

Machine Learning Approaches to Data–Driven Transition Modeling

Muhammad Irfan Zafar

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Aerospace Engineering

Heng Xiao, Chair
Meelan Choudhari, Co-chair
K. Todd Lowe
Christopher J. Roy

April 5, 2023
Blacksburg, Virginia

Keywords: Laminar–Turbulent Transition, Machine Learning, Convolutional Neural
Network, Recurrent Neural Network, Neural Operators
Copyright 2023, Muhammad Irfan Zafar

Machine Learning Approaches to Data-Driven Transition Modeling

Muhammad Irfan Zafar

(ABSTRACT)

Laminar-turbulent transition has a strong impact on aerodynamic performance in many practical applications. Hence, there is a practical need for developing reliable and efficient transition prediction models, which form a critical element of the CFD process for aerospace vehicles across multiple flow regimes. This dissertation explores machine learning approaches to develop transition models using data from computations based on linear stability theory. Such data provide strong correlation with the underlying physics governed by linearized disturbance equations. In the proposed transition model, a convolutional neural network-based model encodes features from boundary layer profiles into integral quantities. Such automated feature extraction capability enables generalization of the proposed model to multiple instability mechanisms, even for those where physically defined shape factor parameters cannot be defined/determined in a consistent manner. Furthermore, sequence-to-sequence mapping is used to predict the transition location based on the mean boundary layer profiles. Such an end-to-end transition model provides a significantly simplified workflow. Although the proposed model has been investigated for two-dimensional boundary layer flows, the embedded feature extraction capability enables their generalization to other flows as well. Neural network-based nonlinear functional approximation has also been presented in the context of transport equation-based closure models. Such models have been examined for their computational complexity and invariance properties based on the transport equation of a general scalar quantity. The data-driven approaches explored here demonstrate the possibilities for improved transition prediction models.

Machine Learning Approaches to Data-Driven Transition Modeling

Muhammad Irfan Zafar

(GENERAL AUDIENCE ABSTRACT)

Surface skin friction and aerodynamic heating caused by the flow over a body significantly increases due to the transition from laminar to turbulent flow. Hence, efficient and reliable prediction of transition onset location is a critical component of simulating fluid flows in engineering applications. Currently available transition prediction tools do not provide a good balance between computational efficiency and accuracy. This dissertation explores machine learning approaches to develop efficient and reliable models for predicting transition in a significantly simplified manner. Convolutional neural network is used to extract features from the state of boundary layer flow at each location along the body. These extracted features are then processed sequentially using recurrent neural network to predict the amplification of instabilities in the flow, which is directly correlated to the onset of transition. Such an automated nature of feature extraction enables the generalization of this model to multiple transition mechanisms associated with different flow conditions and geometries. Furthermore, an end-to-end mapping from flow data to transition prediction requires no user expertise in stability theory and provides a significantly simplified workflow as compared to traditional stability-based computations. Another category of neural network-based models (known as *neural operators*) is also examined which can learn functional mapping from input variable field to output quantities. Such models can learn directly from data for complex set of problems, without the knowledge of underlying governing equations. Such attribute can be leveraged to develop a transition prediction model which can be integrated seamlessly in flow solvers. While further development is needed, such data-driven models demonstrate the potential for improved transition prediction models.

Acknowledgments

It has been a challenging journey through my doctoral studies, and I am grateful for the support of so many people along the way!

This work would not have been possible without my advisor, Dr. Heng Xiao. His consistent support, constructive feedback, and advice have been very helpful in my growth as a researcher and conducting PhD research. I also feel very lucky to have Dr. Meelan Choudhari as a mentor for the majority of my PhD research. His gracious support, invaluable feedback, and guidance have been key factors towards my learning and progress during these years. I was also fortunate to have learned a great deal from Dr. Christopher Roy through his guidance in research and the courses he taught. I greatly benefited from attending all four of the graduate level courses he teaches at Virginia Tech. Thank you also to Dr. Todd Lowe for providing me with his feedback and advise, and for being part of my PhD committee.

I would also like to thank my colleagues: Dr. Carlos A. Michelén Ströfer with whom I shared office space for two years and learned a great deal from regular discussions, and Xu-Hui Zhou who I worked closely with, discussing, learning, and exploring together. I would also like to acknowledge Dr. Jiequn Han, whose support and feedback was instrumental in parts of my research.

Special thanks go to my family in Pakistan for their unwavering support, love, and patience during these challenging years of my doctoral studies. It has been over four years since I last met most of my family, and I couldn't be happier to return after completing my PhD program. I would also like to express my gratitude to my friends, especially those in Blacksburg, with whom I had a wonderful and memorable time.

Finally, I would like to express my special thanks to the US State Department for funding my PhD studies through the Fulbright fellowship. It served as a tremendous source of support throughout my doctoral studies. I am also grateful to NASA and the Virginia Tech College of Engineering for their funding support.

Contents

1	Introduction	1
1.1	Transition Mechanisms – an Overview	1
1.1.1	Natural Transition	2
1.1.2	Bypass Transition	4
1.1.3	Separation–Induced Transition	5
1.1.4	Other Transition Types	5
1.2	Transition Prediction Methods	6
1.2.1	Physics-based Methods	6
1.2.2	Transition Prediction Modeling	9
1.3	Machine Learning for Transition Modelling	11
1.4	Structure and Content	12
1.5	Key Contributions	13
1.6	Publications	14
1.7	Attributions	14
	Bibliography	15
2	Convolutional neural network for transition modeling based on linear stability theory	23
2.1	Introduction	24
2.2	Methodology	26
2.2.1	The e^N method	27
2.2.2	Convolutional neural network	28
2.2.3	Generation of training data	31
2.3	Results	32
2.3.1	Demonstration of predictive performance	33

2.3.2	Automatic data-driven feature extraction of boundary layer profiles	36
2.3.3	Comparison to a fully connected network with profile inputs	39
2.4	Conclusion	41
	References	42
3	Recurrent neural network for end-to-end modeling of laminar-turbulent transition	45
3.1	Introduction	47
3.2	Recurrent Neural Network	50
3.3	Database of Linear Amplification Characteristics for Airfoil Boundary Layers	54
3.4	Results	56
3.4.1	Demonstration of predictive performance	57
3.4.2	Evaluation of predictive performance for complete database	60
3.4.3	Working with limited database	67
3.5	Conclusion	70
	References	72
	Appendix	74
4	Neural Operators	78
4.1	PDE-based Transition Models	78
4.2	Neural Operators	79
4.2.1	Neural Operators for Transition Modeling	80
	Bibliography	81
5	Frame Invariance and Scalability of Neural Operators for Partial Differential Equations	84
5.1	Introduction	85
5.2	Methodology	90
5.2.1	Generation of data for training and testing	90
5.2.2	Graph kernel network	93

5.2.3	Vector cloud neural network	95
5.2.4	Comparative analysis of GKN and VCNN	97
5.3	Results	98
5.3.1	Frame invariance for GKN	99
5.3.2	Comparison of GKN and VCNN	102
5.4	Conclusion	106
	Appendix A. Performance of an improved VCNN	107
	Appendix B. Neural network architectures details	108
	References	109
6	Conclusions and Future Work	114
6.1	Machine Learning for Stability-based Methods	115
6.2	Neural Operators for PDE-based Transition Models	115
	Bibliography	116

Chapter 1

Introduction

Transition from laminar to turbulent flow significantly alters the fluid behavior, which can have critical impact from a practical standpoint. Laminar flow is characterized by orderly motion of fluid particles with little or no mixing. In contrast, turbulent flow is characterized by chaotic changes, where unsteady vortices (known as eddies) of different sizes interact with each other, consequently leading to enhanced mixing and increased viscous effects. The transition process applies to nearly every type of fluid flow and is most often studied for boundary-layer flow over a solid surface.

The aerodynamic performance of aerospace vehicles can be strongly impacted by laminar–turbulent transition for a variety of applications across different flow regimes. Turbulent boundary layers may have heating rates several times higher as compared to laminar boundary layers. Even in the transitional region, heat transfer increases rapidly and can reach values higher than those in fully turbulent flow [1]. Such strong levels of aerodynamic heating threaten the structural integrity of hypersonic re-entry vehicles. For commercial transport aircraft, delaying transition to turbulent flow via various techniques of laminar flow control typically leads to higher fuel efficiency due to reduced drag. Such performance improvement has been central to the push towards greener technologies in the arena of aerospace research. Boundary layer transition strongly influences the aerodynamic properties of wind turbine blades, which can reduce the power output and energy capture significantly. Similarly, the design of submarines, gas–turbine–engine blades, and subsonic and supersonic commercial and military aircraft is strongly influenced by the boundary layer transition due to large differences in aerodynamic properties between the laminar and fully turbulent regimes.

The significant differences between laminar and turbulent flow, both from a physics and practical engineering standpoint, highlight the need to understand and accurately predict the transition process. Consequently, efficient yet robust modeling of transition forms a critical component in computational fluid dynamics (CFD) for simulating viscous flows [2].

1.1 Transition Mechanisms – an Overview

Laminar–turbulent transition has been a subject of investigation for as long as the phenomenon has been recognized, tracing back to the classical experiments conducted by Reynolds [3]. The transition process is often viewed as the outcome of the loss of stability of the laminar

boundary layer, resulting in the amplification of disturbances generated via the interaction of the boundary layer flow with its disturbance environment. Morkovin [4] categorized the different paths to transition. When a laminar boundary layer develops along a body, transition to turbulence starts with the receptivity process where external disturbances excite the instability modes. These forcing disturbances may exist in the freestream (turbulence or noise) or be generated by the body itself (surface roughness, vibrations etc.). Depending on the external disturbances, model geometry and the flow details, transition to turbulence takes one of the paths [4], attributed as different transition mechanisms, which culminate with the breakdown to fully turbulent flow. These transition mechanisms are categorized and briefly introduced in this section.

1.1.1 Natural Transition

Natural transition corresponds to a low freestream disturbance environment with freestream turbulence intensity (Tu_∞) of less than 0.1% (typical flight conditions and wind tunnels with well-conditioned flow) where slow amplification of unstable waves leads to transition. Under benign disturbance environments, the process of natural transition can be subdivided into three steps.

- **Receptivity** where eigenmodes of a laminar boundary layer are excited as a result of its interaction with the disturbance environment. These disturbances/fluctuations provide the initial conditions for the eventual breakdown of laminar flow.
- The **linear stability** stage involves the amplification of the primary modes of instabilities inside the boundary layer flow. The amplification in this stage is closely influenced by the disturbance environment as well as the pressure gradient in the streamwise direction. Sufficient amplification of the instability waves in this stage triggers secondary and higher order instabilities.
- The last stage involves **nonlinear interactions** between instability modes resulting in the onset of secondary and higher order instabilities, which excite wide range of scales and frequencies in the flow, consequently leading to rapid breakdown to fully turbulent flow. The breakdown process involves the development of turbulent spots, which amplify to finally merge into a fully turbulent flow.

A visualization of a boundary layer flow undergoing natural transition has been shown in the Fig. 1.1. Slow amplification of linear instability waves can be observed happening over a long region (in the left half of the figure), successively followed by the occurrence of nonlinear instability waves, the appearance of turbulent spots and breakdown to turbulence.

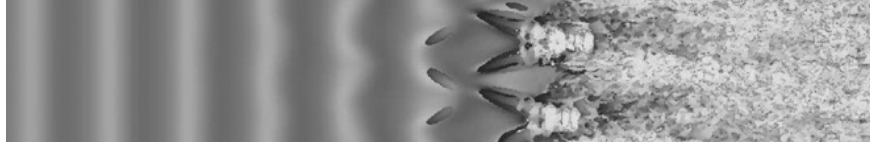


Figure 1.1: Visualization of a flat plate boundary layer flow undergoing natural transition. Contours show the wall-normal velocity perturbations. Figure adapted from Durbin [5] with permission.

Natural Transition Characteristics of Low-speed Flows

For two-dimensional boundary layers in subsonic flows, typical primary modes of instability waves correspond to Tollmien-Schlichting (TS) waves. For three-dimensional boundary layer flows, more complex mechanisms can be observed [6, 7]. For example, in case of a swept wing, the transition process is known to be influenced by centrifugal instabilities on concave walls, and by attachment line and crossflow instabilities. These instability mechanisms are briefly introduced here.

TS waves are the streamwise viscous instabilities, travelling downstream as spanwise oriented vorticity structures. TS waves were first theoretically studied by Tollmien [8] and Schlichting [9] for the Blasius boundary layer, and experimentally observed for the first time by Schubauer and Skramstad [10]. TS waves exist in zero or positive pressure gradient regions where they experience exponential growth of instability waves corresponding to some of the frequencies while decay for the rest. After undergoing slow amplification, initially two-dimensional TS waves become three-dimensional, leading to the emergence of turbulent spots and culmination into a fully turbulent flow. A favorable (negative) pressure gradient has a stabilising effect on TS waves, which is used in designing natural laminar flow airfoils [11]. TS waves are sometimes also referred to as *streamwise instabilities*.

Crossflow instability is a feature of three-dimensional boundary layers which can develop over swept wings or rotating disks due to presence of crossflow profiles. On swept wings, such an instability arises due to the development of velocity component in the sweep direction. Such velocity component results from the imbalance in the boundary layer flow between constant pressure gradient and reduction of streamwise velocity near wall to zero [12]. An inflection point exists in the crossflow profile, which is a necessary condition for the excitation of crossflow instability. Such an instability becomes dominant as crossflow increases. Such an effect can be observed on the swept wings for which the transition location moves forward towards the leading edge as sweep angle of the wing increases.

Görtler vortices are three-dimensional flow instabilities that can occur in boundary layers of a rotating cylinder or a curved surface. They are named after the German physicist Hermann Görtler, who first described the phenomenon in 1940 [13]. In a boundary layer, fluid flows along a solid surface and slows down due to friction. The Görtler vortices arise when the boundary layer flow is subject to a centrifugal force, as in the case of a curved

surface with a convex curvature. The centrifugal force causes the fluid to move outward, and this sets up a secondary flow that consists of a series of vortices that are oriented along the surface. Görtler vortices can appear close to the leading edge stagnation point, where a concave flow situation arises [14]. It seems unclear, however, whether Görtler vortices can lead directly to transition in itself or can act only in aid with other instability mechanisms for transition.

Attachment–line instability corresponds to the non-zero velocity at the leading edge of the swept wings and vertical or horizontal stabilizers. Secondary and tertiary attachment lines might also exist on the fuselage sides. Boundary layer flow due to this non-zero velocity along the attachment line can be unstable, and can undergo transitional state.

Natural Transition Characteristics of High Speed Flows

At high-speed flows, where compressibility effect dominate, transition physics is considerably more complex as compared to that in incompressible flows. As shown by Mack [15] through compressible stability analysis, such added complexity can be attributed to three major factors: dominance of three-dimensional viscous perturbations, presence of generalized inflection point and different acoustic modes (Mack modes) leading to rapid breakdown to turbulence [16]. For supersonic flows with freestream Mach number less than 2, nonlinear interplay between two oblique instability waves, with identical but opposing wave angles, can lead to exponential growth of oblique instability. Such rapid mechanism leads to transition earlier than the classical instability mechanisms. For hypersonic flows, transition process is dominated by the first and second Mack modes [15]. Generally, the first mode dominates in the lower range of Mach numbers and is analogous to TS waves by character and behavior. Second mode dominates in the higher range of Mach numbers. The cross-over point between the dominance of these two modes is dependent on the flow geometry and boundary conditions.

1.1.2 Bypass Transition

Bypass transition corresponds to the strong disturbance environments (high surface roughness, high freestream turbulence intensity ($Tu_\infty > 1$), etc.) [17]. In such cases, longitudinal vortices are generated by freestream turbulence. Presence of longitudinal vortices in laminar boundary layer can lead to rapid appearance of streamwise streaks. Transition to turbulence occurs when the energy of the streaks grow significantly. Fig. 1.2 shows an example of smoke visualization for a boundary layer flow over a flat plate with $Tu = 6.6\%$, where the streamwise streaks leading to transition are clearly visible. Classical processes of modal interactions are bypassed and breakdown to turbulence under bypass transition can occur at the Reynolds numbers lower than those predicted by the classical linear stability theory. Roughness elements can also play a significant role in triggering streamwise streaks leading



Figure 1.2: Boundary layer flow undergoing bypass transition. The contours show streamwise velocity perturbations. Figure adapted from Durbin [5] with permission.

to transition. Bypass transition has been studied extensively, partly due to its relevance to turbomachinery related flow and unavoidable noisy ground tests for aircraft.

1.1.3 Separation–Induced Transition

In many cases, the boundary layer can often detach and go through a transitional phase under the influence of a strong adverse pressure gradient. This occurs as the disturbances in the unstable laminar layer amplifies. Transition to turbulence leads to enhanced mixing and momentum transfer in the wall–normal direction, resulting in the elimination of reverse flow and the reattachment of boundary layer. This whole phenomenon leads to the presence of a re–circulation zone, referred to as *laminar separation bubble*. Primary instability in such a case stems from the inflection velocity profile upstream of the separation point caused by the adverse pressure gradient [18]. Separation–induced transition is strongly dependent on the Reynolds number. Decreasing the Reynolds number has the influence of increasing viscous damping effects, suppressing transition and, consequentially, delaying reattachment [19].

1.1.4 Other Transition Types

To cover all the possible transition mechanisms, two additional types are mentioned here.

Leading edge contamination phenomenon is associated to the contamination of flow due to the existence of turbulence at the leading edge. Such phenomenon commonly occurs around swept wings with large leading edge radius. It was observed for the first time during experimentation of the swept wings in 1950s. In the swept wings, a strong velocity component in the spanwise direction along the leading edge is induced by the geometrical sweep of the wing. The streamwise component of the boundary layer flow can be strongly effected by the presence of turbulence in the spanwise flow along the leading edge. The turbulence at the leading edge can stem from turbulent attachment line, high freestream disturbance environment or turbulent boundary layer around the fuselage [20]. Such contamination of flow can effect the transition in chord–wise direction of the wing, unless disturbances are suppressed by the strong presence of favorable pressure gradient [21, 22].

Relaminarization is the process of reverse transition process, where the flow undergoes

transition from turbulent to laminar. Narasimha [23] categorizes relaminarization mechanisms into three types. Reynolds number relaminarization can occur due to a decrease in local Reynolds number based on the boundary layer edge velocity. Richardson relaminarization can occur if a flow has to contend with buoyancy or curvature forces. Acceleration relaminarization happens under the influence of strongly accelerated boundary layer flow [24].

1.2 Transition Prediction Methods

1.2.1 Physics-based Methods

Direct Numerical Simulation

Navier–Stokes equations completely describe the flow behavior through equations representing the conservation of mass, momentum and, if required, energy. *Direct numerical simulation* (DNS) refers to a solution obtained for the complete non-linear, time-dependent Navier–Stokes equations through numerical methods. Such numerical solution does not involve any model to resolve flow instabilities in pre-transition laminar region, their nonlinear interactions in the transition region and finally the post-transition turbulent region. Consequently, transition onset due to all possible mechanisms can be predicted through DNS as it provides a complete space–time history of the flow field. The accuracy of DNS hinges on the necessary condition to resolve spatial and temporal scales not only for the broad spectrum of turbulence but also for the tiny–amplitude instabilities in the pre-transition laminar region. Such stringent requirements of scales’ resolution necessitates immense computational resources, which are impractical for most of the applications. However, DNS has proven to be a valuable resource of data that has been used for the understanding of transition mechanisms, going back as early as 1980s, as discussed by Kleiser et al. [25]. Accurate data from DNS has shown the ability to challenge classical transition theories and provide new insights in flow physics, while having cheaper and better data acquisition than classical experiments [26]. Furthermore, DNS data primarily serve as a valuable resource for the calibration and validation of simplified transition models. Nevertheless, direct use of DNS for transition prediction are significantly constrained by required computational resources for foreseeable future, which necessitates the development of other methods for transition prediction.

Large Eddy Simulation

Large eddy simulation (LES) resolves only the important large scales, while the behavior of small scales are *modeled* using subgrid–scale models [27, 28]. As a result, this approach results in considerably reduced computational cost, compared to DNS, due to less resolved grid. For transitional wall–bounded flows, Piomelli [29] demonstrated that LES results on a grid of $23 \times$

48×24 points shows good agreement with the DNS results on a grid of $144 \times 144 \times 224$ points. Recently, Jee and coworkers have also demonstrated significant alleviation of computational cost using the stability analysis in the pre-transition region [30, 31, 32, 33]. Although, flow physics of large scale is preserved in LES, modeling of subgrid scale effects has its consequence. Transition prediction using LES has shown significant sensitivity towards the choice of subgrid-scale model. For example, the choice of Smagorinsky constant to determine the subgrid eddy viscosity has significant effect on the predicted transition location [34, 35]. Other subgrid scale models [36, 37, 38] and different formulations of LES [39, 40] have been proposed to overcome these challenges, many of which have shown their limitations as well. Nevertheless, the computational cost of simulating transition using LES is still prohibitively high for practical applications. Furthermore, specification of boundary conditions and initial solution provides with further limitations towards the use of LES. Transition prediction is an initial value problem and different freestream conditions (noise or turbulence) can strongly effect the transition predictions.

Linear Stability Theory and e^N Method

Linear stability theory (LST) involves tracking the linear amplification of instability waves characterized by respective frequencies. The onset of transition occurs only after the amplitude of the most amplified wave has grown sufficiently. For two-dimensional boundary layer flow in a curvilinear coordinate system (s, n) , the governing equations of disturbances (fluctuations) can be obtained from Navier-Stokes equations by decomposing flow variables to mean-flow and fluctuating components and subtracting the undisturbed basic flow equations. Small amplitude, spatially evolving disturbances (u, v) to the mean flow (U, V) can be represented in a wave-form as:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} U(n) \\ V(n) \end{bmatrix} \exp(i[\varphi(s) - \omega t]) \quad (1.1)$$

where ω and $d\varphi/ds = \alpha$ represent the real-valued frequency and complex streamwise wave number of the disturbance wave, respectively. $i = \sqrt{-1}$ is the imaginary unit. Substituting the above representation in *linearized* governing equations of disturbances and assuming parallel flow ($U \approx U(n)$ and $V \approx 0$) yields quasi-parallel form of disturbance equations. For an incompressible flow, the quasi-parallel disturbance equations can be combined into a well-known Orr-Sommerfeld equation, which can be represented in non-dimensionalized form as:

$$(\alpha U - \omega) (V'' + \alpha^2 V) - \alpha U'' V = (V'''' - 2\alpha^2 V'' + \alpha^4) / (i Re_\theta) \quad (1.2)$$

where prime denotes the derivative with respect to the wall-normal coordinate (n) and Re_θ represents momentum thickness-based Reynolds number. Orr-Sommerfeld equation has been named after Orr [41, 42] and Sommerfeld [43] who independently studied the stability theory of fluid dynamics and arrived at this equation in early 1900s. Equation 1.2, along with appropriate boundary conditions, constitutes an eigenvalue problem to determine local

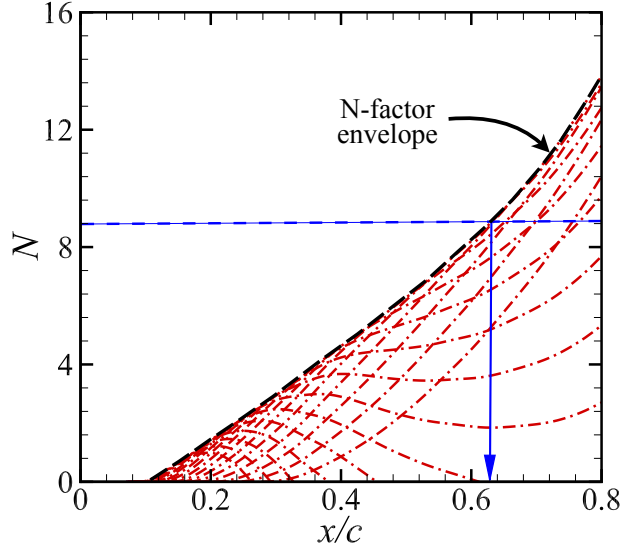


Figure 1.3: N-factor curves (in red) for different frequencies and corresponding N-factor envelope (in black) are plotted along the airfoil surface. Transition location (shown with blue arrow) is related to a threshold value of $N=9$.

values of wave number α for each of the instability wave characterized by frequency parameter ω . These computed wavenumbers determines whether an instability wave will grow or decay, and the local amplification rate of an instability wave is given as:

$$\sigma = -\text{Im}(\alpha(\omega, s))$$

where $\text{Im}(\cdot)$ denotes the imaginary part of a complex quantity. Amplification rate of a disturbance is then integrated along a streamline to obtain logarithmic amplification of the disturbance amplitude as:

$$N(\omega, s) = \int_{s_0}^s \sigma(\omega, \tilde{s}) d\tilde{s}$$

where s_0 represents a neutral station where the disturbance first started amplifying.

While the amplification of linear instabilities can be predicted by Orr-Sommerfeld equation, prediction of transition onset requires some method based on these computed instabilities. Smith and Gamberoni [44] and van Ingen [45] developed a method, known as the e^N method, which leverages the empirically defined thresholds for the most amplified instability wave, at which location transition to turbulence occurs. Such illustration has been made in Fig. 1.3, where N-factor curves corresponding to different frequencies are plotted and N-factor envelope is defined by most amplified instability at each location.

1.2.2 Transition Prediction Modeling

Low-Reynolds Number Models

Low Reynolds-number turbulence models have shown certain ability to predict transition process, especially bypass transition [46]. Such ability can be attributed to the mathematical nature of the turbulence model equations [47]. For k - ω turbulence model, transport equations of turbulent kinetic energy (k) and specific turbulence dissipation rate (ω) contain production and dissipation terms, sum of which represents the net production for each turbulent quantity. Onset of transition can be correlated to the evolution of net production term of turbulent kinetic energy (k) from negative to positive value. Similarly, transitional region can be simulated as the net production of specific turbulence dissipation rate (ω) evolves from negative to position value. Such reasoning can be extended to other turbulence models as well.

Savill [48, 49] analyzed low Reynolds number turbulence models for simulating bypass transition. Non-linear eddy viscosity models have been analyzed for bypass transition by Lardeau et al. [50]. Reynolds stress models have also been analyzed for simulating bypass and separation-induced transition [51, 52]. All these investigations, however, demonstrated that these turbulence models, without special modifications, are not reliable to simulate transition process [16, 53]. Such behavior can be attributed to the fact that the ability of these turbulence models to predict transition in specific cases is not based on the intrinsic physics of transition. To some extent, these models are capable of predicting the statistical behavior of transitional flows, instead of simulating transitional dynamics.

Algebraic Correlation Methods

The earliest correlation methods for transition prediction were algebraic, based on local quantities of interest. Such correlations are generally categorised as local-correlation methods which do not take into account the upstream flow information as required by LST or PSE. Despite this, many of these correlation methods provide a reasonable approximations of transition predictions for several instability mechanisms.

A frequently used correlation method, proposed by Abu-Ghannam and Shaw [54], correlates the Reynolds number based on momentum thickness (Re_θ) with the freestream turbulence intensity (Tu_∞) and local pressure gradient parameter (β). The correlation is given as:

$$Re_{\theta,t} = 163 + \left(F(\beta) - \frac{F(\beta)}{6.91} Tu_\infty \right) \quad (1.3)$$
$$F(\beta) = \begin{cases} 6.91 + 12.75\beta + 63.64\beta^2 & \text{if } \beta \leq 0 \\ 6.91 + 2.48\beta - 12.27\beta^2 & \text{if } \beta > 0 \end{cases}$$

Transition is considered to occur as the computed Re_θ exceeds the value determined from Eq. 1.3. Re_θ values are determined based on fully laminar solution, which is then used to approximate the location of transition onset. Dependence on turbulence intensity of freestream flow allows to account for bypass transition mechanism as well. Earlier, relatively simpler correlations have also been proposed to predict transitional momentum thickness Reynolds number as a function of streamwise Reynolds number [55], or as a function of local pressure gradient and neutral stability location [56]. Correlation based on shape factor, a scalar boundary layer parameter, has also been proposed by Wazzan et. al. [57].

Algebraic intermittency model has also been proposed which makes use of an empirical correlation to describe the streamwise evolution of intermittency (γ) [58]. Transitional flows are characterized by intermittent behavior, where γ is the fraction of time during which the flow is turbulent; $\gamma = 0$ corresponds to laminar flow and $\gamma = 1$ corresponds to fully turbulent flow. The empirical correlation for the evolution of intermittency is given as:

$$\gamma = \begin{cases} 1 - \exp \left[- \left(\frac{-(x-x_t)^2 n \sigma}{U} \right) \right], & (x \geq x_t) \\ 0, & (x \leq x_t) \end{cases} \quad (1.4)$$

where γ is given as function of transition onset location x_t , formation rate of turbulent spots n , propagation rate σ , and the freestream velocity U . Transition onset location x_t can be determined by additional correlations based on freestream turbulence intensity and momentum thickness Reynolds number, few of which have been mentioned earlier. Intermittency factor γ is incorporated algebraically through its multiplication with eddy viscosity, which results in modulation of turbulence through γ in the transitional region. In this manner, such intermittency model provides insight into the flow dynamics within the transitional area, unlike other correlation methods which only predicts the location of transition onset.

Correlation-based methods are generally reliable for the instability mechanisms or flow regimes for which they are calibrated, which limits their generalization capabilities. Furthermore, functional dependence on β and Re_θ makes them difficult to integrate seamlessly with CFD solvers.

Transport Equation Models

PDE-based transport equation models for transition prediction have primarily been proposed to work seamlessly with RANS CFD codes. These models tend to be locally correlated which makes them amenable to parallelization as well as to unstructured mesh applications like other CFD codes. Instead of describing the transition physics, these transport equation models are purposed to mimic the statistical behavior of transitional process. Some of the physics could instead be contained in such models through empirical correlation incorporated in these models' formulations. Such models are briefly introduced here.

- **Laminar Kinetic Energy Model (k_L):** One of the first PDE-based transition model was developed by Walters and Leylek [59] by bifurcating the turbulence kinetic energy

(k) equation, in k - ε turbulence model, to turbulent and laminar components of kinetic energy. Laminar kinetic energy (k_L) represents the streamwise fluctuations before transition to turbulence. These streamwise fluctuations are structurally different from turbulent fluctuations and their intensities can reach several times of freestream level. This model has demonstrated the capability of predicting for both natural and bypass transition with modifications in production terms of transport equations.

- **Intermittency-based Models:** This transition model, developed by Langtry and Menter [17], is based on transport equations for two quantities; intermittency (γ) and transition onset momentum-thickness Reynolds number ($Re_{\theta,t}$). Intermittency (γ) indicates the intermittent behavior of the flow in term of the fraction of time during which flow is turbulent [58]. Intermittency transport equation provides distribution of intermittency across the boundary layer leading to more accurate transition predictions. $Re_{\theta,t}$ serves as a transition-triggering criterion. The combination of these two transport equations are also referred to as γ - Re_{θ} model, whereas γ - Re_{θ} model coupled with SST turbulence model is referred to as 'Langtry-Menter' model. This model makes use of many empirical correlation based on physical reasoning and has shown considerable success in accurate predictions of transition process for wide range of flows [17, 60]. One primary drawback of Langtry-Menter model is the lack of Galilean invariance, as the correlations are explicitly based on velocity vectors [17], for which modification has been proposed to render the model Galilean invariant [61]. Langtry-Menter model has further been recast as one-equation transition model by eliminating $Re_{\theta,t}$ equation, known as γ -model [62]. With additional modification, this model also ensures Galilean invariance. Both γ - Re_{θ} model and γ -model have also been coupled with Spalart-Allmaras turbulence model to reduce the computational expense [63, 64].
- **Amplification Factor Transport (AFT) Model:** This transition model [65, 66] is based on the approximation of envelope e^N method developed by Drela and Giles [67]. AFT models avoids the need for integration paths and nonlocal operations by using correlation based on local pressure-gradient parameter H_L . A single transport equation based on the amplification factor (\tilde{n}) is solved, which accounts for the for the upstream flow history by approximating the growth of N-factor. The source term of the transport equation is dependent on boundary layer shape factor H_{12} which is estimated by using local parameter H_L . AFT transition model has been adapted to be used with Spalart-Allmaras and SST turbulence models.

1.3 Machine Learning for Transition Modelling

Advances in high-performance computing and experimental measurement capabilities have led to the availability of large amounts of data in fluid dynamics, and scientific inquiry has gravitated from first principles towards data-driven approaches. Machine learning (ML)

provides a wealth of modular and agile modeling techniques that can be used to extract information from data and to augment domain knowledge about the underlying physics. In the field of CFD, ML algorithms have shown potential benefits towards enhanced development of reduced order models, improved turbulence modeling, and accelerated high-fidelity simulations [68]. In this research work, similar ML advances are leveraged towards modeling of laminar-turbulence transition.

Many challenges need to be addressed towards achieving a reliable and efficient transition model which can be integrated seamlessly in CFD simulations. Desirable attributes associated with such transition model include its applicability to all relevant transition mechanisms, automated transition prediction, suitability to multiple turbulence models, and compatibility to unstructured grids and parallelizable CFD codes. To this end, transition prediction methods, discussed above, still need to overcome limitations associated with each of them. Linear stability computations are computationally expensive and their non-robust nature requires user expertise in stability theory. Hence they are difficult to integrate in an automated fashion in CFD solvers. Many surrogate models developed are generally based on small set of parameters and lack generalizability to multiple transition mechanisms. Transport-equation based models can be integrated in RANS solvers, however these transition models have limited physics basis. Furthermore, they are designed for selected few transition mechanisms and are suited to only one or two turbulence models.

Different neural network-based models are explored in this work to address a few of these challenges in the context of stability based transition models so that they can be integrated into CFD codes. To the extent that the LST allows direct estimates of the amplification of instability waves via the linearized approximation to the Navier-Stokes equations, the NN-based surrogate models have direct correlation with underlying physics of transition mechanism. The proposed recurrent neural network based model has the potential to enable transition prediction in an automated fashion with greatly simplified workflow.

1.4 Structure and Content

This dissertation follows the *Manuscript* format. The content of each chapter is as follows:

Chapter 1: Introduces the transition problem, presents an overview of the transition mechanisms associated with different flow conditions and geometries, literature review of different transition prediction methods/models, and summarizes the outline, content and contributions of the dissertation.

Chapter 2: Consists of a peer reviewed paper, titled ‘*Convolutional neural network for transition modeling based on linear stability theory*’ and published in Physical Review Fluids (reproduced here with permission from American Physical Society). This paper presents a convolutional neural network based model which processes mean boundary

layer profiles in physics-consistent manner to predict the amplification of instability waves, and uses that information to model the location of transition onset.

Chapter 3: Consists of a peer reviewed paper, titled ‘*Recurrent neural network for end-to-end modeling of laminar-turbulent transition*’ and published in Data-Centric Engineering (reproduced here with permission from Cambridge University Press). This paper presents an end-to-end, recurrent neural network based model which provides a sequence-to-sequence mapping between mean boundary layer data and the amplification of the most amplified instability modes. Such a model provides a greatly simplified workflow for transition predictions, and it has been developed as well as analyzed by using a large dataset for two-dimensional boundary layers over several airfoils.

Chapter 4: Introduces neural operators and discusses their effectiveness in learning NN-based functional representation, mimicking the behavior of transport-equation based closure models for transition.

Chapter 5: Consists of a peer reviewed paper, titled ‘*Frame invariance and scalability of neural operators for partial differential equations*’ and published in Communications in Computational Physics (reproduced here with permission from Global Science Press). This paper presents and examines two neural operators for their invariance properties and computational complexity, based on approximating a transport PDE of a scalar quantity.

1.5 Key Contributions

The main contributions of this dissertation towards the goal of developing efficient and reliable data-driven transition model are listed here:

- Proposed model has the ability to encode information from boundary layer profiles into a set of integral quantities, which show strong correlation with the physically defined shape factor parameter. This should enable the generalization of this model to multiple instability mechanisms for which a shape factor cannot be evaluated in an accurate manner.
- Proposed an end-to-end transition prediction model based on linear stability based correlation, with a significantly simplified workflow requiring no user expertise in stability theory.
- Investigated neural operators to mimic the behavior of transport PDE-based closure models, especially with respect to their invariance properties and computational complexity.

1.6 Publications

Peer-reviewed journal publications included in the dissertation are listed here.

- **Muhammad I Zafar**, Heng Xiao, Meelan M Choudhari, Fei Li, Chau-Lyan Chang, Pedro Paredes, Balaji Venkatachari. *Convolutional neural network for transition modeling based on linear stability theory*. Physical Review Fluids 5 (11), 113903, 2020
- **Muhammad I Zafar**, Meelan M Choudhari, Pedro Paredes, Heng Xiao. *Recurrent neural network for end-to-end modeling of laminar-turbulent transition*. Data-Centric Engineering 2, 2021
- **Muhammad I Zafar**, Jiequn Han, Xu-Hui Zhou, Heng Xiao. *Frame invariance and scalability of neural operators for partial differential equations*. Communications in Computational Physics 32 (2), 336-363, 2022

Other peer-reviewed journal publications published during my Ph.D. studies.

- Pedro Paredes, Balaji Venkatachari, Meelan M. Choudhari, Fei Li, Chau-Lyan Chang, **Muhammad I. Zafar**, and Heng Xiao. *Toward a practical method for hypersonic transition prediction based on stability correlations*. AIAA Journal, 58(10):4475–4484, 2020

1.7 Attributions

For all the journal articles included in this dissertation, I was the major contributor and writer of the manuscript. The contributions of the co-authors of these articles are mentioned here individually.

- *Dr. Heng Xiao* is my Ph.D. advisor. He participated in conceptualization, planning and discussions, provided feedback and guidance, and reviewed and edited my writings.
- *Dr. Meelan Choudhari* is a senior research scientist at NASA Langley Research Center. He participated as co-advisor in conceptualization and discussions of research related to transition. He provided regular feedback and guidance, and reviewed and edited my writings.
- *Dr. Pedro Paredes* is a research engineer at National Institute of Aerospace. He contributed by generating data using NASA computational resources and reviewed my writing.

- *Dr. Jiequn Han* is a research fellow at Flatiron Institute. He worked closely with me in work related to neural operators and reviewed and edited my writing.
- *Xu-Hui Zhou* is a fellow doctoral student. He worked with me in conceptualization and running simulations. He participated in discussions and reviewed my writing.

Bibliography

- [1] H. Schlichting. *Boundary Layer Theory, Seventh Ed.* McGraw-Hill, New York, 1979.
- [2] Jeffrey Slotnick, Abdollah Khodadoust, Juan Alonso, David Darmofal, William Gropp, Elizabeth Lurie, and Dimitri Mavriplis. CFD vision 2030 study: A path to revolutionary computational aerosciences. *NASA/CR-2014-218178*, 2014.
- [3] Osborne Reynolds. An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels. *Philosophical Transactions of the Royal Society of London*, 174:935–982, 1883.
- [4] M. V. Morkovin. On the many faces of transition. In C. Sinclair Wells, editor, *Viscous Drag Reduction*, pages 1–31, Boston, MA, 1969. Springer US. ISBN 978-1-4899-5579-1.
- [5] Paul Durbin. *Advanced Approaches in Turbulence: Theory, Modeling, Simulation and Data Analysis for Turbulent Flows*. Elsevier, 2021. ISBN 978-0-12-820774-1.
- [6] V. Theofilis, A. V. Fedorov, D. Obrist, and U. C. Dallmann. The extended Görtler–Hämmerlin model for linear instability of three-dimensional incompressible swept attachment-line boundary layer flow. *Journal of Fluid Mechanics*, 487:271—313, 2003.
- [7] W S Saric. Görtler vortices. *Annual Review of Fluid Mechanics*, 26(1):379–409, 1994.
- [8] W. Tollmien. Über die entstehung der turbulenz. 1. mitteilung. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1929:21–44, 1928.
- [9] H. Schlichting. Zur entstehung der turbulenz bei der plattenströmung. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1933:181–208, 1933.
- [10] G. B. Schubauer and H. K. Skramstad. Laminar boundary-layer oscillations and stability of laminar flow. *Journal of the Aeronautical Sciences*, 14(2):69–78, 1947.

- [11] William S. Saric, Andrew L. Carpenter, and Helen L. Reed. Passive control of transition in three-dimensional boundary layers, with emphasis on discrete roughness elements. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 369(1940):1352–1364, 2011.
- [12] William S. Saric, Helen L. Reed, and Edward B. White. Stability and transition of three-dimensional boundary layers. *Annual Review of Fluid Mechanics*, 35(1):413–440, 2003.
- [13] Henry Görtler. Über den einfluß der wandkrümmung auf die entstehung der turbulenz. *Zamm-zeitschrift Fur Angewandte Mathematik Und Mechanik*, 20:138–147, 1940.
- [14] Ernst Heinrich Hirschel, Jean Cousteix, and Wilhelm Kordulla. *Three-Dimensional Attached Viscous Flow: Basic Principles and Theoretical Foundations*. Springer Berlin, Heidelberg, 2014.
- [15] L. M. Mack. Transition prediction and linear stability theory. In *Special Course on Stability and Transition of Laminar Flows*, AGARD R-709, 1984.
- [16] Song Fu and Liang Wang. RANS modeling of high-speed aerodynamic flow transition with consideration of stability theory. *Progress in Aerospace Sciences*, 58:36–59, 2013.
- [17] Robin B. Langtry and Florian R. Menter. Correlation-based transition modeling for unstructured parallelized computational fluid dynamics codes. *AIAA Journal*, 47(12):2894–2906, 2009.
- [18] Sourabh S. Diwan and O. N. Ramesh. On the origin of the inflectional instability of a laminar separation bubble. *Journal of Fluid Mechanics*, 629:263–298, 2009.
- [19] A. Saxena. The laminar separation bubble. Technical report, University of Maryland, 2009.
- [20] J.-C. Juillen and D. Arnal. Etude de la transition et de la contamination de bord d’attaque sur ailes en eche. In *Fluid Dynamics of Three-Dimensional Turbulent Shear Flows and Transition*, 1989.
- [21] Philip Hall and Mujeeb R. Malik. On the instability of a three-dimensional attachment-line boundary layer: weakly nonlinear theory and a numerical approach. *Journal of Fluid Mechanics*, 163:257—282, 1986.
- [22] P. Spalart. Direct numerical study of leading-edge contamination. In *Fluid Dynamics of Three-Dimensional Turbulent Shear Flows and Transition*, 1989.
- [23] R. Narasimha. The three archetypes of relaminarisation. In *6th Canadian Conference of Applied Mechanics*, pages 503—518, 1977.

- [24] R. Narasimha and K. R. Sreenivasan. Relaminarization in highly accelerated turbulent boundary layers. *Journal of Fluid Mechanics*, 61(3):417–447, 1973.
- [25] L Kleiser and T A Zang. Numerical simulation of transition in wall-bounded shear flows. *Annual Review of Fluid Mechanics*, 23(1):495–537, 1991.
- [26] Manoj Thapa, Ping Lu, and Chaoqun Liu. DNS study on the origin of the flow randomization in late boundary layer transition. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, AIAA 2013-997, January 2013.
- [27] J. Smagorinsky. General circulation experiments with the primitive equations, Part 1, basic experiments. *Monthly Weather Review*, 91(3):99–164, 1963.
- [28] D. K. Lilly. On the application of the eddy viscosity concept in the inertial sub-range of turbulence. Technical report, National Center for Atmospheric Research, 1966.
- [29] Ugo Piomelli, Thomas A. Zang, Charles G. Speziale, and M. Yousuff Hussaini. On the large-eddy simulation of transitional wall-bounded flows. *Physics of Fluids A: Fluid Dynamics*, 2(2):257–265, 1990.
- [30] J. Lim, M. Kim, S. Kim, S. Jee, and D. Park. Cost-effective and high-fidelity method for turbulent transition in compressible boundary layer. *Aerospace Science and Technology*, 108:106367, 2021.
- [31] Solkeun Jee, Jongwook Joo, and Ray-Sing Lin. Toward Cost-Effective Boundary Layer Transition Computations With Large-Eddy Simulation. *Journal of Fluids Engineering*, 140(11), 2018. 111201.
- [32] M. Kim, J. Lim, S. Kim, S. Jee, J. Park, and D. Park. Large-eddy simulation with parabolized stability equations for turbulent transition using OpenFOAM. *Computers & Fluids*, 189:108–117, 2019.
- [33] M. Kim, J. Lim, S. Kim, S. Jee, and D. Park. Assessment of the wall-adapting local eddy-viscosity model in transitional boundary layer. *Computer Methods in Applied Mechanics and Engineering*, 371:113287, 2020.
- [34] Ali Uzun, Gregory A. Blaisdell, and Anastasios S. Lyrintzis. Sensitivity to the Smagorinsky constant in turbulent jet simulations. *AIAA Journal*, 41(10):2077–2079, 2003.
- [35] S. Lardeau, M. Leschziner, and T. Zaki. Large eddy simulation of transitional separated flow over a flat plate and a compressor blade. *Flow, Turbulence and Combustion*, 88: 19–44, 2012.
- [36] Dieter Forster, David R. Nelson, and Michael J. Stephen. Large-distance and long-time properties of a randomly stirred fluid. *Physical Review A*, 16:732–749, 1977.

- [37] J. D. Fournier and U. Frisch. Remarks on the renormalization group in statistical fluid dynamics. *Phys. Rev. A*, 28:1000–1002, 1983.
- [38] Petri Majander and Timo Siikonen. Evaluation of Smagorinsky-based subgrid-scale models in a finite-volume computation. *International Journal for Numerical Methods in Fluids*, 40(6):735–774, 2002.
- [39] F. F. Grinstein, L. G. Margolin, and W. J. Rider. *Implicit Large Eddy Simulation: Computing Turbulent Fluid Dynamics*. Cambridge University Press, 2007.
- [40] Philippe R Spalart. Hybrid RANS–LES methods. In *Advanced Approaches in Turbulence*, pages 133–159. Elsevier, 2021.
- [41] W. M. Orr. The stability or instability of the steady motions of a perfect liquid and of a viscous liquid, Part I: A perfect liquid. In *Proceedings of the Royal Irish Academy, Section A: Mathematical and Physical Sciences*, Vol. 27, pages 9–68, 1907.
- [42] W. M. Orr. The stability or instability of the steady motions of a perfect liquid and of a viscous liquid, Part II: A perfect liquid. In *Proceedings of the Royal Irish Academy, Section A: Mathematical and Physical Sciences*, Vol. 27, pages 69–138, 1907.
- [43] A. Sommerfeld. Ein beitrag zur hydrodynamische erklärung der turbulenten flüssigkeitsbewegungen. In *Proceedings of the 4th International Congress of Mathematicians*, Vol. III, Rome, Italy, pages 116–124, 1909.
- [44] A.M.O. Smith and N. Gamberoni. Transition, pressure gradient and stability theory. *Douglas Aircraft Co., Report ES 26388*, 1956.
- [45] J. van Ingen. A suggested semi-empirical method for the calculation of the boundary layer transition region. *Technische Hogeschool Delft, Vliegtuigbouwkunde, Rapport VTH-74*, 1956.
- [46] A. M. Savill. By-pass transition using conventional closures. In B. E. Launder and N. D. Sandham, editors, *Closure strategies for turbulent and transitional flows*, pages 464–492. Cambridge University Press, Cambridge, 2002.
- [47] D. C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries, La Canada, CA, 1993.
- [48] A. M. Savill. Some recent progress in the turbulence modelling of by-pass transition. In So RMC, C. G. Speziale, and B. E. Launder, editors, *Near-wall turbulent flows*, page 829. Elsevier, 1993.
- [49] A. M. Savill. One-point closures applied to transition. In M. Hallböck, Henningson D.S., A. V. Johansson, and P. H. Alfredsson, editors, *Turbulence and transition. Lecture notes from the ERCOFTAC/IUTAM Summer school*, pages 233–268. Kluwer Academic Publishers, Stockholm, 1996.

- [50] S. Lardeau, M. A. Leschziner, and N. Li. Modelling bypass transition with low-Reynolds number nonlinear eddy-viscosity closure. *Flow, Turbulence and Combustion*, 73(1):49–76, 2004.
- [51] K. J. A. Westin and R. A. W. M. Henkes. Application of turbulence models to bypass transition. *Journal of Fluids Engineering*, 119(4):859–66, 1997.
- [52] I. Hadžić and K. Hanjalić. Separation-induced transition to turbulence: second-moment closure modelling. *Flow, Turbulence and Combustion*, 63(1):153–173, 2000.
- [53] C. L. Rumsey, B.A. Pettersson-Reif, and T. B. Gatski. Arbitrary steady-state solutions with the $k-\varepsilon$ model. *AIAA Journal*, 44:1586–1592, 2006.
- [54] 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.
- [55] R. Michel. Etude de la transition et calcul de la trainee des profiles d’aile en incompressible. In *ONERA Publ. 58*, 1951.
- [56] P. S. Granville. The calculation of the viscous drag of bodies of revolution. Technical Report David W. Taylor Model Basin Report 849, Navy Department, 1953.
- [57] A. Wazzan, C. Gazley, and A. Smith. Tollmien-schlichting waves and transition: Heated and adiabatic wedge flows with application to bodies of revolution. *Progress in Aerospace Sciences*, 18:351–392, 1979.
- [58] 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.
- [59] D. Keith Walters and James H. Leylek. A new model for boundary layer transition using a single-point rans approach. *Journal of Turbomachinery*, 126(1):193–202, 2004.
- [60] R. Langtry. Extending the $\gamma-Re_\theta$ correlation based transition model for crossflow effects (invited). In *45th AIAA Fluid Dynamics Conference*, 2015.
- [61] Rohit Jain. Computational fluid dynamics transition models validation for rotors in unsteady flow conditions. *Journal of Aircraft*, 59(4):875–895, 2022.
- [62] F.R. Menter, P.E. Smirnov, T. Liu, and R. Avancha. A one-equation local correlation-based transition model. *Flow, Turbulence and Combustion*, 95:583–619, 2015.
- [63] Shivaji Medida and James Baeder. Application of the correlation-based $\gamma-Re_\theta$ transition model to the Spalart-Allmaras turbulence model. In *20th AIAA Computational Fluid Dynamics Conference*, 2011.

- [64] Robert H. Nichols. Addition of a local correlation-based boundary layer transition model to the create-av kestrel unstructured flow solver. In *AIAA Scitech 2019 Forum*, 2019.
- [65] J. G. Coder and M. D. Maughmer. Computational fluid dynamics compatible transition modeling using an amplification factor transport equation. *AIAA Journal*, 52(11):2506–2512, 2014. doi: <https://doi.org/10.2514/1.J052905>.
- [66] James G. Coder. Further development of the amplification factor transport transition model for aerodynamic flows. In *AIAA Scitech 2019 Forum*, 2019.
- [67] M. Drela and M. B. Giles. Viscous-inviscid analysis of transonic and low Reynolds number airfoils. *AIAA Journal*, 25(10):1347–1355, 1987. doi: <https://doi.org/10.2514/3.9789>.
- [68] Steven L. Brunton, Bernd R. Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52(1):477–508, 2020. doi: [10.1146/annurev-fluid-010719-060214](https://doi.org/10.1146/annurev-fluid-010719-060214).
- [69] Andreas Krumbein, Normann Krimmelbein, and Geza Schrauf. Automatic transition prediction in hybrid flow solver, Part 1: Methodology and sensitivities. *Journal of Aircraft*, 46(4):1176–1190, 2009. doi: [10.2514/1.39736](https://doi.org/10.2514/1.39736).
- [70] Andreas Krumbein, Normann Krimmelbein, and Geza Schrauf. Automatic transition prediction in hybrid flow solver, Part 2: Practical application. *Journal of Aircraft*, 46(4):1191–1199, 2009. doi: [10.2514/1.39738](https://doi.org/10.2514/1.39738).
- [71] Jean Perraud, Daniel Arnal, Gregoire Casalis, Jean-Pierre Archambaud, and Raffaele Donelli. Automatic transition predictions using simplified methods. *AIAA Journal*, 47(11):2676–2684, 2009. doi: [10.2514/1.42990](https://doi.org/10.2514/1.42990).
- [72] Balaji Shankar Venkatachari, Jared Carnes, Chau-Lyan Chang, and Meelan M. Choudhari. *Boundary-Layer Transition Prediction Through Loose Coupling of OVERFLOW and LASTRAC*. doi: [10.2514/6.2022-3682](https://doi.org/10.2514/6.2022-3682).
- [73] Nathaniel J. Hildebrand, Chau-Lyan Chang, Meelan M. Choudhari, Fei Li, Eric J. Nielsen, Balaji Shankar Venkatachari, and Pedro Paredes. *Coupling of the FUN3D Unstructured Flow Solver and the LASTRAC Stability Code to Model Transition*. doi: [10.2514/6.2022-1952](https://doi.org/10.2514/6.2022-1952).
- [74] Muchen Yang and Zhixiang 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. doi: [10.1063/5.0008493](https://doi.org/10.1063/5.0008493).
- [75] Harshal D. Akolekar, Fabian Waschowski, Yaomin Zhao, Roberto Pacciani, and Richard D. Sandberg. Transition modeling for low pressure turbines using computational fluid dynamics driven machine learning. *Energies*, 14(15), 2021. doi: [10.3390/en14154680](https://doi.org/10.3390/en14154680).

- [76] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- [77] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3:218–229, 2021.
- [78] Patricio Clark Di Leoni, Lu Lu, Charles Meneveau, George Em Karniadakis, and Tamer A. Zaki. Neural operator prediction of linear instability waves in high-speed boundary layers. *Journal of Computational Physics*, 474:111793, 2023. ISSN 0021-9991. doi: <https://doi.org/10.1016/j.jcp.2022.111793>.
- [79] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017.
- [80] Ali Kashefi, Davis Rempe, and Leonidas J. Guibas. A point-cloud deep learning framework for prediction of fluid flow fields on irregular geometries. *Physics of Fluids*, 33(2): 027104, 2021.
- [81] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. In *Advances in Neural Information Processing Systems*, volume 33, pages 6755–6766, 2020.
- [82] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020.
- [83] Xu-Hui Zhou, Jiequn Han, and Heng Xiao. Frame-independent vector-cloud neural network for nonlocal constitutive modeling on arbitrary grids. *Computer Methods in Applied Mechanics and Engineering*, 388:114211, 2022.
- [84] Pedro Paredes, Balaji Venkatachari, Meelan M. Choudhari, Fei Li, Chau-Lyan Chang, Muhammad I. Zafar, and Heng Xiao. Toward a practical method for hypersonic transition prediction based on stability correlations. *AIAA Journal*, 58(10):4475–4484, 2020. doi: 10.2514/1.J059407.
- [85] Jiequn Han, Xu-Hui Zhou, and Heng Xiao. An equivariant neural operator for developing nonlocal tensorial constitutive models. 2022.
- [86] Jinlong Wu, Heng Xiao, Rui Sun, and Qiqi Wang. Reynolds-averaged navier–stokes equations with explicit data-driven reynolds stress closure can be ill-conditioned. *Journal of Fluid Mechanics*, 869:553–586, 2019. doi: 10.1017/jfm.2019.205.

- [87] Carlos A. Michel'en Strofer and Heng Xiao. End-to-end differentiable learning of turbulence models from indirect observations. *Theoretical and Applied Mechanics Letters*, 11:100280, 2021.
- [88] Carlos A. Michel'en Strofer, Xinlei Zhang, and Heng Xiao. Ensemble gradient for learning turbulence models from indirect observations. *Commun. Comput. Phys.*, 30: 1269–1289, 2021.
- [89] Xin-Lei Zhang, Heng Xiao, Xiaodong Luo, and Guowei He. Ensemble kalman method for learning turbulence models from indirect observation data. *Journal of Fluid Mechanics*, 949:A26, 2022. doi: 10.1017/jfm.2022.744.

Chapter 2

Convolutional neural network for transition modeling based on linear stability theory

Convolutional neural network for transition modeling based on linear stability theory

Muhammad I. Zafar  and Heng Xiao ^{*}

Kevin T. Crofton Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, Virginia 24061, USA

Meelan M. Choudhari , Fei Li, and Chau-Lyan Chang 

NASA Langley Research Center, Hampton, Virginia 23681, USA

Pedro Paredes  and Balaji Venkatachari 

National Institute of Aerospace, Hampton, Virginia 23666, USA



(Received 20 April 2020; accepted 4 November 2020; published 23 November 2020)

Transition prediction is an important aspect of aerodynamic design because of its impact on skin friction and potential coupling with flow separation characteristics. Traditionally, the modeling of transition has relied on correlation-based empirical formulas based on integral quantities such as the shape factor of the boundary layer. However, in many applications of computational fluid dynamics, the shape factor is not straightforwardly available or not well-defined. We propose using the complete velocity profile along with other quantities (e.g., frequency, Reynolds number) to predict the perturbation amplification factor. While this can be achieved with regression models based on a classical fully connected neural network, such a model can be computationally more demanding. We propose a convolutional neural network inspired by the underlying physics as described by the stability equations. Specifically, convolutional layers are first used to extract integral quantities from the velocity profiles, and then fully connected layers are used to map the extracted integral quantities, along with frequency and Reynolds number, to the output (amplification ratio). Numerical tests on classical boundary layers clearly demonstrate the merits of the proposed method. More importantly, we demonstrate that, for Tollmien-Schlichting instabilities in two-dimensional, low-speed boundary layers, the proposed network encodes information in the boundary-layer profiles into an integral quantity that is strongly correlated to a well-known, physically defined parameter—the shape factor.

DOI: [10.1103/PhysRevFluids.5.113903](https://doi.org/10.1103/PhysRevFluids.5.113903)

I. INTRODUCTION

Laminar-turbulent transition of boundary-layer flows can have a strong impact on the performance of flight vehicles because of their influence on surface skin friction and aerodynamic heating. Therefore, transition prediction is a key issue for the design of next-generation aerospace configurations. Indeed, according to the CFD Vision 2030 Study [1], the most critical area in computational fluid dynamics (CFD) simulation capability that will remain a pacing item for the foreseeable future is the ability to adequately predict viscous flows involving transition-to-turbulence and/or flow separation.

^{*} Author to whom all correspondence should be addressed: hengxiao@vt.edu

Under the benign disturbance environment in flight, boundary-layer transition is often initiated by the amplification of linearly unstable eigenmodes of the laminar boundary layer. For two-dimensional (2D) and weakly 3D boundary layers developing over a nominally smooth surface, the dominant instability mechanisms correspond to streamwise propagating Tollmien-Schlichting (TS) waves at subsonic speeds, oblique first-mode disturbances at supersonic edge Mach numbers, and again planar waves of the second mode, i.e., Mach mode type in hypersonic flows. Additionally, centrifugal instabilities in the form of Görtler vortices are known to influence the transition process over surfaces with significant regions of concave longitudinal curvature. Finally, attachment line and crossflow instabilities come into play when the flow becomes three-dimensional.

The transition process begins with the generation of instability waves via the interaction (i.e., receptivity) of the laminar boundary layer to its disturbance environment. However, the onset of turbulence ensues only after the instability waves have gained sufficiently large amplitudes to undergo a sequence of nonlinear processes that culminates with the breakdown to turbulence. Because the nonlinear breakdown tends to be relatively rapid, the slow amplification of the linear instability waves accounts for a majority of the laminar flow region preceding the onset of transition. As a result, the linear amplification ratio, e^N , of the most amplified instability mode can often be used to predict the experimental trends in the transition location. The linear amplification ratio is usually computed by using the classical stability theory based on the assumption of a quasiparallel boundary layer flow. Reviews of the linear stability theory for fluid flows may be found in Refs. [2–5]. The connection between the stability theory and laminar-turbulent transition in boundary-layer flows has been discussed in Refs. [6–8]. Prior work [9,10] has shown that the N -factor values between 9 and 11 correlate with the transition locations measured in a broad class of boundary-layer flows.

Direct computations of boundary-layer stability place rather stringent demands on the accuracy of mean flow calculations, much more so in comparison with that required for the prediction of aerodynamic metrics such as the skin friction drag. In addition, the solution to the eigenvalue problem associated with the discretized version of the linear stability equations incurs a significant computational cost. Furthermore, due to the complex nature of the eigenvalue spectra and their sensitivity to both input parameters and numerical discretization, stability computations are difficult to automate, and they also require significant user expertise in the details of the hydrodynamic stability theory. Consequently, the task of transition prediction based on the N -factor methods has been a specialist's domain, and is often performed as a post-processing step that follows the computation of the laminar boundary layer over the flow configuration of interest. Implicit in this post-processing approach is the assumption of a weak coupling between the transition location and the basic state computation.

The weak-coupling assumption may be justified for fully attached boundary-layer flows such as aircraft wings at the cruise condition. However, a number of technological applications, such as high-lift systems [11], rotorcraft [12], and other configurations involving flow separation, entail a strong viscous-inviscid interaction, requiring an iterative prediction approach that reflects the stronger coupling between transition and the overall flow field. To that end, the CFD Vision 2030 Study has called for transition prediction methods that can be fully integrated into the overall process of aerodynamic prediction. In the past, several attempts have been made to simplify the application of the N -factor methods in the engineering environment, ranging from analytical but potentially complex data fits [9,13,14] to numerical, table-lookup procedures based on a prior database of stability results [15–22]. In recent work, such empirical fits have also been incorporated into CFD integrated transition models based on auxiliary transport equations, such as the amplification factor transport model [23].

In the above-mentioned database-query techniques, a response surface model is developed in terms of a small number of scalar input parameters representing the combination of the global flow parameters, selected measures of boundary-layer profiles, and the relevant disturbance characteristics such as frequency and wave-number parameters. Almost universally, one or more shape factors of the boundary-layer profiles have been used to encapsulate the complex dependence of the disturbance amplification rates on the underlying mean flow. This tends to limit the expressive

power of the model as discussed by Crouch *et al.* [24]. Secondly, while such shape factors can be easily evaluated when the mean flow computation is based on the classical boundary-layer theory, it is not easy to compute them in a consistent and accurate manner when Navier-Stokes codes are used for the basic state computation, especially for unstructured grid solutions. Such situations arise rather commonly in high-speed applications, such as the flow past blunt nosed bodies, where the inviscid flow beyond the edge of the boundary layer includes nonzero vorticity as a result of vorticity generation at the curved shock. The database methods can perform rather poorly in these cases, as demonstrated by Paredes *et al.* [25].

Stability predictions based on artificial neural networks [24,26,27] allow additional features of the boundary-layer profiles to be taken into account without sacrificing the computational efficiency and robustness of the conventional methods based on a previously computed database of amplification characteristics. The neural network methods can also be easily generalized to higher-dimensional input features. This allows the multiparameter dependence of the stability characteristics to be accounted for, whereas the conventional methods for database query do not scale very well as the number of independent parameters becomes significantly large. Neural-network-based stability predictions for free-shear layer flows were first presented by Fuller *et al.* [26]. However, a significant advance related to transition prediction was made by Crouch *et al.* [24], who found that the expressivity of the model could be improved by augmenting the set of scalar variables used in conventional database methods via the slopes of the appropriately normalized velocity profiles at six equidistant points across the boundary layer. The details of the neural network architecture used in that paper are somewhat limited; however, a feed-forward network based on fully connected hidden layers was used to approximate the maximum amplification rate among all unstable modes at any given station. This maximum amplification rate was integrated along the airfoil surface to evaluate the N -factors.

The selection of a smaller number of input features by Crouch *et al.* [24] was somewhat arbitrary and is unlikely to lend itself to other instability mechanisms without further modifications. In the present work, we present an alternate approach based on convolutional neural networks that can automatically learn a reduced-order representation of the boundary-layer profiles in terms of a specified number of most significant features that can optimally predict the targeted linear stability characteristics across the training space. As such, the proposed architecture can be easily adapted to predict the amplification characteristics of a broad range of very different instability mechanisms.

The objective of this paper is to present a preliminary proof of concept to establish the potential of convolutional neural networks (CNNs) to distill the latent features of the boundary layer profiles. To that end, it is sufficient to consider the simplest case of TS instability waves in two-dimensional, incompressible boundary layers. Because the architecture of the CNN is not related to the specific physics of the TS instability mechanism or to the flow geometry, the proposed neural network can be easily generalized to other classes of instability waves.

The rest of the paper is organized as follows. An overview of the methodology is presented in Sec. II, which includes a summary of modal stability analysis for two-dimensional, incompressible flows, followed by a description of the proposed architecture of a hybrid convolutional/fully connected neural network. Section III presents the results based on the training and validation of the proposed network architecture, including an assessment of its generalization capability for real-world applications. In particular, we highlight the capability of the proposed network for encoding boundary-layer profiles into integral parameters in an automatic, data-driven manner. Section IV concludes the paper.

II. METHODOLOGY

Using the e^N method, the onset of laminar-turbulent transition is predicted to occur where the logarithmic amplification ratio N of the most amplified instability mode reaches an empirically defined critical value, denoted herein as $N = N_{tr}$. As described in Sec. II A, the logarithmic amplification ratio N can be computed by solving the governing equations based on linear stability theory.

Thus, N is a direct function of the laminar boundary-layer profiles (e.g., U , dU/dy , and d^2U/dy^2), the flow Reynolds number Re_θ based on local momentum thickness of the boundary layer, and disturbance parameters such as the frequency of the instability wave ω and, for 3D disturbances, the spanwise wave number. The objective herein is to develop a surrogate transition model based on a neural network that would incorporate the physics of the transition phenomenon and predict the transition onset location without requiring the direct computations using linear stability theory. Since this data-driven surrogate transition model is being developed based on linear stability theory, any limitations of the quasiparallel theory are inherited by the present implementation of the neural-network-based surrogate model. However, since the methodology developed here has no intrinsic dependence on the quasiparallel theory, the basic framework can be extended to use any advanced simulation method, such as parabolized stability equations (PSEs), to generate the data and thus to overcome of the limitations of the classical theory. This would be a fruitful avenue for future work.

We propose a hybrid neural network (consisting of convolutional and fully connected layers) that allows the relevant flow information (boundary-layer profiles and scalar disturbance characteristics) to be processed in a physically informed manner. The resulting regression models predict the local instability amplification rates, σ , for the relevant frequencies of the instability waves as they propagate through the laminar boundary layer. The N -factor curves can be computed by integrating the predicted σ values to allow the transition onset location to be predicted on the basis of the empirically defined critical value, N_{tr} . Moreover, as a baseline to compare the performance of the proposed neural network, we also consider the fully connected neural network architecture based on the idea proposed earlier by Crouch *et al.* [24]. In the first part of this section, we discuss the basis of the e^N method and the linear stability theory. The transition models based on the proposed neural network and the fully connected neural network are outlined in the second part. The database used to train these neural-network-based transition models is discussed in the last part of this section.

A. The e^N method

For simplicity, we outline the transition prediction procedure in the context of an incompressible, fully attached flow over a two-dimensional airfoil. An orthogonal, body-fitted, curvilinear coordinate system (s, n) is introduced such that s denotes the distance from the stagnation point, measured along the airfoil contour on either the suction or the pressure side of the airfoil, and n represents the outward surface normal. Consistent with the boundary-layer character of the basic state, both coordinates are normalized by a length scale comparable to the local thickness of the boundary layer, which is taken to be the local momentum thickness $\theta(s)$ without any loss of generality. The two-dimensional boundary-layer flow over the airfoil is represented by the velocity field (U, V) , where the velocities are normalized by a local velocity scale $U_e(s)$, taken to be the flow speed at the edge of the mean boundary layer. The Reynolds-number parameter based on the free-stream speed and the momentum thickness is denoted by Re_θ .

We consider small-amplitude, time-harmonic, spatially evolving perturbations to the mean boundary-layer flow of the form

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} U(n) \\ V(n) \end{bmatrix} \exp(i[\varphi(s) - \omega t]), \quad (1)$$

where $i = \sqrt{-1}$ is the imaginary unit, $\omega \equiv 2\pi f$ denotes the real valued disturbance frequency, f is the frequency parameter in Hz, t is the appropriately normalized time, and $d\varphi/ds = \alpha$ denotes the complex streamwise wave number. Substituting the above normal mode ansatz into the linearized Navier-Stokes equations and neglecting the weak nonparallel effects associated with the $O(Rde^{-1})$ velocity component \mathbf{V} and the slow streamwise evolution $\partial U/\partial s$ of the tangential velocity field, one obtains the quasiparallel form of the disturbance equations that must be solved with homogeneous boundary conditions for U and V at the surface ($n = 0$) as well as in the free stream ($n \rightarrow \infty$). For an incompressible flow, the quasiparallel disturbance equations can be combined into a single

equation for the wall-normal velocity perturbation, yielding an eigenvalue problem based on the well known Orr-Sommerfeld (OS) equation [28]:

$$(\alpha U - \omega)(V'' + \alpha^2 V) - \alpha U'' V = (V'''' - 2\alpha^2 V'' + \alpha^4)/(i \text{Re}_\theta), \quad (2)$$

along with homogeneous Dirichlet boundary conditions:

$$V(0) = V'(0) = 0 \quad \text{and} \quad V(\infty) = V'(\infty) = 0, \quad (3)$$

where a prime denotes the derivative with respect to the wall-normal coordinate n . The solution to the eigenvalue problem [Eqs. (2) and (3)] at a given station s determines the local value of the complex streamwise wave number α as a function of the frequency parameter ω . The local, streamwise amplification rate of a disturbance at frequency ω corresponds to $\sigma = -\text{Im}(\alpha(\omega, s))$, where $\text{Im}(\cdot)$ denotes the imaginary part of a complex quantity. Hence, the logarithmic amplification of the disturbance amplitude with respect to the neutral station, where the disturbance first begins to amplify, is given by

$$N(\omega, s) = \int_{s_0}^s \sigma(\omega, \tilde{s}) d\tilde{s}, \quad (4)$$

where the subscript “0” denotes the neutral station. The e^N method postulates that transition is likely to occur when the envelope N -factor, $N_e(s) = \sup(N(\omega, s))$, reaches the critical value of $N_e(s) = N_{tr}$. Here “sup” denotes the maximum over the frequency range of all unstable disturbances. Values of $N_{tr} = 9$ –11 have been found to correlate with the onset of transition in a number of subsonic and supersonic flows [9,10].

B. Convolutional neural network

A neural network is a sequence of composite functions representing the mapping from an input vector \mathbf{q} to output vector \mathbf{y} . Each member of the sequence is parametrized by the weight matrix \mathbf{W} and bias vector \mathbf{b} , which can both be learned iteratively by using the available training data. This sequence of composite functions is arranged in the form of layers that consist of several neurons in general. For example, a neural network with one intermediate (hidden) layer \mathbf{h} between the input layer (\mathbf{q}) and output layer (\mathbf{y}) may be represented by the following composite functional mapping:

$$\begin{aligned} \mathbf{y} &= \mathbf{W}^{(2)} \mathbf{h} + \mathbf{b}^{(2)} \quad \text{with} \quad \mathbf{h} = f[\mathbf{W}^{(1)} \mathbf{q} + \mathbf{b}^{(1)}], \text{ or equivalently} \\ \mathbf{y} &= \mathbf{W}^{(2)} (f[\mathbf{W}^{(1)} \mathbf{q} + \mathbf{b}^{(1)}]) + \mathbf{b}^{(2)}, \end{aligned} \quad (5)$$

where f is an activation function, and $\mathbf{W}^{(i)}$ and $\mathbf{b}^{(i)}$ represent the weight matrix and biases vector for the i th layer, respectively. Activation functions introduce the nonlinearity in the composite functions that enables them to represent arbitrarily complex functional relationships. Several different activation functions have been proposed for this purpose, such as the sigmoid function $f(x) = 1/(1 + e^{-x})$ or the rectified linear unit (ReLU) $f(x) = \max(0, x)$. The training of the neural network consists of successive adjustments of the weights and the biases in order to minimize the squared error between the predicted and truth values of the output feature, i.e., the local amplification rate σ in the present application. Neural networks with at least one hidden layer are *universal approximators*, i.e., they can represent any continuous function on a compact domain to arbitrary accuracy, given a sufficient number of neurons in the hidden layer [29].

In a fully connected neural network, each neuron in a given layer is connected to every neuron in the adjacent layer, yielding a generic connection pattern that makes no assumptions about the input features in the data. The schematic of a fully connected neural network with multiple input features is displayed in Fig. 1. As mentioned in the Introduction, the existing models starting with Ref. [13] have used analytical curve fits or other database query methods to predict the local amplification rate of the instability wave as a function of the disturbance frequency ω , local Reynolds number Re_θ , and a scalar parameter such as the shape factor H . A fully connected neural network such

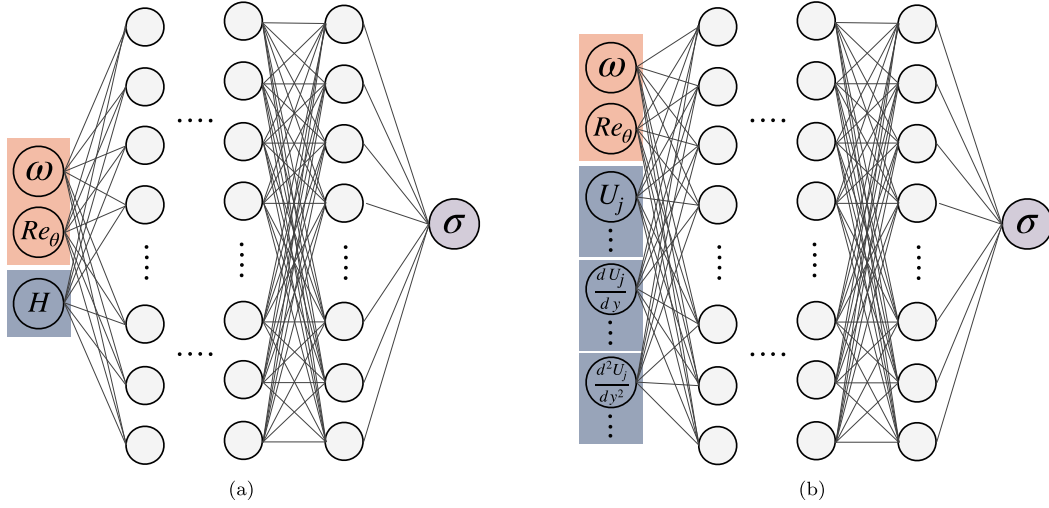


FIG. 1. Fully connected neural networks, (a) with input features including a scalar boundary-layer parameter, shape factor H , and (b) with boundary-layer profiles as direct input features. Other scalar parameters in both architectures represent disturbance characteristics of instability waves, i.e., frequency (ω) and Reynolds number (Re_θ).

as that shown in Fig. 1(a) provides a suitable architecture to achieve a similar functionality by using a neural network in place of the database interpolation or curve fitting. However, we note that the scalar boundary-layer parameter H does not appear directly in the governing equations for linear stability theory, and hence it is only indirectly related to the amplification characteristics of the instability modes. Moreover, the shape factor cannot be determined in a straightforward and/or accurate manner for several boundary layers, such as high-speed flows over blunt nose configurations [25]. Therefore, it is desirable to introduce the boundary-layer profiles directly into the predictive model, which is *physically more consistent* with the underlying linear stability equation. Crouch *et al.* [24] presented a model of this type by including a coarse representation of the mean velocity profiles as part of the input for the fully connected neural network. Whereas they used the velocity and its first derivative at just six equidistant points, Fig. 1(b) presents a model architecture based on a well-resolved representation of the boundary-layer profiles (i.e., velocity profile U and its derivatives dU/dy and d^2U/dy^2) as input features for the fully connected neural network. However, by using a large number of parameters to characterize the boundary-layer profiles, a model architecture of this type risks a potential misrepresentation of the dependence of instability growth rates on the two remaining physical parameters, namely ω and Re_θ . The balance involving the number of input parameters becomes more lopsided for high-speed flows, since the boundary-layer profiles for temperature or density are also required for a reliable prediction of the amplification rate σ .

A convolutional neural network (CNN) [30] is composed of a number of convolutional and pooling layers, which enable it to automatically extract the latent features of the input data, which are considered to be an ordered data structure. Specifically, the boundary-layer profiles can now be considered to be an ordered array, in contrast to the fully connected neural network that makes no assumption about the ordering of basic state quantities across the input profile. Furthermore, the CNN exploits two special attributes to learn efficiently with a smaller number of model parameters. First, each neuron in the convolutional layers has only local connections to the neurons in the previous layer, allowing it to develop a correlation with the neighboring neurons. Second, the convolutional kernel has translational invariance in terms of model parameters, which leads to a drastic reduction in the number of network parameters. These attributes allow the CNN to achieve

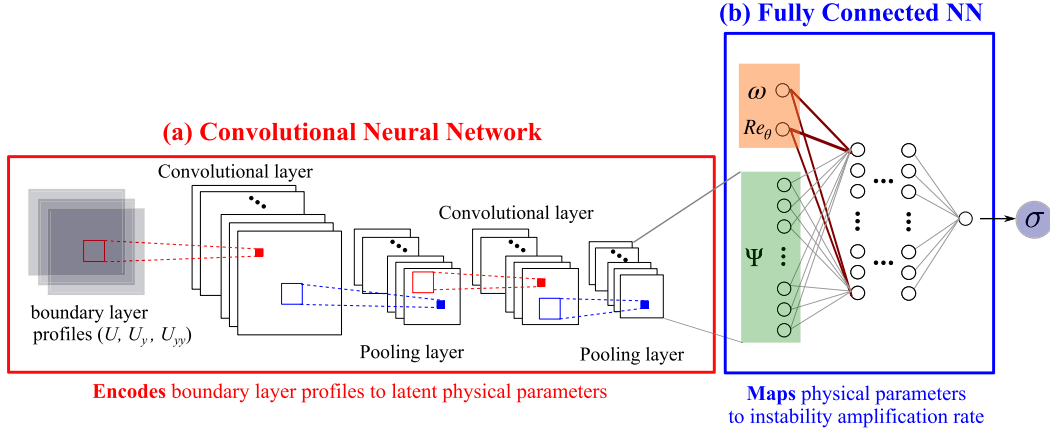


FIG. 2. Proposed hybrid convolutional neural network architecture including (a) a regular CNN, which encodes the boundary-layer profiles to a set of latent physical parameters Ψ , and (b) a fully connected neural network, which maps the CNN-extracted features Ψ along with other physical parameters (frequency of the instability wave ω and Reynolds number Re_θ) to the output (instability amplification rate σ).

a comparable predictive accuracy much more efficiently in terms of the training cost and/or the amount of data required for training [31].

A schematic of the proposed hybrid neural network is presented in Fig. 2. In this network architecture, the CNN first [Fig. 2(a)] maps the boundary-layer profiles to a specified number of latent features in a physically consistent manner while accounting for the spatial proximity of neighboring points across the boundary-layer profiles. The encoding of the boundary-layer profiles in the form of these latent features is denoted by the vector Ψ . Following a preliminary assessment, the number of latent features in vector Ψ was empirically chosen to be 8. However, the results are not significantly sensitive to this parameter. Next, the revised set of input parameters comprising the vector Ψ and the remaining physical scalar parameters (ω and Re_θ) is nonlinearly mapped through a fully connected neural network [Fig. 2(b)] to yield the local instability amplification rate σ as the final output. Observe that the dependence of σ on the physical parameters ω and Re_θ is introduced into the network architecture in an explicit yet flexible manner. In particular, the relationship between the local instability amplification rate and the parameters ω and Re_θ , along with the boundary-layer profiles, is known from the linear stability theory [Eqs. (2) and (3)], and that quantitative relation can be inferred via the training process.

The hyperparameters of the proposed neural network (Fig. 2) and those of fully connected neural networks (Fig. 1) have been empirically selected to yield an adequate complexity of the neural network model for learning all of the required information, without causing an overfitting of the training data. The list of primary hyperparameters includes the number of convolutional layers, the number of channels in each convolutional layer, the number of fully connected layers, the number of neurons in each fully connected layer, and the learning rate for the training of the model. A summary of the relevant model architectures is given in Table I, wherein each category of architecture has been labeled as A, B, or $C_{1,2,3}$ for future reference. The number of neurons in each fully connected layer and the number of channels in each convolutional layer have also been listed for the respective architectures. The number of input channels for networks C_1 , C_2 , and C_3 can be varied to accommodate the desired number of boundary-layer profiles that are to be used as input to the convolutional layers. Convolutional kernels of size 3×1 have been used to extract the latent features from the boundary-layer profiles defined by 41 equidistant points along the wall-normal direction. In all networks, the ReLU is used as the activation function, and the Adam optimization algorithm [32] has been chosen to minimize the sum squared error during the training process. All of

TABLE I. Details of network architectures along with respective input features. Acronyms used: FC, fully connected; NN, neural network; NL, neurons in FC network layers; CH, number of channels in CNN layers. The first fully connected layer of the C_i ($i = 1, 2, 3$) networks has (2+8) neurons, where the first two neurons correspond to the physical parameters (ω, Re_θ), and the eight additional neurons correspond to a vector of parameters, Ψ , that encodes the information from boundary-layer profiles. The C_i^* ($i = 1, 2, 3$) networks use only one scalar value for representing parameter Ψ , hence the first fully connected layer has (2+1) neurons.

Network	Input features	Architecture type	Architecture	Number of parameters
A	$\omega, \text{Re}_\theta, H$	Fig. 1(a): Fully connected NN	NL:[3,96,96,96,96,96,1]	56 353
B	$\omega, \text{Re}_\theta, U_j, \frac{dU}{dy} _j, \frac{d^2U}{dy^2} _j$	Fig. 1(b): Fully connected NN	NL:[125,96,96,96,96,96,1]	57 261
C_1	$\omega, \text{Re}_\theta, U_j, \frac{dU}{dy} _j, \frac{d^2U}{dy^2} _j$	Fig. 2: Convolutional NN + fully connected NN	CH: [3,6,8,4] + NL: [2+8,96,96,96,96,96,1]	57 337
C_2	$\omega, \text{Re}_\theta, U_j, \frac{dU}{dy} _j$	Fig. 2: Convolutional NN + fully connected NN	CH: [2,4,8,4] + NL: [2+8,96,96,96,96,96,1]	57 257
C_3	$\omega, \text{Re}_\theta, U_j$	Fig. 2: Convolutional NN + fully connected NN	CH: [1,4,8,4] + NL: [2+8,96,96,96,96,96,1]	57 245
$C_i^* \ i = 1, 2, 3$	Same as C_i above	Same as C_i above	Same as C_i above, but with only 2+1 input features for the fully connected network	57 245

the neural network architectures considered herein have been implemented in the machine learning framework PYTORCH.

C. Generation of training data

The training database is obtained by solving the Orr-Sommerfeld (OS) eigenvalue problem [Eq. (3)] for the Falkner-Skan family of self-similar boundary-layer profiles over a wide range of pressure gradient parameter β_H and local Reynolds number $\text{Re}_\theta(s)$. It was generated by using the stability analysis software, LSTRAC, developed at the NASA Langley Research Center [33]. LSTRAC is a well-known software suite that has been extensively validated against existing benchmark data including direct numerical simulations.

The convolutional neural network model maps the complex dependence of the local instability amplification rate on the relevant disturbance characteristics and the mean flow parameters. The training database consists of a tabular listing of stability characteristics for the Falkner-Skan boundary layers, which includes a comprehensive, discrete sampling of the complex-valued local wave number α of the TS instability wave as a function of the real-valued frequency of the wave ω and the mean flow parameters, which include the Hartree pressure gradient parameter β_H , or equivalently the shape factor H of the Falkner-Skan velocity profile, the local Reynolds number Re_θ based on the momentum thickness, and the velocity profile and its first- and second-order derivatives.

The Falkner-Skan group of boundary layers supports a single instability mode that corresponds to the viscous-inviscid interactive TS waves by themselves ($\beta_H \geq 0$) or a combination of instability mechanisms involving the TS waves and the predominantly inviscid Rayleigh instabilities ($\beta_H < 0$). All stability calculations were carried out for a compressible boundary-layer flow with a vanishingly

TABLE II. Input features for neural network models.

Feature	Definition	Expression
q_1	Nondimensional frequency of the instability wave	ω
q_2	Reynolds number based on edge velocity and momentum thickness	Re_θ
q_3	Local value of velocity profile shape factor (derived parameter)	H
q_4	Velocity profile as a function of wall normal coordinate y	$U_j, j = 1, 2, \dots, 41$
q_5	First-order derivative of velocity profile	$\frac{dU}{dy} _j, j = 1, 2, \dots, 41$
q_6	Second-order derivative of velocity profile	$\frac{d^2U}{dy^2} _j, j = 1, 2, \dots, 41$

small Mach number of 10^{-5} , and a stagnation temperature of 311.11 K, along with an adiabatic thermal wall boundary condition and zero transpiration velocity at the surface. All parameters included in the database are nondimensional. Lengths are scaled with respect to local momentum thickness, velocities with respect to the flow speed at the edge of the boundary layer, and temperature with respect to the local edge temperature. The database includes Hartree pressure gradient parameters in the range of $\beta_H \in [-0.1988, 1]$, corresponding to the discrete values given by

$$\beta_H = [-0.1988, -0.19, -0.18, -0.16, -0.14, -0.12, -0.10, -0.075, -0.05, -0.025, 0, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5, 0.6, 0.8, 1.0].$$

Because of a rapid change in the instability characteristics for $\beta_H < 0$, especially as $\beta_H \rightarrow -0.1988$ (i.e., when the boundary layer is on the verge of separation), the sampling in β_H is chosen to be denser at the negative values of β_H . This bias in sampling may result in a bias in the learning of the neural network models toward the lower limit of the β_H values. Stability computations were carried out for Reynolds numbers extending from just below the minimum critical Reynolds number (below which all disturbances are predicted to decay) up to $\text{Re}_s \equiv U_\infty s/\nu = 10^{10}$, where ν denotes the kinematic viscosity of the fluid. The frequency range at each Reynolds number included the entire range of unstable disturbances as well as a modest range of stable disturbances in the vicinity of the neutral stability curve. Because the parameter range covered multiple orders of magnitudes, a logarithmic increment was used along both axes. Due to computational considerations, a quarter of the data points in the database were used, which amounts to approximately 400 000. The results are not influenced by the down-sampling of training data. The input parameters used to train and predict the local instability amplification rate correspond to a suitable subset of the various features listed in Table II. Boundary-layer profiles include the velocity profile along with the first- and second-order derivatives, sampled at 41 equidistant points in order to resolve each profile. The scalar input features ($q_1 : \omega$, $q_2 : \text{Re}_\theta$, and $q_3 : H$) have been scaled and shifted to the range of $[0, 1]$. We note in passing that modified input features based on a logarithmic scale along the Re_θ and ω axes were also considered on the basis of the high Reynolds number asymptotic theory of Tollmien-Schlichting waves [34], but no significant improvement in the testing error was noted.

The present database has been generated as part of the NASA Langley Research Center's effort to use machine learning methods to enable robust, CFD-solver-friendly models for boundary-layer transition. This database will be made available in an electronic form to encourage the development of physics-based transition models that can be integrated with CFD solvers.

III. RESULTS

In this section, we demonstrate the predictive performance of the proposed convolutional neural network (Fig. 2) and compare it with the performance of the fully connected neural networks [Figs. 1(a) and 1(b)]. In the first part of this section, the proposed neural network model is validated using the Falkner-Skan database (presented in Sec. II C) by randomly splitting the data, with 90%

TABLE III. Comparison of the validation error corresponding to neural network models from Table I. Training and validation datasets correspond to a random 90–10 % split of the available database over the entire range of the pressure gradient parameter β_H .

Network	Input features	Validation error
A	$\omega, \text{Re}_\theta, H$	0.58%
C_1	$\omega, \text{Re}_\theta, U_j, \frac{dU}{dy} _j, \frac{d^2U}{dy^2} _j$	0.41%
C_2	$\omega, \text{Re}_\theta, U_j, \frac{dU}{dy} _j$	0.44%
C_3	$\omega, \text{Re}_\theta, U_j$	0.46%

of the data points used for training and the remaining 10% for validation. Both the training and validation datasets contain data from the entire range of the pressure gradient parameter β_H . We then assess the proposed model for interpolation and extrapolation cases by splitting the Falkner-Skan database into training and testing datasets based on the data corresponding to each pressure gradient parameter β_H .

Finally, the capability of the network trained on the Falkner-Skan database (with self-similar boundary layers) to generalize its predictions to realistic flow configurations is evaluated by comparing the predictions of the neural network model with actual stability computations for those configurations. Specifically, we consider two different airfoils with non-self-similar boundary layers for this purpose, namely a symmetric 2D HSK airfoil [35] and an asymmetric NLF-0416 airfoil [36,37]. The Reynolds number parameter based on the free-stream speed and the chord length of the airfoil is chosen to be $\text{Re}_c = 1.23 \times 10^6$ for the HSK airfoil and $\text{Re}_c = 9 \times 10^6$ for the NLF-0416 airfoil. At the selected flow conditions, the boundary-layer instability is dominated by the TS instabilities of interest. In all cases, the following metric corresponding to the percent error based on the Frobenius norm is used for the evaluation of the model throughout this paper:

$$\epsilon_\sigma = 100 \times \frac{\|\sigma_{\text{truth}} - \sigma_{\text{predicted}}\|_F}{\|\sigma_{\text{truth}}\|_F}, \quad (6)$$

where the Frobenius norm is defined as $\|X\|_F = \sqrt{\sum_i |X_i|^2}$. In the second part of this section, we analyze the feature learning and encoding capability of the convolutional neural network and how it makes the proposed model more robust and generalizable to other flow regimes. In the third part of this section, the potential advantages of the proposed model over the previously proposed model architecture [24] are analyzed.

A. Demonstration of predictive performance

The predictive performance of the convolutional neural network (Fig. 2) is first validated by using the Falkner-Skan database and is also compared with the performance of the fully connected network [Fig. 1(a)] with scalar input features. In these cases, the training process utilized a randomly sampled subset of the available dataset, amounting to 90% of the total data points. The validation is conducted by using the remaining 10% of the data that were never seen by the neural network model during the training process. The results presented in Table III show that the proposed model demonstrates slightly improved predictive performance (Network C_1 , 0.41%) as compared to that of the fully connected neural network with scalar input features (Network A, 0.58%). Furthermore, the proposed neural network provides qualitatively similar results when the number of velocity profiles used as input features is varied from the velocity and its two derivatives (network C_1) to the velocity profile alone (network C_3). The data presented in Table III indicate that the inclusion of the velocity derivatives decreases the validation error, but the improvement is rather small. The small reduction in validation error is consistent with the fact that the derivative information is contained within the

TABLE IV. Results for interpolation and extrapolation in the Hartree pressure gradient parameter. The testing dataset is comprised of data at a specified value of β_H , while the remaining data are used as the training dataset.

Cases	Testing dataset	Testing error
Interpolation	$\beta_H = -0.1$	3.97%
	$\beta_H = 0.1$	3.02%
	$\beta_H = 0.5$	4.68%
Extrapolation	$\beta_H = -0.1988$	15.34%
	$\beta_H = 1.0$	28.52%

velocity profile itself, and therefore we believe that the observed improvement is attributed to the well-resolved yet finite sampling of the velocity profile. Since the validation error for the neural network based on the velocity profiles alone (i.e., network C_3) is already small, the margin for improvement is rather limited, and the findings in Table III confirm this expected behavior. Given the small differences in validation error corresponding to the networks C_1 through C_3 , all of the remaining assessments reported in this paper are based on a single set of input features, which is chosen to include the velocity profile U and its derivatives dU/dy and d^2U/dy^2 . Thus, the mapping sought by the neural network may be represented as

$$\left(\omega, \text{Re}_\theta, U_j, \left. \frac{dU}{dy} \right|_j, \left. \frac{d^2U}{dy^2} \right|_j \right) \mapsto \sigma, \quad \text{where } j = 1, 2, \dots, 41.$$

We now evaluate the performance of the proposed neural network for more challenging interpolation and extrapolation cases, where the Falkner-Skan database has been split for testing and training based on the Hartree pressure gradient parameter β_H . In these cases, data corresponding to a selected value of β_H are reserved for testing while the data corresponding to the remaining values of β_H are used for the training process. Interpolation cases were considered by isolating a single value of β_H for evaluating the testing error, and the corresponding results for $\beta_H = -0.1$, 0.1 , and 0.5 are given in Table IV. The proposed neural network model is able to interpolate the remaining database at each of these three values of β_H with a testing error of between 3% and 5%. We also considered two additional cases based on testing corresponding to the two extremes of the β_H range, namely $\beta_H = -0.1988$ and 1.0 , which amount to an extrapolation from the training database. As expected, the corresponding results in Table IV indicate significantly higher testing errors with the extrapolation in comparison with the testing errors for the three interpolation cases discussed above. Since most of the neural networks are intrinsically interpolators and do not perform well for something beyond the distribution of training data, the proposed neural network is not able to extrapolate well. Some of the neural network architectures that impose physical knowledge such that the model satisfies the governing equations (i.e., cost functions based on ordinary differential equations) are capable of extrapolating significantly [38]. In current work, while sound physical knowledge has been used to guide learning of underlying functions, the model is still an interpolator just like most of the other neural networks, random forest models, or other regression models [39–41] in physical modeling. Further, the testing error percentage toward the lower limit ($\beta_H = -0.1988$) is significantly lower (15.3%) than that toward the higher limit of the range of pressure gradient parameters (namely, 28.5% error for $\beta_H = 1.0$). This disparity in testing errors may be due to the significantly denser sampling in β_H toward the lower end of the range, indicating that the overall accuracy of the neural network could perhaps be improved by increasing the database size by including additional data at higher values of β_H .

As the underlying purpose of the proposed neural network model is to predict the transition onset location, we next evaluate the predictive performance of the neural network model C_1 for a symmetric HSK airfoil section and an asymmetric NLF-0416 airfoil section. These two examples

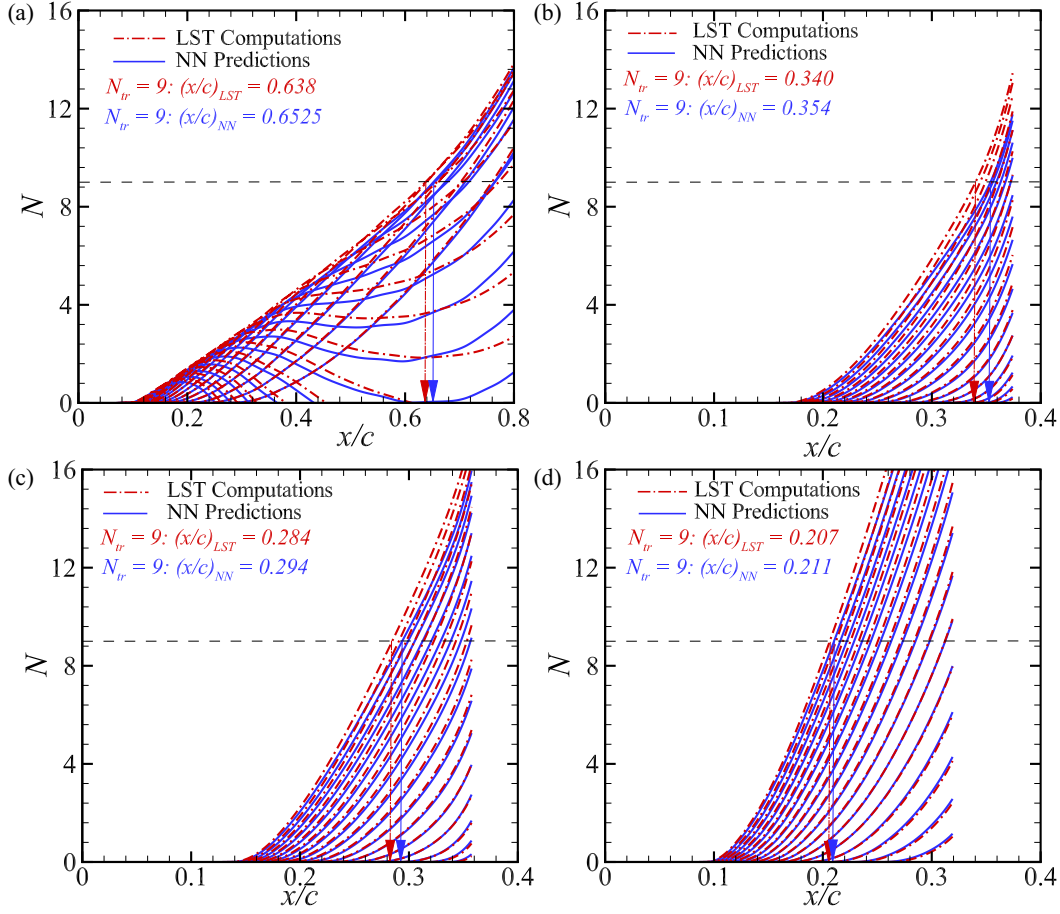


FIG. 3. N -factor curves for non-self-similar boundary-layer profiles over the upper surface of a symmetric HSK airfoil section and an asymmetrical NLF-0416 airfoil at different angles of attack (α). Transition location corresponds to the critical value of $N = 9$ (marked by a dashed line). Corresponding transition onset locations are mentioned on the upper left corner and marked on the horizontal axis as predicted by the proposed neural network C_1 model (blue arrow) and computed by linear stability theory (LST) (red arrow). The neural network model was trained using the Falkner-Skan database (with self-similar boundary layers). Red and blue lines correspond to N -factor curves. (a) HSK airfoil, $\alpha = 0^\circ$, (b) NLF-0416, $\alpha = 0^\circ$, (c) NLF-0416, $\alpha = 2^\circ$, (d) NLF-0416, $\alpha = 5^\circ$.

collectively cover both favorable and adverse pressure gradients. For this assessment, the proposed neural network model has been trained on the complete Falkner-Skan database while the testing dataset corresponds to the upper surface of the airfoil sections. Unlike the Falkner-Skan database, the boundary-layer flows on these airfoils evolve in a non-self-similar manner. Therefore, the evaluation of the neural network trained on the database of self-similar profiles allows us to gauge the practical utility of this model, i.e., the capacity to generalize the predictions to arbitrary, but still attached, boundary-layer profiles. Figure 3 shows the corresponding results where the predicted N -factor curves for instability waves with a selected set of disturbance frequencies have been superposed on those based on the linear stability theory (LST). The predicted N -factor curves are computed by integrating the local amplification rates σ predicted by the proposed neural network model. Figure 3(a) shows the comparison plots for the symmetric HSK airfoil section wherein the abscissa corresponds to the surface location along the upper surface of the airfoil section (scaled with respect

to the airfoil chord length) and the N -factor values are plotted along the ordinate. As discussed in Sec. II A, transition is predicted to occur when the value of N reaches a critical value, which is chosen to be $N_{tr} = 9$ for the purpose of this comparison [9,10]. This critical value of N is marked by a dashed line in the plot, whereas the corresponding transition locations predicted by the neural network and the LST are indicated by blue and red arrows, respectively, and are also listed in the upper left portion of the figure. The error in the neural network prediction for the transition location based on $N_{tr} = 9$ is approximately 2% for the HSK airfoil. Similar results are shown in Figs. 3(b)–3(d) for the predicted transition location along the suction surface of the asymmetric NLF-0416 airfoil section at selected angles of attacks. For these cases, the transition onset location has been predicted to within an error of approximately 2–5 % by the neural network model.

B. Automatic, data-driven feature extraction of boundary layer profiles

The primary advantage of the proposed neural network is the feature extraction capability of the CNN (Fig. 2). The CNN is able to distill information from the boundary-layer profiles in a physically consistent manner, i.e., by considering the boundary-layer profiles as continuous functions and by encoding the information into a set of parameters indicated by the (green) shaded neurons Ψ in Fig. 2(b). The CNN provides a mapping from the space of boundary-layer profiles to the physical parameter space in an automated, data-driven fashion, i.e., without requiring the user to specify an explicitly defined learning target for the CNN. To demonstrate this capability of the proposed neural network, we consider the case in which the CNN maps the distilled information from the boundary-layer profile $U_j (j = 1, 2, \dots, 41)$ to a single parameter Ψ at the interface between the CNN and the fully connected network in Fig. 2. In essence, this process mimics the behavior of the fully connected network A from Table II by choosing the CNN parameters to encode a single feature from the boundary-layer profiles. Because the CNN does not make any prior assumptions about what this single feature should be, one might expect that the predictive performance of the CNN architecture C_1^* would be better than that of the fully connected network A. Somewhat surprisingly, however, comparison of the respective testing errors for the asymmetrical NLF-0416 airfoil indicates that the network A performs slightly better than C_1^* (testing error of 21.2% versus 24.5%). The explanation of this relative performance is left as a topic for future studies. However, it does seem to provide independent evidence that supports the practice of using the analytically defined shape factor H as a nearly optimal scalar representation of the boundary-layer profile for the purpose of predicting the amplification rates. Next, we evaluated the parameter Ψ for each of the 111 boundary-layer profiles along the upper surface of the asymmetrical NLF-0416 airfoil by using the convolutional neural network model that was trained on the full Falkner-Skan database. For convenience of interpretation, we normalize (i.e., shift and scale) the learned parameter Ψ to $\tilde{\Psi}$ such that the latter falls within the range of $[0, 1]$. Figure 4 indicates the variation in $\tilde{\Psi}$ with a similarly normalized shape factor \tilde{H} of the velocity profiles that is defined in the inset of the figure. The plot shows a nearly linear relationship between the CNN-extracted feature $\tilde{\Psi}$ and the physically defined counterpart \tilde{H} . We point out that, because the activation function for the last convolutional layer is linear, the mapping to the parameter Ψ remains unchanged if we scale all of the weights leading into this layer by an arbitrary factor a and the value of all neurons from the layer by $1/a$. As such, the normalization process is well justified.

A similar evaluation of the data-driven feature extraction capability of the CNN was performed by varying the number of boundary-layer profiles used as input to the neural network. Specifically, the correlation analysis from Fig. 4 was repeated by including the first and second derivatives of the velocity profile in addition to the velocity profile itself. The results of this analysis for all three networks (namely, network C_1^* with U , dU/dy , and d^2U/dy^2 profiles as input, C_2^* with U and dU/dy as input, and C_3^* with U only) are presented and compared in Fig. 5. It can be observed that the correlation between $\tilde{\Psi}$ and \tilde{H} is linear for C_3^* , where only the velocity profile U is used as input. In comparison, a mild nonlinearity may be observed in the $\tilde{\Psi}$ - \tilde{H} relations for the C_2^* and C_1^* networks, where the first and second derivatives dU/dy and d^2U/dy^2 of the velocity are

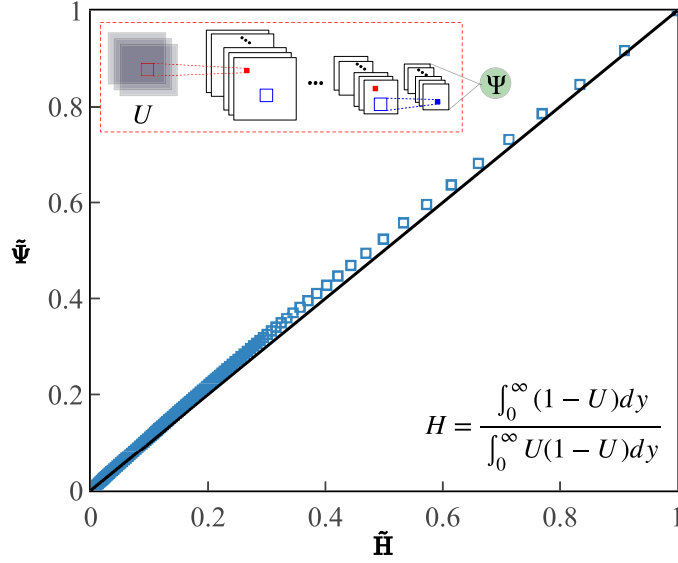


FIG. 4. Correlation between normalized CNN learned parameter ($\tilde{\Psi}$) from non-self-similar boundary-layer profile U and normalized shape factor (\tilde{H}), at 111 locations along the upper surface of an asymmetric NLF-0416 airfoil section at 0 degrees angle of attack. The learned parameter Ψ and the shape parameter H are normalized to within the range $[0, 1]$ to facilitate comparison. The Falkner-Skan database (with self-similar boundary layers) has been used for training.

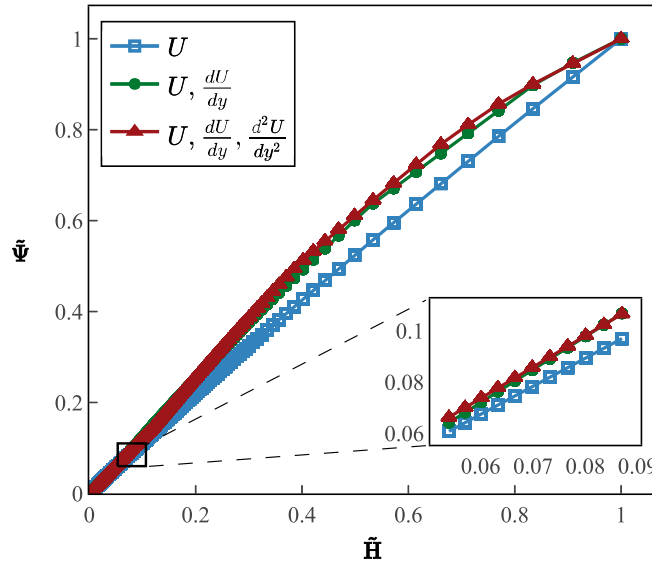


FIG. 5. Comparison of correlation between normalized CNN learned parameter ($\tilde{\Psi}$) for varying sets of non-self-similar boundary layer profiles (corresponding to networks C_1^* , C_2^* , and C_3^*) and normalized shape factor (\tilde{H}), at 111 locations along the upper surface of an asymmetric NLF-0416 airfoil section at 0 degrees angle of attack. Falkner-Skan database (with self-similar boundary layers) has been used for training.

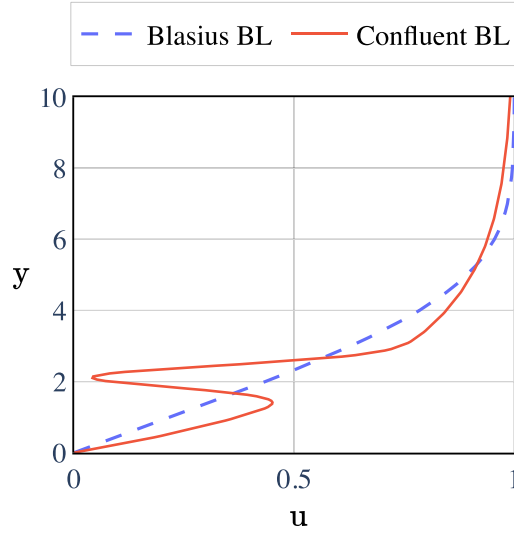


FIG. 6. Comparison of Blasius profile and a confluent boundary layer involving the wake deficit due to an upstream element. The shape factor value of both profiles is 2.59, but the instability characteristics are expected to be very different from each other.

also introduced as additional input features to the CNN. However, whether the relation is linear or nonlinear does not have any major consequence for our purpose. What is more important is that all three networks exhibit a one-to-one correspondence between the CNN-extracted quantity Ψ from the boundary-layer profiles and the physically defined quantity \tilde{H} . Thus, we may conclude that the proposed CNN architecture has a robust performance in goal-oriented feature extraction. In particular, without any explicit instruction from the user, the CNN has been able to encode the boundary-layer profiles (i.e., velocity and its derivatives) into a quantity Ψ that is predictive of the amplification rate σ when used in conjunction with the other physical parameters ω and Re_θ . The quantity Ψ is strongly correlated to the shape factor H , which is known to be correlated with the stability characteristics [13]. This observation points to the physically consistent nature of the proposed neural network architecture. Furthermore, profiles with substantially different stability characteristics can have the same integral shape factor, and therefore the shape factor alone cannot be an adequate predictor of the stability characteristics. As an example, Fig. 6 presents a comparison between the self-similar Blasius boundary-layer profile and a confluent boundary-layer profile with a wake deficit, which may be encountered in the case of boundary layers over multielement airfoil configurations. The integral shape factor value for both profiles is 2.59; however, the stability characteristics of both profiles are expected to be substantially different. The Blasius profile would have the viscous-inviscid interactive instability concentrated closer to the wall, whereas the confluent boundary-layer profile would have primarily inviscid instability within the wake deficit region. However, both profiles would appear the same in the shape factor space. In comparison, the proposed CNN model would use the well-resolved boundary-layer profile to capture the differences between the two profiles toward the prediction of their instability characteristics. In addition to the theoretical importance, the results also have practical importance. Specifically, the results above highlight the intrinsic potential of the proposed network to allow additional features of the boundary-layer profiles to be taken into account with manageable computational costs and without compromising the robustness of the network performance.

Admittedly, the shape factor H of the Falkner-Skan boundary layers can be easily evaluated and then used to map the complex dependence of the disturbance amplification rate on the underlying mean flow through a fully connected neural network [Fig. 1(a)]. However, in several other cases,

such as high-speed flows over blunt nose configurations or flows where the edge of the boundary layer cannot be easily determined, the shape factor H cannot be defined in a consistent and accurate manner and/or computed straightforwardly. References [42,43] highlight some of the difficulties in determining the boundary-layer edge, and hence in computing the shape factor of both transonic [42] and hypersonic [43] laminar boundary-layer flows encountered in practical applications. In such cases, a numerical solution to the full Navier-Stokes equations must be used to compute the boundary-layer flow. For these flows, the proposed convolutional neural network provides a more general and effective architecture for modeling the local instability amplification rates based on the boundary-layer profiles and other relevant physical parameters. Such capability has been demonstrated in a companion paper [44] for the case of a blunt-nosed body in hypersonic flows where, irrespective of the existence of a shape factor (H), the CNN model has been able to predict local instability amplification rates accurately. Although a reduced representation of the input features can also be achieved via dimensionality reduction techniques, such as principal component analysis [20], the proposed neural network architecture provides an easier technique to encode the targeted information from boundary-layer profiles into a smaller set of parameters. This feature extraction capability of the CNN is likely to assume an even greater significance for (i) three-dimensional boundary-layer profiles involving the additional crossflow velocity component, (ii) high-speed flows that involve the profiles of thermodynamic quantities such as the density and/or temperature, and (iii) boundary-layer flows that are inhomogeneous in two spatial coordinates instead of just the wall-normal coordinate, e.g., planar boundary-layer profiles that vary along both the wall-normal and the spanwise directions. One expects that more complex flows such as three-dimensional boundary layers are likely to require increasingly long feature vectors as compared to the Falkner-Skan boundary layers where even a scalar representation of Ψ provided satisfactory results.

Finally, we emphasize that while the shape factor is known to work well for predicting the growth characteristics of TS waves, we emphasize that the proposal to use the shape factor was completely ad hoc and based on domain expertise. The shape factor does not appear anywhere in the mathematical statement of the stability theory, or for that matter in the analytic solutions based on high Reynolds number asymptotic theories of TS waves. The present work presents completely independent evidence extracted from data that the shape factor is a nearly optimal scalar predictor of TS growth rates and how one may develop additional predictors of instability growth rates. Feature extraction capabilities of translation invariant images have been demonstrated in recent work in computer vision and even geosciences [31]; however, the principle of translation invariance does not apply to the boundary-layer velocity profiles, and the success of the CNN as demonstrated here is by no means a foregone conclusion.

C. Comparison to a fully connected network with profile inputs

The proposed convolutional neural network has also been assessed against the straightforward method of directly introducing the full boundary-layer profiles as input features to a fully connected neural network, as shown in Fig. 1(b). The latter architecture may be viewed as a generalization of the architecture in Ref. [24]. Figure 7 shows a comparison between the predictive performances of the proposed convolutional neural network (Fig. 2) and the fully connected network [Fig. 1(b)]. Both neural network models have been trained using the stability database for Falkner-Skan profiles, and the predictive performance is evaluated for instability amplification over the upper surface of the HSK airfoil section. To help ensure a fair comparison, the number of model parameters for both models is kept approximately equal as given in Table I for networks C₁ and B, respectively. The predicted transition onset location based on each model is marked by an arrow on the x -axis and mentioned below the legends in Figs. 7(b) and 7(d). For reference, the predictions based on direct stability computations are also shown. Although the validation plots for instability amplification rates in Figs. 7(a) and 7(c) show better predictive performance for the convolutional neural network, the prediction of transition onset location is better for the fully connected network. The transition

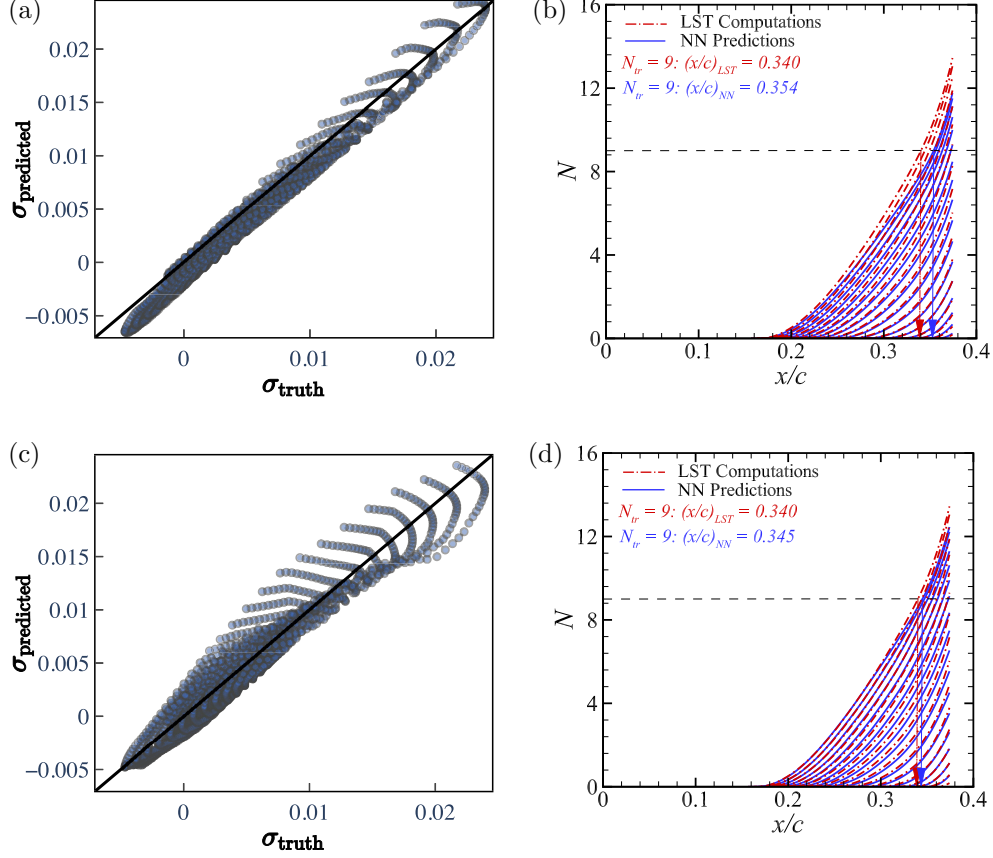


FIG. 7. Comparison between fully connected network B and proposed convolutional neural network C_1 . Validation plots of local instability amplification rates σ and N -factor curves for an asymmetric NLF-0416 airfoil section at 0 degrees angle of attack are given for both networks. Transition location corresponds to the critical value of $N = 9$ (marked by a dashed line). Corresponding transition onset locations are mentioned below the legends in (b) and (d) and marked on the horizontal axis as predicted by the neural network model (NN) (blue arrow) and computed by linear stability theory (LST) (red arrow). (a) Validation plot, growth rate predictions for CNN C_1 , (b) N -factor curves, CNN C_1 , (c) Validation plot, growth rate predictions for fully connected network B and (d) N -factor curves, fully connected network B.

location predicted by the fully connected network is within 1.1% of the transition onset location based on the linear stability theory, whereas the convolutional neural network predicts the same with a 2.3% error. Even though both measures of error are rather small, the qualitative trend is somewhat unexpected and requires further investigation in future studies.

One significant advantage of the proposed convolutional neural network (Fig. 2) over the fully connected network [Fig. 1(b)] pertains to the number of trainable model parameters required to achieve a comparable performance. With full information available as input features, a fully connected network would require a significantly higher number of model parameters, and consequently a higher training cost to provide a comparable performance, whereas the convolutional neural network is likely to provide more robust predictive performance with a smaller number of model parameters. Figure 8 presents the results of the analysis designed to verify this behavior. Here, the test error percentage has been plotted against the size of the neural network model as measured by the total number of trainable model parameters. We observe that the performance of

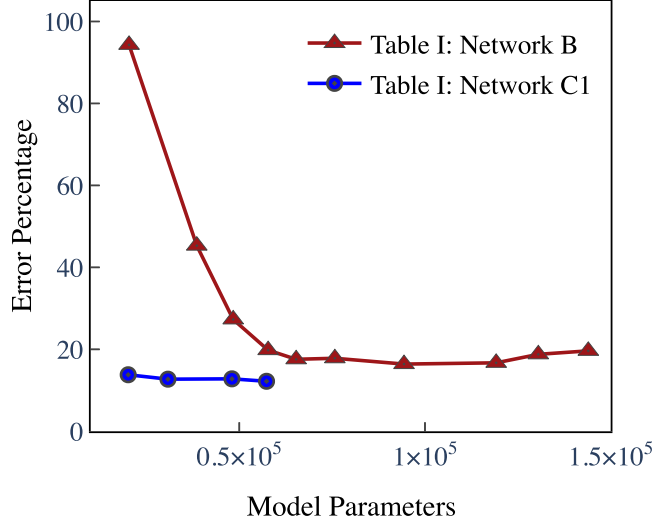


FIG. 8. Comparison of testing error for the fully connected neural network (FC NN) [Fig. 1(b), Network B from Table I] and the convolutional neural network (Fig. 2, Network C₁ from Table I) as a function of the total number of learnable model parameters. Input features for both models include boundary-layer profiles. The Falkner-Skan database (with self-similar boundary layers) was used for training while the testing error was evaluated for an asymmetric NLF-0416 airfoil section with non-self-similar boundary layers at 0 degrees angle of attack.

the fully connected neural network deteriorates significantly as the number of model parameters is reduced, while the proposed convolution-based neural network is able to maintain its predictive performance. The convolutional neural network in the proposed model encodes the information from the boundary-layer profiles into a significantly smaller number of scalar parameters (eight parameters for network C₃) before folding them into the ensuing fully connected portion of the overall network, along with the other physical parameters (namely, frequency of the instability wave ω and the local Reynolds number Re_θ). Such feature engineering for the boundary-layer profiles via the convolutional neural network enables one to achieve a comparable performance with a significantly reduced number of model parameters and training cost.

IV. CONCLUSION

A neural-network-based transition model has been presented that is capable of accurately predicting the transition onset location for incompressible, two-dimensional attached flows in a physically informed manner without requiring the direct computations using the linear stability theory. The proposed model has the ability to encode information, using convolutional layers, from boundary-layer profiles (velocity and its derivatives) into a set of integral quantities. More importantly, the encoded feature Ψ shows strong correlation with, or even one-to-one correspondence to the physically defined shape parameter H , which clearly demonstrates the physically consistent nature of the proposed neural network. These encoded integral quantities are then nonlinearly mapped to local instability amplification rate σ along with other scalar disturbance characteristics (ω and Re_θ). The proposed model is shown to have robust predictive performance, clear physical interpretation, and superior computational efficiency.

The CNN architecture presented herein can be easily generalized to other instability mechanisms and, in follow-on (and as yet unpublished work), we have demonstrated the application of this architecture to second mode instabilities in high-speed boundary layers that cannot be predicted well on the basis of local shape factors of the boundary-layer profiles. Thus, it could become the means

for physics-based transition prediction in practical applications of computational fluid dynamics codes.

ACKNOWLEDGMENTS

This work was supported by the Transformational Tools and Technologies project of the NASA Transformative Aeronautics Concepts Program. The computational resources used for this project were provided by the Advanced Research Computing (ARC) of Virginia Tech, which is gratefully acknowledged.

-
- [1] J. E. Green, Air Travel–Greener by Design Mitigating the environmental impact of aviation: Opportunities and priorities, *Aeronaut. J.* **109**, 361 (2005).
 - [2] L. M. Mack, Review of linear compressible stability theory, in *Stability of Time Dependent and Spatially Varying Flows*, edited by D. L. Dwoyer and M. Y. Hussaini, ICASE NASA LaRC Series (Springer, New York, 1987).
 - [3] H. L. Reed, W. S. Saric, and D. Arnal, Linear stability theory applied to boundary layers, *Annu. Rev. Fluid Mech.* **28**, 389 (1996).
 - [4] M. P. Juniper, A. Hanifi, and V. Theofilis, Modal stability theory: Lecture notes from the flow-nordita summer school on advanced instability methods for complex flows, *Appl. Mech. Rev.* **66**, 024804 (2014).
 - [5] K. Taira, S. L. Brunton, S. T. M. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley, Modal analysis of fluid flows: An overview, *AIAA J.* **55**, 4013 (2017).
 - [6] D. Arnal, Boundary layer transition: Prediction based on linear stability theory, in *Special Course on Progress in Transition Modeling, AGARD-R-793* (AGARD, Neuilly-sur-Seine, France, 1994), pp. 1–63.
 - [7] L. M. Mack, Transition prediction and linear stability theory, in *Special Course on Stability and Transition of Laminar Flows, AGARD R-709* (AGARD, Neuilly-sur-Seine, France, 1984), pp. 1–232.
 - [8] E. Reshotko, Boundary-layer stability and transition, *Annu. Rev. Fluid Mech.* **8**, 311 (1976).
 - [9] J. van Ingen, The e^n method for transition prediction, Historical review of work at TU Delft, in *38th Fluid Dynamics Conference and Exhibit, AIAA 2008-3830* (AIAA, Seattle, 2008), Vol. 1, pp. 1–49.
 - [10] D. M. Bushnell, M. R. Malik, and W. D. Harvey, Transition prediction in external flows via linear stability theory, in *Symposium Transsonicum III*, edited by J. Zierep and H. Oertel (Springer, Berlin, 1989), pp. 225–242.
 - [11] K. Kusunose, L. Wigton, and P. Meredith, A rapidly converging viscous/inviscid coupling code for multi-element airfoil configurations, in *29th Aerospace Sciences Meeting, AIAA-91-0177* (AIAA, Seattle, 1991), Vol. 1, pp. 1–11.
 - [12] C. Sheng, Role of transition modeling in rotor hover predictions, *J. Aircraft* **55**, 23 (2018).
 - [13] M. Drela and M. B. Giles, Viscous-inviscid analysis of transonic and low Reynolds number airfoils, *AIAA J.* **25**, 1347 (1987).
 - [14] J. P. Perraud and A. Durant, Stability-based Mach zero to four longitudinal transition prediction criterion, *J. Spacecr. Rockets* **53**, 730 (2016).
 - [15] J. R. Dagenhart, Amplified cross flow disturbances in the Laminar layer on swept wings with suction, Technical Report No. NASA-TP-1902, NASA Langley Research Center, 1981.
 - [16] H. W. Stock and E. Degenhart, A simplified e^n method for transition prediction in two-dimensional, incompressible boundary layers, *Z. Flugwissen. Weltraumforsch.* **13**, 16 (1989).

- [17] M. Gaster and F. Jiang, Rapid scheme for estimating transition on wings by linear stability theory, in *Proceedings ICAS International Council of the Aeronautical Sciences* (ICAS, Anaheim, CA, 1995), Vol. 3, pp. 1104–1113.
- [18] M. Langlois, C. Masson, F. Kafyeke, and I. Paraschivoiu, Automated method for transition prediction on wings in transonic flows, *J. Aircraft* **39**, 460 (2002).
- [19] A. Krumbein, e^n transition prediction for 3D wing configurations using database methods and a local, linear stability code, *Aerosp. Sci. Technol.* **12**, 592 (2008).
- [20] D. G. Rajnarayan and P. Sturdza, Extensible rapid transition prediction for aircraft conceptual design using modal decomposition, in *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, AIAA 2013-0231 (AIAA, Grapevine, TX, 2013), Vol. 1, pp. 1–21.
- [21] G. Begou, H. Deniau, O. Vermeersch, and G. Casalis, Database approach for laminar-turbulent transition prediction: Navier-Stokes compatible reformulation, *AIAA J.* **55**, 3648 (2017).
- [22] F. Pinna, L. Zanus, S. Demange, and M. Olazabal-Loume, Reduced model for transition prediction in hypersonic flows, in *2018 Fluid Dynamics Conference*, AIAA 2018-3697 (AIAA, Atlanta, GA, 2018), Vol. 1, pp. 1–10.
- [23] J. G. Coder and M. D. Maughmer, Computational fluid dynamics compatible transition modeling using an amplification factor transport equation, *AIAA J.* **52**, 2506 (2014).
- [24] J. D. Crouch, I. W. M. Crouch, and L. L. Ng, Transition prediction for three-dimensional boundary layers in computational fluid dynamics applications, *AIAA J.* **40**, 1536 (2002).
- [25] P. Paredes, B. Venkatachari, M. Choudhari, F. Li, C.-L. Chang, M. I. Zafar, and H. Xiao, Extensible rapid transition prediction for aircraft conceptual design using modal decomposition, in *AIAA SciTech Forum* (AIAA, Orlando, FL, 2020), Vol. 1, pp. 1–21.
- [26] R. M. Fuller, W. R. Saunders, and U. Vandsburger, Neural network estimation of disturbance growth using a linear stability numerical model, in *35th Aerospace Sciences Meeting and Exhibit*, AIAA-97-0559 (AIAA, Reno, 1997), Vol. 1, pp. 1–10.
- [27] F. Danvin, F. Pinna, and M. Olazabal-Loume, Laminar to turbulent transition prediction in hypersonic flows with meta-models, in *2018 Fluid Dynamics Conference*, AIAA-2018-3701 (AIAA, Atlanta, 2018), Vol. 1, pp. 1–18.
- [28] P. G. Drazin and W. H. Reid, *Hydrodynamic Stability* (Cambridge University Press, Cambridge, 1981).
- [29] K. Hornik, M. Stinchcombe, and H. White, Multilayer feedforward networks are universal approximators, *Neural Netw.* **2**, 359 (1989).
- [30] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, 2006).
- [31] J. Wu, X. Yin, and H. Xiao, Seeing permeability from images: fast prediction with convolutional neural networks, *Sci. Bull.* **63**, 1215 (2018).
- [32] D. P. Kingma and L. J. Ba, Adam: A method for stochastic optimization, in *3rd International Conference for Learning Representations* (Ithaca, NY, 2015).
- [33] C.-L. Chang, Langley Stability and Transition Analysis Code (LASTRAC) Version 1.2 User Manual, NASA/TM-2004-213233 (2004).
- [34] F. T. Smith, On the non-parallel flow stability of the Blasius boundary layer, *Proc. R. Soc. London A* **366**, 91 (1979).
- [35] H. Kanner, J. Schetz, and R. Wlezien, The evolution of an acoustic disturbance up to transition in the boundary layer on an airfoil, in *30th Fluid Dynamics Conference*, AIAA-99-3791 (AIAA, Norfolk, VA, 1999), Vol. 1, pp. 1–11.
- [36] D. M. Somers, Design and experimental results for a natural-laminar-flow airfoil for general aviation applications, Technical Report No. NASA-TP-1861, NASA Langley Research Center, 1981.
- [37] A. Gopalarathnam and M. S. Selig, Low-speed natural-laminar-flow airfoils: Case study in inverse airfoil design, *J. Aircraft* **38**, 57 (2001).
- [38] M. Raissi, P. Perdikaris, and G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* **378**, 686 (2019).
- [39] J. Ling, A. Kurzawski, and J. Templeton, Reynolds averaged turbulence modeling using deep neural networks with embedded invariance, *J. Fluid Mech.* **807**, 155 (2016).

- [40] J. Wang, J. L. Wu, and H. Xiao, Physics informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data, [Phys. Rev. Fluids **2**, 034603 \(2017\)](#).
- [41] J. L. Wu, H. Xiao, and E. G. Paterson, Physics-informed machine learning approach for augmenting turbulence models: A comprehensive frame-work, [Phys. Rev. Fluids **3**, 074602 \(2018\)](#).
- [42] J. Schneider and B. Ewald, Calculation of transonic laminar flow airfoils using a Navier-Stokes method and linear stability theory, in *Proceedings of 1994 International Council of the Aeronautical Sciences*, Anaheim ICAS-94-4.7.2, 2456 (AIAA, Anaheim, 1994).
- [43] D. A. Saunders and D. K. Prabhu, Blayer user guide, NASA/TM—2018-219749, NASA Ames Research Center, 2018.
- [44] P. Paredes, B. Venkatachari, M. Choudhari, F. Li, C.-L. Chang, M. I. Zafar, and H. Xiao, Toward a practical method for hypersonic transition prediction based on stability correlations, [AIAA J. **58**, 4475 \(2020\)](#).

Chapter 3

Recurrent neural network for end-to-end modeling of laminar-turbulent transition

RESEARCH ARTICLE

Recurrent neural network for end-to-end modeling of laminar-turbulent transition

Muhammad I. Zafar¹, Meelan M. Choudhari², Pedro Paredes³ and Heng Xiao^{1,*} 

¹Kevin T. Crofton Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, Virginia, USA

²Computational AeroSciences Branch, NASA Langley Research Center, Hampton, Virginia, USA

³National Institute of Aerospace, Hampton, Virginia, USA

*Corresponding author. E-mail: hengxiao@vt.edu

Received: 26 March 2021; **Revised:** 15 June 2021; **Accepted:** 29 June 2021

Keywords: Laminar-turbulent transition; scientific machine learning; recurrent neural network

Abstract

Accurate prediction of laminar-turbulent transition is a critical element of computational fluid dynamics simulations for aerodynamic design across multiple flow regimes. Traditional methods of transition prediction cannot be easily extended to flow configurations where the transition process depends on a large set of parameters. In comparison, neural network methods allow higher dimensional input features to be considered without compromising the efficiency and accuracy of the traditional data-driven models. Neural network methods proposed earlier follow a cumbersome methodology of predicting instability growth rates over a broad range of frequencies, which are then processed to obtain the N-factor envelope, and then, the transition location based on the correlating N-factor. This paper presents an end-to-end transition model based on a recurrent neural network, which sequentially processes the mean boundary-layer profiles along the surface of the aerodynamic body to directly predict the N-factor envelope and the transition locations over a two-dimensional airfoil. The proposed transition model has been developed and assessed using a large database of 53 airfoils over a wide range of chord Reynolds numbers and angles of attack. The large universe of airfoils encountered in various applications causes additional difficulties. As such, we provide further insights on selecting training datasets from large amounts of available data. Although the proposed model has been analyzed for two-dimensional boundary layers in this paper, it can be easily generalized to other flows due to embedded feature extraction capability of convolutional neural network in the model.

Impact Statement

The recurrent neural network (RNN) proposed here represents a significant step toward an end-to-end prediction of laminar-turbulent transition in boundary-layer flows. The general yet greatly simplified workflow should allow even nonexperts to apply the proposed model for predicting transition due to a variety of instability mechanisms, which is a significant advantage over traditional direct computations of the stability theory. The encoding of boundary layer profiles by the convolutional neural network and sequence-to-sequence mapping enabled by the RNN faithfully represent the amplification of flow instability along the surface, exemplifying the direct correlation of the proposed model with the underlying physics. Finally, we use a very large dataset and provide insights and best-practice guidance toward the practical deployment of neural network-based transition models in engineering environments.

© The United States Government as represented by the Administrator of the National Aeronautics and Space Administration, National Institute of Aerospace and the Author(s), 2021. To the extent this is a work of the US Government, it is not subject to copyright protection within the United States. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

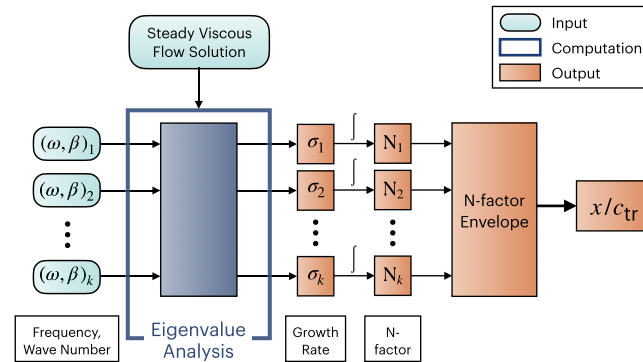
1. Introduction

Laminar-turbulent transition of boundary-layer flows has a strong impact on the performance of flight vehicles across multiple flow regimes due to its effect on surface skin friction and aerodynamic heating. Predicting the transition location in computational fluid dynamics (CFD) simulations of viscous flows remains a challenging area (Slotnick et al., 2014). Transition to turbulence in a benign disturbance environment is typically initiated by the amplification of modal instabilities of the laminar boundary layer. Depending on the flow configuration, these instabilities can be of several types, for example, Tollmien–Schlichting (TS) waves, oblique first-mode instabilities, and planar waves of second-mode (or Mack-mode) type.

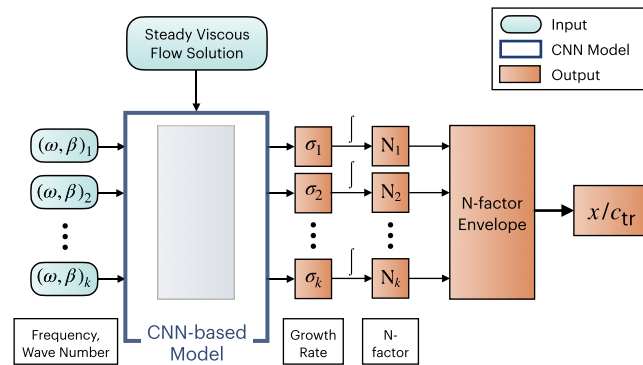
A description of transition prediction methods based on stability correlations can be found in a variety of references (Smith and Gamberoni, 1956; van Ingen, 1956, 2008), but we provide a brief description here to make the paper self-contained. A somewhat expanded description may also be found in Zafar et al. (2020). The transition process begins with the excitation of linear instability waves that undergo a slow amplification in the region preceding the onset of transition. The linear amplification phase is followed by nonlinear processes that ultimately lead to turbulence. Since these nonlinear processes are relatively rapid, it becomes possible to predict the transition location based on the evolution of the most amplified instability mode. The linear amplification ratio, e^N , is generally computed using the classical linear stability theory (Reshotko, 1976; Mack, 1987; Reed et al., 1996; Juniper et al., 2014; Taira et al., 2017). The local streamwise amplification rates (σ) along the aerodynamic surface can be determined by solving an eigenvalue problem for the wall-normal velocity perturbation, which is governed by the Orr–Sommerfeld equation (Drazin and Reid, 1981). These local streamwise amplification rates corresponding to each frequency (ω) are then integrated along the body curvature to obtain the logarithmic amplification of the disturbance amplitude (N-factor) for each disturbance frequency (ω). The e^N method assumes that the occurrence of transition correlates very well with the N-factor of the most amplified instability wave reaching a critical value N_{tr} . For subsonic and supersonic flows, the critical N-factor (denoted herein as N_{tr}) has been empirically found to lie in the range between 9 and 11 (Bushnell et al., 1989; van Ingen, 2008). Such a prediction process may be schematically illustrated as shown in Figure 1a.

Linear stability computations rely on highly accurate computations of the mean boundary-layer flow. Solution of the linear stability equations is also computationally expensive and often leads to the contamination of the unstable part of the spectrum by spurious eigenvalues. The nonrobust nature of the stability computations requires a significant degree of expertise in stability theory on the user's part, making such computations inapt for nonexpert users. For these reasons, transition prediction based on stability computations has been difficult to automate and renders its direct integration in CFD solvers rather impractical. Several aerodynamic applications involving flow separation also entail viscous–inviscid interactions. Such interactions lead to a strong coupling between transition and the overall flow field, which requires an iterative prediction approach. Hence, the integration of the transition prediction method in the overall aerodynamic prediction method remains an important area of research (Slotnick et al., 2014).

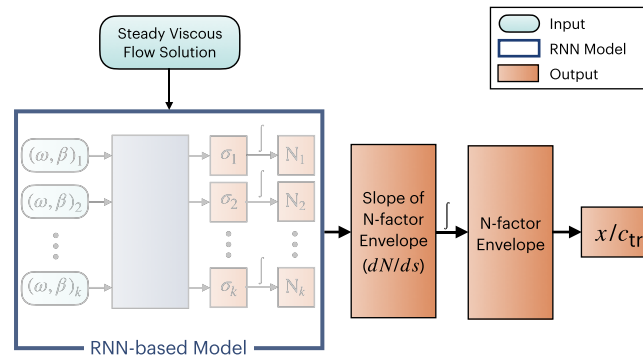
Several methods have been proposed as simplifications or surrogate models of the e^N methods, including database query techniques (Drela and Giles, 1987; van Ingen, 2008; Perraud and Durant, 2016) and data fitting techniques (Dagenhart, 1981; Stock and Degenhart, 1989; Gaster and Jiang, 1995; Langlois et al., 2002; Krumbein, 2008; Rajnarayan and Sturdza, 2013; Begou et al., 2017; Pinna et al., 2018). These methods are generally based on a small set of scalar input parameters representing the mean flow parameters and relevant disturbance characteristics. However, these methods do not scale well with larger sets of parameters, which tends to limit the expressive power of the transition model based on these traditional methods (Crouch et al., 2002). In particular, the shape factor is a commonly used scalar parameter to correlate the disturbance amplification rates to the mean flow of the boundary layer. However, the shape factor cannot be easily computed for many practical flows such as high speed flows



(a) Linear Stability Theory (LST): Growth rates are computed for each instability wave characterized by frequency (ω) and wave number (α). Growth rates are integrated (\int) along the airfoil contour to obtain corresponding N-factor values, from which the N-factor envelope and transition location (x/c_{tr}) are determined for a given value of correlating N-factor ($N = N_{tr}$). The most amplified Tollmien-Schlichting (TS) waves in two-dimensional boundary layers correspond to two-dimensional instability waves ($\beta = 0$) (Rajnarayan et al., 2013).



(b) Convolutional Neural Network-based model (CNN): Instead of using eigenvalue analysis, local growth rates corresponding to each frequency (ω) are predicted using a neural network (41).



(c) Recurrent Neural Network (RNN) model for N-factor envelope modelling: Growth rate of the N-factor envelope (dN/ds) is directly predicted, which is then integrated (\int) along the airfoil contour to obtain the N-factor envelope and estimated transition location (x/c_{tr}) for a given value of correlating N-factor ($N = N_{tr}$).

Figure 1. Comparison of transition prediction methodologies.

over blunt leading edges, which results in a poor predictive performance of the database methods (Paredes et al., 2020).

Neural networks provide a more generalized way of predicting the instability characteristics, while also accounting for their dependency on high-dimensional input features in a computationally efficient and robust manner. Fuller et al. (1997) applied neural network methods to instability problems in predicting the instability growth rates for a jet flow. Crouch et al. (2002) used scalar parameters and the wall-normal gradient of the laminar velocity profile as an input of neural networks to predict the maximum instability growth rates. They demonstrated the generalizability of the neural network method for both TS waves and stationary crossflow instabilities. The data for the gradient of the laminar velocity profile were coarsely defined at six equidistant points across the boundary layer. A fully connected neural network was used, which assumes no spatial structure on the input data. Such treatment of boundary-layer profiles may not be well suited for other instability mechanisms involving, for instance, Mack-mode instabilities in high-speed boundary layers that require input profiles of thermodynamics quantities along with the velocity profiles (Paredes et al., 2020) or the secondary instabilities of boundary-layer flows with finite-amplitude stationary crossflow vortices that include rapid variations along both wall-normal and spanwise coordinates.

By utilizing recent developments in machine learning, Zafar et al. (2020) proposed a transition model based on convolutional neural networks (CNNs), which has the ability to generalize across multiple instability mechanisms in an efficient and robust manner. CNNs were used to extract a set of latent features from the boundary-layer profiles, and the extracted features were used along with other scalar quantities as input to a fully connected network. The hybrid architecture was used to predict the instability growth rates for TS instabilities in two-dimensional incompressible boundary layers. The extracted latent feature showed a strong, nearly linear correlation with the analytically defined shape factor (H) of the boundary-layer velocity profile. The model was trained using a database of Falkner–Skan family of selfsimilar boundary-layer profiles. This CNN-based method is applicable to various instability mechanisms with higher-dimensional input features, since the boundary-layer profiles are treated in a physically consistent manner (i.e., as discrete representation of the profiles accounting for their spatial structures). It has been applied to predict the instability growth rates of Mack-mode instabilities in hypersonic flows over a moderately blunt-nosed body (Paredes et al., 2020). This particular application requires additional input features in the form of boundary-layer profiles of thermodynamic quantities such as temperature and/or density, where the CNN-based model demonstrated highly accurate predictions of the instability growth rates.

Despite clear advantages over earlier neural network-based models, the CNN-based transition model shares a few significant shortcomings with the direct integration of linear stability theory toward transition prediction. Similar to the stability theory, the CNN-based transition model is based on the predictions of the instability growth rates corresponding to each selected disturbance frequency ω (and/or the spanwise wavenumber β in the case of three-dimensional instabilities) and every station from the input set of boundary-layer profiles. The instability growth rates for each individual disturbance are integrated along the aerodynamic surface to predict the growth in disturbance amplitude, or equivalently, the N-factor curve for each combination of frequency and spanwise wavenumber. Finally, one must determine the envelope of the N-factor curves to predict the logarithmic amplification ratio for the most amplified disturbance at each station (denoted as N_{env} herein), which can then be used in conjunction with the critical N-factor (N_{cr}) based on previous experimental measurements to predict the transition location (see Figure 1b for a summary). This overall workflow not only extends over several steps, but also requires the user to estimate the range of frequencies (and/or spanwise wavenumbers) that would include the most amplified instability waves corresponding to the envelope of the N-factor curves. The selection of disturbance parameters for a given flow configuration can be somewhat challenging for nonexpert users. More important, however, the above workflow requires several redundant growth rate computations involving subdominant disturbances that do not contribute to the N-factor envelope used to apply the transition criterion, namely, $N = N_{\text{cr}}$. Finally, and similar to the earlier neural network models (Crouch et al., 2002), the growth rate prediction during the all important first step of the above workflow only uses

the local boundary-layer profiles, and hence, does not utilize any information about the prior history of a given disturbance, for example, any previously estimated instability growth rates at the upstream locations. Since the boundary-layer profiles evolve in a continuous manner, the spatial variation in the disturbance growth rate represents an analytic continuation along the aerodynamic surface. Thus, embedding the upstream history of boundary-layer profiles and/or the disturbance growth rates should lead to more accurate, robust, and computationally efficient models for the onset of transition.

A recurrent neural network (RNN) is a promising approach for modeling the history effects. The RNN is a general-purpose architecture for modeling sequence transduction using an internal state (memory) that selectively keeps track of the information at the preceding steps during the sequence (Graves, 2012). RNNs provide a combination of multivariate internal state as well as nonlinear state-to-state dynamics, which make it particularly well-suited for dynamic system modeling. Faller and Schreck (1997) exploited these attributes of RNNs to predict unsteady boundary-layer development and separation over a wing surface. The RNN architectures have also been used for the modeling of several other complex dynamic systems ranging from near-wall turbulence (Guastoni et al., 2019), the detection of extreme weather events (Wan et al., 2018), and the spatiotemporal dynamics of chaotic systems (Vlachas et al., 2020), among others.

The feature extraction capability of CNN and the sequence-to-sequence mapping enabled by RNN provide a direct correlation with the underlying physics of transition, exemplifying the machine learning models motivated by the physics of the problem, for example, in the modeling of turbulence (Wang et al., 2017; Wu et al., 2018b; Duraisamy et al., 2019). Transport equation-based models, such as the well-known Langtry–Menter 4-equation model Menter et al. (2006), are based on empirical transition correlations that are based on local mean flow parameters, therefore the connection with the underlying physics of the transition process is significantly weaker as compared to stability-based correlation, whether it involves direct computations of linear stability characteristics or a proxy thereto as represented within the proposed RNN model. This paper is aimed at exploiting the sequential dependency of mean boundary-layer flow properties to directly predict maximum growth rates among all unstable modes at a sequence of stations along the airfoil surface. Such sequential growth rates can then be integrated along the airfoil surface to determine the N-factor envelope and corresponding transition location, as has been schematically illustrated in Figure 1c. To this end, an extensive airfoil database has been used that documents mean flow features and linear stability characteristics for a large set of airfoils at a range of flow conditions (Reynolds numbers and angles of attack). Furthermore, we provide insight on the similarity of stability characteristics among different families of airfoils and how a neural network trained on one set of airfoils can generalize to other ones, possibly at different flow conditions.

The rest of the manuscript is organized as follows. The proposed RNN model is introduced in Section 2 along with the input and output features. Section 3 presents the airfoil database used to develop and evaluate the proposed transition model. Section 4 presents the results and discussion for different training and testing cases, which provide insight toward subsampling of training datasets for achieving a reasonable predictive performance from the RNN model. Section 5 concludes the paper.

2. Recurrent Neural Network

A neural network consists of successive composition of linear mapping and nonlinear squashing functions, which aims to learn the underlying relationship between an input vector (q) and an output vector (y) from a given set of training data. The series of functions are organized as a sequence of layers, each containing several neurons that represent specific mathematical functions. The mathematical functions in each layer are parameterized by the weight (W) and bias (b). Intermediate layers between the input layer (q) and output layer (y) are known as hidden layers. The functional mapping of a neural network with a single hidden layer can be expressed as:

$$y = W^{(2)} \left(f \left[W^{(1)} q + b^{(1)} \right] \right) + b^{(2)}, \quad (1)$$

where $W^{(l)}$ and $b^{(l)}$ represent the weight matrix and bias vector for the l th layer, respectively, and f is an activation function. Activation functions enable the representation of complex functional mapping by introducing nonlinearity in the composite functions. The training of a neural network is a process of learning the weights and biases with the objective of fitting the training data.

RNNs are architectures with internal memory (known as the *hidden states*), which make them particularly suitable for sequential data such as time series, spatial sequences, and words in a text. The RNN processes the sequence of inputs in a step-by-step manner while selectively passing the information across a sequence of steps encoded in a hidden state. At any given step i , the RNN operates on the current input vector (q_i) in the sequence and the hidden state h_{i-1} passed on from the previous step, to produce an updated hidden state h_i and an output y_i . Figure 2 shows the schematic of a recurrent neural network. Multiple RNNs can be stacked over each other, as shown in Figure 3, to provide a deep RNN. The functional mapping for an architecture with L layers of RNN stacked over each other can be expressed as:

$$h_i^l = f \left[W_{hh}^l \cdot h_{i-1}^l + W_{qh}^l \cdot h_i^{l-1} \right], \quad (2a)$$

$$y_i = W_{hy} \cdot h_i^L, \quad (2b)$$

where W_{hh}^l , W_{qh}^l , and W_{hy} are the model parameters corresponding to the mapping from a previous hidden state to subsequent hidden state, from an input vector to a hidden state, and from a hidden state to an output vector, respectively. The model parameters (W_{hh}^l and W_{qh}^l) have sequential invariance across each layer, that is, the input vector and hidden state at each step along the sequence are processed by the same parameters within a given layer of the RNN architecture. For the first layer, h_i^{l-1} is equivalent to the input vector q_i , while for subsequent layers, h_i^{l-1} denotes the hidden state from the previous layer at the current step. In this manner, a multilayer RNN transmits the information encoded in the hidden state to the next step in the current layer and to the current step of the next layer by implementing Equation (2a). For the sake of brevity, the bias terms have not been mentioned in these equations.

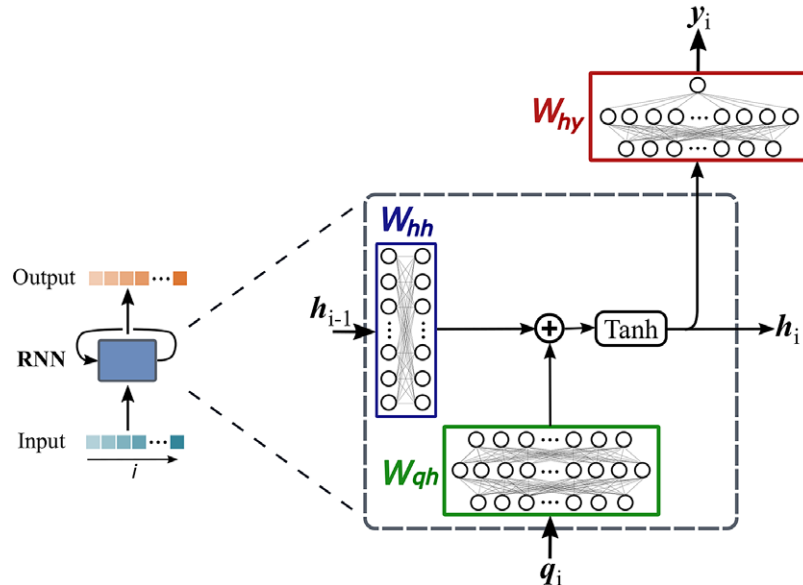


Figure 2. Schematic of the recurrent neural network (RNN) cell shown as a blue box on the left. Within each RNN cell, the arrangement of the weight matrices is shown on the right. At any step i of the sequence, the RNN cell takes input q_i and previous hidden state h_{i-1} , and provides updated hidden state h_i and output y_i .

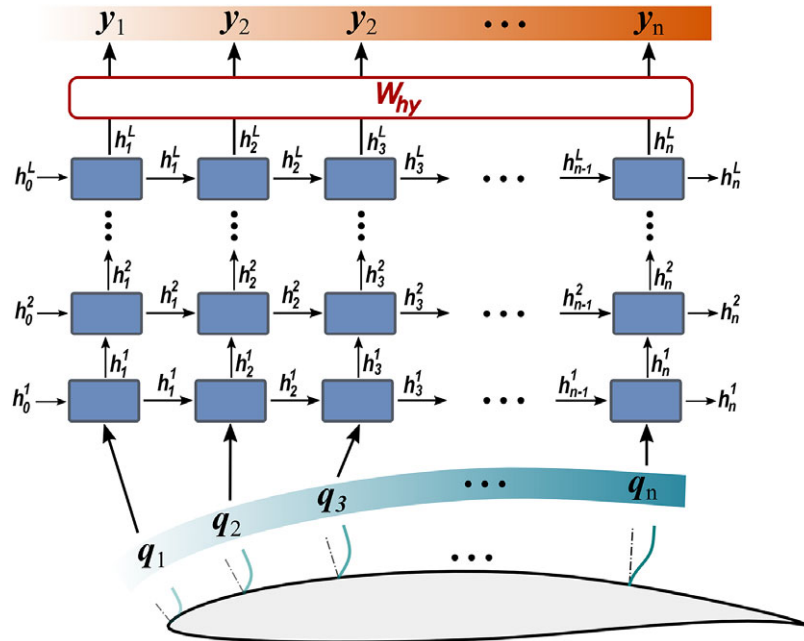


Figure 3. Sequences of input features and output for deep recurrent neural network (RNN) architecture have been illustrated with respect to stations along the airfoil surface.

Like deep feed-forward neural networks, deep RNNs can lead to more expressive models. However, the depth of an RNN can have multiple interpretations. In general, RNN architectures with multiple RNN layers stacked over each other are considered *deep RNNs*, as shown in Figure 3. Such deep RNN architectures have multiple internal memories (hidden states), one in each RNN layer. RNN architectures with multiple recurrent hidden states, stacked over each other, can model varying levels of dependencies (i.e., from short-term to long-term) in each hidden state (Hermans and Schrauwen, 2013). These stacked-RNN architectures can still be considered shallow networks with limited expressivity, as all model parameters (W_{hh} , W_{qh} , and W_{hy}) are generally represented by single linear layers. To allow for more complex functional representation, these single linear layers can be replaced by multiple nonlinear layers. Pascanu et al. (2014) has shown that introducing depth via multiple nonlinear layers to represent W_{hh} , W_{qh} , and W_{hy} can lead to better expressivity of the RNN model. For the transition modeling problem addressed in this paper, multiple nonlinear layers are used within each RNN cell to express the complex physical mapping between the input features and output, as shown in Figure 2. Such architecture resulted in better learning and predictive capability of the RNN model.

The underlying physics of transition does not require long term memory, unlike natural language processing for which more involved models like long short-term memory (LSTM) and transformers have proved to be very effective. For the transition problem, keeping track of last one or two stations have proved to be sufficient. In a study at the start of this research work, an informal investigation showed no advantage of LSTMs over RNNs, despite their added complexity and higher training cost.

With this perspective, the RNN model being proposed in this paper maps the sequential dependency of mean boundary-layer flow properties as input features to instability growth rates corresponding to the N-factor envelope as output features. Such input and output features, summarized in Table 1, have been taken at a sequence of stations along the airfoil surface as shown in Figure 3. Mean boundary-layer flow properties have been introduced in terms of the Reynolds number (Re_θ) based on the local momentum thickness of the boundary layer, the velocity boundary-layer profile (u), and its derivatives (du/dy and d^2u/dy^2). Zafar et al. (2020) proposed a convolutional neural network model that encodes the information

from boundary-layer profiles to a vector of latent features while accounting for the spatial patterns in the input profiles (Michelen-Ströfer et al., 2018; Wu et al., 2018a). Such a treatment of boundary-layer profiles allows the trained neural network models to generalize to all practical flows with different instability mechanisms (Paredes et al., 2020).

The RNN model presented in this paper builds upon this idea and further combines CNN with RNN to account for nonlocal physics in both streamwise and wall-normal directions. This is shown in Figure 4. The hyperparameters of the proposed neural network have been empirically tuned to yield adequate complexity for learning all the required information, without causing an overfitting of the training data. After such hyperparameter tuning of the neural network model, early stopping was not required. With the boundary-layer profiles defined by using 41 equidistant points in the wall-normal direction, the CNN architecture contains 3 convolutional layers with 6, 8, and 4 channels, respectively, in those layers. Kernel size of 3×1 has been used in each convolutional layer. Rectified linear unit (ReLU) is used as the activation function. The CNN encodes the spatial information in the boundary-layer profiles along each station to a vector of latent features, Ψ . The results are not significantly sensitive to the number of latent features in the vector Ψ . However, following a sensitivity study on the size of latent features, the number of elements in Ψ has been set to 8. The latent features Ψ extracted from the boundary-layer profiles are then

Table 1. Input features and output for the RNN model.

Feature/output	Description	Definition
q_1	Reynolds number based on edge velocity and momentum thickness	Re_θ
q_2	Velocity profile as a function of wall normal coordinate y	$u_j, j = 1, 2, \dots, 41$
q_3	First-order derivative of velocity profile	$\frac{du}{dy}\bigg _j, j = 1, 2, \dots, 41$
q_4	Second-order derivative of velocity profile	$\frac{d^2u}{dy^2}\bigg _j, j = 1, 2, \dots, 41$
y_1	Slope of N-factor envelope, corresponding to local growth rate of the most amplified disturbance at that location	dN_{env}/ds

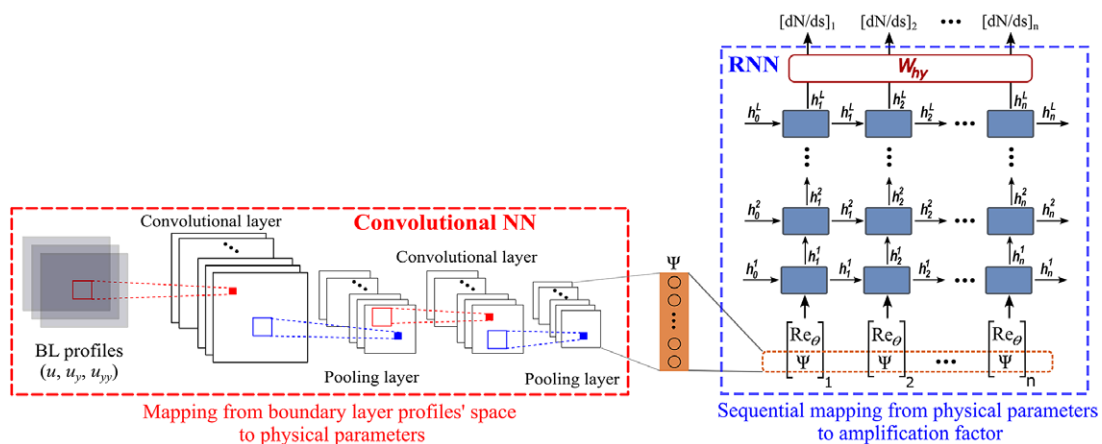


Figure 4. Proposed neural network for transition modeling. Convolutional neural network encodes the information from boundary-layer profiles (u, u_y, u_{yy}) into latent features (Ψ) at each station. RNN processes the input features $(Re_\theta$ and Ψ) in sequential manner to predict the growth rate (dN/ds) of the N-factor envelope.

concatenated with Re_θ at each station, which provides the sequential input features for the RNN architecture to predict the local growth rate of the most amplified instability mode, or equivalently, the slope (dN/ds) of the N-factor envelope at a sequence of stations along the airfoil surface. Each RNN cell (Figure 2) consists of nonlinear mappings from the input to the hidden state (W_{qh}) and from the hidden state to the output (W_{hy}), with each mapping involving two hidden layers with 72 neurons each. The rectified linear activation function (ReLU) is used to introduce nonlinearity in these layers. The hidden state is represented by a vector of length 9 and a linear layer is used for the mapping W_{hh} between the hidden states. The RNN architecture consists of three RNN layers stacked over each other, with corresponding three internal memories (hidden states), each representing varying level of dependency (short to long term) between the output at the current station and the input at the current as well as the preceding stations.

As the CNN architecture is intrinsically dependent on the shape of the training data, the architecture can be tuned to different shapes of training data and the proposed model is expected to maintain its efficiency and accuracy. Future work will explore the vector-cloud neural network Zhou et al. (2021) which can deal with any number of arbitrarily arranged grid points across the boundary layer profiles. Since empirical tuning of hyperparameters provided good results, we did not undertake an extensive optimization of the whole model. In a related unpublished work, more extensive hyperparameter optimization resulted in minor adjustments of the CNN architecture with comparable results.

3. Database of Linear Amplification Characteristics for Airfoil Boundary Layers

A large database of the linear stability characteristics of two-dimensional incompressible boundary-layer flows over a broad set of airfoils was generated for the training and evaluation of the proposed model. These boundary-layer flows can support the amplification of TS waves and the most amplified TS waves at any location along the airfoil correspond to two-dimensional disturbances (i.e., spanwise wavenumber $\beta = 0$). This database documents the amplification characteristics of unstable TS waves under the quasiparallel, no-curvature approximation. A value of $N_{tr} = 9$ has been empirically found to correlate with the onset of laminar-turbulent transition in benign freestream disturbance environments characteristic of external flows at flight altitudes. The airfoil contours were obtained from public domain sources, such as the UIUC Airfoil Coordinates Database (UIUC Applied Aerodynamics Group, 2020). Linear stability characteristics for laminar boundary-layer flows were computed using a combination of potential flow solutions (Drela, 1989) and a boundary-layer solver (Wie, 1992). The computational codes used are industry standard and have been used in number of research works over the years. Inviscid computations using panel method have been performed with 721 points around the airfoil. For boundary layer solver, 300–400 grid points have been considered in wall normal direction using a second order finite difference scheme. We note that the focus of this database is on transition due to TS waves in attached boundary layers, and therefore, flows involving a separation bubble (which cannot be computed with the viscous–inviscid-interactive procedure adopted herein) are not considered in the present work.

Airfoil contours included in the database belong to different categories and were selected randomly to cover a range of practical applications. These categories include different series of NACA airfoils, natural laminar flow airfoils, low Reynolds number airfoils, rotorcraft airfoils, and airfoil contours designed for transonic flows, and so on. Selected airfoils from three of these categories have been plotted in Figure 5, which illustrates the markedly different airfoil contours included in the database. Data corresponding to both upper and lower surfaces of the airfoils has been considered, except for the symmetrical airfoil sections for which the lower-surface data have been excluded to avoid duplication, and hence, a bias in the data sampling. For reference, all 53 airfoils from the database are listed as well as plotted in Appendix A. The range of chord Reynolds numbers (Re_c) included in the database extends over nearly four orders of magnitude $[5 \times 10^4, 2 \times 10^8]$ and a broad range of angles of attack (AOA) $[-6^\circ, 8^\circ]$ has been considered for each of these airfoils. However, some of the boundary layers within the above range of Reynolds numbers and angles of attack are either stable or only weakly unstable (i.e., corresponding to a rather small

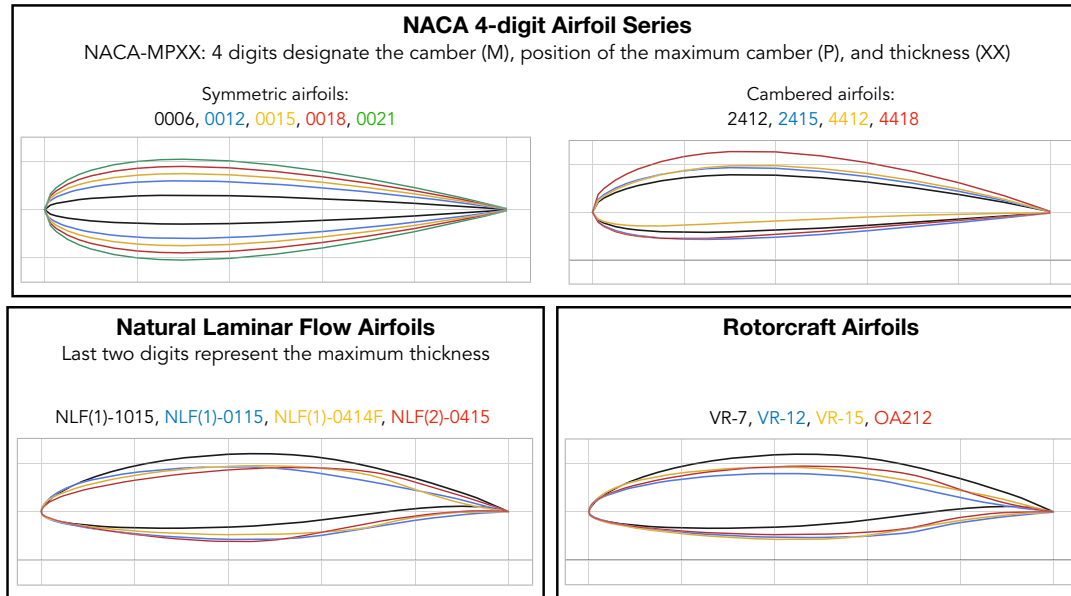


Figure 5. Airfoil sections for three airfoil families in the database. A complete list of airfoils along with their geometries is given in Appendix A.

peak N-factor, $N < 3$). Those flows were excluded from the database, still yielding a total of 31,247 flow cases corresponding to the 53 airfoils in this database. Although the computational cost of generating such a database is only few hours, the associated human effort measured in hours are significantly higher since the process requires manual interventions and expert judgements to avoid spurious modes in linear stability computations. Furthermore, preprocessing of geometrical data to ensure smooth surface curvature also added significant human effort cost to the database generation.

The database documents the mean flow features and the relevant linear stability characteristics in a sequential manner along 121 streamwise stations, starting from a station close to the onset of instability and extending up to either the point where the N-factor envelope reaches $N_{\text{env}} = 25$ or to the end of the chordwise domain (which can be upstream of the trailing edge if the boundary-layer solution terminates due to an incipient flow separation). We keep the sequence length fixed at 121 for all airfoils and flow conditions, which makes it more efficient in handling the data during the training and testing of the RNN model. The location of each station is defined in terms of the arc length along the airfoil surface (s). Besides the parameters given in Table 1, sequential information for several other relevant parameters has also been included in the database, such as the chordwise location of each station (x/c), local edge velocity (U_e), local boundary-layer edge density (ρ_e) and viscosity (μ_e), boundary-layer momentum thickness (θ), and so on.

The present work is aimed at developing an RNN model, which is trained over a subset of the complete dataset, and has the ability to predict the transition location for any boundary-layer flow from the complete dataset with reasonable accuracy. Computational constraints limit the size of the training dataset for RNNs, since they are more expensive to train as compared to simple feedforward neural networks. Reducing the size of the training data via subsampling of the overall database would require the sampling process to avoid any bias toward any specific subset of the database. Such bias can have a dominant effect on the efficacy of the loss function used for training, resulting in a potential overfitting across certain parts of the training data, and worse predictive performance for other subsets of the database. Hence, a large database of this type requires adequate sampling procedures for the selection of the training data so that the resulting model can provide a balanced representation of the entire database.

4. Results

The proposed RNN model predicts the sequence of growth rates of the most amplified disturbances, that is, the slope values of the N-factor envelope as a function of distance along the airfoil contour. The N-factor envelope can then be determined as the cumulative integral over this sequence, with the lower limit of integration corresponding to the airfoil location where the boundary-layer flow first becomes unstable (or, equivalently, the station across which the slope first changes in sign from a negative value to a positive one). The transition onset location can then be estimated as the location where the envelope reaches the critical N-factor determined via correlation with a relevant set of measurements. The sequential data are defined at a fixed number of stations for each flow, but the physical domain length can vary from case to case due to the potential onset of flow separation in a laminar boundary-layer computation. The loss function used for the training process includes a weighting function corresponding to the cell size (ds) in the vicinity of each station. Specifically, the loss function used for training the neural network is defined in terms of a weighted sum over the square of the local error:

$$\mathcal{L} = \sum_{j=1}^m (l_j) \text{ where } l_j = \sum_{i=1}^n \left((Y_i - \hat{Y}_i)^2 \cdot ds_i \right), \quad (3)$$

where m denotes the number of sequences in the dataset, n denotes the number of stations in a sequence, and Y and \hat{Y} represent the true and predicted values, respectively, corresponding to the output quantity. The weighting function serves to reduce the bias due to a nonuniform streamwise grid used in the boundary-layer calculation.

The primary performance indicator of the proposed model is the prediction of the chordwise location of the laminar-turbulent transition. However, besides its obvious dependence on the stability characteristics of the airfoil boundary layer, the transition location also depends on the disturbance environment via the correlating N-factor value, N_{tr} . The measured transition locations in a broad range of flows typically correlate with a finite band of N-factor values in the vicinity of $N_{tr} = 9$ under a benign disturbance environment, such as those encountered by external flows at typical flight altitudes. For this reason, the predictive performance has been assessed considering the flow cases for which N-factor envelope reaches values of upto $N_{env} = 13$. To help provide a meaningful assessment of the model accuracy across a broad range of flows and correlating N-factors, three different error metrics have been defined separately from the loss function, as indicated below.

$$E_{env} = 100 \times \frac{1}{\tilde{m}} \sum_{j=1}^{\tilde{m}} \left(\frac{\|N_{env} - \hat{N}_{env}\|_{ds}}{\|N_{env}\|_{ds}} \right)_j, \quad (4)$$

$$E_{tr} = 100 \times \frac{1}{\tilde{m}} \sum_{j=1}^{\tilde{m}} \left| \frac{x/c_{tr} - \tilde{x}/c_{tr}}{x/c_{tr}} \right|_j, \quad (5)$$

$$E_{x/c} = 100 \times \frac{1}{\tilde{m}} \sum_{j=1}^{\tilde{m}} |x/c_{tr} - \tilde{x}/c_{tr}|_j, \quad (6)$$

where \tilde{m} denotes the number of sequences in the dataset for which the N-factor envelope reaches values of upto $N_{env} = 13$. The first error metric (E_{env}) is based on the $L2$ norm to evaluate the accuracy of the predicted N-factor envelope (N_{env}), determined by integrating the predicted slope values dN/ds . To emphasize a finite band of N-factor values in the vicinity of $N_{tr} = 9$, only the range of $5 < N_{env} < 13$ has been considered for each flow case. The second error metric (E_{tr}) corresponds to the relative discrepancy between the true and predicted chordwise locations of transition onset for the case of $N_{tr} = 9$. The third error metric ($E_{x/c}$) relates to the absolute error in the predicted transition location, scaled by the airfoil chord length.

4.1. Demonstration of predictive performance

Selection of training data for the development of a general purpose model for the airfoil universe requires a balance between multiple requirements that may conflict with each other. It is clearly desirable for the size of the training data to be moderate enough to minimize the training cost. However, the training data must also be large enough in scope to represent the broad application space and must be designed to avoid an unfair bias toward specific subregions from the parameter space of latent features. Translating these requirements into a practical procedure is not a straightforward task. Given the availability of the large database of stability characteristics as described in the previous section, we have evaluated several different strategies for the selection of an appropriate subset of that database for the training process.

We begin using a smaller portion of the available database for training purposes. This baseline case is representative of less ambitious efforts at database generation, as well as being better suited for the case involving a broader application space that includes additional flow parameters such as, for instance, nonzero Mach numbers, nonzero surface transpiration, surface heating/cooling, and so on. The baseline training set consists of five out of the total 53 airfoils, with each of these five airfoils representing a different subgroup of airfoils from Table A1, namely, the NACA0012, ONERAM6, NACA2412, NACA63-415, and NLF(1)-0416 airfoils. These airfoils correspond to airfoil indices of 2, 44, 6, 15, and 25, respectively. Here, the first two airfoils have symmetric contours whereas the latter three correspond to asymmetrical airfoil sections. Table 2 lists the various flow conditions at which boundary-layer solutions are available for each airfoil in the database. To assess the RNN model for interpolation and extrapolation with respect to the angle of attack and chord Reynolds number, respectively, flow conditions marked in red have been included in the testing dataset, while the remaining flow cases constitute the training dataset. Such an arrangement of the total available cases from these five airfoils results in a 60–40% split between the training and testing data.

The sequential information corresponding to the evolution of mean boundary-layer profiles along the airfoil surface has been documented in the database, with a uniform sequence length of 121 stations for all flow cases. An initial assessment was conducted to ascertain the effect of different sequence lengths. The results of this assessment are shown in Figure 6, wherein the error metrics defined in Equations (4) and (5) have been plotted for different sequence lengths. Significant improvements can be observed by reducing the sequence length from 121 to 60, the reason for which is not entirely clear. It could be one of the areas for future investigations of such models. In general, however, this trend may be related to the fact that more information sometimes lead to diluting of information and, consequently, to worse results. Further shortening of the sequence length yields relatively little benefit in terms of model accuracy for either training or test data. In fact, the accuracy of predicting the transition location worsens when the sequence length is reduced beyond 60. This observation is likely to be related to a poorer resolution of the shape of the N-factor envelope achieved when fewer stations are used across the same physical domain length along the airfoil surface. Looking at the similar trends for both error metrics, it was decided that a sequence length of 60 stations would provide an optimal choice for all of the results to be presented in this paper. Figure 6 shows that the predictive performance for the testing dataset is comparable to that for the training

Table 2. Flow conditions for all the cases in the airfoil database.

Angles of attack (deg)	Reynolds numbers
$-6^\circ, -5^\circ, -4.5^\circ, -4^\circ, -3.5^\circ, -3^\circ,$ $-2.5^\circ, -2^\circ, -1.5^\circ, -1^\circ, -0.5^\circ, 0^\circ,$ $0.5^\circ, 1^\circ, 1.5^\circ, 2^\circ, 2.5^\circ, 3^\circ, 3.5^\circ, 4^\circ,$ $5^\circ, 6^\circ, 7^\circ, 8^\circ$	$3.5 \times 10^4, 5.0 \times 10^4, 7.0 \times 10^4, 1.0 \times 10^5, 1.4 \times 10^5,$ $2.8 \times 10^5, 4.0 \times 10^5, 5.6 \times 10^5, 8.0 \times 10^5, 1.1 \times 10^6,$ $1.6 \times 10^6, 2.3 \times 10^6, 3.2 \times 10^6, 4.5 \times 10^6, 6.4 \times 10^6,$ $9.0 \times 10^6, 1.3 \times 10^7, 1.8 \times 10^7, 3.6 \times 10^7, 5.1 \times 10^7,$ $7.2 \times 10^7, 1.0 \times 10^8, 1.4 \times 10^8$

For evaluation of the RNN model, flow conditions used for model testing are marked in red color whereas the flow conditions used for training are indicated in black color.

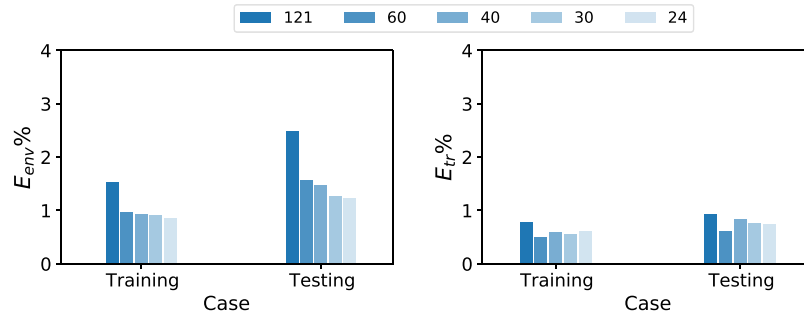


Figure 6. Comparison of prediction error percentage for training and testing datasets with different sequence lengths. Training and testing datasets have been defined based on flow conditions as given in Table 2.

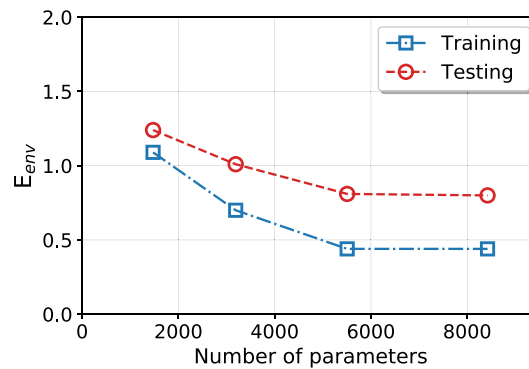


Figure 7. Training and testing errors for a range of sizes of the recurrent neural network (RNN) model indicated by the number of learnable parameters. The number of layers is kept the same while the parameters are varied proportionately in all three mappings, W_{hh} , W_{qh} , and W_{hy} .

dataset, demonstrating the interpolating and extrapolating capability of the proposed RNN model with respect to both AOA and Re_c .

Next, we assess the effect of the size of the RNN model on the prediction error. Figure 7 displays the variation in error percentage as a function of the number of learnable parameters in the RNN model. While the error metric E_{env} decreases as the number of learnable parameters is increased up to 5,500, the error remains nearly constant with a further increase in the number of parameters. One may deduce from these results that an RNN model with 5,500 learnable parameters provides near-optimal learning capability without causing overfitting. Consequently, this model size will be maintained for all the results presented for the current dataset. We note that the training dataset for this baseline case is of much smaller size in comparison to the complete airfoil database and that the use of a larger training dataset will most likely require a larger number of learnable parameters to enhance the learning capacity of the RNN model. Thus, the selection of model size will be discussed again when we work with somewhat larger training datasets in the following subsections.

The architecture of the proposed neural network model has direct correlation with the underlying physics of flow transition. Previously proposed fully connected neural network-based transition models (Fuller et al., 1997; Crouch et al., 2002) did not distinguish the evolution of flow in wall-normal and streamwise directions. At any station along airfoil surface, propagation of boundary layer flow in wall-normal direction is instantaneous (analogous to elliptic behavior of diffusion equation). Hence, Zafar et al.

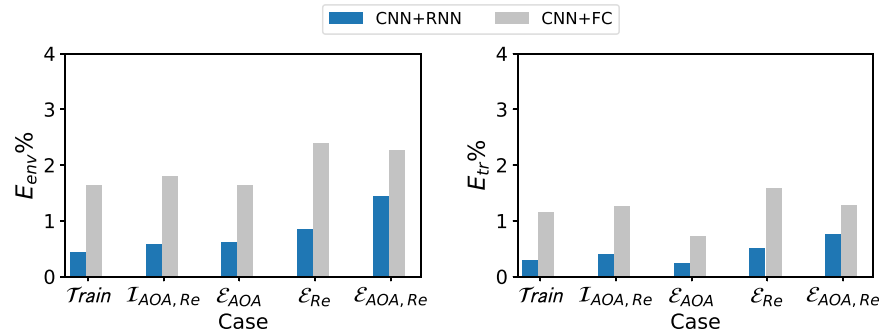


Figure 8. Comparison of results for the recurrent neural network and fully connected network. Testing cases have been subcategorized as: Interpolation with respect to both angles of attack (AOA) and Reynolds number (Re) ($I_{AOA, Re}$), Extrapolation with respect to AOA (E_{AOA}), Extrapolation with respect to Re (E_{Re}), Extrapolation with respect to both AOA and Re ($E_{AOA, Re}$).

(2020) used CNN to encode the information from the boundary layer profiles in wall-normal direction in a vector of latent features. Such treatment of boundary layer profiles provides a stronger correlation with the underlying physics of the flow and also allows the application of CNN-based transition model to various instability mechanisms (Paredes et al., 2020). These characteristics were clearly lacking in previously proposed neural network-based transition models.

The current work uses both CNN and RNN in tandem where the RNN has been used to encapsulate the underlying physics of streamwise evolution of flow instability along with CNN which process the boundary layer profiles in wall-normal direction. Along the streamwise direction, elliptic behavior has already been taken care of in the stability theory however hyperbolic nature of the instability amplification from upstream to downstream requires sequence-to-sequence modeling, for which RNN has been used. To assess the benefit of sequence transduction in the RNN model, its predictive performance has been compared with that of a previously proposed model which does not account for the sequence information among the inputs at different stations (Zafar et al., 2020). For a fair comparison, almost equal number of learnable parameters was used for both networks. The comparison is presented in Figure 8, wherein we include the error percentages for both training and testing datasets. The testing cases have been further categorized in terms of interpolation and extrapolation with respect to the flow conditions. The results corresponding to each dataset show a clearly superior predictive performance for the RNN model vis-a-vis the fully connected neural network.

We also note that, in comparison to transition model based on fully connected neural network, the RNN model has a moderately higher training cost associated to it. For the given training data (comprising of 5 airfoil datasets with sequence length of 60) and a given model size ($\sim 5,500$ learnable parameters), the RNN model took 29 GPU hours to train as compared to 19 GPU hours for the fully connected neural network. A single NVIDIA V100 GPU was used for training purposes.

In comparison to direct stability computations, the RNN model can estimate the transition location three to four orders of magnitude faster. Actual times for direct computations of linear stability can vary depending on various factors, including the specific flow and the level of resolution (in terms of the number of stations in a sequence and the number of frequencies used to determine the N-factor envelope). In that regard, the above estimate is believed to provide a reasonable, if somewhat conservative, estimate of the speed-up due to RNN.

4.2. Evaluation of predictive performance for complete database

We now perform a comparative assessment of the accuracy of the RNN models based on different selections of training datasets. These training datasets have been summarized in the Table 3. The rationale

Table 3. Summary of training dataset cases.

Index	Label	Training dataset	Number of flow cases
I	Five airfoils	NACA-0012, NACA-2412, NACA-63-415, NLF(1)-0416, and ONERA-M6	2,624
II	Random augmentation	Case I + 100 random flow cases from each of the other airfoils	7,026
III	Augmented airfoils set	Case I + Five more airfoils with largest mean error (LRN(1)-1007, NACA-6712, NLF(1)-1015, NLF(2)-0415, and CLARK-Y)	4,455
IV	Error-based augmentation	Case I + Specific flow cases of other airfoils with $E_{\text{env}}\% > 3$ in Case I	5,024
V-A	Random selection (%)	Randomly selected 20% flow cases of each airfoil	6,233
V-B	Random selection (#)	Randomly selected 100 flow cases of each airfoil	5,300

Flow cases corresponding to the mentioned airfoils are included in the training dataset.

behind the selection of each training dataset from Table 3 will be outlined in the course of the discussion of the results, especially as the results for Case I provide the baseline for the selection of training data for the subsequent cases. Since the analysis is focused on the performance of the RNN model in predicting the transition location over an arbitrary airfoil, no distinction has been made between the airfoils and flow conditions used for training and testing, and all the flow cases included in the training dataset are considered alongside the cases that were not used during training. Because the sizes of the training sets from Table 3 are comparable to each other and always less than 24% of the total available data, the error metric based on the entire database was deemed to be a meaningful measure of the model's predictive accuracy.

Case I (denoted as “Five Airfoils”) from Table 3 involves a training dataset that is comprised of the same five airfoils that were used in the earlier assessment of the RNN model size, comparison with fully connected network, and so on. Results for the RNN model trained using the Case I dataset are shown in Figures 9 and 10. Figure 9 presents the mean error percentages for the predicted N-factor envelope and transition location corresponding to all airfoils from the overall database. The figures have been shaded to distinguish between the different groups of airfoils belonging to the airfoil families included within the overall database. Airfoil names corresponding to the indices from Figure 9 are given in Table A1. In general, the mean error percentage for most of the airfoils is below about 3%, which demonstrates the general capability of the RNN model based on the Case I training data. Even though the model has been trained with a significantly smaller subsample of airfoils from the overall database, it is still able to predict the N_{env} and transition location for the entire set of airfoils including different categories with a reasonable accuracy. Laminar to turbulent transition due to TS amplification within the attached flow region is achieved at varying numbers of flow conditions across the different airfoils, and the markers for each airfoil in the figure have been colored on the basis of the dataset size of that airfoil. This feature will be used to gain additional insights during the subsequent discussion as we describe the results for the remaining cases from Table A1.

Predictive performance as a function of the angle of attack and chord Reynolds number is shown in Figure 10, which indicates the distribution of error percentage across the overall database via a color map for the kernel density estimate. In addition, 1% of randomly sampled data points from the overall database have also been included as green dots within the figure. No bias in predictive errors toward specific flow conditions may be observed within the figure, indicating that the model is able to yield comparable

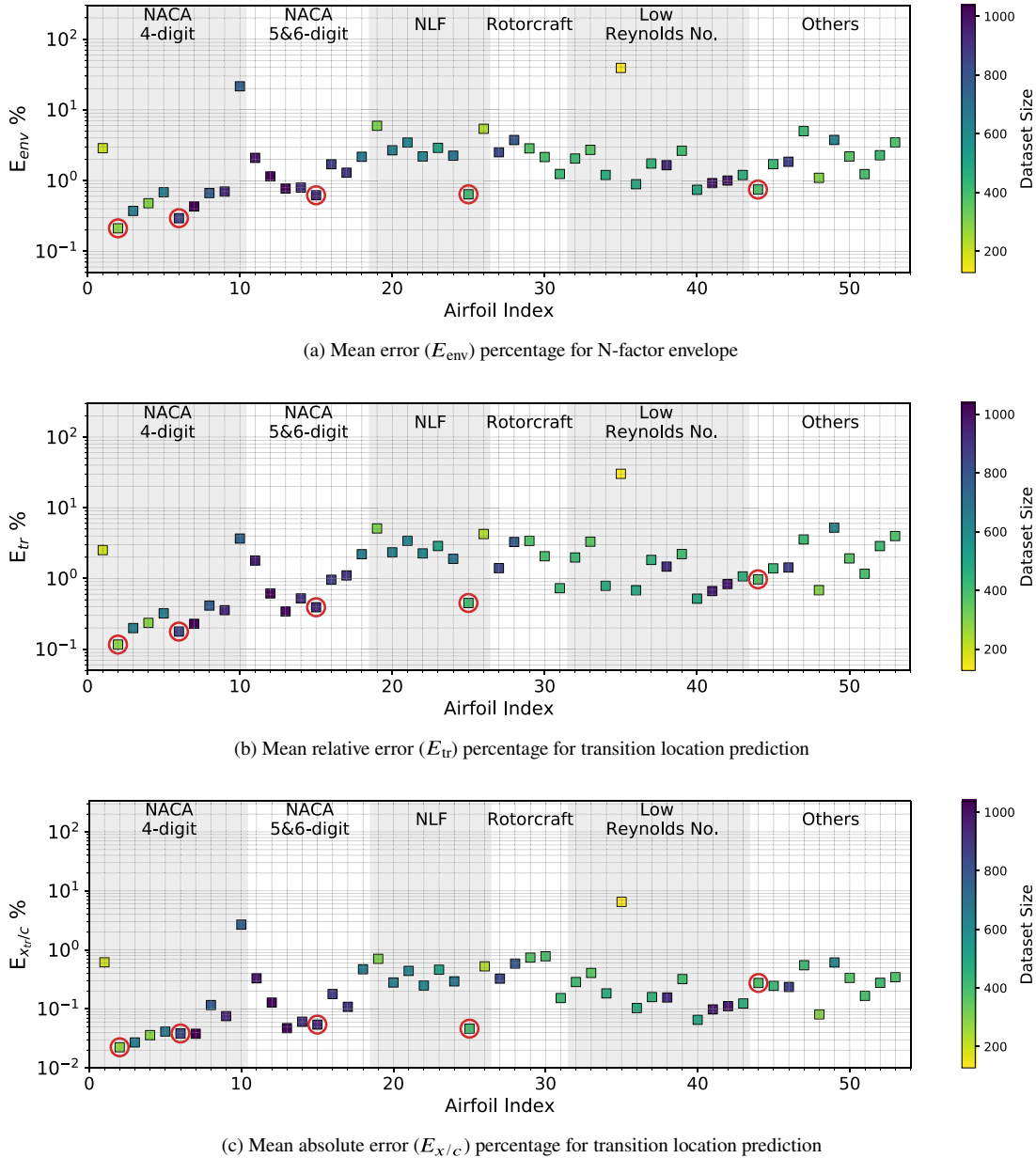


Figure 9. Mean error percentage for each airfoil in the database, corresponding to training dataset of Case I (five airfoils) given in Table 3. Airfoils corresponding to training dataset have been encircled in red color. Markers' color represent the dataset size (number of flow cases) of each airfoil in the database.

accuracy across the entire range of flow conditions. The transition locations for most of the flow cases is predicted with a relative error percentage of (E_{tr}) < 2%, as shown by the higher density region with a darker color in the color map from Figure 10.

Results for Case I, shown in Figure 9, indicate a few outlying airfoils corresponding to a high error percentage in the predictions of the RNN model. In particular, the average error in the prediction of the N-factor envelope for the LRN(1)-1007 airfoil is significantly higher ($E_{env} > 30\%$) as compared to that for most of the other airfoils. LRN(1)-1007, designed for high lift and low drag at $Re = [5 \times 10^4, 1.5 \times 10^5]$,

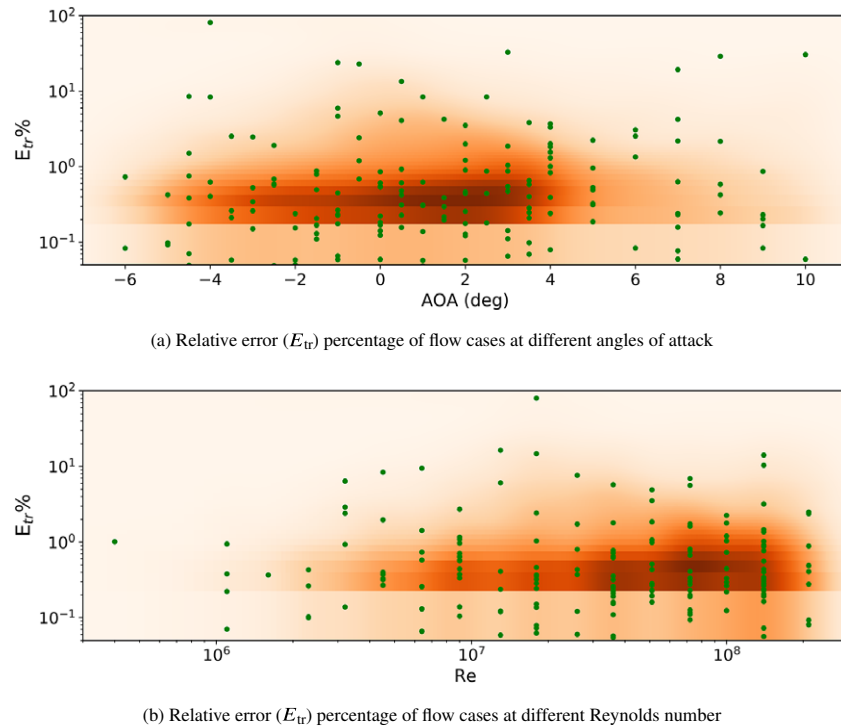


Figure 10. Relative error (E_{tr}) percentage for all flow cases, corresponding to training dataset of Case I (five airfoils) given in Table 3. Green markers (filled circles) show only 1% of the randomly sampled flow cases. The contour shows the kernel density estimated from all the flow cases. Darker region indicates higher probability density. The horizontal lines appearing in the contour plots such as that near an error of 0.2% are due to technical reason (bins have been defined in linear-scale while the vertical axis of the plot is in log-scale) and do not depict any real discontinuity.

has a peculiar airfoil contour and aerodynamic behavior, which is markedly different with respect to the other airfoils in the training dataset. This may explain why the predictive error percentage for the LRN(1)-1007 airfoil is higher by almost an order of magnitude. Similarly, the NACA 6712 airfoil with a highly aft-cambered airfoil section also has a significantly higher error percentage ($E_{env} > 20\%$). In comparison, the predictive performance for the other NACA airfoils is reasonably good, with the average absolute error in predicting the transition location below 1% for most of those airfoils.

Augmenting the training dataset from Case I with additional data provides the most obvious way of improving on the predictive performance of the above RNN model. Several different strategies for data augmentation were evaluated in the course of this work, and they are denoted as Cases II, III, and IV in Table A1. For Case II, 100 randomly selected flow cases from every other airfoil have been added to the training dataset from Case I. Even though these data augmentation causes the size of the training dataset to increase almost threefold with respect to that in Case I, the inclusion of flow cases for every airfoil within the training dataset leads to significantly improved predictive performance of the RNN model. The results for Case II are shown in Figure 11. The overall prediction error percentages have decreased significantly in comparison with Case I, and the maximum absolute error in predicting the transition location ($E_{x/c}$) for any airfoil has reduced from 6.5% in Case I to approximately 1% in Case II.

For Case III, the training dataset has been augmented by including an additional set of airfoils for which the average error (E_{env}) is greater than 3%. Five such airfoils, mentioned in Table 3, along with the original five airfoils from Case I constitute the training dataset for Case III. Results for this training case are plotted in Figure 12. The figure shows that, despite a larger size of the training dataset with respect to that in

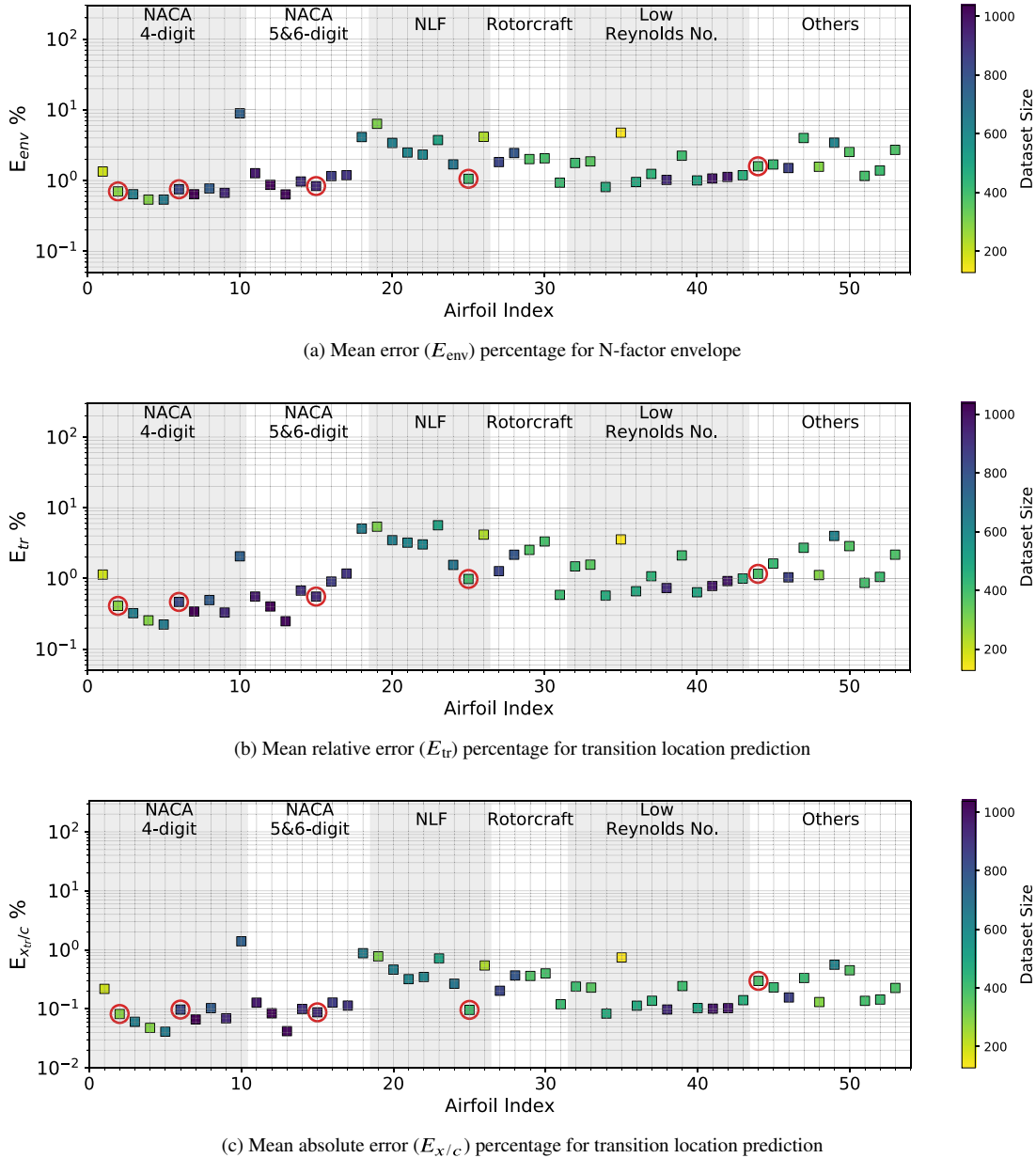


Figure 11. Mean error percentage for each airfoil in the database, corresponding to training dataset of Case II (random augmentation) given in Table 3. Airfoils corresponding to training dataset have been encircled in red color. Markers' color represent the dataset size (number of flow cases) of each airfoil in the database.

Case I, the predictive performance for Case III has worsened. The overall trend can be summarized by looking at the group of natural laminar flow airfoils in Figure 12, for which the model predictions are now significantly worse ($7\% < E_{tr} < 13\%$), except for those airfoils that have been included within the training dataset ($0.7\% < E_{tr} < 2\%$). This observation points to a possible overfitting of the data by the RNN model under consideration. Hence, one may conclude that the augmentation of the original set by five additional airfoils does not provide a good representation of the overall database.

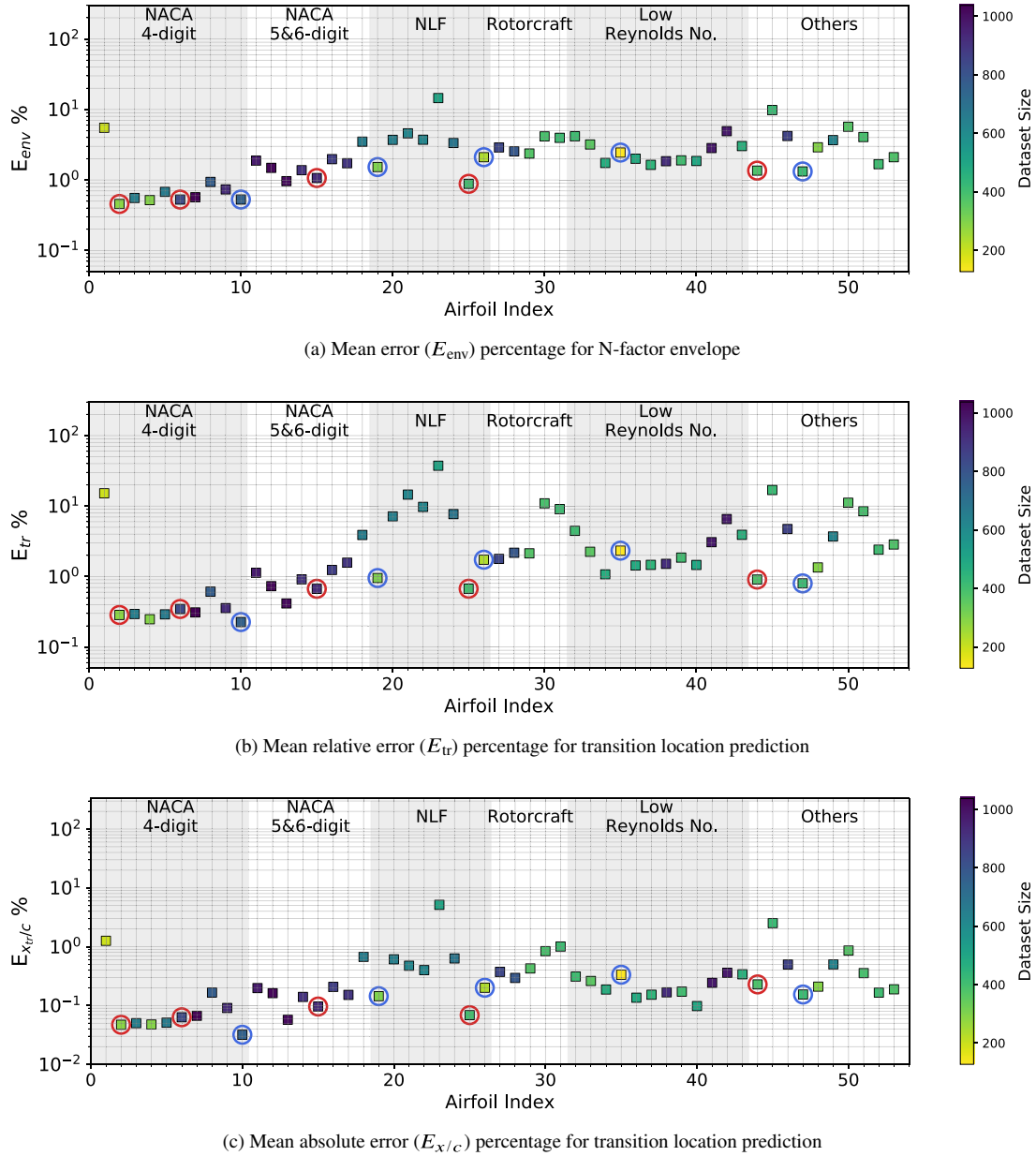


Figure 12. Mean error percentage for each airfoil in the database, corresponding to training dataset of Case III (augmented airfoils set) given in Table 3. Airfoils corresponding to training dataset have been encircled, where airfoils already in the training dataset from Case I have been encircled in red color; while the augmented set of airfoils have been encircled in blue color. Markers' color represent the dataset size (number of flow cases) of each airfoil in the database.

For Case IV, the training dataset from Case I has been augmented with the flow cases from overall database that correspond to the highest predictive error percentage ($E_{env} > 3\%$) associated with the RNN model from Case I. A significant improvement can be observed in the overall predictive performance of the RNN model as compared to all of the previous cases. Based on this error-based data augmentation, the absolute error percentage in the predicted transition location ($E_{x/c}$) over any airfoil from the database is

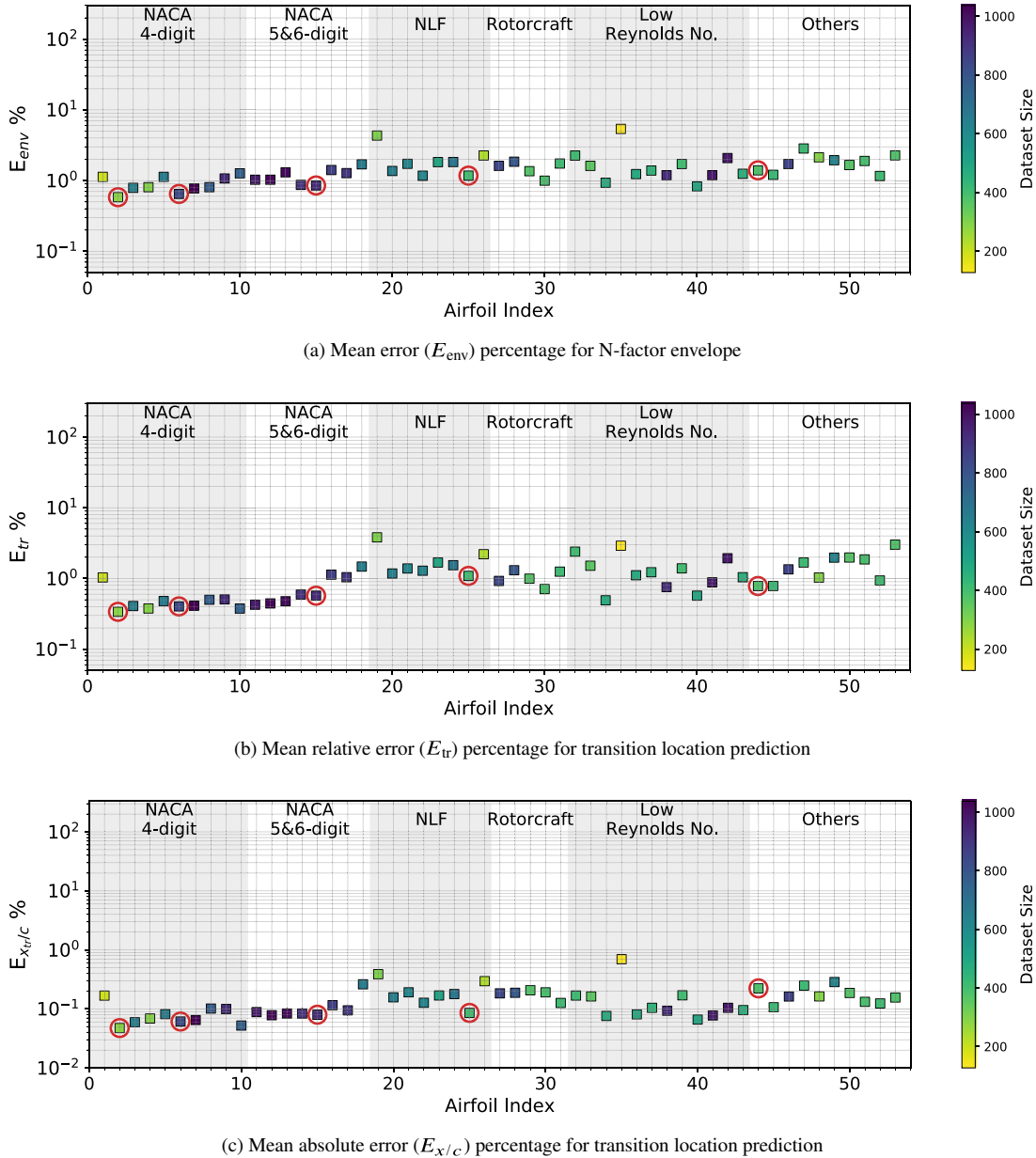


Figure 13. Mean error percentage for each airfoil in the database, corresponding to training dataset of Case IV (error-based augmentation) given in Table 3. Airfoils corresponding to training dataset have been encircled in red color. Markers' color represent the dataset size (number of flow cases) of each airfoil in the database.

less than approximately 0.7%. Figure 14 shows the distribution of the error percentage as a function of the flow conditions. Due to the kind of data augmentation used for this case, one finds that most flow cases tend toward smaller error values in comparison to those in Case I (Figure 10).

The selection of training data for the Cases II, III, and IV was based on the results of Case I, which consisted of five airfoils chosen somewhat arbitrarily (except for the attempt to include some representation from five different groups of airfoils). Although Case IV provides quite good results, such that the

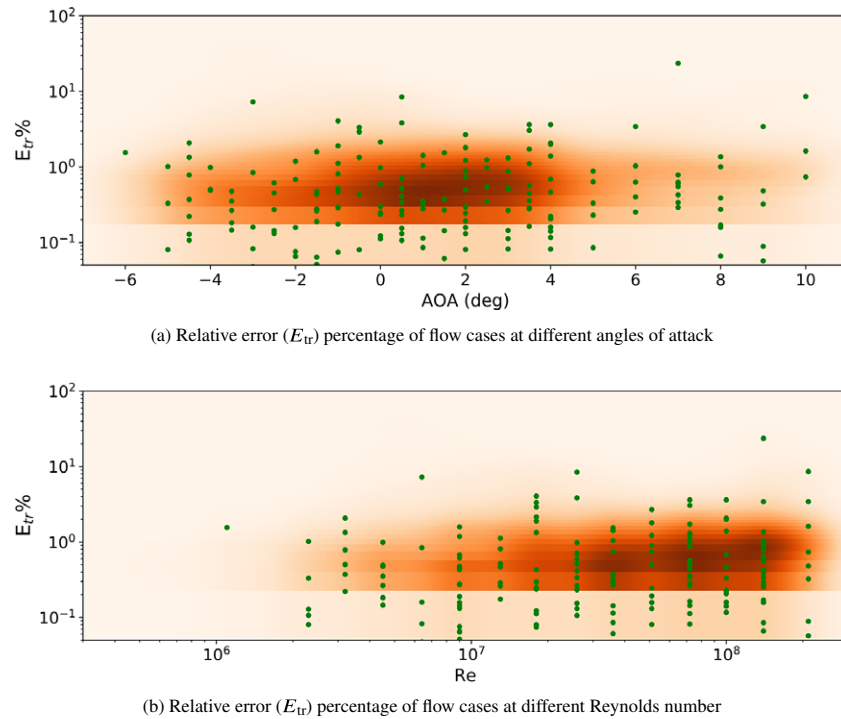


Figure 14. Relative error (E_{tr}) percentage for all flow cases, corresponding to training dataset of Case IV given in Table 3. Green markers (filled circles) show only 1% of the randomly sampled flow cases. The contour shows the kernel density estimated from all the flow cases. Darker region indicates higher probability density. The horizontal lines appearing in the contour plots such as that near an error of 0.2% are due to technical reason (bins have been defined in linear-scale while the vertical axis of the plot is in log-scale) and do not depict any real discontinuity.

absolute error percentage associated with the prediction of the transition location ($E_{x/c}$) over any airfoil is 0.7% or less, the selection of the training dataset has been made in an indirect manner on the basis of the results obtained in Case I.

A more direct strategy for subsampling a training dataset from the entire database has been assessed in Case V, where a completely random subset of varying magnitudes has been selected from the overall database. Two subcases (V-A and V-B) have been assessed in this regard, as summarized in Table 3. Case V-A involves the selection of a fixed percentage of flow cases corresponding to each airfoil, which results in a different number of flow cases for each airfoil in the training dataset. As mentioned earlier, this uneven number of flow cases in the database results naturally from the fact that different airfoils achieve laminar to turbulent transition at different flow conditions and the fact that only upper surface boundary layers are retained in the case of airfoils with symmetric contours. For Case V-B, a fixed number of flow cases corresponding to each airfoil have been selected as training data in order to provide a uniform weighting to each of the airfoils. With this arrangement to define the two subcases, different sizes of training datasets have been used to analyze the variation of error percentage with respect to the size of the training dataset. The results of this study are shown in Figure 15. The figure shows that, in both cases (V-A and V-B), there is an optimal size of the training dataset that leads to a minimum prediction error. For Case V-A, a training dataset size of $\sim 6,200$ (20% flow cases from each airfoil) provides the best predictive performance. Similarly, for Case V-B, a training dataset size of $\sim 5,300$ (100 flow cases of each airfoil) provides the best predictive performance.

Comparing the results for both subcases in Figure 15 shows that Case V-B provides a better prediction accuracy, which can be explained based on the error percentages of three airfoils, NLF(1)-1015,

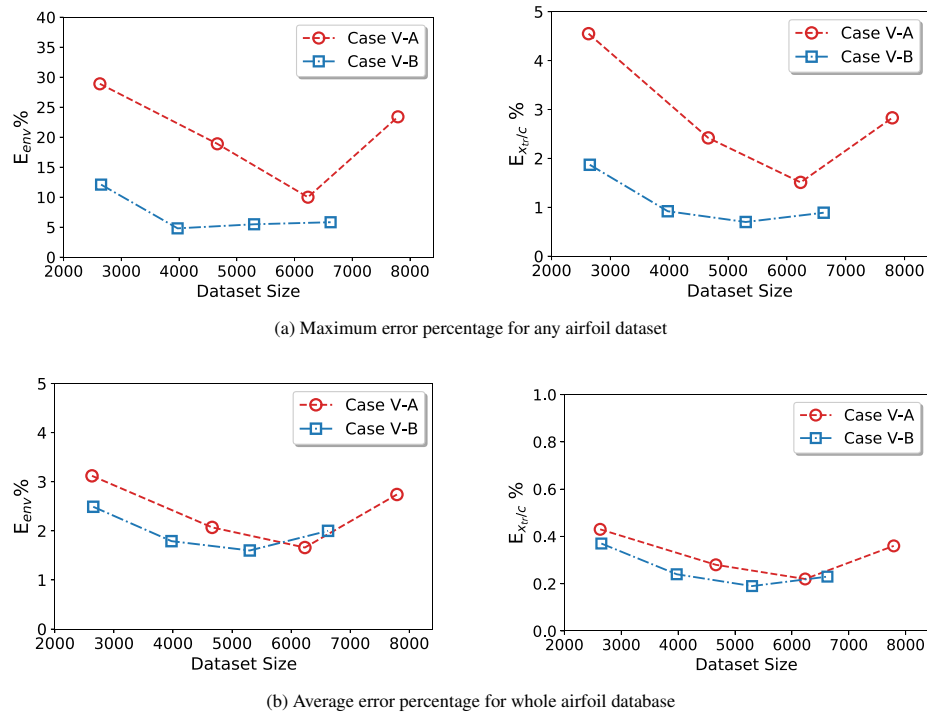


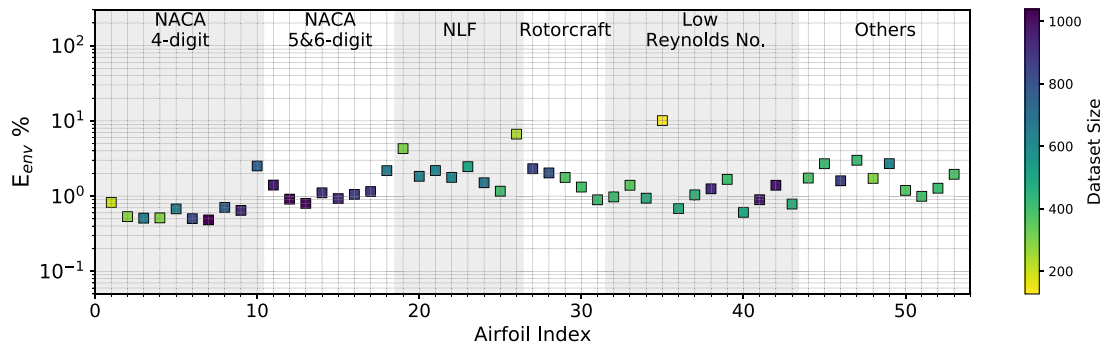
Figure 15. Case V: Variation of error percentage with respect to training dataset size (number of flow cases).

NLF(2)-0415, and LRN(1)-1007. These three airfoils with the highest percentage of error in Figure 16a correspond to a relatively smaller number of flow cases in the database, as seen from the colors of their respective markers. Because the training set in Case V-A includes a fixed percentage of flow cases for each airfoil, the above three airfoils remain relatively underweight with respect to the other airfoils with a larger number of flow cases. On the other hand, using a fixed number of flow cases for each airfoil provides a more balanced representation of the various airfoils within the training dataset, which results in a better overall predictive performance.

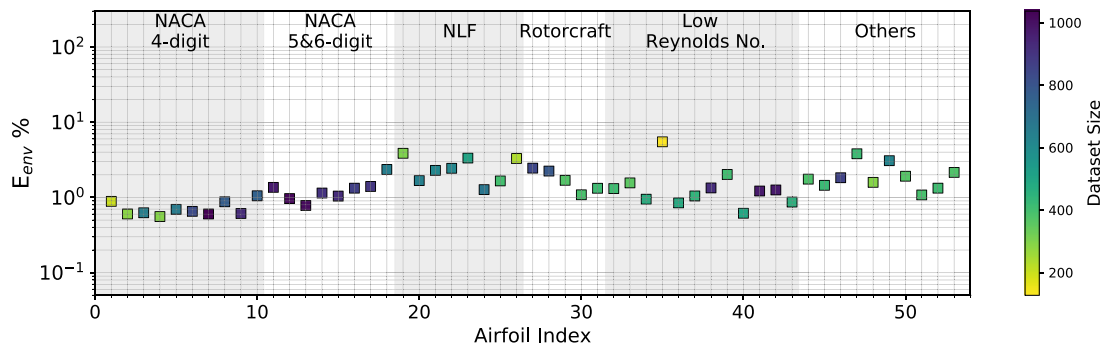
Results for all of the cases discussed in this section are summarized in Table 4. It is interesting to note that the results of Case V-B, which provides a more direct approach for selecting the training dataset, are very comparable in terms of prediction accuracy with the results from Case IV, which uses an indirect approach to select the training dataset and provides the best results among all of the cases discussed herein. Moreover, the training dataset for both of these cases is of almost equal size. Hence, Case V-B provides a direct and convenient approach for selecting a subset from a large database as the training data, while also yielding a good predictive performance over the entire database. Sample plots of the N-factor envelope for arbitrary combinations of airfoil contours and flow conditions are shown in Figure 17. These plots illustrate a qualitative comparison of the N-factor predictions based on the different training cases. One may clearly see that the predictions for certain flows in Cases IV and V-B are accurate even if the corresponding predictions for Cases I-III include significantly larger error.

4.3. Working with limited database

The database of airfoil flows generated during the present effort is relatively extensive in comparison to what may be generally available in a majority of practical situations. For this reason, assessments have been made to understand the predictive performance of the RNN model in selected possible scenarios.



(a) Case V-A with randomly selected 20 percent flow cases from each airfoil



(b) Case V-B with randomly selected 100 flow cases from each airfoil

Figure 16. Comparison of mean error (E_{env}) percentage for N -factor envelopes in Case V-A and V-B. Markers' color represent the dataset size (number of flow cases) of each airfoil in the database.

Table 4. Results for different training dataset cases.

Index	Label	Number of flow cases	Maximum error			Average error		
			E_{env} (%)	E_{tr} (%)	$E_{x/c}$ (%)	E_{env} (%)	E_{tr} (%)	$E_{x/c}$ (%)
I	Five airfoils	2,624	39.2	30.2	6.52	2.95	2.24	0.42
II	Random augmentation	7,026	8.95	5.66	1.40	1.92	1.64	0.26
III	Augmented airfoils set	4,455	14.5	37.4	5.15	2.71	4.14	0.43
IV	Error-based augmentation	5,024	5.38	3.82	0.70	1.53	1.17	0.15
V-A	Random selection (%)	6,233	10.1	35.0	1.51	1.66	2.04	0.22
V-B	Random selection (#)	5,300	5.49	4.22	0.70	1.60	1.38	0.19

Such assessments are based on interpolation and extrapolation with respect to the airfoil contours, under the assumption that a relatively smaller training dataset based on just a few NACA four-digit series airfoils is available. Table 5 outlines a summary of these cases and the corresponding results. Cases VI and VII provide an assessment with respect to the interpolation of airfoil contours within a family of selected symmetric and asymmetric airfoils, respectively. The resulting predictions are seen to be reasonably accurate for the testing dataset with average error of 0.12 and 0.04% of the chord length for both

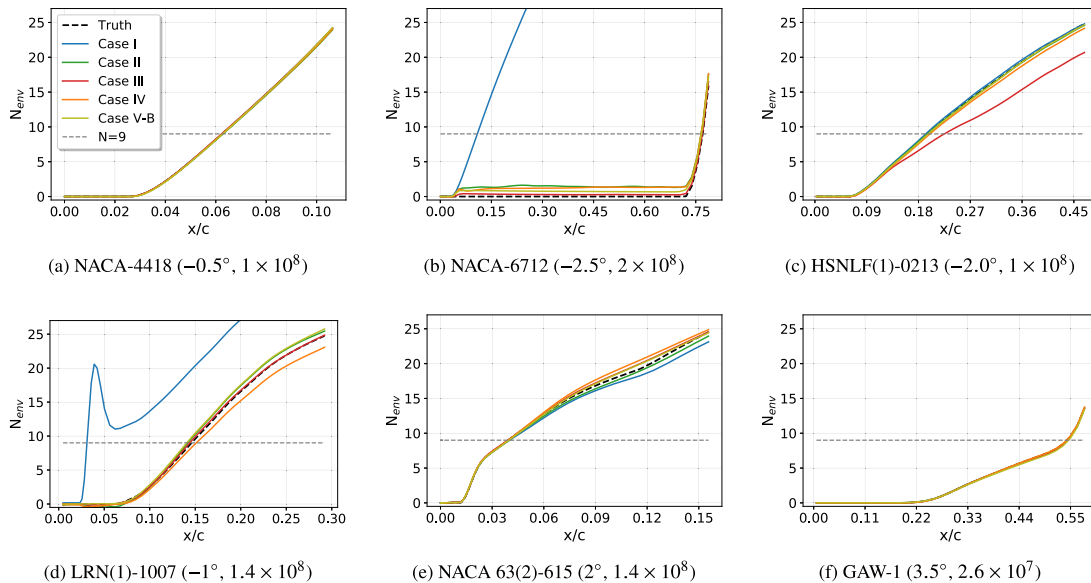


Figure 17. *N-factor envelope plots for arbitrarily chosen flow cases to illustrate the comparison of prediction by different training cases. Corresponding airfoil name and flow conditions (AOA, Re_c) have been mentioned for each plot.*

Table 5. *Assessment cases for different training cases based on NACA four-digit series airfoils.*

Index	Training dataset	Testing dataset	Testing error %		
			E_{env} (%)	E_{tr} (%)	$E_{x/c}$ (%)
VI	NACA-0006, NACA-0018	NACA-0012	0.97	0.55	0.12
VII	NACA-2412, NACA-4412	NACA 2415	0.34	0.17	0.04
VIII	NACA-0006, NACA-0012	NACA-0018	0.56	0.24	0.06

symmetric and asymmetric airfoils, respectively. Similarly, Case VIII targets the evaluation of model performance with respect to the extrapolation of the airfoil thickness, and again, the predictions for the testing dataset are found to be reasonably accurate with average error of 0.06% of the chord length for the given airfoil. Hence, it appears that the RNN model is able to predict well for previously unknown airfoil sections within the same family, regardless of whether the test data involves an interpolation within the distribution of the training data or an extrapolation beyond its boundaries. These findings support the selection strategy underlying Case I, which included five airfoils representing multiple groups from the overall database.

These cases have been studied to understand the model performance when a limited training dataset is available.

Assessment in Case IX involves a training dataset of five NACA four-digit series airfoils, and the predictive performance is evaluated using a testing dataset based on the rest of the airfoils. Results for this case have been shown in Table 6, where it can be observed that the predictive performance in this case is far worse in comparison to Case I, where five airfoils were taken from a different family of airfoils. Hence, a model trained using just a single family of airfoils does not extrapolate well to the other families of airfoils. Results for Case IX have also been shown in Figure 18, wherein the mean error percentages for the remaining families of airfoils are seen to be an order of magnitude higher than the error magnitudes

Table 6. Results for a training dataset comprised of a single family of airfoils and a testing dataset comprised of the rest of the airfoils in the database.

Index	Training dataset	Number of flow cases	Maximum error			Average error		
			E_{env} (%)	E_{tr} (%)	$E_{x/c}$ (%)	E_{env} (%)	E_{tr} (%)	$E_{x/c}$ (%)
IX	NACA-0006, NACA-0018, NACA-2412, NACA-4412, NACA-6712	2,841	67.5	44.8	9.51	9.55	9.43	1.52

associated with airfoils included in the training dataset. This finding reinforces the method adopted in Case I, namely, that a balanced training dataset should contain representation from different families of airfoils to achieve reasonably accurate predictive performance for the overall database.

5. Conclusion

A sequence-to-sequence modeling approach based on a recurrent neural network has been proposed to predict the location of laminar-turbulent transition via linear amplification characteristics of hydrodynamic instabilities in boundary-layer flows. This approach provides an end-to-end transition model, which maps the sequence of mean boundary-layer profiles to corresponding growth rates along the N-factor envelope, and then, to the estimated transition location. In this regard, a large database comprised of the linear growth characteristics of over 33,000 boundary-layer flows over 53 airfoils from a disparate range of applications has been used to train and test the proposed model. The results demonstrate that the RNN model is able to predict the transition location at various test flow conditions and for the entire range of airfoil contours with good accuracy (average error of less than 0.70% of the chord length for any given airfoil) despite being trained by a small subsample (about 16%) of the complete database. To the best of our knowledge, the database used herein is one of the largest of its kind, presumably representing a significant cross-section of the airfoil universe. To assess the techniques to facilitate the selection of representative yet computationally efficient training data, several alternate strategies have been investigated to provide insights into working with large amounts of data. The more easily realizable training set based on a small group of five airfoils (one each from five different groups of airfoil contours) forms the baseline for the selection of training data. A limited database of this type is found to result in substantial errors in transition prediction for a number of other airfoils, with average errors in transition location prediction across multiple test conditions for a single airfoil approaching as much as 6.52% of the chord length for the given airfoil. Data augmentation with additional cases from other airfoils that correspond to worst prediction errors from the baseline model is found to provide the best choice for improving the predictive performance of the RNN model, reducing the average error across all flow conditions in predicting transition location to 0.7% of the chord length for any airfoil. An alternate strategy of using a training dataset consisting of an equally weighted representation of each airfoil was also evaluated and was found to provide equally good predictive performance. Further assessments also showed that the RNN-based model is able to extrapolate/interpolate well within a family of similar airfoils and the predictive performance worsened while extrapolating the predictions to airfoil from other families.

Transition estimates based on the RNN model are easily three to four orders of magnitude faster than those based on direct stability computations. However, the main benefit of the deep learning models is an improved robustness of the prediction process, making it easier for nonexperts in laminar-turbulent transition to perform such computations. On the other hand, the deep learning models are restricted in their generalizability and this paper has addressed some of the issues related to the development of models that cover a broad space of flows. We believe the two types of models to be complementary in nature.

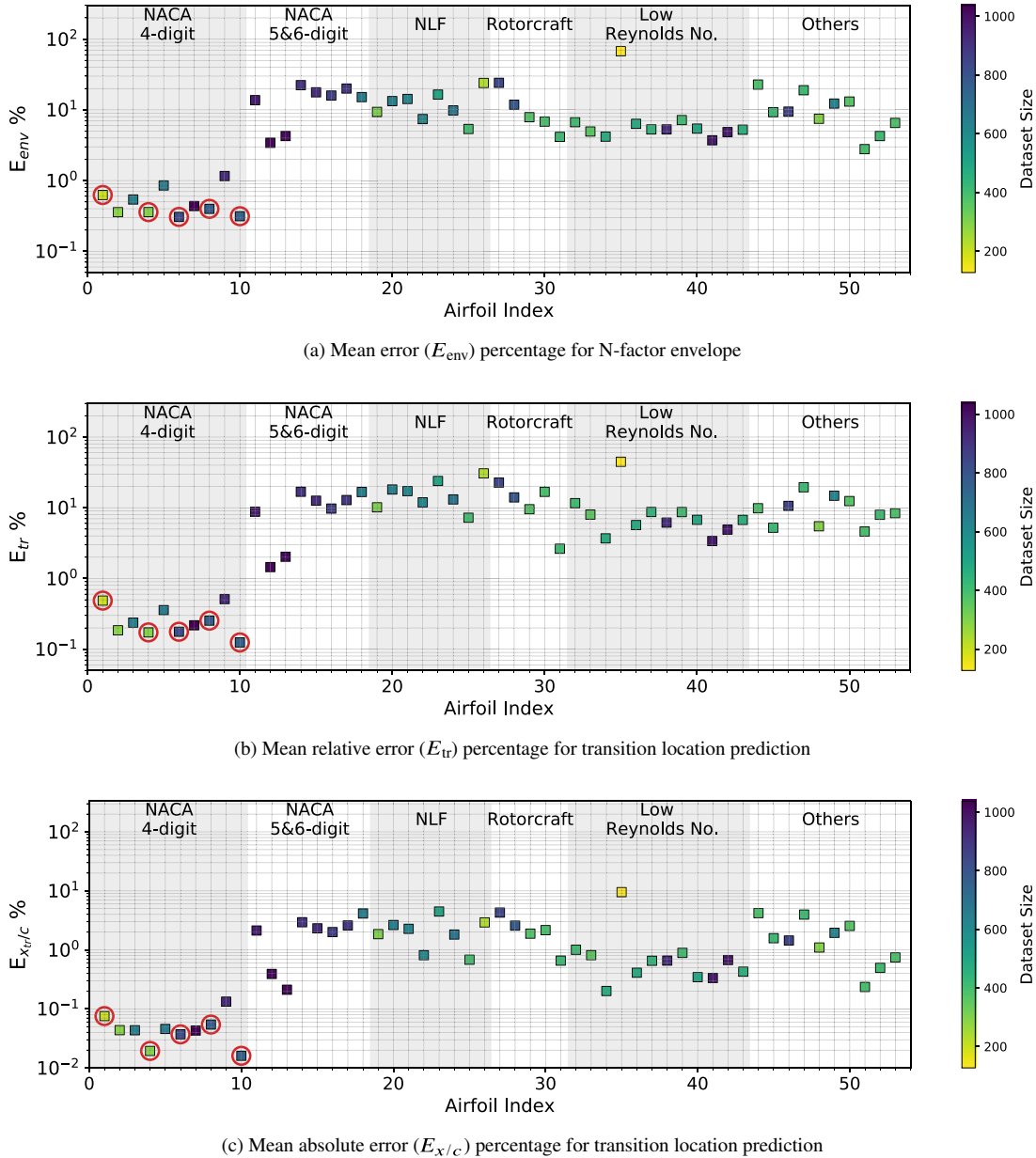


Figure 18. Mean error percentage for each airfoil in the database, corresponding to training dataset of Case IX given in Table 5. Airfoils corresponding to training dataset have been encircled in red color. Markers' color represent the dataset size (number of flow cases) of each airfoil in the database.

A significant advantage of the proposed RNN model over the previously proposed neural network-based transition models is that using the sequential information of the underlying mean flow, the RNN model is able to directly predict the required information of the N-factor envelope and the transition location, without requiring the user to define a range of critical frequencies and predicting the instability growth rates at a number of frequencies in this range. On the other hand, since the RNN model predicts the growth rates of the N-factor envelope in a sequential manner, it cannot be applied in a parallel manner,

unlike conventional methods or previously proposed neural networks that can predict the local growth rates at each station in a parallel manner.

The proposed architecture processes the boundary-layer profiles at each station in a physically consistent manner using the convolutional neural network. This attribute enables its generalization to other instability mechanisms involving three-dimensional boundary-layer profiles involving crossflow velocity components or second-mode instabilities in high speed flows involving the profiles of thermodynamic quantities such as density and/or temperature. Future work could involve the application of the proposed architecture to one of the other instability mechanisms. Furthermore, since the RNN model uses input data for the boundary-layer profiles, that depends on the airfoil contour and flow conditions, future explorations could involve airfoil contours along with angle of attack and Reynolds number as global inputs to predict the N-factor envelope. Use of a vector-cloud neural network can also be explored, as it would allow the user to employ boundary layer profiles defined at any arbitrary and variable number of grid points (Zhou et al., 2021).

Acknowledgments. The computational resources used for this project were provided by the Advanced Research Computing (ARC) of Virginia Tech, which is gratefully acknowledged.

Funding Statement. This research was supported by the Revolutionary Computational AeroSciences discipline of NASA's Transformational Tools and Technologies Project.

Competing Interests. The authors declare no competing interests.

Data Availability Statement. The data and code used in this work is currently proprietary and confidential; it will be considered for public release in the future.

Author Contributions. Conceptualization: M.-M.C. and H.X.; Methodology: M.-M.C., H.X., and M.-I.Z.; Data curation: P.P., M.-M.C., and M.-I.Z.; Formal analysis: M.-I.Z.; Writing-original draft: M.-I.Z. and M.-M.C.; Writing-review & editing: all. All authors approved the final submitted draft.

References

- Begou G, Deniau H, Vermeersch O and Casalis G** (2017) Database approach for laminar-turbulent transition prediction: Navier–Stokes compatible reformulation. *AIAA Journal* 55(11), 3648–3660.
- Bushnell DM, Malik MR and Harvey WD** (1989) Transition prediction in external flows via linear stability theory. In Zierep J., Oertel H. (eds), *Symposium Transonicum III. International Union of Theoretical and Applied Mechanics*. Berlin, Heidelberg: Springer.
- Crouch JD, Crouch IWM and Ng LL** (2002) Transition prediction for three-dimensional boundary layers in computational fluid dynamics applications. *AIAA Journal* 40(8), 1536–1541.
- Dagenhart JR** (1981) Amplified crossflow disturbances in the laminar boundary layer on swept wings with suction. *NASA-TP-1902*.
- Drazin PG and Reid WH** (1981) *Hydrodynamic Stability*. Cambridge: Cambridge University Press.
- Drela M** (1989) Xfoil: an analysis and design system for low Reynolds number airfoils. In Mueller TJ (eds), *Low Reynolds Number Aerodynamics*. Lecture Notes in Engineering, 54 Springer Berlin Heidelberg.
- Drela M and Giles MB** (1987) Viscous-inviscid analysis of transonic and low Reynolds number airfoils. *AIAA Journal* 25(10), 1347–1355.
- Duraisamy K, Iaccarino G and Xiao H** (2019) Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics* 51, 357–377.
- Faller WE and Schreck SJ** (1997) Unsteady fluid mechanics applications of neural networks. *Journal of Aircraft* 34(1), 48–55.
- Fuller RM, Saunders WR and Vandsburger U** (1997) Neural network estimation of disturbance growth using a linear stability numerical model. AIAA Paper 1997-0559.
- Gaster M and Jiang F** (1995) Rapid scheme for estimating transition on wings by linear stability theory. In *International Council of the Aeronautical Sciences Proceedings (ICAS)*, Anaheim, CA.
- Graves A** (2012) Sequence transduction with recurrent neural networks. In *International Conference of Machine Learning (ICML)*, Edinburgh, Scotland.
- Guastoni L, Srinivasan PA, Azizpour H, Schlatter P and Vinuesa R** (2019) On the use of recurrent neural networks for predictions of turbulent flows. In *11th International Symposium on Turbulence and Shear Flow Phenomena, TSFP11*, Southampton, UK.
- Hermans M and Schrauwen B** (2013) Training and analysing deep recurrent neural networks. In *Advances in Neural Information Processing Systems (NIPS 2013)*. Curran Associates, Inc. 72

- Juniper MP, Hanifi A and Theofilis V** (2014) Modal stability theory: Lecture notes from the FLOW-NORDITA summer school on advanced instability methods for complex flows. *Applied Mechanics Review* 66(2), 024804.
- Krumbein A** (2008) e^N transition prediction for 3D wing configurations using database methods and a local, linear stability code. *Aerospace Science and Technology* 12(8), 592–598.
- Langlois M, Masson C, Kafyeke F and Parascivoiu I** (2002) Automated method for transition prediction on wings in transonic flows. *Journal of Aircraft* 39(3), 460–468.
- Mack LM** (1987) Review of linear compressible stability theory. In Dwoyer DL and Hussaini MY (eds), *Stability of Time Dependent and Spatially Varying Flows. ICASE NASA LaRC Series*. New York, NY: Springer.
- Menter F, Langtry R and Völker S** (2006) Transition modelling for general purpose CFD codes. *Flow, Turbulence and Combustion* 77, 277–303.
- Michélin-Ströfer C, Wu J-L, Xiao H and Paterson E** (2018) Data-driven, physics-based feature extraction from fluid flow fields using convolutional neural networks. *Communications in Computational Physics* 25(3), 625–650.
- Paredes P, Venkatachari B, Choudhari MM, Li F, Chang C-L, Zafar MI and Xiao H** (2020) Toward a practical method for hypersonic transition prediction based on stability correlations. *AIAA Journal* 58(10), 4475–4484.
- Pascanu R, Gulcehre C, Cho K and Bengio Y** (2014) How to construct deep recurrent neural networks. In *Proceedings of the Second International Conference on Learning Representations (ICLR 2014)*.
- Perraud JP and Durant A** (2016) Stability-based Mach zero to four longitudinal transition prediction criterion. *Journal of Spacecraft and Rockets* 53(4), 730–742.
- Pinna F, Zanusi L, Demange S and Olazabal-Loume M** (2018) Reduced model for transition prediction in hypersonic flows. AIAA Paper 2018-3697.
- Rajnarayan DG and Sturdza P** (2013) Extensible rapid transition prediction for aircraft conceptual design using modal decomposition. AIAA Paper 2013-0231.
- Reed HL, Saric WS and Arnal D** (1996) Linear stability theory applied to boundary layers. *Annual Review of Fluid Mechanics* 28(1), 389–428.
- Reshotko E** (1976) Boundary-layer stability and transition. *Annual Review of Fluid Mechanics* 8(1), 311–349.
- Slotnick J, Khodadoust A, Alonso J, Darmofal D, Gropp W, Lurie E and Mavriplis D** (2014) CFD vision 2030 study: a path to revolutionary computational aerosciences. NASA/CR–2014-218178.
- Smith A and Gamberoni N** (1956) Transition, pressure gradient and stability theory. Douglas Aircraft Co., Report ES 26388.
- Stock HW and Degenhart E** (1989) A simplified e^N method for transition prediction in two-dimensional, incompressible boundary layers. *Zeitschrift für Flugwissenschaften und Weltraumforschung* 13(8), 16–30.
- Taira K, Brunton SL, Dawson STM, Rowley CW, Colonius T, McKeon BJ, Schmidt OT, Gordeyev S, Theofilis V and Ukeiley LS** (2017) Modal analysis of fluid flows: an overview. *AIAA Journal* 55(12), 4013–4041.
- UIUC Applied Aerodynamics Group** (2020) UIUC airfoil data site. https://m-selig.ae.illinois.edu/ads/coord_database.html.
- van Ingen J** (1956) A suggested semi-empirical method for the calculation of the boundary layer transition region. Technische Hogeschool Delft, Vliegtuigbouwkunde, Rapport VTH-74.
- van Ingen J** (2008). The e^N method for transition prediction. Historical review of work at TU Delft. AIAA Paper 2008-3830.
- Vlachas P, Pathak J, Hunt B, Sapsis T, Girvan M, Ott E and Koumoutsakos P** (2020) Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks* 126, 191–217.
- Wan ZY, Vlachas P, Koumoutsakos P and Sapsis T** (2018) Data-assisted reduced-order modeling of extreme events in complex dynamical systems. *PLoS One* 13(5), e0197704.
- Wang J-X, Wu J-L and Xiao H** (2017) Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on dns data. *Physical Review Fluids* 2, 034603.
- Wie YS** (1992). BLSTA—a boundary layer code for stability analysis. NASA CR 4481.
- Wu J, Yin X and Xiao H** (2018a) Seeing permeability from images: fast prediction with convolutional neural networks. *Science Bulletin* 63(18), 1215–1222.
- Wu J-L, Xiao H and Paterson E** (2018b) Physics-informed machine learning approach for augmenting turbulence models: a comprehensive framework. *Physical Review Fluids* 3, 074602.
- Zafar MI, Xiao H, Choudhari MM, Li F, Chang C-L, Paredes P and Venkatachari B** (2020) Convolutional neural network for transition modeling based on linear stability theory. *Physical Review Fluids* 5, 113903.
- Zhou X-H, Han J and Xiao H** (2021) Frame-independent vector-cloud neural network for nonlocal constitutive modelling on arbitrary grids. *Computer Methods in Applied Mechanics and Engineering*. arXiv preprint arXiv:2103.06685.

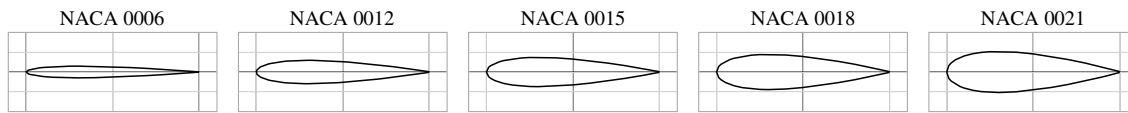
List of Airfoils

A listing of the 53 airfoils included in the database of stability characteristics is given in Table A1. Furthermore, a graphical catalog of all airfoil contours is also included, since that may be of interest to the readers.

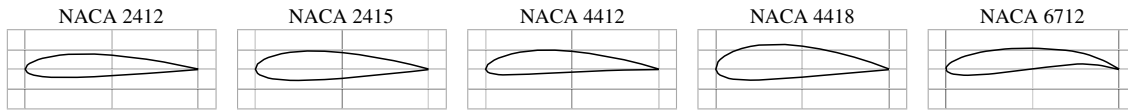
Table A1. List of airfoils in the database.

Index	Airfoil	Index	Airfoil	Index	Airfoil	Index	Airfoil
1	NACA 0006	15	NACA 63-415	29	VR-15	43	S8052
2	NACA 0012	16	NACA 64-215	30	OA209	44	ONERA M6
3	NACA 0015	17	NACA 63(2)-615	31	OA212	45	RAE 2822
4	NACA 0018	18	NACA 66(1)-212	32	E374	46	GA(W)-1
5	NACA 0021	19	NLF(1)-1015	33	E387	47	CLARK Y
6	NACA 2412	20	HSNLF(1)-0213	34	E472	48	LNV109A
7	NACA 2415	21	NLF(1)-0115	35	LRN(1)-1007	49	S8055
8	NACA 4412	22	NLF(1)-0215F	36	SD7003	50	S805a
9	NACA 4418	23	NLF(1)-0414F	37	SD7032	51	PSU 94-097
10	NACA 6712	24	NLF(1)-0414D	38	SD7062	52	SA7036
11	NACA 23015	25	NLF(1)-0416	39	SD7080	53	SA7038
12	NACA 23017	26	NLF(2)-0415	40	SD8020		
13	NACA 23024	27	VR-7	41	S8036		
14	NACA 63-215	28	VR-12	42	S8037		

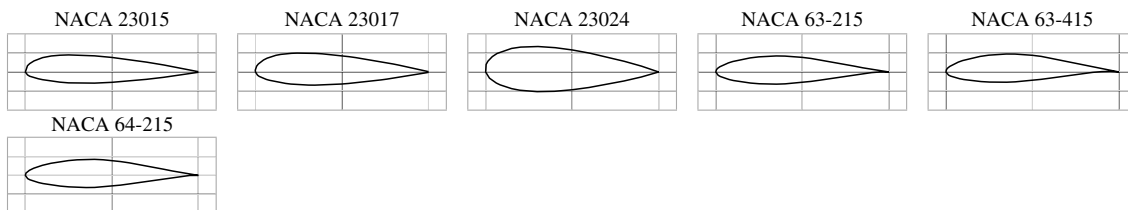
NACA 4-digit Airfoils - Symmetrical



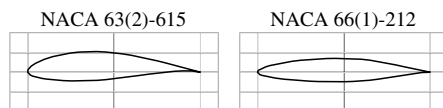
NACA 4-digit Airfoils - Cambered



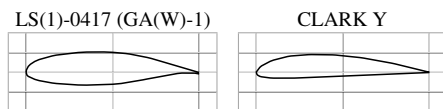
NACA 5-digit Airfoils



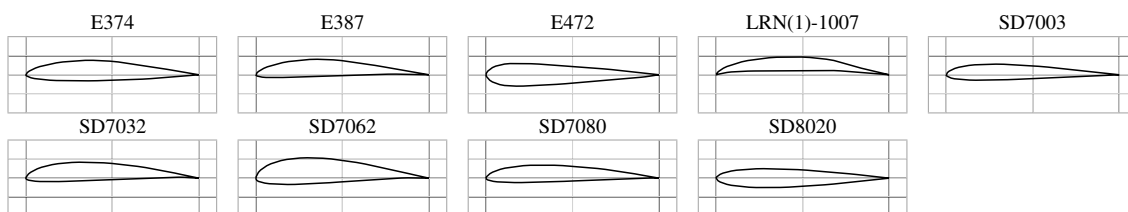
NACA 6-digit Airfoils



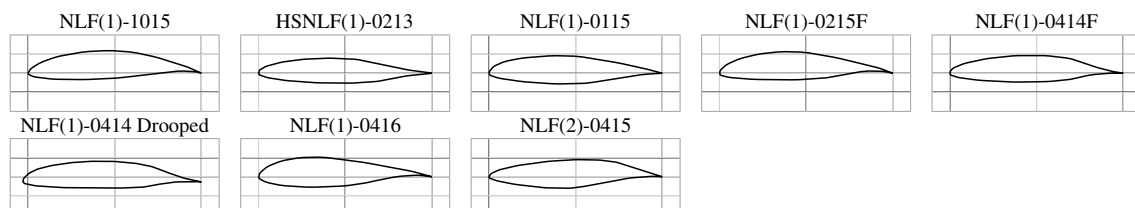
General Aviation Airfoils

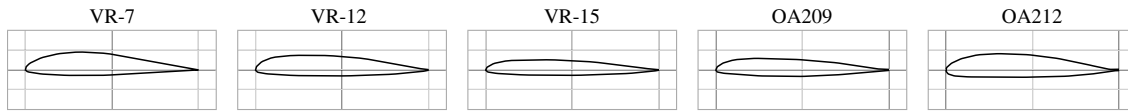
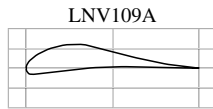
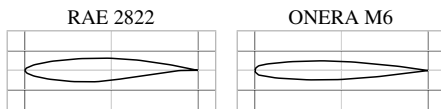
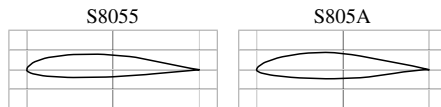
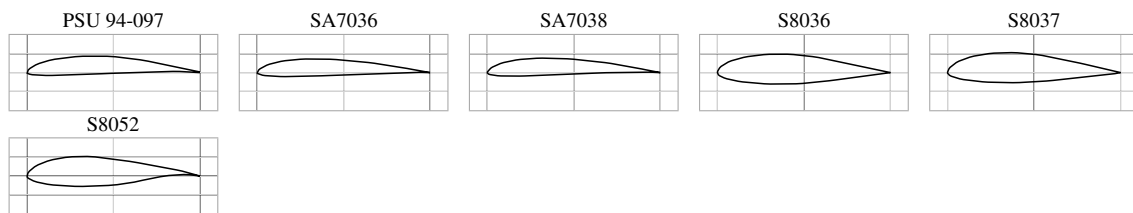


Low Reynolds number Airfoils



Natural Laminar Flow Airfoils



Rotorcraft Airfoils**High Lift Airfoils****Transonic Airfoils****Wind Turbine Airfoils****Miscellaneous Airfoils**

Chapter 4

Neural Operators

For many aerospace applications like aircraft design, stability-based methods (LST or PSE) along with semi-empirical correlations [1, 2] are the most widely used and reliable approaches for transition prediction. These stability-based methods can be used in a loosely coupled manner with RANS solvers for the computation of laminar, transitional and fully turbulent parts of a flowfield. Such coupled frameworks [3, 4, 5, 6, 7] use an iterative process, wherein boundary-layer data is provided either by laminar boundary-layer solver or by the current flow solution by RANS solver. Such data is processed to a format suitable for stability computations, including extraction of wall-normal profiles and streamlines along the surface. Stability computations about mean flow are then performed using LST or PSE to predict transition location, which is fed back to RANS solver for prescribing the regions of laminar and turbulent flow for next iteration. In similar frameworks, to overcome the challenges of increased computational cost and complexity associated with stability computations, CNN and RNN model can provide transition-closure as surrogate model for stability-based methods.

In broader perspective of vastly different application areas, it is important to develop a spectrum of transition methods/models, of which stability approaches form an essential component. However, there is a pressing desire to develop other efficient and reliable transition models which overcome the challenges associated with stability-based methods in many industrial applications. These challenges include their lack of compatibility with general-purpose CFD methods applicable to complex geometries. Required non-local operations, like computing integral boundary layer parameters, can not be performed with reasonable effort in general-purpose CFD codes, especially if parallel computation is being performed using domain decomposition methodology. For this reason, transition models with local formulation is highly desired for day-to-day industrial applications. To address these challenges, another strategy has made significant advancement, especially during the last decade, where transition models have taken the form of supplementary transport equations (PDEs).

4.1 PDE-based Transition Models

Similar to commonly used turbulence models in RANS simulations, PDE-based transition models provide CFD-compatible formulations [8, 9, 10]. Development of such PDE-based transition models has primarily been aimed at achieving compatibility with RANS solvers

using arbitrary grids and dependence on local variables only. Intermittency-based two equation γ - $Re_{\theta t}$ model [8] uses a combination of strain-rate Reynolds number and empirical correlations with standard transport equations. This model is built on the reasoning that strain-rate Reynolds number, computed based on local quantities, provides a link with transition Reynolds number $Re_{\theta t}$. Intermittency-based one equation γ model [11] was also later introduced with only one transport equation for γ , to reduce computational cost, and significantly simplified empirical correlations. Another transition model is based on approximating the envelope amplification factor of linear stability theory using only the local quantities [9]. A governing PDE describes the evolution of the approximate envelop amplification factor, where the source term is function of local shape factor estimated using a local pressure gradient parameter. Other RANS transition model is based on a transport equation for laminar kinetic energy (LKE) [10], which models the fluctuations in the boundary layer before transition. Once fluctuations reach a critical level, turbulence model is activated.

Data-driven techniques have also been proposed to enhance the predictive capability of these RANS-based transition models. Based on LKE transition model, gene expression programming has been used to model production and transfer terms of fluctuations for improved prediction of separation induced transition over low pressure turbines [12]. For intermittency-based transition model, a framework of field inversion and machine learning was proposed to improve four equations k - ω - γ - A_r model (A_r represents roughness amplification factor) [13]. A NN predicted correction term β has been introduced in the effective viscosity representation for improved transition onset prediction over an airfoil.

Akin to eddy viscosity based turbulence models, PDE-based transition models involve empirical correlations especially for production terms. These transition models are calibrated for specific transition mechanisms and developed for specific turbulence models. Assumptions of local dependence are introduced for many variables, although PDE-based formulation keeps track of nonlocal dependencies to some extent through convection and diffusion terms. Furthermore, solving additional transport equation(s) for transition closure adds to the computational costs of RANS simulations. To overcome these limitations, a specific class of purely data-driven NN-based models has the potential to emulate the behavior of PDEs on arbitrary grids. However, for accuracy and physical realizability, it is desired that these NN-based models for transition modeling should possess the attributes of PDE-based closure models. Such models and the desired attributes are discussed below.

4.2 Neural Operators

Neural operators are a recent development in deep learning that allow for the learning of mathematical operators *directly from data* without requiring the knowledge of underlying PDE. As many scientific problems reduce to a set of governing PDEs, these neural operators can be used to learn NN-based functionals replicating the behavior of PDEs and predict the PDE solution for given input function. These operators can be used to solve scientific

problems in a variety of fields, including physics, engineering, and biology. This can lead to more efficient and accurate solutions, as well as the ability to solve problems that were previously too complex to solve using traditional methods. This is in contrast to other NN-models like physics-informed neural network (PINN) [14] that aim to learn solution of a known PDE for a single instance with given initial and boundary conditions.

In the context of fluid dynamics, PDE-based closure models for turbulence/transition imply region-to-point mapping from the discretized flow field variables (like U , p etc.) to the respective closure variables (turbulent kinetic energy k , dissipation of turbulent kinetic energy ϵ , intermittency γ etc.). Furthermore, conventionally used neural network modeling approach employs local mapping (like point-to-point mapping for $[S, \Omega] \mapsto \tau$ in tensor-basis neural network). In contrast, PDE-based closure model manifest non-local dependencies to some extent through convection and diffusion terms of the closure variable. Neural operators can mimic this behavior and incorporate such nonlocal dependencies by taking into account information from neighbouring region around the point of interest where closure variable is to be predicted. Key attributes of neural operators are as follows:

- Purely data-driven, requiring no knowledge of underlying PDE
- Adaptive to different grid resolutions and discretizations
- Able to predict through a forward pass for a new set of parameters

Moreover, closure modeling poses the requirement of frame-invariance i.e. mapping to the respective closure variable should be independent from material reference frame. Ensuring such invariance can significantly improve the generalizability of the learned model.

4.2.1 Neural Operators for Transition Modeling

PDE-based transition closure models involve significant amount of empiricism in production terms as well as in defining the threshold for transition onset. Neural operators provide a useful tool for replacing the PDE-based transition closure models by learning mapping from auxiliary flow field (velocity, pressure, etc.) to closure variable (intermittency γ , amplification factor \tilde{n} , etc.), directly using high-fidelity data. Such NN-based closure model can be developed using data obtained from physics-based methods of stability theory and can be easily calibrated to different instability mechanisms.

Different NN-based formulations have been proposed which can be categorized as neural operators. DeepONet [15] is one such model, which has been presented to approximate solutions to parabolized stability equations which govern the evolution of perturbations in a streamwise boundary layer flow [16]. At a given flow condition, DeepONet was used to predict the evolution of disturbances from an upstream location to a downstream location. However, frame-invariance has not been ensured in the formulation of this NN-based model.

PointNet is another such NN-based framework which provides non-local mapping and ensures rotational invariance by using only scalar input features [17]. It has been presented to approximate solutions for incompressible laminar steady flow past a cylinder with various cross-sectional shapes and map spatial positions to flow quantities [18].

In the next chapter, two neural operators have been presented and examined to mimic the behavior of transport PDE while ensuring frame-invariance property. First neural operator is Graph kernel network (GKN) [19, 20] which operates on graph structured data similar to computational grids. Second neural operator is vector cloud neural network (VCNN) [21] which has embedded frame-invariance in the model and provides non-local mapping from point cloud data to scalar quantity.

Bibliography

- [1] A.M.O. Smith and N. Gamberoni. Transition, pressure gradient and stability theory. *Douglas Aircraft Co., Report ES 26388*, 1956.
- [2] J. van Ingen. A suggested semi-empirical method for the calculation of the boundary layer transition region. *Technische Hogeschool Delft, Vliegtuigbouwkunde, Rapport VTH-74*, 1956.
- [3] Andreas Krumbein, Normann Krimmelbein, and Geza Schrauf. Automatic transition prediction in hybrid flow solver, Part 1: Methodology and sensitivities. *Journal of Aircraft*, 46(4):1176–1190, 2009. doi: 10.2514/1.39736.
- [4] Andreas Krumbein, Normann Krimmelbein, and Geza Schrauf. Automatic transition prediction in hybrid flow solver, Part 2: Practical application. *Journal of Aircraft*, 46(4):1191–1199, 2009. doi: 10.2514/1.39738.
- [5] Jean Perraud, Daniel Arnal, Gregoire Casalis, Jean-Pierre Archambaud, and Raffaele Donelli. Automatic transition predictions using simplified methods. *AIAA Journal*, 47(11):2676–2684, 2009. doi: 10.2514/1.42990.
- [6] Balaji Shankar Venkatachari, Jared Carnes, Chau-Lyan Chang, and Meelan M. Choudhari. Boundary-layer transition prediction through loose coupling of overflow and lastrac. 2022. doi: 10.2514/6.2022-3682.
- [7] Nathaniel J. Hildebrand, Chau-Lyan Chang, Meelan M. Choudhari, Fei Li, Eric J. Nielsen, Balaji Shankar Venkatachari, and Pedro Paredes. Coupling of the fun3d unstructured flow solver and the lastrac stability code to model transition. 2022. doi: 10.2514/6.2022-1952.

- [8] Robin B. Langtry and Florian R. Menter. Correlation-based transition modeling for unstructured parallelized computational fluid dynamics codes. *AIAA Journal*, 47(12): 2894–2906, 2009.
- [9] J. G. Coder and M. D. Maughmer. Computational fluid dynamics compatible transition modeling using an amplification factor transport equation. *AIAA Journal*, 52(11):2506–2512, 2014. doi: <https://doi.org/10.2514/1.J052905>.
- [10] D. Keith Walters and James H. Leylek. A new model for boundary layer transition using a single-point rans approach. *Journal of Turbomachinery*, 126(1):193–202, 2004.
- [11] F.R. Menter, P.E. Smirnov, T. Liu, and R. Avancha. A one-equation local correlation-based transition model. *Flow, Turbulence and Combustion*, 95:583–619, 2015.
- [12] Muchen Yang and Zhixiang 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. doi: 10.1063/5.0008493.
- [13] Harshal D. Akolekar, Fabian Waschowski, Yaomin Zhao, Roberto Pacciani, and Richard D. Sandberg. Transition modeling for low pressure turbines using computational fluid dynamics driven machine learning. *Energies*, 14(15), 2021. doi: 10.3390/en14154680.
- [14] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- [15] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3:218–229, 2021.
- [16] Patricio Clark Di Leoni, Lu Lu, Charles Meneveau, George Em Karniadakis, and Tamer A. Zaki. Neural operator prediction of linear instability waves in high-speed boundary layers. *Journal of Computational Physics*, 474:111793, 2023. ISSN 0021-9991. doi: <https://doi.org/10.1016/j.jcp.2022.111793>.
- [17] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017.
- [18] Ali Kashefi, Davis Rempe, and Leonidas J. Guibas. A point-cloud deep learning framework for prediction of fluid flow fields on irregular geometries. *Physics of Fluids*, 33(2): 027104, 2021.

- [19] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. In *Advances in Neural Information Processing Systems*, volume 33, pages 6755–6766, 2020.
- [20] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020.
- [21] Xu-Hui Zhou, Jiequn Han, and Heng Xiao. Frame-independent vector-cloud neural network for nonlocal constitutive modeling on arbitrary grids. *Computer Methods in Applied Mechanics and Engineering*, 388:114211, 2022.

Chapter 5

Frame Invariance and Scalability of Neural Operators for Partial Differential Equations

Frame Invariance and Scalability of Neural Operators for Partial Differential Equations

Muhammad I. Zafar¹, Jiequn Han^{2,*}, Xu-Hui Zhou¹ and Heng Xiao¹

¹ Kevin T. Crofton Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, Virginia, USA.

² Center for Computational Mathematics, Flatiron Institute, New York, USA.

Received 28 December 2021; Accepted (in revised version) 9 May 2022

Abstract. Partial differential equations (PDEs) play a dominant role in the mathematical modeling of many complex dynamical processes. Solving these PDEs often requires prohibitively high computational costs, especially when multiple evaluations must be made for different parameters or conditions. After training, neural operators can provide PDEs solutions significantly faster than traditional PDE solvers. In this work, invariance properties and computational complexity of two neural operators are examined for transport PDE of a scalar quantity. Neural operator based on graph kernel network (GKN) operates on graph-structured data to incorporate nonlocal dependencies. Here we propose a modified formulation of GKN to achieve frame invariance. Vector cloud neural network (VCNN) is an alternate neural operator with embedded frame invariance which operates on point cloud data. GKN-based neural operator demonstrates slightly better predictive performance compared to VCNN. However, GKN requires an excessively high computational cost that increases quadratically with the increasing number of discretized objects as compared to a linear increase for VCNN.

AMS subject classifications: (or PACs) To be provided by authors

Key words: Neural operators, graph neural networks, constitutive modeling, inverse modeling, deep learning.

1 Introduction

A wide class of important engineering and physical problems are governed by partial differential equations (PDEs) describing the conservation laws. Extensive research efforts have gone into formulating and solving these governing PDEs. Despite significant

*Corresponding author. Email addresses: jiequnhan@gmail.com (J. Han), zmuhammadirfan@vt.edu (M. I. Zafar), xuhuizhou@vt.edu (X.-H. Zhou), hengxiao@vt.edu (H. Xiao)

progress, major challenges remain related to the computational costs of solving complex PDEs for real life problems like turbulent flows, laminar-turbulence transition and climate modeling. To avoid these prohibitively high computational costs, developing accurate and efficient numerical approximations or surrogate models for PDEs has been a key area of research [1–4]. Machine learning based models [5–9] have the potential to provide significantly faster alternatives to the traditional methods [10,11] of surrogate modeling. For accuracy and physical realizability, it is desired that these machine learning based models closely mimic the properties of the governing PDEs.

One of the key features of these PDEs is *frame invariance*, which is an intrinsic property of all equations in classical mechanics from Newton’s second law to Navier Stokes equations. It signifies that the behavior of the physical systems does not depend on the origin or orientation of the reference frame of the observer. An invariance principle reflects a basic symmetry and is a basic requirement of all physical equations and constitutive models [12–14]. It is related to the objectivity of modeling: different modelers choosing different reference frames should arrive at the *same* answer. A model is frame-invariant to a transformation if the transformation of the input data does not alter the output of the function or model. In the context of fluid mechanics, any scalar variable like pressure or velocity magnitude is independent of any translation or rotation of the reference frame. Specifically, for example, a vector-to-scalar constitutive mapping $f: \mathbf{q} \mapsto \tau$ should remain unchanged in the frame rotated by matrix R , i.e., the same mapping should be valid for $f: R\mathbf{q} \mapsto \tau$. In other words, for any mapping $f: \mathbf{q} \mapsto \tau$ to be frame-invariant, the mapping $f: \mathbf{q}' \mapsto \tau$ should hold for any rotation matrix R , with $\mathbf{q}' \equiv R\mathbf{q}$ being the input vector in the new, rotated coordinate system. Invariance with respect to other transformations (e.g., translation of origin or change of reference velocity) can be defined and interpreted similarly. For solid mechanics, the magnitude of the deformation tensor is invariant to the orientation of the reference frame or the origin of the frame – it is an objective quantity regardless of the observer. Clearly, any modeling equations or constitutive relations should faithfully reflect such invariance or symmetries. Furthermore, in the numerical solutions of these PDEs, frame invariance also requires permutational invariance, which ensures independence of the results from the order in which the discretized objects are indexed. Examples of such objects include elements, cells, grid points, or particles, depending on the specific numerical method used. In a hypothetical simple example, assume the scalar τ at a given location x_0 is a function of the vectors \mathbf{q}_1 , \mathbf{q}_2 , and \mathbf{q}_3 at three cells in the neighborhood of x_0 . The output τ must remain identical whether the input is $(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3)$, $(\mathbf{q}_3, \mathbf{q}_2, \mathbf{q}_1)$, or any other ordering of three vectors. In other words, permutation invariance demands that τ must only depend on the set as a whole and not on the ordering of its elements. Embedding these invariance properties in machine learning based models can significantly improve the generalizability of learned models [15–21].

In this paper, we have chosen an airfoil problem to investigate the impact of the frame-invariance property. However, this illustration on a simple geometry of airfoils should not, in any way, undermine the significance and impact of frame-invariance in practical applications. In many cases of aerodynamics simulations, it is straightforward to agree

upon a *common* reference frame based on the model geometry. However, in many practical applications there is no clear preference. A flat plate boundary layer (Fig. 1(a)) is a problem with an intrinsic coordinate system. For example, one can choose the coordinate where the x -axis is aligned with the flow direction, y -axis is in the wall-normal direction, and z -axis is determined by the right-hand rule. Of course, other choices of coordinate systems are equally acceptable, as long as they are unambiguous and consistent between the training and testing systems. However, for the modeling of a complex system like an aircraft, different components such as a fuselage, wing and the flaps (a high-lift device mounted at the trailing edge of a fixed-wing aircraft as shown in Fig. 1(b)) often imply different, conflicting “natural” choices of coordinate systems. Moreover, many aircraft have *twisted wings*, as shown in Fig. 1(c), often to ensure that the wing tip is the last part of the wing surface to stall due to lower angles of attack (AOA). This effectively requires that the chord of airfoil section along the wing span is rotated with respect to each other. Any choice of the reference frame at one location along the wing span would make it a rotated reference frame for other locations along the wing span. Finally, in rotating machinery (e.g., compressor and gas turbine in engines as shown in Fig. 1(d), centrifugal pumps, and wind turbines), the moving components (rotor) and the stationary components (stator, nacelle) also imply different choices of coordinate systems. Therefore, in this work we choose the airfoil at an angle of attack as a simplified yet very relevant example to demonstrate the conflicting choices of coordinate systems in fluid dynamics modeling. Similarly, the same invariance principle forms the basic requirement of constitutive relation between solid deformation response and the applied stresses, whether a simple geometry is modeled or a complex system in industrial application is being considered. The illustrated principle is generally applicable to all physical modeling applications.

In this work, we are particularly interested in using neural network-based solution operators, called *neural operators* [5, 9], to solve PDEs with a guarantee of frame invariance. Mathematically, given a PDE system, the *solution operator* maps the input function such as the initial condition, boundary condition, or auxiliary field to the output PDE solution. The solution operator plays an essential role in many applications like surrogate modeling, constitutive modeling, and Bayesian inverse problem. Note that the solution operator is defined with respect to a family of PDEs with different input functions, in contrast to classical methods like finite difference methods and modern machine learning based methods like physics-informed neural network [24, 25] that aim to determine the solution for a single instance of the PDE. Those methods are computationally expensive when multiple evaluations are required. In contrast, neural operators aim to approximate the solution operator through an explicit mapping (e.g., convolution with a Green’s kernel [26]) starting from the input function so that it operates significantly faster. It is also designed to be mesh-independent. Of course, learning a solution operator is significantly more challenging than determining the solution to a PDE in a single instance. But once the network parameters are properly optimized, the neural operator can predict the solution of a new instance of the PDE fast. In this work we examine two such neural operators, graph kernel networks and vector cloud neural networks.

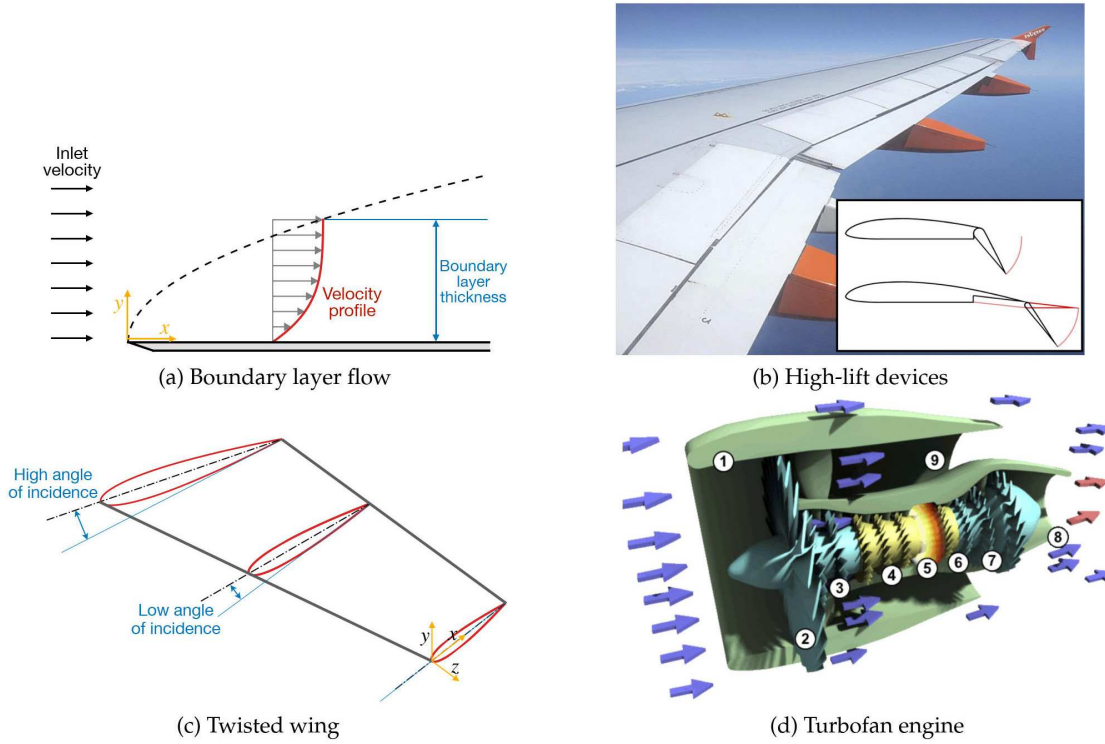


Figure 1: Illustration of significance and impact of frame-invariance in practical applications: (a) boundary layer flow over flat plate where a natural coordinate system is implied; (b) high-lift devices mounted at the trailing edge could imply conflicting reference frames, adapted from [22]; (c) twisted wing with airfoil sections along wing span rotated with respect to the reference frame; (d) turbofan engine in which several aerodynamics components are differently aligned with each other, adapted from [23].

Recently, graph based discretization of the physical domain has found prominence for the development of neural operators and more broadly, scientific simulations [27], using graph neural networks [28]. Graph kernel network [29, 30] is a specialized class of graph neural networks for implementing neural operators. GKN operates on graph-structured data, which is consistent with grids or data structures used in computational methods. A kernel network is used to incorporate nonlocal dependencies from other graph nodes. GKN can exhibit translational invariance if relative spatial information is considered for neighbouring nodes in the graph. In that case global attribute of the graph is invariant to a translation of the graph-structured data in the spatial domain. Similarly, the global arithmetic operation to determine the global attribute of the graph-structured data embeds permutational invariance, which guarantees independence from the order in which graph nodes are indexed. Rotational invariance can be achieved by alternative methods for graph networks. A simpler method, borrowed from computer vision, is to augment the training data with differently rotated coordinate systems. However, such a method is inefficient as it increases the data size and training time (by approximately a

factor of over ten as observed in previous research [15]). Moreover, the data augmentation does not guarantee frame invariance strictly – rather, the trained network only has approximate rotational invariance [31]. Another method to achieve rotational invariance is to use rotational invariant input features. This work will discuss such a method that can be effectively implemented with GKN to achieve frame invariance in a data-efficient manner.

The computational complexity of GKN is dependent on the sparsity of the underlying graph structure. When the graph is complete, i.e., every pair of nodes is connected with an edge, the computation cost scales quadratically with the number of nodes or data points in the graph. For some scientific problems, the graph size for capturing non-local dependencies can be of the order of hundreds or thousands nodes, and the graph is densely connected, like in the case of high Reynolds number flows. In such cases, GKN does not scale well and can become prohibitively expensive. This computational disadvantage of GKN was also discussed in [29] where a graph kernel network was used to emulate the Green’s function for the Poisson equation.

Vector cloud neural network (VCNN) provides an alternative strategy to develop a neural operator with embedding frame invariance [8]. The nonlocal information of the PDE is represented by a *vector cloud*, an arbitrarily arranged group of points akin to graph-structured data, with each point (node) having a feature vector attached to it. Input matrix Q is organised by stacking feature vectors corresponding to every point in the cloud. This vector cloud is mapped to a global scalar attribute of the cloud, through a frame-invariant neural network-based model. Spatial coordinates for neighbouring points in the cloud are organised relative to the central point of the cloud, which ensures translational invariance. Pairwise projection $D' = QQ^\top$ of feature vectors is used to obtain rotational invariant features. Furthermore, an embedding network is used to obtain a learned basis \mathcal{G} and the rotational invariant features are projected on this learned basis as $D = \frac{1}{n^2} \mathcal{G}^\top D' \mathcal{G}$ where n is the number of points in the cloud. Such projections output the average point features on the learned basis and guarantee permutational invariance in the model. Invariant feature matrix D is then nonlinearly mapped to the unknown scalar variable of PDE through a fitting network. For predicting the unknown variable in the entire field, VCNN has shown to be computationally more efficient compared to traditional mesh-based PDE solvers such as those in finite volume methods as the number of mesh cells in the computational domain scales [8]. Another comparable network architecture with embedded frame invariance, PointNet [32], also operates on point clouds to learn point-wise latent features by mapping through a shared embedding network. A global feature is then extracted by using an aggregate function (like max-pooling) to ensure permutational invariance. Rotational invariance is ensured by the use of scalar input features only or together with vector features defined in a rotational equivariant coordinate system [33]. Considering grid vertices of a CFD domain as a point cloud, PointNet has been used to define an end-to-end mapping from the spatial positions of the grid vertices over irregular geometries to the corresponding velocity and pressure fields of the flow [34].

In the present work, we use flows over a symmetric airfoil to investigate the frame invariance property and scalability of the neural operators discussed above. As a proof of concept, we perform this investigation for a neural operator developed for a transport equation of a scalar variable. The behavior of such a scalar variable can also be closely related to scalar quantities in turbulence modeling including turbulent kinetic energy (TKE) and turbulent length scale [35]. In this preliminary work, we consider a hypothetical dimensionless concentration tracer τ , which is transported by a laminar velocity field \mathbf{u} . Although the originally proposed GKN does not ensure rotational invariance, we have modified it to use rotational invariant input features and map them to the scalar variable τ . Such results are then compared against the embedded frame invariance of VCNN. The results are analyzed for accuracy and computational efficiency.

The rest of the manuscript is organised as follows. Methodologies used for this work are presented in Section 2. Generation of data for training and testing of the neural operators is discussed. Both neural operators, GKN and VCNN, are also presented while elaborating on their frame invariance properties. Section 3 presents the analysis of frame invariance properties, predictive performance and computational efficiency of both models. Section 4 concludes the paper.

2 Methodology

2.1 Generation of data for training and testing

Flow over a symmetric NACA-0012 airfoil in a two-dimensional space is considered for this study. The airfoil contour was obtained from a public domain source, UIUC Airfoil Coordinates Database [36]. C-mesh flow domain is used to appropriately distribute the mesh density along the critical areas of flow around an airfoil. For an airfoil with unit chord length, the width of the flow domain extends five times the chord length towards the upstream direction and 10 times the chord length towards the downstream direction of the airfoil leading edge. The domain extends five times the chord length in the direction normal to the airfoil chord.

The scalar quantity τ is a hypothetical, dimensionless concentration tracer, which is transported by the velocity field \mathbf{u} . A simplification is assumed for this study that the velocity field is not affected by the concentration tracer. To compute the scalar field $\tau(\mathbf{x})$, the transport PDE is solved with a steady, laminar flow field $\mathbf{u}(\mathbf{x})$. Resembling the transport equation of other scalar quantities like turbulent kinetic energy, eddy viscosity, etc., the transport PDE for concentration τ is given as:

$$\mathbf{u} \cdot \nabla \tau - \nabla \cdot (C_v \nabla \tau) = P - E, \quad (2.1)$$

where P indicates the production term, E indicates the dissipation term and C_v represents the diffusion coefficient [37]. Both the production and dissipation terms depend on the scalar quantity τ . The production is further dependent on the mixing length ℓ_m and

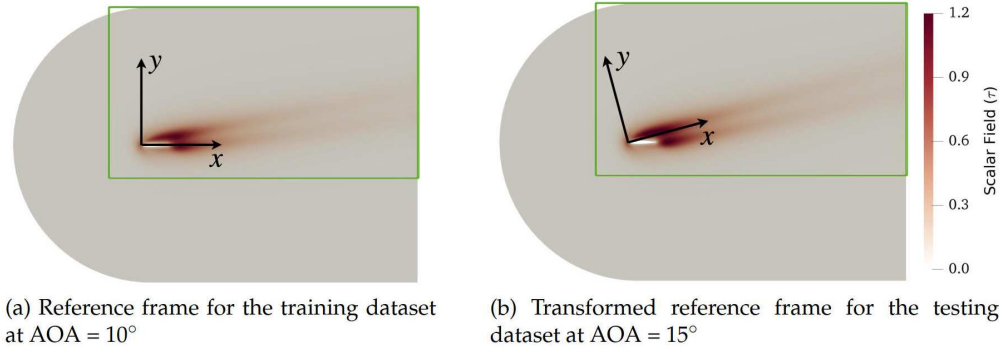


Figure 2: Different frames of reference used for training and testing datasets. Contours show the scalar field τ transported by flow over a symmetric airfoil (NACA-0012). The training dataset is based on Cartesian coordinate system with origin coinciding the airfoil leading edge and x -axis aligned with the airfoil chord. Coordinate system for the testing dataset is transformed/rotated by the given angle of attack (AOA), with origin still coinciding with the airfoil leading edge. Red box indicates the truncated flow domain. Data outside the green box is discarded to avoid redundancy.

strain-rate magnitude $\|s\|$. Both terms are given as:

$$P = C_g \ell_m \sqrt{\tau} s^2 \quad \text{and} \quad E = C_\zeta \tau^2,$$

$$\text{with} \quad s = \|s\| = \left\| \nabla \mathbf{u} + (\nabla \mathbf{u})^\top \right\|,$$

where C_g and C_ζ are production and dissipation coefficients, respectively. Further details for this transport equation and solver have been discussed in detail by [8].

Laminar flow field is simulated first to obtain the velocity field \mathbf{u} . Using this velocity field, scalar field τ is then computed by solving transport PDE for concentration tracer. Data has been generated for a set of angles of attack (AOA) and divided into training and testing dataset as:

$$\begin{aligned} \text{Training dataset:} & \quad [10^\circ, 20^\circ, 30^\circ], \\ \text{Testing dataset:} & \quad [5^\circ, 15^\circ, 25^\circ, 35^\circ]. \end{aligned}$$

Testing dataset has been organised to interpolate to testing data at AOA = (15°, 25°) and extrapolate to testing data at AOA = (5°, 35°) flow conditions. Furthermore, in the testing dataset, a transformed reference frame is used, which is rotated by a value equal to the AOA in the corresponding flow. Such distinction is demonstrated in Fig. 2. This transformation of the reference frame is specifically implemented to analyze the frame invariance behavior of the neural operators. Furthermore, to avoid redundancy, the data is extracted from a truncated domain by discarding the region which does not have any concentration tracer distribution at any specified flow conditions.

To generate data for nonlocal mapping from a patch of flow field $\mathbf{u}(\mathbf{x})$ to concentration τ at the point of interest, the setting is largely adapted from [8, 26]. The flow features' vector $\mathbf{q} = [\mathbf{x}', \mathbf{u}, \mathbf{c}] \in \mathbb{R}^{11}$ used at each point is chosen to include relative coordinate features

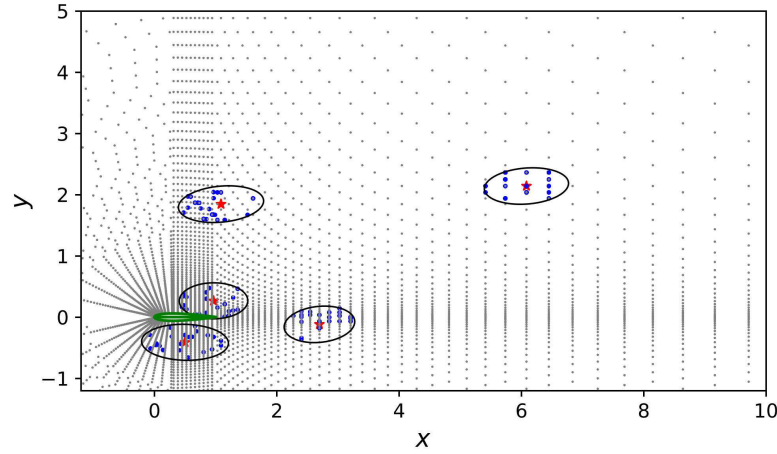


Figure 3: Sampling a cloud of data points for neural operator mapping. Airfoil surface (\bullet) is shown in green color. The gray dots (\bullet) indicate cell centers, showing every seventh cell with respect to the index of the structured mesh for clarity. For any location (\star) where the concentration τ is to be predicted, ellipse (\circ) shows the physical region of influence as determined by the velocity, diffusion and dissipation coefficients at that point; see Eq. (2.2). The blue dots (\bullet) are the randomly sampled cell centers within the region of influence and the feature vectors attached to them are used as input matrix \mathcal{Q} to predict τ .

$\mathbf{x}' = [x', y']^\top$, flow velocity \mathbf{u} and seven additional scalar quantities, collectively denoted by \mathbf{c} . Note that our convention in this paper is that all vectors are column vectors, and $\llbracket \cdot \rrbracket$ denotes vertical concatenation. Those scalar quantities in \mathbf{c} include mesh cell volume (θ), magnitude of strain rate (s), boundary cell indicator (b), velocity magnitude (u), wall distance (η), proximity of cloud center (r), proximity in local velocity frame (r'). Detailed descriptions and mathematical definitions of these scalar quantities can be found in [8].

Flow features are considered at randomly selected fixed number of points within the region of influence around the point of interest where concentration τ has to be determined. Fig. 3 illustrates the regions of influence and respective sampled data points corresponding to different points of interest in the flow domain. The extent of the region of influence, represented by an ellipse, depends on the velocity, diffusion and dissipation coefficients at the point of interest. The half-lengths ℓ_1 and ℓ_2 of the major and minor axis of the ellipse are determined based on the specified relative error tolerance ϵ :

$$\ell_1 = \left\lceil \frac{2\nu \log \epsilon}{\sqrt{|\mathbf{u}|^2 + 4\nu\zeta} - |\mathbf{u}|} \right\rceil \quad \text{and} \quad \ell_2 = \left\lceil \sqrt{\frac{\nu}{\zeta}} \log \epsilon \right\rceil, \quad (2.2)$$

where ν and ζ are the diffusion and dissipation coefficients, for which details can be found in [26]. The major axis of the ellipse aligns with the direction of the velocity \mathbf{u} . Each input-output combination for the neural operators then corresponds to an elliptical patch of flow domain, where input data is the stack of feature vectors arranged as input feature matrix $\mathcal{Q} = [q_1, q_2, \dots, q_n]^\top$ and output is the scalar variable τ at the point of interest in that elliptical patch. Stencil size (n) represents the number of randomly sampled fixed

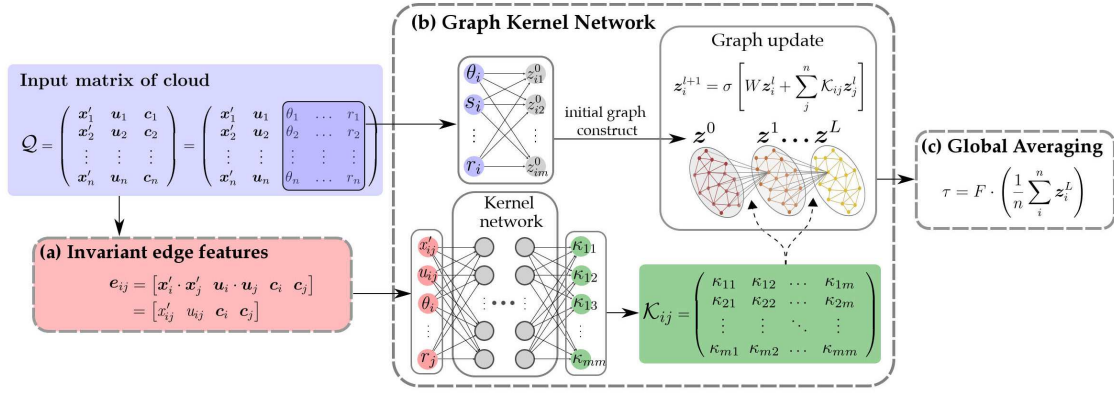


Figure 4: Detailed schematic of graph kernel network for region-to-point mapping from input matrix Q to scalar variable τ : (a) organises rotational invariant edge features between two points/nodes in the cloud; (b) map scalar features from input matrix to initial graph-based representation z^0 in latent space; then update this graph by assimilating nonlocal information in invariant edge features $e_{ij} = e(q_i, q_j)$ through neural network based kernel function $K_{ij} = K(e_{ij})$; (c) perform the averaging operation over z_i^L in the last GKN layer and the inner product with learnable parameters' vector to determine the global attribute of the graph scalar variable τ . $W \in \mathbb{R}^{m \times m}$ and $F \in \mathbb{R}^{1 \times m}$ are matrices of learnable parameters.

number of cloud data points from that elliptical patch, or equivalently the amount of information considered from the region of nonlocal influence. Constant stencil size is considered throughout the flow domain. For the locations where the number of data points in the ellipse is smaller than the specified stencil size, the available data points are repeatedly sampled.

2.2 Graph kernel network

Graph kernel network is based on graph neural networks, a class of neural networks adapted to leverage the structure and properties of graphs. A graph is an extremely powerful and general representation of data used to express a collection of entities (nodes) and the connections (edges) between them. Depending on the application, a graph can have attributes/features attached to the nodes and/or edges, and in some cases each graph can have a global attribute. Such specialized neural networks have found several practical applications such as antibiotic discovery [38], physics simulations [39], fake news identification [40], traffic prediction [41] and recommendation systems [42].

In the context of PDEs, graph nodes can be associated with the mesh cells and the cells can be connected through edges. To predict τ at any particular point, one can construct a graph among the neighbouring nodes in the region of influence and assimilate the nonlocal information through corresponding edges. Scalar quantity τ is the global attribute for each graph which can then be determined by considering the attributes of all the nodes in the graph.

Li et al. [29] proposed GKN to learn mappings between function spaces represented

by graph-based data. The mapping is formulated by using a neural network-based kernel function. Fig. 4 shows a schematic of the GKN architecture in the context of our example. The graph is completely connected, and the connection between two nodes in a graph is attributed by edge features as:

$$\hat{e}(q_i, q_j) = [x'_i, x'_j, u_i, u_j, c_i, c_j], \quad (2.3)$$

which are obtained by *concatenating* the flow feature vectors $q = \llbracket x', u, c \rrbracket$ of the corresponding nodes. Here i and j represent two corresponding nodes of the edge. Unlike scalar features c , relative coordinates x' and velocity u does not possess rotational invariance, rendering edge features in Eq. (2.3) dependent on the choice of frame orientation. Therefore, to ensure rotational invariance, we propose here to use rotational invariant edge features by *projecting the vectors* on to each other and by *concatenating the scalars*, which are specifically given as:

$$e(q_i, q_j) = [x_i'^\top x_j', u_i^\top u_j, c_i, c_j]. \quad (2.4)$$

Note that except the difference between Eq. (2.3) and Eq. (2.4) in edge features, our modified GKN has the same structure as that in [29], as described below.

Given edge features, kernel integration is computed to approximate mapping from one graph-based representation to another. Each graph-based representation is referenced as a layer (l) and multi-layer GKN enables capturing long-range correlations in graph-based data [29]. Initial graph-based representation in latent space was proposed in [29] to be based on the complete feature vector q on each node. However, here we consider only the scalar features c for initialization, which possesses rotational invariance. Each GKN layer update, referred to as graph convolution, can then be expressed as:

$$z_i^0 = Zc_i, \quad (2.5a)$$

$$z_i^{l+1} = \sigma \left[Wz_i^l + \frac{1}{n} \sum_{j=1}^n \mathcal{K}(e_{ij}) z_j^l \right], \quad l = 1, 2, \dots, L-1, \quad (2.5b)$$

where $i = 1, 2, \dots, n$ indexes the nodes in the graph representation, z_i^0 is a vector of m -dimension representing the latent node features of i th node in the initial graph representation, e_{ij} is the shorthand notation of $e(q_i, q_j)$, $Z \in \mathbb{R}^{7 \times m}$ represents the learnable parameters of a fully connected linear layer, $W \in \mathbb{R}^{m \times m}$ represents a matrix of learnable parameters for information passing among nodes, L is the total number of layers or depth of GKN and \mathcal{K} represents the neural network based kernel function. Details of the kernel network architecture have been mentioned in the Appendix B (Table 6). The output of \mathcal{K} is m^2 -dimensional and reshaped to a square matrix. Latent features of each node in the graph representation (z_i^l) are updated using kernel network to incorporate the information from all the connected nodes and learnable parameters' matrix W to process the latent features of the node itself in the previous layer. Hence each node in the updated

graph representation is connected to every node in the previous graph representation. This graph update is performed for specified number of times enumerated as the depth (L) of GKN.

For scalar field τ over an airfoil, GKN is purposed for region-to-point mapping. In this regard, to obtain a global attribute for the graph representation at layer L , a global averaging operation is performed as:

$$\tau = F \cdot \left(\frac{1}{n} \sum_{i=1}^n z_i^L \right), \quad (2.5c)$$

where $F \in \mathbb{R}^{1 \times m}$ represents learnable parameters and z_i^L represents the latent features of i -th node in the last GKN layer (L). Eqs. (2.5)(a-c) together complete the GKN based model. The global averaging operation in (2.5c) guarantees permutational invariance for GKN. With translational invariance already guaranteed, rotational input features and global averaging operation ensures the frame invariance property of the above GKN based model.

2.3 Vector cloud neural network

Zhou et al. [8] recently proposed a frame-independent neural operator, which is able to map to a scalar variable at a point based on the nonlocal information in its neighbourhood. Similar to graph neural networks, such nonlocal information is taken into account as an arbitrarily arranged group of points referred to as *vector cloud*, with a feature vector attached to each point. Since this nonlocal information can be captured from arbitrarily arranged points, irrespective of the ordering of points in the cloud, VCNN is a powerful tool to deal with unstructured meshes in scientific simulations.

Region-to-point mapping from input matrix \mathcal{Q} of vector cloud features to the scalar variable τ is illustrated in Fig. 5 as a schematic figure of VCNN. Input matrix $\mathcal{Q} = [q_1, \dots, q_n]^T \in \mathbb{R}^{n \times 11}$ is organised by stacking feature vectors corresponding to every point in the cloud. Each row in \mathcal{Q} is a feature vector on a point. With the usage of relative coordinates x' and pairwise projections $\mathcal{Q}\mathcal{Q}^T$ among the feature vectors, a direct mapping of the form $\tau = \hat{g}(\mathcal{Q}\mathcal{Q}^T)$ can ensure translational and rotational invariance. However to guarantee frame invariance completely, permutational invariance also needs to be embedded. For such purpose, an embedding network is used, as shown in Fig. 5(a). Note that scalar features c_i already possess translational and rotational invariance. These scalar features c_i are mapped through an embedding network ϕ to obtain m -dimensional embedded weights $\phi(c_i)$ corresponding to each point, and form an embedding matrix as:

$$\mathcal{G} = [\phi(c_1), \dots, \phi(c_n)] \in \mathbb{R}^{n \times m}, \quad (2.6a)$$

corresponding to all the points in the cloud.

To introduce permutational invariance, an order-removing transformation is performed as $\mathcal{G}^T \mathcal{Q}$. Combining it with the pairwise projections $\mathcal{Q}\mathcal{Q}^T$ among the feature vectors, in-

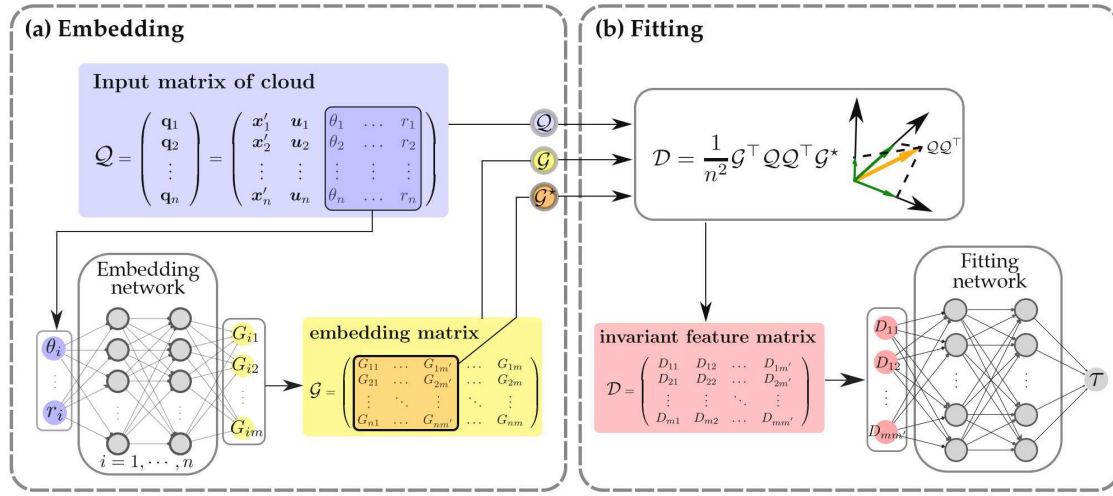


Figure 5: Detailed schematic of the vector cloud neural network for region-to-point mapping from input matrix Q to scalar variable τ : (a) map the scalar features in Q to a set of learned basis \mathcal{G} through an embedding network, $\mathcal{G}^* \in \mathbb{R}^{n \times m'}$ is the first $m' (\leq m)$ columns of \mathcal{G} ; (b) project the pairwise inner-product matrix $Q Q^\top$ to the learned embedding matrix and its submatrix to obtain invariant feature matrix $\mathcal{D} \in \mathbb{R}^{m \times m'}$; which is then mapped through fitting network to predict scalar variable τ .

variant feature matrix can be obtained as

$$\mathcal{D} = \frac{1}{n^2} \mathcal{G}^\top Q Q^\top \mathcal{G}^*, \quad (2.6b)$$

where \mathcal{G}^* is taken as a subset of \mathcal{G} to save computational cost. Equivalently, this can be written as $\mathcal{D} = \mathcal{L} \mathcal{L}^{*\top}$ where $\mathcal{L} = \frac{1}{n} \mathcal{G}^\top Q$ and \mathcal{L}^* similarly defined for \mathcal{G}^* . The normalization by the number n of points in the cloud allows the training and prediction to use different number of sampled point in the cloud. The obtained invariant feature matrix is then mapped through a fitting network (Fig. 5(b)) to the scalar variable τ , denoted as:

$$\tau = \mathcal{F}_{NN}(\mathcal{D}), \quad (2.6c)$$

where \mathcal{F}_{NN} denotes the nonlinear fitting network.

The combination of embedding network (Eq. (2.6a)), projection mapping (Eq. (2.6b)) and fitting network (Eq. (2.6c)) thus forms a complete VCNN model that has embedded frame invariance in the model. Detailed architectures of embedding and fitting networks have been mentioned in the Appendix B (Table 5). All code accompanying this manuscript are available on GitHub [43]. Besides embedded frame invariance attribute, VCNN has demonstrated reasonable computational efficiency compared to traditional mesh-based PDE solvers such as those in finite volume methods for predicting the unknown variable in the entire field, especially as the number of mesh cells in the computational domain scales [8]. With this perspective, the performance and scalability of VCNN to develop a frame-invariant neural operator is compared with that of GKN in the next section.

2.4 Comparative analysis of GKN and VCNN

Both neural operators, GKN and VCNN, presented above are compared schematically in Fig. 6. The same form of input matrix \mathcal{Q} is provided for both models. Both models process the data and ensure frame invariance properties in different manners. Different components of each model can be related with respect to their role in ensuring frame invariance and accurately predicting the scalar variable.

To extract global nonlocal information, VCNN uses a fully-connected neural network (Fig. 6(d2)) for mapping from invariant feature matrix \mathcal{D} to the scalar output τ . In contrast, GKN uses a relatively intricate setting where a neural network based-kernel function incorporates the information from neighbouring points to update the latent graph representation sequentially over several layers, where the mapping between the layers is determined by a nonlinear neural network function \mathcal{K} operating on the edges $e(q_i, q_j)$. Such an elaborate process of incorporating nonlocal information provides GKN with more flexibility in learning the region-to-point mapping. However, the edge-based mapping is also the source of $\mathcal{O}(n^2)$ scaling of GKN in terms of training time and memory footprint, which will be shown later. In contrast, the high dimension of the edge-based features (or equivalently pairwise project of node vectors) is reduced in the projection op-

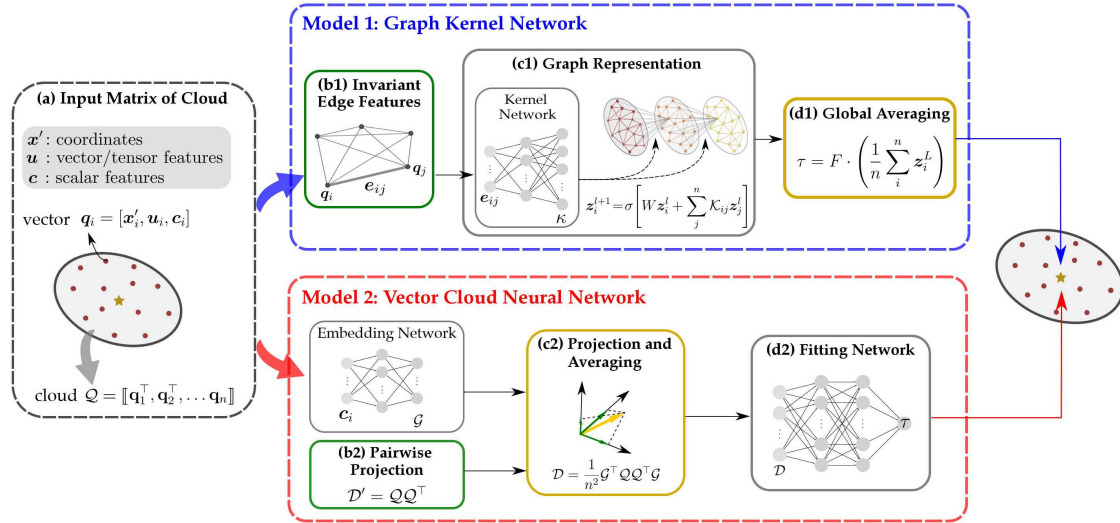


Figure 6: Schematic comparison of GKN and VCNN where different components of both models are outlined in the same color with respect to their role in achieving frame invariance for respective models. Translational invariance is ensured by the use of relative coordinates in the input cloud matrix \mathcal{Q} . Components contributing to rotational invariance are outlined in green where rotational invariant edge features (b1) and pairwise projections $\mathcal{Q}\mathcal{Q}^\top$ (b2) among the feature vectors are used. Components contributing to permutational invariance are outlined in yellow where global averaging (d1) and order-removing transformation $\mathcal{D} = \frac{1}{n^2} \mathcal{G}^\top \mathcal{Q}\mathcal{Q}^\top \mathcal{G}$ in projection mapping (c2) are used. Nonlocal mapping is achieved by the use of kernel network (c1) and by the fitting network (d2).

Table 1: Approximate correspondence between graph kernel network (GKN) and vector-cloud neural network (VCNN).

Element	GKN	VCNN
Ensure rotational invariance	Edge construction ^a : e_{ij} in Eq. (2.4)	Pairwise inner product ^b : $\mathcal{Q}\mathcal{Q}^\top$
Ensure permutational invariance	Plain node averaging: $\frac{1}{n}\sum_i z_i^L$	Weighted node averaging $\mathcal{L} = \frac{1}{n}\mathcal{G}^\top \mathcal{Q}$
Embody inter-node interactions	Kernel network (nonlinear) with edge input $\mathcal{K}(e_{ij})$	Pairwise (linear) projection $\mathcal{Q}\mathcal{Q}^\top$
Nonlinearity in network	Graph convolution with edge-dependent (nonlinear) weights	Node-wise embedding network (Fig.5a), fitting network $\mathcal{D} \mapsto \tau$
Degenerated to one layer	$\frac{1}{n^2}\mathcal{K}[\mathcal{Q}\mathcal{Q}^\top]\mathcal{C} + \frac{\mathcal{C}}{n}$	$\frac{1}{n^2}\sigma[\mathcal{C}]\mathcal{Q}\mathcal{Q}^\top\sigma[\mathcal{C}]^\top$

^a Not a built-in component of GKN, but can be implemented by constructing rotational invariant edge features.

^b Equivalently written as $\mathbf{q}_i^\top \mathbf{q}_j$.

eration $\mathcal{L} = \frac{1}{n}\mathcal{G}^\top \mathcal{Q}$ in the embedding network of the VCNN. In fact, the pairwise project matrix $\mathcal{Q}\mathcal{Q}^\top$ in Eq. (2.6b) is never formed explicitly in the implementation.

Another way to see the comparison between two models is by dropping the weights of both models and assuming one-layer graph convolution in GKN with edge features $\mathbf{q}_i \cdot \mathbf{q}_j$. In this case, if we denote the scalar feature matrix $\mathcal{C} = [c_1, \dots, c_n]$, the feature extraction part of both models can be summarized as

$$\begin{aligned} \text{GKN: } & \frac{1}{n^2}\mathcal{K}[\mathcal{Q}\mathcal{Q}^\top]\mathcal{C} + \frac{\mathcal{C}}{n}, \\ \text{VCNN: } & \frac{1}{n^2}\sigma[\mathcal{C}]\mathcal{Q}\mathcal{Q}^\top\sigma[\mathcal{C}]^\top, \end{aligned}$$

where \mathcal{K} is the kernel network (nonlinear function) in GKN operating on edges, while σ is the nonlinear functions in VCNN operating on the nodes. It is evident that the nonlinearity is applied to edges in GKN and to nodes on VCNN. The comparison is summarized in Table 1.

3 Results

In this section, we demonstrate and analyze the frame invariance property of both neural operators discussed above. Unless otherwise stated, the default stencil size n used to report results is 150, which is close to those used in the original works of GKN and VCNN. The results are presented as contours of the scalar field (τ) as well as in terms of prediction error percentage for the testing datasets at different angles of attack with a

transformed frame of reference. The prediction error percentage is defined as the normalized discrepancy between predicted scalar field $\hat{\tau}$ and the corresponding ground truth τ :

$$\text{Error} = \frac{\sqrt{\sum_{i=1}^N |\hat{\tau}_i - \tau_i|^2}}{\sqrt{\sum_{i=1}^N |\tau_i|^2}}, \quad (3.1)$$

where N represents the total number of points at which the scalar variable has to be predicted. In the first part of this section, we analyze the predictive performance and frame invariance property of the originally proposed GKN [29]. We demonstrate the effect of using rotational invariant input features for GKN. In the second part of this section, we compare two neural operators, VCNN that has frame invariance embedded in the neural operator and GKN for which frame invariance is achieved by introducing rotational invariant input features. Training cost and predictive performance are also analyzed for both neural operators.

3.1 Frame invariance for GKN

The originally proposed GKN [29] holds permutational and translational invariance properties, which are inherent properties of graph neural networks. However, such a network does not ensure rotational invariance, which can be critical for scientific problems. Reference systems can be differently aligned for different datasets, as shown in Fig. 2. A robust neural operator should have the ability to accurately predict for the dataset with a reference system different from that for the training dataset. For the specified problem setup of predicting concentration field τ in a flow over an airfoil, the neural operator should be able to accurately predict τ independent of the reference system of the dataset. In this regard, the frame-invariant characterization of GKN is analyzed here.

To achieve rotational invariance with GKN, we should use rotational invariant input features. The predictive performance of GKN with (Eq. (2.4)) and without (Eq. (2.3)) rotational invariant input features is compared in Table 2, which shows the prediction error percentage for the testing dataset with rotated reference system as compared to the

Table 2: Effect of rotational invariant input features for Graph Neural Network (GKN). Predictive error percentages for two cases correspond to Graph Neural Network (GKN) with and without rotational invariant (RI) input features. Results are computed for stencil size of 150. The training dataset comprises of flow data for three angles of attack (10° , 20° , 30°).

NN Model	Training Error	Test Error – Interpolation		Test Error – Extrapolation	
		AOA = 15°	AOA = 25°	AOA = 5°	AOA = 35°
GKN, no RI	0.27%	16.4%	23.5%	8.5%	27.0%
GKN, with RI	0.33%	1.7%	1.4%	2.7%	2.6%

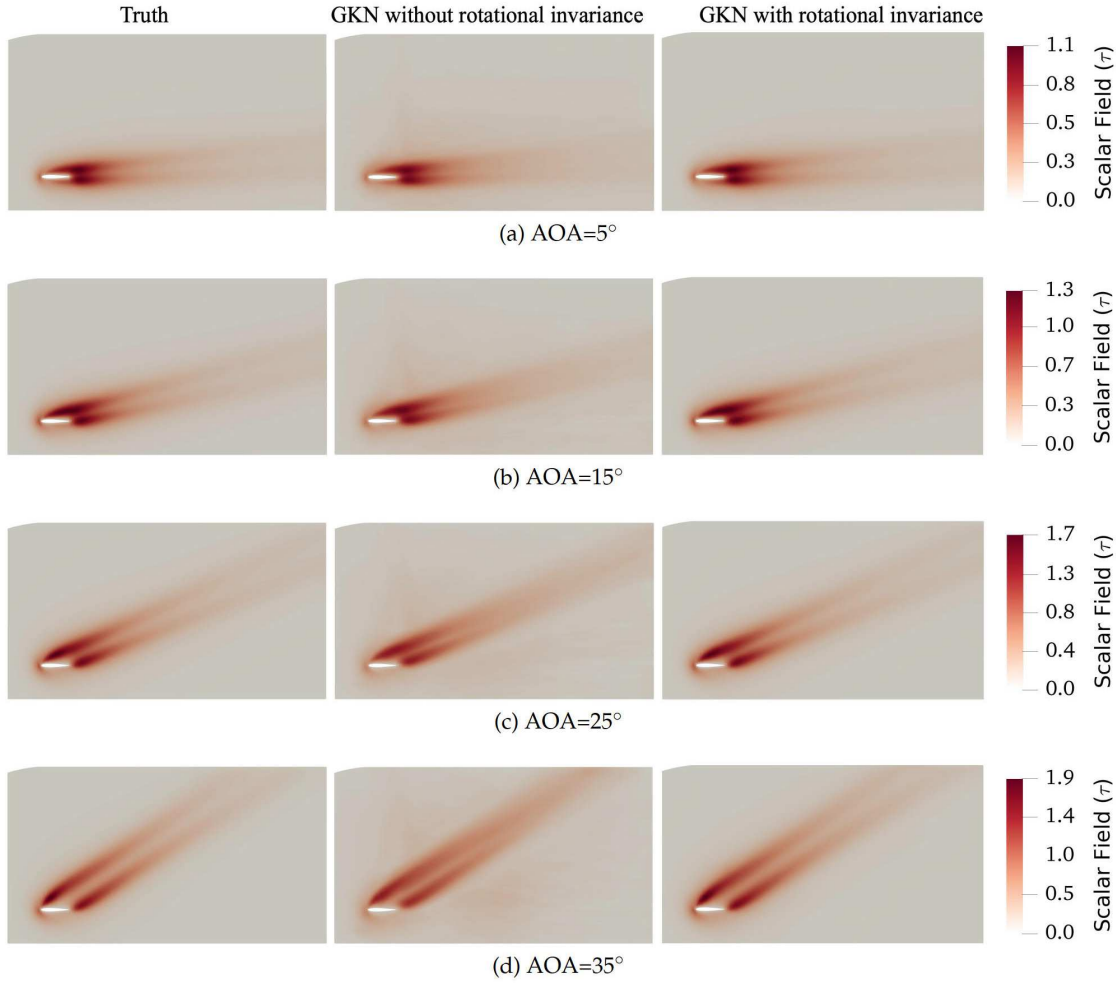


Figure 7: Comparison of graph kernel network predictions with and without rotational invariant input features. Plots show scalar transport field (τ) contours for four testing datasets at different airfoil's angle of attack (AOA).

training dataset (Fig. 2). It can be observed that the prediction error percentage for the testing dataset is much worse for GKN without rotational invariant input features. For the testing dataset, the reference system is rotated equally to the AOA for the specific flow. In that regard, as the rotational transformation of the reference system increases (with increasing AOA), the prediction error percentage increases from 8.5% at 5° AOA to 27% at 35° AOA. In comparison, GKN with rotational invariant input features predicts with significantly higher accuracy for the testing dataset at all angles of attack. Moreover, it follows a general trend of neural network models, i.e., a better prediction accuracy for interpolating in test conditions than extrapolating in test conditions.

The contours of the predicted scalar field τ are shown in Fig. 7, where predictions of

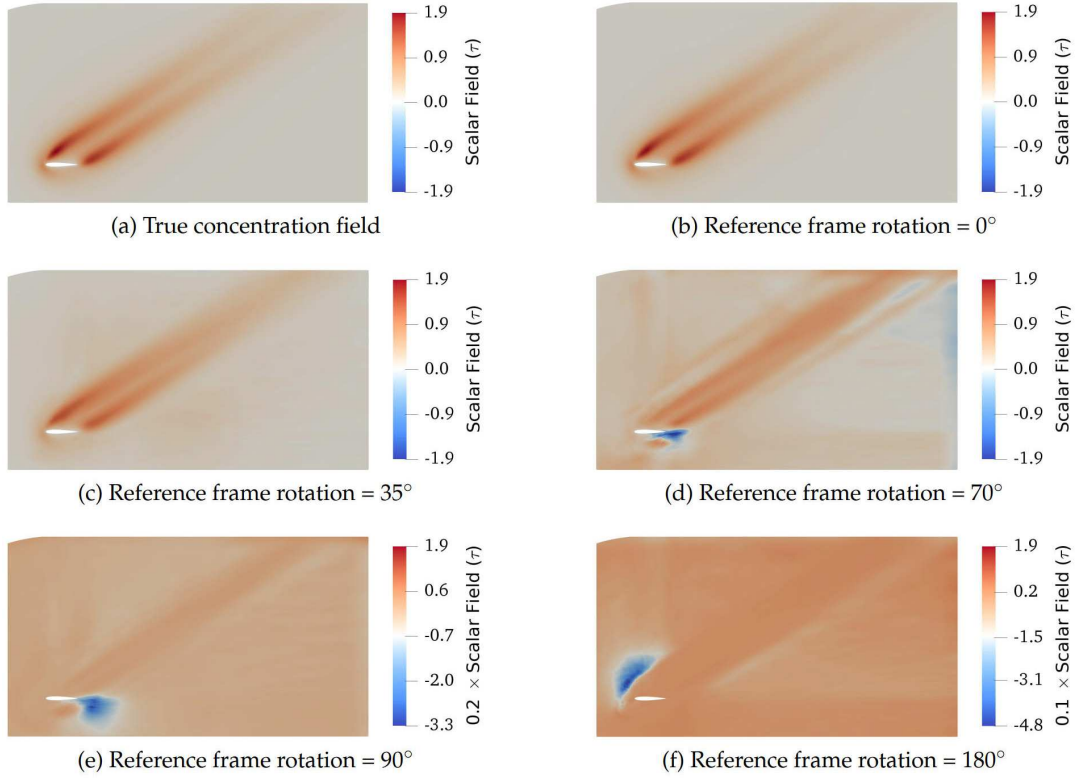


Figure 8: Contours of scalar field τ predicted by graph kernel network without rotational invariance with input data reference frame rotated by arbitrarily selected angles. Results are shown for the testing data corresponding to $\text{AOA}=35^\circ$. Scalar quantity τ has been scaled down by factors of 5 and 10 for reference frame rotation angles of 90° and 180° , respectively.

GKN with and without rotational invariant input features are compared with a true scalar field. In the contours of GKN without rotational invariant input features, erroneously predicted streaks of concentration tracer spread throughout the domain. Moreover for this case, two distinct streams of concentration tracer from upper and lower airfoil surfaces are not clearly distinguishable in the downstream flow. In comparison, the use of rotational invariant input features results in contours that are significantly accurate and imitate those of the true scalar field. These results show that the frame invariance can be achieved with the GKN by introducing rotational invariant input features and such a setup provides a promising candidate for frame-invariant neural operators.

To further highlight the significance of frame invariance for neural operators, GKN without rotational invariance is tested in the case of $\text{AOA}=35^\circ$, with input data that has been transformed/rotated with arbitrarily selected angles. Fig. 8 shows the scalar field τ contours for these results. Reference frame of input data is rotated to different angles (0° , 35° , 70° , 90° or 180°) and the corresponding results are compared with a true concentration field. As the reference frame of the input data is transformed with larger rotation

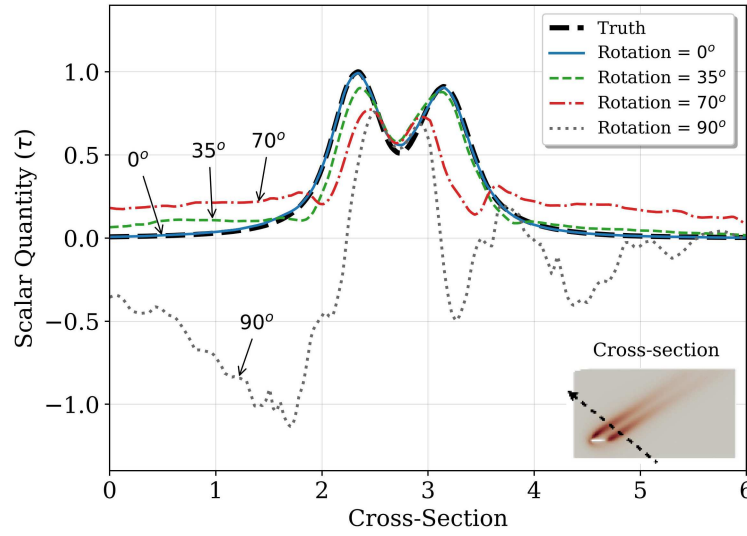


Figure 9: Comparison of profiles along the cross-section in the scalar field τ predicted by graph kernel network without rotational invariance with input data reference frame rotated by arbitrarily selected angles. The cross-section is shown in the inset, where the coordinates of starting and ending points are $(4, -1)$ and $(-1, 3)$, respectively. The arrow of the cross-section indicates the direction along which profiles are plotted. Profiles are normalized with respect to the maximum absolute value of the truth profile. Results correspond to the testing data at $\text{AOA}=35^\circ$.

angles, the predicted output by GKN without rotational invariance becomes more erroneous. For input data transformed to large rotation angles (90° and 180°), the predicted output (Fig. 8(e) and 8(f)) is significantly off-scale compared to true scalar field. In contrast, GKN with rotational invariance predicts accurate concentration field, invariant to the transformations applied to the input data.

These results are further analysed by plotting profiles of scalar quantity τ along a cross-section, as shown in Fig. 9. The cross-section line has been shown in the inset at the right bottom corner of Fig. 9, where the arrow indicates the direction along which profiles have been drawn. Profiles plotted along this cross-section show the discrepancies more recognizably. When no rotational transformation (0°) is applied to the input data, the profile agrees well with the truth data profile. As large rotational transformations are applied to the input data, the predicted output profile clearly deviates from the true data, with erroneous negative values for reference frame rotation of 90° . These results signify the importance of the frame-invariance property of the neural operators for scientific problems where differently aligned reference frames are used in different problem cases.

3.2 Comparison of GKN and VCNN

With two suitable candidates for frame-invariant neural operators solving scientific problems, we present here a comparison of their predictive performance and computational efficiency. The same training and testing dataset setups are used as in the earlier subsec-

Table 3: Predictive performance comparison of graph kernel network (GKN) and vector cloud neural network (VCNN). Rotational invariant (RI) input features are used for GKN. Results are computed for stencil size of 150. The training dataset comprises of flow data for three angles of attack (10° , 20° , 30°).

NN Model	Training Error	Test Error – Interpolation		Test Error – Extrapolation	
		AOA = 15°	AOA = 25°	AOA = 5°	AOA = 35°
GKN – with RI	0.33%	1.7%	1.4%	2.7%	2.6%
VCNN	0.31%	3.5%	3.2%	6.8%	5.3%

tion, i.e., a training dataset at three angles of attack (10° , 20° and 30°) and a testing dataset at four angles of attack (5° , 15° , 25° and 35°) with transformed coordinate systems equal to the respective flow angles of attack. Table 3 shows the comparison of predictive performance for GKN and VCNN. The results of both models are compared using the same stencil size (150), i.e., the same number of sampled data points in the nonlocal region of influence. Furthermore, the number of learnable hyperparameters for GKN and for the fitting network of VCNN are approximately equal (33000). Results in Table 3 show that the predictive performance of GKN is better than VCNN, despite both learning from the training dataset with comparable accuracy.

Reasons for this relatively inferior performance of VCNN are not fully comprehensible yet, despite both networks using similar information as input data. The difference in the predictive performance might be attributable to the processing of this information. A simple feed-forward network is used as a fitting network in VCNN for mapping the invariant input information to the scalar quantity τ . In GKN, the graph representation is updated for each node by *message passing* process through kernel function and the resulting node features are updated twice (equal to the depth of the network). With each graph update, features of all the nodes are updated based on the information from their corresponding neighbouring nodes in the graph. The scalar quantity τ is then predicted by global averaging operation over all the node features. The superior performance of GKN can be attributed to this intricate processing of information. Possible improvement in VCNN requires further investigation and will be a subject of future work.

Qualitative comparison of the predictions made by both models is shown in Fig. 10 and Fig. 11 for interpolating and extrapolating in test conditions, respectively. Contours of the scalar field τ for both models resemble quite well with those of true scalar field τ . As expected, more information generally leads to better predictive performance. This analysis is shown in Fig. 12 where the predictive error percentage for the testing dataset is plotted as a function of the stencil size. It can be observed that the predictive performance improves with increasing stencil size. For VCNN, the predictive performance shows a converging behavior with increasing stencil size. For GKN, predictive performance improves with increasing stencil size; however, no observation about the convergence can be made from the given results.

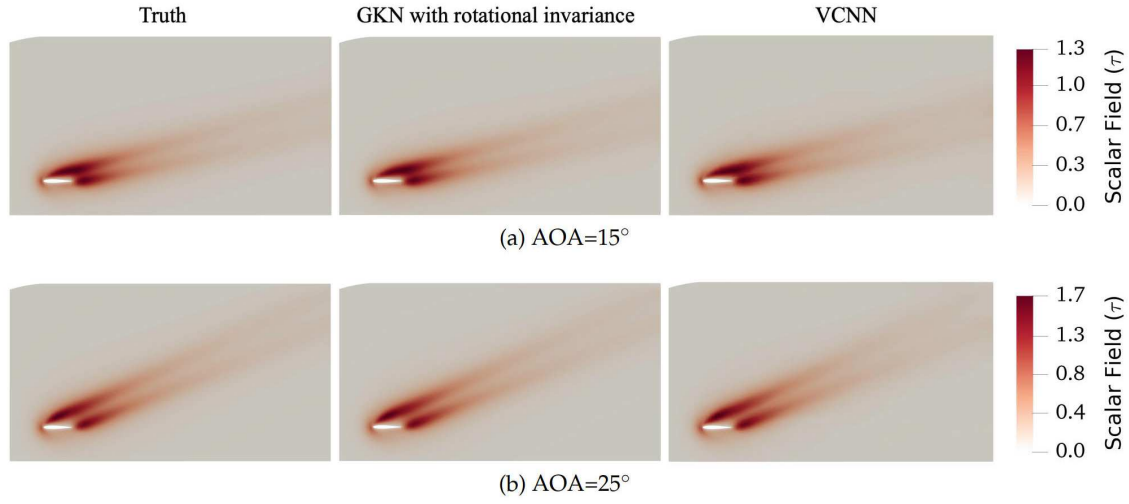


Figure 10: Scalar transport field (τ) contours for interpolating training data to testing data with respect to airfoil's angle of attack.

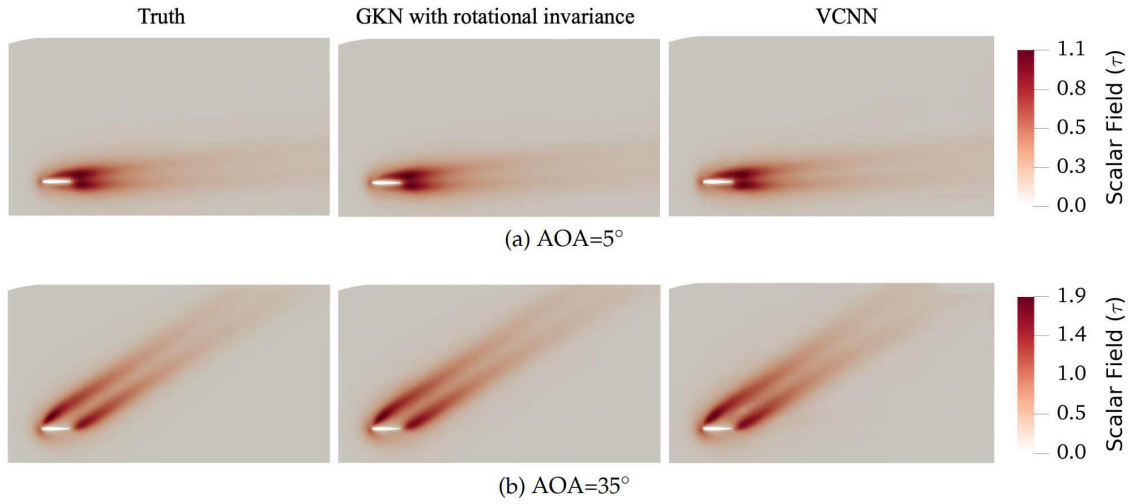


Figure 11: Scalar transport field (τ) contours for extrapolating training data to testing data with respect to airfoil's angle of attack.

In Fig. 12, although the results for VCNN are shown for the maximum stencil size of the 600, same are shown for GKN only until 225, because training GKN is prohibitively expensive with larger stencil size. The computational cost in terms of memory usage and training time presents a major disadvantage for graph kernel networks. For the scalar field dataset over an airfoil, a comparison is shown in Fig. 13, where the memory usage and training times are plotted with increasing stencil size. For both neural operators, the time cost is calculated based on the use of one NVIDIA V100 (“Volta”) GPU for training.

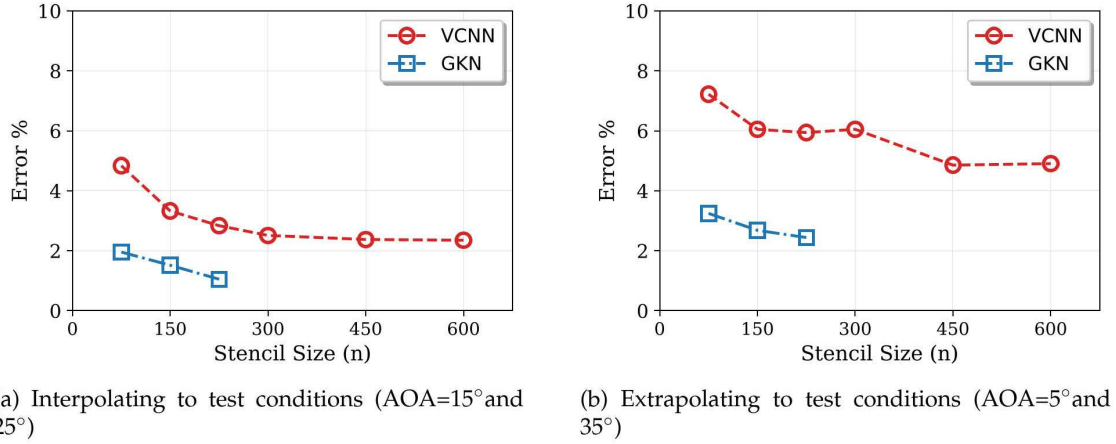


Figure 12: Variation of prediction error percentage at testing flow conditions (angles of attack in degrees). GKN for these results is used with rotational invariant input features.

As the stencil size increases, the data size for GKN quadratically increases as it involves arranging all the pairwise combinations of the nodes in the graph and the corresponding edge features. For each stencil size n , the number of edges has the order of $O(n^2)$. This quadratic increase in data size can also be observed in Fig. 13(a) where CPU memory required to pre-process the training dataset is shown with increasing stencil size. In comparison, the memory usage for VCNN increases linearly. With the same stencil size, memory usage for GKN is approximately two orders of magnitude higher than that of VCNN. For the stencil size of 225, the required CPU memory easily exceeds 100 GB to pre-process the training dataset for GKN, compared to the requirement of 0.3 GB for VCNN. Such an increase in data size is reflected in the training time cost of GKN as well. For VCNN, the linear increase in data size is offset by the parallel computing of GPU. However, the quadratic increase in data size for GKN exceeds the parallel computing capacity of the GPUs, leading to roughly similar order ($\sim O(n^{1.8})$) of increase in training time. With increasing stencil size, the memory requirement and time cost become prohibitively expensive for GKN, which presents a significant disadvantage in comparison with VCNN. On the flip side, for a very small stencil size ($n < 75$), the training time cost of GKN is comparable to that of VCNN, albeit still more expensive in memory. However, such a stencil size range, illustrated by the shaded gray region in Fig. 13, is not a realistic cloud size for many applications in computational physics.

Beyond the CPU memory cost requirement for pre-processing the data, training of GKN using GPUs also presents a challenging requirement for GPU memory. For the current research work, the training dataset was distributed into large number of batches (800) and each batch is moved to GPU for training one by one due to the limitation of GPU memory. With increasing graph size (stencil size), the requirement would be to distribute the training dataset in an even larger number of batches. For many scientific problems, the graph size could be significantly larger than the current problem, which could lead to

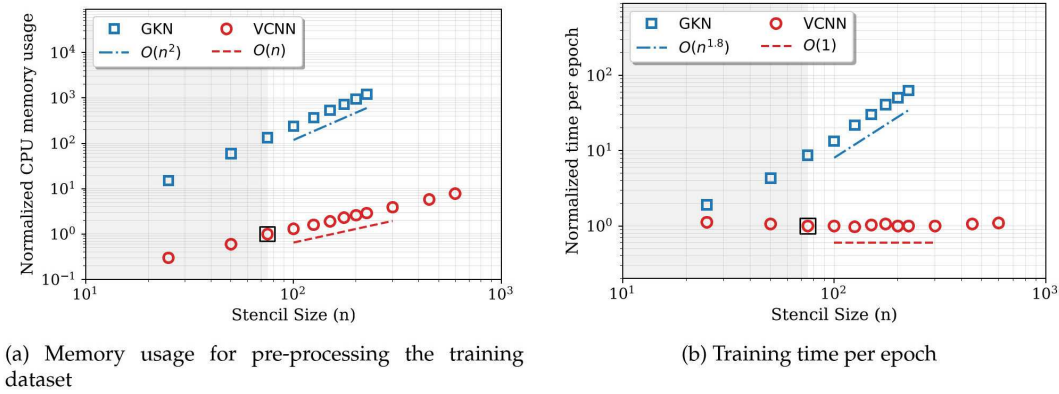


Figure 13: Comparison of normalized computational cost for GKN and VCNN. Memory and time costs are normalized with respect to baseline cost of VCNN for $n=75$ (indicated in black squares), which is 0.1 gigabytes for the memory usage and 3.3 sec for time per epoch. Gray shaded region indicates small stencil size range which might be unrealistic for intended applications in computational physics. Slope lines illustrate the order of increase in cost with increasing stencil size. The reduction of observed time complexity from memory complexity is mainly due to the parallel computing capacity of GPU.

a situation where GPU memory requirement for even one graph could prove excessive. Such issues of handling large memory costs for graph kernel networks is in itself a topic of research [44–47]. In this regard, VCNN provides significantly efficient neural operators for scientific problems.

We remark that in [29] the authors proposed to randomly sample m times sub-graphs with node size n from the region to build input–output data pairs. In their cases, the authors observe that parameters $n=200$ and $m=4$ give a good balance between prediction accuracy and computation cost. In the present work, we have chosen $n=150$ and $m=1$. The general guideline for the choice of n and m in GKN still remains elusive and requires further investigation. Meanwhile, the idea of sampling smaller sub-graphs can also be employed by VCNN to improve the training efficiency; see the discussion in [48].

4 Conclusion

Frame-invariant neural operators for transport partial differential equations (PDEs) have been analyzed. Transport of a scalar quantity in a flow over an airfoil is considered to analyze the frame invariance properties of both neural operators. First, graph kernel network (GKN) model is considered which leverages the structure and properties of graphs to embed nonlocal dependencies. We propose to use invariant input features for the GKN model to guarantee frame invariance. Secondly, we have considered vector-cloud neural network (VCNN) which has both nonlocal dependencies and frame invariance embedded in the model.

GKN with invariant input features has shown to achieve frame invariance, which is demonstrated for the testing datasets with transformed reference frames. VCNN also

demonstrated invariance to transformation in the reference frame; however it showed slightly worse predictive performance as compared to GKN, given the same input data and around same number of learnable parameters. With increasing stencil size, both models showed improved predictive performance. However, as graph-based networks are intrinsically expensive to train, GKN becomes prohibitively expensive for larger stencil size. In comparison, VCNN demonstrated orders of magnitude better computational efficiency for increasing stencil size. This observation is critical for many scientific problems where large mesh sizes and possibly larger number of feature vectors are used. For such cases, using GKN becomes infeasible and VCNN can be used efficiently as a surrogate model for transport PDEs with embedded invariance properties and nonlocal dependencies.

Despite the satisfactory performance and computational efficiency of VCNN, a future study is required to address its slightly inferior performance to GKN. Since GKN uses relatively intricate processing of the invariant input data, different options can be explored for the fitting network of VCNN to enhance model accuracy.

Acknowledgments

H. Xiao is supported by the U.S. Air Force under agreement number FA865019-2-2204. The U.S. Government is authorised to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The computational resources used for this project were provided by the Advanced Research Computing (ARC) of Virginia Tech, which is gratefully acknowledged.

Appendices

A Performance of an improved VCNN

In this appendix we consider a variant of VCNN for potential performance improvement. Pairwise projection QQ^\top among feature vectors is used to determine the invariant feature matrix \mathcal{D} (Eq. (2.6b)). Input feature matrix Q can also be written as a stack of three sub-matrices as:

$$Q = [\mathcal{X} \quad \mathcal{U} \quad \mathcal{C}], \quad (\text{A.1})$$

where sub-matrices \mathcal{X} , \mathcal{U} and \mathcal{C} comprises of relative spatial coordinates, velocity vectors and scalar quantities of the point cloud, respectively. Pairwise projection can then be represented as:

$$QQ^\top = \mathcal{X}\mathcal{X}^\top + \mathcal{U}\mathcal{U}^\top + \mathcal{C}\mathcal{C}^\top. \quad (\text{A.2})$$

Table 4: Comparison of predictive performances of two VCNN architectures along with GKN. VCNN ($\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$) retains more information in input feature matrix $\mathcal{D}=[\mathcal{D}_1 \mathcal{D}_2 \mathcal{D}_3]^\top$ compared to the VCNN (\mathcal{D}) architecture (Fig. 5).

NN Model	Training Error	Test Error – Interpolation		Test Error – Extrapolation	
		AOA = 15°	AOA = 25°	AOA = 5°	AOA = 35°
GKN – with RI	0.33%	1.7%	1.4%	2.7%	2.6%
VCNN (\mathcal{D})	0.31%	3.5%	3.2%	6.8%	5.3%
VCNN ($\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$)	0.20%	2.3%	2.1%	5.0%	5.2%

To retain more information in the invariant feature matrix \mathcal{D} , the pairwise projection has been segregated in order to determine three different invariant feature matrices as:

$$\mathcal{D}_1 = \frac{1}{n^2} \mathcal{G}^\top \mathcal{X} \mathcal{X}^\top \mathcal{G}, \quad (\text{A.3})$$

$$\mathcal{D}_2 = \frac{1}{n^2} \mathcal{G}^\top \mathcal{U} \mathcal{U}^\top \mathcal{G}, \quad (\text{A.4})$$

$$\mathcal{D}_3 = \frac{1}{n} \mathcal{G}^\top \mathcal{C}, \quad (\text{A.5})$$

where \mathcal{D}_1 and \mathcal{D}_2 are based on the pairwise projection of spatial coordinates and velocity vectors, respectively. Three components ($\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$) are then concatenated to determine invariant features (replacing the role of \mathcal{D} in the original VCNN), which are then mapped to the scalar quantity τ through the fitting network. Such form closely resembles the determination of rotational invariant edge features (2.4) of GKN as well.

Since more information is being kept in invariant features with larger dimensions, the number of learnable parameters of the fitting network significantly increases (~ 123000) compared to the earlier presented VCNN with ~ 33000 parameters. Despite the near four-fold increase in model size, the training time per epoch increases only to 4.1s from 3.3s. Based on such representation of invariant feature matrix, results have been compared in Table 4. The reduction of prediction error percentage for training and interpolating the testing datasets can be attributed to the increased number of learnable parameters. However, extrapolating testing error percentage is comparable to that of the previously discussed architecture of VCNN.

B Neural network architectures details

In this appendix, architecture details of both neural network models are given. Details of hyperparameters for embedding and fitting networks in VCNN are mentioned in Table 5. To make a fair comparison between both neural network models, we have kept the total number of learnable parameters almost equal for both GKN and VCNN. Since

Table 5: Architecture details of vector cloud neural network (Fig. 5), which comprises of two neural networks; the embedding network and the fitting network. Neurons in each layer are mentioned in sequence, from the input layer to the hidden layers (if any) and the output layer. The numbers of neurons in the input and output layers are highlighted in bold. Note that the embedding network operates identically on the scalar features $c \in \mathbb{R}^{l'}$ associated with each point in the cloud and output a row of m elements in matrix \mathcal{G} .

	Embedding network	Fitting network ($\mathcal{D} \mapsto \tau$)
No. of input neurons	$l' = 7$	$m \times m' = 256 (\mathcal{D} \in \mathbb{R}^{64 \times 4})$
No. of hidden layers	5	1
No. of output neurons	$m = 64$	1 ($\tau \in \mathbb{R}$)
Neurons in each layer	(7, 32, 64, 64, 64, 64, 64)	(256 , 128, 1)
Activation functions	ReLU	ReLU, Linear (last layer)
No. of trainable parameters	19008	33025

Table 6: Architecture details of graph kernel network (Fig. 5). Neurons in each layer are mentioned in sequence, from the input layer to the hidden layers (if any) and the output layer. The numbers of neurons in the input and output layers are highlighted in bold. Kernel network operates identically on the invariant edge features $e(q_i, q_j) \in \mathbb{R}^d$ associated with each pair of nodes/points in the cloud to output a kernel matrix \mathcal{K}_{ij} .

	Kernel network
No. of input neurons	$d = 16$
No. of hidden layers	$L = 2$
No. of output neurons	$m \times m = 256 (\mathcal{K}_{ij} \in \mathbb{R}^{16 \times 16})$
Neurons in each layer	(16 , 64, 96, 256)
Activation functions	ReLU, Linear (last layer)
No. of trainable parameters	32577

nonlocal mapping from a point cloud or a graph to a scalar quantity τ is performed by the fitting network for VCNN, only the fitting network of VCNN has been considered for equivalency of total learnable parameters. The Adam optimizer [49] is adopted to train both of the neural networks. The neural network architectures have been implemented in the machine learning framework PyTorch, while also using PyTorch Geometric for implementing GKN.

References

- [1] A. Quarteroni, G. Rozza, Reduced Order Methods for Modeling and Computational Reduction, Springer, Heidelberg, 2014.
- [2] D. J. Lucia, P. S. Beran, W. A. Silva, Reduced-order modeling: New approaches for computational physics, Progress in Aerospace Sciences 40 (1) (2004) 51–117.
- [3] B. Peherstorfer, K. Willcox, Data-driven operator inference for nonintrusive projection-based model reduction, Computer Methods in Applied Mechanics and Engineering 306 (2016) 196–215.

- [4] K. Taira, M. S. Hemati, S. L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S. T. M. Dawson, C.-A. Yeh, Modal analysis of fluid flows: Applications and outlook, *AIAA Journal* 58 (3) (2020) 998–1022.
- [5] L. Lu, P. Jin, G. Pang, Z. Zhang, G. E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, *Nature Machine Intelligence* 3 (2021) 218–229.
- [6] S. Wang, H. Wang, P. Perdikaris, Learning the solution operator of parametric partial differential equations with physics-informed DeepONets, *Science Advances* 7 (40) (2021) eabi8605.
- [7] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, in: *International Conference on Learning Representations*, 2021.
- [8] X.-H. Zhou, J. Han, H. Xiao, Frame-independent vector-cloud neural network for nonlocal constitutive modeling on arbitrary grids, *Computer Methods in Applied Mechanics and Engineering* 388 (2022) 114211.
- [9] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, A. Anandkumar, Neural operator: Learning maps between function spaces, in: *AMS Fall Western Virtual Sectional Meeting*, 2021.
- [10] O. C. Zienkiewicz, R. L. Taylor, P. Nithiarasu, J. Zhu, *The Finite Element Method*, Vol. 3, McGraw-Hill London, 1977.
- [11] G. Rozza, D. B. P. Huynh, A. T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations, *Archives of Computational Methods in Engineering* 15 (3) (2008) 229–275.
- [12] D. Gross, The role of symmetry in fundamental physics, *Proceedings of the National Academy of Sciences* 93 (25) (1996) 14256–14259. doi:10.1073/pnas.93.25.14256.
- [13] P. R. Spalart, Philosophies and fallacies in turbulence modeling, *Progress in Aerospace Sciences* 74 (2015) 1–15.
- [14] J. Ling, R. Jones, J. Templeton, Machine learning strategies for systems with invariance properties, *Journal of Computational Physics*.
- [15] J. Ling, R. Jones, J. Templeton, Machine learning strategies for systems with invariance properties, *Journal of Computational Physics* 318 (2016) 22–35.
- [16] W. E, J. Han, L. Zhang, Machine-learning-assisted modeling, *Physics Today* 74 (7) (2021) 36–41.
- [17] M. L. Kaandorp, R. P. Dwight, Data-driven modelling of the Reynolds stress tensor using random forests with invariance, *Computers & Fluids* 202 (2020) 104497.
- [18] H. Frezat, G. Balarac, J. Le Sommer, R. Fablet, R. Lguensat, Physical invariance in neural networks for subgrid-scale scalar flux modeling, *Phys. Rev. Fluids* 6 (2021) 024607.
- [19] R. Wang, R. Walters, R. Yu, Incorporating symmetry into deep dynamics models for improved generalization, in: *International Conference on Learning Representations*, 2021.
- [20] J.-X. Wang, J.-L. Wu, H. Xiao, Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data, *Physical Review Fluids* 2 (3) (2017) 034603.
- [21] J.-L. Wu, H. Xiao, E. G. Paterson, Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework, *Physical Review Fluids* 3 (2018) 074602.
- [22] Wikipedia, Flap (aeronautics), [https://en.wikipedia.org/wiki/Flap_\(aeronautics\)](https://en.wikipedia.org/wiki/Flap_(aeronautics)), [Online; accessed 13-April-2022] (2022).

- [23] Wikipedia, Turbofan, <https://en.wikipedia.org/wiki/Turbofan>, [Online; accessed 13-April-2022] (2022).
- [24] M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational Physics* 378 (2019) 686–707.
- [25] H. Gao, M. J. Zahr, J. Wang, Physics-informed graph neural galerkin networks: A unified framework for solving pde-governed forward and inverse problems, *arXiv preprint arXiv:2107.12146*.
- [26] X.-H. Zhou, J. Han, H. Xiao, Learning nonlocal constitutive models with neural networks, *Computer Methods in Applied Mechanics and Engineering* 384 (2021) 113927.
- [27] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, P. W. Battaglia, Learning mesh-based simulation with graph networks, in: *International Conference on Learning Representations*, 2021.
- [28] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, *IEEE Transactions on Neural Networks* 20 (1) (2009) 61–80.
- [29] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Neural operator: Graph kernel network for partial differential equations, *arXiv preprint arXiv:2003.03485*.
- [30] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, A. Stuart, K. Bhattacharya, A. Anandkumar, Multipole graph neural operator for parametric partial differential equations, in: *Advances in Neural Information Processing Systems*, Vol. 33, 2020, pp. 6755–6766.
- [31] C. Lyle, M. van der Wilk, M. Kwiatkowska, Y. Gal, B. Bloem-Reddy, On the benefits of invariance in neural networks (2020).
- [32] R. Q. Charles, H. Su, M. Kaichun, L. J. Guibas, PointNet: Deep learning on point sets for 3D classification and segmentation, in: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 77–85.
- [33] X. Sun, Z. Lian, J. Xiao, Srinet: Learning strictly rotation-invariant representations for point cloud classification and segmentation, in: *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 980–988.
- [34] A. Kashefi, D. Rempe, L. J. Guibas, A point-cloud deep learning framework for prediction of fluid flow fields on irregular geometries, *Physics of Fluids* 33 (2) (2021) 027104.
- [35] D. C. Wilcox, *Turbulence Modeling for CFD*, 3rd Edition, DCW Industries, 2006.
- [36] UIUC Applied Aerodynamics Group, UIUC airfoil data site, https://m-selig.ae.illinois.edu/ads/coord_database.html, [Online; accessed 23-February-2021] (2020).
- [37] H. Xiao, P. Cinnella, Quantification of model uncertainty in RANS simulations: A review, *Progress in Aerospace Sciences* 108 (2019) 1–31.
- [38] J. Stokes, K. Yang, K. Swanson, W. Jin, A. Cubillos-Ruiz, N. Donghia, C. MacNair, S. French, L. Carfrae, Z. Bloom-Ackermann, V. Tran, A. Chiappino-Pepe, A. Badran, I. Andrews, E. Chory, G. Church, E. Brown, T. Jaakkola, R. Barzilay, J. Collins, A deep learning approach to antibiotic discovery, *Cell* 181 (2) (2020) 475–483.
- [39] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, P. Battaglia, Learning to simulate complex physics with graph networks, in: *Proceedings of the 37th International Conference on Machine Learning*, Vol. 119 of *Proceedings of Machine Learning Research*, PMLR, 2020, pp. 8459–8468.
- [40] F. Monti, F. Frasca, D. Eynard, D. Mannion, M. M. Bronstein, Fake news detection on social media using geometric deep learning, *arXiv preprint arXiv:1902.06673*.
- [41] W. Jiang, J. Luo, Graph neural network for traffic forecasting: A survey, *arXiv preprint arXiv:2101.11174*.

- [42] C. Eksombatchai, P. Jindal, J. Z. Liu, Y. Liu, R. Sharma, C. Sugnet, M. Ulrich, J. Leskovec, Pixie: A system for recommending 3+ billion items to 200+ million users in real-time, International World Wide Web Conferences Steering Committee, 2018, pp. 1775–1784.
- [43] M. I. Zafar, J. Han, X.-H. Zhou, H. Xiao, Frame invariance and scalability of neural operators for partial differential equations, https://github.com/muhammadirfanzafar/pde_neural_operators (2022).
- [44] J. Chen, T. Ma, C. Xiao, FastGCN: Fast learning with graph convolutional networks via importance sampling, in: International Conference on Learning Representations, 2018.
- [45] M. Fey, J. E. Lenssen, Fast graph representation learning with PyTorch Geometric, in: ICLR 2019 Workshop on Representation Learning on Graphs and Manifolds, 2019.
- [46] Z. Lin, C. Li, Y. Miao, Y. Liu, Y. Xu, Paragraph: Scaling GNN training on large graphs via computation-aware caching, in: Proceedings of the 11th ACM Symposium on Cloud Computing, SoCC '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 401–415.
- [47] Z. Jia, S. Lin, M. Gao, M. Zaharia, A. Aiken, Improving the accuracy, scalability, and performance of graph neural networks with roc, in: I. Dhillon, D. Papailiopoulos, V. Sze (Eds.), Proceedings of Machine Learning and Systems, Vol. 2, 2020, pp. 187–198.
- [48] J. Han, X.-H. Zhou, H. Xiao, VCNN-e: A vector-cloud neural network with equivariance for emulating Reynolds stress transport equations, arXiv preprint arXiv:2201.01287.
- [49] D. Kingma, J. Ba, Adam: a method for stochastic optimization, in: Proceedings of the International Conference on Learning Representations, 2015.

Chapter 6

Conclusions and Future Work

This dissertation explored machine learning approaches to data-driven transition modeling. Predicting the onset of transition due to Tollmien–Schlichting instabilities is the primary focus of the first half of this work. Linear stability theory along with the e^N method is a widely used *physics-based* approach for this purpose, particularly in aircraft design computations. To ensure strong correlation with the underlying transition physics, neural network-based models are developed using data from linear stability computations. Even though these models have only been demonstrated for transition due to Tollmien-Schlichting instabilities in 2D, low-speed boundary layers, the capability of convolutional neural network to automatically extract features from boundary layer profiles should enable a generalization of the proposed model to other instability mechanisms. Such capability has already been demonstrated for hypersonic flows over a circular cone with slightly blunted nosetip [1]. Furthermore, an end-to-end prediction model based on recurrent neural network provides a simplified workflow which is amenable to non-expert users, as compared to stability-based methods which require expertise in stability theory on user’s part. The end-to-end prediction of transition-onset location based on the recurrent neural network are easily three to four orders of magnitude faster compared to direct computations based on the linear stability theory. Such benefits in terms of efficiency, accuracy, and generalizability demonstrate the promising capability of such machine learning based models.

In the second half of this work, neural operators are examined for their computational complexity and invariance properties. Neural operators can provide data-driven learning of nonlinear functionals which can approximate the solution to complex problems for given input functions. Such models can be very useful where the underlying governing equations are not explicitly known. In the context of transition modeling, neural operators can improve on the PDE-based transition models which involve significant empiricism. Neural operator-based transition models can be used to provide a nonlocal mapping from the underlying mean flow to the transition closure quantity. Furthermore, the neural operators are adaptive to different grid resolutions and discretizations, and hence, such transition model can be used with general purpose RANS solvers.

The rest of this section discusses possible future extensions to this work and the associated challenges.

6.1 Machine Learning for Stability-based Methods

The proposed transition model can serve as a surrogate model that incorporates the underlying physics from linear stability theory, without requiring time- as well as labor-intensive computations based on the direct solution of the underlying governing equations. However, since the model has no intrinsic dependence on the linear stability theory, the methodology can be extended to other physics-based methods like parabolized stability equations (PSEs) to overcome the limitation of quasiparallel flow assumption from linear stability theory. This could be a fruitful avenue for future work, especially since the nonlinear PSE take into account linear as well as nonlinear growth of instability waves.

Integration of the RNN model in coupled framework with a RANS solver is another future endeavor. Sample frameworks for such application are presented in recent publications [2, 3] where stability-based computations are performed to determine the transition onset location. Application of neural network models can provide significant cost savings in such frameworks while providing reliable predictions. RNN model also needs to be evaluated for instabilities involving planar boundary layer profiles (with both wall-normal and spanwise components) and three-dimensional boundary layer profiles (additional crossflow velocity component for flows over swept wings).

6.2 Neural Operators for PDE-based Transition Models

Development of neural network models for transition closure can leverage several practices from the research on turbulence modeling where closure model for Reynolds stress tensor needs to be developed. PDE-based transition models require the solution of auxiliary transport equations for transition closure variables like intermittency (γ), amplification factor (\tilde{n}), etc. Neural operators can be used to learn nonlocal mapping from mean flow field to such closure variables, which is in effect approximating the solution to the PDE for given underlying flow. Similar approximation using a neural operator has already been performed for Reynolds stress transport model [4].

For training of such neural operators, lack of full field data for transition closure variables can pose a challenge and would constrain the training cases to those for which high-fidelity simulations are performed. Furthermore, training of a transition model directly using data of closure variable in an *a priori* manner can lead to inaccurate predictions of the mean flow in *a posteriori* tests (when trained transition model is integrated with RANS solver). Similar behavior has been observed in turbulence modeling case as well, where even small errors in predicted Reynolds stress can lead to dramatic errors in predicted mean flow field [5]. To overcome these challenges, model-consistent-training methods [6, 7, 8] can be used, where indirect observation data (e.g. pressure, sparse velocity, lift coefficient etc.) is leveraged to

train the closure model by integrating RANS solvers in the training process. Such indirect data are usually available from experimental measurements. In this regards, recently proposed ensemble Kalman method for learning neural network parameters provides a promising approach [8], which has significantly simpler implementation as compared to adjoint-based methods.

Bibliography

- [1] Pedro Paredes, Balaji Venkatachari, Meelan M. Choudhari, Fei Li, Chau-Lyan Chang, Muhammad I. Zafar, and Heng Xiao. Toward a practical method for hypersonic transition prediction based on stability correlations. *AIAA Journal*, 58(10):4475–4484, 2020. doi: 10.2514/1.J059407.
- [2] Balaji Shankar Venkatachari, Jared Carnes, Chau-Lyan Chang, and Meelan M. Choudhari. Boundary-layer transition prediction through loose coupling of overflow and lastrac. 2022. doi: 10.2514/6.2022-3682.
- [3] Nathaniel J. Hildebrand, Chau-Lyan Chang, Meelan M. Choudhari, Fei Li, Eric J. Nielsen, Balaji Shankar Venkatachari, and Pedro Paredes. Coupling of the fun3d unstructured flow solver and the lastrac stability code to model transition. 2022. doi: 10.2514/6.2022-1952.
- [4] Jiequn Han, Xu-Hui Zhou, and Heng Xiao. An equivariant neural operator for developing nonlocal tensorial constitutive models. 2022.
- [5] Jinlong Wu, Heng Xiao, Rui Sun, and Qiqi Wang. Reynolds-averaged navier–stokes equations with explicit data-driven reynolds stress closure can be ill-conditioned. *Journal of Fluid Mechanics*, 869:553–586, 2019. doi: 10.1017/jfm.2019.205.
- [6] Carlos A. Michel’en Strofer and Heng Xiao. End-to-end differentiable learning of turbulence models from indirect observations. *Theoretical and Applied Mechanics Letters*, 11: 100280, 2021.
- [7] Carlos A. Michel’en Strofer, Xinlei Zhang, and Heng Xiao. Ensemble gradient for learning turbulence models from indirect observations. *Commun. Comput. Phys.*, 30:1269–1289, 2021.
- [8] Xin-Lei Zhang, Heng Xiao, Xiaodong Luo, and Guowei He. Ensemble kalman method for learning turbulence models from indirect observation data. *Journal of Fluid Mechanics*, 949:A26, 2022. doi: 10.1017/jfm.2022.744.