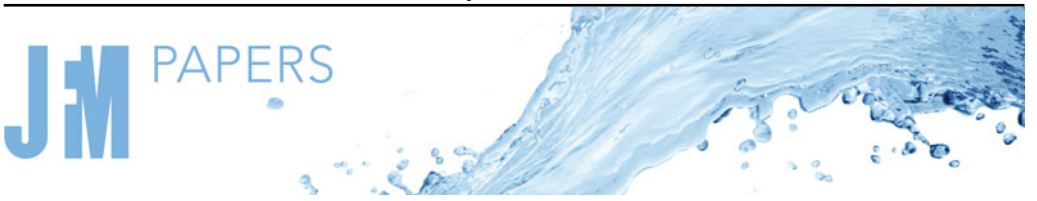
J. Fluid Mech. (2026), *vol.* 1033, A15, doi:10.1017/jfm.2026.11414



Data-driven turbulence closure using a tensor basis neural network without reliance on baseline models

Yao-Hung Tsai[1], Yi Wang[1], Che-Jung Cheng[1], Chen-Yen Hung[1], Yi-En Chou[1], Kuan-Lin Li[3], Chien-Chou Tseng[3], Ruey-Lin Chern[1] and Yi-Ju Chou[1,2]

[1]Institute of Applied Mechanics, National Taiwan University, Taipei, Taiwan

[2]Ocean Center, National Taiwan University, Taipei, Taiwan

[3]Department of Mechanical Engineering, National Cheng Kung University, Tainan, Taiwan

Corresponding author: Yi-Ju Chou, yjchou@iam.ntu.edu.tw

(Received 9 June 2025; revised 6 February 2026; accepted 9 March 2026)

We propose a novel machine-learning-based turbulence closure framework in which a tensor basis neural network (TBNN) is directly embedded into a Reynolds-averaged Navier–Stokes (RANS) formulation, eliminating reliance on traditional baseline turbulence models. The TBNN is trained to predict the Reynolds stress anisotropic tensor from local invariant inputs and geometry-informed features, including stream function and velocity potential. Its output is processed by a regression model that generates an optimised eddy viscosity field, which is then integrated into the RANS equations as a zero-equation turbulence closure. The framework is evaluated on three turbulent flows over complex geometries: a wavy-bottom channel, a smoothed step and a backward-facing step. Incorporating geometry-informed features significantly enhances model robustness, yielding numerically stable and convergent solutions across all cases. The predicted velocity fields and turbulence distributions closely match large eddy simulation (LES) data, confirming the accuracy of the proposed approach and demonstrating its ability to operate independently of conventional turbulence closures.

Key words: machine learning, turbulence modelling

1. Introduction

Machine learning (ML) techniques have rapidly advanced in fluid dynamics, enabling the development of surrogate models that can predict complex flow fields without directly solving the governing equations (Brunton, Noack & Koumoutsakos 2020; Vinuesa & Brunton 2022). These data-driven models significantly accelerate computations and have

© The Author(s), 2026. Published by Cambridge University Press.

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

demonstrated impressive accuracy within their training regimes. However, as largely end-to-end black-box predictors, they have been widely questioned for their physical interpretability and generalisability (Sharma *et al.* 2023; Zhao *et al.* 2024). In particular, when dealing with complex physical phenomena, neural network models often struggle to elucidate the causal relationships between inputs and outputs, and their predictive performance can deteriorate significantly when applied to conditions outside the training data. Such limitations pose substantial challenges for engineering applications that demand reliability, extrapolation capability and consistency with physical principles.

One promising route is to embed ML within the Reynolds-averaged Navier–Stokes (RANS) modelling framework, a strategy of hybrid modelling that retains the governing equations while applying ML to solve the turbulence closure problem (Duraisamy, Iaccarino & Xiao 2019; Beck & Kurz 2021). Such approaches preserve the robustness of the governing equations and leverage ML to enhance model flexibility. RANS remains the dominant tool for industrial turbulent flow simulations due to its favourable balance of computational efficiency and numerical stability compared with large eddy simulation (LES) or direct numerical simulation (DNS). In addition to enforcing conservation of mass and momentum, RANS models provide a degree of physical interpretability, which helps mitigate the limitations of black-box models in terms of physical consistency and generalisability. Introducing ML-based closures can further reduce the reliance on empirical parameters in traditional RANS models and improve prediction accuracy in high-gradient or separated flow regions (Duraisamy 2021). As a result, hybrid architectures that combine RANS with data-driven methods present a feasible path towards advancing the capabilities of current CFD models.

In recent years, there has been a surge of interest in applying ML to turbulence closure modelling. ML models are increasingly used to infer the relationship between flow features and the Reynolds stress based on high-fidelity data. Successful strategies incorporate physical knowledge into the learning process to ensure that the learned model reflects essential physical characteristics of turbulence. Data-driven RANS modelling approaches can generally be categorised into two main frameworks (Duraisamy *et al.* 2019). The first is the tensor-basis invariant modelling approach, which builds on the invariant theory proposed by Pope (1975). This theory provides a complete basis of ten independent tensors, constructed from the strain-rate and rotation-rate tensors, to represent the Reynolds stress anisotropy. The ML model is used to predict the coefficients of this tensor basis expansion. By construction, this ensures that the output tensor satisfies Galilean invariance and other symmetry properties. Ling, Kurzawski & Templeton (2016*b*) pioneered this approach with the tensor basis neural network (TBNN), introducing a deep network architecture with an invariant tensor basis layer to embed Galilean invariance into the learned anisotropic tensor mapping. By comparing it against a standard neural network, they demonstrated that this invariance-preserving architecture improves Reynolds stress predictions and leads to more accurate mean velocity fields.

Following TBNN, various studies have extended and analysed this approach. For example, Milani, Ling & Eaton (2021) applied a similar tensor-basis network to model turbulent scalar fluxes, while Zhang *et al.* (2022) proposed a nonlinear eddy-viscosity model with an embedded structure trained using an ensemble Kalman method. Zhang *et al.* (2023) further investigated the physical interpretability and basis properties of such neural network models. In parallel, researchers have also explored explicit invariant representations via evolutionary algorithms. Weatheritt & Sandberg (2016) used genetic programming to symbolically derive algebraic stress models as combinations of invariant terms, yielding closed-form expressions that were successfully applied to expanding duct flows (Weatheritt & Sandberg 2017) and turbomachinery blade rows (Zhao *et al.* 2020).

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

The second framework is based on the Lumley triangle (Lumley 1979), which characterises the state of turbulence through the invariants of the Reynolds stress eigenvalues. Specifically, Banerjee *et al.* (2007) introduced a convenient barycentric map of the Lumley triangle that represents any anisotropy state as a point within a normalised triangle. Using this representation, one can quantify the 'distance' of a turbulence state from isotropy and ensure that ML predictions remain within physically realisable bounds. Wang, Wu & Xiao (2017) trained a Random Forest model to map the discrepancy between a baseline RANS model and high-fidelity data in terms of anisotropic invariants, projecting the predicted error onto the Lumley triangle. Building on this idea, Wu, Xiao & Paterson (2018) incorporated both linear and nonlinear correction terms trained on RANS–DNS discrepancies to develop a more accurate correction model. Xiao *et al.* (2020) further investigated the behaviour of this method under both interpolation and extrapolation scenarios. One key advantage of the Lumley triangle approach is that it enforces the realisability of the Reynolds stress and provides a clear geometric visualisation of turbulence anisotropy, which aids in both the interpretation and constraint of ML models. Both TBNN and Lumley triangle frameworks demonstrate strong potential and underscore the importance of embedding theoretical physical insights into ML-based turbulence modelling.

Among the growing class of data-driven turbulence modelling techniques, in addition to the aforementioned methods, field inversion and machine learning (FIML) has emerged as a particularly effective approach for improving Reynolds-averaged Navier–Stokes (RANS) predictions in a data-consistent manner. In FIML, the turbulence model is augmented by spatially varying correction fields, often applied to quantities such as eddy viscosity coefficients, production terms or source functions, which are inferred by solving an optimisation problem that minimises the discrepancy between RANS model predictions and available reference data (typically experimental measurements or high-fidelity simulation results) (Parish & Duraisamy 2016; Duraisamy *et al.* 2019). These correction fields can subsequently be expressed as functions of local flow features via machine learning regression, yielding generalised closures that can be deployed across similar flows.

In developing data-driven turbulence closure models, it is essential to distinguish between the *a priori* and *a posteriori* stages, as they serve different objectives. In the *a priori* stage, high-fidelity data are used to train and validate ML models, with a focus on their ability to predict the Reynolds stress or its discrepancy under a given mean flow field. The objective is to learn an accurate mapping that closely reproduces the target data. Many studies have emphasised that feature selection is critical at this stage. The features must be rich enough to capture relevant turbulence physics, yet respect symmetry properties to avoid learning spurious, frame-dependent relationships. For example, Ling, Jones & Templeton (2016*a*) emphasised the use of Galilean-invariant features to improve learning efficiency and reduce frame-dependent errors. Wang *et al.* (2017) further evaluated various candidate turbulence features and identified those most important for predicting Reynolds stress discrepancies. Most of these works rely on local turbulence features as input. However, using only local features can lead to an ill-posed mapping problem. When the training data span only a narrow range of flow conditions, a given set of local features may correspond to multiple possible Reynolds stress states, resulting in ambiguity or instability in the predictions (Zhang *et al.* 2023). This issue often manifests as scatter or multi-valued behaviour in the learned functional relationship, and such ill-posedness ultimately limits the model's generalisability.

The *a posteriori* stage refers to integrating the learned closure model into a RANS solver and assessing its performance in producing a self-consistent flow solution.

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

The ultimate test of a turbulence model is *a posteriori*, which demonstrates whether it can be stably and accurately integrated into the RANS equations. Existing works in ML-based turbulence modelling typically relied on coupling the learned model with an existing baseline RANS model. The baseline RANS model, hereafter referred to as the baseline model, is the solution obtained from the RANS formulation using the same parameters as the predicted case and involves solving traditional two-equation models, such as the k–ϵ and k–ω models, for turbulence closure. For example, Ling *et al.* (2016*b*) incorporated the anisotropic tensor predicted by TBNN into an eddy-viscosity framework within their in-house RANS solver. Weatheritt & Sandberg (2016) embedded their symbolic explicit algebraic stress model into the standard k–ω SST model to evaluate performance. Similarly, Wu *et al.* (2018) required a baseline model, training a correction model to predict the difference between the baseline RANS stresses and high-fidelity stresses. During RANS solving, this correction was added on top of the baseline model's prediction. In all of these studies, the data-driven closure model relies on an existing baseline turbulence model (Duraisamy *et al.* 2019; Mandler & Weigand 2022).

More recent advances have focused on CFD-driven optimisation strategies (Waschkowski *et al.* 2022; Agrawal & Koutsourelakis 2024), where model training is based on minimising discrepancies in flow fields (such as velocity or pressure) rather than directly matching Reynolds stresses. This paradigm shift significantly reduces the role of baseline models during training. However, turbulence transport equations from baseline models remain necessary to compute turbulence scales for input standardisation. This dependency introduces an additional source of error and often requires retraining or retuning for different flow cases or baseline models. In short, such ML-based closures cannot operate independently. They must be used in conjunction with a specific conventional model, which limits their generalisation ability and robustness.

Building on the insights discussed previously, we pose the following question. Is it possible to develop an ML-based turbulence closure model for RANS that operates like a traditional closure model – without requiring additional baseline computations? This study aims to answer that question by developing a self-contained, ML-based turbulence closure model that can be directly integrated into a RANS solver without dependence on any baseline model. To this end, we make two key contributions that address the challenges encountered in both the *a priori* learning and *a posteriori* deployment phases of data-driven turbulence modelling. First, to enhance the expressiveness and robustness of the *a priori* model, we introduce geometry-informed features as additional inputs to the neural network. Specifically, in addition to the original local invariant features, we incorporate the stream function and velocity potential, which are obtained in a straightforward manner prior to solving the governing equations. We demonstrate that including these geometry-informed features improves predictive accuracy. Second, we propose an explicit turbulence model that can be integrated into a RANS solver without relying on any baseline turbulence model. Starting from the general theoretical framework of the linear eddy viscosity model, a set of explicit algebraic relations is derived and formulated as a zero-equation framework, enabling closure of the RANS equations without the need for additional transport equations. Based on these expressions, a neural network is constructed to directly predict the Reynolds stress anisotropic tensor in the *a priori* stage. In the *a posteriori* stage, an additional ML model is employed to generate an optimised turbulent viscosity field during the RANS solution process. This implementation eliminates the need for estimates from baseline models and enables direct integration into the RANS solver. By removing reliance on predefined turbulence models, this approach avoids model-specific biases, enabling the ML model to function as an independent closure mechanism. It is demonstrated that this *a posteriori* integration yields stable, converged solutions and

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

improves accuracy over traditional models, indicating that the learned model can serve as a reliable surrogate for turbulence closure in RANS.

The remainder of this paper is organised as follows. Section 2 details the proposed ML-based turbulence modelling framework. We begin by reviewing the RANS governing equations and the conventional closure problem. We then describe the architecture of the TBNN-based network. In § 3, the process of obtaining an optimised turbulent viscosity field from the predicted Reynolds stress is then explained. In § 4, we outline the dataset used. The implementation details for coupling the neural network with a RANS solver for *a posteriori* testing is also provided. Section 5 presents the demonstration results and discussion. We evaluate the model on cases of flow over a periodic wavy bottom, a smoothed step and a backward-facing step. Finally, concluding remarks are given in § 6.

2. Mathematical formulation

2.1. *Governing equations*

The steady RANS equations describe the conservation of mass and momentum for incompressible flows in the mean field. They can be written as

$$\frac{\partial U_i}{\partial x_i} = 0\,, \tag{2.1}$$

$$U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\nu \frac{\partial U_i}{\partial x_j}\right) - \frac{\partial \tau_{ij}}{\partial x_j}\,, \tag{2.2}$$

where the velocity field is decomposed as $u_i \equiv U_i + u_i'$ into a mean component U_i and a fluctuation u_i'. Here, P is the mean pressure, ν is the kinematic viscosity and ρ is the fluid density. The Reynolds stress τ_{ij}, defined as the mean of the fluctuation quantity $u_i' u_j'$, represents the effects of turbulence on the mean flow and makes the system underdetermined. This gives rise to the turbulence closure problem, necessitating the development of additional turbulence models to express τ_{ij} in terms of known mean-flow variables.

The Reynolds stress tensor τ_{ij} is typically decomposed into an isotropic component $(2/3)k\,\delta_{ij}$ and an anisotropic component a_{ij}, i.e.

$$\tau_{ij} = \frac{2}{3}k\delta_{ij} + a_{ij}\,, \tag{2.3}$$

where k is the turbulent kinetic energy and δ_{ij} is the Kronecker delta. The isotropic part can be absorbed into the pressure term and it is sufficient to model only the anisotropic component a_{ij} for closing the RANS equations. To do this, following the previous studies in this context, we start from the effective viscosity model proposed by Pope (1975) , which is expressed as

$$b_{ij} = \frac{a_{ij}}{k} = b_{ij}\left(\boldsymbol{S}, \boldsymbol{\Omega}\right), \tag{2.4}$$

where

$$S_{ij} = \frac{1}{2}\frac{k}{\epsilon}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) \tag{2.5}$$

and

$$\Omega_{ij} = \frac{1}{2}\frac{k}{\epsilon}\left(\frac{\partial U_i}{\partial x_j} - \frac{\partial U_j}{\partial x_i}\right) \tag{2.6}$$

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

$$\boldsymbol{T}^{(1)} = \boldsymbol{S} \qquad \boldsymbol{T}^{(2)} = \boldsymbol{S\Omega} - \boldsymbol{\Omega S} \qquad \boldsymbol{T}^{(3)} = \boldsymbol{\Omega}^2 - \frac{1}{3}\boldsymbol{I}\,\mathrm{tr}(\boldsymbol{S}^2)$$

$$\boldsymbol{T}^{(4)} = \boldsymbol{\Omega}^2 - \frac{1}{3}\boldsymbol{I}\,\mathrm{tr}(\boldsymbol{\Omega}^2) \qquad \boldsymbol{T}^{(5)} = \boldsymbol{\Omega S}^2 - \boldsymbol{S}^2\boldsymbol{\Omega} \qquad \boldsymbol{T}^{(6)} = \boldsymbol{\Omega}^2\boldsymbol{S} + \boldsymbol{S\Omega}^2 - \frac{2}{3}\boldsymbol{I}\,\mathrm{tr}(\boldsymbol{S\Omega}^2)$$

$$\boldsymbol{T}^{(7)} = \boldsymbol{\Omega S\Omega}^2 - \boldsymbol{\Omega}^2\boldsymbol{S\Omega} \qquad \boldsymbol{T}^{(8)} = \boldsymbol{S\Omega S}^2 - \boldsymbol{S}^2\boldsymbol{\Omega S} \qquad \boldsymbol{T}^{(9)} = \boldsymbol{\Omega}^2\boldsymbol{S}^2 + \boldsymbol{S}^2\boldsymbol{\Omega}^2 - \frac{2}{3}\boldsymbol{I}\,\mathrm{tr}(\boldsymbol{S}^2\boldsymbol{\Omega}^2)$$

$$\boldsymbol{T}^{(10)} = \boldsymbol{\Omega S}^2\boldsymbol{\Omega}^2 - \boldsymbol{\Omega}^2\boldsymbol{S}^2\boldsymbol{\Omega}$$

Table 1. Definitions of the ten integrity basis tensors following Pope (1975). In this table, $\mathrm{tr}(\boldsymbol{X})$ denotes the trace of a tensor $\boldsymbol{X}$, $\boldsymbol{S}$ and $\boldsymbol{\Omega}$ are the dimensionless strain-rate and rotation-rate tensors, respectively, and $\boldsymbol{I}$ is the identity tensor.

are strain rate and rotation rate tensors, respectively, and ϵ is the turbulent dissipation. The dimensionless form of (2.4) allows it to be expressed using tensor representation (Pope 1975), i.e.

$$b_{ij} = \sum_{k=1}^{10} G^{(k)} T_{ij}^{(k)}, \tag{2.7}$$

where T_{ij}^{k} are the integrity basis, as listed in table 1, and $G^{(k)}$ are the corresponding coefficients. While more details to obtain G^{k} in the present model framework are provided in § 2.2, in what follows, we present rescaling for (2.4) using the characteristic flow quantities.

Invoking the second-moment quantities in (2.4) through (2.6) necessitates the solution of scalar transport equations in traditional two-equations models, such as $k{-}\epsilon$ and $k{-}\omega$ models. In existing studies of ML-based turbulence closure model, this can be easily obtained from the baseline model (Zhang *et al.* 2022; Agrawal & Koutsourelakis 2024). However, while the data-driven model is aimed to eliminate the use of *ad hoc* parameters in the traditional closure model, contradictions arise when the scaling quantities (i.e. k) are obtained from the traditional closure model with untuned *ad hoc* parameters.

To mitigate this issue, we introduce $\mathcal{U}$ and $\mathcal{L}$ as the flow velocity and length scales, respectively. Based on these two scales, one may rewrite the eddy viscosity model (2.4) as

$$b_{ij}^{*} = \frac{a_{ij}}{\mathcal{U}^2} = \frac{k}{\mathcal{U}^2} b_{ij}\left(\boldsymbol{S}, \Omega\right), \tag{2.8}$$

where the superscript $*$ indicates the rescaled dimensionless quantities using characteristic scales, $\mathcal{U}$ and $\mathcal{L}$. In (2.8), the strain rate and rotation rate tensors are written in terms of rescaled quantities as

$$S_{ij} = \frac{1}{2}\frac{k}{\epsilon}\frac{\mathcal{U}}{\mathcal{L}}\left(\frac{\partial U_i^{*}}{\partial x_j^{*}} + \frac{\partial U_j^{*}}{\partial x_i^{*}}\right) = \frac{k}{\epsilon}\frac{\mathcal{U}}{\mathcal{L}} S_{ij}^{*} \tag{2.9}$$

and

$$\Omega_{ij} = \frac{1}{2}\frac{k}{\epsilon}\frac{\mathcal{U}}{\mathcal{L}}\left(\frac{\partial U_i^{*}}{\partial x_j^{*}} - \frac{\partial U_j^{*}}{\partial x_i^{*}}\right) = \frac{k}{\epsilon}\frac{\mathcal{U}}{\mathcal{L}} \Omega_{ij}^{*}, \tag{2.10}$$

respectively.

In what follows, we apply the scaling argument for ϵ by Pope (2000), who adopts the concept in Richardson's view (Richardson 1922) of the energy cascade. We begin

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

by introducing u_0 and l_0 as the characteristic velocity and length scales associated with the largest eddies. The dissipation rate, ϵ, which represents the rate of transfer of energy from large to small scales, can thus be scaled as u_0^3/l_0. According to Pope (2000), the length scale l_0 is comparable to the flow scale $\mathcal{L}$, and the characteristic velocity scale u_0 is of the same order of the root mean square (r.m.s) turbulent intensity, $\sqrt{2k/3}$, which is comparable to $\mathcal{U}$. The second argument implies that $u_0 \sim \sqrt{k} \sim \mathcal{U}$ and proper scaling for ϵ can be expressed as

$$\epsilon \sim \frac{k\mathcal{U}}{\mathcal{L}}. \tag{2.11}$$

Hence, substituting (2.11) into (2.9) and (2.10), (2.8) can be expressed as

$$b_{ij}^* = b_{ij}^* \left(\boldsymbol{S}^*, \boldsymbol{\Omega}^* \right), \tag{2.12}$$

where $\boldsymbol{S}^*$ and $\boldsymbol{\Omega}^*$ are defined in (2.9) and (2.10), respectively. One may argue that the absence of k and ϵ in (2.9) and (2.10) could result in the loss of certain 'local features'. However, including them would hinder our goal of developing a direct RANS flow solver that incorporates the ML-based turbulence closure. Therefore, we omit them using a scaling argument and, instead, introduce two new flow features, stream function and velocity potential, for our ML model, detailed in § 2.3.

It is worthwhile to mention that one may choose to use only the scaling $u_0 \sim \sqrt{k}$, so that $\epsilon \sim u_0^3/l_0 \sim k^{1.5}/\mathcal{L}$. Substituting this into (2.9) and (2.10) yields a leading-order scaling coefficient of $\mathcal{U}/\sqrt{k}$ on the right-hand side, which can be scaled with a function of Reynolds number (Re) (Pope 2000). That is, if one disregards the scaling $\sqrt{k} \sim \mathcal{U}$, an alternative scaling emerges, resulting in a new scaling relationship, $S_{ij} = f(Re) S_{ij}^*$ and $\Omega_{ij} = f(Re)\Omega_{ij}^*$. Although this Re-dependent scaling is not required in our numerical examples, as presented in § 5, it may become more relevant when the training data spans a wide range of Reynolds numbers.

2.2. *Tensor-basis neural networks with embedded invariance*

Following Pope (1975), the dimensionless anisotropic stress tensor b_{ij} is expressed as a linear combination of ten integrity basis tensors $T_{ij}^{(k)}$ $(k = 1 \sim 10)$, each weighted by a scalar coefficient function $G^{(k)}$ $(k = 1 \sim 10)$ that depends on five scalar invariants λ_n $(n = 1{-}5)$, constructed from the strain-rate and rotation-rate tensors. These invariants and integrity bases are derived via the Cayley–Hamilton theorem to ensure uniqueness, linear independence and invariance under Galilean and rotational transformations.

Building on this theoretical foundation, the present study integrates the invariant theorem, and redefines the mapping using the characteristic scales $\mathcal{U}$ and $\mathcal{L}$. This yields a modified tensor representation, (2.8). The dimensionless anisotropic tensor can be further expressed as

$$b_{ij}^* = \sum_{k=1}^{10} G^{*(k)}\left(\lambda_n^* \mid n \in \{1{-}5\}\right)\ T_{ij}^{*(k)}, \tag{2.13}$$

where $T_{ij}^{*(k)}$ are the dimensionless integrity basis tensors and $G^{*(k)}$ are the corresponding scalar functions of the invariants λ_n^*, constructed from S_{ij}^* and R_{ij}^*. Definitions of these invariants and basis tensors are provided in table 2.

Ling *et al.* (2016*b*) later extended this framework by proposing the TBNN, which integrates the tensor-basis expansion with a deep neural network to inherently enforce

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

Tensor invariants				
$\lambda_1 = \mathrm{tr}(\boldsymbol{S}^2)$	$\lambda_2 = \mathrm{tr}(\boldsymbol{S}^3)$	$\lambda_3 = \mathrm{tr}(\boldsymbol{\Omega}^2)$	$\lambda_4 = \mathrm{tr}(\boldsymbol{\Omega}^2\boldsymbol{S})$	$\lambda_5 = \mathrm{tr}(\boldsymbol{\Omega}^2\boldsymbol{S}^2)$

Table 2. Definitions of the five tensor invariants following Pope (1975). In this table, normal-font symbols represent scalar quantities, while boldface symbols represent tensors. $\mathrm{tr}(\boldsymbol{X})$ denotes the trace of a tensor $\boldsymbol{X}$, and $\boldsymbol{S}$ and $\boldsymbol{\Omega}$ are the dimensionless strain-rate and rotation-rate tensors, respectively.

Galilean and rotational invariance during learning. Following Ling *et al.* (2016*b*), in our architecture, the neural network models the nonlinear relationship between the scalar invariants λ_n^* and the corresponding coefficient functions $G^{*(k)}$. These coefficients are subsequently combined with the non-dimensional integrity bases $T_{ij}^{*(k)}$ to reconstruct the anisotropic Reynolds stress tensor b_{ij}^*.

2.3. *Geometric effects through stream functions and velocity potentials*

While the original TBNN framework provides a robust representation of turbulence anisotropy based on local invariant features, it relies solely on local quantities such as the strain-rate and rotation-rate tensors. In canonical flows like a channel flow, the turbulence field exhibits strong spatial inhomogeneity, particularly in the wall-normal direction due to the presence of solid boundaries. Key turbulence quantities such as the turbulent kinetic energy, Reynolds stresses and mean velocity profile vary significantly between the near-wall region and the free stream, with the near-wall region being dominated by intense shear and viscous effects. The resulting inhomogeneities violate the homogeneity assumptions underlying the original invariant theorem, limiting its applicability in complex, non-uniform geometries.

To address these limitations, previous studies have developed alternative learning frameworks based on physically interpretable, non-dimensional flow features (Ling & Templeton 2015; Wang *et al.* 2017). Instead of using tensor invariants, these methods rely on scalar quantities that encode turbulence characteristics such as the Q criterion and turbulence intensity. Notably, several features, such as the pressure gradient along streamlines, streamline curvature, and the non-orthogonality between velocity and its gradient, implicitly reflect aspects of flow topology and boundary influence. Motivated by this, we enhance the TBNN framework by introducing the stream function and velocity potential as supplementary scalar inputs. These quantities naturally encode spatial information and geometric constraints, thereby compensating for the lack of non-local representation in the standard invariant-based formulation. Moreover, rather than those involving the solution from the baseline RANS model in most previous studies, the stream function and velocity potential can be efficiently computed prior to solving the flow field without the need of the baseline model. Both scalar quantities are invariant under Galilean transformations and rotations, so they can be seamlessly integrated into the TBNN's input without violating the network's invariance properties.

With the inclusion of the two geometry-informed features, the extended TBNN model is defined as

$$\begin{aligned} b_{ij}^* &= \mathcal{F}_{b_{ij}^*}\left(S_{ij}^*,\ \Omega_{ij}^*,\ \psi^*,\ \phi^*\right) \\ &= \sum_{k=1}^{10} G^{(k)}\left(\lambda_n^* \mid n \in \{1 \sim 5\},\ \psi^*,\ \phi^*\right)\ T_{ij}^{*(k)}, \end{aligned} \tag{2.14}$$

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

where ψ^* and ϕ^* are values of dimensionless stream function and velocity potential normalised by the characteristic scales, $\mathcal{U}$ and $\mathcal{L}$.

In this study, the TBNN model is trained using mean flow data based on (2.14), obtained from LES. The b_{ij}^* obtained from LES is treated as ground truth. The loss function $L_{b_{ij}}$ used during training is defined as

$$L_{b_{ij}^*} = \frac{1}{N_t} \sum_{k=1}^{N_t} \left| b_{ij,gt}^* - b_{ij}^* \right|, \tag{2.15}$$

where $|\cdot|$ denotes the absolute-value norm, $b_{ij,gt}^*$ is the ground-truth anisotropic stress tensor obtained from LES, b_{ij}^* is the corresponding TBNN prediction and N_t is the total number of training points. Details of the network architecture and training setup are presented in Appendix A.

2.4. *Optimal eddy viscosity*

Accurately incorporating the Reynolds stress tensor into the RANS equations remains a key challenge in turbulence modelling. As demonstrated by Thompson *et al.* (2016), Wu *et al.* (2019), directly embedding Reynolds stresses obtained from DNS into the RANS equations can lead to an ill-conditioned system, ultimately preventing the solver from converging. To address this issue, we adopt the linear eddy-viscosity hypothesis to transform the predicted dimensionless anisotropic tensor into an equivalent eddy viscosity, thereby improving the robustness of the solver. The closure is expressed as

$$b_{ij}^* \approx -2\, \nu_t^*\, S_{ij}^*, \tag{2.16}$$

where $\nu_t^* = \nu_t/(\mathcal{U}\mathcal{L})$ is the dimensionless eddy viscosity.

To obtain the optimal ν_t^* that minimises the approximation error in (2.16) for each directional component in tensor b_{ij}^*, we follow the approach of McConkey, Yee & Lien (2022) and formulate a non-negative least-squares optimisation problem,

$$\nu_t^{*\dagger} = \arg\min_{\nu_t \geq 0} \left\| b_{ij}^* - \left(-2\nu_t^* S_{ij}^*\right) \right\|, \tag{2.17}$$

where $\|\cdot\|$ denotes the Euclidean norm. Differentiating this objective function and enforcing the non-negativity constraint yields a closed-form solution,

$$\nu_{t,opt}^{*\dagger} = \max\left(0,\ -\frac{1}{2}\frac{b_{ij}^* S_{ij}^*}{S_{ij}^* S_{ij}^*}\right), \tag{2.18}$$

with $A_{ij}B_{ij} = \boldsymbol{A} : \boldsymbol{B}$. This formulation ensures that the tensor-wise optimal eddy viscosity remains non-negative.

The objective of this study is to develop a self-contained turbulence model capable of producing accurate flow fields within the RANS framework. During the iterative process used to compute the steady-state solution, both b_{ij}^* and S_{ij}^* are updated at each iteration step. It is important to note that even minor variations in any component of S_{ij}^* or b_{ij}^* during iteration can result in significant changes in $\nu_{t,opt}^{*\dagger}$, as defined in (2.18). This sensitivity is especially pronounced for values not well represented in the training data. Such an ill-posed situation may hinder convergence or cause the iteration to settle on an incorrect flow field. To ensure the well-posedness of $\nu_{t,opt}^{*\dagger}$ throughout the iteration process, it is essential to construct a functional relationship that maps any given S_{ij}^* and b_{ij}^* to an appropriate value of $\nu_{t,opt}^{*\dagger}$. Leveraging the available training data, this functional relationship can be

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

developed through regression, i.e.

$$\nu_{t,reg}^{*\dagger} = \mathcal{F}\left(b_{ij}^{*},\ S_{ij}^{*}\right), \tag{2.19}$$

where $\mathcal{F}$ denotes the regression operator that maps the tensor pair (b_{ij}^{*}, S_{ij}^{*}) to the optimal eddy viscosity $\nu_{t,reg}^{*\dagger}$. It is worth noting that the functional relationship defined by (2.19) can be highly nonlinear, which may lead to overfitting and hinder the convergence of the iterations in the *a posteriori* step. We have explored several alternative machine learning methods for learning the functional relationship between eddy viscosity and the local velocity gradient, including Gaussian process regression, polynomial fitting, and gradient boosting (e.g. XGBoost). However, these approaches did not yield stable or convergent results when embedded within the steady RANS solver. In particular, polynomial models exhibited poor generalisation and produced unstable behaviour outside the training domain, while Gaussian processes introduced excessive curvature in the predicted eddy viscosity, leading to divergence or non-physical solutions. Boosted decision trees (e.g. XGBoost) produced highly sensitive predictions due to their sequential fitting nature, which significantly hindered convergence in the iterative CFD solver.

In contrast, the ensemble averaging inherent in the Random Forest algorithm results in smooth, bounded and relatively insensitive predictions, which are crucial when the model is tightly coupled with the iterative solver. Moreover, Random Forest predictions tend to interpolate within the bounds of the training data and are less prone to overfitting to noise or outliers, which contributes to the overall stability and robustness of the RANS solution process. These characteristics make Random Forests especially suitable for use in ML-based turbulence closure models.

It should also be noted that using the linear form as our final Reynolds stress model could lead to the underexploitation of TBNN, which consists of various nonlinear terms. However, we found it challenging to ensure the convergence of iteration in the *a posteriori* stage when adding nonlinear terms, an issue that has also been addressed by Wu *et al.* (2019). Therefore, we compromise by employing a linear model (2.16) for the *a posteriori* stage, while retaining a nonlinear regression relationship for b_{ij}. While the present framework aligns with the traditional RANS approach, it may fail to accurately predict flows with significant anisotropic features, such as duct flow. In previous studies, the anisotropic nature was typically addressed by introducing discrepancies into the traditional RANS framework to account for the nonlinear contributions to the Reynolds stress (Wu *et al.* 2018; Amarloo, Forooghi & Abkar 2022; Cai *et al.* 2024; Macedo *et al.* 2024). While this may pose challenges for ensuring convergence of iteration in the *a posteriori* stage, an alternative could be the incorporation of the anisotropic eddy viscosity model as a replacement for (2.17), which presents potential for future investigation.

To obtain training targets for this regression model, we transform the TBNN-predicted anisotropic tensor into the corresponding optimal eddy viscosity using the expression given in (2.18). The computed eddy viscosity values are treated as ground truth in the training process. The regression model is then optimised by minimising the error between the predicted viscosities and the target values. The loss function is defined as

$$L_{\nu_t^{\dagger}}^{*} = \frac{1}{N_t}\sum_{k=1}^{N_t} \left\| \nu_{t,opt}^{*\dagger} - \nu_{t,reg}^{*\dagger} \right\|. \tag{2.20}$$

In summary, the incorporation of an additional ML model to construct (2.19) is a critical component in the development of the present ML-based RANS framework. This model constrains any variations in (b_{ij}^{*}, S_{ij}^{*}) during iteration to a fixed functional mapping for

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

computing the eddy viscosity. By serving a role analogous to traditional closure models such as k–ϵ and k–ω, it ensures stable convergence throughout the RANS iteration process. For simplicity, all physical quantities in the remainder of the paper are presented in dimensionless form, omitting the superscript *.

3. TBNN integrated RANS modelling

The turbulence modelling framework proposed in this study comprises two stages: an *a priori* training phase and an *a posteriori* deployment phase. In the *a priori* phase, we train the TBNN model using mean flow fields from LES. The input features include five tensor invariants, two additional geometry-informed features and ten tensor bases. Based on these inputs, the model learns a nonlinear mapping to accurately predict the anisotropic Reynolds stress tensor. Next, we employ the Random Forest regressor to map the anisotropic Reynolds stress and strain-rate tensors to the optimal eddy viscosity. In the subsequent *a posteriori* phase, the trained TBNN and regression models are tightly integrated into a steady RANS solver. The optimal eddy viscosity predicted by these models is directly embedded into the momentum equations,

$$U_m \frac{\partial U_j}{\partial x_m} = -\frac{\partial P}{\partial x_j} + \frac{\partial}{\partial x_m}\left[\frac{1}{Re}\frac{\partial U_j}{\partial x_m} - \nu_{t,reg}^{\dagger} S_{mj}\right], \tag{3.1}$$

where all variables are non-dimensionalised using the reference flow velocity $\mathcal{U}$ and length scale $\mathcal{L}$. Through iterative computation, the flow field converges to a stable solution. This represents a tightly coupled strategy that enables a data-driven turbulence closure without reliance on traditional turbulence models. The overall implementation of the proposed framework is summarised in figure 1 and proceeds through the following steps.

(i) Feature extraction. Extract mean flow fields from LES data, and compute the local tensor invariants as well as the stream function and velocity potential to form the complete input feature vector (*a priori* stage, see figure 1).
(ii) Neural network training. Train the TBNN on the extracted feature vectors to establish a nonlinear mapping to b_{ij} (*a priori* stage).
(iii) Eddy viscosity estimation. Using the predicted b_{ij} and S_{ij}, solve for the optimal eddy viscosity $\nu_{t,reg}^{\dagger}$ via a least-squares approach, and train a Random Forest model to emulate this process (*a priori* stage).
(iv) Initialisation. Initialise the mean velocity and eddy viscosity fields for the RANS solver, then extract the features, and apply the trained TBNN and regression model to compute $\nu_{t,reg}^{\dagger}$ (*a posteriori* stage).
(v) Flow field iteration. Input the computed $\nu_{t,reg}^{\dagger}$ into the RANS solver to update the mean flow field (*a posteriori* stage).
(vi) Convergence check. Compare the velocity fields between iteration n and $n+1$. If the change is below a specified threshold, the solution is deemed converged; otherwise, repeat steps 4–6 until convergence (figure 1, *a posteriori* stage).

This tightly coupled framework, which integrates data-driven turbulence modelling with RANS, enables a zero-equation formulation that eliminates the need for traditional transport equations for second moment quantities. To the best of the authors' knowledge, this study presents the first implementation of such a hybrid turbulence closure strategy, combining explicit algebraic modelling with tensor-based machine learning. The resulting

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

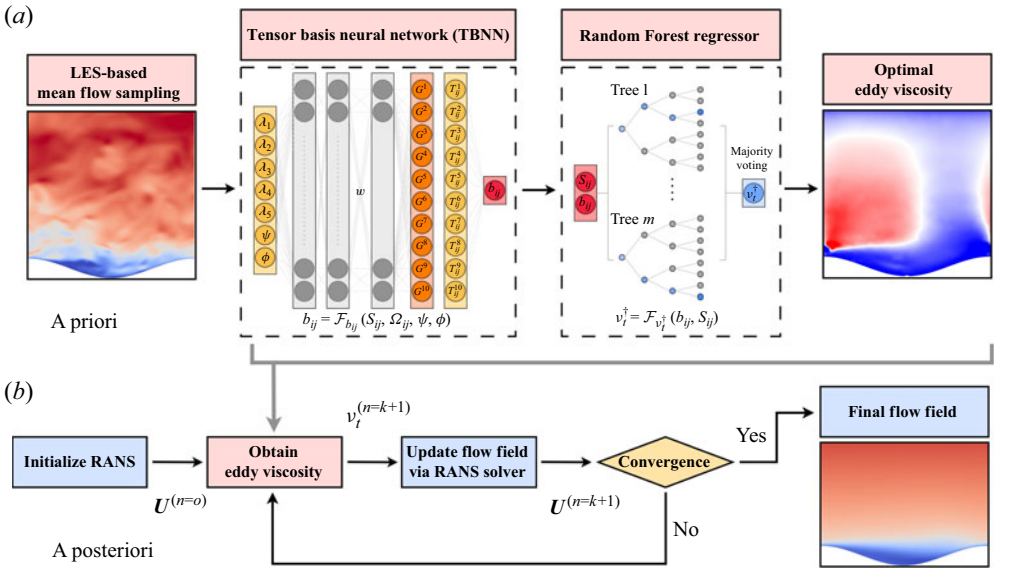


Figure 1. Flowchart of the tightly coupled TBNN framework proposed in this study. The procedure consists of an *a priori* phase for feature extraction, model training and eddy viscosity estimation, followed by an *a posteriori* phase for RANS simulation and convergence evaluation.

formulation is compact, robust and broadly applicable to non-homogeneous turbulent flows.

4. Data and solver

The proposed TBNN-based framework was applied to model flows in three types of configurations featuring distinct bottom geometries, which are: (i) a four-periodic wavy bottom (WB); (ii) a smoothed step (SS); and (iii) a backward-facing step (BS). To train the TBNN model in the *a priori* phase, a comprehensive dataset was generated via LES. In each case, the largest flow depth, H, is used as the characteristic length scale, $\mathcal{L}$, and the free stream velocity at the section of the maximum flow depth is used as the characteristic velocity scale, $\mathcal{U}$, to define the Reynolds number, Re. Either the free stream velocity or bottom geometry is varied in each configuration. The latter is made by changing the waveheight in the wavy-bottom case or the step height in the smoothed step and backward-facing step cases. These features, corresponding to either wave height or step height depending on the case, are collectively denoted as h. Because the flows are driven by pressure gradient, $\mathrm{d}P_d/\mathrm{d}x$, each case can also be characterised by the shear velocity $u_\tau = \sqrt{H\,\mathrm{d}p_d/\mathrm{d}x}$. Based on this, a frictional Reynolds number is defined as $Re_\tau = u_\tau H/\nu$ for each case. In each case, the mean values are obtained by performing both spanwise and temporal averaging of the flow field after the quasi-steady state has been reached. Spanwise averaging is carried out over all computational slices in the spanwise direction, while temporal averaging is performed using 160 samples collected over a physical time of 60 s, which corresponds to $O(10)$ units of characteristic time ($\mathcal{L}/\mathcal{U}$), varying between different cases due to differences in set-up. We found this sample size to be sufficient for obtaining statistically converged mean values of the data.

A summary of case name and parameters is provided in table 3. Additional details regarding numerical set-up can be found in Appendix B. For comparison purposes,

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

Configuration	Case name	Re	Re_τ	h
Wavy bottom	WBRe200h12	2 670	200	0.12
	WBRe250h09	4 090	250	0.09
	WBRe250h12	3 490	250	0.12
	WBRe250h15	3 350	250	0.15
	WBRe250h18	3 190	250	0.18
	WBRe310h12	4 110	310	0.12
	WBRe500h12	6 530	500	0.12
	WBRe710h12	9 140	710	0.12
	WBRe1000h12	12 290	1 000	0.12
Smoothed step	SSRe310h11	5 280	310	0.11
	SSRe500h11	8 530	500	0.11
	SSRe710h08	13 600	710	0.08
	SSRe710h09	13 360	710	0.09
	SSRe710h11	12 390	710	0.11
	SSRe710h12	11 990	710	0.12
	SSRe710h13	11 660	710	0.13
	SSRe1000h11	17 950	1 000	0.11
Backward-facing step	BSRe1160h33	10 730	1 160	0.33
	BSRe1270h33	11 540	1 270	0.33
	BSRe1330h17	16 960	1 330	0.17
	BSRe1330h25	14 050	1 330	0.25
	BSRe1330h33	12 280	1 330	0.33
	BSRe1380h33	12 540	1 380	0.33

Table 3. Summary of case names, values of the Reynolds number (Re), frictional Reynolds number (Re_τ) and geometric feature (h) for all the LES simulations cases. Case names use the prefixes 'WB', 'SS' and 'BS' for the four-periodic wavy bottom, smoothed step and backward-facing step cases, respectively.

we also simulate flow in the same configuration for the predicted cases using the standard RANS model with the k–ϵ and k–ω closure models. These simulations were performed using the flow solver available in ANSYS Fluent, with all turbulence-related parameters set to their default values.

The stream function ψ is obtained by solving the Laplace equation, $\nabla^2\psi = 0$, subject to the following boundary conditions: $\psi = 0$ along the bottom wall and $\psi = 1$ along the top boundary, with no horizontal variation at both the inlet and outlet boundaries. The velocity potential ϕ is similarly determined by solving the Laplace equation, $\nabla^2\phi = 0$, with zero normal gradient imposed on the top and bottom boundaries. In addition, a Dirichlet condition $\phi = 0$ is applied at the left boundary, while at the right boundary, ϕ is set equal to the non-dimensional domain length (i.e. $\mathrm{d}\phi/\mathrm{d}x = 1$).

The *a posteriori* simulations are conducted using FEniCSx, an open-source finite-element method framework for solving partial differential equations (Logg, Mardal & Wells 2012). FEniCSx is selected as the *a posteriori* flow solver primarily due to its flexibility, which facilitates the modification of turbulence closure models and the integration of our TBNN framework. Nevertheless, we believe that the current TBNN model can be incorporated into other open-source CFD solvers in a straightforward manner.

To obtain the solution of the RANS equations, the Newton–Raphson method is employed. In the following simulations, we use approximately 27 000, 60 000 and 81 000 elements for the wavy bottom, smoothed step and backward-facing step cases, respectively.

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

Training type	Training cases	Testing case
Type I (wavy bottom): training data varying in both Re_τ and h	WBRe250h09, WBRe200h120, WBRe310h12, WBRe250h12, WBRe710h12, WBRe1000h12, WBRe250h15, WBRe250h18.	WBRe500h12
Type II (wavy bottom): training data varying in Re_τ	WBRe200h12, WBRe310h12, WBRe500h12, WBRe710h12. WBRe1000h12	WBRe250h12
Type III (wavy bottom): training data varying in h	WBRe250h09, WBRe250h15, WBRe250h18	WBRe250h12
Type I (smoothed step):	SSRe310h11, SSRe500h11, SSRe710h08, SSRe710h09, SSRe710h12, SSRe710h13, SSRe1000h11	SSRe710h11
Type II (smoothed step):	SSRe310h11, SSRe500h11, SSRe1000h11	SSRe710h11
Type III (smoothed step):	SSRe710h08, SSRe710h09, SSRe710h12, SSRe710h13	SSRe710h11
Type I (backward-facing step)	BSRe1160h33, BSRe1330h33, BSRe1330h25, BSRe1330h17, BSRe1380h33.	BSRe1270h33
Type II (backward-facing step)	BSRe1160h33, BSRe1330h33, BSRe1380h33	BSRe1270h33
Type III (backward-facing step)	BSRe1330h17, BSRe1330h33,	BSRe1330h25

Table 4. Summary of training and testing datasets corresponding to configuration cases and training types. Detailed information for each case is provided in table 3.

The mesh is refined near both the top and bottom boundaries, with the finest and coarsest mesh sizes being approximately $0.01H$ and $0.04H$, respectively, across all configurations.

5. Results and discussion

This section presents both *a priori* and *a posteriori* evaluations across three representative geometrical configurations. As summarised in table 3, the training dataset includes variations in both the friction Reynolds number (Re_τ) and the geometric parameter h. Based on these variations, we consider three types of training data. The first type includes combined variations in both Re_τ and h (type I), the second type includes variations in Re_τ only (type II), and the third includes variations in h only (type III). Details of these variations are provided in table 4.

Each configuration is evaluated using two TBNN variants. The first variant, TBNN-i5, uses five local invariant-based features, following the original TBNN framework (Ling *et al.* 2016*a*,*b*). The second variant, TBNN-i7, incorporates both the velocity potential and stream function values. Detailed comparisons through *a priori* and *a posteriori* evaluations are presented for the first type of training data, which include combined variations in Re_τ and h. To keep the presentation concise, only *a posteriori* comparisons are provided for the second and third training cases. These comparisons

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

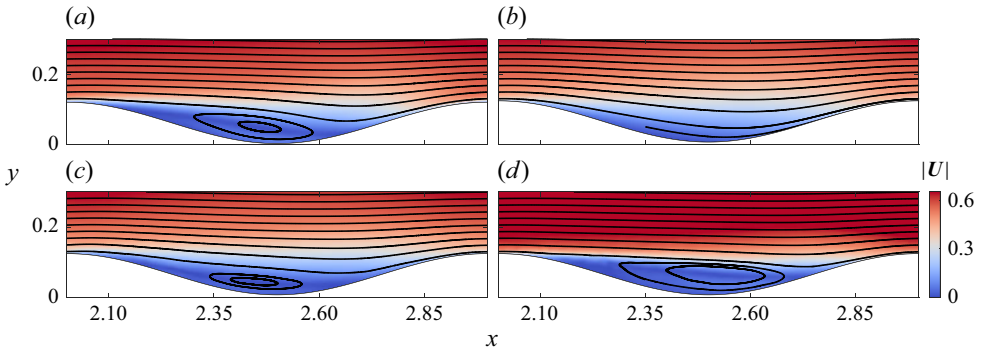


Figure 2. Streamlines overlaid on velocity magnitude contours in the near-wall region for wave-bottom channel flow, shown for (*a*) LES data, and RANS results using (*b*) the k–ϵ model, (*c*) the TBNN-i7 model and (*d*) the TBNN-i5 model.

involve the TBNN-i7 model, conventional RANS turbulence models and LES data. As previously mentioned, for the evaluation of *a posteriori* in each case, we compared the results with those obtained using traditional turbulence closure models, k–ϵ and k–ω. However, for the sake of conciseness, we present only the model that best matches our LES mean flow results for the flow field comparison.

5.1. *Channel flow with a four-period wavy bottom*

Figure 2 presents the mean flow patterns predicted by the four different model approaches, represented by the mean velocity magnitudes and streamlines, at the trough of the wavy bed. Figure 2(*a*) shows that the flow separates at the lee side of the wave, forming a recirculation zone, and reattaches at the windward side of the following wave crest. This type of bottom geometry has been frequently used to mimic periodically varying bed geometries in various geophysical flow problems. Accurately capturing the separation zone using traditional closure models is challenging but essential for predicting the associated form drag. This geometry was also used as a demonstration case in the pioneering TBNN study by Ling *et al.* (2016*b*). In what follows, we first focus on predicting the anisotropic Reynolds stress tensor and eddy viscosity in the *a priori* stage, and return to the predicted mean velocity field in the *a posteriori* stage.

5.1.1. *A priori evaluation with training data varying in both Re_τ and h*

Figure 3 compares the predicted b_{12} from the present TBNN model with those obtained from the LES mean flow field for the wavy-bottom case. Both TBNN-i7 and TBNN-i5 are able to predict the anisotropic tensor in the free stream region (i.e. $y > 0.3$). However, only TBNN-i7 successfully captures its negative distribution observed in LES results (figure 3*b*), which corresponds to the shear layer region above the recirculation zone. In contrast, TBNN-i5 predicts a significantly smaller and fragmented negative region (figure 3*c*), indicating that it fails to reproduce the directional structure of the Reynolds stress in this zone. Although not shown here, similar trends are observed for the other anisotropic stress components, b_{11} and b_{22}.

The comparison of eddy viscosity fields further highlights the differences in predictive capabilities among models. Figure 4 presents the predicted eddy viscosities, $\nu_{t,opt}^{\dagger}$ and $\nu_{t,reg}^{\dagger}$. Here, $\nu_{t,reg}^{\dagger}$ is obtained through regression using the Random Forest model, (2.19). In LES, $\nu_{t,opt}^{\dagger}$ is computed using (2.18) directly from the LES mean flow field of the

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

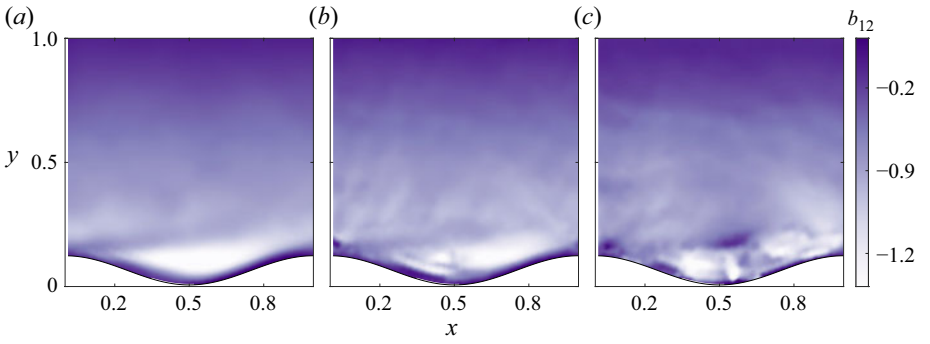


Figure 3. Distributions of the anisotropic Reynolds stress tensor b_{12} for the wavy-bottom case from (*a*) the original LES data, (*b*) TBNN-i7 prediction and (*c*) TBNN-i5 prediction.

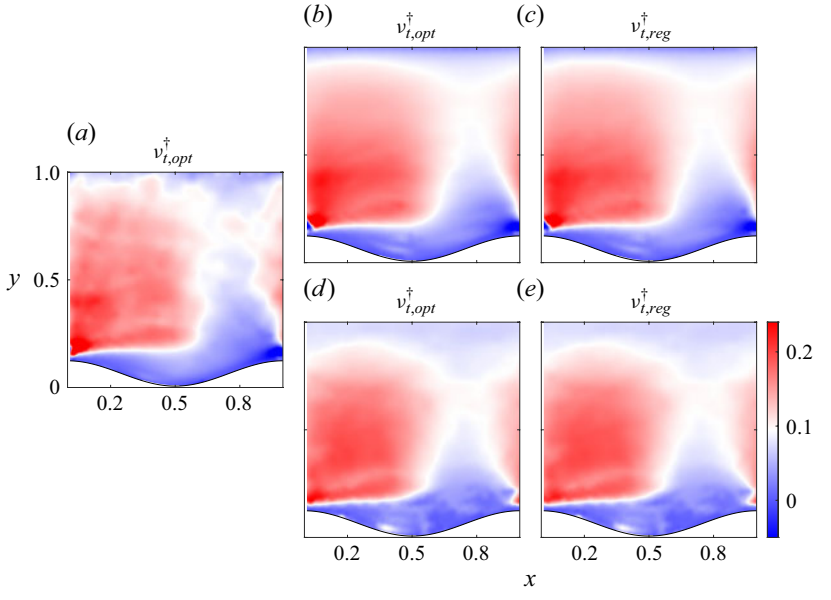


Figure 4. Comparison of eddy viscosity distributions in the case with wavy-bottom channel flow obtained from (*a*) LES, (*b*,*c*) TBNN-i7 and (*d*,*e*) TBNN-i5. Panels (*c*) and (*e*) show the eddy viscosity fields obtained from the regression model (i.e. Random Forest).

predictive case, while in TBNN, $\nu^{\dagger}_{t,opt}$ is also calculated using (2.18) but based on data predicted from training cases in table 4. From the LES results, as shown in figure 4(*a*), it is observed that the eddy viscosity significantly decreases within the recirculation zone downstream of each crest, while a sharp increase appears in the free shear layer above, forming a high eddy viscosity region.

The optimised eddy viscosity fields reconstructed from the TBNN-predicted Reynolds stress tensors show that TBNN-i7 (figure 4*b*) successfully reproduces the characteristic LES structure. It is worthwhile to mention that although our sampling size ensures the statistical convergence of the mean flow field in LES, the second-moment quantities, such as b_{12} in figure 3(*a*), still exhibit slight irregularities. Such irregularities are even more apparent in ν_t, as shown in figure 4(*a*), as it is obtained by dividing the mean velocity gradient. The same features can also be found in the other two examples. While such

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

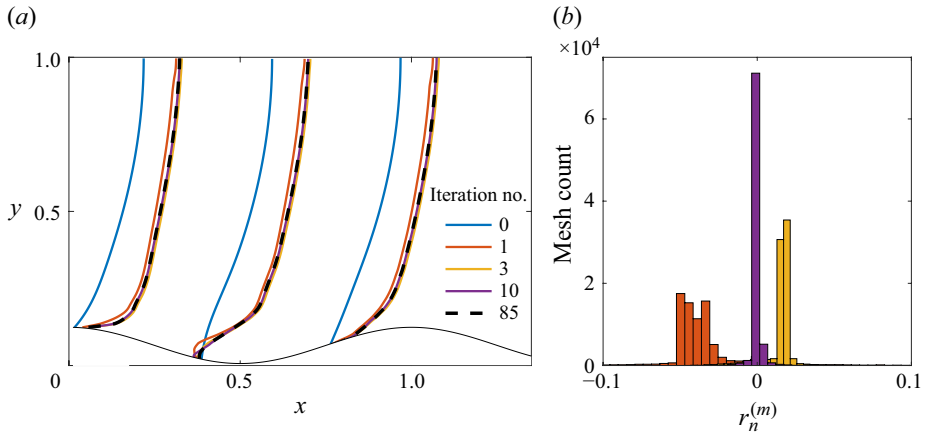


Figure 5. Convergence behaviour of the iterative solution process used to compute velocity profiles for the wavy-bottom channel flow, shown over a single geometric period. (*a*) Evolution of the velocity field across selected iterations. (*b*) Histograms of the residuals defined in (5.1) at the 1st, 3rd and 10th iteration steps, with colours corresponding to those in panel (*a*).

issues could be improved by extending the sampling size (i.e. a longer period of sampling time), as shown in figure 4(*b*), these small-scale irregularities can be effectively filtered out by the present TBNN.

In contrast to TBNN-i7, as shown in figure 4(*d*), TBNN-i5 introduces spurious extrema within the recirculation zone. These results demonstrate that the inclusion of geometric features in TBNN-i7 enhances its ability to capture eddy viscosity variations in inhomogeneous regions, compared with the limited expressiveness of TBNN-i5. In addition, as shown in figures 4(*c*) and 4(*e*), the eddy viscosity fields predicted by the regression model show strong consistency with the TBNN results (figures 4*b* and 4*d*, respectively), confirming the effectiveness of our regression model.

5.1.2. *A posteriori evaluation with training data varying in both Re_τ and h*

Since ensuring solution convergence during iteration is a key aspect of the flow solver, we examine convergence in figure 5. This figure presents the calculated velocity profiles at representative iteration steps towards the steady-state solution, along with the corresponding histograms of residuals. The residual at the mth iteration step during the iterative process is defined as

$$r_n^{(m)} = \frac{1}{\mathcal{U}}\left(\boldsymbol{U}_n^{(m)} - \boldsymbol{U}_{n,ref}\right), \tag{5.1}$$

where n is the index of the grid point and the subscript ref denotes the reference value. In figure 5, we use the iterative solution at the 85th step as the reference. As shown in figure 5(*a*), the velocity profiles converge after just a few iteration steps. Moreover, figure 5(*b*) illustrates that the residuals evolve toward a normal distribution centred near zero as the iteration progresses. In this study, the iteration process is terminated when the following convergence criterion is satisfied:

$$\frac{\left\|\boldsymbol{U}^{(m)} - \boldsymbol{U}^{(m-1)}\right\|_2}{\left\|\boldsymbol{U}^{(m-1)}\right\|_2} < 10^{-10}, \tag{5.2}$$

where the superscript (k) indicates the iteration step.

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

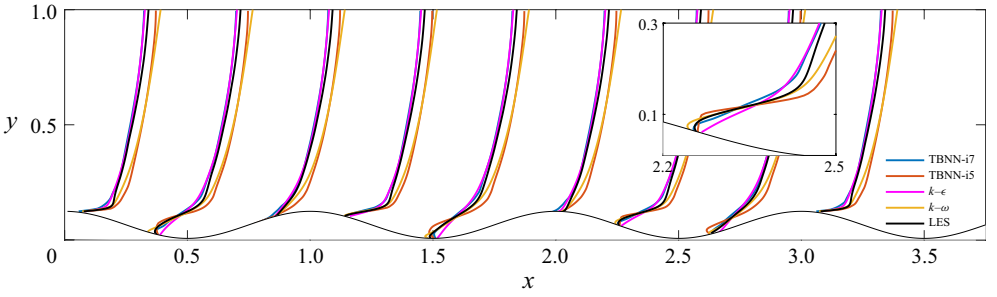


Figure 6. Comparison of velocity profiles from different simulation approaches in the case of wavy-bottom channel flow.

Figure 6 presents the predicted velocity profiles obtained from LES, TBNN, $k-\epsilon$ and $k-\omega$ closure models at various streamwise locations. As shown in figure 6, all models exhibit steep velocity gradients near the wall, indicating a consistent ability to capture viscosity-dominated regions. However, both TBNN-i5 and $k-\omega$ tend to overpredict the velocity profiles, whereas TBNN-i7 and $k-\epsilon$ more closely match the LES results. The discrepancies observed in the traditional closures (i.e. k–ϵ and k–ω) are mainly attributed to the misprediction of near-wall turbulent viscosity. While we adopt the default values in ANSYS Fluent for the empirical coefficients in the transport equations of the second-moment quantities (i.e. k, ϵ and ω), these discrepancies could potentially be mitigated by tuning those empirical values. Such considerations, however, fall outside the scope of the present study. Here and in the following numerical tests, we focus on discussing the TBNN-based methods and include the traditional closure models primarily for reference.

The overprediction in TBNN-i5 arises from its inability to yield a b_{12} field with sufficient accuracy in the near-wall region. As shown in figure 3, compared with the original LES data and the prediction by TBNN-i7, TBNN-i5 (figure 3c) fails to accurately capture the decrease in b_{12} towards the wall boundary. This leads to an overall overprediction of the anisotropic Reynolds stress in the near-wall region. Although both TBNN-i7 and TBNN-i5 produce eddy viscosity values similar to the original LES results, as shown in figure 4, the overprediction of b_{12} in the latter case results in an overestimation of the mean velocity profile, as illustrated in figure 6. In the near-wall region behind the crest, where flow separation occurs, TBNN-i7 outperforms the other three models (see the inset), but slightly underperforms k–ϵ on the crest.

In addition, the flow patterns represented by the velocity magnitudes and streamlines within the separation and reattachment regions presented in figure 2 further highlight the differences among model predictions. As shown in figure 2(a), the LES results exhibit a typical shear layer and recirculation structure, accompanied by noticeable streamline deflection. This recirculation is a classical manifestation of flow separation driven by an adverse pressure gradient. In comparison, as shown in figure 2(b), k–ϵ shows the main flow closely attached to the wall, with minimal recirculation behaviour. In figure 2(c), the flow field predicted by TBNN-i7 deviates slightly from the LES results in the location of the shear layer, but successfully captures parts of the reverse flow and the reattachment region. As shown in figure 2(d), TBNN-i5 produces qualitatively similar results to TBNN-i7, but overpredicts both the free stream velocity and the extent of the recirculation zone. These observations indicate that both TBNN models outperform the k–ϵ closure model in capturing separated flow features, with the inclusion of spatial and geometric descriptors in TBNN-i7 significantly improving flow prediction accuracy.

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

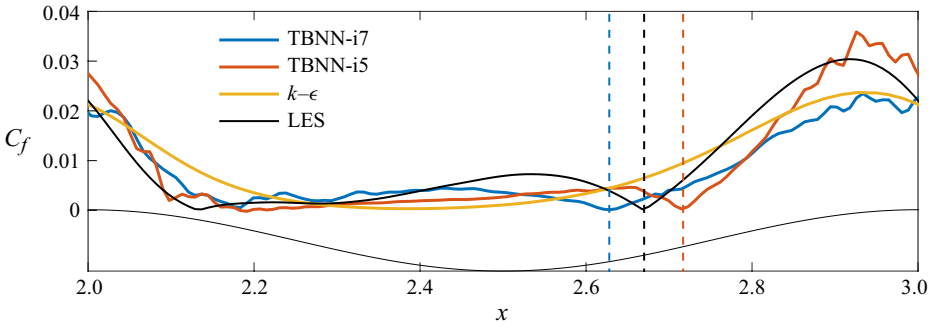


Figure 7. Comparison of the bottom friction coefficient (C_f) from different simulation approaches in the case of wavy-bottom channel flow. The dash lines indicate the locations where the separated flow reattaches in each case.

In addition to the velocity field, we also compare the modelled friction coefficient among different simulation cases. The bottom drag coefficient is obtained by calculating the bottom drag using the non-dimensional quantities,

$$C_f = (\nu + \nu_t)\left.\frac{\mathrm{d}|\boldsymbol{U}|}{\mathrm{d}\xi}\right|_{btm}, \tag{5.3}$$

where $|\boldsymbol{U}| = \sqrt{U_1^2 + U_2^2}$, ξ represents the distance along the direction normal to the bottom geometry and the subscript btm indicates the bottom boundary. In this study, a second-order finite difference scheme that incorporates the no-slip boundary condition and the first two grid points from the bottom is employed to approximate the values of C_f.

Figure 7 presents a comparison of C_f among the different simulation approaches. The results from the TBNN model exhibit non-physical fluctuations. This occurs because the pointwise eddy viscosity obtained from the TBNN can lead to non-smoothness in the predicted eddy viscosity field, which in turn induces fluctuations in the resulting velocity field. These fluctuations become particularly evident when examining the near-bed shear stress. Applying an appropriate smoothing procedure to the TBNN-generated eddy viscosity, while preserving the global extrema, could help mitigate this issue in future work. Compared with the LES result, the TBNN-i7 case shows slight overprediction of drag at the lee side, but underprediction at the trough and windward side, whereas the TBNN-i5 case shows overprediction near the peak. Flow reattachment can be assessed by the x-location where $C_f = 0$, indicated by the dashed lines. The k–ϵ model fails to capture drag reversal at the lee and windward sides due to its inability to predict flow separation, consistent with figure 2(*b*). Compared with LES, predicted reattachment occurs slightly upstream in TBNN-i7 and slightly downstream in TBNN-i5. Despite these discrepancies, in contrast to the k–ϵ model, both TBNN-i5 and TBNN-i7 capture the overall pattern of drag reflection associated with flow detachment and reattachment, although the locations of reflection are slightly shifted relative to the LES result.

5.1.3. *A posteriori evaluation with training data varying in either Re_τ or h*

Figure 8 shows the comparison of predicted mean velocity profiles from LES data and various models for cases trained with data that vary either Re_τ or h in the wavy-bottom configuration. Overall, both TBNN-i7 and k–ω show good agreement with the LES results for both training types. However, in the near-wall region on the lee side of the bottom wave ($x \sim 0.3$), the TBNN-i7 model produces velocity profiles that are closer to the LES

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

	k–ϵ	k–ω	TBNN-i7	TBNN-i5
Type I	4.8 %	13.2 %	4.6 %	14.1 %
Type II	16.7 %	8.9 %	7.3 %	N/A
Type III	16.7 %	8.9 %	3.7 %	N/A

Table 5. Summary of global error values (E), as defined in (5.4), for the k–ϵ, k–ω and TBNN-i7 models in the four-period wavy-bottom channel flow case, using different training data types.

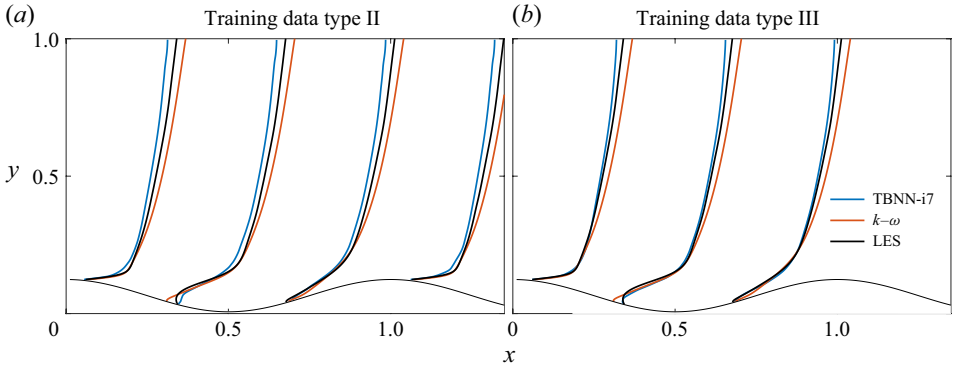


Figure 8. Comparison of velocity profiles from different simulation approaches in the case of wavy-bottom channel flow with training data across either various (*a*) Re_τ (type II) or (*b*) h (type III).

data compared with those from the k–ω model. The good performance of TBNN-i7 is consistent with the results observed in the combined variation training cases, as shown in figure 6, which further confirms the capability of the TBNN-i7 model in handling variations in both free stream flow and geometry.

A more quantitative assessment for different models is made by comparison of global errors among different models. The global error (E) is defined as

$$E = \frac{\sum_{n=1}^{N} \left| \boldsymbol{U}_{LES,n} - \boldsymbol{U}_n \right| \Delta_n}{\sum_{n=1}^{N} \left| \boldsymbol{U}_{LES,n} \right| \Delta_n}, \tag{5.4}$$

where $\boldsymbol{U}_{LES}$ denotes the velocity field obtained from LES, n is the grid index, Δ is the area of the grid cell and the summation is taken over the N grid points used for training the TBNN model. Table 5 summarises the global errors (E) for the results obtained using the TBNN-i7 and traditional turbulence closure methods for the wavy-bottom case. It can be seen that $E < 10$ % for both the TBNN-i7 and k–ϵ models when evaluated on training data that varies in both Re_τ and h (type I). When the training data varies in only one of the two parameters, either Re_τ or h (types II and III, respectively), the present TBNN-i7 model still achieves good agreement with the LES results ($E < 10$ %). In contrast, neither the k–ϵ nor the k–ω model is able to provide accurate predictions, primarily due to significant discrepancies in the near-wall region on the lee side.

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

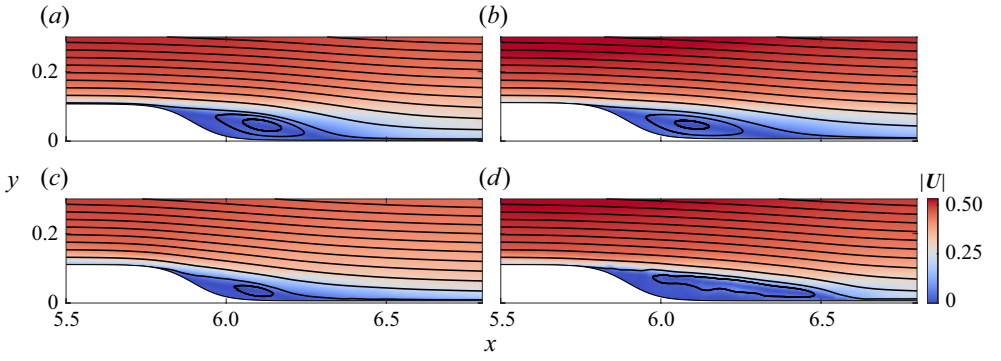


Figure 9. Streamlines overlaid on velocity magnitude contours in the near-wall region for the case of flow over a smoothed step, shown for (*a*) LES data, and RANS results using (*b*) the k–ω model, (*c*) the TBNN-i7 model and (*d*) the TBNN-i5 model.

5.2. *Flow over a smoothed step*

Figure 9 presents the mean flow patterns predicted by the four model approaches, represented by the mean velocity magnitude and streamlines, in a representative case of flow over a smoothed step. Figure 9(*a*) shows that the flow separates at the downslope ramp ($x \sim 6.0$) and reattaches further downstream, forming a recirculation zone. The smoothed ramp configuration serves as a canonical case for studying turbulent separation induced by an adverse pressure gradient in the absence of sharp geometric discontinuities. This makes it highly representative of many real-world engineering flows such as aerofoils, diffusers and internal ducts, where separation is driven primarily by a smooth geometry-induced adverse pressure gradient, rather than sudden expansions. Accurately predicting flow behaviour in such conditions remains a significant challenge for turbulence models, particularly in capturing the onset of separation and reattachment under varying pressure gradients (e.g. Simmons *et al.* 2024).

5.2.1. *A priori evaluation with training data varying in both Re_τ and h*

In the smoothed-step case, the predicted anisotropic Reynolds stress tensor behaves similarly to that observed in the four-period wavy-bottom case, with TBNN-i7 exhibiting higher accuracy than TBNN-i5. As such, the following discussion focuses on comparisons of eddy viscosity. Figure 10 presents the predicted eddy viscosities, $\nu_{t,opt}^{\dagger}$ and $\nu_{t,reg}^{\dagger}$, using the present TBNN model alongside those obtained from the LES mean flow field. The LES results in figure 10(*a*) indicate a low eddy viscosity region near the leading edge of the smoothed step, reflecting a typical acceleration zone. A high eddy viscosity zone appears downstream of the step, corresponding to the free shear layer generated by flow separation. This region is characterised by strong turbulent mixing and momentum exchange. Near the bottom wall, the eddy viscosity decreases to nearly zero. Figure 10(*b*) shows that TBNN-i7 successfully reconstructs these key features, yielding a distribution that closely matches the LES and eliminates small-scale fluctuations present in the original LES data. In contrast, TBNN-i5, as shown in figure 10(*d*), captures the general trend in the free stream region, but underpredicts the eddy viscosity in most of the places except for the upper free stream region behind the step. As shown in figures 10(*c*) and 10(*e*), the eddy viscosity fields obtained from the Random Forest regression model are also highly consistent with the original TBNN-i7 and TBNN-i5 outputs.

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

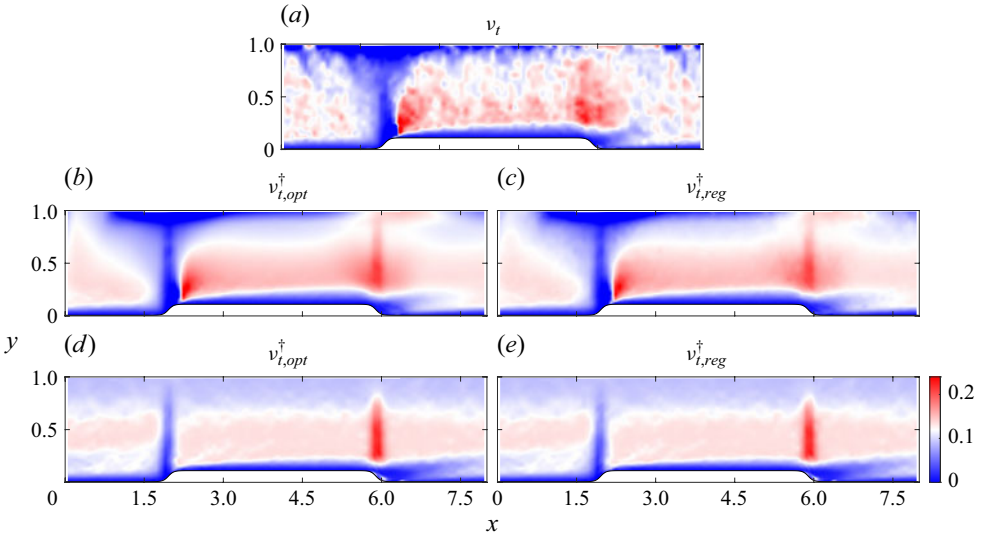


Figure 10. Comparison of eddy viscosity distributions obtained by different methods in the case of flow over a smoothed step: (*a*) original LES result; (*b*) and (*d*) predictions by TBNN-i7 and TBNN-i5, respectively; (*c*) and (*e*) eddy viscosity fields obtained by the Random Forest regression model in cases of TBNN-i7 and TBNN-i5, respectively.

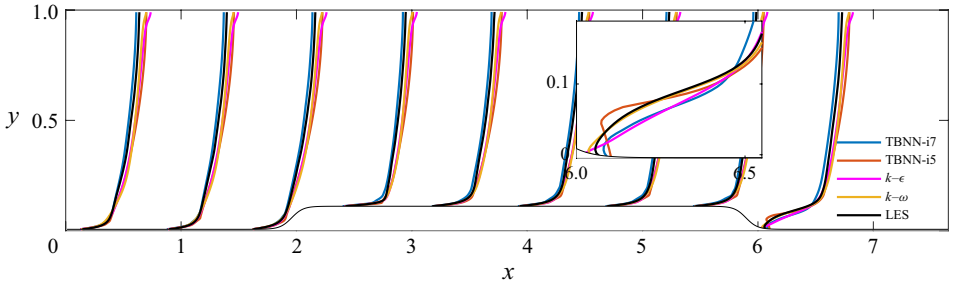


Figure 11. Comparison of velocity profiles from different simulation approaches in the case of flow passing a smoothed step.

5.2.2. *A posteriori evaluation with training data varying in both Re_τ and h*

Figure 11 presents the streamwise velocity profiles predicted by different models. It can be seen that the k–ϵ model significantly underpredicts the velocity magnitude, likely due to its overprediction of the eddy viscosity. This results in the absence of backflow at the downslope ramp, as shown in the inset of figure 11. Another misprediction at the downslope ramp is also observed in the TBNN-i5 profile, which exhibits an unphysical velocity drop towards the bottom (see the inset). This behaviour is consistent with the mispredicted recirculation pattern shown in figure 9(*d*). Overall, the TBNN-i7 prediction shows good agreement with the LES results, accurately capturing the velocity curvature from upstream to the top of the step and maintaining a reasonable near-wall velocity gradient. However, the k–ω model outperforms TBNN-i7 at the downslope ramp, yielding a velocity profile that more closely matches the LES result in that region, though it overestimates the velocity in the free stream core region.

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

	k–ϵ	k–ω	TBNN-i7	TBNN-i5
Type I	7.2 %	6.3 %	4.0 %	9.1 %
Type II	7.2 %	6.3 %	6.0 %	N/A
Type III	7.2 %	6.3 %	2.9 %	N/A

Table 6. Summary of global error values (E), as defined in (5.4), for the k–ϵ, k–ω and TBNN-i7 models in the case of flow over a smoothed step, using different training data types.

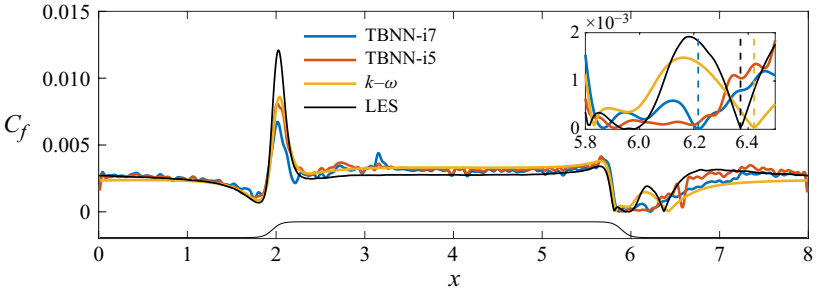


Figure 12. Comparison of the bottom drag coefficient (C_f) from different simulation approaches in the case of smoothed-step channel flow. The dash lines in the inset indicate the locations where the separated flow reattaches in each case.

Further examination of the separated and reattached flow structures shown in figure 9 reveals notable differences between different approaches. The LES result in figure 9(a) shows a stable recirculation bubble, corresponding to the shear layer and a reverse flow structure driven by the adverse pressure gradient. The $k-\omega$ closure method shown in figure 9(b) clearly captures the internal details of the recirculation closely matching the LES data. TBNN-i7, as shown in figure 9(c), reproduces the overall shape of the shear layer and recirculation bubble, although the size of the recirculation is smaller and the velocity gradient appears smoother than in LES. TBNN-i5, as shown in figure 9(d), shows a similar trend to TBNN-i7, but with an irregular recirculation pattern. Overall, while the TBNN-based models exhibit limitations in capturing fine-scale turbulent mixing, they successfully reconstruct the main flow features. For an overall assessment, as shown in table 6 (type I), both TBNN-i7 and $k-\omega$ can achieve $E < 10\,\%$.

Figure 12 presents a comparison of the bottom friction coefficient (C_f) among different modelling approaches for the smoothed-step case. Upstream of the downslope ramp, all approaches capture the overall spatial distribution of C_f, with a slight underprediction on the upslope ramp and an overprediction on the step. As in the previous wavy-bottom example, the TBNN results exhibit noticeable fluctuations. More substantial differences among the modelling approaches appear in the downstream region adjacent to the downslope ramp, as highlighted in the inset of the figure. The increase in bottom friction associated with the circulation zone, clearly observed in the LES results, is well reproduced by the k–ω model, but underpredicted by TBNN-i7 due to its underestimated circulation region (see figure 9c). Consequently, TBNN-i7 predicts an upstream-shifted reattachment point, whereas k–ω aligns more closely with the LES result, as indicated by the dashed lines. TBNN-i5 fails to capture this friction increase because it does not correctly reproduce the circulation zone (see figure 9d).

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

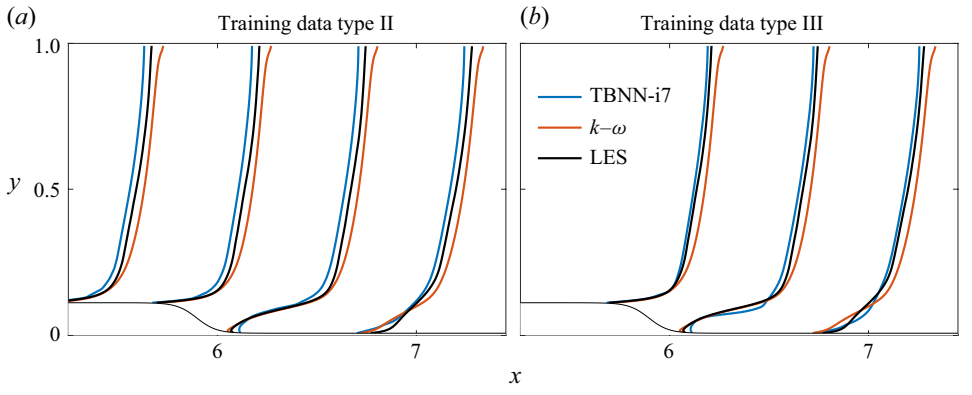


Figure 13. Comparison of velocity profiles from different simulation approaches in the case of flow over a smoothed step with training data across either various (*a*) Re_τ (type II) or (*b*) h (type III).

5.2.3. *A posteriori evaluation with training data varying in either Re_τ or h*

Figure 13 shows the comparison of predicted mean velocity profiles from LES data and various models with training data varying in either Re_τ or h in the smoothed-step configuration. Here, we focus on the downstream half of the domain, containing the downslope ramp. In the case with varying Re_τ, as shown in figure 13(*a*), TBNN slightly underpredicts the external flow region while the k–ω model slightly overpredicts it. In the near-wall region right behind the downslope ramp ($x \approx 6$), the TBNN-i7 tends to slightly underpredict the reverse flow, whereas the k–ω model results agree more closely with the LES data. However, at the further downstream location ($x \approx 7$), the TBNN-i7 results align well with the LES data near the wall, while the k–ω model shows more discrepancies. Despite these slight deviations, both TBNN-i7 and k–ω show good agreement with the LES data when variations in Re_τ are used for training. This is also reflected in table 6, which shows that the global errors in both cases are less than 10 %.

When variations in the geometric feature h are used for training, as shown in figure 13(*b*), the TBNN-i7 again produces results that closely match the LES data, except for a slight underprediction of reverse flow behind the step ($x \approx 6$), similar to the previous case (figure 13*a*). As indicated in table 6, the error in TBNN-i7 reaches 2.9 %.

5.3. *Backward-facing step*

The third case considers the flow field induced by a backward-facing step. Figure 14 presents the mean flow patterns predicted by four different model approaches, represented by the mean velocity magnitude and streamlines, in a representative case of this configuration. Figure 14(*a*) shows that this geometric configuration causes strong flow separation immediately at the corner of the step, forming a large-scale recirculation zone and a distinct shear layer. Compared with the previous two cases, this configuration features the farthest downstream reattachment point and represents the challenging scenario for prediction. Flows over a backward-facing step have been widely studied in the literature (e.g. Geneva & Zabaras 2019; Beetham & Capecelatro 2020; Agrawal & Koutsourelakis 2024).

5.3.1. *A priori evaluation with training data varying in both Re_τ and h*

In the backward-facing step case, TBNN-i7 continues to show good agreement with LES in predicting the anisotropic Reynolds stress tensor, while TBNN-i5 presents noticeable

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

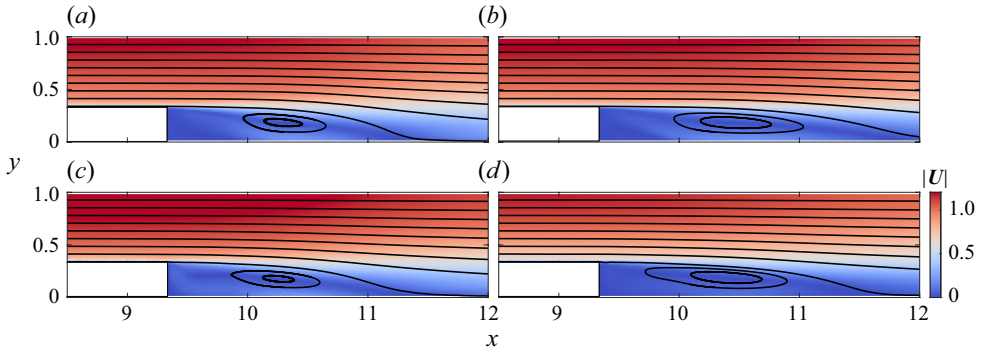


Figure 14. Streamlines overlaid on velocity magnitude contours in the near-wall region for the case of flow over a backward-facing step, shown for (*a*) LES data, and RANS results using (*b*) the k–ω model, (*c*) the TBNN-i7 model and (*d*) the TBNN-i5 model.

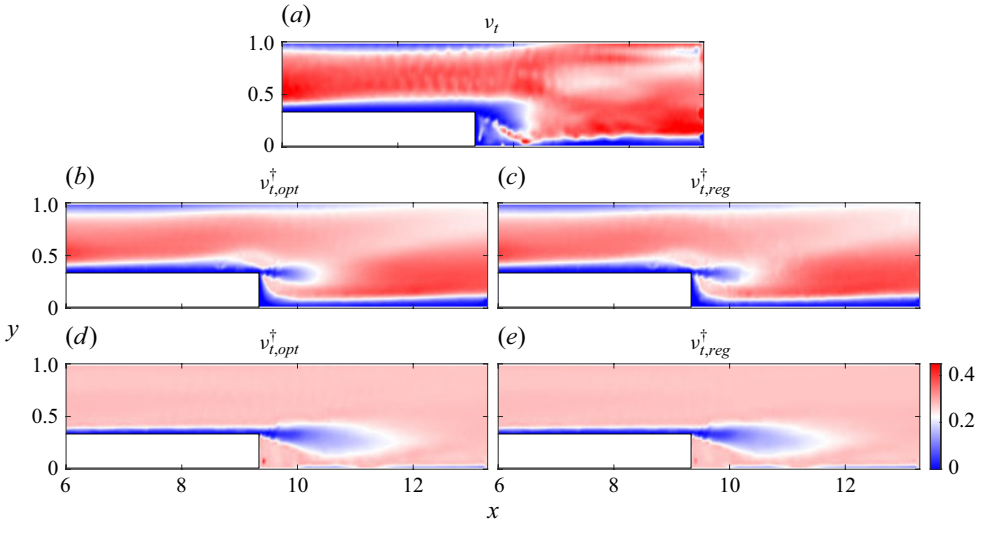


Figure 15. Comparison of eddy viscosity distributions obtained by different methods in the backward-facing step channel flow: (*a*) original LES result; (*b*) and (*d*) predictions by TBNN-i7 and TBNN-i5, respectively; (*c*) and (*e*) eddy viscosity fields obtained by the regression model in cases of TBNN-i7 and TBNN-i5, respectively.

deviations. As these results are consistent with previous cases, the following analysis focuses on the eddy viscosity distribution. Figure 15 presents the eddy viscosities obtained from different model strategies analogous to figure 10. Compared with the LES results (figure 15*a*), TBNN-i7 (figure 15*b*) successfully reproduces the distribution of the eddy viscosity, accurately capturing the low eddy viscosity characteristics behind the step and the local maxima near the reattachment region. In contrast, although TBNN-i5 (figure 15*c*) captures the overall trend around the step, it underpredics the increase in eddy viscosity outside the boundary layer. As shown in figures 15(*d*) and 15(*e*), the eddy viscosity fields obtained by the regression model remain highly consistent with the original TBNN-i7 and TBNN-i5 outputs.

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

	k–ϵ	k–ω	TBNN-i7	TBNN-i5
Type I	14.8 %	4.7 %	5.9 %	8.7 %
Type II	14.8 %	4.7 %	6.5 %	N/A
Type III	14.9 %	6.4 %	8.2 %	N/A

Table 7. Summary of global error values (E), as defined in (5.4), for the k–ϵ, k–ω and TBNN-i7 models in the backward-facing step case, using different training data types.

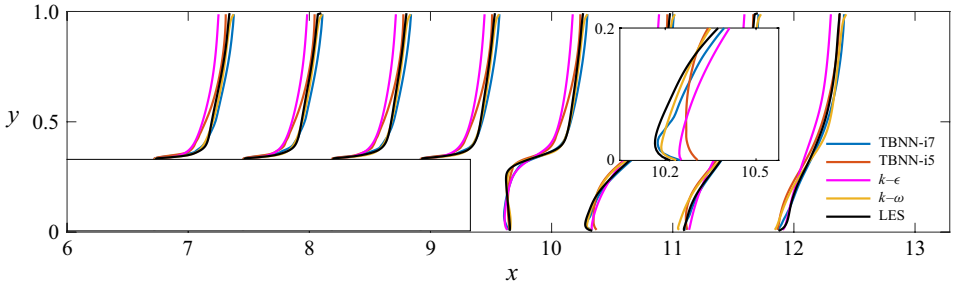


Figure 16. Comparison of velocity profiles from different simulation approaches in the case of flow passing a backward-facing step.

5.3.2. *A posteriori evaluation with training data varying in both Re_τ and h*

Figure 16 presents the streamwise velocity profiles predicted by different models. It shows that both TBNN-i7 and k–ω exhibit robust predictive performance, closely matching the LES results in most places except that the former slightly underpredicts the flow reversal at the downstream separation behind the step (see the inset) and the latter slightly underpredicts the flow at the downstream reattachment point ($x \approx 12$). Due to the misprediction of the eddy viscosity in the vicinity of the wall, neither TBNN-i5 nor k–ϵ accurately predicts the boundary-layer thickness, leading to an underprediction of the velocity above the boundary layer. As summarised in table 7, both TBNN-i7 and $k-\omega$ achieve the global error $E < 10\,\%$ with the latter being slighly lower, while k–ϵ remains $E \sim O(10\,\%)$.

As highlight by streamlines overlying the velocity magnitudes, figure 14 allows the further inspection of the separation and reattachment structures. As shown in figure 14(a), the LES result exhibits a well-defined recirculation bubble and shear layer behind the step, representing a typical separation driven by an adverse pressure gradient. The $k-\omega$ prediction shown in figure 14(b) agrees well with LES in the near-wall velocity distribution, but exhibits slight discrepancies in the location and extent of the recirculation bubble. TBNN-i7 shown in figure 14(c) successfully reconstructs both the recirculation and reattachment structures, with a velocity field that closely resembles LES. TBNN-i5 shown in figure 14(d) also identifies the main separated and recirculation regions, but the depth of the recirculation bubble and the reattachment location are slightly shifted further downstream. Overall, TBNN-i7 provides accurate near-wall velocity predictions, while TBNN-i5 and k–ω maintain consistent reconstruction of the overall structure, but remain less accurate in predicting the reattachment location.

Figure 17 presents a comparison of the bottom friction coefficient (C_f) among different modelling approaches for the backward-facing step flow. On the step's upper flat surface, the predicted drag coefficients generally agree well with the LES results, except that TBNN-i7 slightly underpredicts C_f, and all TBNN-based methods exhibit fluctuations,

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

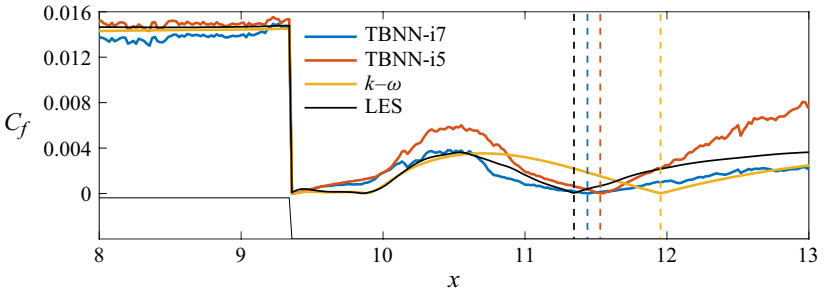


Figure 17. Comparison of the bottom drag coefficient (C_f) from different simulation approaches in the case of backward-facing step flow. The dash lines indicate the locations where the separated flow reattaches in each case.

consistent with the previous two examples. Downstream of the backward-facing step, C_f drops to zero and then gradually increases to a peak due to flow circulation inside the separation bubble. The friction coefficient subsequently decreases to zero at flow reattachment. In general, TBNN-i5 overpredicts C_f downstream of the step, primarily due to overestimation of the eddy viscosity near the wall in this region, as shown in figure 15(*e*). Both $k-\omega$ and TBNN-i7 capture the magnitude of the drag increase induced by circulation. As indicated by the dashed lines, the reattachment points predicted by the TBNN-based methods are shifted slightly downstream relative to the LES result. In the k–ω case, the reattachment point is located even farther downstream ($x \approx 12$), consistent with the delayed reattachment observed in figure 14(*b*). Downstream of reattachment, both k–ω and TBNN-i7 underpredict the friction coefficient, while TBNN-i5 shows overprediction.

5.3.3. *A posteriori evaluation with training data varying in either Re_τ or h*

Figure 18 presents a comparison of the predicted mean velocity profiles from LES data and various models, using training data that vary either in Re_τ or in h, for the backward-facing step configuration. As shown in figure 18(*a*), TBNN-i7 underpredicts the free shear layer immediately downstream of the step at $x \approx 9.5$, while the k–ω model slightly overpredicts the velocity shear further downstream at $x \approx 10$. Despite these minor deviations from the LES data, both TBNN-i7 and the k–ω model show good agreement with the LES results. As indicated in table 7, the global errors for both models are less than 10 %, whereas the k–ϵ model exhibits significantly larger error. When training data vary in h, the discrepancy between TBNN-i7 predictions and LES results becomes more pronounced immediately behind the step, leading to a slightly higher global error compared with the type II training case, as shown in table 3. Nevertheless, both TBNN-i7 and the k–ω model effectively capture the overall trend of the velocity field in the backward-facing step configuration.

5.4. *Combination of different bottom geometries*

Having demonstrated the applicability of the present TBNN framework to flows over three types of bottom geometry, we now assess its potential and limitations across different geometries. For this, all type I training data from the various geometries are used to predict the type I testing cases for each individual geometry, as summarised in table 4.

We consider three training strategies. The first uses the proposed seven flow features (λ_{1-5}, ϕ and ψ) across all geometries, denoted TBNN-C-i7. The second adopts the original TBNN with five features (λ_{1-5}) without ϕ and ψ, denoted TBNN-C-i5, to

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

	TBNN-i7	TBNN-C-i5	TBNN-C-i7	TBNN-C-i7oh
Wavy bottom	4.6 %	9.1 %	20.5 %	5.7 %
Smoothed step	4.0 %	8.1 %	59.6 %	4.3 %
Backward-facing step	5.9 %	16.5	38.9 %	6.7 %

Table 8. Summary of global error values (E), as defined in (5.4), for the cases using combined type I training data across different geometries, along with the values from TBNN-i7 for comparison.

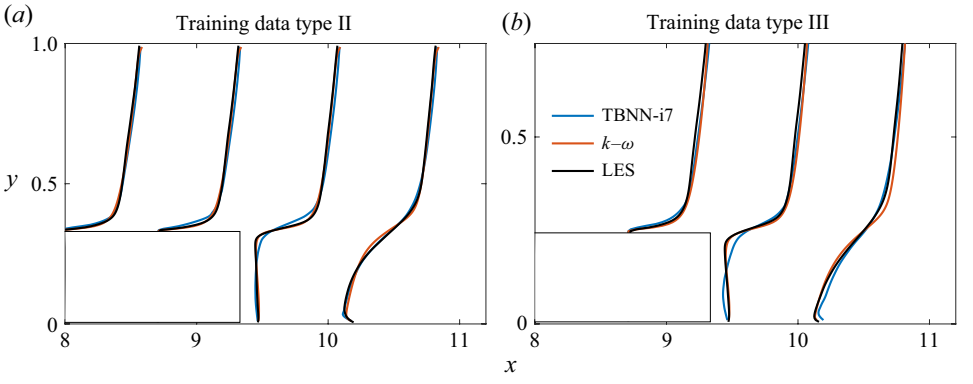


Figure 18. Comparison of velocity profiles from different simulation approaches in the case of backward-facing step with training data across either various (*a*) Re_τ (type II) or (*b*) h (type III).

evaluate the effect of the additional input features on model generalisation. The third strategy uses the seven features, but adds one-hot encoding for geometry: [1 0 0] for the wavy bottom, [0 1 0] for the smoothed step and [0 0 1] for the backward-facing step, referred to as TBNN-C-i7oh. We focus on strategies that can be readily extended from the current framework. While advanced encoding techniques exist, this study provides a preliminary assessment of geometric generalisation without exploring alternative encoding approaches.

Figure 19 presents a comparison of the predicted velocity profiles for three different bottom geometries using the aforementioned training strategies, together with the results from TBNN-i7 and LES. For a more quantitative assessment, table 8 summarises the prediction errors (E) within the near-wall regions shown in figures 2, 9 and 14, along with those from TBNN-i7 for reference. From figure 19 and table 8, it can be seen that TBNN-C-i7 performs poorly, showing large deviations from the LES results in all three geometric cases. In contrast, and different from the results obtained when training with individual geometries, the prediction from TBNN-C-i5 is significantly improved when the stream function and velocity potential are excluded. In this case, the global error is reduced to less than 10 % in the wave-bottom and smoothed-step cases, although it remains large for the backward-facing step (figure 19*c*).

The deterioration in prediction when including ψ and ϕ arises because these two additional features tend to limit the model applicability to the specific geometry used for training. Since their local values are not readily comparable across different geometries, combining datasets that include ψ and ϕ can degrade TBNN prediction accuracy in varying geometries. Furthermore, because each geometry type in this study uses a distinct domain configuration, the absence of a unified domain definition may further worsen the predictions when these two features are included. This limitation may be mitigated

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

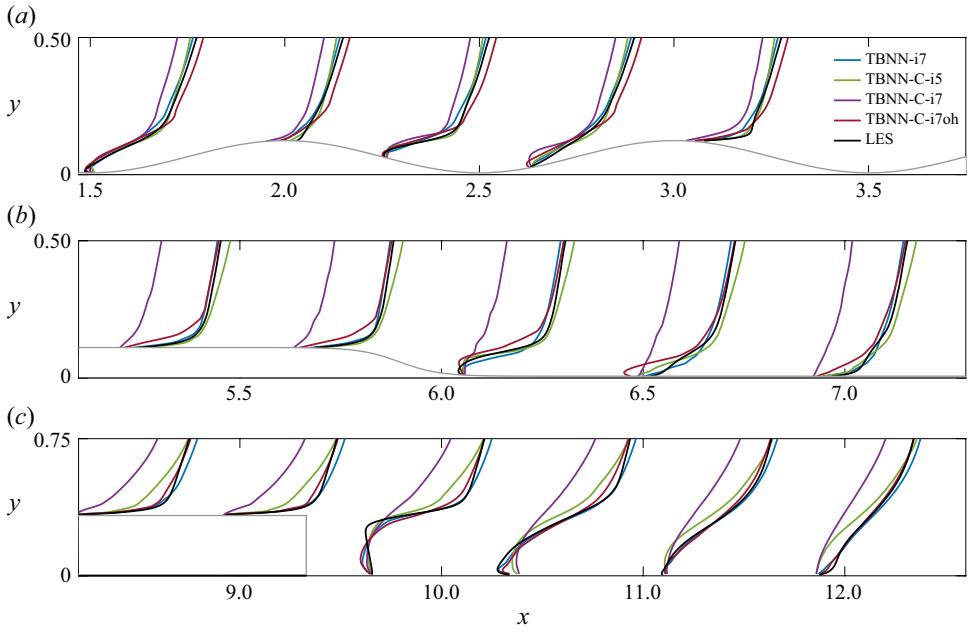


Figure 19. Predicted velocity profiles for the three bottom geometries using the combined-dataset training strategies TBNN-C-i5, TBNN-C-i7 and TBNN-C-i7oh, shown together with the predictions from TBNN-i7 and the LES results.

by introducing additional features that better characterise the global geometry or by incorporating explicit geometric categorisation during training, as in the TBNN-C-i7oh model, which uses one-hot encoding to distinguish different geometry types. As shown in figure 19 and table 8, TBNN-C-i7oh yields noticeably improved velocity predictions. These results suggest that more advanced ML-based geometric encoding strategies may provide a promising direction for further enhancing the generalisability of the present TBNN framework.

6. Conclusion

In this study, we present an ML-based turbulence closure model that can be directly embedded into a RANS solver without relying on any baseline model calculations. The closure model is developed based on the theoretical framework of the algebraic eddy viscosity model proposed by Pope (1975) and the modern ML framework introduced by Ling & Templeton (2015).

We propose two key contributions to address challenges encountered in both the *a priori* learning and *a posteriori* deployment stages of data-driven turbulence modelling. First, to enhance the expressiveness and robustness of the *a priori* model, we introduce geometry-informed features as additional inputs to the neural network. Specifically, in addition to the original local invariant features, we incorporate the stream function and velocity potential, both of which can be obtained straightforwardly before solving the governing equations. Combined with a rescaled eddy viscosity formulation, these additions eliminate the need for any baseline RANS model to obtain second-moment quantities.

For the *a posteriori* stage, we introduce a regression model using Random Forest to predict an appropriate eddy viscosity from given mean strain rates and anisotropic Reynolds stress tensors. This regression model is directly embedded into a RANS solver,

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

allowing the flow solution to be computed iteratively in a manner similar to conventional turbulence models, without relying on any baseline model calculations. We present three canonical flow configurations, namely the wavy bottom, smoothed step and backward-facing step, to demonstrate the capabilities of our model in channel flow with a variety of bottom geometries. We demonstrate that incorporating stream functions and velocity potentials significantly improves the model's predictive performance for the anisotropic stress tensor in the *a priori* stage. Furthermore, we show that the addition of the Random Forest regression model effectively predicts eddy viscosities.

While in the *a posteriori* evaluation, our TBNN model consistently delivers accurate predictions of the mean flow field when compared with the LES datasets, several open issues require further development and improvement. First, because RANS does not account for the anisotropic nature of turbulence, incorporating anisotropy, either by adopting an anisotropic eddy viscosity or by introducing an external source term, could be a potential improvement. However, how these additions affect solution convergence in the *a posteriori* stage warrants further investigation. Second, although we focus on channel flows in this study, extending the model to a broader variety of flow configurations remains an important direction for future research. In particular, the use of the stream function and velocity potential in external flows (e.g. flow around blunt bodies) or in three-dimensional flows may differ from the present setting. Third, while we have demonstrated simple training and prediction across different geometries, incorporating advanced ML techniques for categorising or mapping complex geometries could be a crucial component for future development.

Despite the need for further exploration to broaden the applicability of the present model, this study presents the first ML-based turbulence closure framework that eliminates the need for baseline model calculations. The proposed two-layer ML architecture, which combines TBNN with a regression model, offers a novel structure for ML-based turbulence closure.

Acknowledgements. The authors gratefully acknowledge the National Center for High-performance Computing (NCHC), National Applied Research Laboratories (NARLabs), Taiwan, for providing computational and storage resources.

Funding. This work was supported by the National Science and Technology Council (NSTC), Taiwan, under the Thermal Science and Fluid Dynamics Program (Grant Nos. 109-2221-E-002-028-MY3 and 112-2221-E-002-139-MY3) and the Ocean Engineering Division (Grant No. 112-2221-E-002-186-MY3).

Declaration of interests. The authors report no conflict of interest.

Appendix A. Neural network implementation

This study employs two learning components: a TBNN and an auxiliary regression model. The TBNN is responsible for predicting the dimensionless anisotropic Reynolds stress tensor, while the regression model provides a stable and consistent eddy viscosity estimate to enhance the robustness of the coupled RANS solver. This section outlines the preprocessing steps, network architectures and optimisation strategies for both components.

A crucial aspect of integrating machine learning (ML) into computational fluid dynamics (CFD) lies in the preprocessing of input data. To improve predictive accuracy, all input features are constrained to lie within a bounded range. For flow-related tensor quantities, it is essential to distinguish between non-dimensionalisation and feature scaling (Fang *et al.* 2023). In this study, all input features are first non-dimensionalised to ensure generalisability across different flow regimes and then scaled to the range [0, 1] to mitigate numerical instabilities caused by large variations in magnitude. Since the tensor invariants

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

and integrity basis tensors are constructed from the dimensionless strain-rate and rotation-rate tensors, whose magnitudes typically range from $O(1)$ to $O(10)$, large disparities among features may dominate the network's internal representation. To prevent any single feature from overwhelming the learning process, we normalise each component of the strain-rate and rotation-rate tensors by the maximum absolute value observed in the training dataset, ensuring their magnitudes do not exceed $O(1)$. The normalised tensors are defined as $S^*_{ij}/|S^*_{ij}|_{max}$ and $R^*_{ij}/|R^*_{ij}|_{max}$, where $|\cdot|_{max}$ denotes the maximum absolute value computed over the training samples.

To train the model, the flow field is sampled at discrete spatial locations to extract input–output vector pairs. These sampling points are generated from a finite-element mesh constructed using polygonal triangulation, with local refinement applied near boundaries to balance the distribution of feature magnitudes. For each flow field, approximately 4×10^3 to 5×10^3 training samples N_t are collected. This procedure transforms the continuous flow field into a representative dataset suitable for supervised learning, containing input tensor features and the corresponding anisotropic tensors.

The proposed TBNN adopts a one-point training approach, in which each spatial location is treated independently using a multilayer, fully connected feed-forward network. The architecture consists of an input layer, several hidden layers and an output layer. The input layer maps the features to a hidden representation of prescribed width. Each hidden layer applies a linear transformation followed by a LeakyReLU activation function with a negative slope of 0.01, and the output layer yields the coefficients associated with the tensor basis. The overall structure is summarised as

$$\texttt{Input} \rightarrow [\texttt{Linear} \rightarrow \texttt{LeakyReLU}]^{(L-2)} \rightarrow \texttt{Linear (Output)}\,, \tag{A1}$$

where L is the total number of layers, which is treated as a tuneable hyperparameter along with the hidden-layer width.

The model is implemented in `PyTorch`. To identify an optimal set of hyperparameters, the Optuna framework is employed to perform 50 optimisation trials. The tuneable parameters include the hidden-layer width (ranging from 60 to 180), the number of layers (between 5 and 10), the number of training epochs (200 to 800), batch size (16 to 256), learning rate (from 10^{-4} to 10^{-2} in logarithmic scale) and optimiser type (either Adam or SGD). The optimiser is dynamically selected using Optuna's categorical suggestion mechanism and instantiated via Python's `getattr` interface. To prevent overfitting and reduce computational cost, an early stopping criterion is applied: training is terminated if no improvement is observed in the validation loss for 50 consecutive epochs. This optimisation strategy allows efficient exploration of the hyperparameter space while maintaining model generalisation. Once the optimal hyperparameters are identified, the training data are partitioned into 20 subsets for K-fold cross-validation. This strategy is employed to improve the generalisation capability of the trained TBNN model. During the training phase, each fold is used as a validation set, while the remaining folds serve as the training data, ensuring that every sample contributes to both training and validation. In the inference stage, the final prediction is obtained by averaging the outputs of the 20 independently trained models. This ensemble approach helps to reduce variance and enhance the robustness of the TBNN predictions.

Additionally, a regression-based closure is constructed using a tree-based ensemble method. Similar to the TBNN, this model adopts a one-point training approach, where each spatial location is treated independently. The regression model takes as input the dimensionless strain-rate tensor and the TBNN-predicted anisotropic tensor, while the target variable is the eddy viscosity obtained from the *a priori* analysis. All input features are standardised using z-score normalisation. Hyperparameter tuning is performed via grid

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

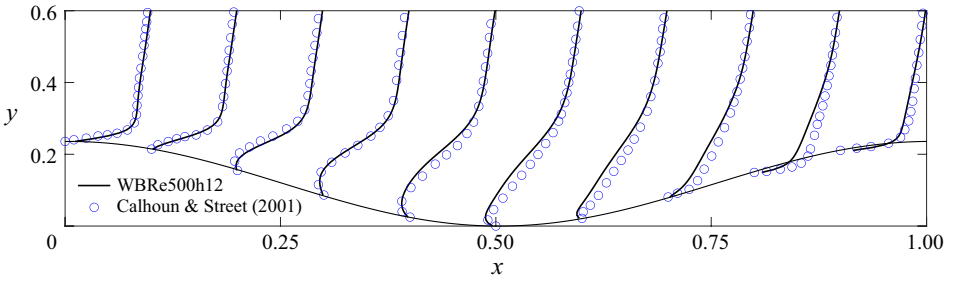


Figure 20. Comparison of the simulated mean velocity profiles at representative streamwise locations between case WBRe500h12 in the present LES dataset and the results of Calhoun & Street (2001).

search over the number of estimators (50, 100, 200), maximum tree depth (none, 5, 10, 20) and minimum samples per split (2, 5, 10). The optimal configuration is selected using five-fold cross-validation with the negative mean squared error as the scoring criterion. The final model is trained on the complete dataset using the `RandomForestRegressor` implementation provided by the `scikit-learn`.

Appendix B. LES dataset

For the cases involving a four-period wavy bottom and a smoothed step, LES were performed using an in-house finite-volume solver originally developed by Zang, Street & Koseff (1994). This solver has been successfully applied to various flow scenarios (e.g. Zang *et al.* 1994; Fringer & Street 2003; Arthur & Fringer 2014). The flow solver employs a dynamic mixed model as the LES sub-grid-scale (SGS) closure. This SGS model consists of a dynamic Smagorinsky model along with a momentum source reconstruction based on self-similarity (Zang, Street & Koseff 1993). For the backward-facing step configuration, simulations were carried out using the commercial software ANSYS Fluent. In this LES flow solver, we use the dynamic Smagorinsky–Lilly model (Germano *et al.* 1991; Lilly 1992) as the SGS model. In all cases, the top and bottom boundaries were specified as free-slip and no-slip walls, respectively, while periodic boundary conditions were imposed in both the streamwise (x) and spanwise (z) directions. The flow was driven by a constant pressure gradient along the x-axis. In all cases, the grids are refined towards the bottom so that the resolution of the bottom-most grid cell is approximately one wall unit. This allows the LES to be performed without employing a wall model. Details of computational set-up and data validation for each case are outlined as follows.

(i) For the four-periodic wavy-bottom case, using the mean flow depth H as the characteristic length scale ($\mathcal{L}$), the computational domain is defined as $(L,\ H,\ W) = (4,\ 1,\ 1)$, with a grid resolution of $(N_x,\ N_y,\ N_z) = (512,\ 192,\ 128)$. The bottom surface consists of four periods of a cosine wave with amplitude a. A total of nine cases were simulated, spanning various friction Reynolds numbers Re_τ and wave heights $h\ (= 2a)$, as summarised in table 3. The base case WBRe500h12 (see table 3) follows the set-up of Calhoun & Street (2001), who used the same numerical solver as in the present study, albeit with a coarser grid resolution. In our simulation, the grid is compressed and stretched vertically, resulting in a resolution of approximately 0.9 wall unit at the wave crest and 0.5 wall unit at the wave trough in the base case. Figure 20 compares the mean velocity profiles from our LES results with those of Calhoun & Street (2001), showing close agreement between the two studies. All other

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

simulation cases in this flow configuration were conducted using the same grid set-up as the base case.

(ii) For the case of flow over a smoothed step, using the largest depth as the characteristic length scale ($\mathcal{L}$) for spatial normalisation, the domain size is (8, 1, 1), with a grid resolution of $(N_x,\ N_y,\ N_z) = (512,\ 192,\ 128)$. The bottom boundary features a smoothed step geometry, which is described by the piecewise hyperbolic tangent function,

$$h(x) = \begin{cases} 0, & 0.000 \le x < 1.57, \\ \dfrac{h}{2}\left[\tanh\left(k(x-1.96)\right)+1\right], & 1.57 \le x < 2.35, \\ h, & 2.35 \le x < 5.50, \\ \dfrac{h}{2}\left[1-\tanh\left(k(x-5.90)\right)\right], & 5.50 \le x < 6.29, \\ 0, & 6.29 \le x < 8.00\,. \end{cases} \tag{B1}$$

Here, h denotes the step height and is set to $h = 0.11$. The steepness parameter k is fixed at $k = 10$. Same as in the previous configuration, the computational grid is compressed and stretched such that the resolution is sufficiently fine near the bottom wall to resolve detailed turbulent flow structures. A total of eight cases were simulated, covering various friction Reynolds numbers Re_τ and step heights h, as summarised in table 3. Among these cases, variations in step height correspond to ramp slopes ranging approximately from 0.37 to 0.62. We found that identifying an existing study with exactly the same geometry remains challenging. Therefore, to validate our numerical data, we focus on the flow over the downslope ramp and refer to a previous numerical study by El-Askary (2009), who employed a similar Reynolds number and geometry, but with an unsmoothed concave bottom angle for the downslope ramp. Case SSRe500h105 was used as the base case for comparison with the results of El-Askary (2009). In this case, the finest grid resolution near the bottom boundary is approximately 0.8 wall units at the top of the step and 0.9 wall units at the deepest point of the channel. Figure 21 compares the mean velocity profiles from our LES results with those from El-Askary (2009). It should be noted that the slight geometric differences between El-Askary (2009) and present study can result in a somewhat different circulation pattern immediately behind the backward-facing slope, which is reflected in the minor deviations observed in the velocity profiles. Despite this, figure 21 demonstrates that our LES simulation successfully captures the overall flow trend over the backward-facing slope of the smoothed step.

(iii) Simulation of flow over a backward-facing step was conducted by applying periodic boundary conditions to a channel flow with a rectangular step positioned at the centre. Using the largest depth as the characteristic length scale ($\mathcal{L}$) for spatial normalisation, the computational domain is defined as (13, 1, 0.67), with a grid resolution of $(N_x,\ N_y,\ N_z) = (512,\ 192,\ 128)$. A total of six cases were simulated, covering various friction Reynolds numbers Re_τ and step heights h, as summarised in table 3. We use case BSRe1380h33 as the base case for comparison with results from the numerical study by Le, Moint & Kim (1997). In the base case, the finest grid resolution near the bottom boundary is approximately 1.5 wall units at the top of the step and 1.3 wall units at the deepest part of the channel. Figure 22 compares the mean velocity profiles from our LES results with those of Le *et al.* (1997), showing close agreement between the two studies.

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

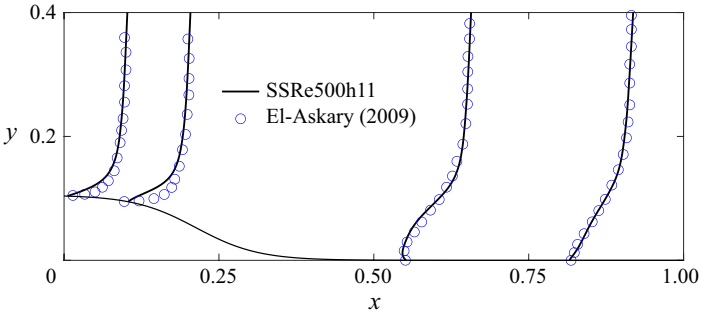


Figure 21. Comparison of the simulated mean velocity profiles at representative streamwise locations between case SSRe500h11 in the present LES dataset and the results of El-Askary (2009).

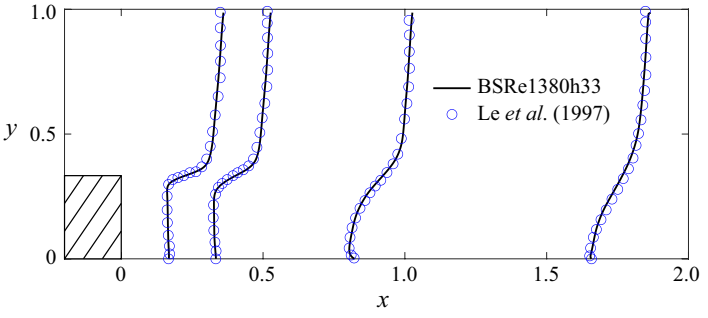


Figure 22. Comparison of the simulated mean velocity profiles at representative streamwise locations between case BSRe1380h33 in the present LES dataset and the results of Le *et al.* (1997).

References

Agrawal, A. & Koutsourelakis, P.-S. 2024 A probabilistic, data-driven closure model for RANS simulations with aleatoric, model uncertainty. *J. Comput. Phys.* **508**, 112982.

Amarloo, A., Forooghi, P. & Abkar, M. 2022 Frozen propagation of Reynolds force vector from high-fidelity data into Reynolds-averaged simulations of secondary flows. *Phys. Fluids* **34**, 115102.

Arthur, R.S. & Fringer, O.B. 2014 The dynamics of breaking internal solitary waves on slopes. *J. Fluid Mech.* **761**, 360–398.

Banerjee, S., Krahl, R., Durst, F. & Zenger, C. 2007 Presentation of anisotropy properties of turbulence, invariants versus eigenvalue approaches. *J. Turbul.* **8**, N32.

Beck, A. & Kurz, M. 2021 A perspective on machine learning methods in turbulence modeling. *GAMM-Mitt.* **44** (1), e202100002.

Beetham, S. & Capecelatro, J. 2020 Formulating turbulence closures using sparse regression with embedded form invariance. *Phys. Rev. Fluids* **5** (8), 084611.

Brunton, S.L., Noack, B.R. & Koumoutsakos, P. 2020 Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **52**, 477–508.

Cai, J., Angeli, P.-E., Martinez, J.-M., Dambli, G. & Lucor, D. 2024 Revisiting tensor basis neural network for Reynolds stress modeling: application to plane channel and square duct flows. *Comput. Fluids* **275**, 106246.

Calhoun, R.J. & Street, R.L. 2001 Turbulent flow over a wavy surface: neutral case. *J. Geophys. Res. Oceans* **9293** (C5), 9277–9293.

Duraisamy, K. 2021 Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence. *Phys. Rev. Fluids* **6** (5), 050504.

Duraisamy, K., Iaccarino, G. & Xiao, H. 2019 Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.* **51**, 357–377.

El-Askary, W.A. 2009 Turbulent boundary layer structure of flow over a smooth-curved ramp. *Comput. Fluids* **38**, 1718–1730.

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

FANG, Y., ZHAO, Y., WASCHKOWSKI, F., OOI, A.S.H. & SANDBERG, R.D. 2023 Toward more general turbulence models via multicase computational-fluid-dynamics-driven training. *AIAA J.* **61** (5), 1668–1681.
FRINGER, O.B. & STREET, R.L. 2003 The dynamics of breaking progressive interfacial waves. *J. Fluid Mech.* **494**, 319–353.
GENEVA, N. & ZABARAS, N. 2019 Quantifying model form uncertainty in Reynolds-averaged turbulence models with Bayesian deep neural networks. *J. Comput. Phys.* **383**, 125–147.
GERMANO, M., PIOMELLI, U., MOIN, P. & CABOT, W.H.A. 1991 A dynamic mixed subgrid-scale model and its application to turbulent recirculation flows. *Phys. Fluids* **3**, 1760–1765.
LE, H., MOINT, P. & KIM, J. 1997 Direct numerical simulation of turbulent flow over a backward-facing step. *J. Fluid Mech.* **330**, 349–374.
LILLY, D.K. 1992 A proposed modification for the Germano subgrid-scale closure model. *Phys. Fluids* **4**, 633–635.
LING, J. & TEMPLETON, J. 2015 Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier–Stokes uncertainty. *Phys. Fluids* **27** (8), 085103.
LING, J., JONES, R. & TEMPLETON, J. 2016*a* Machine learning strategies for systems with invariance properties. *J. Comput. Phys.* **318**, 22–35.
LING, J., KURZAWSKI, A. & TEMPLETON, J. 2016*b* Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* **807**, 155–166.
LOGG, A., MARDAL, K.-A. & WELLS, G. 2012 *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*, vol. 84. Springer Science & Business Media.
LUMLEY, J.L. 1979 Computational modeling of turbulent flows. *Adv. Appl. Mech.* **18**, 123–176.
MACEDO, M.S.S., CRUZ, M.A., BRENER, B.P. & THOMPSON, R.L. 2024 A data-driven turbulence modeling for the Reynolds stress tensor transport equation. *Intl J. Numer. Meth. Fluids* **96**, 1194–1214.
MANDLER, H. & WEIGAND, B. 2022 On frozen-RANS approaches in data-driven turbulence modeling: practical relevance of turbulent scale consistency during closure inference and application. *Intl J. Heat Fluid Flow* **95**, 108980.
MCCONKEY, R., YEE, E. & LIEN, F.S. 2022 Deep structured neural networks for turbulence closure modeling. *Phys. Fluids* **34**, 035110.
MILANI, P.M., LING, J. & EATON, J.K. 2021 On the generality of tensor basis neural networks for turbulent scalar flux modeling. *Intl Commun. Heat Mass Transfer* **128**, 105626.
PARISH, E.J. & DURAISAMY, K. 2016 A paradigm for data-driven predictive modeling using field inversion and machine learning. *J. Comput. Phys.* **305**, 758–774.
POPE, S.B. 1975 A more general effective-viscosity hypothesis. *J. Fluid Mech.* **72** (2), 331–340.
POPE, S.B. 2000 *Turbulent Flows*. Cambridge University Press.
RICHARDSON, L.F. 1922 *Weather Prediction by Numerical Process*. Cambridge University Press.
SHARMA, P., CHUNG, W.T., AKOUSH, B. & IHME, M. 2023 A review of physics-informed machine learning in fluid mechanics. *Energies* **16** (5), 2343.
SIMMONS, D.J., THOMAS, F.O., CORKE, T.C. & GLUZMAN, I. 2024 Experimental characterization and similarity scaling of smooth-body flow separation and reattachment on a two-dimensional ramp geometry. *J. Fluid Mech.* **1000**, A12.
THOMPSON, R.L., SAMPAIO, L.E.B., DE BRAGANÇA ALVES, F.A.V., THAIS, L. & MOMPEAN, G. 2016 A methodology to evaluate statistical errors in DNS data of plane channel flows. *Comput. Fluids* **130**, 1–7.
VINUESA, R. & BRUNTON, S.L. 2022 Enhancing computational fluid dynamics with machine learning. *Nat. Comput. Sci.* **2** (6), 358–366.
WANG, J.-X., WU, J.-L. & XIAO, H. 2017 Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data. *Phys. Rev. Fluids* **2** (3), 034603.
WASCHKOWSKI, F., ZHAO, Y., SANDBERG, R. & KLEWICKI, J. 2022 Multi-objective CFD-driven development of coupled turbulence closure models. *J. Comput. Phys.* **452**, 110922.
WEATHERITT, J. & SANDBERG, R. 2016 A novel evolutionary algorithm applied to algebraic modifications of the RANS stress–strain relationship. *J. Comput. Phys.* **325**, 22–37.
WEATHERITT, J. & SANDBERG, R.D. 2017 The development of algebraic stress models using a novel evolutionary algorithm. *Intl J. Heat Fluid Flow* **68**, 298–318.
WU, J., XIAO, H. & PATERSON, E. 2018 Physics-informed machine learning approach for augmenting turbulence models: a comprehensive framework. *Phys. Rev. Fluids* **3** (7), 074602.
WU, J., XIAO, H., SUN, R. & WANG, Q. 2019 Reynolds-averaged Navier–Stokes equations with explicit data-driven Reynolds stress closure can be ill-conditioned. *J. Fluid Mech.* **869**, 553–586.
XIAO, H., WU, J.-L., LAIZET, S. & DUAN, L. 2020 Flows over periodic hills of parameterized geometries: a dataset for data-driven turbulence modeling from direct simulations. *Comput. Fluids* **200**, 104431.

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press

ZANG, Y., STREET, R.L. & KOSEFF, J.R. 1993 A dynamic mixed subgrid-scale model and its application to turbulent recirculation flows. *Phys. Fluids* **5**, 3186–3196.

ZANG, Y., STREET, R.L. & KOSEFF, J.R. 1994 A non-staggered grid, fractional step method for time-dependent incompressible Navier–Stokes equations in curvilinear coordinates. *J. Comput. Phys.* **114** (1), 18–33.

ZHANG, X., XIAO, H., LUO, X. & HE, G. 2022 Ensemble Kalman method for learning turbulence models from indirect observation data. *J. Fluid Mech.* **949**, A26.

ZHANG, X.-L., XIAO, H., JEE, S. & HE, G. 2023 Physical interpretation of neural network-based nonlinear eddy viscosity models. *Aerosp. Sci. Technol.* **142**, 108632.

ZHAO, C., ZHANG, F., LOU, W., WANG, X. & YANG, J. 2024 A comprehensive review of advances in physics-informed neural networks and their applications in complex fluid dynamics. *Phys. Fluids* **36** (10), 101301.

ZHAO, Y., AKOLEKAR, H.D., WEATHERITT, J., MICHELASSI, V. & SANDBERG, R.D. 2020 RANS turbulence model development using CFD-driven machine learning. *J. Comput. Phys.* **411**, 109413.

https://doi.org/10.1017/jfm.2026.11414 Published online by Cambridge University Press