Nuclear Engineering and Design 457 (2026) 115034

Contents lists available at ScienceDirect

Nuclear Engineering and Design

journal homepage: www.elsevier.com/locate/nucengdes

Development of a data-driven turbulence model for coarse-grid CFD simulation of hydrogen dispersion in large spaces

Xiang Zhang *, Fabian Wiltschko, Aurelian Florin Badea, Xu Cheng

Institute for Applied Thermofluidics (IATF), Karlsruhe Institute of Technology (KIT), Kaiserstrasse 12, 76131 Karlsruhe, Germany

ARTICLE INFO

Keywords:
Hydrogen dispersion
Reynolds stress
Data-driven turbulence model
Machine learning
Coarse-grid CFD

ABSTRACT

Hydrogen safety in reactor containments is a critical issue, because hydrogen released during accidents may accumulate in large spaces and create the risk of combustion or explosion. Fine-grid CFD (FG-CFD) simulation can accurately predict hydrogen dispersion, but the high computational cost limits its application to transient processes in large spaces. In this study, a data-driven turbulence model was developed and used to construct an acceleration method, using coarse-grid CFD (CG-CFD) simulations. A high-fidelity FG-CFD database ($Gr = 3.02 \times 10^{10}$–1.90×10^{15}) was built, and a mapping guideline was proposed to transfer fine-grid data to coarse grids. Based on this database, a machine learning Reynolds stress (MLRS) model was developed using deep neural network (DNN) and integrated into OpenFOAM to replace traditional turbulence models. Validation results show that the proposed method significantly reduces computational cost while maintaining high accuracy in predicting hydrogen concentration fields and exhibits good generalization capability for unseen cases within the considered range of *Gr* numbers. This work provides a feasible approach for efficient and reliable analysis of hydrogen dispersion in large spaces to support containment safety evaluation.

1. Introduction

Due to the characteristics of low molecular weight, high diffusivity, and strong combustibility, hydrogen leakage in confined environments may cause severe safety issues. In nuclear engineering, for example, hydrogen released during accidents can accumulate in the containment, and uncontrolled diffusion may result in combustion or even explosion (Gharari et al., 2018). Therefore, accurate prediction of hydrogen distribution and concentration is essential for risk assessment and safety design in both nuclear power plants and hydrogen energy systems.

Computational fluid dynamics (CFD) has become a powerful tool for analyzing hydrogen dispersion, as it can provide detailed three-dimensional flow and species transport information. Numerous studies have successfully applied CFD to investigate buoyancy-driven natural circulation, with applications to nuclear containments (Kampili et al., 2021), refueling stations (Ma et al., 2024), new-energy vehicles (Hou et al., 2023), and residential garages (Hajji et al., 2015). However, for reliable prediction of local hydrogen concentrations, sufficiently fine grids are required. Fine-grid CFD (FG-CFD) simulations can resolve buoyancy-driven turbulent mixing caused by density differences between gas mixtures, thereby providing high-fidelity concentration fields. However, the large computational domain, strong turbulence and transient characteristics of these problems need a large number of cells and small time steps, leading to prohibitive computational costs for large-scale or long-duration situations, and limiting the possibility of three-dimensional transient analyses (Malakhov et al., 2020).

With the development of artificial intelligence, machine learning (ML) has been a promising approach to accelerate CFD simulations, with major research directions including ML-accelerated solvers (Ajuria-Illaramendi et al., 2020), reduced-order modeling (Halder et al., 2024), super-resolution reconstruction (Fukami et al., 2019), and ML-assisted turbulence closures (Duraisamy, 2021). Among these, the data-driven surrogate models for turbulence, trained on high-fidelity data, have been confirmed to be an effective approach. These models aim to replace parts of traditional turbulence models with ML surrogates to capture complex flow physics and improve efficiency. For example, Tracey et al. (2015) explored the potential of using a feed-forward neural network to approximate the Spalart-Allmaras turbulence model, showing that the trained network could accurately predict flow behaviors under various conditions. Liu et al. (2021) proposed an iterative ML-RANS framework in which a neural network was trained to predict dimensionless eddy viscosity from extended input features, ensuring consistent training prediction procedures and improved

* Corresponding author.
E-mail address: xiang.zhang@kit.edu (X. Zhang).

https://doi.org/10.1016/j.nucengdes.2026.115034
Received 14 February 2026; Received in revised form 29 April 2026; Accepted 31 May 2026
Available online 8 June 2026

Nomenclature

e	mean value
g	gravity /m^2·s^{-2}
Gr	Grashof number
H	height /m
k	turbulent kinetic energy /m^2·s^2
K	number of the nearest-neighbors
L	length /m
n	diamentional
n_i	unit vector normal to the wall
N	number of data points
p	mesh point
P	pressure /Pa
R	Reynolds stress /m^2·s^{-2}
S	modulus of mean strain rate tensor /s^{-1}
S_{ij}	mean strain rate tensor /s^{-1}
t	time /s
u	velocity /m·s^{-1}
W	width /m
y_i	actual value

Greek letters

ρ	density /kg·m^{-3}
μ	viscosity /kg·m^{-1}·s^{-1}
Ω	modulus of mean rotation rate tensor /s^{-1}
Ω_{ij}	mean rotation rate tensor /s^{-1}
ν_t	turbulent kinematic viscosity /m^2·s^{-1}
ε	turbulent dissipation rate /m^2·s^{-1}
σ	standard deviation

interpolation capability across different flow cases. Maulik et al. (2021) introduced a surrogate model for turbulent eddy viscosity in steady-state RANS simulations, achieving 5 times speed up on backward-facing step flow calculations. To address transient thermal-hydraulic problems, Liu et al. (2022a) developed a coarse-grid (CG) turbulence model combining convolutional neural networks with long short-term memory networks to predict time-resolved eddy viscosity fields in thermal stratification analyses. However, most of these studies focused on predicting a scalar turbulent eddy viscosity under the linear eddy-viscosity approximation, which assumes an isotropic relationship between the Reynolds stresses and the mean strain rate. As a result, directional dependencies in the Reynolds stress tensor are neglected, limiting the ability of these models to capture turbulence anisotropy in complex flows.

Buoyancy-driven turbulent natural circulation and hydrogen mixing with multiple gases are typical anisotropic turbulent flows. In such cases, the eddy viscosity assumption fails to represent anisotropy of turbulence accurately. Several studies have attempted to overcome this limitation by developing data-driven surrogate models that explicitly account for anisotropy. For instance, McConkey et al. (2022) proposed a physics-informed deep learning framework that predicting both the eddy viscosity and the nonlinear part of the Reynolds stress tensor, Berrone and Oberto (2022) developed a neural-network RANS closure directly predicting the Reynolds stress tensor, and Chen and Deng (2024) introduced a data-driven closure combining linear Boussinesq terms with nonlinear residual corrections using neural networks. However, although achieving notable improvements in anisotropic turbulence modeling, these studies have been mostly trained and validated on special cases such as square duct and periodic hill flows, which differ significantly from buoyancy-driven natural circulation and multiple gas mixing conditions in driving mechanisms, species transport, and turbulence anisotropy characteristics. Therefore, it is necessary to investigate the acceleration method for CFD simulation of hydrogen dispersion in large spaces and develop data-driven turbulence models for these more complex flows.

In this work, a data-driven turbulence model for coarse-grid CFD (CG-CFD) simulations is developed to improve the efficiency and accuracy of hydrogen dispersion analysis in large enclosures. Section 2 introduces the general acceleration framework. Section 3 describes the construction of a comprehensive FG-CFD database covering a wide range of dimensionless parameters and the validation of the turbulence model used in FG-CFD. Section 4 presents a mapping guideline for transferring FG-CFD data onto coarse meshes and develops a machine learning Reynolds stress model (MLRS). Finally, in Section 5, the data-driven turbulence model, which employs the MLRS model, is implemented in OpenFOAM, where the accuracy and computational efficiency of CG-CFD simulations are evaluated.

2. Acceleration method

To overcome the high computational cost of FG-CFD in large-scale hydrogen diffusion simulations, in this study, an acceleration methodology is attempted by integrating OpenFOAM for CFD simulations with TensorFlow (Abadi et al., 2016) for machine-learning model construction. As illustrated in Fig. 1, the method consists of three key components. First, high-fidelity FG-CFD simulations are conducted in OpenFOAM to generate a comprehensive training and validating database of flow fields, including velocity, velocity gradient and Reynolds stress components. It is worth noting that the turbulence model, which can reflect the turbulence anisotropy, will be employed in FG-CFD simulations. The range of the boundary conditions is designed by the physical analysis of non-dimensional numbers. Second, the training database is processed and employed to train a Machine Learning Reynolds Stress model (MLRS) in TensorFlow by DNN. This stage incorporates an efficient mapping method to transfer fine-grid CFD results onto coarse grids, enabling the MLRS model to learn the relationship between flow features and the Reynolds stress tensor. The trained model is exported in a format compatible with OpenFOAM for subsequent implementation. A data-driven turbulence model, which calculates the Reynolds stress tensor by MLRS model, is incorporated into the physics-based solving process within OpenFOAM. Third, the data-driven turbulence model is integrated into CG-CFD, replacing traditional turbulence models. The CG-CFD simulations are evaluated in terms of prediction accuracy and computational efficiency compared to the reference FG-CFD results.

3. FG-CFD simulations for database generation

3.1. Turbulence model and numerical method

The momentum equation of the Reynolds Averaged Navier-Stokes equations is written as:

$$\frac{\partial(\rho\overline{u}_i)}{\partial t}+\frac{\partial\left(\rho\overline{u}_i\overline{u}_j\right)}{\partial x_j}=-\frac{\partial\overline{p}}{\partial x_i}+\frac{\partial}{\partial x_j}\left(\mu\frac{\partial\overline{u}_i}{\partial x_j}-\rho\overline{u_i'u_j'}\right) \tag{1}$$

where $\overline{u_i'u_j'}$ is the Reynolds-stress tensor, which includes six unknown nonlinear quantities when solving a three-dimensional turbulence problem.

Lien et al. (1996) proposed a non-linear eddy-viscosity model to consider the non-linear relations between the Reynolds stress and strain rate as well as vorticity tensors, enabling to describe turbulence anisotropy. The Lien Cubic k-ε model calculates the Reynolds stress with quadratic and cubic terms:

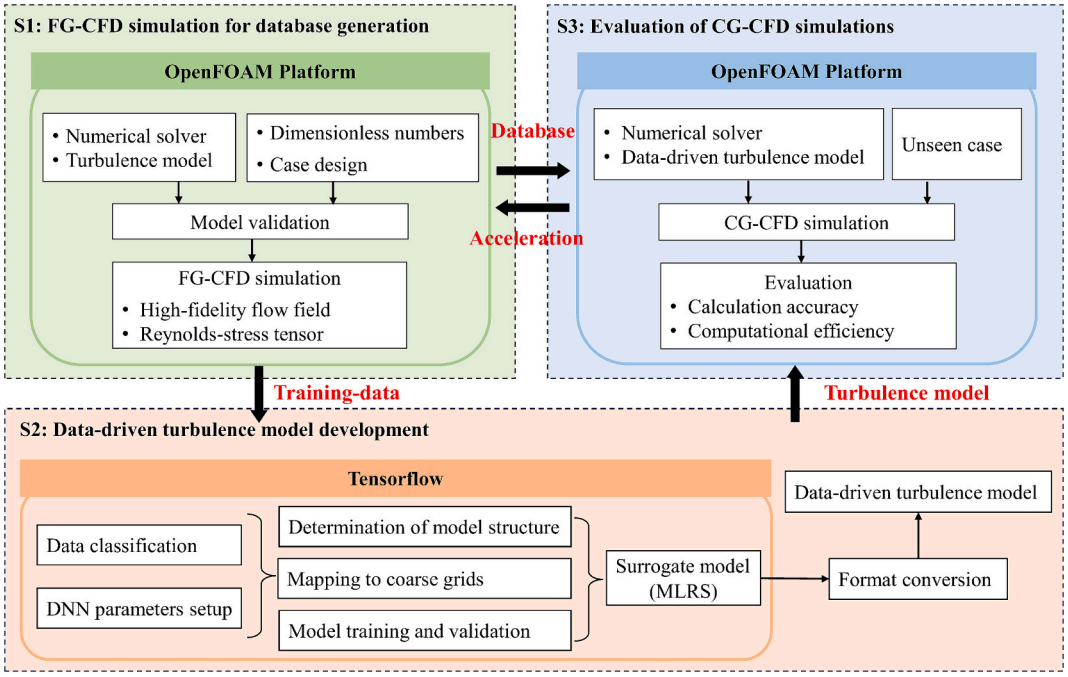


Fig. 1. The framework of the method.

$$\frac{\overline{u_i'u_j'}}{k} = \frac{2}{3}\delta_{i,j} - \frac{\nu_t}{k}S_{ij} + C_1\frac{\nu_t}{\varepsilon}\left(S_{ik}S_{kj} - \frac{1}{3}\delta_{i,j}S_{kl}S_{kl}\right)$$
$$+C_2\frac{\nu_t}{\varepsilon}\left(\Omega_{ik}S_{kj} + \Omega_{jk}S_{ki}\right) + C_3\frac{\nu_t}{\varepsilon}\left(\Omega_{ik}S_{jk} - \frac{1}{3}\delta_{i,j}\Omega_{kl}\Omega_{kl}\right) \quad (2)$$
$$+C_4\frac{\nu_t k}{\varepsilon^2}\left(S_{ki}\Omega_{lj} + S_{kj}\Omega_{li}\right)S_{kl} + C_5\frac{\nu_t k}{\varepsilon^2}\left(S_{kl}\Omega_{kl} - S_{kl}\Omega_{kl}\right)S_{ij}$$

where S_{ij} and Ω_{ij} are strain rate tensor and vorticity tensor, separately, that are defined as:

$$\begin{cases} S_{ij} = \dfrac{\partial u_i}{\partial x_j} + \dfrac{\partial u_j}{\partial x_i} \\ \Omega_{ij} = \dfrac{\partial u_i}{\partial x_j} - \dfrac{\partial u_j}{\partial x_i} \end{cases} \quad (3)$$

The turbulent kinematic viscosity ν_t is calculated by:

$$\nu_t = C_\mu f_\mu \frac{k^2}{\varepsilon} \quad (4)$$

The model coefficients of C_μ and C_1 to C_5 are summarized in Table 1.

The modulus of the mean strain rate tensor S and the mean rotation rate tensor Ω are calculated by:

$$\begin{cases} S = \dfrac{k}{\varepsilon}\sqrt{\dfrac{1}{2}S_{ij}S_{ij}} \\ \Omega = \dfrac{k}{\varepsilon}\sqrt{\dfrac{1}{2}\Omega_{ij}\Omega_{ij}} \end{cases} \quad (5)$$

The damping function is calculated by:

$$f_\mu = \left[1 - \exp(-0.0198y^*)\right]\left(1 + \frac{5.29}{y^*}\right) \quad (6)$$

Table 1
Model coefficients in Lien model.

C_μ	C_1	C_2	C_3	C_4	C_5
$\dfrac{0.667}{1.25 + S + 0.9\Omega}$	$\dfrac{3/4}{(1000 + S^3)}$	$\dfrac{15/4}{(1000 + S^3)}$	$\dfrac{19/4}{(1000 + S^3)}$	$-10C_\mu^2$	$-2C_\mu^2$

where

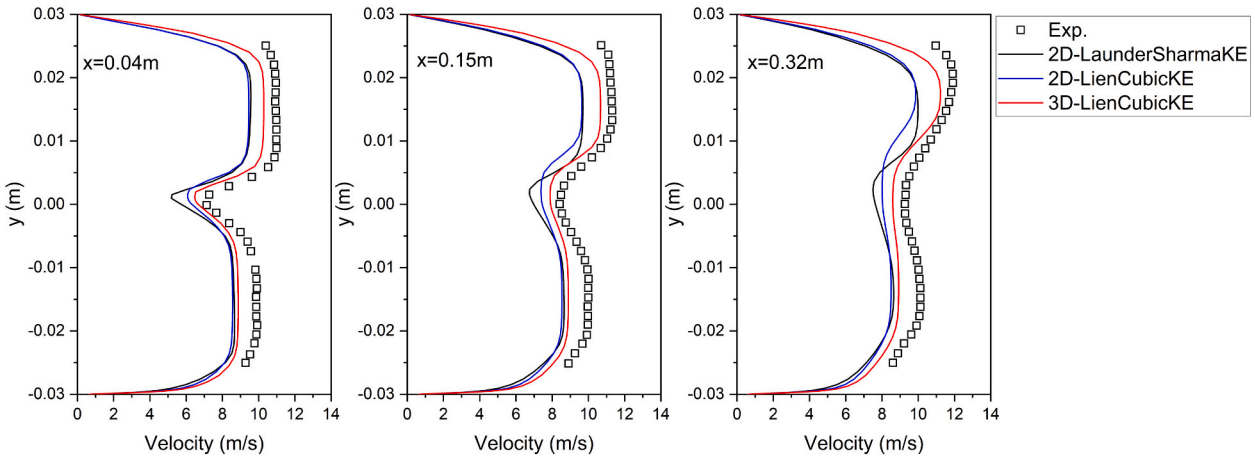
$$y^* = y\sqrt{k}/\varepsilon \quad (7)$$

The MulticomponentFluid solver in OpenFOAM was employed for the simulations. This pressure-based solver can be used for transient simulations of multi-species, compressible flows. It can resolve turbulent flows with strong coupling between species transport and momentum equations and is therefore particularly suitable for non-reacting flows involving mixing processes of different gases. For transient computations in this study, the PIMPLE algorithm is implemented to ensure robust pressure-velocity coupling and improved numerical convergence.

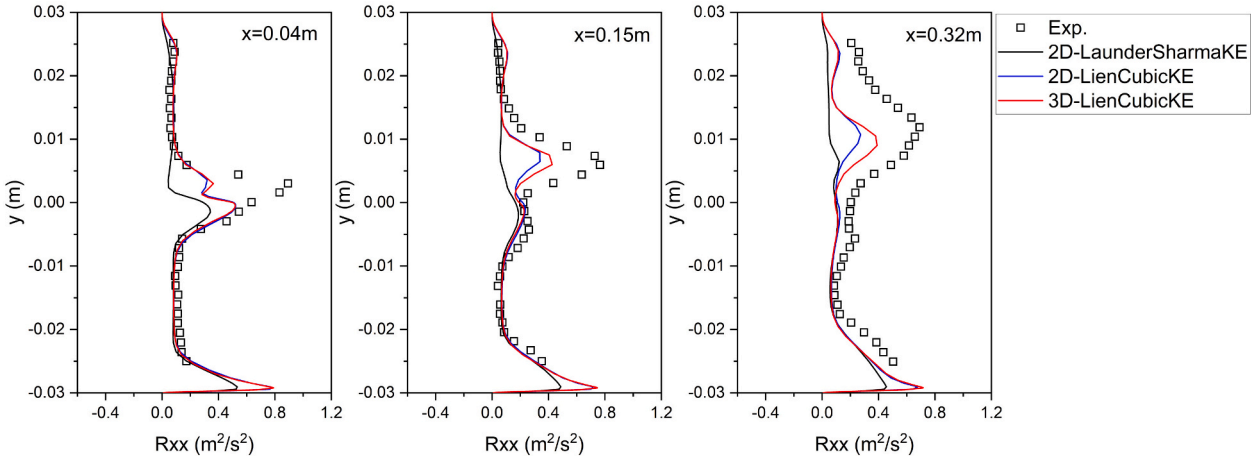
3.2. Model validation for FG-CFD

In order to assess the performance of the Lien Cubic k-ε turbulence model, numerical results were validated against the gas mixing experiment data obtained by the HOrizontal Mixing Experiment in a Rectangular channel (HOMER) (Krohn et al., 2015). In the HOMER experiments, two separated gases (nitrogen and helium) were introduced into a rectangular mixing channel with square cross-section via a horizontal splitter plate. The flow field in the vertical mid-plane was measured using Particle Image Velocimetry (PIV), providing experimental data of velocity as well as Reynolds stress components.

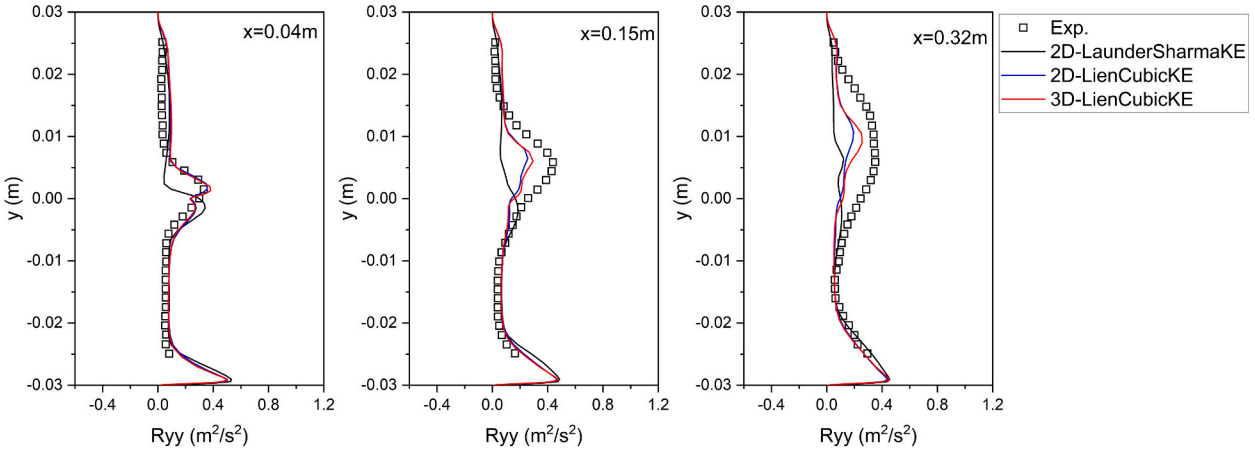
In the simulations, the Lien Cubic k-ε model, based on non-linear eddy-viscosity approach, was employed for both a three-dimensional geometry and a simplified two-dimensional geometry. For comparison, the Launder-Sharma k-ε model (Launder and Sharma, 1974), which uses a linear eddy-viscosity approach, was also applied. The calculated velocity and Reynolds stresses were compared with the experimental measurements, as shown in Fig. 2. The results indicate that the Lien Cubic k-ε model provides a significantly better prediction than the Launder-Sharma k-ε model, due to its improved capability to account for turbulence anisotropy. Besides, the Lien Cubic k-ε model performs better in the 3D geometry, compared to the 2D case, particularly in reproducing the asymmetric development of the mixing field and the distribution of turbulent stresses, which demonstrates that the 2D simplifications are not acceptable for gas mixing cases with strong



(a) Velocity



(b) Reynolds stress Rxx



(c) Reynolds stress Ryy

Fig. 2. Validation results against HOMER experiment.

anisotropic turbulence.

As a result, the following FG-CFD simulations will adopt the Lien Cubic k-ε turbulence model and be performed in a full 3D geometry.

3.3. Case description

In this study, a cubic enclosure with equal length, width, and height was designed to investigate the diffusion of hydrogen in a large space, as shown in Fig. 3. The gas inlet and outlet are located at diagonal positions on the same side wall of the enclosure. Both the inlet and outlet are square openings of $0.5 \times 0.5\ m^2$. At the initial time $t = 0$, a mixture gas of hydrogen and steam with a hydrogen mass fraction of 0.4 is injected from the bottom opening at an initial velocity of 1 m/s, normally on the inlet surface. The initial gas within the large space is air with an initial velocity of 0 m/s.

Grashof number (Gr) represents the ratio of buoyancy forces to viscous forces in a fluid, and it is the most important dimensionless number in buoyancy-driven natural convection. Thus, the Gr number, defined in Eq. (8), was used for the natural circulation of hydrogen in a large enclosure. As summarized in Table 2, 13 cases were designed for the FG-CFD simulations. Due to the Gr number is a function of fluid properties and the characteristic length, Cases 1–12 were used to generate training data for the mapping process by varying pressure, temperature, and enclosure height to cover a wide range of Gr numbers, from 3.02×10^{10} to 1.90×10^{15}. Case 13 will be employed for the validation of the CG-CFD simulations, as an independent working condition with its Gr number falling within the range of the training dataset.

$$\mathrm{Gr} = \frac{\Delta \rho g H^3}{\rho_{mix} \upsilon^2} \tag{8}$$

3.4. Mesh sensitivity analysis

Based on Case 7, a grid sensitivity study was performed for the FG-CFD simulations. Six different grids were tested, as summarized in Table 3, where Grids 1–5 are uniformly structured meshes. As shown in Fig. 4, Grid 4 achieves a grid independent solution. However, the large number of cells in Grid 4 results in extremely high computational cost, making it unsuitable for simulating all 13 working conditions. Further analysis shows that Grid 2 already provides a grid independent solution in the central region of the domain, while the near-wall regions require a finer resolution comparable to that of Grid 4. According to this observation, an improved grid (Grid 6) was constructed by combining the Grid 2 cell size in the central area with refined near-wall cells according to the ratio between Grid 2 and Grid 4. As shown in Fig. 4, Grid 6 achieves a grid-independent solution nearly identical to Grid 4, while significantly reducing the total cell numbers and improving computational efficiency. Therefore, the meshing scheme of Grid 6 was used for all FG-CFD simulations to generate a comprehensive database for machine learning.

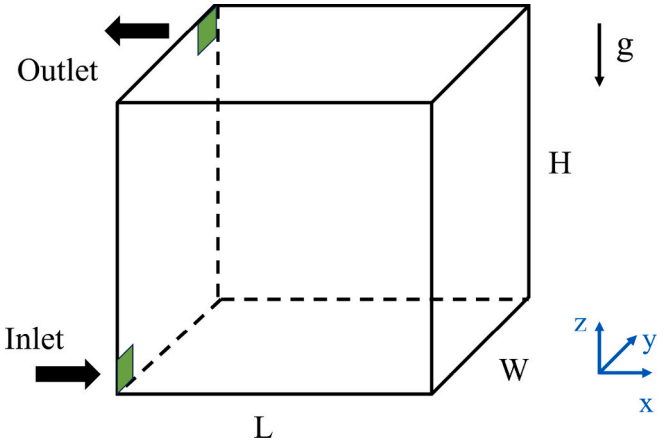


Fig. 3. Geometry of simulations.

Table 2
Boundary conditions.

	Pressure (MPa)	Temperature (K)	Height (m)	Simulation time (s)	Gr number
Case 1	0.1	550	5	3000	3.02×10^{10}
Case 2	0.5	550	5	3000	7.42×10^{11}
Case 3	1	550	5	3000	2.91×10^{12}
Case 4	5	550	5	3000	5.92×10^{13}
Case 5	10	640	5	3000	1.33×10^{14}
Case 6	15	640	5	3000	2.37×10^{14}
Case 7	0.1	550	10	20,000	4.38×10^{11}
Case 8	0.5	550	10	20,000	5.94×10^{12}
Case 9	1	550	10	20,000	2.33×10^{13}
Case 10	5	550	10	20,000	4.73×10^{14}
Case 11	10	640	10	20,000	1.07×10^{15}
Case 12	15	640	10	20,000	1.90×10^{15}
Case 13	2	640	10	20,000	8.91×10^{13}

Table 3
Mesh Information.

No.	Number of cells	Grid distribution	Core-hours
Grid 1	216,000	Uniform	2368
Grid 2	512,000	Uniform	7360
Grid 3	1,000,000	Uniform	23,400
Grid 4	1,728,000	Uniform	30,888
Grid 5	2,744,000	Uniform	36,396
Grid 6	681,472	Non-uniform	13,632

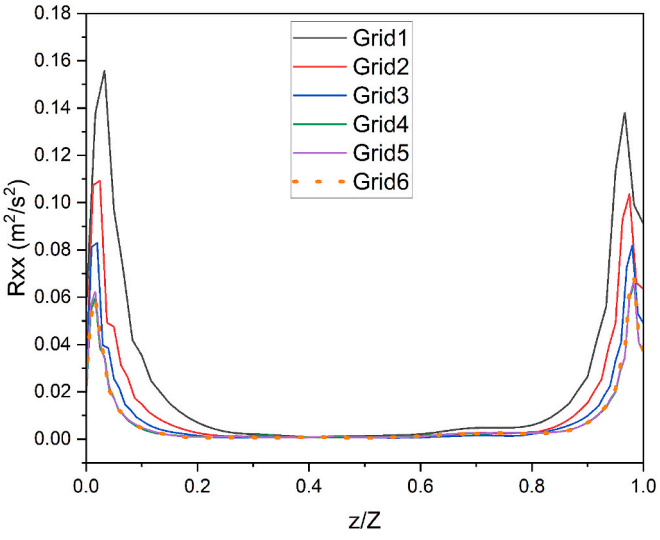


Fig. 4. Mesh sensitivity analysis.

4. Data-driven turbulence model development

4.1. Determination of the model structure

In this study, a machine-learning-based surrogate model for Reynolds stress will be developed. As described in Section 3.1 for the equations of Reynolds stress and the turbulence closure of the Lien Cubic k-ε model, a total of 13 flow-field parameters were selected as input features, including density (ρ), velocity (U_i), and velocity gradients $\left(\partial U_i/\partial x_j\right)$. To further account for near-wall effects, 4 additional parameters were introduced: the distance from the local grid point to the nearest wall (l) and the components of the unit direction vector pointing toward the wall (n_i). As a result, the MLRS model employs 17 input parameters in total and predicts the 6 output components of the Reynolds stress, as summarized in Eq. (9).

$$\mathbb{M}_1 : \rho(p), U_i(p), \frac{\partial U_i(p)}{\partial x_j}, l(p), n_i(p) \rightarrow R_{ij}(p) \tag{9}$$

where p is the mesh point.

The selected input features were determined based on their physical relevance to buoyancy driven turbulent flow and Reynolds stress transport. The density and velocity components characterize the local flow state and buoyancy behavior, while the velocity gradient components provide information associated with strain and rotation effects, which are directly related to turbulence production and anisotropy. Since the strain rate tensor and rotation rate tensor can be constructed from the velocity gradient tensor, the inclusion of velocity gradient components allows the model to access important local kinematic information relevant to Reynolds stress modeling. In addition, the wall distance and wall direction parameters were included to represent near wall anisotropy effects. However, it should be noted that the present feature sets are selected based on physical considerations rather than a systematic feature ablation study. Therefore, the relative contribution of each individual input feature to the prediction accuracy was not quantified in the present work. In addition, it is also not formulated in an invariant tensor-basis framework as in Ling et al. (2016), which remains a limitation of the present study.

4.2. Data preprocessing: mapping

In building the ML database, the FG-CFD results obtained from high-resolution nonuniform CFD grids are unsuitable directly used for model training. This is because that, in a nonuniform mesh the data points do not represent equal spatial weight, whereas neural network training assumes that each sample contributes equally. Besides, the fine-mesh resolution is inconsistent with the intended application, which aims to develop a constitutive relation applicable in coarse-grid 3D CFD simulations. If training is done on a high-resolution mesh, the resulting model cannot be directly applied to the coarse-mesh CFD framework, especially several gradient parameters are employed as inputs. What's more, the large number of cells in the fine-grid setup leads to excessive computational cost and hinders the efficiency of model training. To overcome these limitations, the CFD results need to be mapped from the original nonuniform fine-grid onto a uniform coarse-grid, which provides data of consistent weight, compatible resolution, and manageable size for subsequent ML model development.

4.2.1. Guideline for mapping

Given the critical role of mapping-parameter selection, in particular the fine-to-coarse grid ratio ($Ratio = L_{FG}/L_{CG}$) and the number of nearest neighbors (k), a theoretical guideline is developed in this study to provide a consistent and robust framework for selecting these parameters across different simulations.

For an n-dimensional simulation with a given mapping ratio the number of nearest neighbor cells k is determined according to Eq.(10). In addition, a criterion is introduced to define the valid region of influence for each coarse cell. For example, in three-dimensional simulations, a spherical region is constructed around the center of each coarse grid cell, with the diameter equal to the cell's diagonal length. Only the fine grid cells located inside this sphere are considered as effective contributors for the mapping. The corresponding radius is calculated by Eq.(11). The mapped values for each coarse grid cell are obtained as a distance-weighted average of its effective neighbors.

$$k = (\lfloor Ratio + 0.5 \rfloor + 1)^n \tag{10}$$

$$L_{\text{limit}} = \sqrt{n}\,\frac{L_{CG}}{2} \tag{11}$$

Fig. 5 presents the sensitivity analysis results of different mapping ratios (Ratio = 1.6, 2, and 2.4) by employing this mapping guideline for Case 7. Since velocity gradient is one of the key parameters for calculating Reynolds stress, it is listed here as a representative. The results show weak sensitivity to variations in the mapping ratio, confirming that the proposed guideline provides stable and physically consistent mapping for different grid configurations.

4.2.2. Scalable kNN-based mapping implementation

In the previous study, the mapping process was performed by using the KNNImputer function in scikit-learn ML library (Liu et al., 2025). It is based on the k-nearest-neighbors (kNN) concept of interpolation, which identifies k nearest points from the FG to each cell in CG and obtains interpolations. However, the original KNNImputer function needs a long computational time. To illustrate the magnitude of the cost, even with k = 5 in Case 7, which is substantially smaller than the calculated value of our guideline (k = 27 for Ratio = 2), mapping only a single time-step required more than 50 h. Using the guideline value would increase the cost further, making it impractical to generate a large amount of data for ML applications.

In this study, a more efficient mapping was implemented based on the interpolation concept of kNN. Specifically, due to the most of the computational cost in the original KNNImputer function was spent on locating the k nearest neighbors, a KD-Tree algorithm was employed to rapidly identify the nearest neighbors and the values at the target points were obtained through distance-weighted averaging of the neighboring FG-CFD data. Compared with the direct use of the KNNImputer function in scikit-learn, this implementation retains the same kNN interpolation principle but provides much greater flexibility and computational efficiency. In addition, the parallel processing of multiple time-steps is supported in our mapping implementations. For example, on the 64-

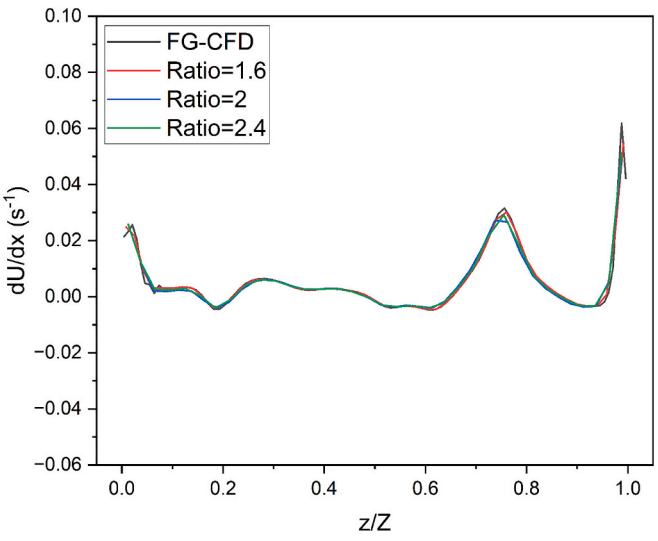


Fig. 5. Sensitivity study of mapping ratio.

thread workstation at the Institute for Applied Thermofluidics (IATF), up to 64 time-steps can be mapped simultaneously, which significantly reduces the total computational cost.

As summarized in Table 4, both mapping implementations employ 5 nearest neighbors for the mapping to compare the time costs. The results demonstrate that the current kNN-based mapping implementation with KD-Tree accelerating and parallel calculating reduces the mapping time by orders of magnitude compared with the original implementations, thereby making large-scale mapping feasible and greatly enhancing the efficiency.

4.3. MLRS model training

The MLRS model was constructed and trained in Python using the widely adopted TensorFlow library, which supports the implementation of complex data-driven approaches. A DNN architecture was manually designed based on prior knowledge and empirical evaluation. The configuration of 3 hidden layers with 512 neurons each was selected to provide sufficient nonlinear mapping capability for the problem while maintaining stable training. It should be noted that the present study does not aim to perform a systematic optimization of the neural-network architecture. Therefore, the selected architecture should be regarded as a practical configuration for the present MLRS closure model rather than a globally optimized network structure. The final DNN model is summarized in Table 5.

The Rectified Linear Unit (ReLU) activation function was applied to all hidden layers to introduce nonlinearity, while the Mean Squared Error (MSE), defined in Eq. (12), was used as the loss function. Additionally, the performance of the machine learning was assessed by Mean Absolute Error (MAE), Symmetric Mean Absolute Percentage Error (SMAPE) and Coefficient of Determination (R^2), that were defined in Eq. (13)–(15). The model was trained using the Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 0.001, which was adaptively reduced by a factor of 0.8 when the validation loss did not decrease anymore. The network was trained for up to 5000 epochs with a batch size of 1000. To prevent overfitting, early stopping was triggered when the validation loss failed to improve for 50 consecutive epochs.

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - y_i)^2 \tag{12}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - y_i| \tag{13}$$

$$\mathrm{SMAPE} = \frac{100}{N}\sum_{i=1}^{N}\frac{2|y_i - y_i|}{y_i + |y_i| + 1\times 10^{-8}} \tag{14}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - y_i)^2}{\sum_{i=1}^{N}(y_i - \overline{y_i})^2} \tag{15}$$

where N is the number of data points, y_i, y_i, and $\overline{y_i}$ are actual value, predicted value, and averaged true value, separately.

Compared with tensor-basis neural network approaches (Cai et al., 2024) and invariant Reynolds-stress modeling approaches (Fu et al., 2024), the present plain DNN formulation is simpler to implement and easier to couple with the current CG-CFD framework based on OpenFOAM. It also allows inclusion of local features, such as density, velocity gradients, and wall-distance information, which are relevant to multicomponent buoyancy-driven hydrogen dispersion. However, the price of this simplicity is that important symmetry properties are not enforced by construction, and the physical consistency of the learned mapping relies more strongly on the training data and post-processing corrections. By contrast, invariant/tensor-basis approaches can embed symmetry constraints directly into the model structure and may therefore provide better physical consistency and improved robustness for extrapolation, although at the cost of a more constrained formulation and higher implementation complexity. In addition, uncertainty-aware ML-RANS methods, such as Bayesian or probabilistic formulations (Agrawal and Koutsourelakis, 2024; Man et al., 2025), can provide predictive uncertainty information and reliability indicators, which are valuable for practical deployment, although such methods were not explored in the present study.

Table 4
Mapping time for 4 time-steps.

Parameter	Mapping implementation	Time
Case 3	kNNImputer + serial	3722 s
	KD-Tree + parallel	30 s
Case 7	kNNImputer + serial	> 200 h
	KD-Tree + parallel	194 s

Table 5
Deep neural network parameters.

Parameter	Value
Hidden layer number	3
Neurons number	512
Batch size	1000
Epochs number	5000
Optimizer	Adam
Initial learning rate	10^{-3}
Activation function	ReLU
Loss function	MSE

The FG-CFD simulation database obtained from Cases 1–12, as listed in Table 2, was mapped to the CG mesh with a mapping ratio of 2. For cases with a side length of 5 m, data were stored every 100 s over a total simulation time of 3000 s, while for cases with a side length of 10 m, data were stored every 1000 s over 20,000 s of simulation time. After mapping, this resulted in a total of 9,596,640 data points, which were subsequently divided into 80% for training and 20% for testing. The temporal correlation was not explicitly considered in the training / test splitting strategy, while the transient simulation results will be validated later. The Case 13, with boundary conditions different from those of the training cases but within the same *Gr* number range, was reserved for independent validation. Before training, all inputs and outputs were normalized by Eq. (16), to ensure consistent scaling and enhance convergence:

$$z = \frac{x - \mu}{\sigma} \tag{16}$$

where μ is the mean value of the un-scaled data x, and σ is the standard deviation. It yields scaled data z, which is distributed around a mean value of 0 with a standard deviation of 1.

To preserve the realizability of the predicted Reynolds stress tensor, an eigenvalue-based correction was applied after the MLRS model predictions. Any negative eigenvalues were replaced by zero, and the tensor was reconstructed from the corrected eigensystem. In this way, the Reynolds stress tensor used in the momentum equations was ensured to be positive semi-definite.

Fig. 6 compares the MLRS model predictions with CFD computed values of the Reynolds stress component *Ryy* for both the training and testing datasets. The black line denotes the ideal prediction, while the red lines indicate the ±20% error bounds. In both sets, the predicted values show good agreement with CFD results, with the most data points falling within the ±20% error bounds. Fig. 7 shows the performance of the MLRS model on the validation set of case 13, where the boundary conditions are different from those used in the training cases but *Gr* number remains within the training range. Although differences in boundary conditions compared with the training cases, most predictions remain within the ±20% error bounds, highlighting the model's robustness and ability to maintain accuracy across independent conditions.

Table 6 quantitatively summarizes the statistical performance of the

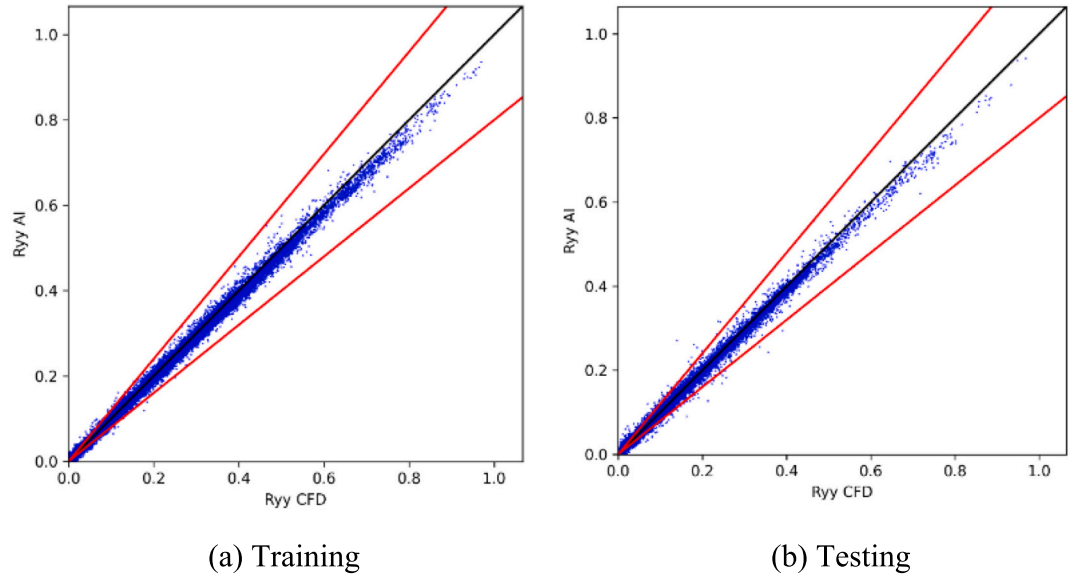


Fig. 6. Machine learning training of MLRS model.

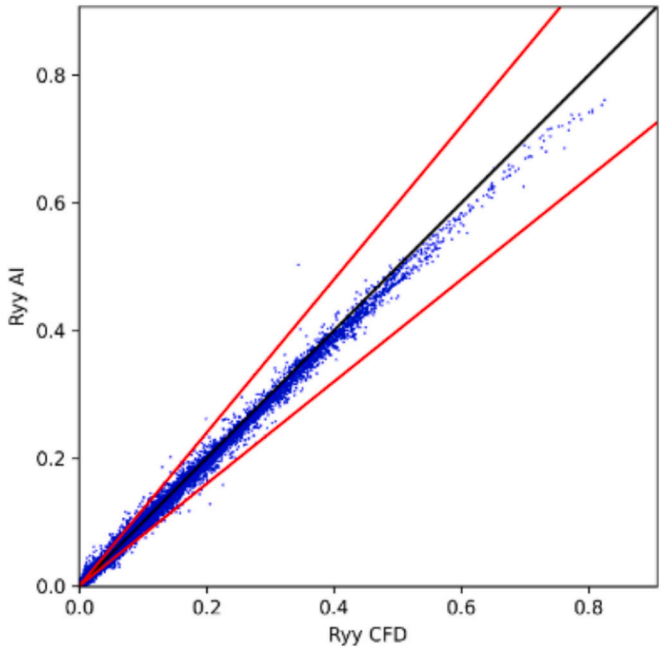


Fig. 7. Validation of MLRS model on Case 13.

Table 6
Statistical parameters.

Parameter	Training sets	Testing sets	Validation sets
MAE	4.31×10^{-4}	4.49×10^{-4}	4.11×10^{-4}
MSE	1.03×10^{-6}	1.42×10^{-6}	1.11×10^{-6}
SMAPE	39.25%	39.38%	42.63%
R^2	0.9964	0.9951	0.9947

MLRS model. The MAE and MSE values are on the order of 10^{-4} and 10^{-6}, respectively, across all the training, testing, and validation sets. Besides, the coefficient of determination R^2 exceeded 0.99 for all sets. Although the SMAPE values are relatively higher, that are between 39% and 43%, this can be attributed to the presence of quite small actual values (close to 0 as shown in Fig. 6 and Fig. 7) that enlarge percentage errors. These results indicate that the trained MLRS model successfully captures the nonlinear relationship between flow features and the Reynolds stress tensor, achieving a good accuracy and stability. The close agreement of the results between the validation sets and the training sets demonstrates the strong generalization capability of the MLRS model when applied to previously unseen data.

The validated MLRS model was then integrated into OpenFOAM, enabling its application in CG-CFD simulations as described in Section 5.

5. Evaluation of CG-CFD simulations

5.1. Implementation of the MLRS model in OpenFOAM

To enable CG-CFD simulations, the MLRS model was integrated into the OpenFOAM platform using the TensorFlow C API. The trained model was exported in the protobuffer format to allow deployment in a C++ environment. During the simulation, it will be loaded through the TensorFlow C API, which manages the computational graph and performs inference within the OpenFOAM framework. Finally, a data-driven turbulence model employed by the MLRS model was compiled and integrated into the MulticomponentFluid solver in OpenFOAM for subsequent CG-CFD simulations.

5.2. Results and discussion

As shown in Fig. 8, the hydrogen mass fraction at three vertical positions along the mid-plane (z/Z = 0.2, 0.5, 0.8) were extracted and compared between the CG-CFD simulation results and the reference FG-CFD simulation results. To comprehensively evaluate the performance of the CG-CFD simulations, two representative cases were selected: Case 3, which falls within the training dataset of the MLRS model, and Case 13, which is the independent case excluded from the training to examine the CG-CFD's generalization capability. In both Case 3 and Case 13, the CG-CFD approach successfully captures both the evolution and magnitude of hydrogen concentration over time. Although minor discrepancies are observed during the intermediate stages, particularly at lower heights where CG-CFD slightly underpredicts the concentration in Case 3 and overpredicts it in Case 13, these deviations remain within acceptable limits and have negligible influence on the long-term trends. The results confirm that the CG-CFD approach, coupled with the data-driven MLRS model, can effectively reproduce the key features of hydrogen transport dynamics in multicomponent mixing flow over a

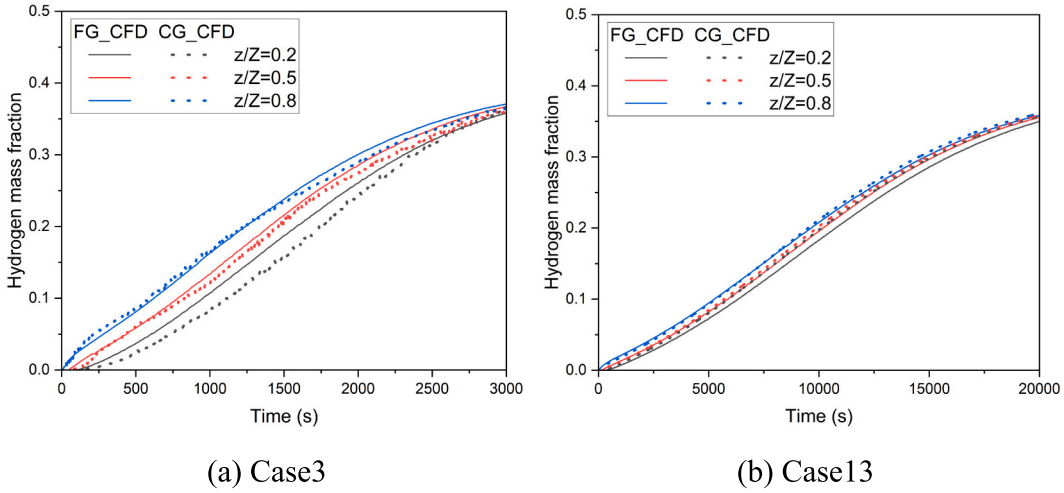


Fig. 8. Hydrogen concentration calculated by CG-CFD.

long transient process. It should be noted that the present work did not investigate transient cases with explicitly time-varying inlet boundary conditions within a single simulation. Therefore, although the model performed well in long transient simulations and in an independent case with different boundary conditions, further validation under real transient simulation where boundary conditions would vary is still needed.

Fig. 9 presents the predicted velocity field along the vertical direction in Case 13, with comparisons made between the CG-CFD simulation results, the FG-CFD simulation results, and the mapping data. The coordinates and gravity direction are shown in Fig. 3. It can be seen that CG-CFD simulations successfully capture the key flow features, matching both the trends and magnitudes of the FG-CFD data. Deviations are primarily observed in the near-wall regions, where sharp velocity gradients exist and the solution may be more sensitive to mesh resolution. The velocity peaks predicted by CG-CFD appear slightly farther away from the wall compared with FG-CFD. These errors may be attributed to the accumulated error in Reynolds stress prediction that influences momentum solution and flow structure. Nevertheless, the current CG-CFD framework coupled with the MLRS model demonstrates reliable performance and strong potential for predicting velocity fields on coarse grids.

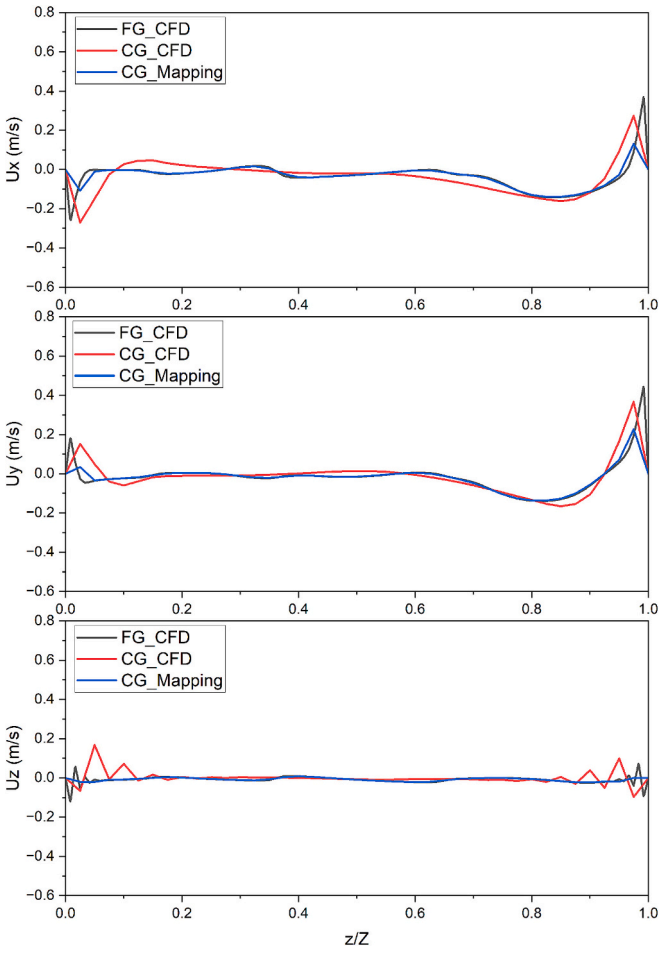


Fig. 9. Velocity distribution at 10000 s in Case 13.

Fig. 10 shows the predicted Reynolds stress components at 10000 s in Case 13, with comparisons between the CG-CFD results, the reference FG-CFD results, and the mapping data. It can be seen that the CG-CFD approach successfully captures the distributions of the Reynolds stresses across the domain. The integration of the MLRS model enables the coarse-grid solver to preserve key features of turbulence anisotropy. However, the CG-CFD predictions exhibit a tendency to overestimate the Reynolds stresses in the near-wall regions, where sharp velocity gradients occur, as also illustrated in Fig. 9. In particular, for the normal Reynolds-stress components, the maximum absolute errors are approximately 0.090, 0.092, and 0.096 m^2/s^2 for Rxx, Ryy, and Rzz, occurring at $z/Z = 0.925$, 0.931, and 0.958, respectively.

Several factors may contribute to these deviations. For example, the errors introduced by mapping process may contribute to the difference, especially considering that the coarse-grid is uniform while the fine-grid is nonuniform. Besides, the numerical difference between training data generation and the coarse-grid CFD can affect the performance of data-driven model, as discussed by Liu et al. (2022b). Although the mapping procedure described in Section 4.2 helps reduce this inconsistency, it was not investigated systematically and should be regarded as a limitation of the present work. Additionally, the limited generalization ability of the current MLRS model may introduce inaccuracies when applied to flow conditions that differ from the training dataset. Overall, these deviations are most evident in near-wall regions with strong gradients, but they do not change the main conclusion that the proposed CG-CFD coupled with MLRS model can reproduce the flow features within the investigated validation range.

Table 7 summarizes the total simulation time for both FG-CFD and CG-CFD under Case 3 and Case 13. All simulations were performed on the same computing station equipped with 4 × AMD Opteron 6274 CPUs (64 threads total, base frequency 2.2 GHz). As shown in Table 7, the CG-CFD approach achieves a significant reduction in computational cost

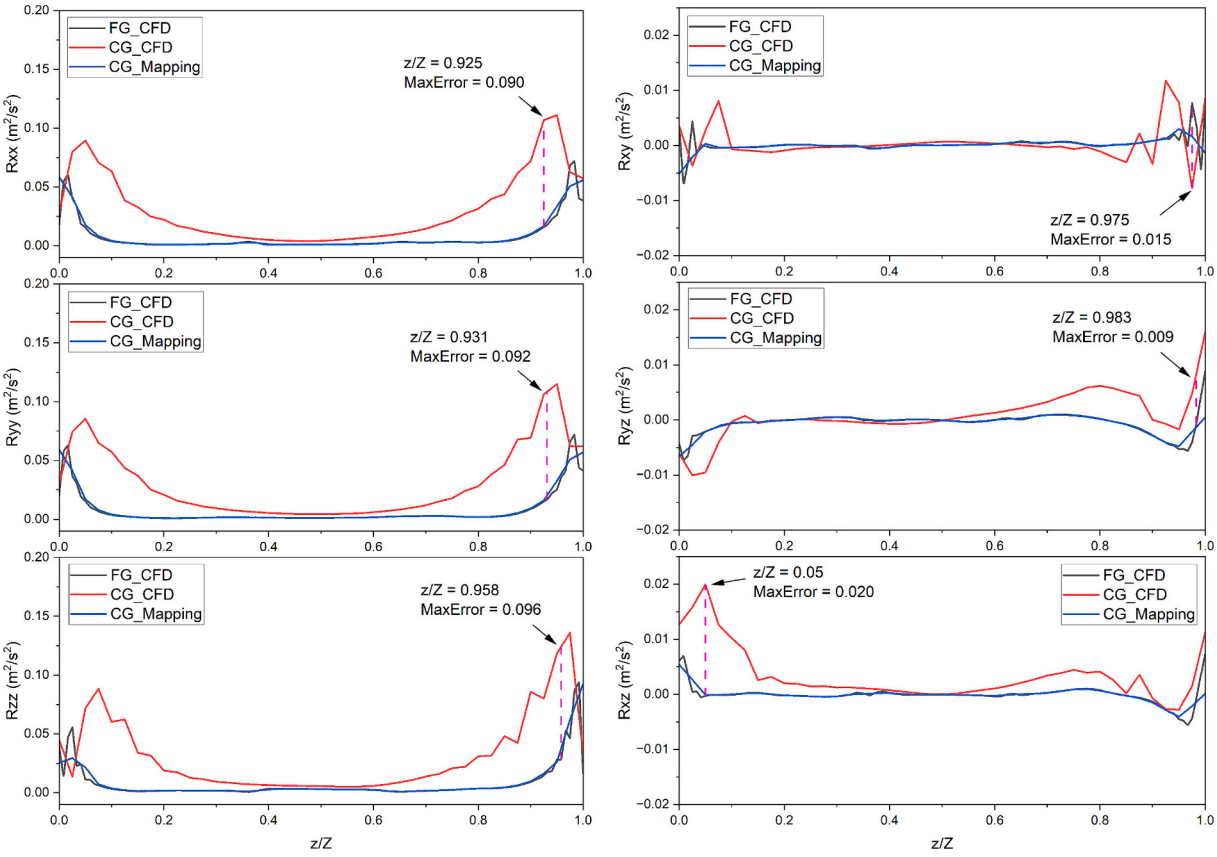


Fig. 10. Reynolds stress distribution at 10000 s in Case 13.

Table 7
Comparison of simulation time.

Case	Method	Number of cells	Turbulence model	Simulation time (s)	Wall-clock time (h)
Case 3	FG-CFD	110,592	Lien Cubic k-ε	3000	~ 12
	CG-CFD	10,648	Data-driven model	3000	~ 6
Case 13	FG-CFD	681,472	Lien Cubic k-ε	20,000	~ 472
	CG-CFD	64,000	Data-driven model	20,000	~ 40

compared to the FG-CFD simulations. For Case 3, the CG-CFD simulation is completed in approximately 6 h, representing a 50% reduction relative to the 12-h FG-CFD simulation. For the larger space in Case 13, the total runtime is reduced from approximately 472 h to 40 h, providing a 91.5% reduction in wall-clock time. These results demonstrate the efficiency advantage of the CG-CFD framework combined with the data-driven turbulence model. The different speedup factors mainly arise from the different computational scales of the two cases. As given in Table 2, Case 3 is a smaller and shorter transient problem, for which the baseline FG-CFD cost is relatively limited. Therefore, although the CG-CFD model reduces the computational cost, the wall-clock reduction remains moderate. In contrast, Case 13 involves a larger space and a much longer transient simulation, so the baseline FG-CFD cost is substantially higher. Under such conditions, the reduced mesh resolution and simplified turbulence closure in CG-CFD lead to a much more pronounced reduction in total runtime. In other words, the efficiency advantage of the proposed framework becomes more significant as the original FG-CFD problem becomes more computationally expensive.

It should be noted that the runtimes reported in Table 7 represent the computational cost of the CG-CFD simulations only. The cost of FG-CFD database generation and model training is not included in this comparison. Therefore, the proposed framework is most suitable for repeated applications within the trained *Gr* range, where the cost becomes more efficiency when the trained model is repeatedly applied to many new cases.

Although the proposed method shows good potential for predicting hydrogen dispersion, the uncertainty quantifications of the MLRS model have not been performed in this work. Specifically, the uncertainties of the present MLRS model arise from several aspects. First, the MLRS model is trained on FG-CFD data generated using the Lien Cubic k-ε model rather than on DNS or experimental data. Therefore, the MLRS model may have the modeling uncertainty from the reference turbulence closure, although it was validated against the HOMER experiment. Second, the mapping procedure from the nonuniform FG mesh to the uniform CG mesh introduces an additional source of uncertainty. While a limited sensitivity study of the mapping ratio was presented, the influence of mapping parameters on the final MLRS and CG-CFD results has not yet been quantified. Third, there is uncertainty associated with the neural network closure itself. Overall, these sources of uncertainty require further investigation in future work.

6. Conclusion

In this study, a data-driven turbulence model used in CG-CFD simulations was developed to significantly accelerate CFD simulations of hydrogen dispersion in large enclosures. A comprehensive FG-CFD database capturing buoyancy-driven, anisotropic hydrogen mixing flows was constructed within the range of *Gr* numbers from 3.02×10^{10} to 1.90×10^{15}. After data preprocessing, a DNN method was trained on the high-fidelity dataset to predict the six Reynolds stresses from coarse-grid flow features, forming a MLRS model. By integrating the MLRS model into the OpenFOAM, a data-driven turbulence model was

developed. Finally, the CG-CFD simulations employing the data-driven turbulence model was evaluated against database. The main conclusions are as follows:

(1) The proposed mapping guideline can provide robust mapped fields with weak sensitivity to the mapping ratio.
(2) The MLRS model trained with 17 input features can accurately predict the Reynolds stress tensor for both the training and unseen datasets.
(3) The developed data-driven turbulence model, when integrated into the CG-CFD simulations, achieved substantial reductions in computational cost while maintaining high accuracy in hydrogen dispersion simulations.
(4) The model presents sustaining acceptable accuracy under previously unseen cases within the range of *Gr* numbers covered by the training set.

This work primarily serves as a methodological validation of a data-driven framework for accelerating the simulation of transient hydrogen dispersion in large spaces. A limitation of the present MLRS model is that Galilean invariance is not guaranteed by construction and positive semi-definiteness is enforced after training. Therefore, the model is validated within the investigated range of present study, rather than being presented as a universal Reynolds stress closure. Furthermore, to enhance the robustness and applicability of this approach, future work should be complemented by comprehensive sensitivity analyses, uncertainty quantification, and the incorporation of stronger physical realizability constraints.

CRediT authorship contribution statement

Xiang Zhang: Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Data curation. **Fabian Wiltschko:** Software, Methodology, Formal analysis. **Aurelian Florin Badea:** Writing – review & editing, Supervision. **Xu Cheng:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors gratefully acknowledge the financial support from KSB Stiftung (Nr.1.1386.2023.1). They also thank Dr. Ronald Lehnigk from Helmholtz-Zentrum Dresden-Rossendorf for technical support with OpenFOAM.

Data availability

Data will be made available on request.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., 2016. Tensorflow: large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv, 1603.04467.

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

Ajuria-Illaramendi, E., Alguacil, A., Bauerheim, M., Misdariis, A., Cuenot, B., Benazera, E., 2020. Towards a Hybrid Computational Strategy Based on Deep Learning for Incompressible Flows. AIAA Aviation Forum, Virtual Event, United States, pp. 1–17.

Berrone, S., Oberto, D., 2022. An invariances-preserving vector basis neural network for the closure of Reynolds-averaged Navier–Stokes equations by the divergence of the Reynolds stress tensor. Phys. Fluids 34, 095136.

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

Chen, Z., Deng, J., 2024. Data-driven RANS closures for improving mean field calculation of separated flows, p. 12.

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

Fu, X., Fu, S., Liu, C., Zhang, M., Hu, Q., 2024. Data-driven approach for modeling Reynolds stress tensor with invariance preservation. Comput. Fluids 274, 106215.

Fukami, K., Fukagata, K., Taira, K., 2019. Super-resolution reconstruction of turbulent flows with machine learning. J. Fluid Mech. 870, 106–120.

Gharari, R., Kazeminejad, H., Mataji Kojouri, N., Hedayat, A., 2018. A review on hydrogen generation, explosion, and mitigation during severe accidents in light water nuclear reactors. Int. J. Hydrogen Energy 43, 1939–1965.

Hajji, Y., Jouini, B., Bouteraa, M., Elcafsi, A., Belghith, A., Bournot, P., 2015. Numerical study of hydrogen release accidents in a residential garage. Int. J. Hydrogen Energy 40, 9747–9759.

Halder, R., Ataei, M., Salehipour, H., Fidkowski, K., Maki, K., 2024. Reduced-order modeling of unsteady fluid flow using neural network ensembles. Phys. Fluids 36, 077140.

Hou, X., Lan, H., Zhao, Z., Li, J., Hu, C., Li, Y., 2023. Effect of obstacle location on hydrogen dispersion in a hydrogen fuel cell bus with natural and mechanical ventilation. Process Saf. Environ. Prot. 171, 995–1008.

Kampili, M., Vijaya Kumar, G., Kelm, S., Arul Prakash, K., Allelein, H.-J., 2021. CFD simulations of stratified layer erosion in MiniPanda facility using the tailored CFD solver containmentFOAM. Int. J. Heat Mass Transf. 178.

Kingma, D.P., Ba, J., 2014. Adam: a method for stochastic optimization. arXiv preprint arXiv, 1412.6980.

Krohn, B., M, S., Niceno, B., M, P., H, B., Shams, A., Roelofs, F., 2015. Turbulent Gas Mixing in Strong Density Stratified Shear and Non- Shear Flows.

Launder, B.E., Sharma, B.I., 1974. Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc. Letters Heat Mass Transfer 1, 131–137.

Lien, F.S., Chen, W.L., Leschziner, M.A., 1996. Low-Reynolds-Number Eddy-Viscosity Modelling Based on Non-Linear Stress-Strain/Vorticity Relations.

Ling, J., Kurzawski, A., Templeton, J., 2016. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. J. Fluid Mech. 807, 155–166.

Liu, W., Fang, J., Rolfo, S., Moulinec, C., Emerson, D.R., 2021. An iterative machine-learning framework for RANS turbulence modeling. Int. J. Heat Fluid Flow 90, 108822.

Liu, Y., Hu, R., Kraus, A., Balaprakash, P., Obabko, A., 2022a. Data-driven modeling of coarse mesh turbulence for reactor transient analysis using convolutional recurrent neural networks. Nucl. Eng. Des. 390, 111716.

Liu, Y., Hu, R., Zou, L., Nunez, D., 2022b. SAM-ML: integrating data-driven closure with nuclear system code SAM for improved modeling capability. Nuclear Eng. Design 400, 112059.

Liu, Z., Zhao, P., Florin, B.A., Cheng, X., 2025. Thermal stratification prediction in reactor system based on CFD simulations accelerated by a data-driven coarse-grid turbulence model. Nucl. Eng. Technol. 57, 103288.

Ma, Q., Guo, Y., Zhong, M., Ya, H., You, J., Chen, J., Zhang, Z., 2024. Numerical simulation of hydrogen explosion characteristics and disaster effects of hydrogen fueling station. Int. J. Hydrogen Energy 51, 861–879.

Malakhov, A.A., Avdeenkov, A.V., du Toit, M.H., Bessarabov, D.G., 2020. CFD simulation and experimental study of a hydrogen leak in a semi-closed space with the purpose of risk mitigation. Int. J. Hydrogen Energy 45, 9231–9240.

Man, A., Jadidi, M., Keshmiri, A., Yin, H., Mahmoudi, Y., 2025. Uncertainty and error quantification for data-driven Reynolds-averaged turbulence modelling with mean-variance estimation networks. J. Comput. Phys. 543, 114388.

Maulik, R., Sharma, H., Patel, S., Lusch, B., Jennings, E., 2021. A turbulent eddy-viscosity surrogate modeling framework for Reynolds-averaged Navier-stokes simulations. Computers & Fluids 227, 104777.

McConkey, R., Yee, E., Lien, F.S., 2022. Deep structured neural networks for turbulence closure modeling. Phys. Fluids 34, 035110.

Tracey, B.D., Duraisamy, K., Alonso, J.J., 2015. A machine learning strategy to assist turbulence model development, 53rd AIAA aerospace sciences meeting. American Institute of Aeronautics and Astronautics. AIAA Paper 2015-1287, pp. 1–22.