

Predicting the skin friction's evolution in a forced turbulent channel flow

A. Martín-Gil ^{a,b}, O. Flores ^{a,*}

^a Department of Aerospace Engineering, Universidad Carlos III de Madrid, Leganés, Spain

^b TNO, Heat Transfer and Fluid Dynamics, The Netherlands

ARTICLE INFO

Keywords:

Wall-bounded turbulence
Predictability
Flow control
Drag reduction
Machine learning

ABSTRACT

The present paper reports on the ability of neural networks (NN) and linear stochastic estimation (LSE) tools to predict the evolution of skin friction in a minimal turbulent channel ($Re_\tau = 165$) after applying an actuation near the wall that is localized in space and time. Two different NN architectures are compared, namely multilayer perceptrons (MLP) and convolutional neural networks (CNN). The paper describes the effect that the predictive horizon and the type/size/number of wall-based sensors have on the performance of each estimator. The performance of MLPs and LSEs is very similar, and becomes independent of the sensor's size when they are smaller than 60 wall units. For sufficiently small sensors, the CNN outperforms MLPs and LSEs, suggesting that CNNs are able to incorporate some of the non-linearities of the near-wall cycle in their prediction of the skin friction evolution after the actuation. Indeed, the CNN is the only architecture able to achieve reasonable predictive capabilities using pressure sensors only. The predictive horizon has a strong effect on the predictive capacity of both NN and LSE, with a Pearson correlation coefficient that varies from 0.95 for short times (i.e., of the order of the actuation time) to less than 0.4 for times of the order of an eddy turn-over time. The analysis of the weights and filters in the LSE and NNs show that all estimators are targeting wall-signatures consistent with streaks, which is interpreted as the streak being the most causal feature in the near-wall cycle for the present forcing.

1. Introduction

In most engineering applications, turbulent motions play a crucial role in the transport of momentum, heat and mass. As a consequence, turbulence control is one of the most important technological challenges of today's industry. Particularly important within the topic of turbulence control is the control of turbulent skin friction drag. In the aerospace industry alone, skin friction drag accounts for roughly 50% of the fuel consumption of commercial airplanes, thus having an important impact on CO₂ emissions as well [1]. Turbulent skin friction is also important in other engineering areas, representing about 60% of the total drag of large ships [2], and roughly 100% of the drag in the transport of gas/oil through pipelines.

There are many academic works on the topic of turbulence control for skin friction reduction on wall-bounded flows, as summarized in the review articles by Brunton and Noack [3] and Zhang et al. [4]. Different control strategies have been proposed, which can be classified into passive control (like riblets, see [5,6], among others) and active control strategies. The latter are usually divided into open-loop and closed-loop control, depending on whether the actuation is defined a priori or the control uses the state of the turbulent flow as input. Examples of active open-loop (i.e., predefined) control include periodic spanwise forcing

like wall oscillations or travelling-waves (see review by Ricco et al. [7]), and uniform blowing/suction through the wall [8,9]. They are conceptually less complex than closed-loop control strategies, but are only efficient near the design point of operation. On the other hand, closed-loop control strategies offer better control performance with smaller power consumption than open-loop control. They are also more robust and versatile, at the expense of requiring a complex network of sensors to measure the state of the flow [10].

One of the most common actuation systems used in closed-loop control strategies for reducing skin-friction drag are fluidic actuators [11], that work injecting a wall-normal mass flux through the wall (i.e., blowing/suction). This actuators can result in skin friction reduction either by lifting the boundary layer and creating a low-shear region downstream of the actuator [8,9], or by disrupting the near-wall cycle of the quasi-streamwise vortices and streaks [12]. The largest local skin friction reductions seem to be associated with the former, having a more persistent effect downstream of the actuator [13]. For instance, in a recent paper [14] reports skin friction reductions of up to 70% with open-loop active control, where the injection of low-momentum fluid and the disruption of the near-wall cycle account for

* Corresponding author.

E-mail address: oflores@ing.uc3m.es (O. Flores).

77% and 23% of the reduction, respectively. While the injection of low-momentum fluid can be an effective control strategy in many situations, it requires a supply of fluid that might not be available in some cases. In the rest of the paper, we will focus on control strategies targeting the disruption of the near-wall cycle, which require zero-net-mass-flux actuation [11].

Ranked among the best performances in terms of skin friction reduction associated to the disruption of the near-wall cycle is the 20% drag reduction achieved by Lee et al. [15], using a closed-loop active control on a turbulent channel. They applied control at the wall by changing the wall-normal boundary condition (blowing/suction), and the flow was sensed using the instantaneous viscous shear stresses at the wall. The original controller was a non-linear neural network (NN), although [16] showed later that similar performance could be achieved using a linear controller. The control schemes produced by these data-driven works were consistent with the model-driven *opposition control* proposed by Choi et al. [17]: to inhibit vertical velocity fluctuations in the buffer layer by applying at the wall a vertical velocity that is opposite to the estimated vertical velocity at some (small) distance from the wall.

One of the main limitations to the use of these opposition control schemes in practical applications is the fact that sensing and actuation are based on continuous variables distributed over the whole wall [10]. In any practical application, sensors and actuators have finite sizes and are *localized* in space (i.e., the whole wall is not covered with sensors and actuators), limiting what the sensor *sees* and what the actuator can *do*. This results in lower performances for systems based on localized sensors and actuators. For instance, Abbassi et al. [18] reports an implementation of the opposition control of Choi et al. [17] with localized actuators and sensors, yielding a limited 3% drag reduction (compare to the 20% of Lee et al. [15]). Moreover, similarly to Cheng et al. [14], approximately 66% of the measured drag reduction was associated to the injection of low-momentum fluid, and only the remaining 33% was associated to the damping of the near-wall cycle vertical velocity structures. This rises the question of whether there might be more advantageous control strategies to disrupt the near-wall cycle using localized sensing/forcing than a local implementation of the opposition control of Choi et al. [17].

Pursuing this idea, Pastor et al. [19] developed a Monte Carlo experiment to evaluate the effect of a localized control (in space and time) in the near-wall cycle of a low-Reynolds number turbulent channel flow. The authors run simultaneously simulations of channel flows with and without a localized vertical force that actuated for a short time. Roughly two thousand episodes were recorded, ensuring that the actuation sampled a significant portion of the possible states of the flow. For the cases with a positive (negative) vertical force, the wall shear and pressure fields conditioned to drag decrease showed high (low) velocity streaks aligned with the actuator, consistent with the idea of opposition control. However, a preliminary attempt to define localized, finite-size wall-sensors to drive the forcing was unsuccessful. The main objective of the present work is to remedy that.

The problem of sensing the flow from the wall has received increasing attention over the last decade. Various tools have been used to that purpose, from different types of neural networks [20–23], to linear and non-linear transfer functions [24], linear statistical models (like the Linear Stochastic Estimation used in [25]), and models based on linearized dynamics of the Navier–Stokes equations [26,27], among others. A common finding on these studies is that, irrespective of the method used, flow features can only be reconstructed at a given wall-distance if they leave a footprint at the measurement plane. For wall measurements, that implies that only Townsend’s attached eddies can be reconstructed, and only large scale flow features are recovered in the logarithmic and outer regions. Another interesting observation is that the reconstructions of large Re_τ flows (i.e., when the logarithmic region exists) using non-linear methods (i.e., NN) are usually just marginally better than those obtained using linear methods (i.e., LSE). Encinar

and Jiménez [25] argue that this is probably a consequence of the relatively linear dynamics of the larger attached eddies [28], which are the ones leaving a more distinct footprint at the wall. Note that, when the Reynolds number is small (i.e., $Re_\tau \approx 180$), non-linear methods are more likely to outperform linear methods [23].

The present work focuses on the estimation (using an array of finite-size sensors distributed over the wall) of the change of skin friction drag after a localized actuation. The configuration selected for this study is a turbulent channel flow with low-Reynolds number, so that the turbulent scales in the channel consist only on the near-wall region [29,30]. Given that the proposed problem shares some similitude with the flow reconstruction problem described above, we will use similar tools. In particular, we will evaluate the performance of two architectures of neural networks (namely, multilayer perceptrons MLP, and convolutional neural networks CNN) and linear stochastic estimators (LSE) to predict the evolution of the flow after the actuation, with the aim of discerning the ability of the former to exploit the nonlinear near-wall dynamics to provide more accurate predictions. For all cases, the input for the estimator will be provided by a grid of finite-size pressure and skin friction sensors at the wall. Special attention will be paid to the effect that the density of sensors and the time horizon of the prediction have on the ability of each method to properly estimate the difference in the skin friction with respect to the base flow.

The paper is organized as follows. The numerical database used for the study is presented first (Section 2.1), followed by a description of the NNs (Section 2.2) and the LSE (Section 2.3). Section 3 presents the results of the study, including the variation of the estimator metrics with the predictive horizon and the number of sensors (Section 3.1), and the evaluation of the wall signatures targeted by the NNs and the LSE to perform their predictions (Section 3.2), followed by a comparison in between the two NNs considered: MLP and CNN (Section 3.3). Conclusions are presented in Section 4.

2. Methodology

This section describes the database of turbulent channel flows used in this work, as well as the main characteristics of the considered predictive tools: neural networks (NN) and linear stochastic estimation (LSE).

2.1. Numerical database

We use the database of turbulent channel flows developed by Pastor et al. [19]. The flow configuration consists on an incompressible turbulent flow driven by a pressure gradient between two infinite walls, separated a distance $2h$. The friction Reynolds number ($Re_\tau = u_\tau h/\nu = 165$, where u_τ is the friction velocity and ν is the kinematic viscosity) and the size of the periodic computational domain ($L_x = \pi h$ and $L_z = 0.5\pi h$ in the streamwise and spanwise directions, respectively) are slightly larger than those of the minimal flow units of Jiménez and Moin [29], providing a reliable representation of the near-wall dynamics and kinematics [30] at a minimal computational cost (CPU/GPU time and storage). Note that, even if the Reynolds number is small, the problem is still relevant for higher Reynolds number flows. First, the dynamics of the logarithmic region is relatively similar to the dynamics of the near-wall region [30], with the generation of elongated streaks and periodic bursting. Second, most active control strategies (i.e., like the opposition control of Choi et al. [17]) and passive control strategies (i.e., riblets) focus on modifying the near-wall eddies, since they are responsible for a large fraction of the total skin friction.

The simulations are performed with a GPU-enabled pseudo-spectral Direct Numerical Simulation solver [31]. Time integration is performed with a semi-implicit three-step Runge–Kutta. Spatial discretization uses Fourier expansions in the wall-parallel directions, and compact finite differences in the wall-normal (y) direction. The spatial resolution is $\Delta x^+ = 8$, $\Delta z^+ = 4$ in the streamwise and spanwise directions before

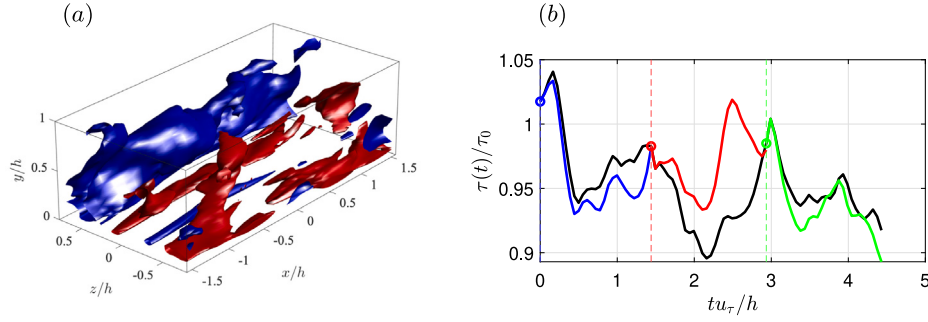


Fig. 1. (a) Instantaneous visualization of the channel flow. The isosurfaces correspond to streamwise velocity fluctuations equal to $\pm 2u_\tau$ (high velocity streak in red, and low-velocity streak in blue). (b) Instantaneous evolution of $\langle \tau_b^k \rangle(t)$ (in black) and of $\langle \tau_f^k \rangle(t)$ for three consecutive episodes, coloured in blue, red and green, respectively.

dealiasing (where the $+$ superscript indicates variables normalized in wall units, using u_τ and ν). In the wall-normal direction, the non-uniform grid provides $\Delta y^+ = 0.4$ at the walls and $\Delta y^+ = 5.4$ at the centre of the channel. For convenience, the wall at $y = 0$ will be referred to as bottom wall. Fig. 1(a) shows an instantaneous snapshot of the bottom half of the channel simulation.

The numerical database consists of $N_{epi} = 35000$ episodes. In each episode two simulations are run using the same initial condition: a plain turbulent channel (referred to as the base flow), and a turbulent channel with a localized volumetric forcing near the bottom wall (referred to as the forced flow). Both simulations are run for $1.5h/u_\tau$, and the final state of the base flow is used as initial condition for the next episode (see Fig. 1b). Hence, each episode allows a direct comparison of the flow evolution with and without the forcing.

The volumetric force used in the forced cases only acts on the vertical direction, given by

$$f_y(x, y, z, t) = \begin{cases} f_0 \frac{y}{h} \exp\left(\frac{-4(x^2 + y^2 + z^2)}{L_f^2}\right) & \text{if } t_0 \leq t \leq T_f, \\ 0 & \text{else,} \end{cases} \quad (2.1)$$

with $f_0 = 8u_\tau^2/h$, $L_f^+ = 100$ and $T_f = 0.3h/u_\tau$ ($T_f^+ = 50$). Note that the force in Eq. (2.1) is localized, both in time and space. Indeed, the values of the parameters controlling the volumetric force are chosen so that they produce a maximum effect on the flow while maintaining a localized character (i.e., without the forcing directly affecting the whole length of the channel). The interested reader can find more details about the effect of the different parameters appearing in Eq. (2.1) in [19].

2.2. Neural networks

Two different types of neural networks were used in this study, namely MultiLayer Perceptron (MLP) and Convolutional Neural Network (CNN). While MLPs are universal estimators for multivariate functions, CNNs are deep learning models used for analysing visual data. They extract features from images using specialized layers and filters, enabling them to learn hierarchical representations and spatial derivatives. CNNs excel in tasks like image classification and object detection by aggregating and classifying the learned features. They have been successfully used in flow reconstruction problems to extract kinematic information (i.e., flow velocities) from concomitant wall measurements [22], which justifies the interest in checking if they are able to extract more information about the dynamics of the system (i.e., the evolution of the skin friction after actuation) than MLPs.

First, common aspects in the pipeline of MLPs and CNNs are described, focusing afterwards in the differences among them. For both of them, the output layer has a single neuron, with a linear activation function, as it is customary on regression NNs.

The input for the NNs is computed from the values of wall-pressure and viscous stresses, $p^k(x, z)$ and $\tau^k(x, z)$, at the initial condition of the

k th episode. Since the NNs require a discrete number of inputs, we divide the bottom wall surface ($L_x \times L_z$) in a Cartesian grid of $n \times n$ sensors. The values of the wall-pressure and viscous stress on the i th sensor, $p_s^k(x_i, z_i)$ and $\tau_s^k(x_i, z_i)$, correspond to the averaged values of $p^k(x, z)$ and $\tau^k(x, z)$ within the i th cell of the grid, centred on (x_i, z_i) .

The values of the pressure and viscous stresses on the sensors are normalized and truncated between 0 and 1 before being fed into the NNs. The normalized variables are thus defined as

$$p_n^k(x_i, z_i) = \frac{1}{2} + \frac{p_s^k(x_i, z_i) - p_0}{\beta_p \sigma_p}, \quad (2.2)$$

$$\tau_n^k(x_i, z_i) = \frac{1}{2} + \frac{\tau_s^k(x_i, z_i) - \tau_0}{\beta_\tau \sigma_\tau}, \quad (2.3)$$

where the mean (p_0, τ_0) and the standard deviations (σ_p, σ_τ) are computed averaging the point-wise wall-pressure and viscous stresses over the whole database of base flows (i.e., space, time, and ensemble average over $N_{epi} = 35000$ episodes). The values of $\beta_p = 2$ and $\beta_\tau = 4.85$ are selected to ensure that the probability of having p_n or τ_n outside the range $0 - 1$ is smaller than 10% for the grid of sensors $n \times n = 1024$. Fig. 2 shows the probability density function (PDF) of the normalized pressure and viscous shear at the sensors for the finer ($n^2 = 1024$) and the coarser ($n^2 = 9$) grids of sensors considered in this study. Different kinds of normalization were also tried out, such as min-max scaling or z-score normalization, but the one presented in here provided the best results.

The objective of the NNs is to predict the time-averaged skin friction change produced by a given action (forcing). Hence, we define the instantaneous skin friction difference between the base and forced case of the k th episode,

$$\Delta\tau^k(t) = \langle \tau_b^k \rangle(t) - \langle \tau_f^k \rangle(t), \quad (2.4)$$

where the brackets $\langle \rangle$ indicate spatial average in the wall-parallel directions. We also define the corresponding time-averaged skin friction difference

$$\overline{\Delta\tau}^k(T_H) = \frac{1}{T_H} \int_0^{T_H} \Delta\tau^k(t) dt, \quad (2.5)$$

where the time averaging period is the predictive horizon, T_H . With these definitions, drag reduction in episode k over a time T_H corresponds to $\overline{\Delta\tau}^k(T_H) > 0$.

We train the NNs to produce a scalar output $\overline{\Delta\tau}^k(T_H)/\tau_0 \cdot 10^4$, where the factor of 10^4 is introduced to facilitate the convergence of the optimization algorithm. The training of the NNs is performed during 100 epochs with batch sizes of 512, using the Adam optimizer to minimize the mean square error between the predicted skin friction change and the DNS output:

$$\mathcal{L}(u_{NN}; u_{DNS}) = \frac{1}{N} \sum_k^n (u_{DNS,k} - u_{NN,k})^2, \quad (2.6)$$

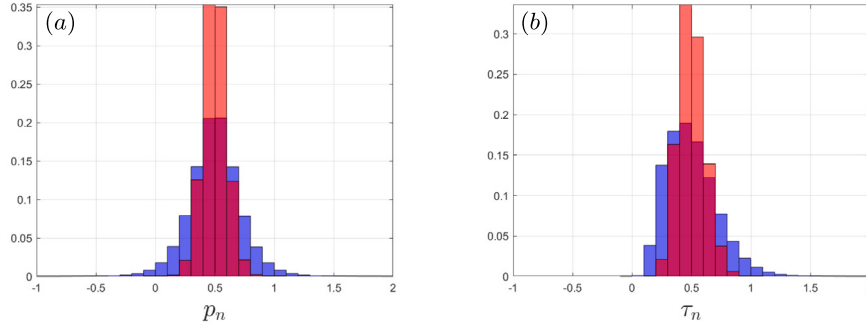


Fig. 2. Probability density functions of (a) p_n and (b) τ_n . Blue for $n^2 = 1024$, red for $n^2 = 9$.

Table 1

Architecture of the MLPs. Number of neurons per hidden layers (N_n) for different number of sensors (N_s), and the corresponding number of parameters (N_p).

N_s	2048	1024	512	128	32	18	9
N_n	32, 16, 16	32, 16, 16	32, 16, 8	32, 8, 4	16, 8, 4	16, 4	5, 4
N_p	66385	33617	17089	4433	705	377	79

Table 2

Architecture of the CNNs. Number of parameters (N_p) for different number of sensors (N_s).

N_s	2048	1024	512	128	32	18	9
N_p	38661	33765	8503	1524	82	48	44

where N is the number of episodes of the training set batch, u_{NN} is the output of the NN regression scheme and u_{DNS} is the ground truth. The database of $N_{epi} = 35000$ episodes is split into training ($N_T = 20000$), validation ($N_v = 5000$) and testing sets ($N_{test} = 10000$).

Once a general overview of the common parameters to both NNs is analysed, we focus on the differences among them, which lies in the internal architecture of the network.

In the MLPs, all hidden layers are fully connected, with a Rectified Linear Unit (ReLU) activation function [32]. The number of hidden layers, and number of neurons in each layer, depends of the total number of sensors, N_s , as shown in Table 1, and result from a hyperparameter tuning optimization to minimize the loss function. For MLPs with pressure and viscous stress sensors, $N_s = 2n^2 = 18, 32, 128, 512, 2048$. For MLPs with pressure sensors only, or with viscous stress sensors only, the total number of sensors is $N_s = n^2 = 9$ or 1024. The maximum density of sensors ($n^2 = 1024$) correspond to having a single sensor for each grid point of the DNS, analogous to sensing performed in the simulations of Lee et al. [15].

With regards to the architecture of CNNs, residual blocks are used, as it is the state of the art in image recognition [33] and turbulence [34]. The number of trainable parameters for each sensor input can be found in Table 2. Note that the number of parameters is reduced for all the sensors with respect to the MLP, as expected due to the usage of filters. The architecture of the CNN used is the same for all N_s , whilst the size and number of filters and neurons is correspondingly adapted as the grid gets coarser, to minimize the loss function for each configuration. The architecture for the CNN of $N_s = 2048$ is shown in Fig. 3, where convolutional layers are shown in yellow, with the right side of the block coloured in dark yellow if the activation function (PReLU) is applied, MaxPooling layers are coloured in blue and dense layers in red, with dark red indicating the (ReLU) activation function. The numbers indicate the output of the convolutional and dense layers. Note that the dense part has only one hidden layer of 10 neurons, with an input layer of 1024 neurons and an output layer of 1 neuron that provides $\overline{\Delta\tau}^k(T_H)/\tau_0 \cdot 10^4$.

In order to understand the decision making of the neural networks, we will analyse their weights and features using different tools. The

wall-signatures being targeted by the NNs will be characterized in terms of the weights of the input layer of the MLP, and the output of the convolutional filters for the CNN (see Section 3.2 below). Besides that, we will use the SHAP method (SHapley Additive exPlanations, see [35]) to evaluate the contribution of the inputs sensors to the prediction of the network. SHAP is a post-hoc method that belongs to the class of eXplainable Artificial Intelligence (XAI) algorithms. It is an Additive Feature Attribution method inspired in cooperative game theory. It creates a simplified surrogate model $g(x)$ to explain the operations of the original model $f(x)$ as:

$$f(x) \approx g(x') = \phi_0 + \sum_{i=1}^M \phi_i x'_i, \quad (2.7)$$

where $x' \in \{0, 1\}^M$ is the simplified input that follows the mapping $x = h_x(x')$, ϕ_0 is referred to as the base value and accounts for the mean output value, and the variables $\phi_i \in \mathbb{R}$ are the SHAP values, computed as:

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f(z') - f(z' \setminus i)], \quad (2.8)$$

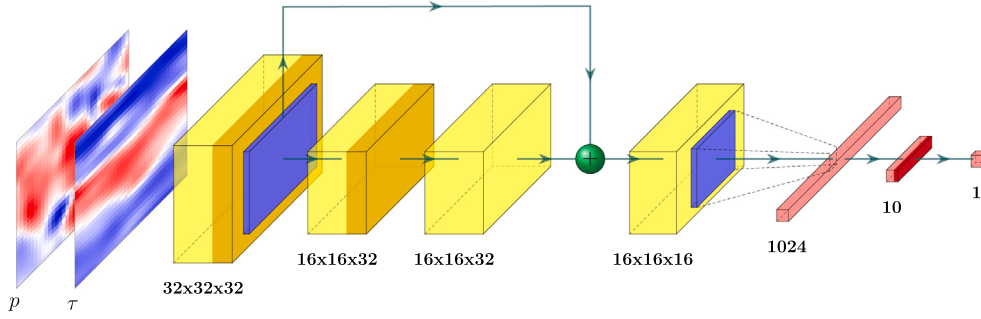
where $|z'|$ is the number of non-zero elements in z' , and $z' \setminus i$ denotes setting $z_i = 0$. This formula calculates the SHAP value (ϕ_i) for each feature (x_i) by considering its contribution across all possible subsets of features (z'), ensuring that the contributions are fairly distributed among the features. In our case, the SHAP value ϕ_i indicates how much the i th sensor reading (i.e., $p_n^k(x_i, z_i)$ and/or $\tau_n^k(x_i, z_i)$) contributes to the $\Delta\tau(T_H)$ predicted by the network for the k th episode. Of the available implementations of SHAP we use here DeepSHAP [36], which combines two XAI algorithms: DeepLIFT and SHAP, and sets $x_i = 1$, optimizing the method to explain deep models.

2.3. Linear stochastic estimation

The second method used here to predict the evolution of the flow after actuation is the Linear Stochastic Estimation (LSE). The main motivation to choose LSE over other linear estimators (such as regressors trained with stochastic gradient descent) is because there is abundant literature on applications of LSE to reconstruct a vector of unknown flow features, $\mathbf{u}$, from an observable, $\mathbf{E}$, leveraging the statistical correlation between them [25,37,38]. For the present analysis, we define the observable as the normalized pressure and viscous stresses on the sensors at the wall (i.e., $\mathbf{E} = [p_n(x_1, z_1), \dots, p_n(x_{N_s}, z_{N_s}), \tau_n(x_1, z_1), \dots, \tau_n(x_{N_s}, z_{N_s})]$). Our unknown is a single element, namely the skin friction difference between base and forced case, averaged up to the predictive horizon (i.e., $u = \overline{\Delta\tau}(T_H)$). The estimation u' is defined as a linear function of the observable,

$$u' = [L] \mathbf{E}, \quad (2.9)$$

where the estimator $[L]$ is a $1 \times N_s$ matrix. This means that for LSE, the number of trainable parameters is $N_p = N_s$.

Fig. 3. Architecture of the residual CNN for $N_s = 2048$ sensors.

The estimator $[L]$ is computed by minimizing the l_2 norm of $(u - u')$ over the training set. This requires computing the autocorrelation matrix of the observables and the cross-correlation matrix of observables and unknowns,

$$[L] \left(\begin{bmatrix} \mathbf{E}_1, \dots, \mathbf{E}_{N_T} \end{bmatrix} \begin{bmatrix} \mathbf{E}_1, \dots, \mathbf{E}_{N_T} \end{bmatrix}^T \right) = \left\{ u_1, \dots, u_{N_T} \right\} \begin{bmatrix} \mathbf{E}_1, \dots, \mathbf{E}_{N_T} \end{bmatrix}^T. \quad (2.10)$$

These matrices are computed using the $N_T = 20000$ episodes of the training set, for the sensor configurations reported in Table 1 and for different values of the predictive horizon T_H . Note that the LSE has no need for a validation set, which could be used to increase N_T . However, in this work we choose to train the LSE with the same episodes used to train the NNs. Consequently, both methods are trained with the same amount of data (i.e., same N_T), ensuring a fair comparison between them in section Section 3.

Finally, note that with sufficient data, the LSE estimator should be equivalent to the MLP or the CNN with linear activator functions. In other words, the differences between LSE and MLP/CNNs arise from the non-linearities introduced by the activator functions (which are reflected in the larger ratio of N_p/N_s for MLP/CNN than for LSE, see Tables 1 and 2) and from the differences in the optimization method (Least Squares for LSE, Adams optimizer for MLP/CNN).

3. Results

In this section, we analyse the performance of LSE, MLP and CNN in predicting the evolution of the skin friction on the forced flow (Section 3.1), evaluating the effect that the time horizon of the prediction and the number of sensors at the wall have on the accuracy of the predictions. We also study the structure of the estimator $[L]$ and the distribution of weights and filters on the trained NNs (Section 3.2) to identify which flow patterns are being targeted by the LSE and the NNs, respectively. Finally, we delve into the reasons behind the differences in the performance of MLP and CNN (Section 3.3).

3.1. Prediction of the forced flow

We first evaluate the effect that T_H , N_s and the type of sensors (pressure and shear, only pressure, or only shear) have on the ability of the NNs and LSE to predict the value of $\Delta \tau(T_H)$. In other words, we test their performance answering the question *how much will the actuation increase or reduce the skin friction averaged over a time T_H* ? To measure this performance we use the Pearson correlation coefficient (ρ). Perfect regressors have $\rho = 1$, while uninformative regressors have $\rho = 0$, since we only consider positive correlation in this study.

Fig. 4a shows the variation of ρ with T_H , for MLPs and LSE with $N_s = 2048, 128$ and 18, using pressure and viscous shear stresses sensors. As expected, the correlation coefficient drops gradually as T_H increases for all estimators. For very short times ($T_H = 0.2h/u_\tau$) ρ is ≈ 0.9 (i.e., quasi-perfect regressor), but for $T_H \gtrsim h/u_\tau$ ρ drops to values below

0.4. For $T_H \gtrsim 1.5h/u_\tau$ both estimators are basically random regressors ($\rho \rightarrow 0$). This behaviour is consistent with Pastor et al. [19], who reported that the effect of the forcing on $\Delta \tau$ becomes uncorrelated with the forcing for times of the order of the eddy turn-over time (h/u_τ).

Interestingly, Fig. 4a shows no differences in the performance of MLPs and LSE when $N_s = 18$ and 128, for all values of T_H . The differences between MLPs and LSE are only apparent for $N_s = 2048$, with the LSE with $N_s = 2048$ having the worst performance of all estimators, significantly below the LSE with $N_s = 18$ and 128. As discussed in Section 3.2.1, this behaviour can be linked to a lack of convergence of the estimator $[L]$ for $N_s = 2048$ with the value of $N_T = 20000$ considered here.

Fig. 4b shows the effect of varying the number of sensors on the performance of the MLP. When both pressure and viscous shear stress sensors are used, reducing N_s from 2048 to 128 has little effect on ρ . This suggests that the low-pass-filtering of $p(x, z)$ and $\tau(x, z)$ associated with a reduction of N_s does not affect the ability of the MLP to identify dynamically-relevant flow features. Note that for $N_s = 2048$ the sensor size equals the spatial resolution of the DNS, while for $N_s = 128$ the sensor size is $l_{x,s}^+ \times l_{z,s}^+ = 65 \times 32$. The performance of the MLP starts to degrade for $N_s \lesssim 32$ (i.e., $l_{x,s}^+ \times l_{z,s}^+ \gtrsim 130 \times 65$), although the loss in ρ is small (i.e., ρ is 0.03 less for $N_s = 18$ than for $N_s = 2048$).

Fig. 4b also shows that the performance of the MLP with $N_s = 32 = 2 \cdot 4^2$ depends on the alignment of the sensors with the actuator. Staggering the sensors with respect to the actuator results in a decreased performance, lower than that of aligned sensors with $N_s = 32$ and $N_s = 18$. For $N_s \geq 128$, the alignment of the sensors and the actuator does not affect significantly to ρ , probably because the size of the sensors ($l_{z,s}^+ \lesssim 32$) is small enough to accurately position flow features (near wall streaks and vortices, with spanwise sizes and separation of the order of 100 wall-units) with respect to the actuator ($L_f^+ = 100$).

When only pressure sensors are considered, Fig. 4b shows that the performance of the MLP is drastically reduced (green and magenta symbols, 0.1 to 0.4 lower ρ than MLP with pressure and shear sensors in red). On the other hand, the performance of the MLP trained only with shear sensors is very similar to the performance of the MLP trained with pressure and shear sensors (black and blue symbols, 0.01 to 0.04 lower ρ than red symbols). This result agrees with Encinar and Jiménez [25], who showed that a LSE reconstruction of the streamwise velocity component using only $\tau(x, z)$ can recover approximately 75% of the kinetic energy associated to the streamwise velocity in the buffer region, with marginal contributions from wall-pressure and the spanwise shear (not considered here).

Overall, the results of Fig. 4 suggest that the proposed MLPs are failing to capture the non-linear dynamics of the evolution of the perturbation introduced by the forcing. In the near-wall cycle, the growth and amplification of the near-wall streaks is a linear process [39], which pertains relatively large structures. The non-linear dynamics are important in the bursting and regeneration of the streaks [40], which involve smaller scale structures (i.e., the vortices). Hence, it is expected that taking advantage of the non-linear process involved in the

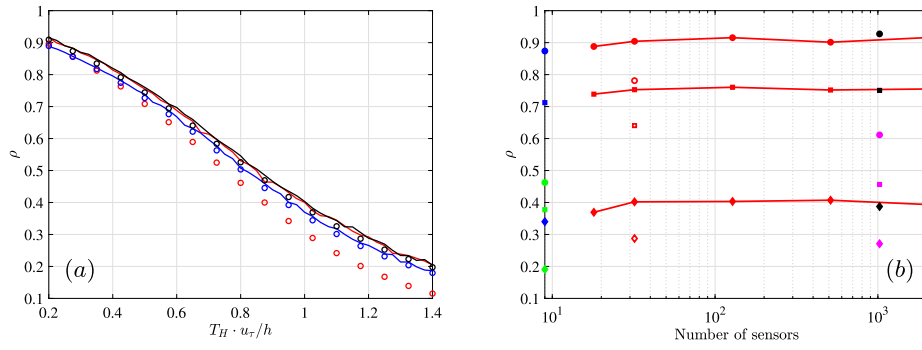


Fig. 4. (a) Pearson correlation coefficient (ρ) versus the time horizon, T_H . Red for $N_s = 2048$, black for $N_s = 128$, blue for $N_s = 18$. Continuous line for MLP, circles for LSE. (b) Effect of the type and number of sensors, N_s , on the performance of the MLP. Symbols indicate time horizons: circles for a $T_H = 0.2h/u_\tau$, squares for $T_H = 0.5h/u_\tau$ and diamonds for $T_H = 1.0h/u_\tau$. Colours indicate type of sensors: red for MLPs using both shear and pressure sensors, blue for $N_s = 9$ shear sensors, green for $N_s = 9$ pressure sensors, black for $N_s = 1024$ shear sensors and magenta for $N_s = 1024$ pressure sensors. Solid/open symbols for sensors aligned/staggered with actuator.

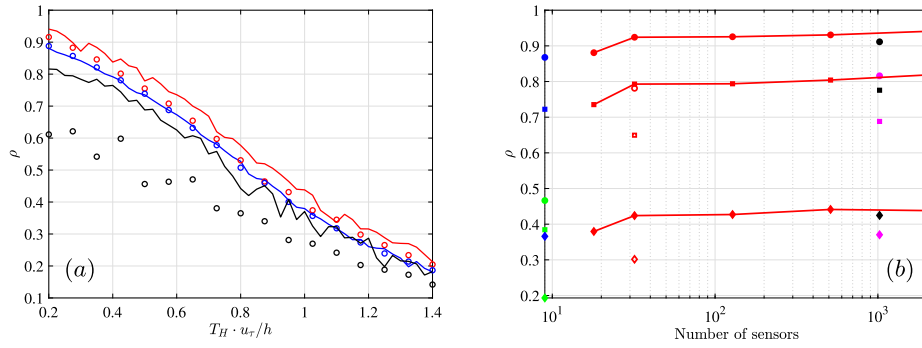


Fig. 5. (a) Pearson correlation coefficient (ρ) versus the time horizon, T_H . Red for $N_s = 2048$, black for $N_s = 1024$ pressure sensors, blue for $N_s = 18$. Continuous line for CNN, circles for MLP. (b) Effect of the type and number of sensors, N_s , on the performance of the CNN. Symbols indicate time horizons: circles for a $T_H = 0.2h/u_\tau$, squares for $T_H = 0.5h/u_\tau$ and diamonds for $T_H = 1.0h/u_\tau$. Colours indicate type of sensors: red for CNNs using both shear and pressure sensors, blue for $N_s = 9$ shear sensors, green for $N_s = 9$ pressure sensors, black for $N_s = 1024$ shear sensors and magenta for $N_s = 1024$ pressure sensors. Solid/open symbols for sensors aligned/staggered with actuator.

interaction between the forcing and the near-wall cycle requires non-linear estimators and sufficiently small sensors. However, the virtually identical performance of LSE and MLPs in Fig. 4a and the lack of improvement on ρ of the MLPs for $N_s > 128$ in Fig. 4b show that either the benefits of including non-linearities in the control are small, in agreement with the results reported by Kim [16], or the MPL is failing at including the non-linear effects of the forcing on $\Delta\tau$. In any case, the comparable performance of LSE and MLP suggests that the forcing interacts with the near-wall cycle mostly through the transient growth of the near-wall streaks (i.e., the linear part of the near-wall cycle), with maybe a weak effect on the bursting and regeneration process.

Fig. 5a shows the variation of ρ with T_H , for MLPs and CNNs with $N_s = 2048, 18$ using pressure and viscous shear stresses sensors, and with $N_s = 1024$ using only pressure sensors. For the configurations using pressure and shear stresses, the CNN outperforms the MLP, specially for the highest sensor density, $N_s = 2048$. Moreover, the CNN using $N_s = 1024$ pressure sensors achieve remarkable performances (around 85% of the value obtained by CNNs using shear and pressure), especially when compared with the MLPs using pressure sensors only (which only achieve around 50% of performances of MLPs using shear and pressure sensors). This suggests that CNNs with sufficiently high sensor densities are able to capture some of the non-linearities discussed in the previous paragraph, explaining their superior performance with respect to LSE and MLP. This result is in agreement with the flow reconstructions of the near-wall region by Nakamura et al. [23], and their comparison of linear and non-linear CNNs. In all cases, the increase in performance is gradually lost for longer T_H , in line with the loss of causality between the action and the skin friction reduction as the temporal horizon becomes longer [19].

Fig. 5b presents the effect of varying the number of sensors on the performance of the CNN. It shows that the increase in performance

for the case of using only pressure sensors only applies to the case of 1024 sensors, i.e. when we have full resolution. For the case using only 9 pressure sensors, as well as for the cases using $N_s = 18$ and $N_s = 9$ shear sensors, the performance of CNNs and MLPs is virtually the same (Figs. 4b and 5b). Given these results, it seems that the overperformance of the CNN architecture over the MLPs is related to the effectiveness of the convolutional filters in capturing the small scale non-linearities in the pressure and shear fields at the wall. When the sensor density decreases this spatial information is lost, and the advantage of CNNs over MLPs and LSE disappears.

To further characterize the regression provided by the NNs and the LSE, Fig. 6 depicts the joint PDF of the ground truth (i.e., the actual value of $\Delta\tau(T_H)$) and the outputs of the regression scheme (u_{MLP} for the MLP, u_{CNN} for the CNN and u_{LSE} for the LSE). The figures show the joint PDFs for different T_H , and for two sensors configurations.

First, let's consider the joint PDFs and $T_H = 0.2h/u_\tau$, shown in Fig. 6a,d and g. The three joint PDF show a behaviour consistent with a quasi-perfect regressor, with most of the joint PDF's mass along the line $u = \Delta\tau(T_H)/\tau_0$ (black dashed line). As T_H increases (Fig. 6b,e and h), the joint PDFs become broader, and the correlation coefficients become smaller. At the longest T_H considered (Fig. 6c, f and i) the loss of correlation becomes more acute, and the joint PDF begins to resemble a circle. Note that the joint PDF of MLPs (top row of Fig. 6) with complete information from the wall ($N_s = 2048$ pressure and viscous shear stress sensors) are very similar to those with minimal information from the wall ($N_s = 9$ viscous shear stress sensors), in agreement with the values of ρ shown in Fig. 4. The same is not true for the joint PDFs of the CNNs (middle row in Fig. 6), which show consistently better performance for the finer grid of sensors.

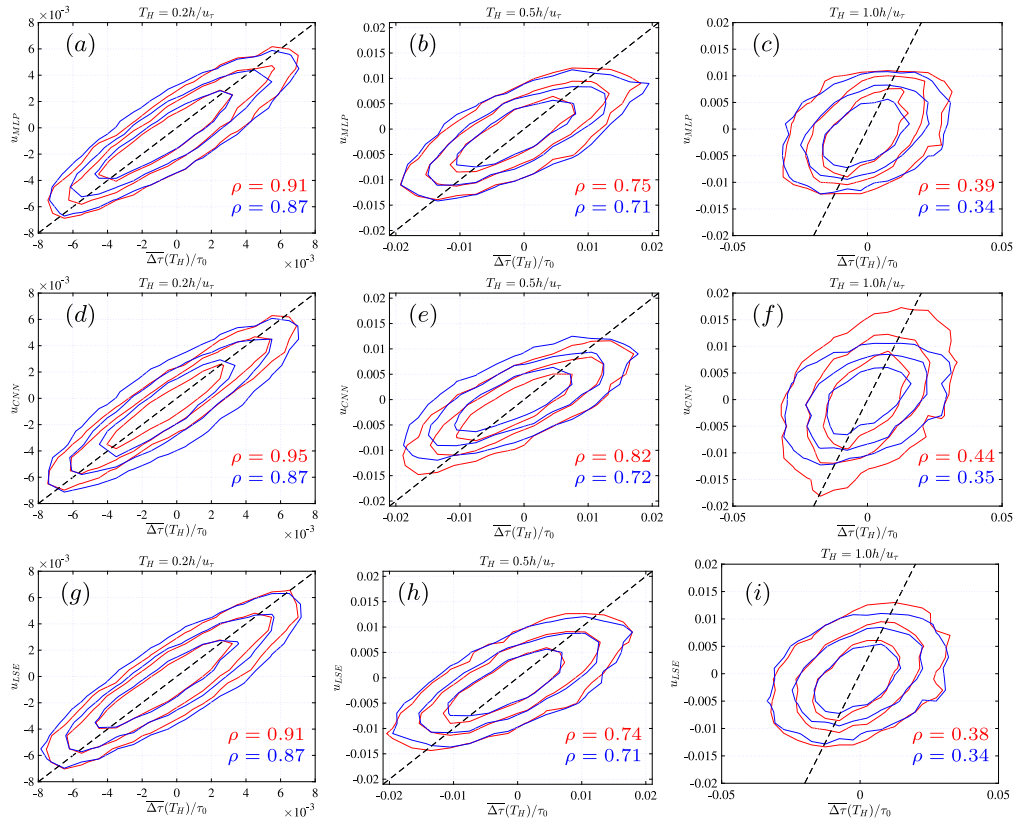


Fig. 6. Joint PDF of the estimation provided by the MLPs (upper row) u_{MLP} , CNNs (middle row) u_{CNN} , LSE (bottom row) u_{LSE} and the ground truth, $\overline{\Delta\tau(T_H)}$. The contours plotted contain 50, 80 and 95% of the PDF's mass. Two sensor configurations are considered: $N_s = 2048$ (MLPs and CNNs) and $N_s = 128$ (LSE) with pressure and viscous shear stress in red, $N_s = 9$ with viscous shear stress sensors only in blue. (a,d,g) $T_H = 0.2h/u_\tau$, (b,e,h) $T_H = 0.5h/u_\tau$, (c,f,i) $T_H = 1.0h/u_\tau$. The black dashed line corresponds to $u = \overline{\Delta\tau(T_H)}/\tau_0$.

3.2. Wall signatures

In this section, we characterize the wall signatures identified by the LSE, MLP and CNN as the most responsive to the actuation in terms of changes in the skin friction.

3.2.1. Linear stochastic estimation

Since the LSE is a linear operator, the wall signatures (i.e., the patterns) that are being targeted by the estimator can be inspected by plotting the values of the elements of the estimator $[L]$ in Eq. (2.9). The observable $\mathbf{E}$ can be split into $\mathbf{E}_p$ and $\mathbf{E}_\tau$ for the data from pressure and shear stress sensors, respectively. Consequently, the estimator can be split into $[L_p]$ and $[L_\tau]$,

$$\mathbf{u}' = [L]\mathbf{E} = [L_p]\mathbf{E}_p + [L_\tau]\mathbf{E}_\tau. \quad (3.1)$$

Fig. 7 shows the effect of the number of sensors (N_s) and the predictive horizon (T_H) on $[L_p]$ and $[L_\tau]$. The figure shows the grid of sensors on the bottom wall of the channel, colouring the area associated to each sensor with the normalized values of the estimators ($[L_\tau]_n = [L_\tau]/\sigma(L_\tau)$ and $[L_p]_n = [L_p]/\sigma(L_p)$, where $\sigma(L_\tau)$ and $\sigma(L_p)$ are the standard deviations of the elements of each estimator). As a consequence, the plot does not show the relative importance of pressure sensors over shear stress sensors.

For $N_s = 18$ (Fig. 7, top row) and 128 (Fig. 7, middle row), the wall signatures observed in $[L_\tau]_n$ are consistent with a high-velocity streak ($\tau > 0$) at the same spanwise position as the actuator. Similar wall signatures are observed for the LSE using only shear stress sensors (not shown). This wall-signature is consistent with an opposition control scenario, in which a positive f_y would act lifting up the streak, reducing the skin friction (i.e., positive $\Delta\tau$). For $T_H = 0.2h/u_\tau$, the sensors downstream of the actuator have zero weights, indicating that for the forcing to be successful it only needs a short high speed streak just

upstream of the actuator (remember that the channel is periodic in the streamwise direction). For $T_H = 0.5h/u_\tau$ and $T_H = h/u_\tau$, the LSE seems to target longer streaks, covering the whole length of the channel.

On the other hand, the wall signatures obtained from $[L_p]_n$ for $N_s = 18$ and 128 indicate that the LSE is targeting a streak with a positive pressure gradient along the streamwise direction, which could be interpreted as a decelerating streak. It should be noted that, as previously discussed, the importance of the pressure sensors on the estimation provided in the LSE is small. For $N_s = 18$ and $N_s = 128$, the Pearson correlation coefficient of the LSE using only shear stress sensors is $\lesssim 1\%$ lower than when using pressure and shear stress sensors, for the range of T_H considered here. This suggests that the wall signatures observed in the pressure sensors (i.e., the positive pressure gradient along the streak) might not be very relevant, and that the response of streaks with positive and negative pressure gradients to the forcing are very similar.

While the wall signatures obtained for $N_s = 18$ are very clear for all predictive horizons, those obtained for $N_s = 128$ become more noisy as T_H increase. For $N_s = 2048$ (Fig. 7, bottom row), the noise impedes the identification of coherent wall signatures in $[L_p]_n$ and $[L_\tau]_n$. The reason for this behaviour is statistical convergence: $N_T = 20000$ is insufficient training data to converge $[L]$ when N_s and T_H increases. This can be quantified computing the variability of $[L]$ for different subsets of $N_T = 20000$ training episodes. For $N_s = 18$ and $T_H = 0.2h/u_\tau$ the variability in $[L]$ is $\lesssim 4\%$ of $\sigma(L)$ ($\lesssim 15\%$ for $T_H = h/u_\tau$). For $N_s = 2048$ this variability becomes $\gtrsim 100\%$, irrespective of T_H .

3.2.2. Multilayer perceptron

Compared to the analysis for the LSE, identifying the wall signatures targeted by the MLPs is more complicated. Let's consider the simplest MLP trained in this work ($N_s = 9$, using only shear stress sensors) at $T_H = 0.5h/u_\tau$. The architecture of this MLP consists of an input layer

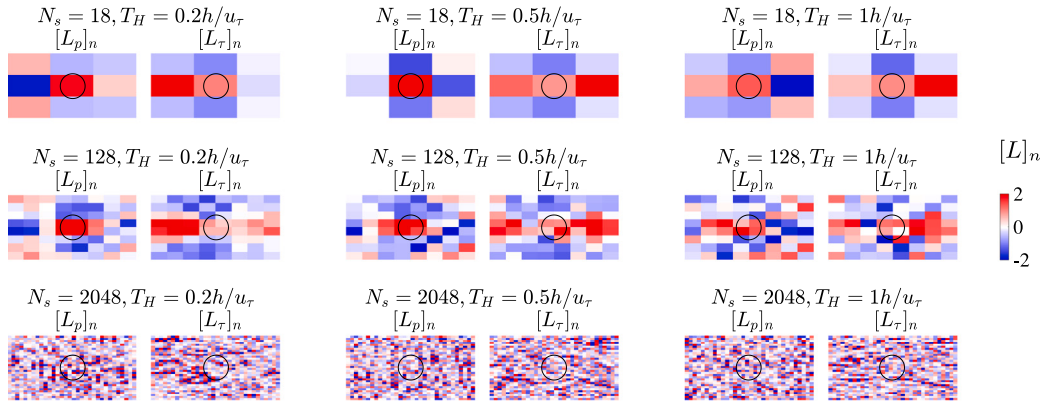


Fig. 7. Spatial distribution of the elements of the estimator $[L]$. Contributions from pressure ($[L_p]_n = [L_p]/\sigma(L_p)$) and shear sensors ($[L_\tau]_n = [L_\tau]/\sigma(L_\tau)$) are plotted side by side, normalized with their corresponding standard deviations. T_H increases from left to right. N_s increases from top to bottom. The black circles show the position of the actuator.

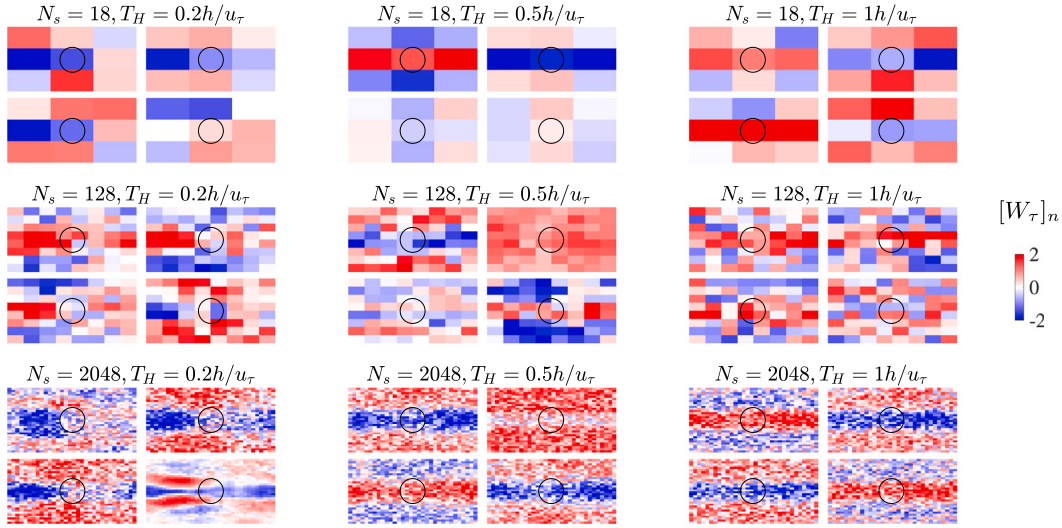


Fig. 8. Effect of N_s and T_H on the spatial distribution of the normalized weights of the input layer associated to stress sensors, $[W_\tau]_n$. For each NN, the four input layer neurons shown have the largest weights on the first hidden layer. Weights are normalized with their standard deviation. The black circles show the position of the actuator.

with 9 neurons, two hidden layers with 5 and 4 neurons, and an output layer with 1 output. Mathematically, the NN can be expressed as

$$\eta = f_3([W_{4 \times 1}^3]\{a_2\} - B_{1 \times 1}) \quad (3.2)$$

$$\{a_2\} = f_2([W_{4 \times 5}^2]\{a_1\} - \{B_{4 \times 1}\}) \quad (3.3)$$

$$\{a_1\} = f_1([W_{5 \times 9}^1]\{\tau_s\} - \{B_{5 \times 1}\}) \quad (3.4)$$

where $[W]$ are the weight matrices, $\{B\}$ are the bias vectors, and f_i are the corresponding activation functions. Each row of $[W_{5 \times 9}^1]$ represents the weights that are applied to the input of the MLP (i.e., $\{\tau_s\}$), analogous to the elements of $[L]$ in the LSE. But in the MLP the output of the first layer (i.e., $\{a_1\}$) is then weighted by the next layer, and so on). Hence, in order to evaluate the wall signatures that are being targeted by the MLP, we will look at the spatial distributions of the weights of the first layer (i.e., the rows of $[W^1]$) associated to the 4 more important neurons of the second layer (i.e., the indices corresponding to the four columns of $[W^2]$ with the largest norm).

Fig. 8 shows the four strongest wall-signatures of the MLPs for different values of N_s and T_H . The MLPs selected for the plot have pressure and shear sensors because they are more relevant for the prediction than pressure sensors (see discussion in Section 3.1). Overall, the effects of N_s and T_H of the wall signatures targeted by the MLPs are similar to those shown for the LSE in Fig. 7. For $T_H = 0.2h/u_\tau$ the MLP targets a streak just upstream of the actuator, with low (or incoherent) weights

associated to the sensors downstream of the actuator. For longer T_H the weights shown streaky patterns, becoming less clear as T_H and N_s increase. Of note, the wall signatures of the MLP with $N_s = 128$ are also show streaks, albeit more noisy than those appearing on $N_s = 18$ and $N_s = 2048$.

Overall, the results in Fig. 8 show that the wall signatures targeted by the MLPs are the same high-speed streaks targeted by the LSE (see Fig. 7). None of the input layer neurons of the MLPs targets small scale structures, supporting the conclusion of Section 3.1 that both LSE and MLPs are focusing on the linear response of the flow to the actuation. The results presented here are also consistent with the wall signatures obtained in the conditional analysis of Pastor et al. [19], and with the opposition control strategy of Choi et al. [17]. They are also consistent with the weight distributions computed by Fukagata and Kasagi [41] using sub-optimal control theory, for a linear control based on blowing/suction at the wall.

3.2.3. Convolutional neural network

The output of the convolutional layer of a CNN is

$$F_{i,j} = \sum_p \sum_q I_{i-p,j-q} K_{p,q}, \quad (3.5)$$

where $I \in \mathbb{R}^{i_1 \times i_2}$ is the input to the convolutional layer, $K \in \mathbb{R}^{k_1 \times k_2}$ is the kernel (or filter) and contains the trainable parameters of the network, and F is the feature map. As it can be checked in Eq. (3.5), there is

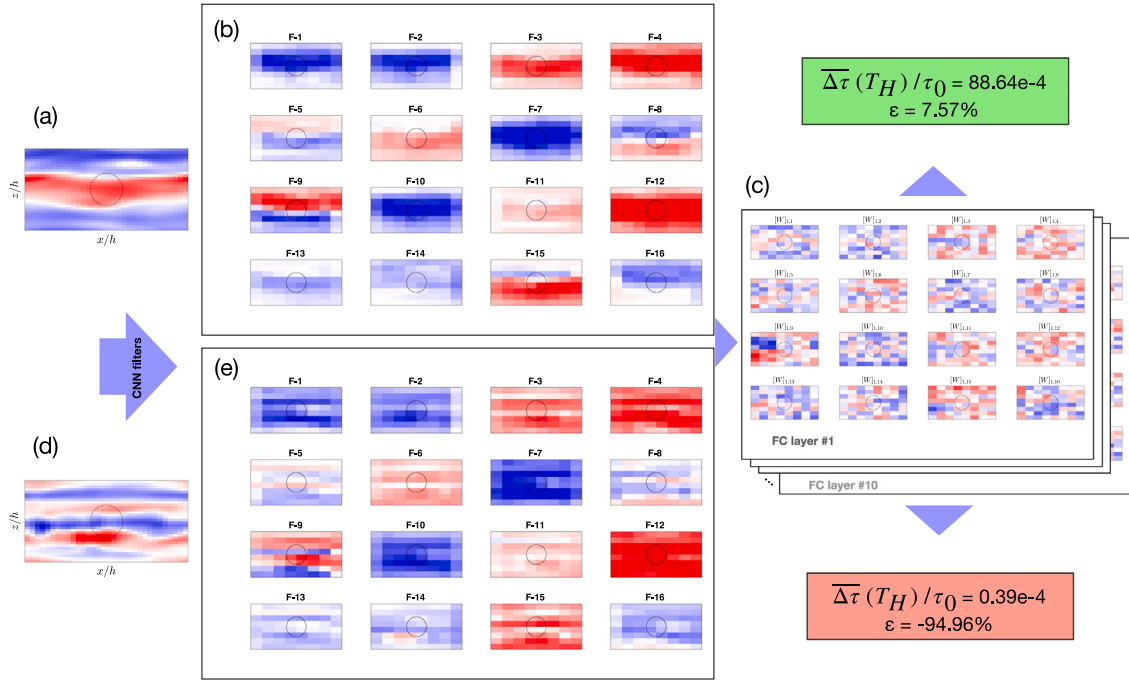


Fig. 9. Analysis of the CNN with $N_s = 1024$ shear stress sensors at $T_H = 0.2h/u_\tau$. (a,d) Shear stress field at the input layer. (b,e) Feature maps F at the exit of the convolutional layer of the CNN. (c) Corresponding weights $[W]_{i,j}$ of the most important neuron in the first layer of the dense part of the CNN. Panels (a,b) and the green box correspond to the episode with the lowest prediction error, $\varepsilon = 7.57\%$. Panels (d,e) and the red box correspond to the episode with the largest prediction error, $\varepsilon = -94.96\%$.

one feature map per filter used, so due to the fact that more than one filter is trained for each layer, the output of each layer consists on a set of stacked feature maps. The fact that the same filter is applied to the whole input field explains why CNNs are able to exploit the spatial structure better than MLPs, which treat each input sensor as an independent input. Moreover, the convolutional layers of the CNNs are invariant to translation, hence when they learn a pattern they can detect it anywhere in the field, unlike MLPs that only detect structures relative to the location of the actuation.

Fig. 9 depicts the feature maps output by the CNN after the convolutional layers (i.e. after the filters are applied) and how these maps are weighted by the most important neuron of the hidden layer of the dense part of the network, for the best and worst prediction of the CNN for $N_s = 1024$ shear sensors, respectively. Inspection of the filtered fields (panels b and e) reveals that features F-8, F-9 and F-15 accounts for derivatives of the input field with respect to z , while F-10 and F-4 seem to consist of spatial averages of the input field. Inspection of the weights (panel c) shows that the weights associated to F-9 are the largest, specially upstream of the actuator. This suggests that the CNN depicted in Fig. 9 is trying to locate a near wall streak aligned with the actuator by focusing on the value of $d\tau/dz$ just upstream of the actuator. For the case in which the input field is a streak aligned with the actuator (panel a), this results in a large drag reduction and a very small error in the prediction of the CNN (green panel in Fig. 9) For the case in which the input field consists of smaller streaks, not aligned with the actuator (panel d), the CNN prediction is completely off (94% error in the red panel of Fig. 9).

Interpreting how the CNN uses the information of pressure sensors is more challenging. Fig. 10 shows the best and the worst predictions for the CNN with $N_s = 1024$ pressure sensors. The pressure fields have smaller scales than the shear stress fields, and the corresponding filtered fields are more complex as well. Visual inspection of the filtered maps in Fig. 10b and e shows that most of the maps are combinations of displacements, averages and derivatives, without any clear stereotypical filter. Similar conclusions can be obtained from the weights distributions (Fig. 10c), which do not show any feature that clearly

dominates over the others. Overall, the results presented in Fig. 10 show that the present architecture results in non-interpretable CNNs for the pressure, a problem that is well described in the literature of NNs and CNNs.

3.3. MLP vs. CNN

In this section we perform a set of analysis in order to better understand why the CNN for $N_s = 1024$ shear or pressure sensors outperforms the MLP ones, trying to draw some mechanistic conclusion from the analysis.

First, we will evaluate the ability of MLPs and CNNs to localize in space a given wall signature with respect to the actuator. This is done by performing a sensitivity analysis of the output of the NN with respect to shifts of the input fields in the horizontal directions (i.e., streamwise and spanwise). If the output of the corresponding NN (i.e., MLP or CNN) to the k th field is

$$u_{NN}^k = F_{NN}(\tau_n^k(x_i, z_i), p_n^k(x_i, z_i)), \quad (3.6)$$

we can define the output to the shifted fields as

$$u_{NN}^k(r_x, r_z) = F_{NN}(\tau_n^k(x_i + r_x, z_i + r_z), p_n^k(x_i + r_x, z_i + r_z)), \quad (3.7)$$

where the shifts (r_x, r_z) are applied to the input fields taking into account that the horizontal directions are periodic. We then evaluate the sensitivity of the NN to (r_x, r_z) using the root mean square of $u_{NN}(r_x, r_z) - u_{NN}(0, 0)$, averaged over the fields of the test set:

$$RMS_{NN}(r_x, r_z) = \left[\frac{1}{N_{test}} \sum_{k=1}^{N_{tests}} (u_{NN}^k(r_x, r_z) - u_{NN}^k(0, 0))^2 \right]^{1/2} \quad (3.8)$$

Fig. 11 shows $RMS_{CNN}(r_x, r_z)$ and $RMS_{MLP}(r_x, r_z)$ for different values of T_H , and for two sensor configurations: $N_s = 1024$ pressure sensors, and $N_s = 1024$ shear sensors. Overall, both types of NNs are more sensitive to spanwise shifts than to streamwise shifts. Indeed, the streamwise sensitivity is only apparent for $r_z = 0$. The sensitivity of CNNs and MLPs are very similar when shear sensors are considered, but large differences are observed for NNs with pressure sensors only.

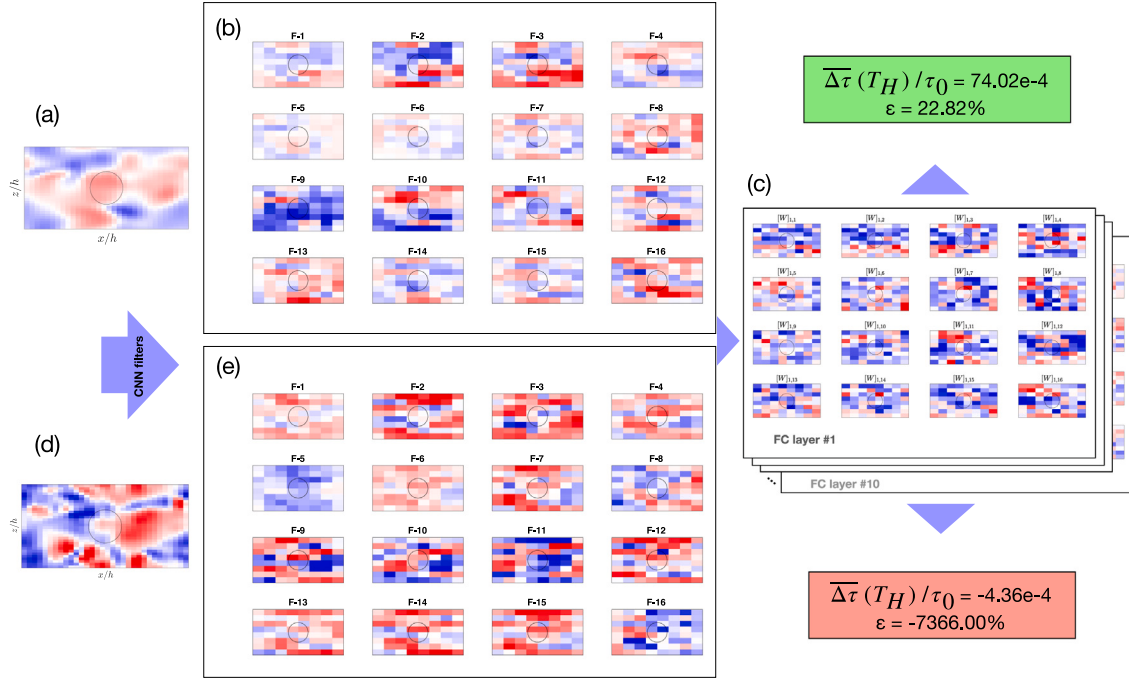


Fig. 10. Analysis of the CNN with $N_s = 1024$ pressure sensors at $T_H = 0.2h/u_\tau$. (a,d) Pressure field at the input layer. (b,e) Feature maps F at the exit of the convolutional layer of the CNN. (c) Corresponding weights $[W]_{i,j}$ of the most important neuron in the first layer of the dense part of the CNN. Panels (a,b) and the green box correspond to the episode with the lowest prediction error, $\epsilon = 22.82\%$. Panels (d,e) and the red box correspond to the episode with the largest prediction error, $\epsilon = -7366\%$.

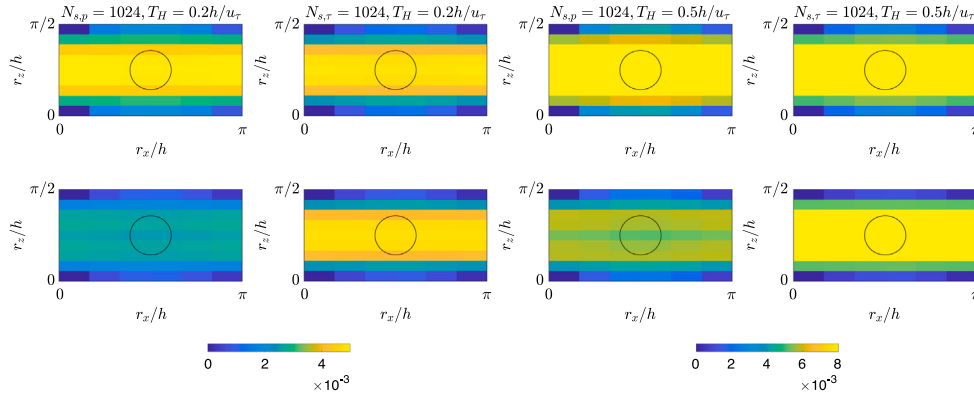


Fig. 11. Effect of (r_x, r_z) on the prediction of the NNs. Top row is RMS_{CNN} , and bottom row is RMS_{MLP} , defined in Eq. (3.8). Two time horizons ($T_H = 0.2h/u_\tau$ and $T_H = 0.5h/u_\tau$) and two sensor configurations ($N_s = 1024$ shear sensors and $N_s = 1024$ pressure sensors) are considered.

While the CNNs with pressure sensors are able to keep a comparable level of sensitivity as the CNNs with shear sensors, the MLPs with pressure sensors have a lower sensitivity. The fact that the MLP has more difficulties in detecting shifts in the pressure fields than the CNN is probably linked to the smaller scales of the pressure fields (i.e., compared with the shear stress field), and is consistent with a lower predictive capability of MLPs with pressure sensors than MLPs with shear sensors or the CNNs with either type of sensor.

Second, we use the SHAP values defined in Section 2.2 to characterize how each sensor contributes to the prediction of the NNs (i.e., which parts of the input field are causal to the predicted value of $\overline{\Delta\tau}(T_H)$). Fig. 12 shows the SHAP values for both the MLP and CNN, for $N_s = 1024$ shear and pressure sensors. The input fields used in this analysis are the same used in Figs. 9–10, corresponding to the fields that produce the best and the worst prediction of each network. By construction $\phi_0 = 0$ in our database, since our low amplitude volumetric forcing is introduced in a statistically steady channel, and has a zero net effect on the skin friction when all episodes are averaged.

Because of this, the SHAP values sum up to the output predicted by the NNs:

$$\overline{\Delta\tau}(T_H)/\tau_0 \cdot 10^4 = \sum_i \phi_i + \epsilon, \quad (3.9)$$

where the small error (i.e., $\epsilon/\phi_i \approx 10^{-4}$) comes from the fact that the algorithm uses a simplified model g for the calculation of the values, as described in Section 2.2.

Several observations can be made of the distribution of the SHAP values shown in Fig. 12. First, the SHAP values show clear structure, which resembles to some extent the input fields. For instance, in the 1st and 2nd row of Fig. 12 (best and worst predictions based on shear sensors) we see that the SHAP values associated to the high and low speed streak regions have large absolute values, while the regions where the streak changes sign have SHAP values close to zero. This further supports the idea that the NNs are trying to identify specific wall signatures. Second, the spatial distribution of the SHAP values of CNN and MLP is similar, which suggests that both networks are deriving their predictions from the same wall signatures. This means that the

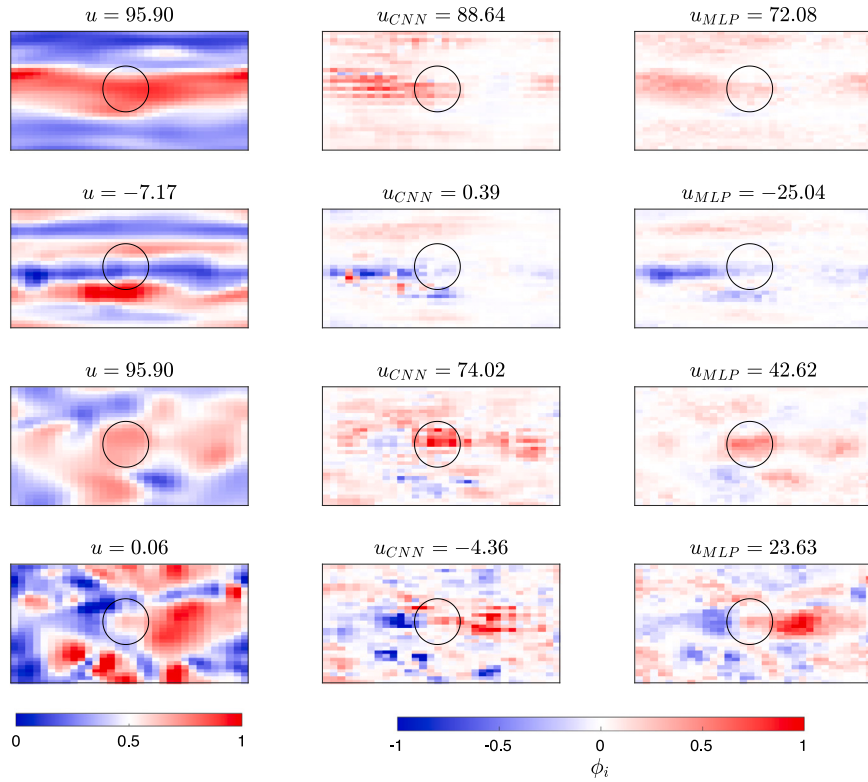


Fig. 12. SHAP values ϕ_i of the CNN (second column) and MLP (third column) for the input fields shown in the first column. SHAP values are computed for $T_H = 0.2h/u_\tau$ and $N_s = 1024$ shear sensors (rows one and two) and pressure sensors (rows three and four). The title of the first column indicates the true value of $\overline{\Delta\tau}(T_H)/\tau_0 \cdot 10^4$, whereas the titles of the second and third column indicate the predicted values for each network.

main difference between CNNs and MLPs is not which structures they are trying to detect, but how good they are in detecting them (i.e., see discussion of Fig. 11) and in weighting their importance for the change of skin friction after the actuation. Third, the SHAP values of the CNNs seem somewhat noisier, with small scale contributions embedded within large scale contributions of the opposite sign. The reason for this is not clear at the moment, especially since both CNN and MLP with $N_s = 1024$ have roughly the same number of degrees of freedom (see Tables 1 and 2). Finally, the SHAP values of shear sensors seem to be larger upstream of the actuator, and lower downstream of the actuator. Conversely, SHAP values for pressure sensors seem to be larger at the location of the actuator or downstream. This disposition of causal shear upstream of causal pressure is reminiscent of the velocity fields conditioned to vortex clusters and to intense Reynolds stress regions in the near wall region [42,43], and it might be related to the fact that the maximum effectiveness of the actuation is probably in the *smooth* part of the streak, where the momentum introduced by the volumetric force is not dispersed by small scale turbulence.

4. Conclusions

In this paper we analyse the capability of three different methods to predict the effect that a localized forcing introduced into a low-Reynolds number turbulent channel flow has on the skin friction drag. The effect on the skin friction is measured directly, running a base case (without actuation) and a forced case (with actuation) from the same initial conditions. The differences between the averaged skin friction of base and forced channels (i.e., $\Delta\tau$) are used to train three different data-driven estimators: a multilayer perceptron (MLP), a convolutional neural network (CNN) and a linear stochastic estimation (LSE). The input signal for all estimators consists of a grid of pressure and shear stress sensors located the wall. Note that while the LSE is a linear method, the neural networks allow a certain degree of non-linearity.

Overall, the performance of LSE and MLP is very similar. The analysis of the wall-signatures targeted by the estimator shows that both methods are looking for high/low velocity streaks. Their predictive capability decays monotonically with the predictive horizon T_H , reaching Pearson correlation coefficients between the ground truth and the predicted data of the order $\rho \approx 0.4$ when $T_H \approx h/u_\tau$, in agreement with the observation of Pastor et al. [19] of the time at which the effect of the localized forcing is lost on the flow. Reducing the size of the sensors improves the predictive capability of LSE and MLPs while $l_z^+ \gtrsim 65$. Smaller sensors do not seem to result in better estimations of the effect of the forcing on the evolution of the averaged skin friction. Finally, LSE and MLP seem to be able to gain substantially more information from shear sensors than from pressure sensors: the Pearson correlation coefficient of a MLP with $N_s = 9$ pressure sensors at $T_H u_\tau/h = 0.5$ is $\rho = 0.39$, while $N_s = 9$ shear sensors yields $\rho = 0.71$, and $N_s = 9$ shear sensors plus $N_s = 9$ pressure sensors yields $\rho = 0.74$.

Interestingly, the performance of the CNN is only similar to MLP and LSE when the sensors are relatively large ($l_z^+ \gtrsim 35$). When the sensors are sufficiently small, CNNs are able to outperform MLP and LSE, both in terms of the values of the Pearson correlation coefficients, and in terms of the sensibility of the estimator with respect to shifts of the input data in horizontal planes. Moreover, CNNs with only pressure sensors were able to perform significantly better than their LSE and MLP counterparts: a MLP with $N_s = 1024$ pressure sensors yields $\rho = 0.45$ at $T_H u_\tau/h = 0.5$, while a CNN in the same conditions yields $\rho = 0.7$. Although not completely unexpected, the better performance of CNN is somewhat surprising, since some of the usual advantages of convolutional filters (i.e., translational invariance and multi-scale capability) are moot in the present setup (i.e., localized actuation and low Reynolds number).

These results suggest that the MLPs are acting as relatively linear estimators, predicting the change in the skin friction after actuation based on the linear amplification (or damping) of the near-wall streaks

by the actuator. From this point of view, the main difference between MLPs and LSE resides in the training algorithm, which is more robust for MLPs than for LSE when the number of sensors is large. On the other hand, CNNs with sufficiently small sensors seem to be able to model part of the non-linear dynamics occurring in the near-wall region, which explains the increased significance of the pressure sensors (i.e., the vertical velocity structures) on the estimation.

Finally, it is interesting to note that all estimators seem to be targeting wall-signatures corresponding to an elongated near wall streak aligned with the actuator. In the LSE and MLP this seems to be enacted varying the signs of the weights associated to each sensor based on their spanwise position. In the CNN, the streak detection seems to be based on a d/dz of the input shear stress field (performed by the convolutional filters) and a weighting operation of this derivative upstream of the actuator (performed in the fully connected layers).

The fact that all estimators, linear and non-linear, target the near wall streak has two important implications: First, in terms of a causality analysis, this fact can be interpreted as the streak being the most *causal* part of the flow for the present actuation [44]. In other words, the trained estimators are able to predict reasonably well what will happen to $\Delta\tau$ when the actuation occurs on a low/high velocity streak, but their accuracy is lower when the actuation occurs elsewhere. This interpretation is consistent with the interventional experiments of Osawa and Jiménez [45] (where the most causal structures near the wall are high-speed streaks), and with the analysis of the Perron–Frobenius operator in minimal turbulent channels by Jiménez [46] (where the growth of the streak is identified as a highly-predictable phase of the log-layer dynamics). This idea is further supported by the SHAP values presented in section Section 3.3, showing that the most informative (i.e., the most causal) regions of the shear stress fields are those corresponding to the streak aligned with the actuator, in the region upstream of the actuator. On the other hand, the most informative/causal region of the pressure field seems to be downstream of the most informative/causal region of the shear field.

Second, from the point of view of extrapolation to higher Reynolds number, it is clear that while the same strategy used here could be used to control a passing near-wall streak over a specific sensor/actuator (scaled to have the same sizes in wall-units as in the present work), the number of sensors/actuators necessary to have a significant impact on the skin friction of a high Re_τ flow should increase accordingly. On the other hand, the existence of a hierarchy of streaks within the logarithmic layer [30] opens the possibility to use a similar strategy as the one shown here to predict the effect of actuation on larger streaks (i.e., scaling the size of sensors/actuators with the size of the streak), paving the way for future investigations into skin friction control in higher Reynolds number flows.

Funding

This work has been supported by the European Research Council, under the COTURB grant ERC-2014.AdG-669505.

CRediT authorship contribution statement

A. Martín-Gil: Writing – original draft, Visualization, Validation, Software, Formal analysis. **O. Flores:** Writing – review & editing, Supervision, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

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

Data availability

Data will be made available on request.

References

- [1] Abbas A, De Vicente J, Valero E. Aerodynamic technologies to improve aircraft performance. *Aerosp Sci Technol* 2013;28(1):100–32.
- [2] Mäkiharju SA, Perlin M, Ceccio SL. On the energy economics of air lubrication drag reduction. *Int J Nav Arch Ocean* 2012;4(4):412–22.
- [3] Brunton S, Noack B. Closed-loop turbulence control: Progress and challenges. *Appl Mech Rev* 2015;67(5). 050801.
- [4] Zhang L, Shan X, Xie T. Active control for wall drag reduction: Methods, mechanisms and performance. *IEEE Access* 2020;8:7039–57.
- [5] García-Mayoral R, Jiménez J. Convolutional-network models to predict wall-bounded turbulence from wall quantities. *J Fluid Mech* 2011;678:317–47.
- [6] Ran W, Zare A, Jovanović M. Model-based design of riblets for turbulent drag reduction. *J Fluid Mech* 2021;906.
- [7] Ricco P, Skote M, Leschziner M. A review of turbulent skin-friction drag reduction by near-wall transverse forcing. *Prog Aerosp Sci* 2021;123:100713.
- [8] Park J, Choi H. Effects of uniform blowing or suction from a spanwise slot on a turbulent boundary layer flow. *Phys Fluids* 1999;11(10):3095–105.
- [9] Kametani Y, Fukagata K. Direct numerical simulation of spatially developing turbulent boundary layers with uniform blowing or suction. *J Fluid Mech* 2011;681:154–72.
- [10] Kasagi N, Hasegawa Y, Fukagata K. Toward cost-effective control of wall turbulence for skin friction drag reduction. In: *Advances in turbulence XII*. Springer; 2009, p. 189–200.
- [11] Cattafesta III L, Sheplak M. Actuators for active flow control. *Annu Rev Fluid Mech* 2011;43:247–72.
- [12] Jimenez J, Kawahara G. Dynamics of wall-bounded turbulence. In: Davidson P, Kaneda J, Sreenivashan K, editors. *Ten chapters in turbulence*. Cambridge, UK: Cambridge University Press; 2013, p. 221–68.
- [13] Stroh A, Hasegawa Y, Schlatter P, Frohnäpfel B. Global effect of local skin friction drag reduction in spatially developing turbulent boundary layer. *J Fluid Mech* 2016;805:303–21.
- [14] Cheng X, Qiao Z, Zhang X, Quadrio M, Zhou Y. Skin-friction reduction using periodic blowing through streamwise slits. *J Fluid Mech* 2021;920.
- [15] Lee C, Kim J, Babcock D, Goodman R. Application of neural networks to turbulence control for drag reduction. *Phys Fluids* 1997;9(6):1740–7.
- [16] Kim J. Control of turbulent boundary layers. *Phys Fluids* 2003;15(5):1093–105.
- [17] Choi H, Moin P, Kim J. Active turbulence control for drag reduction in wall-bounded flows. *J Fluid Mech* 1994;262:75–110.
- [18] Abbassi M, Baars W, Hutchins N, Marusic I. Skin-friction drag reduction in a high-Reynolds-number turbulent boundary layer via real-time control of large-scale structures. *Int J Heat Fluid Flow* 2017;67:30–41.
- [19] Pastor R, Vela-Martín A, Flores O. Wall-bounded turbulence control: statistical characterisation of actions/states. *J Phys: Conf Ser* 2020;1522:012014.
- [20] Milano M, Koumoutsakos P. Neural network modeling for near wall turbulent flow. *J Comput Phys* 2002;182(1):1–26.
- [21] Güemes A, Discetti S, Ianiro A. Sensing the turbulent large-scale motions with their wall signature. *Phys Fluids* 2019;31(12):125112.
- [22] Guastoni L, Güemes A, Ianiro A, Discetti S, Schlatter P, Azizpour H, Vinuesa R. Convolutional-network models to predict wall-bounded turbulence from wall quantities. *J Fluid Mech* 2021;928.
- [23] Nakamura T, Fukami K, Fukagata K. Identifying key differences between linear stochastic estimation and neural networks for fluid flow regressions. *Sci Rep* 2022;12(1):3726.
- [24] Sasaki K, Vinuesa R, Cavalieri A, Schlatter P, Henningson D. Transfer functions for flow predictions in wall-bounded turbulence. *J Fluid Mech* 2019;864:708–45.
- [25] Encinar M, Jiménez J. Logarithmic-layer turbulence: A view from the wall. *Phys Rev Fluids* 2019;4.
- [26] Illingworth S, Monty J, Marusic I. Estimating large-scale structures in wall turbulence using linear models. *J Fluid Mech* 2018;842:146–62.
- [27] Amaral F, Cavalieri A, Martini E, Jordan P, Towne A. Resolvent-based estimation of turbulent channel flow using wall measurements. *J Fluid Mech* 2021;927.
- [28] Jiménez J. How linear is wall-bounded turbulence? *Phys Fluids* 2013;25(11):110814.
- [29] Jiménez J, Moin P. The minimal flow unit in near-wall turbulence. *J Fluid Mech* 1991;225:213–40.
- [30] Flores O, Jiménez J. Hierarchy of minimal flow units in the logarithmic layer. *Phys Fluids* 2010;22(7):071704.
- [31] Vela-Martín A, Encinar M, García-Gutiérrez A, Jiménez J. A low-storage method consistent with second-order statistics for time-resolved databases of turbulent channel flow up to $Re_\tau = 5300$. *J Comput Sci* 2021;56:101476.
- [32] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014, arXiv preprint arXiv:1412.6980.
- [33] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, p. 770–8.
- [34] Cremades A, Hoyas S, Quintero P, Lellep M, Linkmann M, Vinuesa R. Explaining wall-bounded turbulence through deep learning. 2023.

- [35] Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, editors. *Advances in neural information processing systems*. Vol. 30, Curran Associates, Inc.; 2017.
- [36] Chen H, Lundberg S, Lee S-I. Explaining models by propagating Shapley values of local components. In: Shaban-Nejad A, Michalowski M, Buckeridge DL, editors. *Explainable AI in healthcare and medicine: building a culture of transparency and accountability*. Cham: Springer International Publishing; 2021, p. 261–70.
- [37] Adrian R. Stochastic estimation of conditional structure: a review. *Appl Sci Res* 1994;53(3):291–303.
- [38] Beneddine S, Yegavian R, Sipp D, Leclaire B. Unsteady flow dynamics reconstruction from mean flow and point sensors: an experimental study. *J Fluid Mech* 2017;824:174–201.
- [39] Schoppa W, Hussain F. Coherent structure generation in near-wall turbulence. *J Fluid Mech* 2002;453:57–108.
- [40] Jiménez J, Pinelli A. The autonomous cycle of near-wall turbulence. *J Fluid Mech* 1999;389:335–59.
- [41] Fukagata K, Kasagi N. Suboptimal control for drag reduction via suppression of near-wall Reynolds shear stress. *Int J Heat Fluid Flow* 2004;25(3):341–50.
- [42] del Alamo J, Zandonade P, Moser R. Self-similar vortex clusters in the turbulent logarithmic region. *J Fluid Mech* 2006;561:329–58.
- [43] Lozano-Durán A, Flores O, Jiménez J. The three-dimensional structure of momentum transfer in turbulent channels. *J Fluid Mech* 2012;694:100–30.
- [44] Lozano-Durán A, Arranz G. Information-theoretic formulation of dynamical systems: causality, modeling, and control. *Phys Rev Res* 2022;4(2):023195.
- [45] Osawa K, Jiménez J. Causally significant structures in fully developed wall turbulence. In: *Proceedings of 76th annual meeting of the division of fluid dynamics*. 2023, p. J42.00006.
- [46] Jiménez J. A Perron–Frobenius analysis of wall-bounded turbulence. *J Fluid Mech* 2023;968:A10.