



# Physics-constrained deep reinforcement learning for flow field denoising

Mustafa Z. Yousif<sup>1,2</sup>, Meng Zhang<sup>1</sup>, Linqi Yu<sup>1</sup>, Yifan Yang<sup>1</sup>, Haifeng Zhou<sup>1</sup> and HeeChang Lim<sup>1,†</sup>

<sup>1</sup>School of Mechanical Engineering, Pusan National University, 2, Busandaehak-ro 63beon-gil, Geumjeong-gu, Busan 46241, Republic of Korea

<sup>2</sup>LSTME-Busan Branch, German Engineering Research and Development Center, 1276 Jisa-dong, Gangseo-gu, Busan, 46742, Republic of Korea

(Received 22 February 2023; revised 8 September 2023; accepted 13 September 2023)

---

A multi-agent deep reinforcement learning (DRL)-based model is presented in this study to reconstruct flow fields from noisy data. A combination of reinforcement learning with pixel-wise rewards, physical constraints represented by the momentum equation and the pressure Poisson equation, and the known boundary conditions is used to build a physics-constrained deep reinforcement learning (PCDRL) model that can be trained without the target training data. In the PCDRL model, each agent corresponds to a point in the flow field and learns an optimal strategy for choosing pre-defined actions. The proposed model is efficient considering the visualisation of the action map and the interpretation of the model operation. The performance of the model is tested by using direct numerical simulation-based synthetic noisy data and experimental data obtained by particle image velocimetry. Qualitative and quantitative results show that the model can reconstruct the flow fields and reproduce the statistics and the spectral content with commendable accuracy. Furthermore, the dominant coherent structures of the flow fields can be recovered by the flow fields obtained from the model when they are analysed using proper orthogonal decomposition and dynamic mode decomposition. This study demonstrates that the combination of DRL-based models and the known physics of the flow fields can potentially help solve complex flow reconstruction problems, which can result in a remarkable reduction in the experimental and computational costs.

**Key words:** computational methods, Navier–Stokes equations, machine learning

---

† Email address for correspondence: [hclim@pusan.ac.kr](mailto:hclim@pusan.ac.kr)

## 1. Introduction

The understanding of fluid flows plays a crucial role in life (for instance, in medicine, construction, transportation, aerospace and astronomy). However, fluid flow problems are usually complex with highly nonlinear behaviour, especially turbulent flows, which occur at generally high Reynolds numbers. In most cases, data from experiments and simulations are used to understand and describe the behaviour of fluids with various accuracy levels that are related to the experimental and numerical set-ups. Numerous methods have been developed to improve the accuracy and practicality of the obtained flow fields. However, several limitations still exist. One of the most notable limitations of the experimental approach is the noise of the obtained flow fields due to the experimental set-up, so that obtaining measurements with an acceptable signal-to-noise ratio is practically impossible in some cases. Therefore, several methods for the reconstruction of flow fields have been introduced. Methods based on linear data-driven approaches, such as proper orthogonal decomposition (POD) (Lumley 1967) and dynamic mode decomposition (DMD) (Schmid 2010), have shown their capability to enhance the resolution of the flow data and filter noisy flow data (Gunes & Rist 2007; He & Liu 2017; Fathi *et al.* 2018; Nonomura, Shibata & Takaki 2019; Scherl *et al.* 2020). Additionally, various denoising methods for particle image velocimetry (PIV) measurements, such as convolution filters, wavelet methods and Wiener filters, have had various levels of success (Vétel, Garon & Pelletier 2011). All the aforementioned methods showed limited success in terms of denoising flow fields because they are based on linear mapping or handcrafted filtering processes, which are mostly incapable of dealing with highly nonlinear fluid problems (Brunton, Noack & Koumoutsakos 2020).

With the recent rapid development in machine learning (ML) and graphic processing units, new data-driven methods have been introduced to provide efficient solutions for problems in various fields, such as image processing, natural language processing, robotics and weather forecasting. Several ML algorithms have been recently used to address problems in fluid dynamics and have shown promising results (Duraismy, Iaccarino & Xiao 2019; Brunton *et al.* 2020; Vinuesa & Brunton 2022). In contrast to linear methods, ML-based techniques can deal with complex nonlinear problems. This feature has paved the way to exploring the feasibility of applying ML to various problems in complex turbulent flows (Guastoni *et al.* 2021; Yousif *et al.* 2023*b*). Several supervised and unsupervised ML-based methods have been proposed for flow reconstruction from spatially limited or corrupted data (Discetti & Liu 2022). Recently, promising results have been reported from using deep learning (DL) by applying end-to-end trained convolutional neural network (CNN)-based models (Fukami, Fukagata & Taira 2019; Liu *et al.* 2020) and generative adversarial network (GAN)-based models (Kim *et al.* 2021; Yu *et al.* 2022; Yousif *et al.* 2023*a*), where deep learning is a subset of machine learning, in which neural networks with multiple layers are used in the model (LeCun, Bengio & Hinton 2015). The GAN-based models have shown better performance than the traditional CNN-based models. Nonetheless, a drawback of such methods lies in the need for the target (high-resolution or uncorrupted) flow data to train the model, which are difficult or impossible to obtain in most cases. Therefore, attempts have been recently made to address this issue under certain conditions, for instance in the case of super-resolution reconstruction of randomly seeded flow fields (Güemes, Vila & Discetti 2022) or applying physical constraints in the loss function of the model to reconstruct high-resolution steady flows from low-resolution noisy data (Gao, Sun & Wang 2021). However, insufficient explainability and interpretability are the main concerns of using ML-based methods, for which no concrete explanation nor control of the model performance is available.

Alternatively, reinforcement learning (RL), which is an ML method where an agent learns to make decisions by interacting with an environment, has shown remarkable results in areas such as robotics, game playing and optimisation problems (Hickling *et al.* 2022). In RL, the agent takes actions and receives feedback in the form of rewards or penalties. Over time, it aims to learn the optimal actions to maximise cumulative rewards and achieve its objectives through trial and error. This approach to learning makes deep reinforcement learning (DRL) a good candidate method to apply to several problems in fluid dynamics, such as flow control (Rabault *et al.* 2019), design optimisation (Viquerat *et al.* 2021), computational fluid dynamics (Novati, de Laroussilhe & Koumoutsakos 2021) and others (Garnier *et al.* 2021; Viquerat *et al.* 2022).

This paper presents a DRL-based approach that can be used for reconstructing flow fields from noisy data. The main advantages of the presented model lie in overcoming the need for the target data in the training process and the explainable filtering process of the noisy data.

The remainder of this paper is organised as follows. Section 2 explains the method of reconstruction of denoised flow fields using the proposed DRL model. Section 3 describes the generation and preprocessing of the data used for training and testing the model. Section 4 discusses the results of testing the proposed model. Finally, the conclusions of this study are presented in § 5.

## 2. Methodology

In contrast to supervised and unsupervised learning, reinforcement learning is based on the Markov decision process, which is an iterative process where an agent interacts with an environment. This process comprises four elements: the state  $s$ , action  $a$ , policy  $\pi(a|s)$  and reward  $r$ . The action is an operation that is applied by the agent. The policy represents the action selection strategy of the agent. In other words, at each iteration step, the agent obtains a state and chooses an action according to the policy. Owing to the action taken, the state in the environment is then changed and the agent receives an immediate reward, which is feedback showing the usefulness of the action taken. The agent gains experience from the collected states, actions and rewards after several iterations, which it uses to find an optimal policy  $\pi^*(a|s)$  that maximises the long-term reward. In DRL, a deep neural network is used to obtain the optimal policy.

This study presents a physics-constrained deep reinforcement learning (PCDRL) model that is built on reinforcement learning with pixel-wise rewards (PixelRL) (Furuta, Inoue & Yamasaki 2020), which is a CNN-based multi-agent DRL method for image processing (Li *et al.* 2020; Vassilo *et al.* 2020; Jarosik *et al.* 2021). In PixelRL, the asynchronous advantage actor–critic (A3C) algorithm (Mnih *et al.* 2016) is applied for learning policies, which determine the actions that are represented by the choice of basic filters for each pixel. In other words, each pixel has one agent in PixelRL. A model that applies optimal policies to change the velocity values is investigated in this study by choosing the suitable actions for each point in the flow field at an instant, as shown in figure 1. In contrast to image processing problems that require the target data in the training process (Furuta *et al.* 2020), the physics of the flow represented by the governing equations and the known boundary conditions are used to train the model.

Let  $\chi_{i,j}^n$  be the value of an instantaneous velocity component at iteration step  $n$  and in location  $(i, j)$  of the field. Herein, each location has its own agent with a policy  $\pi_{i,j}(a_{i,j}^n|s_{i,j}^n)$ , where  $a_{i,j}^n \in \mathcal{A}$ , which is a pre-defined set of actions (Appendix D).

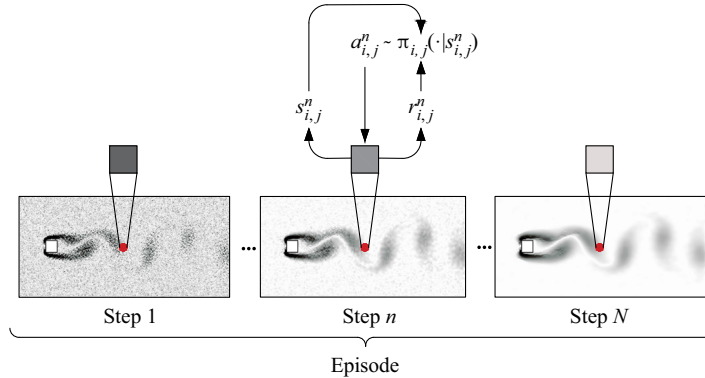


Figure 1. Learning process in the PCDRL model. Each agent at each iteration step in the episode obtains a state from a point in the flow, calculates the reward and applies an action according to the policy.

Each agent obtains the next state, that is,  $s_{i,j}^{n+1}$ , and reward  $r_{i,j}^{n+1}$  from the environment by taking the action  $a_{i,j}^n$ .

Physical constraints represented by the momentum equation, the pressure Poisson equation and the known boundary conditions are embedded in the reward function, which enables the model to follow an optimal denoising strategy that results in changing the noisy data to the true flow field distribution. Hence, the objective of the model is to learn the policy that maximises the expected long-term rewards:

$$\pi_{i,j}^* = \operatorname{argmax}_{\pi_{i,j}} E_{\pi_{i,j}} \left( \sum_{n=1}^N \gamma^{(n-1)} r_{i,j}^n \right), \quad (2.1)$$

where  $\gamma^{(n-1)}$  is the  $(n - 1)$ th power of the discount factor  $\gamma$ , which determines the weights of the immediate rewards in the iteration steps. In this study, the value of  $\gamma$  is set to 0.95.

The combination of the momentum equation,

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u}, \quad (2.2)$$

and the pressure Poisson equation,

$$\nabla \cdot (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla^2 p, \quad (2.3)$$

is used to build the physics-based immediate reward,  $(r_{i,j}^n)_{Physics}$ , where  $\mathbf{u}$ ,  $p$ ,  $t$  and  $\nu$  are the velocity vector, pressure (divided by density), time and kinematic viscosity, respectively.

At each iteration step, the pressure field is obtained by numerically solving (2.3). Herein, the pressure gradient calculated from the pressure field  $((\nabla p_{i,j}^n)_{Poisson})$  is used in (2.2) such that

$$(r_{i,j}^n)_{Physics} = - \left| \left( \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} - \nu \nabla^2 \mathbf{u} \right)_{i,j}^n + (\nabla p_{i,j}^n)_{Poisson} \right|. \quad (2.4)$$

The pressure integration in (2.3) is done by using a Poisson solver that applies a standard five-point scheme (second-order central difference method) (Van der Kindere *et al.* 2019) with the initial pressure field being estimated from numerically integrating the pressure gradient obtained from the initial noisy data in (2.2) (van Oudheusden *et al.*

2007). Notably, the central difference method is applied for all the spatial discretisations. Regarding the temporal discretisation, for the first and the last time steps in each training mini-batch, the forward difference and the backward difference are used, respectively, and the central difference is applied for the other time steps. Furthermore, Neumann and Dirichlet boundary conditions according to each case used in this study are enforced in the calculations.

Additionally, the velocity values obtained after each action  $a_{i,j}^n$  are directly made divergence-free by applying Helmholtz–Hodge decomposition (Bhatia *et al.* 2013) using Fourier transformation. Furthermore, the known boundary conditions are used to obtain the boundary conditions-based immediate reward  $(r_{i,j}^n)_{BC}$  for the velocity by considering the absolute error of the reconstructed data at the boundaries of the domain.

Thus, the combined immediate reward function can be expressed as

$$r_{i,j}^n = (r_{i,j}^n)_{Physics} + \beta(r_{i,j}^n)_{BC}, \quad (2.5)$$

where  $\beta$  is a weight coefficient and its value is empirically set to 20.

This approach considers the convergence of the model output to satisfy the governing equations and boundary conditions as a measure of the model performance without the need for the target training data. Furthermore, the reward function is designed to mimic the denoising process of PIV velocity field data without the need for measured pressure field data in the model. Nine iteration steps for each episode, that is,  $N = 9$ , are used in this study. In addition, the size of the training mini-batch is set to 4. The model is applied to direct numerical simulation (DNS)-based data (corrupted by different levels of additive zero-mean Gaussian noise) and real noisy PIV data of two-dimensional flow around a square cylinder at Reynolds numbers  $Re_D = 100$  and  $200$ , respectively. Herein,  $Re_D = u_\infty D/\nu$ , where  $u_\infty$  and  $D$  are the free stream velocity and the cylinder width, respectively. Details regarding the source code of the proposed model, A3C, PixelRL and the selected pre-defined denoising action set can be found in [Appendices A, B, C and D](#), respectively.

### 3. Data description and preprocessing

#### 3.1. Synthetic data

DNS data of a two-dimensional flow around a square cylinder at a Reynolds number of  $Re_D = 100$  are considered as an example of synthetic data. The open-source computational fluid dynamics finite-volume code OpenFOAM-5.0x is used to perform the DNS. The domain size is set to  $x_D \times y_D = 20 \times 15$ , where  $x$  and  $y$  are the streamwise and spanwise directions, respectively. The corresponding grid size is  $381 \times 221$ . Local mesh refinement is applied using the stretching mesh technique near the cylinder walls. Uniform inlet velocity and pressure outlet boundary conditions are applied to the inlet and the outlet of the domain, respectively. No-slip boundary conditions are applied to the cylinder walls and the symmetry plane to the sides of the domain. The dimensionless time step of the simulation, that is,  $u_\infty \Delta t/D$ , is set to  $10^{-2}$ . The DNS data are corrupted by additive zero-mean Gaussian noise, that is,  $\mathcal{S} \sim \mathcal{N}(0, \sigma^2)$ , where  $\mathcal{S}$ ,  $\mathcal{N}$  and  $\sigma^2$  represent the noise, the normal distribution and the variance, respectively. The signal-to-noise ratio, for which a large value yields a low noise level, is used to evaluate the noise level. Herein,  $SNR = \sigma_{DNS}^2/\sigma_{noise}^2$ , where  $\sigma_{DNS}^2$  and  $\sigma_{noise}^2$  denote the variances of the DNS and the noise data, respectively. Three levels of noise are applied,  $1/SNR = 0.01, 0.1$  and  $1$ . The interval between the collected snapshots of the flow fields is set to 10 times the simulation time step; 1000 snapshots are used for training the model, whereas 200 snapshots are used for testing the performance of the model.

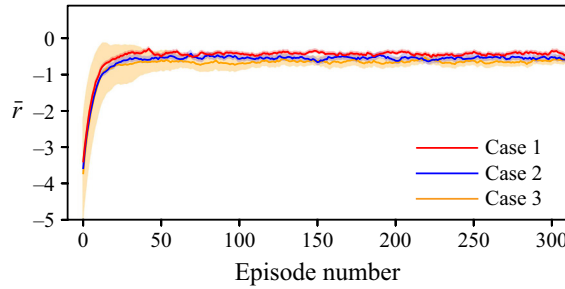


Figure 2. Progress of the mean reward during the training process. Cases 1, 2 and 3 represent the noisy DNS data at noise levels  $1/SNR = 0.01, 0.1$  and  $1$ , respectively.

### 3.2. Experimental data

Two PIV experiments are performed to generate noisy and clear (uncorrupted) data (for comparison) of flow over a square cylinder to investigate the performance of the proposed PCDRL model on real experimental data. The noisy data are generated by using a return-type water channel. The test section size of the water channel is  $1\text{ m}$  (length)  $\times$   $0.35\text{ m}$  (height)  $\times$   $0.3\text{ m}$  (width). The free stream velocity is set to  $0.02\text{ m s}^{-1}$ , with a corresponding  $Re_D$  of  $200$ . The background noise is generated at relatively high levels due to the external noise and the sparse honeycomb configuration of the water channel. The channel was seeded by polyamide12 seed particles from INTECH SYSTEMS with  $50\text{ }\mu\text{m}$  diameter. A high-speed camera (FASTCAM Mini UX 50) and a continuous laser with a  $532\text{ nm}$  wavelength are used to build the complete PIV system. The snapshot frequency is set to  $24\text{ Hz}$ . Herein,  $2000$  and  $500$  instantaneous flow fields are used for the model training and testing of its performance, respectively. Meanwhile, clear data of the flow are generated by using a return-type wind tunnel. The test section size of the wind tunnel is  $1\text{ m}$  (length)  $\times$   $0.25\text{ m}$  (height)  $\times$   $0.25\text{ m}$  (width). The free stream velocity is set to  $0.29\text{ m s}^{-1}$ , with a corresponding  $Re_D$  of  $200$ . The turbulence intensity of the free stream is less than  $0.8\%$ . The wind tunnel is seeded by olive oil droplets generated by a TSI 9307 particle generator. The PIV system used in the wind tunnel comprises a two-pulsed laser (Evergreen, EVG00070) and a CCD camera (VC-12MX) with  $4096 \times 3072$  pixel resolution. Herein, the snapshot frequency is set to  $15\text{ Hz}$ . In the water channel and wind tunnel experiments, the square cylinder model comprised an acrylic board and the cross-section of the model is set to  $1 \times 1\text{ cm}$ . The model is not entirely transparent. Thus, a shadow region is generated in the area below the bluff body when the laser goes through the model.

## 4. Results and discussion

### 4.1. Performance of the model

The capability of the PCDRL model to denoise flow fields is investigated in this study qualitatively and quantitatively by using the DNS and PIV data. The model is primarily applied to DNS-based data. Figure 2 shows the progress of the mean reward during the training process, that is,

$$\bar{r} = \frac{1}{IJN} \sum_{i=1}^I \sum_{j=1}^J \sum_{n=1}^N r_{i,j}^n. \quad (4.1)$$



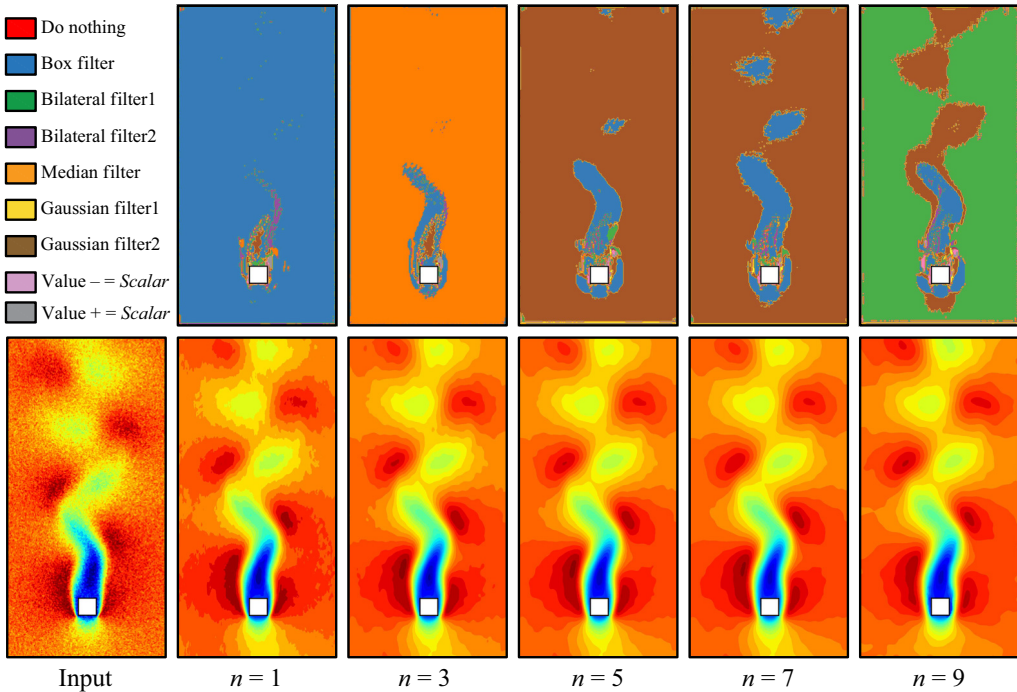


Figure 3. Action map of the prediction process for an instantaneous streamwise velocity field. The top panels show the types of filters used in the process and the action map in each iteration step, and the bottom panels show the corresponding velocity field. Results for the DNS noisy data at noise level  $1/SNR = 0.1$ .

The solid line and light area indicate  $\bar{r}$  and the standard deviation of the reward at nine iteration steps, respectively. As shown in the figure, the reward for the three different noise levels rapidly increases and approaches its optimal level after a few episodes. This finding indicates that the agents in PixelRL learn the policy in a few episodes in the training process, compared with the other multi-agent networks, because they share the information represented by the network parameters and also because of the averaged gradients (Furuta *et al.* 2020). Thus, this approach can significantly reduce the computational cost of the model. Furthermore, as expected, the magnitude of the optimal  $\bar{r}$  decreases with the increase in noise level.

Figure 3 shows a visual overview of the prediction process of the PCDRL model. The figure reveals that the choice of filters changes with the spatial distribution of the velocity data and also with each iteration step in the episode. The visualisation of the action map is one of the model features, providing additional access to the model considering the action strategy. Furthermore, it can be seen that the action map is strongly correlated with the physics of the flow, which is represented in this case by the vortex shedding behind the square cylinder.

The instantaneous denoised flow data are presented in figure 4(a) by employing the vorticity field ( $\omega$ ). The figure reveals that the model shows a remarkable capability to reconstruct the flow field even when using an extreme level of noise in the input data of the model.

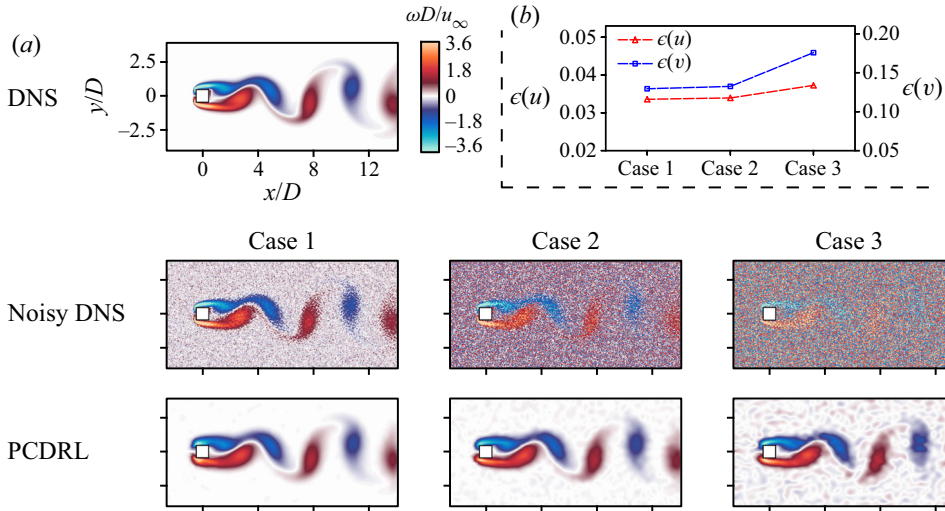


Figure 4. (a) Instantaneous vorticity field; (b) relative  $L_2$ -norm error of the reconstructed velocity fields. Cases 1, 2 and 3 represent the noisy DNS data at noise levels  $1/SNR = 0.01, 0.1$  and  $1$ , respectively.

The general reconstruction accuracy of the model is examined via the relative  $L_2$ -norm error of the reconstructed velocity fields,

$$\epsilon(\chi) = \frac{1}{K} \sum_{k=1}^K \frac{\|\chi_k^{PCDRL} - \chi_k^{DNS}\|_2}{\|\chi_k^{DNS}\|_2}, \quad (4.2)$$

where  $\chi_k^{PCDRL}$  and  $\chi_k^{DNS}$  represent the predicted velocity component and the ground truth (DNS) one, respectively, and  $K$  is the number of test snapshots. Figure 4(b) shows that the values of the error are relatively small for the velocity components and are proportional to the increase in noise level.

Figure 5 shows probability density function (p.d.f.) plots of the streamwise ( $u$ ) and spanwise ( $v$ ) velocity components. Herein, the p.d.f. plots obtained from the reconstructed velocity fields are generally consistent with those obtained from DNS, indicating that the proposed model could successfully recover the actual distribution of flow data. Furthermore, the scatter plots of the maximum instantaneous velocity values in all the test data are presented in figure 6. The figure reveals that the predicted data are generally in commendable agreement with the DNS data for the entire range of each velocity component, with a slight reduction in the consistency as the noise level increases.

The power spectral density (PSD) of the streamwise velocity fluctuations at two different locations is plotted in figure 7 to examine the capability of the model to reproduce the spectral content of the flow. Commendable agreement with the DNS results can be observed, with a slight deviation in the high frequencies for the noise level  $1/SNR = 1$ .

The statistics of the velocity fields, represented by the spanwise profiles of the root mean square of the velocity ( $u_{rms}, v_{rms}$ ) and Reynolds shear stress ( $\overline{u'v'}$ ), are presented in figure 8. The figure shows an accurate reconstruction of the statistics at two different streamwise locations in the domain, indicating that the model could successfully reproduce the statistics of the flow despite the extreme noise level.

The model performance is further examined by using actual noisy PIV data. The reconstructed instantaneous vorticity field is shown in figure 9(a). The figure reveals that the model could successfully denoise the velocity fields with commendable accuracy



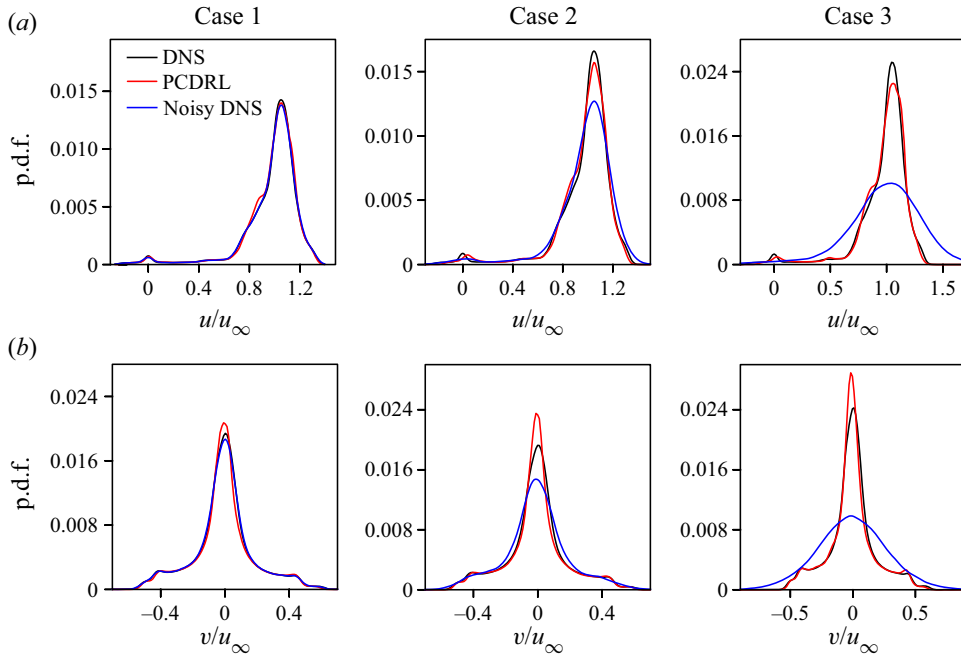


Figure 5. Probability density function plots of the (a) streamwise and (b) spanwise velocity components. Cases 1, 2 and 3 represent the noisy DNS data at noise levels  $1/SNR = 0.01, 0.1$  and  $1$ , respectively.

considering the noisy input data to the model. In addition, the model shows capability of recovering the corrupted regions in the flow due to the experimental set-up. Furthermore, the relative difference of the spanwise profile of the vorticity root mean square ( $\omega_{rms}$ ) between the reconstructed data and the clear PIV data ( $\varepsilon(\omega_{rms})$ ) presented in figure 9(b) shows that the results from the model exhibit a smooth behaviour that is generally consistent with that of the clear PIV data. These results indicate that the PCDRL model can be practically applied to noisy PIV data.

#### 4.2. POD and DMD results

In this section, the accuracy of the results from the PCDRL model is examined in terms of flow decomposition. First, the results of applying POD to the denoised data are compared with the POD results of the ground truth data. Figure 10 shows the contour plots of the leading POD modes for the vorticity field obtained from the DNS data. As can be observed from the figure, for the case of the highest noise level, i.e.  $1/SNR = 1$ , all the seven true leading modes can be recovered using the denoised data, while only three modes can be recovered using the noisy data and no distinguishable features can be seen for the other modes. Furthermore, the energy plots in figure 11 represented by the normalised POD eigenvalues show that even for the case of the flow with the highest noise level, the energy contribution values of the POD modes are consistent with those obtained from the ground truth DNS data. As expected, the results from the noisy data reveal a different behaviour, especially for the cases of noise levels  $1/SNR = 0.1$  and  $1$ . Figure 12 shows a reconstructed instantaneous vorticity field of the DNS data using the first ten POD modes. As shown in the figure, the result from the PCDRL model reveals a commendable reconstruction accuracy as compared with the ground truth DNS results, whereas the result obtained

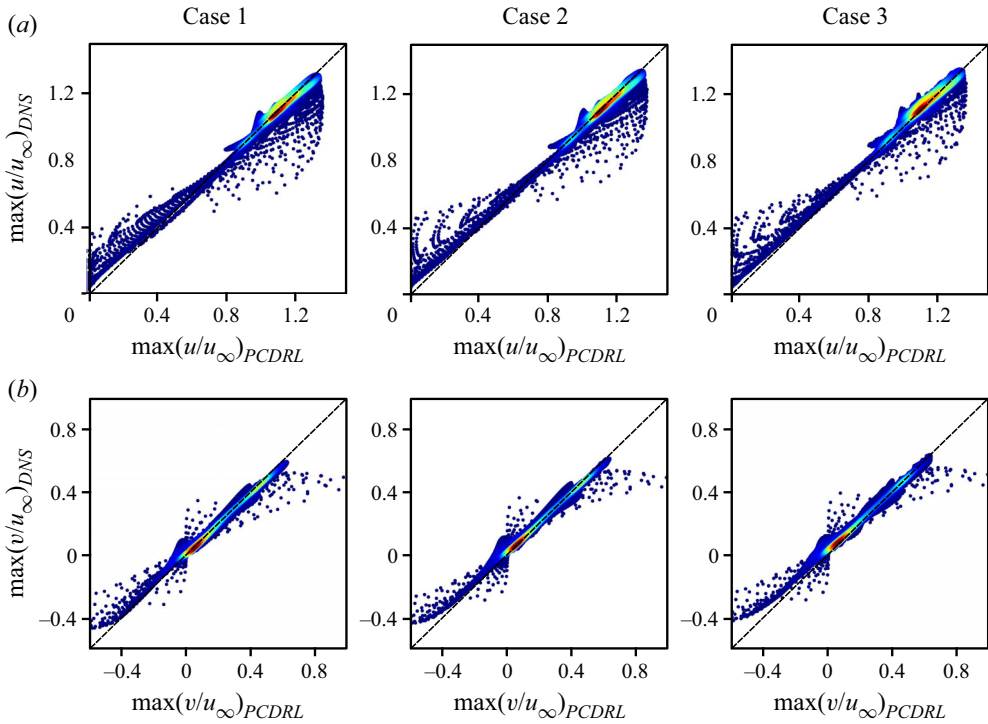


Figure 6. Scatter plots of the maximum instantaneous values of the (a) streamwise and (b) spanwise velocity components. Cases 1, 2 and 3 represent the results from the PCDRL model using noisy DNS data at noise levels  $1/SNR = 0.01, 0.1$  and  $1$ , respectively. The contour colours (from blue to red) are proportional to the density of points in the scatter plot.

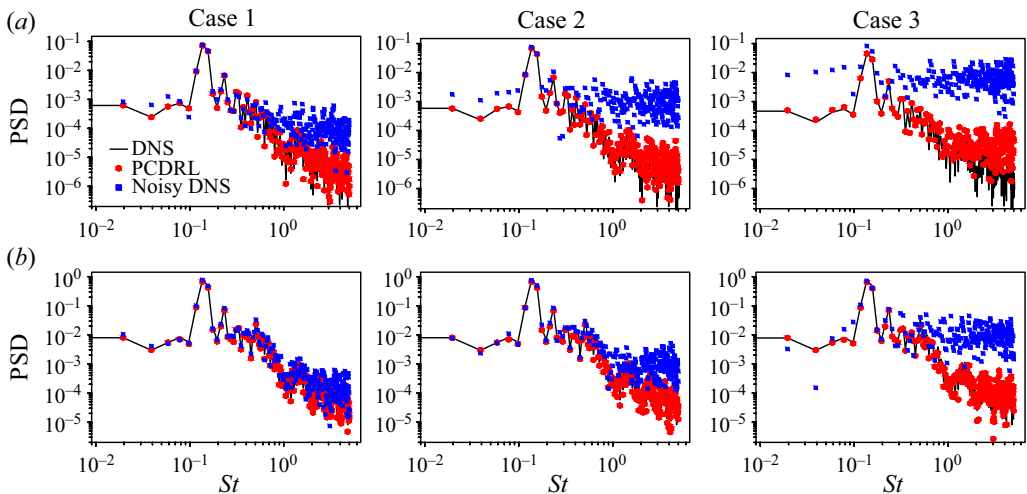


Figure 7. Power spectral density plots of the streamwise velocity fluctuations at two different locations: (a)  $(x/D, y/D) = (1, 1)$  and (b)  $(x/D, y/D) = (6, 1)$ . The dimensionless frequency is represented by the Strouhal number,  $St = fD/u_\infty$ , where  $f$  is the frequency. Cases 1, 2 and 3 represent the noisy DNS data at noise levels  $1/SNR = 0.01, 0.1$  and  $1$ , respectively.

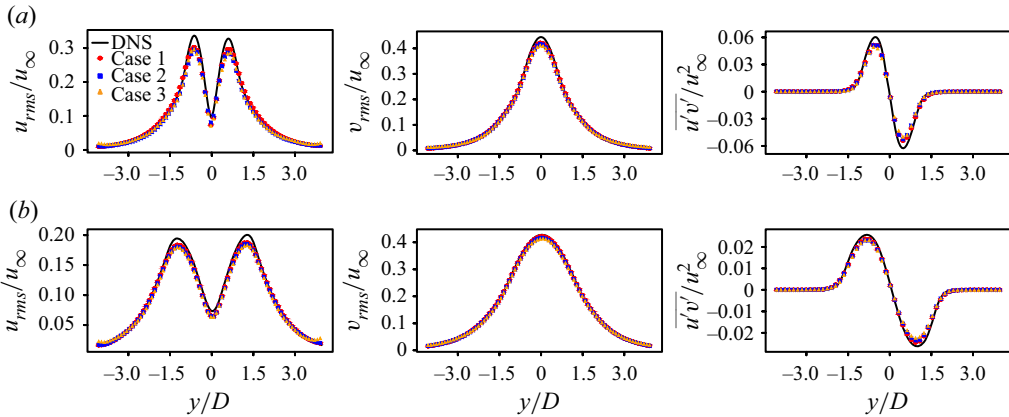


Figure 8. Spanwise profiles of flow statistics  $u_{rms}$  (left column),  $v_{rms}$  (middle column) and  $\overline{u'v'}$  (right column) at two different streamwise locations: (a)  $x/D = 3$ ; (b)  $x/D = 6$ . Cases 1, 2 and 3 represent the results from the PCDRL model using noisy DNS data at noise levels  $1/SNR = 0.01, 0.1$  and  $1$ , respectively.

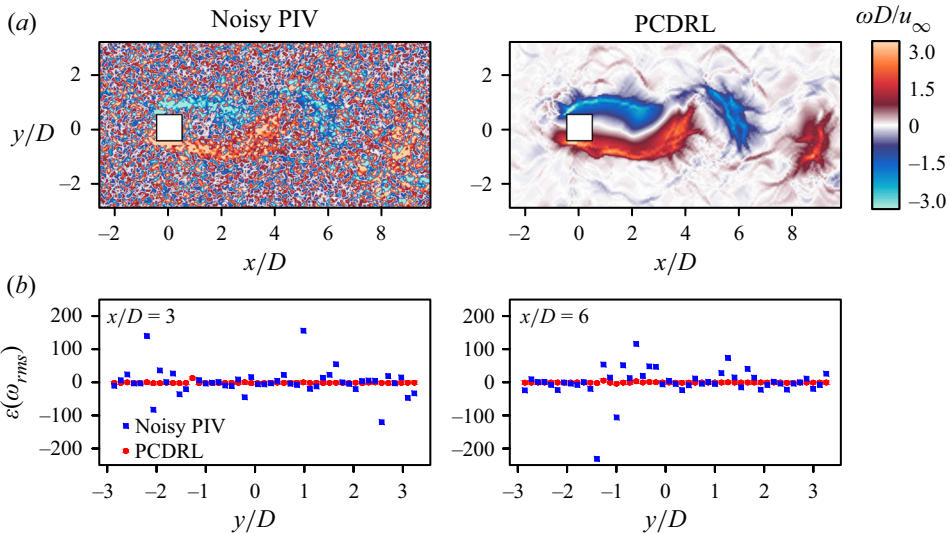


Figure 9. (a) Instantaneous vorticity field of the noisy (left column) and denoised PIV data obtained from the PCDRL model (right column); (b) relative difference of the spanwise profile of the vorticity root mean square at two different streamwise locations.

from the noisy data indicates the limitation of POD in recovering the flow with the right physics.

Similar results can be obtained by applying the POD to the PIV data. As can be observed from figure 13, the seven leading POD modes obtained from the denoised PIV data are relatively consistent with the modes obtained from the clear PIV data, considering that the clean PIV data are obtained using a different experimental set-up, whereas the noisy PIV data fail to recover the modes after the third mode. Notably, the shadow region is clearly visible in some of the modes obtained from the clear PIV data, whereas no such region can be seen in the modes obtained from the denoised data. This is consistent with results from figure 9(a). The results from figure 14 further indicate the ability of the model to reconstruct the flow data with POD modes that generally have a behaviour

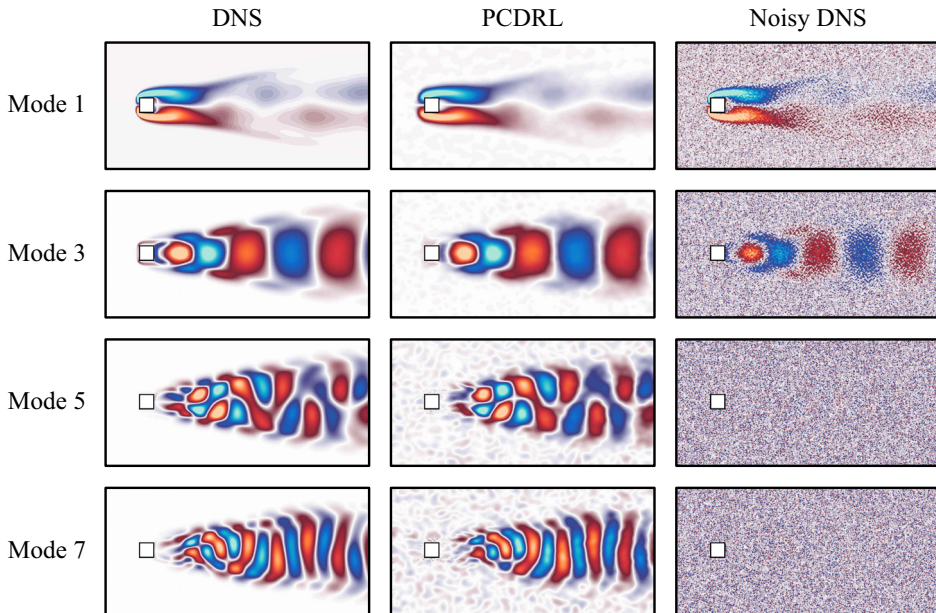


Figure 10. Leading POD modes obtained from the DNS data. Results from the ground truth DNS (left column), PCDRL (middle column) and noisy data with  $1/SNR = 1$  (right column).

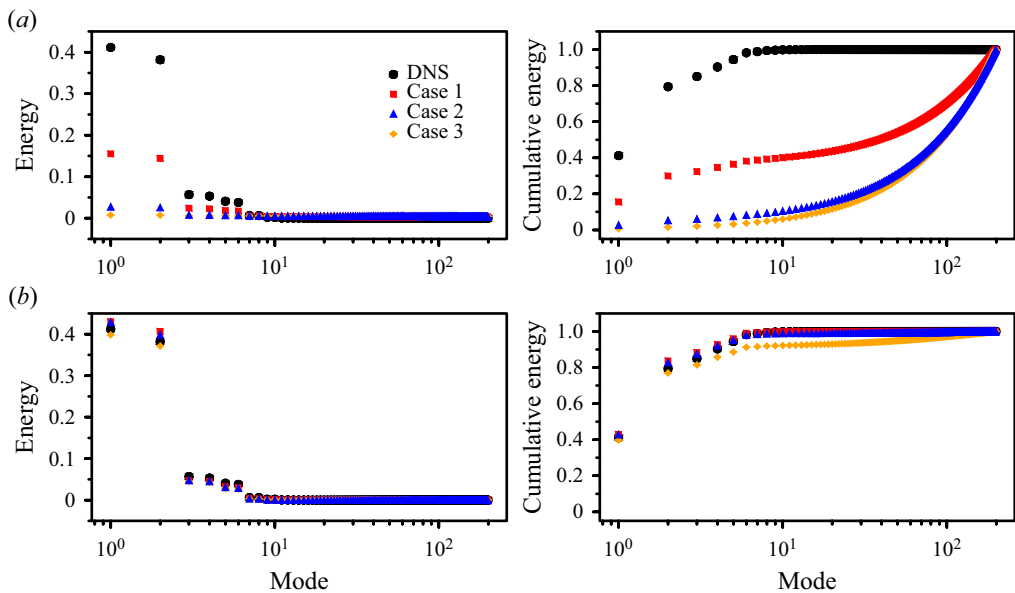


Figure 11. Normalised energy (left column) and cumulative energy (right column) of the POD modes obtained from the DNS data: (a) noisy data, where Cases 1, 2 and 3 represent the noisy DNS data at noise levels  $1/SNR = 0.01, 0.1$  and  $1$ , respectively; (b) results from the PCDRL model.

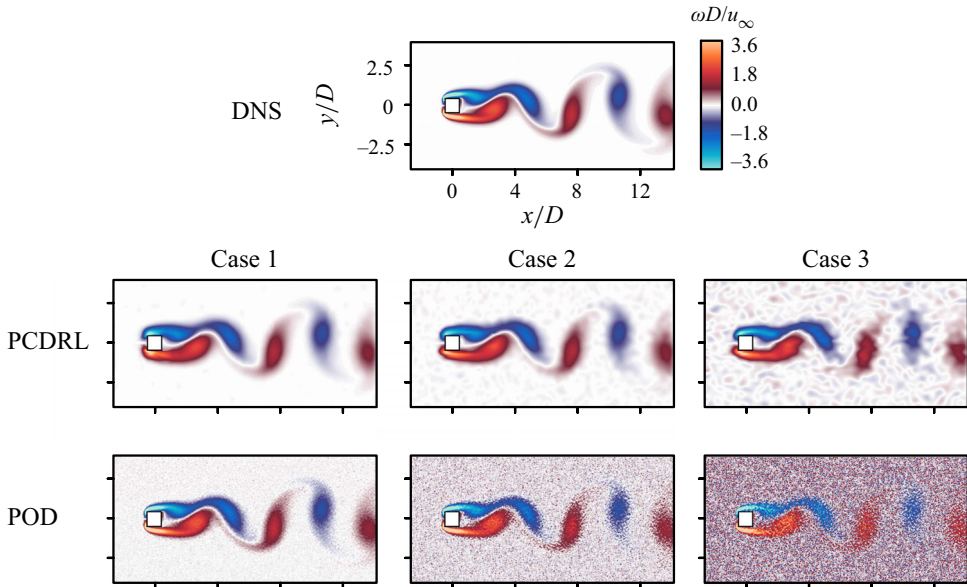


Figure 12. Reconstructed instantaneous vorticity field obtained from the DNS data using the first ten POD modes. Cases 1, 2 and 3 represent the results of using noisy DNS data at noise levels  $1/SNR = 0.01, 0.1$  and  $1$ , respectively.

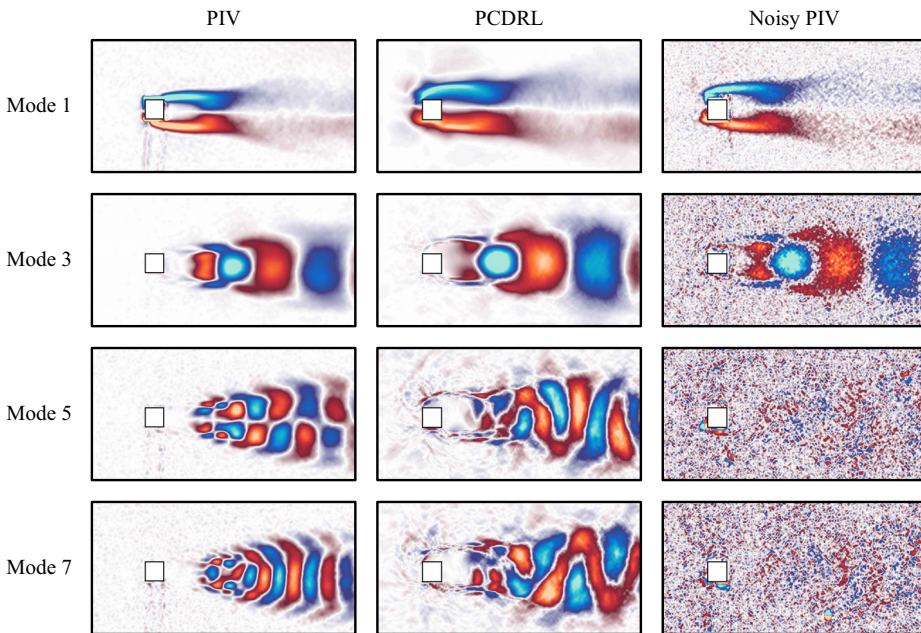


Figure 13. Leading POD modes obtained from the PIV data. Results from clear PIV (left column), PCDRL (middle column) and noisy PIV data (right column).



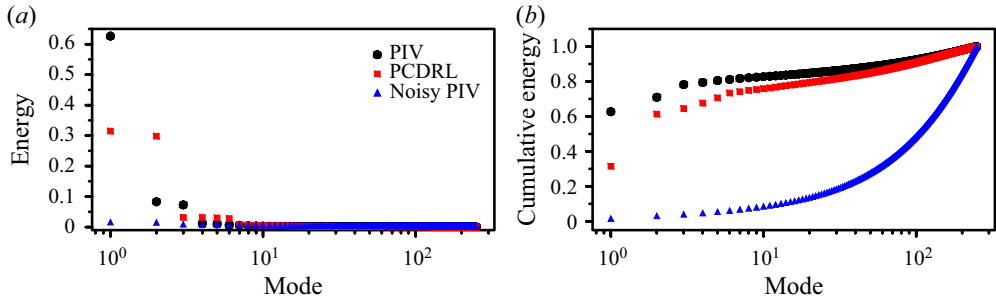


Figure 14. (a) Normalised energy and (b) cumulative energy of the POD modes obtained from the PIV data.

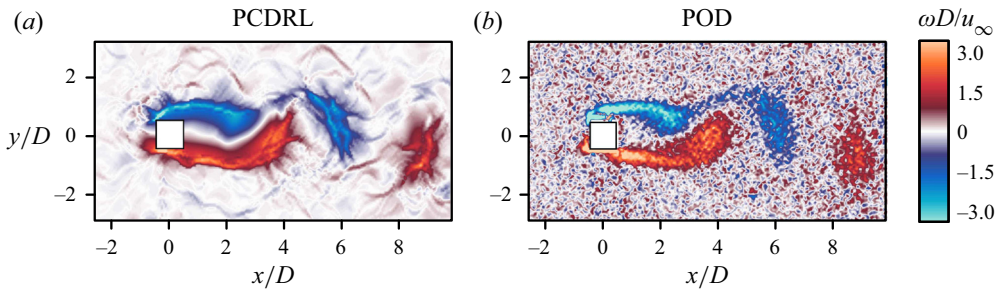


Figure 15. Reconstructed instantaneous vorticity field obtained from the results of the (a) PCDRL model and (b) noisy PIV data using the first ten POD modes.

similar to that of the clear PIV data. Furthermore, as shown in figure 15, the reconstructed instantaneous vorticity field using the first ten modes of the denoised data shows a realistic flow behaviour that is expected from the case of flow around a cylinder.

To further investigate the dynamics of the denoised flow data, DMD is then applied to the flow data. As shown in figure 16, even in the case of the DNS data corrupted with the level of noise  $1/SNR = 1$ , the DMD eigenvalues of the vorticity field show a behaviour close to that of the ground truth DNS data, whereas for the noisy data the eigenvalues scatter inside the unit circle plot, indicating a non-realistic behaviour of the system.

As for the denoised PIV data, figure 17 reveals that the eigenvalues also show good agreement with those from the clear PIV data. Notably, the leading DMD eigenvalues of the clear PIV data are not exactly located on the circumference of the unit circle as in the case of DNS data. This behaviour can be attributed to the fact that DMD is known to be sensitive to noise (Bagheri 2014; Dawson *et al.* 2016; Hemati *et al.* 2017; Scherl *et al.* 2020) and, unlike the DNS data, the clear PIV contains a relatively low level of noise, which can affect the flow decomposition.

## 5. Conclusions

This study has proposed a DRL-based method to reconstruct flow fields from noisy data. The PixelRL method is used to build the proposed PCDRL model, wherein an agent that applies actions represented by basic filters according to a local policy is assigned to each point in the flow. Hence, the proposed model is a multi-agent model. The physical constraints represented by the momentum equation, the pressure Poisson equation and the boundary conditions are used to build the reward function. Hence, the PCDRL model

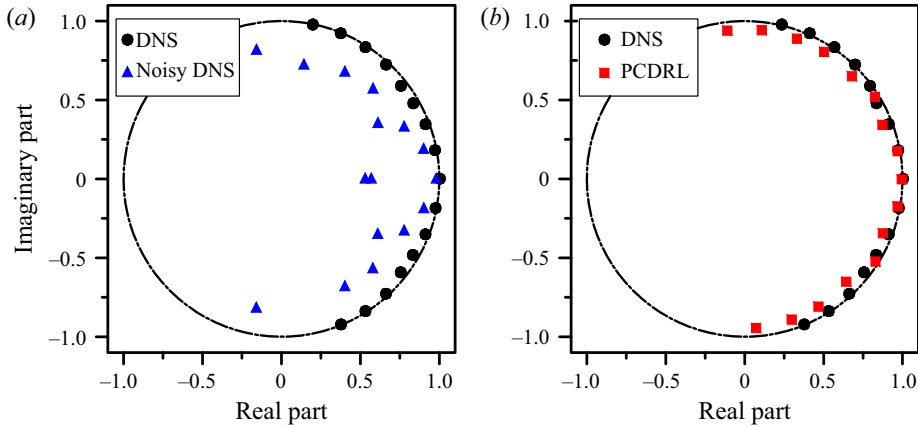


Figure 16. (a) DMD eigenvalues of the noisy DNS data at noise level  $1/SNR = 1$  and (b) the results from the PCDRL model visualised on the unit circle.

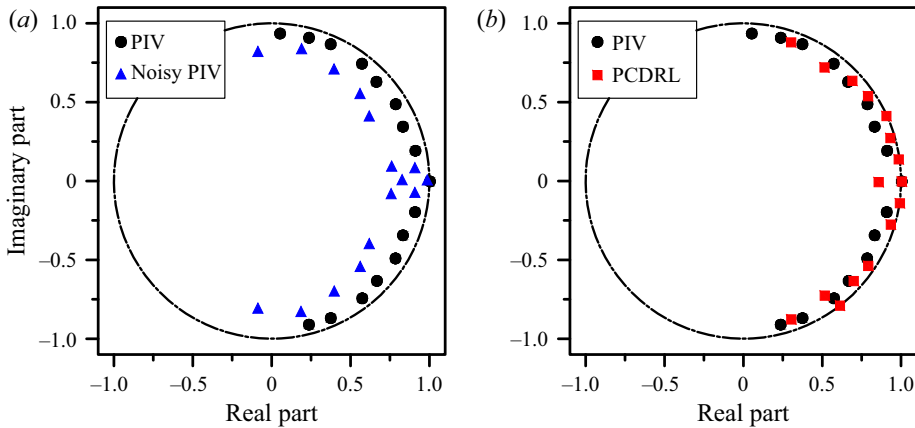


Figure 17. (a) DMD eigenvalues of the noisy PIV data and (b) the results from the PCDRL model visualised on the unit circle.

is label-training data-free; that is, target data are not required for the model training. Furthermore, visualisation and interpretation of the model performance can be easily achieved owing to the model set-up.

The model performance was first investigated using DNS-based noisy data with three different noise levels. The instantaneous results and the flow statistics revealed a commendable reconstruction accuracy of the model. Furthermore, the spectral content of the flow was favourably recovered by the model, with reduced accuracy as the noise level increased. Additionally, the reconstruction error had relatively low values, indicating the general reconstruction accuracy of the model.

Real noisy and clear PIV data were used to examine the model performance. Herein, the model demonstrated its capability to recover the flow fields with the appropriate behaviour.

Furthermore, the accuracy of the denoised flow data from both DNS and PIV was investigated in terms of flow decomposition by means of POD and DMD. Most of the leading POD modes that describe the main features (coherent structures) of the flow were successfully recovered with commendable accuracy and outperformed the results

of directly applying POD to the noisy data. Additionally, the DMD eigenvalues obtained from the denoised flow data exhibited behaviour similar to that of true DMD modes. These results further indicate the model’s ability to recover the flow data with most of the flow physics.

This study demonstrates that the combination of DRL, the physics of the flow, which is represented by the governing equations, and prior knowledge of the flow boundary conditions can be effectively used to recover high-fidelity flow fields from noisy data. This approach can be further extended to the reconstruction of three-dimensional turbulent flow fields, for which more sophisticated DRL models with more complex spatial filters are needed. Applying such models to flow reconstruction problems can result in considerable reduction in the experimental and computational costs.

**Funding.** This work was supported by the ‘Human Resources Program in Energy Technology’ of the Korea Institute of Energy Technology Evaluation and Planning (KETEP), granted financial resources from the Ministry of Trade, Industry & Energy, Republic of Korea (no. 20214000000140). In addition, this work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (no. 2019R1I1A3A01058576).

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

**Author ORCIDs.**

- 📧 Mustafa Z. Yousif <https://orcid.org/0000-0002-5542-5474>;
- 📧 Meng Zhang <https://orcid.org/0000-0003-2393-5215>;
- 📧 Linqi Yu <https://orcid.org/0000-0002-5674-6261>;
- 📧 Yifan Yang <https://orcid.org/0009-0002-9114-1951>;
- 📧 Haifeng Zhou <https://orcid.org/0009-0007-2886-706X>;
- 📧 HeeChang Lim <https://orcid.org/0000-0001-8504-0797>.

**Appendix A. Open-source code**

The open-source library Pytorch 1.4.0 (Paszke *et al.* 2019) was used for the implementation of the PCDRL model. The source code of the model is available at [click here](#).

**Appendix B. Asynchronous advantage actor–critic (A3C)**

The asynchronous advantage actor–critic (A3C) (Mnih *et al.* 2016) algorithm is applied in this work. A3C is a variant of the actor–critic algorithm, which combines the policy- and value-based networks to improve performance. [Figure 18](#) shows that the actor generates an action  $a^n$  for the given state  $s^n$  based on the current policy, whilst the critic provides the value function  $V(s^n)$  to evaluate the effectiveness of the action.

Based on understanding of the actor–critic algorithm, A3C also has two sub-networks: the policy and value networks. Herein,  $\theta_p$  and  $\theta_v$  are used to represent the parameters of each network. The gradients of  $\theta_p$  and  $\theta_v$  can be calculated as follows:

$$R^n = r^n + \gamma r^{n+1} + \dots + \gamma^{(N-1)} r^N + \gamma^{(N)} V(s^N), \tag{B1}$$

$$d\theta_v = \nabla_{\theta_v} (R^n - V(s^n))^2, \tag{B2}$$

$$A(a^n, s^n) = R^n - V(s^n), \tag{B3}$$

$$d\theta_p = -\nabla_{\theta_p} \log \pi(a^n, s^n) A(a^n, s^n), \tag{B4}$$

where  $A(a^n, s^n)$  is the advantage.

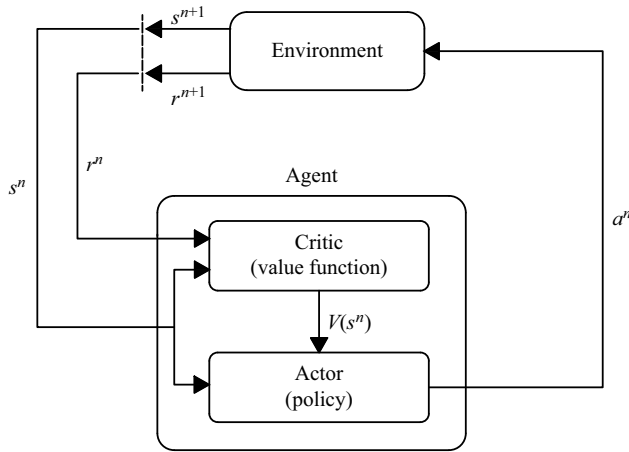


Figure 18. Architecture of the actor-critic algorithm.

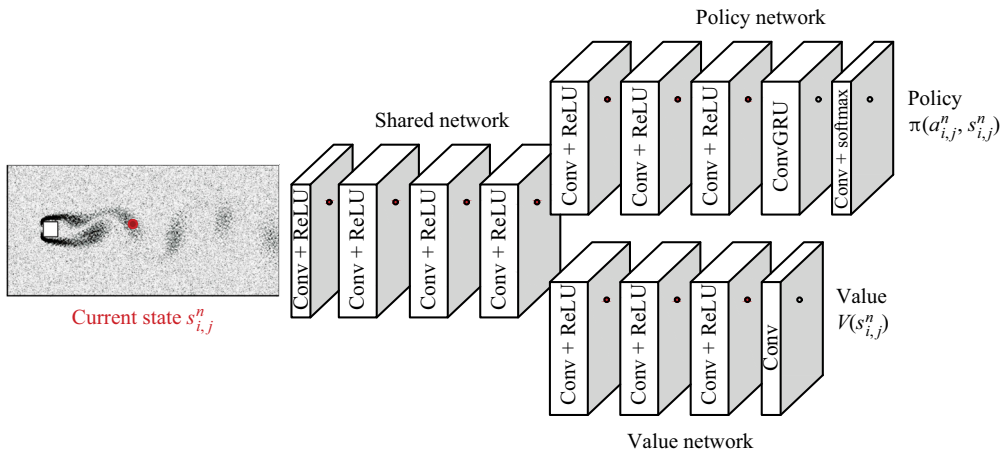


Figure 19. Architecture of the fully convolutional A3C.

### Appendix C. PixelRL

A3C is modified in this study to a fully convolutional form (Furuta *et al.* 2020), and its architecture can be found in figure 19. Through this approach, all the agents share the same parameters, which saves on computational cost and trains the model more efficiently compared with the case where agents need to train their models individually. The size of the receptive field can also affect the performance of the CNN network, and a large receptive field can result in superior capture connections between points. Therefore, a receptive field ( $3 \times 3$ ) is used in the architecture; that is, the outputs of the policy and value networks at a specific pixel will be affected by the pixel and its surrounding neighbour pixels. Figure 19 shows that the input flow field data first pass through four convolutional and leaky rectified linear unit (ReLU) (Goodfellow, Bengio & Courville 2016) layers and are then inputted to the policy and value networks, respectively. The policy network comprises three convolutional layers with a ReLU activation function, a ConvGRU layer and a convolutional layer with a SoftMax activation function (Goodfellow *et al.* 2016), and its output is the policy. The first three layers of the value network are the same as the first

Number	Action type	Parameters
1	Do nothing	filter size = $5 \times 5$
2	Box filter	filter size = $5 \times 5$
3	Bilateral filter1	filter size = $5 \times 5$ , $\sigma_c = 1.0$ , $\sigma_s = 5.0$
4	Bilateral filter2	filter size = $5 \times 5$ , $\sigma_c = 0.1$ , $\sigma_s = 5.0$
5	Median filter	filter size = $5 \times 5$
6	Gaussian filter1	filter size = $5 \times 5$ , $\sigma = 1.5$
7	Gaussian filter2	filter size = $5 \times 5$ , $\sigma = 0.5$
8	Value + = <i>Scalar</i>	
9	Value - = <i>Scalar</i>	

Table 1. Action set for the denoising process.

three layers of the policy network, and the value function is finally obtained through the convolutional layer with a linear function.

The gradients of the parameters  $\theta_p$  and  $\theta_v$  are then defined on the basis of the architecture of the fully convolutional A3C:

$$\mathbf{R}^n = \mathbf{r}^n + \gamma \mathbf{W} * \mathbf{r}^{n+1} + \dots + \gamma^{(N-1)} \mathbf{W}^{(N-1)} * \mathbf{r}^N + \gamma^{(N)} \mathbf{W}^{(N)} * V(\mathbf{s}^N), \quad (\text{C1})$$

$$d\theta_v = \nabla_{\theta_v} \frac{1}{I \times J} \mathbf{1}^\top \{(\mathbf{R}^n - V(\mathbf{s}^n)) \odot (\mathbf{R}^n - V(\mathbf{s}^n))\} \mathbf{1}, \quad (\text{C2})$$

$$\mathbf{A}(\mathbf{a}^n, \mathbf{s}^n) = \mathbf{R}^n - V(\mathbf{s}^n), \quad (\text{C3})$$

$$d\theta_p = -\nabla_{\theta_p} \frac{1}{I \times J} \mathbf{1}^\top \{\log \pi(\mathbf{a}^n, \mathbf{s}^n) \odot \mathbf{A}(\mathbf{a}^n, \mathbf{s}^n)\} \mathbf{1}, \quad (\text{C4})$$

where  $\mathbf{R}^n$ ,  $\mathbf{r}^n$ ,  $V(\mathbf{s}^n)$ ,  $\mathbf{A}(\mathbf{a}^n, \mathbf{s}^n)$  and  $\pi(\mathbf{a}^n, \mathbf{s}^n)$  are the matrices whose  $(i, j)$ th elements are  $R_{i,j}^n$ ,  $r_{i,j}^n$ ,  $V(s_{i,j}^n)$ ,  $A(a_{i,j}^n, s_{i,j}^n)$  and  $\pi(a_{i,j}^n, s_{i,j}^n)$ , respectively. Here,  $*$  is the convolution operator,  $\mathbf{1}$  is the all-ones vector and  $\odot$  is element-wise multiplication. Additionally,  $\mathbf{W}$  is a convolution filter weight, which is updated simultaneously with  $\theta_p$  and  $\theta_v$  such that

$$d\mathbf{W} = -\nabla_{\mathbf{W}} \frac{1}{I \times J} \mathbf{1}^\top \{\log \pi(\mathbf{a}^n, \mathbf{s}^n) \odot \mathbf{A}(\mathbf{a}^n, \mathbf{s}^n)\} \mathbf{1} + \nabla_{\mathbf{W}} \frac{1}{I \times J} \mathbf{1}^\top \{(\mathbf{R}^n - V(\mathbf{s}^n)) \odot (\mathbf{R}^n - V(\mathbf{s}^n))\} \mathbf{1}. \quad (\text{C5})$$

Notably, after the agents complete their interaction with the environment, the gradients are acquired simultaneously, which means that the number of asynchronous threads is one; that is, A3C is equivalent to advantage actor-critic (A2C) in the current study (Clemente, Castejón & Chandra 2017).

#### Appendix D. Denoising action set

The action set for removing the noise of the flow fields is shown in table 1. The agent can take the following nine possible actions: do nothing, apply six classical image filters or plus/minus a *Scalar*. The actions in this study are discrete and determined empirically. The table shows that the parameters  $\sigma_c$ ,  $\sigma_s$  and  $\sigma$  represent the filter standard deviation in the colour space, the coordinate space and the Gaussian kernel, respectively. The *Scalar* in the 8th and 9th actions is determined on the basis of the difference between the variance of the clear and noisy data.



REFERENCES

- BAGHERI, S. 2014 Effects of weak noise on oscillating flows: linking quality factor, Floquet modes, and Koopman spectrum. *Phys. Fluids* **26** (9), 094104.
- BHATIA, H., NORGARD, G., PASCUCCI, V. & BREMER, P.-T. 2013 The Helmholtz-Hodge decomposition—a survey. *IEEE Trans. Vis. Comput. Graphics* **19** (8), 1386–1404.
- BRUNTON, S.L., NOACK, B.R. & KOUMOUTSAKOS, P. 2020 Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **52** (1), 477–508.
- CLEMENTE, A.V., CASTEJÓN, H.N. & CHANDRA, A. 2017 Efficient parallel methods for deep reinforcement learning. [arXiv:1705.04862](https://arxiv.org/abs/1705.04862).
- DAWSON, S.T.M., HEMATI, M.S., WILLIAMS, M.O. & ROWLEY, C.W. 2016 Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. *Exp. Fluids* **57**, 42.
- DISCETTI, S. & LIU, Y. 2022 Machine learning for flow field measurements: a perspective. *Meas. Sci. Technol.* **34** (2), 021001.
- DURAISAMY, K., IACCARINO, G. & XIAO, H. 2019 Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.* **51** (1), 357–377.
- FATHI, M.F., BAKHSHINEJAD, A., BAGHAIE, A. & D’SOUZA, R.M. 2018 Dynamic denoising and gappy data reconstruction based on dynamic mode decomposition and discrete cosine transform. *Appl. Sci.* **8** (9), 1515.
- FUKAMI, K., FUKAGATA, K. & TAIRA, K. 2019 Super-resolution reconstruction of turbulent flows with machine learning. *J. Fluid Mech.* **870**, 106–120.
- FURUTA, R., INOUE, N. & YAMASAKI, T. 2020 PixelRL: fully convolutional network with reinforcement learning for image processing. *IEEE Trans. Multimedia* **22** (7), 1704–1719.
- GAO, H., SUN, L. & WANG, J.-X. 2021 Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. *Phys. Fluids* **33** (7), 073603.
- GARNIER, P., VIQUERAT, J., RABAULT, J., LARCHER, A., KUHNLE, A. & HACHEM, E. 2021 A review on deep reinforcement learning for fluid mechanics. *Comput. Fluids* **225**, 104973.
- GOODFELLOW, I., BENGIO, Y. & COURVILLE, A. 2016 *Deep Learning*. MIT Press.
- GUASTONI, L., GÜEMES, A., IANIRO, A., DISCETTI, S., SCHLATTER, P., AZIZPOUR, H. & VINUESA, R. 2021 Convolutional-network models to predict wall-bounded turbulence from wall quantities. *J. Fluid Mech.* **928**, A27.
- GÜEMES, A., VILA, C.S. & DISCETTI, S. 2022 Super-resolution GANs of randomly-seeded fields. *Nat. Mach. Intell.* **4**, 1165–1173.
- GUNES, H. & RIST, U. 2007 Spatial resolution enhancement/smoothing of stereo-particle-image-velocimetry data using proper-orthogonal-decomposition-based and kriging interpolation methods. *Phys. Fluids* **19** (6), 064101.
- HE, C. & LIU, Y. 2017 Proper orthogonal decomposition-based spatial refinement of TR-PIV realizations using high-resolution non-TR-PIV measurements. *Exp. Fluids* **58** (7), 86.
- HEMATI, M.S., ROWLEY, C.W., DEEM, E.A. & CATTAFESTA, L.N. 2017 De-biasing the dynamic mode decomposition for applied Koopman spectral analysis of noisy datasets. *Comput. Fluid Dyn.* **31**, 349–368.
- HICKLING, T., ZENATI, A., AOUF, N. & SPENCER, P. 2022 Explainability in deep reinforcement learning, a review into current methods and applications. [CoRR.abs/2207.01911](https://arxiv.org/abs/2207.01911).
- JAROSIK, P., LEWANDOWSKI, M., KLIMONDA, Z. & BYRA, M. 2021 Pixel-wise deep reinforcement learning approach for ultrasound image denoising. In *2021 IEEE International Ultrasonics Symposium (IUS)*, pp. 1–4. IEEE.
- KIM, H., KIM, J., WON, S. & LEE, C. 2021 Unsupervised deep learning for super-resolution reconstruction of turbulence. *J. Fluid Mech.* **910**, A29.
- LECUN, Y., BENGIO, Y. & HINTON, G. 2015 Deep learning. *Nature* **521**, 436–444.
- LI, W., FENG, X., AN, H., NG, X.Y. & ZHANG, Y.-J. 2020 MRI reconstruction with interpretable pixel-wise operations using reinforcement learning. *Proc. AAAI Conf. Artif. Intell.* **34** (1), 792–799.
- LIU, B., TANG, J., HUANG, H. & LU, X.-Y. 2020 Deep learning methods for super-resolution reconstruction of turbulent flows. *Phys. Fluids* **32** (2), 025105.
- LUMLEY, J.L. 1967 The structure of inhomogeneous turbulent flows. In *Atmospheric Turbulence and Radio Wave Propagation*, pp. 166–177. Nauka.
- MNIH, V., BADIA, A.P., MIRZA, M., GRAVES, A., LILICRAP, T., HARLEY, T., SILVER, D. & KAVUKCUOGLU, K. 2016 Asynchronous methods for deep reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 1928–1937. PMLR, MLResearchPress.
- NONOMURA, T., SHIBATA, H. & TAKAKI, R. 2019 Extended-Kalman-filter-based dynamic mode decomposition for simultaneous system identification and denoising. *PLoS ONE* **14**, 1–46.

- NOVATI, G., DE LAROUSSILHE, H.L. & KOUMOUTSAKOS, P. 2021 Automating turbulence modelling by multi-agent reinforcement learning. *Nat. Mach. Intell.* **3**, 87–96.
- VAN OUDHEUSDEN, B.W., SCARANO, F., ROOSENBOOM, E.W.M., CASIMIRI, E.W.F. & SOUVEREIN, L. 2007 Evaluation of integral forces and pressure fields from planar velocimetry data for incompressible and compressible flows. *Exp. Fluids* **43**, 153–162.
- PASZKE, A., *et al.* 2019 Pytorch: an imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 8026–8037. Curran Associates.
- RABAUULT, J., KUCHTA, M., JENSEN, A., RÉGLADE, U. & CERARDI, N. 2019 Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *J. Fluid Mech.* **865**, 281–302.
- SCHERL, I., STROM, B., SHANG, J.K., WILLIAMS, O., POLAGYE, B.L. & BRUNTON, S.L. 2020 Robust principal component analysis for modal decomposition of corrupt fluid flows. *Phys. Rev. Fluids* **5** (5), 054401.
- SCHMID, P.J. 2010 Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **656**, 5–28.
- VAN DER KINDERE, J.W., LASKARI, A., GANAPATHISUBRAMANI, B. & DE KAT, R. 2019 Pressure from 2D snapshot PIV. *Exp. Fluids* **60**, 32.
- VASSILO, K., HEATWOLE, C., TAHA, T. & MEHMOOD, A. 2020 Multi-step reinforcement learning for single image super-resolution. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2160–2168. IEEE.
- VÉTEL, J., GARON, A. & PELLETIER, D. 2011 Denoising methods for time-resolved PIV measurements. *Exp. Fluids* **51** (4), 893–916.
- VINUESA, R. & BRUNTON, S.L. 2022 Enhancing computational fluid dynamics with machine learning. *Nat. Comput. Sci.* **2** (6), 358–366.
- VIQUERAT, J., MELIGA, P., LARCHER, A. & HACHEM, E. 2022 A review on deep reinforcement learning for fluid mechanics: an update. *Phys. Fluids* **34**, 111301.
- VIQUERAT, J., RABAUULT, J., KUHNLE, A., GHRAIEB, H., LARCHER, A. & HACHEM, E. 2021 Direct shape optimization through deep reinforcement learning. *J. Comput. Phys.* **428**, 110080.
- YOUSIF, M.Z., YU, L., HOYAS, S., VINUESA, R. & LIM, H. 2023a A deep-learning approach for reconstructing 3D turbulent flows from 2D observation data. *Sci. Rep.* **13**, 2529.
- YOUSIF, M.Z., ZHANG, M., YU, L., VINUESA, R. & LIM, H.C. 2023b A transformer-based synthetic-inflow generator for spatially developing turbulent boundary layers. *J. Fluid Mech.* **957**, A6.
- YU, L., YOUSIF, M.Z.G., ZHANG, M., HOYAS, S., VINUESA, R. & LIM, H. 2022 Three-dimensional enhanced super-resolution generative adversarial network for super-resolution reconstruction of turbulent flows with tricubic interpolation-based transfer learning. *Phys. Fluids* **34**, 125126.