

# Convolutional-network models to predict wall-bounded turbulence from wall quantities

Luca Guastoni<sup>1,2,†</sup>, Alejandro Güemes<sup>3</sup>, Andrea Ianiro<sup>3</sup>, Stefano Discetti<sup>3</sup>, Philipp Schlatter<sup>1,2</sup>, Hossein Azizpour<sup>2,4</sup> and Ricardo Vinuesa<sup>1,2</sup>

<sup>1</sup>SimEx/FLOW, Engineering Mechanics, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden

<sup>2</sup>Swedish e-Science Research Centre (SeRC), SE-100 44 Stockholm, Sweden

<sup>3</sup>Aerospace Engineering Research Group, Universidad Carlos III de Madrid, 28911 Leganés, Spain

<sup>4</sup>Division of Robotics, Perception, and Learning, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden

(Received 22 June 2020; revised 30 August 2021; accepted 15 September 2021)

Two models based on convolutional neural networks are trained to predict the two-dimensional instantaneous velocity-fluctuation fields at different wall-normal locations in a turbulent open-channel flow, using the wall-shear-stress components and the wall pressure as inputs. The first model is a fully convolutional neural network (FCN) which directly predicts the fluctuations, while the second one reconstructs the flow fields using a linear combination of orthonormal basis functions, obtained through proper orthogonal decomposition (POD), and is hence named FCN-POD. Both models are trained using data from direct numerical simulations at friction Reynolds numbers  $Re_\tau = 180$  and 550. Being able to predict the nonlinear interactions in the flow, both models show better predictions than the extended proper orthogonal decomposition (EPOD), which establishes a linear relation between the input and output fields. The performance of the models is compared based on predictions of the instantaneous fluctuation fields, turbulence statistics and power-spectral densities. FCN exhibits the best predictions closer to the wall, whereas FCN-POD provides better predictions at larger wall-normal distances. We also assessed the feasibility of transfer learning for the FCN model, using the model parameters learned from the  $Re_\tau = 180$  dataset to initialize those of the model that is trained on the  $Re_\tau = 550$  dataset. After training the initialized model at the new  $Re_\tau$ , our results indicate the possibility of matching the reference-model performance up to  $y^+ = 50$ , with 50% and 25% of the original training data. We expect that these non-intrusive sensing models will play an important role in applications related to closed-loop control of wall-bounded turbulence.

**Key words:** turbulence simulation

† Email address for correspondence: [guastoni@mech.kth.se](mailto:guastoni@mech.kth.se)

© The Author(s), 2021. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted re-use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. Introduction

In this work, we assess the potential of deep neural networks (DNNs, see LeCun, Bengio & Hinton 2015) to perform non-intrusive sensing, that consists of using measurable quantities in a fluid flow to reconstruct its behaviour in another location in the domain or to predict its dynamics in the future, without using probes that affect the flow itself. For instance, it is possible to accurately measure time-resolved quantities at the wall, such as the wall-shear stress or the pressure, and then correlate these measurements with the flow farther away. A reliable flow estimation is typically a prerequisite for closed-loop control applications, where the actuation is applied with the aim of suppressing the effect of certain structures in the flow (Choi, Moin & Kim 1994). In order to effectively perform closed-loop control it is necessary to monitor the instantaneous state of the flow so as to devise the best way to affect it, however, characterizing the flow state can be extremely challenging, particularly at very high Reynolds numbers where the near-wall structures become progressively smaller.

Before the appearance of DNNs, flow-field predictions were performed mainly through linear methods. Among them, the linear stochastic estimation (LSE) introduced by Adrian (1988) stands out. Recently Suzuki & Hasegawa (2017) and Encinar & Jiménez (2019) have used LSE to reconstruct the velocity field on a wall-parallel plane in a turbulent channel flow employing wall measurements. The latter study showed that LSE can only reconstruct the large wall-attached eddies in the outer part of the logarithmic region. An extension of the LSE method in the spectral domain (Tinney *et al.* 2006) was shown to be more suitable for noisy predictions in turbulent flows. More recently, Baars & Tinney (2014) proposed a method based on proper orthogonal decomposition (POD) to improve the spectral-LSE approach. Borée (2003) reported the possibility of projecting a synchronized field on the POD temporal modes of another quantity; this method is known as extended POD (EPOD). The correlation matrix between the temporal POD coefficients of two given quantities can be used to predict one based on the other. The work of Borée (2003) proved EPOD to be equivalent to LSE when all modes are retained in the reconstruction. Using remote probes, EPOD has been used to provide predictions in turbulent jets (Tinney, Ukeiley & Glauser 2008), channel flows (Discetti, Raiola & Ianiro 2018), pipe flows (Discetti *et al.* 2019) and wall-mounted obstacles (Bourgeois, Noack & Martinuzzi 2013; Hosseini, Martinuzzi & Noack 2016). Note, however, that, in the latter work, quadratic terms are included in the model of the POD coefficient dynamics, hinting at the need of nonlinear estimation even for a relatively simple, predominantly oscillatory flow. In this regard, Sasaki *et al.* (2019) recently assessed the capabilities of both linear and nonlinear transfer functions with single and multiple inputs to provide turbulent-flow predictions. They documented a significant improvement in the predictions when the transfer functions were designed to account for nonlinear interactions between the inputs and the flow field. The improved prediction capabilities of nonlinear methods over linear ones were also reported by Mokhasi, Rempfer & Kandala (2009) and Nair & Goza (2020).

DNNs are nonlinear models that have found application in many research areas (Jean *et al.* 2016; De Fauw *et al.* 2018; Norouzzadeh *et al.* 2018; Ham, Kim & Luo 2019; Udrescu & Tegmark 2020; Vinuesa *et al.* 2020). Due to their potential applications in flow modelling, identification of turbulence features and flow control, DNNs have recently received extensive attention in the fluid-mechanics research community (Kutz 2017; Jiménez 2018; Duraisamy, Iaccarino & Xiao 2019; Brunton, Noack & Koumoutsakos 2020). Here, we provide a brief overview of the recent neural-network applications in fluid mechanics, before coming back to the flow estimation problem we investigated in this work. In the case of turbulence modelling, DNNs have been reported to improve

the results of Reynolds-averaged Navier–Stokes (Ling, Kurzawski & Templeton 2016; Wu, Xiao & Paterson 2018) models and large-eddy simulations (Beck, Flad & Munz 2019; Lapeyre *et al.* 2019; Maulik *et al.* 2019). There are also a number of ongoing efforts towards including the constraints from the Navier–Stokes equations into prediction models through the so-called physics-informed neural networks (Wang, Wu & Xiao 2017; Raissi, Perdikaris & Karniadakis 2019). Furthermore, several artificial-intelligence-based solutions have been proposed to perform optimal control on different types of flows, such as the wake behind one or more cylinders (Rabault *et al.* 2019; Raibaud *et al.* 2020). Other promising applications of machine learning to fluid mechanics include generation of inflow conditions (Fukami *et al.* 2019*b*) and extraction of flow patterns (Raissi, Yazdani & Karniadakis 2020).

DNNs have also been used in temporal prediction of dynamical systems. As an example, Srinivasan *et al.* (2019) compared the capabilities of the multi-layer perceptron (MLP, also known as a fully connected neural network) and several long–short-term memory (LSTM) networks to predict the coefficients of a low-order model for near-wall turbulence (Moehlis, Faisst & Eckhardt 2004). While the most relevant flow features are captured by both architectures, the LSTM network outperformed the MLP in terms of ability to predict turbulence statistics and the dynamics of the flow. This work has been extended by Eivazi *et al.* (2021), where the LSTM network has been compared with a Koopman-based framework which accounts for nonlinearities through external forcing. Although both approaches provide accurate predictions of the dynamical evolution of the system, the latter outperforms the LSTM in terms of time and data required for training. Similar temporal predictions of the near-wall model (Moehlis *et al.* 2004) were conducted by Pandey, Schumacher & Sreenivasan (2020) using echo state networks. Moreover, nonlinear autoregressive exogeneous networks have been used by Lozano-Durán, Bae & Encinar (2020) to exploit the relation between the temporal dynamics of the Fourier coefficients of a minimal turbulent channel flow. Their results showed accurate predictions of the bursting events in the logarithmic layer from buffer-region data. Other related work, in the context of control of the Kuramoto–Sivashinsky chaotic system, was recently conducted by Bucci *et al.* (2019). Note, however, that the use of temporal sequences implies a high computational cost to generate well-resolved temporal data. Furthermore, longer sequences require higher memory requirements in order to predict the future behaviour. For these reasons, several neural-network-based models that learn spatial relations have been proposed in the literature. Convolutional neural networks (CNNs) have become increasingly popular during the last years due to the hierarchical structure of their input (Fukushima 1980, 1988; LeCun *et al.* 1989; Lecun *et al.* 1998). For instance, Fukami, Fukagata & Taira (2019*a*, 2021); Güemes *et al.* (2021) have shown that flow fields from the laminar wake of a cylinder and a turbulent channel can be reconstructed from extremely coarse data with remarkable success. CNNs have also been used to investigate the dynamical features of the flow without *a priori* knowledge, as shown by Jagodinski, Zhu & Verma (2020).

Neural networks are mathematical models which exhibit very appealing properties, such as being universal approximators (Hornik, Stinchcombe & White 1989). This means that they can potentially represent any continuous function with the adequate model parametrization, even though there is no guarantee that it is possible to infer the value of the parameters from data sampled from the original function. Nonetheless, neural-network parameters are typically tuned through data-driven training, and as such they have been compared and used together with other data-driven methods. For instance, the relationship between POD (see Lumley 1967) and the MLP is well documented in the literature

(Bourlard & Kamp 1988; Baldi & Hornik 1989). These works showed that a MLP with a single hidden layer is equivalent to POD if a linear activation function is used. One of the early applications on a fluid-dynamics dataset was proposed by Milano & Koumoutsakos (2002), who compared the results of POD-based neural networks with linear and nonlinear functions for the prediction of near-wall velocity fields, showing that nonlinear POD has significantly better predictive capabilities. More recently, the emergence of autoencoder architectures has motivated a renewed interest in the application of neural networks for dimensionality reduction. Hinton & Salakhutdinov (2006) proposed the use of deep autoencoders to obtain a low-order representation of high-dimensional data, showing that this approach is able to retain more information than POD. It is interesting to note that this work avoids the inherent difficulty of optimizing weights in deep autoencoders by training each layer with a restricted Boltzmann machine. Murata, Fukami & Fukagata (2020) used an autoencoder with convolutional layers to obtain a low-order representation of the flow around a cylinder. Their results suggest that CNN autoencoders with linear activation functions reproduce the same dimensionality reduction as POD, while the use of nonlinear activation functions improves the reconstruction performance. On a related note, flow reconstruction based on shallow neural networks was studied by Erichson *et al.* (2020) in several fluid-mechanics examples.

This work is not the first in which neural networks are used to perform non-intrusive sensing in wall-bounded flows: in a seminal study over 20 years ago, Lee *et al.* (1997) tested a two-layer neural network (with a single nonlinear activation function) to predict the wall actuation, based on the wall-shear-stress components, in order to reduce the drag at the wall. Guzmán Iñigo, Sipp & Schmid (2014) predicted the velocity field in a transitional boundary-layer flow using wall-shear measurements, by projecting the velocity fields on a POD basis and using a dynamic observer to predict the dynamics of the flow. More recently, Kim & Lee (2020) used the two wall-shear-stress components to predict the instantaneous wall-normal heat flux with satisfactory results. The same wall information was used by Guastoni *et al.* (2020) to predict the instantaneous streamwise flow fields at several wall-normal positions using convolutional networks. Their results show that these neural networks provide better predictions than linear methods (see below) in terms of instantaneous predictions and second-order statistics. The predictions were limited to one velocity component and to a low Reynolds number. In this work, all the velocity components are predicted and a higher  $Re$  is investigated as well. Furthermore, in the work by Güemes, Discetti & Ianiro (2019) the information of the most-energetic scales was encoded into a POD basis, and a CNN was used to predict that information at different wall-normal locations from streamwise wall-shear-stress measurements. Their results demonstrated that CNNs can significantly outperform linear methods in the prediction of POD time coefficients for low-order reconstruction of the velocity fields. Their convolutional network would predict only one POD temporal coefficient at a time, in this work all the coefficients are computed at the same time thanks to the implementation of an improved network architecture.

DNNs perform best when training and test data are taken from the same distribution, i.e. for the same flow and at the same Reynolds number in our case. However, in a real-world application the flow conditions will be continuously varying and/or it might be unfeasible to perform a full training at exactly the same conditions. If the flow behaviour at the Reynolds number of interest is roughly the same as the training Reynolds number and if the neural network can successfully approximate it, then the model should be able to perform consistently across the different Reynolds numbers, as shown in active flow control applications by Tang *et al.* (2020). In order to improve the performance at a different Reynolds number, it would then be possible to exploit initial training at

a certain flow condition and transfer this knowledge to another one. Such knowledge transfer could reduce significantly the amount of data needed for training and improve the network applicability for industrial applications. Transfer learning (Pan & Yang 2009) is the suitable learning framework to address this issue, and it is discussed in detail below.

The methods proposed by Guastoni *et al.* (2020) and Güemes *et al.* (2019), henceforth referred to as fully convolutional network (FCN) and FCN-POD respectively, are extended in the present study. Both models are able to provide a nonlinear characterization of the relation between wall features and the flow on wall-parallel planes. The purpose of this work is to provide a detailed comparison of the two aforementioned nonlinear methods regarding their capabilities in predicting turbulent flow fields from wall information. Their improvement over linear methods is measured using EPOD as a reference. Furthermore, transfer learning was applied to the FCN approach with the purpose of evaluating to what extent a network trained at one Reynolds number can be used at a different one.

The remainder of this article is organized as follows. Section 2.1 describes the numerical databases used for training and testing the neural networks and § 2.2 provides a brief comparison between the considered models. The FCN and FCN-POD methods are further detailed in § 2.4 and § 2.5, respectively, while EPOD is discussed in § 2.3. Results from the considered prediction methods are presented and compared in § 3, including instantaneous fields in § 3.1, second-order statistics in § 3.2 and spectra in § 3.3. Furthermore, an assessment of transfer learning between different Reynolds numbers is presented in § 4. Finally, the main conclusions of the work are presented in § 5. Two appendices are provided containing additional information about the training of the neural-network-based models and regarding the predicted instantaneous flow fields.

## 2. Methodology

### 2.1. Datasets

All the DNN variants in this study have been trained using the data generated from direct numerical simulations (DNS) of a turbulent open-channel flow. Periodic boundary conditions are imposed in the  $x$ - and  $z$ -directions (which are the streamwise and spanwise coordinates, respectively), and a no-slip condition is applied at the lower boundary ( $y = 0$ , where  $y$  is the wall-normal coordinate). Differently from a standard channel-flow simulation, a symmetry condition is imposed at the upper boundary. In standard channel flows, the wall-attached coherent structures may extend beyond the channel centreline, thus affecting the other wall (Lozano-Durán, Flores & Jiménez 2012). On the other hand, in open-channel flows there is no upper wall. This makes the simulation more suitable to investigate to which extent the neural networks are able to learn the dynamics of near-wall turbulence, since the interaction of the large scales with both walls is not present.

The simulation is performed using the pseudo-spectral code SIMSON (Chevalier *et al.* 2007) with constant mass flow rate, in a domain  $\Omega = L_x \times L_y \times L_z = 4\pi h \times h \times 2\pi h$  (where  $h$  is the channel height), as shown in figure 1. Two friction Reynolds numbers  $Re_\tau$  (based on  $h$  and the friction velocity  $u_\tau = \sqrt{\tau_w/\rho}$ , where  $\tau_w$  is the wall-shear stress and  $\rho$  is the fluid density) are considered, as summarized in table 1. The flow field is represented with  $N_y$  Chebyshev modes in the wall-normal direction and with  $N_x$  and  $N_z$  Fourier modes in the streamwise and spanwise directions, respectively. The instantaneous fields are obtained at constant time intervals from the time-advancing scheme, which is a second-order Crank–Nicolson scheme for the linear terms and a third-order Runge–Kutta method for the nonlinear terms. Dealiasing using a standard 3/2 rule is employed in the wall-parallel Fourier directions. The velocity fields to be used as ground truth for



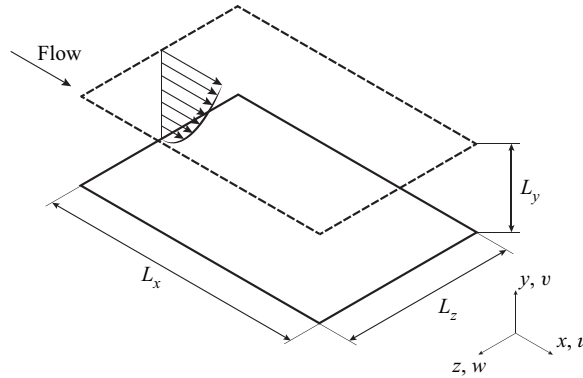


Figure 1. Computational domain and frame of reference for the DNS of the turbulent open channel considered in this study.

$Re_\tau$	$N_x \times N_z \times N_y$	No. fields	$\Delta t_{s,train}^+$	$T_{train}^+$	Test fields	$\Delta t_{s,test}^+$	$T_{test}^+$
180	$192 \times 192 \times 65$	50 400	5.08	$\approx 2.56 \times 10^5$	3125	1.69	$\approx 5.3 \times 10^3$
550	$512 \times 512 \times 193$	19 920	1.49	$\approx 2.97 \times 10^4$	3320	1.49	$\approx 5 \times 10^3$

Table 1. Description of the DNS datasets used for computing the EPOD and training/testing the CNN-based models. The number of fields refers to the number of training and validation fields.

training and testing are sampled at the following inner-scaled wall-normal coordinates:  $y^+ = 15, 30, 50$  and  $100$ . Note that ‘+’ denotes viscous scaling, i.e. in terms of the friction velocity  $u_\tau$  or the viscous length  $\ell^* = \nu/u_\tau$  (where  $\nu$  is the fluid kinematic viscosity). A dataset is defined as a collection of samples, each consisting of the shear-stress and pressure fields at the wall as inputs, along with the corresponding velocity fields at the target wall-normal locations as outputs. The training/validation dataset at  $Re_\tau = 180$  consists of 50 400 instantaneous fields, with a sampling period of  $\Delta t_s^+ = 5.08$ . The sampling period at  $Re_\tau = 550$  is set to  $\Delta t_s^+ = 1.49$  and the training/validation dataset includes 19 920 fields. The resolution in viscous units of the individual fields is the same in both  $Re_\tau$  datasets. This is necessary to represent all the flow features; however, this also implies that the number of Fourier modes in the wall-parallel directions is higher at  $Re_\tau = 550$ , as shown in table 1, even if the domain  $\Omega$  is the same. A higher number of modes corresponds to a higher number of spatial locations in which the different quantities are sampled from the DNS and this partially compensates the lower number of fields since the two proposed neural-network-based methods act locally on the input data, as will be shown in § 2.2.

In both  $Re_\tau$  cases, the dataset is split into training and validation sets, with a ratio of 4 : 1. The training and validation sets are obtained from a group of randomly initialized simulations. With the number of samples chosen for our investigation at either  $Re_\tau$ , there is a simulation whose samples are used both for training and for validation. Note that there is no temporal overlapping between the two datasets, as the first samples from the ‘shared’ simulation are sequentially added to the training set until the required number of training samples is reached. The remaining samples are then added to the validation set. This is done to reduce the correlation between the two datasets. While using the separate groups of simulations or adding an additional time separation between the last sample

of the training set and the first sample of the validation would have further reduced the correlation, this was not enforced, since the validation error is only used during training to tune the network hyperparameters and avoid overfitting the training dataset. If the training provides satisfactory results, it is common practice to use both training and validation sets for the training of the final models. This was not implemented in our case, but it allows the model to make use of all the available data to improve its performance. The validation error may be significantly different from the one computed during testing, hence all the comparisons in this work are based on the error computed on the test dataset. The predictions used to assess the performance of the trained models were obtained from additional simulations. This is done both at  $Re_\tau = 180$  and  $Re_\tau = 550$ , and it is necessary to ensure that the training and test datasets are completely uncorrelated, both in space and time. Test samples were taken from simulations initialized with a random seed, different from that of the training-data simulation. The correlation between training simulations and between training and test simulations was checked with the cross-correlations  $\rho_{ij}(h)$ , defined as (Makarashvili *et al.* 2017)

$$\rho_{ij}(h) = \frac{\gamma_{ij}(h)}{\sqrt{\gamma_i(0)\gamma_j(0)}}, \quad (2.1)$$

where  $h$  is the time lag and  $\gamma_{ij}(h) = \mathbb{E}[(x_{i,t+h} - \mu_i)(x_{j,t} - \mu_j)]$  are the cross-covariances.  $i$  and  $j$  refer to the two simulations and  $x$  is the considered quantity, in this case one of the velocity components at a given wall-normal location. By computing  $\rho_{ij}(0)$ , it was possible to verify that the simulations had run sufficiently long to ensure that they are overall statistically uncorrelated, before starting the sampling to build the datasets.

The size of the test dataset (more than 3000 fields for both  $Re_\tau$ ) is sufficient to achieve convergence of the turbulence statistics from the predicted flow, and then these are compared with the reference values from the DNS.

## 2.2. Summary of the methods

In this study we consider three different data-driven methods to estimate the instantaneous two-dimensional fields of the velocity fluctuations, at a given wall-normal distance. The velocity-fluctuation fields are obtained from the sampled DNS fields by subtracting the mean flow, averaged in time and in the two spatially homogeneous directions. Two models (FCN and FCN-POD) are introduced and compared with the EPOD method, which is used as a baseline model, highlighting the capabilities (and limitations) of linear estimation. Despite using the same input data and predicting the same output quantities, the three models leverage different tools in order to extract the information from the inputs and estimate the outputs. Here, we briefly review the differences and similarities between the models, each method is described in detail in the dedicated sections.

The three methods mainly differ when it comes to the use of POD (see Lumley 1967) on the input and output data, as well as the presence of a neural network in the prediction model. These differences are summarized in table 2, but it is important to note that the use of these mathematical tools can substantially differ from one method to the other, as detailed later. EPOD and FCN-POD use POD, which has the advantage of allowing us to filter out the noise content represented by small and uncorrelated scales, taking advantage of the energy optimality of POD modes. Additionally, spatial and temporal dynamics are separated.

	POD (input)	POD (output)	Nonlinear mapping (neural network)
EPOD	✓	✓	—
FCN-POD	—	✓	✓
FCN	—	—	✓

Table 2. Summary of the properties from each method.

EPOD decomposes both the input and the output: the flow field at a certain wall-normal distance is reconstructed as a linear combination of orthogonal modes  $\phi_i(\mathbf{x})$

$$\mathbf{u}(\mathbf{x}, t) \approx \sum_{i=1}^{N_m} \psi_i(t) \sigma_i \phi_i(\mathbf{x}), \tag{2.2}$$

where  $N_m$  is the total number of POD modes,  $\psi_i(t)$  is the temporal POD coefficient corresponding to mode  $i$  and  $\sigma_i$  is its corresponding root-squared energy contribution. While the orthogonal modes are computed from a POD of the training dataset, the temporal coefficients are estimated by decomposing input wall quantities using POD and assuming a linear correlation between the known temporal modes of the wall quantities  $\psi_{w,i}(t)$  and the temporal modes  $\psi_i(t)$  of the output flow field of the test dataset. Further details are provided in § 2.3. On the other hand, FCN-POD applies POD only on the output flow fields, using the spatial modes from the training set and predicting the corresponding temporal coefficients using a neural network. This is a development of the model used by Güemes *et al.* (2019), note, however, that, in that study, the domain employed and reconstructed provided a compact POD eigenspectrum, here, the availability of a larger domain in the streamwise and spanwise directions spreads the energy content over a wider set of POD modes. This makes the predictions of temporal coefficients more difficult, especially those associated with the least energetic modes. To address this issue, the large instantaneous flow fields were subdivided into  $N_{s_x} \times N_{s_z}$  smaller regions (henceforth referred to as subdomains). The neural network predicts the temporal coefficients for all subdomains at the same time.

The FCN and FCN-POD models consider the instantaneous two-dimensional fields of the streamwise and spanwise wall-shear-stress components and of the wall pressure as inputs. In the physical coordinates representation of these fields, the presence of coherent features motivates the use of convolutional layers in our neural-network models to process the information. In these layers, the inputs are processed at the same time, hence a convolution in three dimensions is performed and it is defined as

$$F_{i,j} = \sum_l \sum_m \sum_n I_{l-m,j-n,l} K_{m,n,l}, \tag{2.3}$$

following Goodfellow, Bengio & Courville (2016), where  $\mathbf{I} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$  is the input,  $\mathbf{K} \in \mathbb{R}^{k_1 \times k_2 \times k_3}$  is the so-called kernel (or filter) containing the learnable parameters of the layer and the transformed output  $\mathbf{F}$  is the feature map. Note that we consider  $d_3 = k_3$ , meaning that the resulting feature map is two-dimensional. Multiple feature maps can be stacked and sequentially fed into another convolutional layer as input. This allows the next layer to combine the features individually identified in each feature map, enabling the prediction of larger and more complex features for progressively deeper convolutional networks. Note that a convolutional layer can be rewritten as a matrix multiplication,



hence it is mathematically equivalent to a fully connected layer (Ma & Lu 2017). If we assume that the input features are spatially localized, using a convolutional layer allows translational equivariance to be enforced in both periodic directions. Furthermore, since  $k_i \ll d_i \forall i \neq 3$ , the use of kernels greatly reduces the number of parameters that need to be learned during training (in comparison with fully connected MLP networks).

Both neural-network-based models use fully convolutional neural network (hence FCN in the names). This architecture is similar, but conceptually different from, CNNs, which typically have several convolutional layers followed by one or more fully connected layers (which are the building blocks of MLP networks). In CNNs, the localized information processed by the individual convolutional kernels is combined to obtain a global prediction, whereas in FCNs only convolutional layers are present and the network architecture is based on the assumption that the relation between input and output variables is spatially localized. The input region from which a single point of the output can draw information is called receptive field and it can be computed based on the network architecture, as described by Dumoulin & Visin (2016). In the FCN model, the instantaneous two-dimensional velocity fluctuations are directly predicted from the input fields by using the fully convolutional neural network. Additional details are provided in § 2.4. On the other hand, the neural network in the FCN-POD model predicts the temporal coefficients for each of the  $N_{s_x} \times N_{s_z}$  subdomains at once, as described in § 2.5.

### 2.3. Extended POD

EPOD provides a linear relation between input and output and it is the reference method for all the following comparisons. Doing so, it is possible to assess the prediction improvement with nonlinear, neural-network-based methods in the context of wall-bounded turbulence. Wall quantities at each instant can be rearranged as a row vector and used to assemble a snapshot matrix

$$\mathbf{W} = \begin{bmatrix} \left. \frac{\partial u}{\partial y} \right|_{x_1}^{t_1} & \cdots & \left. \frac{\partial u}{\partial y} \right|_{x_{N_p}}^{t_1} & \left. \frac{\partial w}{\partial y} \right|_{x_1}^{t_1} & \cdots & \left. \frac{\partial w}{\partial y} \right|_{x_{N_p}}^{t_1} & p_{x_1}^{t_1} & \cdots & p_{x_{N_p}}^{t_1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \left. \frac{\partial u}{\partial y} \right|_{x_1}^{t_{N_t}} & \cdots & \left. \frac{\partial u}{\partial y} \right|_{x_{N_p}}^{t_{N_t}} & \left. \frac{\partial w}{\partial y} \right|_{x_1}^{t_{N_t}} & \cdots & \left. \frac{\partial w}{\partial y} \right|_{x_{N_p}}^{t_{N_t}} & p_{x_1}^{t_{N_t}} & \cdots & p_{x_{N_p}}^{t_{N_t}} \end{bmatrix}. \quad (2.4)$$

Here,  $N_t$  refers to the total number of snapshots (equal to the number of instantaneous flow fields  $N_f$  for the EPOD) and  $N_p$  refers to the total number of grid points in one field. Such formulation applies for regular grids; for non-regular grids, the snapshot matrix should be adapted to take into account the area associated with each grid point. Using the method proposed by Sirovich (1987), it is possible to decompose this matrix into POD modes as

$$\mathbf{W} = \Psi_w \Sigma_w \Phi_w, \quad (2.5)$$

with  $\Psi_w \in \mathbb{R}^{N_f \times N_f}$  and  $\Phi_w \in \mathbb{R}^{N_f \times 3N_p}$  being the temporal and spatial mode matrices, respectively, and  $\Sigma_w \in \mathbb{R}^{N_f \times N_f}$  being a diagonal matrix containing the singular values. Note that we perform an economy-size decomposition, such that the size of the square matrix  $\Sigma_w$  is  $\min(N_f, 3N_p)$ . The EPOD modes (Borée 2003), corresponding to the

projection of the wall quantities on the flow-field temporal basis, are defined as

$$\mathbf{L} = \Psi_w^T \mathbf{U}, \tag{2.6}$$

where  $\mathbf{U}$  is the snapshot matrix in which the instantaneous velocity fluctuations in the three directions are rearranged

$$\mathbf{U} = \begin{bmatrix} u_{x_1}^{t_1} & \cdots & u_{x_{N_p}}^{t_1} & v_{x_1}^{t_1} & \cdots & v_{x_{N_p}}^{t_1} & w_{x_1}^{t_1} & \cdots & w_{x_{N_p}}^{t_1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ u_{x_1}^{t_{N_t}} & \cdots & u_{x_{N_p}}^{t_{N_t}} & v_{x_1}^{t_{N_t}} & \cdots & v_{x_{N_p}}^{t_{N_t}} & w_{x_1}^{t_{N_t}} & \cdots & w_{x_{N_p}}^{t_{N_t}} \end{bmatrix}. \tag{2.7}$$

If the dataset is sufficiently large to reach statistical convergence, the matrix  $\mathbf{L}$  describes the relationship between the temporal POD coefficients of a certain distribution of wall features, and those of the corresponding flow field. Once the temporal correlation matrix is known, an out-of-sample flow field  $\mathbf{u}$  can be reconstructed using  $\mathbf{L}$  and the instantaneous realization of wall features as follows:

$$\mathbf{u} = \psi_w \mathbf{L}, \tag{2.8}$$

where  $\psi_w$  is the vector containing the temporal coefficients of the wall fields used for prediction. Note that  $\psi_w$  is retrieved by projecting the out-of-sample wall field  $\mathbf{w}$  on the POD basis:  $\psi_w = \mathbf{w} \Phi_w^T \Sigma_w^{-1}$ , where  $\Phi_w^T \Sigma_w^{-1}$  is readily available from the training dataset.

An important remark is that, due to the homogeneity in the streamwise and spanwise directions and the very large number of fields for the lower Reynolds number  $Re_\tau = 180$ , the wall-based matrix  $\Sigma_w$  is ill conditioned. In fact, due to the correlation between subsequent time-resolved snapshots, the rank of  $\Sigma_w$  is smaller than  $N_f$ , which is the number of snapshots (here smaller than  $N_p$ ). A well- or ill-conditioned matrix is based on the condition number: if the condition number is large the matrix is ill conditioned, while if it is small the matrix is well conditioned. Defining the condition number as  $\kappa(\Sigma_w) = \max(\Sigma_w)/\min(\Sigma_w)$ , the large difference between the first and the last POD modes leads to an ill-conditioned matrix, which is difficult to be inverted numerically. To address this issue, a reduced-order representation of the matrix  $\Sigma_w$  is employed, using a number of modes equal to the rank of the matrix. The energy contribution of the excluded modes is zero, and thus the predictions are numerically equivalent to a full-rank estimation. Differently from the  $Re_\tau = 180$  case, the singular values matrix at the wall  $\Sigma_w$  has a lower condition number for  $Re_\tau = 550$  case, possibly because of the smaller number of snapshots that are used in this case. As a result, it is not necessary to resort to a reduce-order representation in order to invert the matrix  $\Sigma_w$ . Even when  $\Sigma_w$  has rank equal to  $N_f$ , it might be convenient to truncate the matrix  $\mathbf{L}$  (Discetti *et al.* 2018). Decomposing the flow quantities as  $\mathbf{U} = \Psi \Sigma \Phi$ , similarly to what is done for the wall quantities in (2.5), it can be observed that the product of the two matrices  $\Psi \Psi_w^T$  in (2.6) returns a unitary-norm matrix with rank equal to those of  $\Psi$  and  $\Psi_w^T$ , which are bases in the  $\mathbb{R}^{N_f}$  vector space. As a consequence, a certain  $j$ th wall mode, uncorrelated with any field mode, would not result in a corresponding null row or column. To ensure the removal of the uncorrelated content from the matrix  $\Psi \Psi_w^T$ , Discetti *et al.* (2018) proposed to set to zero all the entries of the matrix with absolute values smaller than a threshold proportional to the matrix standard deviation. This operation reduces the expressivity of the EPOD, reducing the number of modes that can be used to represent the output fields. On the other hand, for the predictions in this work, we have found a mean-squared error reduction of approximately 10 percentage points with respect to the standard EPOD procedure.

Despite this, since the EPOD is used as a benchmark for the performance of linear methods with respect to the FCN-based approaches proposed herein, the filtered EPOD by Discetti *et al.* (2018) is not included in this comparison for brevity.

#### 2.4. Fully convolutional neural-network predictions

Fully convolutional networks are commonly used in applications where the input and output domains have structural similarities. Image segmentation (Long, Shelhamer & Darrell 2015) is one such case, since the output has the same spatial dimension as the input, as in our predictions of two-dimensional flow fields. The inputs of the network are the wall-shear-stress components in the streamwise and spanwise directions, as well as the pressure at the wall. Each of the inputs of the network is normalized using the respective mean and standard deviation, as computed from the training/validation set. The predictions are performed using the same mean and standard deviation values on the test dataset inputs. The outputs are the instantaneous velocity fluctuations, denoted as  $u$ ,  $v$  and  $w$  (corresponding to the streamwise, wall-normal and spanwise velocities, respectively), at a given distance from the wall. Note that the predictions are carried out at the same time as the input fields. In our previous work (Guastoni *et al.* 2020), a similar FCN was used to predict the streamwise component of instantaneous flow fields. In the present study the predictions are extended to the wall-normal and spanwise components, however, the back-propagation algorithm that is used to train the networks works best when the error in the prediction of three outputs (i.e. the three velocity components) has a similar magnitude for all of them. Thus, the fluctuations are scaled as follows:

$$\hat{u} = u, \quad \hat{v} = v \frac{u_{rms}}{v_{rms}}, \quad \hat{w} = w \frac{u_{rms}}{w_{rms}}, \quad (2.9a-c)$$

where *rms* refers to root-mean-squared (r.m.s.) quantities. During inference (i.e. when the predictions are computed from the inputs in the test dataset), the outputs of the network are re-scaled back to their original magnitude. The network is trained to minimize the following loss function:

$$\mathcal{L}_{FCN}(\hat{\mathbf{u}}_{FCN}; \hat{\mathbf{u}}_{DNS}) = \frac{1}{N_x N_z} \sum_{i=1}^{N_x} \sum_{j=1}^{N_z} |\hat{\mathbf{u}}_{FCN}(i, j) - \hat{\mathbf{u}}_{DNS}(i, j)|^2, \quad (2.10)$$

which is the mean-squared error (MSE) between the instantaneous prediction  $\hat{\mathbf{u}}_{FCN}$  and the true velocity fluctuations  $\hat{\mathbf{u}}_{DNS}$ , as computed by the DNS and scaled in the same way as the network outputs. Training details are further discussed in Appendix A. The FCN architecture is shown in figure 2. Each convolution operation (except for the last one) is followed by batch normalization (Ioffe & Szegedy 2015) and a rectified-linear-unit (ReLU, see Nair & Hinton 2010) activation function.

The chosen inputs and outputs allow the FCN to learn only the spatial relation between the quantities at the wall and the fluctuations farther away from it. Note that it would also be possible to consider predictions in time, and in that case convolutional neural networks could be used (van den Oord *et al.* 2016) treating time as another spatial coordinate, or it would be possible to use recurrent networks, specifically designed to learn temporal sequences as we have recently shown with LSTM networks (Guastoni *et al.* 2019; Srinivasan *et al.* 2019). In both cases, the need for multiple samples in time makes the model less flexible than one that relies only on spatial correlations, both during training and testing. These models usually assume a constant sampling time for the data sequence, which might be difficult to enforce, for example if the fields are taken from a numerical

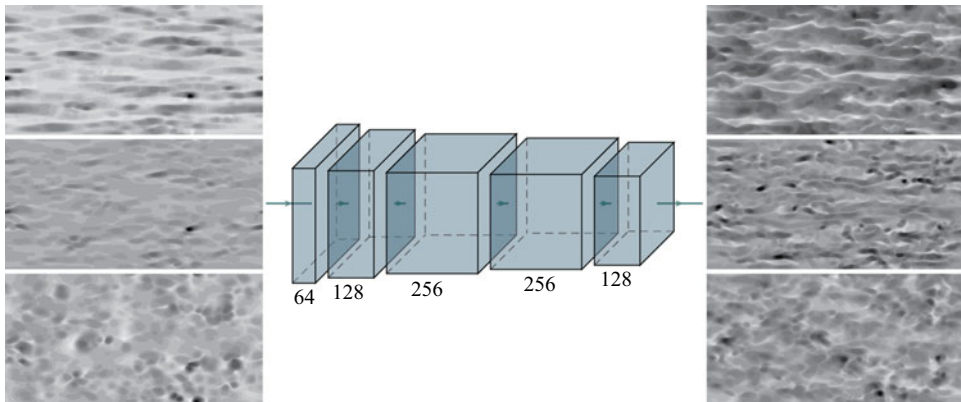


Figure 2. Schematic representation of the considered FCN architecture. The input fields are on the left (from top to bottom: streamwise wall-shear stress, spanwise wall-shear stress and wall pressure) and the outputs are on the right (from top to bottom: streamwise, wall-normal and spanwise fluctuations). The numbers indicate the number of kernels applied to each of the layers. The kernels (not represented in the figure) have size  $(5 \times 5)$  in the first convolutional layer, and  $(3 \times 3)$  in the subsequent layers. A darker colour corresponds to a higher value of the represented quantity.

simulation with an adaptive timestep. During inference, models that work with sequences would require input fields at different times to perform the prediction. On the other hand, a single input sample is sufficient for the FCN to predict the output.

Input and output fields of the FCN model are obtained from a simulation with periodic boundary conditions in the streamwise and spanwise directions. Such constraints could be added to the loss function, however, this would imply that periodicity would only be satisfied in a least-square sense and for this reason they are not used in the present study. Instead, in our implementation we are able to strictly enforce periodicity in both wall-parallel directions by leveraging the fact that the convolutional-network output is deterministic and influenced only by the local information in the receptive field. In other words, if the network receives a certain local input, the local output value will always be the same, regardless of the local position within the input field. In order to have the same values on both edges of the domain, the inputs fields are padded in the periodic directions, i.e. they are extended on both ends, using the values from the other side of the fields. The padding procedure is exemplified in [figure 3](#), which highlights how the information in the receptive field at the two ends of the domain is indeed the same. The receptive field for this architecture is  $15 \times 15$  points, hence 16 points are added to each field in both streamwise and spanwise directions. Note that this padding would determine a network output size that is slightly larger than the size of the velocity fields from the DNS, and therefore the network output is cropped to match the size of the reference flow fields. The padding involves a computational overhead due to the increased size of the input fields, however, it is important to highlight that the padding is architecture dependent and not input dependent, meaning that the input is  $\approx 17\%$  bigger with a  $192 \times 192$  field resolution (at  $Re_\tau = 180$ ), but only  $\approx 6\%$  bigger when the fields have a size of  $512 \times 512$  (at  $Re_\tau = 550$ ).

### 2.5. POD-based predictions with convolutional neural networks

The FCN-POD model is built upon the previous work by Güemes *et al.* (2019). In the present work a different neural-network architecture, a fully convolutional one, is used

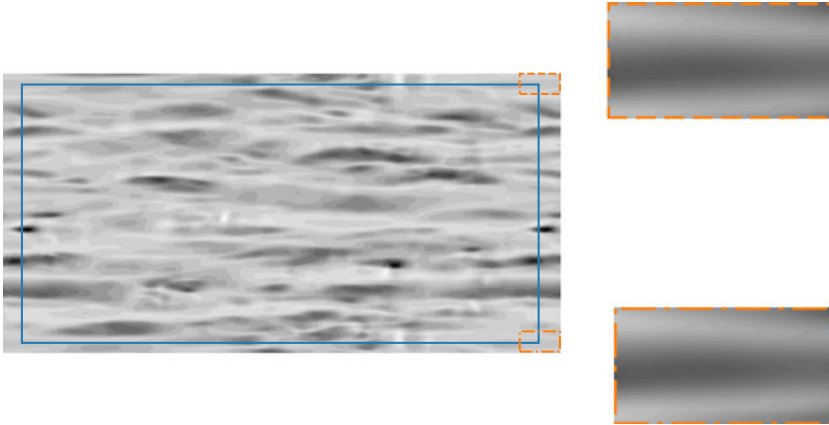


Figure 3. Example of padding of the input streamwise wall-shear-stress field. The solid blue line indicates the size of the original field. The orange boxes show the information available to the FCN for the reconstruction of a single point of the output fields, at both edges of the domain. A darker colour corresponds to a higher value of the represented quantity.

---

$Re_\tau$	$N_f$	$N_{s_x} \times N_{s_z}$	$N_t$	$N_p$	$N_m$	$N_r$
180	40 320	$12 \times 12$	5 806 080	256	768	64
550	15 936	$32 \times 32$	16 318 464	256	768	128

---

Table 3. Description of the POD settings for the subdomain decomposition. Note that only the training set has been used to compute the POD modes.

instead of a CNN. The network takes as inputs the wall-shear-stress and wall-pressure fields as sampled from the DNS to predict the temporal coefficients of the POD modes in each of the  $N_{s_x} \times N_{s_z}$  subdomains in which the output velocity-fluctuation field is divided. The advantage of this approach compared with directly decomposing the full field lies in the fact that, in these subdomains, the first POD modes contain a very large fraction of the total energy content. This is a direct consequence of including the energy of the structures larger than the domain into the first POD mode (Liu, Adrian & Hanratty 2001; Wu 2014). As a result, a lower number of modes is needed in each subdomain to reconstruct a large fraction of the total energy, as shown in figure 5(a,c). The same amount of energy would have been spread over a larger number of modes if the entire field had been considered at once. The number of subdomains in each homogeneous direction, as well as the other parameters of the POD for the two  $Re_\tau$  cases are reported in table 3. Further details regarding the subdomain choice are provided at the end of this subsection.

Similarly to what was done for EPOD in § 2.3, the snapshot matrix  $\mathbf{U}$  is assembled. The most important difference is that in this case the total number of snapshots  $N_t$  is equal to the number of instantaneous flow fields  $N_f$  times the number of subdomains per each flow field ( $N_{s_x} \times N_{s_z} \times N_f$ ), and  $N_p$  refers to the total number of grid points in one subdomain. This organization of the snapshots implies that the spatial modes are assumed to be the same for each individual subdomain. These POD spatial modes can be computed solving



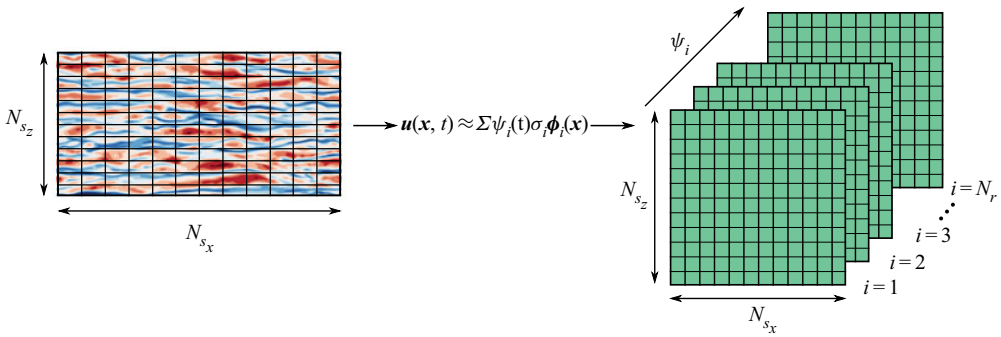


Figure 4. Schematic representation of the encoding of turbulent-flow fields into target tensors of the network, containing the temporal POD coefficients. When  $Re_\tau = 180$ , the flow field is divided into  $12 \times 12$  subdomains in the streamwise and spanwise directions respectively; a three-dimensional tensor is built, where the first two dimensions correspond to the streamwise and spanwise position of the subdomain, while the third one accounts for the number of POD modes to be predicted.

the eigenvalue problem of the spatial correlation matrix  $\mathbf{C}$  as follows:

$$\mathbf{C} = \mathbf{U}^T \mathbf{U} = \mathbf{\Phi}^T \mathbf{\Lambda} \mathbf{\Phi}, \tag{2.11}$$

where  $\mathbf{\Phi}$  is a matrix the rows of which contain the spatial POD modes, while  $\mathbf{\Lambda}$  is a diagonal matrix with elements  $\lambda_i = \sigma_i^2$ , which represent the variance content of each mode. The POD coefficients  $\psi_i(t)$  are obtained by projecting the flow fields on the spatial POD modes computed with (2.11).

The economy-size decomposition just described returns a number of POD modes  $N_m = 3N_p$ . This implies that the flow field in each subdomain is represented with  $N_m$  modes without approximation. On the other hand, the FCN-POD only predicts the POD temporal coefficients corresponding to a limited number of modes, so each turbulent-flow field is encoded using a tensor of size  $N_{s_x} \times N_{s_z} \times N_r$ , where  $N_r < N_m$  is the number POD modes to be predicted on each subdomain, as illustrated in figure 4. This tensor contains a set of  $N_r$  temporal POD coefficients for each subdomain in which the output flow field is divided. Even if the spatial modes are the same for all the subdomains, the temporal coefficient can have different values on the individual subregions, in order to provide a faithful local reconstruction of the flow. In this regard, it is important to note that the temporal coefficients for the subdomains are predicted together, and each subdomain is receiving information from the surrounding ones through the convolutional operations. This is a significant difference with the implementation in Güemes *et al.* (2019), where subdomains were not required, having considered a smaller flow field. Furthermore, in the previous cited work a different network is used to predict each mode, while in this study a single network is used to predict the full set of POD coefficients.

The FCN-POD architecture considered here blends the FCN shown in figure 2 and the network used by Güemes *et al.* (2019) (see figure 1 in that work). As in the FCN approach, each convolution operation (except for the last one) is followed by batch normalization (Ioffe & Szegedy 2015) and a ReLU (Nair & Hinton 2010) activation function. After each activation function a max pooling layer is added. Differently from what was done in the FCN approach, here and in the EPOD approach the output velocity components were not scaled before the decomposition, in order to keep the physical encoding based on the turbulent kinetic energy (TKE) of the flow. Note that by modifying the relative contribution of the velocity components to the energy norm, the modes would have been sorted based

on a norm different from the TKE. The network is trained to minimize the loss function

$$\mathcal{L}_{FCN-POD}(\psi_{POD}; \psi_{DNS}) = \frac{1}{N_{s_x} N_{s_z} N_r} \sum_{i=1}^{N_{s_x}} \sum_{j=1}^{N_{s_z}} \sum_{k=1}^{N_r} |\psi_{POD}(i, j, k) - \psi_{DNS}(i, j, k)|^2, \quad (2.12)$$

which is the MSE between the predicted and the actual POD temporal coefficients of the DNS data. Similarly to the FCN, the predictions are performed at the same instant as that of the input fields, hence the neural network does not require the knowledge of the input at previous timesteps. Parameter settings for training are detailed in [Appendix A](#).

During inference, the velocity-fluctuation fields need to be reconstructed from the POD temporal coefficients. For each subdomain, the  $N_r$  predicted POD temporal coefficients are used to reconstruct the local fluctuation field within the subdomain. This is done with the orthonormal basis functions retrieved from the training data, even at test time. The underlying assumption is ergodicity, i.e. both the training and test datasets share the same statistical features and, consequently, the same spatial modes. This requires a sufficiently large training dataset to ensure convergence of the spatial modes, which is generally ascribed to the convergence of second-order statistics. Once the velocity-fluctuation fields are reconstructed within each subdomain, they are assembled together to provide the full-domain prediction. Note that the tiling of the subdomains does not provide any guarantee of smoothness across the edges of the subdomains because of the finite number of modes that are used to reconstruct the flow and because of the prediction error in the temporal coefficients.

As mentioned earlier, dividing the output velocity-fluctuation fields in subdomains allows us to represent the flow in these subregions with a compact orthogonal basis, as shown in [figure 5\(a,c\)](#). The energy distribution reported for the two  $Re_\tau$  cases is very similar, only the energy distribution at  $y^+ = 15$  becomes slightly more compact for the low- $Re_\tau$  case. The size of the subdomains in the streamwise and spanwise directions is roughly  $h \times 0.5h$  for the  $Re_\tau = 180$  case, and  $0.4h \times 0.2h$  at  $Re_\tau = 550$ . This is smaller than the domain considered in the previous work by Güemes *et al.* (2019), where the domain had size of  $h \times h$ . The choice of the size of the subdomains is the result of a compromise between network capability of reconstructing the majority of the energy content of the flow and compression of the information. In particular, it is important to reconstruct a significant portion of the total energy using a reduced number of modes. As shown in [figure 5\(b,d\)](#), the first 64 POD modes account for 90% of the total energy at  $Re_\tau = 180$ , while 128 modes are needed at  $Re_\tau = 550$  to retain a similar amount of energy. Therefore our predictions are truncated at  $N_r = 64$  and 128 for  $Re_\tau = 180$  and 550, respectively. As larger and larger subdomains are considered, an increasing number of modes would be needed to represent the same amount of energy. This would increase the number of coefficients that have to be predicted by the FCN and since the temporal coefficients of the less-energetic modes are difficult to correlate with the wall quantities used as input, the use of larger subdomains can effectively hinder the network learning process.

### 3. Results

The predictions of the trained models are compared with the data obtained from the DNS at  $Re_\tau = 180$  and 550. All the models use the same input data and predict the same output fields, in an effort to provide a fair comparison. Other metrics, such as the number of adjustable parameters, can be difficult to compare for different models. Furthermore,

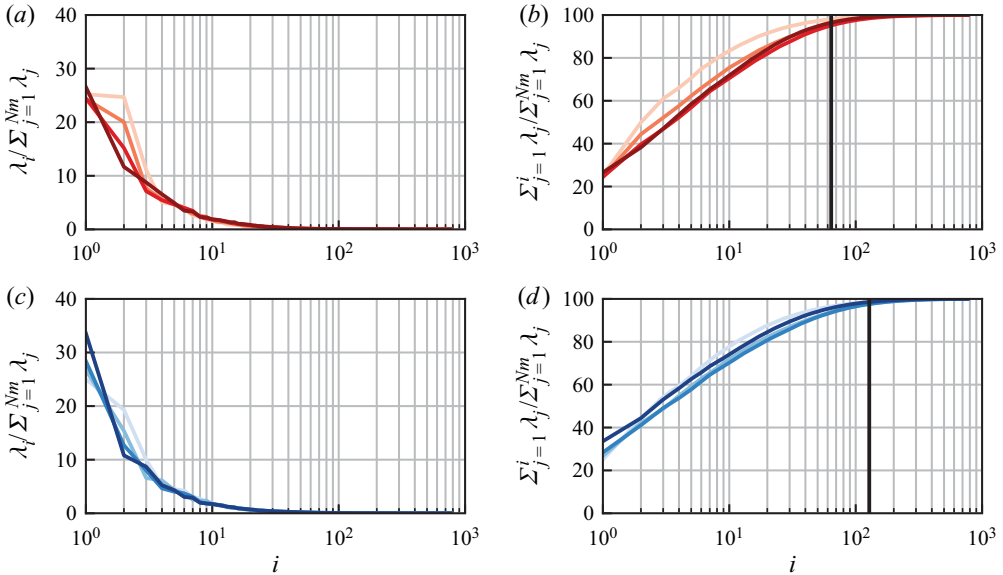


Figure 5. Distribution of (a,c) POD eigenvalues  $\lambda_i = \sigma_i^2$  (where  $i$  denotes mode number) and (b,d) cumulative eigenspectra  $\sum_{j=1}^i \lambda_j$  normalized with the cumulative sum of the eigenvalues  $\sum_{j=1}^{N_m} \lambda_j$  of each case. The colour refers to the wall-normal locations described in § 2.1, where darker colours indicate larger distance from the wall. Note that (a,b) correspond to  $Re_\tau = 180$ , while (c,d) correspond to  $Re_\tau = 550$ . The solid black vertical lines in (b,d) refer to the number of modes selected for prediction in this study.

a higher number of parameters may not translate immediately to a better performance, as illustrated by Discetti *et al.* (2018) for the EPOD method. The performance assessment is carried out first from a qualitative point of view and subsequently from a quantitative perspective, based on predictions of instantaneous fields, turbulence statistics and spectra.

### 3.1. Instantaneous predictions

The predicted fluctuation fields are first qualitatively inspected. Note that the fluctuation flow fields are the direct output of the FCN models, while in the FCN-POD models the temporal coefficients need to be processed to reconstruct the fluctuations, as outlined above. In this work, the sampling period in the simulation is fixed, however, we showed in our previous work (Guastoni *et al.* 2020) that using less correlated samples during training (i.e. larger sampling period, that corresponds to a longer sampled time for the same number of samples gathered) can effectively improve the quality of the instantaneous predictions of the FCN method, provided that the neural-network capacity is sufficient to generalize over the training dataset.

In figure 6, the predictions of an instantaneous field of streamwise velocity fluctuations based on the various methods (namely FCN, FCN-POD and EPOD) are compared with the reference DNS. The predictions of the wall-normal and spanwise fluctuations at the same instant are shown in Appendix B. At  $y^+ = 15$  all the methods provide accurate results, although the EPOD overestimates the fluctuations from the high-speed streaks. At  $y^+ = 30$  such overestimation is reduced, although EPOD does not seem to reach a level of accuracy comparable to the other two models. The CNN-based methods start to exhibit some deviations with respect to the reference at  $y^+ = 50$ , where the FCN-POD

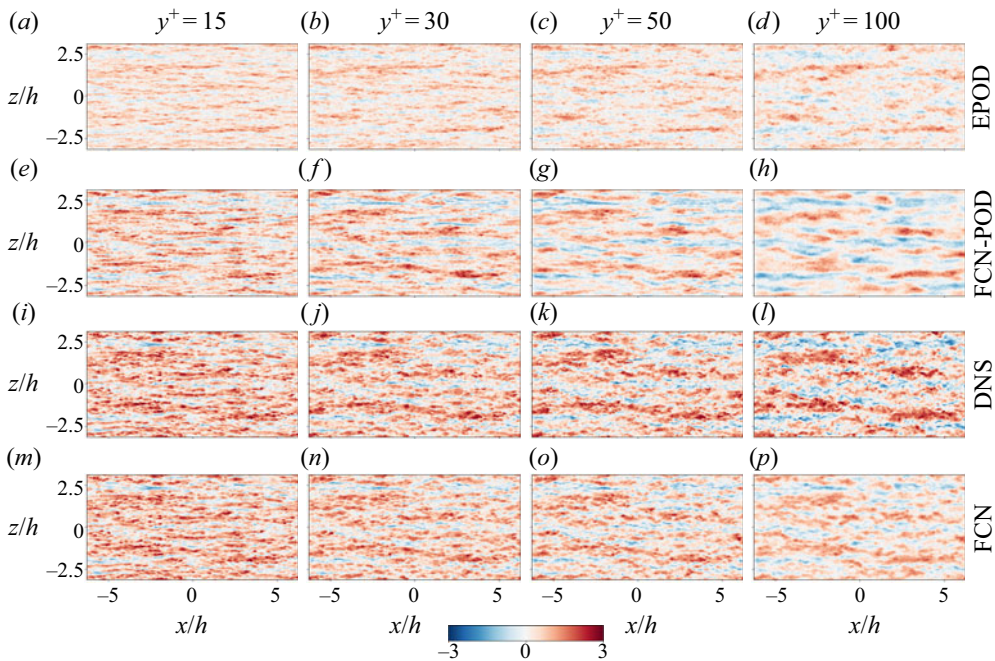


Figure 6. Comparison of the streamwise fluctuation fields at  $Re_\tau = 180$ , scaled with the corresponding  $u_{rms}$ , from EPOD (a–d), FCN-POD (e–h), reference DNS (i–l) and FCN (m–p). Results at  $y^+ = 15$  (a,e,i,m),  $y^+ = 30$  (b,f,j,n),  $y^+ = 50$  (c,g,k,o) and  $y^+ = 100$  (d,h,l,p).

field is smoother and the FCN is slightly noisier than the DNS. Farther from the wall, the footprint of the large scales at the wall (through linear superposition, see Dogan *et al.* 2019) is less pronounced, and therefore the ability of EPOD (which is a linear method) to predict the flow in this region is significantly reduced. In fact, the fields predicted through EPOD at  $y^+ > 15$  are qualitatively very similar to the DNS, although the fluctuations become increasingly attenuated at larger  $y^+$ . Furthermore, the FCN-POD method tends to merge neighbouring regions with high- or low-velocity fluctuations, predicting more elongated streak-like patterns than in the reference field. This is more evident at  $y^+ = 100$ , leading to an overestimation of the amplitude of the regions in the flow where the velocity fluctuations are higher. At this location, the FCN is not able to provide a reliable prediction of the flow field, capturing only the regions in which the magnitude of the fluctuations is higher. The corresponding structures probably have a distinct footprint at the wall, which allows the FCN to identify them.

As discussed in § 2.5, the FCN-POD approach does not guarantee flow smoothness across the edges of the subdomains. Close inspection of the predictions from the FCN-POD method reveals the edges of the subdomains at all  $y^+$ , and the tiling is more evident in the streamwise direction because of the discontinuities located at the same spanwise location, orthogonally to the main flow structures. Despite these limitations we can observe that the velocity fields are generally smooth, without steep discontinuities at the edges of the subdomains: the variations of the velocity magnitude at the edges are of the same order as the fluctuations at the corresponding wall-normal distance. These observations can be explained by the combination of two concurring elements. On one hand, the first POD modes of each subdomain contain the energy of all the structures that are larger than the individual subdomain. On the other hand, the temporal coefficients

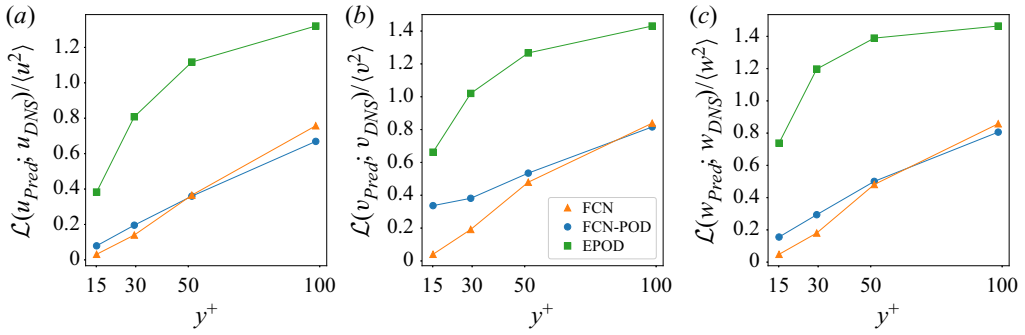


Figure 7. Mean-squared error in the instantaneous fields (scaled with the corresponding r.m.s. components) predicted by the three models at  $Re_\tau = 180$ , for the streamwise (a), wall-normal (b) and spanwise (c) velocity fluctuations.

of the POD modes are predicted for all the subdomains at the same time. These two factors allow the FCN-POD network to predict the turbulent structures larger than a single subdomain, also providing smooth transitions at the edges of the subdomains.

The qualitative observations discussed above are complemented with a quantitative assessment of the instantaneous prediction performance, by analysing the MSE  $\mathcal{L}$  between the instantaneous predictions (denoted by 'Pred') and the reference (defined for each of the fluctuations independently), as shown in figure 7. The error in the streamwise velocity fluctuations is defined as

$$\mathcal{L}(u_{Pred}; u_{DNS}) = \frac{1}{N_x N_z} \sum_{i=1}^{N_x} \sum_{j=1}^{N_z} |u_{Pred}(i, j) - u_{DNS}(i, j)|^2, \quad (3.1)$$

and similarly for the wall-normal and spanwise fluctuations. Note that the velocity-fluctuation fields need to be reconstructed from the temporal coefficients in POD-based models (FCN-POD and EPOD). Both neural-network models (FCN and FCN-POD) are trained using a stochastic algorithm and, in order to show the robustness of the optimal configuration, the statistics at each  $y^+$  are averaged over 3 different models, with different initial random weight initializations. Since the EPOD algorithm is completely deterministic, one single prediction is needed. The neural-network-based models consistently provide a lower error than the EPOD, with the FCN yielding a slightly better instantaneous performance closer to the wall than the FCN-POD approach. The gap between the two is reduced when moving away from the wall, where the prediction error of the streamwise fluctuations is approximately the same for both models at  $y^+ = 50$ , and it is slightly higher for the FCN at  $y^+ = 100$ .

The FCN architecture reported by Guastoni *et al.* (2020) would only predict the streamwise velocity component of the velocity field at the target  $y^+$ . The addition of the two other components implies that the FCN has multiple outputs that need to be optimized at the same time. We note that adding the two additional fluctuating components as outputs leads to slightly less accurate predictions with respect to those reported by Guastoni *et al.* (2020) for one single output. This is not surprising, since the capacity of the network remained unchanged, however, we tested a variation of the model architecture based on this observation, in order to have more layers dedicated to the prediction of each individual component. This network variation has a common part, identical to the original FCN up to the fourth convolutional layer, in which the weights are optimized using the information from the error gradients computed for all the outputs. The last



two convolution operations are replicated for each velocity component and the weights of these layers are updated only with the error associated with the respective output. Such a network, despite its higher capacity, provided worse predictions. A strong causal relation between the different components of the velocity (Lozano-Durán *et al.* 2020) can be a possible explanation for this result, which shows that updating all the weights with information from the three components at the same time can be beneficial for the quality of the predictions. Note that it is not trivial to design an architecture able to provide the best trade-off between single-component predictions and usage of the information from all the components, and obtaining such an architecture would require further investigation. For the FCN-POD model the multiple-component predictions were obtained as discussed by Güemes *et al.* (2019). The temporal POD coefficients can be projected on spatial POD modes involving the three velocity components, thus requiring only one output to predict the three fluctuations. The network architecture is different than the one used by Güemes *et al.* (2019), since it predicts directly all the needed time coefficients for each snapshot. While the final fully connected layer included in the network architecture by Güemes *et al.* (2019) improves the robustness of the prediction, the FCN-POD implementation used here has a much smaller number of weights, thus significantly reducing the computational cost, and retains a larger number of POD modes (and thus more energy).

Predictions of the streamwise fluctuation fields from the various methods at  $Re_\tau = 550$  are shown in figure 8, while the results for the wall-normal and spanwise components are presented in Appendix B. Despite the higher friction Reynolds number, the FCN maintains a performance similar to the one achieved for  $Re_\tau = 180$ , at all wall-normal locations. Note that the FCN has the same architecture as the lower-Reynolds-number case, i.e. it has the same number of trainable parameters, while in the case of the FCN-POD approach, the network was modified to reconstruct approximately the same amount of energy as at  $Re_\tau = 180$ . Despite the higher number of employed subdomains, the tiling is more apparent at  $Re_\tau = 550$ . The prediction performance of the FCN-POD model degrades less quickly than the FCN when moving away from the wall, however, the latter still performs better at  $y^+ = 50$ , as shown in figure 9. On the other hand, the EPOD also exhibits similar error levels as those reported for  $Re_\tau = 180$ , except at  $y^+ = 15$ , where the reconstruction of the streamwise fluctuation field is significantly worse.

### 3.1.1. Inclination of coherent structures

The coherent structures in wall-bounded turbulence are inclined (Marusic & Heuer 2007), with a slope that can be computed by finding the maximum spatial correlation  $\mathcal{R}_{ij}(\delta x)$  between the inputs at the wall (index  $i$ ) and the outputs (index  $j$ ), with  $\delta x$  representing the distance in the streamwise direction at which the correlation is computed. By including a streamwise shift in the output fields, it is possible to obtain the maximum correlation at  $\delta x = 0$ , ensuring that the footprint of the coherent structure is included in the receptive field of the output. The use of such a shift was also discussed by Sasaki *et al.* (2019) in a similar context. By considering the maximum correlation between the wall-shear stress in the streamwise direction and the streamwise velocity at a certain  $y^+$ , we obtain an angle

$$\theta_{shift} = \tan^{-1} \left( \frac{\arg \max_{\delta x} R_{11}(\delta x)}{\delta y} \right) \approx 11^\circ, \quad (3.2)$$

at  $Re_\tau = 180$ , where  $\delta y$  indicates the considered distance from the wall. The obtained value is slightly lower than previous observations (Marusic & Heuer 2007; Sasaki *et al.* 2019), possibly because of the low Reynolds number. The inclusion of the shift information has been tested using two alternative implementations: first, by modifying the target output

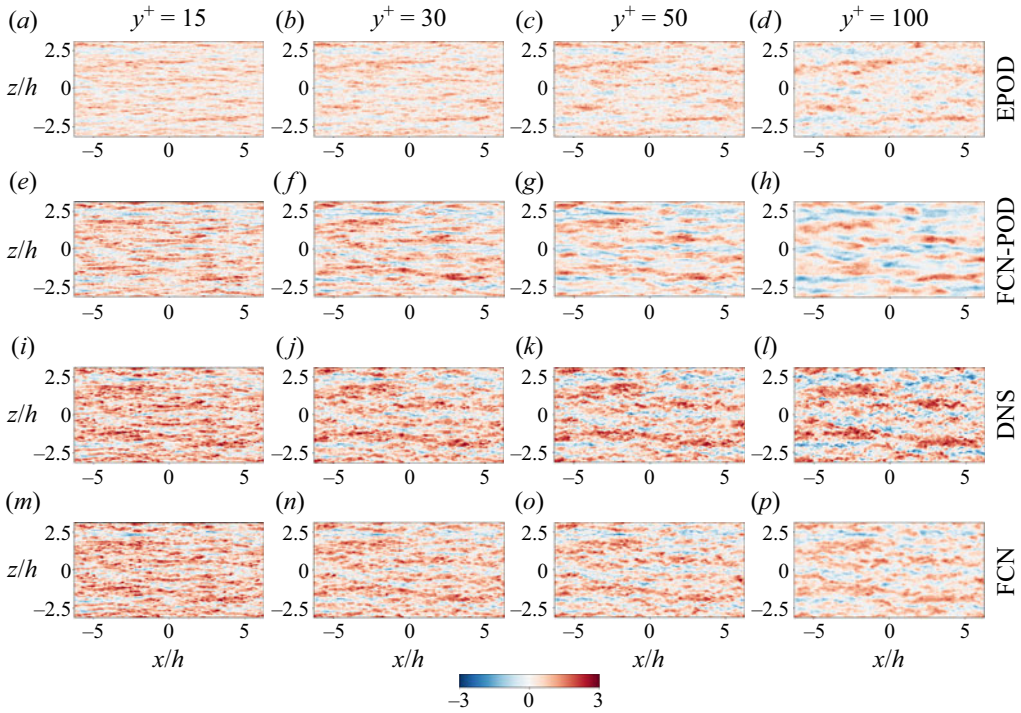


Figure 8. Comparison of the streamwise fluctuation fields at  $Re_\tau = 550$ , scaled with the corresponding  $u_{rms}$ , from EPOD (a–d), FCN-POD (e–h), reference DNS (i–l) and FCN (m–p). Results at  $y^+ = 15$  (a,e,i,m),  $y^+ = 30$  (b,f,j,n),  $y^+ = 50$  (c,g,k,o) and  $y^+ = 100$  (d,h,l,p).

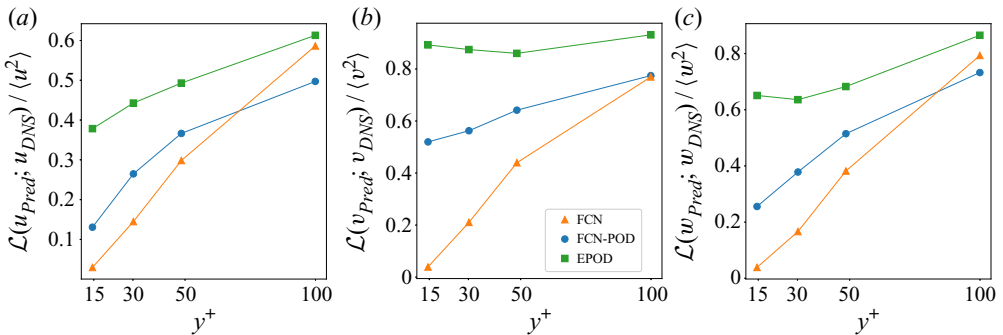


Figure 9. Mean-squared error in the instantaneous fields (scaled with the corresponding r.m.s. components) predicted by the three models at  $Re_\tau = 550$ , for the streamwise (a), wall-normal (b) and spanwise (c) velocity fluctuations.

field, i.e. considering a field that has been sampled later in the simulation, although the accuracy of the introduced shift is limited by the value of the sampling period. The second approach makes use of the periodicity of the output fields, which are translated in the streamwise direction until the maximum correlation is obtained at  $\delta x = 0$ . This approach allows us to more accurately introduce the shift, however, in this case the underlying hypothesis is that the shift is sufficiently small so that the temporal dynamics modifies the flow in a negligible manner. None of the two shift implementations provided the expected

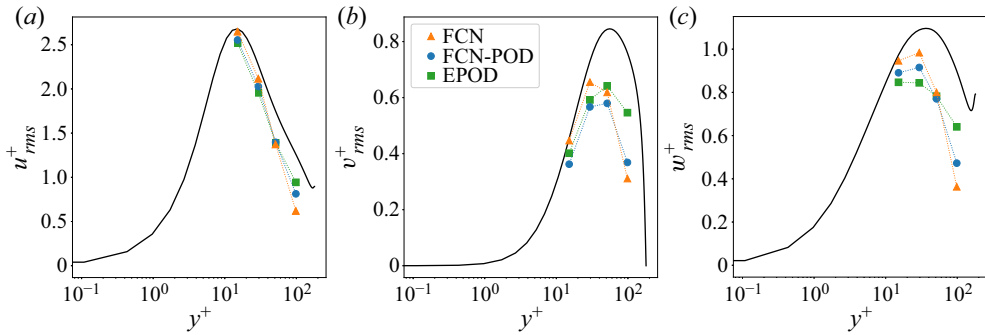


Figure 10. Comparison between the DNS (solid line) and the predictions of streamwise (a), wall-normal (b) and spanwise (c) velocity fluctuations at  $Re_\tau = 180$ .

improvement, and we observed a significant degradation of the prediction performance. These results could possibly be explained by the fact that coherent structures of different size have different inclinations, and imposing a single value is detrimental for the overall network performance, despite having chosen the angle that provides the maximum spatial correlation. Furthermore, the quality of the predictions is measured using the MSE between the prediction and the reference: this error indicator considers all wavelengths at the same time, without considering how the different wavelengths are affected by the shift. Further investigation of this aspect, also at higher  $Re_\tau$ , will be conducted in future work.

### 3.2. Predictions of turbulence statistics

By averaging over the fields obtained from the neural-network models and EPOD, it is possible to evaluate the turbulence statistics of the predicted flow. First we consider the dataset at  $Re_\tau = 180$ : the predicted r.m.s. fluctuations of the three components are shown in figure 10, together with the reference DNS profiles. The error in these statistical quantities is defined as

$$E_{rms}^+(u) = \frac{|u_{rms,Pred}^+ - u_{rms,DNS}^+|}{u_{rms,DNS}^+}, \quad (3.3)$$

for the streamwise component, and similarly for the other two components. As above, the subscripts ‘DNS’ and ‘Pred’ refer to the reference and predicted profiles, respectively. An important premise is that neither of the neural-network-based models is explicitly optimized to reproduce the statistics of the original simulation. This prevents the neural networks from learning only the average behaviour of the flow, however, the predictions may be less statistically accurate, with the aim of maximizing the instantaneous performance. Note that here we favor instantaneous performance because our motivation is to use non-intrusive sensing for closed-loop flow control.

The prediction errors in the various r.m.s. profiles are summarized in table 4, and they are averaged over the different training runs for the FCN and FCN-POD models. Note that the average is performed over the fluctuation-intensity values and not on the predictions, because that would alter the statistical properties of the predicted flow fields. The comparison of the errors from the different models shows that the statistical performance mimics the one observed for the instantaneous predictions at  $y^+ = 15$  and  $y^+ = 30$ , with

$E_{rms}^+(\cdot)$ (%)		$y^+ = 15$	$y^+ = 30$	$y^+ = 50$	$y^+ = 100$
$u$	EPOD	6.03	13.87	20.50	25.15
	FCN	1.16 ( $\pm 0.74$ )	6.79 ( $\pm 1.31$ )	21.47 ( $\pm 1.97$ )	50.82 ( $\pm 2.19$ )
	FCN-POD	4.70 ( $\pm 0.02$ )	10.70 ( $\pm 0.02$ )	20.15 ( $\pm 0.03$ )	35.46 ( $\pm 0.02$ )
$v$	EPOD	11.68	18.89	23.97	28.10
	FCN	1.74 ( $\pm 1.00$ )	10.18 ( $\pm 1.67$ )	26.66 ( $\pm 0.76$ )	59.05 ( $\pm 1.61$ )
	FCN-POD	20.29 ( $\pm 0.02$ )	22.32 ( $\pm 0.02$ )	31.32 ( $\pm 0.01$ )	51.48 ( $\pm 0.04$ )
$w$	EPOD	13.01	22.48	27.27	28.72
	FCN	2.79 ( $\pm 0.36$ )	9.65 ( $\pm 1.07$ )	25.60 ( $\pm 1.214$ )	59.59 ( $\pm 1.310$ )
	FCN-POD	8.50 ( $\pm 0.04$ )	15.95 ( $\pm 0.06$ )	28.38 ( $\pm 0.004$ )	47.42 ( $\pm 0.001$ )

Table 4. Percentage error in the prediction of the various r.m.s. fluctuations at the different wall-normal locations. Results at  $Re_\tau = 180$ .

the FCN performing better than the FCN-POD and EPOD models. Furthermore, the FCN model provides a similar performance for the fluctuations of all three velocity components, while POD-based methods are more accurate in the predictions of  $u_{rms}^+$ . This is related to the choice of not scaling the different velocity components in the FCN-POD and EPOD approaches, and the fact that near the wall the most energetic dynamical structures of the flow are in the streamwise direction. Taking into account the standard deviation in the results of the neural-network-based methods, the three models provide similar error levels at  $y^+ = 50$ . At  $y^+ = 100$  the scenario is opposite to what we observed close to the wall: the FCN exhibits the highest errors, while the EPOD provides the best results. The error in the prediction of  $u_{rms}^+$  from the FCN-POD model is between those of the two other models, while the wall-normal and spanwise intensities are closer to the errors from the FCN, due the reasons outlined above.

The statistical analysis is repeated also for the models trained at  $Re_\tau = 550$ , with the predicted r.m.s. fluctuations shown in figure 11 and the relative error with respect to the reference simulation in table 5. FCN-based models do not show a significant variation in the prediction of the streamwise fluctuations with respect to the results at  $Re_\tau = 180$ , whereas the EPOD exhibits higher errors at this Reynolds number (also in the other two fluctuating components). The FCN has a consistent behaviour also for the fluctuations in the  $y$ - and  $z$ -directions, however, the FCN-POD performs slightly worse than before, following the same trend but with higher error levels. The FCN-POD method outperforms the FCN approach only at  $y^+ = 100$ , confirming the results of the instantaneous performance at  $Re_\tau = 550$ .

### 3.3. Predictions of power-spectral density

The energetic scales present in the predicted fields, as well as their associated energy, are compared with those in the reference DNS data through spectral analysis. In figure 12 we show the pre-multiplied two-dimensional power-spectral density of the streamwise, wall-normal and spanwise fluctuations (denoted by  $\phi_{uu}$ ,  $\phi_{vv}$  and  $\phi_{ww}$ , respectively) at  $Re_\tau = 180$ , where  $\lambda_x$  and  $\lambda_z$  denote the streamwise and spanwise wavelengths, whereas  $k_x$  and  $k_z$  are the corresponding wavenumbers. These results confirm the observations made in § 3.1: at  $y^+ = 15$ , all the considered models are able to correctly predict the energy content of the flow at all wavelengths, with the FCN slightly outperforming the two POD-based approaches. Note that the FCN-POD model is able to reconstruct the energy

Wall-bounded turbulence from wall quantities

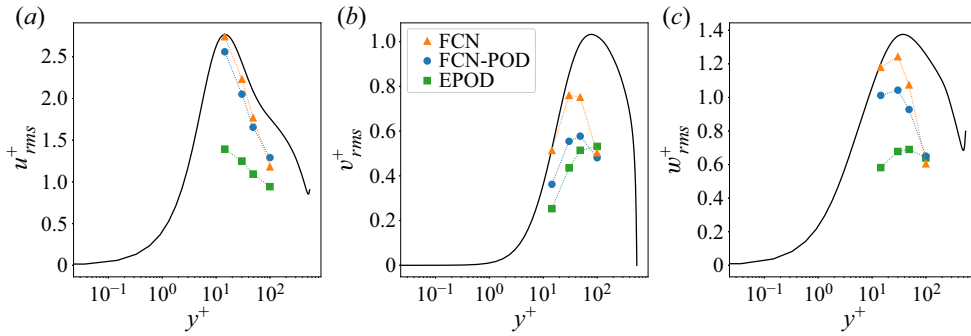


Figure 11. Comparison between the DNS (solid line) and the predictions of streamwise (a), wall-normal (b) and spanwise (c) velocity fluctuations at  $Re_\tau = 550$ .

$E_{rms}^+(\cdot)$ (%)		$y^+ = 15$	$y^+ = 30$	$y^+ = 50$	$y^+ = 100$
$u$	EPOD	49.69	48.53	47.62	46.53
	FCN	0.98 ( $\pm 0.66$ )	8.10 ( $\pm 0.62$ )	15.33 ( $\pm 0.22$ )	33.06 ( $\pm 0.30$ )
	FCN-POD	7.53 ( $\pm 0.27$ )	15.53 ( $\pm 0.39$ )	20.75 ( $\pm 0.85$ )	26.79 ( $\pm 0.36$ )
$v$	EPOD	51.57	49.13	48.13	48.07
	FCN	1.74 ( $\pm 0.11$ )	11.21 ( $\pm 1.41$ )	24.20 ( $\pm 1.38$ )	50.82 ( $\pm 0.26$ )
	FCN-POD	30.73 ( $\pm 0.06$ )	35.20 ( $\pm 0.09$ )	41.75 ( $\pm 0.10$ )	53.04 ( $\pm 0.18$ )
$w$	EPOD	51.56	50.35	49.42	49.08
	FCN	1.86 ( $\pm 0.60$ )	9.03 ( $\pm 0.31$ )	21.21 ( $\pm 1.27$ )	51.83 ( $\pm 0.38$ )
	FCN-POD	15.74 ( $\pm 0.05$ )	23.63 ( $\pm 0.07$ )	31.97 ( $\pm 0.10$ )	48.20 ( $\pm 0.24$ )

Table 5. Percentage error in the prediction of the various r.m.s. fluctuations at the different wall-normal locations. Results at  $Re_\tau = 550$ .

content of the flow at wavelengths that are longer than the size of the subdomains, proving that this is not a limiting factor for the model. However, a small jump, probably due to a lack of smoothness at the edges of the subdomains, can be observed in the streamwise wavelength for the 10 %-energy level in the wall-normal and spanwise components. These jumps are found at a wavelength  $\lambda_x^+ \approx 180$ , corresponding to the subdomain size. At  $y^+ = 30$  there is a slight energy attenuation which becomes increasingly more noticeable when the predicted flow is farther away from the wall. At  $y^+ = 100$  the POD-based methods perform better than the FCN model, a fact that can be explained by considering two concurring aspects: the first is that POD methods only predict the temporal dynamics of the system, thus the overall energy-scale distribution stored in the POD spatial modes does not need to be predicted. This allows reconstruction of more than 50 % of the flow fields, at least in the streamwise component. The second aspect is the fact that the receptive field from the FCN method, while sufficient for planes closer to the wall, is not large enough to reproduce the large scales present at larger  $y^+$ .

Figure 13 reports the reference and predicted power-spectral densities at  $Re_\tau = 550$ . As opposed to what was observed for the instantaneous predictions and the turbulence statistics, the spectra highlight the differences and similarities between the models used for the two Reynolds numbers. As noted above, the FCN architecture is the same for both  $Re_\tau$ , however, this implies that the receptive field is smaller at higher Reynolds number when it is measured in outer units. This can potentially help in the prediction of the small



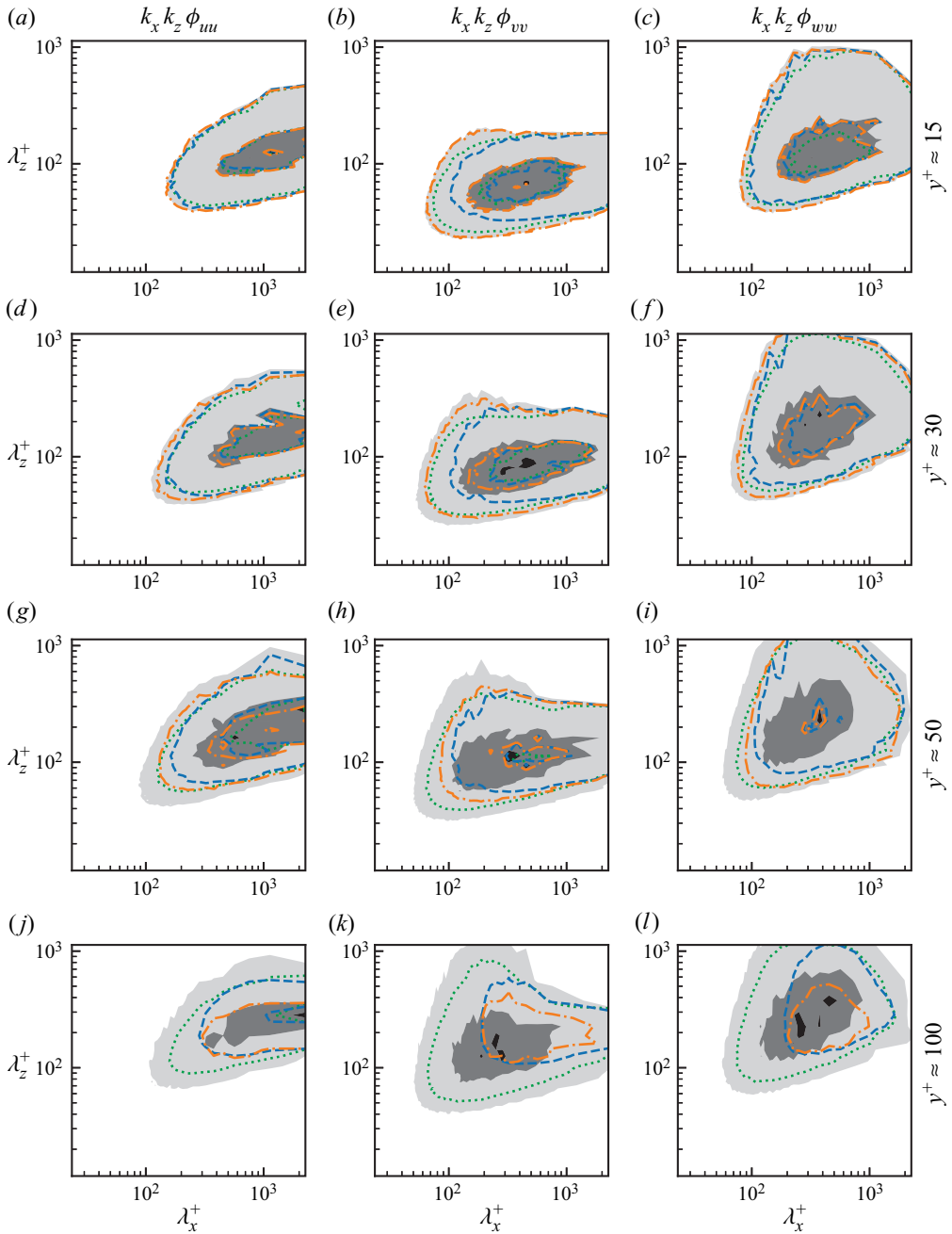


Figure 12. Pre-multiplied two-dimensional power-spectral densities for  $Re_\tau = 180$ . The contour levels contain 10%, 50% and 90% of the maximum DNS power-spectral density. Shaded contours refer to the reference data, while contour lines refer to (orange dashed dotted line) FCN, (blue dashed line) FCN-POD and (green dotted line) EPOD predictions, respectively.

Wall-bounded turbulence from wall quantities

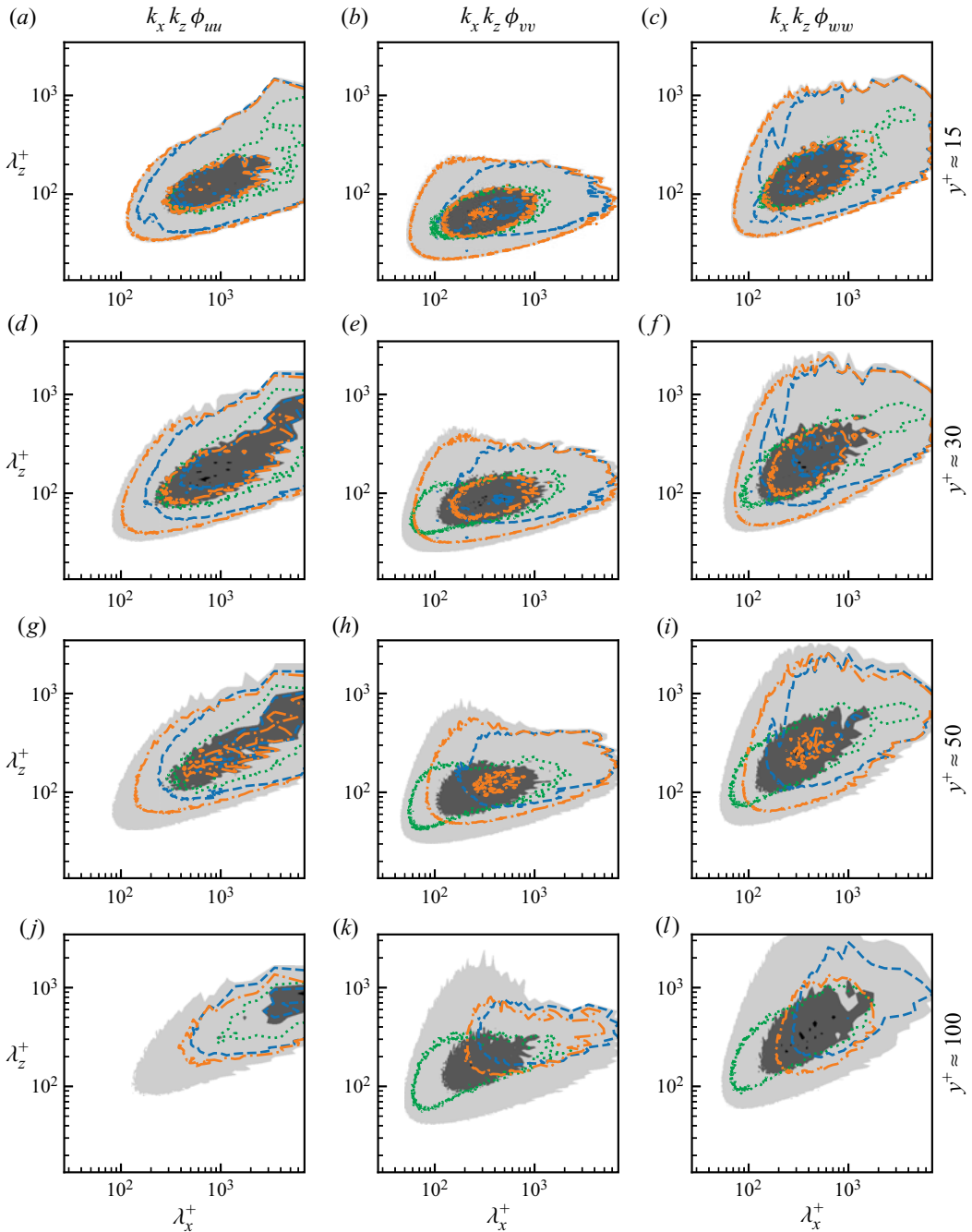


Figure 13. Pre-multiplied two-dimensional power-spectral densities for  $Re_\tau = 550$ . The contour levels contain 10%, 50% and 90% of the maximum DNS power-spectral density. Shaded contours refer to the reference data, while contour lines refer to (orange dashed dotted line) FCN, (blue dashed line) FCN-POD and (green dotted line) EPOD predictions, respectively.

scales, but it can also be detrimental for the larger scales. Note that the different size of the receptive field does not seem to affect the predicted energy content of the FCN, which shows the same trends as for the low- $Re_\tau$  case. On the other hand, the spectra of the FCN-POD with  $12 \times 12$  subdomains (the same amount as in the low- $Re_\tau$  case, not shown) exhibit spurious periodic peaks due to the tiling. This observation motivated the increased number of subdomains considered at  $Re_\tau = 550$ . As in the low- $Re_\tau$  case, the FCN-POD approach is able to reconstruct the scales larger than the subdomain size, but the contour line exhibits a small jump in the streamwise wavelength for the 10%-energy level. This jump appears at a wavelength  $\lambda_x^+ \approx 200$ , corresponding to the subdomain size employed in the high- $Re_\tau$  case. This jump is also appreciated in the streamwise component at  $y^+ = 15$ . The POD-based approaches are outperformed by the FCN in the range of  $y^+ = 15\text{--}50$ . Farther from the wall, the accuracy of the FCN is matched by the FCN-POD method. It is interesting to note that the EPOD does not follow the same attenuation process as the FCN-based methods in the wall-normal and spanwise components. As one moves farther from the wall, the FCN-based methods fail to reproduce a wider range of small scales, whereas the EPOD exhibits more difficulties predicting the large scales. When it comes to the spectral peaks far from the wall, while the FCN-based methods produce noisier predictions than in the low- $Re_\tau$  case, the EPOD is not able to reproduce that part of the spectra. Taking inspiration from the FCN-POD approach, it might be possible to improve the EPOD performance by considering a local estimation on the subdomains, blended with a global estimation on the entire flow field. However, the way these two estimations should be combined is not trivial and this approach is left for future studies.

#### 4. Transfer learning

A number of more advanced techniques have also started to be adopted from the specialized machine-learning literature and applied to fluid-dynamics research. One notable example is transfer learning (Pan & Yang 2009), a method that allows the transfer of knowledge from one neural-network model to another one, thus reducing the amount data and time required for training. Guastoni *et al.* (2020) showed that the training time at a given wall-normal location may be significantly reduced if the network parameters are initialized using the optimized parameters of a previously trained network at another wall-normal location. Similarly, Kim & Lee (2020) used the convolutional network trained at a low Reynolds number to predict the flow at a higher Reynolds number.

Transfer learning represents an appealing solution for the main drawback of neural networks, which is the need to train them with a sufficient amount of data. Training typically requires specialized hardware and in our specific application the computational cost of generating the training and test datasets is not negligible. Furthermore, this cost grows as  $Re_\tau$  increases, making the generation of training data through DNS unfeasible at the Reynolds numbers that are relevant for engineering applications. In this regard, it is important to make an efficient use of the data and the trained models at our disposal. In this work, the possibility of transferring knowledge between models trained at different friction Reynolds numbers is investigated. At a fixed wall-normal distance, the weights of the FCN model trained on the dataset at  $Re_\tau = 180$  are loaded before training the network with the higher- $Re_\tau$  dataset. This is possible because the network has the same number of trainable parameters in both cases, as noted above. The learning rate is the only parameter that needs to be modified: a lower value has to be set, in order to prevent the optimizer from diverging too quickly from the weight configuration used for initialization. While in Guastoni *et al.* (2020) we froze the first layers of the initialized network because the input

Wall-bounded turbulence from wall quantities

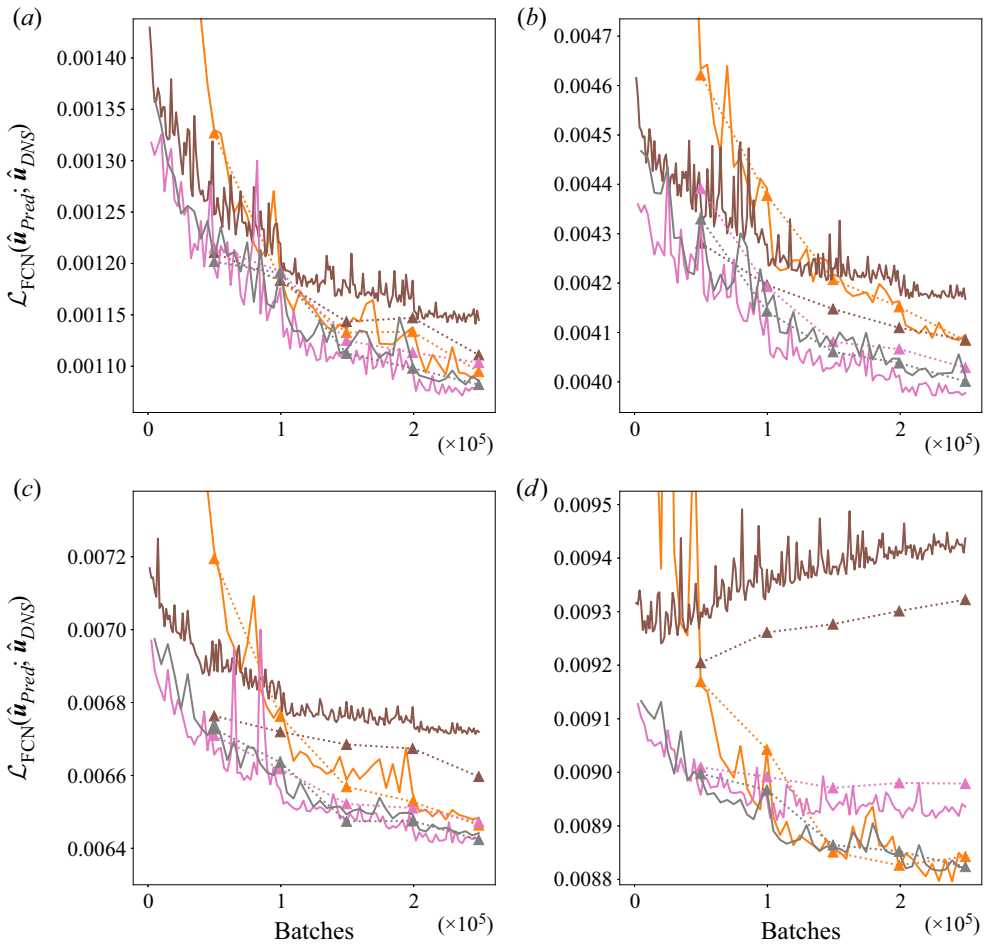


Figure 14. Validation (solid line) and test (dashed line) loss in the FCN prediction at (from left to right, top to bottom):  $y^+ = 15, 30, 50$  and  $100$ . Orange represents the models trained with the full dataset and random initialization, grey the models trained with the full dataset and initialized with previously trained networks, pink and brown represent models initialized with the parameters from the  $Re_\tau = 180$  network, trained with 50 % and 25 % of the original dataset, respectively.

was the same at the different wall-normal locations, in this case all the layers are trainable because the input distribution changes from one  $Re_\tau$  to the other one.

The transfer-learning numerical experiments that follow are designed with two objectives in mind. First, we investigate the effect of using the parameters of previously trained model to initialize a model that will be trained at a different target  $Re_\tau$  than the first model. Second, we verify whether the non-random initialization that we described can potentially improve the training process, reducing the amount of data at higher  $Re_\tau$  that is needed to achieve the same performance of a model trained from scratch, using the entire training/validation dataset, as we did in § 3. To this end, we initialized a FCN model with the parameters learnt on the  $Re_\tau = 180$  dataset and we trained such model on the full training/validation datasets at  $Re_\tau = 550$ . Subsequently, the same initialization is used for models that are trained on a reduced dataset at  $Re_\tau = 550$ , namely 25 % and 50 % of the full dataset. Differently from the previous sections, only one training run was

$E_{rms}^+(\cdot)$ (%)		$y^+ = 15$	$y^+ = 30$	$y^+ = 50$	$y^+ = 100$
$u$	Ref.	0.98 ( $\pm 0.66$ )	8.10 ( $\pm 0.62$ )	15.33 ( $\pm 0.22$ )	33.06 ( $\pm 0.30$ )
	100 %	1.22	7.15	16.09	32.75
	50 %	2.94	7.11	16.33	34.11
	25 %	1.15	7.74	14.78	33.78
$v$	Ref.	1.74 ( $\pm 0.11$ )	11.21 ( $\pm 1.41$ )	24.20 ( $\pm 1.38$ )	50.82 ( $\pm 0.26$ )
	100 %	1.86	9.40	24.40	51.59
	50 %	2.40	9.46	25.96	50.90
	25 %	1.71	11.33	23.15	50.43
$w$	Ref.	1.86 ( $\pm 0.60$ )	9.03 ( $\pm 0.31$ )	21.21 ( $\pm 1.27$ )	51.83 ( $\pm 0.38$ )
	100 %	1.75	8.65	21.05	53.04
	50 %	2.61	7.70	21.34	52.60
	25 %	1.22	9.67	20.35	49.75

Table 6. Percentage error in the prediction of the various r.m.s. fluctuations at the different wall-normal locations from models initialized with parameters from the  $Re_\tau = 180$  FCN. The statistics of the different initialized models are computed after 250 000 updates, and they are shown together the models with full dataset and random initialization, which is included as a reference. Results at  $Re_\tau = 550$ .

performed for each case. In order to compare models trained with datasets of different sizes, we considered the number of weight updates through the optimization algorithm during training. In figure 14 the validation and test losses are compared for the models trained with the full dataset and a random initialization. When the initialized model is trained on the full dataset, the performance is consistently better than that of the random initialization, both in terms of validation and test loss. The improvement is more evident close to the wall, whereas at  $y^+ = 100$  the two models provide approximately the same results after the first 150 000 updates. Transferring knowledge between different Reynolds numbers is then not only feasible, but also advantageous in terms of performance when the same amount of data is considered. Note that we kept the same ratio (4 : 1) when dividing training and validation sets even when the size of the dataset was reduced. In particular, when 50 % or 25 % of the samples are used, the validation set becomes too small to provide a reliable estimate of the error. On the other hand, the size of the test dataset is the same as the previous experiments. Up to  $y^+ = 50$ , the initialized networks are able to provide a performance that is very similar to that of the reference model with the same number of updates, with significant savings in terms of amount of data needed to train the network. On the other hand, at  $y^+ = 100$  a sufficient number of samples becomes a necessary condition to ensure the convergence to an optimal configuration: the loss of the network trained with 50 % of the training dataset does not improve after the first 100 000 updates, while with 25 % of the original dataset the network exhibits overfitting. The initialized models are able to provide a comparable accuracy also from the statistical point of view, as reported in table 6. We stress once again that the networks are not explicitly optimized to reduce the error in the statistics and that small variations in these error figures can be ascribed to the stochastic nature of the optimization algorithm. Overall, these results demonstrate the feasibility of knowledge transfer from models at different  $Re_\tau$ : with careful tuning of the hyperparameters it should be possible to substantially reduce the training time, as well as the amount of data needed for training. Although not tested, the transfer between different wall-normal locations described in Guastoni *et al.* (2020) is still applicable in this case, thus enabling a more efficient prediction of the flow at different wall-normal locations.



## 5. Conclusions

In this work, we introduced and compared two different models based on fully convolutional neural networks, for prediction of the velocity fluctuations at a given wall-normal distance, using quantities measured at the wall as inputs. The FCN and FCN-POD models are improved versions of previous architectures, used by Guastoni *et al.* (2020) and Güemes *et al.* (2019), respectively. Both of them are able to provide predictions in very good agreement with the reference data, simulated by means of the pseudo-spectral DNS code SIMSON (Chevalier *et al.* 2007), up to  $y^+ = 50$ . Such an agreement is verified by comparing the error in instantaneous predictions, turbulence statistics (namely r.m.s. fluctuations) and the energy content at the different wavelengths (i.e. spectral analysis). Both models show better prediction capabilities than EPOD (which is a linear method) in almost all the wall-normal locations and investigated features, thanks to their ability to predict nonlinear scale interactions. Furthermore, we showed that these architectures can be used at two different friction Reynolds numbers ( $Re_\tau = 180$  and 550) with minimal modifications, providing satisfactory results on both datasets.

The two models are designed under the assumption that local information at the wall is sufficient to predict the flow farther away, however, the FCN-POD model partially encodes further physical information of the system by using the spatial modes obtained through POD of the training dataset. On the other hand, features like the periodicity of the flow are enforced in the FCN by exploiting the mathematical characteristics of the model. These architectural differences are associated with performance discrepancies at the tested wall-normal locations; the FCN provides higher accuracy than the FCN-POD model closer to the wall, i.e. up to  $y^+ = 30$  at  $Re_\tau = 180$  and up to  $y^+ = 50$  at  $Re_\tau = 550$ . Farther from the wall, the FCN-POD method produces the most accurate predictions. The choice between these two models is motivated by the application into which the prediction model is integrated.

Despite the encouraging results discussed here, both models can be improved in terms of network architecture and training. An attempt to embed further physical information into the FCN did not result in improved predictions, as reported in § 3.1.1. The correct way of incorporating this information to enhance the predictions is an active area of research. As another example, the high-frequency noise in the FCN predictions could be reduced with appropriate filtering, possibly adding a trainable layer to the network to perform this operation. The FCN-POD model has a higher number of hyperparameters to be set, such as the number of predicted temporal modes or the size of the subdomains. A more thorough inspection of the hyperparameter space may provide a significant improvement in the prediction performance. Differently from the FCN, in the FCN-POD model the velocity components are not scaled to have the same magnitude: such a modification could help to predict the wall-normal and spanwise components of the velocity more accurately, even though it would also modify the POD mode sorting because of the different energy norm. Furthermore, the FCN-POD results exhibit lack of smoothness at the subdomain edges in the flow predictions. Finally, both models are trained to minimize a loss function based on the instantaneous error. Such a function could be modified to improve other physical characteristics of the predicted flow, for example the turbulence statistics and the spectral energy content.

To reduce the training time in view of industrial applications, the implementation of transfer learning was tested for the FCN model. Transfer learning can exploit a network trained at a lower Reynolds number to provide the weight initialization for training at a higher Reynolds number, thus reducing the requirements in terms of training time and data. The results are very encouraging, showing that it is possible to train the network with

50 % and even 25 % of the original training dataset, obtaining a performance similar to that of the reference model up to  $y^+ = 50$ .

Once the neural networks are trained, they are computationally cheap to evaluate, and they can become even cheaper by pruning the parts that have a negligible contribution to the final result off the network. Such an operation is not possible *a priori*, since the training determines how the inputs have to be processed to obtain the output. By reducing the computational cost of the evaluation it is possible to deploy the model using low-powered hardware and/or potentially run it in real time. Thus, the proposed FCN-based methods could be used for non-intrusive sensing of the flow, which is needed for closed-loop control applications. Furthermore, since the FCN models are able to reproduce nonlinear interactions in wall-bounded turbulence, new promising avenues in turbulence research could be opened by the network interpretation (Fan *et al.* 2021), as shown by Iten *et al.* (2020), who demonstrated that neural networks can provide relevant physical insights.

**Funding.** The authors acknowledge the funding provided by the Swedish e-Science Research Centre (SeRC), the G. Gustafsson Foundation and the Knut and Alice Wallenberg (KAW) Foundation. The numerical simulations were carried out on resources provided by the Swedish National Infrastructure for Computing (SNIC) at PDC and HPC2N. Part of this study was conducted in the context of the 4th Madrid Turbulence Workshop, in the context of the COTURB project (Coherent Structures in Wall-bounded Turbulence), funded by the European Research Council (ERC), under grant ERC-2014.AdG-669505. S.D. and A.I were partially supported by the project ARTURO, ref. PID2019-109717RB-I00/AEI/10.13039/501100011033, funded by the Spanish State Research Agency.

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

**Data availability statement.** The codes used to obtain the findings of this study are openly available on GitHub (<https://github.com/KTH-FlowAI/FCN-turbulence-predictions-from-wall-quantities>).

**Author ORCIDs.**

- ① Luca Guastoni <https://orcid.org/0000-0002-8589-1572>;
- ① Alejandro Güemes <https://orcid.org/0000-0002-1673-9956>;
- ① Andrea Ianiro <https://orcid.org/0000-0001-7342-4814>;
- ① Stefano Discetti <https://orcid.org/0000-0001-9025-1505>;
- ① Philipp Schlatter <https://orcid.org/0000-0001-9627-5903>;
- ① Hossein Azizpour <https://orcid.org/0000-0001-5211-6388>;
- ① Ricardo Vinuesa <https://orcid.org/0000-0001-6570-5499>.

## Appendix A

This appendix contains the training details of the neural-network models in the FCN and FCN-POD method. Both models introduced in § 2 need to be tuned to perform optimally on the chosen dataset. The networks were trained using the Adam (Kingma & Ba 2015) optimization algorithm for 50 epochs, with a scheduled exponential learning-rate decay. We used the optimizer hyperparameters suggested in the original paper, except the  $\epsilon$  parameter that was set to 0.1 following TensorFlow recommendations (Abadi *et al.* 2016). The total number of trainable parameters in the FCN is 1 264 131 and it does not depend on the  $Re_\tau$  of the dataset. On the other hand, this number is  $Re_\tau$ -dependent for the FCN-POD model, as the number of subdomains and reconstructed POD modes changes. For the  $Re_\tau = 180$  case the number of trainable parameters is 4 733 248, while for the  $Re_\tau = 550$  case it is 5 028 224.

For the FCN model the  $Re_\tau = 180$  dataset, composed by input and output fields, has a size of about 70 GB per wall-normal distance, while the  $Re_\tau = 550$  dataset occupies 120 GB. Depending on the amount of available RAM it might not be possible to load

## Wall-bounded turbulence from wall quantities

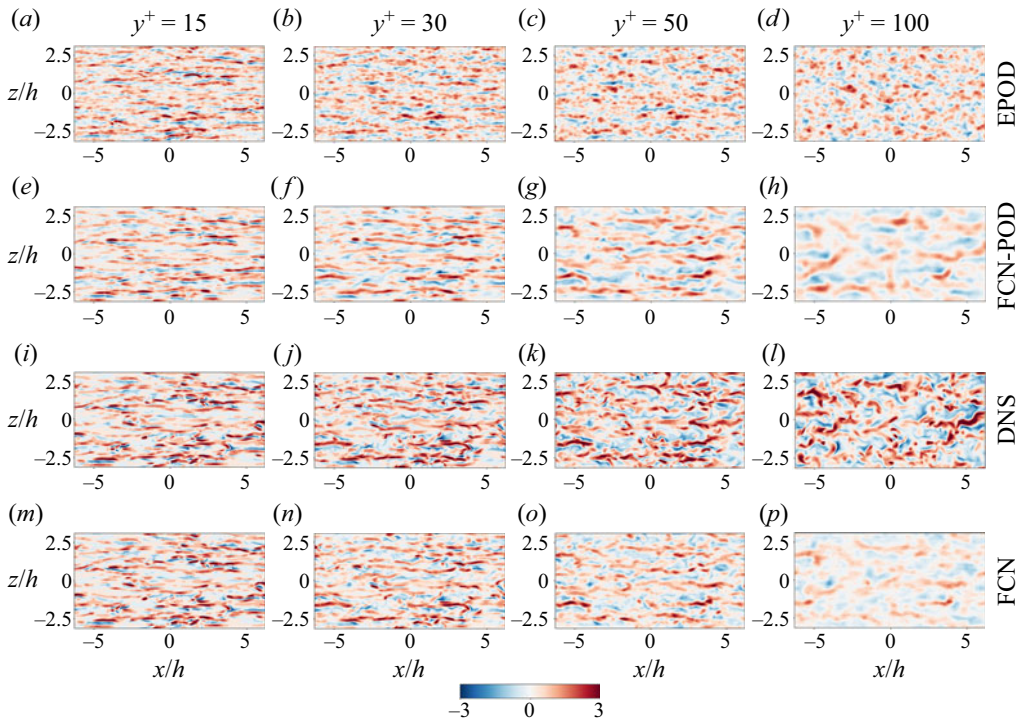


Figure 15. Comparison of the wall-normal fluctuation fields at  $Re_\tau = 180$ , scaled with the corresponding  $v_{rms}$ , from EPOD (a–d), FCN-POD (e–h), reference DNS (i–l) and FCN (m–p). Results at  $y^+ = 15$  (a,e,i,m),  $y^+ = 30$  (b,f,j,n),  $y^+ = 50$  (c,g,k,o) and  $y^+ = 100$  (d,h,l,p).

the entire dataset in memory, however, our implementation does not require to load it at once, allowing training on less-performing computer as well. The training was performed on two different GPUs, namely NVIDIA K80 and NVIDIA RTX2080Ti. The former has 24 GB of GPU memory, whereas the latter only 12 GB. The turbulent field snapshots have a comparatively high resolution with respect to the samples used in image processing and the GPU memory can also be a limiting parameter. For a given dataset, the amount of memory that is allocated on the GPU depend on the batch size, that is the number of samples used to estimate the loss function gradient at each update of the learnable parameters of the network. In the  $Re_\tau = 550$ , the batch size is limited to 2 samples when the faster RTX2080Ti is used. With this setting the gradient approximation might be too noisy and the training performance might be degraded, therefore we suggest using at least a batch size of 4, using two RTX2080Ti GPUs.

Due to the reduced-order nature of FCN-POD method, the size of the training/validation datasets is lower: 40 GB for  $Re_\tau = 180$  dataset and 70 GB for  $Re_\tau = 550$  dataset. Thanks to the max pooling operations that are not present in the FCN model, the size of the tensors allocated on GPU is smaller and this allows training of the model with a batch size of 8 for both datasets.

## Appendix B

This appendix contains the wall-normal and spanwise fluctuations corresponding to the fields shown in figures 6 and 8 for  $Re_\tau = 180$  and 550, respectively. If we observe

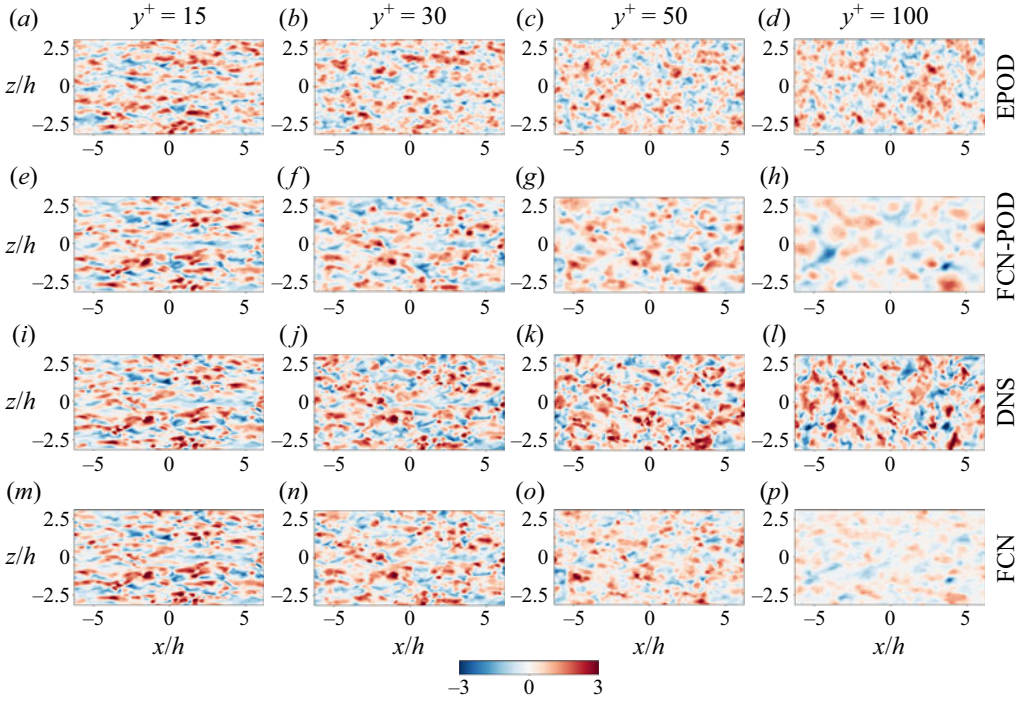


Figure 16. Comparison of the spanwise fluctuation fields at  $Re_\tau = 180$ , scaled with the corresponding  $w_{rms}$ , from EPOD (a–d), FCN-POD (e–h), reference DNS (i–l) and FCN (m–p). Results at  $y^+ = 15$  (a,e,i,m),  $y^+ = 30$  (b,f,j,n),  $y^+ = 50$  (c,g,k,o) and  $y^+ = 100$  (d,h,l,p).

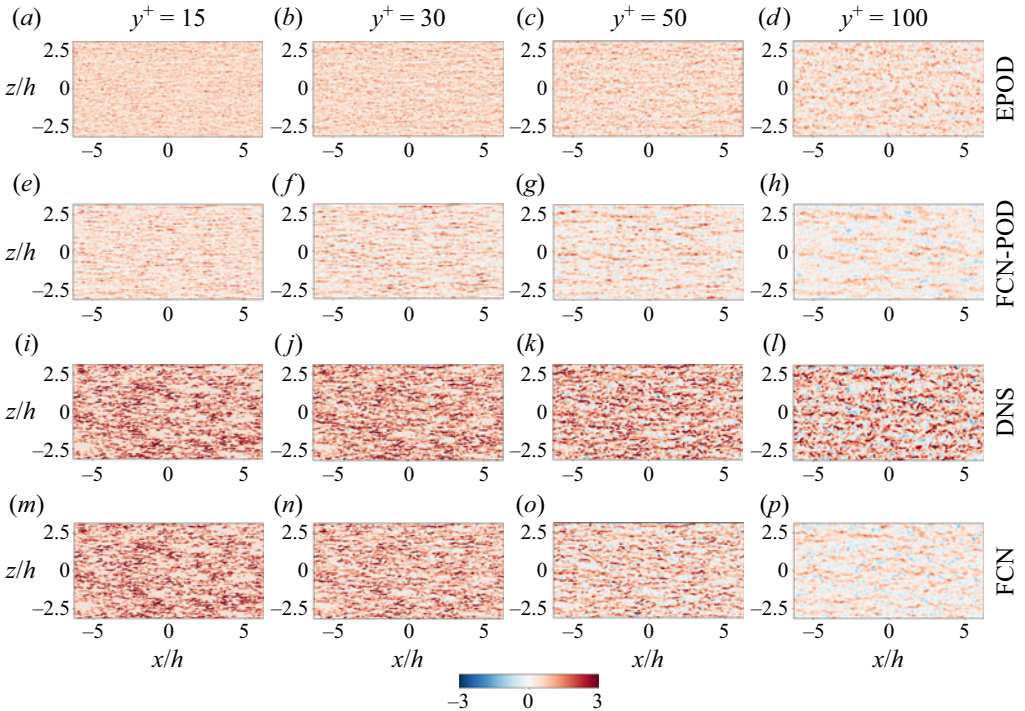


Figure 17. Comparison of the wall-normal fluctuation fields at  $Re_\tau = 550$ , scaled with the corresponding  $v_{rms}$ , from EPOD (a–d), FCN-POD (e–h), reference DNS (i–l) and FCN (m–p). Results at  $y^+ = 15$  (a,e,i,m),  $y^+ = 30$  (b,f,j,n),  $y^+ = 50$  (c,g,k,o) and  $y^+ = 100$  (d,h,l,p).



Wall-bounded turbulence from wall quantities

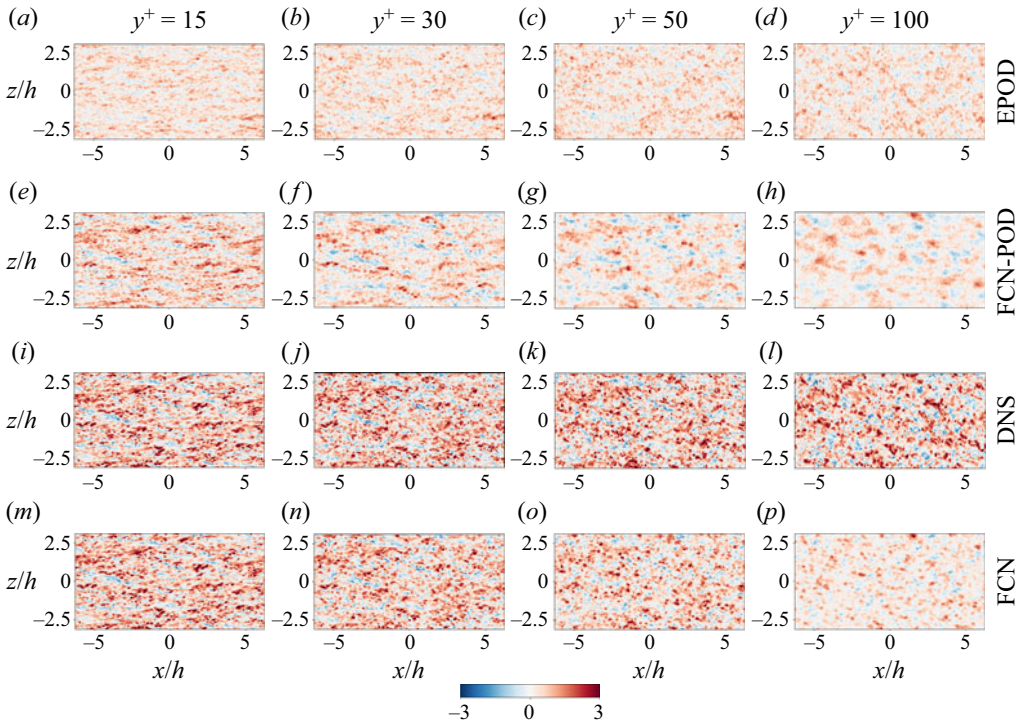


Figure 18. Comparison of the spanwise fluctuation fields at  $Re_\tau = 550$ , scaled with the corresponding  $w_{rms}$ , from EPOD (a–d), FCN-POD (e–h), reference DNS (i–l) and FCN (m–p). Results at  $y^+ = 15$  (a, e, i, m),  $y^+ = 30$  (b, f, j, n),  $y^+ = 50$  (c, g, k, o) and  $y^+ = 100$  (d, h, l, p).

the wall-normal fluctuations at  $Re_\tau = 180$  in figure 15, the FCN model provides better predictions than POD-based methods at  $y^+ = 15$ . The tiling of the subdomains is more apparent in the FCN-POD predictions, while the EPOD predictions contain small features that are not present in the DNS reference field. Similar observations can be made at  $y^+ = 30$ . As we move farther away from the wall, at  $y^+ = 50$ , the accuracy of the neural-network-based models is similar. For the FCN-POD the flow features in the streamwise direction appear smoothed out with respect to the DNS fields; on the other hand, the same features have more jagged edges in the FCN predictions, possibly as a consequence of the localized nature of the predictions. As observed for the streamwise velocity fluctuations, the performance degrades as we move farther away from the wall. Note that some features are completely missing in the EPOD predictions, due to the fact that the three velocity components are predicted at the same time. From the observation of figure 16, which shows the spanwise fluctuation predictions, the behaviour of the three methods is similar to the one in the wall-normal components. Note, however, that, closer to the wall, the FCN-POD provides more accurate predictions for the spanwise fluctuation component than for the wall-normal component, as can be observed in figure 7.

When a higher Reynolds number is considered, the error trends of the neural-network-based models in figure 9 are comparable to the one observed at  $Re_\tau = 180$ , for the wall-normal and spanwise fluctuations. On the other hand, the EPOD predictions improve when moving from  $y^+ = 15$  to  $y^+ = 30$ , before the performance starts degrading as we move farther from the wall. Additionally, figures 17 and 18 show that, close to the wall, the intensity of the fluctuations in the wall-normal and spanwise direction is



significantly underestimated by the POD-based methods, when compared with FCN. As the wall-normal distance increases, the fluctuations range predicted by the FCN reduces as well and at  $y^+ = 100$  the range is comparable for all three methods.

#### REFERENCES

- ABADI, M., *et al.* 2016 Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283. USENIX.
- ADRIAN, R. 1988 Stochastic estimation of organized turbulent structure: homogeneous shear flow. *J. Fluid Mech.* **190**, 531–559.
- BAARS, W.J. & TINNEY, C.E. 2014 Proper orthogonal decomposition-based spectral higher-order stochastic estimation. *Phys. Fluids* **26** (5), 055112.
- BALDI, P. & HORNIK, K. 1989 Neural networks and principal component analysis: learning from examples without local minima. *Neural Networks* **2** (1), 53–58.
- BECK, A.D., FLAD, D.G. & MUNZ, C.-D. 2019 Deep neural networks for data-driven LES closure models. *J. Comput. Phys.* **398**, 108910.
- BORÉE, J. 2003 Extended proper orthogonal decomposition: a tool to analyse correlated events in turbulent flows. *Exp. Fluids* **35** (2), 188–192.
- BOURGOIS, J.A., NOACK, B.R. & MARTINUZZI, R.J. 2013 Generalized phase average with applications to sensor-based flow estimation of the wall-mounted square cylinder wake. *J. Fluid Mech.* **736**, 316–350.
- BOURLARD, H. & KAMP, Y. 1988 Auto-association by multilayer perceptrons and singular value decomposition. *Biol. Cybern.* **59** (4–5), 291–294.
- BRUNTON, S.L., NOACK, B.R. & KOUMOUTSAKOS, P. 2020 Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **52**, 477–508.
- BUCCI, M.A., SEMERARO, O., ALLAUZEN, A., WISNIEWSKI, G., CORDIER, L. & MATHELIN, L. 2019 Control of chaotic systems by deep reinforcement learning. *Proc. R. Soc. Lond. A* **475**, 20190351.
- CHEVALIER, M., SCHLATTER, P., LUNDBLADH, A. & HENNINGSON, D.S. 2007 A pseudospectral solver for incompressible boundary layer flows. *Tech. Rep. TRITA-MEK 2007:07*. KTH Mechanics, Stockholm, Sweden.
- CHOI, H., MOIN, P. & KIM, J. 1994 Active turbulence control for drag reduction in wall-bounded flows. *J. Fluid Mech.* **262**, 75–110.
- DE FAUW, J. 2018 Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nat. Med.* **24**, 1342–1350.
- DISCETTI, S., BELLANI, G., ÖRLÜ, R., SERPIERI, J., SANMIGUEL VILA, C., RAIOLA, M., ZHENG, X., MASCOTELLI, L., TALAMELLI, A. & IANIRO, A. 2019 Characterization of very-large-scale motions in high-Re pipe flows. *Exp. Therm. Fluid Sci.* **104**, 1–8.
- DISCETTI, S., RAIOLA, M. & IANIRO, A. 2018 Estimation of time-resolved turbulent fields through correlation of non-time-resolved field measurements and time-resolved point measurements. *Exp. Therm. Fluid Sci.* **93**, 119–130.
- DOGAN, E., ÖRLÜ, R., GATTI, D., VINUESA, R. & SCHLATTER, P. 2019 Quantification of amplitude modulation in wall-bounded turbulence. *Fluid Dyn. Res.* **51**, 011408.
- DUMOULIN, V. & VISIN, F. 2016 A guide to convolution arithmetic for deep learning. [arXiv:1603.07285](https://arxiv.org/abs/1603.07285).
- DURAISAMY, K., IACCARINO, G. & XIAO, H. 2019 Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.* **51**, 357–377.
- EIVAZI, H., GUASTONI, L., SCHLATTER, P., AZIZPOUR, H. & VINUESA, R. 2021 Recurrent neural networks and Koopman-based frameworks for temporal predictions in a low-order model of turbulence. *Intl J. Heat Fluid Flow* **90**, 108816.
- ENCINAR, M.P. & JIMÉNEZ, J. 2019 Logarithmic-layer turbulence: a view from the wall. *Phys. Rev. Fluids* **4**, 114603.
- ERICHSOHN, N.B., MATHELIN, L., YAO, Z., BRUNTON, S.L., MAHONEY, M.W. & KUTZ, J.N. 2020 Shallow neural networks for fluid flow reconstruction with limited sensors. *Proc. R. Soc. Lond. A* **476**, 20200097.
- FAN, F., XIONG, J., LI, M. & WANG, G. 2021 On interpretability of artificial neural networks: a survey. *IEEE Trans. Radiat. Plasma Med. Sci.* [doi:10.1109/TRPMS.2021.3066428](https://doi.org/10.1109/TRPMS.2021.3066428).
- FUKAMI, K., FUKAGATA, K. & TAIRA, K. 2019a Super-resolution reconstruction of turbulent flows with machine learning. *J. Fluid Mech.* **870**, 106–120.
- FUKAMI, K., FUKAGATA, K. & TAIRA, K. 2021 Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows. *J. Fluid Mech.* **909**, A9.

- FUKAMI, K., NABAE, Y., KAWAI, K. & FUKAGATA, K. 2019b Synthetic turbulent inflow generator using machine learning. *Phys. Rev. Fluids* **4**, 064603.
- FUKUSHIMA, K. 1980 Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* **36** (4), 193–202.
- FUKUSHIMA, K. 1988 Neocognitron: a hierarchical neural network capable of visual pattern recognition. *Neural Networks* **1** (2), 119–130.
- GOODFELLOW, I., BENGIO, Y. & COURVILLE, A. 2016 *Deep Learning*. MIT Press.
- GUASTONI, L., ENCINAR, M.P., SCHLATTER, P., AZIZPOUR, H. & VINUESA, R. 2020 Prediction of wall-bounded turbulence from wall quantities using convolutional neural networks. *J. Phys.: Conf. Ser.* **1522**, 012022.
- GUASTONI, L., SRINIVASAN, P.A., AZIZPOUR, H., SCHLATTER, P. & VINUESA, R. 2019 On the use of recurrent neural networks for predictions of turbulent flows. In *Proceedings of the 11th International Symposium on Turbulence and Shear Flow Phenomena (TSFP11)*, Southampton, UK, July 30 – August 2.
- GÜEMES, A., DISCETTI, S. & IANIRO, A. 2019 Sensing the turbulent large-scale motions with their wall signature. *Phys. Fluids* **31** (12), 125112.
- GÜEMES, A., DISCETTI, S., IANIRO, A., SIRMACEK, B., AZIZPOUR, H. & VINUESA, R. 2021 From coarse wall measurements to turbulent velocity fields through deep learning. *Phys. Fluids* **33** (7), 075121.
- GUZMÁN IÑIGO, J., SIPP, D. & SCHMID, P.J. 2014 A dynamic observer to capture and control perturbation energy in noise amplifiers. *J. Fluid Mech.* **758**, 728–753.
- HAM, Y.-G., KIM, J.-H. & LUO, J.J. 2019 Deep learning for multi-year ENSO forecasts. *Nature* **573**, 568–572.
- HINTON, G.E. & SALAKHUTDINOV, R.R. 2006 Reducing the dimensionality of data with neural networks. *Science* **313** (5786), 504–507.
- HORNIK, K., STINCHCOMBE, M. & WHITE, H. 1989 Multilayer feedforward networks are universal approximators. *Neural Networks* **2** (5), 359–366.
- HOSSEINI, Z., MARTINUZZI, R.J. & NOACK, B.R. 2016 Modal energy flow analysis of a highly modulated wake behind a wall-mounted pyramid. *J. Fluid Mech.* **798**, 717–750.
- IOFFE, S. & SZEGEDY, C. 2015 Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 448–456.
- ITEN, R., METGER, T., WILMING, H., DEL RIO, L. & RENNER, R. 2020 Discovering physical concepts with neural networks. *Phys. Rev. Lett.* **124**, 010508.
- JAGODINSKI, E., ZHU, X. & VERMA, S. 2020 Uncovering dynamically critical regions in near-wall turbulence using 3D convolutional neural networks. [arXiv:2004.06187](https://arxiv.org/abs/2004.06187).
- JEAN, N., BURKE, M., XIE, M., DAVIS, W.M., LOBELL, D.B. & ERMON, S. 2016 Combining satellite imagery and machine learning to predict poverty. *Science* **353**, 790–794.
- JIMÉNEZ, J. 2018 Machine-aided turbulence theory. *J. Fluid Mech.* **854**, R1.
- KIM, J. & LEE, C. 2020 Prediction of turbulent heat transfer using convolutional neural networks. *J. Fluid Mech.* **882**, A18.
- KINGMA, D.P. & BA, J. 2015 Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015*.
- KUTZ, J.N. 2017 Deep learning in fluid dynamics. *J. Fluid Mech.* **814**, 1–4.
- LAPEYRE, C.J., MISDARIIS, A., CAZARD, N., VEYNANTE, D. & POINSOT, T. 2019 Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates. *Combust. Flame* **203**, 255.
- LECUN, Y., BENGIO, Y. & HINTON, G. 2015 Deep learning. *Nature* **521** (7553), 436–444.
- LECUN, Y., BOSER, B., DENKER, J.S., HENDERSON, D., HOWARD, R.E., HUBBARD, W. & JACKEL, L.D. 1989 Backpropagation applied to handwritten ZIP code recognition. *Neural Comput.* **1** (4), 541–551.
- LECUN, Y., BOTTOU, L., BENGIO, Y. & HAFFNER, P. 1998 Gradient-based learning applied to document recognition. In *Proc. of the IEEE*, pp. 2278–2324.
- LEE, C., KIM, J., BABCOCK, D. & GOODMAN, R. 1997 Application of neural networks to turbulence control for drag reduction. *Phys. Fluids* **9**, 1740–1747.
- LING, J., KURZAWSKI, A. & TEMPLETON, J. 2016 Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* **807**, 155–166.
- LIU, Z., ADRIAN, R.J. & HANRATTY, T.J. 2001 Large-scale modes of turbulent channel flow: transport and structure. *J. Fluid Mech.* **448**, 53–80.
- LONG, J., SHELHAMER, E. & DARRELL, T. 2015 Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440.
- LOZANO-DURÁN, A., BAE, H.J. & ENCINAR, M.P. 2020 Causality of energy-containing eddies in wall turbulence. *J. Fluid Mech.* **882**, A2.

- LOZANO-DURÁN, A., FLORES, O. & JIMÉNEZ, J. 2012 The three-dimensional structure of momentum transfer in turbulent channels. *J. Fluid Mech.* **694**, 100–130.
- LUMLEY, J.L. 1967 The structure of inhomogeneous turbulent flows. In *Atmospheric Turbulence and Radio Wave Propagation* (ed. A.M. Yaglom & V.I. Tartarsky), pp. 166–178. Nauka.
- MA, W. & LU, J. 2017 An equivalence of fully connected layer and convolutional layer. [arXiv:1712.01252](https://arxiv.org/abs/1712.01252).
- MAKARASHVILI, V., MERZARI, E., OBABKO, A., SIEGEL, A. & FISCHER, P. 2017 A performance analysis of ensemble averaging for high fidelity turbulence simulations at the strong scaling limit. *Comput. Phys. Commun.* **219**, 236–245.
- MARUSIC, I. & HEUER, W.D.C. 2007 Reynolds number invariance of the structure inclination angle in wall turbulence. *Phys. Rev. Lett.* **99**, 114504.
- MAULIK, R., SAN, O., RASHEED, A. & VEDULA, P. 2019 Subgrid modelling for two-dimensional turbulence using neural networks. *J. Fluid Mech.* **858**, 122–144.
- MILANO, M. & KOUMOUTSAKOS, P. 2002 Neural network modeling for near wall turbulent flow. *J. Comput. Phys.* **182** (1), 1–26.
- MOEHLIS, J., FAISST, H. & ECKHARDT, B. 2004 A low-dimensional model for turbulent shear flows. *New J. Phys.* **6**, 56.
- MOKHASI, P., REMPFER, D. & KANDALA, S. 2009 Predictive flow-field estimation. *Physica D* **238**, 290–308.
- MURATA, T., FUKAMI, K. & FUKAGATA, K. 2020 Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *J. Fluid Mech.* **882**, A13.
- NAIR, N.J. & GOZA, A. 2020 Leveraging reduced-order models for state estimation using deep learning. *J. Fluid Mech.* **897**, R1.
- NAIR, V. & HINTON, G.E. 2010 Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 807–814.
- NOROUZZADEH, M.S., NGUYEN, A., KOSMALA, M., SWANSON, A., PALMER, M.S., PACKER, C. & CLUNE, J. 2018 Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proc. Natl Acad. Sci. USA* **115**, E5716–E5725.
- VAN DEN OORD, A., DIELEMAN, S., ZEN, H., SIMONYAN, K., VINYALS, O., GRAVES, A., KALCHBRENNER, N., SENIOR, A. & KAVUKCUOGLU, K. 2016 Wavenet: A generative model for raw audio. [arXiv:1609.03499](https://arxiv.org/abs/1609.03499).
- PAN, S.J. & YANG, Q. 2009 A survey on transfer learning. *IEEE Trans. Knowl. Data Engng* **22** (10), 1345–1359.
- PANDEY, S., SCHUMACHER, J. & SREENIVASAN, K.R. 2020 A perspective on machine learning in turbulent flows. *J. Turbul.* **21**, 567–584.
- RABAULT, J., KUCHTA, M., JENSEN, A., RÉGLADE, U. & CERARDI, N. 2019 Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *J. Fluid Mech.* **865**, 281–302.
- RAIBAUDO, C., ZHONG, P., NOACK, B.R. & MARTINUZZI, R.J. 2020 Machine learning strategies applied to the control of a fluidic pinball. *Phys. Fluids* **32** (1), 015108.
- RAISSI, M., PERDIKARIS, P. & KARNIADAKIS, G.E. 2019 Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707.
- RAISSI, M., YAZDANI, A. & KARNIADAKIS, G.E. 2020 Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. *Science* **367**, 1026–1030.
- SASAKI, K., VINUESA, R., CAVALIERI, A.V.G., SCHLATTER, P. & HENNINGSON, D.S. 2019 Transfer functions for flow predictions in wall-bounded turbulence. *J. Fluid Mech.* **864**, 708–745.
- SIROVICH, L. 1987 Turbulence and the dynamics of coherent structures. II. Symmetries and transformations. *Q. Appl. Maths* **45** (3), 573–582.
- SRINIVASAN, P.A., GUASTONI, L., AZIZPOUR, H., SCHLATTER, P. & VINUESA, R. 2019 Predictions of turbulent shear flows using deep neural networks. *Phys. Rev. Fluids* **4** (5), 054603.
- SUZUKI, T. & HASEGAWA, Y. 2017 Estimation of turbulent channel flow at  $Re_\theta = 100$  based on the wall measurement using a simple sequential approach. *J. Fluid Mech.* **830**, 760–796.
- TANG, H., RABAULT, J., KUHNLE, A., WANG, Y. & WANG, T. 2020 Robust active flow control over a range of Reynolds numbers using an artificial neural network trained through deep reinforcement learning. *Phys. Fluids* **32** (5), 053605.
- TINNEY, C.E., COIFFET, F., DELVILLE, J., HALL, A.M., JORDAN, P. & GLAUSER, M.N. 2006 On spectral linear stochastic estimation. *Exp. Fluids* **41** (5), 763–775.
- TINNEY, C.E., UKEILEY, L.S. & GLAUSER, M.N. 2008 Low-dimensional characteristics of a transonic jet. Part 2. Estimate and far-field prediction. *J. Fluid Mech.* **615**, 53–92.

## Wall-bounded turbulence from wall quantities

- UDRESCU, S.-M. & TEGMARK, M. 2020 AI Feynman: a physics-inspired method for symbolic regression. *Sci. Adv.* **6**, 1–16.
- VINUESA, R., AZIZPOUR, H., LEITE, I., BALAAM, M., DIGNUM, V., DOMISCH, S., FELLÄNDER, A., LANGHANS, S.D., TEGMARK, M. & FUSO NERINI, F. 2020 The role of artificial intelligence in achieving the sustainable development goals. *Nat. Commun.* **11**, 233.
- WANG, J.X., WU, J.L. & XIAO, H. 2017 Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data. *Phys. Rev. Fluids* **2** (3), 034603.
- WU, J.-L., XIAO, H. & PATERSON, E. 2018 Physics-informed machine learning approach for augmenting turbulence models: a comprehensive framework. *Phys. Rev. Fluids* **3**, 074602.
- WU, Y. 2014 A study of energetic large-scale structures in turbulent boundary layer. *Phys. Fluids* **26** (4), 045113.