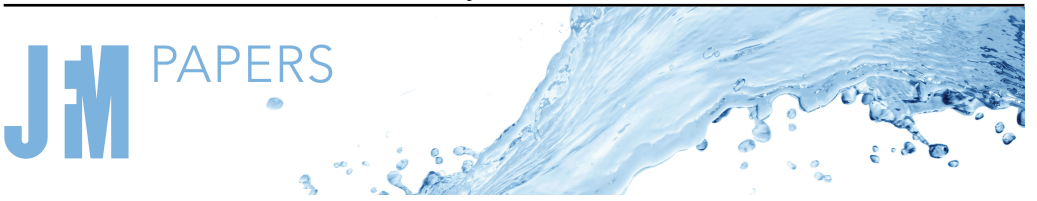
J. Fluid Mech. (2026), *vol.* 1036, A54, doi:10.1017/jfm.2026.11654



Progressive mixture-of-experts with autoencoder routing for continual RANS turbulence modelling

Haoyu Ji[1,2], **Yinhang Luo**[1], **Hanyu Zhou**[1,2] and **Yaomin Zhao**[1,2]

[1]HEDPS, Center for Applied Physics and Technology, School of Mechanics and Engineering Science, Peking University, Beijing 100871, PR China

[2]State Key Laboratory for Turbulence and Complex Systems, School of Mechanics and Engineering Science, Peking University, Beijing 100871, PR China

Corresponding author: Yaomin Zhao, yaomin.zhao@pku.edu.cn

(Received 1 January 2026 UTC; revised 5 April 2026 UTC; accepted 8 May 2026 UTC)

Developing Reynolds-averaged Navier–Stokes (RANS) turbulence models that remain accurate across diverse flow regimes is a long-standing challenge. In this work, we propose a novel framework, termed the progressive mixture-of-experts (PMoE), designed to enable continual learning for RANS turbulence modelling. The framework employs a modular autoencoder-based router to associate each flow scenario with a specialised turbulence model, referred to as an expert. When a new flow regime cannot be adequately represented by the existing router and expert set, a new expert together with its routing component can be introduced at low cost, without modifying or degrading previously trained ones, thereby naturally avoiding catastrophic forgetting. The framework is applied to a range of flows with distinct physical characteristics, including aerofoil wake, channel, periodic hill and square duct flows. The resulting PMoE model effectively integrates multiple experts and achieves improved predictive accuracy across both seen and unseen test cases that differ in operating conditions or configurations. Owing to sparse activation, model expansion does not incur additional computational cost during inference. The proposed framework therefore provides a scalable pathway towards lifelong-learning turbulence models for industrial computational fluid dynamics.

Key words: turbulence modelling, turbulence simulation

1. Introduction

Reynolds-averaged Navier–Stokes (RANS), supported by turbulence closure models, remains a predominant computational fluid dynamics (CFD) approach in industry. The RANS approach maintains a balance between computational efficiency and accuracy

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

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

for predicting mean flow quantities, making it critical for applications across aerospace, automotive and energy sectors (Rumsey 2014; Bush *et al.* 2019). The main difficulty, however, lies in closing the Reynolds stress tensor, especially for complex flows such as boundary layer transition, separation and secondary flows. Traditional closures are based on physical arguments, empirical tuning or, more recently, data-driven calibration (Launder & Sharma 1974; Spalart & Allmaras 1992; Wilcox 2008; Ling, Kurzawski & Templeton 2016).

The demand for improving turbulence models has led to the use of machine learning methods (Duraisamy, Iaccarino & Xiao 2019; Brunton, Noack & Koumoutsakos 2020). Many machine learning methods have been introduced into turbulence modelling, including Bayesian optimisation (Xiao *et al.* 2016), ensemble Kalman filter (Zhang *et al.* 2022), adaboost decision trees (Ling & Templeton 2015), random forests (Wang, Wu & Xiao 2017), tensor basis neural networks (Ling *et al.* 2016), and symbolic regression (Weatheritt & Sandberg 2016; Schmelzer, Dwight & Cinnella 2020; Zhao *et al.* 2020), among others. In these studies, the modelling strategies vary considerably, from correcting the parameters of linear eddy viscosity models to forming completely new nonlinear eddy viscosity models, with the optimisation targets varying from predicting transition (Duraisamy, Zhang & Singh 2015; Zhang *et al.* 2023) to characterising boundary layer separation (Zhu *et al.* 2019; Wu & Zhang 2023). These models, which are typically trained to improve predictive accuracy for specific cases, can thus be termed as expert models.

Despite considerable advances, achieving robust generalisation for data-driven turbulence models remains not only an active research area but also a significant challenge. Based on the accumulated experience from industrial turbulence simulations, evidence suggests that a universal, simple and local turbulence model may be difficult to achieve (Rumsey & Coleman 2022). To enhance generalisation for traditional turbulence models, one approach is to perform parameter calibration in various application scenarios. The GEKO model, for example, provides empirical parameter combinations for the $k-\omega$ model under a wide range of cases (Menter & Matyushenko 2025). This cross-case generalisation capability, i.e. the capability to make accurate predictions in diverse flow scenarios, is particularly important and even more challenging for models trained by machine learning methods (Sandberg & Zhao 2022).

To achieve generalisation across diverse flow scenarios, researchers have explored numerous approaches that leverage model architecture design, training data diversity, feature engineering and physical constraints. These efforts can be broadly categorised into three distinct paradigms. The first strategy focuses on parameter calibration using data-driven approaches to enhance adaptability. For instance, Bin, Huang & Yang (2023) constructed a unified model by fine-tuning the parameters of the Spalart–Allmaras (SA) model, thereby extending its applicability across various flows. The second approach seeks to improve generalisation through multi-case training strategies, where the model is exposed to distinct geometries and physical characteristics during the learning phase. Notable examples include the symbolic-regression-based training developed by Fang *et al.* (2023), the multi-case surrogate optimisation by Amarloo *et al.* (2023), and the progressive data-augmented framework to incrementally enhance cross-case generalisation by Rincón *et al.* (2025). The third category involves model aggregation or ensemble methods, which utilise machine-learning-based weighting functions to blend multiple baseline models. In a series of recent studies (de Zordo-Banliat *et al.* 2024; Cherroud *et al.* 2025; Oulghelou *et al.* 2025), machine learning methods were employed to train weighting functions, successfully leveraging the strengths of different data-driven models to achieve improved accuracy across multiple cases. Parallel to these RANS-based developments, significant progress has also been made in wall-modelled large-eddy simulations (WMLES)

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

(Lozano-Durán & Bae 2023; Arranz *et al.* 2024), in which a model trained on several canonical flows exhibits good agreement with reference data for both canonical tests and realistic aircraft configurations.

It should be noted that most existing data-driven models,enhanced by either multi-case joint training or model aggregation, are considered reliable mostly within the vicinity of their training distributions, while applying to completely unknown flow regimes remains a critical challenge. In practical CFD applications, users are frequently confronted with novel flow configurations and operating conditions distinct from prior experiences. Directly training the existing model with newly obtained data often degrades its original performance, which is known as catastrophic forgetting (French 1999). A common approach is to retrain the model jointly on both pre-existing and new data, but this approach might face a series of challenges. As new regimes arise continuously in realistic applications, joint retraining can be hindered by complex optimisation tasks, data accessibility issues, and performance degradation on previously well-modelled cases, all of which motivate continual learning. In order to achieve continual learning for WMLES, Zhang *et al.* (2025) proposed an additive framework for training wall models, showing promising success in a series of cases. Nevertheless, novel frameworks with the capability of continual learning are still needed for developing generalisable RANS models.

To achieve a framework capable of sustainable generalisation, we propose a novel approach termed progressive mixture-of-experts (PMoE) in the present study. The PMoE method is founded on the mixture-of-experts (MoE) architecture (Jacobs *et al.* 1991), which has been the subject of much research in the field of machine learning (Shazeer *et al.* 2017; Fedus, Zoph & Shazeer 2022) and has become very popular in the field of large language models due to its scalability (Du *et al.* 2022; Dai *et al.* 2024). Building on the typical architecture consisting of a router and a group of experts, the MoE offers a distinct advantage for turbulence modelling, i.e. the decomposition of a complex, high-dimensional physical problem into manageable sub-tasks. Central to this architecture is the mechanism of sparse activation. Unlike monolithic models, the router in an MoE network activates only a specific subset of expert models for a given local flow state, leaving the majority silent during the inference process. This property is crucial for industrial CFD applications, as it does not incur the prohibitive computational cost usually associated with large-scale neural networks, while achieving sufficient representational expressiveness in order to deal with various scenarios featuring different flow physics.

One important novel feature of the present PMoE framework, which is missing for the generic MoE structure, is the continual learning capability. This is in order to achieve sustainable generalisation, enabling the model to incorporate newly introduced datasets without full retraining. Accordingly, the PMoE is designed to be equipped with a modular autoencoder-based router. The PMoE framework is applied to develop a turbulence model by successively introducing the data of four flow regimes, including aerofoil wake, channel, periodic hill and square duct flows. The trained PMoE model is then extensively assessed, validating its performance in training cases and unseen cases with various operating conditions and configurations. The objective is mainly to show the ability of PMoE to incorporate new flow scenarios without retraining or degrading existing experts.

Before introducing the framework in detail, it is important to highlight three key issues that need to be addressed to develop a PMoE turbulence model generalisable for various flow regimes. First, the PMoE framework needs to classify diverse flow scenarios in an unsupervised manner using only RANS data, and this requirement brings a significant challenge regarding the selection and extraction of appropriate physical features for the pre-processing procedure to serve as effective model inputs. Second, the router architecture must support continual learning capabilities, determining whether a new case

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

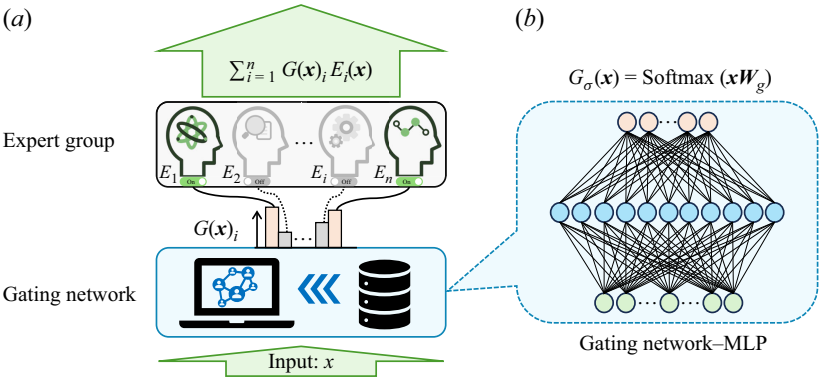


Figure 1. Schematic of the generic MoE framework with an MLP Softmax gating network. (*a*) The whole process of the MoE framework. (*b*) Zoom-in view of a gating network based on MLP.

belongs to a known flow regime or represents a novel one. For the latter, the architecture must allow for the rapid addition of new modules without the need for retraining. Finally, the distinct nature of turbulence across different flows implies that a unified model form may be insufficient (Duraisamy *et al.* 2019), and different flow regimes often require fundamentally different correction forms. Consequently, the design of specific expert architectures tailored to these varying physical characteristics is crucial for the success of a truly generalisable model. These topics will be addressed in detail in the following sections.

This paper is organised as follows. The structure of the new framework is discussed in § 2. The training process and the setting of cases are introduced in § 3. The model assessment, validation and computational cost analysis are given in § 4. Finally, conclusions are offered in § 6.

2. Methodology

The MoE framework, which has the potential to develop generalisable turbulence models, is introduced in § 2.1. Moreover, the router part of a generic MoE framework is typically implemented by multi-layer perceptron (MLP), which enables multi-task learning but requires further improvement for continual learning. In § 2.2, we introduce the PMoE structure, demonstrating how the newly proposed framework facilitates continual generalisation of turbulence modelling.

2.1. *Introduction to the generic MoE framework*

As shown in figure 1, the generic MoE consists of a set of n experts $\{E_1, \ldots, E_n\}$ and a gating network (also named a router) G, and the key advantage of this framework is the decomposition of a complex, high-dimensional problem into manageable sub-tasks dealt by different experts. For turbulence modelling, each expert E_i can be viewed as a specialised closure model tailored for a specific flow regime, while the router G functions as a flow regime classifier.

For a set of n experts, the output $\boldsymbol{y}$ for a given input flow state $\boldsymbol{x}$ is the weighted sum of expert predictions:

$$\boldsymbol{y} = \sum_{i=1}^{n} G(\boldsymbol{x})_i \; E_i(\boldsymbol{x}). \tag{2.1}$$

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

Here, $G(\boldsymbol{x})_i$ is the ith component of the output vector of G, and represents the probability (or confidence) that the corresponding flow $\boldsymbol{x}$ belongs to the regime governed by expert i. A simple choice of the gating function (Jordan & Jacobs 1994), as shown in figure 1(*b*) for example, is to multiply the input $\boldsymbol{x}$ by a trainable weighting matrix $\boldsymbol{W}_g$ and then apply the Softmax$(\cdot)$ function, as

$$G_\sigma(\boldsymbol{x}) = \text{Softmax}\left(\boldsymbol{x}\boldsymbol{W}_g\right), \tag{2.2}$$

where the subscript σ denotes the Softmax nonlinearity.

The key advantage of the MoE framework lies in its sparse activation mechanism, which dramatically reduces computational costs while maintaining model capacity. Typically, sparse MoE employs a top-k routing strategy where only the most relevant k experts are activated for each input. For example, we set $k = 1$ in our current implementation, thus the router selects the one most relevant expert per input token, ensuring that the other $n - 1$ experts remain completely dormant during model inference. With this feature of sparse activation, the computational cost of the MoE model, despite having a large pool of experts $\{E_1, \ldots, E_n\}$, scales only with the size of a single expert rather than the entire ensemble. Consequently, the inference cost remains nearly constant regardless of the total number of experts, enabling the deployment of extremely large-scale models with billions or even trillions of parameters with relatively low computational requirements (Shazeer *et al.* 2017; Fedus *et al.* 2022).

While the generic MoE offers a path towards modular turbulence modelling, applying it to industrial CFD requires a capability that standard architectures lack, i.e. continual learning. A truly generalisable turbulence model must be able to assimilate new flow physics (e.g. transitioning from simple shear flows to complex corner separations) without 'forgetting' previously learned regimes. However, standard MLP-based routers in figure 1 are highly susceptible to catastrophic forgetting (French 1999), as updating the weights to accommodate a new flow regime often degrades the classification accuracy for existing ones, highlighting the urgent need to extend the framework with continual learning capability.

In the broader machine learning literature, continual learning is attracting more and more interest, and the corresponding strategies can be generally classified into three paradigms (Wang *et al.* 2024): (i) regularisation-based methods, which constrain weight updates to preserve important parameters (e.g. Kirkpatrick *et al.* 2017); (ii) replay-based methods, which retain or generate representative samples from earlier tasks for joint retraining (e.g. Rolnick *et al.* 2019); and (iii) architecture-based methods, which dedicate separate model components to different tasks to prevent interference (e.g. Rusu *et al.* 2016). Among architecture-based approaches, Aljundi, Chakravarty & Tuytelaars (2017) proposed Expert Gate, a lifelong learning framework in which each task is associated with a dedicated autoencoder gate and expert network. At test time, the autoencoder with the lowest reconstruction error routes the input to its corresponding expert, while the reconstruction errors also quantify inter-task relatedness to guide knowledge transfer when training new experts. This design demonstrated effective sequential task learning for image classification. The present PMoE framework builds on this broad principle of modular autoencoder-based routing, while introducing several novel strategies essential for turbulence modelling, as discussed in the following sections.

2.2. *The PMoE framework for turbulence modelling*

As shown by the schematic in figure 2, the PMoE framework consists of three key components, including the procedure of pre-processing to extract flow features as model inputs, the router structure responsible for classifying various types of flows, and a group

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

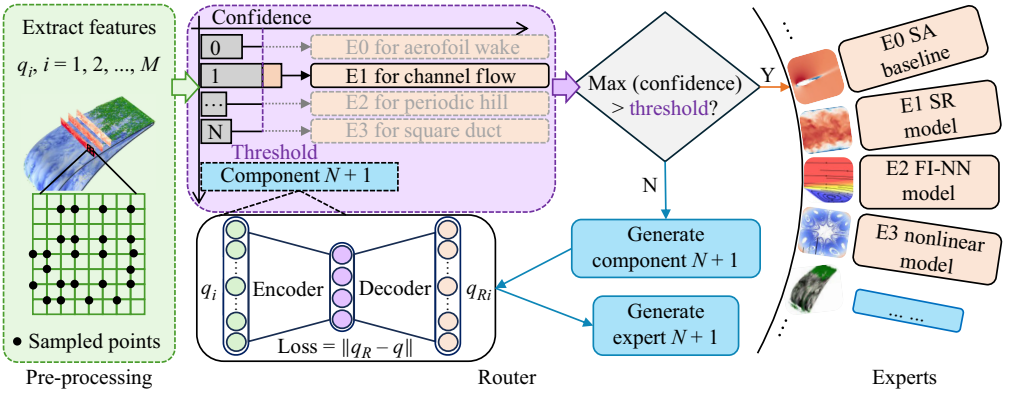


Figure 2. Schematic of the proposed PMoE framework. The framework consists of three parts: (i) a pre-processing stage that extracts physics-informed features from the baseline RANS field; (ii) a modular autoencoder-based router that classifies the incoming flow regime by comparing reconstruction errors across components, and triggers model expansion when no existing component achieves sufficient confidence; and (iii) a sparsely activated group of heterogeneous experts, each tailored to a specific flow regime. Here, SR means symbolic regression, and FI-NN means field inversion neural network.

of experts each trained for a specified flow scenario. In particular, the PMoE replaces the MLP router with a modular bank of autoencoders, each dedicated to recognising a specific flow signature. This ensures that the introduction of a new expert and its corresponding router module does not interfere with the existing modules. Accordingly, the PMoE framework is expected to develop progressively with new datasets introduced successively, and the details are discussed in the following.

2.2.1. *Feature extraction and physics-informed sampling*

During the pre-processing stage, the input to the PMoE framework relies on local flow features extracted from a baseline RANS calculation. In the present study, we use the SA model (Spalart & Allmaras 1992) as the baseline due to its wide applications. Following previous data-driven RANS efforts (Ling & Templeton 2015; Wang *et al.* 2017), we construct an input vector $\boldsymbol{x} = [q_1, q_2, \ldots, q_M]$ consisting of $M = 7$ flow features, as detailed in table 1. Note that the listed variables are normalised using $A/(|A| + |B|)$ to ensure consistent scaling across different flow regimes.

We note that the features listed in table 1 are not universally effective across all flow regimes. Prior studies (Yin *et al.* 2020; de Zordo-Banliat *et al.* 2024; Cherroud *et al.* 2025) suggest that a subset of these variables may often be sufficient for specific cases, and methods such as permutation feature importance (PFI) (Wu *et al.* 2025) can be employed to identify the relevant features for a specific case. Nevertheless, to accommodate future applications that may involve a broad range of flow regimes with distinct critical features, we adopt the full set as the model input.

A critical challenge in training a generalisable router is ensuring that the training data represent the dynamically active regions of the flow, rather than being dominated by the free stream or quiescent zones. Therefore, we employ a physics-informed sampling strategy during the pre-processing stage. Rather than uniformly sampling the entire computational domain, the sampling region is restricted based on the underlying flow physics. Grid points are selected only where either the strain-rate magnitude $\|\boldsymbol{S}\|$ or the vorticity magnitude $\|\boldsymbol{W}\|$ exceeds a specific threshold relative to the domain statistics.

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

Feature (q_i)	Description	Raw feature ($\hat{q}_i$)	Normalisation factor (q_i^*)
q_1	Vorticity magnitude	$\|\boldsymbol{W}\|$	$\|\boldsymbol{S}\|$
q_2	Ratio of excess rotation rate to strain rate (Q criterion)	$\frac{1}{2}(\|\boldsymbol{W}\|^2 - \|\boldsymbol{S}\|^2)$	$\|\boldsymbol{S}\|^2$
q_3	Ratio of pressure normal stresses to shear stresses	$\sqrt{\frac{\partial \boldsymbol{P}}{\partial x_i}\frac{\partial \boldsymbol{P}}{\partial x_i}}$	$\frac{1}{2}\rho\frac{\partial U_k^2}{\partial x_k}$
q_4	Distance to the nearest wall	d	L_{ref}
q_5	SA viscosity	$\tilde{\nu}$	100ν
q_6	Pressure gradient along streamline	$U_l\frac{\partial \boldsymbol{P}}{\partial x_l}$	$\sqrt{\frac{\partial \boldsymbol{P}}{\partial x_j}\frac{\partial \boldsymbol{P}}{\partial x_j}U_iU_i}$
q_7	Non-orthogonality between velocity and its gradient (Gorle *et al.* 2012)	$\left\|U_iU_j\frac{\partial U_i}{\partial x_j}\right\|$	$\sqrt{U_lU_lU_i\frac{\partial U_i}{\partial x_j}U_k\frac{\partial U_k}{\partial x_j}}$

Table 1. Flow features used as inputs in the framework: $\boldsymbol{W}$ is the rotation-rate tensor, $\boldsymbol{S}$ is the strain-rate tensor, ρ is the fluid density, ν is the fluid viscosity, $\tilde{\nu}$ is the SA modified turbulent eddy viscosity, d is the distance to the nearest wall, L_{ref} is the reference height, and U_i is the mean velocity. The normalised feature q_i is obtained by normalising the corresponding raw features value $\hat{q}_i$ with normalisation factor q_i^* according to $q_i = \hat{q}_i/(|\hat{q}_i| + |q_i^*|)$.

Specifically, we define the sampling mask $\mathcal{M}$ such that a point is included if

$$\|\boldsymbol{W}_j\| > \epsilon \times \text{median}(\|\boldsymbol{W}\|) \quad \text{or} \quad \|\boldsymbol{S}_j\| > \epsilon \times \text{median}(\|\boldsymbol{S}\|), \tag{2.3}$$

where $\|\boldsymbol{W}_j\|$ and $\|\boldsymbol{S}_j\|$ denote the magnitudes of vorticity and strain-rate at the jth sampled point, respectively, and ϵ is a thresholding parameter set to 0.05 in the present study. A sensitivity test has been performed, and the results are robust within $\epsilon \in \{0.01, 0.05, 0.1\}$. This criterion ensures that the selected samples are concentrated in boundary layers, shear layers and separation zones, etc., effectively filtering out grid points of limited physical significance such as those in free-stream regions. From this masked region, N points are randomly sampled to form the input matrix $\boldsymbol{X} \in \mathbb{R}^{N\times M}$. The masking criterion in (2.3) is applied only during the sampling stage for the PMoE router. In regions with limited physical significance, such as free-stream regions, the baseline RANS predictions are generally adequate to provide accurate predictions, thus no extra corrections are needed.

2.2.2. *Modular autoencoder-based router for continual learning*

The core of the PMoE framework is its ability to determine whether the incoming case belongs to a known scenario or represents a novel one that requires model expansion. As illustrated in figure 3, the PMoE router features a modular architecture, in which each router component adopts a standard autoencoder architecture following the design of Hinton & Salakhutdinov (2006), aiming to compress and reconstruct the feature space of a specific flow regime. The idea of learning internal representations through minimising input reconstruction traces back to the back-propagation framework of Rumelhart, Hinton & Williams (1986). The network in this study consists of an encoder f_θ mapping the input $\boldsymbol{x}$ to a latent representation $\boldsymbol{z}$, and a decoder g_ϕ reconstructing it as $\hat{\boldsymbol{x}}$. The network parameters of the encoder and the decoder shown in figure 3(*a*) are represented by θ and ϕ, respectively. The training objective for the autoencoder is to minimise the reconstruction loss over the N sampled points, as

$$\boldsymbol{z} = f_\theta(\boldsymbol{x}), \quad \hat{\boldsymbol{x}} = g_\phi(\boldsymbol{z}), \quad \theta, \phi = \arg\min_{\theta,\phi} \mathcal{L}(\boldsymbol{x}, \hat{\boldsymbol{x}}). \tag{2.4}$$

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

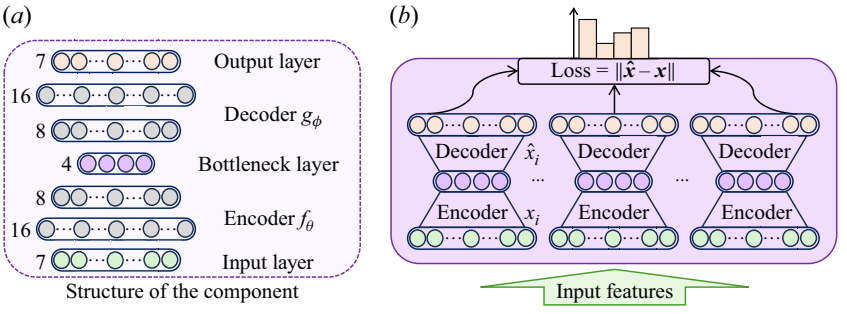


Figure 3. Structure of the router based on an autoencoder. (*a*) Neural network architecture design for the autoencoder component. (*b*) The modular architecture of the router.

Here, the reconstruction error $\mathcal{L}(\boldsymbol{x}, \hat{\boldsymbol{x}})$ is assessed as

$$\mathcal{L}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{N}\sum_{j}^{N} L_j = \frac{1}{N}\sum_{j}^{N}\sqrt{\frac{1}{M}\sum_{i}^{M}(\hat{x}_{i,j} - x_{i,j})^2}, \tag{2.5}$$

where for the jth point, $x_{i,j}$ is the ith component of its input feature vector, and $L_j = \sqrt{(1/M)\sum_{i=1}^{M}(\hat{x}_{i,j} - x_{i,j})^2}$ refers to its corresponding reconstruction error over the M features.

For every router component, a compact architecture is adopted, consisting of several hidden layers with moderate width to balance representational capacity and generalisation as shown in figure 3(*a*). Leaky rectified linear unit (Maas, Hannun & Ng 2013) activations are employed between layers to introduce nonlinearity, and the Adam optimiser (Kingma & Ba 2015) is used with initial learning rate 10^{-2}. A step-based learning-rate scheduler is employed, where the learning rate is reduced by a factor 0.98 every 100 epochs to improve convergence stability. The number of training epochs is capped at 5000 based on empirical convergence studies, while an early-stopping strategy is employed to improve efficiency and mitigate overfitting. For all component trainings, the loss consistently reaches a plateau well before the maximum epoch count, and extending the training further does not yield noticeable performance improvement. After the training of the kth router component C_k is converged, we set the 99.9th percentile of the reconstruction error of the training data as the corresponding threshold T_k, in order to eliminate the misleading influence of points with extreme distributions. A sensitivity test has been performed, and the results are robust with $T_k > 99\,\%$.

During the inference process, the router must determine if a new input flow belongs to a known regime or represents a novel physical scenario through the reconstruction error. For a new input case with N sampled points, a point j is considered recognised by component C_k if its local reconstruction error satisfies $L_j < T_k$. The global confidence level p_k of expert k regarding the current flow is defined as the relative frequency of recognised points

$$p_k = \frac{N_k}{N}, \quad \text{where } N_k = \sum_{j}^{N}\mathbb{I}(L_j < T_k), \tag{2.6}$$

where $\mathbb{I}(\cdot)$ is the indicator function.

The framework employs a winner-takes-all or top-1 gating strategy. The router selects the expert $E_{\mathcal{K}}$ corresponding to the component with the maximum confidence as

$$\mathcal{K} = \arg\max_{k} p_k. \tag{2.7}$$

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

However, activation is conditional and governed by a global acceptance threshold T_{accept}. When the maximum confidence $p_{\mathcal{K}}$ exceeds a threshold $T_{accept} = 90\,\%$, the expert $E_{\mathcal{K}}$ will be activated. Conversely, if $p_{\mathcal{K}}$ falls below T_{accept}, then the flow is deemed unknown, which triggers the continual learning process. In this case, a new autoencoder component C_{new} and a new expert E_{new} are initialised and trained on the new dataset. It should be emphasised that the expansion process is strictly additive. When a new regime is introduced, the corresponding router component and expert are trained, while all previously learned components and experts remain frozen. This modular design ensures isolation between old and new components during training, thus inherently overcoming catastrophic forgetting. Note that while a threshold $T_{accept} = 90\,\%$ is found to be effective for the distinct regimes in the present study, future industrial applications may require this hyperparameter to be tuned based on the desired sensitivity to novel flow physics.

The rationale for employing an autoencoder for flow classification rests on the premise that the hierarchical importance of physical features varies significantly across different flow scenarios, which can be seen from the feature importance analysis in Appendix A. During the routing stage, an input feature matrix $\boldsymbol{X} \in \mathbb{R}^{N\times M}$ is compressed into a low-dimensional bottleneck representation $\mathcal{H}_\theta = f_\theta(\boldsymbol{X})$ to extract the feature combinations that are most critical for reconstructing the local flow physics. Following the information bottleneck principle proposed by Tishby, Pereira & Bialek (2000), the training objective is formulated as

$$\min_{\theta} \left[I(\boldsymbol{X};\, \mathcal{H}_\theta) - \gamma\, I(\mathcal{H}_\theta;\, \boldsymbol{Y}) \right], \tag{2.8}$$

where θ denotes the encoder parameters, $I(\cdot;\,\cdot)$ represents mutual information, and γ governs the trade-off between compression and relevance. For the reconstruction task, the output matrix $\boldsymbol{Y} \in \mathbb{R}^{N\times M}$ is taken to be consistent with the input $\boldsymbol{X}$. The information bottleneck formulation implies the existence of a minimal informative feature subset $S \subseteq \{1, 2, \ldots, M\}$ such that the sub-matrix $\boldsymbol{X}_S \in \mathbb{R}^{N\times|S|}$ satisfies

$$I(\boldsymbol{X}_S;\, \boldsymbol{Y}) = I(\boldsymbol{X};\, \boldsymbol{Y}), \quad I(\boldsymbol{X}_{S'};\, \boldsymbol{Y}) < I(\boldsymbol{X};\, \boldsymbol{Y}), \quad \forall S' \subset S. \tag{2.9}$$

This implies that the autoencoder effectively isolates feature combinations that are strictly relevant to flow reconstruction. Consequently, for distinct flow scenarios, the network relies on different minimal subsets. This mechanism enables flow identification by exploiting the dependence of the latent representation $\mathcal{H}_\theta$ on flow-specific informative features. Note that the dimension of the latent bottleneck layer is a critical parameter that determines how much the input data can be compressed. We set the latent layer to have four nodes in the present study as shown in figure 3(*a*), which is based on a series of sensitivity tests.

2.2.3. *Sparsely activated expert group with heterogeneous formulation*

A distinct advantage of the PMoE framework is its explicit exploitation of heterogeneous expert formulations. Unlike data-driven models that typically enforce a single-form closure correction, e.g. a neural network predicting Reynolds stress anisotropy (Ling *et al.* 2016), the PMoE allows each expert E_k to adopt the mathematical structure best suited for its specific flow regime. This flexibility ensures that simple flows can be modelled by interpretable algebraic corrections, while complex non-equilibrium flows can leverage high-capacity neural networks. We remark that although conventional MoE architectures can in principle accommodate different expert structures, standard implementations almost universally employ homogeneous experts due to the requirement of joint end-to-end

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

training. In the PMoE framework, each expert is trained independently, removing this constraint and enabling fundamentally different closure strategies.

In the present implementation, we utilise the SA model (Spalart & Allmaras 1992) as the baseline model. The standard transport equation for the modified eddy viscosity $\tilde{\nu}$ is given by

$$\frac{\mathrm{D}\tilde{\nu}}{\mathrm{D}t} = \underbrace{C_{b1}\tilde{S}\tilde{\nu}}_{\mathcal{P}} + \underbrace{\frac{1}{\sigma}\left[\nabla\cdot((\nu+\tilde{\nu})\,\nabla\tilde{\nu}) + C_{b2}(\nabla\tilde{\nu})^2\right]}_{\mathcal{T}} - \underbrace{C_{w1}f_w\left(\frac{\tilde{\nu}}{d}\right)^2}_{\mathcal{D}}, \tag{2.10}$$

where $\mathcal{P}$, $\mathcal{T}$ and $\mathcal{D}$ represent the production, transport and destruction terms, respectively. The standard definitions for the model constants (such as C_{b1}, C_{b2}, σ and C_{w1}) and closure functions such as f_w are retained from Spalart & Allmaras (1992), and we only present the terms that will be modified here for brevity. In particular, the turbulent eddy viscosity is computed from $\nu_t = \tilde{\nu} f_{v1}$, and f_{v1} is defined as

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi \equiv \tilde{\nu}/\nu, \tag{2.11}$$

where $c_{v1} = 7.1$, and χ represents the ratio between $\tilde{\nu}$ and the molecular viscosity ν. Moreover, $\tilde{S}$ is the modified vorticity magnitude. Within the PMoE framework, we demonstrate the integration of three distinct expert formulations.

Type I: parameter correction for wall-attached flows. For equilibrium boundary layers where the baseline model structure is sound but parameter calibration is suboptimal, the expert provides a scalar correction to existing coefficients. For the wall-attached expert E_1, we employ a symbolic regression approach (Weatheritt & Sandberg 2016) to recalibrate the damping function f_{v1} as suggested by Bin *et al.* (2023). The expert predicts a corrected functional form that replaces the standard definition in (2.11) to better capture the logarithmic layer velocity profile.

Type II: correction of production term for flow separation. For flows dominated by separation and strong pressure gradients, the equilibrium assumption in the linear eddy viscosity hypothesis often fails. In such regions, conventional models may mischaracterise the eddy viscosity distribution near separation and within recirculation zones, which in turn affects the prediction of separation and reattachment. This behaviour is associated with the difficulty of linear eddy viscosity models in representing the non-equilibrium balance between production and dissipation under adverse pressure gradients and rapidly varying strain rates. Here, the expert E_2 is formulated as a neural network derived from the field inversion and machine learning (FIML) method. This method is chosen because it has been widely applied and demonstrated success in separation corrections (Parish & Duraisamy 2016), produces spatially varying correction fields that are well suited for flows where modelling errors are localised, and is compatible with standard RANS solvers. It imposes a spatially varying correction term β into the production term of the transport equation:

$$\mathcal{P}^* = \beta(\boldsymbol{x}) \cdot C_{b1}\tilde{S}\tilde{\nu}. \tag{2.12}$$

This multiplicative correction allows the model to locally suppress or enhance turbulence production in recirculation zones without altering the model behaviour in the free stream.

Type III: constitutive relation modification for secondary flows. For corner flows where linear eddy viscosity models fail to predict secondary motions due to the isotropy assumption, which constrains Reynolds stress anisotropy and suppresses cross-stream momentum redistribution, the expert E_3 modifies the stress–strain relationship itself. We adopt a

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

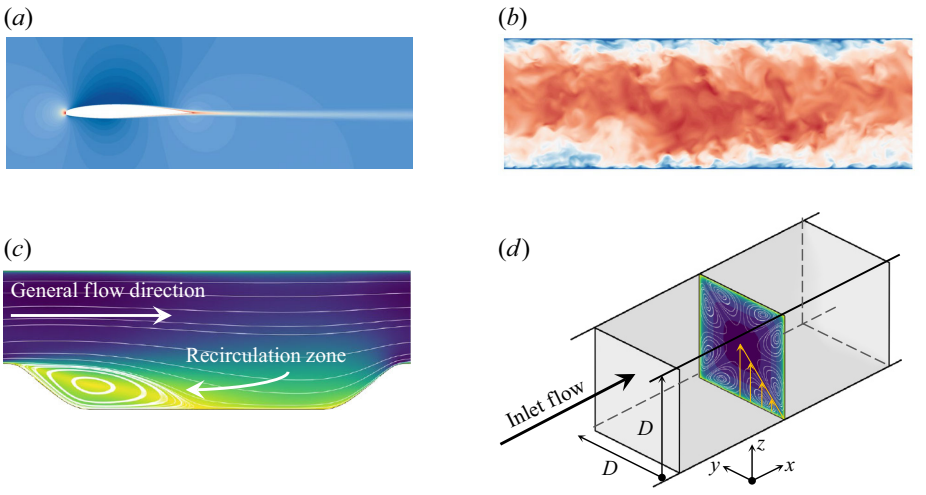


Figure 4. Schematic diagrams of typical examples of various flow regimes: (*a*) 2DANW, (*b*) fully developed channel flow, (*c*) periodic hill flow, (*d*) square duct flow.

data-driven calibration of the quadratic constitutive relation (QCR) (Spalart 2000):

$$\tau_{ij,QCR} = \tau_{ij}^{linear} - C_{cr1}(\boldsymbol{x})\left(O_{ik}\tau_{jk} + O_{jk}\tau_{ik}\right), \tag{2.13}$$

where O_{ik} is the normalised rotation tensor. Unlike the standard QCR where C_{cr1} is constant, the expert E_3 predicts a spatially varying C_{cr1} field to accurately capture corner vortices.

This modularity allows the PMoE framework to progressively integrate experts ranging from concise symbolic expressions to complex neural mappings without structural conflict, effectively decoupling the model architecture from the constraints of a single baseline formulation. This flexibility allows developers to select the most suitable expert design strategy for a given flow regime, and to update or expand the expert group in step with advances in the literature, thereby further enhancing the practicality, extensibility and efficiency of the framework.

While the formulation of individual experts reflects physical modelling considerations, the assignment of flow regimes during training and deployment is entirely feature-driven. No manual flow-type specification is required when processing a new case. The sampling strategy is likewise based on local kinematic indicators, ensuring minimal dependence on prior physical labelling.

3. Continual learning PMoE model with various cases introduced progressively

To demonstrate the sustainable generalisation capability of the PMoE framework, we employ a curriculum learning strategy. As shown in figure 4, each dataset is introduced sequentially on a case-by-case basis in order of increasing geometric complexity, starting from the aerofoil wake case (free-shear turbulence away from solid walls) and progressing to the channel flow cases (equilibrium wall-bounded flow with parallel flat wall), periodic hill cases (curved wall featuring separation), and the square duct case (featuring corner-induced secondary flow). It should be noted that the identification of whether a new dataset belongs to a known or novel regime is performed automatically by the router via the reconstruction-error-based confidence metric. However, the order in which datasets are introduced, and the design of the corresponding expert formulation at each stage, are

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

Flow regime	Case	Abbreviation	Stages	References
Stage 0: Wake	Two-dimensional aerofoil near wake	2DANW	Training	Nakayama (1985)
Stage 1: Channel	Fully developed channel at $Re_\tau = 2000$	C2000	Training	Lee & Moser (2015) and Yamamoto & Tsuji (2018)
	Fully developed channel at $Re_\tau = 5200$	C5200	Training	
	Fully developed channel at $Re_\tau = 8000$	C8000	Validation	
Stage 2: Periodic hill	Periodic hill with $\alpha = 0.8$	PH0p8	Validation	Xiao *et al.* (2020)
	Periodic hill with $\alpha = 1.0$	PH1p0	Training	
	Periodic hill with $\alpha = 1.2$	PH1p2	Validation	
	Periodic hill with $\alpha = 1.5$	PH1p5	Validation	
Stage 3: Square duct	Square duct at $Re = 2500$	SD2500	Training	Pinelli *et al.* (2010) and Vinuesa *et al.* (2018)
	Square duct at $Re = 3500$	SD3500	Validation	
	Square duct at $Re = 5693$	SD5693	Training	

Table 2. Summary of flow cases used for training and validation.

prescribed by the user, analogous to a practitioner sequentially encountering new flow scenarios in an industrial workflow. This obviously relies on a modeller's experience and physical insight. Importantly, because each router component and expert is trained independently with all previous modules frozen, the final model is invariant to the order of regime introduction.

The evolution of the model is denoted by stages S_0 to S_3, with the corresponding trained PMoE denoted as PMoE-S0, PMoE-S1, PMoE-S2 and PMoE-S3, respectively. At each stage, the router evaluates the novelty of the incoming data. We extract features from sampled points in the baseline RANS flow field, and construct input vector $\boldsymbol{x}$ and matrix $\boldsymbol{X} \in \mathbb{R}^{N\times M}$, where $M = 7$, $N = 10\,000$ in our training process. The input data are then evaluated through the existing router modules. If the maximum confidence p_K falls below the acceptance threshold $T_{accept} = 90\,\%$, then the system triggers the expansion mechanism: a new autoencoder component is trained to recognise the regime, and a specialised expert is trained to close the RANS equations. The specific datasets and their role in the curriculum are detailed in table 2. It should be noted that each router (and expert model) is designed to handle a range of flow conditions over the same geometry. Thus for each flow regime, only some of the working conditions are used for the training, while the rest are left for model validation in § 4. To demonstrate the feasibility of the PMoE framework while maintaining minimal computational cost, the current training dataset is intentionally kept as small as possible. Further reducing the training set will inevitably degrade the model's generalisability. For each flow regime, the router and the corresponding expert are trained using the same subset of cases. The numerical set-up of RANS calculation for each case is summarised in Appendix B.

While completing the expansion of the router from PMoE-S0 to PMoE-S3, we also need to synchronously train the corresponding expert model for each newly identified flow regime. It should be particularly pointed out that the regimes are distinguished by the router automatically rather than pre-specified by humans during the training process.

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

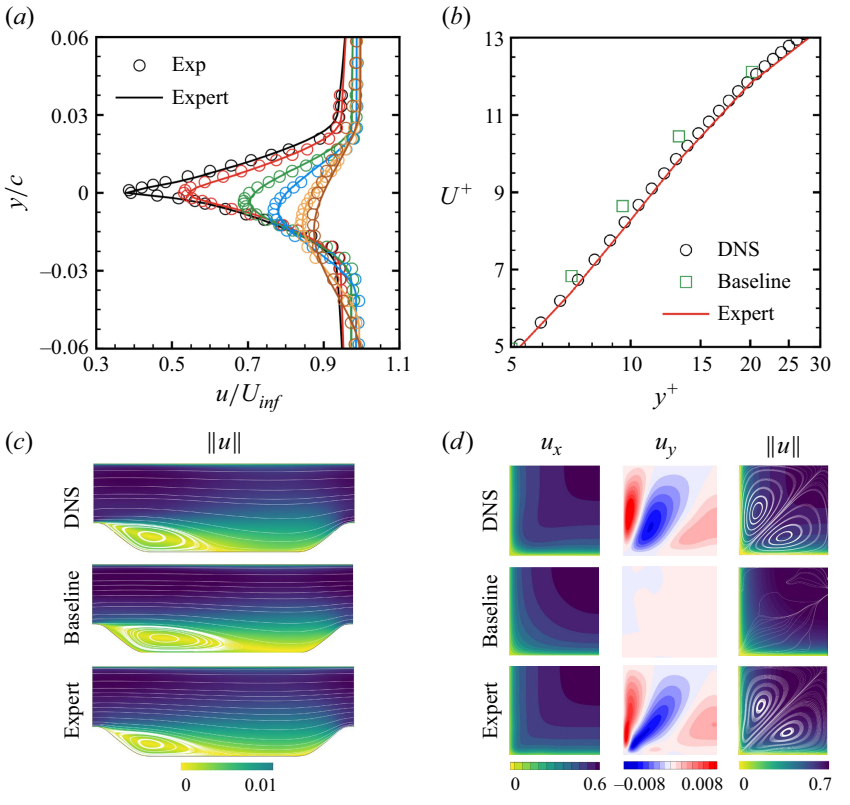


Figure 5. Velocity profiles and contours predicted by the baseline SA model, the expert model and high-fidelity data: (*a*) expert for S0 trained by 2DANW; (*b*) expert for S1 trained by C5200; (*c*) expert for S2 trained by PH1p0; (*d*) expert for S3 trained by SD2500 and SD5693.

3.1. *Stage 0: baseline initialisation*

The initial model, PMoE-S0, is established using the two-dimensional (2-D) aerofoil near wake (2DANW) case. This case is a verification of the aerofoil wake flow, with a uniform inflow, no-slip boundary conditions on the aerofoil surface, and a far-field free-stream condition. Detailed case configurations are referred to the NASA web page (data available online at https://tmbwg.github.io/turbmodels/airfoilwakeverif500c.html), and the grid used for the present RANS calculations is given in Appendix B. This regime represents simple free-shear turbulence where the standard SA model is known to perform adequately (Bin *et al.* 2023). Consequently, we designate the SA model as baseline expert E_0, and its performance in the 2DANW case is presented in figure 5(*a*). Furthermore, the first router component C_0 is trained on the wake features, establishing the baseline latent space distribution. As shown in table 3, the router identifies this regime with 99.9 % confidence, serving as the anchor for future expansions.

3.2. *Stage 1: adaptation to wall-bounded turbulence*

In the second phase, the model is exposed to fully developed turbulent channel flows at $Re_\tau = 2000$ and 5200, which are driven by a volumetric body force, with cyclic boundary conditions at the inlet and outlet and no-slip walls. The grid sizes used for the present RANS calculations are provided in Appendix B. When these data are fed into PMoE-S0,

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

	S0	S0 → S1	S1 → S2	S2 → S3
New regime	2DANW	C2000, C5200	PH1p0	SD2500, SD5693
MAX confidence with existing router	—	0.0 %	58.4 %	56.7 %
Introduced expert	SA baseline	Symbolic regression expression	FI-NN	FI-NN
Confidence of new router component	99.9 %	99.5 %	93.5 %	97.0 %

Table 3. Progressive construction of the PMoE model.

the router yields confidence 0.00 % as presented in table 3, correctly identifying that the physics of the wall turbulence differs fundamentally from the free-shear wake.

This triggers the creation of a new expert E_1. As discussed in § 2.2.3, we employ offline symbolic regression to derive a correction for the damping function f_{v1}. The resulting expert formulation is

$$f_{v1}^{*} = 0.985 \tanh\left(\frac{\chi^{1.3}}{31.1}\right). \tag{3.1}$$

Applying this expert to RANS calculations, the corrected model is able to provide more accurate wall-normal profiles compared to the baseline SA model, agreeing well with the direct numerical simulations (DNS) result (Lee & Moser 2015) as shown in figure 5(*b*). Simultaneously, a new router component C_1 is trained. The updated model, PMoE-S1, subsequently recognises the channel data with 99.5 % confidence, demonstrating its ability to capture wall-bounded flow characteristics (as represented here by channel flow), while retaining expert 0 for wakes.

3.3. *Stage 2: capturing flow separation*

Most industrial flows exhibit features such as strong pressure gradients, streamline curvature and separation, which violate the equilibrium assumption underlying conventional RANS models. The periodic hill flow is usually adopted as a canonical benchmark for separated turbulence, as it combines adverse pressure gradients, separation and reattachment in a simple, well-defined geometry, with the accurate prediction of the separation bubble remaining a key modelling challenge. The Reynolds number of the periodic hill cases is set as 5600, consistent with DNS simulation (Xiao *et al.* 2020), and the hill slope defined by $\alpha = l/h$ varies across cases, where l and h are the hill width and height, respectively.

The curriculum thus proceeds to the periodic hill case with $\alpha = 1.0$ as shown in table 2. The PMoE-S1 router produces maximum confidence 58.4 % as presented in table 3. Note that this relatively high level of confidence is likely due to shared wall-bounded features. Nevertheless, it falls below the 90 % threshold, flagging the regime as novel.

To address the non-equilibrium physics in the recirculation bubble, we train a new expert E_2 using the FIML method. This expert injects a source term $\beta(\boldsymbol{x})$ into the production term as presented in (2.12). The details are elaborated in Appendix C. As shown in figure 5(*c*), the trained separation flow expert can more accurately capture the phenomena of separation and reattachment, while the estimation of the separation bubble size by the basic SA model is significantly larger than that of the expert model and DNS results. Furthermore, upon integrating E_2 and its corresponding router component, the identification confidence for separated flows (as represented here by periodic hill flow) rises to 93.5 %, yielding model PMoE-S2.

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

3.4. *Stage 3: modelling corner-induced anisotropy*

Finally, the model encounters the square duct flow at $Re = 2500$, 5693. This case is driven by a volumetric body force, with cyclic inlet and outlet boundaries and no-slip walls on all duct surfaces. This case is physically distinct due to the secondary motions in the corner region, which are impossible for linear eddy viscosity models. The training data are obtained from published DNS simulations (Pinelli *et al.* 2010; Vinuesa, Schlatter & Nagib 2018). The Reynolds number is based on the edge half-length $h = D/2$ of the square duct and the bulk velocity U_b as shown in figure 4(*d*).

The PMoE-S2 router yields confidence 56.7 % as presented in table 3, again triggering expansion. A new expert E_3 is then trained to predict the nonlinear stress coefficient $C_{cr1}(\boldsymbol{x})$ in the QCR formulation in (2.13). The details are elaborated in Appendix C. As shown in figure 5(*d*), the trained expert model successfully captures the generation of secondary flow, while the baseline SA model is unable to calculate the secondary flow. The final model, PMoE-S3, achieves recognition confidence 97.0 % for this regime (as represented here by square duct flow).

4. Model validation

The validation of the PMoE framework focuses on three critical performance metrics: the interpretability of the unsupervised routing, the prevention of catastrophic forgetting on previously learned regimes, and the generalisation capability to cases with different operating conditions or configurations. All results presented herein utilise the final PMoE-S3 model to demonstrate its cumulative capabilities.

4.1. *Validation for the PMoE router*

A prerequisite for reliable expert selection is the router's ability to distinguish flow regimes based on local RANS features. We analyse the four-dimensional latent space of the autoencoder components using the Mahalanobis distance (Mahalanobis 1936) D_M, which measures the distance between a point z and a distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$:

$$D_M^2(z) = (z - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (z - \boldsymbol{\mu}). \tag{4.1}$$

Unlike the Euclidean distance, D_M accounts for the correlation structure of the learned representations. Based on this metric, we compute both the intra-cluster dispersion, which quantifies the compactness of samples within a regime, and the inter-cluster dispersion measuring the separation between regime centroids, as illustrated in figure 6. The ratio between the averaged inter- and intra-cluster dispersions is adopted as a diagnostic indicator of the latent-space organisation. A high ratio implies that different flow regimes are topologically distinct within the component's latent space, whereas a low ratio suggests possible overlap.

This analysis is carried out for all four components. Each component is exposed to data from all four flow regimes to assess its discriminative behaviour, and table 4 summarises these statistics. The intra-cluster dispersion remains consistent as ≈ 1.8 across all components, indicating that the autoencoders compress different flow regimes with comparable efficiency. Furthermore, the ratios between the averaged inter- and intra-cluster dispersions are also presented, and all four components exhibit ratios exceeding 3.4, indicating that the corresponding clusters have little overlap as illustrated by the well-separated configuration in figure 6. Although the router decision is ultimately based on the reconstruction error along the full encoder–decoder pathway, the latent-space analysis still provides an interpretable metric for its behaviour, as the observed

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

Dispersion	C_0 Wake	C_1 Channel	C_2 Periodic hill	C_3 Square duct
Average intra-cluster	1.8207	1.8383	1.8438	1.6947
Average inter-cluster	6.6845	6.3698	7.6183	7.8923
Inter-cluster/intra-cluster	3.6713	3.4650	4.1319	4.6569

Table 4. Dispersion analysis of the latent space clusters using Mahalanobis distance.

Case	Unknown (%)	C_0 Wake (%)	C_1 Channel (%)	C_2 Periodic hill (%)	C_3 Square duct (%)
ANW	0.0	**99.7**	0.3	0.0	0.0
C2000	0.2	0.0	**98.1**	0.5	1.2
C5200	0.0	0.0	**99.4**	0.0	0.6
C8000	3.9	0.0	**94.5**	1.3	0.3
PH0p8	0.3	0.0	5.7	**91.6**	2.5
PH1p0	0.1	0.0	5.4	**91.5**	3.0
PH1p2	0.1	0.0	4.8	**92.0**	3.2
PH1p5	0.2	0.0	4.7	**91.9**	3.2
SD2500	0.4	0.0	0.3	0.0	**99.3**
SD3500	0.5	0.0	0.0	0.2	**99.3**
SD5693	0.0	0.0	2.2	0.8	**97.0**

Table 5. Confidence distribution of the PMoE-S3 router over different components for all flow cases. Boldface in the table highlights confidence values that exceed the threshold T_{accept} of 90 %.

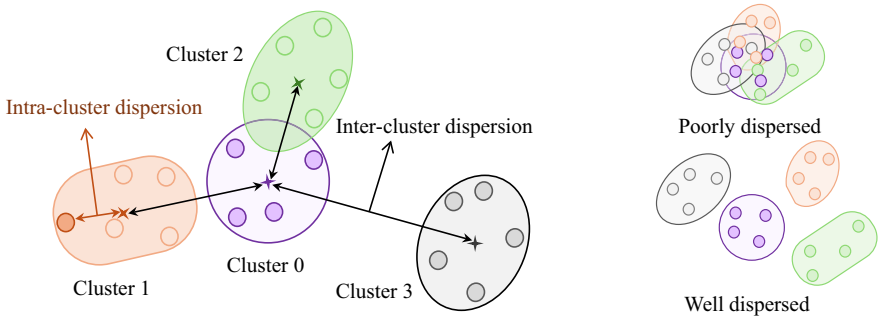


Figure 6. Schematic illustration of intra- and inter-cluster dispersion based on the Mahalanobis distance.

clustering patterns reflect the feature representations that govern reconstruction quality and hence routing outcomes. This mechanism, combined with the competitive winner-takes-all strategy, ensures robust classification. As evidenced in table 5, the router achieves >90 % classification accuracy on the validation sets, confirming that the present router successfully distinguishes the tested regimes.

4.2. *Prevention of catastrophic forgetting via in-distribution tests*

A primary failure mode in continual learning is catastrophic forgetting, where optimising for new tasks degrades performance on previous ones (French 1999). To assess whether the current PMoE framework is able to avoid catastrophic forgetting, we apply the final PMoE-S3 model back to the initial training cases. The corresponding results are presented in figure 7. The number shown in the top-left corner of each panel denotes the relative root

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

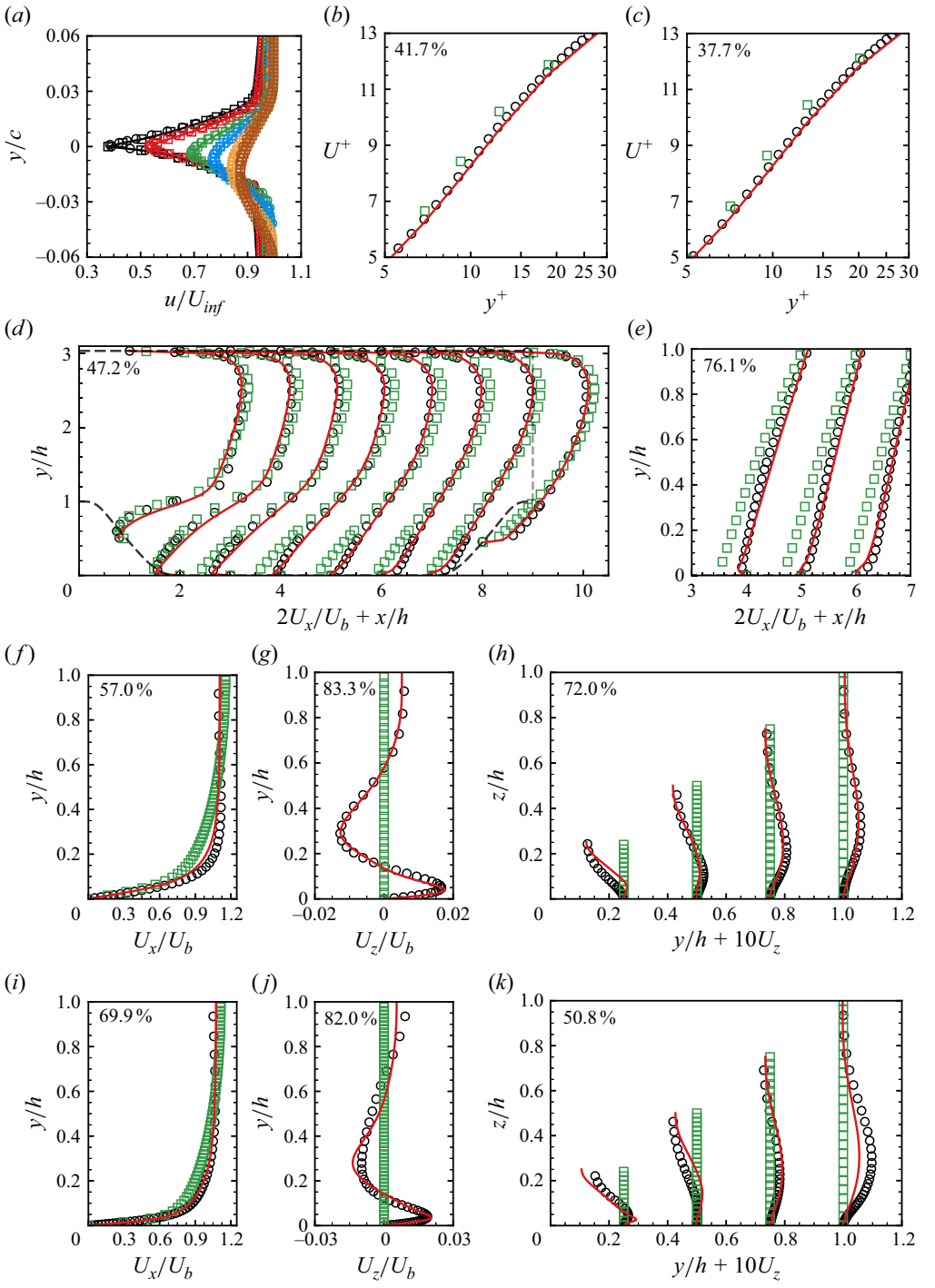


Figure 7. Mean velocity profiles for trained cases: (*a*) 2DANW (colours indicate streamwise locations), (*b*) C2000, (*c*) C5200, (*d*,*e*) PH1p0, (*f*–*h*) SD2500, (*i*–*k*) SD5693. Circles denote DNS/experimental data; green squares denote baseline SA; solid red lines denote PMoE-S3. The number shown in the top left corner of each panel is the relative error defined by (4.2).

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

Case	DNS	Baseline	PMoE-S3
PH0p8	5.21	7.34	7.19
PH1p0	5.04	7.31	5.87
PH1p2	4.50	7.88	5.36
PH1p5	4.21	8.32	4.71

Table 6. Reattachment locations of periodic hill cases.

mean square error (RMSE) reduction for the PMoE model $RMSE_{PMoE}$ with respect to the baseline $RMSE_{base}$, defined as

$$\mathcal{R} = (RMSE_{base} - RMSE_{PMoE})/RMSE_{base}, \tag{4.2}$$

where $RMSE = \sqrt{(1/N)\sum_{i=1}^{N}(u_i - u_i^{ref})^2}$. Here, u_i denotes the model prediction (baseline SA or PMoE, respectively) at the ith reference data location, the error is evaluated for the velocity component shown in each panel, and u_i^{ref}, $i = 1, \ldots, N$, are the corresponding reference values, with model predictions interpolated onto these locations when necessary. No value is shown in in figure 7(a) as expert E_0 coincides with the baseline.

Figure 7(a) presents the wake velocity profiles for the 2DANW case (stage 0), with each colour representing a different observation position: black for $x/c = 1.01$, red for $x/c = 1.05$, green for $x/c = 1.20$, blue for $x/c = 1.40$, yellow for $x/c = 1.80$, and brown for $x/c = 2.19$. Despite the model undergoing three subsequent expansions involving wall-bounded and secondary flows, the prediction for the free-shear wake remains virtually identical to the baseline SA model, and agrees well with experimental data. This suggests that in this near-wake region, the flow is dominated by the trailing-edge shear layer and vortex shedding, which are largely insensitive to wall effects. This confirms that the modular router successfully shields the baseline expert E_0 from interference.

Similarly, figures 7(b) and 7(c) show the mean velocity profiles extracted from the channel flow cases at $Re_\tau = 2000$ and $Re_\tau = 5200$ (stage 1), while figures 7(d) and 7(e) present the periodic hill at $\alpha = 1.0$ (stage 2). The final PMoE-S3 model retains the specialised accuracy of experts E_1 and E_2 in according tests. Particularly in the periodic hill case, the reattachment locations are quantified in table 6. For the training PH1p0 case, the baseline model predicts delayed reattachments near $x/h = 7.31$, whereas the PMoE model provides more accurate predictions for the separation bubble size with reattachment point $x/h = 5.87$. This improvement reflects the model's ability to capture the enhanced turbulence production and shear-layer dynamics responsible for earlier flow reattachment.

The capability of the final model PMoE-S3 for a trained square duct ($Re = 2500$) is shown in figures 7(f–h). Figures 7(f) and 7(g) show the velocity profiles at $z/h = 0.3$, $h = D/2$, while figure 7(h) represents $y/h = 0.25, 0.5, 0.75, 1.0$ from left to right just as the orange arrows shown in figure 4(d). The square duct case contains typical corner-induced secondary flow, which cannot be characterised by the SA model based on linear eddy viscosity. The results of PMoE-S3, however, have a significant improvement compared to the baseline model. This suggests that the trained PMoE-S3 model can effectively represent the secondary flow in the non-flow direction due to the introduction of nonlinear correction based on QCR. Moreover, figures 7(i–k) present the results of the SD5693 case. The performance of PMoE-S3 remains excellent, with significant improvements in both flow velocity and secondary flow representation compared to the baseline.

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

4.3. *Generalisation to cases with different operating conditions*

The robustness of the PMoE-S3 model is further tested against cases with different operating conditions never seen during training. The mean velocity profiles of all the cases are shown in figure 8. Compared with the cases in § 4.2, these new tests evaluate the model's generalisation ability to different Reynolds numbers and periodic hill geometries.

The model is applied to a channel flow at $Re_\tau = 8000$ and a square duct at $Re = 3500$. In both cases, the Reynolds numbers are different from the specific training conditions. The router successfully maps these flows to experts E_1 and E_3, respectively, as shown in table 5. The velocity profiles show excellent agreement with DNS data, indicating that the learned corrections, including the damping function modification in E_1 and the QCR coefficient field in E_3, capture the underlying physics rather than merely overfitting the training data.

We further test the periodic hill flows with varying slopes $\alpha = 0.8, 1.2, 1.5$. These geometric changes significantly alter the adverse pressure gradient and the separation bubble size. As shown in figures 8(*b*), 8(*d*) and 8(*f*), the PMoE-S3 model consistently outperforms the baseline, despite the model being trained only for the $\alpha = 1.0$ case. Similar to the conclusion drawn from § 4.2, the baseline SA model overestimates the size of the separation bubble in the lower hill area, while the PMoE-S3 model shows much better agreement with the result of DNS as shown in table 6. This suggests that the FIML-trained expert E_2 has learned a generalised correction for separation control that is robust to geometric deformation.

Up to now, the PMoE-S3 model has shown satisfactory predictive accuracy in a series of testing cases with different operating conditions, successfully overcoming catastrophic forgetting. These results indicate that although each expert model is trained for one or two operating conditions, it can actually handle a relatively wide range of operating conditions within the same geometry. For extreme operating conditions that exceed the generalisation boundaries of the expert models, however, the PMoE framework can be extended by fine-tuning only the relevant expert and its router, a far more efficient approach than retraining a monolithic model, highlighting its practical advantage.

4.4. *Computational efficiency*

A key practical consideration for the PMoE framework is its computational cost during inference. The complete inference pipeline consists of three stages: (i) a baseline SA simulation, from which input features are extracted; (ii) autoencoder-based routing; and (iii) a corrected RANS simulation using the activated expert. In many industrial workflows, a baseline RANS solution is already available as part of the standard design or analysis process. In such cases, only stages (ii) and (iii) constitute additional cost. When the router assigns the case to expert E_0 (the standard SA model), the baseline solution is itself the final prediction, and no additional simulation is required.

To quantify the computational overhead, we performed wall-clock timing tests using both the baseline SA model and the PMoE-S3 model on identical hardware. (All tests were conducted on a system equipped with an Intel® Xeon® Gold 6530 CPU and an NVIDIA® AD102 (GeForce RTX 4090) GPU.) The results are summarised in table 7. Across all tested cases, the routing time remains below 10 s, confirming that the autoencoder-based router is sufficiently lightweight for its overhead to be negligible. The full-pipeline overhead, which accounts for the baseline SA run, ranges from approximately 100 % to 104 %, reflecting the cost of effectively running RANS twice. This represents the worst-case scenario in which no prior baseline solution exists.

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

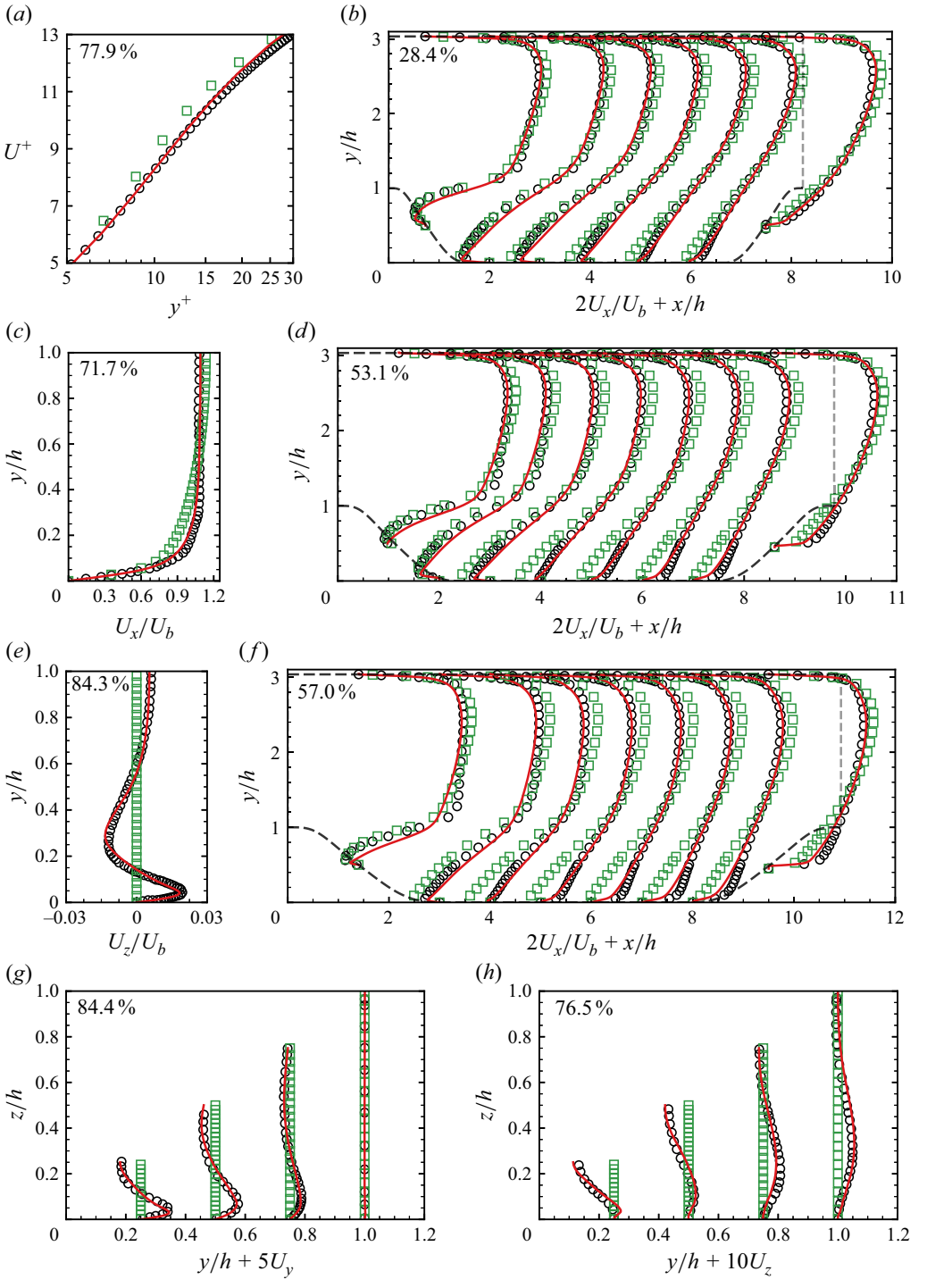


Figure 8. Mean velocity profiles for unseen cases with different operating conditions: (*a*) C8000, (*b*) PH0p8, (*d*) PH1p2, (*f*) PH1p5, and (*c*,*e*,*g*,*h*) SD3500. Circles denote DNS/experimental data; green squares denote baseline SA; solid red lines denote PMoE-S3. The number shown in the top left corner of each panel is the relative error defined by (4.2).

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

Case	T_{base} baseline simulation (s)	T_{route} routing (s)	T_{exp} expert simulation (s)	T_{tot} total (s)	$(T_{tot} - T_{base})/T_{base}$ increase ratio (%)
ANW	1943	9.82	–	1952.82	0.51
C5200	425	8.01	434	867.01	104.00
PH1p5	2810	6.80	2884	5700.80	102.88
SD5693	2077	9.29	2124	4210.29	102.71

Table 7. Comparison of computational wall-clock time between the baseline SA and PMoE-S3 models. Here, $T_{tot} = T_{base} + T_{route} + T_{exp}$.

Importantly, owing to the sparse activation mechanism, model expansion through continual learning does not lead to a proportional increase in inference cost. Regardless of the total number of experts in the PMoE model, only one expert is activated per case, ensuring that the per-case simulation cost scales with the size of a single expert rather than the entire ensemble.

5. Discussion

The results in § 4 have established that the PMoE framework achieves accurate predictions across canonical flows with different operating conditions while effectively avoiding catastrophic forgetting through its modular design. Two key questions arise when considering practical deployment: (i) how the model generalises to flow configurations that differ from the training set not only in operating conditions but also in geometry and boundary condition type; and (ii) how the current routing strategy performs when multiple flow regimes coexist within a single computational domain, and what alternative strategies may be considered. To address these questions, § 5.1 presents additional test cases with geometrically distinct configurations, and § 5.2 investigates routing strategies for multi-regime flows using a representative three-dimensional (3-D) configuration.

5.1. *Generalisation to geometrically distinct configurations*

The PMoE framework's generalisation capability arises from two complementary mechanisms: the router, which identifies the most relevant expert for a given flow based on learned feature-space representations, and the intrinsic generalisation of the expert models themselves, which encode physically meaningful corrections rather than case-specific data fitting. To probe the limits of both mechanisms, we introduce three additional test cases spanning distinct geometrical configurations, as shown in figure 9, including the 2-D NASA wall-mounted hump separated flow (2DWMH), the curved backward-facing step at $Re = 13\,700$ (CBFS), and the rectangular duct flow at $Re = 2500$ with aspect ratio 3 (RD2500). These cases are deliberately chosen to test the framework on geometries and boundary conditions that were not encountered during training, while retaining physical mechanisms, i.e. separation under adverse pressure gradients and corner-induced secondary flow, that overlap to varying degrees with the trained regimes. Notably, the 2DWMH case also incorporates an upstream zero-pressure-gradient boundary layer developing beneath a free stream, thereby simultaneously testing the coupling between free-stream, wall-attached and separated flow regions within a single configuration.

5.1.1. *Router discrimination on unseen geometries*

Table 8 summarises the confidence distribution of the PMoE-S3 router for the three additional cases. The confidence levels span a considerable range, from a maximum of

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

Case	Unknown (%)	C_0 Wake (%)	C_1 Channel (%)	C_2 Periodic hill (%)	C_3 Square duct (%)
2DWMH	10.0	0.0	35.2	**54.7**	0.0
CBFS	0.9	0.0	22.4	**76.7**	0.0
RD2500	0.0	0.0	1.8	7.4	**90.8**

Table 8. Confidence distribution of the PMoE-S3 router over different components for additional flow cases. Boldface in table indicates the highest router confidence of all components.

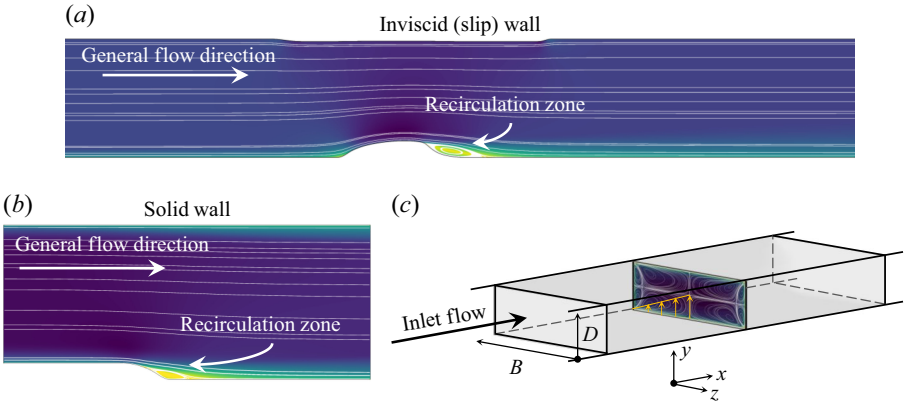


Figure 9. Schematic diagrams of additional test cases with different configurations: (*a*) 2DWMH, (*b*) CBFS, (*c*) RD2500.

90.8 % for RD2500 to 54.7 % for the most challenging case (2DWMH). These differences can be understood by examining how the geometric and physical characteristics of each new case align with the feature signatures learned during training, as characterised by the PFI analysis in Appendix A.

The RD2500 rectangular duct preserves the most similar features of the training square duct flow. It is a fully developed, pressure-driven wall-bounded flow with no-slip conditions on all walls and corner-induced secondary motions generated by Reynolds stress anisotropy. The only systematic deviation is the aspect ratio, which changes from 1 to 3, modifying the relative strength and spatial extent of the corner vortices. As a result, the flow retains the same wall-driven shear representation, and the router assigns the highest confidence (90.8 %) to the square duct class C_3.

The CBFS case shares the dominant physics of the periodic hill training case, an adverse-pressure-gradient-driving shear-layer separation followed by reattachment, but introduces a sharp-to-curved geometric expansion that alters the pressure-gradient distribution and shear-layer inception. Although the geometric expansion differs, the resulting flow structure remains governed by pressure–shear interaction. Consequently, the feature representation remains close to that of the periodic hill, and the router identifies class C_2 with confidence 76.7 %, reflecting a substantial overlap in the underlying flow dynamics.

The 2DWMH case likewise involves separation driven by an adverse pressure gradient, yet the separation occurs over a convex hump surface in an external-flow setting rather than within a confined channel, introducing greater geometric and boundary condition dissimilarity relative to the periodic hill. In particular, the presence of an upstream attached boundary layer and the absence of confinement lead to a more mixed flow character. This reduces the similarity in the encoded representation, resulting in a lower router confidence of 54.7 % for C_2, with a secondary attribution of 35.2 % to the channel class C_1.

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

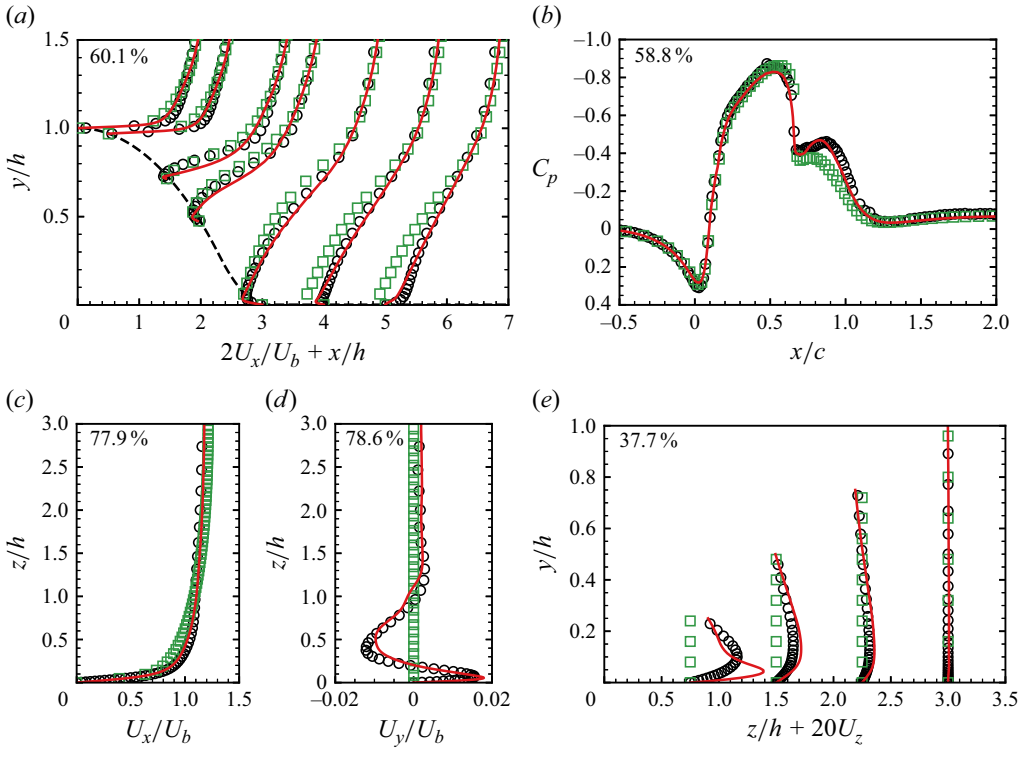


Figure 10. Predictions obtained with the highest-confidence experts for additional test cases: (*a*) CBFS, (*b*) 2DWMH, (*c–e*) RD2500. Circles denote DNS/experimental data; green squares denote baseline SA; solid red lines denote expert model. The number shown in the top left corner of each panel is the relative error defined by (4.2).

The pattern emerging from table 8 is physically intuitive. Flows that retain stronger similarity to the training regimes yield higher router confidence, whereas configurations with more pronounced geometric and boundary condition deviations exhibit lower and more distributed confidence.

5.1.2. *Expert prediction*

Beyond the router's discriminative capability, the expert models themselves exhibit strong generalisation when activated on the basis of the highest confidence assignment. For the CBFS case, the periodic hill expert E_2 substantially reduces the overprediction of the separation bubble extent produced by the baseline SA model, yielding corrected velocity profiles that faithfully capture the interaction between the adverse pressure gradient and shear-layer growth, and accurately locate the reattachment point as shown in figure 10(*a*). The reference data are obtained from the LES of Bentaleb, Lardeau & Leschziner (2012). The underlying correction, a recalibration of the non-equilibrium balance between turbulence production and dissipation under adverse pressure gradients, transfers effectively to this geometrically related but distinct separated-flow configuration.

The 2DWMH case provides an even more stringent test: although the router confidence is appreciably lower, activating E_2 still yields improved pressure coefficient (figure 10*b*) distributions, with particular gains in the separated region and the downstream recovery zone where the baseline model fails to predict the correct separation onset and pressure recovery. The reference data are obtained from the experiment of Naughton, Viken &

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

Greenblatt (2006) and Greenblatt *et al.* (2006*a*,*b*). This result demonstrates that the FIML-trained expert encodes transferable corrections for the fundamental mechanisms governing turbulent separation. Specifically, the suppression of excessive turbulence production in adverse-pressure-gradient regions that remain effective even when the external geometry and boundary layer state differ substantially from those of the training case.

Finally, for the RD2500 case, the square duct expert E_3, which predicts a spatially varying nonlinear stress coefficient $C_{cr1}(\boldsymbol{x})$, accurately reproduces the corner-induced secondary vortices driven by anisotropic Reynolds stresses and cross-stream pressure gradients, and improves the streamwise velocity profile relative to the baseline SA prediction, despite the change in aspect ratio. This confirms that the expert has encoded the essential physics of corner-induced anisotropy rather than a geometry-specific correction. The reference data are obtained from the DNS of Vinuesa *et al.* (2018). The mean velocity profiles shown in figures 10(*c*) and 10 (*d*) are taken at $y/h = 0.5$, $h = D/2$, where D denotes the duct height, while those in figure 10(*e*) are extracted along the lines indicated in figure 9(*c*).

5.1.3. *Synthesis*

Collectively, these results demonstrate that the generalisation capability of the PMoE framework arises from the combined action of its two core components. The router provides a physically interpretable confidence measure that quantifies the relevance of each trained expert to an unseen flow, and serves as an indicator of similarity in the learned feature space. The expert models encode physically meaningful corrections rather than case-specific data fitting, and these corrections remain effective across variations in geometry and boundary conditions as long as the dominant physical mechanism is preserved. Even when the router confidence does not reach the acceptance threshold 90 %, the expert with the highest confidence can still provide noticeable improvement over the baseline model, as demonstrated by the 2DWMH case. Conversely, when no expert is sufficiently relevant, the acceptance threshold T_{accept} ensures that the framework defaults to the unmodified baseline model, avoiding the introduction of spurious corrections. This combination of a discriminative router and robust physics-based experts provides a degree of generalisation and fault tolerance that is not typically achieved by monolithic data-driven models. It also offers a practical strategy for extension. When the limit of an expert is reached for a substantially different flow configuration, only the relevant expert and its associated router component need to be refined, rather than retraining the entire model.

5.2. *Routing strategies for more complex flows*

5.2.1. *Assumptions and limitations of the current routing strategy*

The present PMoE router classifies flow types based on the baseline SA model's predictions. The effectiveness generally rests on several key assumptions. Specifically, the flow under consideration can be adequately characterised by the features listed in table 1; and the baseline SA model provides a qualitatively correct flow field, from which meaningful features can be extracted. As demonstrated in the preceding cases, even when the SA model exhibits deviations, such as overestimating the separation bubble in the periodic hill case or failing to capture the secondary flows in the square duct, these inaccuracies have only a limited effect on the routing outcomes. This robustness stems from the design of the router, which considers a comprehensive set of normalised invariant features that characterise the local flow.

Conversely, the framework may become unreliable when these assumptions are violated. This can arise if the baseline model produces a qualitatively incorrect flow topology,

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

such as failing to capture separation or transition, thereby corrupting the extracted features.

Furthermore, the results presented in §§ 4 and 5.1 employ a top-1 (winner-takes-all) routing strategy, in which the expert with the highest router confidence is exclusively activated for the entire computational domain. This approach is effective when the flow field is dominated by a single physical regime, as demonstrated for the canonical test cases. For flows closely resembling the trained cases (e.g. RD2500), the corresponding expert can be directly activated for optimal corrections. For moderately similar flows (e.g. CBFS and 2DWMH), the router's top recommendation provides effective corrections even at reduced confidence. For unknown cases identified by the PMoE router, the baseline SA model will be invoked, ensuring physically consistent results. Note that the top-1 routing is similar to industrial strategies such as GEKO (Menter & Matyushenko 2025). Applying a single expert globally ensures numerical stability. While a single expert is applied globally, the corrections that it produces are inherently locally determined by the pointwise input features, so the expert remains quiescent in regions where the local flow state does not trigger its correction (e.g. $\beta(\boldsymbol{x}) \approx 1$ in equilibrium regions for expert E_2).

For complex industrial configurations, when multiple flow regimes coexist with comparable importance within a single domain, a single top-1 routing decision may no longer be sufficient to represent the flow physics. In such situations, additional reference data or more advanced routing strategies may be required to ensure reliable performance.

5.2.2. *Multi-regime test: the 3-D diffuser*

To examine the behaviour of the PMoE framework under conditions of genuinely coexisting flow regimes, we consider the 3-D diffuser case (Cherry, Elkins & Eaton 2008). As shown in figure 11(*a*), the geometry consists of a straight square duct inlet section that transitions into a rectangular expansion section, where one wall diverges at a prescribed angle while the opposite wall remains flat. The boundary conditions include a fully developed duct inflow, no-slip conditions on all walls, and a zero-gradient outflow at the diffuser exit. The upstream section features fully developed square duct flow with corner-induced secondary motions, while the expansion region introduces an adverse pressure gradient, streamline curvature, and ultimately 3-D flow separation from the diverging wall. The interplay between the corner vortices carried from the inlet section and the incipient separation renders this flow particularly challenging for RANS turbulence models (Abe & Ohtsuka 2010).

This configuration is obviously beyond the scope of the current top-1 strategy. The router identifies three competing regimes with comparable confidence: the periodic hill class C_2 at 40.7 %, the channel class C_1 at 36.5 %, and the square duct class C_3 at 21.5 %. No single component achieves a dominant confidence, reflecting the genuinely multi-physics nature of the case.

As a preliminary exploration of strategies beyond the top-1 paradigm, we conducted a test in which all three major expert models (E_1, E_2 and E_3) were simultaneously activated across the entire 3-D diffuser domain. This combined application exploits the inherently local nature of the expert corrections. Specifically, E_1 primarily acts in near-wall regions through the damping function modification, E_2 adjusts the production term via $\beta(\boldsymbol{x})$ in zones of strong adverse pressure gradients and recirculation, and E_3 introduces nonlinear stress corrections through $C_{crl}(\boldsymbol{x})$ in corner regions. In flow regions that do not exhibit strong similarity to the training cases, i.e. where the local features do not trigger significant

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

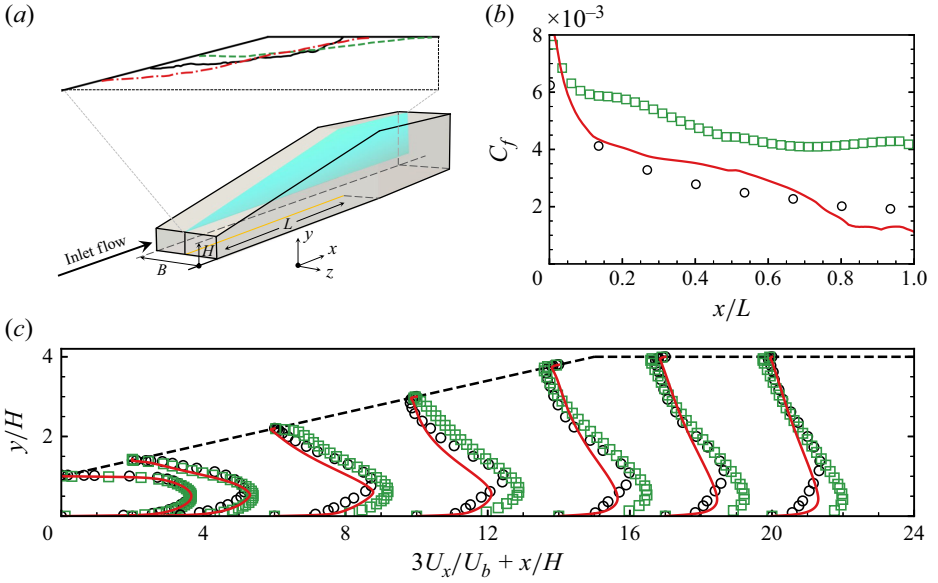


Figure 11. Performance evaluation of the mixed model for the 3-D diffuser case. (*a*) Zero streamwise-velocity contours on the x–y plane at $z = B/2$ (midplane), where black, green, and red denote DNS, baseline SA, and the PMoE experts, respectively. (*b*) Skin-friction coefficient (C_f) distribution along the bottom wall centreline ($z = B/2$), extracted along the yellow line (length L) indicated in (*a*). (*c*) Mean velocity profiles on the midplane. Circles denote DNS/experimental data; green squares denote baseline SA; solid red lines denote expert model.

corrections, the predicted corrections remain small, and the model behaviour effectively reverts to that of the baseline SA model. As shown in figure 11(*b*), this simple combined strategy already yields encouraging results. In particular, the skin-friction coefficient C_f along the bottom wall shows clear improvement compared to the baseline SA prediction, and is notably closer to the DNS data of Ohlsson *et al.* (2010). This demonstrates that the modular expert design, while developed for single-regime flows, has the potential to be meaningfully extended to multi-regime configurations.

It should be noted, however, that the 3-D diffuser involves a substantially higher degree of geometric and physical complexity than any individual training case, including 3-D corner-driven separation interacting with pressure-induced detachment. Consistently, the results reveal mixed performance: while the adverse-pressure-gradient effects near the bottom wall are better captured, as reflected by both the improved C_f distribution and the near-wall velocity profiles (figure 11*c*). In contrast, the velocity profiles in the core region show larger deviations compared to the baseline SA prediction, indicating that the concurrent activation of multiple experts does not uniformly improve the solution throughout the flow field. Furthermore, the separation extent inferred from the $u = 0$ contour indicates that directly activating multiple experts shifts the separation location from the baseline's overly downstream prediction to an overcorrected upstream position. The current expert library, designed and trained on canonical 2-D and simple 3-D configurations, is not expected to fully capture such interactions. The improvement observed here should therefore be viewed as a demonstration of the framework's extensibility and of the transferability of the expert corrections, rather than as a claim that the current model is sufficient for this class of flows.

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

5.2.3. *Toward region-aware routing*

The 3-D diffuser results motivate several directions for extending the current routing strategy. A natural next step is region-wise application, in which the computational domain is partitioned into subregions and each subregion is assigned to the expert whose local confidence is highest. Conceptually, this approach exploits the spatial distribution of the router's reconstruction error to create a patchwork of expert-corrected zones. This strategy has the potential to improve predictions in complex flows by locally matching the correction to the prevailing physics. However, it introduces practical challenges, including the treatment of interfaces between expert regions, where discontinuities in the correction field may arise and compromise solver convergence, and the sensitivity of the partitioning to the threshold and sampling density.

An alternative and potentially more robust approach is a top-k ($k > 1$) gating strategy with soft blending, in which the corrections from the k most confident experts are combined using continuous weights derived from the reconstruction-error-based confidence p_k. Because p_k varies spatially through its dependence on local features, such a strategy would enable smooth transitions between regimes without the hard boundaries inherent in region-wise partitioning. The modular autoencoder architecture of the PMoE router is inherently compatible with such extensions, as each component can independently evaluate reconstruction error at the point level.

It is important to note, however, that the current heterogeneous expert formulations, which range from symbolic expressions modifying f_{v1} to neural-network-based corrections of different target quantities (β for production, C_{cr1} for the constitutive relation), preclude straightforward soft blending, since the expert outputs are not directly commensurable. A future implementation targeting smooth region-wise routing would benefit from a unified expert format, e.g. all experts predicting the same correction type, such as a Reynolds stress anisotropy tensor or a common multiplicative field. This represents a fundamental design trade-off. The heterogeneous formulations adopted in the present work maximise physical fidelity by allowing each expert to employ the correction form best suited to its regime, whereas homogeneous formulations would sacrifice some of this flexibility in exchange for the numerical smoothness required for blending. Resolving this trade-off, and developing the associated interface-treatment and blending algorithms, will be a key focus of future work. We anticipate that such developments will enable improved performance not only for the 3-D diffuser case but also for other complex 3-D industrial flows in which multiple physical mechanisms interact.

6. Conclusions

In this work, we propose a progressive modular mixture-of-experts framework, termed PMoE, to address generalisation and continual learning in data-driven RANS turbulence modelling. The framework is based on a successively trainable autoencoder-based similarity router that enables the progressive introduction of specialised turbulence model experts as new flow regimes emerge. Using training data covering four representative regimes, namely aerofoil wakes, channel flows, periodic hill flows and square duct flows, we construct the PMoE-S3 model and demonstrate its ability to deliver accurate predictions for both seen and unseen test cases.

A key advantage of the proposed framework lies in its support for continual learning without catastrophic forgetting. New flow regimes can be incorporated by adding dedicated experts and router components, while previously trained components remain unchanged, preserving established predictive capabilities. Moreover, the modular design of PMoE allows experts of fundamentally different forms to coexist within a single

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

framework, offering the flexibility to integrate turbulence models based on different formulations and assumptions as needed. Importantly, the training procedure relies on unsupervised feature extraction through the autoencoder-based router, making the framework particularly well suited to industrial workflows where labelled data and repeated full retraining are often impractical. Beyond its demonstrated performance, PMoE provides a scalable and extensible platform for RANS modelling. Owing to the sparse activation mechanism, model expansion does not lead to a proportional increase in computational cost during inference, which is essential for industrial CFD applications. Its plug-and-play structure enables modular updates and reduces both development and deployment costs, while its extensibility allows the turbulence modelling community to contribute new experts targeting previously unexplored flow regimes, establishing a living framework that can evolve alongside advances in modelling strategies.

Building on the modular and extensible nature of the framework, future developments may incorporate experts based on alternative turbulence closures, such as the k–ω shear stress transport model (Menter 1994) or Reynolds stress models (Launder, Reece & Rodi 1975), to broaden its applicability. For more complex flows, region-aware routing is expected to become increasingly important for applying different experts within different spatial zones. Although the current set of input features is sufficient for distinguishing the flow regimes examined here, further enrichment of the feature representation may be required as additional flow types are considered.

Funding. This work has been supported by the National Natural Science Foundation of China (grant nos 12588201, 12572247 and 12432010).

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

Appendix A. Feature importance analysis

Feature importance analysis is commonly employed to improve the interpretability of neural networks by quantifying the contribution of each input feature to the model prediction (Breiman 2001). In this work, to explore whether the importance distribution of different features varies in various flow regimes, and thereby demonstrate the feasibility of identifying flow types based on feature importance, it is necessary to conduct a feature importance analysis.

Among such approaches, PFI provides a model-agnostic assessment of the sensitivity of a fitted model to individual features using a prescribed tabular dataset (Fisher, Rudin & Dominici 2019; Mandler & Weigand 2023). It is achieved by shuffling values of a single feature and subsequently assessing the consequent deterioration in the model performance. This approach can determine the extent to which the model depends on a specific feature by disrupting the intrinsic connection between the input features and the model prediction. The procedure of the PFI method is illustrated in figure 12. The inputs include a neural network used for reconstruction, an input feature matrix $\boldsymbol{X} \in \mathbb{R}^{N\times M}$ comprising M features and N sampling points, and the corresponding targets, which coincide with $\boldsymbol{X}$ for this reconstruction task. For each feature q_i, a permuted matrix $\boldsymbol{X}_i^S$ is generated by shuffling the ith column of $\boldsymbol{X}$, and the reconstruction error e_i^S is recomputed as $\|\boldsymbol{X}_{Ri}^S - \boldsymbol{X}\|$. The importance associated with each feature q_i is defined as Ψ_i, the change in error induced by permutation, providing a quantitative measure of the model's sensitivity to that feature. All features are analysed in this manner, and the ranking is obtained from repeated trials with averaged results to minimise statistical variability.

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

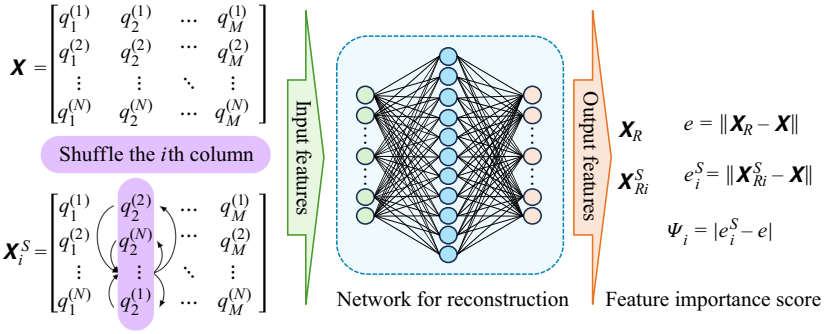


Figure 12. Schematic illustration of the formulation and computational procedure of the PFI.

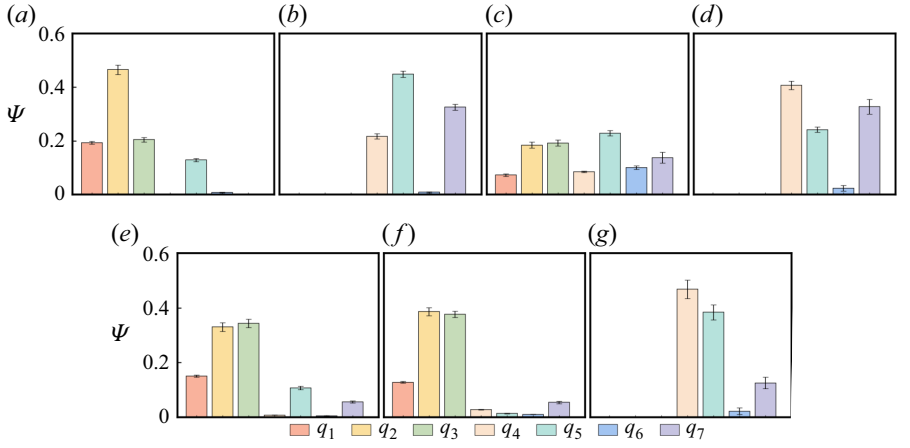


Figure 13. Normalised feature importance scores obtained from the PFI for the the training data and additional test cases: (*a*) wake, (*b*) channel, (*c*) periodic hill, (*d*) square duct, (*e*) 2DWMH, (*f*) CBFS, (*g*) RD2500.

To eliminate the interference caused by network capacity, we choose networks with sufficient parameters to test whether the four types of flowing data that we adopt have different orders of feature importance in the reconstruction task. To facilitate the comparison of relative feature importance both within and across heterogeneous datasets, the raw PFI scores were normalised by the sum of seven feature importance scores. In figure 13, the bar plots display the normalised mean PFI values, with error bars representing the standard deviation computed from 20 repeated permutation trials, indicating the stability of the importance estimates. Features with higher importance scores are considered more informative. The four flow regimes exhibit distinct importance profiles, indicating that different subsets of features govern the reconstruction. A more detailed inspection of the dominant feature combinations reveals that each canonical flow type admits a distinct low-dimensional physical representation in the SA-generated solution space. In the following, the four baseline flow categories are discussed separately. For the wake flows, the reconstruction is mainly governed by q_2, q_1 and q_3, indicating that the flow is encoded by shear–rotation dynamics and their interaction with the

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

pressure gradient, consistent with a free shear-layer structure. For the channel flow, the dominant features are q_5, q_4 and q_7, corresponding to eddy viscosity, wall distance and shear anisotropy. This indicates that the flow is fully characterised by wall-bounded turbulence scaling, with negligible influence from rotational or pressure-driven effects. For the periodic hill flow, the importance is distributed across q_5, q_2, q_3 and q_7, reflecting the coupled roles of turbulent transport, shear-layer instability, pressure gradient and recirculation. This confirms that separated flows require a multi-physics representation. For the square duct flow, the dominant features remain q_5, q_4 and q_7, indicating a wall-driven anisotropic shear system. The weak contribution of q_2 and q_3 suggests that secondary-flow-related rotational and pressure effects are not explicitly represented in the SA solution. These class-dependent importance patterns lead to substantially different parameter distributions when training the modular experts of the router network. This explains why the approach of using reconstruction errors as the identification basis can effectively distinguish different flow types of data, and still maintain a high accuracy rate after modular expansion.

For the CBFS case, the dominant contributions from q_2 and q_3, together with secondary roles of q_1 and q_7, place it within the same feature subspace as the periodic hill flow. Accordingly, the router identifies the periodic hill class C_2 with confidence 76.7 %, reflecting a strong overlap in the pressure–shear-dominated representation of separated flows. For the 2DWMH case, the PFI distribution remains dominated by q_2 and q_3, accompanied by non-negligible contributions from q_1 and q_5. This indicates that while the flow retains a separation mechanism governed by pressure–shear coupling similar to the periodic hill, the role of turbulent transport becomes more pronounced. At the same time, the presence of an upstream attached boundary layer introduces additional dependence on wall-related structures, leading to a more distributed feature representation. As a result, the router confidence for the periodic hill class C_2 decreases to 54.7 %, with a secondary attribution of 35.2 % to the channel class C_1, reflecting the mixed character of external separation and partial wall-bounded behaviour. For the rectangular duct (RD2500), the dominant features remain q_5, q_4 and q_7, closely matching the square duct flow. This consistency in feature space explains why the router assigns the highest confidence (90.8 %) to the square duct class C_3, indicating that the SA solution preserves the same wall-driven anisotropic shear representation.

Appendix B. Case settings

Numerical simulations in this paper are performed using the open-source computational framework OpenFOAM (open source field operation and manipulation, version 2012). The SA baseline model mentioned in this paper refers to the model and default parameter settings presented by Spalart & Allmaras (1992). Grids and computation domain are shown in table 9. The numerical set-ups exhibit minor variations across cases due to differences in geometry and flow conditions, but follow consistent practices based on standard OpenFOAM configurations for canonical flows. A representative set-up employs second-order spatial discretisation, with Gauss linear schemes for gradients and diffusion terms, and a linear upwind scheme for the convective term of velocity. Turbulence transport equations are discretised using bounded upwind schemes. Pressure–velocity coupling is handled using the SIMPLE algorithm with one non-orthogonal corrector. Standard under-relaxation factors are applied. Convergence is assessed based on residual reduction and stabilisation of key flow quantities.

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

Case	Box	Grid
ANW	$1000c \times 1000c$	$337 \times 449,\ 513 \times 225$
C2000	$4\pi h \times 2h$	250×100
C5200	$4\pi h \times 2h$	500×200
C8000	$4\pi h \times 2h$	800×320
PH0p8	$8.23h \times 3.036h$	99×149
PH1p0	$9.0h \times 3.036h$	99×149
PH1p2	$9.77h \times 3.036h$	99×149
PH1p5	$10.93h \times 3.036h$	99×149
SD2500	$2.5D \times D \times D$	$75 \times 48 \times 48$
SD3500	$2.5D \times D \times D$	$75 \times 48 \times 48$
SD5693	$2.5D \times D \times D$	$75 \times 48 \times 48$
2DWMH	$6c \times c$	137×424
CBFS	$22.7h \times 9.5h$	712×104
RD2500	$2.5D \times D \times 3D$	$75 \times 48 \times 144$

Table 9. Computational domains and grid resolutions for all cases considered in this study.

Appendix C. Training details

In this appendix, we provide the detailed training procedures for the three experts E_1, E_2 and E_3, which are used in the PMoE framework. These procedures include the generation of training data, the optimisation objectives, and the implementation details of the symbolic regression and neural network models. The description is organised sequentially from E_1 to E_3, highlighting the specific methodologies adopted for each expert.

C.1. *Expert E_1: symbolic regression of the damping function*

The damping function correction f_{v1} for E_1 is obtained via offline symbolic regression following the methodology of Weatheritt & Sandberg (2016). Candidate expressions are generated and evolved using gene expression programming (GEP) under predefined function and terminal sets. The function set includes the basic operators $\{+, -, \times, \div,$ exponentiation, square, tanh$\}$, while the terminal set comprises the variable $\chi = \tilde{\nu}/\nu$, the constant 1.0, and randomly generated constants sampled from the interval $[-5, 5)$. The GEP algorithm is initialised with a population of 1000 expressions, and evolved over 10 000 generations through mutation and recombination.

Each candidate expression is evaluated on the training data using the objective

$$\mathcal{J} = \frac{1}{N}\sum_{i=1}^{N}\left(f_{v1}^{pred}(\chi_i) - f_{v1}^{target}(\chi_i)\right)^2, \tag{C1}$$

where f_{v1}^{pred} denotes the symbolic prediction, and f_{v1}^{target} is the corresponding reference value. The training data are extracted from the corrected $f_{v1}(\chi)$ profiles reported by Bin *et al.* (2023), which were originally generated using a neural-network-based approach. These profiles capture the near-wall behaviour of the turbulence eddy viscosity, which is critical for accurately predicting the mean velocity distribution in wall-bounded flows. At the end of this offline regression stage, the optimal expression $f_{v1}(\chi)$ is selected according to the defined objective, and is subsequently hard-coded into the SA solver as E_1 for RANS simulations, improving near-wall predictions without additional tuning.

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

C.2. *Expert E_2: learning the correction of production term for flow separation*

Expert E_2 is trained to capture the non-equilibrium physics in the recirculation bubble using a two-step FIML procedure. First, an adjoint-based field inversion is performed on the periodic hill case PH1p0 using the DAFoam framework (He *et al.* 2018, 2020). The optimisation objective is to minimise the discrepancy between the RANS-predicted velocity field and DNS reference data (Xiao *et al.* 2020) by adjusting a spatially varying production-term multiplier $\beta(\boldsymbol{x})$, yielding an optimal β field on the computational grid.

Second, a neural network is trained to learn the mapping from the input features $\boldsymbol{x} = [q_1, \ldots, q_7]$ to the inverted β field. The dataset comprises grid-point-level pairs $\{(\boldsymbol{x}_j, \beta_j)\}_{j=1}^{N_{grid}}$ extracted from the PH1p0 case, and is randomly divided into 80 % training and 20 % testing subsets. Note that no extra sampling is applied at this stage. The network is implemented as a three-layer fully connected feed-forward network in PyTorch (Paszke *et al.* 2019). All input features are further normalised to the range [0, 1] using min–max scaling, which is a standard pre-processing step to improve numerical conditioning during neural network optimisation. The network has seven input nodes, two hidden layers with 64 and 32 neurons, and a single scalar output. Rectified linear unit (Glorot, Bordes & Bengio 2011) activation functions are applied between layers to introduce nonlinearity. The network is trained using the Adam optimiser with a cosine-annealed learning-rate schedule and an adaptive reduction strategy triggered by validation-loss plateaus to enhance convergence robustness. The mean squared error loss function is used during training. This compact architecture provides sufficient representational power for nonlinear feature interactions while limiting the risk of overfitting. The trained network subsequently injects the predicted $\beta(\boldsymbol{x})$ into the production term of the SA model, as defined in (2.12), for subsequent RANS simulations.

C.3. *Expert E_3: learning the constitutive relation modification for secondary flows*

Expert E_3 is trained following the same two-step FIML procedure as E_2, but applied to the square duct cases SD2500 and SD5693. First, an adjoint-based field inversion optimises the spatially varying nonlinear stress coefficient $C_{cr1}(\boldsymbol{x})$ in the QCR formulation (2.13), with the objective of minimising the discrepancy between the RANS-predicted velocity field – including both streamwise and secondary-flow components – and DNS reference data (Pinelli *et al.* 2010; Vinuesa *et al.* 2018).

Second, a neural network with the same architecture and training procedure as E_2 is employed to learn the mapping $\boldsymbol{x} \rightarrow C_{cr1}(\boldsymbol{x})$. For brevity, we do not repeat the details of the network architecture and training procedure, which are identical to those described for E_2 above. The trained network subsequently provides the $C_{cr1}(\boldsymbol{x})$ field for injection into the QCR-based RANS simulations.

Overall, the training of these three experts combines data-driven model extraction and physics-informed optimisation, enabling the PMoE framework to accurately capture both near-wall and non-equilibrium turbulent flow features across a variety of canonical cases.

References

Abe, K. & Ohtsuka, T. 2010 An investigation of LES and hybrid LES/RANS models for predicting 3-D diffuser flow. *Intl J. Heat Fluid Flow* **31** (5), 833–844.

Aljundi, R., Chakravarty, P. & Tuytelaars, T. 2017 Expert Gate: lifelong learning with a network of experts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3366–3375.

Amarloo, A., Rincón, M.J., Reclari, M. & Abkar, M. 2023 Progressive augmentation of turbulence models for flow separation by multi-case computational fluid dynamics driven surrogate optimization. *Phys. Fluids* **35** (12), 125154.

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

ARRANZ, G., LING, Y., COSTA, S., GOC, K. & LOZANO-DURÁN, A. 2024 Building-block-flow computational model for large-eddy simulation of external aerodynamic applications. *Commun. Engng* **3** (1), 127.

BENTALEB, Y., LARDEAU, S. & LESCHZINER, M.A. 2012 Large-eddy simulation of turbulent boundary layer separation from a rounded step. *J. Turbul.* **13**, N4.

BIN, Y., HUANG, G. & YANG, X.I.A. 2023 Data-enabled recalibration of the Spalart–Allmaras model. *AIAA J.* **61** (11), 4852–4863.

BREIMAN, L. 2001 Random forests. *Mach. Learn.* **45** (1), 5–32.

BRUNTON, S.L., NOACK, B.R. & KOUMOUTSAKOS, P. 2020 Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **52**, 477–508.

BUSH, R.H., CHYCZEWSKI, T.S., DURAISAMY, K., EISFELD, B., RUMSEY, C.L. & SMITH, B.R. 2019 Recommendations for future efforts in RANS modeling and simulation. In *AIAA Scitech 2019 Forum, AIAA Paper* 2019-0317. American Institute of Aeronautics and Astronautics.

CHERROUD, S., MERLE, X., CINNELLA, P. & GLOERFELT, X. 2025 Space-dependent aggregation of stochastic data-driven turbulence models. *J. Comput. Phys.* **527**, 113793.

CHERRY, E.M., ELKINS, C.J. & EATON, J.K. 2008 Geometric sensitivity of three-dimensional separated flows. *Intl J. Heat Fluid Flow* **29** (3), 803–811.

DAI, D. *et al.* 2024 DeepSeekMoE: towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, vol 1. Long Papers, pp. 1280–1297. Association for Computational Linguistics.

DU, N. *et al.* 2022 GLaM: efficient scaling of language models with mixture-of-experts. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, vol. 162, pp. 5547–5569. PMLR.

DURAISAMY, K., IACCARINO, G. & XIAO, H. 2019 Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.* **51**, 357–377.

DURAISAMY, K., ZHANG, Z.J. & SINGH, A.P. 2015 New approaches in turbulence and transition modeling using data-driven techniques. In *53rd AIAA Aerospace Sciences Meeting, AIAA Paper* 2015-1284. American Institute of Aeronautics and Astronautics.

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), 2100–2115.

FEDUS, W., ZOPH, B. & SHAZEER, N. 2022 Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.* **23** (120), 1–39.

FISHER, A., RUDIN, C. & DOMINICI, F. 2019 All models are wrong, but many are useful: learning a variable's importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.* **20** (177), 1–81.

FRENCH, R.M. 1999 Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci.* **3** (4), 128–135.

GLOROT, X., BORDES, A. & BENGIO, Y. 2011 Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 15, pp. 315–323. PMLR.

GORLE, C., EMORY, M. & IACCARINO, G. 2012 RANS modeling of turbulent mixing for a jet in supersonic cross flow: model evaluation and uncertainty quantification. In *Proceedings of the Seventh International Symposium on Turbulence Heat and Mass Transfer (THMT-12)*, pp. 1794–1797. Begell House Inc.

GREENBLATT, D., PASCHAL, K.B., YAO, C.S. & HARRIS, J. 2006*a* Experimental investigation of separation control part 2: zero mass-flux oscillatory blowing. *AIAA J.* **44** (12), 2831–2845.

GREENBLATT, D., PASCHAL, K.B., YAO, C.S., HARRIS, J., SCHAEFFLER, N.W. & WASHBURN, A.E. 2006*b* Experimental investigation of separation control part 1: baseline and steady suction. *AIAA J.* **44** (12), 2820–2830.

HE, P., MADER, C.A., MARTINS, J.R. & MAKI, K.J. 2018 An aerodynamic design optimization framework using a discrete adjoint approach with OpenFOAM. *Comput. Fluids* **168**, 285–303.

HE, P., MADER, C.A., MARTINS, J.R. & MAKI, K.J. 2020 DAFoam: an open-source adjoint framework for multidisciplinary design optimization with OpenFOAM. *AIAA J.* **58** (3), 1304–1319.

HINTON, G.E. & SALAKHUTDINOV, R.R. 2006 Reducing the dimensionality of data with neural networks. *Science* **313** (5786), 504–507.

JACOBS, R.A., JORDAN, M.I., NOWLAN, S.J. & HINTON, G.E. 1991 Adaptive mixtures of local experts. *Neural Comput.* **3** (1), 79–87.

JORDAN, M.I. & JACOBS, R.A. 1994 Hierarchical mixtures of experts and the EM algorithm. *Neural Comput.* **6** (2), 181–214.

KINGMA, D.P. & BA, J. 2015 Adam: a method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, pp. 1–15.

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

KIRKPATRICK, J. *et al.* 2017 Overcoming catastrophic forgetting in neural networks. *Proc. Natl Acad. Sci. USA* **114** (13), 3521–3526.

LAUNDER, B.E., REECE, G.J. & RODI, W. 1975 Progress in the development of a Reynolds-stress turbulence closure. *J. Fluid Mech.* **68** (3), 537–566.

LAUNDER, B.E. & SHARMA, B.I. 1974 Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc. *Lett. Heat Mass Transfer* **1** (2), 131–137.

LEE, M. & MOSER, R.D. 2015 Direct numerical simulation of turbulent channel flow up to $Re_\tau = 5200$. *J. Fluid Mech.* **774**, 395–415.

LING, J., KURZAWSKI, A. & TEMPLETON, J. 2016 Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* **807**, 155–166.

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.

LOZANO-DURÁN, A. & BAE, H.J. 2023 Machine learning building-block-flow wall model for large-eddy simulation. *J. Fluid Mech.* **963**, A35.

MAAS, A.L., HANNUN, A.Y. & NG, A.Y. 2013 Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, vol. 30, p. 3. JMLR.

MAHALANOBIS, P.C. 1936 On the generalised distance in statistics. *Proc. Natl Inst. Sci. India* **2**, 49–55.

MANDLER, H. & WEIGAND, B. 2023 Feature importance in neural networks as a means of interpretation for data-driven turbulence models. *Comput. Fluids* **265**, 105993.

MENTER, F.R. 1994 Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA J.* **32** (8), 1598–1605.

MENTER, F.R. & MATYUSHENKO, A. 2025 Generalized k–ω (GEKO) two-equation turbulence model. *AIAA J.* **63** (11), 4590–4606.

NAKAYAMA, A. 1985 Characteristics of the flow around conventional and supercritical airfoils. *J. Fluid Mech.* **160**, 155–179.

NAUGHTON, J.W., VIKEN, S. & GREENBLATT, D. 2006 Skin friction measurements on the NASA hump model. *AIAA J.* **44** (6), 1255–1265.

OHLSSON, J., SCHLATTER, P., FISCHER, P.F. & HENNINGSON, D.S. 2010 Direct numerical simulation of separated flow in a three-dimensional diffuser. *J. Fluid Mech.* **650**, 307–318.

OULGHELOU, M., CHERROUD, S., MERLE, X. & CINNELLA, P. 2025 Machine-learning-assisted blending of data-driven turbulence models. *Flow Turbul. Combust.* **115** (3), 1095–1132.

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.

PASZKE, A. *et al.* 2019 PyTorch: an imperative style, high-performance deep learning library. *Adv. Neural Inform. Process. Syst.* **32**, 8024–8035.

PINELLI, A., UHLMANN, M., SEKIMOTO, A. & KAWAHARA, G. 2010 Reynolds number dependence of mean flow structure in square duct turbulence. *J. Fluid Mech.* **644**, 107–122.

RINCÓN, M.J., RECLARI, M., YANG, X.I.A. & ABKAR, M. 2025 A generalisable data-augmented turbulence model with progressive and interpretable corrections for incompressible wall-bounded flows. *Intl J. Heat Fluid Flow* **116**, 109970.

ROLNICK, D., AHUJA, A., SCHWARZ, J., LILLICRAP, T.P. & WAYNE, G. 2019 Experience replay for continual learning. *Adv. Neural Inform. Process. Syst.* **32**, 350–360.

RUMELHART, D.E., HINTON, G.E. & WILLIAMS, R.J. 1986 Learning representations by back-propagating errors. *Nature* **323** (6088), 533–536.

RUMSEY, C.L. 2014 Turbulence modeling verification and validation. In *52nd AIAA Aerospace Sciences Meeting, AIAA Paper* 2014-0201. American Institute of Aeronautics and Astronautics.

RUMSEY, C.L. & COLEMAN, G.N. 2022 NASA symposium on turbulence modeling: roadblocks, and the potential for machine learning. Technical report, Langley Research Center.

RUSU, A.A., RABINOWITZ, N.C., DESJARDINS, G., SOYER, H., KIRKPATRICK, J., KAVUKCUOGLU, K., PASCANU, R. & HADSELL, R. 2016 Progressive neural networks. arXiv:1606.04671.

SANDBERG, R.D. & ZHAO, Y. 2022 Machine-learning for turbulence and heat-flux model development: a review of challenges associated with distinct physical phenomena and progress to date. *Intl J. Heat Fluid Flow* **95**, 108983.

SCHMELZER, M., DWIGHT, R.P. & CINNELLA, P. 2020 Discovery of algebraic Reynolds-stress models using sparse symbolic regression. *Flow Turbul. Combust.* **104** (2), 579–603.

SHAZEER, N., MIRHOSEINI, A., MAZIARZ, K., DAVIS, A., LE, Q., HINTON, G. & DEAN, J. 2017 Outrageously large neural networks: the sparsely-gated mixture-of-experts layer. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, pp. 1–19.

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press

SPALART, P.R. 2000 Strategies for turbulence modelling and simulations. *Intl J. Heat Fluid Flow* **21** (3), 252–263.

SPALART, P.R. & ALLMARAS, S.R. 1992 A one-equation turbulence model for aerodynamic flows. In *30th Aerospace Sciences Meeting and Exhibit, AIAA Paper* 1992-439. American Institute of Aeronautics and Astronautics.

TISHBY, N., PEREIRA, F.C. & BIALEK, W. 2000 The information bottleneck method. arXiv:physics/0004057.

VINUESA, R., SCHLATTER, P. & NAGIB, H.M. 2018 Secondary flow in turbulent ducts with increasing aspect ratio. *Phys. Rev. Fluids* **3** (5), 054606.

WANG, J., WU, J. & XIAO, H. 2017 Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data. *Phys. Rev. Fluids* **2**, 034603.

WANG, L., ZHANG, X., SU, H. & ZHU, J. 2024 A comprehensive survey of continual learning: theory, method and application. *IEEE Trans. Pattern Anal. Mach. Intell.* **46** (8), 5362–5383.

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.

WILCOX, D.C. 2008 Formulation of the k–ω turbulence model revisited. *AIAA J.* **46** (11), 2823–2838.

WU, C. & ZHANG, Y. 2023 Enhancing the shear-stress-transport turbulence model with symbolic regression: a generalizable and interpretable data-driven approach. *Phys. Rev. Fluids* **8**, 084604.

WU, C., ZHANG, X.-L., XU, D. & HE, G. 2025 A framework for learning symbolic turbulence models from indirect observation data via neural networks and feature importance analysis. *J. Comput. Phys.* **537**, 114068.

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.

XIAO, H., WU, J.-L., WANG, J.-X., SUN, R. & ROY, C.J. 2016 Quantifying and reducing model-form uncertainties in Reynolds-averaged Navier–Stokes simulations: a data-driven, physics-informed Bayesian approach. *J. Comput. Phys.* **324**, 115–136.

YAMAMOTO, Y. & TSUJI, Y. 2018 Numerical evidence of logarithmic regions in channel flow at $Re_\tau = 8000$. *Phys. Rev. Fluids* **3** (1), 012602.

YIN, Y., YANG, P., ZHANG, Y., CHEN, H. & FU, S. 2020 Feature selection and processing of turbulence modeling based on an artificial neural network. *Phys. Fluids* **32** (10), 105117.

ZHANG, T., CHEN, J., ZENG, F., TANG, D. & YAN, C. 2023 Improvement of transition prediction model in hypersonic boundary layer based on field inversion and machine learning framework. *Phys. Fluids* **35** (2), 024104.

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

ZHANG, F., ZHOU, Z., YANG, X. & HE, G. 2025 Knowledge-integrated additive learning for consistent near-wall modelling of turbulent flows. *J. Fluid Mech.* **1011**, R1.

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.

ZHU, L., ZHANG, W., KOU, J. & LIU, Y. 2019 Machine learning methods for turbulence modeling in subsonic flows around airfoils. *Phys. Fluids* **31** (1), 015105.

DE ZORDO-BANLIAT, M., DERGHAM, G., MERLE, X. & CINNELLA, P. 2024 Space-dependent turbulence model aggregation using machine learning. *J. Comput. Phys.* **497**, 112628.

https://doi.org/10.1017/jfm.2026.11654 Published online by Cambridge University Press