Neural network-based eddy-viscosity correction for RANS simulations of flows over bi-dimensional bumps

Pedro Stefanin Volpiani, Raphaella Fusita Bernardini, Lucas Franceschini

Neural network-based eddy-viscosity correction for RANS simulations of flows over bi-dimensional bumps

Pedro Stefanin Volpiani,[1,*] Raphaella Fusita Bernardini,[1] and Lucas Franceschini[2,3]

[1] *ONERA, The French Aerospace Lab. 8 Rue des Vertugadins, 92190 Meudon, France*

[2] *University of São Paulo, Escola Politécnica, Department of Mechanical Engineering, Av. Prof. Mello Moraes, 2231, 05508-030, São Paulo SP, Brazil*

[3] *University of Twente, Max Planck Center for Complex Fluid Dynamics, Physics of Fluids Group, 7500 AE Enschede, The Netherlands*

(Dated: January 10, 2023)

The improvement of Reynolds-Averaged Navier-Stokes (RANS) models has become one major issue in the field of computational fluid dynamics (CFD). Despite being largely used in industry, eddy-viscosity models still lack accuracy even for simple flow configurations. In this scenario, we investigate new data-driven approaches for the development of machine-learning augmented turbulence models. The main contribution of this work is providing a machine-learning oriented turbulence model that estimates directly the eddy-viscosity correction, and that does not require the use of additional transport equations. The configuration studied is a turbulent flow over a parametric set of bumps characterized by different levels of curvature, pressure gradient and flow separation. An artificial neural network (ANN) model is trained, cross-validated and tested to construct a mapping between the input features and the two quantities of interest : the eddy viscosity discrepancy and the true eddy viscosity estimated from the LES. We showed that the ANN predicts the eddy viscosity discrepancy with good accuracy but when coupled to a RANS solver, the improved solution is very noisy. On the other hand, when predicting the eddy viscosity directly, the ANN-based model successfully reproduce the general flow-field behaviour, in terms of pressure and skin-friction distributions. The present methodology was proved to be robust even in predicting extrapolated flows. The methods and results of this work provide useful guidance for turbulence model developers.

I. INTRODUCTION

Reynolds-Averaged Navier-Stokes (RANS) simulations are still a workhorse in the aerospace, mechanical, and chemical industries. Thanks to their low computational cost, they remain an indispensable tool in the design, analysis, and optimization of many aerodynamic components. In this approach, only the mean flow is considered and a mathematical model is required to capture the effect of turbulence via the unknow Reynolds stresses. The essence of RANS modeling is based on expressing these unknown terms as a function of known mean quantities. One of the most widespread assumption in RANS modeling is based on the Boussinesq hypothesis, which assumes a linear relationship between the turbulent Reynolds stresses and the mean-velocity gradient tensor. The proportionality constant is called the turbulence-eddy viscosity, ν_t. Thus, the effect of turbulence is taken into account by increasing the fluid viscosity, based on local flow characteristics. Popular eddy-viscosity models are the Spalart-Allmaras (SA) model [1], relying on a modeled transport equation for the modified eddy viscosity, the shear-stress transport (SST) model [2], relying on two transport equations for the kinetic turbulence energy and the specific dissipation rate, and the $k-\varepsilon$ model that transports the turbulent kinetic energy and the turbulent dissipation [3, 4]. Eddy-viscosity RANS turbulence models gained attractiveness in industrial applications for three reasons: (i) they are relatively easy to implement, (ii) they present good convergence properties because of the additional dissipation and (iii) they have proven to give reliable predictions, especially when dealing with attached flows [5, 6]. However, it is well known that complex effects such as flows with separations, high streamline-curvature, strong pressure gradients, etc. are poorly modeled by this approach.

With the undeniable success of machine learning (ML) in numerous fields, from image and speech recognition, up to self-driven cars and medical diagnosis, CFD aerodynamic calculations are also beginning to benefit from this technology [7–9]. It is therefore natural to use this technique to improve the performance of RANS turbulence models.

In early works, machine learning was applied with the goal of correcting directly the Reynolds stress tensor. Tracey *et al.* [10] used kernel regression to correct the eigenvalues of the Reynolds stress anisotropy tensor by using input features from a low-fidelity model and training over a DNS dataset. Ling *et al.* [11, 12] have proposed a novel neural network architecture to embed key physical modeling properties with Galilean invariance into the predicted Reynolds stress anisotropy tensor. Wang *et al.* [13] proposed a data-driven, physics-informed machine learning technique based on random forests for reconstructing discrepancies in RANS modeled Reynolds stresses. They observed

that small errors in the machine-learning-predicted Reynolds stresses could lead to large errors in the mean velocity field when solving the RANS equations. To fix this issue, Wu *et al.* [14] proposed to decompose the Reynolds stresses into linear and nonlinear parts and to treat them separately to enhance the learning procedure. Weatheritt and Sandberg [15] created models for the Reynolds stress anisotropy in terms of invariants of the velocity gradient tensor using symbolic regression and gene-expression programming (GEP). Their method provides an algebraic expression, which can be straightforwardly implemented in RANS solvers and, depending on their form, easily interpretable from a physical point of view. These GEP-trained models have shown improved predictive accuracy in *a posteriori* tests such as rectangular ducts [16] and turbomachinery flows [17].

In contrast, Duraisamy and co-workers [18, 19] introduced a full-field multiplicative correction coefficient into the production term of the transport equations of turbulent quantities (e.g., $\widetilde{\nu}$ in the SA model and ω in the $k-\omega$ models). The correction terms were inferred employing data-assimilation (DA) techniques, termed field-inversion, based on DNS and experimental data and transformed into model forms using machine learning. Good results were obtained for channel flows and flows over airfoils. Following Franceschini *et al.* [20], Volpiani *et al.* [21] opted to introduce a correction to the Boussinesq-hypothesis by adding a forcing term in the momentum equations. They employed variational data assimilation to infer the vectorial source correction from high-fidelity numerical data and machine learning to reconstruct this quantity from the local mean-flow features. Their methodology showed excellent results when applied to periodic hill configurations.

In this paper, we aim at performing machine-learning based RANS simulations of a family of two-dimensional bumps. The initial geometry was proposed by Webster *et al.* [22], who investigated experimentally a turbulent boundary layer flowing over a 2D bump providing sufficient surface curvature and streamwise pressure gradient effects. A sketch of the baseline geometry is shown in Fig. 1. The bump profile consists of a circular arc of length L = 254 mm and height h = 20.1 mm. It was defined by three tangential circular arcs arranged in a way that the leading and trailing edges were tangent to the flat surface. The same geometry was used in the LES study carried out by Wu and Squires [23] at relatively lower inlet momentum-thickness Reynolds number (Re_θ = 1500 in the LES and Re_θ = 4030 in the experiment). The simulations showed good predictions of the mean flow and turbulence intensities in comparison with the experimental data of Webster *et al.* [22]. Since the flow remained attached in the original setup, more recently, Matai and Durbin [24] generated a LES database of flow over bi-dimensional bumps that exhibit different levels of flow separation by changing the bump height. This high-fidelity database is used in this study in the training and validation steps of our ML algorithm.

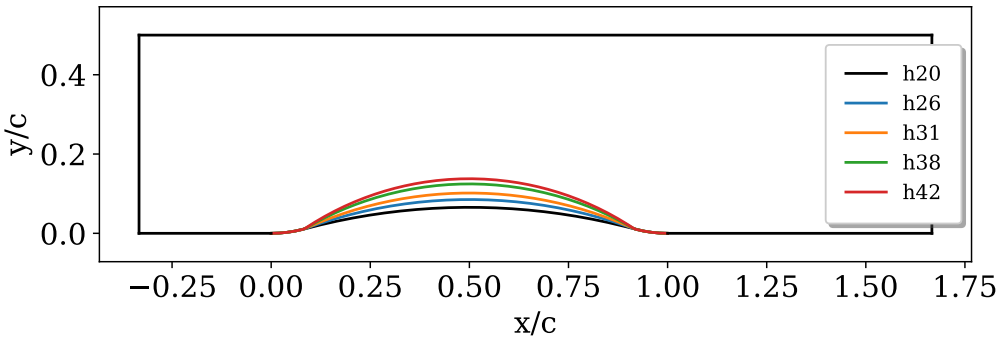

FIG. 1. Computational domain for the flows over the family of bumps. The baseline geometry is shown in black.

The novelty of this work is that, instead of focusing on correcting the Reynolds-stress tensor or a specific term in the transport equation of turbulent quantities, we focus on correcting the turbulent-eddy viscosity directly. A big advantage of this approach is that the new model is extremely simple and easy to implement. For example, full second-moment Reynolds stress models not only have a transport equation for each one of the 6 Reynolds stress components, but also a seventh transport equation for the length-scale-determining variable [25, 26]. In the present formulation, there is no need to transport unclosed terms, which can reduce implementation, verification, and computational time. The inconvenience of using an eddy-viscosity-based turbulence model is that it is inherently restricted to the Boussinesq approximation. Even if this hypothesis is inaccurate in many cases [27], we still believe that such a correction can be useful for the fluid-mechanics community. Therefore, in the proposed methodology, that seeks a machine-learned correction of the SA model, we only need to solve mass and momentum conservation equations. However, input features of the neural-network architecture are derived from the baseline RANS-SA simulations, which are used as initial conditions in the computation of the corrected fields. Another interesting aspect of the current work is the absence of the field-inversion step. This is circumvented by computing directly the estimated eddy-viscosity field from statistics of the reference flow. This makes the model design easier, but has the inconvenience of relying on detailed information of the statistics.

TABLE I. Summary of configurations studied in this work. The reference data was performed by Matai and Durbin [24].

Simulation	Height (mm)	Characteristics	Usage
h20	20 (0.0659C)	No separation	training
h26	26 (0.0878C)	Incipient separation	testing
h31	31 (0.1032C)	Separated flow	training
h38	38 (0.1259C)	Separated flow	testing
h42	42 (0.1377C)	Separated flow	training

II. CONFIGURATION AND REFERENCE FLOWFIELD

The high-fidelity reference dataset is the solution of large-eddy simulations of flows around a family of bidimensional bumps made available by Matai and Durbin [24]. The baseline geometry, primarily investigated in wind tunnel by Webster *et al.* [22], is a circular arc of length 254.0 mm, with concave fillets in both sides, so that the total length of the arc is c = 305.0 mm. In the LES, the computational domain starts at $x/c = -1/3$, ends at $x/c = 5/3$ and the top wall is located at $c/2$. The bump starts at x/c = 0.0 and ends at x/c = 1.0, as it is represented in figure 1. The simulations were performed for five geometries, in which the bump heights h were set as different values (see figure 1), summarized in Table I.

The incompressible wall-resolved LES was computed using OpenFOAM with the dynamic Smagorinsky subgrid model. The inlet boundary condition is a fully developed turbulent flow, with a boundary layer momentum thickness of θ = 3.6 mm and free stream velocity of u_∞ = 16.8 m/s. The Reynolds number based on these two quantities is Re_θ = 2500, which leads to a kinetic viscosity of $\nu = 2.4192 \cdot 10^{-5}$ m^2/s. More details about the simulation can be found in reference [24].

The flow configuration studied here is characterized by different levels of curvature, pressure gradient and flow separation. For the lowest bump height (h20), the flow remains attached all along the bottom wall. Case h26 is characterized by a flow on the verge of separation. The other configurations (h31, h38, h42) develop a small separated region near the end of the bump and its length increases with the protuberance height.

Figure 2 displays the reference pressure and skin friction coefficients for cases h20, h31 and h42 and confirms these findings. As the height of the bump h increases, the minimum value of C_p at the crest decreases, since the flow will be more accelerated. The opposite is observed for the maximum value of C_f, which will increase with the bump height. In addition, for cases $h31$ and $h42$, in which there is separation, the C_p curve between $0.8 \leq x/c \leq 1.0$ becomes flat. At the end of

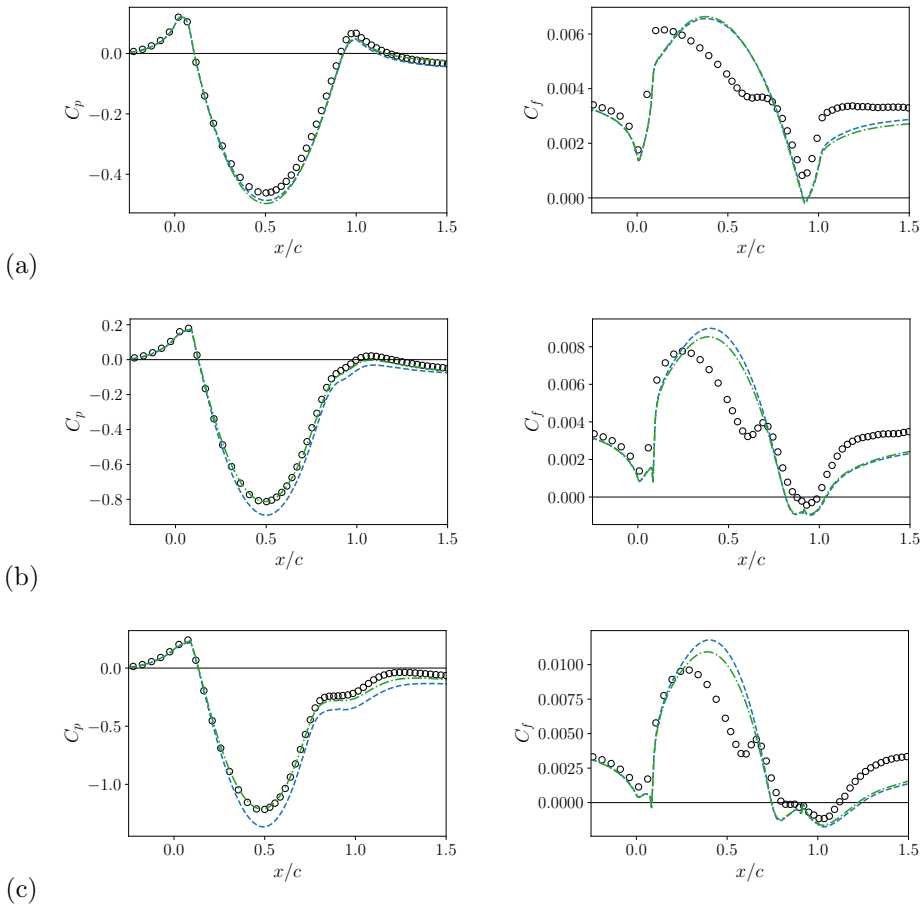

FIG. 2. Pressure (left) and skin friction (right) coefficients for cases h20 (a), h31 (b) and h42 (c). Reference LES simulation (symbols) and baseline RANS-SA with original domain and slip BC on the top wall (dashed lines) and with shorter domain (dash-dotted lines).

the bump, the adverse pressure gradient becomes stronger, and the C_f curve progressively presents a region of negative values between for cases h31, h38 and h42.

III. GOVERNING EQUATIONS

The incompressible steady-state RANS system of equations is derived by decomposing the flow variables into mean and fluctuating terms, here formalized as $u_i(x,t) = \overline{u_i}(x) + u_i^{'}(x,t)$ for the velocity and $p(x,t) = \overline{p}(x) + p^{'}(x,t)$ for the pressure. The resulting system of equations reads:

$$\frac{\partial \overline{u}_i}{\partial x_i} = 0, \tag{1}$$

$$\overline{u}_i \frac{\partial \overline{u}_j}{\partial x_j} = -\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial (2\nu S_{ij})}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_j} \tag{2}$$

with the mean strain tensor $S_{ij} = (\overline{u}_{i,j} + \overline{u}_{j,i})/2$ and the molecular viscosity ν. The term $\tau_{ij} = -\overline{u_i' u_j'}$ is the Reynolds stress tensor, and can be obtained through high-fidelity simulations.

In the RANS framework, the Reynolds stress tensor is frequently modeled following the Boussinesq approximation:

$$\tau_{ij} = 2\nu_t S_{ij} - \frac{2}{3} k \delta_{ij} \quad \text{with} \quad k = \frac{1}{2}\overline{u_i' u_i'} \,. \tag{3}$$

The turbulent kinetic energy k is usually incorporated in the pressure. The kinetic-eddy viscosity ν_t is determined by solving the one equation Spalart-Allmaras turbulence model for the quantity $\widetilde{\nu}$ ([1]):

$$u_j \frac{\partial \widetilde{\nu}}{\partial x_j} - \nabla \cdot \left(\sigma^{-1} (\nu + \tilde{\nu}) \nabla \tilde{\nu} \right) = P_{\tilde{\nu}} \left(\tilde{\nu}, \nabla \overline{\boldsymbol{u}} \right) - D_{\tilde{\nu}} \left(\tilde{\nu}, \nabla \overline{\boldsymbol{u}} \right) + C_{\tilde{\nu}} \left(\nabla \tilde{\nu} \right) \tag{4}$$

where the terms $P_{\tilde{\nu}}$, $D_{\tilde{\nu}}$ and $C_{\tilde{\nu}}$ are the production, destruction and cross-diffusion terms of the quantity $\tilde{\nu}$, and are given by:

$$P_{\tilde{\nu}} = c_{b1} \widetilde{S} \widetilde{\nu}, \;\; D_{\tilde{\nu}} = c_{w1} f_w \left[\frac{\widetilde{\nu}}{d} \right]^2 , \;\; C_{\tilde{\nu}} = \frac{c_{b2}}{\sigma} \frac{\partial \widetilde{\nu}}{\partial x_k} \frac{\partial \widetilde{\nu}}{\partial x_k} \,. \tag{5}$$

More details about the model variables and the physical definitions of each term are found in ref. [1]. Further information about the numerical implementation in the finite-element software FreeFEM [28] is available in [20, 21].

In our simulations, at the inlet, velocity profiles were set to be identical to the reference flow, and the modified turbulent viscosity was estimated from the LES solution (in the same way that will be discussed in the following section). This allows for the RANS solution to be developed in the same conditions as the LES. At the outlet, we impose zero normal stresses; and zero normal gradient of $\tilde{\nu}$. No-slip boundary conditions were imposed at the lower wall, and initially, slip boundary conditions were used at the top. However, in the reference solution files available from the NASA website [29], the boundary conditions on the top wall are questionably different from the article specification. It seems that there is a drastically reduction of the velocity at the top wall, yet the mesh size close to this boundary does not allow the correct representation of a boundary layer. RANS simulations using slip boundary conditions give erroneous pressure distribution across the lower wall (see Fig. 2). To remedy any inconsistencies, we decided to use a shorter domain and impose the exact LES variables at the top boundary. Even though we know this fix is not ideal, at least we have compatible boundary conditions. We would like to highlight the importance of having

high-fidelity data with well-defined numerical conditions. The results obtained with the new top boundary conditions are also shown in Fig. 2. The difference between RANS using the original and the shorter domain relies on the C_p distributions. The new condition gives much closer results if compared to the original LES. As for the C_f curve, this quantity was found to be much less sensitive with respect to the boundary condition specification. The major misprediction between RANS/LES resides above the bump: while the baseline RANS simulation presents an increase in C_f, with its maximum value at the crest, followed by a decrease until the end of the bump, the reference C_f curve presents a different behavior between $0.6 \leq x/c \leq 0.8$, right after the change in pressure gradient. The RANS-SA model differs from the reference results by overestimating the drop at $x/c = 0.9$, and, after the bump, underestimating the recovery of C_f for all three cases.

IV. TURBULENT EDDY-VISCOSITY ESTIMATION

In the RANS framework, the most widespread models in the aerospace industry relies on the Boussinesq assumption and the computation of a turbulent-eddy viscosity. Classical models evaluate the turbulent-eddy viscosity by solving partial differential equations (PDE) that describe the transport of turbulent variables such as a modified viscosity $\tilde{\nu}$, the turbulence kinetic energy k, its specific rate of dissipation ω, the turbulent dissipation ε, etc. The main goal of this work is to provide a PDE-free eddy-viscosity model that evaluates ν_t by means of machine learning techniques. With this in mind, a good and realistic estimate of the turbulent-eddy viscosity is needed to perform the neural network training. The most immediate method to obtain the exact field of ν_t is by calculating it directly from the reference mean and fluctuating fields. From the Boussinesq relation (3), we can derive the following turbulent-eddy viscosity expression [30]:

$$\nu_t^{LES} = \frac{-(\overline{u_i' u_j'} - 2/3 k \delta_{ij})\, \partial_j \overline{u_i}}{2 S_{ij} S_{ij}} \,. \tag{6}$$

To avoid divisions by zero or negative values of turbulent viscosites, instead of Eq. (6) we used

$$\nu_t^{LES} = \frac{\max\left(0, -(\overline{u_i' u_j'} - 2/3 k \delta_{ij})\, \partial_j \overline{u_i}\right)}{\max\left(0, 2 S_{ij} S_{ij}\right) + \epsilon} \,. \tag{7}$$

where ϵ is a small parameter.

Yet, the insertion of a physical quantity obtained from DNS or LES into a RANS simulation does not guarantee an improvement in the solution. Thompson *et al.* [31], for instance, computed

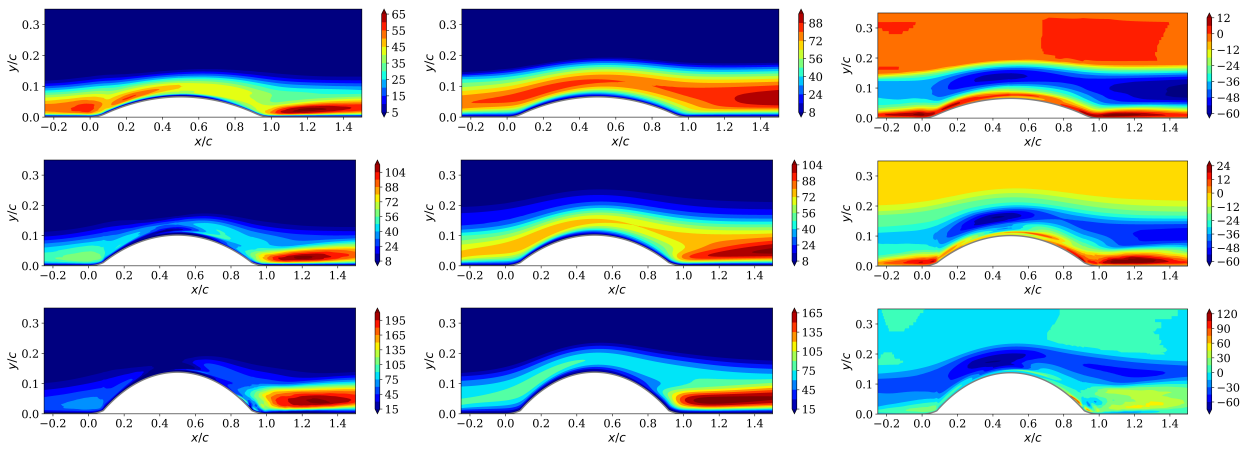

FIG. 3. Normalized turbulent viscosity ν_T/ν computed from the LES (left), computed using the SA model (middle) and their difference (right) for cases h20 (top), h31(middle) and h42 (bottom).

the true Reynolds stress tensor fields directly from DNS data and did not obtain sufficiently improved RANS solutions. Raiesi *et al.* [32] also did not reach better accuracy when calculating the turbulent kinetic energy and dissipation rate directly from DNS and LES data-sets. Therefore, it is possible that the direct extraction of the exact turbulent viscosity may not necessarily imply an augmented RANS solution. We believe that this evaluation step is important, hence, simulations were performed with the modified ν_t^{LES} to assess the effective improvement of the resulting flow.

The second way to estimate the turbulent-eddy viscosity is through a field inversion. In this approach, the estimation of the turbulent viscosity requires the correction of a RANS simulation, with the purpose of approximating the assimilated flow to the reference solution. However, Franceschini *et al.* [20] carried out a few of such inversions on a backward-facing step, which resulted in poor assimilated solutions, presenting strong unphysical distortions in separation regions. Even if other parameters could be assimilated to correct ν_t, in this work we focus solely in the direct computation of the turbulent viscosity.

The turbulent viscosity fields obtained from Eq. (7), the baseline simulation and their differences $\Delta\nu_t = \nu_t^{LES} - \nu_t^{SA}$ are shown in Figures 3 for cases $h20$, $h31$ and $h42$. The modeled eddy viscosity ν_t^{SA} resembles qualitatively the one obtained by direct calculation. In general, both contours show that there is a zone of significant turbulent viscosity right before and after the bump. The turbulent viscosity also increases in the rear region when the height of the bump increases. The main divergence between the modeled and the exact ν_t is that the first one is overpredicted in the shear layer over the crest and after the obstacle and underpredicted in the boundary-layer region close to the wall. The patterns of $\Delta\nu_t$ depend on the bump high and it is quite hard for a

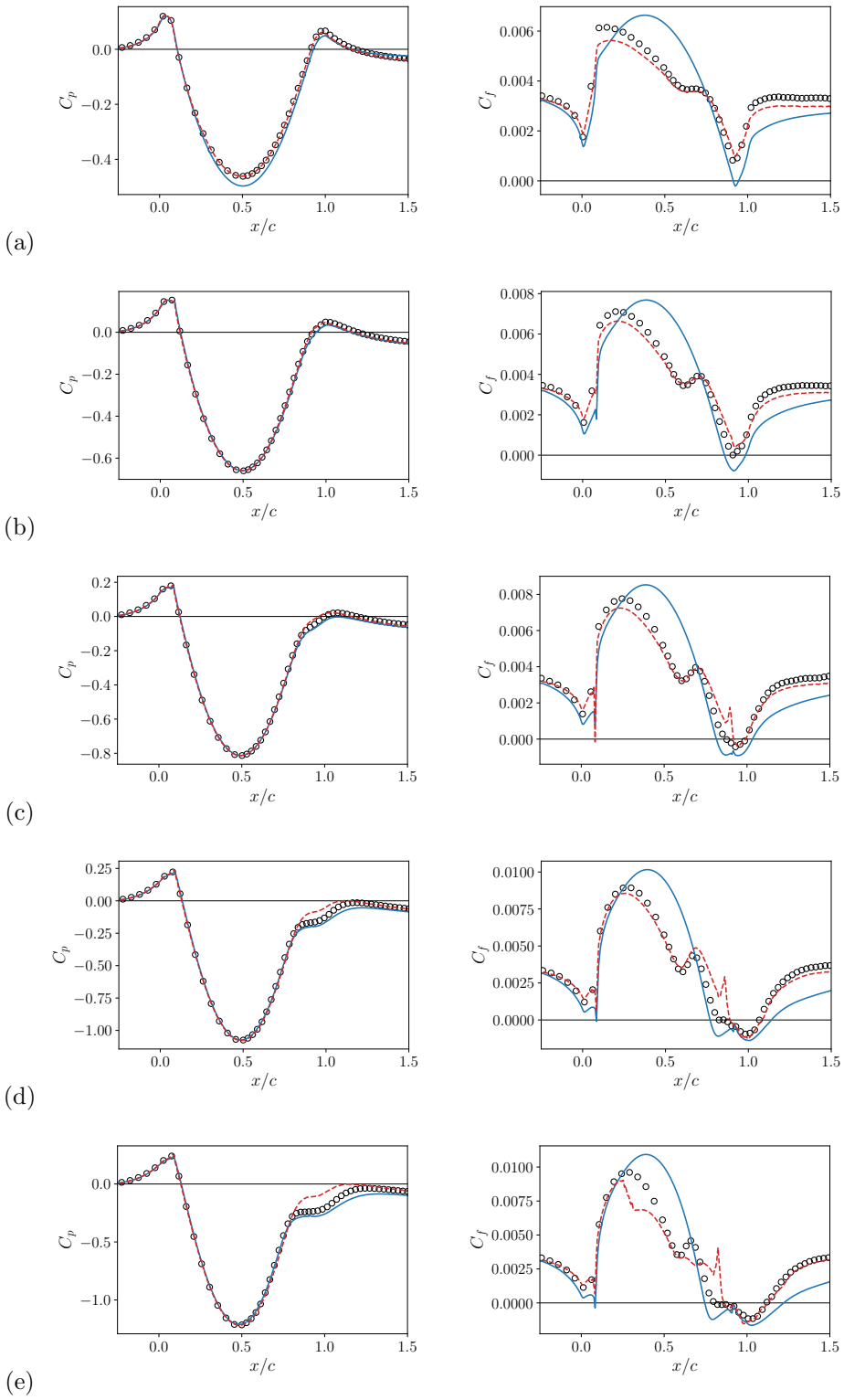

FIG. 4. Pressure (left) and skin friction (right) coefficients for cases h20 (a), h26 (b), h31 (c), h38 (d) and h42 (e). C_p and C_f curves are shown for the reference LES simulation (symbols), baseline RANS-SA (solid) and RANS-ν_t^{LES} (dashed line).

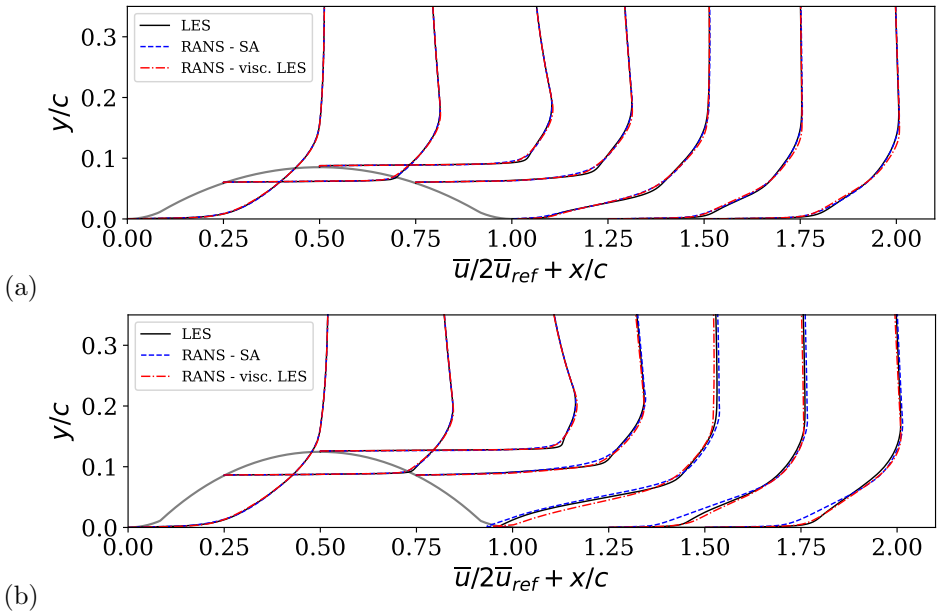

FIG. 5. Profiles of streamwise velocity component for cases (a) h26 and (b) h38. Profiles are shown for the reference LES simulation (solid), baseline RANS-SA (dashed) and RANS-ν_t^{LES} (dashed-dotted line).

turbulence model to find the right amount of ν_t for each scenario. With this in mind, we expect to predict this quantity using artificial neural networks.

Once the turbulent-viscosity fields are estimated for each case, Eqs. (1) and (2) can be solved for the velocity and pressure with no further modeling. Most of our analysis are focused on the prediction of mean pressure and skin-friction distributions, because of their major importance in the aerospace industry. The comparison between LES, baseline RANS and RANS using ν_t^{LES} is shown in Fig. 4. Figure 4 puts in evidence that the RANS simulations done with ν_t^{LES} present significant improvement over the RANS-SA solutions. Both quantities of interest, C_p and C_f approximate very well to the reference curves, indicating that the extraction of the turbulent viscosity from the high-fidelity solution leads to a satisfactory correction of the flow, especially for flows without separation. As the bump height increases, it is observed that the C_p and C_f curves in figure 4, lose some accuracy, in particular between $0.8 \leq x/c \leq 1.0$. Yet, this change in precision is expected since the modified RANS simulation is still constrained to the Boussinesq hypothesis, which lacks accuracy in regions of separation. Figure 5 shows a comparison of streamwise velocity profiles between LES, baseline RANS-SA and RANS using ν_t^{LES} for cases h26 and h38. The baseline and modified RANS profiles are similar for the attached case h26. This indicates that the standard SA model manages to predict the resulting flow field in cases with small or no separation (at least faraway from the wall). However, very close to the wall the skin friction is not correctly predicted,

as indicates Fig. 4. For case h38, we notice a better prediction of the mean velocity field for the simulation employing the modified eddy viscosity. It is worth mentioning that for flows presenting large separations, the approximation (7) may lose accuracy. Still, in general, the RANS results using ν_t^{LES} showed that it is a very good estimation of the true turbulent viscosity. We believe that trying to predict $\Delta\nu_t$ or ν_t^{LES} using machine learning can improve considerably the prediction of the mean RANS solutions. We highlight that Tan *et al.* [33] also tried to predict the eddy viscosity discrepancy but using the random forest algorithm. In their paper, however, the reference viscosity was given by the SST model and training and validation was done for a single case. They analysed the influence of several input features in the output prediction, but unfortunately did not test their algorithm in a RANS solver.

V. MACHINE LEARNING STRATEGY

The objective of an artificial neural network (also known as multi-layer perceptron) is to improve complex computational models. Artificial neural networks are nonlinear functions parameterized by weights w and biases b that can be learned from data. The concept of neural networks was inspired from the structure of biological neural circuits. Its basic unit is called perceptron. The ensemble of perceptrons organized in layers composes our neural network. The activation $q_i^{(l)}$ of perceptron i in the l^{th} layer can be represented as

$$q_i^{(l)} = \phi\left(\sum_j w_{ij}^{(l)} q_j^{(l-1)} + b_i^{(l)}\right). \tag{8}$$

In other words, the input data from the $(l-1)^{\text{th}}$ layer are multiplied by a weight w, linearly combined and then passed through a nonlinear activation function ϕ. Therefore, ANN are built by consecutive composition of linear functions (matrix-vector multiplication) followed by nonlinear activation functions. Some of the widely used activation functions are the sigmoid function $\sigma(x) = 1/(1+\exp^{-x})$ or the rectifier linear unit (ReLU), $\sigma(x) = \max(0, x)$. For deep networks, ReLU is known to be a good candidate [34] and, for this reason, was kept in this study.

This feedforward algorithm is a mathematical model equivalent to a non-linear function with a high number of parameters $(\boldsymbol{w}, \boldsymbol{b})$, that relates the inputs in the first layer to the outputs in the last layer. This function is referred here as $N(\boldsymbol{Q}(\boldsymbol{x}))$, where $\boldsymbol{Q}(\boldsymbol{x})$ is the vector representing the input layer variables and N is the final neural network output.

A. Training and validation

The five available LES data sets are divided into: training cases, from which the neural network will learn; and testing cases, which will be used to validate our augmented RANS model. This separation, summarized in table I, aims towards a realistic scenario of data-driven models' usage in the future, since it represents a situation where high-fidelity data is available for a certain number of flows, and prediction is needed for an entirely new flow, somewhat similar to the training cases, but for which a reference solution is not available. This division reaches for the neural network's predictive capability of doing an interpolation between training cases, for a flow configuration that presents curvature, pressure gradient and separation. In particular, bump $h26$ represents a test case of a flow in the verge of separation, between an attached flow, $h20$, and a separated flow, $h31$. Even further, bump $h38$ represents an intermediate case between two separated flows, $h31$ and $h42$. Therefore, the neural network model must adapt to different types of flow conditions. Once the training data is defined, the neural network learning process consists of an optimization problem, where the objective is to tune the weights and biases such that the resulting network is fitted to the training data. The goal is therefore to search for the optimal parameters $(\boldsymbol{w}, \boldsymbol{b})$ that minimize the cost function:

$$L(\boldsymbol{w}, \boldsymbol{b}) = \frac{1}{2} \sum_{k=1}^{n} \left(N_{train_k} - N(\boldsymbol{Q}_{train_k}, \boldsymbol{w}, \boldsymbol{b}) \right)^2 \tag{9}$$

Here, n is the size of the training data and k is one data point, to which is associated a set of reference inputs $\boldsymbol{Q}_{train_k}$ and output N_{train_k}. Moreover, $N(\boldsymbol{Q}_{train_k}, \boldsymbol{w}, \boldsymbol{b})$ is the neural network's predicted output for the given reference inputs, which evidently depends on the parameters $(\boldsymbol{w}, \boldsymbol{b})$ as well.

Consequently, the optimal parameters will lead to a neural network that minimizes the difference between the predicted and reference outputs. In other words, it will provide an approximation of the functional form $N(\boldsymbol{Q}(\boldsymbol{x}))$ that better describes the training data.

Validation of the training procedure is done by previously separating the dataset into training and validation data. Here, 75% of the $h20$, $h38$ and $h42$ datasets were used for training, and 25% were used for validation. A fit function, relative to the standard deviation of the validation data, was defined to measure the predictive capability of the neural network during training.

$$fit = 1 - \frac{\sum_{k=1}^{m} (N(\boldsymbol{Q}_{val_k}) - N_{val_k})^2}{\sum_{k=1}^{m} (\overline{N_{val}} - N_{val_k})^2} \tag{10}$$

Here, m is the size of the validation data-set and k is one data point, to which is associated a set of reference inputs $\boldsymbol{Q}_{val_k}$ and output N_{val_k}. This fit parameter is evaluated during the training process to assess the quality of the network. It is taken into consideration for the choice of the hyper-parameters such as: number of hidden layers, number of neurons in each hidden layer and optimizer learning rate. The low memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) method was chosen as the optimizer with a learning rate of 0.1, and the back-propagation algorithm was applied for gradient calculation. The weights were initialized following the method proposed by Glorot & Bengio [35], and all the biases were initialized as 0.1. The open-source library *PyTorch* is employed in order to re-use well-implemented methods for these procedures. Appropriate hyper-parameters for the network training are found by a grid search implementation. The training of the network was terminated using an early-stopping criterion. If validation errors did not improve after 5 epochs, the optimization was interrupted.

B. Input features

For the prediction of our quantity of interest, a set of physical quantities is formulated to serve as inputs for the neural network. These input features are created from physical reasoning and are based only on information about the mean flow, in accordance with the nature of RANS solutions. Knowing that the formulation of input features is equivalent to defining model variables for ν_t/ν, one important remark is that the inputs should be non-dimensional and rotationally invariant, in order to respect the turbulence modeling principles of dimensional homogeneity and objectivity.

Ling and Templeton [36] have composed a collection of 12 input features, among which we highlight: Q-criterion; ratio of pressure stresses to shear stresses; Gorlé *et al.* [37]'s marker for deviation from parallel shear flow; streamline pressure gradient; viscosity ratio; and turbulence intensity. In this paper, we also added turbulence quantities related to the model that we aim to fix. We employ the SA ratio of production to destruction and the ratio of production to diffusion. All input features are summarised in Table II. Since we do not transport the turbulent kinetic energy, we reconstruct this variable using the 2013 version of the Quadratic Constitutive Relation (QCR) [38], with $k_{qcr} = \frac{3}{2} c_{cr2} \nu_t \sqrt{2 S_{ij} S_{ij}}$. To avoid feature domination, input features are

TABLE II. Set #1 of local input features.

Feature	Description	Formula
q_1	Q-criterion	$\frac{\left\|\mathbf{\Omega}\right\|^2 - \left\|\mathbf{S}\right\|^2}{\left\|\mathbf{\Omega}\right\|^2 + \left\|\mathbf{S}\right\|^2}$
q_2	Ratio of pressure normal stresses to shear stresses	$\frac{\sqrt{\frac{\partial \bar{p}}{\partial x_k}\frac{\partial \bar{p}}{\partial x_k}}}{\sqrt{\frac{\partial \bar{p}}{\partial x_k}\frac{\partial \bar{p}}{\partial x_k}} + \frac{1}{2}\frac{\partial \bar{u}_k^2}{\partial x_k}}$
q_3	Gorlé *et al.* [37] marker	$\frac{\left\|\bar{u}_i\bar{u}_j\frac{\partial \bar{u}_i}{\partial x_j}\right\|}{\left\|\bar{u}_i\bar{u}_j\frac{\partial \bar{u}_i}{\partial x_j}\right\| + \sqrt{\bar{u}_l\bar{u}_l\bar{u}_i\frac{\partial \bar{u}_i}{\partial x_j}\bar{u}_k\frac{\partial \bar{u}_k}{\partial x_j}}}$
q_4	Streamline pressure gradient	$\frac{\bar{u}_k\frac{\partial \bar{p}}{\partial x_k}}{\left\|\bar{u}_k\frac{\partial \bar{p}}{\partial x_k}\right\| + \sqrt{\frac{\partial \bar{p}}{\partial x_k}\frac{\partial \bar{p}}{\partial x_k}\bar{u}_i\bar{u}_i}}$
q_5	Viscosity ratio	$\frac{\nu_t}{\nu_t + 100\nu}$
q_6	SA ratio of production to destruction	$\frac{c_{b1}\widetilde{S}\widetilde{\nu}}{\left\|c_{b1}\widetilde{S}\widetilde{\nu}\right\| + c_{w1}f_w\left(\frac{\widetilde{\nu}}{d}\right)^2}$
q_7	SA ratio of production to diffusion	$\frac{c_{b1}\widetilde{S}\widetilde{\nu}}{\left\|c_{b1}\widetilde{S}\widetilde{\nu}\right\| + \frac{c_{b2}}{\sigma}\frac{\partial \widetilde{\nu}}{\partial x_k}\frac{\partial \widetilde{\nu}}{\partial x_k}}$
q_8	Turbulence intensity	$\frac{k_{qcr}}{k_{qcr} + \frac{1}{2}\bar{u}_i^2}$

normalized in a way that all features exhibit a maximum value of approximately 1 and a minimum value of around 0 if the feature is positive (or -1 if the feature is both positive and negative).

C. Output quantity and assessment of the quality of the neural network

In this study, we explored two ways of correcting the turbulence-eddy viscosity field by means of ML techniques. In a first moment, we trained artificial neural networks to compute the normalized discrepancy between LES- and SA- based turbulence viscosities: $\Delta\nu_t/\nu$; and in a second moment, we focus on predicting the normalized turbulence viscosity ν_t^{LES}/ν directly. We then investigated the performance and robustness of each formulation.

Once the training was carried out using the input features from Table II, we compared the performance of each neural network taking into consideration the fit parameter given by Eq. (10). The resulting neural networks architecture and performance are shown in Table III for both approaches. Even though table III presents the fit parameter for the entire dataset, the choice of the best model performance cannot take into account the validation cases. This is because, in a realistic scenario, the high-fidelity data would not be available for bumps $h26$ and $h38$, therefore

the evaluation of the resulting models must be made by considering only the training data. We observe that the neural network that predicts ν_t/ν has a slight advantage over the one that predicts $\Delta\nu_t/\nu$. Nonetheless, both performances are still outstanding. The fit parameter is close to 99% in both cases.

Moreover, we note in Fig. 6(a) that it is easier to approximate ν_t/ν rather than $\Delta\nu_t/\nu$ since the ANN needs less iterations to maximize the fit function. Figure 6(b) shows the evolution of the mean training and validation cost functions throughout the iterations for NN2. The figure confirms that training was stopped before the validation loss and fit parameter became a plateau, respecting the defined criteria to avoid overfitting.

To have a more visual representation, the predicted outputs from NN1 and NN2 are plotted respectively in Figs. 7 and 8 together with their true fields for cases h26 and h38 that have not been used during training. We note that the neural network captures, in general, the physical behaviour of the expected quantities. Concerning the viscosity discrepancy, despite the noisy prediction, the ANN successfully predicts the right levels of $\Delta\nu_t$ in most of the domain. Some miss predictions still exist very locally. Regarding the total turbulence viscosity, the overall prediction is also excellent. Close to the wall, the ANN was able to reproduce the regions of high ν_t upstream and downstream of the bump, and at the crest. And towards the free stream, the predicted ν_t becomes zero, as in the reference field.

Finally, Fig. 9 shows a visual representation of how the training process makes an effort to approximate the network's outputs to the training data. Most of the points reside in the vicinity of the reference line, yet they are not exactly identical, and there is still a part of the dataset that does not approach the line as desired. We also remark that the comparison between subfigures 9(a) and 9(b), for instance, confirms that the network that predicts ν_t/ν approximates slightly better to the reference datasets than the one that predicts $\Delta\nu_t/\nu$.

VI. A POSTERIORI RESULTS

In this section, predictive tests are performed using the new NN-based model. The first step of the algorithm is to compute the input features from the initial baseline RANS-SA solution. Then, the neural-network routine generates the output parameter, which is subsequently introduced into the RANS solver. The converged state can be then compared with the reference and baseline solutions. We point out that the predictive test-cases should not include any data employed in the training phase.

Label	Output variable	NN structure	fit (%)					
			$h20$ (T)	$h26$ (V)	$h31$ (T)	$h38$ (V)	$h42$ (T)	all
NN1	$\Delta\nu_t/\nu$	$8\times(4\times80)\times1$	99.1	98.6	98.8	96.2	98.6	98.0
NN2	ν_t/ν	$8\times(4\times80)\times1$	99.6	98.9	99.6	98.7	99.6	99.3

TABLE III. Evaluation of the NN performance through the fit parameter. (T) indicates the training cases and (V) the verification cases not used during training. NN structure is in the form $N_{inputs}\times(N_{layers}\times N_{neurons})\times N_{outputs}$, where N_{layers} is the number of hidden layers and $N_{neurons}$ the number of neurons per hidden layer.

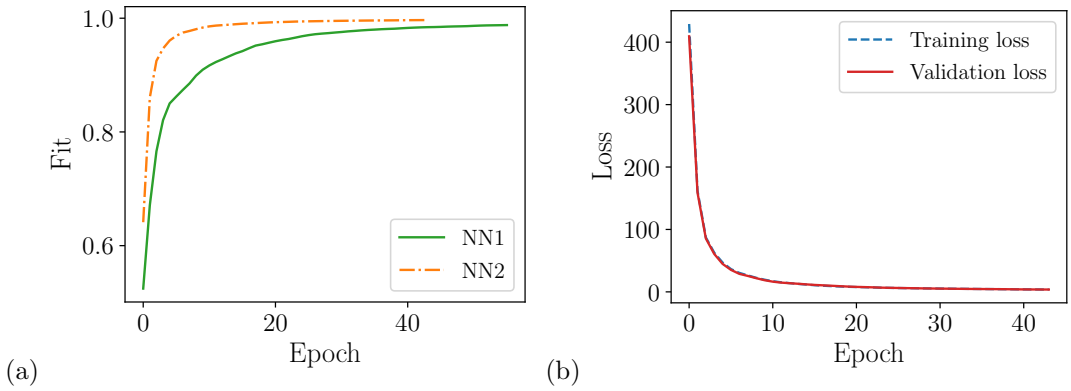

FIG. 6. Evolution of the fit parameter (a) and of the mean losses (b) throughout the iterations.

In a first stage, we want to predict the following non-dimensional correction term $\Delta\nu_t/\nu$ given by NN1. Figure 10 displays the distribution of the mean pressure and skin-friction coefficients for the reference solution, baseline RANS and NN1-based RANS. Despite the good prediction of the difference between the correct and modeled eddy viscosities (Fig. 7), when this quantity is given as an input for a RANS solver, the resulting flow field comes out noisy. However, we note that the shape of the C_p and C_f distributions are correctly captured by the augmented model. The new NN1-model also manages to predict the separation and reattachment locations in both scenarios, which is quite impressive taking into consideration that they present different flow physics. It is important mentioning that in order to have an exploitable model, two corrections needed to be implemented in the solver: (i) we needed to ensure that the total eddy viscosity was always positive and (ii) to enforce a zero viscosity at the wall as in the original SA model. These two considerations were necessary in order to bypass numerical instabilities or nonphysical results in the wall region (see Fig. 11).

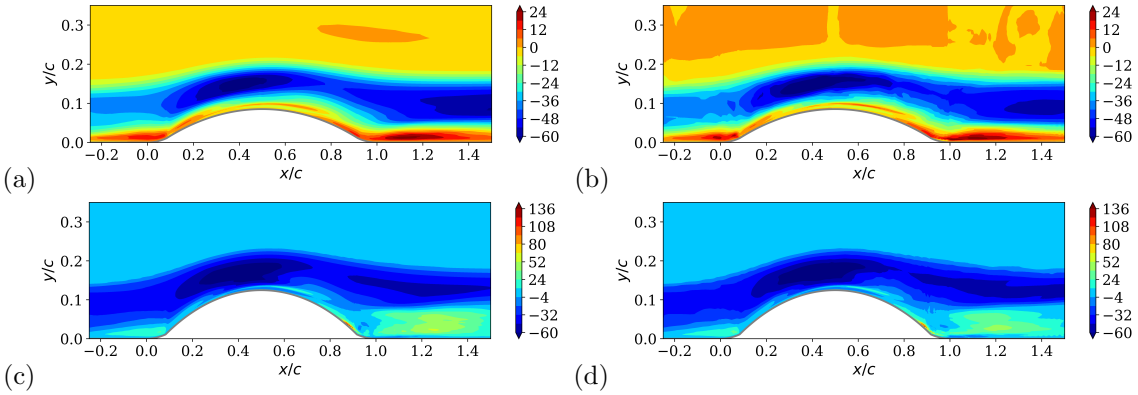

FIG. 7. Comparison of turbulent viscosities for cases h26 (top) and h38 (bottom). Left: $\Delta\nu_t/\nu$. Right: $\Delta\nu_t^{NN}/\nu$ predicted by NN1.

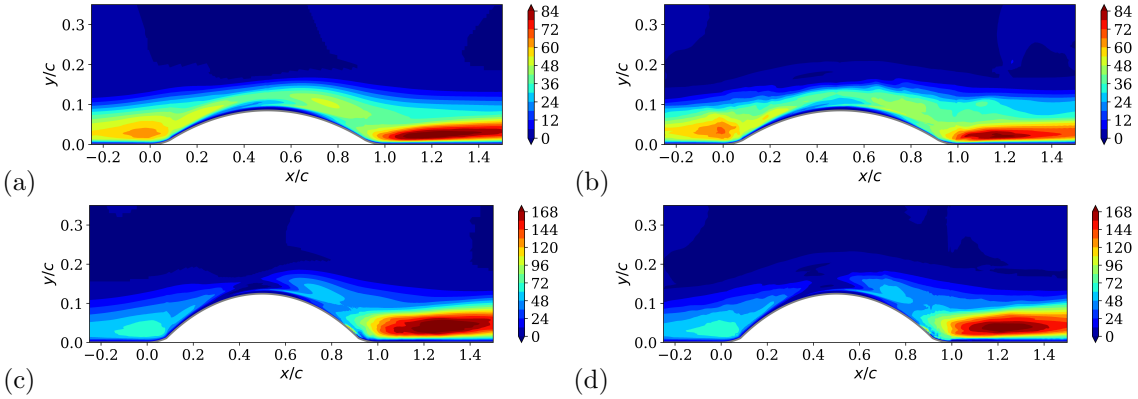

FIG. 8. Comparison of turbulent viscosities for cases h26 (top) and h38 (bottom). Left: ν_t^{LES}/ν. Right: ν_t^{NN}/ν predicted by NN2.

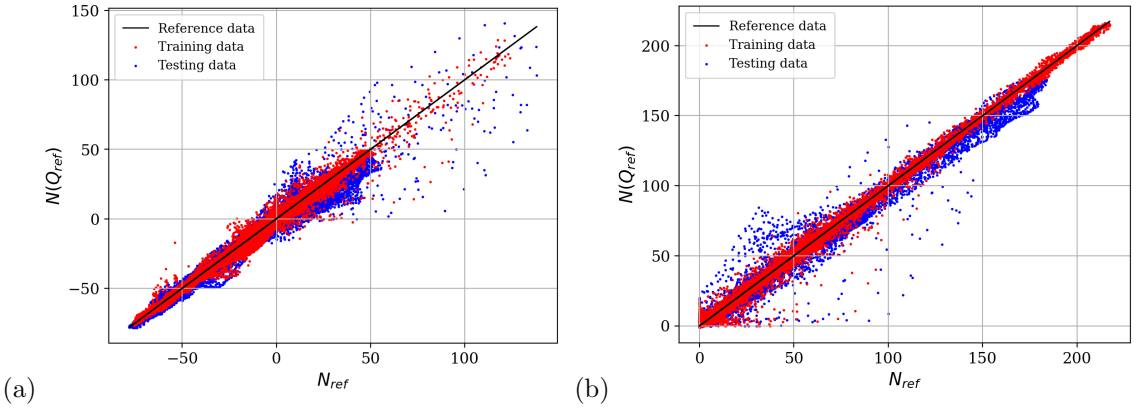

FIG. 9. Normalized output quantities from neural network NN1 (a) and NN2 (b). The scatter points should approximate to the solid line plotted as reference.

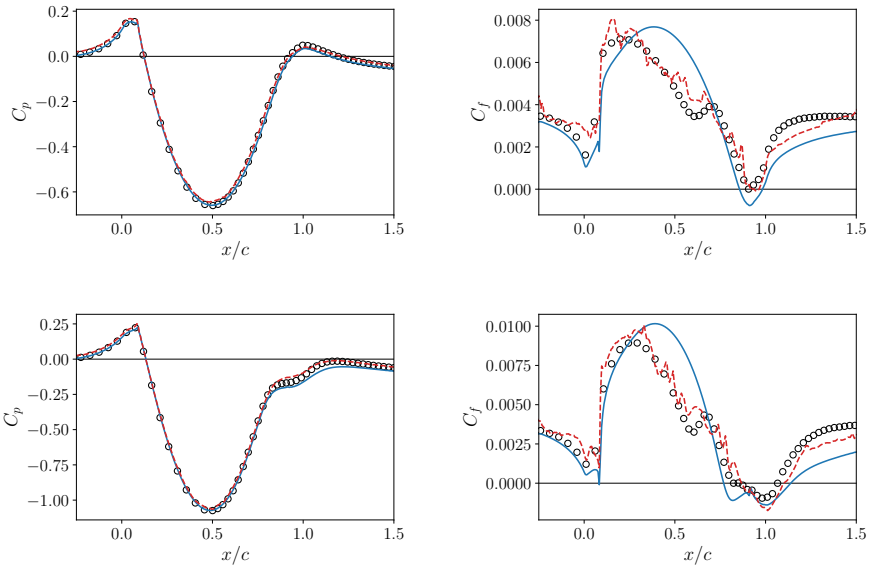

FIG. 10. Pressure (left) and skin friction (right) coefficients for testing cases h26 (top) and h38 (bottom). Reference LES simulation (symbols), baseline RANS-SA (solid lines), RANS-NN1 (dashed lines)

Results concerning the NN2-based model is shown in Fig. 12 for test cases h26 and h38. Despite not being perfect, this second model successfully predicts a flow with incipient separation for case h26 and the correct size of the separation bubble for case h38 and the results are much smoother than in the former simulation. Note also that, arguably, simulation h38 performed slightly better than h26. This can be explained by the fact that in the training database only one configuration presented attached flow, while the other two presented some kind of separation. Therefore, the majority of the training set was closer to configuration h38 rather than h26. We believe that the overall prediction could be improved if more cases were considered. We plot in Fig. 13 the streamwise velocity profiles for the LES, RANS-SA and RANS-NN2 for validation purposes. We note that the overall flow field is improved and resembles the one from simulation using ν_t^{LES} (Fig. 5). We conclude that predicting ν_t^{NN} using a neural network and subsequently adding the difference $\nu_t^{NN} - \nu_t^{SA}$ to the SA field inside the RANS solver gives better results than predicting this difference $\Delta\nu_t^{NN}$ directly with the neural network.

One of the reviewers questioned the performance of the NN-based model to extrapolate to configurations with larger heights. Unfortunately the original database does not cover cases larger than h42. So, to answer this question, we opted to train a new model (named NN3) using configurations h20, h26, h31 and h38. Having four cases (instead of three) in the training set enriches the resulting model and makes it unbiased (two configurations present attached and two configurations

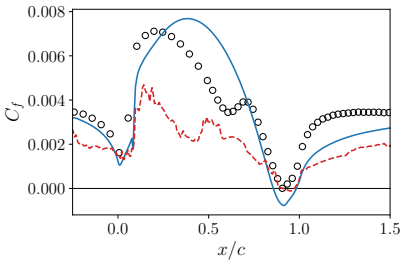

FIG. 11. Skin friction coefficient for testing case h26 without the null viscosity at the wall. Reference LES simulation (symbols), baseline RANS-SA (solid line), RANS-NN1 (dashed line)

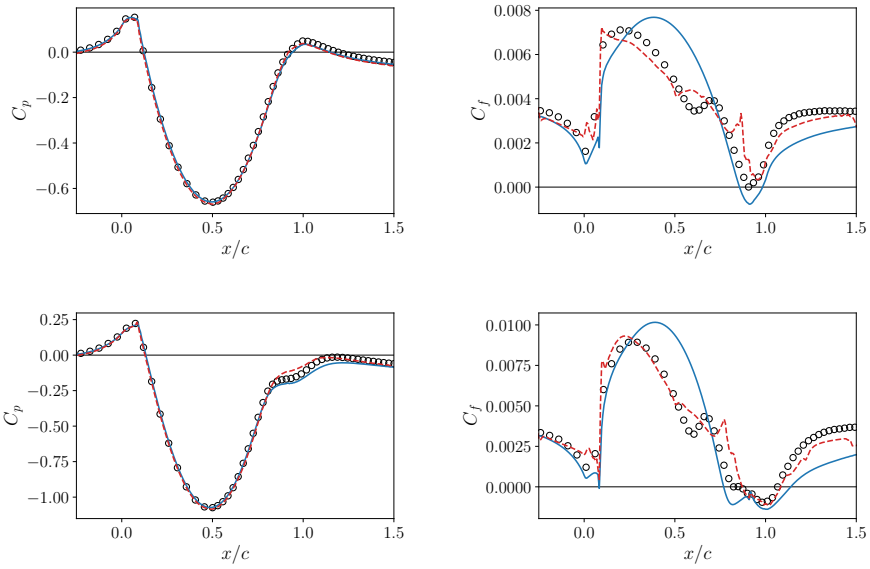

FIG. 12. Pressure (left) and skin friction (right) coefficients for testing cases h26 (top) and h38 (bottom). Reference LES simulation (symbols), baseline RANS-SA (solid lines), RANS-NN2 (dashed lines)

present separated flows). The performance of NN3 model is displayed in Table IV. Results are outstanding for the training cases (fit parameter over 99%) and for the validation case (fit parameter around 98 %). Testing the NN3 model in a RANS solver also showed excellent performance if compared to the baseline RANS-SA simulation (Fig. 14). Surprisingly, the results are even more accurate than the ones displayed in Fig. 5. So, in some way, the neural network managed to find an improved function that better described the validation set. We conclude that machine-learning assisted turbulence models can have a great impact in the prediction of key flow features.

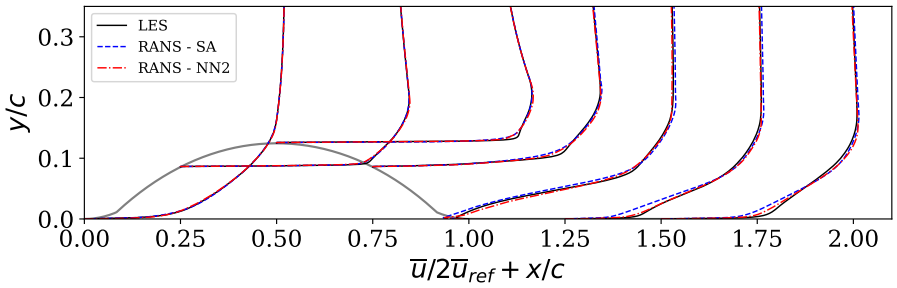

FIG. 13. Profiles of streamwise velocity component for case h38. Profiles are shown for the reference LES simulation (solid), baseline RANS-SA (dashed) and RANS-NN2 (dashed-dotted line).

Label	Output variable	NN structure	fit (%)					
			$h20$ (T)	$h26$ (T)	$h31$ (T)	$h38$ (T)	$h42$ (V)	all
NN3	ν_t/ν	$8 \times (4 \times 80) \times 1$	99.7	99.8	99.8	99.7	98.0	99.0

TABLE IV. Evaluation of the NN performance through the fit parameter. (T) indicates the training cases and (V) the verification cases not used during training. NN structure is in the form $N_{inputs} \times (N_{layers} \times N_{neurons}) \times N_{outputs}$, where N_{layers} is the number of hidden layers and $N_{neurons}$ the number of neurons per hidden layer.

VII. SUMMARY

In this work, a data-driven turbulence model has been developed based on artificial neural networks to improve the accuracy of classical RANS models. The present model was conceived within the constraints of the Boussinesq hypothesis, and it aims at correcting the prediction of the turbulent viscosity field when compared to the baseline Spalart-Allmaras model. Taking the

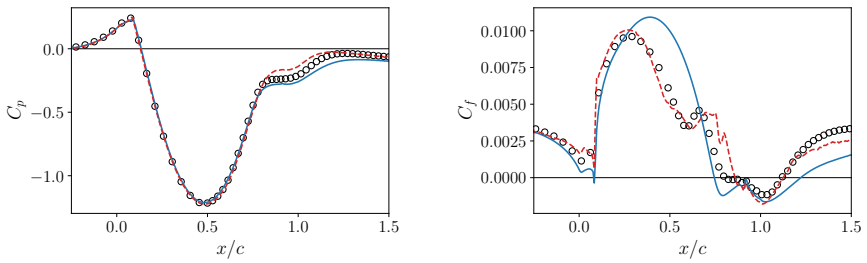

FIG. 14. Pressure (left) and skin friction (right) coefficients for testing case h42. Reference LES simulation (symbols), baseline RANS-SA (solid lines), RANS-NN3 (dashed lines)

LES solution as the true field, it was observed that for a bump configuration the baseline model mispredicts the pressure and velocity fields, reflecting on a significant difference in the C_f evolution above the bump. To enhance the model prediction, the neural network is treated as a non-linear approximator that corresponds to an estimated functional form of the turbulent viscosity or its discrepancy. This form is defined through supervised learning, where the network parameters are optimized such that the function fits to the given reference data-set. In this way, the novelty of the proposed model is that it corrects directly the eddy-viscosity field and no additional turbulence quantity needs to be transported, leading to a PDE-free eddy-viscosity model.

Therefore, in a first step, the reference turbulent viscosity was extracted from the true fields of LES simulations. This method lead to an important augmentation of the resulting RANS solution, generating a more accurate description of C_p and C_f. In a second step, a set of local input features were formulated following physical reasoning and modeling constraints. Two output quantities: the turbulence viscosity discrepancy $\Delta\nu_t = \nu_t^{LES} - \nu_t^{SA}$ and the expected turbulence viscosity itself ν_t^{LES} were inferred using artificial neural networks. Both NN architectures seemed to give good prediction for each quantity of interest. The predicted field still contains noise due to the approximate nature of the neural networks, which might be a problem when injected in a RANS solver. Subsequently, both neural networks were tested in two scenarios never seen in the training process. Results obtained with NN1 that predicted the eddy-viscosity discrepancy did show an improved solution, despite the noisy skin-friction profile. On the other hand, NN2 that focused on predicting the correct eddy viscosity was able to reproduce the general flow-field behaviour with a much smoother solution. The method was proved to be robust even in predicting extrapolated flows. The results achieved using the proposed method are encouraging, but new analysis are still necessary to improve *a posteriori* computations.

ACKNOWLEDGMENTS

This work was supported by the European project HiFi-TURB (High Fidelity LES/DNS Data for Innovative Turbulence Models) - EU H2020 RGY under Grant Agreement No. 814837 and the internal project MODDA (Modèles Data-Driven pour l'Aérodynamique.)

---

* pedro.stefanin_volpiani@onera.fr

[1] Philippe Spalart and Steven Allmaras, "A one-equation turbulence model for aerodynamic flows," (1994) pp. 5–21.

[2] Florian R Menter, "Two-equation eddy-viscosity turbulence models for engineering applications," AIAA journal **32**, 1598–1605 (1994).

[3] W Peter Jones and Brian Edward Launder, "The prediction of laminarization with a two-equation model of turbulence," International journal of heat and mass transfer **15**, 301–314 (1972).

[4] Brian Edward Launder and Bahrat I Sharma, "Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc," Letters in heat and mass transfer **1**, 131–137 (1974).

[5] David C Wilcox *et al.*, *Turbulence modeling for CFD*, Vol. 2 (DCW industries La Canada, CA, 1998).

[6] Werner Haase, Bertrand Aupoix, Ulf Bunge, and Dieter Schwamborn, *FLOMANIA-a European initiative on flow physics modelling: results of the European-Union funded project, 2002-2004*, Vol. 94 (Springer Science & Business Media, 2006).

[7] Steven L Brunton, Bernd R Noack, and Petros Koumoutsakos, "Machine learning for fluid mechanics," Annual Review of Fluid Mechanics **52**, 477–508 (2020).

[8] Karthik Duraisamy, "Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence," Physical Review Fluids **6**, 050504 (2021).

[9] Richard D Sandberg and Yaomin Zhao, "Machine-learning for turbulence and heat-flux model development: A review of challenges associated with distinct physical phenomena and progress to date," International Journal of Heat and Fluid Flow **95**, 108983 (2022).

[10] Brendan Tracey, Karthik Duraisamy, and Juan Alonso, "Application of supervised learning to quantify uncertainties in turbulence and combustion modeling," in *51st AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition* (2013) p. 259.

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

[12] Julia Ling, Reese Jones, and Jeremy Templeton, "Machine learning strategies for systems with invariance properties," Journal of Computational Physics **318**, 22–35 (2016).

[13] Jian-Xun Wang, Jin-Long Wu, and Heng Xiao, "Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data," Physical Review Fluids **2**, 034603 (2017).

[14] Jin-Long Wu, Heng Xiao, and Eric Paterson, "Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework," Physical Review Fluids **3**, 074602 (2018).

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

[16] J Weatheritt and RD Sandberg, "The development of algebraic stress models using a novel evolutionary algorithm," International Journal of Heat and Fluid Flow **68**, 298–318 (2017).

[17] Yaomin Zhao, Harshal D Akolekar, Jack Weatheritt, Vittorio Michelassi, and Richard D Sandberg, "RANS turbulence model development using cfd-driven machine learning," Journal of Computational Physics **411**, 109413 (2020).

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

[19] Anand Pratap Singh, Shivaji Medida, and Karthik Duraisamy, "Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils," AIAA Journal , 2215–2227 (2017).

[20] Lucas Franceschini, Denis Sipp, and Olivier Marquet, "Mean-flow data assimilation based on minimal correction of turbulence models: Application to turbulent high Reynolds number backward-facing step," Physical Review Fluids **5**, 094603 (2020).

[21] Pedro Stefanin Volpiani, Morten Meyer, Lucas Franceschini, Julien Dandois, Florent Renac, Emeric Martin, Olivier Marquet, and Denis Sipp, "Machine learning-augmented turbulence modeling for RANS simulations of massively separated flows," Physical Review Fluids **6**, 064607 (2021).

[22] DR Webster, DB DeGraaff, and JK Eaton, "Turbulence characteristics of a boundary layer over a two-dimensional bump," Journal of Fluid Mechanics **320**, 53–69 (1996).

[23] Xiaohua Wu and Kyle D Squires, "Numerical investigation of the turbulent boundary layer over a bump," Journal of Fluid Mechanics **362**, 229–271 (1998).

[24] Racheet Matai and Paul Durbin, "Large-eddy simulation of turbulent flow over a parametric set of bumps," Journal of Fluid Mechanics **866**, 503–525 (2019).

[25] René-Daniel Cécora, Rolf Radespiel, Bernhard Eisfeld, and Axel Probst, "Differential Reynolds-stress modeling for aeronautics," AIAA Journal **53**, 739–755 (2015).

[26] Bernhard Eisfeld, Chris Rumsey, and Vamshi Togiti, "Verification and validation of a second-moment-closure model," AIAA Journal **54**, 1524–1541 (2016).

[27] François G Schmitt, "About Boussinesq's turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity," Comptes Rendus Mécanique **335**, 617–627 (2007).

[28] F. Hecht, "New development in freefem++," J. Numer. Math. **20**, 251–265 (2012).

[29] Racheet Matai and Paul Durbin, "NASA turbulence modeling resource," (2019).

[30] Clément Mettot, Denis Sipp, and Hervé Bézard, "Quasi-laminar stability and sensitivity analyses for turbulent flows: prediction of low-frequency unsteadiness and passive control," Physics of Fluids **26**, 061701 (2014).

[31] Roney L Thompson, Luiz Eduardo B Sampaio, Felipe AV de Bragança Alves, Laurent Thais, and Gilmar Mompean, "A methodology to evaluate statistical errors in DNS data of plane channel flows," Computers & Fluids **130**, 1–7 (2016).

[32] Hassan Raiesi, Ugo Piomelli, and Andrew Pollard, "Evaluation of turbulence models using direct numerical and large-eddy simulation data," Journal of Fluids Engineering **133** (2011).

[33] Jianheng Tan, Xiao He, Georgios Rigas, and Mehdi Vahdati, "Towards explainable machine-learning-assisted turbulence modeling for transonic flows," in *14 th European Conference on Turbomachinery*

Fluid dynamics & Thermodynamics (2021).

[34] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)* (2010) pp. 807–814.

[35] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, Vol. 9 (PMLR, 2010) pp. 249–256.

[36] Julia Ling and J Templeton, "Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty," Physics of Fluids **27**, 085103 (2015).

[37] C Gorlé, Johan Larsson, Michael Emory, and Gianluca Iaccarino, "The deviation from parallel shear flow as an indicator of linear eddy-viscosity model inaccuracy," Physics of Fluids **26**, 055105 (2014).

[38] Mortaza Mani, Deric Babcock, Chad Winkler, and Philippe Spalart, "Predictions of a supersonic turbulent flow in a square duct," in *51st AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition* (2013) p. 860.