博 士 学 位 論 文

論文題目　Turbulence Modeling for Large Eddy Simulation by Data-Driven Approach

提　出　者　　東北大学大学院情報科学研究科

応用情報科学　　専　攻

学籍番号　　C0ID4502

氏　名　　Golsa Tabe Jamaat

TOHOKU UNIVERSITY
Graduate School of Information Sciences

Turbulence Modeling for Large Eddy Simulation by Data-Driven Approach (データ駆動科学の手法によるラージ・エディ・シミュレーションのための乱流モデリング)

A dissertation submitted for the degree of Doctor of Philosophy
(Information Sciences)

Department of Applied Information Sciences

by

Golsa TABE JAMAAT

July 7, 2023

Turbulence Modeling for Large Eddy Simulation by Data-Driven Approach

Golsa TABE JAMAAT

Abstract

The numerical investigation of the turbulent flows is of cardinal importance as they can be found almost everywhere either in nature or industries. Therefore, it is crucial to conduct reliable simulations for such flows and look for numerical approaches with reasonable accuracy and computational cost. The direct numerical simulation (DNS) which is the most accurate method for the simulation of turbulent flows is computationally extremely expensive. In this method, all scales of turbulence are resolved. Therefore, it is impossible to use it for very high Reynolds number flows due to the extremely fine grid requirement for such flows. The other approach which is computationally more efficient than the DNS is called large eddy simulation (LES). In this approach, filtering is applied to the Navier-Stokes equations which enables us to use a coarser grid size than the DNS due to the filtering operation. As a result of filtering, a term called subgrid-scale (SGS) stress appears in the filtered Navier-Stokes equations which needs to be modeled. In order to have an accurate simulation using the LES approach, it is crucial to use an SGS model with sufficient accuracy and reasonable cost. Over the years, numerous studies have been carried out with the goal of finding an accurate SGS model; however, the efforts are still going on as there is no SGS model found yet with reasonable performance in any kind of flow. The other issue with the LES is related to its applications for wall-bounded flows. Although LES is a computationally efficient approach, its cost rises significantly for the wall-bounded flows as the grid must be sufficiently fine in the near-wall region to capture the small flow structures. The situation becomes worse for the very high Reynolds numbers, and it has been shown that for these flows about 99 percent of the grid points in LES are used for the inner-layer which can make the LES impossible for the simulation of the wall-bounded flows at very high Reynolds numbers. One solution to overcome this problem is to use wall models which can make it feasible to use a coarse grid size in the near-wall region by modeling it. However, in this case, it is important to use a reasonably accurate wall model to have a reliable simulation. In the past years, various wall models and approaches have been proposed for the wall-modeled LES. One of the proposed approaches is the hybrid Reynolds-averaged Navier-Stokes (RANS)-LES method which uses the RANS approach for the inner layer and LES for the outer layer. In the second approach, wall-stress model is used to model the near-wall region. In fact, the wall-stress models propose a model for estimating the wall shear-stress and apply it to the wall as the boundary condition. However, the presented wall models have also shown to not be successful for all kinds of flows. For instance, they may fail under strong non-equilibrium conditions. Thus, the studies still continue to find wall models with better performance and generalizability.

In the recent years, with the improvement in the computational resources, the availability of the DNS data, and the proven ability of the machine learning methods in making predictions, it has become of interest to use them for turbulence modeling. Although there have been numerous studies on the data-driven SGS modeling, there are some deficiencies with the developed models. For instance, the developed models usually are computationally expensive and suffer from numerical instability when used without any additional treatment. Thus, it is required to perform studies with the aim of finding an efficient accurate data-driven SGS model. The data-driven methods have been used for the wall modeling in LES after the SGS modeling. Therefore, there are not many studies yet on this topic compared to the data-driven SGS modeling. In this dissertation, two separate studies have been performed on the SGS modeling and wall modeling in LES using data-driven methods.

In Chapter 2, the objective is to find a data-driven SGS model for the Burgers equation which can be generalized to various conditions and be used for large filter sizes. The reason for choosing the Burgers equation are its similarities to the Navier-Stokes equations and the possibility of using a very fine grid size for the DNS and consequently, applying a very large filter size to the obtained data which is not possible in higher dimensional flows due to the significantly high cost of simulations for those flows. The results obtained in this chapter show that the successful fully connected neural network model can be generalized to various conditions, including the Burgers equation with different forcing terms, viscosities, and grid sizes. The performance of the obtained model is comparable to the dynamic Smagorinsky model (DSM) and even in some cases it is better while its computational cost is much lower than the DSM.

In Chapter 3, the objective is to investigate the ability of the convolutional neural network (CNN) in establishing a wall model for the turbulent channel flow and to present the results of the *a priori* test. CNN has already shown its ability in extracting hidden features and has been widely used in various applications like image recognition, but it has not been used for the wall modeling in LES. Therefore, it can be an appropriate tool for finding a wall model. The results of the *a priori* test presented in this chapter show that the obtained CNN wall model has a reasonable accuracy in predicting the wall shear stress, better than two existing wall models (shifted model and ODE-based model). The model has been tested extensively in the *a priori* test and been applied to higher Reynolds number cases and coarser grid sizes. The results show that the model is successful in predicting the wall shear stress. However, among the tested conditions it has shown more sensitivity to a very coarse grid size which can be expected since the CNN model gets the input data from a two-dimensional domain.

In Chapter 4, *a posteriori* test is performed by embedding the CNN wall model trained in Chapter 3 in an actual LES. In this chapter, the performance of the model is compared with the ODE-based wall model. The results of this chapter show that the CNN wall model is successful in predicting the mean flow and its performance is comparable to the ODE based wall model. However, in some cases, it slightly underpredicts

the Reynolds stress more than the ODE-based the wall model. The results of the distribution and the probability density function (PDF) of the wall shear stress for the CNN and ODE-based wall models show that for the CNN wall model, the results are more similar to those of the filtered DNS and unlike the ODE-based wall model, the wall shear stress is poorly correlated with the velocity at the matching location where the wall model inputs are provided which is in agreement with the filtered DNS results.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Turbulent flow simulation approaches

Turbulent flows can be seen almost everywhere either in the nature or engineering applications like the water flow in the rivers and pipes. Therefore, it is crucial to simulate such flows using an accurate method with reasonable computational cost. The turbulent flows are unsteady and chaotic, and are characterized by the irregular pattern in their motions. The velocity field in such flows changes in both space and time. There are mainly three different approaches for the numerical simulation of turbulent flows. The most accurate approach is the direct numerical simulation (DNS). In this approach, all scales of motions need to be resolved and since turbulent flows consist of a wide range of length scales, the computational grid must be very fine to resolve all scales of motions which makes this approach inapplicable to the high Reynolds number flows because of the extremely fine grid requirement and the limitations in the computational resources.

The other technique for the simulation of turbulent flows is the Reynolds-averaged Navier-Stokes (RANS). In contrast to DNS, in this method all scales of motions are modeled and the RANS equations are solved for the mean velocity field. In these equations, the Reynolds stress term appears in the equations as the result of averaging which needs to be modeled. The Reynolds-stress terms are usually modeled using the turbulent viscosity models [4]. In such models, the turbulent viscosity can be based either on an algebraic equation like the mixing-length model or can be obtained using the turbulence quantities like the $k - \epsilon$ model which also depends on the empirical coefficients which can be varied depending on the flow type. The third method for simulating turbulent flows is the large eddy simulation (LES) which lies between DNS and RANS in terms of accuracy and cost. The energy spectrum and the scales of motions which can be resolved in each approach are shown in Fig 1.1. In the next section (Sec. 1.2), LES will be explained in detail.

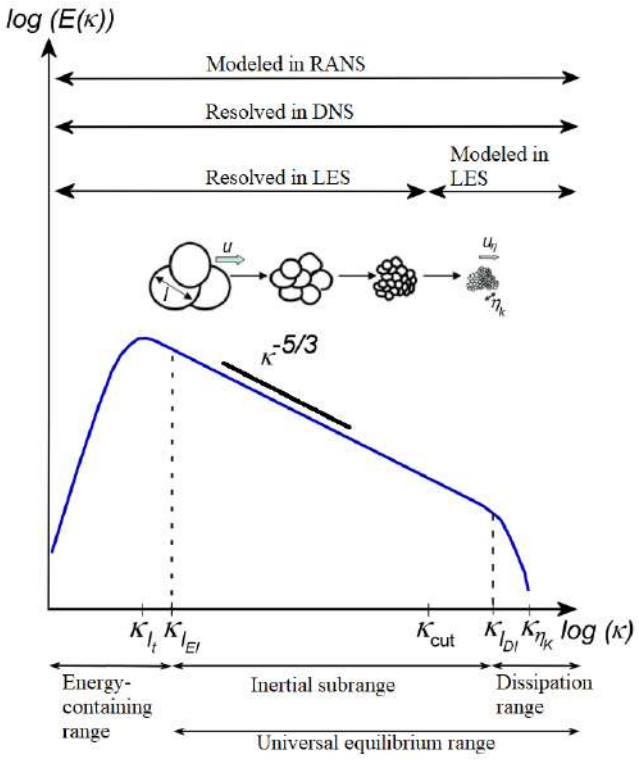


Figure 1.1. Energy spectrum [1, 2]. The range of resolved and modeled scales for RANS, DNS, and LES approaches.

1.2 Large eddy simulation

Large eddy simulation (LES) is considered as a powerful approach for numerical simulation of turbulent flows since in comparison to direct numerical simulation (DNS), a coarser grid can be used, and its computational cost is much lower. Additionally, compared to Reynolds averaged Navier-Stokes (RANS) approach, it is more accurate and is able to capture unsteady flow structures. In LES, the flow field is decomposed to the grid-scale flow field and the subgrid-scale (SGS) fluctuations, and the filtered Navier-Stokes (NS) equations are solved for the grid-scale flow field. The effect of SGS fluctuations appears as SGS stress in the filtered NS equations and needs to be modeled.

1.2.1 Subgrid-scale modeling in LES

There are a number of models proposed for modeling the SGS stress including the Smagorinsky model [5], the gradient model [6], the scale-similarity model, the mixed model [7], and the dynamic versions of these models [8–10]. The Smagorinsky model is too dissipative as it provides no backward scatter of SGS energy into the grid-scale; on the other hand, the scale-

similarity model is not sufficiently dissipative because of giving too much backward scatter. The gradient model has a good accuracy in predicting the SGS stress for the small grid size, but it becomes unstable as the grid size increases since it is insufficiently dissipative and does not have a correct prediction of the energy transfer. The dynamic versions of the SGS models provide a better accuracy, but they are computationally expensive. In addition to these well-known models, some other approaches and models have been proposed for LES of the one-dimensional Burgers equation [11–13] and three-dimensional turbulence [14–17]. However, there is no model which is accurate for any flows. Therefore, it is crucial to find a computationally efficient SGS model with reasonable accuracy.

1.2.2 Wall modeling in LES

Although LES can be generally considered as a computationally efficient method, its computational cost increases significantly for the wall-bounded flows, especially, at high Reynolds numbers as the grid near the wall must be sufficiently fine to capture the turbulence scales in the near-wall region. Piomelli and Balaras [3] have estimated that in the wall-resolved LES (WRLES), 99% of the points are used for resolving the inner layer at $Re_L = O(10^6)$, which can make the LES computationally inapplicable to the high-Reynolds-number flows. The number of grid points required for resolving the inner layer and the outer layer also confirms that the total number of grid points is dominated by the number of points required for resolving the inner layer when the Reynolds number becomes very large as shown in Fig. 1.2 [3]. In order to avoid the fine grid requirement, wall-modeling is usually employed to reduce the cost and the grid points required for the LES considerably by modeling the near-wall region. There have already been studies on the grid point requirements for LES [18, 19]. The most recent study by Yang and Griffin [20] revealed that the total cost of DNS, WRLES and wall-modeled LES (WMLES) scale as $Re_{L_x}^{2.91}$, $Re_{L_x}^{2.72}$, and $Re_{L_x}^{1.14}$, respectively, which further confirms the much lower cost of WMLES compared to DNS and LES.

In the wall-modeled LES, it is necessary to model the effects of the near-wall region since due to the coarse grid, the sharp velocity gradient and consequently, the momentum flux at the wall cannot be calculated directly using discrete differentiation. A schematic of the near-wall grid resolution for the wall-resolved LES and the wall-modeled LES is shown in Fig. 1.3. When the grid is coarse enough to contain a large number of eddies, only the average effect of the

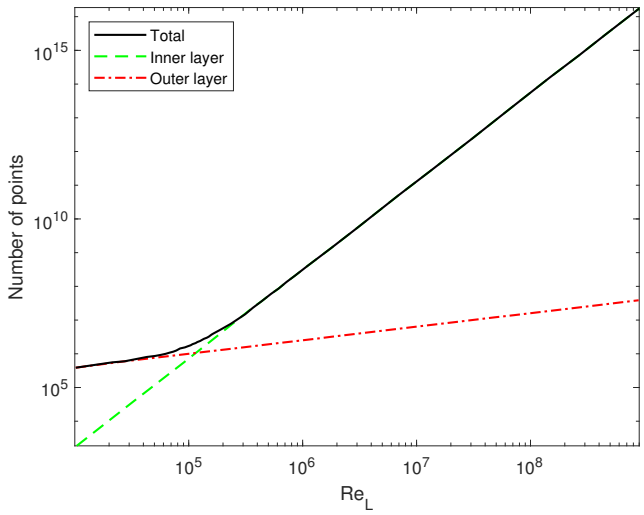


Figure 1.2. Dependency of the number of grid points in the inner layer and the outer layer on Reynolds number [3].

eddies needs to be presented by the wall model. In this case, it can be possible to use the Reynolds-averaged Navier-Stokes (RANS) equations in the near-wall region. This assumption is considered to be true for a very large grid size, the order of 1500 and 700 wall units in the streamwise and spanwise directions [3].

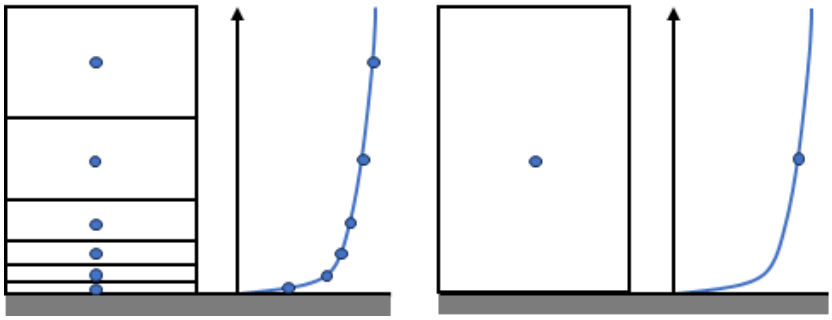


Figure 1.3. Schematic of near-wall grid points for (a) wall-resolved LES and (b) wall-modeled LES.

The wall-modeling approaches in LES are mainly divided into two groups [21]. The first one is the approximate boundary condition [22–25] which can be based either on mathematical foundation or on physical assumption. In this method, the LES is carried out for the whole computational domain. The inner layer of the boundary layer is modeled and the wall shear stress is usually applied as the boundary condition for LES at the wall [26–29]. This makes it possible to use a coarser grid in the near-wall region and leads to a significant decrease in the computational cost. The conventional wall-stress models are usually derived using relations such as law of the wall which are typically based on the mean flow. Furthermore, despite the fact that

the wall shear stress is poorly correlated with the velocity at the sampling locations [30], the predicted wall shear stress by the conventional wall-stress models is usually perfectly correlated with the velocity [31, 32]. The other wall modeling approach is the hybrid LES/RANS which solves the RANS equations in the inner layer and employs LES far from the wall [33–37].

Although the wall modeling in LES is quite popular, there are still some important issues with it that need to be resolved. For instance, the conventional approximate boundary conditions are generally limited to the flows with the simple geometries and cannot be applied to the complex geometries in general as the wall shear stress in these models is obtained using some form of the logarithmic law of the wall which may not be valid for the complex geometries [3]. Furthermore, there is a possibility that the equilibrium stress models fail in a nonequilibrium flow. On the other hand, the problem with the approaches based on the RANS equations is their dependency on the the empirical coefficients. In fact, these approaches depend on the different parameters which are not unique for all types of flow and must be obtained empirically depending on the state of the flow [32]. Therefore, the efforts still continue to propose wall models with better accuracy. In general, the wall modeled LES are shown to have log-layer mismatch which can be due to the wall model and the implemented LES method [21].

1.3 Data-driven turbulence modeling

In recent years, due to the increase in computational resources and consequently, the availability of more DNS data, and the ability of the machine learning (ML) approaches in making predictions [38, 39], the desire for developing data-driven turbulence models for both RANS and LES has increased. There have been various efforts on the data-driven turbulence modeling in RANS method [40]. Ling et al. [41] used deep neural network to find a model for the Reynolds stress anisotropy tensor. Fang et al. [42] proposed modifications to a standard neural network to model the Reynolds stress closure for the turbulent flow in a channel. Sanhueza et al. proposed a method based on machine learning to enhance the accuracy of conventional RANS turbulence models in channel flows that experience significant fluctuations in their thermophysical properties [43]. Wu et al. proposed a physics-based implicit treatment to model Reynolds stress by using machine learning techniques [44]. There have also been various studies on the turbulence modeling in LES which will be explained in detail in this section.

1.3.1 Subgrid-scale modeling in LES using data-driven approaches

Like the efforts on developing data-driven Reynolds-stress models for the RANS approach, there have been studies on establishing machine learning-based SGS models in LES. Sarghini et al. [45] used a two hidden layer neural network as a substitute for Bardina's scale-similarity SGS model. They showed that the neural network SGS model is faster than the scale-similar SGS model; however, switching to a higher Reynolds number requires a novel training. Maulik and San [46] proposed an approach for the deconvolution of the spatially filtered flow variables using an artificial neural network. Gamahara and Hattori [47] tested a neural network as a tool for finding a new model for the SGS stress in LES of a channel flow and showed that the obtained model is similar to the gradient model. Wang et al. [48] used two algorithms of random forests and neural network to establish SGS models for LES and showed that the neural network algorithm has better performance than the random forests algorithm for the regression problem. Zhou et al. [49] chose the velocity gradient tensor together with the filter width as the input features to obtain a new SGS model for LES of isotropic flows using a single hidden layer neural network. Beck et al. [50] used the DNS data of the decaying homogeneous isotropic turbulence to train a deep neural network for the turbulence modeling in LES. Xie et al. [51] used the first-order derivative of the filtered velocity at different spatial locations as the input to develop a model for the SGS forces using spatial artificial neural network. Yuan et al. [52] constructed a model for the SGS stress in LES using a deconvolutional neural network. They used filtered velocities at different spatial points as input features and showed that the new trained model can predict the velocity statistics well for different filter widths without any fine-tuning. Miyazaki and Hattori [53] developed an SGS model for the LES of homogeneous isotropic flow using neural network. They showed that weighting the training data and considering the second-order derivative of the velocity as an additional input variable can improve the accuracy of the model, and the established models are close to the gradient models. Xie et al. [54] developed neural network nonlinear algebraic models to reconstruct the SGS stress in LES of turbulence. Park and Choi [55] established a new model for the SGS stress using a fully connected neural network. They showed that despite yielding accurate results in the *a priori* test, considering the filtered strain rate or the velocity gradient tensor at multiple points as an input does not provide a stable solution in the *a posteriori* test unless backscatter clipping is used. There are also some other studies on the SGS modeling in LES using data-driven approaches [56–60].

1.3.2 Wall modeling in LES using data-driven approaches

Although there have been numerous efforts on the subgrid-scale (SGS) modeling in LES for the Burgers turbulence [61, 62], homogeneous isotropic turbulence [48, 49, 52–54] and three-dimensional turbulent channel flow [47, 55] using data-driven approaches, there are not many attempts on developing a wall model for LES using machine learning methods. Yang et al. [63] incorporated the physical insights to obtain a wall model for LES of channel flow using fully connected neural networks (FCNN). In their study, they chose the combination of the wall-parallel velocity and the wall-normal distance as the inputs of the neural network and showed that considering the filter size as an additional input does not make any significant changes in the performance of the wall model. They compared the obtained wall model with the equilibrium wall model and showed that both models are successful in capturing the law of the wall, but the overall performance of the FCNN wall model is slightly better. Radhakrishnan et al. [64] used the gradient boosted decision trees to develop a wall model for the LES of turbulent channel flow. They showed that the performance of their model is comparable to that of the equilibrium wall model. Zhou [65] et al. used the WRLES data to develop a wall model for LES of the turbulent flow over periodic hills using FCNN. In the *a priori* test, they showed that the FCNN model in general has reasonable accuracy; however, it produces low correlation coefficient between the wall shear stress calculated using the wall-resolved LES data and predicted by the wall model and large error near the crest of the hill. They also tested the FCNN model for the turbulent channel flow and the flow over periodic hills in the *a posteriori* test. Their results show that the FCNN wall model is able to capture the law of the wall for the turbulent channel flow, however, it is not successful in predicting the mean velocity for the flow over periodic hills. Bae and Koumoutsakos [32] developed two types of wall models, the velocity-based wall model and the log-law-based wall model, using scientific multi-agent reinforcement learning (SciMARL). They showed that the velocity-based wall model does not give a good prediction at high Reynolds numbers when the non-dimensionalized wall-normal distance for the wall model input does not fall within the training range. Additionally, they pointed out that the results obtained using their model are comparable to those of the equilibrium wall model. Vardot et al. [66] further improved the development of wall modeling using reinforcement learning and showed that the obtained model is capable of recovering the law of the wall. There have also been several other studies in which either the reinforcement learning or FCNN is used for wall modeling in LES [67–69].

1.4 Objective and outline of dissertation

Although there are many SGS models proposed using neural network for the 3D turbulence, it is still a great challenge to find a model which requires a lower computational cost compared to the existing models and can be applied to any flows without using any stabilization techniques. Since the computational cost rises as the number of neurons and hidden layers increases, it is crucial to consider the possible choices for the inputs and find a model using optimized numbers of neurons and hidden layers. Another challenge which can be proposed for the SGS models in LES is their applicability to large filter widths. Besides data-driven SGS modeling in LES, making efforts to develop wall models for LES using data-driven approaches can be helpful to find a model which can predict the wall shear stress with reasonable accuracy. There have been some studies on the wall modeling in LES using data-driven approaches like the fully-connected neural network and reinforcement learning [32, 63]. However, since this research topic is almost new, there are still other machine learning approaches like convolutional neural network which can be tested to investigate the applicability of the data-driven approaches further in establishing wall model with the aim of finding a better wall model.

The first objective of this dissertation is to establish data-driven SGS models for the LES of the Burgers turbulence, which has good generalizability and can be used for the large filter sizes with minimum numerical costs. In this part, FCNN is used as a tool to find an SGS model. A schematic of FCNN is shown in Fig. 1.4. As the DNS data are often used for establishing an SGS model using data-driven approaches to make the model more reliable, it is necessary to have a high resolution grid size for the DNS data in order to make it possible to choose a large filter size. Since it is computationally extremely expensive to perform DNS for very high Reynolds number for the 3D turbulence, the Burgers equation can be a good candidate for this study, which is computationally less expensive compared to the 3D turbulence. While this equation is considered as a one-dimensional analog of the NS equations due to its quadratic nonlinear convection term and its similar inertial range to that of the NS equation, the formation of the strong shocks at high Reynolds numbers makes it a challenging system. The Burgers equation has previously been used in numerous studies, other than the SGS modeling using data-driven approaches, as a test bed for evaluating the behavior of the SGS models in LES. Love [70] used the Burgers equation to investigate the performance of the SGS models at high Reynolds numbers. Basu [71] examined the ability of the eddy-viscosity SGS models in capturing the

inertial range properties of the stochastic Burgers equation. Li and Wang [72] performed *a priori* and *a posteriori* tests to evaluate the performance of the SGS models for decaying Burgers equation at high Reynolds numbers using third-order flux reconstruction or correction procedure via reconstruction scheme. Their results show that in both tests the scale-similarity model and the mixed model have better performance than the (dynamic) Smagorinsky model and the linear unified RANS-LES model. Maulik and San [73] carried out an *a posteriori* test on the LES of the Burgers equation using explicit and implicit approaches. Their results reveal that the performance of the explicit approaches depends on the filter type and involving parameter in the filter as well as the SGS model coefficient. They also showed that the implicit approaches converge to the DNS as the grid size becomes finer. Additionally, their study demonstrates the ability of the relaxation filtering and the compact reconstructed weighted essential non-oscillatory (CRWENO) schemes in producing accurate results without energy pile-up.

In this study, the effect of the input choices on the accuracy of the neural network SGS models is investigated as well. For the data-driven models, it is quite important to ensure the generalizability of the models, since in some cases, due to overfitting, the models do not produce accurate results when applied to the unseen data despite yielding good results for the cases whose data are used for training. Therefore, in this study, the data-driven SGS models are tested extensively for the equations under different conditions to verify their performance.

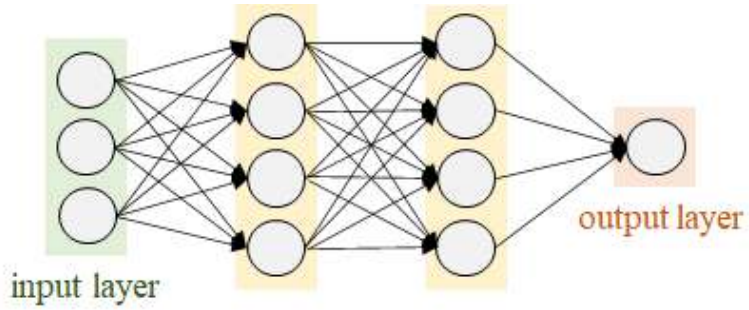


Figure 1.4. Schematic of FCNN.

The second objective of this thesis is to investigate the ability of the convolutional neural network (CNN) in establishing a wall-stress model for LES of turbulent channel flow. A schematic of the turbulent channel flow and the vortical structure of such flows are shown in Fig. 1.5. CNN is a nonlocal approach and has originally been developed for image recognition [74]; however, due to its ability in prediction and extracting features, it can be a good tool for the turbulence modeling. Moriya et al. [75] developed a model using CNN which estimates the

virtual wall velocity for the LES of turbulent channel flow. In their approach, the first grid point is still chosen close to the wall ($y^+ \leq 10$). The purpose of their model is to save the computational cost of LES while still resolving the near-wall region. Guastoni et al. [76] developed two data-driven models using CNN for the prediction of velocity fluctuations at different distances off the wall in an open channel flow. One of their models is solely based on the CNN and the other one used both the proper orthogonal decomposition (POD) and CNN. They showed that their models are in a good agreement with the DNS data. Zhao et al. [77] proposed a super-resolution reconstruction model using the meta-learning deep CNN for the subgrid-scale turbulent flow in LES. They showed that the obtained model is able to recover the turbulence small-scale information effectively using the data of low-resolution LES. Liu et al. [78] developed a subgrid-scale model for LES using CNN by choosing the primitive flow quantities as the CNN inputs which are simpler than the inputs generally selected for the FCNN-based SGS models (e.g. $\partial u_i/\partial x_j$). They compared the CNN-based SGS model with the FCNN-based SGS model developed using the velocity gradient tensor as the input and showed that the CNN model has better performance compared to the FCNN model while requiring less computational cost. Beck et al. [50] used CNN to develop an LES closure model for three-dimensional homogeneous isotropic turbulence. Their results show that the CNN model is successful in predicting the closure term and the deeper networks lead to a better prediction. Furthermore, there have been several studies on SGS modeling using CNN for large eddy simulation of two-dimensional turbulence [79–83]. Although the convolutional neural networks has been used in various studies on LES, especially for the SGS modeling [50, 60, 78, 80, 81, 84] , it has not been applied to the wall modeling in LES to the best of our knowledge. Thus, it is worth trying to develop a CNN wall model which predicts the wall shear stress as the output of the CNN with improved accuracy as CNN succeeded in modeling SGS stress.

This dissertation is organized as follows. In Chapter 2, the SGS modeling is performed for the forced Burgers turbulence using a data-driven approach called fully-connected neural network. In Chapter 3, *a priori* test is performed on the wall modeling for LES of turbulent channel flow using convolutional neural network. In Chapter 4, *a posteriori* test is conducted for the established CNN wall model. Finally, the conclusions of this dissertation are presented in Chapter 5.

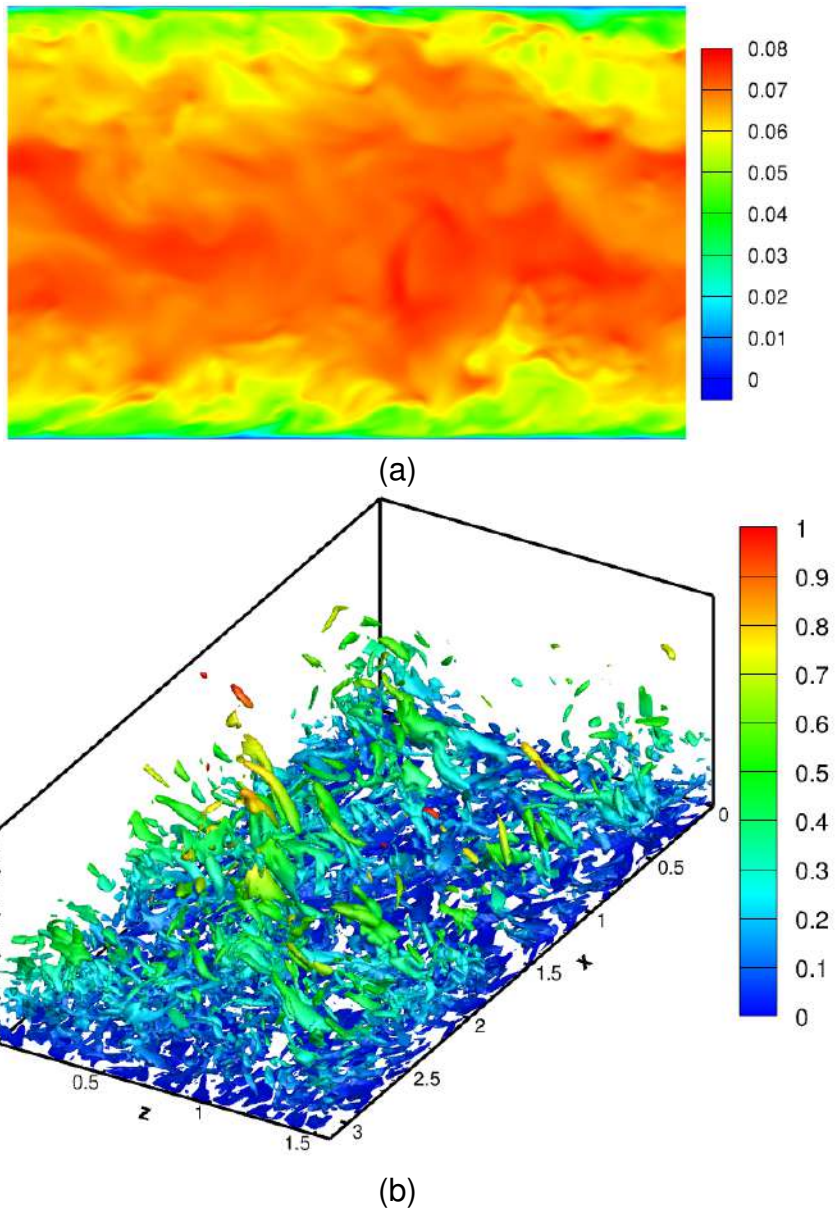


Figure 1.5. Flow field and vortical structure of turbulent channel flow obtained from DNS at $Re_\tau = 600$. (a) streamwise velocity, and (b) Isosurface of the second invariant of the deformation tensor (Q=0.02). The legends in (a) and (b) are streamwsie velocity (u) and wall-normal distance (y), respectively.

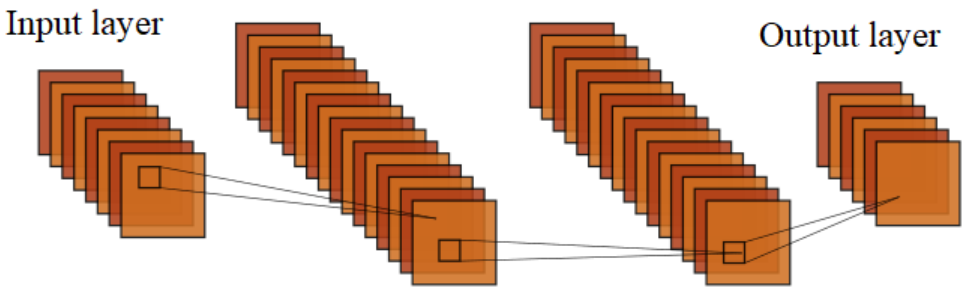


Figure 1.6. Schematic of CNN.

2. Data-driven SGS modeling for Burgers turbulence

2.1 Introduction

Although there have been many SGS models proposed during the years, the efforts still continue to find a better SGS model than the existing models which is computationally efficient. One approach for finding a proper closure model is to directly establish a functional relationship between the grid-scale flow field and the SGS stress using a data-driven approach without any assumptions. The data required for such methods can be provided by the DNS results which makes the obtained model more reliable.

In the data-driven approaches, it is crucial to have sufficient datasets for the training process to obtain a reliable model. For the three-dimensional turbulence, especially at high Reynolds numbers, a huge number of grid points is required for performing the DNS, which makes it quite difficult to collect adequate data for training a model. Therefore, the Burgers equation has always been proposed as a good choice for developing new methods since it shares similarities with the Navier-Stokes equations while its low computational cost makes it possible to provide sufficient data for the data-driven approaches. The Burgers equation has frequently been used for testing and developing new data-driven approaches. Bassenne and Lozano-Duran [85] analyzed the applicability of the reinforcement learning (RL) in model discovery by considering the Burgers equation for finding the analytic form of the missing terms. Xie et al. [86] used the Burgers equation to test the data-driven model for the closure term in the reduced order model (ROM) obtained by residual neural networks (ResNet). Alcala and Timofeyev [87] used generative adversarial networks (GAN) to develop a framework for subgrid-scale parametrization in the Burgers equation. Manrique de Lara and Ferrer [88] used neural network to reduce the computational cost of high-order discontinuous Galerkin method. Despite using the Burgers equation in various studies on developing data-driven approaches, there is not much data available for the SGS modeling of the Burgers equation using data-driven approaches. Subel et al. [61] focused on establishing a neural network SGS model for the LES of the Burgers equation which can be generalized to the higher Reynolds numbers via transfer learning. They considered a neural network architecture including six hidden layers, with 250 neurons each, and showed that

the obtained model produces a stable and accurate solution in the *a posteriori* test. It should be noted that the computational cost of a neural network model is significantly affected by the number of neurons and the hidden layers. Therefore, this large number of neurons and hidden layers can make the model computationally expensive. Furthermore, they performed DNS with 8192 grid points, which is not large compared to the present study. Similar to the study by Subel et al. [61], in the other studies on the Burgers equation, the number of grid points has been small except for Ref. [73], where the number of grid points is 32768. Basu [71], and Li and Wang [72] solved the Burgers equation using 8192 modes. In the papers by Das and Moser [11] and LaBryer et al. [12], 2048 grid points are chosen. In Refs. [61, 89], 1024 modes are considered. We emphasize that in the present study using a large number of grid points not only makes the grid sufficiently fine to even capture the strong shocks in the domain well, but also gives the chance to choose a filter size which is much more larger than the grid size. Therefore, the obtained results will give invaluable knowledge for turbulence modeling for large filter widths, which has not been done in the previous works.

In this chapter, the objective is to find an SGS model for the Burgers turbulence which can be used for large filter sizes and has reasonable accuracy, low computational cost, and generalizability to any condition without using any stabilization method in the actual simulation. The dataset used for training the models is provided by performing high-resolution DNSs in order to make it possible to apply large filter widths to the data, which is computationally extremely expensive for the 3D turbulence. In this study, initially, the training dataset is collected. Then, the parametric study will be performed on the input choices and neural network hyperparameters. In the next step, the model is tested in the *a priori* test and the performance of the neural network SGS model is compared with the SGS model based on random forest. The effect of input choices is investigated extensively as well. After that, the models will be tested in the *a posteriori* test by applying the models into an actual LES code. Finally, the performance of the models will be checked for the Burgers equation with different forcing functions and decaying Burgers turbulence.

2.2 Numerical methods

The general form of the forced Burgers equation is given by

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} + f(x,t), \tag{2.1}$$

where u is the velocity field, ν is the kinematic viscosity, and f is a forcing term. The forcing function used in this study is of random type, and in the Fourier space it has a general form of

$$\hat{f}(k) = |k|^{-\frac{\beta}{2}}\hat{\sigma}(k), \quad |k| \leq k_f, \tag{2.2}$$

where k_f denotes the highest forced wavenumber, β is the spectral slope of the noise and $\hat{\sigma}_k$ indicates the Fourier transform of the Gaussian random variable with zero mean and standard deviation of $\sqrt{N}$, where N is the number of grid points. The Burgers equation in the Fourier space is

$$\frac{d\hat{u}(k)}{dt} = -N(k) - \nu k^2\hat{u}(k) + \hat{f}(k). \tag{2.3}$$

where $N(k)$ is the Fourier transform of the nonlinear term. The spatially filtered version of the Burgers equation is

$$\frac{\partial \bar{u}}{\partial t} + \bar{u}\frac{\partial \bar{u}}{\partial x} = \nu\frac{\partial^2 \bar{u}}{\partial x^2} + \bar{f}(x,t) - \frac{1}{2}\frac{\partial \tau}{\partial x}, \tag{2.4}$$

where overbar denotes filtered variables. τ in Eq. (2.4) shows the SGS stress which is defined as

$$\tau = \overline{uu} - \bar{u}\bar{u}. \tag{2.5}$$

The zero initial condition and the periodic boundary condition are used for solving this equation, and the domain length is 2π. The Burgers equation is solved using the Fourier spectral method, and the fourth-order Runge-Kutta scheme is employed for time-marching

Table 2.1. DNS datasets used for training.

Case	β	ν
DNS1	-0.75	8×10^{-6}
DNS2	0	1×10^{-5}
DNS3	1	2×10^{-5}

$$
\begin{aligned}
&\frac{\partial \bar{u}}{\partial t} = RHS(\bar{u}) \\
&\bar{u}^{(1)} = \bar{u}^n + \frac{\Delta t}{2} RHS(\bar{u}^n) \\
&\bar{u}^{(2)} = \bar{u}^n + \frac{\Delta t}{2} RHS(\bar{u}^{(1)}) \\
&\bar{u}|^{(3)} = \bar{u}^n + \frac{\Delta t}{2} RHS(\bar{u}^{(2)}) \\
&\bar{u}^{n+1} = \bar{u}^n + \frac{\Delta t}{6}(RHS(\bar{u}^n) + 2RHS(\bar{u}^{(1)}) + 2RHS(\bar{u}^{(2)}) + RHS(\bar{u}^{(3)})).
\end{aligned}
\tag{2.6}
$$

The convective term is computed in the physical space and dealiasing is performed in the Fourier space using the $2/3$ rule.

The DNS is performed for three different values of ν and β in Eq. (2.2) on 65536 grid points ($\Delta = 2\pi/65536$) as shown in Table 2.1, with the time step of 2.5×10^{-5}. The effective maximum wavenumber, k_{max}, is $N/3 = 21845$. In this study, the number of Fourier modes is considerably high which makes it feasible to choose a very large ratio of filter width to grid size. The data are filtered with $\bar{\Delta}/\Delta$ of 256, 128, and 64, where $\bar{\Delta}$ denotes the filter width.

The velocity field at $t = 240$ and the energy spectrum are shown in Fig. 2.1. As the number of Fourier modes is sufficiently high and the viscosity is very low, the inertial range is large, which gives a wide range for choosing the filter size, and there can be seen the sawtooth structures and the narrow shocks of strong negative gradient in the domain. The DNS parameters and the flow characteristics are listed in Table 2.2, where δ, u_{rms}, Re_λ, $\varepsilon = 2\nu\langle S_{ij}S_{ij}\rangle$, and $\eta = (\nu^3/\varepsilon)^{1/4}$ represent the shock thickness of the strongest shock, the root-mean-square of velocity, the Reynolds number based on the Taylor microscale, the rate of energy dissipation, and the Kolmogorov length scale, respectively. δ is calculated as follows

$$\delta = \frac{\Delta u}{(\partial u/\partial x)_{min}}, \tag{2.7}$$

where Δu denotes the velocity difference just before and after the shock with the minimum velocity gradient. λ denotes the Taylor microscale which is defined as [90]

$$\lambda = \sqrt{\frac{5\nu}{\varepsilon}} u_{rms}. \tag{2.8}$$

All parameters are calculated at $t = 240$. The calculated parameters indicate that the grid is fine enough to capture the smallest scales of turbulence.

Three types of spatial filter functions are used for filtering the DNS data. Table 2.3 shows the filter kernels in spectral space, $\hat{G}(k)$. H represents the Heaviside function, and $k_c = \pi/\Delta$ denotes the cut-off wavenumber in Table 2.3. Filtering is performed in the spectral space which is given by

$$\hat{\bar{u}}(k,t) = \hat{G}(k)\hat{u}(k,t), \tag{2.9}$$

where $\hat{u}(k,t)$ represents the velocity in the spectral space. As seen in Fig. 2.2, the top-hat filter is not effective at high wavenumbers. The cut-off filter is sharp in the spectral space and nonlocal in the physical space. Hence, the Gaussian filter is the best choice; however, the correlation coefficient between the true SGS stress and two SGS models which is defined as

$$C.C. = \frac{\langle (\tau_{11}^{SGS} - \langle \tau_{11}^{SGS} \rangle)(\tau_{11}^{DNS} - \langle \tau_{11}^{DNS} \rangle) \rangle}{[\langle (\tau_{11}^{SGS} - \langle \tau_{11}^{SGS} \rangle)^2 \rangle]^{\frac{1}{2}} [\langle (\tau_{11}^{DNS} - \langle \tau_{11}^{DNS} \rangle)^2 \rangle]^{\frac{1}{2}}}, \tag{2.10}$$

is calculated to see the effects of the filter function. The results presented in Table 2.4 show that the highest correlation coefficient for the SGS stress between the filtered DNS and SGS models is achieved when the Gaussian filter is used. Therefore, the DNS data used for training the neural network are filtered using the Gaussian filter function. In Table 2.4, $C.C._{SM}$ and $C.C._{GM}$ show the correlation coefficients for the Smagorinsky model (SM) and the gradient model (GM), respectively.

Table 2.2. DNS parameters of Burgers equation at $t = 240$.

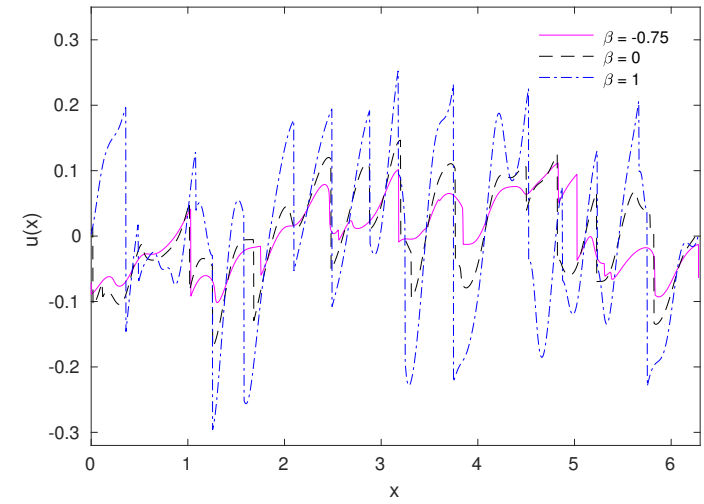
Case	λ	$\delta(\times 10^{-4})$	λ/Δ	u_{rms}	Re_λ	ε	$\eta(\times 10^{-3})$	η/Δ
DNS1	0.02415	4.78	251.894	0.0528	159.327	9.55×10^{-5}	1.52	15.854
DNS2	0.01731	4.73	180.550	0.0690	119.470	3.98×10^{-4}	1.26	13.423
DNS3	0.01388	3.55	144.774	0.1214	84.259	3.82×10^{-3}	1.20	12.516

Table 2.3. Filter functions in spectral space.

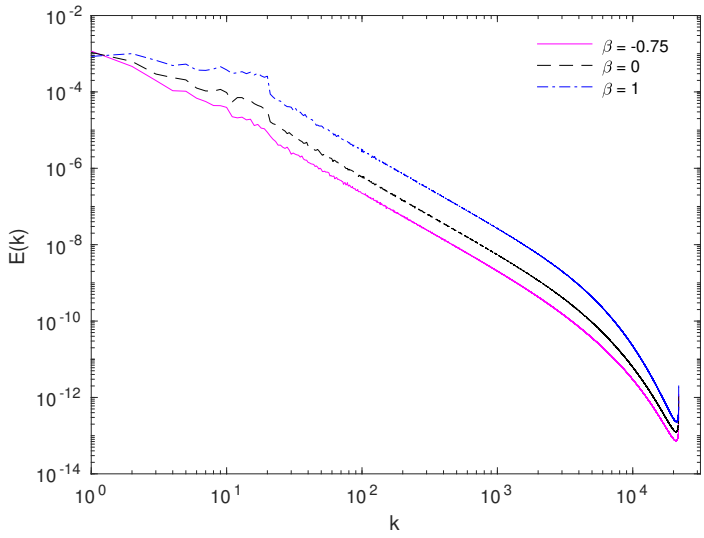
Filter type	Top-hat	Gaussian	Cut-off
$\hat{G}(k)$	$\frac{sin(\frac{1}{2}k\Delta)}{\frac{1}{2}k\Delta}$	$exp(-\frac{k^2\Delta^2}{24})$	$H(k_c - \lvert k\rvert)$

Table 2.4. Correlation coefficient for different filter functions.

$\bar{\Delta}/\Delta$	Filter type	$C.C._{SM}$	$C.C._{GM}$
256	Top-hat	0.91254	0.91329
	Gaussian	0.98449	0.98403
	Cut-off	0.81871	0.81315
128	Top-hat	0.92001	0.92015
	Gaussian	0.98595	0.98578
	Cut-off	0.81651	0.80603
64	Top-hat	0.92999	0.93003
	Gaussian	0.98714	0.98710
	Cut-off	0.79376	0.78322



(a)



(b)

Figure 2.1. (a) Velocity field and (b) energy spectrum for the DNS.

2.3 SGS modeling using neural network

In this section., the SGS modeling using fully-connected neural network is explained. Initially, the information on the fully-connected neural netwrok is provided. Then, the input of output choices of the neural network are explained.

2.3.1 Feed-forward neural network

In this research, feed-forward neural network (FFNN) with one hidden layer is used as a tool to find a proper SGS model by constructing a functional relationship between the grid-scale flow field and the SGS stress. Filtered DNS data at all grid points are used for providing the input

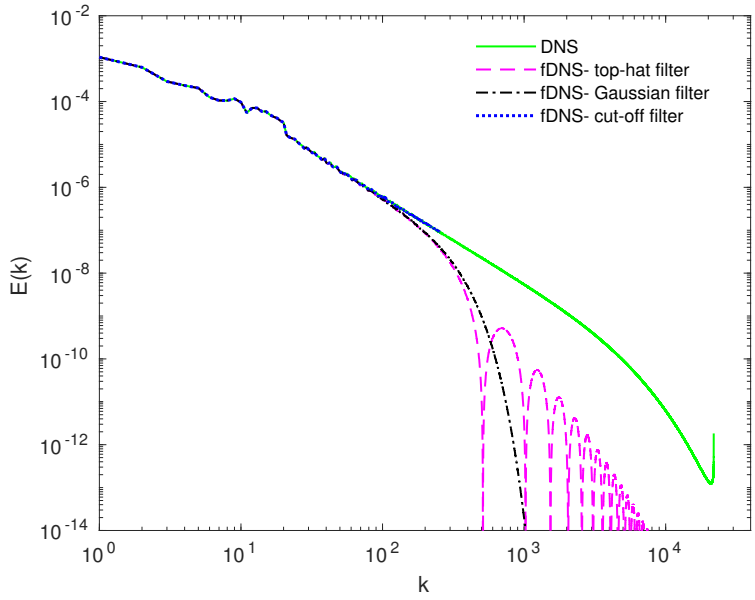


Figure 2.2. DNS data filtered using three types of filter functions.

and the output of the neural network. Three DNSs are performed for the Burgers equation with different viscosities and forcing functions. The DNS data are filtered using three different filter sizes ($\bar{\Delta}/\Delta = 256, 128, 64$) to make the model applicable to a wide range of filter sizes. 70%, 15% and 15% of the DNS data are dedicated to training, validation and testing, respectively. The activation function is hyperbolic tangent, and Levenberg-Marquardt (LM) algorithm is used as the training algorithm. The initial weights are chosen randomly and the initial biases are zero. A schematic diagram of a fully connected feed-forward neural network and a neuron are shown in Fig. 2.3. The following equation is used to calculate the output of each neuron in the l-th layer

$$x_j^l = f(b^l + \sum_{i=1}^{n} x_i^{l-1} w_{ij}^l), \tag{2.11}$$

where f is the activation function, b is the bias parameter, n is the number of neurons in the previous layer, x_i^{l-1} is the output of the i-th neuron in the previous layer, and w is the weight.

2.3.2 Choice of input and output features

Five sets of input variables are considered as shown in Table 2.5. The Galilean invariance [4] is often important in turbulence modeling. In the present case, the dimension of the space is one, which implies that the Galilean invariance is equivalent to the invariance under translation. Thus, the velocity gradient and thereby the input variables listed in Table 2.5 are Galilean invariant.

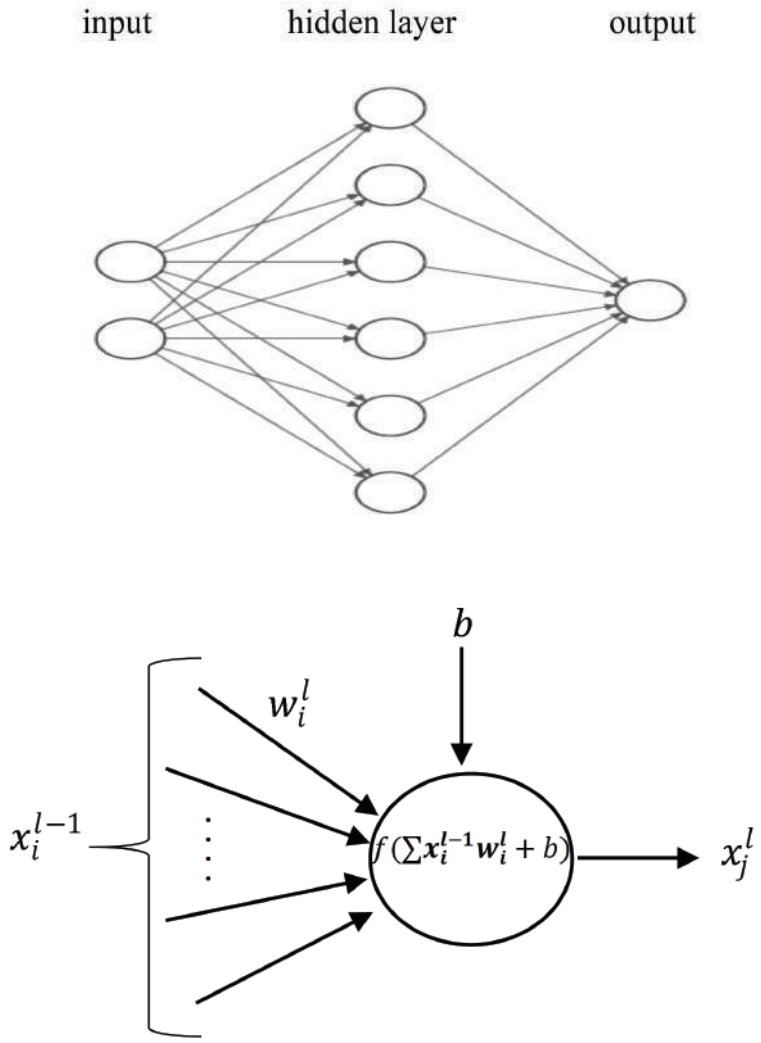


Figure 2.3. Schematic diagram of a feed-forward neural network (left) and a neuron (right).

Since most existing SGS models include the velocity gradient or the strain rate, which are identical for the Burgers equation, this component is considered in all sets of input. In NN1, the velocity gradient is considered as the input, and the output (τ) is normalized by $\bar{\Delta}^2$. In the Wong SGS model [91], the exponent of $4/3$ is used for the filter size. Therefore, for NN2, the exponents of the filter size for the input and the output are chosen in a way to imitate the Wong SGS model while having a compact range of data. Due to the importance of the filter size in LES calculations [92], in NN3 this parameter is considered as an additional input in order to make the effect of the filter size more pronounced. Zhou et al. [49] also used this set of input for SGS modeling of isotropic turbulent flow by neural network. They considered 300 neurons in the hidden layer and used the ReLU activation function. They showed that the obtained model gives a high correlation coefficient and predicts the energy transfer accurately. For NN4 and NN5, the velocity gradient multiplied by its absolute value is used as an input to see whether this model which has an input similar to the Smagorinsky model can improve the accuracy of the SGS model and outperform the existing SGS models. The results of the neural network models are compared with the existing SGS models, including the gradient model

$$\tau_{ij} = \frac{\bar{\Delta}^2}{12} \frac{\partial \bar{u}_i}{\partial x_k} \frac{\partial \bar{u}_j}{\partial x_k}, \tag{2.12}$$

the Smagorinsky model

$$\tau_{ij} - \frac{\delta_{ij}}{3} \tau_{kk} = -2C_s^2 \bar{\Delta}^2 |\bar{S}| \bar{S}_{ij}, \tag{2.13}$$

and the dynamic Smagorinsky model. In Eq. (2.13), $\bar{S} = \frac{1}{2}(\frac{\bar{u}_i}{x_j} + \frac{\bar{u}_j}{x_i})$ is the strain rate and $|\bar{S}| = (2\bar{S}_{ij}\bar{S}_{ij})^{\frac{1}{2}}$ is the magnitude of strain rate tensor. However, for the Burgers equation, since it is one-dimensional, $\bar{S} = \frac{\partial \bar{u}}{\partial x}$. In the dynamic Smagorinsky model, the coefficient C_s is not constant and it is calculated as follows,

$$C(z,t) = -\frac{1}{2} \frac{\langle L_{ij} M_{ij} \rangle}{\langle M_{ij} M_{ij} \rangle}, \tag{2.14}$$

where $C_s = \sqrt{C}$. For the dynamic Smagorinsky model (DSM), backscatter clipping is used, and

Table 2.5. Neural network input(s) and output choices.

Case	Input(s)	Output
NN1	$d\bar{u}/dx$	$\tau/\bar{\Delta}^2$
NN2	$\bar{\Delta}^{\frac{1}{3}}d\bar{u}/dx$	$\tau/\bar{\Delta}$
NN3	$\bar{\Delta}d\bar{u}/dx$, $\bar{\Delta}$	τ
NN4	$\bar{\Delta}^2 d\bar{u}/dx\|d\bar{u}/dx\|$	τ
NN5	$\bar{\Delta}d\bar{u}/dx\|d\bar{u}/dx\|$	$\tau/\bar{\Delta}$

if the C is negative, it will be set to zero. In Eq. (2.14), $\langle . \rangle$ for the Burgers equation denotes averaging over the domain, and

$$L_{ij} = \widehat{\bar{u}_i \bar{u}_j} - \hat{\bar{u}}_i \hat{\bar{u}}_j \tag{2.15}$$

$$M_{ij} = \hat{\Delta}^2 |\hat{\bar{S}}| \hat{\bar{S}}_{ij} - \bar{\Delta}^2 \widehat{|\bar{S}| \bar{S}_{ij}} \tag{2.16}$$

where $\hat{\Delta} = 2\bar{\Delta}$ is the test filter.

Based on the results of L_2 error for the SGS stress in the *a priori* test, energy spectrum, and L_2 error for the velocity in the *a posteriori* test, it is found that the coefficient of $C_s = 0.2$ gives rise to the most accurate results for the Smagorinsky model in this study, which was also chosen by Li and Wang [72]. However, in the study carried out by Maulik and San [73], it is shown that the best performance for the energy spectrum is obtained when $C_s = 0.3$. The reason for the difference in choosing the best C_s can be due to the absence of a forcing term for the Burgers equation in their paper.

2.4 Numerical results and discussions

In the present section, first, the results of the *a priori* test are presented and discussed. Then, the ability of the neural network will be compared with the random forest in establishing an SGS model and the impact of choosing other inputs for the neural network will be discussed. Finally, the models will be embedded in an actual LES and the generalizability of the neural network models will be checked extensively, their performance will be compared with the existing models.

2.4.1 *A priori* test

It can be extremely time-consuming if the neural network architecture is complicated. Employing a large number of neurons in the hidden layer can make the obtained SGS model computationally inefficient since it needs to be embedded in the computational fluid dynamics (CFD) code and calculated in each iteration. Therefore, it is necessary to have an optimized number of neurons and simple structure for the neural network. The effect of increasing the number of neurons in the hidden layer on the correlation coefficient between the true and predicted SGS stress for the training and test datasets is shown in Fig. 2.4. The values of the correlation coefficient are averaged over ten times of running the neural network. The correlation coefficients for the training and test datasets are nearly similar which shows that there is no severe overfitting for these cases. As it is observed, a hidden layer with six neurons is sufficient to attain a high correlation coefficient for all sets of the input(s) and output in Table 2.5.

After determining the optimum number of neurons and obtaining the neural network models, the correlation coefficient and the L_2 norm of error are calculated for different SGS models. As seen in Table 2.6, for all neural network models, the correlation coefficient is high and comparable to that of the gradient model and the Smagorinsky model. However, among the neural network models, NN1, NN2, and NN5 produce higher correlation coefficients than NN3 and NN4 in most cases. Therefore, it can be deduced that scaling of the input and the output data can give rise to more accurate results, at least in the *a priori* test. Li and Wang [72] carried out an *a priori* test for the high Reynolds number decaying Burgers equation at initial instances, when the solution is still smooth. Their results show that the change in the LES grid resolution does not affect the correlation coefficient for different SGS models. In contrast to the present results, the Smagorinsky model is poorly correlated with the filtered DNS data in their study. Similar to the correlation coefficient, Table 2.7 shows that the L_2 norm of error for the neural network models is low and comparable to the existing SGS models. The error decreases for NN1, NN4, NN5, and the existing models as the filter size becomes smaller; however, no definite trend is observed for NN2 and NN3. Fig. 2.5 shows the comparison between the SGS stress obtained using the SGS models and the filtered DNS data for DNS2. As shown, the gradient model and the Smagorinsky model underpredict the SGS stress while other models roughly overpredict the SGS stress. According to the results, the behavior of the obtained neural network models is similar to the dynamic Smagorinsky model, but they are more successful in predicting the SGS

Table 2.6. Correlation coefficient between the SGS models and the SGS stress obtained using filtered DNS data for the Burgers equation with different forcing terms.

Case	$\bar{\Delta}/\Delta$	GM	SM	NN1	NN2	NN3	NN4	NN5
DNS1	256	0.98261	0.98275	0.96982	0.98371	0.95953	0.97140	0.98218
	128	0.98489	0.98495	0.98338	0.98677	0.98435	0.97391	0.98737
	64	0.98644	0.98646	0.98922	0.98847	0.93956	0.97612	0.98893
DNS2	256	0.98403	0.98449	0.98554	0.99198	0.98135	0.97775	0.98529
	128	0.98578	0.98595	0.98539	0.99283	0.98522	0.97924	0.98962
	64	0.98710	0.98714	0.99038	0.99280	0.97466	0.98058	0.99102
DNS3	256	0.98308	0.98331	0.98270	0.98627	0.98296	0.98547	0.98578
	128	0.98513	0.98526	0.98761	0.98622	0.98110	0.98695	0.98751
	64	0.98630	0.98634	0.98821	0.98858	0.98131	0.98758	0.98751

Table 2.7. L_2 norm of error for the SGS models for the Burgers equation with different forcing functions.

Case	$\bar{\Delta}/\Delta$	GM	SM	DSM	NN1	NN2	NN3	NN4	NN5
DNS1	256	0.03312	0.02044	0.02753	0.03312	0.01812	0.02805	0.02431	0.02404
	128	0.02176	0.01262	0.02083	0.01367	0.01599	0.01423	0.02176	0.01350
	64	0.01331	0.00709	0.01580	0.00628	0.01611	0.03934	0.02014	0.00770
DNS2	256	0.07410	0.04497	0.05635	0.03887	0.02421	0.04421	0.04329	0.04170
	128	0.04853	0.02765	0.04620	0.02819	0.02496	0.03317	0.03478	0.02180
	64	0.02943	0.01538	0.03560	0.01281	0.02980	0.04621	0.02960	0.01311
DNS3	256	0.35526	0.21947	0.25677	0.16285	0.20133	0.16052	0.13354	0.13306
	128	0.23572	0.13709	0.21465	0.08632	0.10766	0.10632	0.08850	0.08734
	64	0.14683	0.07936	0.16553	0.05661	0.05706	0.07622	0.06468	0.05803

stress. Among the neural network models, the overall performance of NN5 is better.

It is crucial to further evaluate the models by performing the *a posteriori* test to see if the neural network models provide accurate results when embedded in an actual LES and if the stability issue occurs. But, before proceeding to the *a posteriori* test, two more studies have been performed to show the reasonable accuracy of the chosen machine learning method and the input feature.

2.4.2 Random forests

In addition to feed-forward neural network, random forests [93] can be proposed as a method to derive a data-driven SGS model. In this subsection, the random forests are used to construct an SGS model and compare the results with those obtained by FFNN. The random forest is comprised of a number of decision trees, which is shown in Fig. 2.6. Each node in the decision tree is divided into several branches. The outputs of each node are classified into different categories, called branches, by defining a threshold. In order to find an accurate model using random forests, the number of the decision trees (N_t) and the depth of them should be determined. The important hyperparameters in determining the depth of decision trees are the minimum number of samples per leaf (SL_{min}), minimum number of samples per node (SN_{min}), and the maximum number of decision splits (DS_{max}). Based on a parametric study on different hyperparameters for the training datasets in Table 2.1, the optimized value for each parameter is determined as shown in Table 2.8. The input and output of the random forest model are the same as those of NN1. In order to check the accuracy of the obtained model using random forests, the L_2 norm of error is calculated for the DNS2 dataset, which is used in the training process. Furthermore, to investigate the generalizability of the model, it is tested for the unseen data in Table 2.18 (Case1 and Case2), and the results are compared with the neural network SGS model.

As the results in Table 2.9 show, the random forest model has a good performance when used for the data which are included in the training process (DNS2). However, when the model is tested for the unseen data (Case1 and Case2), it is not very accurate compared to the neural network model.To further see the differences between the performance of the neural network and random forests model, the predicted residual stress by these models are shown in Fig. 2.7. From this figure, it also can be understood that the predictions of both models are roughly similar for DNS2, which its data are implemented in the training process. However, for Case1, the performance of the neural network model is much better, which can clearly confirm the better generalizability of the neural network model compared to the random forest model. It also should be noted that the considered structure for the random forests model is more complex than the neural network model. Therefore, it can be deduced that the neural network can be a better choice for obtaining an SGS model since its generalizability is much better than the random forests, even though a complicated structure is chosen for the random forests model. Wang [48] et al. also compared the random forests algorithm with the neural network method for SGS

Table 2.8. Chosen hyperparameters for random forests model.

Case	N_t	SL_{min}	SN_{min}	DS_{max}
RF	150	2	5	75

Table 2.9. Comparison of L_2 error for neural network model and random forests model.

Case	$\bar{\Delta}/\Delta$	NN1	RF
DNS2	128	0.0282	0.0237
Case1	128	0.0253	0.0319
Case2	64	0.1194	0.1516

modeling of homogeneous isotropic turbulence, and showed that the random forests model is not as accurate as the neural network model for obtaining an SGS model.

2.4.3 Comparison to simple inputs

In this section, the effect of choosing a simple type of input is investigated. It is seen that the velocity gradient and the velocity have widely been used as the input for SGS modeling using neural network [47–49, 51, 53, 55]. In the study performed by Subel et al. [61] on SGS modeling of Burgers equation using neural network, the velocity, which is not Galilean invariant, is used as the input and the training data are normalized by subtracting the mean and dividing by the standard deviation. Such kind of scaling can affect the generalizability of the model and make it inappropriate for applying to the unseen data due to requiring a priori knowledge of the mean value and the standard deviation of the input and the output. In all of the referenced papers, a roughly complex structure is chosen for the neural network to establish an SGS model, which can make the model computationally extremely expensive, even much more than the dynamic Smagorinsky model [48]. In order to investigate the performance of the neural network SGS models using simple inputs, like velocity and velocity gradients, two neural networks with the same structure as that of the previous section are trained by considering the velocity and the velocity gradient as the input, and the residual stress as the output, using the training datasets in Table 2.1. Table 2.10 shows the correlation coefficient and the error obtained by testing the models on the DNS2 dataset which is included in the training process. The results of Table 2.10 clearly show that the neural network SGS models constructed using the velocity and velocity

Table 2.10. Correlation coefficient (C.C.) and L_2 error for the neural network models with simple choices of input.

Input	$\bar{\Delta}/\Delta$	C.C.	L_2
$\bar{u}$	256	0.0109	0.1641
	128	0.0065	0.1165
	64	0.0039	0.0823
$d\bar{u}/dx$	256	0.9526	0.0537
	128	0.9695	0.0974
	64	0.9421	0.1869

gradient are not accurate compared to the neural network models already discussed. Especially, when the velocity chosen as the input, the correlation coefficient is extremely low, and the model is not even able to predict the trend of the residual stress. On the other hand, as shown in Fig. 2.8, when the velocity gradient is selected as the input, the neural network model overpredicts the large values of the residual stress, significantly, which leads to a huge error in predicting the residual stress. Thus, a neural network model with a simple structure obtained using either velocity or velocity gradient does not have a good performance. Choosing a complex structure for the neural network makes the model computationally inappropriate and applying similar normalization to Subel et al. [61] can make the model less likely to be generalized. Hence, trying to make improvements using such approaches does not look to be proper in practice.

2.4.4 *A posteriori* test

In the *a posteriori* test, the simulation is conducted for DNS2 to see the performance of the SGS models in an actual LES. Three ratios of $\bar{\Delta}/\Delta_{LES}$ are considered to see the effect of mesh resolution by keeping the filter width constant. The LES parameters are shown in Table 2.11; the time step is the same as that of DNS. The correlation coefficient and the L_2 error are calculated at $t = 240$. According to Table 2.12, for LES1, the L_2 error of NN1, NN4, and NN5 is comparable to that of dynamic Smagorinsky model and lower than the other models. For the largest value of $\bar{\Delta}/\Delta_{LES}$, the neural network SGS models except NN1 and NN2 have a better accuracy than the existing models. NN2 is much more sensitive to $\bar{\Delta}/\Delta_{LES}$ than NN1, and its error significantly increases for LES3. Therefore, a neural network model with only velocity gradient as an input, may not be accurate enough for large $\bar{\Delta}/\Delta_{LES}$. As it is seen, NN4 and NN5 are in better agreement with the filtered DNS results and have a reasonable accuracy for all

Table 2.11. LES parameters of Burgers equation.

Case	N_{LES}	$\bar{\Delta}/\Delta_{LES}$	k_{max}
LES1	1024	2	341
LES2	2048	4	682
LES3	4096	8	1365

Table 2.12. L_2 norm of error for velocity field after implementing the SGS models in the LES code.

SGS model	LES1	LES2	LES3
NN1	0.31630	0.24170	0.19045
NN2	0.44121	0.22219	0.93448
NN3	0.41913	0.22601	0.16292
NN4	0.32163	0.22787	0.16055
NN5	0.32235	0.22846	0.16313
GM	2.16753	0.41024	-
SM	0.34191	0.23646	0.16975
DSM	0.32105	0.22752	0.16366

three ratios of $\bar{\Delta}/\Delta_{LES}$. The error for these two models is lower than the gradient model and the Smagorinsky model and is comparable to the dynamic Smagorinsky model. NN3 is less accurate for a low number of modes; however, for larger $\bar{\Delta}/\Delta_{LES}$, its accuracy improves, and the error for this model is similar to that of NN4 and NN5.

Unlike the results shown in the *a priori* test, according to the L_2 error in Table 2.12, scaling does not necessarily improve the results and the choice of inputs look to be more important. Furthermore, it shows that the results of the *a priori* test cannot guarantee the performance of the model. In contrast to the study performed by Li and Wang [72] for the decaying Burgers equation, increasing the grid resolution for the LES generally improves the accuracy of the solution. In their study, the correlation coefficient obtained for the LES using different SGS models was roughly independent of $\bar{\Delta}/\Delta_{LES}$. Maulik and San [73] used different approaches for LES of decaying Burgers with $\nu = 5 \times 10^{-4}$, and in contrast to the current study, they showed that the Smagorinsky model with the coefficient $C_s = 0.2$, is not successful in preventing energy pile-up. They indicated that the dynamic version of this model can make improvement over this model as observed in the current results, however, its performance depends on the choice of test filter. In their paper, the grid size and the filter width are equal, $\bar{\Delta} = \Delta_{LES}$.

Table 2.13. C.C. for production term after implementing the SGS models in the LES code.

SGS model	LES1	LES2	LES3
NN1	0.27297	0.62522	0.86759
NN2	0.20610	0.68822	0.61606
NN3	0.24941	0.67960	0.90449
NN4	0.25856	0.67856	0.90815
NN5	0.24256	0.66457	0.90082
GM	0.01033	0.44385	-
SM	0.23946	0.61399	0.89155
DSM	0.22545	0.67725	0.90282

The correlation coefficient is calculated for the production term, which indicates the energy transfer from the resolved-scale flow field to the SGS component and is defined as

$$P = -\tau_{11}\bar{S}_{11}, \tag{2.17}$$

for the Burgers equation, in order to evaluate the performance and the accuracy of the models. Since the production term is the combination of the SGS stress and the filtered strain rate, it is important in determining the accuracy of an SGS model. The results shown in Table 2.13 and Table 2.14 indicate that NN3, NN4, and NN5 are more successful than the other neural network SGS models in predicting the production term. Although NN1 has an overall good prediction, it is not as successful as NN3, NN4, and NN5 for the cases of LES2 and LES3. The correlation coefficient and the norm of error for the production term further confirms that NN2 is not in good agreement with the filtered DNS for the high ratio of $\bar{\Delta}/\Delta_{LES}$. On the whole, the accuracy of NN3, NN4, and NN5 are comparable to the (dynamic) Smagorinsky model in predicting the production term. All neural network models outperform the gradient model in predicting the production term. The gradient model does not have a good accuracy in predicting the production term, especially when the grid is coarse, and the correlation coefficient for this model is extremely low for LES1.

Table 2.15 shows the production ratio, which is calculated by averaging the ratio of the production term obtained using LES to that of filtered DNS from $t = 120$ to 240, to see how accurate the model is in predicting the production term during the simulation. The results of Table 2.15 indicate that the production term obtained using LES is in reasonable agreement with that of filtered DNS. For LES1, the production ratio is slightly underpredicted, and it is

Table 2.14. L_2 error for production term after implementing the SGS models in the LES code.

SGS model	LES1	LES2	LES3
NN1	0.23036	0.24494	0.20537
NN2	0.24421	0.22264	0.33531
NN3	0.23727	0.22525	0.17430
NN4	0.23713	0.22841	0.17221
NN5	0.23610	0.23336	0.17926
GM	0.42716	0.32860	-
SM	0.23054	0.26064	0.19337
DSM	0.23949	0.22539	0.17567

Table 2.15. Average production ratio from $t = 120$ to 240 after implementing the SGS models in the LES code.

SGS model	LES1	LES2	LES3
NN1	0.99287	1.00496	1.00583
NN2	0.97348	1.00640	0.98565
NN3	0.98154	1.00581	1.00563
NN4	0.98291	1.00233	1.00268
NN5	0.98356	1.00283	1.00327
GM	0.64869	0.98285	-
SM	0.95888	0.99423	0.99444
DSM	0.99336	1.00557	1.00616

approximately equal to one for the other two cases. As previously shown, NN2 is less accurate compared to the other SGS models in predicting the production term for the largest value of $\bar{\Delta}/\Delta_{LES}$.

Fig. 2.9 shows the energy spectrum obtained for the LES using different SGS models. For LES1, the gradient model is not able to predict the energy spectrum well, and the other neural network models outperform this model. For LES2, all neural network models have better performance than the gradient model and the Smagorinsky model, and are comparable to the dynamic Smagorinsky model. In this case, the performance of the gradient model improves in predicting the inertial range. When $\bar{\Delta}/\Delta_{LES}$ increases up to eight, it is observed that NN2 is not as successful as before in predicting the energy spectrum. In fact, the energy at the high wavenumbers does not drop sufficiently, and the model does not have an accurate prediction of the dissipation range as well. Therefore, this model is not accurate enough to be used for high ratios of $\bar{\Delta}/\Delta_{LES}$ although the other neural network models are still able to have a slightly better prediction of the energy spectrum than the (dynamic) Smagorinsky model.

Fig. 2.10 shows the probability density function (PDF) of the velocity gradient for the LES performed using different SGS models and the filtered DNS data. For the coarsest grid size, the PDF predicted by all SGS models is wider than that of filtered DNS. The SGS models are not very accurate especially in predicting the PDF of positive velocity gradients. The gradient model does not give a good prediction for the negative values either. The PDF predicted by this model is much wider than the other SGS models. Among the other SGS models, the dynamics Smagorinsky model, NN1, and NN5 are slightly closer to the filtered DNS results. The dynamic Smagorinsky model has a better accuracy in the positive velocity gradient region, but it is slightly less accurate than NN1 and NN5 in predicting the peak of PDF. For LES2, the PDF for all SGS models becomes much closer to that of the filtered DNS. Although the performance of the gradient model is improved substantially, the other SGS models are still in better agreement with the filtered DNS. In the positive region, the dynamic Smagorinsky model and the neural network models except NN4 are more accurate, but the dynamic Smagorinsky model and NN2 overpredicts and underpredicts the peak of PDF, respectively. Therefore, NN1, NN3 and NN5 show better performance on the whole. For LES3, the PDF of the velocity gradient for NN2 becomes much wider, which shows that this model is not accurate for the high ratios of $\bar{\Delta}/\Delta_{LES}$. The performance of the other SGS models is improved in this case, and they are in reasonable

Table 2.16. Root-mean-square-error (RMSE) for PDF. The bin width (velocity gradient difference between two edges of each bin) of the histogram used for calculating the PDF is 1.5.

SGS model	LES1	LES2	LES3
NN1	0.0113	0.00050	0.00013
NN2	0.0118	0.00023	0.00380
NN3	0.0102	0.00150	0.00005
NN4	0.0109	0.00210	0.00015
NN5	0.0113	0.00048	0.00013
GM	0.0153	0.00200	-
SM	0.0063	0.00068	0.00022
DSM	0.0089	0.00120	0.00010

agreement with the filtered DNS data, while NN5 has a better accuracy compared to the other models. The results also show that scaling can improve the model performance since NN5 outperforms NN4 in all cases. Table 2.16 shows the root-mean-square-error (RMSE) for the PDF of velocity gradient. The error in this table is calculated by obtaining the PDF using the histogram data, where the velocity gradient difference between two edges for each bin is 1.5. The bin width is chosen in a way to have a sufficient number of bins for all cases ($30 \sim 40$). However, it should be pointed out that the results might be slightly affected by the number of bins. As the results of Table 2.16 indicate, for LES1, the error for all SGS models is much higher than the other two cases, and the largest error is for the gradient model. However, as $\bar{\Delta}/\Delta_{LES}$ increases, the error decreases significantly. For LES2, based on the results of Fig. 2.10 and Table 2.16, NN1 and NN5 have a better performance than the other models. According to Table 2.16, the error for LES3 is extremely low for all SGS models except NN2, which can also be confirmed by Fig. 2.10.

Table 2.17 shows the average computational cost of the SGS models per iteration which corresponds to the CPU time calculated for calling only the function related to the calculation of the SGS stress term. The averaged values are obtained by taking the average of the CPU time over 100 times steps. As seen, due to the simple structures for the neural network models, they are not costly and their computational cost is much lower than the DSM. The number of operation counts divided by the number of grid points is shown in Table 2.17 as well which is roughly consistent with the CPU time of the models. For the neural network models except the activation function which is hyperbolic tangent, all other operations are simple like the multiplication and addition. However, for DSM, the number of FFTs performed in each operation is much larger than the other models which makes the DSM computationally more expensive than the other

Table 2.17. Computational time and operation counts for different models.

SGS model	CPU time (s)	×,/	All excluding (+,-,×,/,FFT)	All excluding FFT	FFT
NN1	4.232×10^{-4}	16	6	41	1
NN2	4.057×10^{-4}	17	7	43	1
NN3	4.645×10^{-4}	21	6	52	1
NN4	3.948×10^{-4}	17	6	42	1
NN5	3.997×10^{-4}	17	6	42	1
GM	1.284×10^{-4}	6	0	6	1
SM	1.372×10^{-4}	10	1	11	1
DSM	1.059×10^{-3}	20	0	25	8

models.

2.4.5 Testing the neural network SGS models for forced Burgers equation with different conditions

The obtained SGS models are tested for two more cases as described in Table 2.18, in order to realize how well the neural network SGS models can be generalized to different conditions. The viscosity for Case1 and Case2 are chosen lower and higher than that of DNSs used for training, respectively. Furthermore as shown in Table 2.18, the range of wavenumber of the forcing function is different from the previous DNSs. According to the results of the previous section, for $\bar{\Delta}/\Delta_{LES} = 4$, the SGS models give roughly accurate results; therefore, this ratio is chosen for conducting the LES in this section.

Table 2.19 shows the comparison of L_2 error between different SGS models for Case1 and Case2. The results show that in both cases, the L_2 error for NN2, NN4 and NN5 is comparable to that of the dynamic Smagorinsky model and is lower than the Smagorinsky model. The gradient model does not converge for Case1 which shows that the grid is not sufficiently fine to use this SGS model. Another point is that for these two cases, the L_2 norm of error for NN1 and NN2 is lower than NN3. This shows that although NN3 was as accurate as NN1 and NN2 in the simulation which its data were used for training the neural network, it is not as successful as before in the new simulations. It demonstrates that scaling the data can improve the model's generalization ability.

Fig. 2.11 shows the energy spectrum for the LES using different SGS models. The energy spectrum for Case1 shows that NN1 and NN3 are more dissipative than the other neural network

Table 2.18. Parameters of the Burgers equations used for testing the SGS models.

Case	N_{DNS}	N_{LES}	ν	β	k_f	$\bar{\Delta}/\Delta_{LES}$
Case1	32768	1024	1×10^{-6}	-0.75	30	4
Case2	8192	512	5×10^{-4}	1	10	4

Table 2.19. L_2 norm of error for the velocity field after implementing the SGS models in an LES code for Case1 and Case2.

SGS model	NN1	NN2	NN3	NN4	NN5	GM	SM	DSM
Case1	0.16036	0.13950	0.17337	0.14273	0.13845	-	0.14542	0.13761
Case2	0.40822	0.31840	0.47273	0.31938	0.32518	0.31988	0.32688	0.32488

models. NN2, NN4, and NN5 show reasonable agreement with the filtered DNS, and their performance is better than the dynamic Smagorinsky model, but NN2 is slightly more dissipative. Comparison of the energy spectrum for Case2 shows that NN3 behaves differently from the other SGS models. It is not able to predict the energy spectrum correctly, and its accuracy is lower than the gradient model and the Smagorinsky model. In this case, NN2, NN4, and NN5 are able to predict the energy spectrum better than the other SGS models, and they are in reasonable agreement with the filtered DNS.

The PDF of the velocity gradient is shown in Fig. 2.12. For Case1, it is seen that in the negative region, the difference between NN1 and NN3, and the filtered DNS is larger compared to the other SGS models. This can confirm that these models are not accurate enough in capturing shocks. On the other hand, their accuracy is better than the other SGS models in predicting the PDF of positive velocity gradient. NN5 is in good agreement with the filtered DNS. The discrepancy between NN4 and the filtered DNS is higher in the positive region than the other SGS models. The PDF of the velocity gradient for Case2 shows that NN3 does not give proper prediction of the velocity gradient. In this case, the other SGS models are in reasonable agreement with the filtered DNS. The error for PDF of velocity gradient is shown in Table 2.20, which indicates that the smallest error among the neural network SGS models for Case1 belongs to NN5. For Case2, the largest error is for NN3 which can also be confirmed by the PDF plot in Fig. 2.12. After NN3, the NN1 SGS model has the largest error compared to the other models. Therefore, the results of PDF error are also in reasonable agreement with the error for the velocity field which shows that for Case2, NN1 and NN3 are less accurate than the other SGS models.

Table 2.20. Root-mean-square-error (RMSE) for PDF. The bin width (velocity gradient difference between two edges of each bin) of the histogram used for calculating the PDF is 0.5.

SGS model	Case1	Case2
NN1	0.0052	0.0021
NN2	0.0023	0.0005
NN3	0.0051	0.0069
NN4	0.0076	0.0004
NN5	0.0017	0.0005
GM	-	0.0005
SM	0.0009	0.0002
DSM	0.0010	0.0004

2.4.6 LES of forced Burgers equation with very low viscosity and high $\bar{\Delta}/\Delta_{LES}$

Before testing the model for the decaying Burgers equation, the performance of the neural network models is further investigated by applying them to LES of the forced Burgers equation with very low viscosity and high ratio of $\bar{\Delta}/\Delta_{LES}$. The detailed information of the simulation is presented in Table 2.21. As seen in the table, for this case, all parameters in the forcing function are different from those which their data are used for training. The L_2 error for different SGS models is shown in Table 2.22. As seen, NN1 and NN3 produce a larger error compared to the other SGS models. The smallest error among the neural network models belongs to NN5, which is comparable to the dynamic Smagorinsky model. To further investigate the difference between the performance of different models, and also to figure out where the error in the velocity profile is concentrated, the velocity profile for the DNS and LES, and their difference are shown in Fig. 2.13. The results of this figure clearly show the lower accuracy of NN1 and NN3 in comparison to the other models. According to this figure, on the whole, all models are able to predict the velocity profile, and the largest error for the velocity profiles occurs at the shocks. Especially, for NN3, the velocity difference at the shocks is much higher than the other models. Among the neural network models, NN2, NN4, and NN5 are shown to produce better results for a very low viscosity case while NN5 is still slightly better than NN2 and NN4. Fig. 2.14 shows the energy spectrum for the LES using different SGS models, which demonstrates that the energy for NN1 decays faster than the other models at high wavenumbers. Furthermore, NN3 is not able to have an accurate prediction for the energy spectrum. The mention points are in reasonable agreement with the velocity profile results which show that NN1 and NN3 are less accurate than the other models.

Table 2.21. Parameters of the forced Burgers equation with very low viscosity.

Case	N_{DNS}	N_{LES}	ν	β	k_f	$\bar{\Delta}/\Delta_{LES}$
F1	65536	1024	2.5×10^{-7}	0.1	6	8

Table 2.22. L_2 norm of error for the velocity field after implementing the F1 SGS model in an LES code.

SGS model	NN1	NN2	NN3	NN4	NN5	SM	DSM
F1	0.08834	0.06844	0.16555	0.07016	0.06805	0.07442	0.06735

2.4.7 The possibility of using the neural network SGS models for decaying Burgers equation

In this section, the obtained SGS models are tested for the LES of decaying Burgers equation to further investigate the performance of the SGS models. The initial velocity profile is chosen randomly as shown in Fig. 2.15. The results of Table 2.24 show that NN1, NN4, and NN5 have a reasonable accuracy, better than the other neural network models and the Smagorinsky model. Similar to the previous cases, NN2 is not accurate when the ratio of $\bar{\Delta}/\Delta_{LES}$ is high. It is also seen that NN3 does not have a good prediction for any of these cases which can confirm that this model does not have the generalization ability. Fig. 2.16 shows the energy spectrum for the decaying Burgers equation for two different ratios of filter size to grid size. The results of this figure show that NN3 is not able to have a good prediction of energy spectrum for the decaying Burgers equation. For $\bar{\Delta}/\Delta_{LES} = 8$, NN2 cannot predict the energy spectrum well. In the lower wavenumber range, there is some discrepancy between NN2 and filtered DNS, and the difference even becomes much more significant at high wavenumbers. NN1, NN4, and NN5 have a better performance than the Smagorinsky model in predicting energy spectrum for both cases of D1 and D2. Considering the results of Table 2.24, it can be concluded that on the whole, NN4 and NN5 are better than the other SGS models.

Table 2.23. Parameters of decaying Burgers equation.

Case	N_{DNS}	N_{LES}	ν	$\bar{\Delta}/\Delta_{LES}$
D1	8192	512	10^{-4}	4
D2	8192	1024	10^{-4}	8

Table 2.24. L_2 norm of error for the velocity field after implementing the SGS models in an LES code for decaying Burgers equation at $t = 10$.

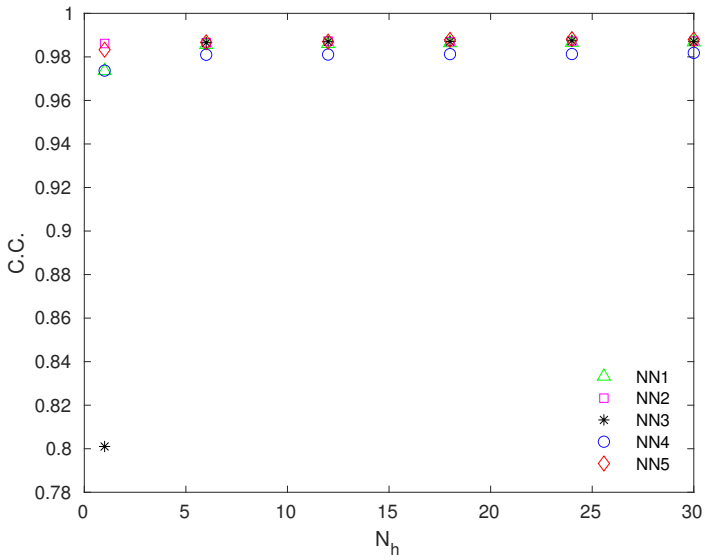
Case	NN1	NN2	NN3	NN4	NN5	SM
D1	0.15793	0.16596	0.27974	0.15449	0.15456	0.16242
D2	0.12462	0.66491	0.30442	0.11896	0.12010	0.12729

2.5 Conclusions

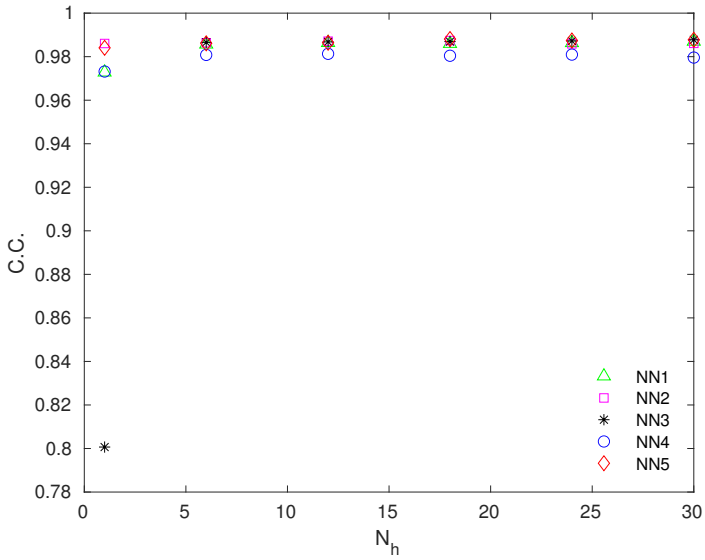
In the present study, SGS modeling for the Burgers equation using single hidden layer neural network has been investigated. Five sets of input(s) were provided using the combination of the filter width and the filtered velocity gradient. For three cases, the output data were scaled using the filter width. The *a priori* test shows that the obtained neural network models produce a high correlation coefficient and are in reasonable agreement with the filtered DNS. The results also show that the performance of the neural network models is comparable to the existing models and is even better when scaling is performed for the input(s) and output datasets. In the *a posteriori* test, it has been shown that all obtained models are stable when implemented in an LES code without using any stabilization technique. The results indicate that good performance in the *a priori* test does not guarantee a model to be accurate in the *a posteriori* test. The neural network models produce accurate results, better than the gradient model and the Smagorinsky model, and comparable to the dynamic Smagorinsky model when embedded in an LES code. By considering the forced Burgers equation with different conditions from training and decaying Burgers equation, it is investigated if a model trained using the DNS data of the Burgers equation with specific forcing functions can be used in other simulations with different parameters. The results show that the obtained models except NN1 and NN3 have a good performance for the LES of Burgers equation under different conditions. NN2 is shown not to be accurate for the large ratio of filter width to grid size for the higher viscosity cases. Additionally, it is indicated that in general, scaling can improve the model's accuracy, especially in the cases of which data are not considered for training the neural network. On the whole, using $\bar{\Delta} du/dx|du/dx|$ as the input and $\tau/\bar{\Delta}$ as the output of the neural network leads to a more accurate model compared to the other neural network SGS models. The results obtained using this model are in a better agreement with the filtered DNS. The accuracy of this model is better than the dynamic Smagorinsky model in some cases.

It is worth mentioning that performing LES using the neural network SGS models takes much less time than the dynamic Smagorinsky model, while comparable to the gradient model and the Smagorinsky model, since the neural network structure is not complicated.

It should be noted that achieving accurate results for SGS modeling of the Burgers equation can make the chosen inputs a good candidate for developing SGS models for three-dimensional turbulence. Although they look to be more complex than the velocity gradient, which has been frequently used as the input feature for the data-driven SGS models in previous works, they can be computationally more efficient by making it possible to use a more simple structure for the neural network.



(a)



(b)

Figure 2.4. C.C. for different number of neurons used in the hidden layer. (a) training and (b) test datasets.

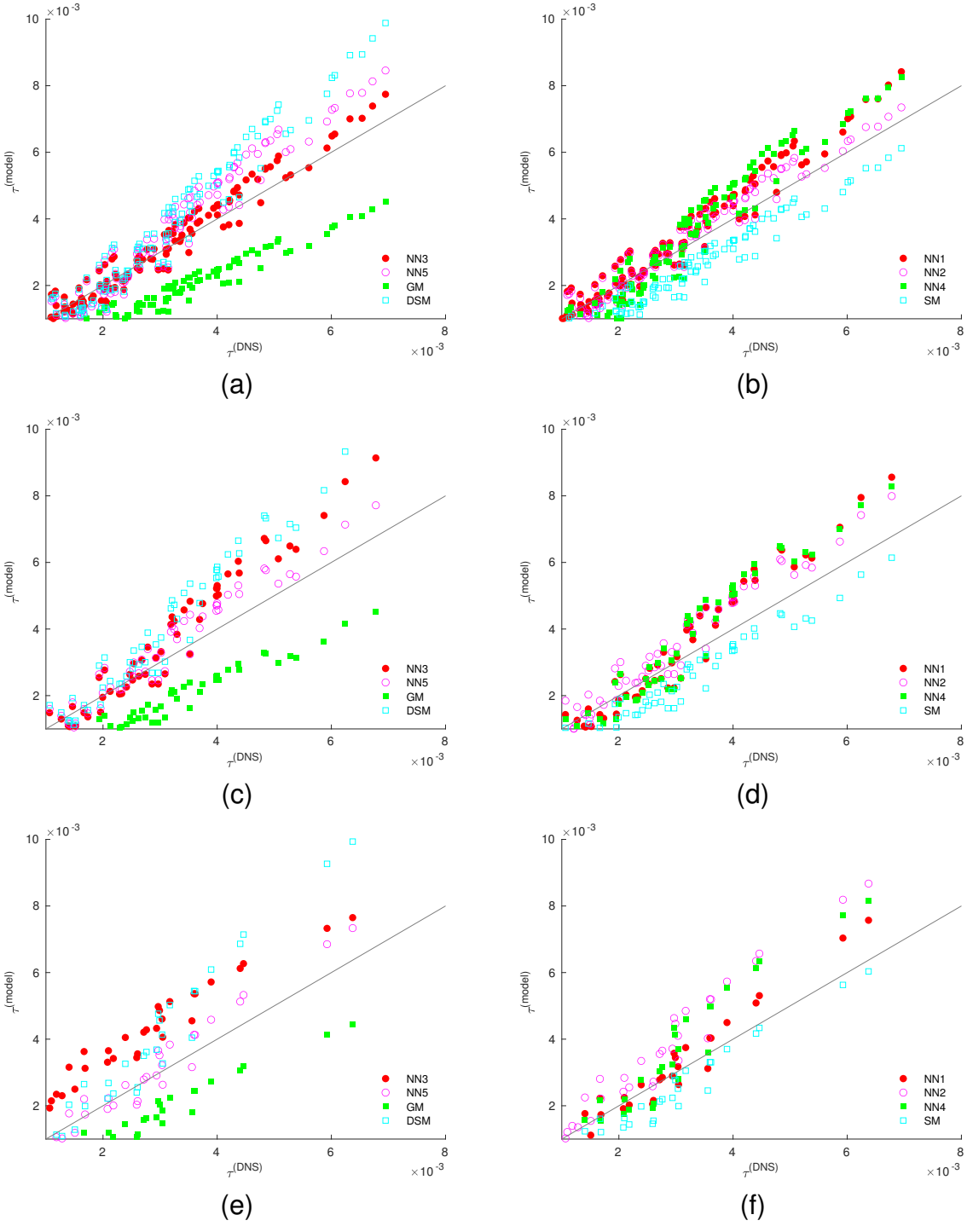


Figure 2.5. Comparison between SGS models and true SGS stress for $\bar{\Delta}/\Delta$. (a,c,e) NN3, NN5, gradient model (GM), and dynamic Smagorinsky model (DSM), (b,d,f) NN1, NN2, NN4, and Smagorinsky model (SM). (a,b) 256, (c,d) 128, and (e,f) 64.

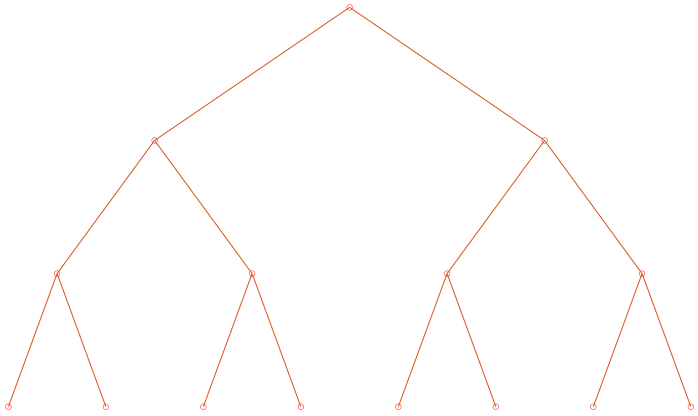


Figure 2.6. Schematic of a decision tree.

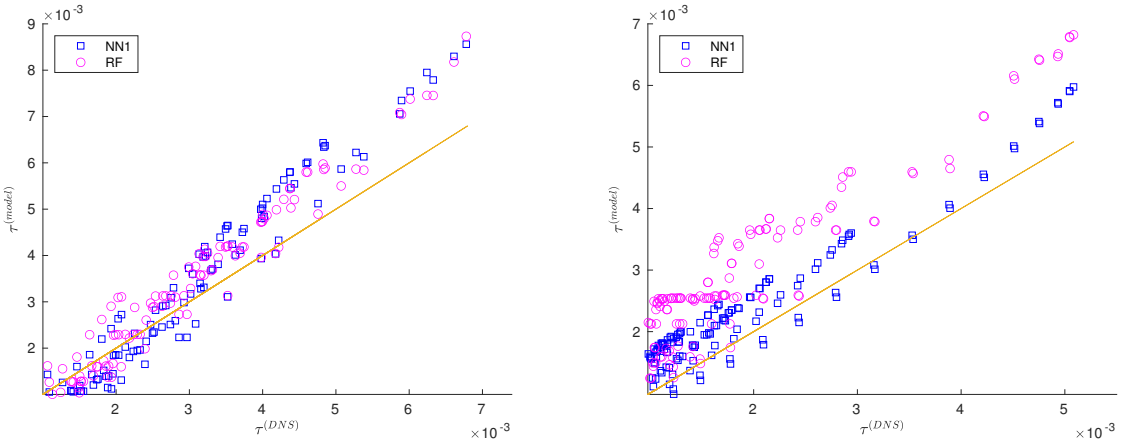


Figure 2.7. Comparison between the SGS stress predicted by NN1 and RF for (a) DNS2 and (b) Case1 datasets.

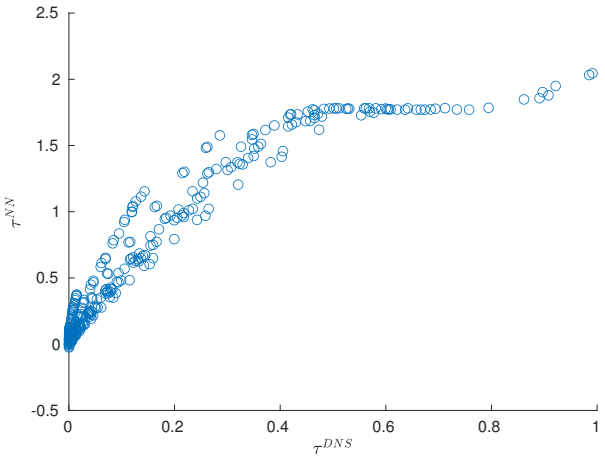


Figure 2.8. Comparison of true and predicted residual stress when velocity gradient is used as the neural network input. Both axes are normalized by the maximum absolute value of τ^{DNS}.

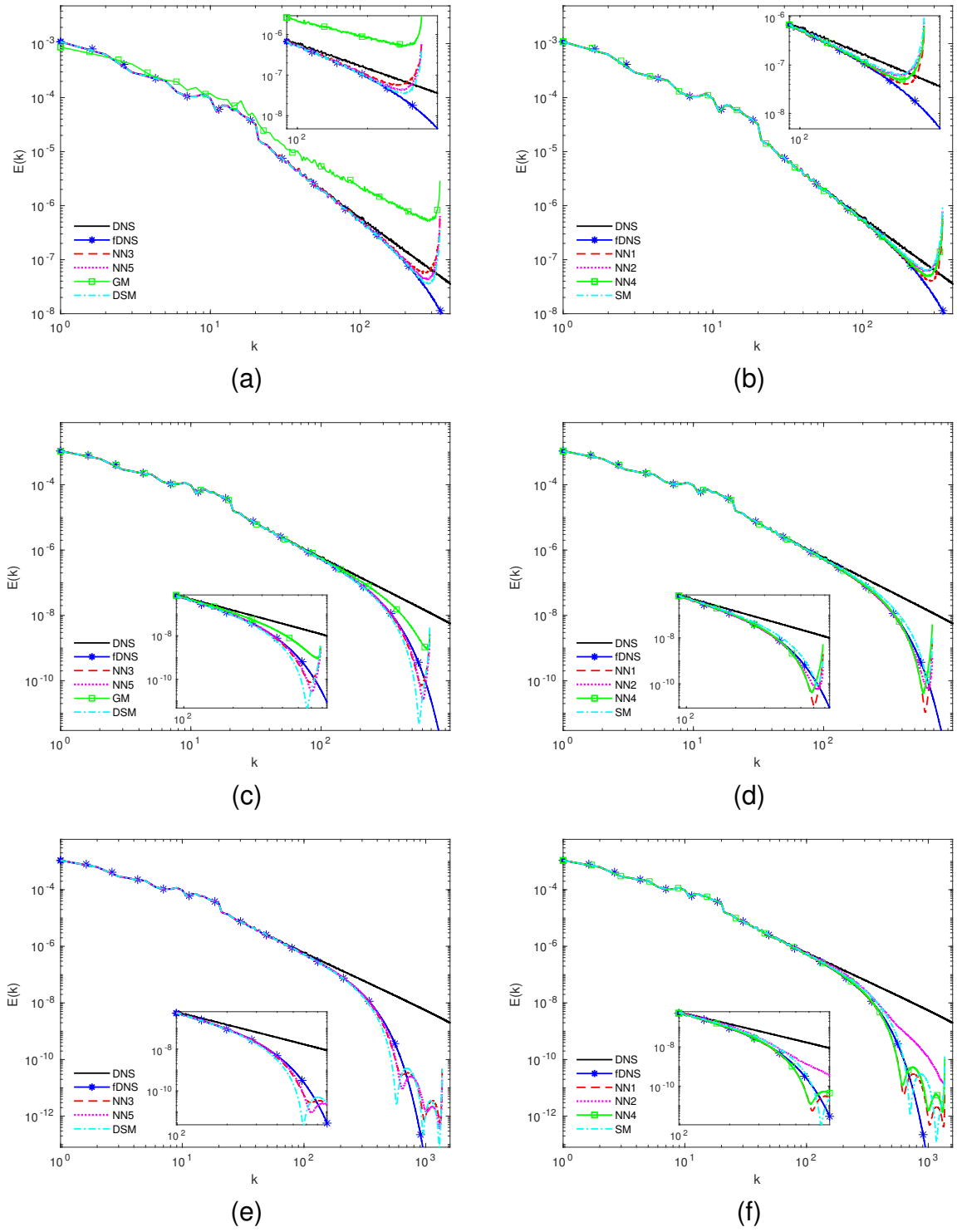


Figure 2.9. Energy spectrum of Burgers equation. Comparison between (a,c,e) filtered DNS (fDNS), NN3, NN5, gradient model (GM), and dynamic Smagorinsky model (DSM) and (b,d,f) filtered DNS, NN1, NN2, NN4, and Smagorinsky model (SM). (a,b) LES1, (c,d) LES2, and (e,f) LES3.

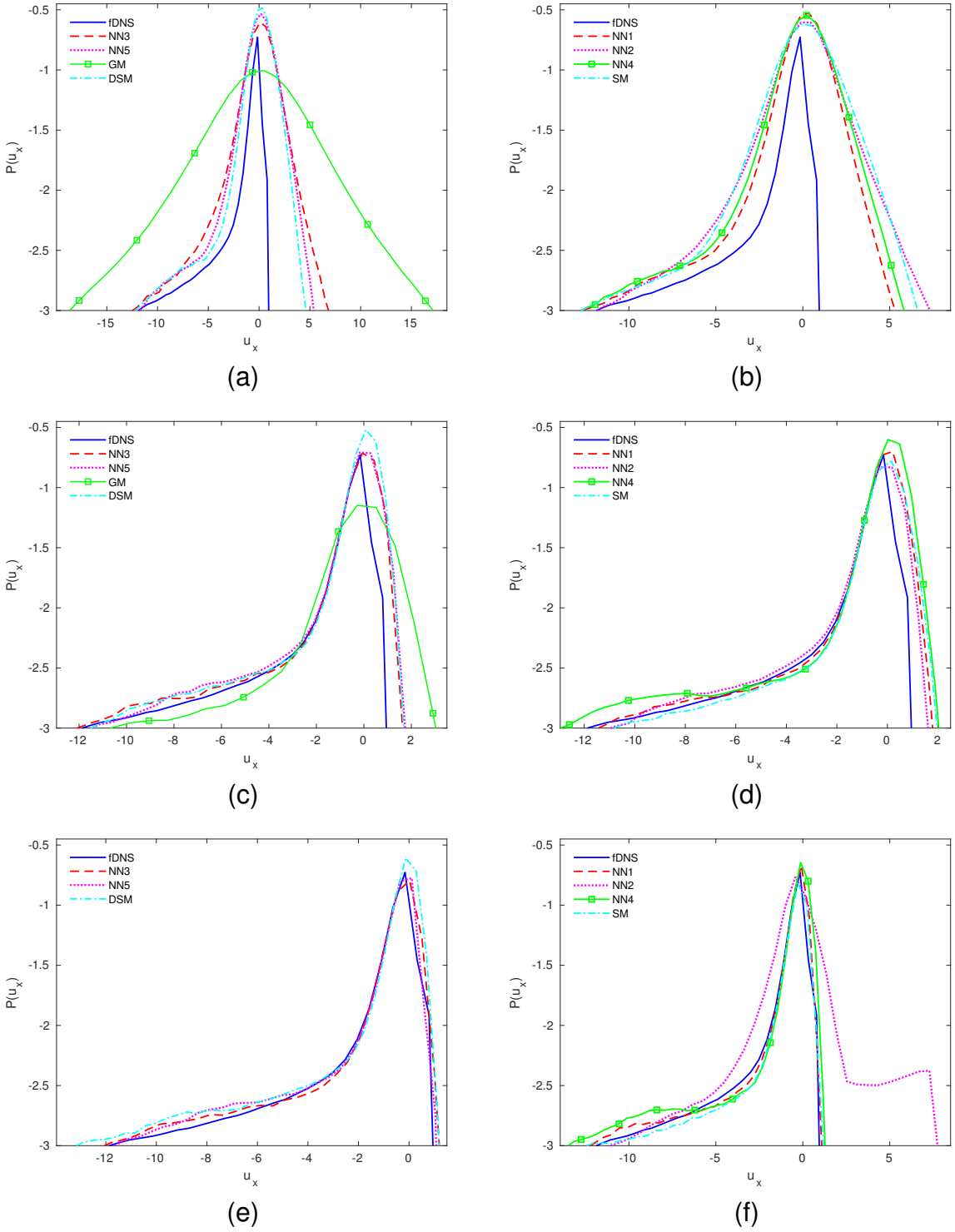


Figure 2.10. PDF of velocity gradient of Burgers equation; the vertical axis is in logarithmic scale. (a,b) LES1, (c,d) LES2, and (e,f) LES3.

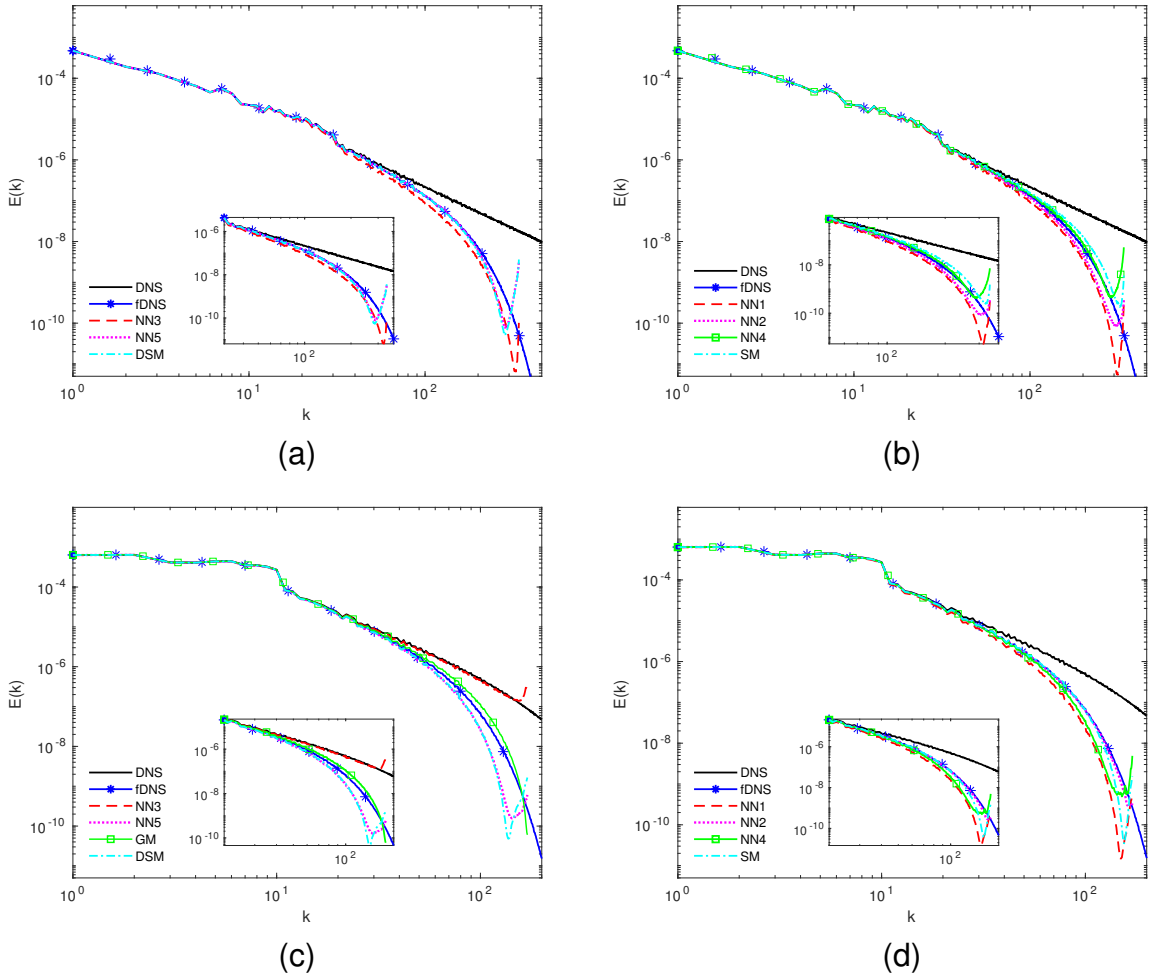


Figure 2.11. Energy spectrum for Case1 (a,b) and Case2 (c,d).

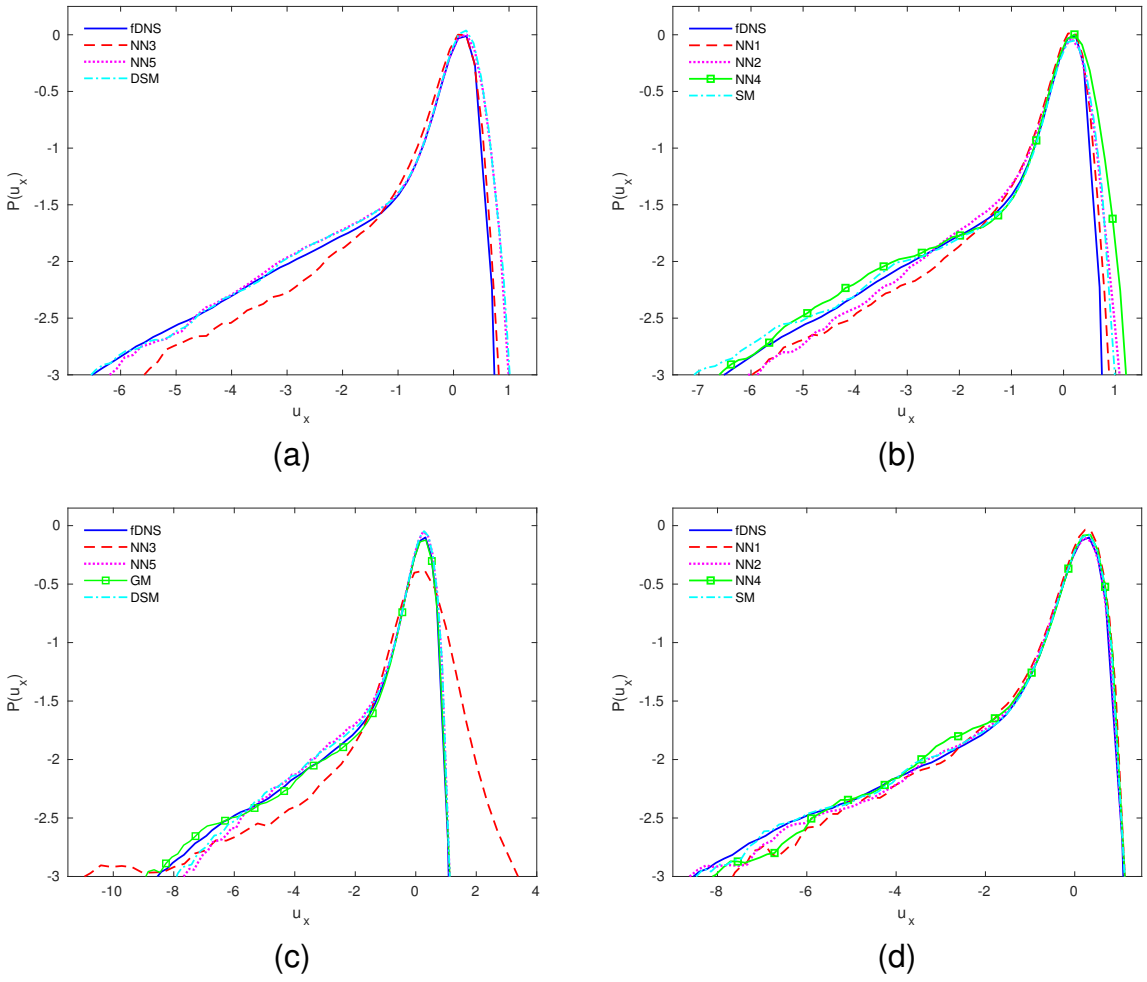


Figure 2.12. PDF of velocity gradient; the vertical axis is in logarithmic scale. (a,b) Case1 and (c,d) Case2.

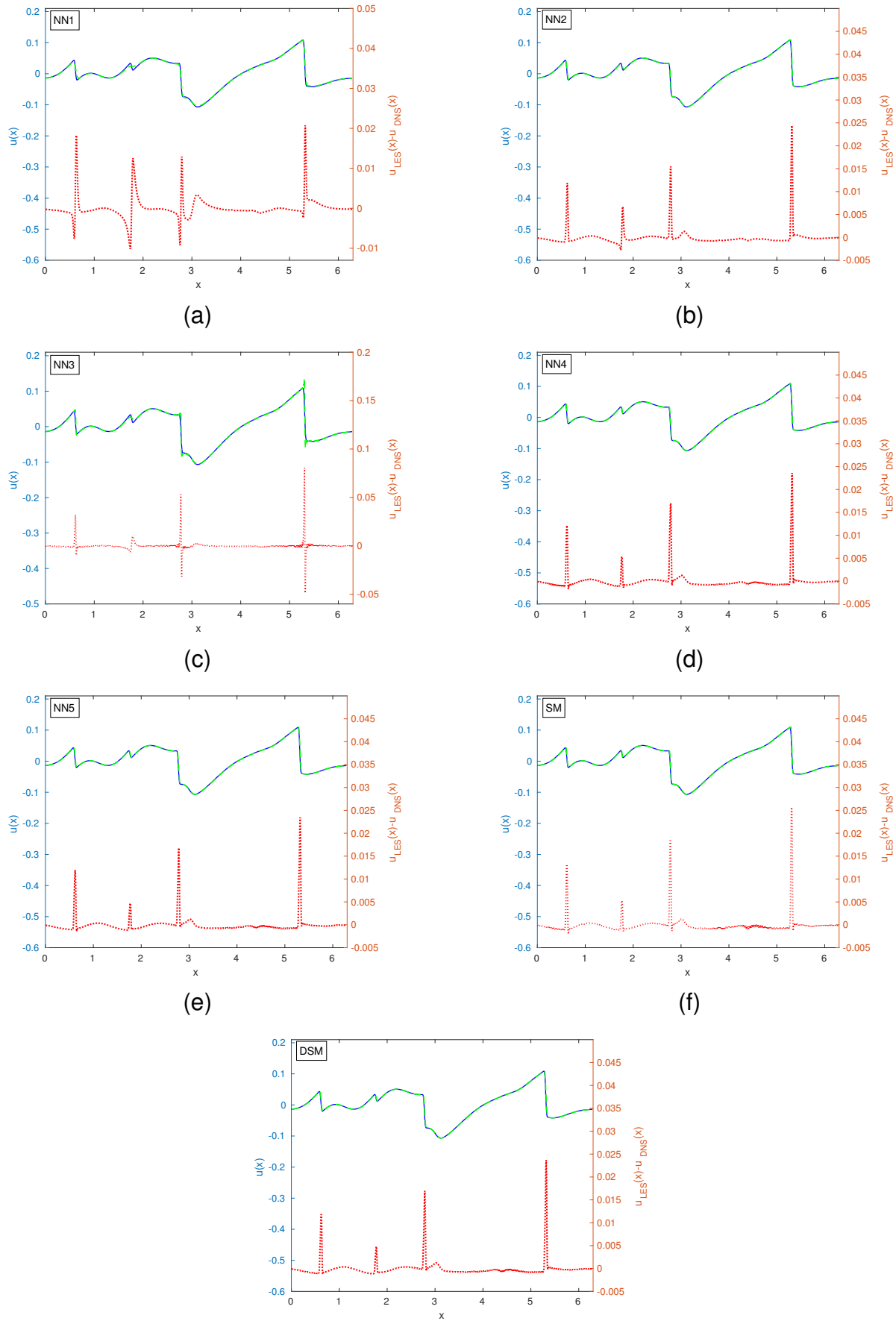


Figure 2.13. Velocity profile for the DNS and LES, and velocity difference between DNS and LES for F1. The scale of velocity difference axis for NN3 is different from other models.

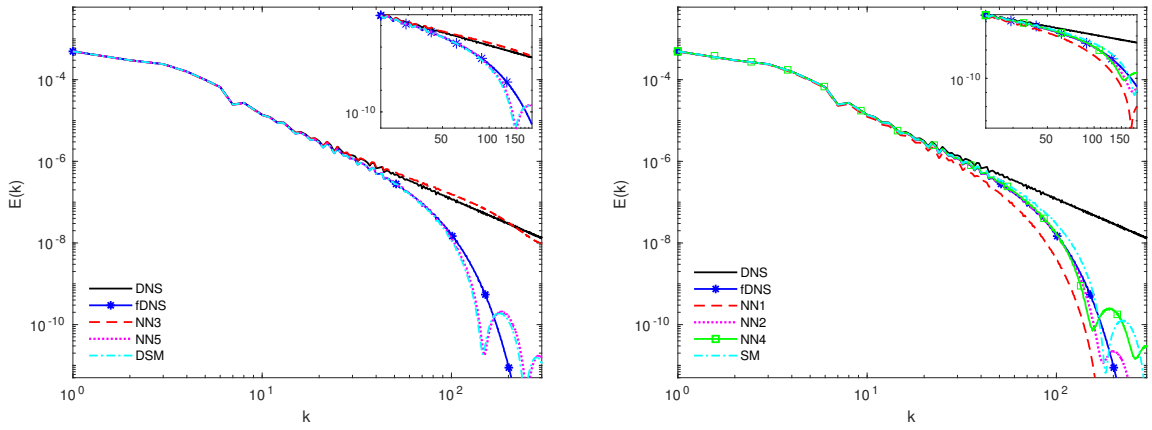


Figure 2.14. Energy spectrum for different SGS models for F1.

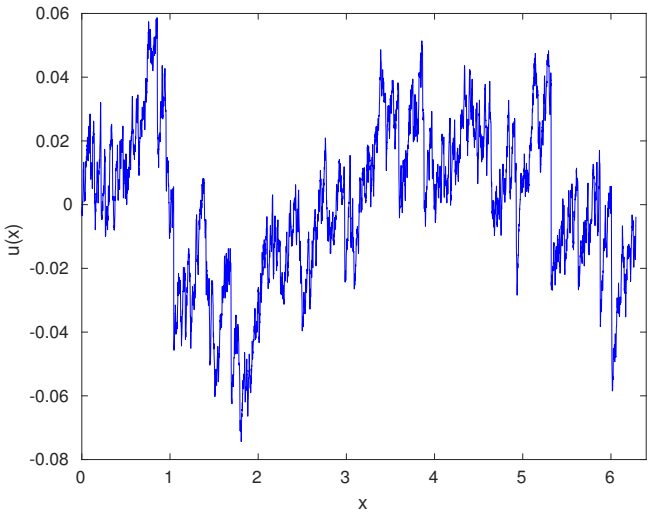


Figure 2.15. Velocity profile for decaying Burgers equation at $t = 0$.

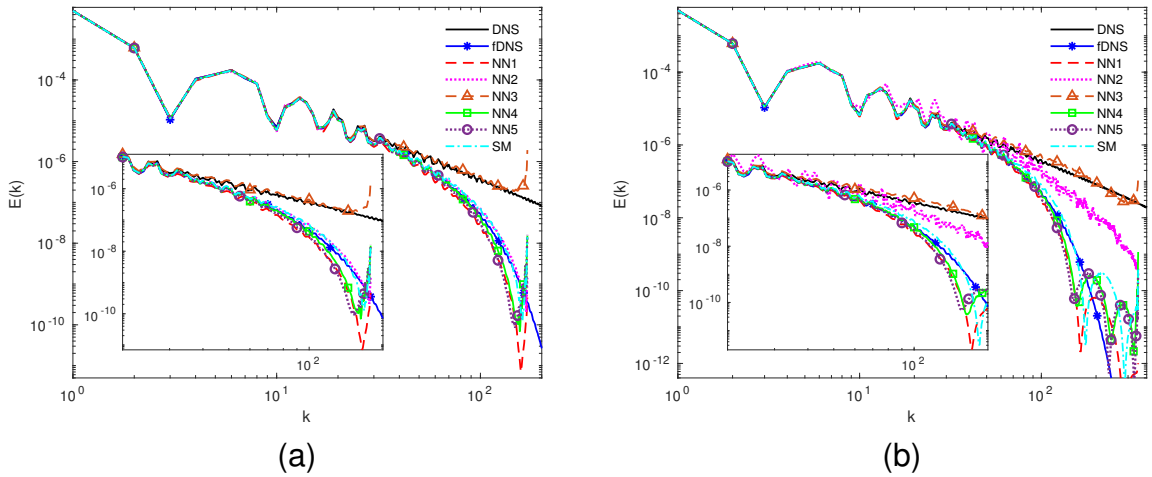


Figure 2.16. Energy spectrum of decaying Burgers equation, averaged from $t = 9$ to 10. (a) $\bar{\Delta}/\Delta_{LES} = 4$ and (b) $\bar{\Delta}/\Delta_{LES} = 8$.

3. Data-driven wall modeling, *a priori* test

3.1 Introduction

The wall modeling in LES is of cardinal importance as the computational cost of LES rises significantly for high Reynolds number wall-bounded flows which can make the LES impossible for such flows. Therefore, it is important to find an accurate wall model with reasonable accuracy and computational cost. There are mainly two different approaches for the wall-modeled LES. The first one is the hybrid RANS/LES method which solves the RANS equations in the near-wall region and uses LES for the outer layer. This approach may not be computationally efficient compared to the second one which is called wall-stress model since a fine grid is still required near the wall because of using RANS approach. The wall stress models, on the other hand make it possible to use LES for the whole domain. However, a fine grid is not required near the wall as by using these models the region near the wall will be modeled. Such models are generally applied as the approximate boundary condition at the wall. The approximate boundary condition can affect the accuracy of the simulation; therefore, it is important to find an accurate model for the wall shear stress which will be applied as the wall boundary condition. One of the common problems which can occur in the wall modeled LES is called log-layer mismatch which means that the wall shear stress deviates from the true wall shear stress by 15 percent, approximately. In the previous studies various treatments have been proposed to improve this problem such as applying temporal or spatial filtering, adding a stochastic forcing, and moving the matching location which will be the position of the LES data provided as the wall model input [94]. Additionally, the wall-modeled LES might be sensitive to the SGS model and not have good performance for a simpler SGS model like the static Smagorinsky model [25]. Therefore, sometimes it is required to modify the SGS models or use more complex SGS models to obtain reasonable results.

In the present chapter, a data-driven wall model is developed using convolutional neural network which has not been already done in the previous studies. CNN has a good ability in recognizing the patterns within images and is able to encode the features specific to the image [95]. In fact, as the inputs propagate through the deeper layers, it gains the ability to

extract the abstract features [96]. Therefore, it can be a good tool for finding a wall model. In this chapter, first, a study will be performed on the neural network parameters like the number of convolution filters and their sizes, and the effect of input choices. After obtaining the CNN wall models, their performance will be checked under various conditions in the *a priori* test by applying them to a different grid and domain sizes, and higher Reynolds number flows. Furthermore, the wall shear stress predicted by these models will be compared with that of the existing models, including shifted wall model and ODE-based wall model.

3.2 Wall modeling in LES

In LES, the Navier-Stokes equations are solved for the grid-scale (GS) flow field by applying the spatial filter to the flow field variables

$$\overline{\phi(x,t)} = \int_{-\infty}^{\infty} \phi(r,t) G(x-r)\, dr \tag{3.1}$$

where the overbar and G denote the spatial filtering and the filter kernel, respectively.

The general form of the filtered Navier-Stokes equations for an incompressible flow are

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \tag{3.2}$$

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial (\bar{u}_i\, \bar{u}_j)}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_k \partial x_k} - \frac{\partial \tau_{ij}}{\partial x_j} \tag{3.3}$$

where τ_{ij} is the SGS stress and is defined as

$$\tau_{ij} = \overline{u_i u_j} - \overline{u_i}\,\overline{u_j} \tag{3.4}$$

For WRLES, the inner layer is resolved and the no-slip boundary condition is implemented in the wall-normal direction while for the WMLES, since the near-wall region is modeled, the boundary condition is usually applied as the Neumann boundary condition

$$\frac{\partial \bar{u}}{\partial y}\Big|_w = \frac{\tau_w}{(\nu_t + \nu)|_w} \tag{3.5}$$

where τ_w, ν_t and ν denote the wall shear stress provided by the wall model, the eddy viscosity and the kinematic viscosity, respectively. In Eq. (3.5), ν_t can be considered to be zero. When it is not zero, it is usually estimated using the eddy viscosity value at the first point off the wall. For instance in the staggered grid, it can be estimated as follows [97]

$$\nu_t|_w = \nu_t|_{\Delta y/2}. \tag{3.6}$$

Recently, there have also been several studies in which the no-slip boundary condition is implemented at the wall and the wall shear stress is applied using augmented eddy viscosity [32, 37, 98].

$$\nu_t|_w = \frac{\partial \bar{u}}{\partial y}\Big|_w^{-1} \tau_w - \nu \tag{3.7}$$

In the wall-modeled LES, the computational cost decreases significantly compared to the WRLES, especially for the high Reynolds number flows, as there is no need to have a fine grid near the wall. However, since the boundary condition at the wall is provided by the wall-stress model, it is crucial to have a reliable model for the wall shear stress in order to have a reasonably accurate simulation. A schematic of the wall model implementation in LES is shown in Fig. 3.1. As shown in this figure, the data from the LES solution at the distance, h_{wm}, off the wall are fed into the wall model to estimate the wall shear stress which is to be applied as the boundary condition for the WMLES.

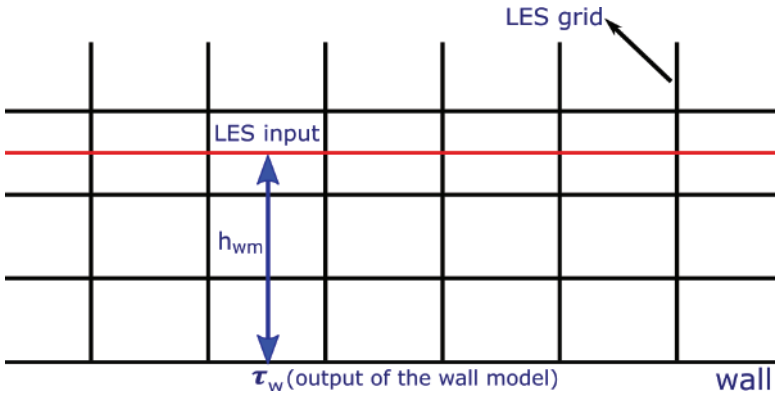


Figure 3.1. Schematic of wall model implementation. The inputs for the wall model are provided by the LES data at the distance h_{wm} off the wall.

In the present study, the CNN wall model is compared with two existing wall-stress models. The models for the wall shear stress are usually derived based on the law of the wall [27–29, 99]

or by solving the momentum equation in the near-wall region based on physical assumptions [22, 99–101]. The wall models used in the present study for comparison are the shifted wall model [28] and the ODE-based wall model [22] which are briefly explained below.

In the shifted model, the streamwise and the spanwise components of the wall shear stress (τ_w) are defined as

$$\tau_{12}(x,0,z) = \frac{<\tau_w>}{<\bar{u}(x,h_{wm},z)>}\bar{u}(x+\Delta_s,h_{wm},z), \tag{3.8}$$

$$\tau_{32}(x,0,z) = \frac{<\tau_w>}{<\bar{u}(x,h_{wm},z)>}\bar{w}(x+\Delta_s,h_{wm},z) \tag{3.9}$$

where $<\tau_w>$ is the average wall shear stress calculated using the logarithmic law of the wall with the plane average velocity, $<\bar{u}>$, at h_{wm} as input. Δ_s denotes a streamwise displacement which is approximately $h_{wm}\cot 8^\circ$ for $30 < h^+_{wm} < 50$ and $h_{wm}\cot 13^\circ$ for $h^+_{wm} \geq 50$.

The ODE-based wall model obtained by the equilibrium assumption is

$$\frac{d}{dy}\left[(\nu+\nu_{t,wm})\frac{dU}{dy}\right] = 0 \tag{3.10}$$

where U is the wall-parallel velocity. Eq. (3.10) can also be solved for the streamwise and spanwise velocity components, separately [100]. The eddy viscosity for the wall model is

$$\nu_{t,wm} = ku_\tau y[1-\exp(-y^+/A^+)]^2 \tag{3.11}$$

where $k = 0.41$ is the von Kármán constant, $u_\tau = \sqrt{\tau_w}$ is the friction velocity, $y^+ = yu_\tau/\nu$ is the wall-normal distance in wall unit, and $A^+ = 17$. Eq. (3.10) can be solved using the one-dimensional finite volume method between the wall with the no-slip boundary condition and the matching location (h_{wm}), where the boundary condition is provided by LES ($u_{h_{wm}} = u_{LES}$). If Eq. (3.10) is solved for the wall-parallel velocity, the wall shear stress can be calculated in the streamwise and spanwise directions, separately, by assuming that the velocities are aligned with the wall shear stress components as follows

$$\tau_{12}|_w = \tau_w\frac{u|_{h_{wm}}}{U|_{h_{wm}}}, \qquad \tau_{32}|_w = \tau_w\frac{w|_{h_{wm}}}{U|_{h_{wm}}} \tag{3.12}$$

where $U_{h_{wm}}$ is the wall-parallel velocity at h_{wm} and τ_w is the wall shear stress calculated from Eq. (3.10). Although in the above wall models the wall shear stress is obtained for both the streamwise and spanwise directions, there are some studies in which the Neumann ($\partial\bar{w}/\partial y = 0$) or no-slip boundary condition is considered for the spanwise velocity [37, 102]. In the present study, the main focus is on obtaining a model for the wall shear stress in the streamwise direction. However, in Sec. 3.4.4, the ability of the CNN in predicting the wall shear stress in both the streamwise and the spanwise directions will be investigated.

3.3 Data-driven wall modeling in LES

The approximate boundary condition is a popular approach for the wall modeling in LES. This method is computationally inexpensive compared to the hybrid LES/RANS approaches, and estimates τ_w using a wall model. The calculated wall shear stress can then be applied through the boundary condition [37, 100] at the wall. With the proven ability of the machine-learning approaches in prediction, they can be a good choice for establishing a wall-stress model. The details of the data-driven wall model developed in the present study are provided in this section.

3.3.1 Convolutional neural network

In the present study, CNN is used for establishing a wall model. Unlike fully connected neural network (FCNN), in CNN, each neuron receives data from a restricted region. The filters slide along the two-dimensional input data and perform a convolution operation. The generated feature map passes through the activation function to produce the output feature map. A schematic of a convolution layer in CNN is shown in Fig. 3.2. The outputs of each layer are calculated based on the inputs of the previous layer as follows

$$x^l_{i,j,k} = f\Big(\sum_{p=-f_h/2}^{f_h/2}\sum_{q=-f_w/2}^{f_w/2}\sum_{n=1}^{f_n} w^l_{p,q,n,k} x^{l-1}_{i+p,j+q,n} + b^l_k\Big) \tag{3.13}$$

where f_h and f_w are the filter height and filter width, respectively, f_n is the number of filters, w represents the filter kernel, b is the bias, and f denotes the activation function. In the present study, f_h and f_w have the same size and in the last layer $f_n = 1$. The activation function used in the present study is the exponential linear unit (ELU) [103], which is defined as

$$f(x) = \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x \leq 0 \end{cases} \tag{3.14}$$

where $\alpha = 1$. The linear activation function is used in the last layer. The reason for choosing the ELU activation function is that not only does it lead to a faster learning for the deep neural networks, but also it is shown to have a good generalization performance for the networks with more than five layers [103]. The Adam optimizer [104] is adopted as the training algorithm and the loss function is the mean squared error (MSE). In the neural network approaches, since the data used for the training process is limited, the overfitting situation may occur as the result of the unavailability of the new data which means that the model learns how to make prediction for the training data well, but it does not have a good performance when applied to the new data. This issue usually occurs when the model has been trained extensively for the available training data. It can affect the generalizability of the model and cause the neural network not to perform well when used for the new data. One method to reduce the risk of overfitting for the neural network approaches is to add a regularization term [105] to the loss function which increases the variability of the data within different stages of the neural network [106]. Therefore, the overall loss function will be

$$loss = \frac{1}{2N}(\sum_{m=1}^{M}(\tau_{w,m}^{(CNN)} - \tau_{w,m}^{(DNS)})^2 + \lambda||w||^2) \tag{3.15}$$

where N is the total number of training samples and M is the total number of responses over all training samples. λ is the regularization factor which determines how much effect the regularization function has on the objective function and is set to 0.00025, and w is the weight vector.

3.3.2 Wall modeling using CNN

In the present study, CNN is used to find a data-driven wall model for the LES of turbulent channel flow. A schematic of channel flow is shown in Fig. 3.3. In contrast to previous studies on the data-driven wall modeling [63, 65], where the wall shear stress is predicted in a pointwise manner using FCNN, CNN is a nonlocal approach which receives the inputs of the wall model

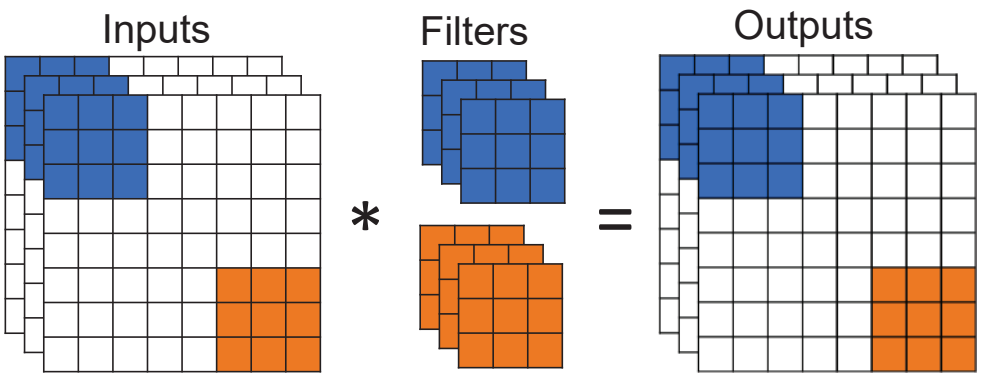


Figure 3.2. A schematic of a convolution layer in CNN.

in the two-dimensional form. As shown in Fig. 3.4, the input data on a wall-parallel plane at a distance, h, off the wall are fed into the CNN to predict the shear stress at the wall. Therefore, for each input-output pair in the training dataset, the input features are constructed by taking the data in 2D shape for each CNN input on an xz plane at a wall-normal distance h, from the wall, and the output feature is the shear stress on the wall plane. In the present study, all the data used in the training process are normalized by the wall unit (u_τ, ν).

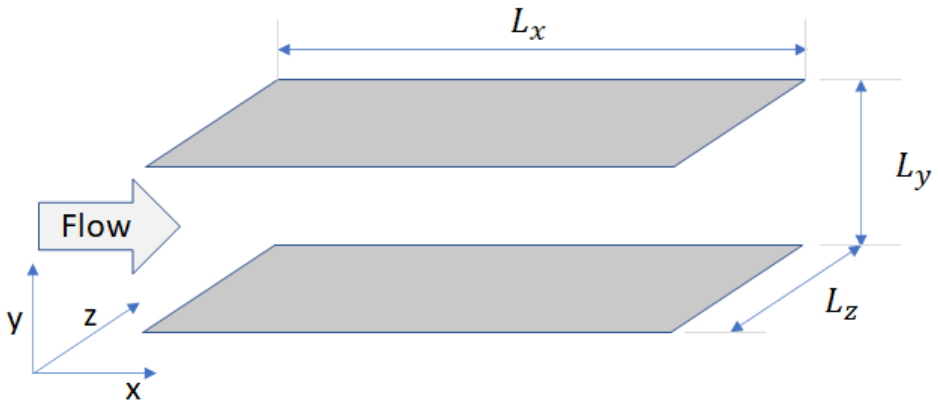


Figure 3.3. A schematic of channel flow.

3.3.3 Data preparation for training

In this study, the training datasets are provided by carrying out DNS of the turbulent channel flow. The no-slip boundary condition is chosen at the walls and periodic boundary condition is used in the streamwise and spanwise directions. In the wall-normal direction, a non-uniform grid is adopted and for the streamwise and spanwise directions, the Fourier collocation method is implemented and the grid points are distributed uniformly [47].

For the DNS which is conducted to obtain the training dataset, the spatial discretization in

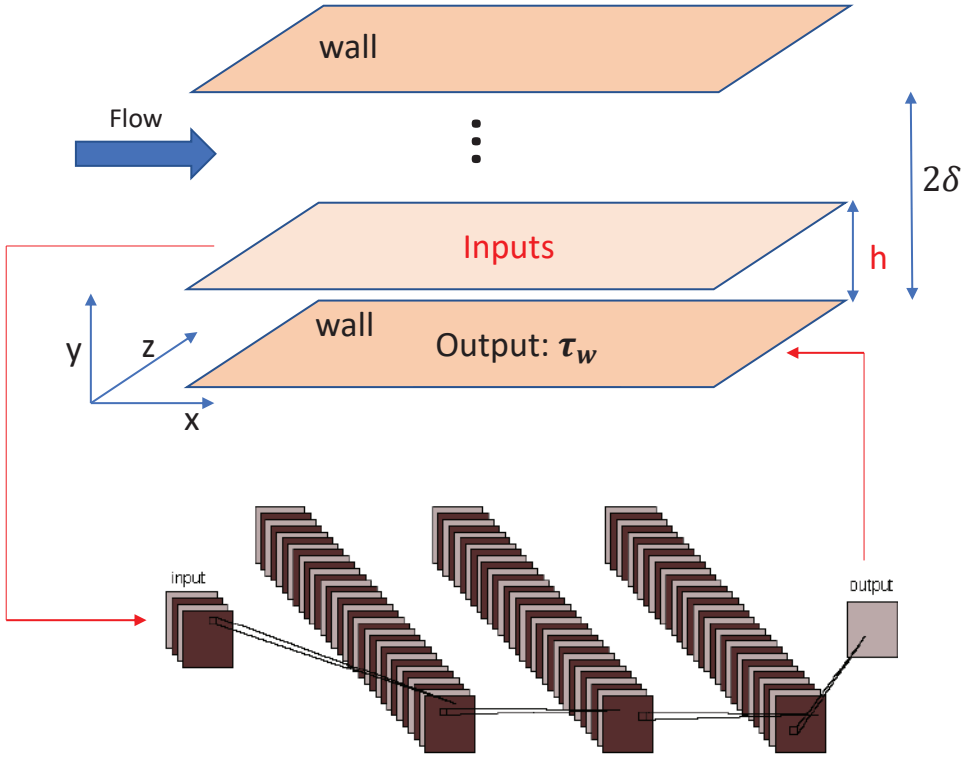


Figure 3.4. A schematic of wall modeling for turbulent channel flow using CNN.

the wall-normal direction is implemented using the sixth-order compact scheme [47, 107]. For the first order derivative the formaulation is

$$\beta f'_{i-2} + \alpha f'_{i-1} + f'_i + \alpha f'_{i+1} + \beta f'_{i+2} = a\frac{f_{i+1} - f_{i-1}}{2h} + b\frac{f_{i+2} - f_{i-2}}{4h} + c\frac{f_{i+3} - f_{i-3}}{6h} \quad (3.16)$$

where the constraints for the equation are

$$a + b + c = 1 + 2\alpha + 2\beta \quad (2^{nd}\ order) \quad (3.17)$$

$$a + 2^2 b + 3^2 c = 2\frac{3!}{2!}(\alpha + 2^2\beta) \quad (4^{th}\ order) \quad (3.18)$$

$$a + 2^4 b + 3^4 c = 2\frac{5!}{4!}(\alpha + 2^4\beta) \quad (6^{th}\ order) \quad (3.19)$$

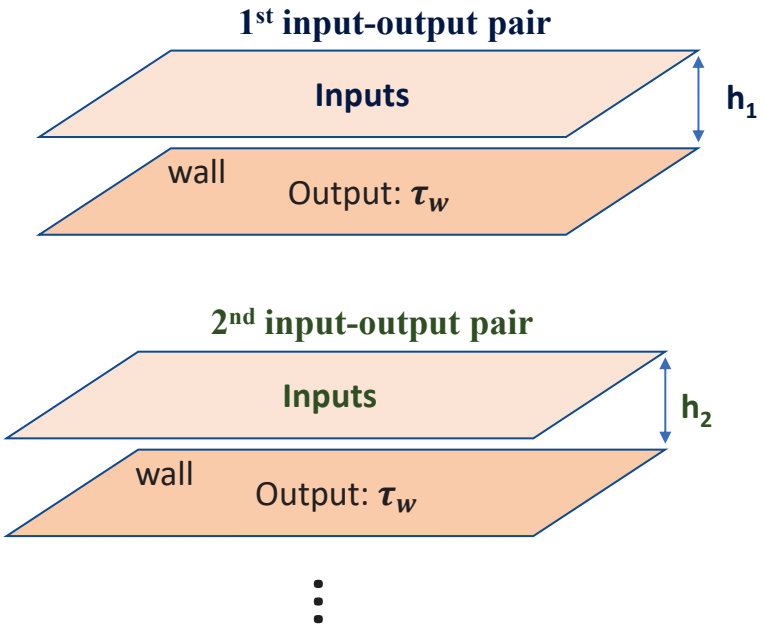


Figure 3.5. Input-output pairs in the training dataset ($N_x \times N_z = 64 \times 64$).

and for the second order derivative we have

$$\beta f''_{i-2}+\alpha f''_{i-1}+f''_i+\alpha f''_{i+1}+\beta f''_{i+2} = a\frac{f_{i+1}-2f_i+f_{i-1}}{h^2}+b\frac{f_{i+2}-2f_i+f_{i-2}}{4h^2}+c\frac{f_{i+3}-2f_i+f_{i-3}}{9h^2} \tag{3.20}$$

where the constraints for the equation are

$$a+b+c = 1+2\alpha+2\beta \quad (2^{nd}\ order) \tag{3.21}$$

$$a+2^2b+3^2c = \frac{4!}{2!}(\alpha+2^2\beta) \quad (4^{th}\ order) \tag{3.22}$$

$$a+2^4b+3^4c = \frac{6!}{4!}(\alpha+2^4\beta). \quad (6^{th}\ order) \tag{3.23}$$

At the boundary ($i = 1$), the following relation is used for the first order derivative

$$f_1' + \alpha f_2' = \frac{af_1 + bf_2 + cf_3 + df_4}{h} \tag{3.24}$$

and for the second order derivative, the implemented formula is

$$f_1'' + 11f_2'' = \frac{13f_1 - 27f_2 + 15f_3 - df_4}{h^2} \tag{3.25}$$

In the DNS code, the formulation presented by Kim et al. is implemented [47, 108] to solve the governing equations. The governing equations for the incompressible flow can be considered in the following form

$$\frac{\partial u_i}{\partial t} = -\frac{\partial p}{\partial x_i} + H_i + \frac{1}{Re_\tau}\nabla^2 u_i, \tag{3.26}$$

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{3.27}$$

where H_i includes the convective terms and the mean pressure gradient. All variables are nondimensionalized by the channel half-width δ and the friction velocity u_τ. Reynolds number in Eq. (3.26) is defined as $Re = u_\tau \delta/\nu$ where $\delta = 1$. The combination of Eq. (3.26) and (3.27) can give a fourth order equation for the wall-normal velocity (v) and a sceond order equation for the normal component of vorticity.

$$\frac{\partial \nabla^2 v}{\partial t} = h_v + \frac{1}{Re_\tau}\nabla^4 v, \tag{3.28}$$

$$\frac{\partial g}{\partial t} = h_g + \frac{1}{Re_\tau}\nabla^2 g, \tag{3.29}$$

After solving the Eq. (3.26) and (3.27), the following equations will be used to obtain the velocity components.

$$f + \frac{\partial v}{\partial y} = 0, \tag{3.30}$$

$$f = \frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}, \tag{3.31}$$

$$g = \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}, \tag{3.32}$$

$$h_v = -\frac{\partial}{\partial y}(\frac{\partial H_1}{\partial x} + \frac{\partial H_3}{\partial z}) + (\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2})H_2, \tag{3.33}$$

$$h_g = \frac{\partial H_1}{\partial z} - \frac{\partial H_3}{\partial x}. \tag{3.34}$$

In the homogeneous directions, the Fourier spectral method is employed and the grid is uniform. In the wall-normal direction, the sixth order compact scheme is used which will be explained in detail in this chapter. The time advancement is performed using a semi-implicit scheme. For the nonlinear terms and viscous terms, the Adams-Bashforth and Crank-Nicolson are used, respectively. After implementing the time-advancement methods, Eq. (3.28) reduces to the following set of equations

$$(1 - \frac{\Delta t}{2Re_\tau}\nabla^2)\phi^{n+1} = \frac{\Delta t}{2}(3h_v^n - h_v^{n-1}) + (1 + \frac{\Delta t}{2Re_\tau}\nabla^2)\phi^n, \tag{3.35}$$

$$\nabla^2 v^{n+1} = \phi^{n+1}, \tag{3.36}$$

$$v^{n+1}(0) = v^{n+1}(L_y) = \frac{\partial v^{n+1}}{\partial y}(0) = \frac{\partial v^{n+1}}{\partial y}(L_y) = 0. \tag{3.37}$$

Considering the boundary conditions presented in Eq. (3.37), the solution of the Eq. (3.35) and (3.36) has the following form

$$\phi = \phi_p + c_1\phi_1^{n+1} + c_2\phi_2^{n+1} \tag{3.38}$$

$$v = v_p + c_1 v_1^{n+1} + c_2 v_2^{n+1} \tag{3.39}$$

where ϕ_1, ϕ_2, v_1 and v_2 are the general solutions of the homogeneous differential equation. Therefore, the equations for the homogeneous part are as follows

$$(1 - \frac{\Delta t}{2Re_\tau}\nabla^2)\phi_1^{n+1} = 0, \quad \phi_1^{n+1}(0) = 1, \quad \phi_1^{n+1}(L_y) = 0 \tag{3.40}$$

$$\nabla^2 v_1^{n+1} = \phi_1^{n+1}, \quad v_1^{n+1}(0) = v_1^{n+1}(L_y) = 0 \tag{3.41}$$

which leads to the following solutions for ϕ_1 and v_1

$$\phi_1 = \frac{e^{\sqrt{c}(y-L_y)} - e^{-\sqrt{c}(y+L_y)}}{1 - e^{-2\sqrt{c}L_y}} \tag{3.42}$$

$$v_1 = \frac{2Re_\tau}{\Delta t}\Big(\frac{e^{|k|(y-L_y)} + e^{-|k|(y+L_y)}}{1 - e^{-2|k|L_y}} - \phi_1\Big), \tag{3.43}$$

and

$$(1 - \frac{\Delta t}{2Re_\tau}\nabla^2)\phi_2^{n+1} = 0, \quad \phi_2^{n+1}(0) = 0, \quad \phi_2^{n+1}(L_y) = 1 \tag{3.44}$$

$$\nabla^2 v_2^{n+1} = \phi_2^{n+1}, \quad v_2^{n+1}(0) = v_2^{n+1}(L_y) = 0, \tag{3.45}$$

which leads to the following solutions for ϕ_2 and v_2

$$\phi_2 = \frac{e^{\sqrt{c}(y-2L_y)} + e^{-\sqrt{c}y}}{1 - e^{-2\sqrt{c}L_y}} \tag{3.46}$$

$$v_2 = \frac{2Re_\tau}{\Delta t}\Big(\frac{e^{|k|(y-2L_y)} + e^{-|k|y}}{1 - e^{-2|k|L_y}} + \phi_2\Big), \tag{3.47}$$

where c is

$$c = \frac{2Re}{\Delta t}\Big(1 + \frac{\Delta t}{2Re_\tau}|k^2|\Big). \tag{3.48}$$

For the particular solution, ϕ_p we have

$$(1 - \frac{\Delta t}{2Re_\tau}\nabla^2)\phi_p^{n+1} = \frac{\Delta t}{2}(3h_v^n - h_v^{n-1}) + (1 + \frac{\Delta t}{2Re_\tau}\nabla^2)\phi^n, \tag{3.49}$$

$$\phi_p^{n+1}(0) = \phi_p^{n+1}(L_y) = 0, \tag{3.50}$$

$$\nabla^2 v_p^{n+1} = \phi_p^{n+1}, v_p^{n+1}(0) = v_p^{n+1}(L_y) = 0. \tag{3.51}$$

In order to obtain the coefficients c_1 and c_2 of the homogeneous part, we take the derivative of Eq. (3.39) in the wall-normal direction and apply the boundary conditions of Eq. (3.37).

Table 3.1. DNS parameters.

Re_τ	$L_x \times L_y \times L_z$	$N_x \times N_y \times N_z$	Δx^+	Δz^+	Δy^+_{min}	Δy^+_{max}
180	$4\pi \times 2 \times 2\pi$	$192 \times 128 \times 160$	11.781	7.069	0.905	5.358
400	$2\pi \times 2 \times \pi$	$256 \times 192 \times 256$	9.817	4.909	0.920	7.880
600	$\pi \times 2 \times \pi/2$	$256 \times 256 \times 256$	7.363	3.682	0.788	8.146
800	$\pi \times 2 \times \pi$	$384 \times 384 \times 768$	6.545	3.272	0.449	7.749

$$\left.\frac{\partial v}{\partial y}\right|_{y=0} = \left.\frac{\partial v_p}{\partial y}\right|_{y=0} + c_1 \left.\frac{\partial v_1}{\partial y}\right|_{y=0} + c_2 \left.\frac{\partial v_2}{\partial y}\right|_{y=0}, \tag{3.52}$$

$$\left.\frac{\partial v}{\partial y}\right|_{y=L_y} = \left.\frac{\partial v_p}{\partial y}\right|_{y=L_y} + c_1 \left.\frac{\partial v_1}{\partial y}\right|_{y=L_y} + c_2 \left.\frac{\partial v_2}{\partial y}\right|_{y=L_y}, \tag{3.53}$$

Solving the above system of equations leads to

$$c_1 = \frac{\frac{\partial v_p}{\partial y}|_{y=0} \frac{\partial v_2}{\partial y}|_{y=L_y} - \frac{\partial v_p}{\partial y}|_{y=L_y} \frac{\partial v_2}{\partial y}|_{y=0}}{\frac{\partial v_1}{\partial y}|_{y=0} \frac{\partial v_2}{\partial y}|_{y=L_y} - \frac{\partial v_2}{\partial y}|_{y=0} \frac{\partial v_1}{\partial y}|_{y=L_y}} \tag{3.54}$$

$$c_2 = \frac{\frac{\partial v_p}{\partial y}|_{y=0} \frac{\partial v_1}{\partial y}|_{y=L_y} - \frac{\partial v_p}{\partial y}|_{y=L_y} \frac{\partial v_1}{\partial y}|_{y=0}}{\frac{\partial v_1}{\partial y}|_{y=0} \frac{\partial v_2}{\partial y}|_{y=L_y} - \frac{\partial v_2}{\partial y}|_{y=0} \frac{\partial v_1}{\partial y}|_{y=L_y}} \tag{3.55}$$

The data used for training and testing of the data-driven wall model are obtained by the DNS at Re_τ =180, 400, 600 and 800, where $Re_\tau = u_\tau \delta/\nu$. The DNS parameters are shown in Table 3.1, where L and N represent the domain size and the number of grid points, respectively. Δx^+, Δy^+, and Δz^+ denote the non-dimensionalized grid spacings in the streamwise, wall-normal and spanwise directions, respectively. To check the validity of the DNS, the mean streamwise velocity profiles resulted from the DNS are shown in Fig. 3.6 which confirm that the profiles of the mean velocity follow the law of the wall.

The Gaussain filter, which has a filter kernel of

$$\hat{G}(k) = \exp(-\frac{k^2\Delta^2}{24}) \tag{3.56}$$

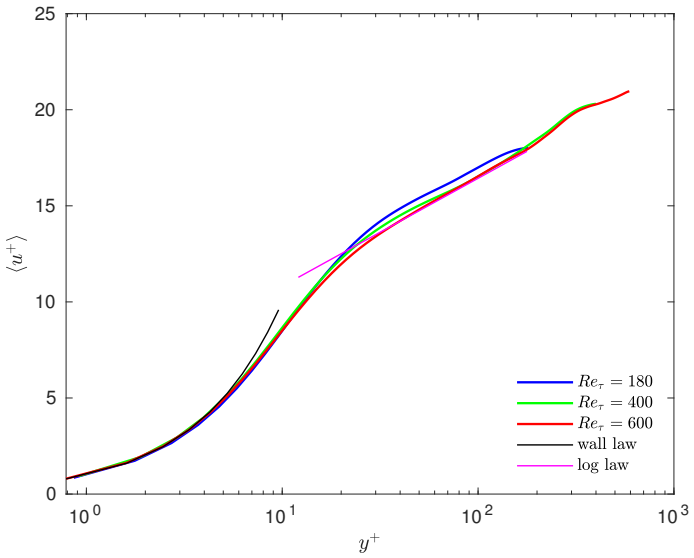


Figure 3.6. Validity of DNS data; wall law and log law represent $u^+ = y^+$ and $u^+ = \frac{1}{k}\log(y^+) + B$, respectively, where $k = 0.41$ and $B = 5.2$

in the spectral space is used to filter the DNS data employed for the training where Δ denotes the filter width. Unlike the top-hat filter and the cut-off filter, the Gaussian filter is compact and has a smooth response in both the physical and spectral spaces which can make it possible to achieve a more reliable data-driven model. In the previous studies on the data-driven subgrid-scale modeling, it was shown that the Gaussian filter leads to a higher correlation coefficient between the true solution and the data-driven model [53, 62].

3.3.4 Input choices

In order to identify the proper inputs for the CNN wall model, it is crucial to understand the flow behavior near the wall. The flow in this region can be governed by the thin boundary layer equation (TBLE), which is a simplified form of the momentum equation and can be used for the near-wall flow. For the streamwise velocity, it is given by [99]

$$\frac{\partial \overline{u}}{\partial t} + \frac{\partial (\overline{u}\,\overline{u_j})}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial}{\partial y}\left[(\nu + \nu_t)\frac{\partial \overline{u}}{\partial y}\right] \tag{3.57}$$

Integrating Eq. (3.57) in the wall-normal direction and performing time averaging leads to an equation for the wall shear stress

$$\langle \tau_w \rangle = \nu \frac{\partial \langle \overline{u} \rangle}{\partial y}\Big|_w = -\int_0^{h_w} \frac{1}{\rho}\frac{\partial \langle \overline{p} \rangle}{\partial x}\, dy + (\nu + \nu_{t,h_w})\frac{\partial \langle \overline{u} \rangle}{\partial y}\Big|_{h_w} - \int_0^{h_w} \Big(\frac{\partial \langle \bar{u}\bar{u} \rangle}{\partial x} + \frac{\partial \langle \bar{u}\bar{w} \rangle}{\partial z}\Big)\, dy - \langle \bar{u}\bar{v} \rangle\Big|_{h_w} \tag{3.58}$$

where h_w represents the wall-normal distance. This equation can give an insight for choosing the proper inputs for CNN. By considering the terms in Eq. (3.58), different input choices can be tested for the CNN. As all the terms in Eq. (3.58) are the functions of the primitive quantities of the flow, the velocity components (u, v, w) can also be a candidate for the input features. The effect of input choices will be checked in Sec. 3.3.5.

3.3.5 Parametric study

In order to find the best structure for the CNN wall model, initially, a study has been performed on the CNN input choices. For the training process of CNN in this appendix, the filtered DNS data of $Re_\tau = 180$ are used. The input is fed into the CNN in the form of $N_x \times N_z \times N_f$ where $N_x \times N_z = 48 \times 40$ and N_f is the number of features. For the input features, the data in the the buffer layer and the logarithmic layer are included in the training dataset and the considered CNN consists of 5 convolutional layers with 32 filters in each layer with the size of 5×5. A skip-connected layer is considered in the CNN architecture as it is shown to make the training easier for the deep networks [109].

For the CNNs trained using different input choices, the correlation coefficient between the wall shear stress predicted by the CNN wall model and obtained using the filtered DNS data,

$$C.C. = \frac{\langle (\tau_w^{CNN} - \langle \tau_w^{CNN} \rangle)(\tau_w^{DNS} - \langle \tau_w^{DNS} \rangle) \rangle}{[\langle (\tau_w^{CNN} - \langle \tau_w^{CNN} \rangle)^2 \rangle]^{\frac{1}{2}} [\langle (\tau_w^{DNS} - \langle \tau_w^{DNS} \rangle)^2 \rangle]^{\frac{1}{2}}} \tag{3.59}$$

is calculated to evaluate the impact of the CNN inputs on the model performance. Table 3.2 shows the effect of the input choices on the correlation coefficient between the wall shear stress calculated from the filtered DNS data and predicted by the CNN models. Comparing $C1$ with $C2$ clearly reveals that adding velocity gradient to the input variables increases the correlation coefficient. However, comparing $C2$ with $C4$ and $C3$ with $C4$ shows that adding either u or uv

to the input features can increase the correlation coefficient considerably. It is also seen in Table 3.2 that choosing velocity components as the CNN inputs ($C5$) gives rise to a high correlation coefficient, comparable to $C4$. Therefore, $C4$ and $C5$ which correspond to the input choices of $\{u, \partial u/\partial y, uv, h\}$ and $\{u, v, w\}$, respectively, are selected as the input choices of CNN. The CNN wall models obtained using these two sets of input will be compared with each other.

Table 3.2. Input choices for CNN, correlation coefficient (C.C.) for training (tr) and validation (val) datasets.

Case	Input	$C.C._{tr}$	$C.C._{val}$
$C1$	u, h	0.7015	0.6999
$C2$	$u, \partial u/\partial y, h$	0.7290	0.7230
$C3$	$\partial u/\partial y, uv, h$	0.7108	0.7035
$C4$	$u, \partial u/\partial y, uv, h$	0.7833	0.7796
$C5$	u, v, w	0.7654	0.7631

After selecting the input features, a parametric study is performed to determine the hyperparameters of the CNN. Initially, the impact of the filter size on the model prediction is investigated. Filter size is one of the parameters that contributes to enlarging the receptive field used for the convolution operation. The filter size of the CNN has a square shape. Fig. 3.7 indicates that for a small filter size the correlation coefficient is not large. On the other hand, it shows that increasing the filter size greatly does not necessarily enhance the performance of the model. Based on Fig. 3.7, $f_h = 9$ is chosen for the CNN as the highest correlation coefficient is achieved for this filter size.

It has been indicated in the previous studies that adding the skip-connected layers can make the training more stable [110]. In some studies on turbulence modeling, it has also been shown that using ResNet can lead to a model with reasonable accuracy [78, 86]. After checking the effect of adding skip-connected layers, it has been found that adding two skip-connected layers leads to the highest correlation coefficient for the CNN model.

The number of filters and the depth of CNN are the other parameters which affect the performance of the CNN. Like filter size, the the depth of CNN can enlarge the receptive field. It has been shown that for the large-scale data increasing the number of convolutional layers can improve the accuracy of the CNN in image classification [111]. Although increasing the number of filters generally improves the accuracy of CNN, choosing a very large number of filters or

a highly deep CNN may not lead to a more accurate model. Furthermore, it can increase the computational costs required for the training process and the wall model when embedded in an actual simulation, significantly. The effect of the number of filters (f_n) on the CNN model performance is shown in Fig. 3.8. As seen in this figure, choosing a very large number of filters does not necessarily lead to a more accurate model and for $f_n = 62$, the correlation coefficient reaches its highest value. Also a parametric study on the number of convolutional layers (N_{CL}) has revealed that the maximum correlation coefficient is obtained for $N_{CL} = 6$.

The selected values for the CNN hyperparameters and the correlation coefficient obtained by training the network using these parameters are presented in Table 3.3. As seen in this table, the chosen CNN architecture gives rise to a high correlation coefficient both for the training and the validation datasets. The schematic of the CNN is shown in Fig. 3.9 as well, where $m = 9$ and $n = 62$.

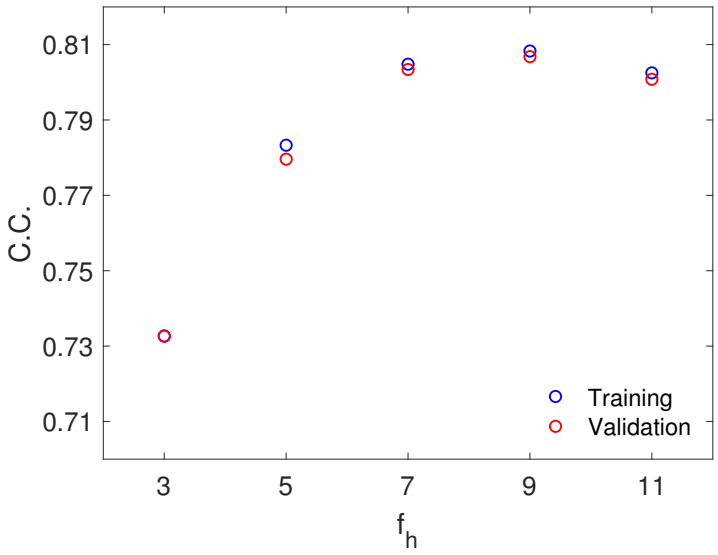


Figure 3.7. Effect of filter size (f_h) on the correlation coefficient between the wall shear stress predicted by the CNN model and calculated using the filtered DNS data.

Table 3.3. Correlation coefficient for CNN trained using best values for hyperparameters.

inputs	f_h	N_{CL}	f_n	$C.C._{tr}$	$C.C._{val}$
$u, \partial u/\partial y, uv, h$	9	6	62	0.9498	0.9472

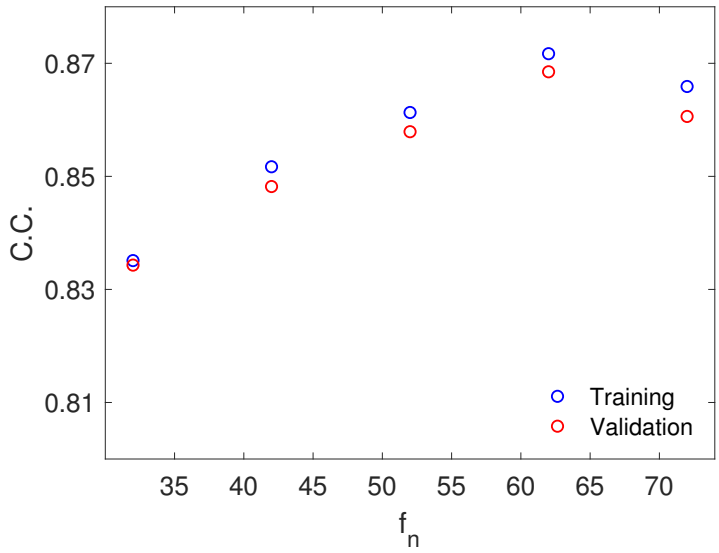


Figure 3.8. Effect of number of filters (f_n) on the correlation coefficient between the wall shear stress predicted by the CNN model and calculated using the filtered DNS data.

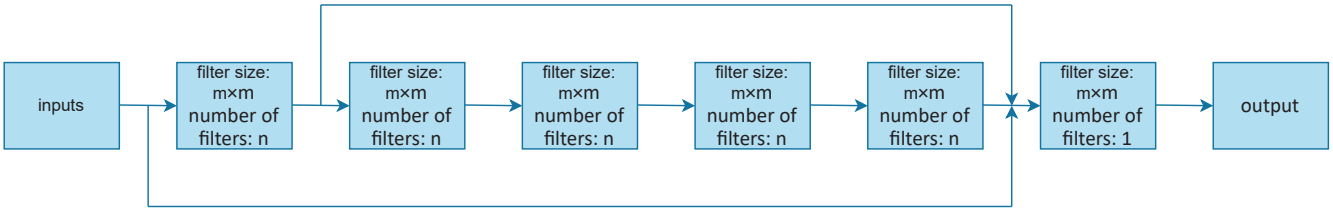


Figure 3.9. Chosen architecture for CNN after parametric study.

3.4 Results and discussion

3.4.1 *A Priori* study

In this section, the same CNN architecture as that obtained in Sec. 3.3.5 is used to develop the CNN wall model. Due to the differences in the characteristics of the near-wall regions, the model can be more reliable if it is limited to a particular region. The flow in the inner layer, $y/\delta < 0.1$, has a universal behavior and the velocity in this region is determined by u_τ and y^+. On the other hand, the logarithmic layer has its own characteristics and the mean flow follows the law of the wall in this region. Many of the existing wall-stress models are based on the logarithmic law of the wall [21, 27, 28, 99]. Thus, two different ranges for the wall-normal distance are chosen for the CNN input data to train the convolutional neural network. In the first case, only the data in the inner layer, excluding $y^+ \leq 10$, are considered for training the model. The data in the region of $y^+ \leq 10$ are excluded as it includes the effects of the viscous sublayer, and the departure from the linear relation, $u^+ = y^+$, is insignificant in this region [4]. In the second case, the training

data are limited to the logarithmic region where the log-law holds ($y^+ > 30$ and $y/\delta \lesssim 0.15$). Therefore, in total, four CNNs are trained based on the chosen CNN inputs and the range of the wall-normal distance considered for the input features as shown in Table 3.4. In the present study, the inner layer-based and log layer-based wall models are referred to {I1, I2} and {L1, L2}, respectively. It should be noticed that as already discussed in this section for the inner-layer model, $y^+ \leq 10$ is not included in the training range of the models. In the comparison between considering the whole range of $y^+ > 10$ and $y/\delta \lesssim 0.15$ and a limited region, like the regions shown in Table 3.4, for the training process, the latter has better performance when applied to a higher Reynolds number flow.

For the training process, the data of $Re_\tau = 400$ are used. In the streamwise and spanwise directions, the DNS data are filtered with the filter sizes of $\overline{\Delta x}^+ = 39.3$ and $\overline{\Delta z}^+ = 19.6$. The number of grid points for the CNN inputs is $N_x \times N_z = 64 \times 64$. The total number of snapshots used for the training datasets is 422; for one flow-through time, 8 snapshots of the instantaneous flow field obtained using DNS are extracted. The instants of the snapshots are well separated so that they are nearly uncorrelated. Based on the available DNS data in the selected ranges of the wall-normal distance, 15 and 11 wall-parallel planes are chosen for the inner layer-based and log layer-based models, respectively, for each snapshot. Therefore, the total number of sample data used for training the inner layer and log layer wall models is 6330 and 4642, respectively, which is sufficiently large and is comparable to the previous studies where CNN is employed for regression (e.g. SGS modeling). [75, 83, 112]. In the training process, about 70% and 30% of the data are used for training and validation, respectively.

The correlation coefficients for the training and the validation datasets are shown in Table 3.4. As seen in this table, the CNN wall models based on both inputs show good performance and achieve a high correlation coefficient. However, the correlation coefficient is slightly higher for the CNN with $\{u, \partial u/\partial y, uv, h\}$ as the inputs. Furthermore, for both sets of input, a higher correlation coefficient is achieved when the data in the inner layer (excluding $y^+ \leq 10$) are used for training the model.

In this section, the CNN wall models are evaluated for two values of h^+. Since all CNN models show good performance when the wall-normal distance for the wall model inputs is included in or is slightly outside the training range of the models, the h^+ values are chosen to

test the CNN models outside the training ranges. The first one ($h^+ = 17.89$ or $h/\delta = 0.045$) is not included in the training range of L1 and L2 and the second chosen value ($h^+ = 59.24$ or $h/\delta = 0.148$) is not included in the training of I1 and I2.

Also, in order to further check the effect of choosing the primitive flow variables in establishing a wall model, two other CNNs are trained based on the data in the inner layer (excluding $h^+ \leq 10$) as well, as shown in Table 3.5. As shown in this table, when one of the input choices is eliminated, a high correlation coefficient can still be achieved which means that the CNN has still the ability to extract the hidden features. Additionally, it can be seen that adding the wall-normal velocity as the input choice can increase the correlation coefficient, but still the highest correlation coefficient can be achieved when all the velocity components are included in the CNN inputs. For this reason, all velocity components are included in the input choices. However, comparing the results of Table 3.4 and Table 3.5 shows that compared to the wall-normal velocity, the spanwise velocity does not have a noticeable effect in increasing the correlation coefficient.

Table 3.4. Average correlation coefficient for training based on the data of $Re_\tau = 400$.

Case	Inputs	Input data range	$C.C._{tr}$	$C.C._{val}$
I1	$u, \partial u/\partial y, uv, h$	$h^+ > 10, h/\delta < 0.1$	0.9324	0.9172
L1	$u, \partial u/\partial y, uv, h$	$h^+ > 30, h/\delta \lesssim 0.15$	0.8802	0.8558
I2	u, v, w	$h^+ > 10, h/\delta < 0.1$	0.9149	0.8997
L2	u, v, w	$h^+ > 30, h/\delta \lesssim 0.15$	0.8514	0.8308

Table 3.5. Average correlation coefficient for training based on the data of $Re_\tau = 400$ for input choices other than those selected.

Inputs	$C.C._{tr}$	$C.C._{val}$
u	0.8801	0.8655
u, v	0.9119	0.8979

3.4.1.1 Wall model input at $h^+ = 17.89$

The distributions of the wall shear stress are shown in Fig. 3.10 for $h^+ = 17.89$ ($h/\delta = 0.045$). The results of the CNN wall models are compared with two existing wall-stress models, the shifted wall model and the ODE-based wall model. As shown in Fig. 3.10, good agreement can be seen between the wall shear stress calculated from the filtered DNS data and the wall

shear stress predicted by the CNN wall models compared to the existing wall models. However, taking a closer look, it can be seen that the models L1 and L2 are less accurate than I1 and I2 in predicting the distribution of the wall shear stress, and the wall shear stress is underpredicted in some locations, which can be due to the fact that the chosen h^+ is not included in the training process for L1 and L2 and is much smaller than the range of h^+ for training of these two models. The results of Fig. 3.10 are compatible with Table 3.8 which shows that lower correlation coefficient and larger error are obtained for L1 and L2 when $h^+ = 17.89$ compared to the other CNN models. The slope of the regression line obtained from a plot with the wall shear stress predicted by the wall model and calculated from the filtered DNS data as the vertical and horizontal axes, respectively, is shown in Table 3.8 for the wall models. The results for the slope of the regression line are consistent with those of the correlation coefficient and error and confirm that for $h^+ = 17.89$, I1 and I2 outperform the other models.

The joint PDF of the wall shear stress obtained by the wall models and that of the filtered DNS data is compared in Fig. 3.11. It can be seen that I1 and I2 show better accuracy than the other models as the PDF is concentrated near the diagonal line. L1 and L2 also show overall reasonable accuracy, roughly better than the shifted and ODE-based wall models, but the joint PDF for these two models slightly deviates from the diagonal line.

The regression line for the plot of the wall shear stress versus velocity is shown in Fig. 3.12. As seen in this figure, for the filtered DNS data, the plot is scattered as the wall shear stress is not only the function of velocity. It should be mentioned that as shown in Fig. 3.13, for the CNN model, the plot is scattered as well since there are other inputs for the CNN model which are involved in the calculation of the wall shear stress. For both the filtered DNS data and the CNN model, the wall shear stress is concentrated near the regression line. However, there are some discrepancies between the scatter plot for the CNN model and the filtered DNS data which is expected for the data-driven wall model. The CNN wall model is established by the training process and it is most likely that prediction becomes inaccurate at some points due to the lack of physical information. In Fig. 3.12 only the regression lines are shown for the CNN models so that it is easier to compare them with the other wall models. On the other hand, for the ODE-based model, there is no scatter plot at a certain wall-normal distance as they are function of the velocity. For the shifted model the plot will not be scattered as well if the predicted wall shear stress is shifted by Δ_s. According to Fig. 3.12, it can be seen that on

average, the prediction of L1 and L2 is nearly similar to the shifted model and is less accurate than I1 and I2, especially for the large values of velocity.

The mean value of the wall shear stress which corresponds to the average of the wall shear stress over the wall parallel ($x - z$) plane is shown in Table 3.6 for one of the CNN models (I2) and the existing models. As shown in this table, the mean values predicted by the I2 and ODE-based models are close to the mean wall shear stress calculated using the filtered DNS data. The value predicted by the shifted model is underpredicted and is farther from that of filtered DNS compared to the other two models which can be expected to some extent since the h^+ chosen in this case is not located in the logarithmic layer.

Table 3.6. Mean value of wall shear stress (τ_w) for filtered DNS, I2, and existing models. $h^+ = 17.89$ for the wall model inputs.

Case	fDNS	I2	shifted	ODE
τ_w^{mean}	0.9913	1.0085	0.9264	1.0072

3.4.1.2 Wall model input at $h^+ = 59.24$

The distributions of the wall shear stress for $h^+ = 59.24$ ($h/\delta = 0.148$) are shown in Fig. 3.14. As seen in this figure, for this case, L1 and L2 perform better than the other wall models. It should be noted that in this case, the chosen h^+ is not considered in the training process for I1 and I2. However, it can be seen that on the whole, these models are still able to predict the wall shear stress. By comparing the distributions of the wall shear stress with the velocity distribution in Fig. 3.14, it can also be found that as shown in Fig. 3.10, the ODE-based wall model is perfectly correlated with the wall shear stress. Additionally, in the shifted wall model, the predicted wall shear stress is perfectly correlated with the velocity at the off-wall plane if it is shifted by Δ_s in the upstream direction. This is in contrast to the results of the study by Mathis et al. [30] which shows that the wall shear stress should be poorly correlated with the velocity at an off-wall plane. Unlike the existing wall models used in the present study and similar to the filtered DNS data, the wall shear stress predicted by the CNN models is poorly correlated with the velocity at the plane off the wall (≈ 0.3) which is along with the results obtained by Mathis et al. [30]. The slope of the regression line (R) is shown in Table 3.9 for the wall models together with the results of the correlation coefficient and error; it shows that the log layer-based models show better performance on average for $h^+ = 59.24$. It should be mentioned that for the CNN

models, the slope of the regression line is affected by the inaccurate prediction of the wall shear stress at some points, especially where the values of the wall shear stress are large; however, the CNN models still outperform the existing models.

The joint PDF for the wall models is presented in Fig. 3.15 which indicates that on the whole all CNN models show roughly good accuracy, better than the existing models, but the log layer-based CNN models (L1 and L2) are more accurate than I1 and I2. According to Fig. 3.15, the shifted and ODE-based wall models show similar behavior in this case.

Fig. 3.16 shows the plot of wall shear stress versus velocity at the distance $h^+ = 59.24$ from the wall. By comparing this figure with Fig. 3.12, it can be realized that for a larger wall-normal distance, the slope of the regression lines gets smaller. As shown in Fig. 3.16, I2 shows similar performance to the ODE-based model, but I1 has good accuracy on average although h^+ is outside of its training range. It should be noted that for $h^+ = 59.24$, although the chosen h^+ is outside of the training range of I1 and I2, these models have slightly better accuracy compared to L1 and L2 in Sec. 3.4.1.1, where h^+ is not included in their training range.

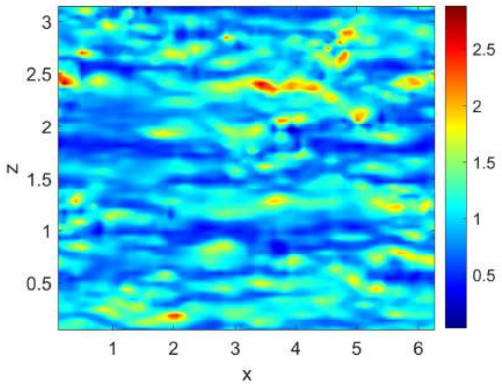
Like Sec. 3.4.1.1, the plane-averaged wall shear stress is calculated and is shown in Table 3.7 for the CNN model, I2, and the existing models. As seen in this Table, the mean value predicted by I2 is closer to the filtered DNS than the existing models. Also, the value predicted by the ODE-based model is closer to that of the filtered DNS compared to the shifted wall model. However, in comparison to the results of Table 3.6, the value obtained using shifted model is closer to that of the filtered DNS. It should be noted that in contrast to Sec. 3.4.1.1, in this case, the h^+ chosen is located in the logarithmic layer.

Table 3.7. Mean value of wall shear stress (τ_w) for filtered DNS, I2, and existing models. $h^+ = 59.24$ for the wall model inputs.

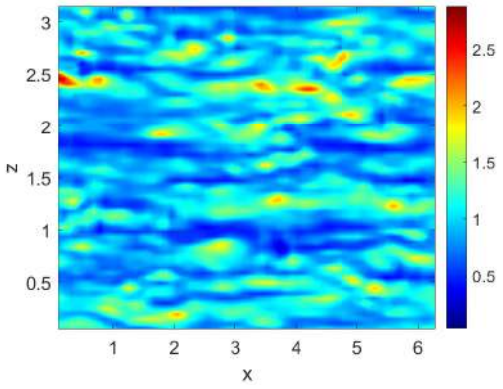
Case	fDNS	I2	shifted	ODE
τ_w^{mean}	0.9913	0.9941	1.0161	0.9847

3.4.2 Evaluation of CNN wall model

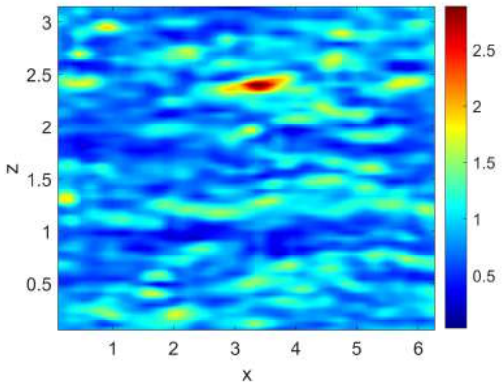
It is important for the data-driven models to have a reasonable prediction when tested under conditions different from those of the data used for the training process. In this section, in order



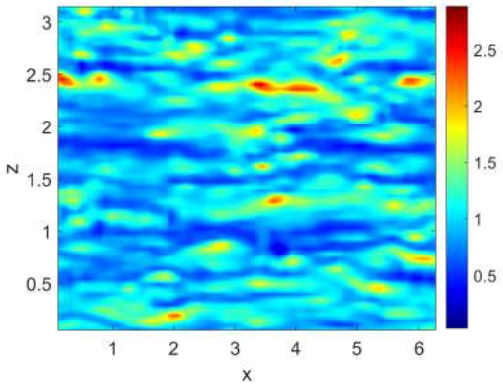
(a) fDNS



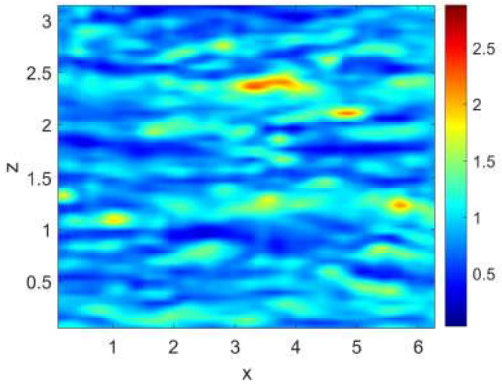
(b) l1



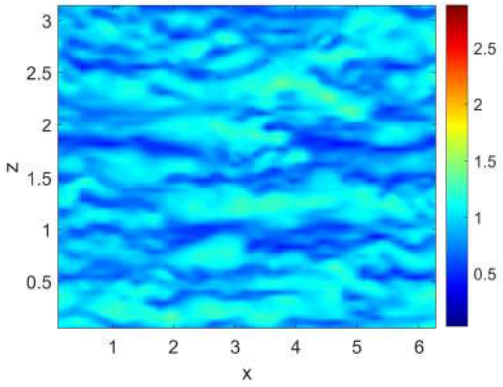
(c) L1



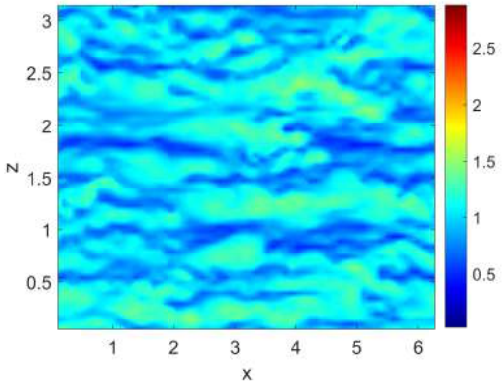
(d) l2



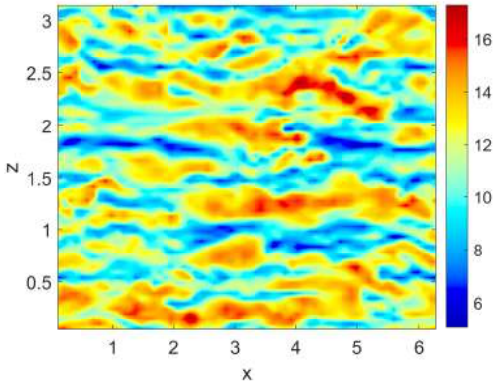
(e) L2



(f) shifted model

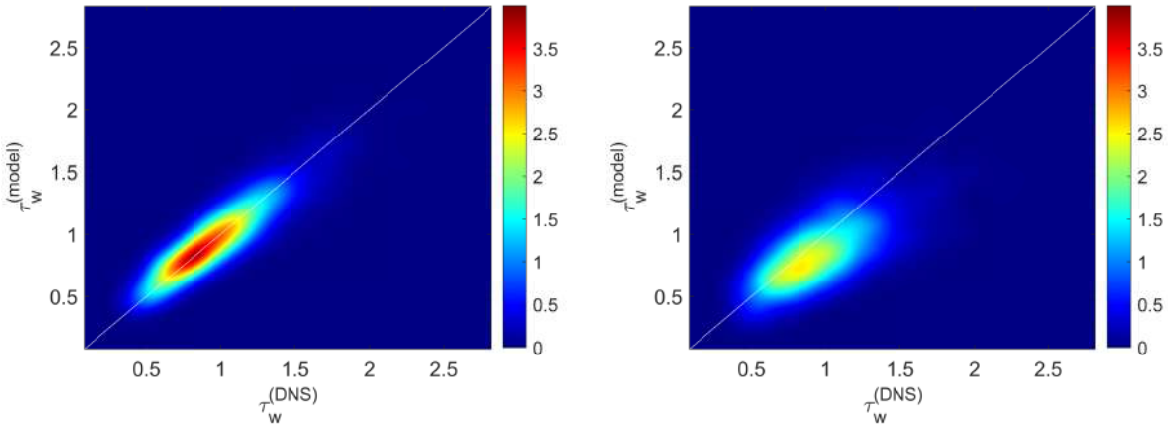


(g) ODE-based model

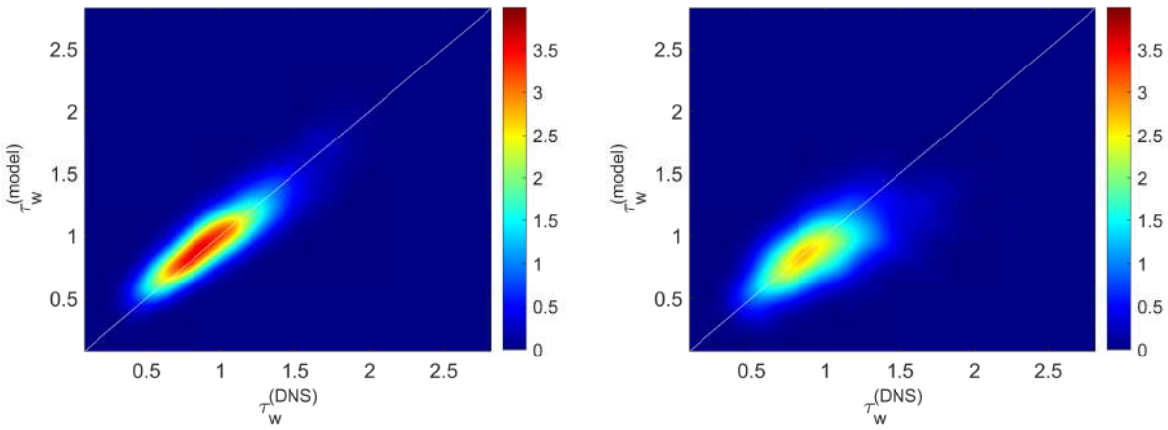


(h) u

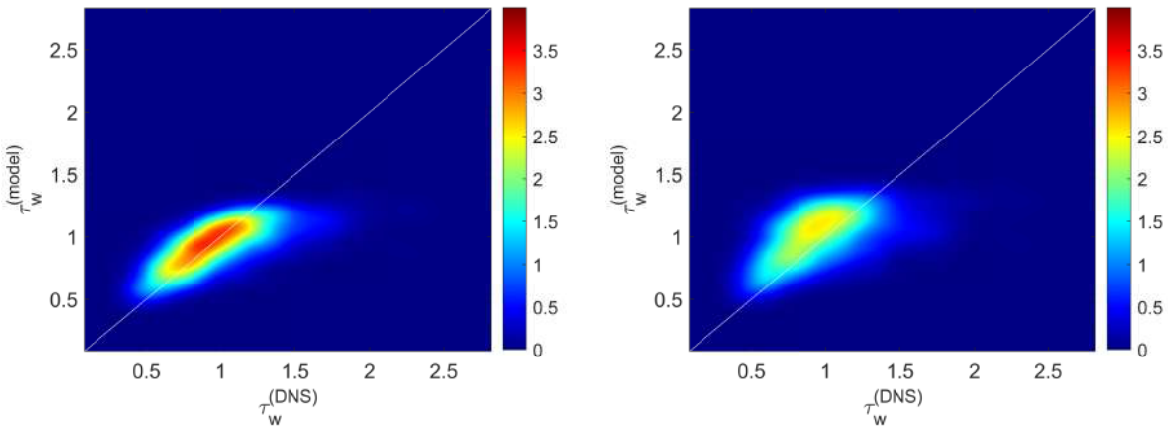
Figure 3.10. Spatial distribution of wall shear stress (τ_w) and streamwise velocity (u) for $Re_\tau = 400$ at $h^+ = 17.89$. fDNS in (a) refers to τ_w calculated using the filtered DNS data.



(a) l1 (b) L1



(c) l2 (d) L2



(e) shifted model (f) ODE-based model

Figure 3.11. Joint PDF of wall shear stress (τ_w) for $Re_\tau = 400$ at $h^+ = 17.89$.

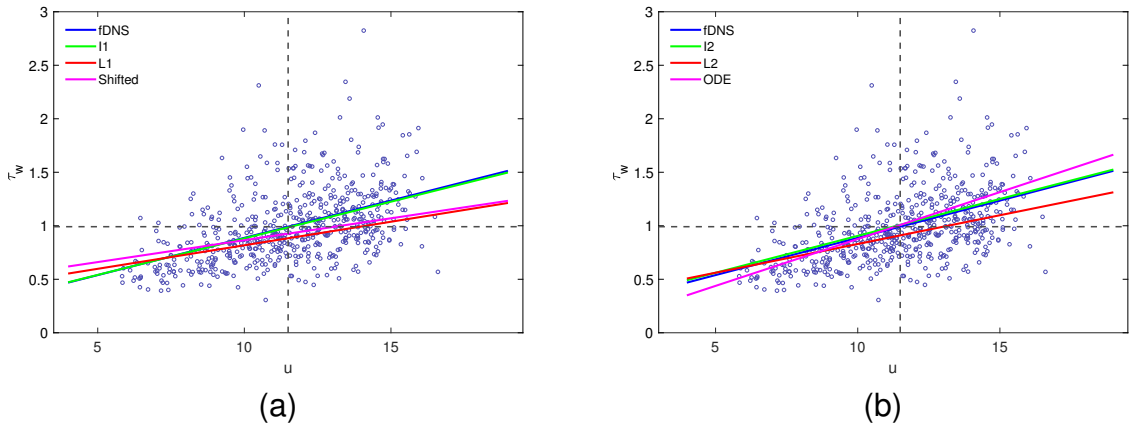


Figure 3.12. Wall shear stress as a function of streamwise velocity at $h^+ = 17.89$. Comparison between (a) filtered DNS, I1, L1, and shifted model, and (b) filtered DNS, I2, L2, and ODE-based model.

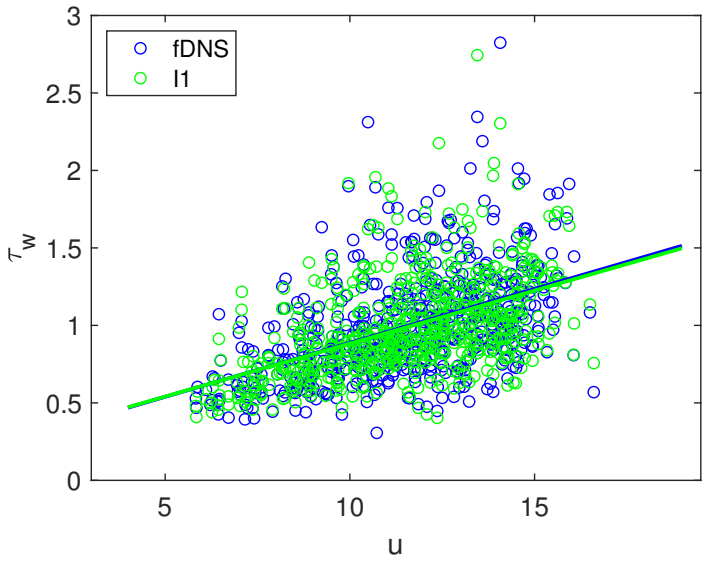
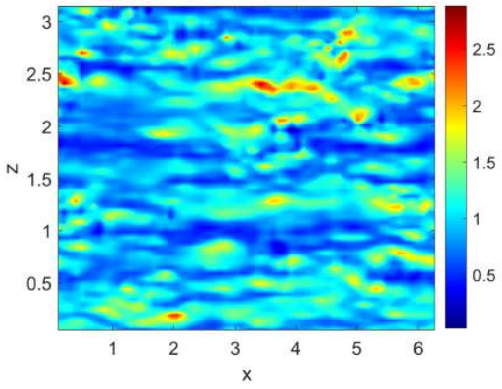


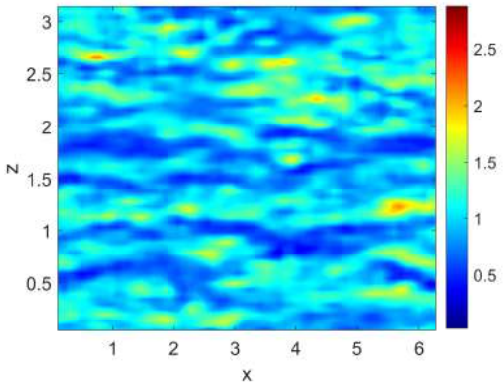
Figure 3.13. Scatter plot of wall shear stress (τ_w) versus velocity (u) at $h^+ = 17.89$. The blue and green lines show the regression lines for the filtered DNS data and CNN model (I1), respectively.

Table 3.8. Slope of the regression line (R) obtained from a plot with τ_w^{DNS} and τ_w^{model} as the horizontal and vertical axis, respectively; correlation coefficient (C.C.) between the model prediction and τ_w calculated using filtered DNS data, and root mean squared error (RMSE) for the wall models ($h^+ = 17.89$, $Re_\tau = 400$). Shifted and ODE represent the shifted and ODE-based wall models.

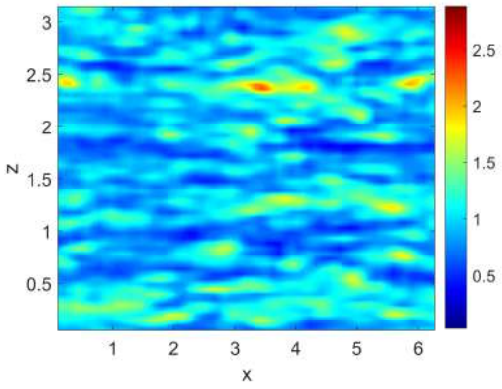
Case	R	C.C.	RMSE
I1	0.8336	0.9033	0.1448
L1	0.5954	0.6723	0.2810
I2	0.8398	0.9043	0.1450
L2	0.5260	0.6408	0.2777
shifted	0.3632	0.6593	0.2639
ODE	0.3003	0.4885	0.2973



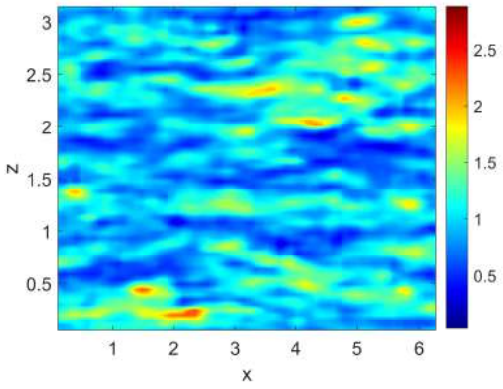
(a) fDNS



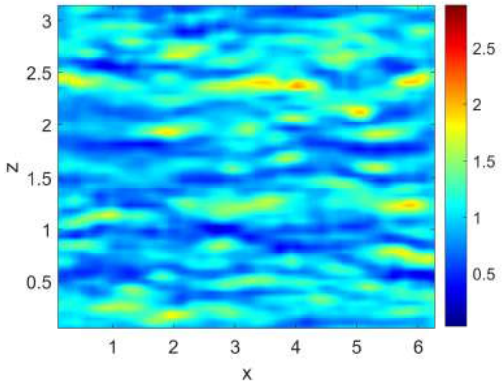
(b) l1



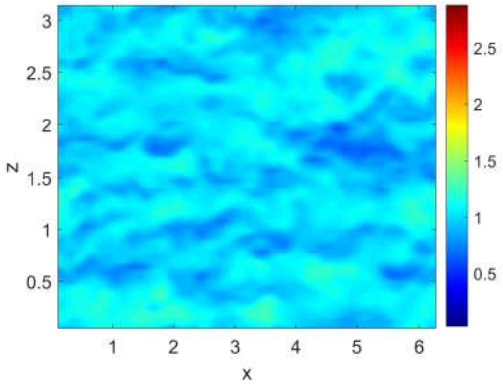
(c) L1



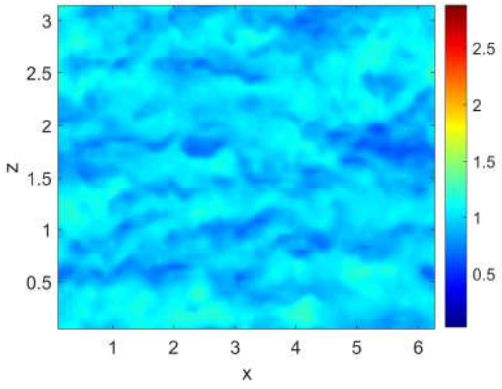
(d) l2



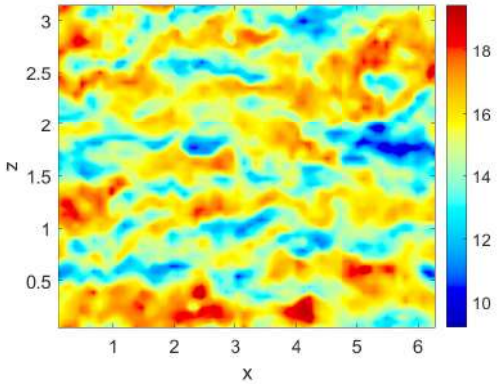
(e) L2



(f) shifted model

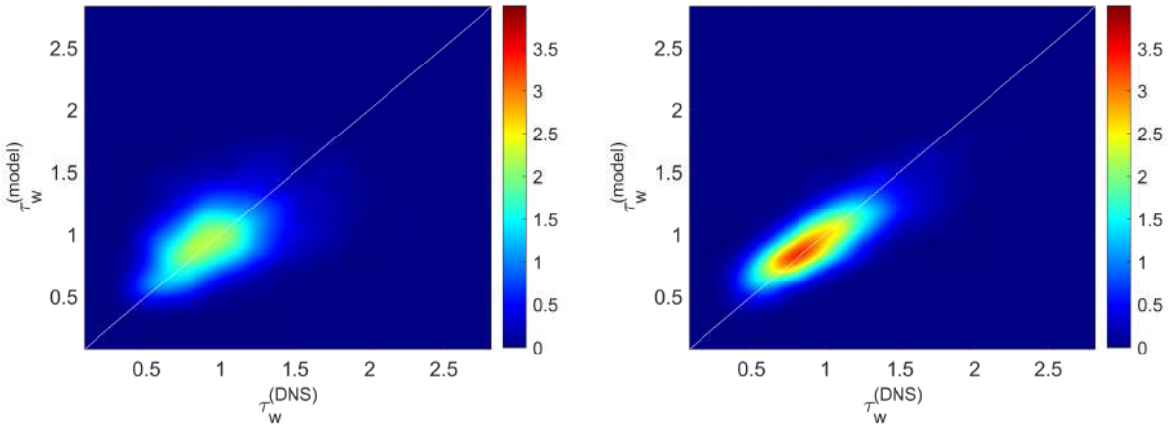


(g) ODE-based model

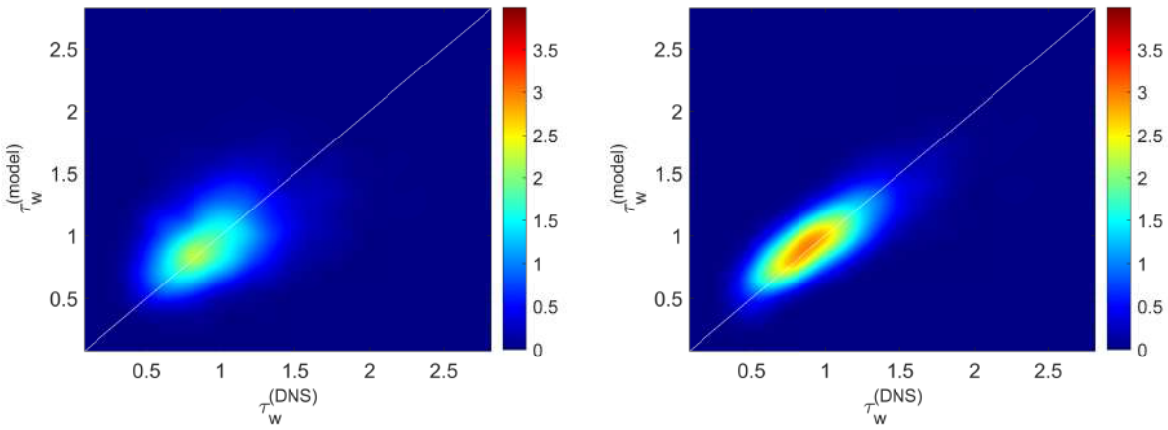


(h) u

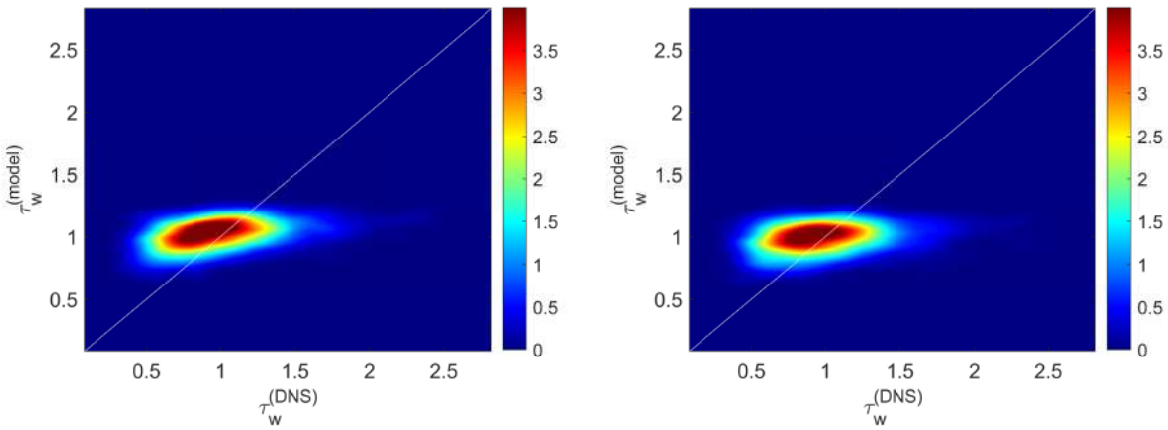
Figure 3.14. Spatial distribution of wall shear stress (τ_w) and streamwise velocity (u) for $Re_\tau = 400$ at $h^+ = 59.24$. fDNS in (a) refers to τ_w calculated using the filtered DNS data.



(a) l1 (b) L1



(c) l2 (d) L2



(e) shifted model (f) ODE-based model

Figure 3.15. Joint PDF of wall shear stress (τ_w) for $Re_\tau = 400$ at $h^+ = 59.24$.

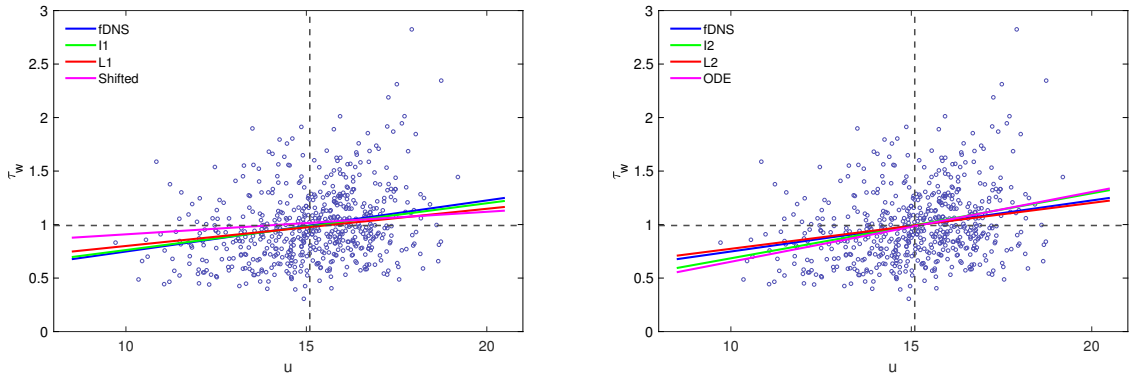


Figure 3.16. Wall shear stress as a function of streamwise velocity at $h^+ = 59.24$. Comparison between (a) filtered DNS, I1, L1, and shifted model, and (b) filtered DNS, I2, L2, and ODE-based model.

Table 3.9. Slope of the regression line (R) obtained from a plot with τ_w^{DNS} and τ_w^{model} as the horizontal and vertical axis, respectively; correlation coefficient (C.C.) between the model prediction and τ_w calculated using filtered DNS data, and root mean squared error (RMSE) for the wall models ($h^+ = 59.24$, $Re_\tau = 400$). Shifted and ODE represent the shifted and ODE-based wall models.

Case	R	C.C.	RMSE
I1	0.4000	0.4937	0.3116
L1	0.6343	0.8142	0.1964
I2	0.4084	0.4539	0.3355
L2	0.6771	0.8137	0.1958
shifted	0.1205	0.3694	0.3142
ODE	0.0788	0.2402	0.3283

to check the generalizability of the CNN wall models, they are tested for a different domain size and a higher Reynolds number case. In this section, the wall-normal distance (h^+) for the wall model inputs is chosen from a region which is included in or close to the training range of the CNN models in order to have a more meaningful comparison between the models because the CNN models have lower accuracy when h^+ is considerably outside of their training range as shown in Sec. 3.4.1.

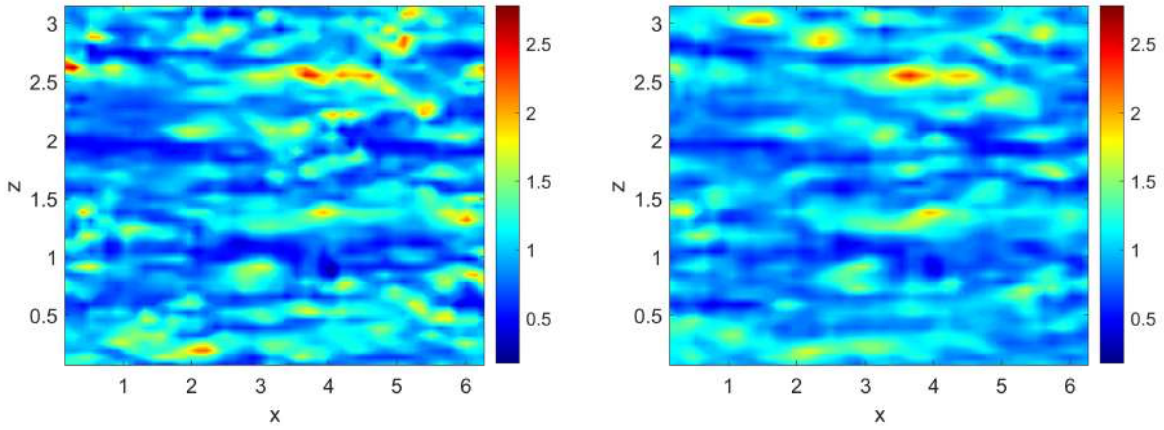
3.4.2.1 Different grid size

Unlike FCNN, in CNN the input data are two-dimensional. Hence, it is important to check how well the CNN performs if the size of the input data is changed. To assess the sensitivity of the model to the change in the input size, the CNN wall model trained using the training data

with the size of $N_x \times N_z = 64 \times 64$ and the same specifications as I2 in Table 3.4 is tested for the input size of $N_x \times N_z = 48 \times 48$, which corresponds to the filter sizes of $\overline{\Delta x}^+ = 49.1$ and $\overline{\Delta z}^+ = 24.5$. Fig. 3.17 shows the distribution of the wall shear stress when the wall model inputs are provided from the data at $h^+ = 42.06$ ($h/\delta = 0.105$), which is slightly larger than the upper limit of the training range for the inner layer-based wall models. As indicated in this figure, the model is still successful when the size of the CNN inputs is changed. Furthermore, the joint PDF shown in Fig. 3.18 confirms that the model still has reasonable accuracy and performs better than the existing models although it is tested for a different grid resolution. It also predicts the wall shear stress well on average as shown in Fig. 3.19. To check the performance of the CNN model further, the model I2 is tested for the grid size of $N_x \times N_z = 32 \times 32$ which is much coarser than the size of training data. As shown in Fig. 3.20, the CNN model is still able to have a reasonable prediction of the distribution of the wall shear stress, better than the existing models. Furthermore, in contrast to the other wall models, it is poorly correlated with the velocity field. However, it should be mentioned that in comparison to the finer grids, the CNN model is slightly less accurate for a very coarse grid in the prediction of the wall shear stress distribution which is expected due to the large difference between the grid size of the training data and this test case. In the study on the SGS modeling using CNN it has been shown as well that the CNN-based SGS model is less accurate when used for a grid size much coarser than that of used for the training data [78]. Fig. 3.21 shows the joint PDF of wall shear stress for different wall models for the grid size of $N_x \times N_z = 32 \times 32$. As shown in this figure, the joint PDF for the CNN wall model is more concentrated near the center compared to the existing wall models which shows the better accuracy of the CNN model. Similarly, as Fig. 3.22 shows, the regression line for the CNN model is closer to that of the filtered DNS compared to the Shifted and ODE-based models. Therefore, the CNN model is still able to have a reasonable prediction of the wall shear stress although it has been tested for a very coarse grid.

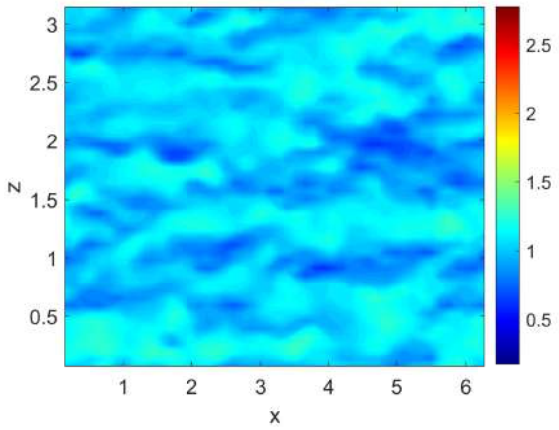
3.4.2.2 Higher Reynolds number flow

One of the important issues which needs to be checked for the data-driven turbulence models is their ability in making sufficiently accurate predictions when applied to a higher Reynolds number case. In fact, one of the ultimate goals for developing turbulence models (including wall models) is to use them for the very high Reynolds number flows whose the DNS data are not available. Therefore, it is crucial to check the performance of the models for a higher Reynolds

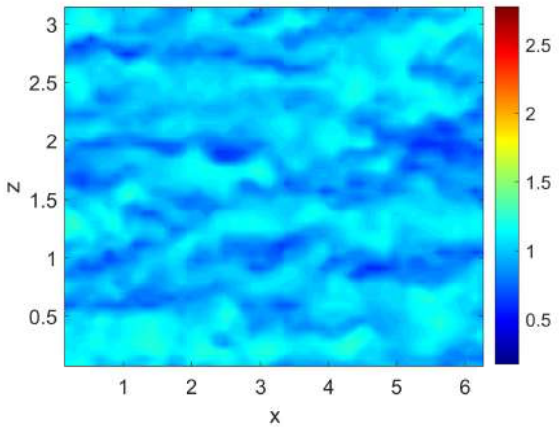


(a) fDNS

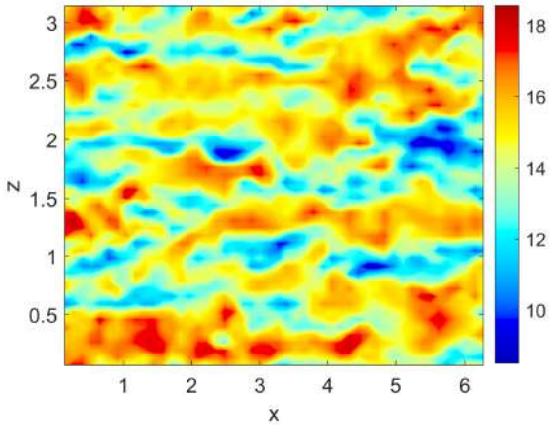
(b) CNN model



(c) shifted model

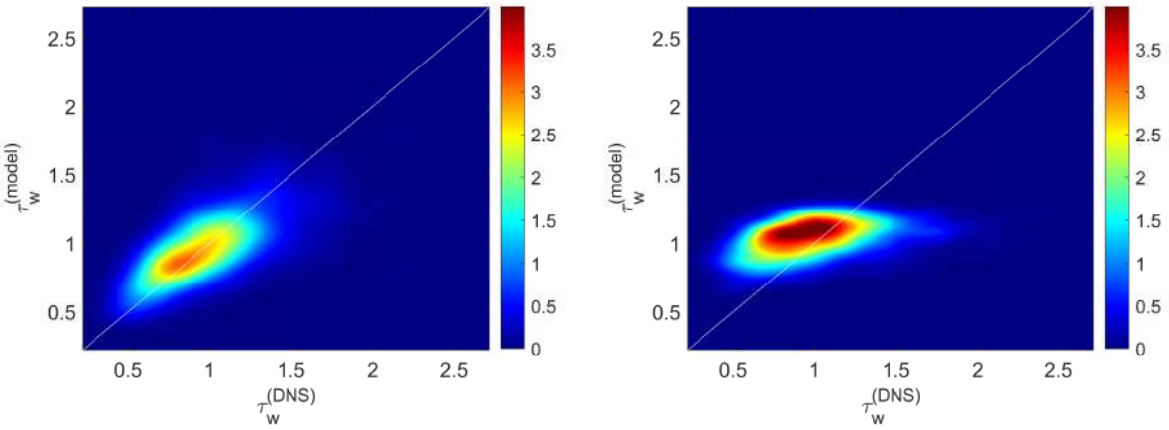


(d) ODE-based model



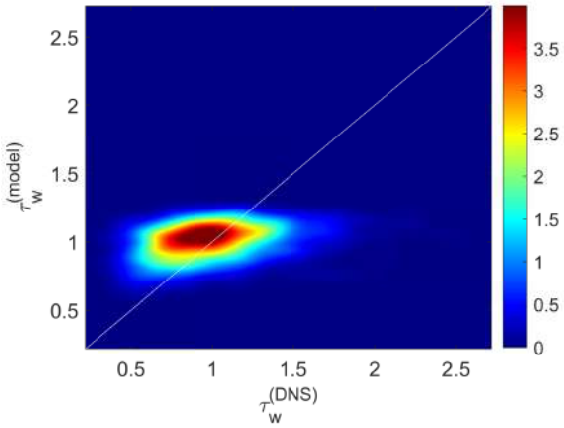
(e) u

Figure 3.17. Spatial distribution of wall shear stress (τ_w) and streamwise velocity (u) for $Re_\tau = 400$ at $h^+ = 42.06$ with the grid size of $N_x \times N_z = 48 \times 48$. fDNS in (a) refers to τ_w calculated using the filtered DNS data.



(a) CNN model

(b) shifted model



(c) ODE-based model

Figure 3.18. Joint PDF of wall shear stress (τ_w) for $Re_\tau = 400$ at $h^+ = 42.06$ with the grid size of $N_x \times N_z = 48 \times 48$.

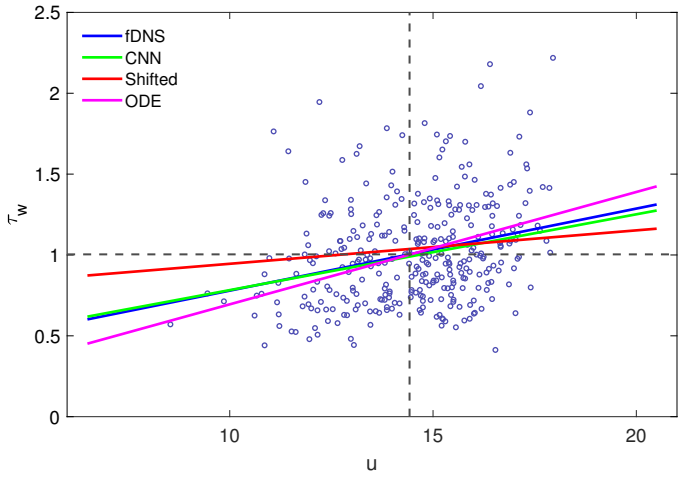
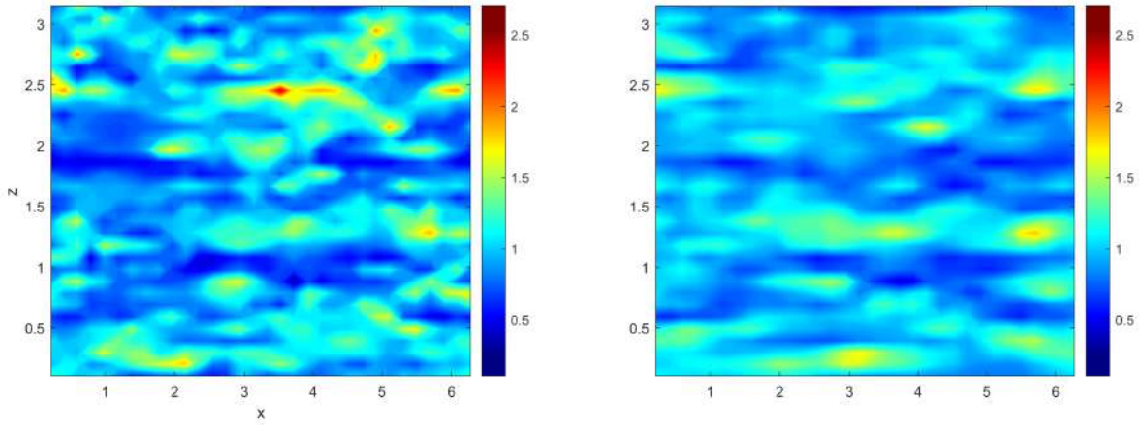
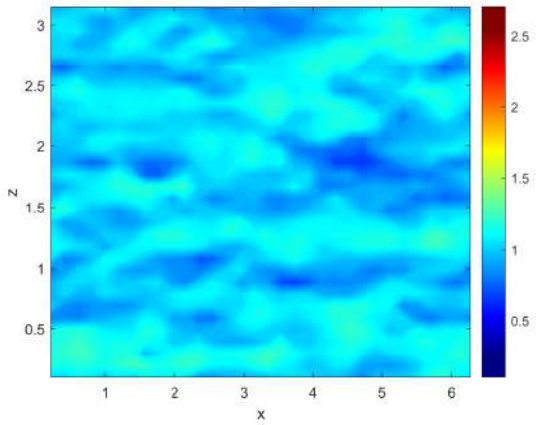


Figure 3.19. Wall shear stress as a function of streamwise velocity. $h^+ = 42.06$ and $N_x \times N_z = 48 \times 48$.

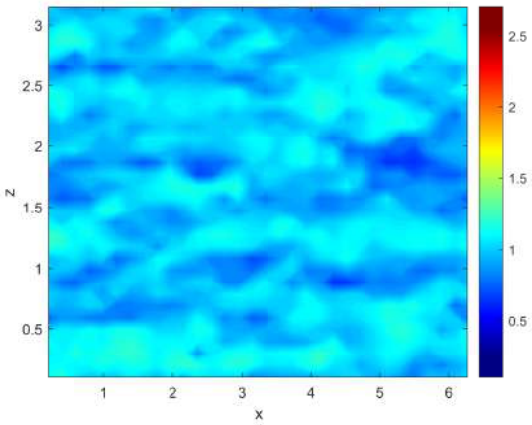


(a) fDNS

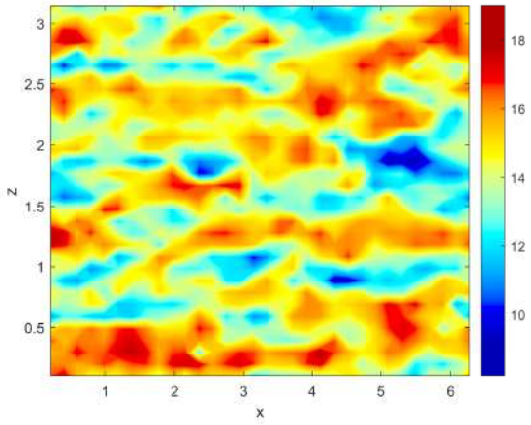
(b) CNN model



(c) shifted model

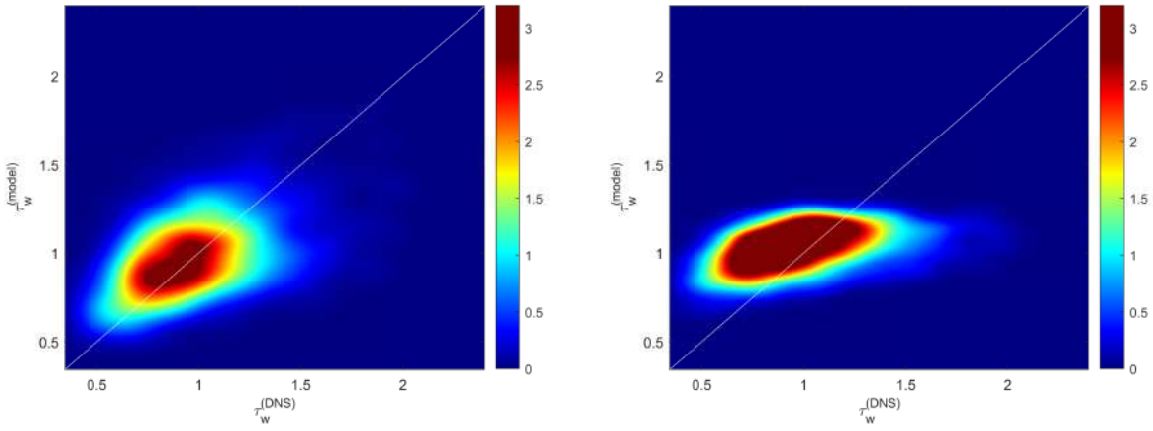


(d) ODE-based model



(e) u

Figure 3.20. Spatial distribution of wall shear stress (τ_w) and streamwise velocity (u) for $Re_\tau = 400$ at $h^+ = 42.06$ with the grid size of $N_x \times N_z = 32 \times 32$. fDNS in (a) refers to τ_w calculated using the filtered DNS data.



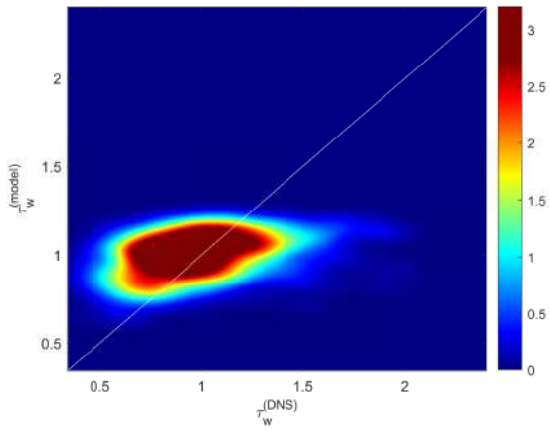


Figure 3.21. Joint PDF of wall shear stress (τ_w) for $Re_\tau = 400$ at $h^+ = 42.06$ with the grid size of $N_x \times N_z = 32 \times 32$.

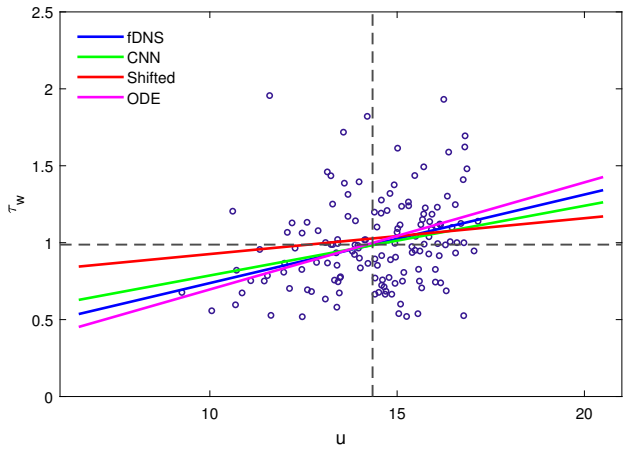


Figure 3.22. Wall shear stress as a function of streamwise velocity. $h^+ = 42.06$ and $N_x \times N_z = 32 \times 32$.

number flow. In this section, the CNN wall models are applied to a higher Reynolds number case ($Re_\tau = 600$), whose the DNS data are available, to investigate if their performance is good when applied to the data of an unseen Reynolds number. The details of the domain size are provided in Table 3.1. The filter size in the streamwise and spanwise directions is $\overline{\Delta x}^+ = 29.45$ and $\overline{\Delta z}^+ = 14.73$, respectively.

The distributions of the wall shear stress are shown in Fig. 3.23. As shown in this figure, the CNN wall models are poorly correlated with the velocity at the plane off the wall which provides the wall model inputs like the filtered DNS data and in contrast to the existing wall models. The CNN wall models predict the wall shear stress satisfactorily and perform better than the existing wall models. However, I1 is in slightly better agreement with the filtered DNS data compared to the other CNN models. The results for the joint PDF, shown in Fig. 3.24, also indicate the better accuracy of the CNN wall models compared to the shifted and ODE-based wall models. On the whole, among the CNN models, the PDF is more concentrated near the diagonal line for the inner layer-based models, especially I1, which together with the distribution of the wall shear stress (Fig. 3.23) confirms the better accuracy of these models. Fig. 3.25 shows the plot of the wall shear stress versus velocity for $Re_\tau = 600$ and the regression lines for different models. As shown in this figure, the CNN models are in better agreement with the wall shear stress obtained using the filtered DNS data than the existing models; however, for all CNN models the wall shear stress is underpredicted for the small values of velocity. It is expected to be different from the wall shear stress obtained using the filtered DNS data at rare values of velocity as shown in the study on wall modeling using FCNN [63]. Based on the discussion in Sec. 3.3.4, the wall shear stress is the function of not only velocity but also other quantities which are not included in the CNN wall model. Therefore, it is expected that the CNN model has discrepancies with the wall shear stress calculated using the filtered DNS data at some points due to insufficient physical information. In order to check the applicability of the CNN wall model further to the higher Reynolds number flow, one of the CNN wall models (I2) is tested for the $Re_\tau = 800$ which is considerably larger than the Reynolds number of the training data. It also should be noted that for both tested higher Reynolds number cases the filter size and domain size are different from those of the training data at least in one of the streamwise and spanwise directions which can be more challenging. As Fig. 3.26 shows, the CNN model has a good prediction of the wall shear stress distribution, and the model performance does not change compared to the $Re_\tau = 600$. The joint PDF of the wall shear stress which is presented in Fig. 3.27 further confirms the better

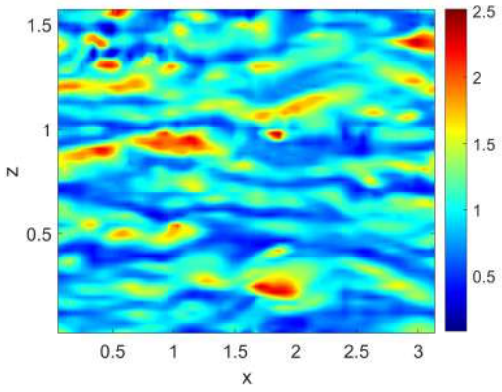
accuracy of the CNN wall model compared to the existing models. Additionally, as shown in Fig. 3.28, the regression line for the CNN model is very close to that of the filtered DNS data compared to the existing models. Therefore, based on the results, the CNN wall model has a good generalizability to the higher Reynolds number cases.

3.4.3 Local learning

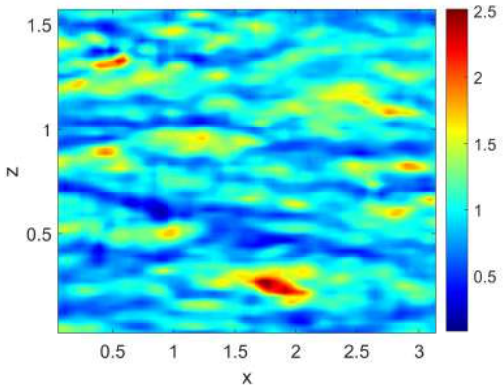
The goal of this section is to check the sensitivity of the CNN model to the domain size using local learning. The local learning approach is not only a technique for the data augmentation, but also a method for investigating the generalizability of the CNN model [113, 114]. In this section, the CNN model, I1, which is presented in Table 3.4, is retrained using the data of one quarter of the domain. The architecture and all the hyperparameters of CNN are the same as those of Sec. 3.4.1 except the size of CNN inputs and output which is 32×32. For testing the model, the inputs and output of the trained CNN wall model are resized to be used for the whole domain. Table 3.10 shows the correlation coefficients for the training and the validation datasets after retraining the model using local learning; they are high and comparable to those in Sec. 3.4.1. Comparison of these results with those of Sec. 3.4.1, where the training is performed over the whole domain, clearly shows that the obtained model performs similarly to that of Sec. 3.4.1, and it is almost insensitive to the domain size. It is still in good agreement with the wall shear stress resulted from the filtered DNS data as the distributions in Fig. 3.29 indicate. The joint PDF in Fig. 3.30 also further confirms the reasonable accuracy of the obtained model for different wall-normal distances. However, as shown in Fig. 3.30, for a closer wall-normal distance the performance is slightly better as also seen for I1 and I2 in Sec. 3.4.1. The results of the regression lines for the plot of wall shear stress versus velocity which is shown in Fig. 3.31 also shows that the obtained model is much more closer to the wall shear stress calculated using the filtered DNS data compared to the existing models. Additionally, it should be mentioned that similar to Sec. 3.4.2.2, the model is reasonably accurate when applied to a higher Reynolds number case and the results are not shown for brevity.

Table 3.10. Average correlation coefficient for training based on the data of $Re_\tau = 400$ for one quarter of domain.

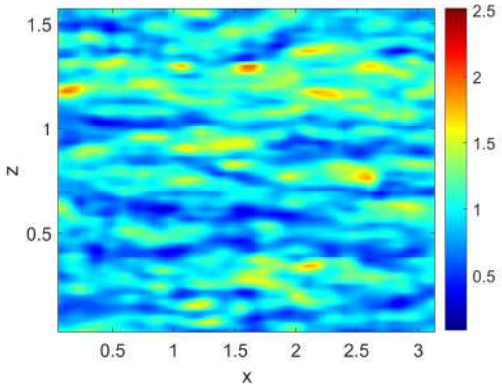
Case	Input	Input data range	$C.C._{tr}$	$C.C._{val}$
$Q1$	$u, \partial u/\partial y, uv, h$	$y^+ > 10, y/\delta < 0.1$	0.9088	0.8927



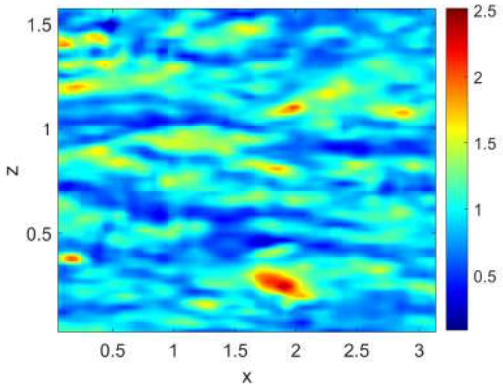
(a) fDNS



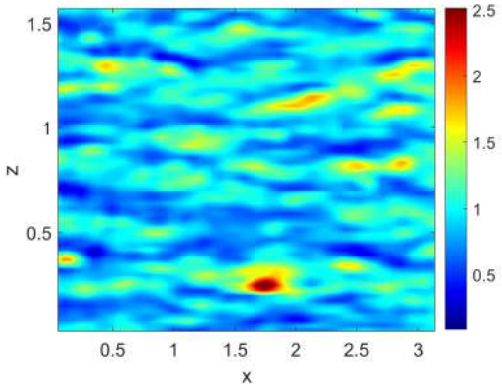
(b) I1



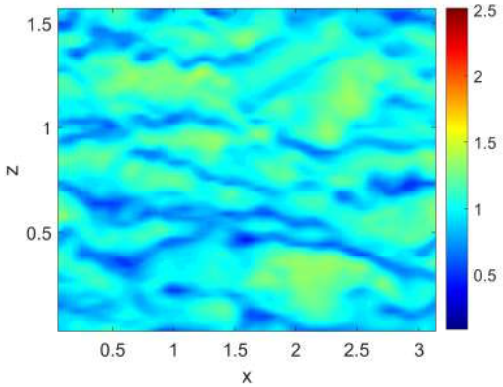
(c) L1



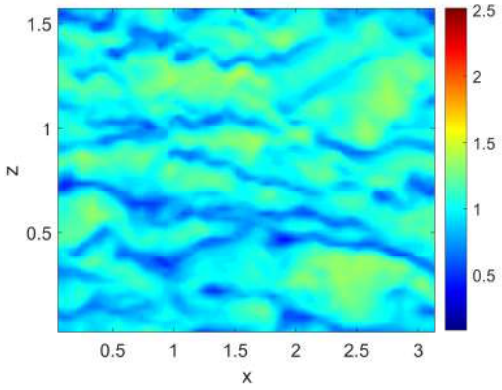
(d) I2



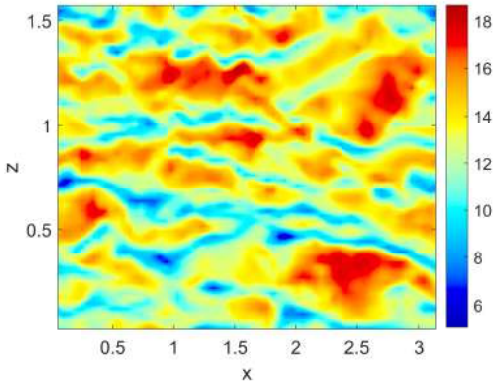
(e) L2



(f) shifted model

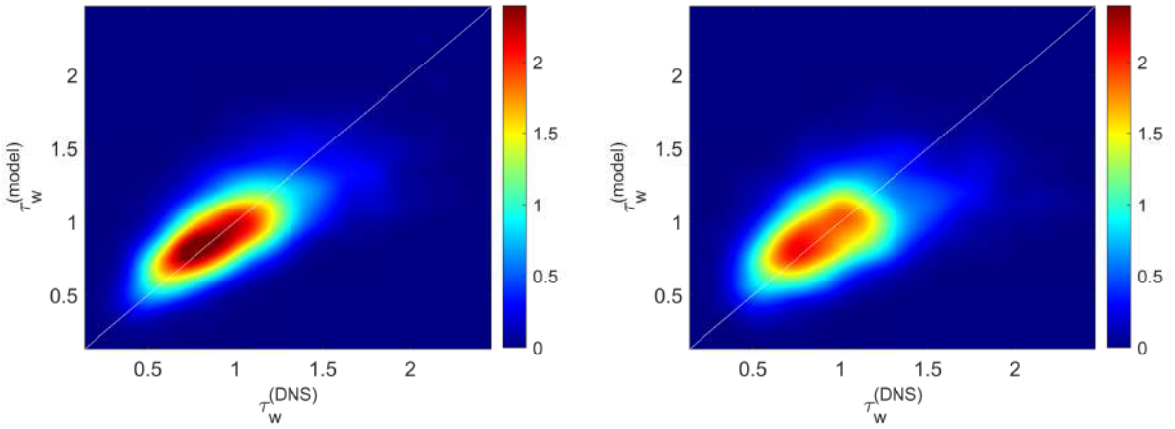


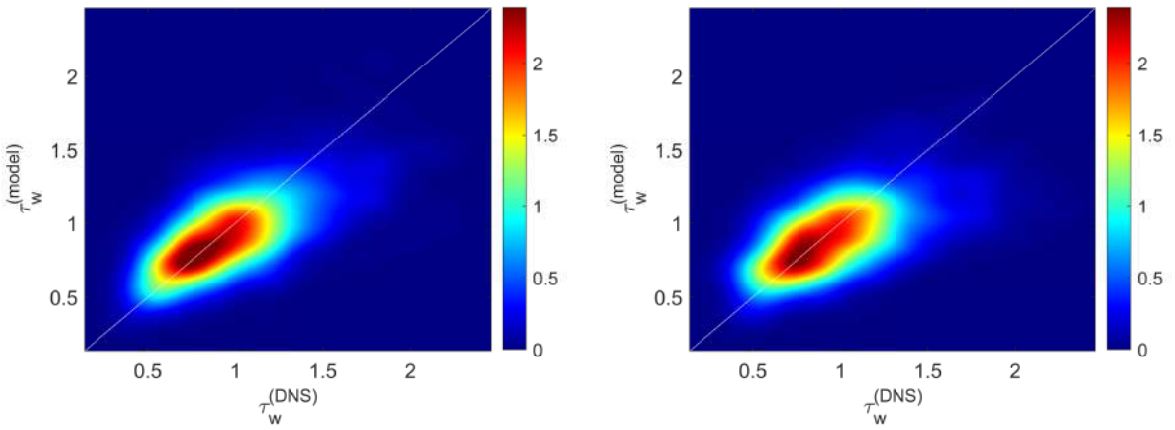
(g) ODE-based model



(h) u

Figure 3.23. Spatial distribution of wall shear stress (τ_w) and streamwise velocity (u) for $Re_\tau = 600$ at $h^+ = 26.89$. fDNS in (a) refers to τ_w calculated using the filtered DNS data.





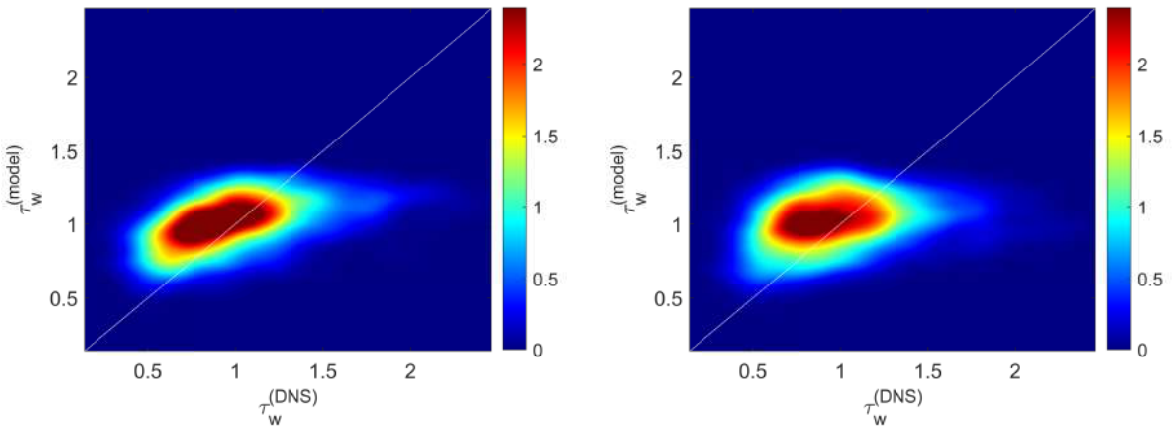


Figure 3.24. Joint PDF of wall shear stress (τ_w) for $Re_\tau = 600$ at $h^+ = 26.89$.

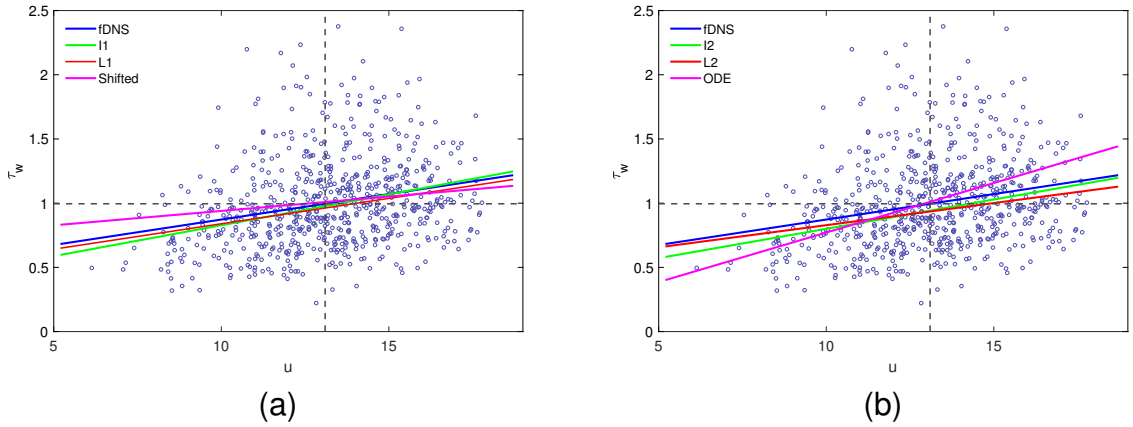


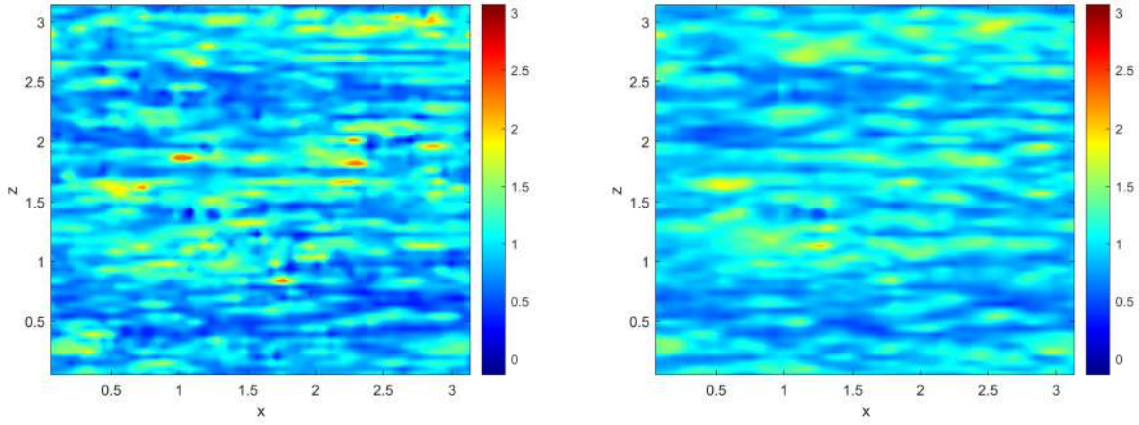
Figure 3.25. Wall shear stress as a function of streamwise velocity. $Re_\tau = 600$ and $h^+ = 26.89$. Comparison between (a) filtered DNS, I1, L1, and shifted model, and (b) filtered DNS, I2, L2, and ODE-based model.

3.4.4 Spanwise wall shear stress

As discussed in Sec. 3.2, the Neumann boundary condition ($\partial w/\partial y = 0$) may be imposed in the spanwise direction. Furthermore, as observed in the Eq. (3.12), in the wall-stress model, the wall shear stress can be assumed to be perfectly aligned with the wall-parallel velocity [21]. Therefore, based on this assumption, once the streamwise wall shear stress is obtained, the spanwise component can be calculated or if the magnitude of the wall shear stress is known, it can be decomposed to calculate the wall shear stress in the streamwise and spanwise directions as shown in Eq. (3.12). In the previous study on the data-driven wall modeling in LES [63], the magnitude of the wall shear stress is considered as the output of FCNN.

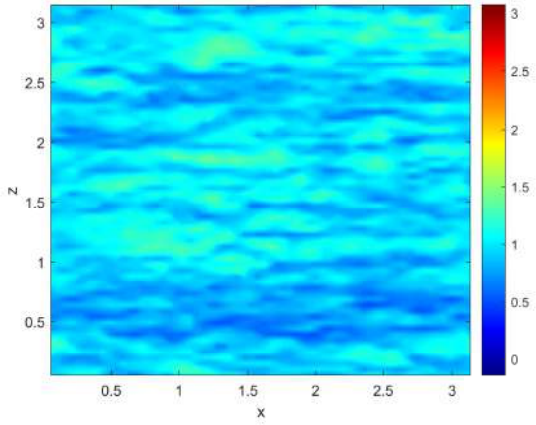
In this section, a CNN is trained to output the wall-shear stress both in the streamwise and spanwise directions using the same inputs and input data range as I2 in Table 3.4. The inputs of I2 include both the streamwise and spanwise velocity components; therefore, it is more likely to establish a successful wall model using these inputs compared to the other set of inputs used for I1 and L1. The results for the correlation coefficients are shown in Table 3.11. As shown in this table, for both components of the wall shear stress, high correlation coefficient is achieved for the training and validation datasets. Thus, choosing the velocity components as the CNN inputs can be a proper choice for predicting both the streamwise and the spanwise wall shear stress and leads to a high correlation coefficient.

Similar to Sec. 3.4.1, in this section the effect of choosing less number of inputs is investigated

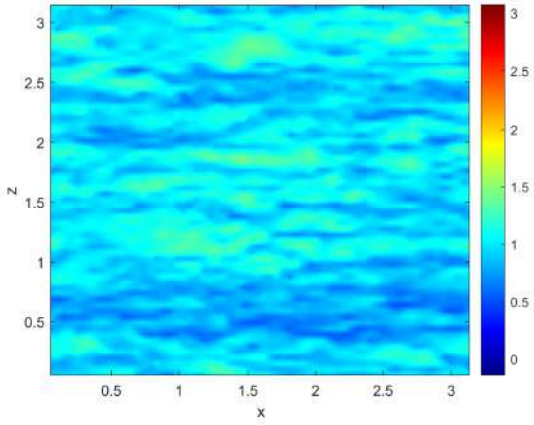


(a) fDNS

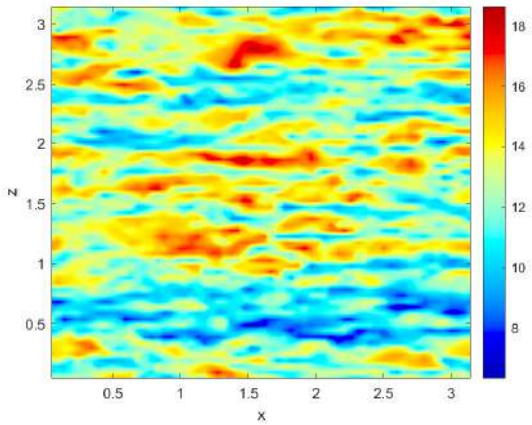
(b) CNN model



(c) shifted model

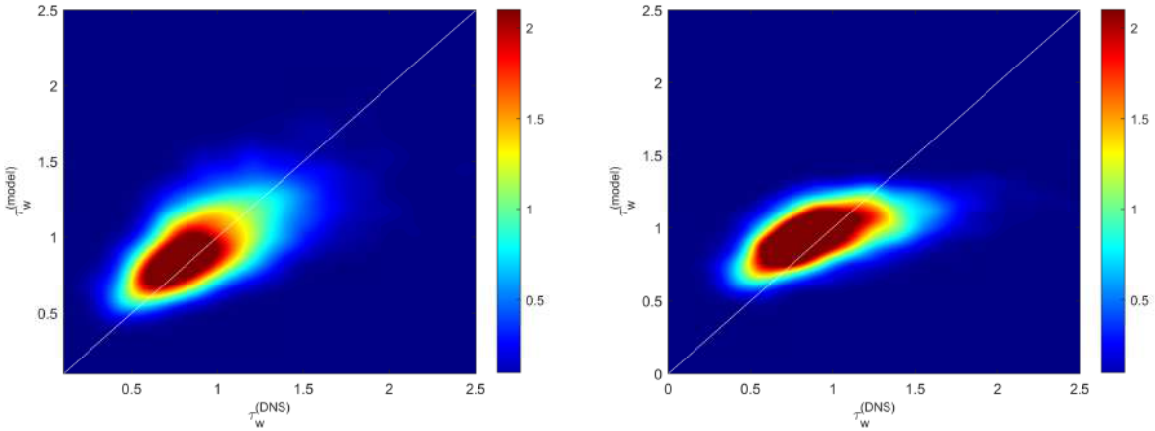


(d) ODE-based model

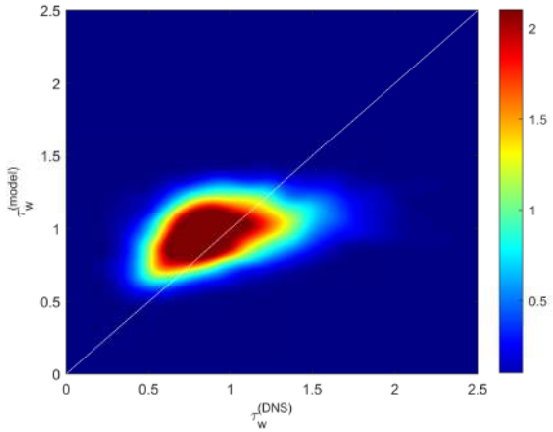


(e) u

Figure 3.26. Spatial distribution of wall shear stress (τ_w) and streamwise velocity (u) for $Re_\tau = 800$ at $h^+ = 25.26$. fDNS in (a) refers to τ_w calculated using the filtered DNS data.



(a) CNN model (b) shifted model



(c) ODE-based model

Figure 3.27. Joint PDF of wall shear stress (τ_w) for $Re_\tau = 800$ at $h^+ = 25.26$.

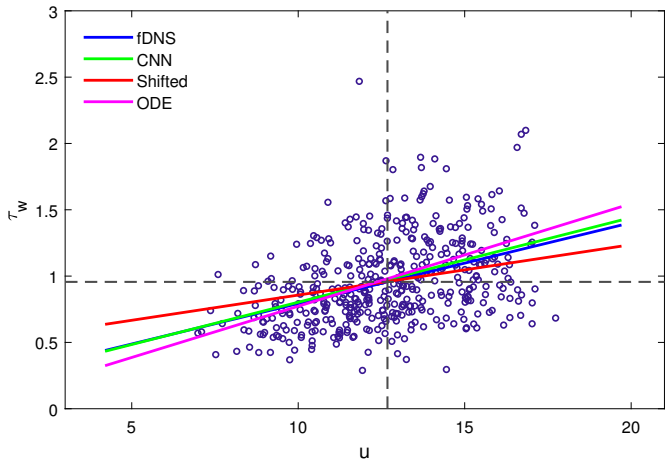


Figure 3.28. Wall shear stress as a function of streamwise velocity. $Re_\tau = 800$ and $h^+ = 25.26$.

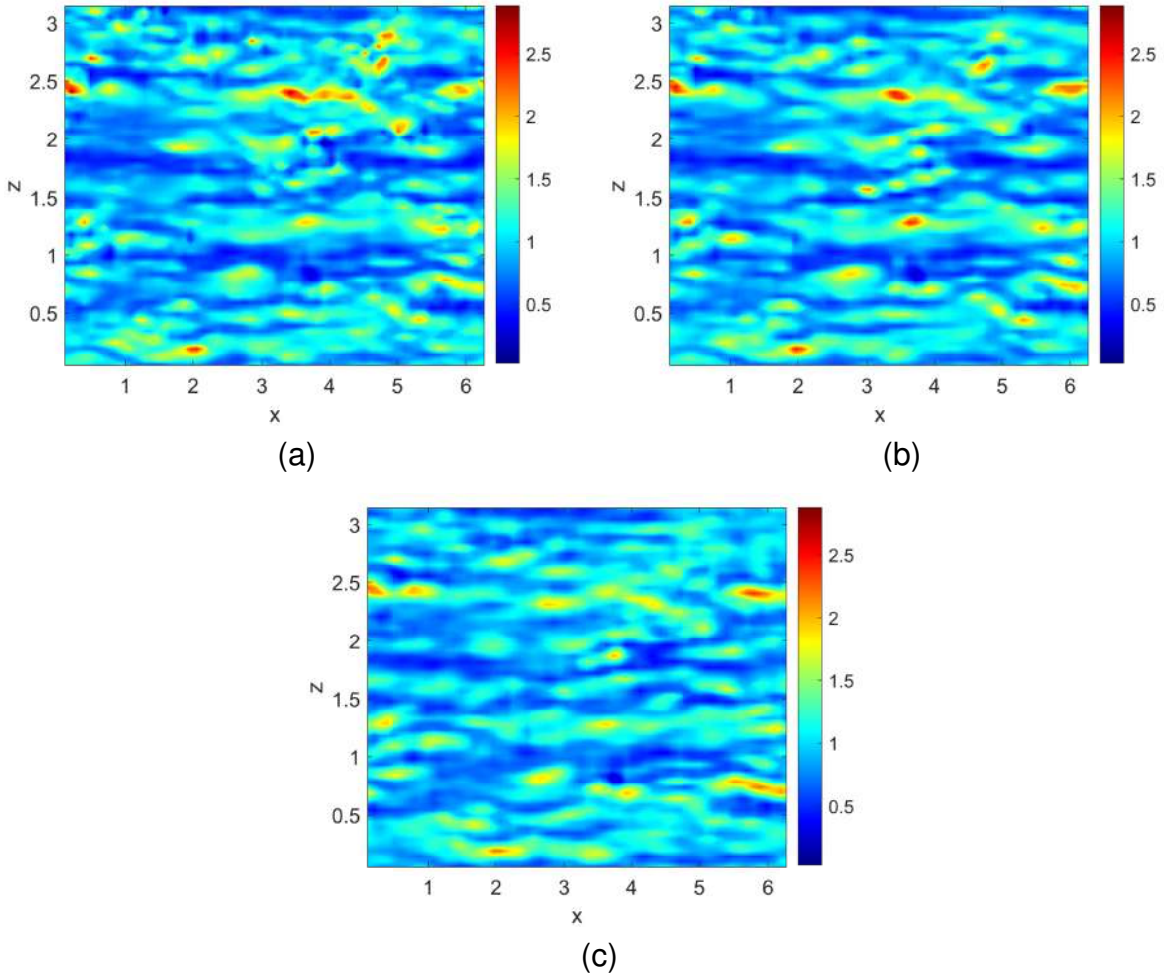


Figure 3.29. Spatial distribution of wall shear stress (τ_w) for Q1 at $Re_\tau = 400$. (a) Filtered DNS, (b) CNN model ($h^+ = 17.89$) and (c) CNN model ($h^+ = 39.45$).

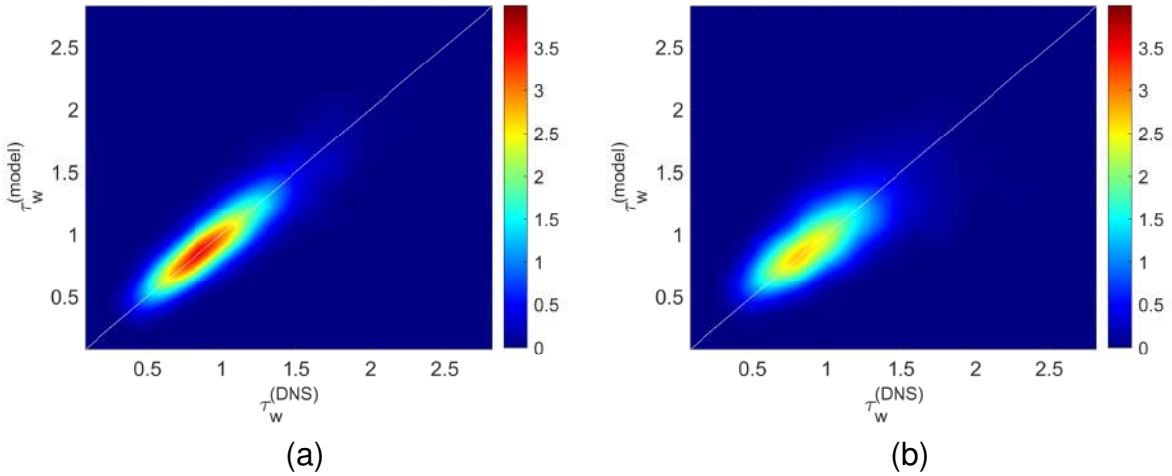


Figure 3.30. Joint PDF of wall shear stress (τ_w) for Q1 at $Re_\tau = 400$. CNN model at (a) $h^+ = 17.89$ and (b) $h^+ = 39.45$.

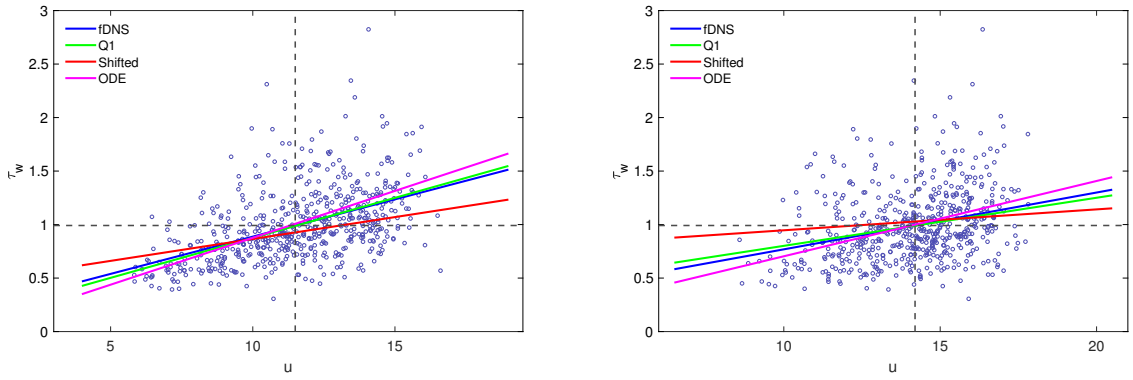
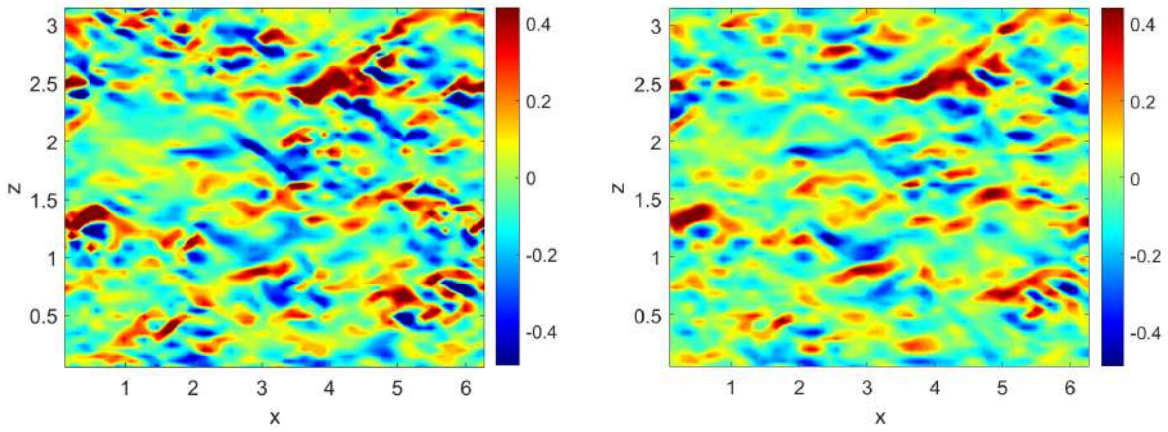


Figure 3.31. Wall shear stress as a function of streamwise velocity at (a) $h^+ = 17.89$ and (b) $h^+ = 39.45$.

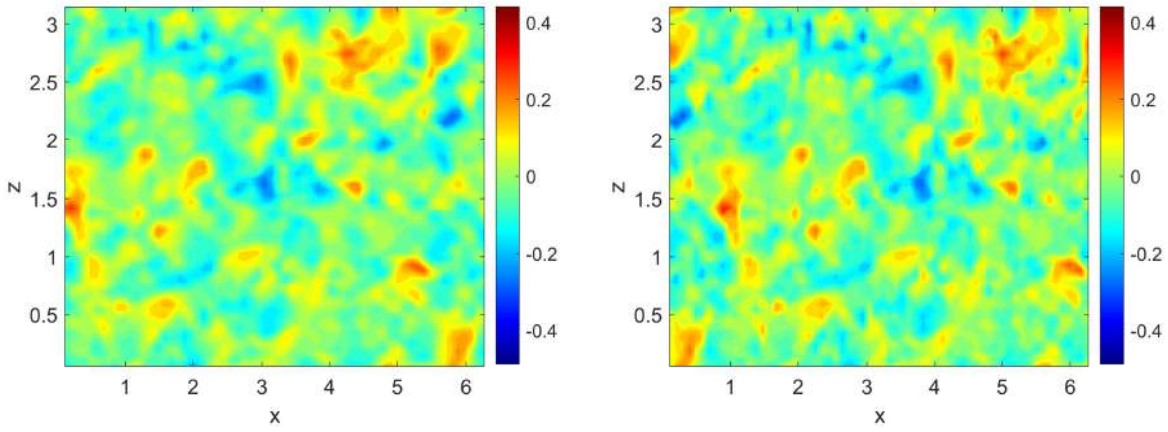
as well. Comparing the results of Tsble 3.11 and Table 3.12 shows that when the wall-normal velocity component is included in the input choices the correlation coefficient increases which is consistent with the results obtained in Sec. 3.4.1. Therefore, choosing all velocity components can be a reasonable choice.

The distributions of wall shear stress in the spanwise direction for $Re_\tau = 400$ and $Re_\tau = 600$ are presented in Fig. 3.32 and Fig. 3.34, respectively. The CNN wall model predicts the spanwise wall shear stress for the same Reynolds number as training and a higher Reynolds number with reasonable accuracy. The existing models underpredict and overpredict the spanwise wall shear stress in many parts of domain. Furthermore, as in the case of the streamwise component, the ODE-based wall model and the shifted wall model (if shifted by Δ_s) are perfectly correlated with the spanwise velocity, which is not true as shown in Mathis et al. [30]. However, the spanwise wall shear stress obtained by the CNN model is poorly correlated with the spanwise velocity at an off-wall plane. Like the results in Sec. 3.4.1, good agreement with the wall shear stress of the filtered DNS data is achieved for the streamwise component as well. Additionally, the joint PDFs which are presented in Fig. 3.33 and Fig. 3.35 for $Re_\tau = 400$ and $Re_\tau = 600$ further confirm that the ability of the wall model in predicting the spanwise wall shear stress is much better than that of the existing wall models. The results presented in Table 3.13 and Table 3.14 also show good agreement between the wall shear stress calculated using the filtered DNS data and that predicted by the CNN model compared to the existing models. Based on these results, the CNN wall model outperforms the existing models, and it has lower error and higher correlation coefficient than the shifted and ODE-based wall models.



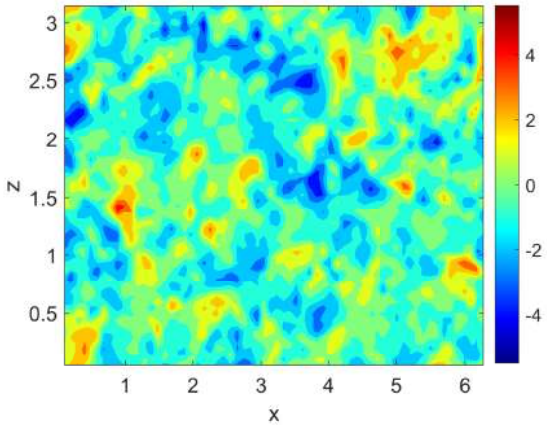
(a) fDNS

(b) CNN model



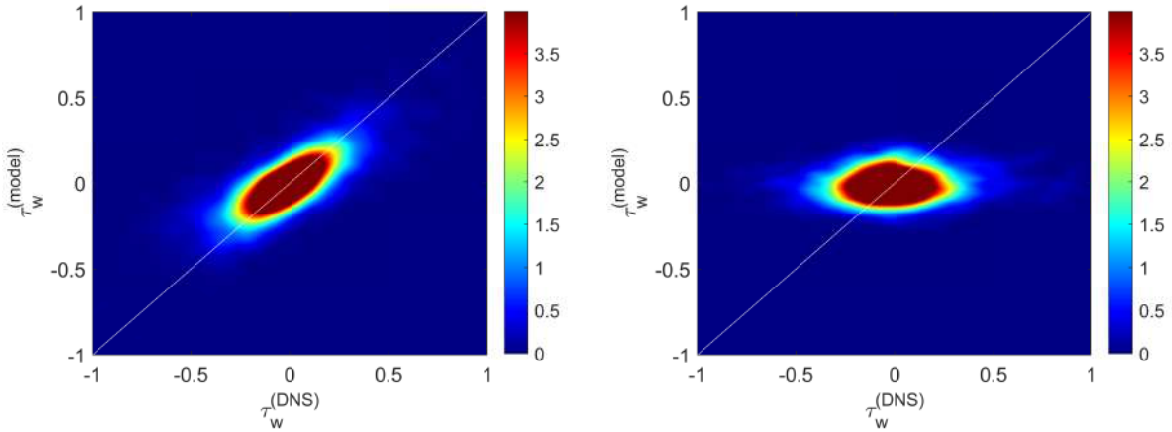
(c) shifted model

(d) ODE-based model



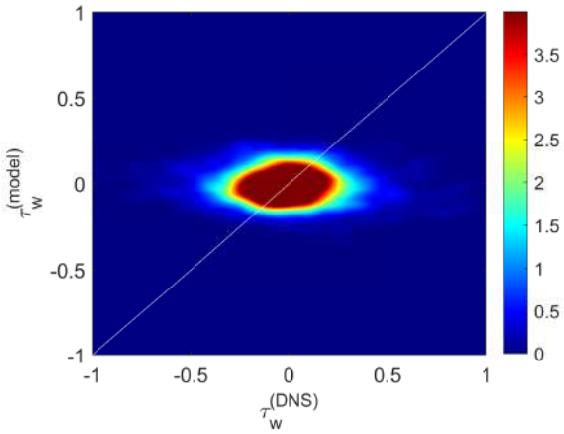
(e) w

Figure 3.32. Spatial distribution of wall shear stress (τ_w) in the spanwise direction and spanwise velocity (w) for $Re_\tau = 400$ at $h^+ = 42.06$. fDNS in (a) refers to τ_w calculated using the filtered DNS data.



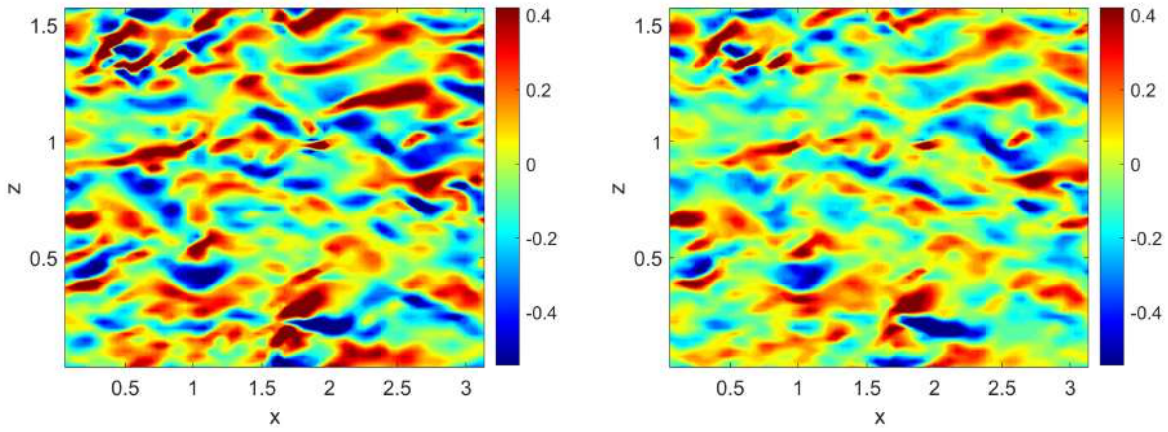
(a) CNN model

(b) shifted model



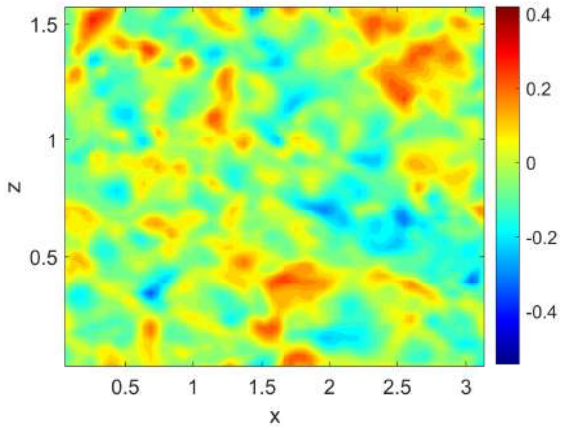
(c) ODE-based model

Figure 3.33. Joint PDF of wall shear stress (τ_w) in the spanwise direction for $Re_\tau = 400$ at $h^+ = 42.06$.

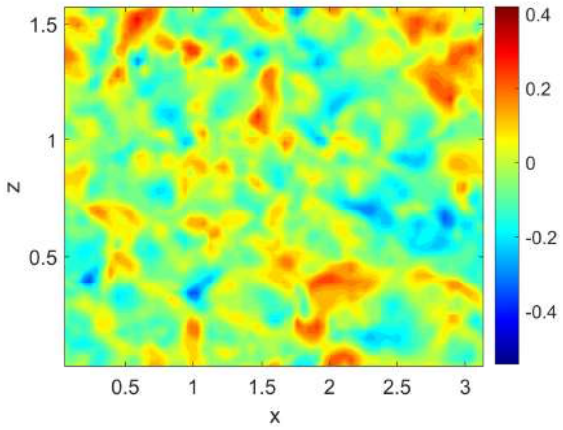


(a) fDNS

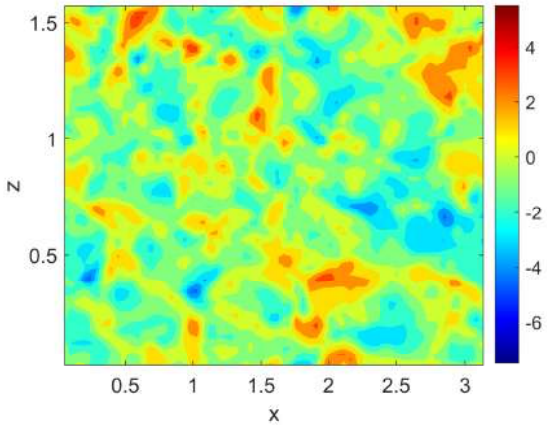
(b) CNN model



(c) shifted model

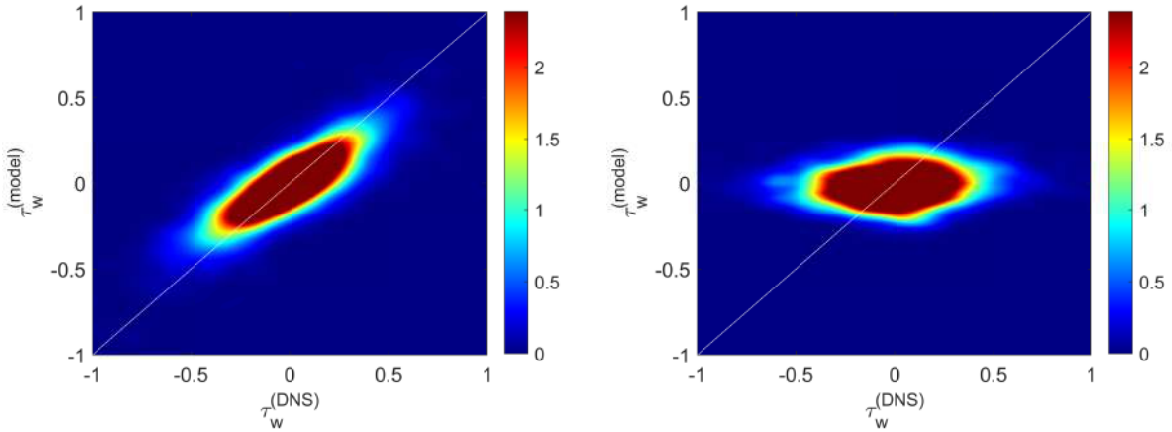


(d) ODE-based model



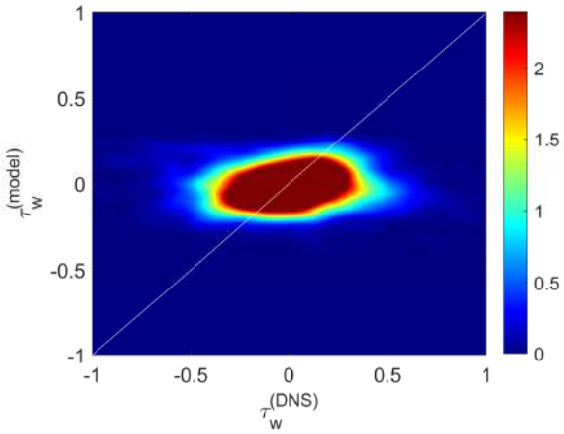
(e) w

Figure 3.34. Spatial distribution of wall shear stress (τ_w) in the spanwise direction and spanwise velocity (w) for $Re_\tau = 600$ at $h^+ = 26.89$. fDNS in (a) refers to τ_w calculated using the filtered DNS data.



(a) CNN model

(b) shifted model



(c) ODE-based model

Figure 3.35. Joint PDF of wall shear stress (τ_w) in the spanwise direction for $Re_\tau = 600$ at $h^+ = 26.89$.

Table 3.11. Correlation coefficient for streamwise (τ_{12}) and spanwise (τ_{32}) components of wall shear stress.

CNN output	$C.C._{tr}$	$C.C._{val}$
τ_{12}	0.9404	0.9305
τ_{32}	0.9071	0.8978

Table 3.12. Correlation coefficient for streamwise (τ_{12}) and spanwise (τ_{32}) components of wall shear stress for CNN trained using u and w as the input choices.

CNN output	$C.C._{tr}$	$C.C._{val}$
τ_{12}	0.9097	0.8948
τ_{32}	0.8430	0.8277

3.5 Conclusions

In the present study, *a priori* assessment has been carried out on the ability of the convolutional neural network in establishing a wall model for large eddy simulation. Initially, a parametric study has been performed on the CNN hyperparameters and the input choices, and two sets of inputs which give the highest correlation coefficients were selected. Due to the difference in the characteristics of the flow in the different regions near the wall, two CNN wall models have been trained using the data in the inner layer (excluding $y^+ \leq 10$) and the data in the logarithmic layer for each set of inputs. The obtained models have shown to perform well when the chosen wall-normal distance for the input data is in the training range. However, their prediction becomes less accurate when used for the wall normal distance which is considerably outside the training range. Among the CNN wall models, on the whole, the CNN wall models trained using the data in the inner layer (excluding $y^+ \leq 10$), especially I1, show better performance than the other CNN models.

In the next step, the performance of the CNN model has been tested under various conditions to check the generalizability of the model. The obtained model has been applied to a different grid size and a higher Reynolds numbers flow. The results show that the CNN model still gives reasonably accurate prediction when used for a flow different from that of the training data. The sensitivity of the CNN wall model to the domain size has been checked as well using local learning, and it has been shown that the CNN model has reasonable accuracy when applied to a

Table 3.13. Slope of the regression line (R) obtained from a plot with τ_w^{DNS} and τ_w^{model} as the horizontal and vertical axis, respectively; correlation coefficient (C.C.) between the model prediction and τ_w calculated using filtered DNS data, and root mean squared error (RMSE) for the wall models ($h^+ = 42.06$, $Re_\tau = 400$). Shifted and ODE represent the shifted and ODE-based wall models. All the results are for the wall shear stress in the spanwise direction.

Case	R	C.C.	RMSE
CNN	0.5962	0.7720	0.1390
shifted	0.0318	0.0853	0.2265
ODE	-0.0200	-0.0509	0.2387

Table 3.14. Slope of the regression line (R) obtained from a plot with τ_w^{DNS} and τ_w^{model} as the horizontal and vertical axis, respectively; correlation coefficient (C.C.) between the model prediction and τ_w calculated using filtered DNS data, and root mean squared error (RMSE) for the wall models ($h^+ = 26.89$, $Re_\tau = 600$). Shifted and ODE represent the shifted and ODE-based wall models. All the results are for the wall shear stress in the spanwise direction.

Case	R	C.C.	RMSE
CNN	0.6593	0.8431	0.1392
shifted	0.0371	0.1011	0.2650
ODE	0.0291	0.0755	0.2687

different domain size. The results were also compared with the shifted model and the ODE-based model; the CNN models have shown to provide better accuracy than the existing models.

In order to further investigate the performance of the CNN model, in the future, *a posteriori* test will be carried out by using the CNN wall model in an actual LES. Although the CNN-based wall model has shown to have a potential in establishing a wall-stress model in the *a priori* test and has been successful when tested under various conditions, *a posteriori* test is necessary to evaluate the model in an actual LES since there are some issues which cannot be addressed in the *a priori* test. To be precise, there are different factors which affect the performance of the WMLES such as the subgrid-scale model, the method of applying the wall model, and the computational grid. Therefore, *a posteriori* test is required as well to assess the sensitivity of the CNN wall model to the above factors and check the performance of the model in an actual simulation.

It should be noted that the present study is the first step toward the application of CNN for wall modeling in LES. Based on the results of the *a priori* assessment obtained in this study, CNN

has the ability to extract the flow features and establish a wall model which can predict the wall shear stress with a reasonable accuracy, better than the existing wall models, even with the simple choice of inputs like the primitive flow variables. The CNN model was shown to be successful when applied to a significantly coarser grid (larger filter size) and higher Reynolds number flow and different domain sizes. Among the tested conditions, it has shown more sensitivity to the very coarse grid size. It was expected from the CNN model as it is a nonlocal approach which receives the data in a two-dimensional form. However, even in those conditions it outperforms the existing models.

One approach to further increase the generalizability of the CNN model is to provide a larger training dataset which includes the data of different Reynolds numbers. In this case, for the same domain size and grid size, the non-dimensionalized filter size will be different for the datasets depending on their Reynolds numbers. In the study on the SGS modeling for Burgers turbulence using FCNN it has been shown that including the datasets obtained by applying different forcing functions to the Burgers turbulence can increase the generalizability of the SGS model [62]. Additionally, as pointed out in Sec. 3.4.3, using local learning can increase the size of training datasets and make the model more independent of the domain.

Although the CNN wall model is shown to be applicable to the channel flow with a reasonable accuracy, its application to the complex geometries and flow regimes can still be challenging. In fact, although the CNN model leverages using the neighboring data in the domain for establishing a model, its applicability to a more complex geometry is more difficult than the FCNN models. Therefore, some measures like transforming the data are required to make the CNN applicable to the complex geometries. In order to consider a wider range of flow regimes, the data from the flow under various pressure gradients can be considered in the training datasets. However, the data-driven wall modeling for the flows with more complex regimes can still be challenging and the previous attempt on the wall modeling for the flows over periodic hills was not successful in establishing an accurate and reliable wall model for such flows [65]. Therefore, it is crucial to continue the efforts on further improving the generalizability of the CNN wall models in the future work.

4. Data-driven wall modeling, *a posteriori* test

4.1 Introduction

In this chapter, *a posteriori* test is carried out for the obtained CNN wall model to check how well the model performs in an actual LES. First, a code written in C++ is developed for the wall-modeled LES of turbulent channel flow. Next, the CNN wall model obtained in Sec. 3.4.4 which predicts the wall shear stress in both the streamwise and spanwise directions is tested under various conditions including the same grid size and Reynolds number as used for the training, a coarser grid size and a higher Reynolds number. To obtain the data-driven wall model, CNN has been trained using the data in the inner layer (excluding $h^+ \leq 10$), and the CNN inputs are the velocity components $\{u, v, w\}$. The performance of the CNN model is compared with the ODE-based wall model as well. The developed CNN wall model is embedded in the code using a C++ library.

4.2 Numerical method

For the wall-modeled LES in the present study, the filtered Navier-Stokes equations are solved on a staggered grid. A schematic of the staggered grid is shown in Fig. 4.1. As shown in this figure, for the staggered grid the velocity components and pressure are defined at the cell faces and cell center, respectively. The spatial discretization is preformed using second-order central finite difference scheme and the grid is uniform in all directions.

In order to solve the governing equations which are the filtered Navier-Stokes equations, the fractional-step (projection) method is used which consists of two steps and couples the continuity equation and the pressure field for the incompressible flows [115,116]. If we consider the explicit Euler's method for the time advancement for the momentum conservation equation, it gives the following equation

$$\bar{u}^{n+1} = \bar{u}^n + \Delta t(A^n - \nabla P^n + B^n + C^n), \tag{4.1}$$

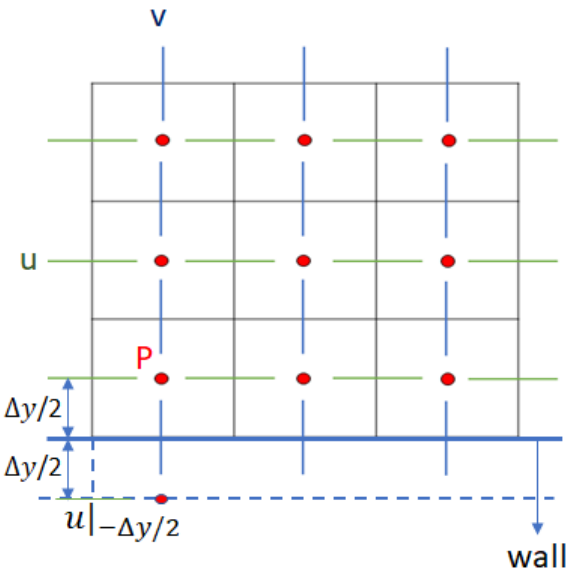


Figure 4.1. A schematic of staggered grid in 2D. The green and blue lines show the location of streamwise velocity (u) and wall-normal velocity (v), respectively, and the red circle shows the location of pressure.

where $\bar{P} = \bar{p}/\rho + \frac{1}{3}\tau_{ii}$ is the modified pressure which includes the isotropic SGS stress. A and B are defined as

$$A = -\nabla.(\bar{u}\bar{u}), \quad B = \nabla.(\nu[\nabla\bar{u} + (\nabla\bar{u})^T]), \tag{4.2}$$

and by considering the eddy viscosity SGS model, we will have the following formulation for the SGS stress term

$$C = \nabla.(2\nu_t\bar{S}). \tag{4.3}$$

In the present study, the static Smagorinsky model is used as the SGS model which is defined as

$$\tau_{ij} - \frac{\delta_{ij}}{3}\tau_{kk} = -2\nu_T\bar{S}_{ij} = -2C\bar{\Delta}^2|\bar{S}|\bar{S}_{ij}, \tag{4.4}$$

where $\bar{S}_{ij} = \frac{1}{2}(\frac{\bar{u}_i}{x_j} + \frac{\bar{u}_j}{x_i})$ is the strain rate and $|\bar{S}| = (2\bar{S}_{ij}\bar{S}_{ij})^{\frac{1}{2}}$ is the magnitude of the strain rate

tensor. The characteristic filter width in this study is set to

$$\bar{\Delta} = \sqrt[3]{\bar{\Delta}_x \bar{\Delta}_y \bar{\Delta}_z}. \tag{4.5}$$

The coefficient in Eq. 4.4 is set to $C = 0.01$ in the present study, which corresponds to the Samgorinsky constant of $C_s = 0.1$.

For the staggered grid, if we want to write the discretized form of the convection term for the momentum equations in the streamwise and wall-normal directions, by considering a two dimensional domain which is shown in Fig. 4.1 and the divergence form of the finite-difference approximation, we will have

$$\begin{aligned}[\frac{\partial(\bar{u}\bar{u})}{\partial x} + \frac{\partial(\bar{u}\bar{v})}{\partial y}]_{i+\frac{1}{2},j} &= \frac{1}{\Delta x}[(\frac{\bar{u}_{i+\frac{3}{2},j} + \bar{u}_{i+\frac{1}{2},j}}{2})^2 - (\frac{\bar{u}_{i+\frac{1}{2},j} + \bar{u}_{i-\frac{1}{2},j}}{2})^2] \\ &+ \frac{1}{\Delta y}[(\frac{\bar{u}_{i+\frac{1}{2},j+1} + \bar{u}_{i+\frac{1}{2},j}}{2})(\frac{\bar{v}_{i+1,j+\frac{1}{2}} + \bar{v}_{i,j+\frac{1}{2}}}{2}) - (\frac{\bar{u}_{i+\frac{1}{2},j} + \bar{u}_{i+\frac{1}{2},j-1}}{2})(\frac{\bar{v}_{i+1,j-\frac{1}{2}} + \bar{v}_{i,j-\frac{1}{2}}}{2})],\end{aligned} \tag{4.6}$$

for the momentum equation in the streamwise direction, and

$$\begin{aligned}[\frac{\partial(\bar{u}\bar{v})}{\partial x} + \frac{\partial(\bar{v}\bar{v})}{\partial y}]_{i,j+\frac{1}{2}} &= \frac{1}{\Delta x}[(\frac{\bar{u}_{i+\frac{1}{2},j+1} + \bar{u}_{i+\frac{1}{2},j}}{2})(\frac{\bar{v}_{i+1,j+\frac{1}{2}} + \bar{v}_{i,j+\frac{1}{2}}}{2}) - (\frac{\bar{u}_{i-\frac{1}{2},j+1} + \bar{u}_{i-\frac{1}{2},j}}{2}) \\ &(\frac{\bar{v}_{i,j+\frac{1}{2}} + \bar{v}_{i-1,j+\frac{1}{2}}}{2})] + \frac{1}{\Delta y}[(\frac{\bar{v}_{i,j+\frac{3}{2}} + \bar{v}_{i,j+\frac{1}{2}}}{2})^2 - (\frac{\bar{v}_{i,j+\frac{1}{2}} + \bar{v}_{i,j-\frac{1}{2}}}{2})^2],\end{aligned} \tag{4.7}$$

for the momentum equation in the wall-normal direction. In Eq. 4.2, for the constant viscosity $B = \nu\nabla^2 u$. When the fractional-step method is used, Eq. (4.1) is decomposed into two steps

$$\bar{u}^F = \bar{u}^n + \Delta t(A^n + B^n + C^n), \tag{4.8}$$

$$\bar{u}^{n+1} = \bar{u}^F - \Delta t \nabla \bar{P}^{n+1}. \tag{4.9}$$

By substituting Eq. (4.9) into continuity equation and forcing the continuity equation to be satisfied in the next time step

$$\nabla . \bar{u}^{n+1} = 0, \tag{4.10}$$

the Poisson equation for the pressure will be obtained

$$\nabla^2 \bar{P}^{n+1} = \frac{1}{\Delta t} \nabla . \bar{u}^F. \tag{4.11}$$

By considering the two-dimensional domain in Fig. 4.1, the continuity equation will have the following form on the staggered grid using the second order central finite difference method for the spatial discretization

$$\frac{\bar{u}^F_{i+\frac{1}{2},j} - \bar{u}^F_{i-\frac{1}{2},j}}{\Delta x} + \frac{\bar{v}^F_{i,j+\frac{1}{2}} - \bar{v}^F_{i,j-\frac{1}{2}}}{\Delta y} = 0. \tag{4.12}$$

The Poisson equation for the pressure is solved using the Fast Fourier Transform (FFT) algorithm by considering periodicity in the streamwise and spanwise directions. Performing Fourier transform in the periodic directions (x, z) gives

$$\frac{\tilde{\phi}_{k_x,k_z,j-1}}{\Delta y^2} - [\frac{2(1 - cosk_x\Delta_x)}{\Delta x^2} + \frac{2(1 - cosk_z\Delta_z)}{\Delta z^2} + \frac{2}{\Delta y^2}]\tilde{\phi}_{k_x,k_z,j} + \frac{\tilde{\phi}_{k_x,k_z,j+1}}{\Delta y^2} = \tilde{\psi}(k_x, k_z, j), \tag{4.13}$$

which can be solved in the wall-normal direction using Thomas algorithm. In Eq. (4.13), the Neumann boundary condition is used for the top and bottom boundaries. However, when $k_x = k_z = 0$, the Dirichlet boundary condition is used for the bottom boundary.

In the present study, the second-order Adams Bashforth method is used for the time advancement for the predictor step and the Euler method is used for the final step. Therefore, Eq. (4.8) will be modified to

$$\bar{u}^F = \bar{u}^n + \Delta t\Big[\frac{3}{2}(A^n + B^n) - \frac{1}{2}(A^{n-1} + B^{n-1} + C^{n-1})\Big]. \tag{4.14}$$

As the solution is prone to become unstable due to the aliasing error and using second order central finite difference scheme, in order to make the solution stable, the flow variables are filtered using compact scheme at the regular intervals. For the first three points close to the the wall, the 6^{th} order compact filter scheme [117] is implemented and for the other points, the 8^{th} order compact filter scheme is used. The formulation for the boundary points is

$$\alpha_f\hat{\phi}_{i-1} + \hat{\phi}_i + \alpha_f\hat{\phi}_{i+1} = \sum_{n=1}^{7} a_{n,i}\phi_n, \tag{4.15}$$

and for the other points

$$\alpha_f\hat{\phi}_{i-1} + \hat{\phi}_i + \alpha_f\hat{\phi}_{i+1} = \sum_{n=0}^{4} \frac{a_n}{2}(\phi_{i+n} + \phi_{i-n}), \tag{4.16}$$

where $\alpha_f = 0.495$.

The wall shear stress predicted by the wall model is applied as the Neumann boundary condition at the wall. As shown in Fig. 4.1, for the staggered grid, the wall boundary conditions for the streamwise and spanwise velocity components can be applied by adding fictitious cells to the physical domain. The Neumann boundary condition is defined as

$$\frac{\partial \bar{u}}{\partial y}|_w = \frac{\tau_{xy,w}}{\nu}, \tag{4.17}$$

in the streamwise direction and

$$\frac{\partial \bar{w}}{\partial y}|_w = \frac{\tau_{zy,w}}{\nu}, \tag{4.18}$$

in the spanwise direction, where the wall shear stress is provided by the wall model. For the streamwise velocity we will have

$$\bar{u}|_{-\Delta\ \mathrm{y}/2} = \bar{u}|_{\Delta\ \mathrm{y}/2} - \frac{\tau_{xy,w}\Delta\ \mathrm{y}}{\nu}, \tag{4.19}$$

and similarly, the spanwise velocity at the boundary will be obtained as follows

$$\bar{w}|_{-\Delta\ \mathrm{y}/2} = \bar{w}|_{\Delta\ \mathrm{y}/2} - \frac{\tau_{zy,w}\Delta\ \mathrm{y}}{\nu}. \tag{4.20}$$

For the wall-normal velocity, the no slip boundary condition ($\bar{v} = 0$) is implemented at the wall.

4.3 *A posteriori* test for the CNN wall model

In this section, the CNN model is embedded in an actual LES under different conditions, and the results are compared with the ODE-based wall model. In the CNN wall model, the input data are normalized by the friction velocity before feeding to the CNN. First, the wall-modeled LES will be conducted for the same Reynolds number as used for the training ($Re_\tau = 400$). Next, the applicability of the model to a higher Reynolds number case will be checked ($Re_\tau = 1000$). After that, the sensitivity of the CNN wall model to the grid size, matching location, and normalizing by the friction velocity will be investigated.

4.3.1 Wall modeled LES for $Re_\tau = 400$

In this section, the wall-modeled LES is performed for $Re_\tau = 400$. The domain size in this simulation is $L_x \times L_y \times L_z = 2\pi \times 2 \times \pi$. The grid size is $N_x \times N_y \times N_z = 64 \times 64 \times 64$ and the times step is 0.01. The data at $y/\delta = 0.078$ are used for the calculation of the friction velocity and the wall model inputs. The chosen wall-normal distance is included in the training range of

the CNN wall model. The results for the mean velocity profile, the streamwise root mean square (RMS) velocity, and the Reynolds shear stress are presented in Fig. 4.2. As seen in this figure, the CNN wall model can predict the mean velocity profile well. The results for the streamwise RMS velocity fluctuation are also comparable to that predicted using the ODE-based wall model. For the Reynolds shear stress, the ODE-based wall model tends more to overpredict the values. However, the CNN wall model tends more to underpredict the Reynolds shear stress. In order to check the performance of the CNN model in predicting the wall shear stress in the *a posteriori* test, the PDF for the wall shear stress is presented in Fig. 4.3. As seen in this figure, the PDF calculated using the filtered DNS data is wider compared to the wall-modeled LES performed using the CNN and ODE-based wall models, and the peak of PDF in the wall modeled LES is overpredicted. However, the PDF predicted by the simulation using CNN wall model is closer to the filtered DNS data if we compare the range and the shape of the PDF. The overprediction of the peak of PDF for the CNN wall model is also less than that of the ODE-based wall model.

The distribution of the streamwise velocity at the matching location and the distribution of the wall shear stress obtained from the simulations using the CNN and ODE-based wall models are shown in Fig. 4.4. As seen in these figures, the results of the *a posteriori* test are consistent with the results of the *a priori* test. In the deterministic wall stress models like the ODE-based wall model, the wall shear stress is perfectly correlated with the velocity at the matching location. It is in contrast to the correlation coefficient observed in the DNS [30], which shows that the velocity at the sampling point is poorly correlated with the wall shear stress. On the other hand, as shown in Fig. 4.4, the results of CNN wall model lead to a relatively low correlation coefficient between the wall shear stress and the velocity at the sampling point ($C.C. = 0.4676$), which is in agreement with the results from the DNS data presented in Ref. [30]. The scatter plot of the wall shear stress vs. streamwise velocity is shown in Fig. 4.5. As seen in this figure, the results for the CNN wall model and the ODE-based wall model are in agreement with those of the *a priori* test. For the CNN wall model, the plot is scattered as the wall shear stress is not only the function of streamwise velocity and it looks similar to that of the filtered DNS seen in the *a priori* test. However, for the ODE-based wall model the predicted wall shear stress is concentrated near the regression line.

The vortical structures obtained from the instantaneous flow field are presented in Fig. 4.6. In order to visualize the vortical structures, the second invariant of the deformation tensor [118]

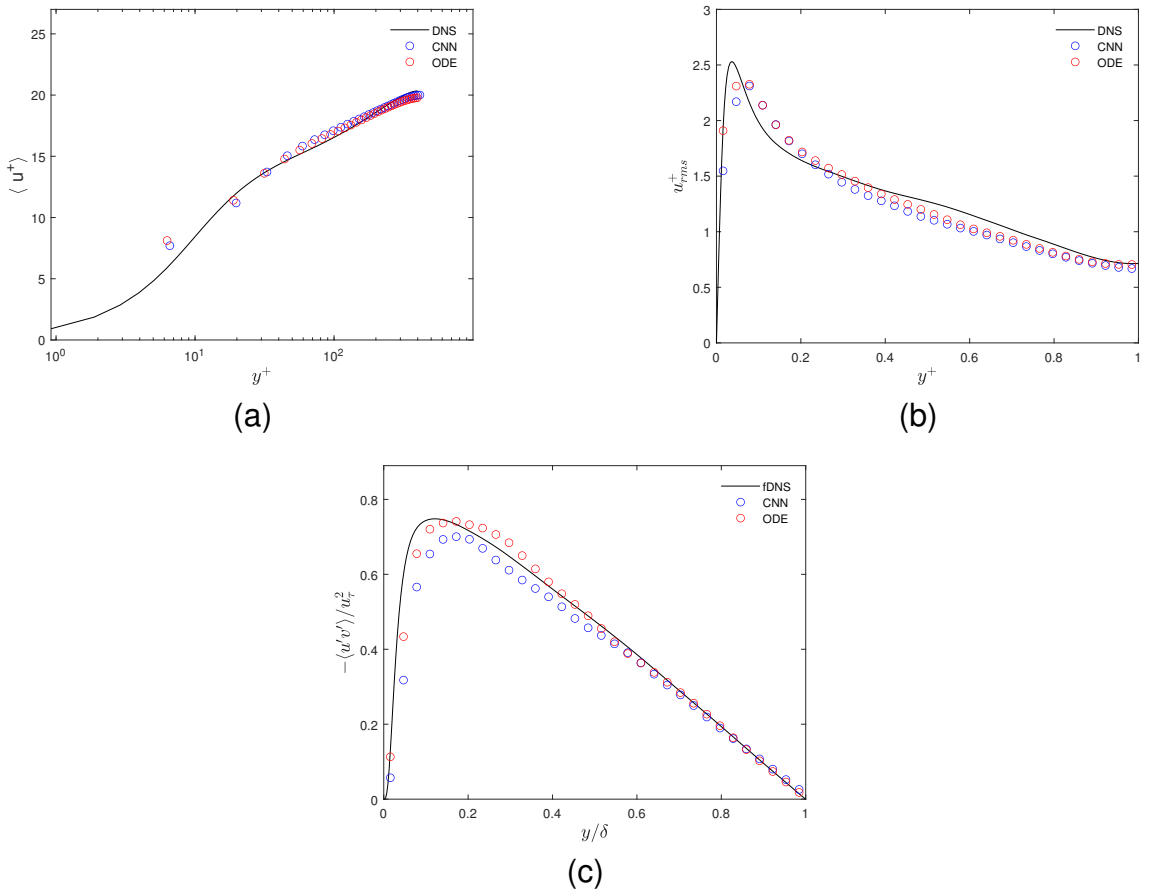
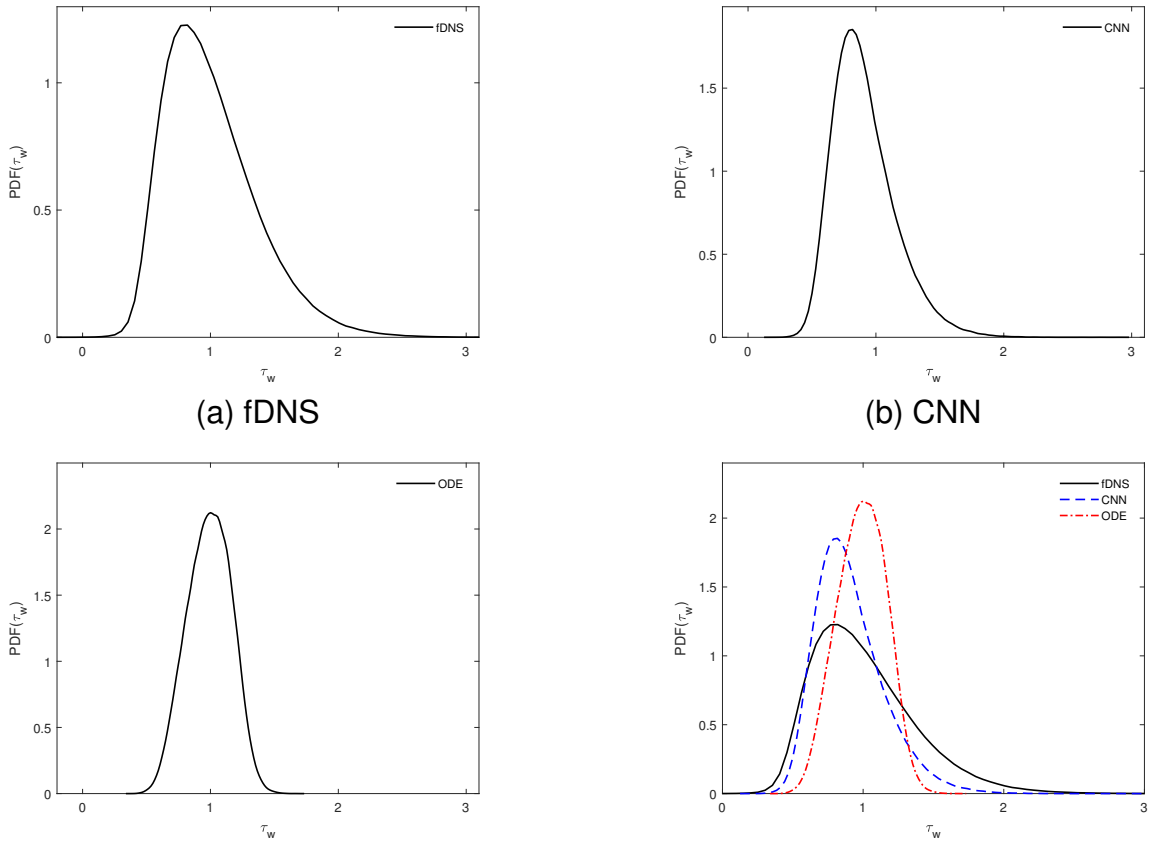


Figure 4.2. Mean velocity profile (a), streamwise RMS velocity fluctuation (b), and Reynolds shear stress (c) for $Re_\tau = 400$.

is calculated, which is defined as

$$Q = \frac{1}{2}(||\Omega||^2 - ||S||^2), \tag{4.21}$$

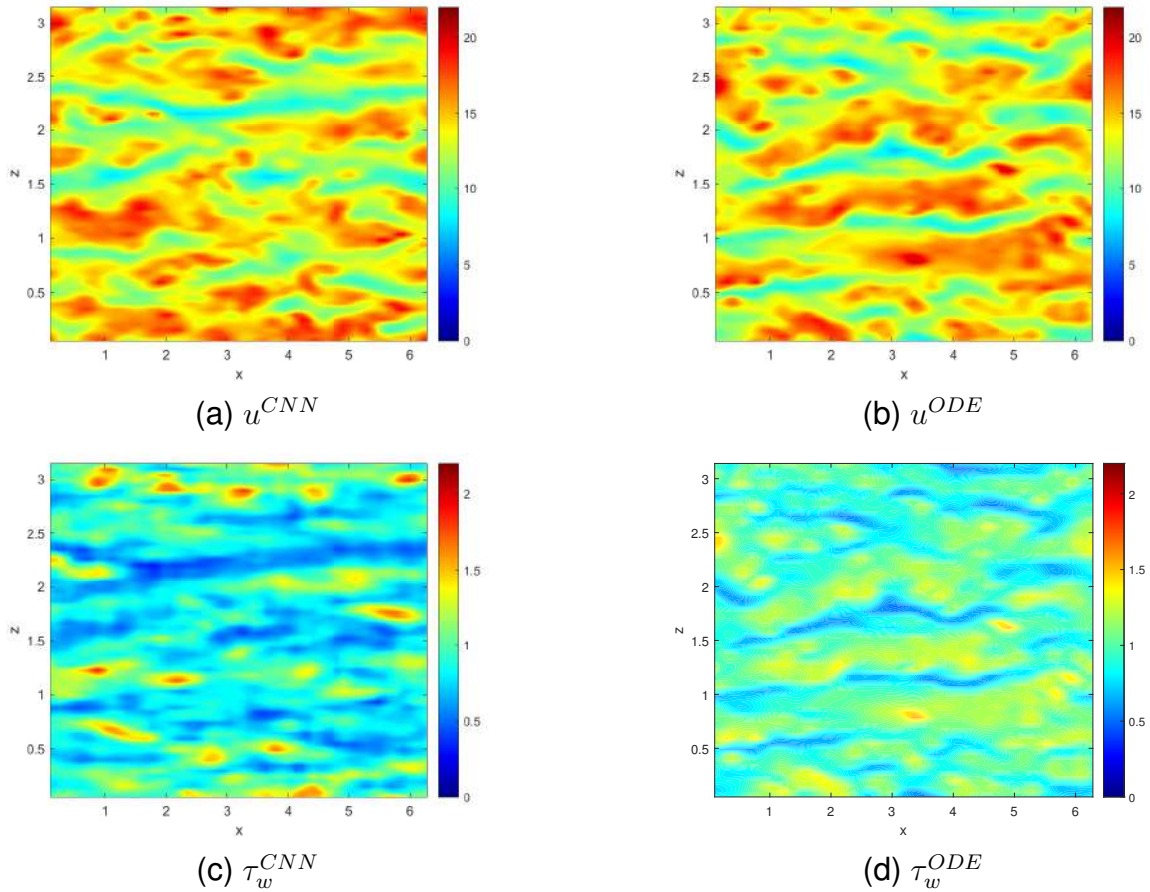
where Ω is the rotation rate tensor and S is the strain rate tensor. Q-criterion has already been used in various studies as a measure to identify the vortical structures [55, 118–120]. The positive values of Q show the areas in the flow field where the vorticity dominates and the negative values of Q indicate the regions where the strain rate dominates. As seen in Fig. 4.6, it is expected that the vortical structures in the DNS are different form those of the wall modeled LES with CNN and ODE-based wall models due to the difference in the grid resolution and applying filter in LES. However, for the PDNS shown in this figure which is obtained by reducing the number of grid points in DNS and applying projection, the vortical structures have become more similar to those of the wall modeled LES. The difference can be seen between the DNS and fDNS which is due to applying filter to the DNS data for fDNS. Also, it can be seen that the vortical structures for the CNN and ODE-based wall models are similar to each other.



(a) fDNS (b) CNN

(c) ODE (d) All on the same plot

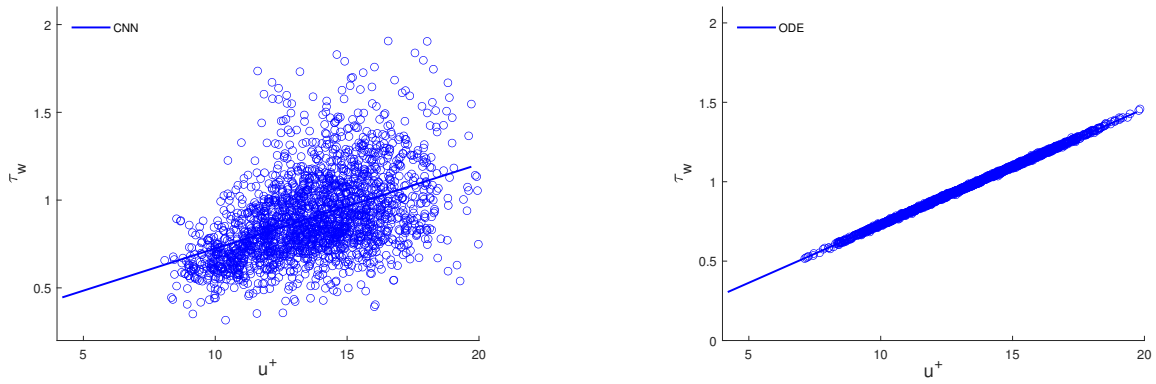
Figure 4.3. PDF of wall shear stress (τ_w) for $Re_\tau = 400$.



(a) u^{CNN} (b) u^{ODE}

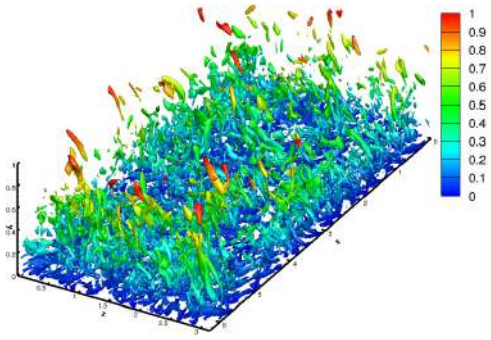
(c) τ_w^{CNN} (d) τ_w^{ODE}

Figure 4.4. The distribution of streamwise velocity (a, b) and wall shear stress (c, d) for the CNN and ODE-based wall models ($Re_\tau = 400$).

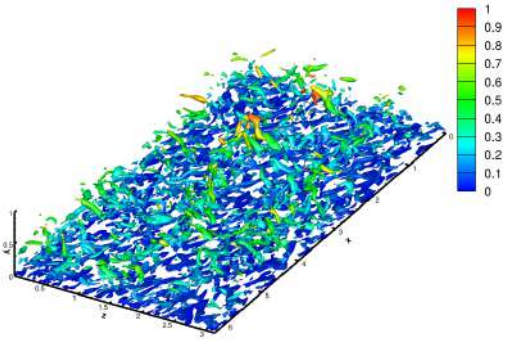


(a) CNN (b) ODE

Figure 4.5. Scatter plot of wall shear stress (τ_w) vs. streamwise velocity for CNN and ODE-based wall models ($Re_\tau = 400$).

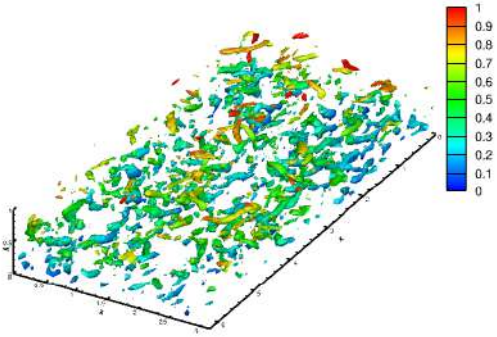


(a) DNS



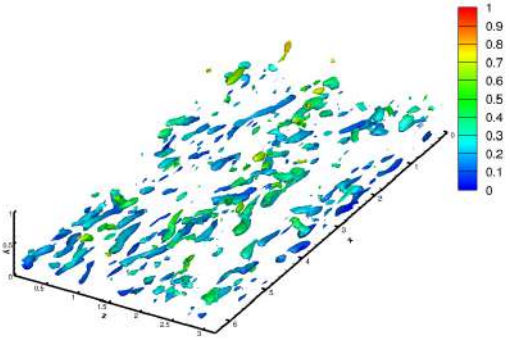
fDNS

(b)

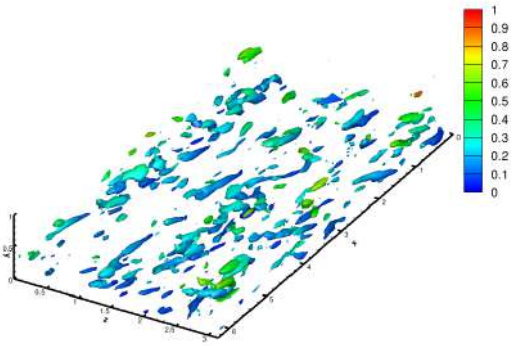


PDNS

(c)



(d) CNN



(e) ODE

Figure 4.6. Instantaneous vortical structures shown by isosurface of the second invariant of the deformation tensor (Q=0.01) for $Re_\tau = 400$. (a) DNS, (b) filtered DNS, (c) The size of the DNS data is reduced to the grid size of $N_x \times N_y \times N_z = 64 \times 64 \times 64$; then, the projection is applied to satisfy the continuity equation. (d) CNN model, and (e) ODE-based model.

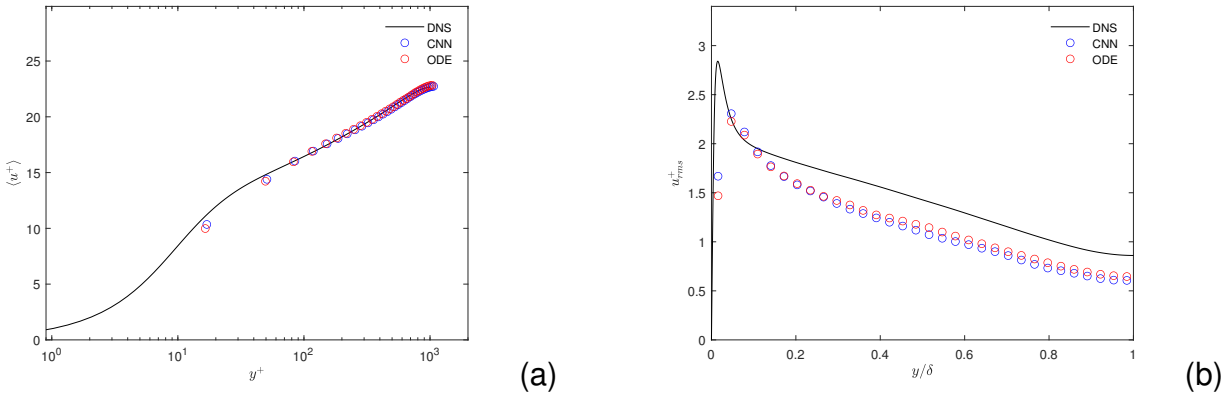


Figure 4.7. Mean velocity profile (a) and streamwise RMS velocity fluctuation (b) for $Re_\tau = 1000$.

4.3.2 Wall modeled LES for higher Reynolds number flow ($Re_\tau = 1000$)

In order to check the generalizability of the CNN wall model, the model is applied to the turbulent channel flow at $Re_\tau = 1000$, which is much larger than that used for the training of CNN. The data for DNS of turbulent channel flow at $Re_\tau = 1000$ are available from the previous studies [63, 121–123]. In the present study, the data for the Reynolds number of $Re_\tau = 1000$ have been taken from Ref. [123]. The domain size used for the wall modeled LES is $L_x \times L_y \times L_z = \pi \times 2 \times \pi$, and the grid size is $N_x \times N_y \times N_z = 64 \times 64 \times 64$. It is worth mentioning that in this case the domain size and the nondimensionalized filter size are different from those of training data as well. The data at the first grid point off the wall are used for the calculation of the wall model inputs and for the calculation of friction velocity the data at the third grid point off the wall are used. As the results in Fig. 4.7 show, the CNN wall model can predict the mean velocity well and its prediction for the streamwise RMS velocity fluctuation is comparable to that of the ODE-based wall model. As the underprediction for the streamwise RMS fluctuation occurs for both wall models, it can be related to the factors affecting the simulation like the SGS model used in this study. As mentioned, the static Smagorinsky model has been used as the SGS model which is one of the simplest SGS models and has shown to be very dissipative. Therefore, using a more complex SGS model may improve the results like the flow statistics [124–127]. It also should be mentioned that the suggestions made in Refs. [22, 94] like choosing that data at a location farther from the wall or filtering the data before feeding them to the wall model might be helpful to improve the results.

4.3.3 Applicability of the CNN model to coarser grid size

In this section, the CNN wall model is applied to a coarser grid size than that used for the training process. The domain size is the same as that used for the training, but the grid size is $N_x \times N_y \times N_z = 48 \times 48 \times 48$. For the CNN models it is important to check the applicability of them to the coarser grid size. Since the models based on CNN are trained on a two-dimensional domain as the input and output, they may show sensitivity to the grid size. The results of testing the model for the simulation on a coarser grid are shown in Fig. 4.8. From this figure it can be seen that the model can predict the mean velocity well and its behaves similarly to the ODE-based model in predicting the RMS velocity fluctuation. For the Reynolds shear stress, it can be seen that it underpredicts the peak value slightly more than when it is applied to a finer grid size. For the ODE-based model a similar performance to the CNN model in predicting the Reynolds shear stress is observed. In the study on the SGS modeling in LES using CNN it was shown that when the CNN model is applied to a coarser grid size than that used for the training, the mean velocity is overpredicted and the Reynolds shear stress is underpredicted compared to when it is applied to the same grid size as used for the training process [78]. As seen, the same occurs for the Reynolds shear stress for the wall modeled LES with the CNN-based wall model as well. However, it should be mentioned that the model does not show a severe sensitivity and is applicable to a coarser grid size.

4.3.4 Sensitivity of the CNN model to wall normal distance

After checking the CNN model for a coarser grid size, the sensitivity of the CNN wall model to the wall normal distance is checked. All the simulation parameters in this section are the same as those in Sec. 4.3.1 except for the wall normal distance for the wall model inputs, which is $y/\delta = 0.141$ in this section and is outside the training range of the CNN wall model which can be challenging. In the *a priori* test, it was shown that that the trained CNN wall models are sensitive to the wall-normal distance of the CNN inputs and their performance becomes worse when the wall-normal distance of the CNN inputs is outside their training range. However, the inner layer-based model used in this section was shown to perform better than the log layer-based wall model when is used for the data outside its training range. The results for the mean velocity, RMS velocity fluctuation, and Reynolds shear stress are presented in Fig. 4.9. As the results show, in this case the mean velocity profile is slightly overpredicted compared to when the

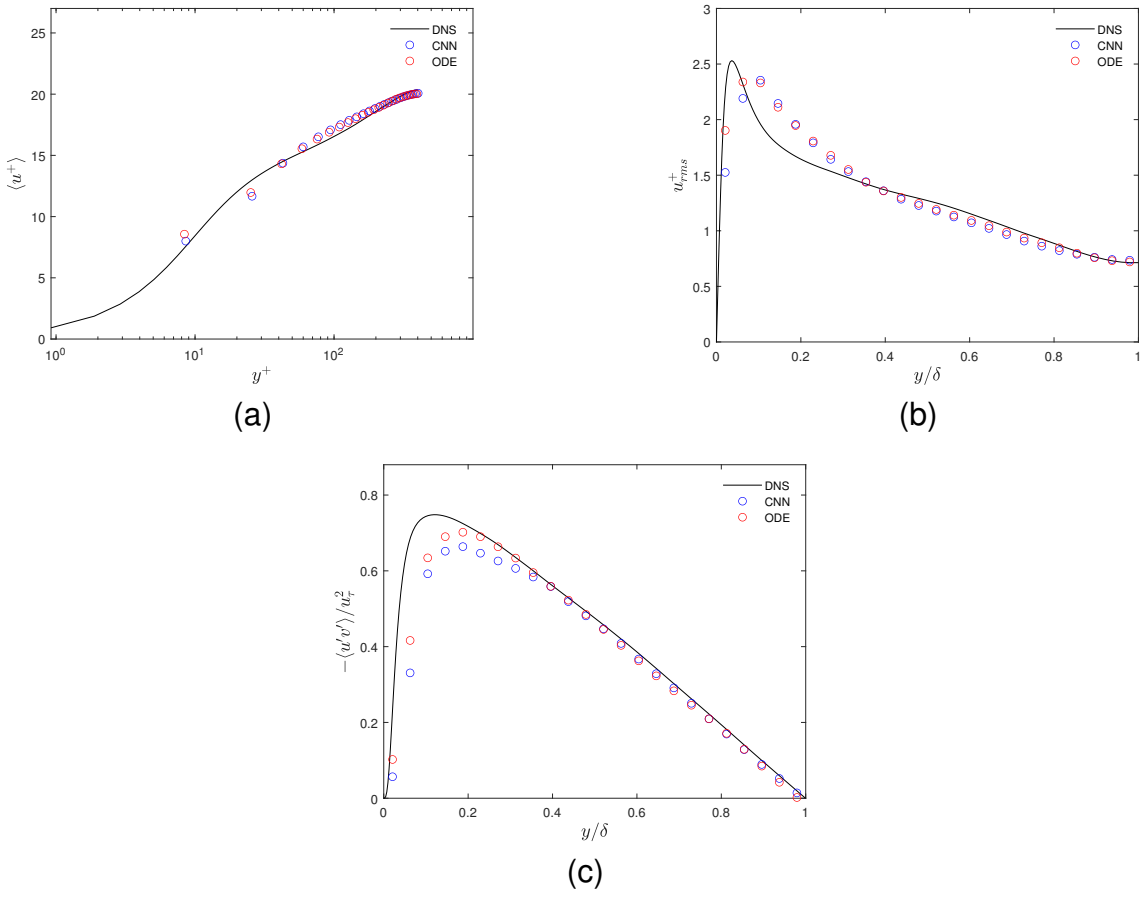


Figure 4.8. Mean velocity profile (a), streamwise RMS velocity fluctuation (b), and Reynolds shear stress (c) for $Re_\tau = 400$ ($N_x \times N_y \times N_z = 48 \times 48 \times 48$).

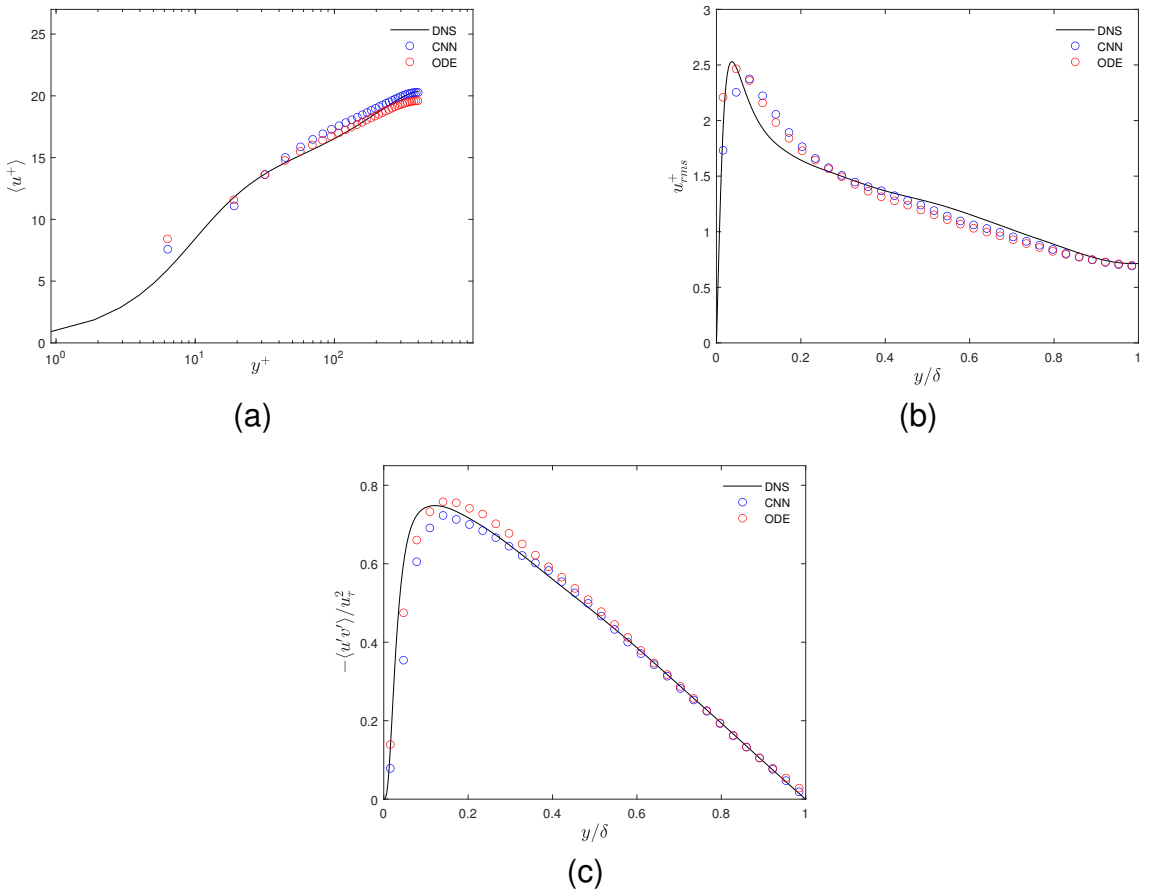


Figure 4.9. Mean velocity profile (a), streamwise RMS velocity fluctuation (b), and Reynolds shear stress (c) for $Re_\tau = 400$. Wall model inputs are provided by the LES data at $y/\delta = 0.141$.

wall-normal distance is in its training range. However, it can be seen that the results for the RMS velocity fluctuation and the Reynolds shear stress are comparable to the ODE-based wall model. Therefore, it can be said that on the whole the model has roughly reasonable performance although it is used for the wall-normal distance which is completely outside its training range.

4.3.5 Effect of normalization by friction velocity (u_τ)

Finally, the effect of the wall-normal distance of the data which are used for the calculation of the friction velocity is investigated. The calculation of the friction velocity is important for the wall-modeled LES with the CNN wall model. As mentioned, for the CNN wall model trained in the present study, all the CNN inputs and outputs are normalized by the wall units (u_τ, ν). In fact, the CNN inputs are normalized by the friction velocity before feeding to the CNN and the predicted wall shear stress is non-normalized by the viscosity and friction velocity to be used as the boundary condition for the streamwise and spanwise velocity components. In this section, two more wall normal distances, the 5^{th} and 7^{th} points off the wall are used for the

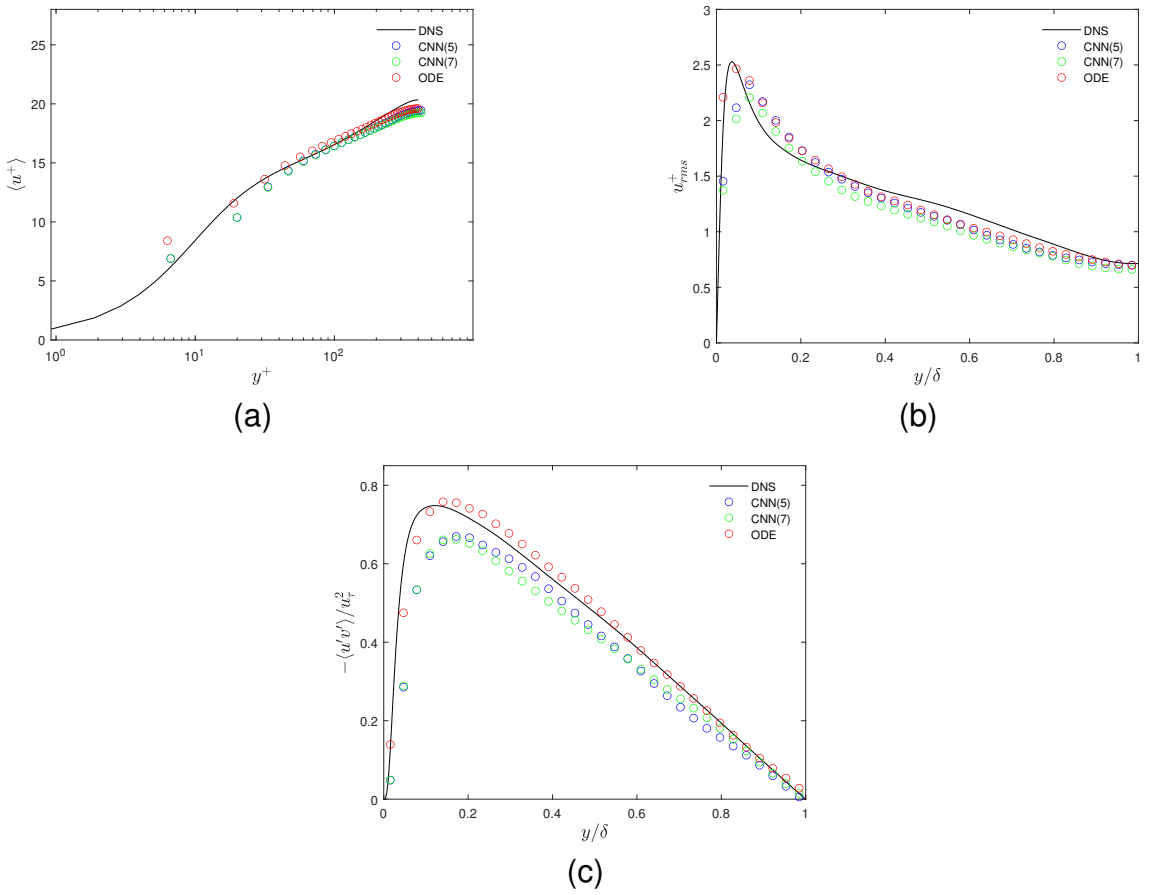


Figure 4.10. Effect of using the data at different wall-normal distance for calculating the friction velocity (u_τ). Mean velocity profile (a), streamwise RMS velocity (b), and Reynolds shear stress (c) for $Re_\tau = 400$. CNN(5) and CNN(7) mean that the data at the 5_{th} and 7^{th} point off the wall are used for calculating the friction velocity which correspond to $y/\delta = 0.141$ and $y/\delta = 0.203$, respectively.

calculation of the friction velocity. As seen in Fig. 4.10, for the mentioned cases the difference is not significant. However, by having a closer look and comparing the results with those of Fig. 4.2, it can be understood that choosing the data at a wall-normal distance farther from the wall for the calculation of friction velocity can lead to a slightly more underprediction in the mean velocity, RMS velocity fluctuation, and the Reynolds shear stress. Therefore, based on the numerical study performed in this section choosing a wall-normal distance within the logarithmic layer which is closer to the wall can be a safer choice and lead to better results.

4.4 Conclusions

In this chapter, *a posteriori* test was performed for the CNN wall model to investigate its performance in an actual wall modeled LES under different conditions. It was shown that on the whole, the CNN is successful in establishing a wall model. The performance of the model was

also compared with the ODE-based wall model. The results showed that the CNN wall model is successful in predicting the mean velocity profile for the same Reynolds number as training. It was also shown that the results for the streamwise RMS velocity fluctuation is comparable to the ODE-based wall model. However, for the Reynolds shear stress the results for the CNN model were slightly undepredicted. The wall shear stress predicted by the CNN model was compared with that of the ODE-based wall model, and it was shown that the CNN wall model is more successful in predicting the PDF of wall shear stress. It was also shown that the unlike the ODE-based wall model, the wall shear stress predicted by the CNN wall model is poorly correlated with the velocity which is consistent with the DNS results. The applicability of the CNN model to a higher Reynolds number flow and a coarser grid size was also investigated, and it was shown that the CNN model is successful in predicting the mean velocity and Reynolds stress. However, for the coarser grid size, the Reynolds shear stress was slightly underpredicted more than the ODE-based wall model. After that, the model was tested for a condition in which the wall-normal distance of the CNN model is outside of its training range. It was shown that the results are almost comparable to those of the ODE-based wall model and no significant change was observed in predicting the statistics compared to when the matching location is in its training range. However, the mean velocity profile in this case was slightly overpredicted. The effect of the wall-normal distance of the data used for calculating the friction velocity was checked as well. The calculation of friction velocity in the CNN wall model is important as the wall model inputs are normalized by the friction velocity before feeding to the CNN. The results showed that choosing a wall normal distance in the logarithmic layer which is closer to the wall leads to better results. However, in the future additional studies are required to find out the criteria for choosing the wall-normal distance of the inputs used for the calculation of the friction velocity.

In the future, the effect of normalization can be investigated further by trying other parameters for the normalization like the wall-normal distance. In order to check if the results can be improved further and to check the sensitivity of the CNN model to the SGS stress, more complex SGS models can be tested as well. Also, the other CNN models trained in Chapter 3 can be tested to check their performance and to see if they can lead to better results in predicting the statistics of the flow field.

5. Conclusions

5.1 Summary and conclusions

In this dissertation, two separate studies were performed on the applicability of the data-driven approaches for the turbulence modeling in LES. In Chapter 2, a study was performed on the SGS modeling for the LES of Burgers turbulence using FCNN. Chapter 3 was focused on developing a data-driven wall model using CNN for LES of turbulent channel flow and investigating the accuracy of the obtained model in the *a priori* test, and also comparing its performance in predicting the wall shear stress with the existing models. In Chapter 4, the performance of the obtained CNN wall model was checked in the *a posteriori* test by using it in an actual simulation.

The objective of Chapter 2 was to develop an SGS model for the Burgers equation which can be applied to the large filter sizes and be generalized to various conditions different from those used for obtaining the training dataset. Since one of the goals in this study was to find an SGS model which can be applicable to the large filter sizes, it was required to perform DNS with fine grid resolution which is not possible for turbulent flows with higher dimensions. This makes it possible to apply large filter sizes to the DNS data for obtaining the training datasets. For this reason, the Burgers equation was chosen in this study since while it shares similarities with the Navier-Stokes equations and the strong shocks occurring in the domain makes it challenging, it is computationally cheap due to being one-dimensional. Therefore, it gives us the chance to perform DNS with a large number of grid points. The Gaussian filter was used to filter the data as compared to the box filter and cut-off filter, it led to a higher correlation coefficient between the SGS stress predicted by the SGS model and that calculated using the DNS data. In order to improve the generalizability of the obtained SGS model, the training data were taken from three DNSs with different forcing terms and viscosities. Additionally, the data were filtered using three different filter sizes to include the effect of filter size. Considering the relations for the eddy viscosity models, different input choices were considered and the effect of including the filter size either as the input choice or as a parameter for making normalization was checked as well. Furthermore, the performance of the FCNN SGS models was compared with the existing SGS

models. Initially, the performance of the models was checked in the *a priori* test. The results showed that the neural network models are successful in predicting the SGS stress in the *a priori* test and their performance is better than the gradient model, comparable to DSM, and in some cases even better than DSM. It was also shown that among the existing models they tend more to overpredict the SGS stress instead of underpredicting it, which is like the DSM. After the *a priori* test, the performance of the models were checked extensively in the *a posteriori* test by using them in an actual LES. The results of the *a posteriori* showed that not all the trained FCNN models are successful for the LES of Burgers turbulence under various conditions and adding the filter width as an additional input does not have any obvious effect in improving the performance of the model while normalizing by the filter width can slightly improve the model performance. Among the tested cases, the models which include $d\bar{u}/dx|d\bar{u}/dx|$ in their input were shown to perform better than the FCNN models with other input choices. This input choice for the Burgers equation is similar to find the coefficient (C) for the DSM. The results showed that the models with the mentioned input choices can be generalized to any condition and their performance is better than the gradient model, and comparable to the DSM while their computational cost is much lower than the DSM.

The objective of Chapter 3 was to investigate the applicability of CNN in constructing a wall model for LES of turbulent channel flow and to check the performance of the obtained model in predicting the wall shear stress. CNN has already shown a great ability in extracting the hidden features and has been widely used for the SGS modeling in LES. Therefore, it is worth it to investigate its ability in establishing a wall model. In this chapter, initially, a parametric study was performed on the input choices and on the hyperparameters of CNN. Two input choices were selected for training the CNN model and for each input choices, the data from two different regions, one limited to the inner layer (excluding $h^+ \leq 10$) and the other one limited to the logarithmic layer were considered due to the different characteristics of these two regions. The results showed that for CNN even with the simple input choices like the velocity components, a high correlation coefficient is achieved between the wall shear stress calculated using the filtered DNS data and that predicted by the CNN model. After developing the model, *a priori* test was carried out and the performance of the models was checked under various conditions and compared with two existing models, the shifted model and the ODE-based model. Initially, the effect of the wall-normal distance of the wall model inputs on the prediction of the wall shear stress was checked. The results showed that the CNN models are successful in

predicting the wall shear stress, better than the existing models. However, when the wall normal distance is outside the training range of the CNN wall models, their performance becomes worse compared to when it is inside their training range. However, the overall performance of the inner layer-based wall models was better when they were used for the wall-normal distance outside their training range. Then, the models were applied to two higher Reynolds number cases ($Re_\tau = 600, 800$) and coarser grid sizes ($N_x \times N_z = 32 \times 32, 48 \times 48$). The results showed that the models are applicable to higher Reynolds number flows and coarser grid sizes, and their performance in these cases is better than the existing models (closer to the filtered DNS data) as well in predicting the distribution of wall shear-stress, the joint PDF of wall shear-stress, and the scatter plot of wall shear stress vs. velocity at the matching point. It should be mentioned that when the CNN model was used for a very coarse grid size, it became less accurate in predicting the wall shear-stress, however, it was still more accurate than the existing models in predicting the wall shear stress. It was expected as the CNN wall model is trained on the two-dimensional domain. Finally, the sensitivity of the model to the domain size was checked by performing local learning. In this case, the CNN model was trained on a quarter of the domain and was resized to predict the wall shear stress for the whole domain. The results of this part showed that the model has reasonable prediction of the wall shear stress when used for a different domain size and no obvious change was observed compared to when it was trained on the whole domain.

In Chapter 4, the objective was to perform *a posteriori* test for the CNN wall model and check the performance of the model in an actual simulation. The performance of the model was compared with the ODE-based wall model as well. Initially, the model was tested for the same Reynolds number as training. The results showed that the CNN model has reasonable accuracy in predicting the mean velocity profile. The performance of the model in predicting the streamwise RMS velocity fluctuation is comparable to the ODE-based wall model. However, for the Reynolds shear stress ($\langle u'v' \rangle$), it slightly underpredicts the values more than the ODE-based wall model. Then, the CNN model was tested for a higher Reynolds number case ($Re_\tau = 1000$). The results showed that the model can predict the mean velocity well. The model also behaved similarly to the ODE-based wall in predicting the streamwise RMS velocity fluctuation. In this case for both the CNN and ODE-based wall models the streamwise RMS velocity fluctuation was underpredicted. However, in this study, the static Smagorinsky model was used as the SGS model, and it can be expected that the performance of the models in predicting the RMS velocity fluctuation improves by using a more complex SGS model. After that, the sensitivity of the

model to a coarser grid size was checked and the model was successful in predicting the mean velocity. However, it still underpredicts the peak of the Reynolds shear stress slightly more than the ODE-based wall model. The model was tested for a matching location outside the training range of CNN as well. In this case, the model slightly overpredicts the mean velocity profile. It was expected from the results of the *a priori* test that the model has lower accuracy when the wall-normal distance of the wall model inputs is outside the training range. However, in predicting the streamwise RMS velocity fluctuation and Reynolds shear stress, its performance is roughly comparable to the ODE-based wall model. The CNN model inputs are normalized in each time step by the friction velocity. Therefore, the wall-normal distance whose data were used for calculating the friction velocity was checked as well. The results showed that when the data at a farther location off the wall are used for calculating the friction velocity, the mean velocity profile and the Reynolds shear stress are underpredicted compared to when the data at a closer location in the logarithmic region are used. Therefore, it is better to use the data closer to the wall in the logarithmic region for calculating the friction velocity which is used for normalizing the CNN inputs. The PDF of the wall shear stress, the distribution of the wall shear stress, and the scatter plot of the wall shear stress vs. velocity were also plotted for the CNN and ODE-based wall models. The results were in agreement with those of the *a priori* test. The PDF of the wall shear stress for the CNN wall model was closer to that of the filtered DNS compared to the ODE-based wall model. Additionally, for the CNN model the wall shear stress is poorly correlated with the velocity at the matching location which is in agreement with the filtered DNS data.

5.2 Future work

As the future work, there are some studies can be done based on the results obtained in the present dissertation to investigate the ability of the FCNN and CNN approaches further in establishing the SGS model and wall model, respectively. In the second chapter of the present dissertation, a study was performed on the SGS modeling for Burgers turbulence using FCNN, and it was shown that $d\bar{u}/dx|d\bar{u}/dx|$ is a proper input choice. It was also shown that choosing the mentioned input and including the filter width as a parameter for normalization as well as using the data from simulations under different conditions and filtering the data using different filter sizes can lead to find an SGS model which is has reasonable cost and accuracy, and can be generalized to various conditions. Since the ultimate goal for developing the turbulence

models is to check their applicability to the three-dimensional flow, as a future direction, the mentioned points can be considered to develop an FCNN SGS model for the three-dimensional flows which is more challenging due to the complexity of such flows. In the Burgers equation, we have only one component for the SGS stress term (τ_{xx}) which is normally positive for the Burgers equation and means that no backscatter exists in this case. However, negative values might be predicted by some SGS models. Additionally, due to having other velocity components the input choice will not be as simple as Burgers equations. But it is obvious that for the other existing models the computational cost will increase significantly as well compared to the one-dimensional case. Therefore, there are several issues which can be tested in the future for the three-dimensional flows based on the results on the present study. The first one is to check if choosing the inputs similar to that of the Burgers equation in the three dimensional case can lead to find a model with a comparable (or better) performance to the dynamic Smagorinsky model with lower computational cost and see if it can be stable in the *a posteriori* without any stabilization technique. The second one is to check if including the filter size as a parameter for normalization can improve the model performance. The last one is to investigate if including the data from the simulations under various conditions like different Reynolds numbers and filtering them using different filter sizes can improve the generalizability of the data-driven SGS model. However, as mentioned, for the three-dimensional case we are not still able to apply a very large filter size to the datasets since due to the limitations in the computational resources we cannot perform simulations on a very fine grid as we did for the Burgers equation.

In the third and fourth chapters of this dissertation a wall model was developed using CNN. The results of the *a priori* test showed that applying the CNN wall model to the grid size much coarser than that used for training the model can make the model less accurate in predicting the wall shear stress. Also, the results of the *a posteriori* test showed that the wall-normal distance of the data used for the calculation of the friction velocity can affect the simulation results as the CNN inputs are normalized by the friction velocity. Therefore, as a future work, in order to increase the generalizability of the model, we can use the data of different Reynolds numbers which have the same input size for the CNN while are filtered using different nondimensionalized filter sizes as the training data. In this case, the CNN might be able to learn the effect of filter size and Reynolds number. Additionally, we can test the effect of using other parameters for nondimensionalizing the training datasets like the wall-normal distance. As done in this study, we can also use the local learning for developing a wall model which can make the model less

sensitive to the domain and increase the size of the training dataset. In Chapter 4, the *a posteriori* test was carried out only for the CNN wall model with the velocity components as the input choice which was trained using the data in the inner layer (excluding $h^+ \leq 10$) since in the last part of that chapter a wall model was developed with these specifications to predict the wall shear stress in both the streamwise and spanwise directions. However, in the future, the other trained models can be tested in an actual simulation to see if a better model can be found.

In the simulations performed in the *a posteriori* test, the static Smagorinsky model was used as the SGS model which is one of the simplest SGS models and has been shown to be very dissipative. In the future, the other SGS models should be tested in the simulations to check the sensitivity of the CNN wall model to the SGS model and to enhance the results obtained for the statistics of the flow, including the streamwise RMS velocity fluctuation and Reynolds shear stress. Finally, it should be mentioned that although CNN has shown to be successful in establishing a wall model for the turbulent channel flow, its applications to the complex geometries and flow regimes like the flows which experience a strongly non-equilibrium effect can be challenging. Regarding the complex geometry, since the CNN gets the input in two-dimensional form, some measures like the data transformation should be considered if it wants to be used for such flows. However, due to the existence of these flows and their importance, it is worth it to consider it in the future studies.

References

[1] Pierre Sagaut, Marc Terracol, and Sebastien Deck. *Multiscale and multiresolution approaches in turbulence-LES, DES and Hybrid RANS/LES Methods: Applications and Guidelines*. World Scientific, 2013.

[2] Luis Roberto Tay Wo Chong Hilares. *Numerical simulation of the dynamics of turbulent swirling flames*. PhD thesis, Technische Universität München, 2012.

[3] Ugo Piomelli and Elias Balaras. Wall-layer models for large-eddy simulations. *Annual review of fluid mechanics*, 34(1):349–374, 2002.

[4] Stephen B Pope and Stephen B Pope. *Turbulent flows*. Cambridge university press, 2000.

[5] Joseph Smagorinsky. General circulation experiments with the primitive equations: I. the basic experiment. *Monthly weather review*, 91(3):99–164, 1963.

[6] Robert A Clark, Joel H Ferziger, and William Craig Reynolds. Evaluation of subgrid-scale models using an accurately simulated turbulent flow. *Journal of fluid mechanics*, 91(1):1–16, 1979.

[7] Jorge Bardina, J Ferziger, and WC Reynolds. Improved subgrid-scale models for large-eddy simulation. In *13th fluid and plasmadynamics conference*, page 1357, 1980.

[8] Massimo Germano, Ugo Piomelli, Parviz Moin, and William H Cabot. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3(7):1760–1765, 1991.

[9] Bert Vreman, Bernard Geurts, and Hans Kuerten. Large-eddy simulation of the temporal mixing layer using the clark model. *Theoretical and Computational Fluid Dynamics*, 8(4):309–324, 1996.

[10] Yan Zang, Robert L Street, and Jeffrey R Koseff. A dynamic mixed subgrid-scale model and its application to turbulent recirculating flows. *Physics of Fluids A: Fluid Dynamics*, 5(12):3186–3196, 1993.

[11] Arup Das and Robert D Moser. Optimal large-eddy simulation of forced burgers equation. *Physics of Fluids*, 14(12):4344–4351, 2002.

[12] A LaBryer, PJ Attar, and P Vedula. A framework for large eddy simulation of burgers turbulence based upon spatial and temporal statistical information. *Physics of Fluids*, 27(3):035116, 2015.

[13] Romit Maulik and Omer San. A dynamic closure modeling framework for large eddy simulation using approximate deconvolution: Burgers equation. *Cogent Physics*, 5(1):1464368, 2018.

[14] Marcel Lesieur and Olivier Metais. New trends in large-eddy simulations of turbulence. *Annual review of fluid mechanics*, 28(1):45–82, 1996.

[15] Jacob A Langford and Robert D Moser. Optimal les formulations for isotropic turbulence. *Journal of fluid mechanics*, 398:321–346, 1999.

[16] Charles Meneveau and Joseph Katz. Scale-invariance and turbulence models for large-eddy simulation. *Annual Review of Fluid Mechanics*, 32(1):1–32, 2000.

[17] Thomas JR Hughes, Luca Mazzei, and Kenneth E Jansen. Large eddy simulation and the variational multiscale method. *Computing and visualization in science*, 3(1):47–59, 2000.

[18] Haecheon Choi and Parviz Moin. Grid-point requirements for large eddy simulation: Chapman's estimates revisited. *Physics of fluids*, 24(1):011702, 2012.

[19] Saleh Rezaeiravesh, Mattias Liefvendahl, and Christer Fureby. On grid resolution requirements for les of wall-bounded flows. In *ECCOMAS Congress 2016, June 5–10, Crete, Greece*, pages 7454–7465. European Community on Computional Methods in Applied Sciences (ECCOMAS), 2016.

[20] Xiang IA Yang and Kevin P Griffin. Grid-point and time-step requirements for direct numerical simulation and large-eddy simulation. *Physics of Fluids*, 33(1):015108, 2021.

[21] Johan Larsson, Soshi Kawai, Julien Bodart, and Ivan Bermejo-Moreno. Large eddy simulation with modeled wall-stress: recent progress and future directions. *Mechanical Engineering Reviews*, 3(1):15–00418, 2016.

[22] Soshi Kawai and Johan Larsson. Wall-modeling in large eddy simulation: Length scales, grid resolution, and accuracy. *Physics of Fluids*, 24(1):015105, 2012.

[23] Sanjeeb T Bose and Parviz Moin. A dynamic slip boundary condition for wall-modeled large-eddy simulation. *Physics of Fluids*, 26(1):015104, 2014.

[24] XIA Yang, J Sadique, R Mittal, and Charles Meneveau. Integral wall model for large eddy simulations of wall-bounded turbulent flows. *Physics of Fluids*, 27(2):025112, 2015.

[25] Hyunji Jane Bae, Adrián Lozano-Durán, Sanjeeb T Bose, and Parviz Moin. Dynamic slip wall model for large-eddy simulation. *Journal of fluid mechanics*, 859:400–432, 2019.

[26] James W Deardorff. A numerical study of three-dimensional turbulent channel flow at large reynolds numbers. *Journal of Fluid Mechanics*, 41(2):453–480, 1970.

[27] Ulrich Schumann. Subgrid scale model for finite difference simulations of turbulent flows in plane channels and annuli. *Journal of computational physics*, 18(4):376–404, 1975.

[28] Ugo Piomelli, Joel Ferziger, Parviz Moin, and John Kim. New approximate boundary conditions for large eddy simulations of wall-bounded flows. *Physics of Fluids A: Fluid Dynamics*, 1(6):1061–1068, 1989.

[29] Elie Bou-Zeid, Charles Meneveau, and Marc B Parlange. Large-eddy simulation of neutral atmospheric boundary layer flow over heterogeneous surfaces: Blending height and effective surface roughness. *Water Resources Research*, 40(2), 2004.

[30] Romain Mathis, Ivan Marusic, Sergei I Chernyshenko, and Nicholas Hutchins. Estimating wall-shear-stress fluctuations given an outer region input. *Journal of Fluid Mechanics*, 715:163–180, 2013.

[31] Franck Nicoud, JS Baggett, Parviz Moin, and William Cabot. Large eddy simulation wall-modeling based on suboptimal control theory and linear stochastic estimation. *Physics of fluids*, 13(10):2968–2984, 2001.

[32] H Jane Bae and Petros Koumoutsakos. Scientific multi-agent reinforcement learning for wall-models of turbulent flows. *Nature Communications*, 13(1):1–9, 2022.

[33] Jeffrey S Baggett. On the feasibility of merging les with rans for the near-wall region of attached turbulent flows. *Annual Research Briefs*, pages 267–277, 1998.

[34] NV Nikitin, Franck Nicoud, B Wasistho, KD Squires, and Philippe R Spalart. An approach to wall modeling in large-eddy simulations. *Physics of fluids*, 12(7):1629–1632, 2000.

[35] Ugo Piomelli, Elias Balaras, Hugo Pasinato, Kyle D Squires, and Philippe R Spalart. The inner–outer layer interface in large-eddy simulations with wall-layer models. *International Journal of heat and fluid flow*, 24(4):538–550, 2003.

[36] Mikhail L Shur, Philippe R Spalart, Mikhail Kh Strelets, and Andrey K Travin. A hybrid rans-les approach with delayed-des and wall-modelled les capabilities. *International journal of heat and fluid flow*, 29(6):1638–1649, 2008.

[37] H Jane Bae and Adrián Lozano-Durán. Effect of wall boundary conditions on a wall-modeled large-eddy simulation in a finite-difference framework. *Fluids*, 6(3):112, 2021.

[38] Sotiris B Kotsiantis, Ioannis Zaharakis, P Pintelas, et al. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1):3–24, 2007.

[39] Brian D Ripley. *Pattern recognition and neural networks*. Cambridge university press, 2007.

[40] Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. Turbulence modeling in the age of data. *Annual review of fluid mechanics*, 51:357–377, 2019.

[41] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.

[42] Rui Fang, David Sondak, Pavlos Protopapas, and Sauro Succi. Neural network models for the anisotropic reynolds stress tensor in turbulent channel flow. *Journal of Turbulence*, 21(9-10):525–543, 2020.

[43] Rafael Diez Sanhueza, Stephan HHJ Smit, Jurriaan WR Peeters, and Rene Pecnik. Machine learning for rans turbulence modeling of variable property flows. *Computers & Fluids*, 255:105835, 2023.

[44] Jin-Long Wu, Heng Xiao, and Eric Paterson. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Physical Review Fluids*, 3(7):074602, 2018.

[45] Fabrizio Sarghini, G De Felice, and Stefania Santini. Neural networks based subgrid scale modeling in large eddy simulations. *Computers & fluids*, 32(1):97–108, 2003.

[46] Romit Maulik and Omer San. A neural network approach for the blind deconvolution of turbulent flows. *Journal of Fluid Mechanics*, 831:151–181, 2017.

[47] Masataka Gamahara and Yuji Hattori. Searching for turbulence models by artificial neural network. *Physical Review Fluids*, 2(5):054604, 2017.

[48] Zhuo Wang, Kun Luo, Dong Li, Junhua Tan, and Jianren Fan. Investigations of data-driven closure for subgrid-scale stress in large-eddy simulation. *Physics of Fluids*, 30(12):125101, 2018.

[49] Zhideng Zhou, Guowei He, Shizhao Wang, and Guodong Jin. Subgrid-scale model for large-eddy simulation of isotropic turbulent flows using an artificial neural network. *Computers & Fluids*, 195:104319, 2019.

[50] Andrea Beck, David Flad, and Claus-Dieter Munz. Deep neural networks for data-driven les closure models. *Journal of Computational Physics*, 398:108910, 2019.

[51] Chenyue Xie, Jianchun Wang, and E Weinan. Modeling subgrid-scale forces by spatial artificial neural networks in large eddy simulation of turbulence. *Physical Review Fluids*, 5(5):054606, 2020.

[52] Zelong Yuan, Chenyue Xie, and Jianchun Wang. Deconvolutional artificial neural network models for large eddy simulation of turbulence. *Physics of Fluids*, 32(11):115106, 2020.

[53] Satoshi Miyazaki and Yuji Hattori. Improving accuracy of turbulence models by neural network. *arXiv preprint arXiv:2012.01723*, 2020.

[54] Chenyue Xie, Zelong Yuan, and Jianchun Wang. Artificial neural network-based nonlinear algebraic models for large eddy simulation of turbulence. *Physics of Fluids*, 32(11):115101, 2020.

[55] Jonghwan Park and Haecheon Choi. Toward neural-network-based large eddy simulation: application to turbulent channel flow. *Journal of Fluid Mechanics*, 914, 2021.

[56] Lei Wu and Zuoli Xiao. Subgrid-scale stress modeling based on artificial neural network. *Chinese Journal of Theoretical and Applied Mechanics*, 53(10):2667–2681, 2021.

[57] Robin Stoffer, Caspar M Van Leeuwen, Damian Podareanu, Valeriu Codreanu, Menno A Veerman, Martin Janssens, Oscar K Hartogensis, and Chiel C Van Heerwaarden. Development of a large-eddy simulation subgrid model based on artificial neural networks: a case study of turbulent channel flow. *Geoscientific Model Development*, 14(6):3769–3788, 2021.

[58] Qingjia Meng, Zhou Jiang, and Jianchun Wang. Artificial neural network-based subgrid-scale models for les of compressible turbulent channel flow. *Theoretical and Applied Mechanics Letters*, 13(1):100399, 2023.

[59] Yu Cheng, Marco Giometto, Pit Kauffmann, Ling Lin, Chen Cao, Cody Zupnick, Harold Li, Qi Li, Ryan Abernathey, and Pierre Gentine. Deep learning for subgrid-scale turbulence modeling in large-eddy simulations of the atmospheric boundary layer. *arXiv preprint arXiv:1910.12125*, 2019.

[60] Nathaniel Saura and Thomas Gomez. Subgrid stress tensor prediction in homogeneous isotropic turbulence using 3d-convolutional neural networks. *Available at SSRN 4184202*.

[61] Adam Subel, Ashesh Chattopadhyay, Yifei Guan, and Pedram Hassanzadeh. Data-driven subgrid-scale modeling of forced burgers turbulence using deep learning with generalization to higher reynolds numbers via transfer learning. *Physics of Fluids*, 33(3):031702, 2021.

[62] Golsa Tabe Jamaat and Yuji Hattori. Development of subgrid-scale model for les of burgers turbulence with large filter size. *Physics of Fluids*, 34(4):045120, 2022.

[63] XIA Yang, Suhaib Zafar, J-X Wang, and Heng Xiao. Predictive large-eddy-simulation wall modeling via physics-informed neural networks. *Physical Review Fluids*, 4(3):034602, 2019.

[64] Sarath Radhakrishnan, Lawrence Adu Gyamfi, Arnau Miró, Bernat Font, Joan Calafell, and Oriol Lehmkuhl. A data-driven wall-shear stress model for les using gradient boosted decision trees. In *High Performance Computing: ISC High Performance Digital 2021 International Workshops, Frankfurt am Main, Germany, June 24–July 2, 2021, Revised Selected Papers 36*, pages 105–121. Springer, 2021.

[65] Zhideng Zhou, Guowei He, and Xiaolei Yang. Wall model based on neural networks for les of turbulent flows over periodic hills. *Physical Review Fluids*, 6(5):054610, 2021.

[66] Aurélien Vadrot, Xiang IA Yang, H Jane Bae, and Mahdi Abkar. Log-law recovery through reinforcement-learning wall model for large-eddy simulation. *arXiv preprint arXiv:2302.14391*, 2023.

[67] Xinyi LD Huang, Xiang IA Yang, and Robert F Kunz. Wall-modeled large-eddy simulations of spanwise rotating turbulent channels—comparing a physics-based approach and a data-based approach. *Physics of Fluids*, 31(12):125105, 2019.

[68] Di Zhou, Michael P Whitmore, Kevin P Griffin, and H Jane Bae. Multi-agent reinforcement learning for wall modeling in les of flow over periodic hills. *arXiv preprint arXiv:2211.16427*, 2022.

[69] Aurélien Vadrot, Xiang IA Yang, and Mahdi Abkar. A survey of machine learning wall models for large eddy simulation. *arXiv preprint arXiv:2211.03614*, 2022.

[70] MD Love. Subgrid modelling studies with burgers' equation. *Journal of Fluid Mechanics*, 100(1):87–110, 1980.

[71] Sukanta Basu. Can the dynamic eddy-viscosity class of subgrid-scale models capture inertial-range properties of burgers turbulence? *Journal of Turbulence*, (10):N12, 2009.

[72] Yanan Li and ZJ Wang. A priori and a posteriori evaluations of sub-grid scale models for the burgers' equation. *Computers & Fluids*, 139:92–104, 2016.

[73] Romit Maulik and Omer San. Explicit and implicit les closures for burgers turbulence. *Journal of Computational and Applied Mathematics*, 327:12–40, 2018.

[74] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

[75] Naoki Moriya, Kai Fukami, Yusuke Nabae, Masaki Morimoto, Taichi Nakamura, and Koji Fukagata. Inserting machine-learned virtual wall velocity for large-eddy simulation of turbulent channel flows. *arXiv preprint arXiv:2106.09271*, 2021.

[76] Luca Guastoni, Alejandro Güemes, Andrea Ianiro, Stefano Discetti, Philipp Schlatter, Hossein Azizpour, and Ricardo Vinuesa. Convolutional-network models to predict wall-bounded turbulence from wall quantities. *Journal of Fluid Mechanics*, 928:A27, 2021.

[77] Qingyi Zhao, Guodong Jin, and Zhideng Zhou. Deep learning method for the super-resolution reconstruction of small-scale motions in large-eddy simulation. *AIP Advances*, 12(12):125304, 2022.

[78] Bo Liu, Huiyang Yu, Haibo Huang, Nansheng Liu, and Xiyun Lu. Investigation of nonlocal data-driven methods for subgrid-scale stress modeling in large eddy simulation. *AIP Advances*, 12(6):065129, 2022.

[79] Suraj Pawar and Omer San. Data assimilation empowered neural network parametrizations for subgrid processes in geophysical flows. *Physical Review Fluids*, 6(5):050501, 2021.

[80] Yifei Guan, Ashesh Chattopadhyay, Adam Subel, and Pedram Hassanzadeh. Stable a posteriori les of 2d turbulence using convolutional neural networks: Backscattering analysis and generalization to higher re via transfer learning. *Journal of Computational Physics*, 458:111090, 2022.

[81] Suraj Pawar, Omer San, Adil Rasheed, and Prakash Vedula. Frame invariant neural network closures for kraichnan turbulence. *arXiv preprint arXiv:2201.02928*, 2022.

[82] Yifei Guan, Adam Subel, Ashesh Chattopadhyay, and Pedram Hassanzadeh. Learning physics-constrained subgrid-scale closures in the small-data regime for stable and accurate les. *Physica D: Nonlinear Phenomena*, page 133568, 2022.

[83] Adam Subel, Yifei Guan, Ashesh Chattopadhyay, and Pedram Hassanzadeh. Explaining the physics of transfer learning in data-driven turbulence modeling. *PNAS nexus*, 2(3):pgad015, 2023.

[84] Suraj Pawar, Omer San, Adil Rasheed, and Prakash Vedula. A priori analysis on deep learning of subgrid-scale parameterizations for kraichnan turbulence. *Theoretical and Computational Fluid Dynamics*, 34(4):429–455, 2020.

[85] Maxime Bassenne and Adrián Lozano-Durán. Computational model discovery with reinforcement learning. *arXiv preprint arXiv:2001.00008*, 2019.

[86] Xuping Xie, Clayton Webster, and Traian Iliescu. Closure learning for nonlinear model reduction using deep residual neural network. *Fluids*, 5(1):39, 2020.

[87] Jeric Alcala and Ilya Timofeyev. Subgrid-scale parametrization of unresolved scales in forced burgers equation using generative adversarial networks (gan). *Theoretical and Computational Fluid Dynamics*, 35(6):875–894, 2021.

[88] Fernando Manrique de Lara and Esteban Ferrer. Accelerating high order discontinuous galerkin solvers using neural networks: 1d burgers' equation. *Computers & Fluids*, page 105274, 2022.

[89] Omer San, Romit Maulik, and Mansoor Ahmed. An artificial neural network framework for reduced order modeling of transient flows. *Communications in Nonlinear Science and Numerical Simulation*, 77:271–287, 2019.

[90] Takashi Ishihara, Toshiyuki Gotoh, and Yukio Kaneda. Study of high–reynolds number isotropic turbulence by direct numerical simulation. *Annual Review of Fluid Mechanics*, 41:165–180, 2009.

[91] VC Wong and DK Lilly. A comparison of two dynamic subgrid closure methods for turbulent thermal convection. *Physics of Fluids*, 6(2):1016–1023, 1994.

[92] Stephen B Pope. Ten questions concerning the large-eddy simulation of turbulent flows. *New journal of Physics*, 6(1):35, 2004.

[93] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[94] Xiang IA Yang, George Ilhwan Park, and Parviz Moin. Log-layer mismatch and modeling of the fluctuating wall stress in wall-modeled large-eddy simulations. *Physical review fluids*, 2(10):104601, 2017.

[95] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[96] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.

[97] Xiang IA Yang and Mahdi Abkar. A hierarchical random additive model for passive scalars in wall-bounded flows at high reynolds numbers. *Journal of fluid mechanics*, 842:354–380, 2018.

[98] H Jane Bae and Adrian Lozano-Duran. Eddy-viscosity wall boundary condition for wall-modeled large-eddy simulation in a finite-difference framework. *Bulletin of the American Physical Society*, 2022.

[99] W Cabot and Parviz Moin. Approximate wall boundary conditions in the large-eddy simulation of high reynolds number flow. *Flow, Turbulence and Combustion*, 63(1):269–291, 2000.

[100] Meng Wang and Parviz Moin. Dynamic wall modeling for large-eddy simulation of complex turbulent flows. *Physics of Fluids*, 14(7):2043–2051, 2002.

[101] George Ilhwan Park and Parviz Moin. Space-time characteristics of wall-pressure and wall shear-stress fluctuations in wall-modeled large eddy simulation. *Physical review fluids*, 1(2):024404, 2016.

[102] Jungil Lee, Minjeong Cho, and Haecheon Choi. Large eddy simulations of turbulent channel and boundary layer flows at high reynolds number with mean wall shear stress boundary condition. *Physics of Fluids*, 25(11):110808, 2013.

[103] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[104] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[105] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[106] Claudio Filipi Gonçalves Dos Santos and João Paulo Papa. Avoiding overfitting: A survey on regularization methods for convolutional neural networks. *ACM Computing Surveys (CSUR)*, 54(10s):1–25, 2022.

[107] Sanjiva K Lele. Compact finite difference schemes with spectral-like resolution. *Journal of computational physics*, 103(1):16–42, 1992.

[108] John Kim, Parviz Moin, and Robert Moser. Turbulence statistics in fully developed channel flow at low reynolds number. *Journal of fluid mechanics*, 177:133–166, 1987.

[109] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[110] Samet Akçay, Amir Atapour-Abarghouei, and Toby P Breckon. Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[111] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[112] Masaki Morimoto, Kai Fukami, Kai Zhang, Aditya G Nair, and Koji Fukagata. Convolutional neural networks for fluid flow analysis: toward effective metamodeling and low dimensionalization. *Theoretical and Computational Fluid Dynamics*, 35(5):633–658, 2021.

[113] Agnieszka Mikołajczyk and Michał Grochowski. Data augmentation for improving deep learning in image classification problem. In *2018 international interdisciplinary PhD workshop (IIPhDW)*, pages 117–122. IEEE, 2018.

[114] Masaki Morimoto, Kai Fukami, Kai Zhang, and Koji Fukagata. Generalization techniques of neural networks for fluid flow estimation. *Neural Computing and Applications*, 34(5):3647–3669, 2022.

[115] Takeo Kajishima and Kunihiko Taira. *Computational fluid dynamics: incompressible turbulent flows*. Springer, 2016.

[116] John Kim and Parviz Moin. Application of a fractional-step method to incompressible navier-stokes equations. *Journal of computational physics*, 59(2):308–323, 1985.

[117] Datta V Gaitonde and Miguel R Visbal. Pade-plusmn;-type higher-order boundary filters for the navier-stokes equations. *AIAA journal*, 38(11):2103–2112, 2000.

[118] Jinhee Jeong and Fazle Hussain. On the identification of a vortex. *Journal of fluid mechanics*, 285:69–94, 1995.

[119] R Cucitore, M Quadrio, and A Baron. On the effectiveness and limitations of local criteria for the identification of a vortex. *European Journal of Mechanics-B/Fluids*, 18(2):261–282, 1999.

[120] Hiroyuki Abe, Hiroshi Kawamura, and Yuichi Matsuo. Direct numerical simulation of a fully developed turbulent channel flow with respect to the reynolds number dependence. *J. Fluids Eng.*, 123(2):382–393, 2001.

[121] J Graham, K Kanov, XIA Yang, M Lee, N Malaya, CC Lalescu, R Burns, G Eyink, A Szalay, RD Moser, et al. A web services accessible database of turbulent channel flow and its use for testing a new integral wall model for les. *Journal of Turbulence*, 17(2):181–215, 2016.

[122] Myoungkyu Lee and Robert D Moser. Direct numerical simulation of turbulent channel flow up to. *Journal of fluid mechanics*, 774:395–415, 2015.

[123] Johns Hopkins turbulence database. http://turbulence.pha.jhu.edu.

[124] Fernando Porté-Agel, Charles Meneveau, and Marc B Parlange. A scale-dependent dynamic model for large-eddy simulation: application to a neutral atmospheric boundary layer. *Journal of Fluid Mechanics*, 415:261–284, 2000.

[125] Elie Bou-Zeid, Charles Meneveau, and Marc Parlange. A scale-dependent lagrangian dynamic model for large eddy simulation of complex turbulent flows. *Physics of fluids*, 17(2), 2005.

[126] Rob Stoll and Fernando Porté-Agel. Dynamic subgrid-scale models for momentum and scalar fluxes in large-eddy simulations of neutrally stratified atmospheric boundary layers over heterogeneous terrain. *Water Resources Research*, 42(1), 2006.

[127] Wybe Rozema, Hyun J Bae, Parviz Moin, and Roel Verstappen. Minimum-dissipation models for large-eddy simulation. *Physics of Fluids*, 27(8), 2015.

Acknowledgements

First of all, I would like to thank my advisor, Prof. Yuji Hattori for his invaluable support and guidance throughout my research and PhD studies. The advice he gave me, and his expertise, patience, and encouragement helped me a lot to concentrate on my research while enjoying it.

I am deeply grateful to my committee members, Prof. Satoru Yamamoto, Prof. Jun Ishimoto, and Prof. Soshi Kawai for their suggestions and comments on this dissertation. Especially, I would like to thank Prof. Kawai for his helpful advice and discussions.

I am thankful to the associate professor of our lab, Prof. Makoto Hirota for his helpful comments in our lab seminars. I also thank all our lab members for always being kind and helpful.

I am grateful to my family, my parents and my sister, and my grandparents for being supportive and helpful during this journey. I would also like to thank my friends for their support during my PhD studies.

The computations in this study were done with the Institute of Fluid Science's (IFS) supercomputing system, Tohoku University. I would like to thank all people related to the system.

Finally, I would like to thank the Tohoku University Fellowship Foundation Program for their financial support.

List of published papers

Golsa Tabe Jamaat (C0ID4502)

Tohoku university, Graduate school of information sciences

Department of applied information sciences

[Journal papers]

1. Golsa Tabe Jamaat and Yuji Hattori. "A priori assessment of nonlocal data-driven wall modeling in large eddy simulation." Physics of Fluids 35.5 (2023). (https://doi.org/10.1063/5.0146770)
2. Golsa Tabe Jamaat and Yuji Hattori. "Development of subgrid-scale model for LES of Burgers turbulence with large filter size." Physics of Fluids 34.4 (2022): 045120. (https://doi.org/10.1063/5.0087761)
3. Golsa Tabe Jamaat, Yuji Hattori, and Soshi Kawai. "A posteriori study of wall modeling in LES using a nonlocal data-driven approach" (in preparation)

[International conferences (proceedings)]

1. Golsa Tabe Jamaat and Yuji Hattori. "Investigating a non-local data-driven approach for wall modeling in large eddy simulation." (To be presented in the 18th International Conference on Flow Dynamics, 2023)
2. Golsa Tabe Jamaat and Yuji Hattori. "A data-driven approach using CNN for wall modeling in Large Eddy Simulation." Bulletin of the American Physical Society (2022).
3. Golsa Tabe Jamaat and Yuji Hattori. "Searching for a wall model in LES using a data-driven approach." Proceeding of 19th International Conference on Flow Dynamics (2022).
4. Golsa Tabe Jamaat and Yuji Hattori. "Searching for subgrid-scale model for Burgers equation using neural network." Proceeding of 18th International Conference on Flow Dynamics (2021).
5. Golsa Tabe Jamaat and Yuji Hattori. "Investigation of the effect of the forcing function on the Burgers equation." Proceeding of 17th International Conference on Flow Dynamics (2020).

[Domestic conferences (proceedings)]

1. Golsa Tabe Jamaat and Yuji Hattori. “A posteriori test of convolutional-neural-network wall model for large eddy simulation.” (To be presented in the Japan Society of Fluid Mechanics Annual Meeting, 2023).

2. Golsa Tabe Jamaat and Yuji Hattori. “A data-driven wall model for LES of turbulent channel flow.” Japan Society of Fluid Mechanics Annual Meeting (2022).

3. Golsa Tabe Jamaat and Yuji Hattori. “Subgrid-scale modeling using a data-driven approach.” Japan Society of Fluid Mechanics Annual Meeting (2021).