tilde{\rho}}\left(\mu + \frac{\mu_t}{\tilde{\rho}}\right)\frac{\partial \tilde{\rho}}{\partial x_i}\left(\frac{3}{2}\frac{\tilde{\epsilon}}{\tilde{\rho}}\frac{\partial \tilde{\rho}}{\partial x_i} + \frac{\partial \tilde{\epsilon}}{\partial x_i}\right), \\
&\mu_t = f_\mu C_\mu \frac{k^2}{\epsilon}.
\end{aligned} \tag{2.2.1}$$

It is worth highlighting that, for the $k-\epsilon$ model, the Catris and Aupoix correction adds a "rapid compression" type term (which will be discussed in a subsequent section) to the ϵ equation, $\rho\tilde{\epsilon}\left(\frac{\partial \tilde{u}_i}{\partial x_i}\right)$, whose sign is opposite of that recommended by the rapid compression corrections. The CA-correction will, in general, have the net effect of decreasing the size of the separation region in SBLIs.

2.2.2. Length-scale limiting

Two-equation models are known to massively over-predict wall heat flux at the point of separation in hypersonic SBLIs [100]. Various approaches have been proposed to address this, including algebraic heat flux models and variable turbulent Prandtl number models [90, 6, 45, 99]. One straightforward correction that has gained some use is to limit the turbulent length-scale. Coakley, Huang, and co-authors[13, 12] established that the turbulent length-scale becomes much larger than a distance-to-wall-based length-scale in SBLI flows. To address this deficiency, they propose limiting the length-scale used in two-equation models as

$$\ell_{\text{rans}} = \min\left(C_\ell d, \frac{k^{3/2}}{\epsilon}\right). \tag{2.2.2}$$

The constant $C_\ell = 2.5$ is based on the relation $\kappa C_\mu^{-3/4} \approx 2.5$, where $\kappa = 0.41$. Practically, the length-scale limiter is implemented by clipping the secondary variable in a two-equation model (e.g., ϵ). Noting that, ignoring low-Reynolds number functions, the turbulent eddy viscosity is given by

$$\mu_t = C_\mu \tilde{\rho} \sqrt{k} \ell,$$

limiting the turbulent length-scale reduces the turbulent eddy viscosity, which in turn reduces the magnitude of both momentum transfer and heat transfer.

2.2.3. Rapid compression corrections

In addition to over-predicting peak heat transfer in SBLIs, many RANS models do a poor job at predicting the size of the separation region in hypersonic SBLIs. As an example, a baseline $k-\epsilon$ model or $k-\omega$ model without a stress limiter usually under-predicts the size of the separation region in SBLIs [100]. To address this issue, Coakley and Huang proposed modifications that reduce the turbulent length-scale through a strong compression. This reduction in turbulent length-scale enhances the size of the separation region. To achieve this, Coakley and Huang combine the $\tilde{k}$ and $\tilde{\epsilon}$ equations to form a model equation for the turbulent length-scale. They propose amplifying the impact of dilatation in the ϵ equation so that the length-scale contracts in a compression and expands in a compression.

The corrections proposed by Coakley and Huang reduce both the turbulent length-scale and turbulent eddy viscosity through a compression. On the flip side, they will increase the turbulent length-scale and turbulent eddy viscosity through an expansion. Here, we note that we can add rapid compression terms to both the k equation and the secondary equation that enables control over both the net change of the turbulent length-scale and turbulent viscosity. Specifically, we could

examine corrections of the form

$$\begin{aligned}
\frac{\partial \overline{\rho}\tilde{k}}{\partial t} + \frac{\partial}{\partial x_i}\left(\overline{\rho}\tilde{u}_i\tilde{k}\right) &= \frac{\partial}{\partial x_i}\left[\left(\tilde{\mu} + \frac{\mu_t}{\sigma_k}\right)\frac{\partial \tilde{k}}{\partial x_i}\right] + P_k - \rho\epsilon + C_{\mathrm{k}}\rho\tilde{k}\left(\frac{\partial \tilde{u}_i}{\partial x_i}\right) \\
\frac{\partial \overline{\rho}\tilde{\epsilon}}{\partial t} + \frac{\partial}{\partial x_i}\left(\overline{\rho}\tilde{u}_i\tilde{\epsilon}\right) &= \frac{\partial}{\partial x_i}\left[\left(\mu + \frac{\mu_t}{\sigma_\epsilon}\right)\frac{\partial \tilde{\epsilon}}{\partial x_i}\right] + \frac{\tilde{\epsilon}}{\tilde{k}}\left(C_{\epsilon 1}f_1 P_k - C_{\epsilon 2}f_2\overline{\rho}\tilde{\epsilon}\right) + S_\epsilon + C_\epsilon\rho\tilde{\epsilon}\left(\frac{\partial \tilde{u}_i}{\partial x_i}\right),
\end{aligned} \tag{2.2.3}$$

where the final terms on the right-hand side of the above equations are the rapid compression corrections.

We can examine the impact of the rapid compression corrections by deriving transport equations for both the turbulent length-scale and turbulent viscosity. First we can derive an equation for the turbulent length-scale, $l = \frac{k^{3/2}}{\epsilon}$, via chain rule by $\frac{dl}{dt} = 3\frac{\sqrt{k}}{2\epsilon}\frac{dk}{dt} - \frac{\sqrt{k^3}}{\epsilon^2}\frac{d\epsilon}{dt}$ [12]. The contributions arising from the RC corrections detailed in Eq. (2.2.3) lead to the following change to the length-scale equation,

$$\left[\frac{dl}{dt}\right]_{\mathrm{rc}} - \left[\frac{dl}{dt}\right]_{\mathrm{standard}} = l\left(\frac{3}{2}C_{\mathrm{k}} - C_\epsilon\right)\nabla\cdot\tilde{\mathbf{u}}.$$

For the corrections to shrink the length-scale in a compression and grow it in an expansion, which makes sense physically, we require $\frac{3}{2}C_{\mathrm{k}} > C_\epsilon$.

In addition to the turbulent length-scale, we can derive a governing equation for the eddy viscosity. Ignoring wall functions, $\mu_t = \overline{\rho}C_\mu\frac{k^2}{\epsilon}$. By chain rule, it is straightforward to show that the net effect of the rapid compression corrections on the governing equations is

$$\left[\frac{d\mu_t}{dt}\right]_{\mathrm{rc}} - \left[\frac{d\mu_t}{dt}\right]_{\mathrm{standard}} = \left(2C_{\mathrm{k}} - C_\epsilon\right)\mu_t\nabla\cdot\tilde{\mathbf{u}}.$$

If $2C_{\mathrm{k}} > C_\epsilon$, the eddy viscosity will shrink in a compression and increase in an expansion. On the other hand, if $2C_{\mathrm{k}} < C_\epsilon$, then the eddy viscosity will grow in a compression and shrink in an expansion.

Two interesting cases are worth commenting on. First, consider $C_\epsilon = \frac{3}{2}C_{\mathrm{k}}$. In this case, the rapid compression corrections will have no net impact on the turbulent length-scale, but they will still modify the eddy viscosity. The net impact is

$$\left[\frac{d\mu_t}{dt}\right]_{\mathrm{rc}} - \left[\frac{d\mu_t}{dt}\right]_{\mathrm{standard}} = \frac{1}{2}C_{\mathrm{k}}\mu_t\nabla\cdot\tilde{\mathbf{u}}.$$

This result suggests that we can suppress/grow the turbulent viscosity by increasing/decreasing the constant C_{k} while holding the turbulent length-scale constant. Second, consider $C_\epsilon = 2C_{\mathrm{k}}$. In this case, the rapid compression corrections will not have a net impact on the turbulent viscosity, but they will have an impact on the length-scale. To ensure that the length-scale decreases in a compression and increases in an expansion, we require $\frac{3}{2}C_{\mathrm{k}} - C_\epsilon = -\frac{1}{2}C_{\mathrm{k}} > 0$, which implies that $C_{\mathrm{k}} < 0$.

2.2.4. Stress limiters

The rapid correction corrections proposed by Coakley and Huang have the net impact of reducing turbulent viscosity, which enhances separation. A similar strategy is the use of stress limiters. These methods, which are typically used in $k-\omega$-based models, limit the turbulent eddy viscosity. For example, for a standard $k-\omega$ model, the turbulent eddy viscosity is given by

$$\mu_t = \frac{\tilde{\rho}\tilde{k}}{\hat{\omega}}$$

where

$$\hat{\omega} = \max\left(\omega, C_{\text{lim}}\sqrt{\frac{2\tilde{S}_{ij}\tilde{S}_{ij}}{\beta^*}}\right).$$

This limiter reduces the eddy viscosity in regions where the specific dissipation rate is very high. For high speed SBLIs, the stress limiter can promote separation [100].

2.2.5. Modifications to the Spalart–Allmaras model for compressible flow

Several modifications to the SA model for compressible boundary layer flows were proposed in [4]. First, the low-Reynolds number correction of [86] is applied, which raises the model's skin friction for low-Reynolds number boundary layers. Second, a modified near-wall viscous damping function inspired by hypersonic boundary layer DNS data sets and compressibility transformation arguments is given by:

$$f_{v1}^c = \beta^{-1}\left(1 + 0.8M_{t_{SA}}^2\right)^{-1} f_{v1}^i, \quad M_{t_{SA}} = \frac{\sqrt{\tilde{\nu}S/0.3}}{c}, \tag{2.2.4}$$

$$\beta = 1 + \left(\frac{1}{2\rho}\frac{d\rho}{dy} - \frac{1}{\mu}\frac{d\mu}{dy}\right) y, \quad f_{v1}^i = \left[1 - \exp\left(-\frac{\chi}{7}\right)\right]^2. \tag{2.2.5}$$

where y is a coordinate measured from the nearest wall point in the wall-normal direction and $S = ||\Omega||$ is the norm of the vorticity vector. The function is blended with the incompressible damping function,

$$f_{v1}^b = \theta(\chi) f_{v1}^i + (1 - \theta(\chi)) f_{v1}^c, \quad \chi = \frac{\tilde{\nu}}{\nu}, \tag{2.2.6}$$

with

$$\theta(\chi) = \frac{1}{2}\left[\tanh\left(\frac{\chi - 15}{5}\right) + \tanh\left(\frac{\chi + 15}{5}\right)\right]. \tag{2.2.7}$$

A further blending is required to ensure the corrected form does not activate in a low-Reynolds-number region outside the boundary layer, resulting in the final value of

$$f_{v1} = f_d f_{v1}^i + (1 - f_d) f_{v1}^b, \quad f_d = 1 - \tanh\left([8 r_d]^3\right), \quad r_d = \frac{\nu + \tilde{\nu}}{\kappa^2 d^2 \max(10^{-10}, \left[\sum_{ij} \frac{\partial U_i}{\partial x_j}^2\right]^{1/2})}, \tag{2.2.8}$$

where d is the distance to the nearest wall. Finally, a normal Reynolds stress representation is applied to more accurately calculate the associated wall-normal pressure gradients that can arise in hypersonic boundary layers. The Reynolds stress is computed using

$$\tau_{ij} = 2\mu_t \left(S_{ij} - \frac{1}{3}\frac{du_k}{dx_k}\delta_{ij}\right) - C_{cr2}^{BL} \mu_t S \delta_{ij}. \tag{2.2.9}$$

For a boundary layer, the last term resembles the normal stress term from the SA-QCR2020 model [71], with a modified coefficient $C_{cr2}^{BL} = 1.11$.

3. VALIDATION STUDY OF CURRENT SANDIA MODELS FOR HYPERSONIC FLOWS

Before introducing data-driven corrections, in this chapter we perform a validation study of the RANS models introduced in Chapter 2, along with some of the established compressibility corrections, on a suite of test cases pertinent to hypersonic aerodynamics. These cases include flat plate boundary layers, impinging SBLIs, compression ramps, and test vehicles. For all cases, the RANS models are benchmarked against experimental data. Table 3-1 summarizes the validation cases. While not shown, grid convergence studies were performed on all validation cases, and the results presented here are on grids that yield either fully grid converged solutions or very close to grid converged solutions. We assess the following RANS models:

- $k-\epsilon$. A standard $k-\epsilon$ model, as given in Eq. (2.1.1).
- $k-\epsilon-$RC. A $k-\epsilon$ model with the addition of a rapid compression correction, as given in Eq. (2.2.3) with constants $C_{\mathrm{k}} = 0, C_\epsilon = -0.75$.
- $k-\epsilon-$CA–RC–LC. A $k-\epsilon$ model with the Catris and Aupoix correction, rapid compression corrections, and a length-scale limiter. The model is given by Eq. (2.2.1) with the addition of a length-scale limiter with a constant $C_\ell = 5.0$ and a rapid compression corretion with constants $C_{\mathrm{k}} = 0, C_\epsilon = -2.0$. We note that we observed the recommended value of $C_\ell = 2.5$ from Coakley and Huang [12] had a significant impact on zero pressure gradient boundary layers and degraded performance. We additionally note the use of a "more negative" C_ϵ than in the $k-\epsilon-$RC model to compensate for the rapid-compression-type term added by the CA correction.
- SST. The "standard" SST model as described on the NASA TMR.
- SSTV-2003. The SSTV-2003 variant of the SST model as described on the NASA TMR.
- SSTV-2003 (SPARC). The SSTV-2003 variant implemented in SPARC, which is equivalent to the SSTV-2003 variant detailed on the NASA TMR but with turbulent kinetic energy neglected in the definition of total energy and in the definition of the Reynolds stress in the energy flux.
- SA. The SA-neg variant of the SA model as described on the NASA TMR.
- SA–CA. The SA-neg variant of the SA model with the Catris and Aupoix correction.

Table 3-1. Summary of validation cases.

Case	M_∞	Re_L	Shock/ramp angle	T_w/T_0
Mach 11 flat plate boundary layer	11.1	39.1×10^6	N/A	0.20
CUBRC 2D Compression Corner Run 12	8.2	146.9×10^6	27°	0.29
CUBRC 2D Compression Corner Run 19	8.1	139.3×10^6	33°	0.29
CUBRC 2D Compression Corner Run 24	8.0	145.3×10^6	36°	0.29
Schülein impinging shock	5.0	18.5×10^6	6°	0.73
Schülein impinging shock	5.0	18.5×10^6	10°	0.73
Schülein impinging shock	5.0	18.5×10^6	14°	0.73
Hollow cylinder flare run 13	6.01	16.24×10^6	36°	0.20
Hollow cylinder flare run 18	6.96	5.18×10^6	36°	0.14
HIFiRE-1 Run 30	7.2	16.9×10^6	33°	0.13
HIFiRE-1 Run 46	6.6	30.8×10^6	33°	0.16

- SA–CA–lowRe–f_{v1}–normal. The SA variant of Barone with various corrections for hypersonic boundary layers [4].

3.1. Mach 11 cold-wall boundary layer

We consider a Mach 11, cold-wall boundary layer. This case is designed to match the flow conditions for a Mach 11.1 hypersonic turbulent flat plate boundary layer tested at Calspan–University of Buffalo Research Center (CUBRC) [22, 23]. DNS results for this case are reported in Refs. [105, 35]. Various studies have examined the performance of RANS models on this case [35, 1, 23, 15] and have demonstrated a systematic over prediction of both skin friction and wall heating. To account for the leading edge shock we employ modified free-stream boundary conditions such that the post-shock conditions match those of the corresponding DNS simulations and CUBRC experiments.

Figure 3-1 shows predictions from the various RANS models for wall QoIs. We observe that all models perform similarly with several exceptions.

- Most notably, we observe that the addition of the CA correction to the $k-\epsilon$ model results in an over-prediction in both shear stress and heat flux. This trend is expected [100] since the baseline $k-\epsilon$ model is known over-predict the constant in the law of the wall and, correspondingly, under-predict wall shear stress. We additionally note that the $k-\epsilon$ model presented here over-predicts wall shear stress for subsonic boundary layers.
- While small, the SPARC implementation of SSTV-2003, which neglects TKE in the definition of total energy and in the definition of the Reynolds stress in the energy flux, differs from the

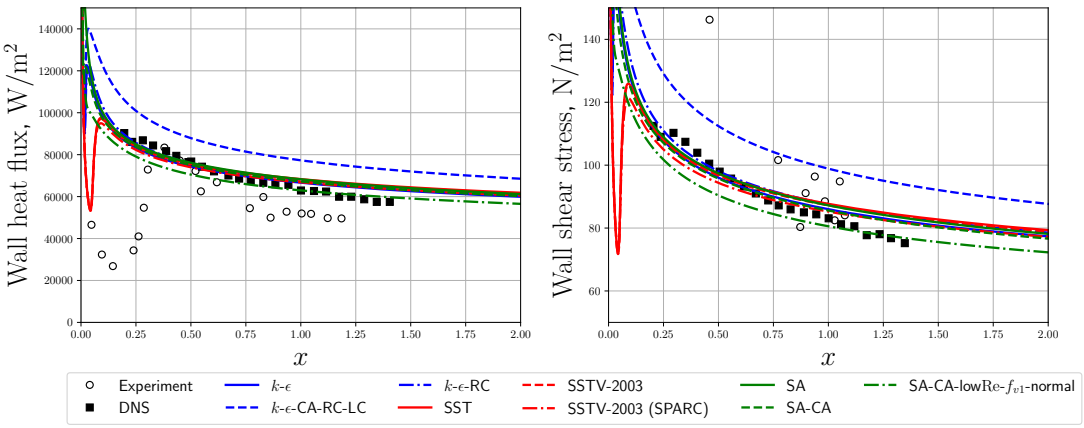


Figure 3-1. Mach 11 flat plate boundary layer. RANS model predictions for wall heat flux (left) and wall shear stress (right).

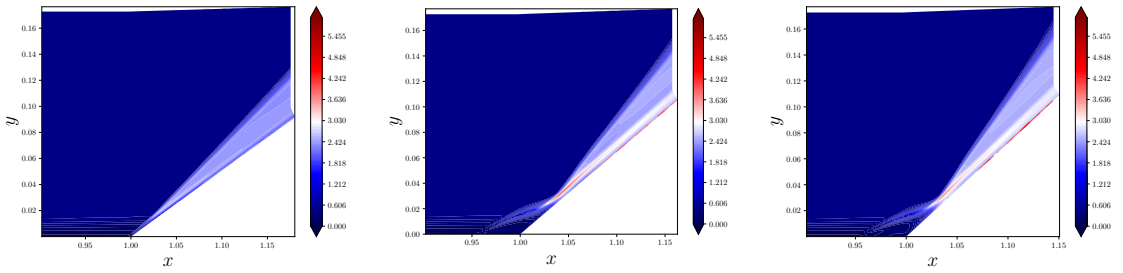
truth implementation of the SSTV-2003 model. While the difference is small, it is notable that it is present for even a flat plate boundary layer.

- The SA–CA–lowRe–f_{v1}–normal model results in slightly improved predictions for wall shear stress and wall heat flux as compared to the standard SA model for higher Reynolds numbers.
- We also note that there is disagreement between DNS data and the corresponding experimental data for both wall heat flux and wall shear stress. The reason for this disagreement is unclear.

3.2. Holden compression ramps

We next consider three Mach 8 compression ramps with deflection angles of 27° (run 12), 33° (run 19), and 36° (run 24). Experimental data for these cases are available from Holden et al. [28]. We note that these cases are at high Reynolds numbers. Figure 3-2 shows density contours predicted by a baseline $k-\epsilon$ model to depict the flow. Figure 3-3 shows predictions from the various RANS models for wall QoIs, while Figure 3-4 shows the associated relative errors of these QoIs as compared to experimental data. We make the following observations.

- SST model variants:
  - Comparing SSTV-2003 to SSTV-2003 (SPARC), we observe that neglecting the turbulent kinetic energy in the definition of total energy and the energy flux has a substantial impact on the solution. Neglecting these terms has the net effect of increasing the size of the separation bubble. This increased separation bubble size is in worse agreement with experimental data.



(a) Holden run 12 **(b) Holden run 19** **(c) Holden run 24**

Figure 3-2. Density contours for Holden compression ramps.

  - Comparing SST to SSTV-2003, we observe that the SSTV-2003 variant results in a larger separation region. This increased separation region is due to the use of the strain invariant in the definition of eddy viscosity for the SSTV-2003 model, which in turn decreases turbulent viscosity and enhances production.

- $k-\epsilon$ model variants:

  - The baseline $k-\epsilon$ model under-predicts the size of the separation bubble and massively over-predicts peak heat flux.

  - Comparing $k-\epsilon-$RC to $k-\epsilon$, we observe that the inclusion of rapid compression corrections results in systematic improvements to the size of the separation region. The peak heat flux is still massively over-predicted.

  - Comparing $k-\epsilon-$CA–RC–LC to $k-\epsilon-$RC, we observe that the length-scale limiter drastically reduces peak wall heat flux and brings it into better agreement with the data. This improvement, however, is at the expense of a reduced peak wall shear stress which is, in general, in worse agreement with experimental data.

- SA model variants:

  - The SA model variants give systematically better predictions for wall heat flux as opposed to the two equation models. This improvement in wall heat flux, however, is at the expense of poor predictions for wall shear stress.

  - The baseline SA and SA–CA model variants tend to under-predict separation. The inclusion of the normal stress correction and the re-calibrated f_{v1} results in slightly too large a separation bubble.

- Model-to-model comments:

  - In general, the $k-\epsilon$-based models out-performs SST- and SA-based models for wall shear stress.

- The SA model results in the best predictions for wall heat flux at the expense of the worse predictions for wall shear stress.

- The $k-\epsilon$ models with the rapid compression corrections tend to give the best predictions for the size of the separation region.

3.3. Schülein shock boundary layer interactions

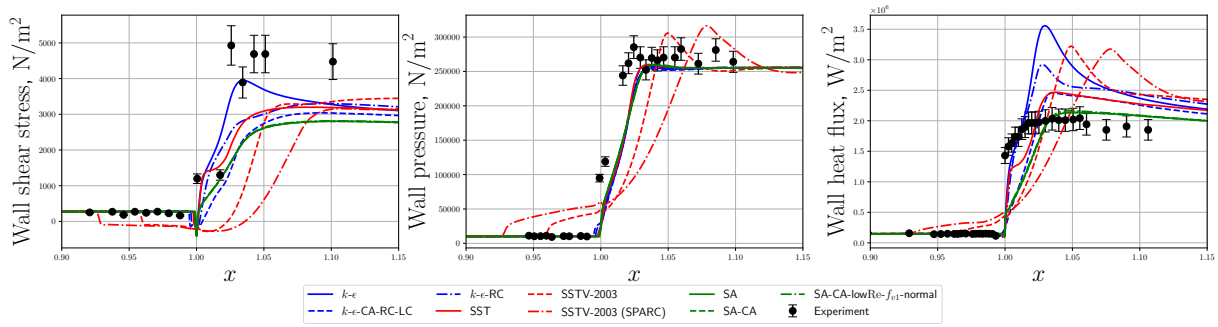
We consider two impinging shock boundary layer interactions at Mach 5 with impinging shock angles of 10° and 14° with experimental data from Schülein [78]. Figure 3-5 shows density contours predicted by a standard $k-\epsilon$ model to depict the flow.

Figure 3-6 shows predictions from the various RANS models for the Schülein SBLI validation cases, while Figure 3-7 shows relative errors for wall QoIs as compared to experimental data. Overall, we make similar observations for the Schülein SBLI cases as for the Holden compression ramps. The $k-\epsilon$ models give the best overall predictions for wall shear stress and wall pressure, but drastically over-predict heat flux. Neglecting the TKE results in erroneous predictions, as seen in the SSTV-2003 (SPARC) model. The SA model tends to under-predict separation, and wall heat flux is well captured at the expense of wall shear.

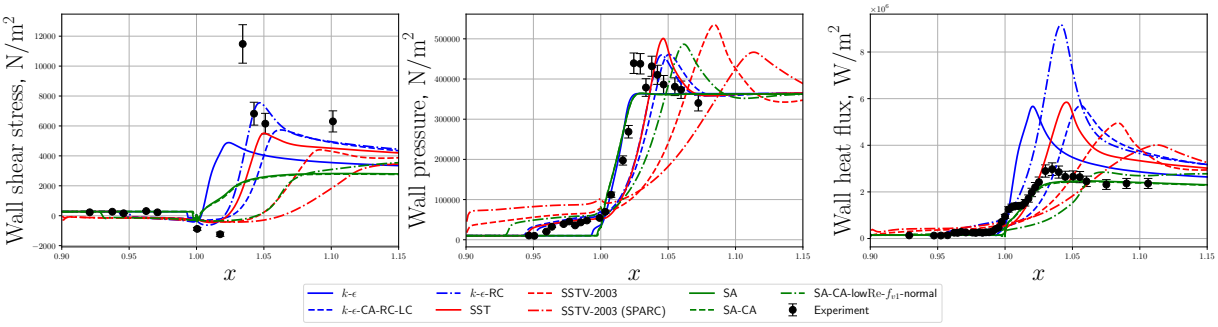
3.4. CUBRC Hollow Cylinder Flare

We consider run 13 and 18 of the CUBRC hollow cylinder flare (HCF). The HCF run 13 corresponds to a Mach 6 axisymmetric hollow cylinder flare with a 36° ramp. The run 18 case employs the same geometry, but is at Mach 7 with a lower Reynolds number. Experimental data for wall quantities are available from CUBRC. Both cases include a transition from laminar to turbulent flow. The transition location is determined from the experimental data and we manually enforce a trip by deactivating the turbulence model production term upstream of this location. A schematic of the experimental setup for the HCF is shown in Figure 3-8.

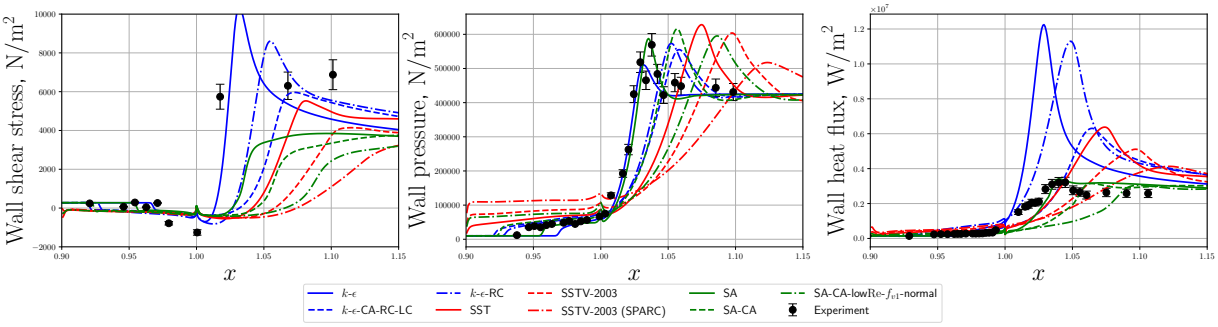
Figure 3-9 shows predictions from the various RANS models for the HCF cases while Figure 3-10 presents relative errors for wall QoIs. We note that experimental data for wall shear stress are not available. The same observations present in the Holden and Schülein cases are again present here. Additionally, we highlight that the discrepancy in peak heat flux for the two-equation models is much higher for run 13 than for run 18 and appears to have a Reynolds number dependence.



(a) Run 12.

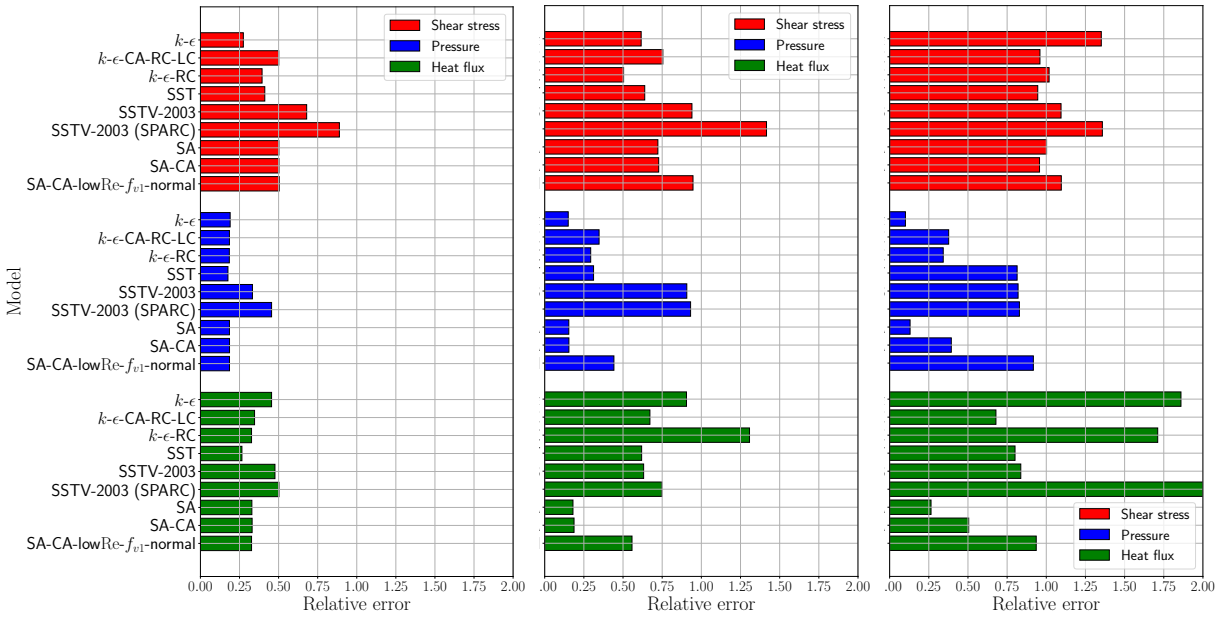


(b) Run 19.



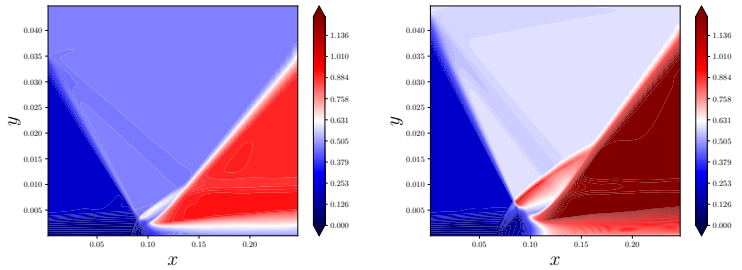
(c) Run 24.

Figure 3-3. Holden compression ramps. RANS model predictions for Holden run 12 (top), run 19 (middle), and run 24 (bottom) compression ramp cases.



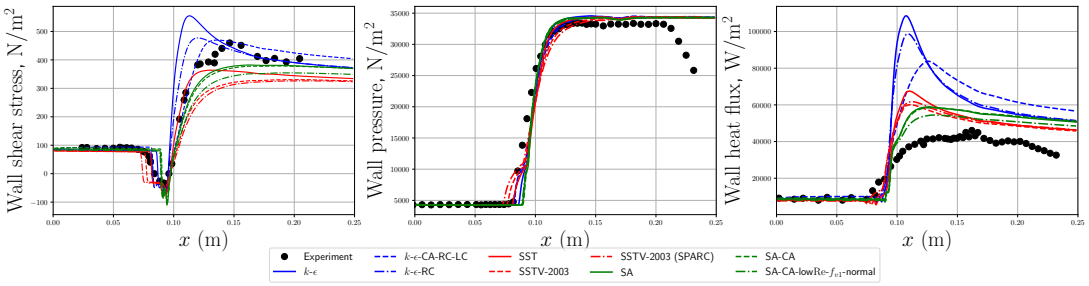
(a) Run 12. **(b) Run 19.** **(c) Run 24.**

Figure 3-4. Holden compression ramps. Relative errors for wall QoIs.

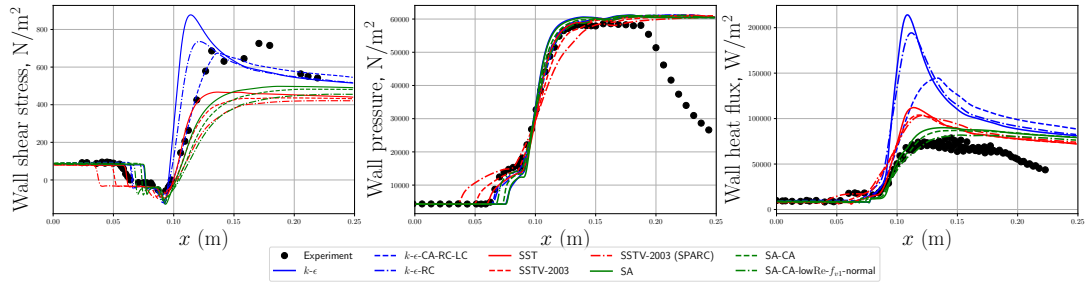


(a) 10° **(b)** 14°

Figure 3-5. Density contours for Schülein impinging SBLIs.

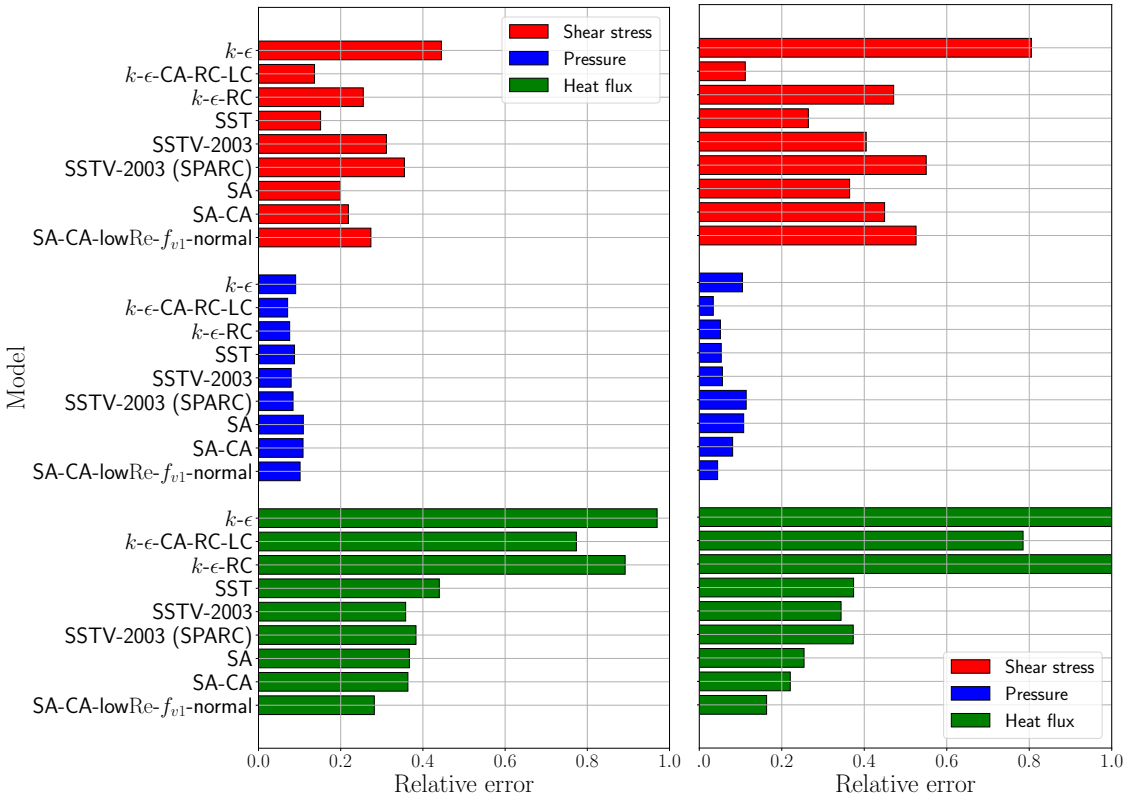


(a) 10° **case.**



(b) 14° **case.**

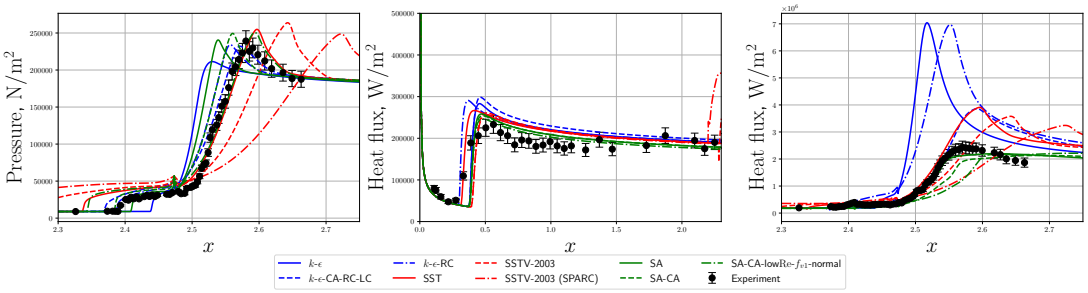
Figure 3-6. Schülein SBLIs. RANS model predictions for wall QoIs for the 10° **(top), and** 14° **(bottom) cases.**



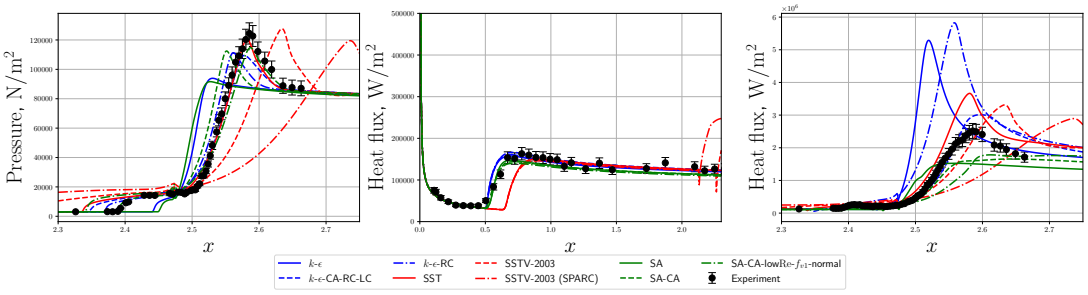
(a) 10° **(b)** 14°

Figure 3-7. Schülein SBLIs. Relative errors for wall QoIs.

Figure 3-8. Hollow cylinder flare geometry. The figure is from Ref. [27].

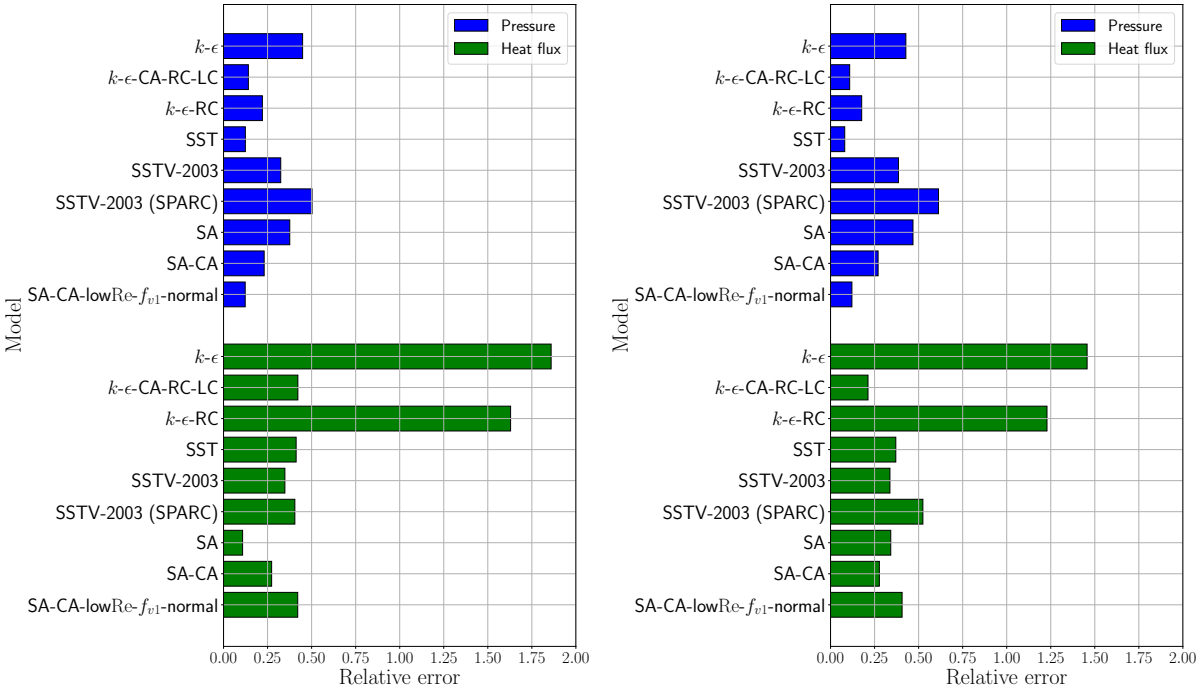


(a) Run 13.



(b) Run 18.

Figure 3-9. CUBRC hollow cylinder flare. RANS model predictions for run 13 (top) and run 18 (bottom).



(a) Run 13. **(b) Run 18.**

Figure 3-10. CUBRC hollow cylinder flare. Relative errors for wall QoIs.

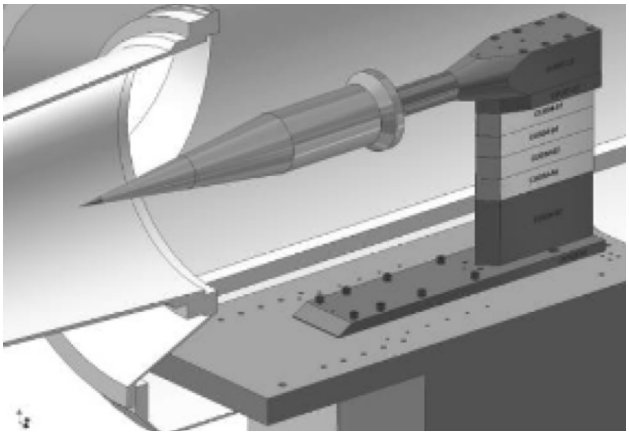


Figure 3-11. HIFiRE ground test geometry. The figure is from Ref. [93].

3.5. HIFiRE ground tests

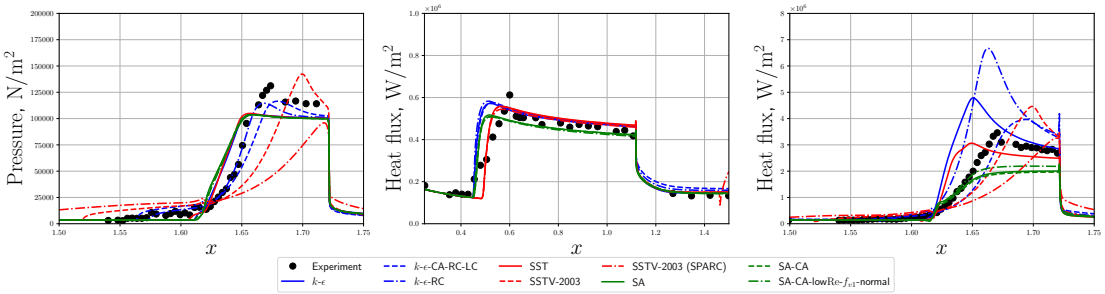
We consider runs 30 and 46 of the HIFiRE test geometry. The cases correspond to approximately Mach 7 flow with an expansion and a terminal 33° flare. Run 46 has a higher Reynolds number than run 30. Experimental data for wall quantities are available for both cases from CUBRC. Both cases again include a transition from laminar to turbulent flow. The transition location is determined from the experimental data and we manually enforce a trip by deactivating the turbulence model production term upstream of this location. A schematic of the problem geometry is shown in Figure 3-11.

Figure 3-12 shows results for wall QoIs, while Figure 3-13 shows the relative errors. Most of the previous observations are applicable to the HIFiRE cases. The baseline SA model performs particularly poor for this case, failing to predict separation for both Reynolds numbers. We emphasize that all models over-predict heat flux by more for the higher Reynolds number case (run 46). This is true for both the peak heat flux as well as for heat flux on the forebody. In particular, we emphasize the apparent difference in slope for wall heat flux on the forebody for run 46.

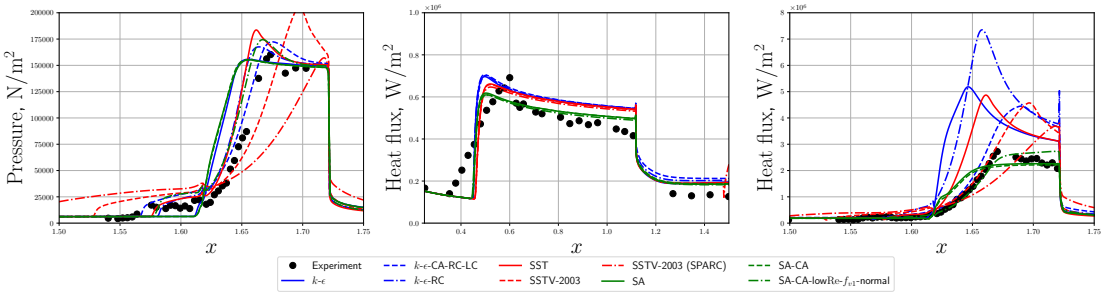
3.6. Summary of validation study

This chapter reported the results of various RANS models applied to a suite of flows pertinent to hypersonic aerodynamics. We assessed SPARC's baseline RANS models along with various corrections relevant to hypersonic flows. The high-level findings of the study are:

- The SA, SST, and $k-\epsilon$ models all fail to yield accurate predictions for both wall shear stress and wall heat flux in SBLI flows. The inclusion and re-calibration of existing corrections for hypersonic flows helps, but does not fully address, this issue.

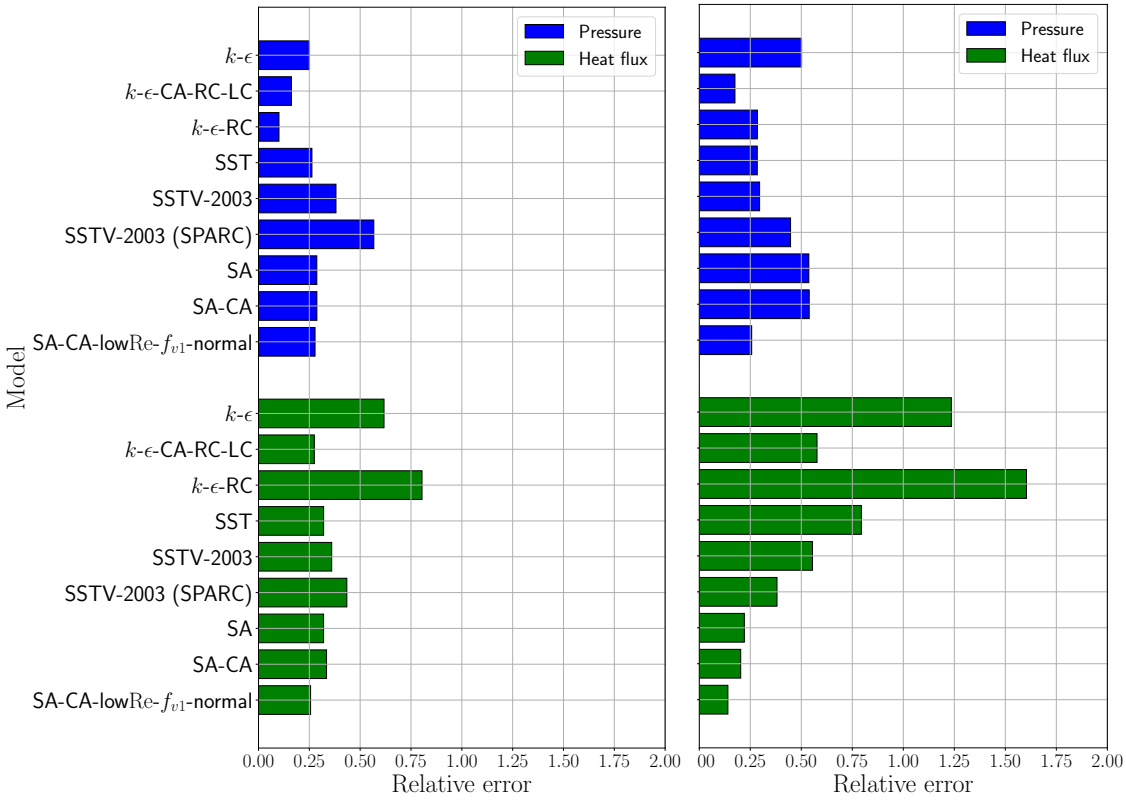


(a) Run 30.



(b) Run 46.

Figure 3-12. Predictions for HIFiRE ground test run 30 (top) and run 46 (bottom).



(a) Run 30 **(b) Run 46**

Figure 3-13. HIFiRE ground tests. Relative errors for wall QoIs.

- Neglecting TKE in the definition of total energy and the definition of the Reynolds stress in the energy equation degrades the predictive capability of the SSTV-2003 model in hypersonic flows. These assumptions impact high Mach number flat plate boundary layer flows. For the SBLI flows studied, these assumptions result in massive over-prediction of the separation region which in turn results in erroneous predictions for wall shear stress, wall pressure, and wall heat flux.
- The SSTV-2003 variant is inferior to the standard SST variant for hypersonic SBLI flows. The SSTV-2003 variant tends to over-predict the separation region in shock-induced separated flows. This over-prediction of separation can be attributed to the use of the strain invariant in the definition of the eddy viscosity and the use of the vorticity magnitude for approximating turbulent production.
- The SA model struggles to predict the size of the separation region in shock-induced SBLIs. The baseline variant usually under-predicts the size of the separation bubble and, for some cases, failed to predict it altogether.
- The SA model gives the best predictions for wall heat flux seemingly at the expense of poor shear stress predictions.
- The $k-\epsilon$ model massively over-predicts the size of the separation region. The use of a length-scale limiter helps address this issue but does not altogether eliminate it.
- Lastly, we highlight the lack of joint high-quality shear stress and heat flux data. Of the validation cases considered with SBLIs, only the Schülein cases and Holden compression ramps had experimental data for both wall shear and wall heat flux. The data for the compression ramp cases, however, have considerable noise. This issue is particularly relevant as there appears to be a decoupling between peak wall shear stress and peak heat flux; this result is supported in Ref. [78]. While $k-\epsilon$ based models massively over-predict wall heat flux, they appear to do better for wall shear stress. However, for the cases considered, there is not enough high-quality data to draw this conclusion with confidence. There is further a challenge that, for the available DNS simulations of hypersonic flows, there is often a significant mismatch between DNS-calculated wall shear stress and experiments [91]. It is our opinion that there is a need for more experimental data of hypersonic flow configurations with high quality measurements of both wall shear stress and heat flux.

This page intentionally left blank.

4. DATA-DRIVEN MODELS FOR THE REYNOLDS STRESS TENSOR

This chapter presents a data-driven turbulence modeling strategy aimed at developing improved models of the Reynolds stress tensor. The strategy outlined has three core aspects: (1) prediction for the discrepancy in the Reynolds stress tensor via machine learning (ML), (2) estimating uncertainties in ML model outputs via out-of-distribution detection, and (3) multi-step training strategies to improve feature-response consistency. Results are presented across a range of cases involving wall-bounded flows, jet flows, hypersonic boundary layer flows with cold walls, and hypersonic SBLIs. We find that one ML-enhanced turbulence model provides consistent improvements for numerous quantities-of-interest across all cases.

4.1. Introduction

We hypothesize that the inaccuracies of the various RANS models depicted in Chapter 3 can largely be attributed to the modeling of the Reynolds stress tensor and the turbulent heat flux. This chapter investigates the former and outlines a data-driven approach for improved modeling of the Reynolds stress tensor. Specifically, we develop an approach leveraging Lipschitz-continuous tensor-basis neural networks (TBNNs) and calibrated deep ensembles to model discrepancies in the Reynolds stress anisotropy tensor.

This chapter proceeds as follows. First, Section 4.2 gives a brief overview of the literature focused on improving predictions for the Reynolds stress tensor via machine learning techniques. Next, Sections 4.3-4.7 will outline the specific formulation employed in the present work. We will outline how we model discrepancies in the Reynolds stress tensor via TBNNs, our approach for quantifying model-form uncertainties and enforcing model smoothness, and how we generate weakly consistent training data via an iterative training procedure. Next, Section 4.9 will present results of the iterative training procedure for a suite of cases. Section 4.10 will show results on out-of-sample test cases.

4.2. Literature overview

Various bodies of work have targeted improving RANS models via data-driven improvements to models for the Reynolds stress tensor. These approaches attempt to build better models for the Reynolds stress tensor with the idea that improved Reynolds stress predictions will propagate

through to improved QoI predictions. Relevant to the present work, Ling et al. pioneered *tensor-basis neural networks* (TBNNs) [46]. TBNNs attempt to learn the anisotropic component of the Reynolds stress tensor via a basis expansion of invariant tensors employed by Pope [65] in his generalized eddy viscosity hypothesis; neural networks are employed to learn the coefficients of this basis as a function of field invariants. TBNNs have been shown to be capable of improved predictions of separated flows, secondary flows in ducts, and flows with curvature. Recent work has extended TBNNs to scalar flux modeling [51, 50]. Around the same time as the development of TBNNs, Xiao et al. developed a similar data-driven strategy, termed physics-informed machine learning (PIML), for improving Reynolds stress predictions [95, 102]. Their approach differs slightly from Ling et al. in that they (1) model terms pertaining to an eigen-decomposition of the Reynolds stress tensor, (2) they model a discrepancy in the turbulent kinetic energy, and (3) they attempt to model only the *discrepancy* in the Reynolds stress (e.g., the difference between a standard RANS model prediction and the truth value), opposed to the entire Reynolds stress. This approach has additionally been shown to provide improved predictions on duct and separated flows. Recent work by Peters et al. has proposed a similar approach in where they model discrepancies to the Reynolds stress tensor in the mean strain-rate tensor eigenframe [63].

The aforementioned bodies of work employ difficult-to-interpret ML techniques for learning improvements to the Reynolds stress tensor. Another body of work has sought to obtain more interpretable corrections. For instance, Weatheritt, Sandberg, and colleagues have developed a framework based on evolutionary algorithms to learn explicit expressions for the anisotropy tensor (discrepancy) [97, 96, 98, 107]. Opposed to TBNNs, and PIML, this evolutionary approach has the advantage that it results in tangible models that can be written down and analyzed. We refer the interested reader to the review [17] for a more complete survey of the literature.

Here, we investigate TBNNs as our primary vehicle for improving predictions for the Reynolds stress tensor. We employ TBNNs opposed to, e.g., evolutionary algorithms due to the following observations and hypotheses:

- TBNNs employ neural networks, which are more expressive than, e.g., symbolic regression algorithms which require a pre-defined dictionary of terms.
- While symbolic regression algorithms result in models that are, in theory, interpretable, the functional forms of models that have been discovered thus far are quite complicated and not amenable to interpretation.
- We prefer to not modify the discrepancy in the turbulent kinetic energy, as is done in PIML, due to concerns about generalizability. Specifically, we hypothesize that learning a generic correction to the turbulent kinetic energy is difficult and will require more data than are available.

We emphasize that we do not discount other approaches and that, while we investigate TBNNs, what follows can be applied to other modeling formulations.

4.3. Tensor-basis neural networks

We aim to improve Reynolds stress predictions via TBNNs [46]. In our approach, we augment a Reynolds stress model with an anisotropy-based discrepancy term. The anisotropy tensor a_{ij} is modeled as

$$a_{ij} \approx a_{ij}^{\text{RANS}} + \Delta a_{ij}^{\text{ML}},$$

where

$$a_{ij} = \frac{-\tau_{ij}}{\overline{\rho u_k'' u_k''}} - \frac{1}{3}\delta_{ij}$$

is the anisotropy tensor, δ_{ij} is the Kronecker delta, $a_{ij}^{\text{RANS}} = \frac{\tau_{ij}^{\text{RANS}}}{2\bar{\rho}\tilde{k}} - \frac{1}{3}\delta_{ij}$ is the anisotropy tensor predicted by a standard RANS model, and $\Delta a_{ij}^{\text{ML}}$ is an ML correction. For the functional form of the correction, we employ a TBNN [46, 65] such that

$$\Delta a_{ij}^{\text{ML}} = \sum_{i=1}^{K_{\text{max}}} \mathbf{g}^k(\boldsymbol{\lambda}) \mathcal{T}_{ij}^k,$$

where $\boldsymbol{\lambda}$ are invariant features, $\mathcal{T}^k : \Omega \rightarrow \mathbb{R}^{3\times 3}$, $k = 1 \ldots, K_{\text{max}}$ are tensors, Ω is the physical domain, and $\mathbf{g}^k : \boldsymbol{\lambda} \mapsto \mathbf{g}^k(\boldsymbol{\lambda})$, $k = 1, \ldots, K_{\text{max}}$ are ML regression models trained from simulation data. We highlight that, unlike Ref [46], we employ TBNNs to only model the discrepancy in the Reynolds stress tensor. We employ a truncated version of the basis employed by Pope [65] with $K_{\text{max}} = 4$ comprising the four basis tensors:

$$\mathcal{T}_{ij}^1 = S_{ij}^*,\ \mathcal{T}_{ij}^2 = S_{ik}^*\Omega_{kj}^* - \Omega_{ik}^* S_{kj}^*,\ \mathcal{T}_3 = S_{ij}^* S_{jk}^* - \frac{1}{3}\delta_{ij}\ \text{Tr}(S^{*2}),\ \mathcal{T}^4 = \Omega_{ik}^*\Omega_{kj}^* - \frac{1}{3}\delta_{ij}\ \text{Tr}(\Omega^{*2}) \tag{4.3.1}$$

where $S_{ij}^* = \left(S_{ij} - \frac{1}{3}\frac{\partial \tilde{u}_k}{\partial x_k}\delta_{ij}\right)\tilde{k}/\tilde{\epsilon}$, $\Omega_{ij}^* = \Omega_{ij}\tilde{k}/\tilde{\epsilon}$ are non-dimensional strain and rotation rate tensors, respectively, $\Omega_{ij} = \frac{1}{2}\left(\frac{\partial \tilde{u}_i}{\partial x_j} - \frac{\partial \tilde{u}_j}{\partial x_i}\right)$ is the rotation rate tensor, and we have used the notation $Q^2 = Q_{ij}Q_{jk}$ for a tensor $Q \in \mathbb{R}^{3\times 3}$. We use only the first four tensor bases since higher-order bases have cubic terms which may introduce undesirable numerical properties of the model. We use four non-dimensional features in our model. The first two features associate with invariants of the tensor basis while the second two features are measures of the local Reynolds number,

$$\boldsymbol{\lambda} = \left[\ \text{Tr}(S^{*2}) \quad \text{Tr}(\Omega^{*2}) \quad \log(\mu_t/\mu) \quad 3\tanh\left(\tfrac{1}{3}d_{\text{wall}}\frac{\sqrt{k}}{50\nu}\right)\right]^T,$$

where d_{wall} is the distance to the wall. We note that the final feature is designed to have a smooth threshold at a value of 3.

4.3.1. Analysis: TBNN simplifications for a parallel flow

A zero-pressure gradient turbulent boundary layer is a foundational physics for RANS modeling. To gain insight into the behavior of the TBNN in this setting, we consider reduction of the tensor-basis neural network when applied to a parallel flow, e.g., a channel. In this setting, all gradients are zero except for those in the wall-normal (here, x_2) direction. The strain and rotation tensors reduce to

$$S_{ij} = \frac{1}{2}\begin{bmatrix} 0 & \frac{\partial \tilde{u}_1}{\partial x_2} & 0 \\ \frac{\partial \tilde{u}_1}{\partial x_2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \Omega_{ij} = \frac{1}{2}\begin{bmatrix} 0 & \frac{\partial \tilde{u}_1}{\partial x_2} & 0 \\ -\frac{\partial \tilde{u}_1}{\partial x_2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The tensor basis (4.3.1) simplifies to

$$\mathcal{T}_1 = \frac{k}{2\epsilon}\begin{bmatrix} 0 & \frac{\partial \tilde{u}_1}{\partial x_2} & 0 \\ \frac{\partial \tilde{u}_1}{\partial x_2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathcal{T}_2 = \frac{k^2}{2\epsilon^2}\begin{bmatrix} -\left(\frac{\partial \tilde{u}_1}{\partial x_2}\right)^2 & 0 & 0 \\ 0 & \left(\frac{\partial \tilde{u}_1}{\partial x_2}\right)^2 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\mathcal{T}_3 = \frac{k^2}{12\epsilon^2}\begin{bmatrix} \left(\frac{\partial \tilde{u}_1}{\partial x_2}\right)^2 & 0 & 0 \\ 0 & \left(\frac{\partial \tilde{u}_1}{\partial x_2}\right)^2 & 0 \\ 0 & 0 & -2\left(\frac{\partial \tilde{u}_1}{\partial x_2}\right)^2 \end{bmatrix}, \mathcal{T}_3 = -\mathcal{T}_4.$$

The above result demonstrates that the second, third, and fourth tensor bases only impact the normal components of the Reynolds stress. Although not shown, the sixth tensor basis modifies the shear component of the Reynolds stress. The remaining tensors impact the normal components of the Reynolds stress or are zero. This has two important repercussions:

1. Consider a data point drawn from a parallel flow dataset. For this data point, the only non-zero velocity gradient will be $\frac{\partial \tilde{u}_1}{\partial x_2}$, and the gradient of the "shear" anisotropy component Δa_{12} — which is the only anisotropy component to enter the governing equations — with respect to the coefficients for tensor bases 2 through 4 will be identically zero. This result implies that, in training, modeling the normal components of the anisotropy tensor does not "interfere" with the shear components so long as the model has high enough capacity.

2. For a parallel flow, the normal components of the Reynolds stress do not impact the governing equations. As a result, the TBNN tensors 2 through 4 will have no impact.

The above result holds exactly for a parallel flow. The result pertaining to training will additionally hold for canonical flows like shear layers and boundary layers under the assumption that $\frac{\partial \tilde{u}_1}{\partial x_2}$ dominates all other gradients. Note that, for high-speed boundary layers, the normal components of the Reynolds stress have a non-negligible impact on the flow, and thus higher-order tensor bases will impact the solution for these flows. This result will be detailed later in this report.

4.4. Ensembles for uncertainty quantification and out-of-distribution detection

Having described the structural form of our machine-learned corrections, we now describe how the discrepancy terms are modeled. We aim to achieve two primary goals with our models:

- **High capacity for predicting test points that are in-distribution with respect to the training set.** We want to ensure that our model is of a high enough fidelity that it can capture complex relationships in high-dimensional feature spaces when it is deployed on testing data that are drawn from the same data-generating distribution as our training data (e.g., the model must be accurate in interpolation).

- **Robustness to out-of-distribution and poorly learned data regimes.** In practical scenarios an ML model will be deployed on data that lie outside of the convex hull (e.g., are extrapolative) of the training set and/or lie in a feature space region where training data are extremely sparse such that the model did not learn an accurate feature-response map. In these scenarios it is critical to identify whether or not the ML model is accurate and, in the case where it is not accurate, make an appropriate decision on how to include the ML prediction to maintain robustness.

Towards this end, we propose a Bayesian-type approach leveraging ensembles of neural networks. Motivated from the concept of "deep ensembles" [43] the approach relies on the empirical result that an ensemble of over-parameterized neural networks tends to provide a similar prediction for testing data that are drawn from the training-data-generating distribution but simultaneously provide different predictions for out-of-distribution testing data. The variance of the ensemble can be used as an empirical error indicator. We now describe this approach within the context of tensor-basis neural networks for the anisotropy discrepancy.

For modeling the Reynolds stress, we learn tensor-basis neural networks mapping input features to the anisotropy discrepancy and a scalar variance,

$$\mathcal{NN} : (\boldsymbol{\lambda}; \boldsymbol{\theta}) \mapsto (\Delta a^{\mathrm{ML}}, \sigma^2)$$

where $\sigma^2 \in \mathbb{R}^+$ is a scalar variance. The ensemble approach trains M such networks. For each network, we optimize the log-likelihood as given by

$$\boldsymbol{\theta} = \underset{\boldsymbol{\theta}^* \in \mathbb{R}^{N_{\boldsymbol{\theta}}}}{\arg\min} \sum_{k=1}^{N_{\mathrm{train}}} \frac{\left(\Delta a_{ij}^{\mathrm{ML}}\left(\boldsymbol{\lambda}^k, \boldsymbol{\theta}^*\right) - \Delta a_{ij}^k\right)^2}{2\sigma^2\left(\boldsymbol{\lambda}^k, \boldsymbol{\theta}^*\right)} + \frac{\sigma^2\left(\boldsymbol{\lambda}^k, \boldsymbol{\theta}^*\right)}{2},$$

where N_θ denotes the total number of weights and biases. The mean and variance of the networks is then given by

$$\overline{\Delta a}^{\mathrm{ML}} = \frac{1}{M}\left(\sum_{i=1}^{M} \Delta a^{\mathrm{ML}}(\boldsymbol{\lambda};\boldsymbol{\theta}_i)\right),$$
$$\widehat{\Delta a}^{\mathrm{ML}} = \frac{\hat{C}}{M}\sum_{i=1}^{M}\left[\frac{1}{9}\sum_{l=1}^{3}\sum_{m=1}^{3}\left(\Delta a_{lm}^{\mathrm{ML}\,2}(\boldsymbol{\lambda};\boldsymbol{\theta}_i) - \overline{\Delta a}_{lm}^{\mathrm{ML}\,2}\right)\right] + \frac{1}{M}\sum_{i=1}^{M}\sigma^2\left(\boldsymbol{\lambda},\boldsymbol{\theta}_i\right),$$

where $\boldsymbol{\theta}_i$ denote the weights and biases for the ith network, $\overline{\Delta a}^{\mathrm{ML}} \in \mathbb{R}^{3\times 3}$ is the mean anisotropy tensor predicted by the networks, $\widehat{\Delta a}^{\mathrm{ML}} \in \mathbb{R}^{+}$ is a mean variance of the anisotropy tensors predicted by the networks, and $\hat{C} \in \mathbb{R}^{+}$ is a constant used to calibrate the ensemble variance. We will discuss selection of this constant shortly.

Given a predicted mean and variance, the correction to the anisotropy tensor in our CFD solver is set to be

$$\Delta a_{ij}^{\mathrm{ML}} = \frac{\sigma_{\mathrm{prior}}^2}{\sigma_{\mathrm{prior}}^2 + \widehat{\Delta a}^{\mathrm{ML}}}\overline{\Delta a}_{ij}^{\mathrm{ML}}. \tag{4.4.1}$$

The approximation (A.0.1) is inspired by a Bayesian formulation. For notational simplicity we define the weighting in the above in terms of a "confidence measure" $C^* \in [0,1]$ where

$$C^* = \frac{\sigma_{\mathrm{prior}}^2}{\sigma_{\mathrm{prior}}^2 + \widehat{\Delta a}^{\mathrm{ML}}}. \tag{4.4.2}$$

We make the following comments on the formulation given in Eq. (A.0.1):

- In the limit $\widehat{\Delta a}^{\mathrm{ML}} \ll \sigma_{\mathrm{prior}}$, then $C^* \to 1$ and $\Delta a^{\mathrm{ML}} \to \overline{\Delta a}^{\mathrm{ML}}$, i.e., the prediction for the anisotropy discrepancy tensor seen by the solver is given by the mean of the ensemble. This case corresponds to where we expect the ML models to be accurate.
- In the limit $\widehat{\Delta a}^{\mathrm{ML}} \gg \sigma_{\mathrm{prior}}$, then $C^* \to 0$ and $\Delta a^{\mathrm{ML}} \to 0$, i.e., we don't correct the standard anisotropy tensor at all. This case corresponds to where we expect the ML models to be inaccurate.
- We note that we employ a mean variance $\widehat{\Delta a}^{\mathrm{ML}}$ for the entire anisotropy tensor as opposed to a unique variance for each component. We do this in an effort to make our model more interpretable and to ensure that the resulting model for the anisotropy tensor is trace free, but employing a different variance for each component is a possible choice.
- The functional form of the update given in Eq. (A.0.1) is motivated from a Bayesian formulation for the anisotropy tensor with Gaussian assumptions and a zero prior.

Employing the discrepancy model in Eq. (A.0.1) requires specification of the scaling factor $\hat{C}$ and prior confidence σ_{prior}. We determine the constant $\hat{C}$ using the following process:

1. Model the error in our ensemble of networks as

$$(\Delta a_{lm} - \overline{\Delta a}^{\text{ML}}_{lm})^2 \approx \frac{\hat{C}}{M} \sum_{i=1}^{M} \left[\left(\Delta a^{\text{ML}^2}_{lm}(\boldsymbol{\lambda}; \boldsymbol{\theta}_i) - \overline{\Delta a}^{\text{ML}^2}_{lm} \right) \right].$$

2. Fit the constant $\hat{C}$ after training the network ensemble by evaluating the prediction error across the validation set.

We set $\sigma_{\text{prior}} = 1.0$. The result of this process is (approximate) control over how much we trust the prediction of the ensemble.

4.5. Lipschitz-continuous neural networks

While our ensemble approach can improve the performance of the ML model for extrapolative data, feed-forward neural networks can still produce non-smooth outputs which may deteriorate the convergence of the RANS solver (e.g., see the summary in Ref [94]). Some previous works have found it necessary to smooth the anisotropy predictions obtained from neural networks for the RANS solver to converge [2, 21]. Rather than applying a smoothing operation on the neural network outputs, we enforce that the input–output map learned by the network is Lipschitz continuous. A real-valued function $f : \mathbb{R} \to \mathbb{R}$ is said to be k-Lipschitz continuous if, $\forall x \in \mathbb{R}, y \in \mathbb{R}$,

$$|f(x) - f(y)| \leq k|x - y|.$$

Lipschitz continuous functions are appealing because the smoothness of the output can be bounded by the smoothness of the input.

We construct Lipschitz continuous neural networks using spectral normalization [53]. In spectral normalization, the weights of each layer are implicitly normalized by their spectral norm,

$$\mathbf{w}^i_* = \frac{\mathbf{w}^i}{\zeta(\mathbf{w}^i)}$$

where

$$\zeta(\mathbf{w}^i) \equiv \max_{\|\boldsymbol{x}\|_2 \leq 1} \|\mathbf{w}^i \boldsymbol{x}\|.$$

Leveraging the fact that common activation functions (e.g., ReLU, tanh) are 1-Lipschitz, the use of spectral normalization results in neural networks that are also 1-Lipschitz. To enable a more flexible

framework, we embed a hyper-parameter into our neural networks by multiplying the normalized weights by a constant k_{lip} at each layer,

$$f_{\text{lip}} : (\boldsymbol{\lambda}, \boldsymbol{\theta}) \mapsto g\left(\cdot; k_{\text{lip}}\mathbf{w}_*^{N_{\text{layer}}}, \mathbf{b}^{N_{\text{layer}}}\right) \circ \cdots \circ g\left(\cdot\ k_{\text{lip}}\mathbf{w}_*^1, \mathbf{b}^1\right).$$

The constant k_{lip} is a hyper-parameter: higher values will result in more expressive networks with "rougher" outputs, while smaller values will result in smooth networks that are less expressive. For notational simplicity, we use $\boldsymbol{\theta} = [\mathbf{w}^*, \mathbf{b}]$ to denote the combined weights and biases from here on out.

4.5.1. Automatic selection of Lipschitz constant

We determine the constant k_{lip} from the training data. Specifically, we loop through input-output training pairs, compute a "local" Lipschitz constant on these pairs, and set the total Lipschitz-constant of the network based on quantiles of the local Lipschitz constants. A lower quantile will result in a lower Lipschitz constant. Algorithm 1 details the algorithm for a simple neural network.

Computing an appropriate Lipschitz constant is more difficult for TBNNs since we aim to learn the coefficients of the tensor basis, and we generally do not have measurements of these coefficients. In the present work we perform a least-squares fit to determine the optimal coefficients, and then use these coefficients to identify the Lipschitz constant.

Algorithm 1 Basic algorithm for estimating the Lipschitz constant

Input: quantile, q
for $i = 1, \ldots, N_{\text{train}}$ **do**
  for $j = 1, \ldots, N_{\text{train}}$ **do**
    $(q_1, q_2) \leftarrow (\boldsymbol{\eta}_i, \boldsymbol{\eta}_j)$ ▷ Sample (normalized) features
    $(r_1, r_2) \leftarrow (y_i^{\text{ML}}, y_j^{\text{ML}})$ ▷ Sample (normalized) response
    $k_{ij} = \frac{\|r_1 - r_2\|_2^2}{\|q_1 - q_2\|_2^2}$ ▷ Compute local Lipschitz constant and store
  end for
end for
$k \leftarrow \text{quantile}(\boldsymbol{k}, q)$ ▷ Compute constant based on quantiles

4.6. Multi-step training framework for feature consistency

We now outline the process for training the model weights $\{\mathbf{w}_i\}_{i=1}^M$. One of the primary challenges in training an ML turbulence model is feature and response consistency. This issue has been discussed in numerous places and we provide only a brief summary here.

We first discuss feature consistency. Consider a feature of the form S^*_{ij} extracted from a high-fidelity dataset. This feature depends on the turbulent kinetic energy and turbulent dissipation due to the non-dimensionalization $S^*_{ij} = S_{ij}k/\epsilon$. Unfortunately, even if the velocity fields match almost exactly, the turbulent kinetic energy and turbulent dissipation in a RANS simulation often differ significantly from their truth values. As a result, the RANS version of a feature S^*_{ij} can differ significantly from its high-fidelity counterpart. An ML model trained on input features extracted from a high-fidelity dataset can thus fail when deployed in a RANS solver as it is receiving different features than what it was trained on.

The second challenge is response consistency. To illustrate this challenge, consider the case where we have trained a perfect machine-learned model for the anisotropy discrepancy tensor Δa_{ij}. The anisotropy discrepancy enters the momentum and energy equations through the form of a Reynolds stress discrepancy, $\Delta\tau_{ij} = -2\rho k\Delta a_{ij}$. If the turbulent kinetic energy in the RANS model differs from that in the high-fidelity data, then a perfect model for the anisotropy tensor will still end up being imperfect for the Reynolds stress.

Various approaches have been pursued to address the above including learning solutions about baseline RANS solutions [46], modified cost functions [89], approaches that learn a discrepancy in the turbulent kinetic energy [95, 102] , approaches that directly learn the Reynolds stress rather than the anisotropy [95], solves for the RANS auxiliary equations with a specified high-fidelity velocity [50, 75], and iterative training procedures [89]. In the present work we employ an iterative training procedure similar to [89]. We employ this approach primarily because it was straightforward to implement with our ML models and solver. The steps for our iterative approach are as follows:

- For each training case, do:

    1. Setup and solve a baseline RANS computation corresponding to a high-fidelity dataset.

    2. Learn a discrepancy model for the anisotropy tensor, where the input features of the model are computed from the baseline RANS solution and the response is computed from the high-fidelity data.

    3. Compute a correction to the anisotropy tensor by querying the learned models about the baseline RANS solution. Freeze these correction fields, and re-converge the solver.

    4. Learn new discrepancy models for the anisotropy tensor, where the input features of the model are extracted from the ML–RANS solution computed in step 3 and the response is still computed from the high-fidelity data.

    5. Compute a new correction to the anisotropy tensor by querying the models learned in step 4 about the ML–RANS solution computed in step 5. Freeze these correction fields, and re-converge the solver. We emphasize that this new correction that is learned is quantifying the discrepancy between the anisotropy tensor predicted from the *baseline*

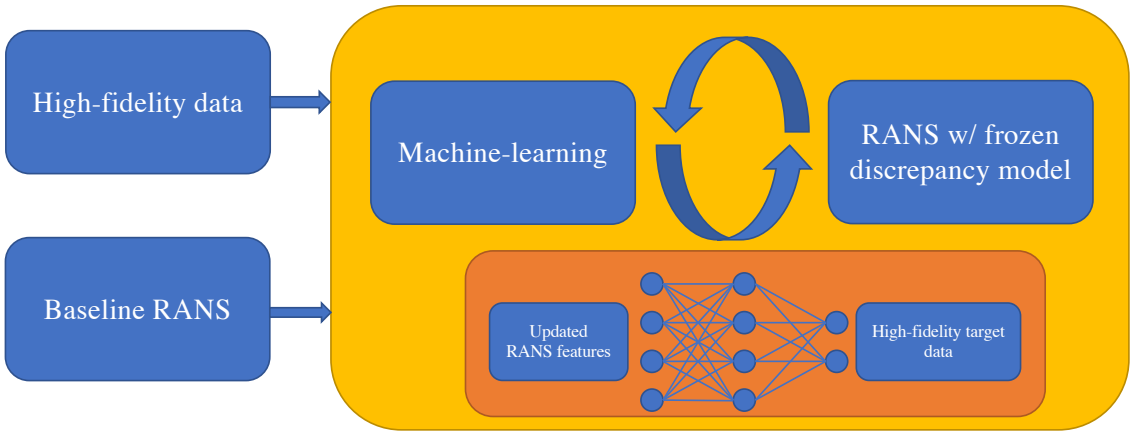


Figure 4-1. Depiction of the iterative training process employed for each case.

RANS model (with improved input features coming from the new fields computed in Step 3) and the truth data.

6. Repeat steps 3-5 until convergence or the maximum number of iterations has been reached. In the present work full convergence was never reached and we terminated the loop after 10 iterations.

- Train one global ML model for all cases with the ML–RANS solutions computed in Step 5 as input features and the high-fidelity data as responses.

Figures 4-1 and 4-2 depict the above steps. Figure 4-1 shows the iterative training process employed for each case, while Figure 4-2 shows the global training process used to develop the final machine-learned model.

Before proceeding we highlight that an outstanding issue with our approach is that, while it improves feature consistency, it does not fully address response consistency. Although the iterative solution process tends to improve predictions for turbulent kinetic energy, there is ultimately still a discrepancy between the turbulent kinetic energy in the RANS equations vs. a high-fidelity dataset. As a result the magnitude of the Reynolds stress correction can still be erroneous.

4.7. Implementation in SPARC

The TBNN discrepancy models for the anisotropy tensor are implemented in Sandia's parallel aerodynamics and re-entry code (SPARC). At the time of this writing, these models are not

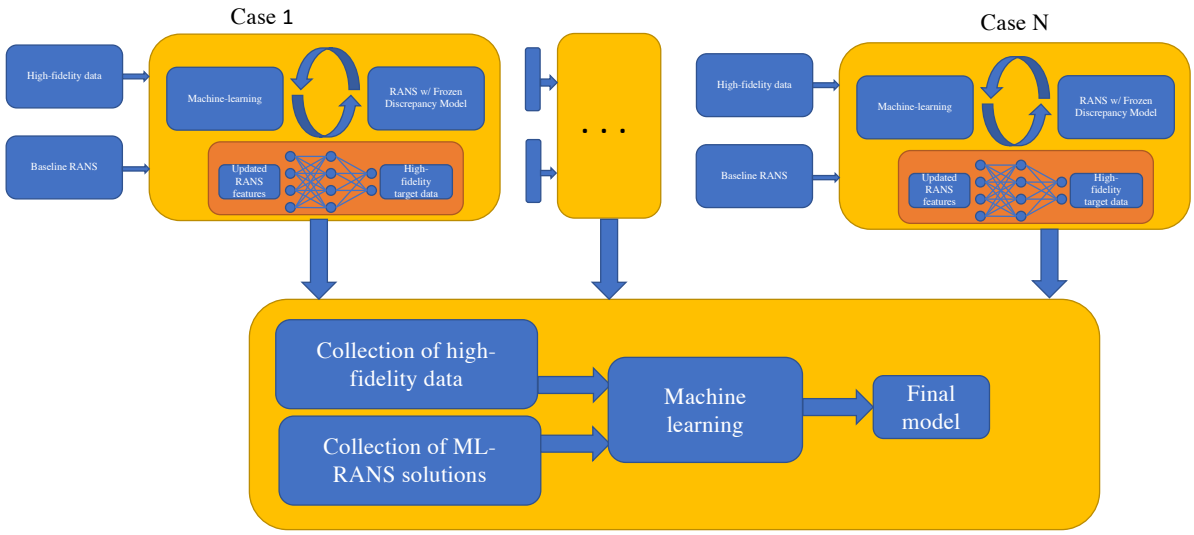


Figure 4-2. Depiction of the global iterative training process employed to construct our final ML model.

implemented on the main branch. SPARC supports various discretizations (finite-volume, finite-difference, finite-element), and in the present work we study the performance of our models within SPARC's second-order cell-centered finite volume method. The anisotropy discrepancy tensor is treated within the viscous flux routine.[1] We currently do not modify any of the Jacobian entries and as such treat the discrepancy model explicitly in the solver; implicit treatment of the discrepancy models is an area of active work. Lastly, we note that in practice we only call the ML model every n_{skip} Newton iterations to accelerate model performance. In the examples presented in this work we use $n_{\text{skip}} = 20$.

4.8. Training

This section details the training data employed in this work and the results of the iterative training procedure applied to each training case.

4.8.1. Training datasets, data cleaning, and data balancing

Presently we employ 12 datasets for training: channel flows at $Re_\tau = 180, 395$, and 590 [33], a duct flow at Re $= 3500$ [64], flow over a periodic hill [20], three hypersonic boundary layers at cold wall conditions [105], and four hypersonic cases over forward-facing walls (FFW) and backward-facing walls (BFW) which correspond to the $\alpha = 0.2$ and $\alpha = 1.0$ cases from [57]. For the FFW and BFW

[1] We have explored implementation of the discrepancy model as both a source term and in the viscous flux routine and observed little difference. We default to the viscous flux routine so that the method is conservative.

Table 4-1. Description of datasets used for training

Dataset name	Dataset size	Simulation type	Dataset dimension
Channel at $Re_\tau = 180$	65	DNS	1D
Channel at $Re_\tau = 395$	129	DNS	1D
Channel at $Re_\tau = 590$	128	DNS	1D
Duct flow	10201	DNS	2D
Periodic hill flow	25088	LES	2D
ZPGFBL at M = 6, $T_w/T_r = 0.25$	188	DNS	1D
ZPGFBL at M = 6, $T_w/T_r = 0.76$	177	DNS	1D
ZPGFBL at M = 14, $T_w/T_r = 0.18$	227	DNS	1D
FFW steep ramp at M = 4.9, $T_w/T_r = 0.91$	320x5	DNS	1.5D
FFW shallow at M = 4.9, $T_w/T_r = 0.91$	320x5	DNS	1.5D
BFW steep at M = 4.9, $T_w/T_r = 0.91$	320x5	DNS	1.5D
BFW shallow at M = 4.9, $T_w/T_r = 0.91$	320x5	DNS	1.5D

cases, the wall shapes were designed such that the $\alpha = 1.0$ wall shapes induced a reasonably strong pressure gradient while not inducing flow separation while the $\alpha = 0.2$ cases were designed such that the flow would separate. Table 4-1 summarizes these data. We note that all of these data are available through the NASA turbulence modeling resource (NASA TMR) and links therein with the exception of the duct flow data.

Table 4-1 highlights that the different datasets are of disparate size. The channel flow and hypersonic boundary layer data comprise one-dimensional wall-normal profiles, while the duct flow and periodic hill are two-dimensional datasets. A naive compilation of these various datasets may result in an ML model that more accurately fits the two-dimensional flows. To minimize this possibility we perform empirical data balancing where we duplicate the one-dimensional datasets to increase their prevalence in the loss function used for training the final ML model. In the present work we replicate each channel data set 8 times so that we end up with around 2500 data points from the channels, we replicate each hypersonic boundary layer data 5 times so that we end up with around 3000 training data points from the hypersonic boundary layers, and we replicate the hypersonic FFW and BFW cases just once so that each case has 1600 training points. The reader may note that there are still significantly more data in the periodic hill and duct flow datasets. This was done intentionally as the interesting physics in the periodic hill and duct flow datasets are contained to a smaller regime of the flow. We additionally note that we did not observe significant sensitivity to the data balancing, and we expect that different choices will result in similar solutions. Lastly, we note that more automated procedures exist, e.g., Barone et al. [5], and employing these techniques will be a subject of future work.

Table 4-2. Hyper-parameters employed in the TBNN and variable Prandtl number models.

Model	ℓ^2 regularization	Initial learning rate	Epochs	Batch size	Learning rate decay
TBNN	10^{-5}	2×10^{-3}	500	50	0.9991

4.8.2. Training details and architectures

Our tensor basis neural networks are implemented in PyTorch. We employ a standard 80/20 training-validation split on the data with standard normalization applied to the input features (i.e., we center each input feature by its mean across the training set and scale by its standard deviation across the training set). We do not normalize the response values. We train an ensemble of 10 Lipschitz-continuous neural networks. All networks employ tanh as the activation function, have three hidden layers, and have 30 nodes per hidden layer. We use an 85% quantile in computation of the Lipschitz constant. The remaining hyper-parameters are given in Table 4-2.

4.9. Training results

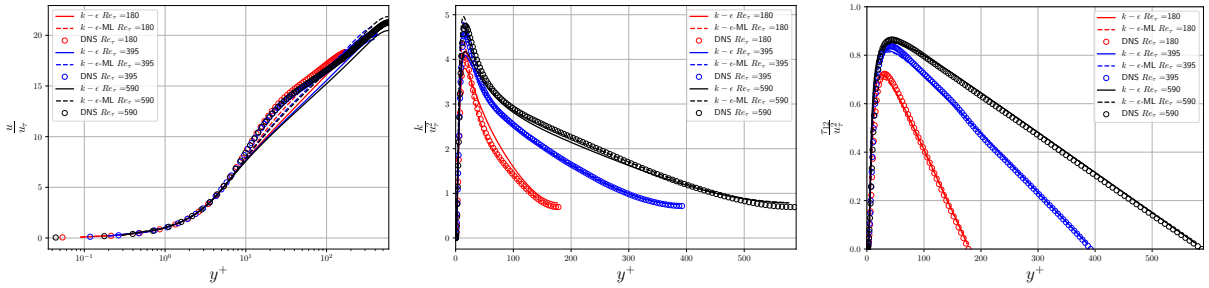
We now present results of the iterative training procedure detailed in Section 4.6 on a set of training cases. We emphasize that in each case we are recursively training an ensemble of networks to learn the anisotropy tensor discrepancy and propagating the resulting corrections through the model. The results of these training cases will be used to create a single global ML model. **In all cases, we employ the $k-\epsilon$ model with the Catris and Aupoix correction and a rapid compression correction with constant $C_\epsilon = -1.0$ as our baseline model.**

4.9.0.1. Channel flows

We first consider results of the iterative training procedure on the three channel flows. All three cases are simulated as one-dimensional flows with a constant pressure gradient forcing term. Figure 4-3 shows results for velocity profiles, turbulent kinetic energy, and Reynolds shear stress for all three Reynolds numbers. Table 4-3 tabulates the relative error between the RANS and DNS for each profile. The relative error is defined as

$$e_q = \int_0^{\mathrm{Re}_\tau} \frac{\left(q^{\mathrm{RANS}} - q^{\mathrm{DNS}}\right)^2}{\left(q^{\mathrm{DNS}}\right)^2} dy^+$$

for $q = u^+, k^+$, and τ_{12}^+. In all cases $k-\epsilon-$ML results in improved Reynolds stress predictions over the $k-\epsilon$ model. We see that this improvement propagates through to improved velocity predictions in every case and improved turbulent kinetic energy predictions in two of the three cases.



(a) Velocity **(b) Turbulent kinetic energy** **(c) Reynolds shear stress.**

Figure 4-3. Channel flow. Results of iterative training procedure.

Table 4-3. Relative errors for channel flow quantities of interest for three Reynolds numbers

	$\mathrm{Re}_\tau = 180$			$\mathrm{Re}_\tau = 395$			$\mathrm{Re}_\tau = 590$		
	e_{u^+}	e_k^+	$e_{\tau_{12}}^+$	e_{u^+}	e_k^+	$e_{\tau_{12}}^+$	e_{u^+}	e_k^+	$e_{\tau_{12}}^+$
k–ϵ	0.0789	0.0987	0.094	0.051	0.045	0.067	0.054	0.062	0.046
k–ϵ–ML	0.0324	0.0896	0.072	0.035	0.055	0.037	0.033	0.050	0.033

4.9.0.2. Duct flow

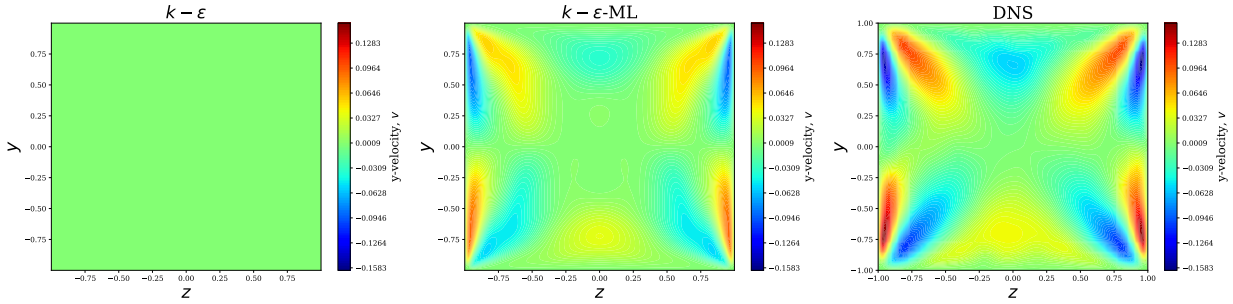
The duct flow is implemented as a two-dimensional periodic flow with a constant pressure gradient. The problem is simulated non-dimensionally on a 2×2 domain with a constant (non-dimensional) viscosity $\tilde{\mu} = 2.2{\times}10^{-3}$, a constant pressure gradient of magnitude 0.5, an isothermal wall boundary condition $\tilde{T}_{\text{wall}} = 178.57$, an initial density of $\overline{\rho} = 1$, and a gas constant of $R = 10$ with the equation of state $\overline{p} = \overline{\rho} R \tilde{T}$. Figure 4-4 shows results for the x_2 velocity and turbulent kinetic energy for the duct flow case while Table 4-4 tabulates QoI errors for the problem. In this case the error for a QoI q is defined as

$$e_q = \sum_{i=1}^{N} \frac{\left(q(\mathbf{x}_i)^{\text{RANS}} - q(\mathbf{x}_i)^{\text{DNS}}\right)^2}{\left(q(\mathbf{x}_i)^{\text{DNS}}\right)^2} \tag{4.9.1}$$

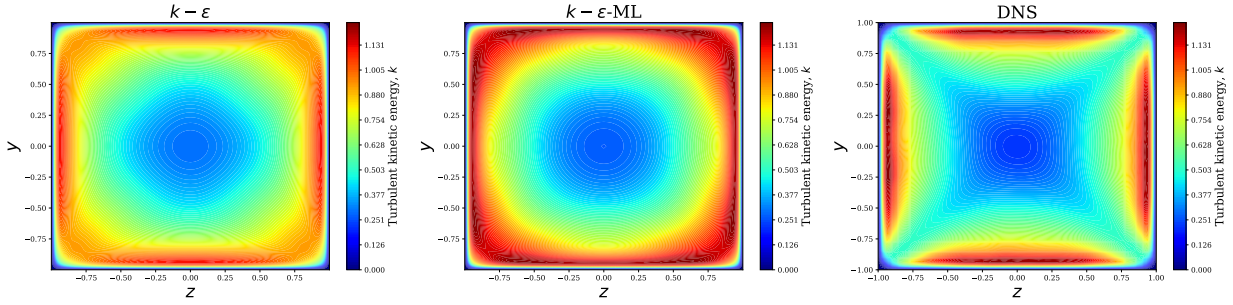
where $\mathbf{x}_i \in \mathbb{R}^2$, $i = 1, \ldots, N$ refer to the spatial locations at which the DNS data are available. We observe improved predictions for all Reynolds stress components but the a_{23} component. This improved prediction again propagates to improved predictions for the velocity.

Table 4-4. Duct flow. Relative errors for QoIs.

Model	e_u	e_v	$e_{a_{11}}$	$e_{a_{22}}$	$e_{a_{33}}$	$e_{a_{12}}$	$e_{a_{23}}$	e_k
k–ϵ	0.137	1.000	1.000	1.000	1.000	0.328	1.000	0.207
k–ϵ–ML	0.099	0.402	0.233	0.306	0.306	0.235	1.194	0.339



(a) x_2 velocity



(b) Turbulent kinetic energy

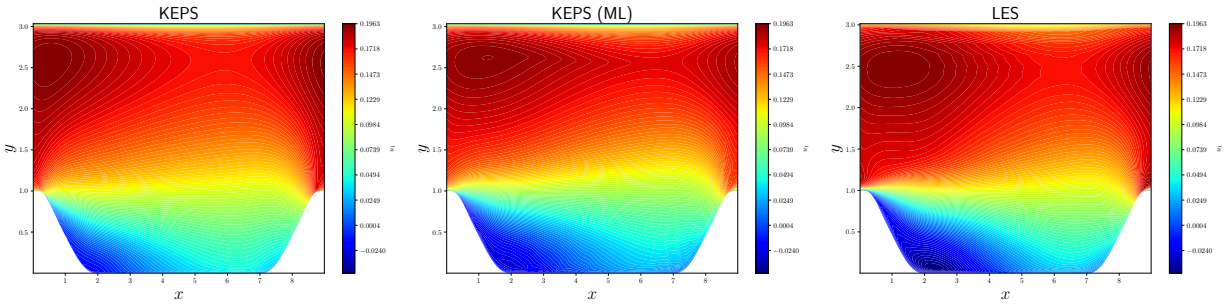
Figure 4-4. Duct flow. Results of iterative training procedure.

Table 4-5. Periodic hill flow. Relative errors for QoIs.

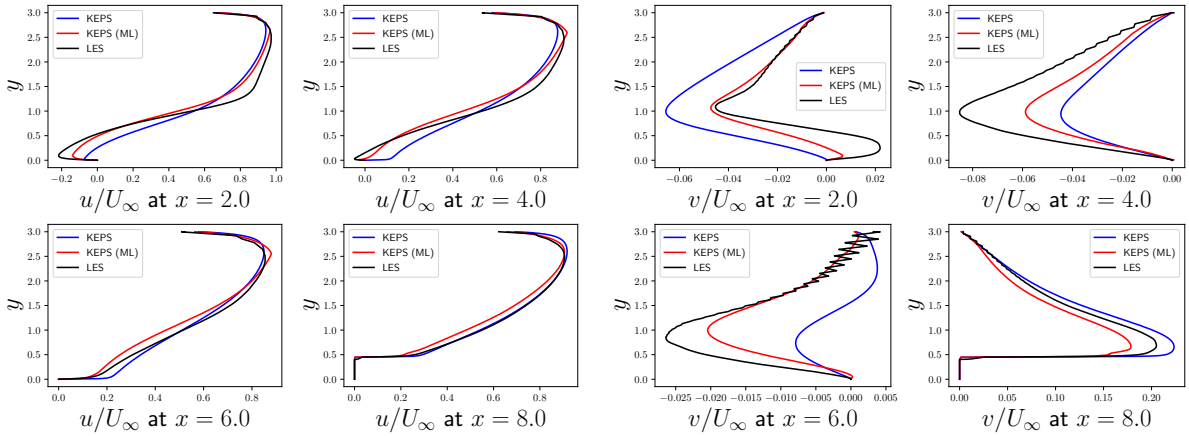
Model	e_u	e_v	$e_{a_{11}}$	$e_{a_{22}}$	$e_{a_{33}}$	$e_{a_{12}}$	e_k
$k-\epsilon$	0.163	0.469	0.309	0.642	0.290	0.596	0.245
$k-\epsilon-$ML	0.105	0.209	0.124	0.214	0.153	0.333	0.230

4.9.0.3. Periodic hill

The periodic hill is implemented as a periodic two-dimensional flow with a constant pressure gradient forcing. The flow is simulated non-dimensionally with viscosity $\tilde{\mu} = 1.888 \times 10^{-5}$, a constant pressure gradient $\overline{p}_x = 0.00033$, an isothermal boundary condition of $\tilde{T}_{\text{wall}} = 1/\gamma$ with $\gamma = 1.4$, a gas constant of $R = 1$ with the equation of state $\overline{p} = \overline{\rho} R \tilde{T}$, and an initial uniform density of $\overline{\rho} = 1$. Figure 4-5 shows results for x_1 velocity contours and velocity profiles as a function of y for various x locations. Table 4-5 tabulates QoI errors for the problem. In this case QoI errors are defined the same as in Eq. (4.9.1). We make the same observations as in the channel and duct flows. All components of the Reynolds stress tensor are computed more accurately by the ML model as compared to the standard model. These improved predictions propagate through to improved predictions for both velocity components as well as turbulent kinetic energy.



(a) x_1 velocity



(b) x_1 velocity profiles **(c) x_2 velocity profiles**

Figure 4-5. Periodic hill flow. Results of iterative training procedure. Note that the "wiggles" in the LES data for the v velocity are an interpolation artifact that are amplified by the small v-velocity magnitude.

Table 4-6. Free-stream conditions for hypersonic boundary layer cases.

	M6Tw025	M6Tw076	M14Tw018
u_∞ (m/s)	871.17	871.45	1883.74
ρ_∞ (kg/m^3)	0.0421	0.0414	0.0149
T_∞ (K)	54.17	54.139	44.802

Table 4-7. M6Tw025 hypersonic boundary layer. Relative errors after iterative training procedure.

Model	e_u^+	e_T	$e_{a_{11}}$	$e_{a_{22}}$	$e_{a_{33}}$	$e_{a_{12}}$	e_k	e_{c_f}	e_{c_h}
$k-\epsilon$	0.068	0.217	0.991	1.002	1.008	0.903	0.269	0.075	0.071
$k-\epsilon$–ML	0.049	0.208	0.236	0.848	0.922	0.903	0.293	0.036	0.020

4.9.0.4. Hypersonic boundary layers

Next we consider training results for three hypersonic boundary layer cases. The flows are simulated on a dimensional two-dimensional domain with a perfect gas with gas constant of $R = 287$ J $\cdot$ kg^{-1} $\cdot$ K^{-1} and heat capacity ratio $\gamma = 1.4$. The free-stream conditions for each case are selected so that flow conditions after the leading edge shock match the DNS conditions provided in Zhang et al. [105]; the free-stream conditions for each case are provided in Table 4-6.

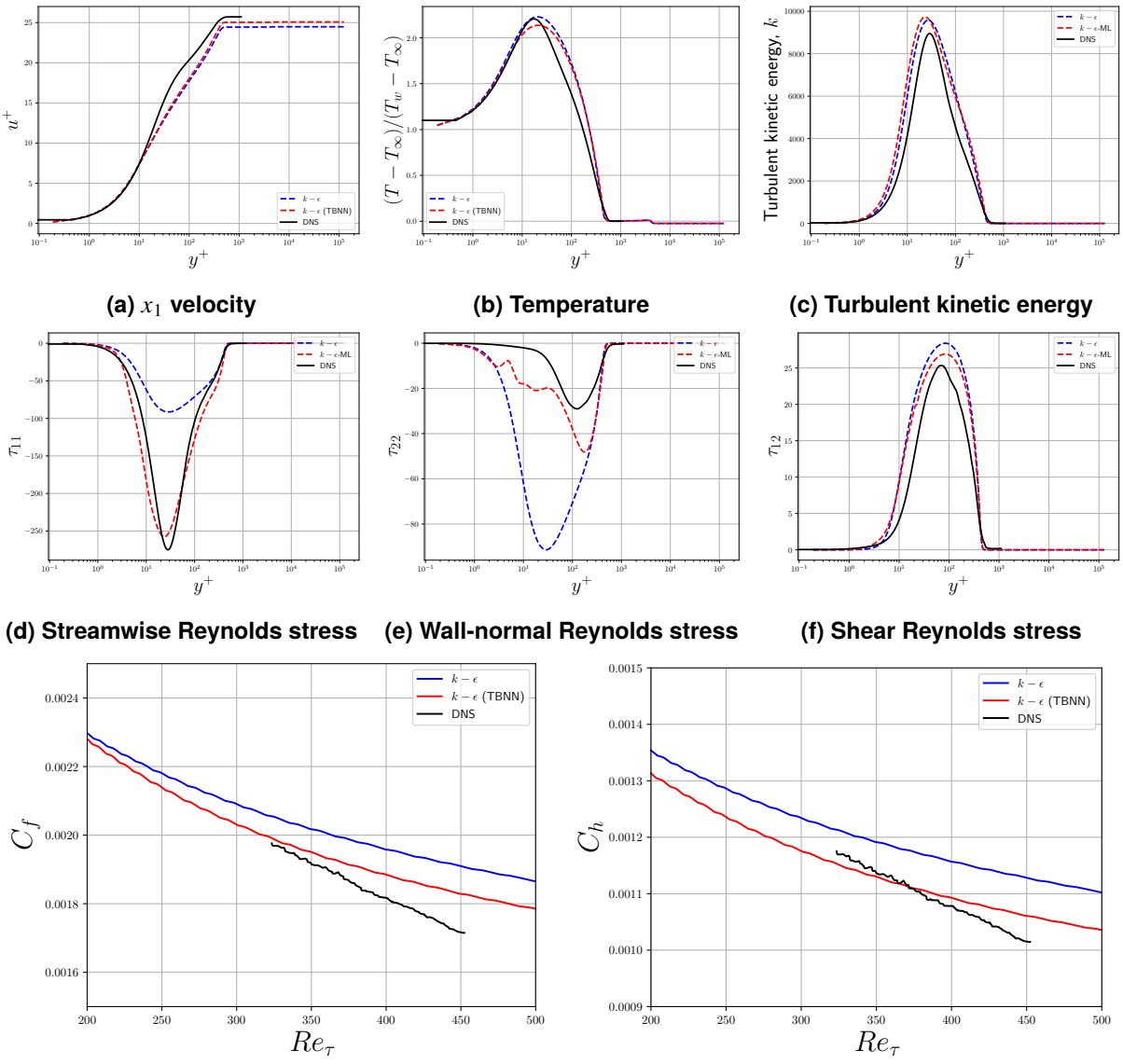
Figures 4-6-4-8 shows predicted QoIs for the M6Tw025, M6Tw076, and M14Tw018 cases, respectively, while Tables 4-7-4-9 tabulate the errors between the RANS predictions and reference DNS solutions. For a QoI q, the errors for this problem are defined as

$$e_q = \int_0^{y^+_{\text{end}}} \frac{\left(q^{\text{RANS}} - q^{\text{DNS}}\right)^2}{\left(q^{\text{DNS}}\right)^2}$$

where y^+_{end} is the final y^+ value listed in the DNS dataset. Numerical integration is performed with the trapezoidal rule. In all cases, we observe that the TBNN reduces skin friction and brings it into better agreement with the DNS data. The wall heat flux is additionally reduced, and is in better agreement with the DNS data for the M6Tw025 and M14Tw018 cases. We observe an under-prediction in heat flux for the M6Tw076 case. We observe that the ML models result in similar velocity and temperature profiles, and the most significant difference is in the Reynolds stress profiles. Specifically, we observe that the TBNN better predicts the peak in τ_{11} near the wall as well as correctly suppresses τ_{22}.

4.9.1. FFW and BFW cases

To simulate the Mach 5 curved wall cases a 2D domain was simulated with a prescribed inflow boundary such that the inflow boundary layer thickness matched DNS predicted boundary layer



(a) x_1 velocity **(b) Temperature** **(c) Turbulent kinetic energy**

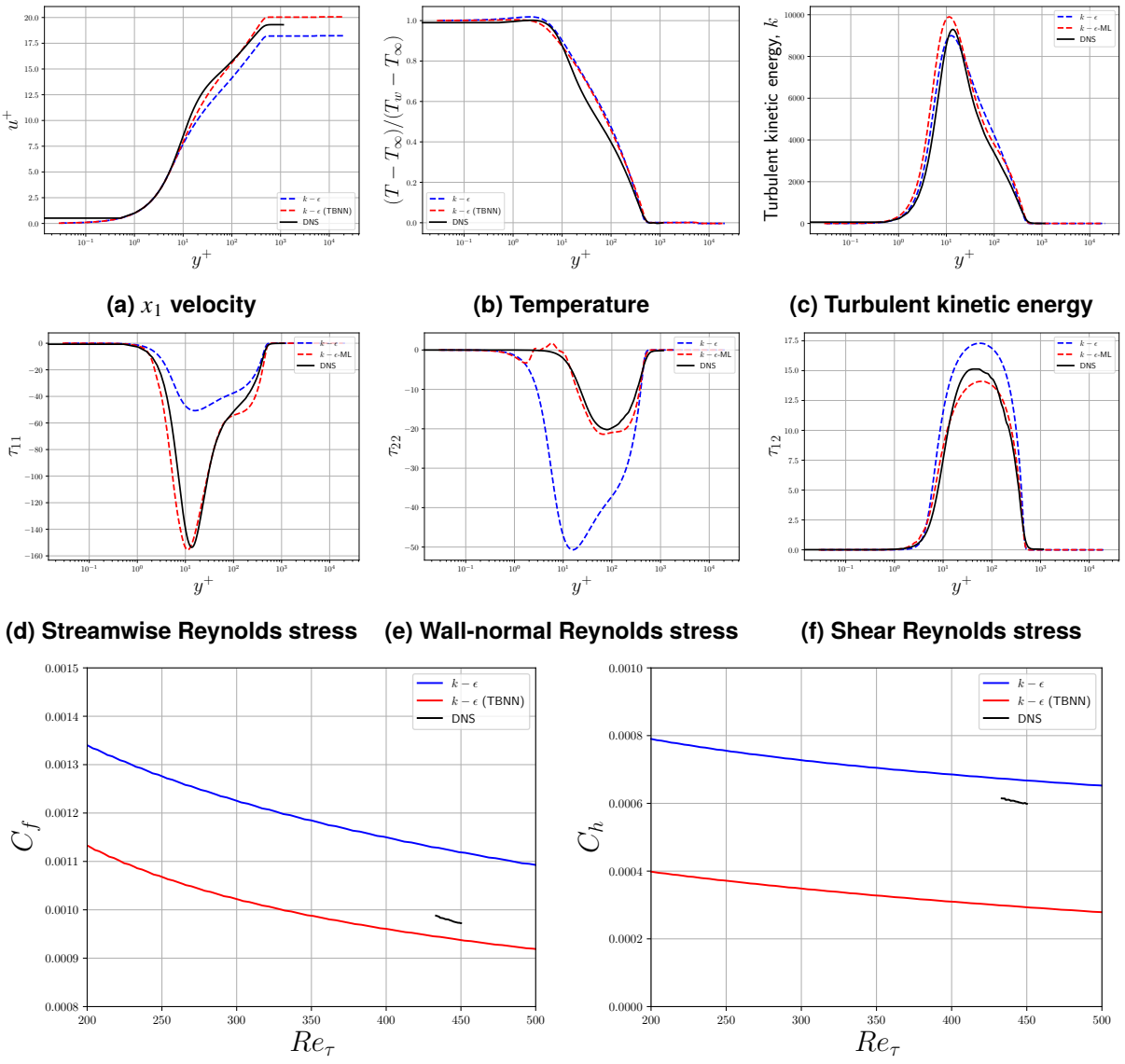
(d) Streamwise Reynolds stress **(e) Wall-normal Reynolds stress** **(f) Shear Reynolds stress**

(g) x_1 velocity **(h) Temperature**

Figure 4-6. Hypersonic boundary layer at $M = 6$, $T_w/T_r = 0.25$. Results of iterative training procedure.

Table 4-8. M6Tw076 hypersonic boundary layer. Relative errors after iterative training procedure.

Model	e_u^+	e_T	$e_{a_{11}}$	$e_{a_{22}}$	$e_{a_{33}}$	$e_{a_{12}}$	e_k	e_{c_f}	e_{c_h}
$k-\epsilon$	0.068	0.150	0.983	0.993	1.006	0.786	0.173	0.148	0.105
$k-\epsilon$-ML	0.035	0.138	0.387	0.895	0.947	0.787	0.156	0.039	0.512



(a) x_1 velocity **(b) Temperature** **(c) Turbulent kinetic energy**

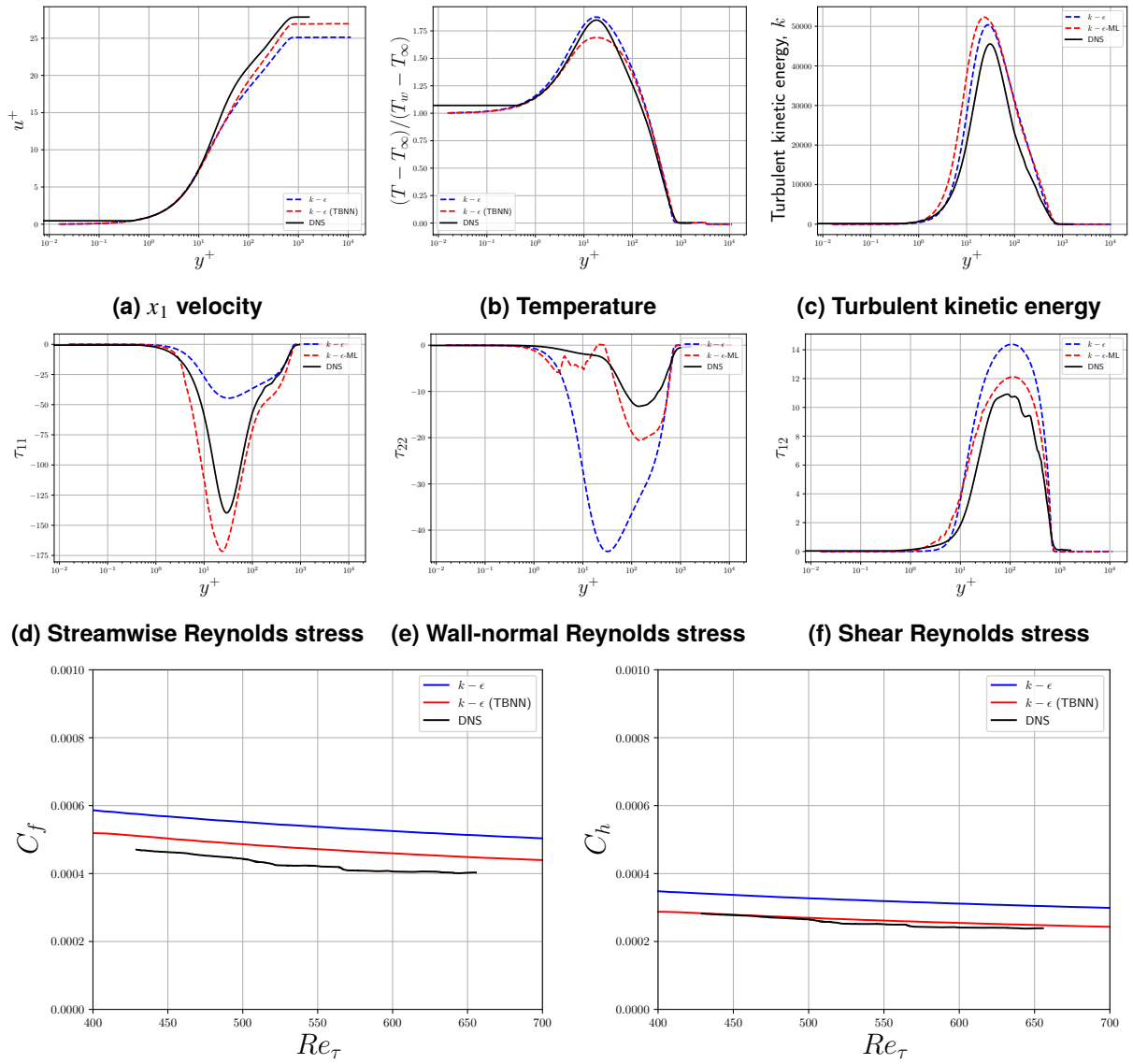
(d) Streamwise Reynolds stress **(e) Wall-normal Reynolds stress** **(f) Shear Reynolds stress**

(g) x_1 velocity **(h) Temperature**

Figure 4-7. Hypersonic boundary layer at $M = 6$, $T_w/T_r = 0.76$. Results of iterative training procedure.

Table 4-9. M14Tw018 hypersonic boundary layer. Relative errors after iterative training procedure.

Model	e_u^+	e_T	$e_{a_{11}}$	$e_{a_{22}}$	$e_{a_{33}}$	$e_{a_{12}}$	e_k	e_{c_f}	e_{c_h}
$k-\epsilon$	0.103	0.095	0.989	0.998	1.010	0.781	0.279	0.264	0.259
$k-\epsilon-$ML	0.043	0.084	0.524	0.920	0.923	0.776	0.320	0.112	0.039



(a) x_1 **velocity** **(b) Temperature** **(c) Turbulent kinetic energy**

(d) Streamwise Reynolds stress **(e) Wall-normal Reynolds stress** **(f) Shear Reynolds stress**

(g) x_1 **velocity** **(h) Temperature**

Figure 4-8. Hypersonic boundary layer at $M = 14$**,** $T_w/T_r = 0.18$**. Results of iterative training procedure.**

Table 4-10. Free-stream and wall conditions for Mach 5 curved wall cases.

u_∞ (m/s)	ρ_∞ (kg/m^3)	T_∞ (K)	T_w (K)
794	0.272	66.2	317

Table 4-11. Relative errors of each model for the curved wall cases.

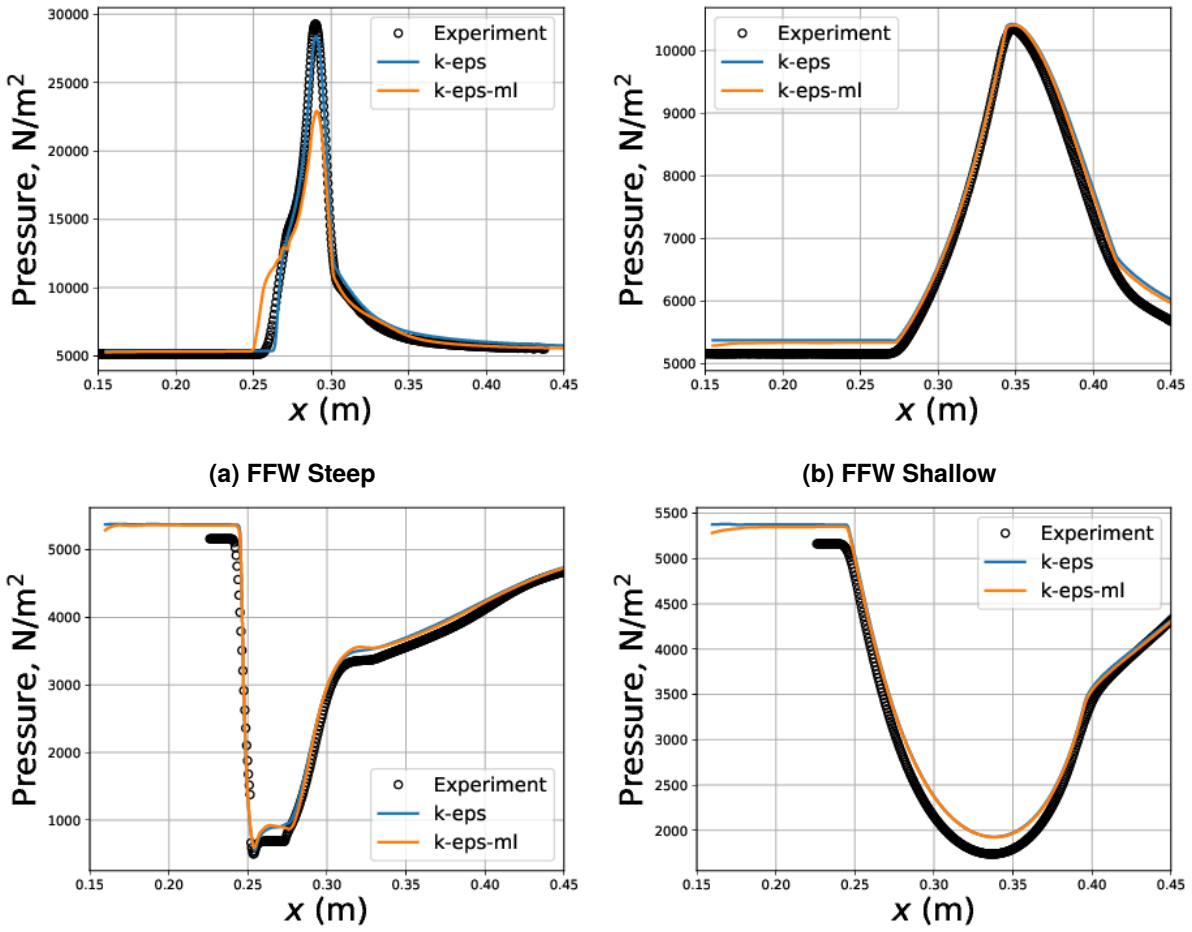
Model	FFW Sharp		FFW Shallow		FFW Sharp		FFW Shallow	
	e_τ	e_p	e_τ	e_p	e_τ	e_p	e_τ	e_p
$k-\epsilon$	0.616	0.045	0.265	0.038	0.183	0.048	0.187	0.047
$k-\epsilon$–ML	0.334	0.165	0.126	0.029	0.127	0.046	0.073	0.043

thickness. This inflow profile was extracted from a precursor flat plate RANS simulation using the baseline $k-\epsilon$ model with the CA correction and C_ϵ set to -1.0, which is also used as the base $k-\epsilon$ model for training purposes. All cases share the same freestream and wall conditions which are provided in Table 4-10. As the separation bubble is highly sensitive to the turbulence model this dataset benefits from an iterative training procedure allowing the model to be retrained as the separation bubble changes, which can significantly alter the downstream profiles. We note that the database includes 5 sample profiles (U2 and L1-L4) for each case. When comparing with DNS, profiles L1 and L3 were allowed to float positions so they would be just before separation or just after reattachment if flow separation occurred. Due to the growth of the separation bubble the steep FFW case profile U2 was also allowed to float so that this profile would be at least 1 δ upstream of separation, i.e., in the undisturbed flow.

The results of the iterative training procedure can be found in Figures 4-9 and 4-10. For all results there is an initial adjustment at the start of the domain due to the inflow profile not including the ML model adjustment, this however does not appear to significantly impact results. First examining pressure, there is relatively negligible change in predicted pressure with the exception of the FFW steep case. Here, predictions are worsened due to a drop in peak pressure. This is however expected with the growth of the separation bubble lowering peak pressure. Wall shear stress shows improvement in the upstream region and in the prediction of peak shear stress even though the separation bubble grows larger as a result of the decreased shear. The relative error for each case is reported in Table 4-11, where shear stress is shown to improve by about a factor of 2 over the domain while pressure shows minor improvement, with the exception to the previously mentioned FFW sharp case.

4.9.2. Global model training

Having computed improved RANS solutions for each of the training cases we now develop global discrepancy models trained on all of the data. As discussed in Section 4.8.1, we balance the dataset by duplicating each channel flow case 8 times and each hypersonic boundary layer case 5 times. We train an ensemble comprising 10 models for the anisotropy discrepancy. For training, all settings are the same as that discussed in Section 4.8.2.



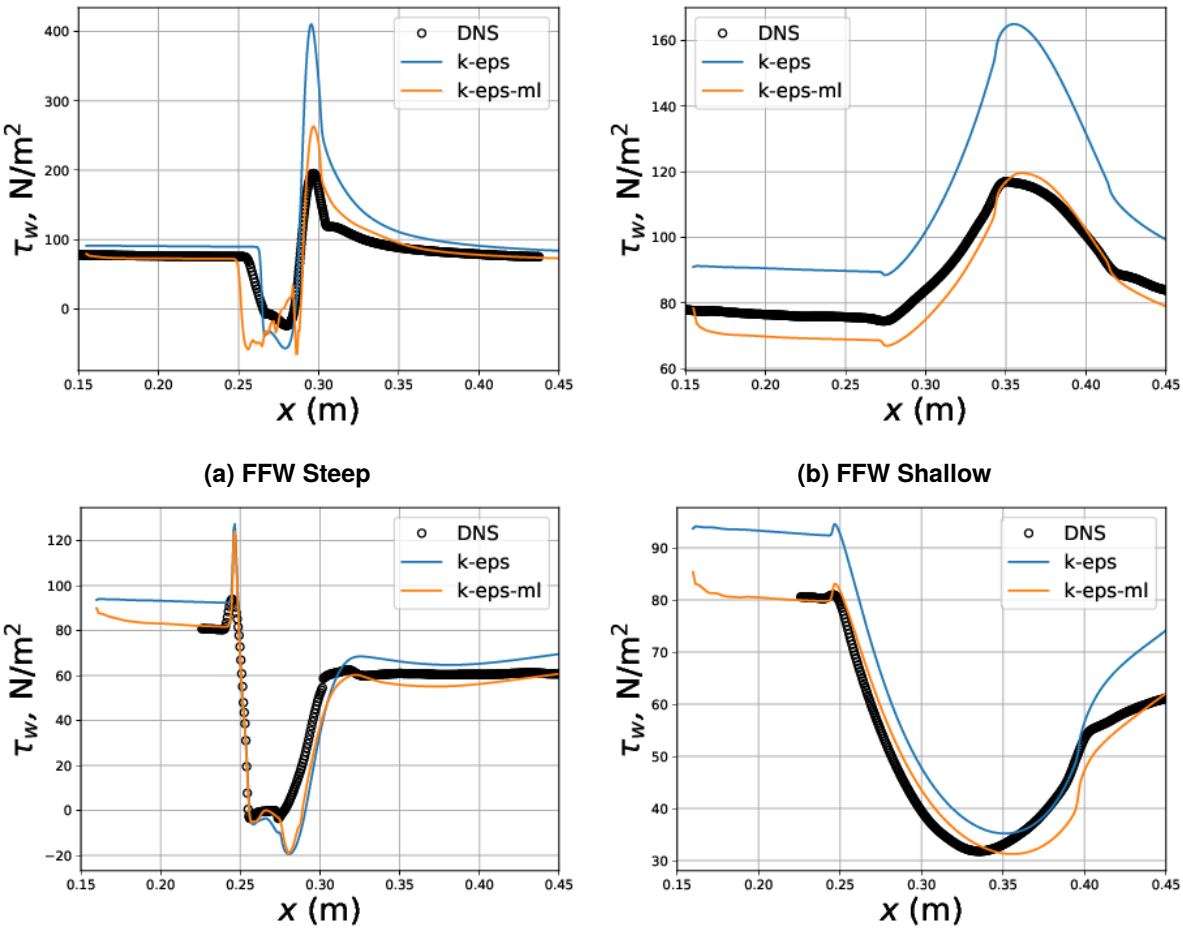
(a) FFW Steep

(b) FFW Shallow

(c) BFW Steep

(d) BFW Shallow

Figure 4-9. Hypersonic curved wall at $M = 4.9$, $T_w/T_r = 0.91$. Results of iterative training procedure on wall pressure predictions.



(a) FFW Steep

(b) FFW Shallow

(c) BFW Steep

(d) BFW Shallow

Figure 4-10. Hypersonic curved wall at $M = 4.9$, $T_w/T_r = 0.91$. Results of iterative training procedure on wall shear stress predictions.

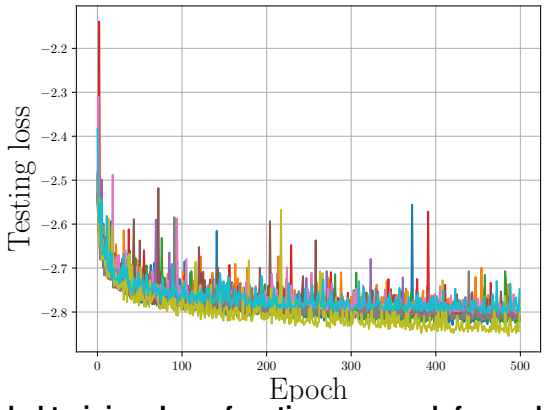


Figure 4-11. Global training. Loss function vs. epoch for each ensemble member.

Figure 4-11 shows the convergence of the loss function on the validation set as a number of epochs for each member. We note that the loss is negative as we minimize the log-likelihood. We observe that all members of the ensemble behave similarly. Application of the calibration procedure described in Section 4.4 results in a scaling constant of $\hat{C} \approx \sqrt{1500}$.

4.10. Test cases

We now consider results of our global ML model applied to various test cases spanning different physics. These physics include low speed boundary layers, jets, separated flows, high speed boundary layers, SBLIs, and compression ramps/flares.

4.10.1. Zero pressure gradient flat plate boundary layer

We first consider a zero-pressure gradient boundary layer. The case setup follows the "2DZP: 2D Zero Pressure Gradient Flat Plate Validation Case" as described on the NASA turbulence modeling resource, and we compare to the DNS data by Sillero at el. [79]. For all simulations, we start from a uniform free-stream value and converge our solver with a uniform time step at a CFL of 1000. Figure 4-12 shows results for the velocity, Reynolds shear stress, and friction coefficient, while Table 4-12 tabulates errors between the RANS models and the DNS data by Sillero at el. [79]. For this problem we define an error metric for skin friction as

$$e_{C_f} = \sqrt{\frac{\int_{4000}^{6660} \left(C_f^{\mathrm{DNS}} - C_f^{\mathrm{RANS}}\right)^2 d\mathrm{Re}_\theta}{\int_{4000}^{6660} \left(C_f^{\mathrm{DNS}}\right)^2 d\mathrm{Re}_\theta}}.$$

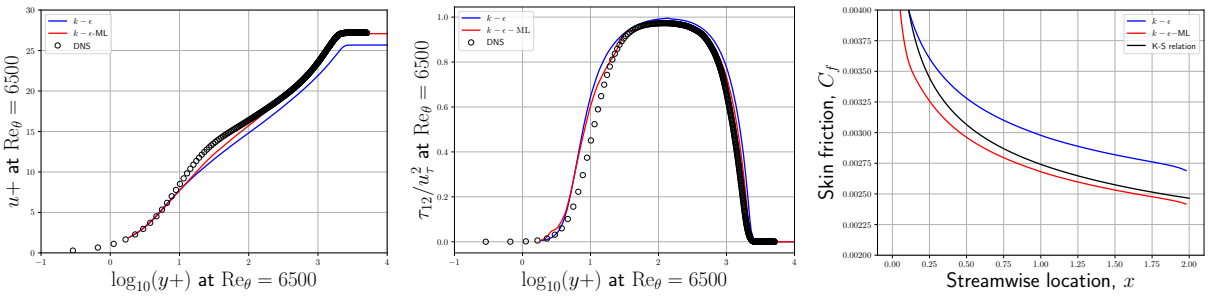


Figure 4-12. Flat plate. Predictions for velocity (left), Reynolds shear stress (center), and friction coefficient (right). K-S stands for the Karman-Schoenherr relation.

Table 4-12. Relative errors for zero-pressure gradient flat plate boundary layer.

Model	e_u	$e_{\tau_{12}}$	e_{c_f}
$k-\epsilon$	0.0041	0.2136	0.1195
$k-\epsilon-$ML	0.0001	0.1283	0.0135

We note that $\mathrm{Re}_\theta = 6600$ is the last value available in the DNS dataset and occurs in the RANS solutions at around $x = 0.6$. We also note that RANS solutions at Reynolds numbers lower than $\mathrm{Re}_\theta = 4000$ are very near the leading edge and are not fully turbulent. We additionally consider QoI errors for (wall-normal) streamwise velocity and Reynolds stress profiles at $\mathrm{Re}_\theta = 6500$ which we define as

$$e_q = \sqrt{\frac{\int_0^{y_2^+} \left(q^{\mathrm{DNS}} - q^{\mathrm{RANS}}\right)^2 dy^+}{\int_0^{y_2^+} \left(q^{\mathrm{DNS}}\right)^2 dy^+}}$$

for $q = u^+, \tau_{12}$ and where $y_2^+ = 5100$ is the last point in the DNS dataset. The $k-\epsilon-$ML model is seen to improve upon the baseline $k-\epsilon$ model for all quantities. The Lipschitz continuous network formulation results in a relatively smooth Reynolds stress prediction, with the exception of a slight kink near the wall and edge of the buffer layer. Before proceeding, we investigate the impact of three aspects of our formulation: (1) the impact of ensembles, (2) the impact of Lipschitz continuity, and (3) the impact of the number of tensor bases.

4.10.1.1. Impact of ensembles

We briefly highlight the impact of ensembles. Figure 4-13 shows residual convergence (left) and QoI convergence (right) for ML-enhanced RANS simulations starting from a uniform freestream utilizing ML models trained with two different formulations: a single Lipschitz-continuous tensor-basis neural network and an ensemble formulation. We note that these networks employ a slightly different training dataset than that described above, and the purpose here is to simply highlight

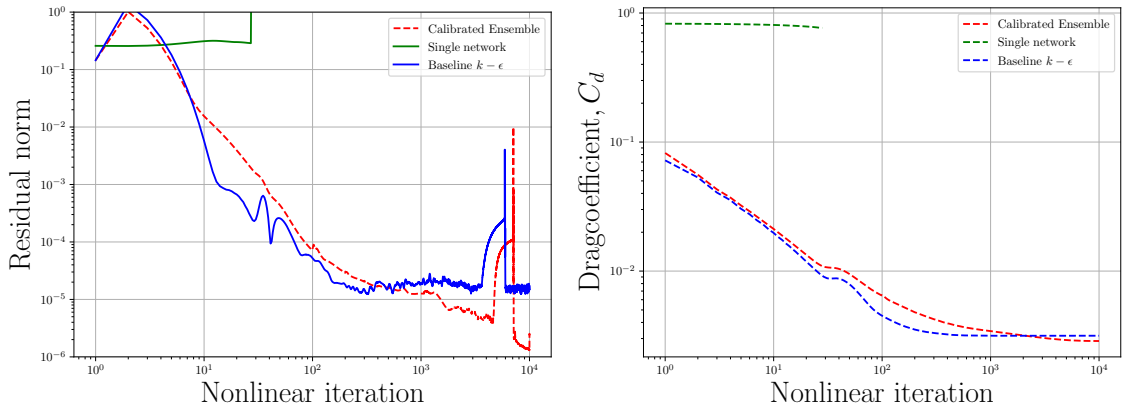


Figure 4-13. Flat plate. Impact of ensemble formulation. Residual convergence (left) and convergence of the drag coefficient (right).

the impact of the ensemble formulation. The standard $k-\epsilon$ model is additionally included. The training configuration between the single network and the ensemble of networks is equivalent. Figure 4-13 shows that the single network formulation leads to an un-converged solution. After about 10 iterations, the solver returns NaNs. In contrast, the ensemble formulation results in iterative convergence on par with the baseline $k-\epsilon$ model. While these results vary from case to case and trained network to trained network we have, in general, observed that the ensemble formulation is more robust than a single network formulation.

4.10.1.2. Impact of Lipschitz-continuous regularization

We briefly assess the impact of the Lipschitz-continuous networks. We train three ensembles of TBNNs with layer-wise Lipschitz constants of $k_{\text{lip}} = 1.75, 3, 4$. All other hyper-parameters are the same between the networks. As expected, we observe that lowering the Lipschitz constant results in a higher testing loss. We again note that these networks employ a slightly different training dataset and configuration than what is described above, and the goal of this section is simply to highlight the impact of Lipschitz continuity. Figure 4-14 shows residual convergence, predicted Reynolds shear stress, and velocity profiles predicted by the three formulations on the flat plate case. Although they (surprisingly) display reasonable residual convergence, the networks with high Lipschitz constants produce noisy Reynolds stress fields. These Reynolds stress fields in turn impact mean profiles and result in velocity profiles that violate the log-law. In general, we have observed that enforcing a lower Lipschitz constant results in more robust networks with smoother outputs. Too low of a Lipschitz constant does reduce network accuracy, however.

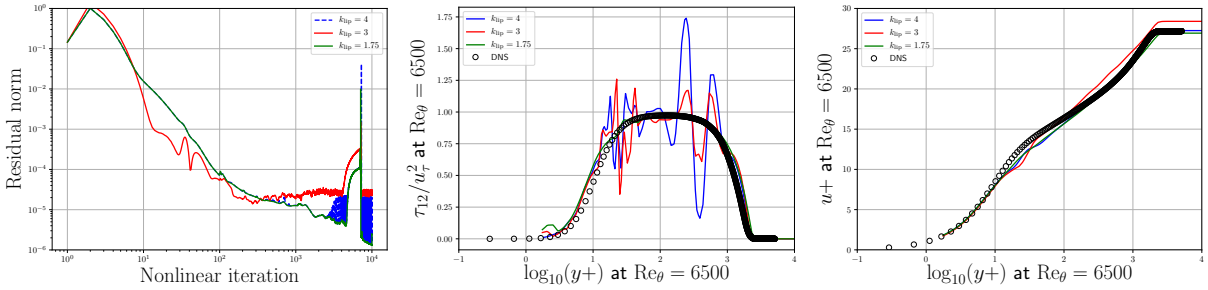


Figure 4-14. Flat plate. $k-\epsilon-$ML results for different layer-wise Lipschitz constants.

4.10.1.3. Impact of number of tensor bases

We consider solutions with an ensemble of tensor-basis neural networks with two configurations: one where we use all four bases of the TBNN ensemble and one where we only use the first basis. We emphasize that the two configurations correspond to the same networks; the one basis ensemble is just a truncated version of the four-basis ensemble. Figure 4-15 shows the velocity profiles, Reynolds stress profiles, and iterative convergence of the different models. As expected from the analysis provided in Section 4.3.1, the one basis and four-bases solutions lie on top of each other. This result demonstrates that the higher-order tensor bases have a minimal impact the dynamics of the flow for a subsonic, zero-pressure gradient boundary layer.

4.10.2. Wall-mounted hump

We next consider the "2D NASA Wall-Mounted Hump Separated Flow" case as summarized on the NASA TMR; Figure 4-16 depicts the problem schematic as described on the TMR. This case comprises flow separating over a smooth body. Experiments of this configuration are summarized in Refs. [25, 24, 55] and data are available on the TMR. For this problem we define error metrics for skin friction and pressure coefficient as

$$e_q = \sqrt{\frac{\int_{x_1}^{x_2} \left(q^{\text{EXP}} - q^{\text{RANS}}\right)^2 dx}{\int_{x_1}^{x_2} \left(q^{\text{EXP}}\right)^2 dx}}$$

for $q = C_f, C_p$. We additionally consider QoIs for field metrics which are defined as Eq. (4.9.1) for $q = \tilde{u}_1, \tilde{u}_2, \tau_{12}$. Figure 4-17 depicts the pressure coefficient and skin friction as a function of the streamwise location for the $k-\epsilon$ and $k-\epsilon-$ML models. The inclusion of the ML model results in small but improved predictions for both skin friction and pressure coefficient. Pressure coefficient is captured slightly better throughout the separation region between roughly $0.6 \leq x \leq 1.2$, and predictions for skin friction are noticeably improved downstream of separation. We do observe that the $k-\epsilon-$ML slightly under predicts the peak skin friction, and we observe small "wiggles" in the

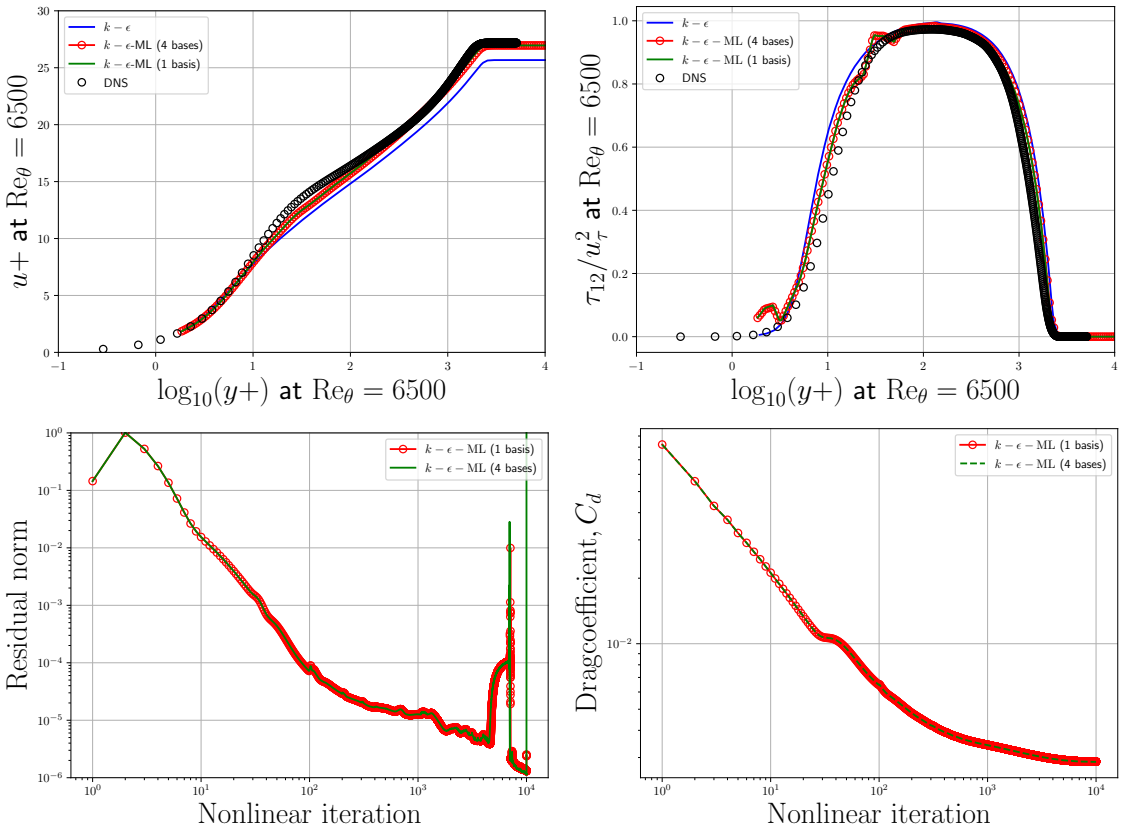


Figure 4-15. Flat plate. Impact of number of TBNN bases. Velocity profiles (top left), Reynolds shear stress (top right), residual convergence (bottom left) and convergence of the drag coefficient (bottom right).

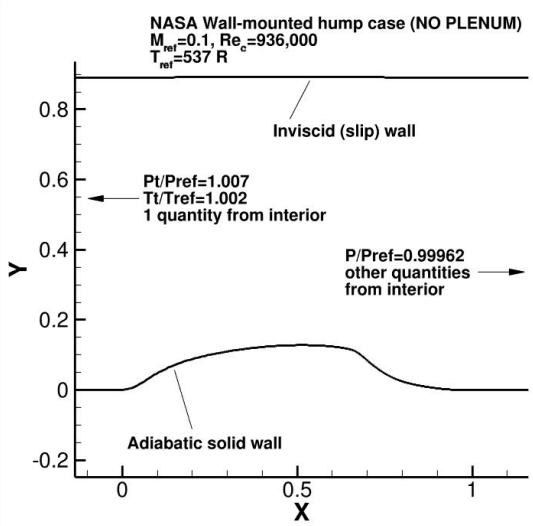


Figure 4-16. NASA hump. Problem configuration (figure taken from NASA TMR).

Table 4-13. Relative errors for wall-mounted hump problem.

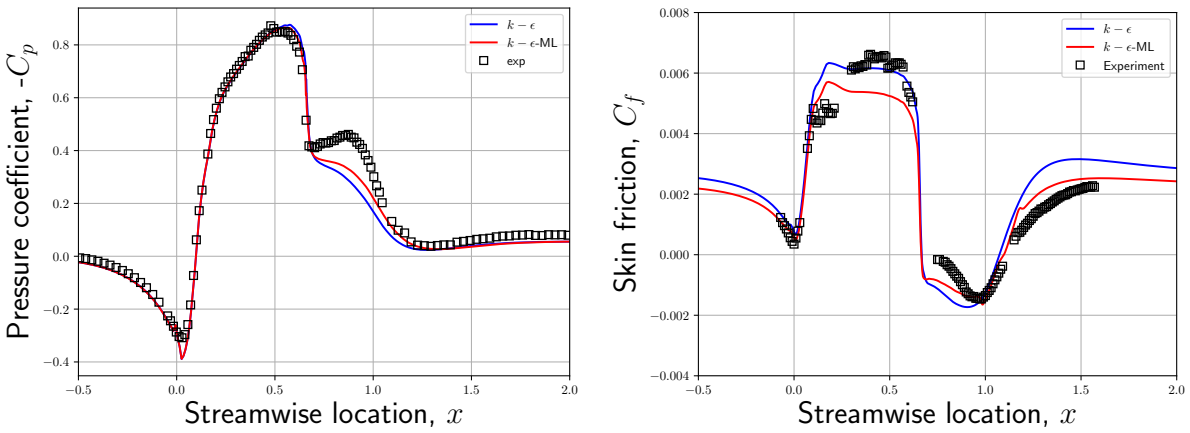
Model	e_u	e_v	$e_{\tau_{12}}$	e_{c_f}	e_{c_p}
$k-\epsilon$	0.0916	0.3767	0.4922	0.2368	0.1502
$k-\epsilon-$ML	0.0683	0.2909	0.4469	0.1832	0.1099

solution near reattachment. Figure 4-18 shows contours of the x_1 and x_2 velocities predicted by the $k-\epsilon$ and $k-\epsilon-$ML models. The addition of the ML model results in small differences. We observe a slightly stronger recirculation region around $x = 0.8$, as indicated by a larger negative x_1 velocity and a larger x_2 velocity.

Table 4-13 tabulates the relative errors between the RANS solution and experiment for skin friction, pressure coefficient, x_1 and x_2 velocities, and Reynolds stress. Comparing $k-\epsilon-$ML to the standard $k-\epsilon$ model, we observe 15-30% lower errors for all metrics. Most pronounced are the 30% improvements to the pressure coefficient and the velocity fields.

4.10.3. Axisymmetric jet

The next flow we consider is an axisymmetric subsonic jet flow as detailed on the NASA TMR. The flow is characterized by a M = 0.5 jet exiting into an almost quiescent M = 0.01 background. Experimental data for this configuration are detailed in [8] and are readily available on the NASA TMR. The problem configuration is shown in Figure 4-19. For a QoI q, error metrics for this problem are defined as in Eq. (4.9.1).



(a) Pressure coefficient **(b) Friction coefficient**

Figure 4-17. NASA hump. Pressure coefficient (left) and skin friction (right) as a function of streamwise location.

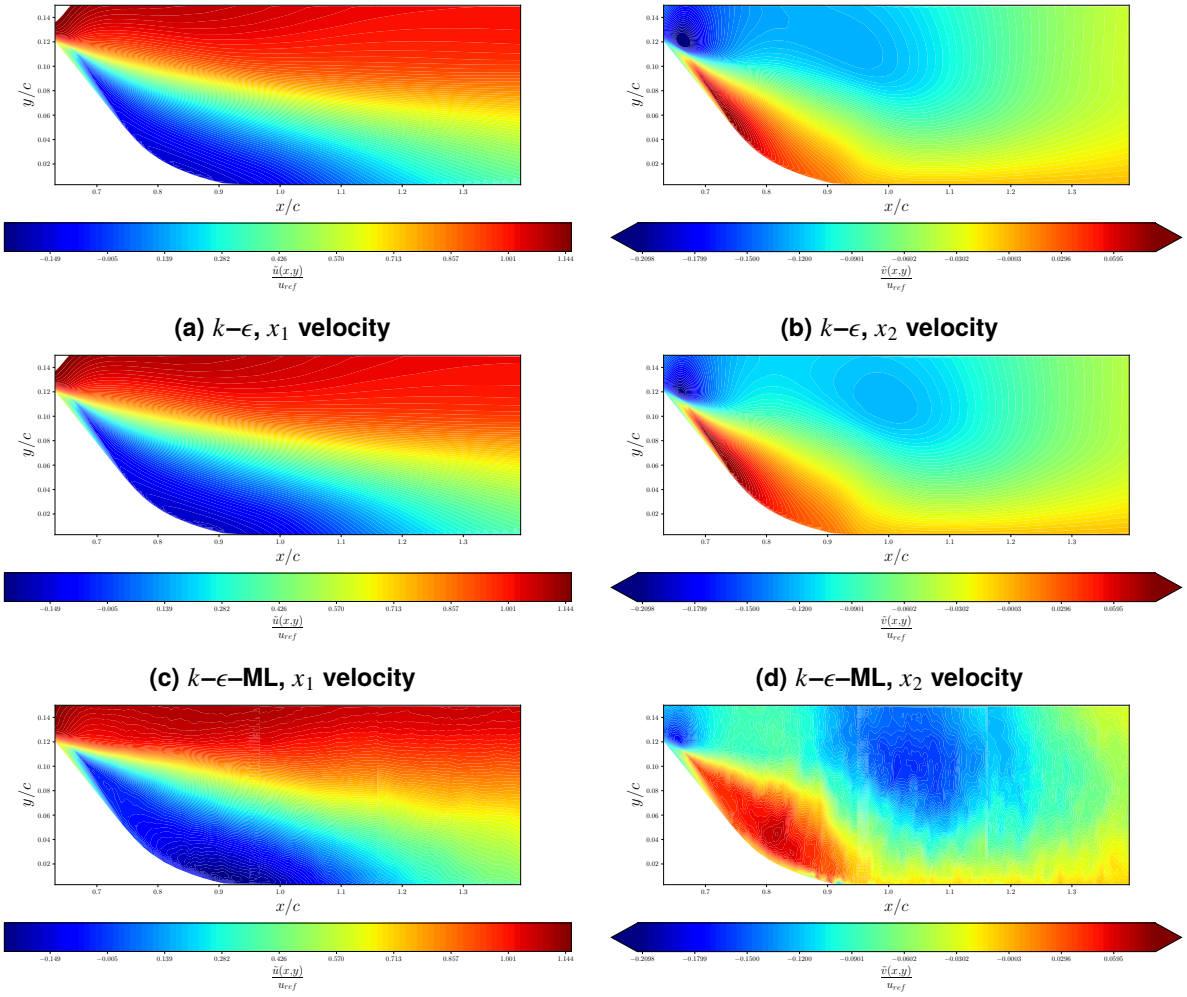
Table 4-14. Relative errors for the axisymmetric jet.

Model	e_u	e_v	$e_{\tau_{12}}$
$k-\epsilon$	0.0878	0.4871	0.3264
$k-\epsilon-$ML	0.0765	0.4183	0.2877

Figure 4-20 shows the centerline velocities as predicted by the $k-\epsilon$ and $k-\epsilon-$ML models as a function of downstream distance at $y = z = 0$. We observe that the $k-\epsilon$ model under-predicts the centerline velocity magnitude by a significant margin, while the $k-\epsilon-$ML model yields improved solutions. This result is enforced in Figure 4-21, which shows contours of the mean flow solutions for the $k-\epsilon$ and $k-\epsilon-$ML models as compared to the experimental PIV measurements. The $k-\epsilon$ model is seen to under-predict the distance which the jet exhausts into ambient background, while the $k-\epsilon-$ML model results in a prediction that is more qualitatively accurate. Table 4-14 tabulates the quantity-of-interest errors for axisymmetric jet problem. The $k-\epsilon-$ML model results in improved predictions across all three quantities of interest.

4.10.4. Mach 11 cold-wall boundary layer

We consider a Mach 11, cold-wall boundary layer under conditions that match a Mach 11.1 hypersonic turbulent flat plate boundary layer tested at Calspan–University of Buffalo Research Center (CUBRC) [22, 23]. DNS results for this case are reported in Refs. [105, 35, 34]. Baseline RANS models result in a systematic over-prediction of both skin friction and wall heating at high Reynolds numbers [35, 1, 23, 15]. As described in Ref. [61], to account for the leading-edge



(a) k–ϵ, x_1 velocity

(b) k–ϵ, x_2 velocity

(c) k–ϵ–ML, x_1 velocity

(d) k–ϵ–ML, x_2 velocity

(e) PIV, x_1 velocity

(f) PIV, x_2 velocity

Figure 4-18. NASA hump. RANS predictions and PIV measurements for x_1 (left) and x_2 (right) components of the velocity.

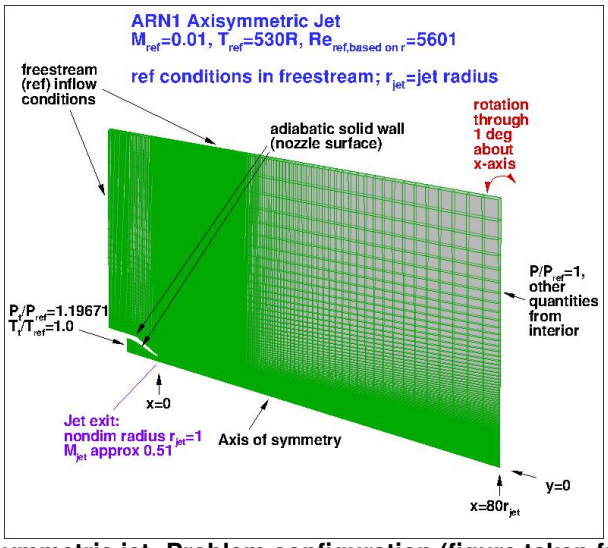


Figure 4-19. Axisymmetric jet. Problem configuration (figure taken from NASA TMR).

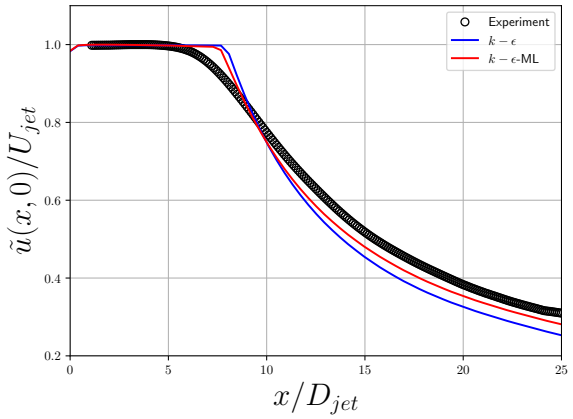
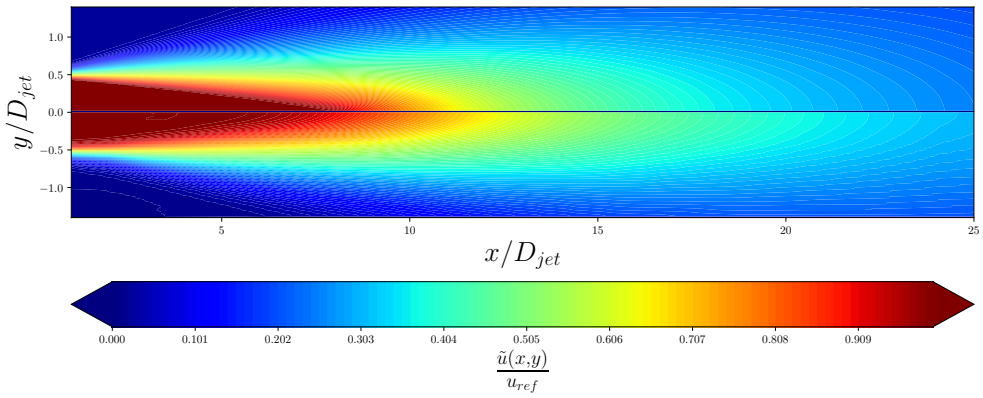
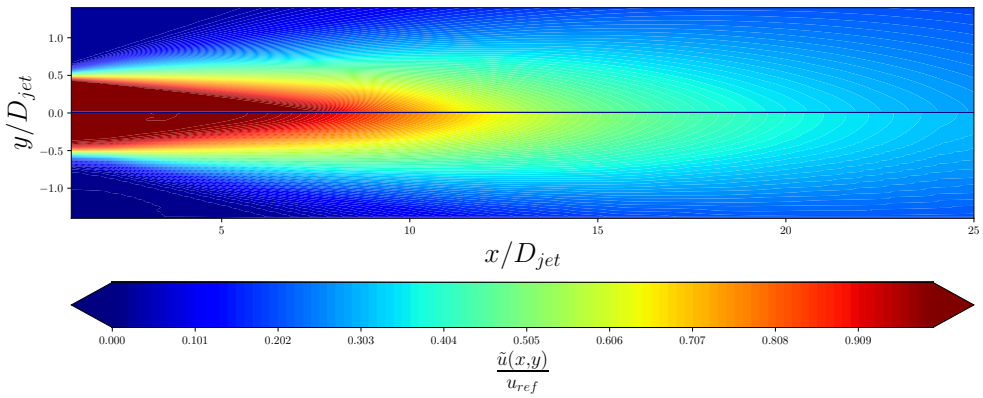


Figure 4-20. Axisymmetric jet. Centerline velocities.



(a) k–ϵ, x_1 velocity



(b) k–ϵ–ML, x_1 velocity

Figure 4-21. Results for axisymmetric jet. In each figure the RANS solution is depicted for $y > 0$ and the PIV data are depicted for $y < 0$.

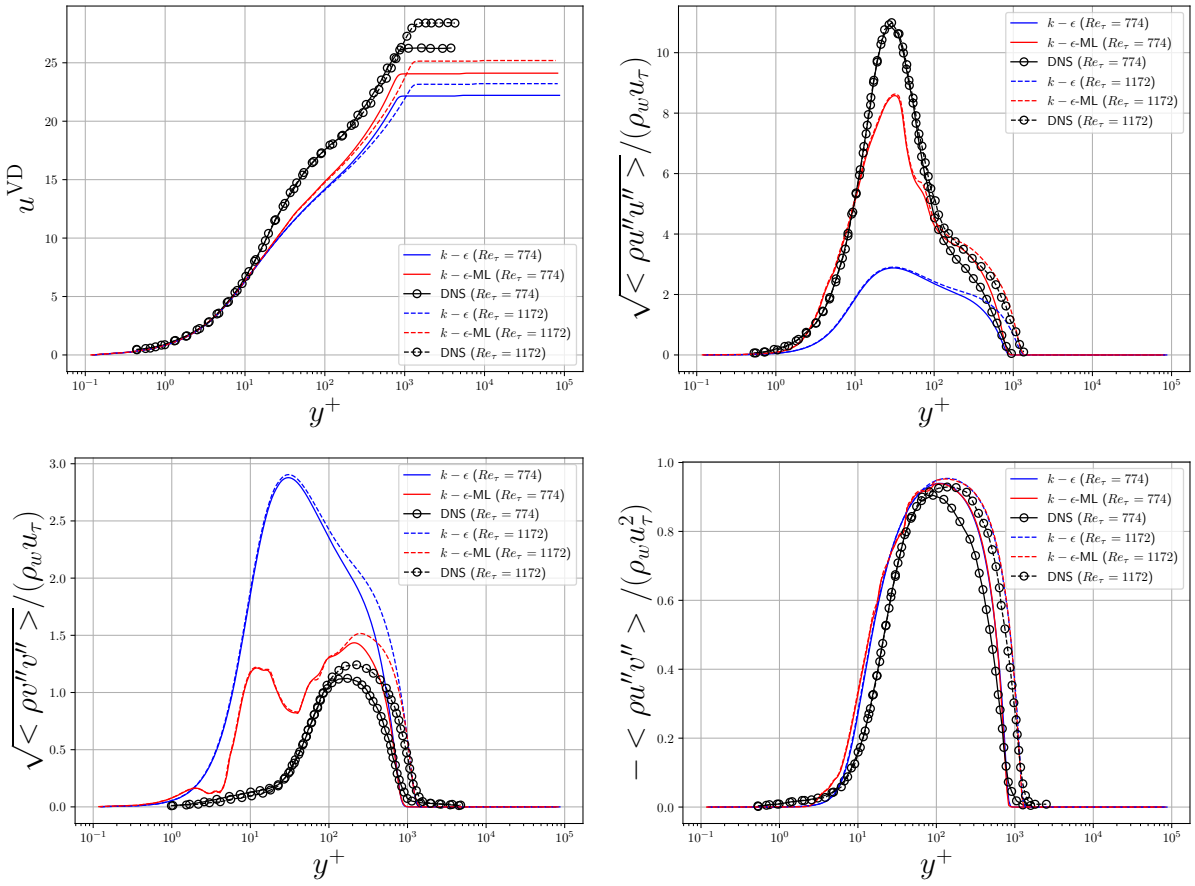


Figure 4-22. Mach 11 boundary layer. Comparison of RANS solutions to the DNS reported in Ref. [34].

shock we employ modified free-stream boundary conditions such that the post-shock conditions match those of the corresponding DNS simulations and CUBRC experiments; these modified boundary conditions are given in Ref. [61]. Experimental free-stream conditions are reported in [49]. Figure 4-22 shows predictions for the Van-Driest transformed velocity, τ_{11}, τ_{22}, and τ_{12} Reynolds stresses, and compares them to the DNS results reported by Huang et al. [34]. Figure 4-23 reports skin friction and wall heat flux results, while QoI errors are given in Table 4-15. The $k-\epsilon-$ML model results in overall improved predictions. In particular, the $k-\epsilon-$ML model results in a decrease in both skin friction and heat flux which are in better agreement with high-fidelity data.

Examining the Reynolds stresses, we observe that the ML model amplifies the τ_{11} component of the Reynolds stress while suppressing the τ_{22} component. The normal components of the Reynolds stress tensor do not make a significant impact on the dynamics of a low-speed boundary layer, as evidenced in Section 4.10.1.3. Exploring the performance of the TBNN, we find that the situation is different for hypersonic boundary layers. Figure 4-24 shows solutions for wall heat flux and wall

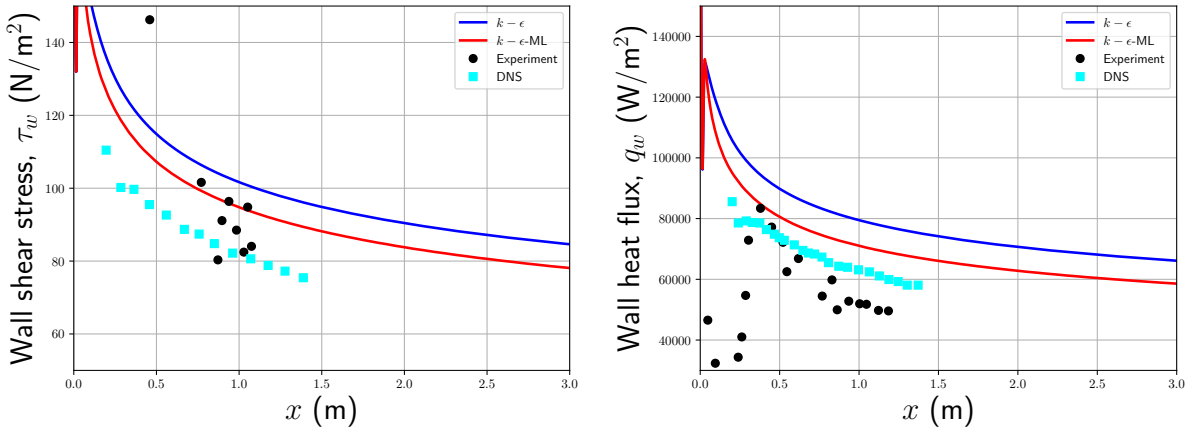


Figure 4-23. Mach 11 boundary layer. Wall shear stress (left) and wall heat flux (right).

Table 4-15. Relative errors for the M11Tw020 case.

Model	e_u	e_T	$e_{\tau_{12}}$	$e_{\tau_{11}}$	$e_{\tau_{22}}$	e_{τ_w}	e_{q_w}
$k-\epsilon$	0.1573	0.2307	0.1601	0.5840	1.1156	0.2386	0.2514
$k-\epsilon-$ML	0.0879	0.3206	0.1636	0.2149	0.3878	0.1555	0.1218

shear stress using a one-basis and four-basis TBNN. Like the subsonic flat plate, note that the two configurations correspond to the same networks; the one basis TBNN is just a truncated version of the four-basis TBNN. The predictions of the two configurations deviate; this is unlike the subsonic case reported in Figure 4-15. Investigation of this discrepancy shows that it can be attributed to the contribution of the wall-normal Reynolds stress in the wall-normal momentum equation. As shown by Barone et al. [4], the wall-normal momentum equation in a flat plate boundary layer reduces to

$$\frac{dp}{dx_2} = \frac{\partial \tau_{22}}{\partial x_2},$$

where we have assumed x_2 to be the wall-normal coodinate. In low-speed boundary layers, the wall-normal component of the Reynolds stress is small compared to the mean pressure, and the above can safely be reduced to $\frac{dp}{dx_2} = 0$. The situation appears to be different for high Mach number boundary layers, where the Reynolds stress becomes non-negligible compared to the pressure. As a result, changes to the wall-normal Reynolds stress in turn impact the pressure (and, as a result, the dynamics of the flow). Although not shown, the four-basis TBNN results in improved predictions for wall-normal pressure and density profiles. In the present context, the four-basis TBNN (which suppresses the wall-normal component of the Reynolds stress) has the net effect of raising wall shear stress and wall heating.

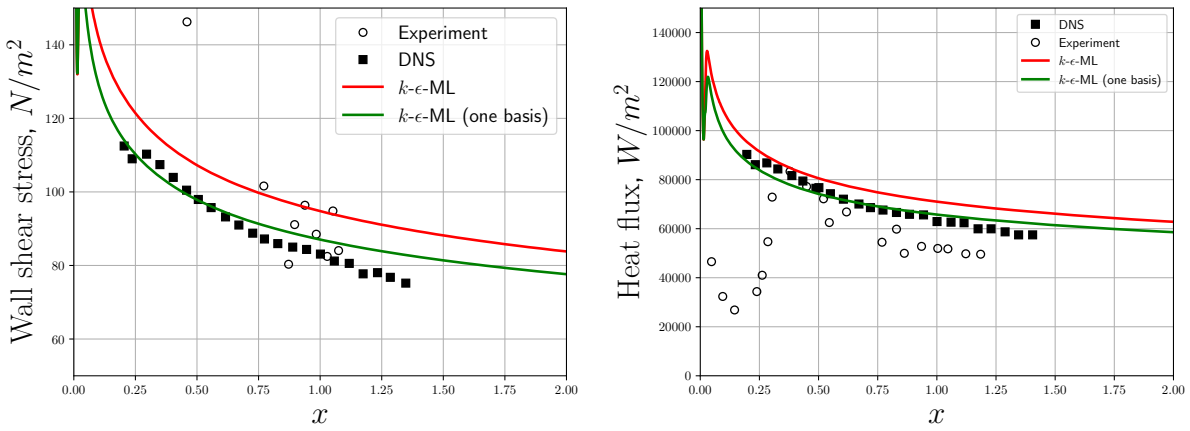
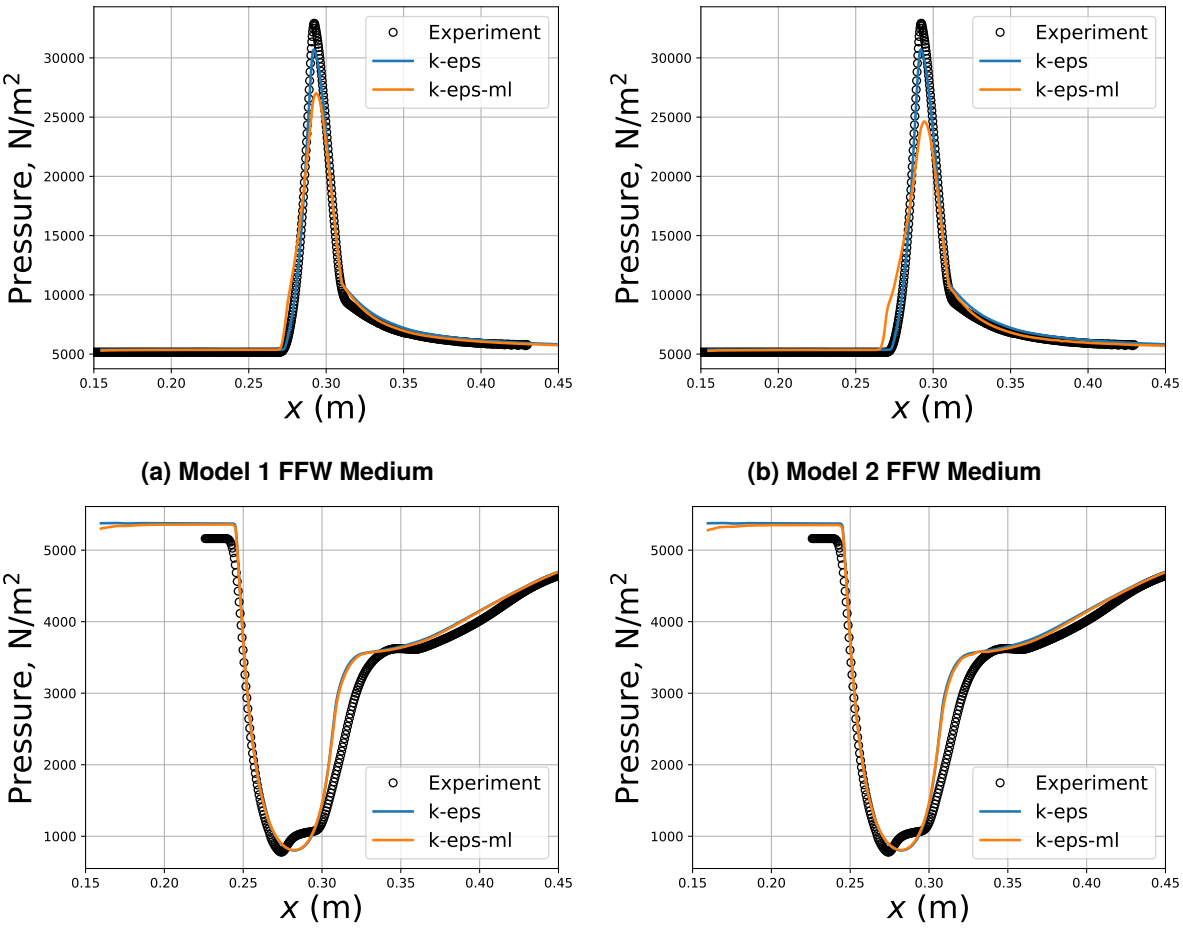


Figure 4-24. Mach 11 boundary layer. Wall shear stress (left) and wall heat flux (right) for four-bases and one-basis TBNN solutions.

4.10.5. Curved wall

We now consider two cases from the curved wall database [57] that were held out from training. Each case comprises a curved wall that has a medium steepness which results in incipient or mean flow separation. In addition to the model described in Section 4.9.2, which we refer to as "Model 1", another model is tested here to explore the sensitivity of the TBNN to training hyper-parameters. This second model employs a layer-wise Lipschitz constant of 2.5, was trained using a standard ℓ^1 loss opposed to the log-likelihood, and employed a training dataset where each curved wall case was replicated two times. We will refer to this model as "Model 2". Figures 4-25 and 4-26 show results of these two models. Both models predict relatively the same thing, however, Model 1 gives smoother results. This indicates that even with changes in the training configuration the model still follows essentially the same trends. As the results are roughly the same further discussion will be on Model 1.

Pressure predictions are essentially unchanged except on the curved wall section of the FFW medium case where predictions are worsened. This is due to the occurrence of flow separation which drops peak pressure. It should be noted that the DNS indicates incipient separation albeit not mean flow separation, so this result is not unreasonable with the corrected shear stress. Shear stress shows modest improvements to the predictions of upstream and peak shear stress even though there is a mismatch in separation prediction. The expected enhancement of separation is apparent with both cases which results in separation for the FFW medium case, while the BFW medium case does not quite separate. Overall the models improve predictions as can be seen in the reduction in relative error noted in Table 4-16.



(a) Model 1 FFW Medium

(b) Model 2 FFW Medium

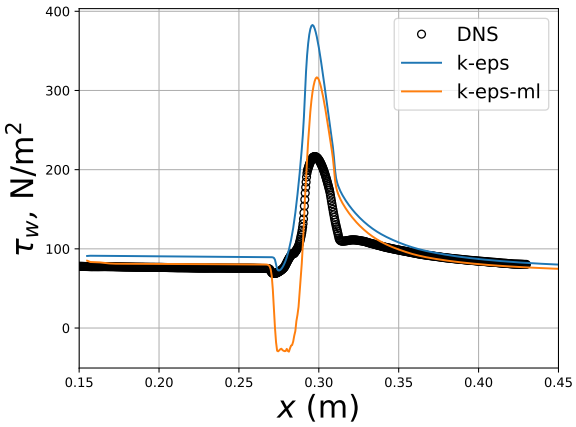
(c) Model 1 BFW Medium

(d) Model 2 BFW Medium

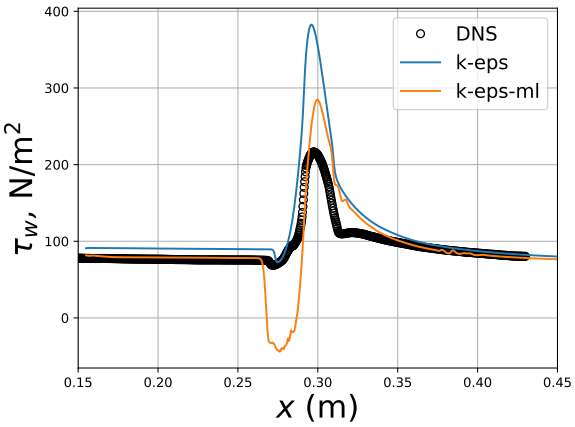
Figure 4-25. Hypersonic curved wall at $M = 4.9$, $T_w/T_r = 0.91$. Results of combined models for wall pressure predictions.

Table 4-16. Relative errors of each model for the curved wall test cases.

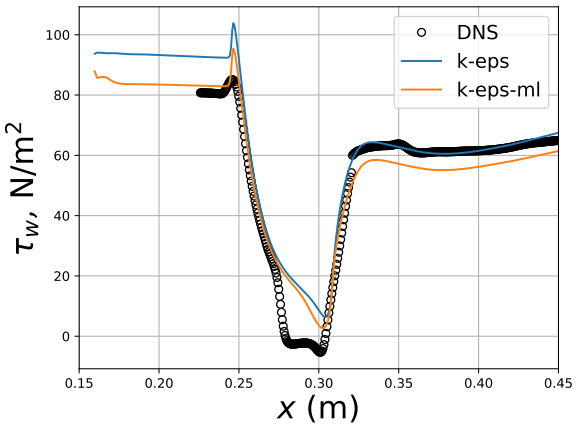
Model	FFW Medium		BFW Medium	
	e_τ	e_p	e_τ	e_p
$k-\epsilon$	0.467	0.048	0.133	0.068
$k-\epsilon-ML1$	0.386	0.127	0.110	0.065
$k-\epsilon-ML2$	0.404	0.189	0.133	0.064



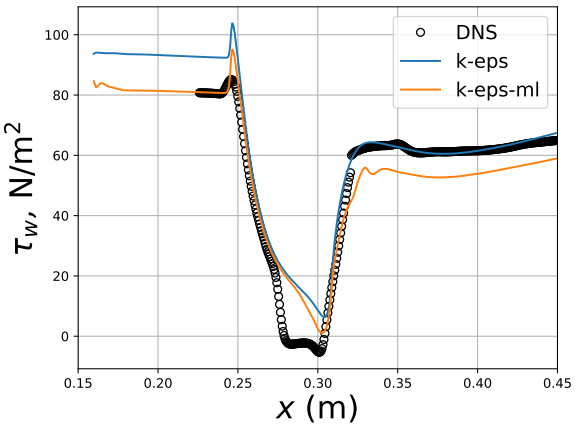
(a) Model 1 FFW Medium



(b) Model 2 FFW Medium



(c) Model 1 BFW Medium



(d) Model 2 BFW Medium

Figure 4-26. Hypersonic curved wall at $M = 4.9$, $T_w/T_r = 0.91$. Results of combined model on wall shear stress predictions.

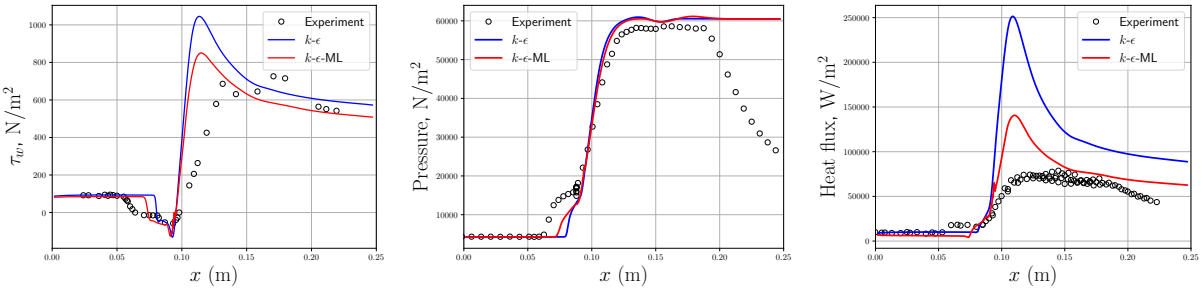


Figure 4-27. Mach 5 shock boundary layer interaction. RANS predictions for QoIs.

Table 4-17. Relative errors for Schülein shock boundary layer interaction.

Model	e_{τ_w}	e_p	e_{q_w}
$k-\epsilon$	0.7608	0.2689	1.2841
$k-\epsilon-$ML	0.5545	0.2635	0.4622

4.10.6. Shock boundary layer interactions at M = 5.0

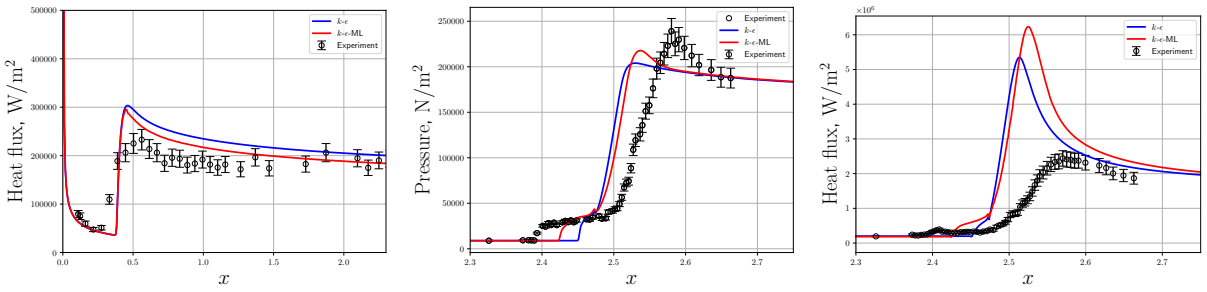
We consider a shock boundary layer interaction with an impinging shock at 14°, a free-stream Mach number of M = 5.0, and with a cooled wall at a wall-to-recovery ratio of $\frac{T_w}{T_r} = 0.8$. The cases match the experimental conditions of Schülein [78]. DNS of these configurations, but at a lower Reynolds number, are reported in Volpiani et al [91].

Figure 4-27 reports results for wall shear stress, wall pressure, and wall heat flux. Table 4-17 tabulates errors for wall quantities for both cases.[2] As previously reported, the baseline $k-\epsilon$ model under-predicts the size of the separation region and dramatically over-predicts heat flux. The $k-\epsilon-$ML model results in substantial improvements. In particular, the size of the separation region is slightly increased and the over-prediction in heat flux is reduced. The wall shear stress at and downstream of reattachment is decreased slightly. Despite the improvements, over-prediction of the heat flux at the point of reattachment is still significant.

4.10.7. CUBRC Hollow Cylinder Flare

The final test case we consider is run 13 of the CUBRC hollow cylinder flare (HCF). The HCF run 13 corresponds to a Mach 6 axisymmetric hollow cylinder flare with a 36° ramp. A schematic of the experimental configuration is shown in the following chapter in Figure 5-12. Figure 4-28 shows the wall pressure and wall heat flux predicted by the baseline and ML-enhanced models. Table 4-18

[2]The reported errors correspond to the relative ℓ^2 norm of the point-wise error between the RANS prediction and experimental data.



(a) Wall heat flux on forebody. **(b) Pressure on flare.** **(c) Wall heat flux on flare.**

Figure 4-28. CUBRC hollow cylinder flare. Run 13.

Table 4-18. Relative errors for HCF run 13.

Model	e_p	e_{q_w}
$k-\epsilon$	0.4644	1.3806
$k-\epsilon-$ML	0.4057	1.5780

tabulates the relative errors between the predictions and the experimental data. We observe that the ML model has a positive impact on the predictions. Similar to the impinging SBLI, we find that the size of the separation region is improved, which in turn results in an improved prediction of peak pressure. We additionally find a better prediction for wall heat flux on the forebody. However, we find that the ML-enhanced model does not improve wall heat flux near the point of reattachment. Indeed, the ML model predicts a higher peak wall heat flux, although we do note that this increased heat flux can largely be attributed to the larger separation region. However, the TBNNs clearly do not address the over-prediction in peak wall heat flux.

4.11. Summary and outlook

This chapter presented an approach for developing improved models for the Reynolds stress tensor via TBNNs. The approach was used to improve upon a baseline $k-\epsilon$ model. To improve robustness of the ML-enhanced model, we developed Lipschitz-continuous versions of the TBNN leveraging calibrated deep ensembles to quantify model uncertainties. We additionally performed analysis of the TBNN architecture for parallel flows, where we demonstrated that most higher-order tensor bases only modulate the normal component of the Reynolds stress. We discussed the implications of this finding for boundary layer flows.

We presented numerical results across a range of low-speed and high-speed exemplars. We observed that the ML-enhanced RANS model led to improved predictions for most QoIs in almost all cases. We further observed that the use of Lipschitz-continuous networks and deep ensembles

increased model robustness. For subsonic boundary layer flows, numerical results demonstrated the higher-order tensor bases have no impact on the solution given that they only modulate the normal component of the Reynolds stress. For hypersonic boundary layers, the machine-learned model resulted in lower wall shear stress and heat flux as compared to the baseline $k-\epsilon$ model. Results for these cases demonstrated that the wall-normal Reynolds stress makes an appreciable contribution to the wall-normal momentum balance and impacts the overall flow. Lastly, for SBLIs and compression ramps, the ML corrections resulted in improved performance as compared to the baseline model. In particular, the size of the separation region was better predicted. However, the resulting model formulation tended to still under-predict the size of the separation region and over-predict heat flux near the point of reattachment.

The results in this chapter demonstrate the ability to develop a generic RANS model resulting in improved results over a variety of cases using data-driven techniques. For hypersonic SBLIs, however, we find that the improvements to the Reynolds stress model via TBNNs result in improved predictions for QoIs that are not, on the surface, "better" than classical improvements shown in Chapter 3, e.g., rapid compression corrections. While a generic model is appealing, the use-proposition for practical analyses remains unclear. The strongest such argument from the authors' perspective is that TBNNs result in a Reynolds stress model that is closer to truth, and fewer additional corrections are then required.

The developed TBNNs did not improve the over-prediction in wall heat flux near the point of reattachment. We hypothesize that this over-prediction is due to modeling of the turbulent heat flux, and the subsequent chapter will examine the development of variable turbulent Prandtl number models to address this issue.

This page intentionally left blank.

5. VARIABLE TURBULENT PRANDTL NUMBER MODELING

We develop a neural-network-based variable turbulent Prandtl number model for the $k-\epsilon$ turbulence model using field inversion and machine learning (FIML) for improved wall heating predictions in hypersonic shock boundary layer interactions (SBLIs). The model is developed by performing a finite-dimensional field inference for a spatially varying turbulent Prandtl number on six canonical SBLIs: three compression ramps at Mach 8 and three impinging shocks at Mach 5. The inference results identify a turbulent Prandtl number that reduces wall heating by systematically directing heat transfer away from the wall. An ensemble of neural networks is then trained on the inferred turbulent Prandtl number fields to develop a predictive model. We test the resulting variable turbulent Prandtl number model on a suite of test cases including the hollow cylinder flare and HIFiRE ground test experiments. We find that the learned model systematically and substantially improves wall heating predictions as compared to the baseline $k-\epsilon$ model, a $k-\epsilon$ model augmented with various high speed corrections, and the SST model.

5.1. Introduction

The modeling improvements to the Reynolds stress tensor proposed in Chapter 4 were seen to improve the predictive capability of RANS models across a suite of test cases. However, the TBNN corrections had a nominal impact when deployed on shock boundary layer interactions. In particular, TBNNs did not address the over-prediction in peak wall heat flux near the point of reattachment. This chapter aims to address this issue by developing a variable turbulent Prandtl number model to improve the model for the turbulent heat flux.

The organization of this chapter proceeds as follows. Section 5.2 provides an overview of the relevant literature. Section 5.3 outlines a field inference approach, while Section 5.4 presents inference results across our training cases. Next, Section 5.5 outlines our framework for learning the inferred fields, while Section 5.6 provides results on a suite of test cases. Section 5.7 provides a brief analysis and interpretation of how the ML model reduces the wall heat flux. Sections 5.8 and 5.9 discuss iterative and grid convergence. A chapter summary is provided in Section 5.10.

5.2. Literature review

Various efforts have targeted improved predictions for surface heating in SBLIs over the years. Here we provide a (necessarily) non-comprehensive review. In Refs. [13, 12], Coakley, Huang, and co-authors propose several modifications to two-equation models to improve their performance in hypersonic flows. In particular, they propose rapid compression corrections to enhance separation in SBLIs which result in improved predictions for wall pressure. They further propose a length-scale limiter that clips the turbulent length-scale near the wall so that it does not exceed an algebraic length-scale based on the distance to the wall. This length-scale correction results in improved predictions for wall heating.[1] Another approach to improve predictions for wall heating in SBLIs is to improve the model for the turbulent heat flux. Classically, the turbulent heat flux is modeled via a gradient-diffusion hypothesis with a constant turbulent Prandtl number model taking on a value of $Pr_t = 0.9$. This value is appropriate for canonical boundary layers, but is known to be deficient in SBLIs. To address this, researchers have pursued variable turbulent Prandtl number models. Xiao et al. [103] and Ott et al. [59] have examined the development of variable turbulent Prandtl number models that are based on the addition of extra transport equations for the temperature variance and its dissipation rate. These models incur a higher computational cost due to the need to solve additional transport equations, and require further development for high-speed SBLIs [59]. Another approach has been proposed by Roy and Sinha [70, 69], who developed a variable turbulent Prandtl number model based on the linearized Rankine Hugoniot conditions applied to shock turbulence interaction. The resulting model is dependent on non-local conditions (e.g., the density ratio across a shock wave), and an additional transport equation for a “shock strength" variable is added to make a local model. The model results in improved wall heating and has been demonstrated on a variety of hypersonic cases, but at the time of this writing, has not gained widespread use. In contrast to non-local or transport-equation-based models, several groups have pursued the development of algebraic turbulent heat flux and variable Prandtl number models [90, 6, 45, 99], several of which are targeted at hypersonic flows [90, 6]. These models are often based on approximations to the transport equations for the turbulent heat flux, but have yet to show systematic success on hypersonic SBLIs.

In recent years, SciML-based approaches have been deployed to improve heat flux modeling, and here we highlight work relevant to learning a variable turbulent Prandtl number model. First, to the best of our knowledge, Jordan [38] is the only work in the open literature that develops a data-driven turbulent Prandtl number tailored to hypersonics. In the approach, Jordan utilizes an equation learning approach and high-fidelity data from hybrid RANS/LES to train a variable turbulent Prandtl number model for the shear stress transport (SST) turbulence model. The model is deployed across a variety of hypersonic SBLI cases and shows consistent improvement. The performance of the model, however, was typically bounded by the performance of the Menter baseline model (BSL) and SST, which is problematic for cases where both models over-predict

[1] However, we note that the length-scale limiter also reduces wall shear which was not examined in Refs. [13, 12]. For the cases considered in this manuscript, this reduced wall shear is in worse agreement with the experimental data.

wall heat flux. Several other bodies of work have considered learning variable turbulent Prandtl number models, but not in the context of hypersonics. Xu et al. learn a variable turbulent Prandtl number model for buoyancy-effected flows using symbolic learning [104]. Milani, Ling, and co-authors develop data-driven models for scalar flux modeling [52, 50, 51]. Initial work focused on learning a gradient-diffusion-based model with a learned turbulent diffusivity, while later works developed scalar flux models by extending tensor-basis neural network formulations. In all works, it is assumed that high-fidelity data for the scalar flux are available. These models have been shown to improve heat transfer predictions in incompressible flows.

A challenge for building a data-driven turbulent Prandtl number model for hypersonics is the lack of experiments and simulations in which high-fidelity full-field data are available. Performing experiments that produce full-field measurements of the turbulent heat flux at hypersonic Mach numbers is extremely difficult, and at the time of this writing, only a few direct numerical simulations of hypersonic shock boundary layer interactions exist [91, 66, 14, 57].[2] There is, however, a considerable amount of experimental data for wall quantities of interest (wall pressure, wall heat flux, and wall shear stress) for hypersonic shock boundary layer interactions across a range of flow conditions. To this end, another relevant body of machine learning efforts are field inference (FI) based approaches [62, 83, 84, 80, 87]. In these methods, a corrective field is embedded in the (deficient) model, and the field is then inferred to minimize the misfit on the quantity of interest (QoI), e.g., wall quantities that have been measured in experiments. In terms of obtaining a predictive model, both two-step [62, 84, 80] (infer the field at each grid point, and then learn the field as a function of features) and one-step [87, 31, 30] (parameterize the corrective field in terms of a model, and directly infer the model parameters) approaches have been considered. Both approaches have resulted in improved model forms.

The goal of this chapter is to improve predictions for wall heating in hypersonic SBLIs. To this end, we develop a variable turbulent Prandtl (variable$-\mathrm{Pr_t}$) number model using the two-step FIML framework [80, 62, 84]. We pursue field inference-based approaches due to the limited amount of field data available for hypersonic SBLIs as well as the significant discrepancy between existing RANS models and truth. We further pursue the classic two-step approach proposed in Refs. [80, 62, 84], rather than the one-step approaches deployed in Refs. [87, 31, 30], so as to more easily enable training on a large suite of cases, explore various feature sets, and more effectively train an accurate neural network. We emphasize that we do not discount a one-step approach, and that the present work provides insight that can be leveraged in a one-step approach (e.g., feature sets that perform well). We develop our variable turbulent Prandtl number for the $k-\epsilon-$RC model documented in Chapter 3 that contains a rapid compression correction calibrated to improve the prediction of the separation region in hypersonic SBLIs. We develop our model for the $k-\epsilon$ model as opposed to the more widely used SST model due to the observation that $k-\epsilon$ gave more reliable predictions for the size of the separation region and peak wall shear (see Chapter 3). This allowed us to focus on improved predictions for wall heating. We emphasize that the presented

[2] We further note that the mismatch between experimental data and DNS simulations for these flows is oftentimes significant [91, 14].

approach is model agnostic and could be used to develop an improved SST model. To compare to other established approaches, we compare our developed variable turbulent Prandtl number model to its baseline variant using a constant turbulent Prandtl number, a $k-\epsilon$ model employing rapid compression corrections, the Catris and Aupoix corrections, and a length-scale limiter, and the SST turbulence model.

5.3. Field inference for a variable turbulent Prandtl number

The aforementioned modifications mitigate, but do not eliminate, the erroneous predictions of wall heating in SBLIs. We hypothesize that these deficiencies are largely attributed to the constant turbulent Prandtl number in the turbulent heat flux model (2.0.2) [105, 70, 91, 39, 57]. To this end, we develop a variable turbulent Prandtl number model via the field inversion and machine learning (FIML) approach.

To describe the FIML approach, we express the compressible Favre-averaged Navier–Stokes equations closed with a turbulence model on the spatial domain Ω as

$$\mathcal{R}\left(\boldsymbol{u}\right) = \mathbf{0},$$

where $\mathcal{R}$ is the residual (conservation equations for mass, momentum, energy, turbulent kinetic energy, and turbulent dissipation) and $\boldsymbol{u}$ is the PDE state vector, with components mass, momentum, energy, and conserved turbulence model variables. Additionally, let $\boldsymbol{s}$ denote a quantity of interest (e.g., wall heat flux).

To improve performance, FIML modifies the governing equations by introducing a field discrepancy term. Mathematically,

$$\mathcal{R}_{\mathrm{FI}}\left(\boldsymbol{u};\beta\right) = \mathbf{0},$$

where $\mathcal{R}_{\mathrm{FI}}$ is the modified residual function and $\beta : \Omega \rightarrow \mathbb{R}$ is the field discrepancy term. In the present work, we focus on modifying the turbulent heat flux for improved wall-heating predictions. To this end, we use the field discrepancy term to modify the turbulent Prandtl number arising in the energy equation. Mathematically, the modified turbulent heat flux is written as

$$q_j' = \beta \frac{c_p \mu_t}{\mathrm{Pr_t}} \frac{\partial T}{\partial x_j}.$$

The objective is to infer a spatially-varying correction β to minimize misfit in the wall heat flux.

5.3.1. Field inference

To infer the corrective field β, we solve the field inference problem

$$\begin{aligned} &\underset{\beta}{\text{minimize}}\ \mathcal{J}\left(\boldsymbol{s}\left(\boldsymbol{u},\beta\right),\mathbf{d}\right) \\ &\text{subject to}\ \mathcal{R}_{\mathrm{FI}}\left(\boldsymbol{u},\beta\right) = \mathbf{0}, \end{aligned} \tag{5.3.1}$$

where $\mathcal{J}$ is the objective function (e.g., mismatch between the RANS wall heat flux and high-fidelity measurements) and $\mathbf{d}$ is high-fidelity data. We employ gradient-based optimization with a steepest decent algorithm to solve the inference problem. We set $\beta^0(x) = 1$, compute the gradient of the objective function, and update the correction as

$$\beta^{n+1} = \beta^n - \epsilon \frac{d\mathcal{J}}{d\beta},$$

where $\epsilon \in \mathbb{R}^+$ is the step size. The optimization continues for $n = 1, 2, \ldots$ until convergence is reached.

A principal challenge for solving the optimization problem (5.3.1) via gradient-based optimization is computation of the (discretely high-dimensional) gradient, $\frac{d\mathcal{J}}{d\beta}$. The most effective approach is (discrete) adjoint-based optimization. In this approach an auxiliary set of adjoint equations are solved, and from this auxiliary set of equations the gradient of the objective function with respect to the field correction can be computed in one step. While effective, adjoint-based formulations for field inversion are not available or easily accessed in a variety of codes.

An alternative approach, and the one employed here, is to perform a finite-dimensional field inference by restricting the corretion to live within a low-dimensional affine subspace, $\beta \in \mathcal{V}$ with $\dim(\mathcal{V}) = K$. To this end, let $\boldsymbol{\phi}_i$, $i = 1, \ldots, K$ with $\text{span}\{\boldsymbol{\phi}_i\}_{i=1}^K + 1 = \mathcal{V}$ denote a (discretely) orthonormal basis spanning $\mathcal{V} - 1$. We note the use of an affine offset of value 1 to exactly represent a constant field with a value of 1, which corresponds to the baseline model (i.e., no correction). The field correction can be written as

$$\beta(x) = \sum_{i=1}^{K} \boldsymbol{\phi}_i(x)\, \hat{\beta}_i + 1.$$

The advantage of this representation is that the derivative of the objective function can be obtained with $O(K)$ forward evaluations of the model; in the present work we employ first-order finite differences, which requires $K + 1$ evaluations. This approach, while more expensive than adjoint-based optimization, requires only a primal solver and is less intrusive.

5.3.2. Random field expansions

Performing the finite-dimensional field inference step outlined above requires identifying the basis functions, $\boldsymbol{\phi}_i$, $i = 1, \ldots, K$. We employ random fields and principal component analysis for this purpose. To this end, let $\mathcal{P}$ denote a probability space and let T denote a random field on Ω such that $\forall x \in \Omega$, $T(x)$ is a random variable on $\mathcal{P}$. Further, let T_i, $i = 1, \ldots, n_s$ denote i.i.d. samples of T. Given these samples, we identify a low-dimensional basis for the random field by solving the optimization problem

$$\underset{\{\boldsymbol{\phi}_i\}_{i=1}^K, (\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)_w = \delta_{ij}}{\text{minimize}} \sum_{i=1}^{n_s} \| (T_i - 1) - \sum_{i=1}^{K} (\boldsymbol{\phi}_j, T_i - 1)_w \boldsymbol{\phi}_j \|_w^2 \tag{5.3.2}$$

where $(\cdot,\cdot)_w$ denotes the $L^2(\Omega)$ weighted inner product, $w : \Omega \to \mathbb{R}^+$ denotes the weighting, and $\|v\|_w^2 = \int_\Omega w(x)v(x)^2 d\Omega$ denotes the weighted $L^2(\Omega)$ norm.[3] Here, we employ a weighting such that the bases are ℓ^2 orthonormal at the discrete level. In practice, this optimization problem can be solved with the singular value decomposition. Algorithm 2 outlines the algorithm used in this work.

Algorithm 2 Algorithm for obtaining random field expansions. We use the $\vec{\cdot}$ notation to denote discrete quantities.

Input: Probability space, $\mathcal{P}$, energy criterion, tol, and number of samples, n_s
Output: (Discrete) basis vectors, $\boldsymbol{\phi}_i$, $i = 1, \ldots, K$.

for $i = 1, \ldots, n_s$ **do**
  $\vec{T}_i \sim \vec{\mathcal{P}}$ ▷ Draw (discrete) sample from probability space
end for
$\mathbf{S} = \left[\vec{T}_1, \ldots, \vec{T}_s\right]$ ▷ Collect samples into snapshot matrix
$\mathbf{U}, \sigma, \mathbf{V} = \text{svd}\,(\mathbf{S})$ ▷ Perform (thin) singular value decomposition
$e(k) = \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{n_s} \sigma_i^2}$ ▷ Compute energy captured in the first k singular values
$K = \arg\min_{k^*} \|e(k^*) - \text{tol}\|$ ▷ Determine reduced basis size using energy criterion
$\boldsymbol{\phi}_i = \mathbf{U}_i, i = 1, \ldots, K$ ▷ Set reduced basis to be the first K left singular vectors

The expressiveness and dimension of the field expansions depends on the probability space $\mathcal{P}$. In the present work, we define the probability space as a d-dimensional Gaussian (where d is the number of spatial dimensions, i.e., 1, 2, or 3) with mean 0 and covariance

$$\begin{aligned} K : (\boldsymbol{x}, \boldsymbol{y}) &\mapsto \sigma^2 e^{(\boldsymbol{x}-\boldsymbol{y})^T \boldsymbol{\Sigma} (\boldsymbol{x}-\boldsymbol{y})} \\ &: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+ \end{aligned} \tag{5.3.3}$$

with

$$\boldsymbol{\Sigma} = \text{diag}\,(\ell_1, \ldots, \ell_d)\,.$$

The length-scales ℓ_d define the correlations in the dth dimension, respectively. The value chosen for these will be detailed in the following section.

Before proceeding, we remark that the field inference results will ultimately depend on the random field expansion, and different probability spaces will result in different solutions. We emphasize that the objective of the random field expansions is not to identify a perfect, unique solution β; rather it is a means to an end to obtain a finite-dimensional inference problem that is used to identify fields that can be learned to develop an improved model. Due to the computational (and

[3] In practice, we solve the optimization problem by simply performing SVD at the discrete level, which formally corresponds to minimizing in a weighted two norm where the weighting is defined by the cell volumes.

Table 5-1. Summary of cases for field inference.

Case	M_∞	Re_L	Shock/ramp angle	T_w/T_0
Holden Compression Corner Run 12	8.2	146.9×10^6	27°	0.29
Holden Compression Corner Run 19	8.1	139.3×10^6	33°	0.29
Holden Compression Corner Run 24	8.0	145.3×10^6	36°	0.29
Schülein impinging shock	5.0	18.5×10^6	6°	0.73
Schülein impinging shock	5.0	18.5×10^6	10°	0.73
Schülein impinging shock	5.0	18.5×10^6	14°	0.73

human time) expense of performing the finite-dimensional field inference across a suite of cases we only present results for one inference process here, but we do note that we have explored various inference hyper-parameters (e.g., correlation lengths, energy cutoffs) and results follow the same patterns.

5.4. Field inference results

We now present field inference results. We perform field inference on six cases: three Mach 8 compression ramps with corner angles of 27°, 30°, and 33° degrees from Holden et al. [28] and three impinging shock boundary layer interactions at Mach 5 with impinging shock angles of 6°, 10°, and 14° with experimental data from Schülein [78]. Figure 5-1 shows contours of the density field for each case to depict the flow and Table 5-1 gives the conditions for each case. All the training cases are two-dimensional ($d = 2$). For all inference cases, the objective function is the square of the ℓ^2 norm of the point-wise error between experimental wall heat flux data and the RANS predictions, i.e.,

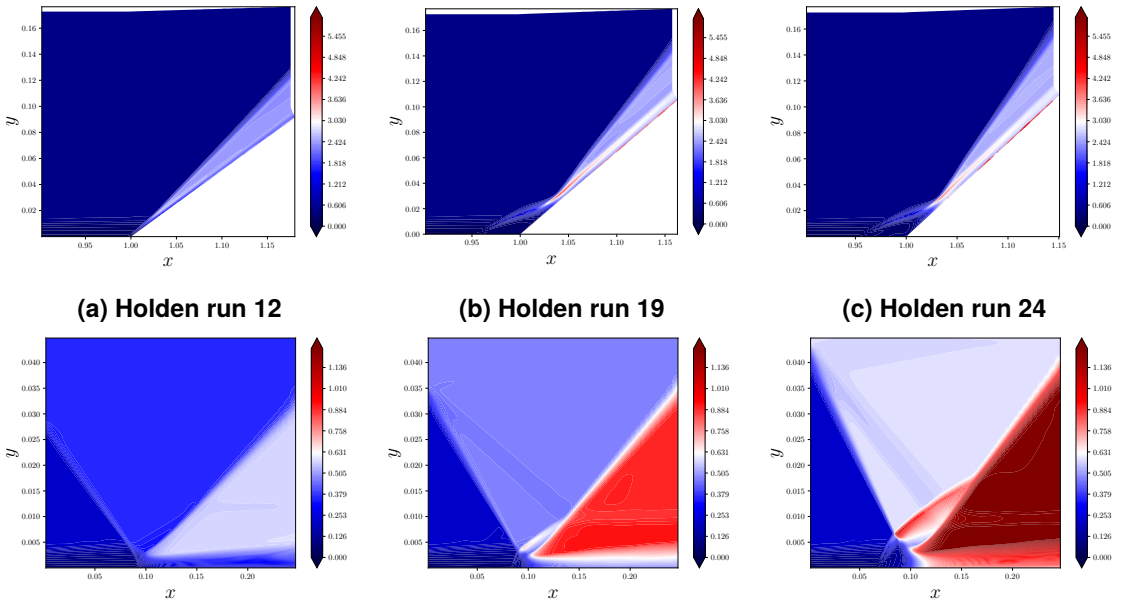
$$\mathcal{J}(\boldsymbol{s}, \mathbf{d}) = \sum_{i=1}^{n_d} \left(q_{\text{wall}}^{\text{exp}}(x^i) - q_{\text{wall}}^{\text{RANS}}(x^i) \right)^2,$$

where n_d are the number of available experimental data points and x^i refers to the wall location at which the data point is recorded.

We now discuss the definition of the probability space. For all inference cases, we employ a Gaussian random field in a transformed coordinate system. Specifically, we set the coordinate system in Equations (5.3.3) to be

$$\boldsymbol{x} = \left[x_1, \log\left(\frac{d_{\text{wall}}}{d_{\text{wall}}^{\text{min}}} \right) \right] \tag{5.4.1}$$

where x_1 (m) is the standard streamwise coordinate, d_{wall} (m) is the distance to wall in meters, and $d_{\text{wall}}^{\text{min}}$ is the minimum distance to wall value in the computational mesh. We use this transformation



(a) Holden run 12 **(b) Holden run 19** **(c) Holden run 24**

(d) Schülein 6° **(e) Schülein** 10° **(f) Schülein** 14°

Figure 5-1. Density contours for Holden compression ramps (top) and the Schülein SBLIs (bottom).

to enable more resolution near the wall. Additionally, we clip the value of the random field to lie within the range $10^{-2} \leq \beta \leq 100$. For all cases, we generate 400 samples of the random field and perform principal component analysis with an energy criterion of 0.95 to determine the basis, i.e., we use Algorithm 2 with inputs tol = 0.95, $n_s = 400$, and a probability space corresponding to Equations (5.3.3) with the length-scales and cutoffs described in this section.

For the Holden compression ramps, the value of $d_{\text{wall}}^{\text{min}}$ is 2.7×10^{-8} (m), 2.46×10^{-8} (m), and 2.42×10^{-8} (m) for run 12, run 19, and run 24, respectively. For length-scales, we employ $\ell_1 = 0.02$ (m) and $\ell_2 = 1.0$. We additionally restrict the random field to only be active for $x \geq 0.95$ (m) and $d_{\text{wall}} \leq 0.03$ (m). For the Schülein SBLIs, $d_{\text{wall}}^{\text{min}} = 5.625 \times 10^{-7}$ (m). As length-scales, we use $\ell_1 = 0.01$ (m) and $\ell_2 = 0.5$. We restrict the random field to only be non-zero for $d_{\text{wall}} \leq 0.02$ (m). When performing field inference, we only infer against experimental data for $x^i \leq 0.175$ (m); after this point there is an expansion wave in the experiment which is not modeled in the RANS.

Lastly, we make a remark on grid convergence. We have observed that obtaining grid-converged solutions for hypersonic SBLI cases like the ones tested here is difficult and requires resolution beyond typical engineering tolerances. The peak heat flux and shear stress near the point of reattachment are, in particular, difficult to converge. Since our inference technique requires many evaluation of the forward model, performing it on our finest level of grids would be very expensive. To accelerate our inference process, it is performed on meshes that are slightly coarser than what is required for a grid converged solution. Further refinement to grid converged solutions results in ±5% changes to peak shear stress and heat flux. Mesh convergence is detailed in Section 5.9.

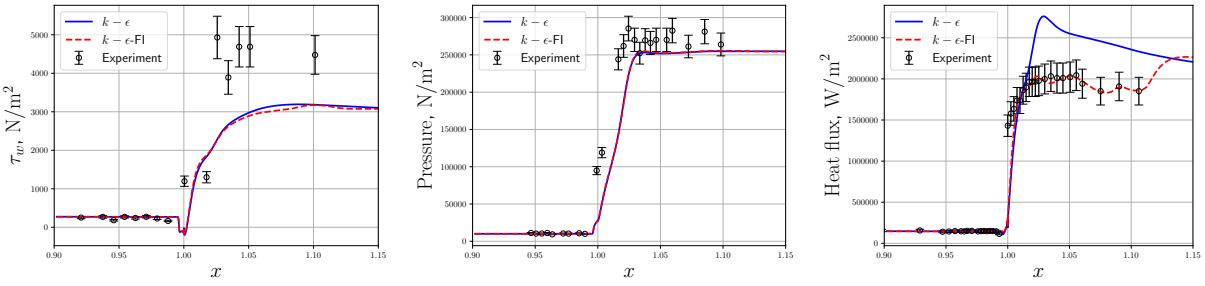
We emphasize that all results will be presented on our finest level of meshes when we assess the predictive capabilities of our trained model in Section 5.6.

Figures 5-2 and 5-3 present inference results for the three Holden compression ramps, while Figures 5-4 and 5-5 present inference results for the Schülein cases. Figures 5-2 and 5-4 show wall QoIs, while Figures 5-3 and 5-5 show the inferred turbulent Prandtl number fields. We observe the following:

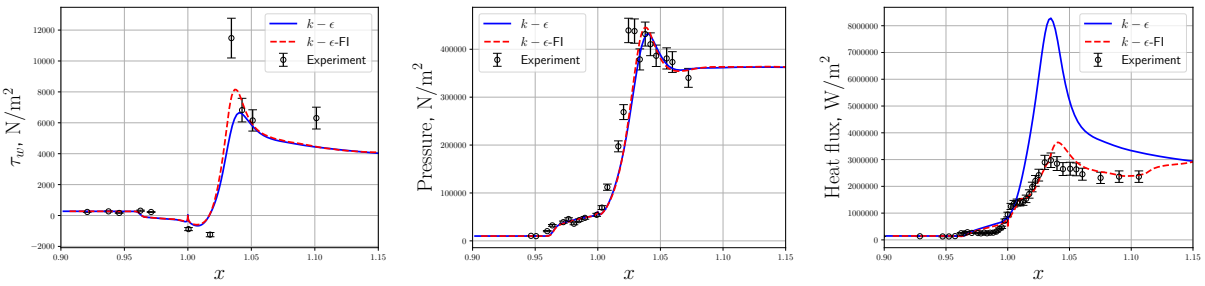
- In all cases, the inferred solution is able to address the over-prediction in wall heat flux at and downstream of reattachment.
- There is a rise in wall heat flux downstream of the interval in which experimental data are available. In practice, this rise in heat flux causes little concern as ultimately this portion of the field will not be used for training.
- Wall shear stress and wall pressure profiles in the inferred solutions are mostly unchanged from the baseline model. The most notable deviations are in wall shear stress for the compression ramps.
- Wall quantities of the inferred models display small oscillations; this is a result of the finite-dimensional inference.
- For the compression ramps, the inferred turbulent Prandtl number fields are characterized by a region of high $\mathrm{Pr_t}$ very near the wall (in many areas, $\mathrm{Pr_t} > 10$) downstream of reattachment. This region plays the critical role in reducing the wall heat flux. As will be detailed Section 5.7, in this region the wall-normal turbulent heat flux is negative (moving heat towards the wall). Increasing the turbulent Prandtl number acts to decrease the magnitude of the turbulent heat flux, which in turn reduces heat transfer towards the wall.
- For the Schülein cases, the inferred turbulent Prandtl number fields are characterized by a coherent and consistent region of low $\mathrm{Pr_t}$ after the interaction in the area $0.1 \leq x \leq 0.2$ (m) and $1 \times 10^{-5} \leq y \leq 1 \times 10^{-3}$ (m). We have observed that this change is the primary driver for a reduced heat flux. An explanation for this is again provided in Section 5.7, where it is shown that, in this region, the wall-normal turbulent heat flux is positive (moving heat away from the wall). Decreasing $\mathrm{Pr_t}$ results in an increased turbulent heat flux magnitude, which in turn moves more heat away from the wall.

5.5. Machine learning

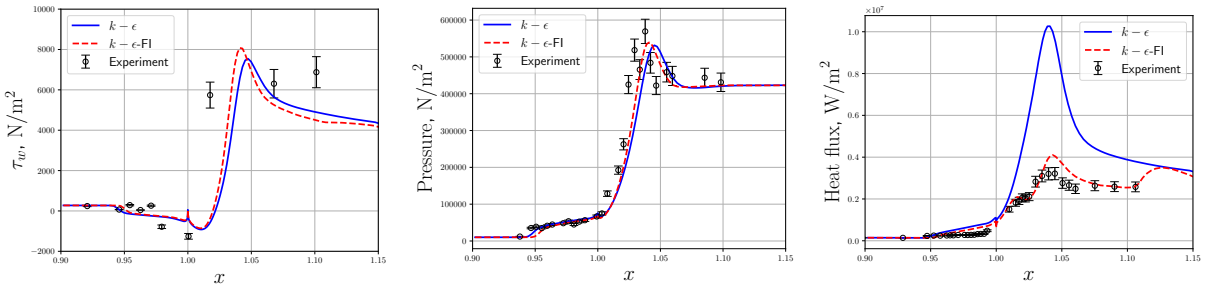
Having inferred β across multiple datasets, the task is now to learn it using local flow features to produce a predictive model. This section describes our ML formulation. Before describing this we remark that, in the above section, we highlighted that regions of high $\mathrm{Pr_t}$ very near the wall played an important role in reducing wall heat flux for compression ramps. Compression ramps are of



(a) Wall shear stress (run 12). **(b) Wall pressure (run 12).** **(c) Wall heat flux (run 12).**

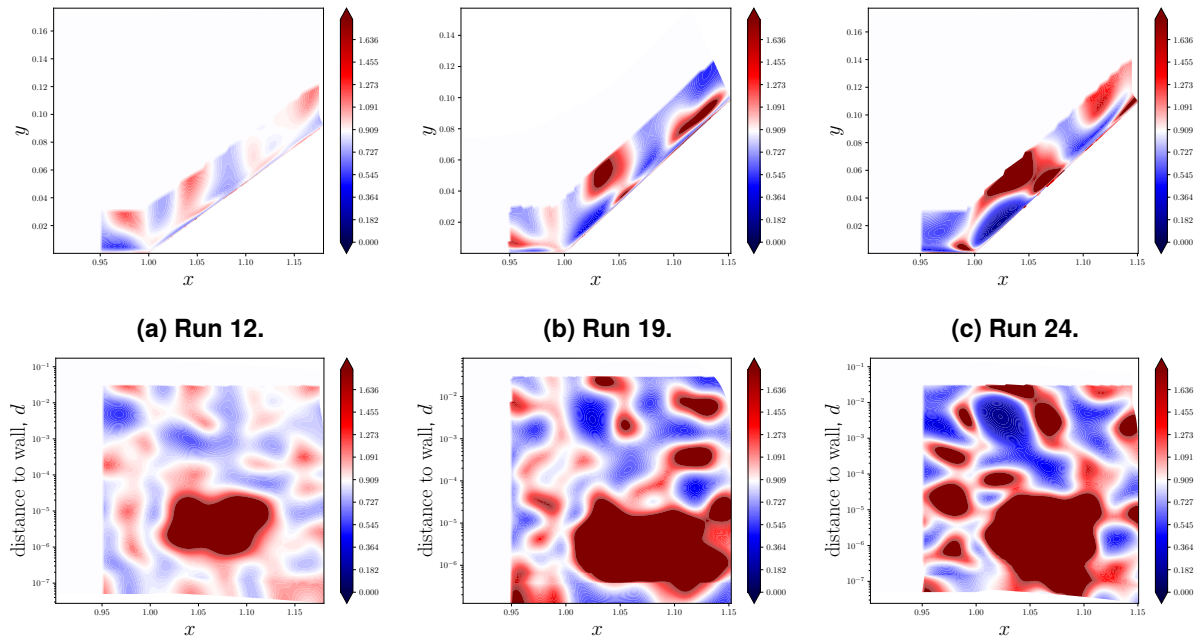


(d) Wall shear stress (run 19). **(e) Wall pressure (run 19).** **(f) Wall heat flux (run 19).**



(g) Wall shear stress (run 24). **(h) Wall pressure (run 24).** **(i) Wall heat flux (run 24).**

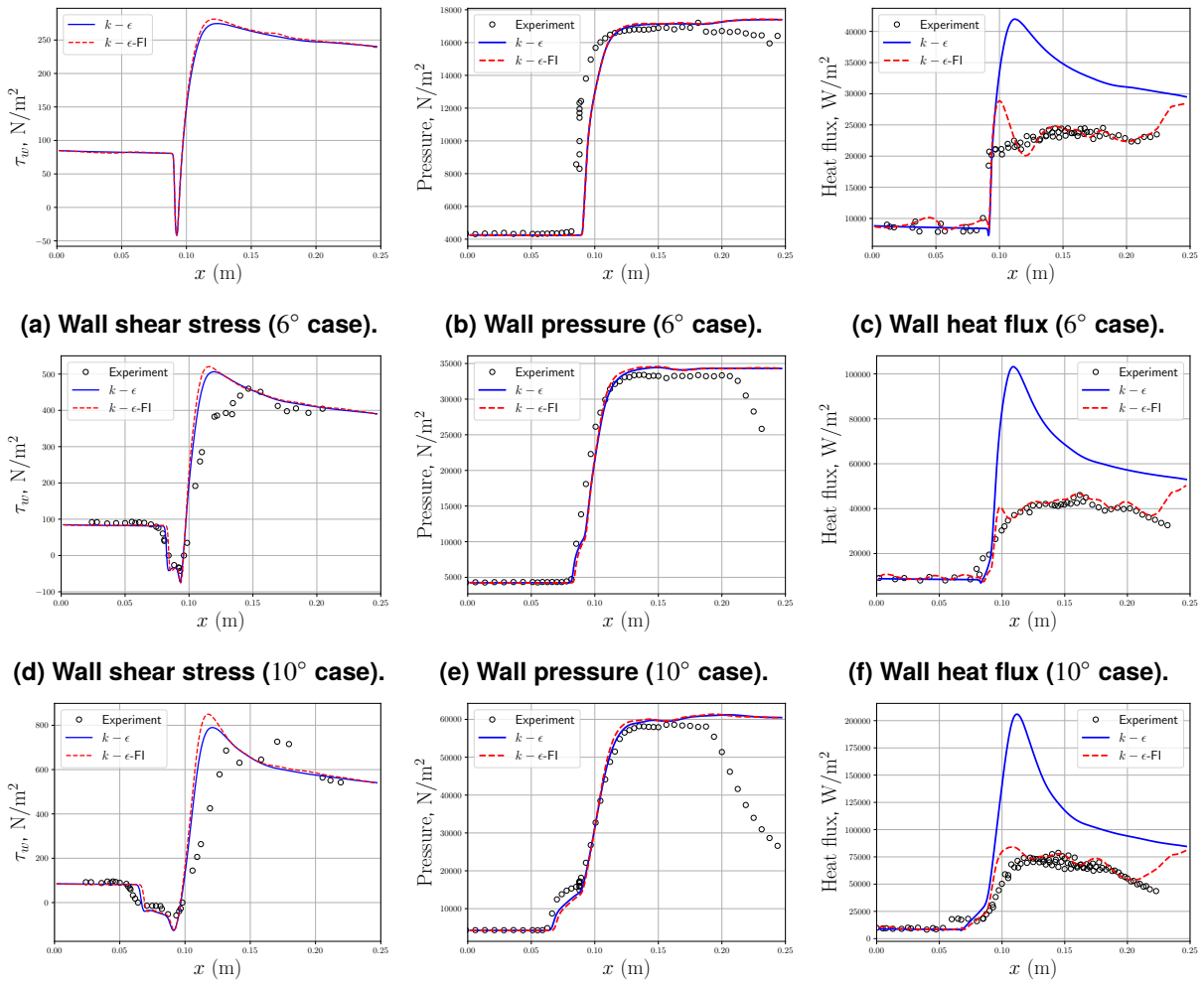
Figure 5-2. Holden compression ramps. Inference results for wall QoIs. The uncertainty bounds on experimental data are estimates given in Ref. [22].



(a) Run 12. **(b) Run 19.** **(c) Run 24.**

(d) Run 12 (log scale). **(e) Run 19 (log scale).** **(f) Run 24 (log scale).**

Figure 5-3. Holden compression ramps. Inferred turbulent Prandtl number, $\mathrm{Pr_t}$. The top row shows results in a linear scale while the bottom results shows results in a log scale using distance to wall as the vertical axis.

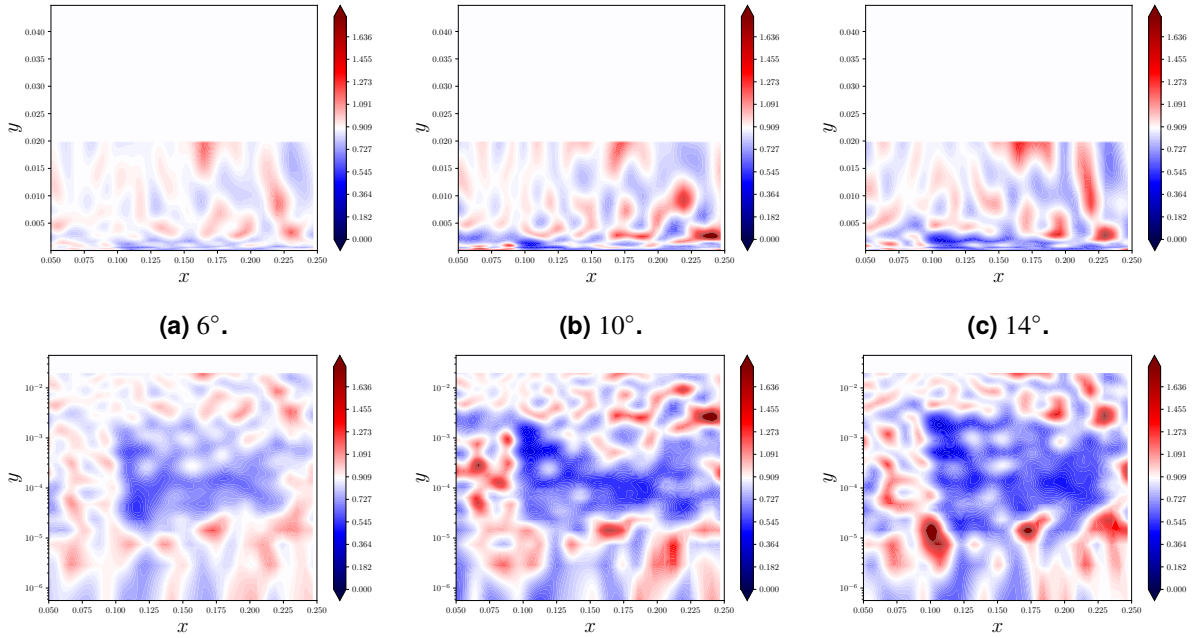


(a) Wall shear stress (6° case). **(b) Wall pressure (6° case).** **(c) Wall heat flux (6° case).**

(d) Wall shear stress (10° case). **(e) Wall pressure (10° case).** **(f) Wall heat flux (10° case).**

(g) Wall shear stress (14° case). **(h) Wall pressure (14° case).** **(i) Wall heat flux (14° case).**

Figure 5-4. Schülein SBLIs. Inference results for wall QoIs for the 6° (top), 10° (middle), and 14° (bottom) cases. Note: The downstream expansion fan present in the experimental data is not represented in the RANS as the present configuration only involves an impinging shock on a flat plate.



(a) 6°. **(b)** 10°. **(c)** 14°.

(d) 6° **(log scale).** **(e)** 10° **(log scale).** **(f)** 14° **(log scale).**

Figure 5-5. Schülein SBLIs. Inferred turbulent Prandtl number for the 6° **(left),** 10° **(center), and** 14° **(right) cases. The top row shows results in a linear scale while the bottom row shows results in a log** y **scale.**

primary interest to us, and to be able to most effectively learn this aspect, we choose to learn on a transformation of $\mathrm{Pr_t}$ of the form

$$y = \log\left(\frac{\beta}{\mathrm{Pr_t^{nom}}}\right), \tag{5.5.1}$$

where $\mathrm{Pr_t^{nom}} = 0.9$ is the standard value. The motivation for this transformation is that it will "stretch" the co-domain of the model for regions where $\mathrm{Pr_t}$ is high and more easily enable us to represent this phenomena.

5.5.1. Features

We use seven features in our variable turbulent Prandtl number model. The first two features are kinematic features based on the invariance of the strain and rotation tensors, the second two features are Reynolds number-based features, while the final three features are designed to be active in regions where compressibility and heat transfer effects are important. The feature set is given by

$$\boldsymbol{\eta} = \left[\mathrm{Tr}(S^{*2}) \quad \mathrm{Tr}(\Omega^{*2}) \quad \log(\mu_t/\mu) \quad 3\tanh\left(\frac{1}{3}d_{\mathrm{wall}}\frac{\sqrt{k}}{50\nu}\right) \quad \frac{\frac{\partial \tilde{T}}{\partial n}\left(\frac{\nu^3}{\tilde{\epsilon}}\right)^{\frac{1}{4}}}{\tilde{T}} \quad \frac{\sqrt{k}}{\sqrt{\gamma R\tilde{T}}} \quad \min\left(\frac{k^{3/2}/\epsilon}{d_{\mathrm{wall}}}, 5\right) \right]^T.$$

Here, $S^*_{ij} = \left(S_{ij} - \frac{1}{3}\frac{\partial \tilde{u}_k}{\partial x_k}\delta_{ij}\right)\tilde{k}/\tilde{\epsilon}$, $\Omega^*_{ij} = \frac{1}{2}\left(\frac{\partial \tilde{u}_i}{\partial x_j} - \frac{\partial \tilde{u}_j}{\partial x_i}\right)\tilde{k}/\tilde{\epsilon}$ are non-dimensional strain and rotation rates, n is the wall-normal vector, and we have used the notation $Q^2 = Q_{ij}Q_{jk}$ for a tensor $Q \in \mathbb{R}^{3\times3}$. The first two features are the first two invariants of the strain and rotation rate tensors, the third is the log of the turbulent viscosity ratio, and the fourth is the wall distance Reynolds number with a smooth cutoff. The fifth feature was selected to highlight thermodynamic effects. We have used the wall-normal temperature gradient instead of the magnitude of the gradient upon the observation that the sign of the temperature gradient matters for wall heating. The sixth feature, which is the turbulent Mach number, enables Mach number sensitivity and increased the capacity of our models when learning on both incident shock boundary layer interactions and compression ramps at different Mach numbers. The last feature is motivated by the work of Coakley and Huang [13] and is the ratio of the length-scale predicted by the RANS model to a distance-to-wall length-scale. We observed that capping this length-scale ratio improved learning.

5.5.2. Fully connected neural networks

We employ fully connected feed forward neural networks for learning y (and hence a variable turbulent Prandtl number). To this end, let $\left(\mathbf{w}^i, \mathbf{b}^i\right)$, $i = 1, \ldots, N_{\text{layer}}$ denote the weights and biases of an N_{layer} neural network. Given input features $\boldsymbol{\eta} \in \mathbb{R}^N$, we learn a (transformed) variable turbulent Prandtl number model as

$$y^{\text{ML}} : (\boldsymbol{\eta}; \boldsymbol{\theta}) \mapsto g\left(\cdot, \mathbf{w}^{N_{\text{layer}}}, \mathbf{b}^{N_{\text{layer}}}\right) \circ \cdots \circ g\left(\boldsymbol{\eta}^*, \mathbf{w}^1, \mathbf{b}^1\right),$$

where g is the activation function, e.g., ReLU, $\boldsymbol{\theta} = \{\mathbf{w}^1, \mathbf{b}^1, \cdots, \mathbf{w}^{N_{\text{layer}}}, \mathbf{b}^{N_{\text{layer}}}\}$ denotes the collective set of weights and biases, and $\boldsymbol{\eta}^*$ denotes the normalized features. To improve the robustness of our model, we include additional techniques to improve regularity and robustness of the neural networks.

5.5.2.1. Lipschitz-continuous neural networks

Similar to the previous chapter, we employ Lipschitz continuous neural networks for learning our variable turbulent Prandtl number model. We refer the reader to Section 4.5 for more details.

5.5.2.2. Ensemble prediction

To further improve the robustness of our networks, we employ an ensemble approach leveraging multiple learned networks. In the approach, we train an ensemble of networks and utilize the mean prediction of the ensemble. While not pursued here, we highlight that such an ensemble approach can be additionally used to approximate model-form uncertainties by analyzing the variance of the ensemble outputs [43]; we use this approach in the previous chapter.

Table 5-2. Summary of training dataset.

Case	M_∞	Re_L	Shock/ramp angle	T_w/T_0	n_{data}
Holden Compression Corner Run 12	8.2	146.9×10^6	27°	0.29	2774
Holden Compression Corner Run 19	8.1	139.3×10^6	33°	0.29	3464
Holden Compression Corner Run 24	8.0	145.3×10^6	36°	0.29	2832
Schülein impinging shock	5.0	18.5×10^6	6°	0.73	10881
Schülein impinging shock	5.0	18.5×10^6	10°	0.73	10881
Schülein impinging shock	5.0	18.5×10^6	14°	0.73	10881
Mach 6 boundary layer	5.84	8.08×10^6	N/A	0.23	5588
Mach 6 boundary layer	5.86	7.91×10^6	N/A	0.69	2775
Mach 14 boundary layer	13.64	6.76×10^6	N/A	0.17	11020

Mathematically, the ensemble approach trains M networks targeted at learning y. For each network, we minimize the loss function

$$\boldsymbol{\theta} = \underset{\boldsymbol{\theta}^* \in \mathbb{R}^{N_{\boldsymbol{\theta}}}}{\arg\min} \sum_{k=1}^{N_{\text{train}}} \ell\left(y_i, y^{\text{ML}}\left(\boldsymbol{\eta}_i; \boldsymbol{\theta}^*\right)\right),$$

where ℓ denotes the loss function, $\boldsymbol{\eta}_i$ and y_i, $i = 1, \ldots, N_{\text{train}}$ denote the ith feature-response pair, and $N_{\boldsymbol{\theta}}$ denotes the total number of weights and biases. The mean of the networks is given by

$$\overline{y}^{\text{ML}} = \frac{1}{M}\left(\sum_{i=1}^{M} y^{\text{ML}}(\boldsymbol{\eta}; \boldsymbol{\theta}_i)\right),$$

where $\boldsymbol{\theta}_i$ denote the weights and biases for the ith network.

5.5.3. Training data collection and machine learning details

We train on data extracted from the field inference solutions presented in Section 5.3. For the Schülein cases, we learn on data for $0.05 \leq x_1 \leq 0.175$ and $x_2 \leq 0.01$. For the Holden cases, we learn on data for $0.95 \leq x_1 \leq 1.1$ and $d_{\text{wall}} \leq 0.01$. Additionally, to maintain the performance of the model for standard boundary layers, we supplement the field inference data with data from three hypersonic zero pressure gradient flat plate boundary layers [105]. For each of these flat plate cases, we assign a target turbulent Prandtl number of $\mathrm{Pr}_t = 0.9$. Table 5-2 summarizes the final dataset on which we learn.

For learning, we employ a standard mean-squared-error loss,

$$\ell(y, y^{\text{ML}}) = \left(y^{\text{ML}} - y\right)^2.$$

Table 5-3. Summary of cases the ML model is tested on. Cases in blue correspond to cases that are included in the training set, while cases in blue are outside of the training set.

Case	M_∞	Re_L	Shock/ramp angle	T_w/T_0
Holden Compression Corner Run 12	8.2	146.9×10^6	27°	0.29
Holden Compression Corner Run 19	8.1	139.3×10^6	33°	0.29
Holden Compression Corner Run 24	8.0	145.3×10^6	36°	0.29
Schülein impinging shock	5.0	18.5×10^6	6°	0.73
Schülein impinging shock	5.0	18.5×10^6	10°	0.73
Schülein impinging shock	5.0	18.5×10^6	14°	0.73
Hollow cylinder flare run 13	6.01	16.24×10^6	36°	0.20
Hollow cylinder flare run 18	6.96	5.18×10^6	36°	0.14
HIFiRE-1 Run 30	7.2	16.9×10^6	33°	0.13
HIFiRE-1 Run 46	6.6	30.8×10^6	33°	0.16

We note that we explored various objective functions, and that the above objective function slightly favored performance on the compression ramp cases which are of more direct interest to us. We use fully connected neural networks with three hidden layers and 30 nodes per layer. We employ an ensemble of size $M = 10$. We use ReLU as an activation function[4]. We train for 1000 epochs using a batch size of 50. We include ℓ^2 regularization with a very weak penalty parameter of $\lambda = 1 \times 10^{-9}$. We use standard normalization (center about the mean and scale by the standard deviation) for the input features. We do not normalize the output target data. The layer-wise Lipschitz constant computed from Algorithm 1 is approximately $k_{\text{lip}} = 2.5$. We use an 80/20 training/validation split in training, use an initial learning rate of $\text{lr} = 1 \times 10^{-4}$, and use a learning rate decay parameter of 0.9998 (i.e., the learning rate after the nth epoch is $\text{lr} \times (0.9998)^n$).

5.5.4. Implementation

The machine-learned variable turbulent Prandtl number model is implemented in SPARC. We do not modify any of the Jacobian entries and as such treat the variable turbulent Prandtl number model explicitly in the solver. Lastly, we note that we only call the ML model every n_{skip} iterations to accelerate the model evaluation. In the examples presented in this work we use $n_{\text{skip}} = 50$.

5.6. Predictions

We now present predictions from the ML-embedded RANS model. First we present training results and feature importance. Next, we present results for the three Holden and three Schülein cases on which the models were trained. Following this, we present predictive results on the CUBRC hollow cylinder flare [27] and HIFiRE ground tests [93]. Table 5-3 summarizes the cases. All cases are computed on refined meshes; we again refer the reader to Section 5.9 for more details on grid convergence. Specific run schedules differ for each case. Residual and QoI convergence for a representative case are presented in Section 5.8, and here we note that employing the ML model in the solver did not have a noticeable impact on iterative convergence. In what follows, we will refer to the $k-\epsilon-$RC model using the ML variable Prandtl number model as the $k-\epsilon-$RC$-$ML model. We will additionally compare results to the $k-\epsilon-$CA–RC–LC model given in Equation (2.2.3) and the standard SST model as detailed on the NASA turbulence modeling resource [73].

5.6.1. Machine learning results

Figure 5-6 shows the loss function as a function of epoch (training iteration) for each of the neural network ensemble members. The loss function computed on the training set is shown in blue, while the loss function computed on the validation set is shown in red. We observe that the loss is decreased by approximately a factor of $5-10$x from the first epoch. We observe some deviation in loss between the different ensemble members, but the overall trends are consistent. The loss between the training and validation data are both still (slowly) decreasing after 1000 epochs, suggesting that the networks are not overfit to the training data.[5]

Next, Figure 5-7 shows feature importance as computed using a feature importance permutation algorithm for the first four ensemble members. The algorithm is provided in Alg. 3. The main idea is to contaminate a feature by shuffling its input data and then assess the impact to the loss function. A high impact suggests that the feature is important. We observe that the ratio of eddy viscosity to laminar viscosity, wall-distance Reynolds number, and non-dimensional wall-normal temperature gradient are consistently identified as the most important features. The kinematic features corresponding to the strain and rotation tensors are consistently the least important features.

5.6.2. Predictions on training cases: Holden compression ramps and Schülein impinging shocks

Figure 5-8 shows predictions across the Holden compression ramp cases, while Table 5-4 tabulates relative wall heat flux errors. While not as accurate as the field inference predictions, the ML

[4] We have observed similar results with tanh.

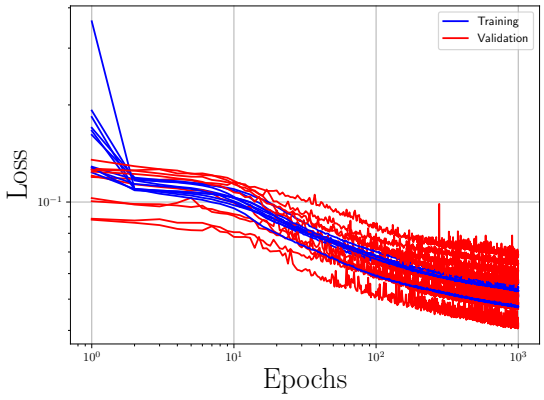
[5] We emphasize that this does not mean the networks will be predictive on out-of-distribution data.

Algorithm 3 Permutation algorithm for feature importance

Input: Feature-response pairs, $(\boldsymbol{\eta}_i, y_i)$, $i = 1, \ldots, N_{\text{train}}$, trained model, y^{ML}
Output: Feature importance scores, S_j, $j = 1, \ldots, N$

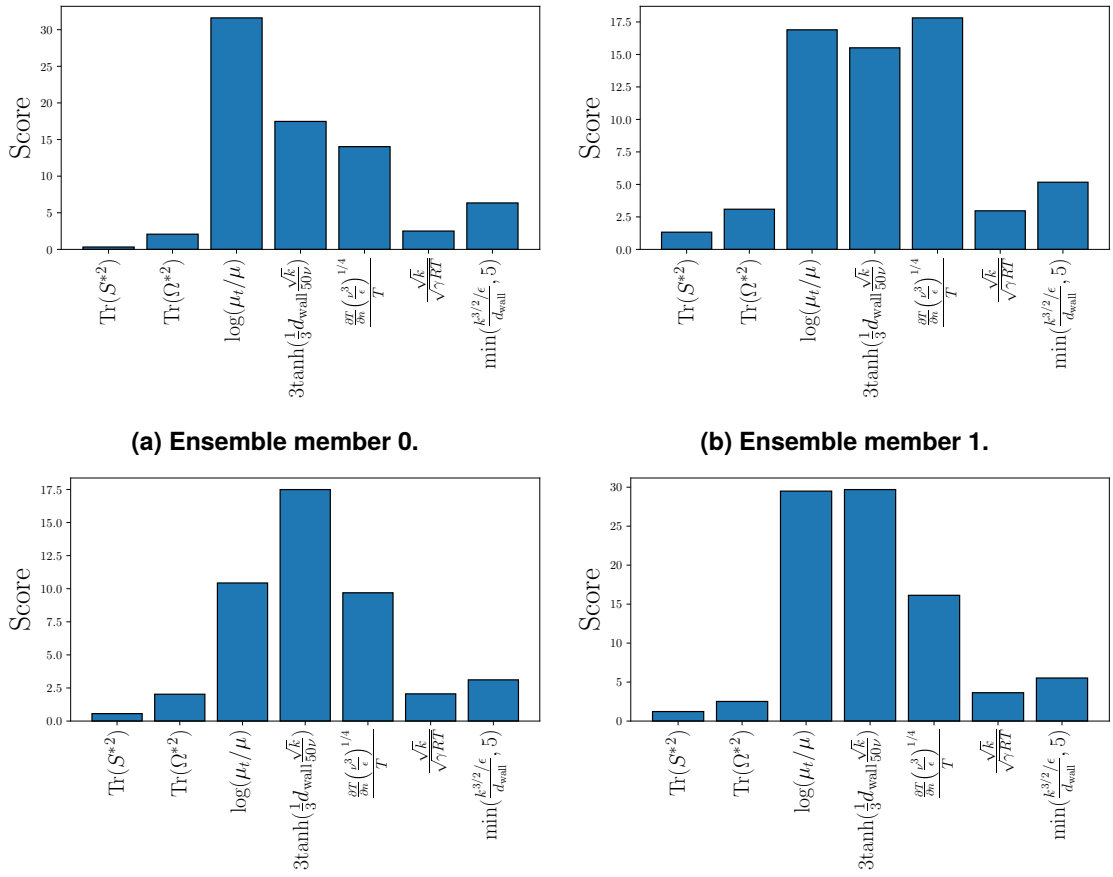
$L = 0$
for $i = 1, \ldots, N_{\text{train}}$ **do**
  $L = L + \ell\left(y_i, y^{\text{ML}}\left(\boldsymbol{\eta}_i; \boldsymbol{\theta}^*\right)\right)$ ▷ Compute loss of the baseline model
end for

for $j = 1, \ldots, N$ **do** ▷ Loop over the number of features
  $L_j = 0$
  $\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta}$ ▷ Copy features
  $\tilde{\boldsymbol{\eta}}^j = \text{shuffle}\left(\boldsymbol{\eta}^j\right)$ ▷ Shuffle data for the jth feature
  for $i = 1, \ldots, N_{\text{train}}$ **do**
    $L_j = L_j + \ell\left(y_i, y^{\text{ML}}\left(\tilde{\boldsymbol{\eta}}_i; \boldsymbol{\theta}^*\right)\right)$ ▷ Compute loss of model with the jth feature shuffled
  end for
  $S_j = \frac{|L_j - L|}{|L|}$ ▷ Compute feature importance score
end for



(a) Loss for $k-\epsilon-$RC$-$ML model.

Figure 5-6. Machine learning results. Decrease in loss function for the $k-\epsilon-$RC$-$ML model.



(a) Ensemble member 0.

(b) Ensemble member 1.

(c) Ensemble member 2.

(d) Ensemble member 3.

Figure 5-7. Machine learning results. Feature importance for the first four members of the ensemble. A high score suggests that the feature is important.

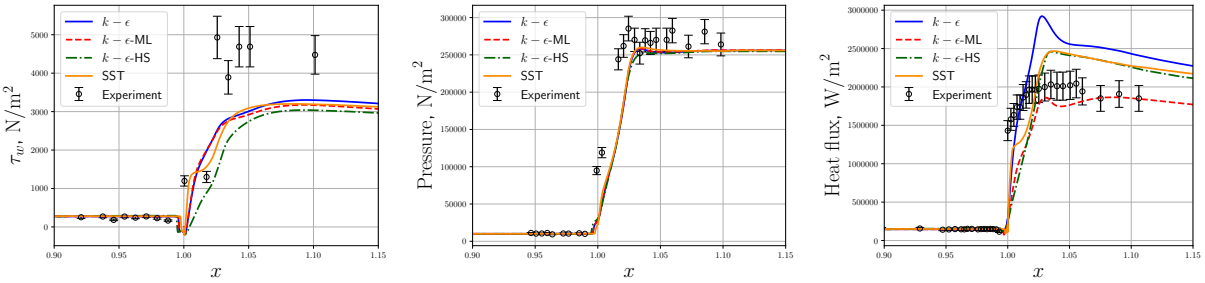
Table 5-4. Holden compression ramps. Relative errors for wall heat flux.

Model	Run 12	Run 19	Run 24
$k-\epsilon$–RC	0.327	1.308	1.711
$k-\epsilon$–RC–ML	0.302	0.451	0.585
$k-\epsilon$–CA–RC–LC	0.346	0.672	0.679
SST	0.265	0.619	0.801

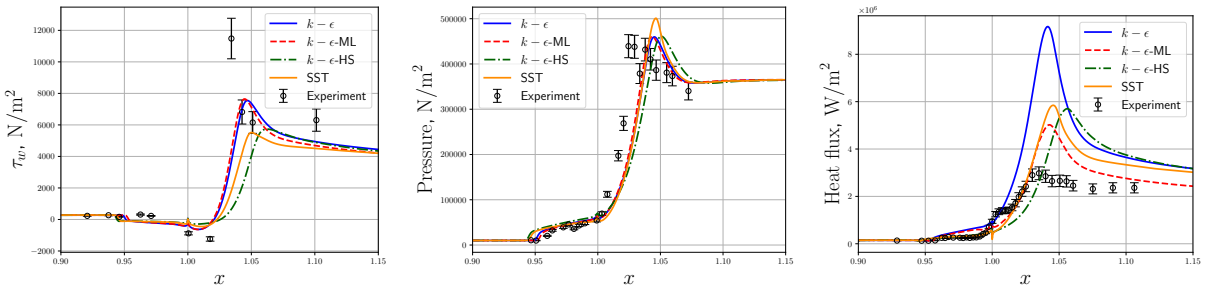
model results in substantially improved heat flux predictions. For run 12, the peak heat flux is quite similar to the experimental data, while for run 19 and run 24, which correspond to stronger shocks, wall heat flux is still over predicted but is significantly improved from the baseline model. Similar to the FI results, minimal changes are observed in the mean wall shear and wall pressure fields. The exception to this is run 24, where we find that the ML model slightly increases peak pressure and reduces the size of the separation bubble; the same trend was seen in the FI solution (Figure 5-2). Upon comparing to the $k-\epsilon$–CA–RC–LC and SST models, we find that the $k-\epsilon$–RC–ML model better captures the value and location of the peak heat flux. The length-scale limiter in the $k-\epsilon$–CA–RC–LC model results in a heat flux rise that lags behind the experimental data. This limiter additionally reduces wall shear stress and brings it further away from the experimental data. The SST model yields similar results to the $k-\epsilon$–CA–RC–LC model in terms of peak heat flux, but noticeably struggles to predict the size of the separation region for run 24. Additionally, while SST yields improved wall heat flux predictions over the baseline $k-\epsilon$ models, it notably does so at the expense of a reduced wall shear stress which is in worse agreement with the experimental data.

Figure 5-9 shows the turbulent Prandtl number predicted by the ML model. As compared to the FI results, we observe that the ML model gives a much more coherent prediction for $\mathrm{Pr_t}$ but the main trends are still present. In particular, we observe that the ML models predict a region of high $\mathrm{Pr_t}$ very near the wall, followed by a region of low $\mathrm{Pr_t}$ further away from the wall. This trend is similar to what was observed in the FI results. We additionally observe minimal impact on $\mathrm{Pr_t}$ upstream of the interaction region.

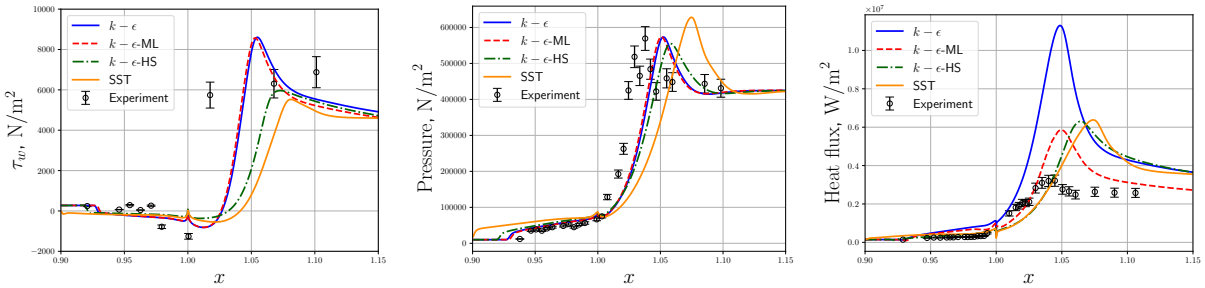
Figure 5-10 presents predicted wall quantities for the ML model on the suite of Schülein cases while Table 5-5 tabulates relative errors in the wall heat flux. Across all cases we observe that, while again not as accurate as the field inference results, the ML model systematically improves wall heat flux predictions as compared to the baseline model. In all cases, the overprediction of the peak wall heat flux is significantly mitigated. We additionally note that the variable–$\mathrm{Pr_t}$ model has a negligible impact on wall shear stress and wall pressure. As compared to the $k-\epsilon$–CA–RC–LC model, we again observe that the ML model results in substantially better predictions for the wall heat flux for all cases. The $k-\epsilon$–RC–ML model yields better results than the SST model on all but the 14° case, where we again note that SST gives an improved prediction to wall heating at the expense of a worse prediction for wall shear. Next, Figure 5-11 shows the predicted $\mathrm{Pr_t}$ fields. We observe that all models predict a reduced turbulent Prandtl number near and downstream of



(a) Wall shear stress (run 12). **(b) Wall pressure (run 12).** **(c) Wall heat flux (run 12).**

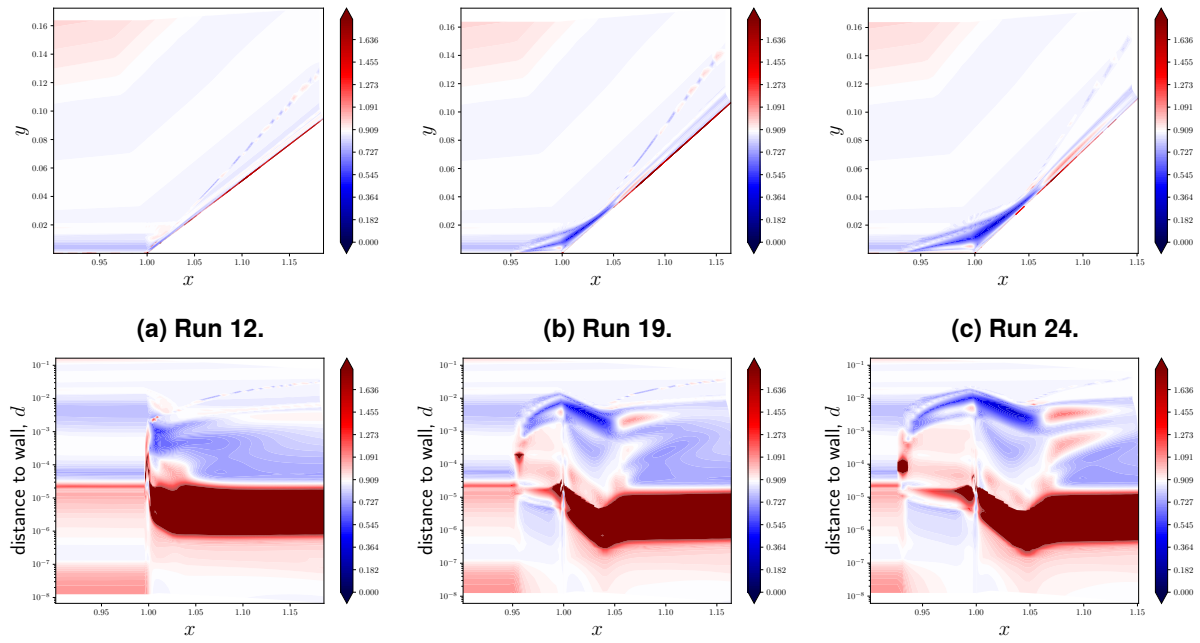


(d) Wall shear stress (run 19). **(e) Wall pressure (run 19).** **(f) Wall heat flux (run 19).**



(g) Wall shear stress (run 24). **(h) Wall pressure (run 24).** **(i) Wall heat flux (run 24).**

Figure 5-8. Holden compression ramps. Predictions for Holden run 12 (top), run 19 (middle), and run 24 (bottom) compression cases.



(a) Run 12. **(b) Run 19.** **(c) Run 24.**

(d) Run 12 (log scale). **(e) Run 19 (log scale).** **(f) Run 24 (log scale).**

Figure 5-9. Holden compression ramps. Turbulent Prandtl number predicted by the ML model. The top row shows results in a linear scale while the bottom row shows results in a log-y scale using the distance to wall as the vertical axis.

Table 5-5. Schülein SBLIs. Relative errors for wall heat flux.

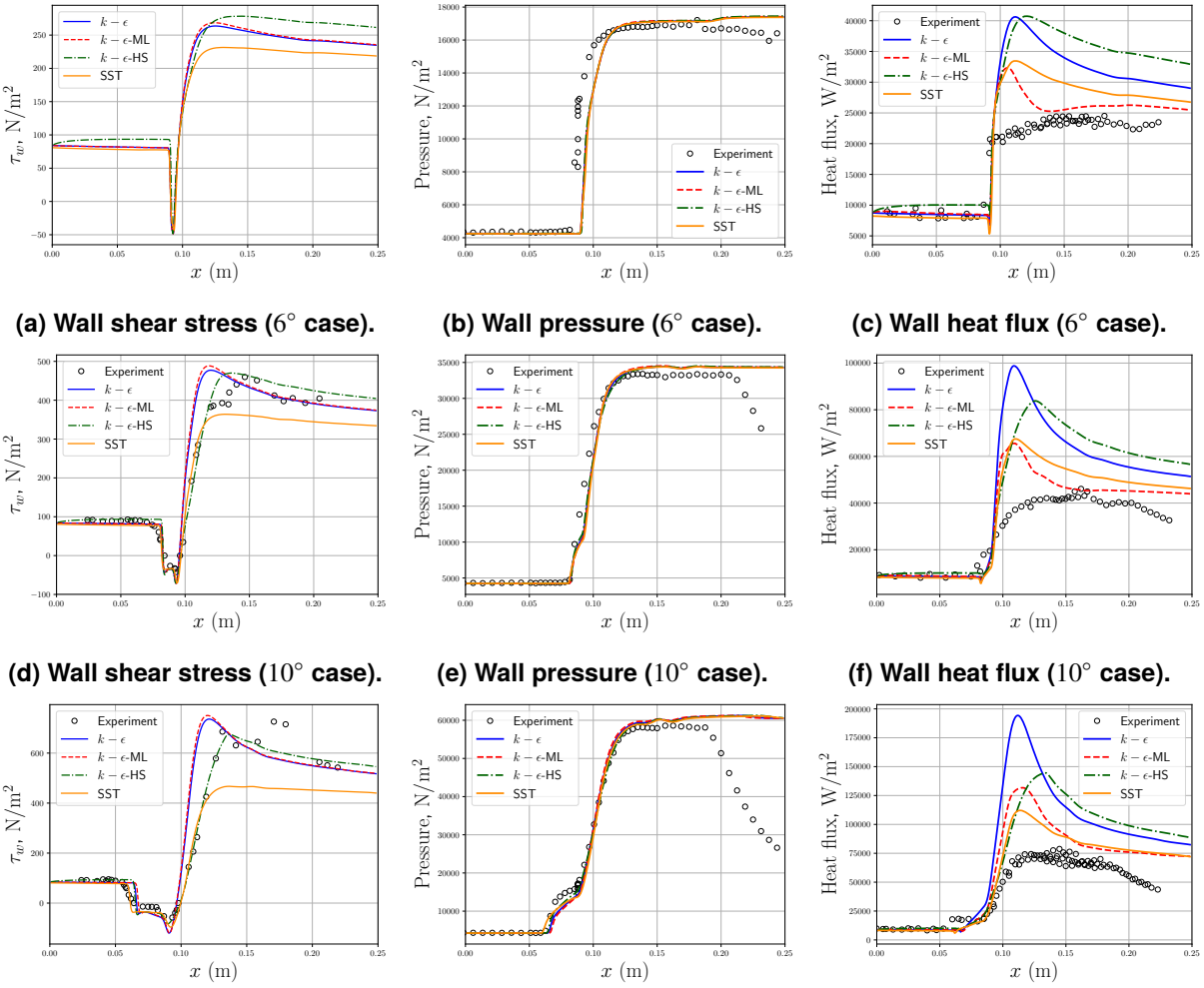
Model	6° case	10° case	14° case
$k-\epsilon$–RC	0.495	0.783	0.971
$k-\epsilon$–RC–ML	0.218	0.329	0.499
$k-\epsilon$–CA–RC–LC	0.597	0.728	0.784
SST	0.510	0.399	0.38

the interaction region. Unlike the inference results, we additionally observe a coherent region of a slightly higher turbulent Prandtl number near the wall.

5.6.3. Predictions on unseen test cases: Hollow cylinder flare and HIFiRE ground test

The previous subsection demonstrated that we were able to learn a variable–Pr_t model that improves wall heating predictions for the cases on which field inference was performed. We now assess the performance of the model on unseen testing cases. As test cases, we consider runs 13 and 18 of the CUBRC hollow cylinder flare (HCF) and runs 30 and 46 of the HIFiRE test geometry. The HCF run 13 corresponds to a Mach 6 axisymmetric hollow cylinder flare with a 36° ramp. The run 18 case employs the same geometry, but is at Mach 7 with a lower Reynolds number. The HIFiRE cases correspond to approximately Mach 7 flow with an expansion and a terminal 33° flare. Run 46 has a higher Reynolds number than run 30. Experimental data for wall quantities are available for both cases from CUBRC. Both cases include a transition from laminar to turbulent flow. The transition location is determined from the experimental data and we manually enforce a trip by deactivating the turbulence model production term upstream of this location. Schematics of the problem geometries for the two cases are shown in Figure 5-12.

Figure 5-13 shows results of the ML-embedded RANS model for HCF runs 13 and 18, while Table 5-6 tabulates the errors in wall heat flux. We highlight that run 13 has a lower Mach number and higher Reynolds number than run 18, and that the baseline models perform worse for the high Reynolds number cases. We observe that, despite being a testing case, the ML model results in improved wall heat flux predictions. For both run 13 and 18, the peak heat flux is decreased and is in better agreement with experimental data. For both cases there is minimal change to wall pressure. Wall shear stress is not shown because experimental data are not available. In comparison to the $k-\epsilon$–CA–RC–LC and SST models, we observe that the $k-\epsilon$–RC–ML model results in improved predictions for peak heat flux as well as for the location of peak heat flux. We additionally highlight that this case displays physics not present in the training cases in that it (1) is an axisymmetric case and (2) undergoes a transition from laminar to turbulent flow. The ML model appears to robustly handle these new physics. Figure 5-14 shows contours of Pr_t predicted by the ML model for run

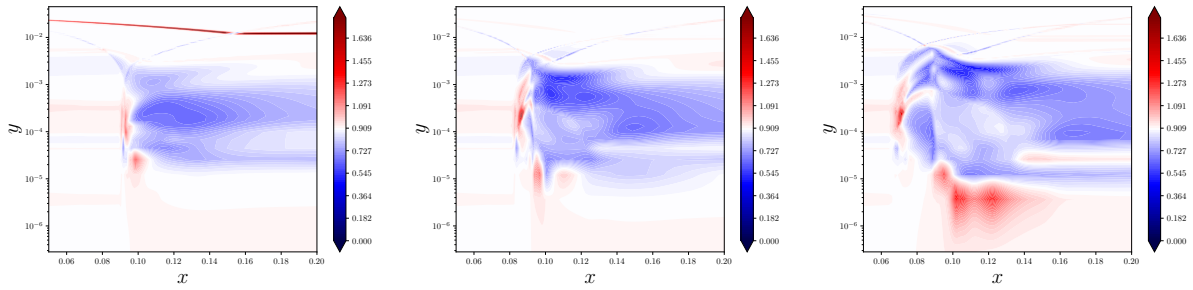


(a) Wall shear stress (6° case). **(b) Wall pressure (6° case).** **(c) Wall heat flux (6° case).**

(d) Wall shear stress (10° case). **(e) Wall pressure (10° case).** **(f) Wall heat flux (10° case).**

(g) Wall shear stress (14° case). **(h) Wall Pressure (14° case).** **(i) Wall heat flux (14° case).**

Figure 5-10. Schülein SBLIs. Predictions for wall QoIs for the 6° (top), 10° (middle), and 14° (bottom) cases.



(a) $6°$. **(b)** $10°$. **(c)** $14°$.

Figure 5-11. Schülein SBLIs. ML predictions for the turbulent Prandtl number for the 6, 10, and $14°$ Schülein cases.

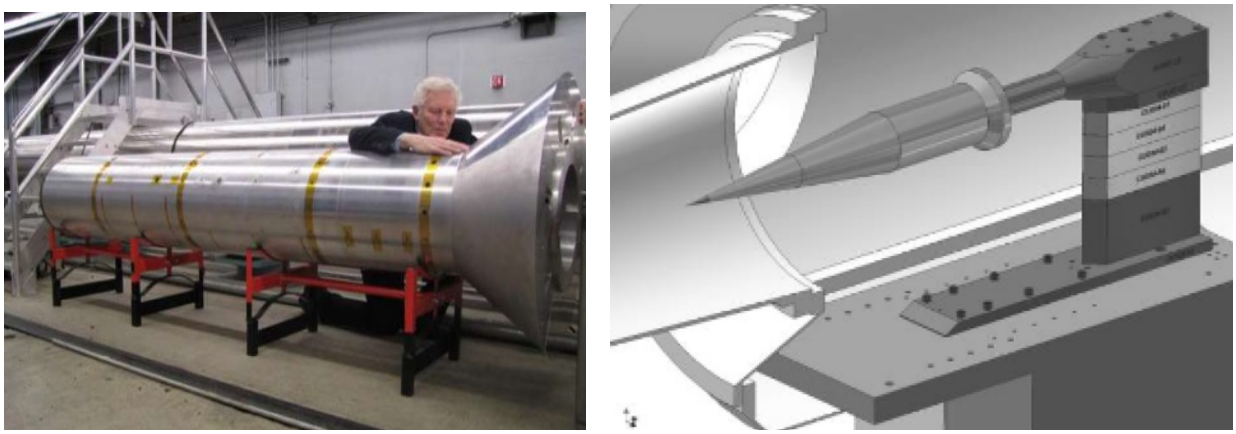
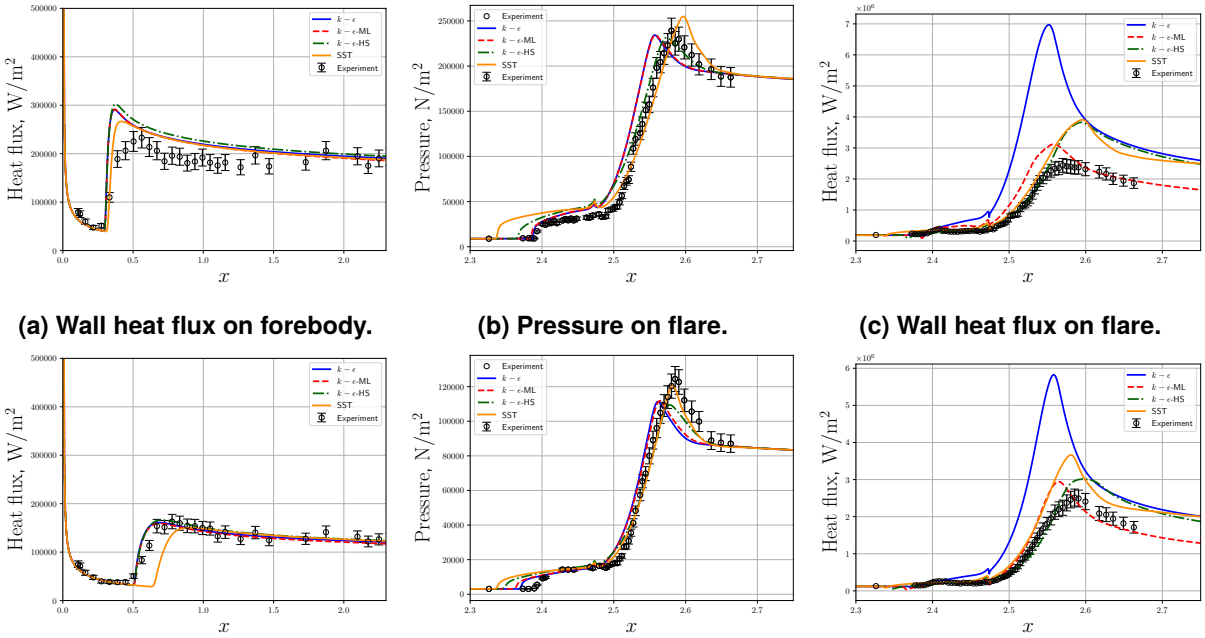


Figure 5-12. Hollow cylinder flare geometry (left) and HiFIRE ground test geometry (right). The figures are from Ref. [27] and [93], respectively.



(a) Wall heat flux on forebody. **(b) Pressure on flare.** **(c) Wall heat flux on flare.**

(d) Wall heat flux on forebody. **(e) Pressure on flare.** **(f) Wall heat flux on flare.**

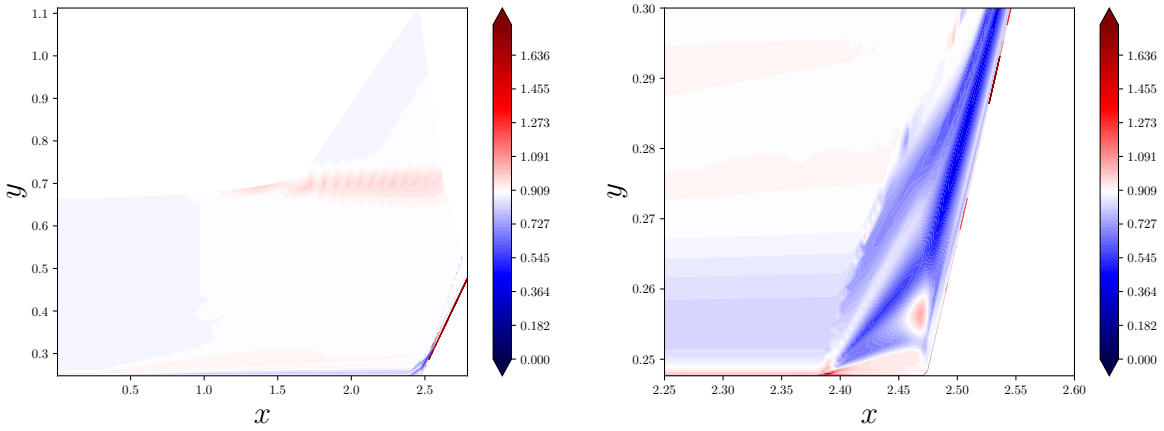
Figure 5-13. CUBRC hollow cylinder flare. Run 13 is shown on the top, while Run 18 is shown on the bottom.

13. We see that the ML model predicts a region of low $\mathrm{Pr_t}$ in the corner, but there is again a region of high $\mathrm{Pr_t}$ very near to the wall.

Figure 5-15 shows results of the ML-embedded RANS model for HIFiRE ground tests run 30 and run 46, while Table 5-7 tabulates the wall heat flux errors. As compared to the HCF cases, the HIFiRE ground tests include an expansion prior to the compression corner; these physics are not included in the training data. For both cases, we again observe that the ML-embedded model results in a decreased peak heat flux. For run 30 the $k-\epsilon-$RC$-$ML model better captures both the rise in heat flux and the peak heat flux location as compared to the baseline $k-\epsilon-$RC model. The $k-\epsilon-$RC$-$ML additionally better predicts peak heat flux and the location of peak heat flux

Table 5-6. CUBRC hollow cylinder flare. Relative errors for wall heat flux.

Model	Run 13	Run 18
$k-\epsilon-$RC	1.629	1.229
$k-\epsilon-$RC$-$ML	0.354	0.231
$k-\epsilon-$CA–RC–LC	0.411	0.213
SST	0.429	0.371



(a) $k-\epsilon$–RC–ML. **(b)** $k-\epsilon$–RC–ML **(close up of flare).**

Figure 5-14. CUBRC hollow cylinder flare. Turbulent Prandtl number predicted from ML models. The top row shows the full geometry, while the bottom row shows a close up of the flare.

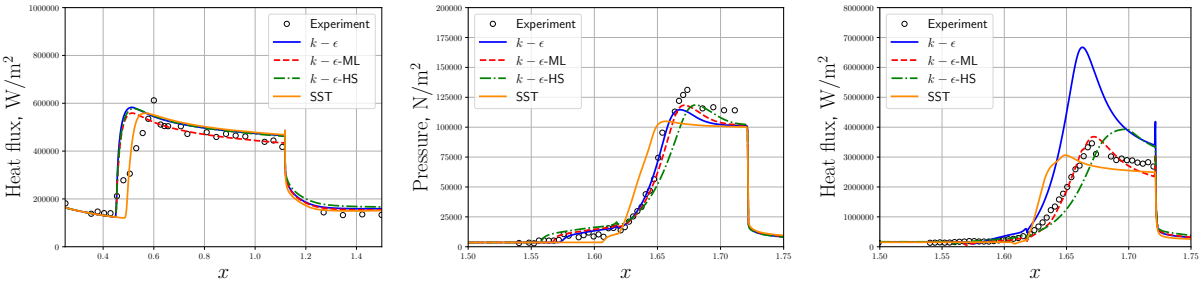
Table 5-7. HIFiRE ground tests. Relative errors for wall heat flux.

Model	Run 30	Run 46
$k-\epsilon$–RC	0.804	1.604
$k-\epsilon$–RC–ML	0.108	0.398
$k-\epsilon$–CA–RC–LC	0.285	0.577
SST	0.320	0.795

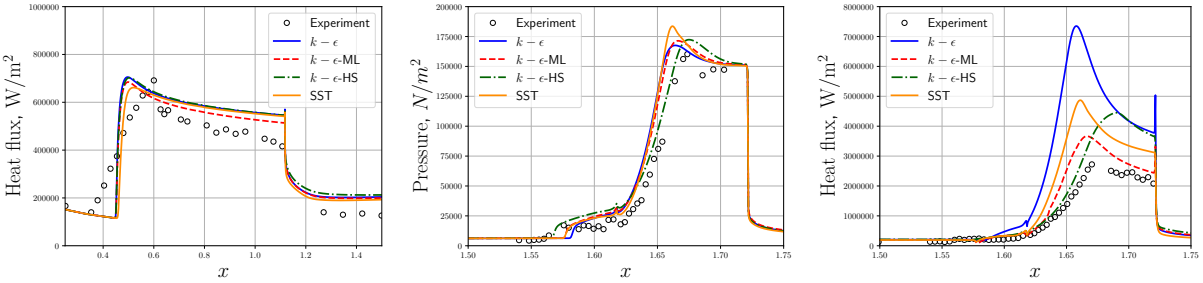
as compared to the $k-\epsilon$–CA–RC–LC and SST models. Predictions for wall heating are mostly unaffected in the upstream boundary layer, but we do observe that the $k-\epsilon$–RC–ML model yields a slightly lower wall heat flux for both cases. The same story is present in the run 46 case. At the higher Reynolds number, the baseline models severely over-predict the maximum wall heat flux. The ML model is able to significantly decrease this over-prediction while improving wall heat flux downstream of reattachment. For both cases, changes to surface pressure are small. Lastly, Figure 5-16 shows contours of the predicted Pr_t for run 30. We again observe a low region of Pr_t near the flare along with a very high value of Pr_t near the wall. We additionally observe a region of high Pr_t around the compression corner shock which is more pronounced than in the other cases.

5.7. Model interpretation

A simple interpretation of the ML-augmented variable–Pr_t model is provided through examination of the solution at the location of peak wall heat flux. In each of the cases considered, this location is downstream of both the shock wave/surface intersection, as well as boundary layer reattachment.

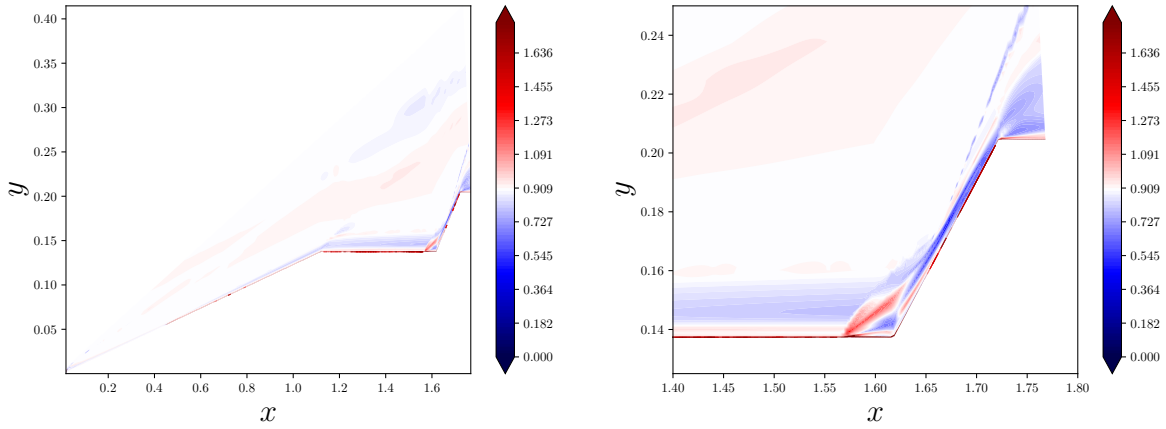


(a) Wall heat flux on forebody. **(b) Wall pressure on flare.** **(c) Wall heat flux on flare.**



(d) Wall heat flux on forebody. **(e) Wall pressure on flare.** **(f) Wall heat flux on flare.**

Figure 5-15. Predictions for HIFiRE ground test run 30 (top) and run 46 (bottom).



(a) $k-\epsilon-$RC$-$ML **(b)** $k-\epsilon-$RC$-$ML **(close up of flare)**

Figure 5-16. Turbulent Prandtl number predicted from ML models for run 30.

In this regime, energy budget analysis as well as wall heat transfer decomposition analysis identifies that the wall-normal turbulent heat flux component dominates over the stream-wise component. The wall-normal turbulent heat flux term may transfer heat either towards the wall (negative sign) or away from the wall (positive sign). Methods that decompose the wall heat flux into constitutive terms (e.g., [3]) show that the overall turbulent heat flux contribution to the wall heat flux is $\int_0^\infty \frac{1}{\gamma} q_{t_n} \mathrm{d}n$, where q_{t_n} is the wall-normal turbulent heat flux and the integral is taken in the wall normal direction n across the boundary layer profile. All else being equal, the effect of an increase in $\mathrm{Pr_t}$ is to decrease the magnitude of the turbulent heat flux, and vice-versa.

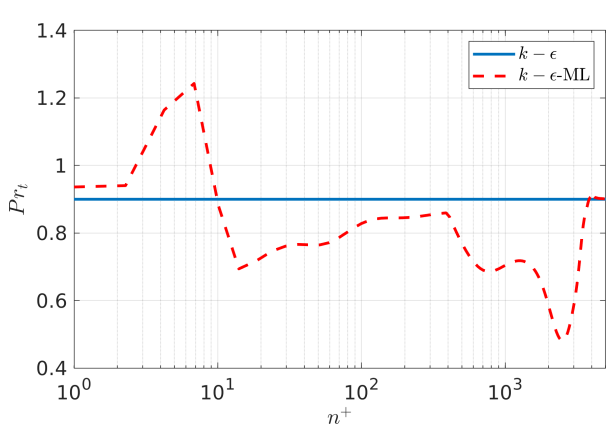
Figure 5-17 shows $\mathrm{Pr_t}$ predicted by the standard $k-\epsilon-$RC model and the $k-\epsilon-$RC$-$ML model for the 14° Schülein case (top) and run 13 of the hollow cylinder flare (bottom) at the location of peak heat flux. As outlined previously, we observe a region of increased $\mathrm{Pr_t}$ near the wall, followed by a low region of $\mathrm{Pr_t}$ further away from the wall. Upon comparing these trends to the turbulent heat flux, we observe that the ML model collectively acts to reduce heat transfer to the wall. In the near wall region where the turbulent heat flux is negative (heat moving towards the wall), the ML model predicts an increased $\mathrm{Pr_t}$. This increased $\mathrm{Pr_t}$ acts to decrease heat transfer to the wall. This near-wall region of negative wall-normal heat flux is much less pronounced for the Schülein 14° case than the HCF run 13 case, which helps explain why the inference process and resulting ML model does not predict as drastic of a rise in $\mathrm{Pr_t}$ near the wall. Similarly, in regions where the wall-normal turbulent heat flux is positive (heat moving away from the wall), the ML model predicts a decreased $\mathrm{Pr_t}$. This decreased $\mathrm{Pr_t}$ acts to increase heat transfer away from the wall.

5.8. Iterative convergence

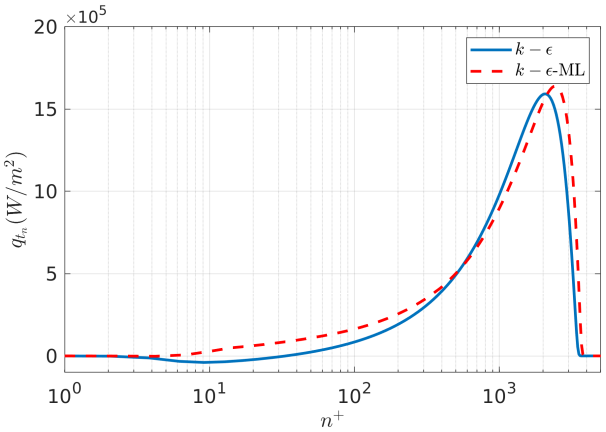
We did not notice a notable difference in residual convergence of the ML-embedded RANS solver as compared to the baseline RANS solver. To depict typical convergence behavior, Figure 5-18 shows nonlinear residual and QoI convergence for the 14° Schülein case for the $k-\epsilon-$RC and $k-\epsilon-$RC$-$ML models. The results for this case are representative of most cases considered, although we do note that residual convergence for both the baseline $k-\epsilon$ and $k-\epsilon-$RC$-$ML models was worse for the HIFiRE ground test cases. The QoI we examine is the maximum value of the heat flux. We observe that the $k-\epsilon-$RC and $k-\epsilon-$RC$-$ML models both display very similar convergence, and that inclusion of the ML model does not deteriorate convergence. This convergence is despite not including the sensitivities of the neural network in the solver Jacobians as well as only updating the turbulent Prandtl number field every fifty iterations.

5.9. Grid convergence

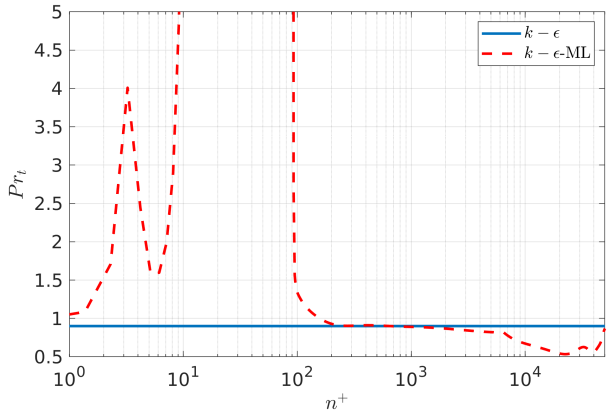
Obtaining grid converged solutions for hypersonic SBLIs is known to be difficult. To mitigate large run times, we perform our field inference on meshes that are slightly coarser than what is required



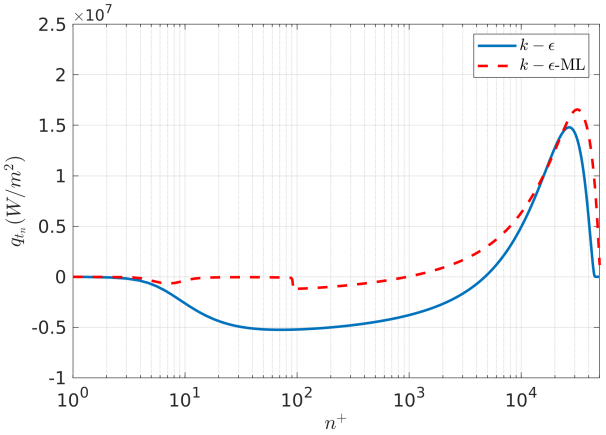
(a) Turbulent Prandtl number (Schülein, 14 ° case).



(b) Wall-normal turbulent heat flux (Schülein, 14 ° case).

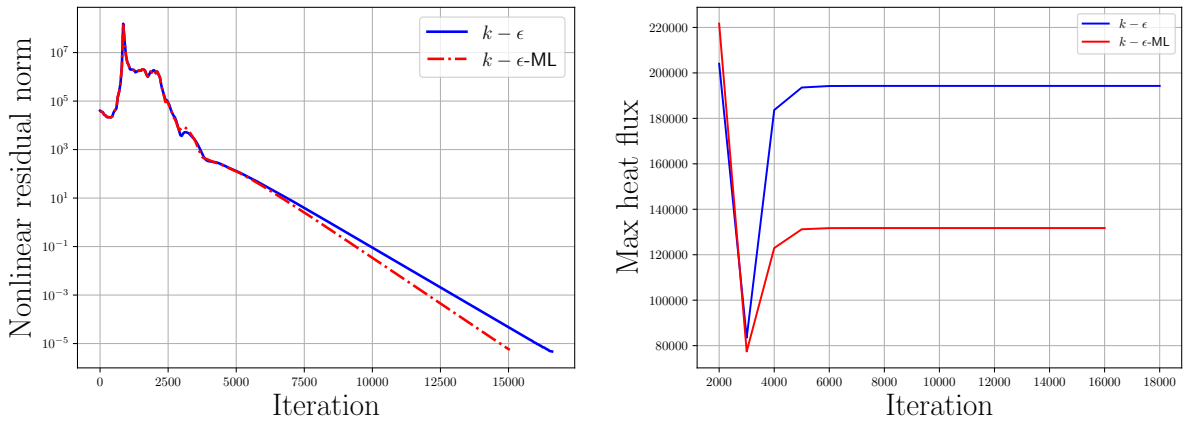


(c) Turbulent Prandtl number (HCF, run 13).



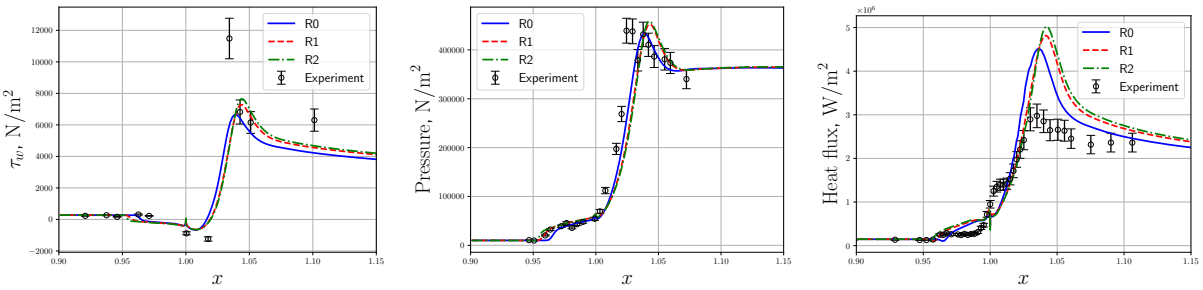
(d) Wall-normal turbulent heat flux (HCF, run 13).

Figure 5-17. Turbulent heat flux analysis. Predictions for turbulent Prandtl number (left) and wall-normal turbulent heat flux (right) at the location of peak wall heat flux. The Schülein 14° case is shown on top, while the hollow cylinder flare run 13 case is shown on the bottom. The x-axis, n^+, is the wall distance in wall units.



(a) Residual norm **(b) Maximum heat flux**

Figure 5-18. Schülein SBLIs. Residual convergence (left) and maximum heat flux (right) for the $k-\epsilon-$RC and $k-\epsilon-$RC$-$ML models.



(a) Wall shear stress. **(b) Wall pressure.** **(c) Wall heat flux.**

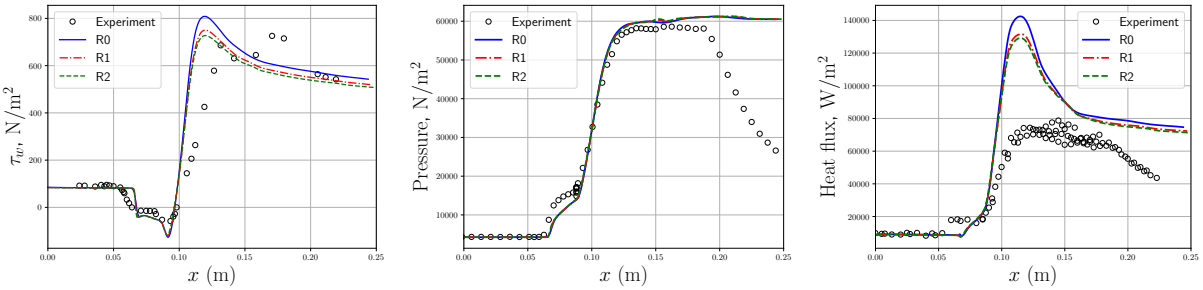
Figure 5-19. Holden compression ramps. Grid convergence for the $k-\epsilon-$RC$-$ML model. R0 is the coarsest grid while R2 is the finest.

for a grid converged solution. To quantify uncertainties due to grid resolution, this section presents grid convergence results for the $k-\epsilon-$RC$-$ML model for the Holden run 19 and Schülein 14° cases; we note that the grid convergence of the $k-\epsilon-$RC model is very similar . Figure 5-19 shows grid convergence results for Holden run 19, while Table 5-8 tabulates details on the mesh. The field inference is performed on the coarse "R0" mesh, while the ML results are shown for the fine mesh. We observe that, despite having a maximum y^+ of under 1, on the coarsest mesh the peak heat flux and peak shear stress are not grid-converged and are under-predicted by approximately 5%. The separation bubble is additionally slightly under-predicted. After one level of refinement (2x in the wall-normal direction and 2x near the compression corner in the streamwise direction), grid convergence is almost reached.

Figure 5-20 shows results grid convergence results for the Schülein 14° case, which is the most

Table 5-8. Summary of meshes for Holden run 19.

Level	n_{x_1}	n_{x_2}	max y^+ on wall
R0	327	151	0.42
R1	501	301	0.21
R2	883	601	0.11



(a) Wall shear stress. **(b) Wall pressure.** **(c) Wall heat flux.**

Figure 5-20. Schülein SBLIs. Grid convergence for the $k-\epsilon-$RC$-$ML model. R0 is the coarsest grid while R2 is the finest.

stringent incident SBLI, while Table 5-9 tabulates details on the mesh. The field inference is performed on the coarse "R0" mesh, while the ML results are shown for the "R2" mesh. We observe that all meshes produce very similar wall pressure, but wall heat flux and wall shear stress are slightly over-predicted on the coarser mesh at and downstream of reattachment. One level of refinement (2x in each direction) results in solutions that are almost identical to those obtained after 2 levels of refinement (4x in each direction).

5.10. Conclusions

We presented a field inversion and machine learning strategy to construct a variable turbulent Prandtl number model to improve wall heat flux predictions in hypersonic shock boundary layer

Table 5-9. Summary of meshes for the Schülein 14° case.

Level	n_{x_1}	n_{x_2}	max y^+
R0	251	201	0.67
R1	501	401	0.33
R2	1001	801	0.16

interactions. We performed a finite-dimensional field inference to obtain corrections to the turbulent Prandtl number with the goal of improving predictions on a suite of SBLI cases with available experimental data. We then developed a predictive model for a variable turbulent Prandtl number by learning these corrective fields as a function of local flow features. Our learning framework leveraged Lipschitz-continuous neural networks and ensembles to further improve robustness.

We trained a variable turbulent Prandtl number model to work in a manner that is consistent with a $k-\epsilon$ model with a rapid compression correction. The model was trained on relatively simple two-dimensional problems and was seen to systematically improve wall heat flux predictions across the training cases as well as on unseen three-dimensional axisymmetric test cases that are geometrically different from the training cases. Additionally, the model was shown to out-perform an additional $k-\epsilon$ model that included various existing fixes to improve predictions for hypersonic flows and the SST model. In particular, the model led to systematic improvements in the prediction of peak heat flux near the point of reattachment, which is a known deficiency of RANS models. Analysis of the resulting solutions shows that the ML-augmented variable turbulent Prandtl number has the overall effect of reducing heat transfer to the wall. In regions where the turbulent heat flux is negative (i.e., heat moving towards the wall), the ML model, in general, predicts an increased turbulent Prandtl number which decreases heat transfer to the wall. Similarly, in regions where the wall-normal turbulent heat flux is positive (heat moving away from the wall), the ML model predicts a decreased turbulent Prandtl number which increases heat transfer away from the wall.

We are careful to remark that the goal of this work was not to develop a general turbulence model. Rather, the focus was on a more pragmatic engineering objective: to develop a framework to improve existing models for wall heat transfer in hypersonic shock-boundary layer interactions and demonstrate generalizability in this class of problems. As future work we will extend the work to include the development of a variable turbulent Prandtl number for an SST model appropriate for hypersonic flows along with the validation of field data (e.g., temperature fields) against high-fidelity datasets. At the present time, direct numerical simulation data of hypersonic SBLIs, in particular, compression ramps, is extremely limited. As these datasets become more available, we will assess the performance of our variable turbulent Prandtl number models for predicting field quantities. Future work will further focus on a fully coupled inference process using, e.g., the LIFE framework [87] such that the inferred fields are guaranteed to be learnable.

This page intentionally left blank.

6. FUTURE DIRECTIONS

This chapter highlights ongoing work that was started during this LDRD and is recommended as future work. We focus on deployment to more complex exemplars, the development of a model that joinly uses TBNNs for improved Reynolds stress predictions and a variable turbulent Prandtl number model for improved wall heat flux predictions, fully coupled model training using the Ensemble Kalman Filter, and fully coupled model training using the LIFE framework.

6.1. Deployment of models on fully 3D exemplars and extended validation set

Both the developed TBNNs and variable turbulent Prandtl number model were only tested on 2D and axisymmetric bodies. As future work, we will continue to deploy the developed models on increasingly complex geometries and expand the validation test suite. Here, we describe the start of such an effort for deploying the variable turbulent Prandtl number model learned in Chapter 5 on Sandia's cone-slice-wedge.

6.1.1. Cone-slice-wedge Predictions

Preliminary simulation results were obtained for a three-dimensional cone-slice-wedge (CSW) conceptual vehicle shape [60] using the variable$-\mathrm{Pr_t}$ model. The CSW model was tested in Sandia's Hypersonic Wind Tunnel. Here, we compare only predictions given by different model variants and leave model validation to future studies. The overall geometry and flow-field topology are shown in Figure 6-1. The flow is at zero angle of attack and a free stream Mach number of 8. The boundary layer transitions ahead of the cone/slice expansion; the turbulent boundary layer then encounters the wedge-shaped control surface, setting up a three-dimensional compression corner SBLI. The flow was calculated using the $k-\epsilon$, $k-\epsilon-$RC, and $k-\epsilon-$ML models at three different flow conditions described in Table 6-1. The nominal Mach 8 wind tunnel condition is at a relatively low Reynolds number and a moderately cold wall. To test the sensitivity of predictions to model variants at different conditions, we also considered a lower wall temperature "cold-wall" case, as well as a high-Reynolds number condition.

Figure 6-2 shows predictions of the wall heat flux along the vehicle symmetry plane for the three model variants at the three flow conditions. For the HWT condition, the $k-\epsilon-$RC model reduces the peak heat flux relative to the $k-\epsilon$ model, while the ML model provides no further change to the

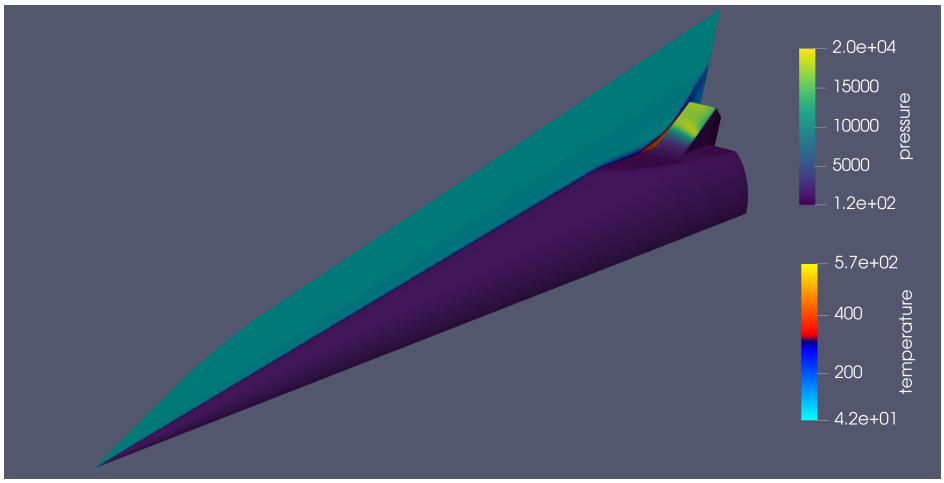
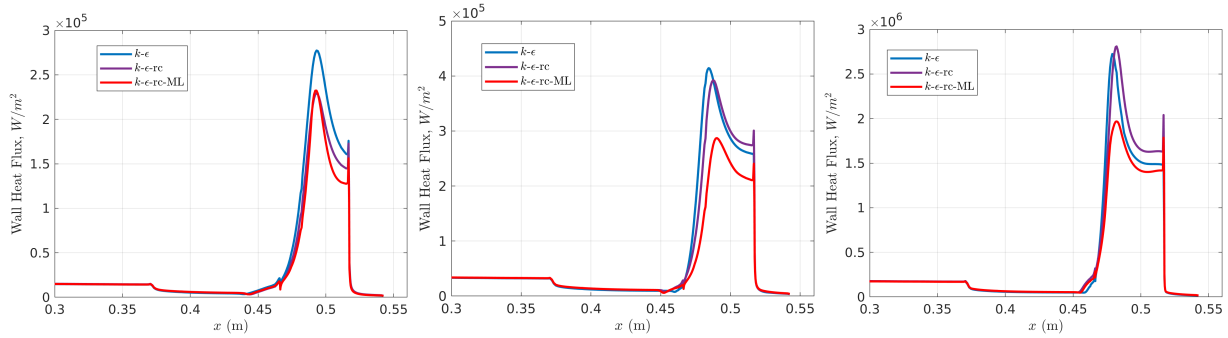


Figure 6-1. Cone-slice-wedge k-ϵ solution at the HWT flow condition. Surface contours of pressure with symmetry plane contours of temperature.

Table 6-1. Flow conditions for the cone-slice-wedge model tests.

Case	M_∞	Re_L	T_w/T_r
HWT Mach 8 Condition	8.1	9.2×10^6	0.56
Low Wall Temperature Condition	8.1	9.2×10^6	0.20
High Reynolds Number Condition	8.1	177×10^6	0.56



(a) HWT flow condition. **(b) Low T_w condition.** **(c) High Re number condition.**

Figure 6-2. Cone-slice-wedge model predictions.

peak heat flux. At the low temperature and high Reynolds number conditions, the RC correction has only small impact on the peak heat flux while the ML model provides a substantial decrease in the peak heat flux below that of the other two model variants. These results indicate the possibility that the ML model mainly activates for relatively cold walls and higher Mach numbers, which might be a result of the training data and process being weighted more towards these conditions. Future work will explore this issue, in conjunction with comparison to experimental measurements.

6.2. Joint Reynolds stress and Variable $\mathrm{Pr_t}$ results

The results presented to this point have only utilized either an improved model for the Reynolds stress via TBNNs or a variable turbulent Prandtl number. The joint development of these models poses an additional challenge. Specifically, these models should be jointly trained to be as consistent as possible. In this section, however, we briefly assess the joint performance of a $k-\epsilon$ model utilizing the TBNN developed in Chapter 4 and the variable turbulent Prandtl number model developed in Chapter 5. We embed these models within a $k-\epsilon$ model with the CA correction and, motivated by the observation that the results in Chapter 4 indicated slightly to small of a separation region, we employ a rapid compression correction with $C_\epsilon = -1.5$.

Figure 6-3 shows predictions of the $k-\epsilon$ model with both the TBNN and variable$-\mathrm{Pr_t}$ correction for the Schülein 14° SBLI, the HCF run 13, and HIFiRE run 46. Figure 6-4 presents the QoI errors. The other models considered in Chapter 3 are additionally presented for reference. We observe that, despite the inconsistencies in how the Reynolds stress and variable$-\mathrm{Pr_t}$ models were trained, the resulting model is able to give state-of-the-art predictions across the three cases. Peak shear stress, peak pressure, and peak heat flux are well captured in all cases. We highlight the decreased peak heat flux for the Schülein 14° case, which the variable$-\mathrm{Pr_t}$ model alone struggled to predict. Close examination of the ML-enhanced model, however, highlights some oscillations on the HIFiRE test

case, and in general we observed the joint model to be more subject to oscillations. The results, however, are overall promising.

6.3. Fully-consistent training

Chapter 5 developed a variable turbulent Prandtl number model using the two-step FIML approach. The advantage of such an approach is that it is relatively affordable and it is easy to add more data to the model. However, the two-step FIML approach has the disadvantage that the inferred fields might not be learnable. An alternative approach is to embed an ML model within the RANS solver and directly infer the model parameters. This process directly results in a learned model which should be more predictive than one obtained via the two-step approach. However, there are several difficulties with a direct training approach:

1. Just like field inference, the resulting optimization problem is a high-dimensional, PDE-constrained optimization problem. The most efficient solution approach is adjoint-based optimization, which is difficult for legacy codes without adjoint capabilities.

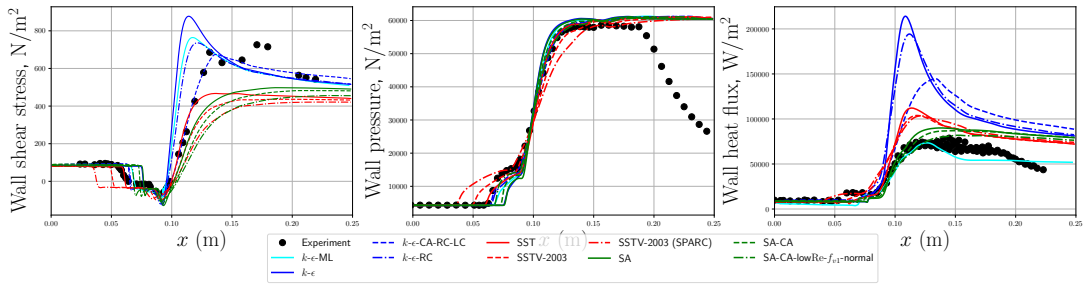
2. Training a neural network via PDE-constrained optimization is largely unexplored. Methods which have been proven necessary for training accurate neural networks, e.g., mini-batching, regularization, stochastic optimization, are not thoroughly developed for PDE-constrained training.

3. Direct training through PDE-constrained optimization is extremely expensive. As an example, in Chapter 5, we trained a neural network model for a turbulent Prandtl number using six training cases for 1000 epochs. Assuming the same number of epochs are required, directly training the network via PDE-constrained optimization would take 6000 forward PDE solves and 6000 adjoint solves to compute sensitivities. This process would need to be repeated for any change to the model configuration, e.g., new training data, new features, etc.

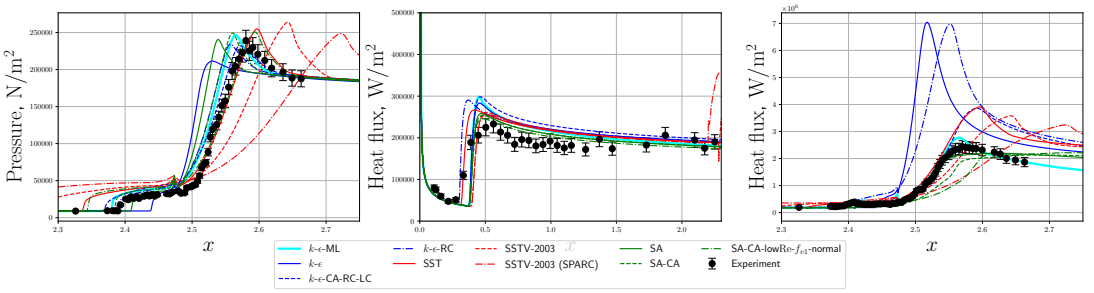
In this work, these challenges precluded the development of a mature neural network variable turbulent Prandtl number trained via a direct approach. However, we did perform an initial exploration of several direct training methods, and these are outlined in this chapter. First, Section 6.4 explores directly training an variable turbulent Prandtl number model via the Ensemble Kalman Filter. Next, Section 6.5 explores training a similar model using the LIFE framework.

6.4. Fully consistent training with the Ensemble Kalman Filter

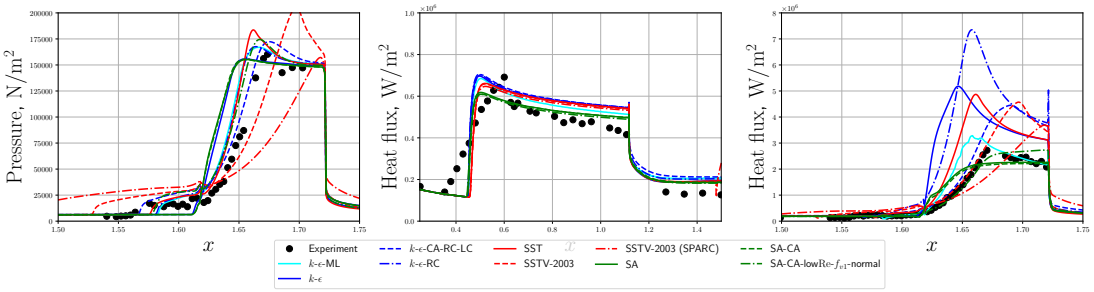
We first consider a variable turbulent Prandtl number trained via the Ensemble Kalman Filter (EnNF). The EnKF is a data assimilation method used for estimating a state variable from noisy data, and it can also be used for model inversion through iterating an ensemble of states with the



(a) Schülein 14° SBLI.

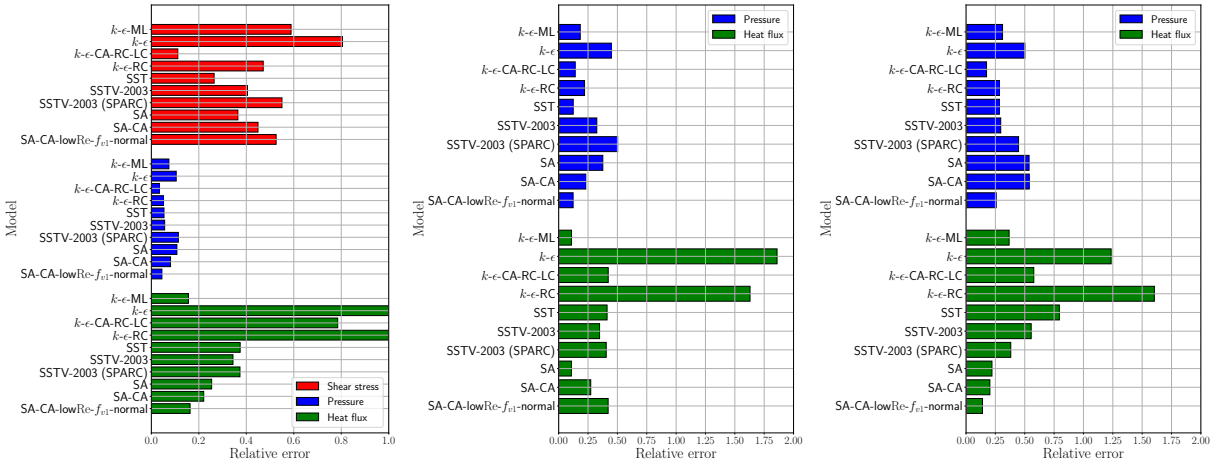


(b) HCF Run 13



(c) HIFiRE Run 46.

Figure 6-3. RANS model predictions for various validation cases.



(a) Schülein 14° SBLIs. **(b) HCF Run 13** **(c) HIFiRE Run 46**

Figure 6-4. Relative errors for wall QoIs.

Kalman Filter algorithm [37]. EnKF only requires forward PDE solves, and as such does not require an adjoint capability.

A variety of EnKF algorithms have been developed that use different cost functions and iteration methods. The algorithm utilized in this work is the EnKF with adaptive stepping, developed in [47] and utilized for data-driven turbulence modeling in [106]. EnKF with adaptive stepping uses approximate gradients based on the Kalman filter and iterative updates based on the Levenberg-Marquardt algorithm.

We describe the EnKF within the context of training a neural network model. Let $\mathcal{M} : \mathbf{w} \mapsto \mathcal{M}(\mathbf{w})$ denote the mapping from the neural network weights $\mathbf{w} \in \mathbb{R}^{N_\mathbf{w}}$ to a QoI (or set of QoIs) of interest. The EnKF optimizes the model weights $\mathbf{w}$ by iteratively minimizing the objective function

$$\mathcal{J} = \|\mathbf{d} - \mathcal{M}(\mathbf{w}^l)\|^2_{\gamma\mathbf{R}} + \|\mathbf{w}^{l+1} - \mathbf{w}^l\|^2_{\mathbf{P}} \tag{6.4.1}$$

where l is the iteration index, $\mathbf{d} \in \mathbb{R}^{n_{\text{data}}}$ is the data vector, $\gamma \in \mathbb{R}^+$ is a scaling parameter, and $\mathbf{P} \in \mathbb{R}^{N_\mathbf{w} \times N_\mathbf{w}}$ and $\mathbf{R} \in \mathbb{R}^{n_d \times n_d}$ are the model and observation covariance matrices. The full algorithm is given in Algorithm 4.

The data $\mathbf{d}$ can consist of experimental data, high-fidelity simulation data, or a combination of them, as one of the advantages of EnKF and adjoint-based methods is their ability to work with disparate data. In this work the data consists of a combination of skin friction, surface pressure, and heat flux measurements.

Algorithm 4 EnKF with adaptive stepping algorithm

$\varepsilon_{best} = \infty$

Initialize N_s samples of neural network weights $\mathbf{w}_j^0 \in \mathbb{R}^{N_\mathbf{w}}$, $j = 1, ..., N_s$ using He initialization [26].

while $l <= N_{\text{iter-max}}$ **do**

Compute the terms

$$\overline{\mathbf{w}}^l = \frac{1}{N_s} \sum_{j=1}^{N_s} \mathbf{w}_j^l \tag{6.4.2}$$

$$\mathbf{S}_\mathbf{w}^l = \frac{1}{\sqrt{N_s - 1}} \left[\mathbf{w}_1^l - \overline{\mathbf{w}}^l, \mathbf{w}_2^l - \overline{\mathbf{w}}^l, ..., \mathbf{w}_{N_s}^l - \overline{\mathbf{w}}^l \right] \tag{6.4.3}$$

$$\mathbf{S}_\mathbf{d}^l = \frac{1}{\sqrt{N_s - 1}} \left[\mathcal{M}(\mathbf{w}_1^l) - \mathcal{M}(\overline{\mathbf{w}}^l), \mathcal{M}(\mathbf{w}_2^l) - \mathcal{M}(\overline{\mathbf{w}}^l), \ldots, \mathcal{M}(\mathbf{w}_{N_s}^l) - \mathcal{M}(\overline{\mathbf{w}}^l) \right] \tag{6.4.4}$$

The Kalman gain matrix is defined as:

$$\mathbf{K} = \mathbf{S}_\mathbf{w} \mathbf{S}_\mathbf{d}^T \left(\mathbf{S}_\mathbf{d} \mathbf{S}_\mathbf{d}^T + \gamma^l \mathbf{R} \right)^{-1} \tag{6.4.5}$$

where $\mathbf{R}$ is the observation error covariance matrix and γ^l is the adaptive step, which will be defined below. The neural network weights are updated with:

$$\mathbf{w}_j^{l+1} = \mathbf{w}_j^l + \mathbf{K} \left(\mathbf{d}_j^l - \mathcal{M}(\mathbf{w}_j^l) \right) \tag{6.4.6}$$

where $\mathbf{d}_j^l$ is the data vector $\mathbf{d}$ perturbed with $\mathbf{d}_j^l = \mathbf{d} + \mathcal{N}(0, \mathbf{R})$, where $\mathcal{N}(0, \mathbf{R})$ is a normal distribution with zero mean and covariance $\mathbf{R}$.

Compute error

$$\epsilon^{l+1} = \|\mathbf{d} - \mathcal{M}(\mathbf{w}^l)\|_{\gamma\mathbf{R}}^2 \tag{6.4.7}$$

for $j = 1, ..., N_s$ **do**

if $\|\mathbf{d} - \mathcal{M}(\mathbf{w}_j^l)\| < \varepsilon_{best}$ **then**

$\varepsilon_{best} = \|\mathbf{d} - \mathcal{M}(\mathbf{w}_j^l)\|$

$\mathbf{w}_{best} = \mathbf{w}_j^l$

end if

end for

if $\epsilon^{l+1} < \epsilon^l$ **then**

$\gamma = 0.8\gamma$

else

$\gamma = 1.6\gamma$

end if

end while

6.4.0.1. EnKF trained variable turbulent Prandtl number model

We consider the EnKF to train a neural network variable turbulent Prandtl number model. Motivated from Chapter 5, we consider a neural network formulated to predict the log of the inverse of the turbulent Prandlt number

$$\mathrm{Pr_t} = \frac{1}{\exp(y^{\mathrm{ML}})} \tag{6.4.8}$$

where y^{ML} is the neural network output. This ensures that the corrections to the turbulent Prandlt number are positive and weights the data so high values of Prandlt number are accurately predicted.

The features used to predict the turbulent Prandtl number are taken to be a simplified version of the features selected in Chapter 5,

$$\boldsymbol{\eta} = \left[\mathrm{Tr}(S^{*2}) \quad \mathrm{Tr}(\Omega^{*2}) \quad \log(\mu_t/\mu) \quad max(d\frac{\sqrt{k}}{50\nu}, 2) \quad ||\nabla\tilde{T}||_2 \right]^T,$$

where the first two features are invariants of the nondimensionalized strain and rotation tensor, the third is the log of the turbulent viscosity ratio, the fourth is the wall distance Reynolds number, and the fifth is the norm of the temperature gradient. We note that the fifth feature is not non-dimensional.

6.4.1. Two-step EnKF training

The EnKF algorithm described above could be used to directly learn a neural network model leveraging datasets spanning multiple cases. Such a process, however, is expensive and difficult to scale. For example, if one added a new training dataset, the entire EnKF would have to be re-run. Further, EnKF-based training of ML techniques equipped with UQ estimates, such as those described in Chapter 4, is unclear. Due to these challenges, we pursue a two-step training process here. In the two-step process, the EnKF algorithm is first deployed individually on each training case. After learning a neural network model (and associated turbulent Prandtl number field) for each case, these data are then aggregated into one large training set, and a neural network is directly learned that maps from features to the turbulent Prandtl number fields. While this process gives up on strong consistency between the ML model and the RANS solver, it is more scalable. We note that the two-step training presented in the previous chapter used field inference aimed at optimizing the Prandtl number field such that the model matched the data, and then trained a neural network ensemble to make predictions. In contrast, the method presented here aims to optimize a number of neural network models for the Prandtl number, then trains an ensemble from that data. The main difference is that the method presented here ensures that each inferred Prandtl number field after EnKF is learnable from the features for that dataset.

Table 6-2. Summary of cases for EnKF. Blue cases are training cases while red cases are testing cases.

Case	M_∞	Re_L	Shock/ramp angle	T_w/T_0
CUBRC 2D Compression Corner Run 19	8.1	139.3×10^6	33°	0.29
Schülein impinging shock	5.0	18.5×10^6	10°	0.73
HIFiRE-1 Run 30	7.2	16.9×10^6	33°	0.13
CUBRC hollow cylinder flare run 13	6.01	16.24×10^6	36°	0.20

6.4.1.1. Deep ensembles

Similar to Chapters 4 and 5, we utilize an ensemble prediction for the final neural network model. The final neural network formulation follows closely that described in Chapter 5, but additionally includes a UQ estimate as described in Chapter 4. Full details are given in the Appendix.

6.4.2. Results

We now present results for an EnKF-trained variable turbulent Prandtl number model. The model is trained on two cases: the 14° Schülein impinging shock and run 19 of the 2D compression corner. The model is tested on HIFiRE run 30 and run 13 of the hollow cylinder flare. Table 6-2 summarizes the cases. The baseline RANS model used for all simulations is the $k-\epsilon-$RC model as described in Chapter 3.

6.4.2.1. EnKF training

Each EnKF realization uses $N_s = 20$ samples and is run until for 50 iterations or until 8 iterations without a decrease in the cost function, defined as

$$\mathcal{L} = \sum_{j=1}^{N_s} \|\mathcal{M}(\mathbf{w}^l) - \mathbf{d}\|^2_{\gamma\mathbf{R}}. \tag{6.4.9}$$

The data covariance matrix $\mathbf{R}$ is chosen to be a diagonal matrix with elements

$$\mathbf{R}_{i,i} = \frac{\mathrm{abs}\,(\mathbf{d}_i)}{10}.$$

The data vector $\mathbf{d}$ is a combination of the wall stress, wall pressure, and wall heat flux. Each quantity is first scaled by the mean values of each before concatenating the values into a vector $\mathbf{d}$. Each EnKF algorithm trains a neural network with 3 hidden layers and 10 nodes per hidden layer.

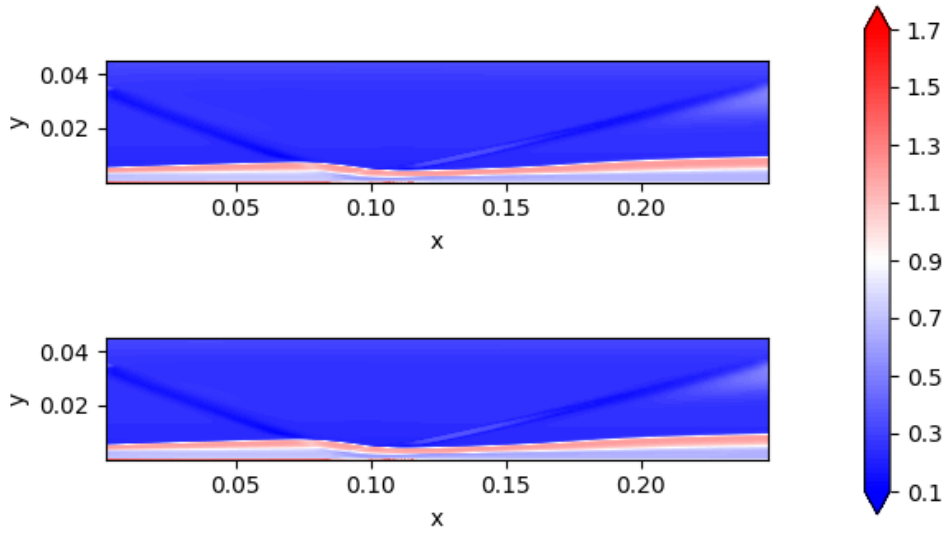


Figure 6-5. Turbulent Prandtl number fields computed from two distinct neural networks trained independently with EnKF for the Schülein 10° case.

The Schülein and CUBRC 2D compression corner cases are used as the training cases and are trained separately using EnKF to compute the neural network weights. Each case was trained twice to examine how much the inferred turbulent Prandtl number is dependent on the random initialization of the algorithm. Figures 6-5 and 6-6 show the turbulent Prandtl number fields found using the neural networks on the Schülein and CUBRC 2D compression corner cases, respectively. For both cases the two realizations produce qualitatively similar results, though differences are more visible for the CUBRC 2D compression corner. The near-wall behavior is probably most important to predicting the wall quantities, so we show the turbulent Prandtl number in regions close to the wall in Figure 6-7 for both cases. There is a thin layer of high turbulent Prandtl number very close to the wall in both cases. Above that, the CUBRC 2D compression corner has very low turbulent Prandtl number while the Schülein case has a region of turbulent Prandlt number around 0.7 with another high turbulent Prandtl number region above that.

Figures 6-8 and 6-9 show the Stanton number, pressure, and skin friction resulting from the EnKF algorithm compared to a standard $k-\epsilon$, standard model with compressibility correction $k-\epsilon-$RC, and the experimental data. Unsurprisingly, the wall quantities are predicted more accurately with EnKF, but for the Schülein 10° case EnKF fails to predict the Stanton number profile near the shock crossing point. This indicates that either EnKF is not converged, the neural network does not contain enough parameters to predict the wall heat flux profile accurately, or the difference is due to model form error from a turbulent Prandtl number assumption.

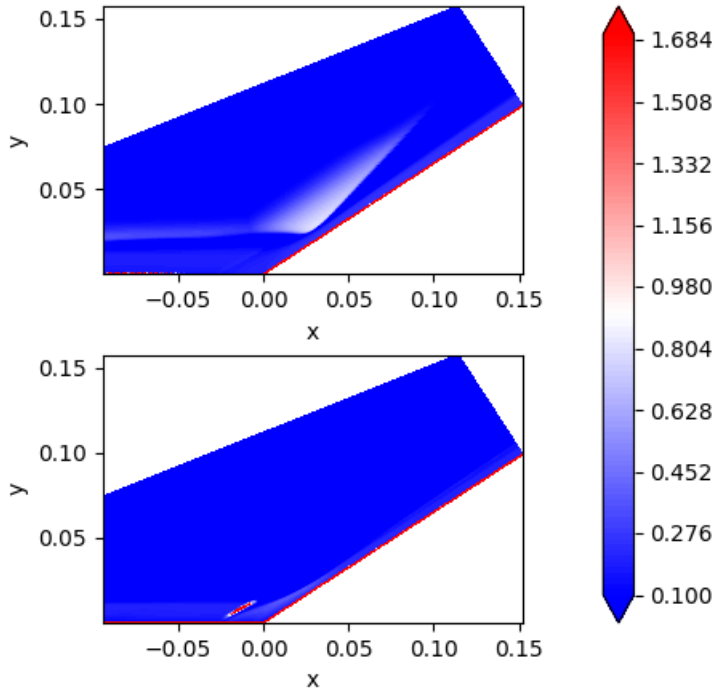


Figure 6-6. Turbulent Prandtl number fields computed from two distinct neural networks trained independently with EnKF for the CUBRC 2D compression corner.

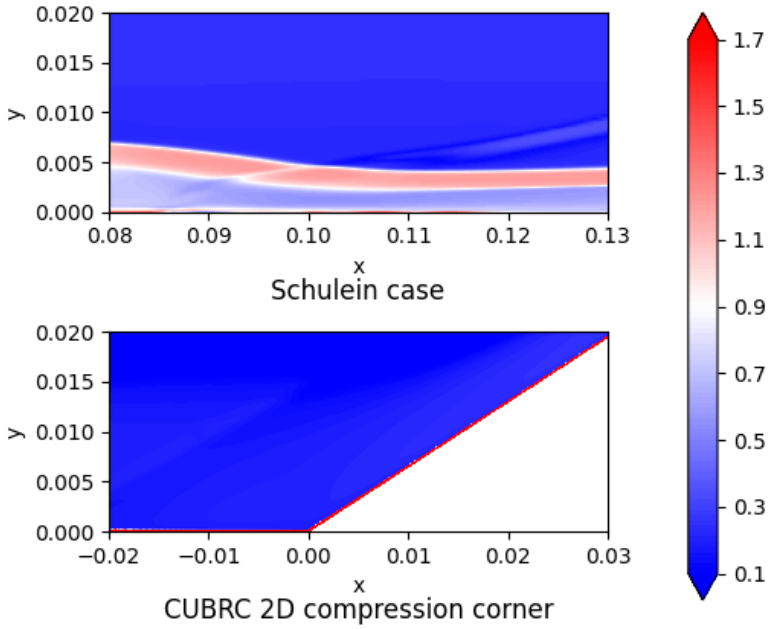
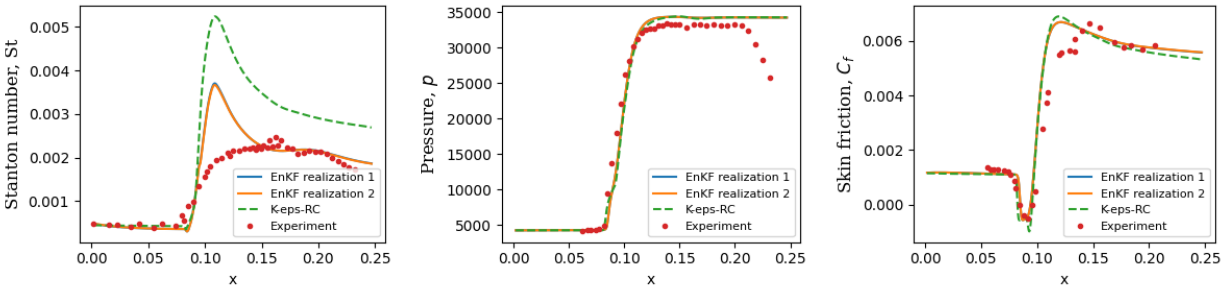
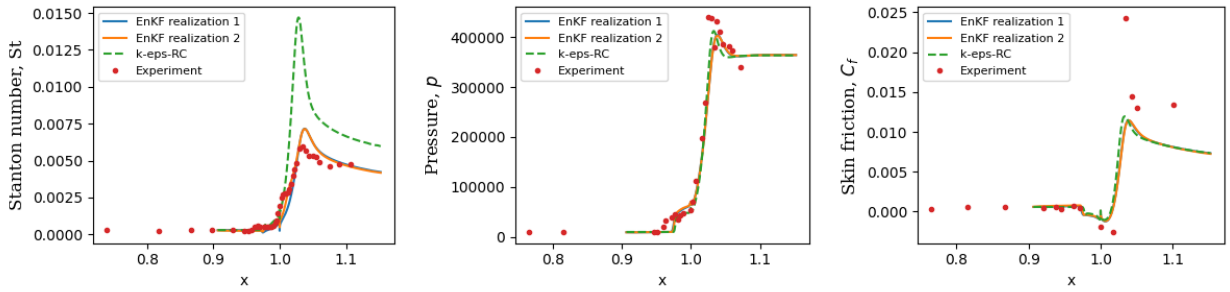


Figure 6-7. Turbulent Prandtl number fields near shock boundary layer interactions for Schülein 10° case (top) and CUBRC 2D compression corner (bottom).



(a) Stanton number **(b) Pressure** **(c) Skin friction**

Figure 6-8. Wall quantities from EnKF for Schülein 10° case compared to baseline models with and without a compressibility correction and experimental data.



(a) Stanton number **(b) Pressure** **(c) Skin friction**

Figure 6-9. Wall quantities from EnKF for CUBRC 2D compression corner case compared to baseline models with and without a compressibility correction and experimental data.

Table 6-3. Ensemble training sets.

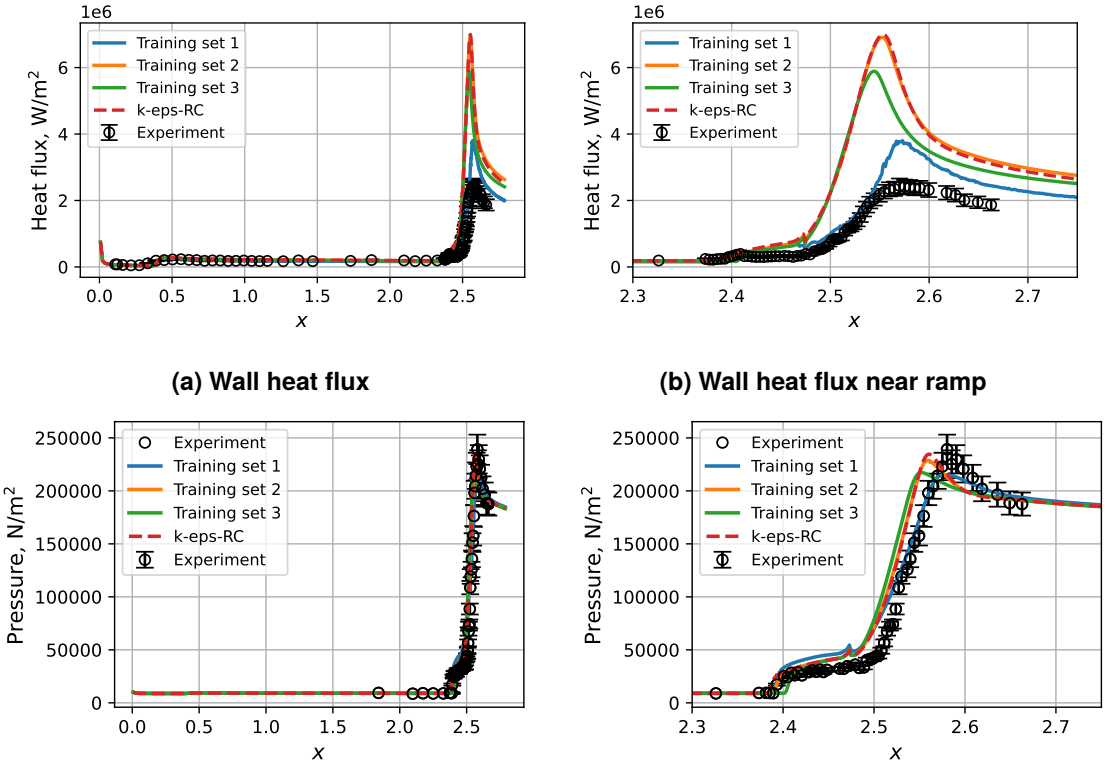
Training set #	Number of CUBRC 2D compression corner case	Number of Schülein 10° cases	Epistemic scaling coefficient
1	2	0	33.
2	0	2	21.
3	2	2	17.

6.4.2.2. Ensemble training

Neural network ensembles are trained to predict the turbulent Prandtl number from the converged EnKF cases. We set up ensembles using three different sets of training data, specified in Table 6-3. The test cases are the HIFiRE-1 run 30 and the CUBRC hollow cylinder flare run 13. Each ensemble contains 10 neural networks with 4 hidden layers and 10 nodes per hidden layer. The neural networks are trained for 500 epochs, at which point the neural networks are converged. The prior is set to $\sigma^2_{\text{prior}} = 1$. The epistemic scaling coefficient $\hat{C}$ varies for each training set and is given in Table 6-3.

Figure 6-10 shows the wall heat flux and wall pressure from the CUBRC hollow cylinder flare case for each ML-RANS model as well as the baseline $k-\epsilon-$RC model. Both wall heat flux and wall pressure are accurately predicted by all the models from the inlet until close to the ramp. However, all the models over-predict the heat flux significantly near the point of reattachment. The ML-RANS models with training set 1 and training set 3, which contain compression ramp data, lower the peak heat flux in comparison with the baseline model, but they still have large heat flux errors.

Fig. 6-11 shows the wall heat flux and wall pressure on the HIFiRE-1. The ML-RANS models using training sets 2 and 3 bring the heat flux down slightly and closer to the experimental data, while the heat flux profile for training set 1 differs from the experiment by having an initial rise



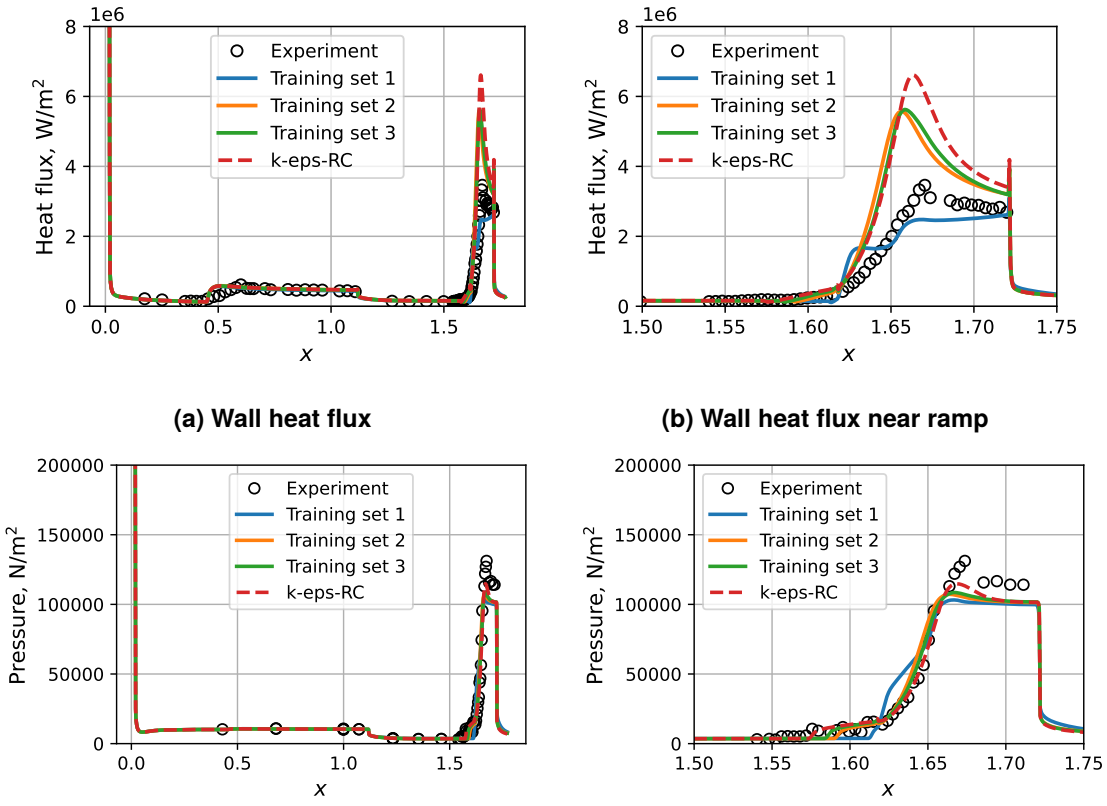
(a) Wall heat flux

(b) Wall heat flux near ramp

(c) Wall pressure

(d) Wall pressure near ramp

Figure 6-10. Wall heat flux and pressure from CUBRC hollow cylinder flare run 13 compared to a baseline $k-\varepsilon$ models with and without compressibility corrections and experimental data.



(a) Wall heat flux

(b) Wall heat flux near ramp

(c) Wall pressure

(d) Wall pressure near ramp

Figure 6-11. Wall heat flux and pressure from HIFiRE-1 run 30 compared to baseline $k-\varepsilon$ models with and without compressibility corrections and experimental data.

near the ramp, then levels off before rising again. Since the solutions with training set 2 and 3 are almost identical, this suggests that the feature set used here cannot adequately capture the physics of the CUBRC 2D compression ramp well enough to transfer information about the Prandtl number to the HIFiRE-1.

6.4.3. Discussion and future work

While the method presented here generally improves heat flux and wall pressure on the test cases, the improvement is only marginal. However, it is encouraging that the ML-RANS models all give reasonable predictions and are generally closer to the experimental data despite being trained on only one or two distinct cases.

There are a large number of directions that future works could go to improve on the results shown here. Some of them are:

- Improving the EnKF algorithm. A number of improvements were explored in Ref. [41] including adding momentum, additional sample variance, expanding ensembles, and variable step size.
- Improving the feature set. We explored utilizing the more detailed feature set given in Chapter 5, but did not have sufficient time for a detailed study.
- Implementing a tensor basis neural network to predict the turbulent heat flux term instead of a turbulent Prandtl number approximation.
- Introducing additional sample variance during EnKF so that the converged ensemble can be used to approximate aleatoric uncertainty.
- Training on a larger number of datasets.

6.5. Development of a variable turbulent Prandtl number using the LIFE framework

Lastly, we consider training a variable turbulent Prandtl number using the LIFE framework opposed to FIML. In the FIML approach, a field inference problem is first solved for the model discrepancy (see Eq. (5.3.1)). This optimization problem is solved for a variety of training cases. Then, one examines field quantities that can be obtained from the simulations, uses them to create features $\boldsymbol{\eta}$ that correlate well with the learned discrepancy fields, and then learns the discrepancy fields as a function of the features $\boldsymbol{\eta}$. Typically, a neural network is used to predict the inferred field β in terms of the features [62, 81].

While this method has had some success, it has some notable drawbacks. First, since the field inference (FI) problems for each training case are completely decoupled, there is no guarantee that the inferred β-fields for different cases behave consistently in the feature-space with respect to each other. Therefore, it might not be possible to learn a model that can accurately predict all of them. Even if this is not the case, the inferred discrepancy fields may not be learnable in terms of the chosen set of features or any possible set of features since FI is not performed with consideration of the augmented model's structure. Thus, there is no guarantee that FIML will yield an adequate model augmentation.

To address these concerns, Holland et al. proposed an approach referred to as integrated inference and machine learning (IIML). Here, the augmentation model is integrated directly into the solver before the learning process begins, and the two-step approach of FIML is replaced by simply directly solving an optimization problem for the model parameters for all of the cases simultaneously. By eliminating the intermediate step, this approach implicitly considers the model's functional form during the learning process, guaranteeing that the inferred discrepancy fields can be calculated by the augmentation model. Furthermore, by coupling the individual cases during the learning process, it ensures that the augmentation is consistent across all of them [31].

These benefits come at a cost, however. In addition to the training-related challenges outlined in Section 6.3, IIML requires developing features and the augmentation function before performing any inference on the data. This may not always be be practical without performing an additional FI step beforehand to learn about how field quantities and potential features behave. Furthermore, while simple IIML does guarantee a learnable model, it does not have any safeguards to ensure that the model is robust and generalizable [87].

Srivastava and Duraisamy proposed the Learning and Inference Assisted by Feature-space Engineering (LIFE) framework to address these issues [87]. It extends the concepts of IIML by providing a set of guiding principles as well as an implementation of them to create robust, generalizable feature-based model augmentations trained on sparse data sets.

One of their key contributions was the idea of localized learning. Localized learning refers to learning techniques that only modify the model in the vicinity of available training data, leaving regions of the feature-space without data untouched. This ensures that, if the augmented model is applied to a case outside of the training regime, at worst, it will behave as badly as the baseline model. Since the baseline model is still a physically realizable solution, it prevents the spurious behavior that could result from simple implementations of nonlinear augmentation functions, such as a neural network [87]. For a more detailed description of the tenets of LIFE, we refer the reader to [87].

6.5.1. Learning feature-based data-driven model augmentations

To describe LIFE, we outline the introduction of a feature-based augmentation model into the equations. This is similar to the approach deployed in Chapter 5, but here the augmentation is implicitly defined as a function of features (instead of spatial coordinate). Mathematically, for a steady-state solver, this can be expressed by

$$\mathcal{R}_m(\tilde{\mathbf{u}}_m, \boldsymbol{\xi}) = 0 \quad \rightarrow \quad \mathcal{R}_{m,aug}(\tilde{\mathbf{u}}_m; \beta(\boldsymbol{\eta}(\tilde{\mathbf{u}}_m), \mathbf{w}), \boldsymbol{\xi}) = 0 \tag{6.5.1}$$

where R_m represents the discretized system of steady-state equations we attempt to find a solution for, $\tilde{\mathbf{u}}_m$ the model states that we solve for, and $\boldsymbol{\xi}$ the boundary conditions and geometry that characterize a specific problem or case. We inject the augmentation model that uses the states $\tilde{\mathbf{u}}_m$ to calculate values for features $\boldsymbol{\eta}$, which are the inputs to the model that calculates β, a discrepancy term calculated for each point in the computational domain. The goal of the learning process is to determine the values for the weights $\mathbf{w}$ that result in the best performing model.

We assess the performance of both the baseline model and the augmented models using a cost function $\mathcal{J}$. While it is explicitly just a function of $\tilde{\mathbf{u}}_m$, it is also implicitly a function of the augmentation $\beta\left(\boldsymbol{\eta}\left(\tilde{\mathbf{u}}_m\right), \mathbf{w}\right)$ since the augmentation will affect the obtained solution. Thus, determining the best performing model can be expressed as the optimization problem

$$\min_{\mathbf{w}} \mathcal{J}(\tilde{\mathbf{u}}_m; \beta(\boldsymbol{\eta}(\tilde{\mathbf{u}}_m), \mathbf{w})) \quad \text{s.t.} \quad \mathcal{R}_{m,aug}(\tilde{\mathbf{u}}_m; \beta(\boldsymbol{\eta}(\tilde{\mathbf{u}}_m), \mathbf{w}), \boldsymbol{\xi}) = 0. \tag{6.5.2}$$

6.5.1.1. The discrete adjoint method

Techniques for solving optimization problems of the form given in Eq. (6.5.2) can be classified into one of two categories – gradient-free or gradient-based. Here, the gradient refers to the derivative of the cost function with respect to the model parameters. Generally, gradient-based methods are significantly more efficient than their counterparts; this is particularly true when the optimization variable is high-dimensional. Despite the cost of calculating the gradients, such methods can much more efficiently traverse the optimization space and require fewer forward evaluations of the model, especially when the number of input parameters is large. A particularly effective method for computing gradients is the discrete adjoint method, which can be described as follows.

Given our cost function $\mathcal{J}(\tilde{\mathbf{u}}_m; \boldsymbol{\beta}(\boldsymbol{\eta}(\tilde{\mathbf{u}}_m), \mathbf{w}))$, its total derivative with respect to the model parameters $\mathbf{w}$ is given by

$$\frac{d\mathcal{J}}{d\mathbf{w}} = \frac{d\mathcal{J}}{d\boldsymbol{\beta}} \frac{\partial \boldsymbol{\beta}}{\partial \mathbf{w}} = \left(\frac{\partial \mathcal{J}}{\partial \boldsymbol{\beta}} + \frac{\partial \mathcal{J}}{\partial \tilde{\mathbf{u}}_m} \frac{d\tilde{\mathbf{u}}_m}{d\boldsymbol{\beta}} \right) \frac{\partial \boldsymbol{\beta}}{\partial \mathbf{w}}. \tag{6.5.3}$$

This formula requires the challenging computation of the total derivative $\frac{d\tilde{\mathbf{u}}_m}{d\boldsymbol{\beta}}$. To avoid this, we can notice that no matter what values for the input parameters we use, we constrain the governing equations to always be satisfied. Therefore, we require that

$$\mathcal{R}_{m,aug}(\tilde{\mathbf{u}}_m; \boldsymbol{\beta}(\boldsymbol{\eta}(\tilde{\mathbf{u}}_m), \mathbf{w}), \boldsymbol{\xi}) = 0 \tag{6.5.4}$$

always, which means that its derivative is defined by and must satisfy

$$\frac{d\mathcal{R}_{m,aug}}{d\boldsymbol{\beta}} = \frac{\partial \mathcal{R}_{m,aug}}{\partial \boldsymbol{\beta}} + \frac{\partial \mathcal{R}_{m,aug}}{\partial \tilde{\mathbf{u}}_m} \frac{d\tilde{\mathbf{u}}_m}{d\boldsymbol{\beta}} = 0. \tag{6.5.5}$$

Solving Eq. (6.5.5) for $\frac{d\tilde{\mathbf{u}}_m}{d\boldsymbol{\beta}}$ and substituting into Eq. (6.5.5) yields

$$\frac{d\mathcal{J}}{d\mathbf{w}} = \left(\frac{\partial \mathcal{J}}{\partial \boldsymbol{\beta}} - \frac{\partial \mathcal{J}}{\partial \tilde{\mathbf{u}}_m} \left[\frac{\partial \mathcal{R}_{m,aug}}{\partial \tilde{\mathbf{u}}_m} \right]^{-1} \frac{\partial \mathcal{R}_{m,aug}}{\partial \boldsymbol{\beta}} \right) \frac{\partial \boldsymbol{\beta}}{\partial \mathbf{w}}. \tag{6.5.6}$$

Each of the partial derivatives in this equation can be calculated directly using backpropagation or algorithmic differentiation techniques. However, this formula still requires a performing a

matrix inversion for the triple-matrix product. To avoid this, either the product $\frac{\partial \mathcal{J}}{\partial \tilde{\mathbf{u}}_m}\left[\frac{\partial \mathcal{R}_{m,aug}}{\partial \tilde{\mathbf{u}}_m}\right]^{-1}$ or $\left[\frac{\partial \mathcal{R}_{m,aug}}{\partial \tilde{\mathbf{u}}_m}\right]^{-1}\frac{\partial \mathcal{R}_{m,aug}}{\partial \boldsymbol{\beta}}$ can be computed separately to result in a simpler two-matrix product. Since the objective function $\mathcal{J}$ is a scalar and input parameters $\mathbf{w}$ are a vector, it is cheaper to calculate the former. This product is referred to as the adjoint vector and is the solution $\boldsymbol{\psi}^T$ of the adjoint system that corresponds to the forward problem, given by

$$\boldsymbol{\psi}^T \frac{\partial \mathcal{R}_{m,aug}}{\partial \tilde{\mathbf{u}}_m} = -\frac{\partial \mathcal{J}}{\partial \tilde{\mathbf{u}}_m}. \tag{6.5.7}$$

Therefore, the desired sensitivities are obtained by calculating

$$\frac{d\mathcal{J}}{d\mathbf{w}} = \left(\frac{\partial \mathcal{J}}{\partial \boldsymbol{\beta}} + \boldsymbol{\psi}^T \frac{\partial \mathcal{R}_{m,aug}}{\partial \boldsymbol{\beta}}\right)\frac{\partial \boldsymbol{\beta}}{\partial \mathbf{w}} \tag{6.5.8}$$

where $\boldsymbol{\psi}^T$ is obtained by solving Eq. (6.5.7).

6.5.2. *Application of LIFE for improved modeling of wall heating in hypersonic SBLIs.*

We assess LIFE for improving the post-interaction wall heat transfer predictions of RANS simulations for hypersonic boundary layers experiencing shock interactions by learning a variable turbulent Prandtl number augmentation. This work was performed at the University of Michigan and leveraged an in-house, second order-in-space, unstructured, finite volume code. Forward solves were performed using a fully coupled implicit Euler approach. For adjoint solves, the code employs the ADOL-C automatic differentiation package to calculate the necessary Jacobian terms and used the MUMPS parallel sparse direct linear solver library to solve the adjoint system.

6.5.2.1. Baseline model and augmentation

For our turbulence model, we use the standard formulation of the Wilcox 2006 $k-\omega$ turbulence model given by

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho u_j k)}{\partial x_j} = P - \beta^* \rho k\omega + \frac{\partial}{\partial x_j}\left[\left(\mu + \sigma_k \frac{\rho k}{\omega}\right)\frac{\partial k}{\partial x_j}\right] \tag{6.5.9a}$$

$$\frac{\partial(\rho \omega)}{\partial t} + \frac{\partial(\rho u_j \omega)}{\partial x_j} = \frac{\gamma\omega}{k}P - \beta\rho\omega^2 + \frac{\partial}{\partial x_j}\left[\left(\mu + \sigma_\omega \frac{\rho k}{\omega}\right)\frac{\partial \omega}{\partial x_j}\right] + \frac{\rho\sigma_d}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j} \tag{6.5.9b}$$

A description of the model and the values for the closure coefficients β, β^* (not the discrepancy field),$\gamma, \sigma_k, \sigma_\omega, \sigma_d$ are given in Ref. [100].

Our variable turbulent Prandtl number augmentation is given by

$$\mathrm{Pr}_{t,0} \quad \rightarrow \quad e^{\beta}\mathrm{Pr}_{t,0} = 0.9e^{\beta}, \tag{6.5.10}$$

where the nominal constant value for the turbulent Prandtl number is $\mathrm{Pr}_{t,0} = 0.9$. This specific mathematical form of the augmentation was chosen for two reasons. First, it made the optimization problem an unconstrained one, which allowed for simpler optimization approaches to be used. Second, since Pr_t appears in the denominator of its corresponding source term in the RANS energy equation, using e^{β} rather than just β prevented numerical conditioning issues whenever the optimization drove $\mathrm{Pr}_t \rightarrow 0$. These findings were confirmed through preliminary FI studies.

6.5.2.2. Full Learning Problem

To quantify how well a RANS model performs with respect to our modeling goal, we used the objective function

$$\mathcal{J} = \frac{1}{n_x}\sum_{i=1}^{n_x}\left(q_{w,m}(x_i) - q_{w,ref}(x_i)\right)^2 \times 10^{-12}, \tag{6.5.11}$$

which is the mean-squared error of the wall heat transfer (q_w) predictions by the model with respect to the reference data at n_x wall faces in the computational domain. The factor of 10^{-12} converts the heat transfer measurements to $\mathrm{MW/m^2}$ from $\mathrm{W/m^2}$ to improve the numerical conditioning of the objective function and sensitivities calculated by the adjoint simulations.

To enforce our learning goal, the reference data was constructed using experimental data from the studies described in Table 6-4 as follows. First, a simulation of the flow using the baseline turbulence model was computed[1] and the wall heat transfer distributions were extracted. These distributions were then compared to the experimental data. The reference data was chosen to be the baseline simulation results up until the point where the baseline simulation and experimental data diverge. Since our goal was to learn a model focused solely on improving post-interaction heat transfer predictions, this approach prevented undesired changes within the zero-pressure-gradient boundary layer upstream of the shock. To prevent overfitting to noise and measurement error, and to ignore the effect of other phenomena present in the data such as expansion waves, the reference data downstream of this point was a piecewise regression of the experimental results that capture the trend we expected to see. Transitions between the baseline RANS and experimental regressions, as

[1]The freestream boundary conditions for these simulations account for the effect of the leading edge shock to obtain the desired freestream/edge flow conditions recorded in the studies.

Table 6-4. Reference data used

Case	Study	Geometry	M_∞	Angle (°)
S6	Schülein (2006) [78]	Oblique incident shock wave	5.00	6
S10				10
S14				14
N026	Nicholson et al. (2024) [57]	Curved forward-facing wall	4.87	≈ 28.4

well as the individual regressions used cubic splines to ensure smoothness and reduce the potential for numerical artifacts that might arise when trying to fit to cusps in the reference data. Using this approach, reference data values are generated for each wall face in the computational domain, allowing for a straightforward, consistent calculation of the objective function.

Thus, the full optimization problem can be stated as

$$\min_{\mathbf{w}} \mathcal{J} \quad \text{s.t.} \quad \mathcal{R}_{m,aug}(\tilde{\mathbf{u}}_m; \boldsymbol{\beta}(\boldsymbol{\eta}(\tilde{\mathbf{u}}_m), \mathbf{w}), \boldsymbol{\xi}) = 0. \tag{6.5.12}$$

Here, the cost function $\mathcal{J}$ is given by Eq. (6.5.11) and the weights $\mathbf{w}$ are the augmented model parameters. No regularization was included in the cost function as it was found to be unnecessary during preliminary FI studies. Furthermore, the case-coupling approach of LIFE and its reduction of dimension in the optimization adds an implicit regularization to the problem [87].

6.5.2.3. Simplified Learning Problem

The above inverse problem is complex and not easy to find an adequate solution for. Hence, a simpler learning problem was required to better understand the influence of data-driven variable turbulent Prandtl number models on RANS simulations for hypersonic SBLIs and characterize the problem. For this purpose, we also solved a simpler inference problem. For this, the cost function is given by

$$\mathcal{J} = \frac{1}{n_x} \sum_{i=1}^{n_x} \left(q_{w,m}(x_i) - q_{w,KC}(x_i) \right)^2 \times 10^{-12}, \tag{6.5.13}$$

which is the mean-squared error of the wall heat transfer predictions by the model with respect to a RANS simulation implementing the baseline turbulence model (Eq. (6.5.9)) and the variable turbulence model of Kays and Crawford [40], which can be stated as

$$\mathrm{Pr_t} = \frac{1}{0.5882 + 0.228\,(\mu_t/\mu) - 0.0441\,(\mu_t/\mu)^2 \left[1 - \exp\left(\frac{-5.165}{\mu_t/\mu}\right)\right]}. \tag{6.5.14}$$

If one divides Eq. (6.5.14) by the nominal Prandtl number value $\mathrm{Pr}_{t,0}$, then the equation can be interpreted as a variable turbulent Prandtl number augmentation to the baseline turbulence model with constant Pr_t. Furthermore, it would be an augmentation with only a single nondimensional input feature of μ_t/μ. Therefore, if we state the learning problem for the LIFE framework as

$$\min_{\mathbf{w}} \mathcal{J} \quad \text{s.t.} \quad \mathcal{R}_{m,aug}\left(\tilde{\mathbf{u}}_m; \boldsymbol{\beta}\left(\eta\left(\mu_t/\mu\right), \mathbf{w}\right), \boldsymbol{\xi}\right) = 0 \tag{6.5.15}$$

where $\mathcal{J}$ is given by Eq. (6.5.13), then one local minimum, and hopefully the global minimum, should be the LIFE model weights that recreate the model given by Eq. (6.5.14) as closely as possible.

Since this inference problem had a clear, desired, single-feature solution, it served as an excellent proxy to characterize the numerics of the full learning problem. If a model trained by LIFE was unable to adequately replicate the Pr_t-field given by Eq. (6.5.14), then there would have been no hope that LIFE would be able to find an adequate solution to the full learning problem.

Furthermore, this problem served as an excellent study for guiding feature design. The domain of Eq. (6.5.14) is $(0, \infty)$[2], which is unbounded. If the augmentation utilized a piecewise multilinear representation for the feature space, then LIFE would be unable to learn a feature of the exact same form as Eq. (6.5.14) as such a representation would require a closed interval for its domain. Thus, we expected that different functional forms of μ_t/μ with a finite range would have varying ability to recreate the trend of Kays and Crawford.

6.5.3. Inferring the Kays-Crawford model

6.5.3.1. LIFE setup

To solve the simplified learning problem of Eqn. 6.5.15, two optimizations using LIFE, each with a different input feature based on $\frac{\mu_t}{\mu}$, were conducted. For these, the training data set was just the three cases from Schülein's study. One used the simple "wrapped" feature

$$\eta = \frac{\mu}{\mu_t + \mu}, \tag{6.5.16}$$

which has a range of $(0, 1]$. The wrapped feature and others with a similar form were successfully used by Srivastava and Duraisamy to improve turbulence transition modeling [87]. It represents one of the simplest ways to bound an unbounded, non-negative quantity. However, when a linear discretization of the feature space is used, features of this form suffer from severe compression and loss of information when the input quantity has values over several orders of magnitude. For the

[2]Note it has a valid limit of 0.5882 as $\mu_t/\mu \to 0$

hypersonic cases used in this work, the μ_t/μ values predicted by RANS encompass a range of over ten orders of magnitude. Since notable regions of the flow attain values throughout this range, we sought a feature more suited to a linear discretization, resulting in the "logarithmed" feature

$$\eta = 5 \tanh\left(\frac{\log_{10}\left(\frac{\mu_t}{\mu} + 10^{-16}\right)}{0.75 \cdot 5}\right), \tag{6.5.17}$$

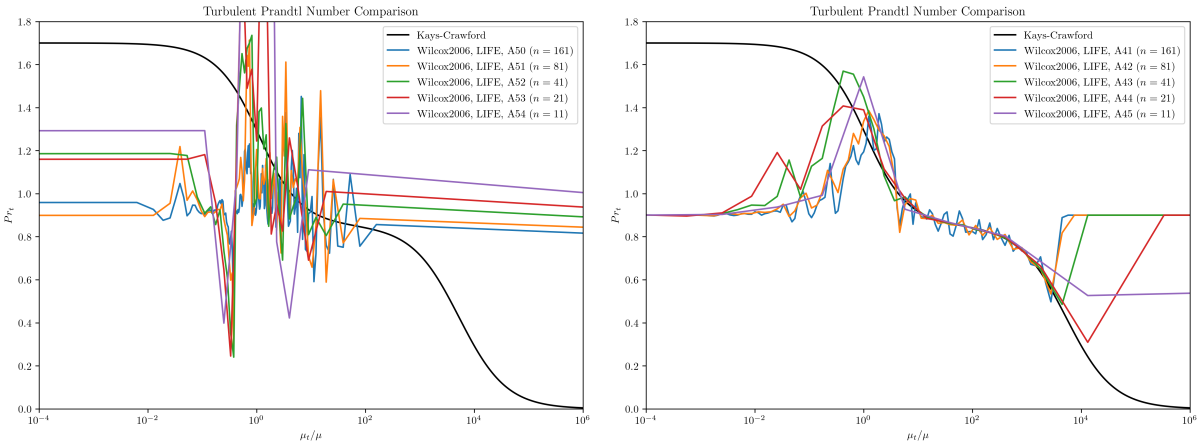
which has an output bounded within $(-5, 5)$. Taking the common logarithm can be interpreted as determining the order of magnitude of μ_t/μ at that point in the mesh. The 10^{-16} is required to prevent $\log_{10}(0)$ from being calculated and causing "not a number" issues. The hyperbolic tangent is used to bound the output of the logarithm. Since the most important values seemed to fall within $[10^{-4}, 10^{4}]$, a value of 5 was chosen to encompass this range while minimizing compression. The factor of 0.75 was included to provide additional fidelity within this range.

6.5.3.2. Results

Figure 6-12 shows the inferred $\mathrm{Pr_t}$ vs. μ_t/μ relationships for several configurations of each model. Both LIFE models were tested with 161, 81, 41, 21, and 11 nodes in the feature space. Regardless of the number of discretization points in the feature space, the logarithmed feature was able to much more closely and smoothly approximate the Kays-Crawford model than the wrapped feature. While it was unable to replicate the asymptotic behavior due to the lack of data for the most extreme values of μ_t/μ, it was able to match the transition between the two almost exactly. The models utilizing fewer nodes matched the Kays-Crawford model more smoothly and also attained higher values of $\mathrm{Pr_t}$ for the smaller μ_t/μ-values. The wrapped feature was simply unable to replicate the trend with any clarity. Due to the extreme compression of μ_t/μ-values far from unity in value, the optimization process yielded oscillations between higher and lower values for $\mathrm{Pr_t}$ in an attempt to have a mean behavior that somewhat resembled the trend. Furthermore, all models with the wrapped feature exhibited a value of $\mathrm{Pr_t} \approx 0.2$ around $\mu_t/\mu = 10^{-0.5}$, going completely against the behavior predicted by the Kays-Crawford model. The source of this has not yet been determined.

Figure 6-12b also provides insight on the balance between localized learning and generalizability for models trained using LIFE. As the number of nodes used decreased, the learned model was able to better match the Kays-Crawford trend for μ_t/μ-values that were higher than range of values presented in the training data. Thus it demonstrates the impact that hyper-parameters for the feature space discretization have on the generalizability of the model and highlights the careful analysis that must be performed by the user.

Figure 6-13 compares the heat transfer distributions of the various RANS models in terms of the Stanton number



(a) LIFE, wrapped feature **(b) LIFE, logarithmed feature**

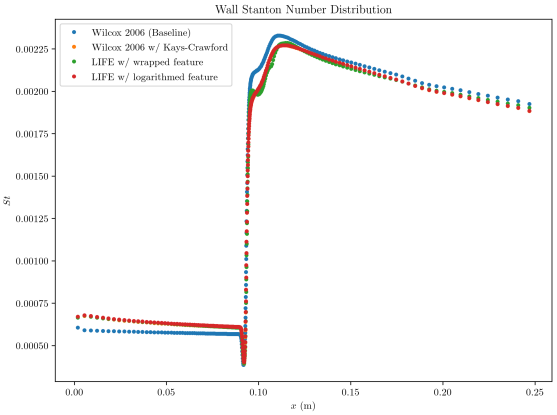
Figure 6-12. LIFE $\mathrm{Pr_t}$ predictions.

$$St = \frac{q_w}{\rho_\infty U_\infty c_p (T_r - T_w)}, \tag{6.5.18}$$

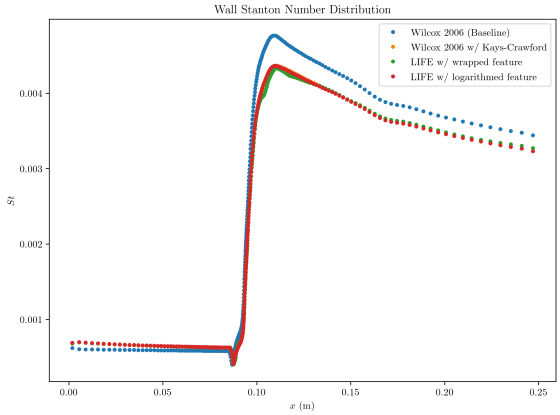
where ρ_∞, U_∞ are the reference freestream density and velocity, $c_p = 1004.675$ is the isobaric specific heat of air, T_r is the recovery temperature, and T_w is the wall temperature. The selected LIFE-inferred models are the ones using 21 nodes for the feature-space discretization. Figures 6-14-6-16 compare the Kays-Crawford $\mathrm{Pr_t}$-field with the inferred ones using a log scale for the wall-normal axis. Blue indicates regions of decreased turbulent Prandtl number, red regions of increase, and white regions of no change. Density contour lines are illustrated the gray lines to help visualize the shocks.

Unsurprisingly, the heat transfer predictions for the logarithmed feature almost perfectly match the ones from the Kays-Crawford model, shown by them lying top of each other. Despite the messy relationship shown in Figure 6-12a, the wrapped feature was still able to match the Kays-Crawford model at a level that would generally be considered satisfactory.

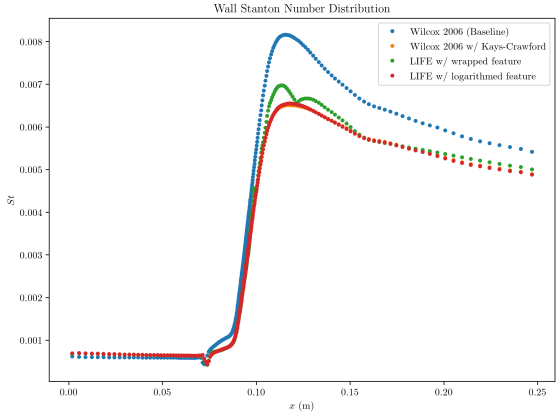
However, the $\mathrm{Pr_t}$ contour plots highlight the issue with the wrapped feature. Disregarding the freestream, the field predicted by the Kays-Crawford model is a very simple and reasonable field. It attains high values at the wall before decreasing throughout the boundary layer. While the logarithmed feature was not able to exactly match the behavior at the wall (which might be ameliorated by finer wall-spacing in the mesh), it was able to match the boundary layer behavior almost exactly. The wrapped feature was unable to do this. The $\mathrm{Pr_t}$ fields for its predictions rapidly switch between increased and decreased values in a manner that does not seem at all realistic. Furthermore, it was not able to obtain the band of very high turbulent Prandtl number near the wall and downstream of the interaction for Case S14.



(a) Case S6

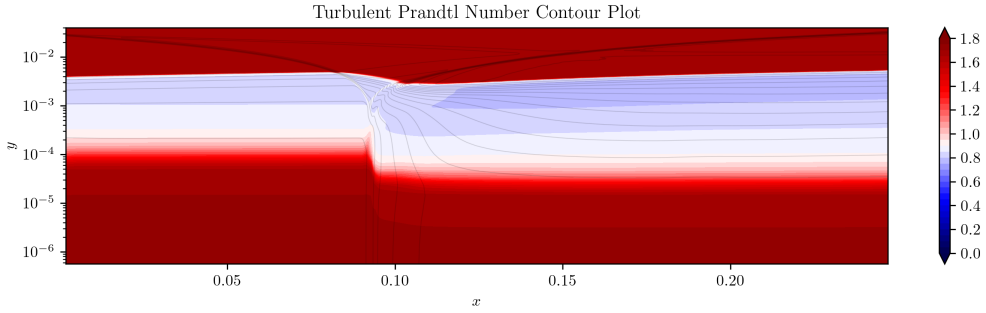


(b) Case S10

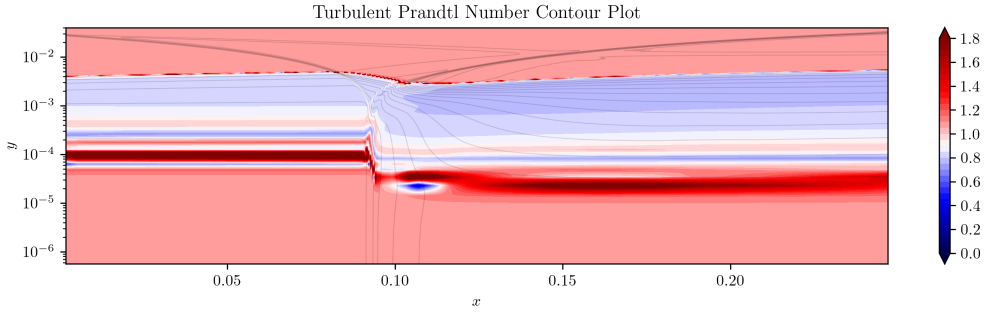


(c) Case S14

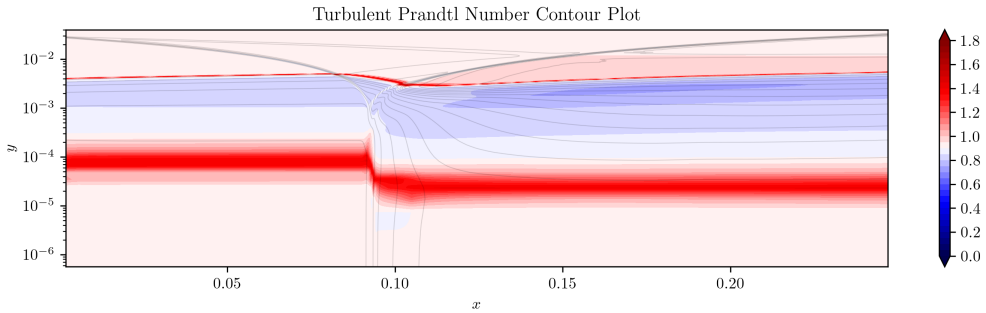
Figure 6-13. LIFE wall heat transfer distributions



(a) Kays-Crawford

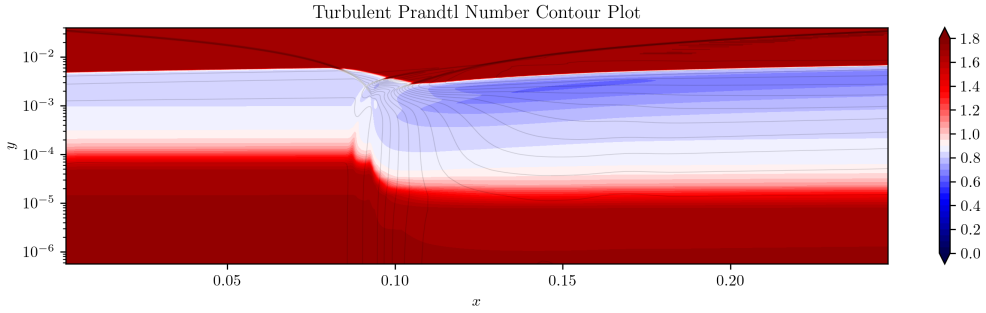


(b) LIFE, wrapped feature

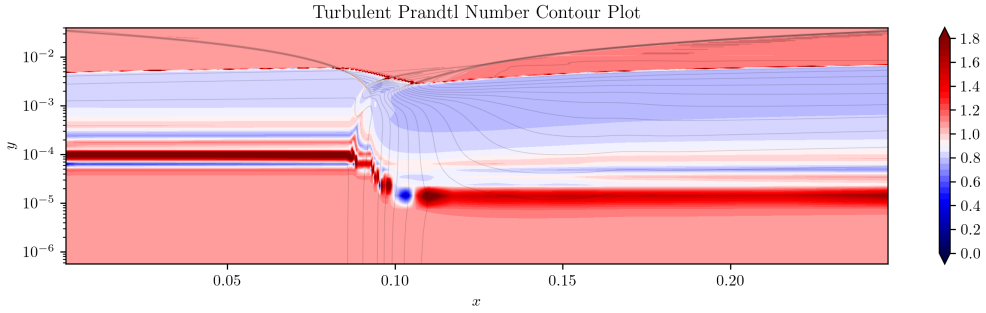


(c) LIFE, logarithmed feature

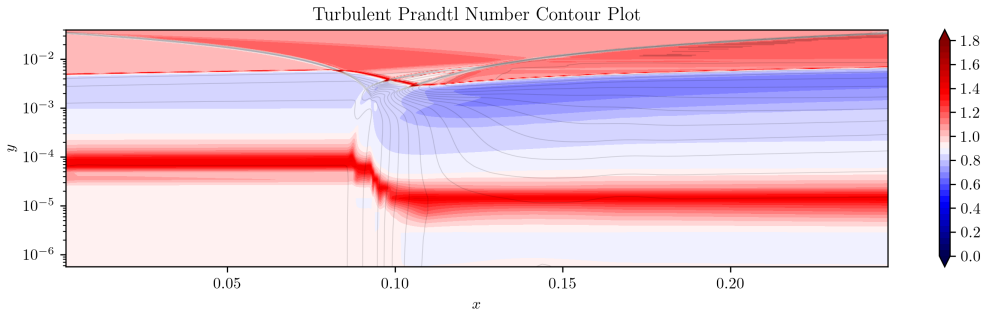
Figure 6-14. Inferred $\mathrm{Pr_t}$-fields for learning the Kays-Crawford model for Case S6



(a) Kays-Crawford

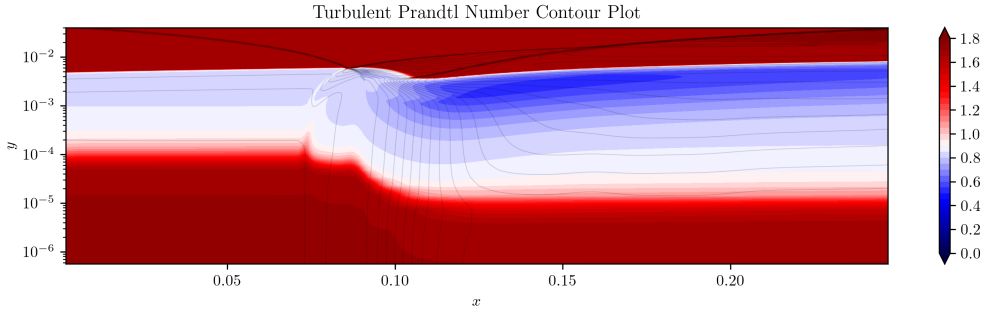


(b) LIFE, wrapped feature

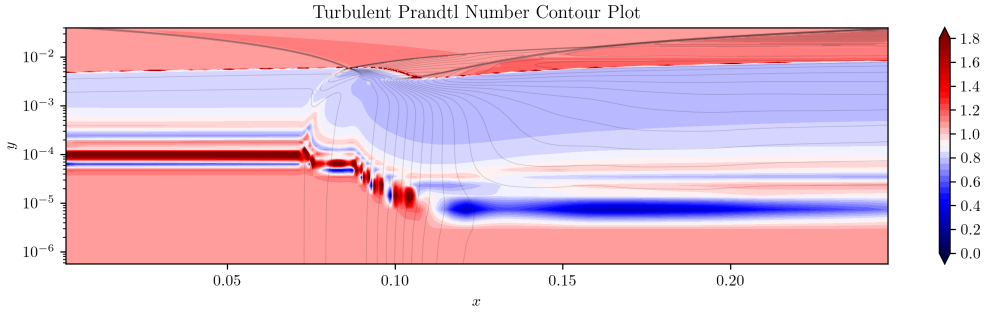


(c) LIFE, logarithmed feature

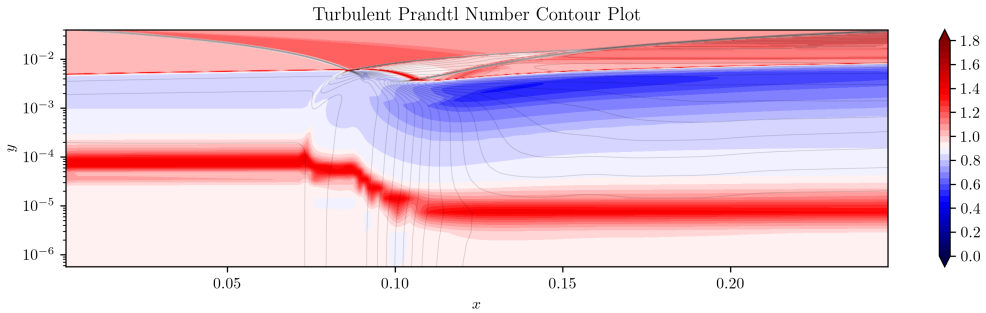
Figure 6-15. Inferred $\mathrm{Pr_t}$-fields for learning the Kays-Crawford model for Case S10



(a) Kays-Crawford



(b) LIFE, wrapped feature



(c) LIFE, logarithmed feature

Figure 6-16. Inferred $\mathrm{Pr_t}$-fields for learning the Kays-Crawford model for Case S14

Table 6-5. Features used for LIFE

Feature	Lower Bound	Upper Bound	# Nodes
$5\tanh\left(\frac{\log_{10}\left(\mu_t/\mu+10^{-16}\right)}{0.75\cdot 5}\right)$ (Eq. (6.5.17))	-5.0	5.0	41
$1.5\tanh\left(\frac{1}{1.5}\frac{\nabla T\cdot\hat{n}}{T/d_{\text{wall}}}\right)$	-1.5	1.5	41

Despite the troubles with the wrapped feature, the logarithmed feature proved that the LIFE framework, provided that carefully designed features are used, can be applied to our hypersonics heat transfer problem as it was able to successfully replicate an existing variable turbulent Prandtl number model. Furthermore, it suggested that, with an appropriate choice for the feature-space discretization, there is potential for predictive modeling to unseen cases.

6.5.4. Inferring and predicting experimental data

6.5.4.1. LIFE setup

In a preliminary attempt to solve the full learning problem given by Eq. (6.5.12), an optimization using LIFE with two input features was performed. The feature space was defined according to Table 6-5. The first is the same logarithmed feature used to learn the Kays-Crawford model. The second is based on the wall-normal temperature gradient nondimensionalized by the cell's local temperature and wall distance. A wall-normal temperature gradient feature was selected after other preliminary work indicated it gave valuable insight on where standard turbulence models failed to accurately predict wall-normal heat transfer. The model was trained on data from all three of the Schülein cases and then tested against one of DNS cases from Nicholson et al. [57].

6.5.4.2. Training results

Figure 6-17 shows the evolution of the objective function during training. For all cases, the cost function was able to improve by over two orders of magnitude. The optimization was terminated after the 21$^{\text{st}}$ step due to forward simulations failing to converge.

Figure 6-18 shows the heat transfer distributions of the experimental data, baseline model, and LIFE model. Figure 6-19 shows the inferred Pr_t-fields, again using a log scale for the wall-normal axis. We can see that the inferred model is able to fairly closely match the experimental data downstream of the interaction. It also accomplishes this with minimal impact on the flow upstream of the interaction and with virtually zero change to the wall pressure and skin friction distributions[3]. There are, however, some visible oscillations in the heat transfer distributions for the weaker interaction cases.

[3]Plots are omitted for brevity.

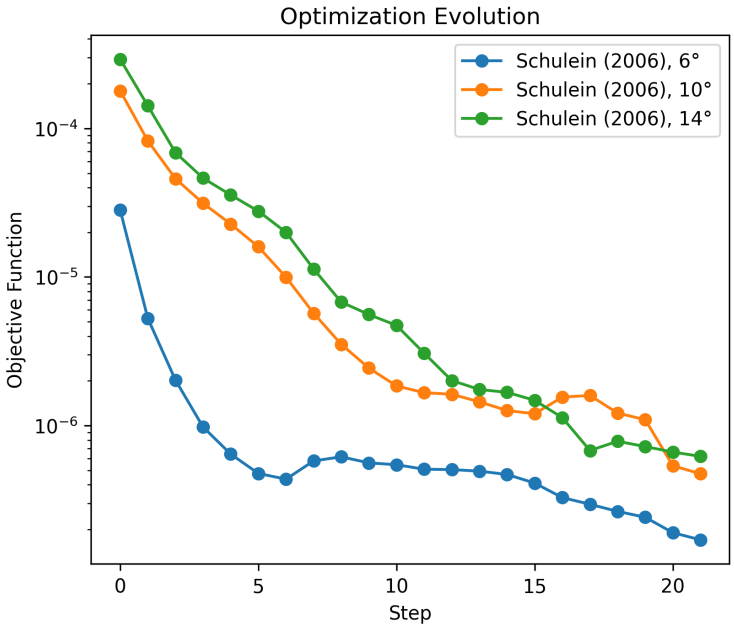
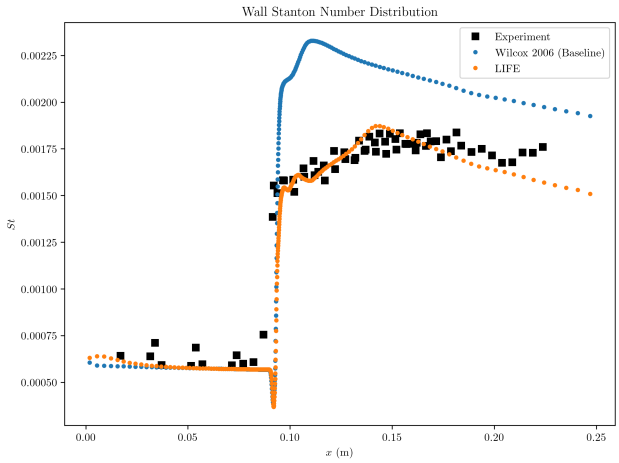


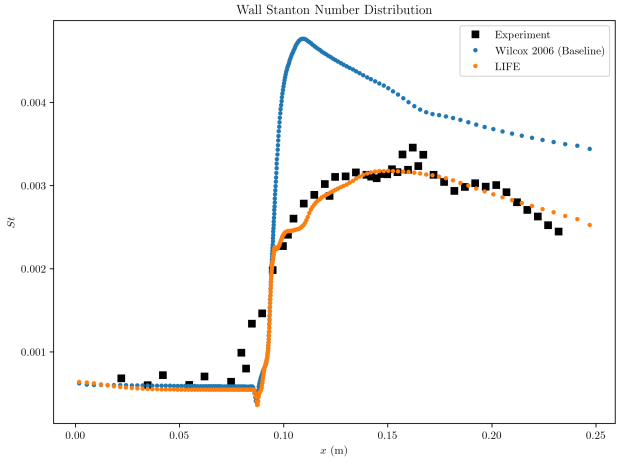
Figure 6-17. Training objective evolution

The inferred turbulent Prandtl number fields are markedly different from the ones suggested by the Kays-Crawford model, but align much better with the fields suggested by preliminary FI studies we conducted. The dominant feature is the large region of significantly decreased Pr_t in the boundary layer at and downstream of the shock interaction. It is most pronounced for Case S14, which is the strongest interaction case. Other minor features are the small regions of increased Pr_t just upstream of the interaction within the boundary layer and in the upper edge of the boundary layer, both of which the size and strength correlate with the shock strength.

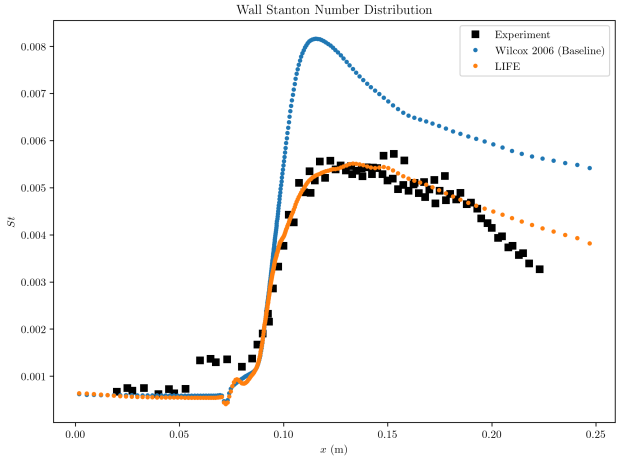
The concerning aspects of the fields are the alternating bands of increased and decreased Pr_t in the upstream regions of all cases and in the downstream regions near the wall, the latter of which is inversely correlated with shock strength. These bands are reminiscent of the oscillatory behavior exhibited by the LIFE-trained model using the wrapped feature to learn the Kays-Crawford relationship. In general, these bands result from the features and their representations being unable to properly distinguish between the various flow phenomena. Regarding the upstream bands, the optimization is driven primarily by cells located near the wall at and downstream of the interaction. However, feature-values in these cells also match some of those within the boundary layer in the upstream regions, affecting the heat transfer prior to the shock. The optimization then tries to compensate for this by adding a region of opposing behavior adjacent to it. Unsurprisingly, this causes a domino effect, resulting in the bands. To address this, a third feature that can distinguish between undisturbed and disturbed boundary layers will likely have to be added. The bands in the downstream region are simply explained by having an inadequate set of features. The two chosen features and their representations are simply unable to properly distinguish the different flow regions. While adding an additional feature will likely help, the features might also need to



(a) Case S6



(b) Case S10



(c) Case S14

Figure 6-18. LIFE wall heat transfer distributions

be modified or replaced.

Despite these concerns, the learned model is still a fully functional augmentation that can use low fidelity states to fairly closely match the desired behavior. Thus, it highlights the LIFE framework's ability to obtain a learnable model provided a suitable set of features and feature space discretization.

6.5.4.3. Testing results

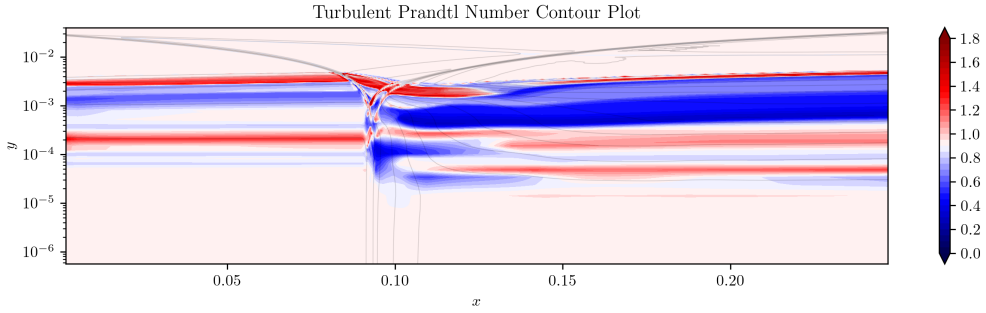
To assess the robustness and generalizability of the learned model, we compared it to the DNS results of a moderate steepness forward-facing curved wall interacting with a turbulent Mach 4.87 boundary layer from Nicholson et al. (referred to as M5FFW0p26 in their study) [57]. While the Mach number and freestream Reynolds number are comparable with the training cases, the geometry, and consequently the nature of the shock interaction is significantly different. Here, a single shock is generated as a result of the flow geometry, whereas in the training data a shock is externally generated and causes a second, reflected shock. Furthermore, the curved nature of the wall will cause the flow features to evolve differently than the abrupt nature of an incident shock.

Figure 6-21 shows that the predicted Pr_t-field for this test case generally matches what was seen in the training cases. The dominant feature is the region of significantly decreased turbulent Prandtl number in the boundary layer downstream of the shock interaction. The oscillatory bands throughout the boundary layer upstream of the shock and in the near-wall region downstream of the shock are again present.

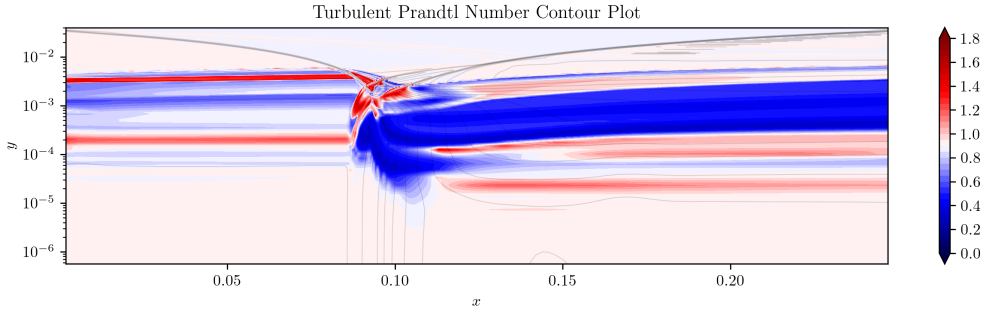
Despite the inability to account for the expansion corner and the overaggressive correction, this preliminary model trained using LIFE was able to significantly improve the heat transfer predictions when applied to a case with a geometry outside of its training regime. In addition, the prediction remained physically realizable; there was no spurious behavior. These highlight the LIFE framework's ability to generate both generalizable and robust predictive models.

6.5.5. Conclusions and future work

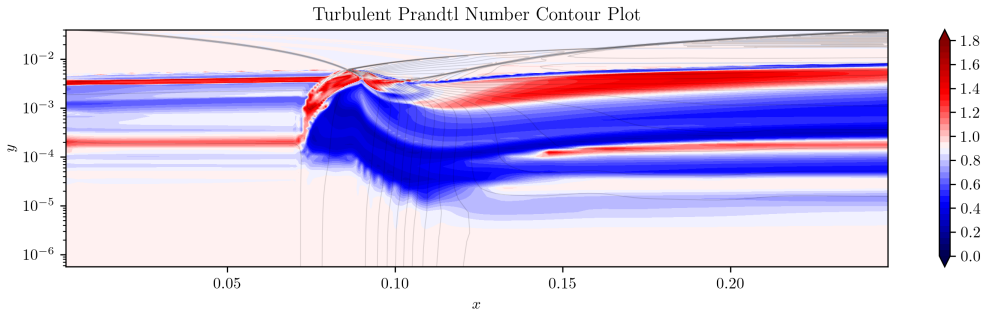
This section presented preliminary results suggesting that the LIFE framework proposed by Srivastava and Duraisamy [87] can be used to learn variable turbulent Prandtl number models to improve the RANS heat transfer of hypersonic SBLIs. We first took the full optimization problem of learning a multiple-feature model to match experimental data and simplified it to a problem for learning a single-feature model to recreate the existing variable turbulent Prandtl number model of Kays and Crawford. We showed that with a carefully constructed feature that provided fidelity over the full range of values, a simple piecewise linear representation of the feature was able to closely match the Kays-Crawford model. This effort gave confidence to the LIFE framework and provided us with valuable insight for feature development. We then expanded on these findings to create a



(a) Case S6

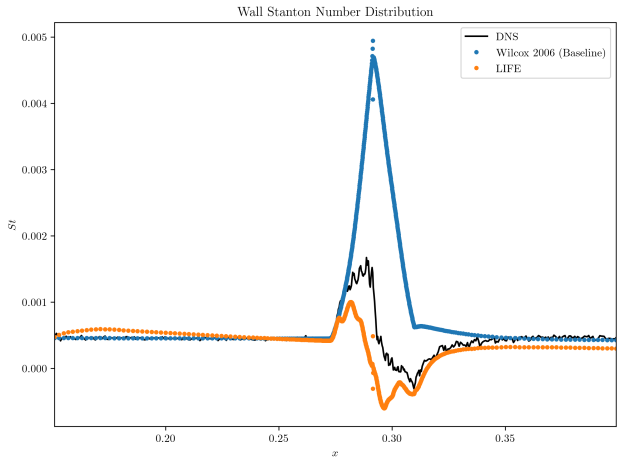


(b) Case S10

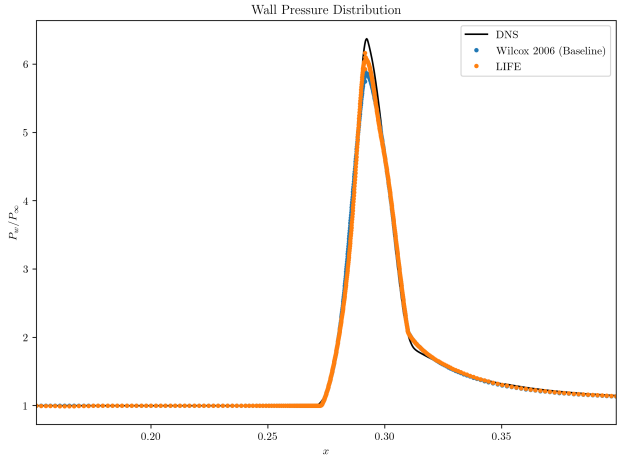


(c) Case S14

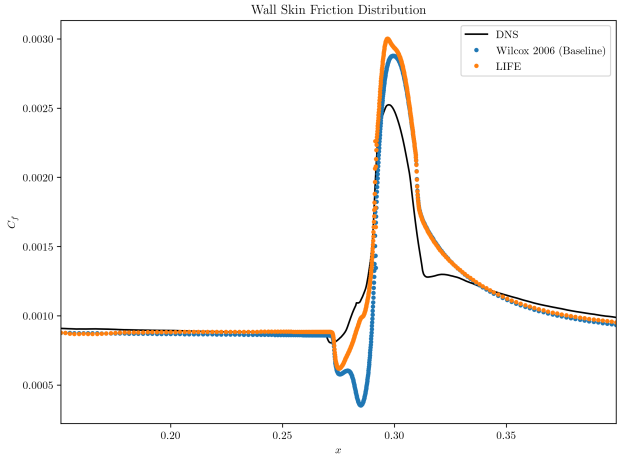
Figure 6-19. Inferred $\mathrm{Pr_t}$-fields for the LIFE-trained model



(a) Heat transfer



(b) Pressure



(c) Skin friction

Figure 6-20. Predicted wall quantities for Case N026

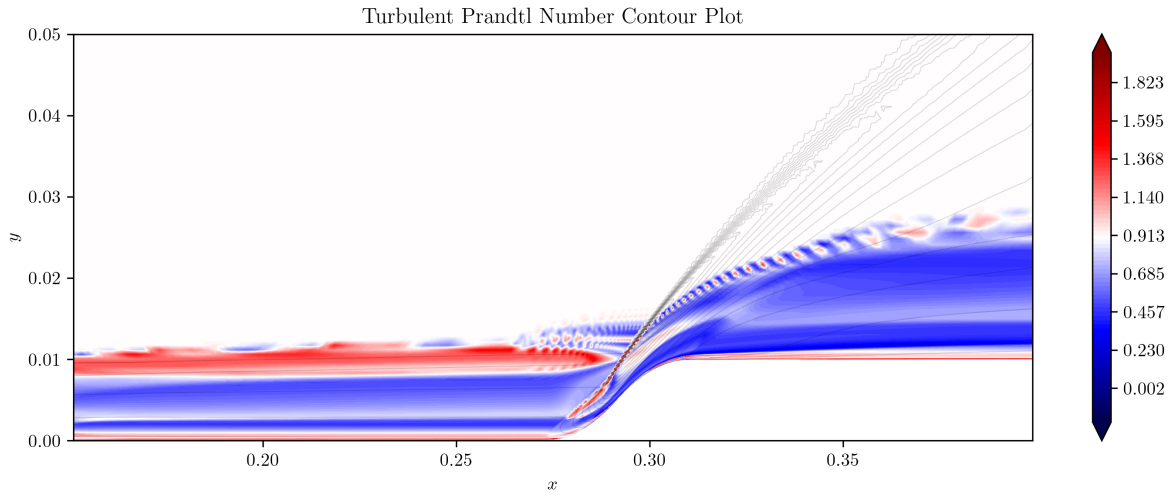


Figure 6-21. Predicted $\mathrm{Pr_t}$-field for Case N026

preliminary two-feature model trained on the experimental data from a Mach 5 oblique SBLI study. LIFE was able to infer a model that resulted in satisfactory predictions on the training cases and significantly improved the heat transfer predictions for a case with a drastically different geometry, highlighting its potential as a learning approach.

The model presented in this report is preliminary and still being improved upon. The current version is unable to properly distinguish between all relevant flow features and regions to create an augmentation that properly isolates and distinguishes between the regions of the flow the baseline model is inaccurate for. Furthermore, this model, and others developed but not covered in this report, have been unable to improve the heat transfer predictions for flows at higher Mach numbers and with more significant adverse pressure gradients. We plan to methodically consider additional features to add and replace the current ones. Specifically, we will first focus on developing features that can ensure that undisturbed boundary layers remain unaffected by the learned augmentation and result in much smoother predicted discrepancy fields. Once this is accomplished, we will expand and modify the model to handle more extreme hypersonic cases.

This page intentionally left blank.

7. CONCLUSIONS

This report summarized work performed from FY22-FY24 focused on improving RANS models for hypersonic flows using data-driven modeling and scientific machine learning. In this report, we investigated the performance of RANS models currently implemented in SPARC on a range of hypersonic problems with a focus on SBLIs. We highlighted deficiencies in current capabilities. To address these deficiencies, we developed data-driven corrections to the Reynolds stress tensor and turbulent heat flux. First, we formulated a discrepancy approach that corrects an existing RANS model with a tensor-basis neural network (TBNN). To further improve the robustness of the TBNN, we developed approaches to enforce smoothness via Lipschitz-continuous networks and quantify model-form uncertainty via calibrated deep ensembles. We demonstrated the approach to improve predictions for a $k-\epsilon$ model. We showed that the resulting ML-enhanced model yields more accurate predictions across a suite of low-speed and high-speed problems displaying various physics. For hypersonic boundary layers, we found that the wall-normal Reynolds stress has an appreciable impact on the solution. When deployed on hypersonic SBLIs, the ML-enhanced models resulted in improved predictions, but significant discrepancies were present. In particular, wall heat flux is significantly over-predicted near the point of reattachment.

To address the issue of over-prediction in wall heat flux, Chapter 5 developed an approach for learning a variable turbulent Prandtl number to improve predictions for wall heating in SBLIs. The developed approach learns a neural-network-based variable turbulent Prandtl number model using field inversion and machine learning (FIML). We demonstrated the approach for improving a $k-\epsilon$ model, and showed that the resulting variable turbulent Prandtl number model improves predictions on a suite of test cases including the hollow cylinder flare and HIFiRE ground test experiments.

Lastly, Chapter 6 highlighted promising future directions. We:

- Showed preliminary predictions of a variable turbulent Prandtl number model on Sandia's cone-slice-wedge geometry,
- Showed preliminary results of a $k-\epsilon$ model employing both a variable turbulent Prandtl number model and a TBNN-based Reynolds stress model,
- Investigated the use of the ensemble Kalman filter for learning a variable turbulent Prandtl number model,
- Showed preliminary results for the LIFE framework deployed to learning a variable turbulent Prandtl number model.

All of these future directions show improved results and highlight the future promise of data-driven turbulence modeling for hypersonic turbulent flows.

BIBLIOGRAPHY

[1] Timothy T. Aiken, Iain D. Boyd, Lian Duan, and Junji Huang. Assessment of Reynolds Averaged Navier-Stokes Models for a Hypersonic Cold-Wall Turbulent Boundary Layer. In *AIAA SCITECH 2022 Forum*, 2022.

[2] AJ Banko and JK Eaton. Estimating performance bounds of machine-learning Reynolds-stress models via optimal tensor basis expansions. *Center for Turbulence Research Annual Research Briefs*, 2020.

[3] M. Barone, G. L. Nicholson, and L. Duan. Internal energy balance and aerodynamic heating predictions for hypersonic turbulent boundary layers. *Phys. Ref. Fluids*, 7:084604, 2022.

[4] Matthew Barone, Eric Parish, and Cyrus Jordan. Data-Driven Modifications to the Spalart–Allmaras Turbulence Model for Supersonic and Hypersonic Boundary Layers. In *AIAA SciTech*, 2024.

[5] Matthew Barone, Jaideep Ray, and Stefan Domino. Feature Selection, Clustering, and Prototype Placement for Turbulence Datasets. *AIAA Journal*, 60(3):1332–1346, 2022.

[6] Rodney D. W. Bowersox. Extension of equilibrium turbulent heat flux models to high-speed shear flows. *Journal of Fluid Mechanics*, 633:61–70, 2009.

[7] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[8] J. Bridges and M. P. Wernet. The NASA Subsonic Jet Particle Image Velocimetry (PIV) Dataset. *Tech. Report*, (NASA/TM-2011-216807), 2011.

[9] Stéphane Catris and Bertrand Aupoix. Density corrections for turbulence models. *Aerospace Science and Technology*, 4(1):1–11, 2000.

[10] Sai Hung Cheung, Todd A. Oliver, Ernesto E. Prudencio, Serge Prudhomme, and Robert D. Moser. Bayesian uncertainty analysis with applications to turbulence modeling. *Reliability Engineering & System Safety*, 96(9):1137–1149, 2011. Quantification of Margins and Uncertainties.

[11] Kenny Chowdhary, Chi Hoang, Kookjin Lee, Jaideep Ray, V.G. Weirs, and Brian Carnes. Calibrating hypersonic turbulence flow models with the HIFiRE-1 experiment using data-driven machine-learned models. *Computer Methods in Applied Mechanics and Engineering*, 401:115396, 2022.

[12] T. Coakley and P. Huang. Turbulence modeling for high speed flows. In *30th Aerospace Sciences Meeting and Exhibit*, 1992.

[13] T. J. Coakley, C. C. Horstman, J. G. Marvin, J. R. Viegas, J. E. Bardina, P. G. Huang, and M. I. Kussoy. Turbulence Compressibility Corrections. Technical report, NASA, 94.

[14] Guanlin Dang, Shiwei Liu, Tongbiao Guo, Junyi Duan, and Xinliang Li. Direct numerical simulation of compressible turbulence accelerated by graphics processing unit: An open-access database of high-resolution direct numerical simulation. *AIP Advances*, 12(12):125111, 12 2022.

[15] Mustafa E. Danis and Paul Durbin. Compressibility Correction to k-w Models for Hypersonic Turbulent Boundary Layers. *AIAA Journal*, 60(11):6225–6234, 2022.

[16] Douglas Dolvin. Hypersonic international flight research and experimentation (HIFiRE) fundamental science and technology development strategy. In *15th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*, page 2581, 2008.

[17] Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. Turbulence Modeling in the Age of Data. *Annual Review of Fluid Mechanics*, 51(1):357–377, 2019.

[18] W. N. Edeling, P. Cinnella, R. P. Dwight, and H. Bijl. Bayesian estimates of parameter variability in the k-e turbulence model. *Journal of Computational Physics*, 258:73–94, February 2014.

[19] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep Ensembles: A Loss Landscape Perspective, 2019.

[20] Jochen Fröhlich, Christopher. P. Mellen, Wolfgang Rodi, Lionel Temmerman, and Michael a. Leschziner. Highly resolved large-eddy simulation of separated flow in a channel with streamwise periodic constrictions. *Journal of Fluid Mechanics*, 526:19–66, 2005.

[21] Sergei Garmaev and Sergey Yakovenko. Turbulence model development using machine learning methods for a channel flow. In *AIP Conference Proceedings*, volume 2504. AIP Publishing, 2023.

[22] Peter Gnoffo, Scott Berry, and John Van Norman. Uncertainty Assessments of 2D and Axisymmetric Hypersonic Shock Wave - Turbulent Boundary Layer Interaction Simulations at Compression Corners. *42nd AIAA Thermophysics Conference*, 2011.

[23] Peter A. Gnoffo, Scott A. Berry, and John W. Van Norman. Uncertainty Assessments of Hypersonic Shock Wave-Turbulent Boundary-Layer Interactions at Compression Corners. *Journal of Spacecraft and Rockets*, 50(1):69–95, 2013.

[24] David Greenblatt, Keith B. Paschal, Chung-Sheng Yao, and Jerome Harris. Experimental Investigation of Separation Control Part 2: Zero Mass-Flux Oscillatory Blowing. *AIAA Journal*, 44(12):2831–2845, 2006.

[25] David Greenblatt, Keith B. Paschal, Chung-Sheng Yao, Jerome Harris, Norman W. Schaeffler, and Anthony E. Washburn. Experimental Investigation of Separation Control Part 1: Baseline and Steady Suction. *AIAA Journal*, 44(12):2820–2830, 2006.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[27] Michael Holden. Measurements in Regions of Shock Wave/Turbulent Boundary Layer Interaction From MACH 4 to 7 at Flight Duplicated Velocities to Evaluate and Improve the Models of Turbulence in CFD Codes. In *2018 Fluid Dynamics Conference*, 2018.

[28] Michael Holden, Timothy Wadhams, Matthew MacLean, and Erik Mundy. Experimental studies of shock wave/turbulent boundary layer interaction in high Reynolds number supersonic and hypersonic flows to evaluate the performance of CFD codes. In *40th Fluid Dynamics Conference and Exhibit*, 2010.

[29] Michael S. Holden, Tim P. Wadhams, and Matthew G. MacLean. Measurements in Regions of Shock Wave/Turbulent Boundary Layer Interaction from Mach 4 to 10 for Open and "Blind" Code Evaluation/Validation. In *21st AIAA Computational Fluid Dynamics Conference*, 2013.

[30] Jonathan R. Holland, James D. Baeder, and Karthik Duraisamy. Towards Integrated Field Inversion and Machine Learning With Embedded Neural Networks for RANS Modeling. In *AIAA Scitech 2019 Forum*, 2019.

[31] Jonathan R. Holland, James D. Baeder, and Karthikeyan Duraisamy. Field Inversion and Machine Learning With Embedded Neural Networks: Physics-Consistent Neural Network Training. 2019.

[32] Micah Howard, Andrew Bradley, Steven W. Bova, James Overfelt, Ross Wagnild, Derek Dinzl, Mark Hoemmen, and Alicia Klinvex. Towards Performance Portability in a Compressible CFD Code. In *23rd AIAA Computational Fluid Dynamics Conference*, volume 1, Denver, CO, 2017. AIAA.

[33] Sergio Hoyas and Javier Jiménez. Reynolds number effects on the Reynolds-stress budgets in turbulent channels. *Physics of Fluids*, 20(10):101511, 2008.

[34] Junji Huang, Lian Duan, and Meelan M. Choudhari. Direct numerical simulation of hypersonic turbulent boundary layers: effect of spatial evolution and reynolds number. *Journal of Fluid Mechanics*, 937:A3, 2022.

[35] Junji Huang, Gary L. Nicholson, Lian Duan, Meelan M. Choudhari, and Rodney D. Bowersox. Simulation and Modeling of Cold-Wall Hypersonic Turbulent Boundary Layers on Flat Plate. 2020.

[36] P. HUANG and T. COAKLEY. Turbulence modeling for complex hypersonic flows. In *31st Aerospace Sciences Meeting*, 1993.

[37] Marco A Iglesias, Kody JH Law, and Andrew M Stuart. Ensemble Kalman methods for inverse problems. *Inverse Problems*, 29(4):045001, 2013.

[38] Cyrus Jordan. *Turbulence Model Development for Hypersonic Shock Wave Boundary Layer Interactions* . PhD thesis, North Carolina State University, 2023.

[39] Cyrus Jordan, Jack R. Edwards, and Douglas L. Stefanski. Evaluation of RANS Closure Models using LES Datasets for Hypersonic Shock Boundary Layer Interactions. In *AIAA Scitech 2021 Forum*, 2021.

[40] William M. Kays and Michael E. Crawford. *Convective Heat and Mass Transfer*. McGraw-Hill, 3rd edition, 1993.

[41] Nikola B Kovachki and Andrew M Stuart. Ensemble Kalman inversion: a derivative-free technique for machine learning tasks. *Inverse Problems*, 35(9):095005, 2019.

[42] Marvin Kussoy and Clifford Horstman. Documentation of Two- and Three-Dimensional Hypersonic Shock Wave/Turbulent Boundary Layer Interaction Flows. Technical report, NASA, 1989.

[43] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. *Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles*, page 6405–6416. NIPS'17. Curran Associates Inc., Red Hook, NY, USA, 2017.

[44] Johan Larsson, Ivan Bermejo-Moreno, and Sanjiva K. Lele. Reynolds- and mach-number effects in canonical shock–turbulence interaction. *Journal of Fluid Mechanics*, 717:293–321, 2013.

[45] B. E. Launder. On the Computation of Convective Heat Transfer in Complex Turbulent Flows. *Journal of Heat Transfer*, 110(4b):1112–1128, 11 1988.

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

[47] Xiaodong Luo, Andreas S Stordal, Rolf J Lorentzen, and Geir Nævdal. Iterative ensemble smoother as an approximate solution to a regularized minimum-average-cost problem: theory and applications. *SPE Journal*, 20(05):962–982, 2015.

[48] Matthew MacLean, Timothy Wadhams, Michael Holden, and Heath Johnson. Ground Test Studies of the HIFiRE-1 Transition Experiment Part 2: Computational Analysis. *Journal of Spacecraft and Rockets*, 45(6):1149–1164, 2008.

[49] Joseph G. Marvin, James L. Brown, and Peter A. Gnoffo. Experimental Database with Baseline CFD Solutions: 2-D and Axisymmetric Hypersonic Shock-Wave/Turbulent-Boundary-Layer Interactions. Technical report, NASA, 2013.

[50] Pedro M. Milani, Julia Ling, and John K. Eaton. On the generality of tensor basis neural networks for turbulent scalar flux modeling. *International Communications in Heat and Mass Transfer*, 128:105626, 2021.

[51] Pedro M. Milani, Julia Ling, and John K. Eaton. Turbulent scalar flux in inclined jets in crossflow: counter gradient transport and deep learning modelling. *Journal of Fluid Mechanics*, 906:A27, 2021.

[52] Pedro M. Milani, Julia Ling, Gonzalo Saez-Mischlich, Julien Bodart, and John K. Eaton. A Machine Learning Approach for Determining the Turbulent Diffusivity in Film Cooling Flows. *Journal of Turbomachinery*, 140(2):021006, 12 2017.

[53] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks, 2018.

[54] Brandon Morgan, K. Duraisamy, N. Nguyen, S. Kawai, and S. K. Lele. Flow physics and RANS modelling of oblique shock/turbulent boundary layer interaction. *J. Fluid Mech.*, 729:231–284, 2013.

[55] J. W. Naughton, S. Viken, and D. Greenblatt. Skin Friction Measurements on the NASA Hump Model. *AIAA Journal*, 44(6):1255–1265, 2006.

[56] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images, 2014.

[57] Gary L. Nicholson, Lian Duan, and Nicholas J. Bisek. Direct Numerical Simulation Database of High-Speed Flow over Parameterized Curved Walls. *AIAA Journal*, 62(6):2095–2118, 2024.

[58] Todd A Oliver and Robert D Moser. Bayesian uncertainty quantification applied to RANS turbulence models. *Journal of Physics: Conference Series*, 318(4):042032, dec 2011.

[59] James Ott, Donald Kenzakowski, and Sanford Dash. Evaluation of Turbulence Modeling Extensions for the Analysis of Hypersonic Shock Wave Boundary Layer Interactions. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2013.

[60] Anshuman Pandey, Katya M. Casper, and Steven J. Beresh. Relaminarization effects in hypersonic flow on a three-dimensional expansion–compression geometry. *Journal of Fluid Mechanics*, 985:A25, 2024.

[61] Eric Parish, David S. Ching, Nathan E. Miller, Steven J. Beresh, and Matthew F. Barone. Turbulence modeling for compressible flows using discrepancy tensor-basis neural networks and extrapolation detection. In *AIAA SCITECH 2023 Forum*, 2023.

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

[63] Eric L. Peters, Riccardo Balin, Kenneth E. Jansen, Alireza Doostan, and John A. Evans. S-frame discrepancy correction models for data-informed Reynolds stress closure. *Journal of Computational Physics*, 448:110717, 2022.

[64] Alfredo Pinelli, Markus Uhlmann, Atsushi Sekimoto, and Genta Kawahara. Reynolds number dependence of mean flow structure in square duct turbulence. *Journal of Fluid Mechanics*, 653:537–537, 2010.

[65] S. B. Pope. A more general effective-viscosity hypothesis. *J. Fluid Mech.*, 72(2):331–340, 1975.

[66] Stephan Priebe and M. Pino Martín. Turbulence in a hypersonic compression ramp flow. *Phys. Rev. Fluids*, 6:034601, Mar 2021.

[67] Jaideep Ray, Sophia Lefantzi, Srinivasan Arunajatesan, and Lawrence Dechant. Bayesian Parameter Estimation of a k-e Model for Accurate Jet-in-Crossflow Simulations. *AIAA Journal*, 54(8):2432–2448, 2016.

[68] Christopher J. Roy and Frederick G. Blottner. Review and assessment of turbulence models for hypersonic flows. *Progress in Aerospace Sciences*, 42(7):469–530, 2006.

[69] Subhajit Roy, Utkarsh Pathak, and Krishnendu Sinha. Variable Turbulent Prandtl Number Model for Shock/Boundary-Layer Interaction. *AIAA Journal*, 56(1):342–355, 2018.

[70] Subhajit Roy and Krishnendu Sinha. Variable turbulent Prandtl number model applied to hypersonic shock/boundary-layer interactions. In *2018 Fluid Dynamics Conference*, 2018.

[71] C. L. Rumsey, J.-R. Carlson, T. H. Pulliam, and P. R. Spalart. Improvements to the Quadratic Constitutive Relation based on NASA juncture flow data. *AIAA J.*, 58(10):4374–4384, 2020.

[72] C. L. Rumsey, T. B. Gatski, T. B. Sellers, W. L. III Vatsa, and S. A. Viken. Summary of the 2004 Computational Fluid Dynamics Validation Workshop on Synthetic Jets. *AIAA J.*, 44(2):194–207, 2006.

[73] Chris Rumsey, Brian Smith, and George Huang. Description of a Website Resource for Turbulence Modeling Verification and Validation. In *40th Fluid Dynamics Conference and Exhibit*, 2010.

[74] Christopher L. Rumsey. Compressibility Considerations for k-w Turbulence Models in Hypersonic Boundary-Layer Applications. *Journal of Spacecraft and Rockets*, 47(1):11–20, 2010.

[75] R. D. Sandberg, R. Tan, J. Weatheritt, A. Ooi, A. Haghiri, V. Michelassi, and G. Laskowski. Applying Machine Learnt Explicit Algebraic Stress and Scalar Flux Models to a Fundamental Trailing Edge Slot. *Journal of Turbomachinery*, 140(10), 09 2018. 101008.

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

[77] K.E. Schoenherr. Resistances of flat surfaces moving through a fluid. *Trans. SNAME*, (40):279–313, 1932.

[78] Erich Schülein. Skin Friction and Heat Flux Measurements in Shock/Boundary Layer Interaction Flows. *AIAA Journal*, 44(8):1732–1741, 2006.

[79] Juan A. Sillero, Javier Jiménez, and Robert D. Moser. One-point statistics for turbulent wall-bounded flows at Reynolds numbers up to $\delta^+ \approx 2000$. *Physics of Fluids*, 25(10):105102, 2013.

[80] Anand Pratap Singh. *A framework to improve turbulence models using full-field inversion and machine learning*. PhD thesis, University of Michigan, 2018.

[81] Anand Pratap Singh and Karthik Duraisamy. Using field inversion to quantify functional errors in turbulence closures. *Physics of Fluids*, 28(4):045110, 04 2016.

[82] Anand Pratap Singh, Karthik Duraisamy, and Brandon E. Morgan. Data-Augmented Modeling of Transition to Turbulence in Rayleigh-Taylor Mixing Layers. 1 2019.

[83] Anand Pratap Singh, Karthikeyan Duraisamy, and Ze Jia Zhang. Augmentation of Turbulence Models Using Field Inversion and Machine Learning. In *55th AIAA Aerospace Sciences Meeting*, 2017.

[84] Anand Pratap Singh, Shivaji Medida, and Karthik Duraisamy. Machine-Learning-Augmented Predictive Modeling of Turbulent Separated Flows over Airfoils. *AIAA Journal*, 55(7):2215–2227, 2017.

[85] Krishnendu Sinha, Krishnan Mahesh, and Graham V. Candler. Modeling shock unsteadiness in shock/turbulence interaction. *Physics of Fluids*, 15(8):2290–2297, 2003.

[86] P. R. Spalart and A. V. Garbaruk. Correction to the Spalart-Allmaras turbulence model, providing more accurate skin friction. *AIAA J.*, 58(5):1903–1905, 2020.

[87] Vishal Srivastava and Karthik Duraisamy. Generalizable physics-constrained modeling using learning and inference assisted by feature-space engineering. *Phys. Rev. Fluids*, 6:124602, Dec 2021.

[88] Cory Stack. Personal Communication, 2023.

[89] Salar Taghizadeh, Freddie D Witherden, and Sharath S Girimaji. Turbulence closure modeling with data-driven techniques: physical compatibility and consistency considerations. *New Journal of Physics*, 22(9):093023, sep 2020.

[90] Paul Batten Uriel C. Goldberg, Sampath Palaniswamy and Vinit Gupta. Variable Turbulent Schmidt and Prandtl Number Modeling. *Engineering Applications of Computational Fluid Mechanics*, 4(4):511–520, 2010.

[91] Pedro S. Volpiani, Matteo Bernardini, and Johan Larsson. Effects of a nonadiabatic wall on hypersonic shock/boundary-layer interactions. *Phys. Rev. Fluids*, 5:014602, Jan 2020.

[92] T. Wadhams, M. MacLean, M. Holden, and E. Mundy. Ground test studies of the HIFiRE-1 transition experiment Part 1: Experimental results. *J. Spacecraft Rockets*, 45(6):1134–1148, 2008.

[93] TP Wadhams, E Mundy, MG MacLean, and MS Holden. Ground test studies of the hifire-1 transition experiment part 1: experimental results. *Journal of Spacecraft and Rockets*, 45(6):1134–1148, 2008.

[94] Jian-Xun Wang, Junji Huang, Lian Duan, and Heng Xiao. Prediction of Reynolds stresses in high-Mach-number turbulent boundary layers using physics-informed machine learning. *Theoretical and Computational Fluid Dynamics*, 33(1):1–19, 2019.

[95] Jian-Xun Wang, Jin-Long Wu, and Heng Xiao. Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data. *Phys. Rev. Fluids*, 2:034603, Mar 2017.

[96] J. Weatheritt and R.D. Sandberg. The development of algebraic stress models using a novel evolutionary algorithm. *International Journal of Heat and Fluid Flow*, 68:298–318, 2017.

[97] Jack Weatheritt and Richard Sandberg. A novel evolutionary algorithm applied to algebraic modifications of the RANS stress–strain relationship. *Journal of Computational Physics*, 325:22–37, 2016.

[98] Jack Weatheritt, Richard D. Sandberg, Julia Ling, Gonzalo Saez, and Julien Bodart. A Comparative Study of Contrasting Machine Learning Frameworks Applied to RANS Modeling of Jets in Crossflow. volume Volume 2B: Turbomachinery of *Turbo Expo: Power for Land, Sea, and Air*, 06 2017. V02BT41A012.

[99] P. M. Wikström, S. Wallin, and A. V. Johansson. Derivation and investigation of a new explicit algebraic model for the passive scalar flux. *Physics of Fluids*, 12(3):688–702, 03 2000.

[100] David Wilcox. *Turbulence Modeling for CFD*. DCW Industries, 2006.

[101] Jin-Long Wu, Jian-Xun Wang, Heng Xiao, and Julia Ling. A Priori Assessment of Prediction Confidence for Data-Driven Turbulence Modeling. *Flow, Turbulence and Combustion*, 99(1):25–46, 2017.

[102] Jin-Long Wu, Heng Xiao, and Eric Paterson. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Phys. Rev. Fluids*, 3:074602, Jul 2018.

[103] X. Xiao, H. A. Hassan, J. R. Edwards, and R. L. Gaffney. Role of Turbulent Prandtl Numbers on Heat Flux at Hypersonic Mach Numbers. *AIAA Journal*, 45(4):806–813, 2007.

[104] Xiaowei Xu, Andrew Ooi, and Richard D. Sandberg. Data-driven algebraic models of the turbulent Prandtl number for buoyancy-affected flow near a vertical surface. *International Journal of Heat and Mass Transfer*, 179:121737, 2021.

[105] Chao Zhang, Lian Duan, and Meelan M. Choudhari. Direct Numerical Simulation Database for Supersonic and Hypersonic Turbulent Boundary Layers. *AIAA Journal*, 56(11):4297–4311, 2018.

[106] Xin-Lei Zhang, Heng Xiao, Xiaodong Luo, and Guowei He. Ensemble Kalman method for learning turbulence models from indirect observation data. *Journal of Fluid Mechanics*, 949:A26, 2022.

[107] Yaomin Zhao, Harshal D. Akolekar, Jack Weatheritt, Vittorio Michelassi, and Richard D. Sandberg. RANS turbulence model development using CFD-driven machine learning. *Journal of Computational Physics*, 411:109413, 2020.

This page intentionally left blank.

APPENDIX A. Deep ensembles for the turbulent Prandtl number

For modeling the ensemble Prandtl number correction described in section 6.4, we learn tensor-basis neural networks mapping input features to the output y^{NN} and a scalar variance σ^2,

$$\mathcal{NN} : (\boldsymbol{\eta}; \boldsymbol{\theta}) \mapsto (y^{NN}, \sigma^2)$$

where $\sigma^2 \in \mathbb{R}^+$ is a scalar variance and $\boldsymbol{\theta}$ denote the weights and biases. The ensemble approach trains M such networks. For each network, we optimize the log-likelihood as given by

$$\boldsymbol{\theta} = \underset{\boldsymbol{\theta}^* \in \mathbb{R}^{N_{\boldsymbol{\theta}}}}{\arg\min} \sum_{k=1}^{N_{\text{train}}} \frac{\left(y^{NN}\left(\boldsymbol{\eta}^k, \boldsymbol{\theta}^*\right) - y^k\right)^2}{2\sigma^2\left(\boldsymbol{\eta}^k, \boldsymbol{\theta}^*\right)} + \frac{\sigma^2\left(\boldsymbol{\eta}^k, \boldsymbol{\theta}^*\right)}{2},$$

where $N_{\boldsymbol{\theta}}$ denotes the total number of weights and biases. The mean and variance of the networks is then given by

$$\overline{y}^{NN} = \frac{1}{M}\left(\sum_{i=1}^{M} y^{NN}(\boldsymbol{\eta}; \boldsymbol{\theta}_i)\right),$$
$$\widehat{y}^{NN} = \frac{\hat{C}}{M}\sum_{i=1}^{M}\left(y^{NN^2}(\boldsymbol{\eta}; \boldsymbol{\theta}_i) - \overline{y}^{NN^2}\right) + \frac{1}{M}\sum_{i=1}^{M}\sigma^2\left(\boldsymbol{\eta}, \boldsymbol{\theta}_i\right),$$

where $\overline{y}^{NN} \in \mathbb{R}^3$ is output of the networks, $\widehat{y}^{NN} \in \mathbb{R}^+$ is a mean variance of the anisotropy tensors predicted by the networks, and $\hat{C} \in \mathbb{R}^+$ is a constant used to calibrate the ensemble variance. The constant is determined by the ratio of the network error the the variance of the ensemble as described in Chapter 4. Given a predicted mean and variance, a weight β is computed as

$$\beta = \frac{\sigma^2_{\text{prior}}}{\sigma^2_{\text{prior}} + \widehat{y}^{NN}} \tag{A.0.1}$$

Then the predicted log inverse Prandtl number is

$$y^{ML} = \beta\,\overline{y}^{NN} + (1-\beta)\log(1/0.9). \tag{A.0.2}$$

The approximation (A.0.1) is inspired by a Bayesian formulation and requires specification of a hyper-parameter for the prior confidence, σ^2_{prior}.

DISTRIBUTION

Email—Internal

Name	Org.	Sandia Email Address
CA Technical Library	8551	cateclib@sandia.gov

This page intentionally left blank.

Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.