

Predicting the temporal dynamics of turbulent channels through deep learning

Giuseppe Borrelli^{a,b}, Luca Guastoni^b, Hamidreza Eivazi^b, Philipp Schlatter^b,
Ricardo Vinuesa^b

^a*Faculty of Aerospace Engineering, Alma Mater Studiorum - University of Bologna, Forlì,
Italy*

^b*FLOW, Engineering Mechanics, KTH Royal Institute of Technology, Stockholm, Sweden*

Abstract

The success of recurrent neural networks (RNNs) has been demonstrated in many applications related to turbulence, including flow control, optimization, turbulent features reproduction as well as turbulence prediction and modeling. With this study we aim to assess the capability of these networks to reproduce the temporal evolution of a minimal turbulent channel flow. We first obtain a data-driven model based on a modal decomposition in the Fourier domain (which we denote as FFT-POD) of the time series sampled from the flow. This particular case of turbulent flow allows us to accurately simulate the most relevant coherent structures close to the wall. Long-short-term-memory (LSTM) networks and a Koopman-based framework (KNF) are trained to predict the temporal dynamics of the minimal-channel-flow modes. Tests with different configurations highlight the limits of the KNF method compared to the LSTM, given the complexity of the flow under study. Long-term prediction for LSTM show excellent agreement from the statistical point of view, with errors below 2% for the best models with respect to the reference. Furthermore, the analysis of the chaotic behaviour through the use of the Lyapunov exponents and of the dynamic behaviour through Poincaré maps emphasizes the ability of the LSTM to reproduce the temporal dynamics of turbulence. Alternative reduced-order

*Corresponding author
Email address: borrelli@kth.se (Giuseppe Borrelli)

models (ROMs), based on the identification of different turbulent structures, are explored and they continue to show a good potential in predicting the temporal dynamics of the minimal channel.

Keywords: Turbulent Flows, Deep-Learning, Minimal Channel Flow, Fourier POD (FFT-POD), Data-driven analysis, Long-short-term-memory (LSTM) networks

1. Introduction

Turbulence is undeniably one of the most fascinating and complex phenomena in nature, and it has attracted significant efforts of the scientific community in the past 150 years. The researches in this area have a great relevance both from the scientific perspective as well as for their applicability in numerous technological fields. Nowadays, it is still not possible to define a theoretical model able to characterize all the structures and mechanisms which come into play when turbulence occurs. The complex nature of this phenomenon has been traditionally studied either through experiments or employing numerical simulations. Recently, machine learning (ML) has offered a third option to enrich the knowledge we have about this subject, also thanks to the development of more powerful deep neural networks (DNNs) over the last years. Some examples include improved modelling results for Reynold-averaged Navier-Stokes (RANS) ([Vinuesa et al., 2020](#)) and large-eddy simulations (LESs), flow predictions ([Kutz, 2017](#); [Jiménez, 2018](#); [Duraisamy et al., 2019](#); [Brunton et al., 2020](#); [Jiang et al., 2021](#); [Guastoni et al., 2021](#)), flow control and optimization strategies ([Rabault et al., 2019](#); [Raibaudo et al., 2020](#); [Vinuesa et al., 2022](#)), generation of inflow conditions ([Fukami et al., 2019b](#)), extraction of flow patterns ([Raissi et al., 2020](#); [Eivazi et al., 2021a,b](#)), machine-learning-based reduced-order models ([Nakamura et al., 2021](#); [Vinuesa and Brunton, 2021](#)) and prediction of the temporal dynamics ([Srinivasan et al., 2019](#); [Eivazi et al., 2020](#)). The capability of a network to predict the temporal evolution of a turbulent flow is the focus of this study. One of the advantages of achieving this lies in the pos-

sibility of generating turbulent datasets employing a limited amount of initial conditions. Successful results have been obtained for a low-order nine-equation model (Moehlis et al., 2004), both for the statistics and the dynamical behaviour, employing models such as the multi-layer perceptron (MLP), long-short-term-memory (LSTM) network and Koopman with non-linear forcing operator (KNF) (Srinivasan et al., 2019; Eivazi et al., 2020).

When applying machine learning to fluid dynamics it is important to understand if ML methods are more convenient when compared to other classical approaches, *i.e.* if they are more accurate and efficient. Moreover, the knowledge we have about the flow has to be embedded in the models in order to improve the efficiency and the accuracy of the training and the predictions (Vinuesa and Brunton, 2021). Finally, we have to account for some possible ML alternatives which might be more appropriate for the solution of some problems, as shown in studies such as the one from Eivazi et al. (2020), where the KNF outperformed the deep-learning models (LSTM and MLP) in the prediction of the temporal dynamics of non-linear systems.

Our purpose is to employ a data-driven framework in order to have a model which is originated by the turbulence data, to overcome the limitations associated with other previous studies. Given this data-driven model, our aim is to assess the capability of some of the aforementioned architectures to predict the temporal evolution of the minimal channel. The possibility of performing predictions on a dataset derived from a minimal channel has already been investigated by Nakamura et al. (2021). In their case, the model reduction is achieved through a convolutional neural-network auto-encoder (CNN-AE) and the training and prediction of the temporal coefficients in the low-dimensional space is handled by the LSTM network. Our model is defined, instead, with an energy-based truncation derived from the outputs of a proper-orthogonal decomposition (POD), first introduced by Lumley (1967) and widely employed to achieve modal decomposition in the context of turbulent flows. We focus on a specific type of POD which is performed on the Fourier transform of the velocity fields (Webber et al., 1997), which is more convenient from the com-

putational standpoint, compared with the standard POD, usually performed in the physical domain.

The article is structured as follows: in §2 simulation and the data-driven model are introduced; in §3 the theory related to recurrent neural networks (RNNs) is presented and the results for the LSTM network are discussed; §4 gives an overview about the Koopman-based framework and highlights the limitations of this method applied to the data-driven case; in section §5 the possibility of implementing reduced-order models is explored; and, finally, a summary and a discussion of the results are provided in §6.

2. Computational framework

2.1. Numerical simulations

We carry out a direct numerical simulation (DNS) of the minimal channel following the parameters described in the work by Jiménez and Moin (1991). The channel flow is schematically represented in figure 1 and the box size is set to $x_l = 0.6\pi h$, $y_l = 2h$ and $z_l = 0.18\pi h$, where $h = 1$ denotes the half-height of the channel. The resolution is $32 \times 129 \times 16$, representing the number of grid points

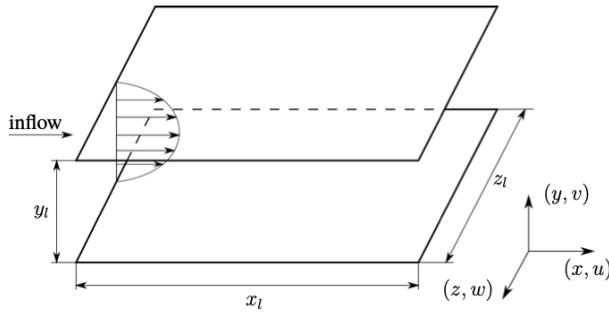


Figure 1: Schematic representation of a channel flow.

in the streamwise (x), wall-normal (y) and spanwise (z) directions, respectively. We introduce the reference velocity U_{cl} which corresponds to the centerline velocity of a laminar parabolic profile with the same volume flux as the one of the minimal channel flow. We will scale all the lengths and the velocities of the

paper with respect to h and U_{cl} , respectively. The laminar centerline-velocity based Reynolds number is set to $Re_{\text{cl}} = U_{\text{cl}}h/\nu = 5,000$, which leads to a friction Reynolds number $Re_\tau = hu_\tau/\nu = 202$ (where ν is the fluid kinematic viscosity and u_τ is the friction velocity). The minimal channel flow was simulated with the Fourier–Chebyshev numerical code SIMSON ([Chevalier et al., 2007](#)), a spectral solver which enables an efficient solution of the Navier–Stokes equations for some canonical cases. A simulation initialized with a superimposed random noise was performed from time $t = 0$ to $t = 10,000$, in order to have a fully-developed turbulent flow. Discarding this initial transient, the condition at $t = 10,000$ is employed as the initial one to generate the turbulent database composed by 800,000 snapshots on a time span of 160,000 time units employing a constant time step for the sampling of $\Delta t_s = 0.2$. A second-order Crank–Nicholson (CN) scheme is employed to model the linear terms of the Navier–Stokes equations, whilst the non-linear ones employ a third-order four-stages Runge–Kutta (RK3) scheme. The flow is driven by a streamwise pressure gradient, which is adjusted at each time step to maintain a constant flow rate.

2.2. Data-driven model

The data-driven model is constructed based on the outputs obtained after performing the proper-orthogonal decomposition (POD). An optimal low-order reconstruction of the flow from the energetic point of view is achievable by defining a truncation, since the contribution of the modes is rearranged with decreasing energy. The method is based on the Karhunen–Loéve theorem ([Berkooz et al., 1993](#)), which allows identifying a deterministic basis and a stochastic signal, given a number of turbulent velocity fields (snapshots). The stochastic part is described as:

$$\mathbf{u}(\mathbf{x}, t) = \sum_{j=1}^{N_p} a_j(t) \Phi_j(\mathbf{x}), \quad (1)$$

and defines the modal decomposition which separates the spatial modes $\Phi_j(\mathbf{x})$ from the associated temporal coefficients $a_j(t)$. Before performing the modal decomposition, the snapshots describing the evolution in time of the three-

dimensional velocity fields are rearranged into a snapshot matrix \mathbf{u}_{POD} as follows:

$$\mathbf{u}_{\text{POD}} = \begin{bmatrix} u_{x_1}^{t_1} & \dots & u_{x_1}^{t_{N_t}} \\ \vdots & \ddots & \vdots \\ u_{x_{N_p}}^{t_1} & \dots & u_{x_{N_p}}^{t_{N_t}} \end{bmatrix}, \quad (2)$$

where N_p is the number of grid points and N_t the total number of snapshots which are collected. In our study we focus on the streamwise component of the velocity only, *i.e.* the u -component. This is a reasonable simplification, due to the fact that this is the main velocity-fluctuation component in channels.

Having x and z as homogeneous directions in the minimal channel, the POD analysis can be performed directly on the Fourier transform of the field in those directions (Webber et al., 1997), *i.e.* we move from $\mathbf{u}(x, y, z, t)$ to $\hat{\mathbf{u}}(m, y, n, t)$, where (m, n) represents the wavenumber pair which, together with the y -dynamics (k), fully specifies the mode. Given a wavenumber pair, the modal decomposition is obtained by either solving the eigenvalue problem or employing the singular-value decomposition (SVD). In this work we employ the SVD approach, which decomposes the snapshot matrix as follows:

$$\mathbf{u}_{\text{POD}} = \mathbf{U}(\mathbf{x}) \cdot \mathbf{s} \cdot \mathbf{V}(t)^H, \quad (3)$$

where superscript H denotes the Hermitian transpose of the matrix, $\mathbf{U}(\mathbf{x})$ represents the spatial basis, $\mathbf{V}(t)$ the temporal modes and \mathbf{s} the singular values, *i.e.* the energy associated with each mode. The sizes of the resulting matrices are: $\mathbf{u}_{\text{POD}}(\mathbf{x}, t) \in \mathbb{C}^{(n_y \times N_t \times n_x n_z)}$, $\mathbf{U}(\mathbf{x}) \in \mathbb{C}^{(n_y \times e \times n_x n_z)}$, $\mathbf{s} \in \mathbb{R}^{(e \times e \times n_x n_z)}$ and $\mathbf{V}(t) \in \mathbb{C}^{(N_t \times e \times n_x n_z)}$, with $e = \min(N_t, n_y)$ being the size associated with the economy-SVD (and is equal to n_y for our case). Once the outputs are derived, the visualization of a single spatial mode, $\mathcal{U}_{(m,n),k} \in \mathbb{C}^{(n_x \times n_y \times n_z)}$, is obtained employing a frequency filter, $\mathcal{N} \in \mathbb{R}^{(n_x \times n_y \times n_z)}$, over the matrix \mathbf{U} (Webber et al., 1997). The filtering operation is performed for each quantum number k :

$$\mathcal{U}_{(m,n),k} = \mathbf{U}_k \otimes \mathcal{N}_{(m,n)}, \quad (4)$$

where the operation \otimes represents the element-wise product between matrices. To visualize the mode in the physical domain an inverse fast Fourier transform is applied on $\mathcal{U}_{(m,n),k}$. It is now possible to reconstruct the velocity field for a given truncation choice, *i.e.* up the the M -th most energetic mode:

$$\mathbf{u}(\mathbf{x}, t) = \sum_{i=0}^M \left(\operatorname{Re}\{a_i(t) \cdot \mathcal{U}_i\} - \operatorname{Im}\{a_i(t) \cdot \mathcal{U}_i\} \right), \quad (5)$$

having $a(t) = \mathbf{s}^{\mathbf{k}} \cdot \mathbf{V}^{\mathbf{k}}(t)^H$ as the temporal coefficient calculated after the FFT-POD method. The subscript i accounts for the mode represented by a triplet $\mathbf{k} = (m, n, k)$. The imaginary contribution of the mode is subtracted when reconstructing the field, otherwise a shift with respect to the original field is observed.

2.2.1. Energy-based truncation

We define our model by carrying out a truncation which employs the first 100 most energetic modes. These modes capture 78.8% of the total energy related to the streamwise fluctuations and generate an error on the reconstruction of 22.2%, where this error is defined as:

$$\frac{\|u_{\text{orig}} - u_{\text{recon}}\|_2}{\|u_{\text{orig}}\|_2}, \quad (6)$$

where u_{orig} are the streamwise fluctuations of the velocity reconstructed with all the modes; and u_{recon} are the ones reconstructed with a reduced amount of modes. An example of how the velocity fields are reconstructed is shown in figure 2. The plots show the two-dimensional comparison for the xy - and xz -planes at $z = 0$ and $y^+ \approx 24$, respectively, where the distance from the wall is given in wall units $y^+ = y \cdot u_\tau / \nu$. Even if a difference due to the truncation is observed, the obtained model is representative of the turbulence in the minimal channel, *i.e.* it reconstructs a significant percentage of the total energy and it includes the most relevant turbulent features present in channel flows. These structures are recognizable in the modes which define the spatial basis of our

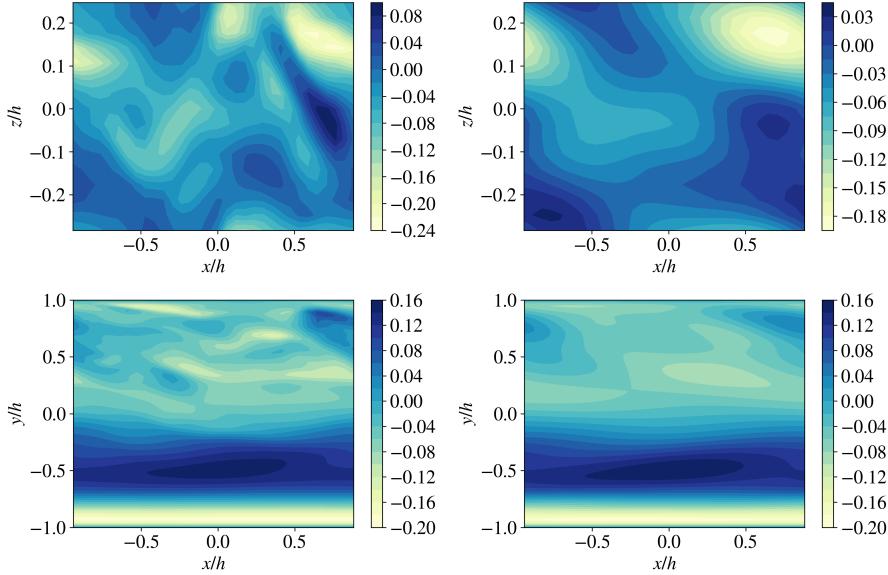


Figure 2: Visualization of the streamwise fluctuations of the velocity (*left*) compared to the reconstruction which employs the 100 most energetic modes (*right*). The xz -plane is shown above ($y^+ \approx 24$) and the xy -plane is shown below ($z = 0$). The example reports the reconstruction at $t = 10,000$ for a dataset composed by 10,000 snapshots, where the initial transient has been discarded.

model. Figure 3 shows the roll modes ($m = 0$), which are usually associated with the higher energy content and represent counter-rotating vortex which move the flow from the wall to the outer region and vice-versa; the streamwise modes ($n = 0$) show a dependence only in the x -direction; and the propagating modes, described as tilted vortex with respect to the wall and with development in the three directions (Webber et al., 1997). Table 1 reports the energy percentage associated with the first modes.

(m, n, k)	(0,1,1)	(0,1,2)	(0,0,1)	(0,0,2)	(1,1,1)	(0,1,3)	(0,1,4)
energy %	12.69	11.67	9.30	5.89	3.82	3.18	3.00

Table 1: Energy percentage associated with the first 7 modes present in the minimal channel.

The procedure through which the fields are simulated accounts separately for the wavenumbers with opposite sign in the homogeneous directions. These exhibit the same energy content and for this reason the concept of degeneracy

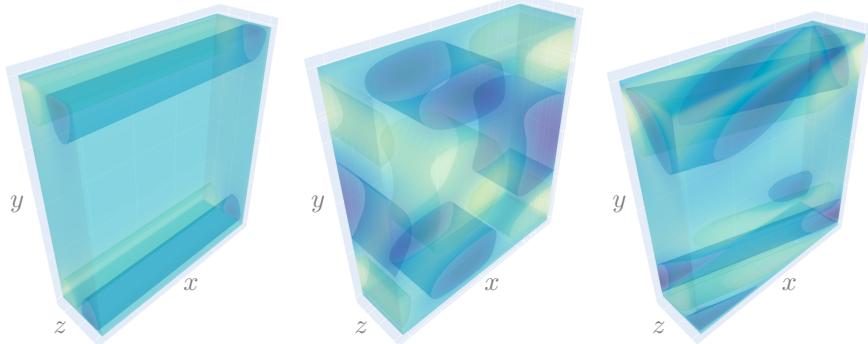


Figure 3: Visualization of roll mode (*left*), streamwise mode (*center*) and propagating mode (*right*).

is applied, *i.e.* these modes are considered as a single one and their energy is summed up together.

One final aspect related to the minimal channel arises from figure 2, *i.e.* the fact that, alternatively, the flow can be turbulent at one wall and (nearly) laminar at the other, leading to a different mean velocity profile in time. This issue was addressed by Jiménez and Moin (1991), who recognized a slow relaxation time for the statistics. For this reason, the comparison between true and predicted statistics will be justified only when employing the same time horizon and when starting from the same time instant.

2.2.2. Dataset analysis

After fixing the spatial basis, we focus on the analysis of the temporal modes. These modes define the dataset employed for the training and the ones that are predicted. We introduce an additional simplification by separating the fluctuations of the temporal coefficients related to the first wavenumber pair $a_{(0,0)}(t)$ and the ones associated with the other wavenumbers $a'(t)$. Doing so, we exclude the net-flux modes, which vary only along the wall-normal direction, and we are able to represent 97% of the total energy associated with $a'(t)$. Moreover, a scaling of the dataset is necessary in order to have a correct training of the network, otherwise the gradients which enter the computations of the cost function will be too small and it will be impossible to generate significant updates of the

parameters of the network. The evolution of each scaled temporal mode $\hat{a}_j(t)$ is given by:

$$\hat{a}_j(t) = \frac{a_j(t) - \langle a_j(t) \rangle}{\sigma[a_j(t)]}, \quad (7)$$

where the scaling is performed by subtracting the mean, denoted by $\langle \cdot \rangle$, and dividing everything by the variance, σ , of the signal.

A difference in the frequency content of the temporal modes of the minimal channel is observed, as highlighted in figure 4. This feature was not reported in

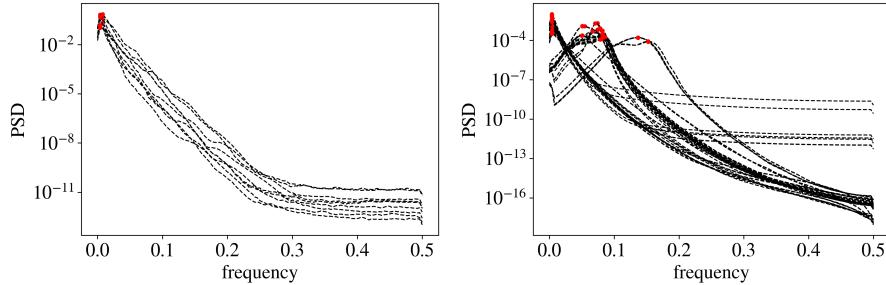


Figure 4: Comparison of the frequency content between the model by [Moehlis et al. \(2004\)](#) (*left*) and the minimal channel (*right*) studied here. Peaks are highlighted.

the studies predicting the nine-equation model ([Srinivasan et al., 2019](#); [Eivazi et al., 2020](#)), where all the modes have similar frequency spectra. A multi-step concept is applied to correctly capture the dynamics for each mode and is achieved by adjusting the sampling rate, employing a larger Δt for the signals with a lower frequency and a smaller one as the high-frequency content becomes more prominent. The main idea of this approach is to have different networks which are responsible for the prediction of a reduced set of modes.

The possibility of applying the multi-step model relies on a correlation analysis performed on the signals of the groups which are identified. If the correlation between signals of different groups is low compared with that of signals from the same group then it is reasonable to use this subdivision. In our case we have low- and high-frequency signals, denoted by [1] and [2] respectively. The correlation factor, defined as $\sigma[\mathcal{C}_{(i,j)}] - \langle \mathcal{C}_{(i,j)} \rangle$ (where $\mathcal{C}_{(i,j)}$ is the correlation vector between the signals related to modes i and j), belongs to the inter-

val $[3 \times 10^{-4}, 2 \times 10^{-3}]$ for signals of the different groups, while it belongs to $[4 \times 10^{-3}, 3 \times 10^{-1}]$ for signals of the same group, leading to the conclusion that it is safe to train the networks separately. We employ a sampling rate $\Delta t_{[1]} = 0.8$ for the low-frequency signals and $\Delta t_{[2]} = 0.4$ for the high-frequency ones.

3. Predictions with long-short-term-memory (LSTM) network

The success of recurrent neural networks (RNNs) to achieve the training on sequential data has been demonstrated with respect to the much simpler architecture of the multi-layer perceptron (MLP) which, instead, is limited to a point-to-point prediction, without taking into account the temporal dependences of the input. The *input* \rightarrow *hidden* \rightarrow *output* layers structure is preserved but, with respect to MLPs, RNNs introduce a feedback loop in the hidden layer (or layers), allowing to combine the information about the internal state of each neuron (also known as memory) with the input vector, to generate the output at a given time. Thanks to the feedback mechanism, this output becomes the input of the same neuron at the next time step and, eventually, it allows the network to learn the dynamics of the system. The training of the network gets more difficult as the input sequence spans a longer range in time and leads to the problem of vanishing gradients. This arises when, for each iteration, the corrections to the weights associated to the back propagation algorithm reduce the gradient until the network is not capable of learning any further.

Introduced by Hochreiter and Schmidhuber (1997), long-short-term-memory (LSTM) networks are ideal for long-term dependencies, such as the ones encountered in turbulent flows, thanks to the ability to control the dynamics of the recurrent connections in time by means of gating mechanisms. Algorithm 1 shows the procedure to compute the output of the cell for each time step. The role of each gate is as follows: the forget gate employs the current input (χ_t) and the output of the previous time instant (ζ_{t-1}) to define the fraction of the cell state (\mathbf{C}_{t-1}), which is kept in the evaluation of the current cell (\mathbf{C}_t); the input gate determines the values of the cell state to be updated (using the same

Algorithm 1: General scheme of the LSTM algorithm.

Input: Sequence $\chi_1, \chi_2, \dots, \chi_p$
Output: Sequence $\zeta_1, \zeta_2, \dots, \zeta_p$

set $\mathbf{h}_0 \leftarrow 0$
set $\mathbf{C}_0 \leftarrow 0$

for $t \leftarrow 1$ **to** p **do**

$\mathbf{f}_t \leftarrow \sigma(\mathbf{W}_f[\chi_t, \zeta_{t-1}] + \mathbf{b}_f)$
$\mathbf{i}_t \leftarrow \sigma(\mathbf{W}_i[\chi_t, \zeta_{t-1}] + \mathbf{b}_i)$
$\tilde{\mathbf{C}}_t \leftarrow \tanh(\mathbf{W}_f[\chi_t, \zeta_{t-1}] + \mathbf{b}_f)$
$\mathbf{C}_t \leftarrow \mathbf{f}_t \otimes \mathbf{C}_{t-1} + \mathbf{i}_t \otimes \tilde{\mathbf{C}}_t$
$\mathbf{o}_t \leftarrow \sigma(\mathbf{W}_o[\chi_t, \zeta_{t-1}] + \mathbf{b}_o)$
$\zeta_t \leftarrow \mathbf{o}_t \otimes \tanh(\mathbf{C}_{t-1})$

quantities as the ones of the forget gate) and it computes candidates ($\tilde{\mathbf{C}}_t$) for the update of the cell state at the present time instant; and the output gate computes the output values (ζ_i) applying the updated cell state \mathbf{C}_t .

The LSTM network is trained over 2,000 epochs with focus on the real part of the temporal coefficients only, having the imaginary description with a very similar dynamics. We use the hyperbolic tangent as the activation function for the internal layer, to account for the non-linearity of the problem, while the dense layer (*i.e.* the one that stores the outputs) employs a linear activation function to consider the correct range of the values, due to the fact that once the dataset is scaled we are not limited to the interval $[-1, 1]$ anymore (which are the limit values of the tanh function). We use a mean-squared error as a loss function and the Adam optimizer to control the evolution of the learning rate ([Kingma and Ba, 2015](#)). Albeit the Adam algorithm itself employs a correction of the learning rate during the training, an exponential decay is introduced to have a further reduction of the losses:

$$\text{LR} = \text{LR}_0 \cdot \alpha_D^{n_C/n_D} = 0.001 \cdot 0.96^{n_C/n_D}, \quad (8)$$

where LR_0 is the initial learning rate, α_D is the decay rate and n_D is the decay step to be considered with respect to the current step n_C . Finally, the dimension of the batch size is constant and equal to 32. This means that every 32

samples an update of the weights is made. The dataset is divided into training and validation sets: 80% of the data is employed for training and 20% for the validation. The model which gives the best loss is stored as the training advances.

3.1. Temporal predictions and statistics

The idea behind the prediction is to select an initial sequence of length p and predict the value at $p + 1$. Moving one step forward with the predictions, the $p + 2$ value is then predicted, still using the p values before, where, this time, one of the value of the sequence is predicted. This explains how, as we advance in time, the error related to the predictions accumulates, leading to a point-by-point prediction which is not exact. Validation losses give an indication of how accurate the predicted values are with respect to the true ones.

The instantaneous predictions for the best model are reported in figure 5 for the short-term horizon and in figure 6 for a longer time span. These are

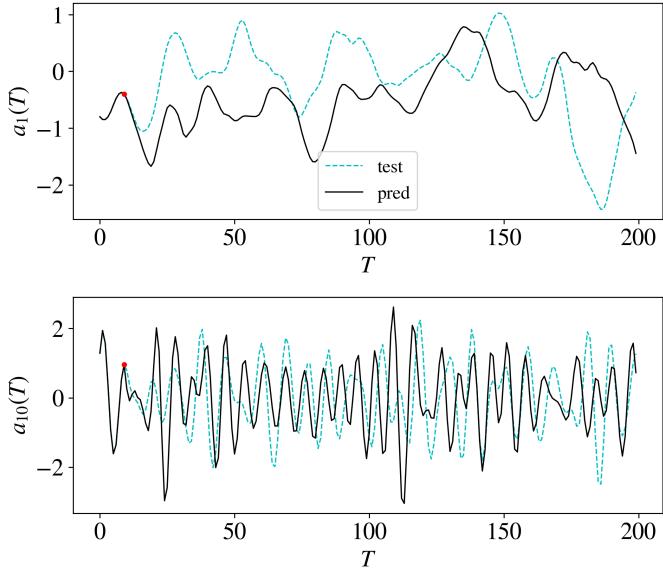


Figure 5: Short-term predictions for mode 1 (*top*) and mode 10 (*below*). These two modes are representative of low-frequency and high-frequency signals, respectively. The red dot represents the point where predictions start. Note that indexing starts with mode 0, so mode 1 is the second most energetic mode.

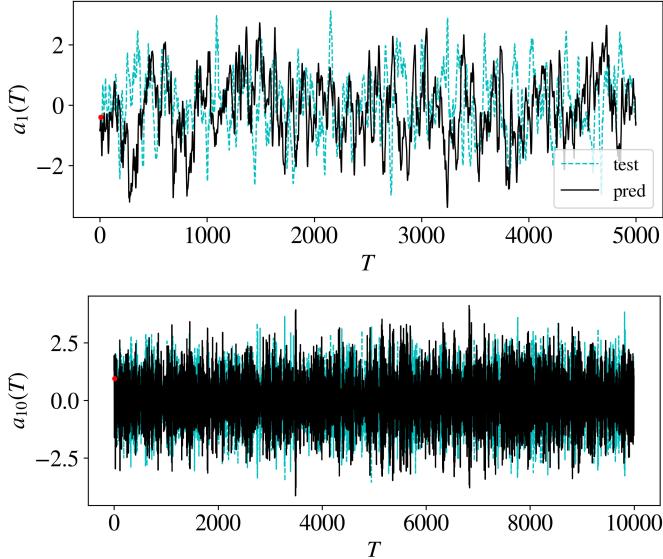


Figure 6: Long-term predictions for mode 1 (*top*) and mode 10 (*below*).

obtained with the LSTM-1-200-10-100 model, *i.e.* a network which employs a single layer with 200 neurons, an initial vector of $p = 10$ values and it has been trained with a dataset of 100,000 snapshots. We observe a temporal evolution which is not exactly the same as it was for the predictions based on the nine-equations model. On the other hand, the networks are still able to reproduce the frequency content of the signals and the coefficients lie in the correct range, providing a solution which is acceptable from the physical standpoint. The validation losses are of the order of 10^{-6} and 10^{-3} for the signals of group [1] and [2], respectively. The effect of each hyper-parameter on the statistics and on the validation is described in table 2. We notice that the best models are the ones which either have a higher capacity, *i.e.* more neurons are employed, or have a shorter initial vector of known coefficients. We also observe how for some cases a non-physical behaviour takes place, for instance if the number of the snapshots of the training dataset is not sufficient or when the number of cells in the hidden layer are not enough. This behaviour manifests itself by showing an intensification of the oscillation after a certain time horizon. Note that the

first attempt was performed by using 90 neurons, which successfully predicted the temporal dynamics of the modes of the nine-equation model, but this model was not able to reproduce a plausible behaviour for the minimal channel. Once the temporal evolution of the modes is predicted we can use this information together with the spatial basis in order to reconstruct the velocity fields using equation (5) with $M = 100$, after having scaled the temporal data back to the original range.

LSTM-<i>ly</i>-cells-<i>p</i>-<i>N</i>	$E_{u_{\text{RMS}}} [\%]$	val. loss 1	val. loss 2
<i>LSTM-1-200-10-10</i>	10.64	1.08×10^{-5}	1.13×10^{-2}
LSTM-1-200-10-50	3.57	1.09×10^{-5}	7.62×10^{-3}
LSTM-1-200-10-100	2.97	7.74×10^{-6}	2.03×10^{-3}
LSTM-1-200-10-200	6.08	3.97×10^{-5}	2.29×10^{-3}
<i>LSTM-1-90-10-50</i>	16.37	1.49×10^{-4}	6.05×10^{-3}
<i>LSTM-1-150-10-50</i>	3.52	2.26×10^{-5}	6.39×10^{-3}
LSTM-1-300-10-50	5.67	5.01×10^{-6}	7.35×10^{-3}
<i>LSTM-1-200-5-50</i>	3.28	9.51×10^{-6}	3.35×10^{-3}
LSTM-1-200-20-50	7.85	2.00×10^{-5}	1.10×10^{-2}
LSTM-1-200-40-50	6.86	1.63×10^{-5}	1.37×10^{-2}
<i>LSTM-2-200-10-50</i>	14.32	6.47×10^{-6}	9.30×10^{-3}

Table 2: Hyper-parameter effect on statistics, where the various architectures are labelled based on the number of layers (ly), number of cells ($cells$), initial sequence length (p) and training-dataset dimension (N), expressed in thousands of snapshots. The various model are grouped according to the hyper-parameter on which the analysis focuses, in order: N , $cells$, p and ly . All the modes are employed for training, accounting for the degeneracy. The validation losses associated with the low- (1) and high-frequencies (2) are also reported. The results in boldface represent the overall best performance for the error over u_{RMS} or for the validation losses. A non-physical behaviour arises in the LSTM architectures highlighted in italics. The underlined models show results averaged from three different training. Statistics obtained over 4,000 time units.

When dealing with turbulent flows the analysis of the statistical behaviour is a relevant tool to assess the correctness of our predictions. The statistics are derived from the reconstructed velocity fields for a time horizon of 4,000 time units with a time step of $\Delta t_{\text{recon}} = 0.8$. We then compare the predicted root-mean-square streamwise velocity-profile, u_{RMS} , with the one of the original simulation. We define the relative error $E_{u_{\text{RMS}}}$ for the streamwise fluctuations and report this quantities (together with the validation losses) in table 2 for all the architectures under study. The smallest errors for the statistics are obtained

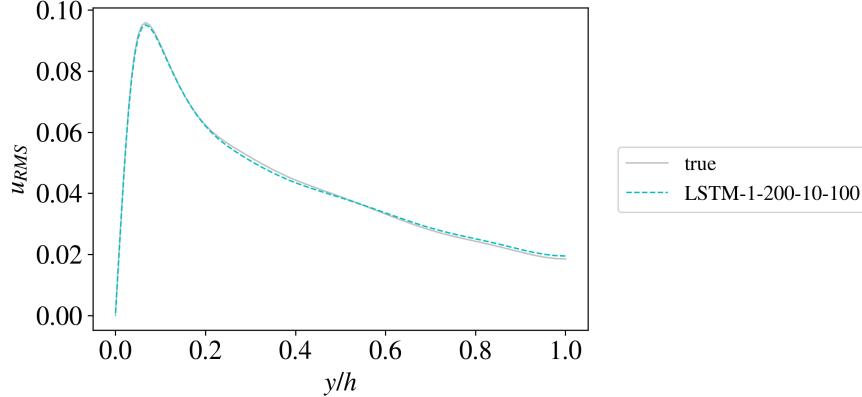


Figure 7: Predicted profiles of the streamwise fluctuations (RMS) for the best model (LSTM-1-200-10-100), compared with the reference. The profiles are averaged with respect to the centerline of the channel ($y = h$).

for the LSTM-1-200-10-100 architecture. The best model associated with this network generates an error of the fluctuations in the streamwise direction of $E_{u_{\text{RMS}}} = 1.3\%$ (note that the results in the table are averaged over 3 networks). The profiles of the true and predicted statistics are illustrated in figure 7. The best architecture for each hyper-parameter analysis is underlined in table 2 and for these cases three different LSTM networks have been trained with the same setup, for both frequency groups, in order to enforce the consistency of the results (average of the errors is reported), due to the stochastic nature of the training process. Table 3 includes the variance associated with the errors on the streamwise velocity fluctuations for these models.

model	LSTM-1-200-10-100	LSTM-1-150-10-50	LSTM-1-200-5-50
$\sigma[E_{u_{\text{RMS}}}]$	1.17	0.43	0.78

Table 3: Variance of the errors calculated over the streamwise velocity fluctuations for the best models related to each hyper-parameter analysis, *i.e.* the underlined architectures in table 2.

3.2. Analysis of the chaotic and dynamic behaviour

One more useful investigation to assess that the predictions lead to reasonable results and to an appropriate reproduction of the physics in the minimal

channel is to study the chaotic behaviour by means of the Lyapunov exponent (λ). This method allows to verify the sensitivity of a chaotic system to the initial conditions. Given two trajectories we introduce an infinitesimal perturbation $\delta\mathbf{A}_0$ and we observe how the difference between the original and the perturbed trajectories $\delta\mathbf{A}$ evolves in time. We assume the initial divergence to be exponential and to be approximated by:

$$|\delta\mathbf{A}(t)| \approx e^{\lambda t} |\delta\mathbf{A}_0|, \quad (9)$$

where the time evolution of the divergence $|\delta\mathbf{A}(t)|$ is given by:

$$|\delta\mathbf{A}(t)| = \left[\sum_{i=1}^n (a_{i,o}(t) - a_{i,p}(t))^2 \right]^{1/2}, \quad (10)$$

where the subscript p represents the perturbed evolution and o the original one. The contribution of all the n modes is summed up together.

A new dataset of velocity fields is generated after having introduced an initial perturbation of the order of 10^{-3} at $t_0 = 10,000$ as a mean energy density of the noise, which translates to a disturbance of $|\delta\mathbf{A}_0| \approx 10^{-7}$ onto the temporal coefficients. The temporal coefficients are derived by projecting the fields on the spatial basis identified in section 2. The analysis is performed separately

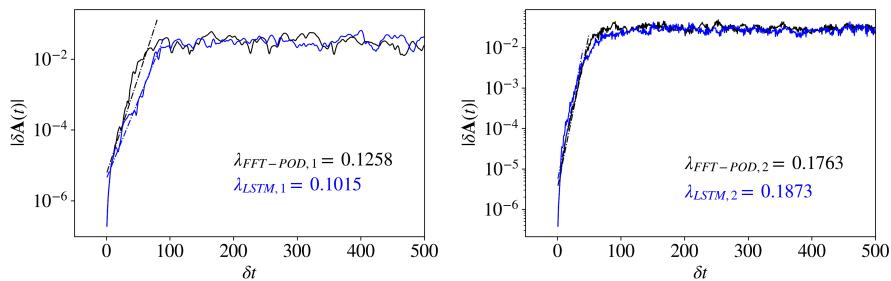


Figure 8: Comparison of the divergence in time between trajectories for the data-driven model (black) and for the predicted data from the LSTM-1-200-10-100 architecture (blue) once an initial perturbation of $|\delta\mathbf{A}_0| \approx 10^{-7}$ is introduced at $t_0 = 10,000$. Example of Lyapunov exponents for the real part of the low-frequency signals (left) and imaginary part of the high-frequency signals (right).

for the real and imaginary parts of the coefficients as well, as for the low-

frequency signals and the high-frequency ones. The predicted perturbed dataset is obtained by employing the LSTM-1-200-10-100 network. Figure 8 shows an example of the evolution in time of the divergence between the trajectories, where the calculated Lyapunov exponents are shown for each case. Generally,

	real		imaginary	
	λ_1	λ_2	λ_1	λ_2
Reference	0.1258	0.1387	0.1167	0.1763
LSTM	0.1015	0.0852	0.1104	0.1873

Table 4: Numerical comparisons of the Lyapunov exponents for the reference model and the LSTM predictions. Note that 1 and 2 denote the low- and the high-frequency models, respectively.

having $\lambda > 0$ is an indication that the system is chaotic and, furthermore, from table 4 we can also conclude that the LSTM networks is able to accurately reproduce the physics of the minimal channel, having a really good agreement between the reference and the predicted Lyapunov exponents. We can also observe from figure 8 how the saturation of the curves, *i.e.* the point where the exponential divergence ends, is found approximately after the same time interval.

Moreover, we want to verify the quality of the predicted dynamic behaviour through the use of the Poincaré maps. These maps represent the intersection of two temporal coefficients a_α and a_β with the hyper-plane $a_\gamma = 0$, *i.e.* where the coefficient a_γ changes its sign ($da_\gamma/dt < 0$). Again, this study is performed separately for the real and imaginary part and employs the multi-step concept. The most energetic temporal modes are analysed and reported in figure 9: intersection of $a_1 - a_6$ with the plane $a_3 = 0$ for group [1] and $a_{10} - a_{14}$ with $a_{12} = 0$ for group [2]. The picture shows a good agreement, leading to the conclusion that the LSTM network is capable of reproducing the correlation between the amplitudes of the modes which are considered. The same good agreement is observed for the intersection of the other temporal coefficients, associated with a lower energy content, with the hyper-planes identified by the other modes belonging to the data-driven model. Analyzing the results obtained in Eivazi

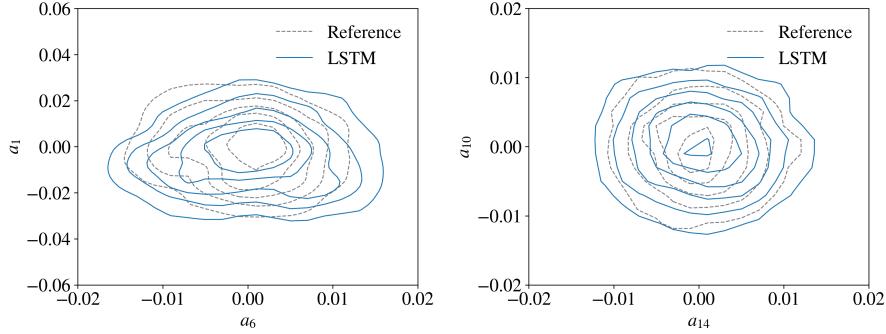


Figure 9: Example of comparison of the Poincaré maps for the most-energetic temporal coefficients belonging to group [1] (*left* - real coefficients: a_1, a_6 , intersection with plane $a_4 = 0$) and to group [2] (*right* - imaginary coefficients: a_{10}, a_{14} , intersection with plane $a_{12} = 0$). The reference maps are reported in grey and the LSTM-predicted ones in blue.

et al. (2020) and Srinivasan et al. (2019), it can be argued that in the data-driven case the reproduction is not as accurate as for the nine-equations model, but this further supports the idea of having a more complex problem to study when dealing with turbulence generated by the minimal channel.

4. Predictions with Koopman-based framework

An alternative to predict the temporal evolution for a high-dimensional system which is characterized by a non-linear dynamics is to exploit the Koopman-operator theory. The idea is to employ a linear operator on an infinite-dimensional space to describe a non-linear behaviour in a finite domain.

Dynamic-mode decomposition (DMD) offers a first example through which we are able to describe a non-linear problem by using linear observables, i.e. data which is rearranged in a vector-valued snapshot sequence to describe the state-vector of the system (Schmid, 2010; Tu et al., 2014). In this approach, the available linear functions might not be enough to describe the non-linear behaviour of the flow. A solution to this issue is offered by the extended version of the DMD (EDMD) routine (Williams et al., 2015). Still a limiting factor arises, *i.e.* the necessity of having a previous knowledge of the dynamics of the system in order to define the appropriate dictionary of linear functions.

Data-driven approaches have been included in the procedure and are able to provide a rich feature space by means of time-delay embedding instead of having to define linear/non-linear observable functions (Li et al., 2017; Lusch et al., 2018; Takeishi et al., 2017). Following this idea, delay embedding has been integrated with DMD in the Hankel-DMD method (HDMD) to study chaotic systems (Arbabi and Mezic', 2017). In the study from Brunton et al. (2017) this model was extended by proposing the Hankel alternative view of Koopman (HAVOK), which incorporates Koopman theory, time-delay embedding and sparse regression. Moreover, Khodkar et al. (2019) presented a new Koopman-based framework in which nonlinearities are modelled through an external forcing and it is capable of successfully predicting the dynamics of highly-chaotic systems.

4.1. Koopman with non-linear forcing (KNF)

The preference of the Koopman framework which employs a non-linear forcing (KNF) over the HDMD is driven by the better accuracy achieved by the first method in the predictions of a chaotic dynamical system (Khodkar et al., 2019). We consider a dynamical system:

$$\mathbf{x}_{t+1} = \mathbf{F}(\mathbf{x}_t), \quad (11)$$

on the state space $\mathcal{M} \subseteq \mathbb{R}^{m'}$, where \mathbf{x} is a spatial coordinate of the state, and $\mathbf{F} : \mathcal{M} \rightarrow \mathcal{M}$ is the operator describing the evolution of the system. The Koopman operator \mathcal{K} acts on functions of state space (called *observables*) $g : \mathcal{M} \rightarrow \mathbb{C}$ as follows:

$$\mathcal{K}g = g \circ \mathbf{F}, \quad (12)$$

where \circ denotes the composition of g with \mathbf{F} . The Koopman operator defines a new dynamic of the system which governs the evolution of the *observables* $g_t = g(\mathbf{x}_t)$ in discrete time and which is linear and infinite-dimensional. We then move to the Koopman-based framework:

$$\mathbf{x}^{t+1} = \mathbf{A}\mathbf{x}^t + \mathbf{B}\mathbf{f}^t, \quad (13)$$

where the external forcing \mathbf{f} models the non-linearities and \mathbf{A} and \mathbf{B} denote the unknown matrices which describe the dynamical system. By knowing these matrices we are able to advance the solution of the state in time. The vector \mathbf{f} includes any candidate nonlinear functions of \mathbf{x} , described by polynomials:

$$\mathbf{f}^i = \begin{bmatrix} 1 & (\mathbf{x}^i)^{p_2} & (\mathbf{x}^i)^{p_3} & \cdots & (\mathbf{x}^i)^{p_n} \end{bmatrix}^T, \quad (14)$$

where T denotes the transpose of the matrix. Here, for instance, $(\mathbf{x}^i)^{p_2}$ and $(\mathbf{x}^i)^{p_3}$ indicate any possible quadratic or cubic non-linearities, respectively ($a_j^i a_k^i$ and $a_j^i a_k^i a_l^i$, where j, k, l denote the modes which are considered). Constant or sinusoidal functions can also be considered to build this vector. Usually, an intuition of the governing equations of our problem is necessary in order to define the forcing term but, on the other hand, the sparse identification of non-linear dynamics (SINDy) method (Brunton et al., 2016b) offers an alternative to achieve this in an optimized way, thanks to the sparsity of the matrix which is considered. We first have an iterative linear regression of $\begin{bmatrix} \mathbf{x}^2 & \mathbf{x}^3 & \cdots & \mathbf{x}^N \end{bmatrix}$ on $\begin{bmatrix} \mathbf{x}\mathbf{f}^1 & \mathbf{x}\mathbf{f}^2 & \cdots & \mathbf{x}\mathbf{f}^{N-1} \end{bmatrix}$, where $\mathbf{x}\mathbf{f}^i = \begin{bmatrix} \mathbf{x}^i & \mathbf{f}^i \end{bmatrix}^T$, and then zero out all the coefficients which are smaller than a threshold value ε . This procedure is performed in an iterative way until a convergence is observed for the non-zero coefficients. Algorithm 2 shows a schematic of the SINDy method, which has an important advantage over deep-learning methods due to its interpretability. Note that this issues of interpretability of deep-learning models are discussed by Vinuesa and Sirmacek (2021).

At this point we employ the Hankel-matrix representation of the data to redistribute the values in a matrix form, so that we express the state vector and the forcing term as \mathbf{X} and \mathcal{F} , respectively:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^1 & \dots & \mathbf{x}^{N'-q} \\ \vdots & \ddots & \vdots \\ \mathbf{x}^q & \dots & \mathbf{x}^{N'-1} \end{bmatrix}, \quad \mathcal{F} = \begin{bmatrix} \mathbf{f}^1 & \dots & \mathbf{f}^{N'-q} \\ \vdots & \ddots & \vdots \\ \mathbf{f}^q & \dots & \mathbf{f}^{N'-1} \end{bmatrix}, \quad (15)$$

Algorithm 2: SINDy algorithm to compute non-linear terms for the KNF method.

Input: $\mathbf{y} = [\mathbf{x}^2 \ \mathbf{x}^3 \ \dots \ \mathbf{x}^N]$, $\mathbf{x} = [\mathbf{x}\mathbf{f}^1 \ \mathbf{x}\mathbf{f}^2 \ \dots \ \mathbf{x}\mathbf{f}^{N-1}]$,
threshold ε

Output: $\mathbf{I}_{\text{active}}$ ▷ Indices of active nonlinearities

$n, m \leftarrow$ number of rows of \mathbf{y}, \mathbf{x}

Initialize $\mathbf{C}(n, m)$ ▷ Coefficients

Initialize $\mathbf{I}(n, m)$ ▷ Active indices, dtype = bool

for $i \leftarrow 1$ **to** Max Iteration = 20 **do**

for $j \leftarrow 1$ **to** n **do**

$\mathbf{I}_j \leftarrow \mathbf{I}[j, :]$

$\mathbf{C}[j, \mathbf{I}_j] \leftarrow$ Ridge Regression($\mathbf{y}[j, :]$, $\mathbf{x}[\mathbf{I}_j, :]$)

$\mathbf{I}_j \leftarrow \text{abs}(\mathbf{C}[j, :]) >= \varepsilon$ ▷ Find big coefficients

$\mathbf{C}[j, \sim \mathbf{I}_j] \leftarrow 0$ ▷ Zero out small coefficients

$\mathbf{I}[j, :] \leftarrow \mathbf{I}_j$

if \mathbf{C} does not change **then**

Break

$\mathbf{I}_{\text{active}} \leftarrow$ Maximum element of each column in $\mathbf{I}[:, n:]$

and equation (13) is expressed in its time-embedded form as:

$$\mathbf{X}^{t+1} = \mathbf{A}\mathbf{X}^t + \mathbf{B}\mathcal{F}^t. \quad (16)$$

The matrices \mathbf{X} and \mathcal{F} have sizes $(m' \times q) \times (N' - q + 1)$ and $(n' \times q) \times (N' - q)$ respectively, where m' is the number of state variables, n' is the size of the forcing vector (usually $n' \gg m'$), N' is the number of vector-valued observables and q is the delay-embedding dimension. We rearrange the data in the Hankel matrices following the exact-DMD algorithm formulation (Arbabi and Mezic', 2017), and we define \mathbf{X}' and \mathbf{Y}' as:

$$\mathbf{X}' = [\mathbf{x}^1 \dots \mathbf{x}^{N'-q}], \quad \mathbf{Y}' = [\mathbf{x}^2 \dots \mathbf{x}^{N'-q+1}], \quad (17)$$

where \mathbf{x}^i denotes the i^{th} column of the Hankel matrix. Ultimately, the matrices \mathbf{A} and \mathbf{B} are derived by using the DMDc algorithm (c stands for control) introduced by Proctor et al. (2016) which relies on the minimization of the Frobenius

norm $\|\mathbf{Y}' - \mathbf{A}\mathbf{X}' - \mathbf{B}\mathcal{F}\|_F$. These are derived as:

$$\mathbf{A} = \hat{\mathbf{U}}^H \mathbf{Y}' \tilde{\mathbf{V}} \tilde{\mathbf{S}}^{-1} \tilde{\mathbf{U}}_1^H \hat{\mathbf{U}}, \quad \mathbf{B} = \hat{\mathbf{U}}^H \mathbf{Y}' \tilde{\mathbf{V}} \tilde{\mathbf{S}}^{-1} \tilde{\mathbf{U}}_2^H. \quad (18)$$

The energy truncation of the matrices derived from the SVD, on \mathbf{Y}' and $[\mathbf{X}' \quad \mathcal{F}]$, is denoted by $(\hat{\cdot})$ and $(\tilde{\cdot})$, respectively. The truncation rank of \mathbf{Y}' is r and it leads to the decomposition $\mathbf{Y}' = \hat{\mathbf{U}} \hat{\mathbf{S}} \hat{\mathbf{V}}^H$, with $\hat{\mathbf{U}} \in \mathbb{R}^{(m' \times q) \times r}$, $\hat{\mathbf{S}} \in \mathbb{R}^{r \times r}$ and $\hat{\mathbf{V}} \in \mathbb{R}^{(N' - q) \times r}$. In the same way $[\mathbf{X}' \quad \mathcal{F}] = \tilde{\mathbf{U}} \tilde{\mathbf{S}} \tilde{\mathbf{V}}^H$. The truncation rank is defined by d for this case, with $\tilde{\mathbf{U}} \in \mathbb{R}^{((m' + n') \times q) \times d}$, $\tilde{\mathbf{S}} \in \mathbb{R}^{d \times d}$ and $\tilde{\mathbf{V}} \in \mathbb{R}^{(N' - q) \times d}$. The choice of the truncation defines a threshold below which the contribution of the modes is considered as negligible from the energetic perspective and is based on SVD rank-truncation methods such as the optimal hard threshold illustrated by [Gavish and Donoho \(2014\)](#). Finally, $\tilde{\mathbf{U}}$ is divided into $\tilde{\mathbf{U}} = [\tilde{\mathbf{U}}_1^H \quad \tilde{\mathbf{U}}_2^H]$, where $\tilde{\mathbf{U}}_1 \in \mathbb{R}^{(m' \times q) \times d}$ enters the computation for \mathbf{A} and $\tilde{\mathbf{U}}_2 \in \mathbb{R}^{(n' \times q) \times d}$ defines \mathbf{B} , which is related to the forcing.

4.2. KNF temporal predictions

As for the LSTM case, the multi-step concept has been applied. This time the training consists of finding the matrices \mathbf{A} and \mathbf{B} , underlining the non-stochastic nature of this process, that lead to have always the same model for the same setup of parameters on which predictions are based (LSTM is deterministic only once it is trained). For all the tests performed with different configurations the method is not capable of capturing the physics of the system, as discussed below.

By assessing the predictions related to the low-frequency group in figure 10 we observe how the predicted fluctuations associated with the temporal modes are not sustained. For this case a delay-embedding dimension of $q = 5$ was chosen. The dataset comprises over 10,000 snapshots of the 5 most energetic modes of group [1] with a sampling time $\Delta t_{[1]} = 0.8$. Non-linearities are modelled with a third-order polynomial without employing sparsity promotion. The energy-based tolerances for matrices \mathbf{A} and \mathbf{B} are $e_r = 10^{-5}$ and $e_p = 10^{-5}$.

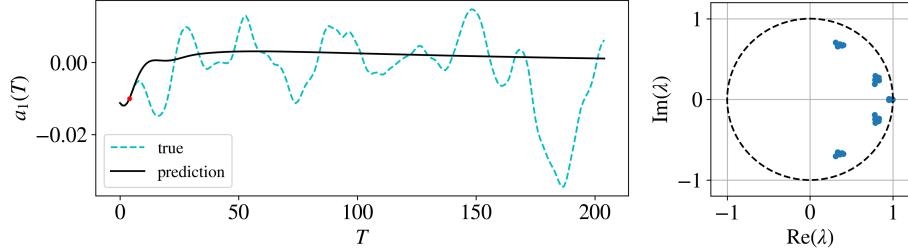


Figure 10: Short-term prediction of the a_1 coefficient with $q = 5$ on the left. The unitary circle representing the eigenvalues of matrix \mathbf{A} is shown on the right.

The unitary circle where the eigenvalues of matrix \mathbf{A} are plotted might be considered as an indicator of why the predicted evolution is damped in time, as most of the eigenvalues are inside this circle ($|\lambda| < 1$), thus a decay is expected. Different tests have been performed by adjusting the tolerances e_r and e_p or by either gradually increasing the dataset dimension or employing more modes, without showing relevant improvements. The modeling of the forcing term has been attempted with different polynomial orders and also by using trigonometric functions. Note that sparsity promotion did not produce noteworthy changes either. Finally, as q increases the fluctuations are sustained for a longer time horizon, but eventually they are flattened out too. One option would be to further increase the delay-embedding dimension, but this requires more data in the initial sequence leading to predictions which are not efficient.

The same method has been applied also to the high-frequency modes, showing a similar trend. The sampling time is of $\Delta t_{[2]} = 0.2$ and non-linearities are modelled with a second-order polynomial. Tolerances are the same as for the low-frequency case and sparsity is promoted. The dataset dimension consists of 10,000 snapshots and it still considers the 5 most energetic modes for this group. The delay-embedding dimension is $q = 4$ for the example reported in figure 11 and this time the opposite behaviour is observed, *i.e.* as we increase q the fluctuations are dampened out earlier, whilst they are sustained for a wider time span for the low-frequency signals. For both the low- and high-frequency signals the forcing term has been modelled with polynomials up to the 7-th

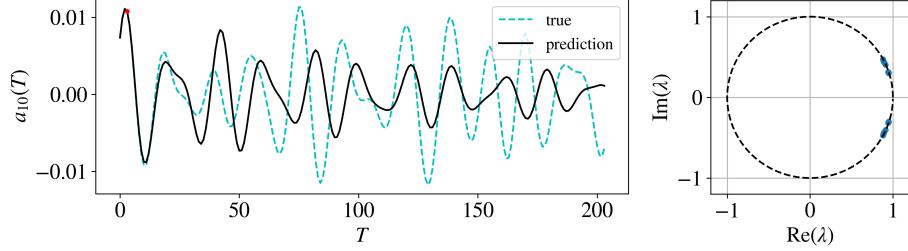


Figure 11: Short-term prediction of the a_{10} coefficient with $q = 4$ on the left. The unitary circle representing the eigenvalues for the matrix \mathbf{A} is shown on the right.

order. No significant improvements were observed for the first group of signals, whereas the behaviour of the predicted fluctuations showed some differences for the high-frequency signals. Figure 12 shows the long-term behaviour of these fluctuations with non-linearities modelled with a 6-th order polynomial and the energy-based tolerances which are $e_r = 10^{-10}$ and $e_p = 10^{-10}$. Although the

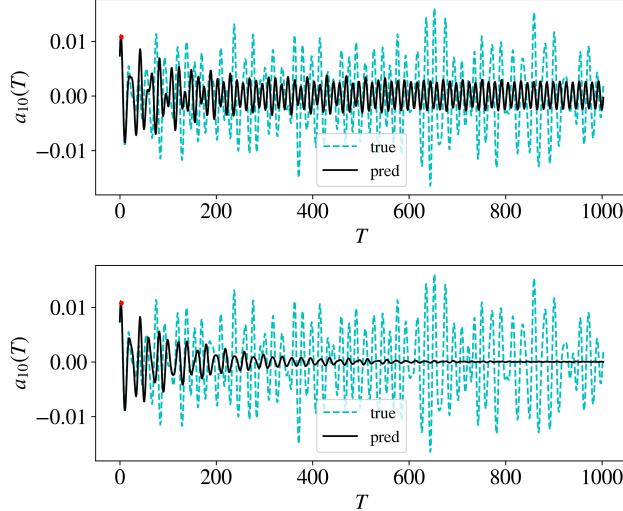


Figure 12: Long-term prediction of the a_{10} coefficient with $q = 4$ and non-linearities modelled with a 6-th order polynomial (top) compared with the long-term prediction for the case in figure 11 (below).

fluctuations of the predicted signal are now sustained for a longer time horizon, the coefficients are still not in the correct range.

In many tests the eigenvalues of matrix \mathbf{A} are on the unitary circle, as it is

reported for the example in figure 11, but a stable behaviour is not observed. Indeed, if we recall equation (13), the system dynamics is not only described by this matrix but also by matrix \mathbf{B} , which expresses the contribution of the forcing term to the evolution of the state in time. By extending the analysis in this direction some more information might be available to understand why the method has not led to successful predictions. The analysis of the \mathbf{B} matrix could also highlight the effect of enhancing or not the sparsity when defining the forcing vector \mathbf{f} , a study that will be carried out in future work.

5. Reduced-order models (ROMs)

In this section we investigate the possibility of implementing reduced-order models (ROMs) and we assess the predictions of an LSTM network on them. The big advantage of these models is given by the possibility to represent turbulence only by using the most relevant structures and, consequently, to handle efficiently the governing physics from large datasets. We also seek a reduction of the time needed to train the network, from the computational point of view. We introduce two different alternatives which are based on using fewer modes for the training of the network with the only difference that (a) in the first case we still want to predict all the 100 modes of the data-driven model (section 5.1) and (b) in the second case we only want to predict the modes which are employed for the training (section 5.2), leading to a reconstruction of the velocity fields which is further simplified, but still able to include the most relevant coherent structures.

The new trained models are built on the identification of sub-groups which share the same features in the homogeneous directions. Six sub-groups are recognized over the first 100 modes, if we exclude the net-flux modes, and have the same wave-number pair. Table 5 shows how the modes are divided and it reports in red an example of the modes which are used for training the new networks. In the example the modes employed for training of what we define ROM1 are highlighted in bold; in this ROM only one mode from each sub-group

enters the training process. A correlation analysis is employed again, this time

Freq. group	Wavenumber pair	Mode index
Low-freq	(0, 1)	1, 4, 7, 9, 24, 26, 42, 60, 64, 94
	(0, 2)	44, 46
High-freq	(1, 0)	19, 21, 37, 39, 47, 50, 63, 74, 95
	(1, 1)	10, 12, 14, 16, 29, 30, 32, 34, 51, 54 56, 58, 77, 85, 90, 91
	(1, 2)	78, 80, 82, 86
	(2, 1)	67, 69, 70, 72, 98

Table 5: Sub-division of the modes according to the various turbulent features. The modes in red are the ones employed to train the reduced-order model ROM1 (and are also the predicted ones for the ROMs which employ an alternative field reconstruction). Degeneracy is retained back when reconstructing the fields.

to understand if the use of fewer modes for a given sub-group is representative of the others. If the correlation factor between signals of the same group is higher than the correlation between signals belonging to different groups, then the signals of the same sub-group have a similar behaviour and they describe the same turbulent structure. This is observed overall in our case, *i.e.* the correlation factor between signals of the same sub-group is of the order of 10^{-2} and it is at least an order of magnitude larger with respect to the correlation factor associated with signals of different sub-groups. This justifies the division in table 5, thus our networks will be able to capture the dynamics of the sub-group with a reduced amount of information required.

5.1. Energy-based reconstruction with 100 modes

The first approach still aims to reconstruct the field with 100 modes in order to compare the predictions of the statistics with the ones of the data-driven case (best model in figure 7). The difference with respect to the latter case is related to the training, where fewer modes are employed. We name the models ROM1 and ROM2, which represent the models which employ either one or two modes for each turbulent feature, respectively. Following the procedure which is exemplified in table 5, this means that we are using either 6 or 12 modes out of 100 for the training instead of 46. Degeneracy needs to be considered

also in this case since, in practice, we only predict one mode for each couple of complex-conjugate modes (thus 46 out of 100, excluding the 7 modes related to the first wavenumber pair).

For the predictions of the reduced models we still employ an LSTM-1-200-10-100 architecture in order to obtain results consistent with the ones reported above. A first aspect which can be noticed is that we achieve a negligible reduction of the time needed to train the network. Interestingly, we can observe a good agreement of the predicted statistics in figure 13, even if the networks employ less information. This is an important benefit when considering a data-driven framework because it shows how turbulence represented by large datasets can be handled efficiently with a reduced amount of data. In order to achieve

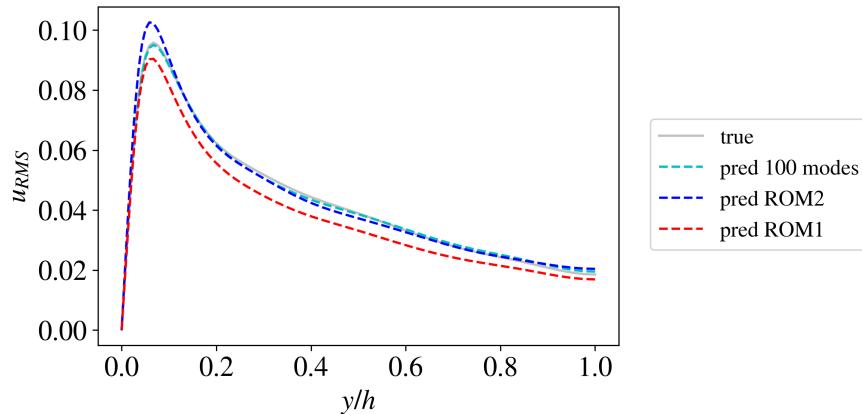


Figure 13: Streamwise velocity fluctuations for the different reduced-order models discussed above. The true profile is reported in grey.

statistical significance, we perform 3 separate trainings (which adds up to six due to the multi-step approach) with the same settings of the hyper-parameters for each reduced-order model case. The profiles in figure 13 represent the predictions of the best model among the three available ones where the errors on the streamwise fluctuations are 6.1% for ROM2 and 9.5% for ROM1. Additionally, it can be observed that an overshoot is present in the case where 12 modes are employed for the training, while the opposite happens when only 6 modes define the training dataset. It might be stated that the specific choice of the modes

employed in the training process significantly affects the fluctuations. Table 6

Model	$E_{u_{\text{RMS}}} [\%]$	val. loss 1	val. loss 2
100 modes	2.97 ± 1.17	7.74×10^{-6}	2.03×10^{-3}
ROM2	8.02 ± 2.19	2.11×10^{-6}	1.49×10^{-3}
ROM1	12.98 ± 3.46	4.35×10^{-6}	1.89×10^{-3}

Table 6: Reduced-order models comparison based on the LSTM-1-200-10-100 architecture. All the 100 modes are considered in the reconstruction. Data is averaged over three different trainings performed with the same parameters.

shows how the error on the statistics increases as fewer modes are employed for training. The variance associated with the different trained models follows this same trend. Note that, the losses are smaller when reducing the number of modes which enter the training process, because the network has less data to validate.

5.2. Reduced models based on an alternative field reconstruction

This approach employs the models developed in the previous section and due to this the validation losses are the same ones as in table 6. The difference lies in the reconstruction of the velocity fields, where here only one or two modes for each turbulent feature are employed. As illustrated in figure 14, the intensity of the streamwise fluctuations is reduced, leading to a less detailed reconstruction, which is still able to incorporate the most relevant structures necessary to describe the minimal-channel turbulence. This is also highlighted in figure 15, where the profiles of the stream-wise fluctuations are reported for the full-model case, *i.e.* the original velocity fields employing all the $n_x \times n_y \times n_z = 66048$ modes, where we consider the number of grid points having $e = n_y$ as the size of the economy-SVD which has to be performed for each wavenumber pair, thus $n_x \times n_z$ times; the data-driven model, where 100 modes are considered, the ROM2 (with 12 modes) and the ROM1 (with 6 modes). The reduction in the reconstructed fluctuations is connected with a smaller fraction of energy captured by the alternative reconstructions, indeed we are able to describe approximately only the 10% or 20% of the total energy of the

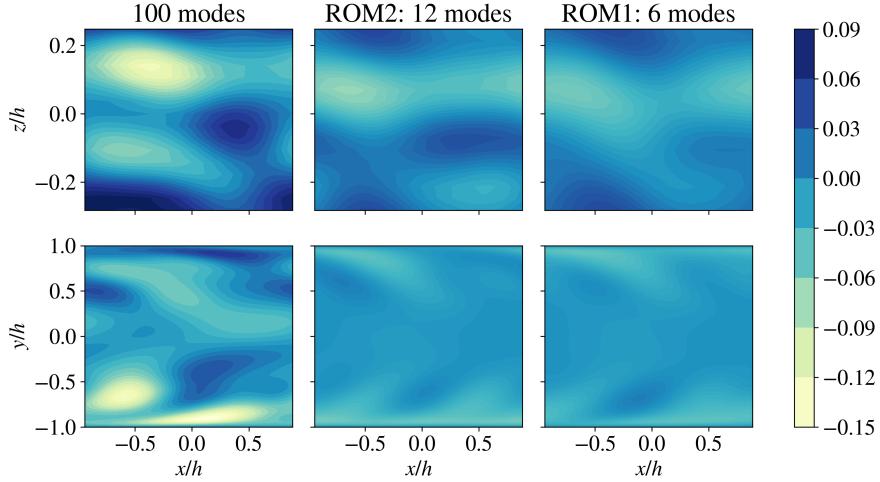


Figure 14: Instantaneous reconstruction of the streamwise velocity fluctuations with: the first 100 most energetic modes (*left*), 12 most energetic modes of each sub-group (ROM2 - *center*) and 6 most energetic modes of each sub-group (ROM1 - *right*). Reconstruction at $t = 10,000$, at (*top*) $z = 0$ and (*bottom*) $y^+ \approx 24$ (distance from the wall in viscous units).

fluctuations $\mathbf{u}'(\mathbf{x}, t)$ with respect to the 97% reconstructed with the 100 modes of the data-driven model, where also degeneracy is accounted for (table 7).

Model	Reconstructed energy %	$E_{u_{\text{RMS}}} [\%]$
100 modes	97.05	2.97 ± 1.17
ROM2: 12 modes	20.71	6.69 ± 3.52
ROM1: 6 modes	10.74	8.51 ± 5.06

Table 7: Alternative ROMs: energy percentage (on fluctuations only) and errors with respect to the reference u_{RMS} profile are reported. In the two ROMs the degeneracy is not considered.

Having the velocity reconstructions which capture a different energy percentage, statistics are compared separately for each one of the alternative models. Once again three different trainings are performed with the same setting to account for the stochastic nature of this process and the average of the errors over the fluctuations is shown in table 7 together with their relative standard deviations. In this case we also observe that the errors and the variance increase as the number of modes which enter the computations is reduced. Figure 16 shows the predictions of the streamwise velocity fluctuation profiles for the ROM2 and

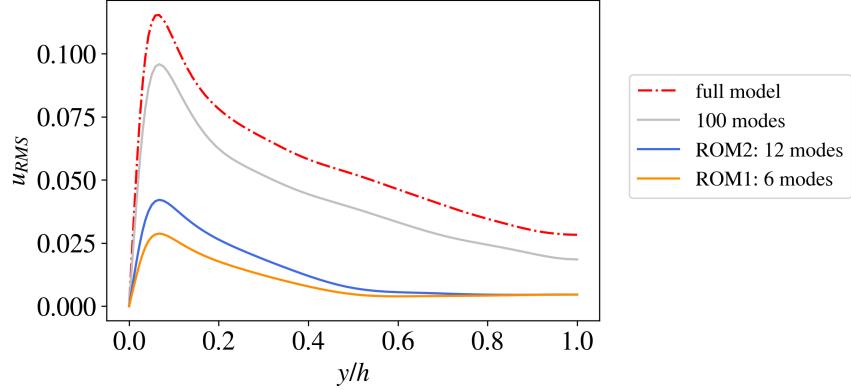


Figure 15: Comparison of the true u_{RMS} profiles when reconstructing the field with all the modes (red), with respect to the data-driven case (100 modes, grey) and the case with 12 and 6 modes (blue and orange, respectively).

ROM1 cases. The true profiles are given by the solid line and the dots show the predicted ones. We have an error of 3.3% with 12 modes included in the reconstruction and 4.7% with 6 modes, whereas 1.3% was the error given by the best model based on a 100-mode energy truncation.

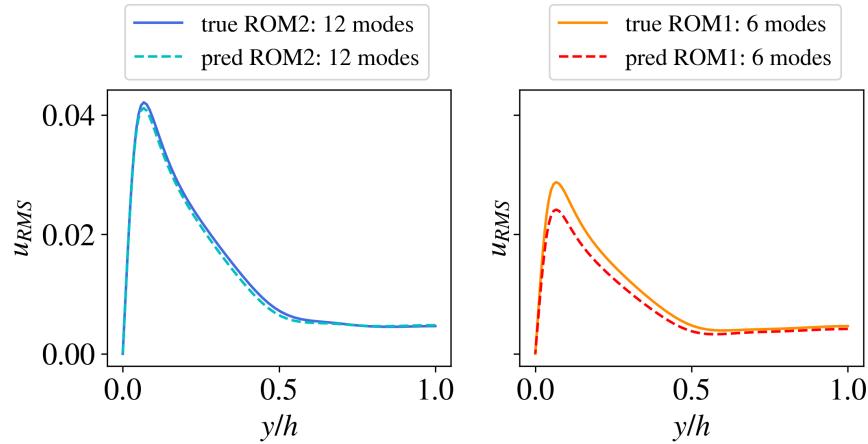


Figure 16: Streamwise velocity fluctuation profiles for the ROM2: 12 modes (*left*) and the ROM1: 6 modes (*right*) alternatives for the best models. The solid line represents the true profile and the dashed shapes the predicted ones.

6. Summary and conclusions

With this analysis we want to explore the capabilities of neural networks to reproduce the chaotic motion of a minimal channel flow. Our previous studies ([Srinivasan et al., 2019](#); [Eivazi et al., 2020](#)) have shown how excellent predictions of the temporal dynamics are achievable for a low-order model of near-wall turbulence ([Moehlis et al., 2004](#)) by using either long-short-term-memory (LSTM) networks or a Koopman-based framework where non-linearities are modelled through a forcing term (KNF). Here, we consider a data-driven model extracted from the outputs of a modal decomposition over which an energy-based truncation is performed to define the number of modes that are needed to describe the minimal channel turbulence in an exhaustive way. The decomposition has been achieved through a proper-orthogonal decomposition in the Fourier domain (FFT-POD), already employed in studies such as the one from [Webber et al. \(1997\)](#). The effectiveness and the benefits of this method have been demonstrated from the computational standpoint, with respect to the POD in the physical domain. The simplified model which enters the computation accounts for the first 100 most energetic modes and captures nearly 80% of the total energy of the fluctuations in the x -direction. The spatial basis associated with the same model also include the most relevant turbulent features which describe the mean flow and its evolution, the streamwise counter-rotating vortices and the tilted vortices, with a dependence in the three directions, which are responsible for the transport of the vorticity. The analysis of the temporal modes has spotlighted the necessity of a multi-step model for the training of the network, having different groups of signal which are characterized by a different frequency content. A final simplification has been obtained by focusing only on the fluctuations not related to wavenumber pair (0,0).

The LSTM network outperforms the KNF, showing an excellent ability to predict the statistical behaviour of the turbulent flow. The best model employs a single layer of 200 neurons with an initial sequence of $p = 10$ and is trained on a dataset comprising 100,000 snapshots (LSTM-1-200-10-100). Predictions with

this network result in an error on the streamwise fluctuations of $E_{u_{\text{RMS}}} = 1.3\%$. Validation losses are of the order of 10^{-6} and 10^{-3} for the low and high frequencies. The instantaneous evolution of the predicted coefficients is in the correct range and it reproduces accurately the frequency content. Moreover, the quality of the predicted physics has been assessed through the Lyapunov exponent and the Poincaré maps, showing the capability of LSTM to reproduce the chaotic nature of the minimal channel turbulence and its dynamic behaviour. The limitations related to the KNF method have been instead highlighted given the data-driven nature of the problem, which makes the dynamics of the system significantly more complex when compared to the simplified case described by the model by [Moehlis et al. \(2004\)](#). We also explored the possibility of implementing several reduced-order models (ROMs). The main idea is to employ a reduced amount of information based on the identification of turbulent structures in order to have a more efficient training. The LSTM network is able to reproduce the statistics associated with the reduced models, even if the accuracy is reduced as fewer modes are accounted during the training.

The present investigation can be extended in different directions: first, the POD analysis could be extended over the three velocity components, so that the complete description of the flow is available. This allows to assess whether the LSTM network is able to derive accurate statistics also in the spanwise and wall-normal directions and to study quantities as the Reynolds shear stress $\langle u'v' \rangle$, for instance. The net-flux modes, *i.e.* fluctuations associated with wave-number pair $(0,0)$, can also be included in the training and predictions. There is still a margin of improvement for the step-by-step predictions by either employing new combinations of the hyper-parameters for the LSTM network or by elaborating a more complex KNF procedure to handle this case. The prediction can also be improved, introducing a more sophisticated network which includes a loss function based on both the instantaneous and statistical behaviour, for instance. Future work will aim at other canonical wall-bounded flows, as pipe and duct flows, open channels ([Guastoni et al., 2021](#)) or boundary layers.

Acknowledgments

RV acknowledges the financial support from the Göran Gustafsson foundation and the Swedish e-Science Research Centre (SeRC). Part of the analysis was performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at PDC and HPC2N.

References

- Arbabi, H., Mezic', I., 2017. Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator. *SIAM J. Appl. Dyn. Syst.* 16, 2096–2126.
- Berkooz, G., Holmes, P., Lumley, J.L., 1993. The proper orthogonal decomposition in the analysis of turbulent flows. *Annu. Rev. Fluid Mech.* 25, 539–575.
- Brunton, S., Proctor, J., Kutz, J., 2016b. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *PANS* 113, 3932—3937.
- Brunton, S.L., Brunton, B.W., Proctor, J.L., Kaiser, E., Kutz, J.N.I., 2017. Chaos as an intermittently forced linear system. *Nat. Commun.* 8, 19.
- Brunton, S.L., Noack, B.R., Koumoutsakos, P., 2020. Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* 52, 477–508.
- Chevalier, M., Schlatter, P., Lundbladh, P., Henningson, D.S., 2007. Simson a pseudo-spectral solver for incompressible boundary layer flows. *Tech. Rep.* .
- 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., 2020. Recurrent neural networks and koopman-based frameworks for temporal predictions in turbulence. *International Journal of Heat and Fluid Flow* 90.

- Eivazi, H., Le Clainche, S., Hoyas, S., Vinuesa, R., 2021a. Towards extraction of orthogonal and parsimonious non-linear modes from turbulent flows URL: <https://arxiv.org/abs/2109.01514>.
- Eivazi, H., Tahani, M., Schlatter, P., Vinuesa, R., 2021b. Physics-informed neural networks for solving reynolds-averaged navier—stokes equations URL: <https://arxiv.org/abs/2107.10711>.
- Fukami, K., Nabae, Y., Kawai, K., Fukagata, K., 2019b. Synthetic turbulent inflow generator using machine learning. *Phys. Rev. Fluids* 4, 064603.
- Gavish, M., Donoho, D., 2014. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Trans. Inf. Theory* 60, 5040—–5053.
- Guastoni, L., Güemes, A., Ianiro, A., Discetti, S., Schlatter, P., Azizpour, H., Vinuesa, R., 2021. Convolutional-network models to predict wall-bounded turbulence from wall quantities. *J. Fluid Mech.* (2021) 928.
- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural Comput.* 9, 1735–1780.
- Jiang, C., Vinuesa, R., Chen, R., Mi, J., Laima, S., Li, H., 2021. An interpretable framework of data-driven turbulence modeling using deep neural networks. *Physics of Fluids* 33. doi:<https://doi.org/10.1063/5.0048909>.
- Jiménez, J., 2018. Machine-aided turbulence theory. *J. Fluid Mech* 854.
- Jiménez, J., Moin, P., 1991. The minimal flow unit in near-wall turbulence. *J. Fluid Mech.* 225, 213–240.
- Khodkar, M., Hassanzadeh, P., Antoulas, A., 2019. A koopman-based framework for forecasting the spatiotemporal evolution of chaotic dynamics with nonlinearities modeled as exogenous forcings. *arXiv preprint arXiv:1909.00076* .
- Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. Conference paper at ICLR 2015 .

- Kutz, J.N., 2017. Deep learning in fluid dynamics. *J. Fluid Mech.* 814, 1–4.
- Li, Q., Dietrich, F., Boltt, E.M., Kevrekidis, I.G., 2017. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator. *Chaos* 27, 103111.
- Lumley, J.L., 1967. The structure of inhomogeneous turbulence. *Atmospheric Turbulence and Wave Propagation* , 166–78.
- Lusch, B., Kutz, J.N., Brunton, S.L., 2018. Deep learning for universal linear embeddings of nonlinear dynamics. *Nat. Commun.* 9, 4950.
- Moehlis, J., Faisst, H., Eckhardt, B., 2004. A low-dimensional model for turbulent shear flows. *New J. Phys.* 6.
- Nakamura, T., Fukami, K., Hasegawa, K., Nabae, Y., Fukagata, K., 2021. Convolutional neural network and long short-term memory based reduced order surrogate for minimal turbulent channel flow. *Phys. Fluids* 33, 025116.
- Proctor, J.L., Brunton, S.L., Kutz, J.N., 2016. Dynamic mode decomposition with control. *SIAM J. Appl. Dyn. Syst.* 15, 142–161.
- 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., 2020. Machine learning strategies applied to the control of a fluidic pinball. *Phys. Fluids* 32, 015108.
- Raissi, M., Yazdani, A., Karniadakis, G.E., 2020. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science* 367, 1026–1030.
- Schmid, P.J., 2010. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* 656, 5–28.

- 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.
- Takeishi, N., Kawahara, Y., Yairi, T., 2017. Learning Koopman invariant subspaces for dynamic mode decomposition, in: *NIPS* 30, pp. 1130–1140.
- Tu, J.H., Rowley, C.W., Luchtenburg, D.M., Brunton, S.L., Kutz, J.N., 2014. On dynamic mode decomposition: Theory and applications. *J. Comput. Dyn.* 1, 391–421.
- Vinuesa, R., Azizpour, H., Leite, I., Balaam, M., Dignum, V., Domisch, S., Felländer, A., Langhans, S.D., Tegmark, M., Nerini, F.F., 2020. The role of artificial intelligence in achieving the sustainable development goals. *Nat. Commun.* 11 233. doi:<https://doi.org/10.1038/s41467-019-14108-y>.
- Vinuesa, R., Brunton, S.L., 2021. The potential of machine learning to enhance computational fluid dynamics URL: <https://arxiv.org/abs/2110.02085>.
- Vinuesa, R., Lehmkuhl, O., Lozano-Durán, A., Rabault, J., 2022. Flow control in wings and discovery of novel approaches via deep reinforcement learning. *Fluids* (Special Issue External Aerodynamics) 62. doi:<https://doi.org/10.3390/fluids7020062>.
- Vinuesa, R., Sirmacek, B., 2021. Interpretable deep-learning models to help achieve the sustainable development goals. *C. Nat. Mach. Intell.* 3 926.
- Webber, G.A., Handler, R.A., Sirovich, L., 1997. The karhunen–loéve decomposition of minimal channel flow. *Phys. Fluids* 9, 213–240.
- Williams, M.O., Kevrekidis, I.G., Rowley, C.W., 2015. A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. *J Nonlinear Sci* 25, 1307–1346.