J. Fluid Mech. (2025), *vol.* 1016, R1, doi:10.1017/jfm.2025.10332



Data-enabled discovery of specific and generalisable turbulence closures

Zhongxin Yang[1,2,3], **Xianglin Shan**[1,2,3], **Xiang I.A. Yang**[4] and **Weiwei Zhang**[1,2,3]

[1]School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, PR China

[2]International Joint Institute of Artificial Intelligence on Fluid Mechanics, Northwestern Polytechnical University, Xi'an 710072, PR China

[3]National Key Laboratory of Aircraft Configuration Design, Xi'an 710072, PR China

[4]Mechanical Engineering, Pennsylvania State University, State College 16802, USA

Corresponding author: Weiwei Zhang, aeroelastic@nwpu.edu.cn

(Received 6 March 2025; revised 27 May 2025; accepted 29 May 2025)

Turbulence closures are essential for predictive fluid flow simulations in both natural and engineering systems. While machine learning offers promising avenues, existing data-driven turbulence models often fail to generalise beyond their training datasets. This study identifies the root cause of this limitation as the conflation of generalisable flow physics and dataset-specific behaviours. We address this challenge using symbolic regression, which yields interpretable, white-box expressions. By decomposing the learned corrections into inner-layer, outer-layer and pressure-gradient components, we isolate universal physics from flow-specific features. The model is trained progressively using high-fidelity datasets for plane channel flows, zero-pressure-gradient turbulent boundary layers (ZPGTBLs), and adverse pressure-gradient turbulent boundary layers (PGTBLs). For example, direct application of a model trained on channel flow data to ZPGTBLs results in incorrect skin friction predictions. However, when only the generalisable inner-layer component is retained and combined with an outer-layer correction specific to ZPGTBLs, predictions improve significantly. Similarly, a pressure-gradient correction derived from PGTBL data enables accurate modelling of aerofoil flows with both favourable and adverse pressure gradients. The resulting symbolic corrections are compact, interpretable, and generalise across configurations – including unseen geometries such as aerofoils and Reynolds numbers outside the training set. The models outperform baseline Reynolds-averaged Navier–Stokes closures (e.g. the Spalart–Allmaras and shear stress transport models) in both *a priori* and *a posteriori* tests. These results demonstrate that explicit identification and retention of generalisable

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

components is key to overcoming the generalisation challenge in machine-learned turbulence closures.

Key words: machine learning, turbulence modelling

1. Introduction

High-fidelity computational fluid dynamics tools, such as direct numerical simulations (DNS) and large-eddy simulations (LES), remain costly despite advancements in high-performance computing (Choi & Moin 2012; Yang & Griffin 2021; Yang *et al.* 2024*a*). The Reynolds-averaged Navier–Stokes (RANS) method will therefore remain the workhorse of fluids engineering for the foreseeable future (Slotnick *et al.* 2014; Li, Yang & Kunz 2022). Conventional RANS closures rely heavily on heuristics and dimensional arguments (Spalart 2000; Durbin 2018). Research in this area dates back to Prandtl and his mixing length model, in which he invoked the Boussinesq hypothesis and modelled the eddy viscosity in a wall-bounded flow as a function of the strain rate and a mixing length:

$$\nu_t = l^2 \left| \frac{\partial U}{\partial y} \right|, \tag{1.1}$$

where ν_t is the eddy viscosity, l is the mixing length, and $\partial U/\partial y$ is the strain rate in the wall-normal direction. Prandtl (1925) drew an analogy between the mixing length l and the mean free path, explaining that 'the mixing length may be considered as the diameter of the masses of fluid moving as a whole in each individual case; or again, as the distance traversed by a mass of the type before it becomes blended in with neighboring masses', thereby giving the mixing length a physical interpretation. Despite its simplicity, the mixing length model remains one of the most widely used turbulence closures, and serves as the foundation for nearly all wall models (Bose & Park 2018).

Following Prandtl, modelling efforts shifted towards transport-type models, leading to the development of the one-equation Spalart–Allmaras (SA) model (Spalart & Allmaras 1992), the two-equation k-ω shear stress transport (SST) model (Menter 1994), and the seven-equation full Reynolds stress model (Cécora *et al.* 2015). Like the mixing length model, transport turbulence closures have clear physical interpretations: their equations can be decomposed into production, destruction and diffusion terms. However, these models lack the accuracy required for designing next-generation aircraft (Slotnick *et al.* 2014). Furthermore, even when high-fidelity data are available, it is unclear how conventional models should be adjusted to accommodate a given flow.

A primary goal of machine-learned models is to address the limitations of conventional approaches. The use of machine learning in turbulence modelling dates back to Sarghini *et al.* (2003) and has since been widely explored. One framework that has received significant attention is field-inversion and machine learning (FIML) (Parish & Duraisamy 2016). By emphasising model-consistent training, FIML ensures that the training process remains aligned with the underlying turbulence model. This contrasts with *a priori* training, where training is performed by directly extracting target and input features from DNS (Wu, Xiao & Paterson 2018), often leading to a mismatch between training and prediction environments (Guo, Xia & Chen 2021).

Another notable framework is gene expression programming (GEP) or symbolic regression (SR). Unlike black-box models such as neural networks, GEP and SR produce interpretable, white-box algebraic equations (Weatheritt & Sandberg 2016). This approach has proven highly effective. Vaddireddy *et al.* (2020) applied GEP and successfully

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

rediscovered the Smagorinsky model when solving the Kraichnan turbulence problem. Wu & Zhang (2023*a*) introduced a data-driven turbulence modelling framework that integrates field inversion and deep SR, demonstrating improved predictive capabilities and enhanced generalisability for complex separated flows compared to traditional RANS models. Wu & Zhang (2023*b*) enhanced the SST model by using SR to generate algebraic corrections that improved the model's adaptability to complex flow scenarios, enabling predictions that generalise across different boundary conditions. He, Wu & Zhang (2024) developed a data-driven turbulence modelling method that enhances the SA model for aerofoil stall prediction using SR, improving predictive accuracy without compromising performance in attached flows. Tang *et al.* (2023) introduced a deep-learning-based SR framework to derive explicit RANS turbulence closures, demonstrating improved prediction accuracy and interpretability compared to traditional machine-learned models. Shan & Zhang (2025) proposed an explicit correction by using SR for the SA model to improve its performance in adverse pressure-gradient flows, enhancing accuracy in separated flows.

While machine learning has shown much promise, a challenge with machine-learned models is their ability to generalise outside the training dataset. Rumsey, Coleman & Wang (2022) found that machine-learned augmentations, while enhancing the baseline model within the training dataset, are detrimental outside it. Vadrot, Yang & Abkar (2023), among others (Kou & Zhang 2021; Chen *et al.* 2023*a*; Li *et al.* 2024), made similar observations. Taghizadeh *et al.* (2021) systematically discussed the difficulties in making deep neural network predict flows they have not seen before, and concluded that there is still much room for neural networks dealing with complex and unknown flow conditions. Such widespread lack of generalisation can be attributed to a lack of distinction between the physics that is generalisable and the physics that is specific to the training dataset. We argue that learning flow physics specific to the training dataset and applying it outside is why machine-learned models do not generalise well. This issue must be resolved before machine-learned RANS models can be used in fluids engineering.

One solution to the generalisability issue is to ignore physics and rely on big data. The logic is that when the database is sufficiently large, all future problems become interpolation, eliminating the need for generalisability. This approach has proven effective in large language model, computer vision and other applications. However, big data is not a viable solution to the turbulence problem since turbulent flows exhibit a wide range of scales, patterns and behaviours that can vary significantly with different conditions and configurations, making it impossible to capture every scenario. This calls for new machine-learning and modelling strategies to address the generalisability issue.

The strategy presented in this work relies on SR, progressive machine learning, data-consistent training, and a fundamental understanding of flow physics. The key novelty of this work is its approach to addressing the generalisability issue. Specifically, we decompose the obtained expressions into components that are specific to the training dataset and those that are generalisable. By retaining the generalisable components and progressively training on increasingly complex flows, we demonstrate that data-driven models can achieve generalisability comparable to, if not better than, conventional models. While direct advancements in turbulence modelling remain modest, addressing the generalisability issue represents a significant step toward developing truly generalisable machine-learning models.

2. Methods

We present the details of our method, focusing on how it enables the distinction between generalisable and data-specific flow physics. Figure 1 provides a schema of the work flow.

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

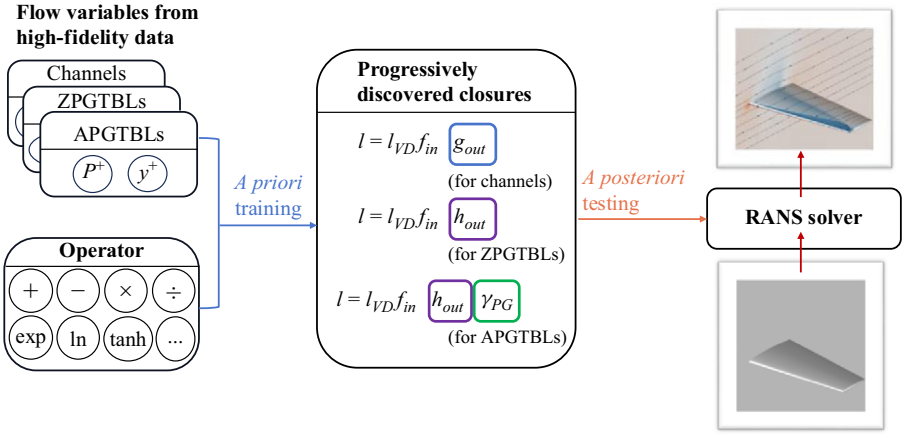


Figure 1. An overview of the workflow (where APGTBL means adverse PGTBL, VD refers to the mixing length proposed by van Driest 1956, and PG corresponds to pressure gradient).

The SR yields white-box models, which we interpret physically, separating generalisable parts and flow-specific parts. Flow-specific parts are removed when conducting transfer learning, whereas generalisable parts are retained. The trained models are applied to tackle RANS of complex (relative to those in the training dataset) flows outside the training dataset.

Machine learning involves numerous user-defined choices. For example, one may opt for *a priori* training or model-consistent training (Zhang & Duraisamy 2015; Parish & Duraisamy 2016), select either tensor-based or primitive flow variables as input features (Tracey, Duraisamy & Alonso 2015; Ling, Kurzawski & Templeton 2016), choose between constrained and unconstrained machine-learning approaches (Gimenez & Bre 2019; Bin *et al.* 2024), or select from single-source or heterogeneous, multi-fidelity data (Cao, Song & Zhang 2024; Wu *et al.* 2024; Zhang *et al.* 2024). This study specifically examines the generalisation of a model trained on one dataset to another, i.e. transfer learning across datasets. To highlight the effectiveness of our approach in transfer learning, we deliberately employ simpler methods in other aspects of the model to ensure that the observed improvements stem from the ability to distinguish between generalisable and data-specific flow physics.

2.1. *Using RANS*

We solve the Reynolds-averaged mass and momentum conservation equations:

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

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial P}{\partial x_i} + \nu \frac{\partial^2 U_i}{\partial x_j \partial x_j} - \frac{\partial \langle u_i' u_j' \rangle}{\partial x_j}. \tag{2.2}$$

Here, U_i and u_i' represent the ensemble average and turbulent fluctuation, respectively, in the ith direction of the Cartesian coordinate system x_i; ν is the kinematic viscosity, P is the mean pressure, ρ is the fluid density, and $-\langle u_i' u_j' \rangle$ is the Reynolds stress tensor, where $\langle \cdot \rangle$ denotes ensemble averaging. Invoking the Boussinesq hypothesis, the Reynolds stress tensor is modelled as

$$-\langle u_i' u_j' \rangle + \tfrac{2}{3}\delta_{ij} k = 2\nu_t S_{ij}, \tag{2.3}$$

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

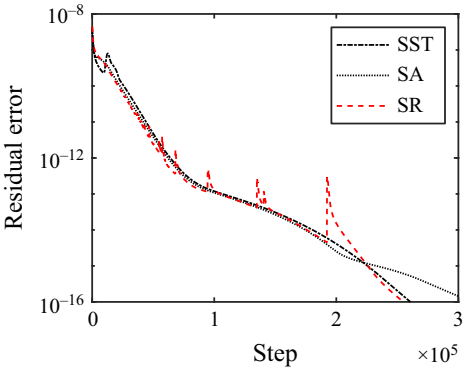


Figure 2. A sample convergence history of RANS equations.

where δ_{ij} is the Kronecker delta, k is the turbulent kinetic energy, and S_{ij} is the strain rate tensor. In this study, we model the eddy viscosity using (1.1). However, in general flows, the term $\partial U/\partial y$ in (1.1) must be replaced with the variable Ω, following the work of Baldwin & Lomax (1978), where

$$\Omega = |\Omega_i \Omega_i|^{1/2}, \quad \Omega_i = \epsilon_{ijk} \frac{\partial U_j}{\partial x_k} \tag{2.4}$$

represents the vorticity magnitude, and ϵ_{ijk} is the Levi–Civita tensor. Additionally, the machine-learned mixing length model requires local turbulent boundary-layer thickness information δ_{99}, which is generally defined as the value of y at which streamwise velocity $U(x, y)$ equals 99 % of the free-stream velocity $U_0(x)$ in the textbook Pope (2001). However, in practical numerical calculations, we typically choose $(1/0.65)$ times the height where the function

$$F(y) = y f_{in} \Omega \tag{2.5}$$

attains its maximum as the boundary-layer thickness, again following the work of Baldwin & Lomax (1978). Here, f_{in} is the wall damping, e.g. the one in van Driest (1956). Besides (1.1), we will invoke the SA model (Spalart & Allmaras 1992) and the SST model (Menter 1994) for comparison purposes. Details of the two models could also be found on the NASA Turbulence Modeling Resource (TMR) site and are not repeated here for brevity.

Three types of flow configurations are considered: plane channel flow, flat-plate turbulent boundary layers (TBLs), and aerofoil flow. These cases are standard test problems for turbulence model verification and validation, and detailed set-ups can also be found on the TMR site (Rumsey 2019). While multiple solutions to the RANS equations have been observed, for the cases here, the solutions are unique. In later sections, we will highlight only the essential details.

All RANS calculations are performed using the second-order finite-volume solver GFSI on sublayer resolved structured grids. The solver has been validated extensively, with prior applications documented in Zhang *et al.* (2015), Gao *et al.* (2017) and Zhu *et al.* (2019, 2021). Here, iterations continue until the domain-integrated residual falls below 10^{-16}. Figure 2 illustrates a sample residual history from one of our RANS calculations.

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

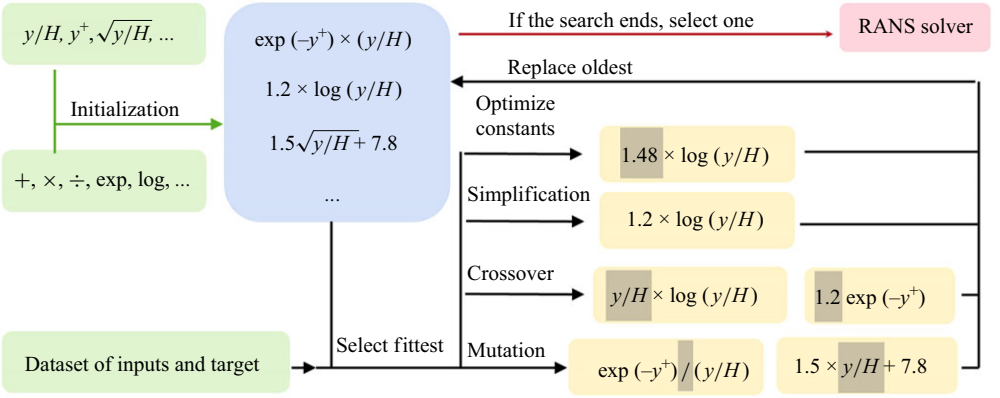


Figure 3. An illustration of SR.

2.2. *Using SR*

Figure 3 illustrates the process of SR, a machine-learning technique that identifies mathematical expressions best fitting a given dataset using a predefined set of operators without assuming a specific functional form. The process begins with an initial population of randomly generated mathematical expressions, each evaluated based on its ability to fit the observed data. The well-performing expressions are selected as parents, and new expressions are created through crossover, where segments of parent expressions are combined, and mutation, where small random modifications are introduced. This iterative process of evaluation, selection, crossover and mutation continues over multiple generations, gradually refining the expressions until a satisfactory one is found or a predefined number of generations is reached. Further details of the algorithm, including a pseudo-code, are included for references purposes.

We implement SR using PySR, a Python library for SR (Cranmer 2023). The search process relies on a predefined set of operators inspired by existing turbulence closures, including unary functions such as $\exp(\cdot)$, $\ln(\cdot)$ and $\tanh(\cdot)$, as well as binary operations such as addition, multiplication, division and exponentiation. The quality of a candidate expression is measured using a logarithmic loss function following Cranmer (2024). The predictive loss function is defined as

$$L_{pred} = \sum \ln\left(\left|\frac{\hat{Y}_i}{Y_i}\right|\right), \tag{2.6}$$

where $\hat{Y}_i$ and Y_i represent the predicted and target of the ith data point in the train set, respectively. The target for each training is listed in table 1. The cost function is not weighted, and as the computational mesh clusters in the wall layer, more weight is given to the wall layer. To avoid overly complex expressions, we also introduce a complexity penalty following Cranmer (2023). Here, the complexity of an expression is defined as

$$C = N_{var} + N_{const} + N_{op}, \tag{2.7}$$

where N_{var}, N_{const} and N_{op} denote the number of variables, constants and operations, respectively. The total loss of an expression includes two parts, the predictive loss and the complexity penalty:

$$L = L_{pred} \times \exp(\text{frecency}\,[C]), \tag{2.8}$$

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

	Plane channel	ZPGTBL	PGTBL
Input	y/H, y^+, $\sqrt{y/H}$, $\sqrt{y^+}$, Re_τ	y/δ_{99}, Re_τ	y^+, $(\mathrm{d}P/\mathrm{d}x)^+$
Target	l/l_{VD}	$l/(l_{VD}f_{in})$	$l/(l_{VD}f_{in}h_{out})$
Unary operator	$\exp(\cdot)$, $\ln(\cdot)$, $\tanh(\cdot)$		
Binary operator	$\cdot+\cdot$, $\cdot\times\cdot$, $\cdot/\cdot$, $\cdot^{\cdot}$,$1-\exp(\cdot/\cdot)$		
Predictive loss function	$\sum\ln(\lvert\hat{Y}_i/Y_i\rvert)$		

Table 1. Input, target, physical loss and operations of SR. Here, y is the distance from the wall, H is the half-width of the channel, y^+ is $(y/H)Re_\tau$ (where $Re_\tau=\delta u_\tau/\nu$), δ is the characteristic length of the flow (e.g. H in channel flow, the thickness of the boundary layer δ_{99} in the TBL) , $u_\tau=\sqrt{\tau_w/\rho}$ is the friction velocity, τ_w is the wall shear stress, $(\mathrm{d}P/\mathrm{d}x)^+=(\delta_\nu/\tau_w)(\mathrm{d}P/\mathrm{d}x)$ is the dimensionless pressure gradient, $\delta_\nu=\nu/u_\tau$ is inner length scale, l is the 'true' mixing length calculated from high-fidelity data, l_{VD} is the mixing length in van Driest (1956), f_{in} is the generalisable inner function learned from plane channel data, and h_{out} is the TBL-specific outer function learned from the ZPGTBL.

where the 'frecency' of C is a combined measure of the recency of expressions occurring at complexity C in the population. The reader is directed to Cranmer (2023) for additional details. Table 1 provides details on the inputs and targets for our training against channel flow, zero-pressure-gradient TBLs (ZPGTBLs) and pressure-gradient TBLs (PGTBLs). The selection of the inputs and the mathematical operators is guided by our physical understanding of the flow problem, the availability of relevant flow variables, and the formulations of prior models. For instance, in the context of near-wall turbulence modelling, variables such as distance from the wall are known to be important. Also, the input features include not only direct flow variables, but also their transformations to reduce the search depth of the SR algorithm. For instance, in channel flow, y^+ is related to (y/H) and Re_τ, yet all three variables are included as inputs. This practice is known as feature engineering. Feature engineering is critical for any machine-learning method. For example, Huang, Yang & Kunz (2019) found that their neural network generalised well only when using the logarithm of a variable instead of the variable itself. Similarly, Chen *et al.* (2022*b*) demonstrated that the generalisability of a neural network depends critically on the choice of input features. These studies have motivated the careful design of the input features here. Hyperparameters such as population size, number of individuals per population, and generations per cycle are carefully chosen to balance efficiency and search effectiveness. In this study, we set the population size to 64, limit each population to 500 individuals, and allow a maximum of 300 generations per cycle. Under these settings, the expressions obtained from SR appear to be robust and repeatable. However, we acknowledge that varying the search depth may yield different expressions with similar levels of fit to the reference data. Note that SR searches for mathematical expressions rather than optimising continuous parameters, making its search space discrete and non-differentiable. As a result, gradient-based methods are not suitable, and evolutionary algorithms or genetic programming must be used instead.

2.3. *Data*

We use data from Schlatter & Örlü (2010), Eitel-Amor, Örlü & Schlatter (2014), Lee & Moser (2015), Bobke *et al.* (2017), Yamamoto & Tsuji (2018), Tanarro, Vinuesa & Schlatter (2020), Hoyas *et al.* (2022) for training and validation. The datasets from Lee & Moser (2015), Yamamoto & Tsuji (2018) and Hoyas *et al.* (2022) correspond

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

Ref.	Case name	Geometry	Re_τ	Re_θ	β	For training
Lee & Moser (2015)	CH-180	Channel	180	\	≈ 0	N
	CH-550	Channel	550	\	≈ 0	N
	CH-1000	Channel	1000	\	≈ 0	Y
	CH-2000	Channel	2000	\	≈ 0	N
	CH-5200	Channel	5200	\	≈ 0	Y
Yamamoto & Tsuji (2018)	CH-8000	Channel	8000	\	≈ 0	N
Hoyas *et al.* (2022)	CH-10000	Channel	10 000	\	≈ 0	N
Schlatter & Örlü (2010)	ZPG-DNS	Boundary layer	[252, 1271]	[677, 4061]	$= 0$	N
Eitel-Amor *et al.* (2014)	ZPG-LES	Boundary layer	[457, 2479]	[1376, 8300]	$= 0$	Y
Bobke *et al.* (2017)	APG-b1	Boundary layer	[190, 862]	[910, 3360]	≈ 1	Y
	APG-b2	Boundary layer	[189, 910]	[940, 4000]	≈ 2	N
	APG-m13	Boundary layer	[190, 896]	[990, 3515]	[0.86, 1.49]	N
	APG-m16	Boundary layer	[189, 934]	[1010, 4000]	[1.55, 2.55]	Y
	APG-m18	Boundary layer	[192, 973]	[990, 4320]	[2.15, 4.07]	N
Tanarro *et al.* (2020)	AF-0012	Aerofoil	[117, 372]	[239, 1667]	[0, 3.7]	N

Table 2. Details of the data in Lee & Moser (2015), Yamamoto & Tsuji (2018), Hoyas *et al.* (2022), Schlatter & Örlü (2010), Eitel-Amor *et al.* (2014), Bobke *et al.* (2017) and Tanarro *et al.* (2020). In the last column, Y and N represent Yes and No.

to plane channel flows. The data from Schlatter & Örlü (2010) and Eitel-Amor *et al.* (2014) pertain to ZPGTBLs, while Bobke *et al.* (2017) provide data for PGTBLs. The dataset from Tanarro *et al.* (2020) corresponds to aerofoil flows. Table 2 provides further details, listing the reference, geometry, range of friction Reynolds number $Re_\tau = u_\tau \delta/\nu$, momentum-thickness-based Reynolds number $Re_\theta = U_e\theta/\nu$ (when applicable), the Clauser pressure-gradient parameter $\beta = (\delta^*/\tau_w)\,\mathrm{d}P/\mathrm{d}x$ (when applicable), and whether the case is used for training. Here, $u_\tau = \sqrt{\tau_w/\rho}$ is the friction velocity, U_e is the free-stream velocity, θ is the momentum thickness, δ^* is the displacement thickness, and τ_w is the wall shear stress.

3. Turbulence closures

3.1. *Details of training*

We apply SR to learn corrections to the mixing length model to account for various types of flow physics. The machine-learning process involves multiple stages, progressively incorporating increasingly complex flows – though none of the flows used for training is complex in an absolute sense. Again, the key novelty lies in addressing the generalisability issue and enabling transfer learning rather than in complex flows.

We first train on plane channel flow, where the mean flow is one-dimensional and depends only on the wall-normal coordinate y. Prior knowledge of the mixing length is incorporated: rather than using the mixing length l as the direct target, we train for l/l_{VD} – a strategy previously adopted in Yang *et al.* (2019), where the target was the mean velocity and the prior knowledge was the law of the wall. This approach has proven effective. Here, $l_{VD} = \kappa y D$, $D = (1 - \exp(-y^+/A^+))$ represents an established model for the mixing length in the inner layer (van Driest 1956), where $A^+ = 26$ is the constant. The training dataset includes channel flow cases CH-1000 and CH-5200, with friction Reynolds numbers $Re_\tau = 1000$ and 5200, respectively.

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

Applying SR to the channel flow data yields

$$l = \kappa y \left[1 - \exp\left(-\frac{y^+}{A^+}\right)\right]\left[1 - \exp\left(-\frac{\sqrt{y^+}}{3.6}\right) + \frac{1.21\left(y^+ + \sqrt{y^+}\right)}{\exp(\sqrt{y^+}) + 8.19}\right]^{-1} \exp\left(-\frac{y}{H}\right). \tag{3.1}$$

Due to the white-box nature of SR, we can decompose the learned closure into the prior model l_{VD}, an inner-layer correction f_{in}, and an outer-layer correction g_{out}:

$$l = l_{VD}\, f_{in}(y^+)\, g_{out}(y/H), \tag{3.2}$$

where

$$f_{in} = \left[1 - \exp\left(-\frac{\sqrt{y^+}}{3.6}\right) + \frac{1.21\left(y^+ + \sqrt{y^+}\right)}{\exp(\sqrt{y^+}) + 8.19}\right]^{-1} \tag{3.3}$$

and

$$g_{out} = \exp\left(-\frac{y}{H}\right). \tag{3.4}$$

Here, the Reynolds number information is implicitly included since $Re_\tau = y^+/(y/H)$. Since the inner layer of all wall-bounded flows is universal, the inner-layer correction f_{in} is expected to generalise across different flows. Conversely, the outer-layer correction g_{out} is specific to plane channel flow. If the obtained model is not decomposed into its generalisable and dataset-specific components, then it would remain applicable only to channel flow – highlighting why prior data-driven models often fail to generalise. By explicitly identifying and retaining only the generalisable components, we enable both generalisation and transfer learning. Equation (3.2) can be obtained through physical arguments. The purpose of this exercise is to show that SR, a machine-learning tool, has the ability to identify the internal structure of a function – a capability that black-box-type function approximators generally lack.

Before progressing to the next flow configuration, several remarks on (3.1) are warranted. First, the form in (3.1) is a result of SR, whereas (3.2) represents our physical interpretation of the white-box model in (3.1). In other words, SR provides a transparent, interpretable model from which (3.2) can be extracted. Once the structure in (3.2) is known, one can indeed use any regression method – such as neural networks – to learn f_{in} and g_{out}. The expression in (3.1) is a result of combined considerations of complexity and accuracy. Second, like any other machine learning, SR is also susceptible to overfitting. Table 3 shows the complexity level, the corresponding expression and the loss function. Figure 4 illustrates the complexity–accuracy trade-off. We observed that beyond complexity level 20, further increases in complexity yield only marginal improvements in the loss function. This poor complexity–accuracy trade-off indicates overfitting – although overfitting is formally defined as a lack of generalisation to unseen data, in SR literature, poor complexity–accuracy trade-off is often used as an indicator of overfitting. The above is why we picked the expression at complexity level 20 ((3.1) corresponds to the expression at complexity level 20). Admittedly, the threshold at which additional complexity yields negligible reduction in the loss function is somewhat subjective. Nonetheless, the form in (3.2) is robust. We could have selected the expressions at complexity levels 17–22, all of which conform to (3.2). For completeness, we also showcase expressions at a few complexity levels from 1 to 30. We see that too low a complexity level leads to clear underfitting, and too high a complexity level leads to overfitting, i.e. an overly complex expression with practically the same loss function.

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

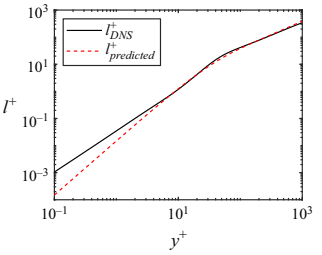
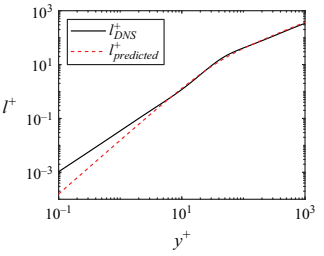
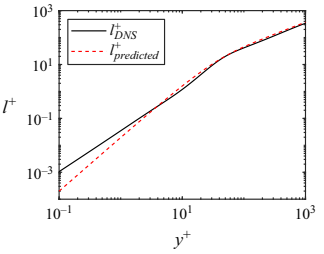
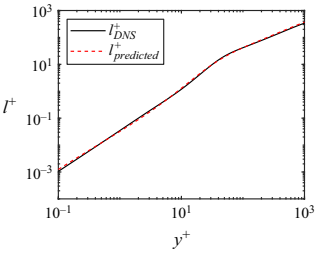
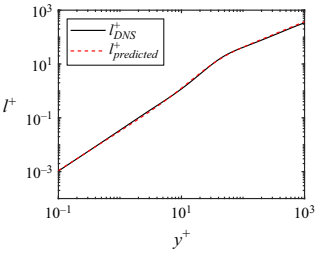
Complexity	Expressions	True values and predicted values	Loss function
1	$0.952 l_{VD}$		0.1937
2	$l_{VD} \exp\left(-\frac{y}{H}\right)$		0.1696
3	$l_{VD}\left[0.789+\sqrt{\frac{y}{H}}\right]^{-1}$		0.1638
17	$l_{VD}\left[\frac{1-\exp\left(\frac{\sqrt{y^+}}{3.30}\right)}{+\frac{y^+}{\exp\left(\sqrt{y^+}\right)+1.91}}\right]^{-1} \times \exp\left(-\frac{y}{H}\right)$		0.0454
20	$l_{VD}\left[\frac{1-\exp\left(\frac{\sqrt{y^+}}{3.61}\right)}{+\frac{\left(y^+ + \sqrt{y^+}\right)}{0.826\exp\left(\sqrt{y^+}\right)+\exp(1.91)}}\right]^{-1} \times \exp\left(-\frac{y}{H}\right)$		0.0360

Table 3. For caption see next page.

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

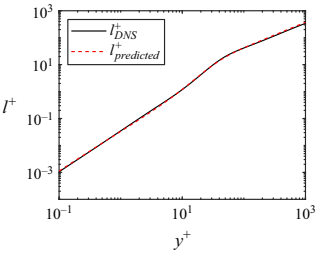
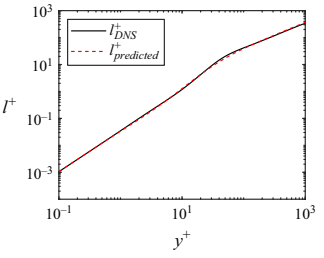
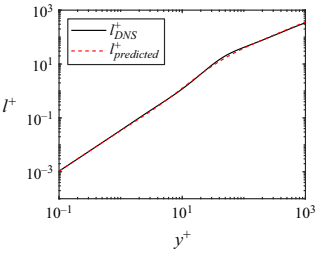
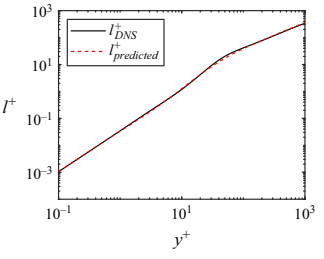
Complexity	Expressions	True values and predicted values	Loss function
22	$l_{VD}\left[\begin{array}{c}1-\exp\left(\frac{\sqrt{y^+}}{3.61}\right)\\ +\frac{\left(y^+ + \sqrt{y^+}\right)}{\exp\left(\sqrt{y^+}\right)+\exp(1.720)}\\ \times \exp\left(-\frac{y}{H}\right)\end{array}\right]^{-1}$		0.0336
27	$1.02 l_{VD}\left[\begin{array}{c}1-\exp\left(-\frac{y^+}{2.84}\right)+\frac{y}{H}\\ +\exp\left(-\sqrt{y^+}\right)\left[1-\exp\left(-\frac{\sqrt{y^+}}{1.91}\right)\right.\\ \left.+\log(9.75\times10^{-3})\frac{(y^+)^2}{Re_\tau}\right]\end{array}\right]^{-1}$		0.0315
28	$1.03 l_{VD}\left[\begin{array}{c}1-\exp\left(-\frac{y^+}{2.84}\right)+\frac{y}{H}\\ +\exp\left(-\sqrt{y^+}\right)\left[1-\exp\left(-\frac{\sqrt{y^+}}{1.91}\right)\right.\\ \left.-0.650\frac{(y^+)^{2.5}}{Re_\tau}\right]\end{array}\right]^{-1}$		0.0270
30	$1.03 l_{VD}\left[\begin{array}{c}1-\exp\left(-\frac{y^+}{2.84}\right)+y/H\\ +\exp\left(-\sqrt{y^+}\right)\left[1-\exp\left(-\frac{\sqrt{y^+}}{1.91}\right)\right.\\ \left.+\left(\frac{y}{H}-0.725\right)\frac{(y^+)^{2.5}}{Re_\tau}\right]\end{array}\right]^{-1}$		0.0258

Table 3 (cntd). The expressions at different complexities, and their predicted values and loss. The expressions at complexities 1, 2, 3 correspond to underfitting. The expressions at complexities 17, 20, 22 are well-generalised, while the expressions at complexities 27, 28, 30 correspond to overfitting.

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

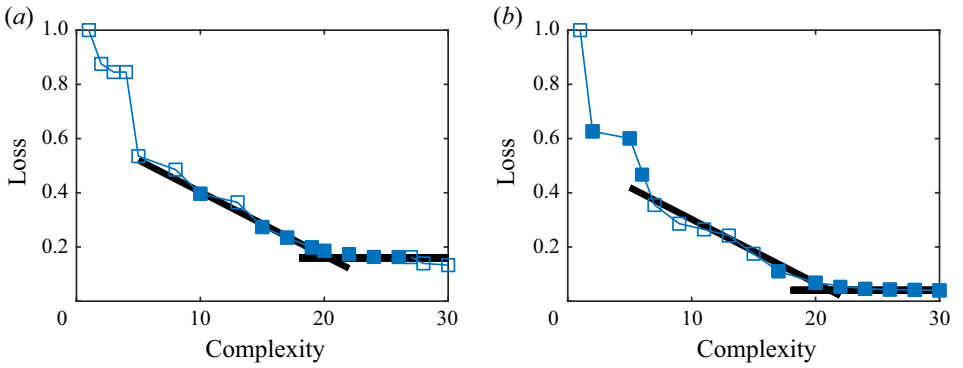


Figure 4. Normalised loss as a function of complexity from (*a*) one realisation corresponding to table 3, and (*b*) another realisation. The losses are normalised by the loss at complexity level 1. Filled symbols indicate that the expressions conform to the structure in (3.2). The two black solid lines represent finite and diminishing returns in loss reduction when increasing complexity before and beyond level 20. The skipping (e.g. 18, 21, 29 in (*a*)) is due to the fact that PySR will not return an expression with a loss function higher than that of the expression of a lower complexity.

Third, the redundant input Re_τ does not show up in the expression, showing that it is indeed redundant and that the training is robust to redundant inputs. Fourth, SR yields the correct asymptotic behaviour for the eddy viscosity at the wall. The eddy viscosity should scale as $\nu_t \sim O(y^3)$ near the wall (Pope 2001) because

$$\lim_{y\to 0} \nu_t = \lim_{y\to 0} \frac{-\langle u'v' \rangle}{\mathrm{d}U/\mathrm{d}y} \sim \lim_{y\to 0} \frac{y^3}{1} \sim y^3; \tag{3.5}$$

however, the conventional model in van Driest (1956) behaves as

$$\lim_{y\to 0} \nu_t = \lim_{y\to 0} l^2 \frac{\mathrm{d}U}{\mathrm{d}y} \sim O(y^4), \tag{3.6}$$

which is inaccurate. The closure in (3.1), on the other hand, gives

$$\lim_{y\to 0} \nu_t = \lim_{y\to 0} (l_{VD}\, f_{in}\, g_{out})^2 \frac{\mathrm{d}U}{\mathrm{d}y} \sim O(y^3), \tag{3.7}$$

which correctly recovers the expected asymptotic behaviour. This is also evident from figure 6(*a*), where we compare the SR model, the conventional van Driest model, and DNS data for Re_τ ranging from 180 to 10 000. The goal here is to validate the SR methodology by showing that the obtained expressions closely match the data. Comparison to existing turbulence models will be introduced in § 4, where we show results for *a posteriori* tests. Fifth, we trained the SR model to predict the ratio l/l_{VD}. This formulation isolates the correction relative to the baseline model. We may train the SR model to directly predict l. The resulting expression still takes the form of (3.2), and the inner layer correction $f_{in,2}$ equals $f_{in}D$ in terms of the performance, as shown in figure 5. However, unless we rearrange $f_{in,2}$, the van Driest function does not emerge explicitly.

We also distinguish between analysis and interpretation. Analysis involves examining the behaviour of a model and can be applied to both black-box models, such as neural networks, and white-box models, such as SR-derived expressions. Physical interpretation, on the other hand, involves parsing a model and assigning clear physical meanings to its components, which is typically possible only for white-box models. For example,

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

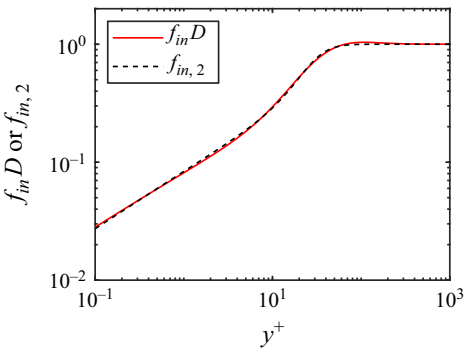


Figure 5. A comparison between $f_{in}D$ and $f_{in,2}$, which is $\left[1+562.4\left[48.5\sqrt{y^+}+\sqrt{y^+}\exp\left(\sqrt{y^+}\right)\right]^{-1}\right]^{-1}$.

(3.7) represents an analysis of the SR model, whereas (3.2) demonstrates its physical interpretation. While analysis may provide insights, it does not inherently lead to generalisability. In contrast, by distinguishing between generalisable and dataset-specific components, physical interpretation provides a path to achieving generalisability. This study focuses on physical interpretation. For the result in (3.1), our interpretation is the parsing of the equation into an inner-layer correction f_{in} and an outer-layer correction g_{out}. We can further interpret the two corrections as follows. First, f_{in} modifies the mixing length in the inner layer, compensating for the overly rapid decay of eddy viscosity in the viscous sublayer. This behaviour is evident from (3.5), (3.6), (3.7) and figure 6(*a*). In contrast, g_{out} adjusts the outer-layer mixing length by accounting for a slower-than-linear growth with distance from the wall.

Next, we train the model on the ZPGTBL. This flow is slightly more complex than in the plane channel. The mean flow is a function of the streamwise coordinate x and the wall-normal coordinate y. If we were to follow the conventional approach, then the model trained using channel flow data would yield poor results for the boundary layer, requiring training from scratch. However, because we parse the white-box model in (3.1) into a generalisable inner-layer part and a data-specific outer-layer part, here, we can take the generalisable part and use it for our training for boundary-layer flows. Specifically, we build on the generalisable inner-layer correction, and only train for an outer-layer correction for the boundary-layer configuration. Data for case ZPG-LES at a few Reynolds numbers between $Re_\theta = 1376$ and $Re_\theta = 8183$ are used for training. The resulting outer-layer correction takes the form

$$h_{out} = \tanh\left[\left(0.215 + (Re_\tau + 1.0086^{Re_\tau})^{-0.62}\right)\left(\frac{y}{\delta_{99}}\right)^{-1}\right], \tag{3.8}$$

where h_{out} plays a similar role to g_{out}, but employs a functional form based on $\tanh(a(\delta_{99}/y))$ to achieve the slower-than-linear growth. The Reynolds-number-dependent prefactor a captures finite Reynolds number effects (Klewicki 2010), and asymptotically approaches 0.215 at high Reynolds numbers. Note that while Re_τ is a redundant input when we train against channel flow data, it is not a redundant input here and cannot be obtained from the other input, y/δ_{99}. When (3.8) is used, the function h_{out} will replace g_{out} in (3.4). Here, we have assumed a multiplicative correction rather than, for example, an additive correction. This choice should not affect the model's performance, as the two are mathematically equivalent. For a baseline model denoted as B, a multiplicative

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

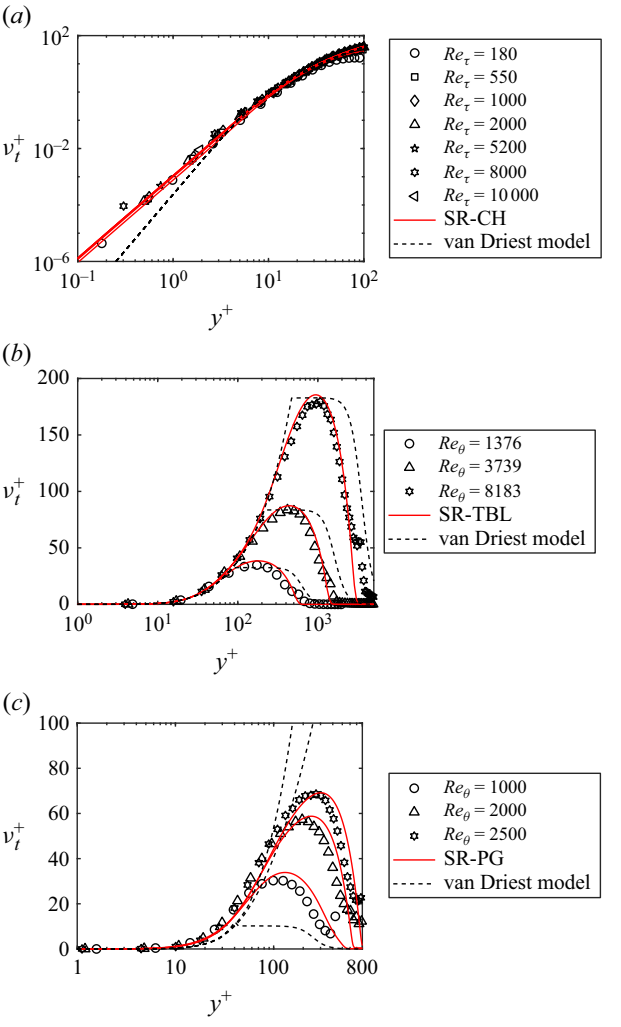


Figure 6. (*a*) Eddy viscosity in channel. (*b*) Eddy viscosity in ZPG-LES. (*c*) Eddy viscosity in APG-b1. The symbols are reference DNS data. The van Driest model corresponds to the black dashed line.

correction $1 + C_m$ and an additive correction C_a satisfy the relationship $BC_m = C_a$. The resulting eddy viscosity is compared against the high-fidelity LES data (Re_θ from 1376 to 8183) in figure 6(*b*), showing a close agreement between the two. The van Driest model is also shown for comparison purposes. We note that the van Driest model alone does not constitute a viable turbulence model as the linear behaviour in the outer layer must be damped using e.g. the outer-layer correction in Baldwin & Lomax (1978). We see that the van Driest model generally overpredicts the eddy viscosity.

Finally, we consider pressure gradient. In external aerodynamics applications, pressure gradients arise due to surface curvatures (Gungor *et al.* 2016), often leading to flow separation, which remains challenging to model. With a few exceptions (Chen *et al.* 2023*b*; Yang *et al.* 2024*b*), previous PGTBL modelling efforts have relied heavily on heuristics and empirical methods (Nickels 2004; Romero *et al.* 2022). Machine learning has also been applied to pressure-gradient-dominated flows, such as periodic hills, backward-facing steps and separated flows (Yan, Zhang & Chen 2022; Shan *et al.* 2023, 2024;

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

Tang *et al.* 2023; Wang & Zhang 2023; Zhou *et al.* 2023; Arranz *et al.* 2024; Zhou & Bae 2024). However, this study represents the first instance of isolating a machine-learned correction specifically for pressure-gradient effects. Here, we train the model on PGTBLs at conditions corresponding to cases APG-b1 and APG-m16. The training data cover a Reynolds number range from $Re_\theta = 940$ to $Re_\theta = 4000$, and a Clauser pressure-gradient parameter range between 1 and 2.55. Building on the previously obtained inner-layer correction f_{in} and the outer-layer correction h_{out} for TBLs, we learn an additional correction for the pressure gradient. Symbolic regression yields the following pressure-gradient correction:

$$\gamma_{PG} = 1 + 18 \left(\frac{\mathrm{d}P}{\mathrm{d}x}\right)^{+} \left[1 - \tanh\left(\frac{y^{+} - 12.7}{14}\right)\right]. \tag{3.9}$$

When used in RANS, the correction factor is incorporated into the mixing length as a multiplicative correction:

$$l = l_{VD}\, f_{in}\, h_{out}\, \gamma_{PG}. \tag{3.10}$$

Here, γ_{PG} modulates the eddy viscosity in the presence of streamwise pressure gradients: it reduces the eddy viscosity in favourable PGTBLs, and increases it in adverse pressure-gradient cases through the $b(\mathrm{d}P/\mathrm{d}x)$ term. The prefactor b, which varies with y^{+}, ensures that the influence of the pressure gradient diminishes near the wall. According to this result, pressure gradients do not significantly affect the validity of the mixing length formulation in the outer region. This finding aligns with the classic work by Kays *et al.* (1980), who noted that for PGTBLs, modifications to the mixing length model are primarily needed in the viscous layer. The fact that the present SR approach correctly identifies this adjustment highlights the strength of our methodology. The resulting eddy viscosity is compared against the data for case APG-b1 in figure 6(*c*), showing close agreement between the two, while the van Driest model performs poorly.

In summary, we apply SR and train our model on plane channel flow, ZPGTBLs and PGTBLs. The inner-layer correction obtained from plane channel flow is generalisable and is retained in all subsequent training. The two outer-layer corrections, g_{out} and h_{out}, are specific to channel flow and TBLs, respectively, with the boundary-layer correction being retained during training on PGTBLs. Finally, a pressure-gradient correction is learned through training on PGTBLs. Again, the novelty of the present approach lies in the identification of generalisable components within machine-learned models, and the systematic utilisation of these components in subsequent training, thereby overcoming the generalisability issue.

4. Verification and validation

We have assigned clear physical interpretations to the models obtained in § 3. The purpose of this section is to evaluate their generalisability. The test cases are chosen carefully. They are simple enough to ensure clarity in the machine-learning methodology without ambiguity in flow physics, yet challenging enough that existing RANS closures struggle – allowing us to demonstrate the advantages of our approach. Furthermore, these tests will illustrate that the failure to distinguish between generalisable and data-specific flow physics is the fundamental reason why machine-learned models often lack generalisability. Throughout this section, we will include results of the SA and SST models for comparison purposes. All flow variables, including those used as inputs in SR models, are computed dynamically as we solve the RANS equations.

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

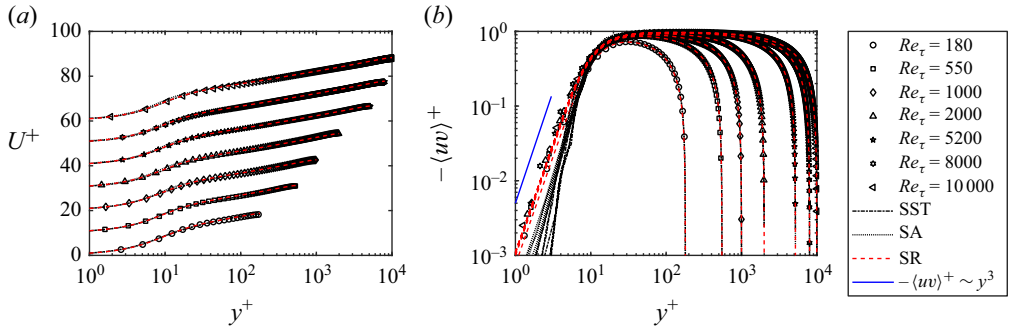


Figure 7. Channel flow results. The SR model contains the inner-layer correction f_{in} and the channel-specific outer-layer correction g_{out}. The present SR model has seen the configuration but not all the Reynolds numbers. (*a*) Mean velocity profile. For illustration purposes, neighbouring profiles are displaced by 10 plus units in the vertical axis. (*b*) Reynolds shear stress. The symbols are DNS reference data.

4.1. *Seen geometries*

We first consider plane channel flow. The SR model in this case contains the inner-layer correction f_{in} and the channel-specific outer-layer correction g_{out}. The RANS details are provided in § 2. Further details of the computational set-up for plane channel flow can be found on the TMR site and are not repeated here for brevity. Figure 7 presents the results, with Reynolds numbers ranging from $Re_\tau = 180$ to $Re_\tau = 10\,000$. While the SR model has been trained on the channel configuration, it has not seen all Reynolds numbers. Figure 7(*a*,*b*) show the mean velocity profiles and the corresponding Reynolds stresses, demonstrating close agreement between the present SR model and the DNS data across all Reynolds numbers. This result confirms that f_{in} and g_{out} generalise well to both seen and unseen Reynolds numbers.

Next, we consider ZPGTBLs. As before, we replicate the set-up from the TMR website. Figure 8(*a*) presents the skin friction coefficient as a function of Re_θ. We compare high-fidelity data, the SA and SST models, the model with the inner-layer correction f_{in} and the TBL-specific outer-layer correction h_{out}, and the model with f_{in} along with the channel-specific outer-layer correction g_{out}. The SST and SA models predict the friction coefficient at high Reynolds number, although there are errors in the low-Reynolds-number regime. While the model containing f_{in} and h_{out} closely matches the high-fidelity data, applying the channel-specific outer-layer correction g_{out} to the boundary-layer configuration results in significant errors. This confirms that failing to distinguish between generalisable and dataset-specific flow physics is the reason why existing machine-learning models struggle to generalise beyond their training datasets. In figure 8(*b–e*), we compare the model incorporating f_{in} and h_{out} with the SA and SST models for their predictions of the mean flow and Reynolds shear stress. Figure 8(*b*,*d*) show the mean velocity profiles at $Re_\theta \approx 670$, which is below the training Reynolds number, and $Re_\theta \approx 8200$, which is above the training Reynolds number, while figure 8(*c*,*e*) show the corresponding Reynolds stresses. The results indicate that the present SR model provides more accurate predictions of the mean flow and Reynolds shear stress compared to the SA and SST models. The SA and SST models do not capture the mean flow at low Reynolds numbers as shown in figure 8(*b*). These findings demonstrate the following: (i) the inner-layer correction f_{in} generalises well across different types of wall-bounded flows; (ii) the outer-layer correction g_{out} is specific to channel flow and degrades performance when applied to boundary-layer flows; (iii) the outer-layer correction h_{out} generalises across different Reynolds numbers.

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

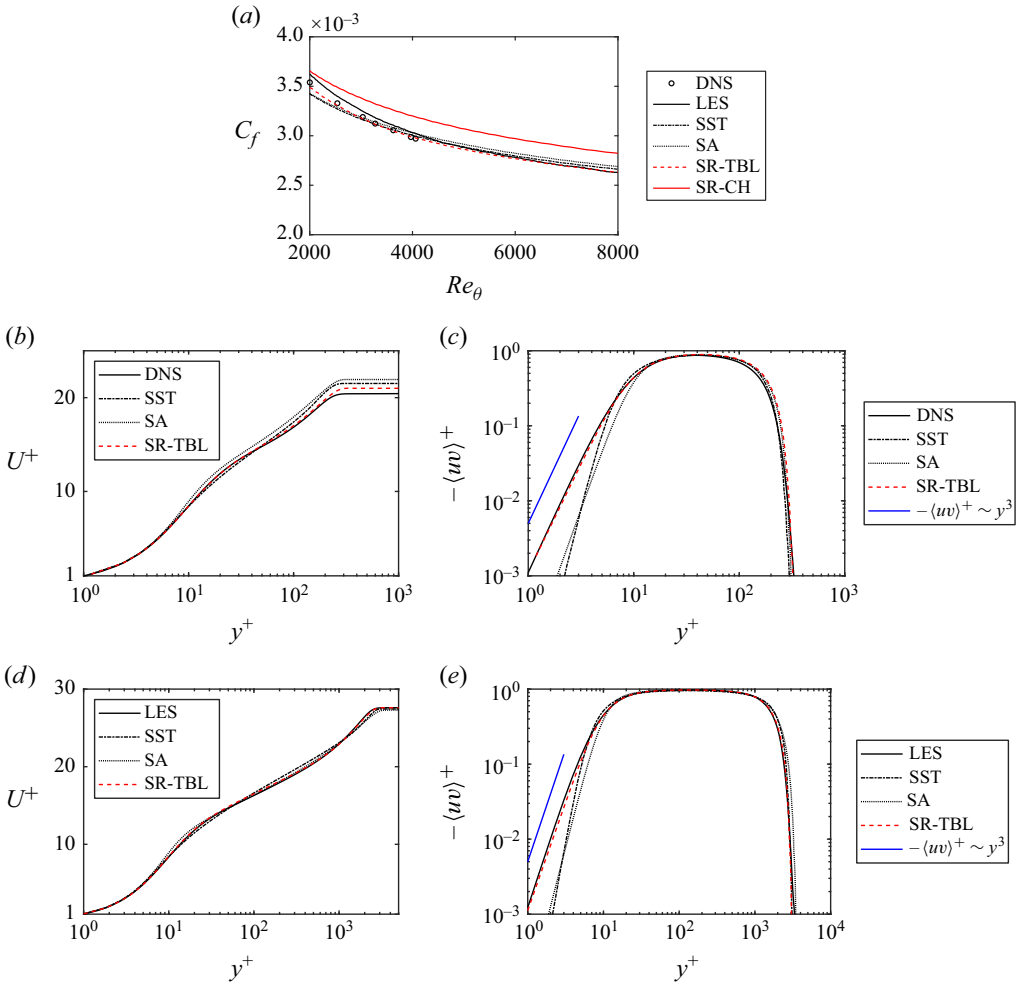


Figure 8. The ZPGTBL results. The present SR model has seen the configuration but not all the Reynolds numbers. (a) The skin friction coefficient: SR-TBL denotes the mixing length model with the inner-layer correction f_{in} and the TBL-specific outer-layer correction h_{out}; SR-CH denotes the mixing length model with the inner-layer correction f_{in} and the channel-specific outer-layer correction g_{out}. Both LES and DNS are of higher fidelity than RANS, and are reference data. (b, d) Mean velocity profile for $Re_\theta \approx 670$ and 8200. (c, e) Reynolds shear stress for $Re_\theta \approx 670$ and 8200.

Finaly, we consider PGTBLs. The flow condition corresponds to case APG-m18. The computational grid matches that of the DNS, with the first off-wall grid point located within $y^+ \leqslant 1$. The pressure-gradient condition in this case exceeds that of the training data; see table 2. The pressure-gradient correction γ_{PG} is incorporated into the mixing length model as a multiplicative correction. Figure 9 presents the results. Figure 9(a) shows the skin friction coefficient as a function of Re_θ. We compare the high-fidelity data, the SA and SST models, the model incorporating the inner-layer correction f_{in}, the outer-layer correction h_{out} and the pressure-gradient correction γ_{PG}, and the model with only the inner-layer correction f_{in} and outer-layer correction h_{out}. The results demonstrate that incorporating the pressure-gradient correction significantly improves the prediction of

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

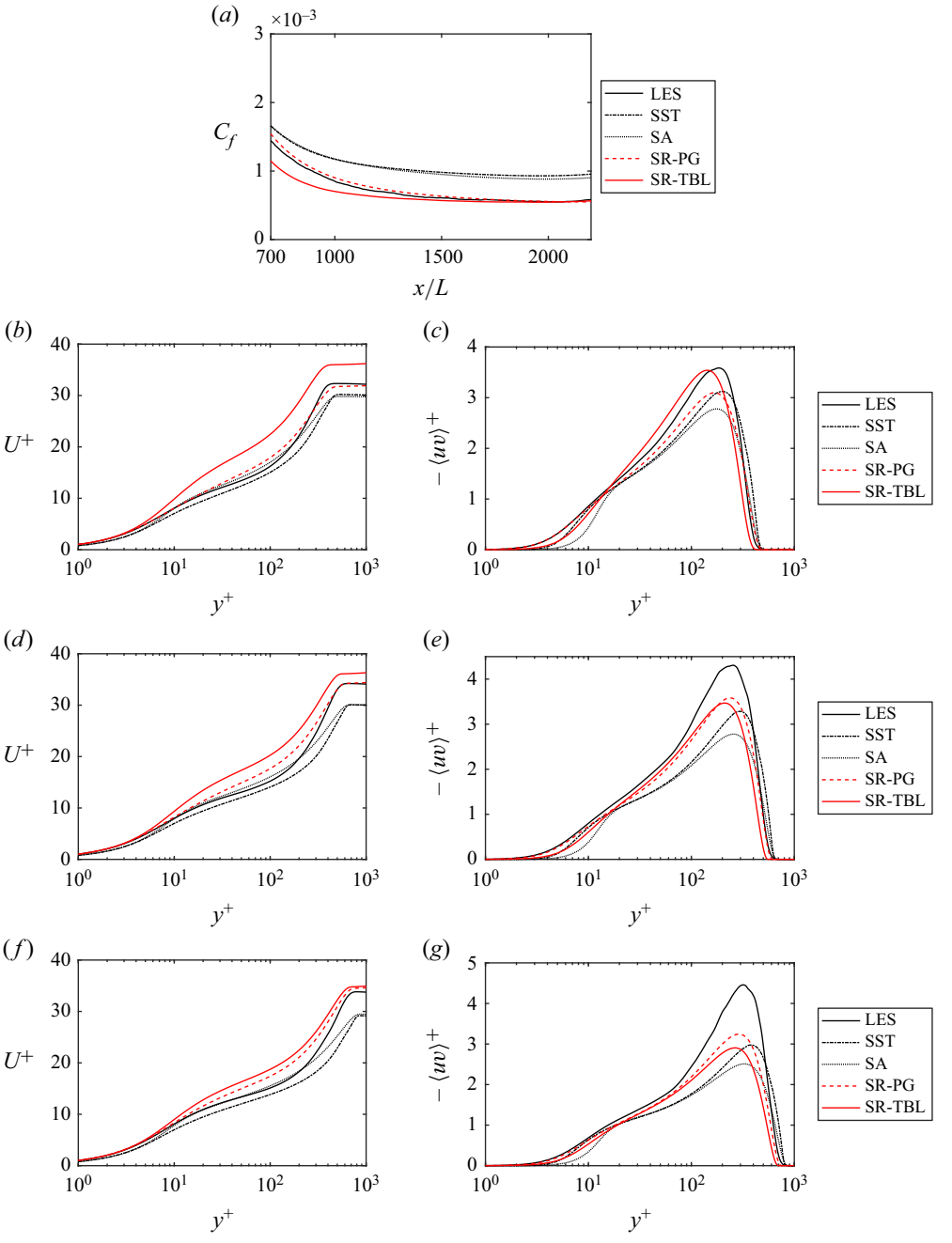


Figure 9. The PGTBL results. The present SR model has seen the configuration but not all the pressure-gradient conditions. (*a*) The skin friction coefficient: SR-PG denotes the SR model that contains the inner-layer correction f_{in}, the boundary-layer-specific outer-layer correction h_{out} and the PG-specific correction γ_{PG}; SR-TBL denotes the SR model that contains the inner-layer correction f_{in} and the boundary-layer-specific outer-layer correction h_{out}. (*b*,*d*,*f*) Velocity profiles at $x/L = 1000$, 1500, 2000. (*c*,*e*,*g*) Reynolds shear stress at $x/L = 1000$, 1500, 2000.

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

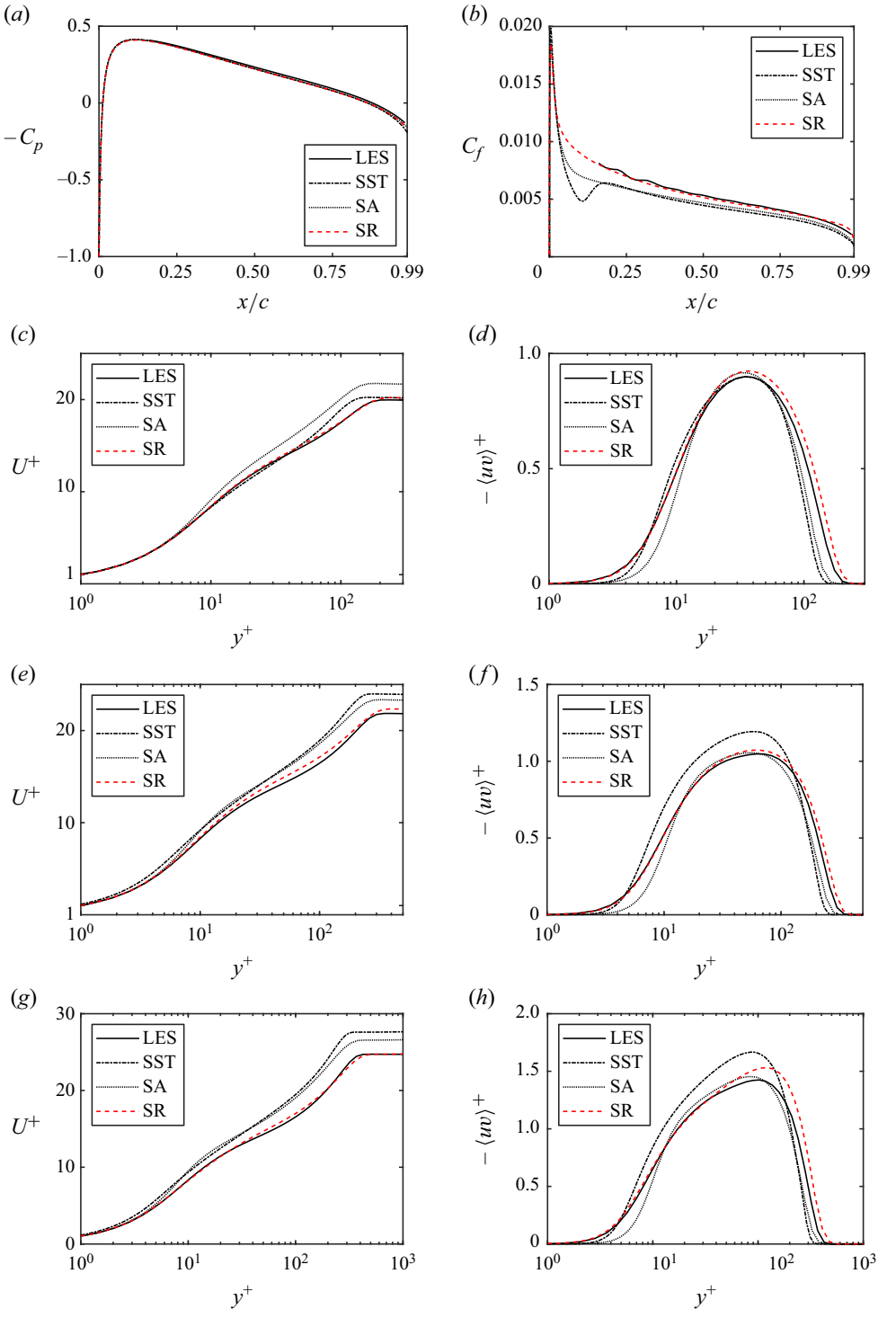


Figure 10. The NACA0012 results. The SR model contains the inner layer correction f_{in}, the boundary-layer specfic outer-layer correction h_{out}, and the pressure-gradient correction γ_{PG}: (*a*) pressure coefficient $-C_p$; (*b*) skin friction coefficient C_f. In the LES, the boundary layer was tripped at 10 % of the chord length from the leading edge on sides of the wing section. As a result, LES data before this location are not available. (c, e, g) Velocity profiles at $x/c = 0.3, 0.6, 0.9$. (d, f, h) Reynolds shear stress at $x/c = 0.3, 0.6, 0.9$.

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

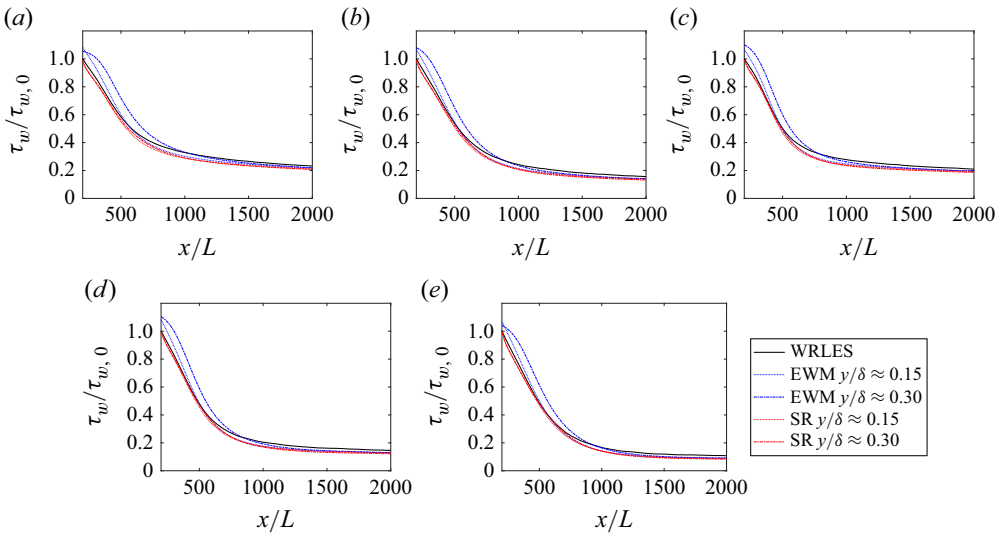


Figure 11. *A priori* test of the SR model for LES wall modelling. Here, the SR model contains the inner layer correction f_{in} and the pressure-gradient correction γ_{PG}: (*a*) APG-b1, (*b*) APG-b2, (*c*) APG-m13, (*d*) APG-m16, (*e*) APG-m18.

the skin friction coefficient. Figure 9(*b–g*) present the mean velocity profiles and Reynolds shear stress results at conditions both within and outside the training dataset. We see that the inclusion of the pressure-gradient correction improves velocity and Reynolds shear stress predictions as well. Errors in the Reynolds stresses are observed in all models, particularly at larger x/L. This is attributed to the limitation of eddy-viscosity models to capture history effects: modelling errors at upstream locations accumulate as the flow progresses downstream (Yang *et al.* 2024*b*). In all, these PGTBL results demonstrate: (i) the outer-layer correction h_{out} generalises to PGTBLs; (ii) the pressure-gradient correction γ_{PG} generalises across pressure gradient beyond the training dataset.

4.2. *Unseen geometry*

We validate the SR models on an unseen configuration, specifically the NACA0012 aerofoil. This case involves both boundary-layer physics and pressure-gradient effects. The set-up follows that on the TMR site (Rumsey 2019). As before, the first-layer cell height is below $y^+ < 1$, and the grid is sublayer resolved. The aerofoil is at angle of attack 0°, with Reynolds number $Re_c = 4\,00\,000$. The flow experiences a favourable pressure gradient near the leading edge, and an adverse pressure gradient near the trailing edge, making it a strong test case for evaluating the pressure-gradient correction γ_{PG}. Figure 10 presents the results. Figure 10(*a*, *b*) show the negative pressure coefficient $-C_p = 2(P_\infty - P)/(\rho U_\infty^2)$ and skin friction coefficient C_f, while figure 10(*c*, *e*, *g*) depict velocity profiles as functions of y^+ at $x/c = 0.3$, 0.6, 0.9. The corresponding Reynolds stresses are shown in figure 10(*d*, *f*, *h*). For comparison, results from the SST and SA models are also included. Several observations emerge. The conventional RANS models underpredict the skin friction coefficient and overpredict velocity. Additionally, their Reynolds shear stress predictions deviate significantly from the reference data. In contrast, the present SR model closely matches the high-fidelity reference data, with only minor errors in Reynolds shear

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

stress. These results validate both the effectiveness and generalisability of the symbolic models.

4.3. *Near-wall turbulence modelling for LES*

The discussion so far has been limited to the relatively simplistic mixing length model in the context of RANS modelling. While this may seem restrictive, as modern RANS models rely primarily on transport equations rather than local algebraic closures, the mixing length model remains the dominant approach for near-wall modelling in LES (Bose & Park 2018). In this subsection, we explore the directly applicability of our SR models for LES wall modelling. For simplicity, we follow Chen *et al.* (2022*a*) and conduct an *a priori* study. Specifically, the wall model equation is solved from the wall, i.e. $y/\delta = 0$ up to the wall model/LES matching location. We then compare the wall model results to those from wall-resolved LES (WRLES). We consider PGTBLs, specifically the five cases detailed in Bobke *et al.* (2017). The wall shear stress is computed using information at $y = 0.15\delta$, 0.30δ, typical wall-model/LES matching locations. We compare the mixing length model incorporating the f_{in} and γ_{PG} corrections with the conventional mixing length model in Piomelli (2008), which constutites the equilibrium wall model (EWM). The results are presented in figure 11. The SR model proves to be highly effective and robust, significantly improving wall shear stress predictions across all pressure-gradient conditions in PGTBLs, and yielding essentially the same results for the two matching locations. The EWM, on the other hand, is less accurate, and shows notable dependence on the matching location. Although the primary focus of this paper is on methodology and addressing the generalisability issue rather than developing new models, these results demonstrate the direct utility of the resulting models in Wall-modeled Large-Eddy Simulations.

5. Conclusions

Machine-learned turbulence models often struggle to generalise beyond their training datasets. Given the inherent complexity of turbulence, reliance on large datasets alone is not a practical solution. Previous efforts to enhance generalisability have focused on preserving symmetries, such as Galilean invariance, but these approaches have achieved only limited success, and the root cause of the lack of generalisability remains unclear.

This paper addresses the generalisability challenge. We first identify its root cause: the failure to distinguish between flow physics that generalise across configurations and those specific to training data. To resolve this issue, we decompose expressions obtained through symbolic regression (SR) into components that are universally applicable and those tied to the training dataset. When applying the model to flows beyond the training set – whether in the context of RANS or model retraining – only the generalisable components are retained. The SR is critical in the initial discovery process because it yields white-box models that are amenable to physical interpretation. A black-box model, on the other hand, would obscure the underlying structure and make it difficult to isolate generalisable versus data-specific physics.

Using this approach, we trained models on channel flow, ZPGTBLs and PGTBLs. This yielded generalisable corrections to the conventional mixing length model, including inner-layer and pressure-gradient corrections, as well as outer-layer adjustments specific to channel and boundary-layer flows. By applying these machine-learned mixing length closures, we achieved superior results in both familiar configurations (channel and boundary-layer flows) and an unfamiliar case (aerofoil flow), demonstrating the robustness

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

and generalisability of the proposed corrections. These results demonstrates that data-driven models can have generalisability similar to that of conventional models.

While this study focuses on the methodology rather than the resulting model, the utility of the SR models is still demonstrated in the context of LES wall modelling. Extending our methodology to transport-type models is also possible. The process follows the same three steps outlined in this paper: first, identifying relevant features such as y^+, y/δ, r, P/D and χ (Srivastava *et al.* 2024); second, applying SR to learn new terms for the transport equation; and third, parsing the learned terms into components that represent generalisable and training-specific flow physics, retaining only the former.

We believe that the results presented here pave the way for developing truly predictive machine-learned turbulence models capable of addressing complex turbulence challenges.

Funding. Z.Y., X.S. and W.Z. acknowledge financial support from the National Natural Science Foundation of China (no. 92152301, no. U2441211, no. U23B6009), the National Key Research and Development Programme of China (no. 2022YFB4300200) and Shaanxi Innovative Research Team of Artificial Intelligence for Fluid Mechanics (no. 2024RS-CXTD-16).

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

Appendix A. Symbolic regression algorithm

Symbolic regression is a machine-learning technique that searches for analytical expressions that best fit a given dataset. Unlike traditional regression, which assumes a predefined functional form (e.g. linear or polynomial), SR explores a vast space of mathematical expressions constructed from a predefined set of operators (e.g. addition, multiplication, exponentiation). In this work, we use the PySR library, which implements a variant of genetic programming optimised for SR. The core idea is to evolve a population of candidate expressions over multiple generations, progressively improving their accuracy and simplicity. The process begins with an initial population of randomly generated expressions. Each expression is evaluated using a fitness score, typically the mean squared error with respect to the training data. The top-performing expressions are retained and used to generate a new population through two key mechanisms: crossover that combines parts of two parent expressions to produce new candidate expressions, and mutation that randomly modifies parts of an expression to introduce diversity. Throughout the iterations, PySR maintains a 'hall of fame' set that stores the best-performing expressions across all populations and generations, ensuring that promising candidates are preserved.

We summarise the SR algorithm as used in this work in Algorithm 1. Our implementation uses 64 independent populations ('islands'), each containing 500 expressions. The algorithm runs for 300 iterations, and includes mechanisms for migration and parallel evolution across populations.

Appendix B. Repeatability of (3.2)

To assess the repeatability of (3.2), we trained the SR algorithm on three distinct groups of data. Under the settings summarised in table 1, SR consistently rediscovered the structure of the mixing length expression in (3.2), as indicated by the fraction in table 4. Because both f_{in} and g_{out} are functions of a single variable, we can visualise and compare them across different runs. As shown in figure 12(a,b), the resulting expressions yield nearly identical profiles, confirming the robustness of the SR procedure under the specified

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

Algorithm 1: Overview of the symbolic regression procedure using PySR

Input: Input features **X,** target output **Y**

Output: Set H of best expressions at each complexity level

Parameters: n_p: number of populations (default = 64)

L: population size per group (default = 500)

n_{iter}: number of iterations (default = 300)

α_H : probability of replacing an expression with one from H (default = 0.05)

α_M : probability of replacing an expression with one from another M_j (default = 0.05)

```
;                                                    /* Initialize populations */
for i ∈ {1, ..., n_p} do
    Initialize P_i with L random expressions of complexity 3;
    Initialize M_i ← Ø (stores best expressions in P_i);
end
Initialize H ← Ø;          /* Global best expressions (Hall of Fame) */
;                                                    /* Main evolution loop */
for n = 1 to n_iter do
    for i ∈ {1, ..., n_p}                          // Parallel over populations do
        P_i ← evolve(P_i, X, Y)                    // Selection, crossover, mutation
        foreach expression E in P_i do
            Simplify E;
            Optimize constants in E;
        end
        M_i ← most accurate expressions in P_i at each complexity;
        H ← best of H ∪ M_i at each complexity level;
        ;                                                          /* Migration */
        foreach expression E in P_i do
            if rand() < α_H then
                Replace E with a random expression from H;
            end
            if rand() < α_M then
                Replace E with a random expression from ∪_{j≠i} M_j;
            end
        end
    end
end
return H
```

configuration. Furthermore, we applied the discovered mixing length model to the CH-10 000 case. The predicted results exhibit excellent agreement across different SR runs, as shown in figure 12(c,d), further demonstrating the repeatability and generalisability of the obtained expression.

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

Dataset name	Data contained	Fraction
Group 1	CH-180, CH-1000	1/5
Group 2	CH-550, CH-2000	5/5
Group 3	CH-1000, CH-5200	5/5

Table 4. The training data contained in each group. The fraction given is the number of times that (3.2) is rediscovered divided by the number of total trials. The expression does not conform to (3.2) every time because CH-180 is at too low a Reynolds number and there are still low Reynolds number effects.

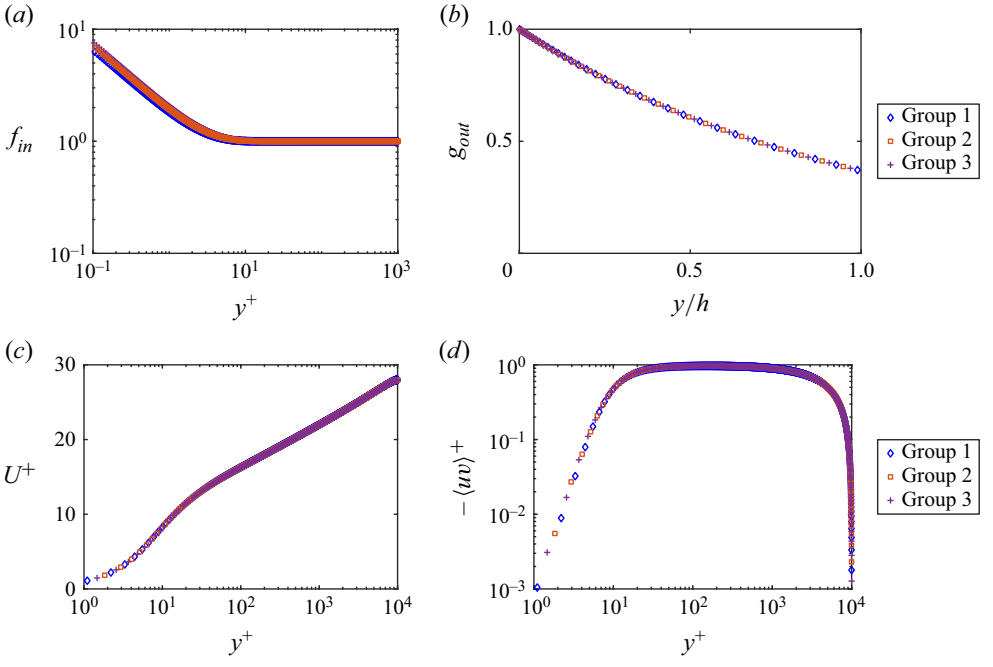


Figure 12. The results of repeated experiments: (*a*) f_{in} discovered across different runs; (*b*) g_{out} discovered across different runs; (*c*) predicted velocity profiles; (*d*) predicted Reynolds shear stress.

References

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.

Baldwin, B. & Lomax, H. 1978 Thin-layer approximation and algebraic model for separated turbulent flows. In *16th Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics.

Bin, Y., Huang, G., Kunz, R. & Yang, X.I.A. 2024 Constrained recalibration of Reynolds-averaged Navier–Stokes models. *AIAA J.* **62** (4), 1434–1446.

Bobke, A., Vinuesa, R., Örlü, R. & Schlatter, P. 2017 History effects and near equilibrium in adverse-pressure-gradient turbulent boundary layers. *J. Fluid Mech.* **820**, 667–692.

Bose, S.T. & Park, G.I. 2018 Wall-modeled large-eddy simulation for complex turbulent flows. *Annu. Rev. Fluid. Mech.* **50** (1), 535–561.

Cao, W., Song, J. & Zhang, W. 2024 A solver for subsonic flow around airfoils based on physics-informed neural networks and mesh transformation. *Phys. Fluids* **36** (2), 027134.

Cécora, R.-D., Radespiel, R., Eisfeld, B. & Probst, A. 2015 Differential Reynolds-stress modeling for aeronautics. *AIAA J.* **53** (3), 739–755.

Chen, P.E.S., Bin, Y., Yang, X.I.A., Shi, Y., Abkar, M. & Park, G.I. 2023*a* *A priori* screening of data-enabled turbulence models. *Phys. Rev. Fluids* **8** (12), 124606.

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

CHEN, P.E.S., LV, Y., XU, H.H.A., SHI, Y., YANG, X.I.A. 2022*a* LES wall modeling for heat transfer at high speeds. *Phys. Fluids* **7** (1), 014608.
CHEN, P.E.S., WU, W., GRIFFIN, K.P., SHI, Y. & YANG, X.I.A. 2023*b* A universal velocity transformation for boundary layers with pressure gradients. *J. Fluid Mech.* **970**, A3.
CHEN, Y., WANG, S. & LIU, W. 2022*b* Data-driven transition models for aeronautical flows with a high-order numerical method. *Aerospace* **9** (10), 578.
CHOI, H. & MOIN, P. 2012 Grid-point requirements for large eddy simulation: Chapman's estimates revisited. *Phys. Fluids* **24** (1), 011702.
CRANMER, M. 2023 Interpretable machine learning for science with PySR and SymbolicRegression.jl, arXiv preprint arXiv: 2305.01582.
CRANMER, M. 2024 PySR: high-performance symbolic regression in Python. Available at: https://astroautomata.com/PySR/api-advanced/ (Accessed: 2024-07-12).
DURBIN, P.A. 2018 Some recent developments in turbulence closure modeling. *Annu. Rev. Fluid. Mech.* **50** (1), 77–103.
EITEL-AMOR, G., ÖRLÜ, R. & SCHLATTER, P. 2014 Simulation and validation of a spatially evolving turbulent boundary layer up to $Re_\theta = 8300$. *Intl J. Heat Fluid Flow* **47**, 57–69.
GAO, C., ZHANG, W., LI, X., LIU, Y., QUAN, J., YE, Z. & JIANG, Y. 2017 Mechanism of frequency lock-in in transonic buffeting flow. *J. Fluid Mech.* **818**, 528–561.
GIMENEZ, J.M. & BRE, F. 2019 Optimization of RANS turbulence models using genetic algorithms to improve the prediction of wind pressure coefficients on low-rise buildings. *J. Wind Engng Ind. Aerodyn.* **193**, 103978.
GUNGOR, A.G., MACIEL, Y., SIMENS, M.P. & SORIA, J. 2016 Scaling and statistics of large-defect adverse pressure gradient turbulent boundary layers. *Intl J. Heat Fluid Flow* **59**, 109–124.
GUO, X., XIA, Z. & CHEN, S. 2021 Computing mean fields with known Reynolds stresses at steady state. *Theor. Appl. Mech. Lett.* **11** (3), 100244.
HE, Z., WU, C. & ZHANG, Y. 2024 A field inversion and symbolic regression enhanced Spalart–Allmaras model for airfoil stall prediction. *Phys. Fluids* **36** (6), 065111.
HOYAS, S., OBERLACK, M., ALCÁNTARA-ÁVILA, F., KRAHEBERGER, S.V. & LAUX, J. 2022 Wall turbulence at high friction Reynolds numbers. *Phys. Rev. Fluids* **7** (1), 014602.
HUANG, X.L.D., YANG, X.I.A. & KUNZ, R.F. 2019 Wall-modeled large-eddy simulations of spanwise rotating turbulent channels – comparing a physics-based approach and a data-based approach. *Phys. Fluids* **31** (12), 125105.
KAYS, W.M., CRAWFORD, M.E. & WEIGAND, B. 1980 *Convective Heat and Mass Transfer*. McGraw-Hill.
KLEWICKI, J.C. 2010 Reynolds number dependence, scaling, and dynamics of turbulent boundary layers. *ASME J. Fluids Engng* **132** (9), 094001.
KOU, J. & ZHANG, W. 2021 Data-driven modeling for unsteady aerodynamics and aeroelasticity. *Prog. Aerosp. Sci.* **125**, 100725.
LEE, M. & MOSER, R.D. 2015 Direct numerical simulation of turbulent channel flow up to $Re_\tau \approx 5200$. *J. Fluid Mech.* **774**, 395–415.
LI, J., BIN, Y., HUANG, G. & YANG, X.I.A. 2024 Enhancing generalizability of machine-learning turbulence models. In *AIAA SCITECH 2024 Forum*, p. 1574. American Institute of Aeronautics and Astronautics.
LI, J.-Q.J.L., YANG, X.I.A. & KUNZ, R.F. 2022 Grid-point and time-step requirements for large-eddy simulation and Reynolds-averaged Navier–Stokes of stratified wakes. *Phys. Fluids* **34** (11), 115125.
LING, J., KURZAWSKI, A. & TEMPLETON, J. 2016 Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* **807**, 155–166.
MENTER, F.R. 1994 Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA J.* **32** (8), 1598–1605.
NICKELS, T.B. 2004 Inner scaling for wall-bounded flows subject to large pressure gradients. *J. Fluid Mech.* **521**, 217–239.
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.
PIOMELLI, U. 2008 Wall-layer models for large-eddy simulations. *Prog. Aerosp. Sci.* **44** (6), 437–446.
POPE, S.B. 2001 Turbulent flows. *Meas. Sci. Technol.* **12** (11), 2020–2021.
PRANDTL, L. 1925 Uber die ausgebildete Turbulenz. *Z. Angew. Math. Mech.* **5**, 136–139.
ROMERO, S., ZIMMERMAN, S., PHILIP, J., WHITE, C. & KLEWICKI, J. 2022 Properties of the inertial sublayer in adverse pressure-gradient turbulent boundary layers. *J. Fluid Mech.* **937**, A30.
RUMSEY, C. 2019 Langley Research Center turbulence modeling resource. Available at: https://turbmodels.larc.nasa.gov/ (Accessed: 2024-07-12).

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

RUMSEY, C.L., COLEMAN, G.N. & WANG, LI 2022 In search of data-driven improvements to RANS models applied to separated flows. In *AIAA Scitech 2022 Forum*, p. 0937. American Institute of Aeronautics and Astronautics.

SARGHINI, F., DE FELICE, G. & SANTINI, S. 2003 Neural networks based subgrid scale modeling in large eddy simulations. *Comput. Fluids* **32** (1), 97–108.

SCHLATTER, P. & ÖRLÜ, R. 2010 Assessment of direct numerical simulation data of turbulent boundary layers. *J. Fluid Mech.* **659**, 116–126.

SHAN, X., LIU, Y., CAO, W., SUN, X. & ZHANG, W. 2023 Turbulence modeling via data assimilation and machine learning for separated flows over airfoils. *AIAA J.* **61** (9), 3883–3899.

SHAN, X., SUN, X., CAO, W., ZHANG, W. & XIA, Z. 2024 Modeling Reynolds stress anisotropy invariants via machine learning. *Acta Mechanica Sin.* **40** (6), 323629.

SHAN, X. & ZHANG, W. 2025 Data-driven adverse pressure gradient correction for turbulence model. *AIAA J.* **63**, 1–17.

SLOTNICK, J.P., KHODADOUST, A., ALONSO, J., DARMOFAL, D., GROPP, W., LURIE, E. & MAVRIPLIS, D. 2014 CFD Vision 2030 study: a path to revolutionary computational aerosciences. Tech. Rep. NASA/CR-2014-218178. NASA Langley Research Center.

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*, p. 439. American Institute of Aeronautics and Astronautics.

SRIVASTAVA, V., RUMSEY, C.L., COLEMAN, G.N. & WANG, L. 2024 On generalizably improving RANS predictions of flow separation and reattachment . In *AIAA SCITECH 2024 Forum*, p. 2520. American Institute of Aeronautics and Astronautics.

TAGHIZADEH, S., WITHERDEN, F.D., HASSAN, Y.A. & GIRIMAJI, S.S. 2021 Turbulence closure modeling with data-driven techniques: investigation of generalizable deep neural networks. *Phys. Fluids* **33** (11), 115132.

TANARRO, Á, VINUESA, R. & SCHLATTER, P. 2020 Effect of adverse pressure gradients on turbulent wing boundary layers. *J. Fluid Mech.* **883**, A8.

TANG, H., WANG, Y., WANG, T. & TIAN, L. 2023 Discovering explicit Reynolds-averaged turbulence closures for turbulent separated flows through deep learning-based symbolic regression with non-linear corrections. *Phys. Fluids* **35** (2), 025118.

TRACEY, B.D., DURAISAMY, K. & ALONSO, J.J. 2015 A machine learning strategy to assist turbulence model development. In *53rd AIAA Aerospace Sciences Meeting*, p. 1287. American Institute of Aeronautics and Astronautics.

VADDIREDDY, H., RASHEED, A., STAPLES, A.E. & SAN, O. 2020 Feature engineering and symbolic regression methods for detecting hidden physics from sparse sensor observation data. *Phys. Fluids* **32** (1), 015113.

VADROT, A., YANG, X.I.A. & ABKAR, M. 2023 Survey of machine-learning wall models for large-eddy simulation. *Phys. Rev. Fluids* **8** (6), 064603.

VAN DRIEST, E.R. 1956 On turbulent flow near a wall. *J. Aeronaut. Sci.* **23** (11), 1007–1011.

WANG, Z. & ZHANG, W. 2023 A unified method of data assimilation and turbulence modeling for separated flows at high Reynolds numbers. *Phys. Fluids* **35** (2), 025124.

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.

WU, C. & ZHANG, Y. 2023*a* Data-driven turbulence modeling using field inversion and deep symbolic regression. In *Aerospace Europe Conference 2023*. EUCASS.

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

WU, J.-L., XIAO, H. & PATERSON, E. 2018 Physics-informed machine learning approach for augmenting turbulence models: a comprehensive framework. *Phys. Rev. Fluids* **3** (7), 074602.

WU, X., ZUO, Z., MA, L. & ZHANG, W. 2024 Multi-fidelity neural network-based aerodynamic optimization framework for propeller design in electric aircraft. *Aerosp. Sci. Technol.* **146**, 108963.

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

YAN, C., ZHANG, Y. & CHEN, H. 2022 Data augmented turbulence modeling for three-dimensional separation flows. *Phys. Fluids* **34** (7), 075101.

YANG, X., ZHANG, W., ABKAR, M. & ANDERSON, W. 2024*a* Computational fluid dynamics: its carbon footprint and role in carbon reduction. *J. Renew. Sustain. Energy* **16** (5), 055906.

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press

YANG, X.I.A., CHEN, P.E.S., ZHANG, W. & KUNZ, R. 2024*b* Predictive near-wall modelling for turbulent boundary layers with arbitrary pressure gradients. *J. Fluid Mech.* **993**, A1.

YANG, X.I.A. & GRIFFIN, K.P. 2021 Grid-point and time-step requirements for direct numerical simulation and large-eddy simulation. *Phys. Fluids* **33** (1), 015108.

YANG, X.I.A., ZAFAR, S., WANG, J.-X. & XIAO, H. 2019 Predictive large-eddy-simulation wall modeling via physics-informed neural networks. *Phys. Rev. Fluids* **4** (3), 034602.

ZHANG, W., LI, X., YE, Z. & JIANG, Y. 2015 Mechanism of frequency lock-in in vortex-induced vibrations at low Reynolds numbers. *J. Fluid Mech.* **783**, 72–102.

ZHANG, W., XUHAO, P., JIAQING, K. & XU, W. 2024 Heterogeneous data-driven aerodynamic modeling based on physical feature embedding. *Chin. J. Aeronaut.* **37** (3), 1–6.

ZHANG, Z.J. & DURAISAMY, K. 2015 Machine learning methods for data-driven turbulence modeling. In *22nd AIAA Computational Fluid Dynamics Conference*, p. 2460. American Institute of Aeronautics and Astronautics.

ZHOU, D. & BAE, H.J. 2024 Sensitivity analysis of wall-modeled large-eddy simulation for separated turbulent flow. *J. Comput. Phys.* **506**, 112948.

ZHOU, D., WHITMORE, M.P., GRIFFIN, K.P. & BAE, H.J. 2023 Large-eddy simulation of flow over Boeing Gaussian bump using multi-agent reinforcement learning wall model. In *AIAA Aviation 2023 Forum*, p. 3985. American Institute of Aeronautics and Astronautics.

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.

ZHU, L., ZHANG, W., SUN, X., LIU, Y. & YUAN, X. 2021 Turbulence closure for high Reynolds number airfoil flows by deep neural networks. *Aerosp. Sci. Technol.* **110**, 106452.

https://doi.org/10.1017/jfm.2025.10332 Published online by Cambridge University Press