Expert Systems With Applications 187 (2022) 115910

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A predictive hybrid reduced order model based on proper orthogonal decomposition combined with deep learning architectures

R. Abadía-Heredia [a], M. López-Martín [b], B. Carro [b], J.I. Arribas [b], J.M. Pérez [a], S. Le Clainche [a,*]

[a] ETSI Aeronáutica y del Espacio, Universidad Politécnica de Madrid, Plaza Cardenal Cisneros, 3, 28040 Madrid, Spain
[b] ETSIT, Universidad de Valladolid, Paseo de Belen 15, Valladolid, 47011, Spain

ARTICLE INFO

Keywords:
Reduced order models
Deep learning architectures
POD
Modal decompositions
Neural networks
Fluid dynamics

ABSTRACT

Solving computational fluid dynamics problems requires using large computational resources. The computational time and memory requirements to solve realistic problems vary from a few hours to several weeks with several processors working in parallel. Motivated by the need of reducing such large amount of resources (improving the industrial applications in which fluid dynamics plays a key role), this article introduces a new predictive Reduced Order Model (ROM) applied to solve fluid dynamics problems. The model is based on physical principles and combines modal decompositions with deep learning architectures. The *hybrid ROM*, reduces the dimensionality of a database via proper orthogonal decomposition (POD), extracting the dominant features leading the flow dynamics of the problem studied. The number of degrees of freedom are reduced from hundred thousands spatial points describing the database to a few (20–100) POD modes. Firstly, POD divides the spatio-temporal data into spatial modes and temporal coefficients (or temporal modes). Next, the temporal coefficients are integrated in time using convolutional or recurrent neural networks. The temporal evolution of the flow is approximated after combining the spatial modes with the new temporal coefficients computed. The model is tested in two complex problems of fluid dynamics, the three-dimensional wake of a circular cylinder and a synthetic jet. The hybrid ROM uses data from the initial transient stage of numerical simulations to predict the temporally converged solution of the flow with high accuracy. The speed-up factor comparing the time necessary to obtain the predicted solution using the hybrid ROM and the numerical solver is ∼140–348 in the synthetic jet and ∼2897–3818 in the three dimensional cylinder wake. The robustness shown in the results presented and the data-driven nature of this ROM, make it possible to extend its application to other fields (i.e. video and language processing, robotics, finances).

1. Introduction

Computational fluid dynamics is connected to high-dimensional systems solving complex problems, generally described by a large number of different spatio-temporal flow scales. The need of using large computational resources to properly solve realistic problems and, the high complexity of the underlying physics describing the flow motion, motivates the search of alternative tools providing accurate results, whilst reducing the computational time and memory requirements in the computational facilities. In this line, deep learning models, which are the state-of-the-art solution in many disciplines (i.e. finance, robotics, video and language processing), are not yet fully applied to fluid dynamics predictions, leaving it open new possibilities to explore in this research area.

Modelling complex flows, such as multi-scale, noisy, transitional or turbulent flows, is a research topic of high interest due to its multiple applications in several types of industrial and engineering processes (i.e.: combustion systems, power plants, unmanned vehicles) (Marusic et al., 2003) or their application in natural systems, for instance, describing the propulsion mechanism driving the flow motion in marine animals (DeMont & Gosline, 1988), describing the wake of flying insects (Muijres et al., 2005) or modelling the blood flow in tumours (Abadia et al., 2021b), to name a few. In some cases, turbulence can produce undesirable effects, such as increasing drag in vehicles rising the fuel consumption and consequently the air pollution, increasing drag in industrial devices producing fatigue loads or structural vibrations, or in the case of biological applications, turbulence is

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail addresses: sr.abadia@upm.es (R. Abadía-Heredia), manuel.lopezm@uva.es (M. López-Martín), belcar@tel.uva.es (B. Carro), jarribas@tel.uva.es (J.I. Arribas), josemiguel.perez@upm.es (J.M. Pérez), soledad.leclainche@upm.es (S. Le Clainche).

https://doi.org/10.1016/j.eswa.2021.115910
Received 16 June 2021; Received in revised form 11 September 2021; Accepted 11 September 2021
Available online 25 September 2021

connected with some pathological situations triggering negative biological responses, for instance, coronary artery diseases as a consequence of medical implants (Corrochano et al., 2021; Ferrari et al., 2006). In some other cases, turbulence can lead to positive effects, such as heat transfer enhancement or promoting the fluid mixing, maximizing the efficiency in some industrial systems (Le Clainche et al., 2013). Motivated by all these applications, modelling and predicting the flow behaviour in complex or turbulent flows is a research topic of high interest. Nevertheless, due to the high complexity of the flow solving realistic systems, developing high-accuracy low-dimensional models, also known as reduced order models (ROMs), predicting the temporal evolution of the flow dynamics, is a challenging task, but very useful, i.e. (i) to assess the performance of different flow control strategies (controlling the presence of flow instabilities or turbulence) improving the system efficiency (Brunton & Noack, 2015; Jones et al., 2015; Noack et al., 2011), or (ii) to identify new physical mechanisms driving the flow motion (Lassila et al., 2014; Le Clainche, 2019; Mendez et al., 2019; Sharma, 2011).

Depending on the data available, it is possible to distinguish two types of ROMs: *pre-processed ROMs*, which are based on the Galerkin projection of the full state equations into sub-spaces of smaller dimension that are defined using a modal basis that can be obtained using several techniques (Luchtenburg et al., 2009; Noack et al., 2011; Quarteroni et al., 2016), and *data-driven ROMs*, which are equation-free models providing accurate descriptions of the flow without a priori knowledge of the underlying equations, very useful in the analysis of experimental databases. This second type of ROMs can be developed based on three main approaches. The first approach is based on modal decomposition techniques, which decompose the data analysed as an expansion of hierarchical modes that extract relevant information, mainly based on physical principles, from the full state equations. Hence the dimensionality of the original data is reduced to a few representative modes, i.e. dynamic mode decomposition (DMD) (Schmid, 2010) or proper orthogonal decomposition (POD) (Sirovich, 1987) modes, that can be extrapolated in time (Le Clainche & Ferrer, 2018; Le Clainche et al., 2018a). The second approach uses machine learning tools, based on e.g. deep neural networks (LeCun et al., 2015) or alternative black-box approaches, for data forecasting. This second approach can also be combined with data dimensionality reduction techniques, such as auto-encoders, accelerating and improving the performance of the deep learning model (Omata & Shirayama, 2019). The need of reducing the data dimensionality in fluid dynamics applications is clear, especially in the analysis of computational fluid dynamics databases, where the number of degrees of freedom scale from thousands to millions of points (Brunton et al., 2020). Finally, the third type of approach, for simplicity called as *hybrid ROM*, is a combination of the two previous one, where the data dimensionality reduction is performed using modal decompositions, extracting the underlying physical mechanism leading the flow motion, and combines this reduced-dimensional system with deep learning architectures to construct predictive models (Le Clainche et al., 2021).

This article proposes a new method to perform a hybrid ROM based on physical principles, which combines a reduced basis of POD modes, taking advantage of their physical insight, with different deep learning architectures to estimate the temporal evolution of the flow. The main goal of this work is to provide a ROM, suitable to model problems in fluid dynamics with different applications, reducing the computational time in the numerical simulations from hundred or even thousands of hours to a few seconds. Reducing the computational resources required to solve these types of problems, could be reflected in some benefits for the industry (i.e. improving time-to-market product efficiency in several industrial sectors, among others).

Some previous works explore the idea of combining deep learning algorithms with modal decompositions in fluid dynamics and in other types of applications. In most of the cases, these authors take advantage of the properties of the DMD or POD modal expansions to decompose the flow into groups of spatial modes (the DMD or POD modes) and temporal coefficients (also known as temporal modes in the case of POD expansions). In this way, the deep learning architectures can be used to reconstruct (interpolation) or to predict (extrapolation) the system dynamics only using the information contained in the temporal coefficients. For instance, Iuliano and Quagliarella (2013) combined POD with radial basis functions, namely RBF (i.e. Gaussian, multi-quadratic, inverse-quadratic), to estimate the evolution in time of the POD coefficients (only interpolation, no temporal prediction) to construct surrogate models for optimization in aerodynamic design. Freitag et al. (2018) combined gappy POD with RBF to reconstruct an incomplete spatial dataset and applied recurrent neural networks to such reconstructed field for temporal forecasting in mechanized tunnelling processes. More in connection with the field of fluid dynamics, Guo and Hesthaven (2019) combined the POD temporal coefficient with non-linear regression techniques (Gaussian regression process) to construct a temporal predictive model that was successfully applied for temporal forecasting in the wake of a two-dimensional circular cylinder. Following in this line, but with applications in more complex fluid dynamic problems, Güemes et al. (2019) combined empirical POD (Discetti et al., 2018) with convolutional neural networks to reconstruct temporal flow fields in wall bounded turbulent flows from temporal information contained in wall sensors, in a way equivalent to a linear stochastic estimation. A similar methodology was used by Guastoni et al. (2020) to reconstruct two-dimensional velocity-fluctuation fields at different wall-normal locations in a turbulent open channel flow. Due to the complexity of these problems, the authors divided the domain into smaller subdomains to improve the general flow predictions.

This article also combines POD and deep learning architectures to construct predictive models applied to solve fluid dynamics problems. In contrast to the work presented in Guo and Hesthaven (2019), the advantage of using neural networks to extrapolate in time the POD temporal coefficients instead of other type of non-linear regression techniques, extends the field of application of the previous methodology to solve more complex fluid dynamic problems (i.e., multiple spatio-temporal flow scales, complex non-linear dynamics). The predictive hybrid ROM is tested in two complex applications of fluid dynamics. The first application solves the three-dimensional wake of a circular cylinder in transitional regime. This is a benchmark problem to test new models and applications in fluid dynamics. The database is formed by a group of data collected in the transient stage of a numerical simulation. The model can predict the temporal evolution of the flow, reaching the saturated stage, where the numerical simulation converges in time. Hence, the computational cost of the simulation is drastically reduced from ∼1600 h to less than 3.5 s using the model. The second database models the flow in an axi-symmetric synthetic jet in transition-to-turbulence regime. This type of jet is formed by the periodic oscillations of a cavity that forces the fluid to pass through a jet nozzle, continuously leaving and re-entering into the cavity. This characteristic property of synthetic jets increases the flow complexity, since the topology patterns of the flow change with the fluid movement. Synthetic jets are used in several industrial applications. Some examples include flow control (Cattafesta & Sheplak, 2010; Glezer & Amitay, 2002), enhancement of fluid mixing (Wang & Menon, 2001) or rise in heat transfer (Pavlova & Amitay, 2006). Moreover, synthetic jets also model the movement of some marine animals such a jellyfish, octopus or squids (DeMont & Gosline, 1988).

This article is an extension of the work presented by Lopez-Martin et al. (2021), who used three-dimensional convolutional neural networks for temporal forecasting in complex fluid dynamics problems. The main novelty of this work lies in the data-dimensionality reduction strategy carried out in the original data, before applying deep learning architectures for time predictions. This dimensionality reduction converts the highly complex data, formed by hundred thousands degrees of freedom, into small spaces composed by 20–100 degrees of freedom,

simplifying the deep learning architecture requirements. More specifically, accurate temporal predictions have been carried out using simple models formed by one-dimensional convolutional and recurrent neural networks. Moreover, the data-driven nature of the model presented, and the robustness presented in the applications carried out, show that this hybrid ROM can be extended to analyse other type of data of similar nature (complex and high-dimensional time-series data) from other areas, such as network intrusion detection, network traffic forecasting or video prediction.

The highlights of this work include: (i) the development of hybrid ROMs based on physical principles, (ii) combining modal decompositions (POD) with neural networks for temporal forecasting, (iii) providing a robust architecture to predict the non-linear dynamics in fluid dynamic problems, (iv) a new model that is able to predict the saturated regime in numerical simulations using data from the transient solution and, (v) the reduction of the computational time to generate fluid dynamic data from hundred or even thousands of hours to a few seconds. Unfortunately, the method has only been tested in periodic solutions. Nevertheless, the good performance of the model and the complexity presented in the two problems analysed, makes it a promising tool that could be extended for the analysis of more complex problems in future research.

The article is organized as follows. Section 2 describes the methodology to construct the hybrid ROM based on combining POD modal decompositions with deep learning strategies. Section 3 introduces the problems studied and shows the main results obtained. Finally, the main conclusions are presented in Section 4.

2. Methodology

This section introduces the main algorithm to generate the new predictive ROM, fully data-driven, presented in this article, called as the hybrid ROM. The algorithm combines POD, to reduce the data dimensionality based on the physical principles of the flow, with deep learning predictive models based on convolutional (CNN) and recursive (RNN) neural networks. It is possible to download the code and test the hybrid ROM in Abadia et al. (2021a).

For simplicity, the data are organized in the following snapshot matrix,

$$\boldsymbol{X} = \boldsymbol{V}_1^K = [\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_k, \boldsymbol{v}_{k+1}, \dots, \boldsymbol{v}_{K-1}, \boldsymbol{v}_K], \tag{1}$$

with K the total number of snapshots and $\boldsymbol{v}_k$ the variable of the flow field considered at time instant t_k. The dimension of the matrix is $J \times K$, with $J = N_x \times N_y \times N_z$, being N_x, N_y and N_z the number of grid points defined along the streamwise, normal (or radial) and spanwise spatial components of the domain. In the applications presented below, each snapshot contains spatio-temporal information, which is modelled by the velocity vector evaluated on each grid point defining the computational domain. Hence, the velocity vector at time instant t_k, described as $\boldsymbol{v}(x, y, z, t_k) = \boldsymbol{v}_k$ where x, y and z represent the streamwise, normal (or radial) and spanwise velocity components, is re-organized as a one dimensional array with dimension $pJ \times 1$, being p the number of components forming the vector. In the two applications presented below, one of them is three-dimensional (x, y and z, with $p = 3$ and $\boldsymbol{v} = (v_x, v_y, v_z)$), while the other is two-dimensional (x and y, with $p = 2$ and $\boldsymbol{v} = (v_x, v_y)$). The dimension of the snapshot matrix (1) is then $pJ \times K$. For simplicity, the algorithm presented below is described considering the three spatial components.

2.1. Hybrid reduced order model based on proper orthogonal decomposition

Proper Orthogonal Decomposition (POD) is a technique introduced by Lumley (1967) to identify coherent structures in turbulent flows. The method decomposes the flow field as a combination of POD modes, $\boldsymbol{\Phi}_j(x, y, z)$, which are obtained by means of optimizing the mean square of the field under consideration. POD modes form an orthogonal basis that is optimal in terms of kinetic energy, which models the spatial patterns of the flow field combined with some temporal coefficients $\boldsymbol{c}_j(t)$, as

$$\boldsymbol{v}(x, y, z, t) \simeq \sum_j \boldsymbol{c}_j(t)\boldsymbol{\Phi}_j(x, y, z). \tag{2}$$

The classical algorithm behind POD is based on the covariance of a state vector changing with time. However, the size of this covariance matrix is a function of the spatial degrees of freedom of the data, generally making prohibitively expensive solving problems in fluid dynamics. These problems are usually defined by two- or three-dimensional grids containing thousands or even millions of points. As an alternative, Sirovich (1987) introduced the snapshot method, also known as singular value decomposition (SVD). Both terms, SVD and POD, are generally used interchangeably in the literature, hence POD and SVD modes are the same.

Applying SVD to the snapshot matrix (1), the spatio-temporal data are decomposed into the POD modes $\boldsymbol{U}$ (orthogonal in space), the temporal coefficients $\boldsymbol{T}$, and the singular values $\boldsymbol{\Sigma}$ (representing the amount of energy of each mode and its contribution to the flow field) as

$$\boldsymbol{V}_1^K \simeq \boldsymbol{U}\,\boldsymbol{\Sigma}\,\boldsymbol{T}^\top, \tag{3}$$

where $()^\top$ denotes the matrix transpose, $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{T}^\top\boldsymbol{T}$ are unitary matrices with dimension $N \times N$, the (diagonal) matrix $\boldsymbol{\Sigma}$, contains the singular values $\sigma_1, \dots, \sigma_N$ and N is the number of selected POD modes. These modes are ranked in decreasing order by their singular values, where the most energetic (highest singular value) modes provide accurate representations of the general flow dynamics (large size flow scales, experimental data free of noise, et cetera). The number of N selected modes is calculated according to a certain tolerance ε (tunable) as

$$\sigma_{N+1}/\sigma_1 \leq \varepsilon. \tag{4}$$

It is remarkable that this tolerance could represent the level of noise (i.e.: in experimental data) or the relative size of the flow structures that will be used to approximate the general flow field. In these two examples, the modes with singular values below the tolerance could be connected with noise, or small flow scales, and will be neglected to represent the general flow dynamics. The error made in the SVD approximation considering N modes compared to the original data, is measured by means of the relative root mean square error ($RRMSE^{POD}$) as

$$RRMSE^{POD} = \frac{\|\boldsymbol{V}_1^K - \boldsymbol{U}\,\boldsymbol{\Sigma}\,\boldsymbol{T}^\top\|}{\|\boldsymbol{V}_1^K\|}, \tag{5}$$

where $\|\cdot\|$ is the L2-norm.

For simplicity, Eq. (3) is re-written as

$$\boldsymbol{V}_1^K \simeq \boldsymbol{U}\,\hat{\boldsymbol{T}}, \tag{6}$$

with $\hat{\boldsymbol{T}} = \boldsymbol{\Sigma}\,\boldsymbol{T}^\top$ called as the *temporal modes* and $\boldsymbol{U}$ are known as the *spatial modes*, or the *POD modes*, as mentioned before. The matrix $\hat{\boldsymbol{T}}$, of dimension $N \times K$, contains the information regarding the temporal evolution (K snapshots) of the flow field in rows. Hence, each row corresponds to each of the N temporal coefficients $\boldsymbol{c}_j(t)$ (with dimension $1 \times K$) from Eq. (2). If these temporal coefficients are extrapolated, or evolved (integrated) in time for P time steps, then, the dimension of the new matrix $\hat{\boldsymbol{T}}'$ is $N \times K'$, with $K' = P + K$. Hence, it is possible to reconstruct the original data using Eq. (3), obtaining a new snapshot matrix $\bar{\boldsymbol{V}}_1^{K'}$ as

$$\bar{\boldsymbol{V}}_1^{K'} \simeq \boldsymbol{U}\,\hat{\boldsymbol{T}}'. \tag{7}$$

The new snapshots, $K+1, K+2, \dots, K'$, of this matrix are the flow predictions. This is the essence of the predictive ROM introduced in this article. More specifically, we use a deep learning architecture based

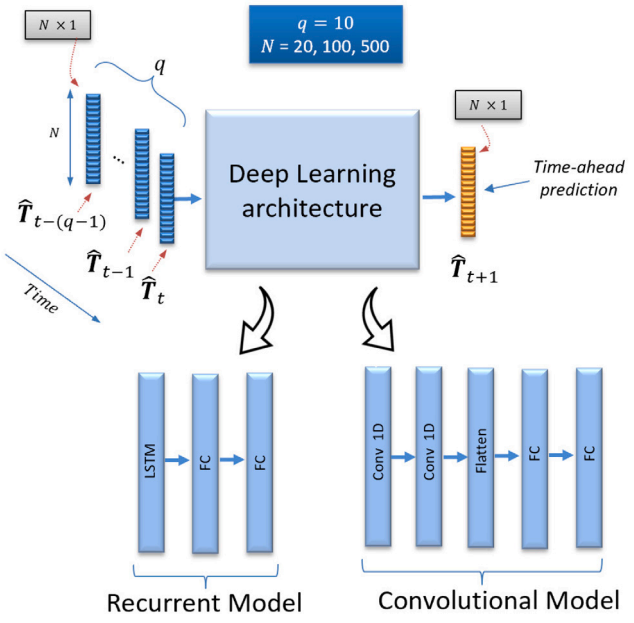

Fig. 1. Types of deep architectures to create the predictive ROM: recurrent and convolutional neural networks. In both cases, the model predict a time-ahead snapshot, $\hat{\boldsymbol{T}}_{t+1}$, based on the previous q snapshots. N is the number of POD modes.

on CNN and RNN to solve (integrate) in time the temporal modes $\hat{\boldsymbol{T}}$, resulting into new snapshots of the flow field obtained at a very reduced computational cost.

The predictive capabilities of this hybrid ROM are tested using the RRMSE, which compares the original data $\boldsymbol{V}_{K+1}^{K'}$ with the predictions carried out by the deep learning model in the physical space $\bar{\boldsymbol{V}}_{K+1}^{K'}$ as:

$$RRMSE = \frac{\|\boldsymbol{V}_{K+1}^{K'} - \bar{\boldsymbol{V}}_{K+1}^{K'}\|}{\|\boldsymbol{V}_{K+1}^{K'}\|}. \tag{8}$$

2.2. Deep learning prediction models

This section introduces the deep learning models that have been applied to predict the temporal evolution of the temporal modes $\hat{\boldsymbol{T}}$. The models proposed predict the snapshot $K+1$ given at time $t+1$ in the temporal matrix, for simplicity written as $\hat{\boldsymbol{T}}_{t+1}$, using the information from the previous q snapshots as $\hat{\boldsymbol{T}}_t, \hat{\boldsymbol{T}}_{t-1}, \hat{\boldsymbol{T}}_{t-2}, \ldots, \hat{\boldsymbol{T}}_{t-q+1}$. As explained in the previous section, each snapshot represents a column in the matrix $\hat{\boldsymbol{T}}$. An sketch representing the deep learning architectures used, RNN and CNN, is shown in Fig. 1.

On the one hand, the recurrent model is a multilayer neural network formed by long short-term memory (LSTM) (Yu et al., 2019) layers and fully connected (FC) layers. We have considered a different number of units (dimensionality of the output space) for the LSTM layer. On the other hand, the convolutional model is composed of convolutional (Conv) layers followed by FC layers. Considering the matrix structure of the dataset (columns of $\hat{\boldsymbol{T}}$) we have used Conv 1D (one-dimensional) layers (Kiranyaz et al., 2021) that apply a one dimensional kernel, with a stride of 1 and no padding. For the convolutional models, it is necessary to include a flatten function between the convolutional and fully connected layers to adapt the matrix structure used by the Conv 1D layers to a vector structure used by the FC layers. The architecture details for the convolutional and recurrent models are presented in Tables 1–2. These tables show the number and type of layers, with specific details for each layer, the activation function, Linear or Rectified Linear Unit (ReLU), and the tensor dimension of the layers.

The training and validation strategy for the deep learning models is presented in Fig. 2.

Table 1
Architecture details in the RNN. The layers are long short-term memory (LSTM) or fully connected (FC), N is the number of SVD modes, the number of predictors is $q = 10$, the activation functions are Linear or Rectified Linear Unit (ReLU). The number of Kernels/neurons (# neurons) and the tensor dimension (Dimension) are indicated for each layer.

# Layer	Layer details	# Neurons	Activation function	Dimension
0	Input	N		$10 \times N$
1	LSTM	100 or 400	ReLU	100 or 400
2	FC	100	ReLU	100
3	FC	N	Linear	N

For such aim, the columns of matrix $\hat{\boldsymbol{T}}$ are separated in three blocks forming three small matrices that are used for the training, validation and test of the neural network. The dimension of these small matrices is $N \times K_{training}$ (training), $N \times K_{validation}$ (validation) and $N \times K_{test}$ (test), where the total number of columns (snapshots or time samples) is represented as $K_{training}+K_{validation}+K_{test} = K$. The test data are extracted with a rolling-window method as outlined in Fig. 3. This method generates data batches with q inputs and one output as expected by the prediction model (Fig. 1). The offset considered between the successive rolling windows is 1 (constant).

To train the models we use batch stochastic gradient descent algorithm to minimize the Mean Squared Error Loss (MSE_{Loss}) between the real ($\hat{\boldsymbol{T}}_t^{real}$) and predicted ($\hat{\boldsymbol{T}}_t^{predicted}$) snapshots of the temporal matrix. This error is firstly calculated for each time prediction, $MSE_{Loss}(t)$ (local error), as

$$MSE_{Loss}(t) = \frac{1}{N}\|\hat{\boldsymbol{T}}_t^{predicted} - \hat{\boldsymbol{T}}_t^{real}\|^2, \tag{9}$$

being N the number of singular values. The global loss (MSE_{Loss}) is based on averaging the local loss calculated for each time prediction, over the total temporal extension of the matrix as

$$MSE_{Loss} = \frac{1}{K_\alpha}\sum_{K_\alpha} MSE_{Loss}(t). \tag{10}$$

This temporal extension is indicated by K_α, which can be $K_{validation}$ or K_{test} (see Fig. 2), depending on whether we obtain the global loss: in the validation or the test sets, respectively. Table 3 provides a summary of the length of the training ($K_{training}$), validation ($K_{validation}$) and test (K_{test}) sets for each one of the datasets used in this research.

To obtain the best network parameters, and as a regularization mechanism, we use *early stopping* over the validation set. In other words, the loss ($MSE_{Loss}(t)$) over the validation set is computed at the end of each epoch (an epoch is a complete pass of all training samples), and the training stops when this loss is not reduced after a certain number of epochs (*patience period*). Then, the total loss is calculated with the final size of the matrix K_α used for the training or validation.

The training parameters for the neural networks used in this research are the following: (a) optimization based on Adam (Kingma & Ba, 2020) using the default values for the parameters ($\alpha = 0.001$ for the learning rate, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ see details in Kingma & Ba, 2020), (b) mini-batch gradient descent with a batch size of 5 and 100 epochs with early stopping and a patience period of 10 epochs, (c) 10 snapshots used to predict the next time-ahead snapshots ($q = 10$).

Finally, the RRMSE is also calculated for each time step to quantify the quality in the predictions carried out by the neural network as

$$RRMSE(t) = \frac{\|\hat{\boldsymbol{T}}_t^{predicted} - \hat{\boldsymbol{T}}_t^{real}\|}{\|\hat{\boldsymbol{T}}_t^{real}\|}. \tag{11}$$

It is remarkable that the strongly sequential and high-dimensional nature of the data, together with a relatively small amount of training data, have been the main drivers for choosing the proposed recurrent and convolutional models. These models are appropriate for a high-dimensional time series data structure, since they use a relatively small number of training parameters, which is necessary to avoid overfitting.

Table 2
Same as Table 1 for the CNN.

# Layer	Layer details	# Neurons	Kernel size	Stride	Padding	Activation	Dimension
0	Input	N					$10 \times N$
1	Conv 1D	30	3	1	No	ReLU	8×30
2	Conv 1D	60	3	1	No	ReLU	6×60
3	Flatten						360
4	FC	100				ReLU	100
5	FC	N				Linear	N

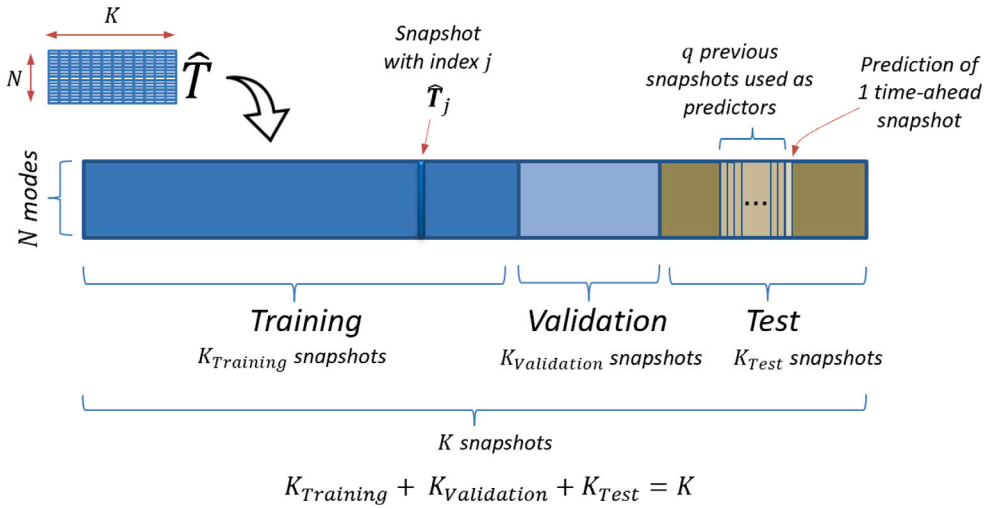

Fig. 2. Data structure used for training, validation and test in the predictive models.

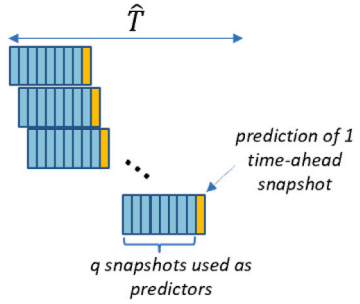

Fig. 3. Rolling window method used to extract the inputs (q) and expected outputs (one) for the predictive models.

Table 3
Number of snapshots for the training, validation and test sets for each one of the datasets analysed: the three-dimensional cylinder wake and the synthetic jet.

Dataset	$K_{training}$	$K_{validation}$	K_{test}	K
Cylinder	255	45	199	499
Jet	3411	601	12100	16112

This is the reason for initially not using other successful alternative architectures, such as those based on non-iterative neural networks with random weights (Cao et al., 2020, 2018; Wang & Cao, 2018), which also have a reduced number of training parameters but have been less studied in their ability to incorporate sequential data, in addition to the resulting high dimensionality of the input vectors in case the q snapshots required as predictors were flattened into a single vector. Nevertheless, studying the application of these alternative architectures remains an open topic for future research.

3. Application to fluid dynamics

The model described in this article has been tested in two different complex problems: the three-dimensional wake of a circular cylinder and a synthetic jet in transition-to-turbulence regime. The databases are generated numerically, and in both cases, the data analysed contain information from the initial transient of the numerical simulations. The hybrid ROM predicts the temporal evolution of the database, modelling both the transient and the temporally converged solutions, when the flow is saturated. The speed-up factor compares the total time necessary to develop and predict a solution using the predictive ROM, which considers the time employed in the training interval (t_{tr}) and the time employed for the temporal predictions (t_{pred}), with the computational time required by the numerical solver (t_{num}) to simulate the predicted interval (K_{test}, see Table 3), as

$$\text{Speed-up} = \frac{t_{num}}{t_{pred} + t_{tr}}. \tag{12}$$

The governing equations describing the movement of the viscous flow, incompressible and Newtonian, for the wake of a circular cylinder and the synthetic jet, are the Navier–Stokes equations, written in non-linear form as

$$\nabla \cdot \boldsymbol{v} = 0, \tag{13}$$

$$\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \nabla)\boldsymbol{v} = -\nabla p + \frac{1}{\mathrm{Re}} \Delta \boldsymbol{v}, \tag{14}$$

where $\boldsymbol{v}$ and p are the non-dimensional velocity vector and pressure, respectively, and Re is the Reynolds number, defined as $\mathrm{Re} = VL/\nu$, where ν is the kinematic viscosity of the fluid, V is the characteristic flow velocity, and L is the characteristic length. As usual, the equations are non-dimensionalized with the units for length and time, L and L/V.

3.1. Predictions in the three-dimensional wake of a circular cylinder

The flow past a circular cylinder is a benchmark problem in fluid dynamics that serves to illustrate a large variety of applications. The

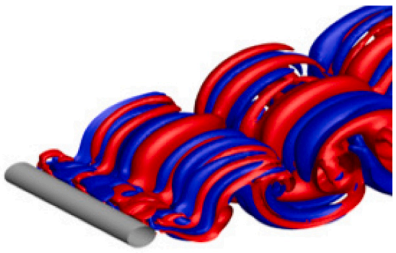

Fig. 4. Iso-surfaces of spanwise vorticity in the three-dimensional wake of a circular cylinder at Re= 280 and $L_z = 1.66$.

flow bifurcations found in the wake of a circular cylinder are characterized by the variations in the Reynolds number, as previously defined Re= VD/ν, where D is the cylinder diameter and V is the incoming free stream velocity. At low Reynolds numbers, the flow remains steady. The first flow bifurcation is identified at Re$\simeq$ 46, where a Hopf bifurcation triggers the two dimensional flow oscillations in the cylinder wake, forming a von Karman vortex street (Jackson, 1987). Increasing the Reynolds number, the flow becomes three-dimensional for some specific values of the spanwise length $L_z = 2\pi/\beta$, where β is the spanwise wavenumber (Barckley & Henderson, 1996).

The present article uses the database presented in Le Clainche et al. (2018c) solving the three-dimensional wake of a circular cylinder at Re= 280 in a computational domain with spanwise length $L_z = 1.66$, where the solution is periodic. Fig. 4 shows the iso-surfaces of spanwise vorticity, defined as $\frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y}$, in the wake of the circular cylinder modelling this flow.

The numerical database has been generated using the open source numerical code Semtex (Blackburn & Henderson, 1999) to solve the fully nonlinear, incompressible Navier–Stokes Eqs. (13)–(14). The dimensions and boundary conditions of the computational domain are set as in the literature (Barckley & Henderson, 1996) and the cylinder diameter is $D = 1$ (the spatial components are normalized with the cylinder diameter). The boundary conditions imposed are: Dirichlet for velocity $v_x = 1$ and $v_y = v_z = 0$ (for the streamwise, normal and spanwise components), and Neumann conditions for the pressure at the inlet, upper and lower boundaries of the domain; Neumann conditions for velocity and Dirichlet conditions for the pressure at the outlet boundary; periodic boundary conditions in the spanwise direction with period L_z, where the equations are discretized using 64 Fourier equidistant collocation planes. The conditions to model the cylinder boundaries are Dirichlet for velocity ($v_x = v_y = v_z = 0$) and Neumann for pressure. The dimensions of the computational domain (non-dimensionalized with the cylinder diameter D) are $L_y = \pm 15$ in the normal direction and $L_x = 15$ and $L_x = 50$ in the streamwise direction upstream and downstream the cylinder. For simplicity, a smaller computational domain, which focuses on the cylinder wake, is used to construct the hybrid model, with dimensions $L_y = \pm 2$ and $L_x = 10$ in the streamwise direction downstream the cylinder.

The database is formed by 599 snapshots collected starting from the beginning of the numerical simulations (time $t = 0$). The snapshots are equi-distant in time with time step $\Delta t = 1$. As presented in Fig. 5, showing the evolution of the streamwise, normal and spanwise velocity components in a representative point of the cylinder wake, from time 0–100 there is a clear transient region in the streamwise and normal velocity components. The spanwise velocity is zero until time 300, and is not fully developed until time $\sim$ 350. Hence, the transitory of the numerical simulation is extended at least to time 350, however, the simulation is not converged at least until time 500 (see details in Le Clainche et al., 2018c).

The database has been analysed to identify the most relevant POD modes and reducing the data dimensionality. As it is presented below, the different flow regimes (transient and saturated) identified in the

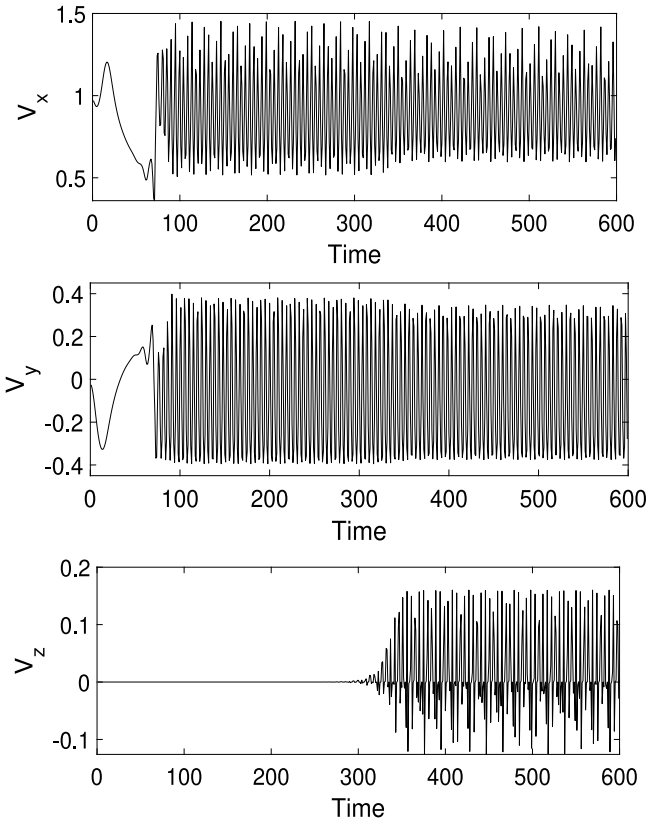

Fig. 5. Temporal evolution of the streamwise (top), normal (middle) and spanwise (bottom) velocity components in the three-dimensional cylinder wake at Re= 280 extracted at $(x, y, z) = (1.5, 1, L_z/2)$.

temporal variations of the velocity components along the numerical simulations, are also identified in such POD modes. The first 100 snapshots (from time 0–100) have been removed from the dataset to avoid the transient region identified in the streamwise and normal velocities, resulting in a new database formed by $K = 499$ snapshots (solution from time 100–599). The POD modes have been computed in four different matrices modelling the cylinder wake containing $K = 50$ (time 100–150), 100 (time 100–200), 300 (time 100–400) and 499 (time 100–599) snapshots. In the first two cases, the spanwise velocity is still not developed, in the third case this velocity is under development, and most of the data are transient, finally the last case also considers the saturated regime of the numerical simulation (fully converged). Fig. 6 compares the POD (spatial) and temporal modes, $\boldsymbol{U}$ and $\hat{\boldsymbol{T}}$, number $N = 1$, 2, 5, 10 and 20.

Fig. 6 - left shows the variation of the POD modes along the different grid points (for the sake of clarity the figure shows a zoomed in view, since the solution is periodic). As seen, in the modes $N = 1$ and 2 the results are similar, some small variations are identified in the mode $N = 5$ and larger variations are identified in the modes $N = 10$ and 20, especially in the cases analysed with 50 and 100 snapshots. This result suggests that even using a small number of snapshots to compute the POD modes, most of them collected in the transient region of the simulation, the spatial structure of the POD modes is quite robust, at least in the most energetic modes, justifying the good performance of the predictive ROM presented here. On the contrary, in the temporal modes (Fig. 6 - right), the variations between the different cases are more evident, in good connection with the variations of the flow dynamics at the different stages of the simulation. Moreover, these variations are also evidenced in the case considering all the snapshots. In the snapshot number 250, which represents the time instant 350, it is possible to identify a clear change in the tendency of the modes,

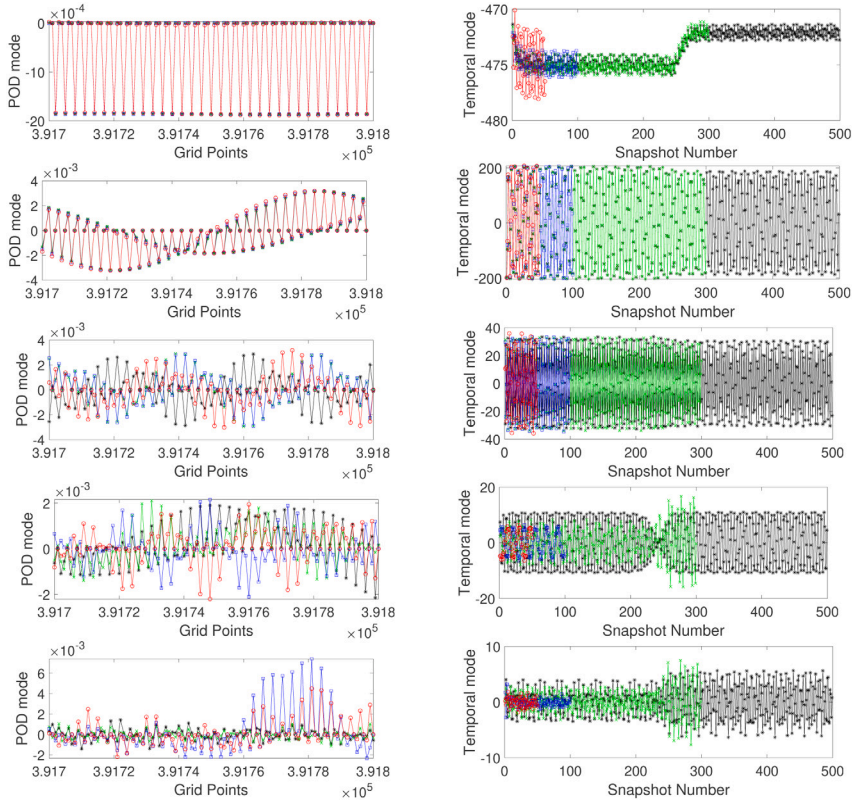

Fig. 6. POD applied in the three-dimensional wake of a circular cylinder. POD is calculated in the snapshot matrices composed by the data collected in the temporal intervals 100–150 (red), 100–200 (blue), 100–400 (green) and 100–499 (black). POD modes $\boldsymbol{U}$ (left) and temporal modes $\hat{\boldsymbol{T}}$ (right) number $N = 1, 2, 5, 10, 20$ (from top to bottom). The POD modes present a zoomed in view in some specific grid points for the sake of clarity (the solution is periodic).

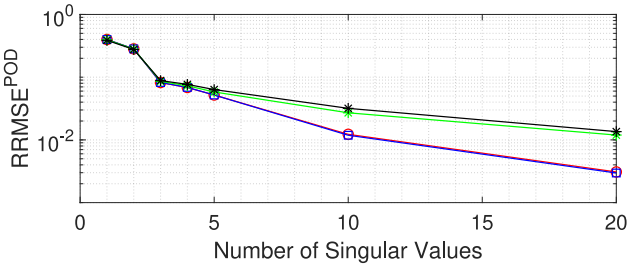

Fig. 7. $RRMSE^{POD}$ (as in Eq. (5)) calculated in the three-dimensional wake of a circular cylinder as function of the number of N POD modes. POD is calculated in the snapshot matrices composed by data from the temporal intervals 100–150 (red), 100–200 (blue), 100–400 (green) and 100–499 (black).

especially in $N = 1$, 10 and 20. This fact is in good agreement with the presence of the spanwise velocity, which is zero up to time 300 and fully developed at time 350. The temporal modes are maintained constant for times larger than 400 (snapshot 300), suggesting that this point could be considered as the end of the transitory in the numerical simulations.

The error made in the reconstruction of the original matrix using a different number of POD modes ($RRMSE^{POD}$) is presented in Fig. 7 for the four matrices analysed. The results show that retaining $N = 20$ POD modes, the reconstruction error is smaller than ∼1.5% in all the cases. Retaining $N = 10$ modes, this error is smaller than 1% in the two smallest matrices and smaller than 2% in the two largest matrices. Finally, if only $N = 3$ modes are retained, the error is smaller than 9% in all the cases. It is important to remark that most of the data analysed are formed by the snapshots collected in the transitory of the numerical simulation.

Based on the previous results, the predictive ROM is constructed using a snapshot matrix of 255 snapshots for the training interval (see Table 3) collected from time 100–355, which considers the transient stage of the simulations, where the spanwise velocity is developed only in the last 5 snapshots of the data analysed (see previous Fig. 5). The goal of this test is to show the capabilities of the predictive hybrid ROM when the information contained in the data analysed is incomplete or deficient, which results in a highly complex problem. The temporal matrix of modes is generated using $N = 20$ POD modes, assuming the error of ∼1.5% in the reconstruction of the data, as explained before. The three different deep learning architectures previously introduced, a 1D CNN (Conv 1D), and the two RNN (LSTM) using 100 and 400 layers have been used to predict the saturated regime of the numerical simulations (up to time 599). The $MSE_{Loss}(t)$ (as in Eq. (9)) and the $RRMSE(t)$ (as in Eq. (11)) is calculated in the test set (predictions) for the different architectures, as presented in Fig. 8. As seen, the order of magnitude of these errors is similar in all the cases, although they present different temporal fluctuations. The $MSE_{Loss}(t)$ is maintained smaller than $2.3 \cdot 10^{-4}$ and the $RRMSE(t)$ is smaller than 2.3%. These results show the good performance of the different architectures to carry out temporal predictions.

The global error, defined in Eq. (8), of the predictions carried out in the three architectures is RRMSE∼8%. It is important to note that this global error assumes the initial error of the database retaining 20 POD modes (∼1.5%) and the error made in the predictions of the neural networks (2.3% as maximum error), hence the total error made in the reconstruction of the physical space can be estimated as RRMSE∼8 − 1.5 − 2.3 = 4.2% when the RRMSE in the neural networks is maximum and as RRMSE∼8 − 1.5 = 6.5%, assuming negligible the error made by the neural networks. Figs. 9 and 10 compare the contours and iso-surfaces of the streamwise, normal and spanwise velocities approximated with the deep learning architectures, with the original

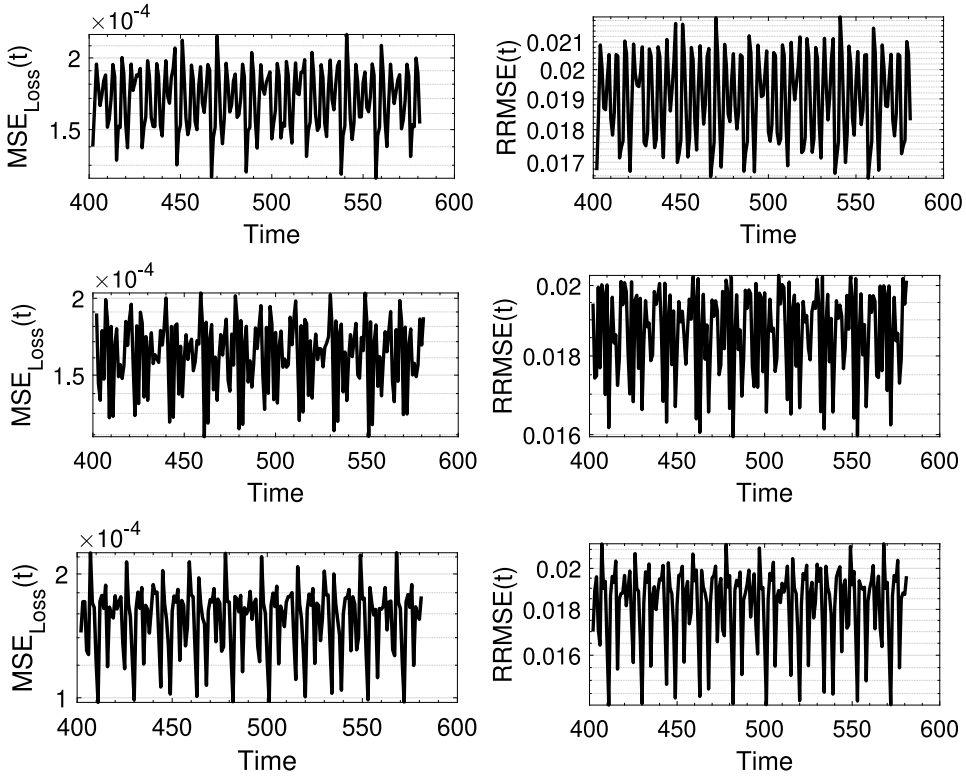

Fig. 8. From left to right: $MSE_{Loss}(t)$ (as in Eq. (9)) and $RRMSE(t)$ (as in Eq. (11)) calculated in the test set ($K_{test} \equiv$ flow predictions). From top to bottom: Conv 1D LSTM with 100 neurons and LSTM with 400 neurons.

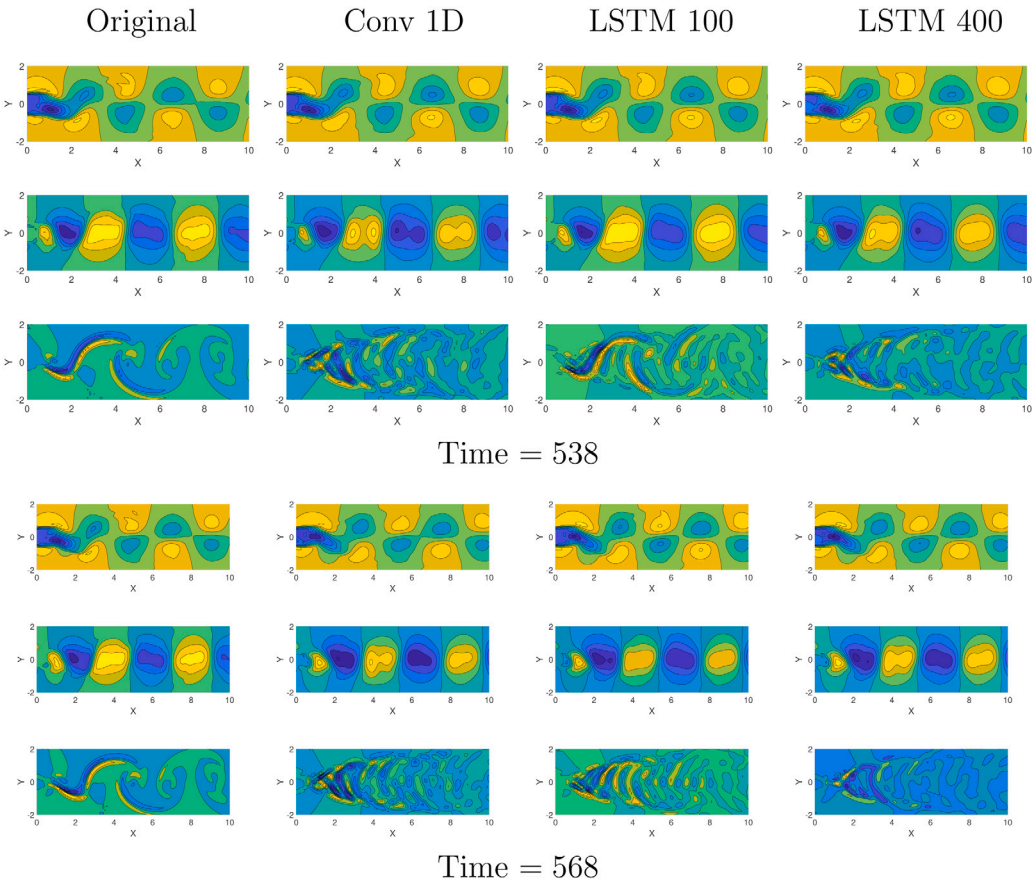

Fig. 9. From top to bottom: streamwise, normal and spanwise velocity components in the three-dimensional wake of a circular cylinder extracted in two representative snapshots of the flow predictions. From left to right: original data and flow reconstruction using the CNN and LSTM architectures with 100 and 400 neurons.

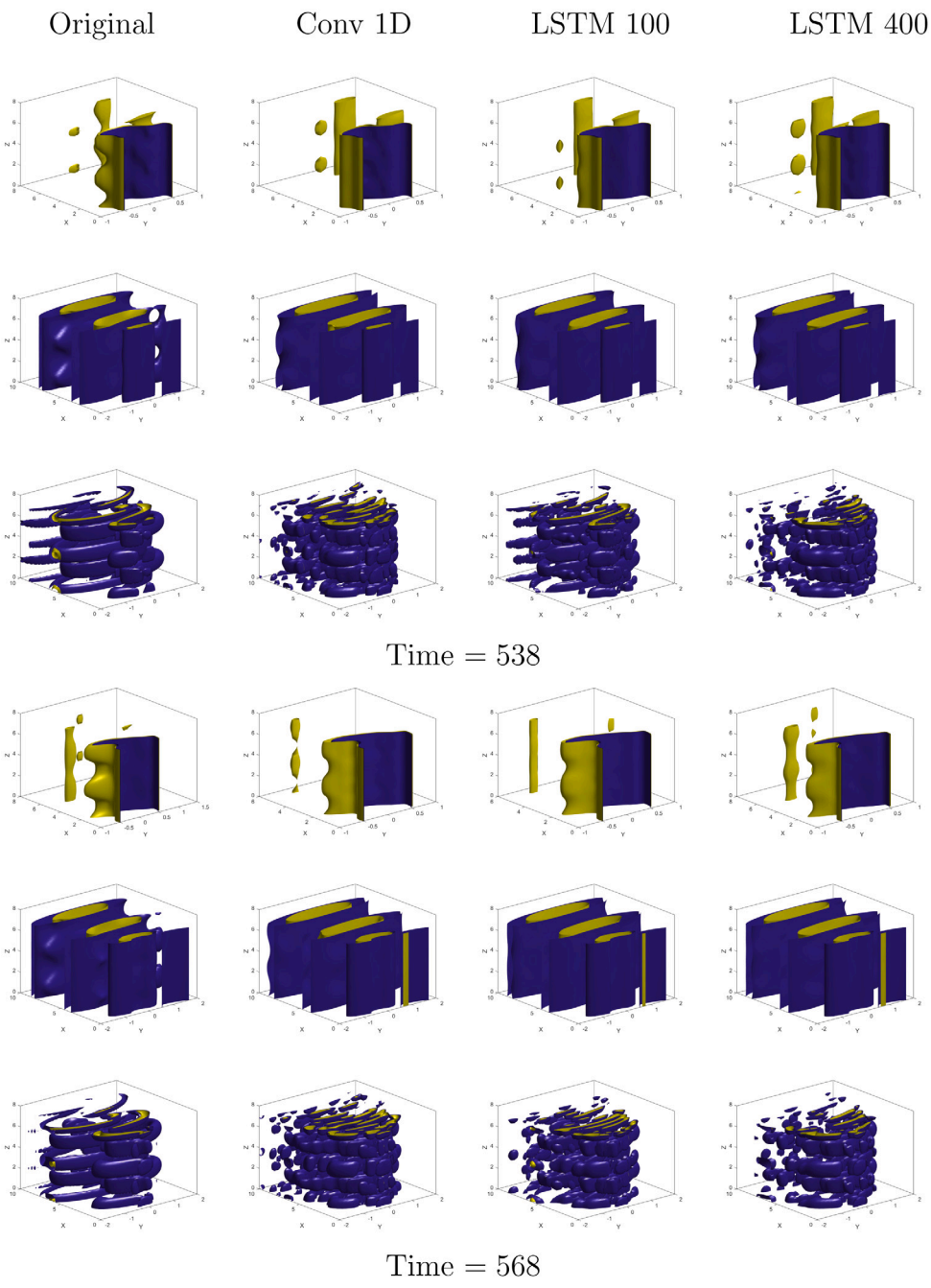

Fig. 10. Same as Fig. 9 but showing the iso-surfaces for the streamwise, normal and spanwise velocities with levels $0.4v_x$ and $0.2v_x$ for the streamwise component, $0.25v_y$ and $0.1v_y$ for the normal component and $0.2v_z$ and $0.1v_z$ for the spanwise component, respectively. The velocity components are normalized with their corresponding maximum value.

data at two representative snapshots in the saturated regime (time 538 and 568). As seen, the streamwise and normal velocity components are in qualitative good agreement in all the cases. However, some differences are found in the spanwise velocity, which justifies the rise in the global RRMSE compared to the predictions carried out by the neural networks. Moreover, the fact that most of the training intervals of the neural networks only consider the transient spanwise velocity justifies this underperformed behaviour. The model LSTM with 100 neurons is able to identify some of the patterns characteristic of this velocity, providing the best results in the spanwise velocity. Using a larger training interval considering more snapshots where the spanwise velocity component is developed, would improve the results presented. Nevertheless, as mentioned before, this article aims at testing and showing the capabilities of using deep learning architectures combined to POD modes to construct predictive ROMs in critical situations, with the aim at showing the potential of this methodology and extend it to solve more complex problems in future research.

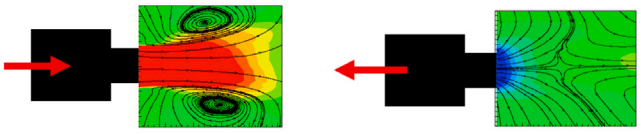

Fig. 11. Streamwise velocity contours and streamlines defining the two characteristic topology patterns in synthetic jets. Left: the flow is injected through the jet nozzle (vortex ring). Right: the flow re-enters into the cavity by the jet nozzle (saddle point).

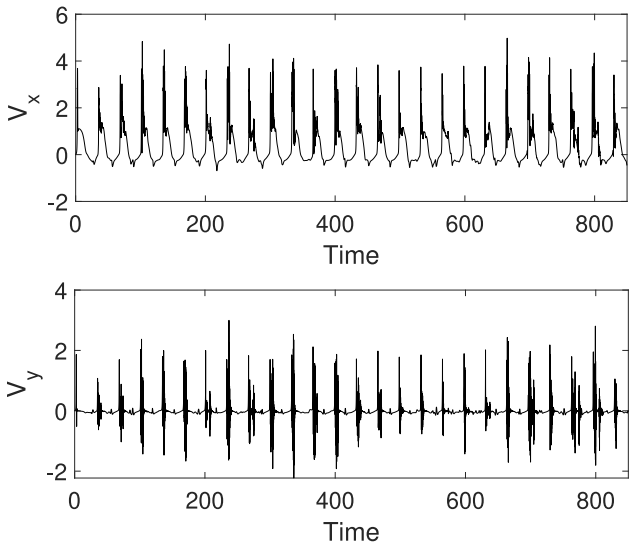

Fig. 12. Temporal evolution of the streamwise (top), and radial (bottom) velocity components in the synthetic jet at Re= 1000 and St= 0.03 calculated in the point $(x, r) = (0.2, 2)$.

The computational time necessary to predict these three-dimensional data using the hybrid ROM and the time required to train the neural networks is presented in Table 4. The table also shows the speed-up factor of this ROM as defined in Eq. (12). The computational time of the numerical solver to solve the saturated regime of the three-dimensional numerical simulations, generating 199 snapshots (K_{test}) is ∼1600 h (100 h using 16 processors in a computer with 64 Gb of RAM memory and a processor *i*5). The simulations assumed a Fourier expansion in the spanwise direction, such that each kernel simulated four adjacent two-dimensional planes. This multiplied by 16 cores gives a total of 64 planes. More details on this simulation can be found in Le Clainche et al. (2018b). Hence, the speed up factor is larger than 2897 in the three architectures presented. As expected, the Conv 1D architecture presents the largest speed-up factor and the LSTM 400 architecture presents the smallest one.

Table 4
Computational time (in minutes) necessary for the temporal predictions and for training the neural networks using the different architectures in the three-dimensional cylinder wake. Speed-up factor, defined as in Eq. (12).

Architecture	Prediction time (t_{pred})	Training time (t_{tr})	Speed-up
Conv 1D	0.006	2.509	3817.097
LSTM 100	0.014	3.138	3045.685
LSTM 400	0.023	3.290	2897.675

3.2. Predictions in a synthetic jet

The predictive hybrid ROM is now tested in an axi-symmetric synthetic jet. This type of flow, also known as zero-net-mass flux jet, is formed by a jet stream that is generated by the oscillations of a piston or a membrane inside a cavity that is periodically injecting and suctioning fluid through a jet nozzle. The jet stream generates momentum, although the net mass flux in the jet is zero.

The complexity of this type of flow is reflected in the two different topology patterns characteristic of this geometry. In the injection phase, a vortex ring transporting momentum is generated from the contact of the flow with the edges of the jet nozzle, while in the suction phase, a saddle point separates the flow travelling downstream from the flow re-entering into the cavity. Fig. 11 shows the streamwise velocity and the streamlines defining the two main topology patterns describing the flow in synthetic jets.

The flow is characterized by two non-dimensional parameters (Carter & Soria, 2002): (i) the Strouhal number, St= $f\,D/V$, where f

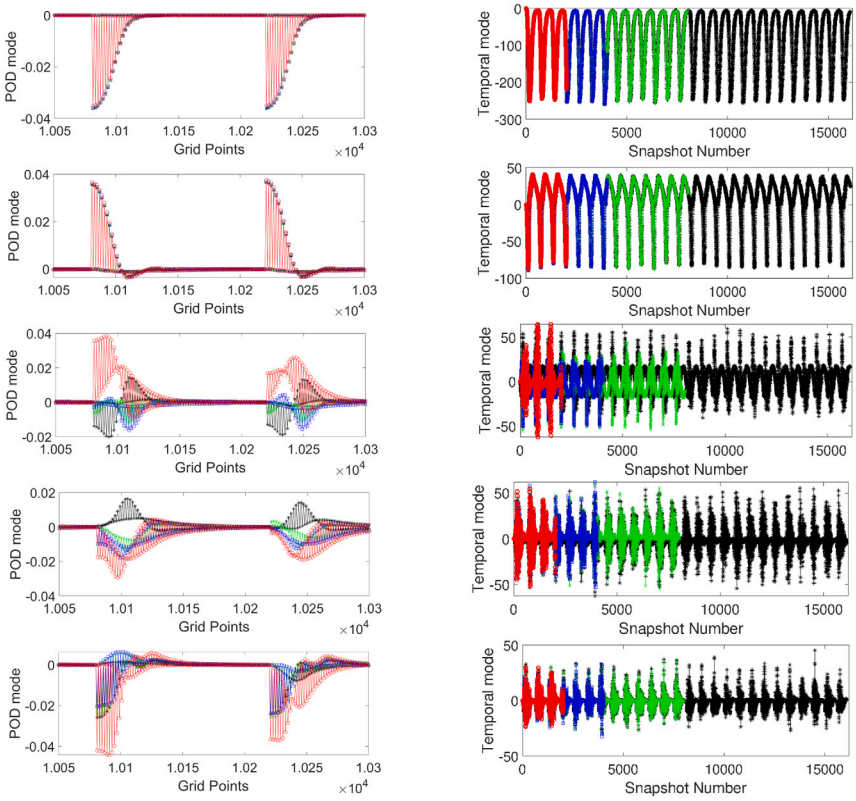

Fig. 13. Same as Fig. 6 for the synthetic jet.

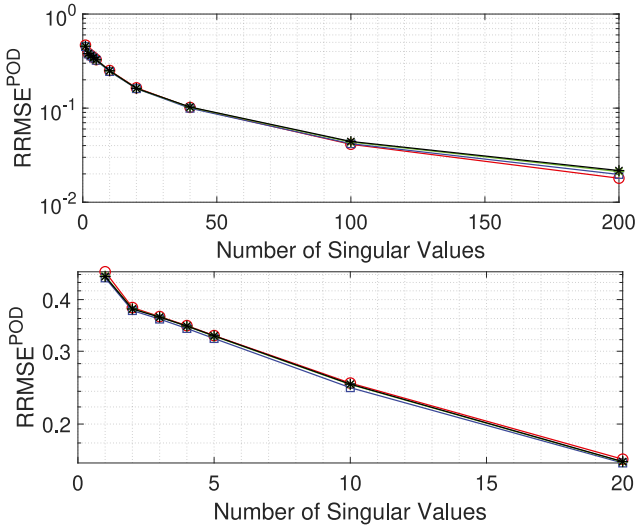

Fig. 14. Same as Fig. 7 for the synthetic jet. Top: general view. Bottom: zoomed in view.

is the piston oscillation frequency, D is the jet orifice diameter and V is a characteristic velocity scale, which is based on the momentum velocity defined as $V = \frac{D_p}{D}\frac{\hat{A}_p}{\sqrt{2}}$, where D_p and $\hat{A}_p$ are the piston diameter and the piston peak oscillation velocity (see Le Clainche et al., 2017 for more details), and (ii) the Reynolds number, as previously defined Re= $V D/\nu$.

The present article uses the database presented in Le Clainche (2019), Le Clainche et al. (2020) solving an axi-symmetric synthetic jet at St= 0.03 and Re= 1000, in transition-to-turbulent regime (see details in the diagram presented in Carter & Soria, 2002). The database is formed by 16112 snapshots equi-distant in time with time step Δt = 0.053, which represent 25.77 cavity oscillation cycles. Each oscillation cycle is formed by 625 snapshots (~33.3 time units), hence the total database represents the evolution of the velocity field in ~854 time units. The evolution of the streamwise and radial velocities in a representative point of the database is presented in Fig. 12 (the spatial components are normalized with the jet diameter D). Although the database contains the transient solution from the numerical simulations (the transitory extends to the 13th cycle as explained in Le Clainche, 2019; Le Clainche et al., 2020), the velocity components show the evolution in time of a regular and periodic signal, in good agreement with the oscillating frequency imposed in the cavity upstream the jet nozzle.

To generate the database, numerical simulations were carried out using the numerical code Nek5000 (Fischer et al., 2008). The simulations are converged in time (saturated flow) after 13 cycles (see details in Le Clainche, 2019; Le Clainche et al., 2020). The boundary conditions set in the numerical solver are $v_x = v_p(t) = \hat{V}_p \cdot \sin(2\pi f t)$ and $v_r = 0$ for the streamwise and radial velocity components, respectively, and Neumann boundary conditions for the pressure at the inlet boundary. Non-slip boundary conditions ($v_x = v_r = 0$) are imposed to model the jet wall. Axi-symmetric conditions are imposed in the bottom part of the domain. Finally, zero-stress and Dirichlet boundary conditions are imposed for the velocity ($\nabla\mathbf{u}\cdot\mathbf{n}$=0, with $\mathbf{n}$ = unit normal) and pressure ($p = 0$), respectively, in the top and outlet surfaces. To avoid boundary reflections, the length and radius defining the external boundaries of the computational domain are $L = 80D$ and $H = 40D$. The remaining dimensions of the computational domain are: $D = 1$ and $D_p = 5D$ for the jet and cavity diameters, and $L_c = 5D$, $L_o = 0.2D$, $H_c = D_p/2 = 2.5D$ and $R = 0.5D$ for the cavity length, jet stroke, cavity radius and jet orifice radius, respectively.

Fig. 13 shows the POD and temporal modes, $\boldsymbol{U}$ and $\hat{\boldsymbol{T}}$, computed in four different snapshot matrices for the synthetic jet, containing the snapshots from the beginning of the numerical simulations.

These matrices are formed by K = 2014, 4028, 8056 and 16,112 snapshots, representing 3.22, 6.44, 12.88 and 25.77 (full database) cavity oscillation cycles. The three first cases represent the transient region of the numerical simulations, when the flow is not converged

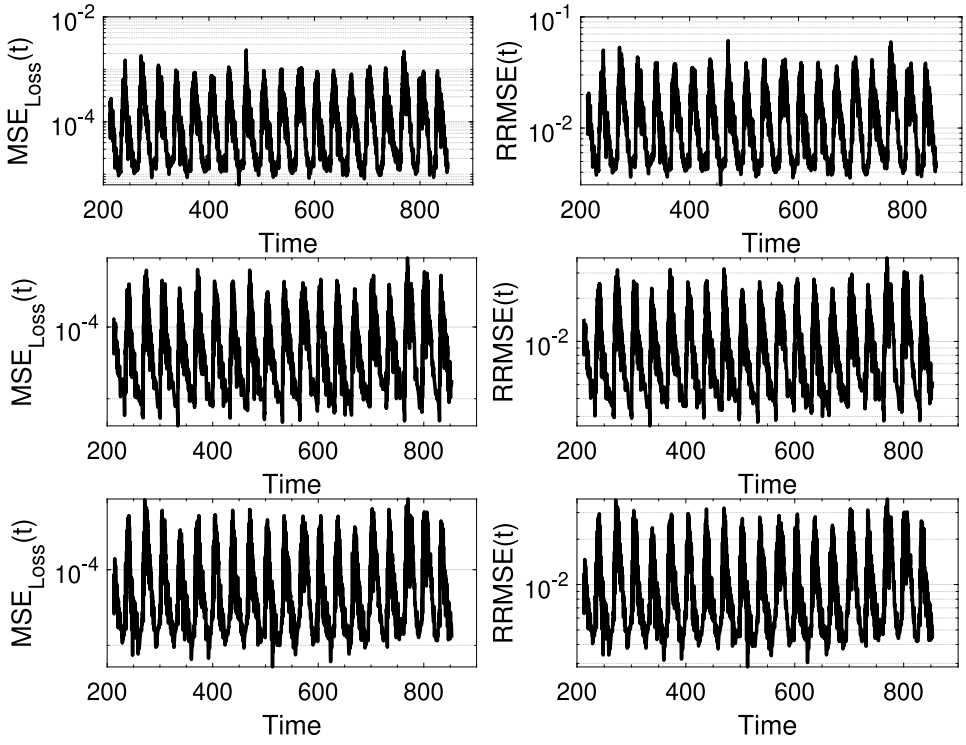

Fig. 15. Same as Fig. 8 for the synthetic jet case with $N = 20$ modes.

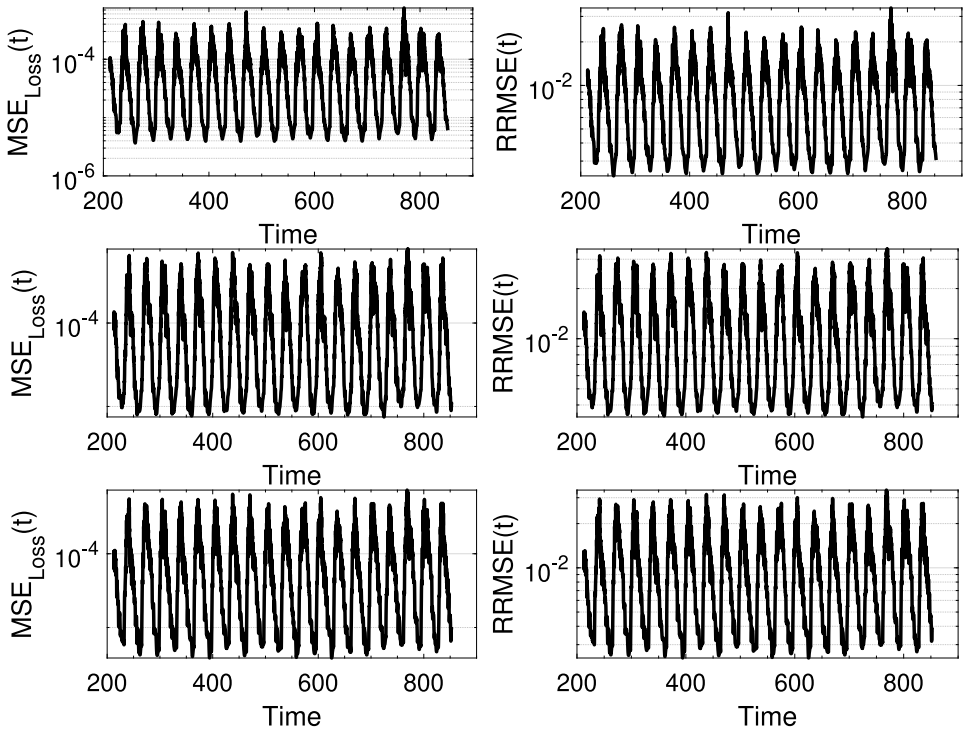

Fig. 16. Same as Fig. 8 for the synthetic jet case with $N = 100$ modes.

and still changing in time. The figure compares the modes number N = 1, 2, 5, 10 and 20. Regarding the spatial modes (Fig. 13 - left), the first and second POD modes are similar for all the test carried out, however, the remaining modes present different shapes among the different test. This result suggests that, as in the cylinder wake, the spatial structure of the most energetic POD modes is quite robust and permanent in time, even when the data analysed are transient. Regarding the temporal modes (Fig. 13 - right), the solution obtained in the four cases analysed is similar for all the modes. This result is justified by the fact that the flow oscillates with an imposed forcing frequency, so the temporal dynamics of the flow is driven by this frequency and its harmonics since the beginning of the numerical simulations (see details about the physical mechanisms driving the flow in Le Clainche et al., 2020).

Fig. 14 shows the error ($RRMSE^{POD}$) made when the original matrix is reconstructed as function of the number of POD modes retained.

The curves present a similar tendency in the four analyses carried out. More specifically, this error is ∼2% for the reconstruction carried out retaining 200 modes, and ∼16% when the number of modes retained is 20. In contrast to the cylinder wake flow presented in the previous section, it is necessary retaining a larger number of modes to obtain the same accuracy in the reconstruction, suggesting that, although the temporal dynamics in the synthetic jet is simpler than in the previous case (the temporal modes presented in Fig. 13 - right are similar in all the cases), the complexity of the spatial structures is much larger in the jet flow.

The predictive model has been tested in a temporal matrix formed by 3411 snapshots (see Table 3), which represents 5.45 cavity oscillation cycles. Two tests have been carried out keeping $N = 20$ and 100 modes for the temporal matrix. The $MSE_{Loss}(t)$ (as in Eq. (9)) and the $RRMSE(t)$ (as in Eq. (11)) calculated in the predictions (test set) for the different architectures is presented in Figs. 15 and 16 for the cases with $N = 20$ and 100 modes. As seen, this error is similar in the three cases, as well as the temporal fluctuations, which follow a periodic scheme, in contrast to the case of the cylinder wake, where the temporal evolution of the case studied was influenced by the development of the spanwise velocity component. The errors made in the Conv

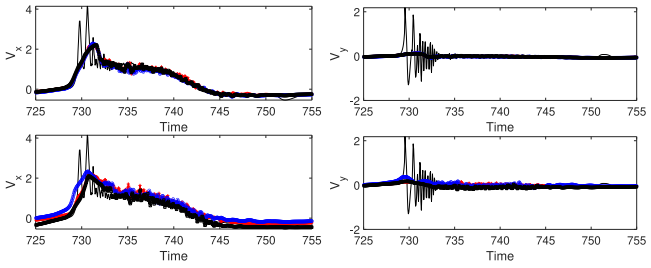

Fig. 17. Streamwise (left) and radial (right) velocities in the synthetic jet in the 24th cavity oscillation cycle. Original data (black thin line) and the predictions carried out with the deep learning architectures: Conv 1D (red), LSTM with 100 neurons (blue) and LSTM with 400 neurons (black thick line). Temporal matrix composed by $N = 20$ (top) and 100 (bottom) modes. Point extracted at $(x, r) = (0.2, 2)$.

1D architecture in the case considering 20 modes is slightly larger than in the remaining cases. More specifically, this error varies within the intervals $MSE_{Loss}(t) \in [6 \cdot 10^{-5}, 3 \cdot 10^{-3}]$ and $RRMSE(t) \in [0.3, 6]\%$ for the Conv 1D in the case with 20 modes and $MSE_{Loss}(t) \in [8 \cdot 10^{-5}, 10^{-3}]$ and $RRMSE(t) \in [0.2, 3.5]\%$ for the remaining architectures in the case with 100 modes.

The global error, defined as in Eq. (8), made in the temporal predictions calculated in the physical space is RRMSE∼18% for both the cases using $N = 20$ and 100 modes. This error considers the prediction error from the neural networks ($RRMSE(t) \in [0.3, 6]\%$ for the Conv 1D with 20 modes and $RRMSE(t) \in [0.2, 3.5]\%$ for the remaining cases) and also the reconstruction error the original matrix retaining, which is ∼16% and 5% for $N = 20$ and 100, respectively. The reason of obtaining this global RRMSE in both test cases is presented in Fig. 17 that compares the evolution of the streamwise and radial velocities on the original data with the reconstructed fields predicted using the different neural network architectures in the 24th cycle (last minus one cycle of the entire database).

As seen, the original flow field presents high frequency peaks with high amplitude localized in the region with the highest velocity, which

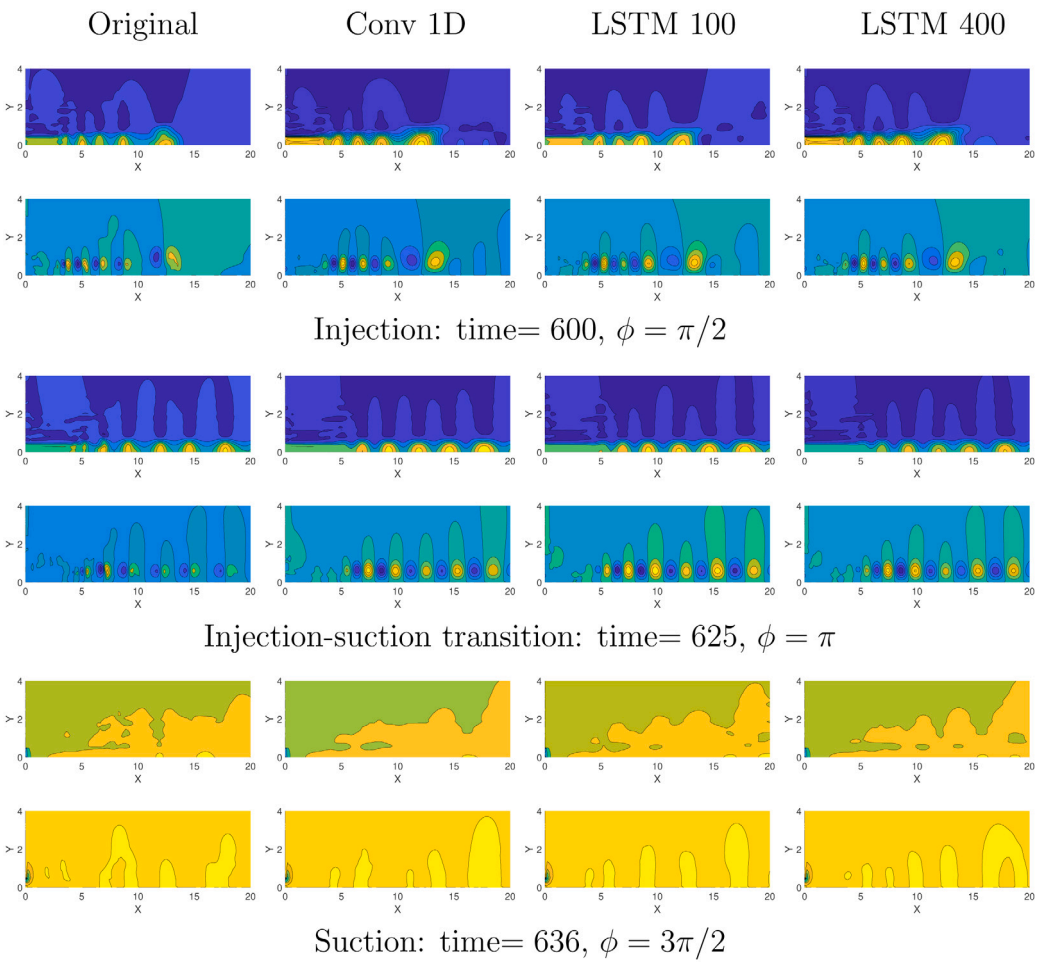

Fig. 18. From top to bottom: streamwise and radial velocity components in the synthetic jet extracted at several representative snapshots from the cycle 18th (time $\in [600, 634]$). From left to right: the original data and the predictions carried out using the CNN and LSTM architectures with 100 and 400 neurons and $N = 20$ POD modes. A complete oscillation cycle is defined in the interval $[0, 2\pi[$, where ϕ represents the phase of the cycle within such interval.

corresponds to the injection phase and the formation of the vortex ring characteristic of this type of flow. These high frequency peaks could be connected with the presence of velocity fluctuations and the origin of flow instabilities in the jet flow, as presented in Le Clainche (2019), Le Clainche et al. (2020), not detailed in this article for the sake of brevity. To properly represent such high frequency peaks it would be necessary to retain a larger number of modes to reconstruct the original solution with high accuracy and increase the snapshots number from the original database. However, (i) this would increase the dimension of the temporal matrix and the complexity of the dataset analysed (consequently it could lead to some difficulties in the performance of the different deep learning architectures) and, (ii) it would also reduce the predictive capabilities of the model, decreasing the speed-up factor compared to the original solution. However, this article aims at showing the performance of a new predictive hybrid ROM combining the most relevant POD modes with a robust architecture of neural networks. This ROM is valid to predict the temporal evolution of the most relevant patterns driving the flow dynamics, since it is based on physical principles. The natural extension of this ROM in future works would be its application to most complex non-periodic flows, where only the dominant patterns can be predicted, as presented in various examples from the literature (Guastoni et al., 2020; Güemes et al., 2019; Le Clainche & Ferrer, 2018; Le Clainche et al., 2018a, 2021). The synthetic jet flow analysed is in transition-to-turbulent regime, which justifies the solution presented. Moreover, similar results in synthetic jets are obtained when using other types of ROMs (see Le Clainche, 2019; Lopez-Martin et al., 2021).

The contours of the streamwise and radial velocities for the predictions carried out with the deep learning architectures in the data formed by $N = 20$ and 100 POD modes are compared in Figs. 18 and 19, respectively, with the original velocity field at several representative snapshots. As seen, the results are in qualitatively good agreement in all the cases.

It is notorious that in the case retaining $N = 100$ modes the results obtained do not improve the case with $N = 20$ modes. This suggests that the relevant patterns representing the flow are described by a few relevant POD modes. Increasing the number of modes, would improve the description of the small flow structures present in the flow (small flow perturbations), reducing the global RRMSE. However, as mentioned before, this would increase the complexity of the neural network. The present predictive ROM is based on physical principles, solving in time the evolution of the main patterns driving the flow dynamics.

Table 5 compares the speed-up factor and the computational time necessary for training the neural networks using the different architectures in the cases with $N = 20$ and 100 modes and the time necessary for the temporal predictions in the synthetic jet. The computational time of the numerical solver to solve ∼20 oscillation cycles of the cavity, and generate 12100 snapshots (K_{test}) is ∼192 h (24 h using 8 processors in a computer with 32 Gb of RAM memory and a processor *i5*). As presented in the table, using the three different deep learning architectures the speed-up factor is always larger than 168 for the case with $N = 20$ and larger than 148 for the case with $N = 100$, being the Conv 1D architecture the case with the largest speed-up factor and the LSTM 400 architecture the case with the smallest one.

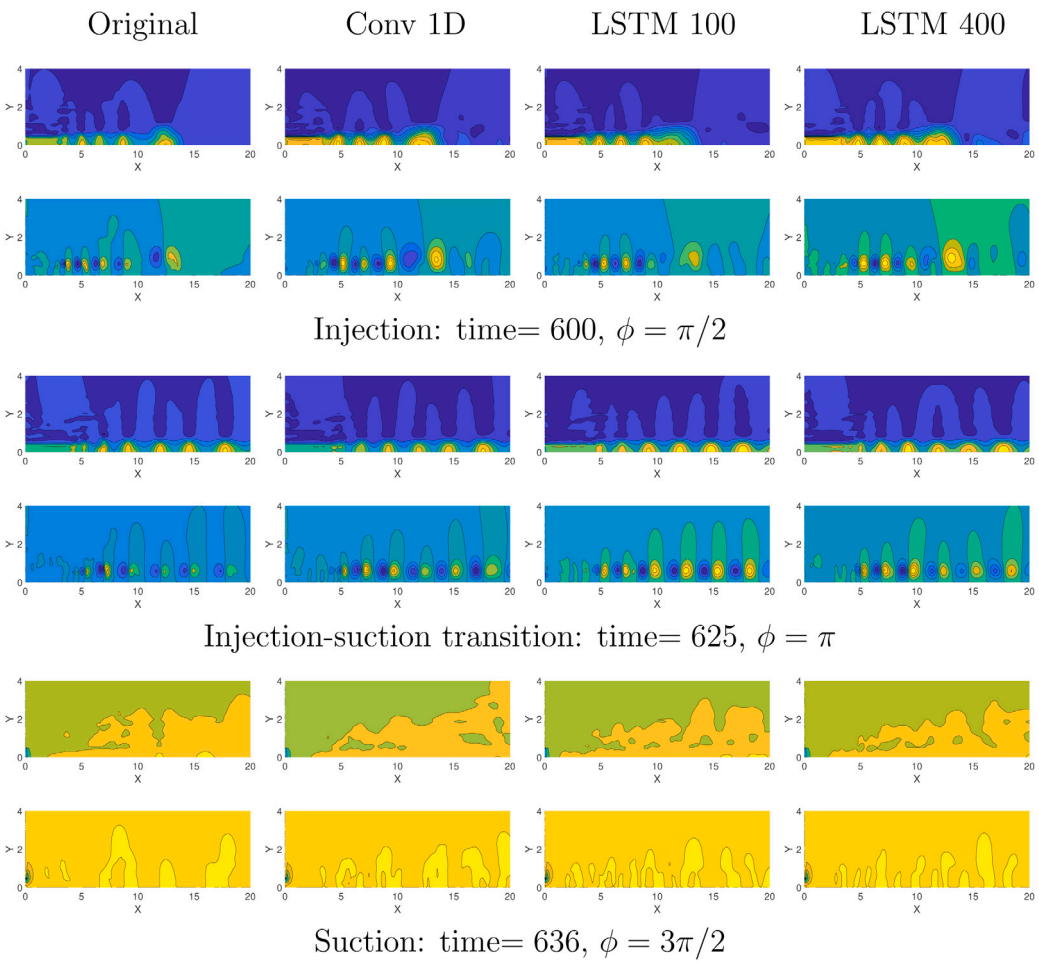

Fig. 19. Same as Fig. 18 for the matrix analysed containing $N = 100$ modes.

Table 5
Same as Table 4 for the synthetic jet in the tests carried out using $N = 20$ and 100 modes.

Architecture	N	Prediction time (t_{pred})	Training time (t_{tr})	Speed-up
Conv 1D		0.31	65.95	347.76
LSTM 100	20	0.79	131.27	174.46
LSTM 400		0.82	141.14	168.28
Conv 1D		0.37	74.91	306.04
LSTM 100	100	0.63	56.24	405.14
LSTM 400		0.95	154.33	148.38

4. Conclusions

This article introduces a new predictive ROM that is based on physical principles. The hybrid ROM combines modal decompositions to reduce the dimensionality of a database, with deep learning architectures to predict the temporal evolution of the dataset studied. The model is applied to study two fluid dynamics problems of interest, the three-dimensional wake of a circular cylinder and a synthetic jet. Nevertheless, the data-driven nature of this ROM and the robustness shown in the results presented, encourage using this model for data forecasting in other research fields, such as language and video processing, finances, robotics, et cetera.

The methodology of this hybrid ROM is summarized in three main steps. Firstly, POD is applied to reduce the data dimensionality from hundred thousands degrees of freedom to a few POD modes (from 20 to 100, depending on the case studied). The original spatio-temporal data are divided into spatial POD modes, which only contain information related to the spatial domain, and temporal modes (or temporal coefficients), which only contain information related to the temporal evolution of the flow. Secondly, a deep learning architecture is applied to the temporal modes to predict more temporal snapshots. Finally, the spatial modes are combined with the new predictions of the temporal modes to reconstruct the original field. Two different deep learning architectures have been used, a one-dimensional convolutional neural network and a recurrent neural network, more specifically, a long-short-term memory (LSTM) scheme using 100 and 400 neurons.

The databases analysed are generated numerically. The training set of the neural networks only uses data from the initial transient stage of the numerical simulations, and they are used to predict the temporally converged solution. The methodology presented is robust and the results obtained are accurate. The speed-up factor in the numerical simulations using this ROM is quite high: the computational time to solve the evolution of the flow in time is reduced from ~1600 h when using the numerical solver, to ~3.5 s (at most, it depends on the type of architecture) when using the hybrid ROM in the three dimensional cylinder wake and from ~192 h to ~155 s (at most, it depends on the type of architecture) in the synthetic jet. These results shed light on the good performance of hybrid ROMs based on physical principles, combining neural networks with modal decompositions.

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.

Acknowledgements

This work was partially supported by the grant RTI2018-098958-B-I00 from Proyectos de I + D + i "Retos investigación", Programa Estatal de I + D + i Orientada a los Retos de la Sociedad, Plan Estatal de Investigación Científica, Técnica y de Innovación 2017–2020. Spanish Ministry for Science, Innovation and Universities; the Agencia Estatal de Investigación (AEI), Spain and the Fondo Europeo de Desarrollo Regional (FEDER). Also, RAH, JMP and SLC acknowledge the support of the grant PID2020-114173RB-I00 from Proyectos I + D + i "Retos de investigación", Plan Estatal de Investigación Científica y Técnica y de Innovación 2017–2020', Spanish Ministry of Science and Innovation.

References

Abadia, S., Lopez-Martin, M., & Le Clainche, S. (2021). Code of the paper: A predictive hybrid reduced order model based on proper orthogonal decomposition combined with deep learning architectures (n.d.). Retrieved july 21, 2021. https://Github.Com/Mlopezm/A-Predictive-Hybrid-ROM-Model-Based-on-Proper-Orthogonal-Decomposition-with-DL-Architectures.

Abadia, S., Pariente, A., Pérez, J., & Le Clainche, S. (2021). Tortuosity in tumors: the need of combining multi-phase flows with machine learning tools. *Res Ing*.

Barckley, D., & Henderson, R. (1996). Three-dimensional floquet stability analysis of the wake of a circular cylinder. *Journal of Fluid Mechanics*, *322*, 215–241.

Blackburn, H. M., & Henderson, R. (1999). A study of two–dimensional flow past an oscillating cylinder. *Journal of Fluid Mechanics*, *385*, 255–286.

Brunton, S. L., & Noack, B. R. (2015). Closed-loop turbulence control: Progress and challenges. *Applied Mechanics Reviews*, *67*, Article 050801.

Brunton, S., Noack, B., & Koumoutsakos, P. (2020). Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, *52*, 477–508.

Cao, W., Hu, L., Gao, J., Wang, X., & Ming, Z. (2020). A study on the relationship between the rank of input data and the performance of random weight neural network. *Neural Computing and Applications*, *16*(32), 278—287.

Cao, W., Wang, X., Ming, Z., & Gao, J. (2018). A review on neural networks with random weights. *Neurocomputing*, (275), 278—287.

Carter, J. E., & Soria, J. (2002). The evolution of round zero-net-mass-flux jets. *Journal of Fluid Mechanics*, *472*, 167–200.

Cattafesta, L., & Sheplak, M. (2010). Actuators for active flow control. *Annual Review of Fluid Mechanics*, *43*, 247–272.

Corrochano, A., Xavier, D., Schlatter, P., Vinuesa, R., & Le Clainche, S. (2021). Flow structures on a planar food and drug administration (FDA) nozzle at low and intermediate Reynolds number. *Fluids*, *6*(1), 4.

DeMont, E., & Gosline, J. (1988). Mechanics of jet propulsion in the hydromedusan jellyfish, polyorchis penicillatus. *The Journal of Experimental Biology*, *134*, 347–361.

Discetti, S., Raiola, M., & Ianiro, A. (2018). Estimation of time-resolved turbulent fields through correlation of non-time-resolved field measurements and timeresolved point measurements. *Experimental Thermal and Fluid Science*, *93*, 119–130.

Ferrari, M., Werner, G., Bahrmann, P., Richartz, B., & Figulla, H. (2006). Turbulent flow as a cause for underestimating coronary flow reserve measured by Doppler guide wire. *Cardiovascular Ultrasound*, *14*(4).

Fischer, P. F., L., J. W., & Kerkemeier, S. G. (2008). nek5000 web page. http://nek5000.mcs.anl.gov.

Freitag, S., Cao, B., Nini', J., & Meschke, G. (2018). Recurrent neural networks and proper orthogonal decomposition with interval data for real-time predictions of mechanised tunnelling processes. *Computers & Structures*, *207*, 258–273.

Glezer, A., & Amitay, M. (2002). Synthetic jets. *Annual Review of Fluid Mechanics*, *34*, 503–529.

Guastoni, L., Güemes, A., Ianiro, A., Discetti, S., Schlatter, P., Azizpour, H., & Vinuesa, R. (2020). Convolutional-network models to predict wall-bounded turbulence from wall quantities. arXiv:2006.12483 [physics.flu-dyn].

Güemes, A., Discetti, S., & Ianiro, A. (2019). Sensing the turbulent large-scale motions with their wall signature. *Physics of Fluids*, *31*, Article 125112.

Guo, M., & Hesthaven, J. (2019). Data-driven reduced order modeling for time-dependent problems. *Computer Methods in Applied Mechanics and Engineering*, *345*, 75–99.

Iuliano, E., & Quagliarella, D. (2013). Proper Orthogonal Decomposition, surrogate modelling and evolutionary optimization in aerodynamic design. *Computers and Fluids*, *84*, 327–350.

Jackson, C. P. (1987). A finite-element study of the onset of vortex shedding in flow past variously shaped bodies. *Journal of Fluid Mechanics*, *182*, 23–45.

Jones, B. L., Heins, P. H., Kerrigan, E. C., Morrison, J. F., & Sharma, A. (2015). Modelling for robust feedback control of fluid flows. *Journal of Fluid Mechanics*, *769*, 687–722.

Kingma, D. P., & Ba, J. L. (2020). Adam: A method for stochastic optimization. Dec. 2015. [Online]. Available: https://Arxiv.Org/Abs/1412.6980v9. (Accessed: 21 Sep. 2020).

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1D convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, *151*, Article 107398.

Lassila, T., Manzoni, A., Quarteroni, A., & Rozza, G. (2014). Model order reduction in fluid dynamics: Challenges and perspectives. In *Reduced order methods for modeling and computational reduction. MS & A - modeling, simulation and applications*. Springer.

Le Clainche, S. (2019). Prediction of the optimal vortex in synthetic jets. *Journal of Fluid Mechanics*, *12*(9), 1635.

Le Clainche, S., Benocci, C., & Parente, A. (2013). Principal component analysis on a LES of a squared ribbed channel. *Intelligent Systems and Computing Springer*, *239*.

Le Clainche, S., & Ferrer, E. (2018). A reduced order model to predict transient flows around straight bladed vertical axis wind turbines. *Energies*, *11*(3), 566.

Le Clainche, S., Lorente, L., & Vega, J. (2018). Wind predictions upstream wind turbines from a LiDAR database. *Energies*, *11*(3), 543.

Le Clainche, S., Pérez, J. M., Soria, J. M., & Vega, J. M. (2020). Near and far field laminar flow structures in an axisymmetric zero-net-mass-flux jet. *Aerospace Science and Technology*, *105*, Article 105920.

Le Clainche, S., Perez, J. M., & Vega, J. M. (2018). Spatio-temporal flow structures in the three- dimensional wake of a circular cylinder. *Fluid Dynamics Research*, *50*, Article 051406.

Le Clainche, S., Pérez, J. M., & Vega, J. M. (2018). Spatio-temporal flow structures in the three-dimensional wake of a circular cylinder. *Fluid Dynamics Research*, *50*(5), Article 051406.

Le Clainche, S., Rosti, M., & Brandt, L. (2021). A data-driven model based on modal decomposition: application to the turbulent channel flow over an anisotropic porous wall. *Journal of Fluid Mechanics* (submitted for publication).

Le Clainche, S., Vega, J. M., & Soria, J. (2017). Higher Order Dynamic Mode Decomposition for noisy experimental data: flow structures on a Zero-Net-Mass-Flux jet. *Experimental Thermal and Fluid Science*, *88*, 336–353.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Lopez-Martin, M., Le Clainche, S., & Carro, B. (2021). Model-free short-term fluid dynamics estimator with a deep 3D-convolutional neural network. *Expert Systems with Applications*, *117*, Article 114924.

Luchtenburg, D., Noack, B., & Schlegel, M. (2009). *Galerkin method for fluid flows with analytical examples and MATLAB source codes*: *Report 01*, Berlin Institute of Technology.

Lumley, J. L. (1967). The structure of inhomogeneous turbulent flows. In A. M. Yaglam, & V. I. Tatarsky (Eds.), *Proceedings of the international colloquium on the fine scale structure of the atmosphere and its influence on radio wave propagation*. Nauka, Moscow: Doklady Akademii Nauk SSSR.

Marusic, I., Candler, G., Interrante, V., Subbareddy, P., & Moss, A. (2003). Real time feature extraction for the analysis of turbulent flows. *Semantic Scholar*.

Mendez, M. A., Balabane, M., & M., B. J. (2019). Multi-scale proper orthogonal decomposition of complex fluid flows. *Journal of Fluid Mechanics*, *870*, 988–1036.

Muijres, F., Lentink, D., van Oudheusden, B., & Van Leeuwen, J. (2005). Visualization of vortex wake interactions of a flapping foil. *Proceedings of PSFVIP-5*.

Noack, B. R., Morzynski, M., & Tadmor, G. (2011). *Reduced-order modelling for flow control*. New York: Springer.

Omata, N., & Shirayama, S. (2019). A novel method of low-dimensional representation for temporal behavior of flow fields using deep autoencoder. *AIP Advances*, *9*, Article 015006.

Pavlova, A., & Amitay, M. (2006). Electronic cooling using synthetic jet impingement. *Journal of Heat Transfer*, *128*(9), 897–907.

Quarteroni, A., Manzoni, A., & Negri, F. (2016). *Reduced basis methods for partial differential equations*. Springer.

Schmid, P. (2010). Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, *656*, 5–28.

Sharma, A. (2011). Model reduction of turbulent fluid flows using the supply rate. *International Journal of Bifurcation and Chaos*, *19*, 1267–1278.

Sirovich, L. (1987). Turbulence and the dynamics of coherent structures. Parts I–III. *Quarterly of Applied Mathematics*, *45*(3), 561–571.

Wang, X., & Cao, W. (2018). Non-iterative approaches in training feed-forward neural networks and their applications. *Soft Computing*, *11*(22), 3473—3476.

Wang, H., & Menon, S. (2001). Fuel-air mixing enhancement by synthetic microjets. *American Institute of Aeronautics and Astronautics*, *39*, 2308–2319.

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: Lstm cells and network architectures. *Neural Computation*, *31*(7), 1235—1270.