

# Learning neural-network-based turbulence models for external transonic flows using ensemble Kalman method

Yi Liu <sup>\*</sup>, Xin-Lei Zhang <sup>†</sup>, and Guowei He <sup>‡</sup>

*State Key Laboratory of Nonlinear Mechanics, Chinese Academy of Sciences, 100190 Beijing, People's Republic of China*

*University of Chinese Academy of Sciences, 100049 Beijing, People's Republic of China*

This paper presents a neural network-based turbulence modeling approach for transonic flows based on the ensemble Kalman method. The approach adopts a tensor basis neural network for the Reynolds stress representation, with modified inputs to consider fluid compressibility. The normalization of input features is also investigated to avoid feature collapsing in the presence of shock waves. Moreover, the turbulent heat flux is accordingly estimated with the neural network-based turbulence model based on the gradient diffusion hypothesis. The ensemble Kalman method is used to train the neural network with the experimental data in velocity and wall pressure due to its derivative-free nature. The proposed framework is tested in two canonical configurations, i.e., 2D transonic flows over the RAE2822 airfoils and 3D transonic flows over the ONERA M6 wings. Numerical results demonstrate the capability of the proposed method in learning accurate turbulence models for external transonic flows.

## Nomenclature

$a$	=	speed of sound
$C_f$	=	friction coefficient
$C_p$	=	pressure coefficient
$c_p$	=	specific heat coefficient at constant pressure
$C$	=	chord length
$E$	=	total energy
$g$	=	coefficient of tensor basis
$H$	=	total enthalpy
$\mathbf{I}$	=	identity matrix
$k$	=	turbulent kinetic energy
$L$	=	reference length

<sup>\*</sup>Postdoctoral Research Associate, LNM, Institute of Mechanics; liuyi@imech.ac.cn

<sup>†</sup>Postdoctoral Research Associate, LNM, Institute of Mechanics; zhangxinlei@imech.ac.cn (Corresponding author).

<sup>‡</sup>Professor, LNM, Institute of Mechanics; hgw@lnm.imech.ac.cn (Corresponding author).

$M$	=	number of samples
$Ma$	=	Mach number
$Ma_n$	=	normal Mach number
$p$	=	static pressure
$\mathbf{P}$	=	model error covariance
$Pr$	=	Prandtl number
$q_j, q_j^{(t)}$	=	laminar heat flux and turbulent heat flux
$\mathbf{R}$	=	observation error covariance
$Re$	=	Reynolds number
$\mathbf{S}, \mathbf{\Omega}$	=	strain-rate tensor and rotation-rate tensor
$\hat{\mathbf{S}}, \hat{\mathbf{\Omega}}$	=	non-dimensionalized strain-rate tensor and rotation-rate tensor
$T$	=	static temperature
$T_\infty$	=	temperature at far-field
$\mathbf{T}$	=	tensor basis
$U_\infty$	=	free-stream velocity
$\mathbf{u}$	=	velocity vector
$w$	=	weights of neural networks
$W$	=	ensemble of weights, $W = \{w\}_{m=1}^M$
$x$	=	Cartesian coordinates
$y$	=	training data
$y^+$	=	nondimensional wall distance
$\alpha$	=	angle of attack
$\beta^*$	=	turbulence model constants
$\mathcal{H}$	=	model operator that maps the neural network weights to the observed quantities
$\mu, \mu_t$	=	dynamic viscosity and turbulent viscosity
$\theta$	=	scalar invariants
$\rho$	=	fluid density
$\sigma$	=	viscous stress
$\tau$	=	Reynolds stress
$\omega$	=	specific dissipation rate of turbulent kinetic energy
$\nabla$	=	gradient operator
$\  * \ $	=	L2 norm

$\text{tr}\{*\}$  = trace operator

#### Superscripts

$(i)$  = index of tensor basis

$n$  = index of learning iteration

$\top$  = transpose

$0$  = prior

#### Subscripts

$i, j, k$  = 1, 2, 3, tensor indices

$m$  = index of sample

$n$  = normal component of vector

$t$  = turbulent

$MAC$  = mean aerodynamic chord

$RSM$  = Reynolds-stress transport model

$\infty$  = at infinity

## I. Introduction

Accurate prediction of transonic flows is of significant interest for prior assessments of buffet onset, which further concerns the safety design of flight vehicles [1, 2]. The Reynolds averaged Navier–Stokes (RANS) simulation with turbulence models is still the workhorse tool for predicting such flows with high Reynolds numbers due to its low computational overhead [3]. Therefore, it is necessary to develop accurate turbulence models for transonic flows.

The most widely used turbulence models are linear eddy viscosity models, such as Spalart–Allmaras model [4] and  $k-\omega$  SST model [5]. These models assume that the Reynolds stress is linearly related to the strain rate via turbulent eddy viscosity based on the Boussinesq assumption. They can provide satisfactory predictions for attached flows but often lead to large discrepancies when encountered with the mean curvature and separation [6, 7]. For transonic flows, there often occur turbulent heat transfer and shock waves, which would lead to additional difficulties in turbulence modeling. Specifically, the turbulent heat transfer requires closure modeling for turbulent heat flux, which is usually estimated based on the gradient diffusion hypothesis. That is, the turbulent heat flux is proportional to the temperature gradient, while the proportional coefficient involves the turbulent eddy viscosity obtained with turbulence models. Hence the turbulence model has to be used for estimating the Reynolds stress as well as turbulent heat flux. Moreover, the transonic flows would produce shock waves, which often interact with the boundary layer and lead to flow separations. Estimating such shock-induced separation requires capturing the interaction between the shock wave and boundary layer, which poses another challenge for turbulence models. Some advanced turbulence models that go beyond the Boussinesq

assumption, e.g., the Reynolds stress transport model, can provide satisfactory predictions for transonic flows but encounter the robustness issue and increased computational cost [8]. The compromise between predictive accuracy and practical usability is often intractable for turbulence modeling.

Machine learning methods [9–11] have been increasingly used to construct high-fidelity turbulence models due to their ability of discovering complicated functional relationships from data. For instance, the tensor basis neural network [12] and the symbolic expression [13] are proposed to represent the Reynolds stress in an algebraic form based on the weak-equilibrium assumption. Also, the vector-cloud neural network [14] is proposed to emulate the Reynolds stress transport equations with embedded non-equilibrium effects. However, these methods usually require full field data of Reynolds stresses to determine the model parameters, i.e., the weights of neural networks. These Reynolds stress data can be categorized as direct data for learning turbulence models. This is in contrast to the indirect data [15] such as velocity and wall pressure, which need to be propagated from the Reynolds stress via the RANS equation. Learning from the direct data would cause poor posterior prediction due to the inconsistency between training and prediction environments [16]. In addition, the full field data of Reynolds stress is difficult to be obtained for transonic flows at high Reynolds numbers. In view of these difficulties, learning turbulence models from indirect data becomes trending since it can reduce the data requirement and avoid the inconsistency issue simultaneously. Such a strategy is often referred to as model-consistent training [16].

Model-consistent training ensures consistency between the training and prediction environments by involving RANS equations in the training process. Different strategies, such as adjoint-based learning [15, 17, 18], ensemble-based learning [19, 20], and genetic programming [21], have been proposed but are mostly applied for low-speed incompressible flows. Over the past few years, the genetic programming method has been increasingly used to learn turbulence models for transonic flows in a model-consistent manner due to its gradient-free merits. That is, the gradient evaluation is not needed to minimize the discrepancies between the training data and model predictions as the adjoint-based method. This method has been used to construct algebraic Reynolds stress models in symbolic forms for the wake of turbine cascade [21] and the separation-induced transition [22] in turbomachines. In contrast to the gradient-free method, the ensemble Kalman method is a gradient-approximation method, which estimates the model gradient and Hessian based on the statistics of random samples. By using the approximated gradient and Hessian information, such methods are able to achieve good training efficiency and also do not require developing adjoint solvers. The ensemble Kalman method is recently used to learn neural network-based turbulence models from indirect data [20] and shows superior efficiency and considerable training accuracy compared to the adjoint-based learning method. The ensemble-based method has only been demonstrated for incompressible flows, and its application for transonic flows is worthy of further investigation.

The present work aims to develop the neural network-based turbulence modeling framework for transonic flows based on the ensemble Kalman method. Most neural network-based turbulence models are constructed for low-speed flows. Transonic flows have different flow features from low-speed flows due to fluid compressibility, shock waves,

and heat transfer, which would require specialized treatments for the neural network-based representation of Reynolds stresses. In this work, the compressibility effects are taken into account in selecting the input features of neural networks. Further, the scaling strategy of input features is investigated to avoid feature collapse due to the presence of shock waves. Moreover, the heat flux closure is modified with the outputs of neural networks based on the Boussinesq hypothesis. The ensemble Kalman method is used to train the neural network-based model with good training efficiency and flexibility in data usage. Finally, the method is tested for transonic flows over the ONERA M6 wing, demonstrating its applicability in 3D flow applications.

The rest of the paper is outlined as follows. The tensor-basis neural network for transonic flows and the ensemble-based training algorithm are introduced in Section II. The test cases and corresponding results are presented and analyzed in Section III. Finally, this paper is concluded in Section IV.

## II. Ensemble-based turbulence modeling for transonic flows

The Reynolds-averaged Navier-Stokes (RANS) equations for compressible flows can be written as [8]

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_j)}{\partial x_j} = 0, \quad (1a)$$

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial \sigma_{ij}}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_j}, \quad (1b)$$

$$\frac{\partial(\rho E)}{\partial t} + \frac{\partial}{\partial x_j}(\rho H u_j) = \frac{\partial(\tau_{ij} u_i)}{\partial x_j} + \frac{\partial(\sigma_{ij} u_i)}{\partial x_j} - \frac{\partial q_j}{\partial x_k} - \frac{\partial q_j^{(t)}}{\partial x_j}. \quad (1c)$$

In the formula above,  $\rho$  and  $p$  represent the density and pressure, respectively,  $u_i$  is the flow velocity component,  $\tau_{ij}$  is the Reynolds stress tensor,  $\sigma_{ij}$  represents the viscous stress, which is described by

$$\sigma_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right),$$

and  $\mu$  is the dynamic viscosity. In the energy equation (1c),  $E$  and  $H$  denote the total energy and total enthalpy,  $q_j^{(t)}$  is the turbulent heat flux, and  $q_j$  is the laminar heat flux. The heat flux  $q_j$  is calculated by

$$q_j = -\frac{c_p \mu}{Pr} \frac{\partial T}{\partial x_j},$$

where  $T$  is the temperature, and  $c_p$  and  $Pr$  are specific heat coefficients at constant pressure and Prandtl number, respectively. To close these equations, the Reynolds stress  $\tau_{ij}$  and turbulent heat flux  $q_j^{(t)}$  need to be modeled, which will be introduced in the following.

### A. Reynolds stress closure with tensor basis neural network

We consider the nonlinear eddy viscosity model as the baseline model to represent the Reynolds stresses [12]. Such a choice is based on the following considerations. First, the nonlinear eddy viscosity model uses the integrity of tensor bases which is more flexible in contrast to the linear eddy viscosity model that uses only the first tensor basis. Moreover, the model is straightforward to be implemented and computationally efficient since no additional transport equations need to be solved. Besides, the nonlinear eddy viscosity models have often been used for numerical predictions of transonic flows [23, 24].

The nonlinear eddy viscosity model represents the deviatoric part of Reynolds stress with ten independent tensor bases based on Cayley-Hamilton theory [25]. Specifically, the Reynolds stress can be formulated as

$$\boldsymbol{\tau} = -2\rho k \sum_{i=1}^{10} g^{(i)} \mathbf{T}^{(i)} - \frac{2\rho k}{3} \mathbf{I}, \quad (2)$$

where  $k$  is turbulent kinetic energy,  $\mathbf{I}$  is identity matrix, and  $g^{(i)}$  is coefficient of the tensor basis  $\mathbf{T}^{(i)}$ . The coefficient  $\mathbf{g}$  can be represented as the function of scalar invariants  $\theta$  of these tensor bases. For compressible flows, we have to consider the non-zero trace of the strain rate, and the tensor bases can be written as [26, 27]

$$\begin{aligned} \mathbf{T}^{(1)} &= [\hat{\mathbf{S}} - \frac{1}{3} \text{tr} \{\hat{\mathbf{S}}\}], & \mathbf{T}^{(2)} &= [\hat{\mathbf{S}}\hat{\boldsymbol{\Omega}} - \hat{\boldsymbol{\Omega}}\hat{\mathbf{S}}], & \mathbf{T}^{(3)} &= [\hat{\mathbf{S}}^2 - \frac{1}{3} \text{tr} \{\hat{\mathbf{S}}^2\}], \\ \mathbf{T}^{(4)} &= [\hat{\boldsymbol{\Omega}}^2 - \frac{1}{3} \text{tr} \{\hat{\boldsymbol{\Omega}}^2\}], & \mathbf{T}^{(5)} &= [\hat{\mathbf{S}}^2\hat{\boldsymbol{\Omega}} - \hat{\boldsymbol{\Omega}}\hat{\mathbf{S}}^2], & \mathbf{T}^{(6)} &= [\hat{\mathbf{S}}\hat{\boldsymbol{\Omega}}^2 + \hat{\boldsymbol{\Omega}}^2\hat{\mathbf{S}} - \frac{2}{3} \text{tr} \{\hat{\mathbf{S}}\hat{\boldsymbol{\Omega}}^2\}], \\ \mathbf{T}^{(7)} &= [\hat{\mathbf{S}}^2\hat{\boldsymbol{\Omega}}^2 + \hat{\boldsymbol{\Omega}}^2\hat{\mathbf{S}}^2 - \frac{2}{3} \text{tr} \{\hat{\mathbf{S}}^2\hat{\boldsymbol{\Omega}}^2\}], & \mathbf{T}^{(8)} &= [\hat{\mathbf{S}}\hat{\boldsymbol{\Omega}}\hat{\mathbf{S}}^2 - \hat{\mathbf{S}}^2\hat{\boldsymbol{\Omega}}\hat{\mathbf{S}}], \\ \mathbf{T}^{(9)} &= [\hat{\boldsymbol{\Omega}}\hat{\mathbf{S}}\hat{\boldsymbol{\Omega}}^2 - \hat{\boldsymbol{\Omega}}^2\hat{\mathbf{S}}\hat{\boldsymbol{\Omega}}], & \mathbf{T}^{(10)} &= [\hat{\boldsymbol{\Omega}}\hat{\mathbf{S}}^2\hat{\boldsymbol{\Omega}}^2 - \hat{\boldsymbol{\Omega}}^2\hat{\mathbf{S}}^2\hat{\boldsymbol{\Omega}}]. \end{aligned} \quad (3)$$

Accordingly, one additional scalar invariant  $\text{tr}(\hat{\mathbf{S}})$  need to be included compared to the incompressible scenario [12], and hence there exist six scalar invariants as

$$\begin{aligned} \theta_1 &= \text{tr} \{\hat{\mathbf{S}}\}, & \theta_2 &= \text{tr} \{\hat{\mathbf{S}}^2\}, & \theta_3 &= \text{tr} \{\hat{\boldsymbol{\Omega}}^2\}, \\ \theta_4 &= \text{tr} \{\hat{\mathbf{S}}^3\}, & \theta_5 &= \text{tr} \{\hat{\mathbf{S}}\hat{\boldsymbol{\Omega}}^2\}, & \theta_6 &= \text{tr} \{\hat{\mathbf{S}}^2\hat{\boldsymbol{\Omega}}^2\}. \end{aligned} \quad (4)$$

In the formula above,  $\hat{\mathbf{S}}$  and  $\hat{\boldsymbol{\Omega}}$  represent the non-dimensionalized strain rate and rotation rate with the turbulence time scale  $\tau_s$ , i.e.,  $\hat{\mathbf{S}} = \tau_s \mathbf{S}$  and  $\hat{\boldsymbol{\Omega}} = \tau_s \boldsymbol{\Omega}$ . The turbulence time scale  $\tau_s$  can be estimated based on  $\tau_s = 1/(\beta^* \omega)$ , where  $\beta^*$  is a model constant,  $\omega$  is specific turbulence dissipation rate that is obtained with  $k$ - $\omega$  SST model in this work. The formula of  $k$ - $\omega$  SST model is presented in Appendix A.

Conclusively, the coefficient  $\mathbf{g}$  of tensor basis is represented with neural networks in this work, and the scalar

invariants are used as the input features of the neural network. These inputs of the tensor basis neural network [12] are modified for compressible flows to consider fluid compressibility. The obtained Reynolds stress tensor is used for the momentum equation, turbulent kinetic energy transport equation, and the energy equation in Eqs. (1).

## B. Heat flux closure with tensor basis neural networks

The turbulent heat flux  $q^{(t)}$  requires to be modeled to close the energy equation (1c). It is often estimated based on the gradient diffusion hypothesis. That is, the heat is iso-tropically diffused as

$$q_j^{(t)} = -\alpha_t \frac{\partial T}{\partial x_j}, \quad (5)$$

where  $\alpha_t$  is the turbulent thermal diffusivity. According to the Reynolds analogy between turbulent heat flux  $q_j^{(t)}$  and momentum flux  $\tau$  [28], the diffusivity  $\alpha_t$  can be calculated as

$$\alpha_t = \frac{\mu_t c_p}{Pr_t}, \quad \text{with} \quad \mu_t = -\frac{g^{(1)} \rho k}{\beta^* \omega}. \quad (6)$$

Here  $Pr_t$  is the turbulent Prandtl number, and  $\mu_t$  is the turbulent eddy viscosity, which is estimated with the first term of the deviatoric part of Reynolds stress in Eq. (2) by following the Boussinesq assumption. The eddy viscosity  $\mu_t$  is also used for the  $k$  and  $\omega$  transport equations. Admittedly, such modification is still under the gradient diffusion hypothesis. Future investigations need to represent heat flux closures based on the generalized gradient diffusion hypothesis [29], which introduces the dependence on individual Reynolds stress components. Moreover, the  $g^{(1)}$  function is constructed with only scalar invariants of the velocity gradient. For the heat flux closure, the invariants of temperature gradient [30, 31] should also be considered based on the representation theorems. It is noted that introducing the invariants associated with temperature gradient would enable the heat flux model to be nonlinear. Such nonlinear heat flux models are inappropriate for passive scalar transport equations in incompressible scenarios. However, for compressible flows, the nonlinear heat flux could be applied since the temperature is not passive scalar due to coupling with momentum equations. Also, we emphasize that this work mainly focuses on turbulence modeling, and the heat flux closure is modified according to the turbulence model. Learning the heat flux closure from the heat flux data is of significant interest and will be pursued in the near future work.

Conclusively, we use the neural network to represent the mapping between the scalar invariant and the coefficient of tensor bases, i.e.,  $\theta \mapsto \mathbf{g}$ . On the one hand, the obtained coefficients are combined with the tensor bases to form the Reynolds stress based on Eq. (2). On the other hand, the first coefficient  $g^{(1)}$  is combined with the temperature gradient to estimate the turbulent heat flux based on Eq. (5). As such, the RANS equations are closed with the neural network-based closures.

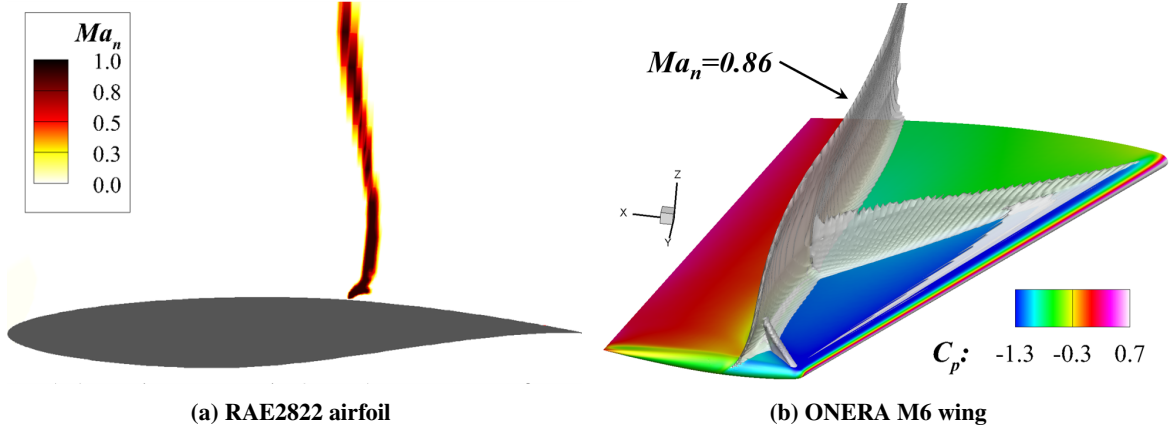
### C. Normalization of the input features

The scalar invariants  $\theta$  are used as the input features of neural networks, which are often scaled in the range of, e.g.,  $[-1, 1]$ , to improve the efficiency of the neural network training. Two strategies, i.e., the min-max normalization and local normalization, are usually used to scale the input features. The former approach uses the global minimum and maximum values to scale the input features, while the latter uses local quantities for scaling. Both approaches have been employed for feature scaling of tensor basis neural networks [15, 32].

The min-max normalization can ensure the input features within  $[0, 1]$ , which is widely used in the machine learning community. However, when encountered with shock waves, local scalar invariants can be extremely large due to the sharp discontinuity in the velocity field, leading to feature collapses, as noted in [20]. To address this issue, the features in the shock wave regime can be identified and removed from the collection of input features. Eliminating the features of shock waves has negligible effects on the model learning since the strong pressure gradients are dominant over the Reynolds stress in these areas. One widely used criteria for stationary shock wave detection can be expressed as [33]

$$Ma_n > 0.86, \quad \text{with} \quad Ma_n = \frac{\mathbf{u} \cdot \nabla p}{a |\nabla p|}, \quad (7)$$

where  $Ma_n$  is normal Mach number,  $\mathbf{u}$  is the velocity vector of the local flow, and  $a$  is the speed of sound. The detected shock wave is shown in Fig. 1 for flows over the RAE2822 airfoil and the ONERA M6 wings. Based on the criteria, we can remove the features associated with the shock wave and thus eliminate their effects on the feature scaling.



**Fig. 1** Plots of the detected shock for RAE2822 case and ONERA M6 based on criteria of  $Ma_n \geq 0.86$ .

Alternatively, the local normalization is also commonly used to scale the input features  $q$  based on  $\hat{q} = q / (\|q\| + q^*)$  [32], where  $q^*$  is local quantities. For the tensor-basis neural network, the time scale  $\tau_s$  is used as the local quantity for normalization. Thus, we can normalize the strain rate and the rotation rate through

$$\hat{\mathbf{S}} = \frac{\mathbf{S}}{\|\mathbf{S}\| + 1/\tau_s} \quad \text{and} \quad \hat{\mathbf{\Omega}} = \frac{\mathbf{\Omega}}{\|\mathbf{\Omega}\| + 1/\tau_s}. \quad (8)$$



Further, the scalar invariants are computed with these normalized  $\hat{\mathbf{S}}$  and  $\hat{\mathbf{\Omega}}$  based on Eq.(4). It is noted that this scaling is slightly different from the commonly used one, i.e.,  $\hat{q} = q/q^*$ . Because the choice of Eq. (8) can ensure the input features within the range of  $[-1, 1]$ , which can improve the convergence rate based on the practice of machine learning [32, 34]. Also, the range of input features is confined and known a priori, which enables pretraining neural network-based models to be feasible. The pertaining step is crucial to accelerate model learning and avoid nonphysical model outputs, such as negative eddy viscosity [20], which will be illustrated in Section II-D.

The min-max normalization and local normalization are analyzed based on the distribution of the scalar invariants  $\theta_1$  and  $\theta_2$  from the flow over the RAE2822 airfoils. The scatter plots of the scalar invariants are presented in Fig. 2. It can be seen that with the min-max normalization the shock wave would lead to most features narrowed down to the range of  $[0.8, 1.0]$ . Such noticeable feature separations between the shock wave and the rest areas can cause sharp discontinuity in the scalar invariants. In contrast, after removing the shock waves, the two scalar invariants  $\theta_1$  and  $\theta_2$  have a relatively uniform distribution in the feature space as shown in Fig. 2(b), which has benefits to distinguish the subtle difference of input features. It is noted that most invariants are distributed around 0. That is because parallel flows with small velocity gradients are dominant in the computational domain, and the invariants with large magnitudes only exist near the surface of airfoils. As for the local normalization, the strategy can scale the features within the range of  $[-1, 1]$  and simultaneously circumvent the feature collapse caused by the shock waves, as shown in Fig. 2(c).

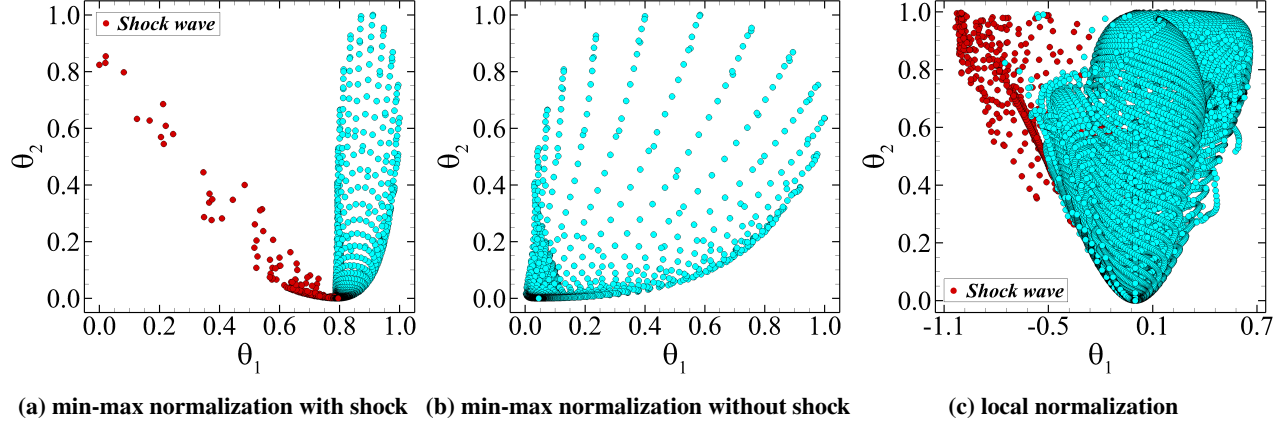
We test the effects of the min-max normalization and the local normalization on the training performance for flows over the RAE2822 airfoil. Generally, the min-max normalization leads to training divergence, likely due to the discontinuity in input features, while the local normalization can achieve good training performance in both accuracy and efficiency. After removing the features in the shock wave regime, the min-max normalization can provide similar reductions in data misfit as the local normalization. However, this strategy is very sensitive to the used criteria for shock wave detection since it requires identifying the shock wave area accurately. In contrast, the local normalization has no such limitations and hence is adopted in this paper.

#### D. Ensemble Kalman method for learning neural-network-based turbulence models

The ensemble Kalman method is a statistical inference method based on Monte Carlo sampling, which has been widely used in various applications [35–37]. This method randomly samples the weights of neural networks and uses the statistics of the weights and model predictions to estimate the gradient and Hessian of the cost function. The cost function can be written as

$$J = \|w^{n+1} - w^n\|_{\mathbf{P}} + \|\mathcal{H}[w^{n+1}] - y\|_{\mathbf{R}},$$

where  $w$  is the weight of the neural network,  $n$  is the iteration index,  $\mathbf{P}$  is the model error covariance,  $\mathbf{R}$  is the observation error covariance,  $y$  is the observation data that obey the normal distribution with zero mean and variance of  $\mathbf{R}$ , and  $\mathcal{H}$  is



**Fig. 2** Effects of shock waves on the distribution of input features. The red dots (●) indicate the features in the shock wave region.

the model operator that maps the neural network weights to the observed quantities. This method uses the ensemble of the realizations  $W$  to estimate the sample mean  $\bar{W}$  and covariance  $P$  as

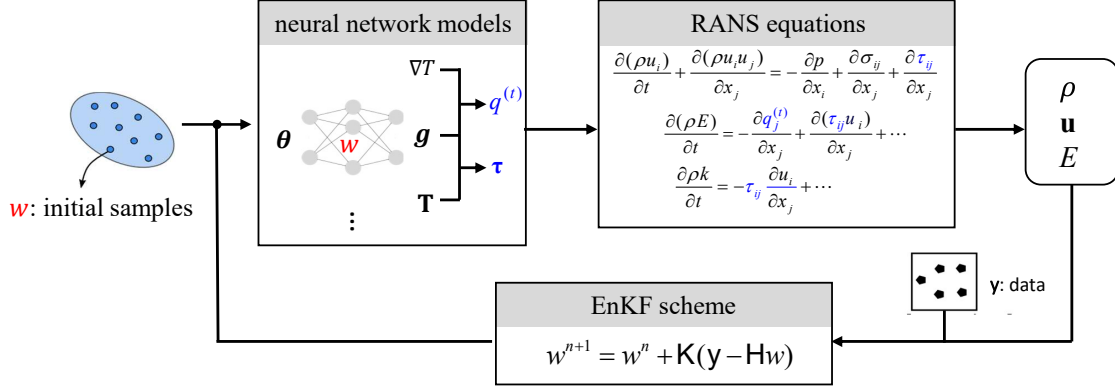
$$\begin{aligned}\bar{W} &= \frac{1}{M} \sum_{m=1}^M w_m, \\ P &= \frac{1}{M-1} (W - \bar{W})(W - \bar{W})^\top,\end{aligned}\tag{9}$$

where  $M$  is the sample size and  $m$  is the sample index. Based on the Gauss-Newton method, the weights require the first and second-order derivatives of the cost function to update the weights. The ensemble Kalman method uses the statistics of these samples to estimate the derivative information [38]. At the  $n$  th iteration, each sample  $w_m$  is updated based on

$$w_m^{n+1} = w_m^n + PH^\top (HPH^\top + R)^{-1} (y_m^n - Hw_m^n),\tag{10}$$

where  $H$  is the tangent linear model operator. The readers are referred to Ref. [20] for details of the ensemble-based turbulence modeling framework.

The schematic of the machine learning framework for compressible flows is shown in Fig. 3. Given the pre-trained weights  $w^0$  and the initial variance, we can draw an ensemble of samples for the neural network weights. The neural network output, i.e., the tensor coefficient  $\mathbf{g}$ , is combined with the tensor bases  $\mathbf{T}$  to construct the Reynolds stress tensors  $\boldsymbol{\tau}$ . Also, the coefficient  $\mathbf{g}$  is combined with the temperature gradient to form the turbulent heat flux  $q^{(t)}$ . The constructed Reynolds stress  $\boldsymbol{\tau}$  is used for the momentum equation, turbulent kinetic energy transport equation, and the energy equation, while the turbulent heat flux  $q^{(t)}$  is used for the energy equation. Through coupling the neural network and the Reynolds stress, the flow prediction is obtained by solving the RANS equations. Further the model prediction and training data are analyzed to update the weight of the neural network based on the ensemble Kalman method.



**Fig. 3 Schematic of ensemble-based training of tensor-basis neural network for transonic flows.**

The detailed training procedure is formulated as follows.

- Pre-training: Conventional weight initialization may lead to negative eddy viscosity, which may further cause the divergence of the solver. Hence, the neural network is pretrained to be a linear eddy viscosity model with  $g^1 = -0.09$ ,  $g^{(2-10)} = 0$ . The samples of neural network weights are drawn around the pretrained weights, which ensure the obtained eddy viscosity close to the baseline. Moreover, the samples that lead to negative eddy viscosity are rejected and redrawn from the prescribed normal distribution.
- Feature extraction: The predicted flow field is used to construct the input features, i.e., the scalar invariants  $\theta$  and the tensor bases  $T$ . Further, the invariants are normalized based on the local time scales as in Eq. (8).
- Coupling neural networks and the RANS equations: The input features are propagated to the  $g$  function through the neural network. The  $g$  function is further combined with tensor bases and temperature gradient to construct the Reynolds stress and turbulent heat flux, respectively. The constructed Reynolds stress and heat flux are used to close the RANS equations for estimating the mean flow field.
- Kalman correction: The flow prediction and the observation data are analyzed to update the weights of each neural network based on the ensemble Kalman method. Return to the second step until the maximum iteration is reached.

An in-house computational fluid dynamics (CFD) solver [39–42] is used to solve the RANS equations (1). The governing equations are discretized with the cell-centered finite volume method on unstructured hybrid meshes composed of hexahedrons, prisms, tetrahedrons, and pyramids. The convective flux terms are computed with the second-order Roe discretization schemes [43], and the viscous flux terms are obtained by the reconstructed central scheme. The LU-SGS relaxation-based implicit backward-Euler scheme is implemented for steady flow simulations. An adaptive local time-stepping method is developed to eliminate the adverse influence of some poor-quality grids on solution stability and convergence. All calculations are performed with double precision on a High-performance computing (HPC) platform with 2.1GHz Intel(R) CPU, and the CFD code is parallelized by using a domain decomposition strategy with the message passing interface (MPI) protocol. Nonblocking communications are used to overlap the computation

with the communication and exploit possible performance gains. The ensemble Kalman method is implemented in the publicly available DAFI code [19].

### III. Numerical results

We present two cases to demonstrate the capability of the proposed framework in learning neural network-based turbulence models for transonic flows. The test cases include 2D flows over the RAE2822 airfoil and 3D flows around the ONERA M6 wings, as summarized in Table 1. The adiabatic wall condition is applied in this work, which is widely used as a thermal boundary condition [44, 45]. Specifically, we implement the adiabatic wall conditions in the energy equation Eq. (1)c by setting the gradient of temperature as zero at walls. The learned turbulence model and heat flux closure apply the solid wall boundary conditions for  $k$  and  $\omega$  by inheriting from the baseline  $k$ - $\omega$  SST model [46]. We note that the learned models can also be applied for isothermal-wall cases by prescribing the wall temperature. The wall boundary conditions for the turbulence model and the heat flux closure are consistent with the adiabatic-wall scenarios, which are independent of the wall temperature. Such implementation strategies are commonly used to numerically investigate the hot or cold wall effects in aerospace applications [47, 48]. The training data for the two cases are from the experimental measurements in velocity and wall pressure, respectively. The learned model for each case is tested in different flow conditions to demonstrate the predictive ability of the neural network-based model.

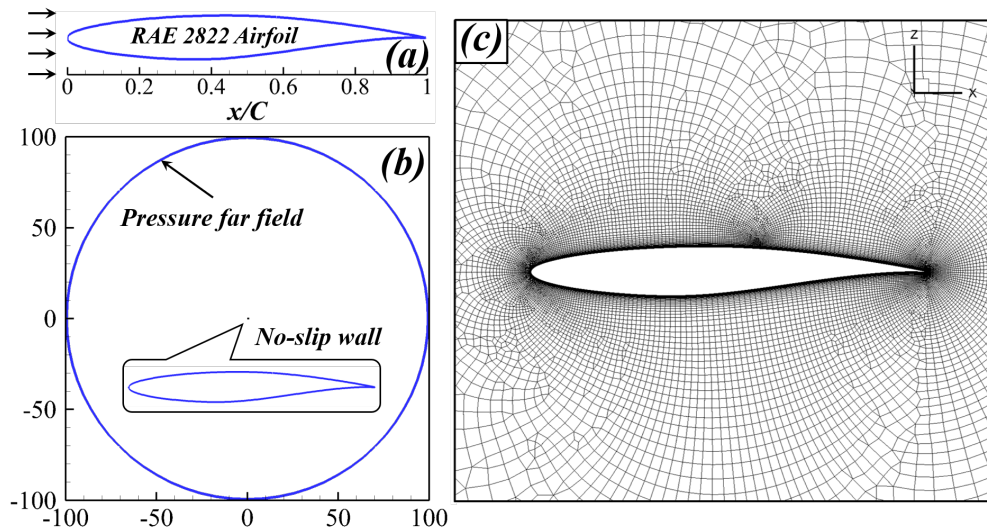
The ensemble Kalman method has good training efficiency since it is a second-order optimization with approximated gradient and Hessian information. Specifically, the method draws an ensemble of samples, and the ensemble statistics are used to estimate the gradient and Hessian [38]. For each sample, the neural network-based turbulence model will be coupled with the RANS equations to predict the flow fields. The ensemble of cases can be run in a parallel manner without requiring communication. Hence, the wall time is not significantly increased when these CFD cases are allocated to different CPUs. Specifically, the computational cost for the RAE2822 case and the ONERA M6 case is about 232.3 and 973.6 core hours, respectively, which are mostly taken by the CFD calculations. In the following, we present the detailed case setup and the corresponding training results for each case.

**Table 1** Details of the numerical settings

Cases	Geometry	Freestream conditions	data $y$	dim( $y$ )
2D training case	RAE 2822 airfoil	$Ma = 0.75, \alpha = 2.80^\circ, Re_C = 6.2 \times 10^6$	$\mathbf{u}$	31
2D test case (1)	RAE 2822 airfoil	$Ma = 0.73, \alpha = 2.80^\circ, Re_C = 6.5 \times 10^6$	$\mathbf{u}$	25
2D test case (2)	RAE 2822 airfoil	$Ma = 0.73, \alpha = 2.80^\circ, Re_C = 2.7 \times 10^6$	$\mathbf{u}$	28
3D training case	ONERA M6 wing	$Ma = 0.84, \alpha = 5.06^\circ, Re_{MAC} = 1.17 \times 10^7$	$C_p$	52
3D test case (1)	ONERA M6 wing	$Ma = 0.84, \alpha = 6.06^\circ, Re_{MAC} = 1.17 \times 10^7$	$C_p$	52
3D test case (2)	ONERA M6 wing	$Ma = 0.84, \alpha = 3.06^\circ, Re_{MAC} = 1.17 \times 10^7$	$C_p$	52

### A. Transonic flows over the RAE2822 airfoils

The hybrid unstructured mesh is used for transonic flows over the RAE2822 airfoil as shown in Fig. 4. The mesh is chosen based on grid sensitivity tests, which contains 68,560 vertices and 45,394 cells. Figure 4 (b) shows the schematic view of the computational domain. The pressure far field (PFF) is a circle with radius of 100 chord lengths and center at the leading edge of the airfoil. The adiabatic no-slip wall condition is employed on the airfoil surface. Body-fitted O-type structured mesh is generated in the nearby region of the airfoil surface, and the triangle and quadrangle meshes are used in the rest of the computational domain. The first cell height in the normal direction corresponds to  $y^+ \approx 0.8$  with a growth rate of 1.15 in the boundary layer. The training case is performed at the freestream Mach number of 0.75, the angle of attack of  $\alpha = 2.8^\circ$ , and the Reynolds number of  $Re_C = 6.2 \times 10^6$  based on the chord length  $C$ . These flow conditions are set according to the case 10 of the experimental measurements [49].

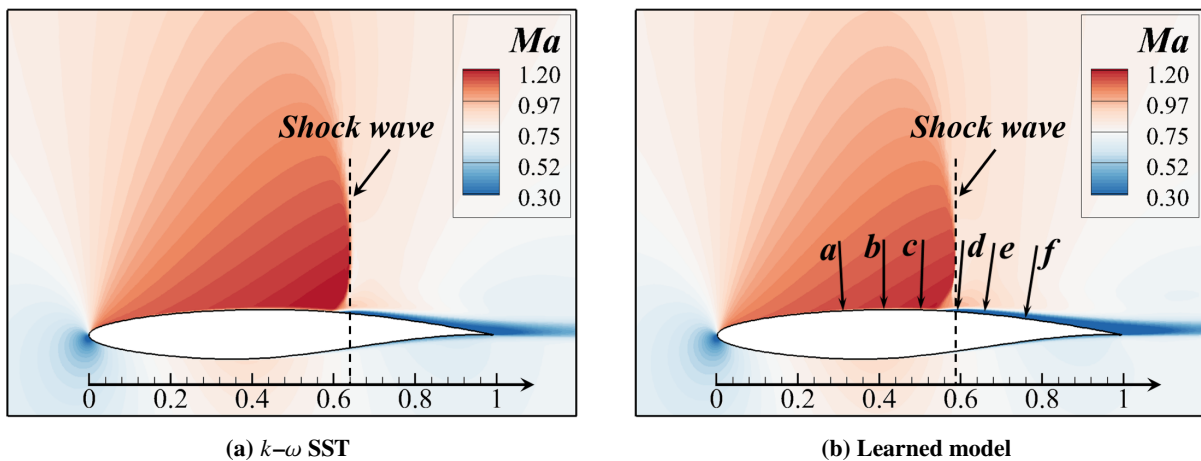


**Fig. 4** The computational setup for the RAE2822 case. (a) RAE 2822 airfoil; (b) Boundary conditions; (c) Zoomed view of the mesh near the airfoil.

We use a neural network with two hidden layers and 5 neurons per layer based on our sensitivity analysis. The details of the sensitivity study to the neural network architecture are presented in Appendix B. For this 2D case, the first three scalar invariants  $\theta^{(1-3)}$  are used as the inputs, and the first three tensor coefficients  $g^{(1-3)}$  are used as the outputs [26, 50]. The four velocity profiles, at  $x/C = 0.319$ ,  $x/C = 0.404$ ,  $x/C = 0.498$  and  $x/C = 0.574$ , are used as training data. The relative standard deviation of the neural network weights is set as 0.3, and the sample size is set as 32 for this case. The relative observation error is set as 0.1%, and the experimental data is the velocity measurements from the case 10 of the experiments [49].

The transonic flows over the RAE2822 airfoil involve shock waves, mean pressure gradient, and interaction between shock wave and boundary layer, which pose difficulties for turbulence modeling to capture such complex flows. Figure 5 presents the contour plots of Mach number with the learned model and the baseline model. We can see that the flows

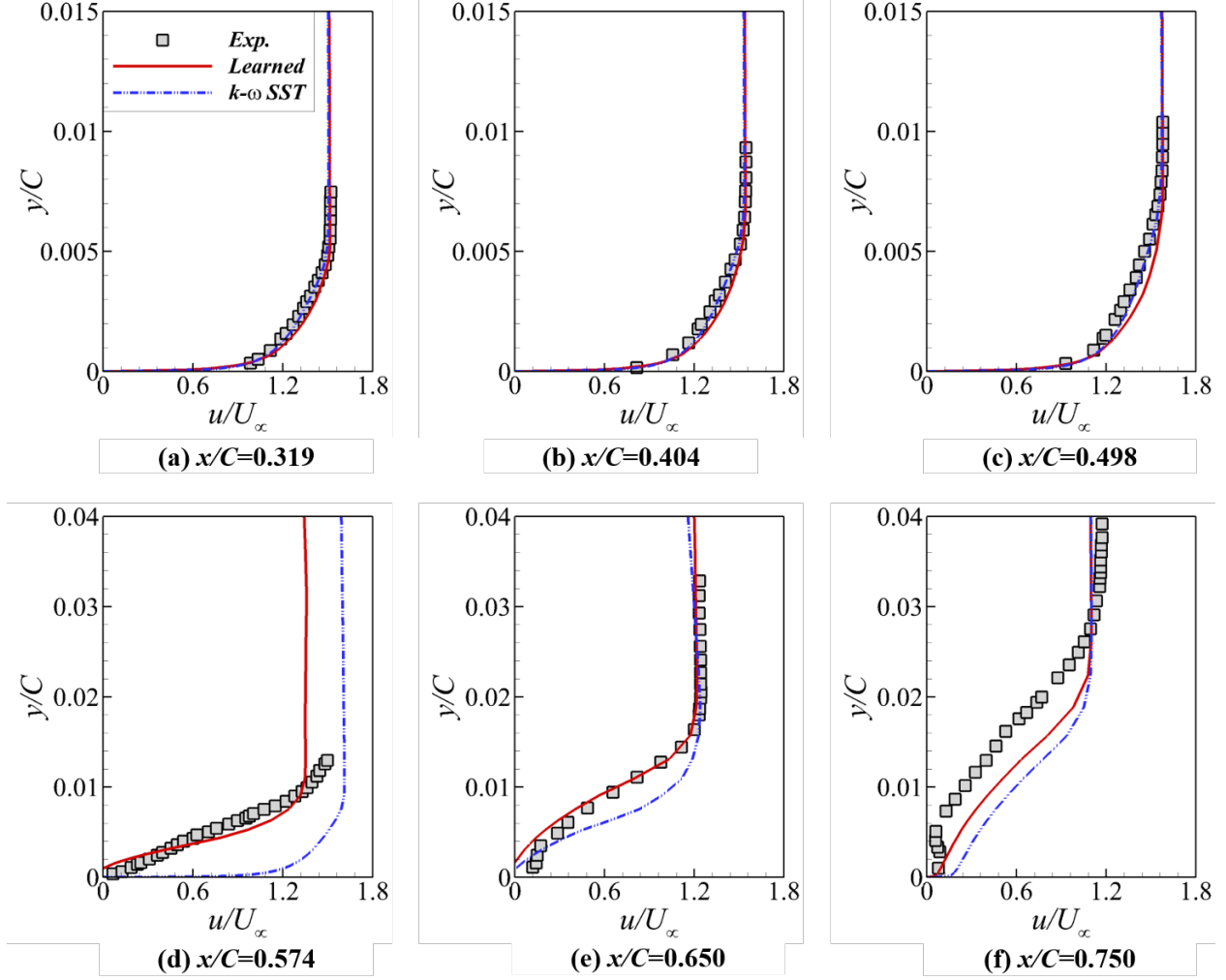
around the leading edge accelerate to the supersonic regime above the airfoil, and the supersonic flows are terminated by a shock wave leading to a supersonic pocket. At the root of the shock wave, the boundary layer thickens due to the strong adverse pressure gradient induced by the shock wave, which develops downstream and eventually forms a wake flow. The presence of the boundary layer increases the effective thickness of the airfoil, which would also affect the location of the shock wave according to the transonic similarity principles [44]. For this reason, it is challenging to predict well the flow in the boundary layer and the position of the shock wave. It can be seen in Fig. 5 that the baseline model predicts the position of the shock waves at  $x/C = 0.645$ . In contrast, the learned model predicts a relatively small supersonic region, and the shock wave occurs at a relatively upstream position, i.e.,  $x/C = 0.586$ . The experimental data of shock wave locations are not available for comparison in this case.



**Fig. 5** Contour plots of Mach number with the baseline model (a) and learned model (b) for the RAE2822 case.

The learned model can significantly improve the estimation of velocity compared to the baseline model. It can be seen from Fig. 6, which presents the velocity profiles at different locations on the upper surface of the RAE2822 airfoil. The locations are distributed on both sides of the shock wave as indicated in Figure 5 (b). The three locations of  $a$ ,  $b$ , and  $c$  are at the upstream of the shock wave, and the locations of  $d$ ,  $e$ , and  $f$  are at the downstream of the shock wave. At the locations of  $a$ ,  $b$ , and  $c$ , the characteristics of the velocity profiles are similar, and the boundary layer thickens slowly. In contrast, at the locations of  $d$ ,  $e$ , and  $f$ , the boundary layer thickens rapidly due to the adverse pressure gradient. The maximum value of the adverse pressure gradient is located at the root of the shock wave, which is between the locations of  $x/C = 0.498$  and  $x/C = 0.574$ . Such strong adverse pressure gradients result in the thickness of the boundary layer  $\delta$  rapidly increasing from  $\delta = 0.0055C$  at the location  $x/C = 0.498$  to  $\delta = 0.0083C$  at the location  $x/C = 0.574$ . When the boundary layer develops downstream to the location  $x/C = 0.750$ , the thickness can reach  $\delta = 0.0335C$ . From the comparison, the learned model can better predict the velocities than the baseline model, particularly at the locations of  $x/C = 0.574$  and  $x/C = 0.650$ .

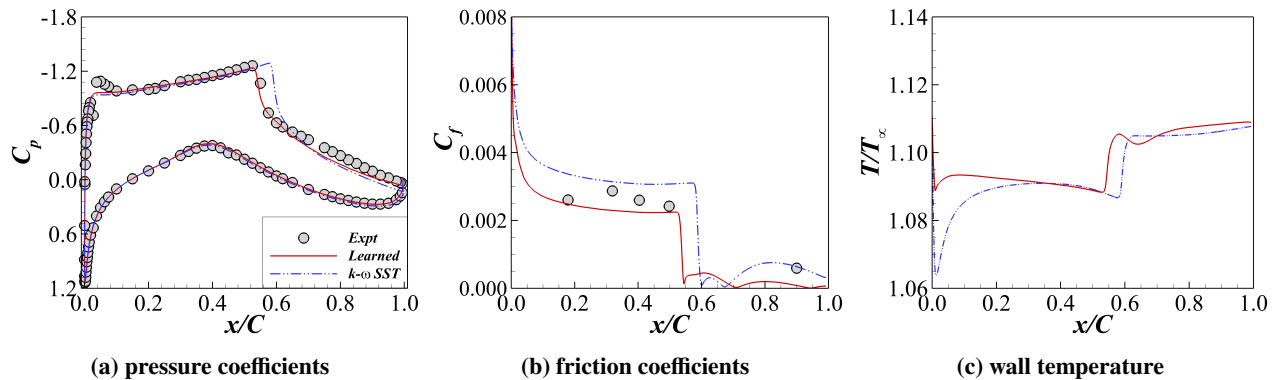
The learned model is able to improve the prediction of the wall pressure, friction coefficient, and wall temperature



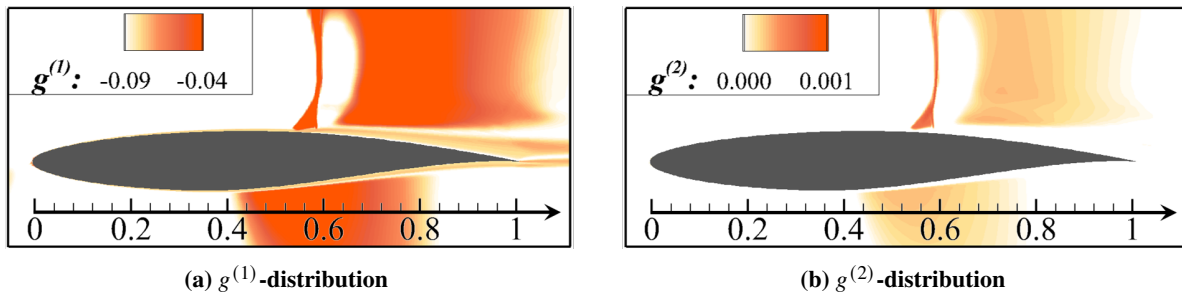
**Fig. 6** Plots of the velocity profiles with the comparison among the baseline model, the learned model, and the experimental data for the RAE2822 case.

from the baseline model. It can be seen clearly in Fig. 7, which provides the pressure and friction coefficient in comparison to the experimental measurements. The learned model provides better agreements with the experimental data in the pressure coefficient, particularly near the shock wave areas, compared to the baseline  $k-\omega$  SST model. This is because the learned model decreases the Reynolds stress in the boundary layer downstream of the shock wave, which leads to the boundary layer thickening. The thickened boundary layer reduces the effective camber of the airfoil, which drives the shock wave upstream and changes the pressure distribution in the rear of the airfoil. Evidence can be found in Figure 8(a), which shows the distribution of the  $g^{(1)}$  and  $g^{(2)}$  coefficients near the airfoil. The  $g^{(1)}$  coefficient in the boundary layer downstream is around  $-0.065$ , which has a smaller magnitude than the baseline value  $-0.09$ . That means that the learned model reduces the Reynolds stress compared to  $k-\omega$  SST model based on Eq. (2). Figure 8 (b) presents the  $g^{(2)}$  distribution in the region near the airfoil. The amplitude of  $g^{(2)}$  coefficient is close to zero in most of the areas with a maximum value of  $1.2 \times 10^{-3}$ , which indicates that the Reynolds stresses are dominated by the

linear part of the learned model for the RAE2822 case. Figure 7 (b) presents the distribution of the friction coefficients, which also shows that the learned model makes better predictions than the baseline model. Figure 7 (c) presents the distribution of the wall temperature on the upper surface. Due to the strong compressibility effect, the flow temperature will increase after passing through the shock wave. Consequently, a small jump in wall temperature can be observed at the root of the shock wave ( $x/C \approx 0.59$ ) for both the learned and the baseline models. Moreover, the learned model predicts a small reduction in the wall temperature after the shock wave.



**Fig. 7** Plots of the predicted pressure coefficients, friction coefficients, and wall temperature with comparison to the experimental data, for the RAE2822 case.

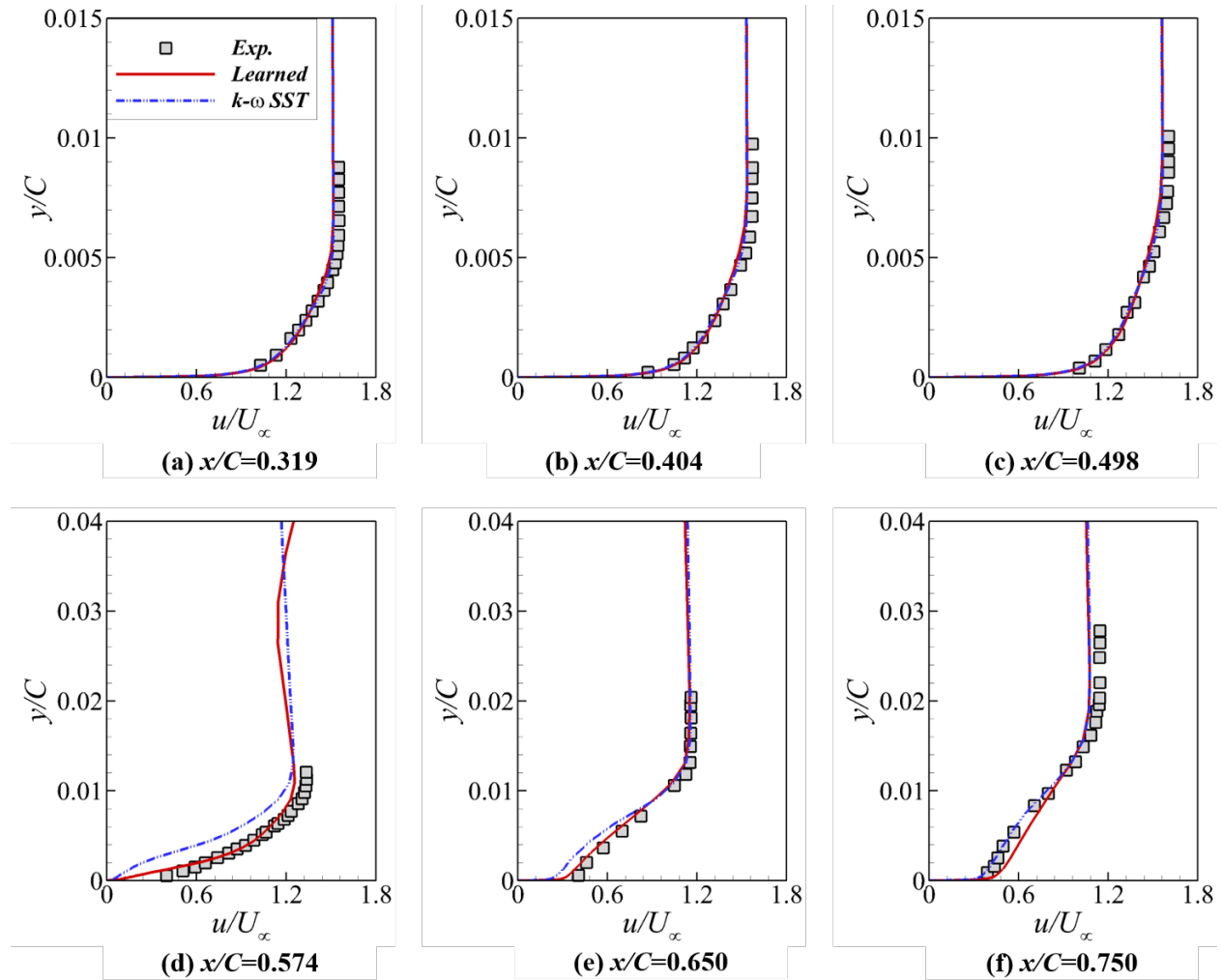


**Fig. 8** Plots of  $g^{(1)}$  and  $g^{(2)}$  coefficients for the RAE2822 case.

The generalization test shows that the trained neural network-based turbulence model is capable of predicting flow fields at different flow conditions. We test the learned turbulence model to predict the flow over the RAE2822 airfoils at two flow conditions. One has a similar Reynolds number  $Re_C = 6.5 \times 10^6$  to the training case, while the other has a much lower Reynolds number  $Re_C = 2.7 \times 10^6$ . Both cases have the Mach number 0.73 and the angle of attack  $2.80^\circ$ , which are similar to the training case. The two cases are also known as Case 9 and Case 12 [49], respectively. The transonic flows are very sensitive to the inflow conditions [51, 52] for the RAE2822 case. Small variations can lead to noticeable changes in shock wave locations. Specifically, for the 2D test case (1), the flow with the learned model provides a relatively upstream location of the shock wave at  $x/C = 0.563$  compared to the  $x/C = 0.586$  of training case, and for the 2D test case (2) the predicted shock wave is located further upstream at  $x/C = 0.498$ . The angle of



attack is given by  $\alpha = 3.19^\circ$  in the experiments, while it is recommended to be set as  $\alpha = 2.8^\circ$  in numerical simulations to consider the wind-tunnel-wall interference [49]. Our results show that the learned model can achieve predictive improvements in the velocity and wall pressure for both the test cases. Only the velocity plots for the 2D test case (1) are shown in Fig. 9 for brevity. It can be seen that the velocity prediction with the learned model can provide noticeable improvements over the baseline  $k-\omega$  SST model. Particularly, in the locations of (d) and (e) behind the shock waves, the learned model achieves significant improvements from the baseline model. At location (f), there are only slight differences between the two models, both of which have satisfactory agreements with the experimental data.



**Fig. 9** Velocity results of generalization tests with comparison among the baseline model, the learned model, and experimental data, for the RAE2822 case.

The error between the prediction and experimental measurements is quantified based on

$$\text{error} = \frac{\|q - q^{\text{exp}}\|}{\|q^{\text{exp}}\|}. \quad (11)$$

The error estimate for the training and test cases are summarized in Table 2. It can be seen that the learned model achieves better prediction accuracy compared to the baseline model. In the training case, the error in terms of velocity is reduced significantly. Specifically, the velocity error is reduced from 14.819% to 7.868%, while the error in  $C_p$  and  $C_f$  is reduced from 12.582% to 7.701% and from 21.742% to 16.531%, respectively. For the 2D test case (1) with  $Ma = 0.73$ ,  $\alpha = 2.80^\circ$ ,  $Re_C = 6.2 \times 10^6$ , the velocity prediction is significantly improved from 9.343% to 6.309% with the learned model. The pressure coefficient  $C_p$  is improved slightly from 8.1% to 6.95%, while the friction coefficient  $C_f$  is varied from 21.568% to 23.3%. Note that only velocity data is used for model training in this case. Hence, the prediction improvement in velocity is expected, while the improvement in pressure and friction coefficients can not be guaranteed unless these data are also used for model training. For the 2D test case (2) with  $Ma = 0.73$ ,  $\alpha = 2.80^\circ$ ,  $Re_C = 2.7 \times 10^6$ , the velocity prediction error is reduced from 9.856% to 6.769%, and the prediction error in wall pressure is reduced from 11.020% to 7.649%. The friction coefficient measurements are not available in this test case.

**Table 2 Summary of the prediction error in the velocity, pressure coefficient, and friction coefficient for the RAE2822 case.**

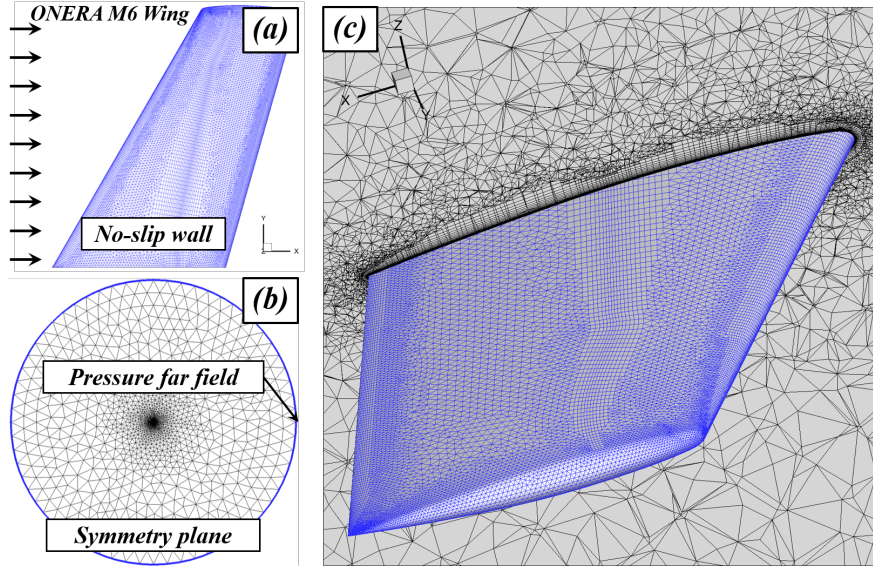
Cases	Turbulence models	Error estimation		
		Velocity	$C_p$	$C_f$
Training case	$k-\omega$ SST	14.819%	12.582%	21.742%
	learned model	7.868%	7.701%	16.531%
Test case (1)	$k-\omega$ SST	9.343%	8.100%	21.568%
	learned model	6.309%	6.950%	23.300%
Test case (2)	$k-\omega$ SST	9.856%	11.020%	— <sup>a</sup>
	learned model	6.769%	7.649%	— <sup>a</sup>

<sup>a</sup> The  $C_f$  measurements of test case (2) are not available.

## B. Transonic flows around 3D ONERA M6 wings

Flows over the ONERA M6 wings are typical external transonic flows, which have been widely used for numerical validation and turbulence model assessments. The flows include shock-induced separation, especially at a relatively high angle of attack, i.e.,  $6.06^\circ$ , which poses a challenge for turbulence modeling. Numerical simulations are performed on a hybrid unstructured mesh, as shown in Fig. 10. The mesh employs a pressure far-field boundary condition at a distance of about 200 chord lengths, a symmetry boundary condition at the inboard wing side, and an adiabatic no-slip wall condition on the wing surface. The mesh contains 3,627,926 cells and 1,602,721 points. The grid points on the wing surface are clustered near the presumed shock wave positions and the wing tip to well capture the shock wave and shock-induced separations, respectively. The dimensionless height of the first cell satisfies  $y^+ < 0.8$ , and the growth rate of the mesh cells is around 1.2 in the boundary layer. The training cases are conducted at Mach number

$Ma = 0.84$ , angle of attack  $\alpha = 5.06^\circ$ , and Reynolds number  $Re_{MAC} = 1.17 \times 10^7$  based on the mean aerodynamic chord (MAC) [53]. The mean aerodynamic chord is 0.64607 m.

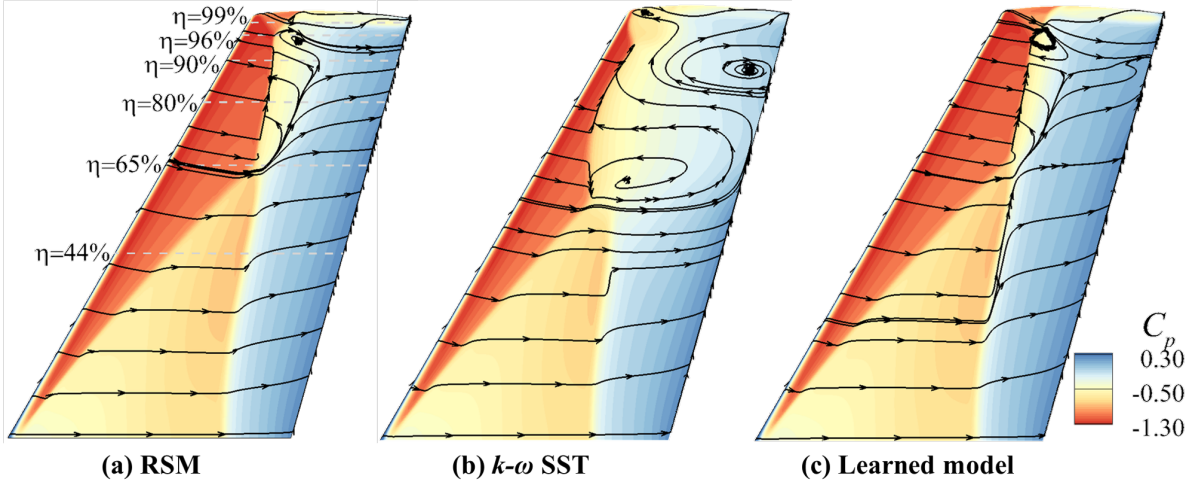


**Fig. 10** Computational setup for flows over ONERA M6 wings. (a) ONERA M6 wing; (b) boundary conditions; (c) mesh grids near the wing.

In the ONERA M6 case, we employ a neural network with two hidden layers and 10 neurons per layer based on our sensitivity analysis. The details of the sensitivity study to the neural network architecture are presented in Appendix B. The six scalar invariants  $\theta^{(1-6)}$  are used as the inputs, and the five tensor coefficients  $g^{(1-5)}$  are used as the outputs based on the minimal representation principle [54]. The relative standard deviation of the neural network weights is set as 0.2, and the sample size is set as 15 for this case. The relative observation error is set as 0.1%, and the observation data is the wall pressure distribution from the experiments [53]. The pressure measurements at four sections, i.e.,  $\eta = 0.44, 0.65, 0.90$ , and  $0.99$ , are used as training data. Herein,  $\eta$  indicates the spanwise location, defined as  $\eta = y/b$ , where  $y$  is the spanwise coordinate, and  $b$  is the spanwise length.

The learned model can better estimate the flow separation than the baseline model. We use the prediction results of the Reynolds stress transport model in the previous study [55] as a reference. Figure 11 shows the surface pressure coefficient and the distribution of streamlines with comparison among the Reynolds stress transport model (RSM),  $k - \omega$  SST model, and the learned model. From the pressure coefficient distributions, the different turbulence models predict similar development of the shock waves. The morphology of the shock waves is characterized by the formation of an  $\lambda$ -like shape on the upper wing. That is, two shock waves are formed from the wing root and gradually approach the wing tip, merging into a stronger shock wave. As can be seen from Figure 11(b), the  $k - \omega$  SST model predicts that the two shock waves on the upper surface of wings are merged at approximately  $\eta = 0.55$ , while the Reynolds stress transport model and the learned model predict the shocks merging at the spanwise  $\eta = 0.65$ . Moreover, the

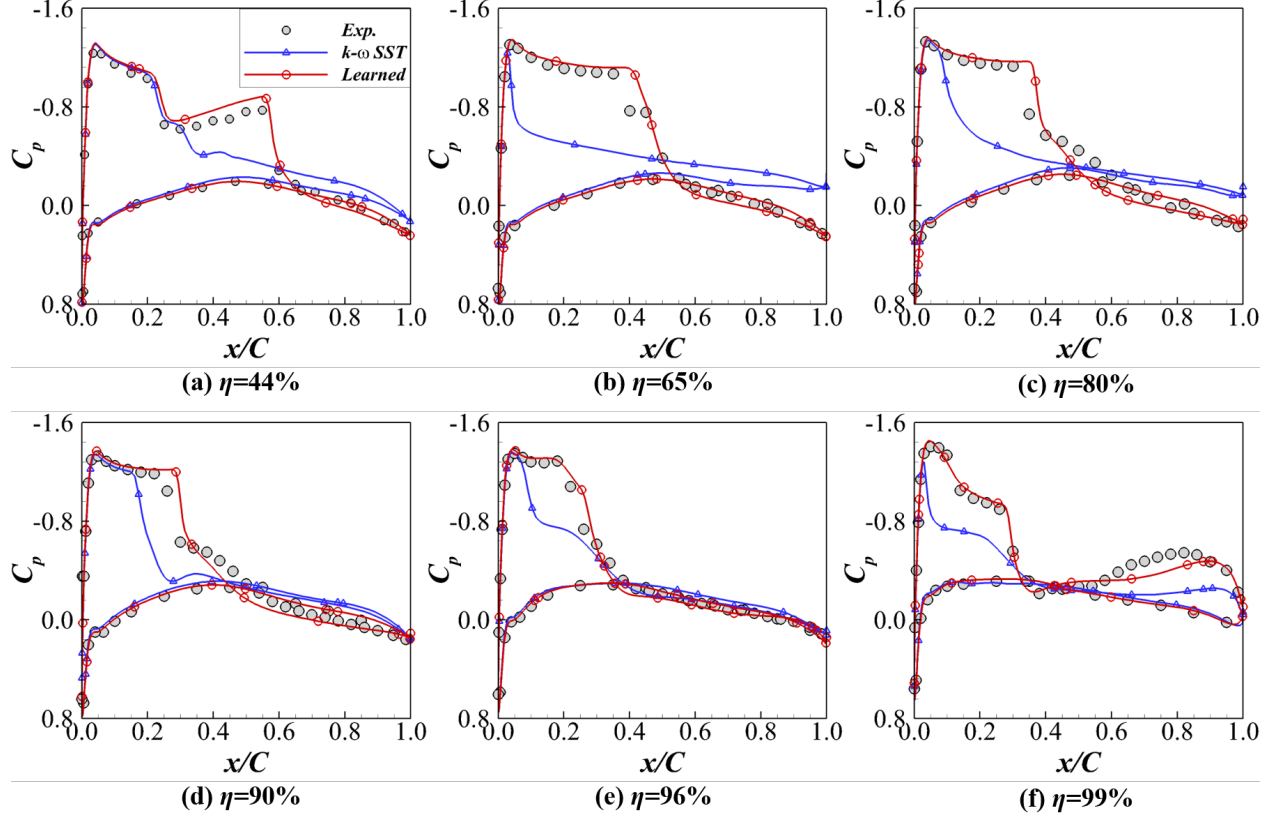
predicted shock-induced flow separation zones are significantly different with the three turbulence models. The  $k - \omega$  SST model shows a large separation area after the shock wave, which develops from the root of the shock wave to the trailing edge of the wing. In contrast, the learned model predicts similar results to the Reynolds stress transport model, with a relatively small shock-induced separation zone. Specifically, the Reynolds stress transport model,  $k - \omega$  SST model, and the learned model predict the flow separation covering the range of  $x_{RSM}/L_{MAC} \in [1.169, 1.323]$ ,  $x_{k\omega}/L_{MAC} \in [1.091, 1.729]$ ,  $x_{ML}/L_{MAC} \in [1.182, 1.349]$  in the streamwise direction, respectively.



**Fig. 11** Plots of surface pressure and streamline with the Reynolds stress transport model, baseline model, and learned model, for the ONERA M6 case.

The learned model can provide improved predictions in terms of the pressure coefficient compared to the baseline model. It is supported by Fig. 12, which shows the pressure coefficient distribution at different spanwise positions on the wing surface in comparison to the experimental data. The position of the span-wise section is indicated in Figure 11(a), i.e.,  $\eta = 0.44, 0.65, 0.80, 0.90, 0.96, 0.99$  from the wing root to the wing tip. It can be seen that the pressure coefficients predicted by the baseline  $k - \omega$  SST model have significant discrepancies from experimental data. In contrast, the learned model is able to predict the pressure coefficient in good agreement with the experimental measurements in different spanwise sections. This is due to the fact that it accurately predicts the location of the  $\lambda$ -type shock waves, as shown in Fig. 11.

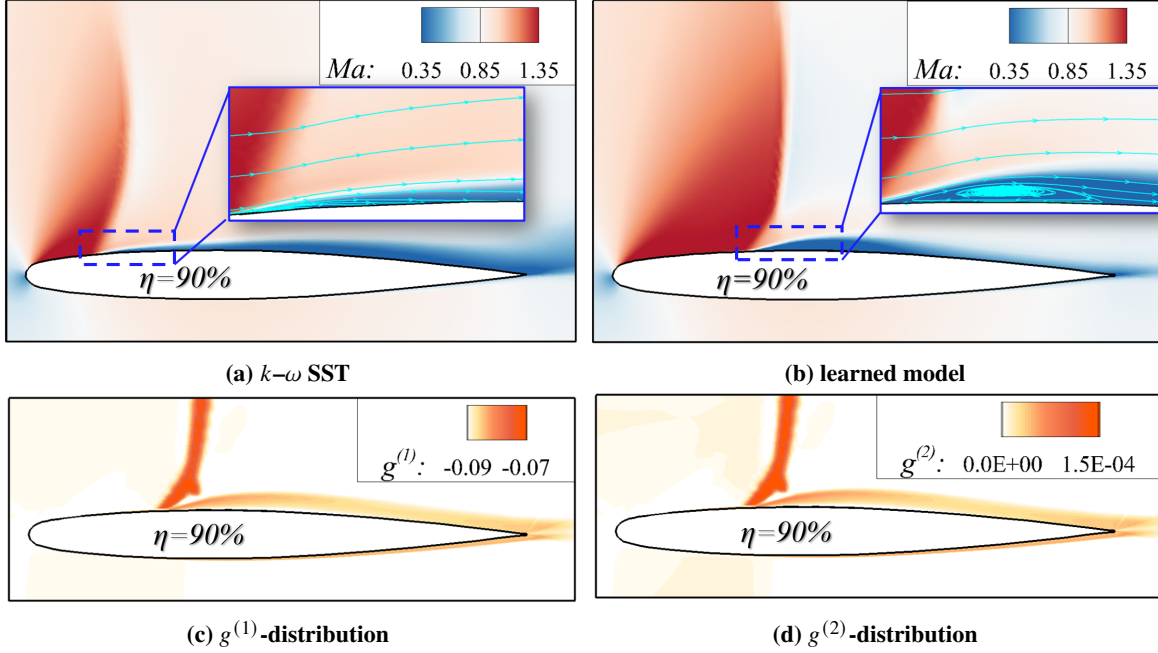
The predictive improvement in the surface pressure with the learned model is likely due to the reduction of Reynolds shear stress downstream of the shock wave. Figure 13 shows the contour plots of Mach number,  $g^{(1)}$  and  $g^{(2)}$  coefficients obtained by the learned model on a cutoff plane  $\eta = 0.90$ , where the flow separation is the most extensive. One can see that the supersonic pocket and the shock-induced separations on the suction side are resolved well. In this separation region, the learned model reduces the magnitude of  $g^{(1)}$  coefficient and further the Reynolds shear stress based on Eq. (2). The reduction of the shear stress weakens the momentum transport within the boundary layer, which leads to the thickening of the boundary layer downstream of the shock wave. In addition, the enlarged separation zone results



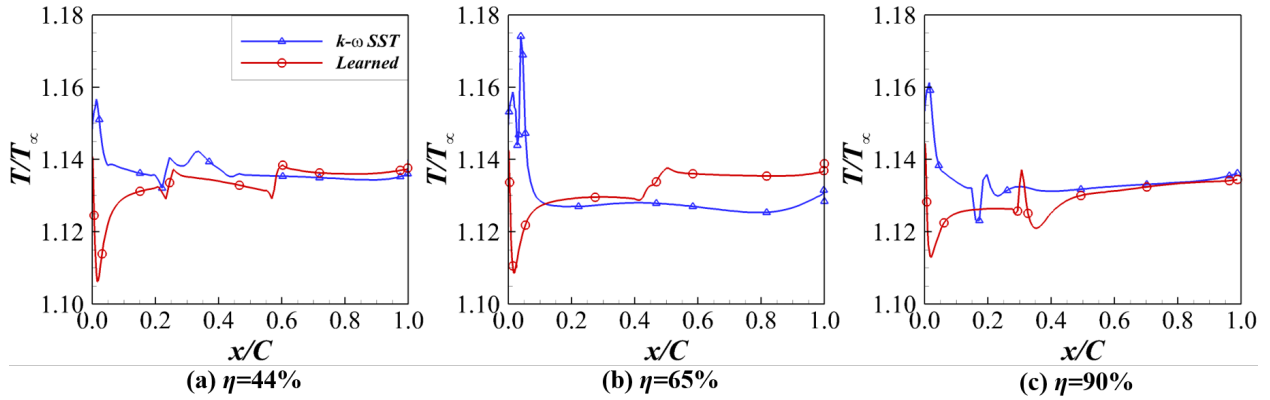
**Fig. 12** Plots of wall pressure coefficients with comparison among the baseline model, the learned model, and experimental data, for the ONERA M6 case.

in a circular-arc bump that would hold the shock wave at a relative downstream position compared to the baseline model. Recall that  $g^{(2)}$  is the coefficient of the second tensor basis, and the distribution of  $g^{(2)}$  coefficient is shown in Figure 13(d). The coefficients  $g^{(3-5)}$  have similar patterns as the  $g^{(2)}$  with very small magnitudes, and hence their plots are omitted for brevity. The relatively large amplitude is mainly in the separation and wake regions, with a maximum value of about  $1.5 \times 10^{-4}$ . It indicates that the nonlinearity of the algebraic Reynolds stress model is relatively weak in the ONERA M6 case.

Figure 14 presents the predicted temperature distribution on the upper surface of the wing with the learned model and the baseline model. The temperature distribution on the surface is significantly affected by the location of the shock wave. One can see that the wall temperature rises across the shock wave region, due to the strong compressibility effects. At the section of  $\eta = 90\%$  in Fig. 14 (c), the wall temperature is significantly reduced behind the shock wave. That is caused by the momentum transport weakening and further heat production decreasing at the shock-induced separation region. Also, it is noticeable that the learned model leads to more severe temperature reductions compared to the baseline model. That is likely because the learned model suppresses the turbulent heat transfer by reducing the magnitude of  $g^{(1)}$  coefficient at the post-shock region as shown in Fig. 13(c).



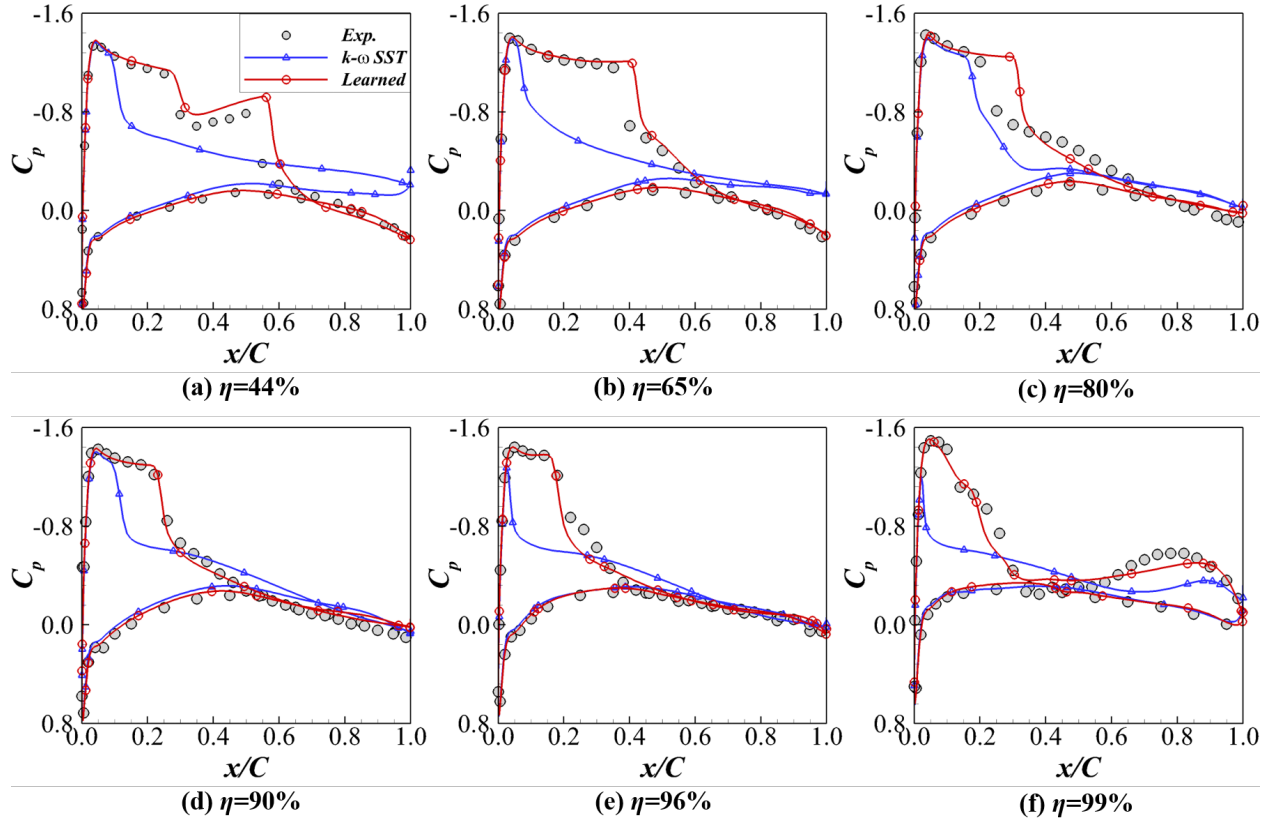
**Fig. 13** Contour plots of the Mach number,  $g^{(1)}$  and  $g^{(2)}$  coefficients at the cutoff plane of  $\eta = 90\%$  for the ONERA M6 case.



**Fig. 14** Distribution of temperature on the upper surface of the ONERA M6 wing.

The learned model can be generalized well to different flow conditions for the ONERA M6 case. Specifically, the learned model is tested for the ONERA M6 wings at the flow condition of  $Ma = 0.84$ ,  $\alpha = 6.06^\circ$ , and Reynolds number based on the mean aerodynamic chord  $Re_{MAC} = 1.17 \times 10^7$ . This flow has a wider flow separation zone than the training case in spite of slight variations in the angle of attack. The predicted pressure coefficients with the baseline model and the learned model are shown in Fig.15 with comparison to the experimental measurements. It shows that the learned model achieves more accurate predictions than the baseline model. In addition, the learned model is also tested for the ONERA M6 wings at the flow condition of  $Ma = 0.84$ ,  $\alpha = 3.06^\circ$ , and  $Re_{MAC} = 1.17 \times 10^7$ . This test case is noticeably different from the training case since the shock wave on the suction side is relatively weak and the flow

is almost attached without separation. For such attached flows, both the baseline model and the learned model can obtain good agreement with the experimental data, and hence the plots are omitted for brevity. The error between the prediction and experiment for the training and test cases is summarized in Table 3, quantitatively showing the better predictive performance of the learned model than the baseline model in both training and test cases. Here the error is computed based on the results at the spanwise locations  $\eta = 44\%, 65\%, 80\%, 90\%, 96\%, 99\%$ .



**Fig. 15** Results of generalization tests in surface pressure coefficients with comparison among the baseline model, the learned model, and experimental data, for the ONERA M6 case.

**Table 3** Summary of the prediction error in wall pressure coefficients for the ONERA M6 case.

Cases	Turbulence models	Error estimation of $C_p$
Training case	$k-\omega$ SST	35.767%
	learned model	12.378%
Test case (1)	$k-\omega$ SST	36.807%
	learned model	15.300%
Test case (2)	$k-\omega$ SST	12.356%
	learned model	11.645%

## IV. Conclusion

In this work, we demonstrate the application of the ensemble Kalman method in learning turbulence models for transonic flows from sparse experimental data. A neural network-based turbulence model for transonic flows is developed by taking into account fluid compressibility. The normalization strategies of input features, including the min-max normalization and local normalization, are investigated with emphasis on the effects of shock waves on flow features. Moreover, turbulent heat fluxes are modified with the tensor basis neural networks based on the Boussinesq assumption. The proposed neural network framework is tested in two external flow cases, i.e., 2D transonic flows around the RAE2822 airfoil and 3D transonic flows around the ONERA M6 wings. The experimental measurements of velocity and wall pressure are used to train the neural network-based model, respectively, highlighting the flexibility of the ensemble Kalman method in the use of training data. Both training and test cases show that the learned model can achieve predictions in good agreement with the training data, which demonstrates the capability of the ensemble-based turbulence modeling framework for transonic flows.

Future works will be conducted to evaluate the machine-learned turbulence models by comparison to classical nonlinear eddy viscosity models [56, 57] in various applications. Moreover, the nonlinear eddy viscosity model is still under the weak equilibrium assumption. The present ensemble method can be extended in the Reynolds stress transport model framework to construct the neural network-based model for the pressure-strain-rate correlation. Besides, the heat flux closure is based on the gradient diffusion model in this work. The generalized gradient diffusion model and additional invariants associated with the temperature gradient need to be considered for predicting the heat flux in future investigations.

### A. Baseline $k$ - $\omega$ SST turbulence model

The  $k$ - $\omega$  Shear Stress Transport (SST) model [5] is used to evaluate the turbulent kinetic energy  $k$  and specific turbulence dissipation rate  $\omega$ . The corresponding transport equations are

$$\frac{\partial \rho k}{\partial t} + \frac{\partial (\rho u_j k)}{\partial x_j} = P - \beta^* \rho k \omega + \frac{\partial}{\partial x_j} \left[ (\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_j} \right], \quad (\text{A.1})$$

$$\begin{aligned} \frac{\partial \rho \omega}{\partial t} + \frac{\partial (\rho u_j \omega)}{\partial x_j} = & \frac{\gamma \rho}{\mu_t} P - \beta \rho \omega^2 + \frac{\partial}{\partial x_j} \left[ (\mu + \sigma_\omega \mu_t) \frac{\partial \omega}{\partial x_j} \right] \\ & + 2\rho (1 - F_1) \frac{\sigma_\omega}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j}. \end{aligned} \quad (\text{A.2})$$

The production of turbulent kinetic energy is stress-based and defined as

$$P = \tau_{ij} \frac{\partial u_i}{\partial x_j}, \quad (\text{A.3})$$



where the Reynolds stress tensor  $\tau_{ij}$  is obtain by the proposed neural-network-based turbulence model. The blending function  $F_1$  is given by

$$\begin{cases} F_1 = \tanh(\Gamma^4), & \Gamma = \min[\max(\Gamma_1, \Gamma_3), \Gamma_2] \\ \Gamma_1 = \frac{500\mu}{\rho\omega d^2} & \Gamma_2 = \frac{4\sigma_{\omega 2}\rho k}{CD_{k\omega}d^2} & \Gamma_3 = \frac{\sqrt{k}}{\beta^*\omega d} \\ CD_{k\omega} = \max\left(2\rho\frac{\sigma_{\omega 2}}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}, 10^{-20}\right) \end{cases}. \quad (\text{A.4})$$

Here,  $d$  is the shortest distance from the wall. Each of the constants in Eqs. (A.1) and (A.2) is blended via

$$\varphi = F_1\varphi_1 + (1 - F_1)\varphi_2. \quad (\text{A.5})$$

The constants are

$$\begin{aligned} \sigma_{k1} = 0.85 \quad \sigma_{k2} = 1.0, \quad \sigma_{\omega 1} = 0.5, \quad \sigma_{\omega 2} = 0.856, \\ \beta_1 = 0.075 \quad \beta_2 = 0.0828, \quad \beta^* = 0.09, \\ \kappa = 0.41 \quad \gamma_1 = \frac{\beta_1}{\beta^*} - \frac{\sigma_{\omega 1}\kappa^2}{\sqrt{\beta^*}} \quad \gamma_2 = \frac{\beta_2}{\beta^*} - \frac{\sigma_{\omega 2}\kappa^2}{\sqrt{\beta^*}}. \end{aligned} \quad (\text{A.6})$$

More details can be found in ref. [5].

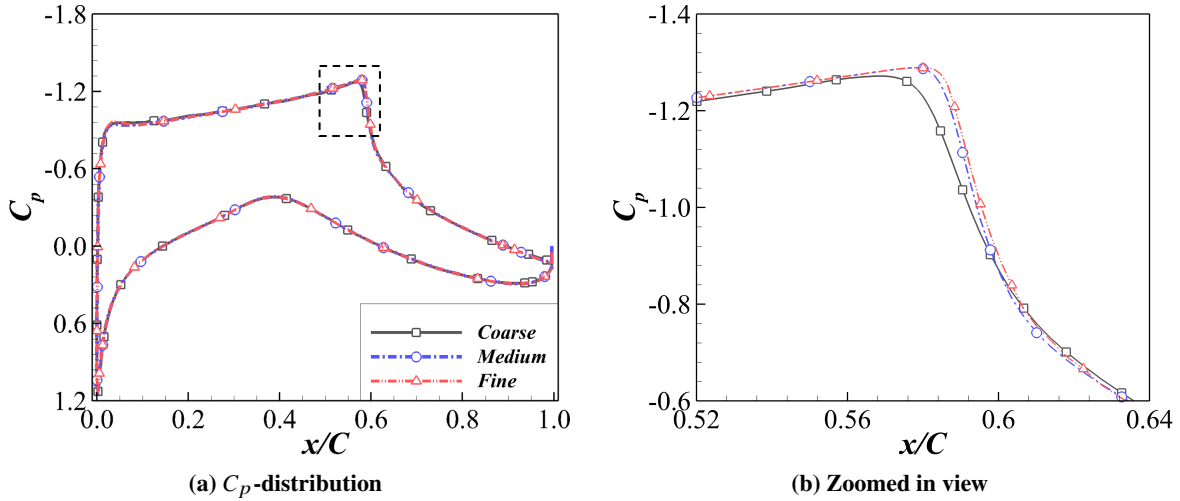
## B. Sensitivity analysis of mesh grid and network architecture

In the RAE2822 cases, the mesh sensitivity is investigated for the training case at the flow condition of  $Ma = 0.75$ ,  $\alpha = 2.8^\circ$ , and  $Re_C = 6.2 \times 10^6$ . Three typical meshes with different resolutions are generated for comparison, and the first layer spacing of all meshes in the normal direction corresponds to  $y^+ \approx 0.8$ . The cell numbers of these meshes are summarized in Table B.1, and all the grids are generated with the same topology. Figure B.1 compares the predicted wall pressure distribution  $C_p$  with various mesh sizes. The wall pressure distribution is affected by the mesh resolution mainly near the shock location, which exhibits that the medium and fine meshes provide very close results. Hence, the medium-density mesh is adopted for the RAE2822 case.

**Table B.1** Summary of mesh grids for mesh sensitivity study of the RAE 2822 case.

Meshes	Coarse	Medium	Fine
Number of cells	32,047	45,394	104,122

We also perform the mesh sensitivity study for the ONERA M6 cases. Three meshes are generated with the same topology, and the cell numbers of each mesh are summarized in Table B.2. All generated meshes are simulated with the training case, i.e., at the flow condition of  $Ma = 0.84$ ,  $\alpha = 5.06^\circ$  and  $Re = 1.17 \times 10^7$ . Figure B.2 shows the comparisons of simulated wall pressure distribution with various mesh sizes. The variation of mesh density affects the

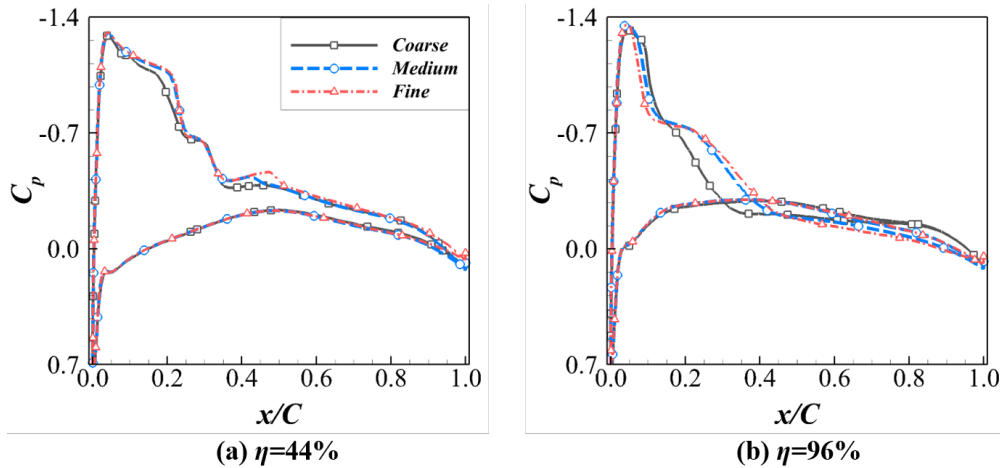


**Fig. B.1** Comparison of predicted wall pressure distribution with various mesh sizes for the RAE 2822 cases.

wall pressure distribution on the suction side, especially at the section of  $\eta = 96\%$ . The differences between the medium and the fine mesh are small in the predicted wall pressure. Therefore, the medium-density mesh is used for the ONERA M6 case.

**Table B.2** Summary of mesh grids for mesh sensitivity study of ONERA M6 cases.

Mesher	Coarse	Medium	Fine
Number of cells	2,735,233	3,627,926	6,563,867



**Fig. B.2** Comparison of predicted wall pressure distribution with various mesh sizes for the ONERA M6 case.

The sensitivity to the architecture of neural networks is investigated for both the RAE 2822 case and the ONERA M6 case. Four neural network architectures are tested: (1) two hidden layers with 5 neurons per layer; (2) two hidden layers with 10 neurons per layer; (3) four hidden layers with 20 neurons per layer; (4) eight hidden layers with 10

neurons per layer. The results of prediction errors for the RAE2822 case and the ONERA M6 case are summarized in Tables B.3 and B.4, respectively. For the RAE2822 case, the neural network with two hidden layers and 5 neurons per layer provides the best predictions in the velocity compared to the other neural network architectures. Moreover, this neural network predicts the pressure coefficient with the error of 7.701%, which has only subtle differences from other cases. As for the ONERA M6 case, the neural network with two hidden layers and 10 neurons per layer provides the best estimation of the wall pressure coefficient  $C_p$ , compared to the other architectures. Therefore, the neural network with two hidden layers and 5 neurons per layer is used for the RAE2822 case, and the neural network with two hidden layers and 10 neurons per layer is adopted for the ONERA M6 case.

**Table B.3 Parameter sensitivity analysis of neural networks for the RAE 2822 cases.**

Hidden layers	Neurons per layer	Trainable parameters	Error estimation	
			Velocity	$C_p$
2	5	68	7.868%	7.701%
2	10	183	8.262%	7.656%
4	20	1403	8.463%	7.902%
8	10	843	8.069%	7.864%

**Table B.4 Parameter sensitivity analysis of neural networks for the ONERA M6 cases.**

Hidden layers	Neurons per layer	Trainable parameters	Error estimation of $C_p$
2	5	68	12.823%
2	10	183	12.378%
4	20	1403	12.460%
8	10	843	12.668%

### Acknowledgment

This work is supported by the NSFC Basic Science Center Program for “Multiscale Problems in Nonlinear Mechanics” (No. 11988102), the National Natural Science Foundation of China (Nos. 12102439 and 12102435), and the China Postdoctoral Science Foundation (Nos. 2021M703290 and 2021M690154). The authors would like to thank the reviewers for their constructive and valuable comments, which helped improve the quality and clarity of this manuscript.

### References

- [1] Jameson, A., and Ou, K., “50 years of transonic aircraft design,” *Progress in Aerospace Sciences*, Vol. 47, No. 5, 2011, pp. 308–318. <https://doi.org/10.1016/j.paerosci.2011.01.001>.
- [2] Giannelis, N. F., Vio, G. A., and Levinski, O., “A review of recent developments in the understanding of transonic shock buffet,” *Progress in Aerospace Sciences*, Vol. 92, 2017, pp. 39–84. <https://doi.org/10.1016/j.paerosci.2017.05.004>.

- [3] Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D., “CFD vision 2030 study: a path to revolutionary computational aerosciences,” NASA CR-2014-218178, Langley Research Center, 2014, pp. 1–49. <https://doi.org/2060/20140003093>.
- [4] Spalart, P. R., and Allmaras, S. R., “A one-equation turbulence model for aerodynamic flows,” AIAA Paper 1992-439, 1992. <https://doi.org/10.2514/6.1992-439>.
- [5] Menter, F. R., “Two-equation eddy-viscosity turbulence models for engineering applications,” AIAA Journal, Vol. 32, No. 8, 1994, pp. 1598–1605. <https://doi.org/10.2514/3.12149>.
- [6] Craft, T., Launder, B., and Suga, K., “Development and application of a cubic eddy-viscosity model of turbulence,” International Journal of Heat and Fluid Flow, Vol. 17, No. 2, 1996, pp. 108–115. [https://doi.org/10.1016/0142-727X\(95\)00079-6](https://doi.org/10.1016/0142-727X(95)00079-6).
- [7] Mompean, G., Gavrilakis, S., Machiels, L., and Deville, M., “On predicting the turbulence-induced secondary flows using nonlinear  $k-\varepsilon$  models,” Physics of Fluids, Vol. 8, No. 7, 1996, pp. 1856–1868. <https://doi.org/10.1063/1.868968>.
- [8] Wilcox, D. C., et al., Turbulence modeling for CFD, Vol. 2, DCW industries La Canada, CA, 1998.
- [9] Brunton, S. L., Noack, B. R., and Koumoutsakos, P., “Machine learning for fluid mechanics,” Annual review of fluid mechanics, Vol. 52, 2020, pp. 477–508. <https://doi.org/10.1146/annurev-fluid-010719-060214>.
- [10] Sun, L., and Wang, J.-X., “Physics-constrained Bayesian neural network for fluid flow reconstruction with sparse and noisy data,” Theoretical and Applied Mechanics Letters, Vol. 10, No. 3, 2020, pp. 161–169. <https://doi.org/10.1016/j.taml.2020.01.031>.
- [11] Wu, T., and He, G. W., “Independent component analysis of streamwise velocity fluctuations in turbulent channel flows,” Theor. Appl. Mech. Lett., Vol. 12, No. 4, 2022, p. 100349. <https://doi.org/10.1016/j.taml.2022.100349>.
- [12] Ling, J., Kurzawski, A., and Templeton, J., “Reynolds averaged turbulence modelling using deep neural networks with embedded invariance,” Journal of Fluid Mechanics, Vol. 807, 2016, pp. 155–166. <https://doi.org/10.1017/jfm.2016.615>.
- [13] Weatheritt, J., and Sandberg, R., “A novel evolutionary algorithm applied to algebraic modifications of the RANS stress–strain relationship,” Journal of Computational Physics, Vol. 325, 2016, pp. 22–37. <https://doi.org/10.1016/j.jcp.2016.08.015>.
- [14] Zhou, X.-H., Han, J., and Xiao, H., “Learning nonlocal constitutive models with neural networks,” Computer Methods in Applied Mechanics and Engineering, Vol. 384, 2021, p. 113927. <https://doi.org/10.1016/j.cma.2021.113927>.
- [15] Michelén-Ströfer, C. A., and Xiao, H., “End-to-end differentiable learning of turbulence models from indirect observations,” Theoretical and Applied Mechanics Letters, Vol. 11, No. 4, 2021, p. 100280. <https://doi.org/10.1016/j.taml.2021.100280>.
- [16] Duraisamy, K., “Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence,” Physical Review Fluids, Vol. 6, No. 5, 2021, p. 050504. <https://doi.org/10.1103/PhysRevFluids.6.050504>.
- [17] Singh, A. P., Medida, S., and Duraisamy, K., “Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils,” AIAA Journal, Vol. 55, No. 7, 2017, pp. 2215–2227. <https://doi.org/10.2514/1.J055595>.

- [18] Holland, J. R., Baeder, J. D., and Duraisamy, K., “Field inversion and machine learning with embedded neural networks: Physics-consistent neural network training,” AIAA Aviation 2019 Forum, 2019, p. 3200. <https://doi.org/10.2514/6.2019-3200>.
- [19] Ströfer, C. A. M., Zhang, X.-L., and Xiao, H., “DAFI: An Open-Source Framework for Ensemble-Based Data Assimilation and Field Inversion,” Communications in Computational Physics, Vol. 29, No. 5, 2021, pp. 1583–1622. <https://doi.org/10.4208/cicp.OA-2020-0178>.
- [20] Zhang, X.-L., Xiao, H., Luo, X., and He, G., “Ensemble Kalman method for learning turbulence models from indirect observation data,” Journal of Fluid Mechanics, Vol. 949, 2022, p. A26. <https://doi.org/10.1017/jfm.2022.744>.
- [21] Zhao, Y., Akolekar, H. D., Weatheritt, J., Michelassi, V., and Sandberg, R. D., “RANS turbulence model development using CFD-driven machine learning,” Journal of Computational Physics, Vol. 411, 2020, p. 109413. <https://doi.org/10.2514/1.J059558>.
- [22] Akolekar, H. D., Waschkowski, F., Zhao, Y., Pacciani, R., and Sandberg, R. D., “Transition modeling for low pressure turbines using computational fluid dynamics driven machine learning,” Energies, Vol. 14, No. 15, 2021, p. 4680. <https://doi.org/10.3390/en14154680>.
- [23] Barakos, G., and Drikakis, D., “Numerical simulation of transonic buffet flows using various turbulence closures,” International Journal of Heat and Fluid Flow, Vol. 21, No. 5, 2000, pp. 620–626. [https://doi.org/10.1016/S0142-727X\(00\)00053-9](https://doi.org/10.1016/S0142-727X(00)00053-9).
- [24] Barakos, G., and Drikakis, D., “Investigation of nonlinear eddy-viscosity turbulence models in shock/boundary-layer interaction,” AIAA Journal, Vol. 38, No. 3, 2000, pp. 461–469. <https://doi.org/10.2514/2.983>.
- [25] Pope, S. B., Turbulent flows, Cambridge university press, 2000.
- [26] Wallin, S., and Johansson, A. V., “An explicit algebraic Reynolds stress model for incompressible and compressible turbulent flows,” Journal of Fluid Mechanics, Vol. 403, 2000, pp. 89–132. <https://doi.org/10.1017/S0022112099007004>.
- [27] Abdol-Hamid, K. S., “Development and documentation of kL-based linear, nonlinear, and full Reynolds stress turbulence models,” NASA No. TM-2018-219820, Langley Research Center, 2018.
- [28] Geankoplis, C., Transport Processes and Separation, 5<sup>th</sup> ed., Prentice Hall NJ, 2018.
- [29] Ling, J., Ryan, K. J., Bodart, J., and Eaton, J. K., “Analysis of turbulent scalar flux models for a discrete hole film cooling flow,” Journal of Turbomachinery, Vol. 138, No. 1, 2016, p. 011006. <https://doi.org/10.1115/1.4031698>.
- [30] Milani, P. M., Ling, J., Saez-Mischlich, G., Bodart, J., and Eaton, J. K., “A machine learning approach for determining the turbulent diffusivity in film cooling flows,” Journal of Turbomachinery, Vol. 140, No. 2, 2018, p. 021006. <https://doi.org/10.1115/1.4038275>.
- [31] Lav, C., Haghiri, A., and Sandberg, R., “RANS predictions of trailing-edge slot flows using heat-flux closures developed with CFD-driven machine learning,” Journal of the Global Power and Propulsion Society, Vol. 2021, No. May, 2021, pp. 1–13. <https://doi.org/10.33737/jgpps/133114>.

- [32] Ling, J., and Templeton, J., “Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty,” Physics of Fluids, Vol. 27, No. 8, 2015, p. 085103. <https://doi.org/10.1063/1.4927765>.
- [33] Lovely, D., and Haimes, R., “Shock detection from computational fluid dynamics results,” 14th computational fluid dynamics conference, 1999, p. 3285. <https://doi.org/10.2514/6.1999-3285>.
- [34] Wu, J.-L., Xiao, H., and Paterson, E., “Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework,” Physical Review Fluids, Vol. 3, No. 7, 2018, p. 074602.
- [35] Zhang, X.-L., Xiao, H., Gomez, T., and Coutier-Delgosha, O., “Evaluation of ensemble methods for quantifying uncertainties in steady-state CFD applications with small ensemble sizes,” Computers & Fluids, 2020, p. 104530. <https://doi.org/10.1016/j.compfluid.2020.104530>.
- [36] Schneider, T., Stuart, A. M., and Wu, J.-L., “Ensemble Kalman inversion for sparse learning of dynamical systems from time-averaged data,” Journal of Computational Physics, Vol. 470, 2022, p. 111559. <https://doi.org/10.1016/j.jcp.2022.111559>.
- [37] Zhang, X.-L., Xiao, H., Wu, T., and He, G., “Acoustic Inversion for Uncertainty Reduction in Reynolds-averaged Navier–Stokes-based Jet Noise Prediction,” AIAA Journal, Vol. 60, No. 4, 2022, pp. 2407–2422. <https://doi.org/10.2514/1.J060876>.
- [38] Luo, X., Stordal, A. S., Lorentzen, R. J., and Nævdal, G., “Iterative ensemble smoother as an approximate solution to a regularized minimum-average-cost problem: theory and applications,” SPE Journal, Vol. 20, No. 05, 2015, pp. 962–982. <https://doi.org/10.2118/176023-PA>.
- [39] Liu, Y., Wang, G., Zhu, H., and Ye, Z., “Numerical analysis of transonic buffet flow around a hammerhead payload fairing,” Aerospace Science and Technology, Vol. 84, 2019, pp. 604–619. <https://doi.org/10.1016/j.ast.2018.11.002>.
- [40] Liu, Y., Wang, G., and Ye, Z., “Dynamic mode extrapolation to improve the efficiency of dual time stepping method,” Journal of Computational Physics, Vol. 352, 2018, pp. 190–212. <https://doi.org/10.1016/j.jcp.2017.09.043>.
- [41] Liu, Y., “Turbulence modeling and its uncertainty quantification for complex aerodynamic flows,” Ph.D. thesis, Northwestern Polytechnical University, 2020.
- [42] Wang, G., Li, Q., and Liu, Y., “IDDES method based on differential Reynolds-stress model and its application in bluff body turbulent flows,” Aerospace Science and Technology, Vol. 119, 2021, p. 107207. <https://doi.org/10.1016/j.ast.2021.107207>.
- [43] Roe, P. L., “Characteristic-based schemes for the Euler equations,” Annual Review of Fluid Mechanics, Vol. 18, No. 1, 1986, pp. 337–365. <https://doi.org/10.1146/annurev.fl.18.010186.002005>.
- [44] Bendiksen, O. O., “Review of unsteady transonic aerodynamics: Theory and applications,” Progress in Aerospace Sciences, Vol. 47, No. 2, 2011, pp. 135–167. <https://doi.org/10.1016/j.paerosci.2010.07.001>.
- [45] Roy, C. J., and Blottner, F. G., “Review and assessment of turbulence models for hypersonic flows,” Progress in Aerospace Sciences, Vol. 42, No. 7-8, 2006, pp. 469–530. <https://doi.org/10.1016/j.paerosci.2006.12.002>.

- [46] Blazek, J., Computational fluid dynamics: principles and applications, 3<sup>rd</sup> ed., Joe Hayton, Sankt Augustin, Germany, 2015. <https://doi.org/10.1016/C2013-0-19038-1>.
- [47] Raje, P., and Sinha, K., “Anisotropic SST turbulence model for shock-boundary layer interaction,” Computers & Fluids, Vol. 228, 2021, p. 105072. <https://doi.org/10.1016/j.compfluid.2021.105072>.
- [48] Huang, J., Bretzke, J.-V., and Duan, L., “Assessment of Turbulence Models in a Hypersonic Cold-Wall Turbulent Boundary Layer,” Fluids, Vol. 4, No. 1, 2019. <https://doi.org/10.3390/fluids4010037>, URL <https://www.mdpi.com/2311-5521/4/1/37>.
- [49] Cook, P., McDonald, M., and Firmin, M., “Aerofoil RAE 2822—pressure distributions, and boundary layer and wake measurements. experimental data base for computer program assessment,” AGARD Report AR-138, 1979.
- [50] Hellsten, A., “New advanced  $k-\omega$  turbulence model for high-lift aerodynamics,” AIAA Journal, Vol. 43, No. 9, 2005, pp. 1857–1869. <https://doi.org/10.2514/1.13754>.
- [51] Witteveen, J., Doostan, A., Pecnik, R., and Iaccarino, G., “Uncertainty quantification of the transonic flow around the RAE 2822 airfoil,” Center for Turbulence Research, Annual Briefs, Stanford University, 2009.
- [52] Pisaroni, M., Nobile, F., and Leyland, P., “A Continuation Multi Level Monte Carlo (C-MLMC) method for uncertainty quantification in compressible inviscid aerodynamics,” Computer Methods in Applied Mechanics and Engineering, Vol. 326, 2017, pp. 20–50. <https://doi.org/10.1016/j.cma.2017.07.030>.
- [53] Schmitt, V., “Pressure distributions on the ONERA M6-wing at transonic mach numbers, experimental data base for computer program assessment,” AGARD AR-138, 1979.
- [54] Fu, S., Wang, C., and Guo, Y., “On the minimal representation of non-linear eddy-viscosity models,” Journal of Turbulence, , No. 12, 2011, p. N47. <https://doi.org/10.1080/14685248.2011.633085>.
- [55] Liu, Y., Zhou, Z., Zhu, L., and Wang, S., “Numerical investigation of flows around an axisymmetric body of revolution by using Reynolds-stress model based hybrid Reynolds-averaged Navier–Stokes/large eddy simulation,” Physics of Fluids, Vol. 33, No. 8, 2021, p. 085115. <https://doi.org/10.1063/5.0058016>.
- [56] Craft, T. J., Launder, B. E., and Suga, K., “Development and application of a cubic eddy-viscosity model of turbulence,” International Journal of Heat and Fluid Flow, Vol. 17, No. 2, 1996, pp. 108–115. [https://doi.org/10.1016/0142-727X\(95\)00079-6](https://doi.org/10.1016/0142-727X(95)00079-6).
- [57] Rubinstein, R., and Barton, J. M., “Nonlinear Reynolds stress models and the renormalization group,” Physics of Fluids A: Fluid Dynamics, Vol. 2, No. 8, 1990, pp. 1472–1476. <https://doi.org/10.1063/1.857595>.