Hybrid deep neural network based prediction method for unsteady flows with moving boundary

Ren-Kun Han· Zhong Zhang· Yi-Xing Wang· Zi-Yang Liu· Yang Zhang· Gang Chen

Received: 29 April 2021 / Accepted: 28 June 2021 / Published online: 13 September 2021
© The Chinese Society of Theoretical and Applied Mechanics and Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract
Any motion, forced or free, of boundary affects the flow field around this boundary. A new kind of reduced order model (ROM) based on hybrid deep neural network is proposed to model flow field evolution process of unsteady flow around moving boundary. This hybrid deep neural network can map the relationship between the flow field at the next time step and the flow field and boundary positions at the previous time steps. Based on the learned information, the hybrid deep neural network can quickly and accurately predict the flow field. Unsteady flows around forced oscillation cylinder with various amplitudes, frequencies, and Reynolds numbers are simulated to establish the training and testing datasets. The prediction results of the hybrid deep neural network and the computational fluid dynamics (CFD) simulation results are consistent with high accuracy. The forces on the moving boundary can be integrated through the predicted flow field data. Good performance makes this new ROM method can be used in many fluid dynamics research fields, which needs fast and accurate simulation.

Keywords Deep neural network · Flow fields prediction · Moving boundary

1 Introduction
In many scientific and engineering problems, flow systems with moving boundaries need to be simulated [1]. The high-fidelity computational fluid dynamics (CFD) techniques are able to accurately simulate this process. Depending on billions of degrees of freedom and dynamic mesh, the computational cost is unbearably high, which is unacceptable in many engineering applications. Therefore, many researchers try to use data driven methods to accurately simulate high dimensional systems with less computational cost [2].

Reduced order model (ROM) [3, 4] greatly improves the calculation efficiency and shortens the calculation time by projecting the original system to several low-dimensional mathematical models, which contain most of the energy of the flow field. A lot of researches have proved the high efficiency of ROM [5]. But most of the ROMs are linear or weakly nonlinear methods. In contrast, most flows in the real world are non-linear, so a ROM that can solve nonlinear problems is urgently needed. Deep neural network [6, 7] is good at finding hidden information from nonlinear systems. Recently, many works using the deep neural network to solve fluid mechanics problems have shown its capability of modeling nonlinear fluid dynamic system.

The goal of building ROMs of nonlinear unsteady flows by using deep neural network requires deep neural network can reduce dimensionality and reconstruct it to high-dimensional flow field. The deep convolutional autoencoder, including an encoding neural network and a decoding neural network, was used for dimensionality reduction of unsteady flow fields [8]. The encoding neural network is a convolutional neural network (CNN), which can reduce dimensionality and capture spatial features. The decoding neural network is a deconvolutional neural network (DeCNN), which can reconstruct low-dimensional features to high-dimensional flow field.

Then researchers tried to use deep neural networks to capture the temporal and spatial features simultaneously. Some
researchers used CNN to capture spatial features and combined the captured features of previous time steps obtained by CNN with DeCNN to reconstruct flow fields [9–11]. The combinations of CNN and convolutional long short-term memory (ConvLSTM) networks also were proposed to model non-linear flow dynamics systems [18, 19]. In those networks, the ConvLSTM was used to predict the spatial features at next time step and DeCNN was used to reconstruct features to high-dimensional flow field. Pawar et al. [12] constructed a method combined of proper orthogonal decomposition (POD) and fully connected neural network. In this method, POD was used to extract the main modes of the flow field. The time coefficients of POD modes were predicted by fully connected neural network. High-dimensional flow field information can be obtained through the combination of modal coefficients and modes. The time coefficients of POD modes also could be predicted by other kind neural networks, like long short-term memory networks (LSTM) [13–16] and CNN [17].

However, all those methods can only predict unsteady flows with no moving boundaries. For a fluid–solid interaction (FSI) system, it is difficult to capture the complex temporal and spatial features of the unsteady flow influenced by moving boundary. Most of the existing researches directly use neural networks to predict force coefficients, while ignoring the evolution of the flow field. Srivastava et al. [20] used recurrent neural networks (RNN) to predicting forces at the next moment based on boundary position at this moment of transonic pitching and plunging wing-fuel tank sloshing system. In order to achieve the goal of using a deep neural network to solve the FSI problem while retaining more information in the system, this paper tries to use the deep neural network to model the evolution of the unsteady flow field with moving boundaries.

Based on previous work [19], a ROM is designed to predict unsteady flow with moving boundary. This ROM is an improved hybrid deep neural network, which is able to capture the temporal and spatial flow dynamics features and predict flow field with low computational cost. Different from previous neural network, the internal structure of the LSTM cell and the way used to input data into LSTM layer are modified so that this hybrid deep neural network can simultaneously learn the temporal evolution characteristics and the influence of boundary position changing.

2 Hybrid deep neural network

2.1 Architecture of the hybrid deep neural network

In CFD simulation, we can calculate the flow field of the next time step by the flow field information of the previous time steps. In the same way, we can use deep neural network to predict the flow field of the next time step with the help of the following relationship

\[ F_i = f(F_{i-1}, F_{i-2}, \ldots, F_{i-k}, G_i), \]  

where \( F_i \) and \( G_i \) represent flow field and grid position at time \( i \), \( k \) indicates how many previous time steps are used to predict the flow field at the next time step. The goal of this work is to model unsteady flow system with moving boundaries. Therefore, boundary position information should be considered into the deep neural network. Based on this, a new architecture of the CNN-LSTM network is proposed, as shown in Fig. 1. This network is a hybrid of CNN, LSTM, and DeCNN. In this network, the CNN is used to extract spatial features from flow field and boundary position information and the ConvLSTM is used to predict the spatial features at next time step. After training the deep neural network, it could be used to predict the unsteady flow, the output of the trained network is recycled as the input of next prediction step to predict flow fields at following steps. So that the goal of predicting the flow fields continuously is achieved. This neural network is able to predict long-time flow fields evolution process only once input.

When predicting flow field with moving boundary, the typical LSTM cell needs to be improved to ConvLSTM cell so that it can simultaneously learn the temporal evolution and the boundary position changing information. The data flow path also needs to be changed to facilitate the introduction of boundary position information. The ConvLSTM layers in this paper can model temporal features evolution of flow fields and the influence of boundary position changing, then predict the spatial features at next time step.

2.2 Improved ConvLSTM cell

For typical LSTM cells, there are three gates: input gate (adding useful information to cell state), forget gate (removing useless information from cell state), and output gate (outputting information to next cell), as shown in Fig. 2.
The inputs are all one-dimensional vectors. Therefore, in order to learn high-dimensional data, it is necessary to flatten high-dimensional inputs to one-dimensional data. In this case, the spatial features of the high-dimensional data will be destroyed. To keep these spatial features, turning point multiplication between vectors into normal convolution operation between tensors is a very useful technique. New convolutional LSTM cell is able to capture spatial features from multidimensional data. It enables that all features of one field obtained by CNN layers can be set as input data for the new ConvLSTM cell. The equations described the improved ConvLSTM cell are defined as

\[
\begin{align*}
  f_t &= \sigma(W_{xf} \odot \text{c}_t + W_{hf} \odot \text{h}_{t-1} + W_{cf} \odot \text{c}_{t-1} + b_f), \\
  i_t &= \sigma(W_{xi} \odot \text{x}_t + W_{hi} \odot \text{h}_{t-1} + W_{ci} \odot \text{c}_{t-1} + b_i), \\
  c_t &= f_t \odot \text{c}_{t-1} + i_t \odot \tanh(W_{xc} \odot \text{x}_t + W_{hc} \odot \text{h}_{t-1} + b_c), \\
  o_t &= \sigma(W_{xo} \odot \text{x}_t + W_{ho} \odot \text{h}_{t-1} + W_{co} \odot \text{c}_{t-1} + b_o), \\
  \text{h}_t &= o_t \odot \tanh(c_t),
\end{align*}
\]

where subscripts \(t\) and \(t-1\) represent the time step of unsteady flow fields. \(i\), \(o\), and \(f\) respectively represents input gate, output gate, and forget gate. \(c\) and \(h\) respectively represents cell input, cell state, and cell output. The weights for each of the gates are represented as \(W\). The biases are represented as \(b\). \(\sigma\) stands for sigmod activation function. \(\cdot\) and \(\odot\) respectively represents operation of dot multiplication and convolution.

In addition, how to input different kinds of data into LSTM layer also needs to be redesigned. Therefore, the data flow path of LSTM layer in this paper is different from that of traditional LSTM layer. For the problem of predicting flow fields with moving boundary, features obtained by convolutional layers from boundary position information needed to be predicted are set as the cell state \(c_{t-1}\) in the first ConvLSTM cell of ConvLSTM layer. Features obtained by convolutional layers from flow fields at previous time steps are set as the \(x_t\) in the ConvLSTM cell. Multiple ConvLSTM cells are connected in series to form a ConvLSTM layer, which means that flow fields in more time steps are added into this hybrid deep neural network as \(x_{t+1}\) in following cells. This process is like painting. The boundary position information is taken as the background color, and the flow features are continuously added on it.

### 2.3 Training method

The goal of predicting flow fields at future time step is considered as an image-to-image regression task. The root mean square error (RMSE) is adopted as the loss function evaluating the prediction accuracy, i.e.

\[
RMSE = \sqrt{\frac{\sum_{j=1}^{N} (\psi_{ij} - \psi_{o,j})^2}{N}},
\]

where \(\psi_{ij}\) and \(\psi_{o,j}\) represent the prediction and ground truth respectively. \(i\) and \(t\) represent position and time step. \(N\) is node number of one flow field data. Since max error always located at the region near to the moving boundary, the weight factors of the errors in the area of twice the diameter are amplified. The network is trained by minimizing the RMSE in Eq. (3) equaling to optimize kernel parameters.

As for how to choose parameters of the architecture of the neural network will be detailed in the last section. Once the neural network structure is determined. The neural networks are trained in TensorFlow. With the iteration of data forward propagation and loss back propagation in the network, this network gets closer to the real system.

### 3 Dataset preparation

#### 3.1 Dataset structure

The datasets used in this paper are obtained by numerical simulation by solving equations as follow

\[
\nabla \cdot \text{u} = 0, \\
\frac{\partial \text{u}}{\partial t} + \nabla \cdot \text{uu} = -\nabla p + \frac{1}{Re} \nabla^2 \text{u},
\]

where \(t\), \(p\), and \(u\) represent time, pressure, and velocity non-dimensionalized by incoming velocity.

Because CNN can only process data on a uniformly distributed grid, the dataset used for training the neural network should be uniformly distributed data, like an image. Therefore, flow fields data should be distributed onto a uniformly distributed grid. Simulation data in a rectangular area are cut out as the training dataset. \(N_x \times N_y\) points are evenly distributed grid. Simulation data in a rectangular area are cut out as the training dataset.
distributed in this rectangular region. Then, the nondimensionalized flow field variables are projected onto these uniformly distributed grids. The values of points inside moving boundary are set as 0 to distinguish fluid domain from solid domain. Flow field variables \( \{p^*, u^*, v^*\} \) and grid position information \( G_i \) are extracted at distributed points. Flow data at each instantaneous field are projected to a \( N_x \times N_y \times 4 \) matrix data. Then the obtained data is arranged in chronological order to keep the time correlation between the data before and after.

### 3.2 Cases conditions and dataset generation

Flow structure around the wake of a forced vibrating cylinder is a common engineering problem. In this article, three cases of unsteady flow around a forced vibrating cylinder are used to demonstrate the predictive power of this hybrid deep neural network. The simulation domain layout is shown as the black box in Fig. 3. The equation of cylinder vibration is defined as

\[
Y = A \sin (2\pi ft). \tag{5}
\]

The training sets are the flow around moving cylinder cases with different vibration amplitudes, frequency, and Reynolds number. The first test case (case 1) is about flow around moving cylinder cases with amplitudes \( A = 0.25D, 0.3D, 0.35D, \) and \( 0.4D \). \( D \) represents the diameter of the cylinder. The second test case (case 2) is about flow around moving cylinder cases with different vibration frequency. When the cylinder is not vibrating, the natural frequency of the flow field is 1.33 Hz at \( Re = 100 \). The frequency ratio, which is the ratio of the cylinder vibration frequency to the natural frequency of the flow field, is used to express different states. The cylinder vibration frequency used in this case are \( f = 1.3 \) Hz, 1.35 Hz, 1.4 Hz, and 1.45 Hz, which are converted into frequency ratios of \( r = 0.98, 1.01, 1.05, \) and 1.09. The third test case (case 3) is about flow around moving cylinder cases with \( Re = 50, 100, 150, \) and 200. The first three cases are only for the modeling of single parameter change system. The fourth test case (case 4) is about the modeling of multiple variables system. The flow around moving cylinder cases with frequency ratios from 0.98 to 1.09 and amplitudes from 0.25D to 0.4D are employed as the training dataset. The variable interval is consistent with case 1 and case 2. We expect that the trained deep neural network can output an accurate prediction to a new input condition even though that is beyond the range of the training datasets. The case of flow around oscillation cylinder with amplitude \( A = 0.45D \) at the same Reynolds numbers and vibration frequency with training data is set as testing condition in case 1. Similarly, the case of \( r = 1.13 \) and \( Re = 250 \) are separately set as testing condition in case 2 and case 3.

The computational domain is shown in Fig. 3. The wake flow area inside the dotted line \((-1.75D \leq x \leq 8.25D, -5D \leq y \leq 5D)\) is chosen as the learning area. \( 200 \times 200 \) uniformly distributed grid points are placed in this area. Then, the nondimensionalized flow field variables and distance between boundary position and CFD mesh grid are projected onto these grid points. CFD mesh grid distance matrices \( G_i \) have the same shape with input data fields. Every data in the matrix means the distance (\( d \)) between this grid position and boundary surface, which is calculated by

\[
d = \sqrt{(x - x_0)^2 + (y - y_0)^2} - D/2. \tag{6}
\]

Three flow variables \( \{p^*, u^*, v^*\} \) and mesh grid distance \( G \) are extracted at each time step.

### 4 Results and discussions

#### 4.1 Parameters chosen of neural network architecture

The main contribution of this paper is the modification of the LSTM layer of HAN’s neural network structure [19]. Therefore, the hyperparameters of CNN and DeCNN layers in the new neural network still use the hyperparameters in the previous network. In this section, only the influence of the hyperparameter value of the LSTM layer on the results prediction is discussed. The hyperparameters of the basic neural network structure used in this article are shown in Table 1. The size in this table represents the size of the matrix. In the proposed hybrid deep neural network, most influential structure parameters mainly include the number of ConvLSTM cells per layer and the number of ConvLSTM layers. Based on the basic neural network structure, the number of ConvLSTM cells per layer and the number of ConvLSTM layers are changed to test their influence on prediction accuracy.
Figure 4a shows how the average RMSE in 150-time steps varies with different number of ConvLSTM cells when there is only one ConvLSTM layer. It can be seen that the number of ConvLSTM cells has a greater impact on the accuracy of the network used to predict results. And the average RMSE is the smallest when the number of cells is 3. It is the best to use flow fields in previous 3-time steps to predict flow field at next step. This means more time step input cannot make flow field prediction at the next moment more accurate. The flow field farther away from the moment to be predicted has less influence on the flow field at the moment to be predicted.

Figure 4b shows how the average RMSE in 150-time steps varies with different number of ConvLSTM layers. The prediction error increases with the number of ConvLSTM layers increases. Increasing the number of ConvLSTM layer except to capture deeper information. But the structure of the testing flow field around a moving cylinder is relatively simple. And more layers increase the neural network variables, which may lead to greater errors. Therefore, for this case, one LSTM layer is enough to capture the temporal and spatial features of unsteady flows around a forced oscillation cylinder. And for the problem with more complicated flow field characteristics, perhaps more ConvLSTM layers are needed to capture the features more accurately.

4.2 Time series prediction with different amplitudes

In this section, the flow around moving cylinder cases with amplitudes $A = 0.25D, 0.3D, 0.35D$, and $0.4D$, $r = 1.0$ at $Re = 100$ are employed as the training dataset, while the condition of $A = 0.45D$ is used for testing. Figure 5 shows the RMSE during the training. After 600,000 iterations of forward propagation and back propagation, the training RMSE loss converges to 0.006.

The average RMSE of the predicted flow fields in 150-time steps is less than one percent. Streamwise velocity of instantaneous fields in 15 time-steps are shown in Fig. 6. It should be clarified that all predictions after $3dt$ are predicted based on the previous prediction data. From comparisons, flow fields predicted by hybrid deep neural network agree well with CFD simulation results. One characteristic
position, shown as the black star in Fig. 3, is selected to show the continuous prediction ability and how many steps does the hybrid deep neural network need to make an accurate prediction. Time series of nondimensionalized velocity and pressure at selected position, network predictions and CFD simulation results are compared in Fig. 7. The results show that this hybrid deep neural network is able to continuously and accurately predict the new condition flow field. The direct integration method is used to calculate the drag coefficient, as shown in the equation below

\[
C_d = \frac{\int_{\text{body}} \left(-p n_x + \tau_{xx} n_x + \tau_{xy} n_y\right) dL}{\frac{1}{2} \rho V^2 l},
\]

where \( p, \tau_{xx}, \tau_{xy} \) represent surface pressure, normal stress and tangential stress on the integral derivative, \( \rho \) represents the density, \( V \) represents the incoming velocity, \( l \) represents the characteristic length and \( n_x, n_y \) represent the \( x \)-axis and \( y \)-axis coordinate vectors.

Time series comparisons of the drag coefficient between network prediction and CFD simulation result are shown in Fig. 8. This hybrid deep neural network is able to accurately predict the flow field both on the surface and in the wake region in a long period of time. The results also show that the hybrid deep neural network has good extrapolatory capability and prediction error does not increase with time increase. To get flow fields in 150-time steps, it only cost 17 s using trained hybrid deep neural network, but 240 s by CFD simulation solver.

### 4.3 Time series prediction with different vibration frequency

In this part, the flow around moving cylinder cases with frequency ratios \( r = 0.98, 1.01, 1.05, \) and 1.09, \( A = 0.3D \) at \( Re = 100 \) are employed as the training dataset, while the flow with frequency ratio \( r = 1.13 \) is used for testing. The structure parameters of hybrid deep neural network are same as case 1. After 600,000 iterations of forward propagation and back propagation, the training RMSE loss converges to 0.005.

The average RMSE of the predicted flow fields in 150-time steps is less than two percent. Time series of predicted nondimensionalized velocity and pressure at selected position are shown in Fig. 9. Comparison of the drag coefficients obtained by integrating the surface pressure is shown in Fig. 10. This means that this hybrid deep neural network is able to give an accurate prediction of pressure on the surface. The trained hybrid deep neural network is able to predict in several cycles. And the error increase is not obvious in the long-term predicting process. Because CNN is a kind of neural network which can extract the main features from global data, some small local errors will be ignored so that there will be no error accumulation. It means that the hybrid deep neural network has captured the influence of the cylinder moving frequency change on the flow field structure change.

### 4.4 Time series prediction with different Reynolds numbers

The impact of Reynolds number changes on the flow field is greater than the impact of amplitude and frequency changes. The flow field structure changes greatly under different Reynolds numbers. Flow around moving cylinder cases at Reynolds number \( Re = 50, 100, 150, \) and 200 are employed as the training dataset, while the flow at \( Re = 250 \) is used for testing. The hybrid deep neural network structure parameters are same as case 1. After 600,000 iterations of forward propagation and back propagation, the training RMSE loss converges to 0.02.

The average RMSE of the predicted flow fields in 150-time steps is less than five percent. Time series of nondimensionalized velocity and pressure at selected position, network predictions and CFD simulation results are compared in Fig. 11. Comparison of the drag coefficient obtained by integrating the surface pressure is shown in Fig. 12. From comparisons, the predictions in this case are not good enough as the previous two cases. The structure characteristics of the flow field between the prediction condition and the training condition are quite different, resulting in larger prediction errors than previous two cases. But the overall change tendency of the predictions is well consistent with the real value. The law of flow field change under different Reynolds numbers is still learned. In the variation space of a certain parameter, the evolution process of unsteady flow field can be accurately predicted under different Reynolds numbers.

### 4.5 Time series prediction with multiple variables

In this section, we try to use neural networks to model multiple variables at the same time. The flow around moving cylinder cases with different frequency ratios from 0.98 to 1.09 and amplitude from 0.25D to 0.4D at \( Re = 100 \) are employed as the training dataset. The variable interval is consistent with case 1 and case 2. After 600,000 iterations of forward propagation and back propagation, the training RMSE loss converges to 0.005.
Comparison of the drag coefficient obtained by integrating the surface pressure is shown in Fig. 13. This figure also shows the prediction results of case 1. It can be seen from the figure that the deep neural network trained with multivariate data has better predictive ability. It agrees better with the CFD data at the peak. Figure 14 shows the comparisons of direct integration drag coefficient at \( Re = 100 \), \( r = 1.13 \), and \( h = 0.45D \). This condition exceeds the range of training data in both frequency and amplitude. From the results, it can be concluded that using multivariate data to train a deep neural network can not only improve the accuracy of univariate state prediction, but also expand the boundary of the prediction range.

5 Conclusions

In order to adapt to the problem of flow field prediction with moving boundary, the ConvLSTM cell of CNNLSTM network was modified making it able to learn temporal and spatial features from flow fields and boundary position change. This new type of hybrid deep neural network is
able to capture the influence of boundary vibration on the flow field. The hybrid deep neural network is used to model the flow around a forced oscillation cylinder with different amplitudes and frequencies. The unsteady flow fields around moving boundaries at different conditions can be easily predicted by trained hybrid deep neural network. Good performance makes this new ROM method can be used in many fluid dynamic research fields that require fast calculations, like flow control and aerodynamic optimization.

This new ROM is expected be able to deal with the problems of fluid structure interactions and flow control for moving boundary system. In the next step, we will focus on how to predict the surface pressure through the wake field information by a deep neural network, so as to obtain more accurate lift force and drag force. Finally, we are going to realize the fast calculation of the end-to-end fluid–structure interactions problem by deep neural network completely.

Fig. 10 Comparisons of direct integration drag coefficients between network predictions and CFD simulation results at Re = 100, r = 1.13, and h = 0.3D

Fig. 12 Comparisons of direct integration drag coefficients between network predictions and CFD simulation results at Re = 250, r = 1.13, and h = 0.3D

Fig. 13 Comparisons of direct integration drag coefficients between network predictions and CFD simulation results at Re = 100, r = 1.0, and h = 0.45D

Fig. 11 Comparisons of ρ*, u*, and v* at selected position between network predictions and CFD simulation results at Re = 250, r = 1.13, and h = 0.3D
Acknowledgements This work was supported by the National Natural Science Foundation of China (Grants 11872293, 11672225), Science and Technology on Reliability and Environment Engineering Laboratory (Grant 6142004190307), and the Program of Introducing Talents and Innovation of Discipline (Grant B18040).

References

1. Li, G., Liu, N., Lu, X.: Dynamic performance and wake structure of flapping plates with different shapes. Acta Mech. Sin. 30(6), 800–808 (2016)
2. Suzuki, T., Chatellier, L., David, L.: A few techniques to improve data-driven reduced-order simulations for unsteady flows. Comput. Fluids 201, 0045–7930 (2020). https://doi.org/10.1016/j.compfluid.2020.104455
3. Chen, G., Li, D., Zhou, Q., et al.: Efficient aeroelastic reduced order model with global structural modifications. Aerosp. Sci. Technol. 76, 1–13 (2018). https://doi.org/10.1016/j.ast.2018.01.023
4. Schmid, P.J.: Dynamic mode decomposition of numerical experimental data. J. Fluid Mech. 656(10), 5–28 (2010). https://doi.org/10.1017/S0022112010001217
5. Chen, G., Li, Y.: Advances and prospects of the reduced order model for unsteady flow and its application. Adv. Mech. 41(06), 686–701 (2011)
6. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521, 436–444 (2015). https://doi.org/10.1038/nature14539
7. Xie, C., Yuan, Z., Wang, J., et al.: Artificial neural network-based subgrid-scale models for large-eddy simulation of turbulence. Chin. J. Theor. Appl. Mech. 53(1), 1–16 (2021)
8. Omata, N., Shirayama, S.: A novel method of low dimensional representation for temporal behavior of flow fields using deep autoencoder. AIP Adv. 9, 015006 (2019). https://doi.org/10.1063/1.5067313
9. Fukami, K., Nabae, Y., Kawai, K., et al.: Synthetic turbulent inflow generator using machine learning. Phys. Rev. Fluids 4, 064603 (2019). https://doi.org/10.1103/PhysRevFluids.4.064603
10. Lee, S., You, D.: Data-driven prediction of unsteady flow over a circular cylinder using deep learning. J. Fluid Mech. 879, 217–254 (2019). https://doi.org/10.1017/jfm.2019.700
11. Agostini, L.: Exploration and prediction of fluid dynamical systems using auto-encoder technology. Phys. Fluids 32, 067103 (2020). https://doi.org/10.1063/5.0012906
12. Pawar, S., Rahman, S.M., Vaddireddy, H., et al.: A deep learning enabler for nonintrusive reduced order modeling of fluid flows featured. Phys. Fluids 31, 085101 (2019). https://doi.org/10.1063/1.5113494
13. Rahman, S.M., Pawar, S., San, O., et al.: Nonintrusive reduced order modeling framework for quasigeostrophic turbulence. Phys. Rev. E 100, 053306 (2019). https://doi.org/10.1103/PhysRevE.100.053306
14. Deng, Z., Chen, Y., Liu, Y., et al.: Time-resolved turbulent velocity field reconstruction using a long short-term memory (LSTM)-based artificial intelligence framework. Phys. Fluids 31, 075108 (2019). https://doi.org/10.1063/1.5111558
15. Ahmed, S.E., Rahman, S.M., San, O., et al.: Memory embedded non-intrusive reduced order modeling of non-ergodic flows. Phys. Fluids 31, 126602 (2019). https://doi.org/10.1063/1.5128374
16. Pawar, S., Ahmed, S.E., San, O., et al.: Long short-term memory embedded nudging schemes for nonlinear data assimilation of geophysical flows. Phys. Fluids 32, 076606 (2020). https://doi.org/10.1063/5.0012853
17. Miyanawala, T.P., Jaiman, R.K.: A Hybrid Data-driven Deep Learning Technique for Fluid-Structure Interaction. arXiv preprint, arXiv:1807.09591v2 (2019).
18. Hasegawa, K., Fukami, K., Murata, T., et al.: Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes. Theoret. Comput. Fluid Dyn. 34, 367–383 (2020). https://doi.org/10.1007/s00162-020-00528-w
19. Han, R., Wang, Y., Zhang, Y., et al.: A novel spatial-temporal prediction method for unsteady wake flows based on hybrid deep neural network. Phys. Fluids 31, 127101 (2019). https://doi.org/10.1063/1.5127247
20. Srivastava, S., Damodaran, M., Damodaran, M., et al.: Machine Learning Surrogates for Predicting Response of an Aero- Structural-Sloshing System. arXiv preprint, arXiv:1911.10043. (2019)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.