Inserting machine-learned virtual wall velocity for large-eddy simulation of turbulent channel flows

Naoki Moriya · Kai Fukami · Yusuke Nabae
Masaki Morimoto · Taichi Nakamura · Koji Fukagata

June 18, 2021

Abstract We propose a supervised-machine-learning-based wall model for coarse-grid wall-resolved large-eddy simulation (LES). Our consideration is made on LES of turbulent channel flows with a first grid point set relatively far from the wall (∼ 10 wall units), while still resolving the near-wall region, to present a new path to save the computational cost. Convolutional neural network (CNN) is utilized to estimate a virtual wall-surface velocity from \(x-z\) sectional fields near the wall, whose training data are generated by a direct numerical simulation (DNS) at \(Re_\tau = 180\). The virtual wall-surface velocity is prepared with the extrapolation of the DNS data near the wall. This idea enables us to give a proper wall condition to correct a velocity gradient near the wall. The estimation ability of the model from near wall information is first investigated as a priori test. The estimated velocity fields by the present CNN model are in statistical agreement with the reference DNS data. The model trained in a priori test is then combined with the LES as a posteriori test. We find that the LES can successfully be augmented using the present model at both the friction Reynolds number \(Re_\tau = 180\) used for training and the unseen Reynolds number \(Re_\tau = 360\) even when the first grid point is located at 5 wall units off the wall. We also investigate the robustness of the present model for the choice of sub-grid scale model and the possibility of transfer learning in a local domain. The observations through the paper suggest that the present model is a promising tool for recovering the accuracy of LES with a coarse grid near the wall.

Keywords Machine learning, computational methods, turbulence simulation

1 Introduction

Large-eddy simulation (LES) has played a crucial role for mechanical and aerospace engineering applications in a practical manner. Capturing momentum transfer and near wall behaviors aided by LES in a reasonable accuracy enable us to analyze complex turbulence phenomena and also understand flow physics. However, it requires the massive computational power with the gigantic number of discretized grid points to handle these simulations since turbulence includes a wide range of scales inside them.

To avoid enormous computational cost to resolve near-wall structures, LES at practically high Reynolds numbers is often performed by modeling the flow in the near-wall region and imposing the boundary condition off the wall (i.e., wall-modeled LES). The well-used strategy for high Reynolds number LES is to resolve turbulent structures in the outer layer region corresponding to approximately 90% of the boundary layer directly, while modeling the rest of 10% in the inner layer region [1]. The fact that the computational cost...
increases by more than the square of the Reynolds number also supports the use of aforementioned strategy \[1\]. \[2\]. Hence, the utilization of a proper wall model is unavoidable to date for capturing near wall behaviors with the reasonable number of computational grid points. Such efforts on the wall model can mainly be divided into two types: (i) hybrid model with RANS (e.g., detached eddy simulation, DES) \[3,4\] and (ii) methods to approximate/augment the wall-shear stress \[5\]. The variety of open source codes enables us to access some implementations of the former model easily in recent years, but the discontinuity between LES and RANS is known as one of the problems, which requires an artificial manipulation to avoid the inconsistency with regard to velocity there \[6,7\]. The latter has been carried out for a long time with the use of an artificial boundary condition. The sophisticated concept has recently been developed, in which viscous-scale grids are embedded in the inner layer so as to solve the boundary layer equation, and its effectiveness against the above velocity mismatch has been reported in Kawai and Larsson \[8\]. However, it is still difficult to accurately predict the turbulent boundary layer at high Reynolds numbers without tuning of empirical parameters and the use of complex control theories.

An open issue is present also for LES with no-slip boundary condition (i.e., wall-resolved LES) if one wants to locate the first grid point outside the viscous sublayer to save the computational cost while resolving the near-wall structure. Consider, for instance, the boundary condition is discretized using a second-order central difference on a staggered grid system, the no-slip boundary condition for the streamwise velocity, say \(u_w = 0\), on the wall (\(y = 0\)) can be discretized to satisfy \((u_0 + u_1)/2 = 0\), i.e., \(u_0 = -u_1\), where \(u_0\) and \(u_1\) denote the streamwise velocity \(u\) at \(y_0 = -\Delta y/2\) (i.e., the first grid point outside the boundary) and \(y_0 = \Delta y/2\) (i.e., the first grid point off the wall), respectively, with \(\Delta y\) being the size of first wall-normal grid, while the velocity gradient on the wall is discretized as \((\partial u/\partial y)_w = (u_1 - u_0)/\Delta y\). Namely, \(u_1\) is always computed as \(u_1 = u_w + (\partial u/\partial y)_w \Delta y/2\). This is reasonable as far as the first grid point is located in the viscous sublayer where the linear law \(\tau^+ = y^+\) (where \(\tau\) is the mean streamwise velocity, and the superscript “+“ denotes the wall units) holds. However, this treatment becomes inappropriate when the first grid point is farther off the wall. Since the stress balance near the wall \((y^+ < Re_{\tau}, \text{where } Re_{\tau} \text{denotes the friction Reynolds number})\) can be expressed in wall units as

\[
\frac{\partial \tau^+}{\partial y^+} - \frac{u^+ v^+}{\tau^+} = 1, \tag{1}
\]

imposition of no-slip boundary condition in the original form overestimates the velocity gradient on the wall by the amount of Reynolds shear stress \(-\bar{u}^+ \bar{v}^+ \) when the first grid point is located outside the viscous sublayer — and this why it is usually advised to place several grid points in the viscous sublayer for a wall-resolved LES. Thus, a proper correction is required for the imposition of discretized no-slip boundary condition even if the Reynolds shear stress is perfectly amended by the SGS model. This argument is similar to the discussion by Kawai and Larsson \[8\] on the log-layer mismatch; however, for wall-resolved LES, similar corrections should be required not only for the mean streamwise velocity but also for all the fluctuating velocity components to capture the near-wall coherent structures.

To address the aforementioned issues, machine learning, which has been known as a good candidate to handle complex fluid flow problems \[9,10\], can be an attractive tool. As for the application to closure modeling, the machine learning has already had a citizenship there \[11,12\]. One of the seminal works is tensor-basis neural network (TBNN) with Galilean invariance embedded by Ling et al. \[13\]. Their model was tested for duct and wavy-wall flows. We have recently been able to see the extension of TBNN to various flow configurations and problem settings, e.g., channel flow at various Reynolds numbers \[14\], a cylindrical and inclined jet in crossflow \[15\], and the pressure-Hessian based closure \[16\]. For the application to LES, the idea to estimate finer (unresolved) scales from solved large-scale information has widely been accepted with the supervised machine learning, whose training data is prepared by direct numerical simulation (DNS) \[17\]. \[18\]. \[19\]. \[20\]. \[21\]. \[22\]. \[23\]. \[24\]. \[25\].

In the present study, we focus on the capability of machine learning for data estimation and reconstruction towards the augmentation of LES, rather than the closure modeling efforts. Because the machine learning is good at extracting hidden features of data, it has also been widely utilized for state estimation tasks \[26\]. \[27\]. Guastoni et al. \[28\] reported that a convolutional neural network (CNN) is able to estimate the state of turbulent channel flow from only wall-sensor measurements by combining to proper orthogonal decomposition. Toward the combination with the opposition control, there are several studies that aim at estimating the
Inserting machine-learned virtual wall velocity for large-eddy simulation of turbulent channel flows

Fig. 1 The present machine-learning-based wall modeling in LES. (a) Concept of a priori test and a posteriori test. (b) Preparation for a virtual wall surface velocity \( u_{yw} \). The vertical blue lines at \( y^+ = 0, 20 \) and 40 represent the locations of cell faces in LES on a staggered grid.

velocity field at \( y^+ \approx 15 \) in a turbulent channel flow from the wall measurements \([29,30]\). More recently, Nakamura et al. \([31]\) compared the capability of linear methods and neural networks in state estimation of minimal turbulent channel flow. These results suggest the hidden relation between the state of turbulent flows and wall quantities. In turn, this also motivates us to expect the possibility to estimate the wall information from the state above the wall. Obtaining the clues from these pieces above, we propose a machine-learning-based wall model for wall-resolved LES. Especially, we focus on giving a proper wall condition which corrects a velocity gradient near the wall for the case with a very coarse staggered grid, with the first point from the wall being located at \( y^+ \sim 10 \). Our model aims to insert the machine-learning-based artificial slip velocity for corrections of not only the mean velocity profile but also all the fluctuating velocity components.

The present paper is organized as follows: we introduce the overview with the covered regression methods in Section 2. The construction of the present wall model (a priori test) and its application to the LES (a posteriori test) are expressed in Section 3. Concluding remarks are provided in Section 4.

2 Methods

2.1 Overview of the present wall modeling for large-eddy simulation

The concept of this study is mainly composed of two parts — a priori test and a posteriori test — as illustrated in Fig. 1. As described in introduction part, we aim at reducing the number of grid points in the near-wall region in an LES using a staggered grid system as much as possible, while still resolving the near-wall flow structure. However, when we apply a no-slip wall boundary condition in the wall-normal direction, a velocity at the first point from the wall may be overestimated as mentioned in the introduction, which causes non-negligible error in the mean velocity distribution. Similarly, although not illustrated in Fig. 1, all the fluctuating velocity components are also subjected to the similar discretization error. Therefore, we here consider the artificial wall slip velocity \( u_{yw} \) to fix all the velocity components at the first point from the wall \( u_{y1} \).

In a priori test, a machine-learning model \( \mathcal{F} \) is constructed to estimate the artificial slip velocity \( u_{yw} \), from the velocity on two \( x-z \) cross sections near wall region \( \{ u_{y1}, u_{y2} \} \) such that \( u_{yw} \approx \mathcal{F}(u_{y1}, u_{y2}) \), where \( y_1 \) and \( y_2 \) denote the locations of the first and second grid points used in LES. Hereafter, this artificial wall velocity is referred to as virtual wall surface velocity. For the training of machine-learning model, we use subsampled \( x-z \) cross-sectional velocity data \( \tilde{u} \) obtained by a direct numerical simulation (DNS) at the same friction Reynolds number \( Re_\tau = 180 \) as that used in the target LES. The subsampling operation for the DNS data enables us to match the streamwise and spanwise grid resolutions to those in the target LES. As illustrated in Fig. 1(b), the virtual wall surface velocity \( u_{yw} \) used for a training process is prepared using
the linear extrapolation from the DNS data at the first point from wall in the target LES, $y_1^+$ (where “+” denotes the wall units), and the next point in DNS, $y_1^++\delta y^+$, as

$$\tilde{u}_{DNS\ w}^{(y_1^+,y_1^++\delta y^+)} = \tilde{u}_{DNS\ w}^{(y_1^+,y_1^+)} - \tilde{u}_{DNS\ w}^{(y_1^+,y_1^+)} + \delta y^+,$$

(2)

Here, the subsampling operation is denoted as $\tilde{()}$. In a priori test, a machine-learning model $F$ estimates the virtual wall surface velocity $\tilde{u}_{ML\ w}$ from the two cross-sectional DNS data at the first and the second points from the wall in a target LES, $\tilde{u}_{DNS\ w}^{(y_1^+,y_1^+)}$ and $\tilde{u}_{DNS\ w}^{(y_1^+,y_1^+)}$, as

$$\tilde{u}_{ML\ w} = F(\tilde{u}_{DNS\ w}^{(y_1^+,y_1^+)}, \tilde{u}_{DNS\ w}^{(y_1^+,y_1^+)}),$$

(3)

and it is compared with the reference, $\tilde{u}_{DNS\ w}^{(y_1^+,y_1^+)}$. The constructed model $F$ is then applied to an LES in a posteriori test by using it as the boundary condition, i.e.,

$$u_{w}^{LES} = \tilde{u}_{ML\ w} = F(u_{w}^{LES,1}, u_{w}^{LES,2}).$$

(4)

2.2 Convolutional neural network

As a machine-learning model, we capitalize on convolutional neural network (CNN) originally developed in image recognition. The filters, trainable parameters inside the CNN, are able to handle high-dimensional data efficiently and extract key features. Thanks to its unique capability in handling high-dimensional data, the use of CNN has also been spread in the fluid dynamics field in recent years.

As shown in Fig. 2(a), the basic operation of CNN is to take a summation of a Hadamard product between a designated region of input data and a trainable filter $h$. Usually, a CNN consists of several convolutional layers to build a certain relationship between inputs and outputs. In this study, velocity fields of $x-z$ cross-sections at two designated $y^+$ are fed into the first convolutional layer and then $q^{(l)}$ will be obtained as an output of the first layer. The procedure in obtaining $q^{(l)}$ from $q^{(l-1)}$ is repeated until $l < l_{max}$, where the final output $q^{(l_{max})}$ corresponds to a virtual wall surface velocity in our case. The operation inside the convolutional layer can be expressed as,

$$q_{ijk}^{(l)} = \phi \left( \sum_{m=1}^{M} \sum_{p=0}^{H_w-1} \sum_{q=0}^{H_h-1} h_{pmq}^{(l)} q_{i+p+C,j+q-C,m}^{(l-1)} + b_k^{(l)} \right),$$

(5)

where $C = \text{floor}(H/2)$, $b_k^{(l)}$ is a bias, $\phi$ is an activation function, $M$ is the number of input data channels and $k$ is the number of filters (equals to number of output data channels), respectively.
Table 1 The structure of the machine learning model. The convolution layer is denoted as Conv2D.

| Layer (filter size, # of filters) | Data size | Activation |
|----------------------------------|-----------|------------|
| Input                            | (64,64,2) |            |
| 1st Conv2D (5,32)                | (64,64,32)| ReLU       |
| 2nd Conv2D (5,32)                | (64,64,32)| ReLU       |
| 3rd Conv2D (5,32)                | (64,64,32)| ReLU       |
| 4th Conv2D (5,32)                | (64,64,32)| ReLU       |
| 5th Conv2D (5,32)                | (64,64,32)| ReLU       |
| 6th Conv2D (5,1)                 | (64,64,1) | Linear     |

Table 2 The covered grid widths in the y direction.

| (y₁⁺, y₂⁺) | y₁⁺ + δyDNS⁺ | Δy⁺ | N⁺_y |
|------------|--------------|-----|------|
| (2.50, 7.50) | 3.75 | 5.00 | 72   |
| (5.00, 15.0)  | 6.25 | 10.0 | 36   |
| (10.0, 30.0)   | 11.3 | 20.0 | 18   |

The details of the proposed model are summarized in Table 1. The weights on the filters w are optimized through the back propagation [43] by minimizing a cost function computed from output q(lmax) and reference data qref, such that

\[
w = \text{argmin}_w |q(l_{\text{max}}) - q_{\text{ref}}|^2 = \text{argmin}_w |F(\{\tilde{u}^{\text{DNS}}_{y_1}, \tilde{u}^{\text{DNS}}_{y_2}\}; w) - \tilde{u}^{\text{DNS}}_w|^2.\]  

We use the L₂ error as the cost function.

2.3 Linear regression analysis

To clarify the advantage of nonlinear CNN, we also perform a linear regression analysis for virtual wall-surface velocity estimation [31,44]. The linear regression is able to express the output as a linear map of input,

\[Q = P\beta,\]  

where P, Q, β is an input matrix, an output matrix, and an weight matrix, respectively. Since we use two velocity sectional fields \{\tilde{u}^{\text{DNS}}_{y_1}, \tilde{u}^{\text{DNS}}_{y_2}\} as the input data, the virtual wall surface velocity estimated by the linear regression \tilde{u}^{LR}_w is expressed as the linear sum of input velocity and weight, such that

\[\tilde{u}^{LR}_w = \tilde{u}^{\text{DNS}}_{y_1}\beta_1 + \tilde{u}^{\text{DNS}}_{y_2}\beta_2,\]  

where a single snapshot of velocity data is reshaped into a one-dimensional vector, β₁ and β₂ are the weight matrices. The weight matrices β₁ and β₂ are optimized so that the L₂ norm between the left-hand side and the right-hand side of Eq. 8 can be minimized over the trained snapshots. It can mathematically be formed as

\[β₁, β₂ = \text{argmin}_{β₁, β₂} |\tilde{u}^{\text{DNS}}_w - (\tilde{u}^{\text{DNS}}_{y_1}\beta_1 + \tilde{u}^{\text{DNS}}_{y_2}\beta_2)|^2.\]  

We do not use the L₁ or L₂ penalization terms for the fair comparison to the CNN (i.e., Eq. 6) [31].
Fig. 3. Visualization of virtual wall surface velocities for each grid-width case in a priori test. The values underneath each contour are the $L_2$ error norm.
Fig. 4 Probability density function of virtual wall surface velocity obtained by the DNS and machine-learning model in a priori test.

3 Results

We apply the present technique to a turbulent channel flow for easiness of assessments. As explained in Section 2.1, we use the DNS data for the training of the machine-learning model in a priori test. The governing equations are the incompressible continuity equation and Navier–Stokes equation,

$$\nabla \cdot u = 0, \quad \frac{\partial u}{\partial t} = -\nabla \cdot (uu) - \nabla p + \frac{1}{Re_*} \nabla^2 u, \tag{10}$$

where $u = [u, v, w]^T$ represents the velocity vector in the streamwise ($x$), wall-normal ($y$) and spanwise ($z$) directions; $p$ is the pressure and $t$ is the time. All physical quantities are made dimensionless by using density $\rho^*$, friction velocity $u_*^*$, and channel half-width $\delta^*$, where $(\cdot)^*$ denotes the dimensional quantities. The DNS is performed under the constant pressure gradient condition at the friction Reynolds number $Re_* = (u_*^* \delta^*)/\nu^* = 180$, where $\nu^*$ denotes the kinematic viscosity. The size of computational domain and grid points here are $(L_x \times L_y \times L_z) = (4\pi \delta \times 2\delta \times 2\pi \delta)$ and $(N_x \times N_y \times N_z) = (256 \times 96 \times 256)$. The time step in the present DNS is $\Delta t_{DNS} = 6.30 \times 10^{-2}$. The present DNS code is the same as that used in the previous study [45]. The governing equations (Eq. 10) are spatially discretized with the energy-conserving fourth-order finite difference scheme on a staggered grid system [46]. The temporal integration is performed using the low-storage, third-order Runge-Kutta/Crank-Nicolson scheme [47] with the higher-order SMAC-like velocity-pressure coupling scheme [48]. The pressure Poisson equation is solved with the fast Fourier...
In this study, three cases in terms of grid width in the wall-normal direction are considered, as summarized in Table 2. For the training of machine-learning model, we use the $x-z$ cross-sectional velocity data subsampled to $(64 \times 64)$, which are obtained by a direct numerical simulation (DNS) at the same friction Reynolds number $Re_\tau = 180$ as that used in the baseline LES, as explained in Section 2.1. The time interval for the training data sampling is $\Delta t_{ML} = 1.26$, which corresponds to 20 time steps in the DNS. We use 3000 snapshots for training the baseline model, although we will discuss the dependence of the estimation ability on the amount of the training snapshots later. Among them, 70% is used for the training data, while 30% is used for the validation data.
3.1 A priori test: construction of machine-learning-based wall model

We construct a machine-learning model to estimate the virtual wall surface velocity, which will be applied for LES. Let us visualize the estimated virtual wall surface velocities for each case in Fig. 3. The values underneath each contour represent the $L_2$ error norm normalized by the velocity fluctuation $\epsilon = ||u_{DNS} - u_{ML}||_2/||u'_{DNS}||_2$. The linear regression cannot estimate the virtual wall surface velocity accurately in terms of both the contours and the $L_2$ error norm. In contrast, the flow fields estimated by the machine-learning model are in qualitative agreement with the reference DNS for all three cases.

We also investigate the ability of models using probability density function (PDF), as presented in Fig. 4(a). The velocity distributions of DNS and machine-learning-based estimation are generally consistent, but the low probability events do not show good agreement with each other. This is because the present model is trained to minimize the $L_2$ error as stated in Eq. 6 thereby leading to output the average value of fluctuation components.

We then evaluate the estimation performance of the machine-learning model on wave space using the energy spectrum, as presented in Fig. 5(b). For each grid width, the machine-learning model is able to estimate well especially the low wavenumber components. However, the overestimation can be found at the high wavenumber counterparts, which implies that machine-learning model preferentially estimates the low wave-number components. This observation is consistent with previous studies of turbulence analysis using supervised machine learning methods [27,49].

The dependence of the estimation ability on the amount of the training snapshots is also examined in Fig. 6. The estimation accuracy improves with increasing the number of snapshots used for training of machine-learning model in all cases. Notably, the decreasing rate of the error becomes smaller with the cases of more than 3000 snapshots. Therefore, we hereafter use machine-learning models trained with 3000 snapshots in a posteriori test.

3.2 A posteriori test: application of machine-learned model to LES

In a posteriori test, the machine-learning model trained in a priori test is applied to the LES. We use f2py [50] to combine a FORTRAN-based simulation codes with a python-based machine-learning module. In LES, the governing equations are the filtered and coarse-grained continuity equation and the Navier–Stokes equation,

\[
\nabla \cdot \bar{u} = 0, \quad \frac{\partial \bar{u}}{\partial t} = -\nabla \cdot (\bar{u} \bar{u}) - \nabla \bar{p} + \frac{1}{Re_b} \nabla^2 \bar{u} + \nabla \cdot \bar{\tau},
\]

where $(\tilde{\cdot})$ represents a filter operation, and $\bar{\tau}$ denotes the sub-grid scale (SGS) stress tensor. The size of the computational domain is the same as that of DNS, i.e., $(L_x \times L_y \times L_z) = (4\pi \delta \times 2\delta \times 2\pi \delta)$, and the number of grid points is $(N_x \times N_y \times N_z) = (64 \times N_y^\dagger \times 64)$, where $N_y^\dagger$ represents the number of grid points in the $y$ direction. The uniform grid is used in the $y$ direction. As the grid width in the $y$ direction, we consider three cases as summarized in Table 2. The time step in the present LES is $\Delta t^+ = 6.30 \times 10^{-2}$. We here use the constant Smagorinsky model [51] as the baseline SGS model. In the present demonstration, four cases of LES are compared as follows:

1. LES without wall models (case 1),
2. LES with van Driest’s damping function [52] (case 2),
3. LES assisted with machine-learned model, but without van Driest’s damping function (case 3),
4. LES assisted with machine-learned model and van Driest’s damping function (case 4).

The flow fields obtained by each LES are visualized using the second invariant of the velocity gradient tensor ($Q^+ = 0.005$) [53] in Fig. 7(a). Note that we only compare among cases 1, 2, and 4 with $\Delta y^+ = 20.0$ because case 3 has shown an unstable behavior of the simulation due to the low accuracy of machine-learned model trained in a priori test. These visualized fields exhibit the vortex structures in a reasonable manner. The time history of bulk Reynolds number $Re_b$ is also evaluated to investigate the correction of the simulation itself as shown in Fig. 7(b). Note that the time history of case 3 with $\Delta y^+ = 20.0$ is only shown until around...
Fig. 7 Summary of the present results in a posteriori test. (a) the second invariant of the velocity gradient tensor ($Q^+ = 0.005$), (b) the time history of bulk Reynolds number $Re_b$, and (c) the mean streamwise velocity profile.

$t^+ = 5.00 \times 10^3$ due to the unstable simulation as mentioned above. The averaged bulk Reynolds numbers provided by DNS are also shown as the gray line in each case for comparison. For each grid width, the bulk Reynolds number is not sufficiently corrected with case 3. On the other hand, case 4 is able to correct it with $\Delta y^+ = \{5.00, 10.0\}$. Note that such correction cannot be observed with $\Delta y^+ = 20.0$. This is likely caused by the low estimation ability of the model trained in a priori test. Moreover, mean streamwise velocity profiles are also compared in Fig. 7(c). Analogous to the observation in Fig. 7(b), case 4 shows its reasonable ability with $\Delta y^+ = \{5.00, 10.0\}$. Note again that the overestimation with $\Delta y^+ = 20.0$ is likely caused by the lack of estimation ability as stated above. Hence, a reasonable performance of a priori test, at least, is required for the present correction method.

To further examine the physical validity of the present LES in each case, the root-mean square (RMS) of velocity and vorticity fluctuations are summarized in Fig. 8. The results in case 4 with $\Delta y^+ = \{5.00, 10.0\}$ show closer distributions to that of the DNS compared to the other cases especially near the wall. This is likely because the SGS viscosity is corrected by applying the van Driest’s damping function, thereby leading to the correction of the RMS values near the wall. Therefore, the utilization of both the van Driest model (i.e., the physical correction for the SGS model) and the machine-learned model (i.e., the correction for the error due to discretization of boundary condition) employ well to capture near wall behavior correctly.
Fig. 8 Root-mean squared values of velocity and vorticity fluctuation in \textit{a posteriori} test.
3.3 Influence on the used SGS model in LES

As mentioned above, we have used the Smagorinsky model for the present analyses in a posteriori test. We here discuss the generalizability of the proposed machine-learning-based wall model with regard to SGS models. In addition to Smagorinsky model, let us consider two SGS models: Wall-Adapting Local Eddy-viscosity model (WALE) \[54\], and Coherent Structure Model (CSM) \[55\]. The results of time history of bulk Reynolds number and the mean streamwise velocity profile are shown in Fig. 9. The augmentation of LES can be seen with $\Delta y^+ = \{5.00, 10.0\}$ while failing with $\Delta y^+ = 20.0$, which is the same trend as the case with the constant Smagorinsky model. Therefore, the proposed machine-learning-based wall model is robust against the choice of SGS model.
3.4 Robustness of machine learning model for Reynolds numbers

Since our method relies on the training data provided by DNS as expressed in Eq. [6] of particular interest here is its capability at higher Reynolds numbers than that in its training process. Let us apply the machine-learned model trained at Re$_\tau$ = 180 to the LES at Re$_\tau$ = 360. The size of computational domain and grid points in the LES at Re$_\tau$ = 360 are ($L_x \times L_y \times L_z$) = ($2\pi\delta \times 2\delta \times \pi\delta$), ($N_x \times N_y \times N_z$) = (64 $\times N_y^* \times 64$). The time step in the present simulation is $\Delta t^+$ = 6.30 $\times 10^{-2}$. A non-uniform grid is applied in the $y$ direction at $y^+ > 40$ multiplied by a given stretch rate, while the uniform grid is considered at $y^+ < 40$. The number of grid points in the $y$ direction $N_y^*$ depends on the grid width, as summarized in Table 3. We here use the constant Smagorinsky model as the SGS model.

The performance of the present model (case 4) is assessed using the time history of bulk Reynolds number and the mean streamwise velocity profile in Fig. 10. For comparison, we also present the results of the DNS at Re$_\tau$ = 360 and case 2 which applies the van Driest function. As can be expected, the model does not work with $\Delta y^+$ = 20.0 due to the lack of the estimation ability. However, what is notable here is that the reasonable simulations can be achieved with $\Delta y^+$ = {5.00, 10.0} despite that the test Reynolds number is considered an extrapolation from the training range. The present investigation suggests that we can expect a reasonable performance of a machine-learned model even at a higher Reynolds number for the grid width where the model employs well in a training Reynolds number range.
Towards practical applications of the proposed method, the generalizability of the machine learning model in terms of the domain size of the training data is preferable. Hence, our interest here is the use of a locally trained ML model for both \textit{a priori} and \textit{a posteriori} tests. One of the techniques for local domain training is to zoom-in and/or -out target images \cite{56}. Morimoto et al. \cite{57} has recently investigated the possibility of this concept and reported the effectiveness for various fluid flow data.

Let us use this zoom-in/out concept and discuss the dependence of the LES performance on the domain size in training data. We consider four cases in terms of the domain size of the training data; half \((L_x \times L_z) = (2\pi\delta \times \pi\delta)\), one quarter \((\pi\delta \times (1/2)\pi\delta)\), and one eighth \(((1/2)\pi\delta \times (1/4)\pi\delta)\) compared to the global domain size. The machine-learning model is trained with each local domain. The constructed model is then applied to the global domain (i.e., \((L_x \times L_z) = (4\pi\delta \times 2\pi\delta)\)) in the test cases as summarized in Fig. 11. We use \(Re_t = 180\) as the training and test Reynolds number for this demonstration.

The virtual wall surface velocities with \(\Delta y^+ = 5.00\) are shown in Fig. 12. The flow fields estimated by the machine-learning models are in reasonable agreement with DNS data for all cases. On the other hand, the higher \(L_2\) errors are shown with especially with the case of one eighth. This is likely because the training data used as the input of the machine-learning model loses the low-wave number components by using zooming-in technique, which leads to decrease the estimation accuracy, since the present model dominantly estimates the low wavenumber components as discussed in Section 3.1. Moreover, the comparison of \(L_2\) error in the other \(y\) combinations are summarized in Fig. 13. The similar trends can be found with \(\Delta y^+ = \{10.0, 20.0\}\) as well as with \(\Delta y^+ = 5.00\). Therefore, we hereafter use the machine-learned models trained with the domain size of one quarter in \textit{a posteriori} test.

The results of time history of bulk Reynolds number and the mean streamwise velocity profile are shown in Fig. 14. Note that the time history with the machine-learning model trained by the local domain with \(\Delta y^+ = 20.0\) is shown until around \(t^+ = 1.00 \times 10^4\) due to the unstable simulation as the same reason mentioned in Section 3.2. Although the fluctuations can be found with time history of bulk Reynolds number compared to the case with the global training, the LES can be augmented with \(\Delta y^+ = \{5.00, 10.0\}\), although not with \(\Delta y^+ = 20.0\). Summarizing above, the present model shows the generalizability in terms of the domain size used in a training pipeline.
4 Conclusions

We assessed the performance of the machine-learning-assisted wall model for wall-resolved large-eddy simulation (LES) especially considering a coarse grid in the wall-normal direction. In order to verify our idea, we first applied it to a turbulent channel flow at Reₜ = 180. In a priori test, we constructed the machine-learning models based on convolutional neural network (CNN) that estimate the artificial slip velocity from the velocity of two x−z cross-sections in the region near the wall. The constructed model was able to estimate the virtual wall-surface velocity well compared to the linear regression method. The machine-learned model was then applied to the present LES in a posteriori test. We found that the LES with a coarse wall-normal grid of Δy⁺ = 10 can be augmented by the proposed model.

We further examined the dependence of the LES performance on the choice of SGS models and the Reynolds numbers. The robustness of the proposed model can be observed for both perspectives even if the cases are extrapolation from the training range. The generalizability of the proposed model in terms of the domain size of the training data was finally investigated, which achieves the reasonable simulation performance compared to the cases with the global training.

We have several outlooks to improve the capability of the present CNN-based velocity estimator. For example, we can consider the probabilistic neural network (PNN) to quantify the uncertainty of its estimation. This view is quite important in the present analysis where the accuracy of models trained in a
priori test highly affects the performance in a posteriori test, since the PNN can tell us how we can rely on results provided by models. Otherwise, unsupervised frameworks may also be helpful for the case where we have no solution data, as well discussed in Kim et al. [25]. Moreover, the combination with temporal prediction models, e.g., long short-term memory [45,59,60,61,62,63] and reservoir computing [64], is also a considerable path for correcting the error due to temporal discretization when the present method is used in LES with a substantially larger computational time step. Although the aforementioned extensions are just examples, we hope that the present paper is able to serve as a significant step to establish the data-driven LES wall model.

Acknowledgements

This work was supported through JSPS KAKENHI Grant Number 18H03758 by Japan Society for the Promotion of Science. We thank Mr. Mitsuaki Matsuo (Keio University) for fruitful discussions.

Declaration of interest

The authors report no conflict of interest.

References

1. P. R. Spalart. Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach. 1st AFOSR Int. Conf. on DNS/LES, 1997.
2. T. Kajishima and K. Taira. Computational fluid dynamics. Springer International Publishing, 1(1), 2017.
3. P. R. Spalart. Detached-eddy simulation. Annu. Rev. Fluid Mech., 41(1):181–202, 2009.
4. F. Hamba. An attempt to combine large eddy simulation with the $k−\varepsilon$ model in a channel-flow calculation. Theor. Comput. Fluid Dyn., 14(1):323–336, 2001.
5. U. Piomelli and E. Balaras. Wall-layer models for large-eddy simulations. *Annu. Rev. Fluid Mech.*, 34(1):349–374, 2002.

6. N. V. Nikitin, F. Nicoud, B. Wasistho, K. D. Squires, and P. R. Spalart. An approach to wall modeling in large-eddy simulations. *Phys. Fluids*, 12(7):1629–1632, 2000.

7. M. L. Shur, P. R. Spalart, M. K. Strelets, and A. K. Travin. A hybrid RANS-LES approach with delayed-DES and wall-modelled LES capabilities. *Int. J. Heat Fluid Flow*, 29(6):1638–1649, 2009.

8. S. Kawai and J. Larsson. Wall-modeling in large eddy simulation: Length scales, grid resolution, and accuracy. *Phys. Fluids*, 24(1):015105, 2012.

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

10. M. P. Brenner, J. D. Eldredge, and J. B. Freund. Perspective on machine learning for advancing fluid mechanics. *Phys. Rev. Fluids*, 4:100501, 2019.

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

12. K. Duraisamy. Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence. *Phys. Rev. Fluids*, 6:055004, 2021.

13. J. Ling, A. Kurzawaski, and J. Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariances. *J. Fluid Mech.*, 807:155–166, 2016.

14. Z. Zhang, X. D. Song, S. R. Ye, Y. W. Wang, C. G. Huang, Y. R. An, and Y. S. Chen. Application of deep learning method to reynolds stress models of channel flow based on reduced-order modeling of dns data. *J. Hydrodyn.*, 31(1):58–65, 2019.

15. P. M. Milani, J. Ling, and J. K. Eaton. Turbulent scalar flux in inclined jets in crossflow: counter gradient transport and deep learning modelling. *J. Fluid Mech.*, 906:A27, 2021.

16. N. Parashar, B. Srinivasan, and S. S. Sinha. Modeling the pressure-hessian tensor using deep neural networks. *Phys. Rev. Fluids*, 5(11):114604, 2020.

17. M. Gamahara and Y. Hattori. Searching for turbulence models by artificial neural network. *Phys. Rev. Fluids*, 2(5):054604, 2017.

18. R. Maulik and O. San. A neural network approach for the blind deconvolution of turbulent flows. *J. Fluid Mech.*, 831:151–181, 2017.

19. R. Maulik, O. San, A. Rasheed, and P. Vedula. Subgrid modelling for two-dimensional turbulence using neural networks. *J. Fluid Mech.*, 858:122–144, 2019.

20. R. Maulik, O. San, A. Rasheed, and P. Vedula. Data-driven deconvolution for large eddy simulations of kraichnan turbulence. *Phys. Fluids*, 30(12):125109, 2018.

21. R. Maulik, O. San, J. D. Jacob, and C. Crick. Sub-grid scale model classification and blending through deep learning. *J. Fluid Mech.*, 870:784–812, 2019.

22. A. Beck, D. Flad, and C. D. Munz. Deep neural networks for data-driven lees closure models. *J. Comput. Phys.*, 398:108910, 2019.

23. X. I. A. Yang, S. Zafar, J. X. Wang, and H Xiao. Predictive large-eddy-simulation wall modeling via physics-informed neural networks. *Phys. Rev. Fluids*, 4(3):034602, 2019.

24. S. Pawar, O. San, A. Rasheed, and P. Vedula. A priori analysis on deep learning of subgrid-scale parameterizations for kraichnan turbulence. *Theor. Comput. Fluid Dyn.*, 34:429–455, 2020.

25. H. Kim, J. Kim, S. Won, and C. Lee. Unsupervised deep learning for super-resolution reconstruction of turbulence. *J. Fluid Mech.*, 910, 2021.

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

27. K. Fukami, K. Fukagata, and K. Taira. Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows. *J. Fluid Mech.*, 909:A9, 2021.

28. L. Guastoni, A. Güemes, A. Ianiro, S. Discetti, P. Schlatter, H. Azizpour, and R. Vinuesa. Convolutional-network models to predict wall-bounded turbulence from wall quantities. arXiv:2006.12483, 2020.

29. B. Z. Han and W. K. Huang. Active control for drag reduction of turbulent channel flow based on convolutional neural networks. *Phys. Fluids*, 32(9):095108, 2020.

30. J. Park and H. Choi. Machine-learning-based feedback control for drag reduction in a turbulent channel flow. *J. Fluid Mech.*, 904:A24, 2020.

31. T. Nakamura, K. Fukami, and K. Fukagata. Comparison of linear regressions and neural networks for fluid flow problems assisted with error-curve analysis. arXiv:2105.00913, 2021.

32. Y. A. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Automatic early stopping using cross validation: quantifying the criteria. *Proc. IEEE*, 86:2278–2324, 1998.

33. K. Fukami, K. Fukagata, and K. Taira. Assessment of supervised machine learning for fluid flows. *Theor. Comput. Fluid Dyn.*, 34(4):497–519, 2020.

34. K. Fukami, Y. Nabe, K. Kawai, and K. Fukagata. Synthetic turbulent inflow generator using machine learning. *Phys. Rev. Fluids*, 4:064603, 2019.

35. H. Salehipour and W. R. Peltier. Deep learning of mixing by two ‘atoms’ of stratified turbulence. *J. Fluid Mech.*, 861:R4, 2019.

36. T. Murata, K. Fukami, and K. Fukagata. Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *J. Fluid Mech.*, 882:A13, 2020.

37. K. Fukami, T. Nakamura, and K. Fukagata. Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data. *Phys. Fluids*, 32:095110, 2020.
38. K. Fukami, K. Hasegawa, T. Nakamura, M. Morimoto, and K. Fukagata. Model order reduction with neural networks: Application to laminar and turbulent flows. [arXiv:2011.10277], 2020.
39. K. Fukami, T. Murata, and K. Fukagata. Sparse identification of nonlinear dynamics with low-dimensionalized flow representations. [arXiv:2010.12177], 2020.
40. M. Morimoto, K. Fukami, K. Zhang, A. G. Nair, and K. Fukagata. Convolutional neural networks for fluid flow analysis: toward effective metamodeling and low-dimensionalization. [arXiv:2101.02535], 2021.
41. K. Fukami, R. Maulik, N. Ramachandra, K. Fukagata, and K. Taira. Global field reconstruction from sparse sensors with Voronei tessellation-assisted deep learning. [arXiv:2101.09054], 2021.
42. M. Matsuoka, T. Nakamura, M. Morimoto, K. Fukami, and K. Fukagata. Supervised convolutional network for three-dimensional fluid data reconstruction from sectional flow fields with adaptive super-resolution assistance. [arXiv:2103.09020], 2021.
43. D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. [arXiv:1412.6980], 2014.
44. T. Suzuki and Y. Hasegawa. Estimation of turbulent channel flow at $Re_{\tau} = 100$ based on the wall measurement using a simple sequential approach. *J. Fluid Mech.*, 830:760–796, 2017.
45. T. Nakamura, K. Fukami, K. Hasegawa, Y. Nabae, and K. Fukagata. Convolutional neural network and long short-term memory based reduced order surrogate for minimal turbulent channel flow. *Phys. Fluids*, 33:025116, 2021.
46. Y. Morinishi, T. S. Lund, O. V. Vasilyev, and P. Moin. Fully conservative higher order finite difference schemes for incompressible flow. *J. Comput. Phys.*, 143(1):90–124, 1998.
47. P. R. Spalart, R. D. Moser, and M. M. Rogers. Spectral methods for the Navier-Stokes equations with one infinite and two periodic directions. *J. Comput. Phys.*, 96(2):336–347, 1991.
48. J. K. Dukowicz and A. S. Dvinsky. Approximate factorization as a high order splitting for the implicit incompressible flow equations. *J. Comput. Phys.*, 102(2):336–347, 1992.
49. I. Scherl, B. Strom, J. K. Shang, O. Williams, B. L. Polagye, and S. L. Brunton. Robust principal component analysis for modal decomposition of corrupt fluid flows. *Phys. Rev. Fluids*, 5:054401, 2020.
50. P. Peterson. F2py: A tool for connecting fortran and python programs. *Int. J. Comput. Sci. Eng.*, 4(4):296–305, November 2009.
51. J. Smagorinsky. General circulation experiments with the primitive equations: I. the basic experiment. *Mon. Weath. Rev.*, 91(3):99–164, 1963.
52. E. R. Van Driest and J. C. Boison. Experiments on boundary-layer transition at supersonic speeds. *J. Aero. Sci.*, 24(12):885–899, 1957.
53. M. S. Chong, A. E. Perry, and B. J. Cantwell. A general classification of three-dimensional flow fields. *Phys. Fluids*, 2(5):765–777, 1990.
54. F. Nicoud and F. Ducros. Subgrid-scale stress modelling based on the square of the velocity gradient tensor. *Flow Turbul. Combust.*, 62(3):183–200, 1999.
55. H. Kobayashi. The subgrid-scale models based on coherent structures for rotating homogeneous turbulence and turbulent channel flow. *Phys. Fluids*, 17(4):045104, 2005.
56. A. Mikolajczyk and M. Grochowski. Data augmentation for improving deep learning in image classification problem. In 2018 International Interdisciplinary PhD Workshop (IIPhDW), pages 117–122, 2018.
57. M. Morimoto, K. Fukami, K. Zhang, and K. Fukagata. Generalization techniques of neural networks for fluid flow estimation. [arXiv:2011.11911], 2020.
58. R. Maulik, K. Fukami, N. Ramachandra, K. Fukagata, and K. Taira. Probabilistic neural networks for fluid flow surrogate modeling and data recovery. *Phys. Rev. Fluids*, 5:054401, 2020.
59. P. A. Srinivasan, L. Guastoni, H. Azizpour, P. Schlatter, and R Vinuesa. Predictions of turbulent shear flows using deep neural networks. *Phys. Rev. Fluids*, 4:054603, 2019.
60. H. Eivazi, L. Guastoni, P. Schlatter, H. Azizpour, and R. Vinuesa. Recurrent neural networks and koopman-based frameworks for temporal predictions in a low-order model of turbulence. *Int. J. Heat Fluid Flow*, 90:108816, 2021.
61. K. Hasegawa, K. Fukami, T. Murata, and K. Fukagata. Data-driven reduced order modeling of flows around two-dimensional bluff bodies of various shapes. *ASME-JSME-KSME Joint Fluids Engineering Conference, San Francisco, USA*, (Paper 5079), 2019.
62. K. Hasegawa, K. Fukami, T. Murata, and K. Fukagata. Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes. *Theor. Comput. Fluid Dyn.*, 34(4):367–388, 2020.
63. K. Hasegawa, K. Fukami, T. Murata, and K. Fukagata. CNN-LSTM based reduced order modeling of two-dimensional unsteady flows around a circular cylinder at different Reynolds numbers. *Fluid Dyn. Res.*, 52(6):065501, 2020.
64. S. Pandey and J. Schumacher. Reservoir computing model of two-dimensional turbulent convection. *Phys. Rev. Fluids*, 5(11):113506, 2020.