



# CHORUS

This is the accepted manuscript made available via CHORUS. The article has been published as:

## Generalizable physics-constrained modeling using learning and inference assisted by feature-space engineering

Vishal Srivastava and Karthik Duraisamy

Phys. Rev. Fluids **6**, 124602 — Published 6 December 2021

DOI: [10.1103/PhysRevFluids.6.124602](https://doi.org/10.1103/PhysRevFluids.6.124602)

# Generalizable Physics-constrained Modeling using Learning and Inference assisted by Feature-space Engineering

Vishal Srivastava\*

*Ph.D. Candidate, Department of Aerospace Engineering,  
University of Michigan, Ann Arbor, 48105*

Karthik Duraisamy†

*Associate Professor, Department of Aerospace Engineering,  
University of Michigan, Ann Arbor, 48105*

(Dated: October 14, 2021)

## Abstract

This work presents a formalism to improve the predictive accuracy of physical models by learning *generalizable* model augmentations from sparse data. Building on recent advances in data-driven turbulence modeling, the present approach, referred to as Learning and Inference assisted by Feature-space Engineering (LIFE), is based on the hypothesis that robustness and generalizability demand a meticulously-designed feature space that is informed by the underlying physics, and a carefully constructed features-to-augmentation map. The critical components of this approach are: (1) Maintaining consistency across the learning and prediction environments to make the augmentation case-agnostic; (2) Tightly-coupled inference and learning by constraining the augmentation to be learnable throughout the inference process to avoid loss of inferred information (and hence accuracy); (3) Identification of relevant physics-informed features in appropriate functional forms to enable significant overlap in feature space for a wide variety of cases to promote generalizability; (4) Localized learning, i.e. maintaining explicit control over feature space to change the augmentation function behavior only in the vicinity of available datapoints. To demonstrate the viability of this approach, it is used in the modeling of bypass transition. The augmentation - in the form of model coefficient functions - is developed using skin friction data from two flat plate cases from the ERCOFTAC dataset. Piecewise linear interpolation on a structured grid in feature-space is used as a sample functional form for the augmentation to demonstrate the capability of localized learning. The impact of using a different function class (neural network) is also assessed. The augmented model is then applied to a variety of flat plate cases which are characterized by different freestream turbulence intensities, pressure gradients, and Reynolds numbers. The predictive capability of the augmented model is also tested on single-stage high-pressure-turbine cascade cases, and the model performance is analyzed from the perspective of information contained in the feature space. The results show consistent improvements across these cases, as long as the physical phenomena in question are well-represented in the training.

---

\*Electronic address: vsriv@umich.edu

†Electronic address: kdur@umich.edu

## I. INTRODUCTION

Traditional modeling of transition and turbulence has generally relied on theory and expert knowledge to determine the model-form, and on data for the calibration of resulting parameters. Recent advances [6, 9, 49] have explored the possibility of using data to introduce complex functional forms within existing models to improve their accuracy and applicability. In the context of turbulence modeling, Oliver and Moser [26] were among the first to quantify model-form uncertainties by introducing discrepancies in the Reynolds stresses as spatially-dependent Gaussian random fields. Dow and Wang [5] applied a similar approach to introduce discrepancies in the eddy viscosity field, and assimilated data from multiple flow fields. These precursors to the current machine learning approaches in RANS modeling are relevant as they treat model inaccuracies<sup>1</sup> as fields instead of parameters.

Tracey et al., [41] in 2013, proposed the idea to transform such inaccuracies from a field variable (a function of space) to a function of features (local functions of states and modeled quantities). This was applied to learn the discrepancies in the eigenvalues of the Reynolds anisotropy tensor between RANS and DNS results. A kernel regression ML model used the following features - eigenvalues of the anisotropy tensor, ratio of production-to-dissipation rate of turbulent kinetic energy and a marker function to mask the free shear layer regions in the flow. Xiao and co-workers [46, 48, 50, 51] modeled the discrepancy in not just the eigenvalues of the anisotropy tensor but also its eigenvectors and turbulent kinetic energy while using a broader feature set. Since the values of features and the inaccuracy term used to train the ML models are extracted from the DNS data, this approach is inherently inconsistent and unsuitable for predictive use via the baseline RANS models. In addition to this, the requirement of full field DNS data to create such ML models takes away from the robustness and versatility of this method.

Ling and Templeton [18] introduced Tensor Basis Neural Networks (TBNNs) to learn the coefficients of a Tensor Basis expansion (w.r.t. mean strain rate and vorticity tensors) intended to approximate the Reynolds stress term. The features used for this work included five invariants based on the mean strain rate and vorticity. TBNNs have, since, been used to model turbulent heat and scalar fluxes [24, 25].

---

<sup>1</sup> These are referred to as model inadequacies in the computational science community

Another class of approaches, which attempts to optimize the *functional form* of the feature-to-augmentation map, is symbolic regression. Weatheritt and Sandberg [47] employed genetic programming to construct models for Reynolds stress anisotropy in terms of invariants of the velocity gradient tensor. Schmelzer et al. [30] used sparse regression over a library of candidate functions to obtain an algebraic Reynolds stress model. An advantage of such approaches is that the resulting model form is simple, and presents the added benefit of interpretability.

Techniques to extract model inaccuracy mentioned above do not explicitly enforce model consistency, i.e. the inaccuracy information embedded in the modified model is extracted from higher-fidelity data *directly* and can hence be inconsistent with the lower-fidelity model. The importance of model consistency is discussed in a recent review [6]. To address model consistency, Duraisamy and co-workers [27, 32, 34, 42] introduced the field inversion and machine learning framework which aims to learn the inaccuracy in the same environment as used for predictive purposes. To achieve this, one needs to solve an additional inverse problem prior to the machine learning step to obtain a model-consistent inaccuracy field which is optimal in the sense that the corresponding outputs from the model match the target high-fidelity data as accurately as possible. While this framework mitigates inconsistency, it also facilitates the use of sparse high-fidelity data from experiments in contrast to full fields of direct numerical simulation data. To solve this so-called *field inversion* problem, an adjoint-driven gradient-based optimization technique is used. The following subsection focuses on the developments and applications that this framework has witnessed since its inception. Other techniques and frameworks introduced later (e.g. [19, 40, 53]) also suggest the use of model-consistent training, and present opportunities of bypassing the requirement for adjoint-driven gradient computations by pursuing weakly-coupled inference.

The original formulation of FIML, hereafter referred to as the classic FIML, has been used by several research groups, with applications including, but not limited to, predictive modeling of adverse pressure gradients flows [20, 33], separated flows [13–15, 34], bypass transition modeling [8], natural transition modeling [52], hypersonic aerothermal prediction for aerothermoelastic analysis [44], turbomachinery flows [11], shock-turbulent boundary layer interactions [31], etc. Matai et al. [20] proposed a zonal version of FIML, where the augmentation field obtained from field inversion was quantized into a set number of clusters, following which a decision tree based architecture was used to classify corresponding features

into appropriate clusters. The quantization reduced the number of classes that needed to be accounted for by the learning architecture and hence improved the training performance across a wide range of flow problems.

A significant evolution in the framework was proposed by Holland et al. [13, 14] in the context of RANS, and Sirignano et al. [35] in the context of LES. These techniques integrate the learning step in the inference process. This is an improvement over classic FIML for two reasons - (1) the augmentation fields obtained using classic FIML may be non-unique which means that two near-optimal augmentation fields might have completely different functional representations in the feature space which might result in convergence problems when learning augmentations from several field inversion problems simultaneously; and (2) significant information loss might occur when learning an augmentation field (relatively high-dimensional vector) to obtain ML-model parameters (relatively low-dimensional vector) which might result in a considerable loss in predictive accuracy. This tight coupling between the two inverse problems, however, requires additional infrastructure for gradient evaluation, and presents convergence challenges due to the additional non-linearity of the learning algorithm. This framework is, hereafter, referred to as integrated inference and learning.

In this work, we introduce another step in this evolution, viz. Learning and Inference assisted by Feature-space Engineering (LIFE). The approach is developed on the hypothesis that robustness and generalizability demand a low-dimensional feature space that is informed by the underlying physics, and that the features-to-augmentation map has to be carefully constructed. LIFE uses tightly coupled inference and learning to use data in order to locally update regions of feature space and can potentially make the augmented model more generalizable, robust and reliable by inherently suppressing spurious behavior and offering the modeler more control over the feature space. Note that, spurious behaviour, in the context of this work, refers to the augmented model predicting non-baseline behaviour by virtue of extrapolation to physical conditions which have not been encountered during training. In addition, this work also introduces new constructs (guiding principles and techniques) across different stages in the inference process that contribute towards boosting the efficacy and generalizability of the resulting augmentation while underlining the importance of human intuition and expert knowledge in the process.

## A. RANS models for bypass transition

To demonstrate the capability of LIFE, we choose the problem of developing a data-driven bypass transition model. Transition modeling has been studied over the past two decades, and several models are in existence. In this work, however, the aim is not to create a universally accurate transition model. Rather, the goal is to develop and demonstrate LIFE and examine the possibility of generalization in the low data limit using bypass transition as an example.

The phenomenon of laminar-to-turbulent transition is of vital importance in the design of aerospace vehicles and energy systems. Transition can occur either by the amplification and non-linear interactions of 2-D instability waves (also known as Tollmien-Schlichting waves) or via the perturbation of an otherwise laminar boundary layer by freestream turbulence, surface roughness, impinging wakes, etc. The latter is referred to as the bypass mode of transition, and is typically the preferred route for transition when freestream turbulence intensities are more than 1% [21]. While Reynolds-Averaged Navier-Stokes (RANS) simulations have remained the workhorse in the industry for design and optimization and a subject of active research for decades, bypass transition models for RANS turbulence closures are relatively in their infancy.

Two major categories of approaches exist in the literature to model bypass transition induced by freestream turbulence, viz. (1) Data correlation based models, where the model predicts the transition onset location based on some criteria (usually a correlation between momentum thickness Reynolds number  $Re_{\theta t}$  and freestream turbulence intensity  $Tu_{\infty}$ ) and switches from laminar to turbulent computation [1, 4, 21, 28]; and (2) transport equation based models, where the turbulence model is modified by introducing a new transport scalar, which is solved for using an additional equation. Further, there are two major approaches within transport equation-based models - laminar fluctuation models and intermittency function models: a) Laminar fluctuation models are based on the modeling of the so-called laminar kinetic energy,  $k_L$ , i.e. the energy of fluctuations in laminar boundary layers corresponding to Klebanoff modes; and the transfer of energy between the laminar disturbances and turbulent eddies by introducing a transfer term in the transport equations of  $k_L$  and  $k$  (modeled turbulent kinetic energy) [17, 22, 45]; b) Intermittency function models, on the other hand, model the intermittency  $\gamma$  (which is a statistical quantity referring to the fraction of the time

the flow is turbulent and varies from 0 in laminar flow to 1 in fully turbulent flow) which is usually multiplied to the production term in the transport equation for  $k$  or directly to the eddy viscosity, thus, attenuating the production of turbulence in the regions supposed to be laminar or transitional [3, 10, 12, 16, 23, 36, 38, 39].

In this work, we utilize the idea of LIFE to create a data-driven intermittency-based bypass transition model using data from only two flat plate cases, and predict the bypass transition triggered by freestream turbulence across different flow conditions and geometries.

The outline of this paper is as follows: The formal problem description of model augmentation, and a background of the Field Inversion and Machine Learning Framework is laid out in section II. In section III, the methodology of LIFE, which includes its philosophy, the required infrastructure and related algorithms is presented. Section IV begins with relevant background of the bypass transition modeling problem necessary for, and followed by, the description of baseline model-form used in this work, how the augmentation is introduced and what features are used to express the functional form of the augmentation. In section V, the training and testing results of the transition model augmentation is shown and analyzed. Section VI summarizes the work. The appendices details the impact of modeling choices on the results of the modeling augmentation.

## II. PROBLEM STATEMENT & BACKGROUND

### A. Premise

#### 1. The closure problem

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

$$\frac{\partial \mathbf{q}}{\partial t} + \mathcal{R}(\mathbf{q}) = 0 \quad \forall \quad \mathbf{x} \in \Omega \quad (1)$$

where  $\mathbf{q}(\mathbf{x}, t)$  represents the highly resolved spatio-temporal field of state variables. For practical problems, like flow over an aircraft wing or inside a gas turbine combustor, direct simulations of the underlying physics with the required spatio-temporal resolution are presently infeasible. In turbulence modeling, the state variables are decomposed into coarse-grained and unresolved parts, represented by  $\tilde{\mathbf{q}}$  and  $\hat{\mathbf{q}}$ , respectively. Performing the coarse-graining



operation on the equations representing the high-fidelity system results in an unknown closure term  $\mathcal{N}(\mathbf{q})$ , arising from any non-linearities in the operator  $\mathcal{R}$  and the unaccounted contribution from  $\widehat{\mathbf{q}}$ . Note that the following representation is not an approximation.

$$\frac{\partial \widetilde{\mathbf{q}}}{\partial t} + \widetilde{\mathcal{R}}(\widetilde{\mathbf{q}}) = 0 \quad \Rightarrow \quad \frac{\partial \widetilde{\mathbf{q}}}{\partial t} + \mathcal{R}(\widetilde{\mathbf{q}}) + \mathcal{N}(\mathbf{q}) = 0 \quad (2)$$

In a generic sense, a closure model can be written as  $\mathcal{N}_m(\widetilde{\mathbf{q}}_m, \widetilde{\mathbf{s}}_m)$ , where  $\widetilde{\mathbf{q}}_m$  refers to the state variables for the reduced-fidelity model which will be different from the coarse-grained variables  $\widetilde{\mathbf{q}}$  because of a *modeled* closure term.  $\widetilde{\mathbf{s}}_m$  represents the vector of any secondary variables (e.g. turbulent kinetic energy and other scale providing variables) which might need to be solved for by using an additional system of equations as shown below.

$$\frac{\partial \widetilde{\mathbf{s}}_m}{\partial t} + \mathcal{G}_m(\widetilde{\mathbf{s}}_m, \widetilde{\mathbf{q}}_m) = 0 \quad \forall \quad \mathbf{x} \in \widetilde{\Omega} \quad (3)$$

where  $\widetilde{\Omega}$  represents the appropriately discretized version of the original domain  $\Omega$ . In this work, the focus is restricted to steady-state reduced-fidelity models, which, using the terminology described above, can be represented as

$$\mathcal{R}(\widetilde{\mathbf{q}}_m) + \mathcal{N}_m(\widetilde{\mathbf{q}}_m, \widetilde{\mathbf{s}}_m) = 0 \quad ; \quad \mathcal{G}_m(\widetilde{\mathbf{s}}_m, \widetilde{\mathbf{q}}_m) = 0 \quad (4)$$

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

## 2. A surrogate measure of model inaccuracy

Reduced-fidelity models of turbulence and transition can be insufficiently accurate for use in predictive simulations that aid design and optimization owing to model-form inaccuracies in the representation of the operators  $\mathcal{N}_m$  and/or  $\mathcal{G}_m$ . To address these inaccuracies, one can “augment” the model which involves appropriately introducing an inaccuracy field,  $\beta(\mathbf{x})$  in the model-form of  $\mathcal{R}_m$ , giving rise to the following augmented model.

$$\mathcal{R}_m(\widetilde{\mathbf{u}}_m; \beta(\mathbf{x}), \boldsymbol{\xi}) = 0 \quad (5)$$

Examples of how such inaccuracy terms have been introduced into turbulence models in the available literature include multiplication with the production term of the Spalart-Allmaras

model by Singh et al. in [34], addition to the eigenvalues of the Reynolds anisotropy tensor by Xiao et al. [41, 51]. Indeed,  $\beta$  can represent the vector of coefficients in the tensor basis expansion for the Reynolds anisotropy tensor as proposed by Ling et al. [18].

### 3. Model consistency and data

The task of quantifying the model inaccuracy now reduces to inferring some ideal  $\beta(\mathbf{x})$  field, that when used in this augmented model, minimizes a chosen measure of discrepancy (e.g. L2 norm) between the data and corresponding predictions for quantities of interest. Note that, even if the full high-fidelity field of state variables is available, data alone should not be used to obtain  $\beta$  values by simply plugging values into the reduced-fidelity model at different locations, as this could result in a field which is inconsistent with the model and in a general case may lead to prediction inaccuracies [6]. In other words, the inferred  $\beta(\mathbf{x})$  field is only a surrogate measure of the inaccuracy, and that the true model inaccuracy *might* be slightly different than this field, as  $\tilde{\mathbf{q}}$  and particularly  $\tilde{\mathbf{s}}$  may be different from  $\tilde{\mathbf{q}}_m$  and  $\tilde{\mathbf{s}}_m$ , respectively.

For a dataset (a dataset in the present context refers to a unique combination of domain geometry and flow conditions) referred to by the index  $i$ , a discrepancy between the vector of high-fidelity data for these observables,  $\mathbf{y}^i$ , and the corresponding vector of predictions using the augmented model,  $\mathbf{y}_m^i(\tilde{\mathbf{u}}_m^i)$ , can be used to formulate the cost function for this case as  $\mathcal{C}^i(\mathbf{y}^i, \mathbf{y}_m^i(\tilde{\mathbf{u}}_m^i))$ . Since the field  $\tilde{\mathbf{u}}_m^i$  necessarily satisfies  $\mathcal{R}_m(\tilde{\mathbf{u}}_m^i; \beta^i, \boldsymbol{\xi}^i) = 0$ , it implicitly depends on the augmentation field,  $\beta^i(\mathbf{x})$ , and so, the cost function is ultimately a function of  $\beta^i(\mathbf{x})$ . Thus, a gradient-based optimization can, potentially, be used to infer  $\beta^i(\mathbf{x})$ . A rudimentary candidate for the cost function can be the square of  $L_2$ -norm of the difference between the two vectors. This process of inference will be explained in greater detail in the section II B.

### 4. Representing inaccuracies via augmentation functions

While the augmentation field,  $\beta(\mathbf{x})$ , provides information about the local inaccuracy in the model, it cannot be used for predictive improvements on its own. However, if a consistent functional relationship could be extracted between  $\beta(\mathbf{x})$  and some carefully-chosen features,

$\boldsymbol{\eta}(\tilde{\mathbf{u}}_m, \boldsymbol{\zeta})$ , this function could replace the  $\beta(\mathbf{x})$  field in the reduced-fidelity model equations for predictive use. Here  $\boldsymbol{\zeta}$  denotes local quantities independent of the state or secondary variables which are used to design features. Note that, for the augmentation to be usable in a predictive setting with a wide range of applicability, it should ideally have no dependence on the domain geometries or boundary conditions used for inference. The augmentation is assumed to be dependent only on local quantities in the present work as including non-local information in the features requires a more sophisticated, and potentially less general treatment. The features must be invariant [6, 48] to appropriate transformations (e.g. rotation, as they must not depend on the orientation of the coordinate axes.) Given the features-to-augmentation map, the model can be written as

$$\mathcal{R}_m(\tilde{\mathbf{u}}_m, \beta(\boldsymbol{\eta}(\tilde{\mathbf{u}}_m, \boldsymbol{\zeta}); \mathbf{w}); \boldsymbol{\xi}) = 0. \quad (6)$$

It should be mentioned here that, once the model class of the augmentation (e.g. a neural network) is chosen, the goal is to infer the parameters  $\mathbf{w}$ . The section II B focuses on variants of the FIML framework to obtain the parameters  $\mathbf{w}$  from sparse data.

## B. Field Inversion and Machine Learning (FIML)

The field inversion and machine learning framework, originally proposed by Duraisamy and co-workers, [27, 42], was formulated to reduce inaccuracies in a given reduced-fidelity model by inferring optimal augmentations such that the predictive accuracy of the model is improved. Point-wise and integrable data of variable sparsity across multiple cases can be used from experiments and higher-fidelity simulations. The two main variants of this framework are been discussed in the following sections.

### 1. Classic FIML

FIML seeks the optimal augmentation into two steps: Firstly, a “field inversion” problem is solved separately over each dataset to infer optimal augmentation fields,  $\beta(\mathbf{x})$ , in the respective discretized domains. Then, the step of machine learning uses these inferred augmentation values and respective features for all spatial locations across all datasets and infers the optimal augmentation function parameters  $\mathbf{w}$ . These two steps are described below.

a. *Field Inversion*

$$\begin{aligned} \beta^{*i}(\mathbf{x}) &= \arg \min_{\beta^i(\mathbf{x})} \{ \mathcal{C}^i(\mathbf{y}^i, \mathbf{y}_m^i(\tilde{\mathbf{u}}_m^i)) + \lambda_\beta^i \mathcal{T}_\beta^i(\beta^i(\mathbf{x})) \} \\ \text{s.t. } \mathcal{R}_m(\tilde{\mathbf{u}}_m^i; \beta^i(\mathbf{x}), \boldsymbol{\xi}^i) &= 0 \quad \forall \quad i = 1, 2, \dots, N \end{aligned} \quad (7)$$

Equation 7 represents the field inversion problem which is solved individually over each of the  $N$  training cases (cases for which available high-fidelity data is to be used to infer the augmentation function parameters) to obtain optimal augmentation fields  $\beta^{*i}(\mathbf{x})$  in the respective domains, where  $i$  is the case index. The augmentation field is optimal in the sense that it minimizes an objective function which consists of a cost function,  $\mathcal{C}^i$ , and a regularization term,  $\mathcal{T}_\beta^i$ , with  $\lambda_\beta^i$  as the regularization constant.

$$\mathcal{C}^i = \|\mathbf{y}^i - \mathbf{y}_m^i(\tilde{\mathbf{u}}_m^i)\|_2^2 \quad ; \quad \mathcal{T}^i = \|\beta^i(\mathbf{x}) - \beta_0\|_2^2 \quad (8)$$

Note here that an additive inaccuracy term which is non-dimensionalized with the same turbulent length and time scales as the source term, can also be viewed as a multiplicative inaccuracy term.

Given the very high-dimensional nature of the augmentation fields that need to be inferred, the field inversion problem is usually solved using a gradient-based optimization technique.

b. *Supervised Learning*

Once the field inversion problem is solved for all available datasets, the supervised learning problem can be solved using the augmentation fields and correspondingly calculated feature fields from all datasets to optimize for the function parameters  $\mathbf{w}$  as  $\mathbf{w} = \arg \min_{\mathbf{w}'} \mathcal{L}(\boldsymbol{\beta}^*, \beta(\boldsymbol{\eta}^*; \mathbf{w}'))$  where,  $\boldsymbol{\beta}^*$  refers to the stacked vector containing optimal augmentation fields across all datasets, and  $\boldsymbol{\eta}^*$  contains the respectively stacked feature values. Note that, while the inferred augmentation from the field inversion step is fully consistent with the underlying model, the augmentation field provided by the field inversion process is not necessarily learnable as a function of the chosen features [14]. This can lead to a loss of information extracted in the field inversion step. A simple measure of learnability can be defined as follows.

$$\text{Learnability} = 1.0 - \frac{1}{n} \sum_{i=0}^N \frac{|\beta(\boldsymbol{\eta}_i; \mathbf{w}^*) - \beta_i^*|}{|\beta(\boldsymbol{\eta}_i; \mathbf{w}^*)| + |\beta_i^*|}$$

where  $\mathbf{w}^*$  is the globally optimal set of parameters that minimizes the discrepancy between the augmentation values obtained using field inversion and corresponding machine learning predictions, and  $n$  is the number of datapoints. This measure can assume values between

0 and 1 such that higher value refers to higher learnability. As discussed previously, the augmentation field obtained from field inversion is not unique, it is possible that the field inversion results obtained from two different datasets correspond to different features-to-augmentation mappings. This inconsistency can also degrade the learnability of the model. The more recent variant of FIML, referred to as integrated inference and learning addresses these concerns.

## 2. Integrated Inference and Learning

The integrated inference and learning approach, proposed by Holland et al. [13, 14] and Sirignano et al. [35] replaces the two-step inference problem introduced in classic FIML by just one inverse problem, mentioned as follows

$$\begin{aligned} \mathbf{w} = \arg \min_{\mathbf{w}'} \bigsqcup_{i=1}^N (\mathcal{C}^i(\mathbf{y}^i, \mathbf{y}_m^i(\tilde{\mathbf{u}}_m^i)) + \lambda_{\beta}^i \mathcal{T}_{\beta}^i(\beta(\boldsymbol{\eta}^i(\tilde{\mathbf{u}}_m^i, \boldsymbol{\zeta}^i); \mathbf{w}')))) \\ \text{s.t. } \mathcal{R}_m(\tilde{\mathbf{u}}_m^i, \beta(\boldsymbol{\eta}(\tilde{\mathbf{u}}_m^i, \boldsymbol{\zeta}^i); \mathbf{w}'); \boldsymbol{\xi}^i) = 0 \quad \forall \quad i = 1, 2, \dots, N, \end{aligned} \quad (9)$$

where  $\bigsqcup$  is a generic assembly operator to build a composite objective function using the individual objective functions calculated for each dataset. It can be as simple as a sum, if all datasets are equally important for the inference, or it could be a weighted sum, if some datasets are to be assigned more importance than others, or it could be something even more complex as designed/needed by the user. Notice here that using several cases at once adds an implicit regularization to the problem. Similar to the field inversion problem, the required sensitivities can be obtained using discrete adjoints as described in Appendix A.

### *Benefits over classic FIML*

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

- Since integrated inference and learning can infer the function parameters  $\boldsymbol{w}$  while simultaneously assimilating data from multiple datasets, the augmentation fields for all datasets is constrained to be realizable w.r.t. the functional form of the augmentation as explained in the previous point. This has an added advantage that this procedure, unlike classic FIML, is by design prevented from learning augmentation fields from different datasets that behave differently in the feature space. In other words, consistency in the features-to-augmentation mapping across datasets is automatically enforced when using integrated inference and learning.
- Building on the previous point, this means that if we have even a handful of DNS(Direct Numerical Simulation) fields from which true augmentation values can be extracted (or are readily available), they can be used to enforce a near-physical relationship between the features and augmentation by simultaneously using a plethora of other sparse field data from experiments or higher fidelity simulations. **This is critical from the viewpoint of generalizable and physical augmentations.**

### *Limitations*

- Unlike the classic FIML approach, the task of designing an appropriate feature space must precede the inference from data. This, at times, could be difficult for the modeler and demonstrates the preliminary need of an independent field inversion step which can lend crucial information about the quantities correlated to the augmentation field that may then be used to formulate features.
- In its original form, integrated inference and learning does not consider the significance of localized learning. For practical problems, the data being used might not populate the entire feature space, which means that if the optimization problem is not constrained to change the augmentation only in the vicinity of the feature space locations for which data is available, it might lead to spurious predictions in other regions, which might not only result in worse accuracy compared to the baseline model but also severely affect the stability of the numerical solver.
- When using complex learning algorithms such as neural networks or decision trees, the augmented model inherits non-linearities from the baseline model **and** the learning

algorithm. This, combined with the previous point, can lead to a disorderly optimization behavior when solving the inference problem, in addition to the aforementioned potential deterioration in accuracy and/or numerical stability with every successive optimization iteration.

### III. MODEL-CONSISTENT LEARNING AND INFERENCE ASSISTED BY FEATURE-SPACE ENGINEERING (LIFE)

To alleviate the limitations in integrated inference and learning, and to make the augmentation robust and generalizable we present a set of guiding principles to choose features along with the notion of localized learning in the feature space which requires controlling how learning takes place in different parts of the feature space. Since a significant effort is put into how features are chosen and how the learning takes place in different parts of the feature space, we call this version of integrated inference and learning as “Learning and Inference assisted by Feature-space Engineering (LIFE)”.

This section is structured as follows. Section III A and III B propose guiding principles to take into consideration when designing the augmentation and features. Section III C deals with the notion of localized learning and provides three different ideas on how to make it work. Finally, III D deals with the algorithm and practical concerns of implementing the LIFE framework.

#### A. How should a model augmentation be designed?

The following aspects could be considered to decide how to introduce an augmentation in the baseline model:

1. **Efficacy:** The augmentation should be capable of effectively addressing the inaccuracy in consideration. For instance, augmenting the coefficient of an isotropic diffusion term in a model cannot correct for an inaccuracy caused as a consequence of the anisotropically diffusive behaviour in the corresponding true system.
2. **Spurious behavior:** The augmentation should be introduced such that it does not corrupt model behavior in regions where the inaccuracy under consideration is not a concern. For instance, a RANS model augmented to predict transition should not change the model

behavior outside the boundary layer or in regions inside the boundary layer in which the flow is fully turbulent.

**3. Sensitivity:** In practice, the predicted augmentation values might have small errors which can have significant adverse effects on the stability and accuracy of the model if it is highly sensitive to the augmentation. Hence, the augmentation should be introduced in a manner that mitigates this as much as possible.

## **B. What features should the augmentation depend on?**

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

### **1. Choice of features using expert knowledge**

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

### **2. Physics-based non-dimensionalization**

Once the quantities to be used as features are chosen, they must be non-dimensionalized in order to be generalizably used for prediction. This is because similar physical phenomena can occur for significantly different magnitudes of dimensional model quantities and, since the LIFE framework aims at using as little data as possible to discover generalizable augmentation functions, properly non-dimensionalization of the features becomes imperative. While statistics from the training datasets can be used to non-dimensionalize features, the



available data might not be sufficiently representative of the complete range of values that could be encountered during prediction on an unseen case. Thus, it is better to make use of model quantities to non-dimensionalize features.

A simple example can be used to illustrate this point. Let us consider that the eddy viscosity  $\nu_t$  is intended for use as a feature. Technically speaking,  $0 \leq \nu_t < \infty$ . With limited data, it seems hopeless that the value of  $\nu_t$  can be used as a feature in a generic predictive setting. However, simply using the turbulent Reynolds number  $\nu_t/\nu$  can immediately improve the quality of results. This is because we take some burden of discerning physical conditions for a given combination of features off of the learning algorithm and make the integrated inference and learning problem better-posed. Note that one can also non-dimensionalize  $\nu_t$  as  $\nu_t|\Omega|/|\mathbf{U}|^2$ , but the physical conditions represented here will be different than the ones represented by  $\nu_t/\nu$ . Which non-dimensionalization is chosen depends on its relevance w.r.t. the inaccuracy to be alleviated. While the example shown here seems trivial, physics-based non-dimensionalization can prove to be a challenge in some scenarios which might require domain-specific theoretical knowledge and/or experimentally obtained correlations to resolve.

### 3. Effectively bounded feature-space

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

### 4. Appropriate functional forms for features

Several functional forms of a non-dimensionalized feature can offer effective boundedness. But, only a few of these forms might address the inaccuracy in a major part of the feature space. For instance, the part of the feature space covered by the feature  $\nu/(\nu_t + \nu)$  between the values of 0 and 0.3 reduces to being covered between the values 0 and 0.000729 for a

different form of the same feature, viz.  $(\nu/(\nu_t + \nu))^6$ . Note that, both features have the same mathematical range between 0 and 1. However, by the virtue of different functional forms, the regions denoting the same physical conditions span very differently in the feature space. For an augmentation which is predominantly affected within the said range, the feature  $\nu/(\nu_t + \nu)$  offers a better-conditioned learning problem. Hence, the choice of which among different effectively bounded non-dimensional functional forms of a feature to use can play a significant role in setting up a well-conditioned learning problem.

### **5. Parsimonious combination of features vs. one-to-one features-to-augmentation mapping**

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

### C. Which class of functions should be used to model this augmentation?

Depending on the complexity of the behavior of the augmentation in feature space, there exist different alternatives to choose the functional form of the augmentation prior to the inference and learning procedure. If the behavior is very simple and the feature space is very low-dimensional, one can hand-fit a function on the inferred set of augmentation values. If the behavior is not intractably complex, a user-defined functional form can be used. Lastly, if the behavior is highly non-linear, a class of functions available in the machine learning literature (decision trees, neural networks, etc.) could be used.

This choice becomes even more important when the available data does not span/represent all parts of the effectively bounded feature space. In such a scenario the augmentation function might extrapolate to predict values for feature space regions corresponding to which no data was available. This could lead to spurious model behavior and could make the prediction capabilities of the augmented model worse than its baseline counterpart. To preempt this complication the idea of localized learning can be used.

**Localized Learning**, here, refers to the class of learning techniques that modifies the augmentation function behavior only in the vicinity of available data without perturbing augmentation values in regions far from available datapoints. Thus, the augmentation value at any point in the feature space could be either “influenced” by one or more datapoints, or remain “uninfluenced”. Three different ways to perform localized learning are mentioned as follows:

1. **Discretizing the feature space** into subdomains and representing the augmentation function as a piecewise combination of local augmentation functions corresponding to each of these subdomains. Here, a datapoint influences only those points in the feature space that lie in its own subdomain.
2. **Introducing artificial datapoints** in the learning process by sampling from the existing augmentation field in regions where no datapoints are available. Here, the sampling strategy determine the influenced regions in the feature space.
3. **Superposing finite support functions** (e.g., truncated Gaussians) centered at available datapoints can be added on top of the existing augmentation to update it. Here the support width of these individual functions characterize the influence of

the datapoints in the feature space.

Choosing how fine/coarse the discretization is (in case of method 1), or the strategy used to sample points from existing augmentation field (in case of method 2), or the support width (in case of method 3) in order to control the influence of datapoints in the feature space requires the user to make a trade-off. If, for a given datapoint, the region of influence is very small, the augmentation would behave as an over-fitted function which, while performing accurately on the training cases, will not be able to generalize well. On the other hand, if the region of influence is too large, the augmentation will be over-regularized and the predictive accuracy will suffer, again resulting in a loss of generalizability.

Localized learning forces the augmentation function to retain the baseline behavior for feature space regions which remain uninfluenced throughout the LIFE process. The certainty that the augmented model reverts back to its baseline behavior when faced with feature combinations (physical conditions) not encountered during the LIFE process means that the performance of the augmented model would always be better than or equal to the baseline version.

Also note that, for steady-state models like RANS, the feature values obtained in one solver iteration influence the feature values obtained in the next iteration. The inference and learning process, however, works with the feature values obtained at solver convergence only. Hence, uninfluenced feature space regions (according to the converged solutions) could be accessed by the solver when the residuals are not converged. So, if the augmentation function is modified in these regions, it could consequently affect the converged result, which is undesirable. Thus, it is even more important that in uninfluenced feature space regions, the augmentation function does not learn from data, and rather predict from the unaltered existing augmentation behavior.

Clearly, there exists ample scope to explore different localized learning techniques (along with strategies to optimally choose regions of influence). However, our main motive in this work w.r.t. localized learning is to introduce it in the LIFE framework and emphasize on its importance in creating robust and generalizable augmentations, and so we choose to deal with these explorations in future works.

---

**Algorithm 1** Simultaneous Integrated Inference and Learning

---

**Inputs:**  $\mathbf{w}^0, \mathbf{y}^1, \dots, \mathbf{y}^N, n, \alpha$

**Outputs:**  $\mathbf{w}^n$

```

1: for  $i = 1 : N$  do [ $N$  is number of datasets]
2:   Solve for  $\tilde{\mathbf{u}}_m^i$  s.t.  $\mathcal{R}_m(\tilde{\mathbf{u}}_m^i; \beta(\boldsymbol{\eta}(\tilde{\mathbf{u}}_m^i, \boldsymbol{\zeta}^i); \mathbf{w}^0), \boldsymbol{\xi}^i) = 0$  [Forward problem]
3:    $\mathcal{J}^i := \mathcal{C}^i(\mathbf{y}^i, \mathbf{y}_m^i(\tilde{\mathbf{u}}_m^i)) + \lambda_\beta^i \mathcal{T}_\beta^i(\beta(\boldsymbol{\eta}(\tilde{\mathbf{u}}_m^i, \boldsymbol{\zeta}^i); \mathbf{w}^0))$  [Objective evaluation]
4: end for
5:  $\mathcal{J} := \bigsqcup_{i=1}^N \mathcal{J}^i$  [Assembly operator (e.g., weighted sum)]
6: for  $k = 1 : n$  do [ $n$  is number of LIFE iterations]
7:   for  $i = 1 : N$  do
8:      $\boldsymbol{\psi}^T := \frac{d\mathcal{C}^i}{d\tilde{\mathbf{u}}_m^i} \left[ \frac{d\mathcal{R}_m}{d\tilde{\mathbf{u}}_m} \Big|_{\boldsymbol{\xi}^i} \right]^{-1}$  where  $\frac{d}{d\tilde{\mathbf{u}}_m^i} = \frac{\partial}{\partial \tilde{\mathbf{u}}_m^i} \Big|_\beta + \frac{\partial}{\partial \beta} \frac{\partial \beta}{\partial \boldsymbol{\eta}} \frac{\partial \boldsymbol{\eta}}{\partial \tilde{\mathbf{u}}_m^i}$  [Adjoints]
9:      $\frac{d\mathcal{J}^i}{d\mathbf{w}^{k-1}} := \mathbf{0}$  [Set sensitivities to zero]
10:    for all  $j$  s.t.  $x_j \in \tilde{\Omega}^i$  do
11:       $\frac{d\mathcal{J}^i}{d\mathbf{w}^{k-1}} = \frac{d\mathcal{J}^i}{d\mathbf{w}^{k-1}} + \lambda_\beta^i \frac{\partial \mathcal{T}_\beta^i}{\partial \beta} \left[ \frac{\partial \beta}{\partial \mathbf{w}^{k-1}} \right]_j^i - \boldsymbol{\psi}^T \left[ \frac{\partial \mathcal{R}_m}{\partial \beta} \right]^i \left[ \frac{\partial \beta}{\partial \mathbf{w}^{k-1}} \right]_j^i$ 
12:    end for
13:  end for
14:   $\frac{d\mathcal{J}}{d\mathbf{w}^{k-1}} := \sum_{i=1}^N \frac{\partial \mathcal{J}}{\partial \mathcal{J}^i} \frac{d\mathcal{J}^i}{d\mathbf{w}^{k-1}}$  [Sensitivity assimilation from all cases]
15:   $\mathbf{w}^k = \mathbf{w}^{k-1} - \alpha \frac{d\mathcal{J}}{d\mathbf{w}^{k-1}} \left\| \frac{d\mathcal{J}}{d\mathbf{w}^{k-1}} \right\|_\infty^{-1}$  [Steepest gradient descent update]
16:  for  $i = 1 : N$  do
17:    Solve for  $\tilde{\mathbf{u}}_m^i$  s.t.  $\mathcal{R}_m(\tilde{\mathbf{u}}_m^i, \beta(\boldsymbol{\eta}(\tilde{\mathbf{u}}_m^i, \boldsymbol{\zeta}^i); \mathbf{w}^k)); \boldsymbol{\xi}^i) = 0$  [Forward problem]
18:     $\mathcal{J}^i = \mathcal{C}^i(\mathbf{y}^i, \mathbf{y}_m^i(\tilde{\mathbf{u}}_m^i)) + \lambda_\beta^i \mathcal{T}_\beta^i(\beta(\boldsymbol{\eta}(\tilde{\mathbf{u}}_m^i, \boldsymbol{\zeta}^i); \mathbf{w}^k))$  [Objective evaluation]
19:  end for
20:   $\mathcal{J} = \bigsqcup_{i=1}^N \mathcal{J}^i$ 
21: end for

```

---

#### D. Implementation details

To be able to evaluate different functional forms for the augmentation, the solver has to be made agnostic of how the augmentation is implemented. To do so, the solver must interact

with only a linearized version of the augmentation function,  $\beta_\ell$ . In other words, given feature values  $\boldsymbol{\eta}_0$ , an implementation of the augmentation function should internally compute the numerical values  $\beta_0 = \beta(\boldsymbol{\eta}_0; \boldsymbol{w})$  and  $\boldsymbol{g}_\beta = \left. \frac{\partial \beta}{\partial \boldsymbol{\eta}} \right|_{\boldsymbol{\eta}_0}$ . These numerical values should then be passed to the solver which should construct the linearized augmentation using AD variables  $\boldsymbol{\eta}$  and non-AD type numerical values for  $\boldsymbol{\eta}_0$  as  $\beta_\ell(\boldsymbol{\eta}) = \beta_0 + \boldsymbol{g}_\beta(\boldsymbol{\eta} - \boldsymbol{\eta}_0)$ . Notice here, that the value of  $\beta_\ell$  at  $\boldsymbol{\eta} = \boldsymbol{\eta}_0$  (which is the value the direct solver uses in the model) remains  $\beta_0$  and the second term involving  $\boldsymbol{g}_\beta$  is present only to record  $\left. \frac{\partial \beta}{\partial \boldsymbol{\eta}} \right|_{\boldsymbol{\eta}_0}$  in the AD tape. Notice that this also presents the advantage of making the AD tape shorter.

The detailed algorithm for the LIFE formulation is presented in Algorithm 1. The jacobians presented in the algorithm can be calculated using hard-coded analytic derivatives or using an automatic differentiation (AD) package. In the transition modeling demonstration presented in this work (section IV), the in-house solver makes use of the open-source automatic differentiation package ADOL-C to calculate these jacobians.

## IV. AUGMENTING A BYPASS TRANSITION MODEL

### A. Relevant ideas from bypass transition modeling literature

#### 1. The Praisner-Clark estimate of transition momentum-thickness Reynolds number

Praisner and Clark [28], in 2004, presented a correlation that can approximate the transition momentum thickness Reynolds number over a wide range of pressure gradients, freestream turbulence intensities and Mach numbers. This correlation is given as follows.

$$Re_{\theta,t} = A \left( Tu_\infty \frac{\theta_t}{\lambda_\infty} \right)^B \quad (10)$$

where  $A = 8.52$  and  $B = -0.956$ . Approximating  $B$  to be -1, recasting this correlation and assuming  $C_\mu \omega = u'/\lambda$  for the Wilcox  $k - \omega$  model one can write this correlation as

$$\theta_t^2 = A_1 \frac{\nu \lambda_\infty}{u'_\infty} = \frac{A_1}{C_\mu} \frac{\nu}{\omega_\infty} \quad (11)$$

where  $A_1 = 0.07 \pm 0.011$ . The universality of this estimate across such a varied set of flow conditions, even though the dataset comes predominantly from turbomachinery setups, is adopted in the model presented in this work.

## 2. A brief review of recent intermittency-based models

Langtry and Menter [16] proposed one of the most widely used intermittency-based transition models with transport equations for intermittency ( $\gamma$ ) and an estimate of transition momentum thickness Reynolds number ( $\widetilde{Re_{\theta,t}}$ ) to supplement the SST turbulence model equations. The additional transport equation for  $\widetilde{Re_{\theta,t}}$  is used to diffuse the values calculated using algebraic correlations from outside the boundary layer to regions inside the boundary layer. This quantity is then used to evaluate the critical momentum thickness Reynolds number,  $Re_{\theta_c}$ , which when compared with the vorticity Reynolds number,  $d^2\Omega/\nu$ , can be used to predict the transition onset and hence trigger intermittency production. The intermittency takes effect in the solver by scaling the production term of the TKE transport equation.

In 2012, Durbin [10] proposed a simple transition model with a single production term which is directly proportional to the local vorticity,  $\Omega$  and  $(\gamma_{\max} - \gamma)\sqrt{\gamma}$ , where  $\gamma_{\max} = 1.1$ , to supplement the Wilcox's 1988  $k - \omega$  turbulence model. The  $(\gamma_{\max} - \gamma)$  term would give a trivial solution of  $\gamma = \gamma_{\max}$  in the entire field if the model is used as it is. For this reason, the author proposed to set the intermittency to zero for all mesh locations where the eddy viscosity was very small compared to the local molecular viscosity and the vorticity Reynolds number is less than a pre-defined threshold, after every solver iteration. After every solver iteration, the values of  $\gamma$  is set as  $\min(\gamma, 1.0)$  as  $\gamma_{\max} > 1$  helps the intermittency to rapidly increase, but at the same time, results in a non-physical growth in turbulent production. As in the Langtry-Menter model, the intermittency is multiplied to the production term in the transport equation for  $k$ .

Ge and Durbin [12] proposed another transition model based on the structure of the previous model in 2014 which attempts to predict bypass transition due to freestream turbulence and flow separation. Separation effects were accounted for by multiplying the production term in the transport equation for  $k$  with a separation modification. It is noteworthy, here, that the presence of a dissipation term in this equation removes the requirement of setting the intermittency to zero, externally.

In the models mentioned above, one can see a common theme. All of these models attenuate the production of turbulent kinetic energy using the intermittency function rather than attenuating the eddy viscosity directly. This is justified as turbulent kinetic energy should

be very low in the laminar regions of the flow, something which is hard to implement by just altering the eddy viscosity term. Secondly, one of the most important terms in all these models is the vorticity Reynolds number, the maximum value of which approximates the momentum thickness Reynolds number fairly well for a zero pressure gradient boundary layer, as shown by van Driest and Blumer [43]. Even in the presence of favorable/adverse pressure gradients, the approximation still provides an excellent tool for modeling the transition onset as a function of a local quantity, as opposed to the momentum thickness Reynolds number which is an integral quantity. These ideas were instrumental in designing the transition model presented in this work.

## B. Chosen baseline model and proposed augmentation

The transport equations that constitute the chosen baseline model have been shown below. This model is a variation of the model proposed by Durbin in 2012 [10] which used Wilcox's 1988  $k$ - $\omega$  model as the underlying turbulence model. The source term for the intermittency transport equation in Durbin's model is expressed as  $F_\gamma|\Omega|(\gamma_{\max} - \gamma)\sqrt{\gamma}$ , where  $|\Omega|$  is the local vorticity magnitude,  $\gamma_{\max} = 1.1$  and  $F_\gamma$  is a piecewise linear function that modulates the source term. Instead, we consider the baseline source term as  $\rho(1.0 - \gamma)\sqrt{\gamma}|\Omega|$  (the additional  $\rho$  is attributed to the compressible formulation of the model). The version of Wilcox's 1988 turbulence model containing a vorticity-based production term is used for this work, in contrast to the strain-rate based production term.

$$\frac{D\rho k}{Dt} = \nabla \cdot \left( \left( \mu + \frac{\mu_T}{\sigma_k} \right) \nabla k \right) + \mu_t \Omega^2 - \frac{2}{3} \rho k \nabla \cdot \mathbf{u} - C_{\mu} k \omega \quad (12)$$

$$\frac{D\rho \omega}{Dt} = \nabla \cdot \left( \left( \mu + \frac{\mu_T}{\sigma_\omega} \right) \nabla \omega \right) + C_{\omega 1} \frac{\omega}{k} \left( \mu_t \Omega^2 - \frac{2}{3} \rho k \nabla \cdot \mathbf{u} \right) - C_{\omega 2} \omega^2 \quad (13)$$

$$\frac{D\rho \gamma}{Dt} = \nabla \cdot \left( \left( \frac{\mu}{\sigma_l} + \frac{\mu_T}{\sigma_\gamma} \right) \nabla \gamma \right) + \rho(1.0 - \gamma)\sqrt{\gamma}|\Omega|, \quad (14)$$

with the eddy viscosity given by  $\mu_t = \rho k / \omega$ . Clearly, this model will give a fully turbulent solution, as the intermittency equation has a trivial solution with the value 1.0 throughout the domain. Multiplying the value 1.0 in the intermittency transport equation with the augmentation  $\beta$ , we have an augmented intermittency transport equation as follows

$$\frac{D\rho \gamma}{Dt} = \nabla \cdot \left( \left( \frac{\mu}{\sigma_l} + \frac{\mu_T}{\sigma_\gamma} \right) \nabla \gamma \right) + \rho(\beta - \gamma)\sqrt{\gamma}|\Omega|. \quad (15)$$



This particular form of baseline model and augmentation was chosen for the following reasons:

- For most cases, the range for this augmentation is bounded between 0 and 1.
- Any discontinuities in the predicted augmentation field will be smoothed over by the intermittency transport equation and will not require any special attention. While augmenting the production term in the  $k$ -equation directly would produce similar results, using the intermittency transport equation, in general, exhibited slightly better residual convergence characteristics (solver stability) and hence was retained in the formulation.
- The knowledge of correlations between the intermittency field and other flow quantities can be readily used to formulate this augmentation, as the converged intermittency field would closely resemble a smoothed augmentation field.

### C. Choice of features and limiters for the augmentation

A clear understanding of the effect of intermittency term  $\gamma$  on the modeled turbulent kinetic energy  $k$  played a vital role in the choice of features. A brief explanation of this effect is given as follows.

From equation 12, it can be observed that the production of  $k$  is directly proportional to eddy viscosity ( $\nu_t$ ) and vorticity ( $\Omega$ ), and inversely proportional to the modeled specific dissipation ( $\omega$ ).  $\Omega$ , for external flows, is usually the highest at the wall and decreases away from the wall. Also, the freestream eddy viscosity (which is usually the initial condition for  $\nu_t$  in RANS solvers) is very small (around an order of magnitude higher than molecular viscosity). Hence, the production of  $k$  starts in a region very close to the wall at a distance where  $\omega$  has sufficiently decreased and  $\Omega$  remains high enough to enable a net production of  $k$  (for a fully turbulent boundary layer, this region lies within the inner layer but outside the viscous sublayer). By the virtue of diffusion,  $k$  starts increasing slightly farther from the wall relative to where it is being produced, thus increasing local  $\nu_t$  and triggering a net production of  $k$ . This process keeps spreading away from the wall until  $\Omega$  drops so low that a net production is no longer possible. Also, the convective term in equation 12 ensures that a downstream production of  $k$  has minimal effect on upstream locations. Using all this

information, it can be concluded that in order to laminarize the upstream of the transition location, the production of  $k$  needs to be attenuated by sufficiently reducing  $\gamma$  only in a region close to the wall. Doing this requires - (1) feature(s) that can indicate the transition onset (the location up to which the augmentation should be effective); (2) feature(s) to uniquely identify the regions close to the wall where the augmentation should be active. The following sections describe the rationale behind the choice of three such features.

**Feature 1: Ratio of Vorticity Reynolds Number and an estimate of  $Re_{\theta,t}$**

Traditional models have used experimentally obtained correlations for transition momentum thickness Reynolds number ( $Re_{\theta,t}$ ) to predict transition by comparing it with the local momentum thickness Reynolds number  $Re_{\theta}$ . As mentioned in section IV A 2, the vorticity Reynolds number,  $Re_{\Omega}$ , provides a good surrogate to use instead of  $Re_{\theta}$ , which is an integral quantity. The vorticity Reynolds number, as defined in [10], is given as follows.

$$Re_{\Omega} = \frac{d^2\Omega}{2.188\nu} \quad (16)$$

The transition momentum thickness Reynolds number,  $Re_{\theta,t}$ , is given as follows using the Praisner-Clark estimate [28] of transition momentum thickness described in section IV A 1.

$$Re_{\theta,t} = \frac{U_{\infty}\theta_t}{\nu} \approx \frac{U_{\infty}}{\nu} \sqrt{\frac{7\nu}{9\omega_{\infty}}} = \overline{Re_{\theta,t}} \quad (17)$$

Note that this correlation does not depend on  $k_{\infty}$  which is physically inconsistent and might render it ineffective in certain scenarios. However, in their study, Praisner and Clark found this correlation seems to work quite well on a range of turbomachinery configurations across different Reynolds numbers, Mach numbers, pressure gradients and freestream turbulence intensities. Hence, this correlation is currently being used as a physics-based non-dimensionalization for the first feature which is formulated as the ratio  $Re_{\Omega}/\overline{Re_{\theta,t}}$ . A conservative upper-bound for this ratio is assumed to be 3. Hence, the feature can be reformulated as follows to strictly bound the feature space as follows.

$$\eta_1 = \min \left( \frac{d^2\Omega}{2.188\nu\overline{Re_{\theta,t}}}, 3.0 \right) \quad (18)$$

Note here, that the augmentation resulting from this feature would be considerably sensitive to the far-field boundary condition of  $\omega$ , and hence the boundary conditions for  $\omega$  must be set carefully to simulate the decay of the freestream turbulence intensity as accurately as possible.

The remaining issue is to extract  $\overline{Re_{\theta,t}}$  from the freestream. In the current work, this is done as follows. A preset distance interval  $[r - \delta r, r + \delta r]$  is chosen for every case. For every discrete element on a wall boundary,  $\partial\tilde{\Omega}_k^w$ , the volume element inside the domain located at the minimum normal distance within the preset distance interval, is chosen to be the corresponding freestream cell,  $\tilde{\Omega}_k^\infty$ . If no such element is present,  $\delta r$  needs to be increased. Now for every volume element,  $\tilde{\Omega}_j$  inside the domain which is at a minimum distance from the wall boundary element  $\partial\tilde{\Omega}_k^w$  among all such wall boundary elements, any freestream quantity is set as the corresponding quantity in  $\tilde{\Omega}_k^\infty$ . While this can be inaccurate at large distances from the wall, it works fairly well inside the boundary layer with appropriate choice of the distance interval. In the interest of clarity, it should be noted that the quantities  $\omega_\infty$  and  $U_\infty$  are not constants throughout the entire domain. Instead, these freestream quantities are extracted from locations which are at some preset distance normal to the wall. Thus, for instance, in a turbomachinery configuration,  $U_\infty$  extracted relative to a point on the rotor will be very different compared to the  $U_\infty$  extracted relative to a point on the stator. Features 2 and 3 distinguish between the volume elements within the boundary layer where the augmentation needs to be significantly lower than the baseline value and volume elements elsewhere. Hence, the freestream estimate of quantities being poor far from walls is of no concern. The distance interval is usually chosen to be as close to the wall and as small as possible, such that the maximum boundary layer thickness should remain within the lower bound of this interval. It was also noted that the augmentation is not very sensitive to small changes in the choice of this interval as demonstrated in appendix C.

Note that although  $\omega_\infty$  and  $U_\infty$  seem to be non-local quantities at a first glance, but since the augmentation mainly affects the flow only within the boundary layer, the freestream quantities virtually remain constant across LIFE iterations. Thus,  $\omega_\infty$  and  $U_\infty$  can be thought of as frozen quantities local to each volume element in the discretized domain. Finally, while  $U_\infty$  can be used in moderate pressure gradient problems, the quantity could be ambiguous for very high pressure gradients. Also, note that the maximum value of  $Re_\Omega$  in the wall-normal direction approximates  $Re_\theta$  for zero pressure gradients and hence,  $\max_{d_w} Re_\Omega$  would also poorly estimate  $Re_\theta$  for very high pressure gradients.

## **Feature 2: Ratio of wall distance and turbulent length scale**

To identify laminar regions within the boundary layer, one can compare the local turbulent length scale  $\sqrt{k}/\omega$  with the wall distance  $d$ . A predominantly laminar region is characterized

by an overwhelmingly larger wall distance compared to the turbulent length scale. A modified functional form for the ratio of these two quantities which is mathematically bounded between the numerical values of 0 and 1 can be written as,

$$\eta_2 = \frac{d}{d + \sqrt{k}/\omega} \quad (19)$$

The laminar regions in the flow are denoted by values of  $\eta_1$  that are significantly close to 1. Note here that, the value of this feature will be close to 1 for regions outside the boundary layer as well.

**Feature 3: Ratio of laminar viscosity and eddy viscosity**

Since the magnitude of  $\nu_t$  in the freestream is usually greater than  $\nu$ , the following modified functional form for the ratio of these two quantities would be less than 0.5 in the freestream while being close to 1 in the laminar boundary layer and the viscous sublayer in the turbulent boundary layer.

$$\eta_3 = \frac{\nu}{\nu_t + \nu} \quad (20)$$

Together with feature 2, this feature can distinguish regions within the boundary layer for locations upstream of the transition location. Note here, that the feature design in this case is inspired from the augmentation being correlated to (and not caused by) the features.

**Limiters for flows without separation**

As we are not considering flows with separation, where the intermittency might be allowed to attain higher values in order to match the physical behavior, the resulting intermittency is simply limited to values between the theoretical bounds of 0 and 1.

**D. Augmentation function as Piecewise Linear Interpolation on Uniformly Structured Grid in Feature-space**

One of the simplest class of functions that can perform localized learning while approximating sensitivities w.r.t. features is linear interpolation on a uniform grid. While this is not the most sophisticated class of functions to approximate an augmentation, it certainly demonstrates the capability of localized learning. Note that the discretization need not be uniform, or structured for that matter, but these assumptions have been made to simplify calculations, implementation and embedding of the augmentation function within the numerical solver.

In this method, the effectively bounded feature space is discretized into a uniformly spaced structured grid. An augmentation value can be attributed to the center of each grid cell. These values can then be linearly interpolated using second order finite difference approximations inside every cell, to obtain the augmentation for any feature space location. Hence, these cell-centered values serve as the augmentation function parameters,  $\mathbf{w}$ .

Given a vector of feature values,  $\boldsymbol{\eta}$ , and a value of the augmentation function at the center of each grid cell  $\beta_{c_\ell}$ , a linearized approximation of the augmentation can be obtained as described below. Note that the vector of values  $\beta_{c_\ell}$ , viz.  $\boldsymbol{\beta}_c$ , here represents the function parameter vector  $\mathbf{w}$  mentioned in the previous sections.

1. Find the grid cell  $\ell$  which  $\boldsymbol{\eta}$  belongs to. Let the features at the center of this grid cell be  $\boldsymbol{\eta}_{c_\ell}$ .
2. Based on the augmentation value at the centers of the neighboring grid cells, calculate a gradient approximation  $\nabla_{\boldsymbol{\eta}}^{FD} \beta_\ell$  for the  $\ell^{\text{th}}$  grid cell using a second order accurate central finite difference scheme along each feature.
3. Return the following linearized approximation of the augmentation function

$$\beta_\ell(\boldsymbol{\eta}) = \beta_{c_\ell} + (\boldsymbol{\eta} - \boldsymbol{\eta}_{c_\ell}) \cdot \nabla_{\boldsymbol{\eta}}^{FD} \beta_\ell \quad (21)$$

The sensitivity approximation of the augmentation function in a locally linear fashion is sufficient for use in method of adjoints which is used to evaluate the sensitivities  $\frac{dJ}{d\mathbf{w}}$  to perform integrated inference and learning as sensitivities by definition only require the linearized functional form. Note here, that the chosen structure of the feature space, i.e. the uniform discretization characterized by  $\Delta\eta_1$  and  $\Delta\eta_2$ , and the functions parameters  $\mathbf{w} = \{\beta_{00}, \beta_{01}, \dots, \beta_{21}, \beta_{22}\}$  are sufficient to determine the linearized approximation of the augmentation as shown above, at any point in the feature space. The sensitivities  $\frac{d\beta_\ell}{d\boldsymbol{\eta}}$  and  $\frac{d\beta_\ell}{d\mathbf{w}}$ , which are used in the algorithm given for integrated inference and learning in Algorithm

1 (also refer Appendix A for the adjoint method), are given as follows:

$$\frac{\partial \beta_\ell}{\partial \boldsymbol{\eta}} = \nabla_{\boldsymbol{\eta}}^{FD} \beta_\ell \quad \frac{\partial \beta_\ell}{\partial w_m} = \frac{\partial \beta_\ell}{\partial \beta_{c_m}} = \begin{cases} 1 & \text{if } \ell = m \\ \frac{\eta_j - \eta_{c_{\ell,j}}}{2\Delta\eta_j} & \text{if } \eta_{c_m,j} = \eta_{c_{\ell,j}} + \Delta\eta_j \\ \frac{\eta_{c_{\ell,j}} - \eta_j}{2\Delta\eta_j} & \text{if } \eta_{c_m,j} = \eta_{c_{\ell,j}} - \Delta\eta_j \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

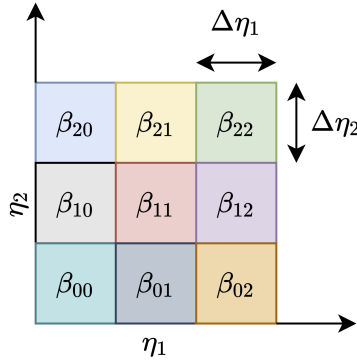


FIG. 1: Schematic of feature space discretization

For instance, given a feature pair  $(\eta_1, \eta_2)$  for the above schematic, such that the feature pair lies in the cell with center  $(1.5\Delta\eta_1, 1.5\Delta\eta_2)$ , the augmentation value can be calculated as

$$\beta(\eta_1, \eta_2) = \beta_{11} + \frac{\beta_{12} - \beta_{10}}{2\Delta\eta_1} (\eta_1 - 1.5\Delta\eta_1) + \frac{\beta_{21} - \beta_{01}}{2\Delta\eta_2} (\eta_2 - 1.5\Delta\eta_2) \quad (23)$$

For this specific function class, a trade-off is required in deciding the feature space grid resolution while learning the augmentation. If the resolution is too high, the augmentation might not be learned for a majority of grid cells as they might not be used during the optimization process, which can lead to a loss of generality of the augmentation. If the resolution is too low, the augmentation would not have enough resolution in the feature space and might lead to sub-optimal learning of the augmentation. This is further complicated by the curse of dimensionality, which restricts the resolution per feature with increasing number of features. Thus, this particular technique of approximating local augmentations using linear interpolation cannot be practically used for a “large” number of features.

## V. RESULTS

We use the datasets for the T3 series of experiments conducted by ERCOFTAC [29] to study bypass transition over flat plates across a range of inflow freestream turbulence intensities and pressure gradients. The T3A and T3B cases from this dataset have zero pressure gradient along the length of the flat plate. The freestream turbulence intensity at the inflow for T3B is significantly higher when compared to T3A. Other cases in the dataset include T3C1, T3C2, T3C3, T3C4 and T3C5. Out of these, T3C4 exhibits flow separation and since we are not considering separation-induced transition in this work, this case has been ignored. All of the T3C cases have a favorable pressure gradient near the leading edge which gradually decreases in magnitude along the length of the plate and turns into an increasingly adverse pressure gradient. T3C1 and T3C5 exhibit bypass transition in the favorable pressure gradient region of the flow, while T3C2 and T3C3 exhibit transition in regions with mild and strong adverse pressure gradients respectively.

**Training data:** The augmented model presented below is obtained by inferring the augmentation from the T3A and T3C1 cases only. Hence, the following results should be observed in the context of generality of the augmentation obtained in the low data limit. To examine sensitivity of the results to the training data, a model trained on just T3A is also shown in appendix B.

**Test cases:** To test the model on unseen data, the T3B, T3C1, T3C2 and T3C3 cases are used from the ERCOFTAC dataset. Further, to test the model on unseen geometries, the VKI dataset [2] for single-stage high-pressure-turbine cascade is used. Specifically, MUR116, MUR129, MUR224 and MUR241 are used from the VKI dataset.

### A. Training the model on T3A and T3C1 simultaneously

#### 1. Mesh for T3A and T3B

The length of the flat plate is 1.5 m with the inlet being 0.04 m upstream of the leading edge of the flat plate. The mesh resolution next to the wall is on the order of  $y^+ \approx 1$ .

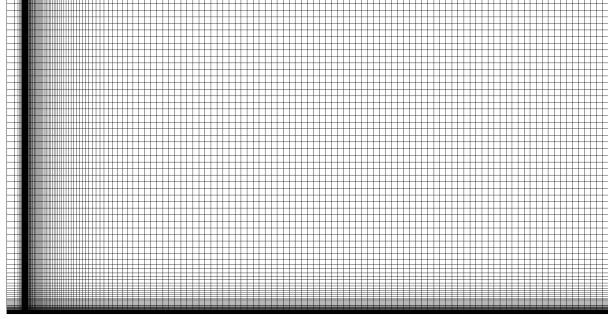


FIG. 2: Mesh used for RANS simulation of T3A and T3B cases

### 2. Mesh for T3C cases

The length of the flat plate is 1.7 m with the inlet being 0.15 m upstream of the leading edge of the flat plate. The contouring of the upper wall follows the correlations given in the work by Suluksna et al [37], which results in the trends of  $U_\infty/U_{in}$  along the length of the plate as shown in figure 4. Here  $U_\infty$  is calculated using the surface pressure,  $p_w$ , at the flat plate wall assuming incompressible flow as

$$U_\infty(x) = \sqrt{2(p_{in} + (1/2)\rho U_{in}^2 - p_w(x)) / \rho}$$

The mesh resolution next to the wall is on the order of  $y^+ \approx 1$ .

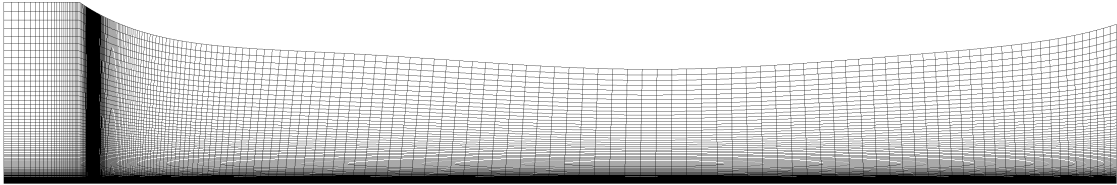


FIG. 3: Mesh used for RANS simulation of T3C cases

### 3. Inflow conditions

The information needed to characterize the inflow for both the training cases has been presented in table I. The turbulence intensity decay plots shown in figure 5 verify the  $\omega$  boundary conditions.



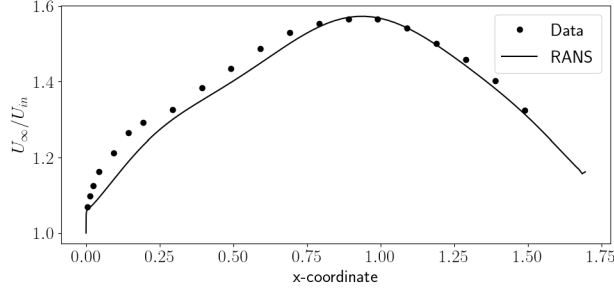
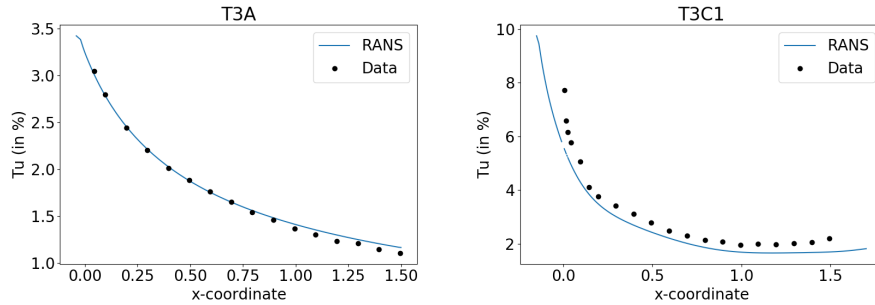


FIG. 4:  $U_\infty/U_{in}$  vs  $x$  for T3C1

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

TABLE I: Inflow conditions for the T3 cases used for training



(a)T3A

(b)T3C1

FIG. 5: Decay of freestream turbulence intensity

#### 4. LIFE on T3A and T3C1

To obtain the augmentation, LIFE was used to infer the functional parameters (see IV D) in the feature space using the T3A and T3C1 cases simultaneously. The cost function for this inference is defined as follows.

$$\mathcal{C} = \sum_{i_{\text{wall}}=0}^{n_{\text{wall}}} \|C_{f,i_{\text{wall}}} - C_{f,\text{data},i_{\text{wall}}}\|_2^2 \quad (24)$$

where  $n_{\text{wall}}$  is the number of faces corresponding to the flat plate in the mesh and  $C_f$  refers to the local skin friction coefficient. The two different problems and the discretization of the feature space acts as an implicit regularizer. Figures 6 and 7 show the optimization and residual histories. Figure 8 shows the  $C_f$  of the baseline ( $\beta = 1$  throughout the domain) and optimal skin friction coefficient distributions compared to data. The initial residual convergence for the direct solver is much better compared to the subsequent optimization iterates. This is due to the change in the feature-to-augmentation relationship, which affects the convergence. However, the magnitude of the residual suggests sufficient convergence. Also, one does not see a significant drop in the residuals because each optimization iterate is restarted from the converged state of the previous iterate. The effectively bounded feature space was uniformly discretized into a cartesian grid with 30, 10 and 10 cells along the features  $\eta_1$ ,  $\eta_2$  and  $\eta_3$ , respectively. It should be noted that the discretization of the feature space presented here is not necessarily the most efficient one. The objective rather is to demonstrate the capability of localized learning. Further, the discretization need not be in the form of a uniform grid as shown here and could be made adaptive. Finally, as mentioned in section IV D, the discretization needs to be just enough to characterize the augmentation satisfactorily. We found that the discretization presented here predicts the transition location well.

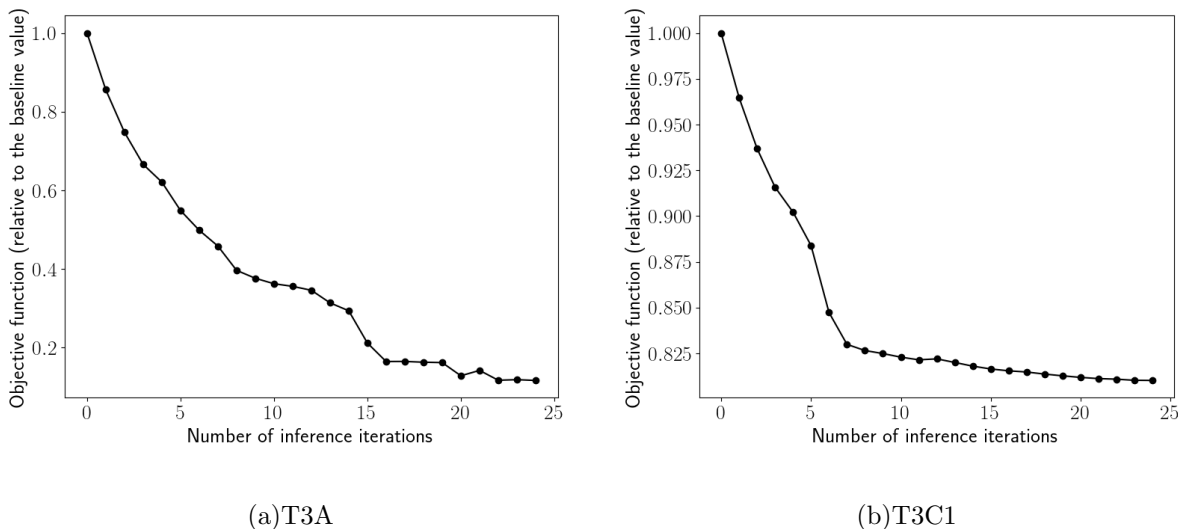
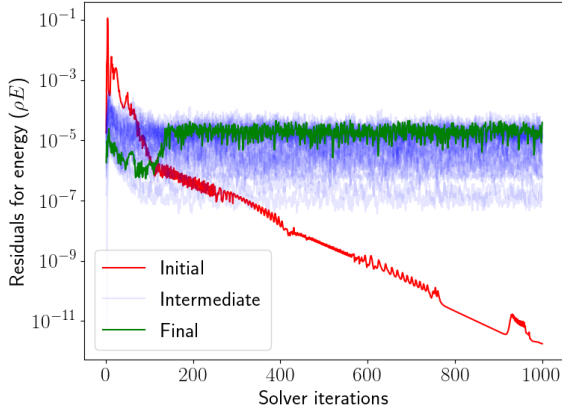
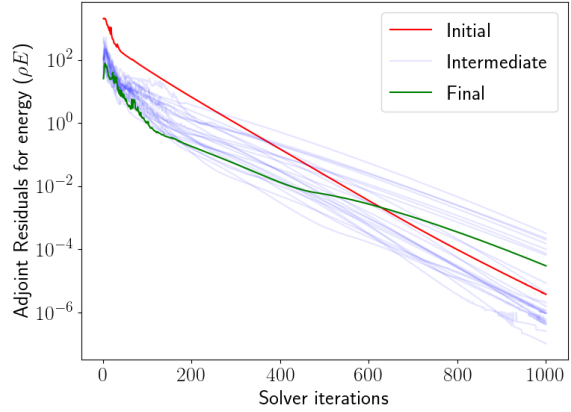


FIG. 6: Field Inversion: Optimization plots

As can be seen in the plots above, the transition locations are inferred with a reasonable accuracy for both the cases. Notice, that in the T3C1 case, the skin friction coefficient

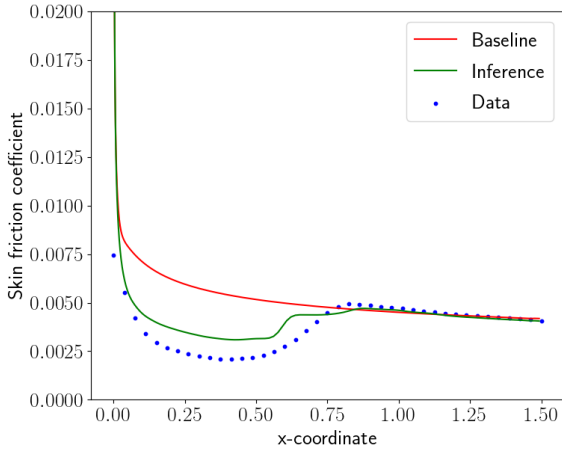


(a) Residual convergence for energy

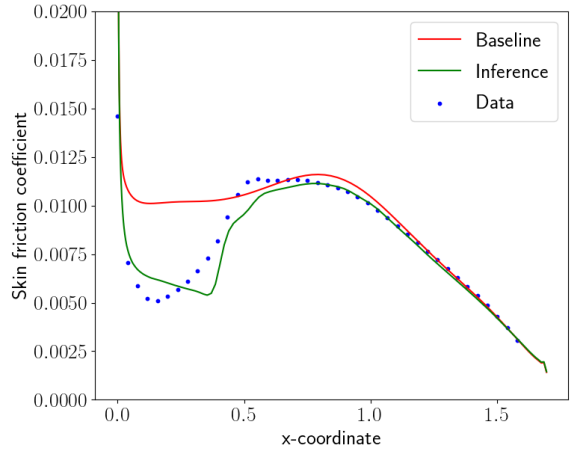


(b) Residual convergence for energy adjoints

FIG. 7:  $\rho E$  Residual histories for all optimization iterates



(a) T3A



(b) T3C1

FIG. 8: Skin friction coefficients along the length of the flat plate

deviates from the data in the vicinity of transition. The same happens for T3A, though the effects there are much subtler. Also, it can be noticed that the laminar part of the flow does not show fully laminar friction coefficients as can be seen when compared to data. This can be attributed to the predicted intermittency having low values in a very narrow region in the flow field. Better results may be obtained by using good limiters on features and/or the predicted augmentation values. For instance, the augmentation value can be set to zero if it is below a certain threshold (e.g., 0.3). Given that such a threshold would have to be extracted using training cases and might not work for certain test cases, these limiters might

be ad hoc.

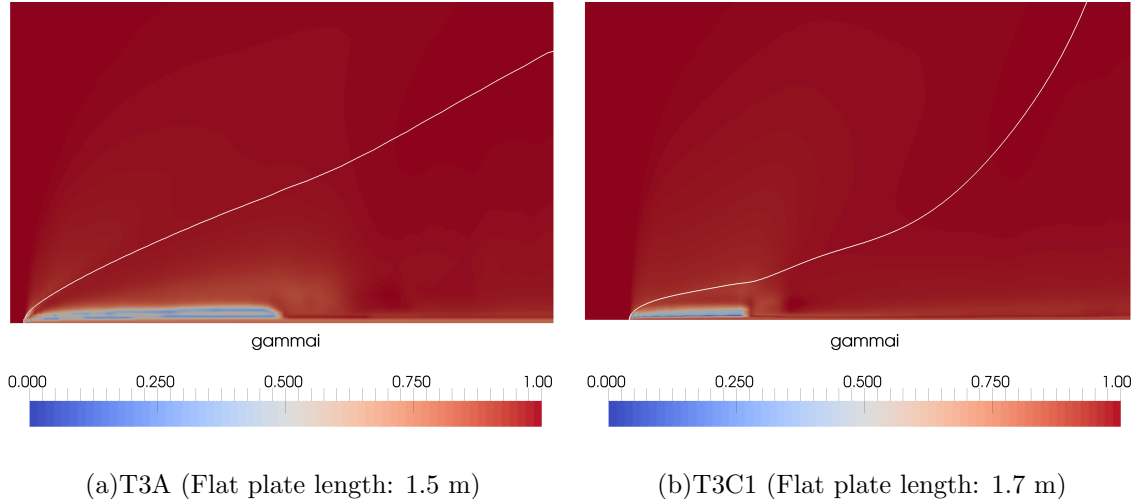


FIG. 9: Intermittency contours (White line marks  $U = 0.95U_\infty$ ), 40x scaling in the y-direction

The contours of augmentation w.r.t. the feature space (slices of  $\eta_1$ ) are shown in Figure 10.

As can be seen in the feature space contours, the augmentation varies w.r.t.  $\eta_1$  until only around  $\eta_1 \approx 0.75$  above which it has a virtually constant value of 1.0 (Baseline). This happens because the augmentation is trained using cases involving transition occurring in regions of zero/favorable pressure gradients. Training the cases for the adverse pressure gradient regions would potentially populate some region with higher values of  $\eta_1$ . The augmentation assumes small values only in regions of high  $\eta_3$  because low intermittency correlates to laminar regions in the flow where  $\eta_3$  will be high. However, this is also the case in the viscous sub-layer and some parts of the buffer layer in the fully turbulent parts of the flow. In such a scenario,  $\eta_2$  distinguishes between the laminar and turbulent parts of the flow. This is a clear example of how adding an extra feature can help in differentiating physical regions where the augmentation must take different values.

For the purpose of comparing different learning techniques, three different set of results obtained - (1) using only the T3A case for training, (2) without using localized learning (i.e. using Neural Networks), and (3) using a finer discretization have been shown in appendices B, D and E, respectively. Looking at these results, we can make the following observations:

1. As seen from the augmentation obtained by only using data from the T3A dataset for training in appendix B which performs worse compared to the augmentation obtained

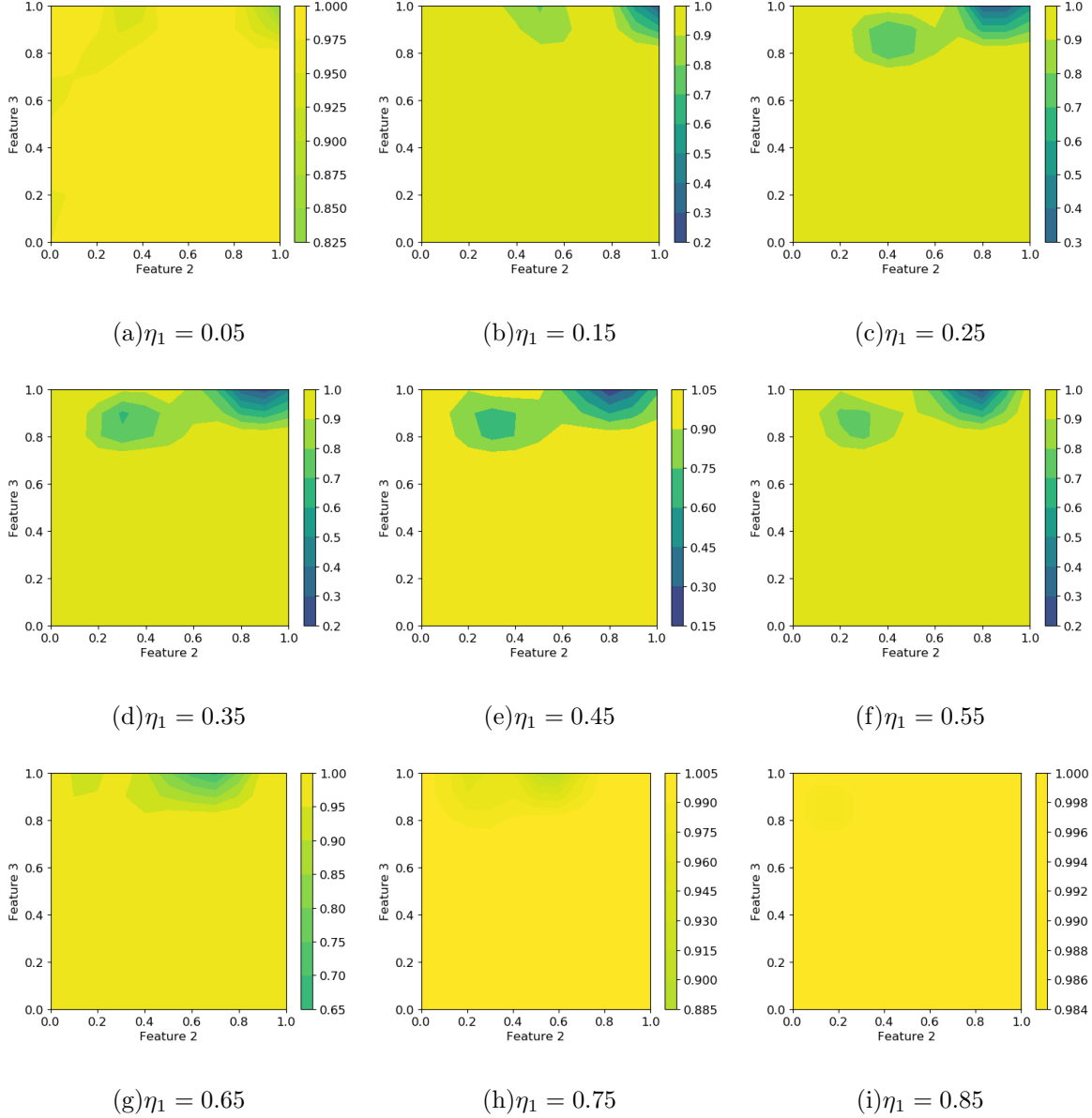


FIG. 10: Augmentation contours on feature space slices after training on T3A and T3C1 datasets

using both the cases T3A and T3C1, adding more datasets which exhibit different physical phenomena compared to the already existing training datasets consistently improves predictive accuracy.

2. Reducing spurious behavior using localized learning can not only make the augmentation more robust but is at times essential to infer a usable augmentation from available data. This is clearly evident when comparing the results presented in this section with the training results presented in appendix D (which do not use localized learning and use neural networks as the functional form for the augmentation and fail miserably by

predicting partially/fully laminar flow at all locations along the flat plate).

3. During localized learning, correctly setting the range of influence that the datapoints have in the feature space is crucial in order to obtain a generalizable augmentation as mentioned in section III C. Also, as mentioned in section III C, this work does not deal with a method to optimally set the range of influence and instead focuses on why localized learning is necessary to improve generalizability and reduce spurious behavior arising from extrapolation within the bounded feature space. Nevertheless, the comparison between the test results for augmentations using different grid resolutions in the feature space (see appendix E where a finer grid leads to the augmentation being learnt in a smaller region of the feature space which results in limited generalizability) demonstrate the importance of setting an optimal range of influence.

## B. Testing the model

### 1. Mesh for VKI turbine cascade cases

The mesh used for the turbine cascade cases can be seen in Figure 11. The blade chord is 0.067 m in length and makes a  $55^\circ$  angle with the streamwise direction. The inlet is situated 0.055 m upstream of the leading edge, while the outlet is located 0.242 m downstream of the leading edge. The mesh resolution next to the wall is on the order of  $y^+ \approx 1$ .

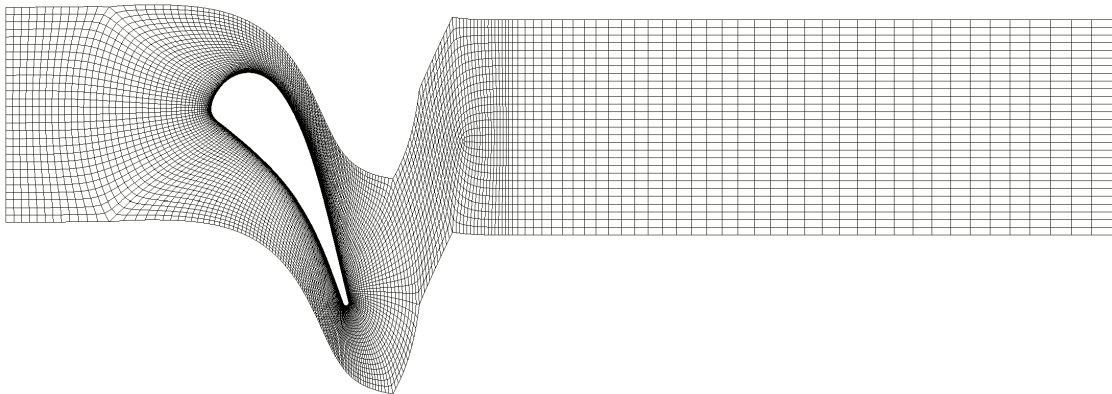


FIG. 11: Mesh used for RANS simulation of VKI turbine cascade

## 2. Inflow conditions for T3 test cases

The information needed to characterize the inflow conditions for the T3 test cases has been mentioned in table II. The turbulence intensity decay plots shown in figure 12 verify the  $\omega$  boundary conditions.

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

TABLE II: Inflow conditions for the T3 test cases

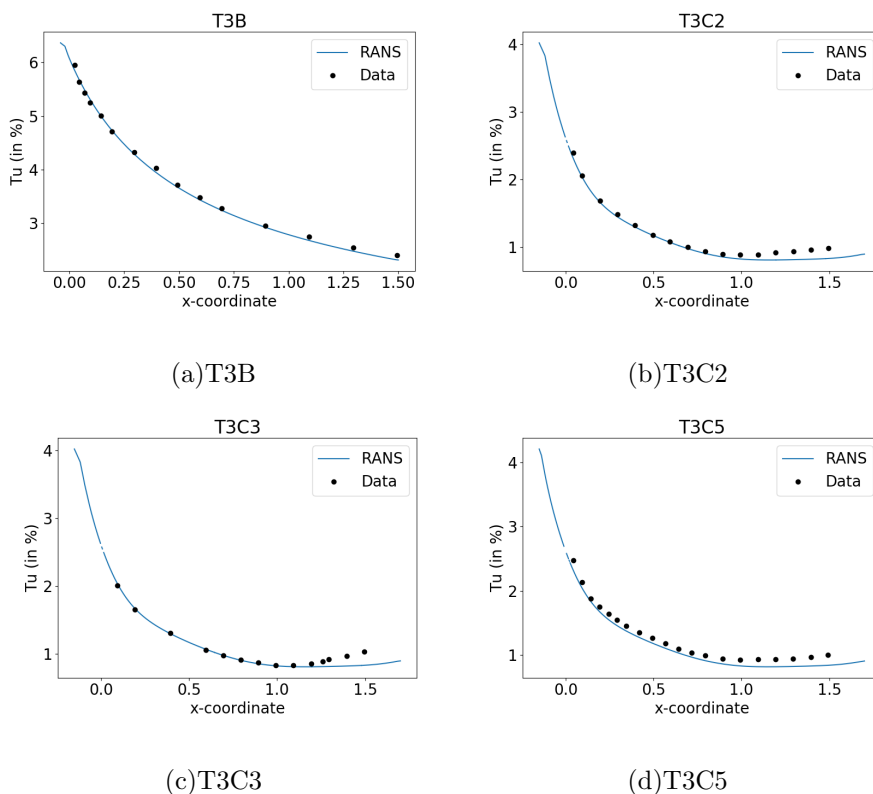


FIG. 12: Decay of freestream turbulence intensity

### 3. Inflow, wall and outflow conditions for VKI test cases

The information needed to characterize the inflow conditions for the VKI turbine cascade test cases along with pressure at outflow boundary and temperature at the isothermal wall is presented in table III. The boundary condition for  $\omega_{\text{in}}$  was set to ensure that the viscosity ratio  $\nu_{t_{\text{in}}}/\nu_{\text{in}}$  remains between the range from 1.5 to 50.

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

TABLE III: Inflow, wall and outflow conditions for VKI test cases

Notice here, that MUR116 is a case with high pressure differential and low  $Tu_{\infty}$ , MUR129 is a case with low pressure differential and low  $Tu_{\infty}$ , MUR241 is a case with high pressure differential and high  $Tu_{\infty}$ , and finally MUR224 is a case with low pressure differential and high  $Tu_{\infty}$ . This distinction will be important later.

The comparison of velocity profiles at three different locations on the turbine blade along the wall normal direction is shown in figure 13 for verification.

### 4. Predictions using the augmented model on the same geometries

Figure 14 shows results obtained by applying the model trained on T3A and T3C1 to the T3B, T3C2, T3C3, and T3C5 cases. The transition locations for T3B, T3C2 and T3C5 cases are predicted reasonably well in the results shown below. This is because the transition occurs in all of these cases in either zero, favorable or very mild adverse pressure gradient regions. The model fails to predict the transition location correctly for T3C3 for this reason and instead assumes the baseline value for  $\beta = 1$  in the adverse pressure gradient regions, thus predicting premature transition. Once again, it can be noticed that the laminar part



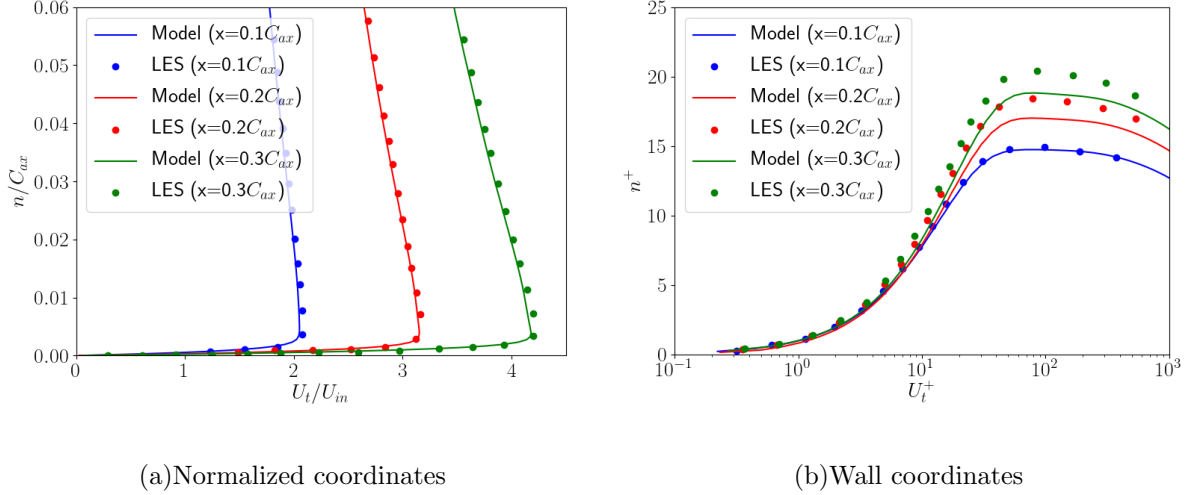


FIG. 13: Verification of velocity profiles for MUR224 using LES data [54] ( $C_{ax}$  refers to chord length in the x-direction)

of the flow does not show fully laminar friction coefficients as can be seen when compared to data. This can be attributed, as explained before, to the predicted intermittency having low values in a very narrow region of the flow field.

##### 5. Predictions using the augmented model on a different geometry

Figure 18 shows the heat transfer coefficients predicted by the model trained on T3A and T3C1 on the VKI high pressure turbine cases, which are not only geometrically different from the training set, but also involve combinations of pressure gradients and freestream turbulence intensities not seen in the training data. Predictions with two different preset distance intervals to extract  $Re_{\theta,t}$  information from the freestream are shown in Appendix C to assess the sensitivity of the solution to such modeling choices. For MUR224 and MUR129, the transition location is predicted fairly accurately and consistently with both the preset distance intervals. For MUR241, the transition location on the suction side is predicted very well. On the pressure side, however, the data shows a gradual transition to turbulence, while the model predicts a sharp transition at different locations within this gradual transition range when using different preset distance intervals. For MUR116, transition is premature for on either sides of the blade.

To diagnose the behavior in the MUR116 case, Figure 16 shows contours of two features and

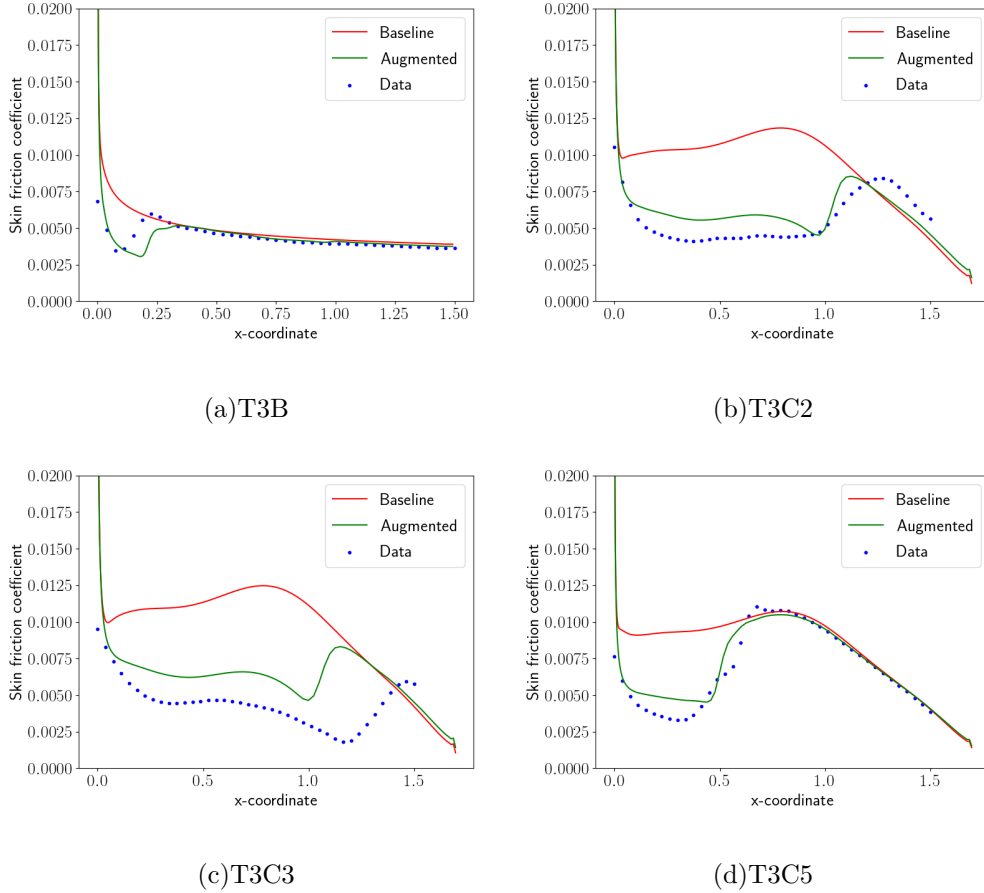
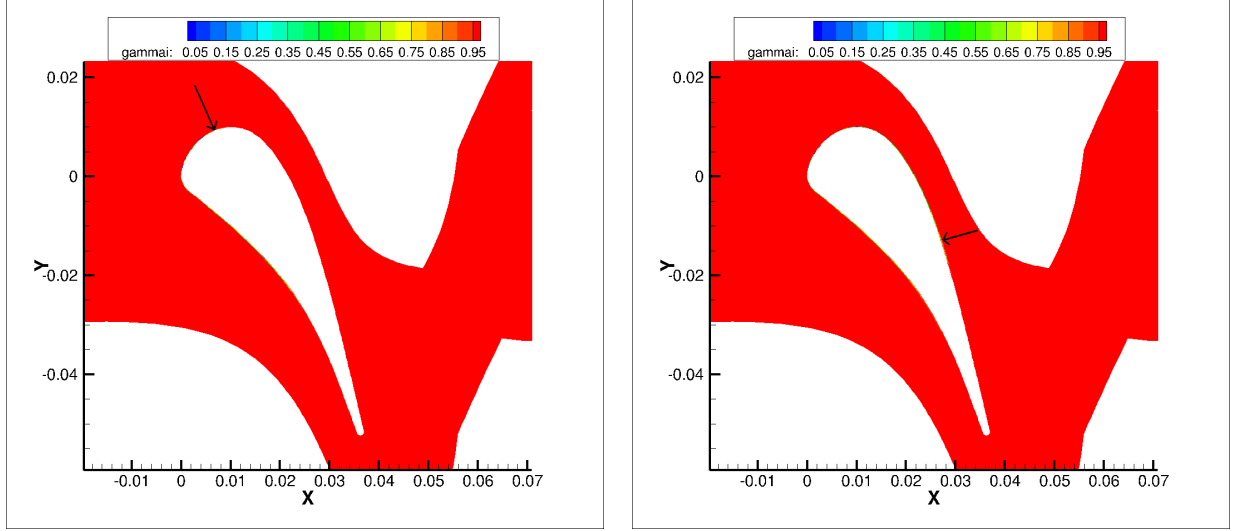


FIG. 14: Predicted Skin friction coefficients for flat plate cases using the model trained on T3A and T3C1.

the intermittency near the transition location. The intermittency is seen to grow rapidly in the region  $0.1 \leq \eta_1 \leq 0.2$  and  $\eta_2 \leq 0.8$ . This corresponds to a region in the feature space in which the augmentation does not learn significantly from the available data, leading to the augmentation reverting to its baseline behavior of  $\beta = 1$ . It should be recognized however, that there is a strong feedback loop between the features, augmentation and transport processes. This means that all these quantities can influence each other in the model irrespective of the physical cause-effect relationships. This behavior can be improved by bringing in more data to populate a larger part of the feature space. For comparison purposes,  $\eta_1$ ,  $\eta_2$  and  $\gamma$  contours for MUR241 are shown in figure 17 which exhibit a physically intuitive growth of the intermittency in a region of high  $\eta_1$  and  $\eta_2$ .

In addition, it can also be observed that - while the coefficient values are close to the data in laminar regions - there are large discrepancies between the two in the fully turbulent regions.



(a)MUR116

(b)MUR241

FIG. 15: Zoomed out views to indicate predicted transition locations on upper surface

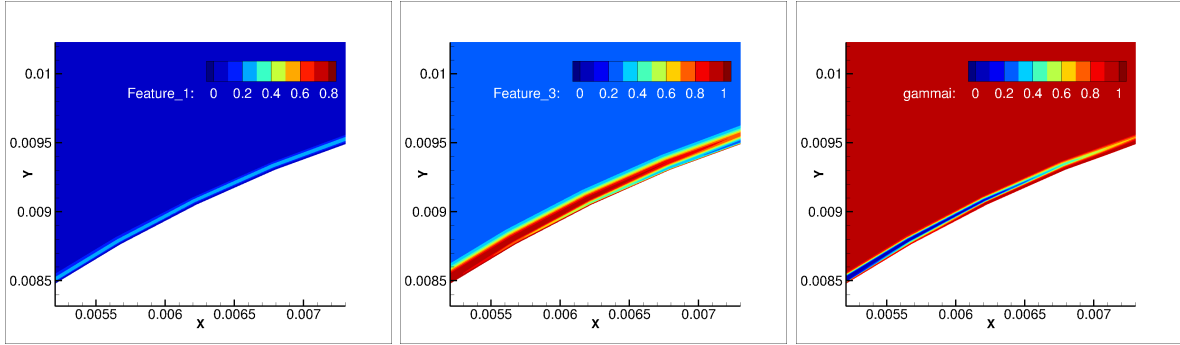
(a)  $\eta_1 = \min\left(\frac{Re_\Omega}{Re_{\theta,t}}, 3\right)$ (b)  $\eta_3 = \frac{\nu}{\nu + \nu_t}$ (c) Intermittency ( $\gamma$ )

FIG. 16: Contours near the transition location on suction surface for MUR116

This is because the turbulence model is imperfect. These discrepancies are not related to the transition phenomena and hence, are beyond the scope of the current augmentation.

## VI. SUMMARY

This work presents a methodology that can be used to build data-driven augmentations to physics-based models. Building on recent work in data-driven turbulence modeling, this approach develops generalizable features-to-augmentation mapping across multiple datasets in a model-consistent manner. The proposed framework, referred to as “Learning and Inference

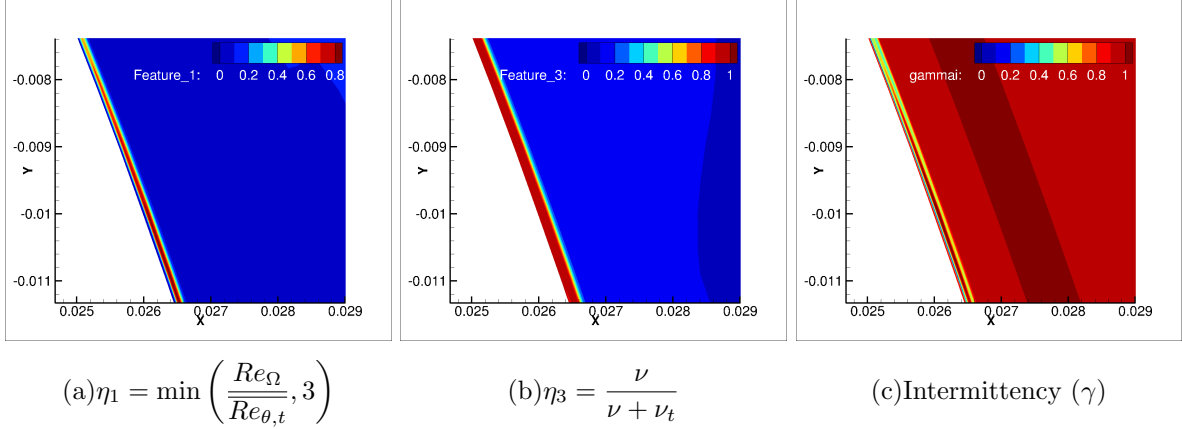
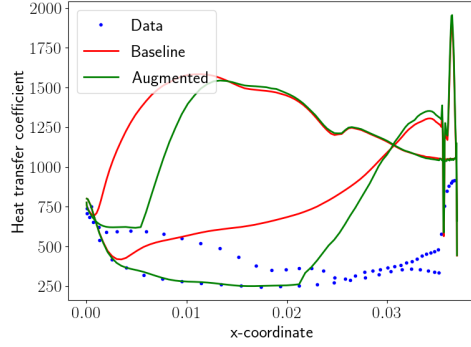
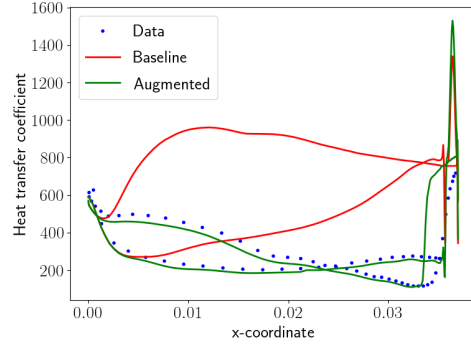


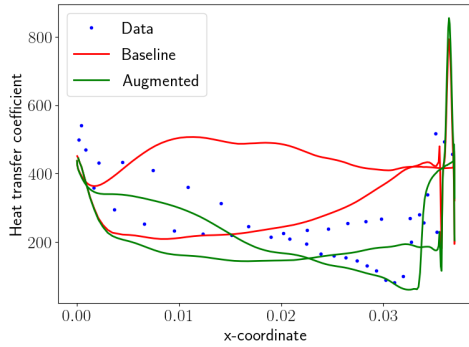
FIG. 17: Contours near the transition location on suction surface for MUR241



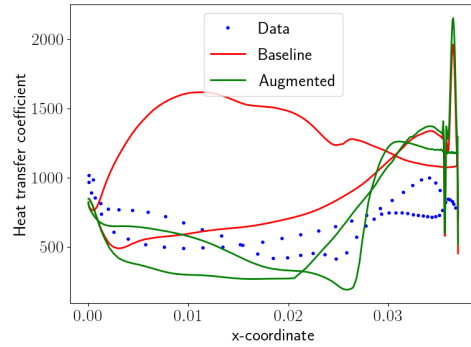
(a) MUR116



(b) MUR129



(c) MUR224



(d) MUR241

FIG. 18: Predicted heat transfer coefficients for VKI turbine cases using model trained on T3A and T3C1.

assisted by Feature-space Engineering (LIFE)” places particular emphasis on the design of the feature space and control of the augmentation behavior in the feature space. Choosing features (along with appropriate physics-based non-dimensionalization) to create an “effec-

tively” bounded feature space is indispensable as this boundedness ensures the tractability of the learning problem. In an ideal scenario, if an augmentation is learnt in this entire bounded region, an extrapolation in geometry/boundary conditions would translate to an interpolation in feature space. Given the availability of a limited number of datasets, human intuition and expert knowledge is essential, in the authors’ opinion to formulate such features. Although automated feature selection techniques do exist, they are heavily (and often prohibitively) expensive to find a parsimonious combination of features that can address the augmentation, especially for complex physical problems like transition and turbulence. To address the issue of the possible lack of data in certain significantly sized regions of the feature space, the notion of localized learning becomes critical to avoid spurious learning outputs. Localized learning refers to the modification of the augmentation function behavior at every learning step only in the vicinity of available datapoints in the feature space. Additional measures, such as preventing a region in the feature space from representing different regions in the physical space where the augmentation must assume different values (i.e. ensuring a one-to-one features-to-augmentation mapping) by selecting an appropriate combination of features, also helps improve robustness. **The LIFE framework offers a framework along with guiding principles and techniques that are intended for use by modelers to develop data-driven augmentations for low-fidelity models using high-fidelity data.**

To demonstrate LIFE in practice, a simple intermittency-based bypass transition model-form based on Wilcox’s  $k-\omega$  turbulence model was augmented appropriately with a careful choice of features. This augmentation function is inferred from two benchmark cases of flat plate transition from the T3 series of experiments. These two cases are characterized by transition under zero pressure gradient and favorable pressure gradient regions respectively, with an inflow freestream turbulence intensity significantly greater than 1%. Linear interpolation on uniformly structured feature-space discretization was used as the functional form for the augmentation in order to implement a very crude variant of localized learning. Though more sophisticated ways to perform localized learning are possible, this implementation is showcased to demonstrate how effective and capable the notion of localized learning can be. This augmented model is then used to predict transition on four other flat plate cases and four single-stage high-pressure-turbine cascade cases. The augmented model is shown to be able to predict the transition locations for problems in which similar physics was

encountered in the dataset. It is noted that the focus of this work is not to develop the ultimate bypass transition model, rather to present a formalism that can be used to develop physics-constrained, data-enabled models. More comprehensive training datasets, especially involving higher pressure gradients and turbulence intensities can help in improving the accuracy and applicability of the model.

Currently, there are three major challenges in the LIFE framework - (1) Feature design for a given application; (2) Deciding the range of the vicinity in which a datapoint updates the augmentation; and (3) Formal uncertainty quantification. Designing features that can capture the behavior of an augmentation function in the simplest possible manner is a challenge that modelers have been dealing with for decades now. While there exist numerical diagnostics and techniques to assist and/or automate feature selection and engineering, the authors feel that domain expertise and intuition is the best way forward for problems as complex as turbulence and transition, at least given the prevailing constraints on data and computational power which render automatic feature selection infeasible. As far as the range of influence that a datapoint has in the feature space is concerned, a trade-off needs to be made between over-fitting and predictive accuracy. If the influence is very limited, the augmented model will perform very well on the training cases, but can revert to baseline behavior for test cases which lie outside the influence of these datapoints. On the other hand, if the influence is too wide, the augmentation behavior might not be sufficiently resolved and hence could adversely affect predictive accuracy. Finally, formal uncertainty quantification techniques need to be developed for use in design of experiments to extend the capability of the LIFE framework.

- 
- [1] B. J. Abu-Ghannam and R. Shaw. Natural transition of boundary layers—the effects of turbulence, pressure gradient, and flow history. Journal of Mechanical Engineering Science, 22(5):213–228, 1980.
  - [2] T. Arts, M. Lambert de Rouvroit, and A. W. Rutherford. Aero-thermal investigation of a highly loaded transonic linear turbine guide vane cascade: A test case for inviscid and viscous flow computations. Available at <http://resolver.tudelft.nl/uuid:1bdc512b-a625-4607-a339-3cc9b371215e>.

- [3] J. R. Cho and M. K. Chung. A  $k-\varepsilon-\gamma$  equation turbulence model. Journal of Fluid Mechanics, 237:301–322, 1992.
- [4] S. Dhawan and R. Narasimha. Some properties of boundary layer flow during the transition from laminar to turbulent motion. Journal of Fluid Mechanics, 3(4):418–436, 1958.
- [5] E. Dow and Q. Wang. Quantification of structural uncertainties in the  $k-w$  turbulence model. In 52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference.
- [6] K. Duraisamy. Perspectives on machine learning-augmented reynolds-averaged and large eddy simulation models of turbulence. Physical Review Fluids, 2021.
- [7] K. Duraisamy and J. Alonso. Adjoint based techniques for uncertainty quantification in turbulent flows with combustion. In 42nd AIAA Fluid Dynamics Conference and Exhibit, page 2711, 2012.
- [8] K. Duraisamy and P. Durbin. Transition modeling using data driven approaches. In Proc. CTR Summer Program, Stanford University, pages 427–434.
- [9] K. Duraisamy, G. Iaccarino, and H. Xiao. Turbulence modeling in the age of data. Annual Review of Fluid Mechanics, 51:357–377, 2019.
- [10] P. Durbin. An intermittency model for bypass transition. International Journal of Heat and Fluid Flow, 36:1–6, 2012.
- [11] A. Ferrero, A. Iollo, and F. Larocca. Field inversion for data-augmented rans modelling in turbomachinery flows. Computers & Fluids, 201:104474, 02 2020.
- [12] X. Ge, S. Arolla, and P. Durbin. A bypass transition model based on the intermittency function. Flow, turbulence and combustion, 93(1):37–61, 2014.
- [13] J. R. Holland, J. D. Baeder, and K. Duraisamy. Field inversion and machine learning with embedded neural networks: Physics-consistent neural network training. AIAA Aviation 2019 Forum, 2019.
- [14] J. R. Holland, J. D. Baeder, and K. Duraisamy. Towards integrated field inversion and machine learning with embedded neural networks for rans modeling. AIAA Scitech 2019 Forum, 2019.
- [15] F. Köhler, J. Munz, and M. Schäfer. Data-driven augmentation of rans turbulence models for improved prediction of separation in wall-bounded flows. In AIAA Scitech 2020 Forum.
- [16] R. B. Langtry and F. R. Menter. Correlation-based transition modeling for unstructured parallelized computational fluid dynamics codes. AIAA journal, 47(12):2894–2906, 2009.

- [17] S. Lardeau, A. Fadaei-Ghotbi, and M. Leschziner. Modelling bypass and separation-induced transition by reference to pre-transitional fluctuation energy. ERCOFTAC bulletin, 80:72–76, 2009.
- [18] J. Ling, R. Jones, and J. Templeton. Machine learning strategies for systems with invariance properties. Journal of Computational Physics, 318:22 – 35, 2016.
- [19] W. Liu, J. Fang, S. Rolfo, C. Moulinec, and D. R. Emerson. An iterative machine-learning framework for turbulence modeling in rans. arXiv preprint arXiv:1910.01232, 2019.
- [20] R. Matai and P. A. Durbin. Zonal eddy viscosity models based on machine learning. Flow, Turbulence and Combustion, pages 93–109.
- [21] R. E. Mayle. The 1991 IGTI Scholar Lecture: The Role of Laminar-Turbulent Transition in Gas Turbine Engines. Journal of Turbomachinery, 113(4):509–536, 10 1991.
- [22] R. E. Mayle and A. Schulz. Heat Transfer Committee and Turbomachinery Committee Best Paper of 1996 Award: The Path to Predicting Bypass Transition. Journal of Turbomachinery, 119(3):405–411, 07 1997.
- [23] F. R. Menter, R. B. Langtry, S. R. Likki, Y. B. Suzen, P. G. Huang, and S. Völker. A Correlation-Based Transition Model Using Local Variables—Part I: Model Formulation. Journal of Turbomachinery, 128(3):413–422, 03 2004.
- [24] P. M. Milani, J. Ling, and J. K. Eaton. Generalization of machine-learned turbulent heat flux models applied to film cooling flows. Journal of Turbomachinery, 142(1), 2020.
- [25] P. M. Milani, J. Ling, and J. K. Eaton. Turbulent scalar flux in inclined jets in crossflow: counter gradient transport and deep learning modelling. Journal of Fluid Mechanics, 906, 2021.
- [26] T. A. Oliver and R. D. Moser. Bayesian uncertainty quantification applied to RANS turbulence models. Journal of Physics: Conference Series, 318(4):042032, dec 2011.
- [27] E. J. Parish and K. Duraisamy. A paradigm for data-driven predictive modeling using field inversion and machine learning. Journal of Computational Physics, 305:758–774, 2016.
- [28] T. Praisner and J. Clark. Predicting transition in turbomachinery: Part i—a review and new model development. In ASME Turbo Expo 2004: Power for Land, Sea, and Air, pages 161–174. American Society of Mechanical Engineers Digital Collection, 2004.
- [29] P. E. Roach and D. H. Brierley. The influence of a turbulent free-stream on zero pressure gradient transitional boundary layer development part 1: Test cases t3a and t3b. Numerical



- Simulation of Unsteady Flows and Transition to Turbulence, 3(7):319–327, 2018.
- [30] M. Schmelzer, R. P. Dwight, and P. Cinnella. Discovery of algebraic reynolds-stress models using sparse symbolic regression. Flow, Turbulence and Combustion, 104(2):579–603, 2020.
- [31] A. P. Singh, K. Duraisamy, and S. Pan. Characterizing and improving predictive accuracy in shock-turbulent boundary layer interactions using data-driven models. In 55th AIAA Aerospace Sciences Meeting.
- [32] A. P. Singh, K. Duraisamy, and Z. J. Zhang. Augmentation of turbulence models using field inversion and machine learning. 55th AIAA Aerospace Sciences Meeting, 2017.
- [33] A. P. Singh, R. Matai, A. Mishra, K. Duraisamy, and P. A. Durbin. Data-driven augmentation of turbulence models for adverse pressure gradient flows. 23rd AIAA Computational Fluid Dynamics Conference, 2017.
- [34] A. P. Singh, S. Medida, and K. Duraisamy. Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. AIAA Journal, 55(7):2215–2227, 2017.
- [35] J. Sirignano, J. F. MacArt, and J. B. Freund. Dpm: A deep learning pde augmentation method with application to large-eddy simulation. Journal of Computational Physics, 423:109811, 2020.
- [36] J. Steelant and E. Dick. Modelling of bypass transition with conditioned navier–stokes equations coupled to an intermittency transport equation. International journal for numerical methods in fluids, 23(3):193–220, 1996.
- [37] K. Suluksna, P. Dechaumphai, and E. Juntasaro. Correlations for modeling transitional boundary layers under influences of freestream turbulence and pressure gradient. International Journal of Heat and Fluid Flow, 30(1):66 – 75, 2009.
- [38] Y. B. Suzen and P. G. Huang. Modeling of Flow Transition Using an Intermittency Transport Equation . Journal of Fluids Engineering, 122(2):273–284, 02 2000.
- [39] Y. B. Suzen, P. G. Huang, L. S. Hultgren, and D. E. Ashpis. Predictions of Separated and Transitional Boundary Layers Under Low-Pressure Turbine Airfoil Conditions Using an Intermittency Transport Equation . Journal of Turbomachinery, 125(3):455–464, 08 2003.
- [40] S. Taghizadeh, F. D. Witherden, and S. S. Girimaji. Turbulence closure modeling with data-driven techniques: physical compatibility and consistency considerations. New Journal of Physics, 22(9):093023, Sep 2020.
- [41] B. Tracey, K. Duraisamy, and J. Alonso. Application of supervised learning to quantify uncer-

- tainties in turbulence and combustion modeling. In 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition.
- [42] B. D. Tracey, K. Duraisamy, and J. J. Alonso. A machine learning strategy to assist turbulence model development. In 53rd AIAA Aerospace Sciences Meeting.
- [43] E. R. VAN DRIEST and C. B. BLUMER. Boundary layer transition- freestream turbulence and pressure gradient effects. AIAA Journal, 1(6):1303–1306, 1963.
- [44] C. A. V. Venegas and D. Huang. Expedient hypersonic aerothermal prediction for aerothermoelastic analysis via field inversion and machine learning. In AIAA Scitech 2021 Forum.
- [45] D. K. Walters and D. Cokljat. A three-equation eddy-viscosity model for reynolds-averaged navier–stokes simulations of transitional flow. Journal of fluids engineering, 130(12), 2008.
- [46] J.-X. Wang, J.-L. Wu, and H. Xiao. Physics-informed machine learning approach for reconstructing reynolds stress modeling discrepancies based on dns data. Physical Review Fluids, 2(3):034603, 2017.
- [47] J. Weatheritt and R. Sandberg. A novel evolutionary algorithm applied to algebraic modifications of the rans stress–strain relationship. Journal of Computational Physics, 325:22 – 37, 2016.
- [48] J.-L. Wu, H. Xiao, and E. Paterson. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. Physical Review Fluids, 3(7):074602, 2018.
- [49] H. Xiao and P. Cinnella. Quantification of model uncertainty in rans simulations: A review. Progress in Aerospace Sciences, 108:1–31, 2019.
- [50] H. Xiao, J.-L. Wu, J.-X. Wang, and E. G. Paterson. Physics-informed machine learning for predictive turbulence modeling: Progress and perspectives. Proceedings of the 2017 AIAA SciTech, 2017.
- [51] H. Xiao, J.-L. Wu, J.-X. Wang, R. Sun, and C. Roy. Quantifying and reducing model-form uncertainties in reynolds-averaged navier–stokes simulations: A data-driven, physics-informed bayesian approach. Journal of Computational Physics, 324:115–136, 2016.
- [52] M. Yang and Z. Xiao. Improving the  $k-\omega-\gamma-a_r$  transition model by the field inversion and machine learning framework. Physics of Fluids, 32(6):064101, 2020.
- [53] Y. Zhao, H. D. Akolekar, J. Weatheritt, V. Michelassi, and R. D. Sandberg. Rans turbulence model development using cfd-driven machine learning. Journal of Computational Physics,

- [54] Y. Zhao and R. D. Sandberg. Bypass transition in boundary layers subject to strong pressure gradient and curvature effects. *Journal of Fluid Mechanics*, 888:A4, 2020.

## Appendix A: Sensitivity evaluation using discrete adjoints

In a field inversion problem, the objective function  $\mathcal{J} = \mathcal{C}(\tilde{\mathbf{u}}_m) + \lambda_\beta \mathcal{T}_\beta(\beta)$  is a function of both  $\tilde{\mathbf{u}}_m$  and  $\beta$ , and since  $\tilde{\mathbf{u}}_m$  is dependent on  $\beta$  via  $\mathcal{R}(\tilde{\mathbf{u}}_m; \beta, \boldsymbol{\xi}) = 0$ , we can write the following.

$$\begin{aligned} \frac{d\mathcal{J}}{d\beta} &= \left[ \frac{\partial \mathcal{J}}{\partial \beta} \right]_{\tilde{\mathbf{u}}_m} + \left[ \frac{\partial \mathcal{J}}{\partial \tilde{\mathbf{u}}_m} \right]_{\beta} \left[ \frac{\partial \tilde{\mathbf{u}}_m}{\partial \beta} \right]_{\mathcal{R}=0} \\ &\quad \left[ \frac{\partial \mathcal{R}}{\partial \beta} \right]_{\tilde{\mathbf{u}}_m} + \left[ \frac{\partial \mathcal{R}}{\partial \tilde{\mathbf{u}}_m} \right]_{\beta} \left[ \frac{\partial \tilde{\mathbf{u}}_m}{\partial \beta} \right]_{\mathcal{R}=0} = 0. \end{aligned}$$

Now, using the above representations one can write,

$$\frac{d\mathcal{J}}{d\beta} = \left[ \frac{\partial \mathcal{J}}{\partial \beta} \right]_{\tilde{\mathbf{u}}_m} - \left[ \frac{\partial \mathcal{J}}{\partial \tilde{\mathbf{u}}_m} \right]_{\beta} \left[ \frac{\partial \mathcal{R}}{\partial \tilde{\mathbf{u}}_m} \right]_{\beta, \boldsymbol{\xi}}^{-1} \left[ \frac{\partial \mathcal{R}}{\partial \beta} \right]_{\tilde{\mathbf{u}}_m, \boldsymbol{\xi}}$$

Here, the sensitivities of a single objective function w.r.t. several variables need to be evaluated. In such a case, it is cheaper to solve for  $\frac{\partial \mathcal{J}}{\partial \tilde{\mathbf{u}}_m} \left[ \frac{\partial \mathcal{R}}{\partial \tilde{\mathbf{u}}_m} \right]^{-1}$  compared to solving for  $\left[ \frac{\partial \mathcal{R}}{\partial \tilde{\mathbf{u}}_m} \right]^{-1} \frac{\partial \mathcal{R}}{\partial \beta}$  and the solution to the former is referred to as the vector of adjoint variables. An algorithmic differentiation (AD) package is used to evaluate the different jacobians involved in the above discrete adjoint approach. The in-house solver [7] used in this work makes use of the ADOL-C package for AD.

The integrated inference and learning framework, however requires additional infrastructure for this to work. Since, there is an additional dependence of  $\beta(\eta(\tilde{\mathbf{u}}_m, \boldsymbol{\zeta}); \mathbf{w})$  on  $\tilde{\mathbf{u}}_m$  to account for, this pathway needs to be included in the calculation of any jacobians w.r.t. states  $\tilde{\mathbf{u}}_m$ . Inclusion of the entire augmentation function in the AD calculation is unnecessary and expensive, however, as one only requires sensitivities of the augmentation w.r.t the features. The calculation of features from states can be implemented within the AD framework relatively easily. Thus, calculating a linear approximation for the augmentation in feature space and using AD for this linearized function is a better and efficient alternative to evaluate this Jacobian. Any other jacobians w.r.t. the augmentation can be converted to the jacobians w.r.t. function parameters  $\mathbf{w}$  using  $\frac{\partial \beta}{\partial \mathbf{w}}$ , which the learning framework being used must

be able to calculate. The sensitivity calculation for an integrated inference and learning approach can, thus, be calculated similar to what has been shown above for field inversion

$$\frac{d\mathcal{J}}{d\mathbf{w}} = \left\{ \left[ \frac{\partial \mathcal{J}}{\partial \beta} \right]_{\tilde{\mathbf{u}}_m} - \left[ \frac{\partial \mathcal{J}}{\partial \tilde{\mathbf{u}}_m} \right]_{\mathbf{w}} \left[ \frac{\partial \mathcal{R}}{\partial \tilde{\mathbf{u}}_m} \right]_{\mathbf{w}, \xi}^{-1} \left[ \frac{\partial \mathcal{R}}{\partial \beta} \right]_{\tilde{\mathbf{u}}_m, \xi} \right\} \frac{\partial \beta}{\partial \mathbf{w}}.$$

## Appendix B: Results when training only with T3A

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

**Optimization convergence and feature space contours:** Although the convergence (Figure 19(a)) appears similar to that observed when both T3A and T3C1 were simultaneously used to learn the augmentation, the skin friction predictions (Figure 19(b)) and augmentation contours in the feature space (figure 20) have significant differences between the two cases. The effect of this difference between the two augmentations can be seen in the following sections. It is however, noteworthy how a single additional case with unseen physical behavior can make a considerable difference in the predictions. This suggests the ability of the framework to effectively extract information about the features-to-augmentation functional relationship.

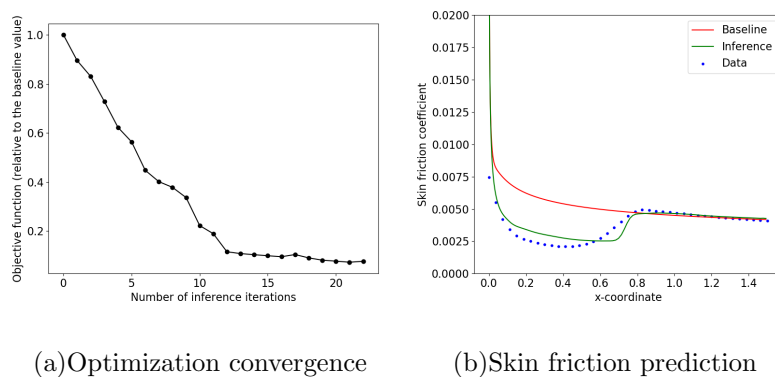


FIG. 19: Augmentation training using data from T3A only

**Predictions on T3 cases:** It can be seen in Figure 21 that the prediction is completely wrong for T3C1 and T3B, while it appears reasonable for the other cases. Another important observation is that the transition locations for T3C1, T3C2 and T3C3 are over-predicted.

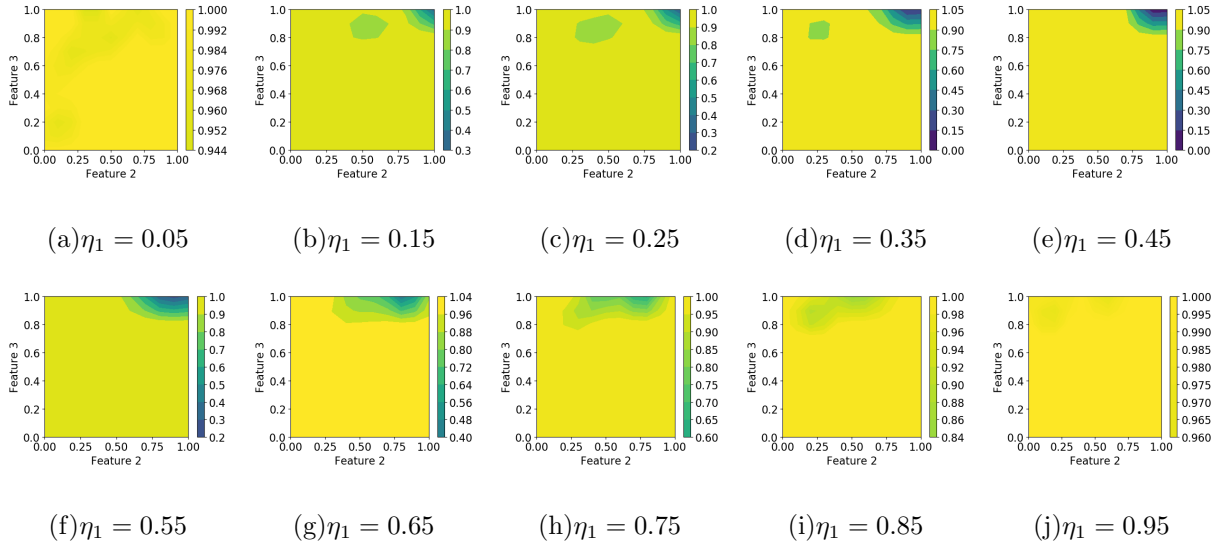


FIG. 20: Feature maps (x-axis:  $\eta_2$ , y-axis:  $\eta_3$ , uniform color-bar range  $[0,1]$  across all plots)

This is due to the fact that the augmentation remains at a lower value for a longer distance spuriously as the augmentation has no way of differentiating between how  $\eta_1$  changes for different pressure gradients as the training is performed only for a zero pressure gradient case.

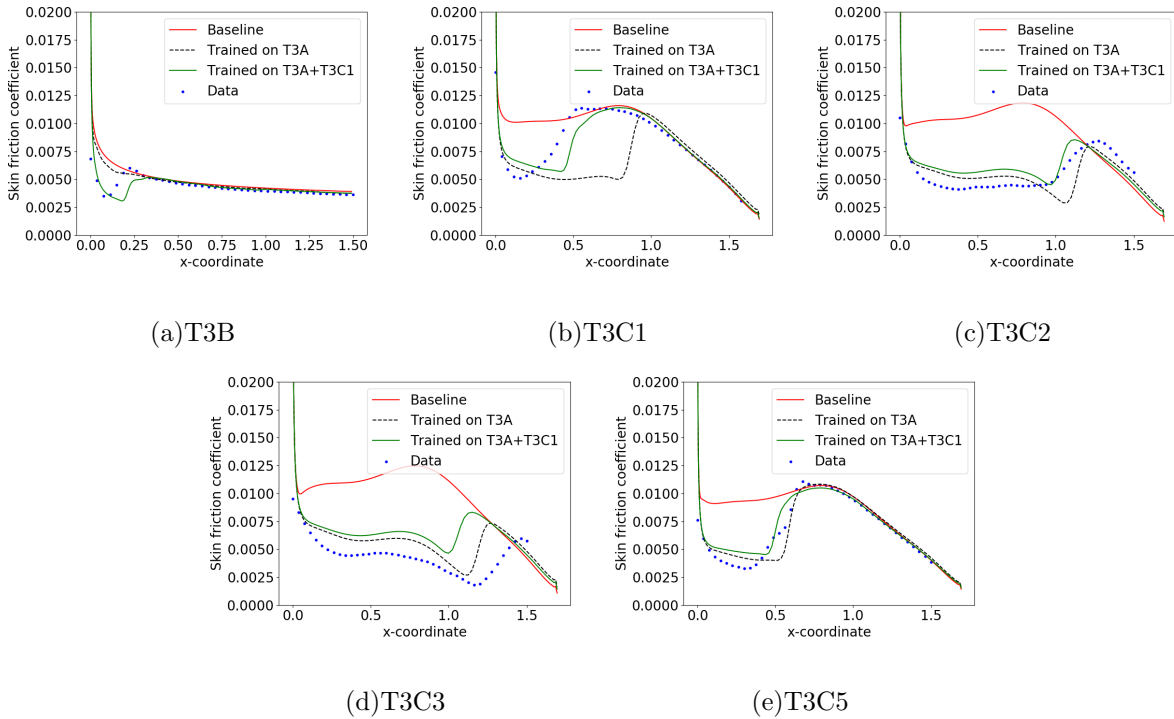


FIG. 21: Skin friction coefficients for T3C cases

**Prediction on VKI cases** As can be seen in Figure 22, training a model only on the T3A data results in significantly inaccurate transition location predictions on at least one side of the blade except MUR224 when compared to the results presented in the main text where both T3A and T3C1 cases were used for training. This is in accordance with the explanation provided in the last section. Since the transition model has little information on how the features behave in the presence of non-zero pressure gradients, additional data which can highlight such behavior (in the main text as the T3C1 case) is required to extract information about the behavior of the augmentation in feature space.

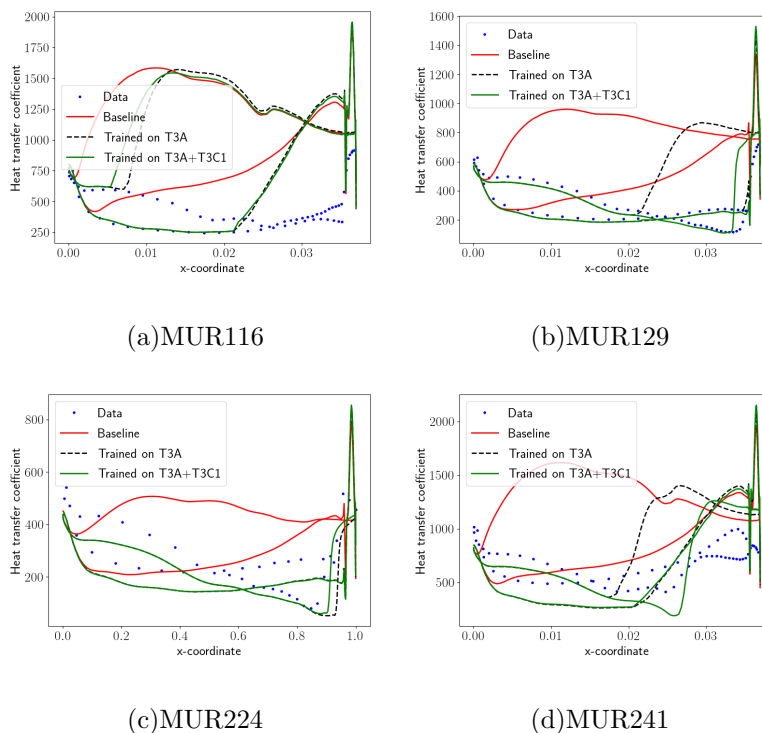


FIG. 22: Heat transfer coefficients for VKI cases

### Appendix C: Results with varying preset distance intervals for $\overline{Re_{\theta,t}}$ extraction

As shown in Figure 23, we found that varying the preset distance usually has a small effect on the predictions for the turbine blades. A minor discrepancy is observed in MUR116 (characterized by a small bump in the heat transfer coefficient), whereas a major discrepancy, resulting in considerable different transition behavior, is seen for MUR241.

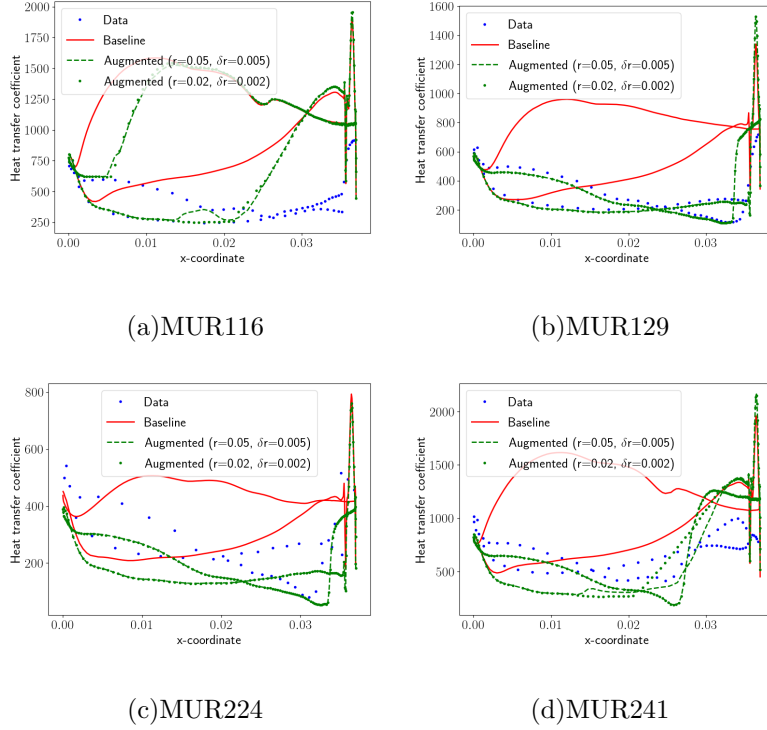


FIG. 23: Comparison between different preset distance intervals

#### Appendix D: Results without localized learning (Neural Networks)

Three different fully connected neural network architectures were utilized as shown in table IV. Owing to the simple nature of augmentation behavior in feature space, a simple neural network with two hidden layers containing seven nodes each was tested. The results obtained during training, as shown in figure 24, are clearly poor as they partially laminarize the entire flow rather than laminarizing the flow only before the transition location. Further, two other architectures with 1 hidden layer containing 600 nodes, and 2 hidden layers containing 45 and 60 nodes respectively were tried out but yielded similarly poor results as shown in figure 24. The number of parameters (weights and biases) for the latter two architectures was kept around 3000 to match the number of parameters used in the main text during localized learning via feature-space discretization.

<b>Network</b>	# of hidden layers	Nodes in hidden layer 1	Nodes in hidden layer 2
Neural Network 1	2	7	7
Neural Network 2	1	600	-
Neural Network 3	2	45	60

TABLE IV: Description of neural network architectures

The results demonstrate that localized learning is essential to both, better condition the optimization problem and also to aid the optimization trajectory in the correct direction in the augmentation space. The second point is especially important, as in several applications (including transition prediction), changing the augmentation behavior in regions where no data is available can in fact lead to spurious model behavior as these parts of the feature space can be accessed before the solver converges. This is further worsened by any feedback present between features and augmentation. It has to be mentioned that using the same software framework, neural networks have been used in the context of integrated inference and learning more successfully in Refs. [13, 14] on a less challenging problem. The authors acknowledge that more sophisticated training methods may yield better results.

### **Appendix E: Results with a finer discretization in the feature space**

For comparison purposes, a finer feature-space discretization was also used to obtain the augmentation function, the training results and augmentation contours on feature-space slices for which have been shown in figures 25 and 26. The feature space was divided into subdomains of size  $1/30$  along all three feature space directions (90, 30, and 30 cells along the first, second, and third features respectively). As can be noticed in figure 26, the influence of the changes made by the data have been restricted to smaller regions owing to the smaller cell sizes. Figures 27 and 28 show the results from testing the augmentation on the T3B, T3C2, T3C3, T3C5, MUR116, MUR129, MUR224, and MUR241 cases. As can be seen from the results, while the augmentation learned on the finer grid seems to predict the transition locations for T3 cases with nearly similar accuracy (with little laminarization in the T3B case and slightly premature transition in T3C5) as its counterpart trained on the coarser grid, almost all results from the VKI cases exhibit premature transition. This



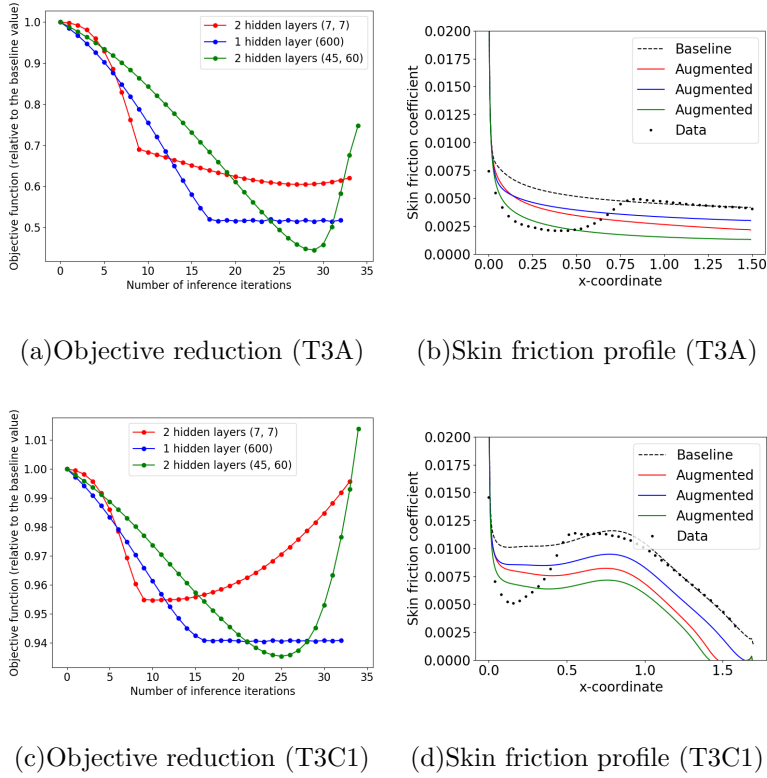
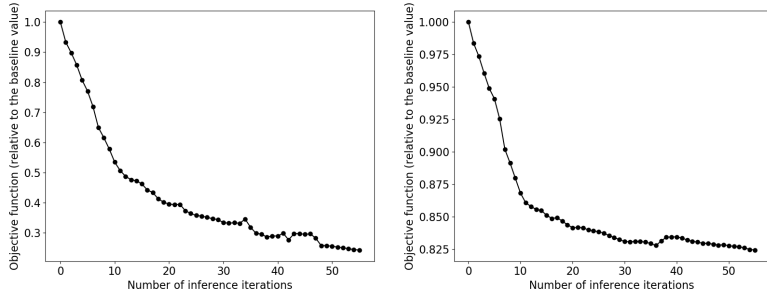
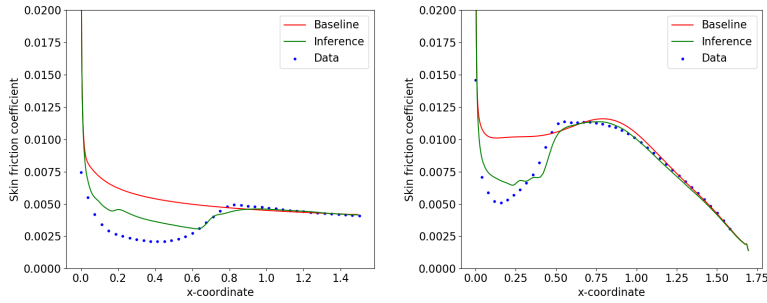


FIG. 24: Training results with Different Neural Network Architectures

happens because the limited region of influence that the available data has in the feature space allow some feature space locations to exhibit baseline behavior (which did not happen when using the augmentation learned on a coarser grid as the region of influence covered a larger part of the feature space).

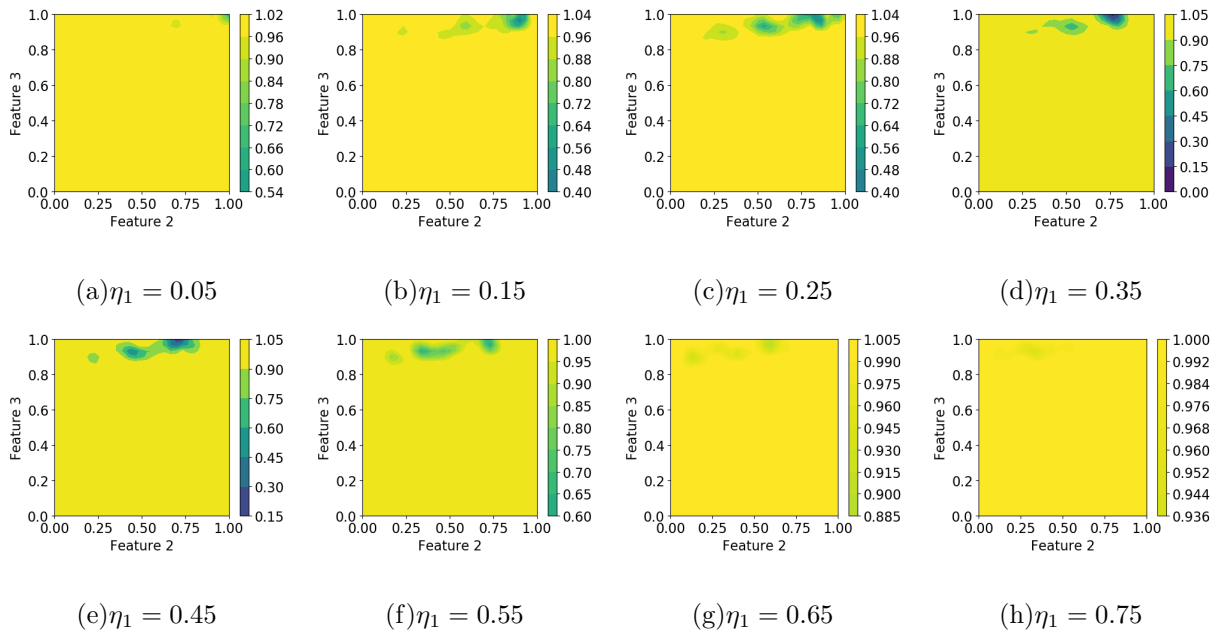


(a) Optimization history (T3A)      (b) Optimization history (T3C1)



(c) Skin friction profile (T3A)      (d) Skin friction profile (T3C1)

FIG. 25: Training results on a finer feature-space grid



(a)  $\eta_1 = 0.05$       (b)  $\eta_1 = 0.15$       (c)  $\eta_1 = 0.25$       (d)  $\eta_1 = 0.35$   
 (e)  $\eta_1 = 0.45$       (f)  $\eta_1 = 0.55$       (g)  $\eta_1 = 0.65$       (h)  $\eta_1 = 0.75$

FIG. 26: Augmentation contours on feature-space slices

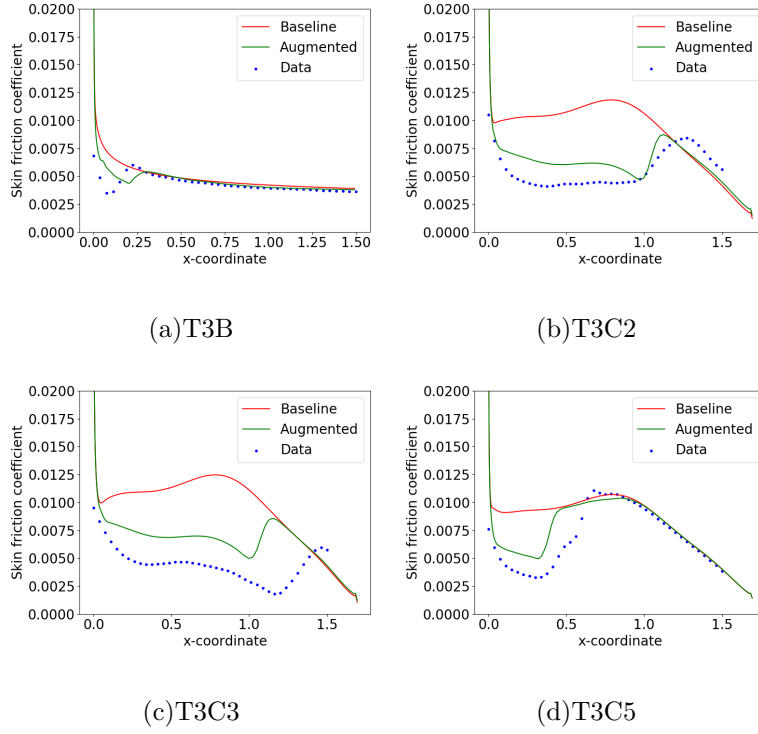


FIG. 27: Skin friction coefficient profiles for the T3 test cases

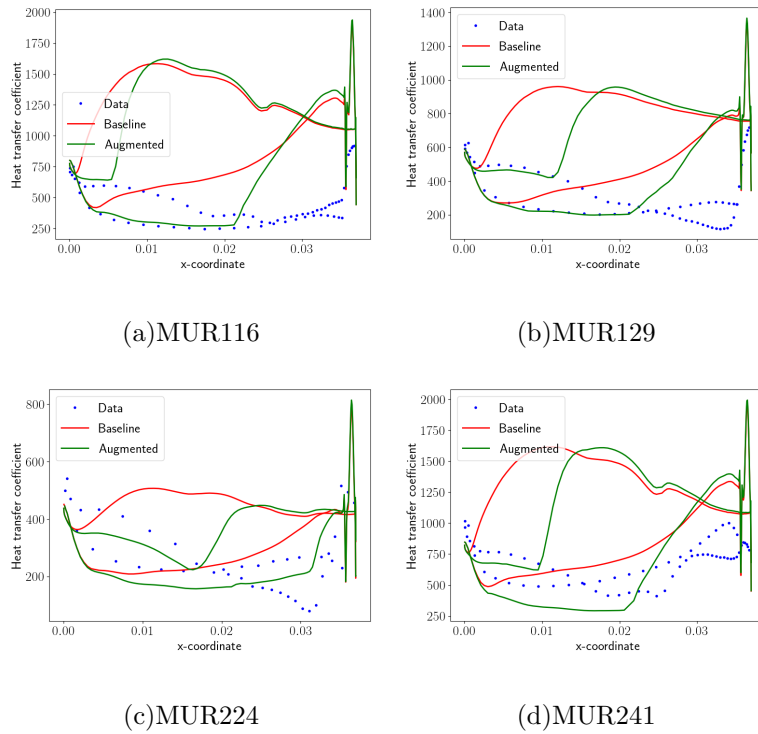


FIG. 28: Heat transfer coefficient profiles for the VKI test cases